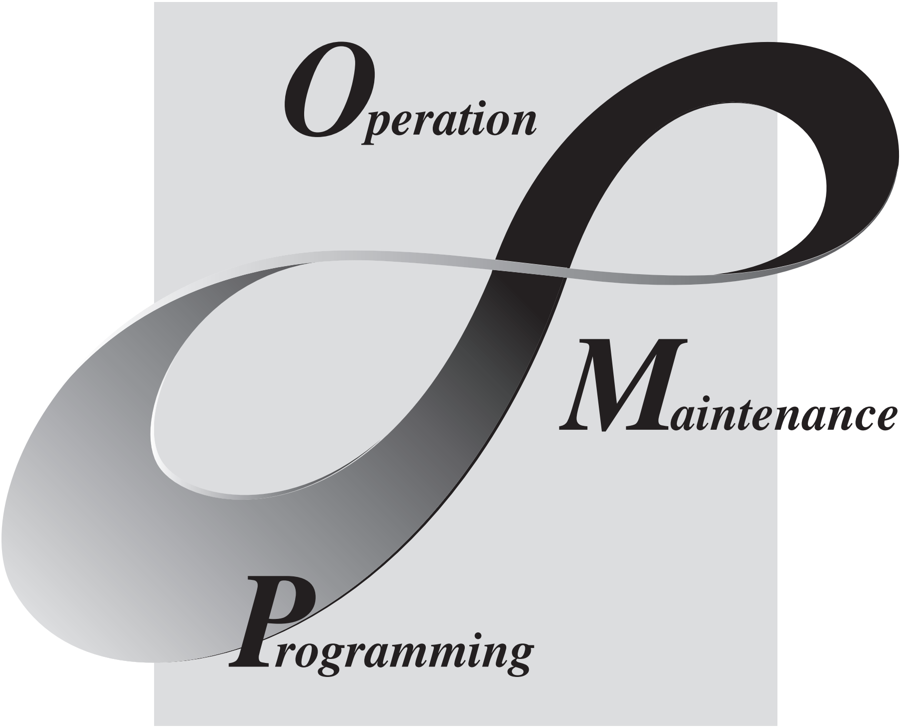
MX Component Version 4

MITSUBISHI

Programming Manual



MELSOFT
Integrated FA Software

SW4DNC-ACT-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.

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 given under "⚠ CAUTION" may lead to serious consequences.
Make sure that the end users read this manual and then keep the manual in a safe place for future reference.

[Design Instructions]

⚠WARNING

● When data change, program change, or status control is performed from a personal computer to a running programmable controller, create an interlock circuit outside the programmable controller to ensure that the whole system always operates safely.
Furthermore, for the online operations performed from a personal computer to a programmable controller CPU, the corrective actions against a communication error due to such as a cable connection fault should be predetermined as a system.

⚠CAUTION

● The online operations performed from a personal computer to a running programmable controller CPU (forced output and operating status changes) must be executed after the manual has been carefully read and the safety has been ensured.
The operation failure may cause the injury or machine damage.

●CONDITIONS OF USE FOR THE PRODUCT●

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

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

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

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

OPERATING CONSIDERATIONS

This section explains the considerations in the following order.

1) Considerations of operating system and personal computer to be used
2) Considerations of installation and uninstallation
3) Programmable controller CPU-related considerations
4) Considerations for using other MELSOFT products
5) Considerations for using Ethernet modules
6) Considerations for using CC-Link modules
7) Considerations for using serial communication modules
8) Considerations of modem communication
9) Considerations of programming
10) Considerations for using Microsoft® Excel®
11) Considerations for using Microsoft® Access®
12) Considerations for using VBScript

Considerations of operating system and personal computer to be used

(1) Restrictions applied when a user without Administrator's authority operates MX Component

Note that the following restrictions are applied when a user without Administrator's authority operates MX Component.

(a) Communication Setup Utility

- The logical station number cannot be created, changed, or deleted.
- Communication settings cannot be imported.
- This utility cannot be started up if the communication settings are set using MX Component earlier than Version 3.00A.[*1]

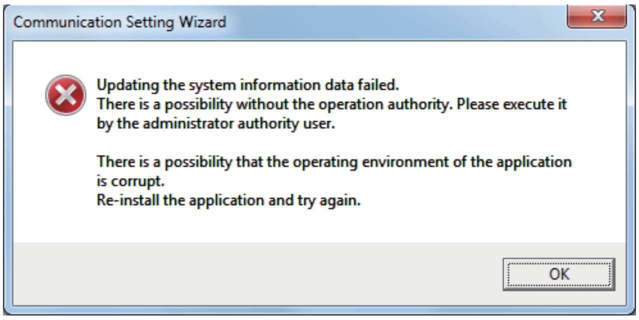
(b) PLC Monitor Utility

- This utility cannot be started up if the communication settings are set using MX Component earlier than Version 3.00A.[*1]
- Device registration cannot be performed on the <<Entry Device>> tab.

(c) Communication board

- Various settings cannot be set on the CC-Link IE Controller Network, CC-Link IE Field Network, MELSECNET/H, and CC-Link board utilities.

*1 : If the following error message is displayed, start up and close the utility as a user with Administrator's authority. This operation enables a user without Administrator's authority to start up the utility.


Communication Setting Wizard

Updating the system information data failed.
There is a possibility without the operation authority. Please execute it by the administrator authority user.

There is a possibility that the operating environment of the application is corrupt.
Re-install the application and try again.

(2) Resume and other functions of personal computer

A communication error may occur when communicating with the programmable controller CPU after setting the resume function, suspend setting, power-saving function, and/or standby mode of the personal computer. Therefore, do not set the above functions when communicating with the programmable controller CPU.

Considerations of installation and uninstallation

(1) Installation

When performing overwrite installation, install the software in the same folder where it is installed previously.

(2) Start menu

When MX Component is uninstalled, the item may remain in the start menu.
In this case, restart the personal computer.

Programmable controller CPU-related considerations

(1) Considerations for performing USB communication

ON/OFF of a programmable controller CPU during communications with the programmable controller CPU may cause a communication error which cannot be recovered.
If it is not recovered, completely disconnect the USB cable and then reconnect it after 5 or more seconds.
(If this error occurs at the initial communication after the above operation, the function will be performed properly in and after the second communications.)

(2) Time data of programmable controller CPU

(a) For QCPU (Q mode), LCPU, and FXCPU, the time data setting can be set if the programmable controller CPU is in the RUN status.

(b) For QCPU (Q mode) and LCPU, the setting can be set regardless of the ON/OFF status of the time setting device "SM1028".

(c) For FXCPU, the setting can be set for FX1N (clock built-in), FX1NC (clock built-in), FX1S (clock built-in), FX2N (clock built-in), FX2NC (when RTC cassette is installed), FXU (when RTC cassette is installed), FX2C (when RTC cassette is installed), and FX3G (clock built-in) only.

(d) Note that an error for transfer time occurs in the time setting.

(3) Restrictions on using FXCPU

(a) When FXCPU is used, the TN devices (timer present values) or CN devices (counter present values) cannot be accessed if the device numbers specified are split across 199 or earlier and 200 or later.

(b) Since FXCPU does not feature the PAUSE switch as the programmable controller CPU, an error is returned if remote pause is specified in SetCpuStatus.

(c) Note that specifying the first I/O number of a non-existing module and executing the WriteBuffer() method will not return an error.

(d) For the index registers (Z, V) of FXCPU, data cannot be written to 2 or more consecutive points using WriteDeviceBlock(). (Data may be written to only one point.)

(4) Serial communication function of Q00UJ/Q00/Q00U/Q01/Q01U/Q02UCPU[*1]

[*1] : In this section, "serial communication function compatible CPU" indicates Q00UJ/Q00/Q00U/Q01/Q01U/Q02UCPU.

When the following conditions are all satisfied, communication between the personal computer and the serial communication function compatible CPU is set at 9600bps speed.

- 1)The serial communication function of the connected CPU is valid.
- 2)The transmission speed settings differ between the personal computer and the serial communication function compatible CPU side.

To increase the communication speed, match the transmission speed of personal computer with that of serial communication function compatible CPU.

(5) Considerations for using built-in Ethernet port CPU

When resetting the programmable controller CPU during TCP/IP connection establishment (during opening) using MX Component, a communication error or receive error occurs at subsequent communication.
In this case, perform the close processing in the application that uses MX Component, and perform the open processing again.

(6) Considerations for using QSCPU

In order to protect the safety programmable controller system, functions to write data to buffer memory, to write/set devices, and to write time data cannot be executed.

Considerations for using other MELSOFT products

(1) Considerations for performing GX Simulator communication

Before executing PLC Monitor Utility, Communication Setup Utility, or a user program, check that GX Simulator and GX Developer are operating.
If GX Simulator or GX Developer is terminated while the user program is running, the user program will not be terminated normally.

Considerations for using Ethernet modules

(1) Resetting programmable controller CPU during TCP/IP connection establishment

When resetting the programmable controller CPU during TCP/IP connection establishment (during opening) using MX Component, a communication error or receive error occurs at subsequent communication.
In this case, perform the close processing in the application that uses MX Component, and perform the open processing again.

(2) Target existence check starting interval of Ethernet module

If the close processing (Close) is executed from the personal computer, the Ethernet module may not perform the close processing (Close). One of its causes is the cable disconnection.
If the open processing (Open) is executed from the personal computer with the Ethernet module not executing the close processing (Close), the open processing (Open) from the personal computer is not terminated normally until the Ethernet module performs a target existence check and executes the close processing (Close).
When terminating the open processing (Open) early from the personal computer, shorten the target existence check starting interval setting of the Ethernet module.
(The default setting of target existence check starting interval of the Ethernet module is 10 minutes.)

(3) Replacement of Ethernet module

If the Ethernet modules are changed during Ethernet communication due to debugging, failure or the like, the other node (personal computer) must be restarted.
(Because the Ethernet addresses (MAC addresses) differ between devices.)

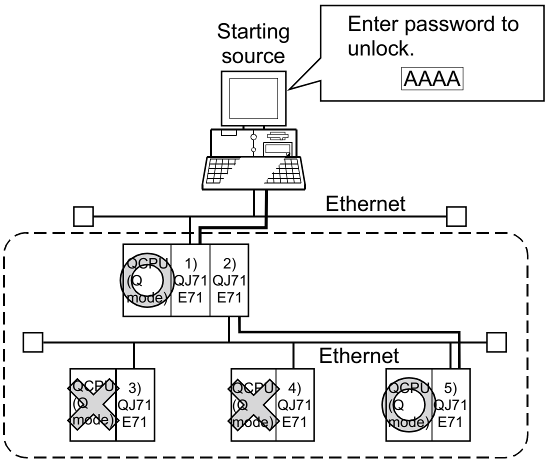
(4) Simultaneous access when using Q series-compatible Ethernet module

The following conditions should be satisfied when communication is performed simultaneously from multiple personal computers to the same module using the TCP/IP protocol.

- Q series-compatible E71 module (except QJ71E71-100) whose first five digits of the serial number is "02122" or higher and whose function version is B or later.
- Using GX Developer Version 6.05F or later, set "MELSOFT connection" in the Ethernet parameter [open system].

(5) Unlocking password when using QJ71E71

The range where the password can be unlocked by remote operation is up to the connection target station. If the password is also set on the lower layer, communication cannot be performed with the programmable controller CPU on the lower layer.



No.	Remote Password
1)	With setting (AAAA)
2)	Without setting
3)	With setting (AAAA)
4)	With setting (BBBB)
5)	Without setting

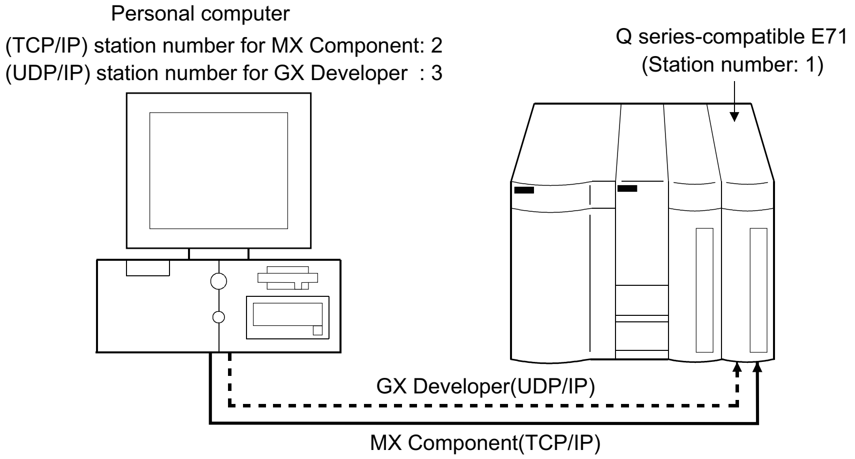
○ : Accessible
× : Inaccessible

1) Unlocking QJ71E71 password enables access to Programmable controller CPUs in this range.

(6) Ethernet communication

(a) The communication line is disconnected if the CPU becomes faulty or the Ethernet module is reset during Ethernet communication (when the protocol is TCP/IP).
In this case, perform the line close processing (Close) and then perform the reopen processing (Open).

(b) When two different communication systems (protocols) are used to access from one personal computer to one Q series-compatible E71, two station numbers for TCP/IP and for UDP/IP must be set. However, it is not required to set different station numbers for TCP/IP and for UDP/IP when using MX Component Version 3 or later and Q series-compatible E71 with serial number 05051 or higher.
Example When MX Component uses TCP/IP and GX Developer uses UDP/IP



Set different station numbers as the (TCP/IP) station number for MX Component and (UDP/IP) station number for GX Developer. If they are set to the same station number, an error will occur on the Ethernet module side.

Considerations for using CC-Link modules

(1) Software version of CC-Link master/local module

For CC-Link master/local modules used in CC-Link communication, use modules of software version "N" or later. Modules of software version "M" or earlier do not operate normally.

Considerations for using serial communication modules

(1) Serial communication

(a) On any serial communication modules, remote "PAUSE" operation will result in an error for all connections.

(b) The FX extended port is required when performing the serial communication using FX$_{0N}$, FX$_{1S}$, FX$_{1N(C)}$, FX$_{3G}$, or FX$_{3U(C)}$CPU.

(2) Considerations for connecting personal computer and serial communication module

(a) When using QJ71C24-R2 of function version A
An MX Component application can use only either of CH1 and CH2.
When the MELSOFT product (GX Developer, GOT, or the like) is using one channel, the application cannot use the other channel.
When QJ71C24-R2 of function version B is used, the application can use both channels.

Considerations of modem communication

(1) Simultaneous modem communications

The simultaneous modem communications using MX Component and other applications (GX Developer or the like) cannot be performed.
Do not perform a modem communication using other applications during a modem communication using MX Component.
If modem communications are simultaneously performed using MX Component and other application, this will result in a communication error, disconnection of telephone line or similar problem.

(2) Considerations for using telephone line

(a) Do not use the call-waiting phone line.
On the call-waiting phone line, data corruption, telephone line disconnection, or similar problem may occur due to interrupt reading sounds.

(b) Do not connect the line to master/slave phones.
If the handset of the slave phone is lifted while the telephone line is connecting to the master/slave phones, the telephone line may be disconnected.

(c) Use an analog 2 wire type telephone line.
When using a digital line, use a terminal adapter.
When the telephone line is 4 wire type, the line may not be connected depending on the wiring type of the modular jack.
For the 4 wire type, conduct connection tests in advance to check for connections.

(3) Considerations for using cellular phone

(a) Modem for radio communication using a cellular phone

Although the modem name is different according to the manufacturer, the modem is generically referred to as the cellular phone communication unit in this manual.

Select the model of the cellular phone communication unit according to the cellular phone used.

For details, contact the company of your cellular phone.

(b) Cellular phone without auto answer function

For the cellular phone without auto answer function, use a cellular phone communication unit that features the ANS/ORG/TEL select switch.

If the cellular phone communication unit does not have the ANS/ORG/TEL select switch, the line cannot be connected.

The line connection procedure is different according to the cellular phone company and cellular phone model.

For details, contact the manufacturer of your cellular phone.

Considerations of programming

(1) Sample programs, test programs, and sample sequence programs

(a) Sample programs and test programs

Sample programs are included for references when creating user programs.

Test programs are included for conducting communication tests.

Use the programs with your responsibility.

(b) Sample sequence programs

Sample sequence programs included in MX Component require modifications according to the system configuration and parameter settings.

Modify the program to suit the system.

Use the programs with your responsibility.

(2) Forced termination of processes during communication

If communication is performed with the same type of control open for multiple processes, forcing one process to be terminated by Task Manager or the like may stop the other processes at the communication function execution area.

(3) Error at communication start

A communication error may occur within the preset time-out period at a communication start, for example, when the communication diagnostic button is pressed, when a monitoring is started, or when any function is executed.

These errors are assumed to be detected before a time-out error.

(Example: When the communication cable is not connected, when the programmable controller power is OFF)

(4) CheckDeviceString

Do not use the CheckDeviceString method of ACT control.

(5) ActUMsg control and ActUWzd control

Installing MX Component registers the ActUMsg control and the ActUWzd control, however, do not use them.

(6) Considerations for using Ethernet modules

(a) Provide an interval longer than the sequence scan time of the Ethernet module mounted station for a period from when the Open method is executed until the Close method is executed.

(b) Provide an interval of at least 500ms for a period from when the Close method is executed until the Open method is executed again.

(7) Considerations for executing the Disconnect function

If a telephone line cannot be disconnected by executing the Disconnect function for some reason, power OFF the modem being used to forcibly disconnect the telephone line.

Considerations for using Microsoft® Excel®

(1) Considerations for using Excel VBA

If the page feed preview function is set in the application that uses Excel VBA, a memory leak or operating system basic operation (file operation, printing, or the like) failure may occur.

(2) Considerations for using Microsoft® Excel®

(a) Occasionally, controls may not be pasted to Excel.
This symptom occurs if the cache file (temporary file) of Excel remains.
In such a case, perform the operation in the following procedure.

Operating procedure

1. **Close Excel.**
2. **Delete "*.exd" in the Excel 8.0 folder of the temp folders.** [*1, *2]
3. **Restart Excel.**

*1 : The location of temp folder differs according to the operating system.

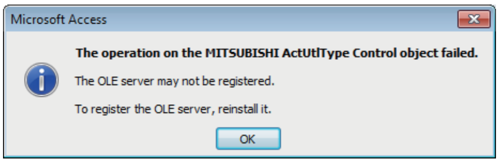
*2 : When the corresponding folder and file are not displayed, set the settings in the folder option setting to display all files and folders.

(b) Resizing of ACT control in Excel does not affect the operation of MX Component.
To restore the size, set the Height and Width properties of ACT control to "24".

Considerations for using Microsoft® Access®

(1) Considerations for using Microsoft® Access®

(a) When the ACT control is pasted to an Access form and the ACT control is double-clicked or the custom control in the property is selected, the following error message is displayed. However, this does not affect the operation of ACT control.
(An error message other than the following message may be displayed.)



Microsoft Access

The operation on the MITSUBISHI ActUtlType Control object failed.

The OLE server may not be registered.

To register the OLE server, reinstall it.

(b) When the ACT control is pasted and the property is displayed, the displayed property name may be collapsed. This symptom only occurs on the display of the property, and this does not affect the functions of the property.

(c) Resizing of ACT control in Access does not affect the operation of MX Component.
To restore the size, set the Height and Width properties of ACT control to "24".

Considerations for using VBScript

(1) Security of the Internet/intranet when using VBScript

MX Component does not feature the Internet/intranet security function.
When the security function is required, set the setting on the user side.

INTRODUCTION

Thank you for your patronage. We appreciate your purchase of the Mitsubishi integrated FA software, MELSOFT series.
This manual is designed for users to understand operations of MX Component.
Before using the product, thoroughly read this manual and related manuals to develop full familiarity with the functions and performance of MX Component and supported modules to ensure correct use.

RELATED MANUALS

The manuals related to this product are shown below.

Refer to the following tables when ordering required manuals.

Manual name < Manual number, model code >	Description
MX Component Version 4 Operating Manual <SH-081084ENG, 13JU75>	Explains the programming procedures, detailed explanations and error codes of the ACT controls.
Type Q80BD-J61BT11N/Q81BD-J61BT11 CC-Link System Master/ Local Interface Board User's Manual (For SW1DNC-CCBD2-B) <SH-080527ENG, 13JR77>	Explains the system configuration, specifications, functions, handling, wiring, and troubleshooting of the type Q80BD-J61BT11N/ Q81BD-J61BT11 CC-Link system master/local interface board.
MELSECNET/H Interface Board User's Manual (For SW0DNC-MNETH-B) <SH-080128, 13JR24>	Explains the system configuration, specifications, functions, handling, wiring, and troubleshooting of the MELSEC/H board.
CC-Link IE Controller Network Interface Board User’s Manual (For SW1DNC-MNETG-B) <SH-080691ENG, 13JZ02>	Explains the system configuration, specifications, functions, handling, wiring, and troubleshooting of the CC-Link IE Controller Network board.
CC-Link IE Field Network Interface Board User’s Manual (For SW1DNC-CCIEF-B) <SH-080980ENG, 13JZ58>	Explains the system configuration, specifications, functions, handling, wiring, and troubleshooting of the CC-Link IE Field Network board.
GX Simulator Version 7 Operating Manual <SH-080468ENG, 13JU51>	Explains the setting and operating method for monitoring the device memory and simulating the machine side operations using GX Simulator.
GX Works2 Version 1 Operating Manual (Common) <SH-080779ENG, 13JU63>	Explains the system configuration of GX Works2 and the functions common to a Simple project and Structured project such as parameter setting, operation method for the online function.

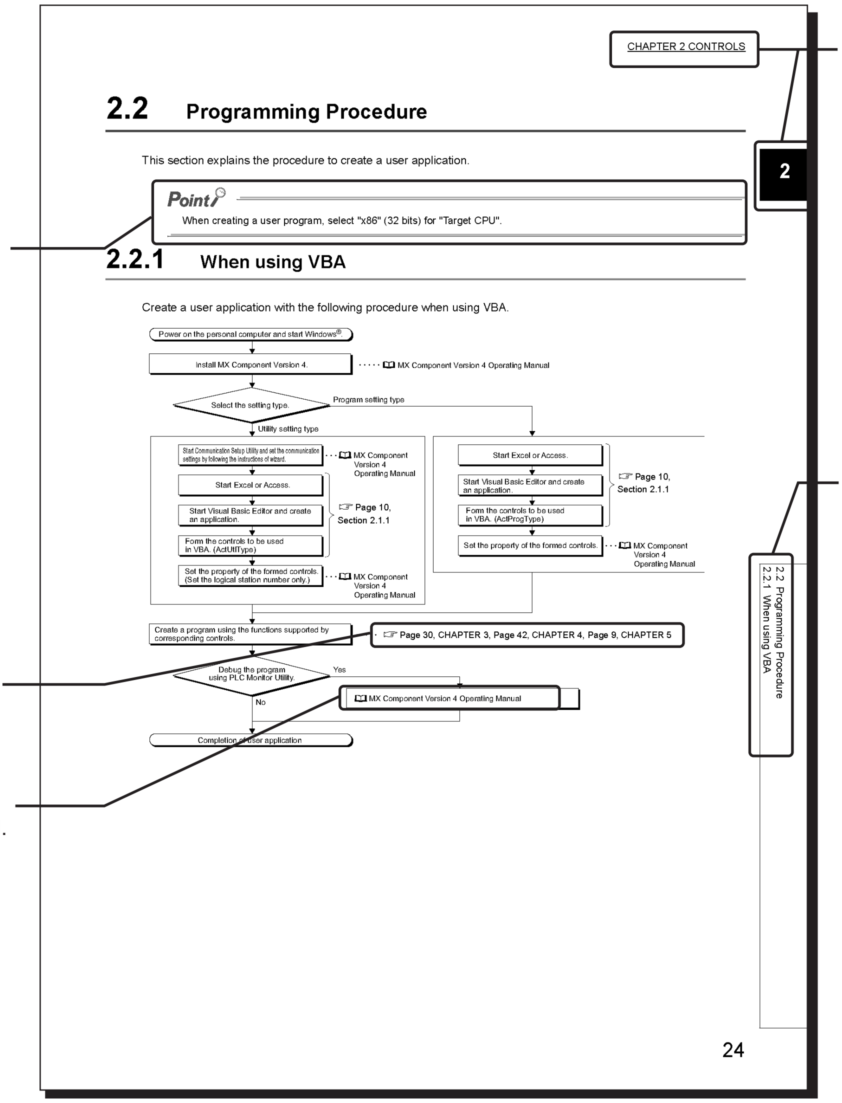
Remark

MX Component Version 4 Operating Manual is included on the CD-ROM of the software package in a PDF file format. Manuals in printed form are sold separately for single purchase. Order a manual by quoting the manual number (model code) listed in the table above.

CONTENTS

HOW TO READ THIS MANUAL

The following explains the page composition and symbols in this manual.
The content of the example page used here are different from the actual content for the intention of explaining how to use this manual.



CHAPTER 2 CONTROLS

2

2.2 Programming Procedure

This section explains the procedure to create a user application.

Point

When creating a user program, select "x86" (32 bits) for "Target CPU".

2.2.1 When using VBA

Create a user application with the following procedure when using VBA.

Power on the personal computer and start Windows®.

Install MX Component Version 4. · · · · · 📖 MX Component Version 4 Operating Manual

Select the setting type.

Utility setting type

Start Communication Setup Utility and set the communication settings by following the instructions of wizard. · · · 📖 MX Component Version 4 Operating Manual

Start Excel or Access.

Start Visual Basic Editor and create an application.

Fom the controls to be used in VBA. (ActUtlType)

☞ Page 10, Section 2.1.1

Set the property of the formed controls. (Set the logical station number only.) · · · 📖 MX Component Version 4 Operating Manual

Program setting type

Start Excel or Access.

Start Visual Basic Editor and create an application.

Fom the controls to be used in VBA. (ActProgType)

☞ Page 10, Section 2.1.1

Set the property of the formed controls. · · · 📖 MX Component Version 4 Operating Manual

Create a program using the functions supported by corresponding controls. · ☞ Page 30, CHAPTER 3, Page 42, CHAPTER 4, Page 9, CHAPTER 5

Debug the program using PLC Monitor Utility.

Yes

📖 MX Component Version 4 Operating Manual

No

Completion of user application

2.2 Programming Procedure
2.2.1 When using VBA

24

Point indicates the particular attention.

Remark indicates the useful tip.

☞ indicates the reference page.

📖 indicates the reference manual.

Indicates the chapter of currently open page.

Indicates the section of currently open page.

The following shows the symbols used in this manual with descriptions and examples.

Notation	Description	Example
[ ]	Menu name on a menu bar	[Tools]⇨[Property]
<< >>	Tab name on a screen	<<.NET>> tab
" "	Item name on a screen	"References"
▭	Button on a screen	OK button

TERMS

This manual uses the terms listed in the following table unless otherwise noted.

Term	Description
MX Component	Generic product name for SWnDNC-ACT-E and SWnDNC-ACT-EA (n: version) -EA indicates a volume-license product.
Personal computer	Generic term for personal computers on which Windows® operates
PC CPU module	Abbreviation for MELSEC Q series-compatible PC CPU module (CONTEC CO., LTD. product)
GX Developer	Generic product name for SWnD5C-GPPW-E, SWnD5C-GPPW-EA, SWnD5C-GPPW-EV, and SWnD5C-GPPW-EVA (n: version) -EA indicates a volume-license product, and -EV an updated product.
GX Works2	Generic product name for SWnDNC-GXW2 (n: version)
GX Simulator	Generic product name for SWnD5C-LLT-E, SWnD5C-LLT-EA, SWnD5C-LLT-EV, and SWnD5C-LLT-EVA (n: version) -EA means a volume-license product, and -EV an updated product.
MELSECNET/H board	Generic term for Q80BD-J71LP21S-25 and Q80BD-J71BR11 Abbreviation for MELSECNET/H interface board
CC-Link IE Controller Network board	Generic term for Q80BD-J71GP21-SX and Q80BD-J71GP21S-SX Abbreviation for CC-Link IE Controller Network interface board
CC-Link IE Field Network board	Abbreviation for Q81BD-J71GF11-T2 CC-Link IE Field Network interface board
CC-Link board	Generic term for Q80BD-J61BT11N and Q81BD-J61BT11 Abbreviation for CC-Link system master/local interface board
QCPU	Generic term for Q00J, Q00UJ, Q00, Q00U, Q01, Q01U, Q02, Q02H, Q02PH, Q02U, Q03UD, Q03UDE, Q04UDH, Q04UDEH, Q06H, Q06PH, Q06UDH, Q06UDEH, Q10UDH, Q10UDEH, Q12H, Q12PH, Q12PRH, Q13UDH, Q13UDEH, Q20UDH, Q20UDEH, Q25H, Q25PH, Q25PRH, Q26UDH, Q26UDEH, Q50UDEH, and Q100UDEH
Built-in Ethernet port QCPU	Generic term for Q03UDE, Q04UDEH, Q06UDEH, Q10UDH, Q10UDEH, Q13UDEH, Q26UDEH, Q50UDEH, and Q100UDEH
LCPU	Generic term for L02, L26CPU-BT
Built-in Ethernet port CPU	Generic term for built-in Ethernet port QCPU and LCPU
FXCPU	Generic term for FX$_{0}$, FX$_{0S}$, FX$_{0N}$, FX$_{1}$, FX$_{1N}$, FX$_{1NC}$, FX$_{1S}$, FX$_{U}$, FX$_{2C}$, FX$_{2N}$, FX$_{2NC}$, FX$_{3G}$, FX$_{3U}$, and FX$_{3UC}$
Q motion CPU	Generic term for Q172, Q173, Q172H, Q173H, Q172D, Q173D, Q172DS, and Q173DS
QSCPU	Abbreviation for a safety CPU module (QS001CPU)
C Controller CPU	Abbreviation for Q12DCCPU-V
Programmable controller CPU	Generic term for QCPU (Q mode), LCPU, FXCPU, Q motion CPU, QSCPU, and C Controller CPU
Q series-compatible C24	Generic term for QJ71C24, QJ71C24-R2, QJ71C24N, QJ71C24N-R2, and QJ71C24N-R4
L series-compatible C24	Generic term for LJ71C24 and LJ71C24-R2
FX extended port	Generic term for FX$_{0N}$-485ADP, FX$_{2NC}$-485ADP, FX$_{1N}$-485-BD, FX$_{2N}$-485-BD, FX$_{3G}$-485-BD, FX$_{3U}$-485-BD, and FX$_{3U}$-485ADP
Serial communication module	Generic term for Q series-compatible C24, L series-compatible C24, and FX extended port
Q series-compatible E71	Generic term for QJ71E71, QJ71E71-B2, QJ71E71-B5, and QJ71E71-100
CC-Link IE Field Network Ethernet adapter module	Abbreviation for NZ2GF-ETB CC-Link IE Field Network Ethernet adapter module
CC-Link G4 module	Abbreviation for AJ65BT-G4-S3 GPP function peripheral connection module
GOT	Abbreviation for Graphic Operation Terminal
GOT1000	Abbreviation for Graphic Operation Terminal GOT1000 series
Serial communication	Abbreviation for communication with programmable controller CPU using the serial communication module

Term	Description
Ethernet communication	Abbreviation for communication by connecting the personal computer to Ethernet module or the built-in Ethernet port CPU
CPU COM communication	Abbreviation of communication performed by connecting the personal computer to the RS-232 or RS-422 connector of programmable controller CPU
CPU USB communication	Abbreviation for communication by connecting personal computer to the USB connector of QCPU (Q mode), LCPU
MELSECNET/H communication	Abbreviation for communication with programmable controller CPU using MELSECNET/H board
CC-Link IE Controller Network communication	Abbreviation for communication with programmable controller CPU using CC-Link IE Controller Network board
CC-Link IE Field Network communication	Abbreviation for communication with programmable controller CPU using CC-Link IE Field Network board
CC-Link communication	Abbreviation for communication with programmable controller CPU using CC-Link board
CC-Link G4 communication	Abbreviation for communication with programmable controller CPU using CC-Link G4 module
Q series bus communication	Abbreviation for communication with programmable controller CPU on the same base using PC CPU module
GX Simulator communication	Abbreviation for communication with GX Simulator
GX Simulator2 communication	Abbreviation for communication using the simulation functions of GX Works2
Modem communication	Abbreviation for communication with programmable controller CPU via modems using Q series-compatible C24, L series-compatible C24 or FXCPU
Gateway function communication	Abbreviation for communication with programmable controller CPU and third-party programmable controllers using the gateway functions of GOT
GOT transparent communication	Abbreviation for communication with programmable controller CPU using the GOT transparent functions of GOT
Utility setting type	A development type for creating user programs using Communication Setup Utility
Program setting type	A development type for creating user programs without using Communication Setup Utility
ACT control	A development type for creating user programs using Communication Setup Utility
.NET control	A development type for creating user programs without using Communication Setup Utility
Redundant CPU	Generic term for Q12PRHCPU and Q25PRHCPU
Redundant type extension base unit	Abbreviation for Q65WRB extension base unit for redundant system
Windows® XP	Generic term for Microsoft® Windows® XP Professional Operating System and Microsoft® Windows® XP Home Edition Operating System
Windows Vista®	Generic term for Microsoft® Windows Vista® Home Basic Operating System, Microsoft® Windows Vista® Home Premium Operating System, Microsoft® Windows Vista® Business Operating System, Microsoft® Windows Vista® Ultimate Operating System, and Microsoft® Window Vista® Enterprise Operating System
Windows® 7	Generic term for Microsoft® Windows® 7 Starter Operating System, Microsoft® Windows® 7 Home Premium Operating System, Microsoft® Windows® 7 Professional Operating System, Microsoft® Windows® 7 Ultimate Operating System, and Microsoft® Windows® 7 Enterprise Operating System "32-bit Windows® 7" is used for indicating 32-bit version only, and "64-bit Windows® 7" is used for indicating 64-bit version only.
Excel	Abbreviation for Microsoft® Excel® 2003, Microsoft® Excel® 2007, 32-bit Microsoft® Excel® 2010
Access	Abbreviation for Microsoft® Access® 2003, Microsoft® Access® 2007, 32-bit Microsoft® Access® 2010
Visual Basic® .NET	Generic term for Visual Basic version Visual Studio® 2005, Visual Studio® 2008, and Visual Studio® 2010
Visual C++® .NET	Abbreviation for creation of an application using .NET Framework

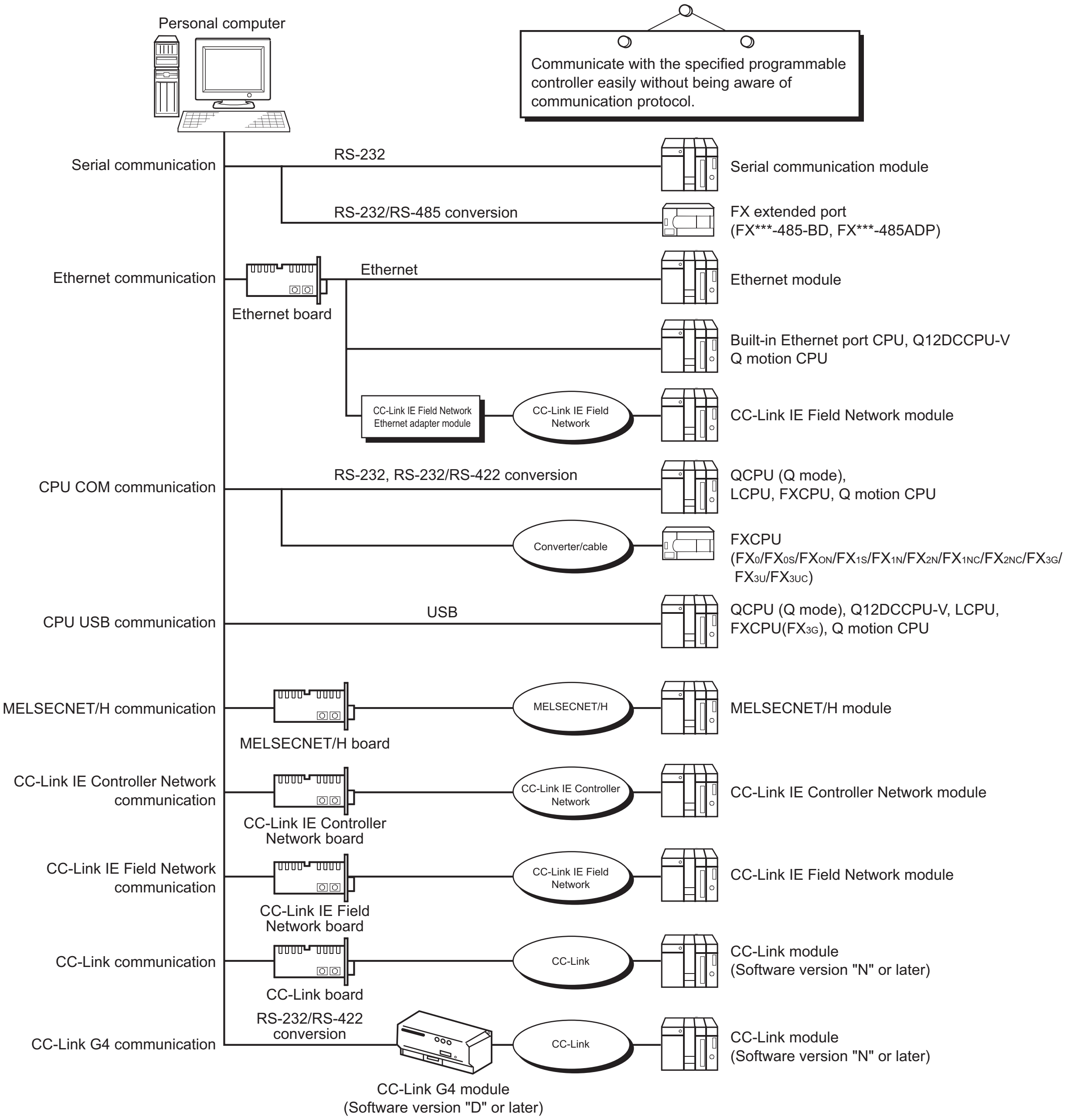
Term	Description
Visual C++® .NET(MFC)	Abbreviation for creation of an application using MC/ATL/Win32

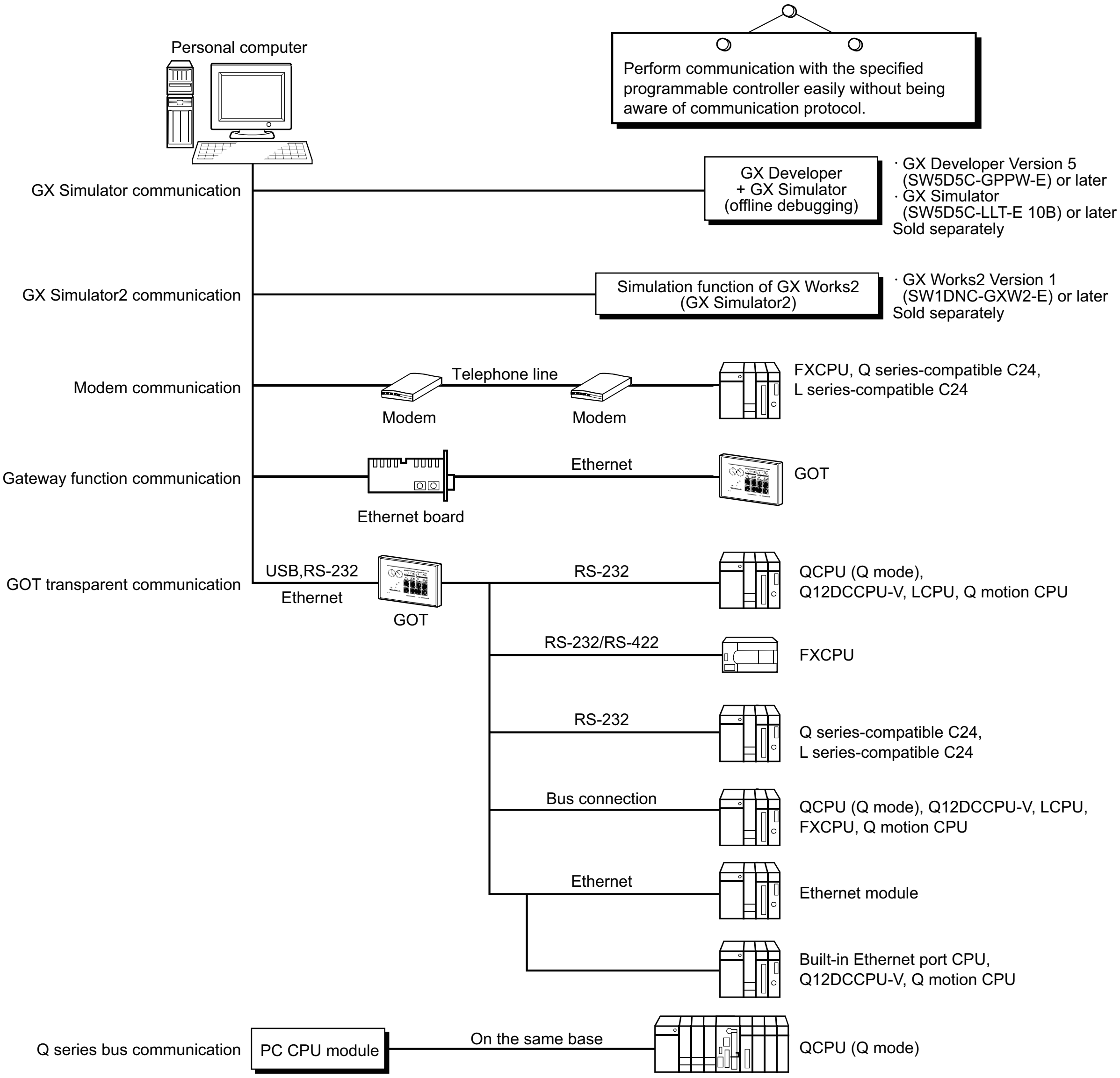
CHAPTER 1 OVERVIEW

This manual explains the procedure and the error codes when creating programs with MX Component Version 4.

1.1 Outline of Controls

This section explains the outline of controls supported by MX Component.
These controls are used to create user programs to communicate with a programmable controller CPU. A user can communicate with programmable controller CPUs without being aware of the hardware and the communication protocol on the other end.


Personal computer
Communicate with the specified programmable controller easily without being aware of communication protocol.
Serial communication
RS-232
Serial communication module
RS-232/RS-485 conversion
FX extended port (FX***-485-BD, FX***-485ADP)
Ethernet communication
Ethernet board
Ethernet
Ethernet module
Built-in Ethernet port CPU, Q12DCCPU-V
Q motion CPU
CC-Link IE Field Network Ethernet adapter module
CC-Link IE Field Network
CC-Link IE Field Network module
CPU COM communication
RS-232, RS-232/RS-422 conversion
QCPU (Q mode), LCPU, FXCPU, Q motion CPU
Converter/cable
FXCPU (FX$_0$/FX$_{0S}$/FX$_{0N}$/FX$_{1S}$/FX$_{1N}$/FX$_{2N}$/FX$_{1NC}$/FX$_{2NC}$/FX$_{3G}$/ FX$_{3U}$/FX$_{3UC}$)
CPU USB communication
USB
QCPU (Q mode), Q12DCCPU-V, LCPU, FXCPU(FX$_{3G}$), Q motion CPU
MELSECNET/H communication
MELSECNET/H board
MELSECNET/H
MELSECNET/H module
CC-Link IE Controller Network communication
CC-Link IE Controller Network board
CC-Link IE Controller Network
CC-Link IE Controller Network module
CC-Link IE Field Network communication
CC-Link IE Field Network board
CC-Link IE Field Network
CC-Link IE Field Network module
CC-Link communication
CC-Link board
CC-Link
CC-Link module (Software version "N" or later)
CC-Link G4 communication
RS-232/RS-422 conversion
CC-Link G4 module (Software version "D" or later)
CC-Link
CC-Link module (Software version "N" or later)


Personal computer
Perform communication with the specified programmable controller easily without being aware of communication protocol.
GX Simulator communication
GX Developer + GX Simulator (offline debugging)
· GX Developer Version 5 (SW5D5C-GPPW-E) or later
· GX Simulator (SW5D5C-LLT-E 10B) or later
Sold separately
GX Simulator2 communication
Simulation function of GX Works2 (GX Simulator2)
· GX Works2 Version 1 (SW1DNC-GXW2-E) or later
Sold separately
Modem communication
Modem
Telephone line
Modem
FXCPU, Q series-compatible C24, L series-compatible C24
Gateway function communication
Ethernet board
Ethernet
GOT
GOT transparent communication
USB,RS-232
Ethernet
GOT
RS-232
QCPU (Q mode), Q12DCCPU-V, LCPU, Q motion CPU
RS-232/RS-422
FXCPU
RS-232
Q series-compatible C24, L series-compatible C24
Bus connection
QCPU (Q mode), Q12DCCPU-V, LCPU, FXCPU, Q motion CPU
Ethernet
Ethernet module
Built-in Ethernet port CPU, Q12DCCPU-V, Q motion CPU
Q series bus communication
PC CPU module
On the same base
QCPU (Q mode)

1.2 Control and Function Lists

This section shows the lists of controls and functions.

1.2.1 Control list

The following tables show the controls included in each DLL supported by MX Component.

(1) ACT control

The following table shows the ActiveX controls supported by MX Component.
Data can be accessed using devices.

DLL name	Included control name		Application
	For VB, VC++, VBA	For VBScript	
ActUtlType.dll	ActUtlType	ActMLUtlType*1	The utility setting type control which is used to create a user program using Communication Setup Utility.
ActProgType.dll	ActProgType	ActMLProgType*1	The program setting type control which is used to create a user program without using Communication Setup Utility.
ActSupportMsg.dll	ActSupportMsg	ActMLSupportMsg	Used for the troubleshooting function.

*1 : Communication is disabled if the communication path is a modem.

(2) .NET control

The following table shows the .NET controls supported by MX Component.
Data can be accessed using labels.

DLL name	Included control name		Application
	For VB, VC++	For VBScript, VBA	
DotUtlType.dll	DotUtlType	—	The utility setting type control which is used to create a user program using Communication Setup Utility.
DotSupportMsg.dll	DotSupportMsg	—	Used for the troubleshooting function.

1.2.2 Function list

The following table shows the features of the functions and the functions that can be used for the controls.

Function name	Feature	Refer
Open	Open a communication line.	Page 214, Section 5.2.1, Page 287, Section 5.3.1
Close	Close a communication line.	Page 216, Section 5.2.2, Page 288, Section 5.3.2
ReadDeviceBlock	Read devices in bulk. (4-byte data)	Page 217, Section 5.2.3, Page 289, Section 5.3.3
WriteDeviceBlock	Write devices in bulk. (4-byte data)	Page 220, Section 5.2.4, Page 292, Section 5.3.4
ReadDeviceRandom	Read devices randomly. (4-byte data)	Page 223, Section 5.2.5, Page 295, Section 5.3.5
WriteDeviceRandom	Write devices randomly. (4-byte data)	Page 226, Section 5.2.6, Page 300, Section 5.3.6
SetDevice	Set one point of device. (4-byte data)	Page 229, Section 5.2.7, Page 305, Section 5.3.7
GetDevice	Acquire data of one point of device. (4-byte data)	Page 231, Section 5.2.8, Page 307, Section 5.3.8
ReadBuffer	Read data from buffer memory.	Page 233, Section 5.2.9, Page 309, Section 5.3.9
WriteBuffer	Write data to buffer memory.	Page 237, Section 5.2.10, Page 311, Section 5.3.10
GetClockData	Read clock data from programmable controller CPU.	Page 241, Section 5.2.11, Page 313, Section 5.3.11
SetClockData	Write clock data to programmable controller CPU.	Page 245, Section 5.2.12, Page 315, Section 5.3.12
GetCpuType	Read programmable controller CPU model.	Page 249, Section 5.2.13, Page 317, Section 5.3.13
SetCpuStatus	Remote RUN/STOP/PAUSE of programmable controller CPU.	Page 253, Section 5.2.14, Page 318, Section 5.3.14
EntryDeviceStatus	Register device status monitor.	Page 257, Section 5.2.15, Page 319, Section 5.3.15
FreeDeviceStatus	Deregister device status monitor.	Page 261, Section 5.2.16, Page 322, Section 5.3.16
OnDeviceStatus	Announce event.	Page 262, Section 5.2.17, Page 323, Section 5.3.17
ReadDeviceBlock2	Read devices in bulk. (2-byte data)	Page 264, Section 5.2.18, Page 325, Section 5.3.18
WriteDeviceBlock2	Write devices in bulk. (2-byte data)	Page 267, Section 5.2.19, Page 328, Section 5.3.19
ReadDeviceRandom2	Read devices randomly. (2-byte data)	Page 270, Section 5.2.20, Page 331, Section 5.3.20
WriteDeviceRandom2	Write devices randomly. (2-byte data)	Page 273, Section 5.2.21, Page 336, Section 5.3.21
SetDevice2	Set one point of device. (2-byte data)	Page 276, Section 5.2.22, Page 341, Section 5.3.22
GetDevice2	Acquire data of one point of device. (2-byte data)	Page 279, Section 5.2.23, Page 343, Section 5.3.23
Connect	Connect a telephone line.	Page 281, Section 5.2.24, Page 345, Section 5.3.24
Disconnect	Disconnect a telephone line.	Page 283, Section 5.2.25, Page 347, Section 5.3.25
GetErrorMessage	Display error definition and corrective action.	Page 285, Section 5.2.26, Page 349, Section 5.3.26

Point

- Considerations for using QSCPU
  In order to protect the safety programmable controller system, an error code is returned when a function to write data to buffer memory, write/set devices, or write clock data is executed.

CHAPTER 2 CONTROLS

This chapter explains settings for using controls, programming procedure, device types, and applicable access ranges.

2.1 Settings for Using Controls

This section explains the settings for using controls.

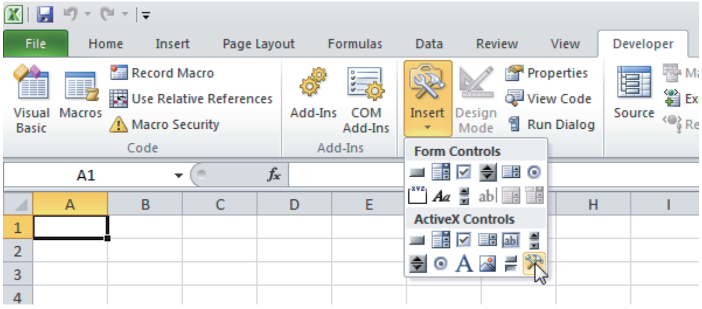
2.1.1 When using VBA

Perform the following setting operation when using VBA.

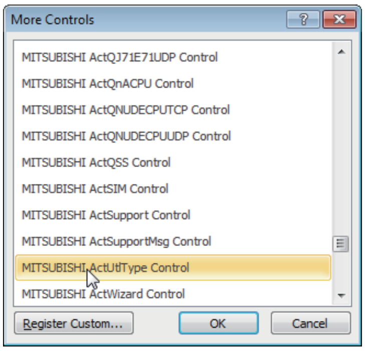
(1) When using Microsoft® Excel 2010

Operating procedure

1. **Start Excel, select [Developer] on the <<Insert>> tab, and click the button.**

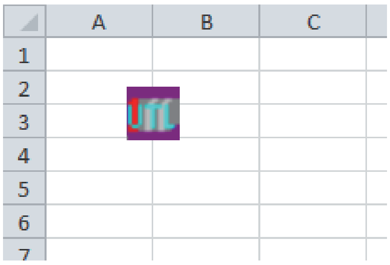


2. **Select the ACT control to be used, and click the OK button.**

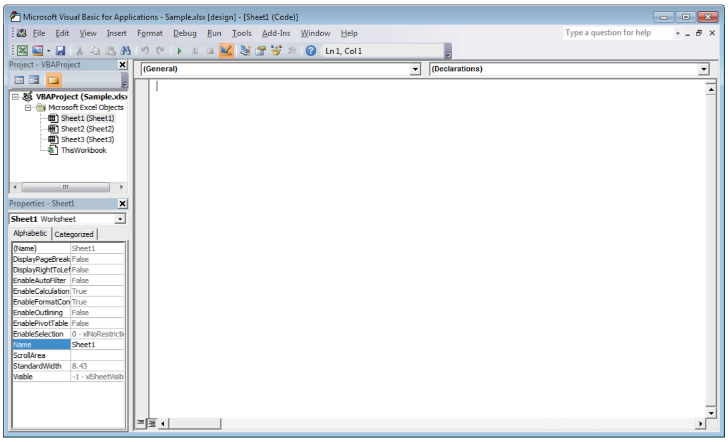

More Controls

MITSUBISHI ActQJ71E71UDP Control
MITSUBISHI ActQnACPU Control
MITSUBISHI ActQNUDECPUTCP Control
MITSUBISHI ActQNUDECPUUDP Control
MITSUBISHI ActQSS Control
MITSUBISHI ActSIM Control
MITSUBISHI ActSupport Control
MITSUBISHI ActSupportMsg Control
MITSUBISHI ActUtlType Control
MITSUBISHI ActWizard Control

3. **Paste the selected ACT control to the sheet.**



4. **Select [Visual Basic] on the <<Developer>> tab to start Visual Basic Editor.**

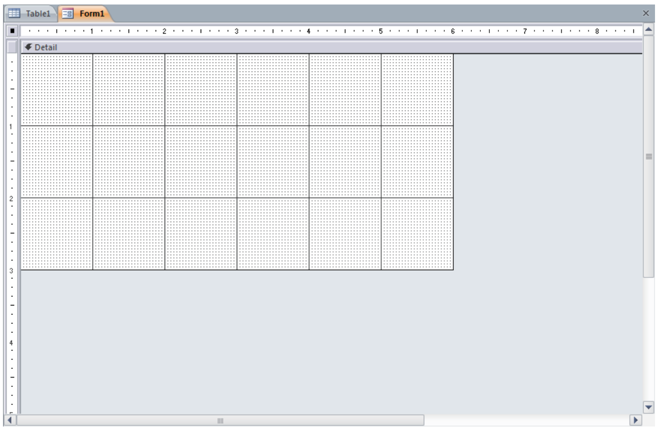


(Name)	Sheet1
DisplayPageBreak	False
DisplayRightToLef	False
EnableAutoFilter	False
EnableCalculation	True
EnableFormatCon	True
EnableOutlining	False
EnablePivotTable	False
EnableSelection	0 - xlNoRestricti
Name	Sheet1
ScrollArea	
StandardWidth	8.43
Visible	-1 - xlSheetVisib

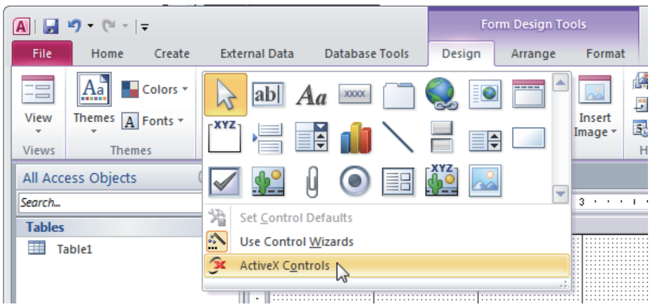
5. **Create a program with Visual Basic Editor.**

(2) When using Microsoft® Access 2010

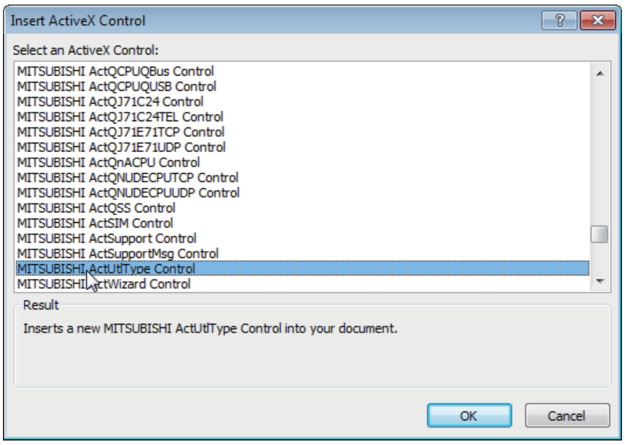
1. **Start Access and activate the database form.**



2. **Select [Controls] ⇒ [ActiveX Controls] on the <<Design>> tab.**



3. **Select the ACT control to be used, and click the OK button.**



Insert ActiveX Control

Select an ActiveX Control:

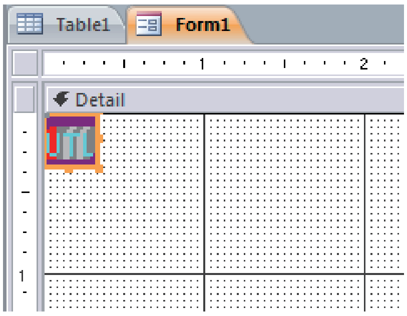
MITSUBISHI ActQCPUQBus Control
MITSUBISHI ActQCPUQUSB Control
MITSUBISHI ActQJ71C24 Control
MITSUBISHI ActQJ71C24TEL Control
MITSUBISHI ActQJ71E71TCP Control
MITSUBISHI ActQJ71E71UDP Control
MITSUBISHI ActQnACPU Control
MITSUBISHI ActQNUDECPUTCP Control
MITSUBISHI ActQNUDECPUUDP Control
MITSUBISHI ActQSS Control
MITSUBISHI ActSIM Control
MITSUBISHI ActSupport Control
MITSUBISHI ActSupportMsg Control
MITSUBISHI ActUtlType Control
MITSUBISHI ActWizard Control

Result

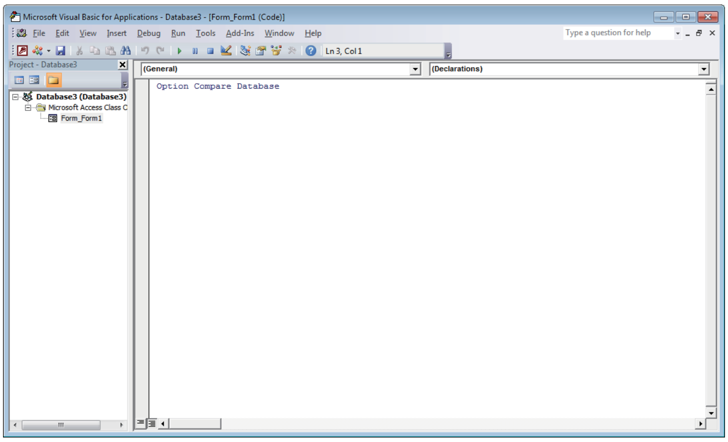
Inserts a new MITSUBISHI ActUtlType Control into your document.

OK Cancel

4. **Paste the selected ACT control to the sheet.**



5. **Select [View Code] on the <<Design>> tab to start Visual Basic Editor.**



```
Option Compare Database
```

6. **Create a program with Visual Basic Editor.**

2.1.2 When using VBScript

Create HTML or ASP using a tool such as notepad, commercially available text editor, or HTML creation tool.
For the grammars of HTML and ASP, refer to the commercially available references.
The HTML and ASP sample programs installed with MX Component can also be referred.

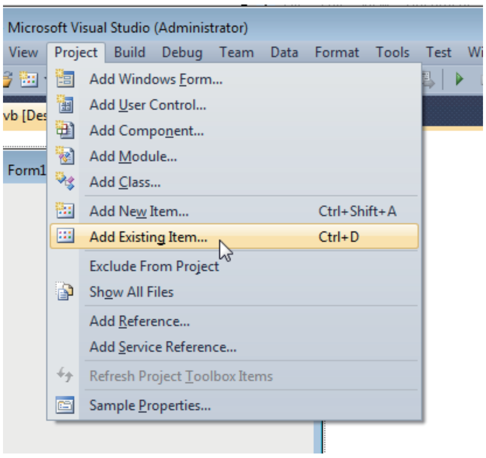
2.1.3 When Using Visual Studio® .NET

Perform the following setting operation when using Visual Studio® .NET.

(1) Setting the include file (for Visual Basic® .NET)

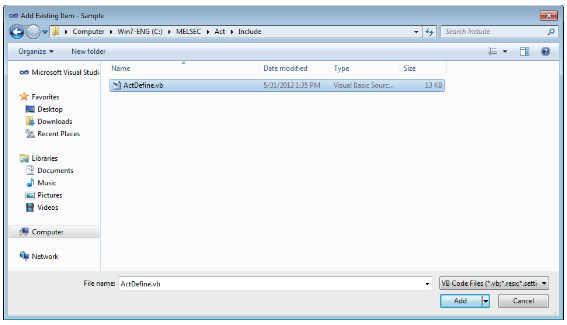
Operating procedure

1. Start the project and select [Project] ⇒ [Add Existing Item].



2. Select the ActDefine.vb file and click the Add button.

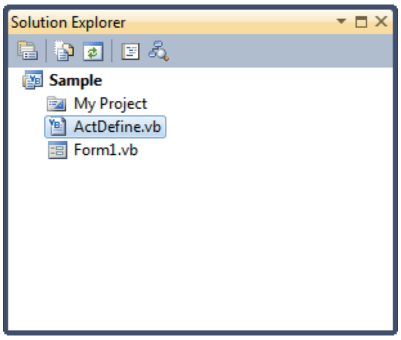
The ActDefine.vb file is stored in the following folder at the time of installation.
<User-specified folder> - <Act> - <Include>



Name	Date modified	Type	Size
ActDefine.vb	5/31/2012 1:35 PM	Visual Basic Sourc...	13 KB

File name: ActDefine.vb

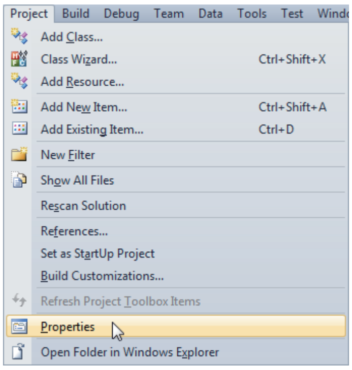
3. The ActDefine.vb file is displayed on the Solution Explorer window.



(2) Setting the include file (for Visual C++® .NET)

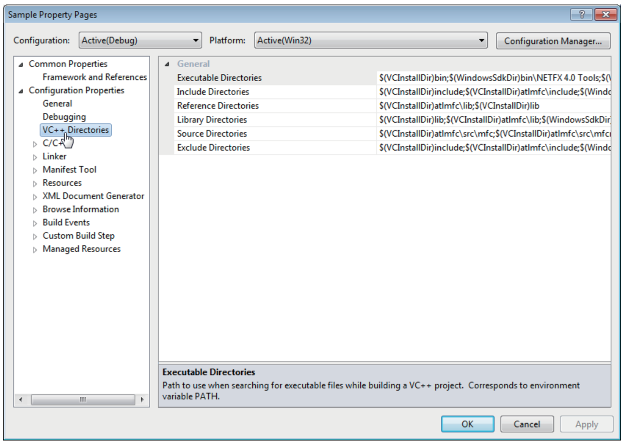
Operating procedure

1. **Start Visual Studio® .NET and select [Project] ⇒ [Property][*1].**



[*1] : For Visual Studio® 2005 and Visual Studio® 2008, select [Tools] ⇒ [Options].

2. **Select [Configuration Properties] ⇒ [VC++ Directories][*2] on the navigation pane displayed on the left side of the screen.**

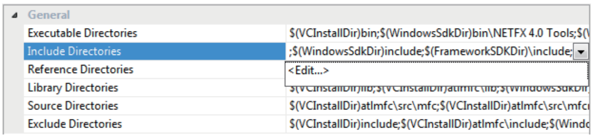


General	
Executable Directories	$(VCInstallDir)bin;$(WindowsSdkDir)bin\NETFX 4.0 Tools;$(\
Include Directories	$(VCInstallDir)include;$(VCInstallDir)atlmfc\include;$(Windc
Reference Directories	$(VCInstallDir)atlmfc\lib;$(VCInstallDir)lib
Library Directories	$(VCInstallDir)lib;$(VCInstallDir)atlmfc\lib;$(WindowsSdkDir)
Source Directories	$(VCInstallDir)atlmfc\src\mfc;$(VCInstallDir)atlmfc\src\mfci
Exclude Directories	$(VCInstallDir)include;$(VCInstallDir)atlmfc\include;$(Windc

Executable Directories
Path to use when searching for executable files while building a VC++ project. Corresponds to environment variable PATH.

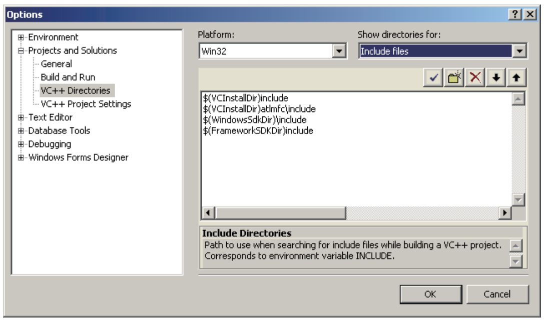
[*2] : For Visual Studio® 2005 and Visual Studio® 2008, select [Projects and Solutions] ⇒ [VC++ Directories].

3. **Right-click on "Include Directories" displayed on the right side of the screen and select <Edit...>.[*3]**



General	
Executable Directories	$(VCInstallDir)bin;$(WindowsSdkDir)bin\NETFX 4.0 Tools;$(\
Include Directories	;$(WindowsSdkDir)include;$(FrameworkSDKDir)\include;
Reference Directories	<Edit...>
Library Directories	$(VCInstallDir)lib;$(VCInstallDir)atlmfc\lib;$(WindowsSdkDir)
Source Directories	$(VCInstallDir)atlmfc\src\mfc;$(VCInstallDir)atlmfc\src\mfci
Exclude Directories	$(VCInstallDir)include;$(VCInstallDir)atlmfc\include;$(Windc

[*3] : For Visual Studio® 2005 and Visual Studio® 2008, select "Include files" for "Shows directories for:" on the top right of the screen.



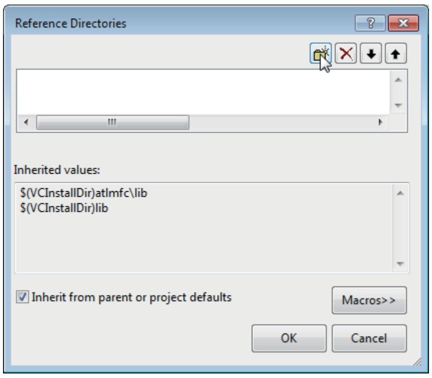
Platform: Win32

Show directories for: Include files

```
$(VCInstallDir)include
$(VCInstallDir)atlmfc\include
$(WindowsSdkDir)\include
$(FrameworkSDKDir)include
```

Include Directories
Path to use when searching for include files while building a VC++ project. Corresponds to environment variable INCLUDE.

4. Click (New line).



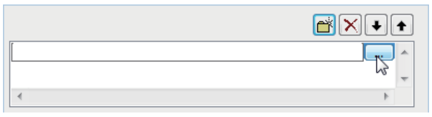
Reference Directories

Inherited values:

$(VCInstallDir)atlmfc\lib
$(VCInstallDir)lib

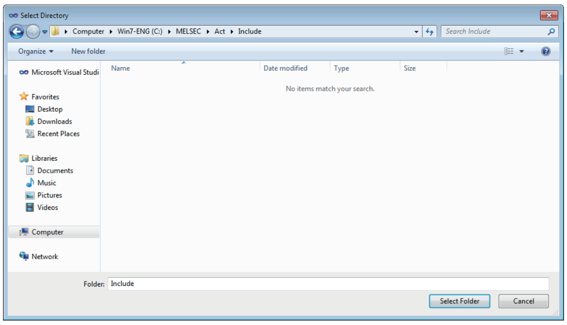
☑ Inherit from parent or project defaults

5. Click ….



6. Select the folder that contains the Include files.

The include files are stored in the following folder at the time of installation.
<User-specified folder> - <Act> - <Include>



Select Directory

Computer ▸ Win7-ENG (C:) ▸ MELSEC ▸ Act ▸ Include

No items match your search.

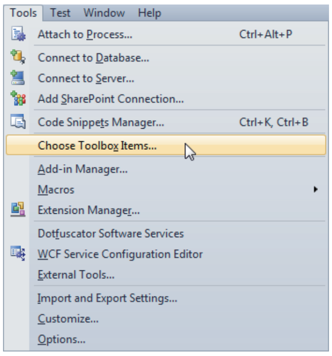
Folder: Include

(3) When pasting a control to the form

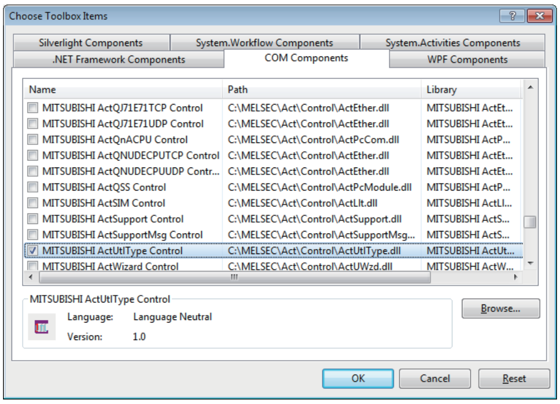
(Common setting for Visual C++® .NET and Visual Basic® .NET projects)

Operating procedure

1. **Select [Tools] ⇒ [Choose Toolbox Items].**


Tools
Test
Window
Help
Attach to Process... Ctrl+Alt+P
Connect to Database...
Connect to Server...
Add SharePoint Connection...
Code Snippets Manager... Ctrl+K, Ctrl+B
Choose Toolbox Items...
Add-in Manager...
Macros
Extension Manager...
Dotfuscator Software Services
WCF Service Configuration Editor
External Tools...
Import and Export Settings...
Customize...
Options...

2. **Select the <<.NET Framework Components>> tab (for pasting .NET control) or the <<COM Components>> tab (for pasting ACT control), select the control to be used, and click the OK button.**


Choose Toolbox Items
Silverlight Components
System.Workflow Components
System.Activities Components
.NET Framework Components
COM Components
WPF Components

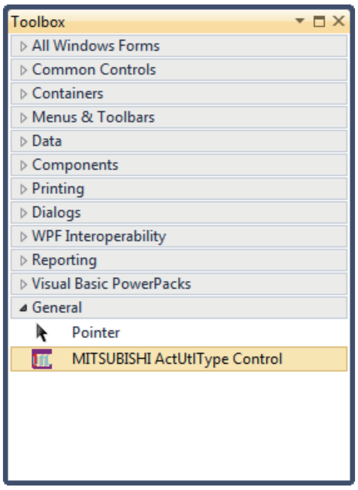
Name	Path	Library
□ MITSUBISHI ActQJ71E71TCP Control	C:\MELSEC\Act\Control\ActEther.dll	MITSUBISHI ActEt...
□ MITSUBISHI ActQJ71E71UDP Control	C:\MELSEC\Act\Control\ActEther.dll	MITSUBISHI ActEt...
□ MITSUBISHI ActQnACPU Control	C:\MELSEC\Act\Control\ActPcCom.dll	MITSUBISHI ActP...
□ MITSUBISHI ActQNUDECPUTCP Control	C:\MELSEC\Act\Control\ActEther.dll	MITSUBISHI ActEt...
□ MITSUBISHI ActQNUDECPUUDP Contr...	C:\MELSEC\Act\Control\ActEther.dll	MITSUBISHI ActEt...
□ MITSUBISHI ActQSS Control	C:\MELSEC\Act\Control\ActPcModule.dll	MITSUBISHI ActP...
□ MITSUBISHI ActSIM Control	C:\MELSEC\Act\Control\ActLlt.dll	MITSUBISHI ActLl...
□ MITSUBISHI ActSupport Control	C:\MELSEC\Act\Control\ActSupport.dll	MITSUBISHI ActS...
□ MITSUBISHI ActSupportMsg Control	C:\MELSEC\Act\Control\ActSupportMsg...	MITSUBISHI ActS...
☑ MITSUBISHI ActUtlType Control	C:\MELSEC\Act\Control\ActUtlType.dll	MITSUBISHI ActUt...
□ MITSUBISHI ActWizard Control	C:\MELSEC\Act\Control\ActUWzd.dll	MITSUBISHI ActW...

MITSUBISHI ActUtlType Control

Language: Language Neutral

Version: 1.0

3. **The control is added to the bottom of the tab selected under "Toolbox".**



Point

- When creating an application using .NET Framework 4
  Add an application configuration file with the following elements (app.config) to the folder which contains the .exe file of the application.

```
<configuration>
  <startup useLegacyV2RuntimeActivationPolicy="true">
    <supportedRuntime version="v4.0"/>
  </startup>
</configuration>
```

For details, refer to Visual Studio® Help.

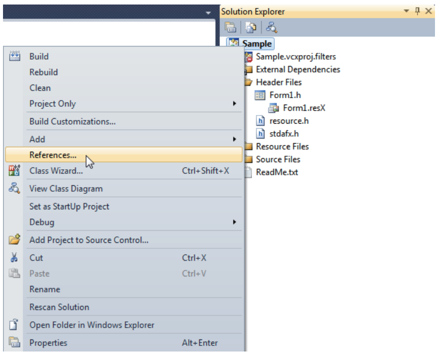
(4) When using control without pasting it to a form (Reference setting)

(Common setting for Visual C++® .NET and Visual Basic® .NET projects)

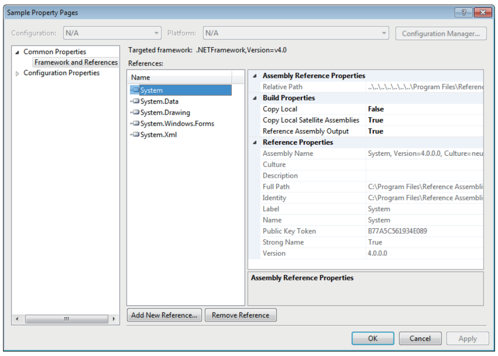
Operating procedure

1. Select [View] ⇒ [Solution Explorer] to display "Solution Explorer".

2. Right-click the project and select "Reference".



3. Select the Add New Reference... button on "Property Pages".



Targeted framework: .NETFramework,Version=v4.0

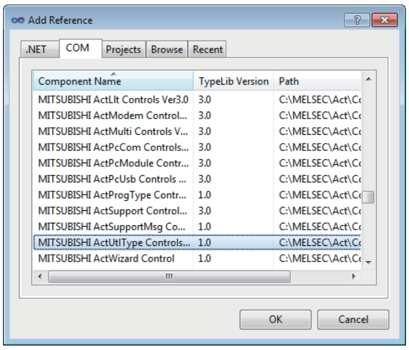
References:

Name
- System
- System.Data
- System.Drawing
- System.Windows.Forms
- System.Xml

Assembly Reference Properties	
Relative Path	..\..\..\..\..\..\..\Program Files\Referen
Build Properties	
Copy Local	False
Copy Local Satellite Assemblies	True
Reference Assembly Output	True
Reference Properties	
Assembly Name	System, Version=4.0.0.0, Culture=neu
Culture	
Description	
Full Path	C:\Program Files\Reference Assembli
Identity	C:\Program Files\Reference Assembli
Label	System
Name	System
Public Key Token	B77A5C561934E089
Strong Name	True
Version	4.0.0.0

Assembly Reference Properties

4. **Select the <<.NET>> tab (for pasting .NET control) or the <<COM>> tab (for pasting ACT control), select the control to be used, and click the** OK **button.**



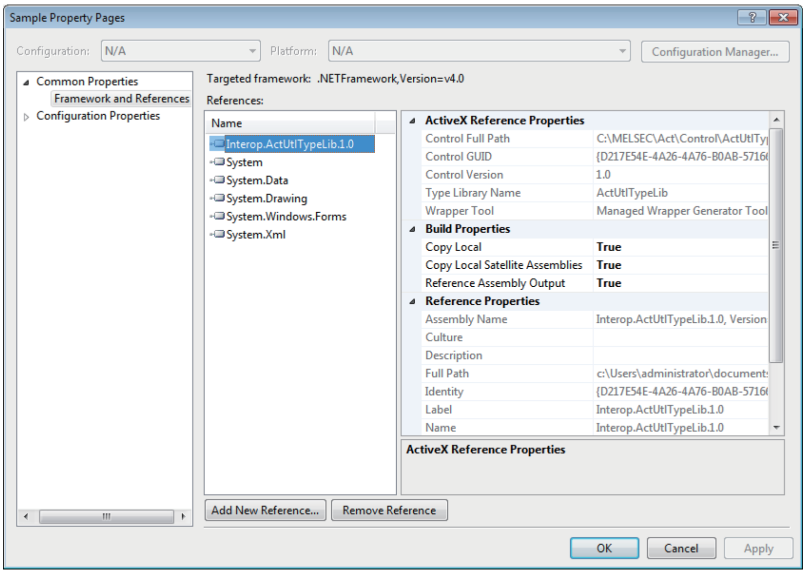
Add Reference

.NET | COM | Projects | Browse | Recent

Component Name	TypeLib Version	Path
MITSUBISHI ActLlt Controls Ver3.0	3.0	C:\MELSEC\Act\C
MITSUBISHI ActModem Control...	3.0	C:\MELSEC\Act\C
MITSUBISHI ActMulti Controls V...	3.0	C:\MELSEC\Act\C
MITSUBISHI ActPcCom Controls...	3.0	C:\MELSEC\Act\C
MITSUBISHI ActPcModule Contr...	3.0	C:\MELSEC\Act\C
MITSUBISHI ActPcUsb Controls ...	3.0	C:\MELSEC\Act\C
MITSUBISHI ActProgType Contr...	1.0	C:\MELSEC\Act\C
MITSUBISHI ActSupport Control...	3.0	C:\MELSEC\Act\C
MITSUBISHI ActSupportMsg Co...	1.0	C:\MELSEC\Act\C
MITSUBISHI ActUtlType Controls...	1.0	C:\MELSEC\Act\C
MITSUBISHI ActWizard Control	1.0	C:\MELSEC\Act\C

OK | Cancel

5. **The library of the component to be used is set to "References" as a reference.**



Sample Property Pages

Configuration: N/A Platform: N/A Configuration Manager...

- Common Properties
  - Framework and References
- Configuration Properties

Targeted framework: .NETFramework,Version=v4.0

References:

Name
- Interop.ActUtlTypeLib.1.0
- System
- System.Data
- System.Drawing
- System.Windows.Forms
- System.Xml

ActiveX Reference Properties	
Control Full Path	C:\MELSEC\Act\Control\ActUtlTy
Control GUID	{D217E54E-4A26-4A76-B0AB-5716
Control Version	1.0
Type Library Name	ActUtlTypeLib
Wrapper Tool	Managed Wrapper Generator Tool
Build Properties	
Copy Local	**True**
Copy Local Satellite Assemblies	**True**
Reference Assembly Output	**True**
Reference Properties	
Assembly Name	Interop.ActUtlTypeLib.1.0, Version
Culture	
Description	
Full Path	c:\Users\administrator\document
Identity	{D217E54E-4A26-4A76-B0AB-5716
Label	Interop.ActUtlTypeLib.1.0
Name	Interop.ActUtlTypeLib.1.0

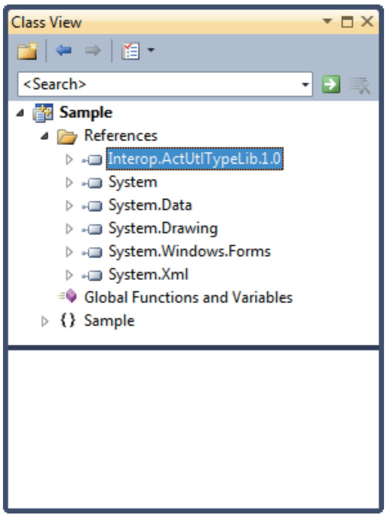
ActiveX Reference Properties

Add New Reference... | Remove Reference

OK | Cancel | Apply

The library of the component to be used can be checked with "References" on "Class View".

(For Visual Studio® 2008 and Visual Studio® 2010, enable "Show Project References" in "Class View Settings".)



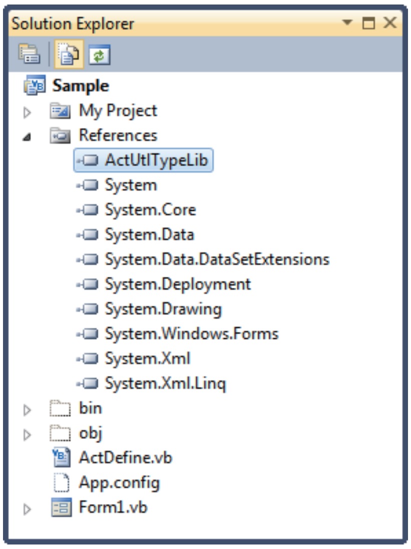
Class View

<Search>

- Sample
  - References
    - Interop.ActUtlTypeLib.1.0
    - System
    - System.Data
    - System.Drawing
    - System.Windows.Forms
    - System.Xml
  - Global Functions and Variables
  - { } Sample

6. **For Visual Basic$^{®}$ .NET of Visual Studio$^{®}$ 2010, click on "Solution Explorer" to display all files.**

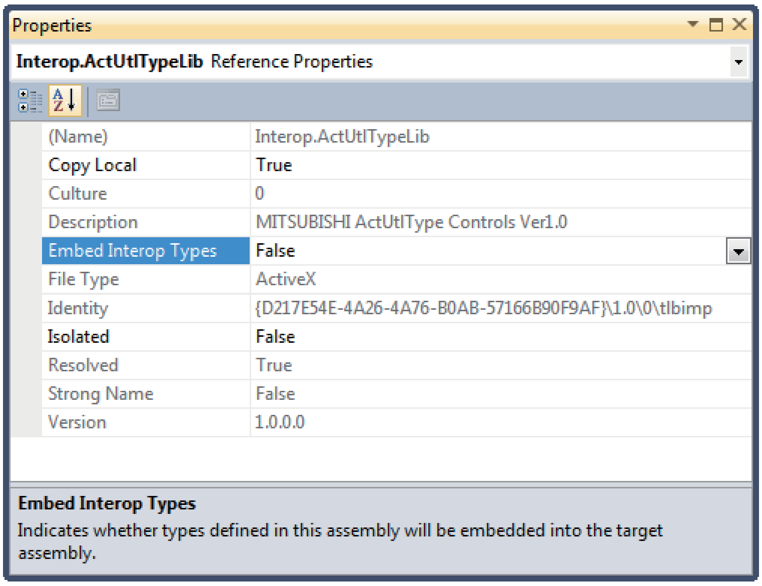
Right-click the library of the component to be used under "References" and select "Properties".



Solution Explorer

- Sample
  - My Project
  - References
    - ActUtlTypeLib
    - System
    - System.Core
    - System.Data
    - System.Data.DataSetExtensions
    - System.Deployment
    - System.Drawing
    - System.Windows.Forms
    - System.Xml
    - System.Xml.Linq
  - bin
  - obj
  - ActDefine.vb
  - App.config
  - Form1.vb

Set "False" for "Embed Interop Types".



Properties

Interop.ActUtlTypeLib Reference Properties

(Name)	Interop.ActUtlTypeLib
Copy Local	True
Culture	0
Description	MITSUBISHI ActUtlType Controls Ver1.0
Embed Interop Types	False
File Type	ActiveX
Identity	{D217E54E-4A26-4A76-B0AB-57166B90F9AF}\1.0\0\tlbimp
Isolated	False
Resolved	True
Strong Name	False
Version	1.0.0.0

Embed Interop Types

Indicates whether types defined in this assembly will be embedded into the target assembly.

Point

- ● When using .NET Control
  The following library is added to "References".
  - For DotUtlType: ActTypeLib
  - For DotSupportMsg: ActSupportMsgLib
- ● When creating an application using .NET Framework 4
  Add an application configuration file with the following elements (app.config) to the folder which contains the .exe file of the application.

