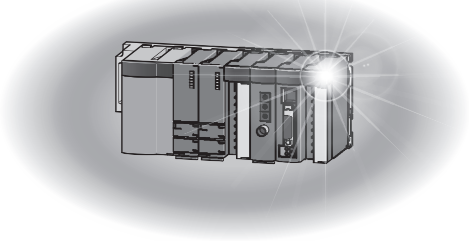
MITSUBISHI ELECTRIC

Programmable Controller

MELSEC Q series

C Controller Module User's Manual (Hardware Design, Function Explanation)

-Q12DCCPU-V(Basic mode)
-Q06CCPU-V
-Q06CCPU-V-B
-SW3PVC-CCPU-E



●SAFETY PRECAUTIONS●

(Read these precautions before using this product.)

Before using this product, please read this manual and the relevant manuals carefully and pay full attention to safety to handle the product correctly.
The precautions given in this manual are concerned with this product only. For the safety precautions of the programmable controller system, refer to the user's manual for the CPU module used.

In this manual, the safety precautions are classified into two levels: "⚠WARNING" and "⚠CAUTION".

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

Under some circumstances, failure to observe the precautions under "⚠ CAUTION" may lead to serious consequences.
Observe the precautions of both levels because they are important for personal and system safety.

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

[Design Precautions]

⚠WARNING

● Configure safety circuits external to the C Controller module to ensure that the entire system operates safely even when a fault occurs in the external power supply or the C Controller module. Failure to do so may result in an accident due to an incorrect output or malfunction.

(1) Configure external safety circuits, such as an emergency stop circuit, protection circuit, and protective interlock circuit for forward/reverse operation or upper/lower limit positioning.

(2) If the following status (a) or (b) occurs, the system will behave accordingly.

(a) When overcurrent or overvoltage protection of the power supply module is activated, the outputs (Y) from the user program and writing to the buffer memory are disabled, and all outputs are turned off.

(b) When the C Controller module detects an error such as a watchdog timer error by the self-diagnostic function, the outputs (Y) from the user program and writing to the buffer memory are disabled. Whether to hold or turn off all outputs can be set by parameters.

All outputs may turn on when an error occurs in the part, such as I/O control part, where the C Controller module cannot detect any error.
To ensure safety operation in such a case, provide a safety mechanism or a fail-safe circuit external to the C Controller module. For a fail-safe circuit example, refer to "CHAPTER 6 PREPARATORY PROCEDURES AND SETTING" in this manual.

[Design Precautions]

⚠ WARNING

(3) Outputs may remain on or off due to a failure of an output module relay or transistor. Configure an external circuit for monitoring output signals that could cause a serious accident.

● In an output module, when a load current exceeding the rated current or an overcurrent caused by a load short-circuit flows for a long time, it may cause smoke and fire. To prevent this, configure an external safety circuit, such as a fuse.

● Configure a circuit so that the C Controller system is turned on first and then the external power supply. If the external power supply is turned on first, an accident may occur due to an incorrect output or malfunction.

● For the operating status of each station after a communication failure, refer to the relevant manuals for the network.
Incorrect output or malfunction may result in an accident.

● For the following controls, configure an interlock circuit in the user program to ensure that the entire system will always operate safely.
(1) Changing data of a running C Controller module from the development environment (personal computer) connected
(2) Changing the operating status
(3) Operation from the development environment (personal computer)
Especially, in the case of a control from an external device to a remote C Controller module, immediate action cannot be taken for a problem on the C Controller module due to a communication failure.
To prevent this, configure an interlock circuit in the user program, and determine corrective actions to be taken between the external device and C Controller module in case of a communication failure.

● When configuring the system, do not leave any slot empty on the base unit.
If a slot is to be left empty, attach a blank cover (QG60).
Failure to do so may cause internal parts of the mounted modules to fly apart during a short-circuit test or when an excess current or voltage is applied to an external I/O part by mistake.

[Design Precautions]

⚠ CAUTION

- ● Do not install the control lines or communication cables together with the main circuit lines or power cables.
  Keep a distance of 100mm (3.94 inches) or more between them.
  Failure to do so may result in malfunction due to noise.
- ● When a device such as a lamp, heater, or solenoid valve is controlled through an output module, a large current (approximately ten times greater than normal) may flow when the output is turned from off to on. Take measures such as replacing the module with one having a sufficient current rating.
- ● After the C Controller module is powered on or is reset, the time taken to enter the RUN status varies depending on the system configuration, parameter setting, and/or program size.
  Design circuits so that the entire system will always operate safely, regardless of the time.

[Security Precautions]

⚠ WARNING

- ● To maintain the security (confidentiality, integrity, and availability) of the C Controller module and the system against unauthorized access, denial-of-service (DoS) attacks, computer viruses, and other cyberattacks from external devices via the network, take appropriate measures such as firewalls, virtual private networks (VPNs), and antivirus solutions.

[Installation Precautions]

⚠ CAUTION

- ● Use the C Controller module in an environment that meets the general specifications in this manual. Failure to do so may result in an electric shock, fire, malfunction, or damage to or deterioration of the product.
- ● To mount the module, while pressing the module mounting lever located in the lower part of the module, fully insert the module fixing projection(s) into the hole(s) in the base unit and press the module until it snaps into place.
  Incorrect mounting may cause malfunction, failure or drop of the module.
  When using the module in an environment of frequent vibrations, fix the module with a screw.
  Tighten the screw within the specified torque range.
  Undertightening can cause drop of the screw, short circuit or malfunction.
  Overtightening can damage the screw and/or module, resulting in drop, short circuit, or malfunction.

[Installation Precautions]

⚠ CAUTION

- When using an extension cable, connect it to the connector of the base unit or module securely.
  Check the connection for looseness.
  Poor contact may cause incorrect input or output.

- When using a CompactFlash card, fully insert it into the CompactFlash card slot.
  Check that it is inserted completely.
  Poor contact may cause malfunction.

- Shut off the external power supply for the system in all phases before mounting or removing the module.
  Failure to do so may result in damage to the product.

- Do not directly touch any conductive part of the module.
  Doing so can cause malfunction or failure of the module.

[Wiring Precautions]

⚠ WARNING

- Shut off the external power supply for the system in all phases before wiring.
  Failure to do so may result in electric shock or damage to the product.

- After wiring, attach the included terminal cover to the module before turning it on for operation.
  Failure to do so may result in electric shock.

[Wiring Precautions]

⚠ CAUTION

- Ground the FG and LG terminals to the protective ground conductor dedicated to the programmable controller.
  Failure to do so may result in electric shock or malfunction.

- Prevent foreign matter such as dust or wire chips from entering the module.
  Such foreign matter can cause a fire, failure, or malfunction.

- Check the rated voltage and terminal layout before wiring to the module, and connect the cables correctly.
  Connecting a power supply with a different voltage rating or incorrect wiring may cause a fire or failure.

- Securely connect the Ethernet and RS-232 cables to respective connectors of the C Controller module.

- Do not install the control lines or the communication cables together with the main circuit lines or power cables.
  Doing so may result in malfunction due to noise.

- Tighten the terminal screw within the specified torque range.
  Undertightening can cause short circuit, fire, or malfunction.
  Overtightening can damage the screw and/or module, resulting in drop, short circuit, or malfunction.

- Place the communication and power cables in a duct or clamp them.
  If not, dangling cable may swing or inadvertently be pulled, resulting in damage to the module or cables or malfunctions due to poor contact.

- When disconnecting a communication or power cable from the module, do not hold and pull the cable part.
  Doing so may cause malfunction or damage to the module or cable.
  For a cable with connector, hold the connected connector by hand to disconnect the cable.
  For a screw-type cable, loosen the screw to disconnect the cable.

- Do not connect outputs of multiple power supply modules in parallel.
  The power supply modules may produce heat, causing a fire or failure.

- Connectors for external connection must be crimped or pressed with the tool specified by the manufacturer, or must be correctly soldered.
  For the tool, refer to the user's manual for the relevant I/O module.
  Incomplete connections could result in short circuit, fire or malfunction.

[Startup and Maintenance Precautions]

⚠WARNING

- Do not touch any terminal while power is on.
  Doing so will cause electric shock.

- Correctly connect the battery connector.
  Do not charge, disassemble, heat, short-circuit, solder, or throw the battery into the fire.
  Doing so will cause the battery to produce heat, explode, or ignite, resulting in injury and fire.

- Shut off the external power supply for the system in all phases before cleaning the module or retightening the terminal screws or module fixing screws.
  Failure to do so may result in electric shock.
  Undertightening the terminal screws can cause short circuit or malfunction.
  Overtightening can damage the screw and/or module, resulting in drop, short circuit, or malfunction.

- For the following controls, configure an interlock circuit in the user program to ensure that the entire system will always operate safely.
  (1) Changing data of the running C Controller module from the development environment (personal computer) connected
  (2) Changing the operating status
  (3) Operation from the development environment (personal computer)
  Especially, in the case of a control from an external device to a remote C Controller module, immediate action cannot be taken for a problem on the C Controller module due to a communication failure.
  To prevent this, configure an interlock circuit in the user program, and determine corrective actions to be taken between the external device and C Controller module in case of a communication failure.

[Startup and Maintenance Precautions]

⚠CAUTION

- Do not disassemble or modify the modules.
  Doing so may cause failure, malfunction, injury, or a fire.

- Before performing online operations (especially, program modification, forced output, and operation status change) for the running C Controller module from the peripheral connected, read relevant manuals carefully and ensure the safety.
  Improper operation may damage machines or cause accidents.

- Shut off the external power supply for the system in all phases before mounting or removing the module.
  Failure to do so may cause the module to fail or malfunction.

- After the first use of the product, do not mount/remove the module to/from the base unit more than 50 times. (IEC 61131-2 compliant)
  Exceeding the limit of 50 times may cause malfunction.

- Use any radio communication device such as a cellular phone or PHS (Personal Handy-phone System) more than 25cm (9.85 inches) away in all directions from the C Controller system.
  Failure to do so may cause malfunction.

- Do not drop or apply shock to the battery to be installed in the module.
  Doing so may damage the battery, causing the battery fluid to leak inside the battery.
  If the battery is dropped or any shock is applied to it, dispose of it without using.

- Before handling the module, touch a grounded metal object to discharge the static electricity from the human body.
  Failure to do so may cause the module to fail or malfunction.

[Disposal Precautions]

⚠ CAUTION

● When disposing of this product, treat it as industrial waste.
When disposing of batteries, separate them from other wastes according to the local regulations.
(For details of the Battery Directive in EU countries, refer to Appendix 4.)

[Transportation Precautions]

⚠ CAUTION

● When transporting lithium batteries, follow the transportation regulations. (For details of the regulated models, refer to Appendix 3.)

●CONDITIONS OF USE FOR THE PRODUCT●

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

(2) The PRODUCT has been designed and manufactured for the purpose of being used in general industries.
MITSUBISHI ELECTRIC SHALL HAVE NO RESPONSIBILITY OR LIABILITY (INCLUDING, BUT NOT LIMITED TO ANY AND ALL RESPONSIBILITY OR LIABILITY BASED ON CONTRACT, WARRANTY, TORT, PRODUCT LIABILITY) FOR ANY INJURY OR DEATH TO PERSONS OR LOSS OR DAMAGE TO PROPERTY CAUSED BY the PRODUCT THAT ARE OPERATED OR USED IN APPLICATION NOT INTENDED OR EXCLUDED BY INSTRUCTIONS, PRECAUTIONS, OR WARNING CONTAINED IN MITSUBISHI ELECTRIC USER'S, INSTRUCTION AND/OR SAFETY MANUALS, TECHNICAL BULLETINS AND GUIDELINES FOR the PRODUCT.
("Prohibited Application")
Prohibited Applications include, but not limited to, the use of the PRODUCT in;

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

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

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

REVISIONS

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

Print date	*Manual number	Revision
Jun., 2009	SH(NA)-080766ENG-A	First edition
Jan., 2010	SH(NA)-080766ENG-B	Partial correction PRECAUTIONS
Aug., 2010	SH(NA)-080766ENG-C	Partial correction SAFETY PRECAUTIONS, ABOUT MANUALS, HOW TO USE THIS MANUAL, GENERIC TERMS AND ABBREVIATIONS, GLOSSARY, Section 1.1, 2.1.3, 2.1.4, 2.2.1, 2.2.2, 2.3, 2.4, 3.2, 3.4.2, 4.1, 4.4, 4.5, 4.13, 5.2 to 5.4, 6.2, 6.11.2, 8.1.1, 8.2.2, 10.1, 11.1.4, 11.3, 13.1, 13.2, 13.6, 14.1, 15.2.4, 16.3, 16.4.1, 16.5.1, Appendix 8, Appendix 9.1, Appendix 10, Appendix 11 Addition CONDITIONS OF USE FOR THE PRODUCT, Section 2.1.5, 4.4, 4.16, 6.8.5, Chapter 9, Section 16.4.2, Appendix 6, Appendix 7, Appendix 9 Modification Section 2.1.5 → Section 2.1.6, Section 4.4 to 4.14 → Section 4.5 to 4.15, Chapter 9 to 15 → Chapter10 to 16, Appendix 6 to 8, Appendix7 to 8 → Appendix10 to 11
Mar., 2011	SH(NA)-080766ENG-D	Partial correction OPERATING PRECAUTIONS, Section 1.1, 2.1.3, 2.3, 2.4, 3.3.1, 3.3.2, 4.3.3, 4.15, 4.16, 5.3.2, 5.4.2, 6.3, 8.2.2, 16.4.1, Appendix 7, Appendix 9, Appendix 12 Addition Appendix 10 Modification Appendix 10 to 11 → Appendix 11 to 12
Jul., 2011	SH(NA)-080766ENG-E	Partial correction ABOUT MANUALS, GENERIC TERMS AND ABBREVIATIONS, Section 2.1.2, 2.1.4, 2.2.1, 2.2.2, 11.1.4
Jan., 2012	SH(NA)-080766ENG-F	Partial correction Section 2.2.1, 2.2.2, 2.4, 3.1
Dec., 2012	SH(NA)-080766ENG-G	Partial correction GENERIC TERMS AND ABBREVIATIONS, OPERATING PRECAUTIONS, Section 2.2.1, 2.3, 3.2, 3.3.2, 4.5, 6.3, 6.4.1, 6.6, 10.1, 11.1, 11.2.1, 11.3, 12.1, 13.3.3, 13.3.4, 13.9, 16.2.3

Print date	*Manual number	Revision
Oct., 2013	SH(NA)-080766ENG-H	Partial correction ABOUT MANUALS, HOW TO USE THIS MANUAL, GENERIC TERMS AND ABBREVIATIONS, GLOSSARY, PACKING LIST, CHAPTER 1, Section 1.1, 2.1.2, 2.2.1, 2.2.2, 2.3, 2.4, 2.5, 2.5.1, 4.14, 6.3, 6.4.1, 6.7, 6.10, 6.10.1, 6.10.2, 6.12.1, 8.2.1, 10.1, 11.1, 12.1, 13.3.1, 16.2, 16.2.2, 16.2.7, 16.2.8, 16.2.11, 16.5.1, Appendix7 Addition PRODUCT ORGANIZATION Delete Appendix 11 Modification Appendix 12 → Appendix 11
Dec., 2013	SH(NA)-080766ENG-I	Partial correction GENERIC TERMS AND ABBREVIATIONS, Section 10.1
Mar., 2014	SH(NA)-080766ENG-J	Partial correction GENERIC TERMS AND ABBREVIATIONS, Section 11.2.1, Section 12.1, Section 12.7, Section 13.3.2, Section 13.3.3, Section 13.3.4, Section 13.9, Appendix11
Jul., 2014	SH(NA)-080766ENG-K	Partial correction ABOUT MANUALS, GENERIC TERMS AND ABBREVIATIONS, CHAPTER 1, Section 11.1, Section 11.3
Feb., 2015	SH(NA)-080766ENG-L	Partial correction Section 2.2.2, Section 6.7.1
Dec., 2015	SH(NA)-080766ENG-M	Partial correction PACKING LIST, Section 2.1.3, Section 2.5.1, Section 3.2, Section 6.4.1, Appendix 2.1, WARRANTY
Jun., 2016	SH(NA)-080766ENG-N	Partial correction SAFETY PRECAUTIONS, GENERIC TERMS AND ABBREVIATIONS, Section 6.6, Section 6.11 Addition DISCONTINUED MODELS
Oct., 2017	SH(NA)-080766ENG-O	Partial correction ABOUT MANUALS, GENERIC TERMS AND ABBREVIATIONS, CHAPTER 1, Section 6.7.1, Section 11.1, Appendix 11
July, 2020	SH(NA)-080766ENG-P	Partial correction SAFETY PRECAUTIONS, CONDITIONS OF USE FOR THE PRODUCT
Dec., 2020	SH(NA)-080766ENG-Q	Partial correction Section 6.8.1

Print date	*Manual number	Revision
Nov., 2023	SH(NA)-080766ENG-R	Partial correction GENERIC TERMS AND ABBREVIATIONS, INTRODUCTION, DISCONTINUED MODELS, Section 6.6, Appendix 11 Addition INFORMATION AND SERVICES, COPYRIGHTS

Japanese manual version SH-080764-T

This manual confers no industrial property rights or any rights of any other kind, nor does it confer any patent licenses. Mitsubishi Electric Corporation cannot be held responsible for any problems involving industrial property rights which may occur as a result of using the contents noted in this manual.

OPERATING PRECAUTIONS

This section provides the following precautions.

Precautions for	Refer to
System configurations	Page A-13
Standard RAM, standard ROM, and CompactFlash card	Page A-14
Battery	Page A-16
Clock setting	Page A-17
Starting the C Controller module	Page A-17
Network configurations	Page A-18
Relation between system tasks and the system watchdog timer, user watchdog timer, and link device refresh cycles	Page A-18
For the Wind River Systems product	Page A-18

(1) Precautions for system configurations

(a) Supported modules
In a system configuration where a C Controller module is used, some main and extension base units cannot be used (☞ Page 2-16, Section 2.1.6), and also use of some modules is restricted by their function versions (☞ Page 2-17, Section 2.2).

(b) Multiple CPU system

1) Applicable CPU modules
For the CPU modules that can be used with a C Controller module in a multiple CPU system, refer to the following.
☞ Page 11-11, Section 11.2.1

2) MT Developer connection (Version 00Y or earlier) in a multiple CPU system
When a multiple CPU system includes a C Controller module set as CPU No.1 and a Motion CPU, a communication test of MT Developer (Version 00Y or earlier) cannot be performed. (The Motion CPU communication test error (error code: 12288) will occur.)
To perform the communication test from MT Developer, use either of the following methods.
- Use MT Developer Version 00Z or later.
- When using MT Developer Version 00Y or earlier, modify the multiple CPU system configuration so that CPU No.1 is a programmable controller CPU.

(c) Time adjustment and time notification of GOT
In a multiple CPU system configuration, when time adjustment or time notification is performed in a GOT connected to C Controller modules for CPU No. 2 to No. 4, it is executed to CPU No. 1.
In this case, to match the clock data between the C Controller modules assigned to CPU No. 2 to No. 4 and the GOT, use the multiple CPU clock synchronization function.
☞ Page 4-33, Section 4.7.1

(d) Connection with GX Works2/GX Developer
For the connection with GX Works2/GX Developer, refer to the following.
☞ Page 2-29, Section 2.4 (7)

(2) Precautions for standard RAM, standard ROM, and CompactFlash card

Table A.1 Memory availability

Memory	Availability		
	Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
Standard RAM	○	×	×
Standard ROM	×	○	○
CompactFlash card	○	○	×

○: Available, ×: N/A

(a) Standard RAM formatting

1) When formatting the standard RAM
Be sure to perform the method described in the following section.
☞ Page 6-77, Section 6.10.1 (6)
Do not format it by using a command from Workbench Shell.

2) During standard RAM formatting
Do not power off or reset the C Controller module during standard RAM formatting.
Doing so causes the standard RAM to be formatted again when the C Controller module is started the next time.

(b) Standard ROM formatting

1) When formatting the standard ROM
Be sure to perform the method described in the following section.
☞ Page 6-85, Section 6.10.2 (4)
Do not format it by using a command from Tornado Shell.

2) During standard ROM formatting
Do not power off or reset the C Controller module during standard ROM formatting.
Doing so causes the standard ROM to be formatted again when the C Controller module is started the next time.

(c) Allowable number of writes to the standard ROM (service life)
A flash ROM is used for the standard ROM and it has a limit in the number of writes (service life). (☞ Page 6-71, Section 6.9)

(d) Allowable number of writes to a CompactFlash card (service life)

- For the live of the QD81MEM and GT05-MEM, refer to the following.
  ☞ Page 6-70, Section 6.8.5
- For the live of commercially available CompactFlash cards, refer to the specifications of each product.

(e) CompactFlash card replacement

Do not power off or reset the system or remove the CompactFlash card while writing file data to the CompactFlash card.
Doing so can cause corruption of the CompactFlash card data or a file system error.

- To remove a CompactFlash card during file writing:
  ☞ Page 6-64, Section 6.8.2
- To power off or reset the system:
  ☞ Page 8-9, Section 8.2.2

(f) File name and directory name

Use alphanumeric characters and special characters (excluding \, /, *, ?, <, >, |, :, and ") for the names of a file and a directory stored in the following memories in the C Controller module:

- Standard RAM
- Standard ROM
- CompactFlash card

If a file or a directory with a name including characters other than alphanumeric characters and special characters (excluding \, /, *, ?, <, >, |, :, and ") is stored in the memories above in the C Controller module, the following phenomena may occur:

- Character corruption on file names and directory names
- Loss of files and directories

(g) Unmounting a CompactFlash card by the RESET/SELECT switch

1) When unmounting during access
Unmounting a CompactFlash card by the RESET/SELECT switch during file writing to the CompactFlash card can cause corruption of the CompactFlash card data or a file system error.
Stop the access to the CompactFlash card before removing the card.
(☞ Page 6-64, Section 6.8.2)

2) When powering off or resetting the system
Before powering off or resetting the system, refer to the following.
☞ Page 8-9, Section 8.2.2

3) Switch operation
If the RESET/SELECT switch is held in the RESET position by mistake, the C Controller module will be reset.
Be careful when unmounting a CompactFlash card by operating the RESET/SELECT switch. (☞ Page 6-67, Section 6.8.3)

(3) Precautions for battery

(a) For the Q12DCCPU-V

1) About file corruption

The battery must be installed before operation.

- If the C Controller system is operated without a battery and then is powered off or reset
  The data in the standard RAM and battery-backed-up RAM may be damaged, or a file system error may occur.
- When the battery is not replaced even after a battery error
  The standard RAM or battery-backed-up RAM data or clock data may be corrupted, or a file system error may occur.

2) Battery replacement

For battery replacement, observe the procedures described in this manual.
(☞ Page 6-53, Section 6.7.3 (4))

(b) For the Q06CCPU-V(-B)

1) About file corruption

- When the battery is not replaced even after a battery error
  The standard ROM and battery-backed-up RAM data being accessed and clock data may be corrupted, or a file system error may occur.
- If the C Controller system is operated without a battery and then shut down improperly
  The standard ROM and battery-backed-up RAM data being accessed may be corrupted, or a file system error may occur.
- When the procedures shown in Page 6-54, Section 6.7.4 are not observed
  The standard ROM and battery-backed-up RAM data being accessed may be corrupted, or a file system error may occur.

2) Restrictions on operations without a battery

- Powering on the system without a battery
  If the C Controller system is powered on with no battery installed in the C Controller module, the system will start with unreliable clock data.
  Set clock data in this case. Otherwise, event history files and programs using clock data cannot function properly. (☞ Page 6-54, Section 6.7.4)
- Operating the system without a battery
  If the system was operated with no battery installed in the C Controller module, perform the shutdown operation before powering off or resetting the C Controller system. (☞ Page 6-55, Section 6.7.4 (2))

3) Battery replacement

For battery replacement, observe the procedures described in this manual.
(☞ Page 6-53, Section 6.7.3 (4))

(4) Precautions for clock setting

(a) Clock setting

1) Setting method
Set the clock of the C Controller module as described in Page 4-30, Section 4.7 (4) (a).
If the year 2100 is exceeded after clock setting, clock data of the year 2100 and later can be used until the C Controller module is restarted.
At the time of restart, year data of the C Controller module will be reset to those of 2000 to 2099.

2) Condition for the setting
The clock of the C Controller module must be set with the QBF_WaitEvent and QBF_WaitUnitEvent functions not being executed.

(5) Precautions for starting the C Controller module

(a) Time required for connection from each utility to the C Controller module
After power-on or reset, the C Controller module is ready to connect to each utility at the following timing.

- When executing the script file, "STARTUP.CMD":
Upon completion of the RUN LED flashing
- When not executing the script file, "STARTUP.CMD":
Fifteen (15) seconds after completion of the start or reset process

Do not attempt a connection before the above timing after power-on or reset.
Changing "Priority" to a larger value in the Option tab of C Controller setting utility increases the time after which the C Controller module can be connected.

(b) Flashing "01" or "02" of the 7-segment LED (Q12DCCPU-V only)
At start-up of the C Controller module, "01" or "02" flashes on the 7-segment LED in the following cases.

- When the C Controller module is in factory-default state
(Because the standard RAM is not initialized)
- When the C Controller module is left unused for a long period of time with no battery installed
(Because the standard and battery-backed-up RAM data are corrupted)

If "01" or "02" flashes on the 7-segment LED, install a battery and perform the module initialization setting procedure. (☞ Page 6-77, Section 6.10.1 (6))

(6) Precautions for network configurations

(a) Precautions for using two Ethernet channels (Q12DCCPU-V only)
When using two channels as Ethernet ports, two different network addresses must be set for CH1 and CH2 of the C Controller module.
In the above case, a response to the message received through each channel is sent as follows:

- Messages (including response packets such as ping) to the device having the same network address as CH1 are sent from CH1.
- Messages (including response packets such as ping) to the device having the same network address as CH2 are sent from CH2.

(7) Relation between system tasks and the system watchdog timer, user watchdog timer, and link device refresh cycles

When using any of the following functions, set a sufficiently long time for each of the system watchdog timer, user watchdog timer, and link device refresh cycles.

- Shell command
- Workbench/Tornado connection
- File access
- Mount/unmount of CompactFlash card
- Ethernet communications
- NFS server communication

If any of the above is used, CPU utilization for a system task with high priority may increase and a system watchdog timer error, a user watchdog timer error, and link refresh timeout may occur more frequently.
For the link refresh timeout, the rate of occurrence may also increase when bus interface driver processing (connections with peripheral devices or communication with an intelligent function module, etc.) is used.

(8) Precautions for the Wind River Systems product

The C Controller module has an embedded real-time operating system, VxWorks, made and sold by Wind River Systems, Inc. in the United States.
We, Mitsubishi, make no warranty for the Wind River Systems product and will not be liable for any problems and damages caused by the Wind River Systems product during use of the C Controller module.
For the problems or specifications of the Wind River Systems product, refer to the corresponding manual or consult Wind River Systems, Inc.
Contact information is available on the following website.
www.windriver.com

INTRODUCTION

Thank you for purchasing the Mitsubishi C Controller module.
Before using the product, please read this manual carefully to understand the features and performance of the C Controller module and use it correctly.
Note that the menu names and operating procedures may differ depending on an operating system in use and its version. When reading this manual, replace the names and procedures with the applicable ones as necessary.

CONTENTS

CHAPTER 15 STARTING A MULTIPLE CPU SYSTEM 15 - 1 to 15 - 16

CHAPTER 16 TROUBLESHOOTING 16 - 1 to 16 - 93

APPENDICES APPX - 1 to APPX - 48

INDEX INDEX - 1 to INDEX - 2

ABOUT MANUALS

The following manuals are relevant to this product.
Order each manual as needed, referring to the table below.

Relevant Manuals

Manual name	Manual number (code)
C Controller Module User's Manual (Utility Operation, Programming) Describes the installation and uninstallation of VxWorks-based setting and monitoring tool for C Controller module (SW3PVC-CCPU), utility operations, and functions and programming. (Sold separately)	SH-080767ENG (13JZ18)
MELSEC-Q C Controller Module User's Manual Describes the system configuration, specifications, functions, handling methods, wiring, troubleshooting, and programming and function of C Controller module (Q24DHCCPU-V, Q24DHCCPU-VG, Q24DHCCPU-LS, Q26DHCCPU-LS, and Q12DCCPU-V (Extended mode)). (Sold separately)	SH-081130ENG (13JZ75)
Setting/Monitoring Tools for the C Controller Module Version 4 Operating Manual Describes the system configuration and operation method of Setting/Monitoring Tools for the C Controller Module (SW4PVC-CCPU). (Sold separately)	SH-081131ENG (13JU76)
CW Workbench Operating Manual Describes the system configuration, installation/uninstallation, specifications, functions, and troubleshooting of the product. (Sold separately)	SH-080982ENG (13JU71)
CW-Sim Operating Manual Describes the system configuration, specifications, functions, and troubleshooting of CW-Sim. (Sold separately)	SH-081159ENG (13JU77)

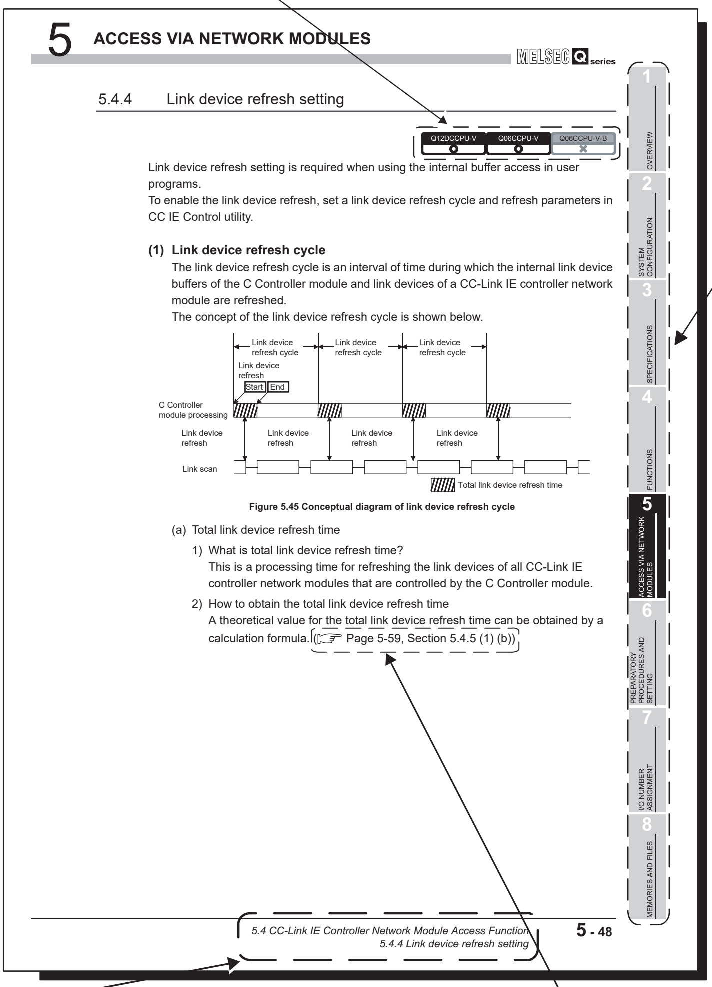
MANUAL PAGE ORGANIZATION

Relevant model

Whether the description of the section applies to each model or not is shown as in the table below.

Chapter

The chapter of the current page can be easily identified by this indication on the right side.



5 ACCESS VIA NETWORK MODULES

MELSEC Q series

5.4.4 Link device refresh setting

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	×

Link device refresh setting is required when using the internal buffer access in user programs.
To enable the link device refresh, set a link device refresh cycle and refresh parameters in CC IE Control utility.

(1) Link device refresh cycle

The link device refresh cycle is an interval of time during which the internal link device buffers of the C Controller module and link devices of a CC-Link IE controller network module are refreshed.
The concept of the link device refresh cycle is shown below.

Link device refresh cycle
Link device refresh cycle
Link device refresh cycle
Link device refresh
Start
End
C Controller module processing
Link device refresh
Link device refresh
Link device refresh
Link device refresh
Link scan
Total link device refresh time

Figure 5.45 Conceptual diagram of link device refresh cycle

(a) Total link device refresh time

1) What is total link device refresh time?
This is a processing time for refreshing the link devices of all CC-Link IE controller network modules that are controlled by the C Controller module.

2) How to obtain the total link device refresh time
A theoretical value for the total link device refresh time can be obtained by a calculation formula.(☞ Page 5-59, Section 5.4.5 (1) (b))

5.4 CC-Link IE Controller Network Module Access Function
5.4.4 Link device refresh setting

5 - 48

Section and title

The section number and title of the current page can be easily identified.

Reference

The section in this manual or another relevant manual that can be referred to is shown after ☞.

* The above page illustration is for explanation purpose only, and is different from the actual page.

Icon			Description
Q12DCCPU-V	**Q06CCPU-V**	**Q06CCPU-V-B**	
Q12DCCPU-V ○	Q06CCPU-V ○	Q06CCPU-V-B ○	All or part of the description applies to each model.
Q12DCCPU-V △	Q06CCPU-V △	Q06CCPU-V-B △	The description applies to each model with some restrictions.
Q12DCCPU-V ×	Q06CCPU-V ×	Q06CCPU-V-B ×	The description does not apply to each model.

This manual is intended to help you understand the system configurations, specifications, functions, preparatory procedures, and troubleshooting of the C Controller module.

Utilize this manual, referring to the following.

Item	Description
CHAPTER 1	Describes the features of the C Controller module.
CHAPTER 2	Describes system configurations that include the C Controller module, and provides a list of the modules that are accessible from the C Controller module.
CHAPTER 3	Describes the specifications and performance of the C Controller module.
CHAPTER 4	Describes the functions of the C Controller module.
CHAPTER 5	Describes the network module access function of the C Controller module.
CHAPTER 6	Explains the procedures for starting the system operation using the C Controller module.
CHAPTER 7	Describes the stage number setting for extension base units and I/O number assignment.
CHAPTER 8	Describes the memories of the C Controller module.
CHAPTER 9	Describes devices that can be used with C Controller modules.
CHAPTER 10 to CHAPTER 15	Describes multiple CPU system configurations, I/O numbers, communications among programmable controller CPUs, I/O modules, and intelligent function modules, and how to start a multiple CPU system.
CHAPTER 16	Provides corrective actions to be taken for respective errors, and the error codes that may be generated during function execution.

Remark

This manual does not explain the SW3PVC-CCPU installation and uninstallation procedures, utility operations, functions, and programming.
For details, refer to the following.
☞ C Controller Module User's Manual (Utility Operation, Programming)

GENERIC TERMS AND ABBREVIATIONS

Unless otherwise specified, this manual uses the following generic terms and abbreviations to explain the C Controller module.

(1) C Controller modules and SW3PVC-CCPU

Generic term/abbreviation	Description
Q12DCCPU-V	Abbreviation for the Q12DCCPU-V C Controller module In principle, 'Q12DCCPU-V' indicates Q12DCCPU-V (Basic mode). When the classification is needed for such as comparison with other modules, 'Q12DCCPU-V (Basic mode)' and ‘Q12DCCPU-V (Extended mode)’ are mentioned.
Q12DCCPU-V (Basic mode)	Status that Q12DCCPU-V is initialized with the basic mode
Q12DCCPU-V (Extended mode)	Status that Q12DCCPU-V is initialized with the extended mode For Q12DCCPU-V (Extended mode), refer to the following manual. ☞ MELSEC-Q C Controller Module User's Manual
Q06CCPU-V	Abbreviation for the Q06CCPU-V C Controller module
Q06CCPU-V-B	Abbreviation for the Q06CCPU-V-B C Controller module
Q06CCPU-V(-B)	Generic term for the Q06CCPU-V and Q06CCPU-V-B
Q24DHCCPU-V	Abbreviation for the Q24DHCCPU-V C Controller module For Q24DHCCPU-V, refer to the following manual. ☞ MELSEC-Q C Controller Module User's Manual
Q24DHCCPU-VG	Abbreviation for the Q24DHCCPU-VG C Controller module For Q24DHCCPU-VG, refer to the following manual. ☞ MELSEC-Q C Controller Module User's Manual
Q24DHCCPU-LS	Abbreviation for the Q24DHCCPU-LS C Controller module For Q24DHCCPU-LS, refer to the following manual. ☞ MELSEC-Q C Controller Module User's Manual
Q26DHCCPU-LS	Abbreviation for the Q26DHCCPU-LS C Controller module For Q26DHCCPU-LS, refer to the following manual. ☞ MELSEC-Q C Controller Module User's Manual
C Controller module	Generic term for the Q06CCPU-V, Q06CCPU-V-B, Q12DCCPU-V, Q24DHCCPU-V, Q24DHCCPU-VG, Q24DHCCPU-LS, and Q26DHCCPU-LS
SW3PVC-CCPU	Abbreviation for the Setting/Monitoring Tools for the C Controller Module (SW3PVC-CCPU-E)
SW4PVC-CCPU	Abbreviation for the Setting/Monitoring Tools for the C Controller Module (SW4PVC-CCPU-E) For SW4PVC-CCPU, refer to the following manual. ☞ Setting/Monitoring Tools for the C Controller Module Version 4 Operating Manual

(2) CPU modules

Generic term/abbreviation	Description
QnACPU	Generic term for the Q2ACPU, Q2ACPU-S1, Q2ASCPU, Q2ASCPU-S1, Q2ASHCPU, Q2ASHCPU-S1, Q3ACPU, Q4ACPU, and Q4ARCPU
Basic model QCPU	Generic term for the Q00CPU and Q01CPU
High Performance model QCPU	Generic term for the Q02CPU, Q02HCPU, Q06HCPU, Q12HCPU, and Q25HCPU
Process CPU	Generic term for the Q02PHCPU, Q06PHCPU, Q12PHCPU, and Q25PHCPU
Redundant CPU	Generic term for the Q12PRHCPU and Q25PRHCPU
Universal model QCPU	Generic term for the Q00UJCPU, Q00UCPU, Q01UCPU, Q02UCPU, Q03UDCPU, Q04UDHCPU, Q06UDHCPU, Q10UDHCPU, Q13UDHCPU, Q20UDHCPU, Q26UDHCPU, Q03UDECPU, Q04UDEHCPU, Q06UDEHCPU, Q10UDEHCPU, Q13UDEHCPU, Q20UDEHCPU, Q26UDEHCPU, Q50UDEHCPU, Q100UDEHCPU, Q03UDVCPU, Q04UDVCPU, Q06UDVCPU, Q13UDVCPU, and Q26UDVCPU
QCPU (Q mode)	Generic term for the Basic model QCPUs, High Performance model QCPUs, Process CPUs, Redundant CPUs, and Universal model QCPUs
Motion CPU	Generic term for the Q172CPUN, Q172CPUN-T, Q172HCPU, Q172HCPU-T, Q173CPUN, Q173CPUN-T, Q173HCPU, Q173HCPU-T, Q172DCPU, and Q173DCPU
CPU module	Generic term for C Controller modules, QCPUs (Q mode), and Motion CPUs
Single CPU system	Control system where a C Controller module is mounted in the CPU slot
Multiple CPU system	Control system where multiple CPU modules are mounted on a main base unit
Control CPU	CPU module that can control I/O modules and intelligent function modules on main and extension base units. For example, when CPU No.2 controls a module mounted in Slot 3, CPU No.2 is a control CPU of the module in Slot 3.
Controlled module	I/O module or intelligent function module that is controlled by a control CPU. For example, when CPU No.2 controls a module mounted in Slot 3, the module in Slot 3 is a controlled module of CPU No.2.
Non-controlled module (Module outside an I/O sharing group)	I/O module or intelligent function module other than controlled modules of a CPU. For example, when CPU No.2 controls a module mounted in Slot 3, the module in Slot 3 is a non-controlled module of CPUs No.1 and No.3.
Non-control CPU	CPU module that is not a control CPU. For example, when CPU No.2 controls a module mounted in Slot 3, CPUs No.1 and No.3 are non-control CPUs of the module in Slot 3.
Battery	Generic term for the Q6BAT and Q7BAT batteries for CPU modules
PC CPU module	Abbreviation for the MELSEC-Q series PC CPU module manufactured by CONTEC CO., Ltd.

(3) Network modules and PC boards

Generic term/abbreviation	Description
CC-Link module	Generic term for the QJ61BT11 and QJ61BT11N
CC-Link/LT module	Generic term for the QJ61CL12
CC-Link board	Abbreviation for the Q81BD-J61BT11 or Q80BD-J61BT11N CC-Link system master/local interface board, the A80BD-J61BT11 CC-Link system master/local interface board, and the A80BD-J61BT13 CC-Link interface board.
CC-Link IE controller network module	Generic term for the QJ71GP21-SX and QJ71GP21S-SX
CC-Link IE controller network interface board	Abbreviation for the Q80BD-J71GP21-SX or Q80BD-J71GP21S-SX CC-Link IE controller network interface board
CC-Link IE field network master/local module	Abbreviation for the QJ71GF11-T2 CC-Link IE field network master/local module
MELSECNET/H	Generic term for the Q series MELSECNET/H network system
MELSECNET/H module	Generic term for the QJ71LP21-25, QJ71LP21S-25, QJ71LP21G, QJ71LP21GE, QJ72LP25-25, QJ72LP25G, QJ72LP25GE, QJ71BR11, QJ72BR15, and QJ71NT11B
MELSECNET/H board	Abbreviation for the Q81BD-J71LP21-25, Q80BD-J71LP21-25, Q80BD-J71LP21G, Q80BD-J71LP21S-25, or Q80BD-J71BR11 MELSECNET/H interface board

(4) Power supply modules and base units

Generic term/abbreviation	Description
Q3□B	Generic term for the Q33B, Q35B, Q38B, and Q312B main base units, to each of which a CPU module, Q series power supply module, Q series I/O modules, intelligent function modules can be mounted.
Q3□SB	Generic term for the Q32SB, Q33SB, and Q35SB slim type main base units, to each of which a C Controller modules, Basic model QCPU, High Performance model QCPU, Universal model QCPU, slim type power supply module, Q series I/O modules, and intelligent function modules can be mounted
Q3□RB	Generic term for the Q38RB main base unit for redundant power supply system, to each of which a CPU module, a redundant power supply module, Q series I/O modules, intelligent function modules can be mounted
Q3□DB	Generic term for the Q35DB, Q38DB, and Q312DB multiple CPU high speed main base units, to each of which a CPU module, a Q series power supply module, Q series I/O modules, intelligent function modules can be mounted
Q5□B	Generic term for the Q52B and Q55B extension base units, to which Q series I/O modules and intelligent function modules can be mounted
Q6□B	Generic term for the Q63B, Q65B, Q68B, and Q612B extension base units, to which Q series power supply modules, Q series I/O modules, and intelligent function modules can be mounted
Q6□RB	Generic term for the Q68RB extension base unit for redundant power supply system, to which a redundant power supply module, Q series I/O modules, and intelligent function modules can be mounted
QA1S6□B	Generic term for the QA1S65B and QA1S68B extension base units, to which AnS series power supply modules, AnS series I/O modules, and special function modules can be mounted
Main base unit	Generic term for the Q3□B, Q3□SB, Q3□RB, and Q3□DB
Extension base unit	Generic term for the Q5□B, Q6□B, Q6□RB, and QA1S6□B
Slim type main base unit	Generic term for the Q3□SB
Redundant power main base unit	Generic term for the Q3□RB
Redundant power extension base unit	Generic term for the Q6□RB
Multiple CPU high speed main base unit	Generic term for the Q3□DB
Base unit	Generic term for the main base unit, extension base unit, slim type main base unit, redundant power main base unit, redundant power extension base unit, and multiple CPU high speed main base unit
Redundant power supply base unit	Generic term for the redundant power main base unit and redundant power extension base unit
Q series power supply module	Generic term for the Q61P-A1, Q61P-A2, Q61P, Q61P-D, Q62P, Q63P, Q64P, and Q64PN power supply modules
Slim type power supply module	Generic term for the Q61SP slim type power supply module
Redundant power supply module	Generic term for the Q63RP, Q64RP, and Q64RPN redundant power supply modules
Power supply module	Generic term for the Q series power supply modules, slim type power supply modules, and redundant power supply modules
Extension cable	Generic term for the QC05B, QC06B, QC12B, QC30B, QC50B, and QC100B extension cables
CPU slot	Slot on the immediate right side of the power supply module on a main base unit

(5) Others

Generic term/abbreviation	Description
Q series	Abbreviation for the programmable controllers, MELSEC-Q series
AnS series	Abbreviation for the programmable controllers, MELSEC-AnS series, which are the downsized series of the MELSEC-A series
Ethernet	Generic term for the 100BASE-TX and 10BASE-T network systems
QD81MEM	Generic term for the QD81MEM-512MBC, QD81MEM-1GBC, QD81MEM-2GBC, QD81MEM-4GBC and QD81MEM-8GBC
GT05-MEM	Generic term for the GT05-MEM-128MC and GT05-MEM-256MC
GOT	Abbreviation for the Mitsubishi Graphic Operation Terminal
GX Works2	Product name for the MELSEC programmable controller software package
GX Developer	
CW Workbench	The abbreviation for the engineering tool for C Controller, CW Workbench
Wind River Workbench	Abbreviation for Workbench 2.6.1 manufactured by Wind River Systems, Inc. For specifications and inquiries of Wind River Workbench, visit the website of Wind River Systems, Inc.: www.windriver.com
Workbench	Generic term for the CW Workbench and Wind River Workbench
Tornado	Abbreviation for Tornado 2.1.0 for Hitachi SuperH Cumulative patch 1 manufactured by Wind River Systems, Inc. For specifications and inquiries of Tornado, visit the website of Wind River Systems, Inc.: www.windriver.com
VxWorks	Product name for the real-time operating system manufactured by Wind River Systems, Inc.

GLOSSARY

The following lists the terms used in C Controller module manuals and the meanings of the terms.

Term	Description
FTP	FTP is an abbreviation for File Transfer Protocol, which is used to transfer data files.
Telnet	Network protocol, or virtual terminal software, that enables remote login in TCP/IP networks.
CompactFlash card (CF card)	Storage card stipulated in the "CF+ and CompactFlash Specification" issued by CompactFlash Association.
Bus interface function	Function offered by SW3PVC-CCPU. This allows input from or output to I/O modules controlled by the C Controller module, access to the buffer memory of an intelligent function module, and the read out or control of the C Controller module status.
MELSEC data link function	Function offered by SW3PVC-CCPU. Standardized communication library that does not depend on communication protocols. When creating a program for communicating with a programmable controller CPU or C Controller module, there is no need to consider the hardware or communication protocol of the target. The MELSEC data link functions support the following. • Q series bus interface communication • CC-Link communication • MELSECNET/H communication • CC-Link IE controller network communication The MELSEC data link functions cannot be used for the Q06CCPU-V-B.

PRODUCT ORGANIZATION

The following shows the C Controller-compatible software.

Supported software	C Controller module			
	Q12DCCPU-V			Q06CCPU-V Q06CCPU-V-B
	"15102" or later[*1]		Earlier than "15102"[*1]	-
	Extended mode	Basic mode[*2]		-
SW4PVC-CCPU	○	×	×	×
SW3PVC-CCPU	×	○	○	○

○: Applicable, ×: Not applicable

[*1] First five digits of serial number

[*2] For Q12DCCPU-V earlier than "15102", the mode cannot be changed. Q12DCCPU-V earlier than "15102" is regarded as the basic mode in this manual.

PACKING LIST

One of the listed C Controller modules is included in the product package.

Model	Product name	Quantity
Q12DCCPU-V	Q12DCCPU-V C Controller module (Endian format (memory layout): Little endian)	1
	Battery (Q6BAT)	1
Q06CCPU-V	Q06CCPU-V C Controller module (Endian format (memory layout): Little endian)	1
	Battery (Q6BAT)	1
Q06CCPU-V-B	Q06CCPU-V-B C Controller module (Endian format (memory layout): Big endian)	1
	Battery (Q6BAT)	1
SW3PVC-CCPU-E	Setting/Monitoring Tools for the C Controller Module (Volume license product) (CD-ROM)	1
	Software License Agreement	1
	Software Registration Form	1
	License Agreement	1
	Industrial development tool purchasing form (Wind River Workbench)	1
	Industrial development tool purchasing form (Tornado)	

DISCONTINUED MODELS

The following models are described in this manual, but have no longer been produced. For the onerous repair term after discontinuation of production, refer to "WARRANTY".

Model	Production discontinuation
Q61P-A1	March 2009
Q61P-A2	March 2009
Q64P	February 2010
Q64RP	October 2015
Q06CCPU-V-B	November 2015
Q06CCPU-V	March 2022

CHAPTER 1 OVERVIEW

This manual describes the specifications, functions, preparatory procedures, and troubleshooting of the C Controller module.

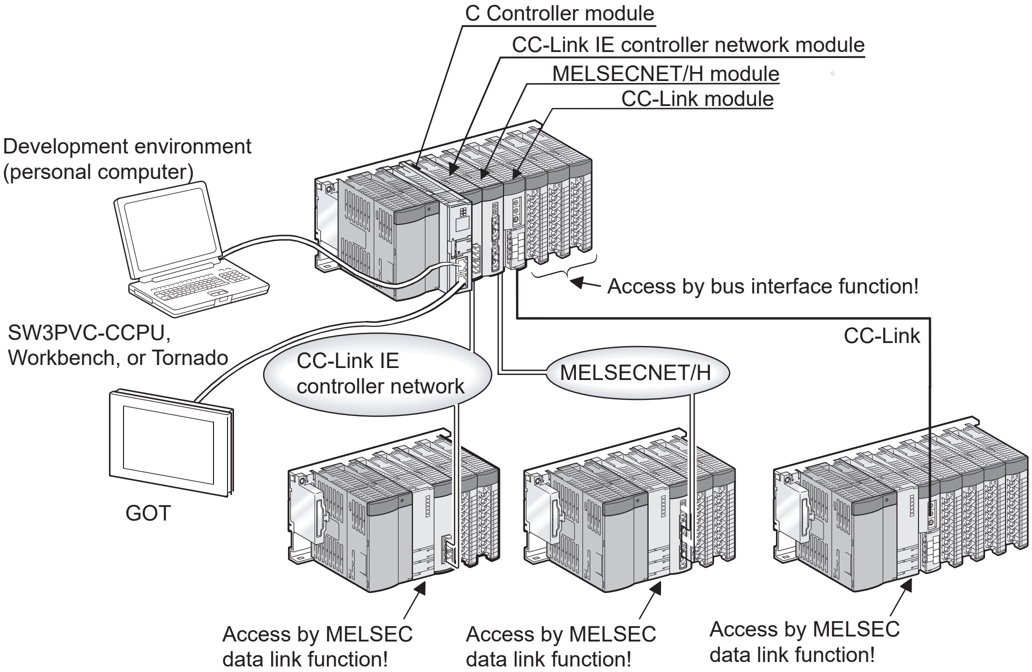


Figure 1.1 Example of a system that includes a C Controller module

Remark

For the utilities and functions offered by the Setting/Monitoring Tools for the C Controller Module (SW3PVC-CCPU), refer to the following.

☞ C Controller Module User's Manual (Utility Operation, Programming)

(1) List of related manuals

The following table lists the manuals that are related to the C Controller module.

For further information such as manual numbers, refer to "ABOUT MANUALS". (☞ Page A-26)

Table 1.1 List of related manuals

	Hardware (Packed)	Hardware Design	Utility Operation	Maintenance and Inspection	Multiple CPU System
Purpose	**C Controller Module User's Manual (Hardware)**	**C Controller Module User's Manual (Hardware Design, Function Explanation)**	**C Controller Module User's Manual (Utility Operation, Programming)**	**QCPU User's Manual (Hardware Design, Maintenance and Inspection)**	**QCPU User's Manual (Multiple CPU System)**
Checking part names and specifications of the C Controller module	Outline	Details *1			
Checking specifications of power supply modules and base units, and selection and installation methods				Details	
Checking how to connect power supply modules, base units, and I/O modules	Outline	Details *1			
Checking the functions of the C Controller module		Details *1			
Single CPU system construction (checking the start-up procedure and I/O No. assignment method)		Details *1			
Multiple CPU system construction (checking the start-up procedure and I/O No. assignment method)		Outline *1			Details
Checking parameter setting and monitoring			Details *2		
Checking programming			Details *1		
Checking troubleshooting and error codes		Details *1			
Checking event No. and error messages in respective utilities			Details *1		

* 1 For Q24DHCCPU-V, Q24DHCCPU-VG, Q24DHCCPU-LS, Q26DHCCPU-LS, or Q12DCCPU-V (Extended mode), refer to the following manual.

☞ MELSEC-Q C Controller Module User's Manual

* 2 For Q24DHCCPU-V, Q24DHCCPU-VG, Q24DHCCPU-LS, Q26DHCCPU-LS, or Q12DCCPU-V (Extended mode), refer to the following manual.

☞ Setting/Monitoring Tools for the C Controller Module Version 4 Operating Manual

Remark

This manual mainly describes the C Controller module itself.
For the C Controller system information listed below, refer to the following.

☞ QCPU User's Manual (Hardware Design, Maintenance and Inspection)

When reading the above manual, interpret the terms, "CPU module" and "programmable controller" as "C Controller module" and "C Controller system" accordingly.

Table 1.2 List of referenced sections

Item	Refer to
Checking the power supply module specifications to make a selection	Chapter 5
Checking the specifications of base units and extension cables to make a selection, and configuring the system	Chapter 6
Compliance of the C Controller system with the EMC and Low Voltage Directives	Chapter 9
Calculating the heat value of the C Controller system	Section 10.2
Mounting modules	Section 10.3
Checking how to set the number of extension base units	Section 10.4
Connecting/disconnecting extension cables	Section 10.5
Wiring the power supply module	Section 10.6
Checking I/O module troubleshooting cases	Section 12.5
Checking external dimensions of power supply modules and base units	Appendix 1

1.1 Features

The features of the C Controller module are described below.

(1) Development environment can reduce the development period and engineering cost.

By adopting a real-time operating system "VxWorks", a highly reliable and robust system can be constructed.

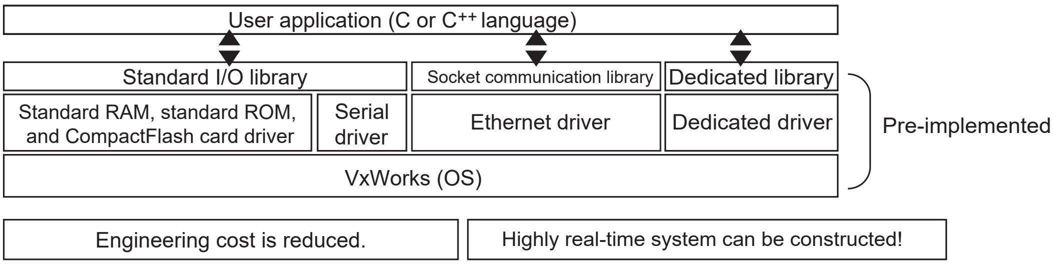

User application (C or C++ language)
Standard I/O library
Socket communication library
Dedicated library
Standard RAM, standard ROM, and CompactFlash card driver
Serial driver
Ethernet driver
Dedicated driver
VxWorks (OS)
Pre-implemented
Engineering cost is reduced.
Highly real-time system can be constructed!

Figure 1.2 Development environment

(2) With a wide range of MELSEC-Q series I/O modules and intelligent function modules, more scalable and flexible systems can be constructed.

Without using sequence programs, the C Controller module can control I/O modules and intelligent function modules from the user programs created with the bus interface functions.

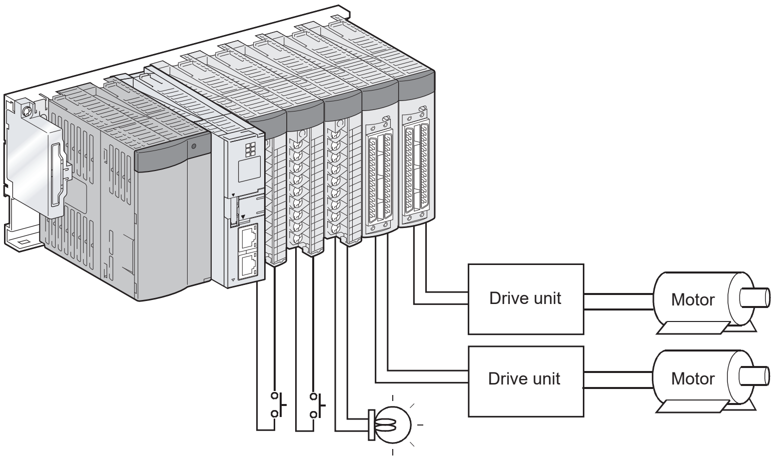


Figure 1.3 Control of I/O modules and intelligent function modules

(3) The system suitable for the scale and control purpose is constructible.

The C Controller module can be used in a multiple CPU system in combination with programmable controller CPU(s) and Motion CPU(s).

In a multiple CPU system, each CPU module can perform the most suitable kind of tasks to obtain the optimum results in the system operation.

Note that the C Controller module set as CPU No.1 can compose a multiple CPU system in combination with Motion CPU(s) (without a programmable controller CPU).

The C Controller module can be also used in a single CPU system configuration.

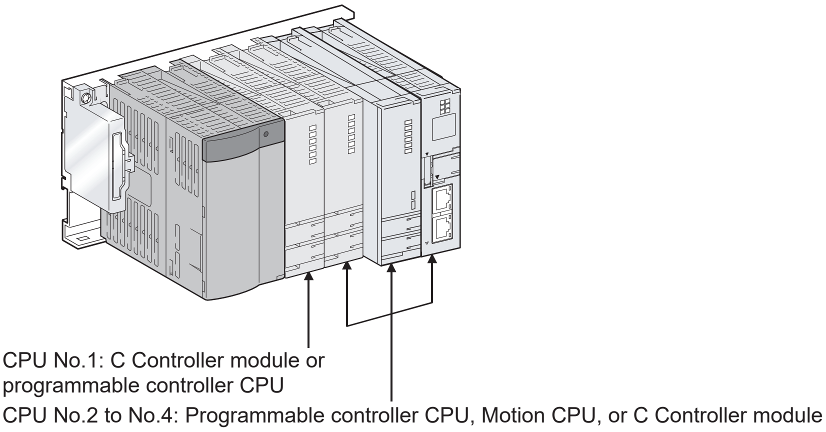

CPU No.1: C Controller module or programmable controller CPU
CPU No.2 to No.4: Programmable controller CPU, Motion CPU, or C Controller module

Figure 1.4 Multiple CPU system

(4) Takt time for the system can be reduced. (Q12DCCPU-V only)

Without using any program, the Q12DCCPU-V can transfer a large volume of data (up to 14K words) at a high speed in cycles of 0.88ms, and can share the data with other CPU modules.

This can reduce the takt time (the maximum time allowed to produce a product in order to meet demand) without considering the multiple CPU system configuration.

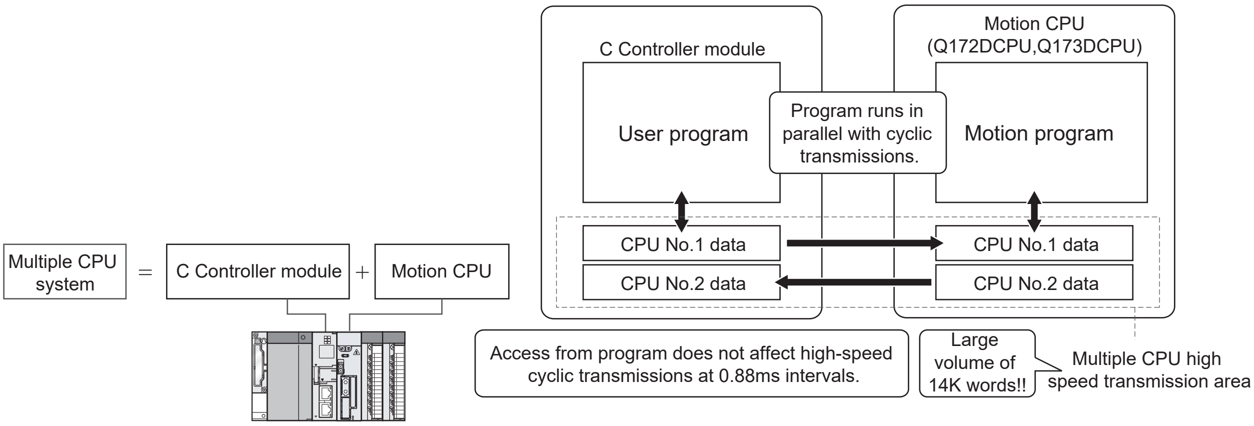

C Controller module
User program
Program runs in parallel with cyclic transmissions.
Motion CPU (Q172DCPU,Q173DCPU)
Motion program
CPU No.1 data
CPU No.2 data
CPU No.1 data
CPU No.2 data
Multiple CPU system = C Controller module + Motion CPU
Access from program does not affect high-speed cyclic transmissions at 0.88ms intervals.
Large volume of 14K words!!
Multiple CPU high speed transmission area

Figure 1.5 Reduced takt time for the system

(5) Sophisticated and high-precision sequential and tracking controls (Q12DCCPU-V only)

The C Controller module can execute an interrupt routine (interrupt program) when one of the following interrupts occurs.

Interrupt routine (interrupt program) execution does not produce a delay that could be caused by user program execution or priority in multitasking.

For this reason, high-performance and high-accuracy sequential controls and tracking controls that respond in real-time can be realized for occurred interrupts.

(a) Multiple CPU synchronous interrupt

The multiple CPU synchronous interrupt is an interrupt that is issued in the same cycles (0.88ms) as Motion CPU (Q172DCPU or Q173DCPU) control cycles.

Data acquisition is available in each operation cycle of a Motion CPU by running an interrupt routine (interrupt program) for the multiple CPU synchronous interrupt.

The multiple CPU synchronous interrupt can also be used for real-time sequential controls synchronized with motion control and also for tracking controls in which the target value is changing every second.

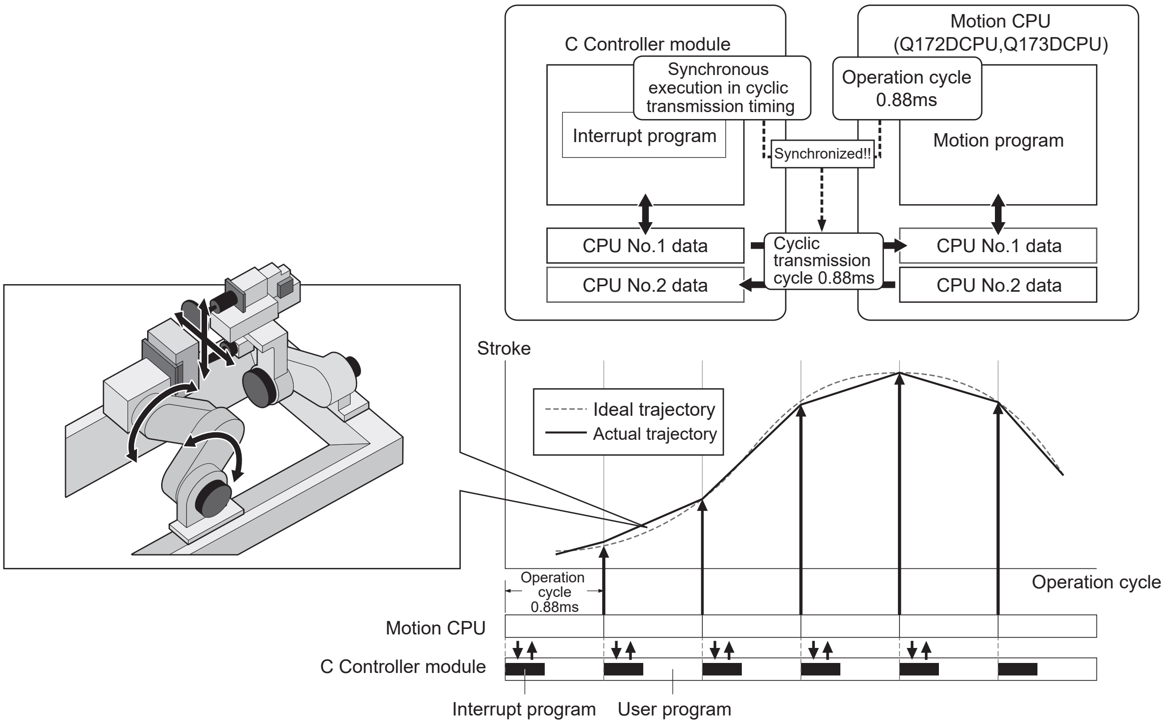


Figure 1.6 Multiple CPU synchronous interrupt

(b) Interrupt from an intelligent function module or an interrupt module

Interrupt routines (interrupt programs) can be executed, corresponding to the interrupts from the intelligent function module or the interrupt module.

An interrupt routine (interrupt program) allows a real-time control responding to an event occurred in the intelligent function module or an input from an external device to the interrupt module.

Interrupt routines (interrupt programs) cannot be executed for the following C Controller modules.

- Q12DCCPU-V with a serial number whose first five digits are "12041" or earlier

(c) Interrupts from other CPUs

Interrupt routines (interrupt programs) can be executed for interrupts from other CPUs (programmable controller CPUs or C Controller modules).

Real-time sequential controls synchronized with controls from another CPU can be used through interrupt routines (interrupt programs).

Interrupt routines (interrupt programs) cannot be executed for the following C Controller modules.

- Q12DCCPU-V with a serial number whose first five digits are "12041" or earlier

(6) Flexible and easy network construction

The C Controller module is equipped with Ethernet port(s).
Ethernet communication programs can be created easily using the VxWorks communication library (socket functions).
With two built-in Ethernet ports (two channels), the Q12DCCPU-V can perform real-time communication that will improve the machine operation rates and productivity.

(a) Case 1: Real-time communication with MES and EES
The C Controller module communicates data such as lot and recipe information with an MES server and time-stamped process data with an EES server, and thereby the system productivity can be improved.

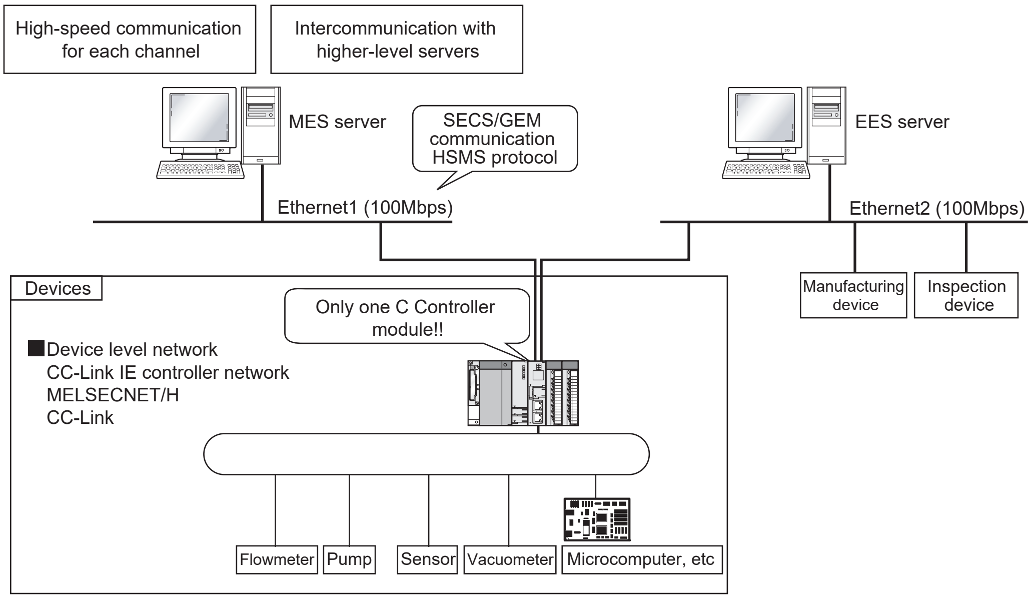


Figure 1.7 Real-time communication with MES and EES

(b) Case 2: Enhanced device level network security

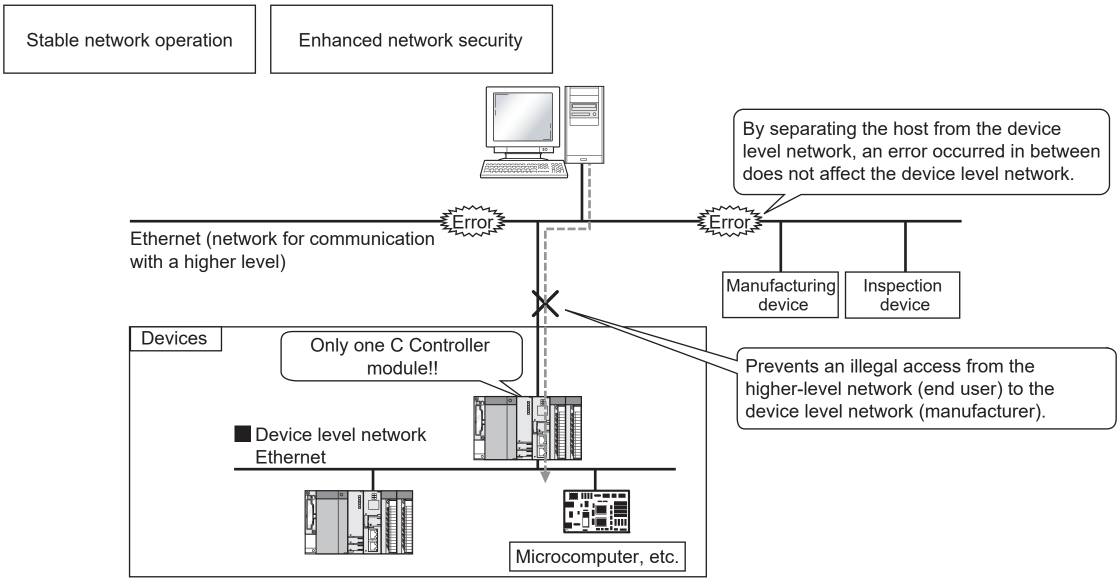

Stable network operation
Enhanced network security
By separating the host from the device level network, an error occurred in between does not affect the device level network.
Error
Error
Ethernet (network for communication with a higher level)
Manufacturing device
Inspection device
Devices
Only one C Controller module!!
Prevents an illegal access from the higher-level network (end user) to the device level network (manufacturer).
■Device level network
Ethernet
Microcomputer, etc.

Figure 1.8 Enhanced device level network security

(7) More efficient debugging and easy device monitoring and troubleshooting using the 7-segment LED display (Q12DCCPU-V only)

Without a computer, users can debug programs, check the operating status, and diagnose some failure easily by viewing the LED display. (Monitoring is available from a personal computer.)

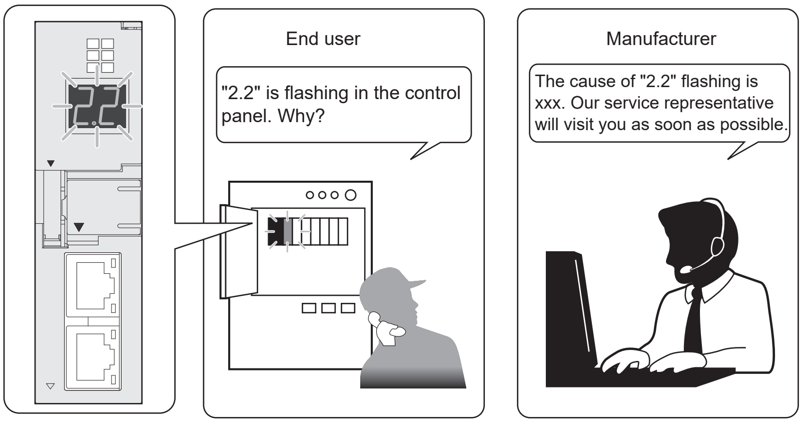

End user
"2.2" is flashing in the control panel. Why?
Manufacturer
The cause of "2.2" flashing is xxx. Our service representative will visit you as soon as possible.

Figure 1.9 7-segment LED display

(8) Easy configuration and startup of positioning modules (Q12DCCPU-V only)

By using GX Configurator-QP, which is a setting, monitoring and testing tool for positioning modules, the engineering cost can be reduced because parameter-setting programs are no more needed.

The programming and debugging efficiency for positioning modules will be remarkably improved.

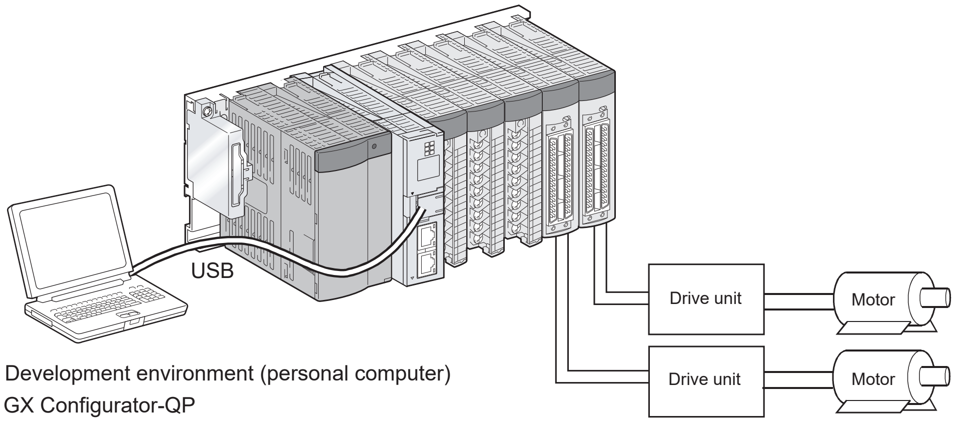


Figure 1.10 Easy configuration and startup of positioning modules

(9) More scalable and flexible system construction utilizing various MELSEC-Q series network products

The following access methods are available through network modules.

(a) Message communication

Using a bus interface function or MELSEC data link function, a message can be transmitted via the following network module. (Message communication with the SEND/RECV function)

- Via a MELSECNET/H module
- Via a CC-Link IE controller network module

Note that, for the Q06CCPU-V-B, MELSEC data link functions cannot be used.

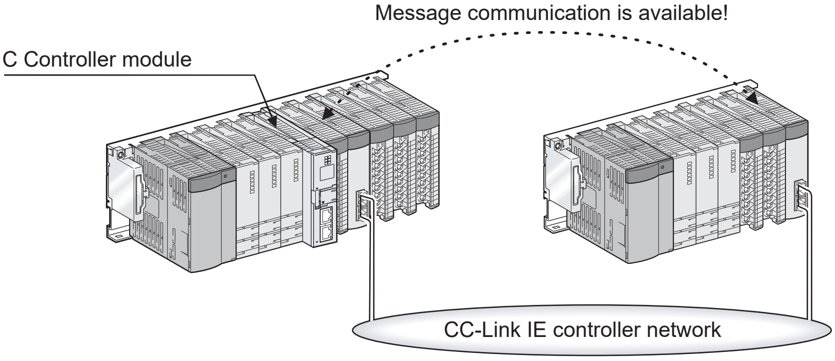

Message communication is available!
C Controller module
CC-Link IE controller network

Figure 1.11 Message communication via a CC-Link IE controller network module

(b) Access to another station's programmable controller CPU or C Controller module
From a user program created with MELSEC data link functions, the C Controller module can access another station's programmable controller CPU or C Controller module via the following network module.
- CC-Link module
- MELSECNET/H module
- CC-Link IE controller network module

The Q06CCPU-V-B cannot access another station's programmable controller CPU.
The following C Controller modules cannot access another station's C Controller module.
- Q12DCCPU-V with a serial number whose first five digits are "12041" or earlier
- Q06CCPU-V(-B)

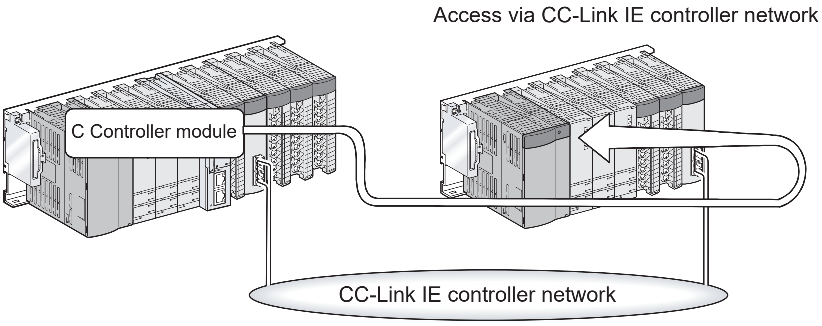

Access via CC-Link IE controller network
C Controller module
CC-Link IE controller network

Figure 1.12 Access to another station's programmable controller via a CC-Link IE controller network module

(10)Control of a wide variety of CC-Link products

The C Controller module can control a wide variety of CC-Link products using user programs.

Up to eight CC-Link modules can be mounted.

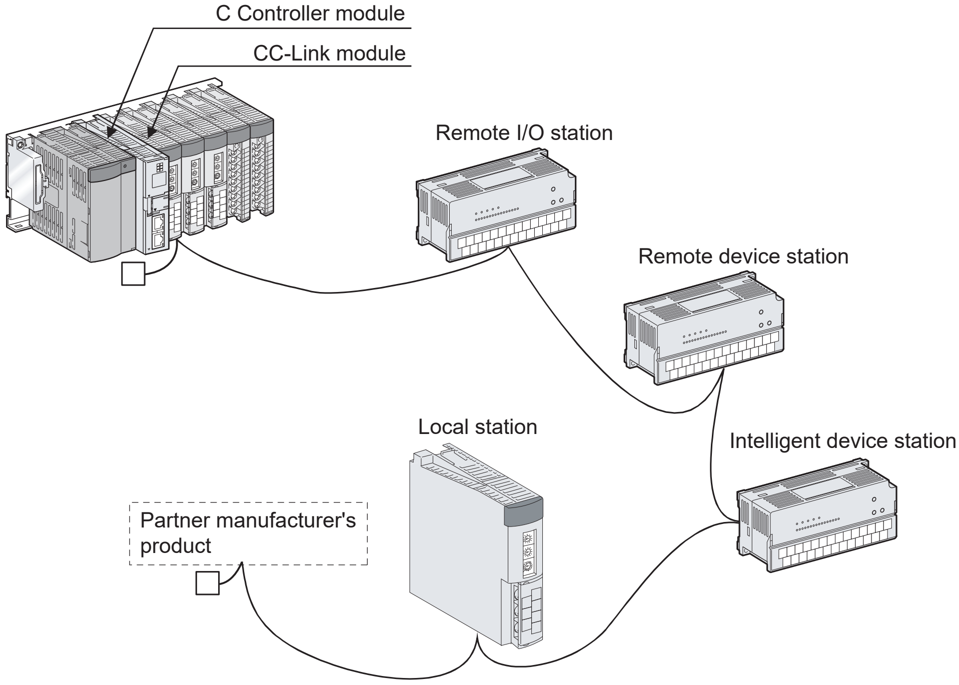


Figure 1.13 Control of CC-Link products

(11)Easy serial communication programming

The C Controller module is equipped with an RS-232 port.

Using the VxWorks communication library, serial communication programs can be created.

The microcomputer connection with a GOT is also available. (☞ Page 4-47, Section 4.14)

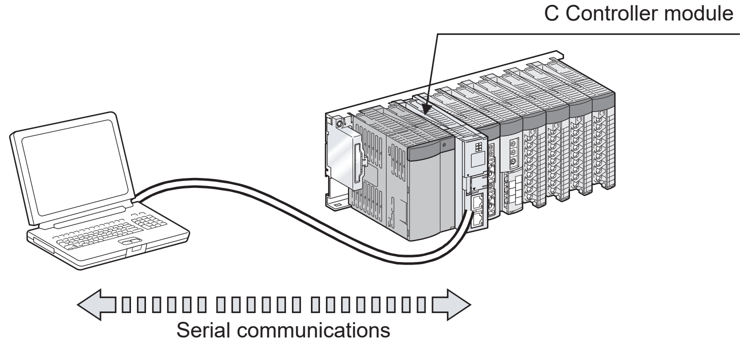
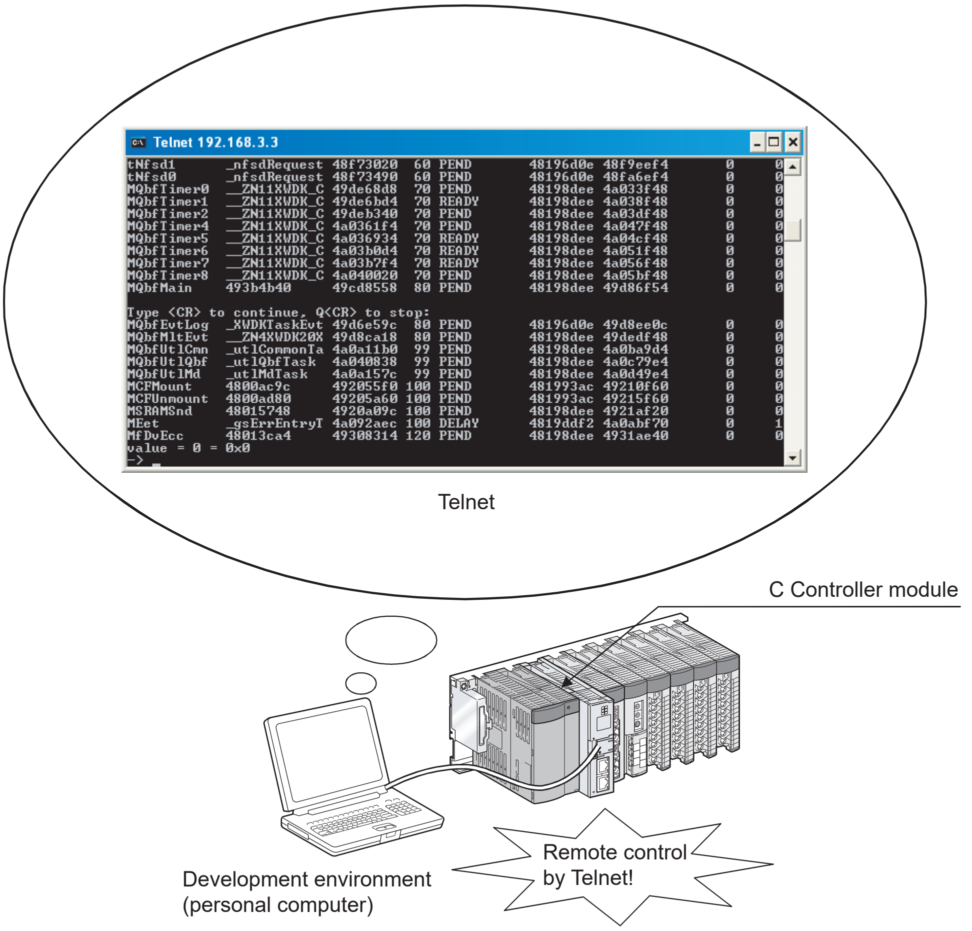


Figure 1.14 Serial communications using RS-232 port

(12)Simple remote debugging of the C Controller module with the supported Telnet function

By executing a Shell command from the Telnet tool of the development environment (personal computer), remote debugging of the C Controller module (task information display, memory dumping, etc.) can be performed with ease.

Simple remote debugging without using Workbench or Tornado is available.



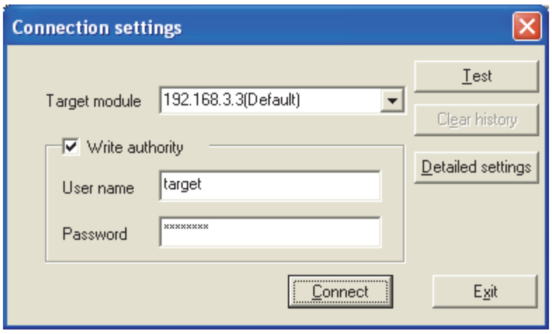
```
tNfsd1      _nfsdRequest 48f73020  60 PEND      48196d0e 48f9eef4       0     0
tNfsd0      _nfsdRequest 48f73490  60 PEND      48196d0e 48fa6ef4       0     0
MQbfTimer0  __ZN11XWDK_C 49de68d8  70 PEND      48198dee 4a033f48       0     0
MQbfTimer1  __ZN11XWDK_C 49de6bd4  70 READY     48198dee 4a038f48       0     0
MQbfTimer2  __ZN11XWDK_C 49deb340  70 PEND      48198dee 4a03df48       0     0
MQbfTimer4  __ZN11XWDK_C 4a0361f4  70 PEND      48198dee 4a047f48       0     0
MQbfTimer5  __ZN11XWDK_C 4a036934  70 READY     48198dee 4a04cf48       0     0
MQbfTimer6  __ZN11XWDK_C 4a03b0d4  70 READY     48198dee 4a051f48       0     0
MQbfTimer7  __ZN11XWDK_C 4a03b7f4  70 READY     48198dee 4a056f48       0     0
MQbfTimer8  __ZN11XWDK_C 4a040020  70 PEND      48198dee 4a05bf48       0     0
MQbfMain    493b4b40     49cd8558  80 PEND      48198dee 49d86f54       0     0

Type <CR> to continue, Q<CR> to stop:
MQbfEvtLog  _XWDKTaskEvt 49d6e59c  80 PEND      48196d0e 49d8ee0c       0     0
MQbfMltEvt  __ZN4XWDK20X 49d8ca18  80 PEND      48198dee 49dedf48       0     0
MQbfUtlCmn  _utlCommonTa 4a0a11b0  99 PEND      48198dee 4a0ba9d4       0     0
MQbfUtlQbf  _utlQbfTask  4a040838  99 PEND      48198dee 4a0c79e4       0     0
MQbfUtlMd   _utlMdTask   4a0a157c  99 PEND      48198dee 4a0d49e4       0     0
MCFMount    4800ac9c     492055f0 100 PEND      481993ac 49210f60       0     0
MCFUnmount  4800ad80     49205a60 100 PEND      481993ac 49215f60       0     0
MSRAMSnd    48015748     4920a09c 100 PEND      48198dee 4921af20       0     0
MEet        _gsErrEntryT 4a092aec 100 DELAY     4819ddf2 4a0abf70       0     1
MfDvEcc     48013ca4     49308314 120 PEND      48198dee 4931ae40       0     0
value = 0 = 0x0
->
```

Figure 1.15 Telnet function

(13)Login user access restriction can prevent illegal access.

Login users can be set (added/deleted) for the C Controller module.

This can restrict parameter writing with FTP or from each utility as well as the usage of the Telnet function.



Connection settings

Target module: 192.168.3.3(Default)

☑ Write authority

User name: target

Password: ********

Figure 1.16 Login user access restriction

(14)User programs can be stored in the built-in standard RAM or ROM.

The C Controller module has a built-in standard RAM or ROM.

Therefore, user programs can be stored even if there is no external memory such as a CompactFlash card.

Data stored in the standard RAM are retained by battery power even if the C Controller module is powered off or reset.

For the memory capacity, refer to the following.

☞ Page 8-5, Section 8.1.1 (3)

(15)CompactFlash card allows large-volume data storage.

By using a CompactFlash card, the C Controller module can store large volumes of data depending on the capacity of the card.

(16)Retention and monitoring of battery-backed-up RAM data

Once data are stored in the battery-backed-up RAM using a user program (QBF_ReadSRAM, QBF_WriteSRAM, QBF_ReadSRAM_ISR, and QBF_WriteSRAM_ISR functions), they will be retained by battery power even if the C Controller module is powered off or reset.

This allows the user program to be executed with the data stored last time (when power is turned off or reset) every time the power is turned on, even if the power is turned off or reset.

The data in the battery-backed-up RAM can be monitored in C Controller setting utility.

For the memory capacity, refer to the following.

☞ Page 8-5, Section 8.1.1 (3)

(17)RAM disk function

The C Controller module can create a RAM disk on the work RAM using a user program.

The FTP function allows transfer of files including user program files from a personal computer to the created RAM disk in the C Controller module.

Compared with the standard ROM or CompactFlash cards, the speed of file access from/to the RAM disk is faster.

Therefore, the RAM disk is a storage medium suitable for the data that do not need battery backup or that will be used only during system operation.

The RAM disk is formatted when the C Controller module is powered off or reset.

For the memory capacity, refer to the following.

☞ Page 8-5, Section 8.1.1 (3)

(18)The device function allows an access similar to that of a programmable controller CPU.

A device such as a programmable controller CPU can be created in the work RAM of the C Controller module.
Access methods from peripheral devices (GOT or MX Component and others) are the same as those for the programmable controller CPU. Therefore, connectivity is improved.

Device functions cannot be used for the following C Controller modules.

- Q12DCCPU-V with a serial number whose first five digits are "12041" or earlier
- Q06CCPU-V(-B)

(a) Communication with a GOT without using any program (connecting a GOT in a bus topology)

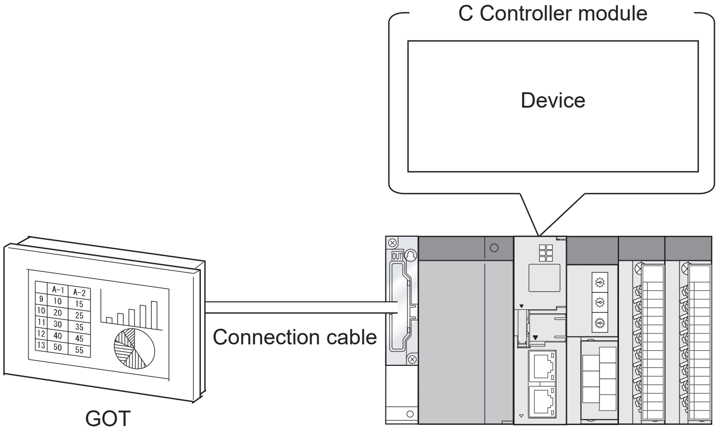
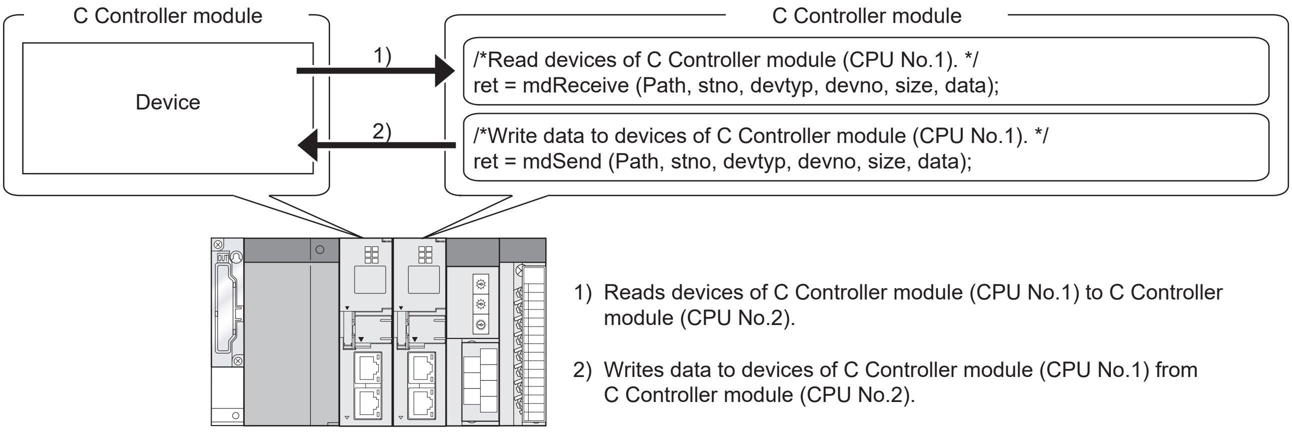


Figure 1.17 Connecting a GOT in a bus topology

(b) Data exchange between two C Controller modules



```
/*Read devices of C Controller module (CPU No.1). */
ret = mdReceive (Path, stno, devtyp, devno, size, data);
```

```
/*Write data to devices of C Controller module (CPU No.1). */
ret = mdSend (Path, stno, devtyp, devno, size, data);
```

1) Reads devices of C Controller module (CPU No.1) to C Controller module (CPU No.2).

2) Writes data to devices of C Controller module (CPU No.1) from C Controller module (CPU No.2).

Figure 1.18 Data exchange between two C Controller modules

(c) Connection with MX Component

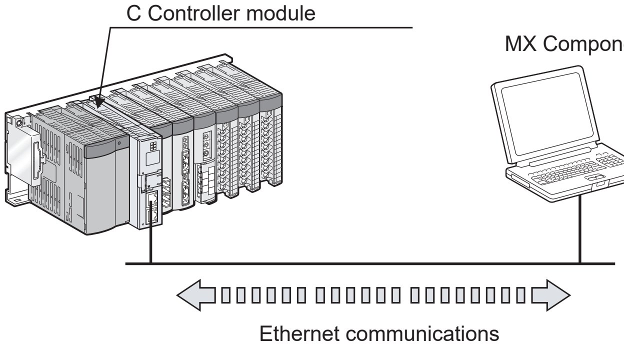


Figure 1.19 Connection with MX Component

(19)Mode change expands the function (For Q12DCCPU-V with a serial number whose first five digits are "15102" or later)

By switching the mode (from basic mode to extended mode) using the module initialization setting, the function of Q12DCCPU-V, such as the dedicated instructions, memory, and network, are added.

For the extended mode, refer to the following manual.

☞ MELSEC-Q C Controller Module User's Manual

CHAPTER 2 SYSTEM CONFIGURATION

This chapter explains system configurations, applicable modules, and connectable devices when using the C Controller module.

2.1 System Configuration

This section introduces C Controller system configurations, including devices used for each base unit, connection with an environment development, and entire system overview. For the multiple CPU system configuration, refer to CHAPTER 10 and CHAPTER 11.

2.1.1 Devices used for each base unit

(1) When using the main base unit (Q3□B)

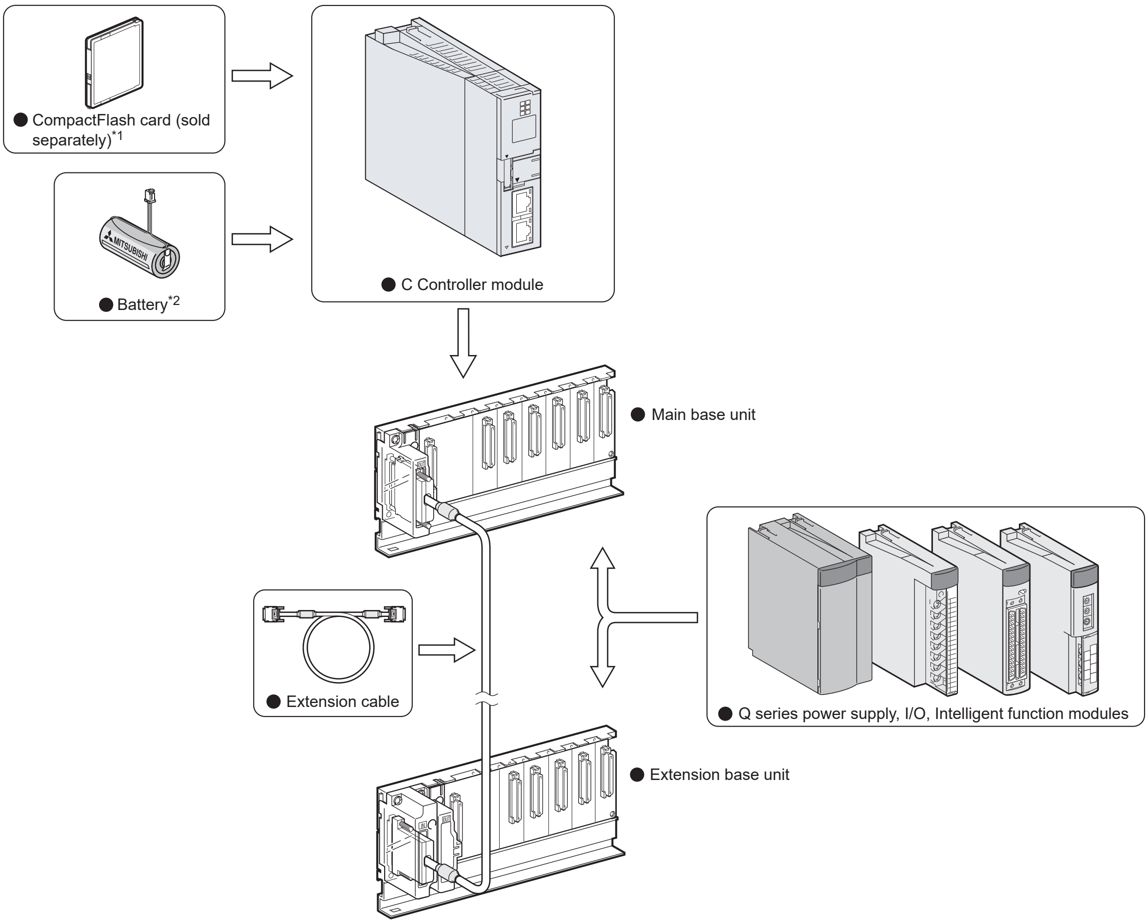

● CompactFlash card (sold separately)*1
● Battery*2
MITSUBISHI
● C Controller module
● Main base unit
● Extension cable
● Q series power supply, I/O, Intelligent function modules
● Extension base unit

Figure 2.1 When a main base unit is used

* 1 For the Q06CCPU-V-B, CompactFlash cards cannot be used.
* 2 For the Q06CCPU-V(-B), the Q7BAT cannot be used.

Remark

- For applicable base units, extension cables, and power supply modules, refer to the following.
  ☞ Page 2-9, Section 2.1.4 (1)
- For applicable I/O modules and intelligent function modules, refer to the following.
  ☞ Page 2-17, Section 2.2.1

(2) When using the slim type main base unit (Q3□SB)

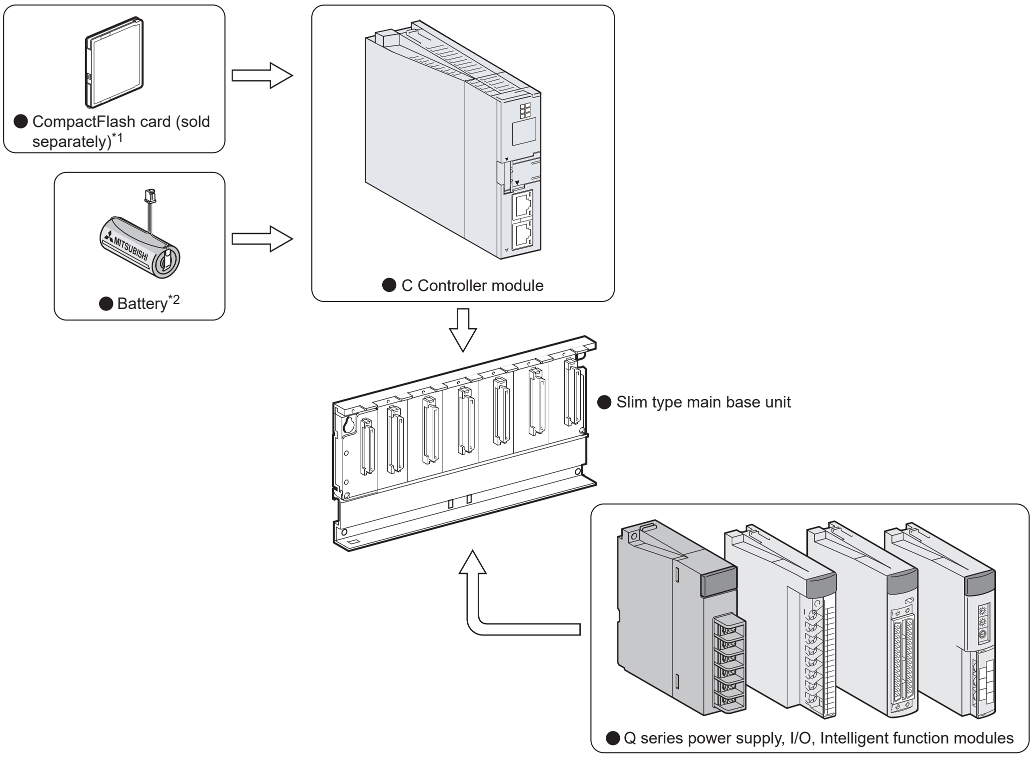

● CompactFlash card (sold separately)*1
● Battery*2
● C Controller module
● Slim type main base unit
● Q series power supply, I/O, Intelligent function modules

Figure 2.2 When a slim type main base unit is used

* 1 For the Q06CCPU-V-B, CompactFlash cards cannot be used.

* 2 For the Q06CCPU-V(-B), the Q7BAT cannot be used.

Remark

- For applicable base units and power supply modules, refer to the following.
  ☞ Page 2-11, Section 2.1.4 (2)
- For applicable I/O modules and intelligent function modules, refer to the following.
  ☞ Page 2-17, Section 2.2.1

(3) When using the multiple CPU high speed main base unit (Q3□DB)

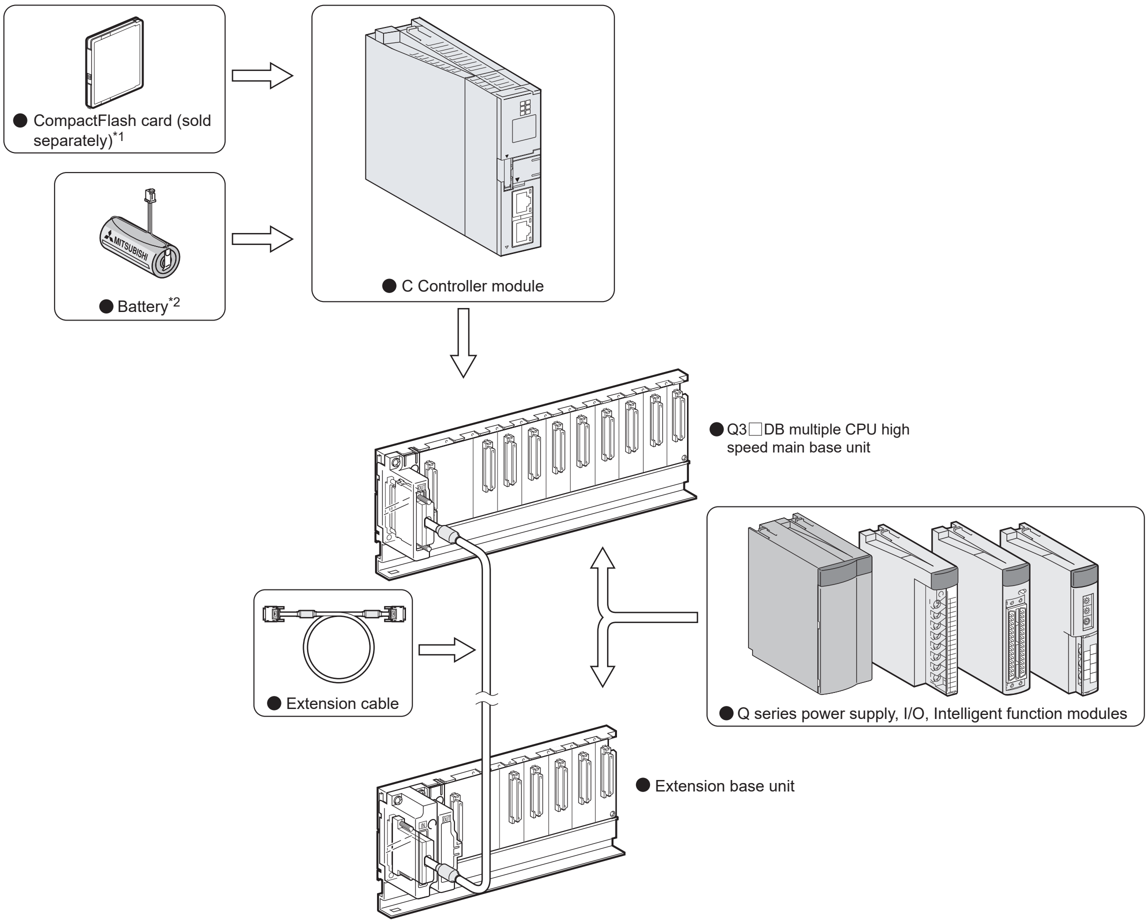

● CompactFlash card (sold separately)*1
● Battery*2
● C Controller module
● Q3□DB multiple CPU high speed main base unit
● Extension cable
● Q series power supply, I/O, Intelligent function modules
● Extension base unit

Figure 2.3 When a multiple CPU high speed main base unit is used

* 1 For the Q06CCPU-V-B, CompactFlash cards cannot be used.
* 2 For the Q06CCPU-V(-B), the Q7BAT cannot be used.

Remark

- For applicable base units and power supply modules, refer to the following.
  ☞ Page 2-12, Section 2.1.4 (3)
- For applicable I/O modules and intelligent function modules, refer to the following.
  ☞ Page 2-17, Section 2.2.1

2.1.2 Connection with a development environment

(1) For the Q12DCCPU-V

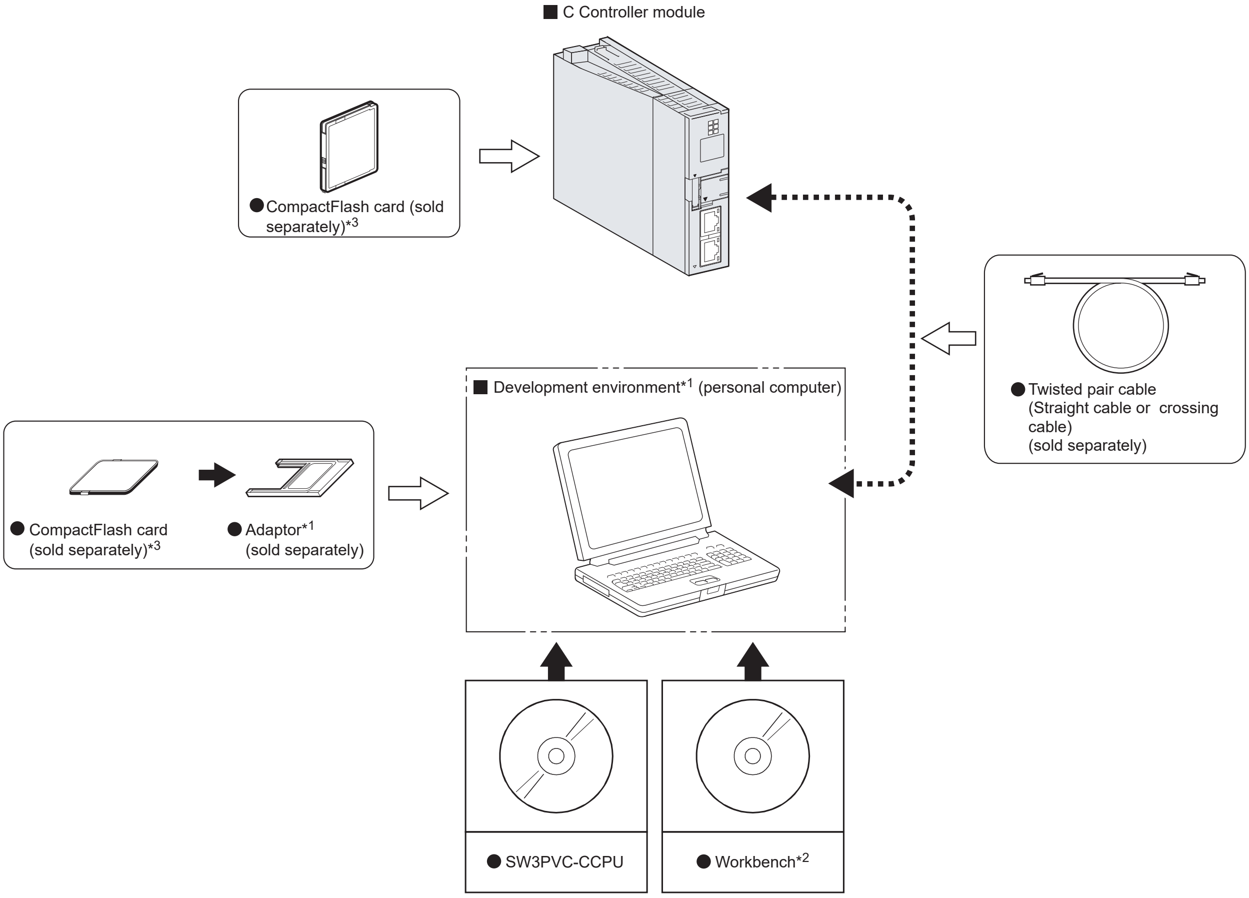

■ C Controller module
● CompactFlash card (sold separately)*3
● Twisted pair cable (Straight cable or crossing cable) (sold separately)
■ Development environment*1 (personal computer)
● CompactFlash card (sold separately)*3
● Adaptor*1 (sold separately)
● SW3PVC-CCPU
● Workbench*2

Figure 2.4 Connection with a development environment

* 1 To write a user file to a CompactFlash card, a personal computer such as a laptop that has the PCMCIA interface, or a personal computer with a CompactFlash card reader/writer connected is required.

* 2 CW Workbench or Wind River Workbench (sold separately).
For Wind River Workbench specifications and inquiries, visit the website of Wind River Systems, Inc.
www.windriver.com

(2) For the Q06CCPU-V(-B)

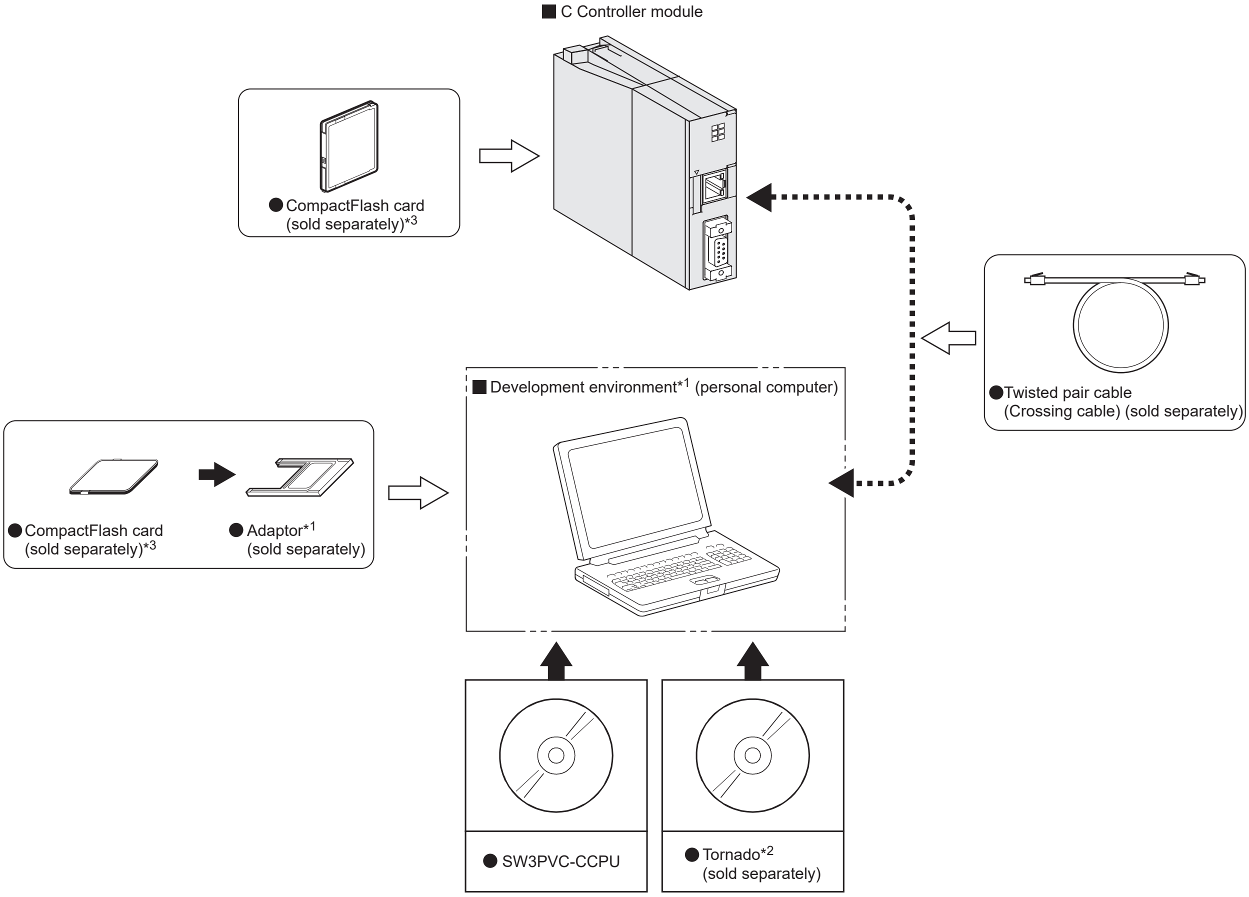

■ C Controller module
● CompactFlash card (sold separately)*3
● Twisted pair cable (Crossing cable) (sold separately)
■ Development environment*1 (personal computer)
● CompactFlash card (sold separately)*3
● Adaptor*1 (sold separately)
● SW3PVC-CCPU
● Tornado*2 (sold separately)

Figure 2.5 Connection with a development environment

* 1 To write a user file to a CompactFlash card, a personal computer such as a laptop that has the PCMCIA interface, or a personal computer with a CompactFlash card reader/writer connected is required.

* 2 For Tornado specifications and inquiries, visit the website of Wind River Systems, Inc. www.windriver.com

* 3 For the Q06CCPU-V-B, the CompactFlash card cannot be used.

2.1.3 Connection with peripheral devices

(1) For the Q12DCCPU-V

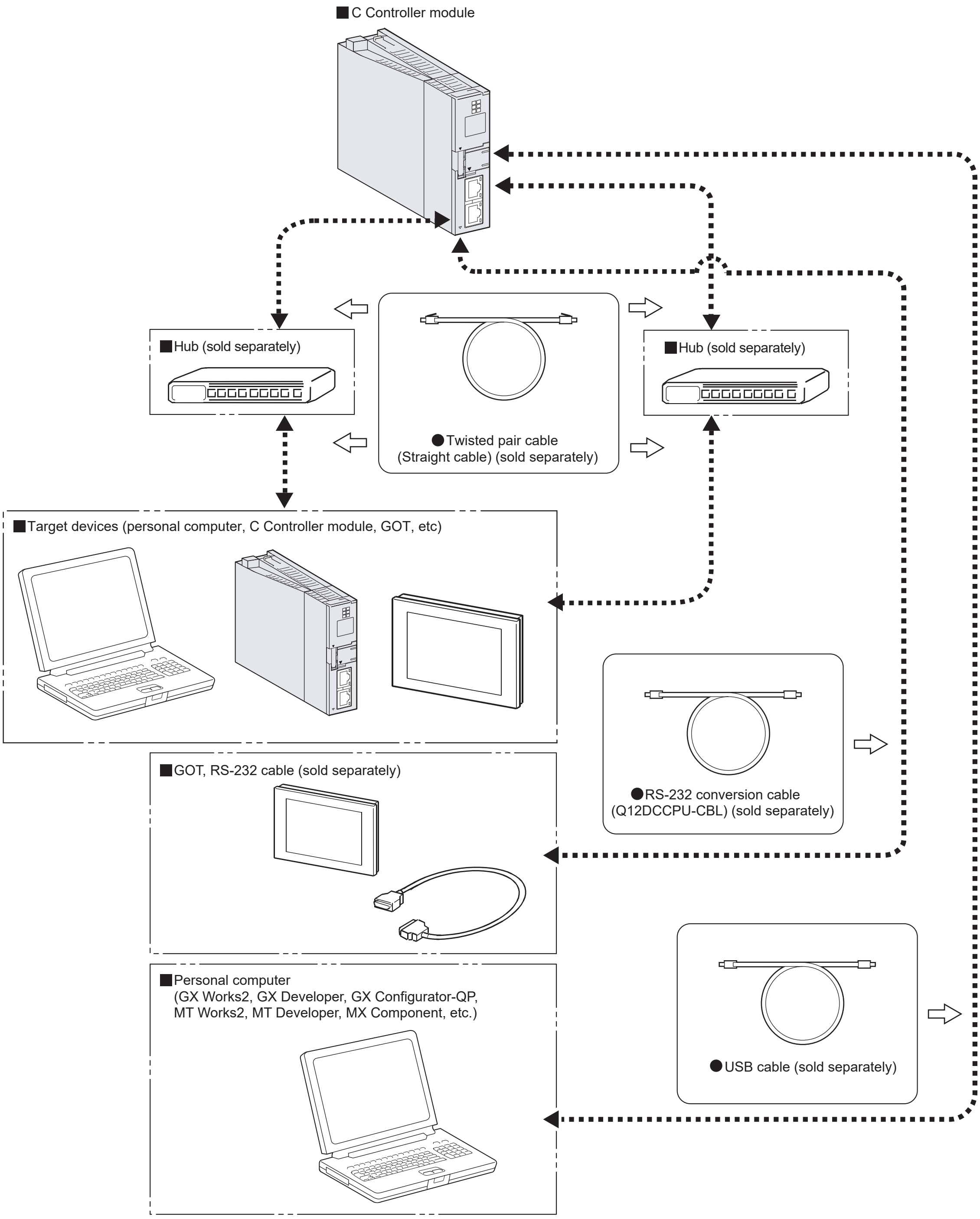

■C Controller module
■Hub (sold separately)
●Twisted pair cable
(Straight cable) (sold separately)
■Hub (sold separately)
■Target devices (personal computer, C Controller module, GOT, etc)
●RS-232 conversion cable
(Q12DCCPU-CBL) (sold separately)
■GOT, RS-232 cable (sold separately)
●USB cable (sold separately)
■Personal computer
(GX Works2, GX Developer, GX Configurator-QP,
MT Works2, MT Developer, MX Component, etc.)

Figure 2.6 Connection with peripheral devices

(2) For the Q06CCPU-V(-B)

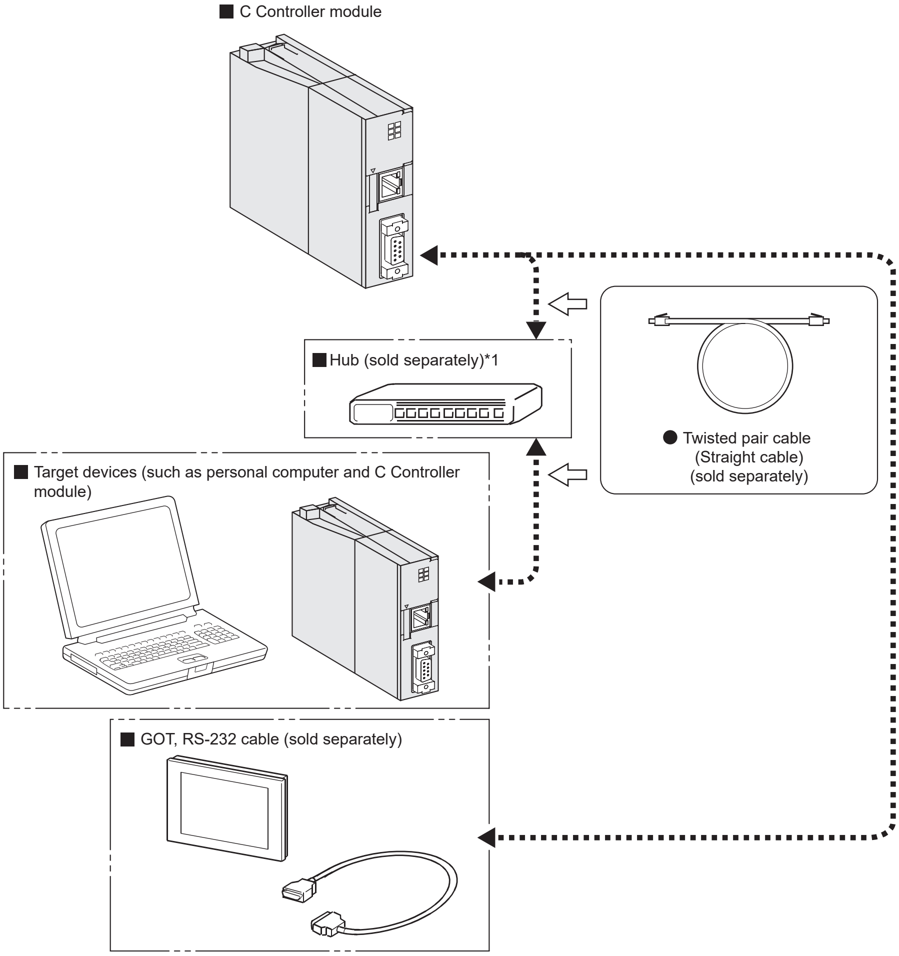

■ C Controller module
■ Hub (sold separately)*1
● Twisted pair cable
(Straight cable)
(sold separately)
■ Target devices (such as personal computer and C Controller module)
■ GOT, RS-232 cable (sold separately)

Figure 2.7 Connection with peripheral devices

* 1 Q06CCPU-V does not support the IEEE802.3x flow control.
Therefore, when the load of an Ethernet line is high in the connection with the hub supporting IEEE802.3x, the data to be sent to the module may be lost.
If the above mentioned phenomenon occurs, add the hubs and reduce the load on the Ethernet line applied on single hub.

2.1.4 System configuration overview

(1) When using the main base unit (Q3□B)

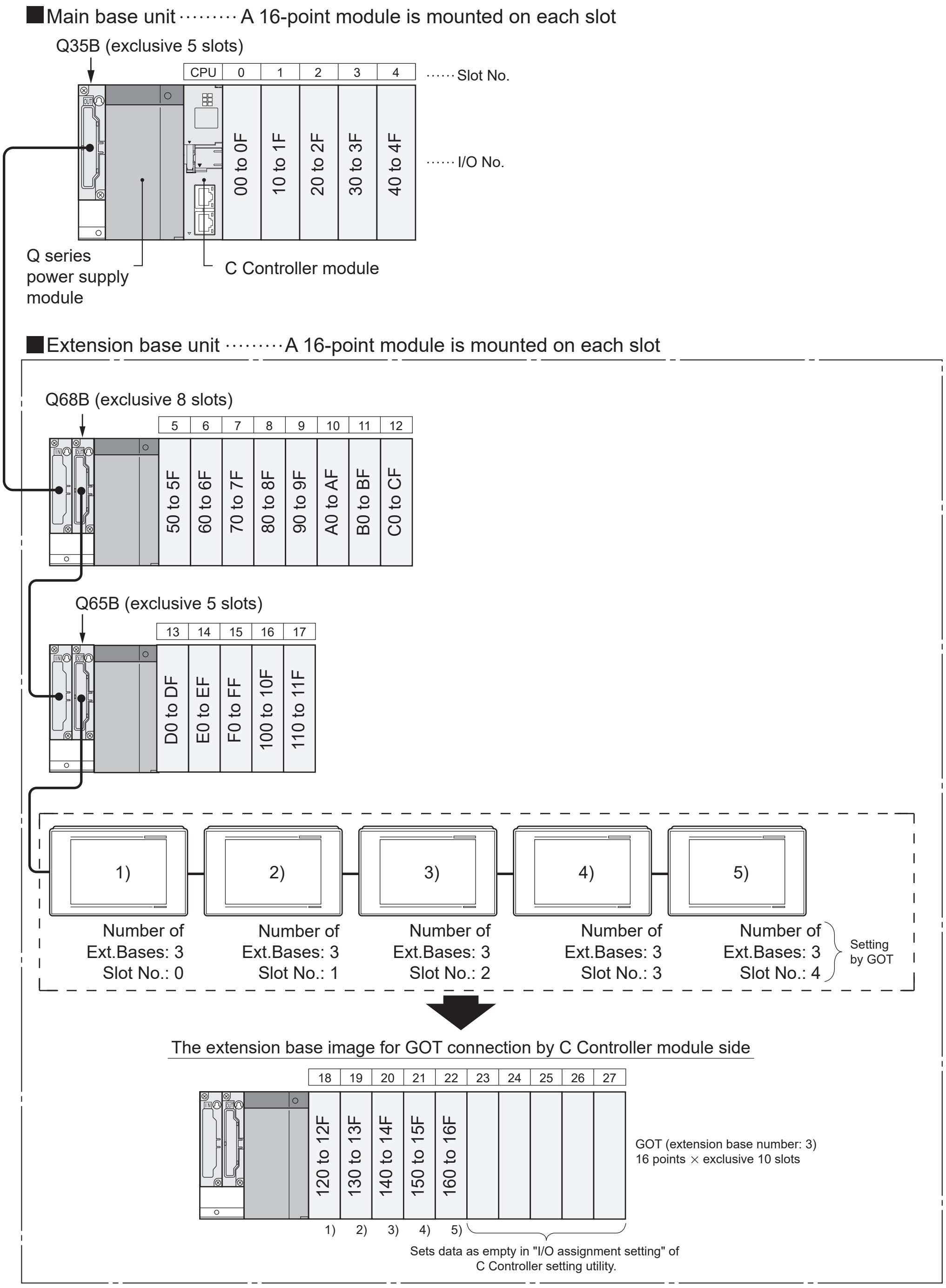

■Main base unit ········· A 16-point module is mounted on each slot
Q35B (exclusive 5 slots)
CPU
0
1
2
3
4
······ Slot No.
00 to 0F
10 to 1F
20 to 2F
30 to 3F
40 to 4F
······ I/O No.
Q series power supply module
C Controller module
■Extension base unit ········· A 16-point module is mounted on each slot
Q68B (exclusive 8 slots)
5
6
7
8
9
10
11
12
50 to 5F
60 to 6F
70 to 7F
80 to 8F
90 to 9F
A0 to AF
B0 to BF
C0 to CF
Q65B (exclusive 5 slots)
13
14
15
16
17
D0 to DF
E0 to EF
F0 to FF
100 to 10F
110 to 11F
1)
2)
3)
4)
5)
Number of Ext.Bases: 3 Slot No.: 0
Number of Ext.Bases: 3 Slot No.: 1
Number of Ext.Bases: 3 Slot No.: 2
Number of Ext.Bases: 3 Slot No.: 3
Number of Ext.Bases: 3 Slot No.: 4
Setting by GOT
The extension base image for GOT connection by C Controller module side
18
19
20
21
22
23
24
25
26
27
120 to 12F
130 to 13F
140 to 14F
150 to 15F
160 to 16F
GOT (extension base number: 3)
16 points × exclusive 10 slots
1)
2)
3)
4)
5)
Sets data as empty in "I/O assignment setting" of C Controller setting utility.

Figure 2.8 Configuration example of a system using a main base unit

Table 2.1 Restrictions on the system configuration

Maximum number of extension base units	7 units	
Maximum number of I/O modules	64 modules	
Main base unit model	Q33B, Q35B, Q38B, Q312B	
Extension base unit model	No power supply module required	Q52B, Q55B
	Q series power supply module required	Q63B, Q65B, Q68B, Q612B
Extension cable model	QC05B, QC06B, QC12B, QC30B, QC50B, QC100B	
Q series power supply module model	Q61P-A1, Q61P-A2, Q61P, Q61P-D, Q62P, Q63P, Q64P, Q64PN	

■ Precautions

- The total extension cable length must be 13.2m (43.31 feet) or less.
- Do not install extension cables together with the main circuit lines (high voltage and large current).
- Set a unique number for each extension base.
- The Q6□RB and QA1S6□B extension base units cannot be used.
- Connect an extension cable from OUT of the extension cable connector of the base unit to IN of the next extension base unit.
- Mounting 65 or more modules causes an error.

(2) When using the slim type base unit (Q3□SB)

■ Slim type main base unit ···A 32-point module is mounted on each slot

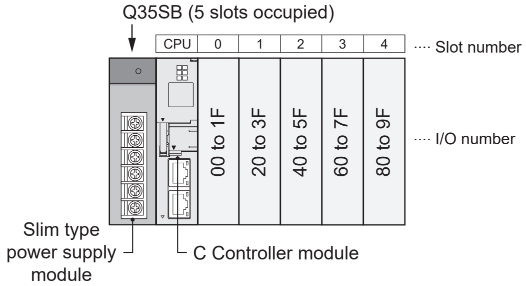


Figure 2.9 Configuration example of a system using a slim type base unit

Table 2.2 Restrictions on the system configuration

Maximum number of I/O modules	5 modules
Slim type base unit model	Q32SB, Q33SB, Q35SB
Slim type power supply module model	Q61SP

■ Precautions

- The slim type main base unit has no extension cable connector.
  Connection of extension base units or GOT (in a bus topology) is not allowed.

(3) When using the multiple CPU high speed main base unit (Q3□DB)

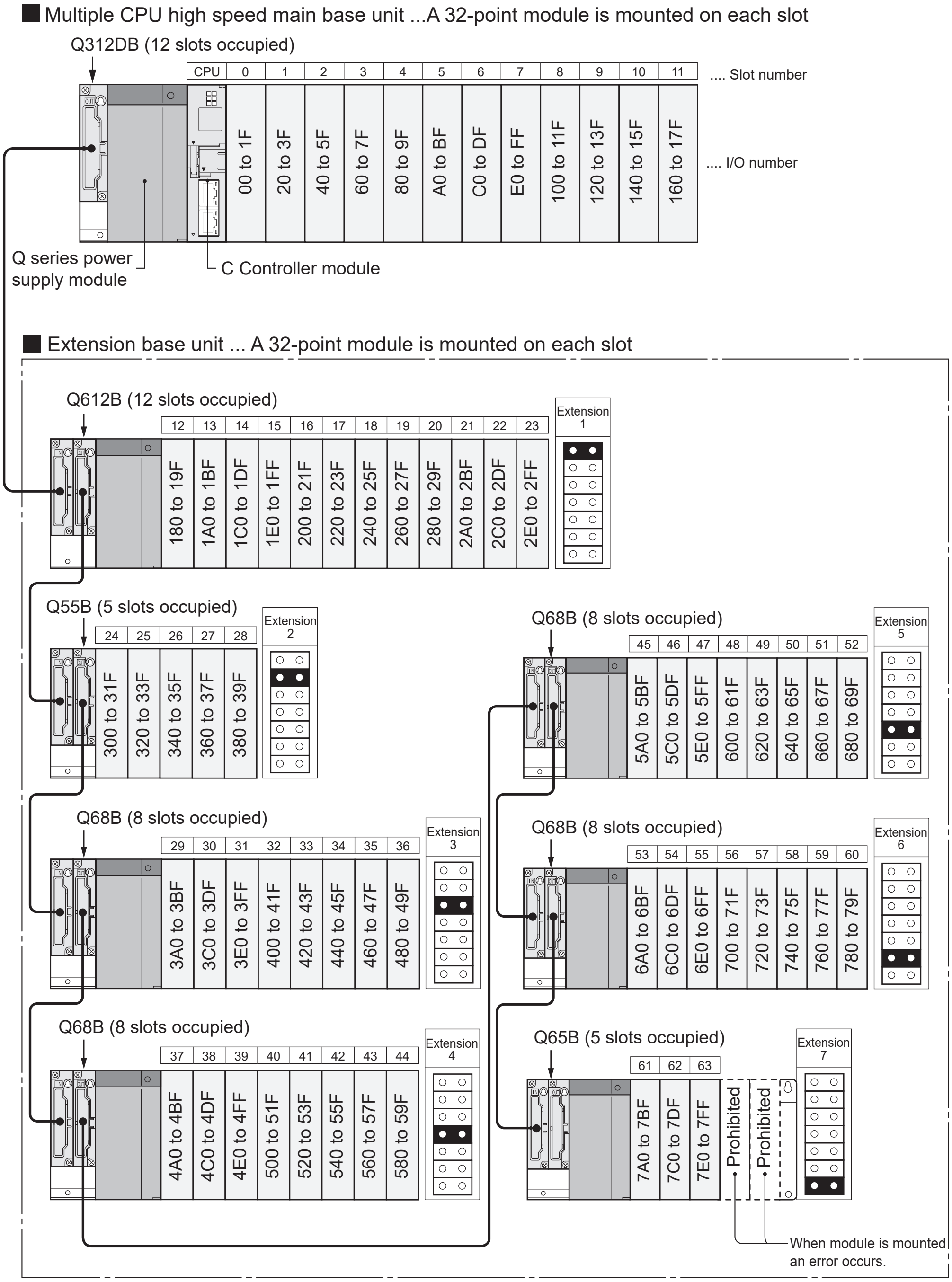


Figure 2.10 Configuration example of a system using a multiple CPU high speed main base unit

Table 2.3 Restrictions on the system configuration

Maximum number of extension base units	7 units	
Maximum number of I/O modules	64 modules	
Applicable main base unit model	Q35DB, Q38DB, Q312DB	
Applicable extension base unit model	No power supply module required	Q52B, Q55B
	Q series power supply module required	Q63B, Q65B, Q68B, Q612B
Extension cable model	QC05B, QC06B, QC12B, QC30B, QC50B, QC100B	
Q series power supply module model	Q61P-A1, Q61P-A2, Q61P, Q61P-D, Q62P, Q63P, Q64P, Q64PN	

■ Precautions

- The total extension cable length must be 13.2m (43.31 feet) or less.
- Do not install the extension cables together with the main circuit lines (high voltage and large current).
- Set a unique number for each extension base.
- The Q6□RB and QA1S6□B extension base units cannot be used.
- Connect the extension cable from OUT of the extension cable connector of the base unit to IN of the next extension base unit.
- Mounting 65 or more modules causes an error.

2.1.5 Connecting a GOT in a bus topology

A GOT can be connected in a bus topology using the extension cable connector of a main or extension base unit in the systems where the C Controller module is used.
This section describes the system configuration for the case where a GOT is connected to the bus.
For details, refer to the following.
☞ GOT1000 Series Connection Manual (Mitsubishi Products)

A GOT cannot be connected in a bus topology with the following C Controller modules.

- Q12DCCPU-V with a serial number whose first five digits are "12041" or earlier
- Q06CCPU-V(-B)

(1) How the C Controller module recognizes the GOT

If a GOT is connected in a bus topology, the C Controller module recognizes the GOT as an intelligent function module of 16 I/O points.
For this reason, the I/O points must be assigned to the C Controller module in GOT setup.
(When a GOT is connected in a bus topology, it needs to occupy one extension base (16 points × 10 slots).)
For details of the GOT setup, refer to the following.
☞ GOT1000 Series Connection Manual (Mitsubishi Products)

(2) Maximum number of GOTs

Up to five GOTs can connected in a bus topology.

(3) Precautions

- In a bus topology, the GOT must be connected to the last base unit. A GOT cannot be connected between base units.

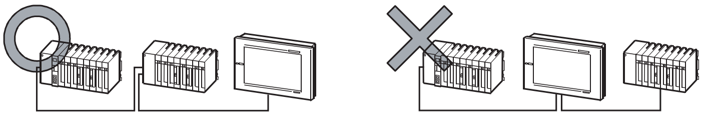


Figure 2.11 Precautions for connecting a GOT in a bus topology

- Select an appropriate extension cable for connecting the GOT to the bus so that the total extension cable length is 13.2 m at maximum.
- A bus extension connector box (A9GT-QCNB) must be used if the first GOT is to be placed at a distance of more than 13.2 meters away.
  ☞ A9GT-QCNB type Bus extension connector box User's Manual
- Configure the bus connection settings (extension base number and slot number settings) on the GOT before activating the C Controller module and GOT.
- Power on the C Controller module and GOT in any of the following sequences.
  - Power on the C Controller module and GOT at the same time.
  - Power on the C Controller module and then GOT.
- Only the GOT 1000 series can be used.

- The following system alarm may occur in the GOT at start-up, depending on the system configuration and GOT settings.
  "402 A communication timeout occurred. Check the communication path or module."
  To avoid the occurrence of the above system alarm, set the GOT title display time to be sufficiently longer than the C Controller module start-up time.
  For how to set the GOT title display time, refer to the following.
  ☞ GT Designer3 Version1 Screen Design Manual (Fundamentals)

(4) System configuration overview

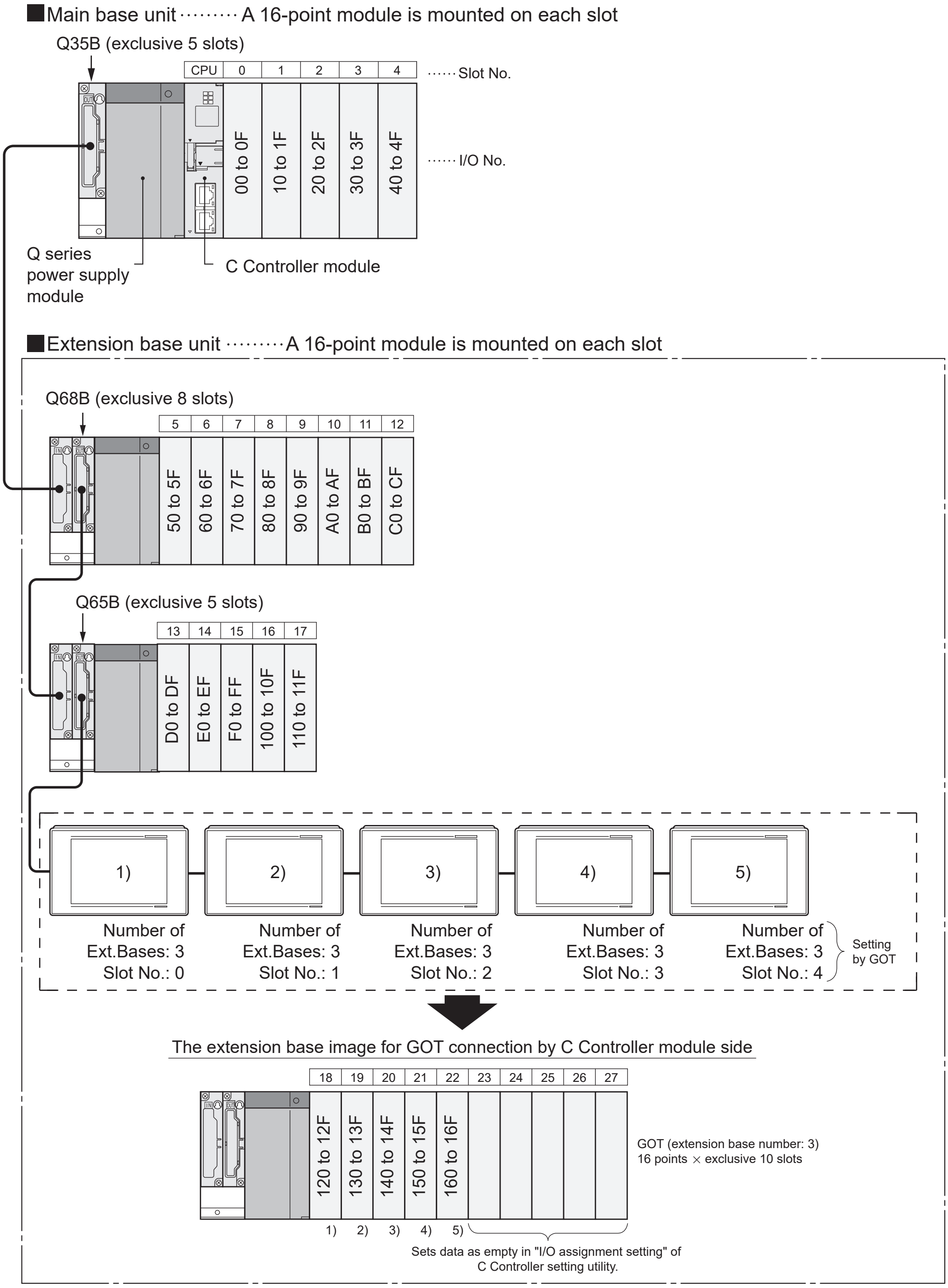

■Main base unit ········· A 16-point module is mounted on each slot
Q35B (exclusive 5 slots)
CPU
0
1
2
3
4
······ Slot No.
00 to 0F
10 to 1F
20 to 2F
30 to 3F
40 to 4F
······ I/O No.
Q series power supply module
C Controller module
■Extension base unit ·········A 16-point module is mounted on each slot
Q68B (exclusive 8 slots)
5
6
7
8
9
10
11
12
50 to 5F
60 to 6F
70 to 7F
80 to 8F
90 to 9F
A0 to AF
B0 to BF
C0 to CF
Q65B (exclusive 5 slots)
13
14
15
16
17
D0 to DF
E0 to EF
F0 to FF
100 to 10F
110 to 11F
1)
2)
3)
4)
5)
Number of Ext.Bases: 3 Slot No.: 0
Number of Ext.Bases: 3 Slot No.: 1
Number of Ext.Bases: 3 Slot No.: 2
Number of Ext.Bases: 3 Slot No.: 3
Number of Ext.Bases: 3 Slot No.: 4
Setting by GOT
The extension base image for GOT connection by C Controller module side
18
19
20
21
22
23
24
25
26
27
120 to 12F
130 to 13F
140 to 14F
150 to 15F
160 to 16F
GOT (extension base number: 3)
16 points × exclusive 10 slots
1)
2)
3)
4)
5)
Sets data as empty in "I/O assignment setting" of C Controller setting utility.

Figure 2.12 Example of a system configuration using GOT

2.1.6 Precautions for system configuration

This section explains the precautions for system configuration

(1) Main base unit not applicable to the C Controller module

The following main base unit is not applicable.

- Redundant power main base unit (Q3□RB)

(2) Extension base units not applicable to the C Controller module

The following extension main base units are not applicable.

- Redundant power extension base unit (Q6□RB)
- AnS series extension base unit (QA1S6□B)

Remark

For applicable base units, extension cables, and power supply modules, refer to the following.

☞ Page 2-9, Section 2.1.4

2.2 Applicable Modules

This section introduces modules applicable to the C Controller module.
Precautions for using I/O modules and intelligent function modules, refer to the following.
☞ Page 2-19, Section 2.2.2

2.2.1 Applicable I/O modules and intelligent function modules

The following table indicates the MELSEC-Q series I/O modules and intelligent function modules that can be used with the C Controller module.
The MELSEC-AnS/Q2AS series I/O modules and special function modules are not applicable.

Table 2.4 Applicable modules

Module type	Model	Restrictions
AC input module	Q series modules are applicable. For details, refer to the manual for respective modules.	-
DC input module		
DC/AC input module		
High-speed input module	QX40H, QX70H, QX80H, QX90H	-
High-speed output module	QY41H	-
Contact output module	Q series modules are applicable. For details, refer to the manual for respective modules.	-
Triac output module		
Transistor output module		
DC input/transistor output combined module		
Interrupt module	QI60	-
	QX40H, QX70H, QX80H, QX90H (The module is available only when the interrupt module is selected by setting the function selector switch (SW2) to OFF.)	
A/D converter module	Q64AD, Q68ADV, Q68ADI	Use the product of function version B or later.
Channel isolated A/D converter module	Q64AD-GH, Q62AD-DGH, Q68AD-G, Q66AD-DG	-
D/A converter module	Q62DAN, Q64DAN, Q68DAVN, Q68DAIN	-
	Q62DA, Q64DA, Q68DAV, Q68DAI	Use the product of function version B or later.
Channel-isolated D/A converter module	Q66DA-G, Q62DA-FG	-
Analog Input/Output Module	Q64AD2DA	-
Load Cell Input Module	Q61LD	-
Channel-isolated temperature input module	Q64TD, Q68TD-G-H01, Q68TD-G-H02, Q68RD3-G	-
	Q64TDV-GH, Q64RD-G	Use the product of function version B or later.
Temperature input module	Q64RD	-
Temperature control module	Q64TCTTN, Q64TCTTBWN, Q64TCRTN, Q64TCRTBWN	-
	Q64TCTT, Q64TCTTBW, Q64TCRT, Q64TCRTBW	Use the product of function version B or later.
Loop control module	Q62HLC	-

(To the next page)

Table 2.4 Applicable modules (continued)

Module type	Model	Restrictions
High-speed counter module	QD62, QD62D, QD62E, QD64D2, QD63P6, QD65PD2	-
Channel-isolated pulse input module	QD60P8-G	-
MODBUS® interface module	QJ71MB91, QJ71MT91	-
Positioning module	QD75M1, QD75MH1, QD75M2, QD75MH2, QD75M4, QD75MH4, QD74MH8, QD74MH16, QD70D4, QD70D8, QD72P3C3, QD75P1N, QD75P2N, QD75P4N, QD75D1N, QD75D2N, QD75D4N	-
	QD75P1, QD75P2, QD75P4, QD75D1, QD75D2, QD75D4, QD70P4, QD70P8	Use the product of function version B or later.
High-speed data logger module	QD81DL96[*1][*2]	Use the product with a serial number (first five digits) "12062" or later.
MES interface module	QJ71MES96[*1][*2]	Use the product with a serial number (first five digits) "12092" or later.
FL-net module	QJ71FL71-T-F01, QJ71FL71-B5-F01, QJ71FL71-T, QJ71FL71-B5	-
	QJ71FL71, QJ71FL71-F01, QJ71FL71-B2-F01, QJ71FL71-B2	Use the product of function version B or later.
CC-Link module	QJ61BT11N	-
	QJ61BT11	Use the product of function version B or later.
CC-Link/LT module	QJ61CL12	Use the product of function version B or later.
AnyWire DB A20 master module	QJ51AW12D2	-
AS-i module	QJ71AS92	Use the product of function version B or later.
Energy measuring module	QE81WH, QE82LG	-
Serial communication module[*6]	QJ71C24, QJ71C24-R2	Use the product of function version B or later.
	QJ71C24N, QJ71C24N-R2, QJ71C24N-R4	-
CC-Link IE controller network module	QJ71GP21-SX[*3][*4], QJ71GP21S-SX[*3][*4]	Use the product with a serial number (first five digits) "09042" or later.
CC-Link IE field network master/local module	QJ71GF11-T2[*1][*5]	Use the product with a serial number (first five digits) "12062" or later.
MELSECNET/H module	QJ71LP21-25, QJ71LP21S-25, QJ71LP21G, QJ71LP21GE, QJ71BR11	Use the product of function version B or later.
ID interface module	QD35ID1, QD35ID2	-
DeviceNet module	QJ71DN91	Use the product of function version B or later.
PROFIBUS-DP module	QJ71PB93D, QJ71PB92V	-
	QJ71PB92D	Use the product of function version B or later.
GOT	GOT1000 series (in a bus topology)[*1][*2] For applicable GOT models, refer to the following. ☞ GOT1000 Series Connection Manual (Mitsubishi Products)	

[*1] This cannot be used with the Q12DCCPU-V with a serial number (first five digits) "12041" or earlier.

[*2] This cannot be used with the Q06CCPU-V(-B)

[*3] This cannot be used with the Q06CCPU-V with a serial number (first five digits) "10011" or earlier.

[*4] This cannot be used with the Q06CCPU-V-B

[*5] This cannot be used with the Q06CCPU-V(-B) with a serial number (first five digits) "12081" or earlier.

[*6] Only Non-Procedural protocol is supported

2.2.2 Precautions for using I/O modules and intelligent function modules

This section explains the precautions for using I/O modules and intelligent function modules.
For the access range with a non-controlled module in a multiple CPU system configuration, refer to the following.
☞ QCPU User's Manual (Multiple CPU System)

☒POINT

The instructions dedicated to intelligent function modules cannot be used for C Controller modules.

(1) Number of connectable modules

The number of connectable modules is shown below by module type.

Table 2.5 Number of connectable modules

Module type	Number of connectable modules
CC-Link module	Up to 8
CC-Link IE controller network module	Up to 4
MELSECNET/H module	
Interrupt module	Up to 1[*1]
Other I/O module, intelligent function module	Up to 64
GOT (only in a bus topology)	Up to 5

[*1] When the interrupt event setting is specified in the following C Controller modules, up to 64 modules can be installed.
•Q12DCCPU-V with a serial number whose first five digits are "12042" or later

(2) Precautions for using a CC-Link module

The following restrictions are applied to the CC-Link module controlled by the C Controller module.

- The CC-Link dedicated instructions cannot be used.
- Interrupt sequence program start is not applicable.
- Automatic CC-Link start is not applicable.
- The standby master function cannot be used.
- The remote I/O network mode is not available.

(3) Precautions for using a MELSECNET/H module

The following restrictions are applied to the MELSECNET/H module controlled by the C Controller module.

- The MELSECNET/H module cannot be used on a remote I/O network.
  Use of the module is allowed on a PLC-to-PLC network only.
- The module cannot be used as a relay station for the data link transfer function or the routing function.
  When using either of these functions, use a MELSECNET/H module controlled by a programmable controller CPU as a relay station.
- In a multiple CPU system, the host CPU (C Controller module) cannot communicate via the MELSECNET/H module controlled by another CPU. Another CPU also cannot communicate via the MELSECNET/H module controlled by the host CPU.
  The above is applied to the following communications.
  - Communications with MELSOFT connection
  - Communications using dedicated instructions
  - Communications using intelligent function modules
- The link dedicated instructions for MELSECNET/H cannot be used.
  Note that the message communication function equivalent to the SEND and RECV instructions are available. (☞ Page 5-10, Section 5.3)
- Interrupt sequence program start is not applicable.
- The network diagnostics (test) function is not available.
- The simple dual-structure network function is not applicable.
  (The network type cannot be set to "MNET/H standby station")
- When accessing to a redundant CPU on another station, the system of the redundant CPU cannot be specified. Only host system access (station-number-specified access) is available.
- The mode cannot be set to "Debug mode".
- The MELSEC data link functions cannot be used for the Q06CCPU-V-B.
  For how to access a CPU module on another station when using the Q06CCPU-V-B, refer to the following.
  ☞ Page 5-10, Section 5.3

(4) Precautions for using a CC-Link IE controller network module

The following restrictions are applied to the CC-Link IE controller network module controlled by the C Controller module.

- The module cannot be used as a relay station for the data link transfer function or the routing function.
  When using either of these functions, use a CC-Link IE controller network module controlled by a programmable controller CPU as a relay station.
- In a multiple CPU system, the host CPU (C Controller module) cannot communicate via the CC-Link IE Controller Network module controlled by another CPU. Another CPU also cannot communicate via the CC-Link IE Controller Network module controlled by the host CPU.
  The above is applied to the following communications.
  - Communications with MELSOFT connection
  - Communications using dedicated instructions
  - Communications using intelligent function modules
- The link dedicated instructions for the CC-Link IE controller network cannot be used.
  Note that the message communication function equivalent to the SEND and RECV instructions are available. (☞ Page 5-38, Section 5.4)
- Interrupt sequence program start is not applicable.
- When accessing to a redundant CPU on another station, the system of the redundant CPU cannot be specified. Only host system access (station-number-specified access) is available.
- The group cyclic transmission function is not applicable.
- The pairing setting for the redundant system is not applicable.
- For the Q06CCPU-V-B, CC-Link IE controller network modules cannot be used.

(5) Precautions for using the CC-Link IE field network master/local module

The following restrictions are applied to the CC-Link IE field network master/local module controlled by the C Controller module.

(a) There are some unsupported functions.
For the unsupported functions, refer to the following.
☞ MELSEC-Q CC-Link IE Field Network Master/Local Module User's Manual

(b) There is no module configuration/monitoring tool for the CC-Link IE field network master/local module.

- To configure parameters, write them to the buffer memory of the CC-Link IE field network master/local module.
  For the parameter setting method, refer to the following.
  ☞ MELSEC-Q CC-Link IE Field Network Master/Local Module User's Manual
- For troubleshooting of the CC IE field network master/local module, use either of the following methods.
  - If the C Controller module is connected to a CC-Link IE field network master/local module that is mounted with a programmable controller CPU, troubleshoot the problem from the CC-Link IE field network diagnostics in GX Works2.
  - Check the error code in the buffer memory, link special relay (SB), and link special register (SW) of the CC-Link IE field network master/local module.

  For the CC-Link IE field network diagnostics, error codes, link special relays (SB), and link special registers (SW), refer to the following.
  ☞ MELSEC-Q CC-Link IE Field Network Master/Local Module User's Manual

(6) Precautions for using a serial communication module

The serial communication module controlled by the C Controller module supports nonprocedural protocol communications only.

(a) The following are not supported.

- MC protocol and bidirectional protocol
- The instructions dedicated to serial communication modules
- The programmable controller CPU monitoring function
- The modem function

(b) The following settings are required for using an interrupt program.

- Set the interrupt event No. in the <<System settings>> tab of C Controller setting utility.
- Write "1" to the buffer memory (address 2010H/2110H) of the serial communication module.
- Modify the user program so that an interrupt event will be received by the QBF_WaitUnitEvent function to start relevant processing.

(7) Precautions for using an FL-net module

The following restrictions are applied to the FL-net module controlled by the C Controller module.

- The word block read/write request messages using the message transmission function cannot be received.
- The auto refresh function is not applicable.

(8) Precautions on interrupt processing

Interrupts are used for communications of the C Controller module.
These communications may be disabled if a program that disables interrupts is executed.
Also, communication may become slow if a program with frequent interrupts is executed.

(9) Precautions for using GX Configurator

The default intelligent function module parameter settings can be read out from a project file of GX Developer into C Controller setting utility.
GX Works2 project, data can be read out if the project is saved in the GX Developer format.
For precautions on readable project files of GX Developer, refer to the following.
☞ C Controller Module User's Manual (Utility Operation, Programming)

(10) Precautions for using MODBUS® interface module

The following restrictions are applied to the MODBUS® interface module controlled by the C Controller module.

- The dedicated instruction cannot be used.
- The MODBUS® device assignment cannot be used.

2.3 Software Packages

The following software packages can be used for the C Controller module.

(1) For the Q12DCCPU-V

Table 2.6 Software packages

Software	Version
SW3PVC-CCPU	Version 3.03D or later
SW4PVC-CCPU[*1]	Version 4.04E or later
GX Works2	Version 1.50C or later
GX Developer	Version 8.76E or later and must be appropriate to the connected CPU.
GX Configurator-QP	Version 2.31H or later
MT Works2	Versions that must be appropriate to the connected Motion CPU
MT Developer	
MX Component	Version 3.14Q or later

[* 1] Can be connected when using Q12DCCPU-V with a serial number whose first five digits are "15102" or later with the extended mode.

(2) For the Q06CCPU-V

Table 2.7 Software package

Software	Version
SW3PVC-CCPU	Version 3.00A or later

(3) For the Q06CCPU-V-B

Table 2.8 Software package

Software	Version
SW3PVC-CCPU	Version 3.01B or later

2.4 Applicable Devices

This section introduces devices connectable to the C Controller module.

(1) CompactFlash card

One CompactFlash card can be installed for a C Controller module.

Use a CompactFlash card manufactured by Mitsubishi listed in the following table. Failure to do so may cause a problem such as data corruption in the CompactFlash card and system stop.

Table 2.9 Applicable CompactFlash cards

Model name	Applicable models		Description
	Q12DCCPU-V	Q06CCPU-V	
GT05-MEM-128MC	×	○	128MB CompactFlash card
GT05-MEM-256MC	×	○	256MB CompactFlash card
QD81MEM-512MBC	○	○	512MB CompactFlash card
QD81MEM-1GBC	○	○	1GB CompactFlash card
QD81MEM-2GBC	○	×	2GB CompactFlash card
QD81MEM-4GBC	○	×	4GB CompactFlash card
QD81MEM-8GBC	○	×	8GB CompactFlash card

○: Available, ×: N/A

☒POINT

CompactFlash cards have limited lives (the limited number of writes).

- For the live of QD81MEM and GT05-MEM, refer to the following.
  ☞ Page 6-70, Section 6.8.5
- For the live of commercially available CompactFlash cards, refer to the specifications of each product.
- For countermeasures against static electricity for commercially available CompactFlash cards to comply with the EMC directives, refer to Page 6-68, Section 6.8.4.

(2) Twisted pair cable

Use the twisted pair cables that meet IEEE802.3 10BASE-T/100BASE-TX standards.

(a) At 100Mbps

Use either of the following cables.

- Unshielded twisted pair (UTP) cable
  Straight cable: Category 5 or higher
  Crossing cable: Category 5 or 5e
- Shielded twisted pair (STP) cable
  Straight cable: Category 5 or higher
  Crossing cable: Category 5 or 5e

(b) At 10Mbps

Use either of the following cables.

- Unshielded twisted pair (UTP) cable
  Straight cable: Category 3 or higher
  Crossing cable: Category 3 to 5e
- Shielded twisted pair (STP) cable
  Straight cable: Category 3 or higher
  Crossing cable: Category 3 or 5e

☒POINT

In the high-speed communication (100Mbps) by 100BASE-TX connection, a communication error may occur due to high frequency noise generated from a device other than the C Controller module, depending on the installation environment.

When configuring a network system, take the following measures on the C Controller module side to eliminate the influence of high frequency noise.

1. Wiring
   - Do not install the twisted pair cables together with the main circuit lines or power cables.
   - Place the twisted pair cables in a duct.
2. Cable
   - In an environment where the system is susceptible to noise, use shielded twisted pair (STP) cables.
3. Retry processing
   - In an environment where the system is susceptible to noise, include the retry processing in the user program.
4. 10Mbps communication
   - Replace the devices connected to the C Controller module with 10Mbps devices for communication at a speed of 10Mbps.

(3) Hub

The C Controller module discriminates between 10BASE-T and 100BASE-TX and between full-duplex and half-duplex communication modes according to the hub. When connecting a hub that does not have the auto negotiation function, set the hub to the half-duplex communication mode.

(a) For the Q12DCCPU-V
CH1 and CH2 of the C Controller module are assigned to different sub-networks. Do not connect CH1 and CH2 of the C Controller module to the same hub (excluding switching hubs).

(4) RS-232 cable

Use an RS-232-compliant cable that is 15m (49.21 feet) or less in length.

[Recommended cable]
7/0.127□P HRV-SV ••• Specify the number of pairs in □ (For 13 pairs, use 7/0.127 13P HRV-SV.) (Oki Electric Cable Company, Limited)

☒POINT

In RS-232 connection, a communication error may occur due to noise generated from the devices other than the C Controller system depending on the installation environment.
In the environment where the system is susceptible to noise, include the retry processing in the user program.

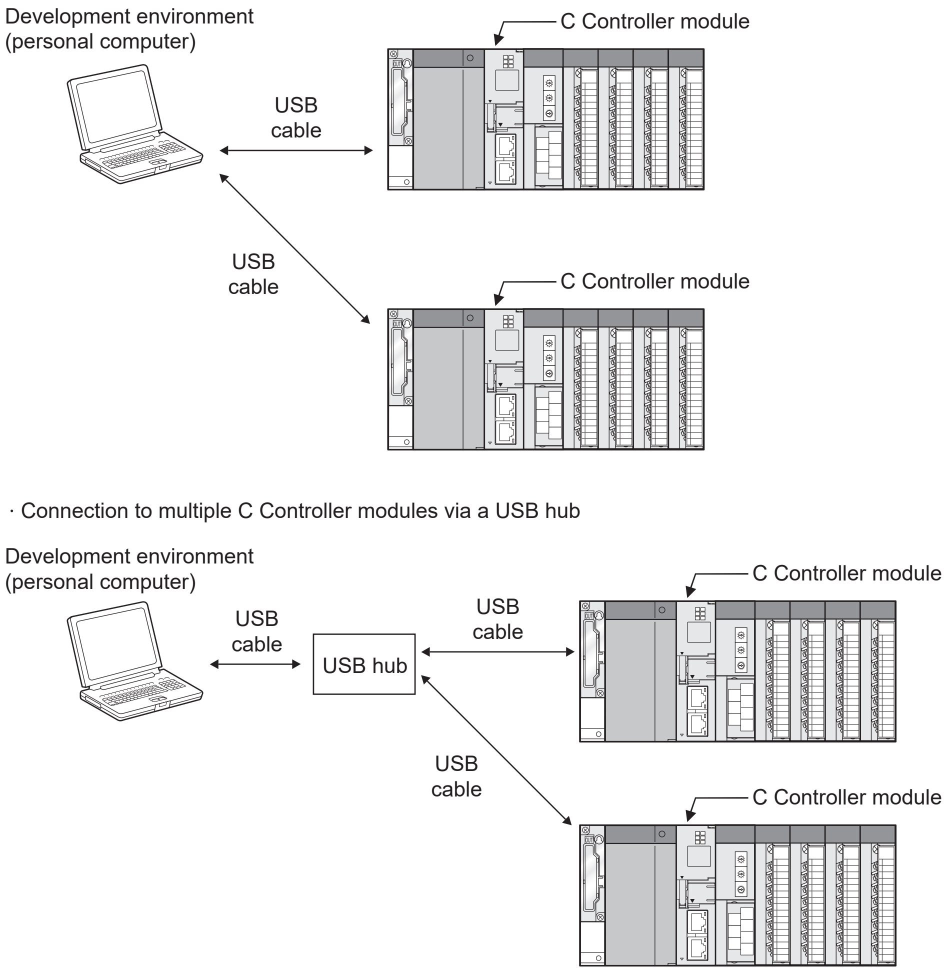
(5) USB cable (Q12DCCPU-V only)

(a) The C Controller module is accessible from the following software packages by pre-installing a USB driver in Windows®.
(For how to install the USB driver, refer to the manual for the software package to be used.)

- GX Works2
- GX Developer
- GX Configurator-QP
- MT Works2
- MT Developer
- MX Component

(b) For use of USB cables, only one C Controller module can be connected.
The following are inappropriate configuration examples.

<Inappropriate configurations>
· Connection from multiple USB ports on a personal computer to multiple C Controller modules



· Connection to multiple C Controller modules via a USB hub

Figure 2.13 Configurations where USB cables cannot be used

(6) Connection of a HMI (Human Machine Interface)

The C Controller module can be connected to a GOT with the following methods.

- Microcomputer connection through the RS-232 interface on the C Controller module*1
- Bus connection through an extension cable connector of a main or extension base unit*1*2

  For the system configurations for connecting a GOT in a bus topology, refer to the following.

  ☞ Page 2-14, Section 2.1.5
- MELSECNET/H connection (PLC-to-PLC network)*1*2
- MELSECNET/10 connection (PLC-to-PLC network)*1*2
- CC-Link IE controller network connection*1*2
- CC-Link connection (intelligent device station)*1*2
- CC-Link connection (via G4)*1*2
- Ethernet connection by the 10BASE-T/100BASE-TX interface of the C Controller module*1*2

* 1 For applicable GOT models, refer to the following.

☞ GOT1000 Series Connection Manual (Mitsubishi Products)

* 2 The module cannot be connected to the following C Controller modules.

- Q12DCCPU-V with a serial number whose first five digits are "12041" or earlier
- Q06CCPU-V(-B)

(7) Connection from GX Works2/GX Developer

(a) Direct connection to the C Controller module

1) For the Q12DCCPU-V

Connection from GX Works2/GX Developer via an RS-232 interface of the C Controller module is not allowed.

Connect the cable to the 10BASE-T/100BASE-TX interface*1 *2 or to the USB interface.

* 1 Connection via the 10BASE-T/100BASE-TX interface is not available with the following C Controller modules.

- Q12DCCPU-V with a serial number whose first five digits are "12041" or earlier

* 2 For connection via the 10BASE-T/100BASE-TX interface, it is necessary to configure the built-in Ethernet port open setting in the <<System Settings>> tab of C Controller setting utility.
For details, refer to the following manual.

☞ C Controller Module User's Manual (Utility Operation, Programming)

For methods for connecting GX Works2/GX Developer, refer to the following.

☞ Page App-25, Appendix 9

2) For the Q06CCPU-V(-B)
GX Works2/GX Developer is not connectable to the C Controller module.

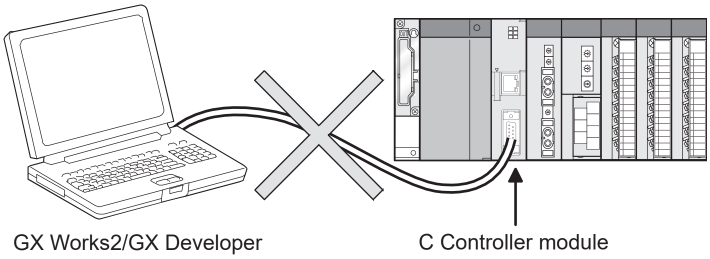


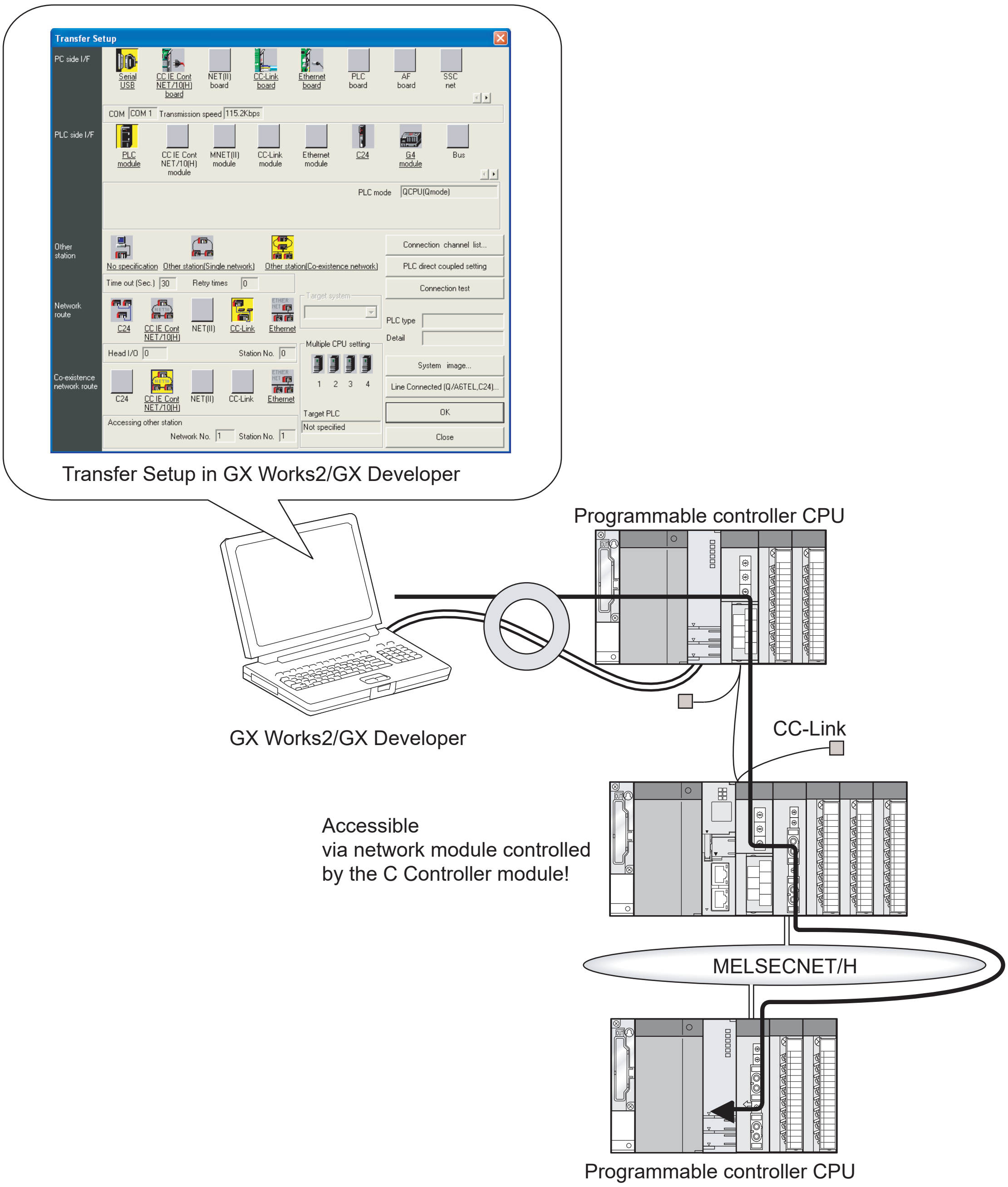
Figure 2.14 Direct connection (not available)

(b) Access to a programmable controller CPU via a network module controlled by a C Controller module

GX Works2/GX Developer connected to a programmable controller CPU can access another programmable controller CPU on another station via a network module (CC-Link module, MELSECNET/H module, or CC-Link IE controller network module) controlled by a C Controller module.

However, GX Developer cannot access a programmable controller CPU on another station via a serial communication module controlled by a C Controller module.

1) Other station access via CC-Link and MELSECNET/H



COM: COM 1 Transmission speed: 115.2Kbps

PLC mode: QCPU(Qmode)

Time out (Sec.): 30 Retry times: 0

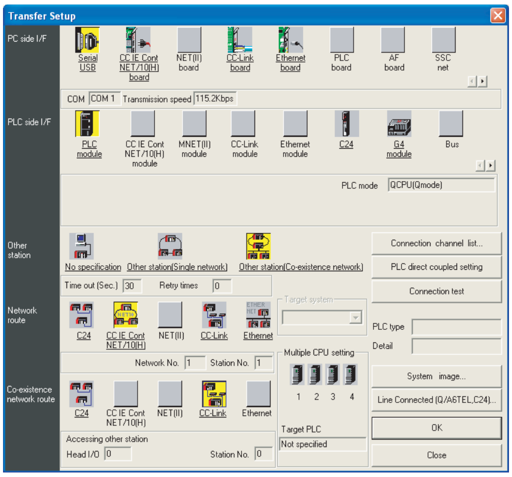
Head I/O: 0 Station No.: 0

Accessing other station Network No.: 1 Station No.: 1

Target PLC: Not specified

Figure 2.15 Other station access via CC-Link and MELSECNET/H

2) Other station access via MELSECNET/H and CC-Link



COM: COM 1 Transmission speed: 115.2Kbps

PLC mode: QCPU(Qmode)

Time out (Sec.): 30 Retry times: 0

Network No.: 1 Station No.: 1

Accessing other station
Head I/O: 0 Station No.: 0

Target PLC: Not specified

Transfer Setup in GX Works2/GX Developer

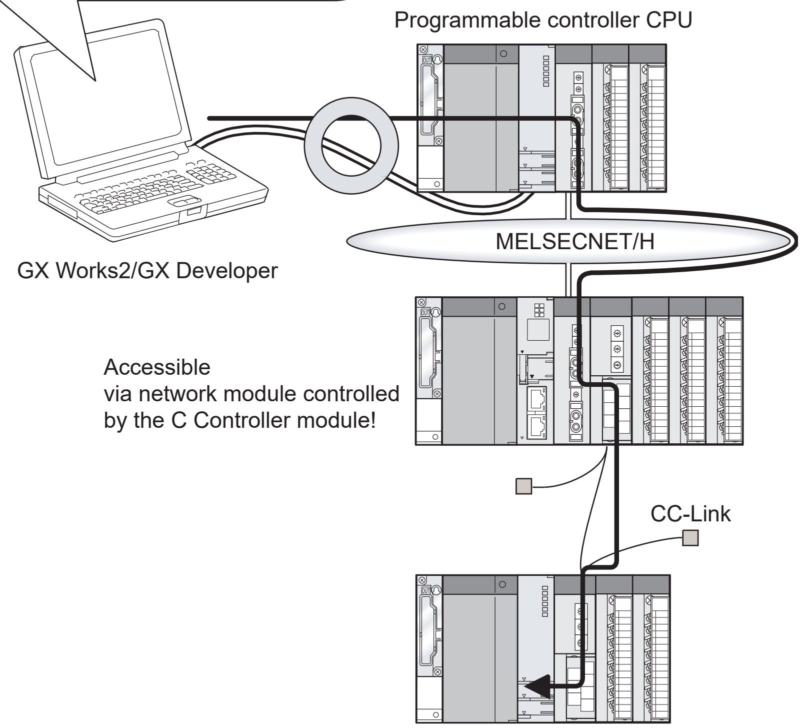

Programmable controller CPU
MELSECNET/H
GX Works2/GX Developer
Accessible
via network module controlled
by the C Controller module!
CC-Link
Programmable controller CPU

Figure 2.16 Other station access via MELSECNET/H and CC-Link

(8) Connection from GX Configurator-QP (Q12DCCPU-V only)

Connection from GX Configurator-QP via the following interfaces of the C Controller module is not allowed.

- RS-232 interface
- 10BASE-T/100BASE-TX interface

Connect the cable to the USB interface.
For connection details, refer to the following.
☞ GX Configurator-QP Operating Manual

(9) Connection from MT Works2 (Q12DCCPU-V only)

Connection from MT Works2 via the RS-232 interface of the C Controller module is not allowed.

Connect the cable to the 10BASE-T/100BASE-TX interface[*1][*2] or to the USB interface.
For details on the connection from MT Works2, refer to the user's manual of the Motion CPU module.

[*1] Connection via the 10BASE-T/100BASE-TX interface is not available with the following C Controller modules.
- Q12DCCPU-V with a serial number whose first five digits are "12041" or earlier

[*2] For connection via the 10BASE-T/100BASE-TX interface, it is necessary to configure the built-in Ethernet port open setting in the <<System Settings>> tab of C Controller setting utility.
For details, refer to the following manual.
☞ C Controller Module User's Manual (Utility Operation, Programming)

(10) Connection from MT Developer (Q12DCCPU-V only)

Connection from MT Developer via the following interfaces of the C Controller module is not allowed.

- RS-232 interface
- 10BASE-T/100BASE-TX interface

Connect the cable to the USB interface.
For details on the connection from MT Developer, refer to the user's manual of the Motion CPU module.

(11) Connection from MX Component (Q12DCCPU-V only)

Connection from MX Component via the RS-232 interface of the C Controller module is not allowed.

Connect the cable to the 10BASE-T/100BASE-TX interface[*1][*2] or to the USB interface.

[*1] Connection via the 10BASE-T/100BASE-TX interface is not available with the following C Controller modules.
- Q12DCCPU-V with a serial number whose first five digits are "12041" or earlier

[*2] For connection via the 10BASE-T/100BASE-TX interface, it is necessary to configure the built-in Ethernet port open setting in the <<System Settings>> tab of C Controller setting utility.
For details, refer to the following manual.
☞ C Controller Module User's Manual (Utility Operation, Programming)

For details on the connection from MX Component, refer to the following.
☞ Page App-37, Appendix 10

(12)Connection to external devices

(a) RS-232 device

The following are precautions for connecting an RS-232 device to the RS-232 interface of the C Controller module.

- A receive error may occur on the target device when the C Controller module or the target device connected to the C Controller module is powered on or off, or reset.
- A receive error occurs on the target device if the target device side system is started during data transmission from the C Controller module to the target device.
- If an error occurred on the target device, take corrective actions according to the instruction manual for the target device.

(b) Ethernet device

The following are precautions for connecting an Ethernet device to the 10BASE-T/100BASE-TX interface of the C Controller module.

1) When the C Controller module was replaced and the IP address was changed

Reset the Ethernet device as well.

(If the Ethernet device retains the Ethernet address (MAC address) of the communication target, ongoing communication may be disabled because the C Controller module replacement and IP address change also changes the Ethernet address (MAC address).)

Likewise, restart the C Controller module if the Ethernet device was replaced.

2) When an error occurred on the Ethernet device side

Take corrective actions according to the instruction manual for the Ethernet device.

3) When VxWorks output an event or a message

VxWorks may output an event or an error message upon detection of a network error when any of the following tools is connected to the C Controller module.

- Workbench Shell or Tornado Shell
- Telnet

For events and messages, take corrective actions according to the following.

- Manual for VxWorks
- Help of Workbench or Tornado

For events and messages that cannot be handled, consult Wind River Systems, Inc.

To contact Wind River Systems, Inc., visit the website below.
www.windriver.com

2.5 Checking Function Version, Serial No., and Software Version

This section explains how to check the function version of the C Controller module and software version of SW3PVC-CCPU.

2.5.1 Checking the function version and serial No. of the C Controller module

The function version and serial No. of the C Controller module can be checked on the rating plate, the serial number display, or the System information screen of SW3PVC-CCPU.

(1) Checking on the rating plate

The rating plate is located on the side face of the module.

(a) For the Q12DCCPU-V

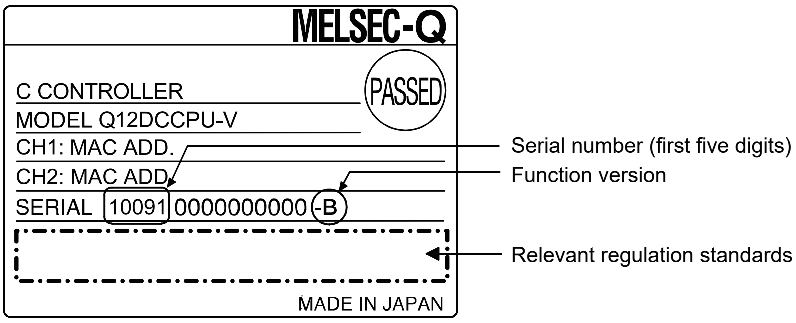


Figure 2.17 Rating plate

(b) For the Q06CCPU-V(-B)

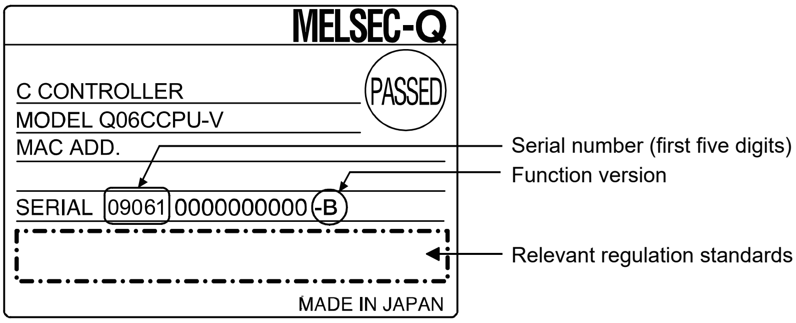

MELSEC-Q
C CONTROLLER
PASSED
MODEL Q06CCPU-V
MAC ADD.
SERIAL 09061 0000000000 -B
MADE IN JAPAN
Serial number (first five digits)
Function version
Relevant regulation standards

Figure 2.18 Rating plate (for the Q06CCPU-V)

(2) Checking on the serial number display

The serial number and function version is printed on the serial number display on the front (at the bottom) of the module.

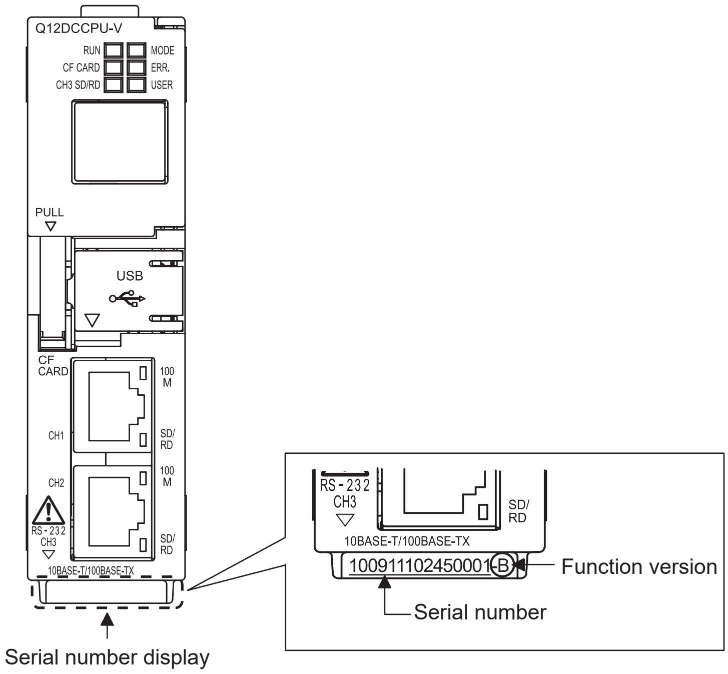


Figure 2.19 Front of the module

Remark

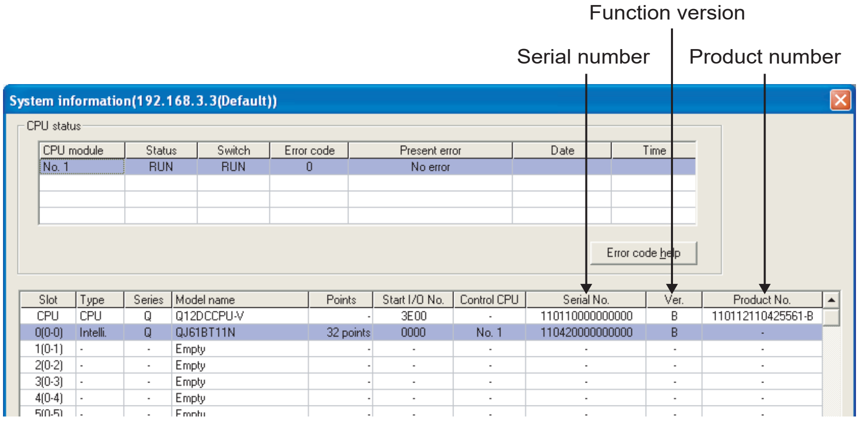
Serial No. printing on the front of the module was started from December in 2008. Note that some of the modules manufactured around the time of change may not have any serial No. printed.