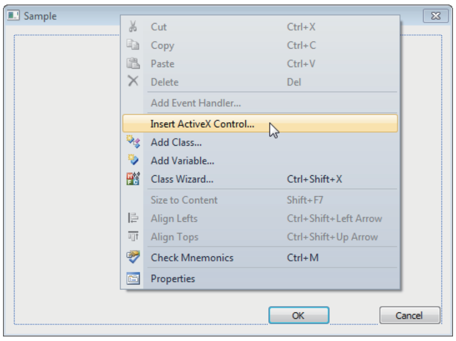
```
<configuration>
  <startup useLegacyV2RuntimeActivationPolicy="true">
    <supportedRuntime version="v4.0"/>
  </startup>
</configuration>
```

For details, refer to Visual Studio$^{®}$ Help.

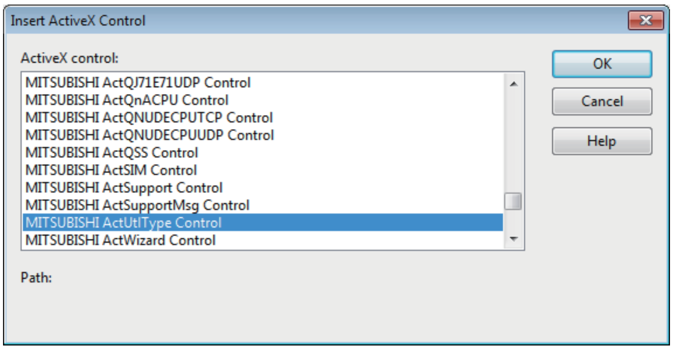
(5) When using ACT control on Visual C++® .NET(MFC)

Operating procedure

1. Right-click the form and select "Insert ActiveX Control".



2. Select the ACT control to be used and click the OK button.

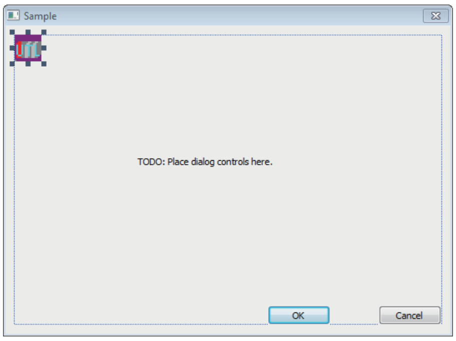


ActiveX control:

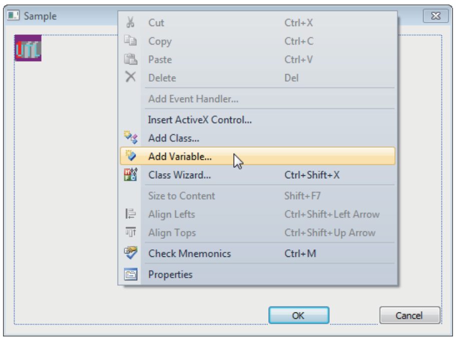
- MITSUBISHI ActQJ71E71UDP Control
- MITSUBISHI ActQnACPU Control
- MITSUBISHI ActQNUDECPUTCP Control
- MITSUBISHI ActQNUDECPUUDP Control
- MITSUBISHI ActQSS Control
- MITSUBISHI ActSIM Control
- MITSUBISHI ActSupport Control
- MITSUBISHI ActSupportMsg Control
- MITSUBISHI ActUtlType Control
- MITSUBISHI ActWizard Control

Path:

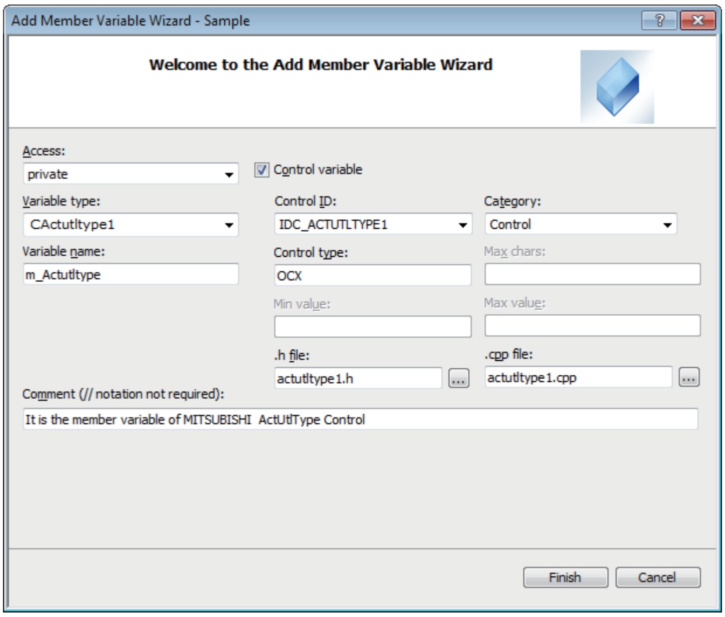
3. The selected ACT control is pasted to the form.



4. Right-click the form and select "Add Variable".



5. **After selecting "Control variable", select the ID of the ACT control for "Control ID", enter the variable name, and click the Finish button.**


Add Member Variable Wizard - Sample
Welcome to the Add Member Variable Wizard
Finish
Cancel

Access: private

☑ Control variable

Variable type: CActutltype1

Control ID: IDC_ACTUTLTYPE1

Category: Control

Variable name: m_Actutltype

Control type: OCX

Max chars:

Min value:

Max value:

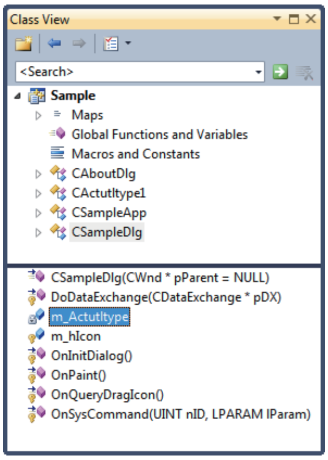
.h file: actutltype1.h

.cpp file: actutltype1.cpp

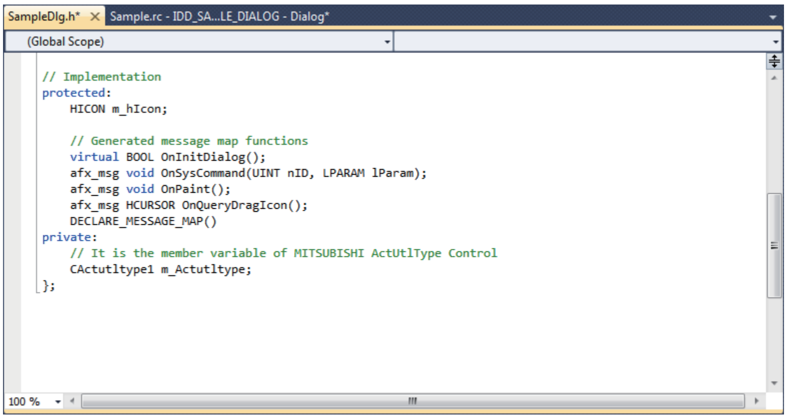
Comment (// notation not required): It is the member variable of MITSUBISHI ActUtlType Control

6. **Check that the member variable set in Step 5 is created.**

- In the form class of Class View



- In the header file of form class



```
// Implementation
protected:
    HICON m_hIcon;

    // Generated message map functions
    virtual BOOL OnInitDialog();
    afx_msg void OnSysCommand(UINT nID, LPARAM lParam);
    afx_msg void OnPaint();
    afx_msg HCURSOR OnQueryDragIcon();
    DECLARE_MESSAGE_MAP()
private:
    // It is the member variable of MITSUBISHI ActUtlType Control
    CActutltype1 m_Actutltype;
};
```

2.2 Programming Procedure

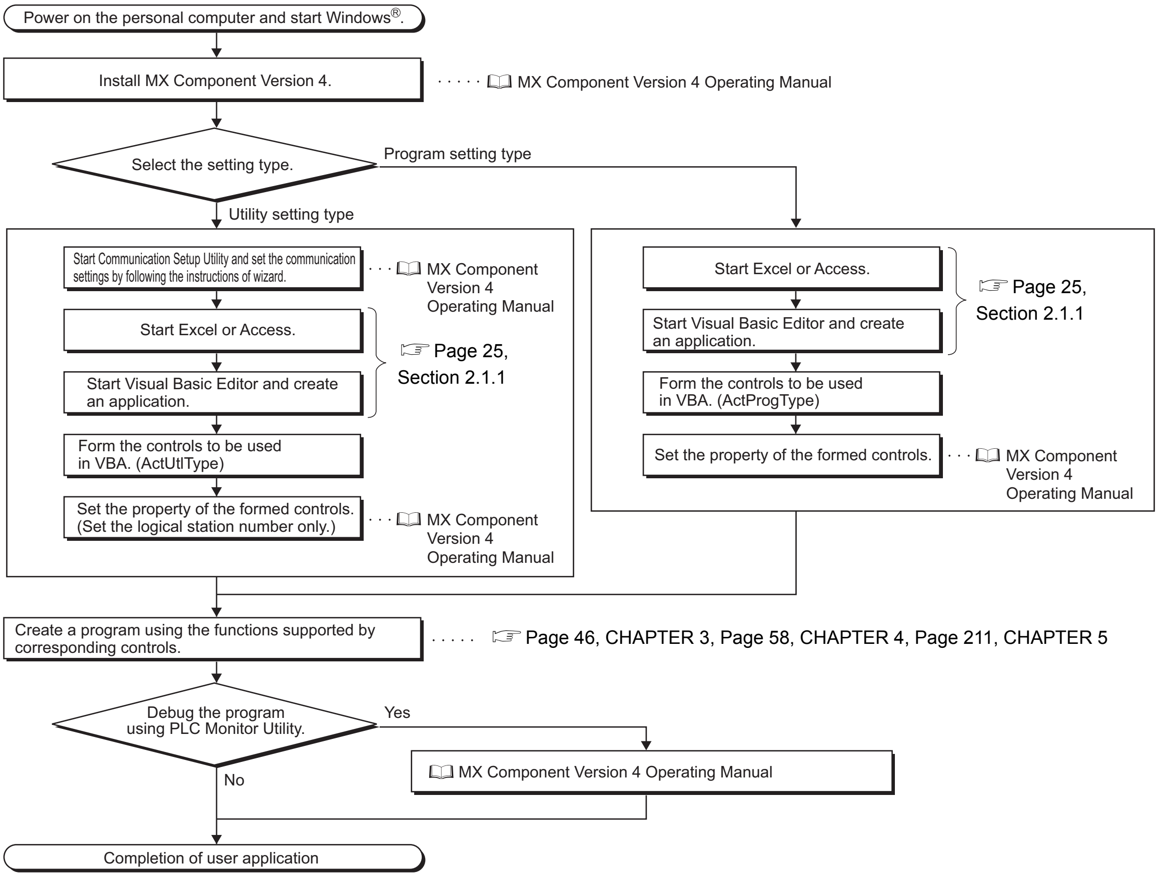
This section explains the procedure to create a user application.

Point

When creating a user program, select "x86" (32 bits) for "Target CPU".

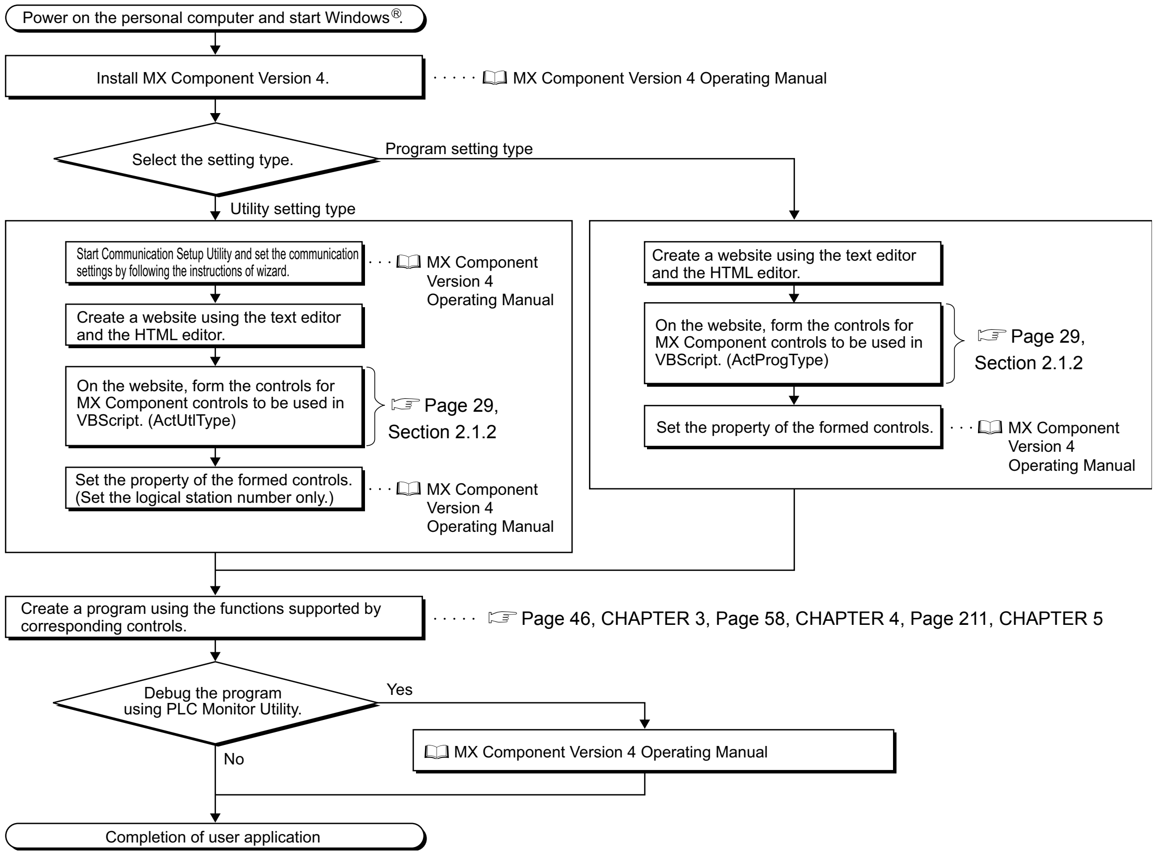
2.2.1 When using VBA

Create a user application with the following procedure when using VBA.


Power on the personal computer and start Windows®.
Install MX Component Version 4.
····· 🕮 MX Component Version 4 Operating Manual
Select the setting type.
Program setting type
Utility setting type
Start Communication Setup Utility and set the communication settings by following the instructions of wizard.
··· 🕮 MX Component Version 4 Operating Manual
Start Excel or Access.
Start Visual Basic Editor and create an application.
☞ Page 25, Section 2.1.1
Form the controls to be used in VBA. (ActUtlType)
Set the property of the formed controls. (Set the logical station number only.)
··· 🕮 MX Component Version 4 Operating Manual
Start Excel or Access.
Start Visual Basic Editor and create an application.
☞ Page 25, Section 2.1.1
Form the controls to be used in VBA. (ActProgType)
Set the property of the formed controls.
··· 🕮 MX Component Version 4 Operating Manual
Create a program using the functions supported by corresponding controls.
····· ☞ Page 46, CHAPTER 3, Page 58, CHAPTER 4, Page 211, CHAPTER 5
Debug the program using PLC Monitor Utility.
Yes
No
🕮 MX Component Version 4 Operating Manual
Completion of user application

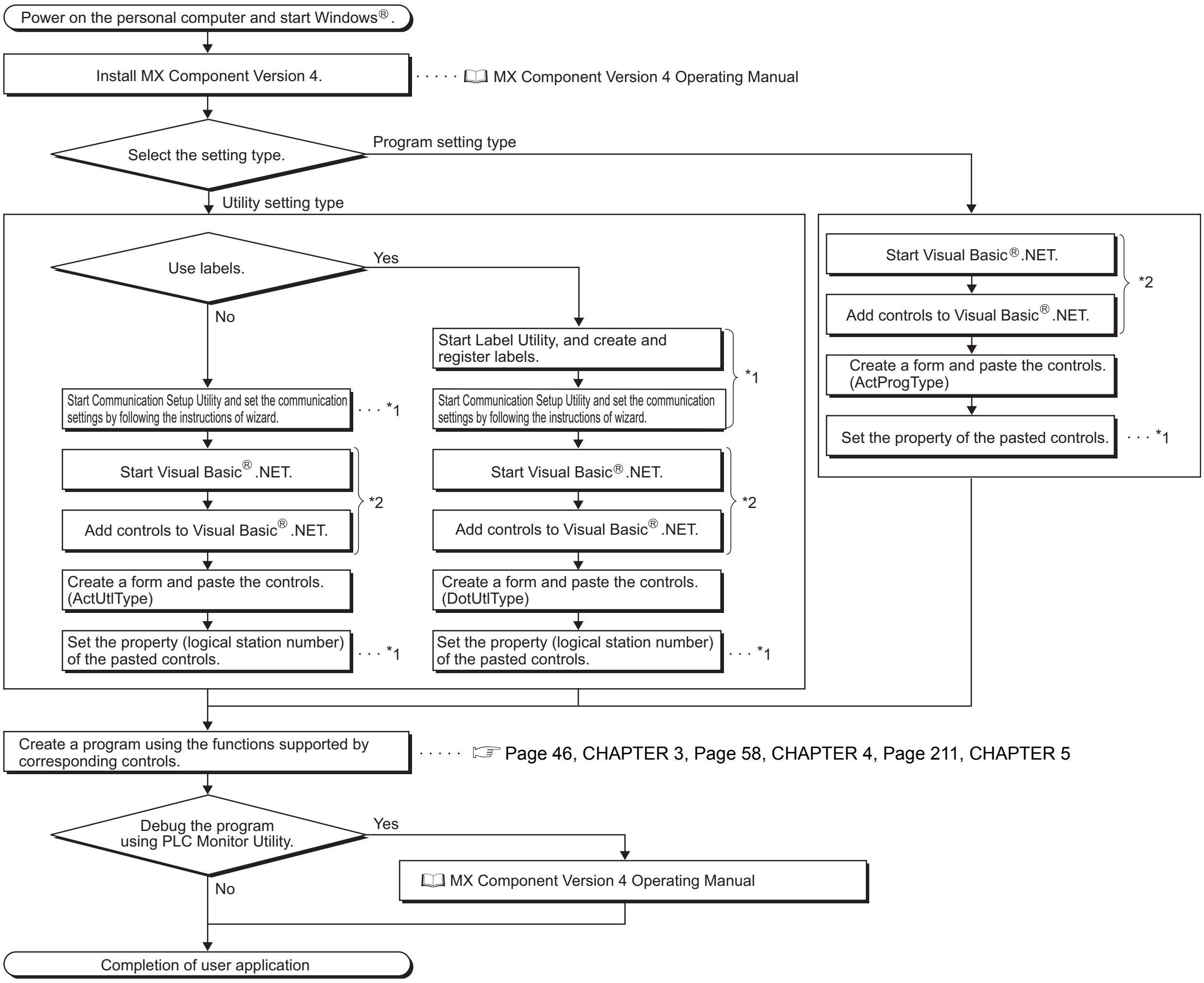
2.2.2 When using VBScript

Create a user application with the following procedure when using VBScript.


Power on the personal computer and start Windows®.
Install MX Component Version 4.
····· 📖 MX Component Version 4 Operating Manual
Select the setting type.
Program setting type
Utility setting type
Start Communication Setup Utility and set the communication settings by following the instructions of wizard.
··· 📖 MX Component Version 4 Operating Manual
Create a website using the text editor and the HTML editor.
On the website, form the controls for MX Component controls to be used in VBScript. (ActUtlType)
☞ Page 29, Section 2.1.2
Set the property of the formed controls. (Set the logical station number only.)
··· 📖 MX Component Version 4 Operating Manual
Create a website using the text editor and the HTML editor.
On the website, form the controls for MX Component controls to be used in VBScript. (ActProgType)
☞ Page 29, Section 2.1.2
Set the property of the formed controls.
··· 📖 MX Component Version 4 Operating Manual
Create a program using the functions supported by corresponding controls.
····· ☞ Page 46, CHAPTER 3, Page 58, CHAPTER 4, Page 211, CHAPTER 5
Debug the program using PLC Monitor Utility.
Yes
No
📖 MX Component Version 4 Operating Manual
Completion of user application

2.2.3 When using Visual Basic® .NET

Create a user application with the following procedure when using Visual Basic® .NET.

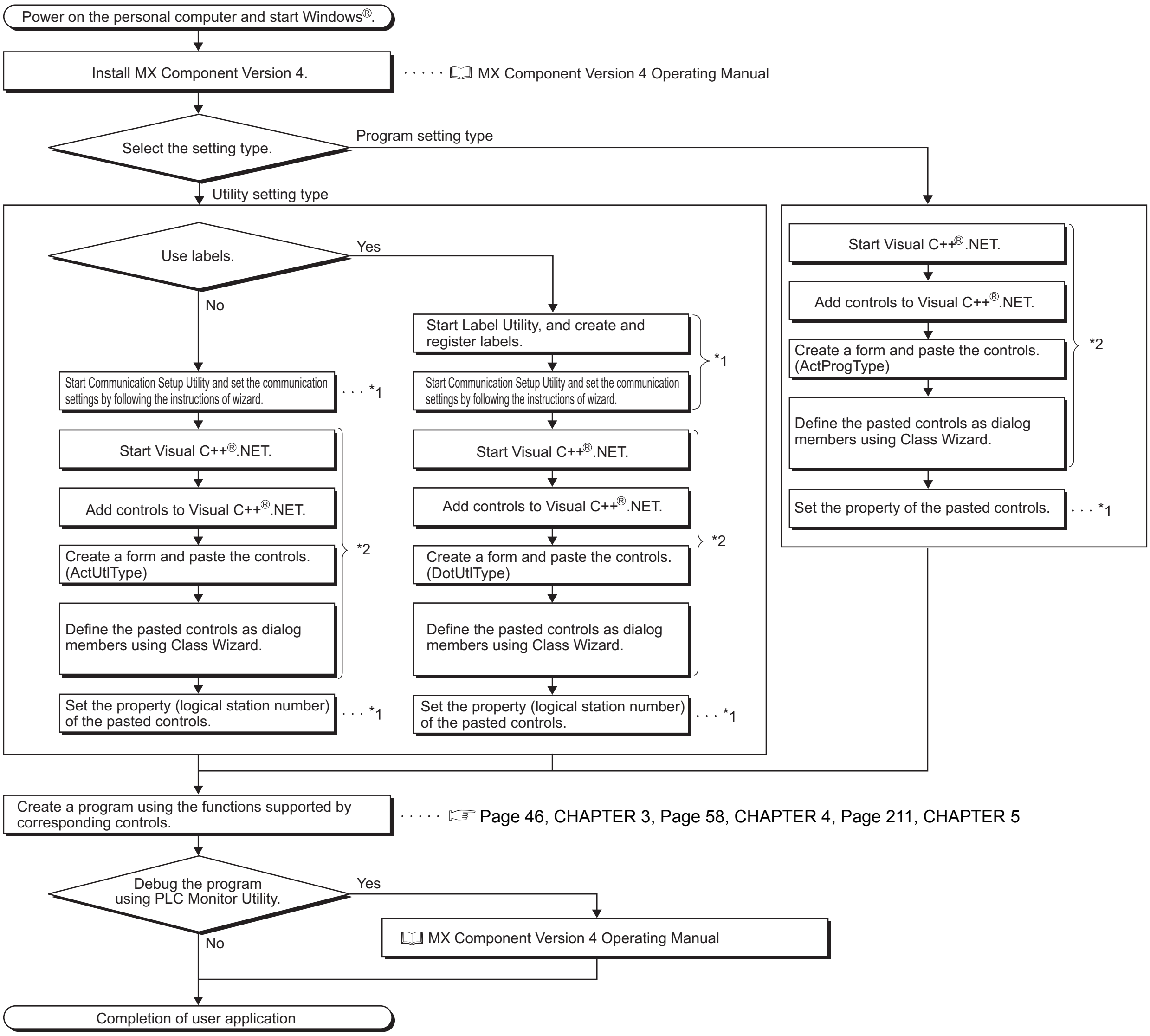

Power on the personal computer and start Windows®.
Install MX Component Version 4.
· · · · · MX Component Version 4 Operating Manual
Select the setting type.
Program setting type
Utility setting type
Use labels.
Yes
No
Start Label Utility, and create and register labels.
Start Communication Setup Utility and set the communication settings by following the instructions of wizard.
· · · *1
Start Visual Basic® .NET.
Add controls to Visual Basic® .NET.
*2
Create a form and paste the controls. (ActUtlType)
Create a form and paste the controls. (DotUtlType)
Set the property (logical station number) of the pasted controls.
Create a form and paste the controls. (ActProgType)
Set the property of the pasted controls.
Create a program using the functions supported by corresponding controls.
· · · · · Page 46, CHAPTER 3, Page 58, CHAPTER 4, Page 211, CHAPTER 5
Debug the program using PLC Monitor Utility.
MX Component Version 4 Operating Manual
Completion of user application

*1 : MX Component Version 4 Operating Manual

*2 : Page 30, Section 2.1.3

2.2.4 When using Visual C++® .NET

Create a user application with the following procedure when using Visual C++® .NET.



Power on the personal computer and start Windows®.
Install MX Component Version 4.
· · · · · MX Component Version 4 Operating Manual
Select the setting type.
Program setting type
Utility setting type
Use labels.
Yes
No
Start Label Utility, and create and register labels.
*1
Start Communication Setup Utility and set the communication settings by following the instructions of wizard.
· · · *1
Start Communication Setup Utility and set the communication settings by following the instructions of wizard.
Start Visual C++®.NET.
Add controls to Visual C++®.NET.
Create a form and paste the controls. (ActUtlType)
Define the pasted controls as dialog members using Class Wizard.
*2
Set the property (logical station number) of the pasted controls.
· · · *1
Start Visual C++®.NET.
Add controls to Visual C++®.NET.
Create a form and paste the controls. (DotUtlType)
Define the pasted controls as dialog members using Class Wizard.
*2
Set the property (logical station number) of the pasted controls.
· · · *1
Start Visual C++®.NET.
Add controls to Visual C++®.NET.
Create a form and paste the controls. (ActProgType)
Define the pasted controls as dialog members using Class Wizard.
*2
Set the property of the pasted controls.
· · · *1
Create a program using the functions supported by corresponding controls.
· · · · · ☞ Page 46, CHAPTER 3, Page 58, CHAPTER 4, Page 211, CHAPTER 5
Debug the program using PLC Monitor Utility.
Yes
No
MX Component Version 4 Operating Manual
Completion of user application

*1 : MX Component Version 4 Operating Manual

*2 : ☞ Page 30, Section 2.1.3

2.3 Device Types

This section explains the devices that can be specified for functions.

Point

- Specify devices with "device name + device number" for any of the following functions.
  For the device numbers, note the differences between octal, decimal, and hexadecimal numbers.
  Target functions: ReadDeviceBlock, ReadDeviceBlock2, WriteDeviceBlock, WriteDeviceBlock2, ReadDeviceRandom, ReadDeviceRandom2, WriteDeviceRandom, WriteDeviceRandom2, SetDevice, SetDevice2, GetDevice, GetDevice2
- When specifying bit devices for ReadDeviceBlock, ReadDeviceBlock2, WriteDeviceBlock, or WriteDeviceBlock2, specify the device number with a multiple of 16.
- Local devices and file registers per program of Q series programmable controller CPU cannot be accessed by specifying a program name.
- Only the devices indicated in this section are supported. Do not use unsupported devices.

(1) Common device types (except for gateway function communication)

The following device types are common to all communication paths except for gateway function communication.

Device		Device name	Device type	Device number
Function input		FX	Bit device	Decimal
Function output		FY	Bit device	Decimal
Function register		FD	Word device*[1]	Decimal
Special relay		SM	Bit device	Decimal
Special register		SD	Word device	Decimal
Input relay		X	Bit device	Hexadecimal*[2]
Output relay		Y	Bit device	Hexadecimal*[2]
Internal relay		M	Bit device	Decimal
Latch relay		L	Bit device	Decimal
Annunciator		F	Bit device	Decimal
Edge relay		V	Bit device	Decimal
Link relay		B	Bit device	Hexadecimal
Data register		D	Word device	Decimal
Link register		W	Word device	Hexadecimal
Timer	Contact	TS	Bit device	Decimal
	Coil	TC	Bit device	Decimal
	Present value	TN	Word device	Decimal
Counter	Contact	CS	Bit device	Decimal
	Coil	CC	Bit device	Decimal
	Present value*[3]	CN	Word device	Decimal
Retentive timer	Contact	SS	Bit device	Decimal
	Coil	SC	Bit device	Decimal
	Present value	SN	Word device	Decimal

*1 : 4 words/1 point. For a bulk operation, the operation is performed continuously in units of one word. For a random operation, only the first one word is read.

*2 : For FXCPU, the device number is octal.

*3 : For FXCPU, the value higher than or equal to 200 is 32-bit data.

Device		Device name	Device type	Device number
Link special relay		SB	Bit device	Hexadecimal
Link special register		SW	Word device	Hexadecimal
Step relay		S	Bit device	Decimal
Accumulator		A *4	Word device	Decimal
Index register		Z *4	Word device	Decimal
		V *4	Word device	Decimal
File register		R *5	Word device	Decimal
		ZR	Word device	Decimal
Extended file register		ER*\R *6	Word device	Decimal
Direct link *7	Link input	J*\X *6	Bit device	Hexadecimal
	Link output	J*\Y *6	Bit device	Hexadecimal
	Link relay	J*\B *6	Bit device	Hexadecimal
	Link special relay	J*\SB *6	Bit device	Hexadecimal
	Link register	J*\W *6	Word device	Hexadecimal
	Link special register	J*\SW *6	Word device	Hexadecimal
Special direct buffer memory *8,*9		U*\G** *6,*10	Word device	Hexadecimal/Decimal

*4 : These devices cannot be used when E71 is relayed.

*5 : For specifying an extended file register, enter "\" between the block number and the file register.
When specifying R**, R of the block No. 0 becomes a target.
When specifying ER0\R**, an error is returned.
When specifying ER**\R**, the extension representation (indirect specification, digit specification) is not applicable.

*6 : For a direct specification, enter "\" between the direct specification and the device specification.

7 : J specifies a network number.

8 : U specifies a special module I/O number (hexadecimal), and G** specifies a buffer memory address (decimal).
(Example: When the special module I/O number is 200H and the buffer memory address is 100, the device name will be "U20\G100".)

*9 : In a QCPU multiple CPU configuration, an error occurs if the shared memory of the host QCPU is specified.
Regardless of whether the CPU is a host CPU or other CPU, an error occurs if data is written to the shared memory.

*10 : Not supported by FX$_0$, FX$_{0S}$, FX$_{0N}$, FX$_1$, FX$_{1N}$, FX$_{1NC}$, FX$_{1S}$, FX$_U$, FX$_{2C}$, FX$_{2N}$, FX$_{2NC}$, and FX$_{3G}$.

(2) For Q motion CPU as access target only

The devices described in the following table can be used only when the access target is a Q motion CPU.

Device	Device name	Device type	Device number
Motion registers	#	Word device	Decimal

(3) For CC-Link communication only

For CC-Link communication only, the devices in the following table can be used when the own board is accessed. These devices cannot be used for other communication paths.

Device	Device name	Device type	Device number	Remarks
Special relay	SM	Bit device	Decimal	Special relay of own board
Special register	SD	Word device	Decimal	Special register of own board
Link special relay (for CC-Link)	SB	Bit device	Hexadecimal	Link special relay of own board
Link special register (for CC-Link)	SW	Word device	Hexadecimal	Link special register of own board
Remote input	X	Bit device	Hexadecimal	RX
Remote output	Y	Bit device	Hexadecimal	RY
Link register	W	Word device	Hexadecimal	–
Remote register (Data write area for CC-Link)	WW	Word device	Hexadecimal	RWw
Remote register (Data read area for CC-Link)	WR	Word device	Hexadecimal	RWr
Buffer memory	ML	Word device	Hexadecimal	Buffer memory of own station CC-Link module
Random access buffer	MC	Word device	Hexadecimal	Random access buffer in buffer memory of own station CC-Link module
Automatic refresh buffer	MF	Bit device	Hexadecimal	Automatic refresh buffer of own station CC-Link module

(4) For gateway function communication only

This section explains how to specify the device name used for gateway function communication.
For specifying devices used for other communication, refer to the following manual.
📖MX Component Version 4 Operating Manual

Device	Device name	Device type	Device number
Gateway device*1	EG	Word device	Decimal

*1 : If a gateway device to which a programmable controller CPU device is not assigned is read, the read data becomes 0.

(5) Device extension representations

The following table shows applicability of device extension representation.
These representations cannot be used for ReadDeviceBlock and WriteDeviceBlock.

Device extension representation	Target CPU						GOT
	QCPU (Q mode)	Q12DC CPU-V	LCPU	QSCPU	FXCPU	Q motion CPU	
Digit specification (Example: K4M0)*1	○	○	○	○	○	×	×
Bit specification (Example: D0.1)	○*3	○	○*3	○*3	○*3	×	○
Index setting (Example: M100Z0)*2	○	×	○	×	×	×	×

○: Applicable ×: Not applicable

*1 : FX/FX, DX/DY, and T/C/ST (contact, coil) cannot be specified.
*2 : FX/FX, DX/DY, T/C/ST (contact, coil), Z, and S cannot be specified.
*3 : Z, V, and T/C/ST (present value) cannot be specified.

2.4 Accessible Devices and Ranges

For devices and ranges that can be accessed in each communication, refer to the following manual.
📖MX Component Version 4 Operating Manual

CHAPTER 3 PROPERTIES OF CONTROLS

This chapter explains the details of properties of the controls.

3.1 Property List

The following table shows the properties of each control.

Control name		Property name
ACT Control	ActUtlType ActMLUtlType	ActLogicalStationNumber
		ActPassword
	ActProgType ActMLProgType	ActATCommand[*1]
		ActATCommandPasswordCancelRetryTimes[*1]
		ActATCommandResponseWaitTime[*1]
		ActBaudRate
		ActCallbackCancelWaitTime[*1]
		ActCallbackDelayTime[*1]
		ActCallbackNumber[*1]
		ActCallbackReceptionWaitingTimeOut[*1]
		ActConnectionCDWaitTime[*1]
		ActConnectionModemReportWaitTime[*1]
		ActConnectUnitNumber
		ActConnectWay[*1]
		ActControl
		ActCpuTimeOut
		ActCpuType
		ActDataBits
		ActDestinationIONumber
		ActDestinationPortNumber
		ActDialNumber[*1]
		ActDidPropertyBit
		ActDisconnectionCDWaitTime[*1]
		ActDisconnectionDelayTime[*1]
		ActDsidPropertyBit
		ActHostAddress
		ActIntelligentPreferenceBit
		ActIONumber
		ActLineType[*1]
		ActMultiDropChannelNumber
		ActNetworkNumber
		ActOutsideLineNumber[*1]
		ActPacketType
		ActParity
		ActPassword

Control name		Property name
ACT Control	ActProgType ActMLProgType	ActPasswordCancelResponseWaitTime*1
		ActPortNumber
		ActProtocolType
		ActSourceNetworkNumber
		ActSourceStationNumber
		ActStationNumber
		ActStopBits
		ActSumCheck
		ActTargetSimulator
		ActThroughNetworkType
		ActTimeOut
		ActTransmissionDelayTime*1
		ActUnitNumber
		ActUnitType
	ActSupportMsg ActMLSupportMsg	—
.NET Control	DotUtlType	ActLogicalStationNumber
		ActPassword
	DotSupportMsg	—

*1 : A property for modem communication. It can only be used for ActProgType.

3.2 Details of Control Properties

This section explains the details of properties set when creating a user application.

(1) Properties of utility setting type controls

The following table shows the properties of the Act(ML)UtlType control and the DotUtlType control.

Property name (Type)	Description	Default value
ActLogicalStation Number (LONG)	A logical station number set in Communication Setup Utility. (Applicable setting range: 0 to 1023)	0(0x00)
ActPassword (BSTR)	Specify a password to disable the password set to the following password protected modules.*1, *2, *3 • Q series-compatible C24 • Q series-compatible E71 • Built-in Ethernet port CPU This setting is ignored when a password protected module is not used.	Empty

*1 : The setting of ActPassword is not necessary if a password is not set.

*2 : Characters exceeded the maximum number of characters for the password are ignored.

*3 : If a characters other than alphanumeric is specified, a character code conversion error (0xF1000001) occurs at the execution of the Open function.

(2) Properties of program setting type controls

The following table shows the properties of the Act(ML)ProgType control.

<table>
<tr><th>Property name
(Type)</th><th>Description</th><th>Default value</th></tr>
<tr><td>ActNetworkNumber
(LONG)</td><td>Specify the network number for MELSECNET/H.
 (Specify "0" (0x00) when specifying the host station.)
Specify the network number for accessing other station with GX Simulator.
Specify the following value for the multi-drop connection (via Q series-compatible C24, QJ61BT11).
<table>
<tr><th>ActIntelligentPreferenceBit value</th><th>Description</th></tr>
<tr><td>0 (0x00)</td><td>Specify the own network.</td></tr>
<tr><td>1 (0x01)</td><td>Specify another network of multi-drop destination.</td></tr>
</table></td><td>0(0x00)</td></tr>
<tr><td>ActStationNumber
(LONG)</td><td>Specify the station number for MELSECNET/H or CC-Link.
 (Specify "0" (0x00) when specifying the host station.)
Specify the station number for accessing other station with GX Simulator.
Specify the following value for the multi-drop connection (via Q series-compatible C24, QJ61BT11).
<table>
<tr><th>ActIntelligentPreferenceBit value</th><th>Description</th></tr>
<tr><td>0 (0x00)</td><td>Specify the own network.</td></tr>
<tr><td>1 (0x01)</td><td>Specify another network of multi-drop destination.</td></tr>
</table></td><td>255(0xFF)</td></tr>
<tr><td>ActUnitNumber
(LONG)</td><td>Specify the module number of the serial communication module or the station number when the target is the Q series-compatible intelligent function module.
This setting is invalid when the target is not a serial communication or Q series-compatible intelligent function module.
For multi-drop link, specify the module number of the target serial communication module.</td><td>0(0x00)</td></tr>
</table>

<table>
<tr><th>Property name (Type)</th><th>Description</th><th>Default value</th></tr>
<tr><td>ActConnectUnit Number (LONG)</td><td>Specify the module number of serial communication module, QE71, or Q series-compatible E71.
For multi-drop link, specify the module number of the requesting serial communication module. For multi-drop link via CPU COM communication, however, the module number of the requesting station is not required. (Specify "0" (00x0))
Specify "0" (0x00) for other than multi-drop link.
For QE71 and Q series-compatible E71, specify the relay target station number. (Fixed to "0" (0x00) for access within the own network)
For access to another network via MELSECNET/10, specify the station number set in the parameter of the connected Ethernet module.</td><td>0(0x00)</td></tr>
<tr><td>ActIONumber (LONG)</td><td>Specify the module I/O number.
For multi-drop link or intelligent function module access, specify the actual I/O number (start I/O number divided by 16) of the target serial communication module or intelligent function module. (For multi-drop link, specify the I/O number of the relayed or requesting station)
Specify "1023" (0x3FF) when making access to another station via the host station CPU or network.</td><td>1023(0x3FF)</td></tr>
<tr><td>ActCpuType (LONG)</td><td>Specify the target CPU to communicate with.
In the parameter, specify any of the CPU types in the following table.
<table>
<tr><th colspan="3">Property value (Property window input value)</th><th rowspan="2">Target CPU</th></tr>
<tr><th>CPU type</th><th>Dec.</th><th>Hex.</th></tr>
<tr><td>CPU_Q00JCPU</td><td>48</td><td>0x30</td><td>Q00JCPU</td></tr>
<tr><td>CPU_Q00UJCPU</td><td>128</td><td>0x80</td><td>Q00UJCPU</td></tr>
<tr><td>CPU_Q00CPU</td><td>49</td><td>0x31</td><td>Q00CPU</td></tr>
<tr><td>CPU_Q00UCPU</td><td>129</td><td>0x81</td><td>Q00UCPU</td></tr>
<tr><td>CPU_Q01CPU</td><td>50</td><td>0x32</td><td>Q01CPU</td></tr>
<tr><td>CPU_Q01UCPU</td><td>130</td><td>0x82</td><td>Q01UCPU</td></tr>
<tr><td>CPU_Q02CPU</td><td>34</td><td>0x22</td><td>Q02(H)CPU</td></tr>
<tr><td>CPU_Q06CPU</td><td>35</td><td>0x23</td><td>Q06HCPU</td></tr>
<tr><td>CPU_Q12CPU</td><td>36</td><td>0x24</td><td>Q12HCPU</td></tr>
<tr><td>CPU_Q25CPU</td><td>37</td><td>0x25</td><td>Q25HCPU</td></tr>
<tr><td>CPU_Q02PHCPU</td><td>69</td><td>0x45</td><td>Q02PHCPU</td></tr>
<tr><td>CPU_Q06PHCPU</td><td>70</td><td>0x46</td><td>Q06PHCPU</td></tr>
<tr><td>CPU_Q12PHCPU</td><td>65</td><td>0x41</td><td>Q12PHCPU</td></tr>
<tr><td>CPU_Q25PHCPU</td><td>66</td><td>0x42</td><td>Q25PHCPU</td></tr>
<tr><td>CPU_Q12PRHCPU</td><td>67</td><td>0x43</td><td>Q12PRHCPU</td></tr>
<tr><td>CPU_Q25PRHCPU</td><td>68</td><td>0x44</td><td>Q25PRHCPU</td></tr>
<tr><td>CPU_Q02UCPU</td><td>131</td><td>0x83</td><td>Q02UCPU</td></tr>
<tr><td>CPU_Q03UDCPU</td><td>112</td><td>0x70</td><td>Q03UDCPU</td></tr>
<tr><td>CPU_Q04UDHCPU</td><td>113</td><td>0x71</td><td>Q04UDHCPU</td></tr>
<tr><td>CPU_Q06UDHCPU</td><td>114</td><td>0x72</td><td>Q06UDHCPU</td></tr>
<tr><td>CPU_Q10UDHCPU</td><td>117</td><td>0x75</td><td>Q10UDHCPU</td></tr>
<tr><td>CPU_Q13UDHCPU</td><td>115</td><td>0x73</td><td>Q13UDHCPU</td></tr>
<tr><td>CPU_Q20UDHCPU</td><td>118</td><td>0x76</td><td>Q20UDHCPU</td></tr>
<tr><td>CPU_Q26UDHCPU</td><td>116</td><td>0x74</td><td>Q26UDHCPU</td></tr>
<tr><td>CPU_Q02CPU_A</td><td>321</td><td>0x141</td><td>Q02(H)CPU-A</td></tr>
<tr><td>CPU_Q06CPU_A</td><td>322</td><td>0x142</td><td>Q06HCPU-A</td></tr>
</table></td><td>34(CPU_ Q02CPU)</td></tr>
</table>

<table>
<tr><th>Property name
(Type)</th><th>Description</th><th>Default value</th></tr>
<tr><td>ActCpuType
(LONG)</td><td>
<table>
<tr><th colspan="3">Property value
(Property window input value)</th><th rowspan="2">Target CPU</th></tr>
<tr><th>CPU type</th><th>Dec.</th><th>Hex.</th></tr>
<tr><td>CPU_Q03UDECPU</td><td>144</td><td>0x90</td><td>Q03UDECPU</td></tr>
<tr><td>CPU_Q04UDEHCPU</td><td>145</td><td>0x91</td><td>Q04UDEHCPU</td></tr>
<tr><td>CPU_Q06UDEHCPU</td><td>146</td><td>0x92</td><td>Q06UDEHCPU</td></tr>
<tr><td>CPU_Q10UDEHCPU</td><td>149</td><td>0x95</td><td>Q10UDEHCPU</td></tr>
<tr><td>CPU_Q13UDEHCPU</td><td>147</td><td>0x93</td><td>Q13UDEHCPU</td></tr>
<tr><td>CPU_Q20UDEHCPU</td><td>150</td><td>0x96</td><td>Q20UDEHCPU</td></tr>
<tr><td>CPU_Q26UDEHCPU</td><td>148</td><td>0x94</td><td>Q26UDEHCPU</td></tr>
<tr><td>CPU_QS001CPU</td><td>96</td><td>0x60</td><td>QS001CPU</td></tr>
<tr><td>CPU_Q50UDEHCPU</td><td>152</td><td>0x98</td><td>Q50UDEHCPU</td></tr>
<tr><td>CPU_Q100UDEHCPU</td><td>154</td><td>0x9A</td><td>Q100UDEHCPU</td></tr>
<tr><td>CPU_Q12DC_V</td><td>88</td><td>0x58</td><td>Q12DCCPU-V</td></tr>
<tr><td>CPU_Q172CPU</td><td>1569</td><td>0x621</td><td>Q172CPU</td></tr>
<tr><td>CPU_Q173CPU</td><td>1570</td><td>0x622</td><td>Q173CPU</td></tr>
<tr><td>CPU_Q172HCPU</td><td>1569</td><td>0x621</td><td>Q172HCPU</td></tr>
<tr><td>CPU_Q173HCPU</td><td>1570</td><td>0x622</td><td>Q173HCPU</td></tr>
<tr><td>CPU_Q172DCPU</td><td>1573</td><td>0x625</td><td>Q172DCPU</td></tr>
<tr><td>CPU_Q173DCPU</td><td>1574</td><td>0x626</td><td>Q173DCPU</td></tr>
<tr><td>CPU_Q172DSCPU</td><td>1578</td><td>0x62A</td><td>Q172DSCPU</td></tr>
<tr><td>CPU_Q173DSCPU</td><td>1579</td><td>0x62B</td><td>Q173DSCPU</td></tr>
<tr><td>CPU_FX0CPU</td><td>513</td><td>0x201</td><td>FX$_{0}$CPU,
FX$_{0S}$CPU</td></tr>
<tr><td>CPU_FX0NCPU</td><td>514</td><td>0x202</td><td>FX$_{0N}$CPU</td></tr>
<tr><td>CPU_FX1CPU</td><td>515</td><td>0x203</td><td>FX$_{1}$CPU</td></tr>
<tr><td>CPU_FX2CPU</td><td>516</td><td>0x204</td><td>FX$_{U}$CPU,
FX$_{2C}$CPU</td></tr>
<tr><td>CPU_FX2NCPU</td><td>517</td><td>0x205</td><td>FX$_{2N}$CPU,
FX$_{2NC}$CPU</td></tr>
<tr><td>CPU_FX1SCPU</td><td>518</td><td>0x206</td><td>FX$_{1S}$CPU</td></tr>
<tr><td>CPU_FX1NCPU</td><td>519</td><td>0x207</td><td>FX$_{1N}$CPU,
FX$_{1NC}$CPU</td></tr>
<tr><td>CPU_FX3GCPU</td><td>521</td><td>0x209</td><td>FX$_{3G}$CPU</td></tr>
<tr><td>CPU_FX3UCCPU</td><td>520</td><td>0x208</td><td>FX$_{3U}$CPU,
FX$_{3UC}$CPU</td></tr>
<tr><td>CPU_BOARD</td><td>1025</td><td>0x401</td><td>For own board access*1</td></tr>
<tr><td>CPU_L02CPU</td><td>161</td><td>0xA1</td><td>L02CPU</td></tr>
<tr><td>CPU_L26CPUBT</td><td>162</td><td>0xA2</td><td>L26CPU-BT</td></tr>
</table>
</td><td>34(CPU_
Q02CPU)</td></tr>
</table>

*1 : Except for CPU boards

<table>
<tr><th>Property name
(Type)</th><th>Description</th><th>Default value</th></tr>
<tr><td>ActPortNumber
(LONG)</td><td>Specify the connection port number of personal computer.
When an Ethernet module is connected, set any value as a port number of the requesting source (personal computer).
When "=0" was specified as a port number, the Station No. ↔ IP information system should be the automatic response system.
(When the system other than the automatic response system via QE71 is selected, set the fixed value "5001".)
When the network board is used, specify the first board as PORT_1, and the second and subsequent boards as PORT_2, PORT_3, and so on.
<table>
<tr><th colspan="3">Property value (Property window input value)</th><th rowspan="2">Description</th></tr>
<tr><th>Port number</th><th>Dec.</th><th>Hex.</th></tr>
<tr><td>PORT_1</td><td>1</td><td>0x01</td><td>Communication port 1</td></tr>
<tr><td>PORT_2</td><td>2</td><td>0x02</td><td>Communication port 2</td></tr>
<tr><td>PORT_3</td><td>3</td><td>0x03</td><td>Communication port 3</td></tr>
<tr><td>PORT_4</td><td>4</td><td>0x04</td><td>Communication port 4</td></tr>
<tr><td>PORT_5</td><td>5</td><td>0x05</td><td>Communication port 5</td></tr>
<tr><td>PORT_6</td><td>6</td><td>0x06</td><td>Communication port 6</td></tr>
<tr><td>PORT_7</td><td>7</td><td>0x07</td><td>Communication port 7</td></tr>
<tr><td>PORT_8</td><td>8</td><td>0x08</td><td>Communication port 8</td></tr>
<tr><td>PORT_9</td><td>9</td><td>0x09</td><td>Communication port 9</td></tr>
<tr><td>PORT_10</td><td>10</td><td>0x0A</td><td>Communication port 10</td></tr>
</table></td><td>1(PORT_1)</td></tr>
<tr><td>ActBaudRate
(LONG)</td><td>Specify the baud rate for serial communication.
<table>
<tr><th>Property value (Property window input value)</th><th>Description</th></tr>
<tr><td>BAUDRATE_300 (300)</td><td>300bps</td></tr>
<tr><td>BAUDRATE_600 (600)</td><td>600bps</td></tr>
<tr><td>BAUDRATE_1200 (1200)</td><td>1200bps</td></tr>
<tr><td>BAUDRATE_2400 (2400)</td><td>2400bps</td></tr>
<tr><td>BAUDRATE_4800 (4800)</td><td>4800bps</td></tr>
<tr><td>BAUDRATE_9600 (9600)</td><td>9600bps</td></tr>
<tr><td>BAUDRATE_19200 (19200)</td><td>19200bps</td></tr>
<tr><td>BAUDRATE_38400 (38400)</td><td>38400bps</td></tr>
<tr><td>BAUDRATE_57600 (57600)</td><td>57600bps</td></tr>
<tr><td>BAUDRATE_115200 (115200)</td><td>115200bps</td></tr>
</table></td><td>19200
(BAUDRATE_ 19200)</td></tr>
<tr><td>ActDataBit
(LONG)</td><td>Specify the number of bits (7 or 8) of the byte data sent and received for serial communication.</td><td>8(DATABIT_8)</td></tr>
<tr><td>ActParity
(LONG)</td><td>Specify the parity system used for serial communication.
<table>
<tr><th>Property value (Property window input value)</th><th>Description</th></tr>
<tr><td>NO_PARITY (0)</td><td>No parity</td></tr>
<tr><td>ODD_PARITY (1)</td><td>Odd</td></tr>
<tr><td>EVEN_PARITY (2)</td><td>Even</td></tr>
</table></td><td>1
(ODD_PARITY)</td></tr>
<tr><td>ActStopBits
(LONG)</td><td>Specify the number of stop bits used for serial communication.
<table>
<tr><th>Property value (Property window input value)</th><th>Description</th></tr>
<tr><td>STOPBIT_ONE (0)</td><td>1 stop bit</td></tr>
<tr><td>STOPBITS_TWO (2)</td><td>2 stop bits</td></tr>
</table></td><td>0
(STOPBIT_ ONE)</td></tr>
</table>

<table>
<tr><th>Property name
(Type)</th><th>Description</th><th>Default value</th></tr>
<tr><td>ActControl
(LONG)</td><td>Specify the control setting of the signal line.
<table>
<tr><th colspan="3">Property value (Property window input value)</th><th rowspan="2">Description</th></tr>
<tr><th>Control setting</th><th>Dec.</th><th>Hex.</th></tr>
<tr><td>TRC_DTR</td><td>1</td><td>0x01</td><td>DTR control</td></tr>
<tr><td>TRC_RTS</td><td>2</td><td>0x02</td><td>RTS control</td></tr>
<tr><td>TRC_DRT_AND_RTS</td><td>7</td><td>0x07</td><td>DTR control and RTS control</td></tr>
<tr><td>TRC_DTR_OR_RTS</td><td>8</td><td>0x08</td><td>DTR control or RTS control</td></tr>
</table></td><td>8
(TRC_DTR_
OR_RTS)</td></tr>
<tr><td>ActHostAddress
(BSTR)</td><td>Pointer which indicates the connection host name (IP address) for Ethernet communication.</td><td>1.1.1.1</td></tr>
<tr><td>ActCpuTimeOut
(LONG)</td><td>Specify the CPU watchdog timer for Ethernet communication. (Unit: Multiplied by 250ms)
Specify the transmission waiting time for serial communication of FXCPU. (Unit: Multiplied by 10ms)</td><td>0(0x00)</td></tr>
<tr><td>ActTimeOut
(LONG)</td><td>Set the time-out value of communication between the personal computer and programmable controller.
(Unit: ms)
A time-out processing may be performed internally depending on the communication path in MX Component.
For details, refer to the following section.
☞ Page 412, Appendix 3 Time-Out Periods</td><td>10000</td></tr>
<tr><td>ActSumCheck
(LONG)</td><td>Specify whether sum check is applied or not.
This setting is valid for serial communication module only.
<table>
<tr><th>Property value (Property window input value)</th><th>Description</th></tr>
<tr><td>NO_SUM_CHECK (0)</td><td>Without sum check</td></tr>
<tr><td>SUM_CHECK (1)</td><td>With sum check</td></tr>
</table></td><td>0
(NO_SUM_
CHECK)</td></tr>
<tr><td>ActSourceNetwork
Number (LONG)</td><td>Specify the requesting network number when the QE71 or Q series-compatible E71 is specified.
Specify the same network number (which is specified in the network parameter) as that of the connected QE71 or Q series-compatible E71.</td><td>0(0x00)</td></tr>
<tr><td>ActSourceStation
Number (LONG)</td><td>Specify the requesting station number (personal computer side station number) when the QE71 or Q series-compatible E71 is specified.
Set the setting to avoid setting the same station number as that of the QE71 set within the same Ethernet loop.</td><td>0(0x00)</td></tr>
<tr><td>ActDestinationPort
Number (LONG)</td><td>Specify the port number of the target when Ethernet communication is specified.
When accessing another network, specify the relay destination port number.
For the system other than the automatic response system, set the following setting.
<table>
<tr><th colspan="2">Communication</th><th>Setting</th></tr>
<tr><td colspan="2">QE71(UDP/IP)</td><td>Fixed to "5001"</td></tr>
<tr><td rowspan="3">Q series-compatible E71
(TCP/IP)</td><td>Other than Redundant CPU</td><td>Fixed to "5002"</td></tr>
<tr><td>MELSOFT connection*2</td><td>Fixed to "5002"</td></tr>
<tr><td>OPS connection*2</td><td>Depending on network parameter</td></tr>
<tr><td colspan="2">Q series-compatible E71(UDP/IP)</td><td>Fixed to "5001"</td></tr>
</table></td><td>0(0x00)</td></tr>
<tr><td>ActDestinationIO
Number (LONG)</td><td>For multi-drop connection (via Q series-compatible C24/CC-Link), specify the actual I/O number (start I/O number divided by 16) of the last access target station. (When the target is an intelligent function module.)
When the target is a CPU, specify "1023" (0x3FF).</td><td>0(0x00)</td></tr>
</table>

*2 : For details, refer to the following manual.
📖 Q Corresponding Ethernet Interface Module User's Manual (Basic)

<table>
<tr><th>Property name
(Type)</th><th>Description</th><th>Default value</th></tr>
<tr><td>ActMultiDropChannel Number (LONG)</td><td>For multi-drop connection (via Q series-compatible C24/CC-Link), specify the multi-drop connection channel number (CH1/CH2).
This setting is invalid for other connections.</td><td>0(0x00)</td></tr>
<tr><td>ActThroughNetwork Type (LONG)</td><td>Specify whether MELSECNET/10 is included in the relayed network when accessing other station via network.
<table>
<tr><th>Property value</th><th>Relayed network</th></tr>
<tr><td>0 (0x00)</td><td>MELSECNET/10 is not included.</td></tr>
<tr><td>1 (0x01)</td><td>MELSECNET/10 is included.</td></tr>
</table></td><td>0(0x00)</td></tr>
<tr><td>ActIntelligent PreferenceBit (LONG)</td><td>For multi-drop connection (via Q series-compatible C24/CC-Link), specify whether to relay the network of the multi-drop link destination. (To differentiate the own network module.)
<table>
<tr><th>Property value</th><th>Relayed network</th></tr>
<tr><td>0 (0x00)</td><td>Another network of multi-drop link destination is not accessed.</td></tr>
<tr><td>1 (0x01)</td><td>Another network of multi-drop link destination is accessed.</td></tr>
</table></td><td>0(0x00)</td></tr>
<tr><td>ActDidPropertyBit (LONG)</td><td>For accessing the Q series-compatible host station intelligent function module (intelligent function module mounted on the host station CPU), the setting of "ActUnitNumber" is not necessary by invalidating the following setting. (Specify the property with the setting of "ActIONumber" only.)
<table>
<tr><th>Property value</th><th>Relayed network</th></tr>
<tr><td>0 (0x00)</td><td>Module number is validated.</td></tr>
<tr><td>1 (0x01)</td><td>Module number is invalidated.</td></tr>
</table></td><td>1(0x01)</td></tr>
<tr><td>ActDsidPropertyBit (LONG)</td><td>For multi-drop connection (via Q series-compatible C24/CC-Link), the setting of "ActDestinationIONumber" is not necessary by invalidating the following setting.
However, when the following setting is invalidated, validate the setting of "ActDidPropertyBit".
(Specify the property with the setting of "ActUnitNumber".)
<table>
<tr><th>Property value</th><th>Relayed network</th></tr>
<tr><td>0 (0x00)</td><td>I/O number of the last access target station is validated.</td></tr>
<tr><td>1 (0x01)</td><td>I/O number of the last access target station is invalidated.</td></tr>
</table></td><td>1(0x01)</td></tr>
<tr><td>ActPacketType (LONG)</td><td>Default setting</td><td>0x01 (PACKET_ PLC1)</td></tr>
<tr><td>ActPassword (BSTR)[*3]</td><td>Specify a password to disable the password set to the following password protected modules.[*4,*5,*6]
• Q series-compatible C24
• Q series-compatible E71
• Built-in Ethernet port CPU
This setting is ignored when a password protected module is not used.</td><td>Empty</td></tr>
</table>

*3 : This property can be used for Ethernet communication.

*4 : The setting of ActPassword is not necessary if a password is not set.

*5 : Characters exceeded the maximum number of characters for the password are ignored.

*6 : If a character other than alphanumeric is specified, a character code conversion error (0xF1000001) occurs at the execution of the Open function.

<table>
<tr><th>Property name
(Type)</th><th>Description</th><th>Default value</th></tr>
<tr><td>ActConnectWay
(LONG)*6</td><td>Set the connection system.
<table>
<tr><th colspan="3">Property value (Property window input value)</th><th rowspan="2">Description</th></tr>
<tr><th>Connection system</th><th>Dec.</th><th>Hex.</th></tr>
<tr><td>TEL_AUTO_CONNECT</td><td>0</td><td>0x00</td><td>Auto line connect</td></tr>
<tr><td>TEL_AUTO_CALLBACK</td><td>1</td><td>0x01</td><td>Auto line connect (Callback fixation)</td></tr>
<tr><td>TEL_AUTO_CALLBACK_NUMBER</td><td>2</td><td>0x02</td><td>Auto line connect (Callback number specification)</td></tr>
<tr><td>TEL_CALLBACK</td><td>3</td><td>0x03</td><td>Callback connect (Fixation)</td></tr>
<tr><td>TEL_CALLBACK_NUMBER</td><td>4</td><td>0x04</td><td>Callback connect (Number specification)</td></tr>
<tr><td>TEL_CALLBACK_REQUEST</td><td>5</td><td>0x05</td><td>Callback request (Fixation)</td></tr>
<tr><td>TEL_CALLBACK_REQUEST_NUMBER</td><td>6</td><td>0x06</td><td>Callback request (Number specification)</td></tr>
<tr><td>TEL_CALLBACK_WAIT</td><td>7</td><td>0x07</td><td>Callback reception waiting</td></tr>
</table></td><td>0(TEL_AUTO_CONNECT)</td></tr>
<tr><td>ActATCommand
(BSTR)*7</td><td>Specify the AT command that initializes the modem. If no value (do not enter the property value) is set, the modem-standard AT command is used. Up to 70 characters can be set.*8</td><td>Empty</td></tr>
<tr><td>ActDialNumber
(BSTR)*7</td><td>Specify the telephone number. Up to 50 characters can be set.*8
(The characters that can be set are: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, -, *, #)</td><td>Empty</td></tr>
<tr><td>ActOutsideLine
Number (BSTR)*7</td><td>Specify the number to access the outside line. Up to 10 characters can be set.*8
(The characters that can be set are: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, -, *, #)</td><td>Empty</td></tr>
<tr><td>ActCallbackNumber
(BSTR)*7</td><td>Specify the callback telephone number. Up to 62 characters can be set.*8
(The characters that can be set are: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, -, *, #)
The setting of the connection system is not necessary when the connection system (ActConnectWay) is other than auto line connect (callback number specification), callback connect (number specification), or callback request (number specification).</td><td>Empty</td></tr>
<tr><td>ActLineType (LONG)*7</td><td>Recognize the line type.
<table>
<tr><th colspan="3">Property value (Property window input value)</th><th rowspan="2">Description</th></tr>
<tr><th>Line type</th><th>Dec.</th><th>Hex.</th></tr>
<tr><td>LINETYPE_PULSE</td><td>0</td><td>0x00</td><td>Pulse (Dial line)</td></tr>
<tr><td>LINETYPE_TONE</td><td>1</td><td>0x01</td><td>Tone (Push button line)</td></tr>
<tr><td>LINETYPE_ISDN</td><td>2</td><td>0x02</td><td>ISDN (ISDN line)</td></tr>
</table></td><td>1(LINETYPE_TONE)</td></tr>
<tr><td>ActConnectionCD
WaitTime (LONG)*7</td><td>Line connection CD signal confirmation time
Set the time for watching the ON/OFF of the CD signal line when the line is connected.
(Applicable setting range: 1 to 999, unit: seconds)
Increase the set time if the CD signal does not turn ON within the preset time depending on the line-connected region (Example: overseas).</td><td>90</td></tr>
<tr><td>ActConnectionModem
ReportWaitTime
(LONG)*7</td><td>Line connection modem waiting time
Set the waiting time for a result code response from the modem after line connection.
(Applicable setting range: 1 to 999, unit: seconds)
Increase the set time if the response speed of the modem is slow.</td><td>5</td></tr>
<tr><td>ActDisconnectionCD
WaitTime (LONG)*7</td><td>Line disconnection CD signal confirmation time
Set the time for watching the ON/OFF of the CD signal line when the line is disconnected.
(Applicable setting range: 1 to 999, unit: seconds)
Increase the set time if the CD signal does not turn OFF within the preset time depending on the line-connected region (Example: overseas).</td><td>5</td></tr>
</table>

*7 : This property can be used for modem communication. It cannot be used for ActMLProgType control.

*8 : If the characters entered exceed the limit of set characters, the characters outside the setting range are ignored.

Property name (Type)	Description	Default value
ActDisconnectionDelay Time (LONG)[*9]	Line disconnection delay time Set the guard time (no communication time) of the escape command sent to the modem. (Applicable setting range: 1 to 999, unit: seconds) Increase the set time if the response speed of the modem is slow.	3
ActTransmissionDelay Time (LONG)[*9]	Data send delay time Set the time to be provided before the AT command is sent. (Applicable setting range: 0 to 999, unit: seconds) Increase the set time if the error code (0xF2100008) is returned though the correct AT command is set. Increase the set time if the response speed of the modem is slow.	0
ActATCommand ResponseWaitTime (LONG)[*9]	AT command send response waiting time (Applicable setting range: 1 to 999, unit: seconds) Increase the set time if the response speed of the modem is slow.	1
ActPasswordCancel ResponseWaitTime (LONG)[*9]	Password cancel response waiting time (Applicable setting range: 1 to 999, unit: seconds) Increase the set time if the quality of the line with the other end is low.	5
ActATCommand PasswordCancelRetry Times (LONG)[*9]	AT command/password cancel send retry count (Applicable setting range: 1 to 999, unit: number of times)	3
ActCallbackCancel WaitTime (LONG)[*9]	Callback line disconnection waiting time (Applicable setting range: 1 to 180, unit: seconds) Increase the set time if the line at the other end (Q series-compatible C24) is not disconnected within the preset time depending on the line-connected region (Example: overseas). The setting is not necessary when the connection system (ActConnectWay) is other than callback connect or callback request.	90
ActCallbackDelayTime (LONG)[*9]	Callback execution delay time (Applicable setting range: 1 to 999, unit: seconds) Increase the set time if the device for relaying connection to the line (Example: modem or like) requires the predetermined time for reconnection after line disconnection. The setting is not necessary when the connection system (ActConnectWay) is other than callback connect or callback request.	20
ActCallbackReception WaitingTimeOut (LONG)[*9]	Callback receive waiting time-out period (Applicable setting range: 1 to 3600, unit: seconds) Set the waiting time for a telephone line connection request from Q series-compatible C24 in a callback receive waiting status. If the set time elapsed, the callback reception waiting status is exited, and since the connection of the telephone line was not completed, the Connect function is terminated abnormally. The setting is not necessary when the connection system (ActConnectWay) is other than callback reception waiting.	120
ActTargetSimulator (LONG)[*10]	Specify the connection destination simulator in start status. When connecting to FXCPU, specify "0" (0x00). **Property value** — **Description** 0 (0x00) — None (When only one simulator is in start status, connects to the simulator in start status. When multiple simulators are in start status, search for the simulators in start status and connect them in alphabetical order.) 1 (0x01) — Simulator A 2 (0x02) — Simulator B 3 (0x03) — Simulator C 4 (0x04) — Simulator D	0(0x00)

*9 : This property can be used for modem communication. It cannot be used for ActMLProgType control.

*10 : This property can be used for GX Simulator2 communication.

<table>
<tr><th>Property name
(Type)</th><th>Description</th><th>Default value</th></tr>
<tr><td>ActUnitType (LONG)</td><td>Specify the module type connected to the physical port.
<table>
<tr><th>Module type</th><th>Connection system</th><th>Value</th></tr>
<tr><td>UNIT_QJ71C24</td><td>Q series-compatible C24 module direct connection</td><td>0x19</td></tr>
<tr><td>UNIT_LJ71C24</td><td>L series-compatible C24 module direct connection</td><td>0x54</td></tr>
<tr><td>UNIT_FX485BD</td><td>FX extended port direct connection (RS-485 connection)</td><td>0x24</td></tr>
<tr><td>UNIT_QJ71E71</td><td>Q series-compatible E71 connection</td><td>0x1A</td></tr>
<tr><td>UNIT_LNETHER</td><td>LCPU Ethernet port connection</td><td>0x52</td></tr>
<tr><td>UNIT_LNETHER_DIRECT</td><td>LCPU Ethernet port direct connection</td><td>0x53</td></tr>
<tr><td>UNIT_QNETHER</td><td>QCPU (Q mode) Ethernet port connection</td><td>0x2C</td></tr>
<tr><td>UNIT_QNETHER_DIRECT</td><td>QCPU (Q mode) Ethernet port direct connection</td><td>0x2D</td></tr>
<tr><td>UNIT_NZ2GF_ETB</td><td>NZ2GF-ETB (Ethernet adapter) connection</td><td>0x59</td></tr>
<tr><td>UNIT_NZ2GF_ETB_DIRECT</td><td>NZ2GF-ETB (Ethernet adapter) direct connection</td><td>0x5A</td></tr>
<tr><td>UNIT_QNCPU</td><td>QCPU (Q mode) RS232C port direct connection</td><td>0x13</td></tr>
<tr><td>UNIT_LNCPU</td><td>LCPU RS232C adapter connection</td><td>0x50</td></tr>
<tr><td>UNIT_QNMOTION</td><td>Q motion CPU-RS232C port direct connection</td><td>0x1C</td></tr>
<tr><td>UNIT_FXCPU</td><td>FXCPU-RS422 port direct connection</td><td>0x0F</td></tr>
<tr><td>UNIT_QNUSB</td><td>QCPU (Q mode) USB port direct connection</td><td>0x16</td></tr>
<tr><td>UNIT_LNUSB</td><td>LCPU USB port direct connection</td><td>0x51</td></tr>
<tr><td>UNIT_QSUSB</td><td>QSCPU (Safety programmable controller) USB connection</td><td>0x29</td></tr>
<tr><td>UNIT_QNMOTIONUSB</td><td>Q motion CPU-USB port direct connection</td><td>0x1D</td></tr>
<tr><td>UNIT_MNETHBOARD</td><td>MELSECNET/H board connection</td><td>0x1E</td></tr>
<tr><td>UNIT_MNETGBOARD</td><td>CC-Link IE Controller Network board connection</td><td>0x2B</td></tr>
<tr><td>UNIT_CCIEFBOARD</td><td>CC-Link IE Field Network board connection</td><td>0x2F</td></tr>
<tr><td>UNIT_CCLINKBOARD</td><td>CC-Link board connection</td><td>0x0C</td></tr>
<tr><td>UNIT_G4QNCPU</td><td>AJ65BT-G4-S3 module direct connection (Q series access)</td><td>0x1B</td></tr>
<tr><td>UNIT_SIMULATOR</td><td>Simulator (GX Simulator) connection</td><td>0x0B</td></tr>
<tr><td>UNIT_SIMULATOR2</td><td>Simulator (GX Simulator 2) connection</td><td>0x30</td></tr>
<tr><td>UNIT_A900GOT</td><td>GOT 900 series/1000 series connection</td><td>0x21</td></tr>
<tr><td>UNIT_GOT_QJ71E71</td><td>GOT (Ethernet) transparent (QJ71E71) connection</td><td>0x40</td></tr>
<tr><td>UNIT_GOT_QNETHER</td><td>GOT (Ethernet) transparent (QCPU (Q mode)) connection</td><td>0x41</td></tr>
<tr><td>UNIT_GOT_LNETHER</td><td>LCPU Ethernet port connection via GOT 1000 series</td><td>0x55</td></tr>
<tr><td>UNIT_GOT_NZ2GF_ETB</td><td>NZ2GF-ETB (Ethernet adapter) connection via GOT 1000 series</td><td>0x5B</td></tr>
<tr><td>UNIT_GOTETHER_QNCPU</td><td>QCPU (Q mode)-RS422 port connection via GOT 1000 series Ethernet port</td><td>0x56</td></tr>
<tr><td>UNIT_GOTETHER_LNCPU</td><td>LnCPU RS232C adapter connection via GOT 1000 series Ethernet port</td><td>0x57</td></tr>
<tr><td>UNIT_GOTETHER_QBUS</td><td>Q series bus connection via GOT 1000 series Ethernet port</td><td>0x58</td></tr>
<tr><td>UNIT_QBF</td><td>Bus connection</td><td>0x1F</td></tr>
</table></td><td>0x13
(UNIT_QNCPU)</td></tr>
</table>

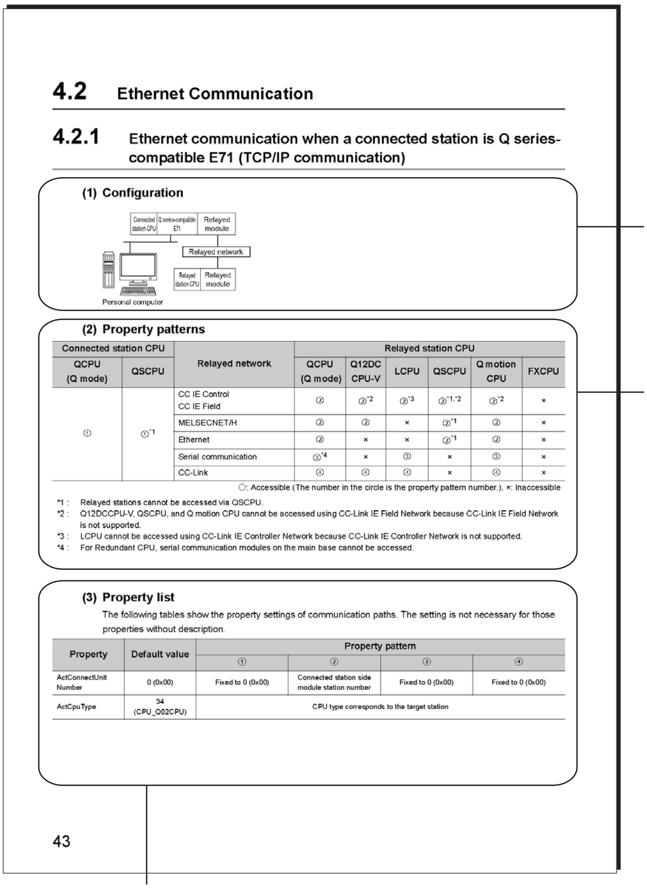
<table>
<tr><th>Property name
(Type)</th><th>Description</th><th>Default value</th></tr>
<tr><td>ActProtocolType
(LONG)</td><td>Specify the communication protocol type of the module (board) to be connected.
<table>
<tr><th>Communication protocol type</th><th>Connection system</th><th>Value</th></tr>
<tr><td>PROTOCOL_SERIAL</td><td>Via serial port</td><td>0x04 (4)</td></tr>
<tr><td>PROTOCOL_USB</td><td>Via USB port</td><td>0x0D (13)</td></tr>
<tr><td>PROTOCOL_TCPIP</td><td>Via TCP/IP</td><td>0x05 (5)</td></tr>
<tr><td>PROTOCOL_UDPIP</td><td>Via UDP/IP</td><td>0x08 (8)</td></tr>
<tr><td>PROTOCOL_MNETH</td><td>Via MELSECNET/H board</td><td>0x0F (15)</td></tr>
<tr><td>PROTOCOL_MNETG</td><td>Via CC-Link IE Controller Network board</td><td>0x14 (20)</td></tr>
<tr><td>PROTOCOL_CCIEF</td><td>Via CC-Link IE Field Network board</td><td>0x15 (21)</td></tr>
<tr><td>PROTOCOL_CCLINK</td><td>Via CC-Link</td><td>0x07 (7)</td></tr>
<tr><td>PROTOCOL_SERIALMODEM</td><td>Via serial port and modem</td><td>0x0E (14)</td></tr>
<tr><td>PROTOCOL_TEL</td><td>Via TEL</td><td>0x0A (10)</td></tr>
<tr><td>PROTOCOL_QBF</td><td>Via Q series bus</td><td>0x10 (16)</td></tr>
<tr><td>PROTOCOL_USBGOT</td><td>Via USB port and GOT</td><td>0x13 (19)</td></tr>
<tr><td>PROTOCOL_SHAREDMEMORY</td><td>Via shared memory server (Simulator)</td><td>0x06 (6)</td></tr>
</table></td><td>0x04
(PROTOCOL_
SERIAL)</td></tr>
</table>

CHAPTER 4 PROPERTY SETTINGS OF COMMUNICATION PATHS

This chapter explains the details of accessible communication paths and property settings.

4.1 Descriptions of Property Setting

The following is the descriptions of how communication paths and properties of control are explained in this chapter.



4.2 Ethernet Communication

4.2.1 Ethernet communication when a connected station is Q series-compatible E71 (TCP/IP communication)

(1) Configuration

(2) Property patterns

Connected station CPU		Relayed network	Relayed station CPU					
QCPU (Q mode)	QSCPU		QCPU (Q mode)	Q12DC CPU-V	LCPU	QSCPU	Q motion CPU	FXCPU
①	①*1	CC IE Control CC IE Field	②	②*2	②*3	②*1,*2	②*2	×
		MELSECNET/H	②	②	×	②*1	②	×
		Ethernet	②	×	×	②*1	②	×
		Serial communication	③*4	×	③	×	③	×
		CC-Link	④	④	④	×	④	×

○: Accessible (The number in the circle is the property pattern number.), ×: Inaccessible

*1 : Relayed stations cannot be accessed via QSCPU.
*2 : Q12DCCPU-V, QSCPU, and Q motion CPU cannot be accessed using CC-Link IE Field Network because CC-Link IE Field Network is not supported.
*3 : LCPU cannot be accessed using CC-Link IE Controller Network because CC-Link IE Controller Network is not supported.
*4 : For Redundant CPU, serial communication modules on the main base cannot be accessed.

(3) Property list

The following tables show the property settings of communication paths. The setting is not necessary for those properties without description.

Property	Default value	Property pattern			
		①	②	③	④
ActConnectUnit Number	0 (0x00)	Fixed to 0 (0x00)	Connected station side module station number	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActCpuType	34 (CPU_Q02CPU)	CPU type corresponds to the target station			

43

(1) Configuration
A diagram of system configuration

(2) Property pattern
Indicates the accessible ranges of the used control and the patterns of the properties.

(3) Property list
(a) Property
Describes the property name.
(b) Default value
・Describes the default value of the property.
・The default value used for changing the property in the program are shown in the parentheses.
(c) Property pattern
Describes the property settings necessary to set the communication settings.
The property pattern numbers correspond to the numbers in the table of "Property pattern".

Point

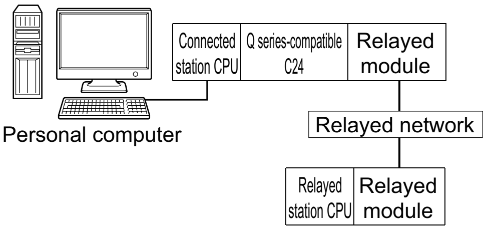
The indicated default values are the property values displayed on the property window of Visual Basic® .NET or Visual C++® .NET.
For the default values required to be entered in values other than the decimal number to change the property values in a program are described in the parentheses.

4.2 Serial Communication

4.2.1 Serial communication when the connected station is Q series-compatible C24

(1) When a relayed module other than the connected station side Q series-compatible C24 exists

(a) Configuration



(b) Property patterns

Connected station CPU		Relayed network	Relayed station CPU					
QCPU (Q mode)	Q motion CPU[1]		QCPU (Q mode)	Q12DC CPU-V	LCPU	QSCPU	Q motion CPU	FXCPU
①	①	CC IE Control CC IE Field	②	②[2]	②[3]	②[2]	②[2]	×
		MELSECNET/H	②	②	×	②	②	×
		Ethernet	②	×	×	②	②	×
		Serial communication	③[4]	×	③	×	③	×
		CC-Link	④	④	④	×	④	④[5]

○: Accessible (The number in the circle is the property pattern number.), ×: Inaccessible

[1] Relayed stations cannot be accessed via Q motion CPU.

[2] Q12DCCPU-V, QSCPU, and Q motion CPU cannot be accessed using CC-Link IE Field Network because CC-Link IE Field Network is not supported.

[3] LCPU cannot be accessed using CC-Link IE Controller Network because CC-Link IE Controller Network is not supported.

[4] For Redundant CPU, serial communication modules on the main base cannot be accessed.

[5] Supported by FX3G and FX3U(C) within the own network only.

(c) Property list

The following table shows the property settings of communication paths. The setting is not necessary for those properties without description.

Property	Default value	Property pattern			
		①	②*1	③	④
ActBaudRate	19200 (BAUDRATE_19200)	Match to the setting of Q series-compatible C24.			
ActConnectUnitNumber	0 (0x00)	Connected station side module station number			
ActControl	8 (TRC_DTR_OR_RTS)	Depending on the used cable.			
ActCpuType	34 (CPU_Q02CPU)	CPU type corresponding to the target station			
ActDestinationIONumber	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No. 1: 992 (0x3E0) No. 2: 993 (0x3E1) No. 3: 994 (0x3E2) No. 4: 995 (0x3E3) For Redundant CPU Control system: 976 (0x3D0) No specification: 1023 (0x3FF)	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No. 1: 992 (0x3E0) No. 2: 993 (0x3E1) No. 3: 994 (0x3E2) No. 4: 995 (0x3E3) For Redundant CPU Control system: 976 (0x3D0) No specification: 1023 (0x3FF)
ActDidPropertyBit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActDsidPropertyBit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActHostAddress	1.1.1.1	Fixed to NULL			
ActIntelligentPreferenceBit	0 (0x00)	Fixed to 0 (0x00)			
ActIONumber*2	1023 (0x3FF)	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No. 1: 992 (0x3E0) No. 2: 993 (0x3E1) No. 3: 994 (0x3E2) No. 4: 995 (0x3E3) For Redundant CPU Control system: 976 (0x3D0) No specification: 1023 (0x3FF)	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No. 1: 992 (0x3E0) No. 2: 993 (0x3E1) No. 3: 994 (0x3E2) No. 4: 995 (0x3E3) For Redundant CPU Control system: 976 (0x3D0) No specification: 1023 (0x3FF)	Connected station side module I/O address	Connected station side module I/O address
ActMultiDropChannelNumber*3	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Multi-drop channel number	Fixed to 0 (0x00)

Property	Default value	Property pattern			
		①	②[*1]	③	④
ActNetwork Number	0 (0x00)	Fixed to 0 (0x00)	Target station side module network number	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActParity	1 (ODD_PARITY)	Match to the setting of Q series-compatible C24.			
ActPortNumber	1 (PORT_1)	Personal computer side COM port number			
ActProtocolType	0x04 (PROTOCOL_ SERIAL)	PROTOCOL_SERIAL (0x04)			
ActStation Number	255 (0xFF)	Fixed to 255 (0xFF)	Target station side module station number	Fixed to 255 (0xFF)	Fixed to 255 (0xFF)
ActThrough NetworkType	0 (0x00)	MELSECNET/10 is not included. : 0 (0x00) MELSECNET/10 is included. : 1 (0x01)			
ActTimeOut	10000	Any value specified by user in ms units			
ActUnitNumber	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station side module station number	Target station side module station number
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_QJ71C24 (0x19)			

[*1] : Note the following considerations when accessing via Ethernet module (Q series-compatible E71).
- For ActNetworkNumber and ActStationNumber, specify the value set in the parameter setting of the target station side Q series-compatible E71.
- Set "Station No. ⇔ IP information" in the parameter setting of Q series-compatible E71. When setting the parameter, specify the setting other than the automatic response system (any of the IP address calculation system, table conversion system, and combined system) for "Station No. ⇔ IP information system".

[*2] : For the I/O address, specify the value of the actual start I/O number divided by 16.

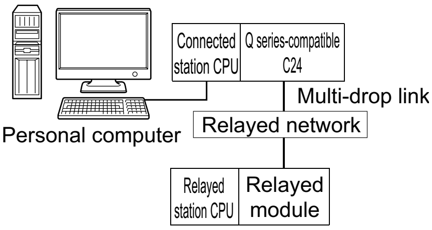
[*3] : Specify the following value for the channel number to be multi-drop linked.
0: Default channel of module
1: Channel 1
2: Channel 2

4

4.2 Serial Communication
4.2.1 Serial communication when the connected station is Q series-compatible C24

(2) When performing multi-drop link on the connected station side Q series-compatible C24 with the relayed module

(a) Configuration



(b) Property patterns

Connected station CPU		Relayed network	Relayed station CPU					
QCPU (Q mode)			QCPU (Q mode)	Q12DC CPU-V	LCPU	QSCPU	Q motion CPU	FXCPU
Independent mode*1	①	Serial communication	②*2	×	②	×	×	×
Synchronous mode*1	×		③*2	×	③	×	×	×

○: Accessible (The number in the circle is the property pattern number.), ×: Inaccessible

*1 : Indicates the CH2 side setting. (The CH1 side is fixed to the independent mode.)

*2 : For Redundant CPU, serial communication modules on the main base cannot be accessed.

(c) Property list

The following table shows the property settings of communication paths. The setting is not necessary for those properties without description.

Property	Default value	Property pattern		
		①	②	③
ActBaudRate	19200 (BAUDRATE_19200)	Match to the setting of Q series-compatible C24.		
ActConnectUnitNumber	0 (0x00)	Connected station side module station number		
ActControl	8 (TRC_DTR_OR_RTS)	Depending on the used cable.		
ActCpuType	34 (CPU_Q02CPU)	CPU type corresponding to the target station		
ActDestinationIONumber	0 (0x00)	Fixed to 0 (0x00)	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No. 1: 992 (0x3E0) No. 2: 993 (0x3E1) No. 3: 994 (0x3E2) No. 4: 995 (0x3E3) For Redundant CPU Control system: 976 (0x3D0) No specification: 1023 (0x3FF)	Fixed to 0 (0x00)

Property	Default value	Property pattern		
		①	②	③
ActDidPropertyBit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 1 (0x01)
ActDsidPropertyBit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 1 (0x01)
ActIntelligentPreferenceBit	0 (0x00)	Fixed to 0 (0x00)		
ActIONumber[*1]	1023 (0x3FF)	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No. 1: 992 (0x3E0) No. 2: 993 (0x3E1) No. 3: 994 (0x3E2) No. 4: 995 (0x3E3) For Redundant CPU Control system: 976 (0x3D0) No specification: 1023 (0x3FF)	Connected station side module I/O address	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No. 1: 992 (0x3E0) No. 2: 993 (0x3E1) No. 3: 994 (0x3E2) No. 4: 995 (0x3E3) For Redundant CPU Control system: 976 (0x3D0) No specification: 1023 (0x3FF)
ActMultiDropChannelNumber[*2]	0 (0x00)	Fixed to 0 (0x00)	Multi-drop channel number	Fixed to 0 (0x00)
ActNetworkNumber	0 (0x00)	Fixed to 0 (0x00)		
ActParity	1 (ODD_PARITY)	Match to the setting of Q series-compatible C24.		
ActPortNumber	1 (PORT_1)	Personal computer side COM port number		
ActProtocolType	0x04 (PROTOCOL_SERIAL)	PROTOCOL_SERIAL (0x04)		
ActStationNumber	255 (0xFF)	Fixed to 255 (0xFF)		
ActThroughNetworkType	0 (0x00)	MELSECNET/10 is not included. : 0 (0x00) MELSECNET/10 is included. : 1 (0x01)		
ActTimeOut	10000	Any value specified by user in ms units		
ActUnitNumber	0 (0x00)	Fixed to 0 (0x00)	Target station side module station number	Fixed to 0 (0x00)
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_QJ71C24 (0x19)		

*1 : For the I/O address, specify the value of the actual start I/O number divided by 16.

*2 : Specify the following value for the channel number to be multi-drop linked.
0: Default channel of module
1: Channel 1
2: Channel 2

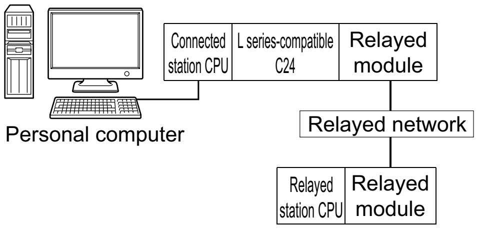
Point

When the connected station side Q series-compatible C24 is set to the synchronous mode, always validate (ON) "sum check (SW06)" for the transmission specification software switch setting of the Q series-compatible C24 parameter.
If it is invalidated (OFF), a communication error occurs and the communication is disabled.

4.2.2 Serial communication when the connected station is L series-compatible C24

(1) When a relayed module other than the connected station side LJ71C24 exists

(a) Configuration



(b) Property patterns

Connected station CPU	Relayed network	Relayed station CPU					
LCPU		QCPU (Q mode)	Q12DC CPU-V	LCPU	QSCPU	Q motion CPU	FXCPU
①	CC IE Field[*1]	④	×	④	×	×	×
	MELSECNET/H	×	×	×	×	×	×
	Ethernet	×	×	×	×	×	×
	Serial communication	②[*2]	×	②	×	×	×
	CC-Link	③	③	③	×	×	×

○: Accessible (The number in the circle is the property pattern number.), ×: Inaccessible

*1 : LCPU cannot be accessed using CC-Link IE Controller Network because CC-Link IE Controller Network is not supported.

*2 : For Redundant CPU, serial communication modules on the main base cannot be accessed.

(c) Property list

The following table shows the property settings of communication paths. The setting is not necessary for those properties without description.

Property	Default value	Property pattern			
		①	②	③	④
ActBaudRate	19200 (BAUDRATE_ 19200)	Match to the setting of L series-compatible C24.			
ActConnectUnit Number	0 (0x00)	Connected station side module station number			
ActControl	8 (TRC_DTR_OR_ RTS)	Depending on the used cable.			
ActCpuType	34 (CPU_Q02CPU)	CPU type corresponding to the target station			

Property	Default value	Property pattern			
		①	②	③	④
ActDestination IONumber	0 (0x00)	Fixed to 0 (0x00)	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No. 1: 992 (0x3E0) No. 2: 993 (0x3E1) No. 3: 994 (0x3E2) No. 4: 995 (0x3E3) For Redundant CPU Control system: 976 (0x3D0) No specification: 1023 (0x3FF)	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No. 1: 992 (0x3E0) No. 2: 993 (0x3E1) No. 3: 994 (0x3E2) No. 4: 995 (0x3E3) For Redundant CPU Control system: 976 (0x3D0) No specification: 1023 (0x3FF)	Fixed to 0 (0x00)
ActDidPropertyBit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Fixed to 1 (0x01)
ActDsidPropertyBit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Fixed to 1 (0x01)
ActIntelligent PreferenceBit	0 (0x00)	Fixed to 0 (0x00)			
ActIONumber[*1]	1023 (0x3FF)	Fixed to 1023 (0x3FF)	Connected station side relayed module I/O address	Connected station side relayed module I/O address	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No. 1: 992 (0x3E0) No. 2: 993 (0x3E1) No. 3: 994 (0x3E2) No. 4: 995 (0x3E3) For Redundant CPU Control system: 976 (0x3D0) No specification: 1023 (0x3FF)
ActMultiDrop ChannelNumber[*2]	0 (0x00)	Fixed to 0 (0x00)	Multi-drop channel number	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActNetworkNumber	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station side module network number
ActParity	1 (ODD_PARITY)	Match to the setting of L series-compatible C24.			
ActPortNumber	1 (PORT_1)	Personal computer side COM port number			
ActProtocolType	0x04 (PROTOCOL_ SERIAL)	PROTOCOL_SERIAL (0x04)			
ActStationNumber	255 (0xFF)	Fixed to 255 (0xFF)	Fixed to 255 (0xFF)	Fixed to 255 (0xFF)	Target station side module station number
ActThroughNetwork Type	0 (0x00)	MELSECNET/10 is not included. : 0 (0x00) MELSECNET/10 is included. : 1 (0x01)			
ActTimeOut	10000	Any value specified by user in ms units			
ActUnitNumber	0 (0x00)	Fixed to 0 (0x00)	Target station side module station number	Target station side module station number	Fixed to 0 (0x00)

Property	Default value	Property pattern			
		①	②	③	④
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_LJ71C24 (0x54)			

*1 : For the I/O address, specify the value of the actual start I/O number divided by 16.

*2 : Specify the following value for the channel number to be multi-drop linked.

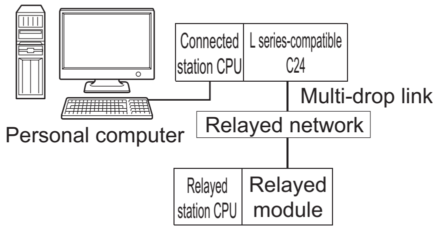
0: Default channel of module

1: Channel 1

2: Channel 2

(2) When performing multi-drop link on the connected station side LJ71C24 with the relayed module

(a) Configuration



(b) Property patterns

Connected station CPU		Relayed network	Relayed station CPU					
LCPU			QCPU (Q mode)	Q12DC CPU-V	LCPU	QSCPU	Q motion CPU	FXCPU
Independent mode	①	Serial communication	②*1	×	②	×	×	×
Synchronous mode	×		③*1	×	③	×	×	×

○: Accessible (The number in the circle is the property pattern number.), ×: Inaccessible

*1 : For Redundant CPU, serial communication modules on the main base cannot be accessed.

(c) Property list

The following table shows the property settings of communication paths. The setting is not necessary for those properties without description.

Property	Default value	Property pattern		
		①	②	③
ActBaudRate	19200 (BAUDRATE_19200)	Match to the setting of L series-compatible C24.		
ActConnectUnitNumber	0 (0x00)	Connected station side module station number		
ActControl	8 (TRC_DTR_OR_RTS)	Depending on the used cable.		
ActCpuType	34 (CPU_Q02CPU)	CPU type corresponding to the target station		
ActDestinationIONumber	0 (0x00)	Fixed to 0 (0x00)	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No. 1: 992 (0x3E0) No. 2: 993 (0x3E1) No. 3: 994 (0x3E2) No. 4: 995 (0x3E3) For Redundant CPU Control system: 976 (0x3D0) No specification: 1023 (0x3FF)	Fixed to 0 (0x00)

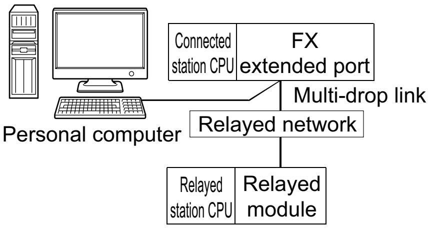
Property	Default value	Property pattern		
		①	②	③
ActDidPropertyBit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 1 (0x01)
ActDsidPropertyBit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 1 (0x01)
ActIntelligentPreferenceBit	0 (0x00)	Fixed to 0 (0x00)		
ActIONumber*1	1023 (0x3FF)	Fixed to 1023 (0x3FF)	Connected station side module I/O address	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No. 1: 992 (0x3E0) No. 2: 993 (0x3E1) No. 3: 994 (0x3E2) No. 4: 995 (0x3E3) For Redundant CPU Control system: 976 (0x3D0) No specification: 1023 (0x3FF)
ActMultiDropChannelNumber*2	0 (0x00)	Fixed to 0 (0x00)	Multi-drop channel number	Fixed to 0 (0x00)
ActNetworkNumber	0 (0x00)	Fixed to 0 (0x00)		
ActParity	1 (ODD_PARITY)	Match to the setting of L series-compatible C24.		
ActPortNumber	1 (PORT_1)	Personal computer side COM port number		
ActProtocolType	0x04 (PROTOCOL_SERIAL)	PROTOCOL_SELIAL (0x04)		
ActStationNumber	255 (0xFF)	Fixed to 255 (0xFF)		
ActThroughNetworkType	0 (0x00)	MELSECNET/10 is not included. : 0 (0x00) MELSECNET/10 is included. : 1 (0x01)		
ActTimeOut	10000	Any value specified by user in ms units		
ActUnitNumber	0 (0x00)	Fixed to 0 (0x00)	Target station side module station number	Fixed to 0 (0x00)
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_LJ71C24 (0x54)		

*1 : For the I/O address, specify the value of the actual start I/O number divided by 16.

*2 : Specify the following value for the channel number to be multi-drop linked.
0: Default channel of module
1: Channel 1
2: Channel 2

4.2.3 Serial communication when the connected station is FX extended port

(1) Configuration



(2) Property patterns

Connected station CPU	Relayed network	Relayed station CPU					
FXCPU		QCPU (Q mode)	Q12DC CPU-V	LCPU	QSCPU	Q motion CPU	FXCPU
①	CC IE Control CC IE Field	×	×	×	×	×	×
	MELSECNET/H	×	×	×	×	×	×
	Ethernet	×	×	×	×	×	×
	Serial communication	×	×	×	×	×	①
	CC-Link	×	×	×	×	×	×

○: Accessible (The number in the circle is the property pattern number.), ×: Inaccessible

(3) Property list

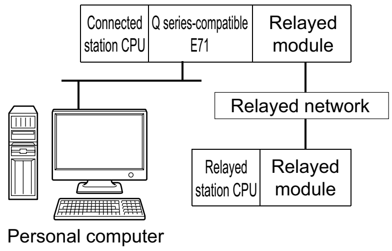
The following table shows the property settings of communication paths. The setting is not necessary for those properties without description.

Property	Default value	Property pattern
		①
ActBaudRate	19200 (BAUDRATE_19200)	Match to the setting of FX extended port
ActControl	8 (TRC_DTR_OR_RTS)	Depending on the used cable
ActCpuTimeOut	0 (0x00)	Any value specified by user in 10ms units
ActCpuType	34 (CPU_Q02CPU)	CPU type corresponding to the target station
ActDataBits	8 (DATABIT_8)	Match to the setting of FX extended port
ActParity	1 (ODD_PARITY)	Match to the setting of FX extended port
ActPortNumber	1 (PORT_1)	Personal computer side COM port number
ActProtocolType	0x04 (PROTOCOL_SERIAL)	PROTOCOL_SERIAL (0x04)
ActStopBits	0 (STOPBIT_ONE)	Match to the setting of FX extended port
ActSumCheck	0 (NO_SUM_CHECK)	Match to the setting of FX extended port
ActTimeOut	10000	Any value specified by user in ms units
ActUnitNumber	0 (0x00)	Target station side module station number
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_FX485BD (0x24)

4.3 Ethernet Communication

4.3.1 Ethernet communication when the connected station is Q series-compatible E71 (TCP/IP communication)

(1) Configuration



(2) Property patterns

Connected station CPU			Relayed network	Relayed station CPU					
QCPU (Q mode)	QSCPU	Q motion CPU*1		QCPU (Q mode)	Q12DC CPU-V	LCPU	QSCPU	Q motion CPU	FXCPU
①	①*2	①*2	CC IE Control CC IE Field	②	②*3	②*4	②*2,*3	②*3	×
			MELSECNET/H	②	②	×	②*2	②	×
			Ethernet	②	×	×	②*2	②	×
			Serial communication	③*5	×	③	×	③	×
			CC-Link	④	④	④	×	④	×

○: Accessible (The number in the circle is the property pattern number.), ×: Inaccessible

*1 : Relayed stations cannot be accessed when the Q motion CPU is Q172CPU, Q173CPU, Q172HCPU, or Q173HCPU.

*2 : Relayed stations cannot be accessed via QSCPU, Q motion CPU.

*3 : Q12DCCPU-V, QSCPU, and Q motion CPU cannot be accessed using CC-Link IE Field Network because CC-Link IE Field Network is not supported.

*4 : LCPU cannot be accessed using CC-Link IE Controller Network because CC-Link IE Controller Network is not supported.

*5 : For Redundant CPU, serial communication modules on the main base cannot be accessed.

(3) Property list

The following table shows the property settings of communication paths. The setting is not necessary for those properties without description.

Property	Default value	Property pattern			
		①	②	③	④
ActConnectUnit Number	0 (0x00)	Fixed to 0 (0x00)	Connected station side module station number	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActCpuType	34 (CPU_Q02CPU)	CPU type corresponds to the target station			
ActDestination IONumber	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No. 1: 992 (0x3E0) No. 2: 993 (0x3E1) No. 3: 994 (0x3E2) No. 4: 995 (0x3E3) For Redundant CPU Control system: 976 (0x3D0) No specification: 1023 (0x3FF)	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No. 1: 992 (0x3E0) No. 2: 993 (0x3E1) No. 3: 994 (0x3E2) No. 4: 995 (0x3E3) For Redundant CPU Control system: 976 (0x3D0) No specification: 1023 (0x3FF)
ActDestination PortNumber	0 (0x00)	5002 for MELSOFT connection Any port number for OPS connection[*1]			
ActDidProperty Bit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActDsidProperty Bit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActHostAddress	1.1.1.1	Host name or IP address of the connected station side module			
ActIONumber[*2]	1023 (0x3FF)	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No. 1: 992 (0x3E0) No. 2: 993 (0x3E1) No. 3: 994 (0x3E2) No. 4: 995 (0x3E3) For Redundant CPU Control system: 976 (0x3D0) No specification: 1023 (0x3FF)	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No. 1: 992 (0x3E0) No. 2: 993 (0x3E1) No. 3: 994 (0x3E2) No. 4: 995 (0x3E3) For Redundant CPU Control system: 976 (0x3D0) No specification: 1023 (0x3FF)	Connected station side relayed module I/O address	Connected station side relayed module I/O address
ActMultiDrop ChannelNumber[*3]	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Multi-drop channel number	Fixed to 0 (0x00)
ActNetwork Number[*4]	0 (0x00)	Target station side module network number	Target station side module network number	Connected station side Q series-compatible E71 network number	Connected station side Q series-compatible E71 network number
ActPassword	Null	Password set to Q series-compatible E71 on the connected station side			

Property	Default value	Property pattern			
		①	②	③	④
ActProtocolType	0x04 (PROTOCOL_ SERIAL)	PROTOCOL_TCPIP (0x05)			
ActSource NetworkNumber	0 (0x00)	Personal computer side network number			
ActSourceStation Number[*5]	0 (0x00)	Personal computer side station number			
ActStation Number[*4]	255 (0xFF)	Target station side module network number	Target station side module network number	Connected station side Q series-compatible E71 network number	Connected station side Q series-compatible E71 network number
ActThrough NetworkType	0 (0x00)	MELSECNET/10 is not included. : 0 (0x00) MELSECNET/10 is included. : 1 (0x01)			
ActTimeOut	10000	Any value specified by user in ms units.			
ActUnitNumber	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station side module station number	Target station side module station number
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_QJ71E71 (0x1A)			

*1 : When using the OPS connection function for Redundant CPU, specify any port number that was set to the network parameter of the CPU. (The setting range is 1025 ≦ port number ≦ 4999 or 5003 ≦ port number ≦ 65534)

*2 : For the I/O address, specify the value of the actual start I/O number divided by 16.

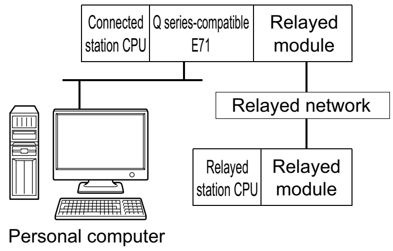
*3 : Specify the following value for the channel number to be multi-drop linked.
0: Default channel of module
1: Channel 1
2: Channel 2

*4 : When the property pattern is ① or ②, specify the value set in the target station side parameter for ActNetworkNumber and ActStationNumber.

*5 : Specify the station number on the personal computer side to avoid the same station number set for Q series-compatible E71 within the same Ethernet loop.

4.3.2 Ethernet communication when the connected station is Q series-compatible E71 (UDP/IP communication)

(1) Configuration



(2) Property patterns

Connected station CPU			Relayed network	Relayed station CPU					
QCPU (Q mode)	QSCPU	Q motion CPU*1		QCPU (Q mode)	Q12DC CPU-V	LCPU	QSCPU	Q motion CPU	FXCPU
①	①*2	①*2	CC IE Control CC IE Field	②	②*3	②*4	②*2,*3	②*3	×
			MELSECNET/H	②	②	×	②*2	②	×
			Ethernet	②	×	×	②*2	②	×
			Serial communication	③*5	×	③	×	③	×
			CC-Link	④	④	④	×	④	×

○: Accessible (The number in the circle is the property pattern number.), ×: Inaccessible

*1 : Relayed stations cannot be accessed when the Q motion CPU is Q172CPU, Q173CPU, Q172HCPU, or Q173HCPU.

*2 : Relayed stations cannot be accessed via QSCPU, Q motion CPU.

*3 : Q12DCCPU-V, QSCPU, and Q motion CPU cannot be accessed using CC-Link IE Field Network because CC-Link IE Field Network is not supported.

*4 : LCPU cannot be accessed using CC-Link IE Controller Network because CC-Link IE Controller Network is not supported.

*5 : For Redundant CPU, serial communication modules on the main base cannot be accessed.

(3) Property list

The following table shows the property settings of communication paths. The setting is not necessary for those properties without description.

Property	Default value	Property pattern			
		①	②	③	④
ActConnectUnit Number	0 (0x00)	Fixed to 0 (0x00)	Connected station side module station number	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActCpuType	34 (CPU_Q02CPU)	CPU type corresponding to the target station			
ActDestination IONumber	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No. 1: 992 (0x3E0) No. 2: 993 (0x3E1) No. 3: 994 (0x3E2) No. 4: 995 (0x3E3) For Redundant CPU Control system: 976 (0x3D0) No specification: 1023 (0x3FF)	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No. 1: 992 (0x3E0) No. 2: 993 (0x3E1) No. 3: 994 (0x3E2) No. 4: 995 (0x3E3) For Redundant CPU Control system: 976 (0x3D0) No specification: 1023 (0x3FF)
ActDidProperty Bit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActDsidProperty Bit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActHostAddress	1.1.1.1	Host name or IP address of the connected station side module			
ActIONumber[*1]	1023 (0x3FF)	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No. 1: 992 (0x3E0) No. 2: 993 (0x3E1) No. 3: 994 (0x3E2) No. 4: 995 (0x3E3) For Redundant CPU Control system: 976 (0x3D0) No specification: 1023 (0x3FF)	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No. 1: 992 (0x3E0) No. 2: 993 (0x3E1) No. 3: 994 (0x3E2) No. 4: 995 (0x3E3) For Redundant CPU Control system: 976 (0x3D0) No specification: 1023 (0x3FF)	Connected station side relayed module I/O address	Connected station side relayed module I/O address
ActMultiDrop ChannelNumber[*2]	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Multi-drop channel number	Fixed to 0 (0x00)
ActNetwork Number[*3]	0 (0x00)	Target station side module network number	Target station side module network number	Connected station side Q series-compatible E71 network number	Connected station side Q series-compatible E71 network number
ActPassword	Null	Password set to Q series-compatible E71 on the connected station side			
ActPortNumber[*4]	1 (PORT_1)	Personal computer side port number			

Property	Default value	Property pattern			
		①	②	③	④
ActProtocolType	0x04 (PROTOCOL_ SERIAL)	PROTOCOL_UDPIP (0x08)			
ActSource NetworkNumber[*4]	0 (0x00)	Personal computer side network number			
ActSourceStation Number[*5]	0 (0x00)	Personal computer side station number			
ActStation Number[*3]	255 (0xFF)	Target station side module network number	Target station side module network number	Connected station side Q series-compatible E71 network number	Connected station side Q series-compatible E71 network number
ActThrough NetworkType	0 (0x00)	MELSECNET/10 is not included. : 0 (0x00) MELSECNET/10 is included. : 1 (0x01)			
ActTimeOut	10000	Any value specified by user in ms units.			
ActUnitNumber	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station side module station number	Target station side module station number
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_QJ71E71 (0x1A)			

*1 : For the I/O address, specify the value of the actual start I/O number divided by 16.

*2 : Specify the following value for the channel number to be multi-drop linked.
0: Default channel of module
1: Channel 1
2: Channel 2

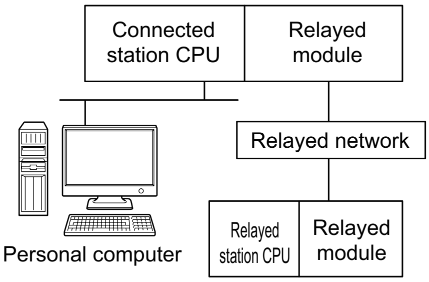
*3 : When the property pattern is ① or ②, specify the value set in the target station side parameter for ActNetworkNumber and ActStationNumber.

*4 : Do not use 1 to 1024 of ActPortNumber.

*5 : Specify the station number on the personal computer side to avoid the same station number set for Q series-compatible E71 within the same Ethernet loop.

4.3.3 Ethernet communication when the connected station is LCPU (TCP/IP communication)

(1) Configuration



(2) Property patterns

Connected station CPU	Relayed network	Relayed station CPU					
LCPU		QCPU (Q mode)	Q12DC CPU-V	LCPU	QSCPU	Q motion CPU	FXCPU
①	CC IE Field*1	④	×	④	×	×	×
	MELSECNET/H	×	×	×	×	×	×
	Ethernet	×	×	×	×	×	×
	Serial communication	②*2	×	②	×	×	×
	CC-Link	③	③	③	×	×	×

○: Accessible (The number in the circle is the property pattern number.), ×: Inaccessible

*1 : LCPU cannot be accessed using CC-Link IE Controller Network because CC-Link IE Controller Network is not supported.

*2 : For Redundant CPU, serial communication modules on the main base cannot be accessed.

(3) Property list

The following table shows the property settings of communication paths. The setting is not necessary for those properties without description.

Property	Default value	Property pattern			
		①	②	③	④
ActCpuType	34 (CPU_Q02CPU)	CPU type corresponding to the target station			

Property	Default value	Property pattern			
		①	②	③	④
ActDestination IONumber	0 (0x00)	Fixed to 0 (0x00)	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No. 1: 992 (0x3E0) No. 2: 993 (0x3E1) No. 3: 994 (0x3E2) No. 4: 995 (0x3E3) For Redundant CPU Control system: 976 (0x3D0) No specification: 1023 (0x3FF)	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No. 1: 992 (0x3E0) No. 2: 993 (0x3E1) No. 3: 994 (0x3E2) No. 4: 995 (0x3E3) For Redundant CPU Control system: 976 (0x3D0) No specification: 1023 (0x3FF)	Fixed to 0 (0x00)
ActDidPropertyBit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Fixed to 1 (0x01)
ActDsidPropertyBit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Fixed to 1 (0x01)
ActHostAddress	1.1.1.1	Host name or IP address of the connected station side module			
ActIntelligent PreferenceBit	0 (0x00)	Fixed to 0 (0x00)	Target station Q mode/Q12DCCPU-V/LCPU : 1 (0x01) Other than the above : 0 (0x00)	Target station Q mode/Q12DCCPU-V/LCPU : 1 (0x01) Other than the above : 0 (0x00)	Fixed to 0 (0x00)
ActIONumber*1	1023 (0x3FF)	For single CPU Fixed to 1023 (0x3FF)	Connected station side relayed module I/O address	Connected station side relayed module I/O address	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No. 1: 992 (0x3E0) No. 2: 993 (0x3E1) No. 3: 994 (0x3E2) No. 4: 995 (0x3E3) For Redundant CPU Control system: 976 (0x3D0) No specification: 1023 (0x3FF)
ActMultiDrop ChannelNumber*2	0 (0x00)	Fixed to 0 (0x00)	Multi-drop channel number	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActNetworkNumber*3	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station side module network number
ActPassword	Null	Password set to the connected station side			
ActProtocolType	0x04 (PROTOCOL_ SERIAL)	PROTOCOL_TCPIP (0x05)			
ActStationNumber*3	255 (0xFF)	Fixed to 255 (0xFF)	Fixed to 255 (0xFF)	Fixed to 255 (0xFF)	Target station side module station number
ActThroughNetwork Type	0 (0x00)	MELSECNET/10 is not included. : 0 (0x00) MELSECNET/10 is included. : 1 (0x01)			
ActTimeOut	10000	Any value specified by user in ms units.			

Property	Default value	Property pattern			
		①	②	③	④
ActUnitNumber	0 (0x00)	Fixed to 0 (0x00)	Target station side module station number	Target station side module station number	Fixed to 0 (0x00)
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_LNETHER (0x52)			

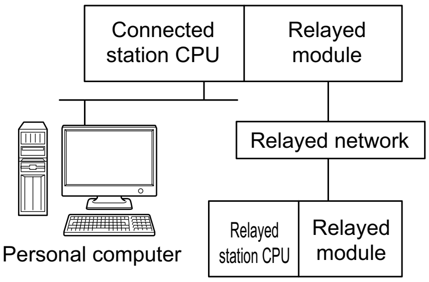
*1 : For the I/O address, specify the value of the actual start I/O number divided by 16.

*2 : Specify the following value for the channel number to be multi-drop linked.
0: Default channel of module
1: Channel 1
2: Channel 2

*3 : When the property pattern is ① or ②, specify the value set in the target station side parameter for ActNetworkNumber and ActStationNumber.

4.3.4 Ethernet communication when the connected station is LCPU (UDP/IP communication)

(1) Configuration



(2) Property patterns

Connected station CPU	Relayed network	Relayed station CPU					
LCPU		QCPU (Q mode)	Q12DC CPU-V	LCPU	QSCPU	Q motion CPU	FXCPU
①	CC IE Field[*1]	④	×	④	×	×	×
	MELSECNET/H	×	×	×	×	×	×
	Ethernet	×	×	×	×	×	×
	Serial communication	②[*2]	×	②	×	×	×
	CC-Link	③	③	③	×	×	×

○: Accessible (The number in the circle is the property pattern number.), ×: Inaccessible

[*1] : LCPU cannot be accessed using CC-Link IE Controller Network because CC-Link IE Controller Network is not supported.

[*2] : For Redundant CPU, serial communication modules on the main base cannot be accessed.

(3) Property list

The following table shows the property settings of communication paths. The setting is not necessary for those properties without description.

Property	Default value	Property pattern			
		①	②	③	④
ActCpuType	34 (CPU_Q02CPU)	CPU type corresponding to the target station			

Property	Default value	Property pattern			
		①	②	③	④
ActDestination IONumber	0 (0x00)	Fixed to 0 (0x00)	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No. 1: 992 (0x3E0) No. 2: 993 (0x3E1) No. 3: 994 (0x3E2) No. 4: 995 (0x3E3) For Redundant CPU Control system: 976 (0x3D0) No specification: 1023 (0x3FF)	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No. 1: 992 (0x3E0) No. 2: 993 (0x3E1) No. 3: 994 (0x3E2) No. 4: 995 (0x3E3) For Redundant CPU Control system: 976 (0x3D0) No specification: 1023 (0x3FF)	Fixed to 0 (0x00)
ActDidPropertyBit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Fixed to 1 (0x01)
ActDsidPropertyBit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Fixed to 1 (0x01)
ActHostAddress	1.1.1.1	For communication with specified IP address: Host name or IP address of the connected station side module For direct communication without specified IP address: The specification is invalid.			
ActIntelligent PreferenceBit	0 (0x00)	Fixed to 0 (0x00)	Target station Q mode/ Q12DCCPU-V/ LCPU :1 (0x01) Other than the above :0 (0x00)	Target station Q mode/ Q12DCCPU-V/ LCPU :1 (0x01) Other than the above :0 (0x00)	Fixed to 0 (0x00)
ActIONumber[*1]	1023 (0x3FF)	For single CPU Fixed to 1023 (0x3FF)	Connected station side module I/O address	Connected station side module I/O address	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No. 1: 992 (0x3E0) No. 2: 993 (0x3E1) No. 3: 994 (0x3E2) No. 4: 995 (0x3E3) For Redundant CPU Control system: 976 (0x3D0) No specification: 1023 (0x3FF)
ActMultiDrop ChannelNumber[*2]	0 (0x00)	Fixed to 0 (0x00)	Multi-drop channel number	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActNetworkNumber[*3]	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station side module network number
ActPassword	Null	Remote password set to the connected station			
ActProtocolType	0x04 (PROTOCOL_ SERIAL)	PROTOCOL_UDPIP (0x08)			
ActStationNumber[*3]	255 (0xFF)	Fixed to 255 (0xFF)	Fixed to 255 (0xFF)	Fixed to 255 (0xFF)	Target station side module station number
ActThroughNetwork Type	0 (0x00)	MELSECNET/10 is not included. : 0 (0x00) MELSECNET/10 is included. : 1 (0x01)			

Property	Default value	Property pattern			
		①	②	③	④
ActTimeOut	10000	Any value specified by user in ms units			
ActUnitNumber	0 (0x00)	Fixed to 0 (0x00)	Target station side module station number	Target station side module station number	Fixed to 0 (0x00)
ActUnitType	0x13 (UNIT_QNCPU)	For communication with specified IP address: UNIT_LNETHER(0x52) For direct communication without specified IP address: UNIT_LNETHER_DIRECT(0x53)			

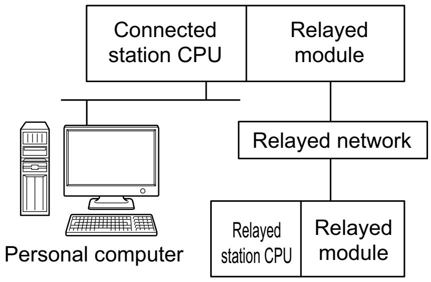
*1 : For the I/O address, specify the value of the actual start I/O number divided by 16.

*2 : Specify the following value for the channel number to be multi-drop linked.
0: Default channel of module
1: Channel 1
2: Channel 2

*3 : When the property pattern is ① or ②, specify the value set in the target station side parameter for ActNetworkNumber and ActStationNumber.

4.3.5 Ethernet communication when the connected station is built-in Ethernet pot QCPU (TCP)

(1) Configuration



(2) Property patterns

Connected station CPU	Relayed network	Relayed station CPU					
QnUDE(H) CPU		QCPU (Q mode)	Q12DC CPU-V	LCPU	QSCPU	Q motion CPU	FXCPU
①	CC IE Control CC IE Field	②	②*1	②*2	②*1	②*1	×
	MELSECNET/H	②	②	×	②	②	×
	Ethernet	②	×	×	②	②	×
	Serial communication	③*3	×	③	×	③	×
	CC-Link	④	④	④	×	④	×

Connected station CPU	Relayed network	Relayed station CPU					
Q12DCCPU-V		QCPU (Q mode)	Q12DC CPU-V	LCPU	QSCPU	Q motion CPU	FXCPU
①	CC IE Control CC IE Field	②	②*1	②*2	②*1	②*1	×
	MELSECNET/H	②	②	×	②	②	×
	Ethernet	×	×	×	×	×	×
	Serial communication	×	×	×	×	×	×
	CC-Link	④	④	④	×	④	×

○: Accessible (The number in the circle is the property pattern number.), ×: Inaccessible

*1 : Q12DCCPU-V, QSCPU, and Q motion CPU cannot be accessed using CC-Link IE Field Network because CC-Link IE Field Network is not supported.

*2 : LCPU cannot be accessed using CC-Link IE Controller Network because CC-Link IE Controller Network is not supported.

*3 : For Redundant CPU, serial communication modules on the main base cannot be accessed.

Connected station CPU	Relayed network	Relayed station CPU					
Q motion CPU*4		QCPU (Q mode)	Q12DC CPU-V	LCPU	QSCPU	Q motion CPU	FXCPU
①	CC IE Control CC IE Field	×	×	×	×	×	×
	MELSECNET/H	×	×	×	×	×	×
	Ethernet	×	×	×	×	×	×
	Serial communication	×	×	×	×	×	×
	CC-Link	×	×	×	×	×	×

○: Accessible (The number in the circle is the property pattern number.), ×: Inaccessible

*4 : Relayed stations cannot be accessed when the Q motion CPU is Q172CPU, Q173CPU, Q172HCPU, or Q173HCPU.

(3) Property list

The following table shows the property settings of communication paths. The setting is not necessary for those properties without description.

Property	Default value	Property pattern			
		①	②	③	④
ActCpuType	34 (CPU_Q02CPU)	CPU type corresponding to the target station			
ActDestination IONumber	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No. 1: 992 (0x3E0) No. 2: 993 (0x3E1) No. 3: 994 (0x3E2) No. 4: 995 (0x3E3) For Redundant CPU Control system: 976 (0x3D0) No specification: 1023 (0x3FF)	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No. 1: 992 (0x3E0) No. 2: 993 (0x3E1) No. 3: 994 (0x3E2) No. 4: 995 (0x3E3) For Redundant CPU Control system: 976 (0x3D0) No specification: 1023 (0x3FF)
ActDestination PortNumber	0 (0x00)	5007			
ActDidProperty Bit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActDsidProperty Bit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActHostAddress	1.1.1.1	Host name or IP address of the connected station side module			
ActIntelligent PreferenceBit	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station Q mode/ Q12DCCPU-V/ LCPU :1 (0x01) Other than the above :0 (0x00)	Target station Q mode/ Q12DCCPU-V/ LCPU :1 (0x01) Other than the above :0 (0x00)

4.3 Ethernet Communication
4.3.5 Ethernet communication when the connected station is built-in Ethernet pot QCPU (TCP)

Property	Default value	Property pattern			
		①	②	③	④
ActIONumber[*1]	1023 (0x3FF)	For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No. 1: 992 (0x3E0) No. 2: 993 (0x3E1) No. 3: 994 (0x3E2) No. 4: 995 (0x3E3)	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No. 1: 992 (0x3E0) No. 2: 993 (0x3E1) No. 3: 994 (0x3E2) No. 4: 995 (0x3E3) For Redundant CPU Control system: 976 (0x3D0) No specification: 1023 (0x3FF)	Connected station side relayed module I/O address	Connected station side relayed module I/O address
ActMultiDrop ChannelNumber[*2]	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Multi-drop channel number	Fixed to 0 (0x00)
ActNetwork Number[*3]	0 (0x00)	Fixed to 0 (0x00)	Target station side module network number	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActPassword	Null	Remote password set to the connected station			
ActProtocolType	0x04 (PROTOCOL_ SERIAL)	PROTOCOL_TCPIP (0x05)			
ActStation Number[*3]	255 (0xFF)	Fixed to 255 (0xFF)	Target station side module station number	Fixed to 255 (0xFF)	Fixed to 255 (0xFF)
ActThrough NetworkType	0 (0x00)	MELSECNET/10 is not included. : 0 (0x00) MELSECNET/10 is included. : 1 (0x01)			
ActTimeOut	10000	Any value specified by user in ms units			
ActUnitNumber	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station side module station number	Target station side module station number
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_QNETHER (0x2C)			

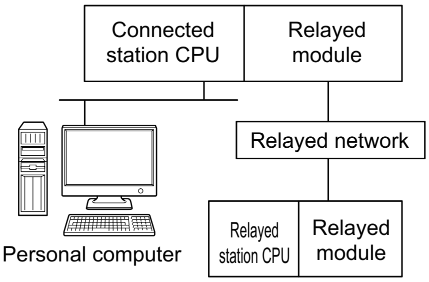
*1 : For the I/O address, specify the value of the actual start I/O number divided by 16.

*2 : Specify the following value for the channel number to be multi-drop linked.
0: Default channel of module
1: Channel 1
2: Channel 2

*3 : When the property pattern is ① or ②, specify the value set in the target station side parameter for ActNetworkNumber and ActStationNumber.

4.3.6 Ethernet communication when the connected station is built-in Ethernet pot QCPU (UDP)

(1) Configuration



(2) Property patterns

Connected station CPU	Relayed network	Relayed station CPU					
QnUDE(H) CPU		QCPU (Q mode)	Q12DC CPU-V	LCPU	QSCPU	Q motion CPU	FXCPU
①	CC IE Control CC IE Field	②	②*1	②*2	②*1	②*1	×
	MELSECNET/H	②	②	×	②	②	×
	Ethernet	②	×	×	②	②	×
	Serial communication	③*3	×	③	×	③	×
	CC-Link	④	④	④	×	④	×

Connected station CPU	Relayed network	Relayed station CPU					
Q12DCCPU-V		QCPU (Q mode)	Q12DC CPU-V	LCPU	QSCPU	Q motion CPU	FXCPU
①	CC IE Control CC IE Field	②	②*1	②*2	②*1	②*1	×
	MELSECNET/H	②	②	×	②	②	×
	Ethernet	×	×	×	×	×	×
	Serial communication	×	×	×	×	×	×
	CC-Link	④	④	④	×	④	×

○: Accessible (The number in the circle is the property pattern number.), ×: Inaccessible

*1 : Q12DCCPU-V, QSCPU, and Q motion CPU cannot be accessed using CC-Link IE Field Network because CC-Link IE Field Network is not supported.

*2 : LCPU cannot be accessed using CC-Link IE Controller Network because CC-Link IE Controller Network is not supported.

*3 : For Redundant CPU, serial communication modules on the main base cannot be accessed.

Connected station CPU	Relayed network	Relayed station CPU					
Q motion CPU*4		QCPU (Q mode)	Q12DC CPU-V	LCPU	QSCPU	Q motion CPU	FXCPU
①	CC IE Control CC IE Field	×	×	×	×	×	×
	MELSECNET/H	×	×	×	×	×	×
	Ethernet	×	×	×	×	×	×
	Serial communication	×	×	×	×	×	×
	CC-Link	×	×	×	×	×	×

○: Accessible (The number in the circle is the property pattern number.), ×: Inaccessible

*4 : Relayed stations cannot be accessed when the Q motion CPU is Q172CPU, Q173CPU, Q172HCPU, or Q173HCPU.

(3) Property list

The following table shows the property settings of communication paths. The setting is not necessary for those properties without description.

Property	Default value	Property pattern			
		①	②	③	④
ActCpuType	34 (CPU_Q02CPU)	CPU type corresponding to the target station			
ActDestination IONumber	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No. 1: 992 (0x3E0) No. 2: 993 (0x3E1) No. 3: 994 (0x3E2) No. 4: 995 (0x3E3) For Redundant CPU Control system: 976 (0x3D0) No specification: 1023 (0x3FF)	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No. 1: 992 (0x3E0) No. 2: 993 (0x3E1) No. 3: 994 (0x3E2) No. 4: 995 (0x3E3) For Redundant CPU Control system: 976 (0x3D0) No specification: 1023 (0x3FF)
ActDestination PortNumber	0 (0x00)	5006			
ActDidProperty Bit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActDsidProperty Bit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActHostAddress	1.1.1.1	For communication with specified IP address: Host name or IP address of the connected station side module For direct communication without specified IP address: The specification is invalid.			
ActIntelligent PreferenceBit	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station Q mode/ Q12DCCPU-V/ LCPU :1 (0x01) Other than the above :0 (0x00)	Target station Q mode/ Q12DCCPU-V/ LCPU :1 (0x01) Other than the above :0 (0x00)

Property	Default value	Property pattern			
		①	②	③	④
ActIONumber[*1]	1023 (0x3FF)	For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No. 1: 992 (0x3E0) No. 2: 993 (0x3E1) No. 3: 994 (0x3E2) No. 4: 995 (0x3E3)	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No. 1: 992 (0x3E0) No. 2: 993 (0x3E1) No. 3: 994 (0x3E2) No. 4: 995 (0x3E3) For Redundant CPU Control system: 976 (0x3D0) No specification: 1023 (0x3FF)	Connected station side relayed module I/O address	Connected station side relayed module I/O address
ActMultiDrop ChannelNumber[*2]	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Multi-drop channel number	Fixed to 0 (0x00)
ActNetwork Number[*3]	0 (0x00)	Fixed to 0 (0x00)	Target station side module network number	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActPassword	Null	Remote password set to the connected station			
ActProtocolType	0x04 (PROTOCOL_ SERIAL)	PROTOCOL_UDPIP (0x08)			
ActStation Number[*3]	255 (0xFF)	Fixed to 255 (0xFF)	Target station side module station number	Fixed to 255 (0xFF)	Fixed to 255 (0xFF)
ActThrough NetworkType	0 (0x00)	MELSECNET/10 is not included. : 0 (0x00) MELSECNET/10 is included. : 1 (0x01)			
ActTimeOut	10000	Any value specified by user in ms units			
ActUnitNumber	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station side module station number	Target station side module station number
ActUnitType	0x13 (UNIT_QNCPU)	For communication with specified IP address: UNIT_QNETHER (0x2C) For direct communication without specified IP address: UNIT_QNETHER_DIRECT (0x2D)			

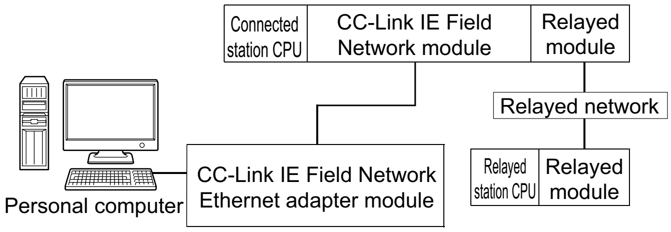
*1 : For the I/O address, specify the value of the actual start I/O number divided by 16.

*2 : Specify the following value for the channel number to be multi-drop linked.
0: Default channel of module
1: Channel 1
2: Channel 2

*3 : When the property pattern is ① or ②, specify the value set in the target station side parameter for ActNetworkNumber and ActStationNumber.

4.3.7 Ethernet communication when the connected station is CC-Link IE Field Network Ethernet adapter module (TCP)

(1) Configuration



(2) Property patterns

Connected station CPU	Relayed network	Relayed station CPU					
QnUDE(H) CPU		QCPU (Q mode)	Q12DC CPU-V	LCPU	QSCPU	Q motion CPU	FXCPU
①	CC IE Control CC IE Field	②	②*1	②*2	×	×	×
	MELSECNET/H	②	②	×	×	×	×
	Ethernet	②	×	×	×	×	×
	Serial communication	③	×	③	×	×	×
	CC-Link	④	④	④	×	×	×

Connected station CPU	Relayed network	Relayed station CPU					
LCPU		QCPU (Q mode)	Q12DC CPU-V	LCPU	QSCPU	Q motion CPU	FXCPU
①	CC IE Field*2	②	×	②	×	×	×
	MELSECNET/H	×	×	×	×	×	×
	Ethernet	×	×	×	×	×	×
	Serial communication	③	×	③	×	×	×
	CC-Link	④	④	④	×	×	×

○: Accessible (The number in the circle is the property pattern number.), ×: Inaccessible

*1 : Q12DCCPU-V cannot be accessed using CC-Link IE Field Network because CC-Link IE Field Network is not supported.

*2 : LCPU cannot be accessed using CC-Link IE Controller Network because CC-Link IE Controller Network is not supported.

(3) Property list

The following table shows the property settings of communication paths. The setting is not necessary for those properties without description.

Property	Default value	Property pattern			
		①	②	③	④
ActCpuType	34 (CPU_Q02CPU)	CPU type corresponding to the target station			
ActDestination IONumber	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No. 1: 992 (0x3E0) No. 2: 993 (0x3E1) No. 3: 994 (0x3E2) No. 4: 995 (0x3E3) For Redundant CPU Control system: 976 (0x3D0) No specification: 1023 (0x3FF)	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No. 1: 992 (0x3E0) No. 2: 993 (0x3E1) No. 3: 994 (0x3E2) No. 4: 995 (0x3E3) For Redundant CPU Control system: 976 (0x3D0) No specification: 1023 (0x3FF)
ActDidProperty Bit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActDsidProperty Bit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActHostAddress	1.1.1.1	Host name or IP address of the CC-Link IE Field Network Ethernet adapter module			
ActIntelligent PreferenceBit	0 (0x00)	Fixed to 0 (0x00)			
ActIONumber*1	1023 (0x3FF)	For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No. 1: 992 (0x3E0) No. 2: 993 (0x3E1) No. 3: 994 (0x3E2) No. 4: 995 (0x3E3)	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No. 1: 992 (0x3E0) No. 2: 993 (0x3E1) No. 3: 994 (0x3E2) No. 4: 995 (0x3E3) For Redundant CPU Control system: 976 (0x3D0) No specification: 1023 (0x3FF)	Connected station side relayed module I/O address	Connected station side relayed module I/O address
ActMultiDrop ChannelNumber*2	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Multi-drop channel number	Fixed to 0 (0x00)
ActNetwork Number*3	0 (0x00)	Connected station side CC-Link IE Field Network module station number	Target station side module network number	Connected station side CC-Link IE Field Network module station number	Connected station side CC-Link IE Field Network module station number
ActProtocolType	0x04 (PROTOCOL_ SERIAL)	PROTOCOL_TCPIP (0x05)			

Property	Default value	Property pattern			
		①	②	③	④
ActStation Number[3]	255 (0xFF)	Connected station side CC-Link IE Field Network module station number	Target station side module station number	Connected station side CC-Link IE Field Network module station number	Connected station side CC-Link IE Field Network module station number
ActThrough NetworkType	0 (0x00)	MELSECNET/10 is not included. : 0 (0x00) MELSECNET/10 is included. : 1 (0x01)			
ActTimeOut	10000	Any value specified by user in ms units			
ActUnitNumber	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station side module station number	Target station side module station number
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_NZ2GF_ETB (0x59)			

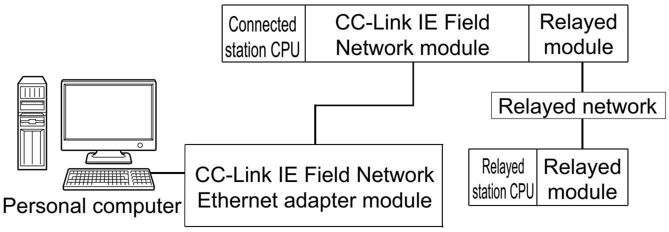
*1 : For the I/O address, specify the value of the actual start I/O number divided by 16.

*2 : Specify the following value for the channel number to be multi-drop linked.
0: Default channel of module
1: Channel 1
2: Channel 2

*3 : When the property pattern is ①, ③, or ④, specify the parameter value set for the CC-Link IE Field Network module on the connected station side for ActNetworkNumber and ActStationNumber.
When the property pattern is ②, specify the value set in the target station side parameter for ActNetworkNumber and ActStationNumber.

4.3.8 Ethernet communication when the connected station is CC-Link IE Field Network Ethernet adapter module (UDP)

(1) Configuration



(2) Property patterns

Connected station CPU	Relayed network	Relayed station CPU					
QnUDE(H) CPU		**QCPU (Q mode)**	**Q12DC CPU-V**	**LCPU**	**QSCPU**	**Q motion CPU**	**FXCPU**
①	CC IE Control CC IE Field	②	②[*1]	②[*2]	×	×	×
	MELSECNET/H	②	②	×	×	×	×
	Ethernet	②	×	×	×	×	×
	Serial communication	③	×	③	×	×	×
	CC-Link	④	④	④	×	×	×

Connected station CPU	Relayed network	Relayed station CPU					
LCPU		**QCPU (Q mode)**	**Q12DC CPU-V**	**LCPU**	**QSCPU**	**Q motion CPU**	**FXCPU**
①	CC IE Control CC IE Field	②	×	②	×	×	×
	MELSECNET/H	×	×	×	×	×	×
	Ethernet	×	×	×	×	×	×
	Serial communication	③	×	③	×	×	×
	CC-Link	④	④	④	×	×	×

○: Accessible (The number in the circle is the property pattern number.), ×: Inaccessible

*1 : Q12DCCPU-V cannot be accessed using CC-Link IE Field Network because CC-Link IE Field Network is not supported.

*2 : LCPU cannot be accessed using CC-Link IE Controller Network because CC-Link IE Controller Network is not supported.

(3) Property list

The following table shows the property settings of communication paths. The setting is not necessary for those properties without description.

Property	Default value	Property pattern			
		①	②	③	④
ActCpuType	34 (CPU_Q02CPU)	CPU type corresponding to the target station			
ActDestination IONumber	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No. 1: 992 (0x3E0) No. 2: 993 (0x3E1) No. 3: 994 (0x3E2) No. 4: 995 (0x3E3) For Redundant CPU Control system: 976 (0x3D0) No specification: 1023 (0x3FF)	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No. 1: 992 (0x3E0) No. 2: 993 (0x3E1) No. 3: 994 (0x3E2) No. 4: 995 (0x3E3) For Redundant CPU Control system: 976 (0x3D0) No specification: 1023 (0x3FF)
ActDidProperty Bit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActDsidProperty Bit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActHostAddress	1.1.1.1	For communication with specified IP address: Host name or IP address of the CC-Link IE Field Network Ethernet adapter module For direct communication without specified IP address: The specification is invalid.			
ActIntelligent PreferenceBit	0 (0x00)	Fixed to 0 (0x00)			
ActIONumber[*1]	1023 (0x3FF)	For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No. 1: 992 (0x3E0) No. 2: 993 (0x3E1) No. 3: 994 (0x3E2) No. 4: 995 (0x3E3)	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No. 1: 992 (0x3E0) No. 2: 993 (0x3E1) No. 3: 994 (0x3E2) No. 4: 995 (0x3E3) For Redundant CPU Control system: 976 (0x3D0) No specification: 1023 (0x3FF)	Connected station side relayed module I/O address	Connected station side relayed module I/O address
ActMultiDrop ChannelNumber[*2]	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Multi-drop channel number	Fixed to 0 (0x00)
ActNetwork Number[*3]	0 (0x00)	Connected station side CC-Link IE Field Network module station number	Target station side module station number	Connected station side CC-Link IE Field Network module station number	Connected station side CC-Link IE Field Network module station number

Property	Default value	Property pattern			
		①	②	③	④
ActProtocolType	0x04 (PROTOCOL_ SERIAL)	PROTOCOL_UDPIP (0x08)			
ActStation Number[*3]	255 (0xFF)	Connected station side CC-Link IE Field Network module station number	Target station side module station number	Connected station side CC-Link IE Field Network module station number	Connected station side CC-Link IE Field Network module station number
ActThrough NetworkType	0 (0x00)	MELSECNET/10 is not included. : 0 (0x00) MELSECNET/10 is included. : 1 (0x01)			
ActTimeOut	10000	Any value specified by user in ms units			
ActUnitNumber	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station side module station number	Target station side module station number
ActUnitType	0x13 (UNIT_QNCPU)	For communication with specified IP address: UNIT_NZ2GF_ETB (0x59) For direct communication without specified IP address: UNIT_NZ2GF_ETB_DIRECT (0x5A)			

*1 : For the I/O address, specify the value of the actual start I/O number divided by 16.

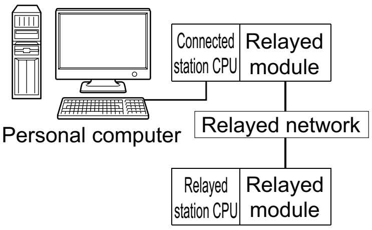
*2 : Specify the following value for the channel number to be multi-drop linked.
0: Default channel of module
1: Channel 1
2: Channel 2

*3 : When the property pattern is ①, ③, or ④, specify the parameter value set for the CC-Link IE Field Network module on the connected station side for ActNetworkNumber and ActStationNumber.
When the property pattern is ②, specify the value set in the target station side parameter for ActNetworkNumber and ActStationNumber.

4.4 CPU COM Communication

4.4.1 CPU COM communication when the connected station is QCPU (Q mode)

(1) Configuration



(2) Property patterns

Connected station CPU	Relayed network	Relayed station CPU					
QCPU (Q mode)		QCPU (Q mode)	Q12DC CPU-V	LCPU	QSCPU	Q motion CPU	FXCPU
①	CC IE Control CC IE Field	②	②*1	②*2	②*1	②*1	×
	MELSECNET/H	②	②	×	②	②	×
	Ethernet	②	×	×	②	②	×
	Serial communication	③*3	×	③	×	③	×
	CC-Link	④	④	④	×	④	④*4

○: Accessible (The number in the circle is the property pattern number.), ×: Inaccessible

*1 : Q12DCCPU-V, QSCPU, and Q motion CPU cannot be accessed using CC-Link IE Field Network because CC-Link IE Field Network is not supported.

*2 : LCPU cannot be accessed using CC-Link IE Controller Network because CC-Link IE Controller Network is not supported.

*3 : For Redundant CPU, serial communication modules on the main base cannot be accessed.

*4 : Supported by FX$_{3G}$ and FX$_{3U(C)}$ within the own network only.

(3) Property list

The following table shows the property settings of communication paths. The setting is not necessary for those properties without description.

Property	Default value	Property pattern			
		①	②*1	③	④
ActBaudRate	19200 (BAUDRATE_ 19200)	BAUDRATE_9600, BAUDRATE_19200, BAUDRATE_38400, BAUDRATE_57600, BAUDRATE_115200			
ActControl	8 (TRC_DTR_OR_ RTS)	Depending on the used cable.			
ActCpuType	34 (CPU_Q02CPU)	CPU type corresponding to the target station			
ActDestination IONumber	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No. 1: 992 (0x3E0) No. 2: 993 (0x3E1) No. 3: 994 (0x3E2) No. 4: 995 (0x3E3) For Redundant CPU Control system: 976 (0x3D0) No specification: 1023 (0x3FF)	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No. 1: 992 (0x3E0) No. 2: 993 (0x3E1) No. 3: 994 (0x3E2) No. 4: 995 (0x3E3) For Redundant CPU Control system: 976 (0x3D0) No specification: 1023 (0x3FF)
ActDidProperty Bit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActDsidProperty Bit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActIntelligent PreferenceBit	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station Q mode/ Q12DCCPU-V/ LCPU :1 (0x01) Other than the above :0 (0x00)	Target station Q mode/ Q12DCCPU-V/ LCPU :1 (0x01) Other than the above :0 (0x00)
ActIONumber*2	1023 (0x3FF)	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No. 1: 992 (0x3E0) No. 2: 993 (0x3E1) No. 3: 994 (0x3E2) No. 4: 995 (0x3E3) For Redundant CPU Control system: 976 (0x3D0) No specification: 1023 (0x3FF)	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No. 1: 992 (0x3E0) No. 2: 993 (0x3E1) No. 3: 994 (0x3E2) No. 4: 995 (0x3E3) For Redundant CPU Control system: 976 (0x3D0) No specification: 1023 (0x3FF)	Connected station side module I/O address	Connected station side module I/O address

Property	Default value	Property pattern			
		①	②[*1]	③	④
ActMultiDrop ChannelNumber[*3]	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Multi-drop channel number	Fixed to 0 (0x00)
ActNetwork Number	0 (0x00)	Fixed to 0 (0x00)	Target station side module network number	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActPortNumber	1 (PORT_1)	Personal computer side COM port number			
ActProtocolType	0x04 (PROTOCOL_ SERIAL)	PROTOCOL_SERIAL (0x04)			
ActStation Number	255 (0xFF)	Fixed to 255 (0xFF)	Target station side module station number	Fixed to 255 (0xFF)	Fixed to 255 (0xFF)
ActThrough NetworkType	0 (0x00)	MELSECNET/10 is not included. : 0 (0x00) MELSECNET/10 is included. : 1 (0x01)			
ActTimeOut	10000	Any value specified by user in ms units.			
ActUnitNumber	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station side module station number	Target station side module station number
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_QNCPU (0x13)			

*1 : Note the following considerations when accessing via Ethernet module (Q series-compatible E71).

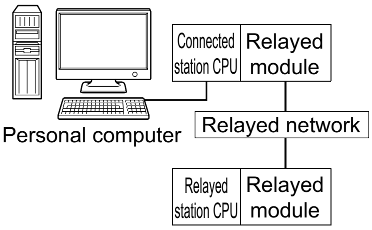
- For ActNetworkNumber and ActStationNumber, specify the value set in the parameter setting of the target station side Q series-compatible E71.
- Set "Station No. ⇔ IP information" in the parameter setting of Q series-compatible E71. When setting the parameter, specify the setting other than the automatic response system (any of the IP address calculation system, table conversion system, and combined system) for "Station No. ⇔ IP information system".

*2 : For the I/O address, specify the value of the actual start I/O number divided by 16.

*3 : Specify the following value for the channel number to be multi-drop linked.
0: Default channel of module
1: Channel 1
2: Channel 2

4.4.2 CPU COM communication when the connected station is LCPU

(1) Configuration



(2) Property patterns

Connected station CPU	Relayed network	Relayed station CPU					
LCPU		QCPU (Q mode)	Q12DC CPU-V	LCPU	QSCPU	Q motion CPU	FXCPU
①	CC IE Field[*1]	④	×	④	×	×	×
	MELSECNET/H	×	×	×	×	×	×
	Ethernet	×	×	×	×	×	×
	Serial communication	②[*2]	×	②	×	×	×
	CC-Link	③	③	③	×	×	×

○: Accessible (The number in the circle is the property pattern number.), ×: Inaccessible

*1 : LCPU cannot be accessed using CC-Link IE Controller Network because CC-Link IE Controller Network is not supported.

*2 : For Redundant CPU, serial communication modules on the main base cannot be accessed.

(3) Property list

The following table shows the property settings of communication paths. The setting is not necessary for those properties without description.

Property	Default value	Property pattern			
		①	②	③	④
ActBaudRate	19200 (BAUDRATE_ 19200)	BAUDRATE_9600, BAUDRATE_19200, BAUDRATE_38400, BAUDRATE_57600, BAUDRATE_115200			
ActControl	8 (TRC_DTR_OR_ RTS)	Depending on the used cable.			
ActCpuType	34 (CPU_Q02CPU)	CPU type corresponding to the target station			

Property	Default value	Property pattern			
		①	②	③	④
ActDestination IONumber	0 (0x00)	Fixed to 0 (0x00)	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No. 1: 992 (0x3E0) No. 2: 993 (0x3E1) No. 3: 994 (0x3E2) No. 4: 995 (0x3E3) For Redundant CPU Control system: 976 (0x3D0) No specification: 1023 (0x3FF)	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No. 1: 992 (0x3E0) No. 2: 993 (0x3E1) No. 3: 994 (0x3E2) No. 4: 995 (0x3E3) For Redundant CPU Control system: 976 (0x3D0) No specification: 1023 (0x3FF)	Fixed to 0 (0x00)
ActDidPropertyBit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Fixed to 1 (0x01)
ActDsidPropertyBit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Fixed to 1 (0x01)
ActIntelligent PreferenceBit	0 (0x00)	Fixed to 0 (0x00)	Target station Q mode/ Q12DCCPU-V/ LCPU :1 (0x01) Other than the above :0 (0x00)	Target station Q mode/ Q12DCCPU-V/ LCPU :1 (0x01) Other than the above :0 (0x00)	Fixed to 0 (0x00)
ActIONumber[*1]	1023 (0x3FF)	Fixed to 1023 (0x3FF)	Connected station side module I/O address	Connected station side module I/O address	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No. 1: 992 (0x3E0) No. 2: 993 (0x3E1) No. 3: 994 (0x3E2) No. 4: 995 (0x3E3) For Redundant CPU Control system: 976 (0x3D0) No specification: 1023 (0x3FF)
ActMultiDrop ChannelNumber[*2]	0 (0x00)	Fixed to 0 (0x00)			
ActNetworkNumber	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station side module network number
ActPortNumber	1 (PORT_1)	Personal computer side COM port number			
ActProtocolType	0x04 (PROTOCOL_ SERIAL)	PROTOCOL_SERIAL (0x04)			
ActStationNumber	255 (0xFF)	Fixed to 255 (0xFF)	Fixed to 255 (0xFF)	Fixed to 255 (0xFF)	Target station side module station number
ActThroughNetwork Type	0 (0x00)	MELSECNET/10 is not included. : 0 (0x00) MELSECNET/10 is included. : 1 (0x01)			
ActTimeOut	10000	Any value specified by user in ms units.			

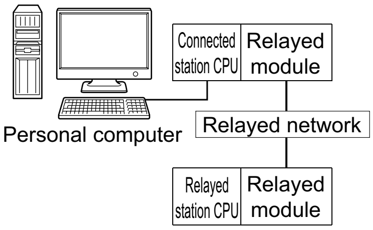
Property	Default value	Property pattern			
		①	②	③	④
ActUnitNumber	0 (0x00)	Fixed to 0 (0x00)	Target station side module station number	Target station side module station number	Fixed to 0 (0x00)
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_LNCPU (0x50)			

*1 : For the I/O address, specify the value of the actual start I/O number divided by 16.

*2 : Specify the following value for the channel number to be multi-drop linked.
0: Default channel of module
1: Channel 1
2: Channel 2

4.4.3 CPU COM communication when the connected station is Q motion CPU

(1) Configuration



(2) Property patterns

Connected station CPU	Relayed network	Relayed station CPU					
Q motion CPU		QCPU (Q mode)	Q12DC CPU-V	LCPU	QSCPU	Q motion CPU	FXCPU
①[*1]	CC IE Control CC IE Field	×	×	×	×	×	×
	MELSECNET/H	×	×	×	×	×	×
	Ethernet	×	×	×	×	×	×
	Serial communication	×	×	×	×	×	×
	CC-Link	×	×	×	×	×	×

○: Accessible (The number in the circle is the property pattern number.), ×: Inaccessible

[*1] : Relayed stations cannot be accessed when the Q motion CPU is Q172CPU, Q173CPU, Q172HCPU, or Q173HCPU.

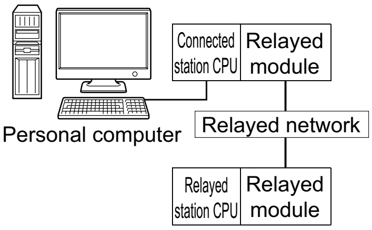
(3) Property list

The following table shows the property settings of communication paths. The setting is not necessary for those properties without description.

Property	Default value	Property pattern
		①
ActBaudRate	19200 (BAUDRATE_19200)	BAUDRATE_9600, BAUDRATE_19200, BAUDRATE_38400, BAUDRATE_57600, BAUDRATE_115200
ActControl	8 (TRC_DTR_OR_RTS)	Depending on the used cable.
ActCpuType	34 (CPU_Q02CPU)	CPU type corresponding to the target station
ActIONumber	1023 (0x3FF)	For multiple CPUs Control CPU: 1023 (0x3FF) No. 2: 993 (0x3E1) No. 3: 994 (0x3E2) No. 4: 995 (0x3E3)
ActPortNumber	1 (PORT_1)	Personal computer side COM port number
ActProtocolType	0x04 (PROTOCOL_SERIAL)	PROTOCOL_SERIAL (0x04)
ActTimeOut	10000	Any value specified by user in ms units.
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_QNMOTION (0x1C)

4.4.4 CPU COM communication when the connected station is FXCPU

(1) Configuration



(2) Property patterns

Connected station CPU	Relayed network	Relayed station CPU					
FXCPU		QCPU (Q mode)	Q12DC CPU-V	LCPU	QSCPU	Q motion CPU	FXCPU
①	CC IE Control CC IE Field	×	×	×	×	×	×
	MELSECNET/H	×	×	×	×	×	×
	Ethernet	×	×	×	×	×	×
	Serial communication	×	×	×	×	×	×
	CC-Link	×	×	×	×	×	×

○: Accessible (The number in the circle is the property pattern number.), ×: Inaccessible

(3) Property list

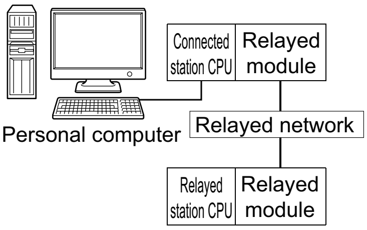
The following table shows the property settings of communication paths. The setting is not necessary for those properties without description.

Property	Default value	Property pattern	
		①	
ActBaudRate	19200 (BAUDRATE_19200)	FX$_{0(S)}$, FX$_{0N}$, FX$_{1}$, FX$_{1S}$, FX$_{2(C)}$	Fixed to BAUDRATE_9600
		FX$_{1N(C)}$, FX$_{2N(C)}$	BAUDRATE_9600, BAUDRATE_19200
		FX$_{3UC}$, FX$_{3G}$	BAUDRATE_9600, BAUDRATE_19200, BAUDRATE_38400, BAUDRATE_57600, BAUDRATE_115200
ActControl	8 (TRC_DTR_OR_RTS)	Depending on the used cable.	
ActCpuType	34 (CPU_Q02CPU)	CPU type corresponding to the target station	
ActPortNumber	1 (PORT_1)	Personal computer side COM port number	
ActProtocolType	0x04 (PROTOCOL_SERIAL)	PROTOCOL_SERIAL(0x04)	
ActTimeOut	10000	Any value specified by user in ms units.	
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_FXCPU(0x0F)	

4.5 USB Communication

4.5.1 USB communication when the connected station is QCPU (Q mode)

(1) Configuration



(2) Property patterns

Connected station CPU	Relayed network	Relayed station CPU					
QCPU (Q mode)		QCPU (Q mode)	Q12DC CPU-V	LCPU	QSCPU	Q motion CPU	FXCPU
①	CC IE Control CC IE Field	②	②[*1]	②[*2]	②[*1]	②[*1]	×
	MELSECNET/H	②	②	×	②	②	×
	Ethernet	②	×	×	②	②	×
	Serial communication	③[*3]	×	③	×	③	×
	CC-Link	④	④	④	×	④	④[*4]

Connected station CPU	Relayed network	Relayed station CPU					
Q12DCCPU-V		QCPU (Q mode)	Q12DC CPU-V	LCPU	QSCPU	Q motion CPU	FXCPU
①	CC IE Control CC IE Field	②	②[*1]	②[*2]	②[*1]	②[*1]	×
	MELSECNET/H	②	②	×	②	②	×
	Ethernet	×	×	×	×	×	×
	Serial communication	×	×	×	×	×	×
	CC-Link	④	④	④	×	④	×

○: Accessible (The number in the circle is the property pattern number.), ×: Inaccessible

[*1] Q12DCCPU-V, QSCPU, and Q motion CPU cannot be accessed using CC-Link IE Field Network because CC-Link IE Field Network is not supported.

[*2] LCPU cannot be accessed using CC-Link IE Controller Network because CC-Link IE Controller Network is not supported.

[*3] For Redundant CPU, serial communication modules on the main base cannot be accessed.

[*4] Supported by FX$_{3G}$ and FX$_{3U(C)}$ within the own network only.

(3) Property list

The following table shows the property settings of communication paths. The setting is not necessary for those properties without description.

Property	Default value	Property pattern			
		①	②*1	③	④
ActCpuType	34 (CPU_Q02CPU)	CPU type corresponding to the target station			
ActDestination IONumber	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No. 1: 992 (0x3E0) No. 2: 993 (0x3E1) No. 3: 994 (0x3E2) No. 4: 995 (0x3E3) For Redundant CPU Control system: 976 (0x3D0) No specification: 1023 (0x3FF)	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No. 1: 992 (0x3E0) No. 2: 993 (0x3E1) No. 3: 994 (0x3E2) No. 4: 995 (0x3E3) For Redundant CPU Control system: 976 (0x3D0) No specification: 1023 (0x3FF)
ActDidProperty Bit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActDsidProperty Bit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActIntelligent PreferenceBit	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station Q mode/ Q12DCCPU-V/ LCPU :1 (0x01) Other than the above :0 (0x00)	Target station Q mode/ Q12DCCPU-V/ LCPU :1 (0x01) Other than the above :0 (0x00)
ActIONumber*2	1023 (0x3FF)	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No. 1: 992 (0x3E0) No. 2: 993 (0x3E1) No. 3: 994 (0x3E2) No. 4: 995 (0x3E3) For Redundant CPU Control system: 976 (0x3D0) No specification: 1023 (0x3FF)	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No. 1: 992 (0x3E0) No. 2: 993 (0x3E1) No. 3: 994 (0x3E2) No. 4: 995 (0x3E3) For Redundant CPU Control system: 976 (0x3D0) No specification: 1023 (0x3FF)	Connected station side module I/O address	Connected station side module I/O address
ActMultiDrop ChannelNumber*3	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Multi-drop channel number	Fixed to 0 (0x00)
ActNetwork Number	0 (0x00)	Fixed to 0 (0x00)	Target station side module network number	Fixed to 0 (0x00)	Fixed to 0 (0x00)

Property	Default value	Property pattern			
		①	②*1	③	④
ActProtocolType	0x04 (PROTOCOL_ SERIAL)	PROTOCOL_USB (0x0D)			
ActStation Number	255 (0xFF)	Fixed to 255 (0xFF)	Target station side module station number	Fixed to 255 (0xFF)	Fixed to 255 (0xFF)
ActThrough NetworkType	0 (0x00)	MELSECNET/10 is not included. : 0 (0x00) MELSECNET/10 is included. : 1 (0x01)			
ActTimeOut	10000	Any value specified by user in ms units			
ActUnitNumber	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station side module station number	Target station side module station number
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_QNUSB (0x16)			

*1 : Note the following considerations when accessing via Ethernet module (Q series-compatible E71).

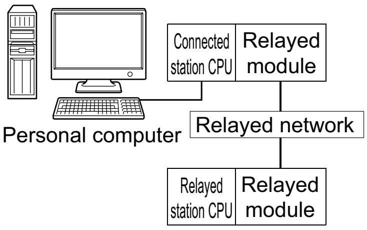
- For ActNetworkNumber and ActStationNumber, specify the value set in the parameter setting of the target station side Q series-compatible E71.
- Set "Station No. ⇔ IP information" in the parameter setting of Q series-compatible E71. When setting the parameter, specify the setting other than the automatic response system (any of the IP address calculation system, table conversion system, and combined system) for "Station No. ⇔ IP information system".

*2 : For the I/O address, specify the value of the actual start I/O number divided by 16.

*3 : Specify the following value for the channel number to be multi-drop linked.
0: Default channel of module
1: Channel 1
2: Channel 2

4.5.2 USB communication when the connected station is LCPU

(1) Configuration



(2) Property patterns

Connected station CPU	Relayed network	Relayed station CPU					
LCPU		QCPU (Q mode)	Q12DC CPU-V	LCPU	QSCPU	Q motion CPU	FXCPU
①	CC IE Field[*1]	④	×	④	×	×	×
	MELSECNET/H	×	×	×	×	×	×
	Ethernet	×	×	×	×	×	×
	Serial communication	②[*2]	×	②	×	×	×
	CC-Link	③	③	③	×	×	×

○: Accessible (The number in the circle is the property pattern number.), ×: Inaccessible

[*1] : LCPU cannot be accessed using CC-Link IE Controller Network because CC-Link IE Controller Network is not supported.

[*2] : For Redundant CPU, serial communication modules on the main base cannot be accessed.

(3) Property list

The following table shows the property settings of communication paths. The setting is not necessary for those properties without description.

Property	Default value	Property pattern			
		①	②	③	④
ActCpuType	34 (CPU_Q02CPU)	CPU type corresponding to the target station			
ActDestination IONumber	0 (0x00)	Fixed to 0 (0x00)	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No. 1: 992 (0x3E0) No. 2: 993 (0x3E1) No. 3: 994 (0x3E2) No. 4: 995 (0x3E3) For Redundant CPU Control system: 976 (0x3D0) No specification: 1023 (0x3FF)	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No. 1: 992 (0x3E0) No. 2: 993 (0x3E1) No. 3: 994 (0x3E2) No. 4: 995 (0x3E3) For Redundant CPU Control system: 976 (0x3D0) No specification: 1023 (0x3FF)	Fixed to 0 (0x00)
ActDidPropertyBit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Fixed to 1 (0x01)
ActDsidPropertyBit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Fixed to 1 (0x01)

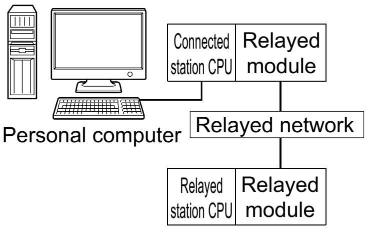
Property	Default value	Property pattern			
		①	②	③	④
ActIntelligent PreferenceBit	0 (0x00)	Fixed to 0 (0x00)	Target station Q mode/ Q12DCCPU-V/ LCPU :1 (0x01) Other than the above :0 (0x00)	Target station Q mode/ Q12DCCPU-V/ LCPU :1 (0x01) Other than the above :0 (0x00)	Fixed to 0 (0x00)
ActIONumber[*1]	1023 (0x3FF)	Fixed to 1023 (0x3FF)	Connected station side module I/O address	Connected station side module I/O address	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No. 1: 992 (0x3E0) No. 2: 993 (0x3E1) No. 3: 994 (0x3E2) No. 4: 995 (0x3E3) For Redundant CPU Control system: 976 (0x3D0) No specification: 1023 (0x3FF)
ActMultiDrop ChannelNumber[*2]	0 (0x00)	Fixed to 0 (0x00)	Multi-drop channel number	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActNetworkNumber	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station side module network number
ActProtocolType	0x04 (PROTOCOL_ SERIAL)	PROTOCOL_USB (0x0D)			
ActStationNumber	255 (0xFF)	Fixed to 255 (0xFF)	Fixed to 255 (0xFF)	Fixed to 255 (0xFF)	Target station side module station number
ActThroughNetwork Type	0 (0x00)	MELSECNET/10 is not included. : 0 (0x00) MELSECNET/10 is included. : 1 (0x01)			
ActTimeOut	10000	Any value specified by user in ms units			
ActUnitNumber	0 (0x00)	Fixed to 0 (0x00)	Target station side module station number	Target station side module station number	Fixed to 0 (0x00)
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_LNUSB (0x51)			

[*1] For the I/O address, specify the value of the actual start I/O number divided by 16.

[*2] Specify the following value for the channel number to be multi-drop linked.
0: Default channel of module
1: Channel 1
2: Channel 2

4.5.3 USB communication when the connected station is QSCPU

(1) Configuration



(2) Property patterns

Connected station CPU	Relayed network	Relayed station CPU					
QSCPU		QCPU (Q mode)	Q12DC CPU-V	LCPU	QSCPU	Q motion CPU	FXCPU
⑤*1	CC IE Control CC IE Field	×	×	×	×	×	×
	MELSECNET/H	×	×	×	×	×	×
	Ethernet	×	×	×	×	×	×
	Serial communication	×	×	×	×	×	×
	CC-Link	×	×	×	×	×	×

○: Accessible (The number in the circle is the property pattern number.), ×: Inaccessible

*1 : Relayed stations cannot be accessed via QSCPU.

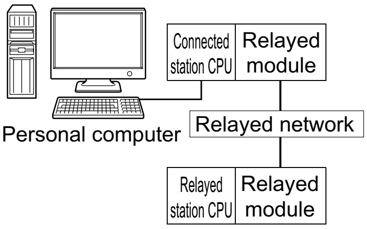
(3) Property list

The following table shows the property settings of communication paths. The setting is not necessary for those properties without description.

Property	Default value	Property pattern
		①
ActCpuType	34 (CPU_Q02CPU)	CPU type corresponding to the target station
ActProtocolType	0x04 (PROTOCOL_SERIAL)	PROTOCOL_USB (0x0D)
ActTimeOut	10000	Any value specified by user in ms units
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_QSUSB (0x29)

4.5.4 USB communication when the connected station is Q motion CPU

(1) Configuration



(2) Property patterns

Connected station CPU	Relayed network	Relayed station CPU					
Q motion CPU		QCPU (Q mode)	Q12DC CPU-V	LCPU	QSCPU	Q motion CPU	FXCPU
①[*1]	CC IE Control CC IE Field	×	×	×	×	×	×
	MELSECNET/H	×	×	×	×	×	×
	Ethernet	×	×	×	×	×	×
	Serial communication	×	×	×	×	×	×
	CC-Link	×	×	×	×	×	×

○: Accessible (The number in the circle is the property pattern number.), ×: Inaccessible

[*1] : Relayed stations cannot be accessed when the Q motion CPU is Q172CPU, Q173CPU, Q172HCPU, or Q173HCPU.

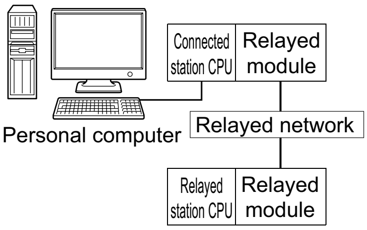
(3) Property list

The following table shows the property settings of communication paths. The setting is not necessary for those properties without description.

Property	Default value	Property pattern
		①
ActCpuType	34 (CPU_Q02CPU)	CPU type corresponding to the target station
ActIONumber	1023 (0x3FF)	For multiple CPUs Control CPU: F1023 (0x3FF) No. 2: 993 (0x3E1) No. 3: 994 (0x3E2) No. 4: 995 (0x3E3)
ActProtocolType	0x04 (PROTOCOL_SERIAL)	PROTOCOL_USB (0x0D)
ActTimeOut	10000	Any value specified by user in ms units
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_QNMOTIONUSB (0x1D)

4.5.5 USB communication when the connected station is FXCPU

(1) Configuration



(2) Property patterns

Connected station CPU	Relayed network	Relayed station CPU					
FXCPU		QCPU (Q mode)	Q12DC CPU-V	LCPU	QSCPU	Q motion CPU	FXCPU
①	CC IE Control CC IE Field	×	×	×	×	×	×
	MELSECNET/H	×	×	×	×	×	×
	Ethernet	×	×	×	×	×	×
	Serial communication	×	×	×	×	×	×
	CC-Link	×	×	×	×	×	×

○: Accessible (The number in the circle is the property pattern number.), ×: Inaccessible

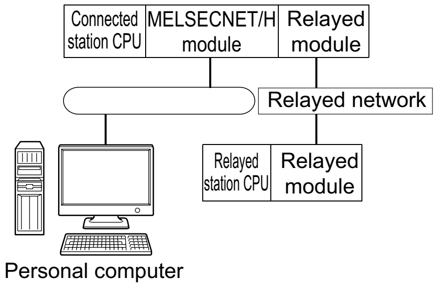
(3) Property list

The following table shows the property settings of communication paths. The setting is not necessary for those properties without description.

Property	Default value	Property pattern
		①
ActCpuType	34 (CPU_Q02CPU)	CPU type corresponding to the target station
ActProtocolType	0x04 (PROTOCOL_SERIAL)	PROTOCOL_USB (0x0D)
ActTimeOut	10000	Any value specified by user in ms units
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_FXCPU (0x0F)

4.6 MELSECNET/H Communication

(1) Configuration



(2) Property patterns

Own Board	Connected station CPU		Relayed network	Relayed station CPU					
	QCPU (Q mode)	QSCPU Q motion CPU		QCPU (Q mode)	Q12DC CPU-V	LCPU	QSCPU	Q motion CPU	FXCPU
①	②	②[*1]	CC IE Control CC IE Field	②	②[*2]	②[*3]	②[*1,*2]	×	×
			MELSECNET/H	②	②	×	②[*1]	×	×
			Ethernet	②	×	×	②[*1]	×	×
			Serial communication module	③[*4]	×	③	×	③	×
			CC-Link	④	④	④	×	④	×

Own Board	Connected station CPU	Relayed network	Relayed station CPU					
	Q12DCCPU-V		QCPU (Q mode)	Q12DC CPU-V	LCPU	QSCPU	Q motion CPU	FXCPU
①	②	CC IE Control CC IE Field	×	×	×	×	×	×
		MELSECNET/H	×	×	×	×	×	×
		Ethernet	×	×	×	×	×	×
		Serial communication module	×	×	×	×	×	×
		CC-Link	④	④	④	×	④	×

○: Accessible (The number in the circle is the property pattern number.), ×: Inaccessible

*1 : Relayed stations cannot be accessed via QCPU and Q motion CPU.

*2 : Q12DCCPU-V and QSCPU cannot be accessed using CC-Link IE Field Network because CC-Link IE Field Network is not supported.

*3 : LCPU cannot be accessed using CC-Link IE Controller Network because CC-Link IE Controller Network is not supported.

*4 : For Redundant CPU, serial communication modules on the main base cannot be accessed.

(3) Property list

The following table shows the property settings of communication paths. The setting is not necessary for those properties without description.