(3) Checking in C Controller setting utility

Check the version in either of the following ways.

(a) System information screen

To display the System information screen, click the System info button on the Module monitoring screen of C Controller setting utility.



CPU status

CPU module	Status	Switch	Error code	Present error	Date	Time
No. 1	RUN	RUN	0	No error		

Slot	Type	Series	Model name	Points	Start I/O No.	Control CPU	Serial No.	Ver.	Product No.
CPU	CPU	Q	Q12DCCPU-V	-	3E00	-	110110000000000	B	110112110425561-B
0(0-0)	Intelli.	Q	QJ61BT11N	32 points	0000	No. 1	110420000000000	B	-
1(0-1)	-	-	Empty	-	-	-	-	-	-
2(0-2)	-	-	Empty	-	-	-	-	-	-
3(0-3)	-	-	Empty	-	-	-	-	-	-
4(0-4)	-	-	Empty	-	-	-	-	-	-
5(0-5)	-	-	Empty	-	-	-	-	-	-

Figure 2.20 System information screen

1) Product number display

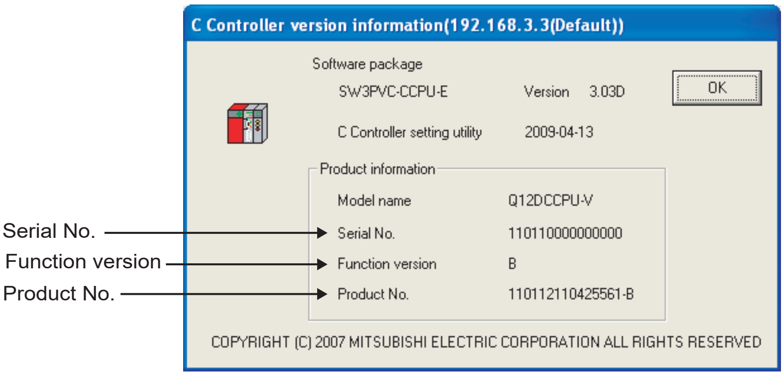
The serial number (product number) printed on the rating plate of the module is displayed in the "Product No." column.[*1]

Note that "-" is displayed for the module that does not support the product number display.

[*1] The Product number fields are active only for the Q12DCCPU-V.

(b) Version information screen

To display the version information screen, select [Version information] from the System menu of C Controller setting utility.



Software package

SW3PVC-CCPU-E Version 3.03D

C Controller setting utility 2009-04-13

Product information

Model name	Q12DCCPU-V
Serial No.	110110000000000
Function version	B
Product No.	110112110425561-B

COPYRIGHT (C) 2007 MITSUBISHI ELECTRIC CORPORATION ALL RIGHTS RESERVED

Figure 2.21 Version information screen

1) Product number display

The serial number (product number) printed on the rating plate of the module is displayed in the "Product No." column.

Note that "-" is displayed for the Q06CCPU-V(-B) because it does not support the product number display.

☒POINT

The serial number displayed on the Version information screen of C Controller setting utility may differ from the ones printed on the rating plate and the front of the module.

- The serial numbers on the rating plate and the front of the module indicate the management information.
- The serial number displayed on the version information screen of C Controller module utility indicates the functional information of the product.
  The functional information of the product will be updated when a function is added.

2.5.2 Checking the software version of software package

To display the version information screen, select [Version information] from the System menu of C Controller setting utility.
(☞ C Controller Module User's Manual (Utility Operation, Programming))

CHAPTER 3 SPECIFICATIONS

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	○

3.1 General Specifications

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	○

The following indicates the general specifications of the C Controller module.

Table 3.1 General specifications

<table>
<tr><th>Item</th><th colspan="6">Specifications *6</th></tr>
<tr><td>Operating ambient temperature</td><td colspan="6">0 to 55℃</td></tr>
<tr><td>Storage ambient temperature</td><td colspan="6">-25 to 75℃ *3</td></tr>
<tr><td>Operating ambient humidity</td><td colspan="6" rowspan="2">5 to 95%RH *4, non-condensing</td></tr>
<tr><td>Storage ambient humidity</td></tr>
<tr><td rowspan="5">Vibration resistance</td><td rowspan="5">Compliant with JIS B 3502 and IEC 61131-2</td><td></td><td>Frequency</td><td>Constant acceleration</td><td>Half amplitude</td><td>Sweep count</td></tr>
<tr><td rowspan="2">Under intermittent vibration</td><td>5 to 8.4Hz</td><td>—</td><td>3.5mm</td><td rowspan="2">10 times each in X, Y and Z directions</td></tr>
<tr><td>8.4 to 150Hz</td><td>9.8m/s^2</td><td>—</td></tr>
<tr><td rowspan="2">Under continuous vibration</td><td>5 to 8.4Hz</td><td>—</td><td>1.75mm</td><td rowspan="2">—</td></tr>
<tr><td>8.4 to 150Hz</td><td>4.9m/s^2</td><td>—</td></tr>
<tr><td>Shock resistance</td><td colspan="6">Compliant with JIS B 3502 and IEC 61131-2 (147m/s^2, 3 times each in 3 directions X, Y, Z)</td></tr>
<tr><td>Operating atmosphere</td><td colspan="6">No corrosive gases</td></tr>
<tr><td>Operating altitude *5</td><td colspan="6">0 to 2000m</td></tr>
<tr><td>Installation location</td><td colspan="6">Inside a control panel</td></tr>
<tr><td>Overvoltage category *1</td><td colspan="6">II or less</td></tr>
<tr><td>Pollution degree *2</td><td colspan="6">2 or less</td></tr>
</table>

* 1 This indicates the section of the power supply to which the equipment is assumed to be connected between the public electrical power distribution network and the machinery within premises.
Category II applies to equipment for which electrical power is supplied from fixed facilities. The surge voltage withstand level for up to the rated voltage of 300V is 2500V.

* 2 This index indicates the degree to which conductive material is generated in terms of the environment in which the equipment is used.
Pollution level 2 is when only non-conductive pollution occurs. A temporary conductivity caused by condensing must be expected occasionally.

* 3 The storage ambient temperature is -20 to 75℃ if the system includes an AnS series module.

* 4 The operating ambient humidity and storage ambient humidity are 10 to 90%RH if the system includes an AnS series module.

* 5 Do not use or store the C Controller system under pressure higher than the atmospheric pressure at sea level.
Doing so can cause malfunction.
For use in a pressurized environment, please consult your local Mitsubishi representative.

* 6 When installing a commercial CompactFlash card into the C Controller module, please follow the specifications of the C Controller module or the CompactFlash card, whichever is lower.

3.2 Performance Specifications

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	○

This section explains the performance specifications of the C Controller module.

Table 3.2 Performance specifications

Item			Specifications		
			Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
Hardware specifications			—		
	Endian format (Memory layout)		Little endian		Big endian
	User file capacity (For user file storage)	Standard RAM	3M bytes	—	
		Standard ROM	—	6M bytes	
		CompactFlash card	Depends on the CompactFlash card used		—
			(Up to 8G bytes)	(Up to 1G bytes)	
	Work RAM (for OS, driver, user program execution)		128M bytes	64M bytes	
	Battery-backed-up RAM		512K bytes[*1]	128K bytes	
	Number of writes to standard ROM		—	Max.100,000 times to the same area	
Software specifications			—		
	Operating system[*2]		VxWorks Version 6.4	VxWorks Version 5.4	
	Programming language		C language (C/C++)		
10BASE-T/100BASE-TX			—		
	Number of channels		2 channels (same specification for CH1 and CH2)	1 channel	
	Interface[*3]		10BASE-T/100BASE-TX		
	Communication method		Full- or half-duplex communication mode		
	Flow control		Full-duplex: Supported (Support to the IEEE802.3x)	Full-duplex: None (Does not support to the IEEE802.3x)	
			Half-duplex: Back pressure congestion control		
	Data transmission speed		10Mbps(10BASE-T)/100Mbps(100BASE-TX)		
	Transmission method		Base band		
	Number of cascaded stages		Up to 4[*4] (10BASE-T)/Up to 2[*4] (100BASE-TX)		
	Maximum segment length (distance between hub and node)		100m (328.08 feet)		
	Connector applicable to external wiring		RJ45		
	Supported function		Auto negotiation function (automatically recognizes 10BASE-T or 100BASE-TX)		
			Auto-MDIX function (automatically recognizes straight or crossing cable)	—	
RS-232			—		
	Interface		Compliant with RS-232		
	Communication method		Full-duplex/half-duplex communication method		
	Synchronization method		asynchronous communication		
	Transmission speed		9600, 14400, 19200, 28800, 38400, 57600, 115200 bps		
	Transmission distance		Up to 15m (49.21 feet)		
	Data format	Start bit	1		
		Data bit	7/8		
		Parity bit	1/None		
		Stop bit	1/2		
	Parity check		Parity check Yes (Even/Odd)/No		
	Sum check code		Sum check Yes/No		
	Transmission control		Flow control (RS/CS control)		
	Recommended cable		7/0.127 □P HRV-SV outside diameter: 8.5mm (0.33 inches) or larger (Oki Electric Cable Company, Limited Specify the number of pairs in □.)		
	Connector applicable to external wiring		Round connector (10-pin)	9-pin D-sub (male) screw type	

(To the next page)

Table 3.2 Performance specifications (continued)

Item		Specifications		
		Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
USB		—		
	Transmission speed	12Mbps (Full Speed Mode: FS)	—	
	Power supply	Self-Powered	—	
	Connector	USB series mini-B connector (Compliant with Universal Serial Bus Specification)	—	
	Other electric characteristics	Compliant with USB Specification Revision 2.0 Use of a hub is not allowed.	—	
CompactFlash card		—		
	Supply power voltage	3.3V±5%		—
	Supply power capacity	Up to 150mA		—
	Card size	TYPE I card TYPE II card is not allowed. I/O cards, such as a modem card are not allowed.		—
	Number of loadable cards	1		—
Number of I/O points (number of points accessible to actual I/O modules)		4096 points (X/Y 0 to FFF)		
Clock function		Year, month, day, hour, minute, second, day of week (automatic leap year detection)		
	Clock accuracy during power ON	Accuracy of VxWorks POSIX Clock Accuracy may vary by the operation program. (☞ Page 4-31, Section 4.7 (5) (a))		
	Clock accuracy during power OFF	Daily error -10.89 to +8.64 seconds (0 to 55℃) Daily error -4.32 to +5.25 seconds (25℃) Before applying power, an additional error of -0.5 to +0.5 seconds may occur.		
Allowable momentary power failure time		Depends on the power supply module		
5V DC internal current consumption		0.97A	0.75A	0.71A
External dimensions		98 (3.86 in.) (H) × 27.4 (1.08 in.) (W) × 115 (4.53 in.) (D)[mm]	98 (3.86 in.) (H) × 27.4 (1.08 in.) (W) × 89.3 (3.52 in.) (D)[mm]	
Weight		0.24kg	0.17kg	

* 1 The number of bytes for the following C Controller modules is 128K bytes.
•Q12DCCPU-V with a serial number whose first five digits are "12041" or earlier

* 2 For the development environment (personal computer), refer to the following manual.
☞ C Controller Module User's Manual (Utility Operation, Programming)

* 3 The C Controller module differentiates 10BASE-T and 100BASE-TX according to the target device.

* 4 This number applies when a repeater hub is used.
When using a switching hub, check the number of cascaded stages with the manufacturer of the hub to be used.

3.3 RS-232 Connector Specifications

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	○

The RS-232 connector specifications are shown below.

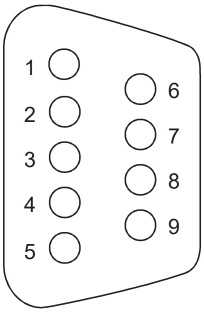
3.3.1 RS-232 connector specifications for the Q12DCCPU-V

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	×	×

(1) Cable connection to the C Controller module through the Q12DCCPU-CBL (RS-232 connector conversion cable)

(a) RS-232 connector specifications

Table 3.3 RS-232 connector specifications



Pin No.	Mnemonic	Signal name	Signal direction Q12DCCPU-CBL ←→ RS-232 device
1	CD(DCD)	Data Carrier Detect	←
2	RD(RXD)	Received Data	←
3	SD(TXD)	Transmitted Data	→
4	ER(DTR)	Data Terminal Ready	→
5	SG	Signal Ground	←→
6	DR(DSR)	Data Set Ready	←
7	RS(RTS)	Request To Send	→
8	CS(CTS)	Clear To Send	←
9	CI(RI)	Ring Indicator	←

(b) Connector shell for connection cable
Use the following connector shell on the Q12DCCPU-CBL side.

- DDK Ltd.
  Plug, shell: 17JE-23090-02(D8A)(-CG)
- Connector fitting screw (M2.6)

3.3.2 RS-232 connector specifications for the Q06CCPU-V(-B)

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
×	○	○

(1) RS-232 connector specifications

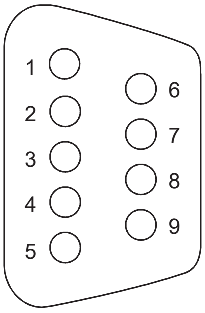


Table 3.4 RS-232 connector specifications

Pin No.	Mnemonic	Signal name	Signal direction
			C Controller module ↔ RS-232 device
1	CD(DCD)	Data Carrier Detect	←
2	RD(RXD)	Received Data	←
3	SD(TXD)	Transmitted Data	→
4	ER(DTR)	Data Terminal Ready	→
5	SG	Signal Ground	↔
6	DR(DSR)	Data Set Ready	←
7	RS(RTS)	Request To Send	→
8	CS(CTS)	Clear To Send	←
9	CI(RI)	Ring Indicator	←

(2) RS-232 interface connector

For the RS-232 interface connector of C Controller module, 9-pin D-sub (female) screw type (fitting screw M2.6) is used.

First six digits of serial number for C Controller module	Manufacturer	Model	Where to contact
"140313" or lower	DDK Ltd.	17L-10090-27(D9AC)-FA	www.ddknet.co.jp/English/index.html
"140314" or higher	OMRON Corporation	XM3B-0922-014	www.omron.com

Use the following connector shell on the C Controller module side.

Manufacturer	Model	Where to contact
DDK Ltd.	17JE-23090-02(D8A)(-CG)	www.ddknet.co.jp/English/index.html

3.4 Operation Processing

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	○

3.4.1 Initial processing

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	○

The initial processing is preprocessing executed upon startup of the C Controller module. When the initial processing is completed, the C Controller module is placed in the status preset by the RUN/STOP/MODE switch.

The following lists the initial processing details performed when the C Controller module is powered on or reset.

Table 3.5 Initial processing list

Initial processing	C Controller module status	
	When powered on	When reset
I/O module initialization	○	○
Parameter check	○	○
Automatic assignment of mounted module I/O numbers	○	○
IP address setting of C Controller module	○	○
CC-Link information setting	○	○
MELSECNET/H information setting	○	○
CC-Link IE controller network information setting	○	○
Intelligent function module switch setting	○	○
Intelligent function module initial value setting	○	○
Script file execution (e.g. starting a user program from the standard RAM, standard ROM, or a CompactFlash card)	○	○

3.4.2 I/O access timing

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	○

This section explains the input (X) and output (Y) data transfer timings of the C Controller module.

(1) Input (X) loading timing

The C Controller module loads input (X) data at one of the following timings.

- When a bus interface function (such as the QBF_X_In_BitEx function) is executed in the user program
- When input (X) data are read out from a peripheral device (such as GOT)

(2) Output (Y) writing timing

The C Controller module writes output (Y) data at one of the following timings.

- When a bus interface function (such as the QBF_Y_Out_BitEx function) is executed in the user program
- When output (Y) data are written from a peripheral device (such as GOT)

Remark

1. For details of the bus interface functions, refer to the following.
   ☞ C Controller User's Manual (Utility Operation, Programming)
2. For processing time of the bus interface functions, refer to the following.
   ☞ Appendix 1

3.4.3 RUN, STOP and PAUSE status operation processing

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	○

The C Controller module has three different operation states: RUN, STOP and PAUSE. This section explains the operation processing of the C Controller module in each operation status.

(1) Operation processing in the RUN status

In the RUN status, output (Y) to each module and buffer memory writing are enabled from the user program in the C Controller module.

(a) Output status when entering the RUN status
When entering the RUN status, the C Controller module may output the output status data saved before entering the STOP status, according to the parameter setting ("Output mode at STOP to RUN").

(2) Operation processing in the STOP status

In the STOP status, output (Y) to each module and buffer memory writing from the user program of the C Controller module are disabled by the RUN/STOP/MODE switch setting or the remote STOP function. (☞ Page 4-6, Section 4.3.1)
The C Controller module is also set in the STOP status when a stop error occurs in it.

(a) Output status when entering the STOP status
When entering the STOP status, the C Controller module saves the output status data and clears all outputs (Y) to OFF.

(3) Operation processing in the PAUSE status

In the PAUSE status, output (Y) to each module and buffer memory writing from the user program of the C Controller module are disabled, with the ON/OFF output data (Y) held by the remote PAUSE function. (☞ Page 4-9, Section 4.3.2)

☒POINT

1. In any of the RUN, STOP, and PAUSE states, output (Y) operation and buffer memory writing are executable from the <<Module monitoring>> tab of C Controller setting utility.
2. In any of the RUN, STOP, and PAUSE states, the C Controller module internally continues user program operation.
   When changing the program processing according to the operation status of the C Controller module, use a bus interface function (QBF_ReadStatusEx) for programming.

3.4.4 Operation processing during momentary power failure

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	○

The C Controller module detects a momentary power failure when the input power voltage supplied to the power supply module falls below the specified range.
On detection of a momentary power failure, the following operation is performed.

(1) When momentary power failure time is within the allowable range

When a momentary power failure occurs, the output status is held and the operation stops.
After power is restored, the error information is registered to the event history file. (The initial time only)

(a) Confirming the number of detections

The C Controller module counts how many time momentary power failure has been detected and holds the data.
The number of momentary power failure detections can be confirmed by the QBF_ReadStatusEx function.

(b) After recovering from a momentary power failure

The operation processing is resumed when the module recovers from a momentary power failure.

(2) When power failure lasts longer than the allowable momentary power failure time

Initial start is needed for the C Controller module.
The same operation processing as the one of the following is performed.

- Powering on the C Controller system
- Resetting by the RESET/SELECT switch
- Remote resetting by C Controller setting utility or QBF_Reset function

CHAPTER 4 FUNCTIONS

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	○

This chapter explains the C Controller module functions.

4.1 Function List

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	○

The following lists the C Controller module functions.

Table 4.1 List of C Controller module functions

Function		Description	Availability			Reference section
			Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B	
I/O module access function		Controls I/O modules or intelligent function modules from the user program of the C Controller module.	○	○	○	Page 4-4, Section 4.2
Intelligent function module access function			○	○	○	
Remote operation function		Controls the execution status of the C Controller module from its user program or development environment.	○	○	○	Page 4-5, Section 4.3
Device function		Creates a device such as a programmable controller CPU in work RAM of the C Controller module.	○	×	×	Page 4-18, Section 4.4
Self-diagnostic function		—				
	Self-diagnostic function	Monitors the operating status of each module, and when an error has occurred, displays error information. For the Q06CCPU-V(-B), the following cannot be detected. • SRAM error • Built-in ROM error	○	△	△	Page 4-20, Section 4.5
	Hardware self-diagnostic function	Perform the specified hardware self-diagnostics in the hardware self-diagnostic mode. For the Q06CCPU-V(-B), the following is not available. • USB diagnostic test • Bus diagnostic test • 7-segment LED test	○	△	△	Page 16-75, Section 16.5
Output (Y) setting for switching from STOP to RUN		Set the output status (Y) for the case where the STOP status is switched to the RUN status.	○	○	○	Page 4-26, Section 4.6
Clock function		Reads clock data in the C Controller module by the user program, and uses them for time control.	○	○	○	Page 4-29, Section 4.7
	Multiple CPU clock synchronization function	Allows clock data synchronization with CPU No.1 when the C Controller module is set as CPU No.2, No.3 or No.4 in a multiple CPU system.	○	×	×	Page 4-33, Section 4.7.1
Input response time selection		Allows selection of the response time for the Q series input modules, I/O combined modules, high-speed input modules, and interrupt modules.	○	○	○	Page 4-34, Section 4.8

○: Available, △: Available but partially restricted, ×: N/A

(To the next page)

Table 4.1 List of C Controller module functions (continued)

Function	Description	Availability			Reference section
		Q12DC CPU-V	Q06C CPU-V	Q06C CPU-V-B	
Error time output mode setting	Set whether to clear or hold the outputs to output modules, I/O combined modules, and intelligent function modules when a stop error occurs in the C Controller module.	○	○	○	Page 4-36, Section 4.9
Hardware error time CPU operating mode setting	Set whether to stop or run the C Controller module when a hardware error occurs in an intelligent function module.	○	○	○	Page 4-37, Section 4.10
Intelligent function module and interrupt module switch setting	Configure the settings for intelligent function modules and interrupt modules. (For details, refer to the manual for each module.)	○	○	○	Page 4-38, Section 4.11
Watchdog timer (WDT)	Detects a hardware error or a user program error of the C Controller module.	○	○	○	Page 4-40, Section 4.12
Interrupt from intelligent function module	Issues an interrupt event (I50 to I255) in response to an interrupt request from an intelligent function module.	○	○	○	Page 4-42, Section 4.13
Connection between C Controller module and GOT (microcomputer connection)	Accesses a GOT through the RS-232 interface of the C Controller module.	○	○	○	Page 4-47, Section 4.14
Telnet function	From the Telnet tool of the development environment (personal computer), allows simple remote debugging (task information display, memory dumping, etc.) of the C Controller. Without using Workbench or Tornado, simple remote debugging is available.	○	○	○	Page 4-48, Section 4.15
Communication function with peripheral devices through an Ethernet port	Connects a peripheral device (such as GX Developer) to the Ethernet port on the C Controller module with an Ethernet cable to perform communications.	○	×	×	Page 4-53, Section 4.16
Unmounting CompactFlash card	CompactFlash card can be unmounted with the RESET/SELECT switch on the C Controller module.	○	○	×	Page 6-24, Section 6.4, Page 6-67, Section 6.8.3
Login user access restriction	Restricts parameter writing from FTP or each utility and use of the Telnet function, by setting (adding/deleting) login users for the C Controller module.	○	○	○	Page 6-88, Section 6.11

○: Available, △: Available but partially restricted, × : N/A

(To the next page)

Table 4.1 List of C Controller module functions (continued)

Function	Description	Availability			Reference section
		Q12DC CPU-V	Q06C CPU-V	Q06C CPU-V-B	
Multiple CPU system	—				
	The C Controller module can build a multiple CPU system in combination with some programmable controller CPUs and Motion CPUs.	○	○	○	Page 10-1, CHAPTER 10 to Page 15-1, CHAPTER 15
Data communications by MELSEC data link functions	Accesses device data of a programmable controller CPU from the C Controller module's user program, which is created with MELSEC data link functions.	○	○	×	Page 13-2, Section 13.1
Event notification	Issues an event to the standby status user program in a C Controller module to resume the user program that was waiting for an interrupt event. The event is issued from either of the following. • Sequence program of programmable controller CPU • User program of C Controller module	○	○	○	Page 13-3, Section 13.2
Multiple CPU synchronous interrupt function	By registering a routine corresponding to a multiple CPU synchronous interrupt by the QBF_EntryMultiCPUSyncInt function, the created program can be executed in synchronization with a Motion CPU (Q172DCPU or Q173DCPU).	○	×	×	Page 13-11, Section 13.2.1
Data communications using CPU shared memory	Transfers data via the CPU shared memory among the C Controller module, programmable controller CPUs, and Motion CPUs.	○	○	○	Page 13-12, Section 13.3
Data communications using multiple CPU high speed transmission area	Transfers data via the multiple CPU high speed transmission area in the CPU shared memory among the C Controller module, programmable controller CPUs, and Motion CPUs.	○	×	×	Page 13-24, Section 13.3.3
Programmable controller remote control function	From the user program of the C Controller module, controls the execution status of a programmable controller CPU.	○	○	×	Page 13-42, Section 13.4
Sequence program control function	From the user program of the C Controller module, controls the execution type of the sequence program.	○	○	○	Page 13-43, Section 13.5
Interrupt issue to Motion CPU	From the user program of the C Controller module, issues an interrupt to a Motion CPU. For the Q172DCPU or Q173DCPU, use the Q12DCCPU-V C Controller module.	○	△	△	Page 13-44, Section 13.6
Motion CPU control instruction	From the user program of the C Controller module, start a Motion CPU's SFC program or servo program or changes the set or present values of the servo. For the Q172DCPU or Q173DCPU, use the Q12DCCPU-V C Controller module.	○	△	△	Page 13-46, Section 13.7
Motion CPU device access	From the user program of the C Controller module, reads data from or writes data to Motion CPU's devices. For the Q172DCPU or Q173DCPU, use the Q12DCCPU-V C Controller module.	○	△	△	Page 13-47, Section 13.8

○: Available, △: Available but partially restricted, ×: N/A

4.2 I/O Module and Intelligent Function Module Access Function

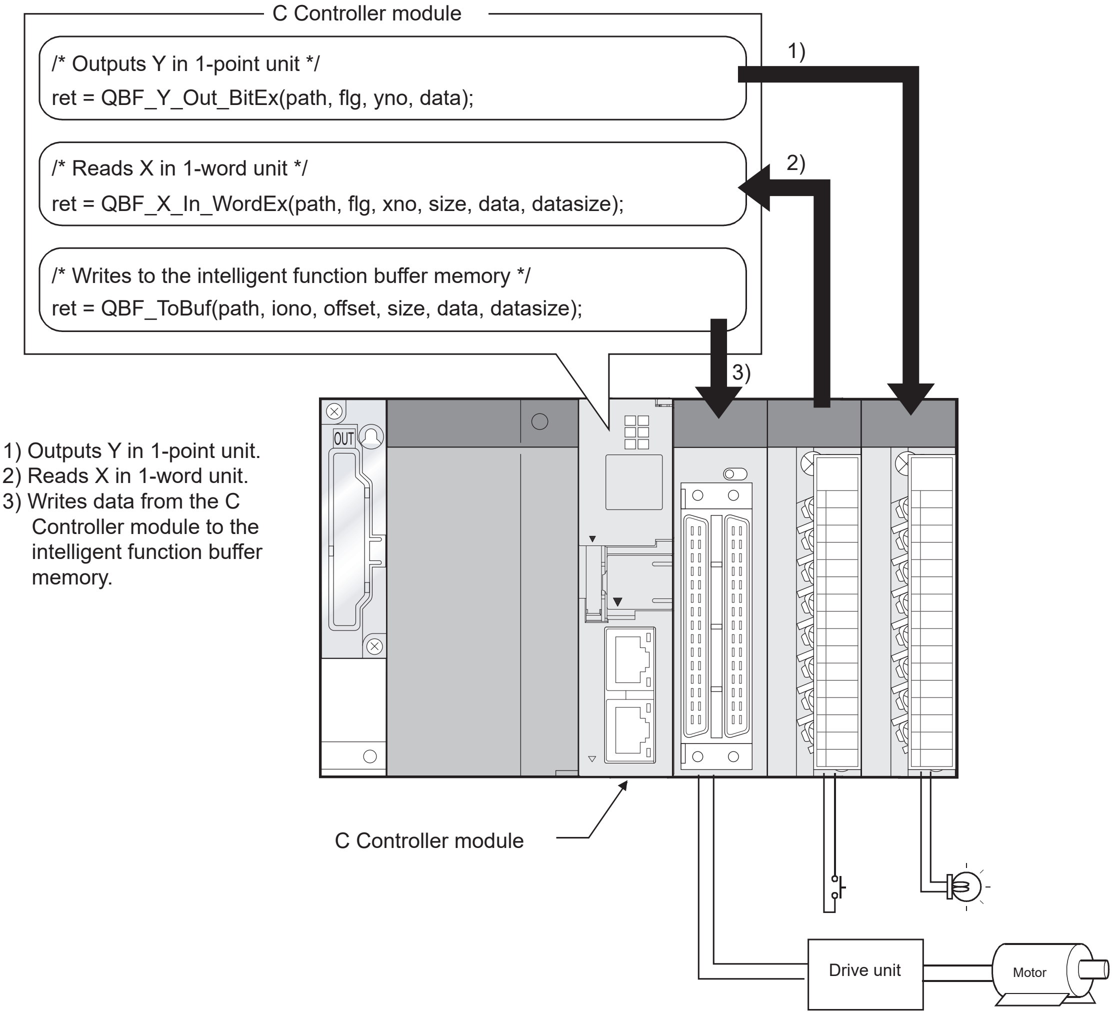
Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	○

(1) About this function

This function controls the I/O modules and intelligent function modules that are controlled by the C Controller module.

Create the user program of the C Controller module using the bus interface functions.

<Example>



C Controller module

```
/* Outputs Y in 1-point unit */
ret = QBF_Y_Out_BitEx(path, flg, yno, data);
```

```
/* Reads X in 1-word unit */
ret = QBF_X_In_WordEx(path, flg, xno, size, data, datasize);
```

```
/* Writes to the intelligent function buffer memory */
ret = QBF_ToBuf(path, iono, offset, size, data, datasize);
```

1) Outputs Y in 1-point unit.
2) Reads X in 1-word unit.
3) Writes data from the C Controller module to the intelligent function buffer memory.

Figure 4.1 Access to I/O modules and an intelligent function module

Remark

For the bus interface functions, refer to the following manual.

☞ C Controller Module User's Manual (Utility Operation, Programming)

4.3 Remote Operation Function

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	○

The remote operation function controls the operation status of the C Controller module or a programmable controller CPU from the user program or C Controller setting utility of the development environment (personal computer).
Create the user program of the C Controller module using bus interface functions.

There are the following three different remote operations.

- Remote RUN/STOP :☞ Page 4-6, Section 4.3.1
- Remote PAUSE :☞ Page 4-9, Section 4.3.2
- Remote RESET :☞ Page 4-11, Section 4.3.3

Remark

Remote RESET is executable for C Controller modules and CPU No.1 (programmable controller CPU).

4.3.1 Remote RUN/STOP

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	○

For the remote RUN/STOP operation from a programmable controller CPU, refer to the manual for the programmable controller CPU.

(1) What is remote RUN/STOP?

The remote RUN/STOP is a control from the user program or development environment (personal computer), by which the C Controller module can run or stop with its RUN/STOP/MODE switch set to RUN.

(2) Application of the remote RUN/STOP

The remote RUN/STOP is useful when:

- The C Controller module is out of reach.
- External signals are used to run or stop the C Controller module inside a control panel.

(3) Operation in the remote RUN/STOP

The remote RUN/STOP makes the C Controller module behave as described below.

(a) Remote STOP

As soon as remote STOP is requested, the C Controller module is stopped, and the output (Y) to each module and buffer memory writing from the user program are disabled.

These operations are executable, however, from <<Module monitoring>> of C Controller setting utility.

(b) Remote RUN

Executing remote RUN in the remote STOP status will change the C Controller module status to RUN again, enabling the output (Y) to each module and buffer memory writing from the user program.

(4) Methods for remote RUN/STOP

The remote RUN/STOP can be executed by either of the following two methods: "Using user program" and "Using development environment (personal computer).

(a) Using user program

Execute a bus interface function (QBF_Control or QBF_ControlEx) from the user program to perform remote RUN/STOP.

Remark

For the bus interface functions, refer to the following manual.

☞ C Controller Module User's Manual (Utility Operation, Programming)

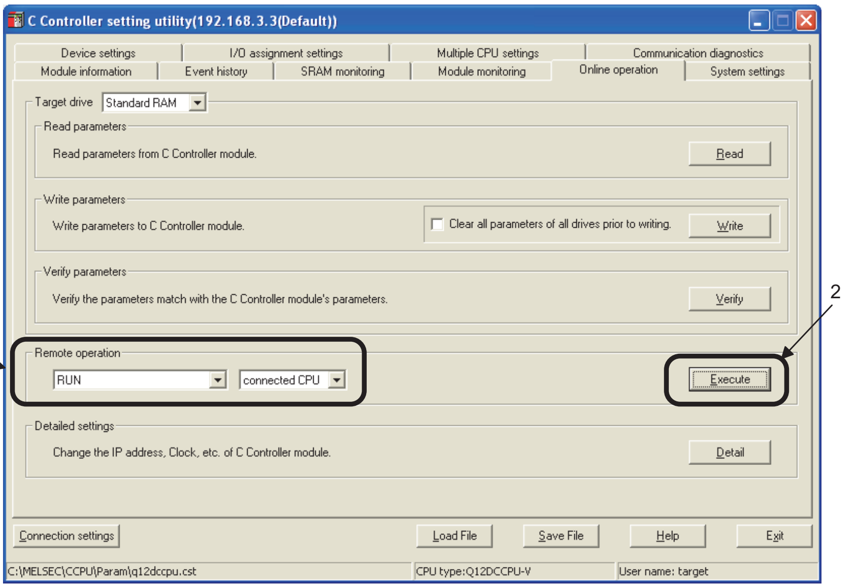
(b) Using development environment (personal computer)

In <<Online operation>> of C Controller setting utility, perform as described below.

1) Set "Remote operation" to "RUN" or "STOP" and select a C Controller module or programmable controller CPU.

2) Click Execute to execute remote RUN/STOP.

1) Set "RUN" or "STOP" and select a target C Controller module or programmable controller CPU.



Target drive: Standard RAM

Read parameters: Read parameters from C Controller module.

Write parameters: Write parameters to C Controller module. □ Clear all parameters of all drives prior to writing.

Verify parameters: Verify the parameters match with the C Controller module's parameters.

Remote operation: RUN, connected CPU

Detailed settings: Change the IP address, Clock, etc. of C Controller module.

C:\MELSEC\CCPU\Param\q12dccpu.cst | CPU type:Q12DCCPU-V | User name: target

2) Click Execute button.

Figure 4.2 Remote RUN/STOP from development environment (personal computer)

(5) Precautions

Pay attention to the following since the C Controller module gives priority to STOP.

(a) STOP timing

The C Controller module is stopped when remote STOP is executed from either of the user program or C Controller setting utility.

(b) Changing the remote STOP status to RUN again

To change the C Controller module status from remote STOP to RUN again, execute remote RUN.

(☞ (4) in this section)

(c) Output (Y) status during STOP

When "Output mode at STOP to RUN" is set to "Previous state" in <<System settings>> of C Controller setting utility, the output (Y) data at the time when RUN is changed to STOP are output in remote RUN mode.

(d) User program during STOP

The user program will not stop even if the C Controller module is placed in the remote STOP status. (Operation continues inside the C Controller module.)
To change the program processing according to the operation status of the C Controller module, utilize bus interface functions (QBF_ReadStatusEx) in programming.

Remark

The operation status of the C Controller module can be confirmed by the following.

- Bus interface function (QBF_ReadStatusEx)
- C Controller setting utility ("CPU operation status" on <<Module information>>)

4.3.2 Remote PAUSE

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	○

For the remote PAUSE from a programmable controller CPU, refer to the manual for the programmable controller CPU.

(1) What is remote PAUSE?

The remote PAUSE is a control from the user program or development environment (personal computer), by which the C Controller module can be placed in the PAUSE status with its RUN/STOP/MODE switch set to RUN.

(a) What is the PAUSE status?

In the PAUSE status, the output (Y) to each module and buffer memory writing from the C Controller module's user program are disabled while the ON/OFF states of all outputs (Y) are held.

However, outputs (Y) from <<Module monitoring>> of C Controller setting utility and buffer memory writing are executable.

(2) Application of the remote PAUSE

The remote PAUSE is useful in process control, for example, to keep the ON status of output (Y) set in the RUN status even if the C Controller module status is changed from RUN to STOP.

(3) Methods for remote PAUSE

The remote PAUSE can be executed by either of the following two methods: "Using user program" and "Using development environment (personal computer).

(a) Using user program

Execute a bus interface function (QBF_Control or QBF_ControlEx) from the user program to perform remote PAUSE.

Remark

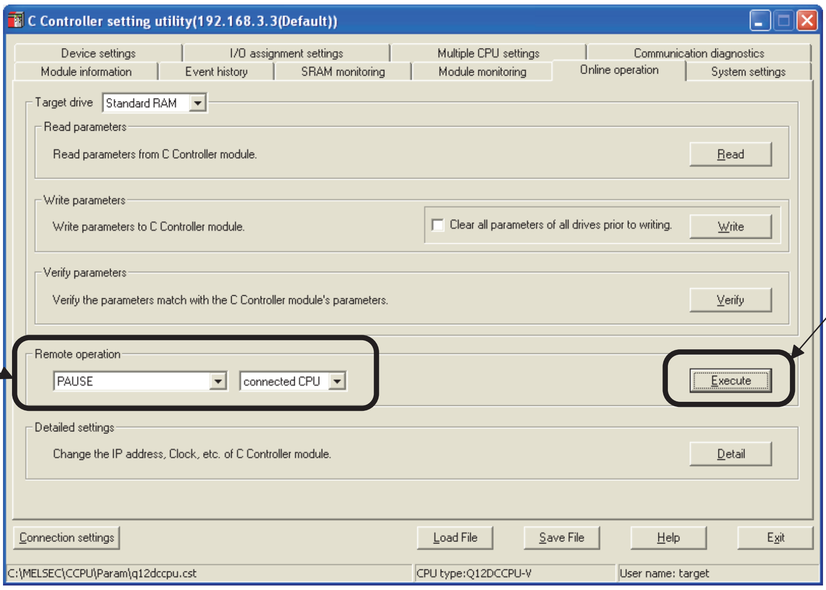
For the bus interface functions, refer to the following manual.

☞ C Controller Module User's Manual (Utility Operation, Programming)

(b) Using development environment (personal computer)
In <<Online operation>> of C Controller setting utility, perform as described below.

1) Set "Remote operation" to "PAUSE" and select a C Controller module or programmable controller CPU.

2) Click Execute to execute remote PAUSE.



C Controller setting utility(192.168.3.3(Default))

Target drive: Standard RAM

Read parameters
Read parameters from C Controller module.

Write parameters
Write parameters to C Controller module.
□ Clear all parameters of all drives prior to writing.

Verify parameters
Verify the parameters match with the C Controller module's parameters.

Remote operation: PAUSE, connected CPU

Detailed settings
Change the IP address, Clock, etc. of C Controller module.

C:\MELSEC\CCPU\Param\q12dccpu.cst | CPU type:Q12DCCPU-V | User name: target

1) Set "PAUSE" and select a C Controller module or programmable controller CPU.

2) Click Execute button.

Figure 4.3 Remote PAUSE from development environment (personal computer)

(4) Precautions

The user program will not stop even if the C Controller module is placed in the remote PAUSE status. (Operation continues inside the C Controller module.)
To change the program processing according to the operation status of the C Controller module, utilize bus interface functions (QBF_ReadStatusEx) in programming.

Remark

The operation status of the C Controller module can be confirmed by the following.

- Bus interface function (QBF_ReadStatusEx)
- C Controller setting utility ("CPU operation status" on <<Module information>>)

4.3.3 Remote RESET

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	○

(1) What is remote RESET?

The remote RESET is a control from the user program or development environment (personal computer), by which the C Controller module or programmable controller CPU can be reset when it is in the STOP status.

(2) Applicable system configurations and methods

(a) Single CPU system configuration

The remote RESET can be executed by using the user program or development environment (personal computer).

(b) Multiple CPU system configuration

The following table shows the availability of the remote RESET in the multiple CPU system configuration.

Table 4.2 Availability of remote RESET in multiple CPU system configuration

CPU No.1	Another CPU	Remote RESET of CPU No.1 by CPU No.1		Remote RESET of CPU No.1 by another CPU	
		By user program	From development environment (personal computer)	By user program	From development environment (personal computer)
C Controller module	C Controller module	○	○	○	○
Programmable controller CPU	C Controller module	For remote RESET of the programmable controller CPU, refer to the manual for the programmable controller CPU used.		○[*1]	○[*1]

○: Available ×: N/A

[*1] Remote RESET cannot be executed from the C Controller module when remote STOP has been executed from another peripheral (GX Developer, etc.)
Cancel the remote STOP from the peripheral.
For the remote operations of the programmable controller CPU, refer to the following manual.
☞ QCPU User's Manual (Function Explanation, Program Fundamentals)

(3) Application of the remote RESET

The C Controller module or programmable controller CPU can be remotely reset when an error occurs in the place where direct control of the RUN/STOP/MODE, RUN/STOP, or RUN/STOP/RESET switch on the CPU module is not available.

(4) Remote RESET timing

The remote RESET is executable in the STOP status only. When the C Controller module or programmable controller CPU is in the RUN status, change the status to STOP by remote STOP.

Even if the RUN/STOP/MODE, RUN/STOP, or RUN/STOP/RESET switch of the C Controller module or programmable controller CPU is set to RUN, it can be reset when it is stopped due to an error that can be detected by the self-diagnostic function.

(5) Confirmation before remote RESET

(a) When the C Controller module is used in a single CPU system, or is CPU No.1 in a multiple CPU system

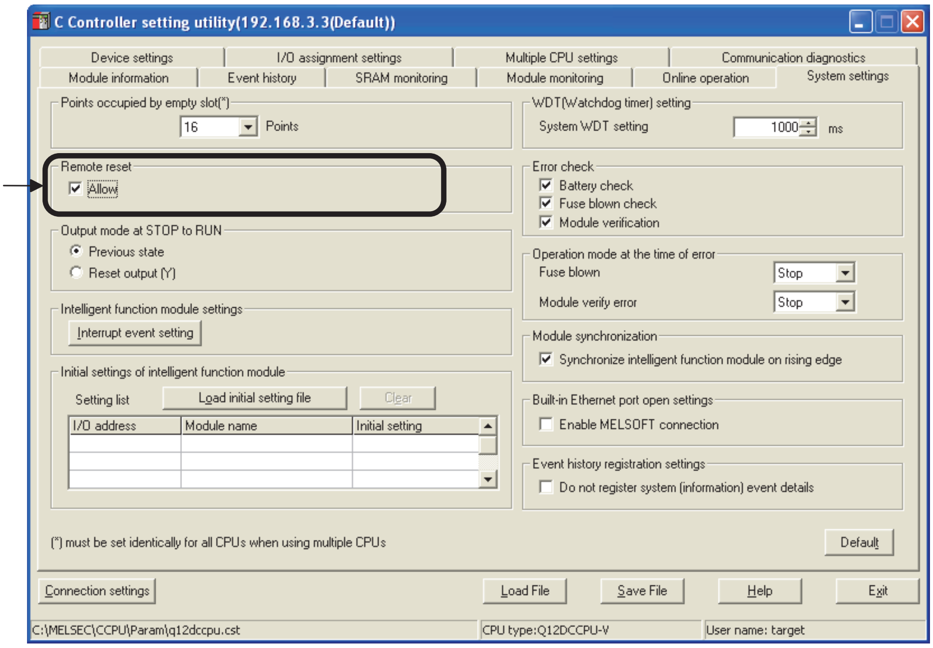
Check the following conditions 1) and 2) before executing remote RESET.

Remote RESET is not available if either of 1) and 2) is not satisfied.

1) Setting required for remote RESET

In <<System settings>> of C Controller setting utility, the "Allow" box must be checked for "Remote reset", and parameters must have already been written to the C Controller module. (Write the parameters and then reset the C Controller module.)

Set to "Allow".



C Controller setting utility(192.168.3.3(Default))

Points occupied by empty slot(*): 16 Points

Remote reset
☑ Allow

Output mode at STOP to RUN
● Previous state
○ Reset output (Y)

Intelligent function module settings

Initial settings of intelligent function module
Setting list

I/O address	Module name	Initial setting

WDT(Watchdog timer) setting
System WDT setting: 1000 ms

Error check
☑ Battery check
☑ Fuse blown check
☑ Module verification

Operation mode at the time of error
Fuse blown: Stop
Module verify error: Stop

Module synchronization
☑ Synchronize intelligent function module on rising edge

Built-in Ethernet port open settings
□ Enable MELSOFT connection

Event history registration settings
□ Do not register system (information) event details

(*) must be set identically for all CPUs when using multiple CPUs

C:\MELSEC\CCPU\Param\q12dccpu.cst | CPU type:Q12DCCPU-V | User name: target

Figure 4.4 Setting remote reset to Allow

2) C Controller module status

The C Controller module should be in the STOP status.

(b) When a programmable controller CPU is CPU No.1 in a multiple CPU system
Check the following conditions 1) and 2) before executing remote RESET. Remote RESET is not available if either of 1) and 2) is not satisfied.

1) Setting required for remote RESET
For CPU No.1, set "Allow" for "Remote reset" in GX Works2/GX Developer, and parameters must have already been written to CPU No.1. (Write the parameters and then reset CPU No.1.)

2) Status of CPU No.1
CPU No.1 should be in the STOP status by the following.
- Setting the RUN/STOP or RUN/STOP/RESET switch to STOP
- Executing remote STOP from the user program (QBF_ControlEx or mdControl function) (MELSEC data link functions are available for the Q06CCPU-V only.)
- Executing remote STOP from the C Controller setting utility

☒POINT

Remote RESET cannot be executed for CPU No.1 from the C Controller module when remote STOP has been executed for it from another peripheral (GX Developer, etc.)
Cancel the remote STOP from the peripheral.
For remote operations of the programmable controller CPU, refer to the following manual.
☞ QCPU User's Manual (Function Explanation, Program Fundamentals)

(6) Methods for remote RESET

The remote RESET can be executed by either of the following two methods: "Using user program" and "Using development environment (personal computer).

(a) Using user program

Execute a bus interface function (QBF_Reset) from the user program to perform remote RESET.

Remark

For the bus interface function, refer to the following manual.

☞ C Controller Module User's Manual (Utility Operation, Programming)

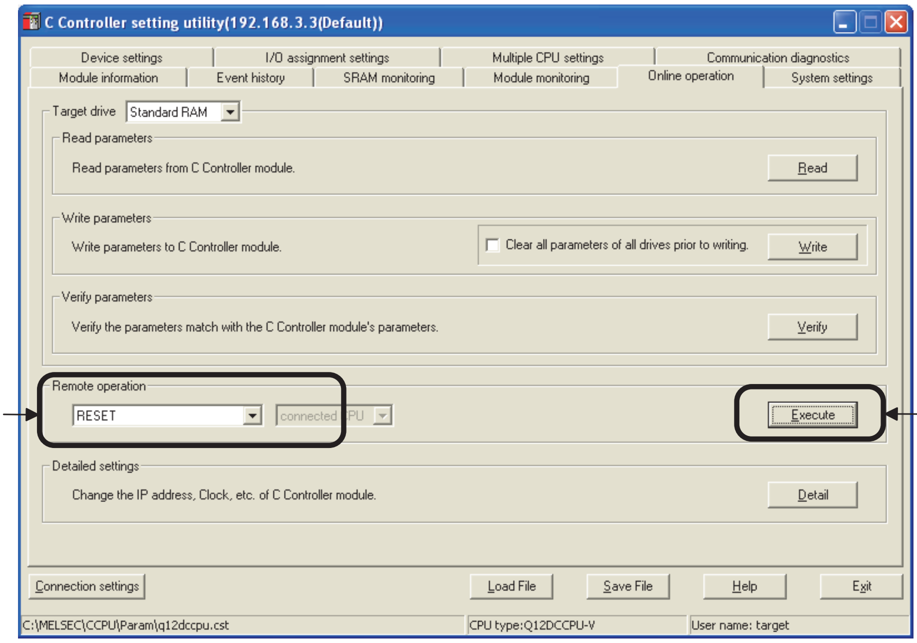
(b) Using development environment (personal computer)

In <<Online operation>> of C Controller setting utility, perform as described below.

1) When files are opened in a user program on the C Controller module, close all programs.

2) Set "Remote operation" to "RESET".

3) Click [Execute] to execute remote RESET.



C Controller setting utility(192.168.3.3(Default))

Target drive: Standard RAM

Read parameters: Read parameters from C Controller module.

Write parameters: Write parameters to C Controller module.

□ Clear all parameters of all drives prior to writing.

Verify parameters: Verify the parameters match with the C Controller module's parameters.

Remote operation: RESET

Detailed settings: Change the IP address, Clock, etc. of C Controller module.

C:\MELSEC\CCPU\Param\q12dccpu.cst | CPU type:Q12DCCPU-V | User name: target

1) Set to "RESET".

2) Click [Execute] button.

Figure 4.5 Remote RESET from development environment (personal computer)

(7) Precautions

(a) Setting for remote RESET
Check the conditions given in (5) in this section before performing remote RESET. If the conditions in (5) in this section are not satisfied, remote RESET is not allowed.

(b) Before performing remote RESET
Close the respective user programs in the C Controller module before performing remote RESET.
Execution of remote RESET during user program operation may damage the user program and data files. (☞ Page 8-9, Section 8.2.2)

(c) Status after completion of resetting
Execution of remote RESET changes the operation status of the target C Controller module or programmable controller CPU to the status set with the RUN/STOP/MODE, RUN/STOP, or RUN/STOP/RESET switch.

1) When the RUN/STOP/MODE, RUN/STOP, or RUN/STOP/RESET switch is set to STOP
Upon completion of remote RESET, the operation status of the target C Controller module or programmable controller CPU changes to STOP.

2) When the RUN/STOP/MODE, RUN/STOP, or RUN/STOP/RESET switch is set to RUN
Upon completion of remote RESET, the operation status of the target C Controller module or programmable controller CPU changes to RUN.

(d) Remote RESET during stop due to an error
If remote RESET is executed when the C Controller module in a single CPU system or CPU No.1 in a multiple CPU system has stopped due to an error, upon completion of the reset, the module status will be changed to the one set by the RUN/STOP/MODE, RUN/STOP, or RUN/STOP/RESET switch. Therefore, special attention must be paid.

(e) Restarting after remote RESET
After remote RESET, the OS and system software of the C Controller module will be restarted.
Note the following cases.

1) When using a bus interface function (QBF_Reset)
There is no return value of the bus interface function (QBF_Reset).

2) When using development environment (personal computer)
Remote RESET will cuts off communication between the development environment (personal computer) and C Controller module.
Reconnect the development environment (personal computer) to the C Controller module after remote RESET.

4.3.4 Relation between remote operation and RUN/STOP status

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	○

This section explains the relation between the remote operations and the RUN/STOP/MODE switch setting of the C Controller module.

(1) Relation between remote operation and RUN/STOP status of C Controller module

The following table indicates the operation status that varies by combination of the remote operation and the C Controller module's RUN/STOP status.

Table 4.3 Relation between RUN/STOP status and remote operation

RUN/STOP/MODE switch	Remote operation			
	RUN	STOP	PAUSE	RESET *1
RUN	RUN	STOP	PAUSE	*2
STOP	STOP	STOP	STOP	RESET *3

* 1 In <<System settings>> of C Controller setting utility, "Remote reset" should be set to "Allow".

* 2 Reset is available when the C Controller module with the RUN/STOP/MODE switch set to RUN has been placed in the STOP status by remote STOP.

* 3 Includes the case where the C Controller module is stopped due to an error.

(2) Remote operations from the same development environment (personal computer) or user program

When more than one remote operation are attempted from the same development environment (personal computer) or the same user program, the status resulted from the last one is effective.

(3) Remote operations from multiple development environments (personal computers) or user programs

For the C Controller module to which remote operation has already been executed, another remote operation can be executed from a different development environment (personal computer) or user program.

(4) Remote operation from the C Controller module to CPU No.1 (programmable controller CPU)

While remote operation is performed from another peripheral (GX Developer, etc.) to CPU No.1, remote operation cannot be executed from the C Controller module.
Execute remote RUN from the peripheral executing the remote operation to cancel the operation.
For example, while another peripheral is executing remote PAUSE, even if a remote operation is executed from the C Controller module, the PAUSE status of CPU No.1 will not change.
In this case, executing remote RUN from the peripheral will cancel the remote PAUSE, enabling a remote operation from the C Controller module.

☒POINT

Remote operation from the C Controller module to CPU No.1 (programmable controller CPU) can be performed by the following methods.

- Remote RESET from C Controller setting utility
- Remote RESET from a bus interface function (QBF_Reset)
- Remote RUN/STOP/PAUSE from a MELSEC data link function (mdControl)
  For the Q06CCPU-V-B, MELSEC data link functions are not available.

4.4 Device Function

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	×	×

(1) Device function

Creates a device such as a programmable controller CPU in work RAM on the C Controller module.

Device function cannot be used for the following C Controller modules.

- Q12DCCPU-V with a serial number whose first five digits are "12041" or earlier

(2) Device list

The following shows devices that can be created.

Table 4.4 Device list

Type	Device name	Default value			Utility-set range
		Number of points	Range		
Bit device	Internal relay	8192 points	M0 to 8191	Decimal	0 to 60K
Word device	Data register	12288 points	D0 to 12287	Decimal	0 to 4086K

(3) Settings for using device functions

Device settings require to set in the <<Device settings>> tab of the C Controller setting utility to use device functions.
For the C Controller setting utility, refer to the following.

☞ C Controller Module User's Manual (Utility Operation, Programming)

POINT

When setting the number of device points, set it in units of 16 points.

(4) Precautions

(a) Number of device points

Set only the required number of device points.

Because a device is created in work RAM, the available range of work RAM, such as in a user program, decreases due to an extra setting.

The consumption of work RAM is calculated by the following formulas.

- Internal relay (M)

  16 points is calculated as one word.

(Consumption of work RAM in the internal relay (M)) = (Number of points assigned to the internal relay (M))/16 [word]

- Data register (D)

  1 point is calculated as one word.

(Consumption of work RAM in the data register (D)) = (Number of points assigned to the data register (D)) [word]

(b) The intelligent function module dedicated instructions and link dedicated instructions

The following intelligent function module dedicated instructions and link dedicated instructions cannot access to device of the C Controller module.

- JP/GP.READ
- JP/GP.SREAD
- JP/GP.WRITE
- JP/GP.SWRITE

4.5 Self-Diagnostic Function

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	△	△

(1) What is the self-diagnostic function?

The self-diagnostic function allows the C Controller module to detect its own error during operation, preventing a malfunction and providing preventive maintenance.

(2) Processing after error detection

When the C Controller module is started in normal operation mode and if an error is detected by the self-diagnostic function, the C Controller module behaves as described below.

(a) Turning on or flashing the ERR. LED
The ERR. LED on the front side of the C Controller module turns on or flashes.

(b) Flashing the 7-segment LED (Q12DCCPU-V only)
The 7-segment LED on the front of the C Controller module flashes, indicating "00", "01" or "02".

(c) Storing error data
An error code is stored and error information is changed.
The error code and error information can be confirmed in "Error code" and "Error information" in <<Module information>> of C Controller setting utility.

(d) Disabling writes and clearing outputs
Upon detection of an error, outputs (Y) from the user program and writing to the buffer memory is disabled, and outputs to output modules, I/O modules, and intelligent function modules are cleared.

In C Controller setting utility, users can select "Stop" or "Continue" for the output (Y) from the user program and writing to the buffer memory, "Clear" or "Hold" for each output, and "All stop" or "Continue" for a multiple CPU system, as described below.

1) For an error detected in an intelligent function module
Whether to stop or continue the output (Y) from the user program and writing to the buffer memory can be selected for the intelligent function module.
Set it in "H/W error time CPU operation mode" in <<I/O assignment settings>> of C Controller setting utility.

2) Outputs to output modules, I/O modules, and intelligent function modules in the case of error detection
Whether to clear or hold the output to each of output modules, I/O modules, and intelligent function modules in the case of error detection can be selected.
Set it in "Error time output mode" in <<I/O assignment settings>> of C Controller setting utility.

3) Fuse blown error or module verify error
For the above errors, whether to stop or continue the output (Y) from the user program and writing to the buffer memory can be selected.
Set it in "Operation mode at the time of error" in <<System settings>> of C Controller setting utility.

4) When an error occurs in CPU No.2, No.3, or No.4
Whether to stop all of the multiple CPU system or not can be selected for the case where a stop error occurs in any of CPU No.2 to No.4 in the multiple CPU system.
Set it in "Operation mode" in <<Multiple CPU settings>> of C Controller setting utility.

(e) Registering error data to the event history file
Historical data of detected errors are registered to the event history file.
The event history file can be viewed in <<Event history>> of C Controller setting utility.

(3) Selectable error detection

If any of the following items under "Error check" is not selected in <<System settings>> of C Controller setting utility, the corresponding error detection is not performed.

1) Battery check

2) Fuse blown check

3) Module verification

(4) Self-diagnostic function list

Table 4.6 lists the self-diagnostic functions available during normal operation.
The numbers under the "Error handling" column heading represent the following:

Table 4.5 Numbers under "Error handling"and corresponding descriptions

No.	Description
1)	Error details can be confirmed in C Controller setting utility. (☞ C Controller Module User's Manual (Utility Operation, Programming)
2)	Output (ON/OFF) to an output module, I/O module, or intelligent function module from the user program is disabled. In "Error time output mode" in <<I/O assignment settings>> of C Controller setting utility, whether to clear or hold the output status of each module can be selected for the case where a stop error occurs in the C Controller module.
3)	Registration to event history file Only when an error is detected and any other error of the same kind does not exist, it is registered to the event history file.

Table 4.6 Self-diagnostic function list

Item	Description	C Controller module status	LED status		Diagnostic timing	Error handling		
			RUN	ERR.		1)	2)	3)
System watchdog timer error detection	A system watchdog timer error is detected.	Stop	Off	Flash	Any time	○	○	○
User watchdog timer error detection	A user watchdog timer error is detected.	Stop	Off	Flash	Any time	○	○	○
Battery error detection*1	A battery error of the C Controller module is detected.	Continue	On	On	Periodic check	○	—	○
Momentary stop detection*2	A momentary stop is detected.	Continue	On	Off	Any time	○	—	○
Module verify error detection*1	The mounting condition of an I/O module is diagnosed.	Stop/ Continue*3	Off/On	Flash/ On	Periodic check	○	○	○
Blown fuse detection*1	The condition of the fuse of an output module is diagnosed.	Stop/ Continue*3	Off/On	Flash/ On	Periodic check	○	○	○
Control bus error detection	The condition of the bus on the base unit is diagnosed.	Stop	Off	Flash	At power-on, at reset, periodic check, or at module access	○	○	○
Intelligent function module error detection	An error in an intelligent function module is detected.	Stop/ Continue*4	Off/On	Flash/ On	At power-on, at reset, periodic check, at module access, or when interrupt occurs	○	○	○
Intelligent function module assignment error detection	The I/O assignment status of an intelligent function module is diagnosed.	Stop	Off	Flash	At power-on or at reset	○	○	○
Parameter error detection	Parameters in the system setting, multiple CPU setting, etc. are diagnosed.	Stop	Off	Flash	At power-on or at reset	○	○	○
Intelligent function module parameter error detection	Parameters for intelligent function modules are diagnosed.	Stop	Off	Flash	At power-on or at reset	○	○	○
Link parameter error detection	Parameters for link modules (CC-Link modules, MELSECNET/H modules, CC-Link IE controller network modules) are diagnosed.	Stop	Off	Flash	At power-on or at reset	○	○	○
Multiple CPU error detection	In a multiple CPU system configuration, an error of another CPU or inconsistency in the multiple CPU system is detected.	Stop/ Continue*5	Off/On	Flash/ On	Any time At power-on or at reset	○	○	○
Main CPU error detection*2	Failure of the power supply module or C Controller module is detected.	Stop	Off	Flash	Any time At power-on or at reset	○	○	○
Power OFF detection*2	AC/DC power failure is detected.	Continue	On	Off	Any time	—	—	○

○: Detectable —: Not detectable

* 1 "Not checked" can be set in "Error check" in <<System settings>> of C Controller setting utility.

* 2 How many times a momentary stop was detected can be confirmed by the QBF_ReadStatusEx function.
While the C Controller module is detecting "Main CPU error", "Momentary stop detection" (including detection count update) and "Power OFF detection" are not performed.

* 3 The CPU operation status for error detection can be changed to "Continue" in "Operation mode at the time of error" in <<System settings>> of C Controller setting utility.

* 4 The CPU operation status for error detection can be changed to "Continue" in "H/W error time CPU operation mode" in <<I/O assignment settings>> of C Controller setting utility.

* 5 Whether to stop the multiple CPU system or not when an error occurs in CPU No.2, No.3, or No.4 can be set in "Operation mode" in <<Multiple CPU settings>> of C Controller setting utility.

(To the next page)

Table 4.6 Self-diagnostic function list (continued)

Item	Description	C Controller module status	LED status		Diagnostic timing	Error handling		
			RUN	ERR.		1)	2)	3)
SRAM error detection	An error of the battery-backed-up RAM (SRAM) is detected. (The standard RAM ("/RAM") is included.) (The Q12DCCPU-V only)	Stop	Off	Flash	Any time At power-on[*6], at reset[*6]	○	○	○
Built-in ROM error detection[*7]	An error of the built-in ROM ("/SYSTEMROM") is detected. (The Q12DCCPU-V only)	Stop	Off	Flash	At power-on, at reset	—	—	—
Other error detections	Stop/continue errors other than the above are detected.	Continue	On	On	At power-on, at reset, periodic check	○	*8 ○	○

○: Detectable —: Not detectable

* 6 The 7-segment LED flashes, indicating "01" or "02".
If this error has occurred, the C Controller module will not start.

* 7 The 7-segment LED flashes, indicating "00."
If this error has occurred, the C Controller module will not start.

* 8 Output is stopped only when a stop error occurs.
For a continuation error, output is not stopped.

Remark

For the C Controller setting utility, refer to the following.

☞ C Controller Module User's Manual (Utility Operation, Programming)

(5) Clearing errors

A C Controller module can clear only continuation errors.

The errors cannot be cleared with the following C Controller modules.

- Q12DCCPU-V with a serial number whose first five digits are "12041" or higher
- Q06CCPU-V(-B)

(a) Error clear procedure

To clear errors, complete the following procedure.

1) Eliminate the reason for the error.

2) Perform one of the following operations or processes.

- Click the Error clear button on the <<Module information>> tab of the C Controller setting utility.
- Execute the QBF_ClearError function.

3) The target error will be cleared.

(b) The status after clearing errors

- When the C Controller module is restored with error clear, the special relay, special register and LED related to the error return to the status before the error occurred.
- When the same error occurs after error clear is performed, the error is registered to the event history again.

POINT

Errors (*1) occurring as a result of cause other than the C Controller module cannot be cleared until the cause of the error is eliminated, even if the error clear operation or processing is performed.

* 1 "SP.UNIT DOWN", etc.

For eliminating of error causes, refer to the following.

☞Page 16-1, CHAPTER 16

4.6 Output (Y) Status Setting for Switching STOP to RUN

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	○

(1) Output (Y) status setting for changing the status from STOP to RUN

When the C Controller module status is switched from RUN to STOP, the output data (Y) in the RUN status are saved in the C Controller module and actual outputs are all turned off.
In the STOP status, the output (Y) status cannot be changed from the user program.
Either of the following two options can be selected for the case of switching STOP to RUN.

- Outputting the output status data before STOP. (Previous state)
- Clearing the output (Y) data. (Reset output (Y))

(2) Setting applications

When the C Controller module status is changed from STOP to RUN, whether to resume the previous output or not can be selected in advance.

- When "Previous state" is selected

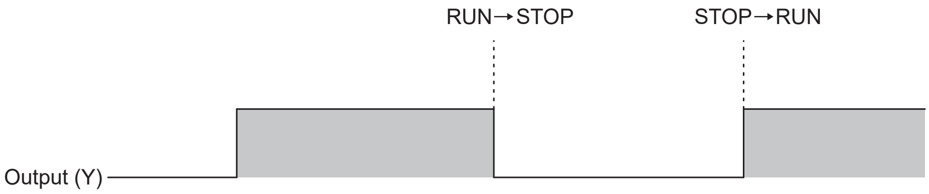


Figure 4.6 Timing chart for "Previous state" setting

- When "Reset output (Y)" is selected

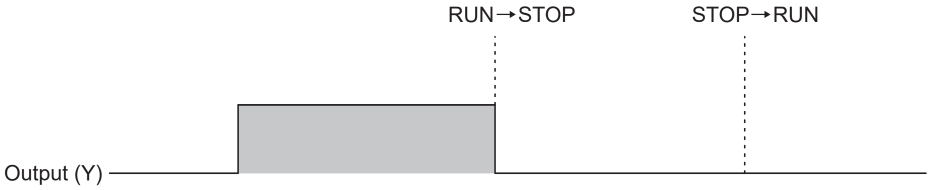


Figure 4.7 Timing chart for "Reset output (Y)" setting

(3) Processing

(a) Previous state (Default)
The output status (Y) immediately before the change to STOP is output.
Then, output from the user program is enabled.

(b) Reset output (Y)
The outputs turn to OFF.
Then, output from the user program is enabled.
For forcing the output (Y) to turn ON in the STOP status, refer to the following.
☞ (5) in this section

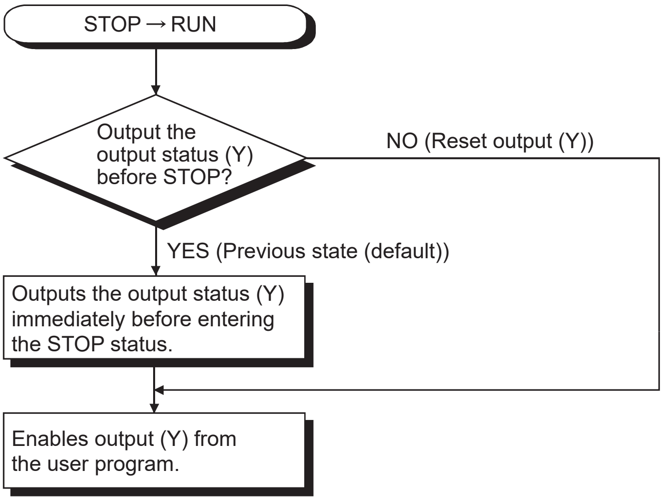
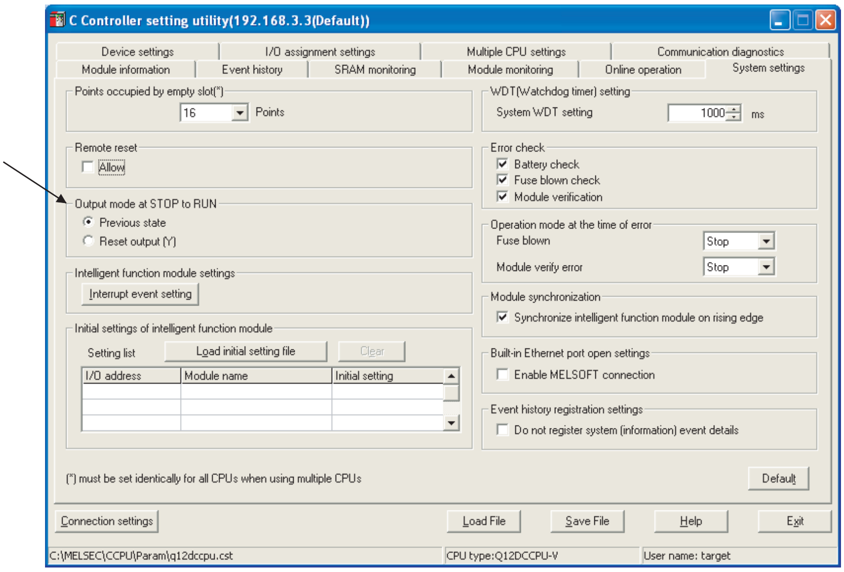

STOP → RUN
Output the output status (Y) before STOP?
NO (Reset output (Y))
YES (Previous state (default))
Outputs the output status (Y) immediately before entering the STOP status.
Enables output (Y) from the user program.

Figure 4.8 Output (Y) processing when the status is changed from STOP to RUN

(4) Output (Y) status setting for switching from STOP to RUN

In <<System settings>> of C Controller setting utility, the output status after switching from STOP to RUN can be selected in advance.

Output mode setting for switching from STOP to RUN



Points occupied by empty slot(*)
16 Points

WDT(Watchdog timer) setting
System WDT setting 1000 ms

Remote reset
□ Allow

Error check
☑ Battery check
☑ Fuse blown check
☑ Module verification

Output mode at STOP to RUN
● Previous state
○ Reset output (Y)

Operation mode at the time of error
Fuse blown: Stop
Module verify error: Stop

Intelligent function module settings

Module synchronization
☑ Synchronize intelligent function module on rising edge

Initial settings of intelligent function module
Setting list

I/O address	Module name	Initial setting

Built-in Ethernet port open settings
□ Enable MELSOFT connection

Event history registration settings
□ Do not register system (information) event details

(*) must be set identically for all CPUs when using multiple CPUs

C:\MELSEC\CCPU\Param\q12dccpu.cst | CPU type:Q12DCCPU-V | User name: target

Figure 4.9 Output mode setting for switching STOP to RUN

(5) Precautions

(a) When the output (Y) is forced to turn ON in the STOP status
The table below shows the output operation when the output (Y) is forced to turn ON with the C Controller module set to STOP and then the STOP status is changed to RUN.

Table 4.7 When output (Y) is forced to turn ON in the STOP status and then the status is changed to RUN

"Output mode at STOP to RUN" setting	Output after changing STOP to RUN
Previous state	Outputs the status before STOP. If the output status (Y) before STOP is OFF, the ON status is not held.
Reset output (Y)	Holds the ON status.

4.7 Clock Function

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	○

(1) What is the clock function?

With the clock function, clock data in the C Controller module are read out by the user program to be used for time control.

These clock data are mainly used for time control of the C Controller module system, such as storing time and date information in the event history.

(2) Clock operation during power off, reset or momentary power failure

Even while the C Controller module is powered off, reset or has failed for longer than the allowable momentary power failure time, the clock operation continues by the battery inside the C Controller module.

(3) Clock data

The following clock data are used in the C Controller module.

Table 4.8 Clock data details

Data name	Description	
YY	4 digits (available from 2000 to 2099)	
MM	1 to 12	
DD	1 to 31 (automatic leap year adjustment)	
Hr.	0 to 23 (24-hour system)	
Min.	0 to 59	
Sec.	0 to 59	
Day of week	0	Sunday
	1	Monday
	2	Tuesday
	3	Wednesday
	4	Thursday
	5	Friday
	6	Saturday

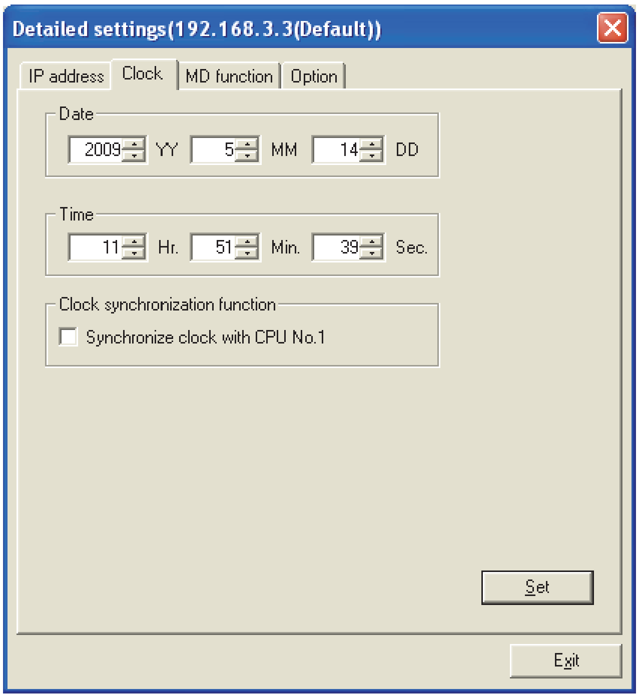
(4) Changing and reading clock data

(a) Changing clock data

Clock data can be changed by either of the following two methods: "Using C Controller setting utility" and "Using a user program".

1) Using C Controller setting utility

In <<Online operation>> of C Controller setting utility, click Detail and in <<Clock>>, change the clock data.



Detailed settings(192.168.3.3(Default))

IP address | Clock | MD function | Option

Date: 2009 YY, 5 MM, 14 DD

Time: 11 Hr., 51 Min., 39 Sec.

Clock synchronization function

□ Synchronize clock with CPU No.1

Figure 4.10 Writing clock data from C Controller setting utility

2) Using a user program

In the user program, change the clock data using the time data setting function (QBF_SetTime).

First, obtain clock data from an external computer or programmable controller CPU on the network.

Then, by setting the clock data again for the C Controller module with the QBF_SetTime function, adjust the clock of the C Controller module.

(b) Reading clock data

To read clock data, use the clock data read function (QBF_GetTime) on the user program.

(5) Clock data accuracy

(a) While the C Controller module is ON

- The clock function accuracy is the same as that of the VxWorks POSIX Clock.
  A higher frequency of interrupt program execution may cause a longer clock delay, depending on the system configuration of the C Controller module.
  For details of POSIX Clock, refer to the following.
  ☞ Manual for VxWorks
- When correcting the clock data, make them synchronized with the clock of the SNTP server computer on the network, using the user program.
  Obtain clock data from the SNTP server computer, and refer to the sample program (SntpTimeSet.c) for setting clock data.

(b) While the C Controller module is OFF
The clock function accuracy varies depending on the ambient temperature of the C Controller module. (☞ Page 3-2, Section 3.2)

(6) Precautions

(a) Setting clock data for the first time
No clock data are set for the C Controller module prior to shipping.
Clock data are used by the C Controller module's system, such as event history, and intelligent function modules.
Therefore, before using the C Controller module for the first time, be sure to set accurate clock time.

(b) Setting clock data when no battery is installed
If no battery is installed in the C Controller module, clock data must be set.
Otherwise, clock data will be incorrect while the C Controller module is ON.

(c) Correcting clock data

1) Even if a part of clock data is to be corrected, all of clock data must be written to the C Controller module again.
2) If the year 2100 is reached after clock setting, clock data of the year 2100 can be used until restart of the C Controller module.
   Set them again to the year 2000 to 2099 when restarting the C Controller module.
3) The C Controller module's clock must set with the QBF_WaitEvent or QBF_WaitUnitEvent function not being executed.

(d) Changing clock data by a user program
When changing clock data by a user program, use the QBF_SetTime function.
If any other clock data setting function is used, correct clock data will not be set in the C Controller module.

(e) Reading clock data by a user program
When reading clock data by the user program, functions other than QBF_GetTime can be used.

Remark

For the bus interface functions, refer to the following.
☞ C Controller Module User's Manual (Utility Operation, Programming)

4.7.1 Multiple CPU Clock Synchronization Function

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	×	×

(1) What is the multiple CPU clock synchronization function?

The multiple CPU clock synchronization function allows clock data synchronization with CPU No.1 (programmable controller CPU or C Controller module) when the C Controller module is set as CPU No.2, No.3, or No.4 in a multiple CPU system.

Clock data of CPU No.1 are automatically set when they are changed in C Controller setting utility or user program.

Clock data will be received from CPU No.1 at the following timing.

- When the multiple CPU system is powered on
- When the multiple CPU system status is changed from RESET or STOP to RUN
- At intervals of one second after startup of the multiple CPU system

The clock data include year, month, day, day of week, hour, minute, and second data.

(2) Setting of the multiple CPU clock synchronization function

In <<Online operation>> of C Controller setting utility, click the Detail button, and then in <<Clock>>, enable or disable the clock synchronization function.

(3) Precautions

(a) Clock setting of CPU No.1 (when the multiple CPU clock synchronization function is enabled)

1) Set the clock for CPU No.1.

2) Because CPU No.1 updates the clock data at intervals of one second, an error of up to one second may be produced.

3) Clock data of CPU No.1 must be set to the year 2000 or later.
The C Controller module supports only the year 2000 or later clock data.

Remark

Synchronization with the clock of the SNTP server computer on the network must be set using a user program.

For the sample program (SntpTimeSet.c), refer to the following.

☞ C Controller Module User's Manual (Utility Operation, Programming)

4.8 Input Response Time Selection (I/O Response Time)

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	○

(1) What is the input response time selection?

The input response time can be changed for each Q series module.

The following table shows the modules whose input response time can be changed and time options for them.

Table 4.9 Modules whose input response time can be changed

Module whose input response time can be changed	Type	Option
Input module	Input	1ms, 5ms, 10ms, 20ms, 70ms (Default: 10ms)
I/O combined module	I/O Mix	
High-speed input module	Hi Input	0.1ms, 0.2ms, 0.4ms, 0.6ms, 1ms (Default: 0.2ms)
Interrupt module	Interrupt	

The above Q series modules load external inputs within the specified input response time.

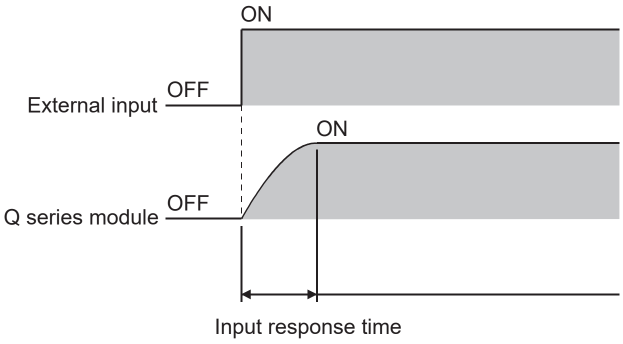


Figure 4.11 Input response time

(2) Input response time setting

The input response time is set in "I/O assignment" in <<I/O assignment settings>> of C Controller setting utility.

1) Configure the I/O assignment settings.

2) Select Detailed settings .

3) Set each input response time in the "I/O module, Intelligent function module detailed settings" window.

1) Configure I/O assignment settings. 2) Click the Detailed settings button. 3) Select an input response time.

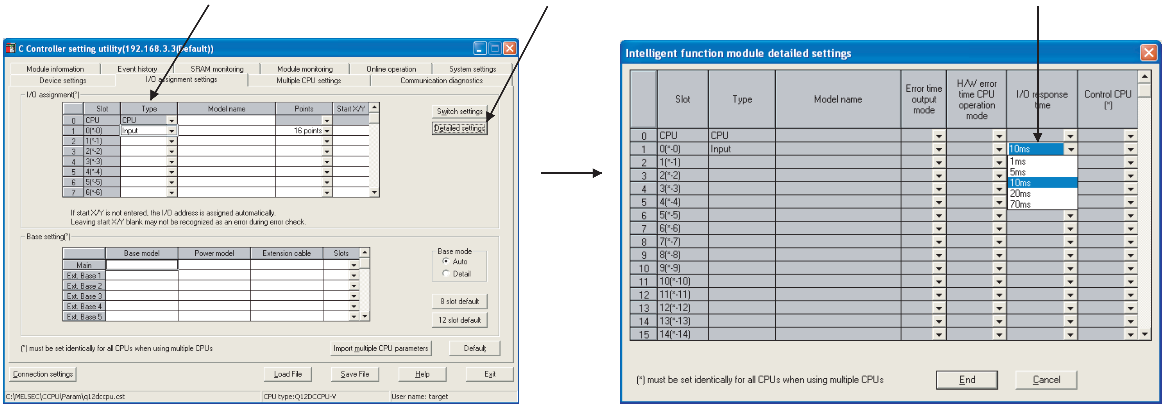


Figure 4.12 Input response time setting

(3) Precautions

(a) When the input response time is reduced

Reducing the input response time value for high-speed response will make the module noise-sensitive.

Take into account the operating environment when setting the input response time.

(b) Timing when the setting takes effect

The input response time setting takes effect when:

- The C Controller system is powered on, or
- The C Controller module is reset.

1 OVERVIEW
2 SYSTEM CONFIGURATION
3 SPECIFICATIONS
4 FUNCTIONS
5 ACCESS VIA NETWORK MODULES
6 PREPARATORY PROCEDURES AND SETTING
7 I/O NUMBER ASSIGNMENT
8 MEMORIES AND FILES

4.9 Error Time Output Mode Setting

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	○

(1) What is the error time output mode setting?

This setting determines whether to clear or hold the outputs to the following Q series modules when a stop error occurs in the C Controller module.

- Output modules
- I/O combined modules
- Intelligent function modules
- Interrupt modules

(2) Error time output mode setting

The error time output mode is set in "I/O assignment" in << I/O assignment settings>> of C Controller setting utility.

1) Configure the I/O assignment settings.

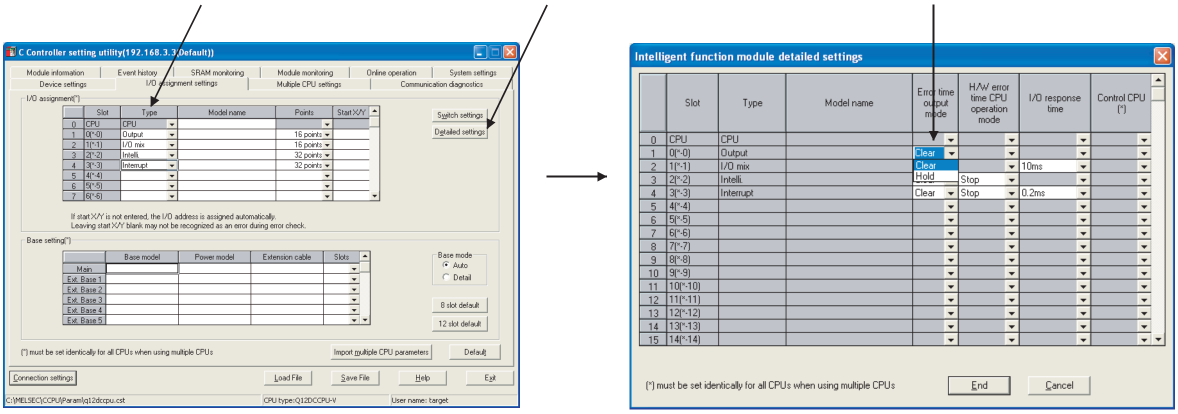
2) Select Detailed settings.

3) Select "Clear" or "Hold" for each slot. (The default is "Clear".)

1) Configure I/O assignment settings.

2) Click the Detailed settings button.

3) Select "Clear" or "Hold."



C Controller setting utility(192.168.3.3(Default))

I/O assignment(*)

	Slot	Type	Model name	Points	Start X/Y
0	CPU	CPU			
1	0(*-0)	Output		16 points	
2	1(*-1)	I/O mix		16 points	
3	2(*-2)	Intelli.		32 points	
4	3(*-3)	Interrupt		32 points	
5	4(*-4)				
6	5(*-5)				
7	6(*-6)				

If start X/Y is not entered, the I/O address is assigned automatically.
Leaving start X/Y blank may not be recognized as an error during error check.

Base setting(*)

	Base model	Power model	Extension cable	Slots
Main				
Ext. Base 1				
Ext. Base 2				
Ext. Base 3				
Ext. Base 4				
Ext. Base 5				

Base mode: ● Auto ○ Detail

(*) must be set identically for all CPUs when using multiple CPUs

C:\MELSEC\CCPU\Param\q12dccpu.cst | CPU type:Q12DCCPU-V | User name: target

Intelligent function module detailed settings

	Slot	Type	Model name	Error time output mode	H/W error time CPU operation mode	I/O response time	Control CPU (*)
0	CPU	CPU					
1	0(*-0)	Output		Clear			
2	1(*-1)	I/O mix				10ms	
3	2(*-2)	Intelli.			Stop		
4	3(*-3)	Interrupt		Clear	Stop	0.2ms	
5	4(*-4)						
6	5(*-5)						
7	6(*-6)						
8	7(*-7)						
9	8(*-8)						
10	9(*-9)						
11	10(*-10)						
12	11(*-11)						
13	12(*-12)						
14	13(*-13)						
15	14(*-14)						

(*) must be set identically for all CPUs when using multiple CPUs

Figure 4.13 Error time output mode setting

(3) Precautions

The error time output mode setting takes effect when:

- The C Controller system is powered on.
- The C Controller module is reset.

4.10 Hardware Error Time CPU Operation Mode Setting

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	○

(1) What is the hardware error time CPU operation mode setting?

This setting determines whether to stop or continue the output (Y) from the C Controller module and writing to the buffer memory if a hardware error occurs in an intelligent function module or interrupt module.

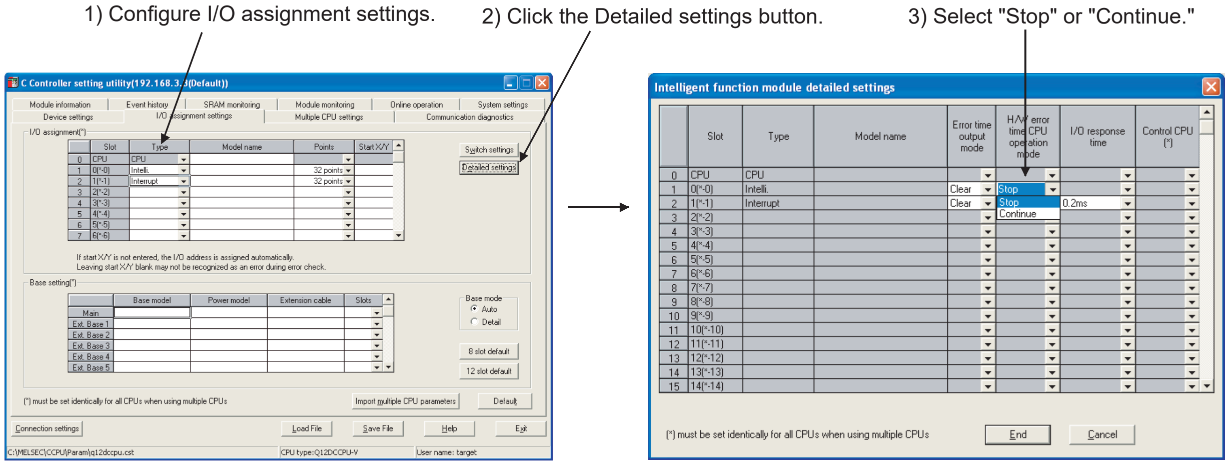
(2) Hardware error time CPU operation mode setting

This is set in "I/O assignment" in << I/O assignment settings>> of C Controller setting utility.

1) Configure the I/O assignment settings.

2) Select Detailed settings.

3) Select a CPU operation mode in case of a hardware error for each slot. (The default is "Stop".)



1) Configure I/O assignment settings.

2) Click the Detailed settings button.

3) Select "Stop" or "Continue."

C Controller setting utility(192.168.3.3(Default))

I/O assignment(*)

	Slot	Type	Model name	Points	Start X/Y
0	CPU	CPU			
1	0(*-0)	Intelli.		32 points	
2	1(*-1)	Interrupt		32 points	
3	2(*-2)				
4	3(*-3)				
5	4(*-4)				
6	5(*-5)				
7	6(*-6)				

If start X/Y is not entered, the I/O address is assigned automatically.
Leaving start X/Y blank may not be recognized as an error during error check.

Base setting(*)

	Base model	Power model	Extension cable	Slots
Main				
Ext. Base 1				
Ext. Base 2				
Ext. Base 3				
Ext. Base 4				
Ext. Base 5				

Base mode: ● Auto ○ Detail

(*) must be set identically for all CPUs when using multiple CPUs

C:\MELSEC\CCPU\Param\q12dccpu.cst
CPU type:Q12DCCPU-V
User name: target

Intelligent function module detailed settings

	Slot	Type	Model name	Error time output mode	H/W error time CPU operation mode	I/O response time	Control CPU (*)
0	CPU	CPU					
1	0(*-0)	Intelli.		Clear	Stop		
2	1(*-1)	Interrupt		Clear	Stop Continue	0.2ms	
3	2(*-2)						
4	3(*-3)						
5	4(*-4)						
6	5(*-5)						
7	6(*-6)						
8	7(*-7)						
9	8(*-8)						
10	9(*-9)						
11	10(*-10)						
12	11(*-11)						
13	12(*-12)						
14	13(*-13)						
15	14(*-14)						

(*) must be set identically for all CPUs when using multiple CPUs

Figure 4.14 Hardware error time CPU operation mode setting

(3) Precautions

The hardware error time CPU operation mode setting takes effect when:

- The C Controller system is powered on.
- The C Controller module is reset.

4.11 Switch Settings for I/O and Intelligent Function Modules

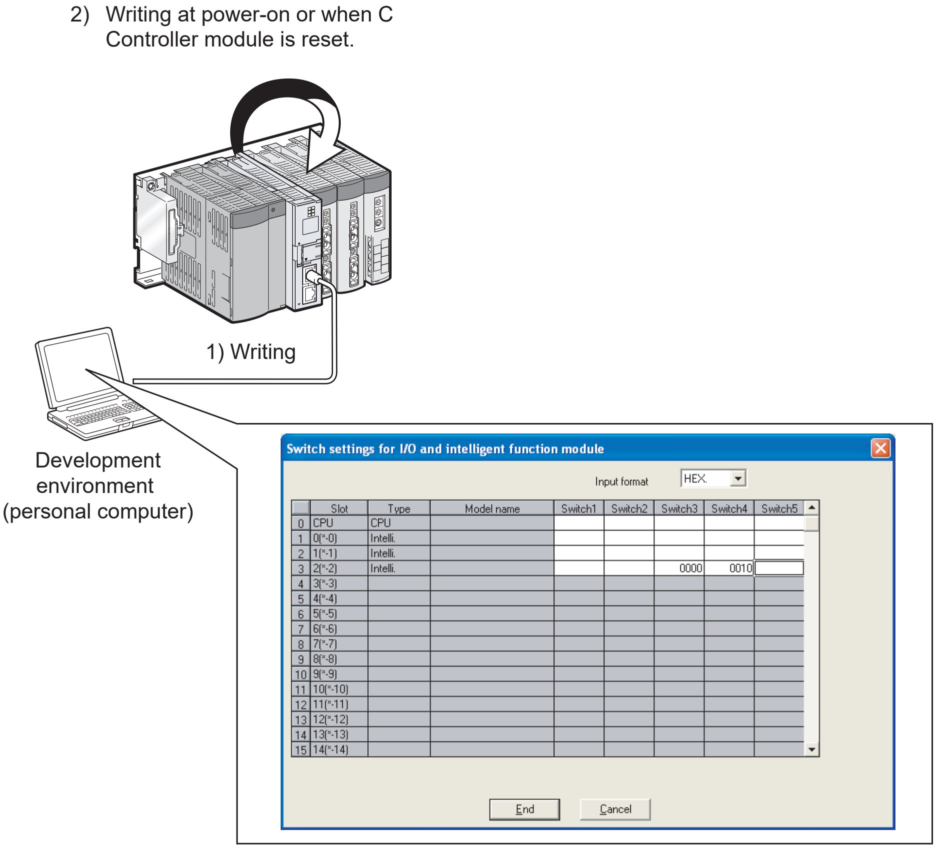
Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	○

(1) What is the switch settings for intelligent function modules and interrupt modules?

The switches of the Q series intelligent function modules and interrupt modules can be set in C Controller setting utility.

(2) Timing of writing the switch settings

The switch settings are written from the C Controller module to each intelligent function module and/or interrupt module when the C Controller system is powered on or the C Controller module is reset.


2) Writing at power-on or when C Controller module is reset.
1) Writing
Development environment (personal computer)
Switch settings for I/O and intelligent function module
End
Cancel

Input format HEX.

	Slot	Type	Model name	Switch1	Switch2	Switch3	Switch4	Switch5
0	CPU	CPU						
1	0(*-0)	Intelli.						
2	1(*-1)	Intelli.						
3	2(*-2)	Intelli.				0000	0010	
4	3(*-3)							
5	4(*-4)							
6	5(*-5)							
7	6(*-6)							
8	7(*-7)							
9	8(*-8)							
10	9(*-9)							
11	10(*-10)							
12	11(*-11)							
13	12(*-12)							
14	13(*-13)							
15	14(*-14)							

Intelligent function module switch setting from <<I/O assignment settings>>

Figure 4.15 Writing switch settings to a module

(3) Switch setting of intelligent function modules and interrupt modules

The switches of intelligent function modules and interrupt modules are set in "I/O assignment" in << I/O assignment settings>> of C Controller setting utility.

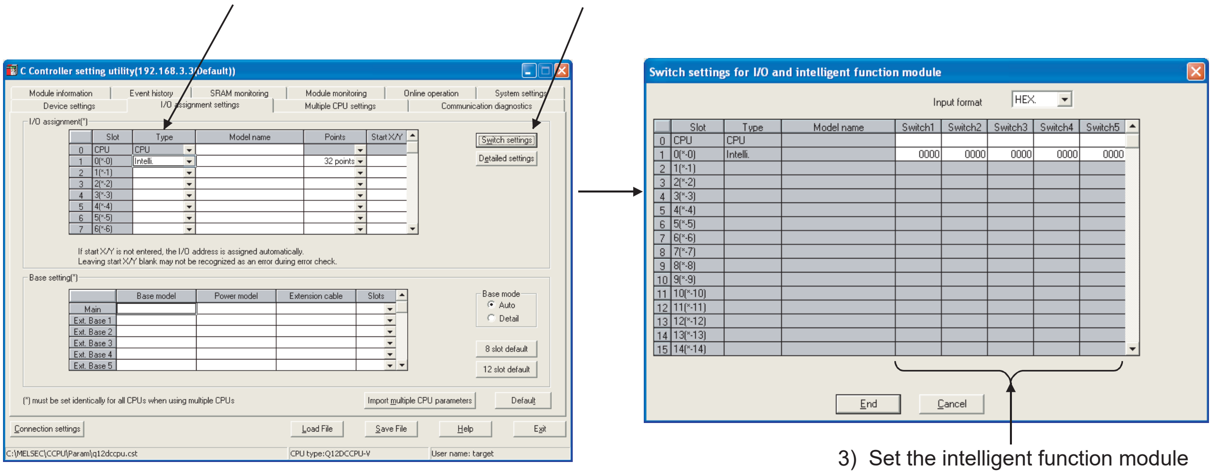
1) Configure the I/O assignment settings.

2) Select Switch settings.

3) Set the switches of each module.

1) Configure I/O assignment settings.

2) Click the Switch settings button.



C Controller setting utility(192.168.3.3(Default))

I/O assignment(*)

	Slot	Type	Model name	Points	Start X/Y
0	CPU	CPU			
1	0(*-0)	Intelli.		32 points	
2	1(*-1)				
3	2(*-2)				
4	3(*-3)				
5	4(*-4)				
6	5(*-5)				
7	6(*-6)				

If start X/Y is not entered, the I/O address is assigned automatically.
Leaving start X/Y blank may not be recognized as an error during error check.

Base setting(*)

	Base model	Power model	Extension cable	Slots
Main				
Ext. Base 1				
Ext. Base 2				
Ext. Base 3				
Ext. Base 4				
Ext. Base 5				

Base mode: ● Auto ○ Detail

(*) must be set identically for all CPUs when using multiple CPUs

C:\MELSEC\CCPU\Param\q12dccpu.cst | CPU type:Q12DCCPU-V | User name: target

Switch settings for I/O and intelligent function module

Input format: HEX.

	Slot	Type	Model name	Switch1	Switch2	Switch3	Switch4	Switch5
0	CPU	CPU						
1	0(*-0)	Intelli.		0000	0000	0000	0000	0000
2	1(*-1)							
3	2(*-2)							
4	3(*-3)							
5	4(*-4)							
6	5(*-5)							
7	6(*-6)							
8	7(*-7)							
9	8(*-8)							
10	9(*-9)							
11	10(*-10)							
12	11(*-11)							
13	12(*-12)							
14	13(*-13)							
15	14(*-14)							

3) Set the intelligent function module and interrupt module switches.

Figure 4.16 Switch setting

(4) Precautions

(a) For the switch settings of intelligent function modules and interrupt modules, refer to the following.

☞ Manual for each intelligent function module or interrupt module

(b) The settings for intelligent function modules and interrupt modules take effect when:

- The C Controller system is powered on.
- The C Controller module reset is completed.

4.12 Watchdog Timer (WDT)

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	○

(1) What is watchdog timer (WDT)?

Watchdog timers are provided inside the C Controller module to detect failure of the hardware or user program.

The C Controller module has the following two different watchdog timers.

(a) System watchdog timer

The system watchdog timer monitors the system of the C Controller module, and is used to detect a fault in the hardware or system software.

(b) User watchdog timer

The user watchdog timer monitors the user programs, and is used to detect an error in the user programs.

(2) When a watchdog timer times out

When a watchdog timer times out, a watchdog timer error is detected and the following conditions occur.

- The C Controller module turns off all the outputs.
- The RUN LED on the front of the C Controller module turns off, and the ERR. LED flashes. (Only for a system watchdog timer error, the MODE LED also turns off.)

(3) Setting and resetting a watchdog timer

(a) Setting a watchdog timer

1) System watchdog timer

Time for the system watchdog timer can be set or changed in <<System settings>> of C Controller setting utility.

The default is 1000ms. (This can be changed within the range of 20ms to 2000ms (in 10ms units).)

2) User watchdog timer

Time for the user watchdog timer can be set by the QBF_StartWDT function within the range of 100ms to 10000ms (in 10ms units).

(b) Resetting a watchdog timer

1) System watchdog timer

The system watchdog timer is periodically reset in the C Controller module.

If the system processing is interrupted for a long time due to a C Controller module hardware error or an interrupt program execution, the system watchdog timer can time out.

2) User watchdog timer
The user watchdog timer starts monitoring by the QBF_StartWDT function, and it is reset by the QBF_ResetWDT function.
The user watchdog timer times out when the user program was not able to complete the processing within the time specified by the QBF_StartWDT function and could not execute the QBF_ResetWDT function.

4.13 Interrupt from Intelligent Function Module

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	○

(1) Interrupt from an intelligent function module

A C controller module can perform the following processing when an interrupt occurs from the intelligent function module or interrupt module.

- A routine registered in the QBF_EntryUnitInt function can be executed as an interrupt routine (interrupt program).*1*2
- A user program which is in interrupt event waiting status by the QBF_WaitUnitEvent function can be restarted.

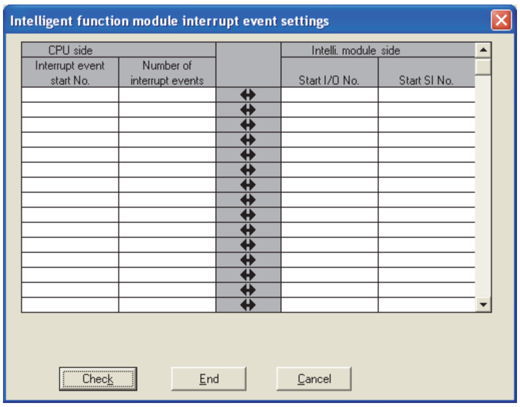
* 1 For the following C Controller modules, the routine cannot be executed as an interrupt routine (interrupt program).
- Q12DCCPU-V with a serial number whose first five digits are "12041" or higher
- Q06CCPU-V(-B)

* 2 Interrupt routine (interrupt program) does not produce a delay caused by user program execution cycles or the priority in multitasking.
For this reason, real-time processing can be performed when an interrupt occurs.
However, functions that can be used are restricted because of ISR (Interrupt Service Routine) of VxWorks 6.4.
For ISR restrictions, refer to the VxWorks manual.

(2) Setting for interrupt from an intelligent function module

To receive an interrupt from an intelligent function module, set "Intelligent function module settings" in <<System settings>> of C Controller setting utility.

(a) Intelligent function module interrupt event settings



Intelligent function module interrupt event settings

CPU side			Intelli. module side	
Interrupt event start No.	Number of interrupt events		Start I/O No.	Start SI No.
		↔		
		↔		
		↔		
		↔		
		↔		
		↔		
		↔		
		↔		
		↔		
		↔		
		↔		
		↔		
		↔		
		↔		
		↔		

Figure 4.17 Intelligent function module interrupt event settings

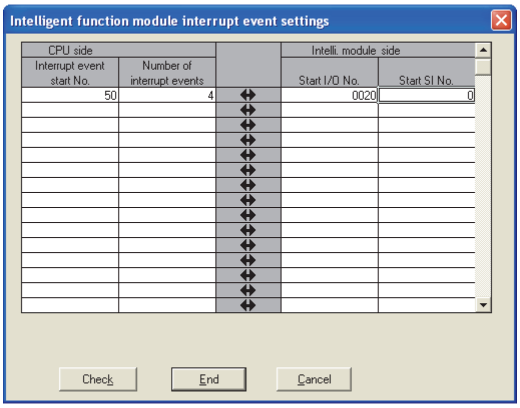
1) CPU side "Interrupt event start No."
Set the start No. of the interrupt event for the C Controller module.
Setting range: 50 to 255

2) CPU side "Number of interrupt events"
Set the number of interrupt execution conditions set by the intelligent function module.
Setting range: 1 to 16

3) Intelli. module side "Start I/O No."
Set the start I/O No. of the intelligent function module from which an interrupt is to be performed.
Setting range: 0000 to 0FF0 or 3E00 to 3E30(H)

4) Intelli. module side "Start SI No."
Set the intelligent function module interrupt pointer No. that was set at the intelligent function module.
Setting range: 0 to 15

(b) Setting example
The following is a setting example, showing that SI 0 to 3 of an intelligent function module mounted to the slot of start I/O No.20 are assigned to interrupt pointers 50 to 53.



Intelligent function module interrupt event settings

CPU side			Intelli. module side	
Interrupt event start No.	Number of interrupt events		Start I/O No.	Start SI No.
50	4	⇔	0020	0
		⇔		
		⇔		
		⇔		
		⇔		
		⇔		
		⇔		
		⇔		
		⇔		
		⇔		
		⇔		
		⇔		
		⇔		
		⇔		
		⇔		

Figure 4.18 Interrupt event setting example

Remark

This section explains the C Controller module side setting.
For the intelligent function module setting, refer to the following.
☞ Manual for the intelligent function module

(3) Processing overview

(a) Executing interrupt routines (interrupt programs)

This section explains how to execute a routine registered in the QBF_EntryUnitInt function as an interrupt routine (interrupt program).

1) This function registers a routine that corresponds to an interrupt from the intelligent function module or interrupt module by the QBF_EntryUnitInt.

2) Enable the routine registered in step 1) by the QBF_EnableUnitInt function or QBF_EnableUnitInt_ISR function.

3) When an interrupt from the intelligent function module or interrupt module is executed in the status of 2), the routine registered in 1) is executed as an interrupt routine (interrupt program).

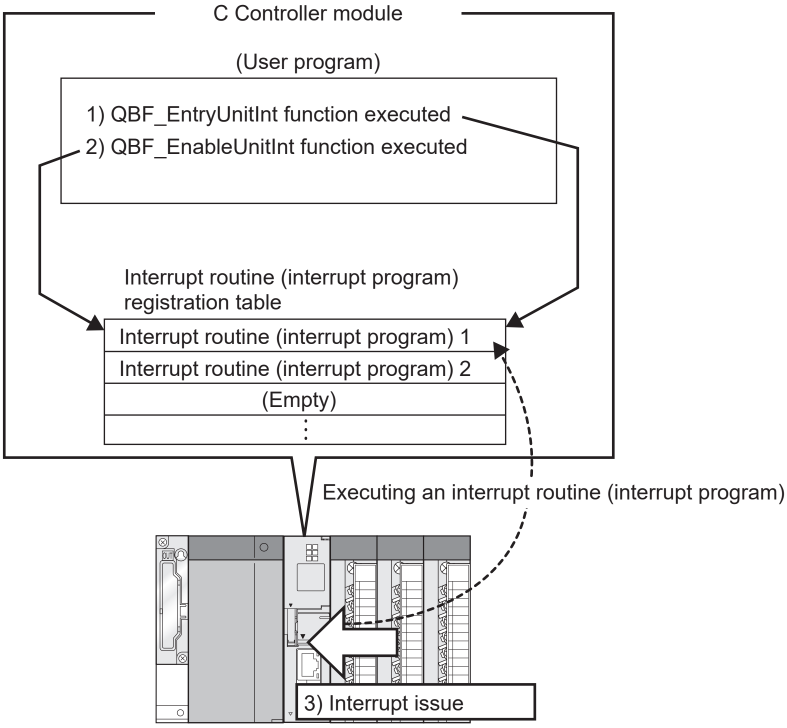

C Controller module
(User program)
1) QBF_EntryUnitInt function executed
2) QBF_EnableUnitInt function executed
Interrupt routine (interrupt program) registration table
Interrupt routine (interrupt program) 1
Interrupt routine (interrupt program) 2
(Empty)
⋮
Executing an interrupt routine (interrupt program)
3) Interrupt issue

Figure 4.19 Executing interrupt routines (interrupt programs)

Not to execute an interrupt routine (interrupt program) when an interrupt occurs, perform the following procedure.

4) Disable the routine registered in 1) with the QBF_DisableUnitInt function or QBF_DisableUnitInt_ISR function.

5) The routine registered in 1) will not be executed even if an interrupt is issued from the intelligent function module or interrupt module in status of 4).

(b) Restarting a user program

This section explains how to restart a user program which is in the interrupt event waiting status by the QBF_WaitUnitEvent function.

1) In the user program, call the QBF_WaitUnitEvent function.
2) The user program is placed into the interrupt event waiting status by 1).
3) An interrupt from the intelligent function module or interrupt module is issued in the status of 2).
4) The user program receives an interrupt event by 3) and restarts.

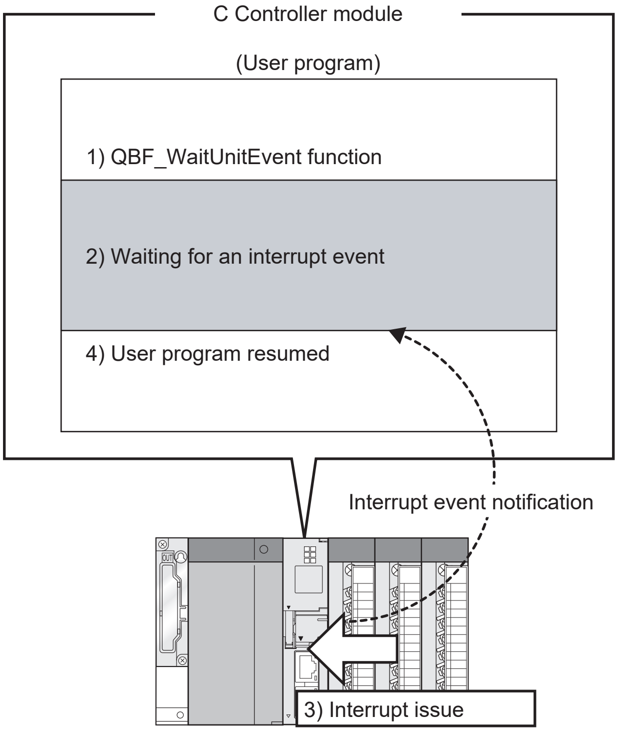
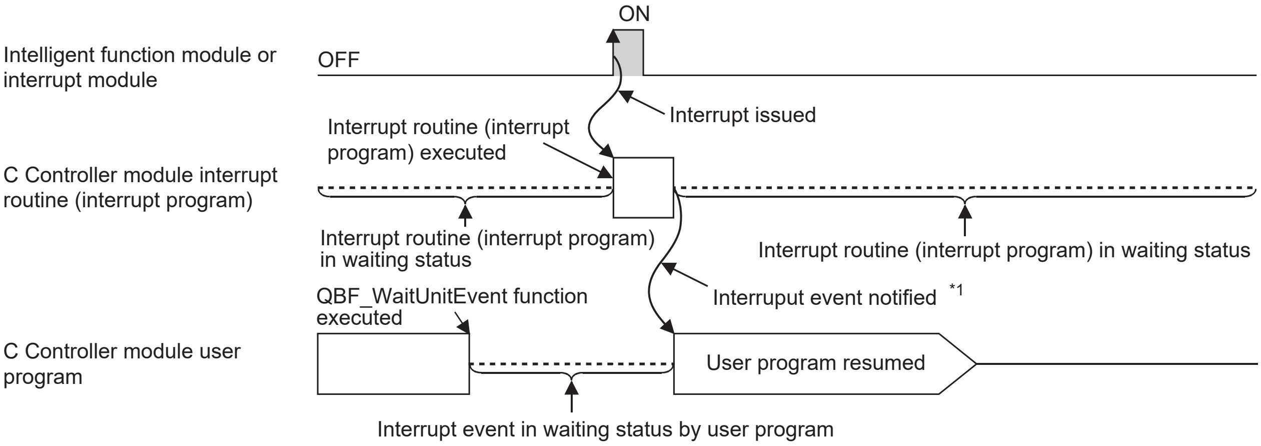

C Controller module
(User program)
1) QBF_WaitUnitEvent function
2) Waiting for an interrupt event
4) User program resumed
Interrupt event notification
3) Interrupt issue

Figure 4.20 Restarting a user program

(c) Operation timing


ON
Intelligent function module or interrupt module
OFF
Interrupt issued
Interrupt routine (interrupt program) executed
C Controller module interrupt routine (interrupt program)
Interrupt routine (interrupt program) in waiting status
Interrupt routine (interrupt program) in waiting status
QBF_WaitUnitEvent function executed
Interruput event notified *1
C Controller module user program
User program resumed
Interrupt event in waiting status by user program

*1 Interrupt event notified after the interrupt routine (interrupt program) ends.

Figure 4.21 Operation timing

(4) Function

The following are functions used in interrupts from the intelligent function module or interrupt module.

Table 4.10 Functions used for an interrupt from the intelligent function module or interrupt module

Function name	Function
QBF_EntryUnitInt	Registers a routine so that it can be called when an interrupt occurs from the intelligent function module or interrupt module.
QBF_EnableUnitInt	Enables the routine registered for interrupt from the intelligent function module or interrupt module.
QBF_DisableUnitInt	Disables the routine registered for interrupt from the intelligent function module or interrupt module.
QBF_EnableUnitInt_ISR	Enables the routine registered for interrupt from the intelligent function module or interrupt module.
QBF_DisableUnitInt_ISR	Disables the routine registered for interrupt from the intelligent function module or interrupt module.
QBF_WaitUnitEvent	Waits for an interrupt event notification from the intelligent function module or interrupt module.

Remark

For bus interface functions, refer to the following.

☞ C Controller Module User's Manual (Utility Operation, Programming)

(5) Precautions

(a) Interrupt event is already notified when the QBF_WaitUnitEvent function is executed

When an interrupt event from the intelligent function module or interrupt module is already notified when the QBF_WaitUnitEvent function is executed, the user program restarts from the interrupt event waiting status simultaneously with the execution of the QBF_WaitUnitEvent function.

If multiple interrupt events with the same interrupt event No. at the time of the QBF_WaitUnitEvent function execution are already notified, the user program processes them as a single interrupt event notification.

(b) Using the QBF_WaitUnitEvent function by multiple user programs

Do not enter the same CPU number or the same interrupt event number by multiple user programs.

Doing so result in the uncertain situation of which user program to receive the interrupt event.

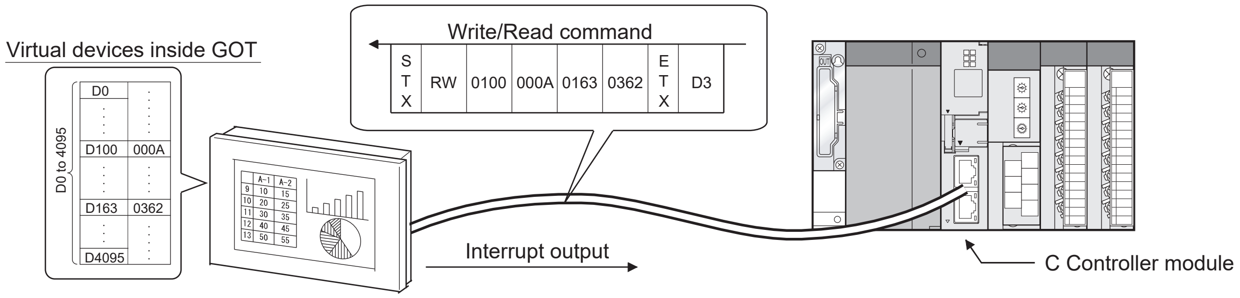
4.14 Connection Between C Controller Module and GOT (Microcomputer Connection)

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	○

(1) Connection between the C Controller module and GOT (Microcomputer connection)

By connecting an RS-232 cable from a GOT to the C Controller module's RS-232 interface, the C Controller module can communicate with the GOT using virtual devices inside the GOT.

For this communication, create a user program referring to a sample program included in SW3PVC-CCPU.



STX	RW	0100	000A	0163	0362	ETX	D3

Figure 4.22 Microcomputer connection of GOT

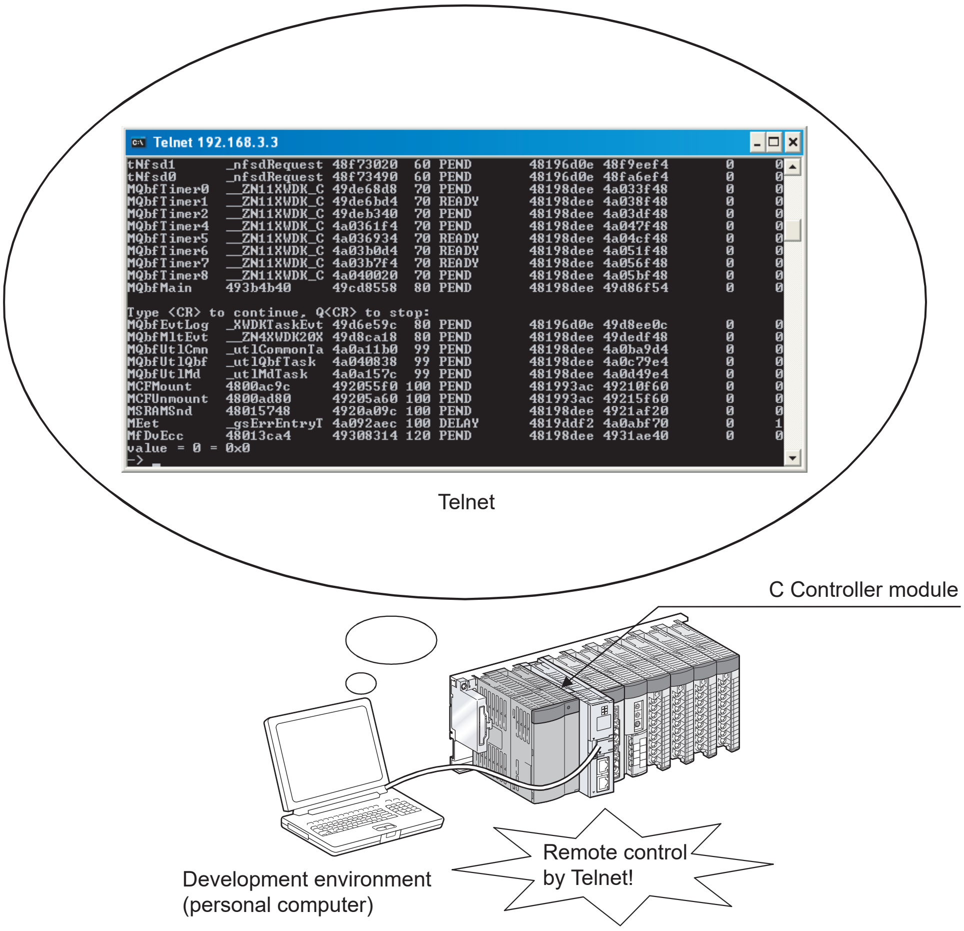
Remark

1. For the sample program, refer to the following.
   ☞ C Controller Module User's Manual (Utility Operation, Programming)
2. For the microcomputer connection of GOT and available communication commands, refer to the following.
   ☞ Manual for the GOT

4.15 Telnet Function

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	○

The Shell commands can be executed from the Telnet tool of the development environment (personal computer) without using Workbench or Tornado.
This allows simple remote debugging (task information display, memory dumping, etc.) of the C Controller module.



Telnet 192.168.3.3

```
tNfsd1      _nfsdRequest 48f73020  60 PEND       48196d0e 48f9eef4       0     0
tNfsd0      _nfsdRequest 48f73490  60 PEND       48196d0e 48fa6ef4       0     0
MQbfTimer0  __ZN11XWDK_C 49de68d8  70 PEND       48198dee 4a033f48       0     0
MQbfTimer1  __ZN11XWDK_C 49de6bd4  70 READY      48198dee 4a038f48       0     0
MQbfTimer2  __ZN11XWDK_C 49deb340  70 PEND       48198dee 4a03df48       0     0
MQbfTimer4  __ZN11XWDK_C 4a0361f4  70 PEND       48198dee 4a047f48       0     0
MQbfTimer5  __ZN11XWDK_C 4a036934  70 READY      48198dee 4a04cf48       0     0
MQbfTimer6  __ZN11XWDK_C 4a03b0d4  70 READY      48198dee 4a051f48       0     0
MQbfTimer7  __ZN11XWDK_C 4a03b7f4  70 READY      48198dee 4a056f48       0     0
MQbfTimer8  __ZN11XWDK_C 4a040020  70 PEND       48198dee 4a05bf48       0     0
MQbfMain    493b4b40     49cd8558  80 PEND       48198dee 49d86f54       0     0

Type <CR> to continue, Q<CR> to stop:
MQbfEvtLog  _XWDKTaskEvt 49d6e59c  80 PEND       48196d0e 49d8ee0c       0     0
MQbfMltEvt  __ZN4XWDK20X 49d8ca18  80 PEND       48198dee 49dedf48       0     0
MQbfUtlCmn  _utlCommonTa 4a0a11b0  99 PEND       48198dee 4a0ba9d4       0     0
MQbfUtlQbf  _utlQbfTask  4a040838  99 PEND       48198dee 4a0c79e4       0     0
MQbfUtlMd   _utlMdTask   4a0a157c  99 PEND       48198dee 4a0d49e4       0     0
MCFMount    4800ac9c     492055f0 100 PEND       481993ac 49210f60       0     0
MCFUnmount  4800ad80     49205a60 100 PEND       481993ac 49215f60       0     0
MSRAMSnd    48015748     4920a09c 100 PEND       48198dee 4921af20       0     0
MEet        _gsErrEntryT 4a092aec 100 DELAY      4819ddf2 4a0abf70       0     1
MfDvEcc     48013ca4     49308314 120 PEND       48198dee 4931ae40       0     0
value = 0 = 0x0
->
```

Telnet
C Controller module
Remote control by Telnet!
Development environment (personal computer)

Figure 4.23 Remote debugging using Telnet

(1) Preparation for use of the Telnet function

A login name and password is required to use the Telnet function.
Set it for the C Controller module before using the Telnet function.

By setting the login user information (login name and password), Telnet users can be restricted. (☞ Page 6-88, Section 6.11)

(2) Available commands

With the Telnet function, the commands that are executable in Workbench or Tornado shells can be used.

(3) Precautions on the Telnet function

(a) Precautions on Telnet connection
If the line is disconnected during use of Telnet, it cannot be reconnected until TCP connection including the Telnet on the C Controller module side times out.
If line disconnection occurs during Telnet connection, reconnect it after timeout.

The timeout time for the C Controller module side Telnet (TCP) connection can be changed by setting the values in the calculation formula and using one of the setting methods introduced for each model.

[Calculation for the Q12DCCPU-V]
The timeout time is determined by the following calculation formula.

Timeout time = net.inet.tcp.keepidle
+ (net.inet.tcp.keepintvl × 8 (number of retries)[*1]) [ms]

Initial value for C Controller module: 30000 (30 seconds)
Initial value for VxWorks: 7800000 (2 hours and 10 minutes)

net.inet.tcp.keepidle: Time elapsed from line disconnection to the first retry (ms)
Initial value for C Controller module: 22000
Initial value for VxWorks: 7200000

net.inet.tcp.keepintvl: Retry interval (ms)
Initial value for C Controller module: 1000
Initial value for VxWorks: 75000

* 1 The number of retries cannot be changed.

[Setting methods for the Q12DCCPU-V]

The following explains how to set the initial value of 30 seconds.

- Setting it while the C Controller module is in operation
  1) Connect the line to the C Controller module with the Telnet tool.
  2) Execute the following two Sysctl() commands with the Telnet tool to set the timeout time to the initial value.

```
Sysctl("net.inet.tcp.keepidle = 22000")
Sysctl("net.inet.tcp.keepintvl = 1000")
```

  3) Close the Telnet connection.

- Setting it when the C Controller module is started
  1) Describe the following two Sysctl() commands on the script file, "STARTUP.CMD".

```
Sysctl("net.inet.tcp.keepidle = 22000")
Sysctl("net.inet.tcp.keepintvl = 1000")
```

  2) Write the above script file, "STARTUP.CMD" to a CompactFlash card, and insert it into the C Controller module.
  3) Upon start of the C Controller module, the timeout time is set to the initial value.

[Calculation for the Q06CCPU-V(-B)]
The timeout time is determined by the following calculation formula.

Timeout time = tcp_keepidle + (tcp_keepintvl ÷ 2 × tcp_keepcnt) [s]

Initial value: 15000 (4 hours and 10 minutes)

tcp_keepidle: Time elapsed from line disconnection to the first retry (s)
Initial value: 14400
tcp_keepintvl: Retry interval (in 0.5s units)
Initial value: 150
tcp_keepcnt: Number of retries
Initial value: 8

[Setting methods for the Q06CCPU-V(-B)]
The following explains how to set the initial value of 4 hours and 10 minutes.

- Setting it while the C Controller module is in operation
  1) Connect the line to the C Controller module with the Telnet tool.
  2) Set the following three external variables with the Telnet tool to change the timeout time to the initial value.

```
tcp_keepidle = 14400
tcp_keepintvl = 150
tcp_keepcnt = 8
```

  3) Close the Telnet connection.

- Setting it when the C Controller module is started
  1) Describe the following three external variable settings on the script file, "STARTUP.CMD".

```
tcp_keepidle = 14400
tcp_keepintvl = 150
tcp_keepcnt = 8
```

  2) Write the above script file, "STARTUP.CMD" to a CompactFlash card, and insert it into the C Controller module.
  3) Upon start of the C Controller module, the timeout time is set to the initial value.

(b) The number of Telnet tools connectable at the same time
Telnet tool must be connected to the C Controller module on a one-to-one basis. Connections from multiple Telnet tools to the same C Controller module is not allowed.
When changing the Telnet tool with another, first close the connection with the currently used Telnet tool, and then connect the line to the C Controller module from another Telnet tool.

(c) When executing a Shell command from the Telnet tool
The Shell commands entered by the Telnet tool of the development environment (personal computer) operate on the tasks of the following priorities

C Controller module	Priority
Q12DCCPU-V	1
Q06CCPU-V(-B)	2

When using a Shell command, pay attention to the following.

- Only alphanumeric characters and special characters can be used.
- Some commands, such as those exclusively using the CPU module and those including characters other than alphanumeric characters and special characters, may be regarded as a command causing a watchdog timeout error, a control code (such as "CTRL + X"), or being garbled. As a result, a system error, such as a system watchdog timer error, or stop may occur in the C Controller module. Pay full attention to the command when entering it.
- Some commands (example: the status-indicating Show command) may disable an interrupt for a long time.
  During that time, processing called from an interrupt routine (interrupt program) (example: bus interface functions for ISR) is not executed.
  Interrupts that are expected to occur at fixed intervals, such as multiple CPU synchronous interrupts, may delay. Be careful when executing a command.

For precautions for program debugging, refer to the following.
☞ C Controller Module User's Manual (Utility Operation, Programming)
A VxWorks message may appear on the Telnet tool screen during a Telnet connection to the C Controller module.
For messages of VxWorks, refer to manuals for VxWorks, Workbench, or Tornado.

(d) Execution of VxWorks reboot command
Do not reboot VxWorks by executing the reboot function or pressing the CTRL + X keys*1. If VxWorks is rebooted, the C Controller module does not start properly.
Reset it on the C Controller module. For resetting, refer to the following.
☞ Page 6-24, Section 6.4

* 1 Do not enter characters other than alphanumeric characters or special characters in Shell either since they may be regarded as a control code.

(e) Command execution without argument specification
If no argument is specified for a command with an argument, the command will be executed with 0 specified as the argument.
Some commands may cause a system error (such as a system watchdog timer error) or a stop in the C Controller module.
Before executing a command, check the specifications and arguments of the command.
Example) Do not execute the "close" command with no argument specified.
Doing so will close the resources reserved in the VxWorks system.

(f) Precautions for program debugging
For precautions for debugging programs with the Telnet function, refer to the following manual.
☞ C Controller Module User's Manual (Utility Operation, Programming)

4.16 Functions for Communicating with Peripheral Devices through an Ethernet Port

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	×	×

Implement communications by connecting a peripheral device (such as GX Developer) to the Ethernet port on the C Controller module with an Ethernet cable.

Functions for communicating with a peripheral device through an Ethernet port cannot be used with the following C Controller modules.

- Q12DCCPU-V with a serial number whose first five digits are "12041" or higher

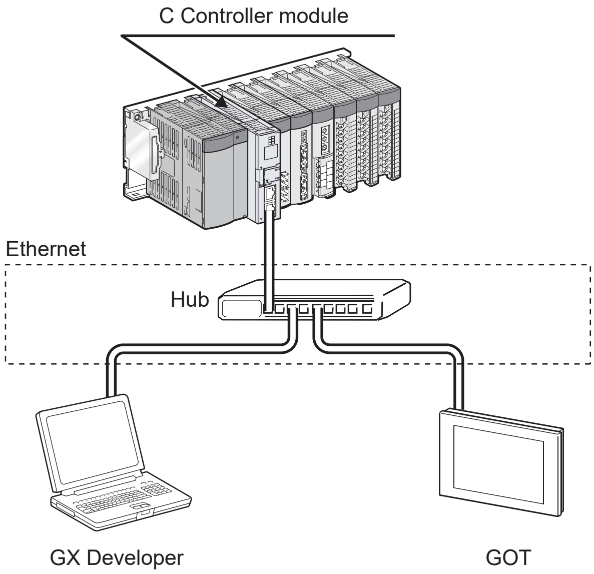


Figure 4.24 Example of the connection with a peripheral device through an Ethernet port

(1) Peripheral devices that can be connected

The following peripheral devices can be connected to the Ethernet port of the C Controller module.

- GX Works2/GX Developer
  For connection methods, refer to the following.
  ☞ Page App-25, Appendix 9
- MT Works2
  For connection methods, refer to the following.
  ☞ User's manual for the Motion CPU module
- MX Component
  For connection methods, refer to the following.
  ☞ Page App-37, Appendix 10
- GOT
  For connection methods, refer to the following.
  ☞ GOT1000 Series Connection Manual (Mitsubishi Products)

(2) Peripheral devices that cannot be connected

The following peripheral devices cannot be connected to the Ethernet port of the C Controller module.

- GX Configurator-QP
- MT Developer

(3) Settings for communication with peripheral devices through an Ethernet port

For communicating with a peripheral device through an Ethernet port, it is necessary to set the built-in Ethernet port open settings in the <<System settings>> tab of the C Controller setting utility.

For details, refer to the following.

C Controller Module User's Manual (Utility Operation, Programming)

(4) Number of available connections

Up to the number of connections shown in the following list can be used in functions for communicating with peripheral devices through an Ethernet port.

Table 4.11 Number of available connections

Protocols	Number of connections
TCP/IP	15
UDP/IP	1

(5) Precautions

(a) Checking by KeepAlive

When protocols are set in TCP, KeepAlive checks the status.

An existence check function is sent 22 seconds after the messaging reception of the last existence check function from the external device to check for a response from the external device.

If there is no response, the existence check function is continuously sent at one-second interval.

If a response cannot be confirmed within 8 seconds, the connection is disconnected due to the lack of existence of the external device. (The connection will be disconnected 30 seconds after the messaging reception of the last existence check function.)

The connection may also be disconnected if the target device is not compatible with the TCP KeepAlive function (a response corresponding to the ACK function for KeepAlive).

(b) Connecting the exceeded number of available connections

Do not over connect the number of available connections.

With TCP connection exceeding the number of available connections, the module may be placed in the following status by the peripheral device.

- The module is placed in a waiting status and cannot be operated.
- The length of a timeout error increases.

(c) Resend processing for TCP connections
When there is no ACK response of the TCP protocols from the external device for sending in TCP connection, resend processing is performed at the following numbers of resends and resend intervals.

Table 4.12 Resend interval

Number of resends	Interval until resend	
	When connection is being established[*1*2]	After connection is established[*1]
First	n seconds	n seconds
Second	n seconds	2n seconds
Third	n seconds	4n seconds
Fourth	n seconds	8n seconds
Fifth	n seconds	16n seconds
Sixth	2n seconds	32n seconds
Seventh	4n seconds	64n seconds
Eighth	8n seconds	64n seconds
Ninth	16n seconds	64n seconds
Tenth	32n seconds	64n seconds
Eleventh	64n seconds	64n seconds
Twelfth	64n seconds	64n seconds

* 1 The variable n is a coefficient calculated dynamically based on the external device and network status.
* 2 The connection is disconnected as an external device error once 30 seconds passed before 12 resends are completed.

After the final resend, if there is no ACK response of the TCP protocol within 64n(*1) seconds, the connection will be disconnected as an external device error.

(d) Connection by UDP
In communication using UDP, simultaneous communication can be performed with only one peripheral device.

(e) Remote STOP/PAUSE for CPU module on another station with GX Works2/GX Developer
Perform remote RUN or remote RESET on other station's CPU module before powering off or resetting the C Controller module if remote stop or remote pause has been performed on other station's CPU module by GX Works2/GX Developer via the Ethernet port on the C Controller module.

(f) Operation using the connections listed below is not guaranteed.
Customers need to confirm the operation before using it.
- Connection using the Internet (a general public line)
- Connection using firewall devices
- Connection using broadband routers
- Connection using wireless LAN

(g) Direct connection to MELSOFT
A C Controller module cannot be connected to MELSOFT directly.

(h) Searching for CPUs with built-in Ethernet on the network
A C Controller module does not respond to search for built-in Ethernet CPUs on the network. (C Controller modules cannot be searched.)

CHAPTER 5 ACCESS VIA NETWORK MODULES

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	△

5.1 Network Module Access Function List

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	△

The following table lists the network module access functions of the C Controller module.

Table 5.1 List of the network module access functions of the C Controller module

Function	Description		Availability			Reference section
			Q12DC CPU-V	Q06C CPU-V	Q06C CPU-V-B	
CC-Link module access function	Allows access from a user program of the C Controller module via CC-Link modules. In a multiple CPU system, access cannot be made via a CC-Link module controlled by another CPU.	Access with a MELSEC data link function	○	○	×	Page 5-2, Section 5.2
		Access with a bus interface function	○	○	○	
MELSECNET/H module access function	Allows access from a user program of the C Controller module via MELSECNET/H modules. In a multiple CPU system, access cannot be made via a MELSECNET/H module controlled by another CPU.	Access with a MELSEC data link function	○	○	×	Page 5-10, Section 5.3
		Access with a bus interface function	○	○	○	
CC-Link IE controller network module access function	Allows access from a user program of the C Controller module via CC-Link IE controller network modules. In a multiple CPU system, access cannot be made via a CC-Link IE controller network module controlled by another CPU.	Access with a MELSEC data link function	○	○	×	Page 5-38, Section 5.4
		Access with a bus interface function	○	○	×	

○: Available, ×: N/A

5.2 CC-Link Module Access Function

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	△

(1) The CC-Link module access function

This function allows access to each station, programmable controller CPU on another station, or C Controller module that is connected to CC-Link via CC-Link modules.

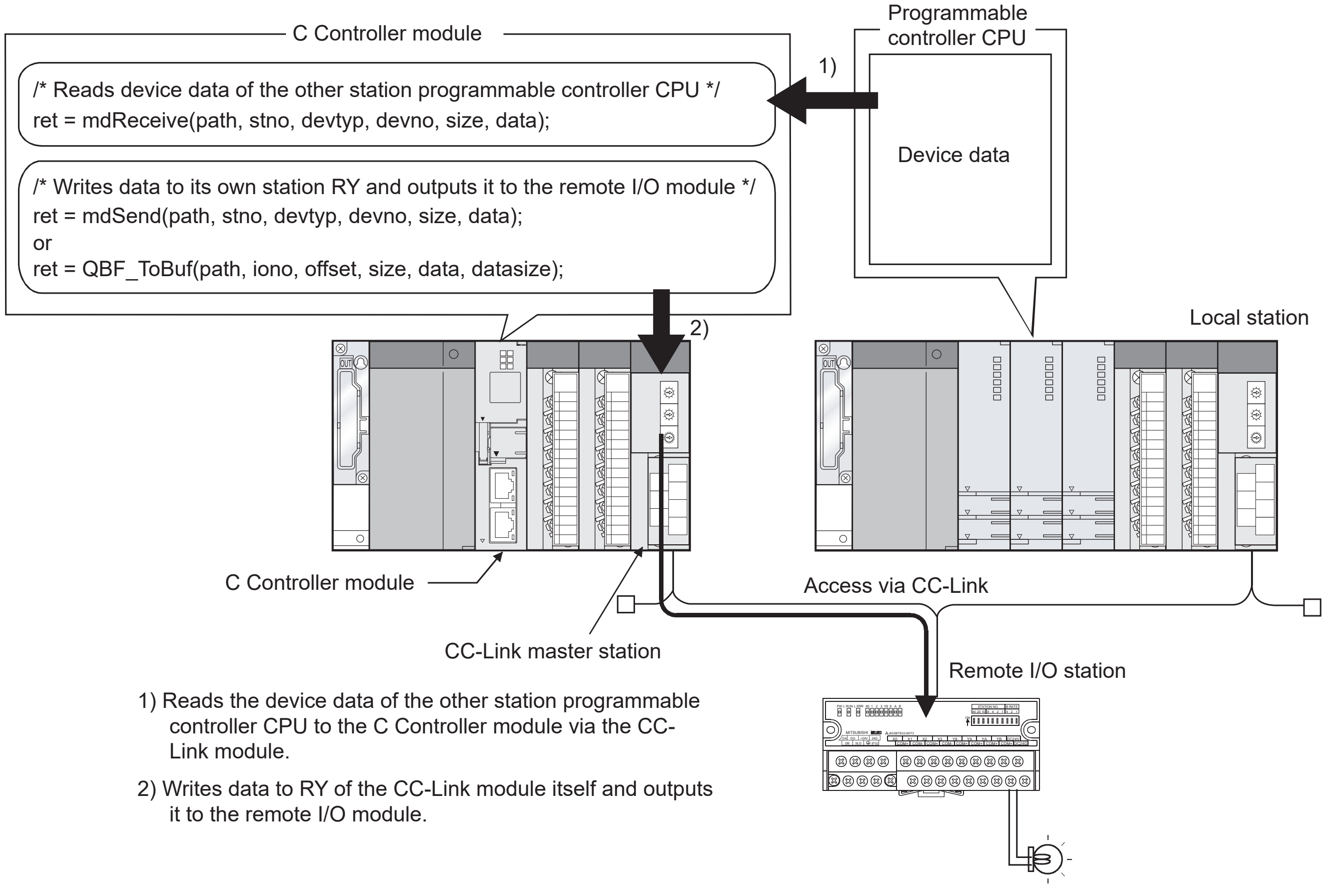
In a multiple CPU system, access cannot be made via a CC-Link module controlled by another CPU.

To create user programs of the C Controller module, use MELSEC data link functions or bus interface functions.

For the Q06CCPU-V-B, MELSEC data link functions cannot be used.

When the Q06CCPU-V-B is used, use cyclic transmission for access to a CPU module on another station.

<Example>



```
/* Reads device data of the other station programmable controller CPU */
ret = mdReceive(path, stno, devtyp, devno, size, data);
```

```
/* Writes data to its own station RY and outputs it to the remote I/O module */
ret = mdSend(path, stno, devtyp, devno, size, data);
or
ret = QBF_ToBuf(path, iono, offset, size, data, datasize);
```

1) Reads the device data of the other station programmable controller CPU to the C Controller module via the CC-Link module.

2) Writes data to RY of the CC-Link module itself and outputs it to the remote I/O module.

Figure 5.1 Access via CC-Link

Remark

For the bus interface functions and MELSEC data link functions, refer to the following.

☞ C Controller Module User's Manual (Utility Operation, Programming)

5.2.1 Block data assurance per station

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	△

The block data assurance per station is a function with which integrity of cyclic data is ensured for each station.
Depending on the read/write timing, a block of cyclic data may be split into new and old data in double-word (32-bit) units.
Use of this function ensures data consistency for each slave station.

(1) Applicable module

The following shows the CC-Link module that supports the block data assurance per station function.

Table 5.2 Module that supports the block data assurance per station function

Model	Version
QJ61BT11N	First 5 digits of serial No. is 08032 or later.

(2) Accessible devices

The following devices are accessible, using the block data assurance per station function.

- Own station's RX
- Own station's RY
- Own station's link register (for sending)
- Own station's link register (for receiving)
- Own station's buffer memory

☒POINT

This function is effective for the ranges of the link refresh devices (buffer memory), which correspond to the stations set by the parameters (Station information settings) for the CC-Link master station.
For details of the link refresh devices (buffer memory) assigned to respective stations, refer to "Slave station offset, size information" (buffer memory address: 3E0H to 5DFH) in the buffer memory list for the CC-Link module.
For details of "Slave station offset, size information", refer to the following.
☞ MELSEC-Q CC-Link System Master/Local Module User's Manual

(3) How to read out cyclic data (from the buffer memory)

Cyclic data can be read out from the buffer memory using the block data assurance per station, by one of the methods described below.
If any other method is used, integrity of cyclic data for each slave station is not guaranteed.

1) Enable the bock data assurance per station in CC-Link utility.

2) Use either of the following methods.

Table 5.3 Reading out cyclic data (from the buffer memory)

Method	Description	See
Automatic cyclic data refresh	Only by executing the QBF_FromBuf function, data are automatically refreshed and read out (in the next link scan).	(a)
Manual cyclic data refresh	Data are manually refreshed with the QBF_RefreshLinkDevice function, and then they are read out by executing the QBF_FromBuf function.	(b)

(a) Automatic cyclic data refresh

1) Execute the QBF_FromBuf function with "automatic" specified for "CC-Link refresh method" of the third argument (ulOffset).

2) In the timing of the next link scan, refreshed cyclic data (in the buffer memory) are read out.
User program processing is suspended until the reading is completed.

3) Upon completion of reading, the user program processing resumes.

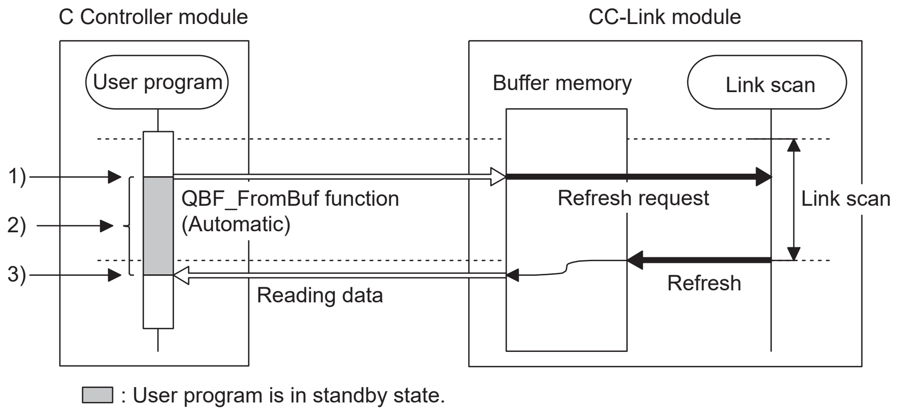


Figure 5.2 Readout operation

☒POINT

During execution of the QBF_FromBuf function, a time of up to one link scan may be spent for standby.

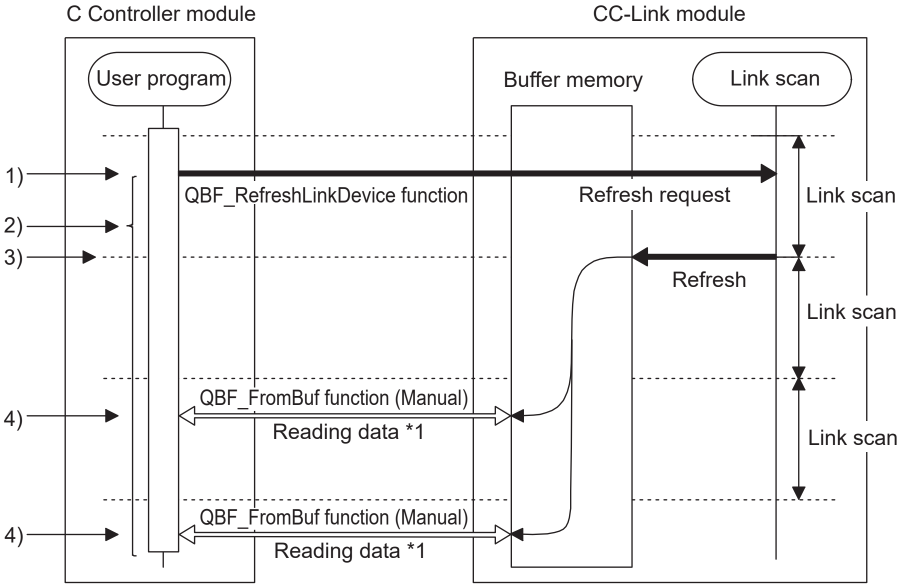
(b) Manual cyclic data refresh

1) Use the QBF_RefreshLinkDevice function to issue a refresh request.

2) The user program is continued.

3) Data are refreshed by the request in 1).

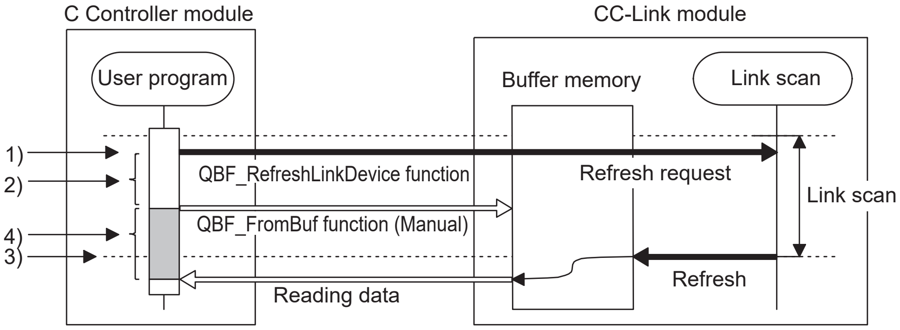
4) Execute the QBF_FromBuf function with "manual" specified for "CC-Link refresh method" of the third argument (ulOffset).
Cyclic data (in the buffer memory) are read out.



*1 The same refresh data are read out.

Figure 5.3 Readout operation

- If execution of the QBF_FromBuf function is attempted before refreshing data by the QBF_RefreshLinkDevice function, user program processing is suspended until completion of the refresh.
  Upon completion of the refresh, cyclic data (in the buffer memory) are read out, and the user program processing resumes.



▨ : User program is in standby state.

Figure 5.4 Readout operation

☒POINT

When any of the following operations is performed for the CC-Link module for which the block data assurance per station is enabled, cyclic data are automatically refreshed even if the QBF_RefleshLinkDevice function is not executed.

- Monitoring on the Module monitoring tab of C Controller setting utility
- Monitoring in Device monitoring utility
- Device reading from another user program (another task) with the QBF_FromBuf function (automatic refresh method) or a MELSEC data link function

Note that the write (send) area of the link refresh device (in the buffer memory) is not refreshed.
Also, cyclic data are not refreshed when monitoring is performed with another station's CC-Link module specified in Device monitoring utility.

Remark

Use the manual cyclic data refresh method if user program processing is to be continued during the waiting time which will occur in the automatic cyclic data refresh.

(4) How to write cyclic data (to the buffer memory)

Cyclic data can be written to the buffer memory using the block data assurance per station, by one of the methods described below.
If any other method is used, integrity of cyclic data for each slave station is not guaranteed.

1) Enable the bock data assurance per station in CC-Link utility.

2) Use either of the following methods.

Table 5.4 Writing cyclic data (to the buffer memory)

Method	Description	See
Automatic cyclic data refresh	Only by executing the QBF_ToBuf function, data are written and automatically refreshed (in the next link scan).	(a)
Manual cyclic data refresh	Data are written with the QBF_ToBuf function, and then they are manually refreshed by executing the QBF_RefreshLinkDevice function.	(b)

(a) Automatic cyclic data refresh

1) Execute the QBF_ToBuf function with "automatic" specified for "CC-Link refresh method" of the third argument (ulOffset).

2) In the timing of the next link scan, data are refreshed.

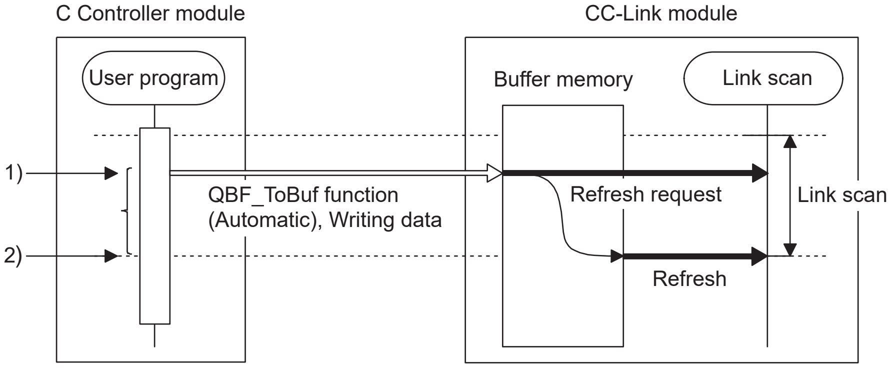


Figure 5.5 Write operation

- If execution of the QBF_ToBuf function is attempted before completion of the previous refresh, data writing is suspended until completion of the refresh.

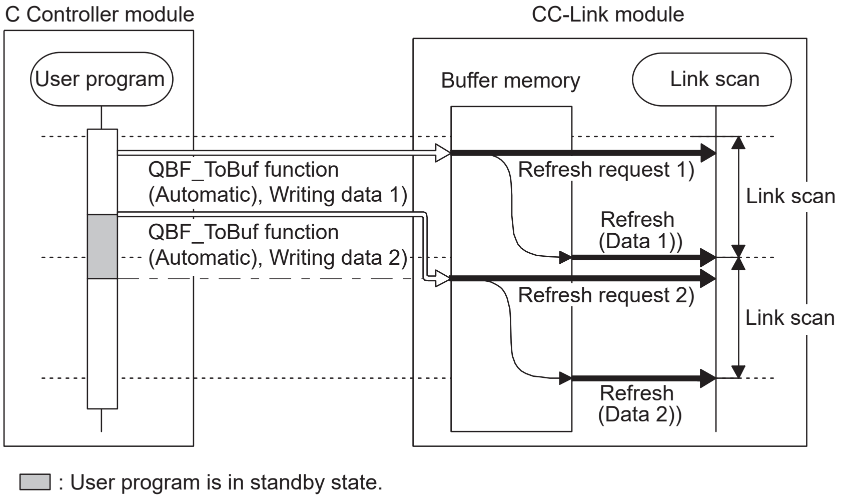


Figure 5.6 Write operation

☒POINT

Up to a time of up to one link scan may be required for refreshing the written data after execution of the QBF_ToBuf function.

(b) Manual cyclic data refresh

1) Execute the QBF_ToBuf function with "manual" specified for "CC-Link refresh method" of the third argument (ulOffset).

2) Use the QBF_RefreshLinkDevice function to issue a refresh request.

3) In the timing of the next link scan, refresh is performed.

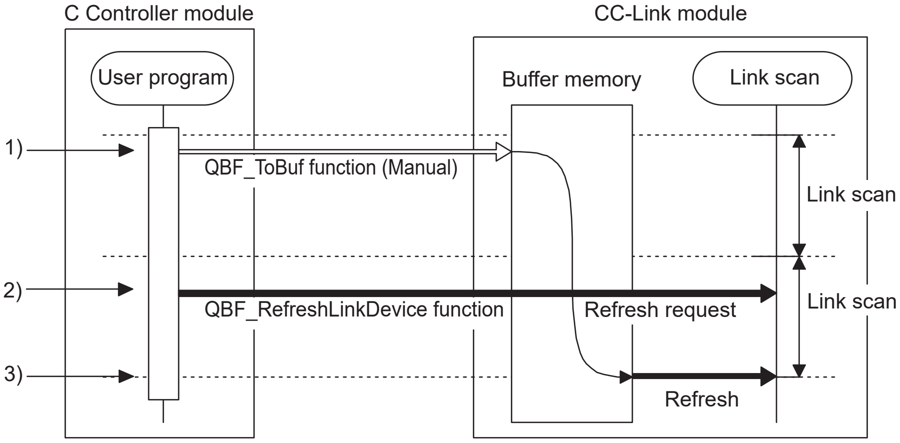
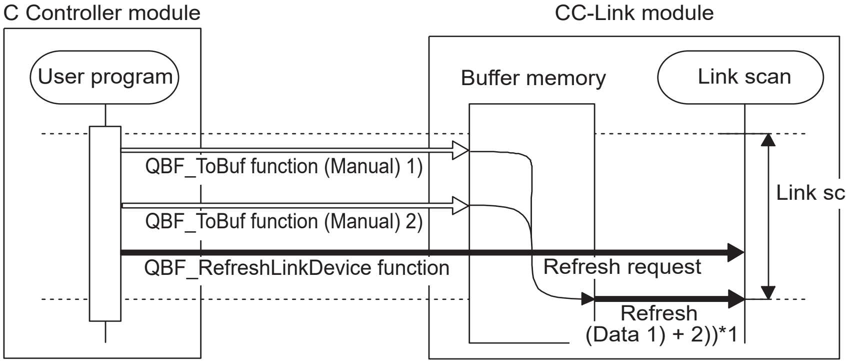


Figure 5.7 Write operation

- If data are written to the same address area before execution of the QBF_RefreshLinkDevice function, the area will be overwritten with the data refreshed later.
  By writing the data to another address area, both of the data can be written.



*1 Since writing these data to the same address overwrites the area in the buffer memory, newly written data are refreshed.
If written to different addresses, both data are refreshed.

Figure 5.8 Write operation

- If data writing is attempted before completion of the refresh after execution of the QBF_RefreshLinkDevice function, the data will be not written until completion of the previous refresh.

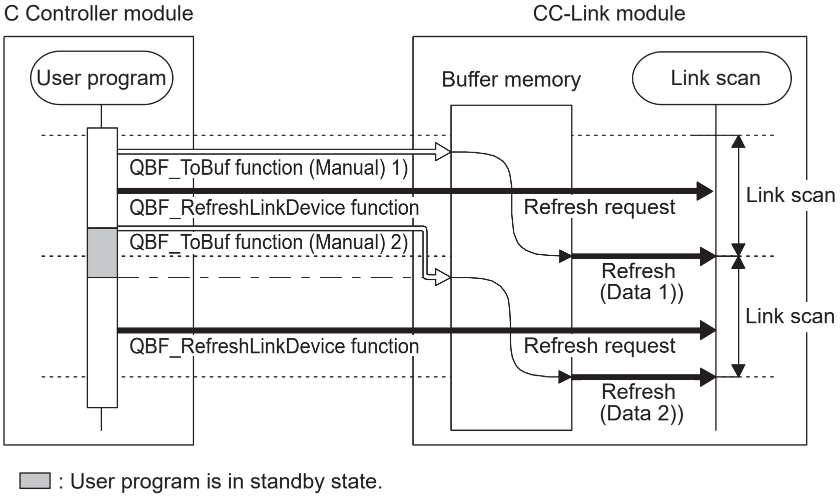

C Controller module
CC-Link module
User program
Buffer memory
Link scan
QBF_ToBuf function (Manual) 1)
QBF_RefreshLinkDevice function
Refresh request
Link scan
QBF_ToBuf function (Manual) 2)
Refresh
(Data 1))
Link scan
QBF_RefreshLinkDevice function
Refresh request
Refresh
(Data 2))
▭ : User program is in standby state.

Figure 5.9 Write operation

☒POINT

When any of the following operations is performed for the CC-Link module for which the block data assurance per station is enabled, cyclic data are automatically refreshed even if the QBF_RefleshLinkDevice function is not executed.

- Forced writing to the buffer memory on the Module monitoring tab of C Controller setting utility
- Device writing in Device monitoring utility
- Device writing from another user program (another task) with the QBF_ToBuf function (automatic refresh method) or a MELSEC data link function

Note that the read (receive) area of the link refresh device (in the buffer memory) is not refreshed.

Also, cyclic data are not refreshed when device writing is performed with another station's CC-Link module specified in Device monitoring utility.

Remark

For CC-Link utility, refer to the following.

☞ C Controller Module User's Manual (Utility Operation, Programming)

5.3 MELSECNET/H Module Access Function

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	△

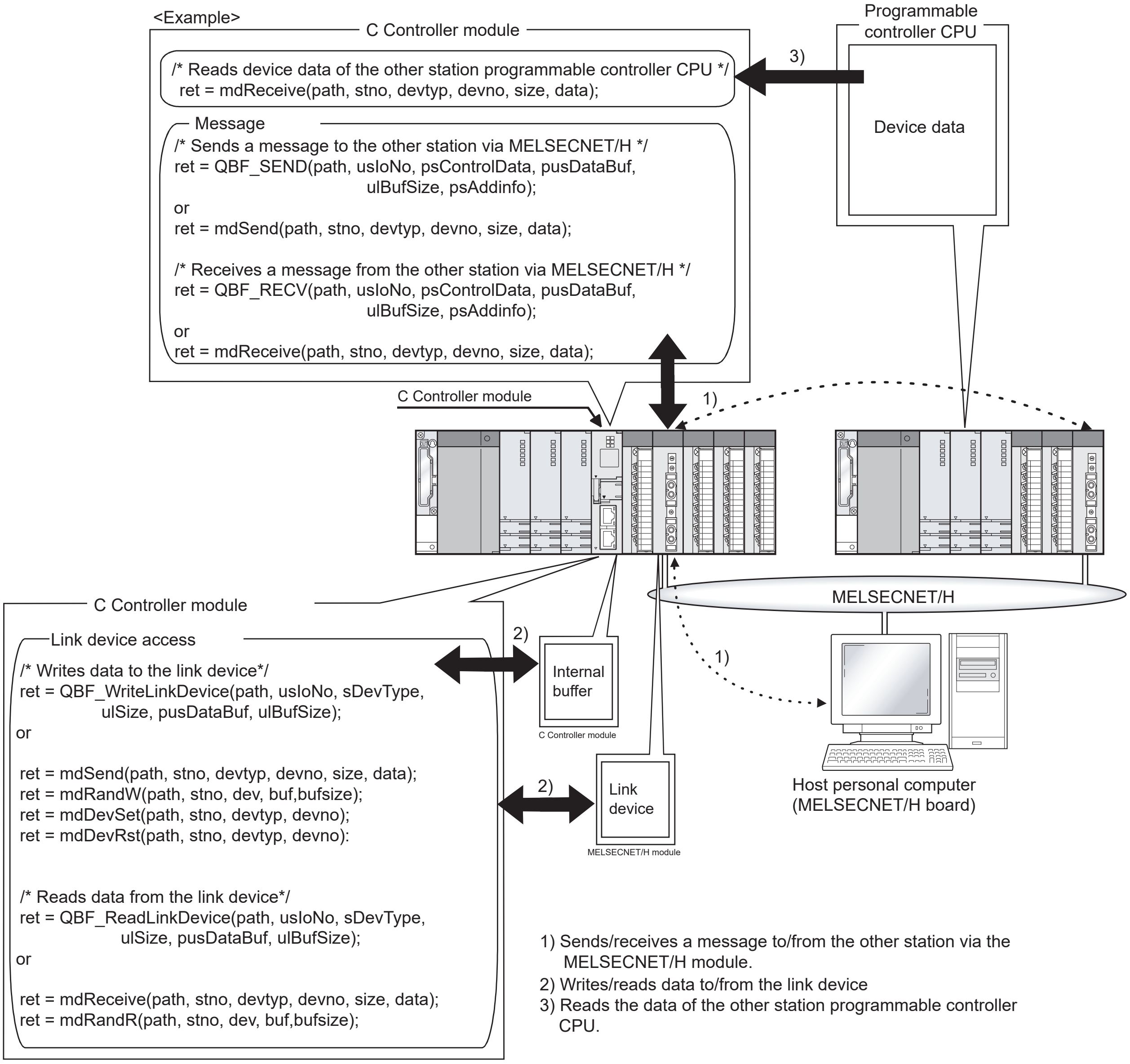
The MELSECNET/H module access function allows access to each station, programmable controller CPU on another station, or C Controller module connected to MELSECNET/H via MELSECNET/H modules.

In a multiple CPU system configuration, access cannot be made via a MELSECNET/H module controlled by another CPU.

To create user programs of the C Controller module, use MELSEC data link functions or bus interface functions.

For the Q06CCPU-V-B, MELSEC data link functions are not available.

When the Q06CCPU-V-B is used, use cyclic transmission for access to a CPU module on another station.



```
/* Reads device data of the other station programmable controller CPU */
  ret = mdReceive(path, stno, devtyp, devno, size, data);
```

Message

```
/* Sends a message to the other station via MELSECNET/H */
ret = QBF_SEND(path, usIoNo, psControlData, pusDataBuf,
                        ulBufSize, psAddinfo);
or
ret = mdSend(path, stno, devtyp, devno, size, data);

/* Receives a message from the other station via MELSECNET/H */
ret = QBF_RECV(path, usIoNo, psControlData, pusDataBuf,
                        ulBufSize, psAddinfo);
or
ret = mdReceive(path, stno, devtyp, devno, size, data);
```

Link device access

```
/* Writes data to the link device*/
ret = QBF_WriteLinkDevice(path, usIoNo, sDevType,
              ulSize, pusDataBuf, ulBufSize);
or

ret = mdSend(path, stno, devtyp, devno, size, data);
ret = mdRandW(path, stno, dev, buf,bufsize);
ret = mdDevSet(path, stno, devtyp, devno);
ret = mdDevRst(path, stno, devtyp, devno):


/* Reads data from the link device*/
ret = QBF_ReadLinkDevice(path, usIoNo, sDevType,
              ulSize, pusDataBuf, ulBufSize);
or

ret = mdReceive(path, stno, devtyp, devno, size, data);
ret = mdRandR(path, stno, dev, buf,bufsize);
```

1) Sends/receives a message to/from the other station via the MELSECNET/H module.
2) Writes/reads data to/from the link device
3) Reads the data of the other station programmable controller CPU.

Figure 5.10 MELSECNET/H access function

5.3.1 Message communication

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	△

The C Controller module can transfer messages (message transmission/reception) by user programs.
Messages can be transferred among the following modules via a MELSECNET/H module.
In a multiple CPU system configuration, any access via a MELSECNET/H module controlled by another CPU is not allowed.

- C Controller module on another station
- Programmable controller CPU
- Personal computer with a MELSECNET/H board

(1) Message transmission

Messages are sent with user programs in which communication functions for message transmission are used.

(2) Message reception

Messages are received with user programs in which communication functions for message reception are used.
The C Controller module has an internal buffer for message reception.
The data stored in this internal receive buffer are read out.

(a) Internal buffer for message reception

1) The internal buffer for message reception stores the data received from the MELSECNET/H module controlled by the C Controller module.

2) The internal receive buffer has the areas for four MELSECNET/H modules, and up to 128 received data can be stored in each area.
If data are received when the buffer area has been full of received data, they will be discarded instead of being stored in the buffer.
Therefore, read received data before the number of the data stored in the internal receive buffer exceeds 128.

3) In message reception, data of the channel specified in the message receiving function are retrieved from the data stored in the internal buffer.

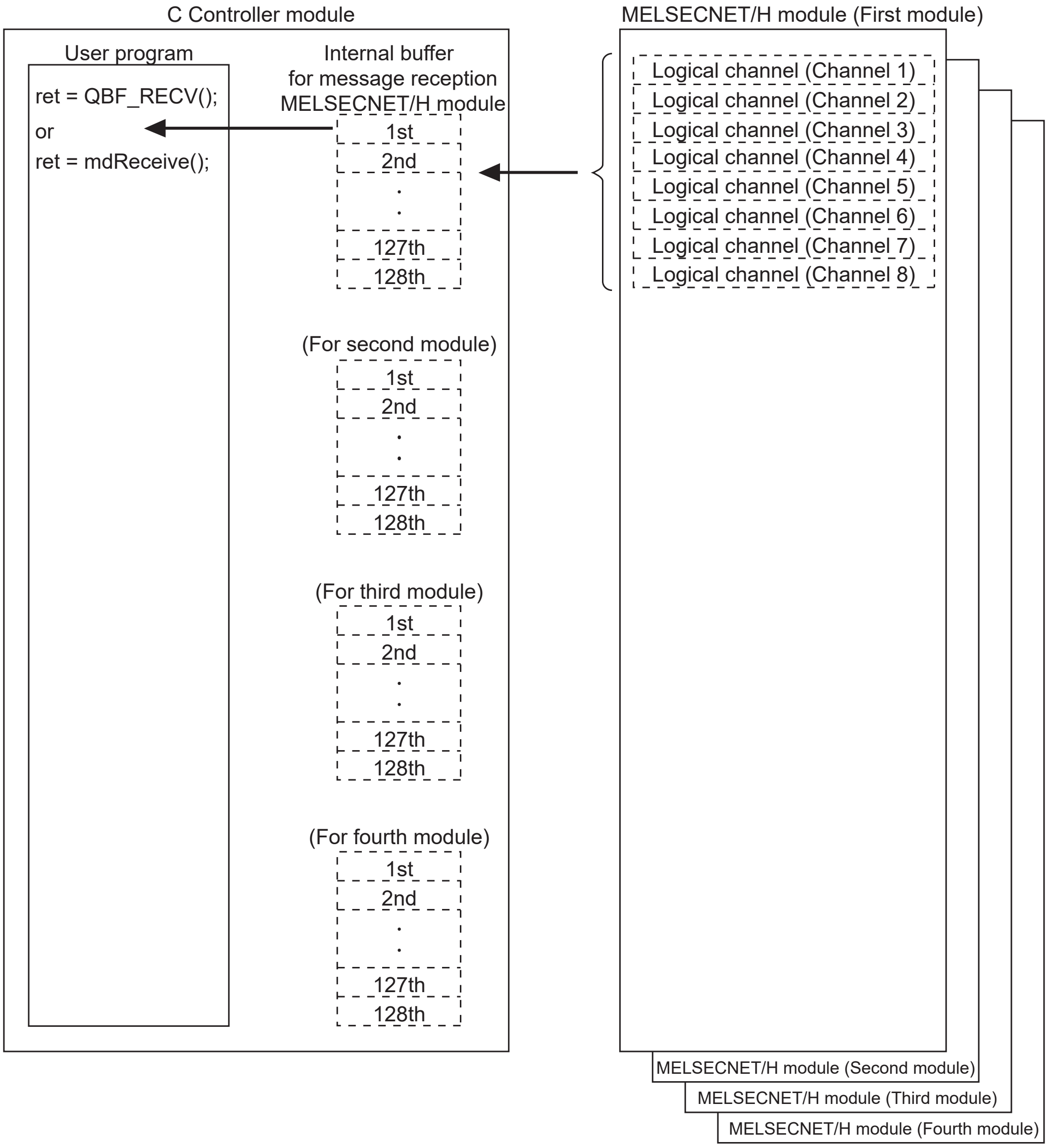

C Controller module
User program
ret = QBF_RECV();
or
ret = mdReceive();
Internal buffer
for message reception
MELSECNET/H module
1st
2nd
127th
128th
(For second module)
1st
2nd
127th
128th
(For third module)
1st
2nd
127th
128th
(For fourth module)
1st
2nd
127th
128th
MELSECNET/H module (First module)
Logical channel (Channel 1)
Logical channel (Channel 2)
Logical channel (Channel 3)
Logical channel (Channel 4)
Logical channel (Channel 5)
Logical channel (Channel 6)
Logical channel (Channel 7)
Logical channel (Channel 8)
MELSECNET/H module (Second module)
MELSECNET/H module (Third module)
MELSECNET/H module (Fourth module)

Figure 5.11 Internal buffer for message reception

Remark

For the bus interface functions and MELSEC data link functions, refer to the following manual.

☞ C Controller Module User's Manual (Utility Operation, Programming)

(3) Message communication example

The following a message communication example.

1) The C Controller module sends a message using a message sending function in the user program.

2) Station No.4 sends the message, by the message sending function in the user program and using channel 3, to the target station data storage channel 5 of station No.5 (logical channel 5).

3) The C Controller module (another station) automatically checks the message stored in station No.5, and stores the message into its own internal receive buffer.

4) The C Controller module (another station) reads the message from the internal receive buffer using a message receiving function.

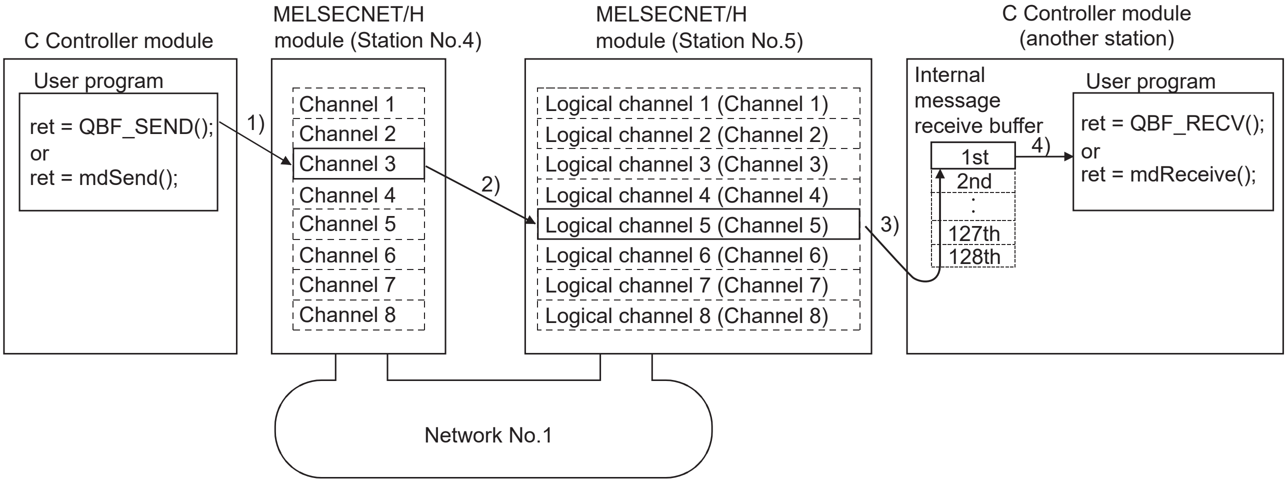
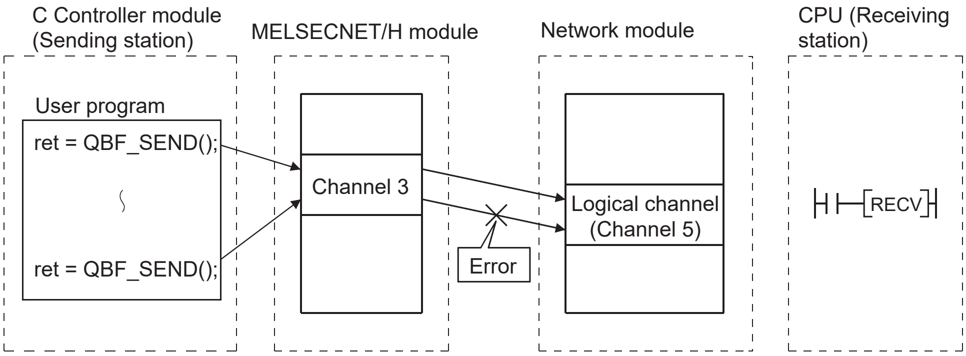


Figure 5.12 Message communication example

☒POINT

Pay attention to the following when sending a message to a MELSECNET/H module controlled by a programmable controller CPU.

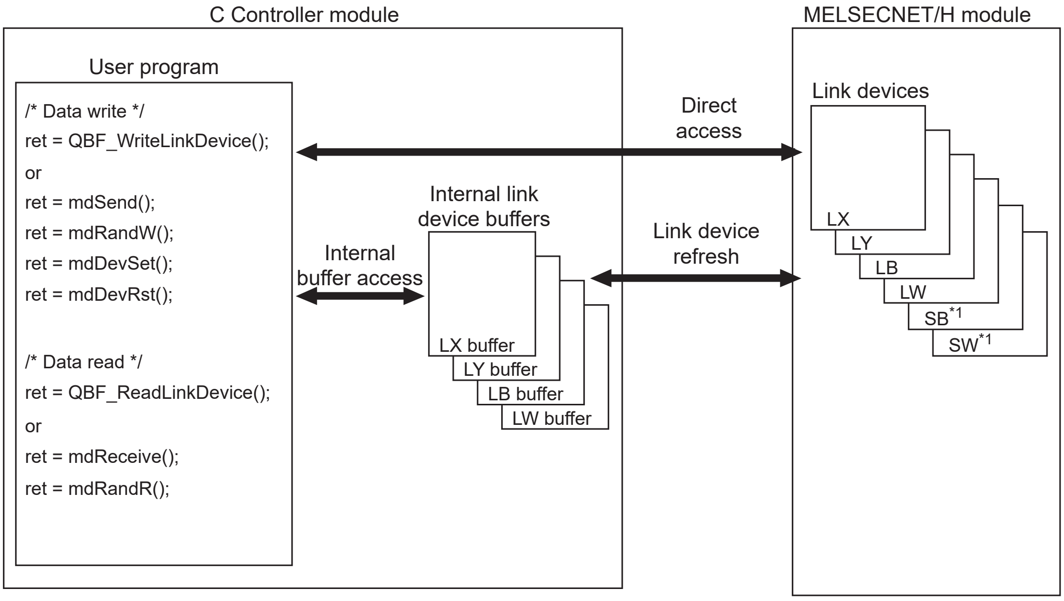
When sending data twice to the same channel of the receiving station with ACK set, execute the second transmission after the receiving station has read out the first data by the RECV instruction.
Failure to do so will cause an error. If an error is detected, retry the transmission after a little while.



5.3.2 Link device access

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	△

Link device data of a MELSECNET/H module controlled by a C Controller module can be read or written from a user program on the C Controller module.
To access from the user program to the link devices, two different methods are available: the internal buffer access and the direct access.



```
/* Data write */
ret = QBF_WriteLinkDevice();
or
ret = mdSend();
ret = mdRandW();
ret = mdDevSet();
ret = mdDevRst();

/* Data read */
ret = QBF_ReadLinkDevice();
or
ret = mdReceive();
ret = mdRandR();
```

*1 Link device refresh is not available for the link devices SB/SW. Use the direct access.

Figure 5.13 Link device access

(1) Internal buffer access

(a) What is the internal buffer access?
The internal buffer access is a method of accessing the internal link device buffers in the C Controller module.
To use this method, the refresh parameters must be previously set in the <<Parameter settings>> tab of MELSECNET/H utility.

(b) Internal link device buffers

1) Internal link device buffers

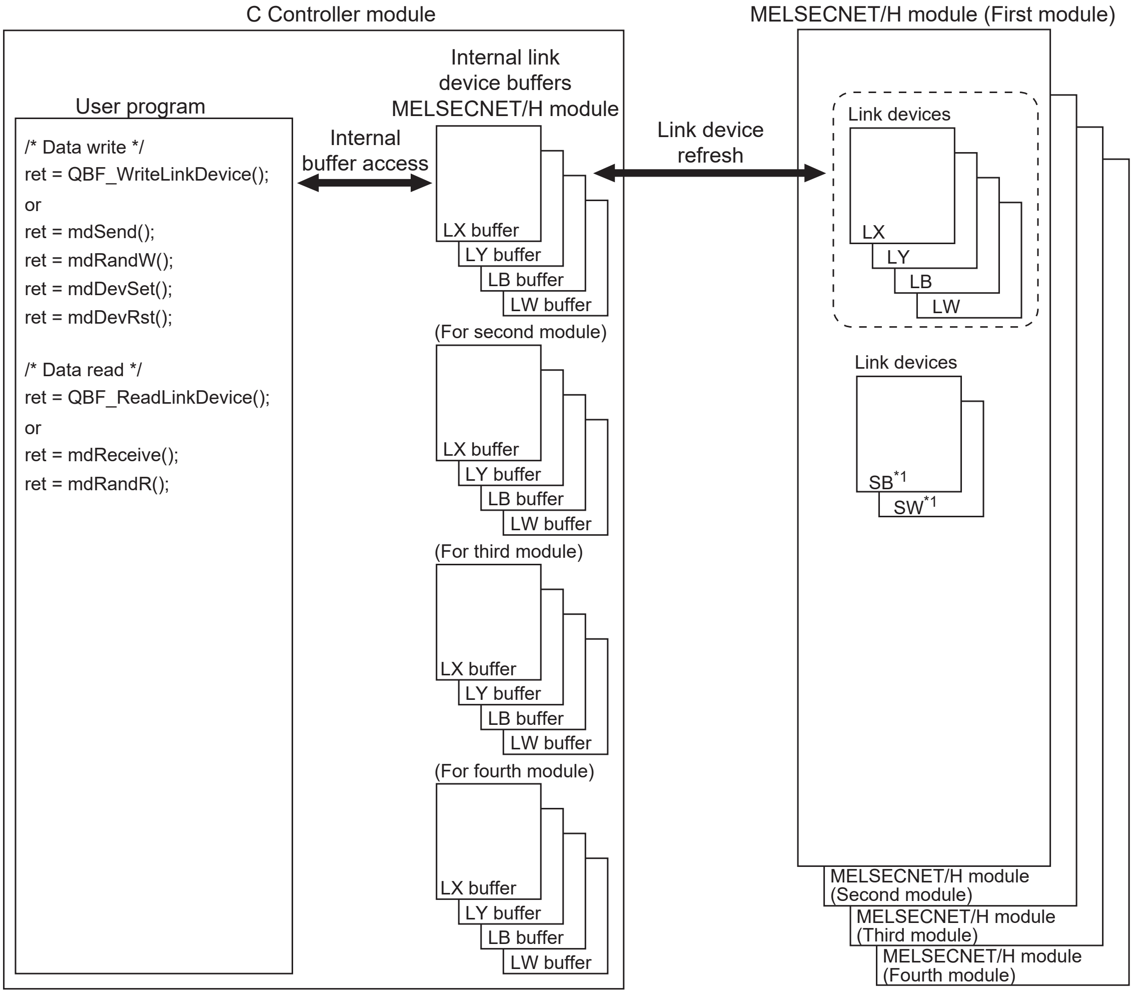
- The internal link device buffer is reserved for each of four MELSECNET/H modules.
- Each of the internal link device buffers (LX, LY, LB, and LW buffers) has the same area size as the corresponding link device (LX, LY, LB, and LW). Set refresh ranges on the Refresh parameter settings screen activated from the <<Parameter settings>> tab of MELSECNET/H utility.
- The internal link device buffers are refreshed with the link device data by link device refresh.

2) Accessible devices

The following devices are available for internal link device buffer access.

Table 5.5 List of accessible devices

Accessible device	No. of points	Address range
LX buffer	8192	0000_H to $1FFF_H$
LY buffer	8192	0000_H to $1FFF_H$
LB buffer	16384	0000_H to $3FFF_H$
LW buffer	16384	0000_H to $3FFF_H$



*1 Link device refresh is not available for the link devices SB/SW. Use the direct access.

Figure 5.14 Internal link device buffer access

(c) Link device refresh

Data are refreshed between the link devices and internal link device buffers in the specified link device refresh cycles and with refresh parameters set in MELSECNET/H utility.
(☞ Page 5-21, Section 5.3.4)

This refresh is performed in the areas that are set by refresh parameters and also set in Network range assignment in the <<Parameter settings>> tab of MELSECNET/H utility.

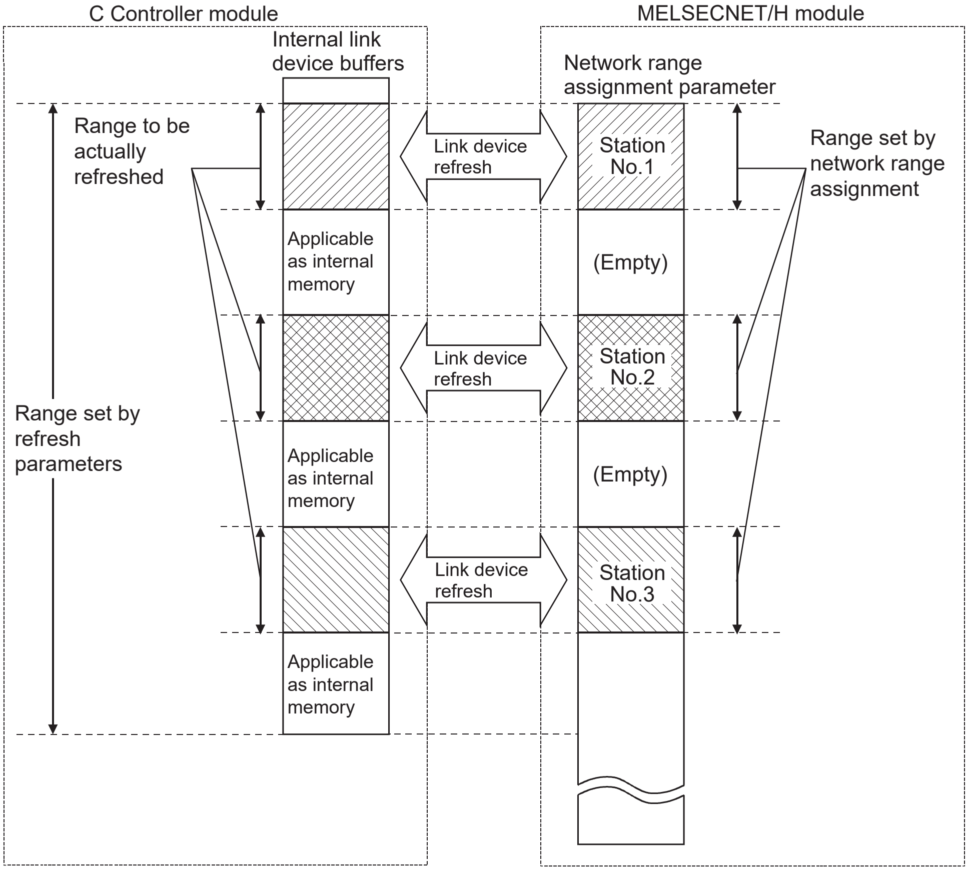

C Controller module
MELSECNET/H module
Internal link device buffers
Network range assignment parameter
Range to be actually refreshed
Range set by network range assignment
Link device refresh
Station No.1
Applicable as internal memory
(Empty)
Link device refresh
Station No.2
Range set by refresh parameters
Applicable as internal memory
(Empty)
Link device refresh
Station No.3
Applicable as internal memory

Figure 5.15 Link device refresh range

☒POINT

An internal link device buffer, which is not used for link device refresh, can be utilized as an internal memory of a user program.

(2) Direct access

(a) What is direct access?

The direct access is a method of directly accessing a link device of a MELSEC/H module from a user program on the C Controller module, independently of the link device refresh.