Property	Default value	Property pattern			
		①	②	③	④
ActCpuType	34 (CPU_Q02CPU)	CPU type corresponding to the target station			
ActDestination IONumber	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No.1: 992 (0x3E0) No.2: 993 (0x3E1) No.3: 994 (0x3E2) No.4: 995 (0x3E3) For Redundant CPU Control system: 976 (0x3D0) No specification: 1023 (0x3FF)	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No.1: 992 (0x3E0) No.2: 993 (0x3E1) No.3: 994 (0x3E2) No.4: 995 (0x3E3) For Redundant CPU Control system: 976 (0x3D0) No specification: 1023 (0x3FF)
ActDidPropertyBit	1 (0x01)	Fixed to 0 (0x00)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActDsidPropertyBit	1 (0x01)	Fixed to 0 (0x00)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActIONumber[*1]	1023 (0x3FF)	Fixed to 0 (0x00)	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No.1: 992 (0x3E0) No.2: 993 (0x3E1) No.3: 994 (0x3E2) No.4: 995 (0x3E3) For Redundant CPU Control system: 976 (0x3D0) No specification: 1023 (0x3FF)	Connected station side relayed module I/O address	Connected station side relayed module I/O address
ActMultiDrop ChannelNumber[*2]	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Multi-drop channel number	Fixed to 0 (0x00)
ActNetworkNumber	0 (0x00)	Fixed to 0 (0x00)	Target station side module network number	Connected station side module network number	Connected station side module network number
ActPortNumber	1 (PORT_1)	Port number of personal computer side MELSECNET/H board, PORT 1 to PORT 4 (first to fourth boards)			
ActProtocolType	0x04 (PROTOCOL_ SERIAL)	PROTOCOL_MNETH (0x0F)			
ActStationNumber	255 (0xFF)	Fixed to 255 (0xFF)	Target station side module station number	Connected station side module station number	Connected station side module station number

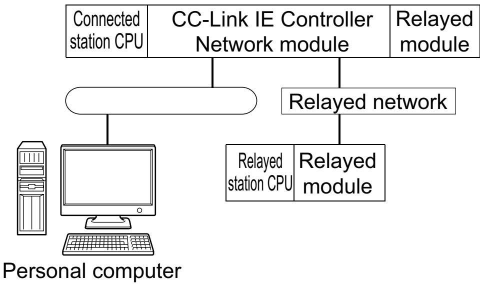
Property	Default value	Property pattern			
		①	②	③	④
ActThroughNetworkType	0 (0x00)	MELSECNET/10 is not included. : 0 (0x00) MELSECNET/10 is included. : 1 (0x01)			
ActUnitNumber	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station side module station number	Target station side module station number
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_MNETHBOARD (0x1E)			

*1 : For the I/O address, specify the value of the actual start I/O number divided by 16.

*2 : Specify the following value for the channel number to be multi-drop linked.
0: Default channel of module
1: Channel 1
2: Channel 2

4.7 CC-Link IE Controller Network Communication

(1) Configuration



(2) Property patterns

Own board	Connected station CPU		Relayed network	Relayed station CPU					
	QCPU (Q mode)	QSCPU Q motion CPU		QCPU (Q mode)	Q12DC CPU-V	LCPU	QSCPU	Q motion CPU	FXCPU
①	②	②*1	CC IE Control CC IE Field	②	②*2	②*3	②*1,*2	×	×
			MELSECNET/H	②	②	×	②*1	×	×
			Ethernet	②	×	×	②*1	×	×
			Serial communication module	③*4	×	③	×	③	×
			CC-Link	④	④	④	×	④	×

Own board	Connected station CPU	Relayed network	Relayed station CPU					
	Q12DCCPU-V		QCPU (Q mode)	Q12DC CPU-V	LCPU	QSCPU	Q motion CPU	FXCPU
①	②	CC IE Control CC IE Field	×	×	×	×	×	×
		MELSECNET/H	×	×	×	×	×	×
		Ethernet	×	×	×	×	×	×
		Serial communication module	×	×	×	×	×	×
		CC-Link	④	④	④	×	④	×

○: Accessible (The number in the circle is the property pattern number.), ×: Inaccessible

*1 : Relayed stations cannot be accessed via QCPU and Q motion CPU.

*2 : Q12DCCPU-V and QSCPU cannot be accessed using CC-Link IE Field Network because CC-Link IE Field Network is not supported.

*3 : LCPU cannot be accessed using CC-Link IE Controller Network because CC-Link IE Controller Network is not supported.

*4 : For Redundant CPU, serial communication modules on the main base cannot be accessed.

(3) Property list

The following table shows the property settings of communication paths. The setting is not necessary for those properties without description.

Property	Default value	Property pattern			
		①	②	③	④
ActCpuType	34 (CPU_Q02CPU)	CPU type corresponding to the target station			
ActDestination IONumber	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No.1: 992 (0x3E0) No.2: 993 (0x3E1) No.3: 994 (0x3E2) No.4: 995 (0x3E3) For Redundant CPU Control system: 976 (0x3D0) No specification: 1023 (0x3FF)	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No.1: 992 (0x3E0) No.2: 993 (0x3E1) No.3: 994 (0x3E2) No.4: 995 (0x3E3) For Redundant CPU Control system: 976 (0x3D0) No specification: 1023 (0x3FF)
ActDidPropertyBit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActDsidPropertyBit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActIONumber*1	1023 (0x3FF)	1023 (0x3FF)	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No.1: 992 (0x3E0) No.2: 993 (0x3E1) No.3: 994 (0x3E2) No.4: 995 (0x3E3) For Redundant CPU Control system: 976 (0x3D0) No specification: 1023 (0x3FF)	Connected station side relayed module I/O address	Connected station side relayed module I/O address
ActMultiDrop ChannelNumber*2	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Multi-drop channel number	Fixed to 0 (0x00)
ActNetworkNumber	0 (0x00)	Fixed to 0 (0x00)	Target station side module network number	Connected station side module network number	Connected station side module network number
ActPortNumber	1 (PORT_1)	Port number of personal computer side CC-Link IE Controller Network board, PORT 1 to PORT 4 (first to fourth boards)			
ActProtocolType	0x04 (PROTOCOL_ SERIAL)	PROTOCOL_MNETG (0x14)			
ActStationNumber	255 (0xFF)	Fixed to 255 (0xFF)	Target station side module station number	Connected station side module station number	Connected station side module station number

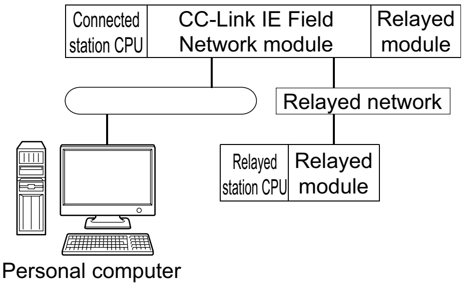
Property	Default value	Property pattern			
		①	②	③	④
ActThroughNetwork Type	0 (0x00)	MELSECNET/10 is not included. : 0 (0x00) MELSECNET/10 is included. : 1 (0x01)			
ActUnitNumber	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station side module station number	Target station side module station number
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_MNETGBOARD (0x2B)			

*1 : For the I/O address, specify the value of the actual start I/O number divided by 16.

*2 : Specify the following value for the channel number to be multi-drop linked.
0: Default channel of module
1: Channel 1
2: Channel 2

4.8 CC-Link IE Field Network Communication

(1) Configuration



(2) Property patterns

Own board	Connected station CPU	Relayed network	Relayed station CPU					
	QCPU (Q mode)		QCPU (Q mode)	Q12DC CPU-V	LCPU	QSCPU	Q motion CPU	FXCPU
①	②	CC IE Control CC IE Field	②	②[*1]	②[*2]	×	×	×
		MELSECNET/H	②	②	×	×	×	×
		Ethernet	②	×	×	×	×	×
		Serial communication module	③	×	③	×	×	×
		CC-Link	④	④	④	×	×	×

Own board	Connected station CPU	Relayed network	Relayed station CPU					
	LCPU		QCPU (Q mode)	Q12DC CPU-V	LCPU	QSCPU	Q motion CPU	FXCPU
①	②	CC IE Field [*2]	②	×	②	×	×	×
		MELSECNET/H	×	×	×	×	×	×
		Ethernet	×	×	×	×	×	×
		Serial communication module	③	×	③	×	×	×
		CC-Link	④	④	④	×	×	×

○: Accessible (The number in the circle is the property pattern number.), ×: Inaccessible

*1 : Q12DCCPU-V and QSCPU cannot be accessed using CC-Link IE Field Network because CC-Link IE Field Network is not supported.

*2 : LCPU cannot be accessed using CC-Link IE Controller Network because CC-Link IE Controller Network is not supported.

(3) Property list

The following table shows the property settings of communication paths. The setting is not necessary for those properties without description.

Property	Default value	Property pattern			
		①	②	③	④
ActCpuType	34 (CPU_Q02CPU)	CPU type corresponding to the target station			
ActDestination IONumber	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No.1: 992 (0x3E0) No.2: 993 (0x3E1) No.3: 994 (0x3E2) No.4: 995 (0x3E3) For Redundant CPU Control system: 976 (0x3D0) No specification: 1023 (0x3FF)	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No.1: 992 (0x3E0) No.2: 993 (0x3E1) No.3: 994 (0x3E2) No.4: 995 (0x3E3) For Redundant CPU Control system: 976 (0x3D0) No specification: 1023 (0x3FF)
ActDidPropertyBit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActDsidPropertyBit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActIONumber[*1]	1023 (0x3FF)	Fixed to 1023 (0x3FF)	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No.1: 992 (0x3E0) No.2: 993 (0x3E1) No.3: 994 (0x3E2) No.4: 995 (0x3E3) For Redundant CPU Control system: 976 (0x3D0) No specification: 1023 (0x3FF)	Connected station side relayed module I/O address	Connected station side relayed module I/O address
ActMultiDrop ChannelNumber[*2]	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Multi-drop channel number	Fixed to 0 (0x00)
ActNetworkNumber	0 (0x00)	Fixed to 0 (0x00)	Target station side module network number	Connected station side module network number	Connected station side module network number
ActPortNumber	1 (PORT_1)	Port number of personal computer side CC-Link IE Field Network board, PORT 1 to PORT 4 (first to fourth boards)			
ActProtocolType	0x04 (PROTOCOL_ SERIAL)	PROTOCOL_CCIEF (0x15)			
ActStationNumber	255 (0xFF)	Fixed to 255 (0xFF)	Target station side module station number	Connected station side module station number	Connected station side module station number

Property	Default value	Property pattern			
		①	②	③	④
ActThrough NetworkType	0 (0x00)	MELSECNET/10 is not included. : 0 (0x00) MELSECNET/10 is included. : 1 (0x01)			
ActUnitNumber	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station side module station number	Target station side module station number
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_CCIEFBOARD (0x2F)			

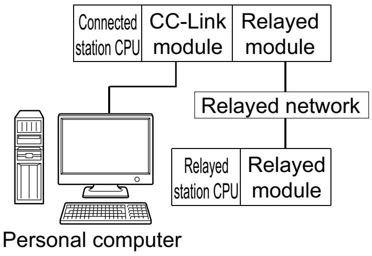
*1 : For the I/O address, specify the value of the actual start I/O number divided by 16.

*2 : Specify the following value for the channel number to be multi-drop linked.

0: Default channel of module
1: Channel 1
2: Channel 2

4.9 CC-Link Communication

(1) Configuration



(2) Property patterns

Own board	Connected station CPU		Relayed network	Relayed station CPU					
	QCPU (Q mode)	Q motion CPU		QCPU (Q mode)	Q12DC CPU-V	LCPU	QSCPU	Q motion CPU	FXCPU
①	②	②[*1]	CC IE Control CC IE Field	③	③[*2]	③[*3]	③[*2]	③[*2]	×
			MELSECNET/H	③	③	×	③	③	×
			Ethernet	③	×	×	③	③	×
			Serial communication module	×	×	×	×	×	×
			CC-Link	×	×	×	×	×	×

Own board	Connected station CPU	Relayed network	Relayed station CPU					
	Q12DCCPU-V		QCPU (Q mode)	Q12DC CPU-V	LCPU	QSCPU	Q motion CPU	FXCPU
①	②	CC IE Control CC IE Field	③	③[*2]	×	③[*2]	③[*2]	×
		MELSECNET/H	③	③	×	③	③	×
		Ethernet	×	×	×	×	×	×
		Serial communication module	×	×	×	×	×	×
		CC-Link	×	×	×	×	×	×

○: Accessible (The number in the circle is the property pattern number.), ×: Inaccessible

*1 : Relayed stations cannot be accessed via Q motion CPU.

*2 : Q12DCCPU-V, QSCPU, and Q motion CPU cannot be accessed using CC-Link IE Field Network because CC-Link IE Field Network is not supported.

*3 : LCPU cannot be accessed using CC-Link IE Controller Network because CC-Link IE Controller Network is not supported.

Own board	Connected station CPU	Relayed network	Relayed station CPU					
	LCPU		QCPU (Q mode)	Q12DC CPU-V	LCPU	QSCPU	Q motion CPU	FXCPU
①	②	CC IE Control CC IE Field	×	×	×	×	×	×
		MELSECNET/H	×	×	×	×	×	×
		Ethernet	×	×	×	×	×	×
		Serial communication module	×	×	×	×	×	×
		CC-Link	×	×	×	×	×	×

○: Accessible (The number in the circle is the property pattern number.), ×: Inaccessible

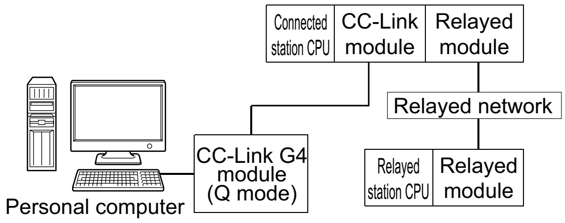
(3) Property list

The following table shows the property settings of communication paths. The setting is not necessary for those properties without description.

Property	Default value	Property pattern		
		①	②	③
ActCpuType	34(CPU_Q02CPU)	CPU type corresponding to the target station		
ActDestinationIONumber	0 (0x00)	Fixed to 0 (0x00)	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No.1: 992 (0x3E0) No.2: 993 (0x3E1) No.3: 994 (0x3E2) No.4: 995 (0x3E3) For Redundant CPU Control system: 976 (0x3D0) No specification: 1023 (0x3FF)	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No.1: 992 (0x3E0) No.2: 993 (0x3E1) No.3: 994 (0x3E2) No.4: 995 (0x3E3) For Redundant CPU Control system: 976 (0x3D0) No specification: 1023 (0x3FF)
ActIONumber	1023 (0x3FF)	Fixed to 0 (0x00)	Fixed to 1023 (0x3FF)	Fixed to 1023 (0x3FF)
ActNetworkNumber	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station side module network number
ActPortNumber	1 (PORT_1)	Port number of personal computer side CC-Link board, PORT 1 to PORT 4 (first to fourth boards)		
ActProtocolType	0x04 (PROTOCOL_SERIAL)	PROTOCOL_CCLINK (0x07)		
ActStationNumber	255 (0xFF)	Fixed to 255 (0xFF)	Target station side CC-Link module station number	Target station side module station number
ActUnitNumber	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Connected station side CC-Link module station number
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_CCLINKBOARD (0x0C)		

4.10 CC-Link G4 communication

(1) Configuration



(2) Property patterns

Connected station CPU		Relayed network	Relayed station CPU					
QCPU (Q mode)	Q motion CPU		QCPU (Q mode)	Q12DC CPU-V	LCPU	QSCPU	Q motion CPU	FXCPU
①	②*1	CC IE Control CC IE Field	②	②*2	②*3	②*2	②*2	×
		MELSECNET/H	②	②	×	②	②	×
		Ethernet	②	×	×	②	②	×
		Serial communication	×	×	×	×	×	×
		CC-Link	×	×	×	×	×	×

Connected station CPU	Relayed network	Relayed station CPU					
Q12DCCPU-V		QCPU (Q mode)	Q12DC CPU-V	LCPU	QSCPU	Q motion CPU	FXCPU
①	CC IE Control CC IE Field	②	②*2	②*3	②*2	②*2	×
	MELSECNET/H	②	②	×	②	②	×
	Ethernet	×	×	×	×	×	×
	Serial communication	×	×	×	×	×	×
	CC-Link	×	×	×	×	×	×

Connected station CPU	Relayed network	Relayed station CPU					
LCPU		QCPU (Q mode)	Q12DC CPU-V	LCPU	QSCPU	Q motion CPU	FXCPU
①	CC IE Control CC IE Field	×	×	×	×	×	×
	MELSECNET/H	×	×	×	×	×	×
	Ethernet	×	×	×	×	×	×
	Serial communication	×	×	×	×	×	×
	CC-Link	×	×	×	×	×	×

○: Accessible (The number in the circle is the property pattern number.), ×: Inaccessible

*1 : Relayed stations cannot be accessed via Q motion CPU.

*2 : Q12DCCPU-V, QSCPU, and Q motion CPU cannot be accessed using CC-Link IE Field Network because CC-Link IE Field Network is not supported.

*3 : LCPU cannot be accessed using CC-Link IE Controller Network because CC-Link IE Controller Network is not supported.

(3) Property list

The following table shows the property settings of communication paths. The setting is not necessary for those properties without description.

Property	Default value	Property pattern		
		①	②	③
ActBaudRate	19200 (BAUDRATE_19200)	BAUDRATE_9600, BAUDRATE_19200, BAUDRATE_38400, BAUDRATE_57600, BAUDRATE_115200		
ActConnectUnitNumber	0 (0x00)	Connected station side module station number		
ActControl	8 (TRC_DTR_OR_RTS)	Depending on the used cable.		
ActCpuType	34 (CPU_Q02CPU)	CPU type corresponding to the target station		
ActIONumber[*1]	1023 (0x3FF)	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No. 1: 992 (0x3E0) No. 2: 993 (0x3E1) No. 3: 994 (0x3E2) No. 4: 995 (0x3E3) For Redundant CPU Control system: 976 (0x3D0) No specification: 1023 (0x3FF)	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No. 1: 992 (0x3E0) No. 2: 993 (0x3E1) No. 3: 994 (0x3E2) No. 4: 995 (0x3E3) For Redundant CPU Control system: 976 (0x3D0) No specification: 1023 (0x3FF)	Connected station side relayed module I/O address
ActNetworkNumber	0 (0x00)	Fixed to 0 (0x00)	Target station side module network number	Fixed to 0 (0x00)
ActPortNumber	1 (PORT_1)	Personal computer side COM port number		
ActProtocolType	0x04 (PROTOCOL_SERIAL)	PROTOCOL_SERIAL (0x04)		
ActStationNumber	255 (0xFF)	Fixed to 255 (0xFF)	Target station side module station number	Fixed to 255 (0xFF)
ActThroughNetworkType	0 (0x00)	MELSECNET/10 is not included. : 0 (0x00) MELSECNET/10 is included. : 1 (0x01)		
ActTimeOut	10000	Any value specified by user in ms units		
ActUnitNumber	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station side module station number
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_G4QNCPU (0x1B)		

*1 : For the I/O address, specify the value of the actual start I/O number divided by 16.

4.11 GX Simulator Communication

(1) Property list

The following table shows the property settings of communication paths. The setting is not necessary for those properties without description.

Property	Default value	Property pattern	
		Host station	Other station
ActCpuType	34 (CPU_Q02CPU)	CPU type corresponding to the target station	
ActNetworkNumber	0 (0x00)	Fixed to 0 (0x00)	Target station side network number
ActProtocolType	0x04 (PROTOCOL_SERIAL)	PROTOCOL_SHAREDMEMORY(0x06)	
ActStationNumber	255 (0xFF)	Fixed to 255 (0xFF)	Target station side station number
ActTimeOut	10000	Any value specified by user in ms units	
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_SIMULATOR (0x0B)	

Point

When any of the following settings is included in the property setting, the host station is accessed.

- ActNetworkNumber: The number outside the range of 1 to 255 is set.
- ActStationNumber: The number outside the range of 0 to 64 is set.

4.12 GX Simulator2 Communication

(1) Property list

The following table shows the property settings of communication paths. The setting is not necessary for those properties without description.

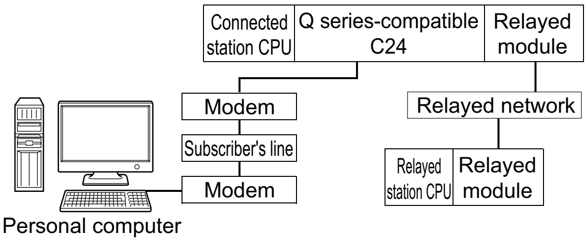
Property	Default value	Property pattern
ActTargetSimulator	0 (0x00)	Refer to the property [ActTargetSimulator] in Section 3.2 .
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_SIMULATOR2 (0x30)

4.13 Modem Communication

4.13.1 Modem communication when the connected module is Q series-compatible C24

(1) When a relayed module other than the connected station side Q series-compatible C24 exists

(a) Configuration



(b) Property patterns

Connected station CPU	Relayed network	Relayed station CPU					
QCPU (Q mode)		QCPU (Q mode)	Q12DC CPU-V	LCPU	QSCPU	Q motion CPU	FXCPU
①	CC IE Control CC IE Field	②	②*1	②*2	②*1	×	×
	MELSECNET/H	②	②	×	②	×	×
	Ethernet	②	×	×	②	×	×
	Serial communication module	③*3	×	③	×	×	×
	CC-Link	④	④	④	×	×	×

○: Accessible (The number in the circle is the property pattern number.), ×: Inaccessible

*1 : Q12DCCPU-V and QSCPU cannot be accessed using CC-Link IE Field Network because CC-Link IE Field Network is not supported.

*2 : LCPU cannot be accessed using CC-Link IE Controller Network because CC-Link IE Controller Network is not supported.

*3 : For Redundant CPU, serial communication modules on the main base cannot be accessed.

(c) Property list

The following table shows the property settings of communication paths. The setting is not necessary for those properties without description.

Property	Default value	Property pattern			
		①	②	③	④
ActATCommand	Null	Any value specified by user			
ActATCommand PasswordCancel RetryTimes	3	Any value specified by user			
ActATCommand ResponseWait Time	1	Any value specified by user (in seconds)			

Property	Default value	Property pattern			
		①	②	③	④
ActBaudRate	19200 (BAUDRATE_ 19200)	BAUDRATE_9600, BAUDRATE_19200, BAUDRATE_38400, BAUDRATE_57600, BAUDRATE_115200 (Match to the setting of Q series-compatible C24.)			
ActCallback CancelWaitTime	90	Any value specified by user (in seconds)			
ActCallback DelayTime	20	Any value specified by user (in seconds)			
ActCallback Number	Null	Telephone number of any value specified by user			
ActCallback ReceptionWaiting TimeOut	120	Any value specified by user (in seconds)			
ActConnection CDWaitTime	90	Any value specified by user (in seconds)			
ActConnection ModemReport WaitTime	5	Any value specified by user (in seconds)			
ActConnectUnit Number	0 (0x00)	Connected station side module station number			
ActConnectWay	0 (TEL_AUTO_ CONNECT)	TEL_AUTO_CONNECT(0), TEL_AUTO_CALLBACK(1), TEL_AUTO_CALLBACK_NUMBER(2), TEL_CALLBACK(3), TEL_CALLBACK_NUMBER(4), TEL_CALLBACK_REQUEST(5), TEL_CALLBACK_REQUEST_NUMBER(6), TEL_CALLBACK_WAIT(7) (Depending on callback function setting)			
ActCpuType	34 (CPU_Q02CPU)	CPU type corresponding to target station			
ActDestination IONumber	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No.1: 992 (0x3E0) No.2: 993 (0x3E1) No.3: 994 (0x3E2) No.4: 995 (0x3E3) For Redundant CPU Control system: 976 (0x3D0) No specification: 1023 (0x3FF)	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No.1: 992 (0x3E0) No.2: 993 (0x3E1) No.3: 994 (0x3E2) No.4: 995 (0x3E3) For Redundant CPU Control system: 976 (0x3D0) No specification: 1023 (0x3FF)
ActDialNumber	Null	Telephone number of any value specified by user			
ActDidProperty Bit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActDisconnection CDWaitTime	5	Any value specified by user (in seconds)			
ActDisconnection DelayTime	3	Any value specified by user (in seconds)			
ActDsidProperty Bit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActIntelligent PreferenceBit	0 (0x00)	Fixed to 0 (0x00)			

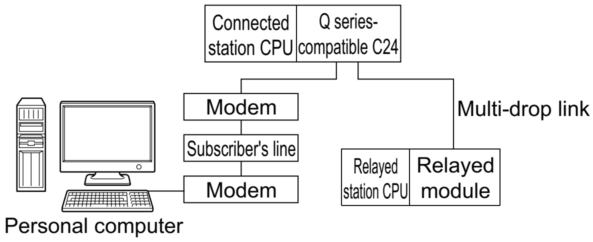
Property	Default value	Property pattern			
		①	②	③	④
ActIONumber[*1]	1023 (0x3FF)	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No.1: 992 (0x3E0) No.2: 993 (0x3E1) No.3: 994 (0x3E2) No.4: 995 (0x3E3) For Redundant CPU Control system: 976 (0x3D0) No specification: 1023 (0x3FF)	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No.1: 992 (0x3E0) No.2: 993 (0x3E1) No.3: 994 (0x3E2) No.4: 995 (0x3E3) For Redundant CPU Control system: 976 (0x3D0) No specification: 1023 (0x3FF)	Connected station side relayed module I/O address	Connected station side relayed module I/O address
ActLineType	1 (LINETYPE_TONE)	LINETYPE_PULSE(0), LINETYPE_TONE(1), LINETYPE_ISDN(2)			
ActMultiDropChannelNumber[*2]	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Multi-drop channel number	Fixed to 0 (0x00)
ActNetworkNumber	0 (0x00)	Fixed to 0 (0x00)	Target station side module network number	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActOutsideLineNumber	Null	Outside line access number of any value specified by user			
ActPassword	Null	Password set to Q series-compatible C24			
ActPasswordCancelResponseWaitTime	5	Any value specified by user (in seconds)			
ActPortNumber	1 (PORT_1)	Personal computer side COM port number			
ActProtocolType	0x04 (PROTOCOL_SERIAL)	PROTOCOL_SERIALMODEM (0x0E)			
ActStationNumber	255 (0xFF)	Fixed to 255 (0xFF)	Target station side module station number	Fixed to 255 (0xFF)	Fixed to 255 (0xFF)
ActThroughNetworkType	0 (0x00)	MELSECNET/10 is not included. : 0 (0x00) MELSECNET/10 is included. : 1 (0x01)			
ActTimeOut	10000	Any value specified by user in ms units			
ActTransmissionDelayTime	0	Any value specified by user (in seconds)			
ActUnitNumber	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station side module station number	Target station side module station number
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_QJ71C24 (0x19)			

*1 : For the I/O address, specify the value of the actual start I/O number divided by 16.

*2 : Specify the following value for the channel number to be multi-drop linked.
0: Default channel of module
1: Channel 1
2: Channel 2

(2) When performing multi-drop link on the connected station side Q series-compatible C24 with the relayed module

(a) Configuration



(b) Property patterns

Connected station CPU		Relayed network	Relayed station CPU					
QCPU (Q mode)			QCPU (Q mode)	Q12DC CPU-V	LCPU	QSCPU	Q motion CPU	FXCPU
Independent mode*1	①	Serial communication	②*2	×	②	×	×	×

○: Accessible (The number in the circle is the property pattern number.), ×: Inaccessible

*1 : The independent mode indicates that the parameters are set as follows.
- CH1 side: Operation setting for transmission setting = independent (0), communication protocol setting = 0
- CH2 side: Operation setting for transmission setting = independent (0)

*2 : For Redundant CPU, serial communication modules on the main base cannot be accessed.

(c) Property list

The following table shows the property settings of communication paths. The setting is not necessary for those properties without description.

Property	Default value	Property pattern	
		①	②
ActATCommand	Null	Any value specified by user	
ActATCommandPasswordCancelRetryTimes	3	Any value specified by user	
ActATCommandResponseWaitTime	1	Any value specified by user (in seconds)	
ActBaudRate	19200 (BAUDRATE_19200)	BAUDRATE_9600, BAUDRATE_19200, BAUDRATE_38400, BAUDRATE_57600, BAUDRATE_115200 (Match to the setting of Q series-compatible C24.)	
ActCallbackCancelWaitTime	90	Any value specified by user (in seconds)	
ActCallbackDelayTime	20	Any value specified by user (in seconds)	
ActCallbackNumber	Null	Any value specified by user (in seconds)	
ActCallbackReceptionWaitingTimeOut	120	Any value specified by user (in seconds)	
ActConnectionCDWaitTime	90	Any value specified by user (in seconds)	
ActConnectionModemReportWaitTime	5	Any value specified by user (in seconds)	
ActConnectUnitNumber	0 (0x00)	Connected station side module station number	
ActConnectWay	0 (TEL_AUTO_CONNECT)	TEL_AUTO_CONNECT(0), TEL_AUTO_CALLBACK(1), TEL_AUTO_CALLBACK_NUMBER(2), TEL_CALLBACK(3), TEL_CALLBACK_NUMBER(4), TEL_CALLBACK_REQUEST(5), TEL_CALLBACK_REQUEST_NUMBER(6), TEL_CALLBACK_WAIT(7) (Depending on callback function setting)	

Property	Default value	Property pattern	
		①	②
ActCpuType	34 (CPU_Q02CPU)	CPU type corresponding to target station	
ActDestinationIONumber	0 (0x00)	Fixed to 0 (0x00)	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No.1: 992 (0x3E0) No.2: 993 (0x3E1) No.3: 994 (0x3E2) No.4: 995 (0x3E3) For Redundant CPU Control system: 976 (0x3D0) No specification: 1023 (0x3FF)
ActDialNumber	Null	Telephone number of any value specified by user	
ActDidPropertyBit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)
ActDisconnectionCDWaitTime	5	Any value specified by user (in seconds)	
ActDisconnectionDelayTime	3	Any value specified by user (in seconds)	
ActDsidPropertyBit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 1 (0x01)
ActIntelligentPreferenceBit	0 (0x00)	Fixed to 0 (0x00)	
ActIONumber*1	1023 (0x3FF)	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No.1: 992 (0x3E0) No.2: 993 (0x3E1) No.3: 994 (0x3E2) No.4: 995 (0x3E3) For Redundant CPU Control system: 976 (0x3D0) No specification: 1023 (0x3FF)	Connected station side relayed module I/O address
ActLineType	1 (LINETYPE_TONE)	LINETYPE_PULSE(0), LINETYPE_TONE(1), LINETYPE_ISDN(2)	
ActMultiDropChannelNumber*2	0 (0x00)	Fixed to 0 (0x00)	Multi-drop channel number
ActNetworkNumber	0 (0x00)	Fixed to 0 (0x00)	
ActOutsideLineNumber	Null	Outside line access number of any value specified by user	
ActPassword	Null	Password set to Q series-compatible C24	
ActPasswordCancelResponseWaitTime	5	Any value specified by user (in seconds)	
ActPortNumber	1 (PORT_1)	Personal computer side COM port number	
ActProtocolType	0x04 (PROTOCOL_SERIAL)	PROTOCOL_SERIALMODEM (0x0E)	
ActStationNumber	255 (0xFF)	Fixed to 255 (0xFF)	
ActThroughNetworkType	0 (0x00)	MELSECNET/10 is not included. : 0 (0x00) MELSECNET/10 is included. : 1 (0x01)	
ActTimeOut	10000	Any value specified by user in ms units	
ActTransmissionDelayTime	0	Any value specified by user (in seconds)	

Property	Default value	Property pattern	
		①	②
ActUnitNumber	0 (0x00)	Fixed to 0 (0x00)	Target station side module station number
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_QJ71C24 (0x19)	

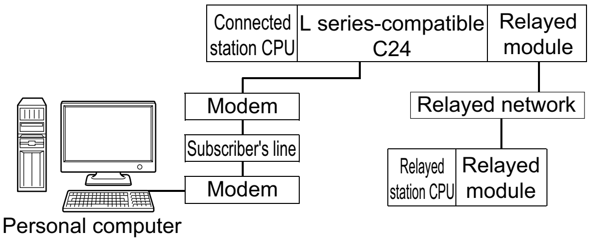
*1 : For the I/O address, specify the value of the actual start I/O number divided by 16.

*2 : Specify the following value for the channel number to be multi-drop linked.
0: Default channel of module
1: Channel 1
2: Channel 2

4.13.2 Modem communication when the connected module is L series-compatible C24

(1) When a relayed module other than the connected station side LJ71C24 exists

(a) Configuration



(b) Property patterns

Connected station CPU	Relayed network	Relayed station CPU					
LCPU		QCPU (Q mode)	Q12DC CPU-V	LCPU	QSCPU	Q motion CPU	FXCPU
①[*1]	CC IE Field [*2]	④	×	④	×	×	×
	MELSECNET/H	×	×	×	×	×	×
	Ethernet	×	×	×	×	×	×
	Serial communication	②[*3]	×	②	×	×	×
	CC-Link	③	③	③	×	×	×

○: Accessible (The number in the circle is the property pattern number.), ×: Inaccessible

*1 : Indicates the CH2 side setting. (CH1 side is fixed to independent mode)

*2 : LCPU cannot be accessed using CC-Link IE Controller Network because CC-Link IE Controller Network is not supported.

*3 : For Redundant CPU, computer link modules on the main base cannot be accessed.

(c) Property list

The following table shows the property settings of communication paths. The setting is not necessary for those properties without description.

Property	Default value	Property pattern			
		①	②	③	④
ActATCommand	Null	Any value specified by user			
ActATCommand PasswordCancel RetryTimes	3	Any value specified by user			
ActATCommand ResponseWaitTime	1	Any value specified by user (in seconds)			
ActBaudRate	19200 (BAUDRATE_ 19200)	BAUDRATE_9600, BAUDRATE_19200, BAUDRATE_38400, BAUDRATE_57600, BAUDRATE_115200 (Match to the setting of LJ71C24)			
ActCallback CancelWaitTime	90	Any value specified by user (in seconds)			
ActCallback DelayTime	20	Any value specified by user (in seconds)			
ActCallbackNumber	Null	Telephone number of any value specified by user			

Property	Default value	Property pattern			
		①	②	③	④
ActCallbackReceptionWaitingTimeOut	120	Any value specified by user (in seconds)			
ActConnectionCDWaitTime	90	Any value specified by user (in seconds)			
ActConnectionModemReportWaitTime	5	Any value specified by user (in seconds)			
ActConnectUnitNumber	0 (0x00)	Connected station side module station number			
ActConnectWay	0 (TEL_AUTO_CONNECT)	TEL_AUTO_CONNECT(0), TEL_AUTO_CALLBACK(1), TEL_AUTO_CALLBACK_NUMBER(2), TEL_CALLBACK(3), TEL_CALLBACK_NUMBER(4), TEL_CALLBACK_REQUEST(5), TEL_CALLBACK_REQUEST_NUMBER(6), TEL_CALLBACK_WAIT(7)			
ActCpuType	34 (CPU_Q02CPU)	CPU type corresponding to target station			
ActDestinationIONumber	0 (0x00)	Fixed to 0 (0x00)	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No.1: 992 (0x3E0) No.2: 993 (0x3E1) No.3: 994 (0x3E2) No.4: 995 (0x3E3) For Redundant CPU Control system: 976 (0x3D0) No specification: 1023 (0x3FF)	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No.1: 992 (0x3E0) No.2: 993 (0x3E1) No.3: 994 (0x3E2) No.4: 995 (0x3E3) For Redundant CPU Control system: 976 (0x3D0) No specification: 1023 (0x3FF)	Fixed to 0 (0x00)
ActDialNumber	Null	Telephone number of any value specified by user			
ActDidPropertyBit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Fixed to 1 (0x01)
ActDisconnectionCDWaitTime	5	Any value specified by user (in seconds)			
ActDisconnectionDelayTime	3	Any value specified by user (in seconds)			
ActDsidPropertyBit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Fixed to 1 (0x01)
ActIntelligentPreferenceBit	0 (0x00)	Fixed to 0 (0x00)			

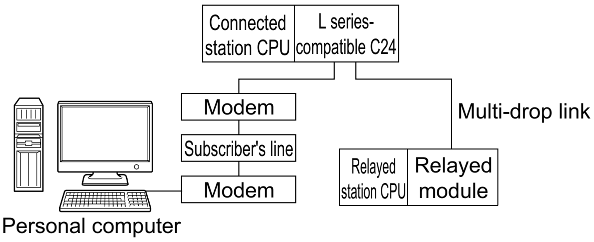
Property	Default value	Property pattern			
		①	②	③	④
ActIONumber[*1]	1023 (0x3FF)	Fixed to 1023 (0x3FF)	Connected station side relayed module I/O address	Connected station side relayed module I/O address	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No.1: 992 (0x3E0) No.2: 993 (0x3E1) No.3: 994 (0x3E2) No.4: 995 (0x3E3) For Redundant CPU Control system: 976 (0x3D0) No specification: 1023 (0x3FF)
ActLineType	1 (LINETYPE_ TONE)	LINETYPE_PULSE(0), LINETYPE_TONE(1), LINETYPE_ISDN(2)			
ActMultiDrop ChannelNumber[*2]	0 (0x00)	Fixed to 0 (0x00)	Multi-drop channel number	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActNetworkNumber	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station side module network number
ActOutsideLine Number	Null	Outside line access number of any value specified by user			
ActPassword	Null	Password set to the module			
ActPasswordCancel ResponseWaitTime	5	Any value specified by user (in seconds)			
ActPortNumber	1 (PORT_1)	Personal computer side COM port number			
ActProtocolType	0x04 (PROTOCOL_ SERIAL)	PROTOCOL_SERIALMODEM (0x0E)			
ActStationNumber	255 (0xFF)	Fixed to 255 (0xFF)	Fixed to 255 (0xFF)	Fixed to 255 (0xFF)	Module station number on target station side
ActThrough NetworkType	0 (0x00)	MELSECNET/10 is not included. : 0 (0x00) MELSECNET/10 is included. : 1 (0x01)			
ActTimeOut	10000	Any value specified by user in ms units			
ActTransmission DelayTime	0	Any value specified by user (in seconds)			
ActUnitNumber	0 (0x00)	Fixed to 0 (0x00)	Target station side module station number	Target station side module station number	Fixed to 0 (0x00)
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_LJ71C24 (0x54)			

*1 : For the I/O address, specify the value of the actual start I/O number divided by 16.

*2 : Specify the following value for the channel number to be multi-drop linked.
0: Default channel of module
1: Channel 1
2: Channel 2

(2) When performing multi-drop link on the connected station side LJ71C24 with the relayed module

(a) Configuration



(b) Property patterns

Connected station CPU		Relayed network	Relayed station CPU					
LCPU			QCPU (Q mode)	Q12DC CPU-V	LCPU	QSCPU	Q motion CPU	FXCPU
Independent mode	①	Serial communication	②[*1]	×	②	×	×	×

○: Accessible (The number in the circle is the property pattern number.), ×: Inaccessible

*1 : For Redundant CPU, computer link modules on the main base cannot be accessed.

(c) Property list

The following table shows the property settings of communication paths. The setting is not necessary for those properties without description.

Property	Default value	Property pattern	
		①	②
ActATCommand	Null	Any value specified by user	
ActATCommandPasswordCancelRetryTimes	3	Any value specified by user	
ActATCommandResponseWaitTime	1	Any value specified by user (in seconds)	
ActBaudRate	19200(BAUDRATE_19200)	BAUDRATE_9600, BAUDRATE_19200, BAUDRATE_38400, BAUDRATE_57600, BAUDRATE_115200 (Match to the setting of LJ71C24)	
ActCallbackCancelWaitTime	90	Any value specified by user (in seconds)	
ActCallbackDelayTime	20	Any value specified by user (in seconds)	
ActCallbackNumber	Null	Any value specified by user (in seconds)	
ActCallbackReceptionWaitingTimeOut	120	Any value specified by user (in seconds)	
ActConnectionCDWaitTime	90	Any value specified by user (in seconds)	
ActConnectionModemReportWaitTime	5	Any value specified by user (in seconds)	
ActConnectUnitNumber	0 (0x00)	Connected station side module station number	
ActConnectWay	0 (TEL_AUTO_CONNECT)	TEL_AUTO_CONNECT(0), TEL_AUTO_CALLBACK(1), TEL_AUTO_CALLBACK_NUMBER(2), TEL_CALLBACK(3), TEL_CALLBACK_NUMBER(4), TEL_CALLBACK_REQUEST(5), TEL_CALLBACK_REQUEST_NUMBER(6), TEL_CALLBACK_WAIT(7)	
ActCpuType	34 (CPU_Q02CPU)	CPU type corresponding to target station	

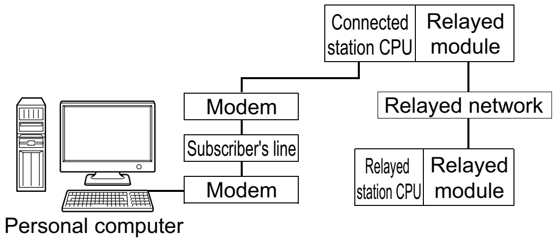
Property	Default value	Property pattern	
		①	②
ActDestinationIONumber	0 (0x00)	Fixed to 0 (0x00)	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No.1: 992 (0x3E0) No.2: 993 (0x3E1) No.3: 994 (0x3E2) No.4: 995 (0x3E3) For Redundant CPU Control system: 976 (0x3D0) No specification: 1023 (0x3FF)
ActDialNumber	Null	Telephone number of any value specified by user	
ActDidPropertyBit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)
ActDisconnectionCDWaitTime	5	Any value specified by user (in seconds)	
ActDisconnectionDelayTime	3	Any value specified by user (in seconds)	
ActDsidPropertyBit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)
ActIntelligentPreferenceBit	0 (0x00)	Fixed to 0 (0x00)	
ActIONumber[*1]	1023 (0x3FF)	Fixed to 1023 (0x3FF)	Connected station side relayed module I/O address
ActLineType	1 (LINETYPE_TONE)	LINETYPE_PULSE(0), LINETYPE_TONE(1), LINETYPE_ISDN(2)	
ActMultiDropChannelNumber[*2]	0 (0x00)	Fixed to 0 (0x00)	Multi-drop channel number
ActNetworkNumber	0 (0x00)	Fixed to 0 (0x00)	
ActOutsideLineNumber	Null	Outside line access number of any value specified by user	
ActPassword	Null	Password set to the module	
ActPasswordCancelResponseWaitTime	5	Any value specified by user (in seconds)	
ActPortNumber	1 (PORT_1)	Personal computer side COM port number	
ActProtocolType	0x04 (PROTOCOL_SERIAL)	PROTOCOL_SERIALMODEM(0x0E)	
ActStationNumber	255 (0xFF)	Fixed to 255 (0xFF)	
ActThroughNetworkType	0 (0x00)	MELSECNET/10 is not included. : 0 (0x00) MELSECNET/10 is included. : 1 (0x01)	
ActTimeOut	10000	Any value specified by user in ms units	
ActTransmissionDelayTime	0	Any value specified by user (in seconds)	
ActUnitNumber	0 (0x00)	Fixed to 0 (0x00)	Target station side module station number
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_LJ71C24 (0x54)	

*1 : For the I/O address, specify the value of the actual start I/O number divided by 16.

*2 : Specify the following value for the channel number to be multi-drop linked.
0: Default channel of module
1: Channel 1
2: Channel 2

4.13.3 Modem communication when the connected station is FXCPU

(1) Configuration



(2) Property patterns

Connected station CPU	Relayed network	Relayed station CPU					
FXCPU		QCPU (Q mode)	Q12DC CPU-V	LCPU	QSCPU	Q motion CPU	FXCPU
①	CC IE Control CC IE Field	×	×	×	×	×	×
	MELSECNET/H	×	×	×	×	×	×
	Ethernet	×	×	×	×	×	×
	Serial communication	×	×	×	×	×	×
	CC-Link	×	×	×	×	×	×

○: Accessible (The number in the circle is the property pattern number.), ×: Inaccessible

(3) Property list

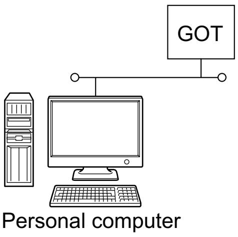
The following table shows the property settings of communication paths. The setting is not necessary for those properties without description.

Property	Default value	Property pattern
		①
ActATCommand	Null	Any value specified by user
ActATCommandPasswordCancelRetryTimes	3	Any value specified by user
ActATCommandResponseWaitTime	1	Any value specified by user (in seconds)
ActCallbackCancelWaitTime	90	Any value specified by user (in seconds)
ActCallbackDelayTime	20	Any value specified by user (in seconds)
ActCallbackNumber	Null	Any value specified by user (in seconds)
ActCallbackReceptionWaitingTimeOut	120	Any value specified by user (in seconds)
ActConnectionCDWaitTime	90	Any value specified by user (in seconds)
ActConnectionModemReportWaitTime	5	Any value specified by user (in seconds)
ActCpuType	34 (CPU_Q02CPU)	CPU type corresponding to target station
ActDialNumber	Null	Telephone number of any value specified by user
ActDisconnectionCDWaitTime	5	Any value specified by user (in seconds)
ActDisconnectionDelayTime	3	Any value specified by user (in seconds)
ActLineType	1 (LINETYPE_TONE)	LINETYPE_PULSE(0), LINETYPE_TONE(1), LINETYPE_ISDN(2)

Property	Default value	Property pattern
		①
ActOutsideLineNumber	Null	Outside line access number of any value specified by user
ActPasswordCancelResponseWaitTime	5	Any value specified by user (in seconds)
ActPortNumber	1 (PORT_1)	Personal computer side COM port number
ActProtocolType	0x04 (PROTOCOL_SERIAL)	PROTOCOL_TEL (0x0A)
ActTimeOut	10000	Any value specified by user in ms units
ActTransmissionDelayTime	0	Any value specified by user (in seconds)
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_FXCPU (0x0F)

4.14 Gateway Function Communication

(1) Configuration



(2) Property list

The following table shows the property settings of communication paths. The setting is not necessary for those properties without description.

Property	Default value	Property pattern
		①
ActHostAddress	1.1.1.1	Host name or IP address of connected GOT
ActPortNumber	1 (PORT_1)	Personal computer side port number
ActProtocolType	0x04 (PROTOCOL_SERIAL)	PROTOCOL_UDPIP (0x08)
ActTimeOut	10000	Any value specified by user in ms units
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_A900GOT (0x21)

4.15 GOT Transparent Function Communication

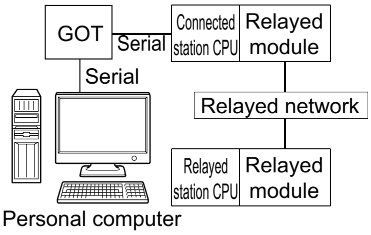
Point

For the applicable system configuration, refer to GOT1000 Series Connection Manual.

4.15.1 Personal computer side port: Serial, GOT1000 side port: Serial, CPU side port: Direct connection

(1) When the connected station is QCPU (Q mode)

(a) Configuration



(b) Property patterns

Connected station CPU		Relayed network	Relayed station CPU					
QCPU (Q mode)*1	Q motion CPU		QCPU (Q mode)*1	Q12DCCPU-V	LCPU	QSCPU	Q motion CPU	FXCPU
①	①*2	CC IE Control CC IE Field	②	②*3	②*4	×	②*3	×
		MELSECNET/H	②	②	×	×	②	×
		Ethernet	②	×	×	×	②	×
		Serial communication	③	×	③	×	③	×
		CC-Link	④	④	④	×	④	×

○: Accessible (The number in the circle is the property pattern number.), ×: Inaccessible

*1 : Redundant CPU cannot be accessed.

*2 : Relayed stations cannot be accessed via Q motion CPU.

*3 : Q12DCCPU-V and Q motion CPU cannot be accessed using CC-Link IE Field Network because CC-Link IE Field Network is not supported.

*4 : LCPU cannot be accessed using CC-Link IE Controller Network because CC-Link IE Controller Network is not supported.

(c) Property list

The following table shows the property settings of communication paths. The setting is not necessary for those properties without description.

Property	Default value	Property pattern			
		①	②	③	④
ActBaudRate	19200 (BAUDRATE_ 19200)	BAUDRATE_9600, BAUDRATE_19200, BAUDRATE_38400, BAUDRATE_57600, BAUDRATE_115200			
ActCpuType	34 (CPU_Q02CPU)	CPU type corresponding to target station			
ActDestination IONumber	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No.1: 992 (0x3E0) No.2: 993 (0x3E1) No.3: 994 (0x3E2) No.4: 995 (0x3E3)	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No.1: 992 (0x3E0) No.2: 993 (0x3E1) No.3: 994 (0x3E2) No.4: 995 (0x3E3)
ActDidProperty Bit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActDsidProperty Bit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActIntelligent PreferenceBit	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station QCPU (Q mode), Q12DCCPU-V, LCPU : 1 (0x01) Other than the above : 0 (0x00)	Target station QCPU (Q mode), Q12DCCPU-V, LCPU : 1 (0x01) Other than the above : 0 (0x00)
ActIONumber*1	1023 (0x3FF)	For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No.1: 992 (0x3E0) No.2: 993 (0x3E1) No.3: 994 (0x3E2) No.4: 995 (0x3E3)	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No.1: 992 (0x3E0) No.2: 993 (0x3E1) No.3: 994 (0x3E2) No.4: 995 (0x3E3)	Connected station side module I/O address	Connected station side module I/O address
ActMultiDrop ChannelNumber*2	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Multi-drop channel number	Fixed to 0 (0x00)
ActNetwork Number	0 (0x00)	Fixed to 0 (0x00)	Target station side module network number	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActPortNumber	1 (PORT_1)	Personal computer side COM port number			
ActProtocolType	0x04 (PROTOCOL_ SERIAL)	PROTOCOL_SERIAL (0x04)			
ActStation Number	255 (0xFF)	Fixed to 255 (0xFF)	Target station side module station number	Fixed to 255 (0xFF)	Fixed to 255 (0xFF)

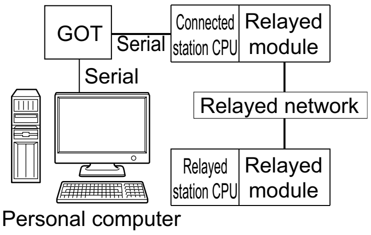
Property	Default value	Property pattern			
		①	②	③	④
ActThrough NetworkType	0 (0x00)	MELSECNET/10 is not included. : 0 (0x00) MELSECNET/10 is included. : 1 (0x01)			
ActTimeOut	10000	Any value specified by user in ms units			
ActUnitNumber	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station side module station number	Target station side module station number
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_QNCPU (0x13)			

*1 : For the I/O address, specify the value of the actual start I/O number divided by 16.

*2 : Specify the following value for the channel number to be multi-drop linked.
0: Default channel of module
1: Channel 1
2: Channel 2

(2) When the connected station is LCPU

(a) Configuration



(b) Property patterns

Connected station CPU	Relayed network	Relayed station CPU					
LCPU		QCPU (Q mode)[*1]	Q12DC CPU-V	LCPU	QSCPU	Q motion CPU	FXCPU
①	CC IE Field [*2]	④	×	④	×	×	×
	MELSECNET/H	×	×	×	×	×	×
	Ethernet	×	×	×	×	×	×
	Serial communication	②	×	②	×	×	×
	CC-Link	③	③	③	×	×	×

○: Accessible (The number in the circle is the property pattern number.), ×: Inaccessible

*1 : Redundant CPU cannot be accessed.

*2 : LCPU cannot be accessed using CC-Link IE Controller Network because CC-Link IE Controller Network is not supported.

(c) Property list

The following table shows the property settings of communication paths. The setting is not necessary for those properties without description.

Property	Default value	Property pattern			
		①	②	③	④
ActBaudRate	19200 (BAUDRATE_ 19200)	BAUDRATE_9600, BAUDRATE_19200, BAUDRATE_38400, BAUDRATE_57600, BAUDRATE_115200			
ActControl	8 (TRC_DTR _OR_RTS)	8 (TRC_DTR_OR_RTS)			
ActCpuType	34 (CPU_Q02CPU)	CPU type corresponding to target station			
ActDestination IONumber	0 (0x00)	Fixed to 0 (0x00)	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No.1: 992 (0x3E0) No.2: 993 (0x3E1) No.3: 994 (0x3E2) No.4: 995 (0x3E3)	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No.1: 992 (0x3E0) No.2: 993 (0x3E1) No.3: 994 (0x3E2) No.4: 995 (0x3E3)	Fixed to 0 (0x00)
ActDidPropertyBit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Fixed to 1 (0x01)

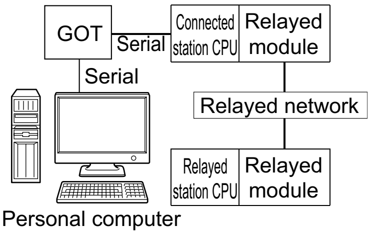
Property	Default value	Property pattern			
		①	②	③	④
ActDsidPropertyBit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Fixed to 1 (0x01)
ActIntelligent PreferenceBit	0 (0x00)	Fixed to 0 (0x00)	Target station QCPU (Q mode), Q12DCCPU-V, LCPU : 1 (0x01) Other than the above : 0 (0x00)	Target station QCPU (Q mode), Q12DCCPU-V, LCPU : 1 (0x01) Other than the above : 0 (0x00)	Fixed to 0 (0x00)
ActIONumber*1	1023 (0x3FF)	Fixed to 1023 (0x3FF)	Connected station side module I/O address	Connected station side module I/O address	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No.1: 992 (0x3E0) No.2: 993 (0x3E1) No.3: 994 (0x3E2) No.4: 995 (0x3E3)
ActMultiDrop ChannelNumber*2	0 (0x00)	Fixed to 0 (0x00)	Multi-drop channel number	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActNetworkNumber	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station side module network number
ActPortNumber	1 (PORT_1)	Personal computer side COM port number			
ActProtocolType	0x04 (PROTOCOL_ SERIAL)	PROTOCOL_SERIAL (0x04)			
ActStationNumber	255 (0xFF)	Fixed to 255 (0xFF)	Fixed to 255 (0xFF)	Fixed to 255 (0xFF)	Target station side module station number
ActThroughNetwork Type	0 (0x00)	MELSECNET/10 is not included. : 0 (0x00) MELSECNET/10 is included. : 1 (0x01)			
ActTimeOut	10000	Any value specified by user in ms units			
ActUnitNumber	0 (0x00)	Fixed to 0 (0x00)	Target station side module station number	Target station side module station number	Fixed to 0 (0x00)
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_LNCPU (0x50)			

*1 : For the I/O address, specify the value of the actual start I/O number divided by 16.

*2 : Specify the following value for the channel number to be multi-drop linked.
0: Default channel of module
1: Channel 1
2: Channel 2

(3) When the connected station is FXCPU

(a) Configuration



(b) Property patterns

Connected station CPU	Relayed network	Relayed station CPU					
FXCPU		QCPU (Q mode)	Q12DC CPU-V	LCPU	QSCPU	Q motion CPU	FXCPU
①	CC IE Control CC IE Field	×	×	×	×	×	×
	MELSECNET/H	×	×	×	×	×	×
	Ethernet	×	×	×	×	×	×
	Serial communication	×	×	×	×	×	×
	CC-Link	×	×	×	×	×	×

○: Accessible (The number in the circle is the property pattern number.), ×: Inaccessible

(c) Property list

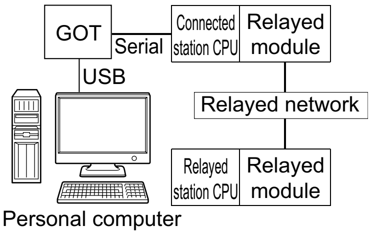
The following table shows the property settings of communication paths. The setting is not necessary for those properties without description.

Property	Default value	Property pattern
		①
ActBaudRate	19200 (BAUDRATE_19200)	BAUDRATE_9600, BAUDRATE_19200, BAUDRATE_38400, BAUDRATE_57600, BAUDRATE_115200
ActCpuType	34 (CPU_Q02CPU)	CPU type corresponding to target station
ActPortNumber	1 (PORT_1)	Personal computer side COM port number
ActProtocolType	0x04 (PROTOCOL_SERIAL)	PROTOCOL_SERIAL (0x04)
ActTimeOut	10000	Any value specified by user in ms units
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_FXCPU (0x0F)

4.15.2 Personal computer side port: USB, GOT1000 side port: USB, CPU side port: Direct connection

(1) When the connected station is QCPU (Q mode)

(a) Configuration



(b) Property patterns

Connected station CPU		Relayed network	Relayed station CPU					
QCPU (Q mode)*1	Q motion CPU		QCPU (Q mode)*1	Q12DC CPU-V	LCPU	QSCPU	Q motion CPU	FXCPU
①	①*2	CC IE Control CC IE Field	②	②*2	②*3	×	②	×
		MELSECNET/H	②	②	×	×	②	×
		Ethernet	②	×	×	×	②	×
		Serial communication	③	×	③	×	③	×
		CC-Link	④	④	④	×	④	×

○: Accessible (The number in the circle is the property pattern number.), ×: Inaccessible

*1 : Redundant CPU cannot be accessed.

*2 : Relayed stations cannot be accessed via Q motion CPU.

*3 : Q12DCCPU-V and Q motion CPU cannot be accessed using CC-Link IE Field Network because CC-Link IE Field Network is not supported.

*4 : LCPU cannot be accessed using CC-Link IE Controller Network because CC-Link IE Controller Network is not supported.

(c) Property list

The following table shows the property settings of communication paths. The setting is not necessary for those properties without description.

Property	Default value	Property pattern			
		①	②	③	④
ActCpuType	34 (CPU_Q02CPU)	CPU type corresponding to target station			
ActDestination IONumber	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No.1: 992 (0x3E0) No.2: 993 (0x3E1) No.3: 994 (0x3E2) No.4: 995 (0x3E3)	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No.1: 992 (0x3E0) No.2: 993 (0x3E1) No.3: 994 (0x3E2) No.4: 995 (0x3E3)
ActDidProperty Bit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActDsidProperty Bit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActIntelligent PreferenceBit	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station QCPU (Q mode), Q12DCCPU-V, LCPU : 1 (0x01) Other than the above : 0 (0x00)	Target station QCPU (Q mode), Q12DCCPU-V, LCPU : 1 (0x01) Other than the above : 0 (0x00)
ActIONumber[*1]	1023 (0x3FF)	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No.1: 992 (0x3E0) No.2: 993 (0x3E1) No.3: 994 (0x3E2) No.4: 995 (0x3E3)	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No.1: 992 (0x3E0) No.2: 993 (0x3E1) No.3: 994 (0x3E2) No.4: 995 (0x3E3)	Connected station side module I/O address	Connected station side module I/O address
ActMultiDrop ChannelNumber[*2]	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Multi-drop channel number	Fixed to 0 (0x00)
ActNetwork Number	0 (0x00)	Fixed to 0 (0x00)	Target station side module station number	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActProtocolType	0x04 (PROTOCOL_ SERIAL)	PROTOCOL_USBGOT (0x13)			
ActStation Number	255 (0xFF)	Fixed to 255 (0xFF)	Target station side module station number	Fixed to 255 (0xFF)	Fixed to 255 (0xFF)
ActThrough NetworkType	0 (0x00)	MELSECNET/10 is not included. : 0 (0x00) MELSECNET/10 is included. : 1 (0x01)			
ActTimeOut	10000	Any value specified by user in ms units			

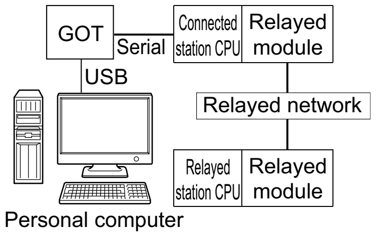
Property	Default value	Property pattern			
		①	②	③	④
ActUnitNumber	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station side module station number	Target station side module station number
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_QNCPU (0x13)			

*1 : For the I/O address, specify the value of the actual start I/O number divided by 16.

*2 : Specify the following value for the channel number to be multi-drop linked.
0: Default channel of module
1: Channel 1
2: Channel 2

(2) When the connected station is LCPU

(a) Configuration



*1 : Property patterns

Connected station CPU	Relayed network	Relayed station CPU					
LCPU		QCPU (Q mode)*1	Q12DC CPU-V	LCPU	QSCPU	Q motion CPU	FXCPU
①	CC IE Field *2	④	×	④	×	×	×
	MELSECNET/H	×	×	×	×	×	×
	Ethernet	×	×	×	×	×	×
	Serial communication	②	×	②	×	②	×
	CC-Link	③	③	③	×	③	×

○: Accessible (The number in the circle is the property pattern number.), ×: Inaccessible

*1 : Redundant CPU cannot be accessed.

*2 : LCPU cannot be accessed using CC-Link IE Controller Network because CC-Link IE Controller Network is not supported.

(b) Property list

The following table shows the property settings of communication paths. The setting is not necessary for those properties without description.

Property	Default value	Property pattern			
		①	②	③	④
ActCpuType	34 (CPU_Q02CPU)	CPU type corresponding to target station			
ActDestination IONumber	0 (0x00)	Fixed to 0 (0x00)	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No.1: 992 (0x3E0) No.2: 993 (0x3E1) No.3: 994 (0x3E2) No.4: 995 (0x3E3)	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No.1: 992 (0x3E0) No.2: 993 (0x3E1) No.3: 994 (0x3E2) No.4: 995 (0x3E3)	Fixed to 0 (0x00)
ActDidPropertyBit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Fixed to 1 (0x01)
ActDsidPropertyBit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Fixed to 1 (0x01)

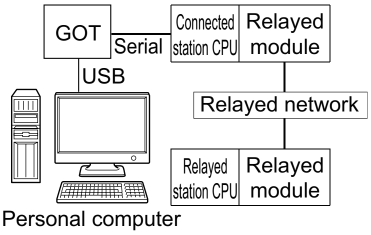
Property	Default value	Property pattern			
		①	②	③	④
ActIntelligent PreferenceBit	0 (0x00)	Fixed to 0 (0x00)	Target station QCPU (Q mode), Q12DCCPU-V, LCPU : 1 (0x01) Other than the above : 0 (0x00)	Target station QCPU (Q mode), Q12DCCPU-V, LCPU : 1 (0x01) Other than the above : 0 (0x00)	Fixed to 0 (0x00)
ActIONumber[*1]	1023 (0x3FF)	Fixed to 1023 (0x3FF)	Connected station side module I/O address	Connected station side module I/O address	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No.1: 992 (0x3E0) No.2: 993 (0x3E1) No.3: 994 (0x3E2) No.4: 995 (0x3E3)
ActMultiDrop ChannelNumber[*2]	0 (0x00)	Fixed to 0 (0x00)	Multi-drop channel number	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActNetworkNumber	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station side module network number
ActProtocolType	0x04 (PROTOCOL_ SERIAL)	PROTOCOL_USBGOT (0x13)			
ActStationNumber	255 (0xFF)	Fixed to 255 (0xFF)	Fixed to 255 (0xFF)	Fixed to 255 (0xFF)	Target station side module station number
ActThroughNetwork Type	0 (0x00)	MELSECNET/10 is not included. : 0 (0x00) MELSECNET/10 is included. : 1 (0x01)			
ActTimeOut	10000	Any value specified by user in ms units			
ActUnitNumber	0 (0x00)	Fixed to 0 (0x00)	Target station side module station number	Target station side module station number	Fixed to 0 (0x00)
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_LNCPU (0x50)			

*1 : For the I/O address, specify the value of the actual start I/O number divided by 16.

*2 : Specify the following value for the channel number to be multi-drop linked.
0: Default channel of module
1: Channel 1
2: Channel 2

(3) When the connected station is FXCPU

(a) Configuration



(b) Property patterns

Connected station CPU	Relayed network	Relayed station CPU					
FXCPU		QCPU (Q mode)	Q12DC CPU-V	LCPU	QSCPU	Q motion CPU	FXCPU
①	CC IE Control CC IE Field	×	×	×	×	×	×
	MELSECNET/H	×	×	×	×	×	×
	Ethernet	×	×	×	×	×	×
	Serial communication	×	×	×	×	×	×
	CC-Link	×	×	×	×	×	×

○: Accessible (The number in the circle is the property pattern number.), ×: Inaccessible

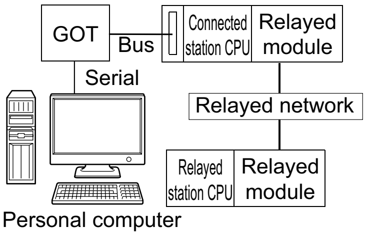
(c) Property list

The following table shows the property settings of communication paths. The setting is not necessary for those properties without description.

Property	Default value	Property pattern
		①
ActCpuType	34 (CPU_Q02CPU)	CPU type corresponding to target station
ActProtocolType	0x04 (PROTOCOL_SERIAL)	PROTOCOL_USBGOT (0x13)
ActTimeOut	10000	Any value specified by user in ms units
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_FXCPU (0x0F)

4.15.3 Personal computer side port: Serial, GOT1000 side port: Serial, CPU side port: Bus connection

(1) Configuration



(2) Property patterns

Connected station CPU		Relayed network	Relayed station CPU					
QCPU (Q mode)*1	Q motion CPU		QCPU (Q mode)*1	Q12DC CPU-V	LCPU	QSCPU	Q motion CPU	FXCPU
①	①*2	CC IE Control CC IE Field	②	②*3	②*4	×	②*3	×
		MELSECNET/H	②	②	×	×	②	×
		Ethernet	②	×	×	×	②	×
		Serial communication	③	×	③	×	③	×
		CC-Link	④	④	④	×	④	×

Connected station CPU	Relayed network	Relayed station CPU					
Q12DCCPU-V		QCPU (Q mode)*1	Q12DC CPU-V	LCPU	QSCPU	Q motion CPU	FXCPU
①	CC IE Control CC IE Field	②	②*3	②*4	×	②*3	×
	MELSECNET/H	②	②	×	×	②	×
	Ethernet	×	×	×	×	×	×
	Serial communication	×	×	×	×	×	×
	CC-Link	④	④	④	×	④	×

○: Accessible (The number in the circle is the property pattern number.), ×: Inaccessible

*1 : Redundant CPU cannot be accessed.

*2 : Relayed stations cannot be accessed via Q motion CPU.

*3 : Q12DCCPU-V and Q motion CPU cannot be accessed using CC-Link IE Field Network because CC-Link IE Field Network is not supported.

*4 : LCPU cannot be accessed using CC-Link IE Controller Network because CC-Link IE Controller Network is not supported.

(3) Property list

The following table shows the property settings of communication paths. The setting is not necessary for those properties without description.

Property	Default value	Property pattern			
		①	②	③	④
ActBaudRate	19200 (BAUDRATE_ 19200)	BAUDRATE_9600, BAUDRATE_19200, BAUDRATE_38400, BAUDRATE_57600, BAUDRATE_115200			
ActCpuType	34 (CPU_Q02CPU)	CPU type corresponding to target station			
ActDestination IONumber	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No.1: 992 (0x3E0) No.2: 993 (0x3E1) No.3: 994 (0x3E2) No.4: 995 (0x3E3)	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No.1: 992 (0x3E0) No.2: 993 (0x3E1) No.3: 994 (0x3E2) No.4: 995 (0x3E3)
ActDidProperty Bit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActDsidProperty Bit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActIntelligent PreferenceBit	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station QCPU (Q mode), Q12DCCPU-V, LCPU : 1 (0x01) Other than the above : 0 (0x00)	Target station QCPU (Q mode), Q12DCCPU-V, LCPU : 1 (0x01) Other than the above : 0 (0x00)
ActIONumber[*1]	1023 (0x3FF)	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No.1: 992 (0x3E0) No.2: 993 (0x3E1) No.3: 994 (0x3E2) No.4: 995 (0x3E3)	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No.1: 992 (0x3E0) No.2: 993 (0x3E1) No.3: 994 (0x3E2) No.4: 995 (0x3E3)	Connected station side module I/O address	Connected station side module I/O address
ActMultiDrop ChannelNumber[*2]	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Multi-drop channel number	Fixed to 0 (0x00)
ActNetwork Number	0 (0x00)	Fixed to 0 (0x00)	Target station side module network number	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActPortNumber	1 (PORT_1)	Personal computer side COM port number			
ActProtocolType	0x04 (PROTOCOL_ SERIAL)	PROTOCOL_SERIAL (0x04)			

Property	Default value	Property pattern			
		①	②	③	④
ActStation Number	255 (0xFF)	Fixed to 255 (0xFF)	Target station side module station number	Fixed to 255 (0xFF)	Fixed to 255 (0xFF)
ActTimeOut*3	10000	Any value specified by user in ms units			
ActUnitNumber	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station side module station number	Target station side module station number
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_A900GOT (0x21)			

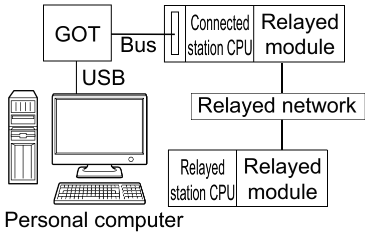
*1 : For the I/O address, specify the value of the actual start I/O number divided by 16.

*2 : Specify the following value for the channel number to be multi-drop linked.
0: Default channel of module
1: Channel 1
2: Channel 2

*3 : When a value between 0 and 5000 [ms] is specified, the value is fixed to 5000ms. When a value greater than 255000ms is specified, the value is fixed to 255000ms.

4.15.4 Personal computer side port: USB, GOT1000 side port: USB, CPU side port: Bus connection

(1) Configuration



(2) Property patterns

Connected station CPU		Relayed network	Relayed station CPU					
QCPU (Q mode)*1	Q motion CPU		QCPU (Q mode)*1	Q12DCCPU-V	LCPU	QSCPU	Q motion CPU	FXCPU
①	①*2	CC IE Control CC IE Field	②	②*3	②*4	×	②*3	×
		MELSECNET/H	②	②	×	×	②	×
		Ethernet	②	×	×	×	②	×
		Serial communication	③	×	③	×	③	×
		CC-Link	④	④	④	×	④	×

Connected station CPU	Relayed network	Relayed station CPU					
FXCPU		QCPU (Q mode)	Q12DCCPU-V	LCPU	QSCPU	Q motion CPU	FXCPU
①	CC IE Control CC IE Field	②	②*3	②*4	×	②*3	×
	MELSECNET/H	②	②	×	×	×	×
	Ethernet	×	×	×	×	×	×
	Serial communication	×	×	×	×	×	×
	CC-Link	④	④	④	×	×	×

○: Accessible (The number in the circle is the property pattern number.), ×: Inaccessible

*1 : Redundant CPU cannot be accessed.

*2 : Relayed stations cannot be accessed via Q motion CPU.

*3 : Q12DCCPU-V and Q motion CPU cannot be accessed using CC-Link IE Field Network because CC-Link IE Field Network is not supported.

*4 : LCPU cannot be accessed using CC-Link IE Controller Network because CC-Link IE Controller Network is not supported.

(3) Property list

The following table shows the property settings of communication paths. The setting is not necessary for those properties without description.

Property	Default value	Property pattern			
		①	②	③	④
ActCpuType	34 (CPU_Q02CPU)	CPU type corresponding to target station			
ActDestination IONumber	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No.1: 992 (0x3E0) No.2: 993 (0x3E1) No.3: 994 (0x3E2) No.4: 995 (0x3E3)	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No.1: 992 (0x3E0) No.2: 993 (0x3E1) No.3: 994 (0x3E2) No.4: 995 (0x3E3)
ActDidProperty Bit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActDsidProperty Bit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActIntelligent PreferenceBit	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station QCPU (Q mode), Q12DCCPU-V, LCPU : 1 (0x01) Other than the above : 0 (0x00)	Target station QCPU (Q mode), Q12DCCPU-V, LCPU : 1 (0x01) Other than the above : 0 (0x00)
ActIONumber[*1]	1023 (0x3FF)	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No.1: 992 (0x3E0) No.2: 993 (0x3E1) No.3: 994 (0x3E2) No.4: 995 (0x3E3)	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No.1: 992 (0x3E0) No.2: 993 (0x3E1) No.3: 994 (0x3E2) No.4: 995 (0x3E3)	Connected station side module I/O address	Connected station side module I/O address
ActMultiDrop ChannelNumber[*2]	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Multi-drop channel number	Fixed to 0 (0x00)
ActNetwork Number	0 (0x00)	Fixed to 0 (0x00)	Target station side module network number	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActProtocolType	0x04 (PROTOCOL_ SERIAL)	PROTOCOL_USBGOT (0x13)			
ActStation Number	255 (0xFF)	Fixed to 255 (0xFF)	Target station side module station number	Fixed to 255 (0xFF)	Fixed to 255 (0xFF)
ActTimeOut[*3]	10000	Any value specified by user in ms units			

Property	Default value	Property pattern			
		①	②	③	④
ActUnitNumber	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station side module station number	Target station side module station number
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_A900GOT (0x21)			

*1 : For the I/O address, specify the value of the actual start I/O number divided by 16.

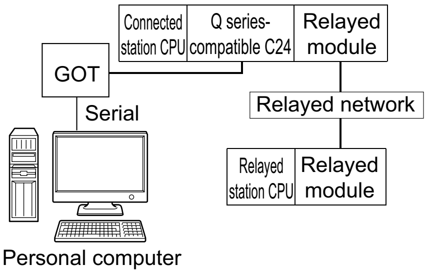
*2 : Specify the following value for the channel number to be multi-drop linked.
0: Default channel of module
1: Channel 1
2: Channel 2

*3 : When a value between 0 and 5000 [ms] is specified, the value is fixed to 5000ms. When a value greater than 255000ms is specified, the value is fixed to 255000ms.

4.15.5 Personal computer side port: Serial, GOT1000 side port: Serial, CPU side port: Q series-compatible C24 or L series-compatible C24

(1) When a relayed module other than the connected station side Q series-compatible C24 exists

(a) Configuration



(b) Property patterns

Connected station CPU		Relayed network	Relayed station CPU					
QCPU (Q mode)*1	Q motion CPU		QCPU (Q mode)*1	Q12DC CPU-V	LCPU	QSCPU	Q motion CPU	FXCPU
①	①*2	CC IE Control CC IE Field	②	②*3	②*4	×	②*3	×
		MELSECNET/H	②	②	×	×	②	×
		Ethernet	②	×	×	×	②	×
		Serial communication	③	×	③	×	③	×
		CC-Link	④	④	④	×	④	×

○: Accessible (The number in the circle is the property pattern number.), ×: Inaccessible

*1 : Redundant CPU cannot be accessed.
*2 : Relayed stations cannot be accessed via Q motion CPU.
*3 : Q12DCCPU-V and Q motion CPU cannot be accessed using CC-Link IE Field Network because CC-Link IE Field Network is not supported.
*4 : LCPU cannot be accessed using CC-Link IE Controller Network because CC-Link IE Controller Network is not supported.

(c) Property list

The following table shows the property settings of communication paths. The setting is not necessary for those properties without description.

Property	Default value	Property pattern			
		①	②	③	④
ActBaudRate	19200 (BAUDRATE_ 19200)	BAUDRATE_9600, BAUDRATE_19200, BAUDRATE_38400, BAUDRATE_57600, BAUDRATE_115200			
ActConnectUnit Number	0 (0x00)	Connected station side module station number			
ActCpuType	34 (CPU_Q02CPU)	CPU type corresponding to target station			

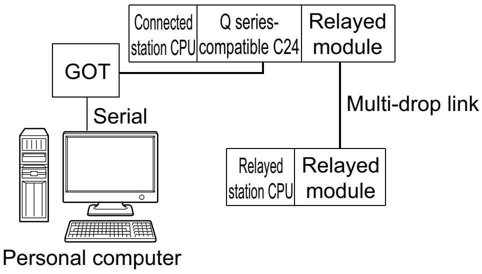
Property	Default value	Property pattern			
		①	②	③	④
ActDestination IONumber	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No.1: 992 (0x3E0) No.2: 993 (0x3E1) No.3: 994 (0x3E2) No.4: 995 (0x3E3)	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No.1: 992 (0x3E0) No.2: 993 (0x3E1) No.3: 994 (0x3E2) No.4: 995 (0x3E3)
ActDidProperty Bit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActDsidProperty Bit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActIntelligent PreferenceBit	0 (0x00)	Fixed to 0 (0x00)			
ActIONumber[*1]	1023 (0x3FF)	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No.1: 992 (0x3E0) No.2: 993 (0x3E1) No.3: 994 (0x3E2) No.4: 995 (0x3E3)	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No.1: 992 (0x3E0) No.2: 993 (0x3E1) No.3: 994 (0x3E2) No.4: 995 (0x3E3)	Connected station side module I/O address	Connected station side module I/O address
ActMultiDrop ChannelNumber[*2]	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Multi-drop channel number	Fixed to 0 (0x00)
ActNetwork Number	0 (0x00)	Fixed to 0 (0x00)	Target station side module network number	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActPortNumber	1 (PORT_1)	Personal computer side COM port number			
ActProtocolType	0x04 (PROTOCOL_ SERIAL)	PROTOCOL_SERIAL (0x04)			
ActStation Number	255 (0xFF)	Fixed to 255 (0xFF)	Target station side module station number	Fixed to 255 (0xFF)	Fixed to 255 (0xFF)
ActThrough NetworkType	0 (0x00)	MELSECNET/10 is not included. : 0 (0x00) MELSECNET/10 is included. : 1 (0x01)			
ActTimeOut	10000	Any value specified by user in ms units			
ActUnitNumber	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station side module station number	Target station side module station number
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_QJ71C24 (0x19)			

[*1] : For the I/O address, specify the value of the actual start I/O number divided by 16.

[*2] : Specify the following value for the channel number to be multi-drop linked.
0: Default channel of module
1: Channel 1
2: Channel 2

(2) When performing multi-drop link on the connected station side Q series-compatible C24 with the relayed module

(a) Configuration



(b) Property patterns

Connected station CPU		Relayed network	Relayed station CPU					
QCPU (Q mode)[*1]			QCPU (Q mode)[*1]	Q12DC CPU-V	LCPU	QSCPU	Q motion CPU	FXCPU
Independent mode[*2]	①	Serial communication	②	×	②	×	×	×

○: Accessible (The number in the circle is the property pattern number.), ×: Inaccessible

*1 : For Redundant CPU, computer link modules on the main base cannot be accessed.

*2 : The independent mode indicates that the parameters are set as follows.
- CH1 side: Operation setting for transmission setting = independent (0), communication protocol setting = 0
- CH2 side: Operation setting for transmission setting = independent (0)

(c) Property list

The following table shows the property settings of communication paths. The setting is not necessary for those properties without description.

Property	Default value	Property pattern	
		①	②
ActBaudRate	19200 (BAUDRATE_19200)	Q series BAUDRATE_9600, BAUDRATE_19200, BAUDRATE_38400, BAUDRATE_57600, BAUDRATE_115200	
ActConnectUnitNumber	0 (0x00)	Connected station side module station number	
ActControl	8 (TRC_DTR_OR_RTS)	Depending on used cable.	
ActCpuType	34 (CPU_Q02CPU)	CPU type corresponding to target station	
ActDestinationIONumber	0 (0x00)	Fixed to 0 (0x00)	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No.1: 992 (0x3E0) No.2: 993 (0x3E1) No.3: 994 (0x3E2) No.4: 995 (0x3E3)
ActDidPropertyBit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)
ActDsidPropertyBit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)
ActIntelligentPreferenceBit	0 (0x00)	Fixed to 0 (0x00)	

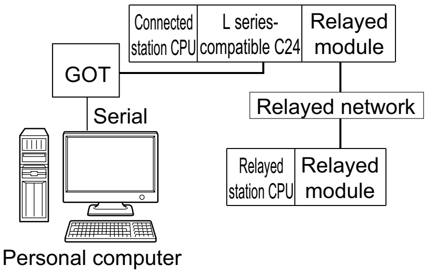
Property	Default value	Property pattern	
		①	②
ActIONumber[*2]	1023 (0x3FF)	For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No.1: 992 (0x3E0) No.2: 993 (0x3E1) No.3: 994 (0x3E2) No.4: 995 (0x3E3)	Connected station side module I/O address
ActMultiDropChannelNumber[*1]	0 (0x00)	Fixed to 0 (0x00)	Multi-drop channel number
ActNetworkNumber	0 (0x00)	Fixed to 0 (0x00)	
ActParity	1 (ODD_PARITY)	Match to the setting of Q series-compatible C24.	
ActPortNumber	1 (PORT_1)	Personal computer side COM port number	
ActProtocolType	0x04 (PROTOCOL_SERIAL)	PROTOCOL_SERIAL (0x04)	
ActStationNumber	255 (0xFF)	Fixed to 255 (0xFF)	
ActThroughNetworkType	0 (0x00)	MELSECNET/10 is not included. : 0 (0x00) MELSECNET/10 is included. : 1 (0x01)	
ActTimeOut	10000	Any value specified by user in ms units	
ActUnitNumber	0 (0x00)	Fixed to 0 (0x00)	Target station side module station number
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_QJ71C24 (0x19)	

*1 : For the I/O address, specify the value of the actual start I/O number divided by 16.

*2 : Specify the following value for the channel number to be multi-drop linked.
0: Default channel of module
1: Channel 1
2: Channel 2

(3) When a relayed module other than the connected station side L series-compatible C24 exists

(a) Configuration



(b) Property patterns

Connected station CPU	Relayed network	Relayed station CPU					
LCPU		QCPU (Q mode)*1	Q12DC CPU-V	LCPU	QSCPU	Q motion CPU	FXCPU
①	CC IE Field*2	④	×	④	×	×	×
	MELSECNET/H	×	×	×	×	×	×
	Ethernet	×	×	×	×	×	×
	Serial communication	②	×	②	×	②	×
	CC-Link	③	③	③	×	③	×

○: Accessible (The number in the circle is the property pattern number.), ×: Inaccessible

*1 : Redundant CPU cannot be accessed.

*2 : LCPU cannot be accessed using CC-Link IE Controller Network because CC-Link IE Controller Network is not supported.

(c) Property list

The following table shows the property settings of communication paths. The setting is not necessary for those properties without description.

Property	Default value	Property pattern			
		①	②	③	④
ActBaudRate	19200 (BAUDRATE_ 19200)	BAUDRATE_9600, BAUDRATE_19200, BAUDRATE_38400, BAUDRATE_57600, BAUDRATE_115200			
ActConnectUnit Number	0 (0x00)	Connected station side module station number			
ActControl	8 (TRC_DTR_OR_ RTS)	Depending on used cable.			
ActCpuType	34 (CPU_Q02CPU)	CPU type corresponding to target station			

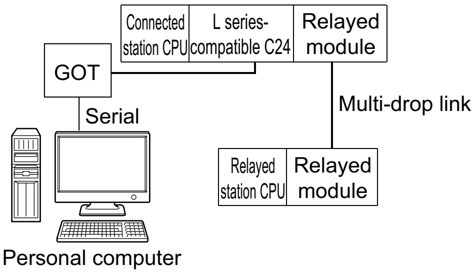
Property	Default value	Property pattern			
		①	②	③	④
ActDestination IONumber	0 (0x00)	Fixed to 0 (0x00)	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No.1: 992 (0x3E0) No.2: 993 (0x3E1) No.3: 994 (0x3E2) No.4: 995 (0x3E3)	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No.1: 992 (0x3E0) No.2: 993 (0x3E1) No.3: 994 (0x3E2) No.4: 995 (0x3E3)	Fixed to 0 (0x00)
ActDidPropertyBit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Fixed to 1 (0x01)
ActDsidPropertyBit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Fixed to 1 (0x01)
ActIntelligent PreferenceBit	0 (0x00)	Fixed to 0 (0x00)			
ActIONumber*1	1023 (0x3FF)	Fixed to 1023 (0x3FF)	Connected station side module I/O address	Connected station side module I/O address	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No.1: 992 (0x3E0) No.2: 993 (0x3E1) No.3: 994 (0x3E2) No.4: 995 (0x3E3)
ActMultiDrop ChannelNumber*2	0 (0x00)	Fixed to 0 (0x00)	Multi-drop channel number	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActNetworkNumber	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station side module network number
ActParity	1 (ODD_PARITY)	ODD_PARITY			
ActPortNumber	1 (PORT_1)	Personal computer side COM port number			
ActProtocolType	0x04 (PROTOCOL_ SERIAL)	PROTOCOL_SERIAL (0x04)			
ActStationNumber	255 (0xFF)	Fixed to 255 (0xFF)	Fixed to 255 (0xFF)	Fixed to 255 (0xFF)	Target station side module station number
ActThrough NetworkType	0 (0x00)	MELSECNET/10 is not included. : 0 (0x00) MELSECNET/10 is included. : 1 (0x01)			
ActTimeOut	10000	Any value specified by user in ms units			
ActUnitNumber	0 (0x00)	Fixed to 0 (0x00)	Target station side module station number	Target station side module station number	Fixed to 0 (0x00)
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_LJ71C24 (0x54)			

*1 : For the I/O address, specify the value of the actual start I/O number divided by 16.