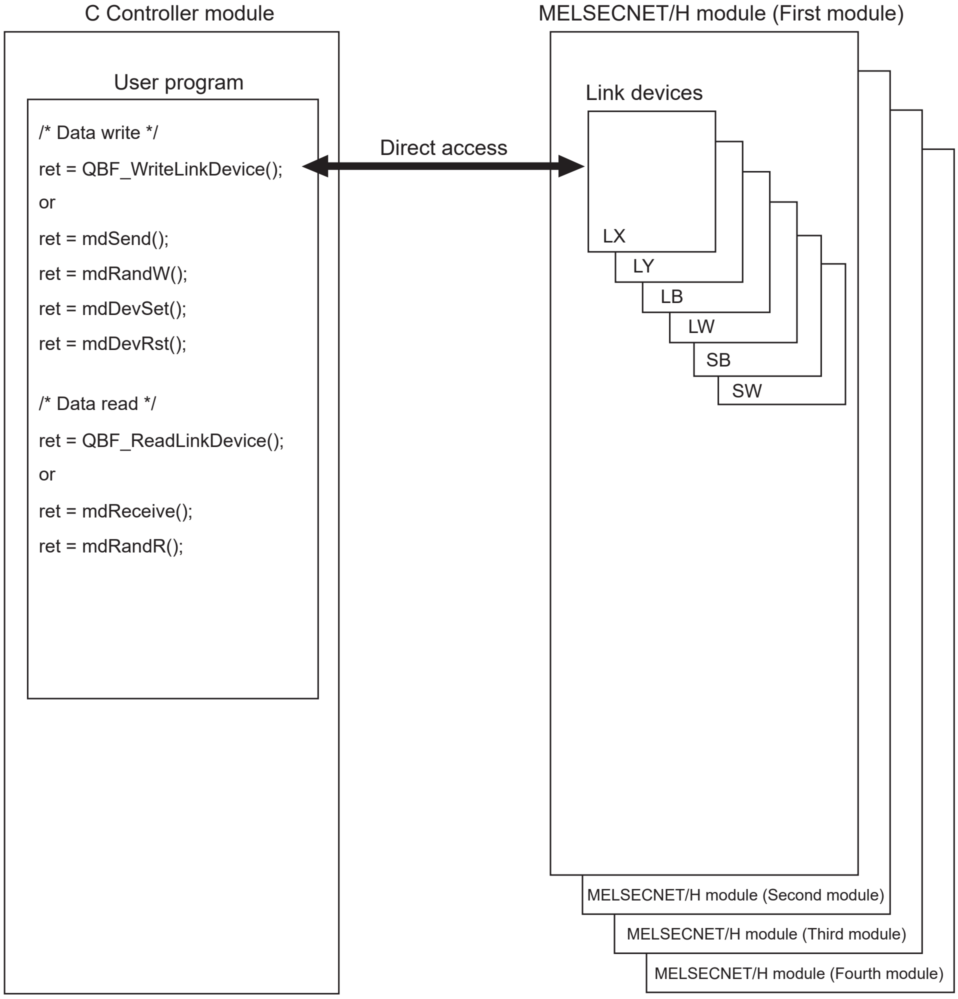
Since direct access does not use any internal link device buffer, the link device refresh cycle and refresh parameters need not be set in MELSECNET/H utility.

(b) Directly accessible devices

The following devices are directly accessible.

Table 5.6 Directly accessible device list

Access method	Accessible device
Direct access	LX
	LY
	LB
	LW
	SB
	SW



```
/* Data write */
ret = QBF_WriteLinkDevice();
or
ret = mdSend();
ret = mdRandW();
ret = mdDevSet();
ret = mdDevRst();

/* Data read */
ret = QBF_ReadLinkDevice();
or
ret = mdReceive();
ret = mdRandR();
```

C Controller module
User program
Direct access
MELSECNET/H module (First module)
Link devices
LX
LY
LB
LW
SB
SW
MELSECNET/H module (Second module)
MELSECNET/H module (Third module)
MELSECNET/H module (Fourth module)

Figure 5.16 Direct access

(c) Selection of internal buffer access or direct access
Select a suitable access method referring to the following.

1) Data suitable for internal buffer access
If a link device is used frequently or the block data assurance per station is needed for a link device, the data of the link device are suitable for internal buffer access.
For this kind of data, set refresh parameters to utilize them for internal buffer access.

2) Data suitable for direct access
Data of a link device that is not used so frequently are suitable for direct access.
Exclude these data from the refresh parameter setting, and utilize them for direct access.

☒POINT

To enable the block data assurance per station, set as follows.

1) Select "Block send data assurance per station" or "Block receive data assurance per station" in Parameter settings (Supplementary settings) for MELSECNET/H module (control station).
This Parameter settings (Supplementary settings) can be configured in MELSECNET/H utility when the control CPU of the MELSECNET/H module (control station) is a C Controller module, or in GX Works2/GX Developer when the control CPU is a programmable controller CPU.
2) In the <<Parameter settings>> tab of MELSECNET/H utility, set 10 or higher for "Link device refresh cycle" and configure "Refresh parameters" settings.
If 0 is set for "Link device refresh cycle" with "Block data assurance per station" enabled, cyclic transmission will not be performed properly and a data link error may occur.

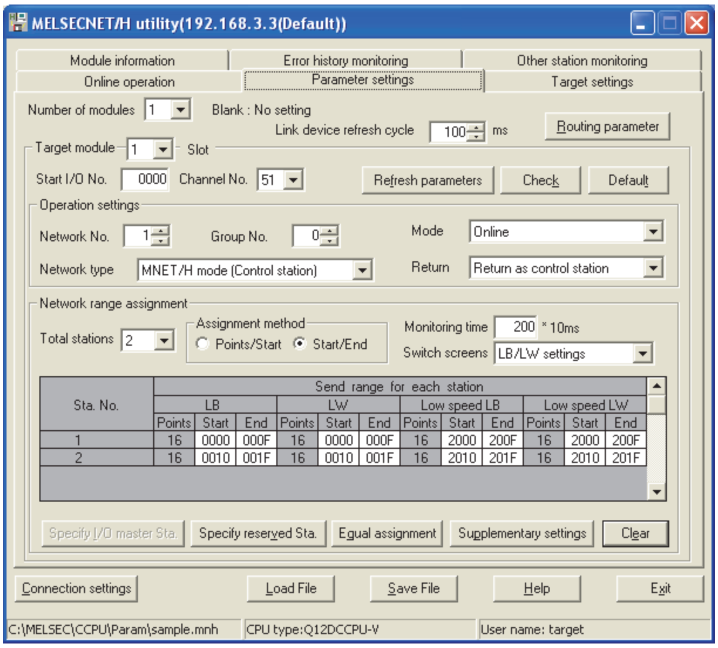
Remark

For MELSECNET/H utility, bus interface functions, and MELSEC data link functions, refer to the following.
☞ C Controller Module User's Manual (Utility Operation, Programming)

5.3.3 Parameter settings

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	○

Set parameters in the <<Parameter settings>> tab of MELSECNET/H utility.
(☞ C Controller Module User's Manual (Utility Operation, Programming))



Number of modules: 1

Blank : No setting

Link device refresh cycle: 100 ms

Target module: 1 Slot

Start I/O No.: 0000

Channel No.: 51

Operation settings

Network No.: 1

Group No.: 0

Mode: Online

Network type: MNET/H mode (Control station)

Return: Return as control station

Network range assignment

Total stations: 2

Assignment method: ○ Points/Start ● Start/End

Monitoring time: 200 * 10ms

Switch screens: LB/LW settings

Sta. No.	Send range for each station											
	LB			LW			Low speed LB			Low speed LW		
	Points	Start	End	Points	Start	End	Points	Start	End	Points	Start	End
1	16	0000	000F	16	0000	000F	16	2000	200F	16	2000	200F
2	16	0010	001F	16	0010	001F	16	2010	201F	16	2010	201F

C:\MELSEC\CCPU\Param\sample.mnh | CPU type:Q12DCCPU-V | User name: target

Figure 5.17 Parameter settings tab

5.3.4 Link device refresh setting

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	○

Link device refresh setting is required when using the internal buffer access in user programs.
To enable the link device refresh, set a link device refresh cycle and refresh parameters in MELSECNET/H utility.

(1) Link device refresh cycle

The link device refresh cycle is an interval of time during which the internal link device buffers of the C Controller module and link devices of a MELSECNET/H module are refreshed.
The concept of the link device refresh cycle is shown below.

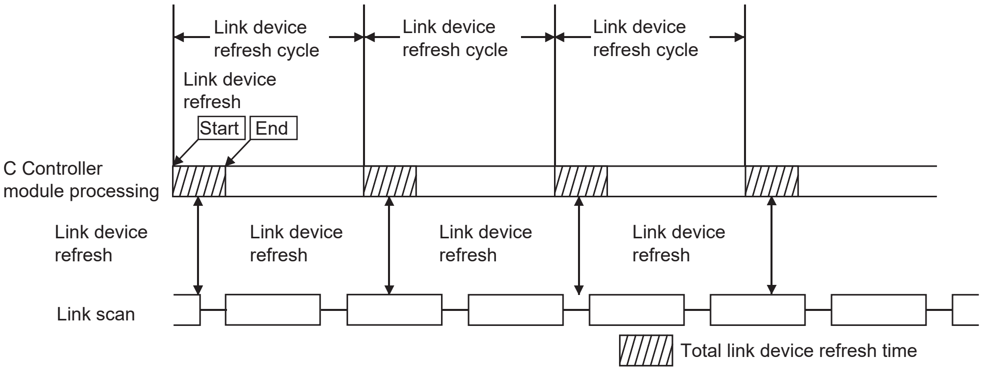

Link device refresh cycle
Link device refresh cycle
Link device refresh cycle
Link device refresh
Start
End
C Controller module processing
Link device refresh
Link device refresh
Link device refresh
Link device refresh
Link scan
▨ Total link device refresh time

Figure 5.18 Conceptual diagram of link device refresh cycle

(a) Total link device refresh time

1) What is total link device refresh time?
This is a processing time for refreshing the link devices of all MELSECNET/H modules that are controlled by the C Controller module.

2) How to obtain the total link device refresh time
A theoretical value for the total link device refresh time can be obtained by a calculation formula. (☞ Page 5-32, Section 5.3.5 (2))

3) Variation in total link device refresh time

The total link device refresh time varies depending on the operating conditions of the C Controller system.

It may take a longer time if the interrupt processing or high-priority task processing, such as the following, is activated during link device refresh.

- Shell command
- Workbench/Tornado connection
- File access
- Mount/unmount of a CompactFlash card
- Ethernet communications
- NFS server communication
- Processing of a bus interface driver[*1]

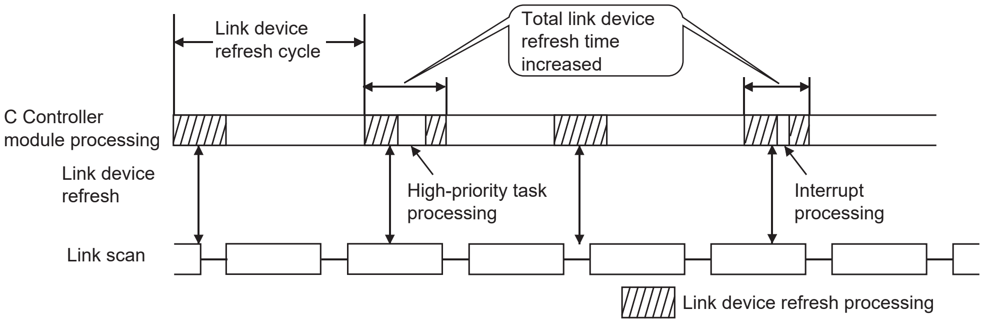

Link device refresh cycle
Total link device refresh time increased
C Controller module processing
Link device refresh
High-priority task processing
Interrupt processing
Link scan
▨ Link device refresh processing

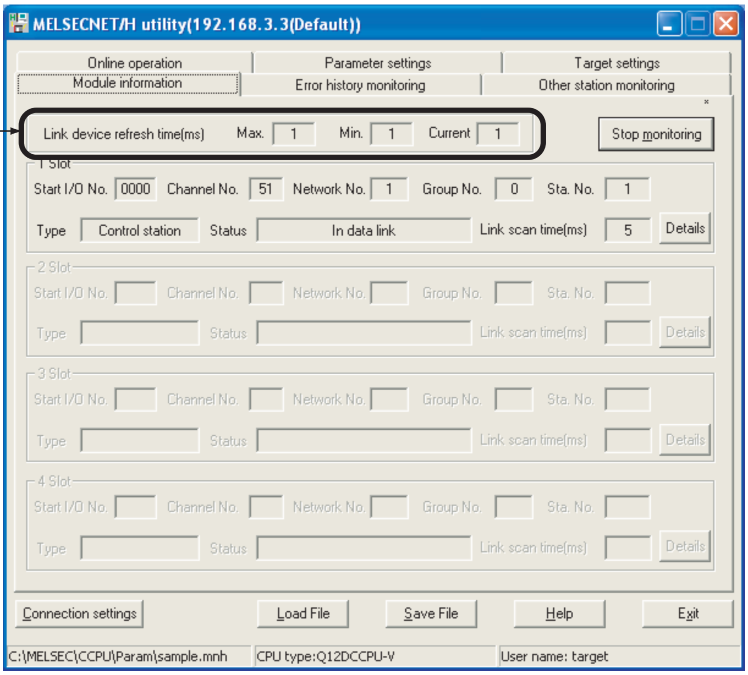
Figure 5.19 Variation in total link device refresh time

[*1] Functions to transmit with peripheral devices, functions to communicate with an intelligent function module, etc.

4) Checking the total link device refresh time

The total link device refresh time during operation of the C Controller system can be checked in "Link device refresh time (Max./Min./Current)" in the <<Module information>> tab of MELSECNET/H utility.

Total link device refresh time during C Controller system operation



Link device refresh time(ms) Max. 1 Min. 1 Current 1

1 Slot
Start I/O No. 0000 Channel No. 51 Network No. 1 Group No. 0 Sta. No. 1
Type Control station Status In data link Link scan time(ms) 5

2 Slot
Start I/O No. Channel No. Network No. Group No. Sta. No.
Type Status Link scan time(ms)

3 Slot
Start I/O No. Channel No. Network No. Group No. Sta. No.
Type Status Link scan time(ms)

4 Slot
Start I/O No. Channel No. Network No. Group No. Sta. No.
Type Status Link scan time(ms)

C:\MELSEC\CCPU\Param\sample.mnh CPU type:Q12DCCPU-V User name: target

Figure 5.20 Link device refresh time

(b) How to determine the link device refresh cycle
Take the steps according to the following flowchart.

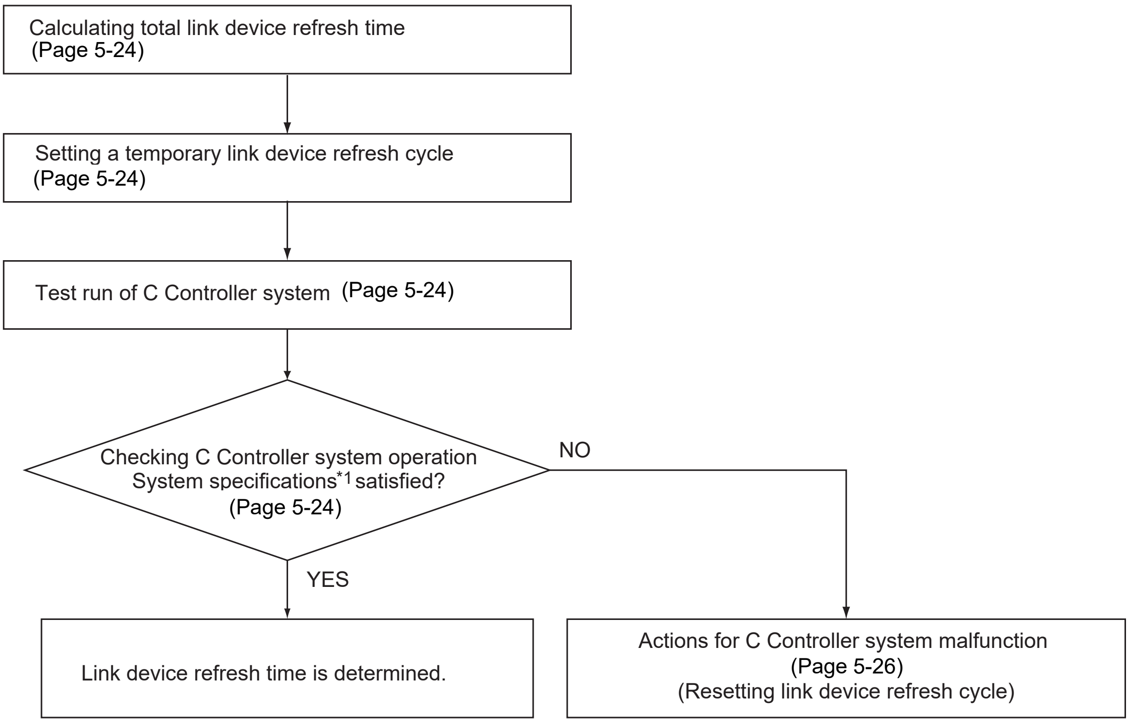

Calculating total link device refresh time (Page 5-24)
Setting a temporary link device refresh cycle (Page 5-24)
Test run of C Controller system (Page 5-24)
Checking C Controller system operation System specifications*1 satisfied? (Page 5-24)
NO
YES
Link device refresh time is determined.
Actions for C Controller system malfunction (Page 5-26) (Resetting link device refresh cycle)

Figure 5.21 Link device refresh cycle setting flowchart

* 1 The system specifications describe the whole system operations including the system functions and processing specifications required for the C Controller system, user program processing, functions, and performance specifications.

1) Calculating the total link device refresh time
Calculate the total link device refresh time.
(☞ Page 5-32, Section 5.3.5 (2))

2) Setting a temporary link device refresh cycle
The link device refresh cycle value and the total link device refresh time value obtained in 1) must satisfy the following relational expression.*1
Set a temporary value for the link device refresh cycle so that the expression is satisfied.

[Relational expression]
Total link device refresh time < Link device refresh cycle

* 1 When the C Controller module (own station) controls a MELSECNET/H module and a CC-Link IE controller network module, the following expression must be also satisfied, in consideration of the total link device refresh time for the MELSECNET/H and CC-Link IE controller networks.

[Relational expression]

$(\text{TLDRT}_{\text{MNETH}} + \text{TLDRT}_{\text{CC-Link IE Control}})$ < Link device refresh cycle

$\text{TLDRT}_{\text{MNETH}}$: Total link device refresh time for MELSECNET/H
$\text{TLDRT}_{\text{CC-Link IE Control}}$: Total link device refresh time for CC-Link IE controller network

3) Test run of the C Controller system
Perform the test run of the C Controller system to check if the system will operate properly with the link device refresh cycle that is temporarily set in 2).

4) Checking the C Controller system operation
In the test run of the C Controller system using the temporary link device refresh cycle setting, check the following 1 and 2.

[Check 1: Check for an error in the C Controller module.]
Check if any error has occurred in the C Controller module.
If the link device refresh is not completed within the specified link device refresh cycle, a "Link device refresh timeout error (error code: 5012, LNKREF.TIME OVER)" occurs in the C Controller module.
For this error, take corrective actions according to 5).

[Check 2: Check if the C Controller system operation meets the system specifications or not.]

Check if the C Controller system operation meets the system specifications or not.

Even if the error stated in Check 1 is not detected, normal operation of the C Controller system must be confirmed.

User program processing in the C Controller module may be suspended when the following is activated, depending on the operating conditions of the C Controller system.

- Link device refresh processing
- Interrupt processing
- High-priority task processing (such as the followings)
  - Shell command
  - Workbench/Tornado connection
  - File access
  - Mount/unmount of a CompactFlash card
  - Ethernet communications
  - NFS server communication
  - Processing of a bus interface driver*1

If the user program operation is interrupted longer then its operation time, the C Controller system specifications may not be satisfied.

If this occurs, take corrective actions according to 5).

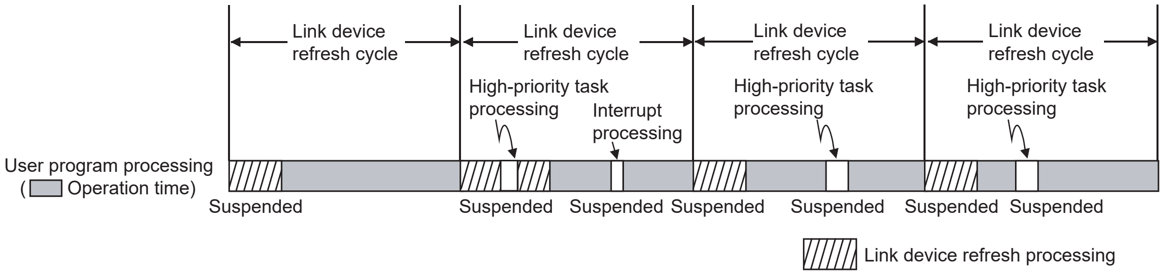

Link device refresh cycle
Link device refresh cycle
Link device refresh cycle
Link device refresh cycle
High-priority task processing
Interrupt processing
High-priority task processing
High-priority task processing
User program processing
(Operation time)
Suspended
Suspended
Suspended
Suspended
Suspended
Suspended
Suspended
Link device refresh processing

Figure 5.22 User program processing of C Controller module

* 1 Functions to transmit with peripheral devices, functions to communicate with an intelligent function module, etc.

5) Actions for C Controller system malfunction

For a C Controller system malfunction or faulty user program processing performance, take actions 1 to 3 if necessary to correct the system operation.

[Action 1: Increase the link device refresh cycle setting value.]

Check the total link device refresh time (Max.) measured during C Controller system operation, in "Link device refresh time (Max.)" in the <<Module information>> tab of either of the following utilities.

- MELSECNET/H utility
- CC IE Control utility

Re-set a link device refresh cycle appropriate to the confirmed link device refresh time (Max.) so that the following relational expression is satisfied.

Also, since the C Controller system specifications must be satisfied, determine the link device refresh cycle, taking into account the user program processing.

[Relational expression]
Measured link device refresh time (Max.) value < Link device refresh cycle

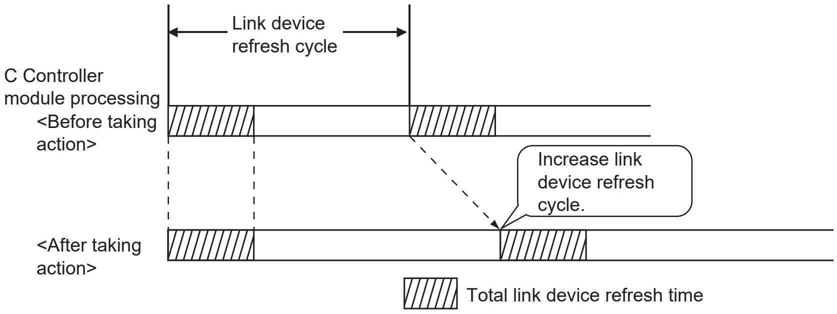

Link device refresh cycle
C Controller module processing
<Before taking action>
Increase link device refresh cycle.
<After taking action>
▨ Total link device refresh time

Figure 5.23 Corrective action 1

[Action 2: Decrease the refresh points.]
Check the refresh parameters and decrease the number of refresh points to reduce the total link device refresh time. (☞ Page 5-35, Section 5.3.5 (3))

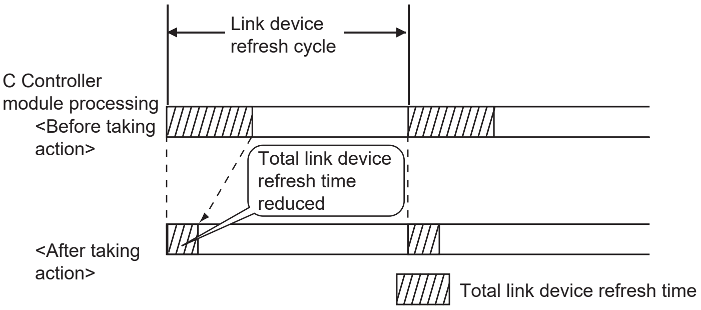
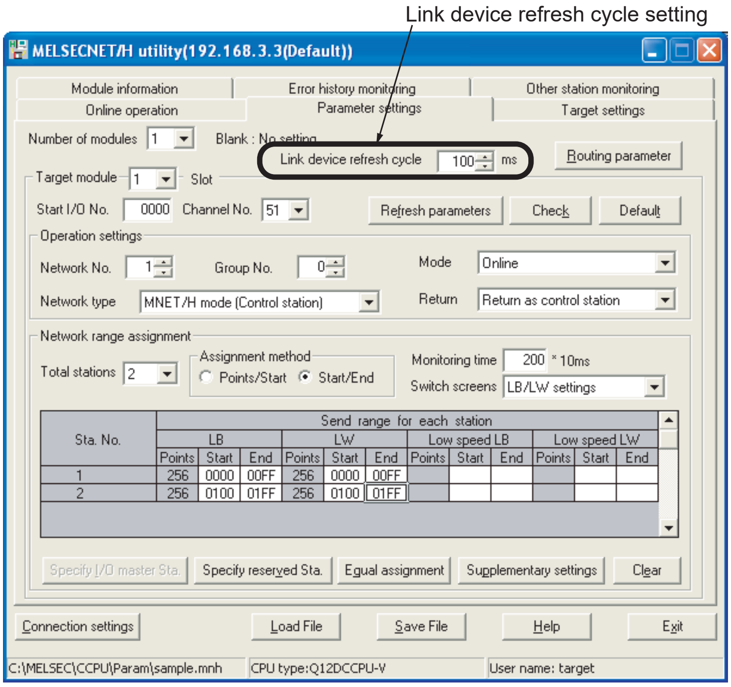

Link device refresh cycle
C Controller module processing
<Before taking action>
Total link device refresh time reduced
<After taking action>
▨ Total link device refresh time

Figure 5.24 Corrective action 2

[Action 3: Modify the user program.]
Modify the task configuration, priority, processing details of the user program so that the C Controller system specifications and user program processing performance can be satisfied.

(c) Link device refresh cycle setting
Set a link device refresh cycle in "Link device refresh cycle" in the <<Parameter settings>> tab.


Link device refresh cycle setting

MELSECNET/H utility(192.168.3.3(Default))

Number of modules: 1
Blank : No setting
Link device refresh cycle: 100 ms
Target module: 1 Slot
Start I/O No.: 0000
Channel No.: 51

Operation settings
Network No.: 1
Group No.: 0
Mode: Online
Network type: MNET/H mode (Control station)
Return: Return as control station

Network range assignment
Total stations: 2
Assignment method: ○ Points/Start ● Start/End
Monitoring time: 200 *10ms
Switch screens: LB/LW settings

Sta. No.	Send range for each station											
	LB			LW			Low speed LB			Low speed LW		
	Points	Start	End	Points	Start	End	Points	Start	End	Points	Start	End
1	256	0000	00FF	256	0000	00FF						
2	256	0100	01FF	256	0100	01FF						

C:\MELSEC\CCPU\Param\sample.mnh
CPU type:Q12DCCPU-V
User name: target

Figure 5.25 Link device refresh cycle setting

(2) Refresh parameters

The refresh parameters are parameters by which data are transferred from the link devices (LX, LY, LB, and LW) of a MELSECNET/H module to the internal link device buffers (LX, LY, LB and LW buffers) of the C Controller module for use in the user program.

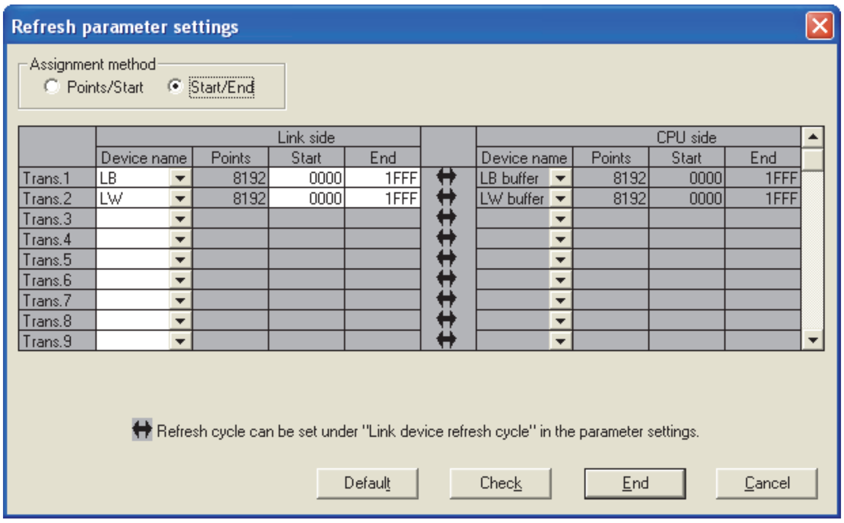
(a) Link refreshable devices

Up to 64 transfer settings are allowed for each MELSECNET/H module.
The following transfer settings are available.

- Transfer the link side device (LX) data to the CPU side device (LX buffer).
- Transfer the link side device (LY) data to the CPU side device (LY buffer).
- Transfer the link side device (LB) data to the CPU side device (LB buffer).
- Transfer the link side device (LW) data to the CPU side device (LW buffer).

(b) Setting refresh parameters

Set refresh parameters in the Refresh parameter settings screen activated from the <<Parameter settings>> tab of MELSECNET/H utility.



Refresh parameter settings

Assignment method
○ Points/Start ● Start/End

	Link side					CPU side			
	Device name	Points	Start	End		Device name	Points	Start	End
Trans.1	LB	8192	0000	1FFF	↔	LB buffer	8192	0000	1FFF
Trans.2	LW	8192	0000	1FFF	↔	LW buffer	8192	0000	1FFF
Trans.3					↔				
Trans.4					↔				
Trans.5					↔				
Trans.6					↔				
Trans.7					↔				
Trans.8					↔				
Trans.9					↔				

↔ Refresh cycle can be set under "Link device refresh cycle" in the parameter settings.

Figure 5.26 Refresh parameter settings screen

(3) Refresh parameter setting example

The following is a refresh parameter setting example.

(a) System configuration

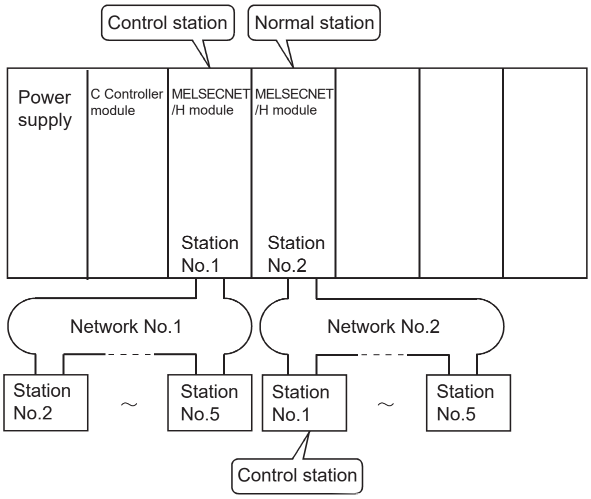


Figure 5.27 System configuration

(b) Parameter assignment

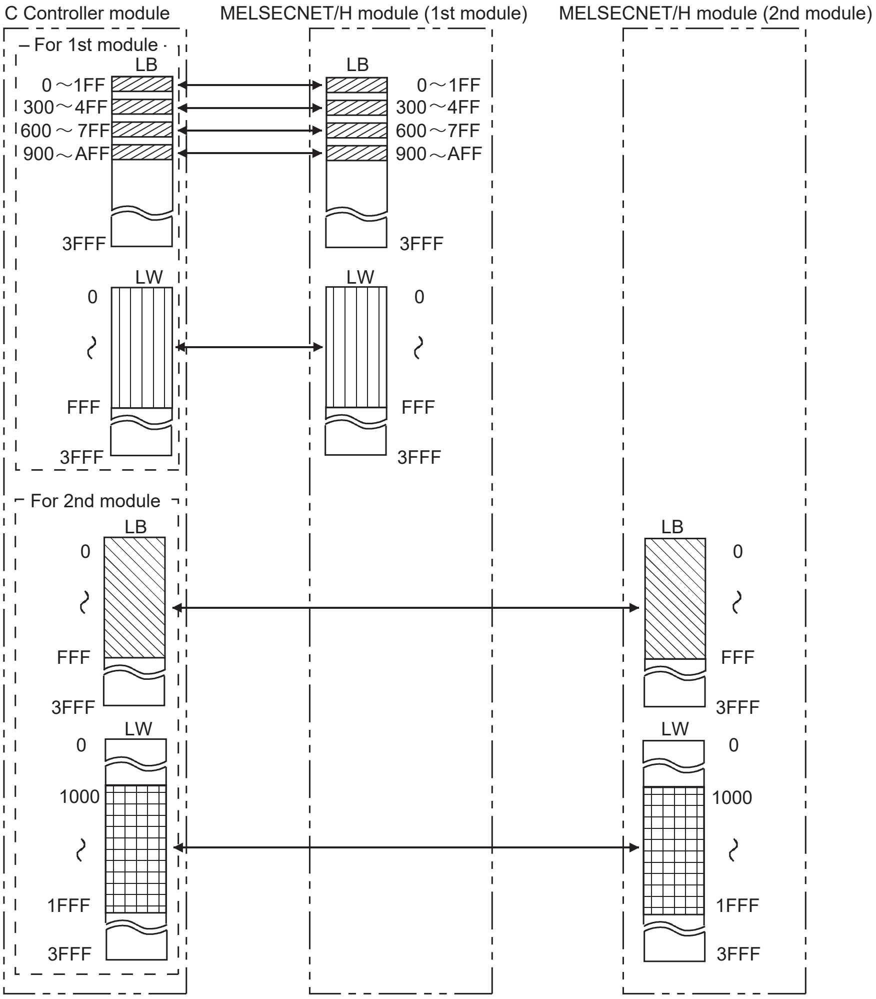
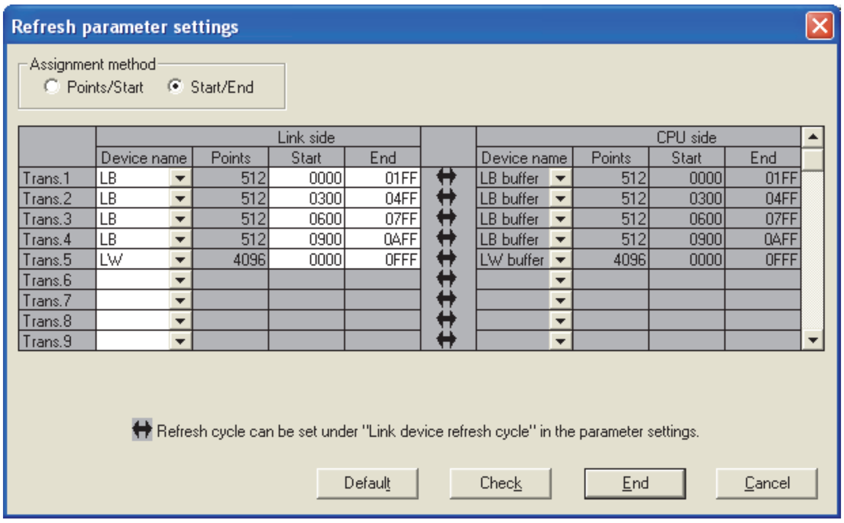


Figure 5.28 Parameter assignment

(c) MELSECNET/H utility: Refresh parameter settings screen

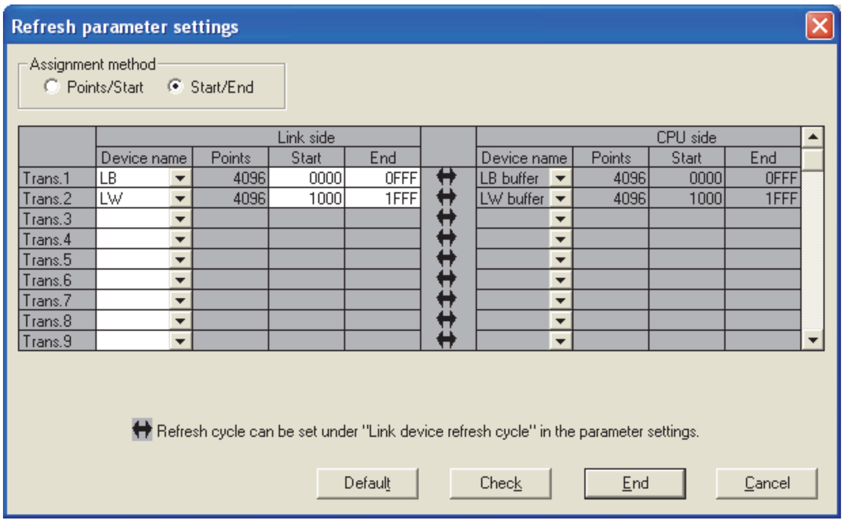

Refresh parameter settings

Assignment method
○ Points/Start ● Start/End

	Link side					CPU side			
	Device name	Points	Start	End		Device name	Points	Start	End
Trans.1	LB	512	0000	01FF	↔	LB buffer	512	0000	01FF
Trans.2	LB	512	0300	04FF	↔	LB buffer	512	0300	04FF
Trans.3	LB	512	0600	07FF	↔	LB buffer	512	0600	07FF
Trans.4	LB	512	0900	0AFF	↔	LB buffer	512	0900	0AFF
Trans.5	LW	4096	0000	0FFF	↔	LW buffer	4096	0000	0FFF
Trans.6					↔				
Trans.7					↔				
Trans.8					↔				
Trans.9					↔				

↔ Refresh cycle can be set under "Link device refresh cycle" in the parameter settings.

Figure 5.29 MELSECNET/H module (1st module) setting


Refresh parameter settings

Assignment method
○ Points/Start ● Start/End

	Link side					CPU side			
	Device name	Points	Start	End		Device name	Points	Start	End
Trans.1	LB	4096	0000	0FFF	↔	LB buffer	4096	0000	0FFF
Trans.2	LW	4096	1000	1FFF	↔	LW buffer	4096	1000	1FFF
Trans.3					↔				
Trans.4					↔				
Trans.5					↔				
Trans.6					↔				
Trans.7					↔				
Trans.8					↔				
Trans.9					↔				

↔ Refresh cycle can be set under "Link device refresh cycle" in the parameter settings.

Figure 5.30 MELSECNET/H module (2nd module) setting

Remark

For the MELSECNET/H utility, refer to the following.

☞ C Controller Module User's Manual (Utility Operation, Programming)

5.3.5 Link data transfer processing time specifications

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	○

This section explains how to calculate the link data transfer time and transmission delay time in the MELSECNET/H network system.

(1) Link data transfer processing

(a) Processing outline

Cyclic transmission uses LB/LW/LX/LY of a MELSECNET/H module.

The link relay (B) on the C Controller module side is explained here.

1) B0 in the LB buffer (sending side) is set to ON.

2) By a link device refresh, the B0 information in the LB buffer is stored in the link data storage area (LB) of the sending side MELSECNET/H module.

3) By a link scan, the B0 information is stored in the receiving side MELSECNET/H module's link data storage area (LB).

4) By a link device refresh, the B0 information in the link data storage area (LB) is stored in the LB buffer of the receiving side C Controller module.

5) B0 in the LB buffer (receiving side) is set to ON.

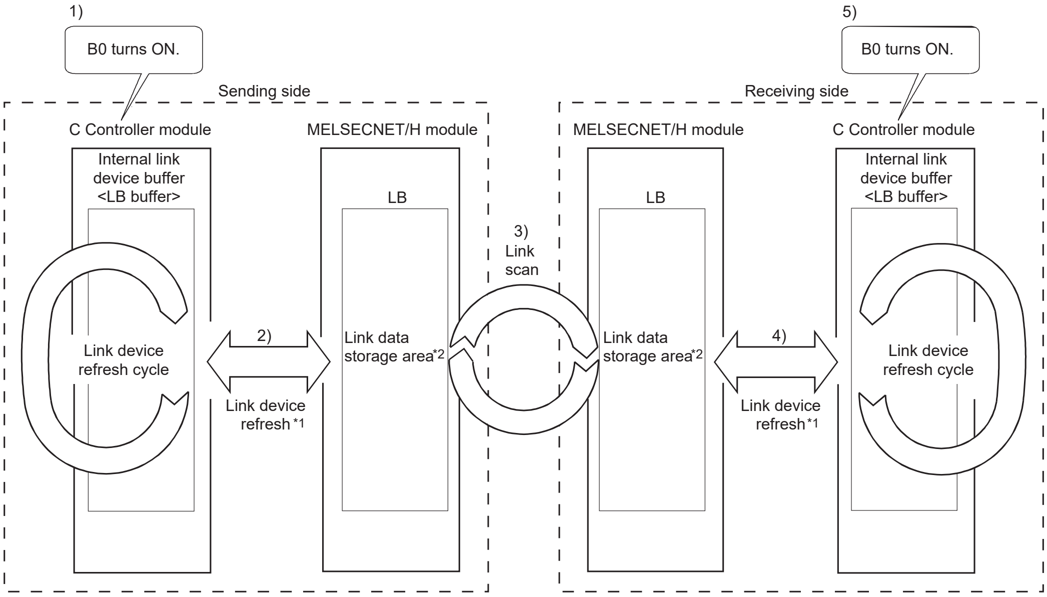


Figure 5.31 Link data transfer processing

* 1 Refresh parameters must be set for link device refreshes.

* 2 Network range assignment parameters must be set for link scans.

(b) Link scan and link device refresh

Link scans are NOT synchronized with link device refreshes of the C Controller module.

Link device refreshes are performed in the link device refresh cycles set for the C Controller module.

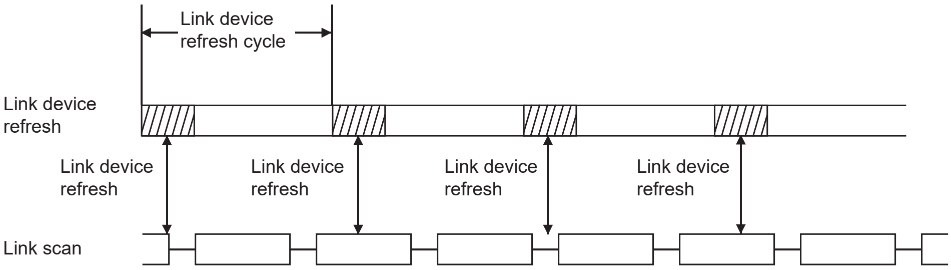


Figure 5.32 Link scan and link device refresh

(2) How to calculate the transmission delay time

(a) Delay time in transmission within the same network

1) Cyclic transmission (periodic LB/LW/LX/LY communications)

The transmission delay time in B/W/Y communications is calculated with the following.

- Link device refresh cycles of the sending and receiving side C Controller modules
- Link device refresh times of the sending and receiving side C Controller modules
- Scan time (excluding the link refresh time) of the programmable controller CPU (receiving side)
- Link refresh time of the programmable controller CPU (receiving side)
- Link scan time

The formulas are as follows:

[Transmission delay time (T_{D1}) of B/W/Y communications]

1) When a C Controller module (another CPU) receives data
(Until data are stored into internal link device buffer)

$$T_{D1} = L_T + \beta_T + (LS \times 1) + \beta_R + L_R \text{ [ms]}$$

2) When a programmable controller CPU receives data

$$T_{D1} = L_T + \beta_T + (LS \times 1) + (S_R + \alpha_R) \times 2 \text{ [ms]}$$

L_T : Link device refresh cycle of C Controller module (sending side)

L_R : Link device refresh cycle of C Controller module (receiving side)

β_T : Total link device refresh time of C Controller module (sending side)*1
☞ Page 5-33

β_R : Total link device refresh time of C Controller module (receiving side)*1
☞ Page 5-33

S_R : Scan time (excluding the link refresh time) of programmable controller CPU (receiving side)*2

α_R : Link refresh time of programmable controller CPU (receiving side)*1,*2

LS : Link scan time*2

[1] Total time for the number of mounted MELSECNET/H modules.

[2] For details, refer to the Q Corresponding MELSECNET/H Network System Reference Manual (PLC to PLC network).

The above calculation formulas are effective under the following conditions:

- No faulty stations
- No transient transmissions

2) Total link device refresh time (β_T, β_R)

The total link device refresh time can be calculated by the following formula, with the points assigned to link devices.

[Total link device refresh time]

$$\beta_T, \beta_R = KM1 + KM2 \times \left\{ \left(\frac{LB+LX+LY}{16} \right) + LW \right\} \text{[ms]}$$

β_T : Total link device refresh time (sending side)

β_R : Total link device refresh time (receiving side)

LB : Total points of link relay (LB) refreshed by the station[*1]

LW : Total points of link register (LW) refreshed by the station[*1]

LX : Total points of link input (LX) refreshed by the station[*1]

LY : Total points of link output (LY) refreshed by the station[*1]

KM1,KM2 : Constants

Table 5.7 KM1 and KM2 values

MELSECNET/H module location	KM1	KM2
Main base unit	1.0	0.43×10^{-3}
Extension base unit	1.0	0.95×10^{-3}

[1] Total of link device points within the range set by refresh parameters and set in the network range assignment. (Excluding the points assigned to reserved stations)
Total points for the MELSECNET/H modules that are controlled by the C Controller module.

(b) Transmission delay time calculation example
In a single system configuration, a delay time in cyclic transmission is calculated as described below.

Table 5.8 System configuration and conditions

Item		Description
CPU on receiving station		C Controller module
Total number of stations per network		2 (Control station : 1, Normal station : 1)
Total link device points	LB/LW	1024 points
	LX/LY	0 points
Link device refresh cycle		100ms
Transient transmission		None
MELSECNET/H module		Installed to Slot 0 of main base unit
Transmission speed		10Mbps

1) Link device refresh time

$$\text{Total link device refresh time} = KM1 + KM2 \times \left\{ \left(\frac{LB+LX+LY}{16} \right) + LW \right\}$$

· Total link device refresh time on sending side

$$\beta_T = 1.0 + 0.43 \times 10^{-3} \times \left(\frac{1024+0+0}{16} + 1024 \right)$$
$$\fallingdotseq 1.47(\text{ms})$$

· Total link device refresh time on receiving side

$$\beta_R \fallingdotseq 1.47(\text{ms})$$

2) Link scan time[*1]

$$\text{Link scan time } LS = KB + (0.45 \times \text{Total no. of stations}) + \left\{ \frac{LB + LY + (LW \times 16)}{8} \times 0.001 \right\}$$
$$= 4.0 + (0.45 \times 2) + \left\{ \frac{1024 + 0 + (1024 \times 16)}{8} \times 0.001 \right\}$$
$$\fallingdotseq 7.076 \text{ (ms)}$$

[*1] For details of the link scan time, refer to the Q Corresponding MELSECNET/H Network System Reference Manual (PLC to PLC network).

3) Cyclic transmission delay

$$\text{Transmission delay time} T_{D1} = L_T + \beta_T + (LS \times 1) + \beta_R + L_R$$
$$= 100 + 1.47 + (7.076 \times 1) + 1.47 + 100$$
$$\fallingdotseq 210.00(\text{ms})$$

(3) Reducing the link device refresh time

The link device refresh time can be reduced by decreasing the number of refresh points for the C Controller module. To decrease the refresh points, modify the refresh and network range assignment parameter settings and use the direct access method for accessing link devices.

(a) Concept of refresh range (points)

Within the range set with refresh parameters in the <<Parameter settings>> tab of MELSECNET/H utility, the areas set in the network range assignment are refreshed.

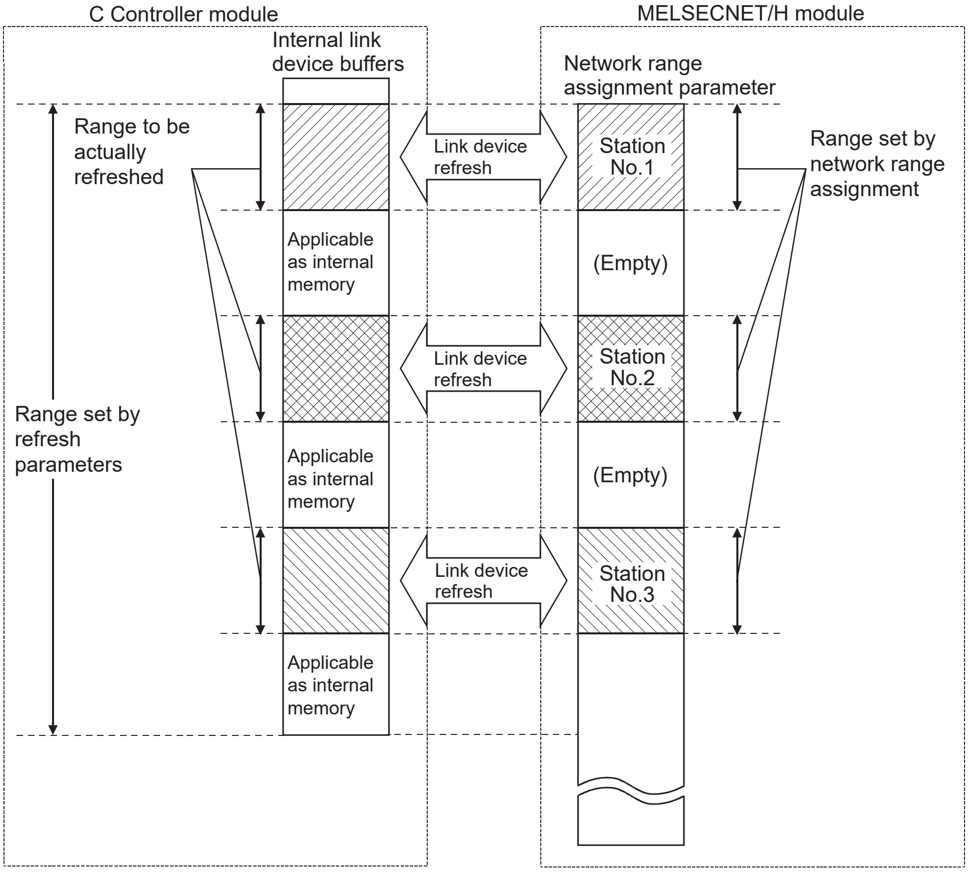

C Controller module
MELSECNET/H module
Internal link device buffers
Network range assignment parameter
Range to be actually refreshed
Range set by network range assignment
Link device refresh
Station No.1
Applicable as internal memory
(Empty)
Link device refresh
Station No.2
Range set by refresh parameters
Applicable as internal memory
(Empty)
Link device refresh
Station No.3
Applicable as internal memory

Figure 5.33 Concept of refresh range

☒POINT

The range from the first address to the last address is set by default in the refresh parameter setting.
(To bring the setting back to the default, click the Default button on the Refresh parameter settings screen activated from the <<Parameter settings>> tab of MELSECNET/H utility.)

(b) How to decrease the refresh points

1) Modifying refresh parameters
Up to 64 refresh ranges can be set for the C Controller module.
For refresh parameters, set only the ranges that need to be refreshed.

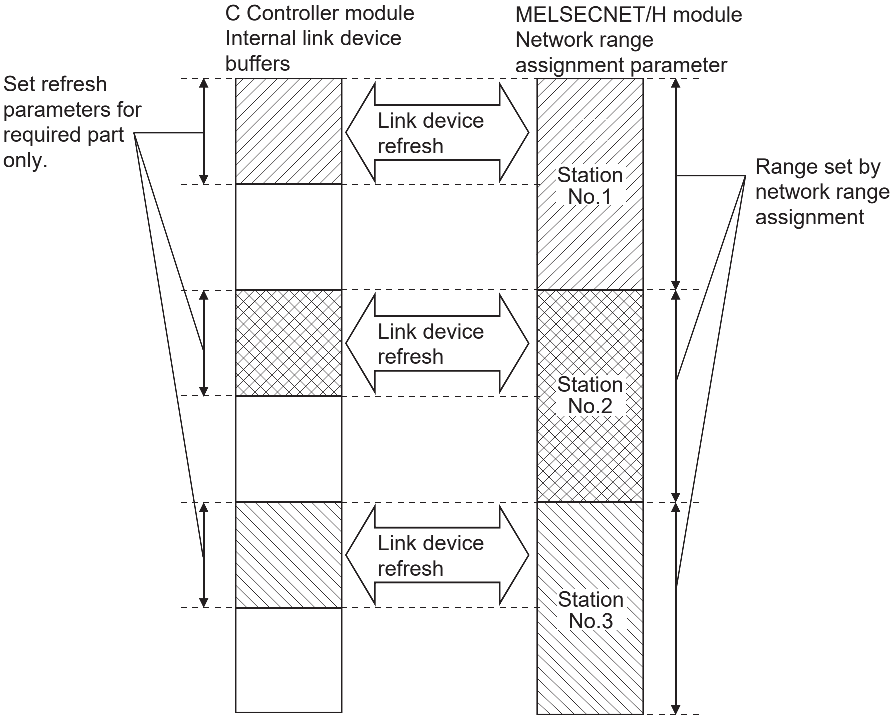


Figure 5.34 Modifying refresh parameters

2) Using direct access
By using direct access for link devices that are infrequently used on its own station and excluding these from the link device refresh ranges, the link device refresh time can be reduced.

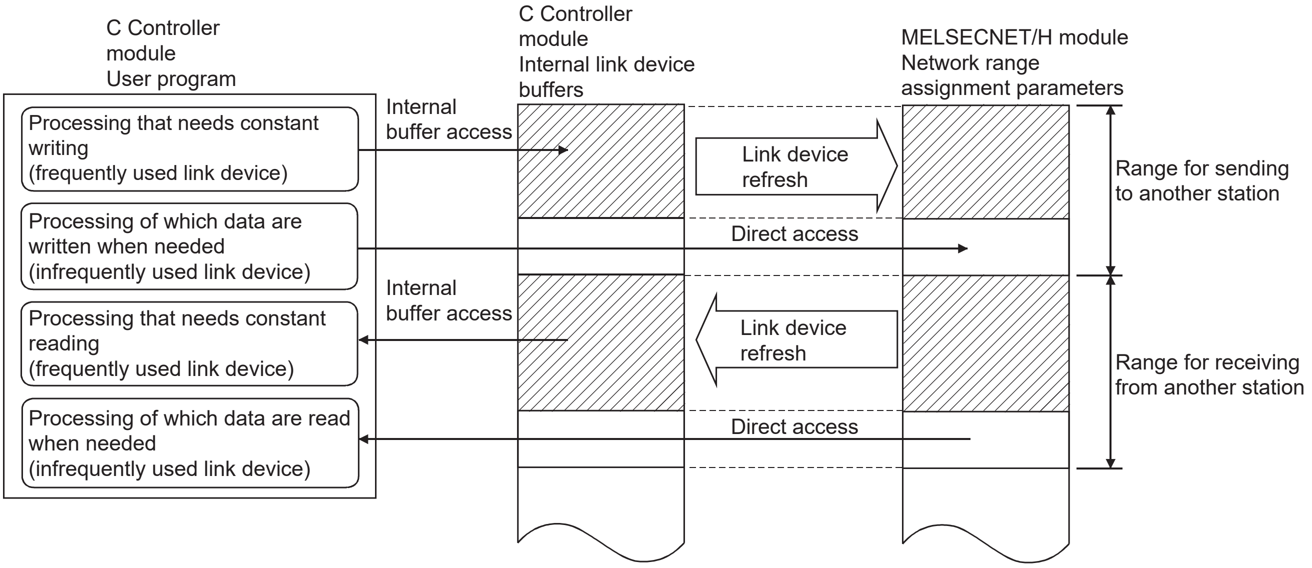

C Controller module User program
Processing that needs constant writing (frequently used link device)
Processing of which data are written when needed (infrequently used link device)
Processing that needs constant reading (frequently used link device)
Processing of which data are read when needed (infrequently used link device)
Internal buffer access
Internal buffer access
C Controller module Internal link device buffers
Link device refresh
Direct access
Link device refresh
Direct access
MELSECNET/H module Network range assignment parameters
Range for sending to another station
Range for receiving from another station

Figure 5.35 Reducing link device refresh time

(4) Reducing the link scan time

The link device refresh data and link scan data (LB/LW) can be reduced by assigning normal cyclic transmission data that are not time-critical to the extended area (2000_H to $3FFF_H$) for low-speed cyclic transmission.

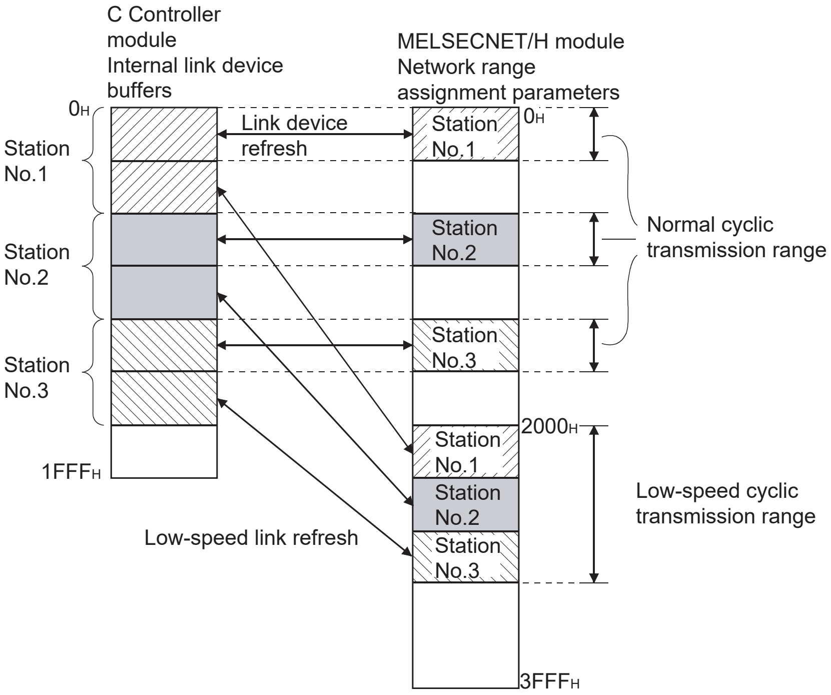

C Controller module Internal link device buffers
0H
Station No.1
Station No.2
Station No.3
1FFFH
Link device refresh
Low-speed link refresh
MELSECNET/H module Network range assignment parameters
0H
Station No.1
Station No.2
Station No.3
2000H
Station No.1
Station No.2
Station No.3
3FFFH
Normal cyclic transmission range
Low-speed cyclic transmission range

Figure 5.36 Reducing the link scan time

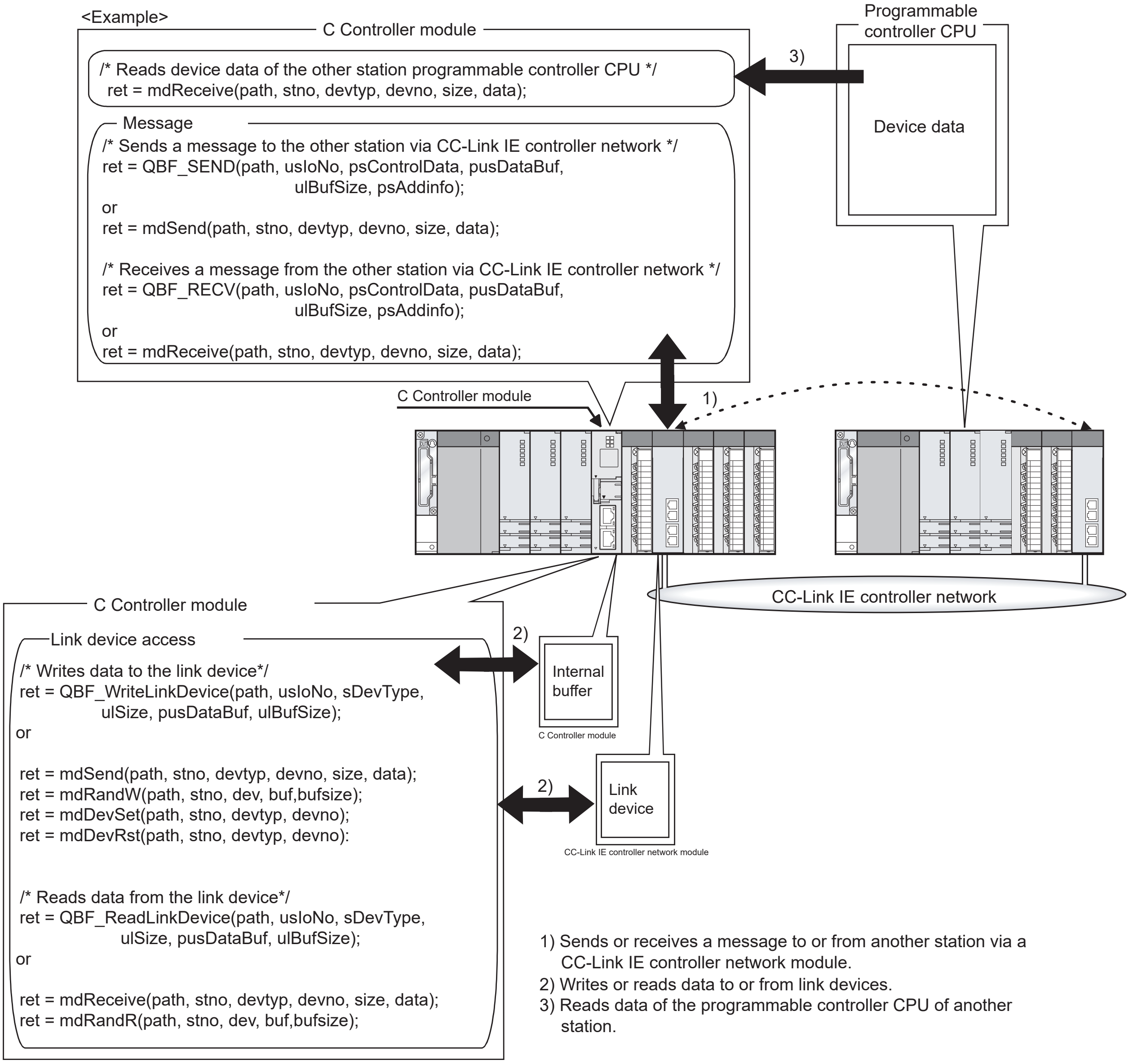
5.4 CC-Link IE Controller Network Module Access Function

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	×

The CC-Link IE controller network module access function allows access to each station, programmable controller CPU on another station, or a C Controller module connected to the CC-Link IE controller network via a CC-Link IE controller network module.

In a multiple CPU system configuration, access cannot be made via a CC-Link IE controller network module controlled by another CPU.

To create user programs of the C Controller module, use MELSEC data link functions or bus interface functions.



```
/* Reads device data of the other station programmable controller CPU */
 ret = mdReceive(path, stno, devtyp, devno, size, data);
```

```
/* Sends a message to the other station via CC-Link IE controller network */
ret = QBF_SEND(path, usIoNo, psControlData, pusDataBuf,
                              ulBufSize, psAddinfo);
or
ret = mdSend(path, stno, devtyp, devno, size, data);

/* Receives a message from the other station via CC-Link IE controller network */
ret = QBF_RECV(path, usIoNo, psControlData, pusDataBuf,
                              ulBufSize, psAddinfo);
or
ret = mdReceive(path, stno, devtyp, devno, size, data);
```

```
/* Writes data to the link device*/
ret = QBF_WriteLinkDevice(path, usIoNo, sDevType,
             ulSize, pusDataBuf, ulBufSize);
or

ret = mdSend(path, stno, devtyp, devno, size, data);
ret = mdRandW(path, stno, dev, buf,bufsize);
ret = mdDevSet(path, stno, devtyp, devno);
ret = mdDevRst(path, stno, devtyp, devno):


/* Reads data from the link device*/
ret = QBF_ReadLinkDevice(path, usIoNo, sDevType,
                ulSize, pusDataBuf, ulBufSize);
or

ret = mdReceive(path, stno, devtyp, devno, size, data);
ret = mdRandR(path, stno, dev, buf,bufsize);
```

1) Sends or receives a message to or from another station via a CC-Link IE controller network module.
2) Writes or reads data to or from link devices.
3) Reads data of the programmable controller CPU of another station.

Figure 5.37 CC-Link IE controller network access function

5.4.1 Message communication

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	×

The C Controller module can transfer messages (message transmission/reception) by a user program.

Messages can be transferred among the following modules via a CC-Link IE controller network module.

In a multiple CPU system configuration, any access via a CC-Link IE controller network module controlled by another CPU is not allowed.

- C Controller module on another station
- Programmable controller CPU

(1) Message transmission

Messages are sent with the user program in which communication functions for message transmission are used.

(2) Message reception

Messages are received with the user program in which communication functions for message reception are used.

The C Controller module has the internal buffer for message reception.

The data stored in this internal receive buffer are read out in message reception.

(a) Internal buffer for message reception

1) The internal buffer for message reception stores the data received from the CC-Link IE controller network module controlled by the C Controller module.

2) The internal receive buffer has areas for four CC-Link IE controller network modules, and up to 128 received data can be stored in each area.
If data are received when the buffer area has been full of received data, they will be discarded instead of being stored in the buffer.
Therefore, read received data before the number of the data stored in the internal receive buffer exceeds 128.

3) In message reception, data of the channel specified in the message receiving function are retrieved from the data stored in the internal buffer.

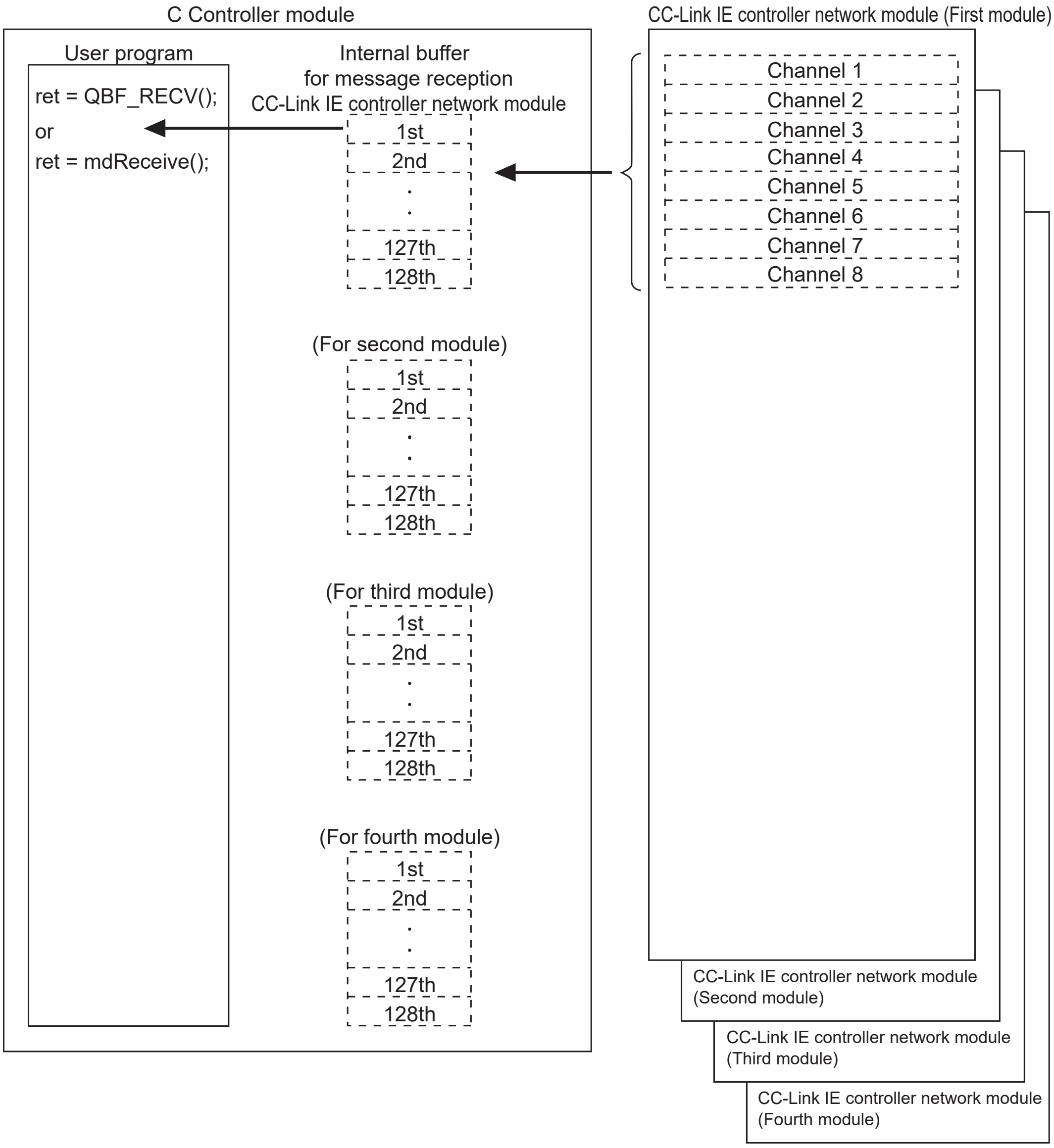


Figure 5.38 Internal buffer for message reception

Remark

For the bus interface functions and MELSEC data link functions, refer to the following manual.

☞ C Controller Module User's Manual (Utility Operation, Programming)

(3) Message communication example

The following a message communication example.

1) The C Controller module sends a message using a message sending function in the user program.

2) Station No.4 sends the message, by the message sending function in the user program and using channel 3, to the target station data storage channel 5 of station No.5.

3) The C Controller module (another station) automatically checks the message stored in station No.5, and stores the message into its own internal receive buffer.

4) The C Controller module (another station) reads the message from the internal receive buffer using a message receiving function.

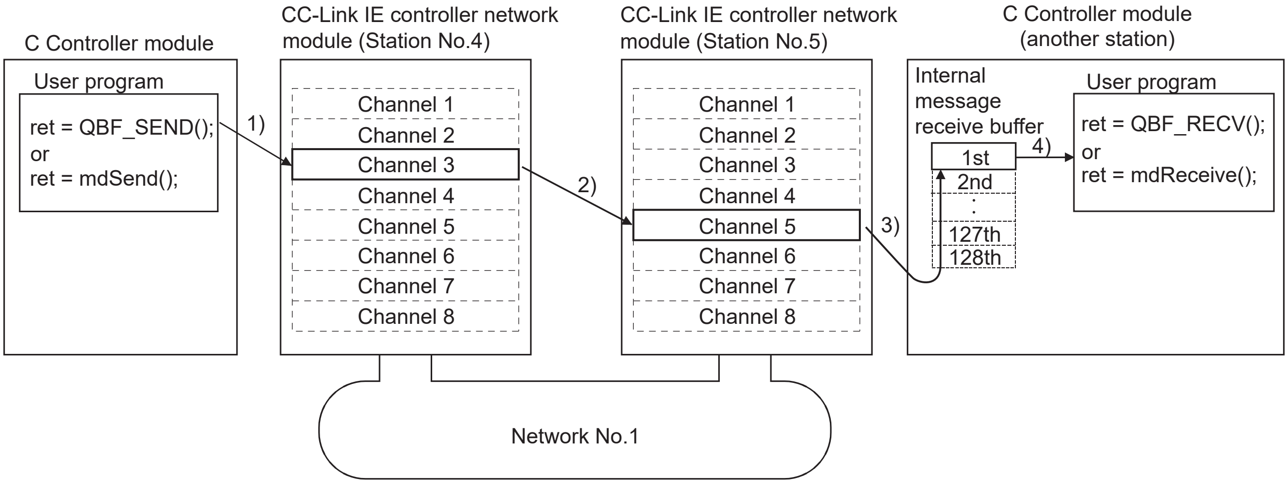


Figure 5.39 Message communication example

☒POINT

Pay attention to the following when sending a message to a CC-Link IE controller network module controlled by a programmable controller CPU.

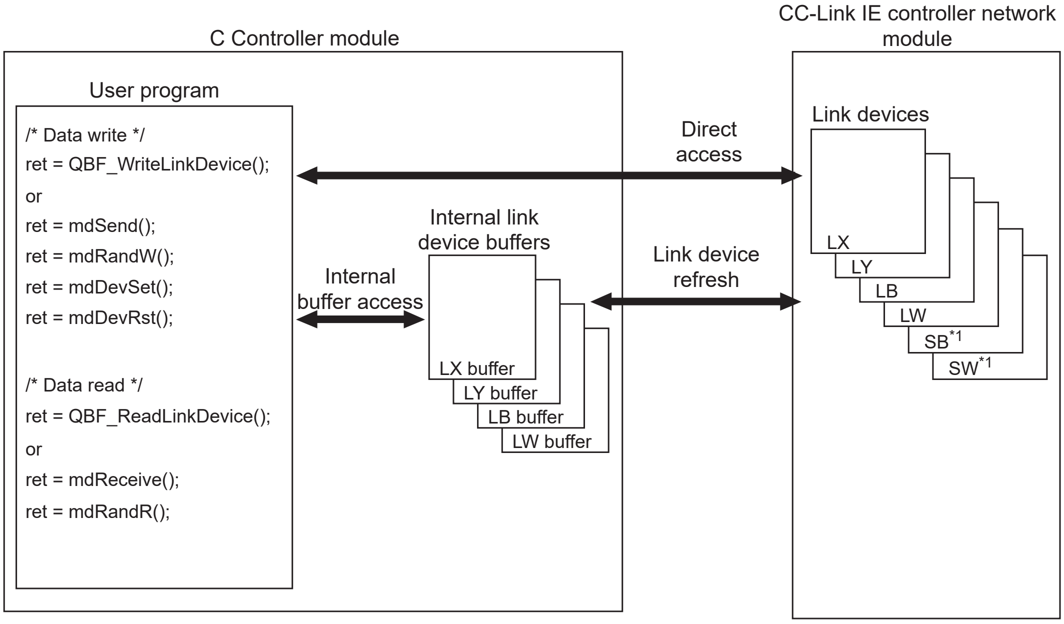
When sending data twice to the same channel of the receiving station with ACK set, execute the second transmission after the receiving station has read out the first data by the RECV instruction.
Failure to do so will cause an error. If an error is detected, retry the transmission after a little while.

5.4.2 Link device access

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	×

The link devices of a CC-Link IE controller network module controlled by the C Controller module can be accessed for data reading or writing from the user program on the C Controller module.

To access from the user program to the link devices, two different methods are available: the internal buffer access and the direct access.



```
/* Data write */
ret = QBF_WriteLinkDevice();
or
ret = mdSend();
ret = mdRandW();
ret = mdDevSet();
ret = mdDevRst();

/* Data read */
ret = QBF_ReadLinkDevice();
or
ret = mdReceive();
ret = mdRandR();
```

*1 Link device refresh is not available for the link devices SB/SW. Use the direct access.

Figure 5.40 Link device access

(1) Internal buffer access

(a) What is the internal buffer access?

The internal buffer access is a method of accessing the internal link device buffers in the C Controller module.

To use this method, the refresh parameters must be set in the <<Parameter settings>> tab of CC IE Control utility.

(b) Internal link device buffers

1) Internal link device buffers

- The internal link device buffer is reserved for each of four CC-Link IE controller network modules.
- Each of the internal link device buffers (LX, LY, LB, and LW buffers) has the same area size as the corresponding link device (LX, LY, LB, and LW). Set refresh ranges on the Refresh parameter settings screen activated from the <<Parameter settings>> tab of CC IE Control utility.
- The internal link device buffers are refreshed with the link device data by link device refresh.

2) Accessible devices

The following devices are available for internal link device buffer access.

Table 5.9 List of accessible devices

Accessible device	No. of points	Address range
LX buffer	8192	0000_H to $1FFF_H$
LY buffer	8192	0000_H to $1FFF_H$
LB buffer	32768	0000_H to $7FFF_H$
LW buffer	131072	0000_H to $1FFFF_H$

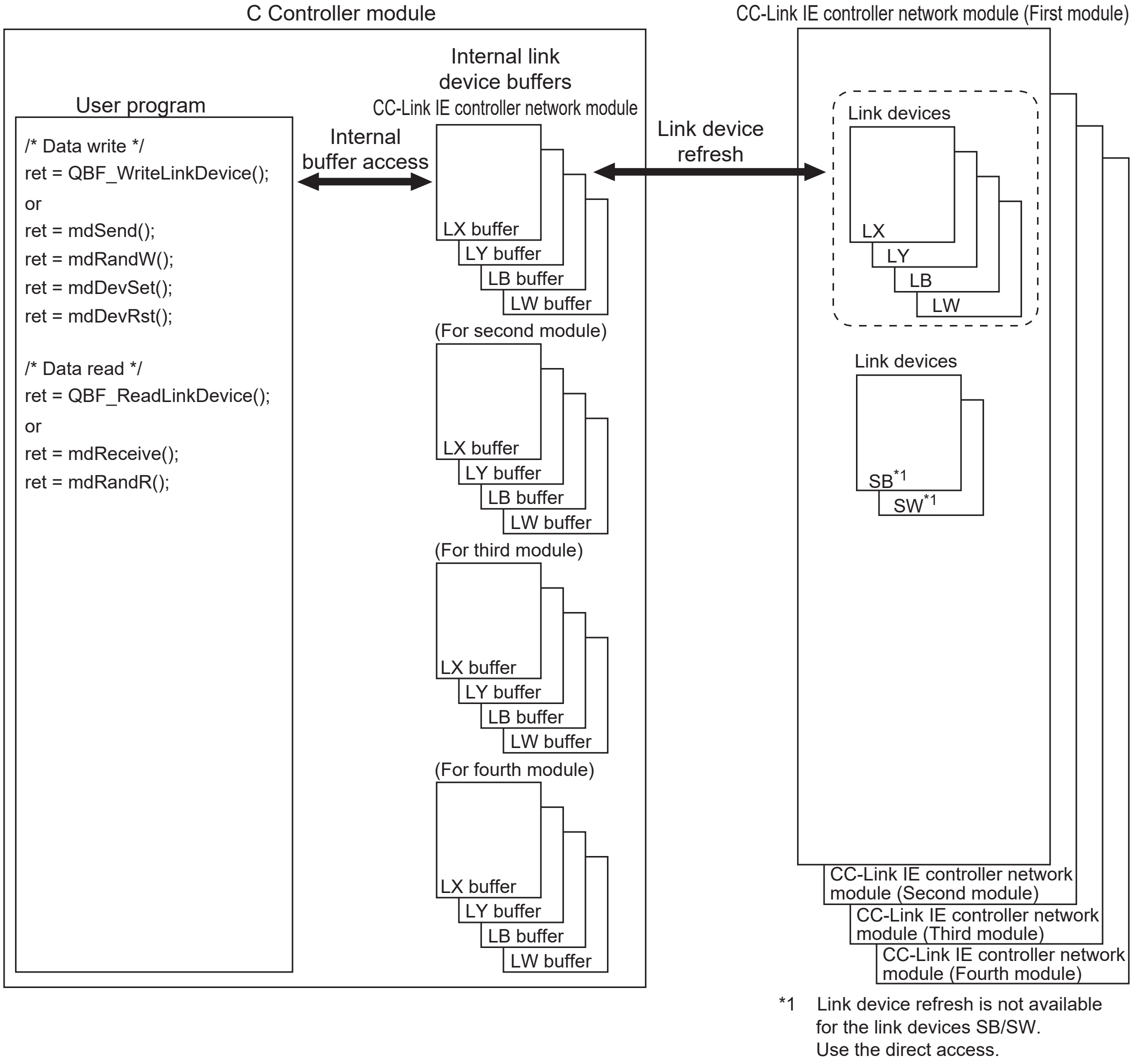

C Controller module
CC-Link IE controller network module (First module)
Internal link
device buffers
CC-Link IE controller network module
User program
/* Data write */
ret = QBF_WriteLinkDevice();
or
ret = mdSend();
ret = mdRandW();
ret = mdDevSet();
ret = mdDevRst();
/* Data read */
ret = QBF_ReadLinkDevice();
or
ret = mdReceive();
ret = mdRandR();
Internal
buffer access
Link device
refresh
LX buffer
LY buffer
LB buffer
LW buffer
(For second module)
(For third module)
(For fourth module)
Link devices
LX
LY
LB
LW
Link devices
SB*1
SW*1
CC-Link IE controller network module (Second module)
CC-Link IE controller network module (Third module)
CC-Link IE controller network module (Fourth module)
*1 Link device refresh is not available for the link devices SB/SW. Use the direct access.

Figure 5.41 Internal link device buffer access

(c) Link device refresh

Data are refreshed between the link devices and internal link device buffers in the specified link device refresh cycles and with refresh parameters set in CC IE Control utility.
(☞ Page 5-48, Section 5.4.4)

This refresh is performed in the area ranges that are set by refresh parameters and also set in Network range assignment in the <<Parameter settings>> tab of CC IE Control utility.

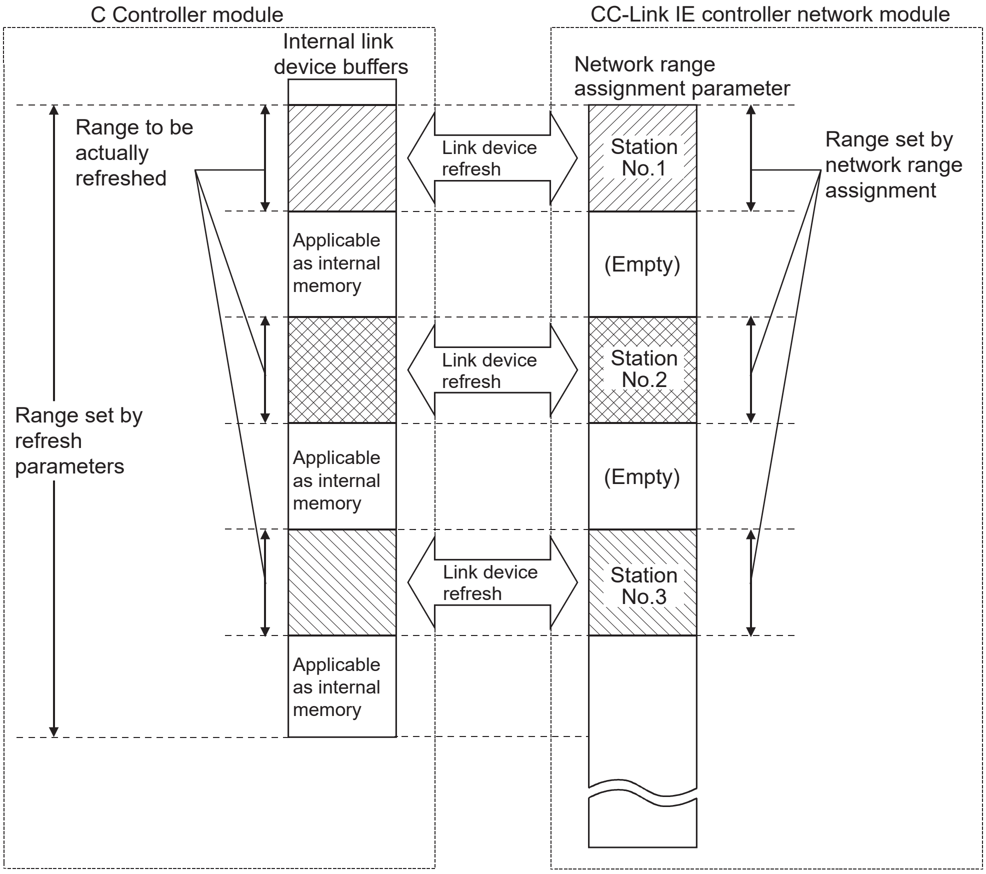

C Controller module
CC-Link IE controller network module
Internal link device buffers
Network range assignment parameter
Range to be actually refreshed
Range set by network range assignment
Range set by refresh parameters
Link device refresh
Link device refresh
Link device refresh
Station No.1
(Empty)
Station No.2
(Empty)
Station No.3
Applicable as internal memory
Applicable as internal memory
Applicable as internal memory

Figure 5.42 Link device refresh range

☒**POINT**

An internal link device buffer, which is not used for link device refresh, can be utilized as an internal memory of a user program.

(2) Direct access

(a) What is direct access?
The direct access is a method of directly accessing a link device of a CC-Link IE controller network module from a user program on the C Controller module, independently of the link device refresh.
Since direct access does not use any internal link device buffer, the link device refresh cycle and refresh parameters need not be set in CC IE Control utility.

(b) Directly accessible devices
The following devices are directly accessible.

Table 5.10 Directly accessible device list

Access method	Accessible device
Direct access	LX
	LY
	LB
	LW
	SB
	SW

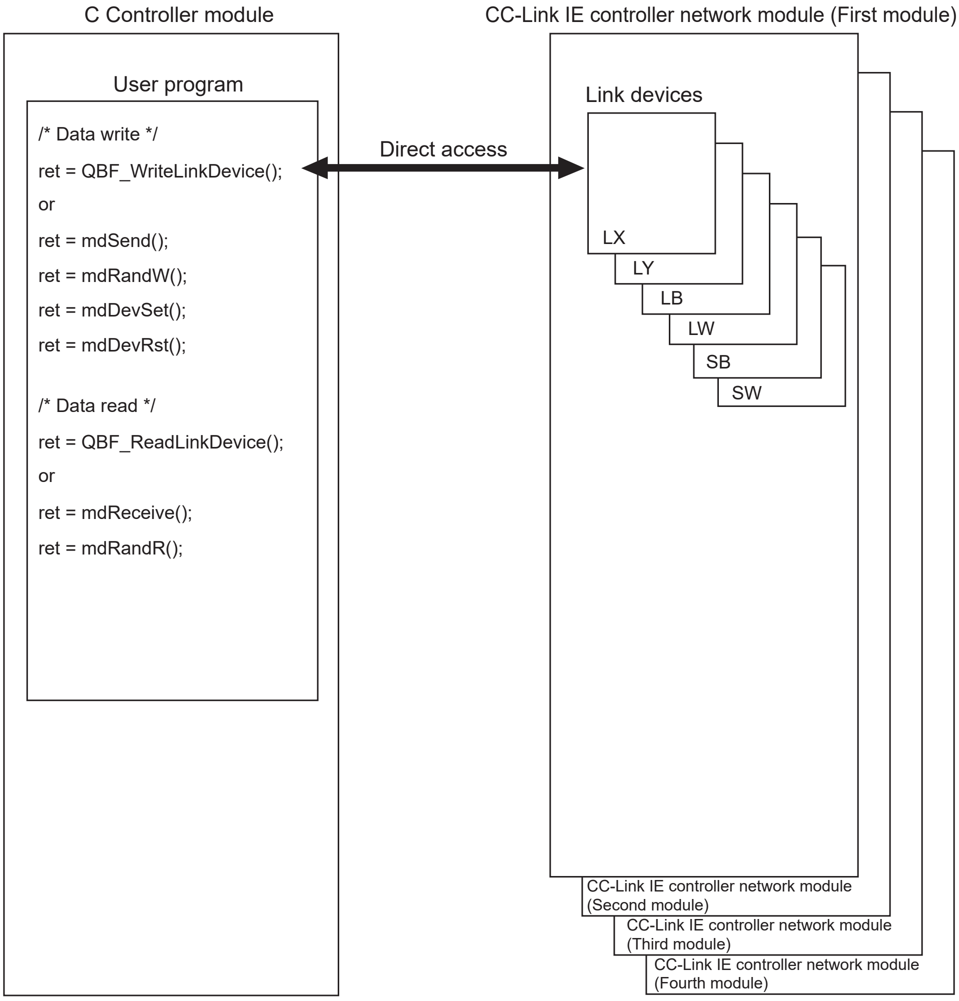

C Controller module
User program
/* Data write */
ret = QBF_WriteLinkDevice();
or
ret = mdSend();
ret = mdRandW();
ret = mdDevSet();
ret = mdDevRst();
/* Data read */
ret = QBF_ReadLinkDevice();
or
ret = mdReceive();
ret = mdRandR();
Direct access
CC-Link IE controller network module (First module)
Link devices
LX
LY
LB
LW
SB
SW
CC-Link IE controller network module (Second module)
CC-Link IE controller network module (Third module)
CC-Link IE controller network module (Fourth module)

Figure 5.43 Direct access

(c) Selection of internal buffer access or direct access
Select a suitable access method referring to the following.

1) Data suitable for internal buffer access
If a link device is used frequently or the block data assurance per station is needed for a link device, the data of the link device are suitable for internal buffer access.
For this kind of data, set refresh parameters to utilize them for internal buffer access.

2) Data suitable for direct access
Data of a link device that is not used so frequently are suitable for direct access.
Exclude these data from the refresh parameter setting, and utilize them for direct access.

☒POINT

To enable the block data assurance per station, set as follows.

1) Select "Block send data assurance per station" or "Block receive data assurance per station" in Parameter settings (Supplementary settings) for CC-Link IE controller network module (control station).
This Parameter settings (Supplementary settings) can be configured in CC IE Control utility when the control CPU of the CC-Link IE controller network module (control station) is a C Controller module, or in GX Works2/GX Developer when the control CPU is a programmable controller CPU.
2) In the <<Parameter settings>> tab of CC IE Control utility, set 10 or higher for "Link device refresh cycle" and configure "Refresh parameters" settings.
If 0 is set for "Link device refresh cycle" with "Block data assurance per station" enabled, cyclic transmission will not be performed properly and a data link error may occur.

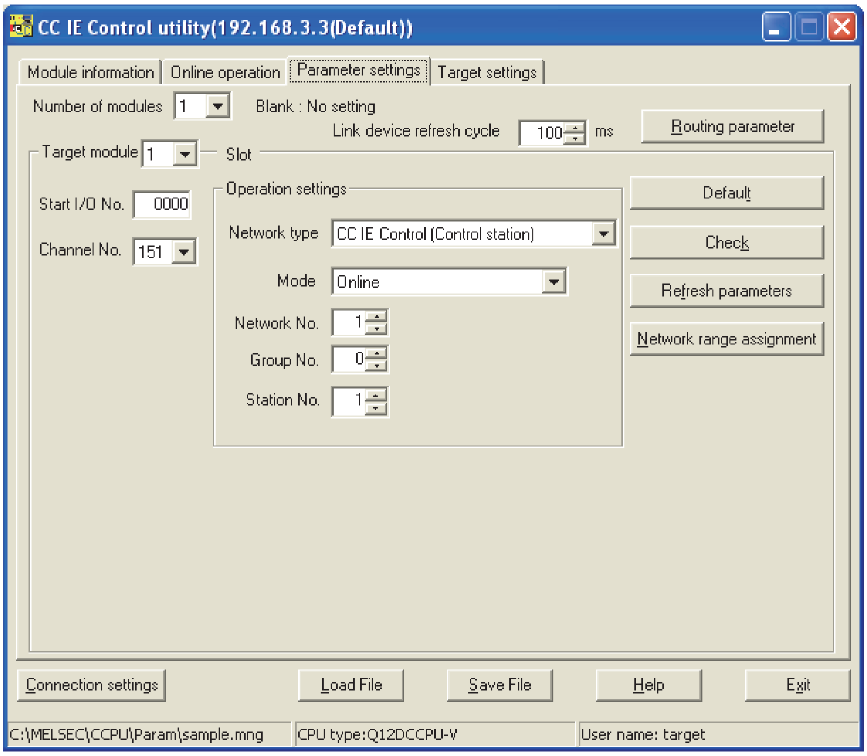
Remark

For CC IE Control utility, bus interface functions, and MELSEC data link functions, refer to the following manual.
☞ C Controller Module User's Manual (Utility Operation, Programming)

5.4.3 Parameter settings

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	×

Set parameters in the <<Parameter settings>> tab of CC IE Control utility.
(☞ C Controller Module User's Manual (Utility Operation, Programming))



CC IE Control utility(192.168.3.3(Default))

Number of modules: 1

Blank : No setting

Link device refresh cycle: 100 ms

Target module: 1

Start I/O No.: 0000

Channel No.: 151

Network type: CC IE Control (Control station)

Mode: Online

Network No.: 1

Group No.: 0

Station No.: 1

C:\MELSEC\CCPU\Param\sample.mng

CPU type:Q12DCCPU-V

User name: target

Figure 5.44 Parameter settings tab

5.4.4 Link device refresh setting

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	×

Link device refresh setting is required when using the internal buffer access in user programs.
To enable the link device refresh, set a link device refresh cycle and refresh parameters in CC IE Control utility.

(1) Link device refresh cycle

The link device refresh cycle is an interval of time during which the internal link device buffers of the C Controller module and link devices of a CC-Link IE controller network module are refreshed.
The concept of the link device refresh cycle is shown below.

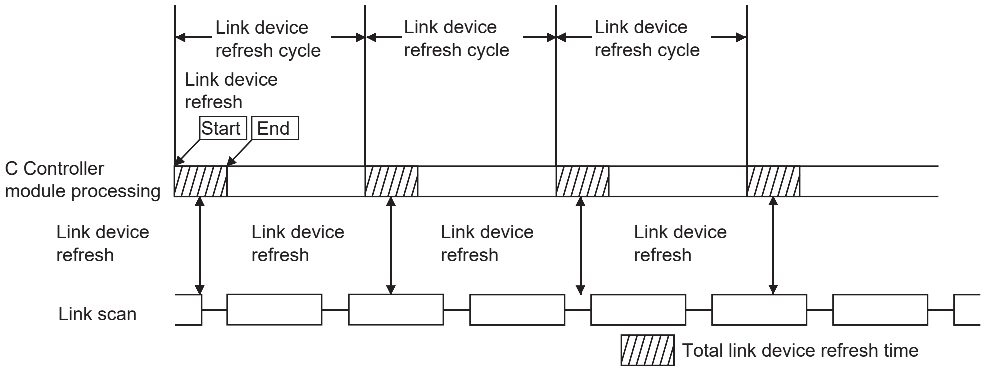

Link device refresh cycle
Link device refresh cycle
Link device refresh cycle
Link device refresh
Start
End
C Controller module processing
Link device refresh
Link device refresh
Link device refresh
Link device refresh
Link scan
Total link device refresh time

Figure 5.45 Conceptual diagram of link device refresh cycle

(a) Total link device refresh time

1) What is total link device refresh time?
This is a processing time for refreshing the link devices of all CC-Link IE controller network modules that are controlled by the C Controller module.

2) How to obtain the total link device refresh time
A theoretical value for the total link device refresh time can be obtained by a calculation formula. (☞ Page 5-59, Section 5.4.5 (1) (b))

3) Variation in total link device refresh time

The total link device refresh time varies depending on the operating conditions of the C Controller system.

It may take longer time if the interrupt processing or high-priority task processing, such as the following, is activated during link device refresh.

- Shell command
- Workbench/Tornado connection
- File access
- Mount/unmount of a CompactFlash card
- Ethernet communications
- NFS server communication
- Processing of a bus interface driver*1

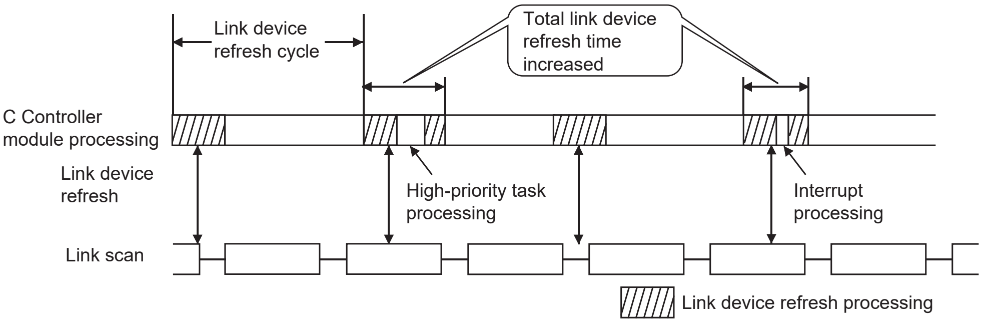

Link device refresh cycle
Total link device refresh time increased
C Controller module processing
Link device refresh
High-priority task processing
Interrupt processing
Link scan
▨ Link device refresh processing

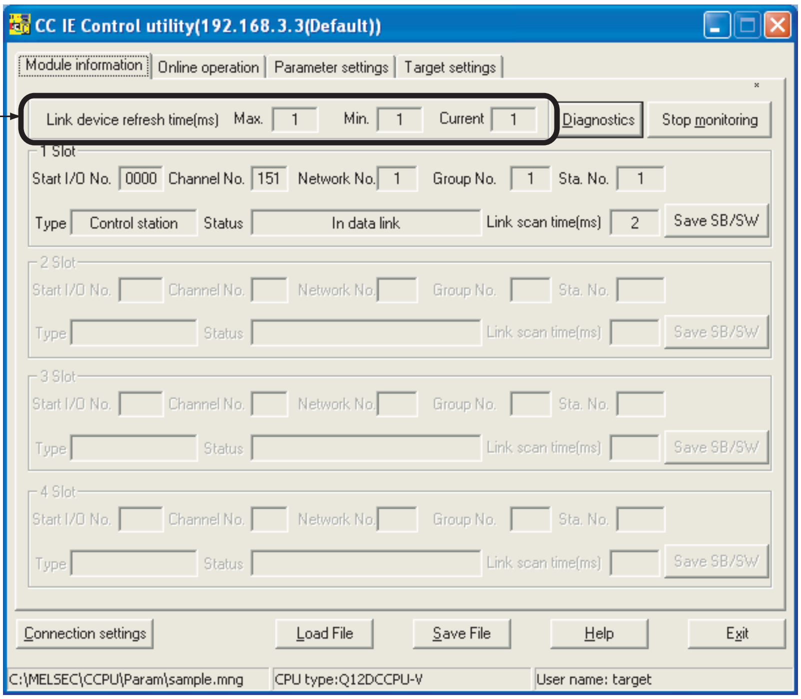
Figure 5.46 Variation in total link device refresh time

* 1 Functions to transmit with peripheral devices, functions to communicate with an intelligent function module, etc.

4) Checking the total link device refresh time

The total link device refresh time during operation of the C Controller system can be checked in "Link device refresh time (Max./Min./Current)" in the <<Module information>> tab of CC IE Control utility.

Total link device refresh time during C Controller system operation



CC IE Control utility(192.168.3.3(Default))

Link device refresh time(ms) Max. 1 Min. 1 Current 1

1 Slot
Start I/O No. 0000 Channel No. 151 Network No. 1 Group No. 1 Sta. No. 1
Type Control station Status In data link Link scan time(ms) 2

2 Slot
Start I/O No. Channel No. Network No. Group No. Sta. No.
Type Status Link scan time(ms)

3 Slot
Start I/O No. Channel No. Network No. Group No. Sta. No.
Type Status Link scan time(ms)

4 Slot
Start I/O No. Channel No. Network No. Group No. Sta. No.
Type Status Link scan time(ms)

C:\MELSEC\CCPU\Param\sample.mng CPU type:Q12DCCPU-V User name: target

Figure 5.47 Link device refresh time

(b) How to determine the link device refresh cycle
Take the steps according to the following flowchart.

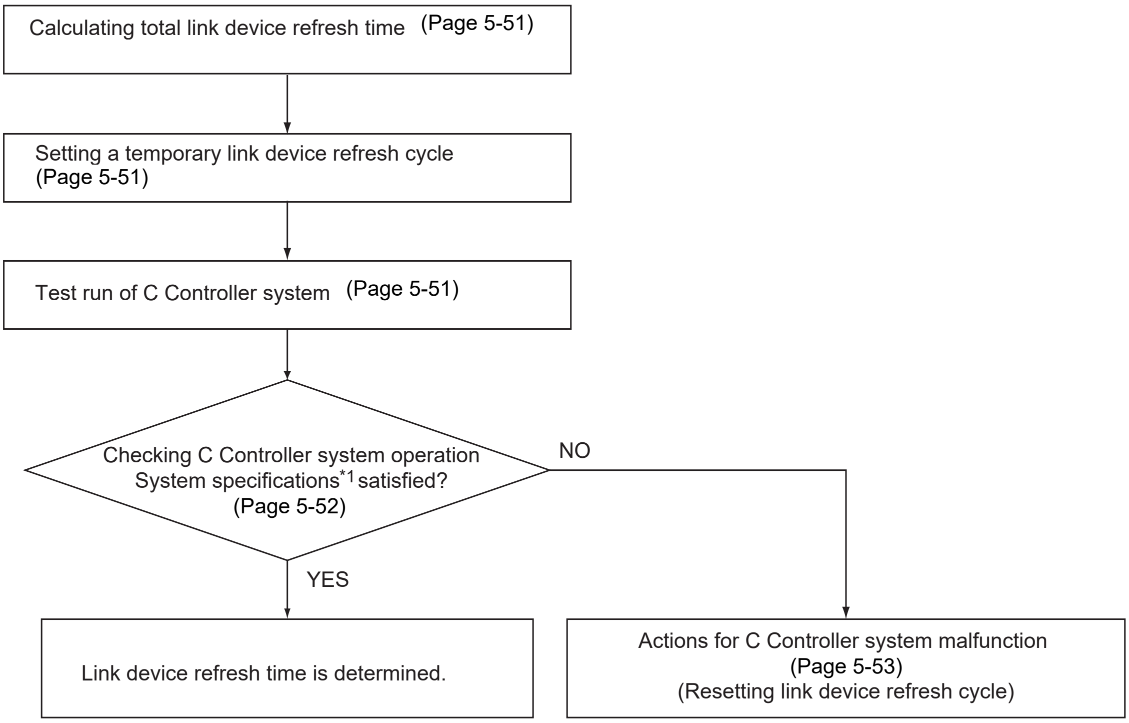

Calculating total link device refresh time (Page 5-51)
Setting a temporary link device refresh cycle (Page 5-51)
Test run of C Controller system (Page 5-51)
Checking C Controller system operation System specifications*1 satisfied? (Page 5-52)
NO
YES
Link device refresh time is determined.
Actions for C Controller system malfunction (Page 5-53) (Resetting link device refresh cycle)

Figure 5.48 Link device refresh cycle setting flowchart

* 1 The system specifications describe the whole system operations including the system functions and processing specifications required for the C Controller system, user program processing, functions, and performance specifications.

1) Calculating the total link device refresh time
   Calculate the total link device refresh time.
   (☞ Page 5-59, Section 5.4.5 (1) (b))

2) Setting a temporary link device refresh cycle
   The link device refresh cycle value and the total link device refresh time value obtained in 1) must satisfy the following relational expression.*1
   Set a temporary value for the link device refresh cycle so that the expression is satisfied.

[Relational expression]

Total link device refresh time < Link device refresh cycle

* 1 When the C Controller module (own station) controls a CC-Link IE controller network module and a MELSECNET/H module, the following expression must be also satisfied, in consideration of the total link device refresh time for the MELSECNET/H and CC-Link IE controller networks.

[Relational expression]

$(\text{TLDRT}_{\text{MNETH}} + \text{TLDRT}_{\text{CC-Link IE Control}})$ < Link device refresh cycle

$\text{TLDRT}_{\text{MNETH}}$: Total link device refresh time for MELSECNET/H

$\text{TLDRT}_{\text{CC-Link IE Control}}$: Total link device refresh time for CC-Link IE controller network

3) Test run of the C Controller system
   Perform the test run of the C Controller system to check if the system will operate properly with the link device refresh cycle that is temporarily set in 2).

4) Checking the C Controller system operation
In the test run of the C Controller system using the temporary link device refresh cycle setting, check the following 1 and 2.
[Check 1: Check for an error in the C Controller module.]
Check if any error has occurred in the C Controller module.
If the link device refresh is not completed within the specified link device refresh cycle, a "Link device refresh timeout error (error code: 5012, LNKREF.TIME OVER)" occurs in the C Controller module.
For this error, take corrective actions according to 5).

[Check 2: Check if the C Controller system operation meets the system specifications or not.]
Check if the C Controller system operation meets the system specifications or not.
Even if the error stated in Check 1 is not detected, normal operation of the C Controller system must be confirmed.
User program processing in the C Controller module may be suspended when the following is activated, depending on the operating conditions of the C Controller system.

- Link device refresh processing
- Interrupt processing
- High-priority task processing (such as the followings)
  - Shell command
  - Workbench/Tornado connection
  - File access
  - Mount/unmount of a CompactFlash card
  - Ethernet communications
  - NFS server communication
  - Processing of a bus interface driver[*1]

If the user program operation is interrupted longer then its operation time, the C Controller system specifications may not be satisfied.
If this occurs, take corrective actions according to 5).

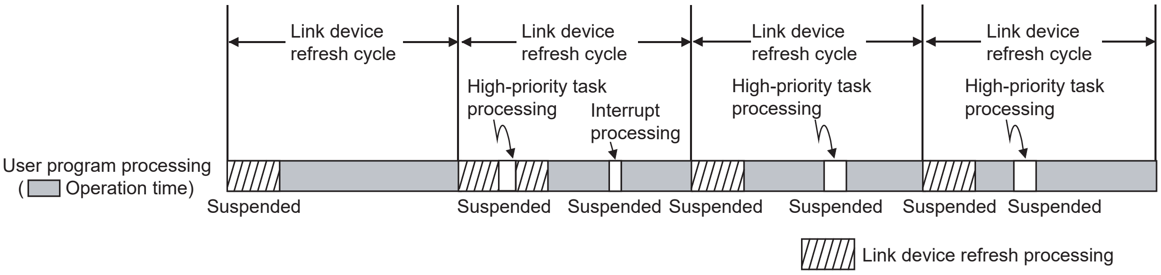

Link device refresh cycle
Link device refresh cycle
Link device refresh cycle
Link device refresh cycle
High-priority task processing
Interrupt processing
High-priority task processing
High-priority task processing
User program processing
(▭ Operation time)
Suspended
Suspended
Suspended
Suspended
Suspended
Suspended
Suspended
▨ Link device refresh processing

Figure 5.49 User program processing of C Controller module

* 1 Functions to transmit with peripheral devices, functions to communicate with an intelligent function module, etc.

5) When the C Controller system is malfunctioning
In the case of C Controller system malfunction or faulty user program processing performance, take actions 1 to 3 if necessary to correct the system operation.

[Action 1: Increase the link device refresh cycle setting value.]
Check the total link device refresh time (Max.) measured during C Controller system operation, in "Link device refresh time (Max.)" in the <<Module information>> tab of either of the following utilities.

- CC IE Control utility
- MELSECNET/H utility

Re-set a link device refresh cycle appropriate to the confirmed link device refresh time (Max.) so that the following relational expression is satisfied. Also, since the C Controller system specifications must be satisfied, determine the link device refresh cycle, taking into account the user program processing.

[Relational expression]
Measured link device refresh time (Max.) value < Link device refresh cycle

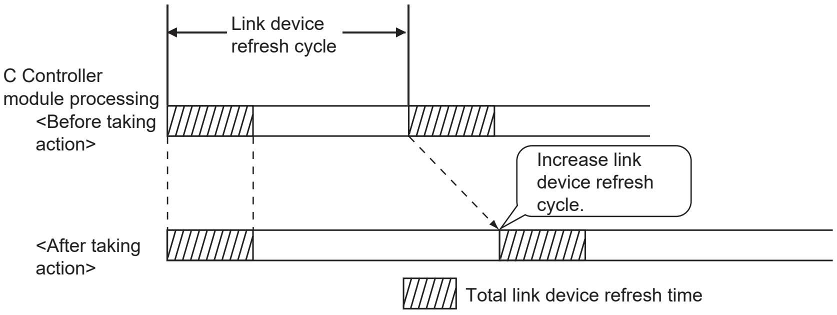

Link device refresh cycle
C Controller module processing
<Before taking action>
Increase link device refresh cycle.
<After taking action>
Total link device refresh time

Figure 5.50 Corrective action 1

[Action 2: Decrease the refresh points.]
Check the refresh parameters and decrease the number of refresh points to reduce the total link device refresh time. (☞ Page 5-62, Section 5.4.5 (3))

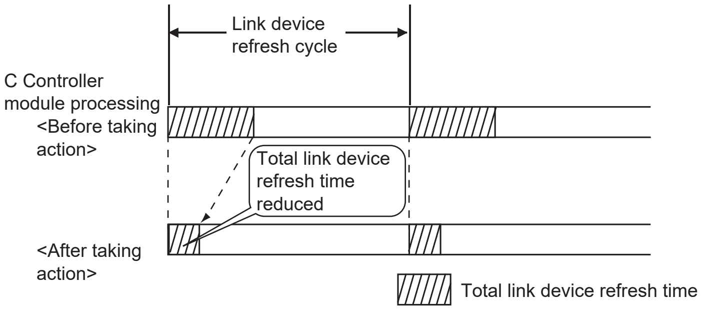
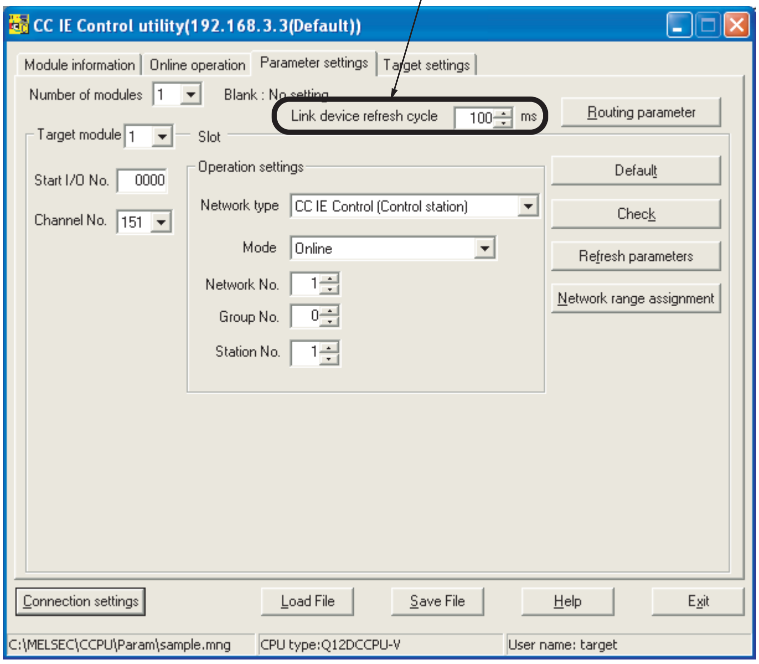

Link device refresh cycle
C Controller module processing
<Before taking action>
Total link device refresh time reduced
<After taking action>
Total link device refresh time

Figure 5.51 Corrective action 2

[Action 3: Modify the user program.]
Modify the task configuration, priority, processing details of the user program so that the C Controller system specifications and user program processing performance can be satisfied.

(c) Link device refresh cycle setting
Set a link device refresh cycle in "Link device refresh cycle" in the <<Parameter settings>> tab.

Link device refresh cycle setting



CC IE Control utility(192.168.3.3(Default))

Number of modules 1
Blank : No setting
Link device refresh cycle 100 ms
Target module 1
Slot
Start I/O No. 0000
Channel No. 151

Operation settings
Network type CC IE Control (Control station)
Mode Online
Network No. 1
Group No. 0
Station No. 1

C:\MELSEC\CCPU\Param\sample.mng
CPU type:Q12DCCPU-V
User name: target

Figure 5.52 Link device refresh cycle setting

(2) Refresh parameters

The refresh parameters are parameters by which data are transferred from the link devices (LX, LY, LB, and LW) of a CC-Link IE controller network module to the internal link device buffers (LX, LY, LB and LW buffers) of the C Controller module for use in the user program.

(a) Link refreshable devices

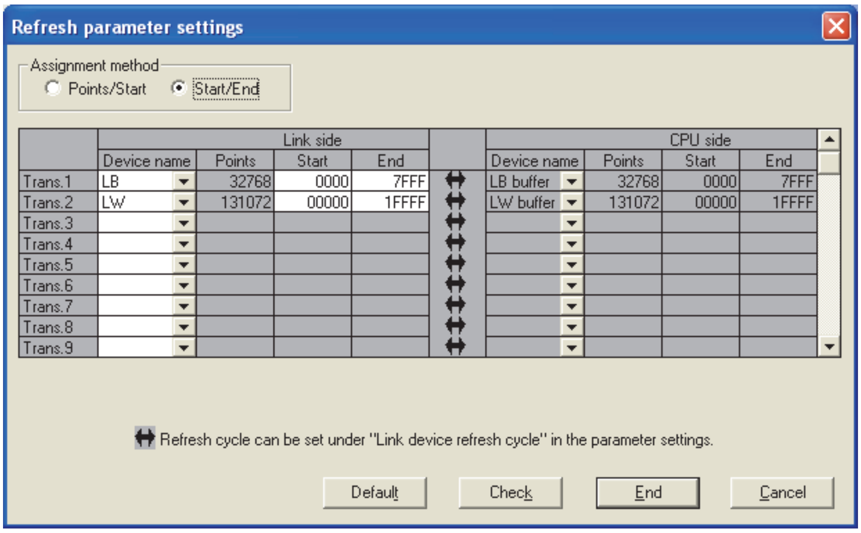
Up to 64 transfer settings are allowed for each CC-Link IE controller network module.

The following transfer settings are available.

- Transfer the link side device (LX) data to the CPU side device (LX buffer).
- Transfer the link side device (LY) data to the CPU side device (LY buffer).
- Transfer the link side device (LB) data to the CPU side device (LB buffer).
- Transfer the link side device (LW) data to the CPU side device (LW buffer).

(b) Setting refresh parameters

Set refresh parameters in the Refresh parameter settings screen activated from the <<Parameter settings>> tab of CC IE Control utility.



Refresh parameter settings

Assignment method
○ Points/Start ● Start/End

	Link side					CPU side			
	Device name	Points	Start	End		Device name	Points	Start	End
Trans.1	LB	32768	0000	7FFF	⇔	LB buffer	32768	0000	7FFF
Trans.2	LW	131072	00000	1FFFF	⇔	LW buffer	131072	00000	1FFFF
Trans.3					⇔				
Trans.4					⇔				
Trans.5					⇔				
Trans.6					⇔				
Trans.7					⇔				
Trans.8					⇔				
Trans.9					⇔				

⇔ Refresh cycle can be set under "Link device refresh cycle" in the parameter settings.

Figure 5.53 Refresh parameter settings screen

(3) Refresh parameter setting example

The following is a refresh parameter setting example.

(a) System configuration

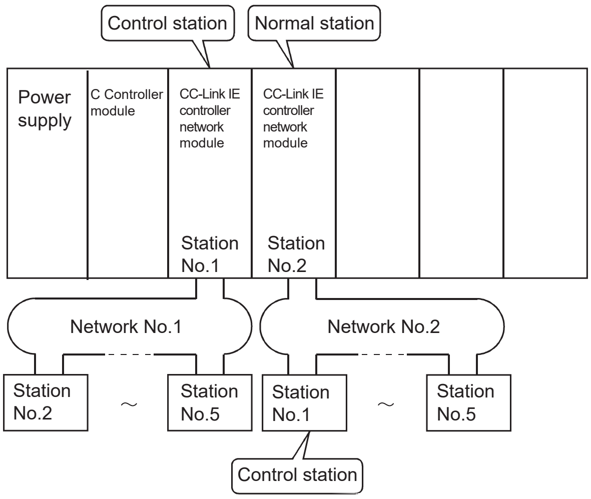


Figure 5.54 System configuration

(b) Parameter assignment

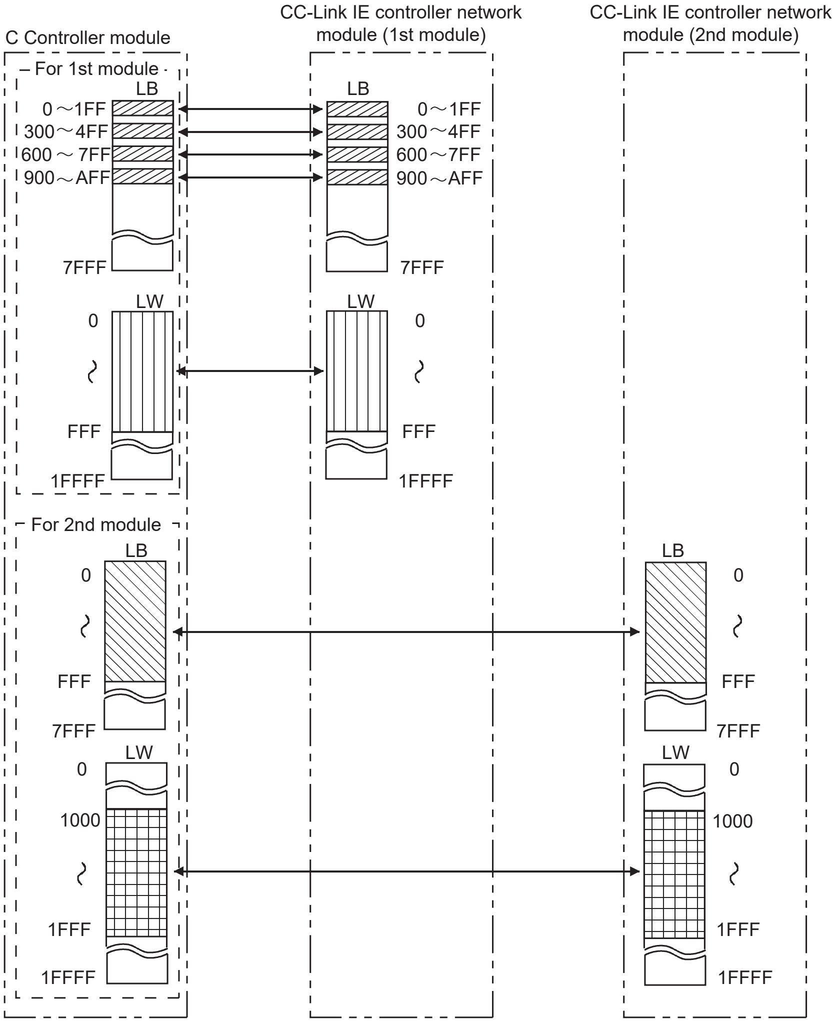
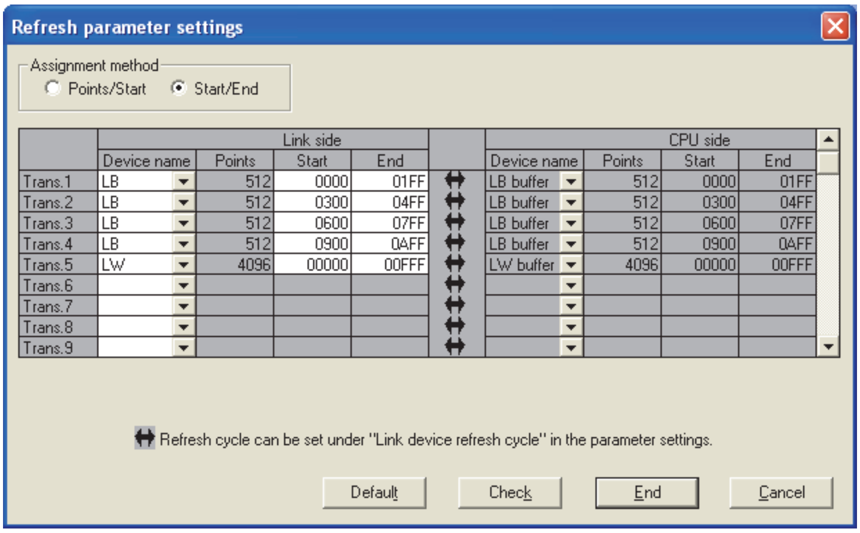


Figure 5.55 Parameter assignment

(c) CC IE Control utility: Refresh parameter settings screen



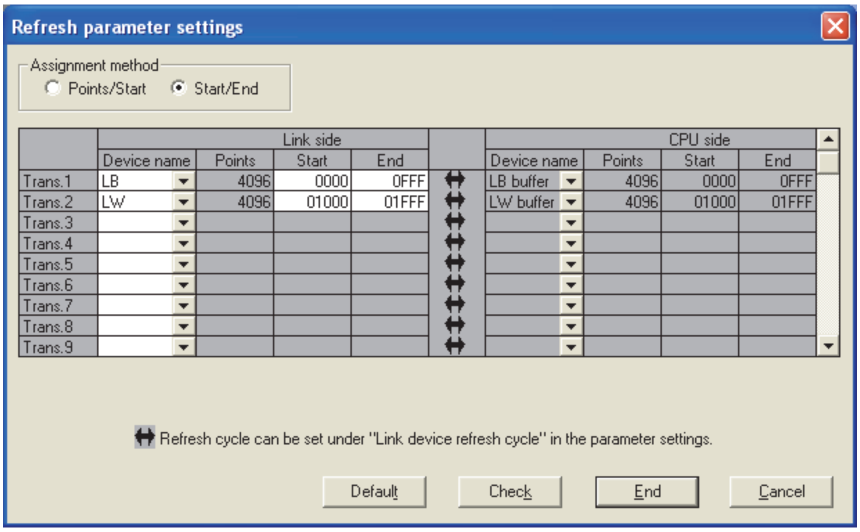
Refresh parameter settings

Assignment method
○ Points/Start ● Start/End

	Link side					CPU side			
	Device name	Points	Start	End		Device name	Points	Start	End
Trans.1	LB	512	0000	01FF	↔	LB buffer	512	0000	01FF
Trans.2	LB	512	0300	04FF	↔	LB buffer	512	0300	04FF
Trans.3	LB	512	0600	07FF	↔	LB buffer	512	0600	07FF
Trans.4	LB	512	0900	0AFF	↔	LB buffer	512	0900	0AFF
Trans.5	LW	4096	00000	00FFF	↔	LW buffer	4096	00000	00FFF
Trans.6					↔				
Trans.7					↔				
Trans.8					↔				
Trans.9					↔				

↔ Refresh cycle can be set under "Link device refresh cycle" in the parameter settings.

Figure 5.56 CC-Link IE controller network module (1st module) setting



Refresh parameter settings

Assignment method
○ Points/Start ● Start/End

	Link side					CPU side			
	Device name	Points	Start	End		Device name	Points	Start	End
Trans.1	LB	4096	0000	0FFF	↔	LB buffer	4096	0000	0FFF
Trans.2	LW	4096	01000	01FFF	↔	LW buffer	4096	01000	01FFF
Trans.3					↔				
Trans.4					↔				
Trans.5					↔				
Trans.6					↔				
Trans.7					↔				
Trans.8					↔				
Trans.9					↔				

↔ Refresh cycle can be set under "Link device refresh cycle" in the parameter settings.

Figure 5.57 CC-Link IE controller network module (2nd module) setting

Remark

For the CC IE Control utility, refer to the following manual.

☞ C Controller Module User's Manual (Utility Operation, Programming)

5.4.5 Link data transfer processing time specifications

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	×

This section describes cyclic transmission and processing time of CC-Link IE controller network modules.

(1) Cyclic transmission function

This function allows periodic data communication between stations on the same network, using link devices (LB/LW/LX/LY) of CC-Link IE controller network modules.

(a) Processing in cyclic transmission

In the following example, the link relay (B) data on the C Controller module side are sent to the link relay (B) of another station's C Controller module.

1) B0 in the LB buffer (sending side) is set to ON.

2) By a link refresh, the B0 information in the LB buffer is stored in the link data storage area (LB) of the sending side CC-Link IE controller network module.

3) By a link scan, the B0 information is stored in the receiving side CC-Link IE controller network module's link data storage area (LB).

4) By a link refresh, the B0 information in the link data storage area (LB) is stored in the LB buffer of the receiving side C Controller module.

5) B0 in the LB buffer (receiving side) is set to ON.

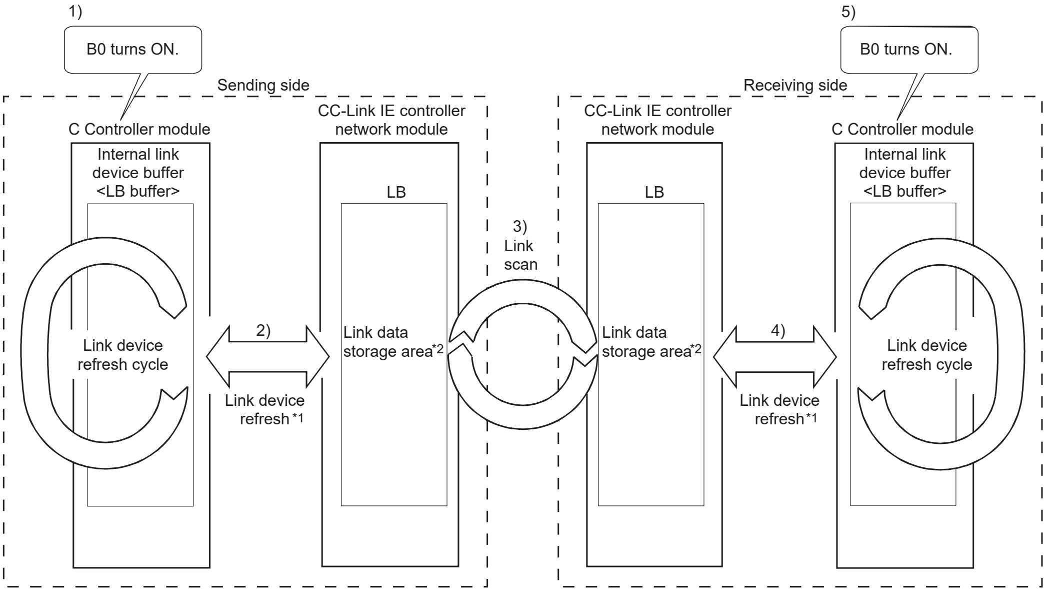

1)
B0 turns ON.
Sending side
C Controller module
Internal link device buffer <LB buffer>
Link device refresh cycle
2)
Link device refresh *1
CC-Link IE controller network module
LB
Link data storage area *2
3) Link scan
Receiving side
CC-Link IE controller network module
LB
Link data storage area *2
4)
Link device refresh *1
5)
B0 turns ON.
C Controller module
Internal link device buffer <LB buffer>
Link device refresh cycle

Figure 5.58 Link data transfer processing

* 1 Refresh parameters must be set for link refreshes.

* 2 Network range assignment parameters must be set for link scans.

(b) Total link device refresh time (β_T, β_R)
The total link device refresh time can be calculated by the following formula, with the points assigned to link devices.

[Total link device refresh time]

$$\beta_T, \beta_R = KM1 + KM2 \times \left\{ \left(\frac{LB+LX+LY}{16} \right) + LW \right\} [ms]$$

β_T: Total link device refresh time (sending side)

β_R: Total link device refresh time (receiving side)

LB: Total points of link relay (LB) refreshed by the station*1

LW: Total points of link register (LW) refreshed by the station*1

LX: Total points of link input (LX) refreshed by the station*1

LY: Total points of link output (LY) refreshed by the station*1

KM1,KM2 : Constants

Table 5.11 KM1 and KM2 values

CC-Link IE controller network module location	KM1	KM2
Main base unit	1.0	0.36×10^{-3}
Extension base unit	1.0	0.96×10^{-3}

* 1 Total of link device points within the range set by refresh parameters and set in the network range assignment. (Excluding the points assigned to reserved stations)
Total points for the CC-Link IE controller network modules that are controlled by the C Controller module.

(2) How to calculate the transmission delay time

(a) Delay time in cyclic transmission

Calculation formulas for cyclic transmission delay time (time required for data to be delivered) are shown below.

1) For a single network system

Table 5.12 Calculation formulas for transmission delay time

Calculated value		Transmission delay when L_T > LS [ms]	
		Block data assurance per station enabled	Block data assurance per station disabled
Normal value	*1	$(L_T + \beta_T) \times 1.5 + LS \times 0.5 + (L_R + \beta_R) \times 0.5$	$L_T + \beta_T + LS \times 0.5 + (L_R + \beta_R) \times 0.5$
	*2	$(L_T + \beta_T) \times 1.5 + LS \times 0.5 + (S_R + \alpha_R) \times 1.5$	$L_T + \beta_T + LS \times 0.5 + (S_R + \alpha_R) \times 1.5$
Max. value	*1	$(L_T + \beta_T) \times 2 + LS \times 1 + L_R + \beta_R$	$L_T + \beta_T + LS \times 1 + L_R + \beta_R$
	*2	$(L_T + \beta_T) \times 2 + LS \times 1 + (S_R + \alpha_R) \times 2$	$L_T + \beta_T + LS \times 1 + (S_R + \alpha_R) \times 2$

Calculated value		Transmission delay when L_T < LS [ms]	
		Block data assurance per station enabled	Block data assurance per station disabled
Normal value	*1	$(L_T + \beta_T) + LS \times 1 + (L_R + \beta_R) \times 0.5$	$L_T + \beta_T + LS \times 0.5 + (L_R + \beta_R) \times 0.5$
	*2	$(L_T + \beta_T) + LS \times 1 + (S_R + \alpha_R) \times 1.5$	$L_T + \beta_T + LS \times 0.5 + (S_R + \alpha_R) \times 1.5$
Max. value	*1	$(L_T + \beta_T) + LS \times 2 + L_R + \beta_R$	$L_T + \beta_T + LS \times 1 + L_R + \beta_R$
	*2	$(L_T + \beta_T) + LS \times 2 + (S_R + \alpha_R) \times 2$	$L_T + \beta_T + LS \times 1 + (S_R + \alpha_R) \times 2$

* 1 When a C Controller module (another CPU) receives data
* 2 When a programmable controller CPU receives data

L_T : Link device refresh cycle of C Controller module (sending side)

L_R : Link device refresh cycle of C Controller module (receiving side)

β_T : Total link device refresh time of C Controller module (sending side)*3 (☞ (1) (b) in this section)

β_R : Total link device refresh time of C Controller module (receiving side)*3 (☞ (1) (b) in this section)

S_R : Scan time (excluding the link refresh time) of the programmable controller CPU (receiving side)*4

α_R : Link refresh time of the programmable controller CPU (receiving side)*3, *4

LS: Link scan time*4

* 3 Total time for the number of mounted CC-Link IE controller network modules.
* 4 For details, refer to the MELSEC-Q CC-Link IE Controller Network Reference Manual.

2) For a multi-network system

Transferring link device data to another network using the interlink transfer function will affect the transfer processing time of CPU modules on relay stations.

For details, refer to the MELSEC-Q CC-Link IE Controller Network Reference Manual.

(b) Transmission delay time calculation example
The following shows a calculation example for cyclic transmission delay time in a single network system.

(System configuration and conditions)

Table 5.13 System configuration and conditions

Item		Description
CPU on receiving station		C Controller module
Total number of stations per network		2 (Control station: 1, Normal station: 1)
Total link device points	LB/LW	1024 points
	LX/LY	0 points
	SB/SW	0 points
Link device refresh cycle		100ms
Transient transmission		None
Block data assurance per station		Enabled
CC-Link IE controller network module		Installed to Slot 0 of main base unit
Faulty station		None

1) Link device refresh time

$$\text{Total link device refresh time} = \text{KM1} + \text{KM2} \times \left\{ \left(\frac{\text{LB}+\text{LX}+\text{LY}}{16} \right) + \text{LW} \right\}$$

· Total link device refresh time on sending side

$$\beta_{\text{T}} = 1.0 + 0.36 \times 10^{-3} \times \left(\frac{1024+0+0}{1} + 1024 \right)$$
$$\fallingdotseq 1.39(\text{ms})$$

· Total link device refresh time on receiving side

$$\beta_{\text{R}} \fallingdotseq 1.39(\text{ms})$$

2) Link scan time

$$\begin{aligned} &\text{Link scan time LS}^{*1} \\ &= [\text{KB} + (\text{n}\times 56) + \{\text{LB} + \text{LY} + (\text{LW}\times 16)\} \div 8 \times 0.016 \\ &\quad + (\text{N}_{\text{T}} \times \text{T} \times 30)] \div 1000 + \text{N}_{\text{C}} \\ &= [1100 + (2\times 56) + \{1024 + 0 + (1024\times 16)\} \div 8 \times 0.016 \\ &\quad + (0\times 2\times 30)] \div 1000 \\ &\fallingdotseq 1.25(\text{ms}) \end{aligned}$$

* 1 For details of the link scan time, refer to the MELSEC-Q CC-Link IE Controller Network Reference Manual.

3) Cyclic transmission delay

Table 5.14 Example of transmission delay time calculation

Calculated value	Transmission delay when L_T > LS [ms]
	Block data assurance per station enabled
Normal value	$(L_T + \beta_T) \times 1.5 + LS \times 0.5 + (L_R + \beta_R) \times 0.5$ (100 + 1.39) × 1.5 + 1.25 × 0.5 + (100 + 1.39) × 0.5 = 203.405[ms]
Max. value	$(L_T + \beta_T) \times 2 + LS \times 1 + L_R + \beta_R$ (100 + 1.39) × 2 + 1.25 × 1 + 100 + 1.39 = 305.42[ms]

(3) Reducing the link device refresh time

The link device refresh time can be reduced by decreasing the number of refresh points for the C Controller module. To decrease the refresh points, modify the refresh and network range assignment parameter settings and use the direct access method for accessing link devices.

(a) Concept of refresh range (points)

Within the range set with refresh parameters in the <<Parameter settings>> tab of CC IE Control utility, the areas set in Network range assignment are refreshed.

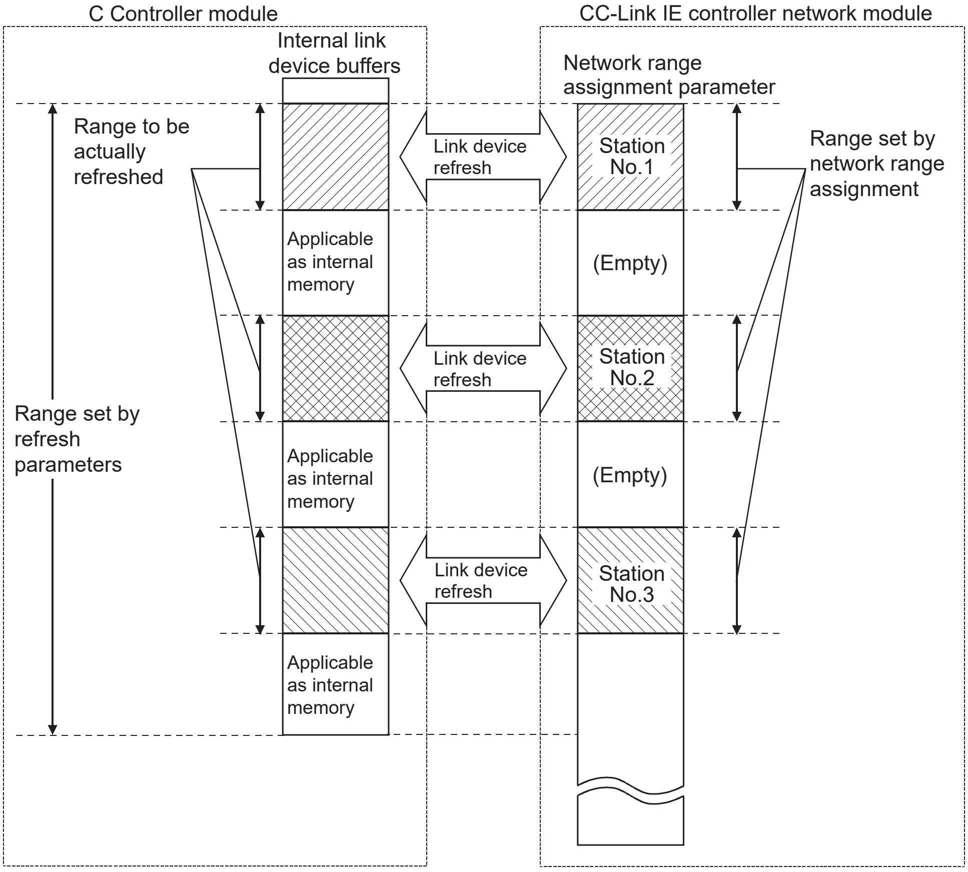

C Controller module
Internal link device buffers
Range to be actually refreshed
Range set by refresh parameters
Applicable as internal memory
Applicable as internal memory
Applicable as internal memory
Link device refresh
Link device refresh
Link device refresh
CC-Link IE controller network module
Network range assignment parameter
Station No.1
(Empty)
Station No.2
(Empty)
Station No.3
Range set by network range assignment

Figure 5.59 Concept of refresh range

☒POINT

The range from the first address to the last address is set by default in the refresh parameter setting.
(To bring the setting back to the default, click the Default button on the Refresh parameter settings screen activated from the <<Parameter settings>> tab of CC IE Control utility.)

(b) How to decrease the refresh points

1) Modifying refresh parameters

Up to 64 refresh ranges can be set for the C Controller module.

For refresh parameters, set only the ranges that need to be refreshed.

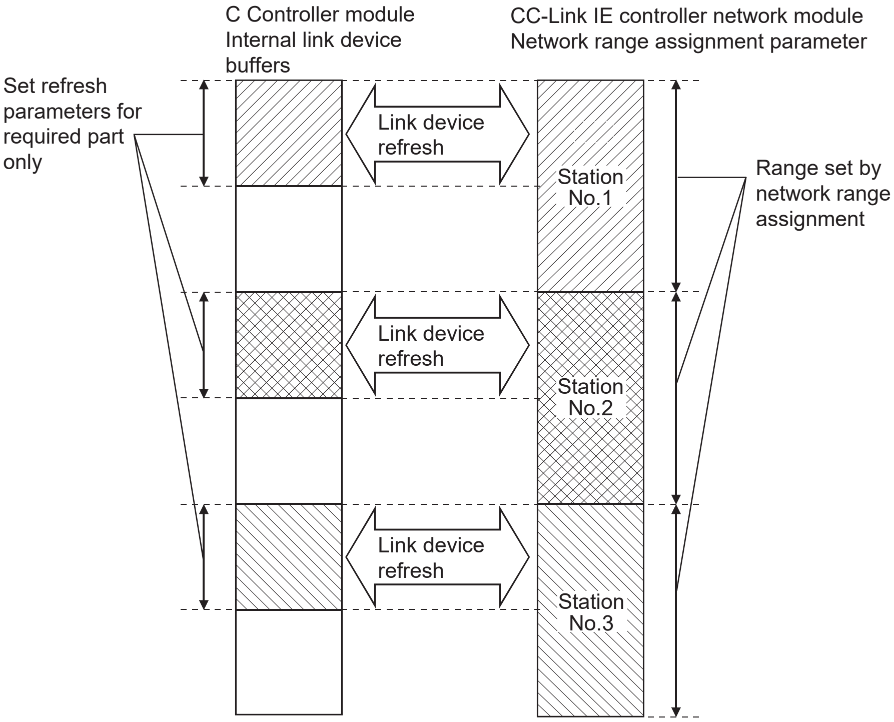

C Controller module
Internal link device buffers
CC-Link IE controller network module
Network range assignment parameter
Set refresh parameters for required part only
Link device refresh
Link device refresh
Link device refresh
Station No.1
Station No.2
Station No.3
Range set by network range assignment

Figure 5.60 Modifying refresh parameters

2) Using direct access

By using direct access for link devices that are infrequently used on its own station and excluding these from the link device refresh ranges, the link device refresh time can be reduced.

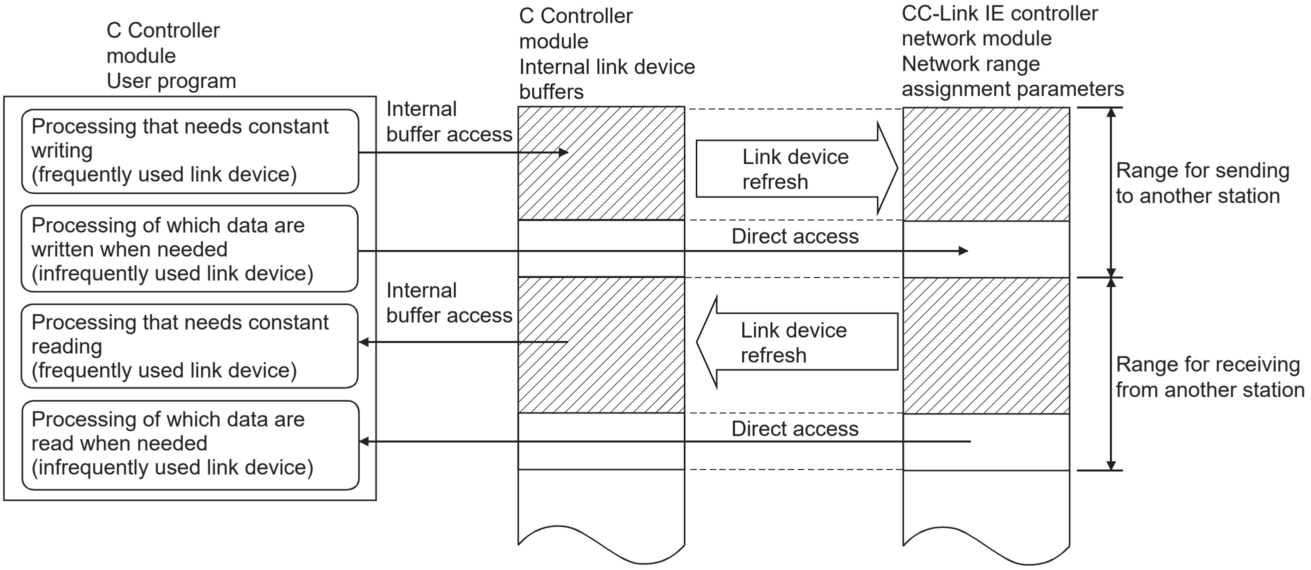

C Controller module User program
Processing that needs constant writing (frequently used link device)
Processing of which data are written when needed (infrequently used link device)
Processing that needs constant reading (frequently used link device)
Processing of which data are read when needed (infrequently used link device)
Internal buffer access
Internal buffer access
C Controller module Internal link device buffers
Link device refresh
Direct access
Link device refresh
Direct access
CC-Link IE controller network module Network range assignment parameters
Range for sending to another station
Range for receiving from another station

Figure 5.61 Reducing link device refresh time

CHAPTER 6 PREPARATORY PROCEDURES AND SETTING

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	○

This chapter explains the preparatory procedures and setting methods for operating the C Controller module in the system.

☒POINT

Before handling the C Controller module, read the safety precautions provided in the beginning of this manual.

6.1 Handling Precautions

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	○

This section explains the handling precautions for the C Controller module.

⚠CAUTION

- Use the C Controller module in an environment that meets the general specifications shown in this manual.
  Failure to do so may result in an electric shock, fire, malfunction, or damage to or deterioration of the product.
- To mount the module, while pressing the module mounting lever located in the lower part of module, fully insert the module fixing projection into the hole in the base unit and press the module until it snaps into place.
  Incorrect mounting may cause malfunction, failure or drop of the module.
  When using the C Controller module in an environment of frequent vibrations, fix the module with a screw.
  Tighten the screw within the specified torque range.
  Undertightening can cause drop of the screw, short circuit or malfunction.
  Overtightening can damage the screw and/or module, resulting in drop, short circuit, or malfunction.
- When using an extension cable, connect it to the connector of the base unit or module securely.
  Check the connection for looseness.
  Poor contact may cause incorrect input or output.
- When using a CompactFlash card, fully insert it into the CompactFlash card slot.
  Check that it is inserted completely.
  Poor contact may cause malfunction.
- Shut off the external power supply for the system in all phases before mounting or removing the module.
  Failure to do so could result in damage to the product.
- Do not directly touch any conductive part of the module.
  Doing so can cause malfunction or failure of the module.

☒POINT

Referring to the following manual, observe the proper mounting procedures.
☞ QCPU User's Manual (Hardware Design, Maintenance and Inspection)

- The case of the C Controller module is made of resin.
  Do not drop or apply strong shock to it.
- Do not remove the printed-circuit board of the module from the case.
  Doing so may cause failure.
- Tighten screws such as a module fixing screw within the following ranges.

Table 6.1 Tightening torque ranges

Screw	Tightening torque range	Remarks
Module fixing screw (normally not required) (M3 screw)[*1]	0.36 to 0.48N•m	—
RS-232 cable connector screw (M2.6 screw) for Q12DCCPU-CBL and Q06CCPU-V(-B)	0.20 to 0.39N•m	Depth of screw hole: L=3.2mm (0.13 inches) or less (Internal length from the surface)

[*1] The module can be easily secured to a base unit by the hook on the module top. However, it is recommended to fix it with the module fixing screw in a place of frequent vibrations.

- Use the dummy CompactFlash card installed before shipping if necessary.
  Do not insert a dummy CompactFlash card with no "A" printed (Figure 6.1) into a module whose serial number (first four digits) is "1008" or later. Doing so may make the card removal difficult, causing damage to the CompactFlash card slot.
  A dummy CompactFlash card with "A" (Figure 6.2) is pre-installed in a module whose serial number (first four digits) is "1008" or later.

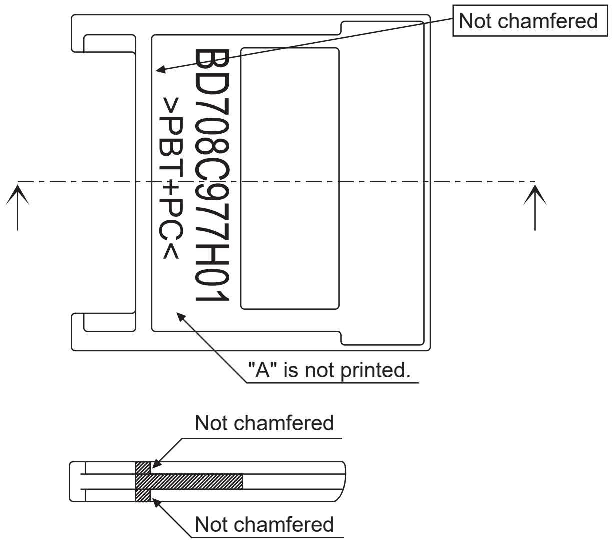


Figure 6.1 Dummy CompactFlash card (without "A")

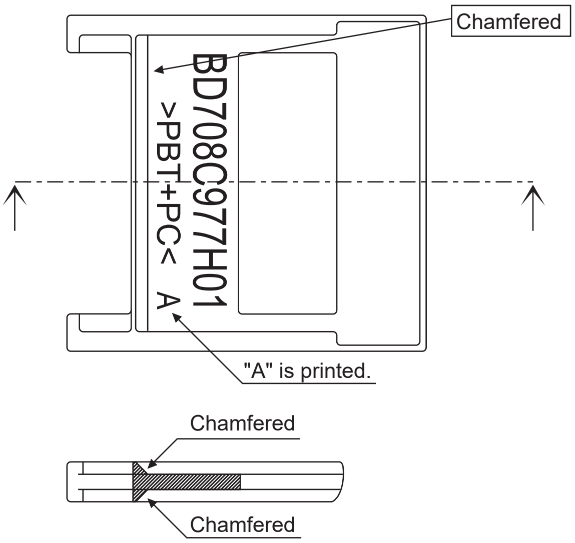


Figure 6.2 Dummy CompactFlash card (with "A")

6.2 Fail-Safe Circuit

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	○

⚠ **WARNING**

- Configure safety circuits external to the C Controller module to ensure that the entire system operates safely even when a fault occurs in the external power supply or the C Controller module. Failure to do so may result in an accident due to an incorrect output or malfunction.
  (1) Configure external safety circuits, such as an emergency stop circuit, protection circuit, and protective interlock circuit for forward/reverse operation or upper/lower limit positioning.
  (2) If the following status (a) or (b) occurs, the system will behave accordingly.
    (a) When overcurrent or overvoltage protection of the power supply module is activated, the outputs (Y) from the user program and writing to the buffer memory are disabled, and all outputs are turned off.
    (b) When the C Controller module detects an error such as a watchdog timer error by the self-diagnostic function, the outputs (Y) from the user program and writing to the buffer memory are disabled. Whether to hold or turn off all outputs can be set by parameters.

    All outputs may turn on when an error occurs in the part, such as I/O control part, where the C Controller module cannot detect any error. To ensure safety operation in such a case, provide a safety mechanism or a fail-safe circuit external to the C Controller module.
  (3) Outputs may remain on or off due to a failure of an output module relay or transistor.
  Configure an external circuit for monitoring output signals that could cause a serious accident.
- In an output module, when a load current exceeding the rated current or an overcurrent caused by a load short-circuit flows for a long time, it may cause smoke and fire. To prevent this, configure an external safety circuit, such as a fuse.
- Configure a circuit so that the C Controller system is turned on first and then the external power supply. If the external power supply is turned on first, an accident may occur due to an incorrect output or malfunction.
- For the operating status of each station after a communication failure, refer to the relevant manuals for the network.
  Incorrect output or malfunction may result in an accident.

⚠WARNING

- For the following controls, configure an interlock circuit in the user program to ensure that the entire system will always operate safely.
  (1) Changing data of a running C Controller module from the development environment (personal computer) connected
  (2) Changing the operating status
  (3) Operation from the development environment (personal computer)
  Especially, in the case of control from an external device to a remote C Controller module, immediate action cannot be taken for a problem on the C Controller module due to a communication failure.
  To prevent this, configure an interlock circuit in the user program and determine corrective actions to be taken between the external device and C Controller module in case of a communication failure.

⚠CAUTION

- Do not install the control lines or communication cables together with the main circuit lines or power cables.
  Keep a distance of 100mm (3.94 inches) or more between them.
  Failure to do so may result in malfunction due to noise.
- When a device such as a lamp, heater, or solenoid valve is controlled through an output module, a large current (approximately ten times greater than normal) may flow when the output is turned from off to on. Take measures such as replacing the module with one having a sufficient current rating.
- When the C Controller module is powered off to on or reset, the time for the C Controller to reach the RUN status varies depending on the system configuration, parameter setting or program capacity.
  Design the system so that the whole system operates at the safe side even when the time required for the C Controller module to reach the RUN status varies.

When the C Controller system is powered on/off, some control output may not be normal for a moment because of a delay or a rise time difference between the C Controller system power supply and the external power supply (especially DC) for the control target.

In the case of a DC output module, for example, when power is applied to the control target first and then to the C Controller system, the DC output module may temporarily generate incorrect output at power-on of the C Controller system.
Therefore, it is necessary to configure a circuit by which power is applied to the C Controller system first.
Also, if the external power supply or C Controller module is faulty, an abnormal operation may occur.

To prevent such abnormal operation from resulting in failure of the entire system, configure fail-safe circuits external to the C Controller module for the areas where an abnormal operation will lead to mechanical damage and/or accidents (emergency stop circuit, protection circuit, interlock circuit, etc.)

A circuit example for system design is provided on the next page.

(1) Circuit example for system design (when not using the ERR. contact of the power supply module)

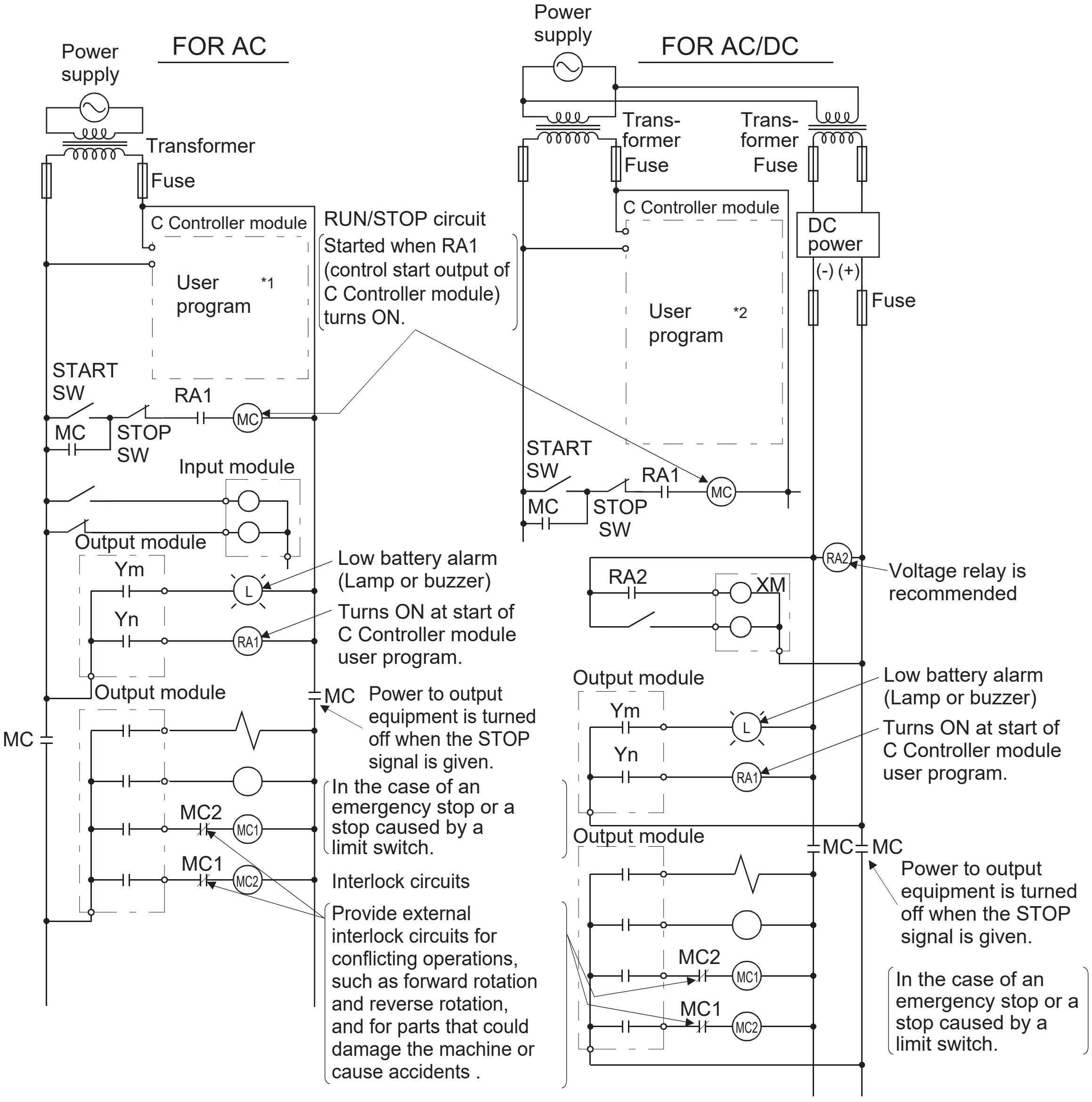

FOR AC
Power supply
Transformer
Fuse
C Controller module
User program *1
RUN/STOP circuit
Started when RA1 (control start output of C Controller module) turns ON.
START SW
RA1
MC
MC
STOP SW
Input module
Output module
Ym
Yn
L
RA1
Low battery alarm (Lamp or buzzer)
Turns ON at start of C Controller module user program.
Output module
MC
MC
Power to output equipment is turned off when the STOP signal is given.
In the case of an emergency stop or a stop caused by a limit switch.
MC2
MC1
MC1
MC2
Interlock circuits
Provide external interlock circuits for conflicting operations, such as forward rotation and reverse rotation, and for parts that could damage the machine or cause accidents .
FOR AC/DC
Power supply
Trans-former
Trans-former
Fuse
Fuse
C Controller module
DC power
(-) (+)
Fuse
User program *2
START SW
RA1
MC
MC
STOP SW
RA2
RA2
XM
Voltage relay is recommended
Output module
Ym
Yn
L
RA1
Low battery alarm (Lamp or buzzer)
Turns ON at start of C Controller module user program.
Output module
MC
MC
Power to output equipment is turned off when the STOP signal is given.
MC2
MC1
MC1
MC2
In the case of an emergency stop or a stop caused by a limit switch.

Figure 6.3 Circuit example for system design (when not using the ERR. contact of the power supply module)

The power-up procedure is as follows:

For AC

[1] Turn on the power.
[2] Set the C Controller module to "RUN".
[3] Turn on the start switch.
[4] When the magnetic contactor (MC) turns on, the output equipment is driven by the user program.*1

For AC/DC

[1] Turn on the power.
[2] Set the C Controller module to "RUN".
[3] When DC power is applied, RA2 is turned on.
[4] When RA2 turns on, XM is turned on.
Upon 100% establishment of the DC input signal, processing is started by the user program.*2
[5] Turn on the start switch.
[6] When the magnetic contact (MC) turns on, the output equipment is driven by the user program.

* 1 Program the system so that the following operation will be performed at start-up of the C Controller module.

1) When a battery voltage drop is detected, Ym is turned on.
In the program, Ym must be turned on by the QBF_Y_Out_BitEx function when "Built-in battery error status" of the QBF_ReadStatusEx function is set to 1 (battery error).

2) Yn is turned on at start of the user program.
In the program, Yn is turned on to release the interlock at start of the user program of the C Controller module.

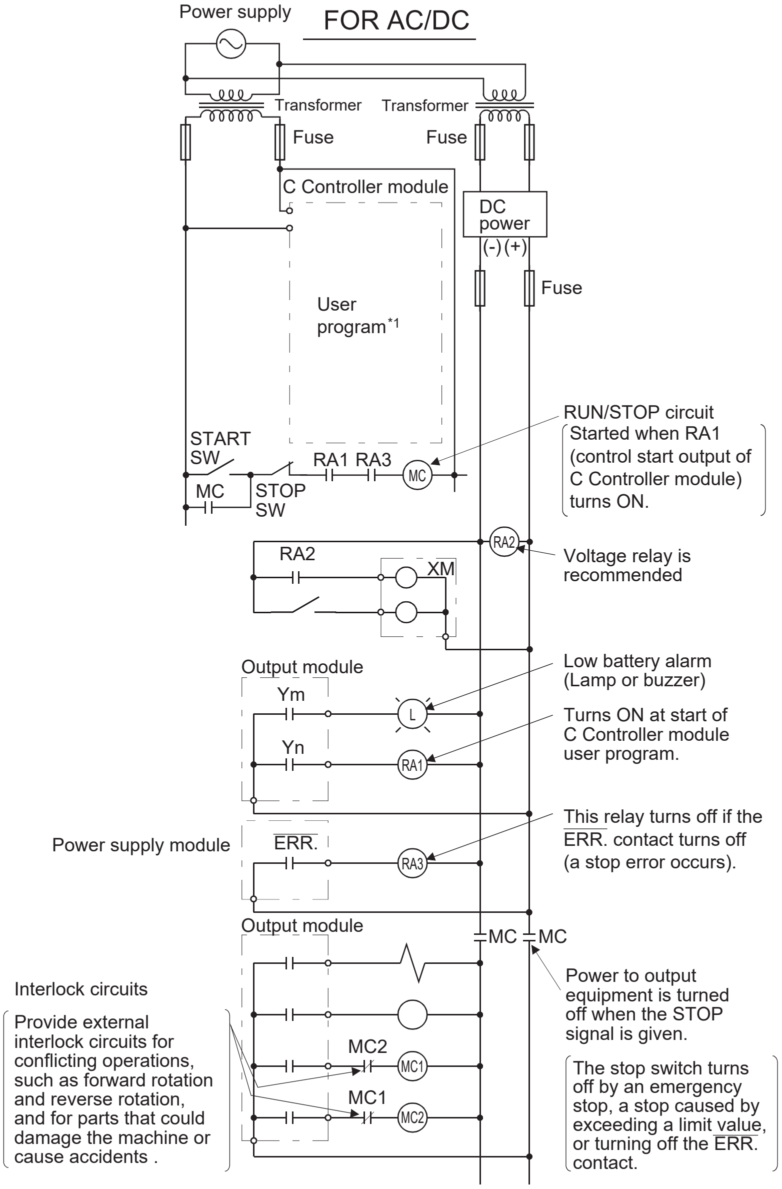
* 2 Program the system so that the following operation will be performed at start-up of the C Controller module.

1) When a battery voltage drop is detected, Ym is turned on.
In the program, Ym must be turned on by the QBF_Y_Out_BitEx function when "Built-in battery error status" of the QBF_ReadStatusEx function is set to 1 (battery error).

2) Yn is turned on at start of the user program.
In the program, Yn must be turned on to release the interlock at start of the user program of the C Controller module.

3) After XM turns on and the DC input signal is established, system processing is started.
Create a program so that system processing will start when the DC input signal establishment time has elapsed after turn-on of XM (the status can be confirmed by the QBF_X_In_BitEx function).
(The DC input signal establishment time is the time taken from "on" of RA2 to 100% establishment of the DC input signal. Set this time to 0.5 seconds.)
However, when a voltage relay is used for RA2, the DC input signal establishment time need not be set in the user program.

(2) Circuit example for system design (When using the $\overline{\text{ERR.}}$ contact of the power supply module)



RUN/STOP circuit
Started when RA1 (control start output of C Controller module) turns ON.

Voltage relay is recommended

Low battery alarm (Lamp or buzzer)

Turns ON at start of C Controller module user program.

This relay turns off if the $\overline{\text{ERR}}$. contact turns off (a stop error occurs).

Power to output equipment is turned off when the STOP signal is given.

The stop switch turns off by an emergency stop, a stop caused by exceeding a limit value, or turning off the $\overline{\text{ERR}}$. contact.

Provide external interlock circuits for conflicting operations, such as forward rotation and reverse rotation, and for parts that could damage the machine or cause accidents .

Figure 6.4 Circuit example for system design (when using the $\overline{\text{ERR.}}$ contact of the power supply module)

The power-up procedure is as follows:

For AC/DC

[1] Turn on the power.
[2] Set the C Controller module to "RUN".
[3] When DC power is applied, RA2 is turned on.
[4] When RA2 turns on, XM is turned on.
Upon 100% establishment of the DC input signal, processing is started by the user program.*1
[5] Turn on the start switch.
[6] When the magnetic contactor (MC) turns on, the output equipment is driven by the user program.

* 1 Program the system so that the following operation will be performed at start-up of the C Controller module.

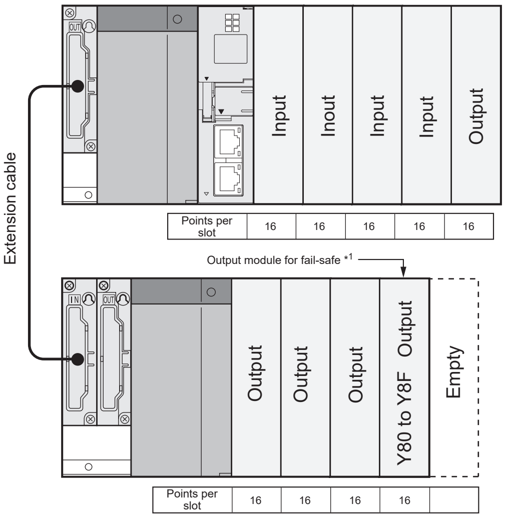
1) When a battery voltage drop is detected, Ym is turned on.
In the program, Ym must be turned on by the QBF_Y_Out_BitEx function when "Built-in battery error status" of the QBF_ReadStatusEx function is set to 1 (battery error).

2) Yn is turned on at start of the user program.
In the program, Yn must be turned on to release the interlock at start of the user program of the C Controller module.

3) After XM turns on and the DC input signal is established, system processing is started.
Create a program so that system processing will start when the DC input signal establishment time has elapsed after turn-on of XM (the status can be confirmed by the QBF_X_In_BitEx function).
(The DC input signal establishment time is the time taken from "on" of RA2 to 100% establishment of the DC input signal. Set this time to 0.5 seconds.)
However, when a voltage relay is used for RA2, the DC input signal establishment time need not be set in the user program.

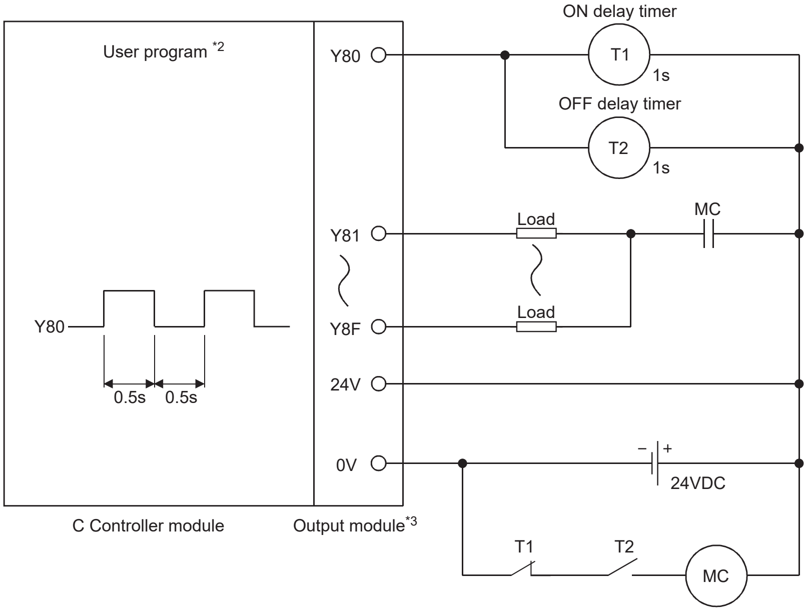
(3) Fail-safe measures against C Controller system failure

The C Controller module can detect its own failure by the self-diagnostic function. However, it may not be able to detect a failure that occurred in an I/O control part. In such a case, all of inputs and outputs may turn on or off depending on the failure, and normal operation of the control target and safety may not be ensured.
Though Mitsubishi products are manufactured under strict quality control, it is recommended to configure external fail-safe circuits so that a C Controller system failure will not result in mechanical damage or accidents.
The following shows a system example and a fail-safe circuit example.



* 1 The output module for fail-safe must be installed to the last slot of the system. (Y80 to Y8F in the above system)

Figure 6.5 System example



* 2 Create a program so that Y80 alternates between ON and OFF at intervals of 0.5 seconds. (Use the QBF_Y_Out_BitEx function to set Y80 to on/off.)
* 3 Since Y80 turns on and off at 0.5s intervals, use a non-contact output module (a transistor in the above example).

Figure 6.6 Fail-safe circuit example

6.3 Preparatory Procedure and Setting

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	○

☒POINT

When using Q12DCCPU-V (Extended mode), refer to the following manual.

☞ MELSEC-Q C Controller Module User's Manual

(1) Starting up the single CPU system

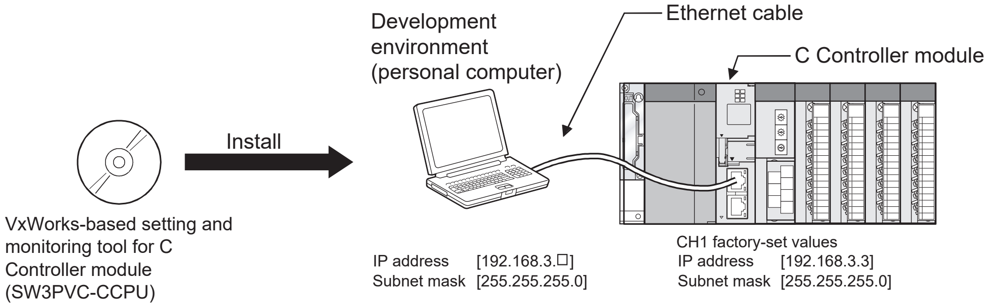
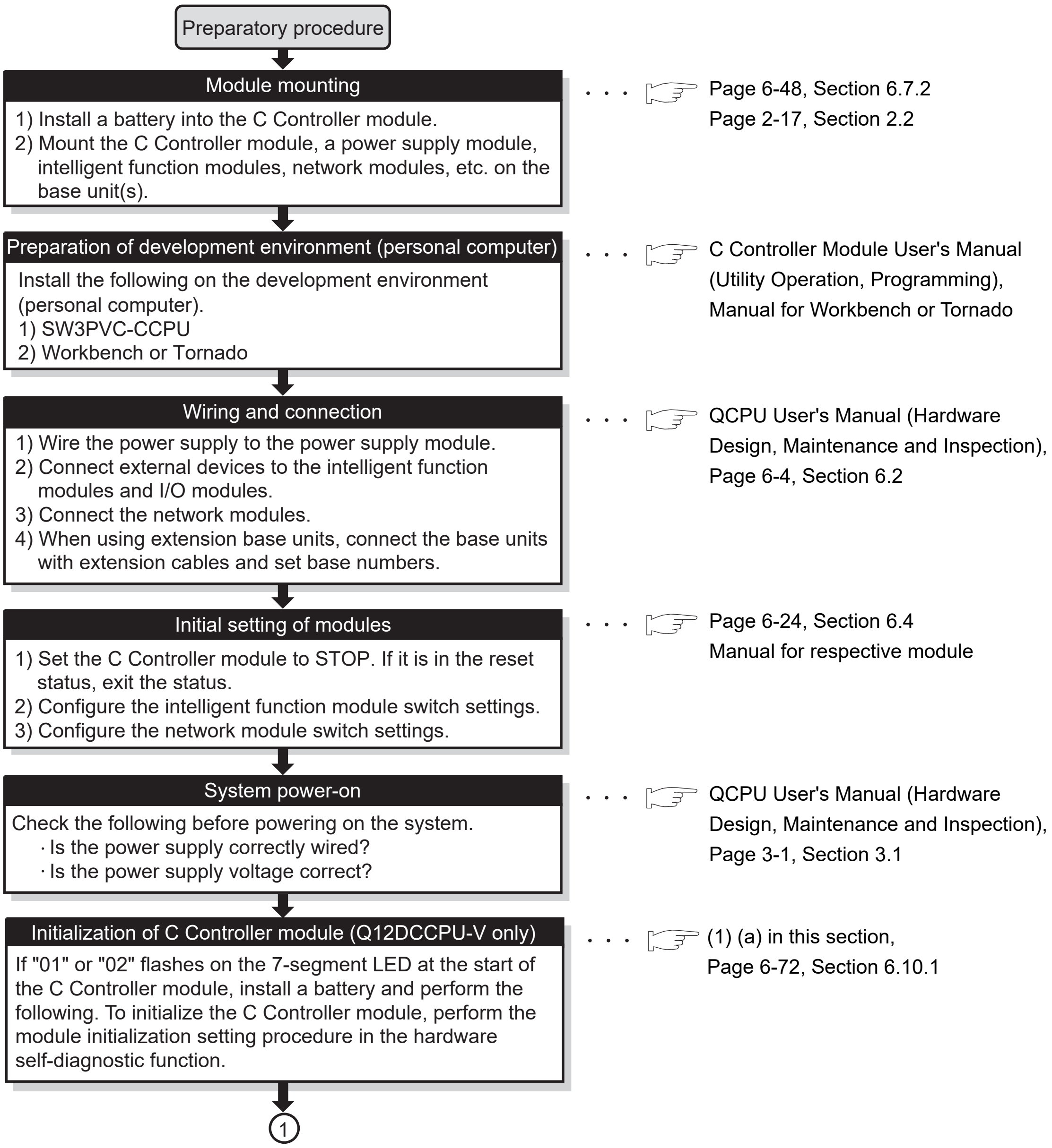

Development environment (personal computer)
Ethernet cable
C Controller module
Install
VxWorks-based setting and monitoring tool for C Controller module (SW3PVC-CCPU)
IP address [192.168.3.□]
Subnet mask [255.255.255.0]
CH1 factory-set values
IP address [192.168.3.3]
Subnet mask [255.255.255.0]

Figure 6.7 Configuration at single CPU system startup



Preparatory procedure

Module mounting

1) Install a battery into the C Controller module.
2) Mount the C Controller module, a power supply module, intelligent function modules, network modules, etc. on the base unit(s).

. . . ☞ Page 6-48, Section 6.7.2
Page 2-17, Section 2.2

Preparation of development environment (personal computer)

Install the following on the development environment (personal computer).
1) SW3PVC-CCPU
2) Workbench or Tornado

. . . ☞ C Controller Module User's Manual (Utility Operation, Programming), Manual for Workbench or Tornado

Wiring and connection

1) Wire the power supply to the power supply module.
2) Connect external devices to the intelligent function modules and I/O modules.
3) Connect the network modules.
4) When using extension base units, connect the base units with extension cables and set base numbers.

. . . ☞ QCPU User's Manual (Hardware Design, Maintenance and Inspection), Page 6-4, Section 6.2

Initial setting of modules

1) Set the C Controller module to STOP. If it is in the reset status, exit the status.
2) Configure the intelligent function module switch settings.
3) Configure the network module switch settings.

. . . ☞ Page 6-24, Section 6.4
Manual for respective module

System power-on

Check the following before powering on the system.
· Is the power supply correctly wired?
· Is the power supply voltage correct?

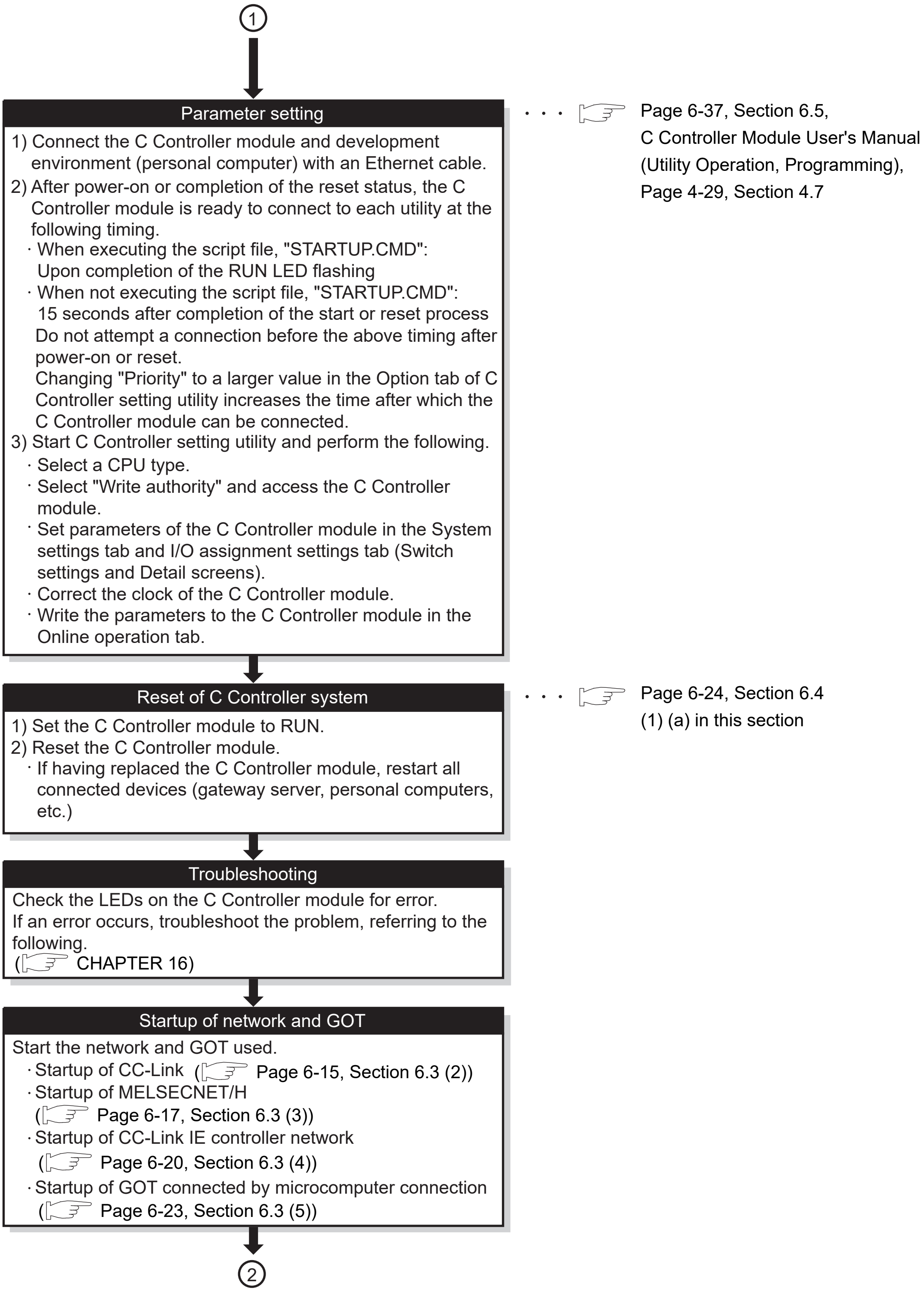
. . . ☞ QCPU User's Manual (Hardware Design, Maintenance and Inspection), Page 3-1, Section 3.1

Initialization of C Controller module (Q12DCCPU-V only)

If "01" or "02" flashes on the 7-segment LED at the start of the C Controller module, install a battery and perform the following. To initialize the C Controller module, perform the module initialization setting procedure in the hardware self-diagnostic function.

. . . ☞ (1) (a) in this section, Page 6-72, Section 6.10.1

Figure 6.8 Flowchart for single CPU system startup



1

Parameter setting

1) Connect the C Controller module and development environment (personal computer) with an Ethernet cable.
2) After power-on or completion of the reset status, the C Controller module is ready to connect to each utility at the following timing.
   - When executing the script file, "STARTUP.CMD": Upon completion of the RUN LED flashing
   - When not executing the script file, "STARTUP.CMD": 15 seconds after completion of the start or reset process

   Do not attempt a connection before the above timing after power-on or reset.
   Changing "Priority" to a larger value in the Option tab of C Controller setting utility increases the time after which the C Controller module can be connected.
3) Start C Controller setting utility and perform the following.
   - Select a CPU type.
   - Select "Write authority" and access the C Controller module.
   - Set parameters of the C Controller module in the System settings tab and I/O assignment settings tab (Switch settings and Detail screens).
   - Correct the clock of the C Controller module.
   - Write the parameters to the C Controller module in the Online operation tab.

. . . Page 6-37, Section 6.5,
C Controller Module User's Manual
(Utility Operation, Programming),
Page 4-29, Section 4.7

Reset of C Controller system

1) Set the C Controller module to RUN.
2) Reset the C Controller module.
   - If having replaced the C Controller module, restart all connected devices (gateway server, personal computers, etc.)

. . . Page 6-24, Section 6.4
(1) (a) in this section

Troubleshooting

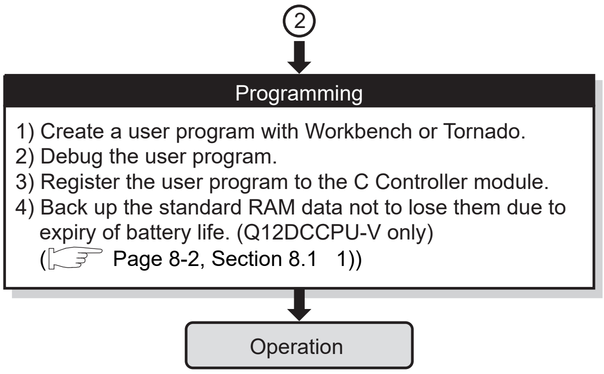
Check the LEDs on the C Controller module for error.
If an error occurs, troubleshoot the problem, referring to the following.
(CHAPTER 16)

Startup of network and GOT

Start the network and GOT used.
- Startup of CC-Link (Page 6-15, Section 6.3 (2))
- Startup of MELSECNET/H (Page 6-17, Section 6.3 (3))
- Startup of CC-Link IE controller network (Page 6-20, Section 6.3 (4))
- Startup of GOT connected by microcomputer connection (Page 6-23, Section 6.3 (5))

2

Figure 6.8 Flowchart for single CPU system startup (continued)



1) Create a user program with Workbench or Tornado.
2) Debug the user program.
3) Register the user program to the C Controller module.
4) Back up the standard RAM data not to lose them due to expiry of battery life. (Q12DCCPU-V only)
(☞ Page 8-2, Section 8.1 1))

· · · ☞ C Controller Module User's Manual (Utility Operation, Programming)

Figure 6.8 Flowchart for single CPU system startup (continued)

☒POINT

When a connection to the C Controller module cannot be established although "192.168.3.3 (default)" is set in Connection settings, check the network settings of the C Controller module and development environment (personal computer).
(☞ Page 6-39, Section 6.6)

(a) Precautions

1) Flashing "01" or "02" of the 7-segment LED (Q12DCCPU-V only)
At start-up of the C Controller module, "01" or "02" flashes on the 7-segment LED when the C Controller module is:
    - In factory-default state.
    (The standard RAM is not initialized.)
    - Left unused for a long period of time with no battery installed.
    (The standard and battery-backed-up RAM data are corrupted.)

If "01" or "02" flashes on the 7-segment LED, install a battery and perform the module initialization setting procedure.

2) When having replaced the C Controller module
Restart all connected devices (gateway server, personal computers, etc.)
(If any of these devices retains an Ethernet address of its communication target, continued communication may be disabled after the replacement because the Ethernet address is changed.)
Likewise, restart the C Controller module when a target device has been replaced.

(2) Starting up the CC-Link

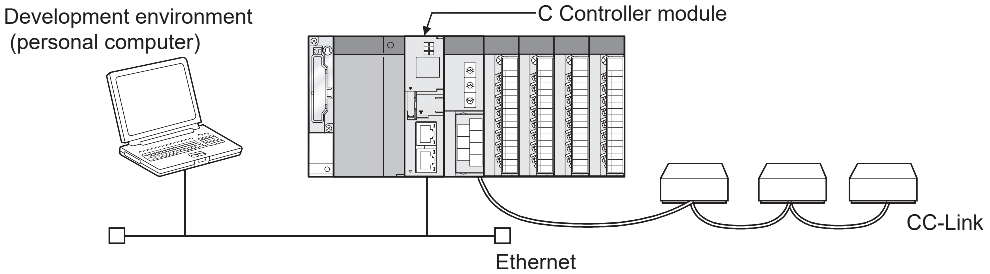
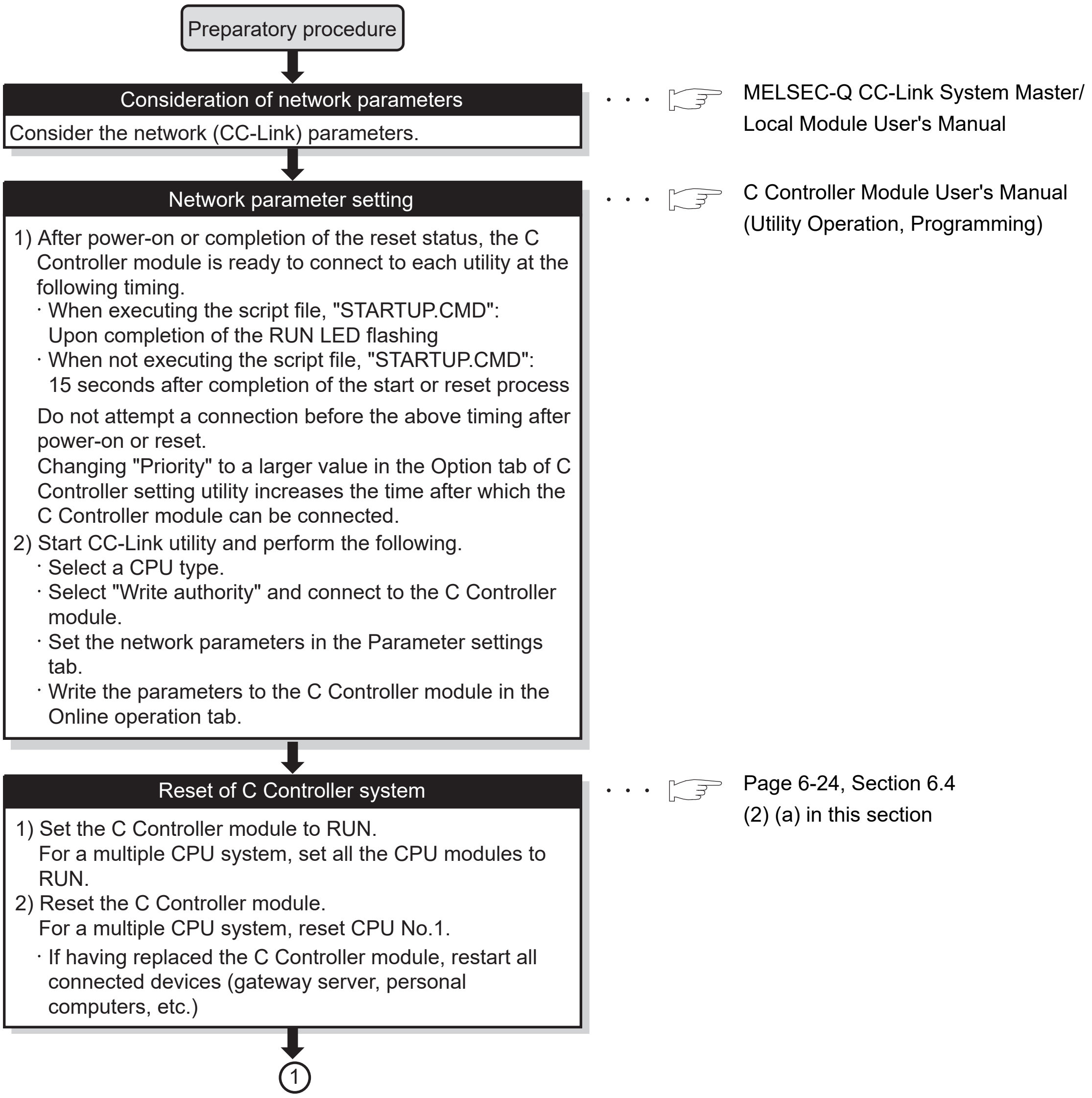


Figure 6.9 Configuration at CC-Link startup



Preparatory procedure

Consideration of network parameters

Consider the network (CC-Link) parameters.

. . . ☞ MELSEC-Q CC-Link System Master/ Local Module User's Manual

Network parameter setting

1) After power-on or completion of the reset status, the C Controller module is ready to connect to each utility at the following timing.
   - · When executing the script file, "STARTUP.CMD": Upon completion of the RUN LED flashing
   - · When not executing the script file, "STARTUP.CMD": 15 seconds after completion of the start or reset process

   Do not attempt a connection before the above timing after power-on or reset.
   Changing "Priority" to a larger value in the Option tab of C Controller setting utility increases the time after which the C Controller module can be connected.
2) Start CC-Link utility and perform the following.
   - · Select a CPU type.
   - · Select "Write authority" and connect to the C Controller module.
   - · Set the network parameters in the Parameter settings tab.
   - · Write the parameters to the C Controller module in the Online operation tab.

. . . ☞ C Controller Module User's Manual (Utility Operation, Programming)

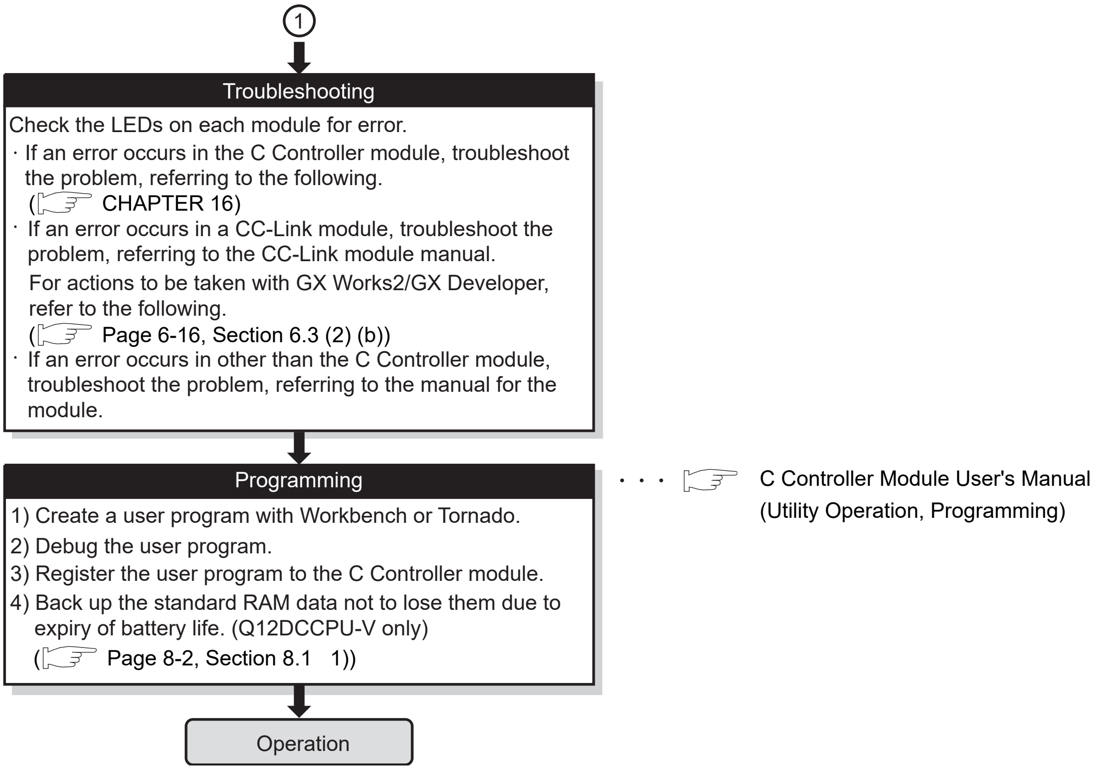
Reset of C Controller system

1) Set the C Controller module to RUN.
   For a multiple CPU system, set all the CPU modules to RUN.
2) Reset the C Controller module.
   For a multiple CPU system, reset CPU No.1.
   - · If having replaced the C Controller module, restart all connected devices (gateway server, personal computers, etc.)

. . . ☞ Page 6-24, Section 6.4 (2) (a) in this section

1

Figure 6.10 Flowchart for CC-Link startup



1

Troubleshooting

Check the LEDs on each module for error.

- If an error occurs in the C Controller module, troubleshoot the problem, referring to the following. (☞ CHAPTER 16)
- If an error occurs in a CC-Link module, troubleshoot the problem, referring to the CC-Link module manual. For actions to be taken with GX Works2/GX Developer, refer to the following. (☞ Page 6-16, Section 6.3 (2) (b))
- If an error occurs in other than the C Controller module, troubleshoot the problem, referring to the manual for the module.

Programming

1) Create a user program with Workbench or Tornado.
2) Debug the user program.
3) Register the user program to the C Controller module.
4) Back up the standard RAM data not to lose them due to expiry of battery life. (Q12DCCPU-V only) (☞ Page 8-2, Section 8.1 1))

・・・ ☞ C Controller Module User's Manual (Utility Operation, Programming)

Operation

Figure 6.10 Flowchart for CC-Link startup (continued)

(a) Precautions

When having replaced a C Controller module, restart all connected devices (gateway server, personal computers, etc.)

(If any of these devices retains an Ethernet address of its communication target, continued communication may be disabled after the replacement because the Ethernet address is changed.)

Likewise, restart a C Controller module when a target device has been replaced.

(b) GX Works2/GX Developer and SW3PVC-CCPU

In troubleshooting according to the CC-Link module manual, actions to be taken from GX Works2/GX Developer are as follows:

Table 6.2 GX Works2/GX Developer and SW3PVC-CCPU

GX Works2/GX Developer	SW3PVC-CCPU
I/O assignment setting	Use the I/O assignment settings tab of C Controller setting utility.
Intelligent function module switch setting	Use the Switch setting for I/O and intelligent function module setting screen of C Controller setting utility.
Sequence program check	Check the user program for error.

(3) Starting up the MELSECNET/H

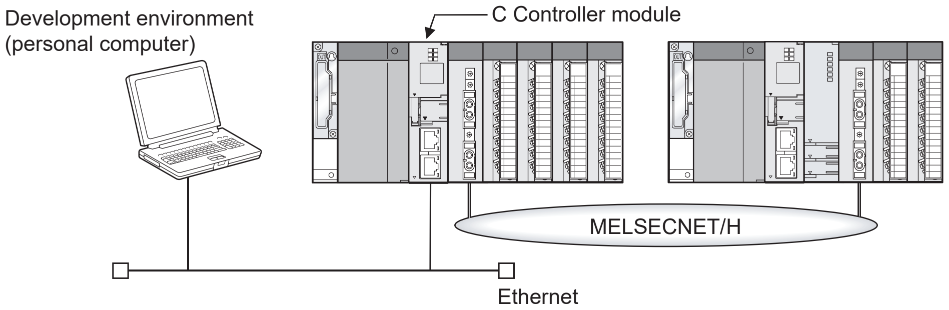
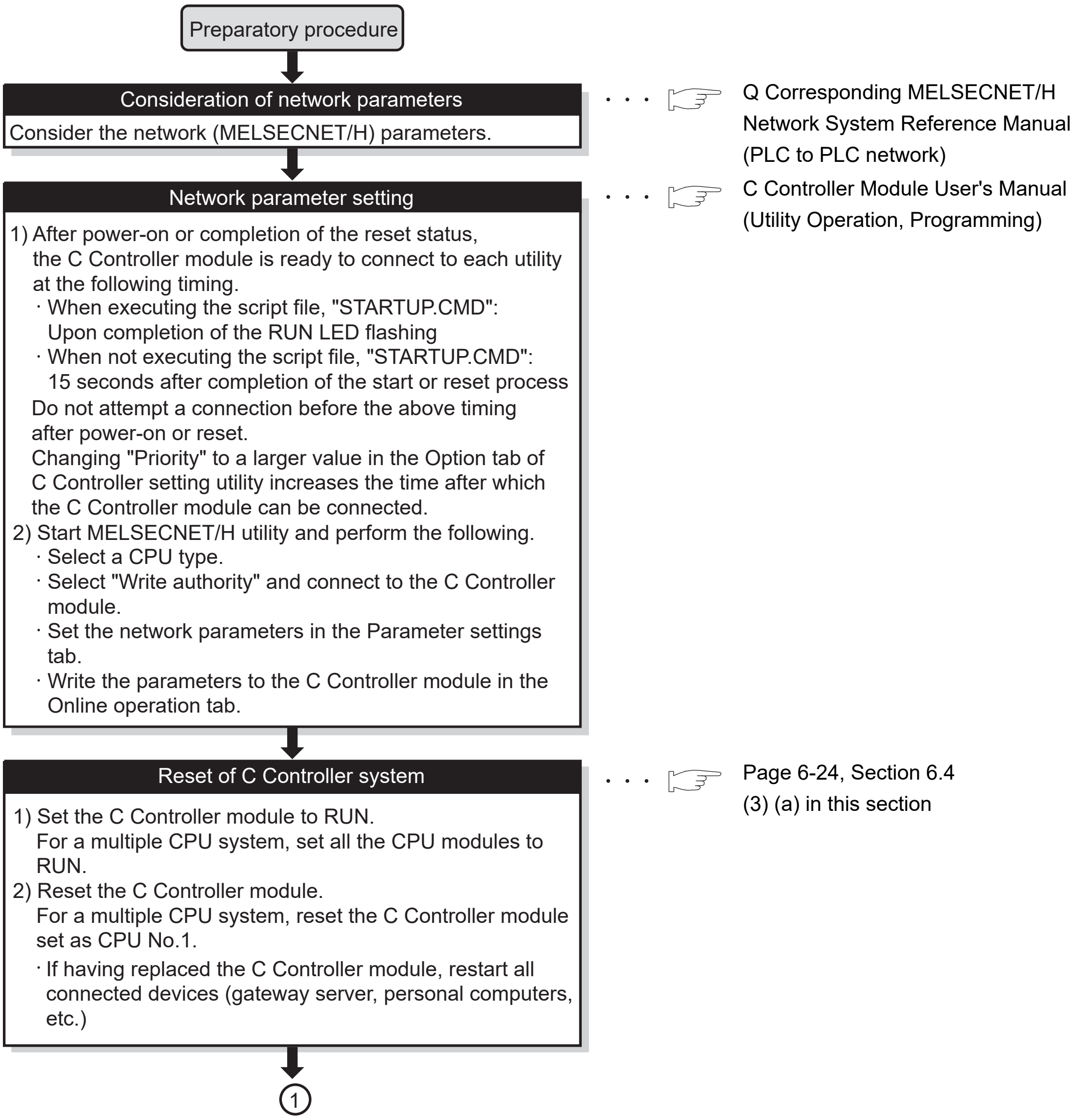


Figure 6.11 Configuration at MELSECNET/H startup



Consideration of network parameters

Consider the network (MELSECNET/H) parameters.

. . . ☞ Q Corresponding MELSECNET/H Network System Reference Manual (PLC to PLC network)

Network parameter setting

1) After power-on or completion of the reset status, the C Controller module is ready to connect to each utility at the following timing.
   - When executing the script file, "STARTUP.CMD": Upon completion of the RUN LED flashing
   - When not executing the script file, "STARTUP.CMD": 15 seconds after completion of the start or reset process

   Do not attempt a connection before the above timing after power-on or reset.
   Changing "Priority" to a larger value in the Option tab of C Controller setting utility increases the time after which the C Controller module can be connected.
2) Start MELSECNET/H utility and perform the following.
   - Select a CPU type.
   - Select "Write authority" and connect to the C Controller module.
   - Set the network parameters in the Parameter settings tab.
   - Write the parameters to the C Controller module in the Online operation tab.

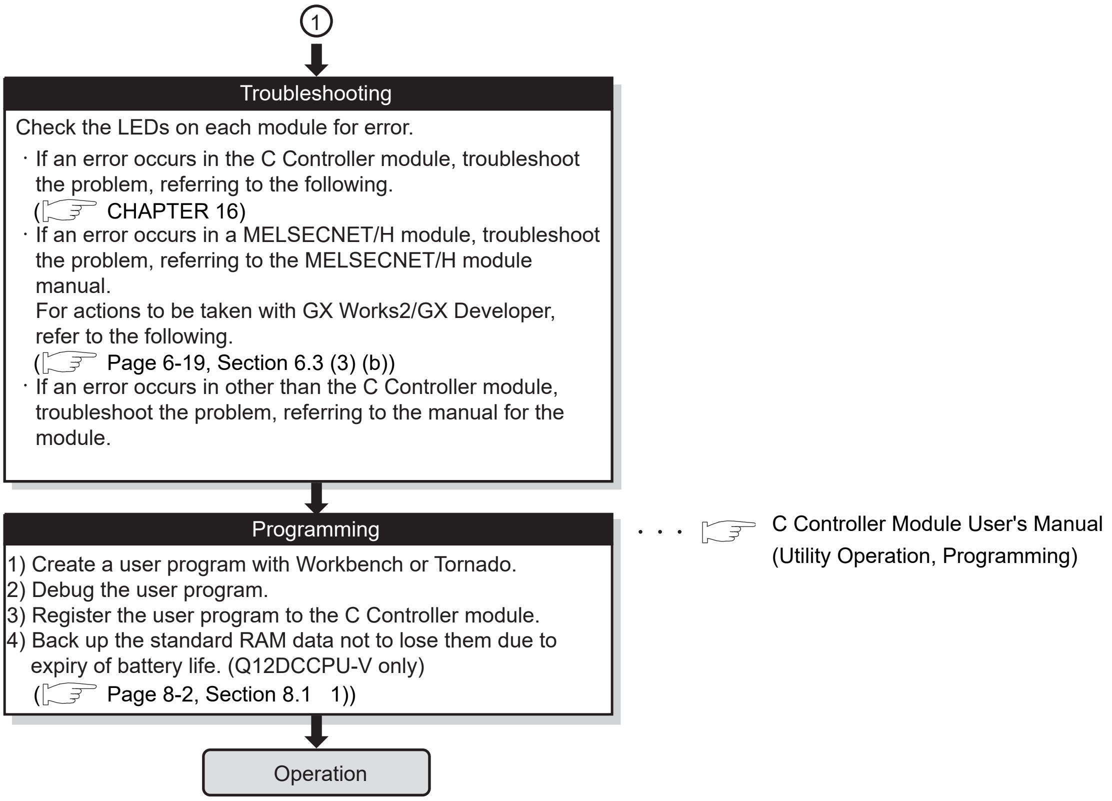
. . . ☞ C Controller Module User's Manual (Utility Operation, Programming)

Reset of C Controller system

1) Set the C Controller module to RUN.
   For a multiple CPU system, set all the CPU modules to RUN.
2) Reset the C Controller module.
   For a multiple CPU system, reset the C Controller module set as CPU No.1.
   - If having replaced the C Controller module, restart all connected devices (gateway server, personal computers, etc.)

. . . ☞ Page 6-24, Section 6.4 (3) (a) in this section

Figure 6.12 Flowchart for MELSECNET/H startup



Check the LEDs on each module for error.

- If an error occurs in the C Controller module, troubleshoot the problem, referring to the following. (☞ CHAPTER 16)
- If an error occurs in a MELSECNET/H module, troubleshoot the problem, referring to the MELSECNET/H module manual. For actions to be taken with GX Works2/GX Developer, refer to the following. (☞ Page 6-19, Section 6.3 (3) (b))
- If an error occurs in other than the C Controller module, troubleshoot the problem, referring to the manual for the module.

1) Create a user program with Workbench or Tornado.
2) Debug the user program.
3) Register the user program to the C Controller module.
4) Back up the standard RAM data not to lose them due to expiry of battery life. (Q12DCCPU-V only) (☞ Page 8-2, Section 8.1 1))

· · · ☞ C Controller Module User's Manual (Utility Operation, Programming)

Figure 6.12 Flowchart for MELSECNET/H startup (continued)

(a) Precautions

When having replaced a C Controller module, restart all connected devices (gateway server, personal computers, etc.)
(If any of these devices retains an Ethernet address of its communication target, continued communication may be disabled after the replacement because the Ethernet address is changed.)
Likewise, restart a C Controller module when a target device has been replaced.

(b) GX Works2/GX Developer and SW3PVC-CCPU

In troubleshooting according to the MELSECNET/H module manual, actions to be taken with GX Works2/GX Developer are as follows.

Table 6.3 GX Works2/GX Developer and SW3PVC-CCPU

GX Works2/GX Developer		SW3PVC-CCPU
PLC diagnostics		Use the System information screen of C Controller setting utility.
Sequence program check		Check the user program for error.
Network parameter		Use the Parameter settings tab of MELSECNET/H utility.
Network diagnostics		—
	Host information	Use the Detailed module information screen of MELSECNET/H utility.
	Other station information	Use the Other station monitoring tab of MELSECNET/H utility.
	Network Monitor Details	Use the Detailed module information screen of MELSECNET/H utility.
	Error history monitor	Use the Error history monitoring tab of MELSECNET/H utility.
	Loop test	Perform a test from GX Works2/GX Developer connected to a programmable controller CPU on another station.
	Setup confirmation test	Perform a test from GX Works2/GX Developer connected to a programmable controller CPU on another station.
	Station order check test	Perform a test from GX Works2/GX Developer connected to a programmable controller CPU on another station.
	Communication test	Perform a test from GX Works2/GX Developer connected to a programmable controller CPU on another station.
	Network test	Perform a test from GX Works2/GX Developer connected to a programmable controller CPU on another station.
Network range assignment, common parameters, station inherent parameters		Use the Parameter settings tab of MELSECNET/H utility.
Module's Detailed Information screen		Use the Module information tab of MELSECNET/H utility for the latest error code. For error codes other than the latest one, use the Module monitoring tab of C Controller setting utility.
Turning on/off a link device by testing from GX Works2/GX Developer		Use Device monitoring utility.
Routing parameters		Use the Routing parameter settings screen of MELSECNET/H utility.
Test between station		Use the Parameter settings tab of MELSECNET/H utility.
Forward loop test/Reverse loop test		Use the Parameter settings tab of MELSECNET/H utility.

(4) Starting up the CC-Link IE controller network

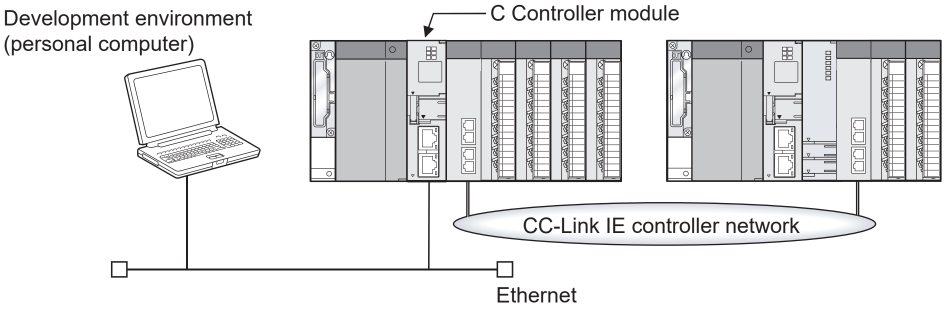
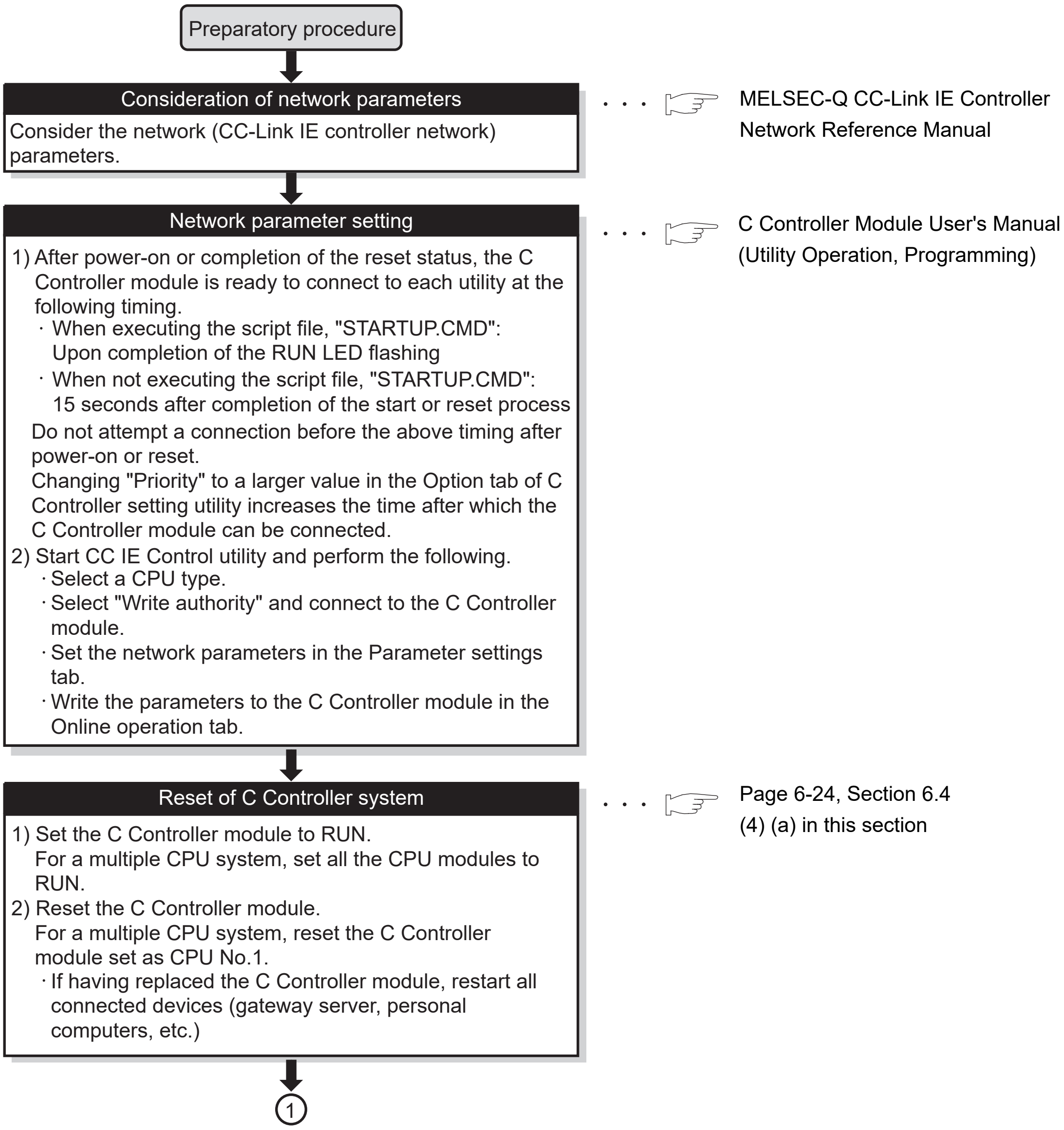


Figure 6.13 Configuration at CC-Link IE controller network startup



Preparatory procedure

Consideration of network parameters

Consider the network (CC-Link IE controller network) parameters.

. . . ☞ MELSEC-Q CC-Link IE Controller Network Reference Manual

Network parameter setting

1) After power-on or completion of the reset status, the C Controller module is ready to connect to each utility at the following timing.
   · When executing the script file, "STARTUP.CMD": Upon completion of the RUN LED flashing
   · When not executing the script file, "STARTUP.CMD": 15 seconds after completion of the start or reset process

   Do not attempt a connection before the above timing after power-on or reset.
   Changing "Priority" to a larger value in the Option tab of C Controller setting utility increases the time after which the C Controller module can be connected.
2) Start CC IE Control utility and perform the following.
   · Select a CPU type.
   · Select "Write authority" and connect to the C Controller module.
   · Set the network parameters in the Parameter settings tab.
   · Write the parameters to the C Controller module in the Online operation tab.

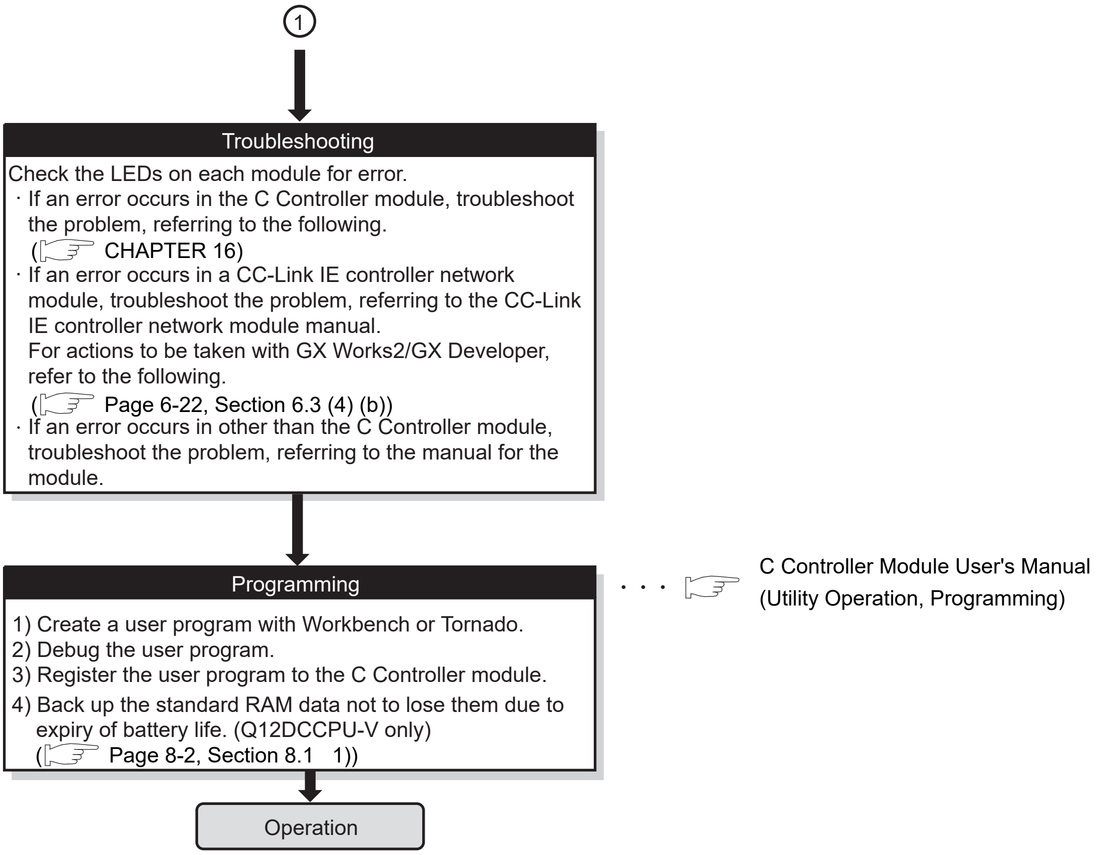
. . . ☞ C Controller Module User's Manual (Utility Operation, Programming)

Reset of C Controller system

1) Set the C Controller module to RUN.
   For a multiple CPU system, set all the CPU modules to RUN.
2) Reset the C Controller module.
   For a multiple CPU system, reset the C Controller module set as CPU No.1.
   · If having replaced the C Controller module, restart all connected devices (gateway server, personal computers, etc.)

. . . ☞ Page 6-24, Section 6.4 (4) (a) in this section

Figure 6.14 Flowchart for CC-Link IE controller network startup



Troubleshooting

Check the LEDs on each module for error.

- If an error occurs in the C Controller module, troubleshoot the problem, referring to the following. (☞ CHAPTER 16)
- If an error occurs in a CC-Link IE controller network module, troubleshoot the problem, referring to the CC-Link IE controller network module manual. For actions to be taken with GX Works2/GX Developer, refer to the following. (☞ Page 6-22, Section 6.3 (4) (b))
- If an error occurs in other than the C Controller module, troubleshoot the problem, referring to the manual for the module.

Programming

1) Create a user program with Workbench or Tornado.
2) Debug the user program.
3) Register the user program to the C Controller module.
4) Back up the standard RAM data not to lose them due to expiry of battery life. (Q12DCCPU-V only) (☞ Page 8-2, Section 8.1 1))

· · · ☞ C Controller Module User's Manual (Utility Operation, Programming)

Operation

Figure 6.14 Flowchart for CC-Link IE controller network startup (continued)

(a) Precautions

When having replaced a C Controller module, restart all connected devices (gateway server, personal computers, etc.)

(If any of these devices retains an Ethernet address of its communication target, continued communication may be disabled after the replacement because the Ethernet address is changed.)

Likewise, restart a C Controller module when a target device has been replaced.

(b) GX Works2/GX Developer and SW3PVC-CCPU
When performing troubleshooting according to the CC-Link IE controller network module manual, actions to be taken with GX Works2/GX Developer are as follows.

Table 6.4 GX Works2/GX Developer and SW3PVC-CCPU

GX Works2/GX Developer	SW3PVC-CCPU
CC IE Control Network Diagnostics	Use the CC-Link IE Controller Network diagnostics result screen of CC IE Control utility.
PLC diagnostics	Use the System information screen of C Controller setting utility.
Error code check	Check the Module information tab or System information screen of C Controller setting utility for error.
Sequence program check	Check the user program for error.
Device test	Use Device monitoring utility.
Device batch monitor	Use Device monitoring utility.
Network parameter	Use the Parameter settings tab of CC IE Control utility.
Network range assignment	Use the Network range assignment screen of CC IE Control utility.
H/W test	Use the Parameter settings screen of CC IE Control utility.
Refresh parameters	Use the Refresh parameter settings screen of CC IE Control utility.
Link start/stop	Use the Link start/stop screen of CC IE Control utility.
Routing parameters	Use the Routing parameter settings screen of CC IE Control utility.
Circuit test	Use the Parameter settings screen of CC IE Control utility.
Test between station	Use the Parameter settings screen of CC IE Control utility.
Communication test	Use the Communication test screen of CC IE Control utility.
Self-loopback test	Use the Parameter settings screen of CC IE Control utility.

(5) Connecting GOT by microcomputer connection

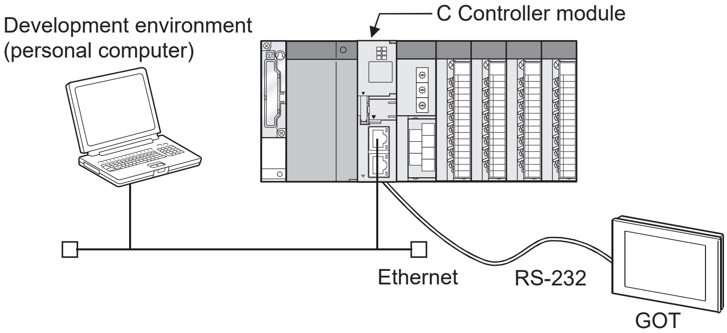
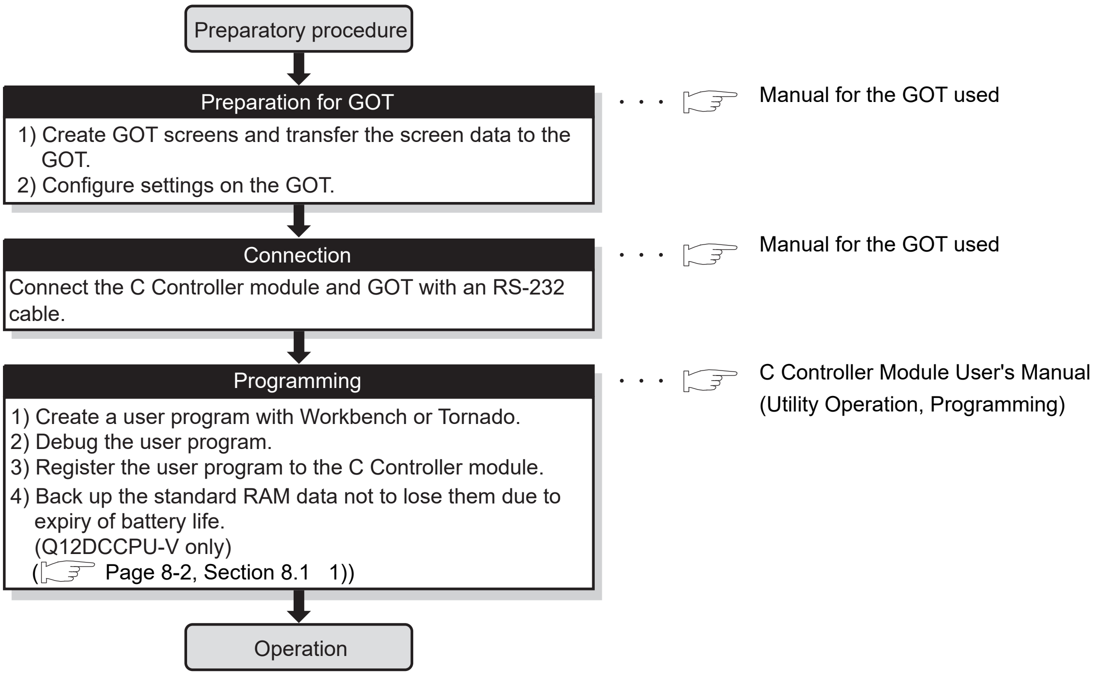


Figure 6.15 Microcomputer connection with GOT



Preparatory procedure

Preparation for GOT . . . ☞ Manual for the GOT used

1) Create GOT screens and transfer the screen data to the GOT.
2) Configure settings on the GOT.

Connection . . . ☞ Manual for the GOT used

Connect the C Controller module and GOT with an RS-232 cable.

Programming . . . ☞ C Controller Module User's Manual (Utility Operation, Programming)

1) Create a user program with Workbench or Tornado.
2) Debug the user program.
3) Register the user program to the C Controller module.
4) Back up the standard RAM data not to lose them due to expiry of battery life.
(Q12DCCPU-V only)
(☞ Page 8-2, Section 8.1 1))

Operation

Figure 6.16 Flowchart for connecting GOT by microcomputer connection

☒POINT

For commands that can be used for data transfer to the GOT in microcomputer connection, refer to the manual for the GOT used.

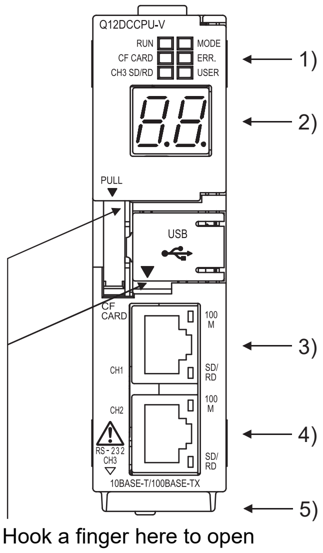
6.4 Part Names and Functions

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	○

6.4.1 Part names and functions of the Q12DCCPU-V

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	×	×

The part names of the Q12DCCPU-V are shown by numbers given in the external views.



Hook a finger here to open the front cover.

Figure 6.17 Front face

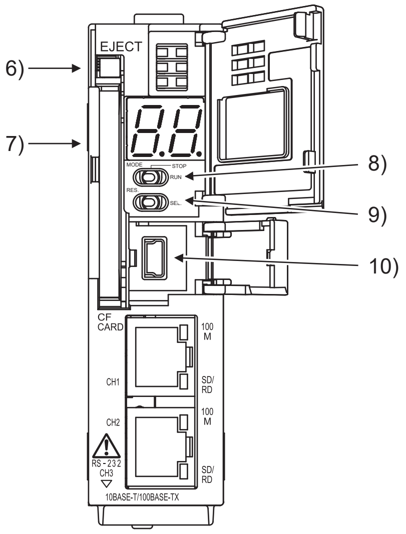


Figure 6.18 Front face with the cover open

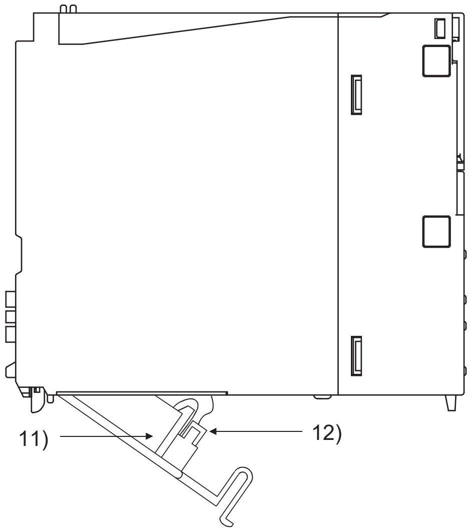


Figure 6.19 Side face

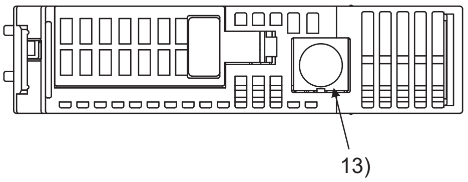


Figure 6.20 Bottom face

Table 6.5 Part names

	Name	Description
1)	Indicator LEDs	For the indicator LEDs, refer to the following. ☞ (1) in this section
2)	7-segment LED	For the 7-segment LED, refer to the following. ☞ (2) in this section
3) 4)	10BASE-T/100BASE-TX interface connector (RJ45)	Used to connect a C Controller module to 10BASE-T/100BASE-TX. (The C Controller module determines 10BASE-T or 100BASE-TX depending on the target device.)
5)	Serial number plate	Indicates the serial number of the C Controller module.
6)	EJECT button	Used to eject a CompactFlash card from a C Controller module.
7)	CompactFlash card installation slot	Slot into which a CompactFlash card is installed. (A dummy CompactFlash card is factory-installed.*1)
8)	RUN/STOP/MODE switch*2	For the RUN/STOP/MODE switch, refer to the following. ☞ (3) in this section
9)	RESET/SELECT switch*2	For the RESET/SELECT switch, refer to the following. ☞ (4) in this section
10)	USB interface connector	Used to connect the C Controller module to a USB (connector type: miniB).
11)	Battery	Used to back up data in the standard RAM, data in the battery-backed-up RAM, and clock data.
12)	Battery connector pin	Used to connect a battery lead wire. (The lead wire is disconnected from the connector before shipping to prevent battery drain.)
13)	RS-232 interface connector	Used to connect a C Controller module to an RS-232 cable.

* 1 Keep the dummy CompactFlash card in a safe place after replacing it so that it can be used again.

* 2 Operate the RUN/STOP/MODE and RESET/SELECT switches by hand.
Using a tool such as a screwdriver can cause a failure.

(1) Indicator LED specifications

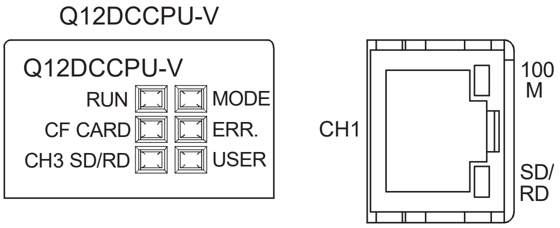


Figure 6.21 Indicator LEDs

(a) In normal operation mode

The operation mode is changed by the switches. (☞ (3) in this section, (4))

Table 6.6 Indicator LEDs and their states in normal operation mode

Indicator LED[*1]	LED status	Description
RUN	On	The C Controller module is in the RUN status. (Output (Y) from a user program and writing to the buffer memory are enabled.)
	Off	The C Controller module is in the STOP/PAUSE status. (Output (Y) from a user program and writing to the buffer memory are disabled.)
	Flashing	The script file "STARTUP.CMD" is in execution.
MODE	On	Normal operation mode (VxWorks is running.) The MODE LED turns ON green for basic mode, turns ON orange for extended mode.
	Off	A hardware failure occurred or the C Controller module is in reset process.
ERR.	On	A continuation error occurred.
	Off	Normal
	Flashing	A stop error occurred.[*2]
USER	On	This LED can be controlled with the QBF_ControlLED function in user programs. (☞C Controller Module User's Manual (Utility Operation, Programming))
	Off	
	Flashing	
CF CARD	On	A CompactFlash card is mounted. (☞ Page 6-60, Section 6.8.1)
	Off	A CompactFlash card is not installed or unmounted. (☞ Page 6-60, Section 6.8.1)
	Flashing	A CompactFlash card is unmounted by the RESET/SELECT switch. (☞ Page 6-67, Section 6.8.3)
CH1 100M	On	Connected at 100Mbps.
	Off	Connected at 10Mbps.
CH1 SD/RD	On	Physical Ethernet connection has been detected.
	Flashing	Data are being sent or received through the 10BASE-T/100BASE-TX interface.
	Off	Physical Ethernet connection has not been detected.
CH2 100M	Same as CH1	
CH2 SD/RD		
CH3 SD/RD	On	Data are being sent or received through CH3 (RS-232 interface).
	Off	No data communication

[*1] All of the LEDs turn off while the module is reset.

[*2] When a system watchdog timer error occurs, the RUN and MODE LEDs turn off and the ERR. LED flashes.

(b) In hardware self-diagnostic operation mode

The operation mode is changed by switches. (☞ (3), (4) in this section)

Table 6.7 Indicator LEDs and their states in hardware self-diagnostic operation mode

Indicator LED	Name	LED status	Description
MODE	MODE	On	Hardware self-diagnostic mode (The MODE LED turns ON orange)
		Off	A hardware failure occurred or the C Controller module is in reset process.
ERR.	ERR.	Off	When all LEDs except the MODE LED turn off, it indicates that the diagnostics and settings have not been executed yet, or have completed normally.
		Flashing	An error occurred during mode setting with the RESET/SELECT switch. The error location can be identified by a value displayed on the 7-segment LED. (☞ Page 16-81, Section 16.5.1 (5))

Remark

For the hardware self-diagnostic function, refer to the following.

☞ Page 16-75, Section 16.5.1

(2) 7-segment LED specifications

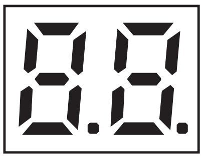
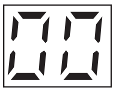
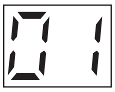
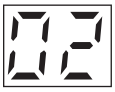


Figure 6.22 7-segment LED

Table 6.8 7-segment LED specifications

Name	LED status	Description
7-segment LED	7-segment LED display	This LED display can be controlled with the QBF_Control7SegLED function in user programs.*1 It indicates the execution mode of the hardware self-diagnostics test or initialization setting when switch operation is performed.
	 Flashing	A built-in ROM data error detected. • An error was detected in the built-in ROM data during startup of the C Controller module. • The built-in ROM data are corrupt. • The C Controller module cannot be started.
	 Flashing*2  Flashing*3	A standard RAM data or a battery-backed-up RAM data error detected. • An error was detected in the standard RAM data or a battery-backed-up RAM data during startup of the C Controller module. • The standard RAM data or a battery-backed-up RAM data are corrupt. • The C Controller module cannot be started.

* 1 The LED cannot be controlled when:
- A system watchdog timer error has occurred.
- A user watchdog timer error has occurred.
- A built-in ROM data error has been detected.
- A standard RAM data error has been detected.

* 2 The LED is flashing when a standard RAM data error detected.

* 3 The LED is flashing when a battery-backed-up RAM data error detected.

(3) RUN/STOP/MODE switch operation

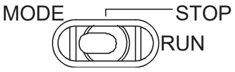


Figure 6.23 The RUN/STOP/MODE switch

Table 6.9 RUN/STOP/MODE switch operation

Position	Operation	
RUN	In normal operation mode	The C Controller module is operating. (Output (Y) from a user program and writing to the buffer memory are enabled.)
	In hardware self-diagnostic operation mode	For the hardware self-diagnostic function, refer to the following. ☞ Page 16-75, Section 16.5.1
STOP	The operation of the C Controller module is stopped.[*1] Switching from RUN to STOP turns off all outputs (Y). (Output (Y) from a user program and writing to the buffer memory are disabled.)	
MODE	Used for the hardware self-diagnostic function. (☞ Page 16-75, Section 16.5.1)	

[* 1] Operations such as forced output (Y) and buffer memory writing can be executed from the <<Module monitoring>> tab of C Controller setting utility.

(4) RESET/SELECT switch operation

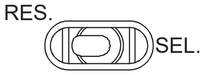


Figure 6.24 The RESET/SELECT switch

Table 6.10 RESET/SELECT switch operation

Position	Operation	
RES.	When the RESET position is held	All LEDs turn off and the hardware is reset.[*2] [*3]
	When the switch returns from RESET to central position	The module exits the reset status. • The module starts in either of the following operation modes depending on the RUN/STOP/MODE switch position. • In RUN or STOP setting, the module restarts from the OS and system software in the normal operation mode. (The MODE LED turns ON green for basic mode, turns ON orange for extended mode.) • In MODE setting, the module starts in the hardware self-diagnostic operation mode. (The MODE LED turns on orange.)
SEL.	In normal operation mode	If the switch is held in the SELECT position, the CompactFlash card is unmounted.[*4] (☞ Page 6-67, Section 6.8.3)
	In hardware self-diagnostic operation mode	Used for the hardware self-diagnostic function. (☞ Page 16-75, Section 16.5.1)

* 2 In a multiple CPU system configuration, reset CPU No.1.
(☞ Page 12-21, Section 12.6)

* 3 Data corruption or file system failure may occur if the C Controller system is reset during writing of a user file in the standard RAM or CompactFlash card.
If a user file in the standard RAM or CompactFlash card is being written, perform the operation shown in the following before resetting the module.
(☞ Page 8-9, Section 8.2.2 (2))

* 4 When the SELECT position is held in either of the following conditions, the CompactFlash card is not unmounted. (The CF CARD LED does not flash.)
- No CompactFlash card is installed.
- The CompactFlash card has already been unmounted.

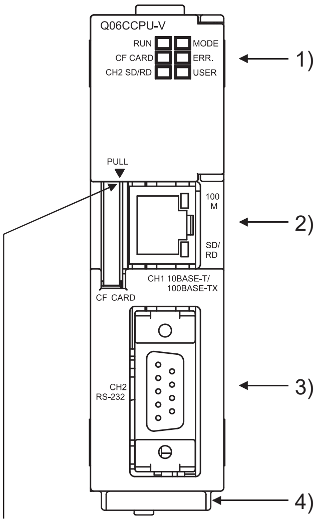
6.4.2 Part names and functions of the Q06CCPU-V(-B)

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
×	○	○

The part names of the Q06CCPU-V and Q06CCPU-V-B are shown by numbers given in external views of the Q06CCPU-V.

The external views of the Q06CCPU-V-B are the same as those of the Q06CCPU-V except:

- Model indication
- Indicator LEDs
- The EJECT button
- CompactFlash card slot



Hook a finger here to open the front cover.

Figure 6.25 Front face

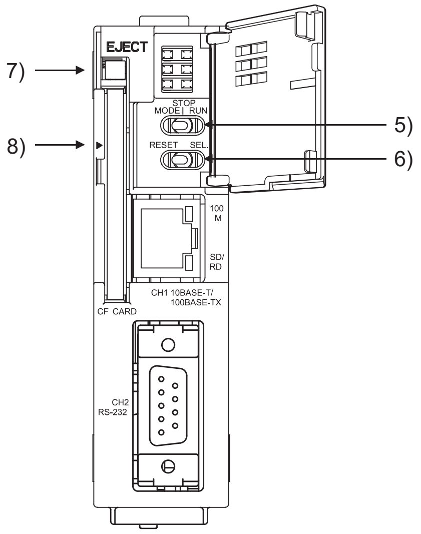


Figure 6.26 Front face with the cover open

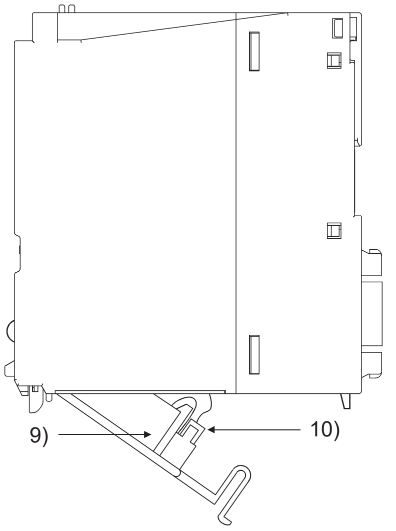


Figure 6.27 Side face

Table 6.11 Part names

	Name	Description
1)	Indicator LEDs	For the indicator LEDs, refer to the following. ☞ (1) in this section
2)	10BASE-T/100BASE-TX interface connector (RJ45)	Used to connect a C Controller module to 10BASE-T/100BASE-TX. (The C Controller module determines 10BASE-T or 100BASE-TX depending on the target device.)
3)	RS-232 interface connector	Used to connect a C Controller module to an RS-232 cable.
4)	Serial number plate	Indicates the serial number of the C Controller module.
5)	RUN/STOP/MODE switch*2	For the RUN/STOP/MODE switch, refer to the following. ☞ (2) in this section
6)	RESET/SELECT switch*2	For the RESET/SELECT switch, refer to the following. ☞ (3) in this section
7)	EJECT button (Q06CCPU-V only)	Used to eject a CompactFlash card from a C Controller module.
8)	CompactFlash card slot (Q06CCPU-V only)	Slot into which a CompactFlash card is installed. (A dummy CompactFlash card is factory-installed.*1)
9)	Battery	Used to back up data in the standard ROM, data in the battery-backed-up RAM, and clock data.
10)	Battery connector pin	Used to connect a battery lead wire. (The lead wire is disconnected from the connector before shipping to prevent battery drain.)

* 1 Keep the dummy CompactFlash card in a safe place after replacing it so that it can be used again.

* 2 Operate the RUN/STOP/MODE and RESET/SELECT switches by hand.
Using a tool such as a screwdriver can cause a failure.

(1) Indicator LED specifications

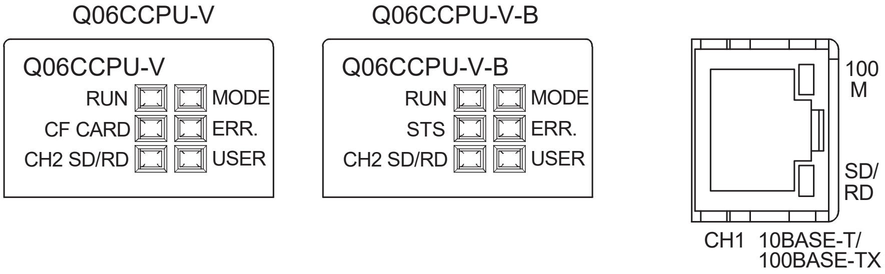


Figure 6.28 Indicator LEDs

(a) In normal operation mode (when the MODE LED is on (green))

The operation mode is changed by switches. (☞ (2), (3) in this section)

Table 6.12 Indicator LEDs and their states in normal operation mode

Indicator LED[*1]	LED status	Description
RUN	On	The C Controller module is in the RUN status. (Output (Y) from a user program and writing to the buffer memory are enabled.)
	Off	The C Controller module is in the STOP/PAUSE status. (Output (Y) from a user program and writing to the buffer memory are disabled.)
	Flashing	The script file "STARTUP.CMD" is in execution.
MODE	On	Normal operation mode (VxWorks is running.)
	Off	A hardware failure occurred or the C Controller module is in reset process.
	Flashing	Shutdown is completed. (☞ Page 6-58, Section 6.7.5 (1))
ERR.	On	A continuation error occurred.
	Off	Normal
	Flashing	A stop error occurred.[*2]
USER	On	This LED can be controlled with the QBF_ControlLED function in user programs. (☞ C Controller Module User's Manual (Utility Operation, Programming))
	Off	
	Flashing	
CF CARD	On	A CompactFlash card is mounted. (☞ Page 6-64, Section 6.8.2)
	Off	A CompactFlash card is not installed or unmounted. (☞ Page 6-64, Section 6.8.2)
	Flashing	A CompactFlash card is unmounted by the RESET/SELECT switch. (☞ Page 6-67, Section 6.8.3)
STS	Off	Always off
100M	On	Connected at 100Mbps.
	Off	Connected at 10Mbps.
SD/RD	On	Data are being sent or received through CH1 (10BASE-T/100BASE-TX interface).
	Off	No data communication
CH2 SD/RD	On	Data are being sent or received through CH2 (RS-232 interface).
	Off	No data communication

[*1] All of the LEDs turn off while the module is reset.

[*2] When a system watchdog timer error occurs, the RUN and MODE LEDs turn off and the ERR. LED flashes.

(b) In hardware self-diagnostic operation mode (when the MODE LED is on (orange))
The operation mode is changed by switches. (☞ (2), (3) in this section)

Table 6.13 Indicator LEDs and their states in hardware self-diagnostic operation mode

Indicator LED	Name	LED status	Description
MODE	MODE	On	Hardware self-diagnostic mode
		Off	A hardware failure occurred or the C Controller module is in reset process.
ERR.	ERR.	Off	When all LEDs except the MODE LED turn off, it indicates that the diagnostics and settings have not been executed yet, or have completed normally.
		Flashing	An error occurred during mode setting (Mode 1, Mode 2, Mode 3, or Default IP setting mode) with the RESET/SELECT switch. The error location can be identified by the status of the following LEDs. "RUN", "CF CARD", "STS", "CH2 SD/RD", and "USER"
RUN	Mode 1/ Error location indication	On	ERR. LED off: Mode 1 is selected. ERR. LED flashing: Indicates the location of the error that occurred during execution of Mode 1 to Mode 3.
		Flashing	Mode 1 is in execution.
CF CARD	Mode 2/ Error location indication*1	On	ERR. LED off: Mode 2 is selected. ERR. LED flashing: Indicates the location of the error that occurred during execution of Mode 1 to Mode 3.
STS		Flashing	Mode 2 is in execution.
CH2 SD/RD	Mode 3/ Error location indication	On	ERR. LED off: Mode 3 is selected. ERR. LED flashing: Indicates the location of the error that occurred during execution of Mode 1 to Mode 3.
		Flashing	Mode 3 is in execution.
USER	Default setting mode/Error location indication*1	On	ERR. LED off: Default IP setting mode is selected. ERR. LED flashing: An error occurred during execution of the default setting mode.
		Flashing	Default IP setting mode is in execution.

* 1 When the CF CARD LED and USER LED, or the STS LED and USER LED turn on simultaneously, it indicates the standard ROM initialization setting.
For the standard ROM initialization setting, refer to the following.
☞ Page 16-84, Section 16.5.2

Remark

For the hardware self-diagnostic function, refer to the following.
☞ Page 16-84, Section 16.5.2

(2) RUN/STOP/MODE switch operation

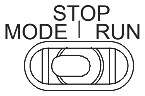


Figure 6.29 The RUN/STOP/MODE switch

Table 6.14 RUN/STOP/MODE switch operation

Position	Operation	
RUN	In normal operation mode	The C Controller module is operating. (Output (Y) from a user program and writing to the buffer memory are enabled.)
	In hardware self-diagnostic operation mode	For the hardware self-diagnostic function, refer to the following. ☞ Page 16-84, Section 16.5.2
STOP	The operation of the C Controller module is stopped.[*1] Switching from RUN to STOP turns off all outputs (Y). (Output (Y) from a user program and writing to the buffer memory are disabled.)	
MODE	Used for the hardware self-diagnostic function. (☞ Page 16-84, Section 16.5.2)	

[*1] Operations such as forced output (Y) and buffer memory writing can be executed from the <<Module monitoring>> tab of C Controller setting utility.

(3) RESET/SELECT switch operation

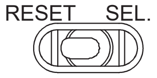


Figure 6.30 The RESET/SELECT switch

Table 6.15 RESET/SELECT switch operation

Position	Operation	
RESET	When the RESET position is held	All LEDs turn off and the hardware is reset.[*2 *3]
	When the switch returns from RESET to central position	The module exits the reset status. • The module starts in either of the following operation modes depending on the RUN/STOP/MODE switch position. • In RUN or STOP setting, the module restarts from the OS and system software in the normal operation mode. (The MODE LED turns on green.) • In MODE setting, the module starts in the hardware self-diagnostic operation mode. (The MODE LED turns on orange.)
SEL.	In normal operation mode	If the switch is held in the SELECT position on the Q06CCPU-V, the CompactFlash card is unmounted.[*4] (☞ Page 6-67, Section 6.8.3) No processing is performed for the Q06CCPU-V-B.
	In hardware self-diagnostic operation mode	Used for the hardware self-diagnostic function. (☞ Page 16-84, Section 16.5.2)

* 2 In a multiple CPU system configuration, reset CPU No.1.
(☞ Page 12-21, Section 12.6)

* 3 Data corruption or file system failure may occur if the C Controller system is reset during writing of a user file in the standard ROM or CompactFlash card
If a user file in the standard ROM or CompactFlash card is being written, perform the operation shown in the following before resetting the module.
(☞ Page 8-9, Section 8.2.2 (2))

* 4 When the SELECT position is held in either of the following conditions, the CompactFlash card is not unmounted. (The CF CARD LED does not flash.)
- No CompactFlash card is installed.
- The CompactFlash card has already been unmounted.

6.5 Cable Connection

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	○

This section explains how to connect cables to the C Controller module.

To fully utilize the C Controller module functionality and achieve a reliable system, wiring resistant to noise is required.

- Ground the shield of the RS-232 cable to a single point.
- When connecting an external device with an RS-232 cable, use a connector shell introduced in the following for the C Controller module side.
  ☞ Page 3-4, Section 3.3
- Take sufficient safety measures when installing a 10BASE-T/100BASE-TX network system.
- Consult qualified service personnel for connections of cable terminals and installation of a trunk line cable.
- Use a 10BASE-T/100BASE-TX cable that meets the standards shown in the following.
  ☞ Page 2-25, Section 2.4
- For the cable bend radius near the connector, refer to the following.
  ☞ Page App-6, Appendix 2
- Confirm the specifications of the external device before connecting it.
- Do not short the FG and SG signals of the RS-232 cable.
  When the FG and SG signals are connected inside of the external device, do not connect the FG signal to the C Controller module.

6.5.1 Compliance of the C Controller module with the EMC Directive

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	○

To make the C Controller module compliant with the EMC Directive, the Ethernet cables, RS-232 cables, and USB cables must meet the noise immunity standards.

(a) Ethernet cable
Use a shielded twisted pair cable for connection to the 10BASE-T/100BASE-TX interface connector.

(b) RS-232 cable
For the RS-232 cable running outside from the control panel, always ground the shield section of the shielded cable.

(c) USB cable (Q12DCCPU-V only)
Use a USB cable of 3m (9.84 feet) or less.

Remark

For the EMC Directive, refer to the following manual.
☞ QCPU User's Manual (Hardware Design, Maintenance and Inspection)

6.6 Network Settings for 1:1 Connection

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	○

When connecting a development environment (personal computer) to a C Controller module, connect them on one-to-one (1:1) basis by using an Ethernet cable.*1

* 1 For the Q12DCCPU-V, use an Ethernet cable (straight cable or crossing cable).
For the Q06CCPU-V(-B), use an Ethernet cable (crossing cable).

After connection with an Ethernet cable, configure the settings of the C Controller module from C Controller setting utility on the development environment (personal computer).
This section describes the network settings on the personal computer for 1:1 connection between the C Controller module (CH1) and the development environment (personal computer).

(1) System configuration for 1:1 connection

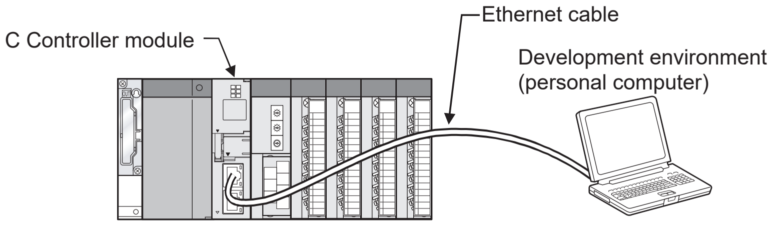


Figure 6.31 System configuration for 1:1 connection

(2) Network settings for the development environment (personal computer)

(a) Network settings for the development environment (personal computer)
Set the same network address as that of the C Controller module.

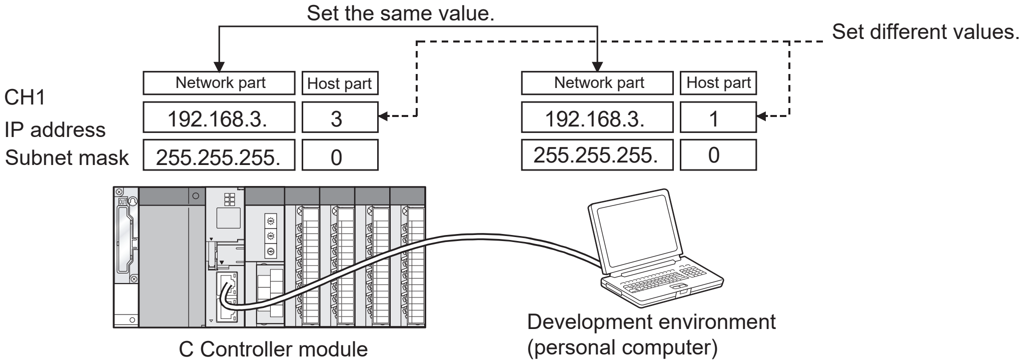


Figure 6.32 Network settings for the development environment (personal computer)

(b) How to configure network settings for the development environment (personal computer)
The settings can be configured on the "Internet Protocol (TCP/IP) Properties" screen.

1) Select [Network and Internet] → [Network and Sharing Center] → [Change adapter settings] in the control panel of Windows.
2) Select and right-click the network connection, and select [Properties] in the shortcut menu.
3) Select "Internet Protocol Version 4 (TCP/IPv4)", and click the [Properties] button in the property screen for network connection.

(c) Restart the development environment (personal computer) to enable the network settings.

(3) Accessing the C Controller module from the development environment (personal computer)

(a) Confirmation of power-on
Check that the C Controller module has been started up (powered on).

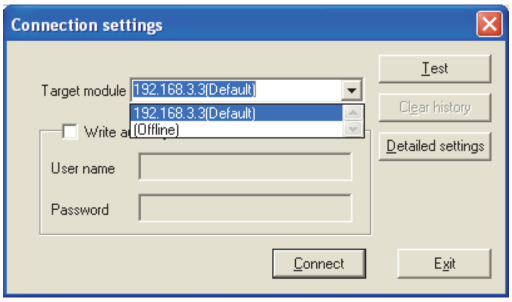
(b) The C Controller module is ready to connect to C Controller setting utility at the following timing.

- When executing the script file, "STARTUP.CMD":
  Upon completion of the RUN LED flashing
- When not executing the script file, "STARTUP.CMD":
  Fifteen (15) seconds after completion of start or reset

Do not attempt a connection before the above timing after power is turned on.
Changing "Priority" to a larger value in the Option tab of C Controller setting utility increases the time after which the C Controller module can be connected.

(c) Setting in C Controller setting utility
Start C Controller setting utility on the development environment (personal computer), and select "192.168.3.3(Default)" for Target module.

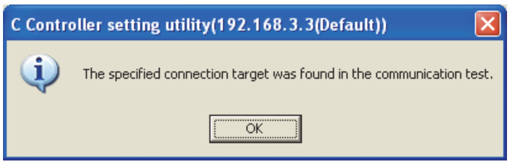
Click the Test button to execute a communication test.


Connection settings

Target module 192.168.3.3(Default)

Figure 6.33 Connection settings

When the following message appears, the C Controller module is accessible.


C Controller setting utility(192.168.3.3(Default))

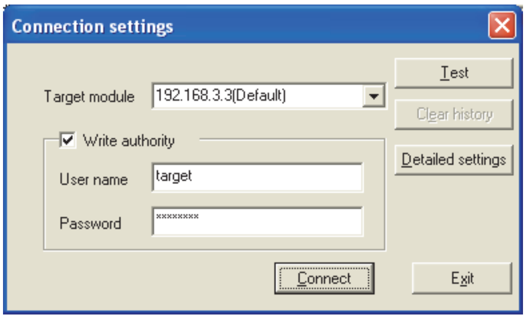
The specified connection target was found in the communication test.

Figure 6.34 Test result

(d) Accessing the C Controller module

Check the "Write authority" box and enter the login user information (User name and Password ☞ Page 6-89, Section 6.11.2).

Click the Connect button to access the C Controller module with the write authority enabled.



Connection settings

Target module: 192.168.3.3(Default)

☑ Write authority

User name: target

Password: ********

Figure 6.35 Connection settings screen

POINT

1. When the C Controller module is not accessible, issue a PING command from the development environment (personal computer) to the C Controller module to check the presence.
2. Disable Windows® Firewall when connecting the development environment (personal computer) to the C Controller module.

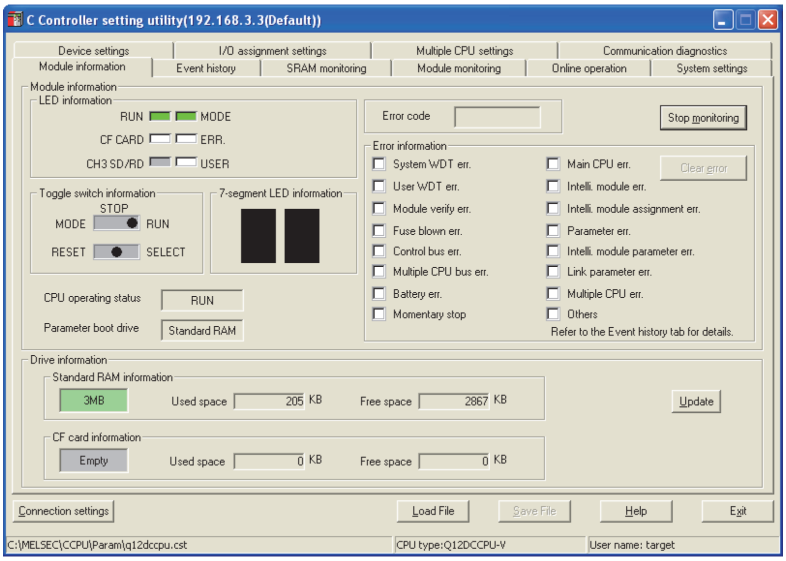
   For Windows® Firewall setting, refer to the following.

   ☞ C Controller Module User's Manual (Utility Operation, Programming)

(4) Monitoring the C Controller module status

Communications with the C Controller module are available from C Controller utility after the connection is complete.

In the tabs such as <<Module information >>, <<Event history>>, and <<Module monitoring>>, check that relevant information can be monitored.



C Controller setting utility(192.168.3.3(Default))

Module information

LED information: RUN 🟩, MODE 🟩, CF CARD □, ERR. □, CH3 SD/RD ▨, USER □

Toggle switch information: MODE STOP/RUN, RESET/SELECT

Error code:

Error information:

☐ System WDT err. ☐ Main CPU err.
☐ User WDT err. ☐ Intelli. module err.
☐ Module verify err. ☐ Intelli. module assignment err.
☐ Fuse blown err. ☐ Parameter err.
☐ Control bus err. ☐ Intelli. module parameter err.
☐ Multiple CPU bus err. ☐ Link parameter err.
☐ Battery err. ☐ Multiple CPU err.
☐ Momentary stop ☐ Others

Refer to the Event history tab for details.

CPU operating status: RUN

Parameter boot drive: Standard RAM

Drive information

Standard RAM information: 3MB, Used space 205 KB, Free space 2867 KB

CF card information: Empty, Used space 0 KB, Free space 0 KB

C:\MELSEC\CCPU\Param\q12dccpu.cst | CPU type:Q12DCCPU-V | User name: target

Figure 6.36 Module information tab (Q12DCCPU-V)

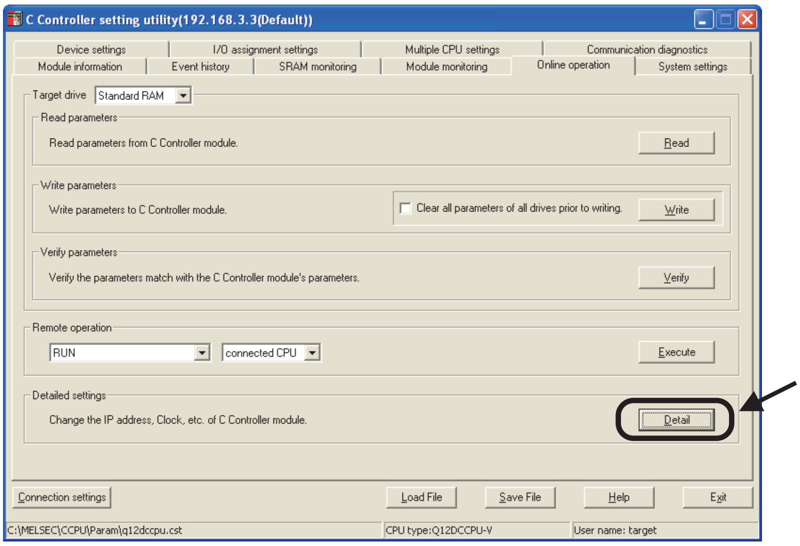
(5) Changing the IP address of the C Controller module

To change the default IP address of the C Controller module, perform the operations described below.

- Default IP address of CH1: 192.168.3.3
- Default IP address of CH2: No setting (Q12DCCPU-V only)

(a) Displaying the relevant tab

In the <<Online operation>> tab of C Controller setting utility, click the Detail button and select the <<IP address>> tab.


C Controller setting utility(192.168.3.3(Default))

Target drive: Standard RAM

Read parameters: Read parameters from C Controller module.

Write parameters: Write parameters to C Controller module. □ Clear all parameters of all drives prior to writing.

Verify parameters: Verify the parameters match with the C Controller module's parameters.

Remote operation: RUN, connected CPU

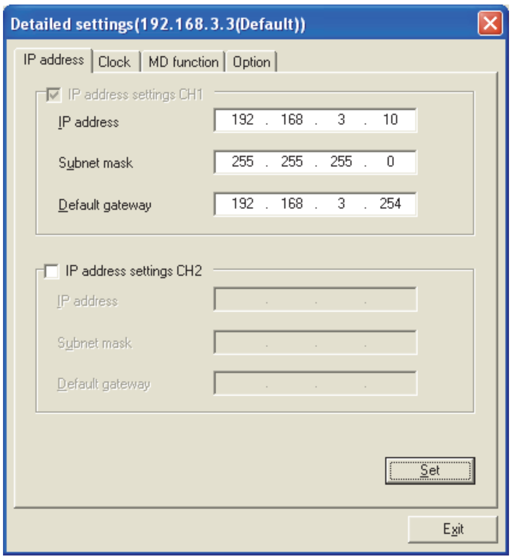
Detailed settings: Change the IP address, Clock, etc. of C Controller module.

C:\MELSEC\CCPU\Param\q12dccpu.cst | CPU type:Q12DCCPU-V | User name: target

Figure 6.37 <<Online operation>> tab

(b) Setting an IP address

Change the preset IP address (and Subnet mask and/or Default gateway as necessary) and click the Set button.


Detailed settings(192.168.3.3(Default))

☑ IP address settings CH1

IP address: 192 . 168 . 3 . 10

Subnet mask: 255 . 255 . 255 . 0

Default gateway: 192 . 168 . 3 . 254

□ IP address settings CH2

IP address: . . .

Subnet mask: . . .

Default gateway: . . .

Figure 6.38 IP address setting

After changing the data, reset the C Controller module to update the IP address settings.

(c) Network settings for the development environment (personal computer)
Change the network settings (IP address) of the development environment (personal computer) as needed, and shutdown the personal computer.

(d) Connection to LAN
Disconnect the Ethernet cable from the C Controller module and development environment (personal computer).
Then, connect the C Controller module and development environment (personal computer) separately to LAN with different Ethernet cables (straight cables).

(e) Access to the C Controller module
Restart the C Controller module and development environment (personal computer), and perform the operations described in (3) in this section.

(f) Confirmation of monitoring
Access the C Controller module from C Controller utility on the development environment (personal computer), and check that each status of the module can be monitored.

(6) Precautions

(a) Precautions for using two Ethernet channels (ports) (Q12DCCPU-V only)
When using two channels as Ethernet ports, two different network addresses must be set for CH1 and CH2 of the C Controller module.
In the above case, a response to the message received through each channel is sent as follows.

- Messages (including response packets such as ping) to the device having the same network address as CH1 are sent from CH1.
- Messages (including response packets such as ping) to the device having the same network address as CH2 are sent from CH2.

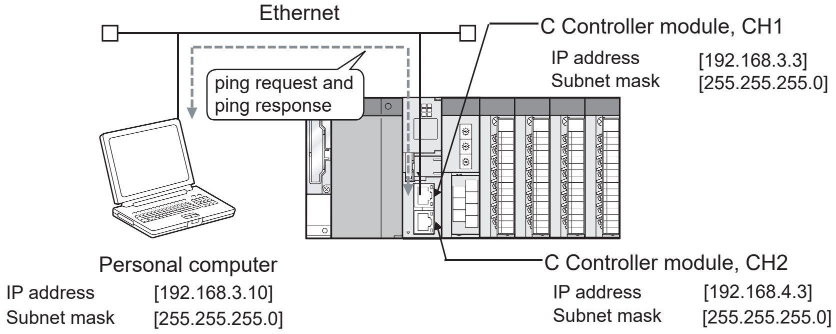
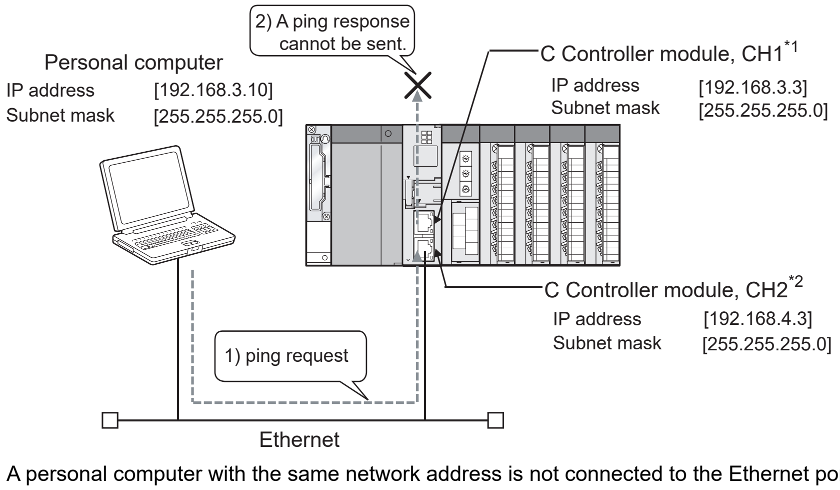

Ethernet
ping request and ping response
C Controller module, CH1
IP address [192.168.3.3]
Subnet mask [255.255.255.0]
Personal computer
IP address [192.168.3.10]
Subnet mask [255.255.255.0]
C Controller module, CH2
IP address [192.168.4.3]
Subnet mask [255.255.255.0]

Figure 6.39 Example of ping execution from a personal computer (communication via CH1)

<Example where communications are not available>
Normal communications are not available with a personal computer of a different network address through an Ethernet port if the configuration is as shown below.


2) A ping response cannot be sent.
Personal computer
IP address [192.168.3.10]
Subnet mask [255.255.255.0]
C Controller module, CH1*1
IP address [192.168.3.3]
Subnet mask [255.255.255.0]
C Controller module, CH2*2
IP address [192.168.4.3]
Subnet mask [255.255.255.0]
1) ping request
Ethernet

* 1 A personal computer with the same network address is not connected to the Ethernet port.
* 2 A personal computer connected to the Ethernet port has a different network address.

Figure 6.40 Example of ping execution from a personal computer (communication not available)

[Communication details]

1) When a ping addressed to CH2 is executed, a ping request reaches CH2.

2) The C Controller module attempts to send a ping response from CH1 whose network address (192.168.3.0) is the same as that of the personal computer. However, because CH1 is not connected to the network, the ping response will not be sent to the personal computer.

[Corrective action]
When the same network address is set to the personal computer and either of the channels (CH1 in the example), take either of the following collective actions.

- Connect the channel of the same network address as the personal computer to the network so that a ping response will be sent to the personal computer.
- Change the target setting of the personal computer to the channel that has the same network address as the personal computer so that a ping response will be sent to the personal computer.

(b) Flashing "01" or "02" of the 7-segment LED (Q12DCCPU-V only)
At start-up of the C Controller module, "01" or "02" flashes on the 7-segment LED when the C Controller module is:

- In factory-default state.
  (The standard RAM is not initialized.)
- Left unused for a long period of time with no battery installed.
  (The standard and battery-backed-up RAM data are corrupted.)

If "01" or "02" flashes on the 7-segment LED, install a battery and perform the module initialization setting procedure.

6.7 Specifications, Installation, and Replacement of the Battery

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	○

This section explains specifications, installation, and replacement of the battery.

6.7.1 Battery specifications

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	○

The following table indicates the specifications of the batteries for the C Controller module.

Table 6.16 Battery specifications

Item	Model	
	Q6BAT	Q7BAT[*1]
Type	Manganese dioxide lithium primary battery	
Initial voltage	3.0V	
Nominal current	1800mAh	5000mAh
Battery life when stored	Actual 5 years (room temperature)	
Battery life when used	☞ Page 6-51, Section 6.7.3 (2)	
Lithium content	0.57g[*3]	1.55g
Application	Backup of the standard RAM files Protection of the standard ROM files Backup of the battery-backed-up RAM data Backup of clock data	
Accessories	-	Battery holder[*2]

[* 1] For the Q06CCPU-V(-B), the Q7BAT cannot be used.

[* 2] Supplied only with Q7BAT-SET.

[* 3] The lithium content of batteries manufactured in November 2017 or earlier differs from this list. For details, refer to the following.
Technical bulletin No. FA-A-0242

Remark

- For the battery life, refer to Page 6-51, Section 6.7.3 (2).
- For the battery directive in EU member states, refer to Appendix 4.

6.7.2 Battery installation

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	○

The battery connector is disconnected from the battery before shipping.
Connect the battery connector before use.

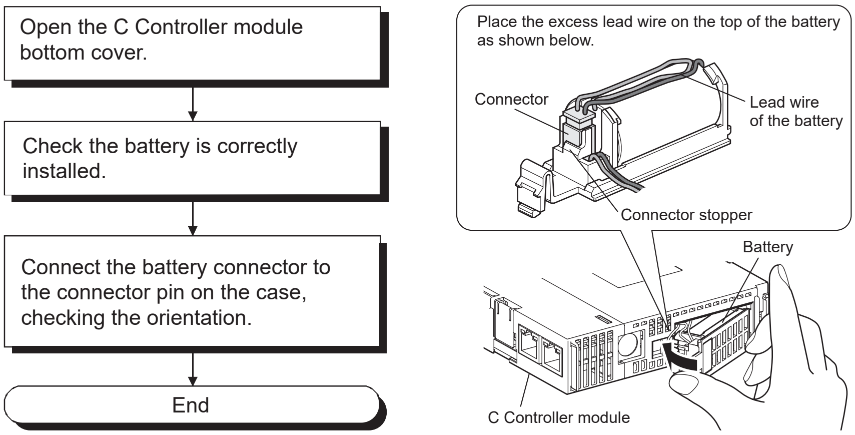

Open the C Controller module bottom cover.
Check the battery is correctly installed.
Connect the battery connector to the connector pin on the case, checking the orientation.
End
Place the excess lead wire on the top of the battery as shown below.
Connector
Lead wire of the battery
Connector stopper
Battery
C Controller module

Figure 6.41 Battery installation

IMPORTANT

A battery must be installed for operation.
When the Q12DCCPU-V is operated without installing a battery and then the C Controller system is powered off or reset, the standard RAM or battery-backed-up RAM data may be damaged, or a file system error may occur.
If the Q06CCPU-V(-B) is operated without battery, set clock data and perform proper shutdown operation. (☞ Page 6-54, Section 6.7.4)

POINT

Firmly push the battery connector all the way in to the connector pin.

6.7.3 Battery replacement

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	○

This section explains the replacement of the C Controller module battery.
The C Controller module battery is used for the following purposes.

- Backup of the standard RAM files
- Protection of the standard ROM files
- Backup of the battery-backed-up RAM data
- Backup of the clock data

The battery must be replaced when the battery voltage becomes low.

(1) Checking for low battery voltage of the C Controller module

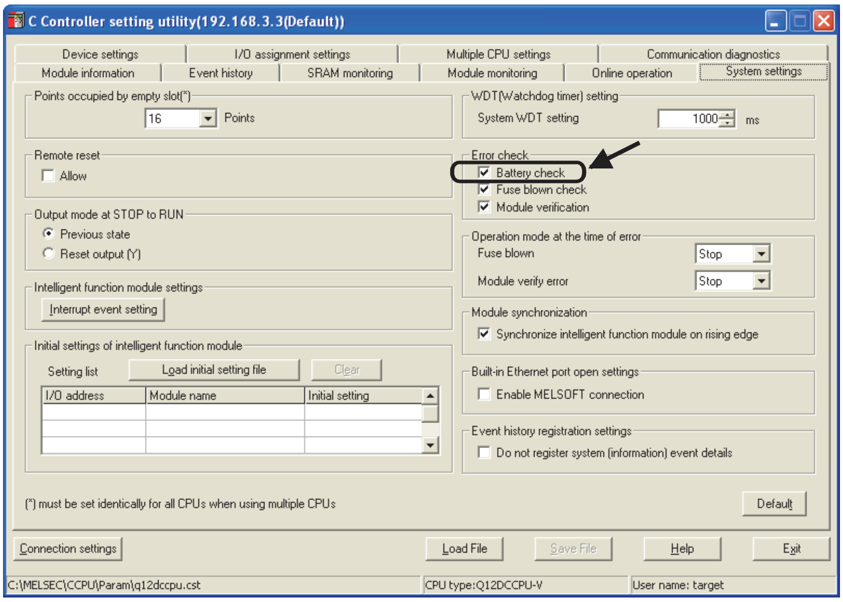
(a) Setting for low battery voltage detection

1) In the <<System settings>> tab of C Controller utility, select "Battery check" in the Error check area.

2) Click the Write button in the <<Online operation>> tab of C Controller utility to write the parameters to the C Controller module.

3) Reset the C Controller module by the RESET/SELECT switch on the module. (In the case of a multiple CPU system, reset CPU No.1.)

A battery error can be detected by the above procedure 1) to 3).



C Controller setting utility(192.168.3.3(Default))

Device settings | I/O assignment settings | Multiple CPU settings | Communication diagnostics
Module information | Event history | SRAM monitoring | Module monitoring | Online operation | System settings

Points occupied by empty slot(*): 16 Points

WDT(Watchdog timer) setting: System WDT setting 1000 ms

Remote reset: □ Allow

Error check: ☑ Battery check ☑ Fuse blown check ☑ Module verification

Output mode at STOP to RUN: ● Previous state ○ Reset output (Y)

Operation mode at the time of error: Fuse blown Stop; Module verify error Stop

Intelligent function module settings: Interrupt event setting

Module synchronization: ☑ Synchronize intelligent function module on rising edge

Initial settings of intelligent function module: Setting list | Load initial setting file | Clear

I/O address	Module name	Initial setting

Built-in Ethernet port open settings: □ Enable MELSOFT connection

Event history registration settings: □ Do not register system (information) event details

(*) must be set identically for all CPUs when using multiple CPUs

Default

Connection settings | Load File | Save File | Help | Exit

C:\MELSEC\CCPU\Param\q12dccpu.cst | CPU type:Q12DCCPU-V | User name: target

Figure 6.42 Setting for low battery voltage detection

(b) When a battery error occurs[*1]

1) The ERR. LED turns on indicating a battery error (error code: 1600, BATTERY ERROR).
The error code can be checked in the <<Module information>> tab of C Controller setting utility.

2) The date and time of the error, event No. (0xC000020E), and battery error (error code: 1600, BATTERY ERROR) are added to the event history.
The event history can be checked in the <<Event history>> tab of C Controller setting utility.

3) By calling the QBF_ReadStatusEx function from the user program, LED information and battery error occurrence can be checked.

[*1] Occurrence of a battery error does not change the CPU operating status.(If it was in the RUN status, it remains in the RUN status.)
File data in the standard RAM, standard ROM, and battery-backed-up RAM will not be lost immediately even if an error occurred.
Note that, however, file data in the standard RAM, standard ROM, and battery-backed-up RAM may be lost if the battery is not replaced within the guaranteed period of time indicated in (2) in this section.

(c) Battery replacement
Replace the battery within the guaranteed period of time after battery error occurrence. (☞ (2) in this section)
To create a battery error detection program, use the QBF_Y_Out_BitEx function to turn on any given output (Y) when a battery error is detected by the QBF_ReadStatusEx function.

(2) Service life of the C Controller module battery

The following indicates the service life of the C Controller module batteries.

Table 6.17 Battery life

C Controller module	Power-on time ratio[1]	Battery life					
		Q6BAT			Q7BAT		
		Guaranteed value[2]	Actual service value (Reference value)[3]	Guaranteed time after battery error occurrence[4]	Guaranteed value[2]	Actual service value (Reference value)[3]	Guaranteed time after battery error occurrence[4]
Q12DCCPU-V	0%	13,700 hours 1.57 years	39,000 hours 4.45 years	600 hours 25 days	37,100 hours 4.25 years	43,800 hours 5 years	600 hours 25 days
	30%	19,100 hours 2.18 years	43,800 hours 5 years	600 hours 25 days	43,800 hours 5 years	43,800 hours 5 years	600 hours 25 days
	50%	25,800 hours 2.96 years	43,800 hours 5 years	600 hours 25 days	43,800 hours 5 years	43,800 hours 5 years	600 hours 25 days
	70%	40,000 hours 4.57 years	43,800 hours 5 years	600 hours 25 days	43,800 hours 5 years	43,800 hours 5 years	600 hours 25 days
	100%	43,800 hours 5 years	43,800 hours 5 years	600 hours 25 days	43,800 hours 5 years	43,800 hours 5 years	600 hours 25 days
Q06CCPU-V, Q06CCPU-V-B	0%	18,000 hours 2.05 years	43,800 hours 5 years	1,100 hours 45 days	For the Q06CCPU-V(-B), the Q7BAT cannot be used.		
	30%	25,714 hours 2.93 years	43,800 hours 5 years	1,100 hours 45 days			
	50%	36,000 hours 4.10 years	43,800 hours 5 years	1,100 hours 45 days			
	70%	43,800 hours 5 years	43,800 hours 5 years	1,100 hours 45 days			
	100%	43,800 hours 5 years	43,800 hours 5 years	1,100 hours 45 days			

[1] The power-on time ratio indicates the percentage of power-on time per day (24 hours). (The power-on time ratio is 50% when the total power-on time is 12 hours and the total power-off or reset time is 12 hours.)

[2] The guaranteed value represents a battery life at 70℃, which is calculated based on characteristic values of manufacturer-supplied memories (SRAM) and on the assumption of storage within the ambient temperature range of -25 to 75℃ (operating ambient temperature of 0 to 55℃).

[3] The actual service value (reference value) represents a battery life that is calculated based on the values measured at storage ambient temperature of 40℃. This value is intended for reference only, as it varies with characteristics of the memory.

[4] The guaranteed time after power off or during reset is 3 minutes in the following cases:
- The battery connector is disconnected.
- The battery lead wire is broken.

IMPORTANT

If a battery is not replaced after the battery error, data in the standard RAM, standard ROM being accessed, or battery-backed-up RAM and/or clock data may be corrupted, or a file system error may occur.

☒**POINT**

1. Use the battery within the time shown by the guaranteed value of the battery life.
2. If the battery may be used exceeding the time shown by the guaranteed battery life value, perform the following.
   - Perform the shutdown operation when powering off or resetting the C Controller module. (☞ Page 6-58, Section 6.7.5 (1))
   - Back up data so that data can be protected even if the battery goes flat at power-off or during reset of the C Controller module.
3. The battery life is five years if the battery has not been installed in the C Controller module.
4. If a battery error occurs, replace the battery as soon as possible although the data are retained within "Guaranteed time after battery error occurrence" shown in Table 6.17.
   Periodic replacement is recommended according to the condition of use even if no battery error is detected.

(3) Battery error detection and log addition to the event history

When "Battery check" is selected in the Error check area in the <<System settings>> tab of C Controller setting utility, the C Controller module detects an error and adds a log to the event history at the following timing.

- When power is turned on
- When the C Controller module is reset
- Periodic check (by the system)

Note that, if a battery error is detected again after another battery error, no battery error will be detected and no log will be added to the event history until power is reapplied or the C Controller module is reset.

(4) Battery replacement procedure

Before the C Controller module battery reaches the limit of its life, replace it by the following procedure.

Before removing the battery, turn on the C Controller system power for 10 minutes or more.

Data will be backed up by the capacitor for a while after removing the battery.

However, quickly replace the battery since the data may be cleared if it takes more than the following guaranteed time.

Table 6.18 Backup power time

Backup power time
3 minutes

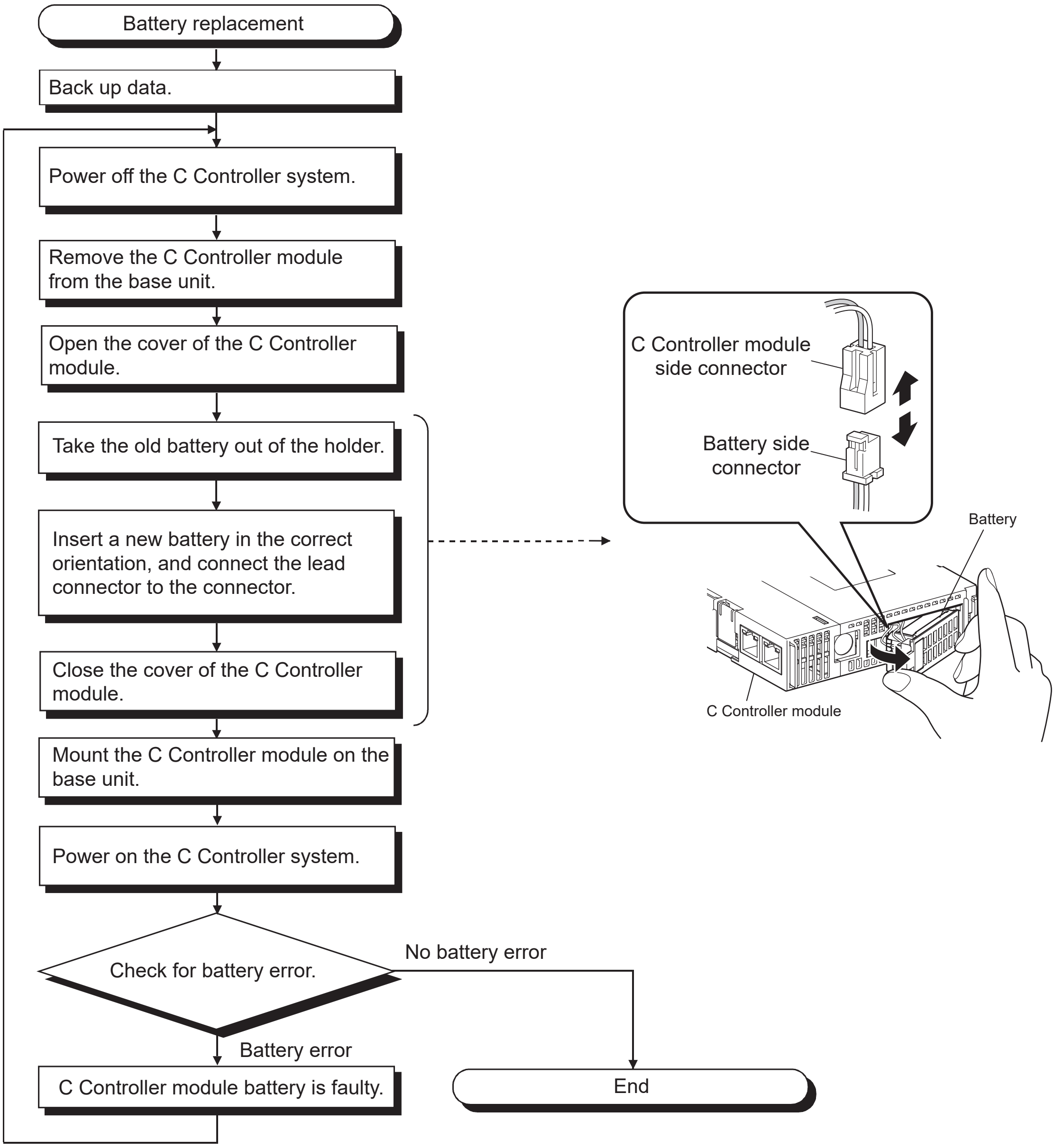

Battery replacement
Back up data.
Power off the C Controller system.
Remove the C Controller module from the base unit.
Open the cover of the C Controller module.
Take the old battery out of the holder.
Insert a new battery in the correct orientation, and connect the lead connector to the connector.
Close the cover of the C Controller module.
Mount the C Controller module on the base unit.
Power on the C Controller system.
Check for battery error.
No battery error
Battery error
C Controller module battery is faulty.
End
C Controller module side connector
Battery side connector
Battery
C Controller module

Figure 6.43 Battery replacement procedure

6.7.4 When the module has been operated without battery

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
×	○	○

This section provides the precautions for having operated the module without battery.

IMPORTANT

A battery must be installed for operation.

If the system has been operated with no battery installed, perform the shutdown operation before powering off or resetting the C Controller system. (☞ (2) in this section)

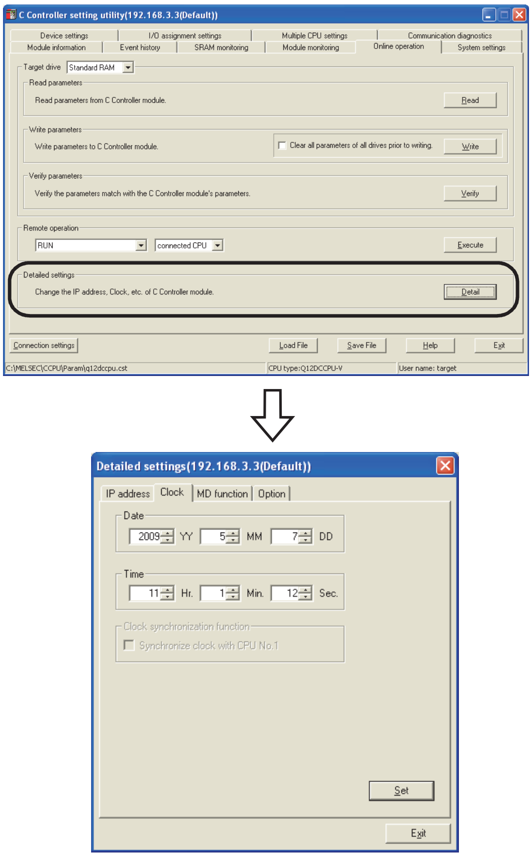
The standard ROM and battery-backed-up RAM data being accessed may be corrupted, or a file system error may occur.

(1) Clock data setting when the C Controller system is powered on without battery installed (clock data not guaranteed)

If the C Controller system is powered on without a battery installed in the C Controller module, the system starts up with unreliable clock data.

If this occurs, the event history file and/or programs using clock data will not operate properly. Set clock data by either of the following methods.

- Open the <<Online operation>> tab of C Controller setting utility, click the Detail button, and set clock data in the <<Clock>> tab.



C Controller setting utility(192.168.3.3(Default))

Target drive: Standard RAM

Read parameters: Read parameters from C Controller module.

Write parameters: Write parameters to C Controller module.

□ Clear all parameters of all drives prior to writing.

Verify parameters: Verify the parameters match with the C Controller module's parameters.

Remote operation: RUN, connected CPU

Detailed settings: Change the IP address, Clock, etc. of C Controller module.

C:\MELSEC\CCPU\Param\q12dccpu.cst | CPU type:Q12DCCPU-V | User name: target

Detailed settings(192.168.3.3(Default))

Date: 2009 YY 5 MM 7 DD

Time: 11 Hr. 1 Min. 12 Sec.

Clock synchronization function: □ Synchronize clock with CPU No.1

Figure 6.44 Clock data setting

- Call the QBF_SetTime function from the user program.

☒POINT

Use the QBF_SetTime function when setting clock data from the user program. Calling a clock data setting function of ANSI-C or VxWorks does not set clock data correctly.

(2) Shutdown operation required for powering off or resetting the C Controller system without a battery installed

Perform the shutdown operation (☞ Page 6-58, Section 6.7.5 (1)) when powering off or resetting the C Controller system with no battery installed in the C Controller module.

If the system is powered off or reset with no battery installed, the following data will be lost.

- Parameter files and event history files in the standard ROM, data being processed, user programs, and data files
- Battery-backed-up RAM data, etc.

(3) Disabling battery error detection

A battery error occurs if a battery is not installed in the C Controller module.[*1]
To disable the battery error detection, perform the procedure (a) below.

[*1] Occurrence of a battery error does not change the CPU operating status.(If it was in the RUN status, it remains in the RUN status.)

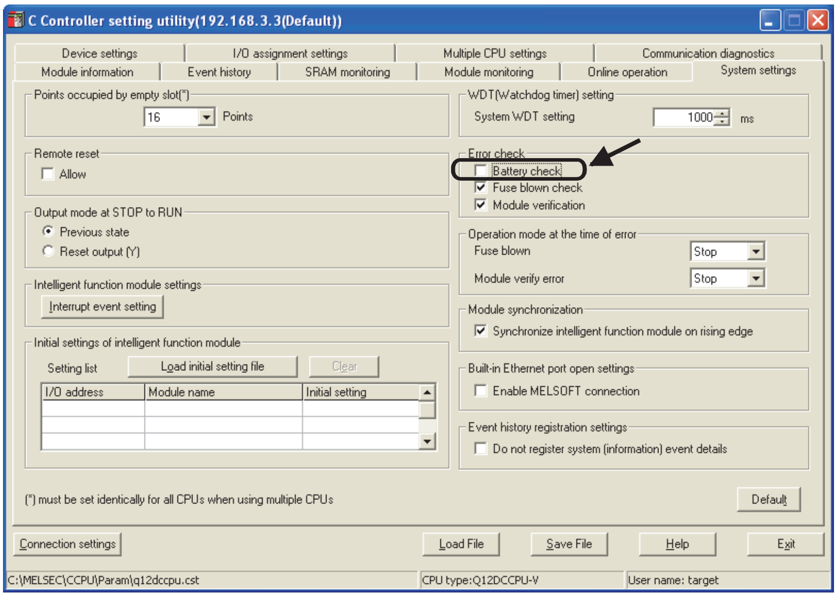
(a) Disabling battery error detection

1) In the <<System settings>> tab of C Controller utility, deselect "Battery check" in the Error check area.

2) Click the Write button in the <<Online operation>> tab of C Controller utility to write the parameters to the C Controller module.

3) Reset the C Controller module by the RESET/SELECT switch on the module. (In the case of a multiple CPU system, reset CPU No.1.)

A battery error will not be detected by the above procedure 1) to 3).



Points occupied by empty slot(*): 16 Points

Remote reset: □ Allow

Output mode at STOP to RUN: ● Previous state ○ Reset output (Y)

Intelligent function module settings

Initial settings of intelligent function module
Setting list

I/O address	Module name	Initial setting

WDT(Watchdog timer) setting: System WDT setting 1000 ms

Error check
□ Battery check
☑ Fuse blown check
☑ Module verification

Operation mode at the time of error
Fuse blown: Stop
Module verify error: Stop

Module synchronization
☑ Synchronize intelligent function module on rising edge

Built-in Ethernet port open settings
□ Enable MELSOFT connection

Event history registration settings
□ Do not register system (information) event details

(*) must be set identically for all CPUs when using multiple CPUs

C:\MELSEC\CCPU\Param\q12dccpu.cst | CPU type:Q12DCCPU-V | User name: target

Figure 6.45 Disabling battery error detection

6.7.5 Removing a battery before storage

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	○

To store the C Controller module with a battery removed, perform the shutdown operation, power off the C Controller system, and then remove the battery.

IMPORTANT

Data in the standard RAM, standard ROM being accessed, or battery-backed-up RAM may be corrupted or a file system error may occur if the procedure given below is not followed.

[Operating procedure]

1) Mount the C Controller module on the base unit and power it on.
2) Perform the shutdown operation (☞ (1) in this section), and power off the C Controller system.
3) Remove the battery from the C Controller module.

(1) Shutdown operation

(a) For the Q12DCCPU-V
Determine a shutdown request signal (Xm) in advance.
Create a program to perform the following operations by using the QBF_X_In_BitEx function or various file access (close) functions.

1) Turn the shutdown request signal (Xm: arbitrary) from off to on by using an external input.

2) Stop file access to the standard RAM, battery-backed-up RAM, and CompactFlash card, and close the opened files.*1
(Use the QBF_X_In_BitExXm function and the close or fclose function in programming so that this operation will be performed when Xm turns on.)

3) Save the battery-backed-up RAM data to a file as necessary.*1
(Use the QBF_ReadSRAM function in programming. Alternatively, save the data by clicking the Save data button in the SRAM monitoring tab of C Controller setting utility.

4) Store all of the standard RAM data on the CompactFlash card.
(Program example)

```
mkdir("/CF/BackUp");
xcopy("/RAM","/CF/BackUp");
sysLedSet(0xFF);
```

5) Unmount the CompactFlash card.*1
(☞ Page 6-60, Section 6.8)

6) "FF" on the 7-segment LED indicates the shutdown is complete.

7) After the shutdown, power off the C Controller system.

* 1 When the CompactFlash card files have not been accessed, closing the files in the CompactFlash card and unmounting are not required.

(b) For the Q06CCPU-V(-B)

Determine a shutdown request signal (Xm) in advance.
Create a program to perform the following operations by using the QBF_X_In_BitEx function, QBF_ShutdownRom function, and various file access (close) functions.
(Use "Standard ROM shutdown status" of the QBF_ReadStatusEx function as necessary.)

1) Turn the shutdown request signal (Xm: arbitrary) from off to on by using an external input.
2) Stop file access to the standard ROM, battery-backed-up RAM, and CompactFlash card, and close the opened files.[*1]
   (Use the QBF_X_In_BitExXm function and the close or fclose function in programming so that this operation will be performed when Xm turns on.)
3) Unmount the CompactFlash card.[*1]
   (☞ Page 6-60, Section 6.8)
4) Shutdown the standard ROM.
   (Use the QBF_ShutdownRom function in programming.)
5) When the MODE LED is flashing, the shutdown is complete.
   (To confirm the shutdown status in the user program, check the "MODE LED status" or "Standard ROM shutdown status" of the QBF_ReadStatusEx function.)
6) After the shutdown, power off the C Controller system.

[*1] When the CompactFlash card files have not been accessed, closing files in the CompactFlash card and unmounting are not required.

IMPORTANT

1. When power is turned off or the system is reset with no battery installed and without performing the above shutdown operation, data in the standard RAM, standard ROM, and battery-backed-up RAM may be corrupted, or a file system error may occur.
   Perform the shutdown operation before powering off or resetting the C Controller system when a battery is not installed in the module.
2. Power off or reset the C Controller system after the shutdown.
   Continuing the C Controller module operation without powering off or resetting will disable normal operation of the module.

Remark

For a sample shutdown program for the C Controller module, refer to the following.

☞ C Controller Module User's Manual (Utility Operation, Programming)

6.8 Inserting/Removing a CompactFlash Card and Access Stop

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	×

This section explains how to insert and remove a CompactFlash card and how to stop access to the CompactFlash card.

IMPORTANT

Before handling a CompactFlash card, touch a grounded metal object to discharge the static electricity from the human body.
Failure to do so may cause failure of the module or malfunction.
Do not power off or reset the system, or remove the CompactFlash card while a file is written to the CompactFlash card.
Doing so can cause corruption of the CompactFlash card data or a file system error.

- To remove the CompactFlash card during file writing: ☞ Page 6-64, Section 6.8.2
- To power off or reset the system: ☞ Page 8-9, Section 8.2.2

6.8.1 Inserting/removing the CompactFlash card

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	×

This section explains how to insert and remove a CompactFlash card.

IMPORTANT

Before handling a CompactFlash card, touch a grounded metal object to discharge the static electricity from the human body.
Failure to do so may cause failure of the module or malfunction.
Insert the CompactFlash card according to the procedure in this section. Failure to do so can cause a corruption of the CompactFlash card data or a file system error.

Insert or remove a CompactFlash card according to the following procedure.
Especially, when removing or replacing the CompactFlash card, stop accessing to the CompactFlash card.

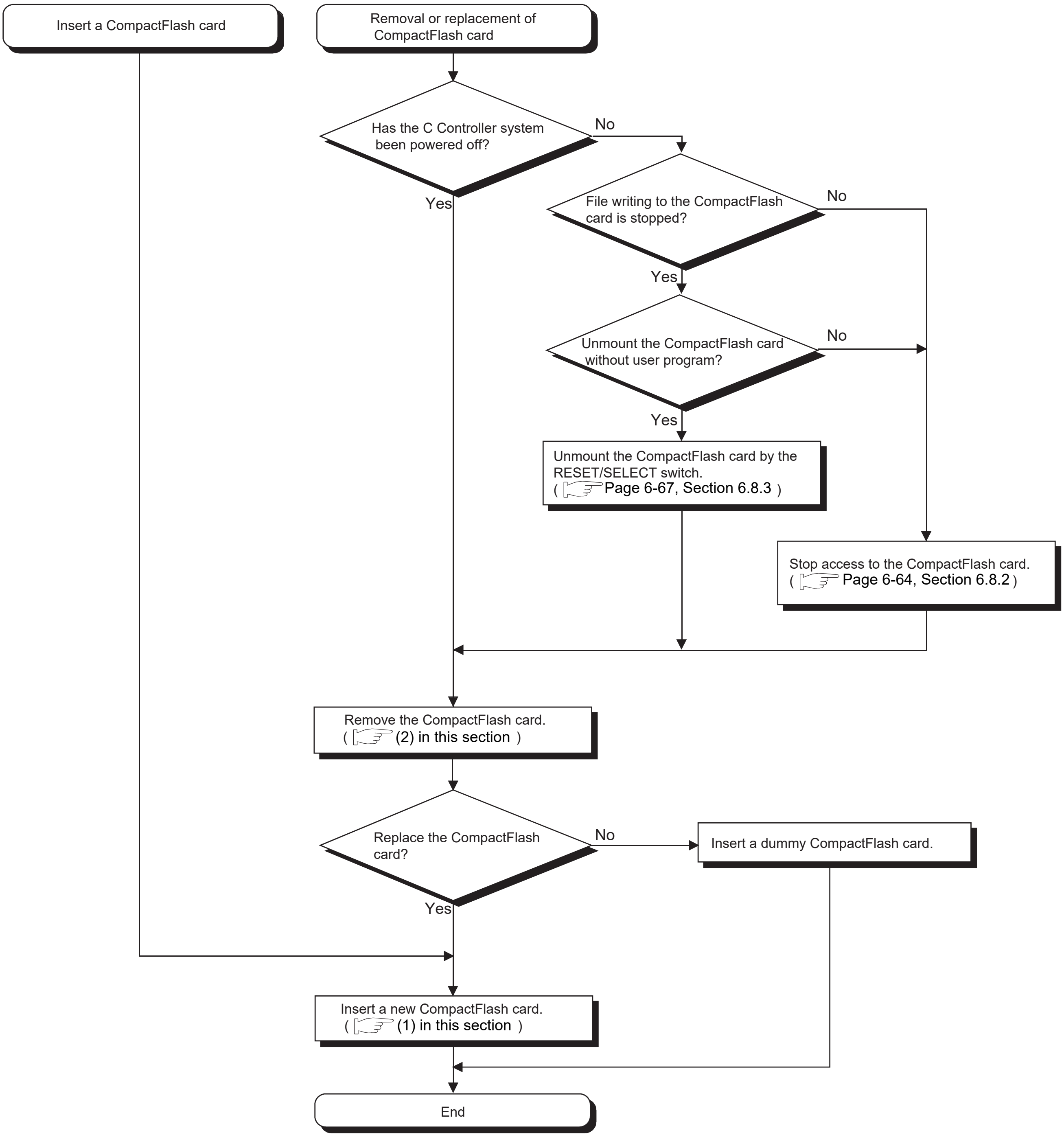

Insert a CompactFlash card
Removal or replacement of CompactFlash card
Has the C Controller system been powered off?
No
Yes
File writing to the CompactFlash card is stopped?
No
Yes
Unmount the CompactFlash card without user program?
No
Yes
Unmount the CompactFlash card by the RESET/SELECT switch. (Page 6-67, Section 6.8.3)
Stop access to the CompactFlash card. (Page 6-64, Section 6.8.2)
Remove the CompactFlash card. ((2) in this section)
Replace the CompactFlash card?
No
Yes
Insert a dummy CompactFlash card.
Insert a new CompactFlash card. ((1) in this section)
End

Figure 6.46 Removing or replacing the CompactFlash card

(1) Inserting a CompactFlash card

(a) Removing the dummy CompactFlash card
Open the C Controller module front cover, and remove the dummy CompactFlash card.

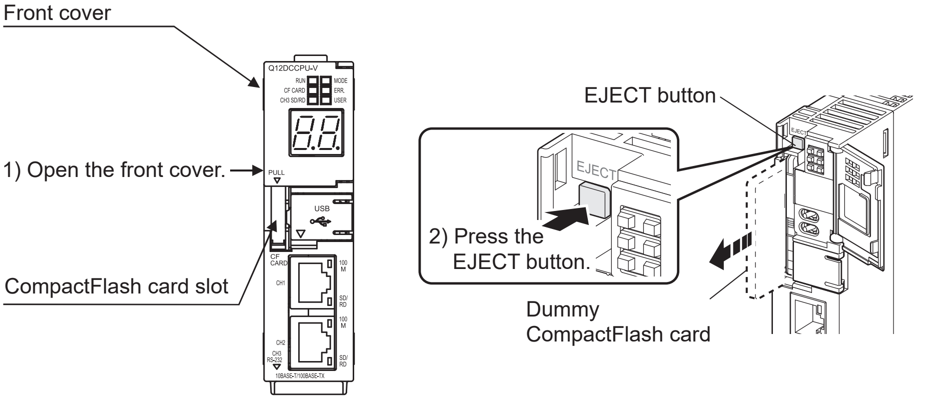


Figure 6.47 Removing the dummy CompactFlash card (for the Q12DCCPU-V)

1) Put your finger on the bottom of the C Controller module front cover, and open the front cover.

2) Press the EJECT button to remove the dummy CompactFlash card.

(b) Inserting a CompactFlash card
When inserting a CompactFlash card into the C Controller module, pay attention to its orientation.
Insert the CompactFlash card into the slot and push it until the card edge is flush with the EJECT button.

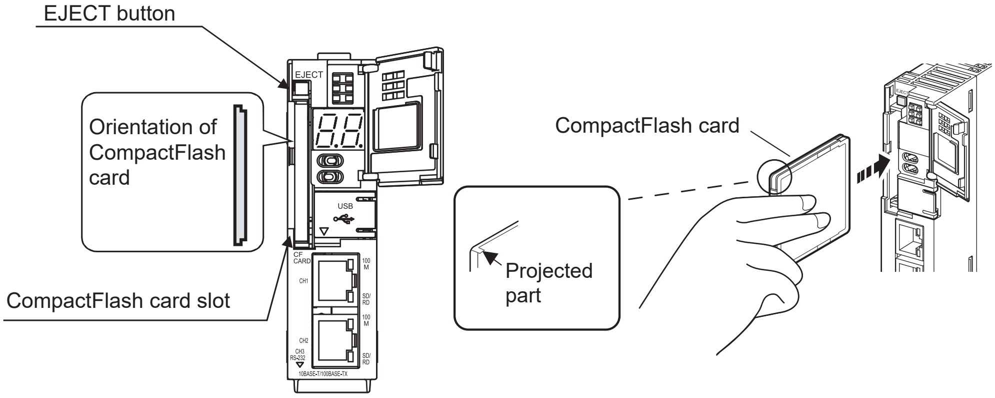


Figure 6.48 Inserting a CompactFlash card (for the Q12DCCPU-V)

(c) Closing the front cover
Close the C Controller module front cover until it clicks.
Store the removed dummy CompactFlash card in a safe place for future use.

(d) Checking the installation
Confirm that the CompactFlash card has been correctly installed in the C Controller module.
(Check that the CF CARD LED turns on when the module is powered on.)

(2) Removing the CompactFlash card

(a) Removing the CompactFlash card
Open the C Controller module front cover, and remove the CompactFlash card.

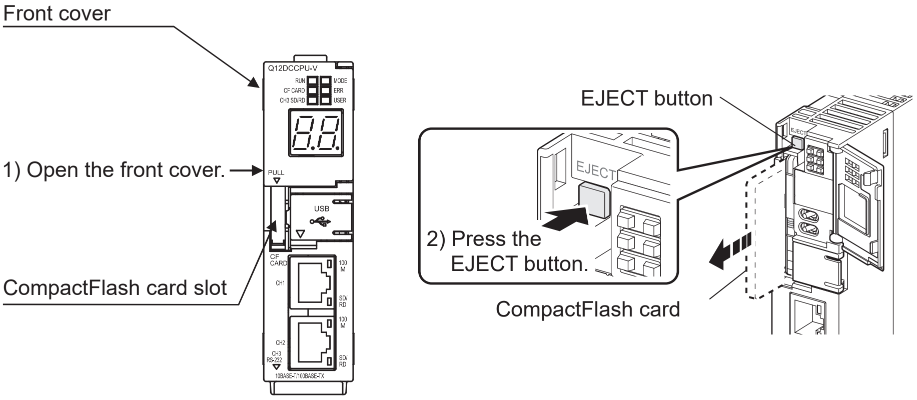


Figure 6.49 Removing the CompactFlash card (for the Q12DCCPU-V)

1) Put your finger on the bottom of the C Controller module front cover, and open the front cover.
2) Press the EJECT button to remove the CompactFlash card.

(b) Inserting a dummy CompactFlash card
While no CompactFlash card is used, a dummy CompactFlash card must be used.

(c) Closing the front cover
Confirm that the C Controller module front cover is closed with a clicking sound.

6.8.2 Stopping access to the CompactFlash card

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	×

To replace the CompactFlash card while the C Controller system power is on, access to the CompactFlash card must be stopped.
This section describes how to stop access to the CompactFlash card.

(1) Access stop procedure

Determine a file access stop request signal (Xn) in advance.
Create a program to perform the following operations by using the QBF_X_In_BitEx function, QBF_UnmountCfCard function, and various file access (open/close) functions.
(Use "CF CARD status" of the QBF_ReadStatusEx function as necessary.)

(a) To remove the CompactFlash card while power is on

IMPORTANT

1. Before handling a CompactFlash card, touch a grounded metal object to discharge the static electricity from the human body.
   Failure to do so may cause the module to fail or malfunction.
2. Do not power off or reset the system or remove the CompactFlash card during file writing to the CompactFlash card.
   Doing so may cause a corruption of the CompactFlash card data or a file system error.
3. To remove the CompactFlash card while a file is written to, perform the procedure for stopping access described in this section.
   For powering off or reset, refer to the following.
   ☞ Page 8-9, Section 8.2.2

1) Turn the file access stop request signal (Xn: arbitrary) from off to on by using an external input.

2) Stop the file access to the CompactFlash card, and close the opened files. (Use the QBF_X_In_BitExXm function and the close or fclose function in programming so that this operation will be performed when Xn turns on.)

3) Unmount the CompactFlash card by using the QBF_UnmountCfCard function.

4) When unmounting of the CompactFlash card is confirmed (the CF CARD LED is off), the card can be removed.
To confirm the unmount status in the user program, check the "CF CARD LED status" or "CF CARD status" of the QBF_ReadStatusEx function.
To confirm the unmount status in C Controller setting utility, check "CF CARD" in the "LED information" area in the <<Module information>> tab.

5) Turn off the file access stop request signal (Xn: arbitrary).
(Xn can also be requested by a pulse input in step 1).)

6) Remove the CompactFlash card.

(b) When inserting a CompactFlash card while power is on

IMPORTANT

Before the following operation, touch a grounded metal object to discharge the static electricity from the human body.
Failure to do so may cause failure of the module or malfunction.
Insert a CompactFlash card according to the above procedure. Failure to do so can cause a corruption of the CompactFlash card data or a file system error.

1) Insert a CompactFlash card.
2) Check that the CompactFlash card is mounted (the CF CARD LED turns on).
   When the CF CARD LED turns on, file access to the CompactFlash card is available.
   To confirm the mount status in the user program, check the "CF CARD LED status" or "CF CARD status" of the QBF_ReadStatusEx function.[*1]
   To confirm the mount status in C Controller setting utility, check the "CF CARD" in the "LED information" area in the <<Module information>> tab.
3) Open a file in the CompactFlash card (by using a function such as the open or fopen function) to start file access.

* 1 When the CompactFlash card is inserted while power is on, the card is automatically mounted. To access a file again without removing the unmounted CompactFlash card, the card must be manually mounted. In this case, call the QBF_MountCfCard function in a program.

Remark

For bus interface functions and a sample program for access stop, refer to the following.
☞ C Controller Module User's Manual (Utility Operation, Programming)

6.8.3 Unmounting the CompactFlash card by the RESET/SELECT switch

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	×

The CompactFlash card can be unmounted by the RESET/SELECT switch.
Use this unmounting method only when file writing to the CompactFlash card is not in execution.

(1) Unmounting procedure by the RESET/SELECT switch

IMPORTANT

1. Do not unmount the CompactFlash card by the RESET/SELECT switch during file writing to the card. Doing so can cause corruption of CompactFlash card data or a file system error.
   To remove the CompactFlash card during file writing, stop access to the CompactFlash card.
   (☞ Page 6-64, Section 6.8.2)
2. For powering off or resetting, refer to the following.
   ☞ Page 8-9, Section 8.2.2
3. If the RESET/SELECT switch is held in the RESET position by mistake, the C Controller module will be reset.
   Be careful when unmounting the CompactFlash card by operating the RESET/SELECT switch.

1) Hold the RESET/SELECT switch in the SELECT position.
   When unmounting is started, the CF CARD LED on the C Controller module starts flashing. Hold the switch in the position.
2) When unmounting is completed, the CF CARD LED turns off.
   After confirming that the LED has turned off, return the switch to the central position.
   Unmounting cannot be completed if the RESET/SELECT switch is returned to the central position before the CF CARD LED turns off.

Remark

For details of the RESET/SELECT switch operations, refer to the following.
Q12DCCPU-V: ☞ Page 6-29, Section 6.4.1 (4)
Q06CCPU-V(-B): ☞ Page 6-36, Section 6.4.2 (3)

6.8.4 Measures against static electricity for commercially available CompactFlash cards in compliance with the EMC directives

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	×

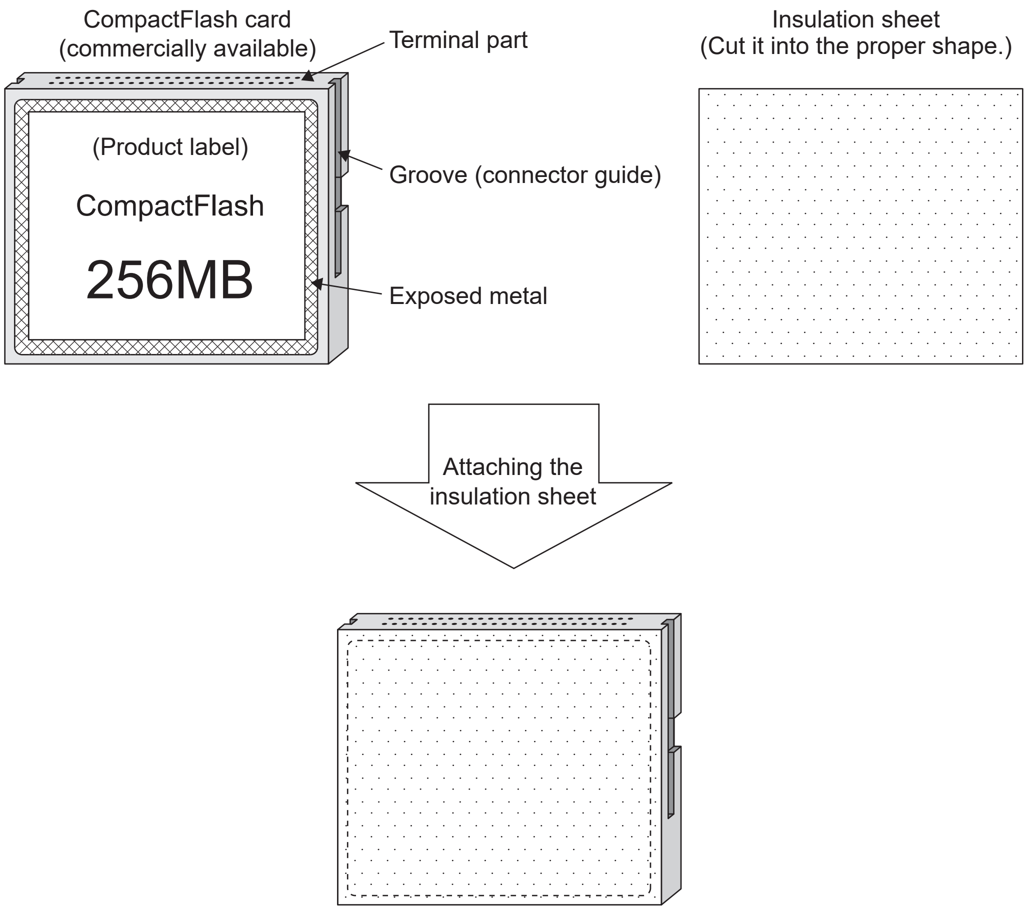
The C Controller system may be affected by static electricity discharged to a commercially available CompactFlash card.
This section explains measures for preventing static electricity discharged to a commercially available CompactFlash card.

☒POINT

Although information in this section is based on the requirements and standards of current regulations, it does not guarantee that the entire machinery constructed according to the information will comply with the EMC Directive.
The compliance with the EMC Directive must be finally confirmed by the manufacturer of each machinery.

(1) Preventive measure example

Fully cover all the exposed metal part of the CompactFlash card except the terminal part with an insulation sheet as shown below.


CompactFlash card (commercially available)
Terminal part
Insulation sheet (Cut it into the proper shape.)
(Product label)
CompactFlash
256MB
Groove (connector guide)
Exposed metal
Attaching the insulation sheet

Attach another sheet on the other side as well.

Figure 6.50 Attaching insulation sheets to a CompactFlash card

(2) Insulation sheet example

Single-sided adhesive sheet for electric insulation

- Base material: 50μm-PET film (UL compliant)
- Adhesive: Acrylic pressure sensitive adhesive

(3) Precautions

The insulation sheet must be as large as the CompactFlash card. Make sure that it is not oversized and does not cover the groove (connector guide) on the side of the CompactFlash card.

If any of these conditions occurs, the CompactFlash card cannot be inserted into the C Controller module properly.

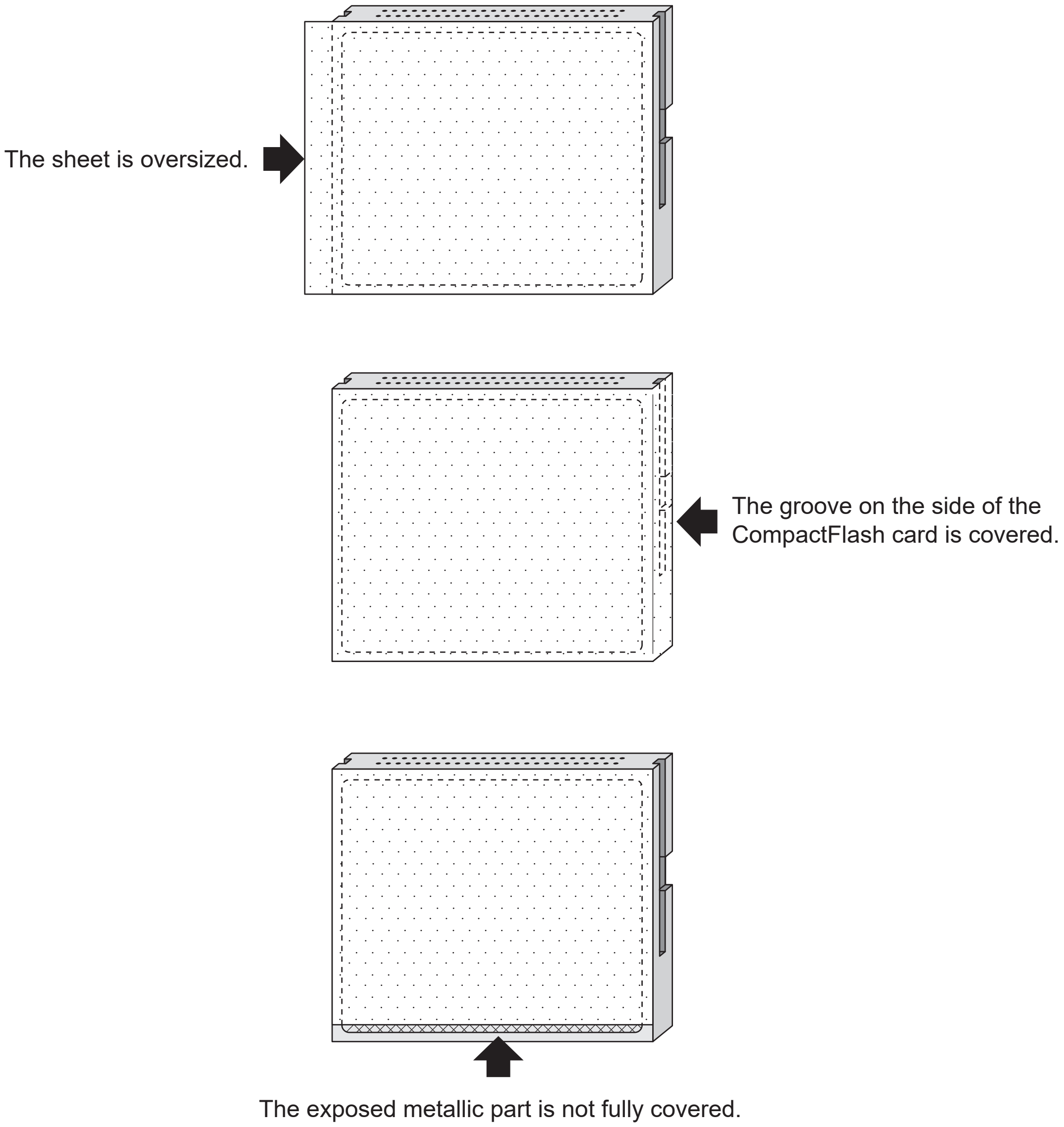

The sheet is oversized.
The groove on the side of the CompactFlash card is covered.
The exposed metallic part is not fully covered.

Figure 6.51 Attaching insulation sheet to a CompactFlash card (bad examples)

6.8.5 Life of CompactFlash card

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	×

A life of CompactFlash card is limited (a limited number of writing). The life calculation method is shown below. The actual life varies depending on operating conditions.
Use this as a guide to estimate the time for replacement.

(1) For the QD81MEM

(a) Calculation of life
The life of a CompactFlash card can be calculated as follows:

Life of CompactFlash card (years)
Total writable size (G bytes)/write size per year (byte/year) × 10^9

(b) Total writable size
The total writable size of CompactFlash card is as follows:

Table 6.19 Total writable size

Model	Total writable size [*1] (G bytes)
QD81MEM-512MBC	0.5G byte (512Mbytes) × 100,000 = 50,000
QD81MEM-1GBC	1G byte × 100,000 = 100,000
QD81MEM-2GBC	2G bytes × 100,000 = 200,000
QD81MEM-4GBC	4G bytes × 100,000 = 400,000
QD81MEM-8GBC	8G bytes × 100,000 = 800,000

* 1 The total writable size means the total size that can be written until a CompactFlash card reaches the end of life (the limitation of writing).

(2) For GT05-MEM

For the life of a CompactFlash card, refer to the following.

Table 6.20 Life of a CompactFlash card

Model	Reference
GT05-MEM-128MC	GOT1000 Series CF Card/Memory Card Adaptor User's Manual
GT05-MEM-256MC	

6.9 Checking the Number of Erasures on the Standard ROM

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
×	○	○

The standard ROM built in the C Controller module is used to save various parameters and user programs.
Since a flash ROM is used as the standard ROM, the number of writes (life) is limited.
The life of the standard ROM must be taken into account when data are regularly written to it.
This section describes the standard ROM life and how to check writing frequency.

(1) Standard ROM life

Data can be written to the same area of the standard ROM to the limit of 100,000 times.
The following shows the standard ROM life (reference value) in the case where files are copied (written) every day.

Standard ROM life (days) = 600000/total size of files copied per day (MB)[*1]

[*1] When the file size is 512 bytes or less, plug 512 into the formula.

(2) Checking writing frequency to the standard ROM

Writing frequency to the standard ROM can be checked by obtaining the maximum number of erasures of the standard ROM from the user program by using a function.
Include the following function in the user program so that the standard ROM life can be estimated.

- Obtain the number of erasures by using plInfo[7] of the QBF_ReadStatusEx function.
  If the returned value exceeds 100,000, writing to the standard ROM may become slower or result in an error.
  Also, if this occurs, the event is registered to the event history.
  Replace the C Controller module when the number of erasures has reached to 100,000 times.

☒POINT

Do not write data to any file in the standard ROM from a user program.
To write data to a file from a user program, use a file on the CompactFlash card.

6.10 Initializing/Changing Mode of C Controller Module

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	○

6.10.1 Initializing/Changing mode of Q12DCCPU-V

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	×	×

(1) Initialization setting for C Controller module

Data such as parameter files saved in the C Controller module can be initialized by performing the following operations.

- The "Default IP setting" mode execution of the C Controller module
  (☞ Page 6-75, Section 6.10.1 (5))
- The module initialization setting procedure of the C Controller module
  (☞ Page 6-77, Section 6.10.1 (6))

(2) Changing mode of Q12DCCPU-V (Basic mode/Extended mode)

The mode switching of the modules can be performed by the initialization setting.
For operation of switching to the extended mode, refer to the following manual.
☞ MELSEC-Q C Controller Module User's Manual

Remark

To initialize login user information of the C Controller module, refer to the following.
☞ Page 6-89, Section 6.11.2

(3) Applications

The C Controller module can be restored by setting it to the initialized status if it does not operate normally by the following reasons.

- Execution of a user program registered to a script file
- Unknown IP address or login user set to the C Controller module

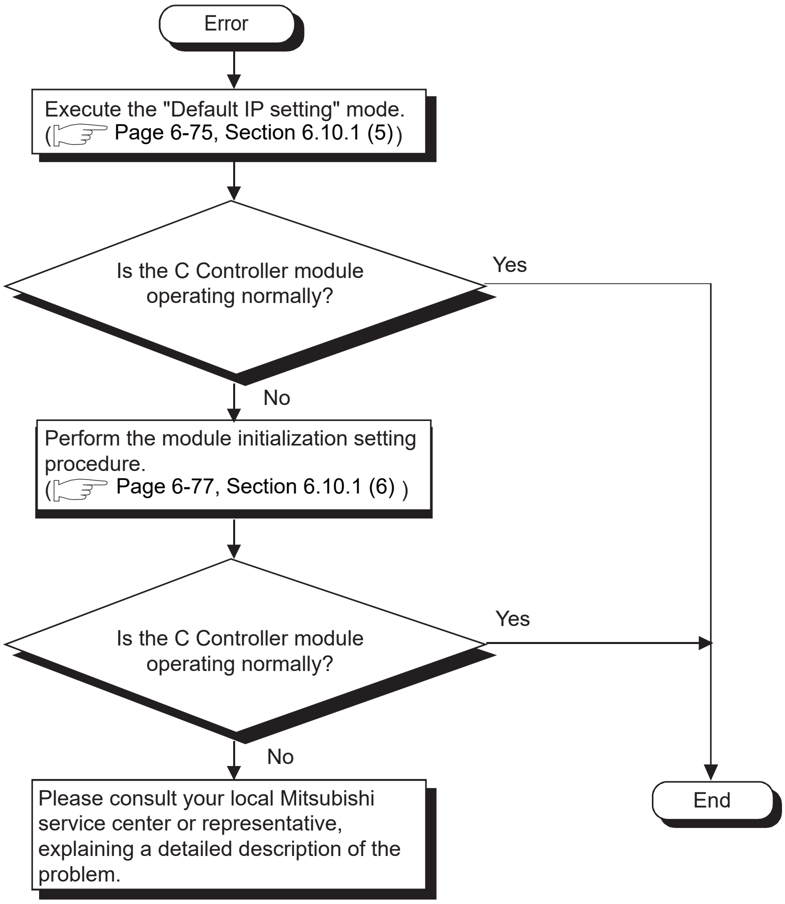

Error
Execute the "Default IP setting" mode. (Page 6-75, Section 6.10.1 (5))
Is the C Controller module operating normally?
Yes
No
Perform the module initialization setting procedure. (Page 6-77, Section 6.10.1 (6))
Is the C Controller module operating normally?
Yes
No
Please consult your local Mitsubishi service center or representative, explaining a detailed description of the problem.
End

Figure 6.52 Operating procedure

(4) Settings and file conditions after the operations

The following table indicates the settings and file conditions after executing the "Default IP setting" mode or performing the module initialization setting procedure.

Table 6.21 File conditions

File type	Settings and file conditions	
	Default IP setting	Module initialization setting
IP address setting	*1	*1
Event history file	—	*2
C Controller module parameters	*3	Default parameters
CC-Link module parameters	*3	Deleted
MELSECNET/H module parameters	*3	Deleted
CC-Link IE controller network module parameters	*3	Deleted
Intelligent function module parameters	*3	Deleted
Script file	*4	Deleted
User program	*3	Deleted
Files used in the user program	*3	Deleted

* 1 The setting returns to the default IP address.
Default IP address for CH1: 192.168.3.3
Default IP address for CH2: No setting

* 2 Previous information is deleted.

* 3 Previous data before execution of the default IP setting mode are retained.

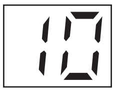
* 4 The startup script file "STARTUP.CMD" in the standard RAM is deleted.
The "STARTUP.CMD" saved in the standard RAM before execution of the default IP setting mode is renamed to "STARTUP.BAK" and saved to the standard RAM. (If "STARTUP.BAK" already exists, it is overwritten.)

(5) Default IP setting mode

To set the IP address back to default, start the C Controller module in the hardware self-diagnostic mode (☞ Page 16-75, Section 16.5.1), and then execute the "Default IP setting" mode.

1) Start up the C Controller module in the hardware self-diagnostic mode (with the RUN/STOP/MODE switch set to MODE).
(The MODE LED turns on in orange.)

2) Repeatedly move the RESET/SELECT switch to SELECT until the 7-segment LED indicates "10".
Check that the 7-segment LED is indicating "10" and the current mode is "Default IP setting".



3) Set the RUN/STOP/MODE switch to RUN to execute the "Default IP setting" mode.
(The RUN LED starts flashing.)
When the "Default IP setting" mode is activated:
   - The RUN LED turns off.
   - The 7-segment LED indicates "00".

4) In the "Default IP setting" mode, reset or restart the C Controller module.
After resetting or restarting the C Controller module, the module will be in the following status:
   - Default IP address of CH1: 192.168.3.3
     Default IP address of CH2: No setting
   - Script file: None

5) Access the C Controller module from C Controller setting utility using the default IP address of CH1 (192.168.3.3) and write default parameters.
Before writing parameters, open the <<Online operation>> tab and select "Clear all parameter before writing" in the "Write parameters" area.
Resetting or restarting the C Controller module sets the module into the initialized status.

☒POINT

1. When the "Default IP setting" mode is executed with a CompactFlash card inserted in the C Controller module, the script file "STARTUP.CMD" in the CompactFlash card will not be deleted.
   To set the script file in the CompactFlash card back to its original status, edit it on a personal computer.
2. If the C Controller module does not operate normally even after execution of the "Default IP setting" mode, perform the module initialization setting procedure.
   (☞ Page 6-77, Section 6.10.1 (6))

(6) Module initialization setting

Each file in the standard RAM will be deleted after performing the module initialization setting procedure.
Before operating the procedure provided below, back up necessary data, user programs, and parameter files. (☞ Page 8-7, Section 8.2.1)

☒POINT

When using Q12DCCPU-V with a serial number whose first five digits are "15102" or later

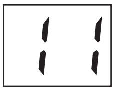
- When performing the initialization, select Mode '11' module initialization setting (Basic mode). When Mode '12' module initialization setting (Extended mode) is selected, the module cannot be connected to SW3PVC-CCPU.
  For the specifications or function of Mode '12' module initialization setting (Extended mode), refer to the following manual.
  ☞ MELSEC-Q C Controller Module User's Manual

(a) Initializing the module by switch operation

Start up the C Controller module in the hardware self-diagnostic mode (☞ Page 16-75, Section 16.5.1), and perform the "Module initialization setting" procedure.

1) Startup the C Controller module in the hardware self-diagnostic mode (with the RUN/STOP/MODE switch set to MODE).
(The MODE LED turns on (orange).)

2) Repeatedly move the RESET/SELECT switch to SELECT until the 7-segment LED indicates "11".
Check that the 7-segment LED is indicating "11" and the current mode is "Module initialization setting".



3) Set the RUN/STOP/MODE switch to RUN to execute the "Module initialization setting" mode. (The RUN LED starts flashing.)
When the "Module initialization setting" mode is activated:
- The RUN LED turns off.
- The 7-segment LED indicates "00".

4) After confirming the above, restart the C Controller module.

5) The standard RAM will be formatted if the C Controller module is restarted.
(The RUN LED flashes and the USER LED flashes green.)

6) When the RUN and USER LEDs stop flashing and the MODE LED is flashing green, the standard RAM formatting is completed.
The ERR. LED will flash if the standard RAM formatting is terminated in error.
(Redo the above procedures if this occurs.)

7) Reset the C Controller module.

Remark

Do not power off or reset the C Controller module while the standard RAM is being formatted.
Doing so will cause the standard RAM to be formatted again when the C Controller module is started at the next time.

(b) Initializing the module by creating a script file in the CompactFlash card

1) Format the standard RAM by using a sample script file for standard RAM formatting.
Write the following script file "STARTUP.CMD" to the CompactFlash card, insert it into the C Controller module, and then start the C Controller module.
(The RUN LED flashes during processing the script file "STARTUP.CMD".)

<Sample script file for standard RAM formatting[*1]>

```
STARTUP.CMD
    diskFormatForCCPU("/RAM")
    sysLedSet(0xFF)
```

[* 1] The sample script file for standard RAM formatting is registered to the development environment (personal computer) after installation of SW3PVC-CCPU.

2) The 7-segment LED indicates "FF" upon completion of standard RAM formatting and shutdown processing.
The ERR. LED flashes if the standard RAM formatting or shutdown processing is terminated in error.
(Redo the above procedures if terminated in error.)

3) After confirming that the 7-segment LED is indicating "FF", power off the C Controller module.

4) Remove the CompactFlash card.

5) After formatting the standard RAM
Files such as parameters and user programs are deleted. (☞Page 6-74, Section 6.10.1 (4))
(After formatting the standard RAM, resetting the module causes a parameter error (no parameters) and the ERR. LED flashes.)
Access the C Controller module from C Controller setting utility by using the default IP address of CH1 (192.168.3.3) and write default parameters (or any given parameters).
Reset the C Controller module after writing parameters and setting the IP address.

IMPORTANT

1.To format the standard RAM, perform the method provided in (6) in this section.
Do not format it by using a command from Workbench Shell.

2.Do not power off or reset the C Controller module while the standard RAM is being formatted.
Doing so will cause the standard RAM to be formatted again when the C Controller module is started at the next time.

Remark

For script files and sample programs, refer to the following.

☞ C Controller Module User's Manual (Utility Operation, Programming)

6.10.2 Setting the Q06CCPU-V(-B) back to the factory-set status

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
×	○	○

Data such as parameter files saved in the C Controller module can be initialized to the factory-set status by performing the following operations.

- Executing the "Default IP setting" mode of the C Controller module
  (☞ Page 6-84, Section 6.10.2 (3))
- Formatting the standard ROM of the C Controller module
  (☞ Page 6-85, Section 6.10.2 (4))

Remark

For initialization of login user information of the C Controller module, refer to the following.

☞ Page 6-89, Section 6.11.2

(1) Applications

The C Controller module can be restored by setting it to the initialized status if it does not operate normally by the following reasons.

- Execution of a user program registered to a script file
- Unknown IP address or login user set to the C Controller module

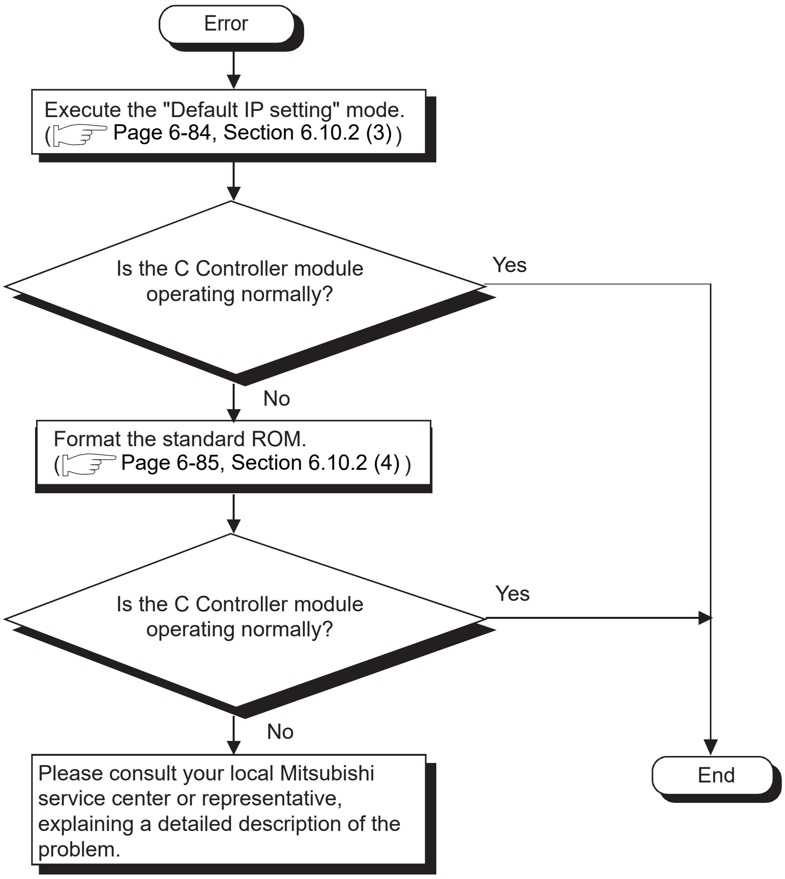

Error
Execute the "Default IP setting" mode.
(☞ Page 6-84, Section 6.10.2 (3))
Is the C Controller module operating normally?
Yes
No
Format the standard ROM.
(☞ Page 6-85, Section 6.10.2 (4))
Is the C Controller module operating normally?
Yes
No
Please consult your local Mitsubishi service center or representative, explaining a detailed description of the problem.
End

Figure 6.53 Operating procedure

(2) Settings and file conditions after the operations

The following table indicates the settings and file conditions after executing the "Default IP setting" mode or formatting the standard ROM.

Table 6.22 File conditions

File type	Settings and file conditions	
	Default IP setting	Standard ROM formatting
IP address setting	*1	*1
Event history file	—	*2
C Controller module parameters	*3	Deleted
CC-Link module parameters	*3	Deleted
MELSECNET/H module parameters	*3	Deleted
CC-Link IE controller network module parameters	*3	Deleted
Intelligent function module parameters	*3	Deleted
Script file	*4	Deleted
User program	*3	Deleted
Files used in the user program	*3	Deleted

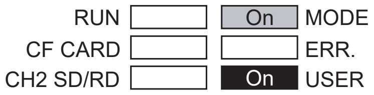
* 1 The setting returns to the default IP address (192.168.3.3).
* 2 Previous information is deleted.
* 3 Previous data before execution of the default IP setting mode are retained.
* 4 The startup script file "STARTUP.CMD" in the standard ROM is deleted. The "STARTUP.CMD" saved in the standard ROM before execution of the default IP setting mode is renamed to "STARTUP.BAK" and saved to the standard ROM. (If "STARTUP.BAK" already exists, it is overwritten.)

(3) Default IP setting mode

To set the IP address back to default, start the C Controller module in the hardware self-diagnostic mode (☞ Page 16-84, Section 16.5.2), and then execute the "Default IP setting" mode.

1) Start up the C Controller module in the hardware self-diagnostic mode (with the RUN/STOP/MODE switch set to MODE).

2) Repeatedly move the RESET/SELECT switch to the SELECT side until the USER LED turns on.
Check that the USER LED is turned on and the current mode is "Default IP setting".



(For the Q06CCPU-V)

3) Set the RUN/STOP/MODE switch to RUN to execute the "Default IP setting" mode.
(The USER LED starts flashing, and it turns off when the "Default IP setting" mode is activated.)

4) In the "Default IP setting" mode, reset or restart the C Controller module.
After resetting or restarting the C Controller module, the module will be in the following status.
   - IP address: Default IP address (192.168.3.3)
   - Script file: None

5) Access the C Controller module from C Controller setting utility by using the default IP address (192.168.3.3) and write default parameters.
Before writing parameters, open the <<Online operation>> tab and select "Clear all parameter before writing" in the "Write parameters" area.
Resetting or restarting the C Controller module sets the module into the factory-set status.

☒POINT

1. When the "Default IP setting" mode is executed with a CompactFlash card inserted in the C Controller module, the script file "STARTUP.CMD" in the CompactFlash card will not be deleted.
To set the script file in the CompactFlash card back to its original status, edit it on a personal computer.
2. If the C Controller module does not operate normally even after execution of the "Default IP setting" mode, format the standard ROM.
(☞ Page 6-85, Section 6.10.2 (4))

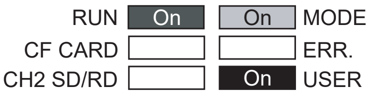
(4) Formatting the standard ROM

The following describes how to format the standard ROM and operations after formatting.
Each file in the standard ROM will be deleted after formatting the standard ROM.
Before operating the procedure provided below, back up necessary data, user programs, and parameter files. (☞ Page 8-7, Section 8.2.1)

(a) Formatting the standard ROM

1) By switch operation
Start up the C Controller module in the hardware self-diagnostic mode (☞ Page 16-84, Section 16.5.2), and perform the "Standard ROM format setting" procedure.

- Start up the C Controller module in the hardware self-diagnostic mode (with the RUN/STOP/MODE switch set to MODE).
- Repeatedly move the RESET/SELECT switch to the SELECT side until both the RUN and USER LEDs turn on.
Check that both the RUN and USER LEDs have turned on and the current mode is "Standard ROM format setting".



(For the Q06CCPU-V)

- Set the RUN/STOP/MODE switch to RUN to execute the "Standard ROM format setting" mode. (Both the RUN and USER LEDs start flashing. When the "Standard ROM format setting" mode is activated, it turns off.)
- After the RUN and USER LEDs turned off, restart the C Controller module.
- The standard ROM will be formatted after the C Controller module is restarted. (The RUN and USER LEDs start flashing.)
- When the RUN and USER LEDs stopped flashing and the MODE LED is flashing green, the standard ROM formatting is completed.
If the formatting of the standard ROM is terminated in error, each LED will be as follows. (Redo the above procedures.)
RUN LED: On
USER LED: On
ERR. LED: Flashing

Remark

Do not power off or reset the C Controller module while the standard ROM is being formatted.
Doing so will cause the standard ROM to be formatted again when the C Controller module is started up at the next time.

2) By creating a script file in the CompactFlash card

- Write the following script file "STARTUP.CMD" to the CompactFlash card, insert it into the C Controller module, and then start the C Controller module.
  (The RUN LED flashes during processing the script file "STARTUP.CMD".)

<Sample script file for standard ROM formatting[*1]>

```
STARTUP.CMD
  diskFormat("/ROM")
  userShutdownRom
```

[* 1] The sample script file for standard ROM formatting is registered to the development environment (personal computer) after installation of SW3PVC-CCPU.

- When the RUN LED stopped flashing and the MODE LED is flashing green, the standard ROM formatting and shutdown processing is completed.
  Power off the C Controller module.
  If the formatting of the standard ROM is terminated in error, the ERR. LED turns on. (Redo the above procedures if terminated in error.)
- Power off the C Controller module and then remove the CompactFlash card.

IMPORTANT

1. To format the standard ROM, perform the method provided in (4) (a) in this section.
   Do not format it by using a command from Tornado Shell.
2. Do not power off or reset the C Controller module while the standard ROM is being formatted.
   Doing so will cause the standard ROM to be formatted again when the C Controller module is started at the next time.

(b) After formatting the standard ROM

Files such as parameters and user programs are deleted. (☞ Page 6-83, Section 6.10.2 (2))
(After formatting the standard ROM, resetting the module causes a parameter error (no parameters) and the ERR. LED flashes.)
Access the C Controller module from C Controller setting utility by using the default IP address of CH1 (192.168.3.3) and write default parameters (or any given parameters).
Reset the C Controller module after writing parameters and setting the IP address.

Remark

For script files and sample programs, refer to the following.

☞ C Controller Module User's Manual (Utility Operation, Programming)

6.11 Login User Setting and Restriction

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	○

File access by FTP and operations from each utility can be restricted by setting a user name and password and starting it by the authorized login user.

POINT

The login user setting and restriction are one of the methods for preventing illegal access (such as program or data corruption) from an external device. However, this function does not prevent illegal access completely. Incorporate measures other than this function if the C Controller system's safety must be maintained against illegal access from an external device. Mitsubishi Electric Corporation cannot be held responsible for any system problems that may occur from illegal access.
Examples of measures for illegal access are shown below.

- Install a firewall.
- Install a personal computer as a relay station, and control the relay of send/receive data with an application program.
- Install an external device for which the access rights can be controlled as a relay station. (For details on the external devices for which access rights can be controlled, consult the network provider or equipment dealer.)

6.11.1 Functions that can be restricted by login user setting

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	○

The C Controller module can restrict the functions listed below.
For details, refer to each reference section shown in the table.

Table 6.23 Functions that can be restricted by login user setting

Function		Access restriction		Reference section
		When logged in	When not logged in	
FTP	/RAM (standard RAM)	Read/write enabled	Access disabled	*1
	/ROM (standard ROM)	Read/write enabled	Access disabled	
	/CF (CompactFlash card)	Read/write enabled	Access disabled	
	/SYSTEMROM (system drive)	Read only enabled	Access disabled	
Operation from C Controller setting utility		Not restricted	Parameter writing from <<Online operation>> tab is disabled.	*1
Operation from CC-Link utility		Not restricted		*1
Operation from MELSECNET/H utility		Not restricted		*1
Operation from CC IE Control utility		Not restricted		*1
Operation from Device monitoring utility		Not restricted		—
Telnet function		Not restricted	Access disabled	Page 4-48, Section 4.15

* 1 ☞ C Controller Module User's Manual (Utility Operation, Programming)

6.11.2 Setting a login user

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	○

This section describes how to set a login user and precautions.

(1) Default login user and setting method

(a) Default login user

The default login user of the C Controller module is as follows.

Table 6.24 Default user name and password

Item	Value
User name	"target"
Password	"password"

(b) Setting a login user

To add a login user account other than the default, or to delete or change any existing account, use the functions offered by the loginLib library in a script file or user program.

For details of the functions, refer to the following.

☞ Manual for VxWorks

Table 6.25 Login user setting operation list

Login user operation	Function name
Password encryption	loginDefaultEncrypt function*1 or vxencrypt command
Addition	loginUserAdd function*1
Deletion	loginUserDelete function*1
Password change	After "Deletion", execute "Addition" with a new password.
Login user list display	loginUserShow function
Login user verification	loginUserVerify function*1

* 1 When "\" is contained in the user name or password, describe "\" as "\\".

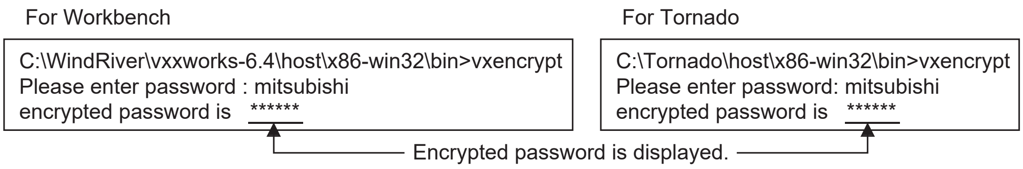
Example: To delete an account of user name "tar\get" and password "pass\word", describe the function as follows:

loginUserDelete("tar\\get", "pass\\word");

<Program example>

When registering "qcpu" as a user name and "mitsubishi" as a password:

1) Encrypt the password with the loginDefaultEncrypt function or the vxencrypt command*1. When using the vxencrypt command, start it from the command prompt of the development environment (personal computer). *2



For Workbench

```
C:\WindRiver\vxxworks-6.4\host\x86-win32\bin>vxencrypt
Please enter password : mitsubishi
encrypted password is  ******
```

For Tornado

```
C:\Tornado\host\x86-win32\bin>vxencrypt
Please enter password: mitsubishi
encrypted password is  ******
```

Encrypted password is displayed.

*1 Refer to the following manual for details of the vxencrypt command.

☞ VxWorks Manual

*2 The message shown below may be displayed during execution of the vxencrypt command. Since it does not mean any problem with the encryption, ignore the message.

<Displayed message>

'stty' is not recognized as an internal or external command, operable program or batch file.

2) Register the user on VxWorks with the loginUserAdd function.
Set it in the script file or user program.

```
loginUserAdd ("qcpu" , "******");
```

Type the encrypted password.

(2) Precautions

(a) Application of the login user setting of the C Controller module
The login user setting of the C Controller module restricts users and module setting operations.
This setting does not completely prevent unauthorized access form outside.

(b) Ensuring safety from illegal access from outside
To ensure the C Controller system safety against unauthorized access from outside, take appropriate measures.
Take into account the following.

1) To prevent unauthorized access from outside, use the module on a LAN.

2) Set an account (user name and password) to prevent it from being leaked.
   - Avoid a simple setting composed of alphanumeric characters only.
   - Include some symbols ($&?) to make the user name and password complicated.
   - Delete the default account by using the loginUserDelete function to prevent unauthorized access by the default account.

<Program example>
Delete the default account by the loginUserDelete function on VxWorks.
Set it in the script file or user program.

```
loginUserDelete ("target", "password");
```

(c) Characters applicable to the user name and password
Refer to the following.
☞ Appendix 5
Functions may not be restricted properly (☞ Page 6-88, Section 6.11.1) if any characters other than those indicated in Appendix 5 are used.

(d) Retaining the login user setting
The set login user information are cleared and returned to default when the C Controller module is powered off or reset.
To retain the login user setting, describe a script file that registers (adds/deletes) the login user setting.

Describe either of the following in the script file.

- Directly describe the login user operation commands (loginUserAdd function and/or loginUserDelete function).
- Describe the procedure for starting the user program task for login user operation.

6.12 Maintenance and Inspection

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	○

⚠WARNING

- ● Do not touch any terminal while power is on.
  Doing so will cause electric shock.
- ● Correctly connect the battery connector.
  Do not charge, disassemble, heat, short-circuit, solder, or throw the battery into the fire.
  Doing so will cause the battery to produce heat, explode, or ignite, resulting in injury and fire.
- ● Shut off the external power supply for the system in all phases before cleaning the module or retightening the terminal screws or module fixing screws.
  Undertightening the terminal screws can cause short circuit or malfunction.
  Overtightening can damage the screw and/or module, resulting in drop, short circuit, or malfunction.

⚠CAUTION

- ● Before performing online operations (especially program modification, forced output, and operation status change), read relevant manuals carefully and ensure the safety.
  Improper operation may damage machines or cause accidents.
- ● Do not disassemble or modify the modules.
  Doing so may cause failure, malfunction, injury, or a fire.
- ● Use any radio communication device such as a cellular phone or PHS (Personal Handy-phone System) more than 25cm (9.85 inches) away in all directions from the C Controller system.
  Failure to do so may cause malfunction.
- ● Shut off the external power supply for the system in all phases before mounting or removing the module.
  Failure to do so may cause the module to fail or malfunction.
- ● After the first use of the product, do not mount/remove the module to/from the base unit more than 50 times respectively (IEC 61131-2 compliant).
  Exceeding the limit of 50 times may cause malfunction.
- ● Do not drop or apply shock to the battery to be installed in the module.
  Doing so may damage the battery, causing the battery fluid to leak inside the battery.
  If the battery is dropped or any shock is applied to it, dispose of it without using.
- ● Before handling the module, touch a grounded metal object to discharge the static electricity from the human body.
  Failure to do so may cause the module to fail or malfunction.

This section explains the items to be inspected daily or periodically to use the C Controller system under normal and optimum conditions at all times.

6.12.1 Daily inspection

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	○

The following table lists the items to be inspected daily.

Table 6.26 Daily inspection

No.	Item		Inspection	Criterion	Action
1	Installation of base unit		Check the fixing screws and cover for looseness.	The base unit is securely installed.	Retighten the screws.
2	Installation of I/O modules		Check the module and module fixing hook for looseness.	The module fixing hooks are securely engaged and the modules are securely mounted.	Securely engage the module fixing hooks.
3	Connection		Check the terminal screws for looseness.	No looseness.	Retighten the terminal screws.
			Check the distance between solderless terminals.	Proper clearance between solderless terminals.	Provide proper clearance between solderless terminals
			Check the connector part of the cable.	Connectors are securely connected.	Retighten the connector fixing screws.
4	Indicator LEDs		—		
		Power supply module POWER LED	Check that the LED is lit.	Lit green. (Faulty if the LED is off)	☞ Page 16-4, Section 16.2.1
		C Controller module MODE LED	Check that the LED is lit. in the RUN status.	Lit*1 (Faulty if the LED is off)	☞ Page 16-5, Section 16.2.2
		C Controller module ERR. LED	Check that the LED is off.	Off (Faulty if the LED is off or flashing)	☞ Page 16-7, Section 16.2.3
		C Controller module 7-segment LED (Q12DCCPU-V only)	Check for on/off of the LED.	On/off reactions of the 7-segment LED are correct. ☞ Page 16-83, Section 16.5.1 (5) (c)	Please consult your local Mitsubishi service center or representative, explaining the details of the problem.
		Input module Input LED	Check for on/off of the LED.	Lit when the input is on. Off when the input is off. (Faulty if the reaction is other than above.)	☞ Manual for the CPU module
		Output module Output LED	Check for on/off of the LED.	Lit when the output is on. Off when the output is off. (Faulty if the reaction is other than above.)	☞ Page 16-20, Section 16.2.11

[1] The MODE LED flashes to notify that the C Controller has completed its shutdown. (Flashing does not indicate an error.)

6.12.2 Periodical inspection

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	○

This section explains the items to be inspected once or twice every six months or every year.
Note that these inspections are required when the system is moved or modified including re-wiring.

Table 6.27 Periodic inspection

No.	Item		Inspection	Criterion	Action
1	Ambient environment		—		
		Ambient temperature	Measure with a thermometer and a hygrometer. Measure corrosive gas.	0 to 55℃	When the module is in the panel, the temperature in the panel is the ambient temperature.
		Ambient humidity		5 to 95%RH[*1]	
		Ambience		No corrosive gas	
2	Power voltage		Measure a voltage across the terminals of 100/200V AC and 24V DC.	85 to 132V AC	Change the power supply.
				170 to 264V AC	
				15.6 to 31.2V DC	
3	Installation		—		
		Looseness, rattling	Move the module to check for looseness and rattling.	The module is securely installed.	Retighten the screws. For the C Controller module, I/O modules, or power supply module, secure the module with screws.
		Dirt, foreign matter	Visual check.	No dirt or foreign matter.	Remove and clean it.
4	Connection		—		
		Looseness of terminal screws	Further tighten with a screwdriver.	No looseness.	Retighten the terminal screws.
		Distance between solderless terminals.	Visual check.	Proper clearance between solderless terminals.	Provide proper clearance between solderless terminals.
		Looseness of connectors.	Visual check.	No looseness.	Retighten the connector fixing screws.
5	Battery		Check the "Battery error" is off in the <<Module information>> tab.	(Preventive maintenance)	Replace the battery if the service life has expired regardless of the indication.

[*1] 10 to 90%RH when an AnS series module is used in a multiple CPU system.

CHAPTER 7 I/O NUMBER ASSIGNMENT

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	○

This chapter explains the I/O number assignment that is required for data communications between the C Controller module and I/O modules and intelligent function modules.

7.1 Number of Base Units and Number of Slots

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	○

The C Controller module can be used in the following system configurations.

- One main base unit and extension base unit(s)
- One slim type main base unit

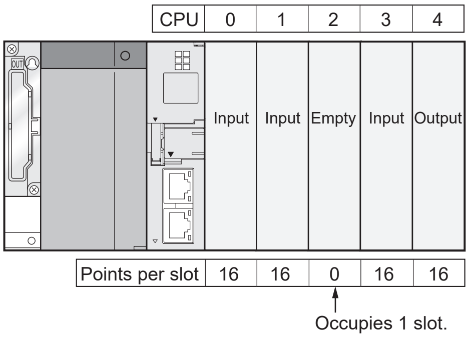
(1) Available configuration and number of slots

For details, refer to t he following.

☞ Page 2-9, Section 2.1.4

POINT

The number of available slots (modules) includes empty slots.
(Even if Slot 2 is set to "Empty, 0 point" as shown below, it is occupied as one slot.)



CPU	0	1	2	3	4
	Input	Input	Empty	Input	Output
Points per slot	16	16	0	16	16

(2) Precautions on the number of mounted modules

The number of mounted modules must be within the range of available slots.

Even if the total number of slots on the main and extension base units is greater than the number of available slots (for example, six 12-slot base units are used), no error will occur as long as the modules are mounted in the allowed number of slots.

If a module is mounted in a slot that is outside the range of available slots, an intelligent function module assignment error (error code: 2124, SP.UNIT LAY ERR.) will occur.

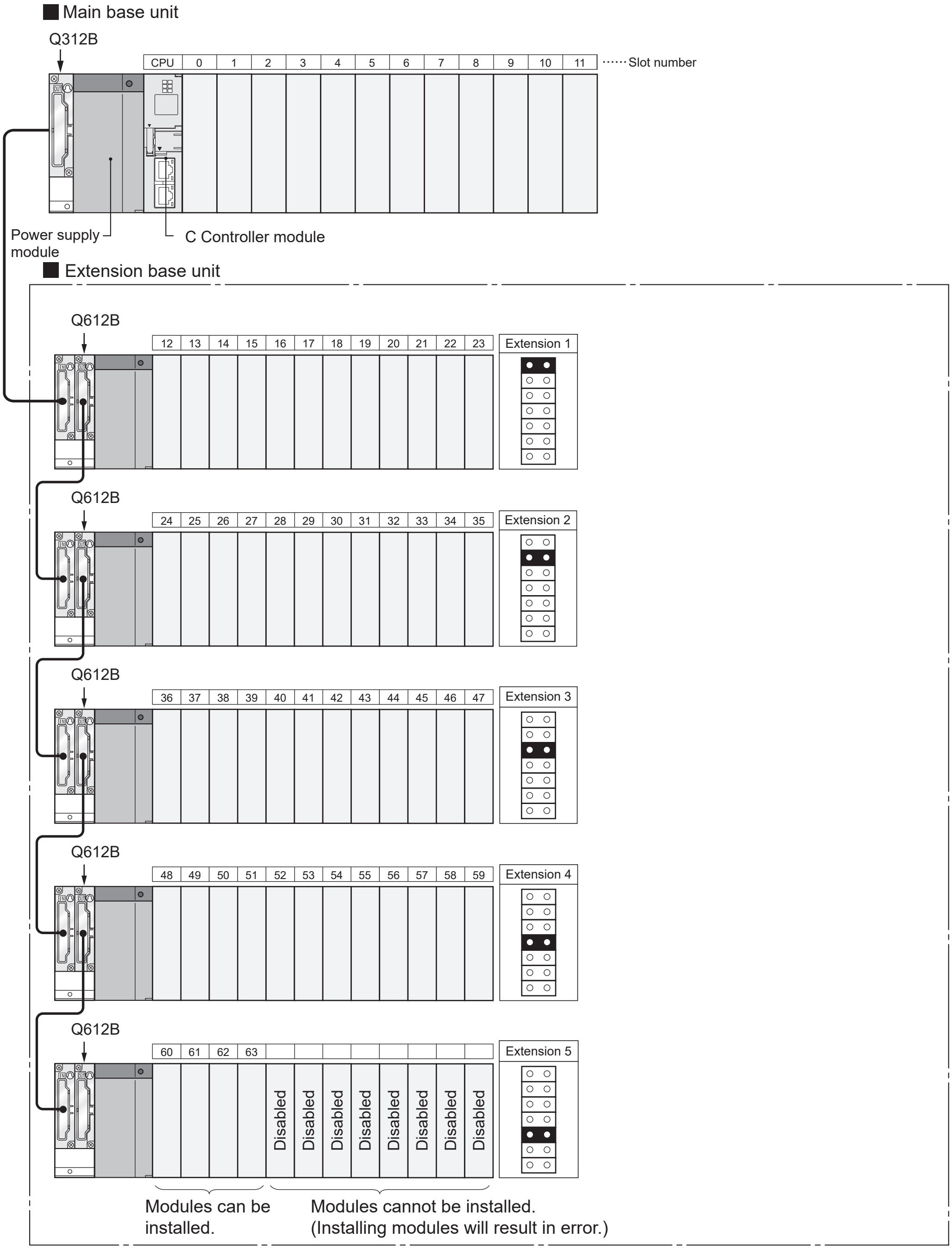


Figure 7.1 Configuration example

7.2 Connecting Extension Base Units and Setting the Number of Stages

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	○

The extension base unit models, Q5□B and Q6□B, are connectable.
The Q6□RB and QA1S6□B cannot be used.

(1) Extension stage number setting and setting order

For each extension base unit, set the extension stage number with the stage number setting connector on the unit.
The stage extension number must be set in order of connection, starting from the extension base unit connected to the main base unit.

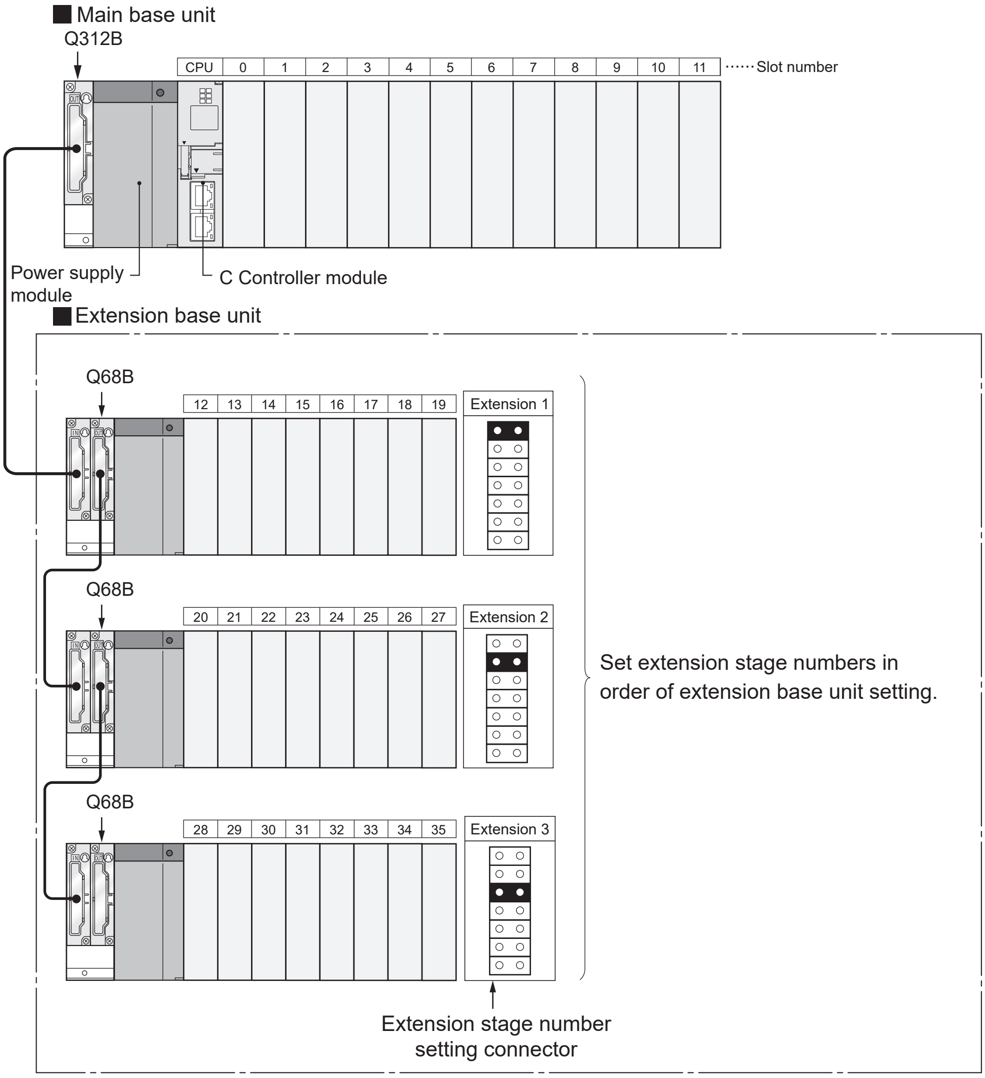


Figure 7.2 Extension stage number setting and setting order

(2) Precautions for extension stage number setting

(a) Extension stage number setting order
The extension stage numbers must be set consecutively.
Even if any extension stage number is skipped when the Auto mode is active for base unit assignment (☞Page 7-7, Section 7.3 (1)), 0 slot is assigned to the skipped stage and the number of available slots does not increase.

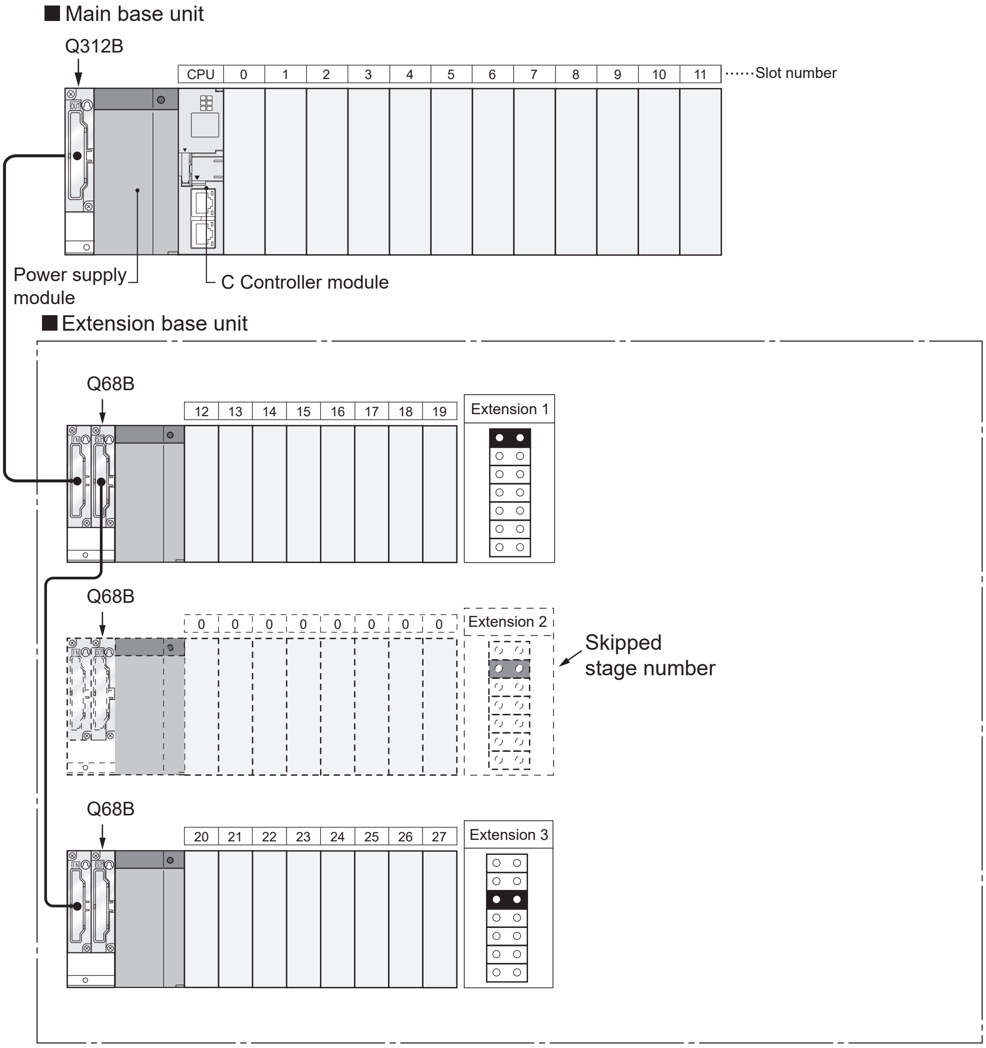


Figure 7.3 Number of slots when extension stage number is skipped

(b) Duplicated extension stage number setting

The same extension stage number cannot be assigned to more than one extension base unit.

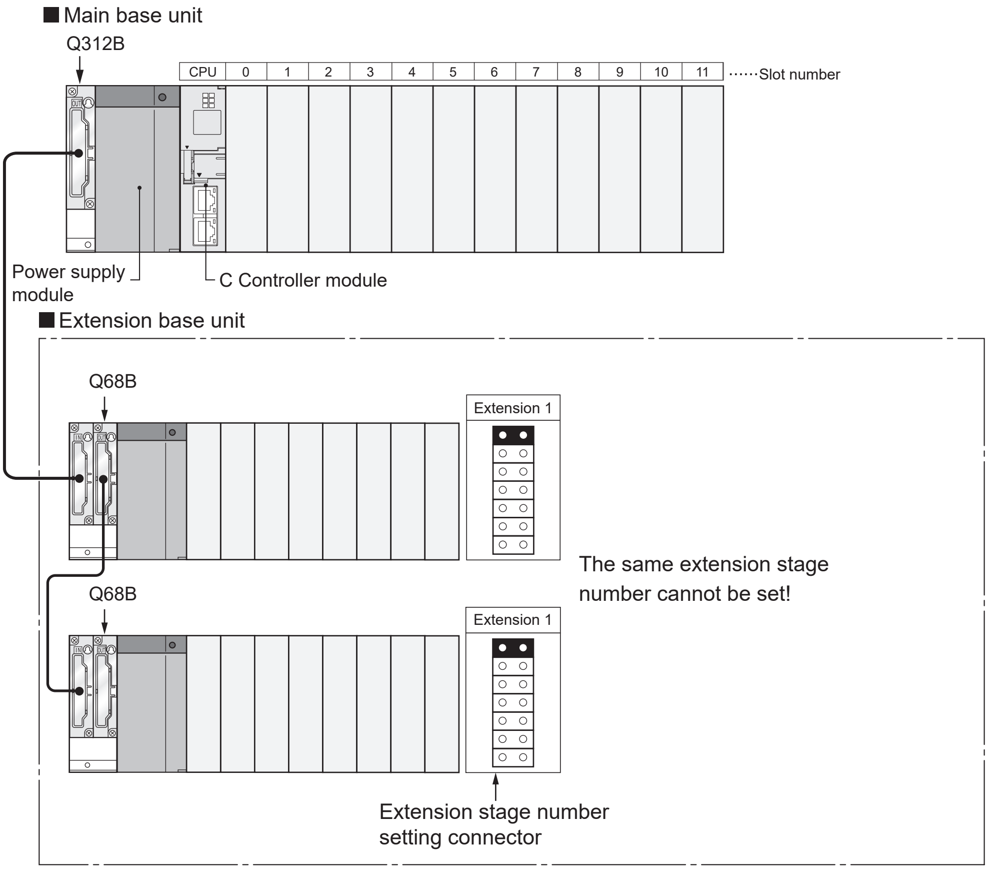


Figure 7.4 Duplicated extension stage number setting

(c) Attaching two or more connectors to different positions or attaching no connector
The system is not operable when two or more extension stage setting connectors are attached to different positions or no connector is attached.