*2 : Specify the following value for the channel number to be multi-drop linked.
0: Default channel of module
1: Channel 1
2: Channel 2

(4) When performing multi-drop link on the connected station side L series-compatible C24 with the relayed module

(a) Configuration



(b) Property patterns

Connected station CPU		Relayed network	Relayed station CPU					
LCPU			QCPU (Q mode)*1	Q12DC CPU-V	LCPU	QSCPU	Q motion CPU	FXCPU
Independent mode*2	①	Serial communication	②	×	②	×	×	×

○: Accessible (The number in the circle is the property pattern number.), ×: Inaccessible

*1 : Redundant CPU cannot be accessed.

*2 : The independent mode indicates that the parameters are set as follows.

- CH1 side: Operation setting for transmission setting = independent (0), communication protocol setting = 0
- CH2 side: Operation setting for transmission setting = independent (0)

(c) Property list

The following table shows the property settings of communication paths. The setting is not necessary for those properties without description.

Property	Default value	Property pattern	
		①	②
ActBaudRate	19200 (BAUDRATE_19200)	BAUDRATE_9600, BAUDRATE_19200, BAUDRATE_38400, BAUDRATE_57600, BAUDRATE_115200	
ActConnectUnitNumber	0 (0x00)	Connected station side module station number	
ActControl	8 (TRC_DTR_OR_RTS)	Depending on used cable.	
ActCpuType	34 (CPU_Q02CPU)	CPU type corresponding to target station	
ActDestinationIONumber	0 (0x00)	Fixed to 0 (0x00)	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No.1: 992 (0x3E0) No.2: 993 (0x3E1) No.3: 994 (0x3E2) No.4: 995 (0x3E3)
ActDidPropertyBit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)
ActDsidPropertyBit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)

Property	Default value	Property pattern	
		①	②
ActIntelligentPreferenceBit	0 (0x00)	Fixed to 0 (0x00)	
ActIONumber[*1]	1023 (0x3FF)	Fixed to 1023 (0x3FF)	Connected station side module I/O address
ActMultiDropChannelNumber[*2]	0 (0x00)	Fixed to 0 (0x00)	Multi-drop channel number
ActNetworkNumber	0 (0x00)	Fixed to 0 (0x00)	
ActParity	1 (ODD_PARITY)	Match to the setting of L series-compatible C24.	
ActPortNumber	1 (PORT_1)	Personal computer side COM port number	
ActProtocolType	0x04 (PROTOCOL_SERIAL)	PROTOCOL_SERIAL (0x04)	
ActStationNumber	255 (0xFF)	Fixed to 255 (0xFF)	
ActThroughNetworkType	0 (0x00)	MELSECNET/10 is not included. : 0 (0x00) MELSECNET/10 is included. : 1 (0x01)	
ActTimeOut	10000	Any value specified by user in ms units	
ActUnitNumber	0 (0x00)	Fixed to 0 (0x00)	Target station side module station number
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_LJ71C24 (0x54)	

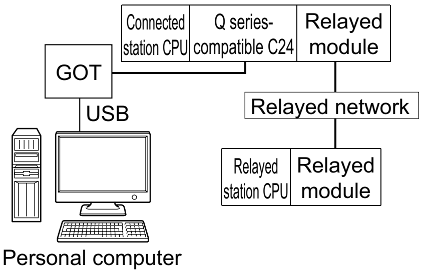
*1 : For the I/O address, specify the value of the actual start I/O number divided by 16.

*2 : Specify the following value for the channel number to be multi-drop linked.
0: Default channel of module
1: Channel 1
2: Channel 2

4.15.6 Personal computer side port: USB, GOT1000 side port: USB, CPU side port: Q series-compatible C24 or L series-compatible C24

(1) When a relayed module other than the connected station side Q series-compatible C24 exists

(a) Configuration



(b) Property patterns

Connected station CPU		Relayed network	Relayed station CPU					
QCPU (Q mode)*1	Q motion CPU		QCPU (Q mode)*1	Q12DC CPU-V	LCPU	QSCPU	Q motion CPU	FXCPU
①	①*2	CC IE Control CC IE Field	②	②*3	②*4	×	②*3	×
		MELSECNET/H	②	②	×	×	②	×
		Ethernet	②	×	×	×	②	×
		Serial communication	③	×	③	×	③	×
		CC-Link	④	④	④	×	④	×

○: Accessible (The number in the circle is the property pattern number.), ×: Inaccessible

*1 : Redundant CPU cannot be accessed.

*2 : Relayed stations cannot be accessed via Q motion CPU.

*3 : Q12DCCPU-V and Q motion CPU cannot be accessed using CC-Link IE Field Network because CC-Link IE Field Network is not supported.

*4 : LCPU cannot be accessed using CC-Link IE Controller Network because CC-Link IE Controller Network is not supported.

(c) Property list

The following table shows the property settings of communication paths. The setting is not necessary for those properties without description.

Property	Default value	Property pattern			
		①	②	③	④
ActConnectUnit Number	0 (0x00)	Connected station side module station number			
ActCpuType	34 (CPU_Q02CPU)	CPU type corresponding to target station			
ActDestination IONumber	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No.1: 992 (0x3E0) No.2: 993 (0x3E1) No.3: 994 (0x3E2) No.4: 995 (0x3E3)	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No.1: 992 (0x3E0) No.2: 993 (0x3E1) No.3: 994 (0x3E2) No.4: 995 (0x3E3)
ActDidProperty Bit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActDsidProperty Bit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActIntelligent PreferenceBit	0 (0x00)	Fixed to 0 (0x00)			
ActIONumber[*1]	1023 (0x3FF)	For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No.1: 992 (0x3E0) No.2: 993 (0x3E1) No.3: 994 (0x3E2) No.4: 995 (0x3E3)	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No.1: 992 (0x3E0) No.2: 993 (0x3E1) No.3: 994 (0x3E2) No.4: 995 (0x3E3)	Connected station side module I/O address	Connected station side module I/O address
ActMultiDrop ChannelNumber[*2]	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Multi-drop channel number	Fixed to 0 (0x00)
ActNetwork Number	0 (0x00)	Fixed to 0 (0x00)	Target station side module network number	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActProtocolType	0x04 (PROTOCOL_ SERIAL)	PROTOCOL_USBGOT (0x13)			
ActStation Number	255 (0xFF)	Fixed to 255 (0xFF)	Target station side module station number	Fixed to 255 (0xFF)	Fixed to 255 (0xFF)
ActThrough NetworkType	0 (0x00)	MELSECNET/10 is not included. : 0 (0x00) MELSECNET/10 is included. : 1 (0x01)			
ActTimeOut	10000	Any value specified by user in ms units			

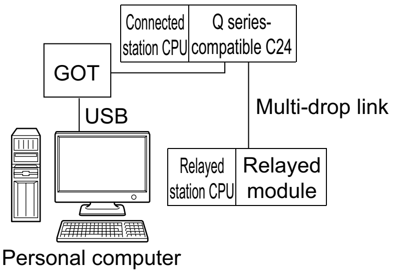
Property	Default value	Property pattern			
		①	②	③	④
ActUnitNumber	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station side module station number	Target station side module station number
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_QJ71C24 (0x19)			

*1 : For the I/O address, specify the value of the actual start I/O number divided by 16.

*2 : Specify the following value for the channel number to be multi-drop linked.
0: Default channel of module
1: Channel 1
2: Channel 2

(2) When performing multi-drop link on the connected station side Q series-compatible C24 with the relayed module

(a) Configuration



(b) Property patterns

Connected station CPU		Relayed network	Relayed station CPU					
QCPU (Q mode)[*1]			QCPU (Q mode)[*1]	Q12DC CPU-V	LCPU	QSCPU	Q motion CPU	FXCPU
Independent mode[*2]	①	Serial communication	②	×	②	×	×	×

○: Accessible (The number in the circle is the property pattern number.), ×: Inaccessible

*1 : Redundant CPU cannot be accessed.

*2 : The independent mode indicates that the parameters are set as follows.

- CH1 side: Operation setting for transmission setting = independent (0), communication protocol setting = 0
- CH2 side: Operation setting for transmission setting = independent (0)

(c) Property list

The following table shows the property settings of communication paths. The setting is not necessary for those properties without description.

Property	Default value	Property pattern	
		①	②
ActConnectUnitNumber	0 (0x00)	Connected station side module station number	
ActCpuType	34 (CPU_Q02CPU)	CPU type corresponding to target station	
ActDestinationIONumber	0 (0x00)	Fixed to 0 (0x00)	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No.1: 992 (0x3E0) No.2: 993 (0x3E1) No.3: 994 (0x3E2) No.4: 995 (0x3E3)
ActDidPropertyBit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)
ActDsidPropertyBit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)
ActIntelligentPreferenceBit	0 (0x00)	Fixed to 0 (0x00)	

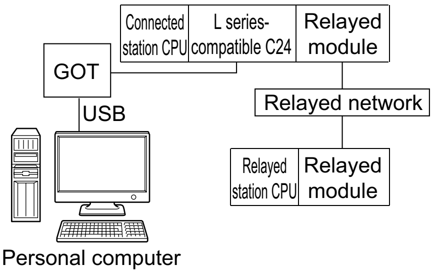
Property	Default value	Property pattern	
		①	②
ActIONumber[*1]	1023 (0x3FF)	For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No.1: 992 (0x3E0) No.2: 993 (0x3E1) No.3: 994 (0x3E2) No.4: 995 (0x3E3)	Connected station side module I/O address
ActMultiDropChannelNumber[*2]	0 (0x00)	Fixed to 0 (0x00)	Multi-drop channel number
ActNetworkNumber	0 (0x00)	Fixed to 0 (0x00)	
ActProtocolType	0x04 (PROTOCOL_SERIAL)	PROTOCOL_USBGOT (0x13)	
ActStationNumber	255 (0xFF)	Fixed to 255 (0xFF)	
ActThroughNetworkType	0 (0x00)	MELSECNET/10 is not included. : 0 (0x00) MELSECNET/10 is included. : 1 (0x01)	
ActTimeOut	10000	Any value specified by user in ms units	
ActUnitNumber	0(0x00)	Fixed to 0 (0x00)	Target station side module station number
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_QJ71C24 (0x19)	

*1 : For the I/O address, specify the value of the actual start I/O number divided by 16.

*2 : Specify the following value for the channel number to be multi-drop linked.
0: Default channel of module
1: Channel 1
2: Channel 2

(3) When a relayed module other than the connected station side L series-compatible C24 exists

(a) Configuration



(b) Property patterns

Connected station CPU	Relayed network	Relayed station CPU					
LCPU		QCPU (Q mode)[*1]	Q12DC CPU-V	LCPU	QSCPU	Q motion CPU	FXCPU
①	CC IE Field[*2]	④	×	④	×	×	×
	MELSECNET/H	×	×	×	×	×	×
	Ethernet	×	×	×	×	×	×
	Serial communication	②	×	②	×	②	×
	CC-Link	③	③	③	×	③	×

○: Accessible (The number in the circle is the property pattern number.), ×: Inaccessible

*1 : Redundant CPU cannot be accessed.

*2 : LCPU cannot be accessed using CC-Link IE Controller Network because CC-Link IE Controller Network is not supported.

(c) Property list

The following table shows the property settings of communication paths. The setting is not necessary for those properties without description.

Property	Default value	Property pattern			
		①	②	③	④
ActConnectUnit Number	0 (0x00)	Connected station side module station number			
ActCpuType	34 (CPU_Q02CPU)	CPU type corresponding to target station			
ActDestination IONumber	0 (0x00)	Fixed to 0 (0x00)	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No.1: 992 (0x3E0) No.2: 993 (0x3E1) No.3: 994 (0x3E2) No.4: 995 (0x3E3)	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No.1: 992 (0x3E0) No.2: 993 (0x3E1) No.3: 994 (0x3E2) No.4: 995 (0x3E3)	Fixed to 0 (0x00)
ActDidPropertyBit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Fixed to 1 (0x01)

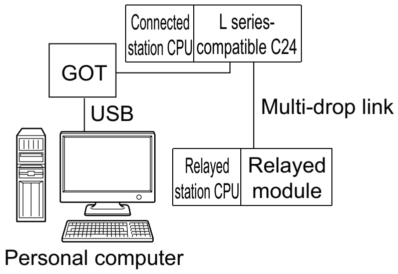
Property	Default value	Property pattern			
		①	②	③	④
ActDsidPropertyBit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Fixed to 1 (0x01)
ActIntelligent PreferenceBit	0 (0x00)	Fixed to 0 (0x00)			
ActIONumber*1	1023 (0x3FF)	Fixed to 1023 (0x3FF)	Connected station side module I/O address	Connected station side module I/O address	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No.1: 992 (0x3E0) No.2: 993 (0x3E1) No.3: 994 (0x3E2) No.4: 995 (0x3E3)
ActMultiDrop ChannelNumber*2	0 (0x00)	Fixed to 0 (0x00)	Multi-drop channel number	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActNetworkNumber	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station side module network number
ActProtocolType	0x04 (PROTOCOL_ SERIAL)	PROTOCOL_USBGOT (0x13)			
ActStationNumber	255 (0xFF)	Fixed to 255 (0xFF)	Fixed to 255 (0xFF)	Fixed to 255 (0xFF)	Target station side module station number
ActThroughNetwork Type	0 (0x00)	MELSECNET/10 is not included. : 0 (0x00) MELSECNET/10 is included. : 1 (0x01)			
ActTimeOut	10000	Any value specified by user in ms units			
ActUnitNumber	0 (0x00)	Fixed to 0 (0x00)	Target station side module station number	Target station side module station number	Fixed to 0 (0x00)
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_LJ71C24 (0x54)			

*1 : For the I/O address, specify the value of the actual start I/O number divided by 16.

*2 : Specify the following value for the channel number to be multi-drop linked.
0: Default channel of module
1: Channel 1
2: Channel 2

(4) When performing multi-drop link on the connected station side L series-compatible C24 with the relayed module

(a) Configuration



(b) Property patterns

Connected station CPU		Relayed network	Relayed station CPU					
LCPU			QCPU (Q mode)*1	Q12DC CPU-V	LCPU	QSCPU	Q motion CPU	FXCPU
Independent mode*2	①	Serial communication	②	×	②	×	×	×

○: Accessible (The number in the circle is the property pattern number.), ×: Inaccessible

*1 : Redundant CPU cannot be accessed.

*2 : The independent mode indicates that the parameters are set as follows.
- CH1 side: Operation setting for transmission setting = independent (0), communication protocol setting = 0
- CH2 side: Operation setting for transmission setting = independent (0)

(c) Property list

The following table shows the property settings of communication paths. The setting is not necessary for those properties without description.

Property	Default value	Property pattern	
		①	②
ActConnectUnitNumber	0 (0x00)	Connected station side module station number	
ActCpuType	34 (CPU_Q02CPU)	CPU type corresponding to target station	
ActDestinationIONumber	0 (0x00)	Fixed to 0 (0x00)	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No.1: 992 (0x3E0) No.2: 993 (0x3E1) No.3: 994 (0x3E2) No.4: 995 (0x3E3)
ActDidPropertyBit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)
ActDsidPropertyBit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)
ActIntelligentPreferenceBit	0 (0x00)	Fixed to 0 (0x00)	
ActIONumber*1	1023 (0x3FF)	Fixed to 1023 (0x3FF)	Connected station side relayed module I/O address

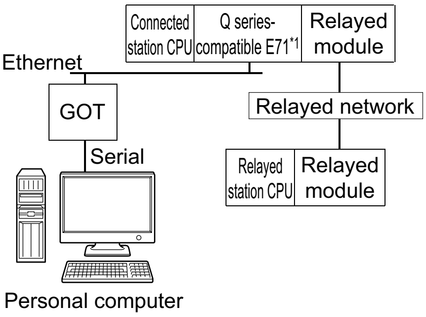
Property	Default value	Property pattern	
		①	②
ActMultiDropChannelNumber[*2]	0 (0x00)	Fixed to 0 (0x00)	Multi-drop channel number
ActNetworkNumber	0 (0x00)	Fixed to 0 (0x00)	
ActProtocolType	0x04 (PROTOCOL_SERIAL)	PROTOCOL_USBGOT (0x13)	
ActStationNumber	255 (0xFF)	Fixed to 255 (0xFF)	
ActThroughNetworkType	0 (0x00)	MELSECNET/10 is not included. : 0 (0x00) MELSECNET/10 is included. : 1 (0x01)	
ActTimeOut	10000	Any value specified by user in ms units	
ActUnitNumber	0 (0x00)	Fixed to 0 (0x00)	Target station side module station number
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_LJ71C24 (0x54)	

*1 : For the I/O address, specify the value of the actual start I/O number divided by 16.

*2 : Specify the following value for the channel number to be multi-drop linked.
0: Default channel of module
1: Channel 1
2: Channel 2

4.15.7 Personal computer side port: Serial, GOT1000 side port: Serial, CPU side port: Q series-compatible E71

(1) Configuration



*1 : The communication cannot be established when a password is set to the connected station side Q series-compatible E71.

(2) Property patterns

Connected station CPU	Relayed network	Relayed station CPU					
QCPU (Q mode)*1		QCPU (Q mode)*1	Q12DC CPU-V	LCPU	QSCPU	Q motion CPU	FXCPU
①	CC IE Control CC IE Field	②	②*2	②*3	×	×	×
	MELSECNET/H	②	②	×	×	×	×
	Ethernet	②	×	×	×	×	×
	Serial communication	③	×	③	×	×	×
	CC-Link	④	④	④	×	×	×

○: Accessible (The number in the circle is the property pattern number.), ×: Inaccessible

*1 : Redundant CPU cannot be accessed.
*2 : Q12DCCPU-V and Q motion CPU cannot be accessed using CC-Link IE Field Network because CC-Link IE Field Network is not supported.
*3 : LCPU cannot be accessed using CC-Link IE Controller Network because CC-Link IE Controller Network is not supported.

(3) Property list

The following table shows the property settings of communication paths. The setting is not necessary for those properties without description.

Property	Default value	Property pattern			
		①	②	③	④
ActBaudRate	19200 (BAUDRATE_ 19200)	BAUDRATE_9600, BAUDRATE_19200, BAUDRATE_38400, BAUDRATE_57600, BAUDRATE_115200			
ActConnectUnit Number	0 (0x00)	Fixed to 0 (0x00)	Connected station side module station number	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActCpuType	34 (CPU_Q02CPU)	CPU type corresponding to target station			

Property	Default value	Property pattern			
		①	②	③	④
ActDestination IONumber	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No.1: 992 (0x3E0) No.2: 993 (0x3E1) No.3: 994 (0x3E2) No.4: 995 (0x3E3)	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No.1: 992 (0x3E0) No.2: 993 (0x3E1) No.3: 994 (0x3E2) No.4: 995 (0x3E3)
ActDestination PortNumber	0 (0x00)	5001			
ActDidProperty Bit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActDsidProperty Bit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActHostAddress	1.1.1.1	Host name or IP address of connected station side module			
ActIONumber[*1]	1023 (0x3FF)	For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No.1: 992 (0x3E0) No.2: 993 (0x3E1) No.3: 994 (0x3E2) No.4: 995 (0x3E3)	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No.1: 992 (0x3E0) No.2: 993 (0x3E1) No.3: 994 (0x3E2) No.4: 995 (0x3E3)	Connected station side relayed module I/O address	Connected station side relayed module I/O address
ActMultiDrop ChannelNumber[*2]	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Multi-drop channel number	Fixed to 0 (0x00)
ActNetwork Number[*3]	0 (0x00)	Target station side module network number	Target station side module network number	Connected station side Q series-compatible E71 network number	Connected station side Q series-compatible E71 network number
ActPortNumber	1 (PORT_1)	Personal computer side COM port number			
ActProtocolType	0x04 (PROTOCOL_SERIAL)	PROTOCOL_SERIAL (0x04)			
ActSource NetworkNumber	0 (0x00)	GOT side network number			
ActSource StationNumber[*4]	0 (0x00)	GOT side station number			
ActStation Number[*3]	255 (0xFF)	Target station side module station number	Target station side module station number	Connected station side Q series-compatible E71 station number	Connected station side Q series-compatible E71 station number
ActThrough NetworkType	0 (0x00)	MELSECNET/10 is not included. : 0 (0x00) MELSECNET/10 is included. : 1 (0x01)			
ActTimeOut	10000	Any value specified by user in ms units			
ActUnitNumber	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station side module station number	Target station side module station number

Property	Default value	Property pattern			
		①	②	③	④
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_GOT_QJ71E71 (0x40)			

*1 : For the I/O address, specify the value of the actual start I/O number divided by 16.

*2 : Specify the following value for the channel number to be multi-drop linked.

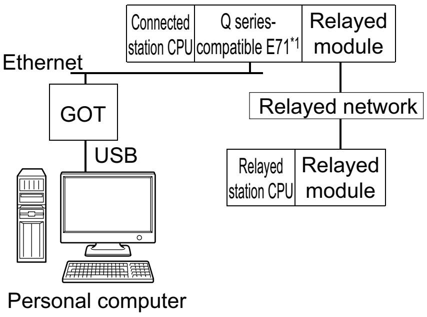
0: Default channel of module
1: Channel 1
2: Channel 2

*3 : For the property pattern of ① or ② , specify the value set in the target station side parameter for ActNetworkNumber and ActStationNumber.

*4 : Specify the station number on the GOT side to avoid setting the same station number as set to the Q series-compatible E71 on the same Ethernet loop.

4.15.8 Personal computer side port: USB, GOT1000 side port: USB, CPU side port: Q series-compatible E71

(1) Configuration



(2) Property patterns

Connected station CPU	Relayed network	Relayed station CPU					
QCPU (Q mode)*1		QCPU (Q mode)*1	Q12DC CPU-V	LCPU	QSCPU	Q motion CPU	FXCPU
①	CC IE Control CC IE Field	②	②*2	②*3	×	×	×
	MELSECNET/H	②	②	×	×	×	×
	Ethernet	②	×	×	×	×	×
	Serial communication	③	×	③	×	×	×
	CC-Link	④	④	④	×	×	×

○: Accessible (The number in the circle is the property pattern number.), ×: Inaccessible

*1 : Redundant CPU cannot be accessed.

*2 : Q12DCCPU-V and Q motion CPU cannot be accessed using CC-Link IE Field Network because CC-Link IE Field Network is not supported.

*3 : LCPU cannot be accessed using CC-Link IE Controller Network because CC-Link IE Controller Network is not supported.

(3) Property list

The following table shows the property settings of communication paths. The setting is not necessary for those properties without description.

Property	Default value	Property pattern			
		①	②	③	④
ActConnectUnit Number	0 (0x00)	Fixed to 0 (0x00)	Connected station side module station number	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActCpuType	34 (CPU_Q02CPU)	CPU type corresponding to target station			
ActDestination IONumber	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No.1: 992 (0x3E0) No.2: 993 (0x3E1) No.3: 994 (0x3E2) No.4: 995 (0x3E3)	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No.1: 992 (0x3E0) No.2: 993 (0x3E1) No.3: 994 (0x3E2) No.4: 995 (0x3E3)
ActDestination PortNumber	0 (0x00)	5001			
ActDidProperty Bit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActDsidProperty Bit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActHostAddress	1.1.1.1	Host name or IP address of connected station side module			
ActIONumber*1	1023 (0x3FF)	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No.1: 992 (0x3E0) No.2: 993 (0x3E1) No.3: 994 (0x3E2) No.4: 995 (0x3E3)	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No.1: 992 (0x3E0) No.2: 993 (0x3E1) No.3: 994 (0x3E2) No.4: 995 (0x3E3)	Connected station side relayed module I/O address	Connected station side relayed module I/O address
ActMultiDrop ChannelNumber*2	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Multi-drop channel number	Fixed to 0 (0x00)
ActNetwork Number*3	0 (0x00)	Target station side module network number	Target station side module network number	Connected station side Q series-compatible E71 network number	Connected station side Q series-compatible E71 network number
ActProtocolType	0x04 (PROTOCOL_ SERIAL)	PROTOCOL_USBGOT (0x13)			
ActSource NetworkNumber	0 (0x00)	GOT side network number			
ActSource StationNumber*4	0 (0x00)	GOT side station number			

Property	Default value	Property pattern			
		①	②	③	④
ActStation Number[*3]	255 (0xFF)	Target station side module station number	Target station side module station number	Connected station side Q series-compatible E71 station number	Connected station side Q series-compatible E71 station number
ActThrough NetworkType	0 (0x00)	MELSECNET/10 is not included. : 0 (0x00) MELSECNET/10 is included. : 1 (0x01)			
ActTimeOut	10000	Any value specified by user in ms units			
ActUnitNumber	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station side module station number	Target station side module station number
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_GOT_QJ71E71 (0x40)			

*1 : For the I/O address, specify the value of the actual start I/O number divided by 16.

*2 : Specify the following value for the channel number to be multi-drop linked.
0: Default channel of module
1: Channel 1
2: Channel 2

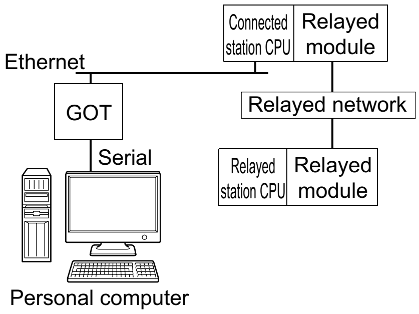
*3 : For the property pattern of ① or ② , specify the value set in the target station side parameter for ActNetworkNumber and ActStationNumber.

*4 : Specify the station number on the GOT side to avoid setting the same station number as set to the Q series-compatible E71 on the same Ethernet loop.

4.15.9 Personal computer side port: Serial, GOT1000 side port: Serial, CPU side port: Ethernet port

(1) When the connected station is QnUDE(H)CPU or Q12DCCPU-V

(a) Configuration



(b) Property patterns

Connected station CPU*1	Relayed network	Relayed station CPU					
QnUDE(H)CPU		QCPU (Q mode)*2	Q12DC CPU-V	LCPU	QSCPU	Q motion CPU	FXCPU
①	CC IE Control CC IE Field	②	②*3	②*4	×	×	×
	MELSECNET/H	②	②	×	×	×	×
	Ethernet	②	×	×	×	×	×
	Serial communication	③	×	③	×	×	×
	CC-Link	④	④	④	×	×	×

Connected station CPU*1	Relayed network	Relayed station CPU					
Q12DCCPU-V		QCPU (Q mode)*2	Q12DC CPU-V	LCPU	QSCPU	Q motion CPU	FXCPU
①	CC IE Control CC IE Field	②	②*3	②*4	×	×	×
	MELSECNET/H	②	②	×	×	×	×
	Ethernet	×	×	×	×	×	×
	Serial communication	×	×	×	×	×	×
	CC-Link	④	④	④	×	×	×

○: Accessible (The number in the circle is the property pattern number.), ×: Inaccessible

*1 : The communication cannot be established when a password is set to the connected station CPU.

*2 : Redundant CPU cannot be accessed.

*3 : Q12DCCPU-V and Q motion CPU cannot be accessed using CC-Link IE Field Network because CC-Link IE Field Network is not supported.

*4 : LCPU cannot be accessed using CC-Link IE Controller Network because CC-Link IE Controller Network is not supported.

(c) Property list

The following table shows the property settings of communication paths. The setting is not necessary for those properties without description.

Property	Default value	Property pattern			
		①	②[*1]	③	④
ActBaudRate	19200 (BAUDRATE_19200)	BAUDRATE_9600, BAUDRATE_19200, BAUDRATE_38400, BAUDRATE_57600, BAUDRATE_115200			
ActCpuType	34 (CPU_Q02CPU)	CPU type corresponding to target station			
ActDestinationIONumber	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No.1: 992 (0x3E0) No.2: 993 (0x3E1) No.3: 994 (0x3E2) No.4: 995 (0x3E3)	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No.1: 992 (0x3E0) No.2: 993 (0x3E1) No.3: 994 (0x3E2) No.4: 995 (0x3E3)
ActDidPropertyBit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActDsidPropertyBit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActHostAddress	1.1.1.1	Host name or IP address of connected station side module			
ActIntelligentPreferenceBit	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station QCPU (Q mode), Q12DCCPU-V, LCPU : 1 (0x01) Other than the above : 0 (0x00)	Target station QCPU (Q mode), Q12DCCPU-V, LCPU : 1 (0x01) Other than the above : 0 (0x00)
ActIONumber[*2]	1023 (0x3FF)	For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No.1: 992 (0x3E0) No.2: 993 (0x3E1) No.3: 994 (0x3E2) No.4: 995 (0x3E3)	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No.1: 992 (0x3E0) No.2: 993 (0x3E1) No.3: 994 (0x3E2) No.4: 995 (0x3E3)	Connected station side module I/O address	Connected station side module I/O address
ActMultiDropChannelNumber[*3]	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Multi-drop channel number	Fixed to 0 (0x00)
ActNetworkNumber[*4]	0 (0x00)	Fixed to 0 (0x00)	Target station side module network number	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActPortNumber	1 (PORT_1)	Personal computer side COM port number			
ActProtocolType	0x04 (PROTOCOL_SERIAL)	PROTOCOL_SERIAL (0x04)			

Property	Default value	Property pattern			
		①	②[*1]	③	④
ActStation Number[*4]	255 (0xFF)	Fixed to 255 (0xFF)	Target station side module station number	Fixed to 255 (0xFF)	Fixed to 255 (0xFF)
ActThrough NetworkType	0 (0x00)	MELSECNET/10 is not included. : 0 (0x00) MELSECNET/10 is included. : 1 (0x01)			
ActTimeOut	10000	Any value specified by user in ms units			
ActUnitNumber	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station side module station number	Target station side module station number
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_GOT_QNETHER (0x41)			

*1 : Note the following considerations when accessing via Ethernet module (Q series-compatible E71).

- For ActNetworkNumber and ActStationNumber, specify the value set in the parameter setting of the target station side Q series-compatible E71.
- Set "Station No. ⇔ IP information" in the parameter setting of Q series-compatible E71. When setting the parameter, specify the setting other than the automatic response system (any of the IP address calculation system, table conversion system, and combined system) for "Station No. ⇔ IP information system".

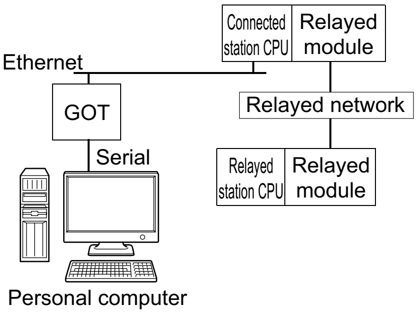
*2 : For the I/O address, specify the value of the actual start I/O number divided by 16.

*3 : Specify the following value for the channel number to be multi-drop linked.
0: Default channel of module
1: Channel 1
2: Channel 2

*4 : For the property pattern of ②, specify the value set in the target station side parameter for ActNetworkNumber and ActStationNumber.

(2) When the connected station is LCPU

(a) Configuration



(b) Property patterns

Connected station CPU[*1]	Relayed network	Relayed station CPU					
LCPU		QCPU (Q mode)[*2]	Q12DC CPU-V	LCPU	QSCPU	Q motion CPU	FXCPU
①	CC IE Field[*3]	④	×	④	×	×	×
	MELSECNET/H	×	×	×	×	×	×
	Ethernet	×	×	×	×	×	×
	Serial communication	②	×	②	×	×	×
	CC-Link	③	③	③	×	×	×

○: Accessible (The number in the circle is the property pattern number.), ×: Inaccessible

*1 : The communication cannot be established when a password is set to the connected station CPU.

*2 : Redundant CPU cannot be accessed.

*3 : LCPU cannot be accessed using CC-Link IE Controller Network because CC-Link IE Controller Network is not supported.

(c) Property list

The following table shows the property settings of communication paths. The setting is not necessary for those properties without description.

Property	Default value	Property pattern			
		①	②	③	④
ActBaudRate	19200 (BAUDRATE_ 19200)	BAUDRATE_9600, BAUDRATE_19200, BAUDRATE_38400, BAUDRATE_57600, BAUDRATE_115200			
ActControl	8 (TRC_DTR_OR_ RTS)	Depending on used cable.			
ActCpuType	34 (CPU_Q02CPU)	CPU type corresponding to target station			

Property	Default value	Property pattern			
		①	②	③	④
ActDestination IONumber	0 (0x00)	Fixed to 0 (0x00)	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No.1: 992 (0x3E0) No.2: 993 (0x3E1) No.3: 994 (0x3E2) No.4: 995 (0x3E3)	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No.1: 992 (0x3E0) No.2: 993 (0x3E1) No.3: 994 (0x3E2) No.4: 995 (0x3E3)	Fixed to 0 (0x00)
ActDidPropertyBit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Fixed to 1 (0x01)
ActDsidPropertyBit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Fixed to 1 (0x01)
ActHostAddress	1.1.1.1	Host name or IP address of connected station side module			
ActIntelligent PreferenceBit	0 (0x00)	Fixed to 0 (0x00)	Target station QCPU (Q mode), Q12DCCPU-V, LCPU : 1 (0x01) Other than the above : 0 (0x00)	Target station QCPU (Q mode), Q12DCCPU-V, LCPU : 1 (0x01) Other than the above : 0 (0x00)	Fixed to 0 (0x00)
ActIONumber*1	1023 (0x3FF)	Fixed to 1023 (0x3FF)	Connected station side module I/O address	Connected station side module I/O address	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No.1: 992 (0x3E0) No.2: 993 (0x3E1) No.3: 994 (0x3E2) No.4: 995 (0x3E3)
ActMultiDrop ChannelNumber*2	0 (0x00)	Fixed to 0 (0x00)	Multi-drop channel number	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActNetworkNumber	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station side module network number
ActPortNumber	1 (PORT_1)	Personal computer side COM port number			
ActProtocolType	0x04 (PROTOCOL_ SERIAL)	PROTOCOL_SERIAL (0x04)			
ActStationNumber	255 (0xFF)	Fixed to 255 (0xFF)	Fixed to 255 (0xFF)	Fixed to 255 (0xFF)	Target station side Module Station number
ActThroughNetwork Type	0 (0x00)	MELSECNET/10 is not included. : 0 (0x00) MELSECNET/10 is included. : 1 (0x01)			
ActTimeOut	10000	Any value specified by user in ms units			
ActUnitNumber	0 (0x00)	Fixed to 0 (0x00)	Target station side Module Station number	Target station side Module Station number	Fixed to 0 (0x00)
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_GOT_LNETHER (0x55)			

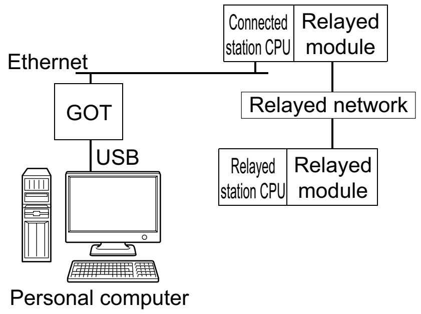
*1 : For the I/O address, specify the value of the actual start I/O number divided by 16.

*2 : Specify the following value for the channel number to be multi-drop linked.
0: Default channel of module
1: Channel 1
2: Channel 2

4.15.10 Personal computer side port: USB, GOT1000 side port: USB, CPU side port: Ethernet port

(1) When the connected station is QCPU (Q mode) or Q12DCCPU-V

(a) Configuration



(b) Property patterns

Connected station CPU*1	Relayed network	Relayed station CPU					
QnUDE(H)CPU		QCPU (Q mode)*2	Q12DC CPU-V	LCPU	QSCPU	Q motion CPU	FXCPU
①	CC IE Control CC IE Field	②	②*3	②*4	×	×	×
	MELSECNET/H	②	②	×	×	×	×
	Ethernet	②	×	×	×	×	×
	Serial communication	③	×	③	×	×	×
	CC-Link	④	④	④	×	×	×

Connected station CPU*1	Relayed network	Relayed station CPU					
Q12DCCPU-V		QCPU (Q mode)*2	Q12DC CPU-V	LCPU	QSCPU	Q motion CPU	FXCPU
①	CC IE Control CC IE Field	②	②*3	②*4	×	×	×
	MELSECNET/H	②	②	×	×	×	×
	Ethernet	×	×	×	×	×	×
	Serial communication	×	×	×	×	×	×
	CC-Link	④	④	④	×	×	×

○: Accessible (The number in the circle is the property pattern number.), ×: Inaccessible

*1 : The communication cannot be established when a password is set to the connected station side QnUDE(H)CPU and Q12DCCPU-V.

*2 : Redundant CPU cannot be accessed.

*3 : Q12DCCPU-V and Q motion CPU cannot be accessed using CC-Link IE Field Network because CC-Link IE Field Network is not supported.

*4 : LCPU cannot be accessed using CC-Link IE Controller Network because CC-Link IE Controller Network is not supported.

(c) Property list

The following table shows the property settings of communication paths. The setting is not necessary for those properties without description.

Property	Default value	Property pattern			
		①	②[1]	③	④
ActCpuType	34 (CPU_Q02CPU)	CPU type corresponding to target station			
ActDestination IONumber	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No.1: 992 (0x3E0) No.2: 993 (0x3E1) No.3: 994 (0x3E2) No.4: 995 (0x3E3)	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No.1: 992 (0x3E0) No.2: 993 (0x3E1) No.3: 994 (0x3E2) No.4: 995 (0x3E3)
ActDidProperty Bit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActDsidProperty Bit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActHostAddress	1.1.1.1	Host name or IP address of connected station side module			
ActIntelligent PreferenceBit	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station QCPU (Q mode), Q12DCCPU-V, LCPU : 1 (0x01) Other than the above : 0 (0x00)	Target station QCPU (Q mode), Q12DCCPU-V, LCPU : 1 (0x01) Other than the above : 0 (0x00)
ActIONumber[2]	1023 (0x3FF)	For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No.1: 992 (0x3E0) No.2: 993 (0x3E1) No.3: 994 (0x3E2) No.4: 995 (0x3E3)	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No.1: 992 (0x3E0) No.2: 993 (0x3E1) No.3: 994 (0x3E2) No.4: 995 (0x3E3)	Connected station side module I/O address	Connected station side module I/O address
ActMultiDrop ChannelNumber[3]	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Multi-drop channel number	Fixed to 0 (0x00)
ActNetwork Number[4]	0 (0x00)	Fixed to 0 (0x00)	Target station side module network number	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActProtocolType	0x04 (PROTOCOL_ SERIAL)	PROTOCOL_USBGOT (0x13)			
ActStation Number[4]	255 (0xFF)	Fixed to 255 (0xFF)	Target station side Module Station number	Fixed to 255 (0xFF)	Fixed to 255 (0xFF)
ActThrough NetworkType	0 (0x00)	MELSECNET/10 is not included. : 0 (0x00) MELSECNET/10 is included. : 1 (0x01)			

Property	Default value	Property pattern			
		①	②[*1]	③	④
ActTimeOut	10000	Any value specified by user in ms units			
ActUnitNumber	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station side Module Station number	Target station side Module Station number
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_GOT_QNETHER (0x41)			

*1 : Note the following considerations when accessing via Ethernet module (Q series-compatible E71).

- For ActNetworkNumber and ActStationNumber, specify the value set in the parameter setting of the target station side Q series-compatible E71.
- Set "Station No. ⇔ IP information" in the parameter setting of Q series-compatible E71. When setting the parameter, specify the setting other than the automatic response system (any of the IP address calculation system, table conversion system, and combined system) for "Station No. ⇔ IP information system".

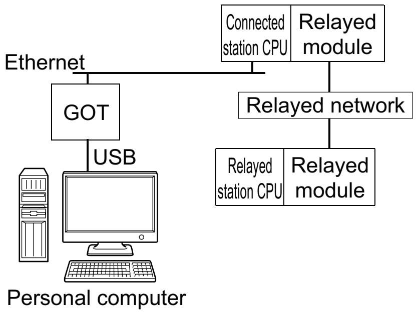
*2 : For the I/O address, specify the value of the actual start I/O number divided by 16.

*3 : Specify the following value for the channel number to be multi-drop linked.
0: Default channel of module
1: Channel 1
2: Channel 2

*4 : For the property pattern of ②, specify the value set in the target station side parameter for ActNetworkNumber and ActStationNumber.

(2) When the connected station is LCPU

(a) Configuration



(b) Property patterns

Connected station CPU[*1]	Relayed network	Relayed station CPU					
LCPU		QCPU (Q mode)[*2]	Q12DC CPU-V	LCPU	QSCPU	Q motion CPU	FXCPU
①	CC IE Field[*3]	④	×	④	×	×	×
	MELSECNET/H	×	×	×	×	×	×
	Ethernet	×	×	×	×	×	×
	Serial communication	②	×	②	×	×	×
	CC-Link	③	③	③	×	×	×

○: Accessible (The number in the circle is the property pattern number.), ×: Inaccessible

*1 : The communication cannot be established when a password is set to the connected station CPU.

*2 : Redundant CPU cannot be accessed.

*3 : LCPU cannot be accessed using CC-Link IE Controller Network because CC-Link IE Controller Network is not supported.

(c) Property list

The following table shows the property settings of communication paths. The setting is not necessary for those properties without description.

Property	Default value	Property pattern			
		①	②	③	④
ActCpuType	34 (CPU_Q02CPU)	CPU type corresponding to target station			
ActDestination IONumber	0 (0x00)	Fixed to 0 (0x00)	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No.1: 992 (0x3E0) No.2: 993 (0x3E1) No.3: 994 (0x3E2) No.4: 995 (0x3E3)	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No.1: 992 (0x3E0) No.2: 993 (0x3E1) No.3: 994 (0x3E2) No.4: 995 (0x3E3)	Fixed to 0 (0x00)
ActDidPropertyBit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Fixed to 1 (0x01)
ActDsidPropertyBit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Fixed to 1 (0x01)

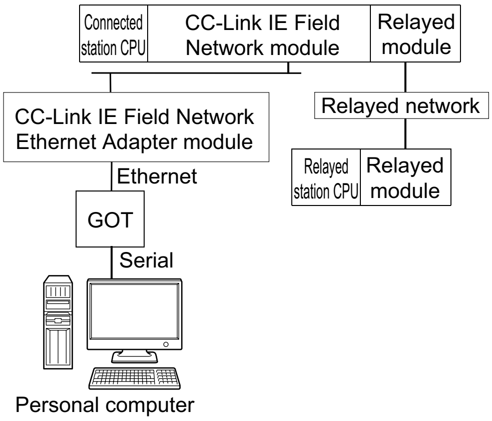
Property	Default value	Property pattern			
		①	②	③	④
ActHostAddress	1.1.1.1	Host name or IP address of connected station side module			
ActIntelligent PreferenceBit	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station QCPU (Q mode), Q12DCCPU-V, LCPU : 1 (0x01) Other than the above : 0 (0x00)	Target station QCPU (Q mode), Q12DCCPU-V, LCPU : 1 (0x01) Other than the above : 0 (0x00)
ActIONumber*1	1023 (0x3FF)	Fixed to 1023 (0x3FF)	Connected station side module I/O address	Connected station side module I/O address	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No.1: 992 (0x3E0) No.2: 993 (0x3E1) No.3: 994 (0x3E2) No.4: 995 (0x3E3)
ActMultiDrop ChannelNumber*2	0 (0x00)	Fixed to 0 (0x00)	Multi-drop channel number	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActNetworkNumber	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station side module network number
ActProtocolType	0x04 (PROTOCOL_ SERIAL)	PROTOCOL_USBGOT (0x13)			
ActStationNumber	255 (0xFF)	Fixed to 255 (0xFF)	Fixed to 255 (0xFF)	Fixed to 255 (0xFF)	Target station side Module Station number
ActThrough NetworkType	0 (0x00)	MELSECNET/10 is not included. : 0 (0x00) MELSECNET/10 is included. : 1 (0x01)			
ActTimeOut	10000	Any value specified by user in ms units			
ActUnitNumber	0 (0x00)	Fixed to 0 (0x00)	Target station side Module Station number	Target station side Module Station number	Fixed to 0 (0x00)
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_GOT_LNETHER (0x55)			

*1 : For the I/O address, specify the value of the actual start I/O number divided by 16.

*2 : Specify the following value for the channel number to be multi-drop linked.
0: Default channel of module
1: Channel 1
2: Channel 2

4.15.11 Personal computer side port: Serial, GOT1000 side port: Serial, CPU side port: CC-Link IE Field Network Ethernet adapter module

(1) Configuration



(2) Property patterns

Connected station CPU	Relayed network	Relayed station CPU					
QnUDE(H)CPU		QCPU (Q mode)*1	Q12DC CPU-V	LCPU	QSCPU	Q motion CPU	FXCPU
①	CC IE Control CC IE Field	②	②*2	②*3	×	×	×
	MELSECNET/H	②	②	×	×	×	×
	Ethernet	②	×	×	×	×	×
	Serial communication	③	×	③	×	×	×
	CC-Link	④	④	④	×	×	×

Connected station CPU	Relayed network	Relayed station CPU					
LCPU		QCPU (Q mode)*1	Q12DC CPU-V	LCPU	QSCPU	Q motion CPU	FXCPU
①	CC IE Field*3	②	×	②	×	×	×
	MELSECNET/H	×	×	×	×	×	×
	Ethernet	×	×	×	×	×	×
	Serial communication	③	×	③	×	×	×
	CC-Link	④	④	④	×	×	×

○: Accessible (The number in the circle is the property pattern number.), ×: Inaccessible

*1 : Redundant CPU cannot be accessed.

*2 : Q12DCCPU-V and Q motion CPU cannot be accessed using CC-Link IE Field Network because CC-Link IE Field Network is not supported.

*3 : LCPU cannot be accessed using CC-Link IE Controller Network because CC-Link IE Controller Network is not supported.

(3) Property list

The following table shows the property settings of communication paths. The setting is not necessary for those properties without description.

Property	Default value	Property pattern			
		①	②	③	④
ActBaudRate	19200 (BAUDRATE_ 19200)	BAUDRATE_9600, BAUDRATE_19200, BAUDRATE_38400, BAUDRATE_57600, BAUDRATE_115200			
ActCpuType	34 (CPU_Q02CPU)	CPU type corresponding to target station			
ActDestination IONumber	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No.1: 992 (0x3E0) No.2: 993 (0x3E1) No.3: 994 (0x3E2) No.4: 995 (0x3E3)	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No.1: 992 (0x3E0) No.2: 993 (0x3E1) No.3: 994 (0x3E2) No.4: 995 (0x3E3)
ActDidProperty Bit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActDsidProperty Bit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActHostAddress	1.1.1.1	Host name or IP address of CC-Link IE Field Network Ethernet adapter module			
ActIntelligent PreferenceBit	0 (0x00)	Fixed to 0 (0x00)			
ActIONumber[*1]	1023 (0x3FF)	For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No.1: 992 (0x3E0) No.2: 993 (0x3E1) No.3: 994 (0x3E2) No.4: 995 (0x3E3)	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No.1: 992 (0x3E0) No.2: 993 (0x3E1) No.3: 994 (0x3E2) No.4: 995 (0x3E3)	Connected station side module I/O address	Connected station side module I/O address
ActMultiDrop ChannelNumber[*2]	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Multi-drop channel number	Fixed to 0 (0x00)
ActNetwork Number[*3]	0 (0x00)	Connected station side CC-Link IE Field Network module network number	Target station side module station number	Connected station side CC-Link IE Field Network module network number	Connected station side CC-Link IE Field Network module network number
ActPortNumber	1 (PORT_1)	Personal computer side COM port number			
ActProtocolType	0x04 (PROTOCOL_ SERIAL)	PROTOCOL_SERIAL (0x04)			
ActStation Number[*3]	255 (0xFF)	Connected station side CC-Link IE Field Network module station number	Target station side Module Station number	Connected station side CC-Link IE Field Network module station number	Connected station side CC-Link IE Field Network module station number

Property	Default value	Property pattern			
		①	②	③	④
ActThrough NetworkType	0 (0x00)	MELSECNET/10 is not included. : 0 (0x00) MELSECNET/10 is included. : 1 (0x01)			
ActTimeOut	10000	Any value specified by user in ms units			
ActUnitNumber	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station side Module Station number	Target station side Module Station number
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_GOT_NZ2GF_ETB (0x5B)			

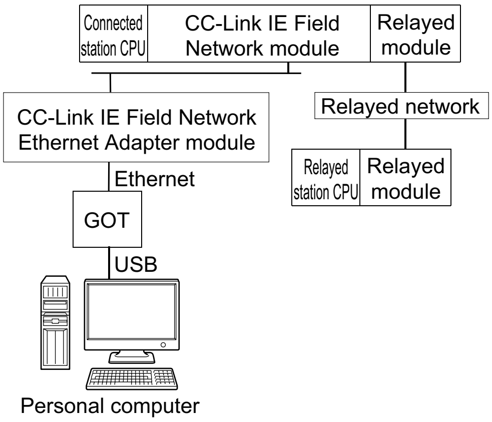
*1 : For the I/O address, specify the value of the actual start I/O number divided by 16.

*2 : Specify the following value for the channel number to be multi-drop linked.
0: Default channel of module
1: Channel 1
2: Channel 2

*3 : For the property pattern of ①, ③ or ④, specify the parameter value set on the CC-Link IE Field Network module on the connected station side for ActNetworkNumber and ActStationNumber.
For the property pattern of ②, specify the value set in the target station side parameter for ActNetworkNumber and ActStationNumber.

4.15.12 Personal computer side port: USB, GOT1000 side port: USB, CPU side port: CC-Link IE Field Network Ethernet adapter module

(1) Configuration



(2) Property patterns

Connected station CPU	Relayed network	Relayed station CPU					
QnUDE(H)CPU		QCPU (Q mode)[*1]	Q12DC CPU-V	LCPU	QSCPU	Q motion CPU	FXCPU
①	CC IE Control CC IE Field	②	②[*2]	②[*3]	×	×	×
	MELSECNET/H	②	②	×	×	×	×
	Ethernet	②	×	×	×	×	×
	Serial communication	③	×	③	×	×	×
	CC-Link	④	④	④	×	×	×

Connected station CPU	Relayed network	Relayed station CPU					
LCPU		QCPU (Q mode)[*1]	Q12DC CPU-V	LCPU	QSCPU	Q motion CPU	FXCPU
①	CC IE Field[*3]	②	×	②	×	×	×
	MELSECNET/H	×	×	×	×	×	×
	Ethernet	×	×	×	×	×	×
	Serial communication	③	×	③	×	×	×
	CC-Link	④	④	④	×	×	×

○: Accessible (The number in the circle is the property pattern number.), ×: Inaccessible

[*1] : Redundant CPU cannot be accessed.

[*2] : Q12DCCPU-V and Q motion CPU cannot be accessed using CC-Link IE Field Network because CC-Link IE Field Network is not supported.

[*3] : LCPU cannot be accessed using CC-Link IE Controller Network because CC-Link IE Controller Network is not supported.

(3) Property list

The following table shows the property settings of communication paths. The setting is not necessary for those properties without description.

Property	Default value	Property pattern			
		①	②	③	④
ActCpuType	34 (CPU_Q02CPU)	CPU type corresponding to target station			
ActDestination IONumber	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No.1: 992 (0x3E0) No.2: 993 (0x3E1) No.3: 994 (0x3E2) No.4: 995 (0x3E3)	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No.1: 992 (0x3E0) No.2: 993 (0x3E1) No.3: 994 (0x3E2) No.4: 995 (0x3E3)
ActDidProperty Bit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActDsidProperty Bit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActHostAddress	1.1.1.1	Host name or IP address of CC-Link IE Field Network Ethernet adapter module			
ActIntelligent PreferenceBit	0 (0x00)	Fixed to 0 (0x00)			
ActIONumber[*1]	1023 (0x3FF)	For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No.1: 992 (0x3E0) No.2: 993 (0x3E1) No.3: 994 (0x3E2) No.4: 995 (0x3E3)	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No.1: 992 (0x3E0) No.2: 993 (0x3E1) No.3: 994 (0x3E2) No.4: 995 (0x3E3)	Connected station side module I/O address	Connected station side module I/O address
ActMultiDrop ChannelNumber[*2]	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Multi-drop channel number	Fixed to 0 (0x00)
ActNetwork Number[*3]	0 (0x00)	Connected station side CC-Link IE Field Network module network number	Target station side module station number	Connected station side CC-Link IE Field Network module network number	Connected station side CC-Link IE Field Network module network number
ActProtocolType	0x04 (PROTOCOL_SERIAL)	PROTOCOL_USBGOT (0x13)			
ActStation Number[*3]	255 (0xFF)	Connected station side CC-Link IE Field Network module station number	Target station side Module Station number	Connected station side CC-Link IE Field Network module station number	Connected station side CC-Link IE Field Network module station number
ActThrough NetworkType	0 (0x00)	MELSECNET/10 is not included. : 0 (0x00) MELSECNET/10 is included. : 1 (0x01)			
ActTimeOut	10000	Any value specified by user in ms units			

Property	Default value	Property pattern			
		①	②	③	④
ActUnitNumber	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station side Module Station number	Target station side Module Station number
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_GOT_NZ2GF_ETB (0x5B)			

*1 : For the I/O address, specify the value of the actual start I/O number divided by 16.

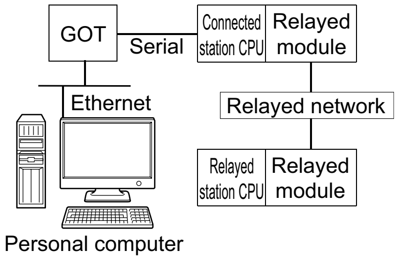
*2 : Specify the following value for the channel number to be multi-drop linked.
0: Default channel of module
1: Channel 1
2: Channel 2

*3 : For the property pattern of ①, ③ or ④, specify the parameter value set on the CC-Link IE Field Network module on the connected station side for ActNetworkNumber and ActStationNumber.
For the property pattern of ②, specify the value set in the target station side parameter for ActNetworkNumber and ActStationNumber.

4.15.13 Personal computer side port: Ethernet port, GOT1000 side port: Ethernet port, CPU side port: Serial

(1) When the connected station is QCPU (Q mode) or Q12DCCPU-V

(a) Configuration



(b) Property patterns

Connected station CPU		Relayed network	Relayed station CPU					
QCPU (Q mode)*1	Q12DC CPU-V		QCPU (Q mode)*1	Q12DC CPU-V	LCPU	QSCPU	Q motion CPU	FXCPU
①	①	CC IE Control CC IE Field	②	②*2	②*3	×	×	×
		MELSECNET/H	②	②	×	×	×	×
		Ethernet	×	×	×	×	×	×
		Serial communication	③	×	③	×	×	×
		CC-Link	④	④	④	×	×	×

○: Accessible (The number in the circle is the property pattern number.), ×: Inaccessible

*1 : Redundant CPU cannot be accessed.

*2 : Q12DCCPU-V and Q motion CPU cannot be accessed using CC-Link IE Field Network because CC-Link IE Field Network is not supported.

*3 : LCPU cannot be accessed using CC-Link IE Controller Network because CC-Link IE Controller Network is not supported.

(c) Property list

The following table shows the property settings of communication paths. The setting is not necessary for those properties without description.

Property	Default value	Property pattern			
		①	②	③	④
ActCpuType	34 (CPU_Q02CPU)	CPU type corresponding to target station			
ActDestination IONumber	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No.1: 992 (0x3E0) No.2: 993 (0x3E1) No.3: 994 (0x3E2) No.4: 995 (0x3E3)	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No.1: 992 (0x3E0) No.2: 993 (0x3E1) No.3: 994 (0x3E2) No.4: 995 (0x3E3)
ActDestination PortNumber	0 (0x00)	GOT port number			
ActDidProperty Bit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActDsidProperty Bit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActHostAddress	1.1.1.1	Host name or IP address of GOT			
ActIntelligent PreferenceBit	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station QCPU (Q mode), Q12DCCPU-V, LCPU : 1 (0x01) Other than the above : 0 (0x00)	Target station QCPU (Q mode), Q12DCCPU-V, LCPU : 1 (0x01) Other than the above : 0 (0x00)
ActIONumber[*1]	1023 (0x3FF)	For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No.1: 992 (0x3E0) No.2: 993 (0x3E1) No.3: 994 (0x3E2) No.4: 995 (0x3E3)	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No.1: 992 (0x3E0) No.2: 993 (0x3E1) No.3: 994 (0x3E2) No.4: 995 (0x3E3)	Connected station side module I/O address	Connected station side module I/O address
ActMultiDrop ChannelNumber[*2]	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Multi-drop channel number	Fixed to 0 (0x00)
ActNetwork Number	0 (0x00)	Fixed to 0 (0x00)	Target station side module station number	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActProtocolType	0x04 (PROTOCOL_ SERIAL)	PROTOCOL_TCPIP (0x05)			
ActStation Number	255 (0xFF)	Fixed to 255 (0xFF)	Target station side module station number	Fixed to 255 (0xFF)	Fixed to 255 (0xFF)

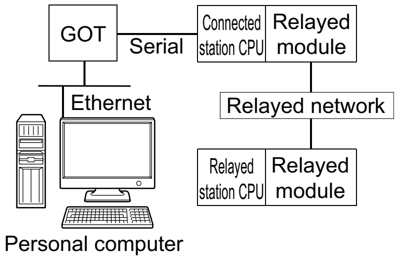
Property	Default value	Property pattern			
		①	②	③	④
ActThrough NetworkType	0 (0x00)	MELSECNET/10 is not included. : 0 (0x00) MELSECNET/10 is included. : 1 (0x01)			
ActTimeOut	10000	Any value specified by user in ms units			
ActUnitNumber	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station side module station number	Target station side module station number
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_GOTETHER_QNCPU (0x56)			

*1 : For the I/O address, specify the value of the actual start I/O number divided by 16.

*2 : Specify the following value for the channel number to be multi-drop linked.
0: Default channel of module
1: Channel 1
2: Channel 2

(2) When the connected station is LCPU

(a) Configuration



(b) Property patterns

Connected station CPU	Relayed network	Relayed station CPU					
LCPU		QCPU (Q mode)*1	Q12DC CPU-V	LCPU	QSCPU	Q motion CPU	FXCPU
①	CC IE Field*2	④	×	④	×	×	×
	MELSECNET/H	×	×	×	×	×	×
	Ethernet	×	×	×	×	×	×
	Serial communication	②	×	②	×	×	×
	CC-Link	③	③	③	×	×	×

○: Accessible (The number in the circle is the property pattern number.), ×: Inaccessible

*1 : Redundant CPU cannot be accessed.

*2 : LCPU cannot be accessed using CC-Link IE Controller Network because CC-Link IE Controller Network is not supported.

(c) Property list

The following table shows the property settings of communication paths. The setting is not necessary for those properties without description.

Property	Default value	Property pattern			
		①	②	③	④
ActCpuType	34 (CPU_Q02CPU)	CPU type corresponding to target station			
ActDestination IONumber	0 (0x00)	Fixed to 0 (0x00)	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No.1: 992 (0x3E0) No.2: 993 (0x3E1) No.3: 994 (0x3E2) No.4: 995 (0x3E3)	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No.1: 992 (0x3E0) No.2: 993 (0x3E1) No.3: 994 (0x3E2) No.4: 995 (0x3E3)	Fixed to 0 (0x00)
ActDestination PortNumber	0 (0x00)	GOT port number			
ActDidPropertyBit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Fixed to 1 (0x01)
ActDsidPropertyBit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Fixed to 1 (0x01)
ActHostAddress	1.1.1.1	Host name or IP address of GOT			

Property	Default value	Property pattern			
		①	②	③	④
ActIntelligent PreferenceBit	0 (0x00)	Fixed to 0 (0x00)	Target station QCPU (Q mode), Q12DCCPU-V, LCPU : 1 (0x01) Other than the above : 0 (0x00)	Target station QCPU (Q mode), Q12DCCPU-V, LCPU : 1 (0x01) Other than the above : 0 (0x00)	Fixed to 0 (0x00)
ActIONumber[*1]	1023 (0x3FF)	Fixed to 1023 (0x3FF)	Connected station side module I/O address	Connected station side module I/O address	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No.1: 992 (0x3E0) No.2: 993 (0x3E1) No.3: 994 (0x3E2) No.4: 995 (0x3E3)
ActMultiDrop ChannelNumber[*2]	0 (0x00)	Fixed to 0 (0x00)	Multi-drop channel number	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActNetworkNumber	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station side module network number
ActProtocolType	0x04 (PROTOCOL_ SERIAL)	PROTOCOL_TCPIP (0x05)			
ActStationNumber	255 (0xFF)	Fixed to 255 (0xFF)	Fixed to 255 (0xFF)	Fixed to 255 (0xFF)	Target station side module station number
ActThroughNetwork Type	0 (0x00)	MELSECNET/10 is not included. : 0 (0x00) MELSECNET/10 is included. : 1 (0x01)			
ActTimeOut	10000	Any value specified by user in ms units			
ActUnitNumber	0 (0x00)	Fixed to 0 (0x00)	Target station side module station number	Target station side module station number	Fixed to 0 (0x00)
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_GOTETHER_LNCPU (0x57)			

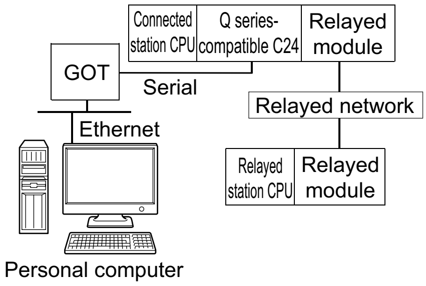
[*1] : For the I/O address, specify the value of the actual start I/O number divided by 16.

[*2] : Specify the following value for the channel number to be multi-drop linked.
0: Default channel of module
1: Channel 1
2: Channel 2

4.15.14 Personal computer side port: Ethernet port, GOT1000 side port: Ethernet port, CPU side port: Q series-compatible C24 or L series-compatible C24

(1) When a relayed module other than the connected station side Q series-compatible C24 exists

(a) Configuration



(b) Property patterns

Connected station CPU		Relayed network	Relayed station CPU					
QCPU (Q mode)*1	Q12DC CPU-V		QCPU (Q mode)*1	Q12DC CPU-V	LCPU	QSCPU	Q motion CPU	FXCPU
①	①	CC IE Control CC IE Field	②	②*2	②*3	×	×	×
		MELSECNET/H	②	②	×	×	×	×
		Ethernet	×	×	×	×	×	×
		Serial communication	③	×	③	×	×	×
		CC-Link	④	④	④	×	×	×

○: Accessible (The number in the circle is the property pattern number.), ×: Inaccessible

*1 : Redundant CPU cannot be accessed.

*2 : Q12DCCPU-V and Q motion CPU cannot be accessed using CC-Link IE Field Network because CC-Link IE Field Network is not supported.

*3 : LCPU cannot be accessed using CC-Link IE Controller Network because CC-Link IE Controller Network is not supported.

(c) Property list

The following table shows the property settings of communication paths. The setting is not necessary for those properties without description.

Property	Default value	Property pattern			
		①	②	③	④
ActCpuType	34 (CPU_Q02CPU)	CPU type corresponding to target station			
ActDestination IONumber	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No.1: 992 (0x3E0) No.2: 993 (0x3E1) No.3: 994 (0x3E2) No.4: 995 (0x3E3)	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No.1: 992 (0x3E0) No.2: 993 (0x3E1) No.3: 994 (0x3E2) No.4: 995 (0x3E3)
ActDestination PortNumber	0 (0x00)	GOT port number			
ActDidProperty Bit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActDsidProperty Bit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActHostAddress	1.1.1.1	Host name or IP address of GOT			
ActIntelligent PreferenceBit	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station QCPU (Q mode), Q12DCCPU-V, LCPU : 1 (0x01) Other than the above : 0 (0x00)	Target station QCPU (Q mode), Q12DCCPU-V, LCPU : 1 (0x01) Other than the above : 0 (0x00)
ActIONumber[*1]	1023 (0x3FF)	For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No.1: 992 (0x3E0) No.2: 993 (0x3E1) No.3: 994 (0x3E2) No.4: 995 (0x3E3)	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No.1: 992 (0x3E0) No.2: 993 (0x3E1) No.3: 994 (0x3E2) No.4: 995 (0x3E3)	Connected station side module I/O address	Connected station side module I/O address
ActMultiDrop ChannelNumber[*2]	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Multi-drop channel number	Fixed to 0 (0x00)
ActNetwork Number	0 (0x00)	Fixed to 0 (0x00)	Target station side module network number	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActProtocolType	0x04 (PROTOCOL_ SERIAL)	PROTOCOL_TCPIP (0x05)			
ActStation Number	255 (0xFF)	Fixed to 255 (0xFF)	Target station side module station number	Fixed to 255 (0xFF)	Fixed to 255 (0xFF)

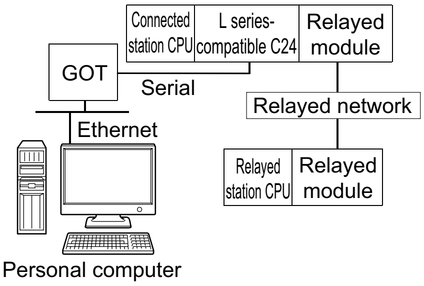
Property	Default value	Property pattern			
		①	②	③	④
ActThrough NetworkType	0 (0x00)	MELSECNET/10 is not included. : 0 (0x00) MELSECNET/10 is included. : 1 (0x01)			
ActTimeOut	10000	Any value specified by user in ms units			
ActUnitNumber	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station side module station number	Target station side module station number
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_GOTETHER_QNCPU (0x56)			

*1 : For the I/O address, specify the value of the actual start I/O number divided by 16.

*2 : Specify the following value for the channel number to be multi-drop linked.
0: Default channel of module
1: Channel 1
2: Channel 2

(2) When a relayed module other than the connected station side L series-compatible C24 exists

(a) Configuration



(b) Property patterns

Connected station CPU	Relayed network	Relayed station CPU					
LCPU		QCPU (Q mode)*1	Q12DC CPU-V	LCPU	QSCPU	Q motion CPU	FXCPU
①	CC IE Field*2	④	×	④	×	×	×
	MELSECNET/H	×	×	×	×	×	×
	Ethernet	×	×	×	×	×	×
	Serial communication	②	×	②	×	×	×
	CC-Link	③	③	③	×	×	×

○: Accessible (The number in the circle is the property pattern number.), ×: Inaccessible

*1 : Redundant CPU cannot be accessed.

*2 : LCPU cannot be accessed using CC-Link IE Controller Network because CC-Link IE Controller Network is not supported.

(c) Property list

The following table shows the property settings of communication paths. The setting is not necessary for those properties without description.

Property	Default value	Property pattern			
		①	②	③	④
ActCpuType	34 (CPU_Q02CPU)	CPU type corresponding to target station			
ActDestinationIO Number	0 (0x00)	Fixed to 0 (0x00)	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No.1: 992 (0x3E0) No.2: 993 (0x3E1) No.3: 994 (0x3E2) No.4: 995 (0x3E3)	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No.1: 992 (0x3E0) No.2: 993 (0x3E1) No.3: 994 (0x3E2) No.4: 995 (0x3E3)	Fixed to 0 (0x00)
ActDestinationPort Number	0 (0x00)	GOT port number			
ActDidPropertyBit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Fixed to 1 (0x01)

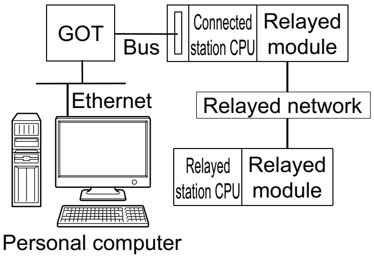
Property	Default value	Property pattern			
		①	②	③	④
ActDsidPropertyBit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Fixed to 1 (0x01)
ActHostAddress	1.1.1.1	Host name or IP address of GOT			
ActIntelligent PreferenceBit	0 (0x00)	Fixed to 0 (0x00)	Target station QCPU (Q mode), Q12DCCPU-V, LCPU : 1 (0x01) Other than the above : 0 (0x00)	Target station QCPU (Q mode), Q12DCCPU-V, LCPU : 1 (0x01) Other than the above : 0 (0x00)	Fixed to 0 (0x00)
ActIONumber*1	1023 (0x3FF)	Fixed to 1023 (0x3FF)	Connected station side module I/O address	Connected station side module I/O address	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No.1: 992 (0x3E0) No.2: 993 (0x3E1) No.3: 994 (0x3E2) No.4: 995 (0x3E3)
ActMultiDrop ChannelNumber*2	0 (0x00)	Fixed to 0 (0x00)	Multi-drop channel number	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActNetworkNumber	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station side module network number
ActProtocolType	0x04 (PROTOCOL_ SERIAL)	PROTOCOL_TCPIP (0x05)			
ActStationNumber	255 (0xFF)	Fixed to 255 (0xFF)	Fixed to 255 (0xFF)	Fixed to 255 (0xFF)	Target station side module station number
ActThrough NetworkType	0 (0x00)	MELSECNET/10 is not included. : 0 (0x00) MELSECNET/10 is included. : 1 (0x01)			
ActTimeOut	10000	Any value specified by user in ms units			
ActUnitNumber	0 (0x00)	Fixed to 0 (0x00)	Target station side module station number	Target station side module station number	Fixed to 0 (0x00)
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_GOTETHER_LNCPU (0x57)			

*1 : For the I/O address, specify the value of the actual start I/O number divided by 16.

*2 : Specify the following value for the channel number to be multi-drop linked.
0: Default channel of module
1: Channel 1
2: Channel 2

4.15.15 Personal computer side port: Ethernet port, GOT1000 side port: Ethernet port, CPU side port: Bus connection

(1) Configuration



(2) Property patterns

Connected station CPU	Relayed network	Relayed station CPU					
QCPU (Q mode)*1		QCPU (Q mode)*1	Q12DC CPU-V	LCPU	QSCPU	Q motion CPU	FXCPU
①	CC IE Control CC IE Field	②	②*2	②*3	×	×	×
	MELSECNET/H	②	②	×	×	×	×
	Ethernet	×	×	×	×	×	×
	Serial communication	③	×	③	×	×	×
	CC-Link	④	④	④	×	×	×

Connected station CPU	Relayed network	Relayed station CPU					
Q12DCCPU-V		QCPU (Q mode)*1	Q12DC CPU-V	LCPU	QSCPU	Q motion CPU	FXCPU
①	CC IE Control CC IE Field	②	②*2	②*3	×	×	×
	MELSECNET/H	②	②	×	×	×	×
	Ethernet	×	×	×	×	×	×
	Serial communication	×	×	×	×	×	×
	CC-Link	④	④	④	×	×	×

○: Accessible (The number in the circle is the property pattern number.), ×: Inaccessible

*1 : Redundant CPU cannot be accessed.
*2 : Q12DCCPU-V and Q motion CPU cannot be accessed using CC-Link IE Field Network because CC-Link IE Field Network is not supported.
*3 : LCPU cannot be accessed using CC-Link IE Controller Network because CC-Link IE Controller Network is not supported.

(3) Property list

The following table shows the property settings of communication paths. The setting is not necessary for those properties without description.

Property	Default value	Property pattern			
		①	②	③	④
ActCpuType	34 (CPU_Q02CPU)	CPU type corresponding to target station			
ActDestination IONumber	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No.1: 992 (0x3E0) No.2: 993 (0x3E1) No.3: 994 (0x3E2) No.4: 995 (0x3E3)	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No.1: 992 (0x3E0) No.2: 993 (0x3E1) No.3: 994 (0x3E2) No.4: 995 (0x3E3)
ActDestination PortNumber	0 (0x00)	GOT port number			
ActDidProperty Bit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActDsidProperty Bit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActHostAddress	1.1.1.1	Host name or IP address of GOT			
ActIntelligent PreferenceBit	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station QCPU (Q mode), Q12DCCPU-V, LCPU : 1 (0x01) Other than the above : 0 (0x00)	Target station QCPU (Q mode), Q12DCCPU-V, LCPU : 1 (0x01) Other than the above : 0 (0x00)
ActIONumber[*1]	1023 (0x3FF)	For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No.1: 992 (0x3E0) No.2: 993 (0x3E1) No.3: 994 (0x3E2) No.4: 995 (0x3E3)	Target station side For single CPU Fixed to 1023 (0x3FF) For multiple CPUs Connected CPU: 1023 (0x3FF) No.1: 992 (0x3E0) No.2: 993 (0x3E1) No.3: 994 (0x3E2) No.4: 995 (0x3E3)	Connected station side module I/O address	Connected station side module I/O address
ActMultiDrop ChannelNumber[*2]	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Multi-drop channel number	Fixed to 0 (0x00)
ActNetwork Number	0 (0x00)	Fixed to 0 (0x00)	Target station side module network number	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActProtocolType	0x04 (PROTOCOL_ SERIAL)	PROTOCOL_TCPIP (0x05)			
ActStation Number	255 (0xFF)	Fixed to 255 (0xFF)	Target station side module station number	Fixed to 255 (0xFF)	Fixed to 255 (0xFF)

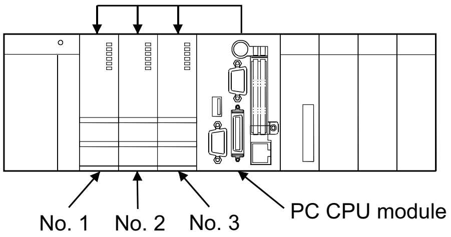
Property	Default value	Property pattern			
		①	②	③	④
ActThrough NetworkType	0 (0x00)	MELSECNET/10 is not included. : 0 (0x00) MELSECNET/10 is included. : 1 (0x01)			
ActTimeOut	10000	Any value specified by user in ms units			
ActUnitNumber	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station side module station number	Target station side module station number
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_GOTETHER_QBUS (0x58)			

*1 : For the I/O address, specify the value of the actual start I/O number divided by 16.

*2 : Specify the following value for the channel number to be multi-drop linked.
0: Default channel of module
1: Channel 1
2: Channel 2

4.16 Q Series Bus Communication

(1) Configuration



(2) Property patterns

Own Base		
QCPU (Q mode)	Q12DCCPU-V	Q motion CPU
①	×	①

○: Accessible (The number in the circle is the property pattern number.), ×: Inaccessible

(3) Property list

The following table shows the property settings of communication paths. The setting is not necessary for those properties without description.

Property	Default value	Property pattern
		①
ActCpuType	34 (CPU_Q02CPU)	CPU type corresponding to the target station
ActIONumber[*1]	1023 (0x3FF)	No.1 992 (0x3E0) No.2 993 (0x3E1) No.3 994 (0x3E2)
ActProtocolType	0x04 (PROTOCOL_SERIAL)	PROTOCOL_QBF (0x10)
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_QBF (0x1F)

CHAPTER 5 FUNCTIONS

This chapter explains the considerations of programming and details of functions.
Read the considerations of programming described in Section 5.1 before creating a program.

(1) Types of functions

Functions for NET control and functions for ACT control are the types of functions. An applicable interface and programming language are different according to the function.

Control	Interface	Programming language	Reference
ACT control	Dispatch interface (Recommended)	VBA Visual C++® .NET (MFC) VBScript Visual Basic® .NET Visual C++® .NET	Page 214, Section 5.2
	Custom interface	Visual C++® .NET	
.NET control	Dispatch interface	Visual Basic® .NET Visual C++® .NET	Page 287, Section 5.3

(2) Interface types

Dispatch interface and custom interface can be used for the functions for ACT control.

Interface	Description	Characteristic
Dispatch interface	Paste controls on the form of Visual Basic® .NET, Visual C++® .NET, or VBA (Excel, Access) to use the controls.	Programming to acquire the interface and to create objects is not necessary. (Programs to create objects without pasting controls on the form can be created.) Programming with dispatch interface is easier than programming with custom interface
Custom interface	Acquire the interface using the interface acquisition function when using controls. Use the object creation function to create objects.	Programs of custom interface are complicated as compared to those of dispatch interface, however the object creation/extinction can be managed in detail.

5.1 Programming Considerations

This section explains the considerations of programming.

(1) Considerations of multi-thread programming

When performing multi-thread programming, follow the rules of COM and ActiveX controls.
For details, refer to the rules and reference books of COM and ActiveX controls.

Point

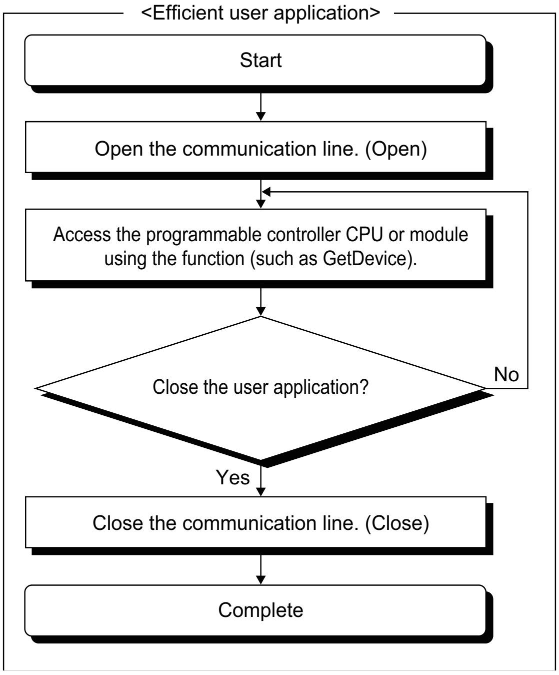
- The ActiveX controls used on MX Component are those of the STA model.
- When passing the interface pointer to another apartment, the interface pointer needs to be marshaled.
Synchronize programs using CoMarshalInterThreadInterfaceInStream or CoGetInterfaceAndReleaseStream of the COM function.

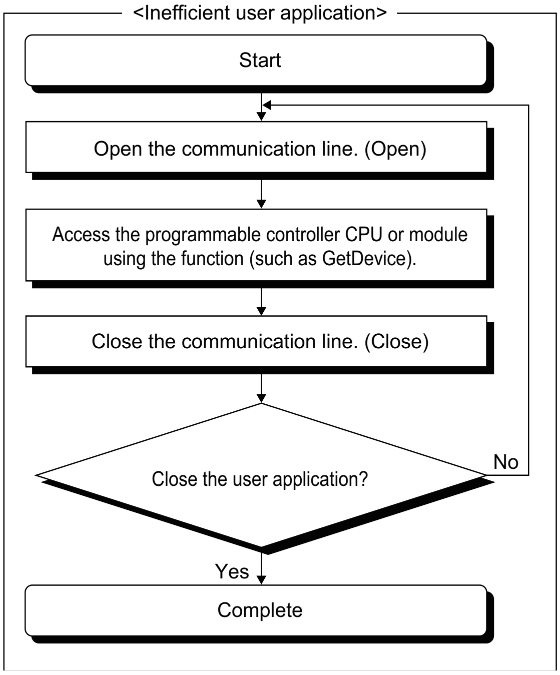
(2) Open function (opening communication line)

The processing may take long time due to the Open function processes: establishment of communication paths, obtaining programmable controller internal information, or the like.

A user needs to create efficient programs in order to improve the speed performance of user applications.

The following figures are flows for creating efficient and inefficient user programs.


<Efficient user application>
Start
Open the communication line. (Open)
Access the programmable controller CPU or module using the function (such as GetDevice).
Close the user application?
No
Yes
Close the communication line. (Close)
Complete


<Inefficient user application>
Start
Open the communication line. (Open)
Access the programmable controller CPU or module using the function (such as GetDevice).
Close the communication line. (Close)
Close the user application?
No
Yes
Complete

(3) Considerations for using QSCPU

The following functions cannot be used for QSCPU.

If any of these functions is used, the error code "0x010A42A0" (mismatched CPU access password) will be returned.

Function Name	Description
WriteDeviceBlock	Write devices in bulk.
WriteDeviceRandom	Write devices randomly.
SetDevice	Set device data.
WriteBuffer	Write data to buffer memory.
SetClockData	Write clock data.
SetCpuStatus	Remote control
WriteDeviceBlock2	Write devices in bulk.
WriteDeviceRandom2	Write devices randomly.
SetDevice2	Set device data.

(4) Differences between LONG type functions and SHORT type/INT type functions

LONG type functions and SHORT type/INT type functions are the functions of MX Component to "read devices in bulk", "write devices in bulk", "read devices randomly", "write devices randomly", "set device data", and "acquire device data".

The following describes the differences between the LONG type functions and SHORT type/INT type functions.

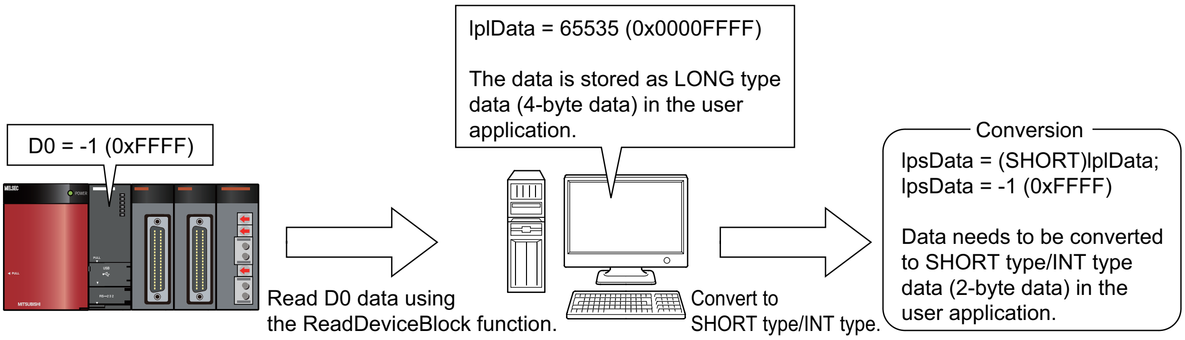
(a) LONG type function (☞ Page 217, Section 5.2.3 to Page 231, Section 5.2.8)

When writing/reading a negative device value using the LONG type function, the LONG type data needs to be converted to the SHORT type/INT type data in the user application.

Write/read negative device values using the SHORT type/INT type functions described in the section (b).

Target function
ReadDeviceBlock, WriteDeviceBlock, ReadDeviceRandom, WriteDeviceRandom, SetDevice, GetDevice

<Example> When reading a device data with a negative value using the ReadDeviceBlock function


lplData = 65535 (0x0000FFFF)
The data is stored as LONG type data (4-byte data) in the user application.
D0 = -1 (0xFFFF)
Conversion
lpsData = (SHORT)lplData;
lpsData = -1 (0xFFFF)
Data needs to be converted to SHORT type/INT type data (2-byte data) in the user application.
Read D0 data using the ReadDeviceBlock function.
Convert to SHORT type/INT type.

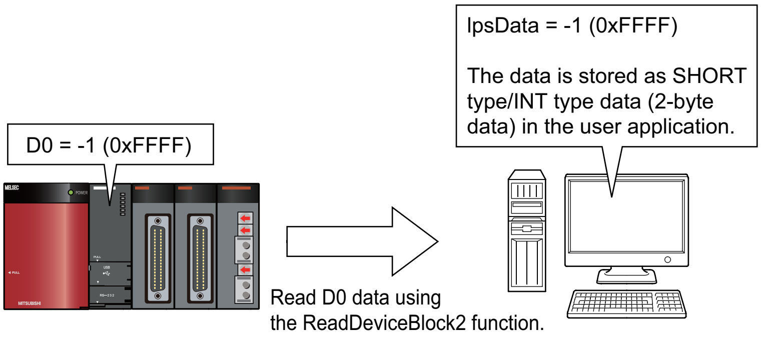
(b) SHORT type/INT type function (☞ Page 264, Section 5.2.18 to Page 279, Section 5.2.23)

When reading/writing a negative device value using the SHORT type/INT type function, the device value can be used as it is without converting it.