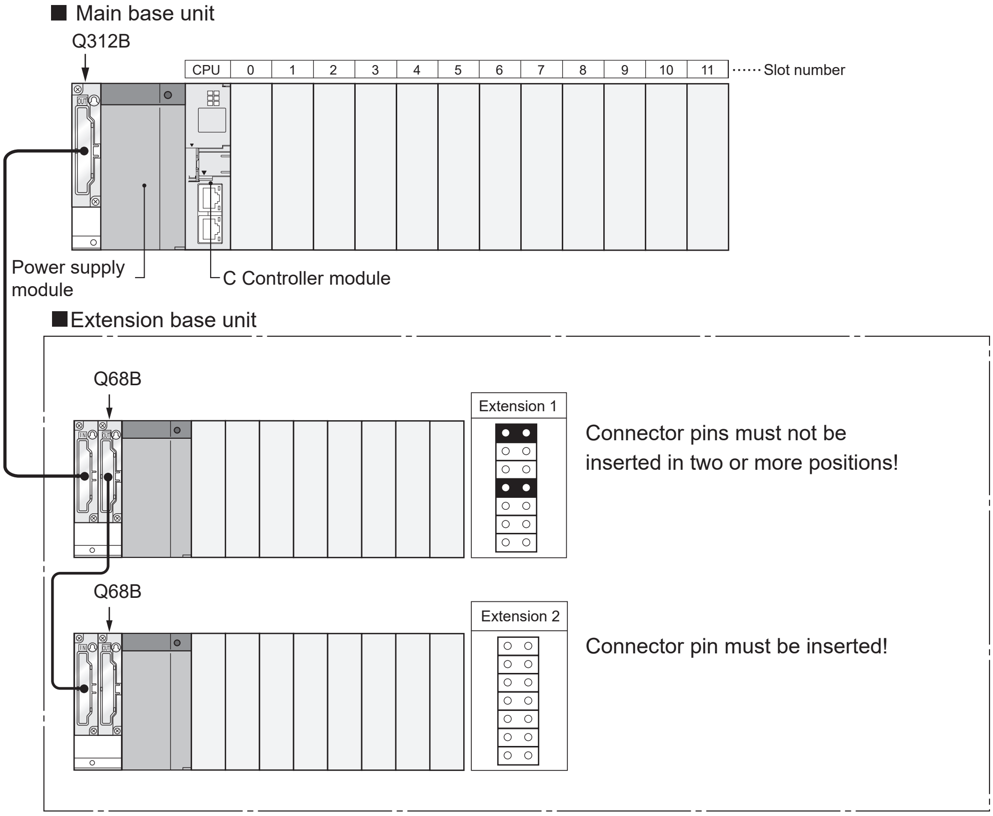


Figure 7.5 Inserting connector pins into two or more positions or no connector insertion

7.3 Base Unit Assignment (Base Mode)

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	○

The Auto and Detail modes are available for the C Controller module to assign the modules to the main base unit, slim type main base unit, and extension base units.

The base mode can be set in the <<I/O assignment settings>> tab of C Controller setting utility. (☞ (3) in this section)

(1) Auto mode

In Auto mode, the slots of the main base unit, slim type main base unit, and extension base units are assigned based on the available number of slots.
I/O numbers are assigned to the modules that can be mounted to each base unit.

(a) For a 3-slot base unit: Three slots are occupied.

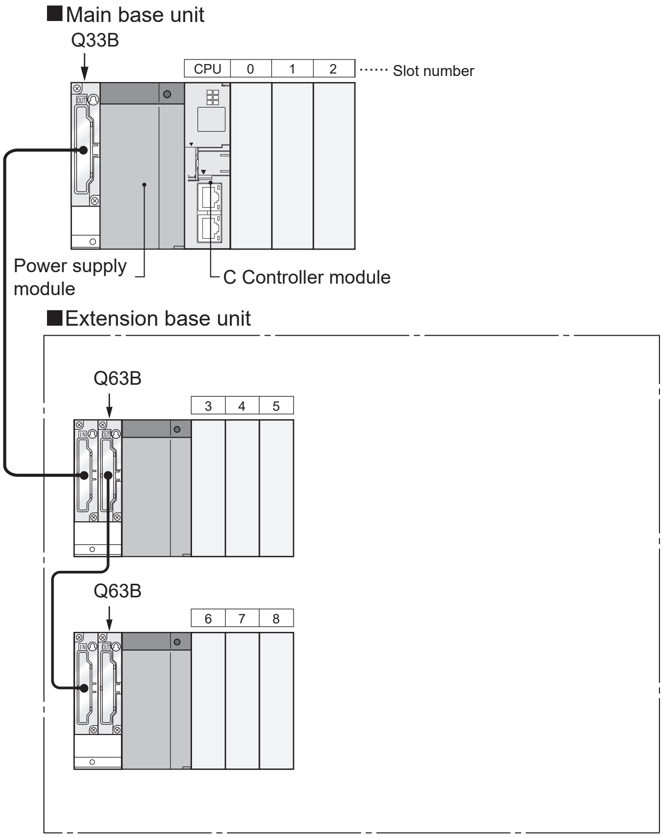


Figure 7.6 3-slot base units

(b) For a 5-slot base unit: Five slots are occupied.

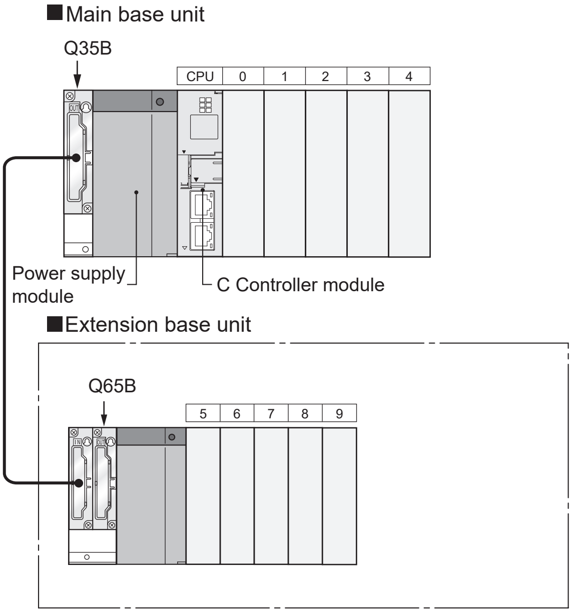


Figure 7.7 5-slot base units

(c) For a 8-slot base unit: Eight slots are occupied.

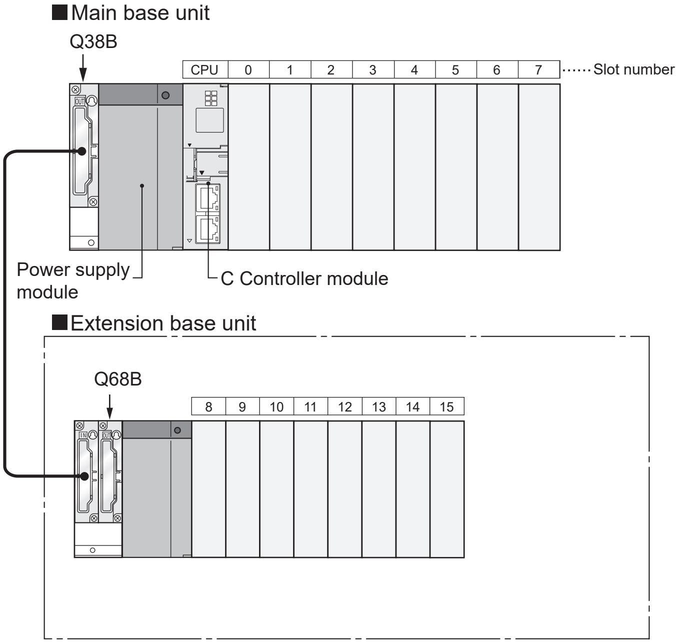


Figure 7.8 8-slot base units

(d) For a 12-slot base unit: Twelve slots are occupied.

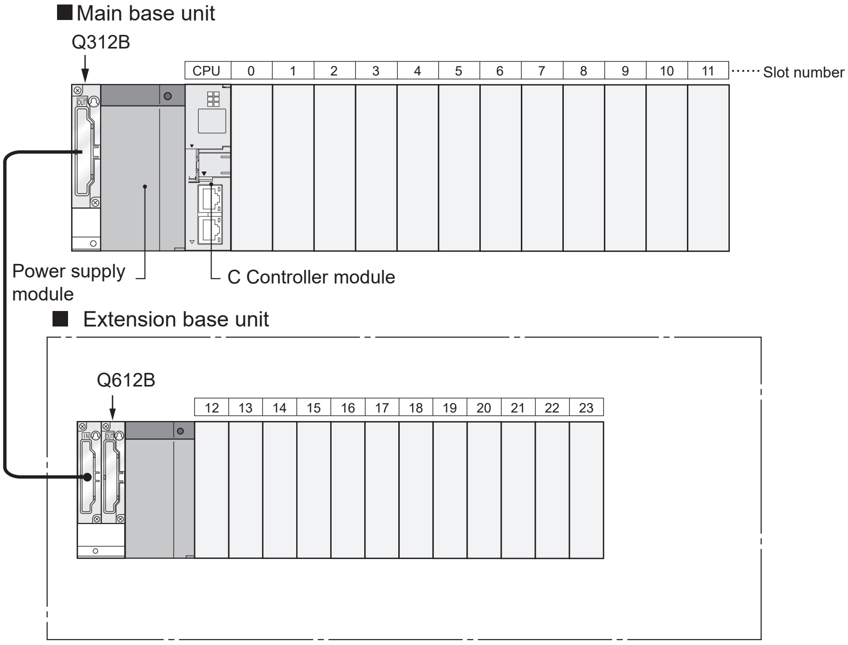


Figure 7.9 12-slot base units

(2) Detail mode

In Detail mode, the number of modules can be set for each base unit in the <<I/O assignment settings>> tab of C Controller setting utility.

(a) Application
Since an empty slot occupies one slot if 0 point is assigned to it, this mode is used to make unused slots unassigned.

(b) Precautions for setting the number of slots
Any number of slots can be set regardless of the number of modules used.

☒POINT

Set the number of slots for all base units used.
I/O assignment may not function properly if the setting is not performed for all base units.

The following describes what happens if the set number of slots is different from the actual number of slots used on a base unit.

1) When the set number of slots is greater than the number of slots used on the base unit
The slots for the set quantity are occupied.
The slot numbers after the used ones are assigned as empty slots.
For example, using five slots in the 8-slot base unit setting will leave three slots empty.

■ Main base unit

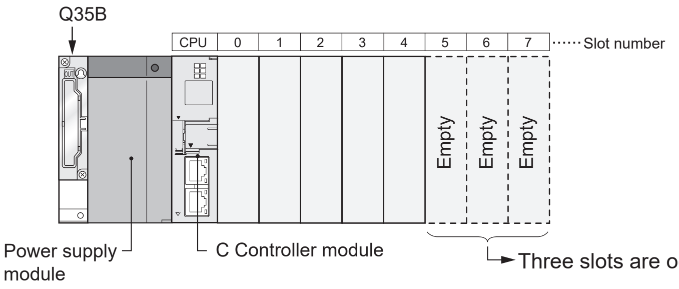


Figure 7.10 When the set number of slots is greater than the number of slots used on the base unit

The number of empty slots is the one set in the <<System settings>> or <<I/O assignment settings>> tab of C Controller setting utility.
(The default is 16 points.)

2) When the set number of slots is less than the number of slots used on the base unit
The slots after the set quantity are disabled.
For example, if the number of slots to be used is set to eight on a 12-slot base unit, the right four slots are disabled.
Mounting a module in a disabled slot causes an intelligent function module assignment error (error code: 2124, SP.UNIT LAY ERR.)

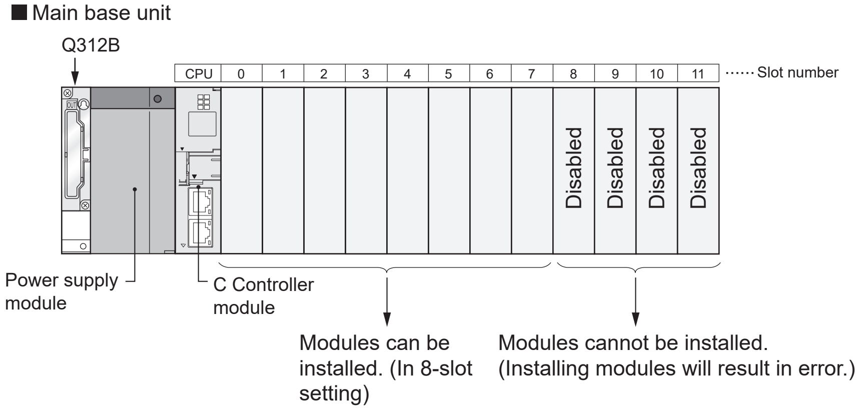
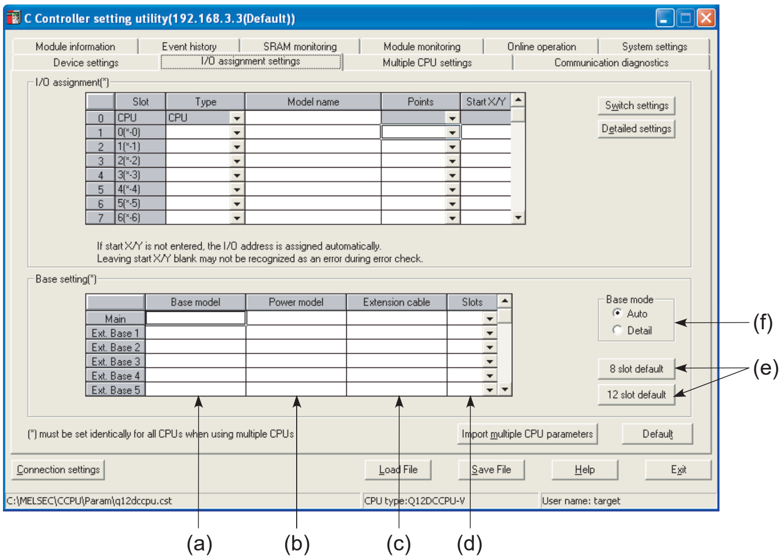


Figure 7.11 When the set number of slots is less than the number of slots used on the base unit

(3) Base mode setting in C Controller setting utility



I/O assignment(*)

	Slot	Type	Model name	Points	Start X/Y
0	CPU	CPU			
1	0(*-0)				
2	1(*-1)				
3	2(*-2)				
4	3(*-3)				
5	4(*-4)				
6	5(*-5)				
7	6(*-6)				

If start X/Y is not entered, the I/O address is assigned automatically.
Leaving start X/Y blank may not be recognized as an error during error check.

Base setting(*)

	Base model	Power model	Extension cable	Slots
Main				
Ext. Base 1				
Ext. Base 2				
Ext. Base 3				
Ext. Base 4				
Ext. Base 5				

Base mode: ● Auto ○ Detail

(*) must be set identically for all CPUs when using multiple CPUs

C:\MELSEC\CCPU\Param\q12dccpu.cst
CPU type:Q12DCCPU-V
User name: target

Figure 7.12 I/O assignment settings

(a) Base model
Enter the base unit model name within 16 characters.
The C Controller module does not recognize this setting. (Use this setting as a note.)

(b) Power model
Enter the power supply module name within 16 characters.
The C Controller module does not recognize this setting. (Use this setting as a note.)

(c) Extension cable
Enter the extension cable model name within 16 characters.
The C Controller module does not recognize this setting. (Use this setting as a note.)

(d) Slots (C Controller module uses this setting.)
Select one of the following options for the base unit.

- 2 (2 slots)
- 3 (3 slots)
- 5 (5 slots)
- 8 (8 slots)
- 10 (10 slots)
- 12 (12 slots)

(e) 8 slot default/12 slot default (C Controller module uses this setting.)
Select either option to set the same number of slots for all base units.

(f) Auto/Detail
Select the Auto or Detail mode for base unit assignment.

7.4 What is I/O Number?

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	○

I/O numbers represent addresses that are used in user programs to perform the following:

- Inputting ON/OFF data into the C Controller module
- Outputting ON/OFF data from the C Controller module to the outside

(1) Inputting and outputting ON/OFF data

Input (X) is used to input ON/OFF data to the C Controller module.
Output (Y) is used to output ON/OFF data from the C Controller module.

(2) I/O number notation

I/O numbers are expressed in hexadecimal notation.
When 16-point I/O modules are used, I/O numbers are consecutively assigned to respective slots, each of which is □□0 to □□F as shown in Figure 7.13.

The following applies to each module.

- For an input module, "X" is prefixed to the I/O number.
- For an output module, "Y" is prefixed to the I/O number.

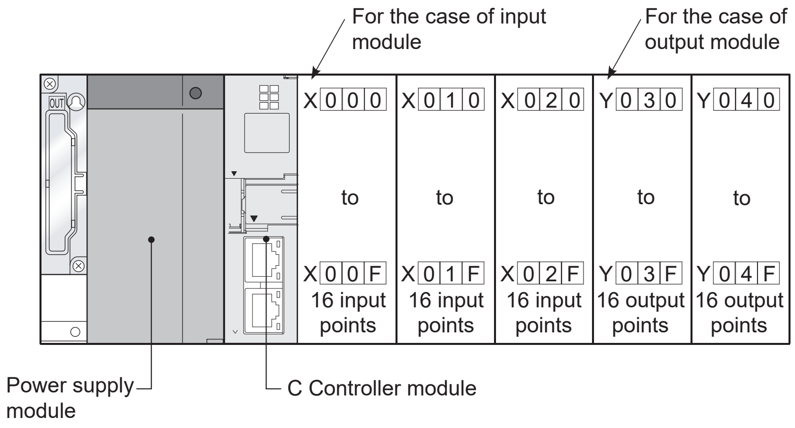


Figure 7.13 Input and output numbers

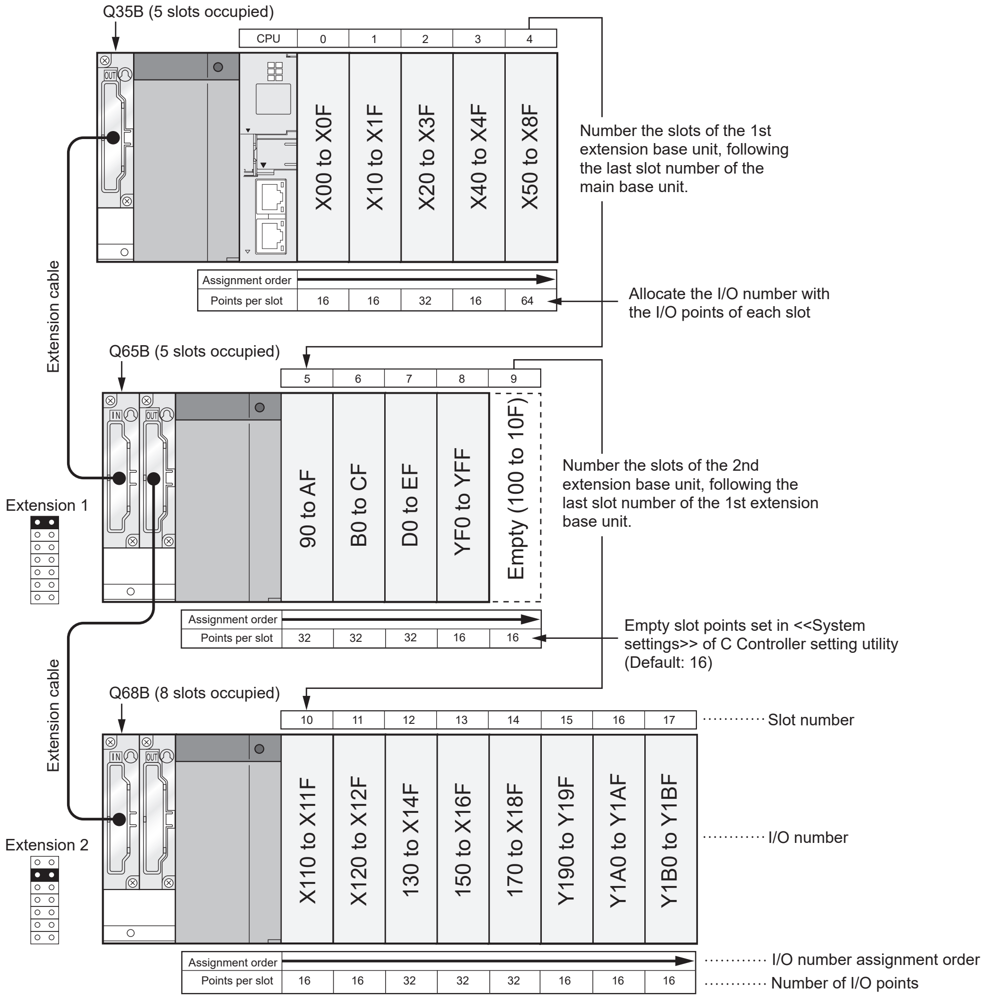
7.5 I/O Number Assignment

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	○

7.5.1 I/O numbers of a base unit

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	○

The C Controller module assigns I/O numbers when it is powered on or reset.
Figure 7.14 shows an example of I/O assignment when a base unit is set in Auto mode and no I/O assignment is manually performed.



X□□0 to □□F: Input module
Y□□0 to □□F: Output module
□□0 to □□F: Intelligent function module

Figure 7.14 I/O number assignment example

In the above example, I/O points of each intelligent function module is 32.
The number of I/O points varies by intelligent function module.
Check the number of I/O points in the manual for each intelligent function module before assigning I/O numbers.

☒POINT

The C Controller module can control the system even if I/O numbers are not assigned in C Controller setting utility.

Assign I/O numbers according to the following.

(1) Number of slots of base units

The number of slots of each base unit is set in the base mode. (☞ Page 7-7, Section 7.3)

(a) In Auto mode
The number of slots allowed for the base unit is set.
For example, five slots are set for a 5-slot base unit, and twelve slots for a 12-slot base unit.

(b) In Detail mode
The number of slots set in <<I/O assignment settings>> of C Controller utility is applied.

(2) Order of I/O number assignment

I/O numbers are assigned consecutively from left to right, starting the slot next to the C Controller module on a main base unit or slim type main base unit set to 0H.

(3) Order of I/O number assignment for extension base units

To extension base units, I/O numbers are assigned following the last I/O number of the main base unit.
Assign them consecutively from left (I/O 0) to right as shown in Figure 7.15, in order of the stage number setting connector settings of the extension base units.

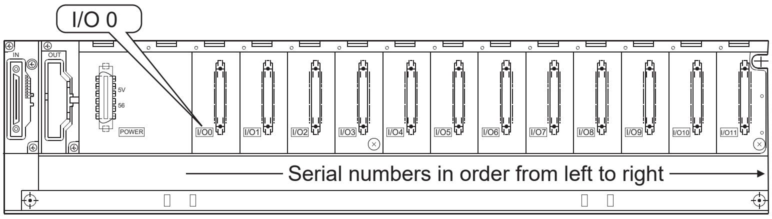


Figure 7.15 Order of I/O number assignment

(4) I/O numbers assigned to respective slots

Each slot of a base unit occupies the points equal to the I/O numbers for each I/O module or intelligent function module.
When a 32-point input module is mounted on the right of the C Controller module, the I/O numbers are X0 to X1F.

(5) I/O numbers of empty slots

To an empty slot on a base unit, where no I/O module or intelligent function module is mounted, the number of points set in the <<System settings>> tab of C Controller setting utility is assigned. (The default is 16.)

☒POINT

When base units are assigned in AUTO mode, if an extension stage number is skipped, it will not be reserved as the number for an empty base unit. (☞ Page 7-4, Section 7.2 (2) (a))
To reserve an extension stage for future expansion, perform the base unit setting in the <<I/O assignment settings>> of C Controller setting utility.

7.6 I/O Assignment by C Controller Setting Utility

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	○

This section describes I/O assignment using C Controller setting utility.

7.6.1 Purposes of I/O assignment by C Controller setting utility

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	○

Use C Controller setting utility for I/O assignment in the following cases.

(1) Reserving points for future replacement with a module other than 16-point module

When a currently used module is to be replaced with another module of any other points in the future, the points can be reserved in advance so that the I/O numbers need not be changed.

For example, a slot where a 16-point input module is currently mounted can be assigned to a 32-point input module.

(2) Preventing I/O number change in module replacement

When an I/O module or intelligent function module other than 16-point modules is removed due to a malfunction, any I/O number change can be prevented.

(3) Changing I/O numbers to those used in the user program

When I/O numbers used in a user program are different from those in the actual system, I/O numbers of each module can be changed to those in the user program.

(4) Setting input response time (I/O response time)

The input response time of an input module, I/O combined module, or interrupt module can be changed to the time suitable to the system. (☞ Page 4-34, Section 4.8)

(5) Setting output mode for error

A desired output mode can be set to output modules, I/O combined modules, and interrupt modules for the time when an error occurs. (☞ Page 4-36, Section 4.9)

(6) Setting switches of intelligent function modules and interrupt modules

The switches of intelligent function modules and interrupt modules can be set. (☞ Page 4-38, Section 4.11)

(7) Setting the output status for a C Controller module error

The output status (hold/clear) of output modules and intelligent function modules can be set for the time when the C Controller module stops its operation due to a stop error. (☞ Page 4-36, Section 4.9)

(8) Setting C Controller module operation for an intelligent function module hardware error

The operation status (continue/stop) of the C Controller module can be set for the time when a hardware error occurs in an intelligent function module. (☞ Page 4-37, Section 4.10)

☒POINT

1. I/O assignment settings take effect when either of the following operations is performed after parameter writing in C Controller setting utility.
   - Restarting the C Controller system (ON→OFF→ON)
   - Resetting the C Controller module
2. I/O assignment settings are required for changing input module response time and setting intelligent function module switches.
3. If an I/O module other than 16-point modules fails with I/O assignment settings not having been configured in C Controller setting utility, I/O numbers of the modules after the faulty module may change and it can result in malfunction.
   It is recommended to configure I/O assignment settings in C Controller setting utility.

7.6.2 Details of I/O assignment by C Controller setting utility

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	○

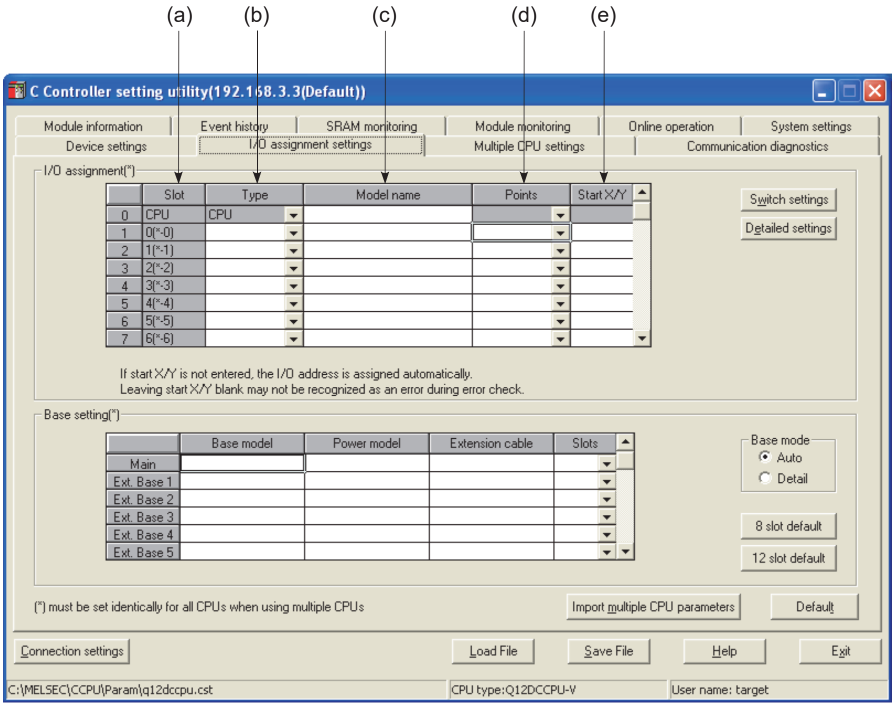
The "Type" (module type), "Points" (I/O points), and "Start X/Y" (start I/O number) can be assigned for each slot of base units.

For example, to change the I/O points of a specific slot, only the respective point value can be changed.

To the items that do not need setting, the mounting status of the base unit is applied.

(1) I/O assignment setting

Set I/O assignment in the <<I/O assignment settings>> tab of C Controller setting utility.



C Controller setting utility(192.168.3.3(Default))

I/O assignment(*)

	Slot	Type	Model name	Points	Start X/Y
0	CPU	CPU			
1	0(*-0)				
2	1(*-1)				
3	2(*-2)				
4	3(*-3)				
5	4(*-4)				
6	5(*-5)				
7	6(*-6)				

If start X/Y is not entered, the I/O address is assigned automatically.
Leaving start X/Y blank may not be recognized as an error during error check.

Base setting(*)

	Base model	Power model	Extension cable	Slots
Main				
Ext. Base 1				
Ext. Base 2				
Ext. Base 3				
Ext. Base 4				
Ext. Base 5				

Base mode
● Auto
○ Detail

(*) must be set identically for all CPUs when using multiple CPUs

C:\MELSEC\CCPU\Param\q12dccpu.cst | CPU type:Q12DCCPU-V | User name: target

Figure 7.16 I/O assignment settings

(a) Slot

Slot No. and what number of the slot on what number of the stage are displayed for each slot.

When a base unit is set into Auto mode, "*" indicates the base unit stage number, and the slot number counted from Slot 0 of the main base unit is displayed.

(b) Type (C Controller module uses this setting.)
Select the type of each module from the following.

- Empty (Empty slot)
- Input (Input module)
- Hi Input (High-speed input module)
- Output (Output module)
- I/O Mix (I/O combined module)
- Intelli. (Intelligent function module)
- Interrupt (Interrupt module)

If no type is specified, the type of the actually mounted module is used for the slot.

(c) Model name
Enter the module model name within 16 characters.
The C Controller module does not recognize this setting. (Use this setting as a note.)

(d) Points (C Controller module uses this setting.)
To change the number of I/O points for each slot, select one from the following.

- 0 (0 points)
- 48 (48 points)
- 256 (256 points)
- 16 (16 points)
- 64 (64 points)
- 512 (512 points)
- 32 (32 points)
- 128 (128 points)
- 1024 (1024 points)

To the slot with no point setting, the points of the actually mounted module is assigned.

(e) Start X/Y (C Controller module uses this setting.)
When changing I/O numbers of each slot, enter a new start I/O number.
To the slot with no start X/Y setting, I/O numbers following those of the configured slot are assigned in order.

(2) Precautions for I/O assignment

(a) I/O-assigned slot status

When I/O assignment is set for a slot, the setting has priority over the actual module-mounting status.

1) When the set number of points is less than the points of the actual I/O module

The I/O points for the actual module are decreased.

For example, if a slot in which a 32-point input module is mounted is set as a 16-point input module in I/O assignment, 16 points of the last half cannot be used.

2) When the set number of points is less than the points of an actual intelligent function module

An intelligent function module assignment error (error code: 2100, SP.UNIT LAY ERR.) occurs.

3) When the set number of points is greater than the points of the actual I/O module

The excess points are not used by the I/O module.

4) Consistency between the actual and assigned module types

The type of the module actually mounted must be the same with the type set in the I/O assignment settings.

A difference between them will cause malfunction.

For intelligent function modules, the I/O points must be also consistent.

Table 7.1 lists the operational behaviors that occur when the type of the mounted module is different from the type set in the I/O assignment settings.

Table 7.1 Operation results of inconsistency between the actual and assigned module types

Actual module	I/O assignment setting	Result
Input module High-speed input module Output module I/O combined module	Intelli./Interrupt	Intelligent function module assignment error (error code: 2100, SP.UNIT LAY ERR.)
Input module High-speed input module I/O combined module	Output	Input/High-speed input/I/O combined (The actual module type takes precedence.)
Output module I/O combined module	Input/Hi. Input	Output/I/O combined (The actual module type takes precedence.)
Intelligent function module Interrupt module	Input/Hi. Input Output/I/O mix	Intelligent function module assignment error (error code: 2100, SP.UNIT LAY ERR.)
Empty slot	Input/Hi. Input Output/I/O mix Intelli./Interrupt	Handled as an empty slot. (All points are off.)
All modules	Empty	Handled as empty slots. (All points are off.)
Other combinations		No errors, but malfunctioning

5) Last I/O number

In the I/O assignment settings, the last I/O number must be FFFH or lower.
If it exceeds FFFH, an intelligent function module assignment error (error code: 2124, SP.UNIT LAY ERR.) will occur.

(b) Precautions on automatic assignment of start X/Y

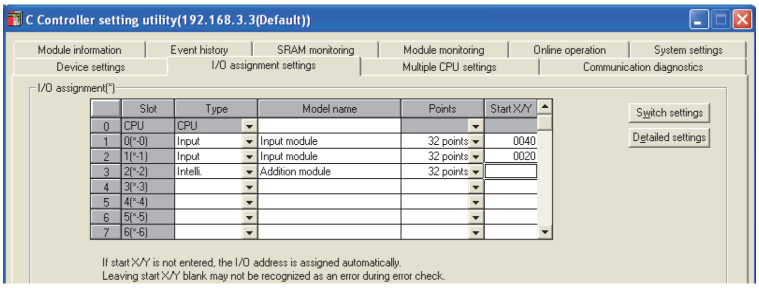
When a start X/Y has not been entered, the C Controller module automatically assigns it.

Because of this, the start X/Y setting of a slot may overlap with the one assigned by the C Controller module in the case of 1) or 2) below.

1) In the Start X/Y column, I/O numbers set in two contiguous fields are not in ascending order.

2) Some slots have start X/Y settings, some do not (automatically assigned).

Figure 7.17 shows an example of overlapped start X/Y.



C Controller setting utility(192.168.3.3(Default))

I/O assignment(*)

	Slot	Type	Model name	Points	Start X/Y
0	CPU	CPU			
1	0(*-0)	Input	Input module	32 points	0040
2	1(*-1)	Input	Input module	32 points	0020
3	2(*-2)	Intelli.	Addition module	32 points	
4	3(*-3)				
5	4(*-4)				
6	5(*-5)				
7	6(*-6)				

If start X/Y is not entered, the I/O address is assigned automatically.
Leaving start X/Y blank may not be recognized as an error during error check.

Figure 7.17 I/O assignment settings with start X/Y overlapped

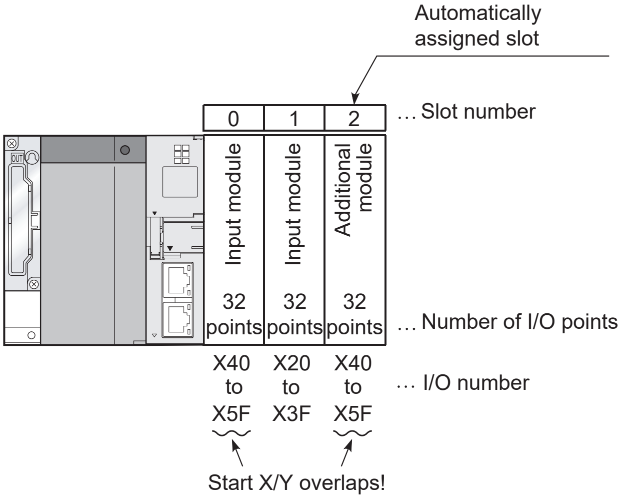


Figure 7.18 Start X/Y values corresponding to the above I/O assignment settings

Be careful not to have the start X/Y of each slot overlapped.

Mounting the slot for duplication of start XY to modules will result in "SP. UNIT LAY ERR."

7.7 I/O Assignment Example

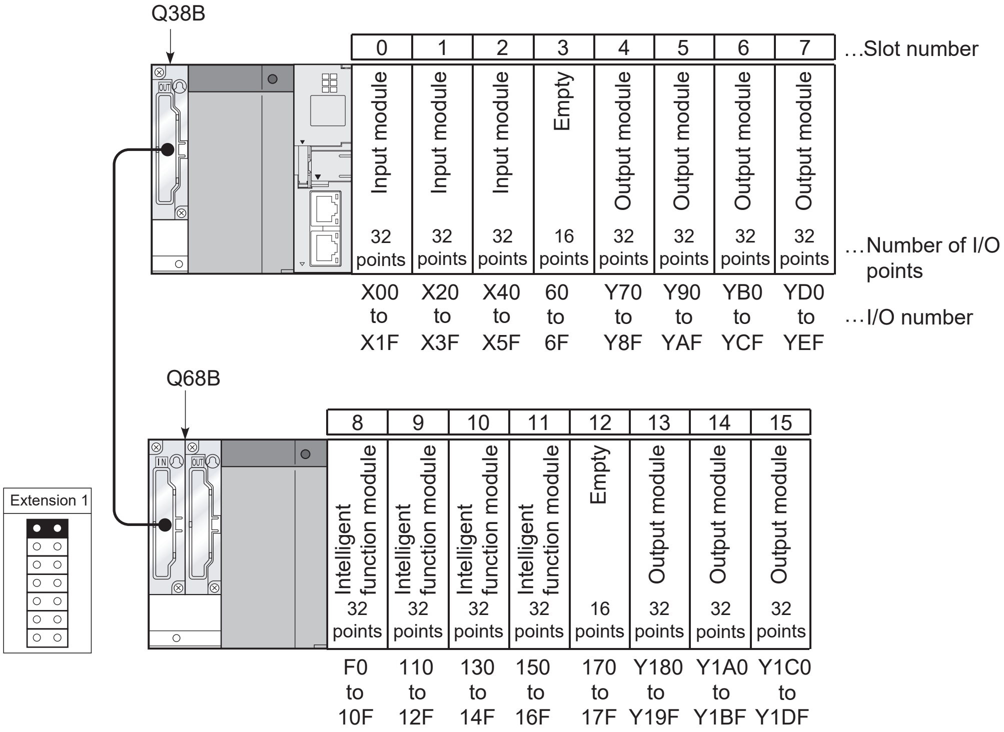
Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	○

This section provides an example of I/O number assignment by using C Controller setting utility.

(1) Changing the points assigned to an empty slot from 16 to 32

To an empty slot (Slot No.3), 32 points are assigned so that I/O numbers for Slot No.4 or higher will not change when a 32-point input module is mounted in the slot in the future. (The points of another empty slot (Slot No.12) is left unchanged as 16 points.)

(a) System configuration and I/O number assignment before change



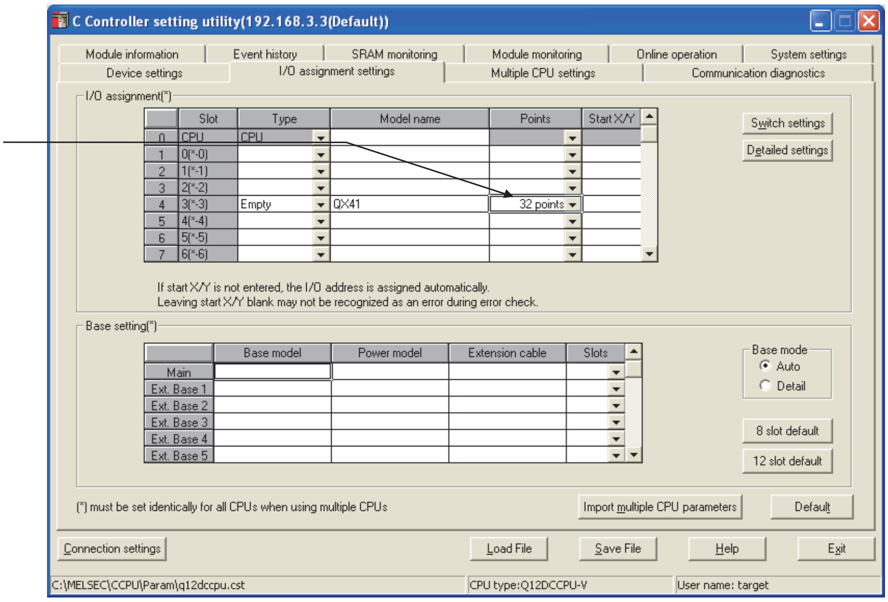
…Slot number	0	1	2	3	4	5	6	7
	Input module	Input module	Input module	Empty	Output module	Output module	Output module	Output module
…Number of I/O points	32 points	32 points	32 points	16 points	32 points	32 points	32 points	32 points
…I/O number	X00 to X1F	X20 to X3F	X40 to X5F	60 to 6F	Y70 to Y8F	Y90 to YAF	YB0 to YCF	YD0 to YEF

	8	9	10	11	12	13	14	15
	Intelligent function module	Intelligent function module	Intelligent function module	Intelligent function module	Empty	Output module	Output module	Output module
	32 points	32 points	32 points	32 points	16 points	32 points	32 points	32 points
	F0 to 10F	110 to 12F	130 to 14F	150 to 16F	170 to 17F	Y180 to Y19F	Y1A0 to Y1BF	Y1C0 to Y1DF

Figure 7.19 I/O number assignment (before changing the points for an empty slot)

(b) I/O assignment setting in C Controller setting utility

In the <<I/O assignment settings>> tab of C Controller setting utility, set Slot No.3 to "32 points".



C Controller setting utility(192.168.3.3(Default))

I/O assignment(*)

	Slot	Type	Model name	Points	Start X/Y
0	CPU	CPU			
1	0(*-0)				
2	1(*-1)				
3	2(*-2)				
4	3(*-3)	Empty	QX41	32 points	
5	4(*-4)				
6	5(*-5)				
7	6(*-6)				

If start X/Y is not entered, the I/O address is assigned automatically.
Leaving start X/Y blank may not be recognized as an error during error check.

Base setting(*)

	Base model	Power model	Extension cable	Slots
Main				
Ext. Base 1				
Ext. Base 2				
Ext. Base 3				
Ext. Base 4				
Ext. Base 5				

Base mode
● Auto
○ Detail

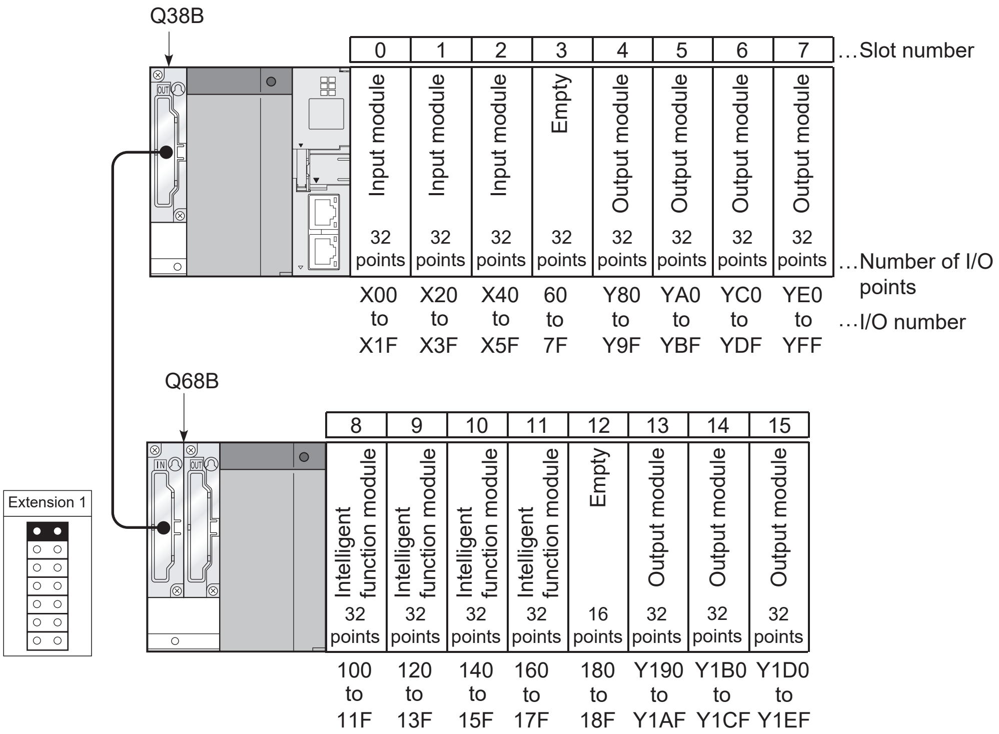
(*) must be set identically for all CPUs when using multiple CPUs

C:\MELSEC\CCPU\Param\q12dccpu.cst | CPU type:Q12DCCPU-V | User name: target

Select 32 points. (When the type is not selected, the type of the installed module will be selected.)

Figure 7.20 I/O assignment setting (when changing the points for empty slot No.3)

(c) I/O number assignment after change



…Slot number	0	1	2	3	4	5	6	7
	Input module	Input module	Input module	Empty	Output module	Output module	Output module	Output module
…Number of I/O points	32 points	32 points	32 points	32 points	32 points	32 points	32 points	32 points
…I/O number	X00 to X1F	X20 to X3F	X40 to X5F	60 to 7F	Y80 to Y9F	YA0 to YBF	YC0 to YDF	YE0 to YFF

Slot number	8	9	10	11	12	13	14	15
	Intelligent function module	Intelligent function module	Intelligent function module	Intelligent function module	Empty	Output module	Output module	Output module
Number of I/O points	32 points	32 points	32 points	32 points	16 points	32 points	32 points	32 points
I/O number	100 to 11F	120 to 13F	140 to 15F	160 to 17F	180 to 18F	Y190 to Y1AF	Y1B0 to Y1CF	Y1D0 to Y1EF

Figure 7.21 I/O number assignment after setting change (change of the points for the empty slot)

(2) Changing the I/O numbers of a slot

I/O numbers of Slot No.3 are changed to X200 to X21F so that I/O numbers for Slot No.4 or higher will not change when a 32-point input module is mounted in the slot in the future.

(a) System configuration and I/O number assignment before change

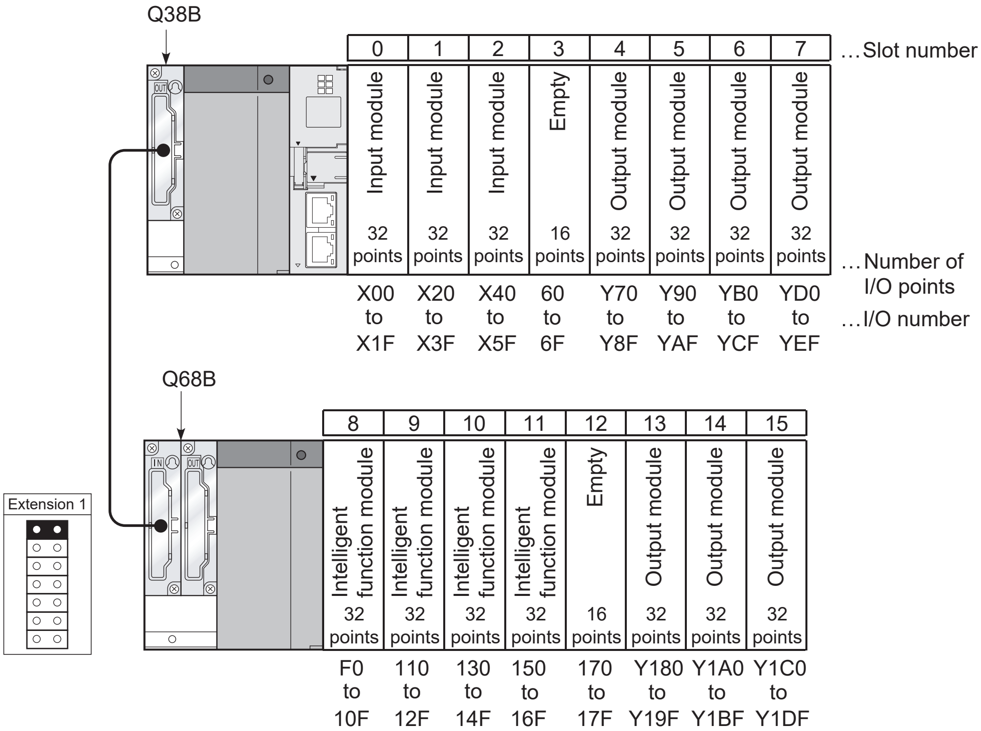


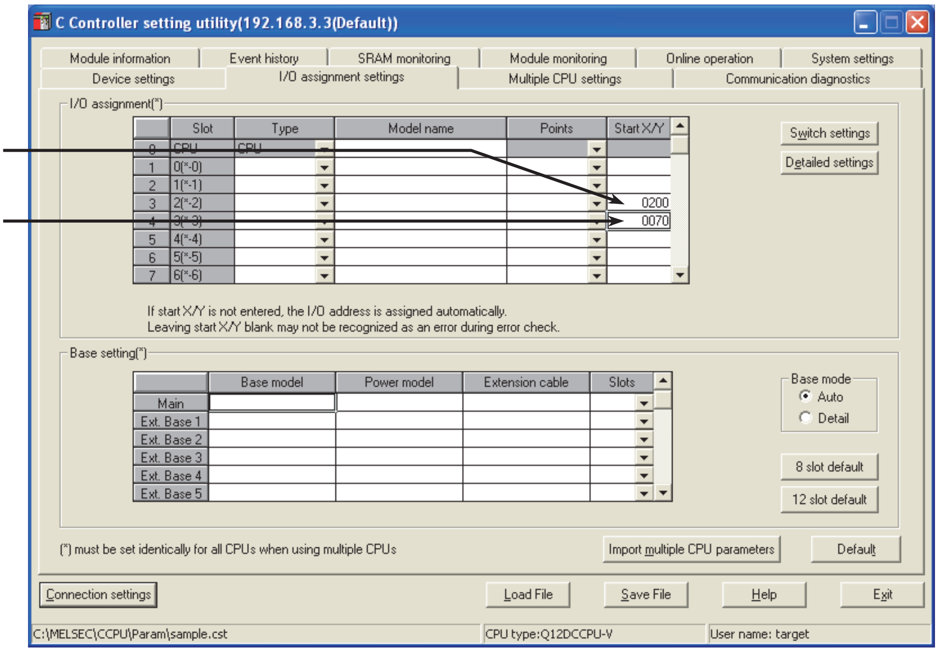
Figure 7.22 I/O number assignment (before changing the I/O numbers of the slot)

(b) I/O assignment setting in C Controller setting utility

In the <<I/O assignment settings>> tab of C Controller setting utility, set "200" for Slot No.3 and "70" for Slot No.4.

Set "200" for the start I/O number.

Set "70" for the start I/O number. (No entry will assign the I/O number next to those of Slot 3.)


C Controller setting utility(192.168.3.3(Default))

I/O assignment(*)

	Slot	Type	Model name	Points	Start X/Y
0	CPU	CPU			
1	0(*-0)				
2	1(*-1)				
3	2(*-2)				0200
4	3(*-3)				0070
5	4(*-4)				
6	5(*-5)				
7	6(*-6)				

If start X/Y is not entered, the I/O address is assigned automatically.
Leaving start X/Y blank may not be recognized as an error during error check.

Base setting(*)

	Base model	Power model	Extension cable	Slots
Main				
Ext. Base 1				
Ext. Base 2				
Ext. Base 3				
Ext. Base 4				
Ext. Base 5				

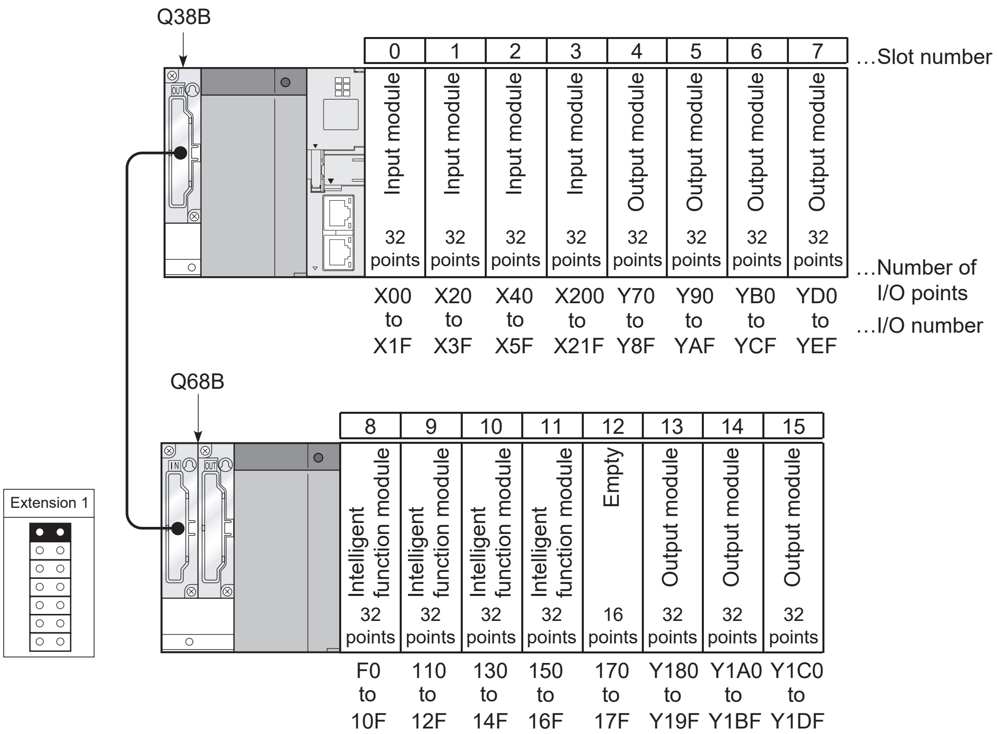
Base mode
● Auto
○ Detail

(*) must be set identically for all CPUs when using multiple CPUs

C:\MELSEC\CCPU\Param\sample.cst
CPU type:Q12DCCPU-V
User name: target

Figure 7.23 I/O assignment setting (when changing the I/O number of Slot 3)

(c) I/O number assignment after change



Q38B

Slot number	0	1	2	3	4	5	6	7
Module	Input module	Input module	Input module	Input module	Output module	Output module	Output module	Output module
Number of I/O points	32 points	32 points	32 points	32 points	32 points	32 points	32 points	32 points
I/O number	X00 to X1F	X20 to X3F	X40 to X5F	X200 to X21F	Y70 to Y8F	Y90 to YAF	YB0 to YCF	YD0 to YEF

Q68B

Slot number	8	9	10	11	12	13	14	15
Module	Intelligent function module	Intelligent function module	Intelligent function module	Intelligent function module	Empty	Output module	Output module	Output module
Number of I/O points	32 points	32 points	32 points	32 points	16 points	32 points	32 points	32 points
I/O number	F0 to 10F	110 to 12F	130 to 14F	150 to 16F	170 to 17F	Y180 to Y19F	Y1A0 to Y1BF	Y1C0 to Y1DF

Figure 7.24 I/O number assignment (after change of the I/O number for the slot)

7.8 Checking I/O Numbers

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	○

The modules mounted with the C Controller module and respective I/O numbers can be checked in "System information" in the <<Module monitoring>> tab of C Controller setting utility.

For C Controller setting utility, refer to the following.

☞ C Controller Module User's Manual (Utility Operation, Programming)

CHAPTER 8 MEMORIES AND FILES

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	○

This chapter explains the memories and file operations of the C Controller module and the precautions for them.

8.1 Memories of the C Controller Module

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	○

This section explains the user memories and system memory used for the C Controller module.

8.1.1 User memories

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	○

The user memories can be read or written by user programs.
The following shows the user memories available for the C Controller module.

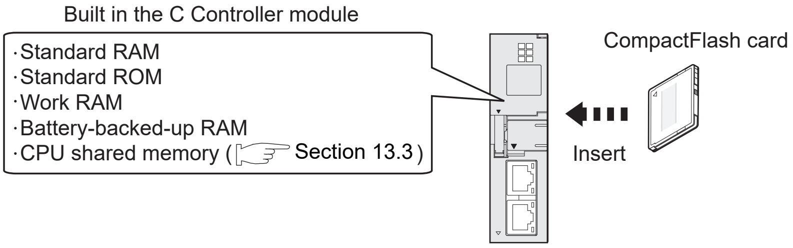


Figure 8.1 User memories

Table 8.1 Availability

Memory	Availability		
	Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
Standard RAM	○	×	×
Standard ROM	×	○	○
Work RAM	○	○	○
Battery-backed-up RAM	○	○	○
CPU shared memory	○	○	○
CompactFlash card	○	○	×

○ : Available, × : N/A

1) Standard RAM
   The standard RAM stores C Controller module parameters and writes, reads and saves data such as a user program file.
   Backup the data stored in the standard RAM since the data in the standard RAM may be lost if the battery life is expired.
   - The following is an example of a script file to be backed up in a CompactFlash card.

```
mkdir("/CF/BackUp")
xcopy("/RAM","/CF/BackUp")
sysLedSet(0xFF)
```

2) Standard ROM
   The standard ROM is used for storing parameters of the C Controller module and data (file) such as the user program.

3) Work RAM
   The work RAM is a volatile memory where a user program of the C Controller module is executed.
   The user programs stored in the standard RAM, standard ROM, and CompactFlash card are read to the work RAM for execution.
   The operating system and system software of the C Controller module are also read to the word RAM for execution.

4) Battery-backed-up RAM
   The battery-backed-up RAM is a memory in which data to be retained after power-off can be stored by a user program.
   To access the battery-backed-up RAM, use any of the following functions in the user program.
   - QBF_WriteSRAM function
   - QBF_ReadSRAM function
   - QBF_WriteSRAM_ISR function
   - QBF_ReadSRAM_ISR function

5) CompactFlash card
   Installing a CompactFlash card into the CompactFlash card interface allows the C Controller module to read and write data (files).

(1) Data storable in user memories of the C Controller module

The following table lists the data storable in user memories of the C Controller module.

Table 8.2 Data that can be stored in user memories

Data	Built in the C Controller module				CompactFlash card
	Standard RAM	Standard ROM	Work RAM	Battery-backed-up RAM	
Parameter (C Controller setting parameter, network parameter)	○*1	○	×	×	○*1*2
Script file	○*3	○*3	×	×	○*3
User program	○*4	○*4	○*6	×	○*4
Data used in the user program (file)	○	○*5	○*6	×	○
Data used in the user program (other than file)	×	×	×	○	×

○ : Storable, × : Not storable

* 1 For the following C Controller modules, store the C Controller setting parameters and network parameters in the same drive, either the standard RAM or CompactFlash card.
- Q12DCCPU-V with a serial number whose first five digits are "12042" or later.

If parameters are stored in different drives, the parameters in the drive where the C Controller setting parameters are stored are valid.
When C Controller setting parameters are stored in both standard ROM and CompactFlash card, the parameters in the CompactFlash card will be valid.

* 2 For the following C Controller modules, parameters cannot be stored in a CompactFlash card.
- Q12DCCPU-V with a serial number whose first five digits are "12041" or higher.
- Q06CCPU-V(-B)

* 3 Store script files in any of the standard RAM, the standard ROM, and a CompactFlash card.

* 4 The stored user program is read out to the work RAM for execution.

* 5 The file read from the user program can be stored.
For writing from the user program, use files in the CompactFlash card. (☞ Page 6-71, Section 6.9)

* 6 A RAM disk must be created by the user program.

Remark

For script files and sample programs, refer to the following.
☞ C Controller Module User's Manual (Utility Operation, Programming)

(2) Drive name

The C Controller module recognizes the following memories by drive name.

- Standard RAM
- Standard ROM
- CompactFlash card

In the user program, a drive name shown in Table 8.3 is used to specify the relevant memory (mentioned above).

These drive names are also used for specifying the target when reading and writing a file to and from the development environment (personal computer) using FTP.

Table 8.3 Memories and drive names

Memory name		Drive name		
		Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
Built in the C Controller module	Standard RAM	/RAM	—	
	Standard ROM	—	/ROM	
	Work RAM	/RAMDISK [*1]		
CompactFlash card		/CF		—

[*1] The drive name in the sample program (MakeRAMDisk.c).
Do not use "/RAM", "/ROM", "/CF", and "/SYSTEMROM" for a drive name of a RAM disk.
The system may not operate properly if any of the above drive name is used.

(3) Memory capacities and necessity of formatting

The following shows respective memory capacities of the C Controller module and whether formatting is necessary or not.

Table 8.4 Memory capacities and necessity of formatting

Memory name	Memory capacity			Formatting
	Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B	
Standard RAM	3M bytes	—		*1
Standard ROM	—	6M bytes		*2
Work RAM	128M bytes	64K bytes		Not necessary
Battery-backed-up RAM	512K bytes[*3]	128K bytes		Not necessary
CompactFlash card	Depends on the capacity of the CompactFlash card installed.		—	*4

* 1 When the IP setting of the C Controller module has returned to the default in the hardware self-diagnostic mode, perform the module initialization setting procedure or format the standard RAM as necessary.

(☞ Page 6-72, Section 6.10.1)

* 2 When the IP setting of the C Controller module has returned to the default in the hardware self-diagnostic mode, perform the module initialization setting procedure or format the standard ROM as necessary.

(☞ Page 6-81, Section 6.10.2)

* 3 128K bytes for the following C Controller modules.
- Q12DCCPU-V with a serial number whose first five digits are "12041" or higher

* 4 For CompactFlash card formatting, refer to the manual for the CompactFlash card used.
The following table shows file systems of a CompactFlash card supported by a C Controller module.

Table 8.5 File systems of a CompactFlash card

File systems	Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
FAT	○	○	A CompactFlash card cannot be used.
FAT32	○	×	
Others	×	×	

○ : Supported, × : Not supported

8.1.2 System memory

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	○

The system memory is a memory used to store data of the operating system, system software, VxWorks image files, etc. of the C Controller module.
The system memory is controlled by the following drive name.

Table 8.6 System memory and drive name

Memory name	Drive name	Remarks
System memory built in the C Controller module	/SYSTEMROM	Read only

☒POINT

When debugging the user program using Workbench or Tornado, check that the VxWorks image file in the system memory of the C Controller module is identical with that of the development environment (personal computer). For the VxWorks image file, refer to the following.
☞ C Controller Module User's Manual (Utility Operation, Programming)

Remark

The system memory data (file) are used in the system of the C Controller module and not required to be used (or referred) in the user program.

8.2 File Operation and Handling Precautions

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	○

8.2.1 File operation

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	○

Files stored in the standard RAM, standard ROM, or a CompactFlash card can be read, written, and verified with the following.

1) C Controller setting utility
2) CC-Link utility
3) MELSECNET/H utility
4) CC IE Control utility
   For the Q06CCPU-V-B, CC IE Control utility is not available.
5) FTP access from the development environment (personal computer)
6) Copying between a CompactFlash card and the standard RAM or standard ROM

The following shows the availability of each operation.

Table 8.7 Availability of file operation

File operation		Availability					
		1)	2)	3)	4)	5)	6)
C Controller module parameters[*1]	Read	○	×	×	×	○[*2]	○[*2 *4]
	Write	○	×	×	×	○[*3]	○[*3 *4]
	Verify parameters	○	×	×	×	×	×
CC-Link module parameters[*1]	Read	×	○	×	×	○[*2]	○[*2 *4]
	Write	×	○	×	×	○[*3]	○[*3 *4]
	Verify parameters	×	○	×	×	×	×
MELSECNET/H module parameters[*1]	Read	×	×	○	×	○[*2]	○[*2 *4]
	Write	×	×	○	×	○[*3]	○[*3 *4]
	Verify parameters	×	×	○	×	×	×
CC-Link IE controller network module parameters[*1]	Read	×	×	×	○	○[*2]	○[2 *4]
	Write	×	×	×	○	○[*3]	○[3 *4]
	Verify parameters	×	×	×	○	×	×
Intelligent function module parameters[*1]	Read	○	×	×	×	○[*2]	○[*2 *4]
	Write	○	×	×	×	○[*3]	○[*3 *4]
Script file	Read	×	×	×	×	○	○
	Write	×	×	×	×	○	○
User program	Read	×	×	×	×	○	○
	Write	×	×	×	×	○	○
File used in the user program	Read	×	×	×	×	○	○
	Write	×	×	×	×	○	○

○ : Available × : N/A

* 1 Parameter files are stored in the "Melco" and "Mqbf" directories and lower layer of the standard RAM or standard ROM.

* 2 To backup a parameter file to a CompactFlash card or another memory, copy the data in "Melco" and "Mqbf" directories and lower layer under the "/RAM" or "ROM" drive.

* 3 To restore a parameter file from a CompactFlash card or another memory, copy all the previously backed-up data (*2 above) to the "/RAM" or "/ROM" drive after the following operation.
Q12DCCPU-V: "/RAM" drive formatting (☞ Page 6-77, Section 6.10.1 (6))
Q06CCPU-V(-B): "/ROM" drive formatting (☞ Page 6-85, Section 6.10.2 (4))

* 4 When SW3PVC-CCPU is installed, the script files used for backup or restoration of parameter files are registered.
For sample programs, refer to the following.
☞ C Controller Module User's Manual (Utility Operation, Programming)

8.2.2 Precautions for handling files

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	○

This section explains precautions for handling files of the C Controller module.

(1) When writing a file

The following are precautions for writing a user file using FTP or copy registration from a CompactFlash card to the standard RAM or standard ROM.

- Terminate all the user programs running in the C Controller module, which use the file to be written.
- Check that the user file to be copied does not affect the running user program.

(2) Powering off (including reset) during file operation

(a) Data corruption or file system failure may occur if the C Controller system is powered off or reset (including remote RESET) while the user file in the following memory is being written.

- Standard RAM
- Standard ROM
- CompactFlash card

When the file in the above memory is being written, close the file before powering off or resetting.

(b) Data may not be updated if the C Controller system is powered off or reset (including remote reset) without closing the user file being used in the following memory.

- Standard RAM
- Standard ROM
- CompactFlash card

Close all the files being used before overwriting.

(3) Operating the same file concurrently

Access to a file being written is not allowed for the C Controller module.
Writing data to a file being accessed is also not allowed.
To perform multiple operations to the same file, wait for one processing to complete before starting another.

(4) File name and directory name

Use alphanumeric characters and special characters (excluding \, /, *, ?, <, >, |, :, and ") for the names of a file and a directory stored in the following memories in the C Controller module:

- Standard RAM
- Standard ROM
- CompactFlash card

If a file or a directory with a name including characters other than alphanumeric characters and special characters (excluding \, /, *, ?, <, >, |, :, and ") is stored in the memories above in the C Controller module, the following phenomena may occur:

- Character corruption on file names and directory names
- Loss of files and directories

(5) The number of files and directories

Up to 256 files and directories in total can be created in the root directory of the following C Controller module memories.

- Standard RAM
- Standard ROM
- CompactFlash card

Note that the maximum number of them may be decreased depending on the file name length.

The number of files in the sub directory must be 500 or less.

Storing more than 500 files may significantly increase the file access time.

(6) Precautions for the storage location of user file

Do not store user files in the following directories of the C Controller module. Doing so can cause the loss of user files since the following folders are used as storage locations for parameter files.

Table 8.8 Parameter storage location directories

Directory	Remarks
/RAM/Melco/	Q12DCCPU-V only
/RAM/Mqbf/	
/ROM/Melco/	Q06CCPU-V(-B) only
/ROM/Mqbf/	
/CF/Melco/	Only when CompactFlash card is inserted
/CF/Mqbf/	

(7) Creating a directory

Do not create a directory in the root "/" of the C Controller module.

Such directory creation processing fails or an unintended directory may be created.

(These errors occur also when transferring a directory to the root "/" through FTP.)

CHAPTER 9 DEVICE DESCRIPTION

Q12DCCPU-V	Q12DCCPU-V	Q06CCPU-V
○	△	△

This chapter explains devices that can be used with C Controller modules.

9.1 Applicable Devices with C Controller Modules

The following table shows devices that can be used with C Controller modules and the ranges of use.

Table 9.1 Device list

Classification	Type	Default value				Change setting range by the utility	Reference
		Device name	Number of points	Range of use			
I/O device	Bit device	Input	4096 points	X0 to FFF	Hexa decimal	N/A	Page 9-2
		Output	4096 points	Y0 to FFF	Hexa decimal		Page 9-2
Internal user device*1	Bit device	Internal relay	8192 points	M0 to 8191	Decimal	Available*2	Page 9-3
	Word device	Data register	12288 points	D0 to 12287	Decimal		Page 9-3
Internal system device*1	Bit device	Special relay	2048 points	SM0 to 2047	Decimal	N/A	Page 9-3
	Word device	Special register	2048 points	SD0 to 2047	Decimal		Page 9-3
Link direct device	Bit device	Link input	8192 points	Jn\X0 to 1FFF	Hexa decimal	N/A	Page 9-4
		Link output	8192 points	Jn\Y0 to 1FFF	Hexa decimal		
		Link relay	32768 points*3	Jn\B0 to 7FFF*4	Hexa decimal		
		Link special relay	512 points	Jn\SB0 to 1FF	Hexa decimal		
	Word device	Link register	131072 points*5	Jn\W0 to 1FFFF*6	Hexa decimal		
		Link special register	512 points	Jn\SW0 to 1FF	Hexa decimal		
Module access device	Word device	Intelligent function module device	65536 points	Un\G0 to 65535*7	Decimal	N/A	Page 9-5
		Multiple CPU area device*8	4096 points	U3En\G0 to 4095	Decimal	Available*9	Page 9-5
			14336 points	U3En\G10000 to 24335			

* 1 These devices cannot be used with the following C Controller modules.
  - Q12DCCPU-V with a serial number whose first five digits are "12041" or earlier
  - Q06CCPU-V(-B)
* 2 The setting range can be changed from the <<Device settings>> tab of the C Controller setting utility.
* 3 For MELSECNET/H, the number of points is 16384.
* 4 For MELSECNET/H, the range is Jn\B0 to 3FFF.
* 5 For MELSECNET/H, the number of points is 16384.
* 6 For MELSECNET/H, the range is Jn\W0 to 3FFF.
* 7 For the number of points that can be actually used, refer to the manual for the intelligent function module to be used.
* 8 These devices can be used only for the multiple CPU system configurations.
* 9 The setting range can be changed from the <<Multiple CPU settings>> tab of the C Controller setting utility.

9.2 I/O Device

This section explains devices that can be used to suit the user application.

9.2.1 Input (X)

Input is used for giving commands and data to the C Controller module from an external device such as a push-button switch, a selector switch, a limit switch, or a digital switch.

☒POINT

(1) Writing to the input (X) is ignored.

(2) If the following input (X) is read out, the read data will be 0 (OFF).

- No module is installed.
- An output module is installed.

9.2.2 Output (Y)

Output is for outputting program control results to an external signal lamp, digital HMI, electromagnetic switch (contactor), solenoid, etc.

☒POINT

If the following output (Y) is read out, the read data will be 0 (OFF).

- No module is installed.
- An output module is installed.

9.3 Internal User Device

This section explains devices which are created on the work RAM of the C Controller module.
This is a device that can be used to suit the user application.

An internal user device is created with device functions.
For the device functions, refer to the following.
☞ Page 4-18, Section 4.4

9.3.1 Internal relay (M)

This is an auxiliary relay used inside the C Controller module.

9.3.2 Data register (D)

This memory can store numerical data (-32768 to 32767 or 0000H to FFFFH).

9.4 Internal System Device

This section explains devices for systems.

9.4.1 Special relay (SM)

The special relay is an internal relay, the specifications for which are determined inside the C Controller module, which stores the status of the C Controller module.

For the special relays, refer to the following.
☞ Page App-15, Appendix 6

9.4.2 Special register (SD)

The special register is an internal register with the determined specifications inside the C Controller module, and the register stores the status (failure diagnostic, system information, etc.) of the C Controller module.

For the special registers, refer to the following.
☞ Page App-16, Appendix 7

9.5 Link Direct Device

This device directly accesses a link device in the CC-Link IE controller network module or in the MELSECNET/H module.
Link devices in the CC-Link IE controller network module or MELSECNET/H module can be read or written by user programs, independently of the link device refresh of the C Controller module.

For accessing the link device, refer to the following.

MELSECNET/H	: ☞ Page 5-15, Section 5.3.2
CC-Link IE controller network	: ☞ Page 5-42, Section 5.4.2

9.6 Module Access Device

9.6.1 Intelligent function module device

This device is for directly accessing the buffer memory of an intelligent function module installed in the base unit from the C Controller module.

9.6.2 Multiple CPU area device

This device accesses the CPU shared memory in each CPU module of the multiple CPU system.

For the CPU shared memory, refer to the following.
☞ Page 13-12, Section 13.3

CHAPTER 10 MULTIPLE CPU SYSTEM OVERVIEW

10.1 What is the Multiple CPU System?

(1) Configuration of a multiple CPU system

A multiple CPU system consists of two or more CPU modules mounted on a main base unit, and each CPU module controls I/O module(s) and/or intelligent function module(s) individually.[*1]

Note that redundant power main base units and redundant power extension base units cannot be used with the C Controller module.

* 1 From the C Controller module, High Performance model QCPU, Process CPU, and Universal Model QCPU (except for the Q00UCPU, Q01UCPU, and Q02UCPU), up to four CPU modules can be used in a multiple CPU system.
From the Basic model QCPU and Universal model QCPU (Q00UCPU, Q01UCPU, and Q02UCPU only), up to three CPU modules can be used in a multiple CPU system.

Applicable CPU models are shown in Table 10.1.
Redundant CPUs cannot be used in multiple CPU systems.
For applicable module versions, refer to the following.
☞ Page 11-11, Section 11.2

Table 10.1 Applicable CPU modules

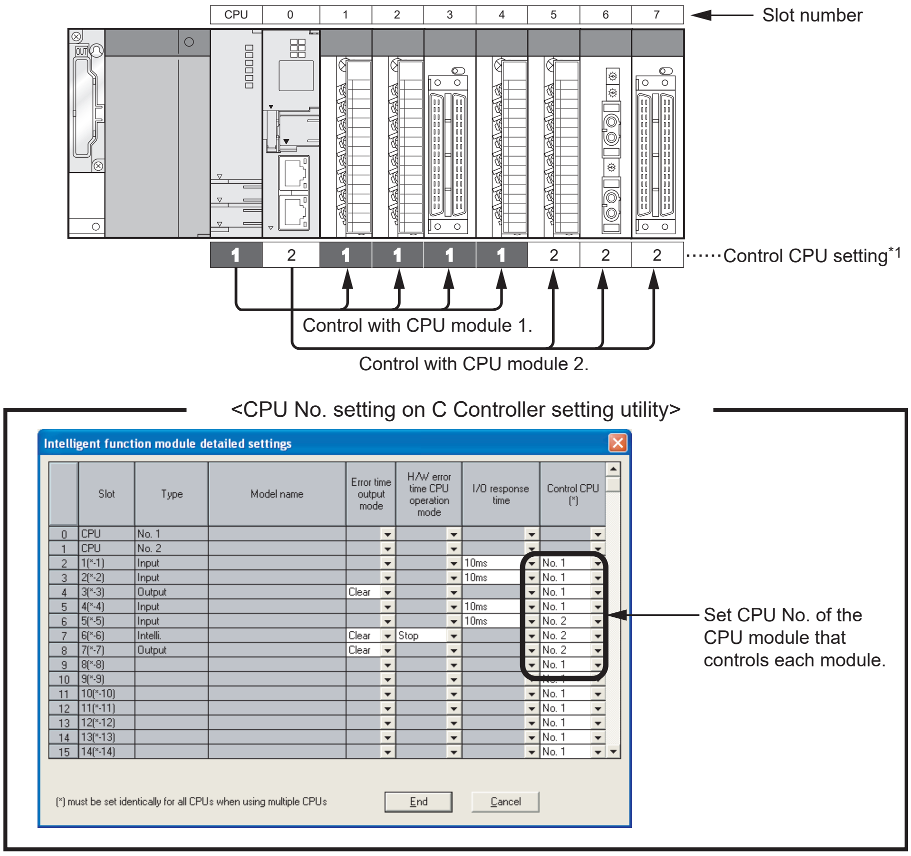
CPU module[*2]		Model
C Controller module		Q06CCPU-V, Q06CCPU-V-B, Q12DCCPU-V (Basic mode), Q12DCCPU-V (Extended mode), Q24DHCCPU-V, Q24DHCCPU-LS
Programmable controller CPU	Basic model QCPU[*3]	Q00CPU, Q01CPU
	High Performance model QCPU	Q02CPU, Q02HCPU, Q06HCPU, Q12HCPU, Q25HCPU
	Process CPU	Q02PHCPU, Q06PHCPU, Q12PHCPU, Q25PHCPU
	Universal model QCPU[*4]	Q00UCPU, Q01UCPU, Q02UCPU, Q03UDCPU, Q04UDHCPU, Q06UDHCPU, Q10UDHCPU, Q13UDHCPU, Q20UDHCPU, Q26UDHCPU, Q03UDECPU, Q04UDEHCPU, Q06UDEHCPU, Q10UDEHCPU, Q13UDEHCPU, Q20UDEHCPU, Q26UDEHCPU, Q50UDEHCPU, Q100UDEHCPU, Q03UDVCPU, Q04UDVCPU, Q06UDVCPU, Q13UDVCPU, Q26UDVCPU
Motion CPU		Q172CPUN, Q173CPUN, Q172CPUN-T, Q173CPUN-T, Q172HCPU, Q173HCPU, Q172HCPU-T, Q173HCPU-T
		Q172DCPU, Q173DCPU

* 2 The PC CPU module cannot be used in combination with the C Controller module in a multiple CPU system.
* 3 The Q00JCPU cannot be used in a multiple CPU system.
* 4 The Q00UJCPU cannot be used in a multiple CPU system.

Select CPU modules that are suitable for the system scale and application.
For combinations of configurable CPU modules, refer to the following.
☞ Page 12-1, Section 12.1

(2) Controlling I/O modules and intelligent function modules

In a multiple CPU system, it is necessary to specify which CPU module controls which I/O module or intelligent function module (control CPU setting).


CPU
0
1
2
3
4
5
6
7
Slot number
1
2
1
1
1
1
2
2
2
Control CPU setting*1
Control with CPU module 1.
Control with CPU module 2.
<CPU No. setting on C Controller setting utility>
Set CPU No. of the CPU module that controls each module.

Intelligent function module detailed settings

	Slot	Type	Model name	Error time output mode	H/W error time CPU operation mode	I/O response time	Control CPU (*)
0	CPU	No. 1					
1	CPU	No. 2					
2	1(*-1)	Input				10ms	No. 1
3	2(*-2)	Input				10ms	No. 1
4	3(*-3)	Output		Clear			No. 1
5	4(*-4)	Input				10ms	No. 1
6	5(*-5)	Input				10ms	No. 2
7	6(*-6)	Intelli.		Clear	Stop		No. 2
8	7(*-7)	Output		Clear			No. 2
9	8(*-8)						No. 1
10	9(*-9)						No. 1
11	10(*-10)						No. 1
12	11(*-11)						No. 1
13	12(*-12)						No. 1
14	13(*-13)						No. 1
15	14(*-14)						No. 1

(*) must be set identically for all CPUs when using multiple CPUs

End Cancel

Figure 10.1 Control CPU setting

* 1 Indicates the grouping configuration on C Controller setting utility.
"No.1" entered for a CPU module represents "CPU No.1", and the "No.1" entered for an I/O module or intelligent function module indicates that the control CPU of the module is CPU No.1.

The CPU module that controls each I/O module or intelligent function module is called "control CPU".

An I/O module or intelligent function module controlled by a control CPU is called "controlled module".

A module that are controlled by another CPU, not by the control CPU, is called "non-controlled module".

(3) Application example of a multiple CPU system

The optimum load distribution system can be configured when programmable controller CPUs process time-critical cyclic tasks and the C Controller module performs processing of C language information.

By adding a Motion CPU to the multiple CPU system, motion control, sequence control, and information processing (data processing and communication) can be integrated into an advanced system.

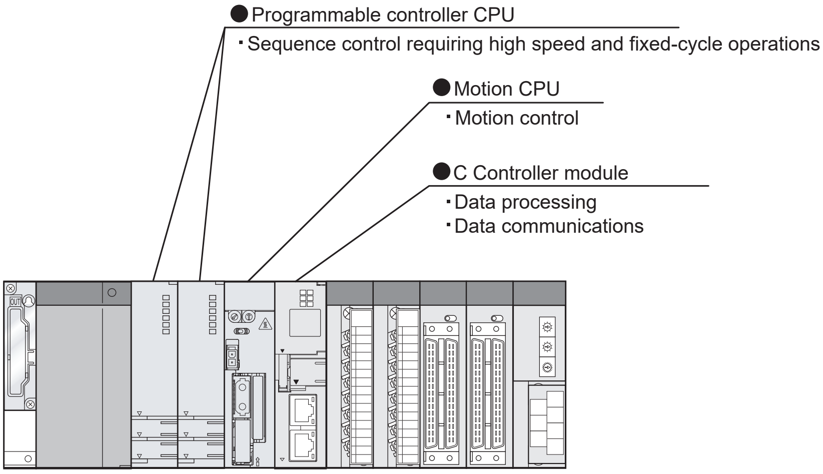

● Programmable controller CPU
・Sequence control requiring high speed and fixed-cycle operations
● Motion CPU
・Motion control
● C Controller module
・Data processing
・Data communications

Figure 10.2 Application example of the multiple CPU system

(4) Data exchange between CPU modules

Since a multiple CPU system has a CPU shared memory for data exchange between CPU modules, each CPU can read and write data from and to another CPU by using the following function.

- Data communications using CPU shared memory (☞ Page 13-12, Section 13.3)

(5) Access to a non-controlled module

Each CPU module in a multiple CPU system can access a non-controlled module.

For example, CPU No.2 can read ON/OFF information of an I/O module and buffer memory data of an intelligent function module that are controlled by CPU No.1.

To access a non-controlled module, enable the "I/O sharing when using Multiple CPUs" in the <<Multiple CPU setting>> tab of C Controller module setting utility. (☞Page 12-13, Section 12.4.2)

(6) Access range of Device monitoring utility

Each of the CPU modules installed in a multiple CPU system can be accessed from Device monitoring utility.

For the access range of Device monitoring utility, refer to the following.

☞ C Controller Module User's Manual (Utility Operation, Programming)

CHAPTER 11 MULTIPLE CPU SYSTEM CONFIGURATION

This chapter explains multiple CPU system configurations and precautions on them.

Remark

For restrictions on using AnS series modules with a programmable controller CPU in a multiple CPU system where a C Controller module is installed, refer to the following.

☞ QCPU User's Manual (Multiple CPU System)

11.1 System Configuration

This section explains system configurations, connection with the development environment, and entire system overview of multiple CPU systems.
For single CPU system configurations, refer to CHAPTER 2.

Remark

For the system configuration where a programmable controller CPU is set as CPU No.1, refer to the following.

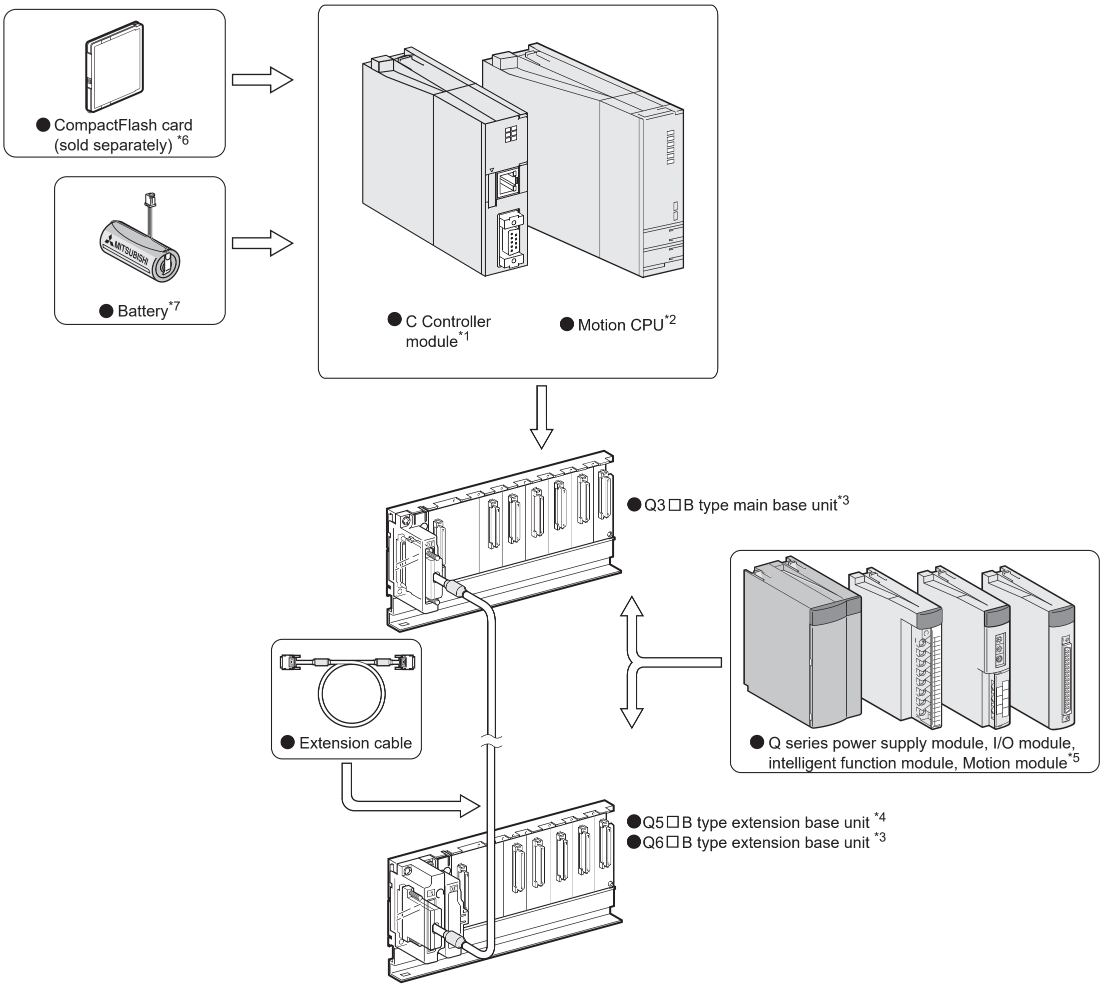
☞ QCPU User's Manual (Multiple CPU System)

For the system configuration where a Q24DHCCPU-V/-VG/-LS, Q26DHCCPU-LS, or Q12DCCPU-V (Extended mode) C Controller module is set as CPU No.1, refer to the following.

☞ MELSEC-Q C Controller Module User's Manual

11.1.1 Devices used for each base unit

(1) When using the main base unit (Q3□B)


● CompactFlash card (sold separately) *6
● Battery*7
● C Controller module*1
● Motion CPU*2
● Q3□B type main base unit*3
● Q series power supply module, I/O module, intelligent function module, Motion module*5
● Extension cable
●Q5□B type extension base unit *4
●Q6□B type extension base unit *3

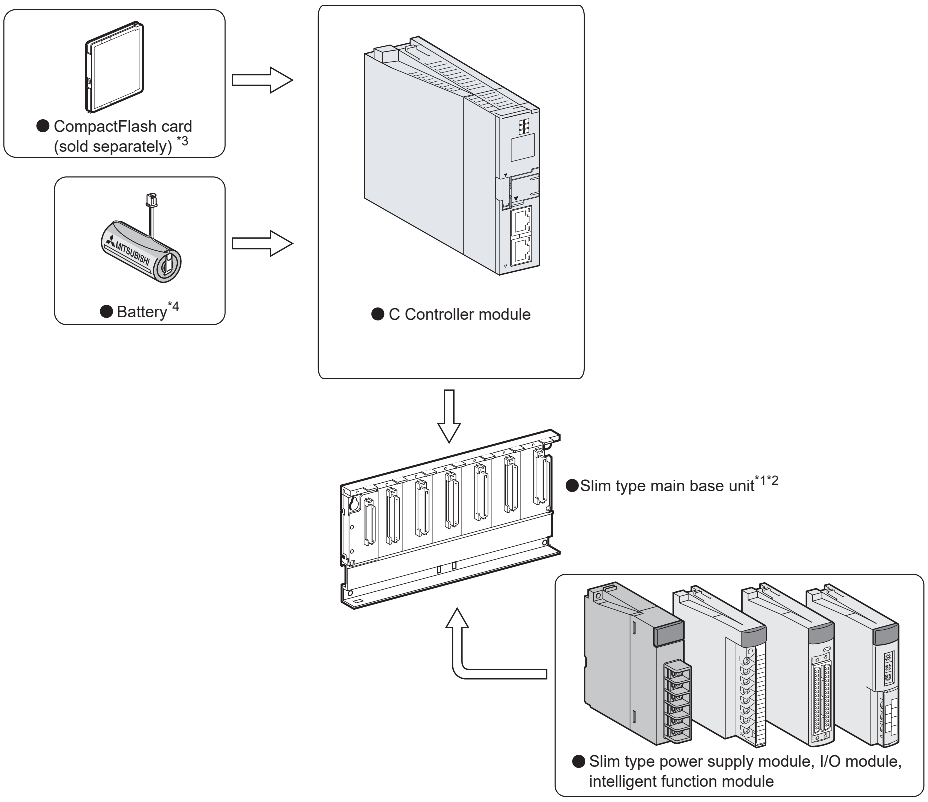
* 1 The Q12DCCPU-V and a Motion CPU cannot be installed on the same base unit.
* 2 To a Motion CPU, a CompactFlash card and a battery cannot be installed.
* 3 Use a Q series power supply module.
The current consumption must be the rated output current or less.
Slim type power supply modules and redundant power supply modules cannot be used.
* 4 No power supply module is required for the Q5□B extension base unit.
* 5 The control CPU of a Motion module must be a Motion CPU.
* 6 For the Q06CCPU-V-B, CompactFlash cards cannot be used.
* 7 For the Q06CCPU-V(-B), the Q7BAT cannot be used.

Figure 11.1 When a main base unit is used

☒POINT

1. For combinations of configurable CPU modules, refer to the following.
   ☞ Page 12-1, Section 12.1
2. When a C Controller module is used in a multiple CPU system, any redundant power base units cannot be used.

(2) When using the slim type main base unit (Q3□SB)


● CompactFlash card (sold separately) *3
● Battery *4
● C Controller module
●Slim type main base unit *1*2
● Slim type power supply module, I/O module, intelligent function module

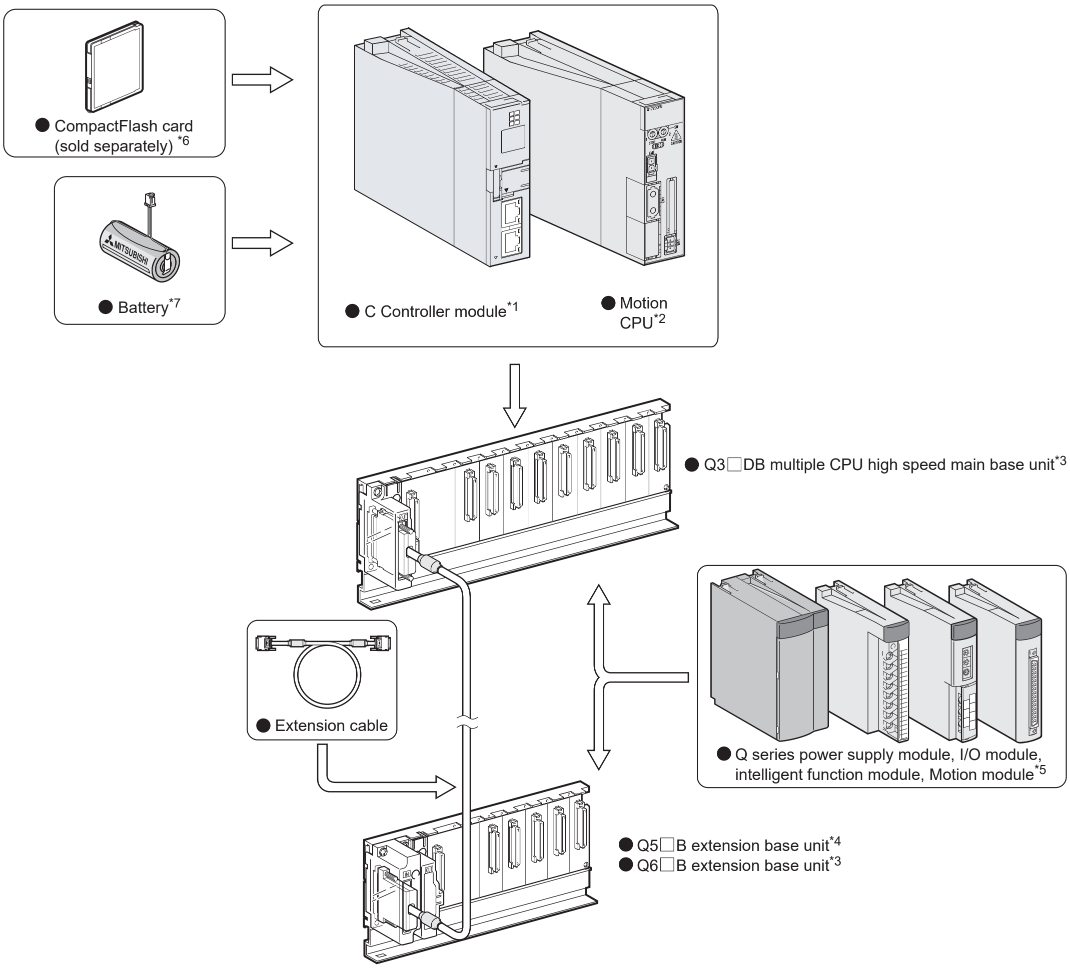
* 1 Use a slim type power supply module.
The current consumption must be the rated output current or less.
Q series power supply modules and redundant power supply modules cannot be used.
* 2 The slim type main base unit does not have an extension cable connector.
Extension base units and a GOT are not connectable.
* 3 For the Q06CCPU-V-B, CompactFlash cards cannot be used.
* 4 For the Q06CCPU-V(-B), the Q7BAT cannot be used.

Figure 11.2 When a slim type main base unit is used

☒POINT

1. For combinations of configurable CPU modules, refer to the following.
☞ Page 12-1, Section 12.1
2. A Motion CPU cannot be used when a slim type main base unit (Q3□SB) is used.

(3) When using the multiple CPU high speed main base unit (Q3□DB)


● CompactFlash card (sold separately)[*6]
● Battery[*7]
● C Controller module[*1]
● Motion CPU[*2]
● Q3□DB multiple CPU high speed main base unit[*3]
● Extension cable
● Q series power supply module, I/O module, intelligent function module, Motion module[*5]
● Q5□B extension base unit[*4]
● Q6□B extension base unit[*3]

* 1 The Q12DCCPU-V and a Motion CPU (except the Q172DCPU and Q173DCPU) cannot be installed on the same base unit.
* 2 To a Motion CPU, a CompactFlash card and a battery cannot be installed.
* 3 Use a Q series power supply module.
The current consumption must be the rated output current or less.
Slim type power supply modules and redundant power supply modules cannot be used.
* 4 No power supply module is required for the Q5□B extension base unit.
* 5 The control CPU of a Motion module must be a Motion CPU.
* 6 For the Q06CCPU-V-B, CompactFlash cards cannot be used.
* 7 For the Q06CCPU-V(-B), the Q7BAT cannot be used.

Figure 11.3 When a multiple CPU high speed main base unit is used

☒POINT

For combinations of configurable CPU modules, refer to the following.
☞ Page 12-1, Section 12.1

11.1.2 Connection with a development environment

Configurations of a development environment in a single CPU system can be applied to multiple CPU systems. (☞ Page 2-5, Section 2.1.2)

11.1.3 Connection with peripheral devices

Configurations of peripheral devices in a single CPU system can be applied to multiple CPU systems. (☞ Page 2-7, Section 2.1.3)

11.1.4 System configuration overview (when CPU No.1 is a C Controller module)

(1) When using the main base unit (Q3□B)

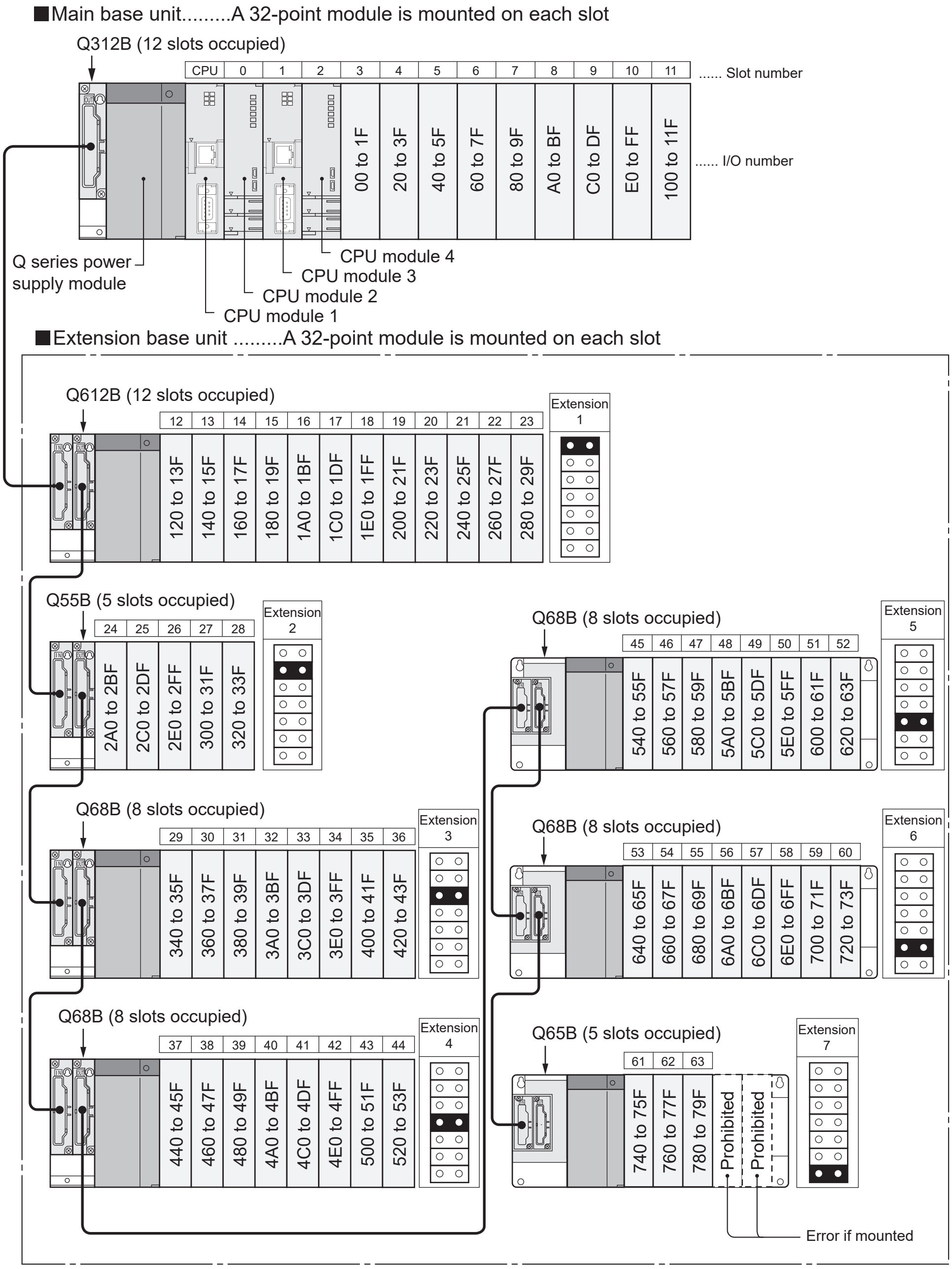


Figure 11.4 Configuration example of a system using a main base unit

Table 11.1 Restrictions on the system configuration

CPU number	CPU module 1: CPU No.1, CPU module 2: CPU No.2, CPU module 3: CPU No.3, CPU module 4: CPU No.4	
Maximum number of extension base units	7 units	
Maximum number of I/O modules	65 minus (Number of CPUs)	
Main base unit model	Q33B, Q35B, Q38B, Q312B	
Extension base unit model	No power supply module required	Q52B, Q55B
	Q series power supply module required	Q63B, Q65B, Q68B, Q612B
Extension cable model	QC05B, QC06B, QC12B, QC30B, QC50B, QC100B	
Q series power supply module model	Q61P-A1, Q61P-A2, Q61P, Q61P-D, Q62P, Q63P, Q64P, Q64PN	

■ Precautions

- The total extension cable length must be 13.2m (43.31 feet) or less.
- Do not install the extension cables together with the main circuit lines (high voltage and large current).
- Set a unique number for each extension base.
- If some Q5□B and Q6□B extension base units are used on the same system, they can be connected in any order.
  However, confirm the applicability of each unit referring to the QCPU User's Manual (Hardware Design, Maintenance and Inspection).
- The QA1S6□B and Q6□RB extension base units cannot be used.
- Connect the extension cable from OUT of the extension cable connector of the base unit to IN of the next extension base unit.
- Mounting 66 or more modules (including CPU modules) causes an intelligent function module assignment error (error code: 2124, SP.UNIT LAY ERR.)
- "Number of CPUs" is the number of CPU modules set for "Number of CPUs" in the <<Multiple CPU settings>> tab of C Controller setting utility.
- For details of the Motion CPU, refer to the manual for the Motion CPU.
- When the Q12DCCPU-V is CPU No.1, no Motion CPU can be installed.
  When the Q06CCPU-V(-B) is CPU No.1, the Q172DCPU and Q173DCPU cannot be installed.
- Programmable controller CPUs cannot be installed.
- When a C Controller module is used in a multiple CPU system, redundant power base units cannot be used.

(2) When using the slim type base unit (Q3□SB)

■Slim type main base unit ....A 32-point module is mounted on each slot

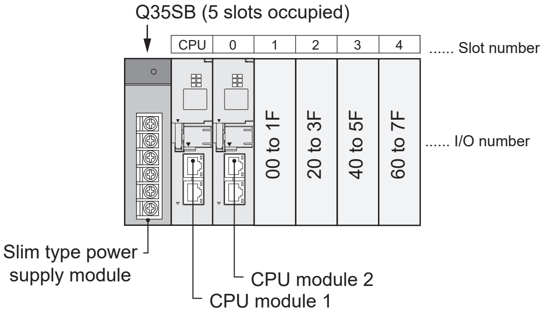


Figure 11.5 Configuration example of a system using a slim type base unit

Table 11.2 Restrictions on the system configuration

CPU number	CPU module 1: CPU No.1 (C Controller module) CPU module 2: CPU No.2 (C Controller module)	
Maximum number of extension base units	Not extendable	
Maximum number of I/O modules	Q32SB	1 module
	Q33SB	2 modules
	Q35SB	4 modules
Main base unit model	Q32SB, Q33SB, Q35SB	
Q series power supply module model	Q61SP	

■ Precautions

- When configuring a multiple CPU system using a slim type main base unit, C Controller modules only can be used.
- The slim type main base unit has no extension cable connector. Connection of extension base units and a GOT (in a bus topology) is not allowed.
- Mounting three or more CPU modules is not allowed since the rating output current of the power supply module (Q61SP) will exceed the current consumption of the CPU modules.

(3) When using the multiple CPU high speed main base unit (Q3□DB)

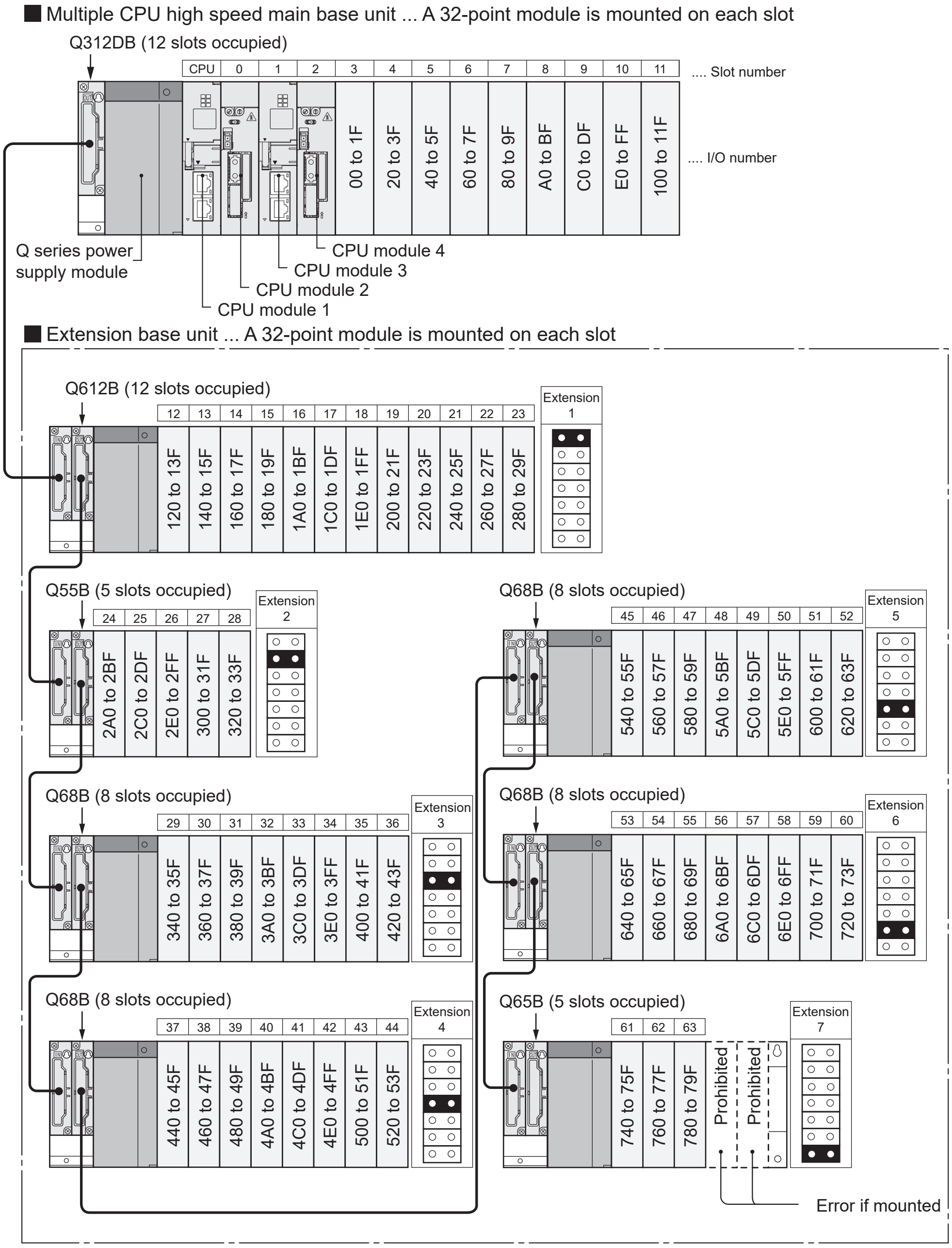


Figure 11.6 Configuration example of a system using a multiple CPU high speed main base unit

Table 11.3 Restrictions on the system configuration

CPU number	CPU module 1: CPU No.1, CPU module2: CPU No.2, CPU module 3: CPU No.3, CPU module 4: CPU No.4	
Maximum number of extension base units	7 units	
Maximum number of I/O modules	65 minus (Number of CPUs)	
Main base unit model	Q35DB, Q38DB, Q312DB	
Extension base unit model	No power supply module required	Q52B, Q55B
	Q series power supply module required	Q63B, Q65B, Q68B, Q612B
Extension cable model	QC05B, QC06B, QC12B, QC30B, QC50B, QC100B	
Q series power supply module model	Q61P-A1, Q61P-A2, Q61P, Q61P-D, Q62P, Q63P, Q64P, Q64PN	

■ Precautions

- The total extension cable length must be 13.2m (43.31 feet) or less.
- Do not install the extension cables together with the main circuit lines (high voltage and large current).
- Set a unique number for each extension base.
- If some Q5□B and Q6□B extension base units are used on the same system, they can be connected in any order.
  However, confirm the applicability of each unit referring to the QCPU User's Manual (Hardware Design, Maintenance and Inspection).
- The QA1S6□B and Q6□RB cannot be used as an extension base unit.
- Connect the extension cable from OUT of the extension cable connector of the base unit to IN of the next extension base unit.
- Mounting 66 or more modules (including CPU modules) causes an intelligent function module assignment error (error code: 2124, SP.UNIT LAY ERR.)
- "Number of CPUs" is the number of CPU modules set for "Number of CPUs" in the <<Multiple CPU settings>> tab of C Controller setting utility.
- For details of the Motion CPU, refer to the manual for the Motion CPU.
- When the Q12DCCPU-V is CPU No.1, only the Q172DCPU and Q173DCPU can be installed.
  When the Q06CCPU-V(-B) is CPU No.1, no Motion CPU can be installed.
- Programmable controller CPUs cannot be installed.

11.2 Applicable Modules

11.2.1 Applicable CPU modules

Table 11.4 and Table 11.5 lists the CPU modules that can configure a multiple CPU system when the C Controller module is CPU No.1.
For how to check the version of each CPU module, refer to the manual for respective modules.

(1) When the Q12DCCPU-V is CPU No.1

Table 11.4 Applicable CPU modules

CPU module	Model	Restriction
C Controller module	Q06CCPU-V	—
	Q06CCPU-V-B	
	Q12DCCPU-V	
	Q24DHCCPU-V	
	Q24DHCCPU-LS	
Motion CPU	Q172DCPU	—
	Q173DCPU	

—: No restrictions

(2) When the Q06CCPU-V(-B) is CPU No.1

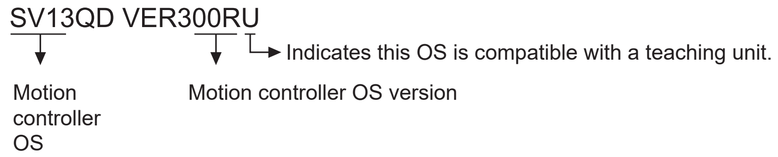
Table 11.5 Applicable CPU modules

CPU module	Model name	Restrictions
C Controller module	Q06CCPU-V	—
	Q06CCPU-V-B	
	Q12DCCPU-V*1	
	Q24DHCCPU-V	
	Q24DHCCPU-LS	
Motion CPU	Q172CPUN	The operating system must be SV13, or the version of SV22 must be "00R" or later*2
	Q172CPUN-T	
	Q172HCPU	—
	Q172HCPU-T	
	Q173CPUN	The operating system must be SV13, or the version of SV22 must be "00R" or later*2
	Q173CPUN-T	
	Q173HCPU	—
	Q173HCPU-T	

—: No restrictions

* 1 The Q12DCCPU-V and a Motion CPU cannot be installed on the same base unit.
* 2 The versions can be applied to multiple CPU systems when the QBF_MotionDDWR or QBF_MotionDDRD function is used in the user program.
When the above functions are not used, no restriction is applied to those versions.

Example) Checking the version for the Motion CPU in MT Developer


SV13QD VER300RU
Indicates this OS is compatible with a teaching unit.
Motion controller OS
Motion controller OS version

Remark

For CPU modules that can be used when a programmable controller CPU is CPU No.1, refer to the following.

☞ QCPU User's Manual (Multiple CPU System)

11.2.2 Precautions for using I/O modules and intelligent function modules

(1) Applicable I/O modules

All I/O modules (QX□ and QY□) can be used in the multiple CPU system.
Any of CPU No.1 to CPU No.4 can be set as a control CPU for each of them.

(2) Applicable intelligent function modules

(a) When using an intelligent function module in a multiple CPU system
The function version of the intelligent function module must be B or later.
Any of CPU No.1 to CPU No.4 can be set as a control CPU for it.

(b) Q series high-speed counter modules
The Q series high-speed counter modules (QD62, QD62D, and QD62E) support multiple CPU systems from function version A.
Any of CPU No.1 to CPU No.4 can be set as a control CPU for each of them.

(c) Q series interrupt modules
The Q series interrupt module does not have a function version, and can be used in multiple CPU systems.
Any of CPU No.1 to CPU No.4 can be set as a control CPU for it.

(3) Access range to controlled and non-controlled modules

In a multiple CPU system, a non-controlled module can be accessed by enabling "I/O sharing when using Multiple CPUs" in the <<Multiple CPU settings>> tab of C Controller setting utility.
For accessibility to the controlled and non-controlled modules in a multiple CPU system, refer to the following.

☞ Page 12-13, Section 12.4

Remark

1. The function version of the intelligent function module can be checked on the rating plate on the module.
2. For the maximum number of intelligent function modules, refer to the following.
   ☞ Page 11-14, Section 11.3
3. For how to enable the "I/O sharing when using Multiple CPUs" for a programmable controller CPU or Motion CPU, refer to the manual for each module.

11.3 Precautions for System Configuration

This section explains restrictions applied when a C Controller module is set as CPU No.1 in a multiple CPU system.

(1) Restrictions on the number of mountable modules

The numbers of the following modules are restricted when they are used in a multiple CPU system.

Use them within the allowable ranges.

Table 11.6 Number of mountable modules

Product	Model	Number of mountable modules per system		Number of mountable modules per CPU	
Q series CC-Link IE controller network module	• QJ71GP21-SX • QJ71GP21S-SX	Up to 4	Up to 4	Up to 4	Up to 4
Q series MELSECNET/H module	• QJ71BR11 • QJ71LP21-25 • QJ71LP21S-25 • QJ71LP21G • QJ71LP21GE	Up to a total of 4 modules in PLC to PLC network		Up to a total of 4 modules in PLC to PLC network	
Q series CC-Link system master/ local module	• QJ61BT11 • QJ61BT11N	Up to 64 (C Controller module can control up to 8 modules.)		Up to 8	
Interrupt module	• QI60 • QX40H[*2] • QX70H[*2] • QX80H[*2] • QX90H[*2]	Up to 4[*1] (C Controller module can control only 1 module)		1[*1]	
GOT	• GOT1000 series (only when in bus connection)	Up to 5		Up to 5	

[* 1] When the interrupt event setting is specified in the following C Controller, up to 64 modules can be inserted.
• Q12DCCPU-V with a serial number whose first five digits are "12042" or later.

[* 2] The module is available only when the interrupt module is selected by setting the function selector switch (SW2) to OFF.

Remark

For a module whose quantity is restricted when CPU No.1 is a programmable controller CPU, refer to the following.

☞ QCPU User's Manual (Multiple CPU System)

For the module whose quantity is restricted when CPU No.1 is Q24DHCCPU-V/-VG/-LS or Q12DCCPU-V (Extended mode), refer to the following.

☞ MELSEC-Q C Controller Module User's Manual

(2) Combinations of power supply modules and base units

Some restrictions are applied to the combinations of power supply modules and base units.

(☞ Page 11-2, Section 11.1.1)

(3) Precautions for GOT connection

The GOTs that can be used for the C Controller module are the same as those usable in a single CPU system. (☞ Page 2-29, Section 2.4 (6))

(4) Precautions for MT Developer (Version 00Y or earlier) connection in a multiple CPU system

When a multiple CPU system includes a C Controller module as CPU No.1 and a Motion CPU, a communication test of MT Developer (Version 00Y or earlier) are cannot be performed. (The Motion CPU communication test error (error code: 12288) will occur.)

To perform the communication test from MT Developer, use either of the following methods.

- Use MT Developer of Version 00Z or later.
- When using MT Developer of Version 00Y or earlier, modify the multiple CPU system configuration so that CPU No.1 is a programmable controller CPU.

(5) Time adjustment and time notification of GOT

In a multiple CPU system configuration, when the time adjustment or notification is performed in GOT connected to C Controller modules for CPU No. 2 to No. 4, it is executed to CPU No. 1.

In this case, to match the clock data between C Controller modules for CPU No. 2 to No. 4 and GOT, use the multiple CPU clock synchronization function.

☞ Page 4-33, Section 4.7.1

CHAPTER 12 CONCEPT OF MULTIPLE CPU SYSTEM

12.1 Mounting Positions of CPU Modules

Table 12.1 to Table 12.3 show applicable combinations of CPU modules in a multiple system where a C Controller module is CPU No.1.

Remark

When a programmable controller CPU is set as CPU No.1 in a multiple CPU system, refer to the following.

☞ QCPU User's Manual (Multiple CPU System)

Table 12.1 When using the main base unit (Q3□B)

CPU No.1 C Controller module	Number of CPU modules mountable as CPU No.2 and higher							Max. No. of mountable modules (including CPU No.1)	Reference
	High Performance model QCPU/ Process CPU/ Universal Model QCPU	Motion CPU		PC CPU module	C Controller module				
		Q172CPUN Q172CPUN-T Q173CPUN Q173CPUN-T Q172HCPU Q172HCPU-T Q173HCPU Q173HCPU-T	Q172DCPU Q173DCPU		Q12DCCPU-V	Q06CCPU-V Q06CCPU-V-B	Q24DHCCPU-V/-LS		
Q12DCCPU-V	—	—	—	—	0 to 3	0 to 3	0 to 1	4	(1) in this section
Q06CCPU-V Q06CCPU-V-B	—	0 to 3	—	—	0 to 3	0 to 3	0 to 1	4	(2) in this section

— : N/A

Table 12.2 When using the slim type main base unit (Q3□SB)

CPU No.1 C Controller module	Number of CPU modules mountable as CPU No.2 and higher							Max. No. of mountable modules (including CPU No.1)	Reference
	High Performance model QCPU/ Process CPU/ Universal Model QCPU	Motion CPU		PC CPU module	C Controller module				
		Q172CPUN Q172CPUN-T Q173CPUN Q173CPUN-T Q172HCPU Q172HCPU-T Q173HCPU Q173HCPU-T	Q172DCPU Q173DCPU		Q12DCCPU-V	Q06CCPU-V Q06CCPU-V-B	Q24DHCCPU-V/-LS		
Q12DCCPU-V	—	—	—	—	0 to 1	0 to 1	—	2	(1) in this section
Q06CCPU-V Q06CCPU-V-B	—	—	—	—	0 to 1	0 to 1	—	2	(2) in this section

— : N/A

Table 12.3 When using the multiple CPU high speed main base unit (Q3□DB)

CPU No.1 C Controller module	Number of CPU modules mountable as CPU No.2 and higher							Max. No. of mountable modules (including CPU No.1)	Reference
	High Performance model QCPU/ Process CPU/ Universal model QCPU	Motion CPU		PC CPU module	C Controller module				
		Q172CPUN Q172CPUN-T Q173CPUN Q173CPUN-T Q172HCPU Q172HCPU-T Q173HCPU Q173HCPU-T	Q172DCPU Q173DCPU		Q12DCCPU-V	Q06CCPU-V Q06CCPU-V-B	Q24DHCCPU-V/-LS		
Q12DCCPU-V	—	—	0 to 3	—	0 to 3	0 to 3	0 to 1	4	(1) in this section
Q06CCPU-V Q06CCPU-V-B	—	—	—	—	0 to 3	0 to 3	0 to 1	4	(2) in this section

— : N/A

(1) When CPU No.1 is the Q12DCCPU-V

The positions of respective CPU modules are shown in Table 12.4.

(a) Mounting position of a C Controller module

Up to four C Controller modules can be mounted on the CPU slot (the slot on the right of the power supply module) and Slot 0 to Slot 2 of the main base unit.

(b) Mounting position of a Motion CPU

1) Up to three Motion CPUs (Q172DCPU and Q173DCPU) can be mounted on Slot 0 to Slot 2.

2) No Motion CPU can be used when the main base unit is the Q3□B or Q3□SB.

(c) "CPU (Empty)" setting

Empty slots can be reserved for CPU modules that will be added in the future.
Set the number of CPU modules including the empty slots in "Number of CPU" in the <<Multiple CPU settings>> tab of C Controller setting utility.
After this, set the slot(s) on the right of those for the mounted CPU module(s) to "CPU (Empty)" in the <<I/O assignment settings>> tab of C Controller setting utility.

(Example) When the number of CPUs is set to 4 in the Multiple CPU settings and two C Controller modules and one Motion CPU are mounted:
Install the C Controller modules to the CPU slot and Slot 0, and the Motion CPU to Slot 1, and then set "CPU (Empty)" to Slot 2.

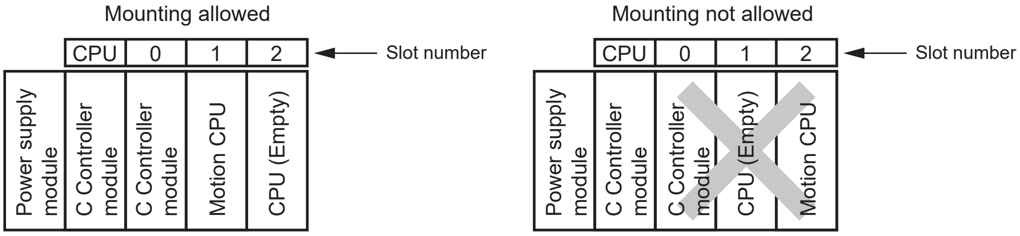


Figure 12.1 "CPU (Empty)" setting

☒POINT

When a C Controller module is used, "CPU (Empty)" cannot be set for a slot located between CPU modules.
Since no priority in the mounting order is provided for C Controller modules and Motion CPUs, any one of them can be installed to an empty slot without shifting the rightmost CPU module to the right.

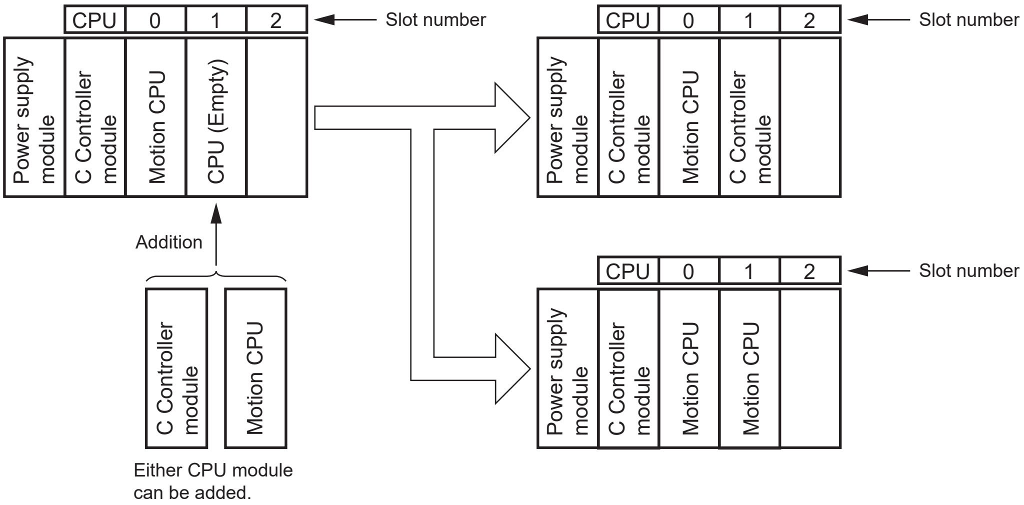


Figure 12.2 Adding a CPU module

Table 12.4 CPU module positions (excluding Q24DHCCPU-V/-LS)

CPU | 0 | 1 | 2 : Slot number

No. of CPUs[*1]	CPU module position		
2	Power supply module CPU: Q12DCCPU-V 0: C Controller module 1: 2:	Power supply module CPU: Q12DCCPU-V 0: Motion CPU[*2] 1: 2:	—
3	Power supply module CPU: Q12DCCPU-V 0: C Controller module 1: C Controller module 2:	Power supply module CPU: Q12DCCPU-V 0: Motion CPU[*2] 1: C Controller module 2:	Power supply module CPU: Q12DCCPU-V 0: C Controller module 1: Motion CPU[*2] 2:
	Power supply module CPU: Q12DCCPU-V 0: Motion CPU[*2] 1: Motion CPU[*2] 2:	—	—
4	Power supply module CPU: Q12DCCPU-V 0: C Controller module 1: C Controller module 2: C Controller module	Power supply module CPU: Q12DCCPU-V 0: Motion CPU[*2] 1: C Controller module 2: C Controller module	Power supply module CPU: Q12DCCPU-V 0: C Controller module 1: Motion CPU[*2] 2: C Controller module
	Power supply module CPU: Q12DCCPU-V 0: C Controller module 1: C Controller module 2: Motion CPU[*2]	Power supply module CPU: Q12DCCPU-V 0: Motion CPU[*2] 1: Motion CPU[*2] 2: C Controller module	Power supply module CPU: Q12DCCPU-V 0: Motion CPU[*2] 1: C Controller module 2: Motion CPU[*2]
	Power supply module CPU: Q12DCCPU-V 0: C Controller module 1: Motion CPU[*2] 2: Motion CPU[*2]	Power supply module CPU: Q12DCCPU-V 0: Motion CPU[*2] 1: Motion CPU[*2] 2: Motion CPU[*2]	—

[*1] No. of CPUs indicates the value set in "Number of CPUs" in the <<Multiple CPU settings>> tab of C Controller setting utility.
For programmable controller CPUs and Motion CPUs, refer to the manual for the respective module.

[*2] Only Q172DCPU and Q173DCPU can be used.

Table 12.5 CPU module positions (including Q24DHCCPU-V/-LS)

CPU	0	1	2	3	4	: Slot number

<table>
<tr><th>No. of CPUs[*1]</th><th colspan="3">CPU module position</th></tr>
<tr><td>2</td>
<td><table><tr><td></td><td>CPU</td><td>0</td><td>1</td><td>2</td></tr><tr><td colspan="2">Power supply module</td><td>Q12DCCPU-V</td><td colspan="2">Q24DHCCPU-V/-LS</td></tr></table></td>
<td>—</td>
<td>—</td></tr>
<tr><td>3</td>
<td><table><tr><td></td><td>CPU</td><td>0</td><td>1</td><td>2</td><td>3</td></tr><tr><td colspan="2">Power supply module</td><td>Q12DCCPU-V</td><td>C Controller module[*3]</td><td colspan="2">Q24DHCCPU-V/-LS</td></tr></table></td>
<td><table><tr><td></td><td>CPU</td><td>0</td><td>1</td><td>2</td><td>3</td></tr><tr><td colspan="2">Power supply module</td><td>Q12DCCPU-V</td><td>Motion CPU[*2]</td><td colspan="2">Q24DHCCPU-V/-LS</td></tr></table></td>
<td>—</td></tr>
<tr><td rowspan="2">4</td>
<td><table><tr><td></td><td>CPU</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td></tr><tr><td colspan="2">Power supply module</td><td>Q12DCCPU-V</td><td>C Controller module[*3]</td><td>C Controller module[*3]</td><td colspan="2">Q24DHCCPU-V/-LS</td></tr></table></td>
<td><table><tr><td></td><td>CPU</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td></tr><tr><td colspan="2">Power supply module</td><td>Q12DCCPU-V</td><td>C Controller module[*3]</td><td>Motion CPU[*2]</td><td colspan="2">Q24DHCCPU-V/-LS</td></tr></table></td>
<td><table><tr><td></td><td>CPU</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td></tr><tr><td colspan="2">Power supply module</td><td>Q12DCCPU-V</td><td>Motion CPU[*2]</td><td>C Controller module[*3]</td><td colspan="2">Q24DHCCPU-V/-LS</td></tr></table></td></tr>
<tr>
<td><table><tr><td></td><td>CPU</td><td>0</td><td>1</td><td>2</td><td>3</td><td>4</td></tr><tr><td colspan="2">Power supply module</td><td>Q12DCCPU-V</td><td>Motion CPU[*2]</td><td>Motion CPU[*2]</td><td colspan="2">Q24DHCCPU-V/-LS</td></tr></table></td>
<td>—</td>
<td>—</td></tr>
</table>

[*1] No. of CPUs indicates the value set in "Number of CPUs" in the <<Multiple CPU settings>> tab of C Controller setting utility.
For programmable controller CPUs and Motion CPUs, refer to the manual for the respective module.

[*2] Only Q172DCPU and Q173DCPU can be used.

[*3] Excluding Q24DHCCPU-V/-LS.

(2) When CPU No.1 is the Q06CCPU-V(-B)

The positions of respective CPU modules are shown in Table 12.6.

(a) Mounting position of a C Controller module

Up to four C Controller modules can be mounted on the CPU slot (the slot on the right of the power supply module) and Slot 0 to Slot 2 of the main base unit.

(b) Mounting position of a Motion CPU

1) Up to three Motion CPUs (except the Q172DCPU and Q173DCPU) can be mounted on Slot 0 to Slot 2

2) No Motion CPU can be used when the main base unit is the Q3□SB or Q3□DB.

(c) "CPU (Empty)" setting

For details, refer to the following section.

☞Page 12-2, Section 12.1 (1) (c)

Table 12.6 CPU module positions (excluding Q24DHCCPU-V/-LS)

CPU | 0 | 1 | 2 : Slot number

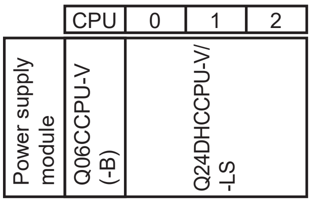
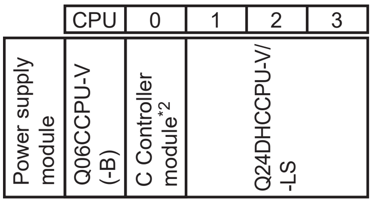
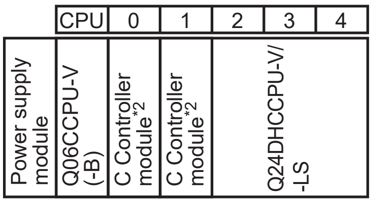
No. of CPUs*1	CPU module position		
2	Power supply module CPU: Q06CCPU-V(-B) 0: C Controller module 1: 2:	Power supply module CPU: Q06CCPU-V(-B) 0: Motion CPU*2 1: 2:	—
3	Power supply module CPU: Q06CCPU-V(-B) 0: C Controller module 1: C Controller module 2:	Power supply module CPU: Q06CCPU-V(-B) 0: Motion CPU*2 1: Q06CCPU-V(-B) 2:	Power supply module CPU: Q06CCPU-V(-B) 0: Q06CCPU-V(-B) 1: Motion CPU*2 2:
	Power supply module CPU: Q06CCPU-V(-B) 0: Motion CPU*2 1: Motion CPU*2 2:	—	—
4	Power supply module CPU: Q06CCPU-V(-B) 0: C Controller module 1: C Controller module 2: C Controller module	Power supply module CPU: Q06CCPU-V(-B) 0: Motion CPU*2 1: Q06CCPU-V(-B) 2: Q06CCPU-V(-B)	Power supply module CPU: Q06CCPU-V(-B) 0: Q06CCPU-V(-B) 1: Motion CPU*2 2: Q06CCPU-V(-B)
	Power supply module CPU: Q06CCPU-V(-B) 0: Q06CCPU-V(-B) 1: Q06CCPU-V(-B) 2: Motion CPU*2	Power supply module CPU: Q06CCPU-V(-B) 0: Motion CPU*2 1: Motion CPU*2 2: Q06CCPU-V(-B)	Power supply module CPU: Q06CCPU-V(-B) 0: Motion CPU*2 1: Q06CCPU-V(-B) 2: Motion CPU*2
	Power supply module CPU: Q06CCPU-V(-B) 0: Q06CCPU-V(-B) 1: Motion CPU*2 2: Motion CPU*2	Power supply module CPU: Q06CCPU-V(-B) 0: Motion CPU*2 1: Motion CPU*2 2: Motion CPU*2	—

* 1 No. of CPUs indicates the value set in "Number of CPUs" in the <<Multiple CPU settings>> tab of C Controller setting utility.
For programmable controller CPUs and Motion CPUs, refer to the manual for the respective module.

* 2 Q172DCPU or Q173DCPU cannot be used.

Table 12.7 CPU module positions (including Q24DHCCPU-V/-LS)

CPU	0	1	2	3	4	: Slot number

No. of CPUs[*1]	CPU module position		
2	 CPU 0 1 2 Power supply module Q06CCPU-V (-B) Q24DHCCPU-V/ -LS	—	—
3	 CPU 0 1 2 3 Power supply module Q06CCPU-V (-B) C Controller module*2 Q24DHCCPU-V/ -LS	—	—
4	 CPU 0 1 2 3 4 Power supply module Q06CCPU-V (-B) C Controller module*2 C Controller module*2 Q24DHCCPU-V/ -LS	—	—

[*1] No. of CPUs indicates the value set in "Number of CPUs" in the <<Multiple CPU settings>> tab of C Controller setting utility.
For programmable controller CPUs and Motion CPUs, refer to the manual for the respective module.

[*2] Excluding Q24DHCCPU-V/-LS.

12.2 CPU No. of CPU Module

(1) CPU No. assignment

CPU numbers are used to identify each CPU module configured in a multiple CPU system.

CPU No.1 is assigned to the CPU slot, and CPU No.2, No.3, and No.4 are assigned to the slots starting from the right of the CPU slot.

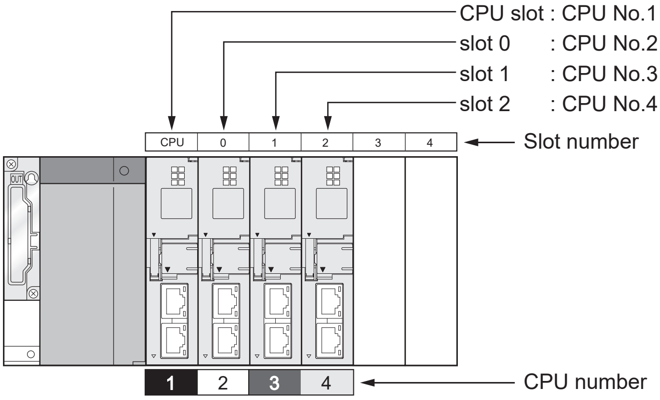
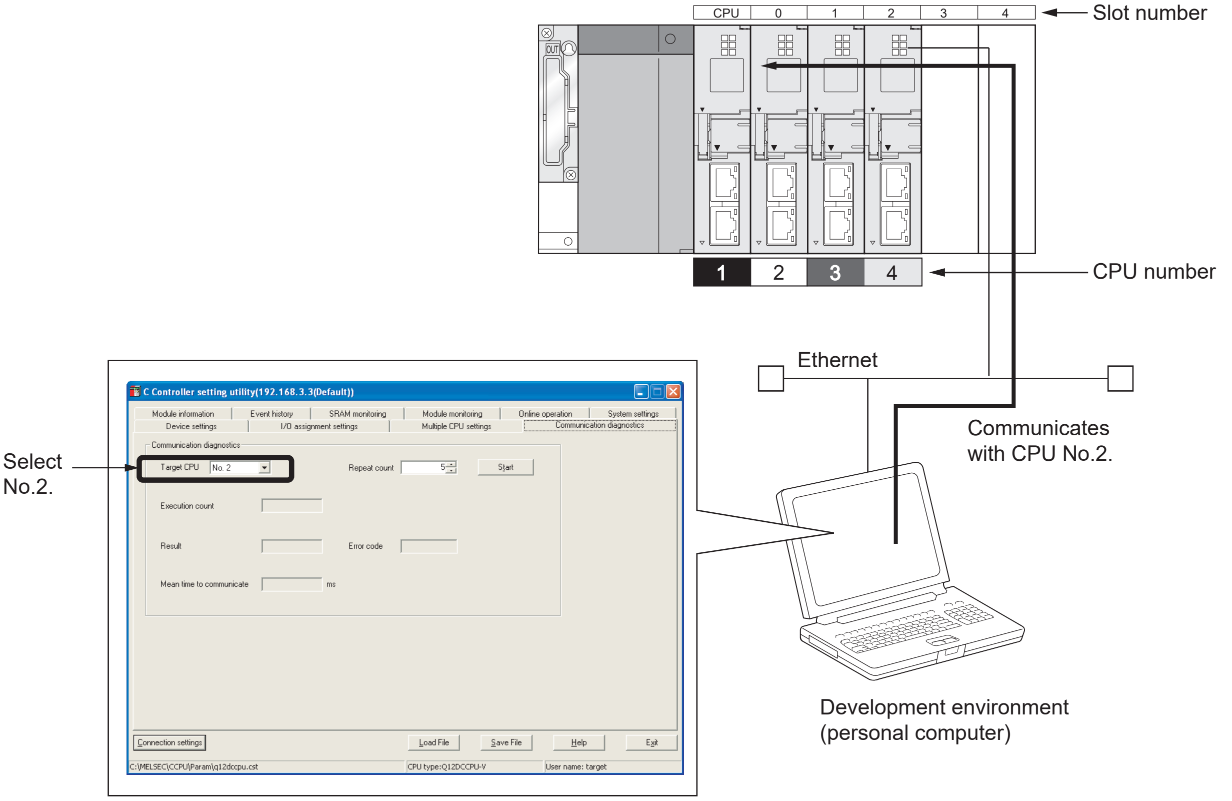


Figure 12.3 CPU No. assignment

The following are CPU No. application examples.

- Diagnosing the communication from the <<Communication diagnostics>> tab of C Controller setting utility



Communication diagnostics

Target CPU: No. 2

Repeat count: 5

Execution count:

Result:

Error code:

Mean time to communicate: ms

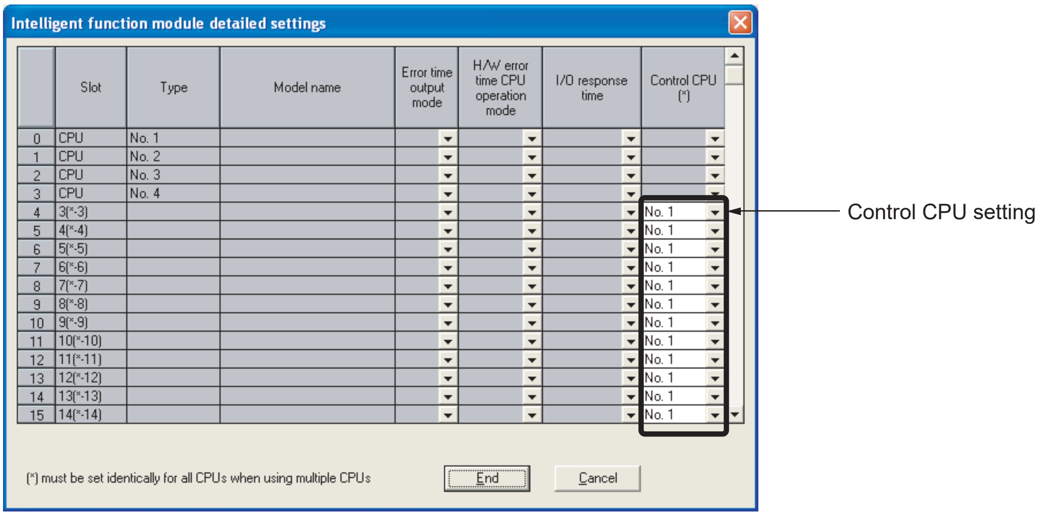
C:\MELSEC\CCPU\Param\q12dccpu.cst

CPU type:Q12DCCPU-V

User name: target

Figure 12.4 Target setting in C Controller setting utility

- Setting control CPU in the <<I/O assignment settings>> tab of C Controller setting utility (☞ Page 14-7, Section 14.1.6)



	Slot	Type	Model name	Error time output mode	H/W error time CPU operation mode	I/O response time	Control CPU (*)
0	CPU	No. 1					
1	CPU	No. 2					
2	CPU	No. 3					
3	CPU	No. 4					
4	3(*-3)						No. 1
5	4(*-4)						No. 1
6	5(*-5)						No. 1
7	6(*-6)						No. 1
8	7(*-7)						No. 1
9	8(*-8)						No. 1
10	9(*-9)						No. 1
11	10(*-10)						No. 1
12	11(*-11)						No. 1
13	12(*-12)						No. 1
14	13(*-13)						No. 1
15	14(*-14)						No. 1

(*) must be set identically for all CPUs when using multiple CPUs

Figure 12.5 Control CPU setting

Remark

For how to check host CPU No. of a programmable controller CPU or Motion CPU, refer to the manual for the relevant CPU module.

12.3 I/O Number Assignment

In the multiple CPU system, I/O numbers are used for communication between each CPU module and I/O modules or intelligent function modules, and for communication between different CPU modules.

12.3.1 I/O number assignment for the module

The position (slot) of the I/O number 00H in the multiple CPU system is different from that in the single CPU system.
Note that the order of I/O number assignment, I/O numbers for each slot and empty slots are the same between both systems. (☞CHAPTER 7)

(1) Position of I/O number "00H"

(a) Slots occupied by CPU modules
In the multiple CU system, CPU modules occupy the slots of the number set in the <<Multiple CPU settings>> tab of C Controller setting utility.

(b) Mounting positions of I/O modules and intelligent function modules
Mount I/O modules and intelligent function modules from the right of the slots occupied by CPU modules.

(c) I/O number assignment
Set "00H" to the I/O module or intelligent function module mounted next to the rightmost CPU module, and then assign consecutive numbers to the right.
Example: When the number of CPUs is set to 2

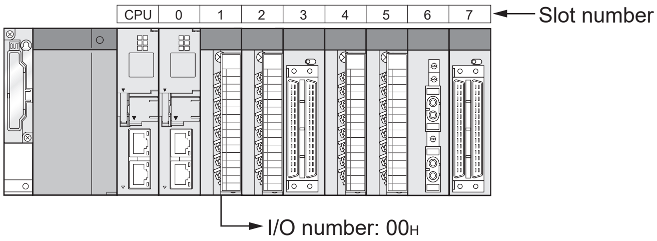


Figure 12.6 Position of I/O number "00H"

Remark

When the number of CPU modules to be mounted on the main base unit is less than the value set in the <<Multiple CPU settings>> tab of C Controller setting utility, set "CPU (Empty)" for any empty slots.
For the "CPU (Empty)" setting, refer to the following.
☞ Page 12-1, Section 12.1

12.3.2 I/O number of each CPU module

In a multiple CPU system configuration, a specific I/O number is assigned to each CPU module for identification.
Each of these I/O number is fixed to the corresponding slot and cannot be changed in the <<I/O assignment settings>> tab of C Controller setting utility.
Table 12.8 shows the I/O numbers assigned to the CPU modules in a multiple CPU system configuration.

Table 12.8 I/O numbers of CPU modules

CPU module position	CPU slot	Slot 0	Slot 1	Slot 2
Start I/O number	3E00H	3E10H	3E20H	3E30H

The C Controller module uses the I/O numbers of the CPU modules in data communications using the CPU shared memory. (☞ Page 13-12, Section 13.3)

12.4 Access Ranges Between a CPU Module and Other Modules

12.4.1 Access to controlled modules

A CPU module can read and write buffer memory data of the I/O modules and intelligent function modules that are controlled by the CPU module. (☞ Page 4-4, Section 4.2)

12.4.2 Access to non-controlled modules

CPU modules can read the following data based on the multiple CPU settings parameters.

- Input (X) on/off data of non-controlled modules
- Output (Y) on/off data of other CPUs

This allows on/off data of the following modules controlled by another CPU to be used as an interlock for the host CPU, and also the output status of external devices controlled by other CPUs can be checked.

- Input module
- I/O combined module
- Intelligent function module

Data in the buffer memory of a non-controlled intelligent function module can be read out regardless of the parameters set in the Multiple CPU settings.

However, a non-control CPU cannot perform the following processing on non-controlled modules.

- Output on/off data to output modules
- Output on/off data to I/O combined modules
- Output on/off data to intelligent function modules
- Write to the buffer memory of intelligent function modules

Table 12.9 indicates accessibility to non-controlled modules in the multiple CPU system.

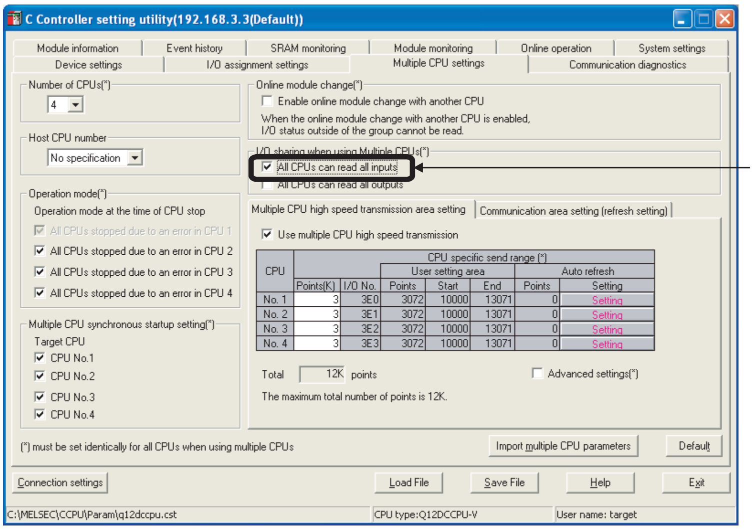
Table 12.9 Accessibility to non-controlled modules

Access target		I/O sharing when using Multiple CPUs	
		Disabled (Unchecked)	Enabled (Checked)
Input (X)		×	○
Output (Y)	Read	×	○
	Write	×	×
Buffer memory of intelligent function module	Read	○	○
	Write	×	×

○ : Accessible, × : N/A

(1) Input (X) loading

Whether to load inputs from input modules and intelligent function modules controlled by other CPUs is determined by the "I/O sharing when using Multiple CPUs" parameter in the <<Multiple CPU settings>> tab of C Controller setting utility.



C Controller setting utility(192.168.3.3(Default))

Module information | Event history | SRAM monitoring | Module monitoring | Online operation | System settings
Device settings | I/O assignment settings | Multiple CPU settings | Communication diagnostics

Number of CPUs(*): 4

Host CPU number: No specification

Operation mode(*)
Operation mode at the time of CPU stop
☑ All CPUs stopped due to an error in CPU 1
☑ All CPUs stopped due to an error in CPU 2
☑ All CPUs stopped due to an error in CPU 3
☑ All CPUs stopped due to an error in CPU 4

Multiple CPU synchronous startup setting(*)
Target CPU
☑ CPU No.1
☑ CPU No.2
☑ CPU No.3
☑ CPU No.4

Online module change(*)
☐ Enable online module change with another CPU
When the online module change with another CPU is enabled, I/O status outside of the group cannot be read.

I/O sharing when using Multiple CPUs(*)
☑ All CPUs can read all inputs
☐ All CPUs can read all outputs

Multiple CPU high speed transmission area setting | Communication area setting (refresh setting)

☑ Use multiple CPU high speed transmission

CPU	CPU specific send range (*)						
			User setting area			Auto refresh	
	Points(K)	I/O No.	Points	Start	End	Points	Setting
No. 1	3	3E0	3072	10000	13071	0	Setting
No. 2	3	3E1	3072	10000	13071	0	Setting
No. 3	3	3E2	3072	10000	13071	0	Setting
No. 4	3	3E3	3072	10000	13071	0	Setting

Total 12K points ☐ Advanced settings(*)

The maximum total number of points is 12K.

(*) must be set identically for all CPUs when using multiple CPUs

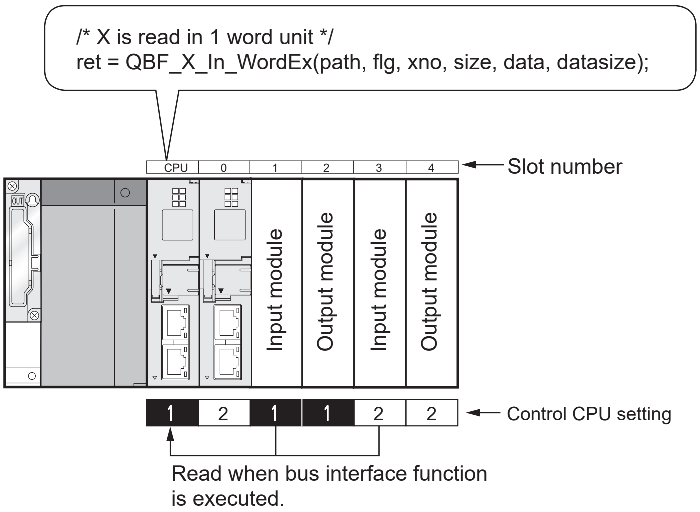
C:\MELSEC\CCPU\Param\q12dccpu.cst | CPU type:Q12DCCPU-V | User name: target

I/O sharing when using Multiple CPUs
☑All CPUs can read all inputs:
"All CPUs can read all inputs" has been set
☐All CPUs can read all inputs:
"Not all CPUs can read all inputs" has been set

Figure 12.7 I/O sharing when using multiple CPUs (Input loading)

(a) When "All CPU s can read all inputs" is selected

1) On/off data are loaded from input modules or intelligent function modules controlled by other CPUs to the input (X) of the host CPU by execution of a bus interface function.



```
/* X is read in 1 word unit */
ret = QBF_X_In_WordEx(path, flg, xno, size, data, datasize);
```

Figure 12.8 When CPU No.1 loads inputs

2) Input (X) data are loaded from the modules indicated in Table 12.10, which are mounted on a main base unit or extension base units.

Table 12.10 Modules from which inputs can be loaded

Type setting in the <<I/O assignment settings>> tab of C Controller setting utility	Mounted module
None	Input module
	High-speed input module
	I/O combined module*1
	Intelligent function module
Input Hi. input I/O mix	Input module
	High-speed input module
	I/O combined module*1
Intelli.	Intelligent function module

* 1 When inputs (X) are loaded from the QX48Y57 I/O combined module, data of Xn8 to XnF that are assigned as an output area are all off.

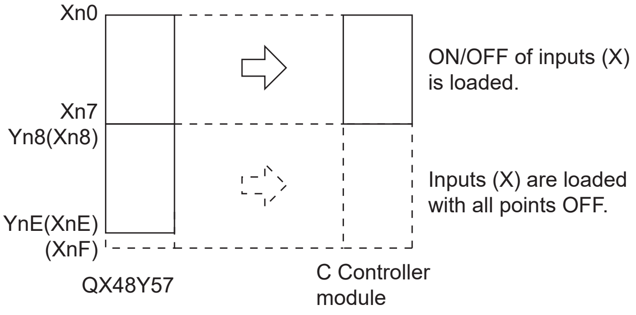

Xn0
Xn7
Yn8(Xn8)
YnE(XnE)
(XnF)
QX48Y57
C Controller module
ON/OFF of inputs (X) is loaded.
Inputs (X) are loaded with all points OFF.

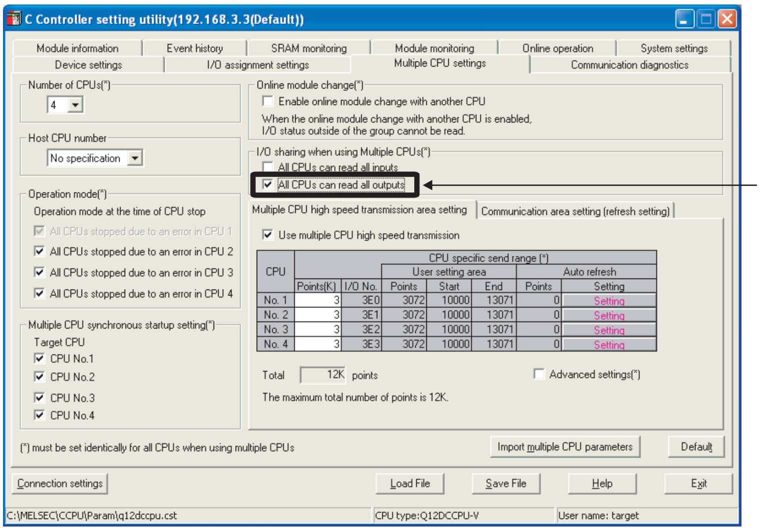
Figure 12.9 Loading inputs (X) from QX48Y57

3) Data assigned to empty slots and cyclic data controlled by other CPUs on the following network cannot be read out directly from the refresh target devices. To read the following cyclic data, assign the refresh target device to the auto refresh area in the CPU shared memory.
    - CC-Link IE controller network
    - MELSECNET/H
    - CC-Link, etc.

(b) When "All CPUs can read all inputs" is deselected
On/off data cannot be loaded from I/O modules and intelligent function modules controlled by other CPUs (they remain off).

(2) Output (Y) loading

Whether to load outputs from output modules and intelligent function modules controlled by other CPUs is determined by the "I/O sharing when using Multiple CPUs" parameter in the <<Multiple CPU settings>> tab of C Controller setting utility.



C Controller setting utility(192.168.3.3(Default))

Number of CPUs(*): 4

Host CPU number: No specification

Operation mode(*)
Operation mode at the time of CPU stop
☑ All CPUs stopped due to an error in CPU 1
☑ All CPUs stopped due to an error in CPU 2
☑ All CPUs stopped due to an error in CPU 3
☑ All CPUs stopped due to an error in CPU 4

Multiple CPU synchronous startup setting(*)
Target CPU
☑ CPU No.1
☑ CPU No.2
☑ CPU No.3
☑ CPU No.4

Online module change(*)
☐ Enable online module change with another CPU
When the online module change with another CPU is enabled,
I/O status outside of the group cannot be read.

I/O sharing when using Multiple CPUs(*)
☐ All CPUs can read all inputs
☑ All CPUs can read all outputs

Multiple CPU high speed transmission area setting | Communication area setting (refresh setting)

☑ Use multiple CPU high speed transmission

CPU	CPU specific send range (*)						
			User setting area			Auto refresh	
	Points(K)	I/O No.	Points	Start	End	Points	Setting
No. 1	3	3E0	3072	10000	13071	0	Setting
No. 2	3	3E1	3072	10000	13071	0	Setting
No. 3	3	3E2	3072	10000	13071	0	Setting
No. 4	3	3E3	3072	10000	13071	0	Setting

Total 12K points   ☐ Advanced settings(*)

The maximum total number of points is 12K.

(*) must be set identically for all CPUs when using multiple CPUs

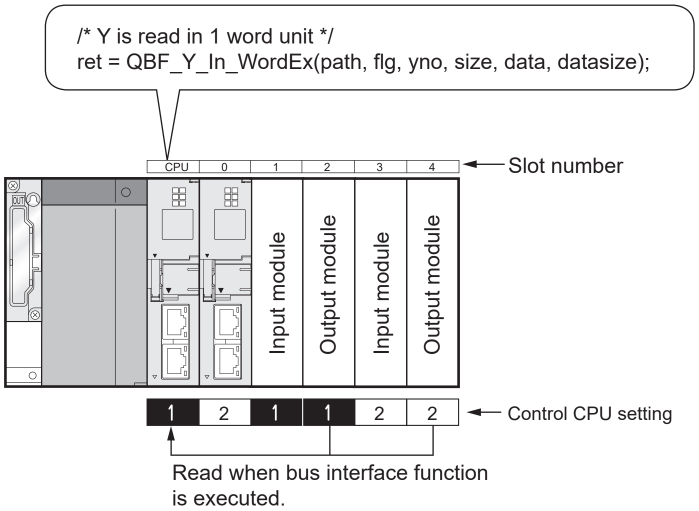
C:\MELSEC\CCPU\Param\q12dccpu.cst | CPU type:Q12DCCPU-V | User name: target

I/O sharing when using Multiple CPUs
☑All CPUs can read all outputs:
"All CPUs can read all outputs" has been set
☐All CPUs can read all outputs:
"Not all CPUs can read all outputs" has been set

Figure 12.10 I/O sharing when using multiple CPUs (Output loading)

(a) When "All CPU s can read all outputs" is selected

1) On/off data are loaded from output modules or intelligen function modules controlled by other CPUs to the output (Y) of the host CPU by execution of a bus interface function.



```
/* Y is read in 1 word unit */
ret = QBF_Y_In_WordEx(path, flg, yno, size, data, datasize);
```

Figure 12.11 When CPU No.1 loads outputs

2) Output (Y) data are loaded from the modules indicated in Table 12.11, which are mounted on a main base unit and extension base units.

Table 12.11 Modules from which outputs are loaded

Type setting in the <<I/O assignment settings>> tab of C Controller setting utility	Mounted module
None	Output module
	I/O combined module
	Intelligent function module
Output	Output module
I/O mix	I/O combined module
Intelli.	Intelligent function module

3) Data assigned to empty slots and cyclic data controlled by other CPUs on the following network cannot be read out directly from the refresh target devices. To the following read cyclic data, assign the refresh target device to the auto refresh area in the CPU shared memory.
    - CC-Link IE controller network
    - MELSECNET/H
    - CC-Link, etc.

(b) When "All CPUs can read all outputs" is deselected
On/off data cannot be loaded from output (Y) modules and intelligent function modules controlled by other CPUs (they remain off).

(3) Output to output modules and intelligent function modules

On/off data cannot be output to non-controlled modules.

Even if turning on or off the output of an output module or intelligent function module controlled by another CPU is attempted from a C Controller module, no output is made to the output module or intelligent function module.

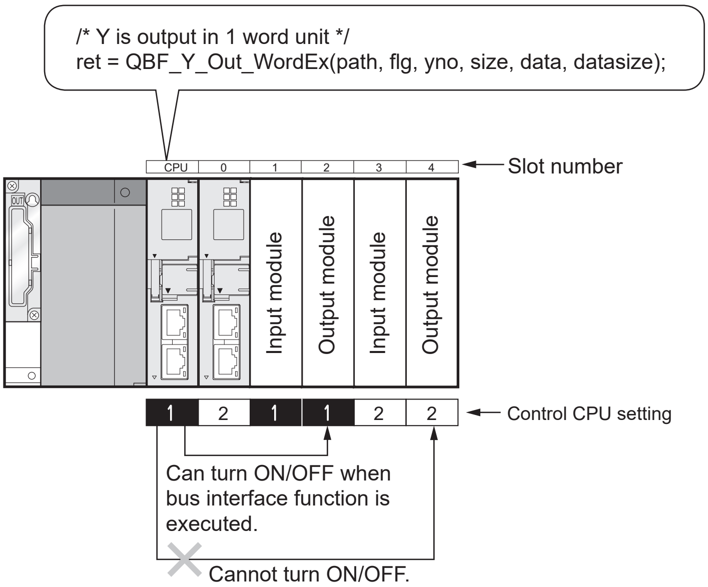

/* Y is output in 1 word unit */
ret = QBF_Y_Out_WordEx(path, flg, yno, size, data, datasize);
CPU
0
1
2
3
4
Slot number
Input module
Output module
Input module
Output module
1
2
1
1
2
2
Control CPU setting
Can turn ON/OFF when bus interface function is executed.
Cannot turn ON/OFF.

Figure 12.12 When CPU No.1 outputs data to modules

(4) Access to the buffer memory of an intelligent function module

(a) Reading from the buffer memory

Buffer memory data of intelligent function modules controlled by other CPUs can be read by using a bus interface function (QBF_FromBuf).

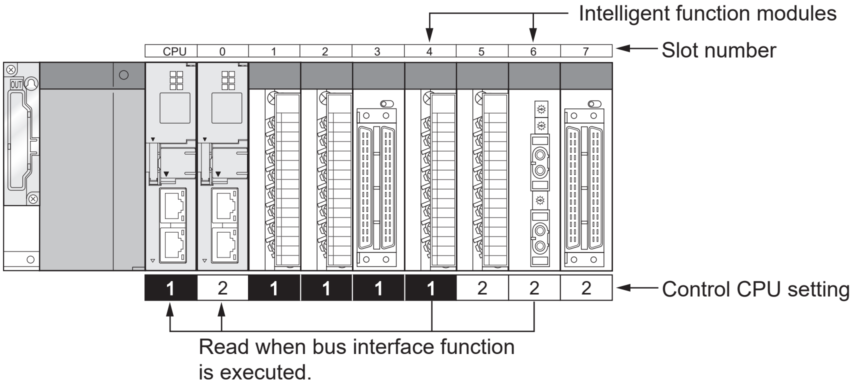

Intelligent function modules
CPU
0
1
2
3
4
5
6
7
Slot number
1
2
1
1
1
1
2
2
2
Control CPU setting
Read when bus interface function is executed.

Figure 12.13 Reading from intelligent function modules

(b) Writing to the buffer memory

Data cannot be written to the buffer memory of any intelligent function module controlled by another CPU.

If writing to an intelligent function module controlled by another CPU is attempted, a non-controlled module write error (return value: -28654) occurs.

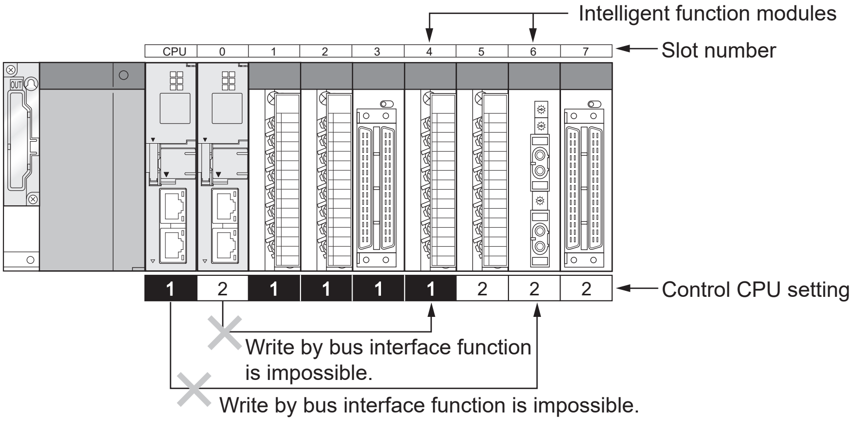

Intelligent function modules
Slot number
CPU
0
1
2
3
4
5
6
7
Control CPU setting
1
2
1
1
1
1
2
2
2
Write by bus interface function is impossible.
Write by bus interface function is impossible.

Figure 12.14 Writing to intelligent function modules

12.5 Access to Link Devices

Only the control CPU can access link devices of CC-Link IE controller network modules or MELSECNET/H modules by user programs.
A CC-Link IE controller network module or MELSECNET/H module that are controlled by another CPU is not accessible by a user program.
Attempting such access causes either of the following errors.

- Non-controlled module read error (return value:-28633)
- Non-controlled module write error (return value:-28654)

Note that, however, a CC-Link IE controller network module controlled by another CPU can be accessed by a user program if the access is made through a CC-Link IE controller network.
Also, a MELSECNET/H module controlled by another CPU is accessible by a user program if the access is made through a MELSECNET/H network.

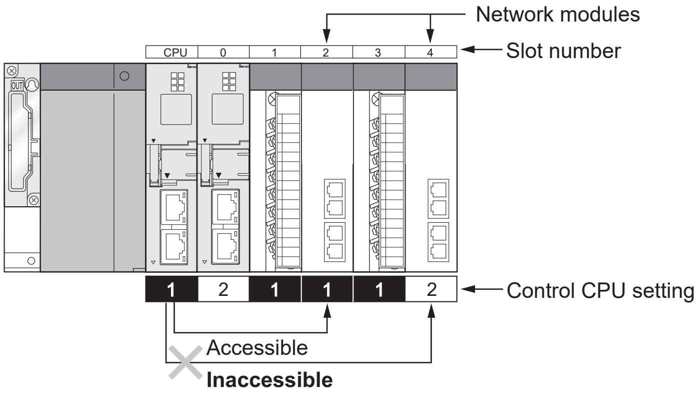


Figure 12.15 Access to link devices

Remark

For accessible ranges and devices of the MELSEC data link functions, refer to the following.
☞ C Controller Module User's Manual (Utility Operation, Programming)

12.6 Resetting CPU Modules

The entire multiple CPU system can be reset by resetting CPU No.1.
CPU modules other than CPU No.1, I/O modules, and intelligent function modules on the system will be all reset when CPU No.1 is reset.

When a stop error has occurred on any of the CPU modules on the multiple CPU system, reset CPU No.1 or restart the multiple CPU system (power on→off→on).

Resetting a CPU module that is in a stop error does not restore the system if the module is other than CPU No.1.
(Example) C Controller module

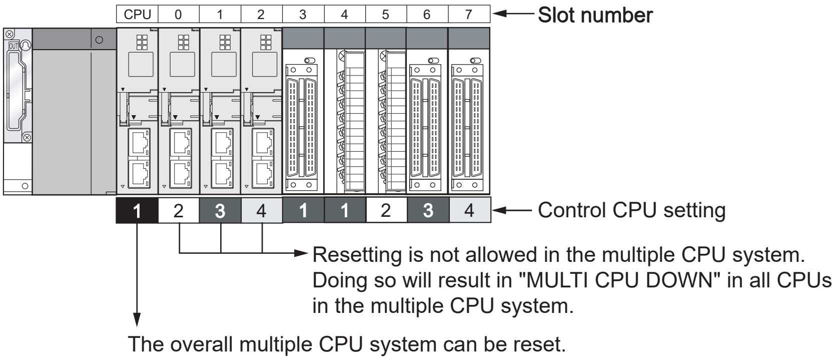

CPU
0
1
2
3
4
5
6
7
Slot number
1
2
3
4
1
1
2
3
4
Control CPU setting
Resetting is not allowed in the multiple CPU system.
Doing so will result in "MULTI CPU DOWN" in all CPUs
in the multiple CPU system.
The overall multiple CPU system can be reset.

Figure 12.16 Resetting of a multiple CPU system

☒**POINT**

1. In the multiple CPU system, CPU modules other than CPU No.1 cannot be reset individually.
   If any CPU module other than CPU No.1 is reset during operation of the multiple CPU system, a multi CPU error (error code: 7000, MULTI CPU DOWN) occurs in the other CPUs and the entire multiple CPU system stops.
   Note that an error other than the multi CPU error (error code: 7000, MULTI CPU DOWN) may stop the other CPU modules depending on timing of resetting a CPU module other than CPU No.1.
2. If any CPU module other than CPU No.1 is reset, a multi CPU error (error code: 7000, MULTI CPU DOWN) occurs regardless of the Operation mode setting (All CPUs stopped due to an error in CPU n/Continue) in the <<Multiple CPU settings>> tab of C Controller setting utility. (☞ Page 12-22, Section 12.7)

12.7 When a CPU Module Stop Error Occurs

The behavior of a multiple CPU system depends on whether a stop error occurs in CPU No.1 or in any of CPU No.2 to No.4.

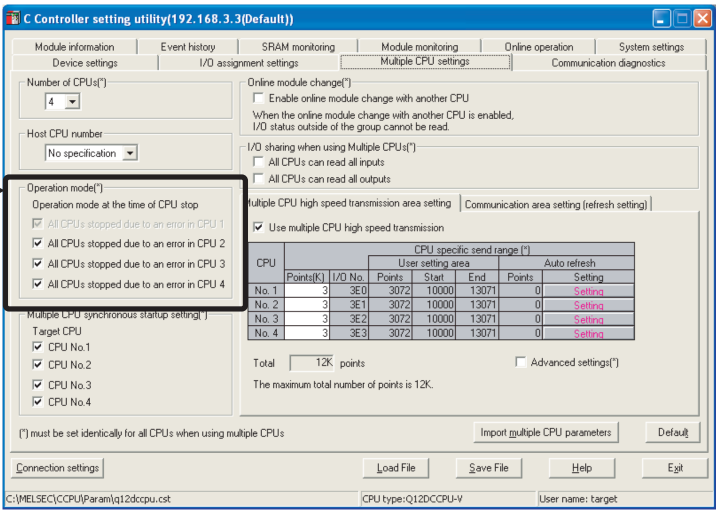
(1) When a stop error occurs in CPU No.1

If a stop error occurs in CPU No.1, the multi CPU error (error code: 7000, MULTI CPU DOWN) occurs in all of the other CPU modules, and the multiple CPU system stops.

(2) When a stop error occurs in other than CPU No.1

If a stop error occurs in other than CPU No.1, whether to stop the entire system or not depends on the "Operation mode" setting in the <<Multiple CPU settings>> tab of C Controller setting utility.
By default, all CPUs are set to be stopped in the event of a stop error.
In order not to stop all CPUs upon a stop error in a specific CPU module, remove the check mark of the relevant "All CPUs stopped due to an error in CPU n" box.

Operation mode
☑All CPUs stopped due to an error in CPU n:
"All CPUs stopped due to an error in CPU n" has been set
□All CPUs stopped due to an error in CPU n:
"No CPUs stopped due to an error in CPU n" has been set



Number of CPUs(*): 4

Host CPU number: No specification

Online module change(*)
□ Enable online module change with another CPU
When the online module change with another CPU is enabled, I/O status outside of the group cannot be read.

I/O sharing when using Multiple CPUs(*)
□ All CPUs can read all inputs
□ All CPUs can read all outputs

Operation mode(*)
Operation mode at the time of CPU stop
☑ All CPUs stopped due to an error in CPU 1
☑ All CPUs stopped due to an error in CPU 2
☑ All CPUs stopped due to an error in CPU 3
☑ All CPUs stopped due to an error in CPU 4

Multiple CPU synchronous startup setting(*)
Target CPU
☑ CPU No.1
☑ CPU No.2
☑ CPU No.3
☑ CPU No.4

☑ Use multiple CPU high speed transmission

CPU	CPU specific send range (*)						
			User setting area			Auto refresh	
	Points(K)	I/O No.	Points	Start	End	Points	Setting
No. 1	3	3E0	3072	10000	13071	0	Setting
No. 2	3	3E1	3072	10000	13071	0	Setting
No. 3	3	3E2	3072	10000	13071	0	Setting
No. 4	3	3E3	3072	10000	13071	0	Setting

Total 12K points
□ Advanced settings(*)
The maximum total number of points is 12K.

(*) must be set identically for all CPUs when using multiple CPUs

C:\MELSEC\CCPU\Param\q12dccpu.cst | CPU type:Q12DCCPU-V | User name: target

Figure 12.17 Operation setting for stop error

(a) When "All CPUs stopped due to an error in CPU n" is selected

If a stop error occurs in the CPU module corresponding to "All CPUs stopped due to an error in CPU n", a stop error or multi CPU error (error code: 7000, MULTI CPU DOWN) is detected in all of the other CPU modules, and the multiple CPU system stops.

(b) When "All CPUs stopped due to an error in CPU n" is deselected

If a stop error occurs in the CPU module corresponding to "All CPUs stopped due to an error in CPU n", a continue error or multi CPU error (error code: 7020, MULTI CPU ERROR) occurs in all of the other CPU modules, and the operation of the system continues.
In addition, if CPU No.n stops due to moderate error or lower, the other CPU modules continue the operation. However, if any critical error occurs in CPU No.n, the parameters are not valid, and all CPUs stop.

☒POINT

When a stop error is detected in a CPU, a multi CPU error (error code: 7000, MULTI CPU DOWN) occurs in the CPU.
Depending on the timing of error detection, "MULTI CPU DOWN" may be detected from a CPU that was placed in the "MULTI CPU DOWN" status due to the first stop error in another CPU.
For example, if a stop error occurred in CPU No.2 and CPU No.3 is halted due to the error, CPU No.1 may be halted due to the stop error in CPU No.3 depending on the timing of error detection.

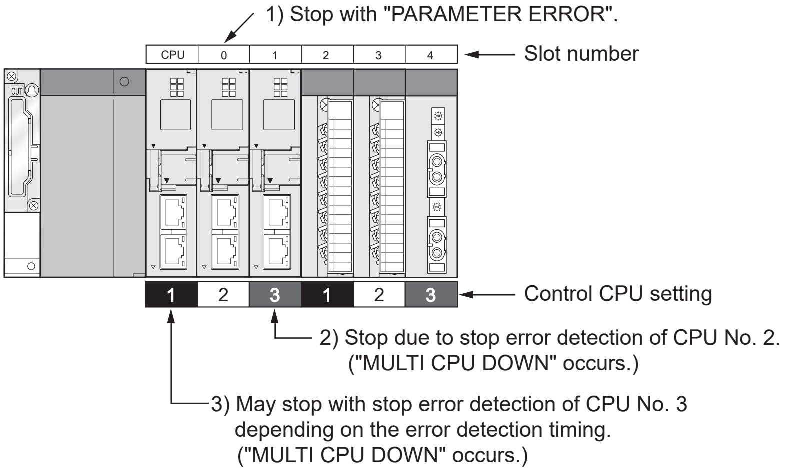
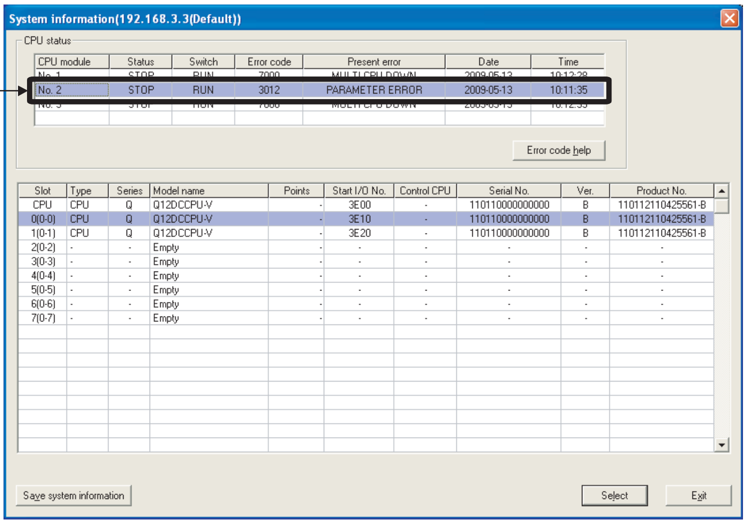

1) Stop with "PARAMETER ERROR".
CPU
0
1
2
3
4
Slot number
OUT
1
2
3
1
2
3
Control CPU setting
2) Stop due to stop error detection of CPU No. 2.
("MULTI CPU DOWN" occurs.)
3) May stop with stop error detection of CPU No. 3
depending on the error detection timing.
("MULTI CPU DOWN" occurs.)

Figure 12.18 Stop error detection

For this reason, CPU No. different from the one for the CPU where the first stop error occurred may be stored in the common error information area.
To restore the system, remove the cause of the error from the CPU stopped due to an error other than "MULTI CPU DOWN".
In Figure 12.19, the cause of the CPU No.2 error must be removed since the error is not "MULTI CPU DOWN".

Error other than
MULTI CPU DOWN


System information(192.168.3.3(Default))

CPU status

CPU module	Status	Switch	Error code	Present error	Date	Time
No. 1	STOP	RUN	7000	MULTI CPU DOWN	2009-05-13	10:12:28
No. 2	STOP	RUN	3012	PARAMETER ERROR	2009-05-13	10:11:35
No. 3	STOP	RUN	7000	MULTI CPU DOWN	2009-05-13	10:12:35

Slot	Type	Series	Model name	Points	Start I/O No.	Control CPU	Serial No.	Ver.	Product No.
CPU	CPU	Q	Q12DCCPU-V	-	3E00	-	110110000000000	B	110112110425561-B
0(0-0)	CPU	Q	Q12DCCPU-V	-	3E10	-	110110000000000	B	110112110425561-B
1(0-1)	CPU	Q	Q12DCCPU-V	-	3E20	-	110110000000000	B	110112110425561-B
2(0-2)	-	-	Empty	-	-	-	-	-	-
3(0-3)	-	-	Empty	-	-	-	-	-	-
4(0-4)	-	-	Empty	-	-	-	-	-	-
5(0-5)	-	-	Empty	-	-	-	-	-	-
6(0-6)	-	-	Empty	-	-	-	-	-	-
7(0-7)	-	-	Empty	-	-	-	-	-	-

Figure 12.19 Error display in System information

(3) Restoring the system

When a C Controller is CPU No.1, restore the system in the following procedure.

1) Check the No. of the faulty CPU and the error cause on the System information screen opened from the <<Module monitoring>> tab of C Controller setting utility.
2) Remove the cause.
3) Either reset CPU No.1 or restart the multiple CPU system (power supply on→off→on).

By resetting CPU No.1 or restarting the multiple CPU system, all the CPUs in the multiple CPU system are reset and the system is restored.

Remark

For restoration of the system where CPU No.1 is a programmable controller CPU, refer to the following.

☞ QCPU User's Manual (Multiple CPU System)

CHAPTER 13 COMMUNICATIONS BETWEEN CPU MODULES

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	○

This chapter explains how the C Controller module can intercommunicate with a programmable controller CPU or Motion CPU in a multiple CPU system.

Table 13.1 List of the functions available between C Controller module and programmable controller CPU or Motion CPU

Function	Refer to
Data communications by MELSEC data link functions	Page 13-2, Section 13.1
Interrupt from Another CPU	Page 13-3, Section 13.2
Data communications using CPU shared memory	Page 13-12, Section 13.3
Programmable controller remote control function	Page 13-42, Section 13.4
Sequence program control function	Page 13-43, Section 13.5
Issuing an Interrupt to Another CPU	Page 13-44, Section 13.6
Motion CPU control instruction	Page 13-46, Section 13.7
Motion CPU device access	Page 13-47, Section 13.8
Multiple CPU Synchronized Boot-Up	Page 13-48, Section 13.9

13.1 Data Communications by MELSEC Data Link Functions

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	×

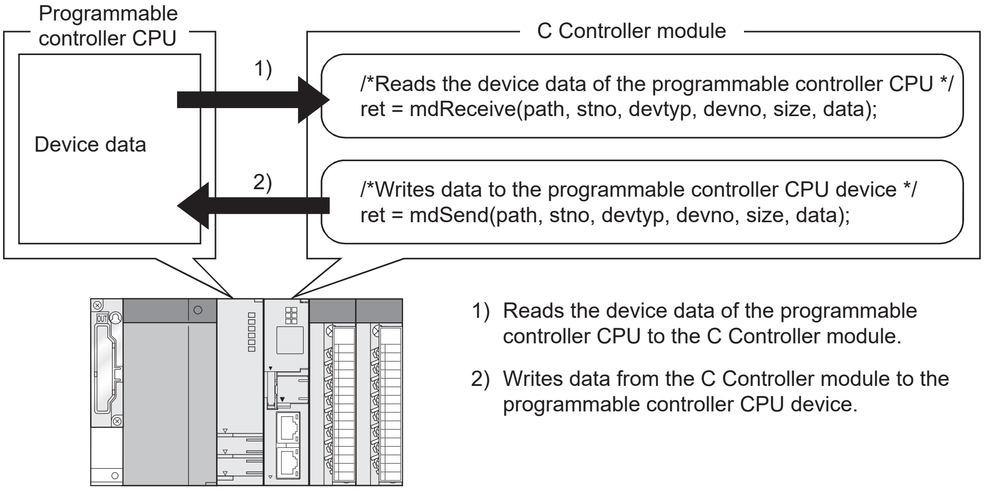
(1) Data communications by MELSEC data link functions

From the C Controller module, access can be made to device data in a programmable controller CPU or C Controller module.

Use MELSEC data link functions when creating a user program of the C Controller module.

For the Q06CCPU-V-B, MELSEC data link functions are not applicable.

To access device data of a programmable controller CPU from the Q06CCPU-V-B, use the CPU shared memory. (☞ Page 13-12, Section 13.3)



```
/*Reads the device data of the programmable controller CPU */
ret = mdReceive(path, stno, devtyp, devno, size, data);
```

```
/*Writes data to the programmable controller CPU device */
ret = mdSend(path, stno, devtyp, devno, size, data);
```

1) Reads the device data of the programmable controller CPU to the C Controller module.

2) Writes data from the C Controller module to the programmable controller CPU device.

Figure 13.1 Data communications by MELSEC data link functions

Remark

For the MELSEC data link functions, refer to the following.

☞ C Controller Module User's Manual (Utility Operation, Programming)

13.2 Interrupt from Another CPU

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	△	△

(1) About interrupt from another CPU

When an interrupt occurs from another CPU (programmable controller CPU or C Controller module), the C Controller module can:

- Execute a routine registered by the QBF_EntryCpuInt function as an interrupt routine (interrupt program).[*1][*2]
- Resume a user program that is waiting for an interrupt event by the QBF_WaitEvent function.

[*1] The following C Controller modules cannot execute it as an interrupt routine (interrupt program).
- Q12DCCPU-V with a serial number whose first five digits are "12041" or earlier
- Q06CCPU-V(-B)

[*2] Interrupt routine (interrupt program) execution does not produce a delay that could be caused by user program execution or priority in multitasking.
For that reason, real-time processing can be performed when an interrupt occurs.
However, functions that can be used are restricted because of ISR (Interrupt Service Routine) restrictions in VxWorks6.4.
For details on ISR restrictions, refer to the manual for VxWorks.

To issue an interrupt, use the following.

- For an interrupt from a programmable controller CPU, use the S(P).GINT or D(P).GINT instruction.
- For an interrupt from a C Controller module, use the QBF_GINT function.

(2) Processing outline

(a) How to execute interrupt routines (interrupt programs)

This section explains how to execute a routine registered by the QBF_EntryCpuInt function as an interrupt routine (interrupt program).

1) With the QBF_EntryCpuInt function, register a routine that corresponds to an interrupt from another CPU (programmable controller CPU or C Controller module).

2) With the QBF_EnableCpuInt or QBF_EnableCpuInt_ISR function, enable the routine registered in step 1).

3) If any of the following is executed in status 2), the routine registered in 1) is executed as an interrupt routine (interrupt program).

- User program of C Controller module (another CPU) (QBF_GINT function)
- Sequence program of a programmable controller CPU (another CPU)
  (S.GINT or D.GINT instruction)

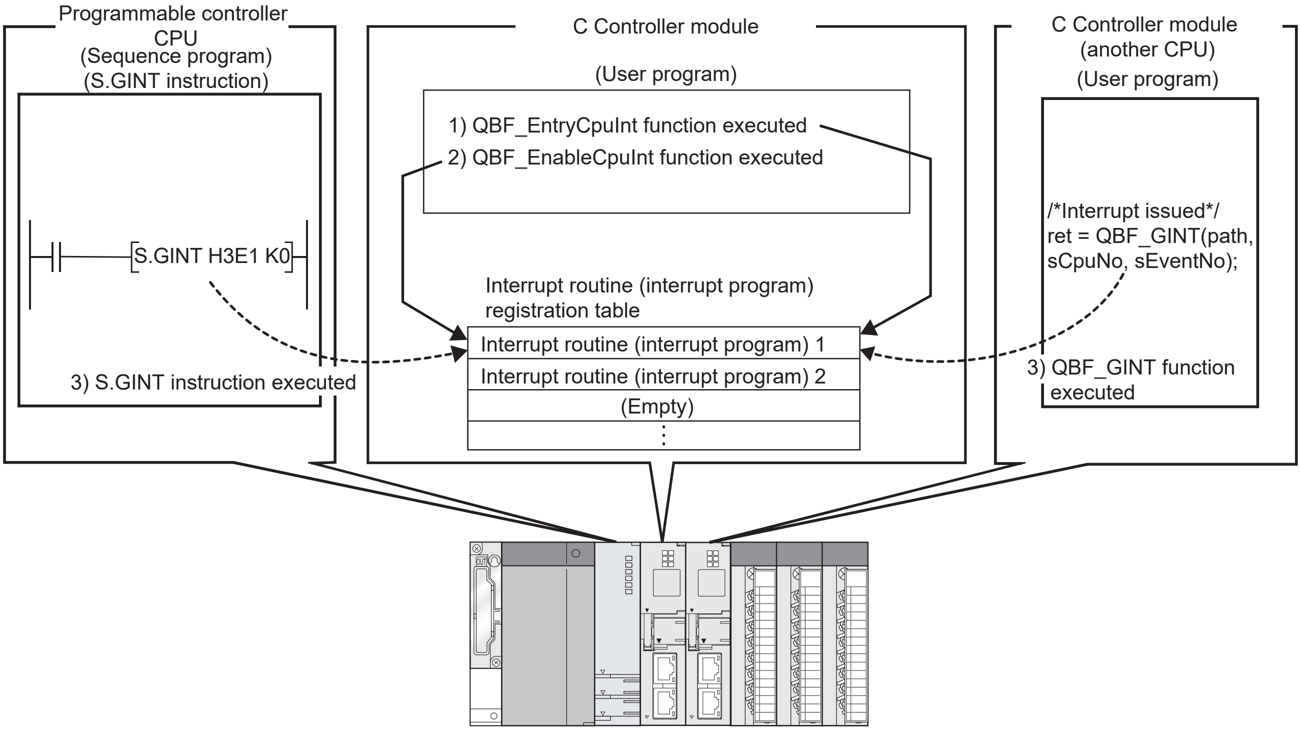

Programmable controller CPU
(Sequence program)
(S.GINT instruction)
S.GINT H3E1 K0
3) S.GINT instruction executed
C Controller module
(User program)
1) QBF_EntryCpuInt function executed
2) QBF_EnableCpuInt function executed
Interrupt routine (interrupt program) registration table
Interrupt routine (interrupt program) 1
Interrupt routine (interrupt program) 2
(Empty)
⋮
C Controller module (another CPU)
(User program)
/*Interrupt issued*/
ret = QBF_GINT(path, sCpuNo, sEventNo);
3) QBF_GINT function executed

Figure 13.2 How to execute interrupt routines (interrupt programs)

To disable an interrupt routine (interrupt program), perform the following.

4) With the QBF_DisableCpuIn or QBF_DisableCpuInt_ISR function, disable the routine registered in 1).

5) Even if an interrupt is issued from another CPU (programmable controller CPU or C Controller) after step 4), the routine registered in 1) will not be executed.

(b) How to resume the user program

This section explains how to resume the user program that is waiting for an interrupt event by the QBF_WaitEvent function.

1) In the user program, call the QBF_WaitEvent function.

2) By 1), the user program is placed into the interrupt event waiting status.

3) In the above 2) status, execute either of the following.
- User program of the other C Controller module (another CPU) (QBF_GINT function)
- Sequence program of a programmable controller CPU (another CPU) (S.GINT or D.GINT instruction)

4) The user program is restored from the event waiting status.

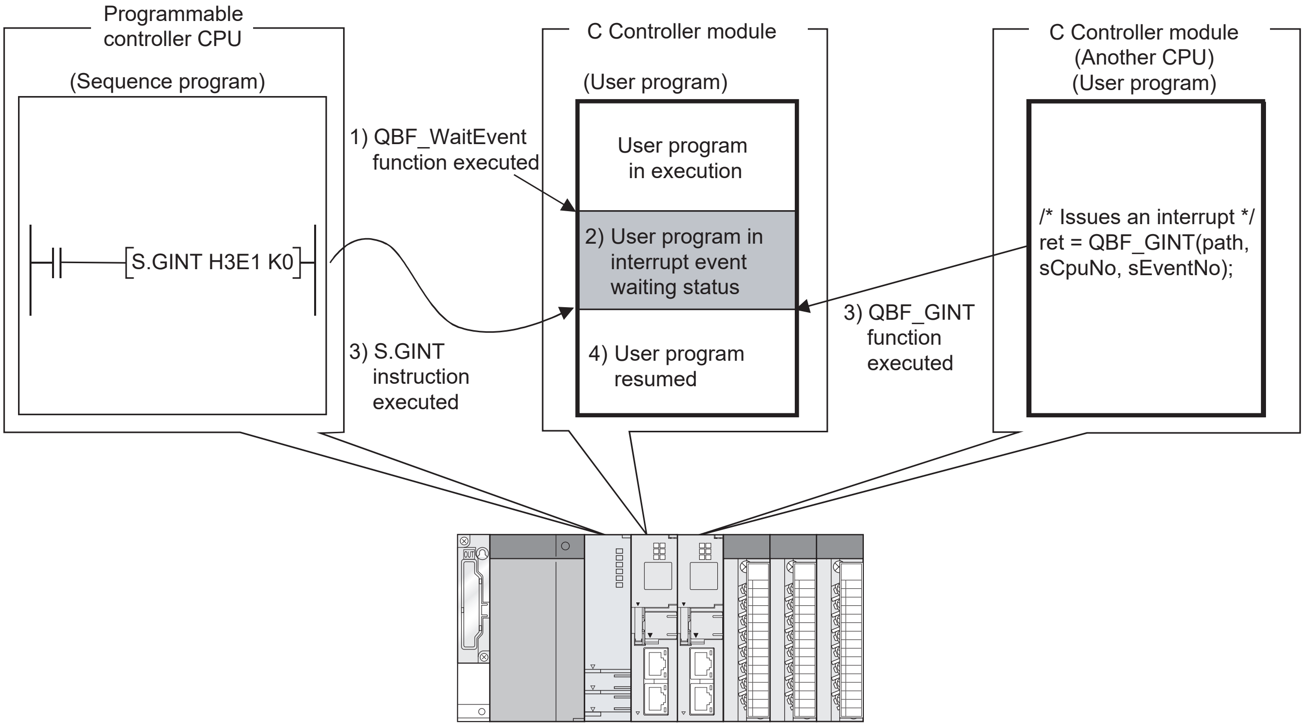

Programmable controller CPU
(Sequence program)
S.GINT H3E1 K0
1) QBF_WaitEvent function executed
3) S.GINT instruction executed
C Controller module
(User program)
User program in execution
2) User program in interrupt event waiting status
4) User program resumed
3) QBF_GINT function executed
C Controller module
(Another CPU)
(User program)
/* Issues an interrupt */
ret = QBF_GINT(path, sCpuNo, sEventNo);

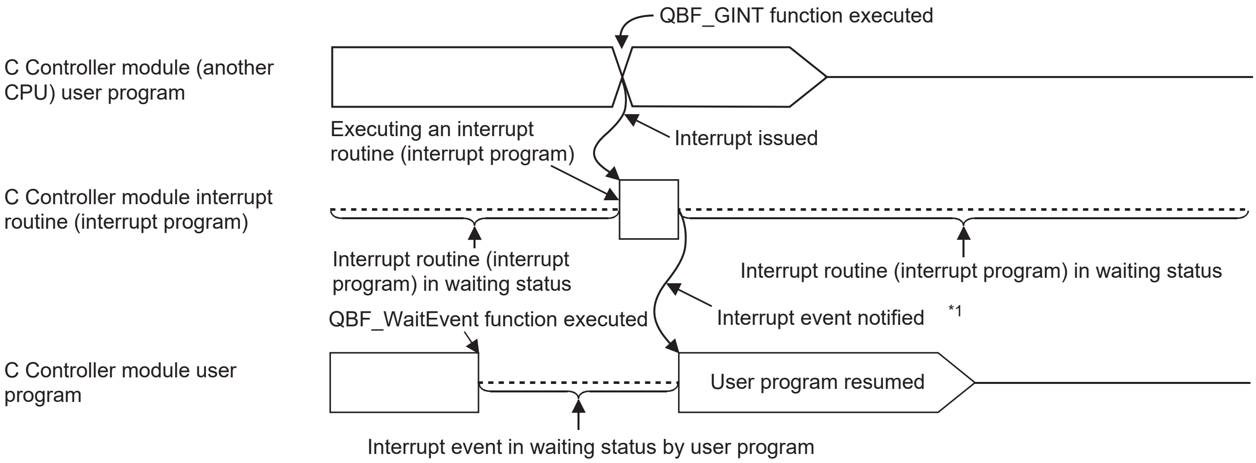
Figure 13.3 Interrupt by the event notification function

Remark

1) For the bus interface functions, refer to the following.
☞ C Controller Module User's Manual (Utility Operation, Programming)

2) The QBF_GINT function can issue an interrupt to a Motion CPU. (☞ Page 13-44, Section 13.6)

(c) Operation timing


QBF_GINT function executed
C Controller module (another CPU) user program
Executing an interrupt routine (interrupt program)
Interrupt issued
C Controller module interrupt routine (interrupt program)
Interrupt routine (interrupt program) in waiting status
Interrupt routine (interrupt program) in waiting status
QBF_WaitEvent function executed
Interrupt event notified *1
C Controller module user program
User program resumed
Interrupt event in waiting status by user program

*1 Interruput event notified after the interrupt routine (interrupt program) ends

Figure 13.4 Operation timing

For details on the operation timing when using dedicated instructions of programmable controller CPUs, refer to the following.

☞ Page 13-9, Section 13.2 (4) (d)

(3) Functions

The following shows the functions used for interrupt from another CPU.

Table 13.2 Functions used for interrupt from another CPU

Function name	Function
QBF_EntryCpuInt	Registers a routine so that it can be called when an interrupt occurs from another CPU.
QBF_EnableCpuInt	Enables the routine registered for interrupt from another CPU.
QBF_DisableCpuInt	Disables the routine registered for interrupt from another CPU.
QBF_EnableCpuInt_ISR	Enables the routine registered for interrupt from another CPU.
QBF_DisableCpuInt_ISR	Disables the routine registered for interrupt from another CPU.
QBF_WaitEvent	Waits for an interrupt event notification from another CPU.

Remark

For the bus interface functions, refer to the following.

☞ C Controller Module User's Manual(Utility Operation, Programming)

(4) Dedicated instructions

The following shows programmable controller CPU dedicated instructions used for interrupt from a programmable controller CPU (another CPU).

Table 13.3 Devices available for S(P).GINT and D(P).GINT instructions

Setting data	Available devices									
	Internal device (System, user)		File register R, ZR		Link direct device J□\□		Intelligent function module U□\G□	Index register, Zn	Constant K, H	Others
	Bit	Word	Bit	Word	Bit	Word				
(n1)	—	○*1	—	○	—				○	—
(n2)	—	○*2	—	○	—				○	—
(D1)*3	△*4	—	△*4	—	—				—	—
(D2)*3	—	△*4	—	△*4	—				—	—

○: Available, △: Partially available, —: N/A

* 1 For the Basic model QCPU, High Performance model QCPU, and Universal model QCPU, index modification is available.

* 2 For the Basic model QCPU and Universal model QCPU, index modification is available.

* 3 Only when both of (D1) and (D2) are omitted, they can be actually omitted.

* 4 Local devices cannot be used.

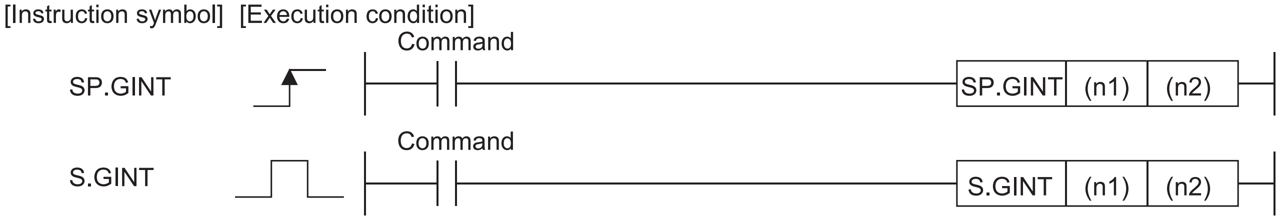


Figure 13.5 S(P).GINT instruction

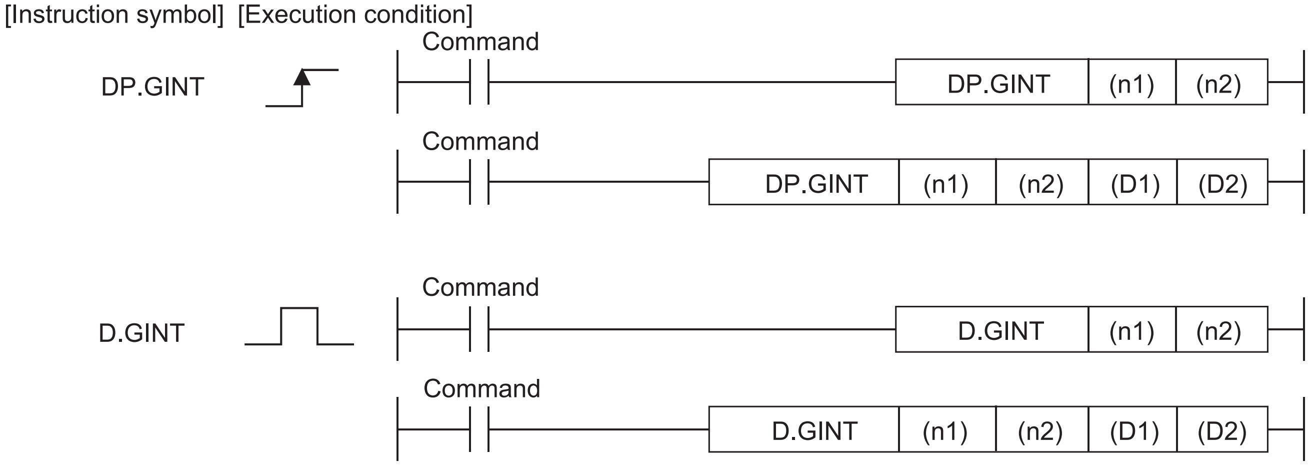


Figure 13.6 D(P).GINT instruction

(a) Setting data

Table 13.4 Setting data in S(P).GINT and D(P).GINT instructions

Setting data	Setting	Data type
(n1)	Start I/O No. of target CPU÷16 Actually specified values are as follows: CPU No.1: 3E0H, CPU No.2: 3E1H, CPU No.3: 3E2H, CPU No.4: 3E3H	BIN 16 bits
(n2)	Interrupt pointer No. (0 to 15)	BIN 16 bits
(D1)*1	Completion device (D1+0): Device that is turned ON for one scan upon completion of the instruction processing (D1+1): Device that is turned ON for one scan upon failure of the instruction processing (When failed, D1+0 is also turned ON.)	Bit
(D2)*1	Device where the completion status data is stored.	Word

* 1 Only when both of (D1) and (D2) are omitted, they can be actually omitted.

(b) Usable devices
The following devices are available for dedicated instructions.

Table 13.5 Usable devices

Internal device		File register	Constant
Bit	Word*1		
M, L, B	D, W, @□	R, ZR	K, H

* 1 Number of bit device digits can be specified for word data.
It can be specified with Number of digits Head No. of bit device .
For example, 16 points from M0 to M15 are specified with K4M0 .

(c) Control details
On the rise (OFF to ON) of the S(P).GINT or D(P).GINT execution command in the sequence program, an interrupt is issued to the C Controller module.
Upon receipt of an interrupt from a programmable controller CPU, the C Controller module performs the following processing.

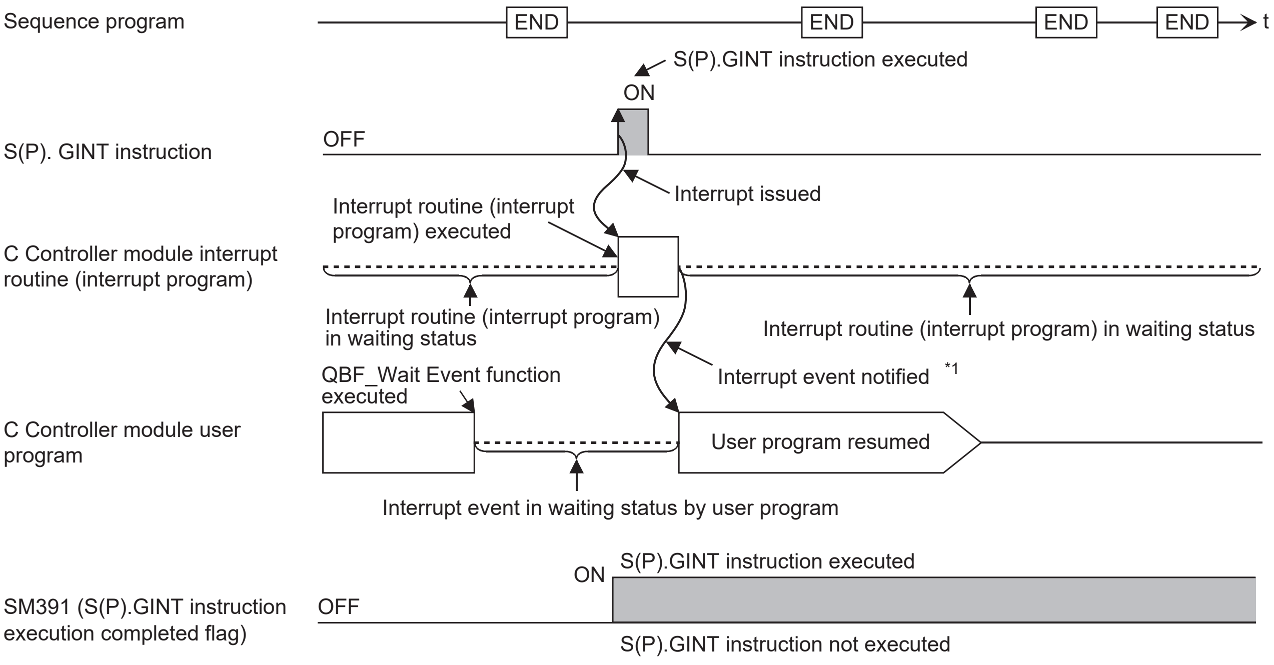
- Executes a routine registered by the QBF_EntryCpuInt function as an interrupt routine (interrupt program).*1
- Resumes the user program that is waiting for an interrupt event by the QBF_WaitEvent function.

* 1 The following C Controller modules cannot execute it as an interrupt routine (interrupt program).
- Q12DCCPU-V with a serial number whose first five digits are "12041" or earlier
- Q06CCPU-V(-B)

1) When transmission of the instruction command to the C Controller module is completed
The SM391 (S(P).GINT or D(P).GINT execution completed) flag of the programmable controller CPU is set to ON.

2) When transmission of the instruction command to the C Controller module failed
The SM391 (S(P).GINT or D(P).GINT execution completed) flag of the programmable controller CPU is set to OFF.

(d) Operation timing


Sequence program
END
END
END
END
t
S(P).GINT instruction executed
ON
S(P). GINT instruction
OFF
Interrupt issued
Interrupt routine (interrupt program) executed
C Controller module interrupt routine (interrupt program)
Interrupt routine (interrupt program) in waiting status
Interrupt routine (interrupt program) in waiting status
QBF_Wait Event function executed
Interrupt event notified *1
C Controller module user program
User program resumed
Interrupt event in waiting status by user program
ON
S(P).GINT instruction executed
SM391 (S(P).GINT instruction execution completed flag)
OFF
S(P).GINT instruction not executed

*1 Interrupt event notified after the interrupt routine (interrupt program) ends.

Figure 13.7 Operation timing of the S(P).GINT instruction

(e) Error details

In any of the following cases, an operation error occurs, the error flag (SM0) of the programmable controller CPU is set to ON, and an error code is stored in SD0.

Table 13.6 Error codes related to S(P).GINT and D(P).GINT instructions

Error code[*1]	Cause	Action
2110	CPU No. of a reserved ("Empty" setting) or non-mounted CPU is specified for the target CPU start I/O No. ÷16 (n1).	Check and correct the sequence program.
2114	The originating CPU is specified for the target CPU start I/O No.÷16 (n1).	
2117	A module that does not support the S(P).GINT and D(P).GINT instructions is specified for the target CPU start I/O No. ÷16 (n1).	
4100	Any of 0 to 3DFH, or 3E4H is specified for the target CPU start I/O No. ÷16 (n1).	

* 1 0000H (Normal)

(f) Program example
In the sequence program below, an interrupt is issued to the C Controller module that is set as CPU No.2.

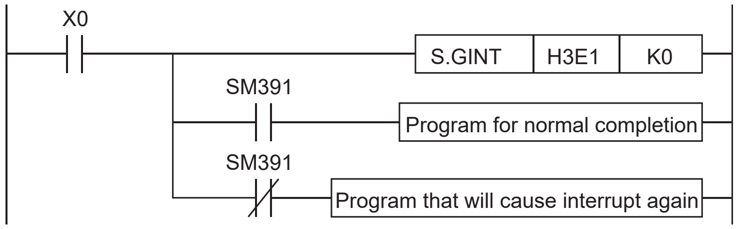

X0
S.GINT
H3E1
K0
SM391
Program for normal completion
SM391
Program that will cause interrupt again

Figure 13.8 Example of a program using S(P).GINT instruction

(5) Precautions

(a) When an interrupt event has already been notified at execution of the QBF_WaitEvent function
When an interrupt event has already been notified from a programmable controller CPU or C Controller module (another CPU) at the time of the QBF_WaitEvent function execution, the user program is restored from the interrupt event waiting status at the same time with the QBF_WaitEvent function execution.
If multiple interrupt events have been notified with the same interrupt event No. at the time of the QBF_WaitEvent execution, the user program processes them as a single interrupt event notification.

(b) When using the event notification function in multiple user programs
Do not set the same CPU No. and the same interrupt event No. on more than one program.
If such settings are specified, which user program will receive the interrupt event will be uncertain.

Remark

For details of the D(P).GINT instruction, refer to the following.
☞ Q173DCPU/Q172DCPU Motion Controller (SV13/SV22) Programming Manual (Motion SFC)

13.2.1 Multiple CPU synchronous interrupt function

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	×	×

(1) Multiple CPU synchronous interrupt function

By registering a routine capable of the multiple CPU synchronous interrupt with the QBF_EntryMultiCPUSyncInt function, a program that is executed synchronously with a Motion CPU (Q172DCPU or Q173DCPU) can be created.

The multiple CPU synchronous interrupt is an interrupt that is issued in the same cycles (0.88ms) as Motion CPU (Q172DCPU or Q173DCPU) control cycles.

(2) Multiple CPU synchronous interrupt function processing

1) With the QBF_EntryMultiCPUSyncInt function, register a routine that is capable of the multiple CPU synchronous interrupt.

2) With the QBF_EnableMultiCPUSyncInt function, enable the routine registered in 1).
(For every multiple CPU synchronous interrupt (periodically), the routine registered in 1) is executed.)

3) With the QBF_DisableMultiCPUSyncInt function, disable the routine registered in 1).
(For every multiple CPU synchronous interrupt, the routine registered in 1) will not be executed.)

Remark

For the bus interface functions, refer to the following.

☞ C Controller Module User's Manual (Utility Operation, Programming)

13.3 Data Communications Using CPU Shared Memory

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	○

(1) Data communications using CPU shared memory

The C Controller module can exchange data with CPU modules using the CPU shared memory.

Use the bus interface functions when creating user programs of the C Controller module.

The following describes the methods for data communications using the CPU shared memory.

(a) Data communication methods using the CPU shared memory

There are the following two kinds of the communication methods.

- Using the auto refresh of programmable controller CPU and Motion CPU
- Not using the auto refresh of programmable controller CPU

(b) Selection of a data communication method

For data communications using the CPU shared memory, select a method that is appropriate to the relevant CPU modules.

The following table lists the availability for each type.

Table 13.7 Availability for each data communication type

Data communication type	Communication method		Refer to
	Using auto refresh	Not using auto refresh	
Programmable controller CPU <-> C Controller module	○	○	Page 13-17, Section 13.3.2, Page 13-32, Section 13.3.4 (1) (a)
Motion CPU <-> C Controller module	○	×	Page 13-17, Section 13.3.2
C Controller module <-> C Controller module	×	○	Page 13-35, Section 13.3.4 (1) (b)

○: Available, ×: N/A

(2) CPU shared memory structure

For the structure of the CPU shared memory of the C Controller module, refer to the following.

☞ Page 13-14, Section 13.3.1

☒POINT

The CPU shared memory is accessible only when the number of CPUs is set to 2 or more in <<Multiple CPU settings>> of C Controller setting utility.
If the CPU shared memory is accessed without setting two or more CPU modules, CPU No. error (return value: -28662) is detected.

Remark

For the bus interface functions, refer to the following.
☞ C Controller Module User's Manual (Utility Operation, Programming)

13.3.1 CPU shared memory structure

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	○

The following shows the CPU shared memory structure and accessibility of the C Controller module.

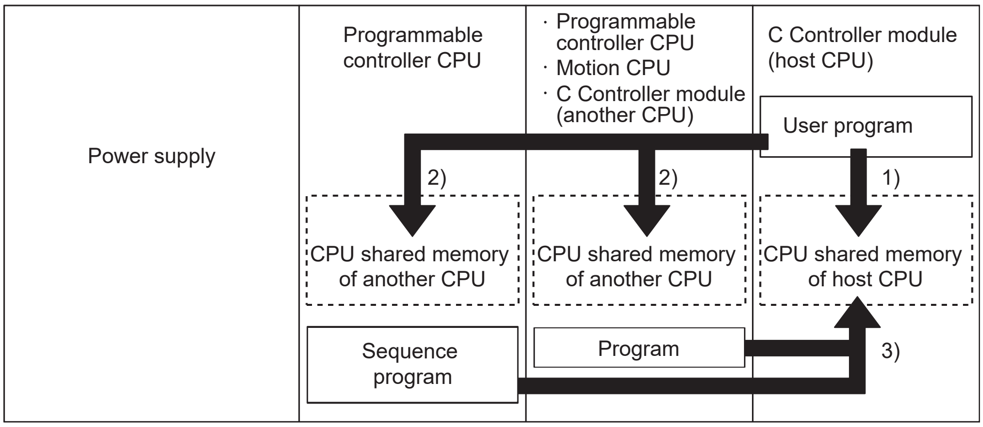

Power supply
Programmable controller CPU
· Programmable controller CPU
· Motion CPU
· C Controller module (another CPU)
C Controller module (host CPU)
User program
2)
2)
1)
CPU shared memory of another CPU
CPU shared memory of another CPU
CPU shared memory of host CPU
Sequence program
Program
3)

Figure 13.9 Access diagram

Table 13.8 Accessibility

Address[*1, *6]	Area name		Access from host CPU				Access from another CPU	
			Access to host 1)		Access to another CPU 2)		Access to host CPU 3)	
			Write[*2]	Read[*3]	Write	Read[*4]	Write	Read[*5]

0H(0H)
to
1FFH(5FH)
	QCPU standard area	Host CPU operation information area	N/A	Accessible	N/A	Accessible	N/A	Accessible

200H(60H)
to
7FFH(BFH)
		System area	N/A	N/A	N/A	Accessible	N/A	Accessible

800H(C0H)
to
FFFH(1FFH)
		Auto refresh area	Accessible	Accessible	N/A	Accessible	N/A	Accessible

		User setting area	Accessible	Accessible	N/A	Accessible	N/A	Accessible
1000H to 270FH	Use prohibited[*7]		N/A	N/A	N/A	N/A	N/A	N/A
2710H to 5F0FH	Multiple CPU high speed transmission area[*7]		Accessible	Accessible	N/A	Accessible	N/A	Accessible

[*1] Indicates addresses of the CPU shared memory.

[*2] To write data to the auto refresh area or user setting area of the host CPU (C Controller module), use the QBF_ToBuf function.

[*3] To read data from the host CPU operation information area, auto refresh area, or user setting area of the host CPU (C Controller module), use the QBF_FromBuf function.

[*4] To read data from the host CPU operation information area, auto refresh area, or user setting area of another CPU (programmable controller CPU or Motion CPU), use the QBF_FromBuf function.

[*5] For information on how to access from another CPU (programmable controller CPU or Motion CPU), refer to the manual for each CPU module.

[*6] When CPU No.1 is a Basic model QCPU, addresses of the CPU shared memory are as shown in parentheses.

[*7] For the Q06CCPU-V(-B), the use prohibited area and Multiple CPU high speed transmission area are not provided.

Table 13.9 CPU shared memory structure

Area name		Description
QCPU standard area	Host CPU operation information area	Error details or operation status of the host CPU (C Controller module) are stored in this area. (☞ Table 13.10)
	System area	This area is used by the system.
	Auto refresh area	This area data are automatically refreshed into devices of the programmable controller CPU or Motion CPU, based on the auto refresh settings. The area size varies depending on the parameter settings.
	User setting area	This area can be used at user's discretion. The area size varies depending on the parameter settings for the Auto refresh area.
Use prohibited		This area cannot be used.
Multiple CPU high speed transmission area		This area is used for the multiple CPU high speed transmission. The auto refresh area or user setting area can be set with a parameter.

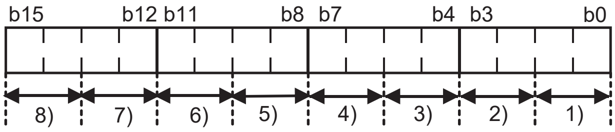
☒POINT

For the Q06CCPU-V(-B), communication using the multiple CPU high speed transmission area is not available.
Use the QCPU standard area.

Table 13.10 List of host CPU operation information areas

Shared memory address	Name	Description	Details
0H	Information presence	Information flag	The area is provided for checking if information is stored in the host CPU's operation information area (1H to 1FH) or not. 0: No information, 1: Information exists.
1H	Diagnostic error	Diagnostic error number	An error number for a diagnostic error is stored as a binary code.*2
2H	Date and time of diagnostic error	Date and time of diagnostic error	The year and month of the error number storage in CPU shared memory area address 1H are stored as a 2-digit BCD code.*2
3H			The day and time of the error number storage in the CPU shared memory area address 1H are stored as a 2-digit BCD code.*2
4H			The minute and second of the error number storage in the CPU shared memory area address 1H are stored as a 2-digit BCD code.*2
5H	Error information identification code	Error information identification code	An identification code is stored in this area to show what error information is stored in the common or individual error information area.*2
6H to 10H	Common error information	Common error information	When an error occurred during diagnostics, common information for the error number is stored.*2
11H to 1BH	Individual error information	Individual error information	When an error occurred during diagnostics, individual information for the error number is stored.*2
1CH	Empty	—	Not used
1DH	Switch status	C Controller module switch status	The C Controller module switch status is stored. 0: RUN, 1: STOP
1EH	LED status	C Controller module LED status	The bit pattern of the C Controller module's LEDs is stored. (☞ Figure 13.10) The same data can be obtained by using the QBF_ReadStatusEx function.
1FH	C Controller module operation status	C Controller module operation status	The operation status of the C Controller module is stored. (☞ Figure 13.11) The same data can be obtained by using the QBF_ReadStatusEx function.

* 2 If no error is detected, 0 is stored.



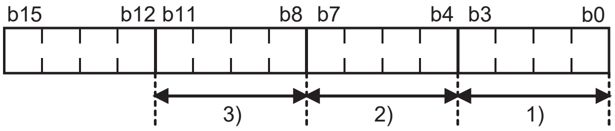
1): RUN
2): ERR.
3): USER
4): CF CARD (other than Q06CCPU-V-B), STS (Q06CCPU-V-B)
5): Not used
6): Not used
7): USER LED color (Q12DCCPU-V only)*1
8): MODE

*1 Valid only when USER LED is on.

<LED status>

1) to 4)	7)	8)
0: Off	0: Red	0: Off
1: On	1: Green	1: Lit green
2: Flashing (slow)	2: Orange	2: Lit orange
3: Flashing (fast)	3: Not used	3: Flashing

Figure 13.10 LED status



1): C Controller module operation status	0: RUN 1: Reserved 2: STOP 3: PAUSE
2): STOP/PAUSE factor (in chronological order)	0: RUN/STOP/MODE switch 1: Reserved 2: Remote operation from C Controller setting utility 3: Execution of QBF_Control function from user program 4: Error 5: Remote operation from another CPU
3): Mode	0: Basic mode 1: Extended mode

Figure 13.11 Operation status

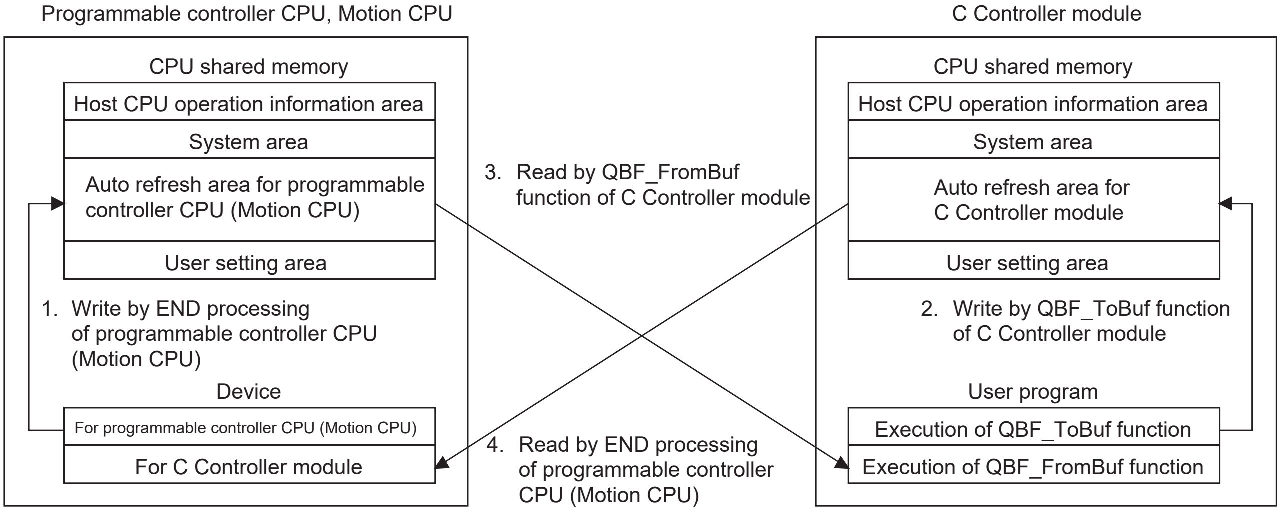
13.3.2 Data communications using auto refresh

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	○

This section explains the processing and setting for communicating data with a programmable controller CPU or Motion CPU, using the CPU shared memory and auto refresh.

(1) Data communications using auto refresh

The following shows the processing of data communication when using auto refresh.


Programmable controller CPU, Motion CPU
CPU shared memory
Host CPU operation information area
System area
Auto refresh area for programmable controller CPU (Motion CPU)
User setting area
1. Write by END processing of programmable controller CPU (Motion CPU)
Device
For programmable controller CPU (Motion CPU)
For C Controller module
3. Read by QBF_FromBuf function of C Controller module
4. Read by END processing of programmable controller CPU (Motion CPU)
C Controller module
CPU shared memory
Host CPU operation information area
System area
Auto refresh area for C Controller module
User setting area
2. Write by QBF_ToBuf function of C Controller module
User program
Execution of QBF_ToBuf function
Execution of QBF_FromBuf function

In END processing of programmable controller CPU or Motion CPU

1. Data in the device for programmable controller CPU (or Motion CPU) are transferred to the auto refresh area in the CPU shared memory.
4. Auto refresh area data in the C Controller module are transferred to the device for C Controller module in the programmable controller CPU (or Motion CPU).

At bus interface function execution of C Controller module

2. QBF_ToBuf function execution transfers user program settings to the auto refresh area in the C Controller module's CPU shared memory.
3. QBF_FromBuf function execution transfers the auto refresh area data of the programmable controller CPU (or Motion CPU) to the user program.

Figure 13.12 Communication using auto refresh

☒POINT

1. The auto refresh in step 4. in Figure 13.12 is performed in the END processing of the programmable controller CPU (or Motion CPU) after execution of the QBF_ToBuf function in step 2. in Figure 13.12.
2. Auto refresh cannot be used for communication between a C Controller module and the Q172DCPU or Q173DCPU.
   For communication between the Q12DCCPU-V and the Q172DCPU or Q173DCPU, utilize the multiple CPU high speed transmission area for auto refresh. (☞ Page 13-24, Section 13.3.3)
   For communication between the Q06CCPU-V(-B) and the Q172DCPU or Q173DCPU, use the user setting area. (☞ Page 13-39, Section 13.3.4 (1) (c))

(2) Auto refresh area setting

Data communications with a programmable controller CPU or Motion CPU using auto refresh require the auto refresh area setting.

Configure the auto refresh area settings in "Communication area setting (refresh setting)" in <<Multiple CPU settings>> of C Controller setting utility.

The "Communication area setting (refresh setting)" details are described below.

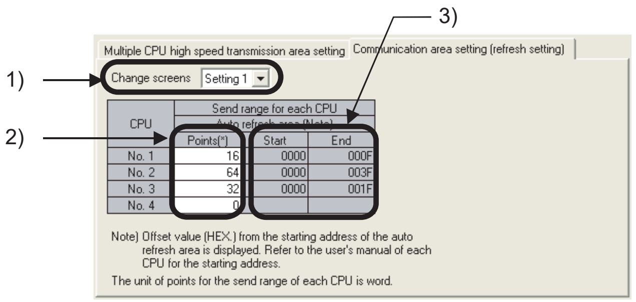
POINT

The auto refresh area setting must be identical for all CPUs in the multiple CPU system.

If any difference is identified in their settings, a parameter error will occur.

(a) "Communication area setting (refresh setting)"

The setting items in "Communication area setting (refresh setting)" are explained below.


3)
1)
2)
Multiple CPU high speed transmission area setting
Communication area setting (refresh setting)
Change screens
Setting 1

CPU	Send range for each CPU		
	Auto refresh area (Note)		
	Points(*)	Start	End
No. 1	16	0000	000F
No. 2	64	0000	003F
No. 3	32	0000	001F
No. 4	0		

Note) Offset value (HEX.) from the starting address of the auto refresh area is displayed. Refer to the user's manual of each CPU for the starting address.

The unit of points for the send range of each CPU is word.

Figure 13.13 Auto refresh area setting screen

1) "Change screens"

Four different ranges can be set for "Communication area setting (refresh setting)" by "Change screens".

2) "Points"

Set auto refresh area points for each CPU module in units of two points (two words).

Up to total 2048 points (2K words)[*1] can be set for four ranges (Setting 1 to 4) per CPU module, and up to total 8192 points (8K words)[*2] for all CPUs in a multiple CPU system.

For a CPU module that does not use the auto refresh, enter "0" in the "Points" field.

[* 1] For a Basic model QCPU, up to total 320 points (320 words) can be set for four ranges (Setting 1 to 4) per CPU module.

[* 2] When CPU No.1 is a Basic model QCPU, up to 4416 points (4416 words) can be set for all CPUs in a multiple CPU system.

3) "Start" and "End"

Setting a "Points" value automatically displays the first and last addresses of the auto refresh area in the "Start" and "End" fields as hexadecimal offset values.

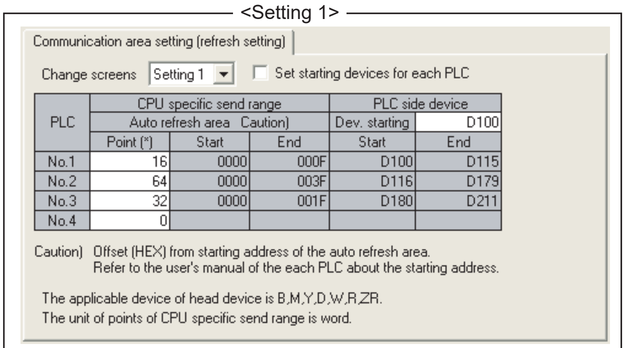
Remark

For the auto refresh area setting for a programmable controller CPU or Motion CPU, refer to the manual for each CPU module.

(b) Setting example

The following is an example of the auto refresh area setting.
In this example, a High Performance model QCPU is used as CPU No.1, and CPU No.4 does not use auto refresh.

<Setting 1>



Communication area setting (refresh setting)

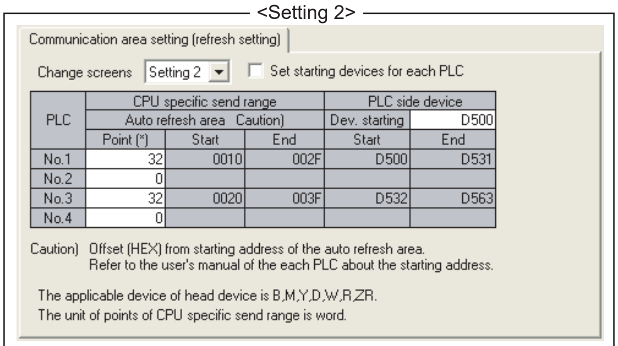
Change screens: Setting 1 ▾ □ Set starting devices for each PLC

PLC	CPU specific send range			PLC side device	
	Auto refresh area Caution)			Dev. starting	D100
	Point (*)	Start	End	Start	End
No.1	16	0000	000F	D100	D115
No.2	64	0000	003F	D116	D179
No.3	32	0000	001F	D180	D211
No.4	0				

Caution) Offset (HEX) from starting address of the auto refresh area.
Refer to the user's manual of the each PLC about the starting address.

The applicable device of head device is B,M,Y,D,W,R,ZR.
The unit of points of CPU specific send range is word.

<Setting 2>



Communication area setting (refresh setting)

Change screens: Setting 2 ▾ □ Set starting devices for each PLC

PLC	CPU specific send range			PLC side device	
	Auto refresh area Caution)			Dev. starting	D500
	Point (*)	Start	End	Start	End
No.1	32	0010	002F	D500	D531
No.2	0				
No.3	32	0020	003F	D532	D563
No.4	0				

Caution) Offset (HEX) from starting address of the auto refresh area.
Refer to the user's manual of the each PLC about the starting address.

The applicable device of head device is B,M,Y,D,W,R,ZR.
The unit of points of CPU specific send range is word.

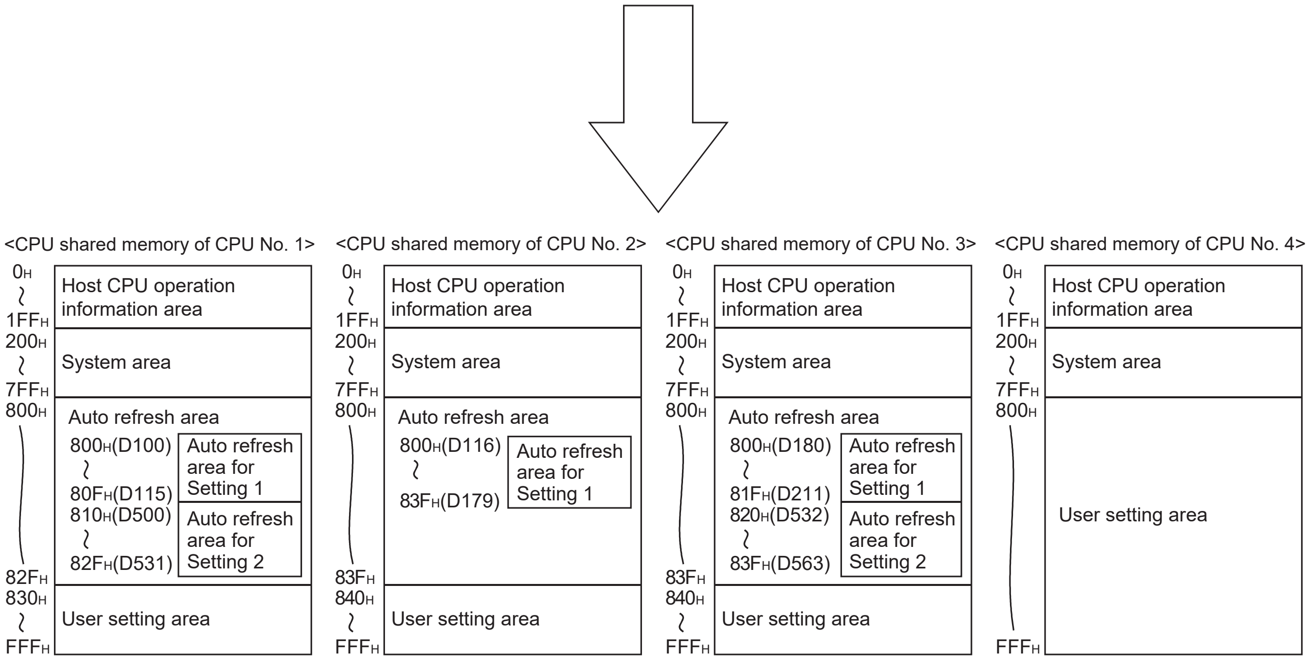

<CPU shared memory of CPU No. 1>
0H ~ 1FFH Host CPU operation information area
200H ~ 7FFH System area
800H Auto refresh area
800H(D100) ~ 80FH(D115) Auto refresh area for Setting 1
810H(D500) ~ 82FH(D531) Auto refresh area for Setting 2
82FH
830H ~ FFFH User setting area
<CPU shared memory of CPU No. 2>
0H ~ 1FFH Host CPU operation information area
200H ~ 7FFH System area
800H Auto refresh area
800H(D116) ~ 83FH(D179) Auto refresh area for Setting 1
83FH
840H ~ FFFH User setting area
<CPU shared memory of CPU No. 3>
0H ~ 1FFH Host CPU operation information area
200H ~ 7FFH System area
800H Auto refresh area
800H(D180) ~ 81FH(D211) Auto refresh area for Setting 1
820H(D532) ~ 83FH(D563) Auto refresh area for Setting 2
83FH
840H ~ FFFH User setting area
<CPU shared memory of CPU No. 4>
0H ~ 1FFH Host CPU operation information area
200H ~ 7FFH System area
800H ~ FFFH User setting area

Figure 13.14 Auto refresh area setting example

(3) Precautions for data communications using auto refresh

Depending on the timing of writing data to the host CPU's auto refresh area or reading data from another CPU, old and new data may be mixed in the area for each CPU. Therefore, configure an interlock program for auto refresh so that data of another CPU will not be used when old and new data are mixed.

(a) Interlock program example

Figure 13.15 shows an interlock program example for sending data from CPU No.1 (programmable controller CPU) to CPU No.2 (C Controller module).

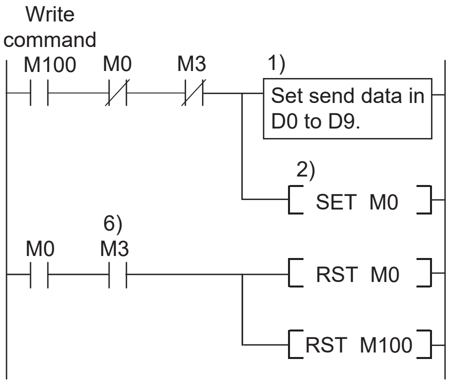
Table 13.11 Example of parameter setting for interlock program

Auto refresh setting for CPU No.1							Direction	Auto refresh setting for CPU No.2				
CPU No.	Transfer No.	CPU specific send range			Device setting			CPU No.	Transfer No.	CPU specific send range		
		Points	Start	End	Start	End				Points	Start	End
No.1	Transfer 1	2	0	1	M0	M31	→	No.1	Transfer 1	2	0	1
	Transfer 2	10	2	11	D0	D9			Transfer 2	10	2	11
No.2	Transfer 1	2	0	1	M32	M63	←	No.2	Transfer 1	2	0	1

The parameter setting in Table 13.11 uses the following interlock devices.

- M0 as an interlock device (Data set completion bit) of CPU No.1
- M32 as an interlock device (Data processing completion bit) of CPU No.2

Sending side program for CPU No.1 (programmable controller CPU)



Receiving side program for CPU No.2 (C Controller module)

```
{
                            :
  unsigned short    usBuf;                    /* Data storage destination */
  unsigned short    pusData[128];         /* Data storage destination */
                            :
   /***********************************************************************/
   /* Confirm that 1 is set in the area corresponding to M0 in the CPU No.1 auto refresh area. */
   /* (CPU No.1 data set completion bit turns ON.)                     */
   /***********************************************************************/
   while(1){
      sRet = QBF_FromBuf(lPath, 0x3E0, 0x800, 1, &usBuf, 1);                    3)
      if( sRet != 0) return(sRet);
      if( (usBuf & 0x1) == 0x1 ) break;        /* When 1 is set, processing gets out from the loop. */
      taskDelay(2);
   }

   /*****************************************************************/
   /* Data are read from the area corresponding to D0 to D9 in the CPU No.1 auto refresh area. */
   /*****************************************************************/
   sRet = QBF_FromBuf(lPath, 0x3E0, 0x802, 10, pusData, 128);                  4)
   if( sRet != 0) return(sRet);

   /*****************************************************************/
   /* 1 is set in the host CPU's auto refresh area corresponding to M32 of CPU No.1.  */
   /* (CPU No.2 data processing completion bit turns ON.)  */
   /*****************************************************************/
   usBuf = 1;
   sRet = QBF_ToBuf(lPath, 0x3E1, 0x800, 1, &usBuf, 1);                        5)
   if( sRet != 0) return(sRet);

   /***********************************************************************/
   /* Confirm that 0 is set in the area corresponding to M0 in the CPU No.1 auto refresh area.  */
   /* (CPU No.1 data set completion bit turns OFF.)                   */
   /***********************************************************************/
   while(1){
      sRet = QBF_FromBuf(lPath, 0x3E0, 0x800, 1, &usBuf, 1);                    7)
      if( sRet != 0) return(sRet);
      if( (usBuf & 0x1) == 0x0 ) break;        /* When 0 is set, processing gets out from the loop. */
      taskDelay(2);
   }

   /*****************************************************************/
   /* 0 is set in the host CPU's auto refresh area corresponding to M32 of CPU No.1.   */
   /* (CPU No.2 data processing completion bit turns OFF.) */
   /*****************************************************************/
   usBuf = 0;
   sRet = QBF_ToBuf(lPath, 0x3E1, 0x800, 1, &usBuf, 1);                        8)
   if( sRet != 0) return(sRet);
}
```

Figure 13.15 Interlock program example

1) CPU No.1 stores send data in D0 to D9.

2) CPU No.1 turns on the data set completion bit (M0).
   - In the END processing of CPU No.1, the above data are written to the auto refresh area in its CPU No.1 send area.
   - The auto refresh area data in the CPU No.1 send area are sent to CPU No.2.

By the user program of CPU No.2, data are read from the CPU No.1 auto refresh area to the internal buffer.

3) CPU No.2 detects ON of the data set completion bit (M0).

4) CPU No.2 processes the received data.

5) CPU No.2 turns on the data processing completion bit (this information is refreshed to M32 of CPU No.1.)

In the END processing of CPU No.1, the data are read from the auto refresh area in its CPU No.2 send area to the specified device.

6) CPU No.1 detects ON of the data processing completion bit (M32), and turns off the data set completion bit (M0).

By the user program of CPU No.2, check that the data set completion bit (M0) of CPU No.1 is OFF.

7) CPU No.2 detects OFF of the data set completion bit (M0).

8) CPU No.2 turns off the data processing completion bit (this information is refreshed to M32 of CPU No.1.)

13.3.3 Communication using the multiple CPU high speed transmission area and auto refresh

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	×	×

(1) Communication using the multiple CPU high speed transmission area and auto refresh

The communication using the multiple CPU high speed transmission area in the CPU shared memory and auto refresh can be executed when all of the following conditions are satisfied.

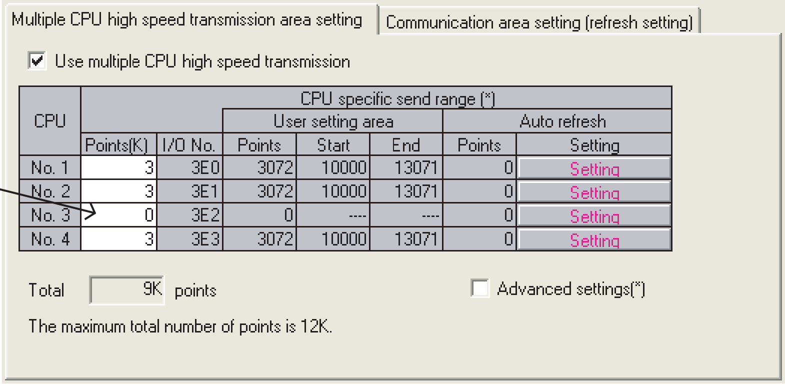
1) A multiple CPU high speed main base unit (Q3□DB) is used.

2) A Universal model QCPU (except the Q00UCPU, Q01UCPU, and Q02UCPU) or the Q12DCCPU-V is used as CPU No.1.

3) Two or more Universal model QCPU(s) (except the Q00UCPU, Q01UCPU, and Q02UCPU), Motion CPU(s) (Q172DCPU or Q173DCPU), Q12DCCPU-V(s) or the Q24DHCCPU-V/-LS are used.

Communication using the multiple CPU high speed transmission area and auto refresh are not available with a CPU module other than those shown in the above 3), even if it is mounted on a multiple CPU high speed main base unit.

If a CPU module other than those shown in 3) is mounted on a multiple CPU high speed main base unit, set "0" in the relevant "Points" field of "CPU specific send range" in "Multiple CPU high speed transmission area setting".



Enter 0 for a CPU module other than the modules shown in condition 3).

Multiple CPU high speed transmission area setting
Communication area setting (refresh setting)

☑ Use multiple CPU high speed transmission

CPU	CPU specific send range (*)						
			User setting area			Auto refresh	
	Points(K)	I/O No.	Points	Start	End	Points	Setting
No. 1	3	3E0	3072	10000	13071	0	Setting
No. 2	3	3E1	3072	10000	13071	0	Setting
No. 3	0	3E2	0	----	----	0	Setting
No. 4	3	3E3	3072	10000	13071	0	Setting

Total 9K points

□ Advanced settings(*)

The maximum total number of points is 12K.

Figure 13.16 Refresh setting example for multiple CPU high speed transmission

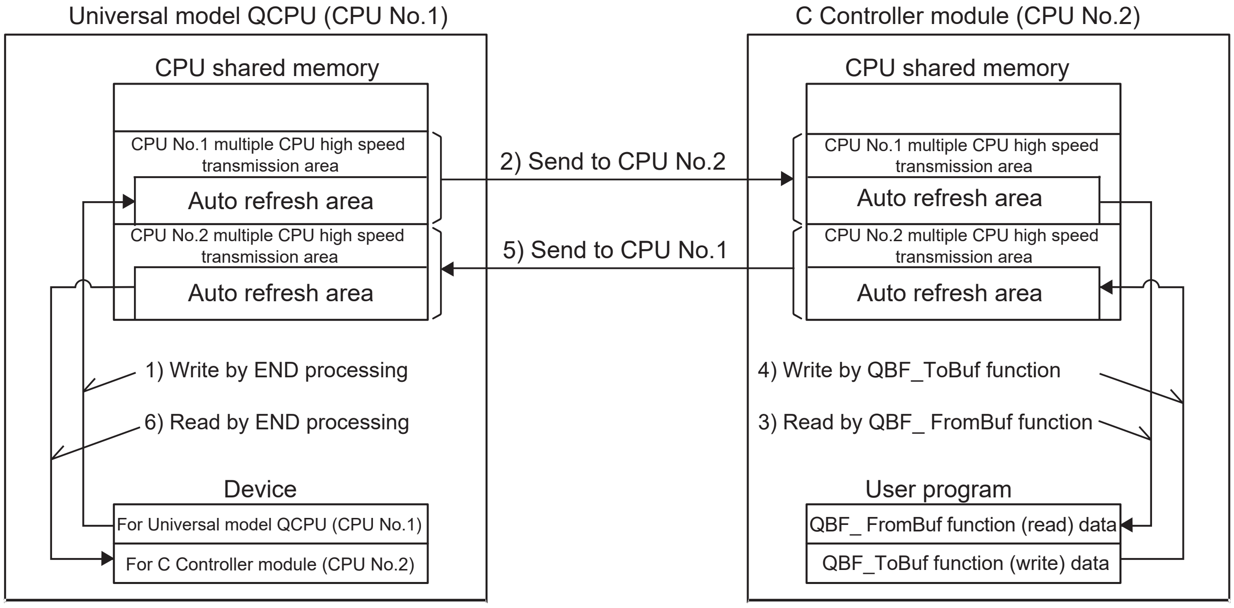
(2) Communication using auto refresh

(a) Outline of auto refresh

Auto refresh is a communication method using the auto refresh area in the multiple CPU high speed transmission area of the CPU shared memory.
Data written to the auto refresh area in the multiple CPU high speed transmission area are sent to another auto refresh area of another CPU at regular intervals (multiple CPU high speed transmission cycle).

To set the communication using auto refresh, open the <<Multiple CPU settings>> tab in C Controller setting utility and set "Multiple CPU high speed transmission area setting".

The auto refresh allows a CPU to use another CPU's device data, which have been received automatically.
Figure 13.17 shows an auto refresh operation of a system in which a Universal model QCPU and a C Controller module is set as CPU No.1 and CPU No.2 respectively.


Universal model QCPU (CPU No.1)
CPU shared memory
CPU No.1 multiple CPU high speed transmission area
Auto refresh area
CPU No.2 multiple CPU high speed transmission area
Auto refresh area
1) Write by END processing
6) Read by END processing
Device
For Universal model QCPU (CPU No.1)
For C Controller module (CPU No.2)
2) Send to CPU No.2
5) Send to CPU No.1
C Controller module (CPU No.2)
CPU shared memory
CPU No.1 multiple CPU high speed transmission area
Auto refresh area
CPU No.2 multiple CPU high speed transmission area
Auto refresh area
4) Write by QBF_ToBuf function
3) Read by QBF_ FromBuf function
User program
QBF_ FromBuf function (read) data
QBF_ToBuf function (write) data

Procedure for the C Controller module (CPU No.2) to read out device data of the Universal module QCPU (CPU No.1)

1) Data for the Universal model QCPU (CPU No.1) are transferred to its own auto refresh area in the END processing of the Universal model QCPU (CPU No.1).
2) The multiple CPU high speed transmission area data in the Universal model QCPU (CPU No.1) are sent to the C Controller module (CPU No.2).
3) The received data are read out into the user program of the C Controller module (CPU No.2)by using the QBF_FromBuf function.

Procedure for the Universal model QCPU (CPU No.1) to read out the user program data of the C Controller module (CPU No.2)

4) Data are written to its own auto refresh area by using the QBF_ToBuf function in the user program of the C Controller module (CPU No.2).
5) The multiple CPU high speed transmission area data in the C Controller module (CPU No.2) are transferred to the Universal model QCPU (CPU No.1).
6) The received data are transferred to the device for the C Controller module (CPU No.2) in the END processing of the Universal model QCPU (CPU No.1).

Figure 13.17 Auto refresh in multiple CPU high speed transmission

(b) Auto refresh execution

Auto refresh is executed when the CPU module is in the RUN, STOP, or PAUSE status.

(c) Memory structure of the multiple CPU high speed transmission area
The following explains memory structure of the multiple CPU high speed transmission area in the CPU shared memory.(☞ Page 13-14, Section 13.3.1)

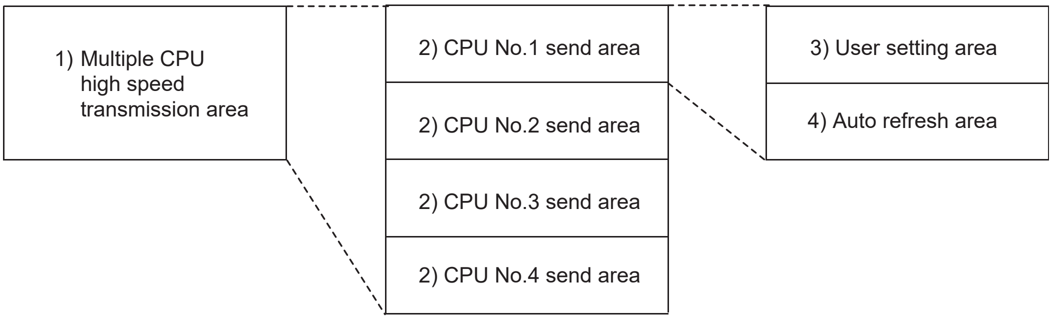


Figure 13.18 Memory structure of the multiple CPU high speed transmission area

Table 13.12 Multiple CPU high speed transmission area

No.	Name	Description	Size	
			Setting range	Setting unit
1)	Multiple CPU high speed transmission area	• The area for data communications between CPU modules in a multiple CPU system. • An area of up to 14K words is divided into sub-areas for respective CPU modules in a multiple CPU system.	0 to 14K words	1K word
2)	CPU No.n send area	• The area where send data of each CPU module are stored. • Data stored in the host CPU's send area are sent to another CPU. • Data received from another CPU are stored in the send area for another CPU.	0 to 14K words	1K word
3)	User setting area	• The area for communicating data with other CPUs using cyclic transmission area devices. • Accesses are performed with the user program that uses cyclic transmission area devices.	0 to 14K words	2 words
4)	Auto refresh area	• The area for communicating device data with other CPUs using auto refresh.	0 to 14K words	2 words

(d) Settings required for auto refresh
To perform auto refresh, the points of data to be sent from each CPU module must be set in "Multiple CPU high speed transmission area setting" in <<Multiple CPU settings>> of C Controller setting utility.

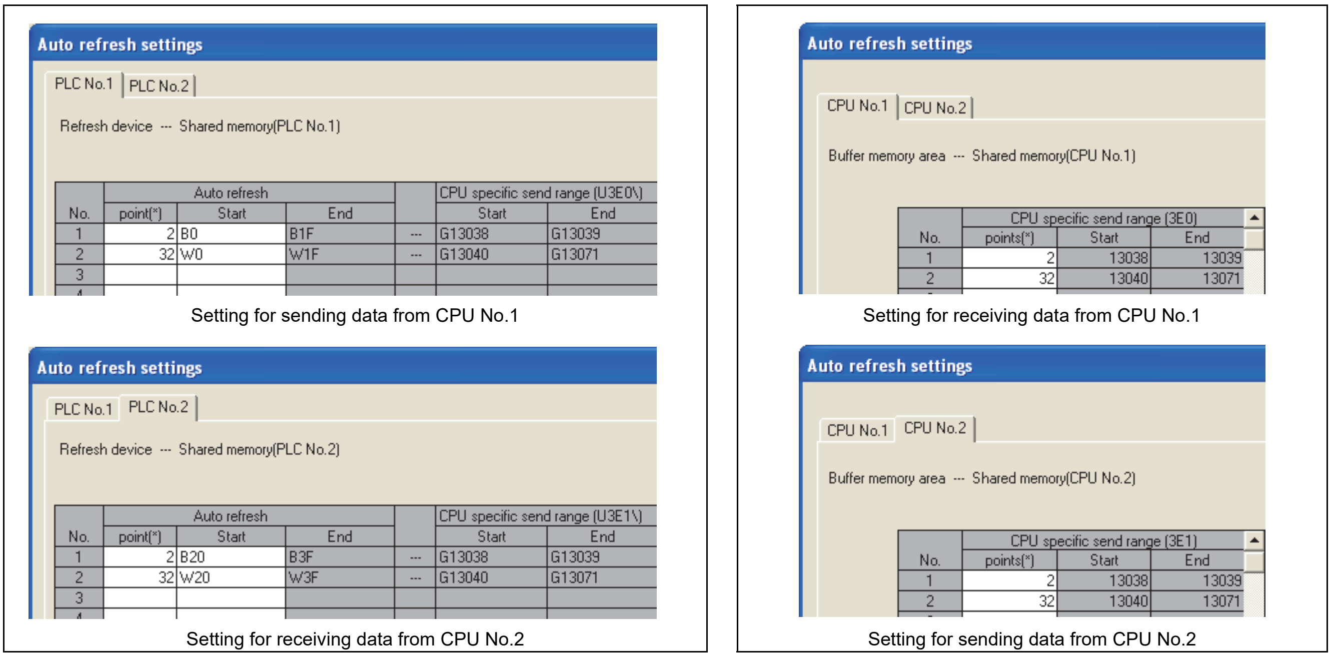
Remark

For C Controller setting utility, refer to the following.
☞ C Controller Module User's Manual (Utility Operation, Programming)

(e) Setting example

When two auto refresh setting ranges are configured for a system comprising of a Universal model QCPU (CPU No.1) and a C Controller module (CPU No.2), the send and receive areas in each auto refresh area are as illustrated below.

If auto refresh settings in Figure 13.19 are configured for each CPU, the auto refresh areas of the Universal model QCPU (CPU No.1) and C Controller module are allocated as shown in Figure 13.20.



Auto refresh settings

PLC No.1 | PLC No.2

Refresh device --- Shared memory(PLC No.1)

No.	Auto refresh				CPU specific send range (U3E0\)	
	point(*)	Start	End		Start	End
1	2	B0	B1F	---	G13038	G13039
2	32	W0	W1F	---	G13040	G13071
3						

Setting for sending data from CPU No.1

Auto refresh settings

PLC No.1 | PLC No.2

Refresh device --- Shared memory(PLC No.2)

No.	Auto refresh				CPU specific send range (U3E1\)	
	point(*)	Start	End		Start	End
1	2	B20	B3F	---	G13038	G13039
2	32	W20	W3F	---	G13040	G13071
3						

Setting for receiving data from CPU No.2

Auto refresh setting for Universal model QCPU (CPU No.1)

Auto refresh settings

CPU No.1 | CPU No.2

Buffer memory area --- Shared memory(CPU No.1)

No.	CPU specific send range (3E0)		
	points(*)	Start	End
1	2	13038	13039
2	32	13040	13071

Setting for receiving data from CPU No.1

Auto refresh settings

CPU No.1 | CPU No.2

Buffer memory area --- Shared memory(CPU No.2)

No.	CPU specific send range (3E1)		
	points(*)	Start	End
1	2	13038	13039
2	32	13040	13071

Setting for sending data from CPU No.2

Auto refresh setting for C Controller module (CPU No.2)

Figure 13.19 Auto refresh setting example

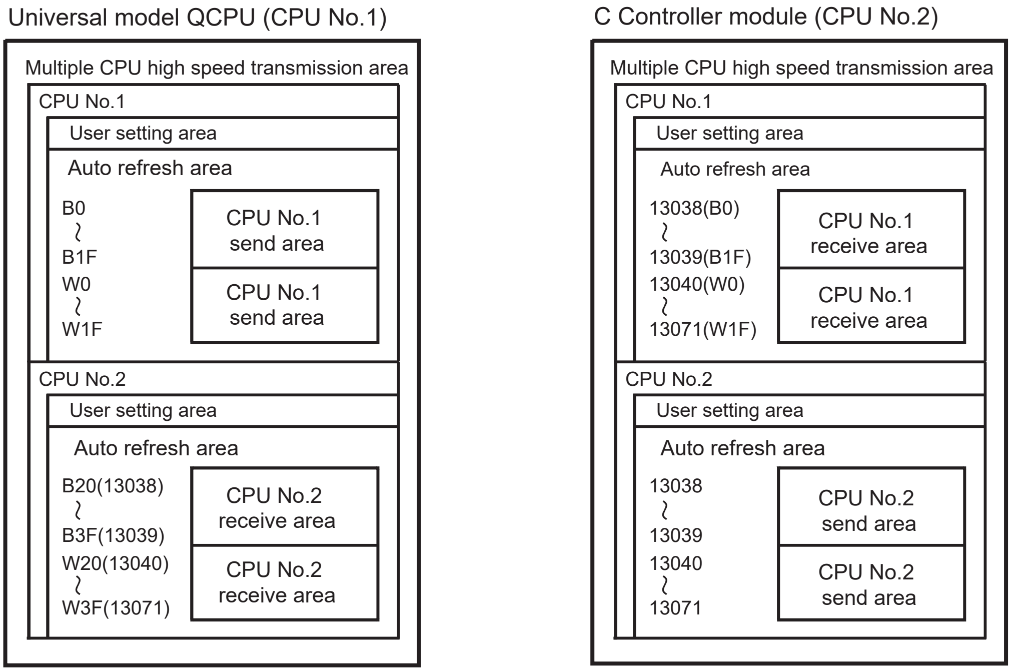


Figure 13.20 Receive and send areas in the auto refresh areas

(3) Precautions

(a) Transmission delay time

A transmission delay caused by auto refresh will be within the following range.

0.09 ms to (1.80 + (Sending-side write time[*1] + Receiving-side reading cycle[*2] × 2)) ms

[*1] For the write time of the QBF_ToBuf function, refer to the following.

☞ Appendix 1

[*2] When the receiving side is a C Controller module, there is no reading cycle (0ms).
When the receiving side is other than a C Controller module, it is the scan time on the receiving side.

(b) Assurance of data sent among CPUs

Depending on the timing of sending data from the host CPU and auto-refreshing data in another CPU, new and old data may be mixed in the area for each CPU (data separation).

The following is how to avoid such data separation in communications using auto refresh.

1) Preventing 32-bit data from being fragmented

Data are sent in units of 32 bits in auto refresh.

Since auto refresh settings are configured in units of 32 bits, 32-bit data are not fragmented.

2) Preventing more than 32-bit data from being fragmented

In the auto refresh method, data are sent in order of decreasing transfer numbers set as the auto refresh setting parameters.

Assigning a transfer number smaller than actual transfer numbers to an interlock device.

An interlock program example is shown in (3) (c) in this section below.

(c) Interlock program example

Figure 13.21 is an interlock program example for sending data from CPU No.1 (programmable controller CPU) to CPU No.2 (C Controller module).

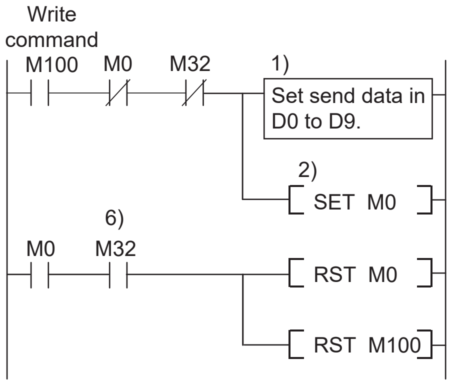
Table 13.13 Parameter setting example for interlock program

Auto refresh setting for CPU No.1							Direction	Auto refresh setting for CPU No.2				
CPU No.	Transfer No.	CPU specific send range			Device setting			CPU No.	Transfer No.	CPU specific send range		
		Points	Start	End	Start	End				Points	Start	End
CPU No.1	Transfer 1	2	0	1	M0	M31	→	CPU No.1	Transfer 1	2	0	1
	Transfer 2	10	2	11	D0	D9			Transfer 2	10	2	11
CPU No.2	Transfer 1	2	0	1	M32	M63	←	CPU No.2	Transfer 1	2	0	1

The parameter setting in Table 13.13 uses the following interlock devices.

- M0 as an interlock device (Data set completion bit) of CPU No.1
- M32 as an interlock device (Data processing completion bit) of CPU No.2

Sending side program for CPU No.1
(programmable controller CPU)



Receiving side program for CPU No.2 (C Controller module)

```
{
                          :
  unsigned short    usBuf;                  /* Data storage destination */
  unsigned short    pusData[128];           /* Data storage destination */
                          :
   /***********************************************************************/
   /* Confirm that 1 is set in the area corresponding to M0 in CPU No.1 auto refresh area.  */
   /* (CPU No.1 data set completion bit turns ON.)                        */
   /***********************************************************************/
   while(1){
      sRet = QBF_FromBuf(lPath, 0x3E0, 10000, 1, &usBuf, 1);             3)
      if( sRet != 0) return(sRet);
      if( (usBuf & 0x1) == 0x1 ) break;      /* When 1 is set, processing gets out from the loop. */
      taskDelay(2);
   }

   /*****************************************************************/
   /* Data are read from the area corresponding to D0 to D9 in the CPU No.1 auto refresh area. */
   /*****************************************************************/
   sRet = QBF_FromBuf(lPath, 0x3E0, 10002, 10, pusData, 128);            4)
   if( sRet != 0) return(sRet);

   /*****************************************************************/
   /* 1 is set in the host CPU's auto refresh area corresponding to M32 of CPU No.1.   */
   /* (CPU No.2 data processing completion bit turns ON.)  */
   /*****************************************************************/
   usBuf = 1;
   sRet = QBF_ToBuf(lPath, 0x3E1, 10000, 1, &usBuf, 1);                  5)
   if( sRet != 0) return(sRet);

   /***********************************************************************/
   /* Confirm that 0 is set in the area corresponding to M0 in CPU No.1 auto refresh area.  */
   /* (CPU No.1 data set completion bit turns OFF.)                       */
   /***********************************************************************/
   while(1){
      sRet = QBF_FromBuf(lPath, 0x3E0, 10000, 1, &usBuf, 1);             7)
      if( sRet != 0) return(sRet);
      if( (usBuf & 0x1) == 0x0 ) break;      /* When 0 is set, processing gets out from the loop. */
      taskDelay(2);
   }

   /*****************************************************************/
   /* 0 is set in the host CPU's auto refresh area corresponding to M32 of CPU No.1.  */
   /* (CPU No.2 data processing completion bit turns OFF.) */
   /*****************************************************************/
   usBuf = 0;
   sRet = QBF_ToBuf(lPath, 0x3E1, 10000, 1, &usBuf, 1);                  8)
   if( sRet != 0) return(sRet);
}
```

Figure 13.21 Interlock program example

1) CPU No.1 stores send data in D0 to D9.

2) CPU No.1 turns on the data set completion bit (M0).
   - In the END processing of CPU No.1, the above data are written to the auto refresh area in its CPU No.1 send area.
   - The auto refresh area data in the CPU No.1 send area are sent to CPU No.2.

By the user program of CPU No.2, data are read from the CPU No.1 auto refresh area to the internal buffer.

3) CPU No.2 detects ON of the data set completion bit (M0).

4) CPU No.2 processes the received data.

5) CPU No.2 turns on the data processing completion bit (this information is refreshed to M32 of CPU No.1.)

In the END processing of CPU No.1, the data are read from the auto refresh area in its CPU No.2 send area to the specified device.

6) CPU No.1 detects ON of the data processing completion bit (M32), and turns off the data set completion bit (M0).

By the user program of CPU No.2, check that the data set completion bit (M0) of CPU No.1 is OFF.

7) CPU No.2 detects OFF of the data set completion bit (M0).

8) CPU No.2 turns off the data processing completion bit (this information is refreshed to M32 of CPU No.1.)

13.3.4 Data communications without using auto refresh

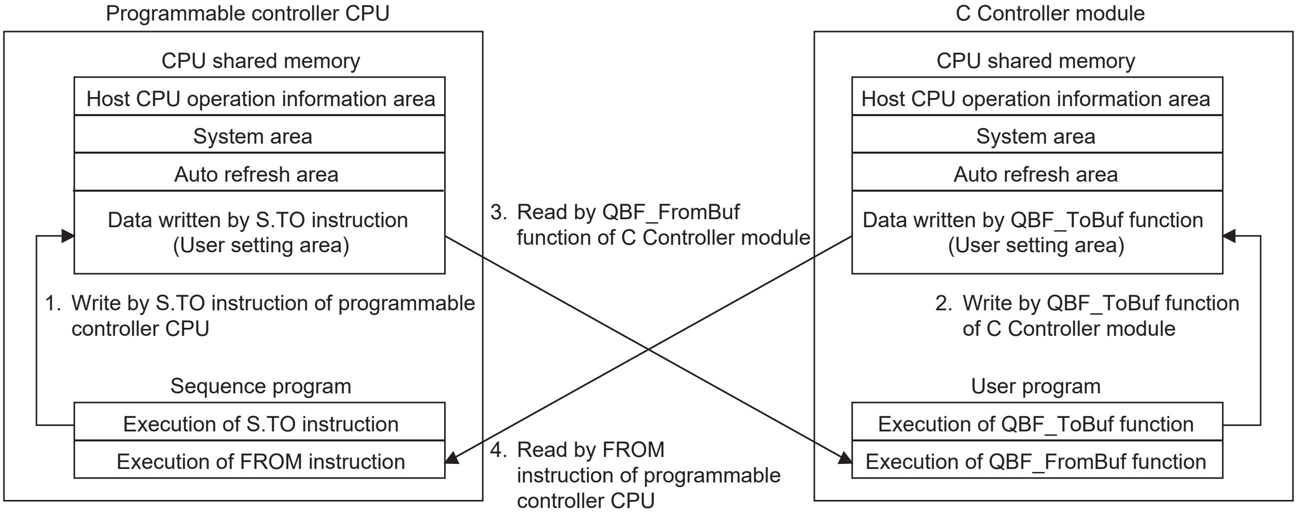
Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	○

This section explains processing outline of data communications in the CPU shared memory without using auto refresh of the programmable controller CPU.

(1) Data communications without using auto refresh

When auto refresh is not used, data are communicated as described below.

(a) Between programmable controller CPU and C Controller module


Programmable controller CPU
CPU shared memory
Host CPU operation information area
System area
Auto refresh area
Data written by S.TO instruction (User setting area)
1. Write by S.TO instruction of programmable controller CPU
Sequence program
Execution of S.TO instruction
Execution of FROM instruction
3. Read by QBF_FromBuf function of C Controller module
4. Read by FROM instruction of programmable controller CPU
C Controller module
CPU shared memory
Host CPU operation information area
System area
Auto refresh area
Data written by QBF_ToBuf function (User setting area)
2. Write by QBF_ToBuf function of C Controller module
User program
Execution of QBF_ToBuf function
Execution of QBF_FromBuf function

When executing sequence program of programmable controller CPU

1. By the S.TO instruction, the data are written to the user setting area in the CPU shared memory of the programmable controller CPU.
4. By the FROM instruction, the data are read out to the specified device in the programmable controller CPU.

At bus interface function execution of C Controller module

2. QBF_ToBuf function execution writes the data to the user setting area in the C Controller module's CPU shared memory.
3. QBF_FromBuf function execution reads out the user setting area data of the programmable controller CPU onto the user program.

Figure 13.22 Processing of communication without using auto refresh

Remark

Figure 13.22 shows an operation example of using the S.TO and FROM instructions on a High Performance model CPU.
For operations on the programmable controller CPU side when not using auto refresh, refer to the manual for the relevant CPU module.

<Interlock program example>

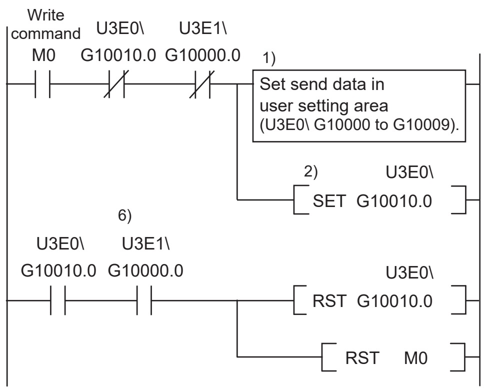
Depending on the following timing, old and new data for each CPU may be mixed (data separation).

- Writing data to the user setting area in another CPU and reading another CPU's user setting area data from the host CPU
- Writing data to the user setting area in the host CPU and reading host CPU's user setting area data from another CPU

The following explains how to avoid data separation in data communications through user setting areas.

Figure 13.23 shows an example, in which an interlock is programmed for sending data from CPU No.1 (programmable controller CPU) to CPU No.2 (C Controller module) using user setting areas in the multiple CPU high speed transmission areas.

Sending side program for CPU No.1 (programmable controller CPU)



Receiving side program for CPU No.2 (C Controller module)

```
{
                              :
  unsigned short   usBuf;              /* Data storage destination */
  unsigned short   pusData[128];       /* Data storage destination */
                              :
  /****************************************************************/
  /* The ON status of CPU No.1 data set completion bit is detected.*/
  /****************************************************************/
  while(1){
     sRet = QBF_FromBuf(lPath, 0x3E0, 10010, 1, &usBuf, 1);           ←   3)
     if( sRet != 0) return(sRet);
     if( (usBuf & 0x1) == 0x1 ) break;     /* When 1 is set, processing gets out from the loop. */
     taskDelay(2);
  }

  /****************************************************************/
  /* Data are read out from the CPU No.1 user setting area.      */
  /****************************************************************/
  sRet = QBF_FromBuf(lPath, 0x3E0, 10000, 10, pusData, 128);            ←   4)
  if( sRet != 0) return(sRet);

  /****************************************************************/
  /* CPU No.2 data processing completion bit turns ON.    */
  /****************************************************************/
  usBuf = 1;
  sRet = QBF_ToBuf(lPath, 0x3E1, 10000, 1, &usBuf, 1);                  ←   5)
  if( sRet != 0) return(sRet);

  /****************************************************************/
  /* The OFF status of CPU No.1 data set completion bit is detected.*/
  /****************************************************************/
  while(1){
     sRet = QBF_FromBuf(lPath, 0x3E0, 10010, 1, &usBuf, 1);           ←   7)
     if( sRet != 0) return(sRet);
     if( (usBuf & 0x1) == 0x0 ) break;     /* When 0 is set, processing gets out from the loop. */
     taskDelay(2);
  }

  /****************************************************************/
  /* CPU No.2 data processing completion bit turns OFF.  */
  /****************************************************************/
  usBuf = 0;
  sRet = QBF_ToBuf(lPath, 0x3E1, 10000, 1, &usBuf, 1);                  ←   8)
  if( sRet != 0) return(sRet);
}
```

Figure 13.23 Interlock program example

1) CPU No.1 writes send data to the user setting area.

2) CPU No.1 writes the data set completion bit ON to the user setting area.

The multiple CPU high speed transmission area data of CPU No.1 are sent to CPU No.2.

3) CPU No.2 detects ON of the data set completion bit.

4) CPU No.2 processes the received data.

5) CPU No.2 writes the data processing completion bit ON to the user setting area.

The multiple CPU high speed transmission area data of CPU No.2 are sent to CPU No.1.

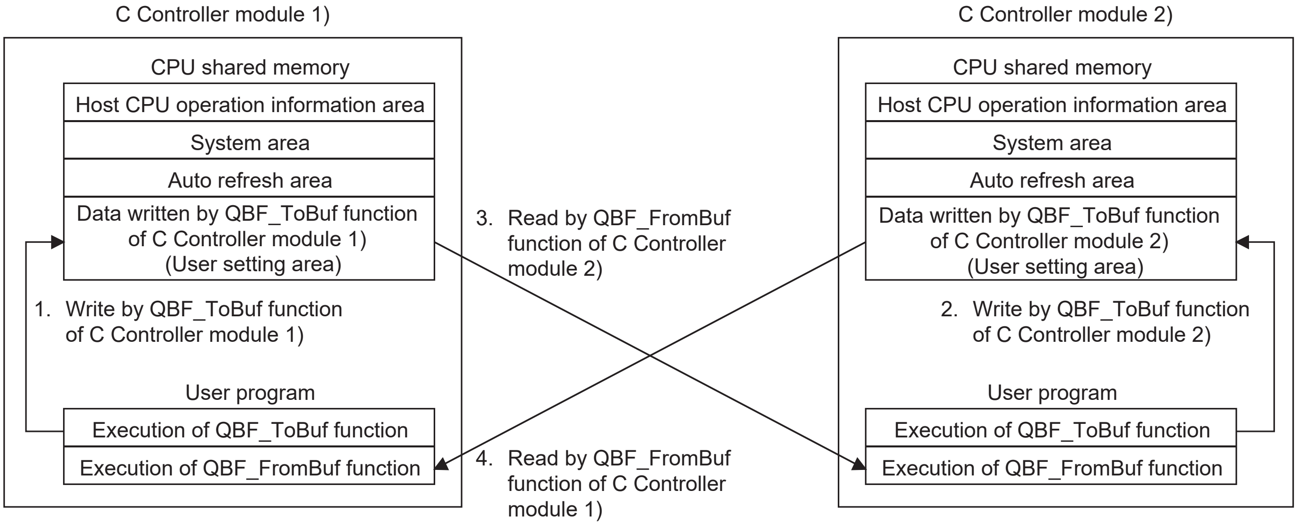
6) CPU No.1 detects ON of the data processing completion bit, and turns OFF the data set completion bit.

The multiple CPU high speed transmission area data of CPU No.1 are sent to CPU No.2.

7) CPU No.2 detects OFF of the data set completion bit.

8) CPU No.2 writes the data processing completion bit OFF to the user setting area.

(b) Between C Controller modules


C Controller module 1)
CPU shared memory
Host CPU operation information area
System area
Auto refresh area
Data written by QBF_ToBuf function of C Controller module 1) (User setting area)
1. Write by QBF_ToBuf function of C Controller module 1)
User program
Execution of QBF_ToBuf function
Execution of QBF_FromBuf function
3. Read by QBF_FromBuf function of C Controller module 2)
4. Read by QBF_FromBuf function of C Controller module 1)
C Controller module 2)
CPU shared memory
Host CPU operation information area
System area
Auto refresh area
Data written by QBF_ToBuf function of C Controller module 2) (User setting area)
2. Write by QBF_ToBuf function of C Controller module 2)
User program
Execution of QBF_ToBuf function
Execution of QBF_FromBuf function

At bus interface function execution of C Controller module 1)

1. QBF_ToBuf function execution writes the data to the user setting area in the CPU shared memory of C Controller module 1).
4. QBF_FromBuf function execution reads out the user setting area data of C Controller module 2) onto the user program.

At bus interface function execution of C Controller module 2)

2. QBF_ToBuf function execution writes the data to the user setting area in the CPU shared memory of C Controller module 2).
3. QBF_FromBuf function execution reads out the user setting area data of C Controller module 1) onto the user program.

Figure 13.24 Communications without using auto refresh

<Interlock program example>

Depending on the following timing, old and new data for each CPU may be mixed (data separation).

- Writing data to the user setting area in another CPU and reading another CPU's user setting area data from the host CPU
- Writing data to the user setting area in the host CPU and reading host CPU's user setting area data from another CPU

The following explains how to avoid data separation in data communications through user setting areas.

Figure 13.25 shows an example, in which an interlock is programmed for sending data from CPU No.1 (C Controller module) to CPU No.2 (C Controller module) using user setting areas in the multiple CPU high speed transmission areas.

Sending side program for CPU No.1 (C Controller module)

```
{
                              :
   unsigned short   usBuf;              /* Data storage destination */
   unsigned short   pusData[128];       /* Data storage destination */
                              :
   /****************************************************************/
   /* CPU No.1 send data are written to the user setting area.     */
   /****************************************************************/
   sRet = QBF_ToBuf(lPath, 0x3E0, 10000, 10, pusData, 128);                ←──   1)
   if( sRet != 0) return(sRet);

   /****************************************************************/
   /* CPU No.1 data set completion bit turns ON.             */
   /****************************************************************/
   usBuf = 1;
   sRet = QBF_ToBuf(lPath, 0x3E0, 10010, 1, &usBuf, 1);                    ←──   2)
   if( sRet != 0) return(sRet);

   /****************************************************************/
   /* The ON status of CPU No.2 data processing completion bit is detected.*/
   /****************************************************************/
   while(1){
      sRet = QBF_FromBuf(lPath, 0x3E1, 10000, 1, &usBuf, 1);              ←──   6)
      if( sRet != 0) return(sRet);
      if( (usBuf & 0x1) == 0x0 ) break;    /* When 0 is set, processing gets out from the loop. */
      taskDelay(2);
   }

   /****************************************************************/
   /* CPU No.1 data set completion bit turns OFF.            */
   /****************************************************************/
   usBuf = 0;
   sRet = QBF_ToBuf(lPath, 0x3E0, 10010, 1, &usBuf, 1);                    ←──   7)
   if( sRet != 0) return(sRet);

}
```

Figure 13.25 Interlock program example

Receiving side program for CPU No.2 (C Controller module)

```
{
                          :
   unsigned short    usBuf;                  /* Data storage destination */
   unsigned short    pusData[128];           /* Data storage destination */
                          :
   /****************************************************************/
   /* The ON status of CPU No.1 data set completion bit is detected.*/
   /****************************************************************/
   while(1){
      sRet = QBF_FromBuf(lPath, 0x3E0, 10010, 1, &usBuf, 1);                ←——  3)
      if( sRet != 0) return(sRet);
     if( (usBuf & 0x1) == 0x1 ) break;       /* When 1 is set, processing gets out from the loop. */
      taskDelay(2);
   }

   /****************************************************************/
   /* Data are read out from CPU No.1 user setting area.   */
   /****************************************************************/
   sRet = QBF_FromBuf(lPath, 0x3E0, 10000, 10, pusData, 128);                ←——  4)
   if( sRet != 0) return(sRet);

   /****************************************************************/
   /* CPU No.2 data processing completion bit turns ON.  */
   /****************************************************************/
   usBuf = 1;
   sRet = QBF_ToBuf(lPath, 0x3E1, 10000, 1, &usBuf, 1);                      ←——  5)
   if( sRet != 0) return(sRet);

   /****************************************************************/
   /* The OFF status of CPU No.1 data set completion bit is detected.*/
   /****************************************************************/
   while(1){
      sRet = QBF_FromBuf(lPath, 0x3E0, 10010, 1, &usBuf, 1);                ←——  8)
      if( sRet != 0) return(sRet);
     if( (usBuf & 0x1) == 0x0 ) break;       /* When 0 is set, processing gets out from the loop. */
      taskDelay(2);
   }

   /****************************************************************/
   /* CPU No.2 data processing completion bit turns OFF. */
   /****************************************************************/
   usBuf = 0;
   sRet = QBF_ToBuf(lPath, 0x3E1, 10000, 1, &usBuf, 1);                      ←——  9)
   if( sRet != 0) return(sRet);
}
```

Figure 13.25 Interlock program example (continued)

1) CPU No.1 writes send data to the user setting area.

2) CPU No.1 writes the data set completion bit ON to the user setting area.

The multiple CPU high speed transmission area data of CPU No.1 are sent to CPU No.2.

3) CPU No.2 detects ON of the data set completion bit.

4) CPU No.2 processes the received data.

5) CPU No.2 writes the data processing completion bit ON to the user setting area.

The multiple CPU high speed transmission area data of CPU No.2 are sent to CPU No.1.

6) CPU No.1 detects ON of the data processing completion bit.

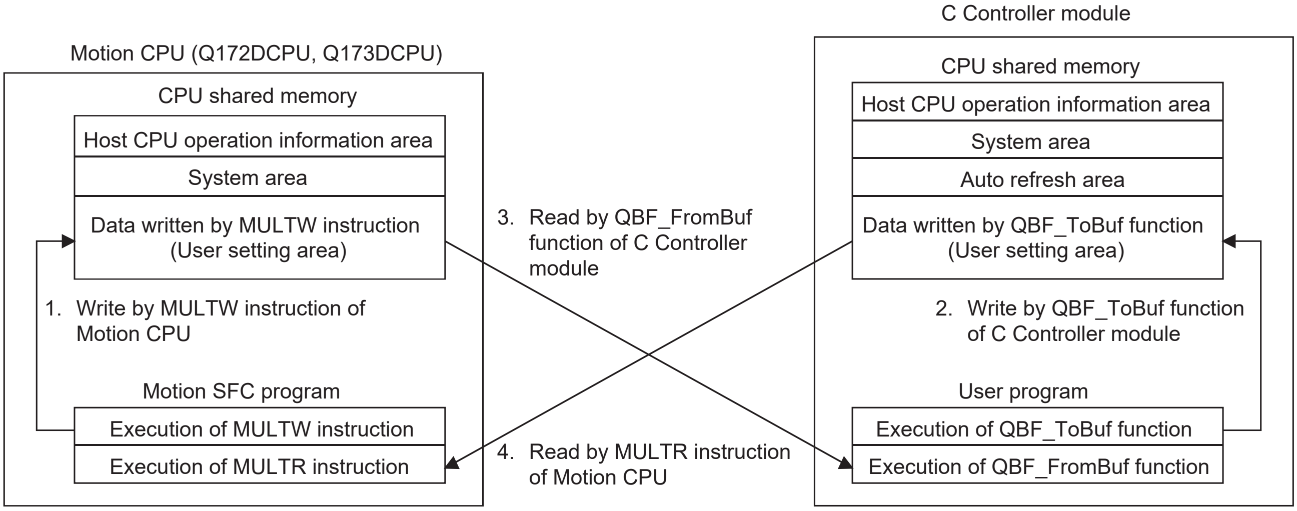
7) CPU No.1 turns OFF the data set completion bit.

The multiple CPU high speed transmission area data of CPU No.1 are sent to CPU No.2.

8) CPU No.2 detects OFF of the data set completion bit.

9) CPU No.2 writes data of the data processing completion bit OFF to the user setting area.

(c) Between Motion CPU (Q172DCPU or Q173DCPU) and C Controller module


C Controller module
Motion CPU (Q172DCPU, Q173DCPU)
CPU shared memory
Host CPU operation information area
System area
Data written by MULTW instruction (User setting area)
1. Write by MULTW instruction of Motion CPU
Motion SFC program
Execution of MULTW instruction
Execution of MULTR instruction
3. Read by QBF_FromBuf function of C Controller module
4. Read by MULTR instruction of Motion CPU
CPU shared memory
Host CPU operation information area
System area
Auto refresh area
Data written by QBF_ToBuf function (User setting area)
2. Write by QBF_ToBuf function of C Controller module
User program
Execution of QBF_ToBuf function
Execution of QBF_FromBuf function

When executing motion SFC program of Motion CPU

1. By the MULTW instruction, the data are written to the user setting area in the CPU shared memory of the Motion CPU.
4. By the MULTR instruction, the user setting area data of the C Controller module are read out to the specified device in the Motion CPU.

At bus interface function execution of C Controller module

2. QBF_ToBuf function execution writes the data to the user setting area in the C Controller module's CPU shared memory.
3. QBF_FromBuf function execution reads out the user setting area data of the Motion CPU onto the user program.

Figure 13.26 Communication without using auto refresh

Remark

1. For communications between the Q12DCCPU-V and the Q172DCPU or Q173DCPU, use the auto refresh area in the multiple CPU high speed transmission area. (☞ Page 13-24, Section 13.3.3)
   For communications between the Q06CCPU-V(-B) and a Motion CPU other than the Q172DCPU and Q173DCPU, use the auto refresh area. (☞ Page 13-17, Section 13.3.2)
2. For instructions of Motion CPUs, refer to the relevant Motion CPU manual.

(2) Addresses of the user setting area and multiple CPU high speed transmission area

Data communications not using auto refresh of the programmable controller CPU are performed by using the following areas in each CPU shared memory.

- User setting area
- User setting area in the multiple CPU high speed transmission area

1) The user setting area in the multiple CPU high speed transmission area is available for the following CPU modules only.
    - Universal model QCPU (except for the Q00UCPU, Q01UCPU, and Q02UCPU)
    - Motion CPU (Q172DCPU, Q173DCPU)
    - Q12DCCPU-V
    - Q24DHCCPU-V/-LS

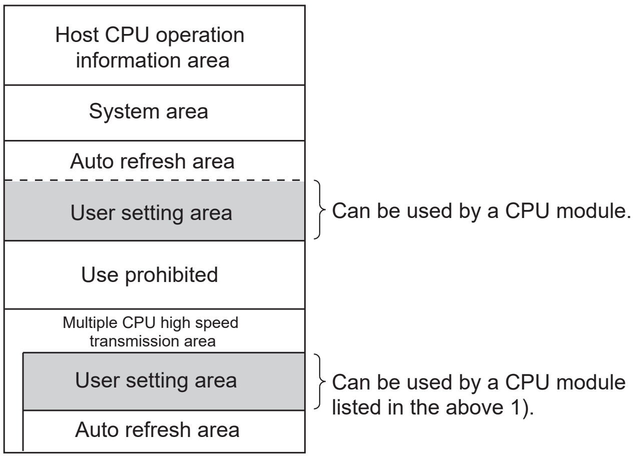

Host CPU operation information area
System area
Auto refresh area
User setting area
Can be used by a CPU module.
Use prohibited
Multiple CPU high speed transmission area
User setting area
Can be used by a CPU module listed in the above 1).
Auto refresh area

Figure 13.27 CPU shared memory available for data communication not using auto refresh

(a) Addresses of the user setting area

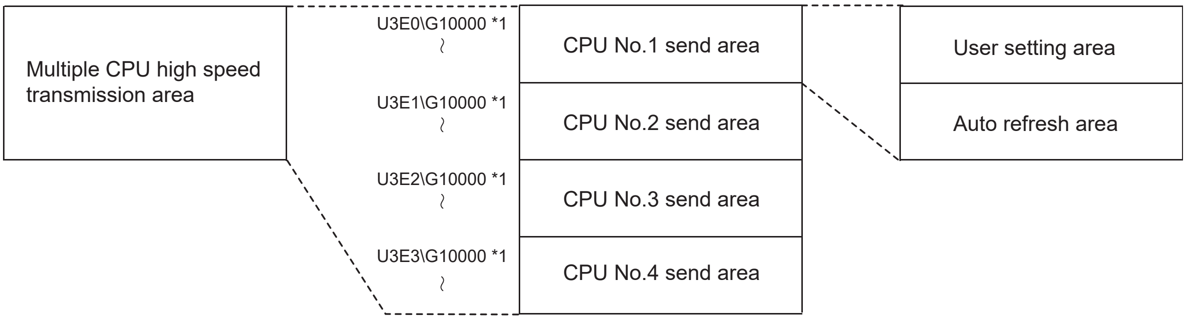
Addresses of the user setting areas vary depending on the CPU module used. For those of C Controller modules, refer to the following.

☞ Page 13-14, Section 13.3.1

(b) Addresses of the multiple CPU high speed transmission area

Figure 13.28 shows the addresses in the multiple CPU high speed transmission area.

The last address of each CPU's send area shown in Figure 13.28 varies depending on the points set in "CPU specific send range" in the "Multiple CPU settings" tab.



* 1 This figure indicates the addresses when specifying the User setting area for each CPU by the multiple CPU device (Specified by the Universal model QCPU)

Figure 13.28 Memory configuration of the Multiple CPU high speed transmission area

For each area of the Multiple CPU high speed transmission area, refer to the following.

☞ Page 13-24, Section 13.3.3

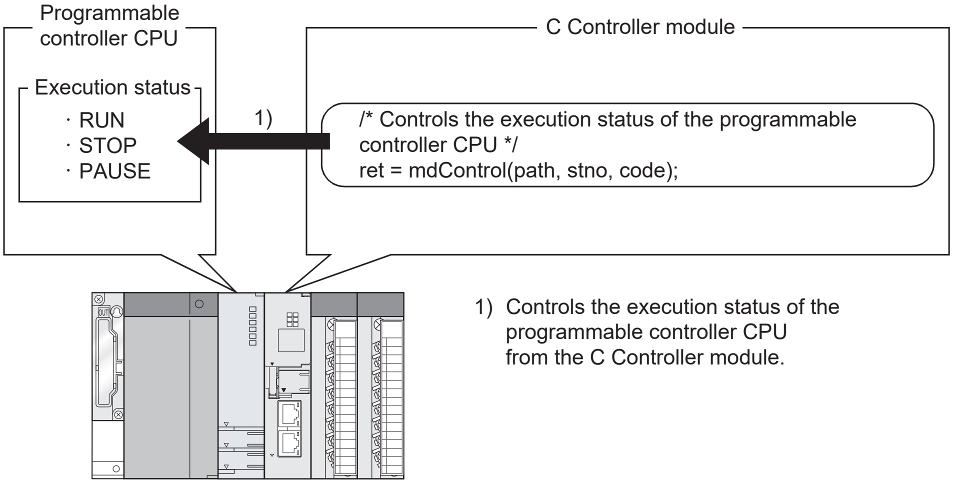
13.4 Programmable Controller Remote Control Function

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	×

(1) Programmable controller remote controller function

This function allows the C Controller module to control the execution status of a programmable controller CPU.

Use a MELSEC data link function when creating a user program of the C Controller module.



```
/* Controls the execution status of the programmable
controller CPU */
ret = mdControl(path, stno, code);
```

1) Controls the execution status of the programmable controller CPU from the C Controller module.

Figure 13.29 Programmable controller remote control

Remark

For the MELSEC data link functions, refer to the following.

☞ C Controller Module User's Manual (Utility Operation, Programming)

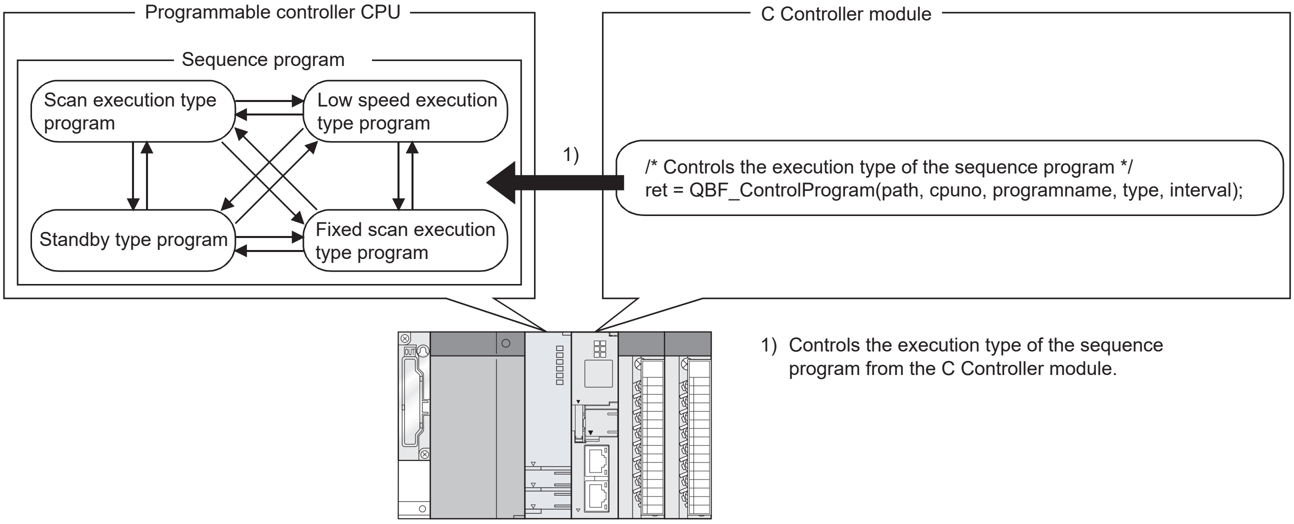
13.5 Sequence Program Control Function

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	○

(1) Sequence program control function

This function allows the C Controller module to control (change) the sequence program execution type of a programmable controller CPU.

Use a bus interface function when creating a user program for the C Controller module.


Programmable controller CPU
Sequence program
Scan execution type program
Low speed execution type program
Standby type program
Fixed scan execution type program
1)
C Controller module

```
/* Controls the execution type of the sequence program */
ret = QBF_ControlProgram(path, cpuno, programname, type, interval);
```

1) Controls the execution type of the sequence program from the C Controller module.

Figure 13.30 Sequence program control

Remark

1) For the bus interface functions, refer to the following.

☞ C Controller Module User's Manual (Utility Operation, Programming)

2) For this function, use a High Performance model QCPU.

13.6 Issuing an Interrupt to Another CPU

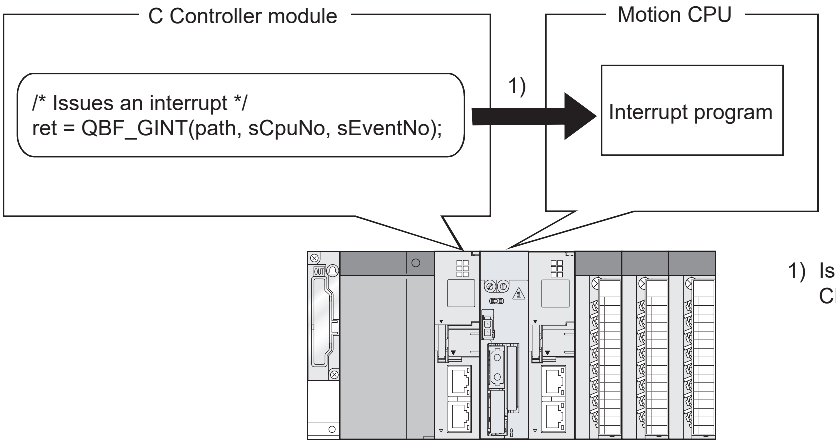
Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	△	△

(1) About the function of issuing an interrupt to another CPU

This function allows the C Controller module to issue an interrupt to another CPU (Motion CPU or C Controller module) from its own user program.
No interrupt can be issued to a programmable controller CPU.

(a) Interrupt to Motion CPU

When an interrupt is issued to a Motion CPU, the Motion CPU starts an interrupt program.



```
/* Issues an interrupt */
ret = QBF_GINT(path, sCpuNo, sEventNo);
```

1) Issues an interrupt to the Motion CPU from the C Controller module.

Figure 13.31 Interrupt to the Motion CPU

☒**POINT**

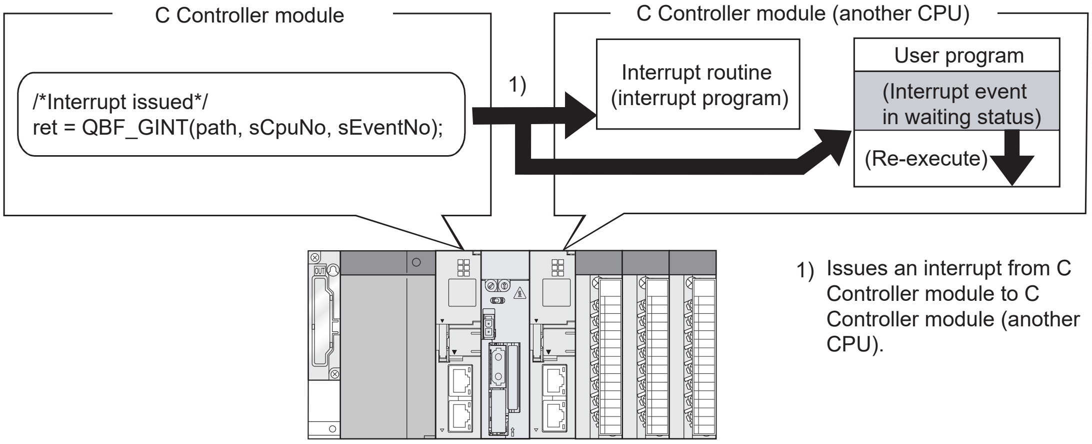
To apply this function to the Q172DCPU or Q173DCPU, the C Controller module must be the Q12DCCPU-V.

(b) Interrupt to C Controller module (another CPU)

When an interrupt is issued to a C Controller module (another CPU), the C Controller module (another CPU) performs the following processing.

- Executes a routine registered by the QBF_EntryCpuInt function as an interrupt routine (interrupt program).*1
- Resumes the user program that is waiting for an interrupt event by the QBF_WaitEvent function.

* 1 The following C Controller modules cannot execute it as an interrupt routine (interrupt program).
- Q12DCCPU-V with a serial number whose first five digits are "12041" or earlier
- Q06CCPU-V(-B)



1) Issues an interrupt from C Controller module to C Controller module (another CPU).

Figure 13.32 Interrupt to C Controller module (another CPU)

(2) Function

The following function is used for an interrupt to another CPU (Motion CPU or C Controller module).

Figure 13.33 Function used for an interrupt to another CPU (Motion CPU or C Controller module)

Function name	Function
QBF_GINT	Issues an interrupt to another CPU. This is equivalent to the S(P).GINT or D(P).GINT instruction of a programmable controller CPU.

Remark

For the bus interface functions, refer to the following.

☞ C Controller Module User's Manual (Utility Operation, Programming)

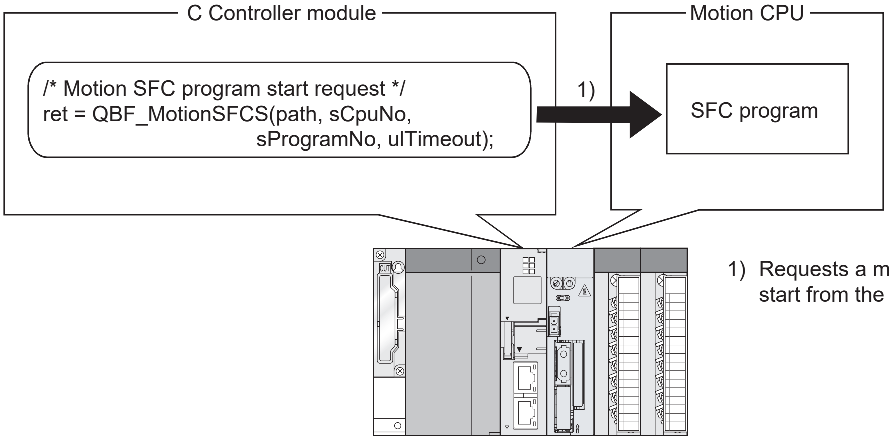
13.7 Motion CPU Control Instruction

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	△	△

(1) Motion CPU control instruction

Control instructions can be given to a Motion CPU from a user program of the C Controller module.

Use bus interface functions when creating a user program of the C Controller module.



```
/* Motion SFC program start request */
ret = QBF_MotionSFCS(path, sCpuNo,
                     sProgramNo, ulTimeout);
```

1) Requests a motion SFC program start from the C Controller module.

Figure 13.34 Motion CPU control instruction

POINT

To apply this function to the Q172DCPU or Q173DCPU, the C Controller module must be the Q12DCCPU-V.

Remark

For the bus interface functions, refer to the following.

☞ C Controller Module User's Manual (Utility Operation, Programming)

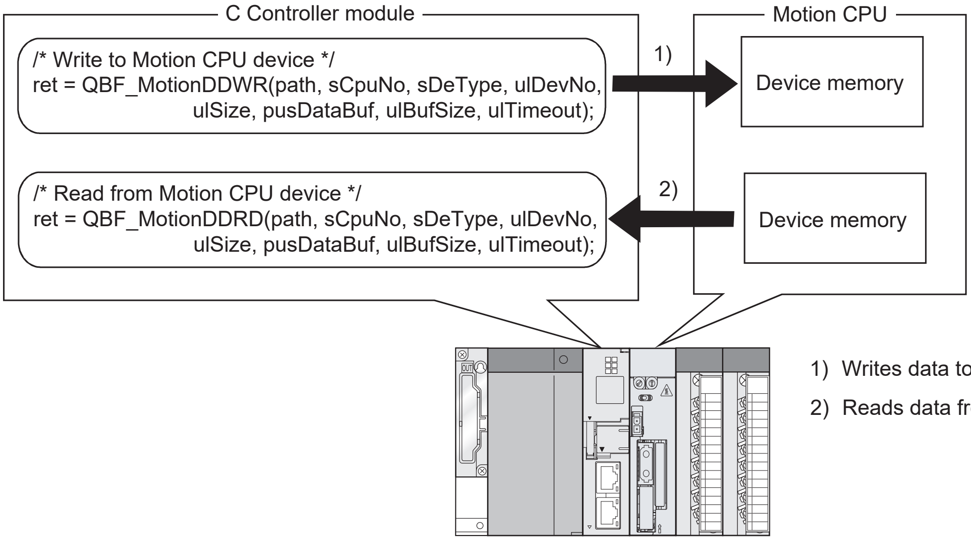
13.8 Motion CPU Device Access

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	△	△

(1) Motion CPU device access

From a user program of the C Controller module, data can be written to and read from Motion CPU's devices.

Use the bus interface functions when creating a user program of the C Controller module.



```
/* Write to Motion CPU device */
ret = QBF_MotionDDWR(path, sCpuNo, sDeType, ulDevNo,
                ulSize, pusDataBuf, ulBufSize, ulTimeout);
```

```
/* Read from Motion CPU device */
ret = QBF_MotionDDRD(path, sCpuNo, sDeType, ulDevNo,
                ulSize, pusDataBuf, ulBufSize, ulTimeout);
```

1) Writes data to the Motion CPU device.

2) Reads data from the Motion CPU device.

Figure 13.35 Motion CPU device access

POINT

To apply this function to the Q172DCPU or Q173DCPU, the C Controller module must be the Q12DCCPU-V.

Remark

For the bus interface functions, refer to the following.

☞ C Controller Module User's Manual (Utility Operation, Programming)

13.9 Multiple CPU Synchronized Boot-Up

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	×	×

The multiple CPU synchronized boot-up function allows synchronized start-up of CPU modules No.1 to No.4.

Since start-up of each CPU can be monitored, an interlock program for checking start-up of another CPU before access is no more required.

Note that, however, the speed of system start-up may slow down because it depends on the slowest CPU module.

POINT

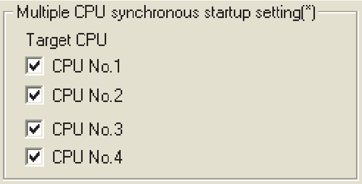
1. The multiple CPU synchronized boot-up function enables access to each CPU module in a multiple CPU system without using an interlock.
   This does not mean that the CPU modules start their operations at the same time after start-up.
2. The multiple CPU synchronized boot-up is available for the following CPU modules:
   - Universal model QCPU (except the Q00UCPU, Q01UCPU, and Q02UCPU)
   - Motion CPU (the Q172DCPU and Q173DCPU)
   - Q12DCCPU-V
   - Q24DHCCPU-V/-LS

(1) Multiple CPU synchronized boot-up setting

In the Multiple CPU settings tab of C Controller setting utility, check the boxes for any target CPUs (CPU 1 to CPU 4).

In the Multiple CPU synchronous startup setting area, all boxes for CPU No.1 to No.4 are checked by default. (The multiple CPU synchronized boot-up is enabled.)

To disable this function, clear the corresponding check boxes.



Multiple CPU synchronous startup setting(*)

Target CPU

☑ CPU No.1

☑ CPU No.2

☑ CPU No.3

☑ CPU No.4

Figure 13.36 Multiple CPU synchronized boot-up setting

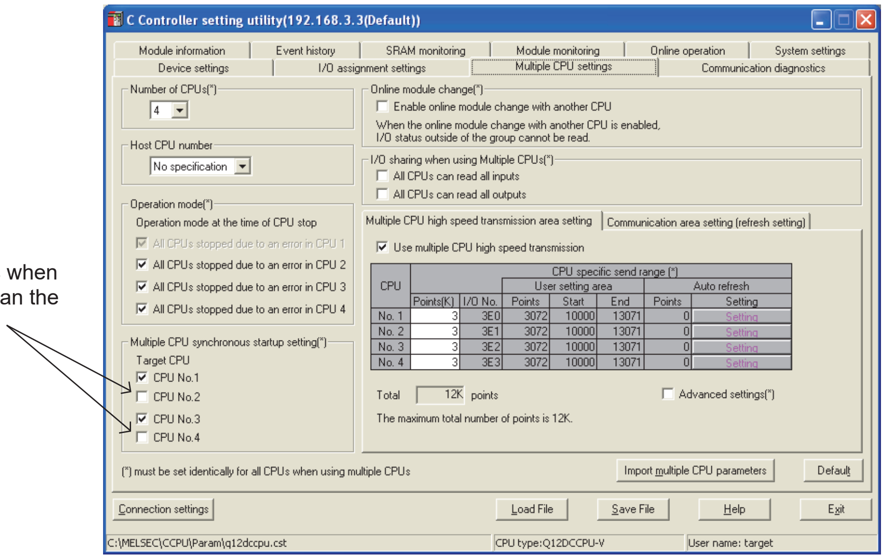
Set the same multiple CPU synchronized boot-up setting for each of all CPU modules comprising the multiple CPU system.

Otherwise, a parameter error (error code: 3015, PARAMETER ERROR) will be detected.

☒POINT

Only the CPU modules listed below support the multiple CPU synchronized boot-up.
For any other CPU model, clear the corresponding check box in the Multiple CPU synchronous startup setting area.
For example, if the Q06CCPU-Vs are used as CPU No.2 and No.4, clear the check boxes of CPU 2 and CPU 4.

- Universal model QCPU (except the Q00UCPU, Q01UCPU, and Q02UCPU)
- Motion CPU (the Q172DCPU and Q173DCPU)
- Q12DCCPU-V
- Q24DHCCPU-V/-LS



Number of CPUs(*): 4

Host CPU number: No specification

Operation mode(*)
Operation mode at the time of CPU stop
☑ All CPUs stopped due to an error in CPU 1
☑ All CPUs stopped due to an error in CPU 2
☑ All CPUs stopped due to an error in CPU 3
☑ All CPUs stopped due to an error in CPU 4

Multiple CPU synchronous startup setting(*)
Target CPU
☑ CPU No.1
□ CPU No.2
☑ CPU No.3
□ CPU No.4

Online module change(*)
□ Enable online module change with another CPU
When the online module change with another CPU is enabled,
I/O status outside of the group cannot be read.

I/O sharing when using Multiple CPUs(*)
□ All CPUs can read all inputs
□ All CPUs can read all outputs

☑ Use multiple CPU high speed transmission

CPU	CPU specific send range (*)						
			User setting area			Auto refresh	
	Points(K)	I/O No.	Points	Start	End	Points	Setting
No. 1	3	3E0	3072	10000	13071	0	Setting
No. 2	3	3E1	3072	10000	13071	0	Setting
No. 3	3	3E2	3072	10000	13071	0	Setting
No. 4	3	3E3	3072	10000	13071	0	Setting

Total 12K points
□ Advanced settings(*)
The maximum total number of points is 12K.

(*) must be set identically for all CPUs when using multiple CPUs

C:\MELSEC\CCPU\Param\q12dccpu.cst | CPU type:Q12DCCPU-V | User name: target

Clear the check boxes when CPU modules other than the listed above are used.

Figure 13.37 Multiple CPU synchronized boot-up setting with CPU No.2 and No.4 not selected

CHAPTER 14 PARAMETERS ADDED FOR MULTIPLE CPU SYSTEMS

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	○

14.1 Parameter List

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	○

(1) Parameters used for multiple CPU systems

The following parameter settings have been added to C Controller setting utility to support not only single CPU systems but also multiple CPU systems.

- Number of CPUs
- Control CPU
- Refresh settings (Auto refresh)

The same parameters must be set to all the CPU modules used in the multiple CPU system except some settings.

(2) C Controller setting utility setting items for multiple CPU systems

Table 14.1 lists the setting items in C Controller setting utility and shows whether each item must be set and whether the same parameter must be set for the entire multiple CPU system or not.

Table 14.1 Setting items in I/O assignment setting (control CPU) and Multiple CPU settings

C Controller setting utility				Necessity of setting*1	Same setting*2	Reference section
I/O assignment settings	I/O assignment					Page 7-19, Section 7.6.2
		Type		—	○	
		Model name		—	—	
		Points		—	○	
		Start X/Y		—	○	
	Base setting					Page 7-7, Section 7.3
		Base model		—	—	
		Power model		—	—	
		Extension cable		—	—	
		Slots		—	○	
	Switch settings			—	—	Page 4-38, Section 4.11
	Detailed settings					Page 4-36, Section 4.9
		Error time output mode		—	—	
		H/W error time CPU operation mode		—	—	Page 4-37, Section 4.10
		I/O response time		—	—	Page 4-34, Section 4.8
		Control CPU		○	○	Page 14-7, Section 14.1.6
System settings	Points occupied by empty slot			—	○	Page 7-19, Section 7.6.2
Multiple CPU settings	Number of CPUs			○	○	Page 14-4, Section 14.1.1
	Host CPU number (Q12DCCPU-V only)			△	—	—
	Operation mode			△	○	Page 14-6, Section 14.1.2
	Multiple CPU synchronous startup setting (Q12DCCPU-V only)			△	○	Page 14-7, Section 14.1.7
	Online module change			△	○	Page 14-6, Section 14.1.3
	All CPUs can read all inputs			△	△	Page 14-6, Section 14.1.4
	All CPUs can read all outputs			△	△	
	Multiple CPU high speed transmission area setting (Q12DCCPU-V only)					Page 14-7, Section 14.1.8
		Use multiple CPU high speed transmission		○	○	
		CPU specific send range		○	○	
			Auto refresh	△	△	
		Advanced settings		△	○	
		system area*3		△	—	
	Communication area setting (refresh setting)					Page 14-6, Section 14.1.5
		Send range for each CPU		△	○	

* 1 Necessity of setting
○ : Required for multiple CPU systems. (Not operable without the setting)
△ : Can be set if necessary for the multiple CPU system. (Default values are applied if not specified.)
— : Set in the same way as single CPU system setting.

* 2 Same setting
○ : Must be the same for all the CPU modules in the multiple CPU system.
△ : Must be the same for all of C Controller modules and programmable controller CPUs in the multiple CPU system. (No setting available for Motion CPUs)
— : Can be set individually for each CPU module of the multiple CPU system.

* 3 The system area can be set when "Advanced settings" is selected.

If a multiple CPU setting parameter is changed, change the corresponding setting for each of all CPUs in the multiple CPU system accordingly.
After that, reset CPU No.1 or restart the multiple CPU system (power it on, off, and on again).

C Controller setting utility has a feature that allows you to utilize existing multiple CPU parameters in the C Controller setting utility or GX Developer. (☞ Page 15-14, Section 15.2.4)
For GX Works2, a project can be used by saving it in GX Developer format.

(3) Checking multiple CPU parameters

When the multiple CPU system is powered on or CPU No.1 is reset, whether the same parameters are set for all CPUs on the items marked ○ and △ in the Same setting column of Table 14.1 is checked (consistency check) as shown in Table 14.2.

(a) When settings of all CPUs are identical
The multiple CPU system starts.

(b) When settings of all CPUs are not identical
The multiple CPU system behaves as shown in Table 14.2.
Check the multiple CPU parameters, and make sure that they are identical for all CPUs.
To start the multiple CPU system, reset CPU No.1 or restart the multiple CPU system (power it on, off, and on again).
For information on resetting CPU No.1, refer to the following.
☞ Page 12-21, Section 12.6

Table 14.2 Consistency check

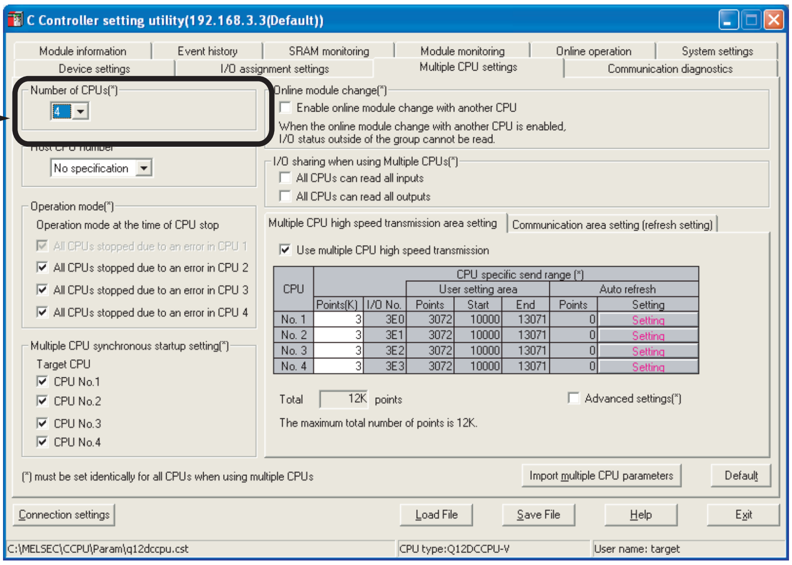
Item	CPU No.1	Other than CPU No.1
When multiple CPU system is powered on	No consistency check for multiple CPU parameters is performed.	• The multiple CPU parameters are compared with those of CPU No.1. • If any mismatch is found, a parameter error (error code: 3012, PARAMETER ERROR) is detected in the CPU.
When CPU No.1 is reset		

14.1.1 Setting the number of CPUs (required)

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	○

(1) Number of CPUs

In the <<Multiple CPU settings>> tab of C Controller setting utility, select the number of CPU modules to be used in the multiple CPU system.



Number of CPUs

Number of CPUs(*): 4

Host CPU number: No specification

Online module change(*)
□ Enable online module change with another CPU
When the online module change with another CPU is enabled, I/O status outside of the group cannot be read.

I/O sharing when using Multiple CPUs(*)
□ All CPUs can read all inputs
□ All CPUs can read all outputs

Operation mode(*)
Operation mode at the time of CPU stop
☑ All CPUs stopped due to an error in CPU 1
☑ All CPUs stopped due to an error in CPU 2
☑ All CPUs stopped due to an error in CPU 3
☑ All CPUs stopped due to an error in CPU 4

Multiple CPU synchronous startup setting(*)
Target CPU
☑ CPU No.1
☑ CPU No.2
☑ CPU No.3
☑ CPU No.4

Multiple CPU high speed transmission area setting | Communication area setting (refresh setting)

☑ Use multiple CPU high speed transmission

CPU	CPU specific send range (*)						
			User setting area			Auto refresh	
	Points(K)	I/O No.	Points	Start	End	Points	Setting
No. 1	3	3E0	3072	10000	13071	0	Setting
No. 2	3	3E1	3072	10000	13071	0	Setting
No. 3	3	3E2	3072	10000	13071	0	Setting
No. 4	3	3E3	3072	10000	13071	0	Setting

Total 12K points □ Advanced settings(*)

The maximum total number of points is 12K.

(*) must be set identically for all CPUs when using multiple CPUs

C:\MELSEC\CCPU\Param\q12dccpu.cst | CPU type:Q12DCCPU-V | User name: target

Figure 14.1 Setting the number of CPUs

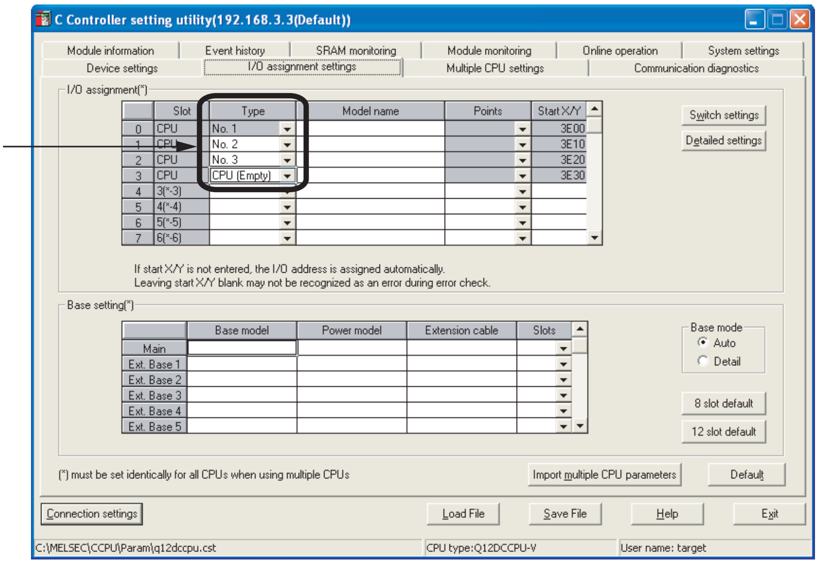
(2) Reserving an empty slot

To reserve an "Empty slot" for mounting a CPU module in the future, select "CPU (Empty)" in the <<I/O assignment settings>> tab of C Controller setting utility.

(☞Page 12-1, Section 12.1)

(Example) When reserving one out of four CPU slots for future use

Set 4 for the Number of CPUs in <<Multiple CPU settings>>, and select "CPU (Empty)" for Slot 3 (the 4th CPU module) in <<I/0 assignment settings>>.



"CPU (Empty)" setting

I/O assignment(*)

	Slot	Type	Model name	Points	Start X/Y
0	CPU	No. 1			3E00
1	CPU	No. 2			3E10
2	CPU	No. 3			3E20
3	CPU	CPU (Empty)			3E30
4	3(*-3)				
5	4(*-4)				
6	5(*-5)				
7	6(*-6)				

If start X/Y is not entered, the I/O address is assigned automatically.
Leaving start X/Y blank may not be recognized as an error during error check.

Base setting(*)

	Base model	Power model	Extension cable	Slots
Main				
Ext. Base 1				
Ext. Base 2				
Ext. Base 3				
Ext. Base 4				
Ext. Base 5				

Base mode
● Auto
○ Detail

(*) must be set identically for all CPUs when using multiple CPUs

C:\MELSEC\CCPU\Param\q12dccpu.cst | CPU type:Q12DCCPU-V | User name: target

Figure 14.2 Reserving an empty slot

☒POINT

The following (1) or (2) may cause an error in the actually mounted CPU module No.1.

(1) CPU modules exceeding the specified number of CPUs are mounted.
(a) When CPU No.1 is a C Controller module
An intelligent function module assignment error (error code: 2126, SP.UNIT LAY ERROR.) or a parameter error (error code: 3010, PARAMETER ERROR) occurs.
(b) When CPU No.1 is a Basic model QCPU or Universal model QCPU
A CPU LAY ERROR (error code: 7030) occurs.
(c) When CPU No.1 is a High Performance model QCPU or Process CPU
A parameter error (error code: 3010, PARAMETER ERROR) occurs.

(2) Any of the set CPU modules is not mounted in the relevant slot
(a) When CPU No.1 is a C Controller module
An intelligent function module assignment error (error code: 2126, SP.UNIT LAY ERROR.) or a parameter error (error code: 3010, PARAMETER ERROR) occurs.
(b) When CPU No.1 is a Basic model QCPU or Universal model QCPU
A CPU LAY ERROR (error code: 7031) occurs.
(c) When CPU No.1 is a High Performance model QCPU or Process CPU
A parameter error (error code: 3010, PARAMETER ERROR) occurs.

An error that has occurred in a C Controller module can be confirmed on any of the following tabs of C Controller setting utility.

- <<Module information>> tab
- <<Event history>> tab
- System information from <<Module monitoring>> tab

For C Controller setting utility, refer to the following.

☞ C Controller Module User's Manual (Utility Operation, Programming)

14.1.2 Operation mode setting (optional)

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	○

When a stop error occurs in a CPU module other than CPU No.1, this setting allows operations of the other CPUs to continue.
The operation mode of CPU No.1 cannot be changed. All CPUs will stop if a stop error occurs in CPU No.1.
(☞ Page 12-22, Section 12.7)

14.1.3 Online module change (optional)

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	○

Enable this setting when a module controlled by a Process CPU will be replaced online.
☞QCPU User's Manual (Hardware Design, Maintenance and Inspection)

14.1.4 I/O sharing when using Multiple CPUs (optional)

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	○

This setting allows each CPU to load I/O data (X and Y) of the I/O modules and intelligent function modules that are controlled by the other CPUs. (☞ Page 12-13, Section 12.4.2)

14.1.5 Communication area setting (refresh setting) (optional)

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	○

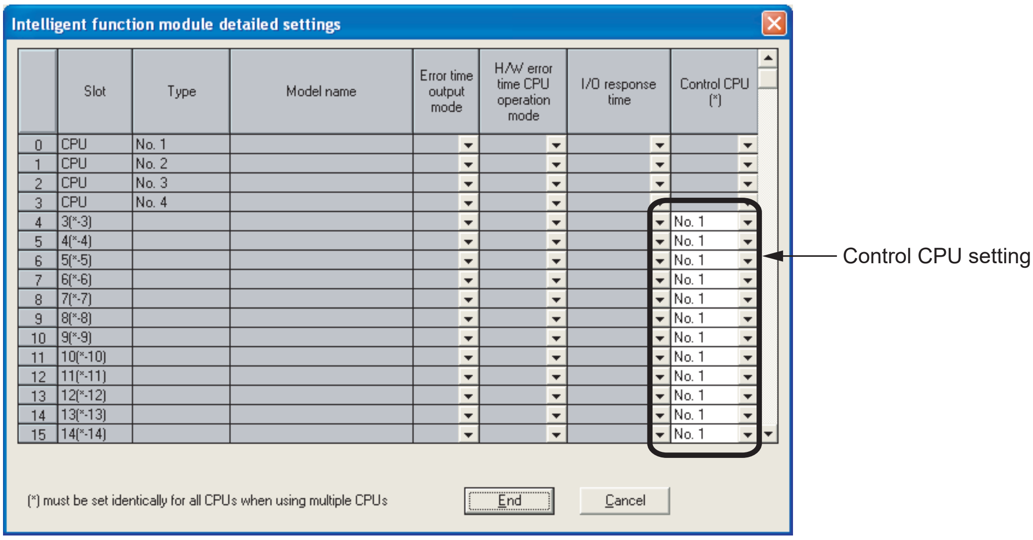
The settings can be configured for the data communications using CPU shared memories (using auto refresh) in a multiple CPU system. (☞ Page 13-17, Section 13.3.2)

14.1.6 Control CPU settings (required)

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	○

Specify a control CPU for each of I/O modules and intelligent function modules that are mounted on a base unit in the multiple CPU system.

CPU No.1 is set for all by default.



Intelligent function module detailed settings

	Slot	Type	Model name	Error time output mode	H/W error time CPU operation mode	I/O response time	Control CPU (*)
0	CPU	No. 1					
1	CPU	No. 2					
2	CPU	No. 3					
3	CPU	No. 4					
4	3(*-3)						No. 1
5	4(*-4)						No. 1
6	5(*-5)						No. 1
7	6(*-6)						No. 1
8	7(*-7)						No. 1
9	8(*-8)						No. 1
10	9(*-9)						No. 1
11	10(*-10)						No. 1
12	11(*-11)						No. 1
13	12(*-12)						No. 1
14	13(*-13)						No. 1
15	14(*-14)						No. 1

(*) must be set identically for all CPUs when using multiple CPUs

Figure 14.3 Control CPU setting

14.1.7 Multiple CPU synchronous startup setting (optional)

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	×	×

In the multiple CPU system, synchronized startup is available for the selected CPU modules. (☞ Page 13-48, Section 13.9)

14.1.8 Multiple CPU high speed transmission area setting (required)

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	×	×

The settings can be configured for the data communications using the multiple CPU transmission areas in CPU shared memories (using auto refresh). (☞ Page 13-24, Section 13.3.3)

CHAPTER 15 STARTING A MULTIPLE CPU SYSTEM

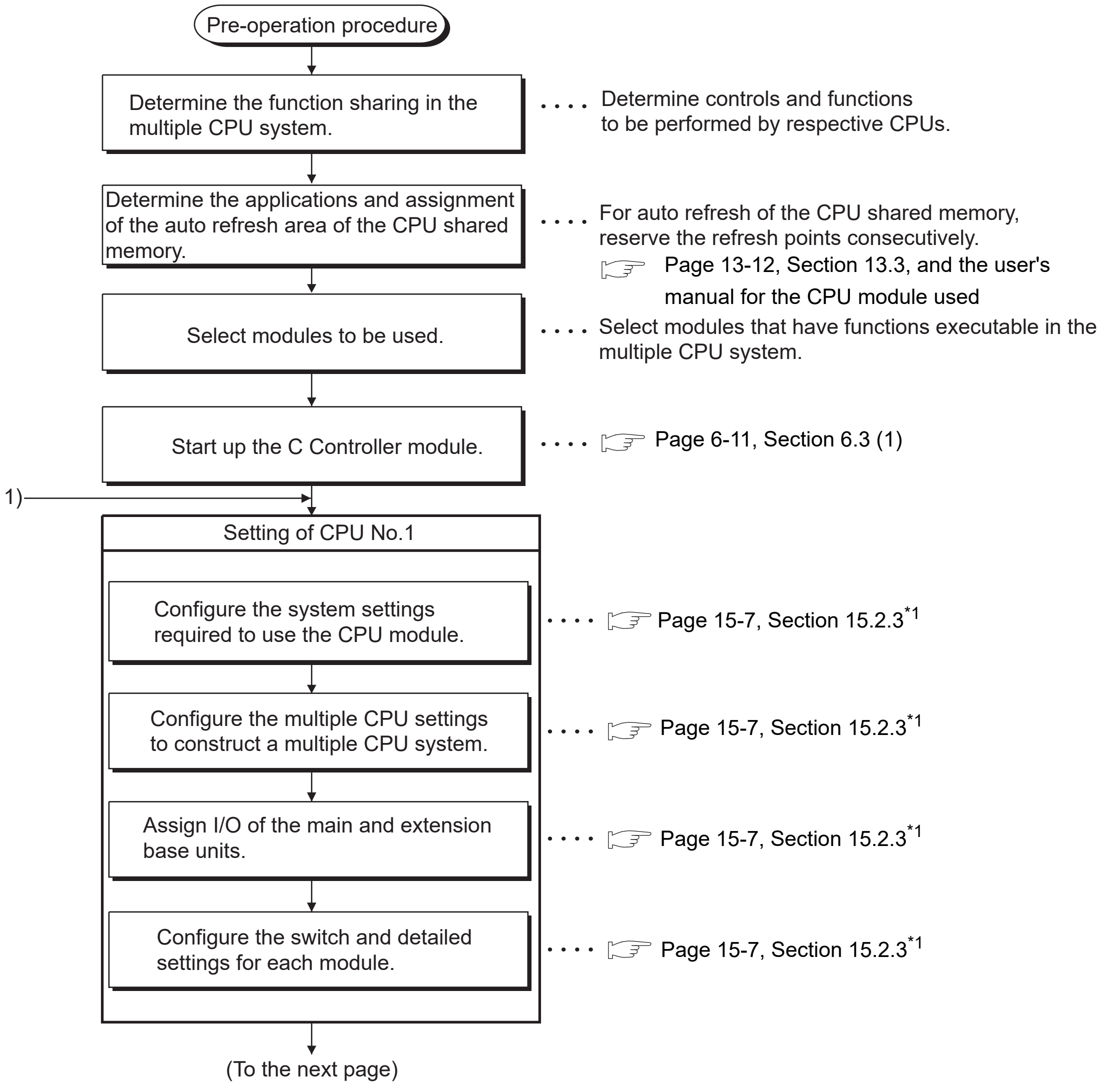
Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	○

This chapter explains the standard startup procedure of a multiple CPU system.

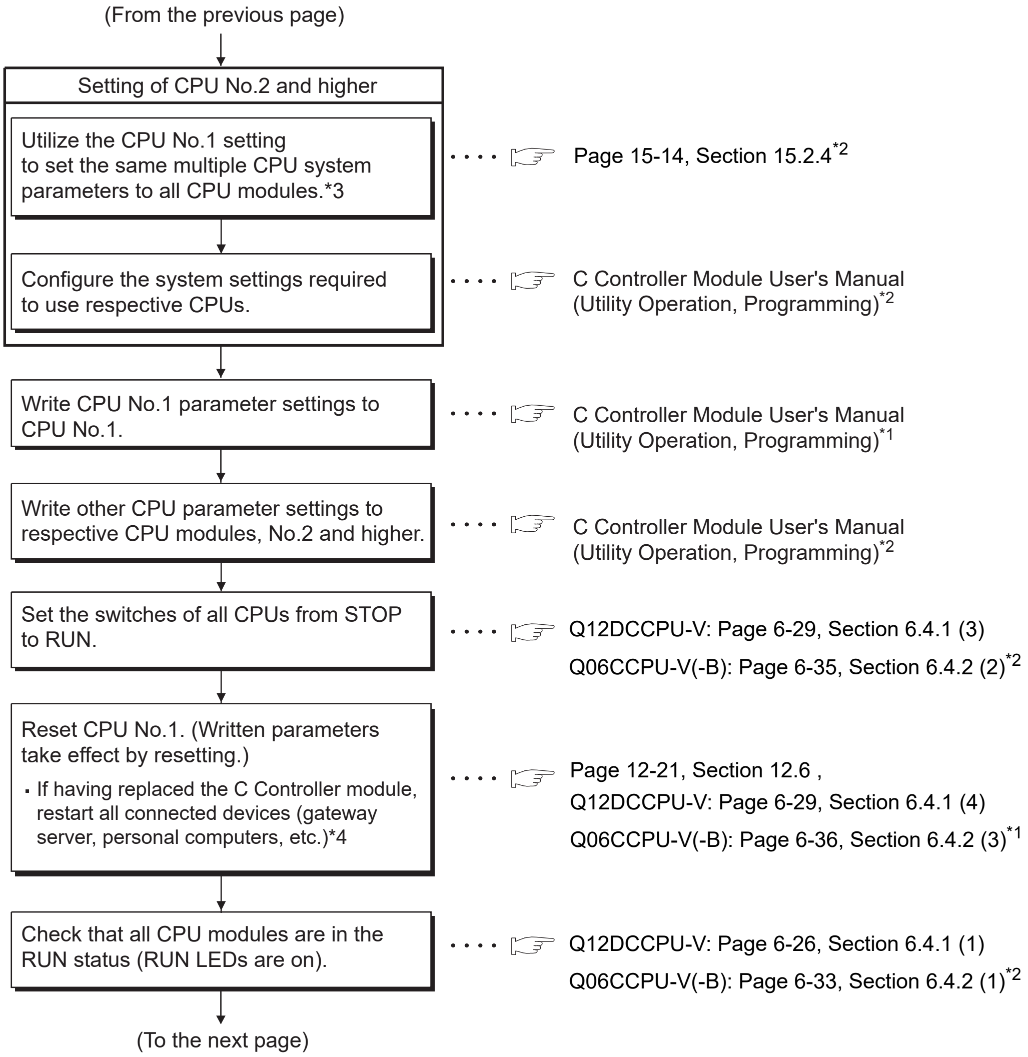
15.1 Flowchart for Starting a Multiple CPU System

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	○

For the parameter setting and programming of programmable controller CPUs and Motion CPUs, refer to the manuals for respective modules.


Pre-operation procedure
Determine the function sharing in the multiple CPU system.
Determine controls and functions to be performed by respective CPUs.
Determine the applications and assignment of the auto refresh area of the CPU shared memory.
For auto refresh of the CPU shared memory, reserve the refresh points consecutively.
Page 13-12, Section 13.3, and the user's manual for the CPU module used
Select modules to be used.
Select modules that have functions executable in the multiple CPU system.
Start up the C Controller module.
Page 6-11, Section 6.3 (1)
1)
Setting of CPU No.1
Configure the system settings required to use the CPU module.
Page 15-7, Section 15.2.3*1
Configure the multiple CPU settings to construct a multiple CPU system.
Page 15-7, Section 15.2.3*1
Assign I/O of the main and extension base units.
Page 15-7, Section 15.2.3*1
Configure the switch and detailed settings for each module.
Page 15-7, Section 15.2.3*1
(To the next page)

* 1 Reference section and manual are for the C Controller module.
For the programmable controller CPU, refer to the following.
QCPU User's Manual (Multiple CPU System)



(From the previous page)

Setting of CPU No.2 and higher

Utilize the CPU No.1 setting to set the same multiple CPU system parameters to all CPU modules.*3 ···· ☞ Page 15-14, Section 15.2.4[*2]

Configure the system settings required to use respective CPUs. ···· ☞ C Controller Module User's Manual (Utility Operation, Programming)[*2]

Write CPU No.1 parameter settings to CPU No.1. ···· ☞ C Controller Module User's Manual (Utility Operation, Programming)[*1]

Write other CPU parameter settings to respective CPU modules, No.2 and higher. ···· ☞ C Controller Module User's Manual (Utility Operation, Programming)[*2]

Set the switches of all CPUs from STOP to RUN. ···· ☞ Q12DCCPU-V: Page 6-29, Section 6.4.1 (3)
Q06CCPU-V(-B): Page 6-35, Section 6.4.2 (2)[*2]

Reset CPU No.1. (Written parameters take effect by resetting.)
- If having replaced the C Controller module, restart all connected devices (gateway server, personal computers, etc.)*4

···· ☞ Page 12-21, Section 12.6 ,
Q12DCCPU-V: Page 6-29, Section 6.4.1 (4)
Q06CCPU-V(-B): Page 6-36, Section 6.4.2 (3)[*1]

Check that all CPU modules are in the RUN status (RUN LEDs are on). ···· ☞ Q12DCCPU-V: Page 6-26, Section 6.4.1 (1)
Q06CCPU-V(-B): Page 6-33, Section 6.4.2 (1)[*2]

(To the next page)

* 1 Reference section and manual are for the C Controller module.
For a programmable controller CPU, refer to the following.
☞ QCPU User's Manual (Multiple CPU System)

* 2 Reference section and manual are for the C Controller module.
For a programmable controller CPU or Motion CPU, refer to the following.
☞ Manual for the CPU module used

* 3 To import parameters for a multiple CPU system, use either of the following buttons.
<When CPU No.1 is a C Controller module>
Click the Load File button of C Controller setting utility.
<When CPU No.1 is a programmable controller CPU>
Click the Import multiple CPU parameters button of C Controller setting utility.
(☞ Page 15-14, Section 15.2.4)

* 4 When having replaced the C Controller module, restart all connected devices such as gateway server and personal computer.
(If the connected device retains an Ethernet address of its communication target, the Ethernet address is changed and continued communication may be disabled after the replacement.)
Likewise, restart the C Controller module when a connected device has been replaced.

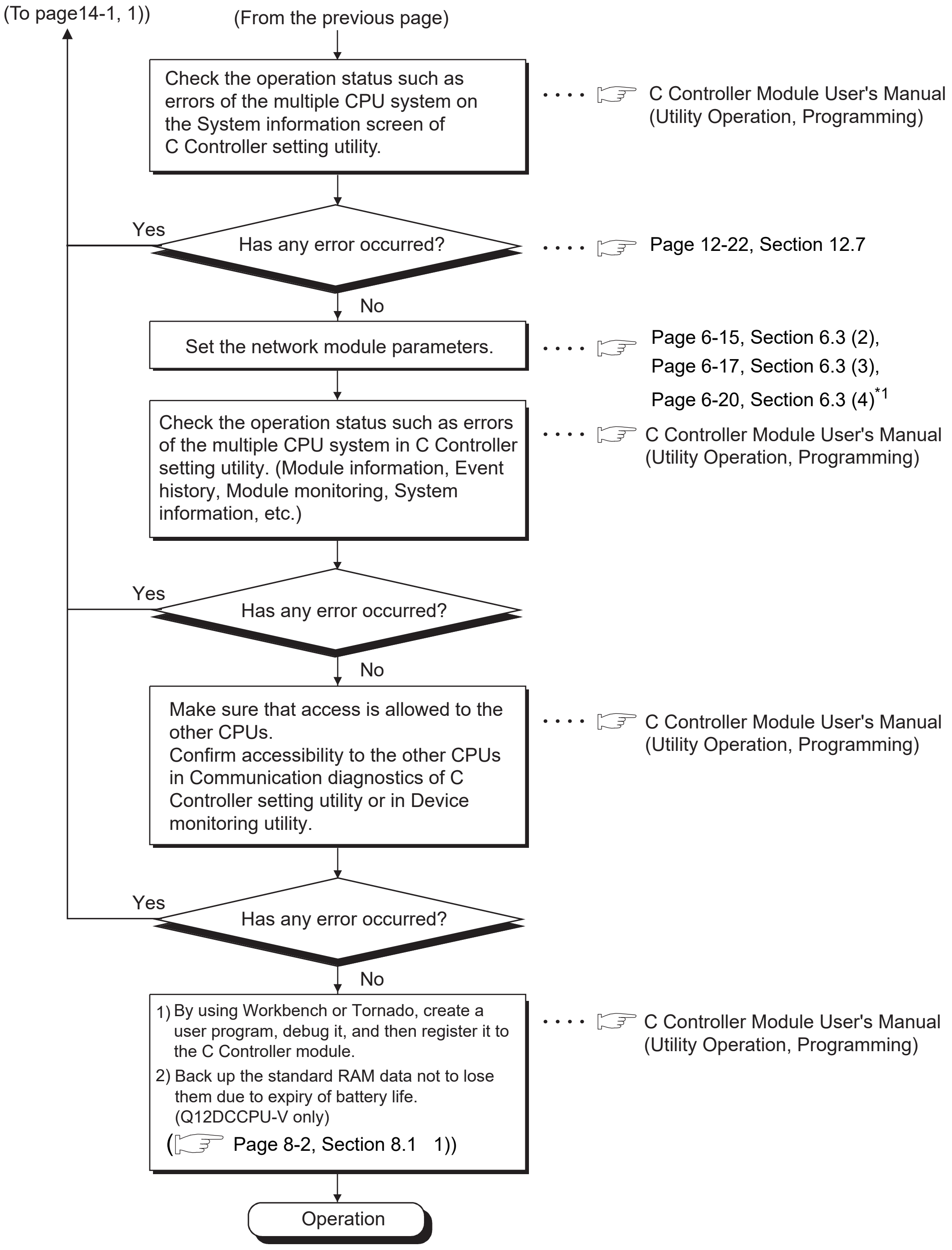

(To page14-1, 1))
(From the previous page)
Check the operation status such as errors of the multiple CPU system on the System information screen of C Controller setting utility.
C Controller Module User's Manual (Utility Operation, Programming)
Yes
Has any error occurred?
Page 12-22, Section 12.7
No
Set the network module parameters.
Page 6-15, Section 6.3 (2), Page 6-17, Section 6.3 (3), Page 6-20, Section 6.3 (4)*1
Check the operation status such as errors of the multiple CPU system in C Controller setting utility. (Module information, Event history, Module monitoring, System information, etc.)
C Controller Module User's Manual (Utility Operation, Programming)
Yes
Has any error occurred?
No
Make sure that access is allowed to the other CPUs. Confirm accessibility to the other CPUs in Communication diagnostics of C Controller setting utility or in Device monitoring utility.
C Controller Module User's Manual (Utility Operation, Programming)
Yes
Has any error occurred?
No
1) By using Workbench or Tornado, create a user program, debug it, and then register it to the C Controller module.
2) Back up the standard RAM data not to lose them due to expiry of battery life. (Q12DCCPU-V only)
(Page 8-2, Section 8.1 1))
C Controller Module User's Manual (Utility Operation, Programming)
Operation

Figure 15.1 Procedure for starting up a multiple CPU system

* 1 Reference section and manual are for the C Controller module.
For a programmable controller CPU, refer to the following.
QCPU User's Manual (Multiple CPU System)

15.2 Setting Parameters Added for Multiple CPU Systems

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	○

This section provides the procedure for setting parameters for the multiple CPU system in C Controller setting utility.

15.2.1 System configuration

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	○

The procedure for setting multiple CPU system parameters is explained by using the system shown in Figure 15.2 as an example.

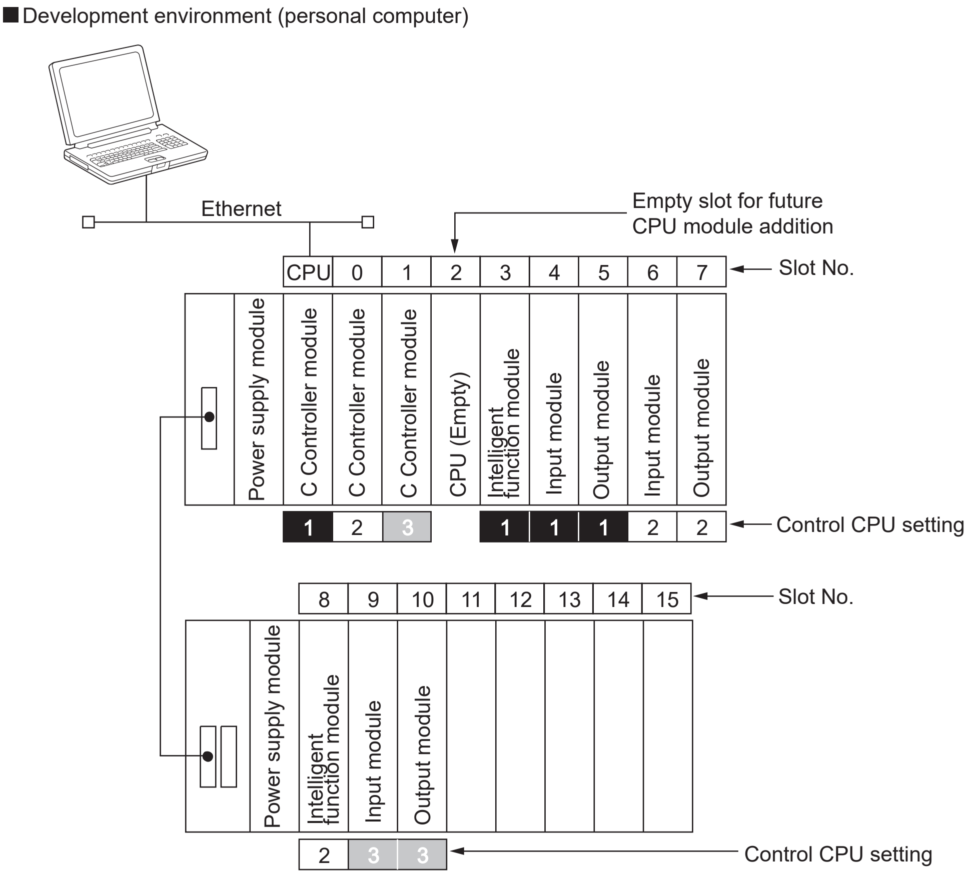


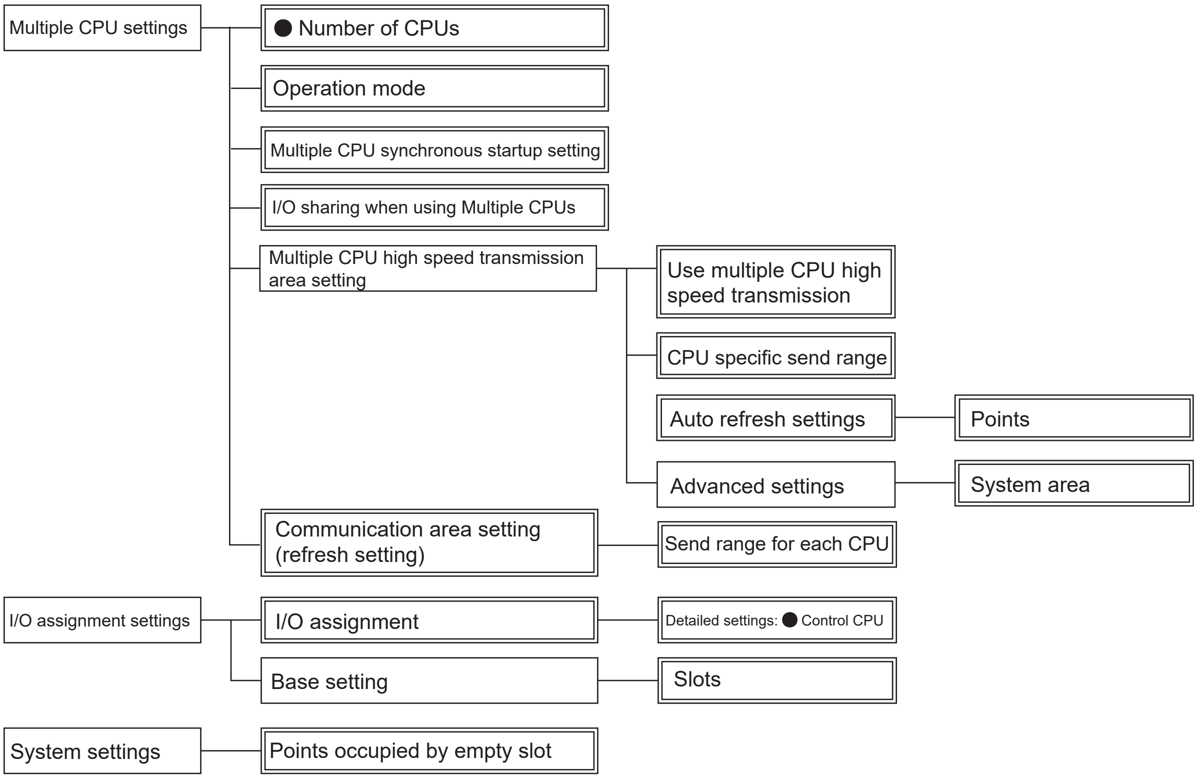
Figure 15.2 Example of multiple CPU system configuration

15.2.2 Parameters required for the multiple CPU system

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	○

The following parameters are required for the multiple CPU system.
The same parameters must be set to the items marked as "Same setting to all CPU modules" for all CPU modules in the multiple CPU system unless otherwise noted.
(☞ Page 14-1, Section 14.1)

(1) For the Q12DCCPU-V


Multiple CPU settings
● Number of CPUs
Operation mode
Multiple CPU synchronous startup setting
I/O sharing when using Multiple CPUs
Multiple CPU high speed transmission area setting
Use multiple CPU high speed transmission
CPU specific send range
Auto refresh settings
Points
Advanced settings
System area
Communication area setting (refresh setting)
Send range for each CPU
I/O assignment settings
I/O assignment
Detailed settings: ● Control CPU
Base setting
Slots
System settings
Points occupied by empty slot

● : Setting required
▭ : Same setting to all CPU modules

Figure 15.3 List of parameters required for the multiple CPU system

(2) For the Q06CCPU-V(-B)

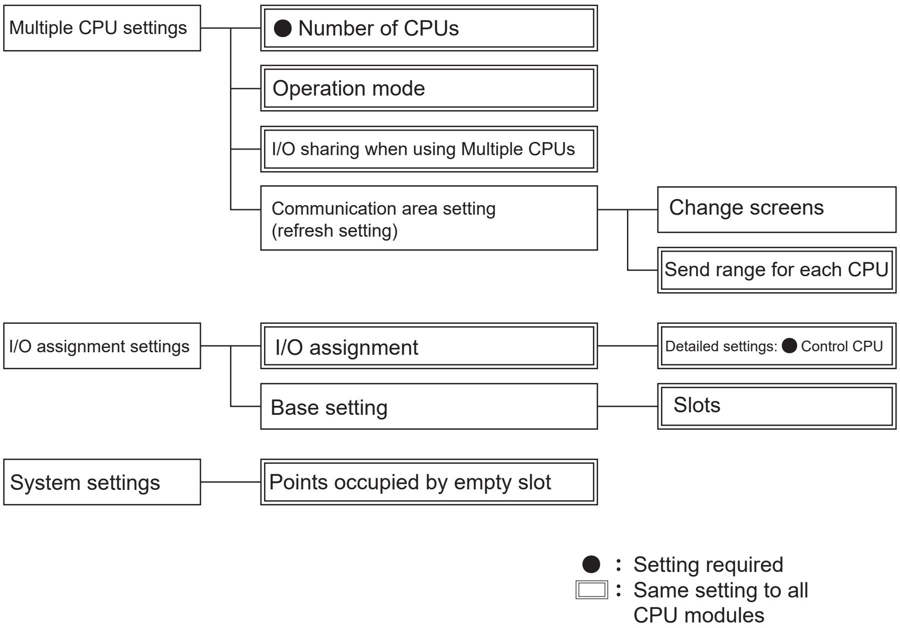


Figure 15.4 List of parameters required for the multiple CPU system

15.2.3 When creating a new system

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	○

(1) For the Q12DCCPU-V

Start

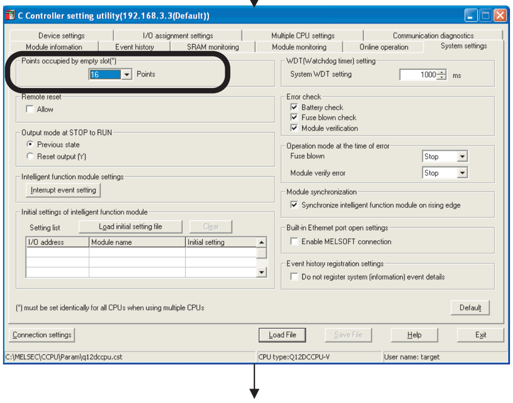
↓

Start C Controller setting utility

↓

Connect to the C Controller module.

↓



Points occupied by empty slot(*): 16 Points

WDT(Watchdog timer) setting: System WDT setting 1000 ms

Remote reset: □ Allow

Error check: ☑ Battery check ☑ Fuse blown check ☑ Module verification

Output mode at STOP to RUN: ● Previous state ○ Reset output (Y)

Operation mode at the time of error: Fuse blown Stop; Module verify error Stop

Intelligent function module settings

Module synchronization: ☑ Synchronize intelligent function module on rising edge

Initial settings of intelligent function module: Setting list

I/O address	Module name	Initial setting

Built-in Ethernet port open settings: □ Enable MELSOFT connection

Event history registration settings: □ Do not register system (information) event details

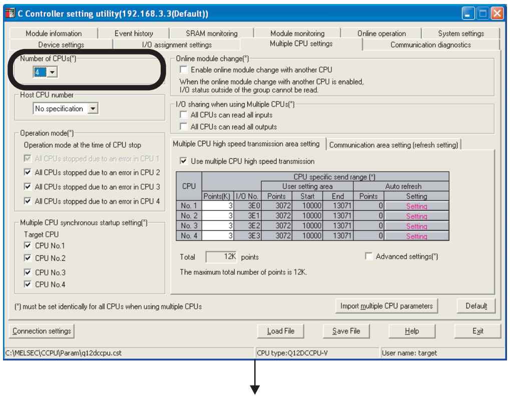
(*) must be set identically for all CPUs when using multiple CPUs

C:\MELSEC\CCPU\Param\q12dccpu.cst | CPU type:Q12DCCPU-V | User name: target

"Points occupied by empty slot" in the <<System settings>> tab (optional)

- Set the points to be occupied by one empty slot.
  Default: 16 points

↓



Number of CPUs(*): 4

Host CPU number: No specification

Online module change(*): □ Enable online module change with another CPU

When the online module change with another CPU is enabled, I/O status outside of the group cannot be read.

I/O sharing when using Multiple CPUs(*): □ All CPUs can read all inputs □ All CPUs can read all outputs

Operation mode(*)
Operation mode at the time of CPU stop
☑ All CPUs stopped due to an error in CPU 1
☑ All CPUs stopped due to an error in CPU 2
☑ All CPUs stopped due to an error in CPU 3
☑ All CPUs stopped due to an error in CPU 4

Multiple CPU synchronous startup setting(*)
Target CPU
☑ CPU No.1
☑ CPU No.2
☑ CPU No.3
☑ CPU No.4

☑ Use multiple CPU high speed transmission

CPU	CPU specific send range (*)						
			User setting area			Auto refresh	
	Points(K)	I/O No.	Points	Start	End	Points	Setting
No. 1	3	3E0	3072	10000	13071	0	Setting
No. 2	3	3E1	3072	10000	13071	0	Setting
No. 3	3	3E2	3072	10000	13071	0	Setting
No. 4	3	3E3	3072	10000	13071	0	Setting

Total 12K points □ Advanced settings(*)

The maximum total number of points is 12K.

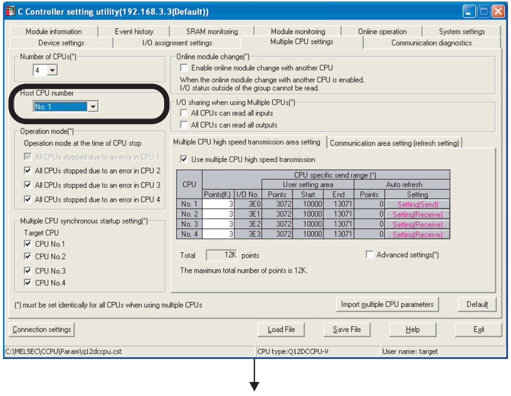
(*) must be set identically for all CPUs when using multiple CPUs

C:\MELSEC\CCPU\Param\q12dccpu.cst | CPU type:Q12DCCPU-V | User name: target

"Number of CPUs" in the <<Multiple CPU settings>> tab (setting required)

- Set the number of CPU modules mounted on the main base unit in the multiple CPU system.

↓



Number of CPUs(*): 4

Host CPU number: No. 1

Online module change(*): □ Enable online module change with another CPU

When the online module change with another CPU is enabled, I/O status outside of the group cannot be read.

I/O sharing when using Multiple CPUs(*): □ All CPUs can read all inputs □ All CPUs can read all outputs

Operation mode(*)
Operation mode at the time of CPU stop
☑ All CPUs stopped due to an error in CPU 1
☑ All CPUs stopped due to an error in CPU 2
☑ All CPUs stopped due to an error in CPU 3
☑ All CPUs stopped due to an error in CPU 4

Multiple CPU synchronous startup setting(*)
Target CPU
☑ CPU No.1
☑ CPU No.2
☑ CPU No.3
☑ CPU No.4

☑ Use multiple CPU high speed transmission

CPU	CPU specific send range (*)						
			User setting area			Auto refresh	
	Points(K)	I/O No.	Points	Start	End	Points	Setting
No. 1	3	3E0	3072	10000	13071	0	Setting(Send)
No. 2	3	3E1	3072	10000	13071	0	Setting(Receive)
No. 3	3	3E2	3072	10000	13071	0	Setting(Receive)
No. 4	3	3E3	3072	10000	13071	0	Setting(Receive)

Total 12K points □ Advanced settings(*)

The maximum total number of points is 12K.

(*) must be set identically for all CPUs when using multiple CPUs

C:\MELSEC\CCPU\Param\q12dccpu.cst | CPU type:Q12DCCPU-V | User name: target

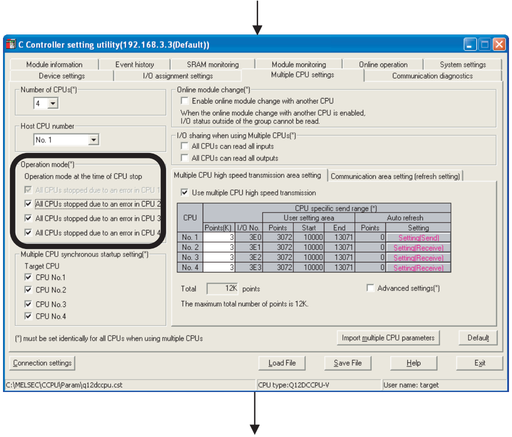
"Host CPU number" in the <<Multiple CPU settings>> tab (optional)

- Set this when checking the host CPU number.
  Default: Not specified

↓

(To the next page)

(From the previous page)



Number of CPUs(*): 4

Host CPU number: No. 1

Online module change(*)
□ Enable online module change with another CPU
When the online module change with another CPU is enabled, I/O status outside of the group cannot be read.

I/O sharing when using Multiple CPUs(*)
□ All CPUs can read all inputs
□ All CPUs can read all outputs

Operation mode(*)
Operation mode at the time of CPU stop
☑ All CPUs stopped due to an error in CPU 1
☑ All CPUs stopped due to an error in CPU 2
☑ All CPUs stopped due to an error in CPU 3
☑ All CPUs stopped due to an error in CPU 4

Multiple CPU synchronous startup setting(*)
Target CPU
☑ CPU No.1
☑ CPU No.2
☑ CPU No.3
☑ CPU No.4

Multiple CPU high speed transmission area setting | Communication area setting (refresh setting)

☑ Use multiple CPU high speed transmission

CPU	CPU specific send range (*)						
			User setting area			Auto refresh	
	Points(K)	I/O No.	Points	Start	End	Points	Setting
No. 1	3	3E0	3072	10000	13071	0	Setting(Send)
No. 2	3	3E1	3072	10000	13071	0	Setting(Receive)
No. 3	3	3E2	3072	10000	13071	0	Setting(Receive)
No. 4	3	3E3	3072	10000	13071	0	Setting(Receive)

Total 12K points

□ Advanced settings(*)

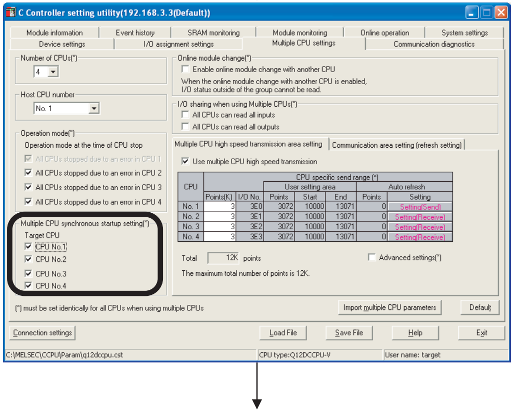
The maximum total number of points is 12K.

(*) must be set identically for all CPUs when using multiple CPUs

C:\MELSEC\CCPU\Param\q12dccpu.cst | CPU type:Q12DCCPU-V | User name: target

"Operation mode" in the <<Multiple CPU settings>> tab (optional)

- Select whether to stop or continue the operations of all CPUs in the event of a stop error.
  Default: All CPUs stop by a stop error in any of CPU No. 2, 3 and 4 (Checked)
- For example, when "All CPUs stopped due to an error in CPU2" is unchecked, CPUs other than CPU No.2 will continue their operations even if a stop error occurs in CPU No.2.
- The operating mode of CPU No.1 cannot be changed.



Number of CPUs(*): 4

Host CPU number: No. 1

Online module change(*)
□ Enable online module change with another CPU
When the online module change with another CPU is enabled, I/O status outside of the group cannot be read.

I/O sharing when using Multiple CPUs(*)
□ All CPUs can read all inputs
□ All CPUs can read all outputs

Operation mode(*)
Operation mode at the time of CPU stop
☑ All CPUs stopped due to an error in CPU 1
☑ All CPUs stopped due to an error in CPU 2
☑ All CPUs stopped due to an error in CPU 3
☑ All CPUs stopped due to an error in CPU 4

Multiple CPU synchronous startup setting(*)
Target CPU
☑ CPU No.1
☑ CPU No.2
☑ CPU No.3
☑ CPU No.4

Multiple CPU high speed transmission area setting | Communication area setting (refresh setting)

☑ Use multiple CPU high speed transmission

CPU	CPU specific send range (*)						
			User setting area			Auto refresh	
	Points(K)	I/O No.	Points	Start	End	Points	Setting
No. 1	3	3E0	3072	10000	13071	0	Setting(Send)
No. 2	3	3E1	3072	10000	13071	0	Setting(Receive)
No. 3	3	3E2	3072	10000	13071	0	Setting(Receive)
No. 4	3	3E3	3072	10000	13071	0	Setting(Receive)

Total 12K points

□ Advanced settings(*)

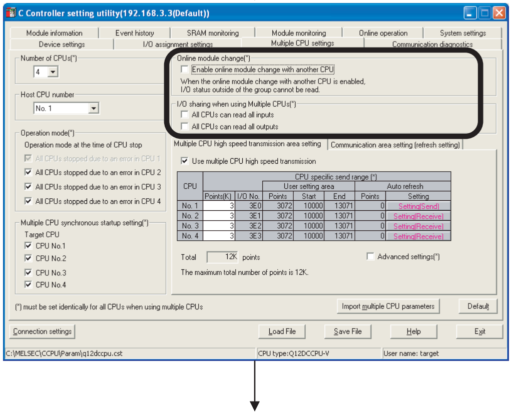
The maximum total number of points is 12K.

(*) must be set identically for all CPUs when using multiple CPUs

C:\MELSEC\CCPU\Param\q12dccpu.cst | CPU type:Q12DCCPU-V | User name: target

"Multiple CPU synchronous startup setting" in the <<Multiple CPU settings>> tab

- Select the modules to be started synchronously in the multiple CPU system.
  Default: All CPUs start synchronously. (Checked)



Number of CPUs(*): 4

Host CPU number: No. 1

Online module change(*)
□ Enable online module change with another CPU
When the online module change with another CPU is enabled, I/O status outside of the group cannot be read.

I/O sharing when using Multiple CPUs(*)
□ All CPUs can read all inputs
□ All CPUs can read all outputs

Operation mode(*)
Operation mode at the time of CPU stop
☑ All CPUs stopped due to an error in CPU 1
☑ All CPUs stopped due to an error in CPU 2
☑ All CPUs stopped due to an error in CPU 3
☑ All CPUs stopped due to an error in CPU 4

Multiple CPU synchronous startup setting(*)
Target CPU
☑ CPU No.1
☑ CPU No.2
☑ CPU No.3
☑ CPU No.4

Multiple CPU high speed transmission area setting | Communication area setting (refresh setting)

☑ Use multiple CPU high speed transmission

CPU	CPU specific send range (*)						
			User setting area			Auto refresh	
	Points(K)	I/O No.	Points	Start	End	Points	Setting
No. 1	3	3E0	3072	10000	13071	0	Setting(Send)
No. 2	3	3E1	3072	10000	13071	0	Setting(Receive)
No. 3	3	3E2	3072	10000	13071	0	Setting(Receive)
No. 4	3	3E3	3072	10000	13071	0	Setting(Receive)

Total 12K points

□ Advanced settings(*)

The maximum total number of points is 12K.

(*) must be set identically for all CPUs when using multiple CPUs

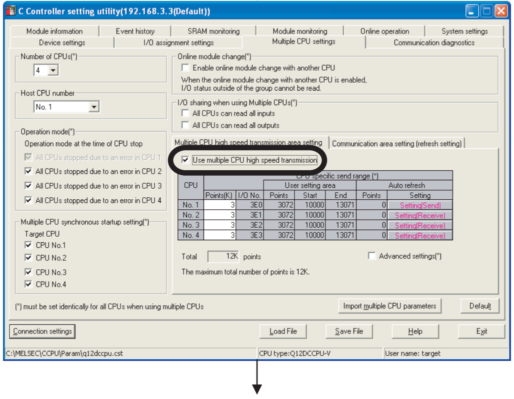
C:\MELSEC\CCPU\Param\q12dccpu.cst | CPU type:Q12DCCPU-V | User name: target

"I/O sharing when using Multiple CPUs" in the <<Multiple CPU settings>> tab (optional)

- Set whether to read input/output status data of non-controlled modules or not.
  Default: Not read. (Not checked)

"Online module change" in the <<Multiple CPU settings>> tab (optional)

- Determine whether to enable the online module change setting.
  Default: Online module change is disabled.
  (Not checked)
- Do not check the check box when setting "I/O sharing when using Multiple CPUs".
- Check the check box when performing online module change.



Number of CPUs(*): 4

Host CPU number: No. 1

Online module change(*)
□ Enable online module change with another CPU
When the online module change with another CPU is enabled, I/O status outside of the group cannot be read.

I/O sharing when using Multiple CPUs(*)
□ All CPUs can read all inputs
□ All CPUs can read all outputs

Operation mode(*)
Operation mode at the time of CPU stop
☑ All CPUs stopped due to an error in CPU 1
☑ All CPUs stopped due to an error in CPU 2
☑ All CPUs stopped due to an error in CPU 3
☑ All CPUs stopped due to an error in CPU 4

Multiple CPU synchronous startup setting(*)
Target CPU
☑ CPU No.1
☑ CPU No.2
☑ CPU No.3
☑ CPU No.4

Multiple CPU high speed transmission area setting | Communication area setting (refresh setting)

☑ Use multiple CPU high speed transmission

CPU	CPU specific send range (*)						
			User setting area			Auto refresh	
	Points(K)	I/O No.	Points	Start	End	Points	Setting
No. 1	3	3E0	3072	10000	13071	0	Setting(Send)
No. 2	3	3E1	3072	10000	13071	0	Setting(Receive)
No. 3	3	3E2	3072	10000	13071	0	Setting(Receive)
No. 4	3	3E3	3072	10000	13071	0	Setting(Receive)

Total 12K points

□ Advanced settings(*)

The maximum total number of points is 12K.

(*) must be set identically for all CPUs when using multiple CPUs

C:\MELSEC\CCPU\Param\q12dccpu.cst | CPU type:Q12DCCPU-V | User name: target

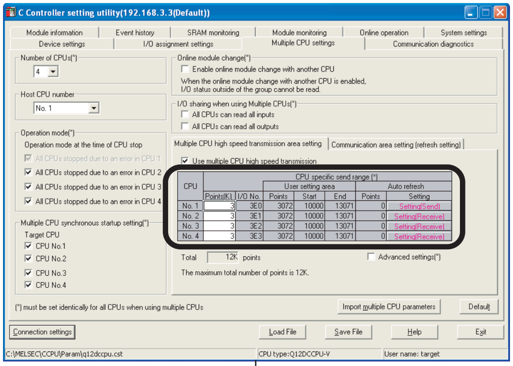
"Use multiple CPU high speed transmission" in the <<Multiple CPU settings>> tab (setting required)

- Select whether to use multiple CPU high speed transmission.
  Default: Use multiple CPU high speed transmission
  (Checked)

(To the next page)

9 DEVICE DESCRIPTION
10 MULTIPLE CPU SYSTEM OVERVIEW
11 MULTIPLE CPU SYSTEM CONFIGURATION
12 CONCEPT OF MULTIPLE CPU SYSTEM
13 COMMUNICATIONS BETWEEN CPU MODULES
14 PARAMETERS ADDED FOR MULTIPLE CPU SYSTEMS
15 STARTING A MULTIPLE CPU SYSTEM
16 TROUBLESHOOTING

(From the previous page)



C Controller setting utility(192.168.3.3(Default))

Module information | Event history | SRAM monitoring | Module monitoring | Online operation | System settings
Device settings | I/O assignment settings | Multiple CPU settings | Communication diagnostics

Number of CPUs(*): 4

Host CPU number: No. 1

Operation mode(*)
Operation mode at the time of CPU stop
☑ All CPUs stopped due to an error in CPU 1
☑ All CPUs stopped due to an error in CPU 2
☑ All CPUs stopped due to an error in CPU 3
☑ All CPUs stopped due to an error in CPU 4

Multiple CPU synchronous startup setting(*)
Target CPU
☑ CPU No.1
☑ CPU No.2
☑ CPU No.3
☑ CPU No.4

Online module change(*)
☐ Enable online module change with another CPU
When the online module change with another CPU is enabled,
I/O status outside of the group cannot be read.