When reading/writing double word devices, use any of the LONG type functions described in the section (a).

Target function
ReadDeviceBlock2, WriteDeviceBlock2, ReadDeviceRandom2, WriteDeviceRandom2, SetDevice2, GetDevice2

<Example> When reading a device data with a negative value using the ReadDeviceBlock2 function


lpsData = -1 (0xFFFF)
The data is stored as SHORT type/INT type data (2-byte data) in the user application.
D0 = -1 (0xFFFF)
Read D0 data using the ReadDeviceBlock2 function.

5.2 Details of Functions (For ACT Control)

5.2.1 Open (Opening communication line)

(1) Applicable controls

Applicable to the Act(ML)ProgType control and the Act(ML)UtlType control.

(2) Feature

Open the communication line.

(3) Format (Dispatch interface)

(a) Visual C++® .NET (MFC), VBA

IRet = object.Open()

Long	IRet	Returned value	Output

(b) VBScript

varRet = object.Open()

VARIANT	varRet	Returned value (LONG type)	Output

(c) Visual Basic® .NET

IRet = object.Open()

Integer	IRet	Returned value	Output

(d) Visual C++® .NET

iRet = object.Open()

int	iRet	Returned value	Output

(4) Format (Custom interface)

(a) Visual C++® .NET (MFC)

hResult = object.Open(*lplRetCode)

HRESULT	hResult	Returned value of COM	Output
LONG	*lplRetCode	Returned value of communication function	Output

(5) Description

Lines are connected according to the set value of the Open function property.

(6) Returned value

Normal termination : 0 is returned.

Abnormal termination: Any value other than 0 is returned. (☞ Page 378, CHAPTER 7 ERROR CODES)

Point

- When modem communication is used, the Open function cannot be executed without the execution of the Connect function.
- If the Open function property is changed after the completion of the Open function, the setting of communication target is not changed. To change the communication setting, close the communication line, set the setting of communication target, and open the communication line again.
- Even when a CPU type which is different from the CPU used for the communication is set to the ActCpuType property, the Open function may be completed normally.
  In such a case, the connection range, usable method, or device range may be narrowed.
  When executing the Open function, set the correct CPU type to the ActCpuType property.

5.2.2 Close (Closing communication line)

(1) Applicable controls

Applicable to the Act(ML)ProgType control and the Act(ML)UtlType control.

(2) Feature

Close the communication line.

(3) Format (Dispatch interface)

(a) Visual C++® .NET (MFC), VBA

lRet = object.Close()

Long	lRet	Returned value	Output

(b) VBScript

varRet = object.Close()

VARIANT	varRet	Returned value (LONG type)	Output

(c) Visual Basic® .NET

lRet = object.Close()

Integer	lRet	Returned value	Output

(d) Visual C++® .NET

iRet = object.Close()

int	iRet	Returned value	Output

(4) Format (Custom interface)

(a) Visual C++® .NET (MFC)

hResult = object.Close(*lplRetCode)

HRESULT	hResult	Returned value of COM	Output
LONG	*lplRetCode	Returned value of communication function	Output

(5) Description

The line connected using the Open function is closed.

(6) Returned value

Normal termination : 0 is returned.
Abnormal termination: Any value other than 0 is returned. (☞ Page 378, CHAPTER 7 ERROR CODES)

5.2.3 ReadDeviceBlock (Reading devices in bulk)

(1) Applicable controls

Applicable to the Act(ML)ProgType control and the Act(ML)UtlType control.

(2) Feature

Read devices in bulk.

(3) Format (Dispatch interface)

(a) VBA

lRet = object.ReadDeviceBlock(szDevice, lSize, lData(0))

Long	lRet	Returned value	Output
String	szDevice	Device name	Input
Long	lSize	Number of read points	Input
Long	lData(n)	Read device value	Output

(b) Visual C++® .NET (MFC)

lRet = object.ReadDeviceBlock(szDevice, lSize, *lplData)

Long	lRet	Returned value	Output
CString	szDevice	Device name	Input
Long	lSize	Number of read points	Input
Long	*lplData	Read device value	Output

(c) VBScript

varRet = object.ReadDeviceBlock(varDevice, varSize, lpvarData)

VARIANT	varRet	Returned value (LONG type)	Output
VARIANT	varDevice	Device name (character string type)	Input
VARIANT	varSize	Number of read points (LONG type)	Input
VARIANT	lpvarData	Read device value (LONG array type)	Output

(d) Visual Basic® .NET

lRet = object.ReadDeviceBlock(szDevice, iSize, iData(0))

Integer	lRet	Returned value	Output
String	szDevice	Device name	Input
Integer	iSize	Number of read points	Input
Integer	iData(n)	Read device value	Output

(e) Visual C++® .NET

iRet = object.ReadDeviceBlock(*szDevice, iSize, *iplData)

int	iRet	Returned value	Output
String	*szDevice	Device name	Input
int	iSize	Number of read points	Input
int	*iplData	Read device value	Output

(4) Format (Custom interface)

(a) Visual C++® .NET (MFC)

hResult = object.ReadDeviceBlock(szDevice, lSize, *lplData, *lplRetCode)

HRESULT	hResult	Returned value of COM	Output
BSTR	szDevice	Device name	Input
LONG	lSize	Number of read points	Input
LONG	*lplData	Read device value	Output
LONG	*lplRetCode	Returned value of communication function	Output

(5) Description

- The device values for the amount specified for lSize (varSize) are read in bulk starting from the device specified for szDevice (varDevice).
- The read device values are stored in lData (lplData or lpvarData).
- For lData (lplData or lpvarData), reserve arrays for more than the amount specified for lSize (varSize).

(6) How to specify devices

The following describes how to specify devices.

<When bit device is specified>

Example: Read 3 points (3 words) of data in 16-point unit starting from M0.

Upper 2 bytes	Lower 2 bytes
Not used (0 is stored.)	M0 to M15[*1]
	M16 to M31[*1]
	M32 to M47[*1]

<When word device is specified>

Example: Read 3 points of data starting from D0.

Upper 2 bytes	Lower 2 bytes
Not used (0 is stored.)	D0
	D1
	D2

<When FXCPU devices of CN200 and later are specified>

Example: Read 6 points of data starting from CN200.[*2]

Upper 2 bytes	Lower 2 bytes
Not used (0 is stored.)	L of CN200
	H of CN200
	L of CN201
	H of CN201
	L of CN202
	H of CN202

<When FD device is specified (4-word device)>

Example: Read 6 points of data starting from FD0.

Upper 2 bytes	Lower 2 bytes
Not used (0 is stored.)	LL of FD0
	LH of FD0
	HL of FD0
	HH of FD0
	LL of FD1
	LH of FD1

<8-bit devices assigned to gateway devices>

Example: Read 4 points of data starting from EG0.
(When 8-bit devices (from E0000 to E0007 of SHARP programmable controller) are assigned from EG0 to EG3.)

Upper 2 bytes	Lower 2 bytes	
Not used (0 is stored.)	EG0	
	(E0001)	(E0000)
	EG1	
	(E0003)	(E0002)
	EG2	
	(E0005)	(E0004)
	EG3	
	(E0007)	(E0006)

*1 : Devices are stored from the lower bit in the order of device number.
*2 : For CN200 or later of FXCPU, 2 words are read for each 2 points. Reading only 1 point of data will result in an error.

(7) Returned value

Normal termination : 0 is returned.
Abnormal termination: Any value other than 0 is returned. (☞ Page 378, CHAPTER 7 ERROR CODES)

Point

- The maximum number of read points that can be specified for lSize (varSize) should be the value which satisfies the following condition.
  Read start device number + Number of read points ≦ Last device number
- When specifying bit devices, a multiple of 16 can be specified as a device number.
- For lData (lplData or lpvarData), reserve a memory area for the number of points specified for lSize (varSize).
  If the memory area is not reserved, a critical error (an application error or the like) may occur.

5.2.4 WriteDeviceBlock (Writing devices in bulk)

(1) Applicable controls

Applicable to the Act(ML)ProgType control and the Act(ML)UtlType control.

(2) Feature

Write devices in bulk.

(3) Format (Dispatch interface)

(a) VBA

lRet = object.WriteDeviceBlock(szDevice, lSize, lData(0))

Long	lRet	Returned value	Output
String	szDevice	Device name	Input
Long	lSize	Number of write points	Input
Long	lData(n)	Device value to be written	Input

(b) Visual C++® .NET (MFC)

lRet = object.WriteDeviceBlock(szDevice, lSize, *lplData)

Long	lRet	Returned value	Output
CString	szDevice	Device name	Input
Long	lSize	Number of write points	Input
Long	*lplData	Device value to be written	Input

(c) VBScript

varRet = object.WriteDeviceBlock(varDevice, varSize, varData)

VARIANT	varRet	Returned value (LONG type)	Output
VARIANT	varDevice	Device name (character string type)	Input
VARIANT	varSize	Number of write points (LONG type)	Input
VARIANT	varData	Device value to be written (LONG array type)	Input

(d) Visual Basic® .NET

lRet = object.WriteDeviceBlock(szDevice, iSize, iData(0))

Integer	lRet	Returned value	Output
String	szDevice	Device name	Input
Integer	iSize	Number of write points	Input
Integer	iData(n)	Device value to be written	Input

(e) Visual C++® .NET

iRet = object.WriteDeviceBlock(*szDevice, iSize, *iplData)

int	iRet	Returned value	Output
String	*szDevice	Device name	Input
int	iSize	Number of write points	Input
int	*iplData	Device value to be written	Input

(4) Format (Custom interface)

(a) Visual C++® .NET (MFC)

hResult = object.WriteDeviceBlock(szDevice, lSize, *lplData, *lplRetCode)

HRESULT	hResult	Returned value of COM	Output
BSTR	szDevice	Device name	Input
LONG	lSize	Number of write points	Input
LONG	*lplData	Device value to be written	Input
LONG	*lplRetCode	Returned value of communication function	Output

(5) Description

- The device values of lSize (varSize) are written in bulk starting from the device specified for szDevice (varDevice).
- The device values to be written are stored in lData (lplData or varData).
- For lData (lplData or varData), reserve arrays for more than the amount specified for lSize (varSize).

(6) How to specify devices

The following describes how to specify devices.

<When bit device is specified>

Example: Write 3 points (3 words) of data in 16-point unit starting from M0.

Upper 2 bytes	Lower 2 bytes
Not used (0 is stored.)	M0 to M15[*1]
	M16 to M31[*1]
	M32 to M47[*1]

<When word device is specified>

Example: Write 3 points of data starting from D0.

Upper 2 bytes	Lower 2 bytes
Not used (0 is stored.)	D0
	D1
	D2

<When FXCPU devices of CN200 and later are specified>

Example: Write 6 points of data starting from CN200.[*2]

Upper 2 bytes	Lower 2 bytes
Not used (0 is stored.)	L of CN200
	H of CN200
	L of CN201
	H of CN201
	L of CN202
	H of CN202

<When FD device is specified (4-word device)>

Example: Write 6 points of data starting from FD0.

Upper 2 bytes	Lower 2 bytes
Not used (0 is stored.)	LL of FD0
	LH of FD0
	HL of FD0
	HH of FD0
	LL of FD1
	LH of FD1

[*1] : Devices are stored from the lower bit in the order of device number.
[*2] : For CN200 or later of FXCPU, 2 words are written for each 2 points. Writing only 1 point of data will result in an error.

<8-bit devices assigned to gateway devices>

Example: Write 4 points of data starting from EG0.
(When 8-bit devices (from E0000 to E0007 of SHARP programmable controller) are assigned from EG0 to EG3.)

Upper 2 bytes	Lower 2 bytes	
Not used (0 is stored.)	EG0	
	(E0001)	(E0000)
	EG1	
	(E0003)	(E0002)
	EG2	
	(E0005)	(E0004)
	EG3	
	(E0007)	(E0006)

(7) Returned value

Normal termination : 0 is returned.

Abnormal termination: Any value other than 0 is returned. (☞ Page 378, CHAPTER 7 ERROR CODES)

Point

- ● The maximum number of write points that can be specified for lSize(varSize) should be the value which satisfies the following condition.
  Write starting device number + Number of write points ≦ Last device number
- ● When specifying bit devices, a multiple of 16 can be specified as a device number.
- ● For lData (lplData or varData), reserve a memory area for the number of points specified for lSize (varSize).
  If the memory area is not reserved, a critical error (an application error or the like) may occur.

5.2.5 ReadDeviceRandom (Reading devices randomly)

(1) Applicable controls

Applicable to the Act(ML)ProgType control and the Act(ML)UtlType control.

(2) Feature

Read devices randomly.

(3) Format (Dispatch interface)

(a) VBA

lRet = object.ReadDeviceRandom(szDeviceList, lSize, lData(0))

Long	lRet	Returned value	Output
String	szDeviceList	Device name	Input
Long	lSize	Number of read points	Input
Long	lData(n)	Read device value	Output

(b) Visual C++® .NET (MFC)

lRet = object.ReadDeviceRandom(szDeviceList, lSize, *lplData)

Long	lRet	Returned value	Output
CString	szDeviceList	Device name	Input
Long	lSize	Number of read points	Input
Long	*lplData	Read device value	Output

(c) VBScript

varRet = object.ReadDeviceRandom(varDeviceList, varSize, lpvarData)

VARIANT	varRet	Returned value (LONG type)	Output
VARIANT	varDeviceList	Device name (character string type)	Input
VARIANT	varSize	Number of read points (LONG type)	Input
VARIANT	lpvarData	Read device value (LONG array type)	Output

(d) Visual Basic® .NET

lRet = object.ReadDeviceRandom(szDeviceList, iSize, iData(0))

Integer	lRet	Returned value	Output
String	szDeviceList	Device name	Input
Integer	iSize	Number of read points	Input
Integer	iData(n)	Read device value	Output

(e) Visual C++® .NET

iRet = object.ReadDeviceRandom(*szDeviceList, iSize, *iplData)

int	iRet	Returned value	Output
String	*szDeviceList	Device name	Input
int	iSize	Number of read points	Input
int	*iplData	Read device value	Output

(4) Format (Custom interface)

(a) Visual C++® .NET (MFC)

hResult = object.ReadDeviceRandom(szDevice, lSize, *lplData, *lplRetCode)

HRESULT	hResult	Returned value of COM	Output
BSTR	szDevice	Device name	Input
LONG	lSize	Number of read points	Input
LONG	*lplData	Read device value	Output
LONG	*lplRetCode	Returned value of communication function	Output

(5) Description

- Data of a device group for the size of lSize (varSize) specified for szDeviceList (varDeviceList) are read.
- The read device values are stored in lData (lplData or lpvarData).
- Use the line feed symbol to separate the character string specified for the device list. The line feed symbol is not necessary to be suffixed to the last device.
  Example

When using Visual Basic®.NET ,VBA,VBScript:	"D0" & vbLf & "D1" & vbLf & "D2"
When using Visual C++®.NET:	D0\nD1\nD2

- For lData (lplData or lpvarData), reserve arrays for more than the amount specified for lSize (varSize).

(6) How to specify devices

The following describes how to specify devices.

Example 1: When devices are specified as follows (Number of points: 3 points)

When using Visual Basic®.NET ,VBA,VBScript:	M0 & vbLf & D0 & vbLf & K8M0
When using Visual C++®.NET:	M0\nD0\nK8M0

Upper 2 bytes	Lower 2 bytes
Not used (0 is stored.)	M0
	D0
M16 to M31*1	M0 to M15*1

Example 2: When devices including FXCPU devices of CN200 and later are specified (Total number of points: 3 points)*2

When using Visual Basic®.NET ,VBA,VBScript:	D0 & vbLf & CN200 & vbLf & D1
When usingVisual C++®.NET:	D0\nCN200\nD1

Upper 2 bytes	Lower 2 bytes
Not used (0 is stored.)	D0
H of CN200	L of CN200
Not used (0 is stored.)	D1

*1 : Devices are stored from the lower bit in the order of device number.
*2 : For CN200 or later of FXCPU, 2 words are read for each point when reading devices randomly.

Example 3: When devices including FD are specified (Total number of points: 3 points)

When using Visual Basic®.NET ,VBA,VBScript:	D0 & vbLf & FD0 & vbLf & D1
When using Visual C++®.NET:	D0\nFD0\nD1

Upper 2 bytes	Lower 2 bytes
Not used (0 is stored.)	D0
	LL of FD0
	D1

Example 4: When 8-bit devices including EG are specified (Total number of points: 3 points)

The following is the example when 8-bit devices (E0000 and E0001 of SHARP programmable controller) are assigned to EG0.

When using Visual Basic®.NET ,VBA,VBScript:	D0 & vbLf & EG0 & vbLf & D1
When using Visual C++®.NET:	D0\nEG0\nD1

Upper 2 bytes	Lower 2 bytes	
Not used (0 is stored.)	D0	
	EG0	
	(E0001)	(E0000)
	D1	

(7) Returned value

Normal termination : 0 is returned.

Abnormal termination: Any value other than 0 is returned. (☞ Page 378, CHAPTER 7 ERROR CODES)

Point

- The maximum number of read points that can be specified for lSize (varSize) is 0x7FFFFFFF points.
- For lData (lplData or lpvarData), reserve a memory area for the number of points specified for lSize (varSize). If the memory area is not reserved, a critical error (an application error or the like) may occur.

5.2.6 WriteDeviceRandom (Writing devices randomly)

(1) Applicable controls

Applicable to the Act(ML)ProgType control and the Act(ML)UtlType control.

(2) Feature

Write devices randomly.

(3) Format (Dispatch interface)

(a) VBA

lRet = object.WriteDeviceRandom(szDeviceList, lSize, lData(0))

Long	lRet	Returned value	Output
String	szDeviceList	Device name	Input
Long	lSize	Number of write points	Input
Long	lData(n)	Device value to be written	Input

(b) Visual C++® .NET (MFC)

lRet = object.WriteDeviceRandom(szDeviceList, lSize, *lplData)

Long	lRet	Returned value	Output
CString	szDeviceList	Device name	Input
Long	lSize	Number of write points	Input
Long	*lplData	Device value to be written	Input

(c) VBScript

varRet = object.WriteDeviceRandom(varDeviceList, varSize, varData)

VARIANT	varRet	Returned value (LONG type)	Output
VARIANT	varDeviceList	Device name (character string type)	Input
VARIANT	varSize	Number of write points (LONG type)	Input
VARIANT	varData	Device value to be written (LONG array type)	Input

(d) Visual Basic® .NET

IRet = object.WriteDeviceRandom(szDeviceList, iSize, iData(0))

Integer	IRet	Returned value	Output
String	szDeviceList	Device name	Input
Integer	iSize	Number of write points	Input
Integer	iData(n)	Device value to be written	Input

(e) Visual C++® .NET

iRet = object.WriteDeviceRandom(*szDeviceList, iSize, *iplData)

int	iRet	Returned value	Output
String	*szDeviceList	Device name	Input
int	iSize	Number of write points	Input
int	*iplData	Device value to be written	Input

(4) Format (Custom interface)

(a) Visual C++® .NET (MFC)

hResult = object.WriteDeviceRandom(szDeviceList, lSize, *lplData,*lplRetCode)

HRESULT	hResult	Returned value of COM	Output
BSTR	szDeviceList	Device name	Input
LONG	lSize	Number of write points	Input
LONG	*lplData	Device value to be written	Input
LONG	*lplRetCode	Returned value of communication function	Output

(5) Description

- Data of a device group for the size of lSize (varSize) specified for szDeviceList (varDeviceList) are written.
- The device values to be written are stored in lData (lplData or varData).
- Use the line feed symbol to separate the character string specified for the device list. The line feed symbol is not necessary to be suffixed to the last device.
  Example

When using Visual Basic®.NET ,VBA,VBScript:	"D0" & vbLf & "D1" & vbLf & "D2"
When using Visual C++®.NET:	D0\nD1\nD2

- For lData (lplData or varData), reserve arrays for more than the amount specified for lSize (varSize).

(6) How to specify devices

The following describes how to specify devices.

Example 1: When devices are specified as follows (Number of points: 3 points)

When using Visual Basic®.NET ,VBA,VBScript:	M0 & vbLf & D0 & vbLf & K8M0
When using Visual C++®.NET:	M0\nD0\nK8M0

Upper 2 bytes	Lower 2 bytes
Not used (0 is stored.)	M0
	D0
M16 to M31[*1]	M0 to M15[*1]

Example 2: When devices including FXCPU devices of CN200 and later are specified (Total number of points: 3 points)[*2]

When using Visual Basic®.NET ,VBA,VBScript:	D0 & vbLf & CN200 & vbLf & D1
When using Visual C++®.NET:	D0\nCN200\nD1

Upper 2 bytes	Lower 2 bytes
Not used (0 is stored.)	D0
H of CN200	L of CN200
Not used (0 is stored.)	D1

[*1] : Devices are stored from the lower bit in the order of device number.

[*2] : For CN200 or later of FXCPU, 2 words are written for each point when writing devices randomly.

Example 3: When devices including FD are specified (Total number of points: 3 points)

When using Visual Basic®.NET ,VBA,VBScript:	D0 & vbLf & FD0 & vbLf & D1
When using Visual C++®.NET:	D0\nFD0\nD1

Upper 2 bytes	Lower 2 bytes
Not used (0 is stored.)	D0
	LL of FD0
	D1

Example 4: When 8-bit devices including EG are specified (Total number of points: 3 points)

The following is the example when 8-bit devices (E0000 and E0001 of SHARP programmable controller) are assigned to EG0.

When using Visual Basic®.NET ,VBA,VBScript:	D0 & vbLf & EG0 & vbLf & D1
When using Visual C++®.NET:	D0\nEG0\nD1

Upper 2 bytes	Lower 2 bytes	
Not used (0 is stored.)	D0	
	EG0	
	(E0001)	(E0000)
	D1	

(7) Returned value

Normal termination : 0 is returned.

Abnormal termination: Any value other than 0 is returned. (☞ Page 378, CHAPTER 7 ERROR CODES)

Point

- The maximum number of write points that can be specified for lSize (varSize) is 0x7FFFFFFF points.
- For lData (lplData or varData), reserve a memory area for the number of points specified for lSize (varSize). If the memory area is not reserved, a critical error (an application error or the like) may occur.
- If a Q motion CPU is accessed, an error is returned.

5.2.7 SetDevice (Setting device data)

(1) Applicable controls

Applicable to the Act(ML)ProgType control and the Act(ML)UtlType control.

(2) Feature

Set one point of device.

(3) Format (Dispatch interface)

(a) VBA

lRet = object.SetDevice(szDevice, lData)

Long	lRet	Returned value	Output
String	szDevice	Device name	Input
Long	lData	Set data	Input

(b) Visual C++® .NET (MFC)

lRet = object.SetDevice(szDevice, lData)

Long	lRet	Returned value	Output
CString	szDevice	Device name	Input
Long	lData	Set data	Input

(c) VBScript

varRet = object.SetDevice(varDevice, lpvarData)

VARIANT	varRet	Returned value (LONG type)	Output
VARIANT	varDevice	Device name (character string type)	Input
VARIANT	varData	Set data (LONG type)	Input

(d) Visual Basic® .NET

IRet = object.SetDevice(szDevice, iData)

Integer	IRet	Returned value	Output
String	szDevice	Device name	Input
Integer	iData	Set data	Input

(e) Visual C++® .NET

iRet = object.SetDevice(*szDevice, iData)

int	iRet	Returned value	Output
String	*szDevice	Device name	Input
int	iData	Set data	Input

(4) Format (Custom interface)

(a) Visual C++® .NET (MFC)

hResult = object.SetDevice(szDevice, lData, *lplRetCode)

HRESULT	hResult	Returned value of COM	Output
BSTR	szDevice	Device name	Input
LONG	lData	Set data	Input
LONG	*lplRetCode	Returned value of communication function	Output

(5) Description

- One point of device specified for szDevice (varDevice) is specified using lData (varData).
- When specifying bit devices, the least significant bit of the lData value (varData value) is valid.

(6) How to specify devices

The following describes how to specify devices.

<When bit device is specified>

Example: M0

Upper 2 bytes	Lower 2 bytes
Not used (0 is stored.)	M0

<When word device is specified>

Example: D0

Upper 2 bytes	Lower 2 bytes
Not used (0 is stored.)	D0

<When double word device is specified>

Example: K8M0

Upper 2 bytes	Lower 2 bytes
M16 to M31[*1]	M0 to M15[*1]

<When CN200 or later of FXCPU is specified>

Example: CN200

Upper 2 bytes	Lower 2 bytes
H of CN200	L of CN200

<When gateway device is specified>

Example: When 8-bit devices (E0000 and E0001 of SHARP programmable controller) are assigned to EG0

<table>
<tr><th>Upper 2 bytes</th><th colspan="2">Lower 2 bytes</th></tr>
<tr><td rowspan="2">Not used
(0 is stored.)</td><td colspan="2">EG0</td></tr>
<tr><td>(E0001)</td><td>(E0000)</td></tr>
</table>

[*1] : Devices are stored from the lower bit in the order of device number.

(7) Returned value

Normal termination : 0 is returned.

Abnormal termination: Any value other than 0 is returned. (☞ Page 378, CHAPTER 7 ERROR CODES)

5.2.8 GetDevice (Acquiring device data)

(1) Applicable controls

Applicable to the Act(ML)ProgType control and the Act(ML)UtlType control.

(2) Feature

Acquire one point of device.

(3) Format (Dispatch interface)

(a) VBA

lRet = object.GetDevice(szDevice, lData)

Long	lRet	Returned value	Output
String	szDevice	Device name	Input
Long	lData	Acquired data	Output

(b) Visual C++® .NET (MFC)

lRet = object.GetDevice(szDevice, *lplData)

Long	lRet	Returned value	Output
CString	szDevice	Device name	Input
Long	*lplData	Acquired data	Output

(c) VBScript

varRet = object.GetDevice(varDevice, lpvarData)

VARIANT	varRet	Returned value (LONG type)	Output
VARIANT	varDevice	Device name (character string type)	Input
VARIANT	lpvarData	Acquired data (LONG type)	Output

(d) Visual Basic® .NET

IRet = object.GetDevice(szDevice, iData)

Integer	IRet	Returned value	Output
String	szDevice	Device name	Input
Integer	iData	Acquired data	Output

(e) Visual C++® .NET

iRet = object.GetDevice(*szDevice, *iplData)

int	iRet	Returned value	Output
String	*szDevice	Device name	Input
int	*iplData	Acquired data	Output

(4) Format (Custom interface)

(a) Visual C++® .NET (MFC)

hResult = object.GetDevice(szDevice, *lplData, *lplRetCode)

HRESULT	hResult	Returned value of COM	Output
BSTR	szDevice	Device name	Input
LONG	*lplData	Acquired data	Output
LONG	*lplRetCode	Returned value of communication function	Output

(5) Description

One point of device data specified for szDevice (varDevice) is stored in lData (lplData or lpvarData).

(6) How to specify devices

The following describes how to specify devices.

<When bit device is specified>

Example: M0

Upper 2 bytes	Lower 2 bytes
Not used (0 is stored.)	M0

<When word device is specified>

Example: D0

Upper 2 bytes	Lower 2 bytes
Not used (0 is stored.)	D0

<When double-word device is specified>

Example: K8M0

Upper 2 bytes	Lower 2 bytes
M16 to M31[*1]	M0 to M15[*1]

<When CN200 or later of FXCPU is specified>

Example: CN200

Upper 2 bytes	Lower 2 bytes
H of CN200	L of CN200

<When gateway device is specified>

Example: When 8-bit devices (E0000 and E0001 of SHARP programmable controller) are assigned to EG0

<table>
<tr><th>Upper 2 bytes</th><th colspan="2">Lower 2 bytes</th></tr>
<tr><td rowspan="2">Not used
(0 is stored.)</td><td colspan="2">EG0</td></tr>
<tr><td>(E0001)</td><td>(E0000)</td></tr>
</table>

*1 : Devices are stored from the lower bit in the order of device number.

(7) Returned value

Normal termination : 0 is returned.

Abnormal termination: Any value other than 0 is returned. (☞ Page 378, CHAPTER 7 ERROR CODES)

5.2.9 ReadBuffer (Reading data from buffer memory)

(1) Applicable controls

Applicable to the Act(ML)ProgType control and the Act(ML)UtlType control.

(2) Applicable communication paths

The following table shows the applicable communication paths.

Communication path		Property setting		Applicability
		ActUnitType	ActProtocolType	
Serial communication	Connected module: Q series-compatible C24	UNIT_QJ71C24	PROTOCOL_SERIAL	○
	Connected module: L series-compatible C24	UNIT_LJ71C24		○
	Connected module: FX extended port	UNIT_FX485BD		×
Ethernet communication		UNIT_QJ71E71	PROTOCOL_TCPIP, PROTOCOL_UDPIP	○
		UNIT_LNETHER	PROTOCOL_TCPIP	
		UNIT_LNETHER, UNIT_LNETHER_DIRECT	PROTOCOL_UDPIP	
		UNIT_QNETHER	PROTOCOL_TCPIP	
		UNIT_QNETHER, UNIT_QNETHER_DIRECT	PROTOCOL_UDPIP	
		UNIT_NZ2GF_ETB	PROTOCOL_TCPIP	
		UNIT_NZ2GF_ETB UNIT_NZ2GF_ETB_DIRECT	PROTOCOL_UDPIP	
CPU COM communication	Connection target CPU: QCPU (Q mode)	UNIT_QNCPU	PROTOCOL_SERIAL	○
	Connection target CPU: LCPU	UNIT_LNCPU		○
	Connection target CPU: Q motion CPU	UNIT_QNMOTION		×
	Connection target CPU: FXCPU	UNIT_FXCPU		○*1
USB communication	Connection target CPU: QCPU (Q mode)	UNIT_QNUSB	PROTOCOL_USB	○
	Connection target CPU: LCPU	UNIT_LNUSB		○
	Connection target CPU: QSCPU	UNIT_QSUSB		○
	Connection target CPU: Q motion CPU	UNIT_QNMOTIONUSB		×
	Connection target CPU: FXCPU	UNIT_FXCPU		○
MELSECNET/H communication		UNIT_MNETHBOARD	PROTOCOL_MNETH	○*2
CC-Link IE Controller Network communication		UNIT_MNETGBOARD	PROTOCOL_MNETG	
CC-Link IE Field Network communication		UNIT_CCIEFBOARD	PROTOCOL_CCIEF	
CC-Link communication		UNIT_CCLINKBOARD	PROTOCOL_CCLINK	
CC-Link G4 communication		UNIT_G4QCPU	PROTOCOL_SERIAL	○
GX Simulator communication		UNIT_SIMULATOR	PROTOCOL_SHAREDMEMORY	○*3

○: Applicable, ×: Not applicable

*1 : An error is returned when the CPU is other than FX2N, FX2NC, FX3U, and FX3UC.

*2 : An error is returned when the own board is accessed.

*3 : An error is returned when the CPU is other than FX0N, FXU, FX2C, FX2N, and FX2NC.

Communication path		Property setting		Applicability
		ActUnitType	ActProtocolType	
GX Simulator2 communication		UNIT_SIMULATOR2	—	○
Modem communication	Connected module: Q series-compatible C24	UNIT_QJ71C24	PROTOCOL_SERIALMODEM	○
	Connected module: L series-compatible C24	UNIT_LJ71C24	PROTOCOL_SERIALMODEM	
	Connection target CPU: FXCPU	UNIT_FXCPU	PROTOCOL_TEL	
Gateway function communication		UNIT_A900GOT	PROTOCOL_UDPIP	×
GOT transparent function communication		UNIT_QNCPU, UNIT_LNCPU, UNIT_FXCPU, UNIT_QJ71C24, UNIT_LJ71C24, UNIT_GOT_QJ71E71, UNIT_GOT_QNETHER, UNIT_GOT_LNETHER, UNIT_A900GOT, UNIT_GOT_NZ2GF_ETB	PROTOCOL_SERIAL, PROTOCOL_USBGOT	○
		UNIT_GOTETHER_QNCPU, UNIT_GOTETHER_LNCPU, UNIT_GOTETHER_QBUS	PROTOCOL_TCPIP	
Q series bus communication		UNIT_QBF	PROTOCOL_QBF	○

○: Applicable, ×: Not applicable

(3) Feature

Read data from the buffer memory of special function module.

(4) Format (Dispatch interface)

(a) VBA

lRet = object.ReadBuffer(lStartIO, lAddress, lReadSize, iData(0))

Long	lRet	Returned value	Output
Long	lStartIO	Start I/O number of module from which values are read	Input
Long	lAddress	Buffer memory address	Input
Long	lReadSize	Read size	Input
Integer	iData(n)	Values read from buffer memory	Output

(b) Visual C++® .NET (MFC)

lRet = object.ReadBuffer(lStartIO, lAddress, lReadSize, *lpsData)

Long	lRet	Returned value	Output
Long	lStartIO	Start I/O number of module from which values are read	Input
Long	lAddress	Buffer memory address	Input
Long	lReadSize	Read size	Input
Short	*lpsData	Values read from buffer memory	Output

(c) VBScript

varRet = object.ReadBuffer(varStartIO, varAddress, varReadSize, lpvarData)

VARIANT	varRet	Returned value (LONG type)	Output
VARIANT	varStartIO	Start I/O number of module from which values are read (LONG type)	Input
VARIANT	varAddress	Buffer memory address (LONG type)	Input
VARIANT	varReadSize	Read size (LONG type)	Input
VARIANT	lpvarData	Values read from buffer memory (SHORT array type)	Output

(d) Visual Basic® .NET

IRet = object.ReadBuffer(iStartIO, iAddress, iReadSize, iData(0))

Integer	IRet	Returned value	Output
Integer	iStartIO	Start I/O number of module from which values are read	Input
Integer	iAddress	Buffer memory address	Input
Integer	iReadSize	Read size	Input
short	iData(n)	Values read from buffer memory	Output

(e) Visual C++® .NET

iRet = object.ReadBuffer(iStartIO, iAddress, iReadSize, *ipsData)

int	iRet	Returned value	Output
int	iStartIO	Start I/O number of module from which values are read	Input
int	iAddress	Buffer memory address	Input
int	iReadSize	Read size	Input
short	*ipsData	Values read from buffer memory	Output

(5) Format (Custom interface)

(a) Visual C++® .NET (MFC)

hResult = object.ReadBuffer(lStartIO, lAddress, lReadSize, *lpsData, *lplRetCode)

HRESULT	hResult	Returned value of COM	Output
LONG	lStartIO	Start I/O number of module from which values are read	Input
LONG	lAddress	Buffer memory address	Input
LONG	lReadSize	Read size	Input
SHORT	*lpsData	Values read from buffer memory	Output
LONG	*lplRetCode	Returned value of communication function	Output

(6) Description

- For the start I/O number of the module specified for lStartIO (varStartIO), specify the value of the actual start I/O number divided by 16.
- Buffer values of buffer memory address specified for lAddress (varAddress) of the special function module of the start I/O number specified for lStartIO (varStartIO) are read for the size of lReadSize (varReadSize).
- When performing CPU COM communication with FXCPU as a connected station or GX Simulator communication, specify the block number (0 to 7) of the special expansion equipment for the module's start I/O number and any value between 0 and 32767 for the buffer memory address.
- For lData (lplData or lpvarData), reserve arrays for more than the amount specified for lReadSize (varReadSize).

(7) Returned value

Normal termination : 0 is returned.
Abnormal termination: Any value other than 0 is returned. (☞ Page 378, CHAPTER 7 ERROR CODES)

(8) Considerations for reading/writing data from/to buffer memory in multiple CPU system configuration

Any of the following errors occurs when the function is executed under the situation where the configured multiple CPU system differs from the I/O assignment set in GX Developer.
Execute the function after checking the I/O assignment set in GX Developer and checking the I/O numbers of the module from/to which values are read/written.

- An error occurs even when the function is executed after the correct I/O numbers are specified.
- When the specific I/O numbers (I/O numbers whose I/O assignment is actually wrong) are specified, data can be read from buffer memory normally, but an error (error code: 0x010A4030, 0x010A4042, etc.) occurs when data are written to the buffer memory.
- Though a programmable controller CPU error (parameter error, SP. UNIT LAY ERR, etc.) does not occur in the multiple CPU system, an error occurs in the user application when the function is executed.

Point

- ● If a Q motion CPU is accessed, an error is returned.
- ● For iData (lpsData or lpvarData), reserve a memory area for the number of points specified for lReadSize (varReadSize). If the memory area is not reserved, a critical error (an application error or the like) may occur.
- ● When reading data from buffer memory (ReadBuffer) on QCPU (Q mode), the read operation can only be performed on the Q series-dedicated modules.
  Furthermore, data cannot be read from the shared memory of QCPU (Q mode).

5.2.10 WriteBuffer (Writing data to buffer memory)

(1) Applicable controls

Applicable to the Act(ML)ProgType control and the Act(ML)UtlType control.

(2) Applicable communication paths

The following table shows the applicable communication paths.

Communication path		Property setting		Applicability
		ActUnitType	ActProtocolType	
Serial communication	Connected module: Q series-compatible C24	UNIT_QJ71C24	PROTOCOL_SERIAL	○
	Connected module: L series-compatible C24	UNIT_LJ71C24		○
	Connected module: FX extended port	UNIT_FX485BD		×
Ethernet communication		UNIT_QJ71E71	PROTOCOL_TCPIP, PROTOCOL_UDPIP	○
		UNIT_LNETHER	PROTOCOL_TCPIP	
		UNIT_LNETHER, UNIT_LNETHER_DIRECT	PROTOCOL_UDPIP	
		UNIT_QNETHER	PROTOCOL_TCPIP	
		UNIT_QNETHER, UNIT_QNETHER_DIRECT	PROTOCOL_UDPIP	
		UNIT_NZ2GF_ETB	PROTOCOL_TCPIP	
		UNIT_NZ2GF_ETB UNIT_NZ2GF_ETB_DIRECT	PROTOCOL_UDPIP	
CPU COM communication	Connection target CPU: QCPU (Q mode)	UNIT_QNCPU	PROTOCOL_SERIAL	○
	Connection target CPU: LCPU	UNIT_LNCPU		○
	Connection target CPU: Q motion CPU	UNIT_QNMOTION		×
	Connection target CPU: FXCPU	UNIT_FXCPU		○*1
USB communication	Connection target CPU: QCPU (Q mode)	UNIT_QNUSB	PROTOCOL_USB	○
	Connection target CPU: LCPU	UNIT_LNUSB		○
	Connection target CPU: QSCPU	UNIT_QSUSB		×
	Connection target CPU: Q motion CPU	UNIT_QNMOTIONUSB		×
	Connection target CPU: FXCPU	UNIT_FXCPU		○
MELSECNET/H communication		UNIT_MNETHBOARD	PROTOCOL_MNETH	○*2
CC-Link IE Controller Network communication		UNIT_MNETGBOARD	PROTOCOL_MNETG	
CC-Link IE Field Network communication		UNIT_CCIEFBOARD	PROTOCOL_CCIEF	
CC-Link communication		UNIT_CCLINKBOARD	PROTOCOL_CCLINK	
CC-Link G4 communication		UNIT_G4QCPU	PROTOCOL_SERIAL	○
GX Simulator communication		UNIT_SIMULATOR	PROTOCOL_SHAREDMEMORY	○*3

○: Applicable, ×: Not applicable

*1 : An error is returned when the CPU is other than FX2N, FX2NC, FX3U, and FX3UC.

*2 : An error is returned when the own board is accessed.

*3 : An error is returned when the CPU is other than FX0N, FXU, FX2C, FX2N, and FX2NC.

Communication path		Property setting		Applicability
		ActUnitType	ActProtocolType	
GX Simulator2 communication		UNIT_SIMULATOR2	—	○
Modem communication	Connected module: Q series-compatible C24	UNIT_QJ71C24	PROTOCOL_SERIALMODEM	○
	Connected module: L series-compatible C24	UNIT_LJ71C24	PROTOCOL_SERIALMODEM	
	Connection target CPU: FXCPU	UNIT_FXCPU	PROTOCOL_TEL	
Gateway function communication		UNIT_A900GOT	PROTOCOL_UDPIP	×
GOT transparent function communication		UNIT_QNCPU, UNIT_LNCPU, UNIT_FXCPU, UNIT_QJ71C24, UNIT_LJ71C24, UNIT_GOT_QJ71E71, UNIT_GOT_QNETHER, UNIT_GOT_LNETHER, UNIT_A900GOT, UNIT_GOT_NZ2GF_ETB	PROTOCOL_SERIAL, PROTOCOL_USBGOT	○
		UNIT_GOTETHER_QNCPU, UNIT_GOTETHER_LNCPU, UNIT_GOTETHER_QBUS	PROTOCOL_TCPIP	
Q series bus communication		UNIT_QBF	PROTOCOL_QBF	○

○: Applicable, ×: Not applicable

(3) Feature

Write data to the buffer memory of special function module.

(4) Format (Dispatch interface)

(a) VBA

lRet = object.WriteBuffer(lStartIO, lAddress, lWriteSize, iData(0))

Long	lRet	Returned value	Output
Long	lStartIO	Start I/O number of module to which values are written	Input
Long	lAddress	Buffer memory address	Input
Long	lWriteSize	Write size	Input
Integer	iData(n)	Values written from buffer memory	Input

(b) Visual C++® .NET (MFC)

lRet = object.WriteBuffer(lStartIO, lAddress, lWriteSize, *lpsData)

Long	lRet	Returned value	Output
Long	lStartIO	Start I/O number of module to which values are written	Input
Long	lAddress	Buffer memory address	Input
Long	lWriteSize	Write size	Input
Short	*lpsData	Values written from buffer memory	Input

(c) VBScript

varRet = object.WriteBuffer(varStartIO, varAddress, varWriteSize, varData)

VARIANT	varRet	Returned value (LONG type)	Output
VARIANT	varStartIO	Start I/O number of module to which values are written (LONG type)	Input
VARIANT	varAddress	Buffer memory address (LONG type)	Input
VARIANT	varWriteSize	Write size (LONG type)	Input
VARIANT	varData	Values written from buffer memory (SHORT array type)	Input

(d) Visual Basic® .NET

IRet = object.WriteBuffer(iStartIO, iAddress, iWriteSize, iData(0))

Integer	IRet	Returned value	Output
Integer	iStartIO	Start I/O number of module to which values are written	Input
Integer	iAddress	Buffer memory address	Input
Integer	iWriteSize	Write size	Input
Short	iData(n)	Values written from buffer memory	Output

(e) Visual C++® .NET

iRet = object.WriteBuffer(iStartIO, iAddress, iWriteSize, *ipsData)

int	iRet	Returned value	Output
int	iStartIO	Start I/O number of module to which values are written	Input
int	iAddress	Buffer memory address	Input
int	iWriteSize	Write size	Input
Short	*ipsData	Values written from buffer memory	Output

(5) Format (Custom interface)

(a) Visual C++® .NET (MFC)

hResult = object.WriteBuffer(lStartIO, lAddress, lWriteSize, *lpsData, *lplRetCode)

HRESULT	hResult	Returned value of COM	Output
LONG	lStartIO	Start I/O number of module to where values are written	Input
LONG	lAddress	Buffer memory address	Input
LONG	lWriteSize	Write size	Input
SHORT	*lpsData	Values written from buffer memory	Input
LONG	*lplRetCode	Returned value of communication function	Output

(6) Description

- For the start I/O number of the module specified for lStartIO (varStartIO), specify the value of the actual start I/O number divided by 16.
- Buffer values of buffer memory address specified for lAddress (varAddress) of the special function module of the start I/O number specified for lStartIO (varStartIO) are written for the size of lWriteSize (varWriteSize).
- When performing CPU COM communication with FXCPU as a connected station or GX Simulator communication, specify the block number (0 to 7) of the special expansion equipment for the module's start I/O number and any value between 0 and 32767 for the buffer memory address.
- For lData (lpsData or varData), reserve arrays for more than the amount specified for lWriteSize (varWriteSize).

(7) Returned value

Normal termination : 0 is returned.
Abnormal termination: Any value other than 0 is returned. (☞ Page 378, CHAPTER 7 ERROR CODES)

(8) Considerations for reading/writing data from/to buffer memory in multiple CPU system configuration

Any of the following errors occurs when the function is executed under the situation where the configured multiple CPU system differs from the I/O assignment set in GX Developer.
Execute the function after checking the I/O assignment set in GX Developer and checking the I/O numbers of the module from/to which the values are read/written.

- An error occurs even when the function is executed after the correct I/O numbers are specified.
- When the specific I/O numbers (I/O numbers whose I/O assignment is actually wrong) are specified, data can be read from buffer memory normally, but an error (error code: 0x010A4030, 0x010A4042, etc.) occurs when data are written to the buffer memory.
- Though a programmable controller CPU error (parameter error, SP. UNIT LAY ERR, etc.) does not occur in the multiple CPU system, an error occurs in the user application when the function is executed.

Point

- ● If a Q motion CPU is accessed, an error is returned.
- ● For iData (lpsData, varData), reserve a memory area for the number of points specified for lWriteSize (varWriteSize). If the memory area is not reserved, a critical error (an application error or the like) may occur.
- ● When buffer memory write (WriteBuffer) is performed for QCPU (Q mode), write operation may be performed for only the Q series-dedicated module. Furthermore, data cannot be written to the shared memory of QCPU (Q mode).

5.2.11 GetClockData (Reading clock data)

(1) Applicable controls

Applicable to the Act(ML)ProgType control and the Act(ML)UtlType control.

(2) Applicable communication paths

The following table shows the applicable communication paths.

Communication path		Property setting		Applicability
		ActUnitType	ActProtocolType	
Serial communication	Connected module: Q series-compatible C24	UNIT_QJ71C24	PROTOCOL_SERIAL	○
	Connected module: L series-compatible C24	UNIT_LJ71C24		○
	Connected module: FX extended port	UNIT_FX485BD		○
Ethernet communication		UNIT_QJ71E71	PROTOCOL_TCPIP, PROTOCOL_UDPIP	○
		UNIT_LNETHER	PROTOCOL_TCPIP	
		UNIT_LNETHER, UNIT_LNETHER_DIRECT	PROTOCOL_UDPIP	
		UNIT_QNETHER	PROTOCOL_TCPIP	
		UNIT_QNETHER, UNIT_QNETHER_DIRECT	PROTOCOL_UDPIP	
		UNIT_NZ2GF_ETB	PROTOCOL_TCPIP	
		UNIT_NZ2GF_ETB UNIT_NZ2GF_ETB_DIRECT	PROTOCOL_UDPIP	
CPU COM communication	Connection target CPU: QCPU (Q mode)	UNIT_QNCPU	PROTOCOL_SERIAL	○
	Connection target CPU: LCPU	UNIT_LNCPU		○
	Connection target CPU: Q motion CPU	UNIT_QNMOTION		×
	Connection target CPU: FXCPU	UNIT_FXCPU		○
USB communication	Connection target CPU: QCPU (Q mode)	UNIT_QNUSB	PROTOCOL_USB	○
	Connection target CPU: LCPU	UNIT_LNUSB		○
	Connection target CPU: QSCPU	UNIT_QSUSB		○
	Connection target CPU: Q motion CPU	UNIT_QNMOTIONUSB		×
	Connection target CPU: FXCPU	UNIT_FXCPU		○
MELSECNET/H communication		UNIT_MNETHBOARD	PROTOCOL_MNETH	○[*1]
CC-Link IE Controller Network communication		UNIT_MNETGBOARD	PROTOCOL_MNETG	
CC-Link IE Field Network communication		UNIT_CCIEFBOARD	PROTOCOL_CCIEF	
CC-Link communication		UNIT_CCLINKBOARD	PROTOCOL_CCLINK	
CC-Link G4 communication		UNIT_G4QCPU	PROTOCOL_SERIAL	○
GX Simulator communication		UNIT_SIMULATOR	PROTOCOL_SHAREDMEMORY	○
GX Simulator2 communication		UNIT_SIMULATOR2	—	○

○: Applicable, ×: Not applicable

*1 : An error is returned when the own board is accessed.

Communication path		Property setting		Applicability
		ActUnitType	ActProtocolType	
Modem communication	Connected module: Q series-compatible C24	UNIT_QJ71C24	PROTOCOL_SERIALMODEM	○
	Connected module: L series-compatible C24	UNIT_LJ71C24	PROTOCOL_SERIALMODEM	
	Connection target CPU: FXCPU	UNIT_FXCPU	PROTOCOL_TEL	
Gateway function communication		UNIT_A900GOT	PROTOCOL_UDPIP	×
GOT transparent function communication		UNIT_QNCPU, UNIT_LNCPU, UNIT_FXCPU, UNIT_QJ71C24, UNIT_LJ71C24, UNIT_GOT_QJ71E71, UNIT_GOT_QNETHER, UNIT_GOT_LNETHER, UNIT_A900GOT, UNIT_GOT_NZ2GF_ETB	PROTOCOL_SERIAL, PROTOCOL_USBGOT	○
		UNIT_GOTETHER_QNCPU, UNIT_GOTETHER_LNCPU, UNIT_GOTETHER_QBUS	PROTOCOL_TCPIP	
Q series bus communication		UNIT_QBF	PROTOCOL_QBF	○

○: Applicable, ×: Not applicable

(3) Feature

Read clock data from a programmable controller CPU.

(4) Format (Dispatch interface)

(a) VBA

lRet = object.GetClockData(iYear, iMonth, iDay, iDayOfWeek, iHour, iMinute, iSecond)

Long	lRet	Returned value	Output
Integer	iYear	Read year value	Output
Integer	iMonth	Read month value	Output
Integer	iDay	Read day value	Output
Integer	iDayOfWeek(n)	Read day-of-week value	Output
Integer	iHour	Read hour value	Output
Integer	iMinute	Read minute value	Output
Integer	iSecond	Read second value	Output

(b) Visual C++® .NET (MFC)

lRet = object.GetClockData(*lpsYear, *lpsMonth, *lpsDay,*lpsDayOfWeek, *lpsHour, *lpsMinute, *lpsSecond)

Long	lRet	Returned value	Output
Short	*lpsYear	Read year value	Output
Short	*lpsMonth	Read month value	Output
Short	*lpsDay	Read day value	Output
Short	*lpsDayOfWeek	Read day-of-week value	Output
Short	*lpsHour	Read hour value	Output
Short	*lpsMinute	Read minute value	Output
Short	*lpsSecond	Read second value	Output

(c) VBScript

varRet = object.GetClockData(lpvarYear, lpvarMonth, lpvarDay, lpvarDayOfWeek, lpvarHour, lpvarMinute, lpvarSecond)

VARIANT	varRet	Returned value (LONG type)	Output
VARIANT	lpvarYear	Read year value (SHORT type)	Output
VARIANT	lpvarMonth	Read month value (SHORT type)	Output
VARIANT	lpvarDay	Read day value (SHORT type)	Output
VARIANT	lpvarDayOfWeek	Read day-of-week value (SHORT type)	Output
VARIANT	lpvarHour	Read hour value (SHORT type)	Output
VARIANT	lpvarMinute	Read minute value (SHORT type)	Output
VARIANT	lpvarSecond	Read second value (SHORT type)	Output

(d) Visual Basic® .NET

lRet = object.GetClockData(iYear, iMonth, iDay, iDayOfWeek, iHour, iMinute, iSecond)

Integer	lRet	Returned value	Output
short	iYear	Read year value	Output
short	iMonth	Read month value	Output
short	iDay	Read day value	Output
short	iDayOfWeek	Read day-of-week value	Output
short	iHour	Read hour value	Output
short	iMinute	Read minute value	Output
short	iSecond	Read second value	Output

(e) Visual C++® .NET

iRet = object.GetClockData(*lpsYear, *lpsMonth, *lpsDay, *lpsDayOfWeek, *lpsHour, *lpsMinute, *lpsSecond)

int	iRet	Returned value	Output
short	*lpsYear	Read year value	Output
short	*lps Month	Read month value	Output
short	*lps Day	Read day value	Output
short	*lps DayOfWeek	Read day-of-week value	Output
short	*lps Hour	Read hour value	Output
short	*lps Minute	Read minute value	Output
short	*lps Second	Read second value	Output

(5) Format (Custom interface)

(a) Visual C++® .NET (MFC)

```
hResult = object.GetClockData( *lpsYear, *lpsMonth, *lpsDay, *lpsDayOfWeek, *lpsHour, *lpsMinute,
          *lpsSecond, *lplRetCode )
```

HRESULT	hResult	Returned value of COM	Output
SHORT	*lpsYear	Read year value	Output
SHORT	*lpsMonth	Read month value	Output
SHORT	*lpsDay	Read day value	Output
SHORT	*lpsDayOfWeek	Read day-of-week value	Output
SHORT	*lpsHour	Read hour value	Output
SHORT	*lpsMinute	Read minute value	Output
SHORT	*lpsSecond	Read second value	Output
LONG	*lplRetCode	Returned value of communication function	Output

(6) Description

- An error is returned when the correct clock data is not set to the programmable controller CPU.
- The values stored in iYear (lpsYear or lpvarYear) are: four digits of the year for QCPU (Q mode), and last two digits of the year for any other CPUs.
  Note that the applicable years for QCPU (Q mode) are from 1980 to 2079.
- The values stored in iDayOfWeek (lpsDayOfWeek or lpvarDayOfWeek) are as follows.

Value	Day of Week
0	Sunday
1	Monday
2	Tuesday
3	Wednesday
4	Thursday
5	Friday
6	Saturday

(7) Returned value

Normal termination : 0 is returned.

Abnormal termination: Any value other than 0 is returned. (☞ Page 378, CHAPTER 7 ERROR CODES)

Point

- If a Q motion CPU is accessed, an error is returned.
- For FXCPU, clock data can be read from FX1N, FX1NC, FX1S, FX2N, or FX3G when it has a built-in clock, or from FXU, FX2C, or FX2NC when it is installed with the RTC cassette.
  An error is returned when clock data is read from FXCPU other than FX1N, FX1NC, FX1S, FXU, FX2C, FX2N, FX2NC, FX3U, FX3UC, and FX3G.
- Note that an error of transfer time may occur in clock setting.

5.2.12 SetClockData (Writing clock data)

(1) Applicable controls

Applicable to the Act(ML)ProgType control and the Act(ML)UtlType control.

(2) Applicable communication paths

The following table shows the applicable communication paths.

Communication path		Property setting		Applicability
		ActUnitType	ActProtocolType	
Serial communication	Connected module: Q series-compatible C24	UNIT_QJ71C24	PROTOCOL_SERIAL	○
	Connected module: L series-compatible C24	UNIT_LJ71C24		○
	Connected module: FX extended port	UNIT_FX485BD		○
Ethernet communication		UNIT_QJ71E71	PROTOCOL_TCPIP, PROTOCOL_UDPIP	○
		UNIT_LNETHER	PROTOCOL_TCPIP	
		UNIT_LNETHER, UNIT_LNETHER_DIRECT	PROTOCOL_UDPIP	
		UNIT_QNETHER	PROTOCOL_TCPIP	
		UNIT_QNETHER, UNIT_QNETHER_DIRECT	PROTOCOL_UDPIP	
		UNIT_NZ2GF_ETB	PROTOCOL_TCPIP	
		UNIT_NZ2GF_ETB UNIT_NZ2GF_ETB_DIRECT	PROTOCOL_UDPIP	
CPU COM communication	Connection target CPU: QCPU (Q mode)	UNIT_QNCPU	PROTOCOL_SERIAL	○
	Connection target CPU: LCPU	UNIT_LNCPU		○
	Connection target CPU: Q motion CPU	UNIT_QNMOTION		×
	Connection target CPU: FXCPU	UNIT_FXCPU		○
USB communication	Connection target CPU: QCPU (Q mode)	UNIT_QNUSB	PROTOCOL_USB	○
	Connection target CPU: LCPU	UNIT_LNUSB		○
	Connection target CPU: QSCPU	UNIT_QSUSB		×
	Connection target CPU: Q motion CPU	UNIT_QNMOTIONUSB		×
	Connection target CPU: FXCPU	UNIT_FXCPU		○
MELSECNET/H communication		UNIT_MNETHBOARD	PROTOCOL_MNETH	○[*1]
CC-Link IE Controller Network communication		UNIT_MNETGBOARD	PROTOCOL_MNETG	
CC-Link IE Field Network communication		UNIT_CCIEFBOARD	PROTOCOL_CCIEF	
CC-Link communication		UNIT_CCLINKBOARD	PROTOCOL_CCLINK	
CC-Link G4 communication		UNIT_G4QCPU	PROTOCOL_SERIAL	○
GX Simulator communication		UNIT_SIMULATOR	PROTOCOL_SHAREDMEMORY	○
GX Simulator2 communication		UNIT_SIMULATOR2	—	×

○: Applicable, ×: Not applicable

[*1] An error is returned when the own board is accessed.

Communication path		Property setting		Applicability
		ActUnitType	ActProtocolType	
Modem communication	Connected module: Q series-compatible C24	UNIT_QJ71C24	PROTOCOL_SERIALMODEM	○
	Connected module: L series-compatible C24	UNIT_LJ71C24	PROTOCOL_SERIALMODEM	
	Connection target CPU: FXCPU	UNIT_FXCPU	PROTOCOL_TEL	
Gateway function communication		UNIT_A900GOT	PROTOCOL_UDPIP	×
GOT transparent function communication		UNIT_QNCPU, UNIT_LNCPU, UNIT_FXCPU, UNIT_QJ71C24, UNIT_LJ71C24, UNIT_GOT_QJ71E71, UNIT_GOT_QNETHER, UNIT_GOT_LNETHER, UNIT_A900GOT, UNIT_GOT_NZ2GF_ETB	PROTOCOL_SERIAL, PROTOCOL_USBGOT	○
		UNIT_GOTETHER_QNCPU, UNIT_GOTETHER_LNCPU, UNIT_GOTETHER_QBUS	PROTOCOL_TCPIP	
Q series bus communication		UNIT_QBF	PROTOCOL_QBF	○

○: Applicable, ×: Not applicable

(3) Feature

Write clock data to a programmable controller CPU.

(4) Format (Dispatch interface)

(a) VBA

lRet = object.SetClockData(iYear, iMonth, iDay, iDayOfWeek, iHour, iMinute, iSecond)

Long	lRet	Returned value	Output
Integer	iYear	Year value to be written	Input
Integer	iMonth	Month value to be written	Input
Integer	iDay	Day value to be written	Input
Integer	iDayOfWeek(n)	Day-of-week value to be written	Input
Integer	iHour	Hour value to be written	Input
Integer	iMinute	Minute value to be written	Input
Integer	iSecond	Second value to be written	Input

(b) Visual C++® .NET (MFC)

lRet = object.SetClockData(sYear, sMonth, sDay, sDayOfWeek, sHour, sMinute, sSecond)

Long	lRet	Returned value	Output
Short	sYear	Year value to be written	Input
Short	sMonth	Month value to be written	Input
Short	sDay	Day value to be written	Input
Short	sDayOfWeek	Day-of-week value to be written	Input
Short	sHour	Hour value to be written	Input
Short	sMinute	Minute value to be written	Input
Short	sSecond	Second value to be written	Input

(c) VBScript

varRet = object.SetClockData(varYear, varMonth, varDay, varDayOfWeek,varHour, varMinute, varSecond)

VARIANT	varRet	Returned value (LONG type)	Output
VARIANT	varYear	Year value to be written (SHORT type)	Input
VARIANT	varMonth	Month value to be written (SHORT type)	Input
VARIANT	varDay	Day value to be written (SHORT type)	Input
VARIANT	varDayOfWeek	Day-of-week value to be written (SHORT type)	Input
VARIANT	varHour	Hour value to be written (SHORT type)	Input
VARIANT	varMinute	Minute value to be written (SHORT type)	Input
VARIANT	varSecond	Second value to be written (SHORT type)	Input

(d) Visual Basic® .NET

IRet = object.SetClockData(iYear, iMonth, iDay, iDayOfWeek, iHour, iMinute, iSecond)

Integer	IRet	Returned value	Output
short	iYear	Year value to be written	Input
short	iMonth	Month value to be written	Input
short	iDay	Day value to be written	Input
short	iDayOfWeek	Day-of-week value to be written	Input
short	iHour	Hour value to be written	Input
short	iMinute	Minute value to be written	Input
short	iSecond	Second value to be written	Input

(e) Visual C++® .NET

iRet = object.SetClockData(sYear, sMonth, sDay, sDayOfWeek, sHour, sMinute, sSecond)

int	iRet	Returned value	Output
short	sYear	Year value to be written	Input
short	sMonth	Month value to be written	Input
short	sDay	Day value to be written	Input
short	sDayOfWeek	Day-of-week value to be written	Input
short	sHour	Hour value to be written	Input
short	sMinute	Minute value to be written	Input
short	sSecond	Second value to be written	Input

(5) Format (Custom interface)

(a) Visual C++® .NET (MFC)

hResult = object.SetClockData(sYear,sMonth, sDay, sDayOfWeek, sHour, sMinute, sSecond,
*lplRetCode)

HRESULT	hResult	Returned value of COM	Output
SHORT	sYear	Year value to be written	Input
SHORT	sMonth	Month value to be written	Input
SHORT	sDay	Day value to be written	Input
SHORT	sDayOfWeek	Day-of-week value to be written	Input
SHORT	sHour	Hour value to be written	Input
SHORT	sMinute	Minute value to be written	Input
SHORT	sSecond	Second value to be written	Input
LONG	*lplRetCode	Returned value of communication function	Output

(6) Description

- An error is returned when the clock data to be set are not correct values.
- The applicable values to be specified for iYear (sYear or varYear) are: four digits of the year for QCPU (Q mode), and last two digits of the year for any other CPUs.
  Note that the applicable years for QCPU (Q mode) are from 1980 to 2079.
  An error occurs when four digits are set to a CPU other than QCPU (Q mode).
- The values to be specified for iDayOfWeek (sDayOfWeek or varDayOfWeek) are as follows.

Value	Day of Week
0	Sunday
1	Monday
2	Tuesday
3	Wednesday
4	Thursday
5	Friday
6	Saturday

(7) Returned value

Normal termination : 0 is returned.
Abnormal termination: Any value other than 0 is returned. (☞ Page 378, CHAPTER 7 ERROR CODES)

Point

- ● If a Q motion CPU is accessed, an error is returned.
- ● For FXCPU, clock data can be set to FX1N, FX1NC, FX1S, FX2N, or FX3G when it has a built-in clock, or to FXU, FX2C, or FX2NC when it is installed with the RTC cassette.
  An error is returned when clock data is set to FXCPU other than FX1N, FX1NC, FX1S, FXU, FX2C, FX2N, FX2NC, FX3U, FX3UC, and FX3G.
- ● Note that an error of transfer time may occur in clock setting.

5.2.13 GetCpuType (Reading programmable controller CPU model)

(1) Applicable controls

Applicable to the Act(ML)ProgType control and the Act(ML)UtlType control.

(2) Feature

Read the model character string and the model code of programmable controller CPU, network board, and GOT.

(3) Format (Dispatch interface)

(a) VBA

lRet = object.GetCpuType(szCpuName, lCpuType)

Long	lRet	Returned value	Output
String	szCpuName	Programmable controller CPU model character string	Output
Long	lCpuType	Programmable controller CPU model code	Output

(b) Visual C++® .NET (MFC)

lRet = object.GetCpuType(*szCpuName, *lplCpuType)

Long	lRet	Returned value	Output
BSTR	*szCpuName	Programmable controller CPU model character string	Output
Long	*lplCpuType	Programmable controller CPU model code	Output

(c) VBScript

varRet = object.GetCpuType(varCpuName, lpvarCpuCode)

VARIANT	varRet	Returned value (LONG type)	Output
VARIANT	varCpuName	Programmable controller CPU model character string (character string type)	Output
VARIANT	lpvarCpuCode	Programmable controller CPU model code (LONG type)	Output

(d) Visual Basic® .NET

lRet = object.GetCpuType(szCpuName, lCpuType)

Integer	lRet	Returned value	Output
String	szCpuName	Programmable controller CPU model character string	Output
Integer	lCpuType	Programmable controller CPU model code	Output

(e) Visual C++® .NET

iRet = object.GetCpuType (**szCpuName, *iplCpuType)

int	iRet	Returned value	Output
String	**szCpuName	Programmable controller CPU model character string	Output
int	*iplCpuType	Programmable controller CPU model code	Output

(4) Format (Custom interface)

(a) Visual C++® .NET (MFC)

hResult = object.GetCpuType(*szCpuName, *lplCpuType, *lplRetCode)

HRESULT	hResult	Returned value of COM	Output
BSTR	*szCpuName	Programmable controller CPU model character string	Output
LONG	*lplCpuType	Programmable controller CPU model code	Output
LONG	*lplRetCode	Returned value of communication function Output	Output

(5) Description

- The model and the model code of the communication target programmable controller CPU are stored in szCpuName (lpvarCpuName) and lCpuType (lplCpuType or lpvarCpuCode) respectively.
- The model character string of the programmable controller CPU is returned in UNICODE.

(6) Model character string and model code of CPU

The following tables show the model character strings and the model codes of programmable controller CPU, network board, and GOT, which are read by the GetCpuType function.

(a) Model character strings list

CPU/ network board/ GOT	Model character string	
	When CPU/own board is connected	When GX Simulator is connected
Q00JCPU	Q00JCPU	Q00JCPU
Q00UJCPU	Q00UJCPU	Q00UJCPU
Q00CPU	Q00CPU	Q00CPU
Q00UCPU	Q00UCPU	Q00UCPU
Q01CPU	Q01CPU	Q01CPU
Q01UCPU	Q01UCPU	Q01UCPU
Q02CPU	Q02CPU	Q02CPU
Q02HCPU	Q02HCPU	Q02HCPU
Q06HCPU	Q06HCPU	Q06HCPU
Q12HCPU	Q12HCPU	Q12HCPU
Q25HCPU	Q25HCPU	Q25HCPU
Q02PHCPU	Q02PHCPU	Q02PHCPU
Q06PHCPU	Q06PHCPU	Q06PHCPU
Q12PHCPU	Q12HCPU	Q12HCPU
Q25PHCPU	Q25HCPU	Q25HCPU
Q12PRHCPU	Q12PRHCPU	Q12PRHCPU
Q25PRHCPU	Q25PRHCPU	Q25PRHCPU
Q02UCPU	Q02UCPU	Q02UCPU
Q03UDCPU	Q03UDCPU	Q03UDCPU
Q04UDHCPU	Q04UDHCPU	Q04UDHCPU
Q06UDHCPU	Q06UDHCPU	Q06UDHCPU
Q10UDHCPU	Q10UDHCPU	Q10UDHCPU
Q13UDHCPU	Q13UDHCPU	Q13UDHCPU
Q20UDHCPU	Q20UDHCPU	Q20UDHCPU
Q26UDHCPU	Q26UDHCPU	Q26UDHCPU
Q03UDECPU	Q03UDECPU	Q03UDECPU
Q04UDEHCPU	Q04UDEHCPU	Q04UDEHCPU
Q06UDEHCPU	Q06UDEHCPU	Q06UDEHCPU
Q10UDEHCPU	Q10UDEHCPU	Q10UDEHCPU
Q13UDEHCPU	Q13UDEHCPU	Q13UDEHCPU
Q20UDEHCPU	Q20UDEHCPU	Q20UDEHCPU
Q26UDEHCPU	Q26UDEHCPU	Q26UDEHCPU
Q50UDEHCPU	Q50UDEHCPU	—
Q100UDEHCPU	Q100UDEHCPU	—
Q02CPU-A	Q02CPU	Q02CPU-A
Q02HCPU-A	Q02HCPU	Q02CPU-A
Q06HCPU-A	Q06HCPU	Q06HCPU-A
L02CPU	L02CPU	—
L26CPU-BT	L26CPU-BT	—
Q12DCCPU-V	Q12DCCPU-V	—
QS001CPU	QS001CPU	—
Q172CPU	Q172CPU	—
Q172HCPU	Q172HCPU	—
Q173CPU	Q173CPU	—
Q173HCPU	Q173HCPU	—
Q172DCPU	Q172DCPU	—
Q173DCPU	Q173DCPU	—
Q172DSCPU	Q172DSCPU	—
Q173DSCPU	Q173DSCPU	—
FX$_{0}$	FX$_{0}$/FX$_{0S}$	FX$_{0}$/FX$_{0S}$
FX$_{0S}$	FX$_{0}$/FX$_{0S}$	FX$_{0}$/FX$_{0S}$
FX$_{0N}$	FX$_{0N}$	FX$_{0N}$
FX$_{1}$	FX$_{1}$	FX$_{1}$
FX$_{1S}$	FX$_{1S}$	FX$_{1S}$
FX$_{1N}$	FX$_{1N}$	FX$_{1N}$
FX$_{1NC}$	FX$_{1N}$	FX$_{1N}$
FX$_{U}$	FX$_{U}$/FX$_{2C}$	FX$_{U}$/FX$_{2C}$
FX$_{2C}$	FX$_{U}$/FX$_{2C}$	FX$_{U}$/FX$_{2C}$
FX$_{2N}$	FX$_{2N}$/FX$_{2NC}$	FX$_{2N}$/FX$_{2NC}$
FX$_{2NC}$	FX$_{2N}$/FX$_{2NC}$	FX$_{2N}$/FX$_{2NC}$
FX$_{3G}$	FX$_{3G}$	FX$_{3G}$
FX$_{3U}$	FX$_{3UC}$	FX$_{3UC}$
FX$_{3UC}$	FX$_{3UC}$	FX$_{3UC}$
A80BD-J61BT11	A80BD-J61BT11	—
A80BD-J61BT13	A80BD-J61BT13	—
Q80BD-J71LP21-25	Q80BD-J71LP21-25	—
Q80BD-J71LP21G	Q80BD-J71LP21G	—
Q80BD-J71BR11	Q80BD-J71BR11	—
Q80BD-J71GF11-T2	Q80BD-J71GF11-T2	—
GOT	*1	—

*1 : A product model number of GOT is displayed.

(b) Model code list

CPU/ network board/ GOT	Model character string	
	When CPU/ own board is connected	When GX Simulator is connected
Q00JCPU	250H	250H
Q00UJCPU	260H	260H
Q00CPU	251H	251H
Q00UCPU	261H	261H
Q01CPU	252H	252H
Q01UCPU	262H	262H
Q02CPU	41H	41H
Q02HCPU	41H	41H
Q06HCPU	42H	42H
Q12HCPU	43H	43H
Q25HCPU	44H	44H
Q02PHCPU	41H	41H
Q06PHCPU	42H	42H
Q12PHCPU	43H	43H
Q25PHCPU	44H	44H
Q12PRHCPU	4BH	4BH
Q25PRHCPU	4CH	4CH
Q02UCPU	263H	263H
Q03UDCPU	268H	268H
Q04UDHCPU	269H	269H
Q06UDHCPU	26AH	26AH
Q10UDHCPU	266H	266H
Q13UDHCPU	26BH	26BH
Q20UDHCPU	267H	267H
Q26UDHCPU	26CH	26CH
Q03UDECPU	268H	268H
Q04UDEHCPU	269H	269H
Q06UDEHCPU	26AH	26AH
Q10UDEHCPU	266H	266H
Q13UDEHCPU	26BH	26BH
Q20UDEHCPU	267H	267H
Q26UDEHCPU	26CH	26CH
Q50UDEHCPU	26DH	—
Q100UDEHCPU	26EH	—
Q02CPU-A	141H	141H
Q02HCPU-A	141H	141H
Q06HCPU-A	142H	142H
L02CPU	541H	—
L26CPU-BT	542H	—
Q12DCCPU-V	2043H	—
QS001CPU	230H	—
Q172CPU	2010H	—
Q173CPU	2011H	—
Q172HCPU	2012H	—
Q173HCPU	2013H	—
Q172DCPU	2014H	—
Q173DCPU	2015H	—
Q172DSCPU	2018H	—
Q173DSCPU	2019H	—
FX0	F0H	F0H
FX0S	F0H	F0H
FX0N	8EH	8EH
FX1	F1H	F1H
FX1S	F2H	F2H
FX1N	9EH	9EH
FX1NC	9EH	9EH
FXU	8DH	8DH
FX2C	8DH	8DH
FX2N	9DH	9DH
FX2NC	9DH	9DH
FX3G	F4H	F4H
FX3U	F3H	F3H
FX3UC	F3H	F3H
A80BD-J61BT11	90H	—
A80BD-J61BT13	90H	—
Q80BD-J71LP21-25	90H	—
Q80BD-J71LP21G	90H	—
Q80BD-J71BR11	90H	—
Q80BD-J71GF11-T2	90H	—
GOT	E340H	—

(7) Returned value

Normal termination : 0 is returned.

Abnormal termination: Any value other than 0 is returned. (☞ Page 378, CHAPTER 7 ERROR CODES)

5.2.14 SetCpuStatus (Remote control)

(1) Applicable controls

Applicable to the Act(ML)ProgType control and the Act(ML)UtlType control.

(2) Applicable communication paths

The following table shows the applicable communication paths.

Communication path		Property setting		Applicability
		ActUnitType	ActProtocolType	
Serial communication	Connected module: Q series-compatible C24	UNIT_QJ71C24	PROTOCOL_SERIAL	○
	Connected module: L series-compatible C24	UNIT_LJ71C24		○
	Connected module: FX extended port	UNIT_FX485BD		○*1 *2
Ethernet communication		UNIT_QJ71E71	PROTOCOL_TCPIP, PROTOCOL_UDPIP	○
		UNIT_LNETHER	PROTOCOL_TCPIP	
		UNIT_LNETHER, UNIT_LNETHER_DIRECT	PROTOCOL_UDPIP	
		UNIT_QNETHER	PROTOCOL_TCPIP	
		UNIT_QNETHER, UNIT_QNETHER_DIRECT	PROTOCOL_UDPIP	
		UNIT_NZ2GF_ETB	PROTOCOL_TCPIP	
		UNIT_NZ2GF_ETB UNIT_NZ2GF_ETB_DIRECT	PROTOCOL_UDPIP	
CPU COM communication	Connection target CPU: QCPU (Q mode)	UNIT_QNCPU	PROTOCOL_SERIAL	○
	Connection target CPU: LCPU	UNIT_LNCPU		○
	Connection target CPU: Q motion CPU	UNIT_QNMOTION		○
	Connection target CPU: FXCPU	UNIT_FXCPU		○*1 *2
USB communication	Connection target CPU: QCPU (Q mode)	UNIT_QNUSB	PROTOCOL_USB	○
	Connection target CPU: LCPU	UNIT_LNUSB		○
	Connection target CPU: QSCPU	UNIT_QSUSB		×
	Connection target CPU: Q motion CPU	UNIT_QNMOTIONUSB		○
	Connection target CPU: FXCPU	UNIT_FXCPU		○
MELSECNET/H communication		UNIT_MNETHBOARD	PROTOCOL_MNETH	○*3
CC-Link IE Controller Network communication		UNIT_MNETGBOARD	PROTOCOL_MNETG	
CC-Link IE Field Network communication		UNIT_CCIEFBOARD	PROTOCOL_CCIEF	
CC-Link communication		UNIT_CCLINKBOARD	PROTOCOL_CCLINK	
CC-Link G4 communication		UNIT_G4QCPU	PROTOCOL_SERIAL	○
GX Simulator communication		UNIT_SIMULATOR	PROTOCOL_SHAREDMEMORY	○

○: Applicable, ×: Not applicable

*1 : An error is returned when PAUSE is specified.

*2 : An error is returned when the CPU is other than FX_{1N}, FX_{1NC}, FX_{2N}, FX_{2NC}, FX_{3U}, and FX_{3UC}.

*3 : An error is returned when the own board is accessed.

Communication path		Property setting		Applicability
		ActUnitType	ActProtocolType	
GX Simulator2 communication		UNIT_SIMULATOR2	—	×
Modem communication	Connected module: Q series-compatible C24	UNIT_QJ71C24	PROTOCOL_SERIALMODEM	○
	Connected module: L series-compatible C24	UNIT_LJ71C24	PROTOCOL_SERIALMODEM	
	Connection target CPU: FXCPU	UNIT_FXCPU	PROTOCOL_TEL	
Gateway function communication		UNIT_A900GOT	PROTOCOL_UDPIP	×
GOT transparent function communication		UNIT_QNCPU, UNIT_LNCPU, UNIT_FXCPU, UNIT_QJ71C24, UNIT_LJ71C24, UNIT_GOT_QJ71E71, UNIT_GOT_QNETHER, UNIT_GOT_LNETHER, UNIT_A900GOT, UNIT_GOT_NZ2GF_ETB	PROTOCOL_SERIAL, PROTOCOL_USBGOT	○
		UNIT_GOTETHER_QNCPU, UNIT_GOTETHER_LNCPU, UNIT_GOTETHER_QBUS	PROTOCOL_TCPIP	
Q series bus communication		UNIT_QBF	PROTOCOL_QBF	○

○: Applicable, ×: Not applicable

(3) Feature

Perform a remote operation of programmable controller CPU.

(4) Format (Dispatch interface)

(a) VBA

lRet = object.SetCpuStatus(lOperation)

Long	lRet	Returned value	Output
Long	lOperation	Remote RUN/STOP/PAUSE	Input

(b) Visual C++® .NET (MFC)

lRet = object.SetCpuStatus(lOperation)

Long	lRet	Returned value	Output
Long	lOperation	Remote RUN/STOP/PAUSE	Input

(c) VBScript

varRet = object.SetCpuStatus(varOperation)

VARIANT	varRet	Returned value (LONG type)	Output
VARIANT	varOperation	Remote RUN/STOP/PAUSE (LONG type)	Input

(d) Visual Basic® .NET

lRet = object.SetCpuStatus(lOperation)

Integer	lRet	Returned value	Output
Integer	lOperation	Remote RUN/STOP/PAUSE	Input

(e) Visual C++® .NET

iRet = object.SetCpuStatus (iOperation)

int	iRet	Returned value	Output
int	iOperation	Remote RUN/STOP/PAUSE	Input

(5) Format (Custom interface)

(a) Visual C++® .NET (MFC)

hResult = object.SetCpuStatus(lOperation, *lplRetCode)

HRESULT	hResult	Returned value of COM	Output
LONG	lOperation	Remote RUN/STOP/PAUSE	Input
LONG	*lplRetCode	Returned value of communication function Output	Output

(6) Description

The operation specified for lOperation (varOperation) is performed.

An error occurs when a value other than the following values is specified.

Value	Operation
0	Remote RUN
1	Remote STOP
2	Remote PAUSE

(7) Returned value

Normal termination : 0 is returned.

Abnormal termination: Any value other than 0 is returned. (☞ Page 378, CHAPTER 7 ERROR CODES)

Point

- Since FXCPU does not have the PAUSE switch as a programmable controller CPU, an error is returned if a remote pause operation is specified using the SetCpuStatus function.
- If a Q motion CPU is accessed and PAUSE is specified, an error is returned.

5.2.15 EntryDeviceStatus (Registering devices for status monitoring)

(1) Applicable controls

Applicable to the Act(ML)ProgType control and the Act(ML)UtlType control.

(2) Feature

Register devices whose status to be monitored.

(3) Format (Dispatch interface)

(a) VBA

lRet = object.EntryDeviceStatus(szDeviceList, lSize, lMonitorCycle, lData(0))

Long	lRet	Returned value	Output
String	szDeviceList	Registered device name list	Input
Long	lSize	Number of registered device points	Input
Long	lMonitorCycle	Status monitoring time interval	Input
Long	lData(n)	Registered device value list	Input

(b) Visual C++® .NET (MFC)

lRet = object.EntryDeviceStatus(szDeviceList, lSize, lMonitorCycle, *lplData)

Long	lRet	Returned value	Output
CString	szDeviceList	Registered device name list	Input
Long	lSize	Number of registered device points	Input
Long	lMonitorCycle	Status monitoring time interval	Input
Long	*lplData	Registered device value list	Input

(c) VBScript

varRet = object.EntryDeviceStatus(varDeviceList, varSize, varMonitorCycle, varData)

VARIANT	varRet	Returned value (LONG type)	Output
VARIANT	varDeviceList	Registered device name list (BSTR type)	Input
VARIANT	varSize	Number of registered device points (LONG type)	Input
VARIANT	varMonitorCycle	Status monitoring time interval (LONG type)	Input
VARIANT	varData	Registered device value list (LONG type)	Input

(d) Visual Basic® .NET

lRet = object.EntryDeviceStatus(szDeviceList, lSize,lMonitorCycle, lData(0))

Integer	lRet	Returned value	Output
String	szDeviceList	Registered device name list	Input
Integer	lSize	Number of registered device points	Input
Integer	lMonitorCycle	Status monitoring time interval	Input
Integer	lData(n)	Registered device value list	Input

(e) Visual C++® .NET

iRet = object.EntryDeviceStatus(szDeviceList, iSize, iMonitorCycle, *iplData)

int	iRet	Returned value	Output
String	*szDeviceList	Registered device name list	Input
int	iSize	Number of registered device points	Input
int	iMonitorCycle	Status monitoring time interval	Input
int	*iplData	Registered device value list	Input

(4) Format (Custom interface)

(a) Visual C++® .NET (MFC)

hResult = object.EntryDeviceStatus(szDeviceList, lSize,lMonitorCycle, *lplData, *lplRetCode)

HRESULT	hResult	Returned value of COM	Output
BSTR	szDeviceList	Registered device name list	Input
LONG	lSize	Number of registered device points	Input
LONG	lMonitorCycle	Status monitoring time interval	Input
LONG	*lplData	Registered device value list	Input
LONG	*lplRetCode	Returned value of communication function	Output

(5) Description

- A device group for the size of lSize (varSize) specified for szDeviceList (varDeviceList) is checked whether it is in the status specified for lData (lplData or varData).
  Specify the check time for lMonitorCycle (varMonitorCycle).
  When the status is established, the OnDeviceStatus function of the user application is executed.
- Use the line feed symbol to separate the character string specified for the device list. The line feed symbol is not necessary to be suffixed to the last device.
  Example

When using Visual Basic®.NET ,VBA,VBScript:	"D0" & vbLf & "D1" & vbLf & "D2"
When using Visual C++®.NET:	D0\nD1\nD2

- The maximum number of device points that can be specified for lSize (varSize) is 20 points.
- Specify a value within the range from 1 second to 1 hour (set between 1 and 3600 in seconds) for lMonitorCycle (varMonitorCycle).
  An error occurs when any other value outside the above range is specified.
- The registered device value list is stored in lData (lplData or varData).

(6) How to specify devices

The following describes how to specify devices.

Example 1: When devices are specified as follows (Number of points: 3 points)

When using Visual Basic®.NET ,VBA,VBScript:	M0 & vbLf & D0 & vbLf & K8M0
When using Visual C++®.NET:	M0\nD0\nK8M0

Upper 2 bytes	Lower 2 bytes
Not used (0 is stored.)	M0
	D0
M16 to M31*1	M0 to M15*1

Example 2: When devices including FXCPU devices of CN200 and later are specified (Total number of points: 3 points)*2

When using Visual Basic®.NET ,VBA,VBScript:	D0 & vbLf & CN200 & vbLf & D1
When using Visual C++®.NET:	D0\nCN200\nD1

Upper 2 bytes	Lower 2 bytes
Not used (0 is stored.)	D0
H of CN200	L of CN200
Not used (0 is stored.)	D1

Example 3: When devices including FD are specified (Total number of points: 3 points)

When using Visual Basic®.NET ,VBA,VBScript:	D0 & vbLf & FD0 & vbLf & D1
When using Visual C++®.NET:	D0\nFD0\nD1

Upper 2 bytes	Lower 2 bytes
Not used (0 is stored.)	D0
	LL of FD0
	D1

Example 4: When 8-bit devices including EG are specified (Total number of points: 3 points)

The following is the example when 8-bit devices (E0000 and E0001 of SHARP programmable controller) are assigned to EG0.