I/O sharing when using Multiple CPUs(*)
☐ All CPUs can read all inputs
☐ All CPUs can read all outputs

Multiple CPU high speed transmission area setting | Communication area setting (refresh setting)

☑ Use multiple CPU high speed transmission

CPU	CPU specific send range (*)						
			User setting area			Auto refresh	
	Points(K)	I/O No.	Points	Start	End	Points	Setting
No. 1	3	3E0	3072	10000	13071	0	Setting(Send)
No. 2	3	3E1	3072	10000	13071	0	Setting(Receive)
No. 3	3	3E2	3072	10000	13071	0	Setting(Receive)
No. 4	3	3E3	3072	10000	13071	0	Setting(Receive)

Total 12K points ☐ Advanced settings(*)

The maximum total number of points is 12K.

(*) must be set identically for all CPUs when using multiple CPUs

Import multiple CPU parameters | Default

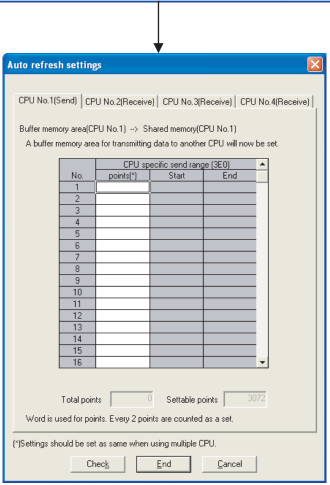
Connection settings | Load File | Save File | Help | Exit

C:\MELSEC\CCPU\Param\q12dccpu.cst | CPU type:Q12DCCPU-V | User name: target

"CPU specific send range" in the <<Multiple CPU settings>> tab (setting required)

- Set the points of data to be sent and received between CPUs.
- The setting range is shown below.

Number of CPUs	Setting range
2	0 to 14K points
3	0 to 13K points
4	0 to 12K points



Auto refresh settings

CPU No.1(Send) | CPU No.2(Receive) | CPU No.3(Receive) | CPU No.4(Receive)

Buffer memory area(CPU No.1) --> Shared memory(CPU No.1)
A buffer memory area for transmitting data to another CPU will now be set.

No.	CPU specific send range (3E0)		
	points(*)	Start	End
1			
2			
3			
4			
5			
6			
7			
8			
9			
10			
11			
12			
13			
14			
15			
16			

Total points 0 Settable points 3072

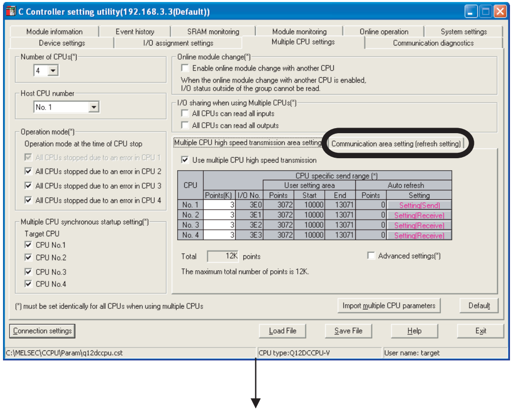
Word is used for points. Every 2 points are counted as a set.

(*)Settings should be set as same when using multiple CPU.

Check | End | Cancel

Auto refresh settings (optional)

- Set the points of data to be sent and received by auto refresh between CPUs.
- Set the points for all CPUs in the multiple CPU system. (Select from CPU No.1 to No.4.)
  After setting values, click the End button to close the Auto refresh settings screen.



C Controller setting utility(192.168.3.3(Default))

Module information | Event history | SRAM monitoring | Module monitoring | Online operation | System settings
Device settings | I/O assignment settings | Multiple CPU settings | Communication diagnostics

Number of CPUs(*): 4

Host CPU number: No. 1

Operation mode(*)
Operation mode at the time of CPU stop
☑ All CPUs stopped due to an error in CPU 1
☑ All CPUs stopped due to an error in CPU 2
☑ All CPUs stopped due to an error in CPU 3
☑ All CPUs stopped due to an error in CPU 4

Multiple CPU synchronous startup setting(*)
Target CPU
☑ CPU No.1
☑ CPU No.2
☑ CPU No.3
☑ CPU No.4

Online module change(*)
☐ Enable online module change with another CPU
When the online module change with another CPU is enabled,
I/O status outside of the group cannot be read.

I/O sharing when using Multiple CPUs(*)
☐ All CPUs can read all inputs
☐ All CPUs can read all outputs

Multiple CPU high speed transmission area setting | Communication area setting (refresh setting)

☑ Use multiple CPU high speed transmission

CPU	CPU specific send range (*)						
			User setting area			Auto refresh	
	Points(K)	I/O No.	Points	Start	End	Points	Setting
No. 1	3	3E0	3072	10000	13071	0	Setting(Send)
No. 2	3	3E1	3072	10000	13071	0	Setting(Receive)
No. 3	3	3E2	3072	10000	13071	0	Setting(Receive)
No. 4	3	3E3	3072	10000	13071	0	Setting(Receive)

Total 12K points ☐ Advanced settings(*)

The maximum total number of points is 12K.

(*) must be set identically for all CPUs when using multiple CPUs

Import multiple CPU parameters | Default

Connection settings | Load File | Save File | Help | Exit

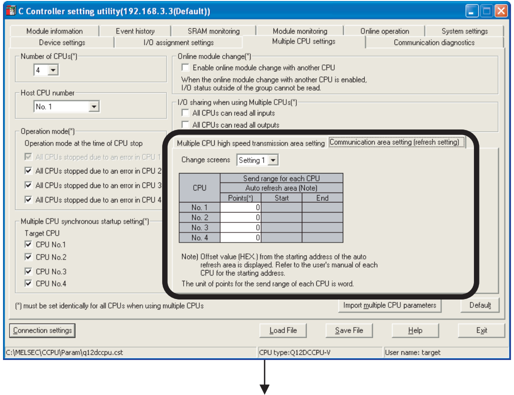
C:\MELSEC\CCPU\Param\q12dccpu.cst | CPU type:Q12DCCPU-V | User name: target

Communication area setting (refresh setting) in the <<Multiple CPU settings>> tab (optional)

- Set this when communicating with the Q06CCPU-V(-B).

(To the next page)

(From the previous page)



Number of CPUs(*): 4

Host CPU number: No. 1

Operation mode(*)
Operation mode at the time of CPU stop
☑ All CPUs stopped due to an error in CPU 1
☑ All CPUs stopped due to an error in CPU 2
☑ All CPUs stopped due to an error in CPU 3
☑ All CPUs stopped due to an error in CPU 4

Multiple CPU synchronous startup setting(*)
Target CPU
☑ CPU No.1
☑ CPU No.2
☑ CPU No.3
☑ CPU No.4

Online module change(*)
□ Enable online module change with another CPU
When the online module change with another CPU is enabled, I/O status outside of the group cannot be read.

I/O sharing when using Multiple CPUs(*)
□ All CPUs can read all inputs
□ All CPUs can read all outputs

Change screens: Setting 1

CPU	Send range for each CPU		
	Auto refresh area (Note)		
	Points(*)	Start	End
No. 1	0		
No. 2	0		
No. 3	0		
No. 4	0		

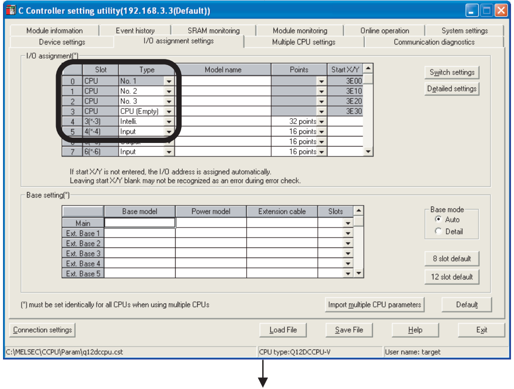
Note) Offset value (HEX.) from the starting address of the auto refresh area is displayed. Refer to the user's manual of each CPU for the starting address.
The unit of points for the send range of each CPU is word.

(*) must be set identically for all CPUs when using multiple CPUs

C:\MELSEC\CCPU\Param\q12dccpu.cst | CPU type:Q12DCCPU-V | User name: target

"Communication area setting (refresh setting)" in the <<Multiple CPU settings>> tab (optional)

- Set the points of the auto refresh area for the data to be sent and received by auto refresh between CPUs.



I/O assignment(*)

	Slot	Type	Model name	Points	Start X/Y
0	CPU	No. 1			3E00
1	CPU	No. 2			3E10
2	CPU	No. 3			3E20
3	CPU	CPU (Empty)			3E30
4	3(*-3)	Intelli.		32 points	
5	4(*-4)	Input		16 points	
6	5(*-5)	Output		16 points	
7	6(*-6)	Input		16 points	

If start X/Y is not entered, the I/O address is assigned automatically.
Leaving start X/Y blank may not be recognized as an error during error check.

Base setting(*)

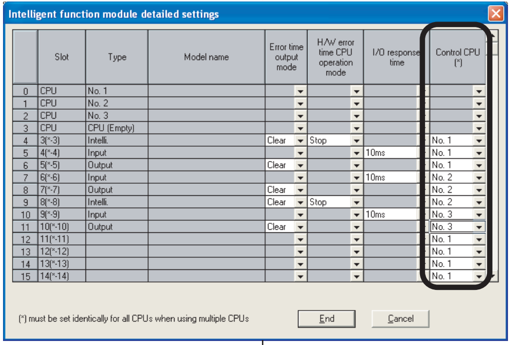
	Base model	Power model	Extension cable	Slots
Main				
Ext. Base 1				
Ext. Base 2				
Ext. Base 3				
Ext. Base 4				
Ext. Base 5				

Base mode
● Auto
○ Detail

(*) must be set identically for all CPUs when using multiple CPUs

C:\MELSEC\CCPU\Param\q12dccpu.cst | CPU type:Q12DCCPU-V | User name: target

In the Type column in the <<I/O assignment settings>> tab, select "CPU (Empty)" for a slot where no module is mounted. Click the Detailed settings button to open the Detailed settings screen.



	Slot	Type	Model name	Error time output mode	H/W error time CPU operation mode	I/O response time	Control CPU (*)
0	CPU	No. 1					
1	CPU	No. 2					
2	CPU	No. 3					
3	CPU	CPU (Empty)					
4	3(*-3)	Intelli.		Clear	Stop		No. 1
5	4(*-4)	Input				10ms	No. 1
6	5(*-5)	Output		Clear			No. 1
7	6(*-6)	Input				10ms	No. 2
8	7(*-7)	Output		Clear			No. 2
9	8(*-8)	Intelli.		Clear	Stop		No. 2
10	9(*-9)	Input				10ms	No. 3
11	10(*-10)	Output		Clear			No. 3
12	11(*-11)						No. 1
13	12(*-12)						No. 1
14	13(*-13)						No. 1
15	14(*-14)						No. 1

(*) must be set identically for all CPUs when using multiple CPUs

Control CPU (setting required)

- Select a control CPU (No.1 to No.4) for each slot.
- This setting is not available for intelligent function modules of function version A.

Set parameters other than these multiple CPU system parameters.

Save the set parameters into a file.

End

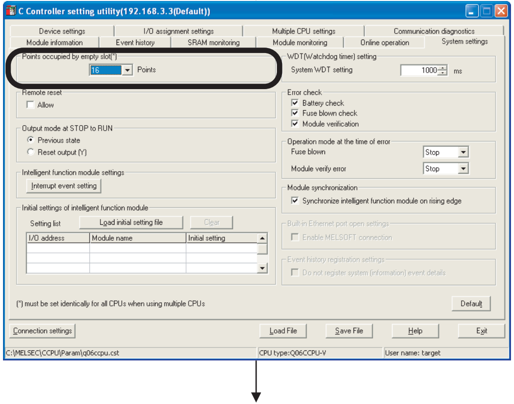
Figure 15.5 Parameter setting procedure for creating a new multiple CPU system

(2) For the Q06CCPU-V(-B)

Start

Start C Controller setting utility.

Connect to the C Controller module.



C Controller setting utility(192.168.3.3(Default))

Points occupied by empty slot(*): 16 Points

WDT(Watchdog timer) setting
System WDT setting: 1000 ms

Remote reset
☐ Allow

Error check
☑ Battery check
☑ Fuse blown check
☑ Module verification

Output mode at STOP to RUN
● Previous state
○ Reset output (Y)

Operation mode at the time of error
Fuse blown: Stop
Module verify error: Stop

Module synchronization
☑ Synchronize intelligent function module on rising edge

Built-in Ethernet port open settings
☐ Enable MELSOFT connection

Event history registration settings
☐ Do not register system (information) event details

Initial settings of intelligent function module

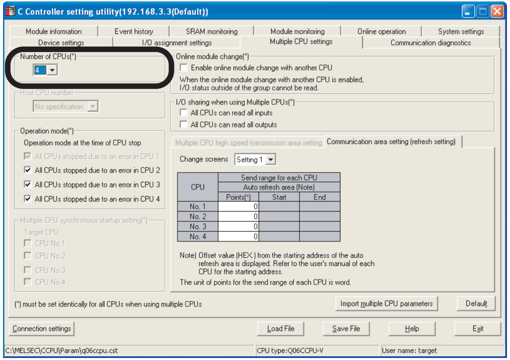
I/O address	Module name	Initial setting

(*) must be set identically for all CPUs when using multiple CPUs

C:\MELSEC\CCPU\Param\q06ccpu.cst | CPU type:Q06CCPU-V | User name: target

"Points occupied by empty slot" in the <<System settings>> tab (optional)

- Set the points to be occupied by one empty slot.
  Default: 16 points



C Controller setting utility(192.168.3.3(Default))

Number of CPUs(*): 4

Host CPU number: No specification

Online module change(*)
☐ Enable online module change with another CPU
When the online module change with another CPU is enabled, I/O status outside of the group cannot be read.

I/O sharing when using Multiple CPUs(*)
☐ All CPUs can read all inputs
☐ All CPUs can read all outputs

Operation mode(*)
Operation mode at the time of CPU stop
☑ All CPUs stopped due to an error in CPU 1
☑ All CPUs stopped due to an error in CPU 2
☑ All CPUs stopped due to an error in CPU 3
☑ All CPUs stopped due to an error in CPU 4

Multiple CPU synchronous startup setting(*)
Target CPU
☐ CPU No.1
☐ CPU No.2
☐ CPU No.3
☐ CPU No.4

Communication area setting (refresh setting)
Change screens: Setting 1

CPU	Send range for each CPU		
	Auto refresh area (Note)		
	Points(*)	Start	End
No. 1	0		
No. 2	0		
No. 3	0		
No. 4	0		

Note) Offset value (HEX.) from the starting address of the auto refresh area is displayed. Refer to the user's manual of each CPU for the starting address.
The unit of points for the send range of each CPU is word.

(*) must be set identically for all CPUs when using multiple CPUs

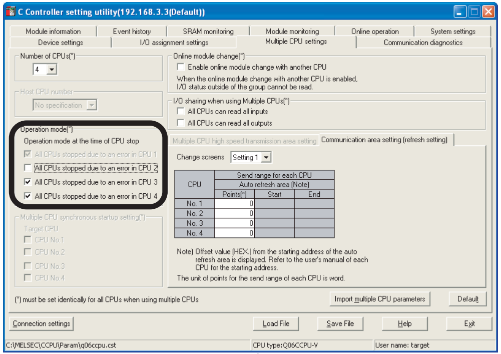
C:\MELSEC\CCPU\Param\q06ccpu.cst | CPU type:Q06CCPU-V | User name: target

"Number of CPUs" in the <<Multiple CPU settings>> tab (setting required)

- Set the number of CPU modules mounted on the main base unit in the multiple CPU system.

(To the next page)

(From the previous page)



Number of CPUs(*): 4

Host CPU number: No specification

Online module change(*)
□ Enable online module change with another CPU
When the online module change with another CPU is enabled, I/O status outside of the group cannot be read.

I/O sharing when using Multiple CPUs(*)
□ All CPUs can read all inputs
□ All CPUs can read all outputs

Operation mode(*)
Operation mode at the time of CPU stop
☑ All CPUs stopped due to an error in CPU 1
□ All CPUs stopped due to an error in CPU 2
☑ All CPUs stopped due to an error in CPU 3
☑ All CPUs stopped due to an error in CPU 4

Multiple CPU synchronous startup setting(*)
Target CPU
□ CPU No.1
□ CPU No.2
□ CPU No.3
□ CPU No.4

Multiple CPU high speed transmission area setting | Communication area setting (refresh setting)

Change screens: Setting 1

CPU	Send range for each CPU		
	Auto refresh area (Note)		
	Points(*)	Start	End
No. 1	0		
No. 2	0		
No. 3	0		
No. 4	0		

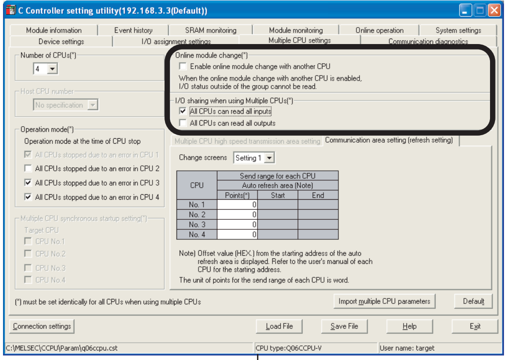
Note) Offset value (HEX.) from the starting address of the auto refresh area is displayed. Refer to the user's manual of each CPU for the starting address.
The unit of points for the send range of each CPU is word.

(*) must be set identically for all CPUs when using multiple CPUs

C:\MELSEC\CCPU\Param\q06ccpu.cst | CPU type:Q06CCPU-V | User name: target

"Operation mode" in the <<Multiple CPU settings>> tab (optional)

- Select whether to stop or continue the operations of all CPUs in the event of a stop error.
  Default: All CPUs stop by a stop error in any of CPU No. 2, 3 and 4 (Checked)
- For example, when "All CPUs stopped due to an error in CPU2" is unchecked, CPUs other than CPU No.2 will continue their operations even if a stop error occurs in CPU No.2.
- The operating mode of CPU No.1 cannot be changed.



Number of CPUs(*): 4

Host CPU number: No specification

Online module change(*)
□ Enable online module change with another CPU
When the online module change with another CPU is enabled, I/O status outside of the group cannot be read.

I/O sharing when using Multiple CPUs(*)
☑ All CPUs can read all inputs
□ All CPUs can read all outputs

Operation mode(*)
Operation mode at the time of CPU stop
☑ All CPUs stopped due to an error in CPU 1
□ All CPUs stopped due to an error in CPU 2
☑ All CPUs stopped due to an error in CPU 3
☑ All CPUs stopped due to an error in CPU 4

Multiple CPU synchronous startup setting(*)
Target CPU
□ CPU No.1
□ CPU No.2
□ CPU No.3
□ CPU No.4

Multiple CPU high speed transmission area setting | Communication area setting (refresh setting)

Change screens: Setting 1

CPU	Send range for each CPU		
	Auto refresh area (Note)		
	Points(*)	Start	End
No. 1	0		
No. 2	0		
No. 3	0		
No. 4	0		

Note) Offset value (HEX.) from the starting address of the auto refresh area is displayed. Refer to the user's manual of each CPU for the starting address.
The unit of points for the send range of each CPU is word.

(*) must be set identically for all CPUs when using multiple CPUs

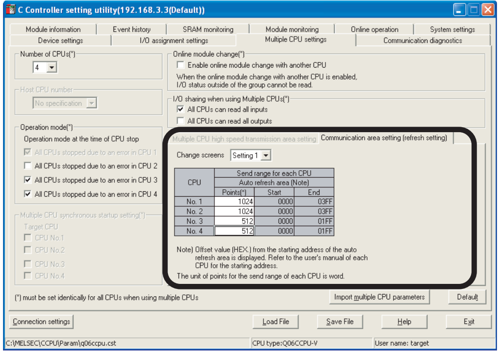
C:\MELSEC\CCPU\Param\q06ccpu.cst | CPU type:Q06CCPU-V | User name: target

"I/O sharing when using Multiple CPUs" in the <<Multiple CPU settings>> tab (optional)

- Set whether to read input/output status data of non-controlled modules or not.
  Default: Not read. (Not checked)

"Online module change" in the <<Multiple CPU settings>> tab (optional)

- Determine whether to enable the online module change setting.
  "Default: Online module change is disabled.
  (Not checked)"
- Do not check the check box when setting "I/O sharing when using Multiple CPUs".
- Check the check box when performing online module change.



Number of CPUs(*): 4

Host CPU number: No specification

Online module change(*)
□ Enable online module change with another CPU
When the online module change with another CPU is enabled, I/O status outside of the group cannot be read.

I/O sharing when using Multiple CPUs(*)
☑ All CPUs can read all inputs
□ All CPUs can read all outputs

Operation mode(*)
Operation mode at the time of CPU stop
☑ All CPUs stopped due to an error in CPU 1
□ All CPUs stopped due to an error in CPU 2
☑ All CPUs stopped due to an error in CPU 3
☑ All CPUs stopped due to an error in CPU 4

Multiple CPU synchronous startup setting(*)
Target CPU
□ CPU No.1
□ CPU No.2
□ CPU No.3
□ CPU No.4

Multiple CPU high speed transmission area setting | Communication area setting (refresh setting)

Change screens: Setting 1

CPU	Send range for each CPU		
	Auto refresh area (Note)		
	Points(*)	Start	End
No. 1	1024	0000	03FF
No. 2	1024	0000	03FF
No. 3	512	0000	01FF
No. 4	512	0000	01FF

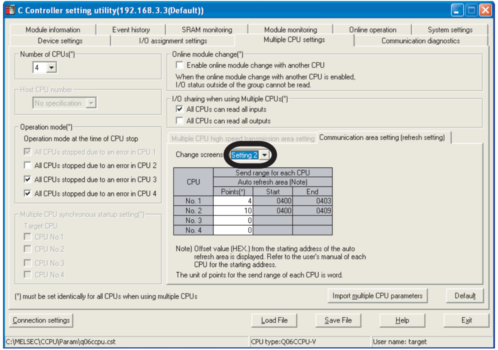
Note) Offset value (HEX.) from the starting address of the auto refresh area is displayed. Refer to the user's manual of each CPU for the starting address.
The unit of points for the send range of each CPU is word.

(*) must be set identically for all CPUs when using multiple CPUs

C:\MELSEC\CCPU\Param\q06ccpu.cst | CPU type:Q06CCPU-V | User name: target

"Communication area setting (refresh setting)" in the <<Multiple CPU settings>> tab (optional)

- Set the points of data to be sent and received by auto refresh between CPUs.



Number of CPUs(*): 4

Host CPU number: No specification

Online module change(*)
□ Enable online module change with another CPU
When the online module change with another CPU is enabled, I/O status outside of the group cannot be read.

I/O sharing when using Multiple CPUs(*)
☑ All CPUs can read all inputs
□ All CPUs can read all outputs

Operation mode(*)
Operation mode at the time of CPU stop
☑ All CPUs stopped due to an error in CPU 1
□ All CPUs stopped due to an error in CPU 2
☑ All CPUs stopped due to an error in CPU 3
☑ All CPUs stopped due to an error in CPU 4

Multiple CPU synchronous startup setting(*)
Target CPU
□ CPU No.1
□ CPU No.2
□ CPU No.3
□ CPU No.4

Multiple CPU high speed transmission area setting | Communication area setting (refresh setting)

Change screens: Setting 2

CPU	Send range for each CPU		
	Auto refresh area (Note)		
	Points(*)	Start	End
No. 1	4	0400	0403
No. 2	10	0400	0409
No. 3	0		
No. 4	0		

Note) Offset value (HEX.) from the starting address of the auto refresh area is displayed. Refer to the user's manual of each CPU for the starting address.
The unit of points for the send range of each CPU is word.

(*) must be set identically for all CPUs when using multiple CPUs

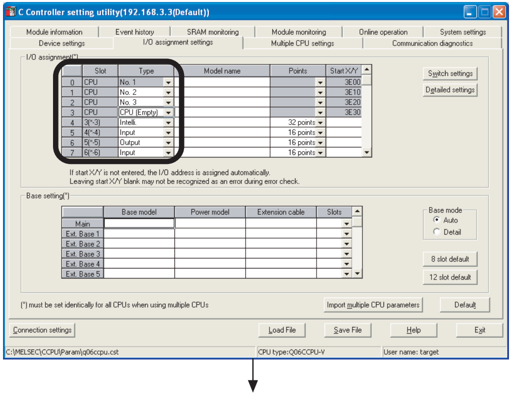
C:\MELSEC\CCPU\Param\q06ccpu.cst | CPU type:Q06CCPU-V | User name: target

"Communication area setting (refresh setting)" in the <<Multiple CPU settings>> tab (optional)

- The "Change screens" enables four different kinds of settings (Setting 1 to Setting 4).

(To the next page)

(From the previous page)



I/O assignment(*)

	Slot	Type	Model name	Points	Start X/Y
0	CPU	No. 1			3E00
1	CPU	No. 2			3E10
2	CPU	No. 3			3E20
3	CPU	CPU (Empty)			3E30
4	3(*-3)	Intelli.		32 points	
5	4(*-4)	Input		16 points	
6	5(*-5)	Output		16 points	
7	6(*-6)	Input		16 points	

If start X/Y is not entered, the I/O address is assigned automatically.
Leaving start X/Y blank may not be recognized as an error during error check.

Base setting(*)

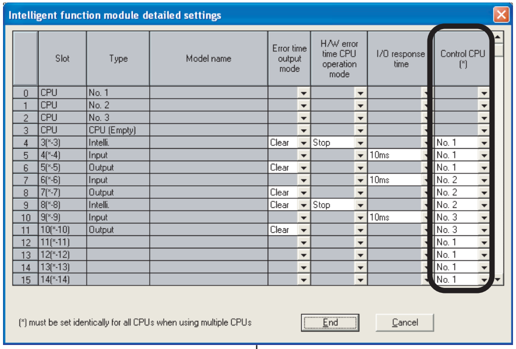
	Base model	Power model	Extension cable	Slots
Main				
Ext. Base 1				
Ext. Base 2				
Ext. Base 3				
Ext. Base 4				
Ext. Base 5				

Base mode
● Auto
○ Detail

(*) must be set identically for all CPUs when using multiple CPUs

C:\MELSEC\CCPU\Param\q06ccpu.cst | CPU type:Q06CCPU-V | User name: target

In the Type column in the <<I/O assignment>> tab, select "CPU (Empty)" for a slot where no module is mounted.
Click the Detailed settings button to open the Detailed settings screen.



	Slot	Type	Model name	Error time output mode	H/W error time CPU operation mode	I/O response time	Control CPU (*)
0	CPU	No. 1					
1	CPU	No. 2					
2	CPU	No. 3					
3	CPU	CPU (Empty)					
4	3(*-3)	Intelli.		Clear	Stop		No. 1
5	4(*-4)	Input				10ms	No. 1
6	5(*-5)	Output		Clear			No. 1
7	6(*-6)	Input				10ms	No. 2
8	7(*-7)	Output		Clear			No. 2
9	8(*-8)	Intelli.		Clear	Stop		No. 2
10	9(*-9)	Input				10ms	No. 3
11	10(*-10)	Output		Clear			No. 3
12	11(*-11)						No. 1
13	12(*-12)						No. 1
14	13(*-13)						No. 1
15	14(*-14)						No. 1

(*) must be set identically for all CPUs when using multiple CPUs

Control CPU (setting required)
- Select a control CPU (No.1 to No.4) for each slot.
- The setting is not available for intelligent function modules of function version A.

Set parameters other than these multiple CPU system parameters.

Save the set parameters into a file.

End

Figure 15.6 Parameter setting procedure for creating a new multiple CPU system

Remark

For C Controller setting utility, refer to the following.

☞ C Controller Module User's Manual (Utility Operation, Programming)

15.2.4 Importing existing multiple CPU parameters

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	○

Remark

To use multiple CPU parameters set by GX Works2, save projects in GX Developer format in advance.

Start

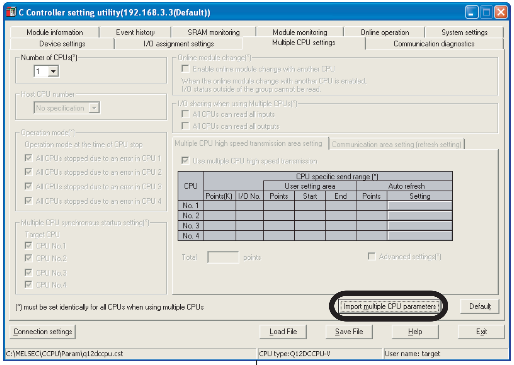
↓

Start C Controller setting utility.

↓

Connect to the C Controller module.

↓



C Controller setting utility(192.168.3.3(Default))

Tabs: Module information | Event history | SRAM monitoring | Module monitoring | Online operation | System settings | Device settings | I/O assignment settings | Multiple CPU settings | Communication diagnostics

Number of CPUs(*): 1

Host CPU number: No specification

Operation mode(*): Operation mode at the time of CPU stop
☑ All CPUs stopped due to an error in CPU 1
☑ All CPUs stopped due to an error in CPU 2
☑ All CPUs stopped due to an error in CPU 3
☑ All CPUs stopped due to an error in CPU 4

Multiple CPU synchronous startup setting(*): Target CPU
☑ CPU No.1
☑ CPU No.2
☑ CPU No.3
☑ CPU No.4

Online module change(*): □ Enable online module change with another CPU
When the online module change with another CPU is enabled, I/O status outside of the group cannot be read.

I/O sharing when using Multiple CPUs(*): □ All CPUs can read all inputs □ All CPUs can read all outputs

Multiple CPU high speed transmission area setting | Communication area setting (refresh setting)

☑ Use multiple CPU high speed transmission

CPU	CPU specific send range (*)						
			User setting area			Auto refresh	
	Points(K)	I/O No.	Points	Start	End	Points	Setting
No. 1							
No. 2							
No. 3							
No. 4							

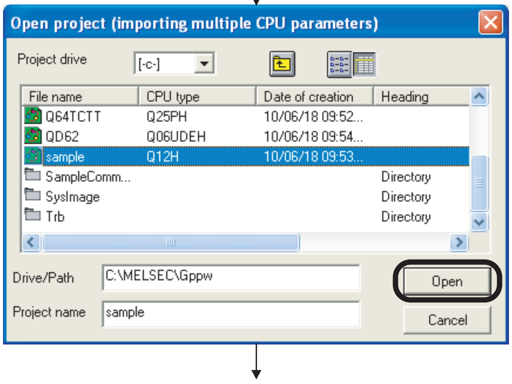
Total points □ Advanced settings(*)

(*) must be set identically for all CPUs when using multiple CPUs

C:\MELSEC\CCPU\Param\q12dccpu.cst | CPU type:Q12DCCPU-V | User name: target

Click the Import multiple CPU parameters button either in the <<Multiple CPU settings>> or <<I/O assignment settings>> tab.

↓



Open project (importing multiple CPU parameters)

Project drive: [-c-]

File name	CPU type	Date of creation	Heading
Q64TCTT	Q25PH	10/06/18 09:52...	
QD62	Q06UDEH	10/06/18 09:54...	
sample	Q12H	10/06/18 09:53...	
SampleComm...			Directory
SysImage			Directory
Trb			Directory

Drive/Path: C:\MELSEC\Gppw

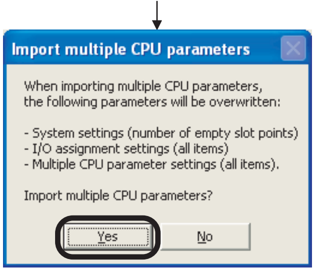
Project name: sample

Setting of the project to be imported
- Select a project in GX Developer or a parameter setting file in C Controller setting utility, from which the number of empty slot points, multiple CPU settings, and I/O assignments will be imported.
- Click the Open button.

↓

(To the next page)

(From the previous page)


Import multiple CPU parameters

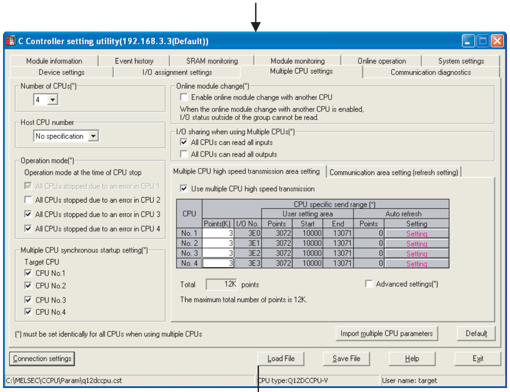
When importing multiple CPU parameters,
the following parameters will be overwritten:

- System settings (number of empty slot points)
- I/O assignment settings (all items)
- Multiple CPU parameter settings (all items).

Import multiple CPU parameters?

Clicking Yes will load the following data from the specified GX Developer project or parameter setting file in C Controller setting utility, and will overwrite the current settings.

- Number of empty slot points
- I/O assignment settings
- Multiple CPU settings


C Controller setting utility(192.168.3.3(Default))

Number of CPUs(*): 4

Host CPU number: No specification

Operation mode(*)
Operation mode at the time of CPU stop
☑ All CPUs stopped due to an error in CPU 1
☐ All CPUs stopped due to an error in CPU 2
☑ All CPUs stopped due to an error in CPU 3
☑ All CPUs stopped due to an error in CPU 4

Multiple CPU synchronous startup setting(*)
Target CPU
☑ CPU No.1
☑ CPU No.2
☑ CPU No.3
☑ CPU No.4

Online module change(*)
☐ Enable online module change with another CPU
When the online module change with another CPU is enabled, I/O status outside of the group cannot be read.

I/O sharing when using Multiple CPUs(*)
☑ All CPUs can read all inputs
☐ All CPUs can read all outputs

Multiple CPU high speed transmission area setting | Communication area setting (refresh setting)

☑ Use multiple CPU high speed transmission

CPU	CPU specific send range (*)						
			User setting area			Auto refresh	
	Points(K)	I/O No.	Points	Start	End	Points	Setting
No. 1	3	3E0	3072	10000	13071	0	Setting
No. 2	3	3E1	3072	10000	13071	0	Setting
No. 3	3	3E2	3072	10000	13071	0	Setting
No. 4	3	3E3	3072	10000	13071	0	Setting

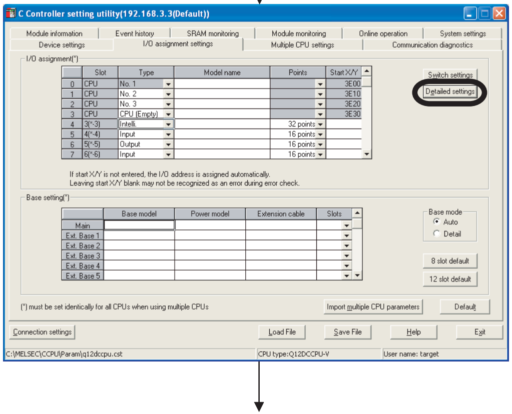
Total 12K points ☐ Advanced settings(*)

The maximum total number of points is 12K.

(*) must be set identically for all CPUs when using multiple CPUs

C:\MELSEC\CCPU\Param\q12dccpu.cst | CPU type:Q12DCCPU-V | User name: target

Verify the multiple CPU settings in the <<Multiple CPU settings>> tab.


C Controller setting utility(192.168.3.3(Default))

I/O assignment(*)

	Slot	Type	Model name	Points	Start X/Y
0	CPU	No. 1			3E00
1	CPU	No. 2			3E10
2	CPU	No. 3			3E20
3	CPU	CPU (Empty)			3E30
4	3(*-3)	Intelli.		32 points	
5	4(*-4)	Input		16 points	
6	5(*-5)	Output		16 points	
7	6(*-6)	Input		16 points	

If start X/Y is not entered, the I/O address is assigned automatically.
Leaving start X/Y blank may not be recognized as an error during error check.

Base setting(*)

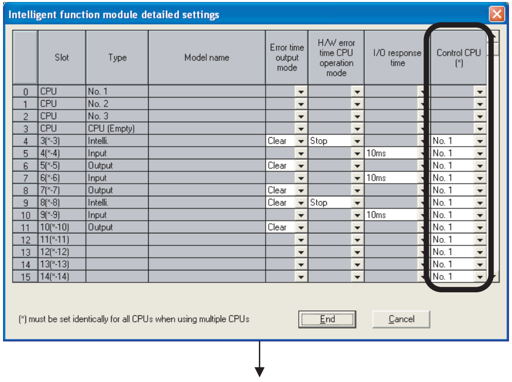
	Base model	Power model	Extension cable	Slots
Main				
Ext. Base 1				
Ext. Base 2				
Ext. Base 3				
Ext. Base 4				
Ext. Base 5				

Base mode
● Auto
○ Detail

(*) must be set identically for all CPUs when using multiple CPUs

C:\MELSEC\CCPU\Param\q12dccpu.cst | CPU type:Q12DCCPU-V | User name: target

Check the I/O assignment and base setting in the <<I/O assignment settings>>.
Click the Detailed settings button to open the Detailed settings screen.


Intelligent function module detailed settings

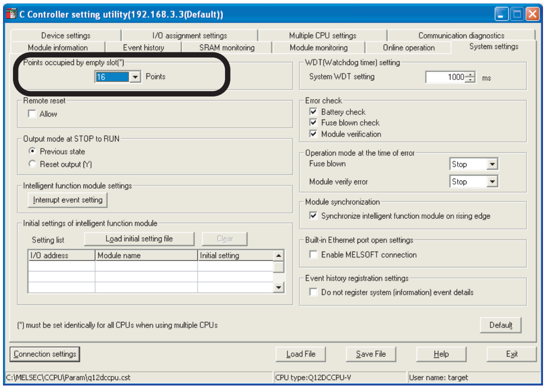
	Slot	Type	Model name	Error time output mode	H/W error time CPU operation mode	I/O response time	Control CPU (*)
0	CPU	No. 1					
1	CPU	No. 2					
2	CPU	No. 3					
3	CPU	CPU (Empty)					
4	3(*-3)	Intelli.		Clear	Stop		No. 1
5	4(*-4)	Input				10ms	No. 1
6	5(*-5)	Output		Clear			No. 1
7	6(*-6)	Input				10ms	No. 1
8	7(*-7)	Output		Clear			No. 1
9	8(*-8)	Intelli.		Clear	Stop		No. 1
10	9(*-9)	Input				10ms	No. 1
11	10(*-10)	Output		Clear			No. 1
12	11(*-11)						No. 1
13	12(*-12)						No. 1
14	13(*-13)						No. 1
15	14(*-14)						No. 1

(*) must be set identically for all CPUs when using multiple CPUs

Confirm the settings of "Control CPU".

(To the next page)

(From previous page)



Points occupied by empty slot(*): 16 Points

Remote reset
- □ Allow

Output mode at STOP to RUN
- ● Previous state
- ○ Reset output (Y)

Intelligent function module settings
- Interrupt event setting

Initial settings of intelligent function module
- Setting list: Load initial setting file / Clear

I/O address	Module name	Initial setting

WDT(Watchdog timer) setting
- System WDT setting: 1000 ms

Error check
- ☑ Battery check
- ☑ Fuse blown check
- ☑ Module verification

Operation mode at the time of error
- Fuse blown: Stop
- Module verify error: Stop

Module synchronization
- ☑ Synchronize intelligent function module on rising edge

Built-in Ethernet port open settings
- □ Enable MELSOFT connection

Event history registration settings
- □ Do not register system (information) event details

(*) must be set identically for all CPUs when using multiple CPUs

C:\MELSEC\CCPU\Param\q12dccpu.cst | CPU type:Q12DCCPU-V | User name: target

Check the "Points occupied by empty slot" setting on the <<System settings>> tab.

Set parameters other than these multiple CPU system parameters.

Save the set parameters into a file.

End

Figure 15.7 Parameter setting procedure for importing multiple CPU system parameters

Remark

For C Controller setting utility, refer to the following.

☞ C Controller Module User's Manual (Utility Operation, Programming)

CHAPTER 16 TROUBLESHOOTING

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	○

This chapter provides error information, return values (error codes) of bus interface functions and MELSEC data link functions, and corrective actions for various problems. For the Q06CCPU-V-B, MELSEC data link functions cannot be used.

16.1 Troubleshooting Basics

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	○

First, perform the following three steps.

(1) Visual check

Check the following:

1) The operating states of the external devices
2) The operating state of the external power supply
3) Whether the cables are connected normally
4) The LEDs on the C Controller module and power supply module to see their operating states
   Example: The RUN and ERR. LEDs on the C Controller module
   The POWER LED on the power supply module
5) Whether the LEDs on the input and output modules turn on reacting to respective input (X) and output (Y) states
6) Whether each module is securely mounted to the base unit

After checking 1) to 6), connect the C Controller module to external devices and check the user program operation.

(2) Checking the fault

Check how the fault state changes by the following operations.

1) Switch the input (X) status between ON and OFF to check if it can be read out by a test program.
2) Switch the output (Y) status between ON and OFF to check if the external device status changes accordingly.

(3) Narrowing down the range

From the results of the above (1) and (2), hardware and/or software that may cause the error can be confirmed.
Narrow down the range from the following to identify the hardware or software, and troubleshoot the problem.

1) C Controller module
2) External device
3) Module (except C Controller module)
4) Cable
5) User program

9 DEVICE DESCRIPTION
10 MULTIPLE CPU SYSTEM OVERVIEW
11 MULTIPLE CPU SYSTEM CONFIGURATION
12 CONCEPT OF MULTIPLE CPU SYSTEM
13 COMMUNICATIONS BETWEEN CPU MODULES
14 PARAMETERS ADDED FOR MULTIPLE CPU SYSTEMS
15 STARTING A MULTIPLE CPU SYSTEM
16 TROUBLESHOOTING

16.2 Troubleshooting

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	○

If a problem has occurred in the C Controller module, first conduct a hardware self-diagnostics to check the C Controller module itself for a hardware failure. (☞ Page 16-75, Section 16.5)

See the following table and refer to the relevant troubleshooting flowchart.

Table 16.1 Troubleshooting list

No.	Problem	Reference
1	The POWER LED on the power supply module turned off.	Page 16-4, Section 16.2.1
2	The MODE LED on the C Controller module is not lit.	Page 16-5, Section 16.2.2
3	The ERR. LED is on or flashing although the C Controller module has started normally.	Page 16-7, Section 16.2.3
4	The RUN LED on the C Controller module remains flashing.	Page 16-8, Section 16.2.4
5	UNIT VERIFY ERR. occurred.	Page 16-9, Section 16.2.5
6	CONTROL-BUS.ERR. occurred.	Page 16-10, Section 16.2.6
7	No communication is available between the development environment (personal computer) and C Controller module.	Page 16-12, Section 16.2.7
8	A file such as a program file cannot be written to the C Controller module.	Page 16-15, Section 16.2.8
9	An error occurs during user program execution although the C Controller module has started normally and the ERR. LED is not on or flashing.	Page 16-16, Section 16.2.9
10	A file system error has occurred on the standard RAM, standard ROM, or CompactFlash card.	Page 16-18, Section 16.2.10
11	The LEDs on an output module do not turn on.	Page 16-20, Section 16.2.11
12	An output load device of an output module does not turn on.	Page 16-21, Section 16.2.12
13	The C Controller module does not operate normally due to script file execution.	Page 16-22, Section 16.2.13
14	SW3PVC-CCPU cannot be uninstalled correctly. ☞ C Controller Module User's Manual (Utility Operation, Programming)	—
15	C Controller module became faulty and it needs to be reset to the factory-set conditions.	Page 6-72, Section 6.10
16	An error occurred during user program download or ld command execution.	Page 16-25, Section 16.2.14
17	Data cannot be read from or written to the device specified with a MELSEC data link function.	Page 16-27, Section 16.2.15
18	An error occurred during reading from or writing to the standard RAM, standard ROM, CompactFlash card, or RAM disc with FTP.	Page 16-28, Section 16.2.16
19	Telnet connection cannot be made.	Page 16-30, Section 16.2.17
20	FTP connection cannot be made.	Page 16-32, Section 16.2.18

Remark

For C Controller setting utility, refer to the following.

☞ C Controller Module User's Manual (Utility Operation, Programming)

9 DEVICE DESCRIPTION
10 MULTIPLE CPU SYSTEM OVERVIEW
11 MULTIPLE CPU SYSTEM CONFIGURATION
12 CONCEPT OF MULTIPLE CPU SYSTEM
13 COMMUNICATIONS BETWEEN CPU MODULES
14 PARAMETERS ADDED FOR MULTIPLE CPU SYSTEMS
15 STARTING A MULTIPLE CPU SYSTEM
16 TROUBLESHOOTING

16.2.1 When the POWER LED on the power supply module turned off

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	○

The following flowchart shows the procedures to be taken when the POWER LED of the power supply module turns off at the time of power-on or during operation of the C Controller module.

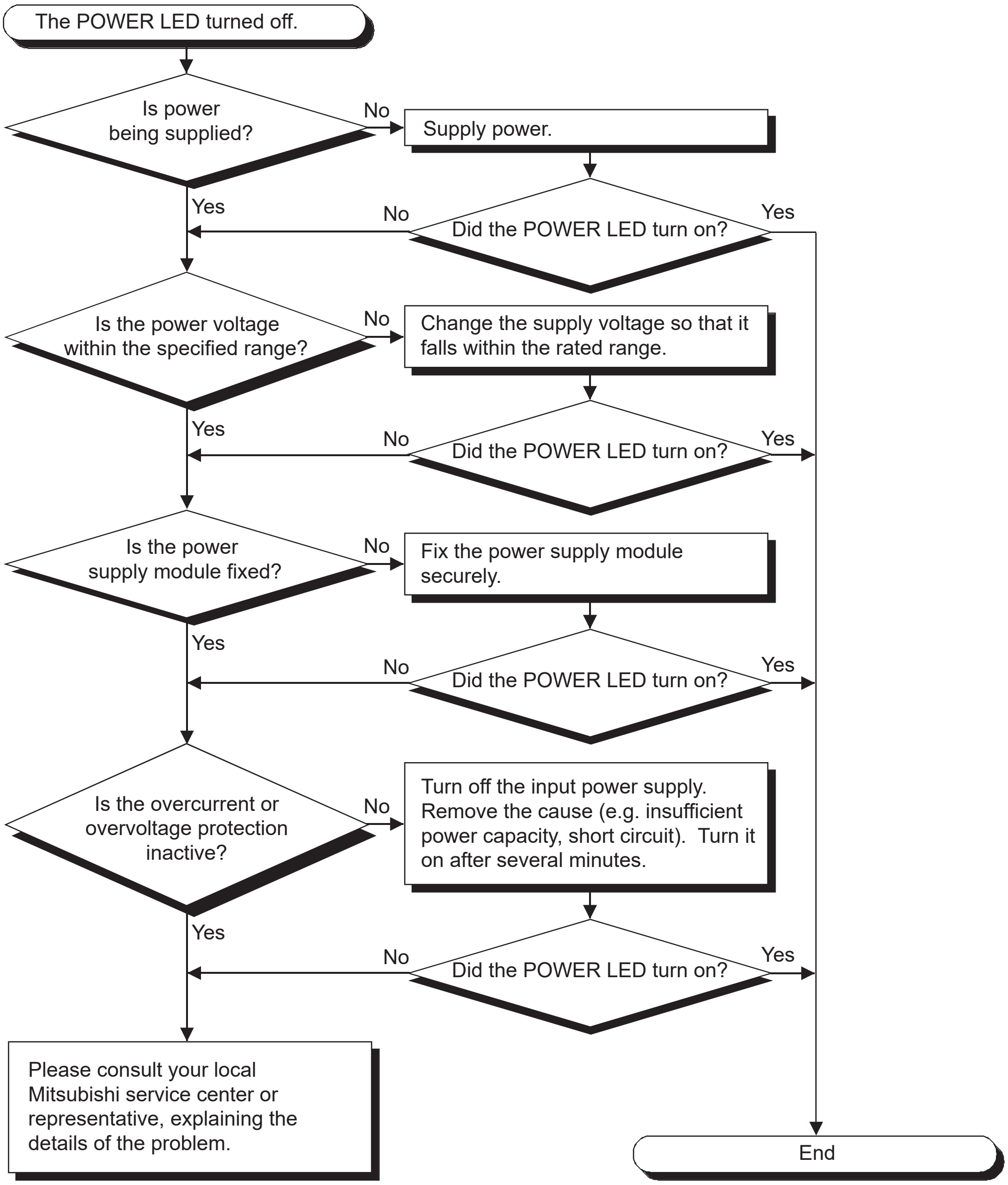

The POWER LED turned off.
Is power being supplied?
No
Supply power.
Yes
No
Did the POWER LED turn on?
Yes
Is the power voltage within the specified range?
No
Change the supply voltage so that it falls within the rated range.
Yes
No
Did the POWER LED turn on?
Yes
Is the power supply module fixed?
No
Fix the power supply module securely.
Yes
No
Did the POWER LED turn on?
Yes
Is the overcurrent or overvoltage protection inactive?
No
Turn off the input power supply. Remove the cause (e.g. insufficient power capacity, short circuit). Turn it on after several minutes.
Yes
No
Did the POWER LED turn on?
Yes
Please consult your local Mitsubishi service center or representative, explaining the details of the problem.
End

Figure 16.1 When the POWER LED of the power supply module turned off

16.2.2 When the MODE LED is not lit

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	○

The following flowchart shows the procedures to be taken when the MODE LED on the C Controller module does not light up when the C Controller system is powered on.

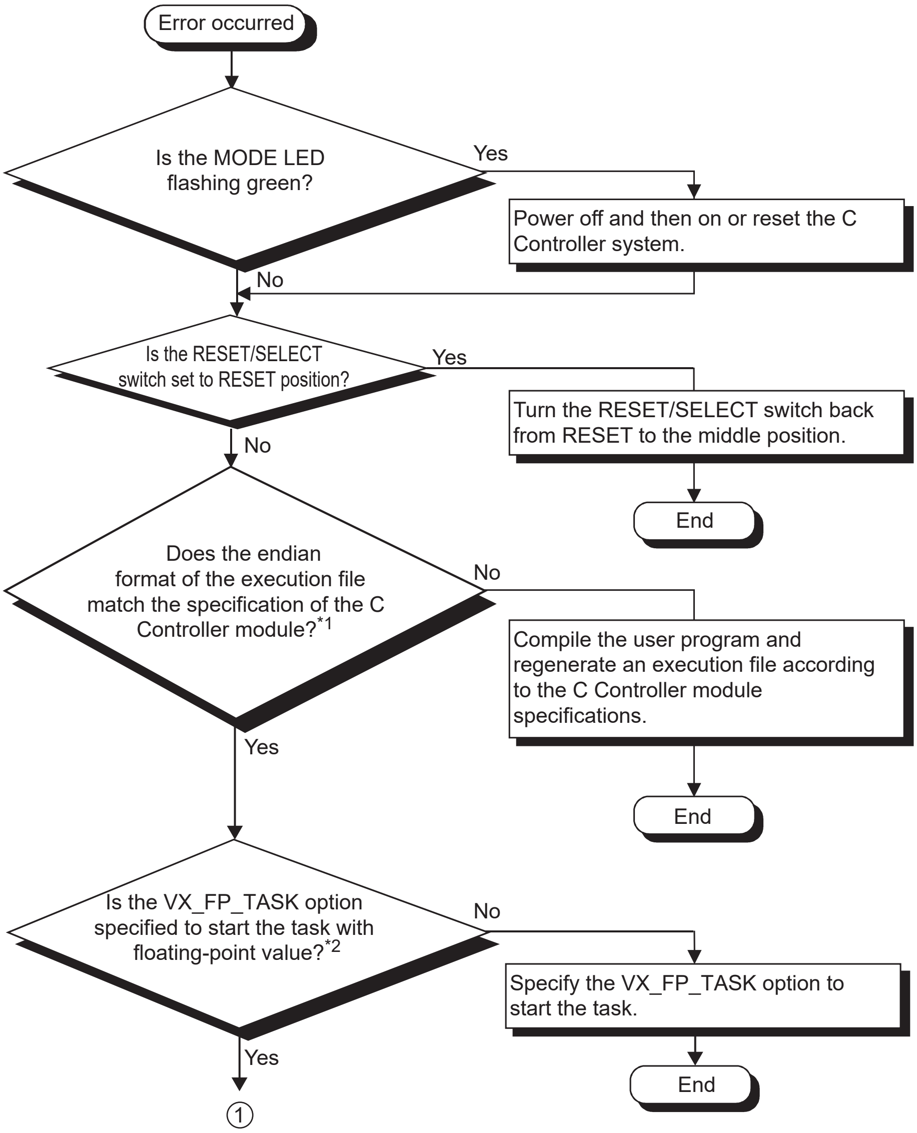

Error occurred
Is the MODE LED flashing green?
Yes
Power off and then on or reset the C Controller system.
No
Is the RESET/SELECT switch set to RESET position?
Yes
Turn the RESET/SELECT switch back from RESET to the middle position.
End
No
Does the endian format of the execution file match the specification of the C Controller module?*1
No
Compile the user program and regenerate an execution file according to the C Controller module specifications.
End
Yes
Is the VX_FP_TASK option specified to start the task with floating-point value?*2
No
Specify the VX_FP_TASK option to start the task.
End
Yes
①

Figure 16.2 When the MODE LED is not lit

* 1 For precautions on the functions, refer to the following.

☞ C Controller Module User's Manual (Utility Operation, Programming)

* 2 Always specify the VX_FP_TASK option for the third argument of taskSpawn when activating a task that:

- Performs floating-point operations
- Calls a function that returns a floating-point value.
- Calls a function that takes a floating-point value as an argument.

If the above task is activated without the VX_FP_TASK option specified, the operating system may run away.
When specifying the VX_FP_TASK option in a script file, refer to the following.

☞ C Controller Module User's Manual (Utility Operation, Programming)

For details on the VX_FP_TASK option, refer to the following.

☞ Manuals for VxWorks

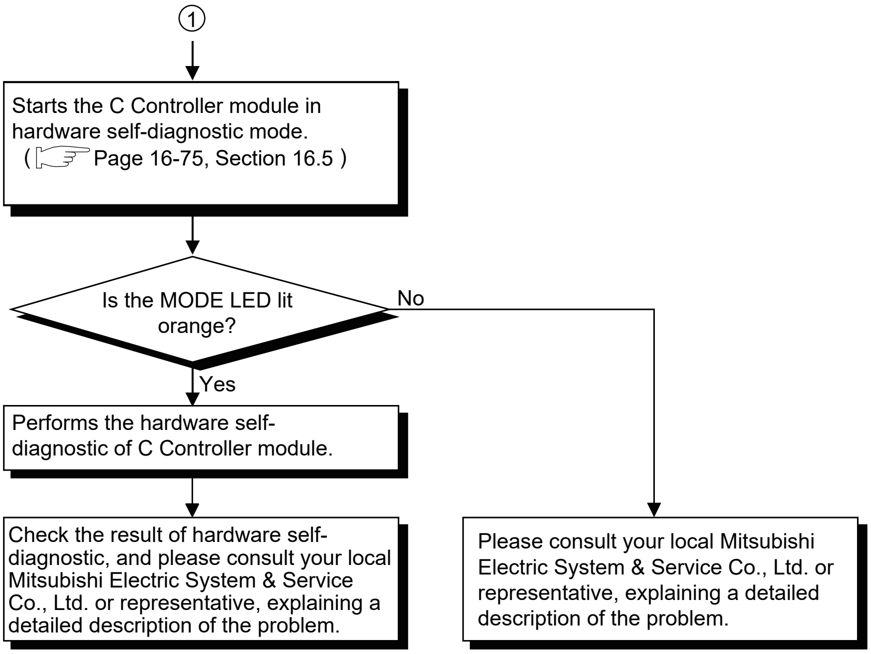

1
Starts the C Controller module in hardware self-diagnostic mode. (☞ Page 16-75, Section 16.5)
Is the MODE LED lit orange?
No
Yes
Performs the hardware self-diagnostic of C Controller module.
Check the result of hardware self-diagnostic, and please consult your local Mitsubishi Electric System & Service Co., Ltd. or representative, explaining a detailed description of the problem.
Please consult your local Mitsubishi Electric System & Service Co., Ltd. or representative, explaining a detailed description of the problem.

Figure 16.2 When the MODE LED is not lit (continued)

16.2.3 When the ERR. LED is on or flashing

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	○

The following flowchart shows the procedures to be taken when the ERR. LED on the C Controller module turns on or flashes at the time of power-on or operation start or during operation of the C Controller system.

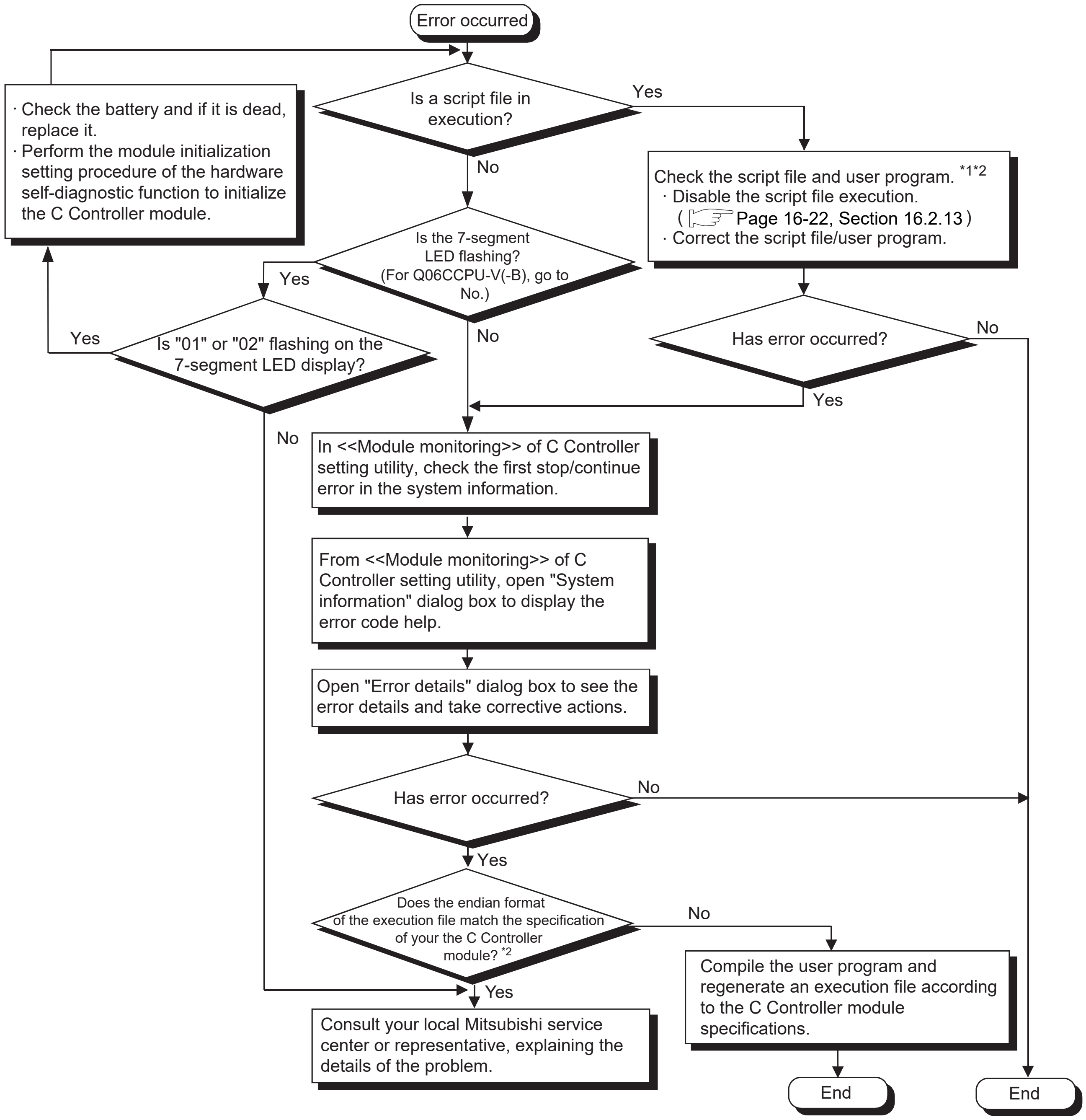

Error occurred
Is a script file in execution?
Yes
No
Check the script file and user program. *1*2
· Disable the script file execution.
(Page 16-22, Section 16.2.13)
· Correct the script file/user program.
Has error occurred?
No
Yes
· Check the battery and if it is dead, replace it.
· Perform the module initialization setting procedure of the hardware self-diagnostic function to initialize the C Controller module.
Is the 7-segment LED flashing? (For Q06CCPU-V(-B), go to No.)
Yes
No
Is "01" or "02" flashing on the 7-segment LED display?
Yes
No
In <<Module monitoring>> of C Controller setting utility, check the first stop/continue error in the system information.
From <<Module monitoring>> of C Controller setting utility, open "System information" dialog box to display the error code help.
Open "Error details" dialog box to see the error details and take corrective actions.
Has error occurred?
No
Yes
Does the endian format of the execution file match the specification of your the C Controller module? *2
No
Yes
Compile the user program and regenerate an execution file according to the C Controller module specifications.
Consult your local Mitsubishi service center or representative, explaining the details of the problem.
End
End

Figure 16.3 When the ERR. LED is on or flashing

* 1 Specify a correct script file or user program with special care. Specifying an incorrect script file or user program may result in unreliable C Controller module operation (an error, communication failure, illegal error code return, etc.)

* 2 For precautions on the functions and script files, refer to the following.
C Controller Module User's Manual (Utility Operation, Programming)

16.2.4 When the RUN LED remains flashing

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	○

When the RUN LED on the C Controller module remains flashing, it means the script file (command) remains in execution.

Perform the following.

- Correct the script file and user program.
- Disable the script file execution. (☞ Page 16-22, Section 16.2.13)

Specify a correct script file and user program with special care. Specifying an incorrect script file or user program may result in unreliable C Controller module operation (the error status, communication failure, illegal error code return, etc.)

Remark

For script files, refer to the following.

☞ C Controller Module User's Manual (Utility Operation, Programming)

16.2.5 When UNIT VERIFY ERR. occurred

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	○

The following flowchart shows the procedures to be taken when a module verify error (error code: 2000, UNIT VERIFY ERR.) occurred when the C Controller system is powered on or during operation.

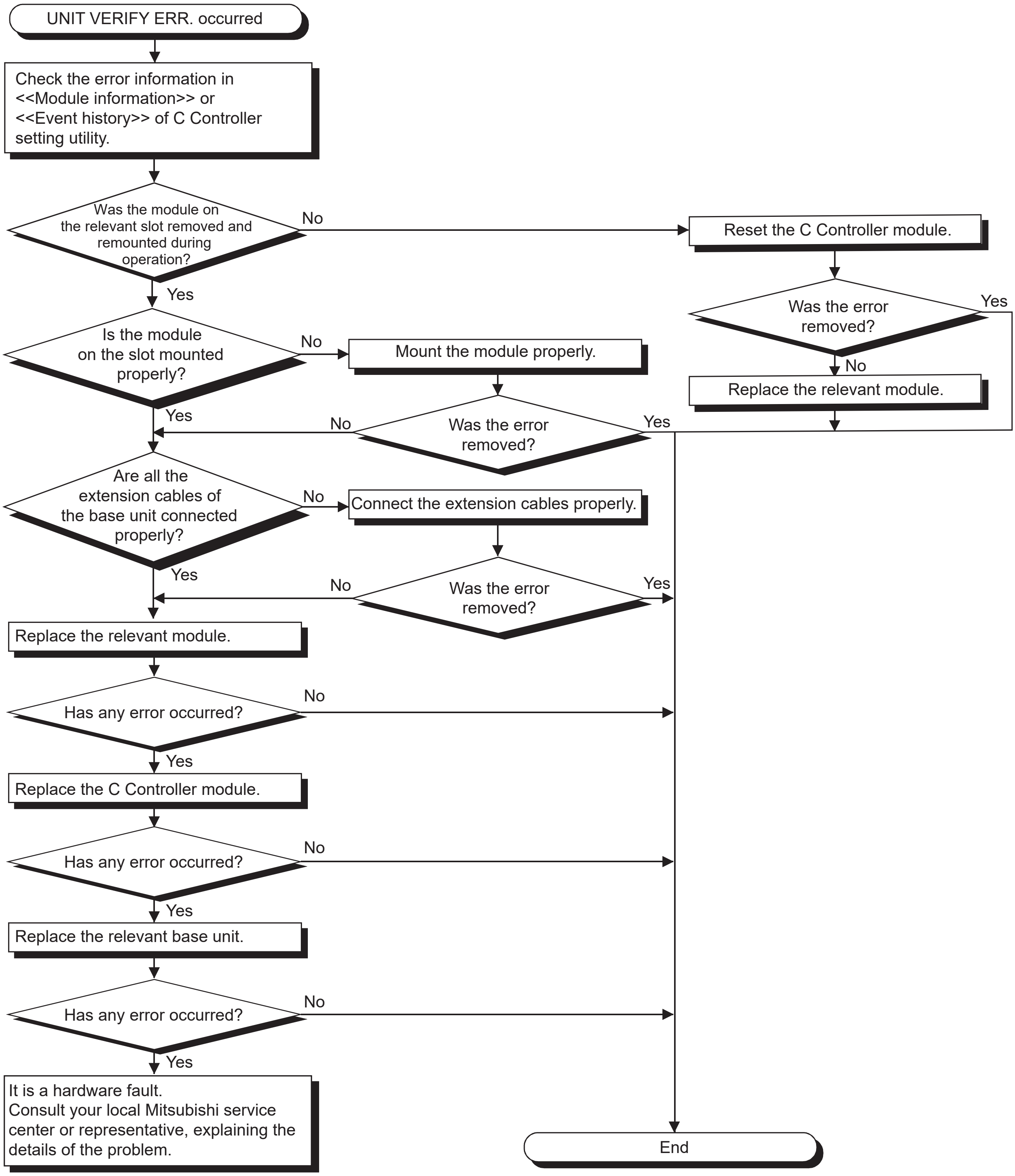

UNIT VERIFY ERR. occurred
Check the error information in <<Module information>> or <<Event history>> of C Controller setting utility.
Was the module on the relevant slot removed and remounted during operation?
No
Reset the C Controller module.
Was the error removed?
Yes
No
Replace the relevant module.
Yes
Is the module on the slot mounted properly?
No
Mount the module properly.
Yes
No
Was the error removed?
Yes
Are all the extension cables of the base unit connected properly?
No
Connect the extension cables properly.
Yes
No
Was the error removed?
Yes
Replace the relevant module.
Has any error occurred?
No
Yes
Replace the C Controller module.
Has any error occurred?
No
Yes
Replace the relevant base unit.
Has any error occurred?
No
Yes
It is a hardware fault. Consult your local Mitsubishi service center or representative, explaining the details of the problem.
End

Figure 16.4 When UNIT VERIFY ERR. occurred

16.2.6 When CONTROL-BUS.ERR. occurred

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	○

The following flowchart shows the procedures to be taken when a control bus error (error code: 1412 to 1416, CONTROL-BUS.ERR.) occurred when the C Controller system is powered on or during operation.

Note that this flowchart can be used only when the relevant slot and base unit can be identified by the error code.

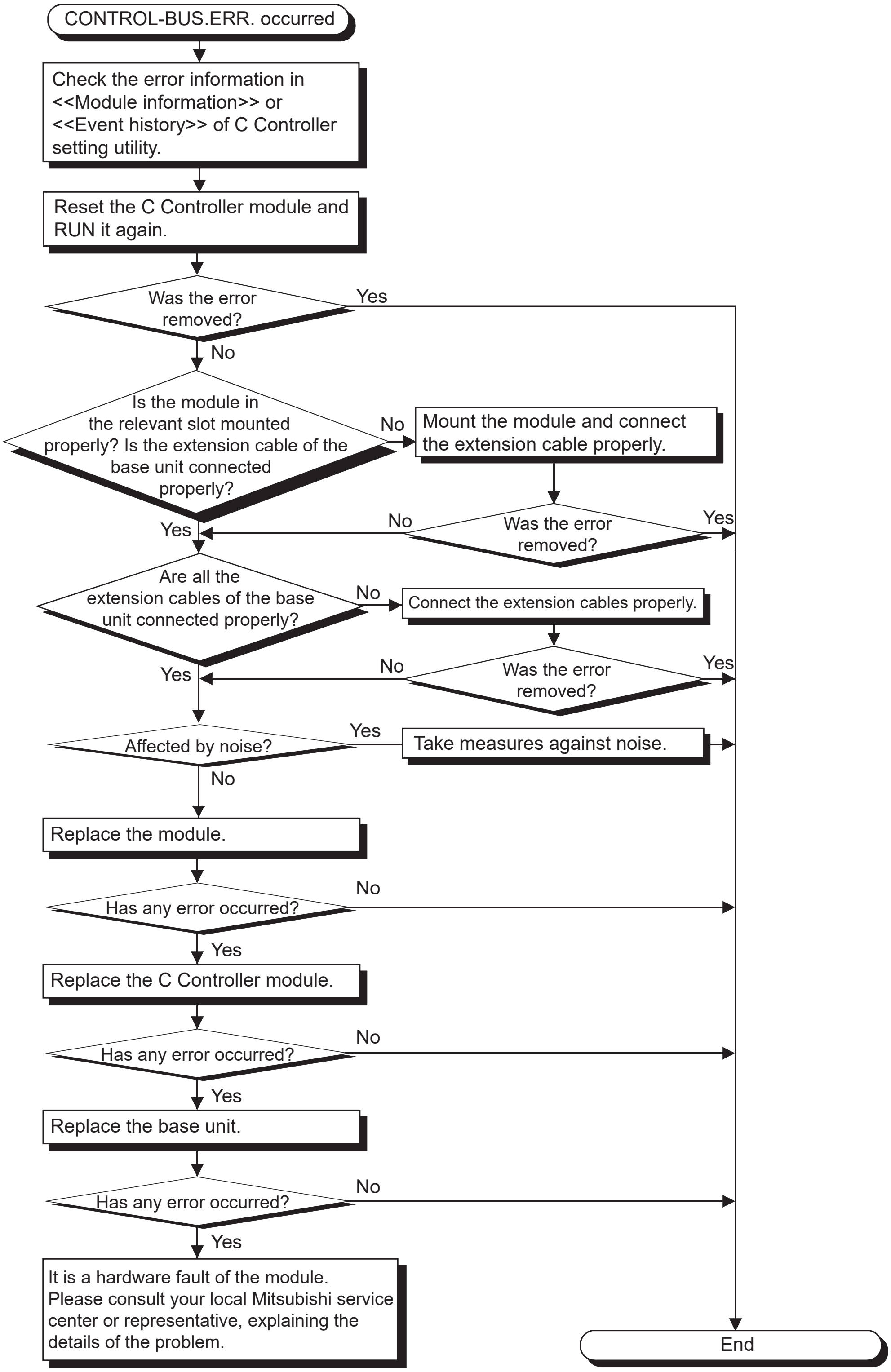

CONTROL-BUS.ERR. occurred
Check the error information in <<Module information>> or <<Event history>> of C Controller setting utility.
Reset the C Controller module and RUN it again.
Was the error removed?
Yes
No
Is the module in the relevant slot mounted properly? Is the extension cable of the base unit connected properly?
No
Mount the module and connect the extension cable properly.
Was the error removed?
No
Yes
Yes
Are all the extension cables of the base unit connected properly?
No
Connect the extension cables properly.
Was the error removed?
No
Yes
Yes
Affected by noise?
Yes
Take measures against noise.
No
Replace the module.
Has any error occurred?
No
Yes
Replace the C Controller module.
Has any error occurred?
No
Yes
Replace the base unit.
Has any error occurred?
No
Yes
It is a hardware fault of the module. Please consult your local Mitsubishi service center or representative, explaining the details of the problem.
End

Figure 16.5 When CONTROL-BUS. ERR. occurred

16.2.7 When no communication is available between the development environment (personal computer) and C Controller module

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	○

The following flowchart shows the procedure to be taken when the C Controller module cannot communicate with the development environment (personal computer) during C Controller system operation.

☒**POINT**

When using Q12DCCPU-V with a serial number whose first five digits are "15102" or later, check if the connected module is set to the basic mode. If not, switch the setting to the basic mode with the initialization setting.

☞ Page 6-72, Section 6.10

(1) When each utility on the development environment (personal computer) cannot communicate with the C Controller module

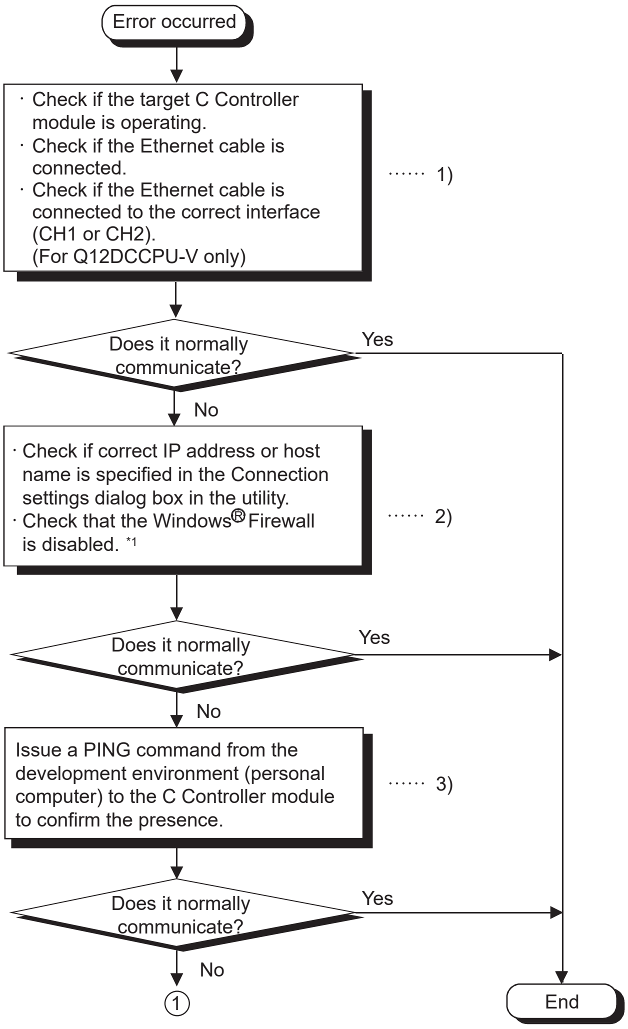

Error occurred
· Check if the target C Controller module is operating.
· Check if the Ethernet cable is connected.
· Check if the Ethernet cable is connected to the correct interface (CH1 or CH2). (For Q12DCCPU-V only)
...... 1)
Does it normally communicate?
Yes
No
· Check if correct IP address or host name is specified in the Connection settings dialog box in the utility.
· Check that the Windows® Firewall is disabled. *1
...... 2)
Does it normally communicate?
Yes
No
Issue a PING command from the development environment (personal computer) to the C Controller module to confirm the presence.
...... 3)
Does it normally communicate?
Yes
No
1
End

Figure 16.6 When communication is not available

*1 For Windows® Firewall settings, refer to the following.

☞ C Controller Module User's Manual (Utility Operation, Programming)

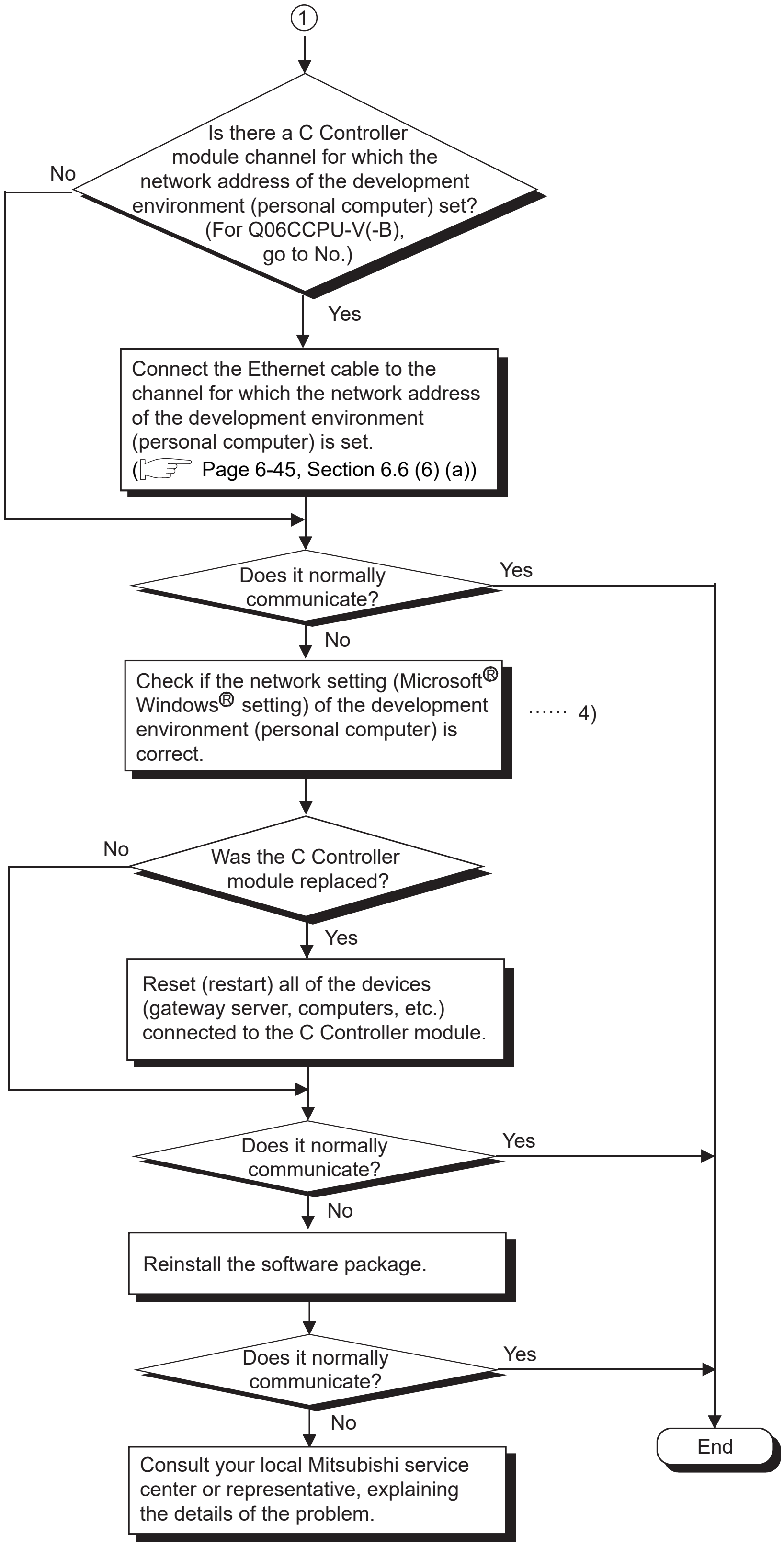

1
Is there a C Controller module channel for which the network address of the development environment (personal computer) set? (For Q06CCPU-V(-B), go to No.)
No
Yes
Connect the Ethernet cable to the channel for which the network address of the development environment (personal computer) is set. (☞ Page 6-45, Section 6.6 (6) (a))
Does it normally communicate?
Yes
No
Check if the network setting (Microsoft® Windows® setting) of the development environment (personal computer) is correct.
...... 4)
Was the C Controller module replaced?
No
Yes
Reset (restart) all of the devices (gateway server, computers, etc.) connected to the C Controller module.
Does it normally communicate?
Yes
No
Reinstall the software package.
Does it normally communicate?
Yes
No
Consult your local Mitsubishi service center or representative, explaining the details of the problem.
End

Figure 16.6 When communication is not available (continued)

(2) When the C Controller module is not accessible from the development environment (personal computer) by FTP

Check the following:

- Steps 1) to 4) of the flowchart in (1) in this section
- Priority of the task in the user program

(3) When Workbench or Tornado on the development environment (personal computer) cannot be connected to the C Controller module

Check the following:

- Steps 1) to 4) of the flowchart in (1) in this section
- Settings of Workbench or Tornado

☒POINT

(1) If the problem cannot be resolved by taking actions with the C Controller module executing a script file, check the script file and user program.

- Correct the script file and user program.
- Disable the script file execution.

(☞ Page 16-22, Section 16.2.13)

Specify a correct script file and user program with special care. Specifying an incorrect script file or user program may result in unreliable C Controller module operation (the error status, communication failure, illegal error code return, etc.)

(2) If a stop error occurs in another CPU in a multiple CPU system, communication between the development environment (personal computer) and C Controller module may be disabled.

In such a case, remove the cause of the stop error in advance.

Remark

For precautions on the functions and program debugging, and script files, refer to the following.

☞ C Controller Module User's Manual (Utility Operation, Programming)

16.2.8 When a file (program) cannot be written

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	○

The following flowchart shows the procedures to be taken when a file such as a user program file cannot be written to the C Controller module.

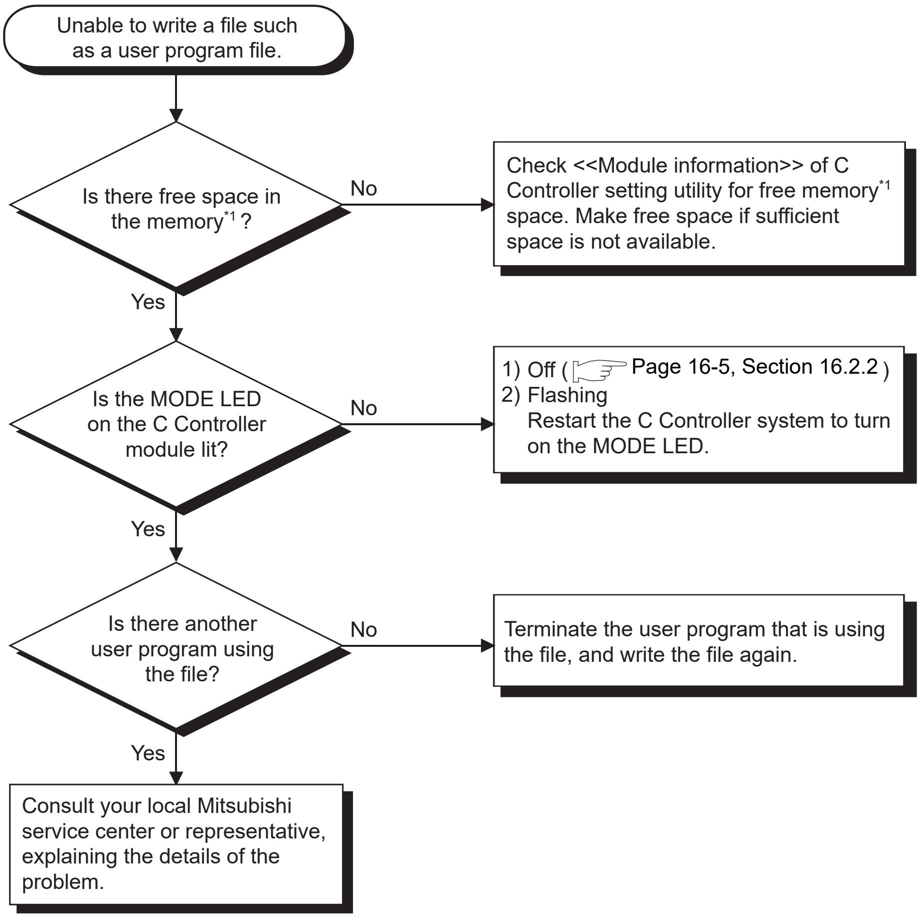

Unable to write a file such as a user program file.
Is there free space in the memory*1 ?
No
Check <<Module information>> of C Controller setting utility for free memory*1 space. Make free space if sufficient space is not available.
Yes
Is the MODE LED on the C Controller module lit?
No
1) Off (Page 16-5, Section 16.2.2)
2) Flashing
Restart the C Controller system to turn on the MODE LED.
Yes
Is there another user program using the file?
No
Terminate the user program that is using the file, and write the file again.
Yes
Consult your local Mitsubishi service center or representative, explaining the details of the problem.

Figure 16.7 When a program cannot be written

* 1 Standard RAM, standard ROM or CompactFlash card

16.2.9 When an error occurs during user program execution

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	○

The following flowchart shows the procedures to be taken when an error occurred during execution of a user program in the C Controller system.

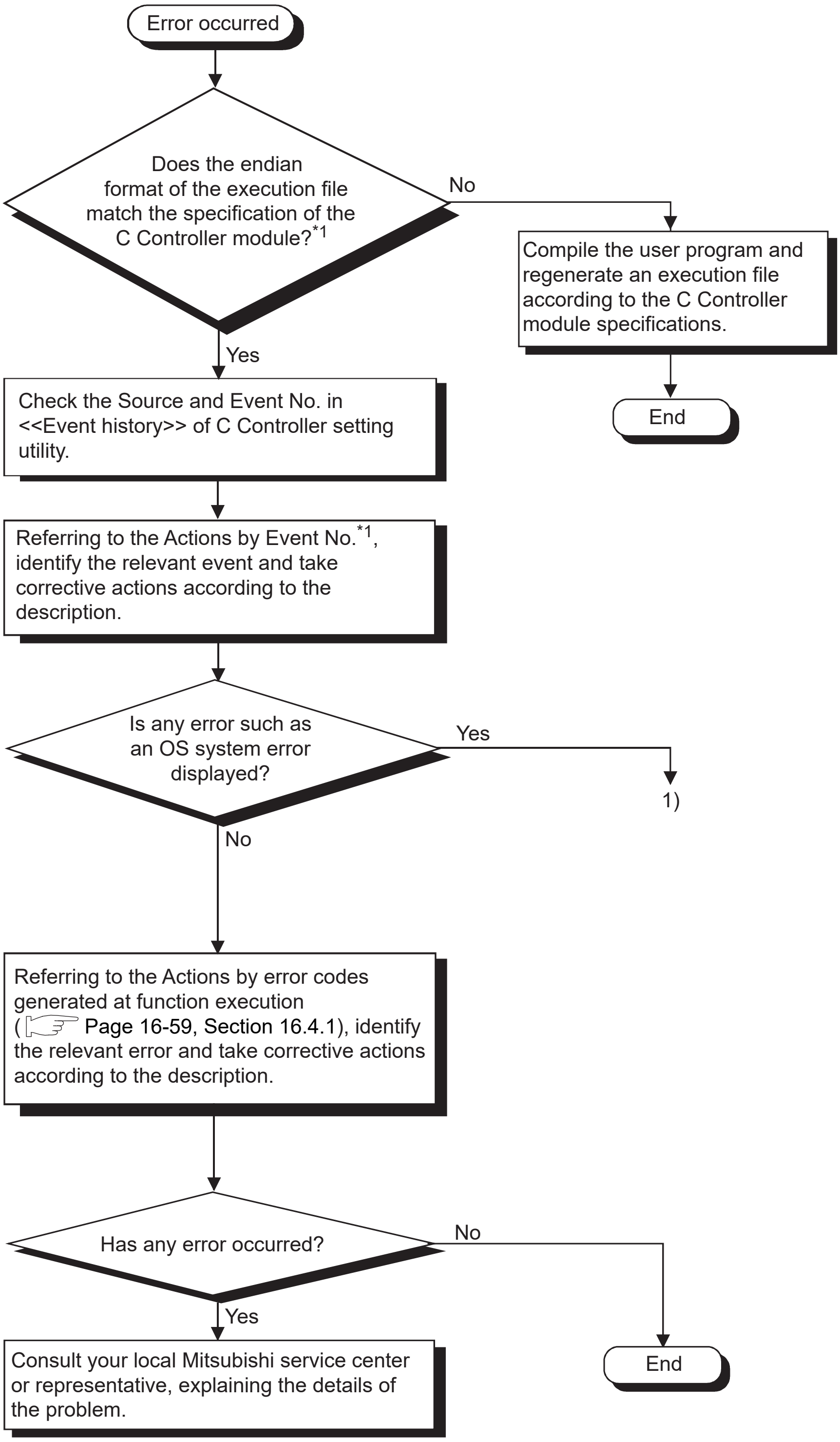

Error occurred
Does the endian format of the execution file match the specification of the C Controller module?*1
No
Compile the user program and regenerate an execution file according to the C Controller module specifications.
End
Yes
Check the Source and Event No. in <<Event history>> of C Controller setting utility.
Referring to the Actions by Event No.*1, identify the relevant event and take corrective actions according to the description.
Is any error such as an OS system error displayed?
Yes
1)
No
Referring to the Actions by error codes generated at function execution (☞ Page 16-59, Section 16.4.1), identify the relevant error and take corrective actions according to the description.
Has any error occurred?
No
End
Yes
Consult your local Mitsubishi service center or representative, explaining the details of the problem.

Figure 16.8 When an error occurs during user program execution

[1] For precautions on user programs and troubleshooting actions by event No., refer to the following.
☞ C Controller Module User's Manual (Utility Operation, Programming)

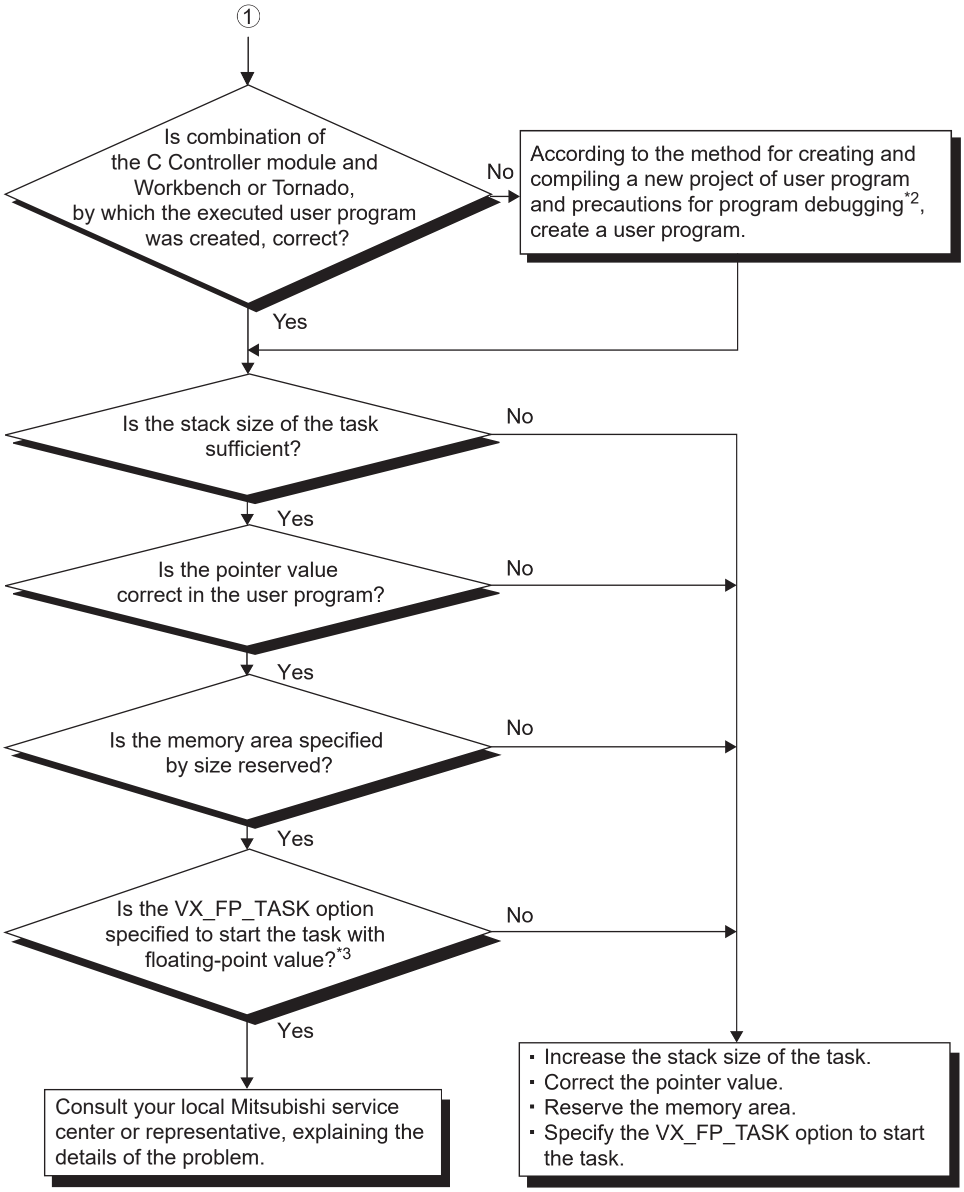

1
Is combination of the C Controller module and Workbench or Tornado, by which the executed user program was created, correct?
No
According to the method for creating and compiling a new project of user program and precautions for program debugging*2, create a user program.
Yes
Is the stack size of the task sufficient?
No
Yes
Is the pointer value correct in the user program?
No
Yes
Is the memory area specified by size reserved?
No
Yes
Is the VX_FP_TASK option specified to start the task with floating-point value?*3
No
Yes
Consult your local Mitsubishi service center or representative, explaining the details of the problem.
▪ Increase the stack size of the task.
▪ Correct the pointer value.
▪ Reserve the memory area.
▪ Specify the VX_FP_TASK option to start the task.

Figure 16.8 When an error occurs during user program execution (continued)

* 2 For creating a new project for the user program, a compiling method, and precautions for program debugging, refer to the following.

☞ C Controller Module User's Manual (Utility Operation, Programming)

* 3 Always specify the VX_FP_TASK option for the third argument of taskSpawn when activating a task that:

- Performs floating-point operations
- Calls a function that returns a floating-point value.
- Calls a function that takes a floating-point value as an argument.

If the above task is activated without the VX_FP_TASK option specified, the operating system may run away.

When specifying the VX_FP_TASK option in a script file, refer to the following.

☞ C Controller Module User's Manual (Utility Operation, Programming)

For details on the VX_FP_TASK option, refer to the following.

☞ Manuals for VxWorks

9 DEVICE DESCRIPTION
10 MULTIPLE CPU SYSTEM OVERVIEW
11 MULTIPLE CPU SYSTEM CONFIGURATION
12 CONCEPT OF MULTIPLE CPU SYSTEM
13 COMMUNICATIONS BETWEEN CPU MODULES
14 PARAMETERS ADDED FOR MULTIPLE CPU SYSTEMS
15 STARTING A MULTIPLE CPU SYSTEM
16 TROUBLESHOOTING

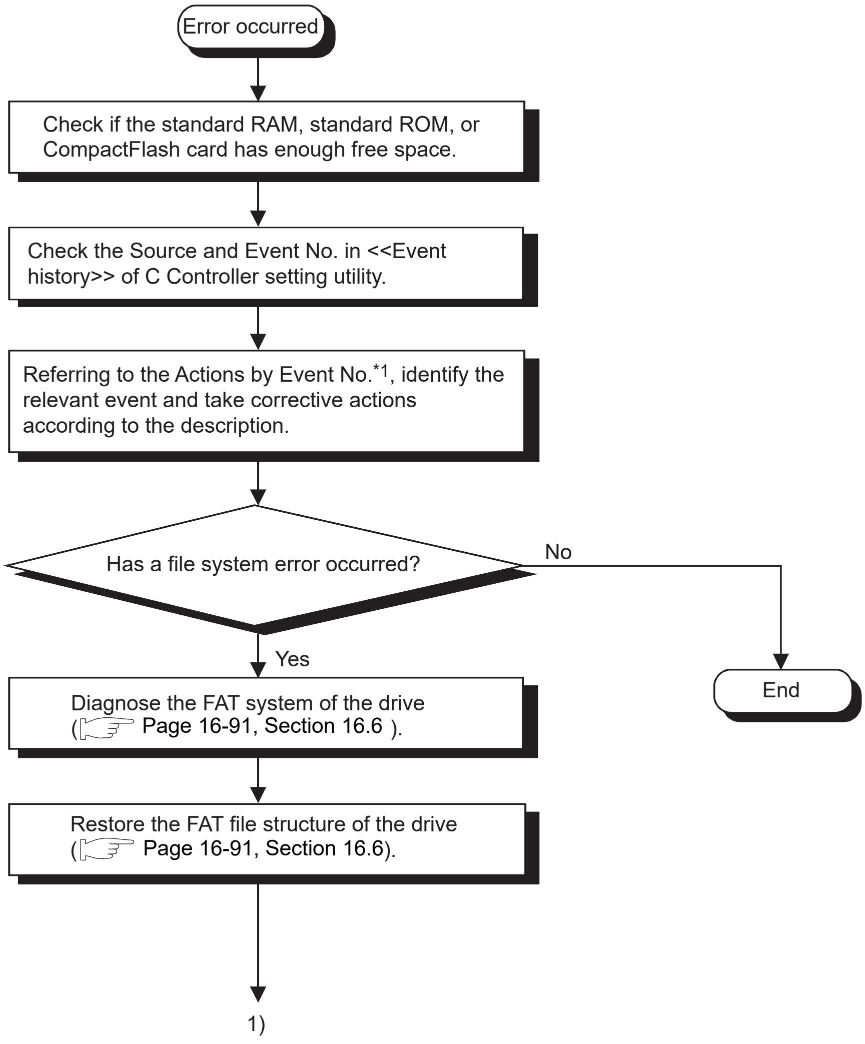
16.2.10 When a file system error occurred

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	○

The following flowchart shows the procedures to be taken when a file system error occurred when the standard RAM, standard ROM, or CompactFlash card is accessed.

☒POINT

Data cannot be written to the standard ROM when the standard ROM has been shut down.
Restart the C Controller module and then write the data again.



Error occurred

Check if the standard RAM, standard ROM, or CompactFlash card has enough free space.

Check the Source and Event No. in <<Event history>> of C Controller setting utility.

Referring to the Actions by Event No.*1, identify the relevant event and take corrective actions according to the description.

Has a file system error occurred?

No

End

Yes

Diagnose the FAT system of the drive (☞ Page 16-91, Section 16.6).

Restore the FAT file structure of the drive (☞ Page 16-91, Section 16.6).

1)

Figure 16.9 When a file system error occurred

* 1 For actions by event No., refer to the following.

☞ C Controller Module User's Manual (Utility Operation, Programming)

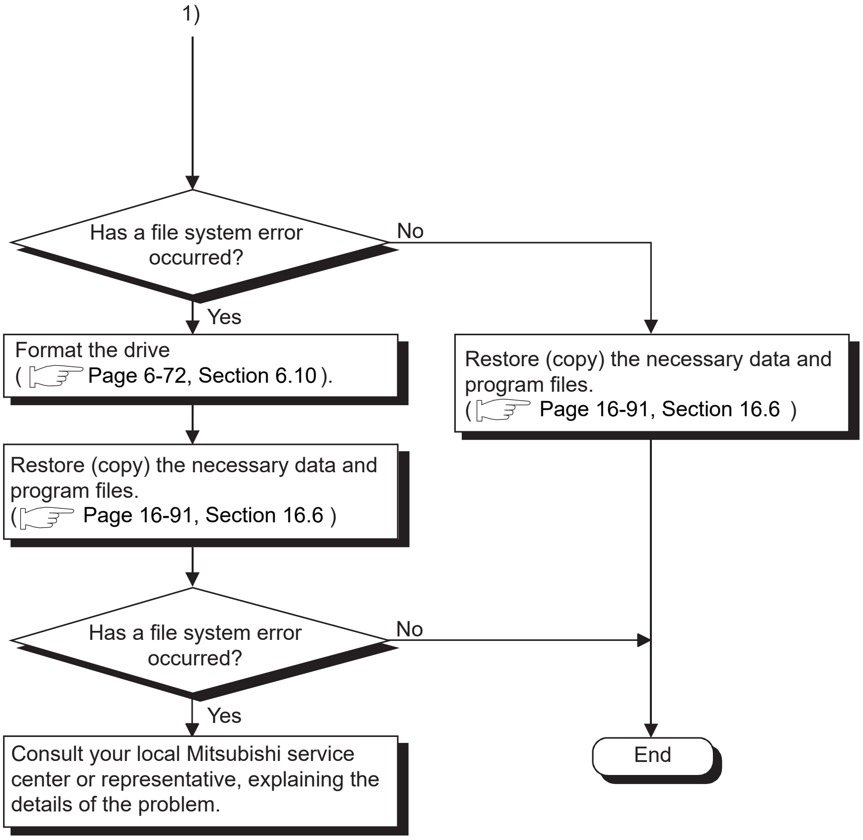

1)
Has a file system error occurred?
No
Yes
Format the drive (Page 6-72, Section 6.10).
Restore (copy) the necessary data and program files. (Page 16-91, Section 16.6)
Restore (copy) the necessary data and program files. (Page 16-91, Section 16.6)
Has a file system error occurred?
No
Yes
Consult your local Mitsubishi service center or representative, explaining the details of the problem.
End

Figure 16.9 When a file system error occurred (continued)

16.2.11 When no LED on an output module turn on

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	○

The following flowchart shows the procedures to be taken when no LED on an output module turn on during C Controller system operation.

☒POINT

Before using the following flowchart, check if the RUN LED on the C Controller module is on.
If the RUN LED is off, set the C Controller module to RUN to turn on the RUN LED.

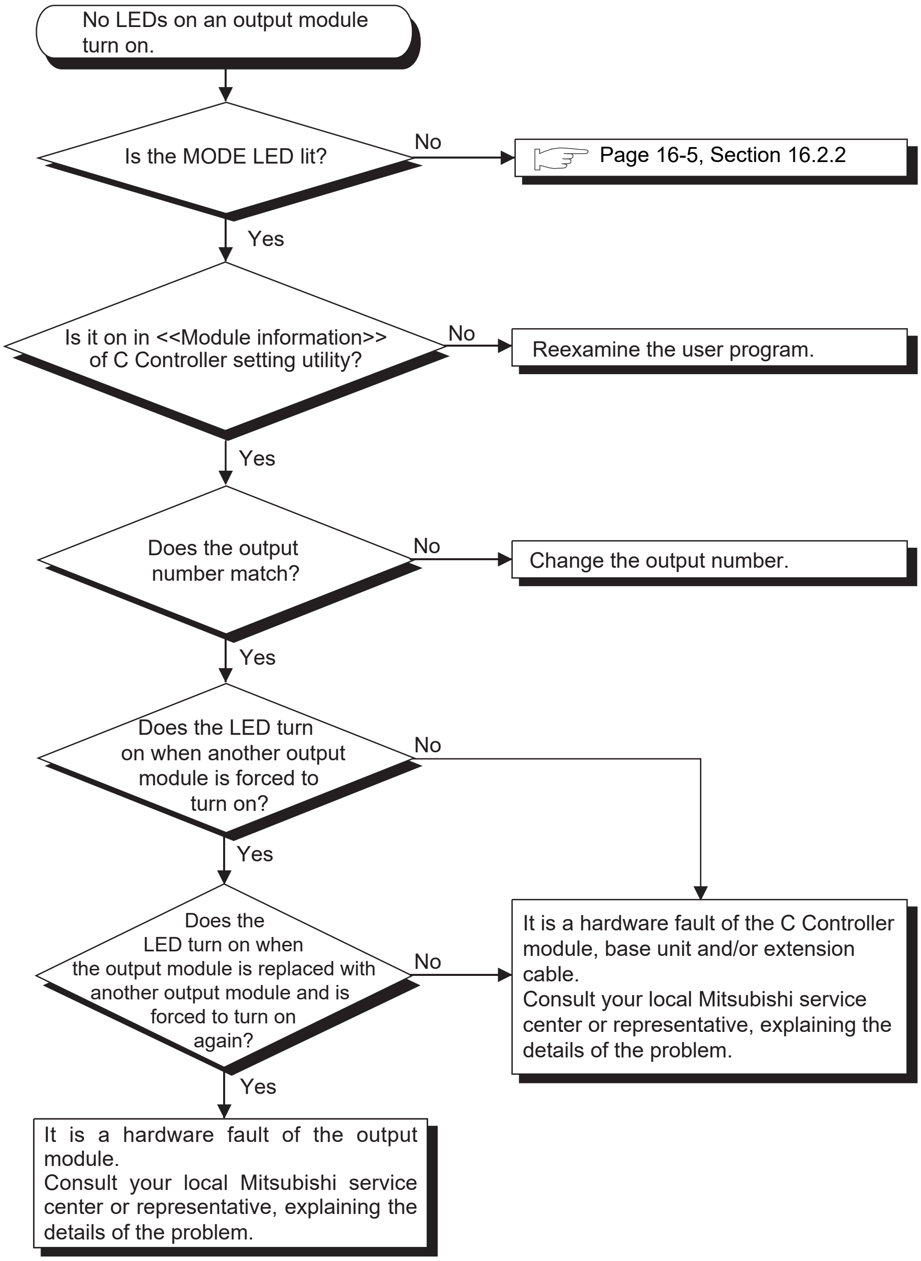

No LEDs on an output module turn on.
Is the MODE LED lit?
No
Page 16-5, Section 16.2.2
Yes
Is it on in <<Module information>> of C Controller setting utility?
No
Reexamine the user program.
Yes
Does the output number match?
No
Change the output number.
Yes
Does the LED turn on when another output module is forced to turn on?
No
Yes
Does the LED turn on when the output module is replaced with another output module and is forced to turn on again?
No
It is a hardware fault of the C Controller module, base unit and/or extension cable.
Consult your local Mitsubishi service center or representative, explaining the details of the problem.
Yes
It is a hardware fault of the output module.
Consult your local Mitsubishi service center or representative, explaining the details of the problem.

Figure 16.10 When no LED of an output module turn on

16.2.12 When an output load device of an output module does not turn on

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	○

The following flowchart shows the procedures to be taken when an output load device of an output module does not turn on during C Controller system operation.

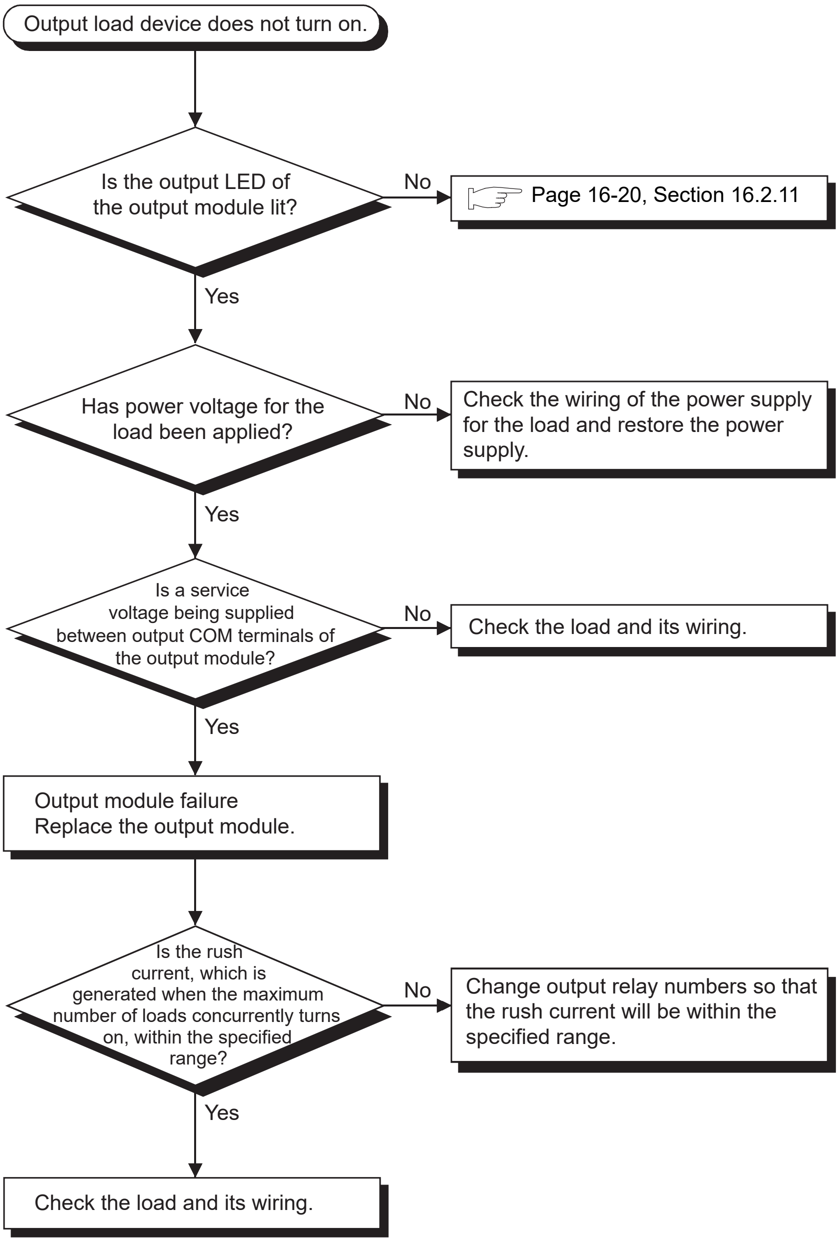

Output load device does not turn on.
Is the output LED of the output module lit?
No
Page 16-20, Section 16.2.11
Yes
Has power voltage for the load been applied?
No
Check the wiring of the power supply for the load and restore the power supply.
Yes
Is a service voltage being supplied between output COM terminals of the output module?
No
Check the load and its wiring.
Yes
Output module failure Replace the output module.
Is the rush current, which is generated when the maximum number of loads concurrently turns on, within the specified range?
No
Change output relay numbers so that the rush current will be within the specified range.
Yes
Check the load and its wiring.

Figure 16.11 When an output load device of an output module does not turn on

16.2.13 When operation is not normal due to script file execution

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	○

The following flowcharts show how to disable script file execution when the C Controller module does not operate normally due to execution of the user program registered to the script file.

(1) For the Q12DCCPU-V and Q06CCPU-V

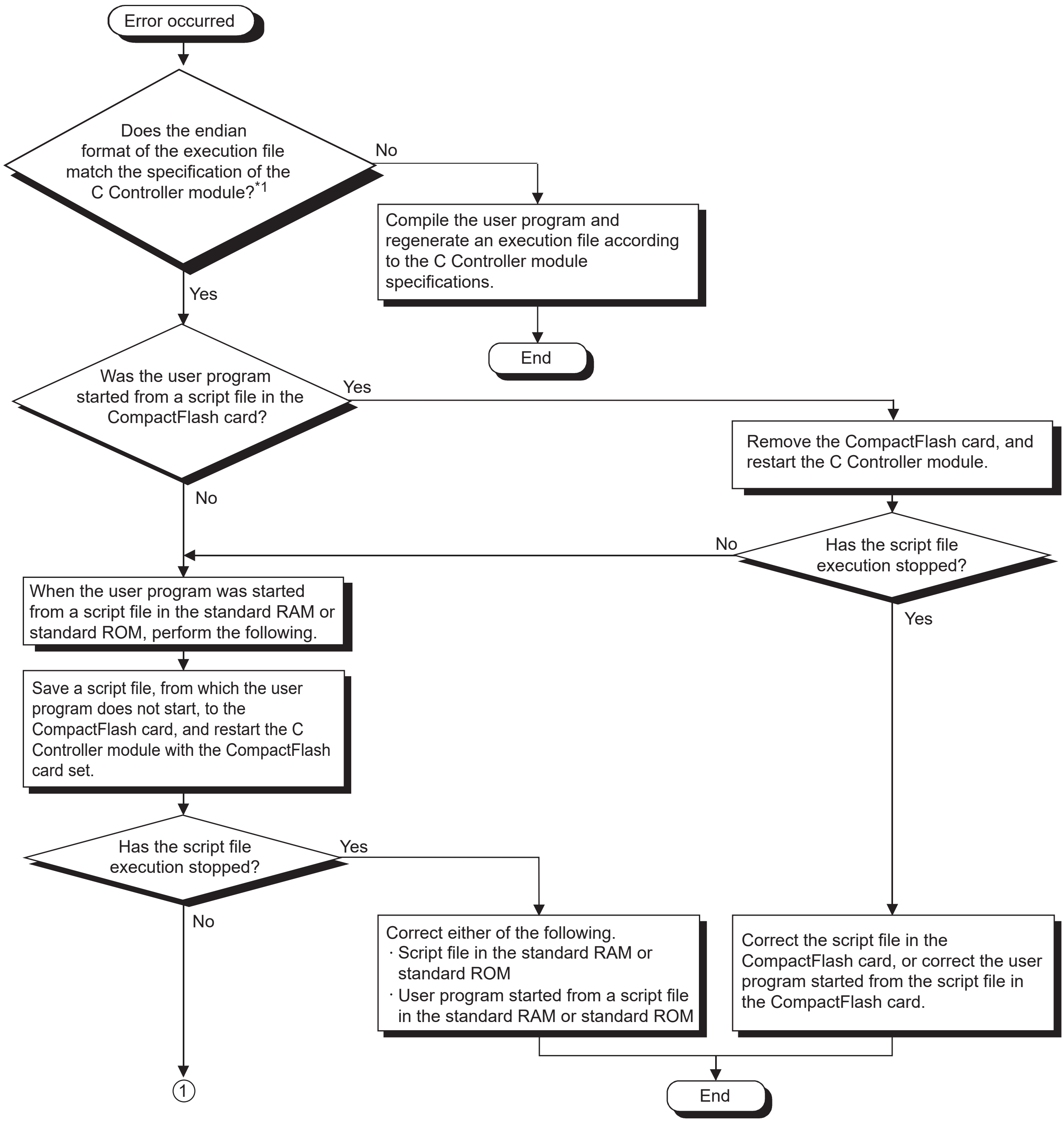

Error occurred
Does the endian format of the execution file match the specification of the C Controller module?*1
No
Compile the user program and regenerate an execution file according to the C Controller module specifications.
End
Yes
Was the user program started from a script file in the CompactFlash card?
Yes
Remove the CompactFlash card, and restart the C Controller module.
No
No
Has the script file execution stopped?
Yes
When the user program was started from a script file in the standard RAM or standard ROM, perform the following.
Save a script file, from which the user program does not start, to the CompactFlash card, and restart the C Controller module with the CompactFlash card set.
Has the script file execution stopped?
Yes
No
Correct either of the following.
· Script file in the standard RAM or standard ROM
· User program started from a script file in the standard RAM or standard ROM
Correct the script file in the CompactFlash card, or correct the user program started from the script file in the CompactFlash card.
1
End

Figure 16.12 Disabling script file execution (for the Q12DCCPU-V and Q06CCPU-V)

* 1 For restrictions on the functions, refer to the following.

☞ C Controller Module User's Manual (Utility Operation, Programming)

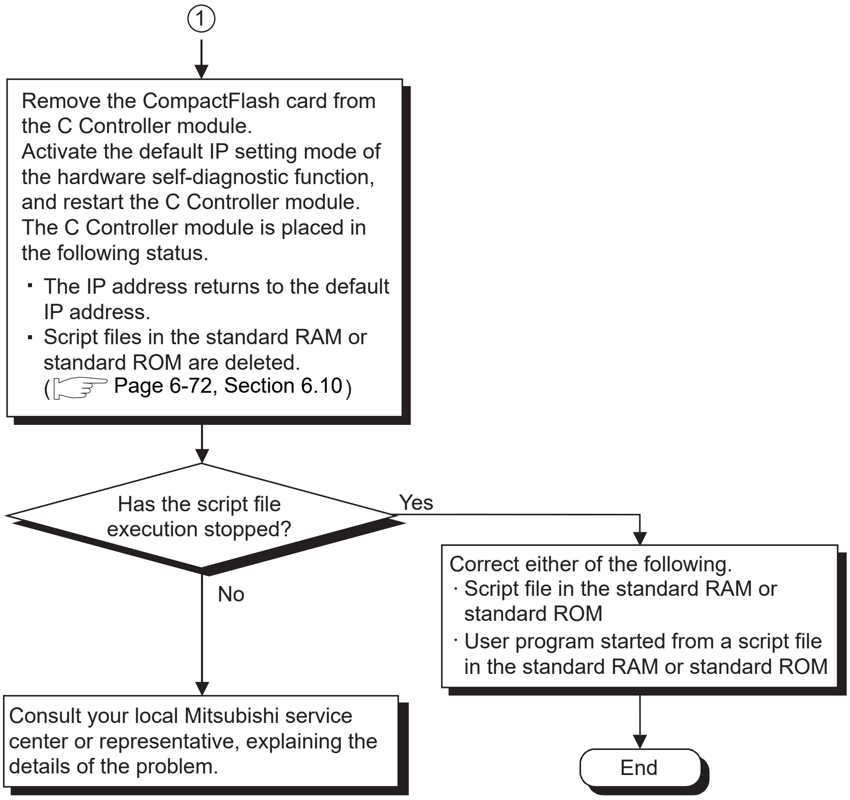

1
Remove the CompactFlash card from the C Controller module.
Activate the default IP setting mode of the hardware self-diagnostic function, and restart the C Controller module.
The C Controller module is placed in the following status.
· The IP address returns to the default IP address.
· Script files in the standard RAM or standard ROM are deleted.
(Page 6-72, Section 6.10)
Has the script file execution stopped?
Yes
No
Correct either of the following.
· Script file in the standard RAM or standard ROM
· User program started from a script file in the standard RAM or standard ROM
Consult your local Mitsubishi service center or representative, explaining the details of the problem.
End

Figure 16.12 Disabling script file execution (for the Q12DCCPU-V and Q06CCPU-V) (continued)

(2) For the Q06CCPU-V-B

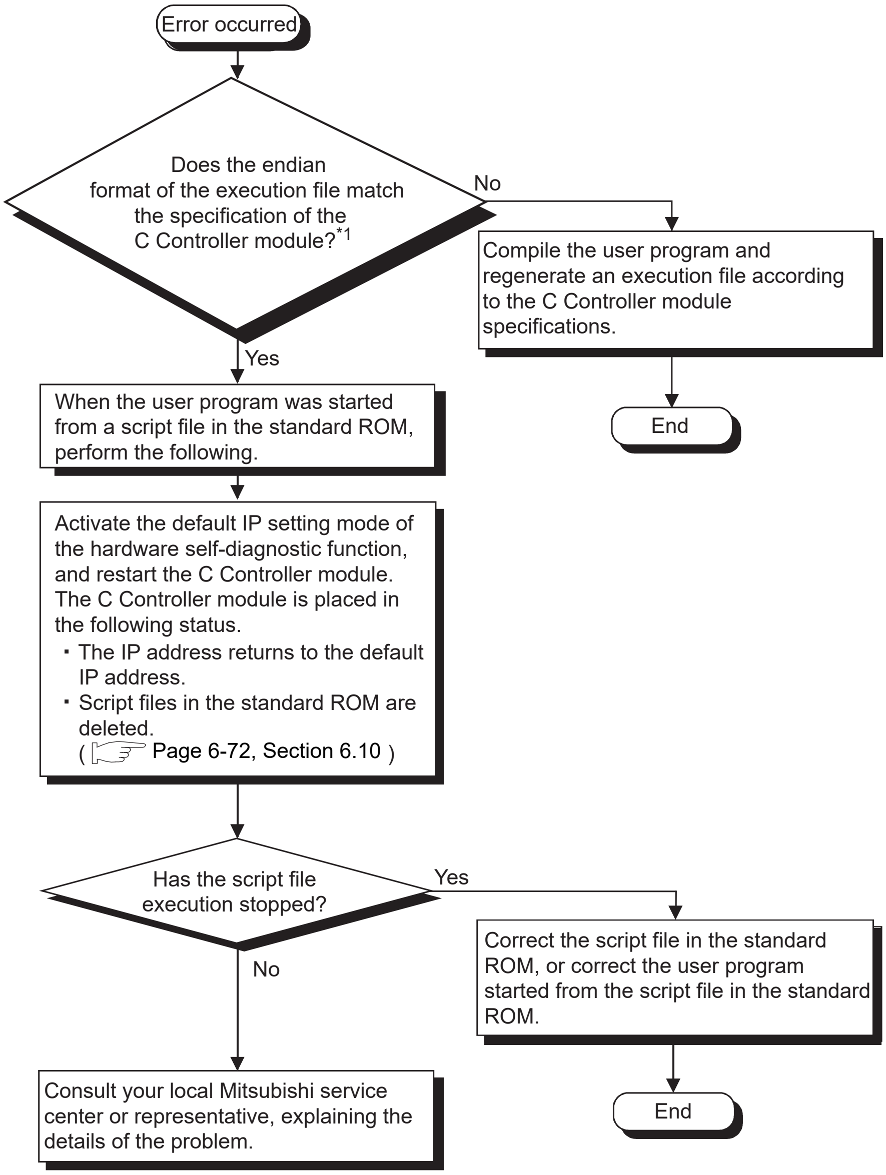

Error occurred
Does the endian format of the execution file match the specification of the C Controller module?*1
No
Compile the user program and regenerate an execution file according to the C Controller module specifications.
End
Yes
When the user program was started from a script file in the standard ROM, perform the following.
Activate the default IP setting mode of the hardware self-diagnostic function, and restart the C Controller module. The C Controller module is placed in the following status.
・The IP address returns to the default IP address.
・Script files in the standard ROM are deleted.
(Page 6-72, Section 6.10)
Has the script file execution stopped?
Yes
Correct the script file in the standard ROM, or correct the user program started from the script file in the standard ROM.
End
No
Consult your local Mitsubishi service center or representative, explaining the details of the problem.

Figure 16.13 Disabling script file execution (for the Q06CCPU-V-B)

* 1 For restrictions on the functions, refer to the following.

C Controller Module User's Manual (Utility Operation, Programming)

16.2.14 When an error occurred during user program download or ld command execution

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	○

If the endian format (memory layout) of the user program is different from that of the C Controller module, an error will occur in the following operation:

- Downloading a user program to a C Controller module
- Executing a user program with the ld command

An error message is displayed as shown below when the endian format does not match. If any of the shown messages appears, take corrective actions referring to (2) in this section.

(1) Error messages indicating the endian format mismatch between the user program and the C Controller module

(a) For the Q12DCCPU-V

1) When downloading a user program to a C Controller module

```
Failed to load "..xxx/SH7750gnu/Demo1.out on "yyyyyyyy@zzzzz:SH-4A"
```

Figure 16.14 Error message

2) When executing a user program with the ld command

```
ld(1,0,"/CF/Demo1.out")
ld(): error loading file (errno = 0x610001).
value = 0 = 0x0
<errno detail>
printErrno 0x610001
errno = 0x610001 : S_loadElfLib_HDR_READ.
value = 0 = 0x0
```

Figure 16.15 Error message

(b) For the Q06CCPU-V(-B)

1) When downloading a user program to a C Controller module

```
Error: This is not an elf module for the SH architecture.
Error: Object module load failed for C:/xxx/Demo1.out
Removed Demo1.out          from target..........done
WTX Error 0x1002e (LOADER_UNKNOWN_OBJ_MODULE_FORMAT)
```

Figure 16.16 Error message

2) When executing a user program with the ld command

```
ld(1,0,"/ROM/Demo1.out")
Incorrect ELF header size: 13312
ld error: error reading file (errno = 0x3d0001).
value = 0 = 0x0
```

Figure 16.17 Error message

(2) Corrective actions

1) Check the endian format of the user program.
For the checking method, refer to the following.
☞ C Controller Module User's Manual (Utility Operation, Programming)

2) If the endian format of the user program is different from that of the C Controller module, recompile the user program so that the same endian format will be used.

16.2.15 When unable to read from or write to the specified device

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	○

The following flowchart shows the procedure to be taken when reading from or writing to the specified device cannot be performed with a MELSEC data link function.

☒**POINT**

Use an extended function when:

- Specifying device No.32768 (8000H) or higher.
- Specifying a size of 32768 (8000H) bytes or more.

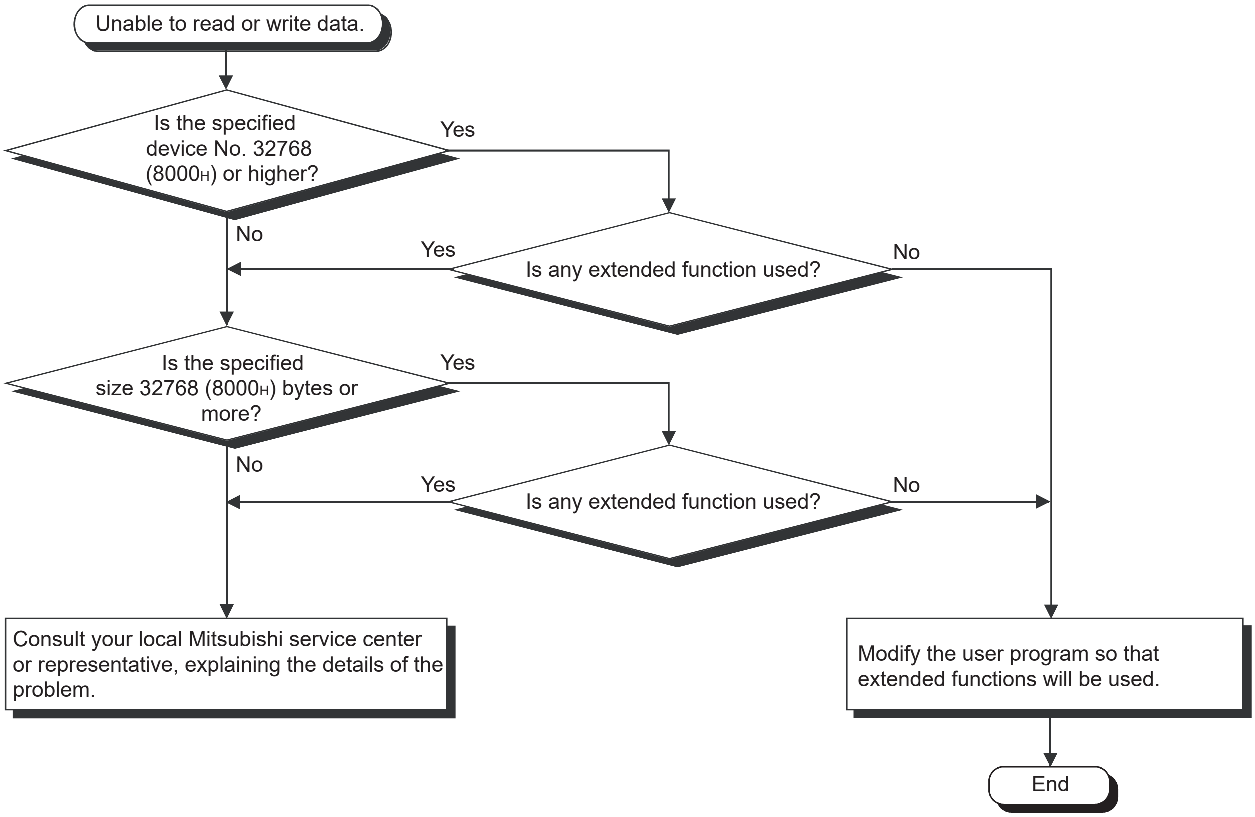

Unable to read or write data.
Is the specified device No. 32768 (8000H) or higher?
Yes
No
Is any extended function used?
Yes
No
Is the specified size 32768 (8000H) bytes or more?
Yes
No
Is any extended function used?
Yes
No
Consult your local Mitsubishi service center or representative, explaining the details of the problem.
Modify the user program so that extended functions will be used.
End

Figure 16.18 When unable to read from or write to the specified device

16.2.16 When an error occurred during reading from or writing to the standard RAM, standard ROM, CompactFlash card, or RAM disk using FTP

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	○

The following flowchart shows the procedures to be taken if the FTP tool of the development environment (personal computer) displays an error message while a file such as a user program file is being read from or written to the following memory:

- Standard RAM
- Standard ROM
- CompactFlash card
- RAM disk

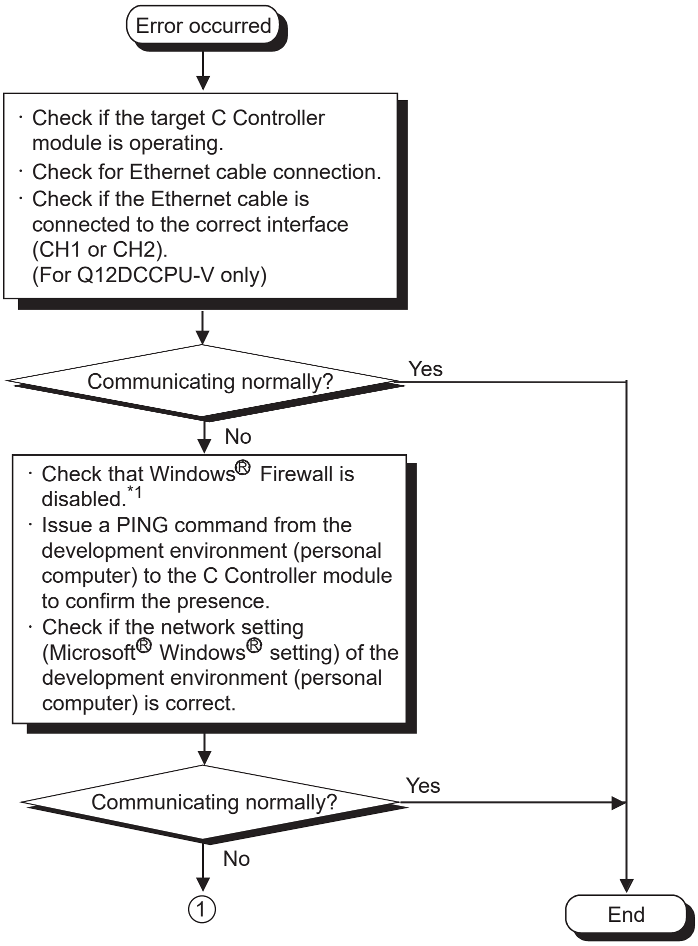

Error occurred
· Check if the target C Controller module is operating.
· Check for Ethernet cable connection.
· Check if the Ethernet cable is connected to the correct interface (CH1 or CH2). (For Q12DCCPU-V only)
Communicating normally?
Yes
No
· Check that Windows® Firewall is disabled.*1
· Issue a PING command from the development environment (personal computer) to the C Controller module to confirm the presence.
· Check if the network setting (Microsoft® Windows® setting) of the development environment (personal computer) is correct.
Communicating normally?
Yes
No
1
End

Figure 16.19 When an error occurred during reading from or writing to the standard RAM, standard ROM, CompactFlash card, or RAM disk using FTP

[1] For Windows® Firewall settings, refer to the following.

☞ C Controller Module User's Manual (Utility Operation, Programming)

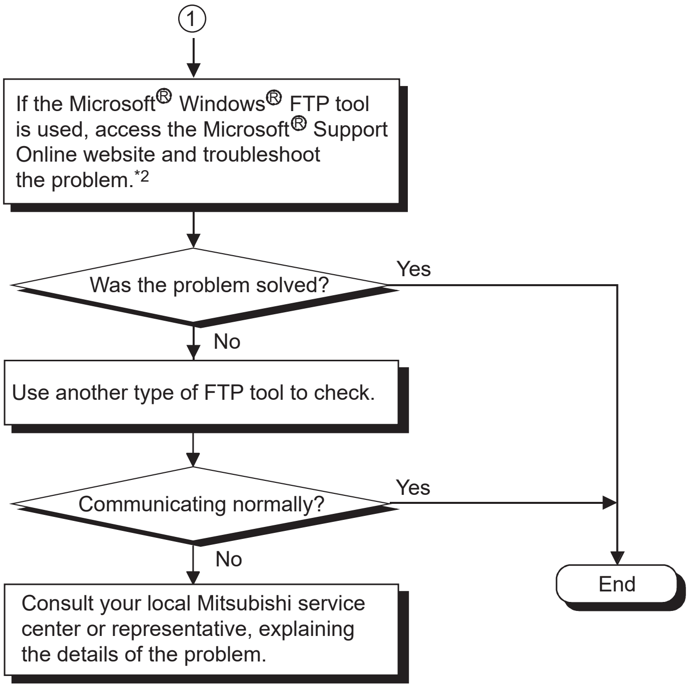

1
If the Microsoft® Windows® FTP tool is used, access the Microsoft® Support Online website and troubleshoot the problem.*2
Was the problem solved?
Yes
No
Use another type of FTP tool to check.
Communicating normally?
Yes
No
Consult your local Mitsubishi service center or representative, explaining the details of the problem.
End

Figure 16.19 When an error occurred during reading from or writing to the standard RAM, standard ROM, CompactFlash card, or RAM disk using FTP (continued)

* 2 For precautions for program registration by FTP, refer to the following.

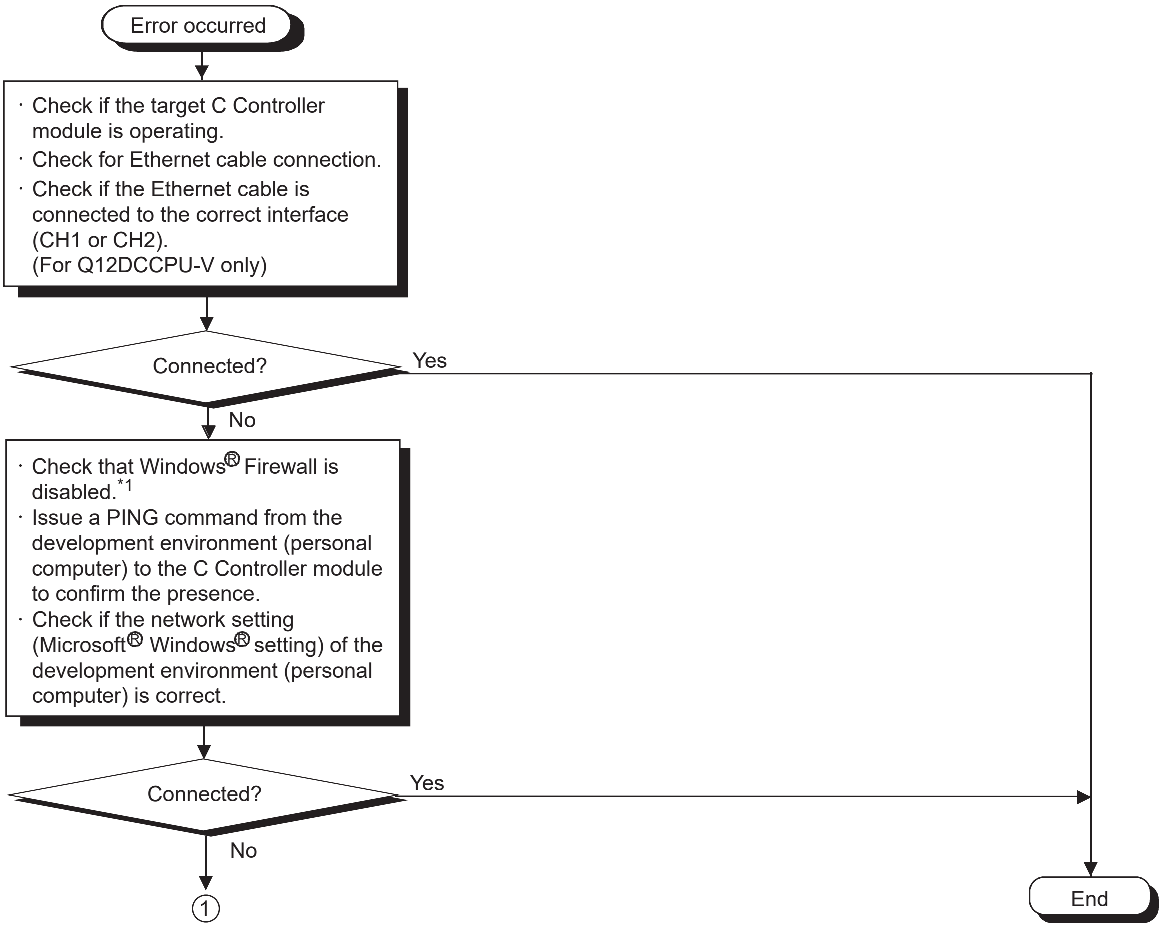
☞ C Controller Module User's Manual (Utility Operation, Programming)

9 DEVICE DESCRIPTION
10 MULTIPLE CPU SYSTEM OVERVIEW
11 MULTIPLE CPU SYSTEM CONFIGURATION
12 CONCEPT OF MULTIPLE CPU SYSTEM
13 COMMUNICATIONS BETWEEN CPU MODULES
14 PARAMETERS ADDED FOR MULTIPLE CPU SYSTEMS
15 STARTING A MULTIPLE CPU SYSTEM
16 TROUBLESHOOTING

16.2.17 When unable to make a Telnet connection

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	○

The following flowchart shows the procedures to be taken when Telnet connection cannot be made to the C Controller module from the command prompt on the development environment (personal computer).



· Check if the target C Controller module is operating.
· Check for Ethernet cable connection.
· Check if the Ethernet cable is connected to the correct interface (CH1 or CH2). (For Q12DCCPU-V only)

· Check that Windows® Firewall is disabled.*1
· Issue a PING command from the development environment (personal computer) to the C Controller module to confirm the presence.
· Check if the network setting (Microsoft® Windows® setting) of the development environment (personal computer) is correct.

Figure 16.20 When unable to make a Telnet connection

* 1 For Windows® Firewall settings, refer to the following.

☞ C Controller Module User's Manual (Utility Operation, Programming)

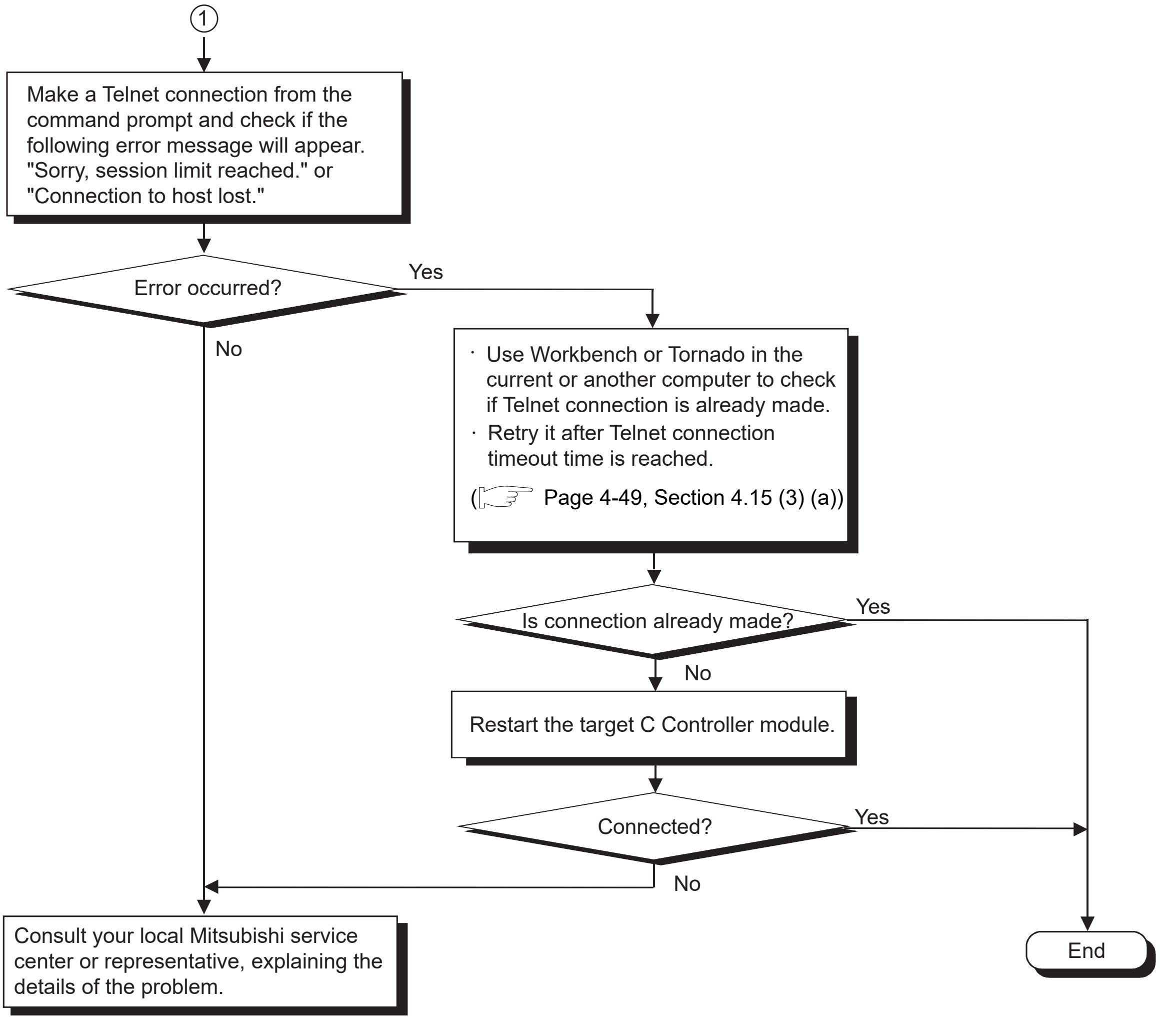

1
Make a Telnet connection from the command prompt and check if the following error message will appear. "Sorry, session limit reached." or "Connection to host lost."
Error occurred?
Yes
No
· Use Workbench or Tornado in the current or another computer to check if Telnet connection is already made.
· Retry it after Telnet connection timeout time is reached.
(Page 4-49, Section 4.15 (3) (a))
Is connection already made?
Yes
No
Restart the target C Controller module.
Connected?
Yes
No
Consult your local Mitsubishi service center or representative, explaining the details of the problem.
End

Figure 16.20 When unable to make a Telnet connection (continued)

16.2.18 When unable to make an FTP connection

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	○

The following flowchart shows the procedures to be taken when an FTP connection cannot be made to the C Controller module from the development environment (personal computer).

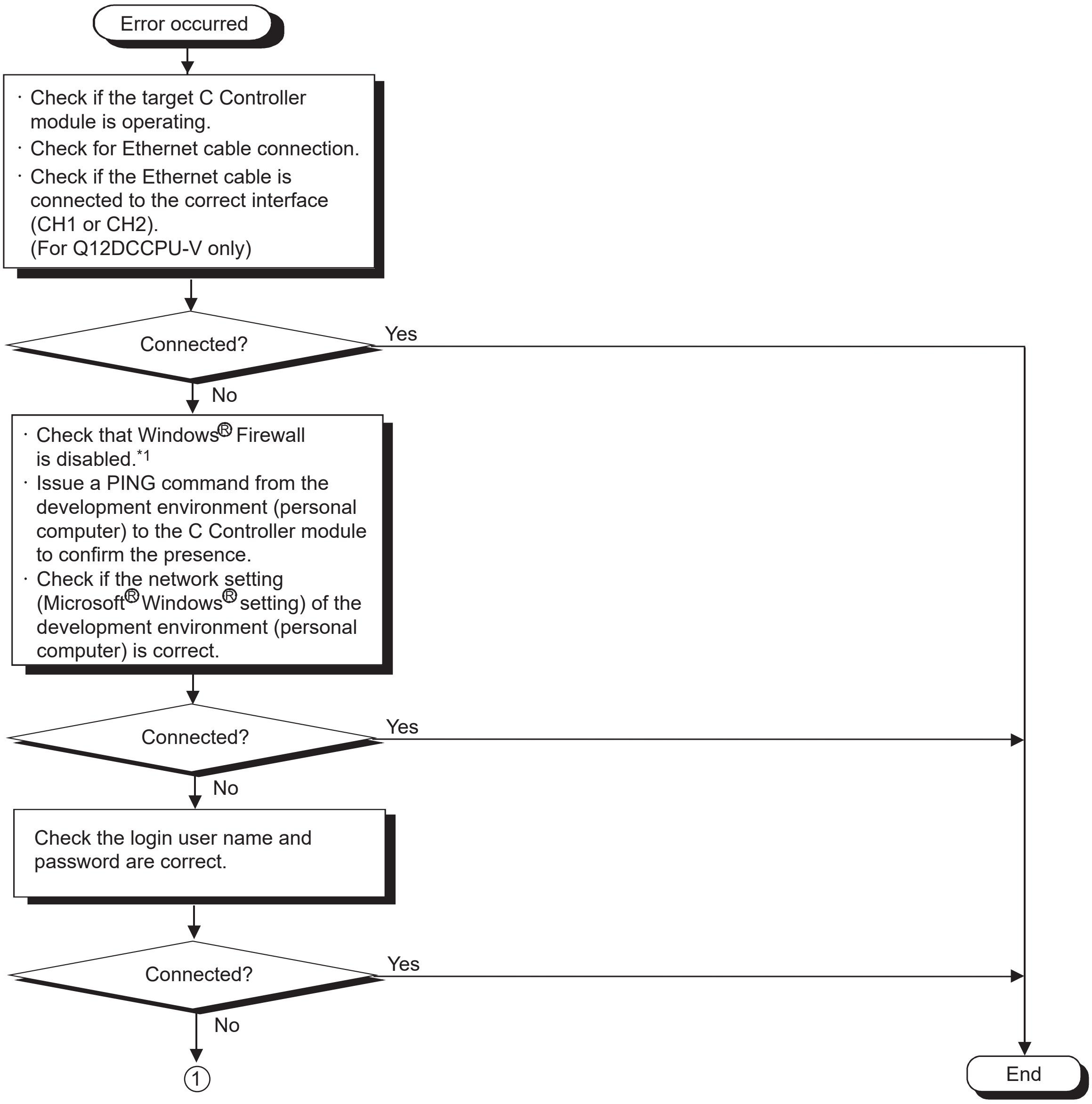

Error occurred
· Check if the target C Controller module is operating.
· Check for Ethernet cable connection.
· Check if the Ethernet cable is connected to the correct interface (CH1 or CH2). (For Q12DCCPU-V only)
Connected?
Yes
No
· Check that Windows® Firewall is disabled.*1
· Issue a PING command from the development environment (personal computer) to the C Controller module to confirm the presence.
· Check if the network setting (Microsoft® Windows® setting) of the development environment (personal computer) is correct.
Connected?
Yes
No
Check the login user name and password are correct.
Connected?
Yes
No
1
End

Figure 16.21 When unable to make an FTP connection

* 1 For Windows® Firewall settings, refer to the following.

☞ C Controller Module User's Manual (Utility Operation, Programming)

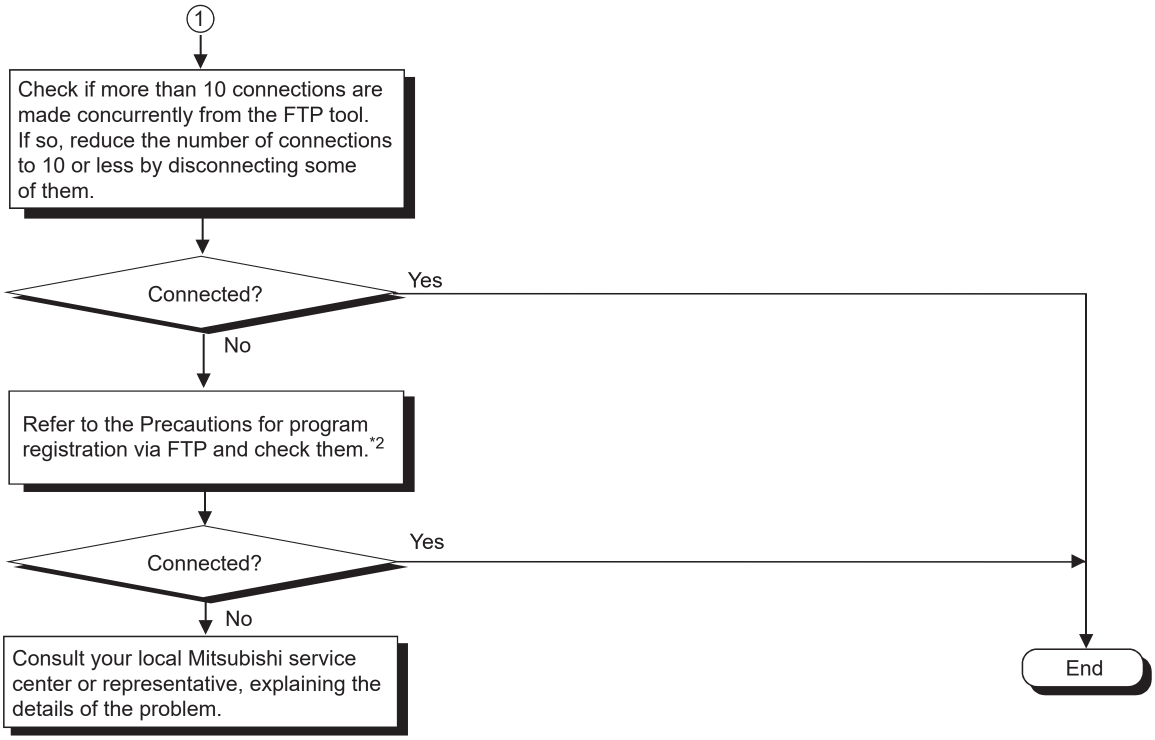

1
Check if more than 10 connections are made concurrently from the FTP tool. If so, reduce the number of connections to 10 or less by disconnecting some of them.
Connected?
Yes
No
Refer to the Precautions for program registration via FTP and check them.*2
Connected?
Yes
No
Consult your local Mitsubishi service center or representative, explaining the details of the problem.
End

Figure 16.21 When unable to make an FTP connection (continued)

* 2 For precautions for program registration by FTP, refer to the following.

☞ C Controller Module User's Manual (Utility Operation, Programming)

16.3 Actions to be Taken When the ERR. LED is On or Flashing

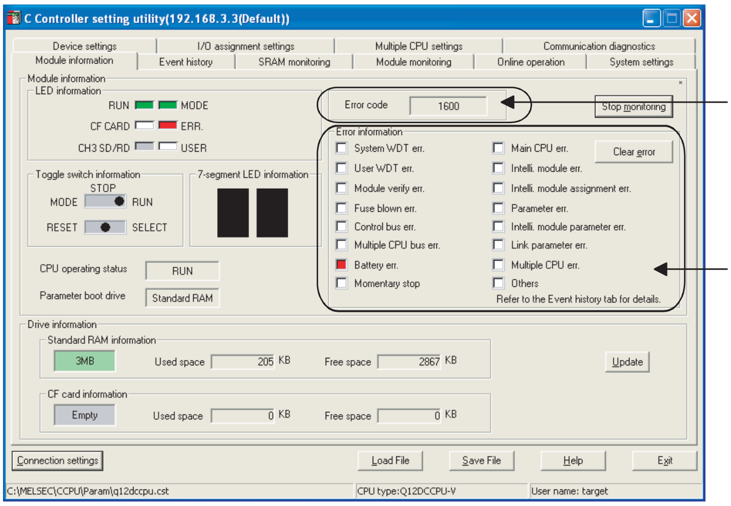
Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	○

If a stop or continuation error occurs in the C Controller module, the color of the corresponding error box changes from □ (white) to 🟥 (red) in "Error information" in the <<Module information>> tab of C Controller setting utility, and an error code is displayed.
The following describes respective error details and how to handle each error when the corresponding box is changed from □ (white) to 🟥 (red).

(1) Checking the error information

When the ERR. LED is on or flashing, check the error information as follows:

1) Select C Controller setting utility from "MELSEC" in Windows Start.
2) Click the <<Module information>> tab.
3) Click the Start monitoring button.
4) An error code is displayed on the screen.



Error code: 1600

□ System WDT err.
□ User WDT err.
□ Module verify err.
□ Fuse blown err.
□ Control bus err.
□ Multiple CPU bus err.
🟥 Battery err.
□ Momentary stop
□ Main CPU err.
□ Intelli. module err.
□ Intelli. module assignment err.
□ Parameter err.
□ Intelli. module parameter err.
□ Link parameter err.
□ Multiple CPU err.
□ Others

Refer to the Event history tab for details.

CPU operating status: RUN
Parameter boot drive: Standard RAM

Standard RAM information: 3MB Used space 205 KB Free space 2867 KB
CF card information: Empty Used space 0 KB Free space 0 KB

C:\MELSEC\CCPU\Param\q12dccpu.cst | CPU type:Q12DCCPU-V | User name: target

An error code is displayed when a box corresponding to an error is changed from □ (white) to 🟥 (red).
The error code is updated constantly during monitoring.
For details of each error code, refer to (3) in this section.

If an error occurs, the corresponding box is changed from □ (white) to 🟥 (red).
For respective error items and details, refer to (2) in this section.

Figure 16.22 Checking the error information

☒POINT

1. Monitoring stops when the active tab is switched from to another during monitoring, and resumes when the <<Module information>> is opened again.
2. If a communication error occurs during access, click the Start monitoring button or specify the connection target again.
3. Communication is not available while the C Controller module is being reset. Start communication after completion of the reset process.
4. The error history and detailed information can be checked in the <<Event history>> tab of C Controller setting utility.
5. An error code is displayed only when the error is a stop or continuation error. The displayed error is the latest error.
   The oldest error can be confirmed in the System information dialog box opened from the <<Module monitoring>> tab.

Remark

For C Controller setting utility, refer to the following.

☞ C Controller Module User's Manual (Utility Operation, Programming)

(2) **Checking the error information and corrective actions by error code**

The following describes how to handle each error when an error occurs and the corresponding box is changed from □ (white) to 🟥 (red).

(a) When several kinds of errors has occurred
In the System information dialog box opened from the <<Module monitoring>> tab of C Controller setting utility, check the information of the stop or continuation error that occurred first (Error code, Present error, Date, and Time).

(b) When an error has occurred in the C Controller module (host CPU)
Take either of the following actions.

- Open the System information dialog box from the <<Module monitoring>> tab of C Controller setting utility, and take corrective action according to Help information displayed.
  Open the Error details dialog box to check the detailed error information.
- In the <<Event history>> tab of C Controller setting utility, check the date and time of each error, and confirm the event information (the error code, error information, etc.) of the first error.
  When several stop and/or continuation errors have occurred, ignore the errors other than the first one.

(c) Error details and corrective actions

Identify the relevant error code in Table 16.2, and refer to the following.

☞ (3) Detailed error information and corrective actions by error code in this section

Table 16.2 Self-diagnostics and error codes

Error item	Self-diagnostic item	Description	Error code
System WDT err.	System watchdog timer error detection	A system watchdog timer error was detected.	5000
User WDT err.	User watchdog timer error detection	A user watchdog timer error was detected.	5001
Module verify err.	Module verify error detection	An error occurred during diagnosis of the I/O module installation.	2000
Fuse blown err.	Blown fuse detection	An error occurred during diagnosis of the output module's fuse.	1300
Control bus err.	Control bus error detection	An error occurred during diagnosis of the bus on the base unit.	1412 1413 1414 1415 1416
Multiple CPU bus err.	Multiple CPU high speed transmission error detection	An error was detected in multiple CPU high speed transmission.	1430 1431 1432 1433 1434 1435 1436 1437 1439
Battery err.	Battery error detection	A battery error of the C Controller module was detected.	1600
Momentary stop	Momentary stop detection	A momentary stop was detected.	1500
Main CPU err.	Main CPU error detection	Any of the following was detected. • Power supply module failure • C Controller module failure • Standard RAM data error	1000 1009 1106
Intelli. module err.	Intelligent function module error detection	An intelligent function module error was detected.	1310 1401 1403

(To the next page)

Table 16.2 Self-diagnostics and error codes (continued)

Error item	Self-diagnostic item	Description	Error code
Intelli. module assignment err.	Intelligent function module assignment error detection	An error occurred during diagnosis of the intelligent function module I/O assignment.	2100 2103 2106 2107 2108 2120 2121 2122 2124 2125 2126 2150
Parameter err.	Parameter error detection	An error occurred during diagnosis of parameters such as system settings, multiple CPU settings, etc.	2200 3000 3001 3010 3012 3013 3014
		An error occurred during diagnosis of multiple CPU parameters.	3015 3016
Intelli. module parameter err.	Intelligent function module parameter error detection	An error occurred during diagnosis of the intelligent function module parameters.	3300 3301 3302 3303
Link parameter err.	Link parameter error detection	An error occurred during parameter diagnosis of any of the following network modules. • CC-Link IE controller network module • MELSECNET/H module • CC-Link module	3100 3101 3102 3103 3104 3105 3107
Multi CPU err.	Multi CPU error detection	In a multiple CPU system configuration, an error of another CPU or a multiple CPU system error was detected.	7000 7002 7010 7011 7013 7020 7036

(To the next page)

Table 16.2 Self-diagnostics and error codes (continued)

Error item	Self-diagnostic item	Description	Error code
Others	Another error detection	A stop or continuation error other than the above was detected.	1510 1520 2502 5012

Remark

For the self-diagnostic item details, refer to the following.

☞ Page 4-20, Section 4.5

(3) Detailed error information and corrective actions by error code

The error codes and respective detailed information and corrective actions are shown below.

If an error code not found in the error code list is detected, please consult your local Mitsubishi service center or representative, explaining the details of the problem.

(a) Error code list (1000 to 1999)

Table 16.3 Error codes

Error code	Error message	Error information	LED status		CPU operation status	
			RUN	ERR.		
1000	MAIN CPU DOWN	—	Off	Flashing	Stop	
1009	MAIN CPU DOWN	—	Off	Flashing	Stop	
1106	RAM ERROR	—	Off	Flashing	Stop	
1300	FUSE BREAK OFF	Module No.*1	Off/On	Flashing/On	Stop/Continue*2	
1310	I/O INT.ERROR	—	Off	Flashing	Stop	

* 1 The error information (module No.) in "Event information" in the <<Event history>> tab of C Controller setting utility is displayed as follows:
"Error information: XX - YYY"
XX (DEC.): Slot No.(0 to 63), CPU No.(1 to 4)
YYY (HEX.): Start I/O No.÷16 (000 to 0FF, 3E0 to 3E3)

Error code	Error details and cause	Corrective action
1000	CPU module drives recklessly or it breaks down. • Malfunction by noise etc. • Hardware error.	• Take a preventive measure against noise. • Reset the CPU module and restart it. If the same error is displayed again, a hardware failure may have occurred in the CPU module. Please consult your local Mitsubishi service center or representative, explaining the details of the problem.
1009	A failure was detected in the power supply module, C Controller module, main base unit, extension base unit, or extension cable.	Reset the C Controller module and set it to RUN again. If the same error is displayed again, the power supply module, CPU module, main base unit, extension base unit, or extension cable is faulty. Replace the faulty module, or consult your local Mitsubishi service center or representative, explaining the details of the problem.
1106	Data error of standard RAM was detected.	• In the hardware self-diagnostic test mode, execute the battery-backed-up RAM diagnostic test (mode 1). <When completed normally> After performing the module initialization setting procedure, restart the C Controller module. <When failed> The C Controller module may be faulty. Please consult your local Mitsubishi service center or representative, explaining the details of the problem. • Check the battery, and if it is dead, replace it. • Take a preventive measure against noise. • Perform the module initialization setting procedure to initialize the module. If the same error persists even after taking the above actions, consult your local Mitsubishi service center or representative, explaining the details of the problem.
1300	The fuse of an output module has been blown.	Check the ERR. LED of each output module, and replace the module whose ERR. LED is on. • Read the error information in C Controller setting utility, and replace the fuse of the output module that corresponds to the value (module No.) • If a GOT is connected to the main or extension base unit in a bus topology, check the extension cable connection and the GOT grounding status.
1310	An A series interrupt module is mounted in the position which set to Empty in the I/O assignment.	Check the I/O assignment and the actually mounted module.
	An interrupt occurred even though no interrupt module is mounted.	A hardware failure may have occurred in any of the mounted modules. Check the modules and replace a faulty module. Or consult your local Mitsubishi service center or representative, explaining the details of the problem.

(To the next page)

* 2 The CPU operation status after error occurrence can be set either to "Stop" or "Continue" in "Error time operation mode" in the <<System settings>> tab of C Controller setting utility. (The LED indication also changes accordingly.)

Table 16.3 Error codes (continued)

Error code	Error message	Error information	LED status		CPU operation status	
			RUN	ERR.		
1401	SP.UNIT DOWN	Module No.*1	Off/On	Flashing/On	Stop/Continue*3	
1403	SP.UNIT DOWN	Module No.*1	Off/On	Flashing/On	Stop/Continue*3	
1412	CONTROL-BUS.ERR.	Module No.*1	Off	Flashing	Stop	
1413	CONTROL-BUS.ERR.	—	Off	Flashing	Stop	
1414	CONTROL-BUS.ERR.	Module No.*1	Off	Flashing	Stop	
1415	CONTROL-BUS.ERR.	Base No.	Off	Flashing	Stop	
1416	CONTROL-BUS.ERR.	Module No.*1	Off	Flashing	Stop	
1430	MULTI-C.BUS.ERR.	—	Off	Flashing	Stop	
1431	MULTI-C.BUS.ERR.	Module No.*1	Off	Flashing	Stop	

* 1 The error information (module No.) in "Event information" in the <<Event history>> tab of C Controller setting utility is displayed as follows:
"Error information: XX - YYY"
XX (DEC.): Slot No.(0 to 63), CPU No.(1 to 4)
YYY (HEX.): Start I/O No.÷16 (000 to 0FF, 3E0 to 3E3)

	Error code	Error details and cause	Corrective action
	1401	• In the initial processing, no response returns from an intelligent function module. • The buffer memory size of an intelligent function module is incorrect. • An unsupported module is mounted.	When unsupported module is mounted, remove the module. If the module is supported, a hardware failure may have occurred in the C Controller module, base unit, or the access target intelligent function module. Please consult your local Mitsubishi service center or representative, explaining the details of the problem.
	1403	• An unsupported module is mounted. • An intelligent function module error was detected. (Handshake error) • An I/O module (including intelligent function modules and special function modules) is being removed, was removed. or mounted during operation.	When unsupported module is mounted, remove the module. If the module is supported, a hardware failure may have occurred in the C Controller module, base unit, or the access target intelligent function module. Please consult your local Mitsubishi service center or representative, explaining the details of the problem.
	1412	The FROM/TO instruction cannot be executed due to a control bus error with an intelligent function module.	Reset the C Controller module and set it to RUN again. If the same error is displayed again, an intelligent function module, CPU module, or base unit is faulty. Please consult your local Mitsubishi service center or representative, explaining the details of the problem.
	1413	In the multiple CPU system, a CPU module incompatible with the multiple CPU system is mounted.	• Remove the incompatible CPU module from the main base unit. Or, replace the incompatible CPU module with a compatible one. Alternatively, mount the CPU modules in the CPU slot and slot 0 to 2. • An intelligent function module, CPU module, or base unit is faulty. Please consult your local Mitsubishi service center or representative, explaining the details of the problem.
		An error was detected on the system bus. • Self-diagnostic error of the system bus • Self-diagnostic error of the CPU module	• Reset the C Controller module and set it to RUN again. If the same error is displayed again, an intelligent function module, CPU module, or base unit is faulty. Please consult your local Mitsubishi service center or representative, explaining the details of the problem.
	1414	• Failure of a mounted module was detected. • In the multiple CPU system, a CPU module incompatible with the multiple CPU system is mounted.	• Remove the incompatible CPU module from the main base unit. Or, replace the incompatible CPU module with a compatible one. • Reset the C Controller module and set it to RUN again. If the same error is displayed again, an intelligent function module, CPU module, or base unit is faulty. Please consult your local Mitsubishi service center or representative, explaining the details of the problem.
	1415	Failure of the main or extension base unit was detected.	Reset the CPU module and set it to RUN again. If the same error is displayed again, an intelligent function module, CPU module, or base unit is faulty. Please consult your local Mitsubishi service center or representative, explaining the details of the problem.
	1416	A bus error was detected when power is turned on or the CPU module is reset.	Reset the CPU module and set it to RUN again. If the same error is displayed again, an intelligent function module, CPU module, or base unit is faulty. Please consult your local Mitsubishi service center or representative, explaining the details of the problem.
	1430	The error of connected CPU was detected at the multiple CPU high speed transmission.	Reset the CPU module and set it to RUN again. If the same error is displayed again, a hardware failure may have occurred in the CPU module. Please consult your local Mitsubishi service center or representative, explaining the details of the problem.
	1431	Error (Disconnected) with other CPU was detected at the multiple CPU high speed transmission.	• Take a preventive measure against noise. • Check if the CPU module is mounted on the main base unit correctly. • Reset the CPU module and set it to RUN again. If the same error is displayed again, a hardware failure may have occurred in the CPU module. Please consult your local Mitsubishi service center or representative, explaining the details of the problem.

(To the next page)

* 3 Either of "Stop" or "Continue" can be set for each module in "H/W error time CPU operation mode" in the <<I/O assignment settings>> tab of C Controller setting utility.

Table 16.3 Error codes (continued)

Error code	Error message	Error information	LED status		CPU operation status	
			RUN	ERR.		
1432	MULTI-C.BUS.ERR.	Module No.*1	Off	Flashing	Stop	
1433	MULTI-C.BUS.ERR.	Module No.*1	Off	Flashing	Stop	
1434	MULTI-C.BUS.ERR.	Module No.*1	Off	Flashing	Stop	
1435	MULTI-C.BUS.ERR.	Module No.*1	Off	Flashing	Stop	
1436	MULTI-C.BUS.ERR.	—	Off	Flashing	Stop	
1437	MULTI-C.BUS.ERR.	—	Off	Flashing	Stop	
1439	MULTI-C.BUS.ERR.	—	Off	Flashing	Stop	
1500	AC/DC DOWN	—	On	Off	Continue	
1510	SINGLE PS.DOWN	Base No./Power supply No.	On	On	Continue	
1520	SINGLE PS.ERROR	Base No./Power supply No.	On	On	Continue	
1600	BATTERY ERROR	—	On	On	Continue	

* 1 The error information (module No.) in "Event information" in the <<Event history>> tab of C Controller setting utility is displayed as follows:
"Error information: XX - YYY"
XX (DEC.): Slot No.(0 to 63), CPU No.(1 to 4)
YYY (HEX.): Start I/O No.÷16 (000 to 0FF, 3E0 to 3E3)

	Error code	Error details and cause	Corrective action
	1432	Communication timeout with other CPU was detected at the multiple CPU high speed transmission.	Reset the CPU module and set it to RUN again. If the same error is displayed again, a hardware failure may have occurred in the CPU module. Please consult your local Mitsubishi service center or representative, explaining the details of the problem.
	1433	Error (Disconnected) with other CPU was detected at the multiple CPU high speed transmission.	• Take a preventive measure against noise. • Check if the CPU module is mounted on the main base unit correctly. • Reset the CPU module and set it to RUN again. If the same error is displayed again, a hardware failure may have occurred in the CPU module. Please consult your local Mitsubishi service center or representative, explaining the details of the problem.
	1434	Error (Disconnected) with other CPU was detected at the multiple CPU high speed transmission.	• Take a preventive measure against noise. • Check if the CPU module is mounted on the main base unit correctly. • Reset the CPU module and set it to RUN again. If the same error is displayed again, a hardware failure may have occurred in the CPU module. Please consult your local Mitsubishi service center or representative, explaining the details of the problem.
	1435	Error (Disconnected) with other CPU was detected at the multiple CPU high speed transmission.	• Take a preventive measure against noise. • Check if the CPU module is mounted on the main base unit correctly. • Reset the CPU module and set it to RUN again. If the same error is displayed again, a hardware failure may have occurred in the CPU module. Please consult your local Mitsubishi service center or representative, explaining the details of the problem.
	1436	The error of the multiple CPU high speed transmission main base unit was detected at the multiple CPU high speed transmission (The error of the multiple CPU high speed transmission was detected).	Reset the CPU module and set it to RUN again. If the same error is displayed again, a hardware failure may have occurred in the CPU module. Please consult your local Mitsubishi service center or representative, explaining the details of the problem.
	1437	The error of the multiple CPU high speed transmission main base unit was detected at the multiple CPU high speed transmission (The error of the multiple CPU high speed transmission was detected).	• Take a preventive measure against noise. • Check if the CPU module is mounted on the main base unit correctly. • Reset the CPU module and set it to RUN again. If the same error is displayed again, a hardware failure may have occurred in the CPU module. Please consult your local Mitsubishi service center or representative, explaining the details of the problem.
	1439	The error of the multiple CPU high speed transmission main base unit was detected at the multiple CPU high speed transmission (The error of the multiple CPU high speed transmission was detected).	Reset the CPU module and set it to RUN again. If the same error is displayed again, a hardware failure may have occurred in the CPU module. Please consult your local Mitsubishi service center or representative, explaining the details of the problem.
	1500	• A momentary power failure occurred in the power supply. (The number of detected momentary power failures can be confirmed by the QBF_ReadStatusEx function.) • The power supply was turned off.	Check the power supply.
	1510	On the redundant power supply base unit, the voltage of one of the power supplies dropped.	Check the supply power to the redundant power supply module on the redundant power supply base unit.
	1520	In the redundant power supply system, a failure was detected in one of the redundant power supply modules.	A hardware failure may have occurred in the redundant power supply module. Please consult your local Mitsubishi service center or representative, explaining the details of the problem.
	1600	• The battery voltage of the CPU module has dropped to the specified value or less. • The lead connector of the battery in the CPU module is not connected. • The lead connector of the CPU module battery is not securely engaged.	• Replace the battery. • Connect the lead connector. • Check the lead connector of the CPU module for looseness. Firmly engage the connector if it is loose.

(b) Error code list (2000 to 2999)

Table 16.4 Error codes

Error code	Error message	Error information	LED status		CPU operation status	
			RUN	ERR.		
2000	UNIT VERIFY ERR.	Module No.[*1]	Off/ On	Flashing/On	Stop/Continue[*2]	
2100	SP.UNIT LAY ERR.	Module No.[*1]	Off	Flashing	Stop	
2103	SP.UNIT LAY ERR.	Module No.[*1]	Off	Flashing	Stop	
2106	SP.UNIT LAY ERR.	Module No.[*1]	Off	Flashing	Stop	
2107	SP.UNIT LAY ERR.	Module No.[*1]	Off	Flashing	Stop	
2108	SP.UNIT LAY ERR.	Module No.[*1]	Off	Flashing	Stop	
2120	SP.UNIT LAY ERR.	—	Off	Flashing	Stop	
2121	SP.UNIT LAY ERR.	—	Off	Flashing	Stop	
2122	SP.UNIT LAY ERR.	—	Off	Flashing	Stop	

[*1] The error information (module No.) in "Event information" in the <<Event history>> tab of C Controller setting utility is displayed as follows:
"Error information: XX - YYY"
XX (DEC.): Slot No.(0 to 63), CPU No.(1 to 4)
YYY (HEX.): Start I/O No.÷16 (000 to 0FF, 3E0 to 3E3)

Error code	Error details and cause	Corrective action
2000	In the multiple CPU system, a CPU module incompatible with the multiple CPU system is mounted.	Replace the incompatible CPU module with a compatible one.
	The setting is different from the I/O module information input at power-on. (The I/O module mounting status was changed during operation. Or, the module is almost disconnected.)	• Read the error information in C Controller setting utility, check the module corresponding to the value (module No.), and replace the module. • If a GOT is connected to the main or extension base unit in a bus topology, check the extension cable connection and the GOT grounding status.
2100	In the I/O assignment parameter setting, the slot of the QI60 is set to other than Intelli. (intelligent function module) and Interrupt (interrupt module).	Correct the I/O assignment parameter setting.
	• In the I/O assignment parameter setting, an I/O module is set as an intelligent function module, or vice versa. • In the I/O assignment parameter setting, a CPU module is set as another module type or empty, or vice versa. • In the I/O assignment parameter setting, the number of points set for an intelligent function module is less than the actual number of points. • In the I/O assignment parameter setting, the switches were set for the module than has no switch setting.	• Correct the I/O assignment parameters to match them to the actual mounting status of intelligent function modules and CPU modules. • In the I/O assignment parameter setting, remove the switch settings.
2103	• Two or more QI60 modules are mounted in the single CPU system. • Two or more QI60/A1SI61 modules are mounted for the same control CPU (programmable controller CPU only) in the multiple CPU system. • Two or more A1SI61 modules are mounted in the multiple CPU system.	• Use one QI60 in the single CPU system. • Change the number of QI60 modules set for the same control CPU to only one in the multiple CPU system. • Use only one A1SI61 in the multiple CPU system. If an interrupt module is used for each QCPU in the multiple CPU system, replace it with the QI60. (Use one A1SI61 and up to three QI60 modules, or the QI60 modules only.)
	Two or more QI60 modules with no interrupt pointer setting are mounted.	• Use only one QI60. • Configure the interrupt event settings for the QI60 modules other than the first one.
2106	A total of five or more MELSECNET/H modules and/or CC-Link IE controller network modules are mounted on the entire system.	Reduce the total number of MELSECNET/H modules and CC-Link IE controller network modules on the entire system to four or less.
	• Five or more MELSECNET/H modules are mounted. • Five or more CC-Link IE controller network modules are mounted. • Five or more Q series Ethernet modules are mounted.	• Reduce the number of MELSECNET/H modules to four or less. • Reduce the number of CC-Link IE controller network modules to four or less. • Reduce the number of Q series Ethernet modules to four or less.
	The same network No. or station No. is set for different networks or stations on the MELSECNET/H or CC-Link IE controller network system.	Check the network numbers or station numbers.
2107	The start X/Y set in the I/O assignment parameter setting is overlapped with the one set for another module.	Correct the I/O assignment parameters to match them to the actual mounting status of intelligent function modules.
2108	• A network module for the A2UCPU (A1SJ71LP21, A1SJ71BR11, A1SJ71LR21, A1SJ71AP21, A1SJ71AR21, or A1SJ71AT21B) is mounted. • A network module for the Q2AS (A1SJ71QLP21(S), A1SJ71QBR11, or A1SJ71QLR21) is mounted.	Replace the network module with a MELSECNET/H module.
2120	• In the single CPU system, the QA□B or QA1S□B base unit is used. • In a multiple CPU system configuration, a C Controller module is set as a CPU that controls modules on the QA□B or QA1S□B.	• Replace the base unit with the Q□B. • Set a High Performance model QCPU as a CPU that controls the modules on the QA□B or QA1S□B.
2121	A CPU module is installed to other than the CPU slot and slot 0 to 2.	Check the slot position of the C Controller module and install the CPU module in the correct slot.
2122	The QA1S□B is used as the main base unit.	Change the main base unit to the Q□B.

(To the next page)

* 2 The CPU operation status after error occurrence can be set either to "Stop" or "Continue" in "Error time operation mode" in the <<System settings>> tab of C Controller setting utility. (The LED indication also changes accordingly.)

Table 16.4 Error codes (continued)

Error code	Error message	Error information	LED status		CPU operation status	
			RUN	ERR.		
2124	SP.UNIT LAY ERR.	—	Off	Flashing	Stop	
2125	SP.UNIT LAY ERR.	Module No.[*1]	Off	Flashing	Stop	
2126	SP.UNIT LAY ERR.	Module No.[*1]	Off	Flashing	Stop	
2150	SP.UNIT VER.ERR.	Module No.[*1]	Off	Flashing	Stop	
2200	MISSING PARA.	—	Off	Flashing	Stop	
2502	CAN'T EXE.PRG.	—	Off	Flashing	Stop	

[*1] The error information (module No.) in "Event information" in the <<Event history>> tab of C Controller setting utility is displayed as follows:
"Error information: XX - YYY"
XX (DEC.): Slot No.(0 to 63), CPU No.(1 to 4)
YYY (HEX.): Start I/O No.÷16 (000 to 0FF, 3E0 to 3E3)

Error code	Error details and cause	Corrective action
2124	• A module is installed in the position of slot 65 or higher. • A module is installed in the position of the slot number that exceeds the number of slots set for the base unit. • A module is installed in the slot where I/O points of 4096 or higher are assigned. • A module is installed in the slot whose assigned I/O points cross over the boundary of 4096.	• Remove the module in slot 65 or higher. • Remove the module installed in the position of the slot number that exceeds the number of slots set for the base unit. • Remove the module installed in the slot where 4096 points or higher are assigned. • Replace the last module with a module whose I/O points do not exceed 4096.
2125	• A module that cannot be recognized is mounted. • No response returns from an intelligent function module.	• Mount applicable modules. • The intelligent function module is faulty. Please consult your local Mitsubishi service center or representative, explaining the details of the problem.
2126	In the multiple CPU system, CPU modules are configured as follows: • The slot on the left of a CPU module is empty. • A CPU module other than High Performance model QCPUs is mounted on the left of a High Performance model QCPU. • When CPU No.1 is a High Performance model QCPU, a CPU module other than C Controller modules is mounted on the right of a C Controller module. • When CPU No.1 is a C Controller module, a programmable controller CPU is mounted on the right of the C Controller module. • When CPU No.1 is a Basic model QCPU, a CPU module is mounted on the right of the C Controller module.	Perform the following. For CPU module positions in a multiple CPU system, refer to Page 12-1, Section 12.1. • Eliminate any empty slot space between CPU modules. (An empty slot must be set on the right side of CPU modules.) • Replace the module other than High Performance model QCPUs on the left of the High Performance model QCPU with a High Performance model QCPU. • Remove the CPU module other than C Controller modules on the right of the C Controller module. • Remove the programmable controller CPU mounted on the right of the C Controller module. • Remove the CPU module mounted on the right of the C Controller module.
2150	In the multiple CPU system, an intelligent function module incompatible with the multiple CPU system is set to be controlled by a CPU other than CPU No.1.	• Change the module to an intelligent function module that supports the multiple CPU system (function version B or later). • Change the control CPU setting for the intelligent function module to CPU No.1.
2200	• No parameter file is set. • The parameters were written to a drive different from a valid one.	• Set the parameter file. • Write the parameters to a valid drive.
2502	• One command (one line) in the script file has 129 or more characters. • The command in the script file cannot be executed. (An error in the syntax, or no command exists.)	Correct the description of the error script file as shown below, and restart the C Controller module. • Reduce the number of characters for one command (one line) in the script file to 128 or less. • Confirm no error in the syntax and the presence of a command.

(c) Error code list (3000 to 3999)

Table 16.5 Error codes

Error code	Error message	Error information	LED status		CPU operation status
			RUN	ERR.	
3000	PARAMETER ERROR	Parameter No.[*1]	Off	Flashing	Stop
3001	PARAMETER ERROR	Parameter No.[*1]	Off	Flashing	Stop
3010	PARAMETER ERROR	Parameter No.[*1]	Off	Flashing	Stop
3012	PARAMETER ERROR	Parameter No.[*1]	Off	Flashing	Stop
3013	PARAMETER ERROR	Parameter No.[*1]	Off	Flashing	Stop
3014	PARAMETER ERROR	Parameter No.[*1]	Off	Flashing	Stop
3015	PARAMETER ERROR	Parameter No.[*1]/CPU No.	Off	Flashing	Stop
3016	PARAMETER ERROR	Parameter No.[*1]/CPU No.	Off	Flashing	Stop

* 1 For the parameter setting location for each parameter number, refer to the following.

☞ Appendix 8

	Error code	Error details and cause	Corrective action
	3000	In the multiple CPU system, an intelligent function module controlled by another CPU is specified in the interrupt event setting of Parameter.	• In the interrupt event setting of Parameter, specify the start I/O No. of the intelligent function module controlled by the host CPU. • Delete the interrupt event setting of Parameter.
		In the parameter setting, the points for an empty slot are not set within the range applicable to the C Controller module.	• Read the error information in C Controller setting utility, check the parameter item corresponding to the value (parameter No.), and correct it. • rite the corrected parameters again to the C Controller module, and restart the C Controller system or reset the C Controller module. • If the same error occurs again, a hardware failure may have occurred. Please consult your local Mitsubishi service center or representative, explaining the details of the problem.
	3001	The parameter data are corrupted.	• Read the error information in C Controller setting utility, check the parameter item corresponding to the value (parameter No.), and correct it. • Write the corrected parameters again to the C Controller module, and restart the C Controller system or reset the C Controller module. • If the same error occurs again, a hardware failure may have occurred. Please consult your local Mitsubishi service center or representative, explaining the details of the problem.
	3010	In the multiple CPU system, the number of CPU modules set in the Parameter setting differs from the number of actually mounted CPU modules.	Correct the number of CPU modules set in the Multiple CPU settings (the number of CPUs - CPU (Empty) setting in I/O assignment) to match it to the actual mounting status.
	3012	In the multiple CPU system, the multiple CPU settings and control CPU settings written to a CPU are different from those of CPU No.1.	Correct the multiple CPU settings and control CPU settings to match them to those of CPU No.1.
	3013	Multiple CPU auto refresh setting was following on the multiple CPU system. • The number of the sending points is odd number. • The total of the sending points exceeds the maximum refresh point.	In the refresh setting of the multiple CPU setting, check the following and correct as follows: • Set the number of send points to an even number. • Change the setting so that the total number of send points does not exceed the maximum refresh points.
	3014	In s multiple CPU system configuration, the online module change parameter (multiple CPU system parameter) setting written to a CPU is different from that of CPU No.1.	Correct the online module change parameter (multiple CPU system parameter) setting written to the CPU to match it to that of CPU No.1.
		In s multiple CPU system configuration, the online module change is enabled even though the mounted CPU module does not support the online module change parameter.	Disable the online module change when the mounted CPU module does not support the online module change parameter.
	3015	When the multiple CPU system, it is different from CPU that the parameter setting verified.	Read individual error information with a peripheral device, check the parameter item corresponding to the value (parameter No./CPU No.) and the target CPU parameters, and correct them.
	3016	On the multiple CPU synchronous startup setting, the CPU module that is not supported the multiple CPU synchronous startup was set as the target of synchronous startup.	Remove the CPU from the multiple CPU synchronized boot-up setting, and correct the setting.

(To the next page)

Table 16.5 Error codes (continued)

Error code	Error message	Error information	LED status		CPU operation status	
			RUN	ERR.		
3100	LINK PARA.ERROR	Parameter No.*1	Off	Flashing	Stop	

* 1 For the parameter setting location for each parameter number, refer to the following.

☞ Appendix 8

Error code	Error details and cause	Corrective action
3100	In the multiple CPU system, a CC-Link IE controller network module controlled by another CPU is assigned to the start I/O No. in the CC-Link controller network parameter setting.	• Delete the network parameters for the CC-Link IE controller network module controlled by another CPU. • Change the setting to the start I/O No. of the CC-Link IE controller network module controlled by the host CPU.
	A network parameter of a CC-Link IE controller network module acting as a normal station was changed to Control station. Or a network parameter of a CC-Link IE controller network module acting as a control station was changed to Normal station. (Network parameters are reflected to the module by resetting.)	Reset the CPU module.
	• The number of modules specified with the CC-Link IE controller network parameter does not match the number of modules actually mounted. • The start I/O No. specified with the CC-Link IE controller network parameter is different from the actual I/O No. • Invalid data are included in the parameters. • The network type of the CC-Link IE controller network was changed. (To change the network type, the status must be changed from RESET to RUN.)	• Check the network parameters and the actual mounting status, and if any difference is identified, make them matched. If a network parameter is corrected, it must be written to the CPU module. • Check the setting for the number of extension base units. • Check the connected conditions of the extension base unit(s) and extension cables. If a GOT is connected to the main or extension base unit in the bus topology, also check the connection status. • If the error occurs even after the above checks, replace the module because the hardware may be faulty. Or, consult your local Mitsubishi service center or representative, explaining the details of the problem.
	• A CC-Link IE controller network module is assigned to the start I/O No. in the MELSECNET/H network parameter setting. • A MELSECNET/H module is assigned to the start I/O No. in the CC-Link IE controller network parameter setting.	
	• No CC-Link IE controller network parameters are set even though a CC-Link IE controller network module is mounted. • No MELSECNET/H network parameters are set even though CC-Link IE controller network module(s) and MELSECNET/H module(s) are mounted.	
	The network parameters were written to a drive different from a valid one.	Write the network parameters to a valid drive.
	In the multiple CPU system, a MELSECNET/H module controlled by another CPU is assigned to the start I/O No. in the MELSECNET/H network parameter setting.	• Delete the MELSECNET/H network parameter for the MELSECNET/H module controlled by another CPU. • Change the setting to the start I/O No. of the MELSECNET/H module controlled by the host CPU.
	A link parameter of a MELSECNET/H module acting as a normal station was changed to Control station. Or a link parameter of a MELSECNET/H module acting as a control station was changed to Normal station. (Link parameters are reflected to the module by resetting.)	Reset the CPU module.
	• The number of modules specified with the MELSECNET/H parameter does not match the number of modules actually mounted. • The start I/O No. specified with the MELSECNET/H parameter is different from the actual I/O No. • Invalid data are included in the parameters. • The station type of the MELSECNET/H was changed while power is on. (To change the station type, the status must be changed from RESET to RUN.)	• Check the network parameters and the actual mounting status, and if any difference is identified, make them matched. If a network parameter is corrected, it must be written to the CPU module. • Check the setting for the number of extension base units. • Check the connected conditions of the extension base unit(s) and extension cables. If a GOT is connected to the main or extension base unit in the bus topology, also check the connection status. • If the error occurs even after the above checks, replace the module because the hardware may be faulty. Or, consult your local Mitsubishi service center or representative, explaining the details of the problem.

(To the next page)

Table 16.5 Error codes (continued)

Error code	Error message	Error information	LED status		CPU operation status	
			RUN	ERR.		
3101	LINK PARA.ERROR	Parameter No.*1	Off	Flashing	Stop	
3102	LINK PARA.ERROR	Parameter No.*1	Off	Flashing	Stop	
3103	LINK PARA.ERROR	Parameter No.*1	Off	Flashing	Stop	
3104	LINK PARA.ERROR	Parameter No.*1	Off	Flashing	Stop	
3105	LINK PARA.ERROR	Parameter No.*1	Off	Flashing	Stop	
3107	LINK PARA.ERROR	Parameter No.*1	Off	Flashing	Stop	
3300	SP.PARA.ERROR	Parameter No.*1	Off	Flashing	Stop	
3301	SP.PARA.ERROR	Parameter No.*1	Off	Flashing	Stop	
3302	SP.PARA.ERROR	Parameter No.*1	Off	Flashing	Stop	
3303	SP.PARA.ERROR	Parameter No.*1	Off	Flashing	Stop	

* 1 For the parameter setting location for each parameter number, refer to the following.

☞ Appendix 8

Error code	Error details and cause	Corrective action
3101	• When the station No. of a MELSECNET/H module is 0, parameters for the PLC-to-PLC network are set. • When the station No. of a MELSECNET/H module is other than 0, parameters for a remote master are set.	Correct the type or station No. of the MELSECNET/H module.
	The refresh parameters for the CC-Link IE controller network are out of range.	• Check the network parameters and the actual mounting status, and if any difference is identified, make them matched. If a network parameter is corrected, it must be written to the CPU module. • Check the setting for the number of extension base units. • Check the connected conditions of the extension base unit(s) and extension cables. If a GOT is connected to the main or extension base unit in the bus topology, also check the connection status. • If the error occurs even after the above checks, replace the module because the hardware may be faulty. Or, consult your local Mitsubishi service center or representative, explaining the details of the problem.
	• The network No. specified with the parameter is different from the actual network No. • The start I/O No. specified with the parameter is different from the actual I/O No. • The network type specified with the parameter is different from the actual network type. • Network refresh parameters for MELSECNET/H or MELSECNET/10 are out of range.	
3102	A network parameter error on the CC-Link IE controller network was detected.	• Correct the network parameters and write them. • If the error occurs after the correction, the hardware is faulty. Please consult your local Mitsubishi service center or representative, explaining the details of the problem.
	• A network module detected a network parameter error. • A MELSECNET/H network parameter error was detected.	
3103	In the multiple CPU system, a Q series Ethernet module controlled by another CPU is specified for the start I/O No. in the Network parameter setting for Ethernet.	• Delete the network parameter of the Q series Ethernet module controlled by another CPU. • Change it to the start I/O No. of the Q series Ethernet module that the host CPU controls.
	• Although the number of modules is set to one or greater in the Ethernet module count parameter setting, the number of actually mounted modules is zero. • The start I/O No. in the Network parameter setting for Ethernet differs from that of the actually mounted module.	• Correct the network parameters and write them. • If the error occurs after the correction, the hardware is faulty. Please consult your local Mitsubishi service center or representative, explaining the details of the problem.
3104	• The Ethernet, CC-Link IE controller network, MELSECNET/H and MELSECNET/10 use the same network number. • The network number, station number or group number set in the Parameter is out of range. • The specified I/O No. is outside the range of the CPU module. • The Ethernet-specific parameter setting is not normal.	• Correct the network parameters and write them. • If the error occurs after the correction, the hardware is faulty. Please consult your local Mitsubishi service center or representative, explaining the details of the problem.
3105	In the multiple CPU system, a Q series CC-Link module controlled by another CPU is specified for the start I/O No. in the CC-Link parameter setting.	• Delete the network parameter of the Q series CC-Link module controlled by another CPU. • Change it to the start I/O No. of the Q series CC-Link module that the host CPU controls.
	• Although the number of modules is set to one or greater in the CC-Link module count parameter setting, the number of actually mounted modules is zero. • The start I/O No. in the common parameters is different from that of the actually mounted module. • The station type in the CC-Link module count parameter setting is different from that of the actual station.	• Correct the network parameters and write them. • If the error occurs after the correction, the hardware is faulty. Please consult your local Mitsubishi service center or representative, explaining the details of the problem.
3107	• The CC-Link parameter setting is incorrect. • The current mode is not allowed for the version of the mounted CC-Link module.	Check the parameter setting.
3300	The start I/O No. of an intelligent function module set in GX Configurator differs from the actual I/O No.	Check the parameter setting.
3301	The refresh parameter of an intelligent function module is outside the range, or the refresh range exceeded the file register capacity.	Check the parameter setting.
3302	An intelligent function parameter is not normal.	Check the parameter setting.
3303	In the multiple CPU system, the automatic refresh or other parameter setting was performed for an intelligent function module controlled by another CPU.	• Delete the automatic refresh or other parameter setting of the intelligent function module controlled by another CPU. • Change it to the automatic refresh or other parameter setting of the intelligent function module that the host CPU controls.

(d) Error code list (5000 to 5999)

Table 16.6 Error codes

Error code	Error message	Error information	LED status		CPU operation status	
			RUN	ERR.		
5000	SYSTEM WDT ERROR	Time (set value: ms)	Off	Flashing	Stop	
5001	USER WDT ERROR	Time (set value: ms)	Off	Flashing	Stop	
5012	LNKREF. TIME OVER	Link device refresh cycle (set value: ms)/ Link device refresh time (measured value: ms)	On	On	Continue	

Error code	Error details and cause	Corrective action
5000	The system watchdog timer controlled by the system detected an error, or an error occurred in the system software. • The time set for the system watchdog timer is too short. • A task of high CPU utilization is running. • A program that will cause an error in the memory, stack, etc. was executed. • It debugged while online connection with Tornado or Workbench. • The command was executed from Shell for debugging.	• Increase the system WDT time value in System settings of C Controller setting utility. • Decrease the CPU utilization of the task, or disable the task operation. • Check the user program. • Check the command executed from Shell. • Please consult your local Mitsubishi service center or representative, explaining the details of the problem.
5001	Since the WDT reset (QBF_ResetWDT function) was not executed within the user WDT setting time, the USER WDT (watchdog timer) controlled by the system detected an error. Or, an error occurred in the user program. • The user WDT setting time is too short. • A task of high CPU utilization is running. • A program that will cause an error in the memory, stack, etc. was executed. • It debugged while online connection with Tornado or Workbench. • The command was executed from Shell for debugging.	• Increase the user WDT time set by the QBF_StartWDT function. • Decrease the CPU utilization of the task, or disable the task operation. • Check the user program. • Check the command executed from Shell. • Please consult your local Mitsubishi service center or representative, explaining the details of the problem.
5012	The time for link device refresh with the CC-Link IE controller network module or MELSECNET/H module exceeded the set value (link device refresh cycle).	• Increase the link device refresh cycle value in CC IE Control utility or MELSECNET/H utility, or check and correct the refresh parameter setting.
	• It debugged while online connection with Tornado or Workbench. • The command was executed from Shell for debugging. • The C Controller module was restarted while online connection with Tornado or Workbench (Q12DCCPU-V only). For precautions for program debugging, refer to the following. ☞ C Controller Module User's Manual (Utility Operation, Programming)	• Restart the C Controller module with Workbench connection disconnected. • Check the command executed from Shell.

(e) Error code list (7000 to 10000)

Table 16.7 Error codes

Error code	Error message	Error information	LED status		CPU operation status	
			RUN	ERR.		
7000	MULTI CPU DOWN	Module No.[*1]	Off	Flashing	Stop	
7002	MULTI CPU DOWN	Module No.[*1]	Off	Flashing	Stop	
7010	MULTI EXE.ERROR	Module No.[*1]	Off	Flashing	Stop	
7011	MULTI EXE.ERROR	Module No.[*1]	Off	Flashing	Stop	
7013	MULTI EXE.ERROR	Module No.[*1]	Off	Flashing	Stop	
7020	MULTI CPU ERROR	Module No.[*1]	On	On	Continue	
7036	CPU LAY ERROR	Module No.[*1]	Off	Flashing	Stop	

[*1] The error information (module No.) in "Event information" in the <<Event history>> tab of C Controller setting utility is displayed as follows:
"Error information: XX - YYY"
XX (DEC.): Slot No.(0 to 63), CPU No.(1 to 4)
YYY (HEX.): Start I/O No.÷16 (000 to 0FF, 3E0 to 3E3)

Error code	Error details and cause	Corrective action
7000	• In the operating mode of the multiple CPU system, a CPU error occurred in the CPU for which "All station stop by stop error of CPU" is selected. • In the multiple CPU system, a CPU module incompatible with the multiple CPU system is mounted. • When CPU No.1 is a Basic model QCPU, a "CPU (Empty)" slot is set on the right side of the C Controller module.	• Identify the error of the CPU module and eliminate the error cause. • Remove the incompatible CPU module from the main base unit. • Change the setting.
	In the multiple CPU system, a stop error occurred in CPU No.1 during power-on, and thereby the other CPUs (No.2 to No.4) cannot start.	Identify the error of the CPU module and eliminate the error cause.
7002	• During the initial communication in the multiple CPU system, no response returns from the target CPU. • In the multiple CPU system, a CPU module incompatible with the multiple CPU system is mounted.	• Reset the CPU module and restart it. If the same error is displayed again, either of the CPU modules is faulty. Please consult your local Mitsubishi service center or representative, explaining the details of the problem. • Remove the incompatible CPU module from the main base unit. Or, replace the incompatible CPU module with a compatible one.
7010	• In the multiple CPU system, a faulty CPU module is mounted. • In the multiple CPU system, a CPU module incompatible with the multiple CPU system is mounted. (A CPU module compatible with the multiple CPU system detects this error.) • In the multiple CPU system, any of the CPU No.2 to No.4 was reset with power ON. (A CPU that released from the reset status detects this error.)	• Replace the faulty CPU. • Replace it with a CPU module that is compatible with the multiple CPU system. • Do not reset any of CPU No.2 to No.4. • Reset CPU No.1 and restart the multiple CPU system.
7011	In a multiple CPU system, the either of following setting was set. • Set the multiple CPU auto refresh setting to the CPU module that cannot execute the multiple CPU auto refresh setting. • Set the I/O sharing when using Multiple CPUs to the CPU module that cannot read the I/O of outside of the group.	• Correct the multiple CPU automatic refresh setting. • Correct the "I/O sharing when using multiple CPUs" setting.
	It doesn't meet the system requirement to use the multiple CPU high speed transmission. • It is not used QnUD(H)CPU or QnDCCPU in CPU No.1. • It is not used the multiple CPU high speed main base unit (Q3□DB). • The send range is set excluding 0 point to the CPU module that is not supported the multiple CPU high speed transmission. • The send range is set excluding 0 point to the CPU module that is not supported the multiple CPU interchangeability area.	• Change the system configuration so that the conditions for using the multiple CPU high speed transmission function are met. • When performing automatic refresh in the multiple CPU compatible area, set the send range of a CPU that does not support the function to zero points.
7013	Q172(H)CPU(N) or Q173(H)CPU(N) is installed in CPU slot and slot 0 to 2 (There is a possibility that the module breaks down.).	Check QCPUs and Motion CPUs that can be used in multiple CPU systems, and change the system configuration.
7020	In the operating mode of the multiple CPU system, an error occurred in the CPU for which "All station stop by stop error of CPU" is not selected. (A normal CPU module detects this error.)	Identify the error of the CPU module, and eliminate the error.
7036	The number of connected CPU set in the multiple CPU setting doesn't match the number of connected CPU that is decided according to fixing point of CPU module.	• Mount the CPU module in the correct slot. • Change the host CPU No. set in the multiple CPU setting to the one determined by the mounting position of the CPU module.

16.4 Error Code List

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	○

This section explains error codes that are returned upon error occurrence.

16.4.1 Error codes and actions (for errors occurred in function execution)

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	○

The error codes returned in bus interface or MELSEC data link function execution are returned as a return value.

(1) Common error codes returned by bus interface functions and MELSEC data link functions

The following table provides the error descriptions and corrective actions to individual error code commonly returned in execution of bus interface functions and MELSEC data link functions.

Table 16.8 Common error codes returned by bus interface and MELSEC data link functions

Return value (HEX)	Error description	Corrective action
0	Completed normally	—
1 (0x1)	Driver not started The driver has not been started.	Correct the error that occurred when starting the driver. Check the status of the standard ROM and system ROM. Check if the operating system is running normally.
2 (0x2)	Timeout error A timeout occurred while waiting for response.	Check the operation status and mounting condition of the accessed station(s). Retry from the user program. Increase MD function timeout value in C Controller setting utility.
66 (0x42)	Already opened The specified channel has already been opened.	Open once.
67 (0x43)	Already closed The specified channel has already been closed.	Close once.
69 (0x45)	Processing code error An unsupported processing code was issued.	Use the supported processing code.
70 (0x46)	Station No. error The specified station No. is incorrect. This station received a request destined for another station. Or, the station No. corresponds to the own station (0xFF) but the network No. is not 0.	Correct the station No. in the user program.
71 (0x47)	Data reception error (When a RECV request is issued.) Data have not been received.	Wait until data are received.
77 (0x4D)	Memory error Sufficient memory could not be allocated.	Terminate other task(s) currently running, or decrease the size of data to be accessed. Check if the operating system is running normally. Reboot the operating system.*1

* 1 In a single CPU system, reset the C Controller module.
In a multiple CPU system, reset CPU No. 1.

(To the next page)

Table 16.8 Common error codes returned by bus interface and MELSEC data link functions (continued)

Return value (HEX)	Error description	Corrective action
85 (0x55)	Network channel No. error (When a SEND/RECV request is issued.) Channel No. error	Check the specified channel No. when a SEND/RECV request is issued.
102 (0x66)	Transmission error Transmitting data has failed.	Retry. Check if the operating system is running normally. Reboot the operating system.[*1]
103 (0x67)	Reception error Receiving data has failed.	Retry. Check if the operating system is running normally. Reboot the operating system.[*1]
130 (0x82)	Device No. error The specified device No. is out of range. Device No. in the bit device settings is not a multiple of 8.	Check the device No. When specifying device No.32768(8000H) or higher, use an extended function.
131 (0x83)	Device points error The specified number of points is out of range. The number of points in the bit device settings is not a multiple of 8.	Check the number of device points. When specifying a size of 32768(8000H) bytes or more, use an extended function.
8193 (0x2001)	The specified device is a device that cannot be used with the Motion CPU. Or, the specified device is out of the device range.	Check the specified device name. Check the specified device No. and size.
8320 (0x2080)	The number of write data points set by the D(P).DDWR instruction is invalid.	Check the specified size.
8321 (0x2081)	The number of read data points set by the D(P).DDRD instruction is invalid.	Check the specified size.
8322 (0x2082)	The interrupt pointer No. set by the D(P).GINT instruction is out of the range of 0 to 15.	Check the specified interrupt number.
8448 (0x2100)	• There were 33 or more instructions simultaneously requested from the PLC CPU / C Controller module to the Motion CPU in D(P).GINT. The Motion CPU cannot process the instructions. • There were 65 or more instructions simultaneously requested to the Motion CPU in D(P).SFCS. The Motion CPU cannot process the instructions. • There were 65 or more instructions simultaneously requested from the PLC CPU / C Controller module to the Motion CPU in D(P).SVST and D(P).CHGA combined. The Motion CPU cannot process the instructions. • There were 65 or more instructions simultaneously requested from the programmable controller CPU / C language controller module to the Motion CPU in D(P).DDRD and D(P).DDWR combined. The Motion CPU cannot process the instructions.	• Wait for the Motion CPU to process instruction requests, and try again. • Reduce the number of instructions requested to the Motion CPU.
8704 (0xx2200)	The Motion SFC program number to be started is out of the range of 0 to 255.	Check the specified SFC program No..
8705 (0x2201)	The servo program number to be executed is out of the range of 0 to 4095.	Check the specified servo program No..
8706 (0x2202)	The axis number set in the D(P).SVST instruction is invalid.	Check the specified axis No..
8707 (0x2203)	The axis number set in the D(P).CHGA instruction is invalid.	Check the specified axis No..
8708 (0x2204)	The axis number set in the D(P).CHGV instruction is invalid.	Check the specified axis No..
8709 (0x2205)	The axis number set in the D(P).CHGT instruction is invalid.	Check the specified axis No..

[*1] In a single CPU system, reset the C Controller module.
In a multiple CPU system, reset CPU No. 1.

(To the next page)

Table 16.8 Common error codes returned by bus interface and MELSEC data link functions (continued)

Return value (HEX)	Error description	Corrective action
16386 (0x4002)	An unprocessable request was received.	Change the request destination.
16400 (0x4010)	Not executable during running	Reset the C Controller module when it is in the STOP status.
16432 (0x4030)	The specified device type does not exist.	Check the device type.
16433 (0x4031)	The specified device No. is out of range. The block No., network No., start I/O No. ÷16, or CPU No. of the specified device is invalid.	Check the device No. Check the block No., network No., start I/O No. ÷16, and CPU No. (device type) of the specified device. On the target, check if the specified device, block No., network No., start I/O No. ÷16, and CPU No. are valid. Use the extended function since 32768(8000H) the specified device No.
16448 (0x4040)	The module does not exist.	Do not issue a request that generated an error to the specified special function.
16449 (0x4041)	The number of device points is out of range.	Check the start address and number of access points. Access within the range. Use the extended function if the specified size is 32768 (8000H) or more.
	The specified device is not accessible.	Check the start address and the number of access points, and access an accessible device.
16450 (0x4042)	Corresponding module error.	Check if the specified module is operating normally.
16451 (0x4043)	The module is not mounted on the specified slot.	Check the start I/O No. of the specified module.
	In a single CPU system configuration, an access to the CPU shared memory of the C Controller module was attempted.	To access the CPU shared memory of the C Controller module, configure a multiple CPU system.
16480 (0x4060)	Online registration error. The online debug function is currently running on another peripheral device.	• After the operation has ended on the other peripheral device, execute the function once again. • In a case where the operation is in progress or the operation has been suspended on the other peripheral device, execute the operation once again on the other peripheral device, wait for the operation to end normally, and then execute the operation once again.
16523 (0x408B)	Remote request not executable	Allow "Remote reset" in the system settings, and then execute remote reset.
16641 (0x4101)	The number of write data is out of the storage device range for data to be written.	Check and revise the program to a proper user program.
17232 (0x4350)	The specified destination CPU module is wrong. (1) A CPU No. set as reserved was specified. (2) A CPU No. not mounted was specified. (3) The start I/O No. of the destination CPU module divided by 16 is out of the range of 3E0H to 3E3H.	Check and revise the program to a proper user program.

(To the next page)

Table 16.8 Common error codes returned by bus interface and MELSEC data link functions (continued)

Return value (HEX)	Error description	Corrective action
17233 (0x4351)	The instruction cannot be executed on the specified target CPU module. (1) The instruction name is wrong. (2) A non-supported instruction was executed on the target CPU module.	Check and revise the program to a proper user program.
17234 (0x4352)	The number of devices of the specified instructions is wrong.	Check and revise the program to a proper user program.
17235 (0x4353)	An inapplicable device was specified in the specified instruction.	Check and revise the program to a proper user program.
17236 (0x4354)	A non-supported string was specified in the specified instruction.	Check and revise the program to a proper user program.
17237 (0x4355)	The number of write data is out of the range of 1 to 20.	Check and revise the program to a proper user program.
18944 (0x4A00)	Network No. or station No. is incorrect. Routing parameter does not exist in other network settings, or the module with relay network No. and station No. specified in the routing parameter settings is not mounted.	Check the specified network No. and station No. Check the routing parameter settings.
19200 (0x4B00)	An error was detected in the target.	Check the target. Confirm the error occurred in the target or relay station, and then take a corrective action.
19201 (0x4B01)	The target is not the bus master CPU.	If the target is not the bus master CPU (CPU 1), it cannot execute remote reset.
19203 (0x4B03)	Unsupported communication path error The request cannot be executed via the specified communication path/target.	Check the communication path/target that supports the request. Do not issue a request that has generated an error to a C Controller module or PC CPU.
19457 (0x4C01)	The instruction for the multiple CPU system is not supported by operating system of the Motion CPU. (the error code returned by the Motion CPU) ☞ Refer to the manual of the Motion CPU currently used.	Check the user program on C Controller module.
19458 (0x4C02)	Motion SFC program No. error (the error code returned by the Motion CPU) ☞ Refer to the manual of the Motion CPU currently used.	Check the user program on C Controller module.
19460 (0x4C04)	Axis No. specified in SVST function is invalid. (the error code returned by the Motion CPU) ☞ Refer to the manual of the Motion CPU currently used.	Check the user program on C Controller module.
19461 (0x4C05)	Axis No. specified in CHGA function is invalid. (the error code returned by the Motion CPU) ☞ Refer to the manual of the Motion CPU currently used.	Check the user program on C Controller module.
19462 (0x4C06)	Axis No. specified in CHGV function is invalid. (the error code returned by the Motion CPU) ☞ Refer to the manual of the Motion CPU currently used.	Check the user program on C Controller module.
19463 (0x4C07)	Axis No. specified in CHGT function is invalid. (the error code returned by the Motion CPU) ☞ Refer to the manual of the Motion CPU currently used.	Check the user program on C Controller module.
19464 (0x4C08)	A total of 32 or more instructions are simultaneously requested to the Motion CPU from the programmable controller CPU/C Controller module, and the Motion CPU cannot process them. (the error code returned by the Motion CPU) ☞ Refer to the manual of the Motion CPU currently used.	Check the user program on C Controller module.

(To the next page)

Table 16.8 Common error codes returned by bus interface and MELSEC data link functions (continued)

Return value (HEX)	Error description	Corrective action
19465 (0x4C09)	CPU No. of the requesting CPU is incorrect. (the error code returned by the Motion CPU) ☞ Refer to the manual of the Motion CPU currently used.	Check the user program on C Controller module.
19584 (0x4C80)	H/W error of the target CPU (the error code returned by the Motion CPU)	Check the operating state of the target CPU. Take a corrective action in the manual of the target CPU.
19585 (0x4C81)	H/W error of the target CPU (the error code returned by the Motion CPU)	Check the operating status of the target CPU. Take a corrective action in the manual of the target CPU.
19587 (0x4C83)	H/W error of the target CPU (the error code returned by the Motion CPU)	Check the operating status of the target CPU. Take a corrective action in the manual of the target CPU.
19588 (0x4C84)	H/W error of the target CPU (the error code returned by the Motion CPU)	Check the operating status of the target CPU. Take a corrective action in the manual of the target CPU.
-475(0xFE25) to -3839(0xF101)	☞ Q Corresponding MELSECNET/H Network System Reference Manual (PLC to PLC network) ☞ For QnA/Q4AR MELSECNET/10 Network System Reference Manual ☞ Type MELSECNET/10 Network system (PLC to PLC network) Reference Manual	Same as in the left.
-2174 (0xF782)	Target station No. error The station No. specified for processing the request is the own station No.	Check the target station No. When sending a request to the own station, select station No. 255 (0xFF).
-7656 (0xE218)	Target station No. error The station No. specified for processing the request is the own station No.	Check the target station No.When sending a request to the own station, select station No. 255 (0xFF).
-4097(0xEFFF) to -8192(0xE000)	☞ MELSEC-Q CC-Link IE Controller Network Reference Manual	Same as in the left
-16386(0xBFFE) to -20222(0xB102)	☞ MELSEC-Q CC-Link System Master/Local Module User's Manual ☞ Control & Communication Link System Master/Local Module type AJ61QBT11/A1SJ61QBT11 User's Manual ☞ Control & Communication Link System Master/Local Module type AJ61BT11/A1SJ61BT11 User's Manual	Same as in the left
-18558 (0xB782)	Target station No. error The station No. specified for processing the request is the own station No.	Check the target station No. When sending a request to the own station, select station No. 255(0xFF).
-18572 (0xB774)	Transient unsupported error A transient request was sent to a non-intelligent device station.	Check the target station No. Check if the target station is an intelligent device station.
-25056 (0x9E20)	Processing code error The target station cannot process the request.	Check the station No. and processing code. Do not execute the function other than mdTypeRead to other station's I/F board.
-26334 (0x9922)	Reset error Another task using the same channel executed reset operation while accessing other station. Reset operation was executed during monitoring from the utility.	Retry. Monitor again.
-26336 (0x9920)	Request error for other loop Routing to other loop was requested to I/F board.	Change the routing request destination to AnUCPU, QnACPU, or QCPU.

(To the next page)

Table 16.8 Common error codes returned by bus interface and MELSEC data link functions (continued)

Return value (HEX)	Error description	Corrective action
-28150 (0x920A)	Device access error during data link stop RX/RY/RWw/RWr was accessed when the own station was not transmitting data.	Restart the data link. Note: Writing/reading data will be performed even when this error occurs. However, the data are not guaranteed.
-28151 (0x9209)	APS NO. error Invalid response data were received.	Check if there is an error in the target CPU, CC-Link module, or MELSECNET/H module. If the status is normal, try again.
-28155 (0x9205)	SRAM wrtie error Battery-backed-up RAM writing failed.	Check if an interrupt routine (interrupt program) is writing data into the same battery backed-up RAM area. Change the C Controller module if the error persists.
-28158 (0x9202)	WDT error Watchdog timer (WDT) error occurred.	Reboot the operating system.*1 Reset the module.
-28410 (0x9106)	Target CPU busy error Target CPU is busy.	Add a program to wait until the processing is completed or to retry. Increase the timeout value in the user program.
-28412 (0x9104)	Target CPU unsupported error An unsupported request was sent to the target CPU.	Change the target CPU No. in the user program.
-28413 (0x9103)	Target CPU down error	Check the operating status of the target CPU and trouble shoot the error.
-28414 (0x9102)	Target CPU abnormal start error A request was sent to the CPU which was not operating normally.	Check the operating status of the target CPU and trouble shoot the error.
-28415 (0x9101)	Target CPU critical error A request was sent to the CPU where a critical error occurred.	Check the operating status of the target CPU and troubleshoot the error.
-28416 (0x9100)	Target CPU not mounted error	Check if the target CPU is mounted properly. Change the target CPU No. specified in the user program.

* 1 In a single CPU system, reset the C Controller module.
In a multiple CPU system, reset CPU No. 1.

(To the next page)

Table 16.8 Common error codes returned by bus interface and MELSEC data link functions (continued)

Return value (HEX)	Error description	Corrective action
-28622 (0x9032)	Target module busy error Target module is busy.	Add a program to wait until the processing is completed or to retry.
-28624 (0x9030)	Function unsupported error Processing was executed for the module that does not support the block data assurance per station function. Processing was executed for the module where the block data assurance per station function was not activated. Own CPU does not control the specified module.	Check the version to see if the target module supports the block data assurance per station function. Check if the block data assurance per station setting is enabled for the target module in the <<Parameter settings>> tab in CC-Link utility. Check if the control CPU of the specified module is own CPU.
-28625 (0x902F)	Intelligent function module offline error Intelligent function module was accessed while offline.	Check the mode state of the intelligent function module and access the module while online.
-28626 (0x902E)	Control data error The specified control data is out of range.	Correct the control data setting.
-28627 (0x902D)	Transient unsupported error A transient request cannot be executed via the specified communication path and target. (With CC-Link communication, when the station No. of the own station was "64", other station was specified.)	Check a communication path and target which supports the transient request. Change the station No. of the own station.
-28628 (0x902C)	Pointer address error Pointer address is invalid. The address of a short type pointer is not a multiple of 2. The address of a long type pointer is not a multiple of 4.	Check the pointer type and address (cast, structure alignment, etc.).
-28629 (0x902B)	WDT not running error WDT was reset when WDT was started up.	Start the WDT and then reset WDT.
-28630 (0x902A)	WDT startup error WDT was started up while WDT was running.	Stop WDT and then restart WDT.

(To the next page)

Table 16.8 Common error codes returned by bus interface and MELSEC data link functions (continued)

Return value (HEX)	Error description	Corrective action
-28631 (0x9029)	Buffer access range error The specified offset is out of range. The specified offset plus size is out of range.	Check the offset. Check the buffer size. Check the offset plus size.
-28632 (0x9028)	I/O No. error The specified I/O No. is out of range. The module is not mounted on the specified I/O No., or the specified module cannot be accessed.	Check the I/O No.
-28633 (0x9027)	Non-controlled module read error Reading from a non-controlled module is not allowed.	Clear "Enable online module change with another CPU" selection in the multiple CPU setting of C Controller utility. Check if the control CPU of the specified module is own CPU (C Controller module/PC CPU).
-28634 (0x9026)	Intelligent function module down error There is a problem in the intelligent function module.	Check if the intelligent function module is mounted properly. Replace the intelligent function module or base unit.
-28635 (0x9025)	Intelligent function module error The intelligent function module is not mounted on the accessed slot.	Check the I/O No. Check if the intelligent function module is mounted properly.
-28636 (0x9024)	Control bus error Control bus error occurred.	Check if there is an error in CPU 1 in the multiple CPU system. Check if the intelligent function module is mounted properly. Replace the intelligent function module or base unit.
-28638 (0x9022)	Multiple CPU unsupported operation error	Reset CPU 1.
-28640 (0x9020)	STOP/PAUSE error Output data was written to output module when the CPU operating status is STOP/PAUSE. Buffer memory data was written to intelligent function module when the CPU operating status is STOP/PAUSE.	Change the CPU operating status to RUN.
-28653 (0x9013)	I/O assignment error Input data was read from output module. Output data was written to input module. Output data was read from input module.	Check the I/O No.
-28654 (0x9012)	Non-controlled module write error Own CPU does not control the specified module.	Check if the control CPU of the specified module is own CPU (C Controller module/PC CPU).
-28660 (0x900C)	Access size error The specified size is out of range.	Check the offset and size.
-28661 (0x900B)	Access area error The specified area cannot be accessed.	Check the offset and size.
-28662 (0x900A)	CPU No. error The specified CPU No. is out of range or invalid.	Check the CPU No. Check the operating status of the specified CPU.
-28663 (0x9009)	Base unit No. error The specified base unit No. is out of range.	Check the base unit No.

(To the next page)

Table 16.8 Common error codes returned by bus interface and MELSEC data link functions (continued)

Return value (HEX)	Error description	Corrective action
-28664 (0x9008)	Data transmission area occupied.	Retry.
-28665 (0x9007)	No registration data error	Reboot the operating system.*1
-28666 (0x9006)	Data length error	Reboot the operating system.*1
-28668 (0x9004)	Reply data stored error Reply data has already been stored.	Resend the request.
-28669 (0x9003)	Area No. error The specified area No., offset address, or mode is out of range.	Check the area No., offset address, or mode.
-28671 (0x9001)	Module identification error	Check the parameter. Check the specified module. Reboot the operating system.*1
-28672 (0x9000)	Processing code error	Reinstall software package. Reboot the operation system.*1

* 1 In a single CPU system, reset the C Controller module.
In a multiple CPU system, reset CPU No. 1.

(2) Error codes returned by bus interface functions

The following provides the error descriptions and corrective actions corresponding to individual error code returned in execution of bus interface functions.
For the common error codes returned by bus interface functions and MELSEC data link functions, refer to the following.

☞ (1) in this section

Table 16.9 Error codes returned by bus interface functions

Return value (HEX)	Error description	Corrective action
0	Completed normally	—
-201 (0xFF37)	Module identification error The specified module identification is invalid.	Check the parameter. Check the specified module. Reboot the operating system.[*1]
-202 (0xFF36)	Path error The specified path is invalid.	Use the path pointer returned by "QBF_Open" function.
-203 (0xFF35)	I/O No. error The specified I/O No. is out of range.	Check the I/O No.
-204 (0xFF34)	I/O access size error The specified I/O access size is out of range.	Check the I/O access size (I/O No. and read/ write size.)
-205 (0xFF33)	I/O No. error The specified I/O No. is out of range.	Check the I/O No.
-206 (0xFF32)	Program execution type error The specified program execution type is out of range.	Check the program execution type.
-208 (0xFF30)	Offset error The specified offset is out of range. AnS series module (buffer memory) was accessed.	Check the offset. Check the I/O No.
-209 (0xFF2F)	Buffer memory size error The specified offset plus size is out of range. The address of data storage buffer pointer is 0.	Check the specified buffer memory size. Check the offset plus size. Check the data storage buffer pointer.
-210 (0xFF2E)	Read area size error The read area size is smaller than the read size.	Check the read size and read area size.
-211 (0xFF2D)	Time setting error Time setting is out of range.	Check the specified time.
-214 (0xFF2A)	Intelligent function module error The intelligent function module is not mounted on the accessed slot.	Check the I/O No. Check if the intelligent function module is mounted properly.
-217 (0xFF27)	Driver not started Driver has not been started.	Check if the driver has been started.
-218 (0xFF26)	Bus already opened "QBF_Open" is executed twice.	Only one instance of "QBF_Open" can be executed.
-219 (0xFF25)	Program name error The specified program name is invalid (The program name does not exist in the programmable controller CPU or is not registered with a parameter).	Check the program name.
-220 (0xFF24)	WDT type error The specified WDT type is out of range.	Check the WDT type.
-222 (0xFF22)	Bus master CPU reset error Remote reset for the bus master CPU (CPU 1) has failed.	Allow "Remote reset" in the system settings for the bus master CPU (CPU 1). Change the status of the bus master CPU (CPU 1) to STOP. Check if the bus master CPU (CPU 1) is programmable controller CPU.

[*1] In a single CPU system, reset the C Controller module.
In a multiple CPU system, reset CPU No. 1.

(To the next page)

Table 16.9 Error codes returned by bus interface functions (continued)

Return value (HEX)	Error description	Corrective action
-223 (0xFF21)	Memory allocation error Sufficient memory could not be allocated.	Check available memory.
-224 (0xFF20)	LED set value error The specified LED control value is out of range.	Check the LED set value.
-225 (0xFF1F)	Event No. error The specified event No. is out of range or duplicated.	Check the event No.
-227 (0xFF1D)	Control code send error Sending the control code has failed.	Retry. Check if the operating system is running normally. Reboot the operating system.*1
-231 (0xFF19)	Event timeout error A timeout occurred while waiting for the event.	Increase the timeout value.
-232 (0xFF18)	CPU No. error The specified CPU No. is incorrect. The specified CPU cannot execute the request.	Specify a different CPU No. Do not issue a request that generated an error to the CPU.
-234 (0xFF16)	Event wait error An error other than timeout occurred while waiting for the event.	Ensure the operating system is not forcibly terminating the program. Check if the operating system is running normally. Reboot the operating system.*1
-235 (0xFF15)	Number of event settings error The specified number of event settings is out of range.	Check the number of event settings.
-236 (0xFF14)	Remote operation code error The specified remote operation code is out of range.	Check the remote operation code.
-237 (0xFF13)	Event source/event information character string error Any character other than one-byte alphanumeric ('a'-'z', 'A'-'Z', '0'-'9') is in event source "pcSrcStr[]". Any character string 11 bytes or more is in event source "pcSrcStr[]". Any character string 201 bytes or more is in event information "pcAddMsg[]".	Change the event source / event information character string.
-238 (0xFF12)	Event log registration error Event log registration has failed.	Check the status of the parameter storage drive (standard ROM/standard RAM) (free space/write authority). Check if the operating system is running normally. Reboot the operating system.*1
-239 (0xFF11)	CF card error CF card is not inserted.	Check if the CompactFlash card (CF card) is inserted.
-240 (0xFF10)	Clock data error The clock data setting/clock data setting to be read is invalid.	Check the clock data setting. If this error occurred when reading the clock data, set the clock data.
-241 (0xFF0F)	Cycle error The specified event cycle is out of range. The event cycle is being registered.	Check the specified event cycle. Check if the event cycle is being registered.
-242 (0xFF0E)	Synchronization type error The specified synchronization type is out of range.	Check the specified synchronization type.
-243 (0xFF0D)	SRAM size error The specified offset plus size is out of range.	Check the specified size. Check the offset plus size.

* 1 In a single CPU system, reset the C Controller module.
In a multiple CPU system, reset CPU No. 1.

(To the next page)

Table 16.9 Error codes returned by bus interface functions (continued)

Return value (HEX)	Error description	Corrective action
-244 (0xFF0C)	Standard ROM shut down error Standard ROM shut down has failed.	Check if the standard ROM is being accessed. Check if all user files are closed. In the case of Q12DCCPU-V, standard ROM is not supported. Correct the user program so that this function is not called.
-245 (0xFF0B)	Not executable during interrupt routine A value other than 1 (ISR) was specified in "sFlg", and the function was executed as interrupt routine.	Specify 1 (ISR) in "sFlg" to correct the routine registered in QBF_EntryWDTInt function.
-246 (0xFF0A)	Timer event registration error Timer event registration has failed.	Retry. Check if the operating system is running normally. Reboot the operating system.*1
-247 (0xFF09)	Program No. error The specified program No. is out of range or invalid. SFC program No. (0 to 255), servo program No. (0 to 4095)	Correct the program No. in the user program.
-248 (0xFF08)	Number of axes error 9 or more axes was specified.	Correct the number of axes in the user program.
-249 (0xFF07)	Axis type error The specified axis type is not axis (stopped axis), synchronous encoder axis, or cam axis.	Correct the axis type in the user program.
-250 (0xFF06)	Axis No. error The specified axis No. is out of range.	Correct the axis No. in the user program.
-252 (0xFF04)	Torque limit value error The specified torque limit value is out of range.	Correct the torque limit value in the user program.
-253 (0xFF03)	Device No. error The specified device No. is out of range. Device No. in the bit device settings is not a multiple of 16.	Correct the start device No.
-254 (0xFF02)	Device type error The specified device type is invalid.	Check if the specified device type is in the device list under the "Device Type" section of the QBF function.
-255 (0xFF01)	Size error The specified start device No. plus size is out of range.	Correct the start device No. and size.
-256 (0xFF00)	Timeout error A timeout occurred while waiting for response from other CPU.	Increase the timeout value in the user program. Correct the user program (including other tasks which execute Motion CPU control/access functions) Check the program (user program, SFC program, or ladder program) of the target CPU. (e.g., Add WAIT instruction to enable execution from other CPU.)
-257 (0xFEFF)	The value specified as the interrupt type is out of range.	Review the specified value, and specify a value that is within range.
-258 (0xFEFE)	The value specified as the interrupt type is out of range.	Review the specified value, and specify a value that is within range.
-259 (0xFEFD)	No interrupt routine registration When an attempt was made to enable processing for an event (interrupt), the processing was not registered.	Register the processing for the event (interrupt) and then execute the operation once again.
-260 (0xFEFC)	CF card mount/unmount error The CF card mounting process or un-mounting process has failed.	Try to execute the operation once again. Verify that the CompactFlash card is not damaged. Replace the CompactFlash card.

* 1 In a single CPU system, reset the C Controller module.
In a multiple CPU system, reset CPU No. 1.

(3) Error codes returned by MELSEC data link functions

The following provides the error descriptions and corrective actions corresponding to individual error code returned in execution of MELSEC data link functions.
For the common error codes returned by bus interface functions and MELSEC data link functions, refer to the following.

☞ (1) in this section

Table 16.10 Error codes returned by MELSEC data link functions

Return value (HEX)	Error description	Corrective action
0	Completed normally	—
-1 (0xFFFF)	Path error The specified path is invalid.	User a path pointer returned by the mdOpen function.
-2 (0xFFFE)	Device No. error The specified device No. is out of range. Device No. in the bit device settings is not a multiple of 8. Device No. plus the number of points specified in mdRandR/mdRandREx/mdRandW/mdRandWEx function exceeds the range for the same block.	Check the start device No. of the specified device. Check the device No. plus the number of points of the same block. When specifying device No.32768(8000H) or higher, use an extended function.
-3 (0xFFFD)	Device type error The specified device type is invalid.	Check if the specified device type is in the device list.
-5 (0xFFFB)	Size error The device No. plus size exceeds the device range. Access was attempted in odd byte(s). The device No. plus size exceeds the range for the same block. The total of the specified point of each block No. of mdRandR, mdRandREx, mdRandW, mdRandWEx exceeds 10240 points.	Check the device size. Check the device No. plus size. Use the extended function since 32768(8000H) the specified device No. Use the extended function if the specified size is 32768(8000H) or more. The total of the specified point of each block No. of mdRandR, mdRandREx, mdRandW, mdRandWEx is adjusted to 10240 points or less.
-6 (0xFFFA)	No. of blocks error The number of blocks specified in dev[0] for mdRandR/mdRandREx/mdRandW/mdRandWEx is out of range.	Check the number of blocks specified in dev[0].
-8 (0xFFF8)	Channel No. error The channel No. specified in the mdOpen function is invalid.	Check the channel No.
-11 (0xFFF5)	Insufficient buffer area Read data storage size is smaller than the read data size.	Check the read data size and the read data storage size.
-12 (0xFFF4)	Block No. error The specified block No. is invalid.	Check the block No. (device type). Check if the specified device and block No. are valid in the target.
-13 (0xFFF3)	Write protect error The specified block No. of the extension file register is overlapping with the write protect area of the memory cassette.	Check the block No. (device type) of the extension file register. Check the write protect DIP switch of the accessed memory cassette.
-16 (0xFFF0)	Station No. error, Network No. error The specified station No. or network No. is out of range.	Check the station No. and network No.
-17 (0xFFEF)	All stations/group No. specification error All stations/group No. specification is not supported by the specified function.	Check if the function allows for specifying all stations/group No. Specify the device type as "no arrival confirmation", when specified station No. is "All stations" or "Group No.".
-18 (0xFFEE)	Remote operation error An inappropriate was specified.	Check the code.

(To the next page)

Table 16.10 Error codes returned by MELSEC data link functions (continued)

Return value (HEX)	Error description	Corrective action
-19 (0xFFED)	SEND/RECV channel No. error The channel No. specified in the SEND/RECV function is out of range.	Specify the channel No. within the range (1 to 8).
-31 (0xFFE1)	Module load error Loading a module required for executing the functions has failed.	Memory may be insufficient. Terminate other task(s) currently running, or decrease the access size. Check the system drive of C Controller module.
-32 (0xFFE0)	The resource is being used by other task(s) or thread(s). The resource is timed out after 30 seconds.	Retry. Memory may be insufficient. Terminate other task(s) currently running. Check if the operating system is running normally. Reboot the operating system*1
-33 (0xFFDF)	Invalid access destination error Connection settings are invalid.	Check if the connection settings are correct in the utility. Check if the target module is supported.
-34 (0xFFDE)	Registry open error Opening parameter files in the registry has failed.	Check if the connection settings are correct in the utility.
-35 (0xFFDD)	Registry read error Reading parameter files in the registry has failed.	Check if the connection settings are correct in the utility. Check if the specified channel No. is valid. Recheck the parameters in the utility, write the parameters, and then reboot the operating system.*1
-36 (0xFFDC)	Registry write error Writing parameter files to the registry has failed.	Check if the standard ROM has been shutdown. Reboot the operating system.*1
-37 (0xFFDB)	Communication initialization error Initializing communications has failed.	Retry. Memory may be insufficient. Terminate other task(s) currently running. Check available memory. Check if the operating system is running normally. Reboot the operating system.*1
-42 (0xFFD6)	Close error Communication cannot be closed.	Retry. Check if the operating system is running normally. Reboot the operating system.*1
-43 (0xFFD5)	ROM operation error A TC setting value was written to the CPU during ROM operation.	Change the TC setting value during RAM operation.

* 1 In a single CPU system, reset the C Controller module.
In a multiple CPU system, reset CPU No. 1.

16.4.2 Error codes and actions (for errors occurred in communication)

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	×	×

The C Controller module returns an error code to a request source if an error occurs at the time of a communication request from the following.

- Peripheral device
- Intelligent function module
- Network system

☒POINT

Error codes shown in this section are not for the errors that can be detected by the self-diagnostic function of the C Controller module. Therefore, none of these are stored in the special register, Diagnostic error (SD0).
When the request source is a peripheral device, a message or an error code will be shown on the peripheral device.
When the request source is an intelligent function module or a network system, an error code for the requested processing will be returned to the request source.

The following table lists the error codes detectable by the C Controller module (4000H to 4FFFH) and troubleshooting actions.

Table 16.11 Error codes that may be detected during communication

Error code (hexadecimal)	Error	Action
4001H	Execution of an unsupported request was attempted.	Check the CPU module model selected for the peripheral device.
4002H		
4010H	The requested task cannot be executed because the C Controller module is in the RUN status.	Set the C Controller module to STOP.
4030H	The specified device name is not valid.	Confirm the specified device name.
4031H	The specified device No. is out of range.	• Check the specified device No. • Check the device settings of the C Controller module.
4040H	The specified intelligent function module cannot handle the request.	Check if the specified intelligent function module has a buffer memory.
4041H	The access range exceeded the buffer memory range of the specified intelligent function module.	Check the start address and the number of access points, and make an access within the range allowed for the intelligent function module.
	The buffer memory address, to which no access is allowed, was specified.	Check the buffer memory address, and correct it to retry.
4042H	Unable to access the specified intelligent function module.	• Check if the specified intelligent function module is operating normally. • Check the specified module for hardware failure.
4043H	The specified intelligent function module does not exist.	Check the I/O number of the specified intelligent function module.

(To the next page)

Table 16.11 Error codes that may be detected during communication (continued)

Error code (hexadecimal)	Error	Action
408BH	The remote request cannot be executed.	• Set the C Controller module into the remote-request-executable state and then make the request again. • For a remote reset operation, enable the remote reset with the parameter.
414AH	In a multiple CPU system, the operation was performed for an intelligent function module that is not in the controlled group.	Retry the operation from the control CPU of the target module.
414CH	The buffer memory address, to which no access is allowed, was specified.	Check the buffer memory address, and correct it to retry.
4201H	The requested task cannot be executed because the online module change function is enabled in the multiple CPU parameter settings.	Do not make the request, or disable the online module change function and then execute the request again.
4A00H	Unable to access the specified station because no routing parameters are set for the source and relay CPU modules. In the case of a relay through a multiple CPU system, the control CPU of the relaying network module has not been activated. In a redundant system configuration, an access to another station through a network module was performed while system A or B is not defined.	• To access the specified station, set the routing parameters for the relevant stations. • Wait a little while, and retry. Or, after confirming activation of the relay system, start the communication. • In the case of a redundant system configuration, activate systems A and B normally with a tracking cable, and then start communication again.
4A01H	The network of the network No. set for the routing parameter does not exist.	Check the routing parameter set for the relevant stations, and correct it.
4A02H	Unable to access the specified station.	• Check the network module or link module for failure or offline mode. • Check the specified network No. and station No. for an error.
4B00H	Failure has occurred at the access target or a relay station. Or, the specified connection destination (I/O number of the requested module) is incorrect.	• Identify the error occurred at the specified access target or a relay station on the route to the access destination, and take action. • Check the access target specification (I/O number or station number of the request target module).
4B01H	The target is not CPU No.1 of the multiple CPU system.	Execute the request to CPU No.1 of the multiple CPU system.
4B02H	The request is not addressed to a CPU module.	Perform the operation to a module that can execute the specified function.
4B03H	The specified module version does not support the route. The communication target CPU module is not mounted.	Check if the specified route is supported.

16.5 Hardware Self-Diagnostic Function

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	○

The hardware self-diagnostics function allows the hardware check of the C Controller module.

16.5.1 Hardware self-diagnostic test and initialization setting of the Q12DCCPU-V

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	×	×

(1) Hardware self-diagnostic test and initialization setting of the Q12DCCPU-V

Starting the C Controller module in the hardware self-diagnostic mode allows the module to perform the following hardware self-diagnostic tests and initialization setting. (☞ (3) in this section)

(a) Hardware self-diagnostic test

Table 16.12 Hardware self-diagnostic test

Mode (*1)	Diagnostic item	Description
Mode 0(00)	Diagnostic test of Mode 1 to Mode 6	Conducts tests in the order from Mode 1 to Mode 6.
Mode 1(01)	ROM diagnostic test	Reads the ROM data and performs error detection.*2
	Work RAM diagnostic test	Checks test data by writing/reading/verifying them on the work RAM.
	Battery-backed-up RAM diagnostic test	Checks test data by writing/reading/verifying them on the battery-backed-up RAM.
Mode 2(02)	Ethernet diagnostic test	Conducts a loopback test of CH1 and CH2 (10BASE-T/100BASE-TX interface).
Mode 3(03)	CompactFlash card interface diagnostic test	Diagnoses the status of the CompactFlash card interface. (This test does not diagnose whether the inserted CompactFlash card is normal or not. Data are not written to the CompactFlash card.)
Mode 4(04)	RS-232 diagnostic test	Conducts a self-loopback test of CH3 (RS-232 interface).
Mode 5(05)	USB diagnostic test	Diagnoses the status of the USB interface.
Mode 6(06)	Bus diagnostic test	Writes/reads/verifies the memory of the internal bus and register.
Mode 7(07)	7-segment LED test	Conducts 7-segment LED indication tests.

* 1 The numeric value is indicated on the 7-segment LED when the mode is switched as described in the following section.

☞ Page 16-80, Section 16.5.1 (3) (c)

* 2 The following C Controller module performs sum check instead of error detection.

- The Q12DCCPU-V with serial number (first five digits) "12041" or earlier

(b) Initialization setting

Table 16.13 Initialization setting

Mode (*1)	Diagnostic item	Description
Default IP setting (10)	—	Changes the IP address of a C Controller module back to default. Execution of a script file from the standard RAM is halted. (☞ Page 6-75, Section 6.10.1 (5))
Module initialization setting (Basic mode) (11)	—	Initializes C Controller module, and switch to the basic mode. (Formats the standard RAM and parameters are set to default.) Initialization of the C Controller module is actually effected next time the C Controller is started after performing the module initialization setting (Basic mode) procedure. (☞ Page 6-77, Section 6.10.1 (6))
Module initialization setting (Extended mode) (12)*2	—	Initializes C Controller module, and switch to the extended mode. For the memory initialized when the extended mode is set, refer to the following manual. ☞ MELSEC-Q C Controller module User's Manual

* 1 The numeric value is indicated on the 7-segment LED when the mode is switched as described in the following section.
☞ Page 16-80, Section 16.5.1 (3) (c)

* 2 Can be selected for Q12DHCCPU-V with a serial number whose first five digits are "15102" or later.

(2) Before executing the hardware self-diagnostic test and initialization setting

(a) Removing modules and distribution cables

Remove all the distribution cables and modules except for the power supply module and C Controller module from the C Controller system.

Note that distribution cables must be connected for executing Mode 0, Mode 2, and Mode 4. (☞ (2) (c) in this section)

(b) Mounting on CPU slot

Mount the C Controller module on the CPU slot. Otherwise, the diagnostics and setting cannot be executed.

(c) Preparation required for each mode

Prepare as follows:

1) When executing Mode 1

Back up all data in the standard RAM and battery-backed-up RAM data.

- How to back up all data in the standard RAM

  The following is a script file example to be backed up in the CompactFlash card.

```
mkdir("/CF/BackUp")
xcopy("/RAM","/CF/BackUp")
sysLedSet(0xFF)
```

- How to back up data in the battery-backed-up RAM

  Use the QBF_ReadSRAM function for programming.

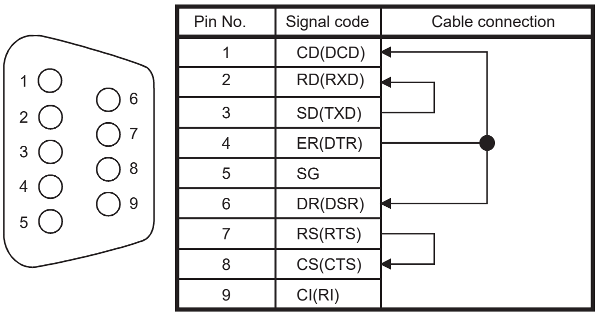
  Alternatively, store the data by clicking the Save data button on the SRAM monitoring tab of C Controller setting utility.

2) When executing Mode 2
Connect CH1 and CH2 by an Ethernet cable (crossing cable).

3) When executing Mode 3
Insert a CompactFlash card into the CompactFlash card slot. (☞ Page 6-62, Section 6.8.1 (1))

4) When executing Mode 4
Connect a cable to CH3 as shown below.

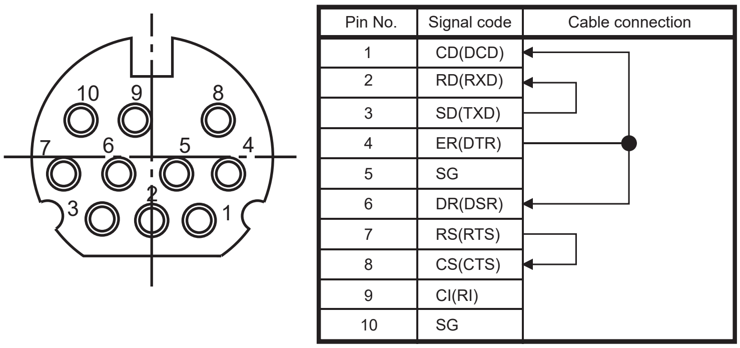
- Connecting a cable to a 9-pin D-sub connector by installing an RS232 conversion cable (Q12DCCPU-CBL) to CH3.



Pin No.	Signal code	Cable connection
1	CD(DCD)	
2	RD(RXD)	
3	SD(TXD)	
4	ER(DTR)	
5	SG	
6	DR(DSR)	
7	RS(RTS)	
8	CS(CTS)	
9	CI(RI)	

Figure 16.23 Connecting method of installing the RS232 connector conversion cable (Q12DCCPU-CBL) to CH3

- Connecting a cable directly to a round miniature connector of CH3



Pin No.	Signal code	Cable connection
1	CD(DCD)	
2	RD(RXD)	
3	SD(TXD)	
4	ER(DTR)	
5	SG	
6	DR(DSR)	
7	RS(RTS)	
8	CS(CTS)	
9	CI(RI)	
10	SG	

Figure 16.24 Connecting method of directly connecting a cable to the round miniature connector of CH3

5) When executing Mode 5
Disconnect the cable being connected to the USB interface connector, if applicable.

6) When executing Mode 7
Visually check the status of 7-segment LED indication.
(☞ (5) (c) in this section)

7) When executing Mode 0
Perform 1) to 5).

(3) Selecting a mode in the hardware self-diagnostic mode

(a) Before switching the mode to the hardware self-diagnostic mode
Be sure to perform operations described in Page 16-77, Section 16.5.1 (2).

(b) Switching to the hardware self-diagnostic mode

1) Move the RESET/SELECT switch to the RESET position, and hold the RUN/STOP/MODE switch at the MODE position.
2) Move the RESET/SELECT switch back to the center.
3) Check that the MODE LED turns on in orange.
(The 7-segment LED indicates "00".)

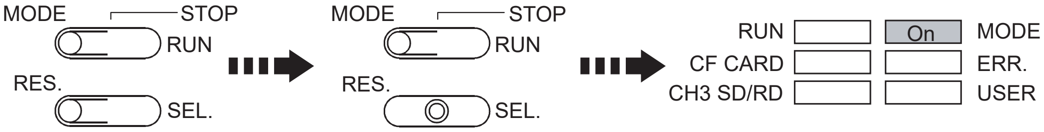


Figure 16.25 Switch operation for the hardware self-diagnostics mode

(c) Switching the mode

Set the RUN/STOP/MODE switch to "STOP", and select the mode by moving the RESET/SELECT switch to the "SELECT" position.

The mode changes from No.1→2→3...→10→1→2...as shown in the table below by every move of the switch, and the 7-segment LED indicates a numeric value corresponding to the mode.

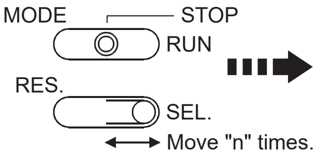


Table 16.14 Modes and corresponding LED indications

No.	Mode	7-segment LED indication
1	Mode 0 (Mode 1 to 6)	00
2	Mode 1	01
3	Mode 2	02
4	Mode 3	03
5	Mode 4	04
6	Mode 5	05
7	Mode 6	06
8	Mode 7	07
9	Default IP setting	10
10	Module initialization setting (Basic mode)	11
11[*1]	Module initialization setting (Extended mode)	12

[*1] Can be selected for Q12DHCCPU-V with a serial number whose first five digits are "15102" or later.

(4) Executing the hardware self-diagnostic test and initialization setting

☒POINT

Do not power off the C Controller system or reset the C Controller module during the self-diagnostics and setting.
Doing so may cause the C Controller module not to start normally.

Shift the RUN/STOP/MODE switch from STOP to RUN to execute the diagnostics and setting.
During execution of the diagnostics and setting, the 7-segment LED indicates the current mode, with the RUN LED flashing green.

- For mode 0, it alternately indicates "01 and the progress level(%)"*1 and then the number changes to 02, 03 ... and back to 00.
- For mode 1, the indication alternates between 01 and the progress level (%).*1

* 1 The LED of the following C Controller module indicates 01 without showing the progress level (%).
- The Q12DCCPU-V with serial number (first five digits) "12041" or earlier

(5) Confirming the hardware self-diagnostic test and initialization setting result

Confirm the diagnostics and setting result by the condition of the RUN LED (on) and ERR. LED (flashing).
For the 7-segment LED diagnostic test for Mode 7, visually check the 7-segment LED status. (☞ Page 16-83, Section 16.5.1 (5) (c))

(a) Normal completion

1) The flashing RUN LED turns off during the diagnostics and setting.

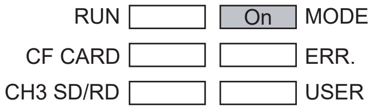


Figure 16.26 LED indication for normal completion

2) Turn the power off to finish the hardware self-diagnostic mode.

Remark

Module initialization setting (formatting of the standard RAM and setting of default parameters) is actually effected next time the C Controller is started after performing the module initialization setting procedure.

(b) Abnormal end

The ERR. LED starts flashing upon error detection during the diagnostics and setting.
The 7-segment LED indicates a numeric value corresponding to the diagnostics or setting of the error.
Upon detection of the failure, the diagnostics and setting will be terminated.

Table 16.15 LED indication for error completion (error detection)

Mode	Error detection	7-segment LED indication
Mode 1	During ROM diagnostic test	E1
	During work RAM diagnostic test	E2
	During battery-backed-up RAM diagnostic test	E3
Mode 2	During Ethernet(CH1) diagnostic test*1	E4
	During Ethernet(CH2) diagnostic test*1	E5
	During Ethernet loopback test*1	E6
Mode 3	During CompactFlash card interface diagnostic test*2	E7
Mode 4	During RS-232 diagnostic test*3	E8
Mode 5	During USB diagnostic test	E9
Mode 6	During bus diagnostic test	EA
Default IP setting	During setting the default IP	Eb
Module initialization setting	During initializing the module setting	EC

* 1 An error may occur if an Ethernet cable (crossing cable) is not connected properly.
Correctly connect and wire the cable again, and then reexecute the self-diagnostic test for Mode 2.

* 2 An error may occur if a CompactFlash card is not inserted.
Insert a CompactFlash card and then reexecute the self-diagnostic test for Mode 3.

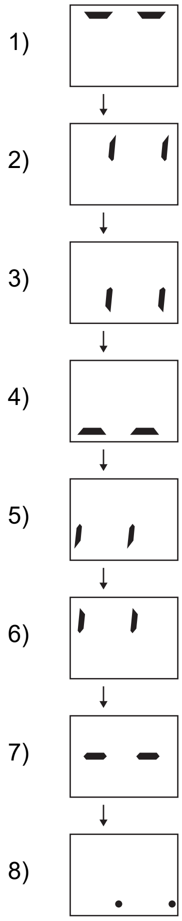
* 3 An error may occur if a self-loopback distribution cable is not connected correctly.
Correctly connect and distribute the cable again, and then reexecute the self-diagnostic test for Mode 4.

Retry the diagnostics and setting if they failed.
If failure is detected again, the hardware of the C Controller module may be faulty.
Please consult your local Mitsubishi representative, explaining the details of the problem.

(c) How to check the 7-segment LED diagnostics

To diagnose the 7-segment LED, repeatedly move the RESET/SELECT switch to the SELECT position.

The indication of the 7-segment LED changes from No.1) to 8) as shown below by every move of the switch.



Visually check that the segments of the 7-segment LED changes in the proper order.

If any of them does not lit, hardware fault of the C Controller module may be the cause.

Please consult your local Mitsubishi representative, explaining the details of the problem.

16.5.2 Hardware self-diagnostic test and initialization setting of the Q06CCPU-V(-B)

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
×	○	○

(1) Hardware self-diagnostic test and initialization setting

Starting the C Controller module in the hardware self-diagnostic mode allows the module to perform the following hardware self-diagnostic tests and initialization setting. (☞ (3) in this section)

(a) Hardware self-diagnostic test

Table 16.16 Hardware self-diagnostic test

Mode	Diagnostic item	Description
Mode 1	ROM diagnostic test	Reads the ROM data and performs a sum check.
	Work RAM diagnostic test	Checks test data by writing/reading/verifying them on the work RAM.
	Battery-backed-up RAM diagnostic test	Checks test data by writing/reading/verifying them on the battery-backed-up RAM.
	Ethernet diagnostic test	Conducts a loopback test of CH1 (10BASE-T/100BASE-TX interface).
Mode 2	CompactFlash card interface diagnostic test	For the Q06CCPU-V, diagnoses the status of the CompactFlash card interface. (This test does not diagnose whether the inserted CompactFlash card is normal or not. Data are not written to the CompactFlash card.)
		For the Q06CCPU-V-B, the test completes normally without diagnosing the CompactFlash card interface.
Mode 3	RS-232 diagnostic test	Conducts the self-loopback test of CH2 (RS-232 interface).
Mode 1+2+3	Diagnostic tests of Mode 1 to Mode 3	Conducts a diagnostic test in the order from Mode 1 to Mode 3.

(b) Initialization setting

Table 16.17 Initialization setting

Mode	Diagnostic item	Description
Default IP setting	—	Changes the IP address of a C Controller module back to default. Execution of a script file from the standard ROM is halted. (☞ Page 6-84, Section 6.10.2 (3))
Standard ROM initialization setting	—	Formats the standard ROM of a C Controller module. Formatting of the standard ROM is actually effected next time the C Controller is started after performing the module initialization setting procedure. (☞ Page 6-85, Section 6.10.2 (4))

(2) Before executing the hardware self-diagnostic test and initialization setting

(a) Removing modules and distribution cables
Remove all the distribution cables and modules except for the power supply module and C Controller module from the C Controller system.
Note that distribution cables must be connected for executing Mode 3 (RS-232 diagnostic test).

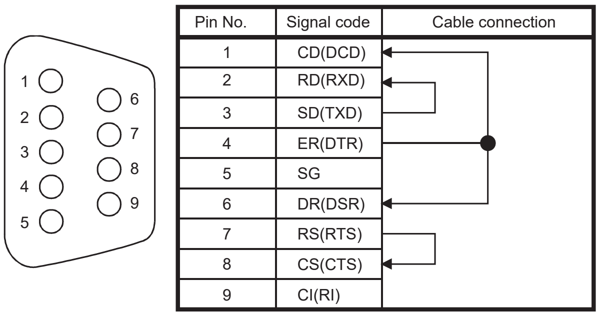
(b) Mounting on CPU slot
Mount the C Controller module on the CPU slot. Otherwise, the diagnostics and setting cannot be executed.

(c) Preparation required for each mode
Prepare as follows:

1) When executing Mode 1
Disconnect the cable from CH1, if connected.

2) When executing Mode 2
For the Q06CCPU-V, insert a CompactFlash card into the CompactFlash card slot. (☞ Page 6-62, Section 6.8.1 (1))
For the Q06CCPU-V-B, no preparation is necessary.

3) When executing Mode 3
Connect a cable to CH2 as shown below.



Pin No.	Signal code	Cable connection
1	CD(DCD)	
2	RD(RXD)	
3	SD(TXD)	
4	ER(DTR)	
5	SG	
6	DR(DSR)	
7	RS(RTS)	
8	CS(CTS)	
9	CI(RI)	

Figure 16.27 Cable connection for executing Mode 3

4) When executing Mode 1 + 2 + 3
Perform 1) to 3).

(3) Selecting a mode in the hardware self-diagnostic mode

(a) Before switching the mode to the hardware self-diagnostic mode
Be sure to perform operations described in Page 16-85, Section 16.5.2 (2).

(b) Switching to the hardware self-diagnostic mode

1) Move the RESET/SELECT switch to the RESET position, and hold the RUN/STOP/MODE switch at the MODE position.

2) Move the RESET/SELECT switch back to the center.

3) Check that the MODE LED turns on in orange.

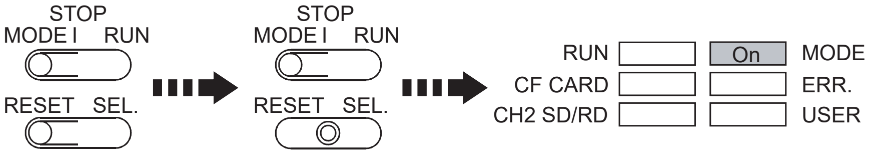


Figure 16.28 Switch operation for the hardware self-diagnostics mode (for the Q06CCPU-V)

(c) Switching the mode

Set the RUN/STOP/MODE switch to "STOP", and select the mode by moving the RESET/SELECT switch to the "SELECT" position.

- Mode 1 : Move the RESET/SELECT switch to the SELECT position once. The RUN LED turns on.
- Mode 2 : Move the RESET/SELECT switch to the SELECT position twice.
  For the Q06CCPU-V, the CF CARD LED turns on.
  For the Q06CCPU-V-B, the STS LED turns on.
- Mode 3 : Move the RESET/SELECT switch to the SELECT position three times.
  The CH2 SD/RD LED turns on.
- Mode 1 + 2 + 3 : Move the RESET/SELECT switch to the SELECT position four times.
  For the Q06CCPU-V, the RUN LED, CF CARD LED, and CH2 SD/RD LED turn on.
  For the Q06CCPU-V-B, the RUN LED, STS LED, and CH2 SD/RD LED turn on.
- Default IP setting : Move the RESET/SELECT switch to the SELECT position five times.
  The USER LED turns on.
- Standard ROM format setting:Move the RESET/SELECT switch to the SELECT position six times.
  The RUN LED and USER LED turn on.

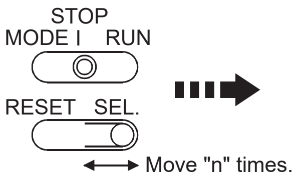


Table 16.18 Modes and corresponding LED indications (for the Q06CCPU-V)

No.	Mode	LED indication
1	Mode 1	RUN On On MODE CF CARD □ □ ERR. CH2 SD/RD □ □ USER
2	Mode 2	RUN □ On MODE CF CARD On □ ERR. CH2 SD/RD □ □ USER
3	Mode 3	RUN □ On MODE CF CARD □ □ ERR. CH2 SD/RD On □ USER
4	Mode 1+2+3	RUN On On MODE CF CARD On □ ERR. CH2 SD/RD On □ USER
5	Default IP setting	RUN □ On MODE CF CARD □ □ ERR. CH2 SD/RD □ On USER
6	Standard ROM format setting	RUN On On MODE CF CARD □ □ ERR. CH2 SD/RD □ On USER

(4) Executing the hardware self-diagnostic test and initialization setting

☒POINT

Do not power off the C Controller system or reset the C Controller module during the self-diagnostics and setting.
Doing so may cause the C Controller module not to start normally.

Shift the RUN/STOP/MODE switch from STOP to RUN to execute the diagnostics and setting.

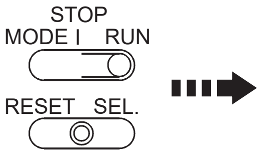


Table 16.19 LED indication during execution (for the Q06CCPU-V)

No.	Diagnosis	LED indication
1	Mode 1 in execution	RUN Flash On MODE CF CARD □ □ ERR. CH2 SD/RD □ □ USER
2	Mode 2 in execution	RUN □ On MODE CF CARD Flash □ ERR. CH2 SD/RD □ □ USER
3	Mode 3 in execution	RUN □ On MODE CF CARD □ □ ERR. CH2 SD/RD Flash □ USER
4	Mode 1 + 2 + 3 in execution*1	RUN Flash On MODE CF CARD On □ ERR. CH2 SD/RD On □ USER
5	Default IP setting in execution	RUN □ On MODE CF CARD □ □ ERR. CH2 SD/RD □ Flash USER
6	Standard ROM format setting in execution	RUN Flash On MODE CF CARD □ □ ERR. CH2 SD/RD □ Flash USER

* 1 The LED corresponding to the currently executing mode flashes.

(5) Confirming the hardware self-diagnostic test and initialization setting result

(a) Normal completion

1) The flashing LED turns off during the diagnostics and setting.

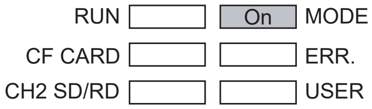


Figure 16.29 LED indication for normal completion (for the Q06CCPU-V)

2) Turn the power off to finish the hardware self-diagnostic mode.

Remark

Standard ROM format setting (formatting of the standard ROM) takes effect next time the C Controller module is started after the setting.
When the setting is completed normally, the RUN and USER LEDs are turned off and the MODE LED starts flashing green.

(b) Abnormal end

The ERR. LED starts flashing upon error detection during the diagnostics and setting, and the LEDs turn on corresponding to the diagnostics or setting of the error.

Upon detection of the failure, the diagnostics and setting will be terminated.

Table 16.20 LED indication for error completion (error detection) (for the Q06CCPU-V)

No.	Error detection	LED indication
1	During ROM diagnostic test	RUN On On MODE CF CARD □ Flash ERR. CH2 SD/RD □ □ USER
2	During work RAM diagnostic test	RUN □ On MODE CF CARD On Flash ERR. CH2 SD/RD □ □ USER
3	During battery-backed-up RAM diagnostic test	RUN On On MODE CF CARD On Flash ERR. CH2 SD/RD □ □ USER
4	During Ethernet diagnostic test	RUN □ On MODE CF CARD □ Flash ERR. CH2 SD/RD On □ USER
5	During CompactFlash card interface diagnostic test*1	RUN On On MODE CF CARD □ Flash ERR. CH2 SD/RD On □ USER
6	During RS-232 diagnostic test*2	RUN □ On MODE CF CARD On Flash ERR. CH2 SD/RD On □ USER
7	Default IP setting in execution	RUN □ On MODE CF CARD □ Flash ERR. CH2 SD/RD □ On USER
8	Standard ROM format setting in execution	RUN On On MODE CF CARD □ Flash ERR. CH2 SD/RD □ On USER

* 1 An error may occur if a CompactFlash card is not inserted.
Insert a CompactFlash card and then reexecute the self-diagnostic test for Mode 2.

* 2 An error may occur if a self-loopback distribution cable is not connected correctly.
Correctly connect and distribute the cable again, and then reexecute the self-diagnostic test for Mode 3.

Retry the diagnostics and setting if it failed.
If the error recurs, hardware fault of the C Controller module may be the cause.
Please consult your local Mitsubishi representative, explaining the details of the problem.

16.6 Diagnostics and Restoration of the Standard RAM, Standard ROM, and CompactFlash Card Drives

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	○

An error may occur in the file system if any of the following drives is incorrectly used.

- Standard RAM drive
- Standard ROM drive
- CompactFlash card drive

The following describes how to diagnose and restore the system when an error has occurred.

(1) How to diagnose the FAT file system of the drive

The file system of the following drives use the FAT format.

- Standard RAM drive
- Standard ROM drive
- CompactFlash card drive

To diagnose the FAT format structure, use the following tools.

(a) VxWorks-standard "chkdsk" command
Execute the chkdsk command in Shell, in Telnet tool, or in the user program.
(☞ (3) in this section)

(b) "chkdsk" command of a personal computer (Windows®) (only for CompactFlash cards)
For the CompactFlash card, errors can be checked by the chkdsk command of a personal computer (Windows®).

(2) How to restore the FAT file system of the drive

If an error is detected in the file system, the FAT format can be restored by either of the following methods.

- Use the restore function of the chkdsk command (☞ (3) in this section)
- Reformat by using the format command (☞ Page 6-85, Section 6.10.2)

If the file system is not restored or the standard RAM drive and standard ROM drive cannot be formatted by either of the above methods, please consult your local Mitsubishi service center or representative, explaining the details of the problem.

☒POINT

1. During restoring the FAT file system, the stored file may be erased and some of them may be corrupted.
   Back up the necessary program files and data before restoration, and write the backed up files and data again after restoration.
2. When the file system restoration has been performed for the standard RAM or standard ROM, write parameters by using C Controller setting utility.

(3) chkdsk command

The following describes the chkdsk command of VxWorks.

(a) Format of the chkdsk command

chkdsk, "Device name", and Flag

Table 16.21 Input values and settings of the chkdsk command

Argument	Input value	Setting
Device name	/RAM	Standard RAM
	/ROM	Standard ROM
	/CF	CompactFlash card
Flag	1	Diagnostics
	2	Restoration

(b) Execution format of the chkdsk command

Execute the chkdsk command in the format where the task is started by the sp command.

```
sp(chkdsk, "Device name", Flag) [Enter]
```

Execute the chkdsk command from either of the following tools.

- Workbench Shell or Tornado Shell
- Telnet

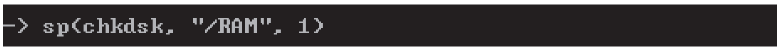
(c) Execution example of the chkdsk command

The following explains how to diagnose the standard RAM of the Q12DCCPU-V by using the chkdsk command from the Telnet tool.

This operation procedure can be applied to diagnostics and restoration of the standard ROM or a CompactFlash card.

1) Execute the chkdsk command. (☞ (3) (b) in this section)

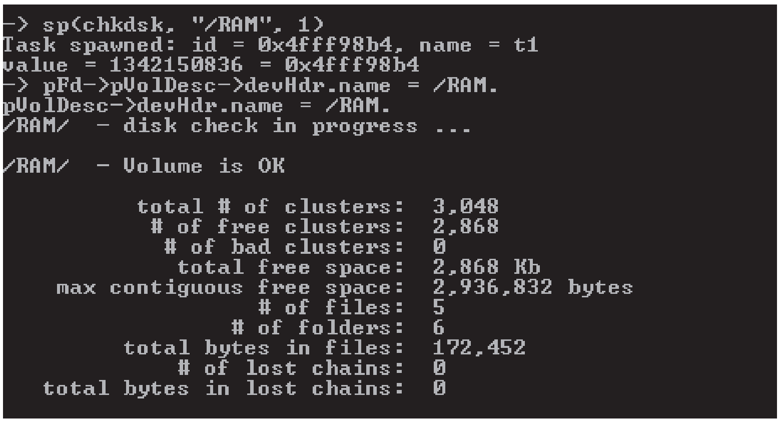
Enter sp(chkdsk, "/RAM", 1), and press the Enter key.



```
-> sp(chkdsk, "/RAM", 1)
```

Figure 16.30 Executing the chkdsk command (for Telnet tool)

2) The following screen appears if the execution is completed normally.



```
-> sp(chkdsk, "/RAM", 1)
Task spawned: id = 0x4fff98b4, name = t1
value = 1342150836 = 0x4fff98b4
-> pFd->pVolDesc->devHdr.name = /RAM.
pVolDesc->devHdr.name = /RAM.
/RAM/  - disk check in progress ...

/RAM/  - Volume is OK

          total # of clusters:  3,048
           # of free clusters:  2,868
            # of bad clusters:  0
             total free space:  2,868 Kb
    max contiguous free space:  2,936,832 bytes
                   # of files:  5
                 # of folders:  6
         total bytes in files:  172,452
             # of lost chains:  0
   total bytes in lost chains:  0
```

Figure 16.31 Execution result of the chkdsk command (normal completion)

Remark

For details, refer to the manual for VxWorks.

APPENDICES

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	○

Appendix 1 Function Processing Time

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	○

This chapter explains the time required for processing functions when one user program created using bus interface functions and/or MELSEC data link functions is executed with the task priority set to 100.
For the Q06CCPU-V-B, MELSEC data link functions cannot be used.

☒POINT

Processing time for the bus interface functions and MELSEC data link functions vary somewhat depending on the operation conditions of the OS, execution condition of the user program, and positions and types of the modules mounted on the base unit.
The access time data shown in the following tables are typical processing time values for respective functions.

(1) Bus interface function processing time

(a) I/O access time

1) When accessing an input module QX42 (Start input number: X0) in a single CPU system

Table APPX.1 Access time

Function		Communication size	Access time		
			Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
QBF_X_In_BitEx	High-speed access	1 bit	7μs	5μs	5μs
	Normal access	1 bit	8μs	6μs	6μs
QBF_X_In_WordEx	High-speed access	1 word	7μs	5μs	5μs
		16 words	25s	23μs	25μs
		64 words	83μs	82μs	82μs
	Normal access	1 word	8μs	6μs	6μs
		16 words	27μs	25μs	25μs
		64 words	89μs	86μs	85μs

2) When accessing an output module QY42P (Start output number: Y0) in a single CPU system

Table APPX.2 Access time

Function		Communication size	Access time		
			Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
QBF_Y_Out_BitEx	High-speed access	1 bit	6μs	5μs	4μs
	Normal access	1 bit	7μs	6μs	5μs
QBF_Y_Out_WordEx	High-speed access	1 word	6μs	5μs	4μs
		16 words	18μs	22μs	22μs
		64 words	70μs	76μs	75μs
	Normal access	1 word	7μs	6μs	5μs
		16 words	18μs	22μs	22μs
		64 words	71μs	79μs	79μs

(b) Buffer memory access time

1) When accessing the QJ61BT11N (Start I/O number: 0) in a single CPU system, where the block data assurance per station function is disabled by CC-Link utility

Table APPX.3 Access time

Function	Communication size	Access time		
		Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
QBF_FromBuf	1 word	15 µs	16 µs	16 µs
	64 words	40 µs	42 µs	41 µs
	512 words	214 µs	222 µs	233 µs
QBF_ToBuf	1 word	15 µs	17 µs	17 µs
	64 words	39 µs	41 µs	43 µs
	512 words	206 µs	212 µs	222 µs

2) When accessing the QJ61BT11N (Start I/O number: 0) in a single CPU system, where the block data assurance per station function is enabled by CC-Link utility

- Accessing the buffer memory that does not support the block data assurance per station function

Table APPX.4 Access time

Function	Communication size	Access time		
		Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
QBF_FromBuf	1 word	16 µs	17 µs	19 µs
	64 words	41 µs	44 µs	45 µs
	512 words	217 µs	223 µs	234 µs
QBF_ToBuf	1 word	16 µs	19 µs	20 µs
	64 words	41 µs	43 µs	47 µs
	512 words	208 µs	215 µs	227 µs

- Accessing the buffer memory that supports the block data assurance per station function

Table APPX.5 Access time

Function	CC-Link refresh method	Communication size	Access time		
			Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
QBF_FromBuf	Automatic	1 word	*1	*1	*1
		64 words			
		512 words			
	Manual	1 word	41 µs	53 µs	54 µs
		64 words	69 µs	80 µs	80 µs
		512 words	245 µs	259 µs	272 µs
QBF_ToBuf	Automatic	1 word	*1	*1	*1
		64 words			
		512 words			
	Manual	1 word	26 µs	34 µs	39 µs
		64 words	52 µs	59 µs	65 µs
		512 words	221 µs	231 µs	246 µs

* 1 The access time significantly changes depending on the timing of access start and link scan.
Minimum: Access time shown in Table APPX.4.
Maximum: Access time shown in Table APPX.4 + Link scan time

3) When accessing an intelligent function module other than the QJ61BT11N in a single CPU system

Table APPX.6 Access time

Function	Communication size	Access time		
		Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
QBF_FromBuf	1 word	14 µs	16 µs	16 µs
	64 words	40 µs	42 µs	41 µs
	512 words	214 µs	222 µs	233 µs
QBF_ToBuf	1 word	15 µs	17 µs	17 µs
	64 words	39 µs	41 µs	43 µs
	512 words	204 µs	212 µs	222 µs

(c) Multiple CPU shared memory access time

1) When accessing the CPU shared memory (auto refresh area or user setting area) of the host CPU in a multiple CPU system (CPU No.1: Q25HCPU, CPU No.2: C Controller module)

Table APPX.7 Access time

Function	Communication size	Access time		
		Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
QBF_FromBuf	1 word	10 µs	13 µs	13 µs
	64 words	18 µs	20 µs	26 µs
	512 words	76 µs	73 µs	125 µs
QBF_ToBuf	1 word	11 µs	14 µs	19 µs
	64 words	16 µs	18 µs	23 µs
	512 words	65 µs	50 µs	57 µs

2) When accessing the CPU shared memory (auto refresh area or user setting area) of CPU No.1 in a multiple CPU system (CPU No.1: Q25HCPU, CPU No.2: C Controller module)

Table APPX.8 Access time

Function	Communication size	Access time		
		Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
QBF_FromBuf	1 word	11 µs	8 µs	9 µs
	64 words	39 µs	38 µs	38 µs
	512 words	222 µs	226 µs	217 µs
QBF_ToBuf	1 word	—	—	—
	64 words	—	—	—
	512 words	—	—	—

3) When accessing the CPU shared memory (multiple CPU high speed transmission area) of the host CPU in a multiple CPU system (CPU No.1: Q26UDHCPU, CPU No.2: C Controller module)

Table APPX.9 Access time

Function	Communication size	Access time		
		Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
QBF_FromBuf	1 word	7 μs	—	—
	64 words	18 μs	—	—
	512 words	94 μs	—	—
QBF_ToBuf	1 word	7 μs	—	—
	64 words	13 μs	—	—
	512 words	68 μs	—	—

4) When accessing the CPU shared memory (multiple CPU high speed transmission area) of CPU No.1 in a multiple CPU system (CPU No.1: Q26UDHCPU, CPU No.2: C Controller module)

Table APPX.10 Access time

Function	Communication size	Access time		
		Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
QBF_FromBuf	1 word	7 μs	—	—
	64 words	18 μs	—	—
	512 words	94 μs	—	—
QBF_ToBuf	1 word	—	—	—
	64 words	—	—	—
	512 words	—	—	—

(2) MELSEC data link function processing time

The MELSEC data link function processing time varies depending on the target module, and the route and devices used for communication.

When the communication target is a programmable controller CPU, the operating status, sequence scan time, and link scan time of the programmable controller CPU may also affect the processing time.

For further details of the MELSEC data link functions, refer to the following.

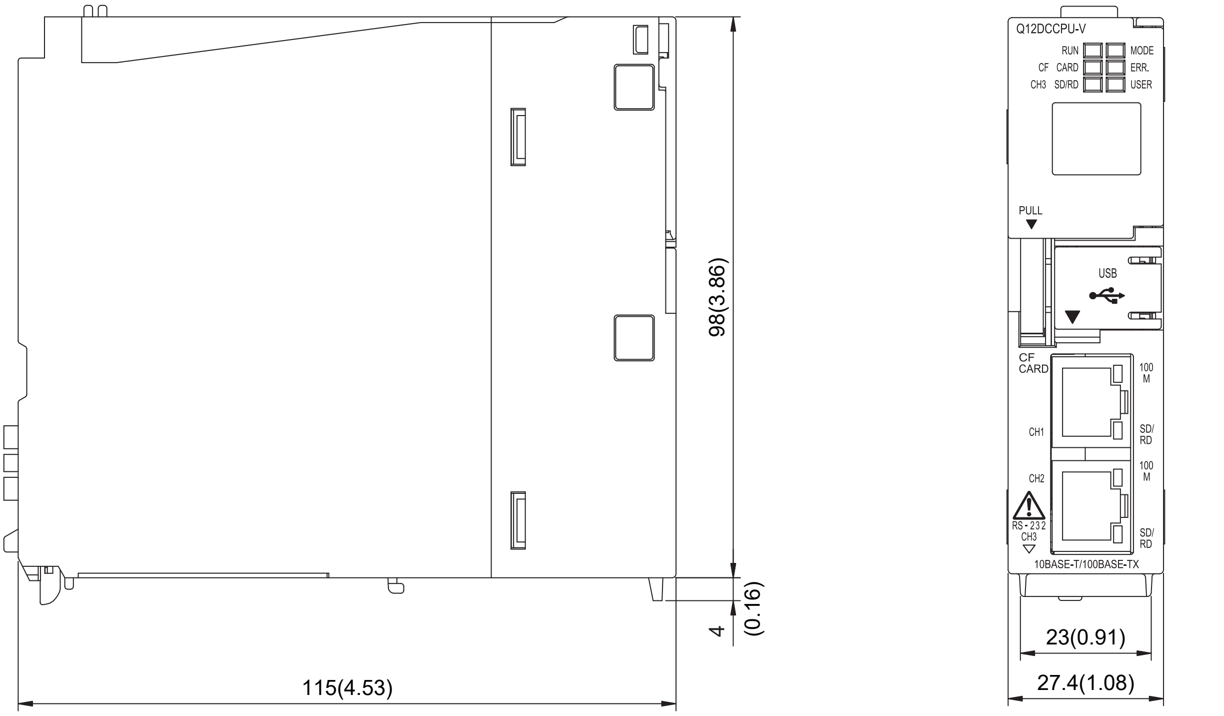
☞C Controller Module User's Manual (Utility Operation, Programming)

Appendix 2 External Dimensions

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	○

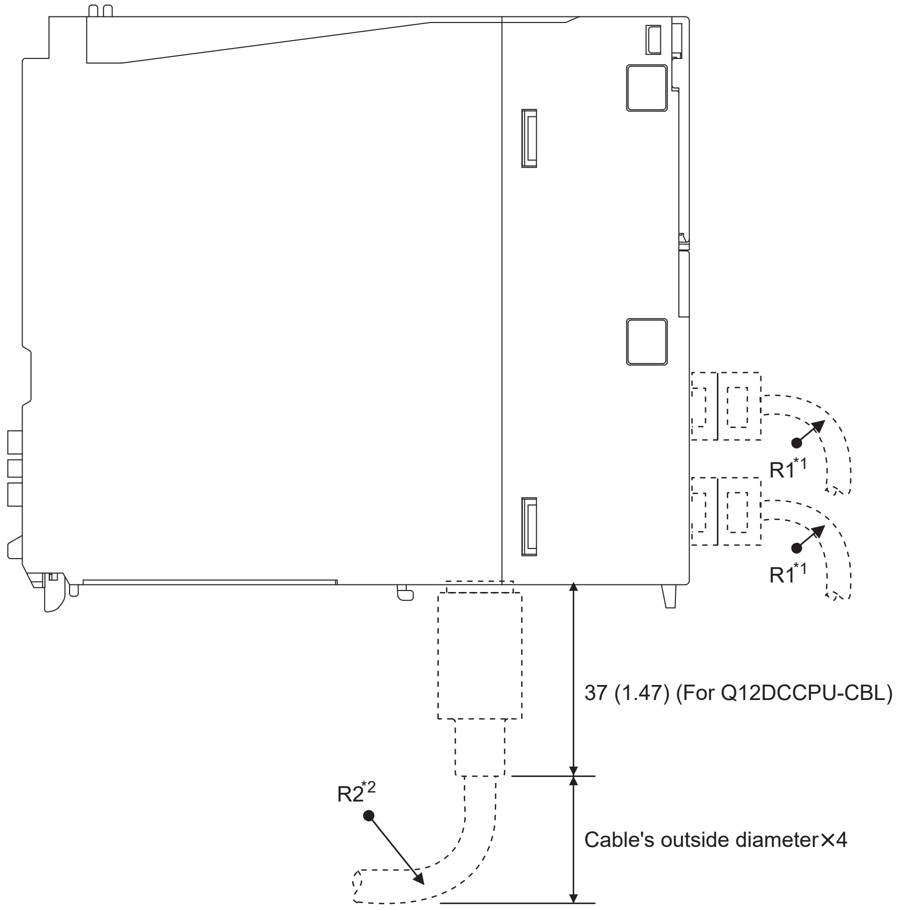
Appendix 2.1 Q12DCCPU-V

(1) Q12DCCPU-V



(Unit: mm (inches))

Figure APPX.1 External dimensions



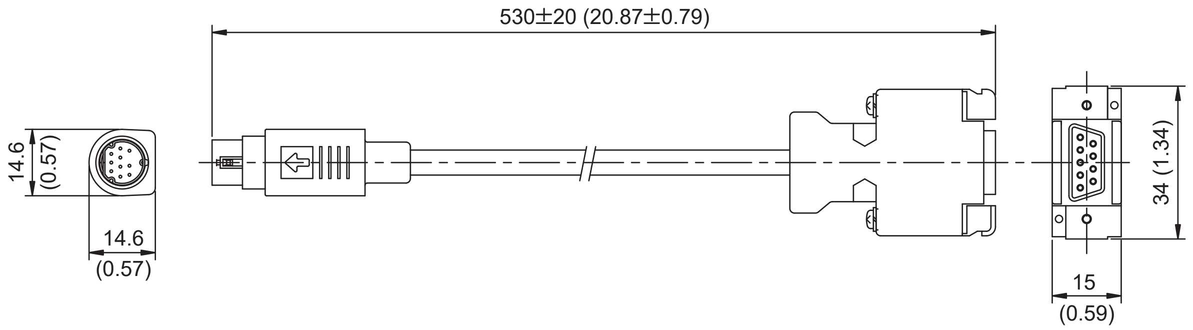
(Unit: mm (inches))

Figure APPX.1 External dimensions (continued)

* 1 The bend radius (R1: reference value) near the connector must be "cable's outside diameter × 4" or more when a twisted pair cable is used.

* 2 The bend radius (R2: reference value) near the connector must be "cable's outside diameter × 4" or more when an RS-232 cable is used.
For the Q12DCCPU-CBL, "cable's outside diameter × 4" is 22 mm (0.87 inches).

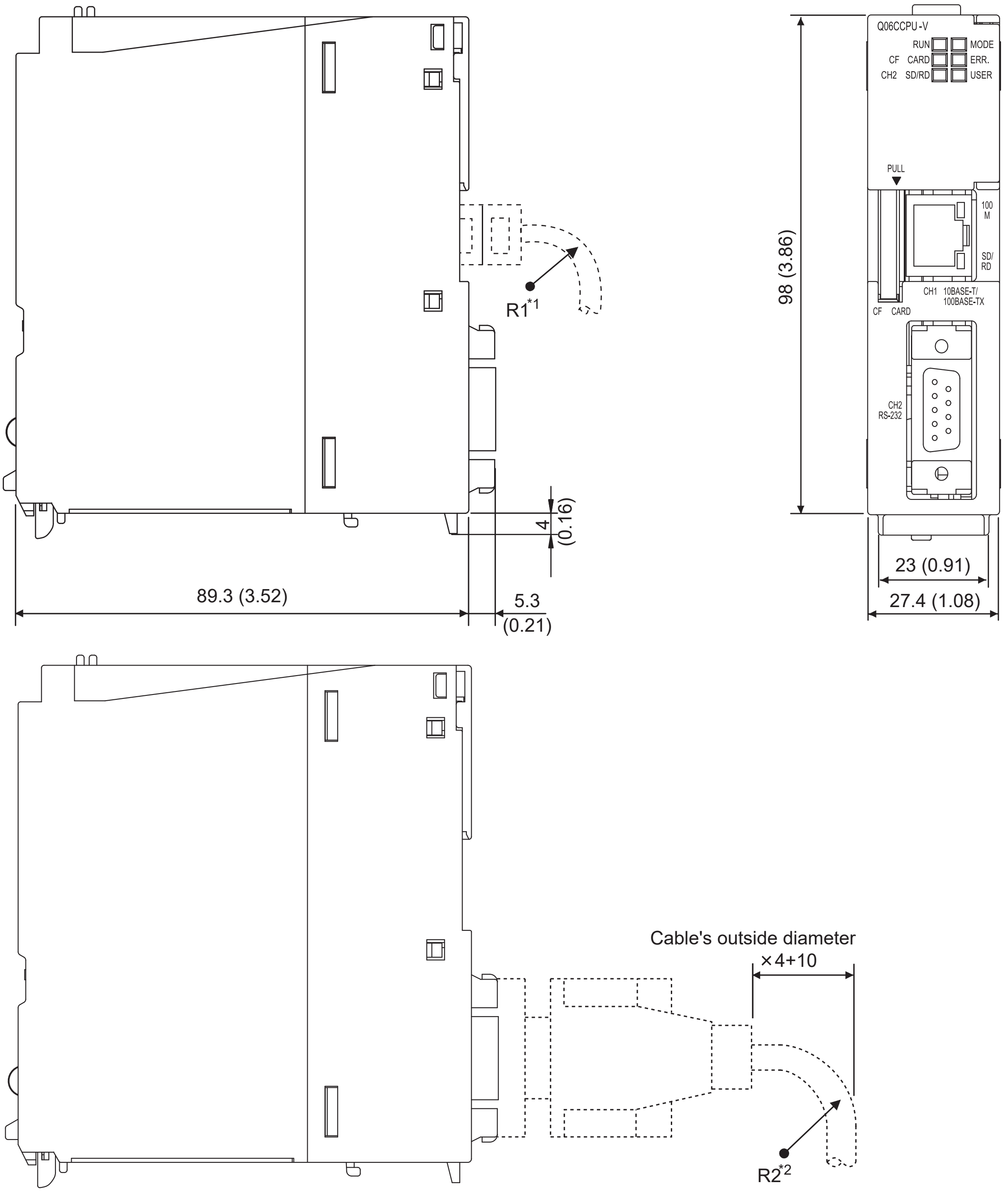
(2) Q12DCCPU-CBL



(Unit: mm (inches))

Figure APPX.2 External dimensions

Appendix 2.2 Q06CCPU-V



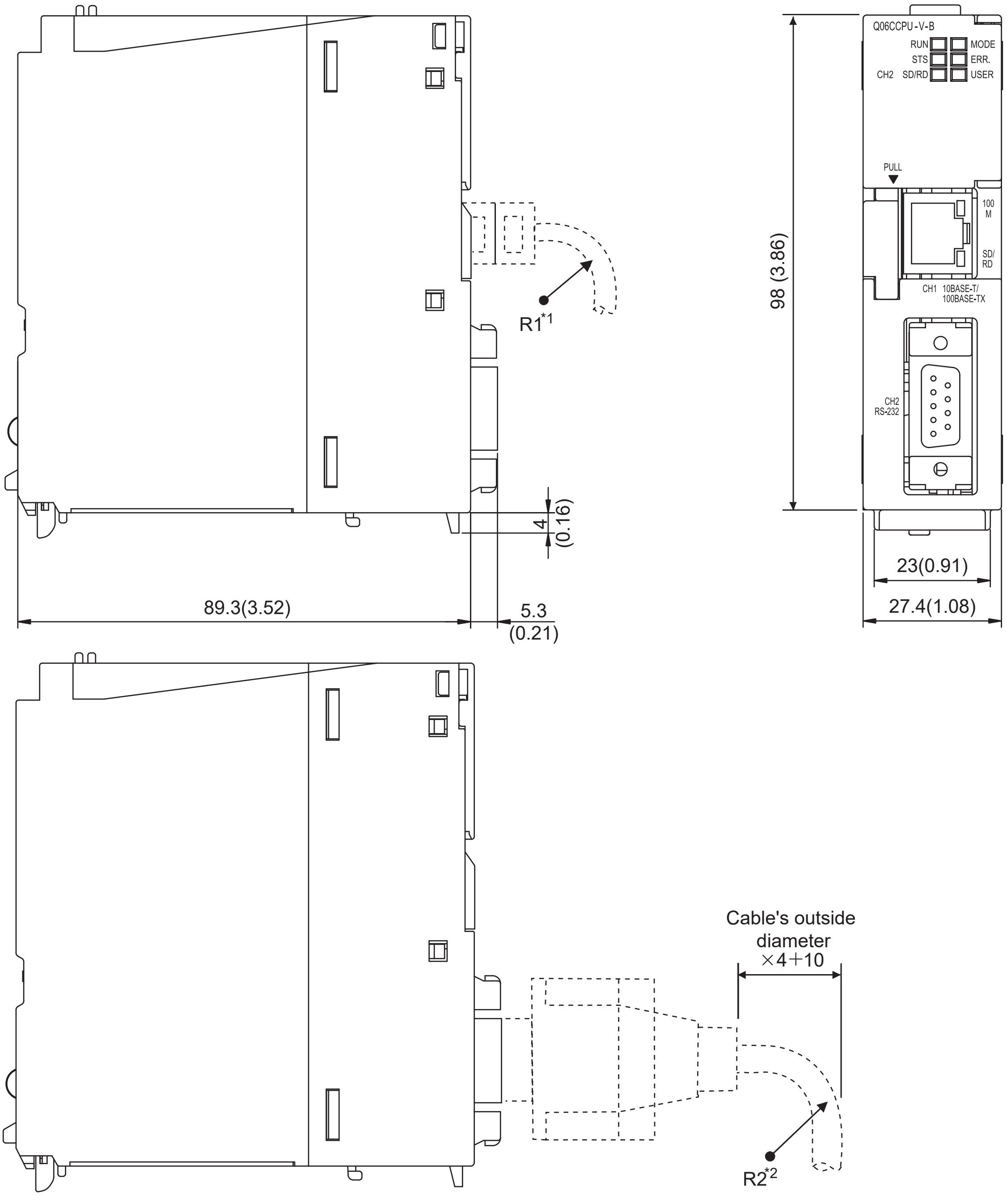
(Unit: mm (inches))

Figure APPX.3 External dimensions

* 1 The bend radius (R1: reference value) near the connector must be "cable's outside diameter × 4" or more when a twisted pair cable is used.

* 2 The bend radius (R2: reference value) near the connector must be "cable's outside diameter × 4" or more when an RS-232 cable is used.

Appendix 2.3 Q06CCPU-V-B



(Unit: mm (inches))

Figure APPX.4 External dimensions

* 1 The bend radius (R1: reference value) near the connector must be "cable's outside diameter × 4" or more when a twisted pair cable is used.

* 2 The bend radius (R2: reference value) near the connector must be "cable's outside diameter × 4" or more when an RS-232 cable is used.

Appendix 3 Transportation Precautions

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	○

When transporting lithium batteries, follow the transportation regulations.

Appendix 3.1 Regulated models

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	○

Lithium batteries for the C Controller module are classified as shown in the table below.

Table APPX.11 Regulated models

Product name	Model	Product supply status	Treated as
Battery for Q series	Q6BAT	Lithium battery	Non-dangerous goods
	Q7BAT	Lithium battery	Dangerous goods

Appendix 3.2 Handling for transportation

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	○

Our products are shipped in packages that meet transportation regulations. When any of these products are needed to be transported after repacking or unpacking, compliance with the IATA Dangerous Goods Regulations, IMDG Code, and national transportation regulations must be ensured.
For details, please consult the shipping carrier.

Appendix 4 Handling of Batteries and Devices with Built-in Batteries in EU Member States

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	○

This section describes the precautions for disposing of waste batteries in EU member status and exporting batteries and/or devices with built-in batteries to EU member states.

Appendix 4.1 Disposal precautions

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	○

In EU member states, there is a separate collection system for waste batteries. Dispose of batteries properly at the local community waste collection/recycling center.

The symbol shown in Figure APPX.5 is printed on the batteries and packaging of batteries and devices with built-in batteries.

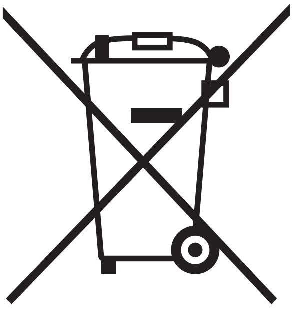


Figure APPX.5 Symbol

Note: This symbol is for EU member states only.
The symbol is specified in the new EU Battery Directive (2006/66/EC) Article 20 "Information for end-users" and Annex II.

The symbol indicates that batteries need to be disposed of separately from other wastes.

Appendix 4.2 Exportation precautions

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	○

The new EU Battery Directive (2006/66/EC) requires the following when marketing or exporting batteries and/or devices with built-in batteries to EU member states.

- To print the symbol on batteries, devices, or their packaging
- To explain the symbol in the manuals of the products

(1) Labelling

To market or export batteries and/or devices with built-in batteries, which have no symbol, to EU member states on September 26, 2008 or later, print the symbol shown in Figure APPX.5 on the batteries, devices, or their packaging.

(2) Explaining the symbol in the manual

To export devices incorporating Mitsubishi programmable controller to EU member states on September 26, 2008 or later, provide the latest manuals that include the explanation of the symbol.

If no Mitsubishi manuals or any old manuals without the explanation of the symbol are provided, separately attach an explanatory note regarding the symbol to each manual of the devices.

☒POINT

The requirements apply to batteries and/or devices with built-in batteries manufactured before the enforcement date of the new EU Battery Directive (2006/66/EC).

Appendix 5 Characters Applicable to User Names and Passwords

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	○

This section explains the characters applicable to user names and passwords set on each utility.

(1) Characters applicable to user names

Alphanumeric characters can be used, and a space, "(quotation mark), and :(colon) are not applicable.

The following ASCII code table shows applicable characters.

Table APPX.12 Characters applicable to user names

		Higher 4 bits															
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Lower 4 bits	0	NUL		(SP)	0	@	P	`	p								
	1			!	1	A	Q	a	q								
	2			”	2	B	R	b	r								
	3			#	3	C	S	c	s								
	4			$	4	D	T	d	t								
	5			%	5	E	U	e	u								
	6			&	6	F	V	f	v								
	7			’	7	G	W	g	w								
	8			(	8	H	X	h	x								
	9			)	9	I	Y	i	y								
	A			*	:	J	Z	j	z								
	B			+	;	K	[	k	{								
	C			,	<	L	¥	l	|								
	D			-	=	M	]	m	}								
	E			.	>	N	^	n	‾								
	F			/	?	O	_	o									

▨ : Applicable

□ : Not applicable

(2) Characters applicable to passwords

Alphanumeric characters can be used, and a space, "(quotation mark), and @(at mark) are not applicable.

The following ASCII code table shows applicable characters.

Table APPX.13 Characters applicable to passwords

		Higher 4 bits															
		0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
Lower 4 bits	0	NUL		(SP)	0	@	P	`	p								
	1			!	1	A	Q	a	q								
	2			”	2	B	R	b	r								
	3			#	3	C	S	c	s								
	4			$	4	D	T	d	t								
	5			%	5	E	U	e	u								
	6			&	6	F	V	f	v								
	7			’	7	G	W	g	w								
	8			(	8	H	X	h	x								
	9			)	9	I	Y	i	y								
	A			*	:	J	Z	j	z								
	B			+	;	K	[	k	{								
	C			,	<	L	¥	l	|								
	D			-	=	M	]	m	}								
	E			.	>	N	^	n	‾								
	F			/	?	O	_	o									

▨ : Applicable

□ : Not applicable

Appendix 6 List of Special Relays

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	×	×

The special relay (SM) is an internal relay whose application is fixed in a C Controller module.
For this reason, the special relay cannot be used in the same way as other internal relays are used in user programs.

The following table shows how to read each item in the list.

Table APPX.14 How to read the list of special relays

Item	Description
Number	• Indicates the number of the special relay.
Name	• Indicates the name of the special relay.
Description	• Indicates descriptions about the special relay.
Detailed description	• Indicates detailed descriptions about the special relay.
Setting side (setting time)	• Explains setting side and setting time at the system side. <Setting side> S : Set at the system side. <Setting time> Error occurrence : Set when an error occurs.

(1) Diagnostic information

Table APPX.15 Special relay

Number	Name	Description	Detailed description	Setting side (setting time)
SM0	Diagnostic error	OFF: No error ON: Error	• Turns on when a diagnostic result error occurs. • Later, on status is held even after returning to normal.	S (Error)
SM1	Self-diagnostic error	OFF: No self-diagnostic error ON: Self-diagnostic error	• Turns on when a diagnostic result error occurs. • Later, on status is held even after returning to normal.	S (Error)
SM5	Common error information	OFF: No common error information ON: Common error information	• When SM0 turns on, the relay turns on if there is common error information.	S (Error)
SM16	Error individual information	OFF: No error individual information ON: Error individual information	• When SM0 turns on, the relay turns on if there is error individual information.	S (Error)
SM60	Fuse blown detection	OFF: Normal ON : Fuse blown module	• Turns on when more than one output module in fuse blown status exist. • Later, on status is held even after returning to normal.	S (Error)
SM61	I/O module verification error	OFF: Normal ON: Error	• Turns on when the I/O module is different from the status that was registered upon power on. • Later, on status is held even after returning to normal.	S (Error)

Appendix 7 List of Special Registers

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	×	×

The special register (SD) is an internal register whose application is fixed in a C Controller module.

For this reason, the special relay cannot be used in the same way as other internal relays are used in user programs.

Data is stored in binary format unless otherwise specified.

The following table shows how to read each item on the list.

Table APPX.16 How to read the list of special registers

Item	Description
Number	• Indicates the number of the special register.
Name	• Indicates the name of the special register.
Description	• Indicates descriptions about the special register.
Detailed description	• Indicates detailed descriptions about the special register.
Setting side (setting time)	• Explains setting side and setting time at the system side. <Setting side> S : Set at the system side. <Setting time> System : Set when a fixed scan or the status is changed at the system side. Initial : Set only at initialization (power on, STOP RUN, etc.) Error : Set when an error occurs.

(1) Diagnostic information

Table APPX.17 Special registers

Number	Name	Description	Detailed description	Setting side (setting time)
SD0	Diagnostic error	Diagnostic error code	• The error code occurred in diagnosis in diagnosis is stored as a binary code. • The information is the same as the latest error code.	S (Error)
SD1	Date and time of diagnostic error	Date and time of diagnostic error	• The year (the last two digits of the year) and month when SD0 data was updated are stored as a 2-digit BCD code. b15 to b8 b7 to b0 YY (0 to 99) | MM (1 to 12) (Example) October, 1995 H9510	S (Error)
SD2			• The date and time when SD0 data was updated are stored as a 2-digit BCD code. b15 to b8 b7 to b0 DD (1 to 31) | Hr. (0 to 23) (Example) Day of 25th, 10:00 H2510	
SD3			• The minute and second when SD0 data was updated are stored as a 2-digit BCD code. b15 to b8 b7 to b0 Min. (0 to 59) | Sec. (0 to 59) (Example) 35 minutes, 48 seconds H3548	
SD4	Error information identification	Error information identification code	Identification codes used to determine what error information stored in the common information (SD5 to SD15) and individual information (SD16 to SD26) are stored. b15 to b8 b7 to b0 Individual information category code | Common information category code • In the common information category codes, the following codes are stored. 0: None 1: Module No. (Slot No./CPU No./Base No.) 3: Time (setting value) 7: Base No./power supply No. • In the individual information category codes, the following codes are stored. 0: None 3: Time (actual measurement value) 5: Parameter No.	S (Error)

(To the next page)

Table APPX.17 Special registers (continued)

<table>
<tr><th>Number</th><th>Name</th><th>Description</th><th>Detailed description</th><th>Setting side
(setting time)</th></tr>
<tr><td>SD5
to
SD15</td><td>Common error information</td><td>Common error information</td><td>

- Common information corresponding to the error code (SD0) is stored.
- The following three types of information are stored.
- Common error information type can be determined by "Common information category code" in SD4.
  (The value of the Common information category code stored in SD4 corresponds to 1), 3), and 7) below.)

1) Module No.

<table>
<tr><th>No.</th><th>Item</th></tr>
<tr><td>SD5</td><td>Slot No./CPU No./BaseNo. *1, *2</td></tr>
<tr><td>SD6</td><td>I/O No. *3</td></tr>
<tr><td>SD7</td><td>(Empty)</td></tr>
<tr><td>SD8</td><td></td></tr>
<tr><td>SD9</td><td></td></tr>
<tr><td>SD10</td><td></td></tr>
<tr><td>SD11</td><td></td></tr>
<tr><td>SD12</td><td></td></tr>
<tr><td>SD13</td><td></td></tr>
<tr><td>SD14</td><td></td></tr>
<tr><td>SD15</td><td></td></tr>
</table>

*1 For multiple CPU systems, the slot No. or CPU No. is stored, depending on the error.
(Regarding which number is stored, refer to each error code.)
No. 1 CPU: 1, No. 2 CPU: 2, No. 3 CPU: 3, No. 4 CPU: 4

*2 Definition of the base No. and slot No.
<Base No.>
The value to identify the base unit mounted on the C Controller module.
The definition of the base No. is shown in the following table.

<table>
<tr><th>BaseNo.</th><th>Definition</th></tr>
<tr><td>0</td><td>Indicates the main base unit which a CPU module is mounted.</td></tr>
<tr><td>1 to 7</td><td>Indicates an extension base unit.
The stage number setting by a base number setting connector of extension base unit turn to be Base No.
When a stage number setting is the first stage of extension module: Base No. = 1
When a stage number setting is the seventh stage of extension module: Base No. = 7</td></tr>
</table>

<Slot No.>
This value to identify the slot of each base unit and the module inserted in the slot.

- I/O slot 0 (the right side of slot for the CPU slot) of the main base unit is the slot of "Slot No. = 0".
- The slot No. order is the main base unit, the extension base unit numbers 1 to 7 in that order, and a serial number is assigned to each slot of each base unit.
- If I/O assignment of C Controller setting utility is used to set the number of slots in the base unit, slot numbers are assigned for the set number of slots.

*3 When FFFF$_H$ is stored in SD6 (I/O No.), the error location is identified by SD5 because I/O numbers such as duplicated I/O numbers cannot be identified when using I/O assignment of C Controller setting utility.

3) Time (setting value)

<table>
<tr><th>No.</th><th>Item</th></tr>
<tr><td>SD5</td><td>(Empty)</td></tr>
<tr><td>SD6</td><td>Hr.: in units of 1ms (0 to 65535ms)</td></tr>
<tr><td>SD7</td><td>(Empty)</td></tr>
<tr><td>SD8</td><td></td></tr>
<tr><td>SD9</td><td></td></tr>
<tr><td>SD10</td><td></td></tr>
<tr><td>SD11</td><td></td></tr>
<tr><td>SD12</td><td></td></tr>
<tr><td>SD13</td><td></td></tr>
<tr><td>SD14</td><td></td></tr>
<tr><td>SD15</td><td></td></tr>
</table>

</td><td>S
(Error)</td></tr>
</table>

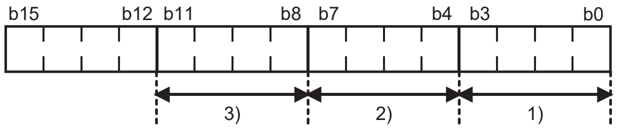
(To the next page)

Table APPX.17 Special registers (continued)

<table>
<tr><th>Number</th><th>Name</th><th>Description</th><th>Detailed description</th><th>Setting side
(setting time)</th></tr>
<tr><td>SD5
to
SD15</td><td>Common error information</td><td>Common error information</td><td>7) Base No./power supply No.
<table>
<tr><th>No.</th><th>Item</th></tr>
<tr><td>SD5</td><td>Base No.</td></tr>
<tr><td>SD6</td><td>Power supply No.</td></tr>
<tr><td>SD7</td><td>(Empty)</td></tr>
<tr><td>SD8</td><td></td></tr>
<tr><td>SD9</td><td></td></tr>
<tr><td>SD10</td><td></td></tr>
<tr><td>SD11</td><td></td></tr>
<tr><td>SD12</td><td></td></tr>
<tr><td>SD13</td><td></td></tr>
<tr><td>SD14</td><td></td></tr>
<tr><td>SD15</td><td></td></tr>
</table>
1: Error occurring in power supply 1
2: Error occurring in power supply 2
"Power supply 1" :
Redundant power supply module mounting on POWER 1 slot of the redundant base unit (Q38RB, Q68RB, and Q65WRB)
"Power supply 2" :
Redundant power supply module mounting on POWER 2 slot of the redundant base unit (Q38RB, Q68RB, and Q65WRB)</td><td>S
(Error)</td></tr>
<tr><td>SD16
to
SD26</td><td>Error individual information</td><td>Error individual information</td><td>• Individual information corresponding to the error code (SD0) is stored.
• Two types of information are stored.
• Error individual information type can be determined by "individual information category code" in SD4.
(The value of the individual information category code stored in SD4 corresponds to 3) and 5) below.)
3) Time (actual measurement value)
<table>
<tr><th>No.</th><th>Item</th></tr>
<tr><td>SD16</td><td>(Empty)</td></tr>
<tr><td>SD17</td><td>Hr.: in units of 1ms (0 to 65535ms)</td></tr>
<tr><td>SD18</td><td>(Empty)</td></tr>
<tr><td>SD19</td><td></td></tr>
<tr><td>SD20</td><td></td></tr>
<tr><td>SD21</td><td></td></tr>
<tr><td>SD22</td><td></td></tr>
<tr><td>SD23</td><td></td></tr>
<tr><td>SD24</td><td></td></tr>
<tr><td>SD25</td><td></td></tr>
<tr><td>SD26</td><td></td></tr>
</table>
5) Parameter No.
<table>
<tr><th>No.</th><th>Item</th></tr>
<tr><td>SD16</td><td>Parameter No.</td></tr>
<tr><td>SD17</td><td>(Empty)</td></tr>
<tr><td>SD18</td><td></td></tr>
<tr><td>SD19</td><td></td></tr>
<tr><td>SD20</td><td></td></tr>
<tr><td>SD21</td><td></td></tr>
<tr><td>SD22</td><td></td></tr>
<tr><td>SD23</td><td></td></tr>
<tr><td>SD24</td><td></td></tr>
<tr><td>SD25</td><td></td></tr>
<tr><td>SD26</td><td></td></tr>
</table></td><td>S
(Error)</td></tr>
<tr><td>SD60</td><td>Fuse blown module No.</td><td>Fuse blown module No.</td><td>• The latest I/O number of the module that generated the blown fuse is stored.</td><td>S
(Error)</td></tr>
<tr><td>SD61</td><td>I/O module verification error No.</td><td>I/O module verification error No.</td><td>• The latest I/O number of the module that generated the I/O module verification error is stored.</td><td>S
(Error)</td></tr>
</table>

(2) System information

Table APPX.18 Special registers

Number	Name	Description	Detailed description	Setting side (setting time)
SD200	Switch status	C Controller module switch status	• The switch status of the C Controller module is stored. 0: RUN, 1: STOP	S (System)
SD203	C Controller module operation status	C Controller module operation status	• The operation status of the C Controller module is stored as shown in the following diagram.  b15 b12 b11 b8 b7 b4 b3 b0 3) 2) 1) 1): C Controller module operation status 0: RUN 1: Reserved 2: STOP 3: PAUSE 2): STOP/PAUSE factor (in chronological order) 0: RUN/STOP/MODE Switch 1: Reserved 2: Remote operation from C Controller setting utility 3: QBF_Control function executed from a user program 4: Error 5: Remote operation from another CPU 3): Mode 0: Basic mode 1: Extended mode	S (System)
SD286	Device assignment	M assignment points (for extension)	• The M assignment points are stored in 32-bit. • Even when M assignment points are 32K or less, the assignment points are stored.	S (Initial)
SD287				
SD308	Device assignment 2 (including extension device settings)	D assignment points (internal + extension)	• The M assignment points are stored in 32-bit. (the total points of the D device and extension D device in the internal device memory area)	S (Initial)
SD309				

(3) Drive information

Table APPX.19 Special registers

Number	Name	Description	Detailed description	Setting side (setting time)
SD670	Parameter-valid drive information	Parameter-valid drive No.	• Information for the valid parameter storage drive is stored. 2: Drive 2 (CompactFlash card) 3: Drive 3 (standard RAM)	S (Initial)

(4) Fuse blown module

Table APPX.20 Special registers

Number	Name	Description	Detailed description	Setting side (setting time)
SD1300 to SD1331	Fuse blown module	The bit pattern of the fuse blown module in units of 16 points 0: No fuse blown 1: Fuse blown	• The output module numbers (units of 16 points), which are in the fuse blown status, are entered in the bit pattern. (If parameters are used, the set numbers are entered.) b15 b14 b13 b12 b11 b10 b9 b8 b7 b6 b5 b4 b3 b2 b1 b0 SD1300: 0 0 0 1 (YC0) 0 0 0 1 (Y80) 0 0 0 0 0 0 0 0 SD1301: 1 (Y1F0) 0 0 0 0 1 (Y1A0) 0 0 0 0 0 0 0 0 0 0 SD1331: 0 0 0 0 1 (Y1FB0) 0 0 0 0 0 0 0 1 (Y1F30) 0 0 0 Indicates the fuse blown status. For a module whose number of output points exceeds 16 points, all bits corresponding to output module numbers within the number of output points occupied by the module (in increments of 16 points) turn on. Ex. When a 64-point module is mounted on the slot 0, b0 to b3 turn on when the fuse has blown. • This is not cleared even when normal status is restored. Perform error clear to clear them.	S (Error)

(5) I/O module verification

Table APPX.21 Special registers

Number	Name	Description	Detailed description	Setting side (setting time)
SD1400 to SD1431	I/O module verification error	Bit pattern of verification error module in units of 16 points 0: No I/O verification error 1: I/O verification error	• When an I/O module having different I/O module information registered upon power on is detected, its I/O module number is entered in the bit pattern. (If parameters are used, the set I/O module numbers are entered.) b15 b14 b13 b12 b11 b10 b9 b8 b7 b6 b5 b4 b3 b2 b1 b0 SD1400: 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 (XY0) SD1401: 0 0 0 0 0 0 1 (XY190) 0 0 0 0 0 0 0 0 0 SD1431: 0 1 (XY1FE0) 0 0 0 0 0 0 0 0 0 0 0 0 0 0 Indicates I/O module verify err. For a module whose number of I/O points exceeds 16 points, all bits corresponding to I/O module numbers within the number of I/O points occupied by the module (in increments of 16 points) turn on. Ex. When a 64-point module is mounted on the slot 0, b0 to b3 turn on when an error is detected. • This is not cleared even when normal status is restored. Perform error clear to clear them.	S (Error)

Appendix 8 Parameter Number List

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	○

A parameter number is displayed on the Detailed event information screen of C Controller setting utility when an error occurs in parameter setting. (☞ Page 16-34, Section 16.3) This section provides the tables indicating parameter numbers and where each parameter is to be set.

(1) Symbols, mn, **, M, and N used in the "Parameter No." column

mn, **, M, and N in the "Parameter No." column denote the following:

mn:	Start I/O No. divided by 16
**:	Any given value
N:	The module number
M:	Network type

Table APPX.22 In CC-Link IE controller network setting

M	Network type
1H	Control station
2H	Normal station

Table APPX.23 In MELSECNET/H setting

M	Network type
1H	MELSECNET/10 mode (Control station), MELSECNET/H mode (Control station), MELSECNET/H Extended mode (Control station)
2H	MELSECNET/10 mode (Normal station), MELSECNET/H mode (Normal station), MELSECNET/H Extended mode (Normal station)
5H	MELSECNET/H (Remote master station)
AH	MELSECNET/H (Standby station)
BH	MELSECNET/H mode multiplexed remote I/O network master station
DH	MELSECNET/H mode multiplexed remote I/O network sub-master station (When there is no parameter setting)
EH	MELSECNET/H mode multiplexed remote I/O network sub-master station (When there is parameter setting)

Table APPX.24 In CC-Link setting

M	Network type
0H	Master station
1H	Local station
2H	Standby master station

(2) Parameter No. list

The following table shows the parameter numbers and where each of them is to be set.

Table APPX.25 Parameter No. list

Parameter No.	Item		Utility
0400H	I/O assignment	Type	C Controller setting utility
		Model name	
		Points	
		Start X/Y (Start I/O No.)	
0401H	Base setting	Base model	C Controller setting utility
		Power model	
		Extension cable	
		Slots	
0403H	Detailed setting	Error time output mode	C Controller setting utility
0405H		I/O response time	C Controller setting utility
0406H		Control CPU	C Controller setting utility
0407H	Switch settings		C Controller setting utility
0409H	Switch settings		C Controller setting utility
05mnH	Group No.		MELSECNET/H utility
0AmnH	Group No.		CC IE Control utility
0CmnH	Operation setting	Type	CC-Link utility
		Mode setting	
		Extended cyclic settings	
		Number of occupied stations	
		Input data at error time	
		Setting during CPU STOP	
0E00H	Number of CPUs		C Controller setting utility
0E01H	Operation mode		C Controller setting utility
0E04H	I/O sharing when using Multiple CPUs	All CPUs can read all inputs	C Controller setting utility
		All CPUs can read all outputs	
1002H	Remote reset		C Controller setting utility
1003H	Output mode at STOP to RUN		C Controller setting utility
1007H	Points occupied by empty slot		C Controller setting utility
100AH	Intelligent function module settings (Interrupt event settings)		C Controller setting utility
100CH	Module synchronization		C Controller setting utility
1016H	Built-in Ethernet port open setting		C Controller setting utility
2000H	Device setting	Number of device points	C Controller setting utility

(To the next page)

Table APPX.25 Parameter No. list (continued)

Parameter No.	Item		Utility
3000H	WDT(Watchdog timer) setting		C Controller setting utility
3001H	Error check	Battery check	C Controller setting utility
		Fuse blown check	
		Module verification	
3002H	Operation mode at the time of error	Fuse blown	C Controller setting utility
		Module verify error	
4004H	Detailed settings	H/W error time CPU operation mode	C Controller setting utility
5000H	Number of modules		MELSECNET/H utility
5003H	Routing parameter		MELSECNET/H utility or CC IE Control utility
5NM0H	Start I/O No.		MELSECNET/H utility
	Network No.		
	Total stations		
	Mode		
5NM1H	Refresh parameters		MELSECNET/H utility
5NM2H	Network range assignment		MELSECNET/H utility
A000H	Number of modules		CC IE Control utility
ANM0H	Start I/O No.		CC IE Control utility
	Network No.		
	Network range assignment	Total number of stations	
	Station number		
	Mode		
ANM1H	Refresh parameters		CC IE Control utility
ANM2H	Network range assignment		CC IE Control utility
ANM3H	Station inherent parameters		CC IE Control utility
C000H	Number of modules		CC-Link utility
CNM2H	Start I/O No.		CC-Link utility
	All connect count		
	Other settings	Retry count	
		Auto return station count	
		Standby master station No.	
		CPU down: data link status	
		Delay information setting	
		Block data assurance per station	
	Station information settings		
E003H	Normal area setting (refresh setting)	CPU specific send range	C Controller setting utility
E006H	Online module change		C Controller setting utility
E008H	Multiple CPU high speed transmission area setting	CPU specific send range	C Controller setting utility
E00BH	Multiple CPU synchronous startup setting		C Controller setting utility
E00CH	Host CPU number		C Controller setting utility
FD00H	Event history registration settings		C Controller setting utility
Start I/O number divided by 10	Refresh parameters of intelligent function module		Software package in which refresh parameters are set

Appendix 9 Connection with GX Works2/GX Developer

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	×	×

This section explains the setting method of when connecting GX Works2/GX Developer to a C Controller module.

This section does not cover precautions and tips on the use of GX Works2/GX Developer. For precautions and tips on the use of GX Works2/GX Developer, refer to the following.

☞ GX Works2/GX Developer Operating Manual

Appendix 9.1 When accessing another CPU of a multiple CPU system

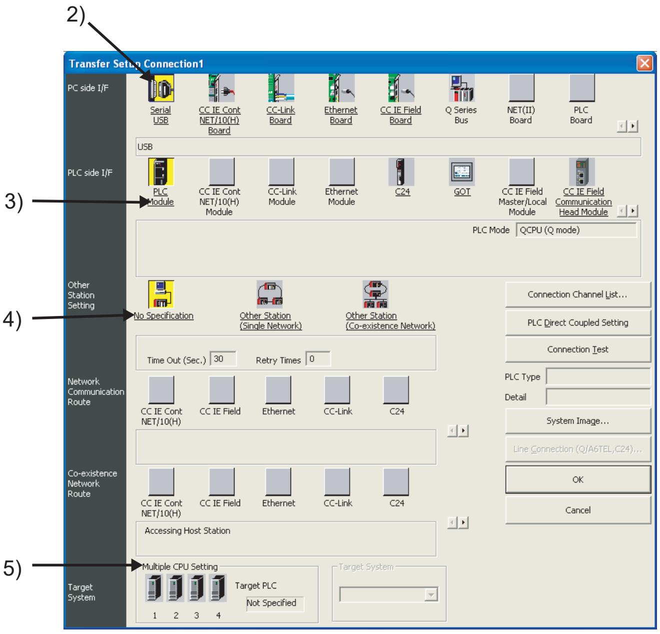
Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	×	×

(1) GX Works2

(a) USB connection

To access another CPU module of a multiple CPU system from the C Controller module via USB, follow the procedure below.

1) Navigation window → Select [Connection Destination].



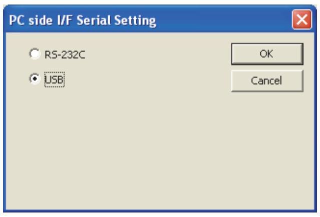
USB

PLC Mode QCPU (Q mode)

Time Out (Sec.) 30 Retry Times 0

Target PLC Not Specified

2) Set "PC side I/F" to "Serial USB".
   Set the detailed setting dialog box as follows.

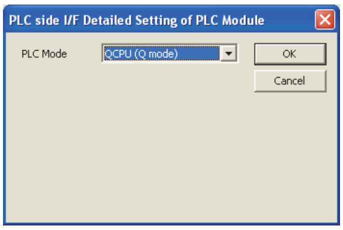


PC side I/F Serial Setting

○ RS-232C
● USB

- Select "USB". RS-232C cannot be used.

3) Set "PLC side I/F" to "PLC module".
   Set the detailed setting dialog box as follows.

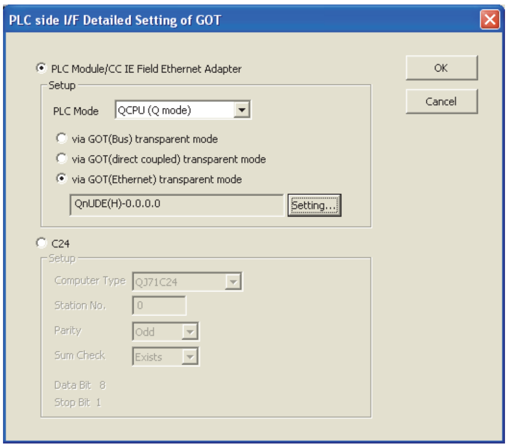


PLC side I/F Detailed Setting of PLC Module

PLC Mode: QCPU (Q mode)

- For "PLC mode", select "QCPU (Q mode)".

To use the GOT transparent function, set "PLC side I/F" to "GOT".
Set the detailed setting dialog box as follows.



PLC side I/F Detailed Setting of GOT

● PLC Module/CC IE Field Ethernet Adapter

Setup

PLC Mode: QCPU (Q mode)

○ via GOT(Bus) transparent mode
○ via GOT(direct coupled) transparent mode
● via GOT(Ethernet) transparent mode

QnUDE(H)-0.0.0.0

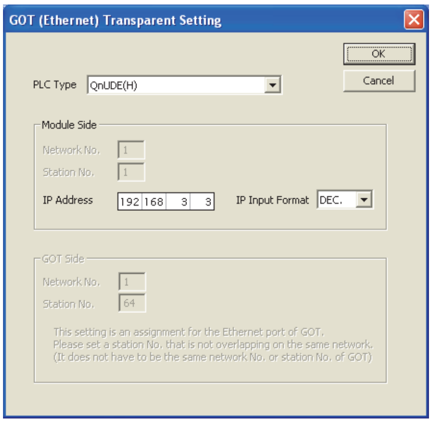
○ C24

Setup

Computer Type: QJ71C24
Station No.: 0
Parity: Odd
Sum Check: Exists
Data Bit 8
Stop Bit 1

- The GOT (direct) transparent function cannot be used.
  Do not check the box.

- To use the GOT (Ethernet) transparent function, set the following GOT (Ethernet) transparent setting dialog box as follows.



GOT (Ethernet) Transparent Setting

PLC Type: QnUDE(H)

Module Side
Network No.: 1
Station No.: 1
IP Address: 192.168.3.3
IP Input Format: DEC.

GOT Side
Network No.: 1
Station No.: 64

This setting is an assignment for the Ethernet port of GOT.
Please set a station No. that is not overlapping on the same network.
(It does not have to be the same network No. or station No. of GOT)

  - For "PLC type", select "QnUDE(H)".
  - For "IP address", set the IP address of the connecting C Controller module.

For details on the GOT transparent, refer to the following.

☞ GOT1000 Series Connection Manual (Mitsubishi Products)

4) Set "Other station" to "No other station".

5) For "Multiple CPU setting", set the CPU to be accessed.

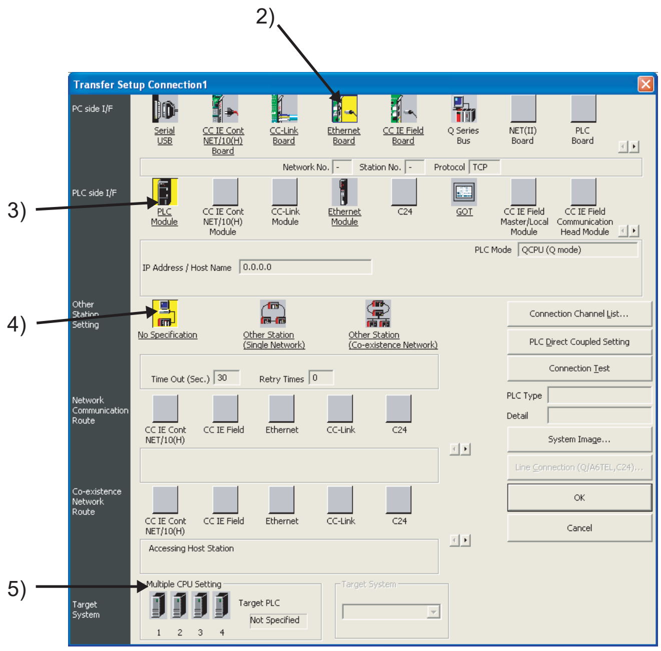
(b) Ethernet connection

To access another CPU module of a multiple CPU system from the C Controller module via Ethernet, follow the procedure below.

POINT

- To connect GX Works2 to the C Controller module via Ethernet, built-in Ethernet port open settings need to be configured on the <<System settings>> tab of the C Controller setting utility.
  For details, refer to the following.
  ☞ C Controller Module User's Manual (Utility Operation, Programming)

1) Navigation window → Select [Connection Destination].



Network No. - Station No. - Protocol TCP

PLC Mode QCPU (Q mode)

IP Address / Host Name 0.0.0.0

Time Out (Sec.) 30 Retry Times 0

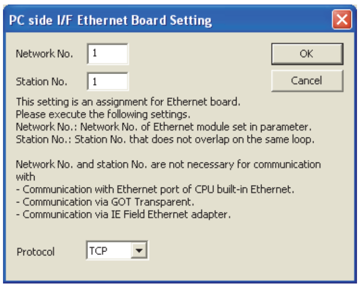
Accessing Host Station

Target PLC Not Specified

PLC Type

Detail

2) Set "PC side I/F" to "Ethernet board".
Set the detailed setting dialog box as follows.



PC side I/F Ethernet Board Setting

Network No. 1

Station No. 1

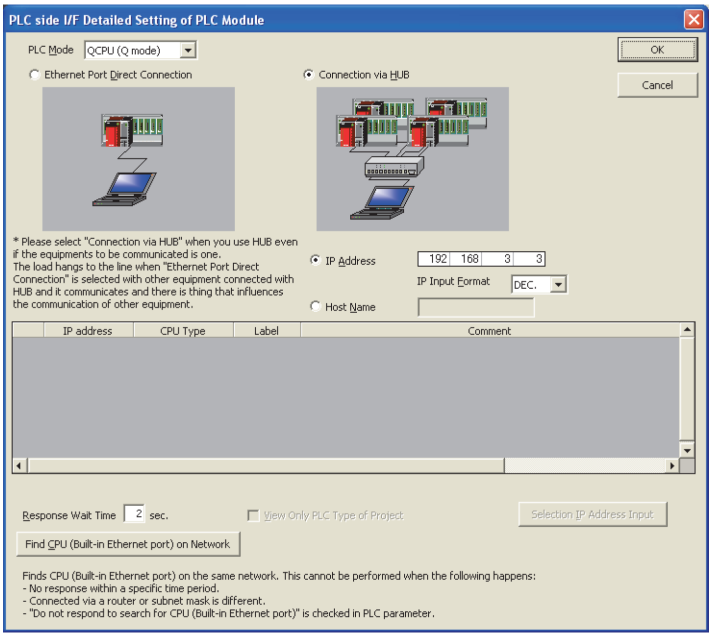
This setting is an assignment for Ethernet board.
Please execute the following settings.
Network No.: Network No. of Ethernet module set in parameter.
Station No.: Station No. that does not overlap on the same loop.

Network No. and station No. are not necessary for communication with
- Communication with Ethernet port of CPU built-in Ethernet.
- Communication via GOT Transparent.
- Communication via IE Field Ethernet adapter.

Protocol TCP

- "Network No." and "Station No."do not need to be changed because they are not used.
- For "Protocol", select either in "TCP" or "UDP".

3) Set "PLC side I/F" to "PLC module".
Set the detailed setting dialog box as follows.



PLC side I/F Detailed Setting of PLC Module

PLC Mode QCPU (Q mode)

○ Ethernet Port Direct Connection

● Connection via HUB

* Please select "Connection via HUB" when you use HUB even if the equipments to be communicated is one.
The load hangs to the line when "Ethernet Port Direct Connection" is selected with other equipment connected with HUB and it communicates and there is thing that influences the communication of other equipment.

● IP Address 192 168 3 3

IP Input Format DEC.

○ Host Name

	IP address	CPU Type	Label	Comment

Response Wait Time 2 sec.

□ View Only PLC Type of Project

Finds CPU (Built-in Ethernet port) on the same network. This cannot be performed when the following happens:
- No response within a specific time period.
- Connected via a router or subnet mask is different.
- "Do not respond to search for CPU (Built-in Ethernet port)" is checked in PLC parameter.

OK
Cancel
Selection IP Address Input
Find CPU (Built-in Ethernet port) on Network

- For "PLC mode", select "QCPU (Q mode)".
- The Ethernet port cannot be connected directly. Do not check the box.
- For "IP address", set the IP address of the connecting C Controller module.
  When selecting "Host Name", input the name of the hosts file in no more than 64 characters.
- C Controller modules cannot be searched by the Find CPU (Built-in Ethernet port) on network button.

4) Set "Other station" to "No other station".

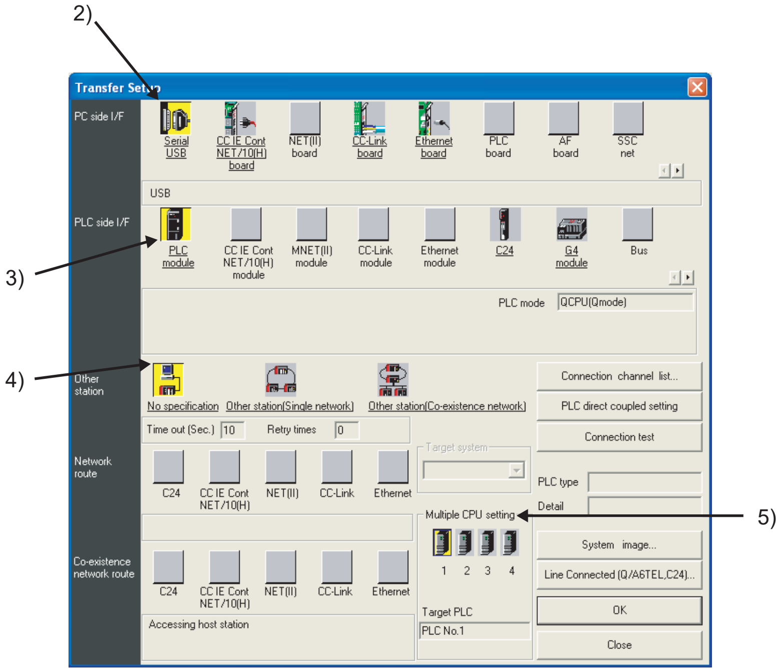
5) For "Multiple CPU setting", set the CPU to be accessed.

(2) GX Developer

(a) USB connection

To access another CPU module of a multiple CPU system from the C Controller module via USB, follow the procedure below.

1) [Online] → Select [Transfer setup].



USB

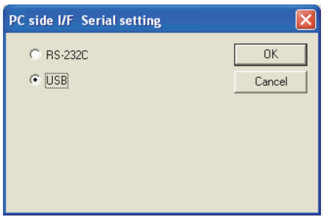
PLC mode QCPU(Qmode)

Time out (Sec.) 10 Retry times 0

Accessing host station

Target PLC PLC No.1

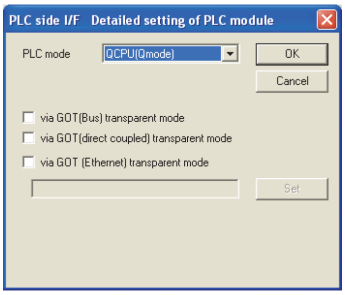
2) Set "PC side I/F" to "Serial USB".
Set the detailed setting dialog box as follows.



○ RS-232C
● USB

- Select "USB". RS-232C cannot be used.

3) Set "PLC side I/F" to "PLC module".
Set the detailed setting dialog box as follows.



PLC side I/F Detailed setting of PLC module

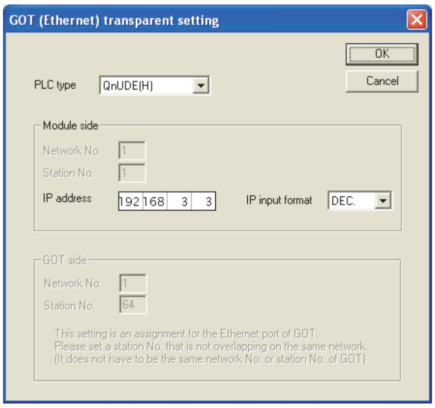
PLC mode: QCPU(Qmode)

□ via GOT(Bus) transparent mode

□ via GOT(direct coupled) transparent mode

□ via GOT (Ethernet) transparent mode

- For "PLC mode", select "QCPU (Q mode)".
- The GOT (direct) transparent function cannot be used.
  Do not check the box.
- To use the GOT (Ethernet) transparent function, set the GOT (Ethernet) transparent setting dialog box as follows.



GOT (Ethernet) transparent setting

PLC type: QnUDE(H)

Module side
Network No.: 1
Station No.: 1
IP address: 192.168.3.3
IP input format: DEC.

GOT side
Network No.: 1
Station No.: 64

This setting is an assignment for the Ethernet port of GOT.
Please set a station No. that is not overlapping on the same network.
(It does not have to be the same network No. or station No. of GOT)

- For "PLC type", select "QnUDE(H)".
- For "IP address", set the IP address of the connecting C Controller module.

For details on the GOT transparent, refer to the following.

☞ GOT1000 Series Connection Manual (Mitsubishi Products)

4) Set "Other station" to "No other station".

5) For "Multiple CPU setting", set the CPU to be accessed.

(b) Ethernet connection

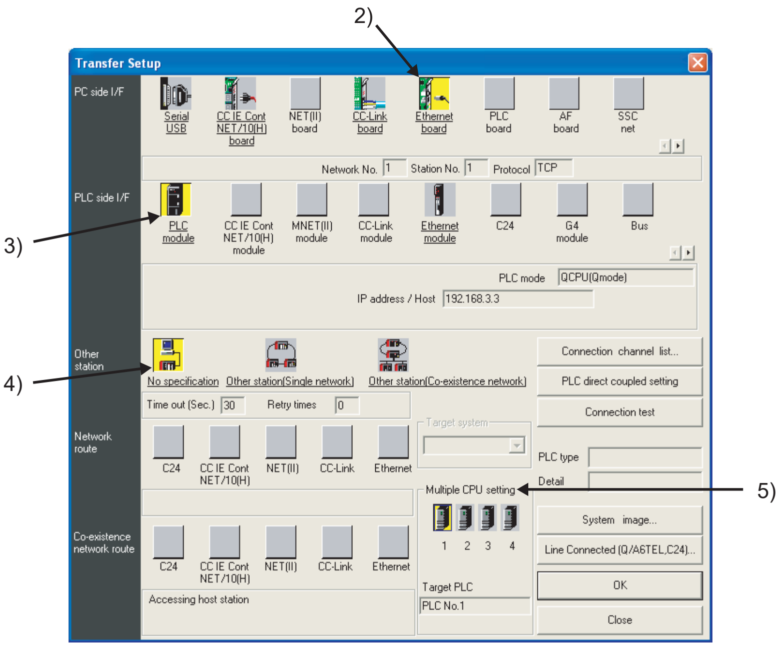
To access another CPU module of a multiple CPU system from the C Controller module via Ethernet, follow the procedure below.

☒POINT

- To connect GX Developer to the C Controller module via Ethernet, built-in Ethernet port open settings need to be configured on the <<System settings>> tab of the C Controller setting utility.
  For details, refer to the following.
  ☞ C Controller Module User's Manual (Utility Operation, Programming)
- When connected to GX Developer through Ethernet, the C Controller module cannot access the following CPU modules.
  - Q00CPU, Q01CPU
  - Q00UCPU, Q01UCPU, Q02UCPU

  Connect them with a USB cable. (☞ Page App-30, Appendix 9.1 (2) (a))

1) [Online] → Select [Transfer setup].



Network No. 1 Station No. 1 Protocol TCP

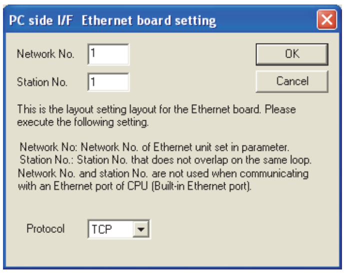
PLC mode QCPU(Qmode)

IP address / Host 192.168.3.3

Time out (Sec.) 30 Retry times 0

Target PLC PLC No.1

2) Set "PC side I/F" to "Ethernet board".
   Set the detailed setting dialog box as follows.



PC side I/F Ethernet board setting

Network No. 1

Station No. 1

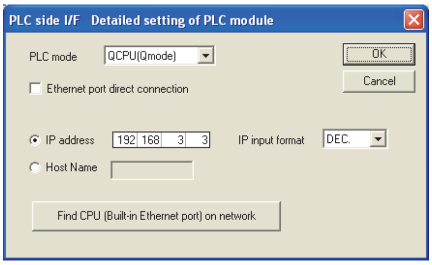
This is the layout setting layout for the Ethernet board. Please execute the following setting.

Network No: Network No. of Ethernet unit set in parameter.
Station No.: Station No. that does not overlap on the same loop.
Network No. and station No. are not used when communicating with an Ethernet port of CPU (Built-in Ethernet port).

Protocol TCP

- "Network No." and "Station No."do not need to be changed because they are not used.
- For "Protocol", select either in "TCP" or "UDP".

3) Set "PLC side I/F" to "PLC module".
   Set the detailed setting dialog box as follows.



PLC side I/F Detailed setting of PLC module

PLC mode QCPU(Qmode)

□ Ethernet port direct connection

● IP address 192 168 3 3 IP input format DEC.

○ Host Name

- For "PLC mode", select "QCPU (Q mode)".
- The Ethernet port cannot be connected directly. Do not check the box.
- For "IP address", set the IP address of the connecting C Controller module.
- C Controller modules cannot be searched by the

  Find CPU (Built-in Ethernet port) on network button.

4) Set "Other station" to "No other station".

5) For "Multiple CPU setting", set the CPU to be accessed.

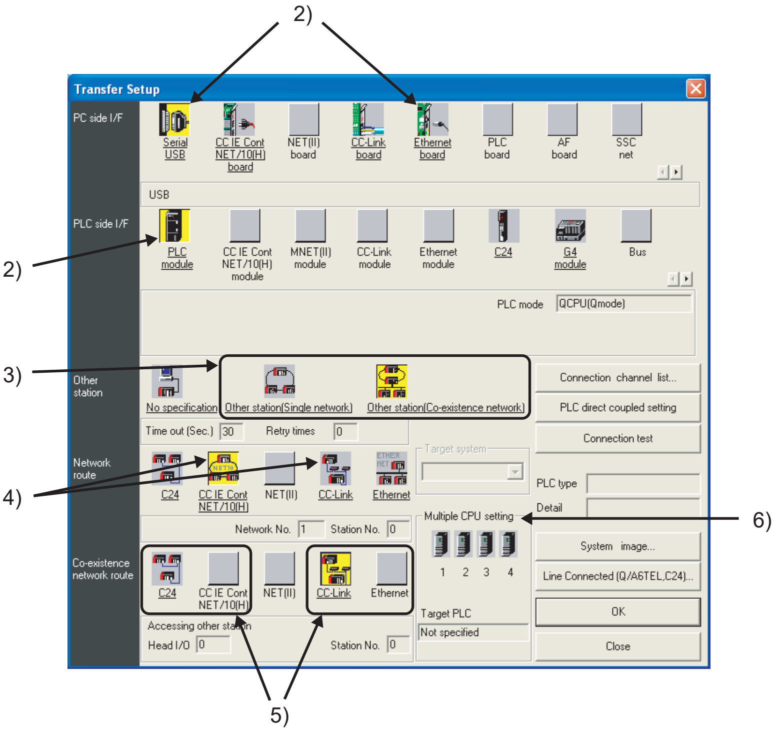
Appendix 9.2 Accessing other stations

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	×	×

To access other stations from the C Controller module via USB or Ethernet, follow the procedure below.

(1) GX Works2

1) lNavigation window→Select [Connection Destination].



PLC mode QCPU(Qmode)

Time out (Sec.) 30 Retry times 0

Network No. 1 Station No. 0

Head I/O 0 Station No. 0

Target PLC Not specified

2) Set "PC side I/F" and "PLC side I/F" in the same way as for accessing another CPU in a multiple CPU system. (☞ Page App-25, Appendix 9.1)

3) If connection via a system consisting of single type of network and multi-layer systems is used, set "Other station setting" to "Other station (Single network)". If connection via two different types of networks is used, set "Other station" to "Other station (Co-existence network)".

POINT

Ethernet is assumed to be the same type of network as CC-Link IE controller network, MELSECNET/H, or MELSECNET/10.
For this reason, when a combined system of Ethernet, CC-Link IE controller network, MELSECNET/H, or MELSECNET/10 is used, specify "Other station (single network)".

4) When "Other station (Single network)" is specified for "Other station", specify accessed network type*1 for "Network route".
When "Other station (Co-existence network)" is specified for "Other station", for "Network route" specify the accessed network type*1 to which the C Controller module is connected.

* 1 "C24" and "Ethernet" cannot be specified for the network type.

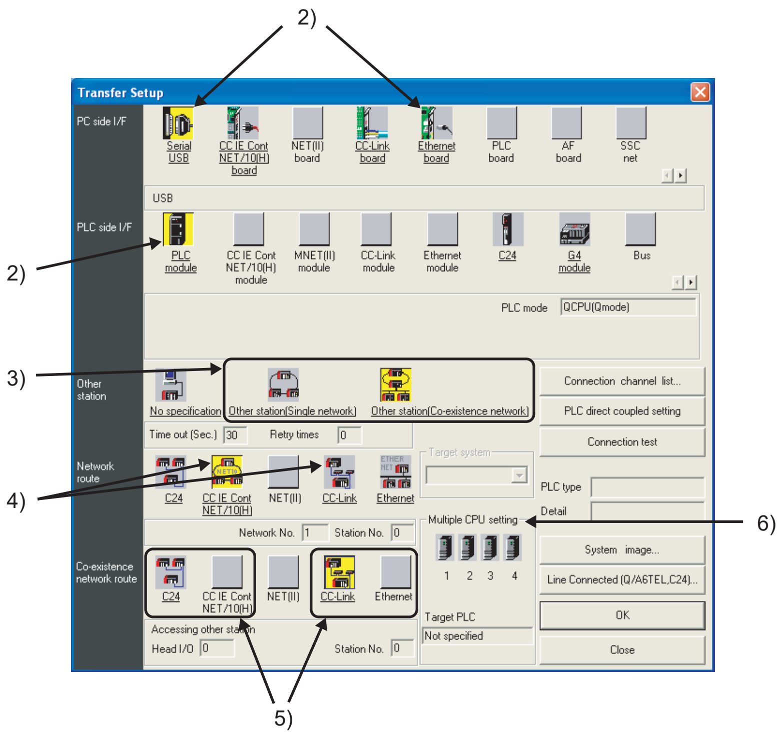
After specifying the network type, set "Network No.", "Station No." and "Head I/O No." Setting items vary depending on the network type.

5) Only when "Other station (coexistence network)" is specified for "Other station", specify the accessed network type for "Co-existence network route".
After specifying the network type, set "Network No.", "Station No." and "Head I/O No." Setting items vary depending on the network type.

6) If the CPU to be accessed is a multiple CPU system, set the accessed CPU for "Multiple CPU setting".
For more details, refer to the following.
☞ GX Works2 Version1 Operating Manual (Common)

(2) GX Developer

1) [Online] → Select [Transfer setup].



USB

PLC mode: QCPU(Qmode)

Time out (Sec.): 30   Retry times: 0

Network No.: 1   Station No.: 0

Accessing other station
Head I/O: 0   Station No.: 0

Target PLC: Not specified

2) Set "PC side I/F" and "PLC side I/F" in the same way as for accessing another CPU in a multiple CPU system. (☞ Page App-25, Appendix 9.1)

3) If connection via a system consisting of single type of network and multi-layer systems is used, set "Other station setting" to "Other station (Single network)".
If connection via two different types of networks is used, set "Other station" to "Other station (Co-existence network)".

☒POINT

Ethernet is assumed to be the same type of network as CC-Link IE controller network, MELSECNET/H, or MELSECNET/10.
For this reason, when a combined system of Ethernet, CC-Link IE controller network, MELSECNET/H, or MELSECNET/10 is used, specify "Other station (single network)".

4) When "Other station (Single network)" is specified for "Other station", specify accessed network type[*1] for "Network route".
When "Other station (Co-existence network)" is specified for "Other station", for "Network route" specify the accessed network type[*1] to which the C Controller module is connected.

[*1] "C24" and "Ethernet" cannot be specified for the network type.

After specifying the network type, set "Network No.", "Station No." and "Head I/O No." Setting items vary depending on the network type.

5) Only when "Other station (coexistence network)" is specified for "Other station", specify the accessed network type for "Co-existence network route".
After specifying the network type, set "Network No.", "Station No." and "Head I/O No." Setting items vary depending on the network type.

6) If the CPU to be accessed is a multiple CPU system, set the accessed CPU for "Multiple CPU setting".
For more details, refer to the following.
☞ GX Developer Version 8 Operating Manual

Appendix 10 Connection with MX Component

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	×	×

This section explains the setting method of when connecting MX Component to a C Controller module.

This section does not cover precautions and tips on the use of MX Component.

For precautions and tips on the use of MX Component, refer to the following.

☞ MX Component Version 3 Operating Manual

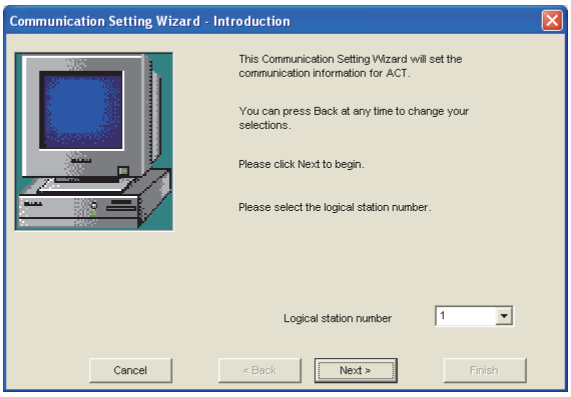
Appendix 10.1 Accessing the host station

(1) USB Connection

To access the host station from the C Controller module via USB, follow the procedure below.

1) Start the communication setting wizard from the communication setting utility.

2) Select a value for logical station number, and click the Next > button.



Communication Setting Wizard - Introduction

This Communication Setting Wizard will set the communication information for ACT.

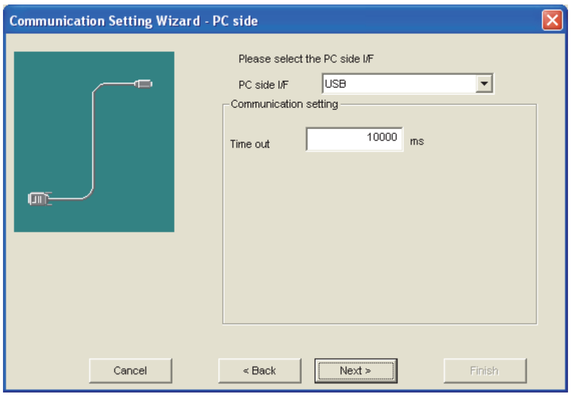
You can press Back at any time to change your selections.

Please click Next to begin.

Please select the logical station number.

Logical station number: 1

3) Set the personal computer side interface as follows, and click the Next > button.



Communication Setting Wizard - PC side

Please select the PC side I/F

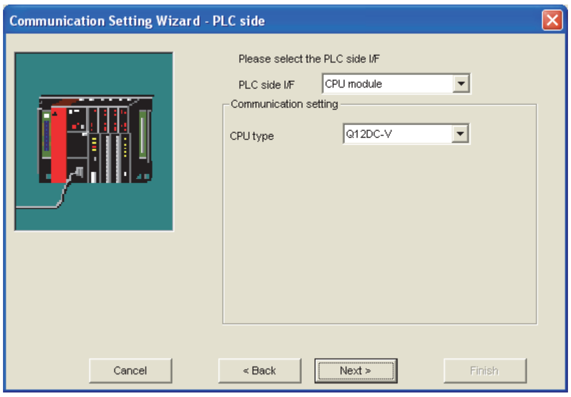
PC side I/F: USB

Communication setting

Time out: 10000 ms

- Select "USB" for "PC side I/F".

4) Set the programmable controller side interface as follows, and click the Next > button.



Communication Setting Wizard - PLC side

Please select the PLC side I/F

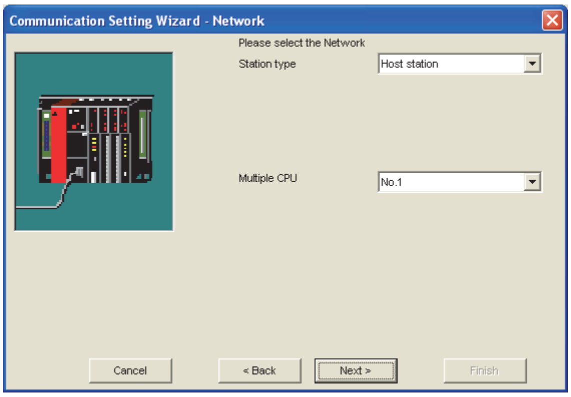
PLC side I/F: CPU module

Communication setting

CPU type: Q12DC-V

- Select "CPU module" for "PLC side I/F".
- Select "Q12DC-V" for "CPU type".

5) Set the network as follows, and click the Next > button.



Communication Setting Wizard - Network

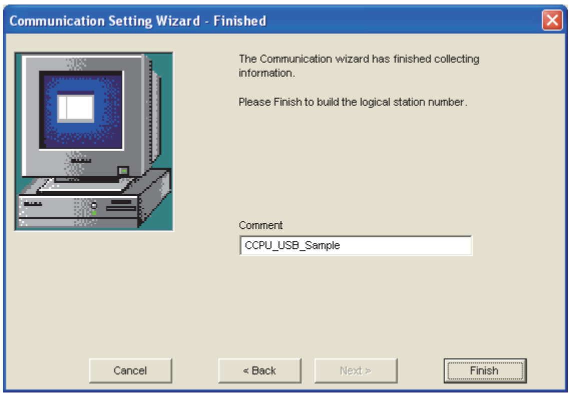
Please select the Network

Station type: Host station

Multiple CPU: No.1

- Select "Host station" for "Station type".
- Select the number of the CPU module to be accessed for "Multiple CPU".

6) Input a comment, and click the Finish button.



Communication Setting Wizard - Finished

The Communication wizard has finished collecting information.

Please Finish to build the logical station number.

Comment: CCPU_USB_Sample

☒POINT

When the access destination is the C Controller module, device function settings need to be configured on the <<Device settings>> tab of the C Controller setting utility.

For details, refer to the following.

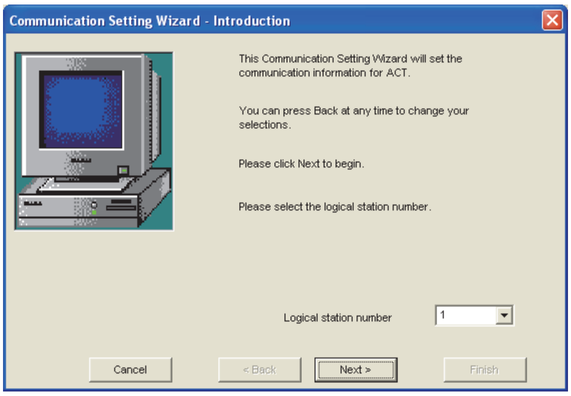
☞ C Controller Module User's Manual (Utility Operation, Programming)

(2) Ethernet connection

To access the host station from the C Controller module via Ethernet, follow the procedure below.

1) Start the communication setting wizard from the communication setting utility.

2) Select a value for logical station number, and click the Next > button.



Communication Setting Wizard - Introduction

This Communication Setting Wizard will set the communication information for ACT.

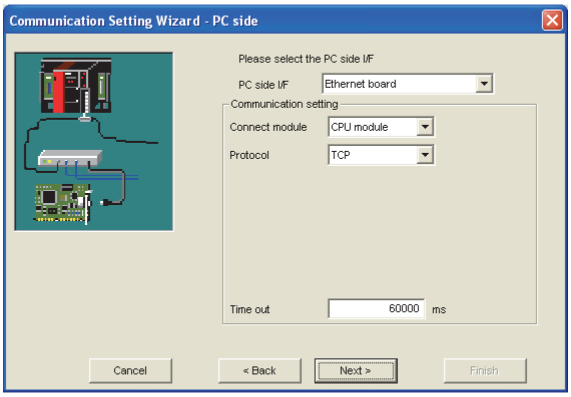
You can press Back at any time to change your selections.

Please click Next to begin.

Please select the logical station number.

Logical station number: 1

3) Set the personal computer side interface as follows, and click the Next > button.



Communication Setting Wizard - PC side

Please select the PC side I/F

PC side I/F: Ethernet board

Communication setting

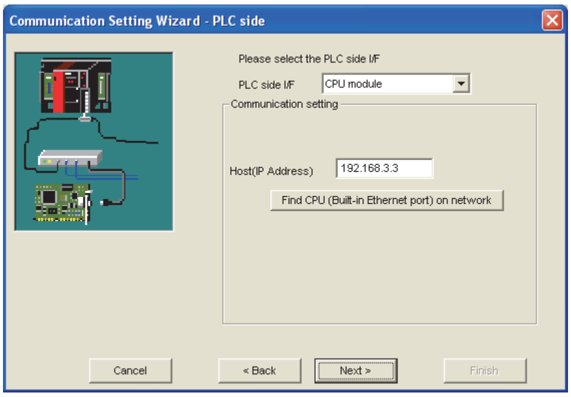
Connect module: CPU module

Protocol: TCP

Time out: 60000 ms

- Select "Ethernet board" for "PC side I/F".
- Select "CPU module" for "Connect module".
- Select "TCP" for "Protocol".

4) Set the programmable controller side interface as follows, and click the Next > button.



Communication Setting Wizard - PLC side

Please select the PLC side I/F

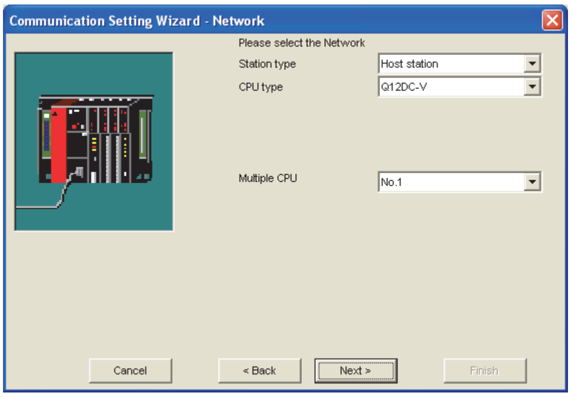
PLC side I/F: CPU module

Communication setting

Host(IP Address): 192.168.3.3

- Select "CPU module" for "PLC side I/F".
- When setting a host name for "Host (IP Address)", input the name of the hosts file in no more than 64 characters.
  When setting an IP address, input the IP address of the C Controller module to be connected.
- C Controller modules cannot be searched by the Find CPU (Built-in Ethernet port) on network button.

5) Set the network as follows, and click the Next > button.



Communication Setting Wizard - Network

Please select the Network

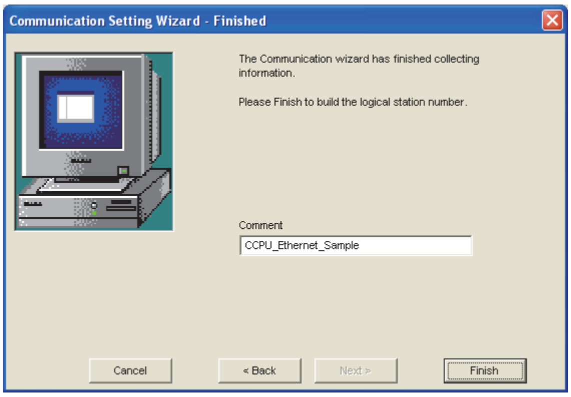
Station type: Host station

CPU type: Q12DC-V

Multiple CPU: No.1

- Select "Host station" for "Station type".
- Select the type of the CPU module to be accessed for "CPU type".
- Select the number of the CPU module to be accessed for "Multiple CPU".

6) Input a comment, and click the Finish button.



The Communication wizard has finished collecting information.

Please Finish to build the logical station number.

Comment: CCPU_Ethernet_Sample

☒POINT

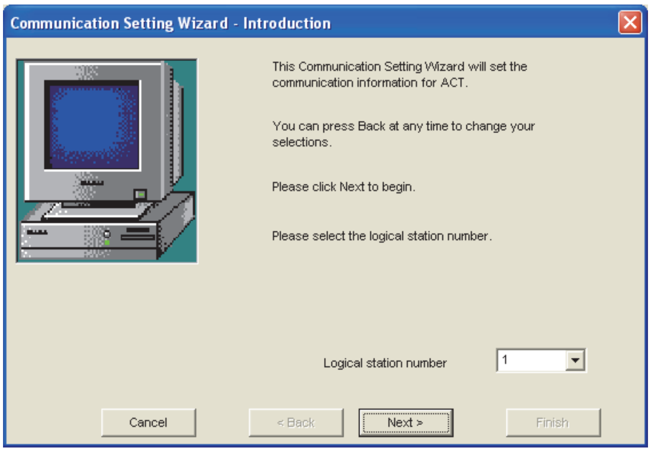
- When the access destination is the C Controller module, device function settings need to be configured on the <<Device settings>> tab of the C Controller setting utility.
  For details, refer to the following.
  ☞ C Controller Module User's Manual (Utility Operation, Programming)
- To connect MX Component to the C Controller module via Ethernet, built-in Ethernet port open settings need to be configured on the <<System settings>> tab of the C Controller setting utility.
  For details, refer to the following.
  ☞ C Controller Module User's Manual (Utility Operation, Programming)

Appendix 10.2 Accessing other stations

To access other stations from the C Controller module via USB or Ethernet, follow the procedure below.

1) Start the communication setting wizard from the communication setting utility.

2) Select a value for logical station number, and click the Next > button.



Communication Setting Wizard - Introduction

This Communication Setting Wizard will set the communication information for ACT.

You can press Back at any time to change your selections.

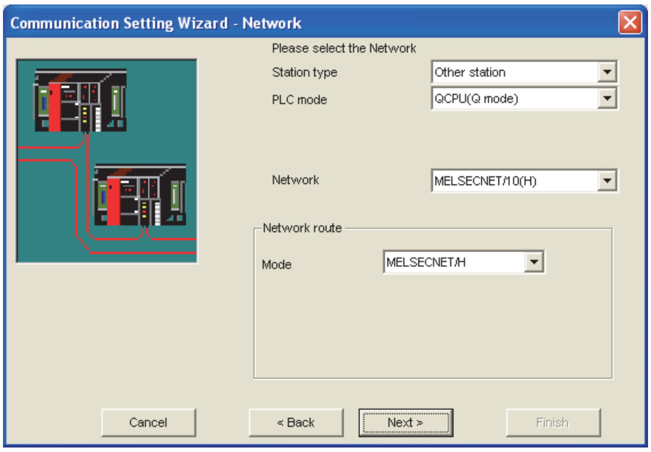
Please click Next to begin.

Please select the logical station number.

Logical station number: 1

3) Set "PC side I/F" and "PLC side I/F" in the same way as for accessing the host station.(☞ Page App-37, Appendix 10.1)

4) Set the network as follows, and click the Next > button.



Communication Setting Wizard - Network

Please select the Network

Station type: Other station

PLC mode: QCPU(Q mode)

Network: MELSECNET/10(H)

Network route

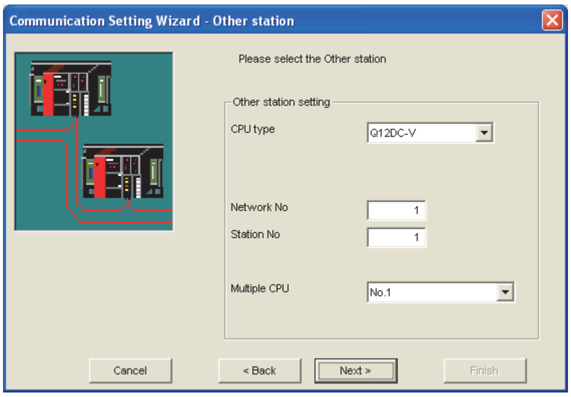
Mode: MELSECNET/H

- Select "Other station" for "Station type".
- Select "QCPU (Q mode)" for "PLC mode".
- Select the network type to be accessed for "Network".[*1]
- Select the network to be accessed for "Network route".[*2]

* 1 "C24" and "Ethernet" cannot be specified as a network type.

* 2 Setting items vary depending on a selected "Network".

5) Configure the other station setting as follows, and click the Next > button.


Communication Setting Wizard - Other station

Please select the Other station

Other station setting

CPU type: Q12DC-V

Network No: 1

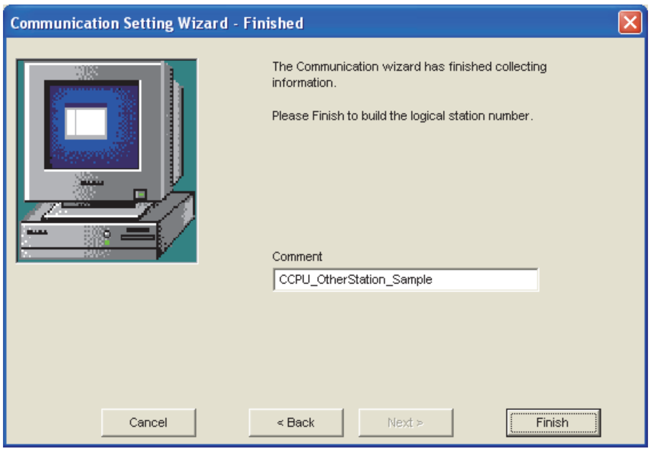
Station No: 1

Multiple CPU: No.1

- Select the CPU type of the other station for "CPU type".
- Input the network number of the other station for "Network No".
- Input the station number of the other station for "Station No".
- Select the number of the CPU module on the other station for "Multiple CPU".[*1]

[*1] Setting items vary depending on a selected "CPU type".

6) Input a comment, and click the Finish button.


Communication Setting Wizard - Finished

The Communication wizard has finished collecting information.

Please Finish to build the logical station number.

Comment: CCPU_OtherStation_Sample

Appendix 11 Functions Added by Version Upgrade

Q12DCCPU-V	Q06CCPU-V	Q06CCPU-V-B
○	○	○

The tables below shows the functions added and the serial numbers of the C Controller module and software versions of SW3PVC-CCPU that support the functions.

(1) C Controller module

Table APPX.26 Functions added to the C Controller module

1)Q12DCCPU-V 2)Q06CCPU-V 3)Q06CCPU-V-B

Function	First 5 digits of serial number			Software version of
	1)	2)	3)	SW3PVC-CCPU
Partial change on C Controller modules				
• Supporting the Q12DCCPU-V • Not supporting the Q06CCPU-V-H01	—	—	—	Version3.03D or later
Supporting more than one QI60 (interrupt module)	12042 or later	×	×	—
Addition of the function that allows access to a buffer memory in an intelligent function module through an interrupt routine (interrupt program)	12042 or later	×	×	—
Change of the battery backup RAM capacity to 512K bytes.	12042 or later	×	×	—
Addition of the following methods for connecting a GOT. • Bus connection • MELSECNET/H connection (PLC to PLC network) • MELSECNET/10 connection (PLC to PLC network) • CC-Link IE controller network connection • CC-Link connection (intelligent device station) • CC-Link connection (via G4) • Ethernet connection by the 10BASE-T/100BASE-TX interface of the C Controller module	12042 or later	×	×	— (The Ethernet connection is Version3.05F or later)
Addition of the following methods for connecting GX Developer. • Ethernet connection by the 10BASE-T/100BASE-TX interface of the C Controller module	12042 or later	×	×	Version3.05F or later
Supporting connection with MX Component	12042 or later	×	×	Version3.05F or later
Supporting connection with MELSOFT (GX Developer, GX Works2, and MT Works2) through Ethernet	12042 or later	×	×	Version3.05F or later
Addition of compact flash card as storing destination drive for parameters	12042 or later	×	×	Version3.05F or later
Addition of the following devices that can be used like programmable controller CPU • Internal relay (M), data register (D) • Special relay (SM), special register (SD)	12042 or later	×	×	Version3.05F or later
Addition of the function that allows execution of an interrupt routine (interrupt program) in the event of an interrupt from: • Intelligent function module • Interrupt module • Programmable controller CPU (another CPU) • C Controller module (another CPU)	12042 or later	×	×	Version3.05F or later
Supporting CC-Link IE controller network • Supporting the QJ71GP21-SX and QJ71GP21S-SX • CC IE Control utility has been added.	—	10012 or later	×	Version3.02C or later

— : Not related to the serial No. or the software version, ×: N/A

(To the next page)

Table APPX.26 Functions added to the C Controller module (continued)

1)Q12DCCPU-V 2)Q06CCPU-V 3)Q06CCPU-V-B

Function	First 5 digits of serial number			Software version of
	1)	2)	3)	SW3PVC-CCPU
Partial change on C Controller modules				
Addition of high speed mode to argument "sFlg" of a function (QBF_WriteSRAM function) used to write data into Battery-backed-up RAM (user area)	13032 or later	×	×	Version3.06G or later
Supporting Q12DCCPU-V (Extended mode)	15102 or later	×	×	Version3.09K or later
Supporting MODBUS® interface module	15102 or later	×	×	Version3.09K or later
Partial change on networks				
Supporting CC-Link IE controller network • Supporting the QJ71GP21-SX and QJ71GP21S-SX • CC IE Control utility has been added.	—	10012 or later	×	Version3.02C or later
Supporting CC-Link IE field network[*1] • Supporting QJ71GF11-T2.	12042 or later	12082 or later	12082 or later	×

—: Not related to the serial No. or the software version, ×: N/A

* 1 Restrictions are applied to some functions. (☞ Page 2-19, Section 2.2.2)

(2) SW3PVC-CCPU

Table APPX.27 Functions added to SW□PVC-CCPU

1)Q12DCCPU-V 2)Q06CCPU-V 3)Q06CCPU-V-B

Function		First 5 digits of serial number			Software version of
		1)	2)	3)	SW3PVC-CCPU
Partial change on the operating environment for SW3PVC-CCPU					
	• Not compatible with Microsoft® WindowsNT® Workstation Operating System Version4.0 • Compatible with Windows Vista®	—	—	—	Version3.03D or later
	• Compatible with Windows® 7	—	—	—	Version3.05F or later
	• Compatible with Windows® 8 and Windows® 8.1	—	—	—	Version3.10L or later
	• Compatible with Windows® 10	—	—	—	Version3.13P or later
	• Compatible with Windows® 11 • Not compatible with Windows XP®, Windows Vista®, Windows® 7, Windows® 8, and Windows® 8.1	—	—	—	Version3.15R or later
Partial change on SW3PVC-CCPU					
	Restoration of the battery-backed-up RAM	—	—	—	Version3.05F or later
	Addition of the CompactFlash card drive as the parameter read, write and verification target drive	12042 or later	×	×	Version3.05F or later
	Supporting continuation error clear	12042 or later	×	×	Version3.05F or later
	Change of the display format for the special direct buffer register and intelligent function module buffer memory in Device monitoring utility.	—	—	—	Version3.05F or later

—: Not related to the serial No. or the software version

Memo

INDEX

WARRANTY

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

1. Gratis Warranty Term and Gratis Warranty Range

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

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

[Gratis Warranty Term]

The gratis warranty term of the product shall be for one year after the date of purchase or delivery to a designated place. Note that after manufacture and shipment from Mitsubishi, the maximum distribution period shall be six (6) months, and the longest gratis warranty term after manufacturing shall be eighteen (18) months. The gratis warranty term of repair parts shall not exceed the gratis warranty term before repairs.

[Gratis Warranty Range]

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

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

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

2. Onerous repair term after discontinuation of production

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

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

3. Overseas service

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

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

Regardless of the gratis warranty term, Mitsubishi shall not be liable for compensation to:

(1) Damages caused by any cause found not to be the responsibility of Mitsubishi.

(2) Loss in opportunity, lost profits incurred to the user by Failures of Mitsubishi products.

(3) Special damages and secondary damages whether foreseeable or not, compensation for accidents, and compensation for damages to products other than Mitsubishi products.

(4) Replacement by the user, maintenance of on-site equipment, start-up test run and other tasks.

5. Changes in product specifications

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

INFORMATION AND SERVICES

For further information and services, please contact your local Mitsubishi Electric sales office or representative.
Visit our website to find our locations worldwide.

MITSUBISHI ELECTRIC Factory Automation Global Website
Locations Worldwide
www.MitsubishiElectric.com/fa/about-us/overseas/

TRADEMARKS

COPYRIGHTS

SH(NA)-080766ENG-R(2311)KWIX
MODEL: Q12DCCPU-V-U-HK-E
MODEL CODE: 13JZ17

MITSUBISHI ELECTRIC CORPORATION

HEAD OFFICE: TOKYO BLDG., 2-7-3, MARUNOUCHI, CHIYODA-KU, TOKYO 100-8310, JAPAN
NAGOYA WORKS: 1-14, YADA-MINAMI 5-CHOME, HIGASHI-KU, NAGOYA 461-8670, JAPAN

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

Specifications subject to change without notice.