When using Visual Basic®.NET ,VBA,VBScript:	D0 & vbLf & EG0 & vbLf & D1
When using Visual C++®.NET:	D0\nEG0\nD1

<table>
<tr><th>Upper 2 bytes</th><th colspan="2">Lower 2 bytes</th></tr>
<tr><td rowspan="4">Not used
(0 is stored.)</td><td colspan="2">D0</td></tr>
<tr><td colspan="2">EG0</td></tr>
<tr><td>(E0001)</td><td>(E0000)</td></tr>
<tr><td colspan="2">D1</td></tr>
</table>

*1 : Devices are stored from the lower bit in the order of device number.

*2 : For CN200 or later of FXCPU, 2 words are read for each point when reading devices randomly.

(7) Returned value

Normal termination : 0 is returned.

Abnormal termination: Any value other than 0 is returned. (☞ Page 378, CHAPTER 7 ERROR CODES)

(8) Considerations for checking word device status

When checking the word device status for negative values of -1 to -32768 (FFFFH to 8000H), set the monitor device value of the EntryDeviceStatus function to any of 65535 to 32768 (0000FFFFH to 00008000H) where "0" is stored in the upper 2 bytes.

Example: When checking the D0 status for "-10"

Set the value "65526 (0000FFF6H)" where "0" is stored in the upper 2 bytes of "-10 (FFFFFFF6H)" for the monitor device value.

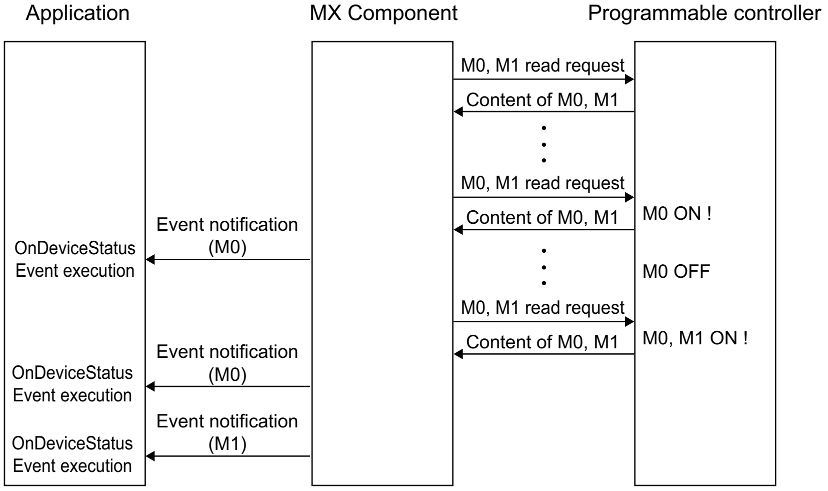
While the type of word devices of the programmable controller CPU is WORD type, the type of monitor device value of the EntryDeviceStatus function is LONG type. Therefore, when current values of programmable controller CPU are compared with monitor device values of the EntryDeviceStatus function, the values do not match and the above setting is required. (When bit devices or double word devices are used, this consideration does not apply.)

For the programming examples regarding this consideration, refer to the following section.

☞ Page 412, Appendix 3 Time-Out Periods

Point

- Device status monitoring may not be performed at the specified status monitoring time intervals depending on the conditions: personal computer performance, currently executed application load, time required for communication with the programmable controller, or the like. Simultaneous use of any other control functions would also be the cause of disabling device status monitoring at the specified status monitoring time intervals.
- For lData (lplData or lpvarData), reserve a memory area for the number of points specified for lSize (varSize). If the memory area is not reserved, a critical error (an application error or the like) may occur.
- An error occurs when the EntryDeviceStatus function is executed during a status monitoring. When changing any status monitor condition, execute the FreeDeviceStatus function and then execute the EntryDeviceStatus function.
- When the status of multiple devices changes at the same time, the OnDeviceStatus event is executed every time the status changes.
(Example: When M0 is monitored)



- This function is a function to check the status establishment under the constant execution of random device read by the control. This function is not a function for a programmable controller CPU to notify the device status establishment to MX Component. Therefore, the control may not be able to check the device status establishment of programmable controller CPU depending on the specified status monitoring time interval.

5.2.16 FreeDeviceStatus (Deregistering devices for status monitoring)

(1) Applicable controls

Applicable to the Act(ML)ProgType control and the Act(ML)UtlType control.

(2) Feature

Deregister devices that are registered using the EntryDeviceStatus function to monitor their status.

(3) Format (Dispatch interface)

(a) Visual C++® .NET (MFC), VBA

lRet = object.FreeDeviceStatus()

Long	lRet	Returned value	Output

(b) VBScript

varRet = object.FreeDeviceStatus()

VARIANT	varRet	Returned value (LONG type)	Output

(c) Visual Basic® .NET

IRet = object.FreeDeviceStatus()

Integer	IRet	Returned value	Output

(d) Visual C++® .NET

iRet = object.FreeDeviceStatus()

int	iRet	Returned value	Output

(4) Format (Custom interface)

(a) Visual C++® .NET (MFC)

hResult = object.FreeDeviceStatus(*lplRetCode)

HRESULT	hResult	Returned value of COM	Output
LONG	*lplRetCode	Returned value of communication function	Output

(5) Description

The devices that are set using the EntryDeviceStatus function to monitor their status are deregistered.

(6) Returned value

Normal termination : 0 is returned.
Abnormal termination: Any value other than 0 is returned. (☞ Page 378, CHAPTER 7 ERROR CODES)

5.2.17 OnDeviceStatus (Event notification)

(1) Applicable controls

Applicable to the Act(ML)ProgType control and the Act(ML)UtlType control.

(2) Feature

Execute the event notification when the device condition registered using the EntryDeviceStatus function is satisfied.

(3) Format (Dispatch interface)

(a) VBA

object.OnDeviceStatus(szDevice, lData, lReturnCode)

String	szDevice	Name of device whose condition is satisfied	Input
Long	lData	Name of device whose condition is satisfied	Input
Long	lReturnCode	Returned value of condition check processing	Input

(b) Visual C++® .NET (MFC)

object.OnDeviceStatus(*szDevice, lData, lReturnCode)

LPCTSTR	*szDevice	Name of device whose condition is satisfied	Input
Long	lData	Name of device whose condition is satisfied	Input
Long	lReturnCode	Returned value of condition check processing	Input

(c) VBScript

object.OnDeviceStatus(varDevice, varData, varReturnCode)

VARIANT	varDevice	Name of device whose condition is satisfied (BSTR type)	Input
VARIANT	varData	Name of device whose condition is satisfied (LONG type)	Input
VARIANT	varReturnCode	Returned value of condition check processing (LONG type)	Input

(d) Visual Basic® .NET

Private Sub object_OnDeviceStatus(ByVal sender As System.Object, ByVal e As objectLib. _objectIEvents_OnDeviceStatusEvent)

sender	Event occurrence source		Output
e	Event data		Output
	The following are the members of e.		
	e.szDevice	Name of device whose condition is satisfied	
	e.lReturnCode	Value of device whose condition is satisfied	
	e.lData	Returned value of condition check processing	

(e) Visual C++® .NET

private: System::Void object_OnDeviceStatus (System::Object * sender,
objectLib. ::_objectIEvents_OnDeviceStatusEvent * e)

sender	Event occurrence source		Output
e	Event data		Output
	The following are the members of e.		
	e->szDevice	Name of device whose condition is satisfied	
	e->lReturnCode	Value of device whose condition is satisfied	
	e->lData	Returned value of condition check processing	

(4) Format (Custom interface)

(a) Visual C++® .NET (MFC)

object.OnDeviceStatus(szDevice, lData, lReturnCode, *lplRetCode)

LPCTSTR	szDevice	Name of device whose condition is satisfied	Input
LONG	lData	Name of device whose condition is satisfied	Input
LONG	lReturnCode	Returned value of condition check processing	Input
LONG	*lplRetCode	Returned value of communication function	Output

(5) Description

- The event is notified to the application when the device condition registered using the EntryDeviceStatus function is satisfied.
  Programming this function in the user application allows the application to receive the event when the registered device condition is satisfied.
- Device values registered using the EntryDeviceStatus function are input to lData (varData).
  Example: When the word device is monitored for the value of "-1"
  Set 65535 (0000FFFFH) as a registered device value using the EntryDeviceStatus function.
  When the value of the target word device of the programmable controller CPU becomes "-1" (FFFFH), the OnDeviceStatus function is executed and (0000FFFFH) is input to lData (varData).

(6) Returned value

None

Point

- When any of the following settings is set in the user application, the event of the OnDeviceStatus function does not occur even if the condition of the device registered to the EntryDeviceStatus function is satisfied. Note that when the event occurrence becomes in wait status, the control function is not returned to the control and device management processing stops until the following setting is terminated.
  - User applications created using Visual Basic® or VBA (Excel)
    The message box is displayed in the user application.
    The InputBox/OutputBox is displayed in the user application.
  - User applications created using Visual Basic® , Visual C++® , VBA (Excel, Access) or VBScript
    The Sleep processing, WaitForSingleObject function, or similar standby function is used in the user application.
- For installing Reference, refer to the sample programs for Reference for VB .NET/VC .NET. (☞ Page 363, Section 6.4, Page 370, Section 6.5)

5.2.18 ReadDeviceBlock2 (Reading devices in bulk)

(1) Applicable controls

Applicable to the Act(ML)ProgType control and the Act(ML)UtlType control.

(2) Feature

Read devices in 2-byte data unit in bulk.

(3) Format (Dispatch interface)

(a) VBA

lRet = object.ReadDeviceBlock2(szDevice, lSize, iData(0))

Long	lRet	Returned value	Output
String	szDevice	Device name	Input
Long	lSize	Number of read points	Input
Integer	iData(n)	Read device value	Output

(b) Visual C++® .NET (MFC)

lRet = object.ReadDeviceBlock2(szDevice, lSize, *lpsData)

Long	lRet	Returned value	Output
CString	szDevice	Device name	Input
Long	lSize	Number of read points	Input
Short	*lpsData	Read device value	Output

(c) VBScript

varRet = object.ReadDeviceBlock2(varDevice, varSize, lpvarData)

VARIANT	varRet	Returned value (LONG type)	Output
VARIANT	varDevice	Device name (character string type)	Input
VARIANT	varSize	Number of read points (LONG type)	Input
VARIANT	lpvarData	Read device value (SHORT type)	Output

(d) Visual Basic® .NET

lRet = object.ReadDeviceBlock2(szDevice, lSize, sData(0))

Integer	lRet	Returned value	Output
String	szDevice	Device name	Input
Integer	lSize	Number of read points	Input
short	sData(n)	Read device value	Output

(e) Visual C++® .NET

iRet = object.ReadDeviceBlock2(*szDevice, iSize, *lpsData)

int	iRet	Returned value	Output
String	*szDevice	Device name	Input
int	iSize	Number of read points	Input
short	*lpsData	Read device value	Output

(4) Format (Custom interface)

(a) Visual C++® .NET (MFC)

hResult = object.ReadDeviceBlock2(szDevice, lSize, *lpsData, *lplRetCode)

HRESULT	hResult	Returned value of COM	Output
BSTR	szDevice	Device name	Input
LONG	lSize	Number of read points	Input
SHORT	*lpsData	Read device value	Output
LONG	*lplRetCode	Returned value of communication function	Output

(5) Description

- The device values of lSize (varSize) are read in bulk starting from the device specified for szDevice (varDevice) as SHORT type data.
- The read device values are stored in iData (lpsData or lpvarData).
- For lData (lpsData or lpvarData), reserve arrays for more than the amount specified for lSize (varSize).

(6) How to specify devices

The following describes how to specify devices.

<When bit device is specified>

Example: Read 3 points (3 words) of data in 16-point unit starting from M0.

2 Bytes
M0 to M15[*1]
M16 to M31[*1]
M32 to M47[*1]

<When word device is specified>

Example: Read 3 points of data starting from D0.

2 Bytes
D0
D1
D2

<When FXCPU devices of CN200 and later are specified>

Example: Read 6 points of data starting from CN200.[*2]

2 Bytes
L of CN200 (Lower 2 bytes)
H of CN200 (Upper 2 bytes)
L of CN201 (Lower 2 bytes)
H of CN201 (Upper 2 bytes)
L of CN202 (Lower 2 bytes)
H of CN202 (Upper 2 bytes)

<When FD device is specified (4-word device)>

Example: Read 6 points of data starting from FD0.

2 Bytes
LL of FD0
LH of FD0
HL of FD0
HH of FD0
LL of FD1
LH of FD1

[*1] : Devices are stored from the lower bit in the order of device number.

[*2] : For CN200 or later of FXCPU, 2 words are read for each 2 points. Reading only 1 point will result in an error.

<8-bit devices assigned to gateway devices>

Example: Read 4 points of data starting from EG0.
(When 8-bit devices (from E0000 to E0007 of SHARP programmable controller) are assigned from EG0 to EG3.)

2 Bytes	
EG0	
(E0001)	(E0000)
EG1	
(E0003)	(E0002)
EG2	
(E0005)	(E0004)
EG3	
(E0007)	(E0006)

(7) Returned value

Normal termination : 0 is returned.
Abnormal termination: Any value other than 0 is returned. (☞ Page 378, CHAPTER 7 ERROR CODES)

Point

- The maximum number of read points that can be specified for lSize (varSize) should be the value which satisfies the following condition.
  Read starting device number + Number of read points ≦ Last device number
- When specifying bit devices, a multiple of 16 can be specified as a device number.
- For iData (lpsData or lpvarData), reserve a memory area for the number of points specified for lSize (varSize).
  If the memory area is not reserved, a critical error (an application error or the like) may occur.

5.2.19 WriteDeviceBlock2 (Writing devices in bulk)

(1) Applicable controls

Applicable to the Act(ML)ProgType control and the Act(ML)UtlType control.

(2) Feature

Write devices in 2-byte data unit in bulk.

(3) Format (Dispatch interface)

(a) VBA

lRet = object.WriteDeviceBlock2(szDevice, lSize, iData(0))

Long	lRet	Returned value	Output
String	szDevice	Device name	Input
Long	lSize	Number of write points	Input
Integer	iData(n)	Device value to be written	Input

(b) Visual C++® .NET (MFC)

lRet = object.WriteDeviceBlock2(szDevice, lSize, *lpsData)

Long	lRet	Returned value	Output
CString	szDevice	Device name	Input
Long	lSize	Number of write points	Input
Short	*lpsData	Device value to be written	Input

(c) VBScript

varRet = object.WriteDeviceBlock2(varDevice, varSize, varData)

VARIANT	varRet	Returned value (LONG type)	Output
VARIANT	varDevice	Device name (character string type)	Input
VARIANT	varSize	Number of write points (LONG type)	Input
VARIANT	varData	Device value to be written (SHORT type)	Input

(d) Visual Basic® .NET

lRet = object.WriteDeviceBlock2(szDevice,lSize, sData(0))

Integer	lRet	Returned value	Output
String	szDevice	Device name	Input
Integer	lSize	Number of write points	Input
short	sData(n)	Device value to be written	Input

(e) Visual C++® .NET

iRet = object.WriteDeviceBlock2(*szDevice, iSize, *lpsData)

int	iRet	Returned value	Output
String	*szDevice	Device name	Input
int	iSize	Number of write points	Input
short	*lpsData	Device value to be written	Input

(4) Format (Custom interface)

(a) Visual C++® .NET (MFC)

hResult = object.WriteDeviceBlock2(szDevice, lSize, *lpsData, *lplRetCode)

HRESULT	hResult	Returned value of COM	Output
BSTR	szDevice	Device name	Input
LONG	lSize	Number of write points	Input
SHORT	*lpsData	Device value to be written	Input
LONG	*lplRetCode	Returned value of communication function	Output

(5) Description

- The device values of lSize (varSize) are written in bulk starting from the device specified for szDevice (varDevice).
- The device values to be written are stored in lData (lplData or varData).
- For lData (lpsData or varData), reserve arrays for more than the amount specified for lSize (varSize).

(6) How to specify devices

The following describes how to specify devices.

<When bit device is specified>

Example: Write 3 points (3 words) of data in 16-point unit starting from M0.

2 bytes
M0 to M15*1
M16 to M31*1
M32 to M47*1

<When word device is specified>

Example: Write 3 points of data starting from D0.

2 bytes
D0
D1
D2

<When FXCPU devices of CN200 and later are specified>

Example: Write 6 points of data starting from CN200.*2

2 bytes
L of CN200 (Lower 2 bytes)
H of CN200 (Upper 2 bytes)
L of CN201 (Lower 2 bytes)
H of CN201 (Upper 2 bytes)
L of CN202 (Lower 2 bytes)
H of CN202 (Upper 2 bytes)

<When FD device is specified (4-word device)>

Example: Write 6 points of data starting from FD0.

2 bytes
LL of FD0
LH of FD0
HL of FD0
HH of FD0
LL of FD1
LH of FD1

*1 : Devices are stored from the lower bit in the order of device number.

*2 : For CN200 or later of FXCPU, 2 words are written for each 2 points. Writing only 1 point will result in an error.

<8-bit devices assigned to gateway devices>

Example: Write 4 points of data starting from EG0.
(When 8-bit devices (from E0000 to E0007 of SHARP programmable controller) are assigned from EG0 to EG3.)

2 bytes	
EG0	
(E0001)	(E0000)
EG1	
(E0003)	(E0002)
EG2	
(E0005)	(E0004)
EG3	
(E0007)	(E0006)

(7) Returned value

Normal termination : 0 is returned.
Abnormal termination: Any value other than 0 is returned. (☞ Page 378, CHAPTER 7 ERROR CODES)

Point

- The maximum number of write points that can be specified for lSize (varSize) should be the value which satisfies the following condition.
  Write starting device number + Number of write points ≦ Last device number
- When specifying bit devices, a multiple of 16 can be specified as a device number.
- For iData (lpsData or varData), reserve a memory area for the number of points specified for lSize (varSize).
  If the memory area is not reserved, a critical error (an application error or the like) may occur.

5.2.20 ReadDeviceRandom2 (Reading devices randomly)

(1) Applicable controls

Applicable to the Act(ML)ProgType control and the Act(ML)UtlType control.

(2) Feature

Read devices in 2-byte data unit randomly.

(3) Format (Dispatch interface)

(a) VBA

lRet = object.ReadDeviceRandom2(szDeviceList, lSize, iData(0))

Long	lRet	Returned value	Output
String	szDeviceList	Device name	Input
Long	lSize	Number of read points	Input
Integer	iData(n)	Read device value	Output

(b) Visual C++® .NET (MFC)

lRet = object.ReadDeviceRandom2(szDeviceList, lSize, *lpsData)

Long	lRet	Returned value	Output
CString	szDeviceList	Device name	Input
Long	lSize	Number of read points	Input
Short	*lpsData	Read device value	Output

(c) VBScript

varRet = object.ReadDeviceRandom2(varDeviceList, varSize, lpvarData)

VARIANT	varRet	Returned value (LONG type)	Output
VARIANT	varDeviceList	Device name (character string type)	Input
VARIANT	varSize	Number of read points (LONG type)	Input
VARIANT	lpvarData	Read device value (SHORT type)	Output

(d) Visual Basic® .NET

lRet = object.ReadDeviceRandom2(szDeviceList, lSize, sData(0))

Integer	lRet	Returned value	Output
String	szDeviceList	Device name	Input
Integer	lSize	Number of read points	Input
short	sData(n)	Read device value	Output

(e) Visual C++® .NET

iRet = object.ReadDeviceRandom(*szDeviceList, iSize, *lpsData)

int	iRet	Returned value	Output
String	*szDeviceList	Device name	Input
int	iSize	Number of read points	Input
short	*lpsData	Read device value	Output

(4) Format (Custom interface)

(a) Visual C++® .NET (MFC)

hResult = object.ReadDeviceRandom2(szDeviceList, lSize, *lpsData, *lplRetCode)

HRESULT	hResult	Returned value of COM	Output
BSTR	szDeviceList	Device name	Input
LONG	lSize	Number of read points	Input
SHORT	*lpsData	Read device value	Output
LONG	*lplRetCode	Returned value of communication function	Output

(5) Description

- Data of a device group for the size of lSize (varSize) specified for szDeviceList (varDeviceList) are read.
- The read device values are stored in iData (lpsData or lpvarData).
- Use the line feed symbol to separate the character string specified for the device list. The line feed symbol is not necessary to be suffixed to the last device.
  Example:

When using Visual Basic®.NET ,VBA,VBScript:	"D0" & vbLf & "D1" & vbLf & "D2"
When using Visual C++®.NET:	D0\nD1\nD2

- For lData (lpsData or lpvarData), reserve arrays for more than the amount specified for lSize (varSize).

(6) How to specify devices

The following describes how to specify devices.

Example 1: When devices are specified as follows (Number of points: 3 points)

When using Visual Basic®.NET ,VBA,VBScript:	M0 & vbLf & D0 & vbLf & K8M0
When using Visual C++®.NET:	M0\nD0\nK8M0

2 Bytes
M0[*1]
D0
M0 to M15[*2]

Example 2: When devices including FXCPU devices of CN200 and later are specified (Total number of points: 3 points)[*3]

When using Visual Basic®.NET ,VBA,VBScript:	D0 & vbLf & CN200 & vbLf & D1
When usingVisual C++®.NET:	D0\nCN200\nD1

2 Bytes
D0
L of CN200 (Lower 2 bytes)[*3]
D1

*1 : The target device from which data is read is only one point of "M0", and "0" or "1" is stored as the device value.

*2 : Devices are stored from the lower bit in the order of device number.
Data are not read from the upper 2 bytes of M16 to M31.

*3 : For FXCPU devices of CN200 and later, the L (lower 2 bytes) of the specified devices is read for each point when the ReadDeviceRandom2 function is executed.
The H (upper 2 bytes) of the specified devices is not read.

Example 3: When devices including FD are specified (Total number of points: 3 points)

When using Visual Basic®.NET ,VBA,VBScript:	D0 & vbLf & FD0 & vbLf & D1
When using Visual C++®.NET:	D0\nFD0\nD1

2 Bytes
D0
LL of FD0 (Lower 2 bytes)
D1

Example 4: When 8-bit devices including EG are specified (Total number of points: 3 points)

The following is the example when 8-bit devices (E0000 and E0001 of SHARP programmable controller) are assigned to EG0.

When using Visual Basic®.NET ,VBA,VBScript:	D0 & vbLf & EG0 & vbLf & D1
When using Visual C++®.NET:	D0\nEG0\nD1

<table>
<tr><th colspan="2">2 Bytes</th></tr>
<tr><td colspan="2">D0</td></tr>
<tr><td colspan="2">EG0</td></tr>
<tr><td>(E0001)</td><td>(E0000)</td></tr>
<tr><td colspan="2">D1</td></tr>
</table>

(7) Returned value

Normal termination : 0 is returned.

Abnormal termination: Any value other than 0 is returned. (☞ Page 378, CHAPTER 7 ERROR CODES)

Point

- The maximum number of read points that can be specified for lSize (varSize) is 0x7FFFFFFF points.
- For iData (lpsData or lpvarData), reserve a memory area for the number of points specified for lSize (varSize). If the memory area is not reserved, a critical error (an application error or the like) may occur.
- When a double word device is specified, only the data of the lower 1 word (2 bytes) are stored using the ReadDeviceRandom2 function. (An error does not occur.)
When reading data from a double word device, use the ReadDeviceRandom function or the GetDevice function.

5.2.21 WriteDeviceRandom2 (Writing devices randomly)

(1) Applicable controls

Applicable to the Act(ML)ProgType control and the Act(ML)UtlType control.

(2) Feature

Write devices in 2-byte data unit randomly.

(3) Format (Dispatch interface)

(a) VBA

lRet = object.WriteDeviceRandom2(szDeviceList, lSize, iData(0))

Long	lRet	Returned value	Output
String	szDeviceList	Device name	Input
Long	lSize	Number of write points	Input
Integer	iData(n)	Device value to be written	Input

(b) Visual C++® .NET (MFC)

lRet = object.WriteDeviceRandom2(szDeviceList, lSize, *lpsData)

Long	lRet	Returned value	Output
CString	szDeviceList	Device name	Input
Long	lSize	Number of write points	Input
Short	*lpsData	Device value to be written	Input

(c) VBScript

varRet = object.WriteDeviceRandom2(varDeviceList, varSize, varData)

VARIANT	varRet	Returned value (LONG type)	Output
VARIANT	varDeviceList	Device name (character string type)	Input
VARIANT	varSize	Number of write points (LONG type)	Input
VARIANT	varData	Device value to be written (SHORT type)	Input

(d) Visual Basic® .NET

IRet = object.WriteDeviceRandom2(szDeviceList, ISize, sData(0))

Integer	IRet	Returned value	Output
String	szDeviceList	Device name	Input
Integer	ISize	Number of write points	Input
short	sData(n)	Device value to be written	Input

(e) Visual C++® .NET

iRet = object.WriteDeviceRandom2(*szDeviceList, iSize, *ipsData)

int	iRet	Returned value	Output
String	*szDeviceList	Device name	Input
int	iSize	Number of write points	Input
short	*ipsData	Device value to be written	Input

(4) Format (Custom interface)

(a) Visual C++® .NET (MFC)

hResult = object.WriteDeviceRandom2(szDeviceList, lSize, *lpsData, *lplRetCode)

HRESULT	hResult	Returned value of COM	Output
BSTR	szDeviceList	Device name	Input
LONG	lSize	Number of write points	Input
SHORT	*lpsData	Device value to be written	Input
LONG	*lplRetCode	Returned value of communication function	Output

(5) Description

- Data of a device group for the size of lSize (varSize) specified for szDeviceList (varDeviceList) are written.
- The device values to be written are stored in iData (lpsData or varData).
- Use the line feed symbol to separate the character string specified for the device list. The line feed symbol is not necessary to be suffixed to the last device.
  Example:

When using Visual Basic®.NET ,VBA,VBScript:	"D0" & vbLf & "D1" & vbLf & "D2"
When using Visual C++®.NET:	D0\nD1\nD2

- For lData (lpsData or varData), reserve arrays for more than the amount specified for lSize (varSize).

(6) How to specify devices

The following describes how to specify devices.

Example 1: When devices are specified as follows (Number of points: 3 points)

When using Visual Basic®.NET ,VBA,VBScript:	M0 & vbLf & D0 & vbLf & K8M0
When using Visual C++®.NET:	M0\nD0\nK8M0

2 Bytes
M0*1
D0
M0 to M15*2

Example 2: When devices including FXCPU devices of CN200 and later are specified (Total number of points: 3 points)*3

When using Visual Basic®.NET ,VBA,VBScript:	D0 & vbLf & CN200 & vbLf & D1
When using Visual C++®.NET:	D0\nCN200\nD1

2 Bytes
D0
L of CN200 (Lower 2 bytes)*3
D1

*1 : The target device to which data is written is only one point of "M0", and the least significant bit of the set 2-byte data is written as the device value.

*2 : Devices are stored from the lower bit in the order of device number. "0" is written to the upper 2 bytes of M16 to M31.

*3 : For FXCPU devices of CN200 and later, data are written to the L (lower 2 bytes) of the specified devices for each point when the WriteDeviceRandom2 function is executed. "0" is written to the H (upper 2 bytes) of the specified devices.

Example 3: When devices including FD are specified (Total number of points: 3 points)

When using Visual Basic®.NET ,VBA,VBScript:	D0 & vbLf & FD0 & vbLf & D1
When using Visual C++®.NET:	D0\nFD0\nD1

2 Bytes
D0
LL of FD0 (Lower 2 bytes)
D1

Example 4: When 8-bit devices including EG are specified (Total number of points: 3 points)

The following is the example when 8-bit devices (E0000 and E0001 of SHARP programmable controller) are assigned to EG0.

When using Visual Basic®.NET ,VBA,VBScript:	D0 & vbLf & EG0 & vbLf & D1
When using Visual C++®.NET:	D0\nEG0\nD1

2 Bytes	
D0	
EG0	
(E0001)	(E0000)
D1	

(7) Returned value

Normal termination : 0 is returned.

Abnormal termination: Any value other than 0 is returned. (☞ Page 378, CHAPTER 7 ERROR CODES)

Point

- The maximum number of write points that can be specified for lSize (varSize) is 0x7FFFFFFF points.
- For iData (lpsData or varData), reserve a memory area for the number of points specified for lSize (varSize). If the memory area is not reserved, a critical error (an application error or the like) may occur.
- When a double word device is specified, data are written to the data area of the lower 1 word (2 bytes) using the WriteDeviceRandom2 function, and "0" is written to the data area of the upper 1 word (2 bytes). When writing a double word device, use the WriteDeviceRandom function or the SetDevice function.
- If a Q motion CPU is accessed, an error is returned.

5.2.22 SetDevice2 (Setting device data)

(1) Applicable controls

Applicable to the Act(ML)ProgType control and the Act(ML)UtlType control.

(2) Feature

Set one point of device in 2-byte data unit.

(3) Format (Dispatch interface)

(a) VBA

lRet = object.SetDevice2(szDevice, iData)

Long	lRet	Returned value	Output
String	szDevice	Device name	Input
Integer	iData	Set data	Input

(b) Visual C++® .NET (MFC)

lRet = object.SetDevice2(szDevice, sData)

Long	lRet	Returned value	Output
CString	szDevice	Device name	Input
Short	sData	Set data	Input

(c) VBScript

varRet = object.SetDevice2(varDevice, varData)

VARIANT	varRet	Returned value (LONG type)	Output
VARIANT	varDevice	Device name (character string type)	Input
VARIANT	varData	Set data (SHORT type)	Input

(d) Visual Basic® .NET

lRet = object.SetDevice2(szDevice, sData)

Integer	lRet	Returned value	Output
String	szDevice	Device name	Input
short	sData	Set data	Input

(e) Visual C++® .NET

iRet = object.SetDevice2(*szDevice, sData)

int	iRet	Returned value	Output
String	*szDevice	Device name	Input
short	sData	Set data	Input

(4) Format (Custom interface)

(a) Visual C++® .NET (MFC)

hResult = object.SetDevice2(szDevice, sData, *lplRetCode)

HRESULT	hResult	Returned value of COM	Output
BSTR	szDevice	Device name	Input
SHORT	sData	Set data	Input
LONG	*lplRetCode	Returned value of communication function	Output

(5) Description

- One point of device specified for szDevice (varDevice) is specified using lData (varData).
- When specifying bit devices, the least significant bit of the iData value (sData value or varData value) is valid.

(6) How to specify devices

The following describes how to specify devices.

<When bit device is specified>

Example: M0

2 Bytes
M0*1

<When word device is specified>

Example: D0

2 Bytes
D0

<When double-word device is specified>

Example: K8M0

2 Bytes
M0 to M15*2

<When CN200 or later of FXCPU is specified>

Example: CN200

2 Bytes
L of CN200 (Lower 2 bytes)*3

<When gateway device is specified>

Example: When 8-bit devices (E0000 and E0001 of SHARP programmable controller) are assigned to EG0

2 Bytes	
EG0	
(E0001)	(E0000)

*1 : The target device to which data is written is only one point of "M0", and the least significant bit of the set 2-byte data is written as the device value.

*2 : Devices are stored from the lower bit in the order of device number.
"0" is written to the upper 2 bytes of M16 to M31.

*3 : For FXCPU devices of CN200 and later, data are written to the L (lower 2 bytes) of the specified devices for each point when the SetDevice2 function is executed.
"0" is written to the H (upper 2 bytes) of the specified devices.

(7) Returned value

Normal termination : 0 is returned.

Abnormal termination: Any value other than 0 is returned. (☞ Page 378, CHAPTER 7 ERROR CODES)

Point

- When a double word device is specified, data are written to the data area of the lower 1 word (2 bytes) using the SetDevice2 function, and "0" is written to the data area of the upper 1 word (2 bytes).
  When writing a double word device, use the WriteDeviceRandom function or the SetDevice function.

5.2.23 GetDevice2 (Acquiring device data)

(1) Applicable controls

Applicable to the Act(ML)ProgType control and the Act(ML)UtlType control.

(2) Feature

Acquire one point of device in 2-byte data unit.

(3) Format (Dispatch interface)

(a) VBA

lRet = object.GetDevice2(szDevice, iData)

Long	lRet	Returned value	Output
String	szDevice	Device name	Input
Integer	iData	Acquired data	Output

(b) Visual C++® .NET (MFC)

lRet = object.GetDevice2(szDevice, lSize, *lpsData)

Long	lRet	Returned value	Output
CString	szDevice	Device name	Input
Short	*lpsData	Acquired data	Output

(c) VBScript

varRet = object.GetDevice2(varDevice, lpvarData)

VARIANT	varRet	Returned value (LONG type)	Output
VARIANT	varDevice	Device name (character string type)	Input
VARIANT	lpvarData	Acquired data (SHORT type)	Output

(d) Visual Basic® .NET

IRet = object.GetDevice2(szDevice, sData)

Integer	IRet	Returned value	Output
String	szDevice	Device name	Input
short	sData	Acquired data	Output

(e) Visual C++® .NET

iRet = object.GetDevice(*szDevice, *ipsData)

int	iRet	Returned value	Output
String	*szDevice	Device name	Input
short	*spsData	Acquired data	Output

(4) Format (Custom interface)

(a) Visual C++® .NET (MFC)

hResult = object.GetDevice2(szDevice, *lpsData, *lplRetCode)

HRESULT	hResult	Returned value of COM	Output
BSTR	szDevice	Device name	Input
SHORT	*lpsData	Acquired data	Output
LONG	*lplRetCode	Returned value of communication function	Output

(5) Description

One point of device data specified for szDevice (varDevice) is stored in iData (lpsData or lpvarData).

(6) How to specify devices

The following describes how to specify devices.

<When bit device is specified>

Example: M0

2 Bytes
M0[*1]

<When word device is specified>

Example: D0

2 Bytes
D0

<When double-word device is specified>

Example: K8M0

2 Bytes
M0 to M15[*2]

<When CN200 or later of FXCPU is specified>

Example: CN200

2 Bytes
L of CN200[*3]

<When gateway device is specified>

Example: When 8-bit devices (E0000 and E0001 of SHARP programmable controller) are assigned to EG0

2 Bytes	
EG0	
(E0001)	(E0000)

*1 : The target device from which data is read is only one point of "M0", and "0" or "1" is stored as the device value.

*2 : Devices are stored from the lower bit in the order of device number.
Data are not read from the upper 2 bytes of M16 to M31.

*3 : For FXCPU devices of CN200 and later, the L (lower 2 bytes) of the specified devices is read for each point when the GetDevice2 function is executed.
The H (upper 2 bytes) of the specified devices is not read.

(7) Returned value

Normal termination : 0 is returned.

Abnormal termination: Any value other than 0 is returned. (☞ Page 378, CHAPTER 7 ERROR CODES)

Point

- When a double word device is specified, only the data of the lower 1 word (2 bytes) are stored using the GetDevice2 function. (An error does not occur.)
- When reading data from a double word device, use the ReadDeviceRandom function or the GetDevice function.

5.2.24 Connect (Connecting telephone line)

(1) Applicable controls

Applicable to the ActProgType control and the ActUtlType control.
Applicable to the modem communication only.

(2) Feature

Connect the telephone line.

(3) Format (Dispatch interface)

(a) Visual C++® .NET (MFC) , VBA

IRet = object.Connect()

Long	IRet	Returned value	Output

(b) Visual Basic® .NET

IRet = object.Connect()

Integer	IRet	Returned value	Output

(c) Visual C++® .NET

iRet = object.Connect()

int	iRet	Returned value	Output

(4) Format (Custom interface)

(a) Visual C++® .NET (MFC)

hResult = object.Connect(*lplRetCode)

HRESULT	hResult	Returned value of COM	Output
LONG	*lplRetCode	Returned value of communication function	Output

(5) Description

- The telephone line is connected according to the property settings of the modem communication control.
- When routing a serial communication module, the telephone line is connected in the connection system set in the ActConnectWay property.
  When auto line connect (callback number specification), callback connect (number specification), or callback request (number specification) is set in the ActConnectWay property, an error occurs if a number is not set in the ActCallbackNumber property.

(6) Returned value

Normal termination : 0 is returned.
Abnormal termination: Any value other than 0 is returned. (☞ Page 378, CHAPTER 7 ERROR CODES)

(7) Considerations for executing the Connect function

- Always connect the telephone line before the execution of the Open function.
- When disconnecting the telephone line, execute the Disconnect function.
  During the execution of the Connect function, the telephone line remains connected even when the Open and Close functions are executed repeatedly.
- If the telephone line is disconnected in the Open status for some reason, always execute the Close function before reconnecting the telephone line.

(8) Considerations when using multiple telephone line connection objects

- When control types, port numbers, and telephone numbers of controls are different
  When control types, port numbers, and telephone numbers set to multiple controls are different, an error (error code: 0xF1000016) occurs if the Connect function is executed to connect the control whose port number and telephone number are different from those of the control which executed the Connect function first.
- When port numbers and telephone numbers of controls are the same
  When control types, port numbers, and telephone numbers set to multiple controls are the same, the termination status are different according to the connection system of the callback function.
  The following table shows the relations between the connection system and the termination status of the callback function.

Connection system of control which executed the Connect function first	Connection system of control which executed the Connect function second or Later			
	Auto line connect Auto line connect (Callback fixation) Auto line connect (Callback number specification)	Callback connect (Fixation) Callback connect (Number specification)	Callback request (Fixation) Callback request (Number specification)	Callback reception waiting
Auto line connect Auto line connect (Callback fixation) Auto line connect (Callback number specification)	○	○	×	×
Callback connect (Fixation) Callback connect (Number specification)	○	○	×	×
Callback request (Fixation) Callback request (Number specification)	○	○	○	○
Callback reception waiting	×	×	×	○

○: Normal termination ×: Abnormal termination (Error occurrence)

5.2.25 Disconnect (Disconnecting telephone line)

(1) Applicable controls

Applicable to the ActProgType control and the ActUtlType control.

Applicable to the modem communication only.

(2) Feature

Disconnect the telephone line.

(3) Format (Dispatch interface)

(a) Visual C++® .NET (MFC)

lRet = object.Disconnect()

Long	lRet	Returned value	Output

(b) Visual Basic® .NET

lRet = object.Disconnect()

Integer	lRet	Returned value	Output

(c) Visual C++® .NET

iRet = object.Disconnect()

int	iRet	Returned value	Output

(4) Format (Custom interface)

(a) Visual C++® .NET (MFC)

hResult = object.Disconnect(*lplRetCode)

HRESULT	hResult	Returned value of COM	Output
LONG	*lplRetCode	Returned value of communication function	Output

(5) Description

The telephone line that was connected using the Connect function is disconnected.

(6) Returned value

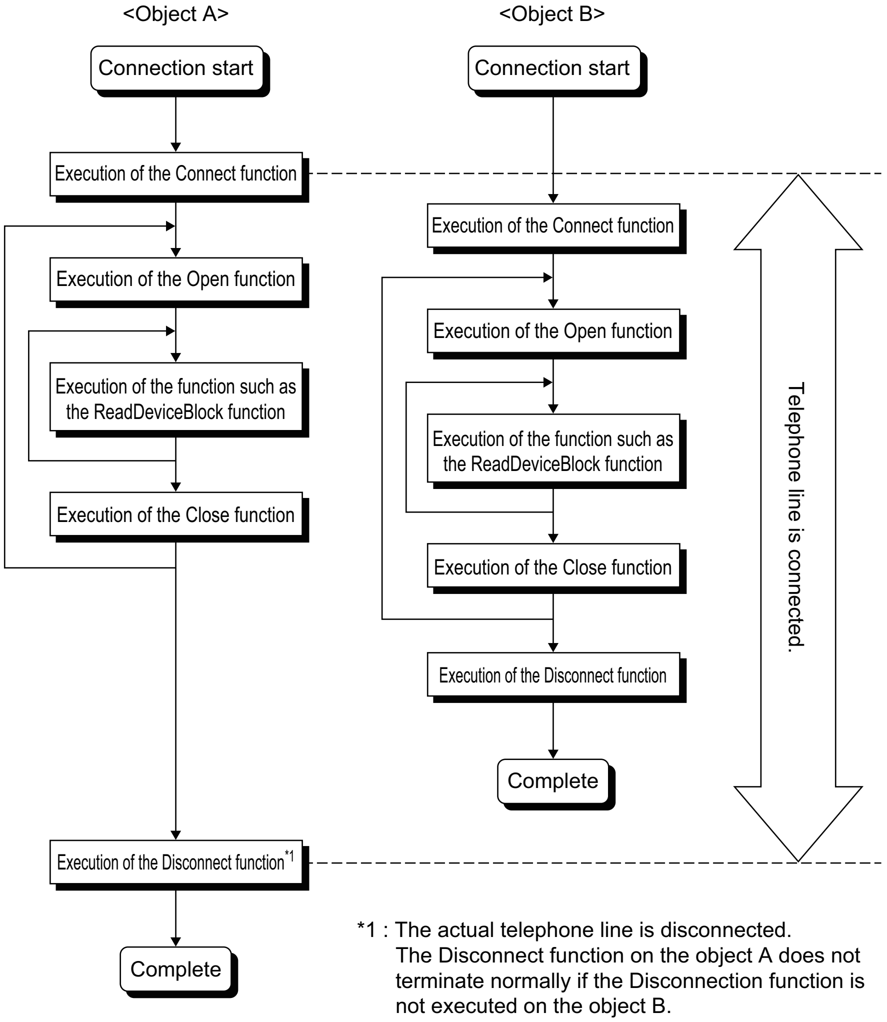
Normal termination : 0 is returned.

Abnormal termination: Any value other than 0 is returned. (☞ Page 378, CHAPTER 7 ERROR CODES)

(7) Considerations for executing the Disconnect function

- When executing the Disconnect function during the execution of the Open function, execute the Close function before executing the Disconnect function.
- If the telephone line is disconnected in the Open status for some reason, always execute the Close function before reconnecting the telephone line.
- When multiple telephone line connection objects are used, execute the Disconnect function on the object which executed the Connect function first after executing the Disconnect function on other objects.
  When multiple telephone line connection objects are used, the telephone line is not disconnected if the Disconnect function is not executed on the object that executed the Connect function first.
  The following figure shows the example of how to use multiple objects simultaneously.

Example: When using two controls simultaneously
(Including the case when the objects A and B are operated by different applications)



<Object A>
<Object B>
Connection start
Connection start
Execution of the Connect function
Execution of the Connect function
Execution of the Open function
Execution of the Open function
Execution of the function such as the ReadDeviceBlock function
Execution of the function such as the ReadDeviceBlock function
Execution of the Close function
Execution of the Close function
Execution of the Disconnect function
Complete
Telephone line is connected.
Execution of the Disconnect function*1
Complete

*1 : The actual telephone line is disconnected.
The Disconnect function on the object A does not terminate normally if the Disconnection function is not executed on the object B.

5.2.26 GetErrorMessage (Receiving error message)

(1) Applicable control

Applicable to the Act(ML)SupportMsg control.

(2) Feature

Receive the error description and its corrective action corresponding to the error code.

(3) Format (Dispatch interface)

(a) VBA

lRet = object.GetErrorMessage(lErrorCode, szErrorMessage)

Long	lRet	Returned value	Output
String	lErrorCode	Error code	Input
String	szErrorMessage	Error message	Output

(b) Visual C++® .NET (MFC)

lRet = object.GetErrorMessage(lErrorCode, *lpszErrorMessage)

Long	lRet	Returned value	Output
Long	lErrorCode	Error code	Input
BSTR	*lpszErrorMessage	Error message	Output

(c) VBScript

varRet = object.GetErrorMessage(varErrorCode, lpvarErrorMessage)

VARIANT	varRet	Returned value (LONG type)	Output
VARIANT	varErrorCode	Error code (LONG type)	Input
VARIANT	lpvarErrorMessage	Error message (character string type)	Output

(d) Visual Basic® .NET

lRet = object.GetErrorMessage(lErrorCode, szErrorMessage)

Integer	lRet	Returned value	Output
Integer	lErrorCode	Error code	Input
String	szErrorMessage	Error message	Output

(e) Visual C++® .NET

iRet = object.GetErrorMessage(iErrorCode, **lpsErrorMessage)

int	iRet	Returned value	Output
int	iErrorCode	Error code	Input
String	**lpsErrorMessage	Error message	Output

(4) Format (Custom interface)

(a) Visual C++® .NET (MFC)

hResult = object.GetErrorMessage(lErrorCode, *lpszErrorMessage, *lplRetCode)

HRESULT	hResult	Returned value of COM	Output
LONG	lErrorCode	Error code	Input
BSTR	*lpszErrorMessage	Error message	Output
LONG	*lplRetCode	Returned value of communication function	Output

(5) Description

- The error description and its corrective action of the error code specified for lErrorCode (varErrorCode) are read.
- The read error description and its corrective action are stored in szErrorMessage (lpszErrorMessage or lpvarErrorMessage).

(6) Returned value

Normal termination : 0 is returned.
Abnormal termination: Any value other than 0 is returned. (☞ Page 378, CHAPTER 7 ERROR CODES)

5.3 Details of Functions (For .NET Control)

5.3.1 Open (Opening communication line)

(1) Applicable control

Applicable to the DotUtlType control.

(2) Feature

Open the communication line.

(3) Format

(a) Visual Basic® .NET

IRet = object.Open()

Integer	IRet	Returned value	Output

(b) Visual C++® .NET

iRet = object.Open()

int	iRet	Returned value	Output

(4) Description

Lines are connected according to the set value of the Open function property.

(5) Returned value

Normal termination : 0 is returned.

Abnormal termination: Any value other than 0 is returned. (☞ Page 378, CHAPTER 7 ERROR CODES)

Point

- When modem communication is used, the Open function cannot be executed without the execution of the Connect function.
- If the Open function property is changed after the completion of the Open function, the setting of communication target is not changed. To change the communication setting, close the communication line, set the setting of communication target, and open the communication line again.
- Even when a CPU type which is different from the CPU used for the communication is set to the ActCpuType property, the Open function may be completed normally.
  In such a case, the connection range, usable method, or device range may be narrowed.
  When executing the Open function, set the correct CPU type to the ActCpuType property.

5.3.2 Close (Closing communication line)

(1) Applicable control

Applicable to the DotUtlType control.

(2) Feature

Close the communication line.

(3) Format

(a) Visual Basic® .NET

IRet = object.Close()

Integer	IRet	Returned value	Output

(b) Visual C++® .NET

iRet = object.Close()

int	iRet	Returned value	Output

(4) Description

The line connected using the Open function is closed.

(5) Returned value

Normal termination : 0 is returned.

Abnormal termination: Any value other than 0 is returned. (☞ Page 378, CHAPTER 7 ERROR CODES)

5.3.3 ReadDeviceBlock (Reading devices in bulk)

(1) Applicable control

Applicable to the DotUtlType control.

(2) Feature

Read devices in bulk.

(3) Format

(a) Visual Basic® .NET

lRet = object.ReadDeviceBlock(szLabel, iSize, iData(0))

Integer	lRet	Returned value	Output
String	szLabel	Label name	Input
Integer	iSize	Number of read points	Input
Integer	iData(n)	Read device values	Output

(b) Visual C++® .NET

iRet = object.ReadDeviceBlock(*szLabel, iSize, *ipiData)

Int	iRet	Returned value	Output
String*	szLabel	Label name	Input
int	iSize	Number of read points	Input
int*	ipiData	Read device values	Output

(4) Description

- The devices for the amount specified for iSize (number of read points) are read in bulk starting from the device specified for szLabel (label name).
- The read device values are stored in iData (ipiData).
- For iData (ipiData), reserve arrays for more than the amount specified for iSize.

(5) How to specify devices

The following describes how to specify label names and device values to be read.

- The following data type can be specified for label name.

Type class		Label data type	Label name format
Array			Label name
	Member	Bit, Word, Double Word, Float (Single Precision), Float (Double Precision), String[*1], Time, Timer, Counter, Retentive Timer	Label name [number of elements] Label name [n1] [n2] [n3]

[*1] : The maximum number of characters that can be specified is 32 + NULL

- Set the following values for the number of read points according to the label data type.

Label (array) data type	Number of applicable words	Number of read points to be specified
Bit	1	Number of label array elements divided by 16 (rounded up)
Word	1	Number of label array elements
Double Word, Float (Single Precision)	2	Number of label array elements multiplied by 2
Float (Double Precision)	4	Number of label array elements multiplied by 4
String	17	Number of label array elements multiplied by 17
Time	2	Number of label array elements multiplied by 2
Timer, Counter, Retentive Timer	1	Number of label array elements

- The read device values are stored as follows.

<When bit device is specified>

Example: Read 3 points of data (3 words = 48 bits) from the devices starting from M0.

- Label setting (Data type: Bit, Number of array elements: 48)

Data type	Device
Bit (0..47)	M0

- Number of read points: 3
- Read device values

Upper 2 bytes	Lower 2 bytes
Not used (0 is stored.)	M0 to M15[*2]
	M16 to M31[*2]
	M32 to M47[*2]

[*2] : Devices are stored from the lower bit in the order of device number.

<When word device is specified>

Example: Read 3 points of data from the devices starting from D0.

- Label setting (Data type: Word, Number of array elements: 3)

Data type	Device
Word (0..2)	D0

- Number of read points: 3
- Read device values

Upper 2 bytes	Lower 2 bytes
Not used (0 is stored.)	D0
	D1
	D2

<When FXCPU devices of CN200 and later are specified>

Example: Read 6 points of data from the devices starting from CN200.[*3]

- Label setting (Data type: Double Word, Number of array elements: 3)

Data type	Device
Double Word (0..2)	CN200

- Number of read points: 6
- Read device values

Upper 2 bytes	Lower 2 bytes
Not used (0 is stored.)	L of CN200
	H of CN200
	L of CN201
	H of CN201
	L of CN202
	H of CN202

[*3] : For FXCPU devices of CN200 and later, 2 points of data are read from each 1 point of device (upper (H) and lower (L) data).
Reading only 1 point of data will result in an error.

<When FD device is specified (4-word device)>

Example: Read 8 points of data from the devices starting from FD0.

- Label setting (Data type: Double Word, Number of array elements: 2)

Data type	Device
Double Word (0..1)	FD0

- Number of read points: 8
- Read device values

Upper 2 bytes	Lower 2 bytes
Not used (0 is stored.)	LL of FD0
	LH of FD0
	HL of FD0
	HH of FD0
	LL of FD1
	LH of FD1
	HL of FD1
	HH of FD1

<When label is Double Word array and word device is specified>

Example: Read 6 points of data from the devices starting from D100.

- Label setting (Data type: Double Word, Number of array elements: 3)

Data type	Device
Double Word (0..2)	D100

- Number of read points: 6
- Read device values

Upper 2 bytes	Lower 2 bytes
Not used (0 is stored.)	D100
	D101
	D102
	D103
	D104
	D105

(6) Returned value

Normal termination : 0 is returned.

Abnormal termination: Any value other than 0 is returned. (☞ Page 378, CHAPTER 7 ERROR CODES)

Point

- The maximum number of read points should be the value which satisfies the following condition.
  Read start device number + Number of read points ≦ Last device number
- When specifying bit type array labels, 0 or a multiple of 16 can be specified as a device number.
- For the number of read points, specify the number of words which applies to the data type specified for the label name.
  For the read device values, reserve a memory area for the number of points specified for the number of read points. If the memory area is not reserved, a critical error (an application error or the like) may occur.
- When a device which corresponds to the label name does not exist, an error occurs and data cannot be read.
  When any one of devices which correspond to multiple label names does not exist, an error occurs and data cannot be read.
- Digit specified bit device and index setting cannot be used.

5.3.4 WriteDeviceBlock (Writing devices in bulk)

(1) Applicable control

Applicable to the DotUtlType control.

(2) Feature

Write devices in bulk.

(3) Format

(a) Visual Basic® .NET

Ret = object.WriteDeviceBlock(szLabel, iSize, iData(0))

Integer	IRet	Returned value	Output
String	szLabel	Label name	Input
Integer	iSize	Number of write points	Input
Integer	iData(n)	Device values to be written	Input

(b) Visual C++® .NET

iRet = object.WriteDeviceBlock(*szLabel, iSize, *ipiData)

int	iRet	Returned value	Output
String*	szLabel	Label name	Input
int	iSize	Number of write points	Input
int*	ipiData	Device values to be written	Input

(4) Description

- The devices for the amount specified for iSize (number of write points) are written in bulk starting from the device specified for szLabel (label name).
- The device values to be written are stored in iData (ipiData).
- For iData (ipiData), reserve arrays for more than the amount specified for iSize.

(5) How to specify devices

The following describes how to specify label names and device values to be written.

- The following data type can be specified for label name.

Type class		Label data type	Label name format
Array			Label name
	Member	Bit, Word, Double Word, Float (Single Precision), Float (Double Precision), String[*1], Time, Timer, Counter, Retentive Timer	Label name [number of elements] Label name [n1] [n2] [n3]

*1 : The maximum number of characters that can be specified is 32 + NULL

- Set the following values for the number of write points according to the label data type.

Label (array) data type	Number of applicable words	Number of read points to be specified
Bit	1	Number of label array elements divided by 16 (rounded up)
Word	1	Number of label array elements
Double Word, Float (Single Precision)	2	Number of label array elements multiplied by 2
Float (Double Precision)	4	Number of label array elements multiplied by 4
String	17	Number of label array elements multiplied by 17
Time	2	Number of label array elements multiplied by 2
Timer, Counter, Retentive Timer	1	Number of label array elements

- Set the device values to be written as follows.

<When bit device is specified>

Example: Write 3 points of data (3 words = 48 bits) to the devices starting from M0.

- Label setting (Data type: Bit, Number of array elements: 48)

Data type	Device
Bit (0..47)	M0

- Number of write points: 3
- Device values to be written

Upper 2 bytes	Lower 2 bytes
Not used	M0 to M15[*2]
	M16 to M31[*2]
	M32 to M47[*2]

*2 : Devices are stored from the lower bit in the order of device number.

<When FXCPU devices of CN200 and later are specified>

Example: Write 6 points of data to the devices starting from CN200.[*3]

- Label setting (Data type: Double Word, Number of array elements: 3)

Data type	Device
Double Word (0..2)	CN200

- Number of write points: 6
- Device values to be written

Upper 2 bytes	Lower 2 bytes
Not used	L of CN200
	H of CN200
	L of CN201
	H of CN201
	L of CN202
	H of CN202

*3 : For FXCPU devices of CN200 and later, 2 points of data are written to each 1 point of device (upper (H) and lower (L) data).
Writing only 1 point of data will result in an error.

<When word device is specified>

Example: Write 3 points of data to the devices starting from D0.

- Label setting (Data type: Word, Number of array elements: 3)

Data type	Device
Word (0..2)	D0

- Number of write points: 3
- Device values to be written

Upper 2 bytes	Lower 2 bytes
Not used	D0
	D1
	D2

<When FD device is specified (4-word device)>

Example: Write 8 points of data to the devices starting from FD0.

- Label setting (Data type: Double Word, Number of array elements: 2)

Data type	Device
Double Word (0..1)	FD0

- Number of write points: 8
- Device values to be written

Upper 2 bytes	Lower 2 bytes
Not used	LL of FD0
	LH of FD0
	HL of FD0
	HH of FD0
	LL of FD1
	LH of FD1
	HL of FD1
	HH of FD1

<When label is Double Word array and word device is specified>

Example: Write 6 points of data to the devices starting from D100.

- Label setting (Data type: Double Word, Number of array elements: 3)

Data type	Device
Double Word (0..2)	D100

- Number of write points: 6
- Device values to be written

Upper 2 bytes	Lower 2 bytes
Not used	D100
	D101
	D102
	D103
	D104
	D105

(6) Returned value

Normal termination : 0 is returned.

Abnormal termination: Any value other than 0 is returned. (☞ Page 378, CHAPTER 7 ERROR CODES)

Point

- ● The maximum number of write points should be the value which satisfies the following condition.
  Write starting device number + Number of write points ≦ Last device number
- ● When specifying bit type array labels, 0 or a multiple of 16 can be specified as a device number.
- ● For the number of write points, specify the number of words which applies to the data type specified for the label name. For the device values to be written, reserve a memory area for the number of points specified for the number of write points. If the memory area is not reserved, a critical error (an application error or the like) may occur.
- ● When a device which corresponds to the label name does not exist, an error occurs and data cannot be written.
  When any one of devices which correspond to multiple label names does not exist, an error occurs and data cannot be written.
- ● Digit specified bit device and index setting cannot be used.

5.3.5 ReadDeviceRandom (Reading devices randomly)

(1) Applicable control

Applicable to the DotUtlType control.

(2) Feature

Read devices randomly.

(3) Format

(a) Visual Basic® .NET

IRet = object.ReadDeviceRandom(szLabel, iSize, iData(0))

Integer	IRet	Returned value	Output
String	szLabel	Label name	Input
Integer	iSize	Number of read points	Input
Integer	iData(n)	Read device values	Output

IRet = object.ReadDeviceRandom(szLabelList, iSize, iData(0))

Integer	IRet	Returned value	Output
String	szLabelList(n)	Label list	Input
Integer	iSize	Number of read points	Input
Integer	iData(n)	Read device values	Output

(b) Visual C++® .NET

iRet = object.ReadDeviceRandom(*szLabel, iSize, *ipiData)

Int	iRet	Returned value	Output
String*	szLabel	Label name	Input
int	iSize	Number of read points	Input
int*	ipiData	Read device values	Output

iRet = object.ReadDeviceRandom(**szLabelList, iSize, *arriData)

Int	iRet	Returned value	Output
String**	szLabelList	Label list	Input
int	iSize	Number of read points	Input
int*	ipiData	Read device values	Output

(4) Description

- Data of a device group for the size of iSize specified for the label name szLabel (szLabelList) are read.
- The read device values are stored in iData (ipiData).
- For iData (ipiData), reserve arrays for more than the amount specified for iSize.

(5) How to specify devices

The following describes how to specify label names and device values to be read.

- The following data type can be specified for label name.

Type class		Label data type	Label name format
Basic type		Bit, Word, Double Word, Float (Single Precision), Float (Double Precision), String*1, Time, Timer, Counter, Retentive Timer	Label name
Array			Label name
	Member	(The label data type can be specified in the same manner as the basic type.)	Label name [number of elements] Label name [n1] [n2] [n3]
Structure			Label name
	Member	(The label data type can be specified in the same manner as the basic type.)	Label name.Member name
Structured array			Label name [number of elements]
	Member	(The label data type can be specified in the same manner as the basic type.)	Label name [number of elements].Member name Label name [n1] [n2] [n3].Element

*1 : The maximum number of characters that can be specified is 32 + NULL.

- For the number of read points, specify the sum of the following values correspond to the elements according to the label data type.

Label data type	Number of applicable words	Number of read points to be specified
Bit, Word	1	Number of label elements
Double Word, Float (Single Precision)	2	Number of label elements multiplied by 2
Float (Double Precision)	4	Number of label elements multiplied by 4
String	17	Number of label elements multiplied by 17
Time	2	Number of label elements multiplied by 2
Timer, Counter, Retentive Timer	1	Number of label elements

- The read device values are stored as follows.

<When bit device and word device are specified>

Example: Read data from each 1 point of M0 and D0.

- Label setting

szLabelList		Data type	Device
[0]	LABEL1	Bit	M0
[1]	LABEL2	Word	D0

- Number of read points: 2
- Read device values

Upper 2 bytes	Lower 2 bytes	Applicable label
Not used (0 is stored.)	M0[*2]	LABEL1
	D0	LABEL2

[*2] : The device to be read is 1 point of "M0", and "0" or "1" is stored for the device value.

<When FXCPU devices of CN200 and later are specified>

Example: Read 3 points of data from the devices including CN200.[*3]

- Label setting

szLabelList		Data type	Device
[0]	LABEL1	Word	D0
[1]	LABEL2	Double Word	CN200
[2]	LABEL3	Word	D1

- Number of read points: 3
- Read device values

Upper 2 bytes	Lower 2 bytes	Applicable label
Not used (0 is stored.)	D0	LABEL1
H of CN200	L of CN200	LABEL2
Not used (0 is stored.)	D1	LABEL3

[*3] : For FXCPU devices of CN200 and later, 4 bytes are read as 1 read point.

<When FD device is specified (4-word device)>

Example: Read 3 points of data from the devices including FD0.

- Label setting

szLabelList		Data type	Device
[0]	LABEL1	Word	D0
[1]	LABEL2	Double Word	FD0
[2]	LABEL3	Word	D1

- Number of read points: 3
- Read device values

Upper 2 bytes	Lower 2 bytes	Applicable label
Not used (0 is stored.)	D0	LABEL1
	LL of FD0[*4]	LABEL2
	D1	LABEL3

[*4] : Only lower 2 bytes are read. Data are not read from the specified devices HH, HL, and LH (upper 6 bytes).

<When data type equivalent to 2 words or more is specified for label>

Example: Read data by specifying labels of Double Word, Float (Single Precision), Float (Double Precision), String[*5], and Time types.

- Label setting

szLabelList		Data type	Device
[0]	LABEL1	Double Word	D0
[1]	LABEL2	Float (Single Precision)	D100
[2]	LABEL3	Float (Double Precision)	D200
[3]	LABEL4	String	D300
[4]	LABEL5	Time	D400

- Number of read points: 27
- Read device values

Upper 2 bytes	Lower 2 bytes	Applicable label
Not used (0 is stored.)	D0	LABEL1
	D1	
	D100	LABEL2
	D101	
	D200	LABEL3
	D201	
	D202	
	D203	
	D300 : D316[*5]	LABEL4
	D400	LABEL5
	D401	

[*5] : The number of points of characters to be read is 17 (32 characters + NULL). The characters need to be converted in a user program because the characters of String type are not converted.

<When array type label is specified>

Example: Read data from the devices by specifying array type labels.

- Label setting

szLabelList		Data type	Device
[0]	LABEL1	Bit (0..1)	M0
[1]	LABEL2	Double Word (0..1)	CN200
[2]	LABEL3	Double Word (0..1)	D0

- Number of read points: 8
- Read device values

Upper 2 bytes	Lower 2 bytes	Applicable label
Not used (0 is stored.)	M0	LABEL1[0]
	M1	LABEL1[1]
H of CN200	L of CN200	LABEL2[0]
H of CN201	L of CN201	LABEL2[1]
Not used (0 is stored.)	D100	LABEL3[0]
	D101	
	D102	LABEL3[1]
	D103	

<When structure type labels are specified>

Example: Read data from the devices by specifying structure type labels.

- Structure setting

Structure name	Label name	Data type
STRUCT	L1	Bit
	L2	Double Word

- Label setting

szLabelList		Data type	Label name	Device
[0]	LABEL1	STRUCT	L1	D0.0
			L2	D0
[1]	LABEL2	STRUCT	L1	M10
			L2	CN200

- Number of read points: 6
- Read device values

Upper 2 bytes	Lower 2 bytes	Applicable label
Not used (0 is stored.)	D0.0	LABEL1.L1
	D0	LABEL1.L2
	D1	
	M0	LABEL2.L1
H of CN200	L of CN200	LABEL2.L2
H of CN201	L of CN201*6	

*6 : Data of two devices are read when the device of CN200 and later is specified for the Double Word type label.

<When labels with combined structure and label are specified>

Example: Read data from the devices by specifying structure type array and structure array type label.

- Structure setting

Structure name	Label name	Data type
STRUCT1	L1	Bit
	L2	Word
STRUCT2	L1	Bit (0..2)
	L2	Double Word

- Label setting

szLabelList		Data type	Label name	Device
[0]	LABEL1	STRUCT1 (0..1)	L1	X0
			L2	D0
[1]	LABEL2	STRUCT2	L1	M0
			L2	D100

- Number of read points: 9
- Read device values

Upper 2 bytes	Lower 2 bytes	Applicable label
Not used (0 is stored.)	X0	LABEL1[0].L1
	D0	LABEL1[0].L2
	X1	LABEL1[1].L1
	D1	LABEL1[1].L2
	M0	LABEL2.L1[0]
	M1	LABEL2.L1[1]
	M2	LABEL2.L1[2]
	D100	LABEL2.L2
	D101	

(6) Returned value

Normal termination : 0 is returned.

Abnormal termination: Any value other than 0 is returned. (☞ Page 378, CHAPTER 7 ERROR CODES)

Point

- The maximum number of read points is 0x7FFFFFFF.
- For the number of read points, specify the number of words which applies to the data type specified for the label name. For the read device values, reserve a memory area for the number of points specified for the number of read points. If the memory area is not reserved, a critical error (an application error or the like) may occur.
- When a device which corresponds to the label name does not exist, an error occurs and data cannot be read. When any one of devices which correspond to multiple label names does not exist, an error occurs and data cannot be read.
- Digit specified bit device and index setting cannot be used.

5.3.6 WriteDeviceRandom (Writing devices randomly)

(1) Applicable control

Applicable to the DotUtlType control.

(2) Feature

Write devices randomly.

(3) Format

(a) Visual Basic® .NET

Ret = object.WriteDeviceRandom(szLabel, iSize, iData(0))

Integer	lRet	Returned value	Output
String	szLabel	Label name	Input
Integer	iSize	Number of write points	Input
Integer	iData(n)	Device values to be written	Input

Ret = object.WriteDeviceRandom(szLabelList, iSize, iData(0))

Integer	lRet	Returned value	Output
String	szLabelList(n)	Label list	Input
Integer	iSize	Number of write points	Input
Integer	iData(n)	Device values to be written	Input

(b) Visual C++® .NET

iRet = object.WriteDeviceRandom(*szLabel, iSize, *ipiData)

int	iRet	Returned value	Output
String*	szLabel	Label name	Input
int	iSize	Number of write points	Input
int*	ipiData	Device values to be written	Input

iRet = object.WriteDeviceRandom(**szLabelList, iSize, *ipiData)

int	iRet	Returned value	Output
String**	szLabelList	Label list	Input
int	iSize	Number of write points	Input
int*	ipiData	Device values to be written	Input

(4) Description

- Data of a device group for the size of iSize specified for the label name szLabel (szLabelList) are written.
- The device values to be written are stored in iData (ipiData).
- For iData (ipiData), reserve arrays for more than the amount specified for iSize.

(5) How to specify devices

The following describes how to specify label names and device values to be written.

- The following data type can be specified for label name.

Type class		Label data type	Label name format
Basic type		Bit, Word, Double Word, Float (Single Precision), Float (Double Precision), String[*1], Time, Timer, Counter, Retentive Timer	Label name
Array			Label name
	Member	(The label data type can be specified in the same manner as the basic type.)	Label name [number of elements] Label name [n1] [n2] [n3]
Structure			Label name
	Member	(The label data type can be specified in the same manner as the basic type.)	Label name.Member name
Structured array			Label name [number of elements]
	Member	(The label data type can be specified in the same manner as the basic type.)	Label name [number of elements].Member name Label name [n1] [n2] [n3].Element

*1 : The maximum number of characters that can be specified is 32 + NULL.

- For the number of write points, specify the sum of the following values correspond to the elements according to the label data type.

Label data type	Number of applicable words	Number of write points to be specified
Bit, Word	1	Number of label elements
Double Word, Float (Single Precision)	2	Number of label elements multiplied by 2
Float (Double Precision)	4	Number of label elements multiplied by 4
String	17	Number of label elements multiplied by 17
Time	2	Number of label elements multiplied by 2
Timer, Counter, Retentive Timer	1	Number of label elements

- Set the device values to be written as follows.

<When bit device and word device are specified>

Example: Write data to each 1 point of M0 and D0.

- Label setting

szLabelList		Data type	Device
[0]	LABEL1	Bit	M0
[1]	LABEL2	Word	D0

- Number of write points: 2
- Device values to be written

Upper 2 bytes	Lower 2 bytes	Applicable label
Not used	M0	LABEL1
	D0	LABEL2

<When FXCPU devices of CN200 and later are specified>

Example: Write 3 points of data to the devices including CN200.[*2]

- Label setting

szLabelList		Data type	Device
[0]	LABEL1	Word	D0
[1]	LABEL2	Double Word	CN200
[2]	LABEL3	Word	D1

- Number of write points: 3
- Device values to be written

Upper 2 bytes	Lower 2 bytes	Applicable label
Not used	D0	LABEL1
H of CN200	L of CN200	LABEL2
Not used	D1	LABEL3

*2 : For FXCPU devices of CN200 and later, 4 bytes are written as 1 read point.

<When FD device is specified (4-word device)>

Example: Write 3 points of data to the devices including FD0.

- Label setting

szLabelList		Data type	Device
[0]	LABEL1	Word	D0
[1]	LABEL2	Double Word	FD0
[2]	LABEL3	Word	D1

- Number of write points: 3
- Device values to be written

Upper 2 bytes	Lower 2 bytes	Applicable label
Not used	D0	LABEL1
	LL of FD0[*3]	LABEL2
	D1	LABEL3

*3 : Only lower 2 bytes can be set. "0" is written to the specified devices HH, HL, and LH (upper 6 bytes).

<When data type equivalent to 2 words or more is specified for label>

Example: Write data by specifying labels of Double Word, Float (Single Precision), Float (Double Precision), String*4, and Time types.

- Label setting

szLabelList		Data type	Device
[0]	LABEL1	Double Word	D0
[1]	LABEL2	Float (Single Precision)	D100
[2]	LABEL3	Float (Double Precision)	D200
[3]	LABEL4	String	D300
[4]	LABEL5	Time	D400

- Number of write points: 27
- Device values to be written

Upper 2 bytes	Lower 2 bytes	Applicable label
Not used	D0	LABEL1
	D1	
	D100	LABEL2
	D101	
	D200	LABEL3
	D201	
	D202	
	D203	
	D300 : D316*4	LABEL4
	D400	LABEL5
	D401	

*4 : The number of points of characters to be written is 17 (32 characters + NULL). The characters need to be converted in a user program because the characters of String type are not converted.

<When array type label is specified>

Example: Write data to the devices by specifying array type labels.

- Label setting

szLabelList		Data type	Device
[0]	LABEL1	Bit (0..1)	M0
[1]	LABEL2	Double Word (0..1)	CN200
[2]	LABEL3	Double Word (0..1)	D0

- Number of write points: 8
- Device values to be written

Upper 2 bytes	Lower 2 bytes	Applicable label
Not used (0 is stored.)	M0	LABEL1[0]
	M1	LABEL1[1]
H of CN200	L of CN200	LABEL2[0]
H of CN201	L of CN201	LABEL2[1]
Not used	D100	LABEL3[0]
	D101	
	D102	LABEL3[1]
	D103	

<When structure type labels are specified>

Example: Write data to the devices by specifying structure type labels.

- Structure setting

Structure name	Label name	Data type
STRUCT	L1	Bit
	L2	Double Word

- Label setting

szLabelList		Data type	Label name	Device
[0]	LABEL1	STRUCT	L1	D0.0
			L2	D0
[1]	LABEL2	STRUCT	L1	M10
			L2	CN200

- Number of write points: 5
- Device values to be written

Upper 2 bytes	Lower 2 bytes	Applicable label
Not used	D0.0	LABEL1.L1
	D0	LABEL1.L2
	D1	
	M0	LABEL2.L1
H of CN200	L of CN200	LABEL2.L2

<When labels with combined structure and label are specified>

Example: Write data to the devices by specifying structure type array and structure array type label.

- Structure setting

Structure name	Label name	Data type
STRUCT1	L1	Bit
	L2	Word
STRUCT2	L1	Bit (0..2)
	L2	Double Word

- Label setting

szLabelList		Data type	Label name	Device
[0]	LABEL1	STRUCT1 (0..1)	L1	X0
			L2	D0
[1]	LABEL2	STRUCT2	L1	M0
			L2	D100

- Number of write points: 9
- Device values to be written

Upper 2 bytes	Lower 2 bytes	Applicable label
Not used	X0	LABEL1[0].L1
	D0	LABEL1[0].L2
	X1	LABEL1[1].L1
	D1	LABEL1[1].L2
	M0	LABEL2.L1[0]
	M1	LABEL2.L1[1]
	M2	LABEL2.L1[2]
	D100	LABEL2.L2
	D101	

(6) Returned value

Normal termination : 0 is returned.

Abnormal termination: Any value other than 0 is returned. (☞ Page 378, CHAPTER 7 ERROR CODES)

Point

- ● The maximum number of write points is 0x7FFFFFFF.
- ● For the number of write points, specify the number of words which applies to the data type specified for the label name. For the device values to be written, reserve a memory area for the number of points specified for the number of write points. If the memory area is not reserved, a critical error (an application error or the like) may occur.
- ● When a device which corresponds to the label name does not exist, an error occurs and data cannot be written. When any one of devices which correspond to multiple label names does not exist, an error occurs and data cannot be written.
- ● Digit specified bit device and index setting cannot be used.

5.3.7 SetDevice (Setting device data)

(1) Applicable control

Applicable to the DotUtlType control.

(2) Feature

Set one point of device.

(3) Format

(a) Visual Basic® .NET

IRet = object.SetDevice(szLabel, iData)

Integer	IRet	Returned value	Output
String	szLabel	Label name	Input
Integer	iData	Device values to be written	Input

(b) Visual C++® .NET

iRet = object.SetDevice(*szLabel, iData)

int	iRet	Returned value	Output
String*	szLabel	Label name	Input
int	iData	Device values to be written	Input

(4) Description

- The values of iData (device values to be written) are written to 1 point of device specified for szLabel (label name).
- When specifying bit devices, the least significant bit of the iData (device values to be written) is valid.

(5) How to specify devices

The following describes how to specify label names and device values to be written.

- The following data type can be specified for label name.

Type class	Label data type	Label name format
Basic type	Bit, Word, Double Word*1, Float (Single Precision)*1, Float (Double Precision)*1, String*1, Time*1, Timer, Counter, Retentive Timer	Label name

*1 : Only lower 2 bytes of start device can be written.
When writing 2 words or more of devices, use the WriteDeviceBlock function.
Note that when any of the following devices is specified, the upper 2 bytes are also written in Double Word type.

- Digit specified bit devices
- FXCPU devices of CN200 and later

- Set the device values to be written as follows.

<When bit device is specified>

Example: Write data to M0.

- Label setting

Data type	Device
Bit	M0

- Device values to be written

Upper 2 bytes	Lower 2 bytes
Not used	M0

<When CN200 or later of FXCPU is specified>

Example: Write data to CN200.[*2]

- Label setting

Data type	Device
Double Word	CN200

- Device values to be written

Upper 2 bytes	Lower 2 bytes
H of CN200	L of CN200

*2 : For FXCPU devices of CN200 and later, 4 bytes are written.

<When data type equivalent to 2 words or more is specified for label>

Example: Write data by specifying String type label.

- Label setting

Data type	Device
String	D0

- Device values to be written

Upper 2 bytes	Lower 2 bytes
Not used	D0[*4]

*4 : Only lower 2 bytes are written to the start device.

<When word device is specified>

Example: Write data to D0.

- Label setting

Data type	Device
Word	D0

- Device values to be written

Upper 2 bytes	Lower 2 bytes
Not used	D0

<When FD device is specified (4-word device)>

Example: Write data by specifying FD0.

- Label setting

Data type	Device
Word	FD0

- Device values to be written

Upper 2 bytes	Lower 2 bytes
Not used	LL of FD0[*3]

*3 : Only lower 2 bytes can be set. "0" is written to the specified devices HH, HL, and LH (upper 6 bytes).

(6) Returned value

Normal termination : 0 is returned.

Abnormal termination: Any value other than 0 is returned. (☞ Page 378, CHAPTER 7 ERROR CODES)

Point

- When a device which corresponds to the label name does not exist, an error occurs and data cannot be written.
- When a Double Word device is specified at the time other than the digit specification for bit device and specification of FXCPU devices of CN200 and later, this function writes data of lower 1 word (2 bytes), and "0" is written to data of upper 1 word (2 bytes).
  When writing Double Word devices, use the WriteDeviceRandom or the WriteDeviceBlock2 function.
- Digit specified bit device and index setting cannot be used.

5.3.8 GetDevice (Acquiring device data)

(1) Applicable control

Applicable to the DotUtlType control.

(2) Feature

Acquire one point of device.

(3) Format

(a) Visual Basic® .NET

IRet = object.GetDevice(szLabel, iData)

Integer	IRet	Returned value	Output
String	szLabel	Label name	Input
Integer	iData	Read device values	Output

(b) Visual C++® .NET

iRet = object.GetDevice(*szLabel, *ipiData)

int	iRet	Returned value	Output
String*	szLabel	Label name	Input
int*	ipiData	Read device values	Output

(4) Description

One point of device data specified for szLabel (label name) is stored in iData (ipiData) (read device values).

(5) How to specify devices

The following describes how to specify label names and device values to be read.

- The following data type can be specified for label name.

Type class	Label data type	Label name format
Basic type	Bit, Word, Double Word[*1], Float (Single Precision)[*1], Float (Double Precision)[*1], String[*1], Time[*1], Timer, Counter, Retentive Timer	Label name

*1 : Only lower 2 bytes of start device can be read.
When reading 2 words or more of devices, use the ReadDeviceBlock function.
Note that when any of the following devices is specified, the upper 2 bytes are also read in Double Word type.

- Digit specified bit devices
- FXCPU devices of CN200 and later

- The read device values are stored as follows.

<When bit device is specified>

Example: Read data from M0.

- Label setting

Data type	Device
Bit	M0

- Read device values

Upper 2 bytes	Lower 2 bytes
Not used (0 is stored.)	M0[*2]

*2 : The device to be read is 1 point of "M0", and "0" or "1" is stored for the device value.

<When CN200 or later of FXCPU is specified>

Example: Read data from CN200.[*3]

- Label setting

Data type	Device
Double Word	CN200

- Read device values

Upper 2 bytes	Lower 2 bytes
H of CN200	L of CN200

*3 : For FXCPU devices of CN200 and later, 4 bytes are read.

<When data type equivalent to 2 words or more is specified for label>

Example: Read data by specifying String type label.

- Label setting

Data type	Device
String	D0

- Read device values

Upper 2 bytes	Lower 2 bytes
Not used (0 is stored.)	D0[*5]

*5 : Only lower 2 bytes are read to the start device.

<When word device is specified>

Example: Read data from D0.

- Label setting

Data type	Device
Word	D0

- Read device values

Upper 2 bytes	Lower 2 bytes
Not used (0 is stored.)	D0

<When FD device is specified (4-word device)>

Example: Read data by specifying FD0.

- Label setting

Data type	Device
Word	FD0

- Read device values

Upper 2 bytes	Lower 2 bytes
Not used (0 is stored.)	LL of FD0[*4]

*4 : Only lower 2 bytes are read. Data are not read from the specified devices HH, HL, and LH (upper 6 bytes).

(6) Returned value

Normal termination : 0 is returned.

Abnormal termination: Any value other than 0 is returned. (☞ Page 378, CHAPTER 7 ERROR CODES)

Point

- ● When a device which corresponds to the label name does not exist, an error occurs and data cannot be read.
- ● For the read device values, reserve 4 bytes of memory area. If the memory area is not reserved, a critical error (an application error or the like) may occur.
- ● Digit specified bit device and index setting cannot be used.

5.3.9 ReadBuffer (Reading data from buffer memory)

(1) Applicable control

Applicable to the DotUtlType control.

(2) Feature

Read data from the buffer memory of special function module.

(3) Format

(a) Visual Basic® .NET

IRet = object.ReadBuffer(iStartIO, iAddress, iReadSize, sData(0))

Integer	IRet	Returned value	Output
Integer	iStartIO	Start I/O number of module from	Input
Integer	iAddress	Buffer memory address	Input
Integer	iReadSize	Read size	Input
Short	sData(n)	Values read from buffer memory	Output

(b) Visual C++® .NET

iRet = object.ReadBuffer(iStartIO, iAddress, iReadSize, *ipsData)

int	iRet	Returned value	Output
int	iStartIO	Start I/O number of module from	Input
int	iAddress	Buffer memory address	Input
int	iReadSize	Read size	Input
short*	ipsData	Values read from buffer memory	Output

(4) Description

- For the start I/O number of the module specified for iStartIO, specify the value of the actual start I/O number divided by 16.
- Buffer values of buffer memory address specified for iAddress of the special function module of the start I/O number specified for iStartIO are read for the size of iReadSize.
- When performing CPU COM communication with FXCPU as a connected station or GX Simulator communication, specify the block number (0 to 7) of the special expansion equipment for the module's start I/O number and any value between 0 and 32767 for the buffer memory address.
- For sData (ipsData), reserve arrays for more than the amount specified for iReadSize .

(5) Returned value

Normal termination : 0 is returned.
Abnormal termination: Any value other than 0 is returned. (☞ Page 378, CHAPTER 7 ERROR CODES)

(6) Considerations for reading/writing data from/to buffer memory in multiple CPU system configuration

Any of the following errors occurs when the function is executed under the situation where the configured multiple CPU system differs from the I/O assignment set in GX Developer.

Execute the function after checking the I/O assignment set in GX Developer and checking the I/O numbers of the module from/to which values are read/written.

- An error occurs even when the function is executed after the correct I/O numbers are specified.
- When the specific I/O numbers (I/O numbers whose I/O assignment is actually wrong) are specified, data can be read from buffer memory normally, but an error (error code: 0x010A4030, 0x010A4042, etc.) occurs when data are written to the buffer memory.
- Though a programmable controller CPU error (parameter error, SP. UNIT LAY ERR, etc.) does not occur in the multiple CPU system, an error occurs in the user application when the function is executed.

Point

- ● If a Q motion CPU is accessed, an error is returned.
- ● For sData (ipsData), reserve a memory area for the number of points specified for iReadSize.
  If the memory area is not reserved, a critical error (an application error or the like) may occur.
- ● When reading data from buffer memory (ReadBuffer) on QCPU (Q mode), the read operation can only be performed on the Q series-dedicated modules.
  Furthermore, data cannot be read from the shared memory of QCPU (Q mode).

5.3.10 WriteBuffer (Writing data to buffer memory)

(1) Applicable control

Applicable to the DotUtlType control.

(2) Feature

Write data to the buffer memory of special function module.

(3) Format

(a) Visual Basic® .NET

lRet = object.WriteBuffer(iStartIO, iAddress, iWriteSize, sData(0))

Integer	lRet	Returned value	Output
Integer	iStartIO	Start I/O number of module to where values will be written	Input
Integer	iAddress	Buffer memory address	Input
Integer	iWriteSize	Write size	Input
Short	sData(n)	Values written from buffer memory	Output

(b) Visual C++® .NET

iRet = object.WriteBuffer(iStartIO, iAddress, iWriteSize, *ipsData)

int	iRet	Returned value	Output
int	iStartIO	Start I/O number of module to where values will be written	Input
int	iAddress	Buffer memory address	Input
int	iWriteSize	Write size	Input
Short*	ipsData	Values written from buffer memory	Output

(4) Description

- For the start I/O number of the module specified for iStartIO, specify the value of the actual start I/O number divided by 16.
- Buffer values of buffer memory address specified for iAddress of the special function module of the start I/O number specified for iStartIO are written for the size of iWriteSize.
- When performing CPU COM communication with FXCPU as a connected station or GX Simulator communication, specify the block number (0 to 7) of the special expansion equipment for the module's start I/O number and any value between 0 and 32767 for the buffer memory address.
- For sData (ipsData), reserve arrays for more than the amount specified for iWriteSize.

(5) Returned value

Normal termination : 0 is returned.
Abnormal termination: Any value other than 0 is returned. (☞ Page 378, CHAPTER 7 ERROR CODES)

(6) Considerations for reading/writing data from/to buffer memory in multiple CPU system configuration

Any of the following errors occurs when the function is executed under the situation where the configured multiple CPU system differs from the I/O assignment set in GX Developer.

Execute the function after checking the I/O assignment set in GX Developer and checking the I/O numbers of the module from/to which the values are read/written.

- An error occurs even when the function is executed after the correct I/O numbers are specified.
- When the specific I/O numbers (I/O numbers whose I/O assignment is actually wrong) are specified, data can be read from buffer memory normally, but an error (error code: 0x010A4030, 0x010A4042, etc.) occurs when data are written to the buffer memory.
- Though a programmable controller CPU error (parameter error, SP. UNIT LAY ERR, etc.) does not occur in the multiple CPU system, an error occurs in the user application when the function is executed.

Point

- ● If a Q motion CPU is accessed, an error is returned.
- ● For sData (ipsData), reserve a memory area for the number of points specified for iWriteSize .
  If the memory area is not reserved, a critical error (an application error or the like) may occur.
- ● When buffer memory write (WriteBuffer) is performed for QCPU (Q mode), write operation may be performed for only the Q series-dedicated module.
  Furthermore, data cannot be written to the shared memory of QCPU (Q mode).

5.3.11 GetClockData (Reading clock data)

(1) Applicable control

Applicable to the DotUtlType control.

(2) Feature

Read clock data from a programmable controller CPU.

(3) Format

(a) Visual Basic® .NET

IRet = object.GetClockData(sYear, sMonth, sDay, sDayOfWeek, sHour, sMinute, sSecond)

Integer	IRet	Returned value	Output
Short	sYear	Read year value	Output
Short	sMonth	Read month value	Output
Short	sDay	Read day value	Output
Short	sDayOfWeek	Read day-of-week value	Output
Short	sHour	Read hour value	Output
Short	sMinute	Read minute value	Output
Short	sSecond	Read second value	Output

(b) Visual C++® .NET

iRet = object.GetClockData(*ipsYear, *ipsMonth, *ipsDay, *ipsDayOfWeek, *ipsHour, *ipsMinute, *ipsSecond)

int	iRet	Returned value	Output
short*	ipsYear	Read year value	Output
short*	ipsMonth	Read month value	Output
short*	ipsDay	Read day value	Output
short*	ipsDayOfWeek	Read day-of-week value	Output
short*	ipsHour	Read hour value	Output
short*	isMinute	Read minute value	Output
short*	ipsSecond	Read second value	Output

(4) Description

- An error is returned when the correct clock data is not set to the programmable controller CPU.
- The values stored in sYear (ipsYear) are: four digits of the year for QCPU (Q mode), and last two digits of the year for any other CPUs.
  Note that the applicable years for QCPU (Q mode) are from 1980 to 2079.
- The values stored in sDayOfWeek (ipsDayOfWeek) are as follows.

Value	Day of Week
0	Sunday
1	Monday
2	Tuesday
3	Wednesday
4	Thursday
5	Friday
6	Saturday

(5) Returned value

Normal termination : 0 is returned.

Abnormal termination: Any value other than 0 is returned. (☞ Page 378, CHAPTER 7 ERROR CODES)

Point

- If a Q motion CPU is accessed, an error is returned.
- For FXCPU, clock data can be read from FX1N, FX1NC, FX1S, FX2N, or FX3G when it has a built-in clock, or from FXU, FX2C, or FX2NC when it is installed with the RTC cassette.
  An error is returned when clock data is read from FXCPU other than FX1N, FX1NC, FX1S, FXU, FX2C, FX2N, FX2NC, FX3U, FX3UC, and FX3G.
- Note that an error of transfer time may occur in clock setting.

5.3.12 SetClockData (Writing clock data)

(1) Applicable control

Applicable to the DotUtlType control.

(2) Feature

Write clock data to a programmable controller CPU.

(3) Format

(a) Visual Basic® .NET

IRet = object.SetClockData(sYear, sMonth, sDay, sDayOfWeek, sHour, sMinute, sSecond)

Integer	IRet	Returned value	Output
Short	sYear	Year value to be written	Input
Short	sMonth	Month value to be written	Input
Short	sDay	Day value to be written	Input
Short	sDayOfWeek	Day-of-week value to be written	Input
Short	sHour	Hour value to be written	Input
Short	sMinute	Minute value to be written	Input
Short	second	Second value to be written	Input

(b) Visual C++® .NET

iRet = object.SetClockData(sYear, sMonth, sDay, sDayOfWeek, sHour, sMinute, sSecond)

int	iRet	Returned value	Output
short	sYear	Year value to be written	Input
short	sMonth	Month value to be written	Input
short	sDay	Day value to be written	Input
short	sDayOfWeek	Day-of-week value to be written	Input
short	sHour	Hour value to be written	Input
short	sMinute	Minute value to be written	Input
short	sSecond	Second value to be written	Input

(4) Description

- An error is returned when the clock data to be set are not correct values.
- The applicable values to be specified for sYear are: four digits of the year for QCPU (Q mode), and last two digits of the year for any other CPUs.
  Note that the applicable years for QCPU (Q mode) are from 1980 to 2079.
  An error occurs when four digits are set to a CPU other than QCPU (Q mode).
- The values to be specified for sDayOfWeek are as follows.

Value	Day of Week
0	Sunday
1	Monday
2	Tuesday
3	Wednesday
4	Thursday
5	Friday
6	Saturday

(5) Returned value

Normal termination : 0 is returned.

Abnormal termination: Any value other than 0 is returned. (☞ Page 378, CHAPTER 7 ERROR CODES)

Point

- If a Q motion CPU is accessed, an error is returned.
- For FXCPU, clock data can be set to FX1N, FX1NC, FX1S, FX2N, or FX3G when it has a built-in clock, or to FXU, FX2C, or FX2NC when it is installed with the RTC cassette.
  An error is returned when clock data is set to FXCPU other than FX1N, FX1NC, FX1S, FXU, FX2C, FX2N, FX2NC, FX3U, FX3UC, and FX3G.
- Note that an error of transfer time may occur in clock setting.

5.3.13 GetCpuType (Reading programmable controller CPU model)

(1) Applicable control

Applicable to the DotUtlType control.

(2) Feature

Read the model character string and the model code of programmable controller CPU, network board, and GOT.

(3) Format

(a) Visual Basic® .NET

IRet = object.GetCpuType(szCpuName, ICpuType)

Integer	IRet	Returned value	Output
String	szCpuName	Programmable controller CPU model character string	Output
Integer	ICpuType	Programmable controller CPU model code	Output

(b) Visual C++® .NET

iRet = object.GetCpuType (**szCpuName, *ipiCpuType)

int	iRet	Returned value	Output
String**	szCpuName	Programmable controller CPU model character string	Output
int*	ipiCpuType	Programmable controller CPU model code	Output

(4) Description

- The model and the model code of the communication target programmable controller CPU are stored in szCpuName and ICpuType (ipiCpuType) respectively.
- The model character string of the programmable controller CPU is returned in UNICODE.

(5) Model character string and model code of CPU

For details, refer to the following section.
☞ Page 249, Section 5.2.13

(6) Returned value

Normal termination : 0 is returned.
Abnormal termination: Any value other than 0 is returned. (☞ Page 378, CHAPTER 7 ERROR CODES)

5.3.14 SetCpuStatus (Remote control)

(1) Applicable control

Applicable to the DotUtlType control.

(2) Feature

Perform a remote operation of programmable controller CPU.

(3) Format

(a) Visual Basic® .NET

lRet = object.SetCpuStatus(lOperation)

Integer	lRet	Returned value	Output
Integer	lOperation	Remote RUN/STOP/PAUSE	Input

(b) Visual C++® .NET

iRet = object.SetCpuStatus (iOperation)

int	iRet	Returned value	Output
int	iOperation	Remote RUN/STOP/PAUSE	Input

(4) Description

The operation specified for lOperation (varOperation) is performed.

An error occurs when a value other than the following values is specified.

Value	Operation
0	Remote RUN
1	Remote STOP
2	Remote PAUSE

(5) Returned value

Normal termination : 0 is returned.

Abnormal termination: Any value other than 0 is returned. (☞ Page 378, CHAPTER 7 ERROR CODES)

Point

- Since FXCPU does not have the PAUSE switch as a programmable controller CPU, an error is returned if a remote pause operation is specified using the SetCpuStatus function.
- If a Q motion CPU is accessed and PAUSE is specified, an error is returned.

5.3.15 EntryDeviceStatus (Registering devices for status monitoring)

(1) Applicable control

Applicable to the DotUtlType control.

(2) Feature

Register devices whose status to be monitored.

(3) Format

(a) Visual Basic® .NET

IRet = object.EntryDeviceStatus(szLabelList, iSize, iMonitorCycle, iData(0))

Integer	IRet	Returned value	Output
String	szLabelList(n)	Registered label name list	Input
Integer	iSize	Number of registered device points	Input
Integer	iMonitorCycle	Status monitor time interval	Input
Integer	iData(n)	Registered device value list	Input

(b) Visual C++® .NET

iRet = object.EntryDeviceStatus(**szLabelList, iSize, iMonitorCycle, *arriData)

int	iRet	Returned value	Output
String**	szLabelList	Registered label name list	Input
int	iSize	Number of registered device points	Input
int	iMonitorCycle	Status monitor time interval	Input
int*	arriData	Registered device value list	Input

(4) Description

- A device group for the size of iSize (varSize) specified for szDeviceList (varDeviceList) is checked whether it is in the status specified for iData (iplData or varData).
  Specify the check time for iMonitorCycle (varMonitorCycle).
  When the status is established, the OnDeviceStatus function of the user application is executed.
- The maximum number of device points that can be specified for iSize (varSize) is 20 points.
- Specify a value within the range from 1 second to 1 hour (set between 1 and 3600 in seconds) for iMonitorCycle (varMonitorCycle).
  An error occurs when any other value outside the above range is specified.
- The registered device value list is stored in iData (iplData or varData).

(5) How to specify devices

The following explains how to specify label names and registered device values.

- The following data type can be specified for label name.

Type class	Label data type	Label name format
Basic type	Bit, Word	Label name

- Set the device values to be registered as follows.

<When bit device and word device are specified>

Example: Register 1 point of M0 and D0.

- Label setting

szLabelList		Data type	Device
[0]	LABEL1	Bit	M0
[1]	LABEL2	Word	D0

- Number of registered device points: 2
- Registered device values

Upper 2 bytes	Lower 2 bytes
Not used	M0
	D0

<When CN200 or later of FXCPU is specified>

Example: Register 3 points of devices including CN200.

- Label setting

szLabelList		Data type	Device
[0]	LABEL1	Word	D0
[1]	LABEL2	Word	CN200
[2]	LABEL3	Word	D1

- Number of registered device points: 3
- Registered device values

Upper 2 bytes	Lower 2 bytes
Not used	D0
H of CN200	L of CN200*1
Not used	D1

*1 : For FXCPU devices of CN200 and later, 4 bytes can be registered.

<When FD device is specified (4-word device)>

Example: Register 3 points of data from FD0.

- Label setting

szLabelList		Data type	Device
[0]	LABEL1	Word	D0
[1]	LABEL2	Word	FD0
[2]	LABEL3	Word	D1

- Number of registered device points: 3
- Registered device values

Upper 2 bytes	Lower 2 bytes
Not used	D0
	LL of FD0*2
	D1

*2 : Only lower 2 bytes can be registered. The specified devices HH, HL, and LH (upper 6 bytes) cannot be registered.

(6) Returned value

Normal termination : 0 is returned.

Abnormal termination: Any value other than 0 is returned. (☞ Page 378, CHAPTER 7 ERROR CODES)

(7) Considerations for checking the word device status

When checking the word device status for negative values of -1 to -32768 (FFFFH to 8000H), set the monitor device value of the EntryDeviceStatus function to any of 65535 to 32768 (0000FFFFH to 00008000H) where "0" is stored in the upper 2 bytes.

(Example) When checking the D0 status for "-10"

Set the value "65526 (0000FFF6H)" where "0" is stored in the upper 2 bytes of "-10 (FFFFFFF6H)" for the monitor device value.

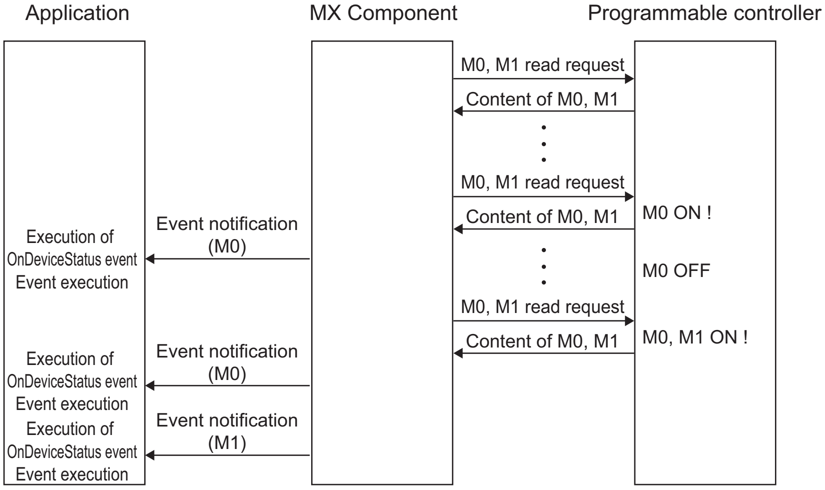
While the type of word devices of the programmable controller CPU is WORD type, the type of monitor device value of the EntryDeviceStatus function is LONG type. Therefore, when current values of programmable controller CPU are compared with monitor device values of the EntryDeviceStatus function, the values do not match and the above setting is required. (When bit devices or Double Word devices are used, this consideration does not apply.)

For the programming examples regarding this consideration, refer to the following section.

☞ Page 412, Appendix 3 Time-Out Periods

Point

- Device status monitoring may not be performed at the specified status monitoring time intervals depending on the conditions: personal computer performance, currently executed application load, time required for communication with the programmable controller, or the like. Simultaneous use of any other control functions would also be the cause of disabling device status monitoring at the specified status monitoring time intervals.
- For iData (iplData or lpvarData), reserve a memory area for the number of points specified for iSize (varSize). If the memory area is not reserved, a critical error (an application error or the like) may occur.
- An error occurs when the EntryDeviceStatus function is executed during a status monitoring. When changing any status monitor condition, execute the FreeDeviceStatus function and then execute the EntryDeviceStatus function.
- When the status of multiple devices changes at the same time, the OnDeviceStatus event is executed every time the status changes.
(Example: When M0 is monitored)



- This function is a function to check the status establishment under the constant execution of random device read by the control. This function is not a function for a programmable controller CPU to notify the device status establishment to MX Component. Therefore, the control may not be able to check the device status establishment of programmable controller CPU depending on the specified status monitoring time interval.
- Digit specified bit device and index setting cannot be used.

5.3.16 FreeDeviceStatus (Deregistering devices for status monitoring)

(1) Applicable control

Applicable to the DotUtlType control.

(2) Feature

Deregister devices that are registered using the EntryDeviceStatus function to monitor their status.

(3) Format

(a) Visual Basic® .NET

IRet = object.FreeDeviceStatus()

Integer	IRet	Returned value	Output

(b) Visual C++® .NET

iRet = object.FreeDeviceStatus()

int	iRet	Returned value	Output

(4) Description

The devices that are set using the EntryDeviceStatus function to monitor their status are deregistered.

(5) Returned value

Normal termination : 0 is returned.

Abnormal termination: Any value other than 0 is returned. (☞ Page 378, CHAPTER 7 ERROR CODES)

5.3.17 OnDeviceStatus (Event notification)

(1) Applicable control

Applicable to the DotUtlType control.

(2) Feature

Execute the event notification when the device condition registered using the EntryDeviceStatus function is satisfied.

(3) Format

(a) Visual Basic® .NET

Private Sub object_OnDeviceStatus(sender, e)

ByVal sender As System.Object	Event occurrence source	Input
ByVal e As objectLib_IobjectIFEvents_OnDeviceStatusEvent	Event data	Input

The following are the members of event data e.

e.szLabel	Name of label whose condition is satisfied
e.lData	Value of device whose condition is satisfied
e.lReturnCode	Returned value of condition check processing

(b) Visual C++® .NET

private: System::Void objectIF_OnDeviceStatus (*sender, e)

System::Object *sender	Event occurrence source	Input
objectLib::objectIF_OnDeviceStatusEvent* e	Event data	Input

The following are the members of event data e.

e->szLabel	Name of label whose condition is satisfied
e->lData	Value of device whose condition is satisfied
e->lReturnCode	Returned value of condition check processing

(4) Description

- The event is notified to the application when the device condition registered using the EntryDeviceStatus function is satisfied.
  Programming this function in the user application allows the application to receive the event when the registered device condition is satisfied.
- Device values registered using the EntryDeviceStatus function are input to iData (varData).
  Example: When the word device is monitored for the value of "-1"
  Set 65535 (0000FFFFH) as a registered device value using the EntryDeviceStatus function.
  When the value of the target word device of the programmable controller CPU becomes "-1" (FFFFH), the OnDeviceStatus function is executed and (0000FFFFH) is input to iData (varData).

(5) Returned value

Normal termination : 0 is returned.

Abnormal termination: Any value other than 0 is returned. (☞ Page 378, CHAPTER 7 ERROR CODES)

Point

● When any of the following settings is set in the user application, the event of the OnDeviceStatus function does not occur even if the condition of the device registered to the EntryDeviceStatus function is satisfied. Note that when the event occurrence becomes in wait status, the control function is not returned to the control and device management processing stops until the following setting is terminated.

- User applications created using Visual Basic® .NET
  The message box is displayed in the user application.
  The InputBox/OutputBox is displayed in the user application.
- User applications created using Visual Basic® .NET, Visual C++® .NET
  The Sleep processing, WaitForSingleObject function, or similar standby function is used in the user application.

5.3.18 ReadDeviceBlock2 (Reading devices in bulk)

(1) Applicable control

Applicable to the DotUtlType control.

(2) Feature

Read devices in 2-byte data unit in bulk.

(3) Format

(a) Visual Basic® .NET

IRet = object.ReadDeviceBlock2(szLabel, iSize, sData)

Integer	IRet	Returned value	Output
String	szLabel	Label name	Input
Integer	iSize	Number of read points	Input
Short	sData(n)	Read device values	Output

(b) Visual C++® .NET

iRet = object.ReadDeviceBlock2(*szLabel, iSize, *ipsData)

int	iRet	Returned value	Output
String*	szLabel	Label name	Input
int	iSize	Number of read points	Input
short*	lpsData	Read device values	Output

(4) Description

- The devices for the amount specified for iSize (number of read points) are read in bulk starting from the device specified for szLabel (label name).
- The read device values are stored in sData (lpsData).
- For sData (lpsData), reserve arrays for more than the amount specified for iSize.

(5) How to specify devices

The following describes how to specify label names and device values to be read.

- The following data type can be specified for label name.

Type class		Label data type	Label name format
			Label name
Array	Member	Bit, Word, Double Word, Float (Single Precision), Float (Double Precision), String[*1], Time, Timer, Counter, Retentive Timer	Label name [number of elements] Label name [n1] [n2] [n3]

*1 : The maximum number of characters that can be specified is 32 + NULL

- Set the following values for the number of read points according to the label data type.

Label (array) data type	Number of applicable words	Number of read points to be specified
Bit	1	Number of label array elements divided by 16 (rounded up)
Word	1	Number of label array elements
Double Word, Float (Single Precision)	2	Number of label array elements multiplied by 2
Float (Double Precision)	4	Number of label array elements multiplied by 4
String	17	Number of label array elements multiplied by 17
Time	2	Number of label array elements multiplied by 2
Timer, Counter, Retentive Timer	1	Number of label array elements

- The read device values are stored as follows.

<When bit device is specified>

Example: Read 3 points of data (3 words = 48 bits) from the devices starting from M0.

- Label setting (Data type: Bit, Number of array elements: 48)

Data type	Device
Bit (0..47)	M0

- Number of read points: 3
- Read device values

2 Bytes
M0 to M15[*2]
M16 to M31[*2]
M32 to M47[*2]

*2 : Devices are stored from the lower bit in the order of device number.

<When word device is specified>

Example: Read 3 points of data from the devices starting from D0.

- Label setting (Data type: Word, Number of array elements: 3)

Data type	Device
Word (0..1)	D0

- Number of read points: 3
- Read device values

2 Bytes
D0
D1
D2

<When FXCPU devices of CN200 and later are specified>

Example: Read 6 points of data from the devices starting from CN200.[*3]

- Label setting (Data type: Double Word, Number of array elements: 3)

Data type	Device
Double Word (0..2)	CN200

- Number of read points: 6
- Read device values

2 Bytes
L of CN200
H of CN200
L of CN201
H of CN201
L of CN202
H of CN202

*3 : For FXCPU devices of CN200 and later, 2 points of data are read from each 1 point of device (upper (H) and lower (L) data).
Reading only 1 point of data will result in an error.

<When FD device is specified (4-word device)>

Example: Read 8 points of data from the devices starting from FD0.

- Label setting (Data type: Double Word, Number of array elements: 2)

Data type	Device
Double Word (0..2)	FD0

- Number of read points: 8
- Read device values

2 Bytes
LL of FD0
LH of FD0
HL of FD0
HH of FD0
LL of FD1
LH of FD1
HL of FD1
HH of FD1

<When label is Double Word array and word device is specified>

Example: Read 6 points of data from the devices starting from D100.

- Label setting (Data type: Double Word, Number of array elements: 3)

Data type	Device
Double Word (0..2)	D100

- Number of read points: 6
- Read device values

2 Bytes
D100
D101
D102
D103
D104
D105

(6) Returned value

Normal termination : 0 is returned.

Abnormal termination: Any value other than 0 is returned. (☞ Page 378, CHAPTER 7 ERROR CODES)

Point

- The maximum number of read points should be the value which satisfies the following condition.
  Read start device number + Number of read points < Last device number
- When specifying bit type array labels, 0 or a multiple of 16 can be specified as a device number.
- For the number of read points, specify the number of words which applies to the data type specified for the label name.
  For the read device values, reserve a memory area for the number of points specified for the number of read points. If the memory area is not reserved, a critical error (an application error or the like) may occur.
- When a device which corresponds to the label name does not exist, an error occurs and data cannot be read.
  When any one of devices which correspond to multiple label names does not exist, an error occurs and data cannot be read.
- Digit specified bit device and index setting cannot be used.

5.3.19 WriteDeviceBlock2 (Writing devices in bulk)

(1) Applicable control

Applicable to the DotUtlType control.

(2) Feature

Write devices in 2-byte data unit in bulk.

(3) Format

(a) Visual Basic® .NET

IRet = object.WriteDeviceBlock2(szLabel, iSize, sData)

Integer	IRet	Returned value	Output
String	szLabel	Label name	Input
Integer	iSize	Number of write points	Input
Short	sData(n)	Device values to be written	Input

(b) Visual C++® .NET

iRet = object.WriteDeviceBlock2(*szLabel, iSize, *ipsData)

int	iRet	Returned value	Output
String*	szLabel	Label name	Input
int	iSize	Number of write points	Input
short*	ipsData	Device values to be written	Input

(4) Description

- The devices for the amount specified for iSize (number of write points) are written in bulk starting from the device specified for szLabel (label name).
- Store the device values to be written in sData (ipsData).
- For sData (ipsData), reserve arrays for more than the amount specified for iSize.

(5) How to specify devices

The following describes how to specify label names and device values to be written.

- The following data type can be specified for label name.

Type class		Label data type	Label name format
Array			Label name
	Member	Bit, Word, Double Word, Float (Single Precision), Float (Double Precision), String[*1], Time, Timer, Counter, Retentive Timer	Label name [number of elements] Label name [n1] [n2] [n3]

[*1] : The maximum number of characters that can be specified is 32 + NULL

- Set the following values for the number of write points according to the label data type.

Label (array) data type	Number of applicable words	Number of write points to be specified
Bit	1	Number of label array elements divided by 16 (rounded up)
Word	1	Number of label array elements
Double Word, Float (Single Precision)	2	Number of label array elements multiplied by 2
Float (Double Precision)	4	Number of label array elements multiplied by 4
String	17	Number of label array elements multiplied by 17
Time	2	Number of label array elements multiplied by 2
Timer, Counter, Retentive Timer	1	Number of label array elements

- Set the device values to be written as follows.

<When bit device is specified>

Example: Write 3 points of data (3 words = 48 bits) to the devices starting from M0.

- Label setting (Data type: Bit, Number of array elements: 48)

Data type	Device
Bit (0..47)	M0

- Number of write points: 3
- Device values to be written

2 Bytes
M0 to M15[*2]
M16 to M31[*2]
M32 to M47[*2]

[*2] : Devices are stored from the lower bit in the order of device number.

<When FXCPU devices of CN200 and later are specified>

Example: Write 6 points of data to the devices starting from CN200.[*3]

- Label setting (Data type: Double Word, Number of array elements: 3)

Data type	Device
Double Word (0..2)	CN200

- Number of write points: 6
- Device values to be written

2 Bytes
L of CN200
H of CN200
L of CN201
H of CN201
L of CN202
H of CN202

[*3] : For FXCPU devices of CN200 and later, 2 points of data are written to each 1 point of device (upper (H) and lower (L) data).
Writing only 1 point of data will result in an error.

<When word device is specified>

Example: Write 3 points of data to the devices starting from D0.

- Label setting (Data type: Word, Number of array elements: 3)

Data type	Device
Word (0..2)	D0

- Number of write points: 3
- Device values to be written

2 Bytes
D0
D1
D2

<When FD device is specified (4-word device)>

Example: Write 8 points of data to the devices starting from FD0.

- Label setting (Data type: Double Word, Number of array elements: 2)

Data type	Device
Double Word (0..1)	FD0

- Number of write points: 8
- Device values to be written

2 Bytes
LL of FD0
LH of FD0
HL of FD0
HH of FD0
LL of FD1
LH of FD1
HL of FD1
HH of FD1

<When label is Double Word array and word device is specified>

Example: Write 6 points of data to the devices starting from D100.

- Label setting (Data type: Double Word, Number of array elements: 3)

Data type	Device
Double Word (0..2)	D100

- Number of write points: 6
- Device values to be written

2 Bytes
D100
D101
D102
D103
D104
D105

(6) Returned value

Normal termination : 0 is returned.

Abnormal termination: Any value other than 0 is returned. (☞ Page 378, CHAPTER 7 ERROR CODES)

Point

- ● The maximum number of write points should be the value which satisfies the following condition.
  Write starting device number + Number of write points ≦ Last device number
- ● When specifying bit type array labels, 0 or a multiple of 16 can be specified as a device number.
- ● For the number of write points, specify the number of words which applies to the data type specified for the label name. For the device values to be written, reserve a memory area for the number of points specified for the number of write points. If the memory area is not reserved, a critical error (an application error or the like) may occur.
- ● When a device which corresponds to the label name does not exist, an error occurs and data cannot be written. When any one of devices which correspond to multiple label names does not exist, an error occurs and data cannot be written.
- ● Digit specified bit device and index setting cannot be used.

5.3.20 ReadDeviceRandom2 (Reading devices randomly)

(1) Applicable control

Applicable to the DotUtlType control.

(2) Feature

Read devices in 2-byte data unit randomly.

(3) Format

(a) Visual Basic® .NET

IRet = object.ReadDeviceRandom2(szLabel, iSize, sData(0))

Integer	IRet	Returned value	Output
String	szLabel	Label name	Input
Integer	iSize	Number of read points	Input
Short	sData(n)	Read device values	Output

IRet = object.ReadDeviceRandom2(szLabelList, iSize, sData(0))

Integer	IRet	Returned value	Output
String	szLabelList(n)	Label list	Input
Integer	iSize	Number of read points	Input
Short	sData(n)	Read device values	Output

(b) Visual C++® .NET

iRet = object.ReadDeviceRandom2(*szLabel, iSize, *arrsData)

Int	iRet	Returned value	Output
String*	szLabel	Label name	Input
int	iSize	Number of read points	Input
short*	arrsData	Read device values	Output

iRet = object.ReadDeviceRandom2(**szLabelList, iSize, *arrsData)

Int	iRet	Returned value	Output
String**	szLabelList	Label list	Input
int	iSize	Number of read points	Input
short*	arrsData	Read device values	Output

(4) Description

- Data of a device group for the size of iSize specified for the label name szLabel (szLabelList) are read.
- The read device values are stored in sData (arrsData).
- For sData (arrsData), reserve arrays for more than the amount specified for iSize.

(5) How to specify devices

The following describes how to specify label names and device values to be read.

- The following data type can be specified for label name.

Type class		Label data type	Label name format
Basic type		Bit, Word, Double Word, Float (Single Precision), Float (Double Precision), String[1], Time, Timer, Counter, Retentive Timer	Label name
Array			Label name
	Member	(The label data type can be specified in the same manner as the basic type.)	Label name [number of elements] Label name [n1] [n2] [n3]
Structure			Label name
	Member	(The label data type can be specified in the same manner as the basic type.)	Label name.Member name
Structured array			Label name [number of elements]
	Member	(The label data type can be specified in the same manner as the basic type.)	Label name [number of elements].Member name Label name [n1] [n2] [n3].Element

[1] The maximum number of characters that can be specified is 32 + NULL.

- For the number of read points, specify the sum of the following values correspond to the elements according to the label data type.

Label data type	Number of applicable words	Number of read points to be specified
Bit, Word	1	Number of label elements
Double Word, Float (Single Precision)	2	Number of label elements multiplied by 2
Float (Double Precision)	4	Number of label elements multiplied by 4
String	17	Number of label elements multiplied by 17
Time	2	Number of label elements multiplied by 2
Timer, Counter, Retentive Timer	1	Number of label elements

- The read device values are stored as follows.

<When bit device and word device are specified>

Example: Read data from each 1 point of M0 and D0.

- Label setting

szLabelList		Data type	Device
[0]	LABEL1	Bit	M0
[1]	LABEL2	Word	D0

- Number of read points: 2
- Read device values

2 Bytes	Applicable label
M0[*2]	LABEL1
D0	LABEL2

*2 : The device to be read is 1 point of "M0", and "0" or "1" is stored for the device value.

<When FXCPU devices of CN200 and later are specified>

Example: Read 3 points of data from the devices including CN200.

- Label setting

szLabelList		Data type	Device
[0]	LABEL1	Word	D0
[1]	LABEL2	Double Word	CN200
[2]	LABEL3	Word	D1

- Number of read points: 3
- Read device values

2 Bytes	Applicable label
D0	LABEL1
L of CN200[*3]	LABEL2
D1	LABEL3

*3 : Only lower 2 bytes are read. Data are not read from the specified device H (upper 2 bytes).

<When FD device is specified (4-word device)>

Example: Read 3 points of data from the devices including FD0.

- Label setting

szLabelList		Data type	Device
[0]	LABEL1	Word	D0
[1]	LABEL2	Double Word	FD0
[2]	LABEL3	Word	D1

- Number of read points: 3
- Read device values

2 Bytes	Applicable label
D0	LABEL1
LL of FD0[*4]	LABEL2
D1	LABEL3

*4 : Only lower 2 bytes are read. Data are not read from the specified devices HH, HL, and LH (upper 6 bytes).

<When data type equivalent to 2 words or more is specified for label>

Example: Write data by specifying labels of Double Word, Float (Single Precision), Float (Double Precision), String[*5], and Time types.

- Label setting

szLabelList		Data type	Device
[0]	LABEL1	Double Word	D0
[1]	LABEL2	Float (Single Precision)	D100
[2]	LABEL3	Float (Double Precision)	D200
[3]	LABEL4	String	D300
[4]	LABEL5	Time	D400

- Number of read points: 27
- Read device values

2 Bytes	Applicable label
D0	LABEL1
D1	
D100	LABEL2
D101	
D200	LABEL3
D201	
D202	
D203	
D300 : D316[*5]	LABEL4
D400	LABEL5
D401	

*5 : The number of points of characters to be read is 17 (32 characters + NULL). The characters need to be converted in a user program because the characters of String type are not converted.

<When array type label is specified>

Example: Read data from the devices by specifying array type labels.

- Label setting

szLabelList		Data type	Device
[0]	LABEL1	Bit (0..1)	M0
[1]	LABEL2	Double Word (0..1)	CN200
[2]	LABEL3	Double Word (0..1)	D0

- Number of read points: 8
- Read device values

2 Bytes	Applicable label
M0	LABEL1[0]
M1	LABEL1[1]
L of CN200	LABEL2[0]
L of CN201	LABEL2[1]
D100	LABEL3[0]
D101	
D102	LABEL3[1]
D103	

<When structure type labels are specified>

Example: Read data from the devices by specifying structure type labels.

- Structure setting

Structure name	Label name	Data type
STRUCT	L1	Bit
	L2	Double Word

- Label setting

szLabelList		Data type	Label name	Device
[0]	LABEL1	STRUCT	L1	D0.0
			L2	D0
[1]	LABEL2	STRUCT	L1	M10
			L2	CN200

- Number of read points: 6
- Read device values

2 Bytes	Applicable label
D0.0	LABEL1.L1
D0	LABEL1.L2
D1	
M0	LABEL2.L1
L of CN200	LABEL2.L2
L of CN201[*6]	

*6 : Data of two devices are read when the device of CN200 and later is specified for the Double Word type label.

<When labels with combined structure and label are specified>

Example: Read data from the devices by specifying structure type array and structure array type label.

- Structure setting

Structure name	Label name	Data type
STRUCT1	L1	Bit
	L2	Word
STRUCT2	L1	Bit (0..2)
	L2	Double Word

- Label setting

szLabelList		Data type	Label name	Device
[0]	LABEL1	STRUCT1 (0..1)	L1	X0
			L2	D0
[1]	LABEL2	STRUCT2	L1	M0
			L2	D100

- Number of read points: 9
- Read device values

2 Bytes	Applicable label
X0	LABEL1[0].L1
D0	LABEL1[0].L2
X1	LABEL1[1].L1
D1	LABEL1[1].L2
M0	LABEL2.L1[0]
M1	LABEL2.L1[1]
M2	LABEL2.L1[2]
D100	LABEL2.L2
D101	

(6) Returned value

Normal termination : 0 is returned.

Abnormal termination: Any value other than 0 is returned. (☞ Page 378, CHAPTER 7 ERROR CODES)

Point

- The maximum number of read points is 0x7FFFFFFF.
- For the number of read points, specify the number of words which applies to the data type specified for the label name. For the read device values, reserve a memory area for the number of points specified for the number of read points. If the memory area is not reserved, a critical error (an application error or the like) may occur.
- When a device which corresponds to the label name does not exist, an error occurs and data cannot be read. When any one of devices which correspond to multiple label names does not exist, an error occurs and data cannot be read.
- When a Double Word device is specified, only the data of the lower 1 word (2 bytes) are stored using this function. (An error does not occur.) When reading data from a Double Word device, use the ReadDeviceRandom function or the ReadDeviceBlock2 function.
- Digit specified bit device and index setting cannot be used.

5.3.21 WriteDeviceRandom2 (Writing devices randomly)

(1) Applicable control

Applicable to the DotUtlType control.

(2) Feature

Write devices in 2-byte data unit randomly.

(3) Format

(a) Visual Basic® .NET

Ret = object.WriteDeviceRandom2(szLabel, iSize, sData(0))

Integer	lRet	Returned value	Output
String	szLabel	Label name	Input
Integer	iSize	Number of write points	Input
Short	sData(n)	Device values to be written	Input

Ret = object.WriteDeviceRandom2(szLabelList, iSize, sData(0))

Integer	lRet	Returned value	Output
String	szLabelList(n)	Label list	Input
Integer	iSize	Number of write points	Input
Short	sData(n)	Device values to be written	Input

(b) Visual C++® .NET

iRet = object.WriteDeviceRandom2(*szLabel, iSize, *arrsData)

int	iRet	Returned value	Output
String*	szLabel	Label name	Input
int	iSize	Number of write points	Input
short*	arrsData	Device values to be written	Input

iRet = object.WriteDeviceRandom2(**szLabelList, iSize, *arrsData)

int	iRet	Returned value	Output
String**	szLabelList	Label list	Input
int	iSize	Number of write points	Input
short*	arrsData	Device values to be written	Input

(4) Description

- Data of a device group for the size of iSize specified for the label name szLabel are written.
- Store the device values to be written in sData (arrsData).
- For sData (arrsData), reserve arrays for more than the amount specified for iSize.

(5) How to specify devices

The following describes how to specify label names and device values to be written.

- The following data types can be specified for label names.

Type class		Label data type	Label name format
Basic type		Bit, Word, Double Word, Float (Single Precision), Float (Double Precision), String*1, Time, Timer, Counter, Retentive Timer	Label name
Array			Label name
	Member	(The label data type can be specified in the same manner as the basic type.)	Label name [number of elements] Label name [n1] [n2] [n3]
Structure			Label name
	Member	(The label data type can be specified in the same manner as the basic type.)	Label name.Member name
Structured array			Label name [number of elements]
	Member	(The label data type can be specified in the same manner as the basic type.)	Label name [number of elements].Member name Label name [n1] [n2] [n3].Element

*1 : The maximum number of characters that can be specified is 32 + NULL.

- For the number of write points, specify the sum of the following values correspond to the elements according to the label data type.

Label data type	Number of applicable words	Number of write points to be specified
Bit, Word	1	Number of label elements
Double Word, Float (Single Precision)	2	Number of label elements multiplied by 2
Float (Double Precision)	4	Number of label elements multiplied by 4
String	17	Number of label elements multiplied by 17
Time	2	Number of label elements multiplied by 2
Timer, Counter, Retentive Timer	1	Number of label elements

- Set the device values to be written as follows.

<When bit device and word device are specified>

Example: Write data to each 1 point of M0 and D0.

- Label setting

szLabelList		Data type	Device
[0]	LABEL1	Bit	M0
[1]	LABEL2	Word	D0

- Number of write points: 2
- Device values to be written

2 Bytes
M0*2
D0

*2 : The device to be written is 1 point of "M0", and the value of least significant bit of 2 byte-data to be set is the device value to be written.

<When FXCPU devices of CN200 and later are specified>

Example: Write 3 points of data to the devices including CN200.*3

- Label setting

szLabelList		Data type	Device
[0]	LABEL1	Word	D0
[1]	LABEL2	Double Word	CN200
[2]	LABEL3	Word	D1

- Number of write points: 3
- Device values to be written

2 Bytes	Applicable label
D0	LABEL1
L of CN200	LABEL2
D1	LABEL3

*3 : 4 bytes are written as 1 point for FXCPU devices of CN200 and later. Only lower 2 bytes can be set. "0" is written to the specified device H (upper 2 bytes).

<When FD device is specified (4-word device)>

Example: Write 3 points of data to the devices including FD0.

- Label setting

szLabelList		Data type	Device
[0]	LABEL1	Word	D0
[1]	LABEL2	Double Word	FD0
[2]	LABEL3	Word	D1

- Number of write points: 3
- Device values to be written

2 Bytes	Applicable label
D0	LABEL1
LL of FD0*4	LABEL2

*4 : Only lower 2 bytes can be set. "0" is written to the specified devices HH, HL, and LH (upper 6 bytes).

<When data type equivalent to 2 words or more is specified for label>

Example: Write data by specifying labels of Double Word, Float (Single Precision), Float (Double Precision), String*5, and Time types.

- Label setting

szLabelList		Data type	Device
[0]	LABEL1	Double Word	D0
[1]	LABEL2	Float (Single Precision)	D100
[2]	LABEL3	Float (Double Precision)	D200
[3]	LABEL4	String	D300
[4]	LABEL5	Time	D400

- Number of write points: 27
- Device values to be written

2 Bytes	Applicable label
D0	LABEL1
D1	
D100	LABEL2
D101	
D200	LABEL3
D201	
D202	
D203	
D300 : D316*5	LABEL4
D400	LABEL5
D401	

*5 : The number of points of characters to be written is 17 (32 characters + NULL). The characters need to be converted in a user program because the characters of String type are not converted.

<When array type label is specified>

Example: Write data to the devices by specifying array type labels.

- Label setting

szLabelList		Data type	Device
[0]	LABEL1	Bit (0..1)	M0
[1]	LABEL2	Double Word (0..1)	CN200
[2]	LABEL3	Double Word (0..1)	D0

- Number of write points: 8
- Device values to be written

2 Bytes	Applicable label
M0	LABEL1[0]
M1	LABEL1[1]
L of CN200	LABEL2[0]
L of CN201	LABEL2[1]
D100	LABEL3[0]
D101	
D102	LABEL3[1]
D103	

<When structure type labels are specified>

Example: Write data to the devices by specifying structure type labels.

- Structure setting

Structure name	Label name	Data type
STRUCT	L1	Bit
	L2	Double Word

- Label setting

szLabelList		Data type	Label name	Device
[0]	LABEL1	STRUCT	L1	D0.0
			L2	D0
[1]	LABEL2	STRUCT	L1	M10
			L2	CN200

- Number of write points: 5
- Device values to be written

2 Bytes	Applicable label
D0.0	LABEL1.L1
D0	LABEL1.L2
D1	
M0	LABEL2.L1
L of CN200	LABEL2.L2
X1	LABEL3.L1
M0 to 15	LABEL3.L2

<When labels with combined structure and label are specified>

Example: Write data to the devices by specifying structure type array and structure array type label.

- Structure setting

Structure name	Label name	Data type
STRUCT1	L1	Bit
	L2	Word
STRUCT2	L1	Bit (0..2)
	L2	Double Word

- Label setting

szLabelList		Data type	Label name	Device
[0]	LABEL1	STRUCT1 (0..1)	L1	X0
			L2	D0
[1]	LABEL2	STRUCT2	L1	M0
			L2	D100

- Number of write points: 9
- Device values to be written

2 Bytes	Applicable label
X0	LABEL1[0].L1
D0	LABEL1[0].L2
X1	LABEL1[1].L1
D1	LABEL1[1].L2
M0	LABEL2.L1[0]
M1	LABEL2.L1[1]
M2	LABEL2.L1[2]
D100	LABEL2.L2
D101	

(6) Returned value

Normal termination : 0 is returned.

Abnormal termination: Any value other than 0 is returned. (☞ Page 378, CHAPTER 7 ERROR CODES)

Point

- The maximum number of write points is 0x7FFFFFFF.
- For the number of write points, specify the number of words which applies to the data type specified for the label name. For the device values to be written, reserve a memory area for the number of points specified for the number of write points. If the memory area is not reserved, a critical error (an application error or the like) may occur.
- When a device which corresponds to the label name does not exist, an error occurs and data cannot be written. When any one of devices which correspond to multiple label names does not exist, an error occurs and data cannot be written.
- When a Double Word device is specified, this function writes data of lower 1 word (2 bytes), and "0" is written to data of upper 1 word (2 bytes). When writing Double Word devices, use the WriteDeviceRandom or the WriteDeviceBlock2 function.
- Digit specified bit device and index setting cannot be used.

5.3.22 SetDevice2 (Setting device data)

(1) Applicable control

Applicable to the DotUtlType control.

(2) Feature

Set one point of device in 2-byte data unit.

(3) Format

(a) Visual Basic® .NET

IRet = object.SetDevice2(szLabel, sData)

Integer	IRet	Returned value	Output
String	szLabel	Label name	Input
Short	sData	Device values to be written	Input

(b) Visual C++® .NET

iRet = object.SetDevice2(*szLabel, sData)

int	iRet	Returned value	Output
String*	szLabel	Label name	Input
short	sData	Device values to be written	Input

(4) Description

- The values of iData (device values to be written) are written to 1 point of device specified for szLabel (label name).
- When specifying bit devices, the least significant bit of the iData (device values to be written) is valid.

(5) How to specify devices

The following describes how to specify label names and device values to be written.

- The following data type can be specified for label name.

Type class	Label data type	Label name format
Basic type	Bit, Word, Double Word[*1], Float (Single Precision)[*1], Float (Double Precision)[*1], String[*1], Time[*1], Timer, Counter, Retentive Timer	Label name

*1 : Only lower 2 bytes of start device can be written.
When writing 2 words or more of devices, use the WriteDeviceBlock2 function.

- Set the device values to be written as follows.

<When bit device is specified>

Example: Write data to M0.

- Label setting

Data type	Device
Bit	M0

- Device values to be written

2 Bytes
M0[*2]

[*2] : The device to be written is 1 point of "M0", and the value of least significant bit of 2 byte-data to be set is the device value to be written.

<When CN200 or later of FXCPU is specified>

Example: Write data to CN200.[*3]

- Label setting

Data type	Device
Double Word	CN200

- Device values to be written

2 Bytes
L of CN200

[*3] : For FXCPU devices of CN200 and later, 4 bytes are written. Only lower 2 bytes can be set. "0" is written to the specified device H (upper 2 bytes).

<When data type equivalent to 2 words or more is specified for label>

Example: Write data by specifying String type label.

- Label setting

Data type	Device
String	D0

- Device values to be written

2 Bytes
D0[*5]

[*5] : Only lower 2 bytes are written to the start device.

<When word device is specified>

Example: Write data to D0.

- Label setting

Data type	Device
Word	D0

- Device values to be written

2 Bytes
D0

<When FD device is specified (4-word device)>

Example: Write data by specifying FD0.

- Label setting

Data type	Device
Word	FD0

- Device values to be written

2 Bytes
LL of FD0[*4]

[*4] : Only lower 2 bytes are written. Data cannot be written to the specified devices HH, HL, and LH (upper 6 bytes).

(6) Returned value

Normal termination : 0 is returned.

Abnormal termination: Any value other than 0 is returned. (☞ Page 378, CHAPTER 7 ERROR CODES)

Point

● When a device which corresponds to the label name does not exist, an error occurs and data cannot be written.

● When a Double Word device is specified, this function writes data of lower 1 word (2 bytes), and "0" is written to data of upper 1 word (2 bytes).
When writing Double Word devices, use the WriteDeviceRandom or the WriteDeviceBlock2 function.

● Digit specified bit device and index setting cannot be used.

5.3.23 GetDevice2 (Acquiring device data)

(1) Applicable control

Applicable to the DotUtlType control.

(2) Feature

Acquire one point of device in 2-byte data unit.

(3) Format

(a) Visual Basic® .NET

IRet = object.GetDevice2(szLabel, sData)

Integer	IRet	Returned value	Output
String	szLabel	Label name	Input
Short	sData	Read device values	Output

(b) Visual C++® .NET

iRet = object.GetDevice2(*szLabel, *lpsData)

int	iRet	Returned value	Output
String*	szLabel	Label name	Input
short*	lpsData	Read device values	Output

(4) Description

One point of device data specified for szLabel (label name) is stored in sData (lpsData) (read device values).

(5) How to specify devices

The following describes how to specify label names and device values to be read.

- The following data type can be specified for label name.

Type class	Label data type	Label name format
Basic type	Bit, Word, Double Word[*1], Float (Single Precision)[*1], Float (Double Precision)[*1], String[*1], Time[*1], Timer, Counter, Retentive Timer	Label name

*1 : Only lower 2 bytes of start device can be read.
When reading 2 words or more of devices, use the ReadDeviceBlock2 function.

- The read device values are stored as follows.

<When bit device is specified>

Example: Read data from M0.

- Label setting

Data type	Device
Bit	M0

- Read device values

2 Bytes
M0[*2]

*2 : The device to be read is 1 point of "M0", and "0" or "1" is stored for the device value.

<When CN200 or later of FXCPU is specified>

Example: Read data from CN200.[*3]

- Label setting

Data type	Device
Double Word	CN200

- Read device values

2 Bytes
L of CN200

*3 : The specified device H (lower 2 bytes) is read as 1 point for FXCPU devices of CN200 and later.
Data are not read from the specified device H (upper 2 bytes).

<When data type equivalent to 2 words or more is specified for label>

Example: Read data by specifying String type label.

- Label setting

Data type	Device
String	D0

- Read device values

2 Bytes
D0[*5]

*5 : Only lower 2 bytes are read to the start device.

<When word device is specified>

Example: Read data from D0.

- Label setting

Data type	Device
Word	D0

- Read device values

2 Bytes
D0

<When FD device is specified (4-word device)>

Example: Read data by specifying FD0.

- Label setting

Data type	Device
Word	FD0

- Read device values

2 Bytes
LL of FD0[*4]

*4 : Only lower 2 bytes are read. Data are not read from the specified devices HH, HL, and LH (upper 6 bytes).

(6) Returned value

Normal termination : 0 is returned.

Abnormal termination: Any value other than 0 is returned. (☞ Page 378, CHAPTER 7 ERROR CODES)

Point

- ● When a device which corresponds to the label name does not exist, an error occurs and data cannot be read.
- ● For the read device values, reserve 2 bytes of memory area. If the memory area is not reserved, a critical error (an application error or the like) may occur.
- ● When a Double Word device is specified, only the data of the lower 1 word (2 bytes) are stored using this function. (An error does not occur.) When reading data from a Double Word device, use the ReadDeviceRandom function or the ReadDeviceBlock2 function.
- ● Digit specified bit device and index setting cannot be used.

5.3.24 Connect (Connecting telephone line)

(1) Applicable control

Applicable to the DotUtlType control.

(2) Feature

Connect the telephone line.

(3) Format

(a) Visual Basic® .NET

IRet = object.Connect()

Integer	IRet	Returned value	Output

(b) Visual C++® .NET

iRet = object.Connect()

int	iRet	Returned value	Output

(4) Description

- The telephone line is connected according to the property settings of the modem communication control.
- When routing a serial communication module, the telephone line is connected in the connection system set in the ActConnectWay property.
  When auto line connect (callback number specification), callback connect (number specification), or callback request (number specification) is set in the ActConnectWay property, an error occurs if a number is not set in the ActCallbackNumber property.

(5) Returned value

Normal termination : 0 is returned.
Abnormal termination: Any value other than 0 is returned. (☞ Page 378, CHAPTER 7 ERROR CODES)

(6) Considerations for executing the Connect function

- Always connect the telephone line before the execution of the Open function.
- When disconnecting the telephone line, execute the Disconnect function.
  During the execution of the Connect function, the telephone line remains connected even when the Open and Close functions are executed repeatedly.
- If the telephone line is disconnected in the Open status for some reason, always execute the Close function before reconnecting the telephone line.

(7) Considerations when using multiple telephone line connection objects

- When control types, port numbers, and telephone numbers of controls are different
  When control types, port numbers, and telephone numbers set to multiple controls are different, an error (error code: 0xF1000016) occurs if the Connect function is executed to connect the control whose port number and telephone number are different from those of the control which executed the Connect function first.
- When port numbers and telephone numbers of controls are the same
  When control types, port numbers, and telephone numbers set to multiple controls are the same, the termination status are different according to the connection system of the callback function.
  The following table shows the relations between the connection system and the termination status of the callback function.

Connection system of control which executed the Connect function first	Connection system of control which executed the Connect function second or Later			
	Auto line connect Auto line connect(Callback fixation) Auto line connect (Callback number specification)	Callback connect (Fixation) Callback connect (Number specification)	Callback request (Fixation) Callback request (Number specification)	Callback reception waiting
Auto line connect Auto line connect (Callback fixation) Auto line connect (Callback number specification)	○	○	×	×
Callback connect (Fixation) Callback connect (Number specification)	○	○	×	×
Callback request (Fixation) Callback request (Number specification)	○	○	○	○
Callback reception waiting	×	×	×	○

○: Normal termination ×: Abnormal termination (Error occurrence)

5.3.25 Disconnect (Disconnecting telephone line)

(1) Applicable control

Applicable to the DotUtlType control.

(2) Feature

Disconnect the telephone line.

(3) Format

(a) Visual Basic® .NET

IRet = object.Disconnect()

Integer	IRet	Returned value	Output

(b) Visual C++® .NET

iRet = object.Disconnect()

int	iRet	Returned value	Output

(4) Description

The telephone line that was connected using the Connect function is disconnected.

(5) Returned value

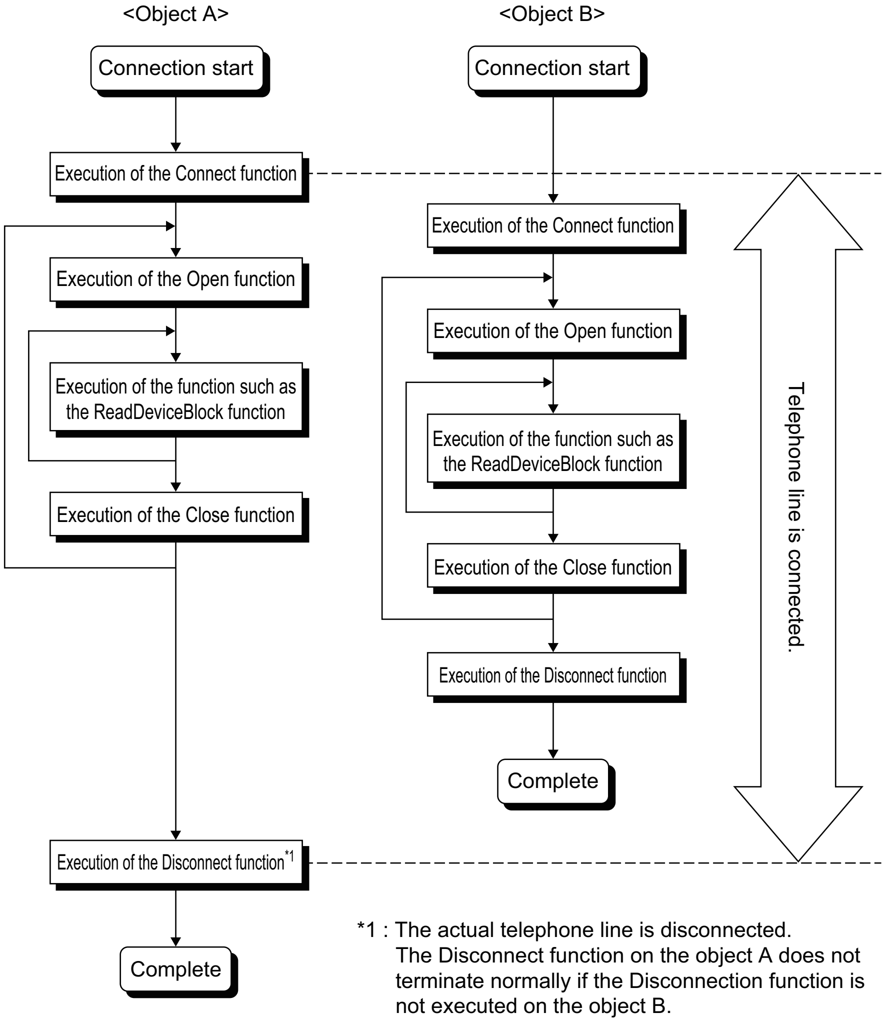
Normal termination : 0 is returned.

Abnormal termination: Any value other than 0 is returned. (☞ Page 378, CHAPTER 7 ERROR CODES)

(6) Considerations for executing the Disconnect function

- When executing the Disconnect function during the execution of the Open function, execute the Close function before executing the Disconnect function.
- If the telephone line is disconnected in the Open status for some reason, always execute the Close function before reconnecting the telephone line.
- When multiple telephone line connection objects are used, execute the Disconnect function on the object which executed the Connect function first after executing the Disconnect function on other objects.
  When multiple telephone line connection objects are used, the telephone line is not disconnected if the Disconnect function is not executed on the object that executed the Connect function first.
  The following figure shows the example of how to use multiple objects simultaneously.

Example: When using two controls simultaneously
(Including the case when the objects A and B are operated by different applications)



<Object A>
<Object B>
Connection start
Connection start
Execution of the Connect function
Execution of the Connect function
Execution of the Open function
Execution of the Open function
Execution of the function such as the ReadDeviceBlock function
Execution of the function such as the ReadDeviceBlock function
Execution of the Close function
Execution of the Close function
Execution of the Disconnect function
Complete
Telephone line is connected.
Execution of the Disconnect function*1
Complete

*1 : The actual telephone line is disconnected.
The Disconnect function on the object A does not terminate normally if the Disconnection function is not executed on the object B.

5.3.26 GetErrorMessage (Receiving error message)

(1) Applicable control

Applicable to the DotSupportMsg control.

(2) Feature

Receive the error description and its corrective action corresponding to the error code.

(3) Format

(a) Visual Basic® .NET

IRet = object.GetErrorMessage(lErrorCode, szErrorMessage)

Integer	IRet	Returned value	Output
Integer	lErrorCode	Error code	Input
String	szErrorMessage	Error message	Output

(b) Visual C++® .NET

iRet = object.GetErrorMessage(iErrorCode, **lpszErrorMessage)

int	iRet	Returned value	Output
int	iErrorCode	Error code	Input
String**	lpszErrorMessage	Error message	Output

(4) Description

- The error description and its corrective action of the error code specified for lErrorCode (iErrorCode) are read.
- The read error description and its corrective action are stored in szErrorMessage (lpszErrorMessage).

(5) Returned value

Normal termination : 0 is returned.

Abnormal termination: Any value other than 0 is returned. (☞ Page 378, CHAPTER 7 ERROR CODES)

CHAPTER 6 SAMPLE PROGRAMS

This chapter explains the sample programs registered at the installation of MX Component.

(1) Sample programs, test programs, and sample sequence programs

(a) Sample programs and test programs

Sample programs are for references when creating user programs.
Test programs are for conducting communication tests.
Use the programs with your responsibility.

(b) Sample sequence programs

Sample sequence programs require modifications according to the system configuration and parameter settings.
Modify the program to suit the system.
Use the programs with your responsibility.

(2) List of sample programs, test programs, and sample sequence programs

The following table shows the list of sample programs registered under [user specified folder] - [Act] - [Samples] when installing MX Component.

Folder name		Description	Supported language	Reference
AccessVBA	Sample	Sample program for ActUtlType	VBA (Access)	Page 356, Section 6.1.3
	TestPro	Test program for ActUtlType*1		—
ExcelVBA	Sample	Sample program for ActUtlType	VBA (Excel)	Page 352, Section 6.1.1
		Sample program for ActUtlType and ActSupportMsg (Reading/writing devices)		Page 354, Section 6.1.2
	TestPro	Test program for ActUtlType and ActSupportMsg*1		—
VBScript	SampleASP	Sample program for ActMLUtlType	HTML (ASP function)	Page 360, Section 6.3
	SampleHTML		HTML	Page 358, Section 6.2
	TestPro	Test program for ActUtlType and ActSupportMsg*1		—
Vb.NET	ModemSample	Modem communication sample program for ActUtlType and ActSupportMsg	Visual Basic®.NET	Page 363, Section 6.4.1
	Sample	Read/Write sample program for ActUtlType and ActProgType		Page 365, Section 6.4.2
	SampleDot	Read/Write sample program for DotUtlType		
	Sample_References	Read/Write sample program for ActUtlType and ActProgType		
	Sample_References Dot	Read/Write sample program for DotUtlType		

*1 : A test program for checking operations.

Folder name		Description	Supported language	Reference Section
Vb.NET	Sample_TypeConv	Type conversion sample program for ActUtlType and ActSupportMsg	Visual Basic®.NET	Page 368, Section 6.4.3
Vc.NET	Sample2005	Read/Write sample program for ActUtlType and ActProgType	Visual C++®.NET (Visual Studio® 2005)	Page 370, Section 6.5.1
	SampleDot2005	Read/Write sample program for DotUtlType		
	Sample2010	Read/Write sample program for ActUtlType and ActProgType	Visual C++®.NET (Visual Studio® 2010)	
	SampleDot2010	Read/Write sample program for DotUtlType		
	Sample_References2005	Read/Write sample program for ActUtlType and ActProgType	Visual C++®.NET (Visual Studio® 2005)	
	Sample_ReferencesDot2005	Read/Write sample program for DotUtlType		
	Sample_References2010	Read/Write sample program for ActUtlType and ActProgType	Visual C++®.NET (Visual Studio® 2010)	
	Sample_ReferencesDot2010	Read/Write sample program for DotUtlType		
	Sample_Support	For DotSupportMsg	Visual C++®.NET (Visual Studio® 2005)	Page 371, Section 6.5.2
Vc	SampleENG	Read/Write sample program for ActUtlType and ActProgType	Visual C++®.NET(MFC) (Visual Studio® 2005)	Page 373, Section 6.6.1
	CustomSampleENG	Read/Write sample program for ActUtlType and ActProgType		Page 375, Section 6.6.2
	Sample_SupportENG	For ActSupportMsg		Page 376, Section 6.6.3
GppW	CCG4A	Sample ladder for CC-Link G4 communication	Sequence program (GX Developer)	MX Component Version 4 Operating Manual.
	FXCPUtel	Sample ladder for modem communication (FXCPU)		
	QJ71C24Callback	Sample ladder for modem communication (Q series-compatible C24) in which the setting 1 is specified for the callback function		
	QJ71C24Callback_Number	Sample ladder for modem communication (Q series-compatible C24) in which the setting 3 is specified for the callback function		
	QJ71C24TEL	Sample ladder for modem communication (Q series-compatible C24)		

6.1 VBA Sample Programs

This section explains the VBA sample programs for Excel and Access.

6.1.1 Sample program for Excel

This sample program is a program to log and graph device values of a programmable controller CPU using the ActUtlType control.
This sample program was created on Excel 2003.

(1) How to use the sample program

1. Open the sample sheet.

2. Enter the logical station number, which was specified for the communication setting in Communication Setup Utility, into the "LogicalStationNumber" input cell.

3. Enter the start device of devices to be read into the "DeviceName" input cell.

4. Enter the logging interval into the "LoggingTiming" input cell.

5. Click the LoggingStart button to start logging.
When the logging starts, the past 10 device values and line graphs are displayed on the graph.

6. Click the LoggingStop button to stop logging.
Note that the logging data on the screen are not cleared.

7. If an error occurs during the process, an error message is displayed on the "Message" output cell and an error code is displayed on the "Return Code" output cell. When an error occurs, perform the troubleshooting corresponds to the error code.
(☞ Page 378, CHAPTER 7 ERROR CODES)

(2) Considerations for using the sample program

- Before executing the sample program, set the communication settings in Communication Setup Utility.
- When changing the input value, click the LoggingStop button to stop logging, change the input value, and click the LoggingStart button to start logging.
- In this sample program, the number of device points to be logged is 10, and the number of logging times is 10.

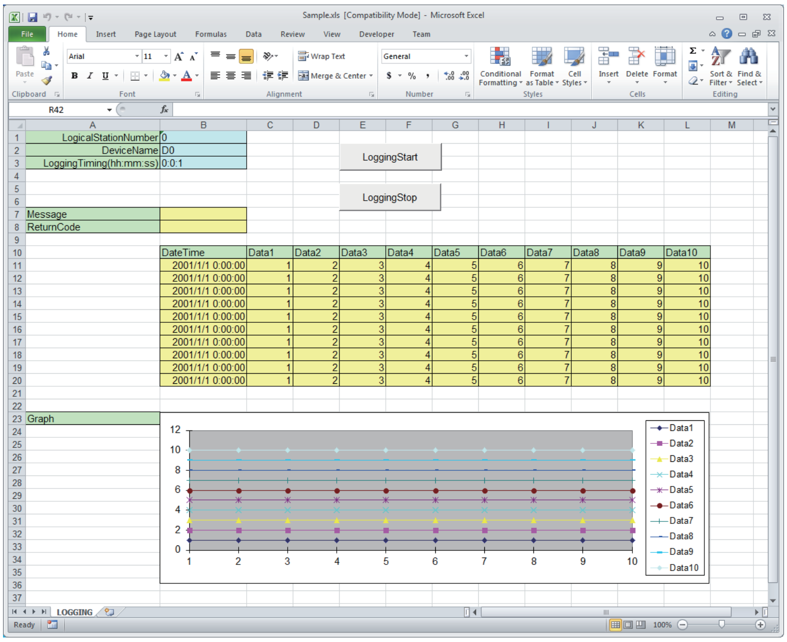
(3) Sample file

The sample program is installed in the following folder when the default pass is selected.

Folder name	File name
[User specified folder] - [Act] - [Samples] - [ExcelVBA] - [Sample]	Sample.xls

(4) Screen

The following explains the settings on the sample program screen.



LogicalStationNumber 0

DeviceName D0

LoggingTiming(hh:mm:ss) 0:0:1

Message

ReturnCode

DateTime	Data1	Data2	Data3	Data4	Data5	Data6	Data7	Data8	Data9	Data10
2001/1/1 0:00:00	1	2	3	4	5	6	7	8	9	10
2001/1/1 0:00:00	1	2	3	4	5	6	7	8	9	10
2001/1/1 0:00:00	1	2	3	4	5	6	7	8	9	10
2001/1/1 0:00:00	1	2	3	4	5	6	7	8	9	10
2001/1/1 0:00:00	1	2	3	4	5	6	7	8	9	10
2001/1/1 0:00:00	1	2	3	4	5	6	7	8	9	10
2001/1/1 0:00:00	1	2	3	4	5	6	7	8	9	10
2001/1/1 0:00:00	1	2	3	4	5	6	7	8	9	10
2001/1/1 0:00:00	1	2	3	4	5	6	7	8	9	10
2001/1/1 0:00:00	1	2	3	4	5	6	7	8	9	10

Graph

Item	Description
LogicalStationNumber	Enter the logical station number which was specified for the communication setting in Communication Setup Utility.
DeviceName	Enter the start device of devices to be read.
LoggingTiming (hh:mm:ss)	Enter the logging interval. Example: Logging at an interval of 1 second 0:0:1 Logging at an interval of 1 hour 30 minutes 1:30:0
Message	Display the function execution result. (Character string)
ReturnCode	Display the function execution result. (Hexadecimal value)
Time	Display the logged system time.
Data01 to 10	Display the logged device values.
Graph	Display the past 10 logged values of 10 devices in line graphs.
LoggingStart button	Start logging.
LoggingStop button	Stop logging.

6.1.2 Sample program for Excel (Reading/writing devices)

This sample program is a program to read/write the programmable controller CPU devices (D0 to D9) using the ActUtlType control.

This program displays the error code and error message on the dialog box using the ActSupportMsg control when an error occurs.

This sample program was created on Excel 2003.

(1) How to use the sample program

1. **Open the sample sheet.**
2. **Enter the logical station number, which was specified for the communication setting in Communication Setup Utility, into the "LogicalStationNumber" input cell.**
3. **Click the** DeviceRead **button to read the device values of D0 to D9 of the programmable controller CPU and display them on the DeviceRead area.**
4. **Enter values into D0 to D9 of the DeviceWrite area and click the** DeviceWrite **button to write the values to D0 to D9 of the programmable controller CPU.**
5. **If reading/writing data from/to the devices D0 to D9 fails, the error message corresponds to the error code is displayed on the dialog box.**

(2) Considerations for using the sample program

Before executing the sample program, set the communication settings in Communication Setup Utility.

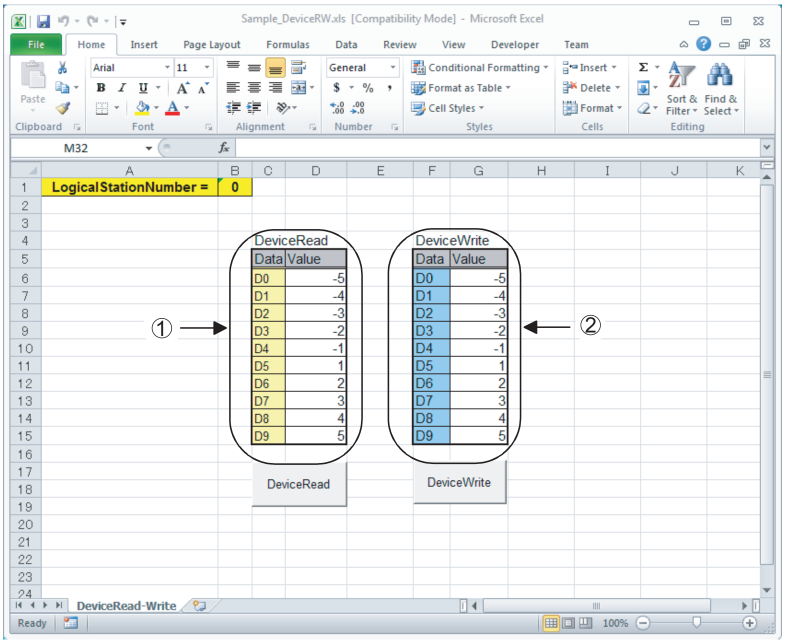
(3) Sample file

The sample program is installed in the following folder when the default pass is selected.

Folder name	File name
[User specified folder] - [Act] - [Samples] - [ExcelVBA] - [Sample]	Sample_DeviceRW.xls

(4) Screen

The following explains the settings on the sample program screen.



LogicalStationNumber = 0

DeviceRead

Data	Value
D0	-5
D1	-4
D2	-3
D3	-2
D4	-1
D5	1
D6	2
D7	3
D8	4
D9	5

DeviceWrite

Data	Value
D0	-5
D1	-4
D2	-3
D3	-2
D4	-1
D5	1
D6	2
D7	3
D8	4
D9	5

Item	Description
LogicalStationNumber	Enter the logical station number which was specified for the communication setting in Communication Setup Utility.
① (DeviceRead area)	Display the device values of D0 to D9 of the programmable controller CPU.
② (DeviceWrite area)	Enter the values to be written into D0 to D9 of the programmable controller CPU.
DeviceRead button	Read the device values of D0 to D9 of the programmable controller CPU and display them on the DeviceRead area.
DeviceWrite button	Write the device values entered into D0 to D9 of the DeviceWrite area to D0 to D9 of the programmable controller CPU.

6.1.3 Sample program for Access

This sample program is a program to log and monitor device values of a programmable controller CPU using the ActUtlType control.
This sample program was created on Access 2003.

(1) How to use the sample program

1. **Open the database.**
2. **Enter the logical station number, which was specified for the communication setting in Communication Setup Utility, into the "LogicalStationNumber" text box.**
3. **Enter the logging interval into the "LoggingTiming" text box.**
4. **Click the** LoggingStart **button to start logging.**
5. **Click the** LoggingStop **button to stop logging.**
   Note that the logging data on the screen are not cleared.
6. **If an error occurs during the process, an error message and error code are displayed on the message box.When an error occurs, perform the troubleshooting corresponds to the error code. (☞ Page 378, CHAPTER 7 ERROR CODES)**

(2) Considerations for using the sample program

- Before executing the sample program, set the communication settings in Communication Setup Utility.
- When changing the input value, click the LoggingStop button to stop logging, change the input value, and click the LoggingStart button to start logging.
- In this sample program, devices "D0" to "D4" are set as the devices to be monitored, and "D10" to "D17" are set as the devices to be logged.
  The monitoring interval is 1 second.
- This sample program logs device values up to 100 times.
  If the number of logging times exceeds 100, the oldest logging data is deleted and the newest logging data is registered.

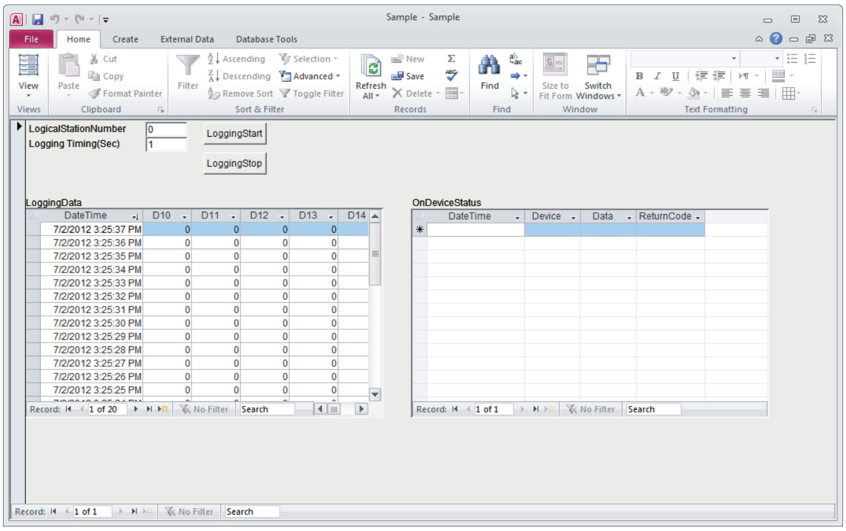
(3) Sample file

The sample program is installed in the following folder when the default pass is selected.

Folder name	File name
[User specified folder] - [Act] - [Samples] - [AccessVBA] - [Sample]	Sample.mdb

(4) Screen

The following explains the settings on the sample program screen.



LogicalStationNumber 0
Logging Timing(Sec) 1

LoggingData

DateTime	D10	D11	D12	D13	D14
7/2/2012 3:25:37 PM	0	0	0	0	
7/2/2012 3:25:36 PM	0	0	0	0	
7/2/2012 3:25:35 PM	0	0	0	0	
7/2/2012 3:25:34 PM	0	0	0	0	
7/2/2012 3:25:33 PM	0	0	0	0	
7/2/2012 3:25:32 PM	0	0	0	0	
7/2/2012 3:25:31 PM	0	0	0	0	
7/2/2012 3:25:30 PM	0	0	0	0	
7/2/2012 3:25:29 PM	0	0	0	0	
7/2/2012 3:25:28 PM	0	0	0	0	
7/2/2012 3:25:27 PM	0	0	0	0	
7/2/2012 3:25:26 PM	0	0	0	0	
7/2/2012 3:25:25 PM	0	0	0	0	

Record: 1 of 20

OnDeviceStatus

DateTime	Device	Data	ReturnCode

Record: 1 of 1

Record: 1 of 1

Item	Description
LogicalStationNumber	Enter the logical station number which was specified for the communication setting in Communication Setup Utility.
LoggingTiming	Enter the logging interval. (In seconds)
LoggingData	Display the logged data.
OnDeviceStatus	Display the devices whose conditions are satisfied among the devices being monitored.
LoggingStart button	Start logging.
LoggingStop button	Stop logging.

6.2 VBScript Sample Program

This section explains the VBScript sample program.
This sample program is a program to define device values of a programmable controller CPU as a capacity and status of a tank, and monitor their values using the ActUtlType control.

This sample program was created on Microsoft® FrontPage® 2000.

(1) How to use the sample program

1. **Open the sample file to open the communication line to the programmable controller CPU.**
2. **The device values of the programmable controller CPU are acquired at an interval of 1 second and they are used to display the capacity and status of the tank.**
3. **If an error occurs during the process, an error message and error code are displayed on the message box. When an error occurs, perform the troubleshooting corresponds to the error code. (☞ Page 378, CHAPTER 7 ERROR CODES)**

(2) Considerations for using the sample program

- Before executing the sample program, set the logical station number to "0" for the communication setting in Communication Setup Utility.
- In this sample program, the device "D100" is used for the tank capacity, and the device "D101" is used for the tank status.

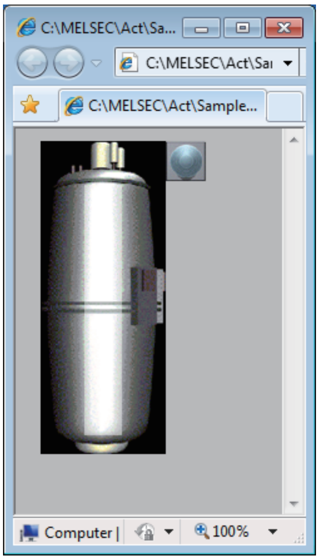
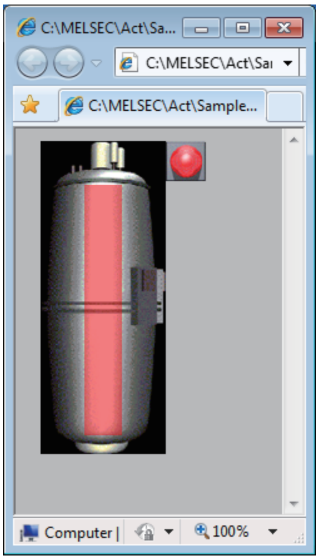
(3) Sample file

The sample program is installed in the following folder when the default pass is selected.

Folder name	File name
[User specified folder] - [Act] - [Samples] - [VBScript] - [SampleHTML]	Sample.html

(4) Screen

The following explains the settings on the sample program screen.

Tank capacity: Empty Tank status: Normal	Tank capacity: Full Tank status: Error
	

Item	Description	Remarks
Tank capacity	Indicate the tank capacity. (Within the range of 0 ≦ device value ≦ 200)	Linked with the device "D100".
Tank status	Indicate the tank status. Normal (device value = 0): Blue lamp ON Error (device value <> 0): Red lamp ON	Linked with the device "D101".

6.3 ASP Sample Programs

This sample program is a program to read programmable controller CPU data and device values using the ActUtlType control.

(1) File structure

- Sample.asp (data input screen)
  The initial display screen used to set data for monitoring.
- SampleControl.asp (data acquisition screen)
  Acquire input data of Sample.asp, store them in global variables, and check them for errors.
- SampleMon.asp (data display screen)
  Display data according to input data of Sample.asp. Display an error description when an error occurs.

(2) How to use the sample program

1. **Store Sample.asp, SampleControl.asp, and SampleMon.asp in the same folder on the WWW server.**
2. **Browse the URL of Sample.asp on the Microsoft® Internet Explorer and display the initial screen.**
3. **Enter data into "MonitorTiming", "LogicalStationNumber", "DeviceName", and "DeviceSize" on the initial screen. Press the MonitorStart button to start the Open processing, ReadDeviceBlock processing, GetCpuType processing, and Close processing.**
4. **The input data for "MonitorTiming:", "LogicalStationNumber:", "DeviceName:", and "DeviceSize:" are displayed on the data display screen.**
5. **An error occurrence processing is displayed for "Message:" when an error occurs in the process.**
6. **The result of the process is displayed for "Return Code:".**
7. **The model of the connected programmable controller CPU is displayed for "CpuType:".**
8. **The device data of the connected programmable controller CPU is displayed for "ReadData(Hex)".**
9. **When an error occurs, perform the troubleshooting corresponds to the error code. (☞ Page 378, CHAPTER 7 ERROR CODES)**
10. **The data display screen repeats updating at the interval set for "MonitorTiming:".**
11. **Press the Back button to return to the initial screen.**

(3) Considerations for using the sample program

- Before executing the sample program, set the communication settings in Communication Setup Utility.

(4) Sample file list

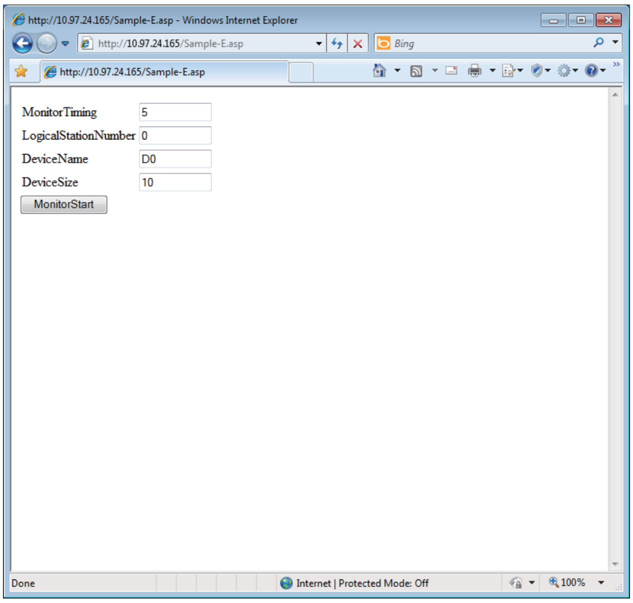
The sample program is installed in the following folder when the default pass is selected.

Folder name	File name
[User specified folder] - [Act] - [Samples] - [VBScript] - [SampleASP]	Sample-E.asp
	SampleControl-E.asp
	SampleMon-E.asp

(5) Screen

The following explains the settings on the sample program screen.

(a) Data input screen (Sample-E.asp)


http://10.97.24.165/Sample-E.asp - Windows Internet Explorer
http://10.97.24.165/Sample-E.asp
Bing
http://10.97.24.165/Sample-E.asp
MonitorStart
Done
Internet | Protected Mode: Off
100%

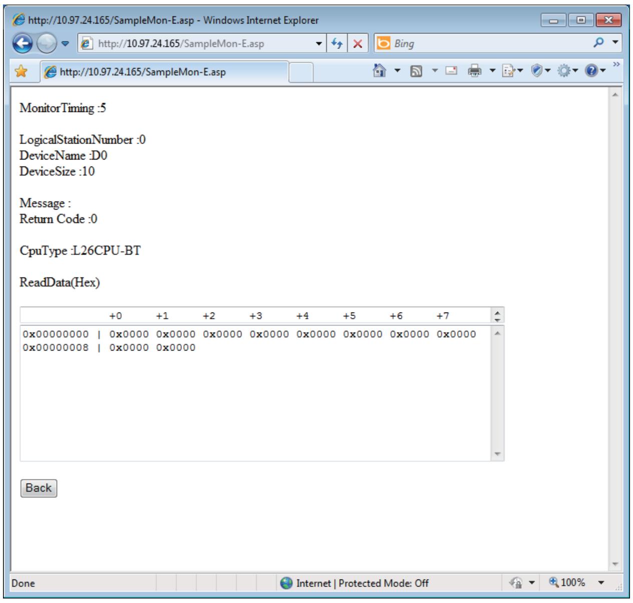
MonitorTiming 5
LogicalStationNumber 0
DeviceName D0
DeviceSize 10

Item	Description
MonitorTiming	Enter the monitoring interval.
LogicalStationNumber	Enter the logical station number.
DeviceName	Enter the device name of the device to be read.
DeviceSize	Enter the number of points of the devices to be read.
MonitorStart button	Start the monitor processing.

(b) Data acquisition screen (SampleControl-E.asp)

The data acquisition screen is a screen used to store the data entered on the data input screen into the global variables and check them for errors. This screen is not displayed on the Internet Explorer screen.

(c) Data display screen (SampleMon-E.asp)


http://10.97.24.165/SampleMon-E.asp - Windows Internet Explorer
http://10.97.24.165/SampleMon-E.asp
Bing
http://10.97.24.165/SampleMon-E.asp
Done
Internet | Protected Mode: Off
100%

MonitorTiming :5

LogicalStationNumber :0
DeviceName :D0
DeviceSize :10

Message :
Return Code :0

CpuType :L26CPU-BT

ReadData(Hex)

```
                 +0     +1     +2     +3     +4     +5     +6     +7
0x00000000 | 0x0000 0x0000 0x0000 0x0000 0x0000 0x0000 0x0000 0x0000
0x00000008 | 0x0000 0x0000
```

Item	Description
MonitorTiming:	Display the monitoring interval.
LogicalStationNumber:	Display the logical station number.
DeviceName:	Display the device name of the device to be read.
DeviceSize:	Display the number of points of the devices to be read.
Message:	Display an error description at an error occurrence.
Return Code:	Display the method execution result.
CpuType:	Display the CPU model.
ReadData(Hex)	Display the read device values.
Back button	Return to the initial screen.

6.4 Visual Basic® .NET Sample Programs

6.4.1 Modem communication sample program

This sample program is a program to monitor devices of a programmable controller CPU corresponds to the specified logical station number using the ActUtlType control.
This program displays the error code and error message on the dialog box using the ActSupportMsg control when an error occurs.

This sample program was created on Visual Basic® .NET (Visual Studio® 2005).

(1) How to use the sample program

1. **Enter the logical station number, which was specified for the modem communication setting in Communication Setup Utility, into the "LogicalStationNumber" text box.**

2. **If a password is set to the module to be connected, enter the password into the "Password" text box.**

3. **Enter the monitoring interval into the "MonitorInterval" text box.**

4. **Enter the device name to be monitored into the "DeviceName" text box, and the number of device points into the "Size" text box.**

5. **Click the** MonitorStart **button to open the communication line after connecting the telephone line, and read device values at the interval specified using the RedDeviceBlock function. (For the logical station number that does not require the telephone line connection, the telephone line is not connected and only the Open function is executed on the communication line.)**

6. **Click the** MonitorStop **button to disconnect the telephone line after closing the communication line.**

7. **If an error occurs during the process of this sample program, an error code is displayed on the "ReturnValue" text box. If the error code is displayed on the "ReturnValue" text box, click the** GetErrorMessage **button to display the error description and corrective action corresponds to the displayed error code.**
When an error occurs, perform the troubleshooting corresponds to the error code.
(☞ Page 378, CHAPTER 7 ERROR CODES)

(2) Considerations for using the sample program

- Since the ActUtlType control is used, set the logical station number in Communication Setup Utility before executing the sample program.
- When changing the logical station number, monitoring interval, device name, and number of read points, click the MonitorStop button to close the communication line, and click the MonitorStart button to resume the communication.

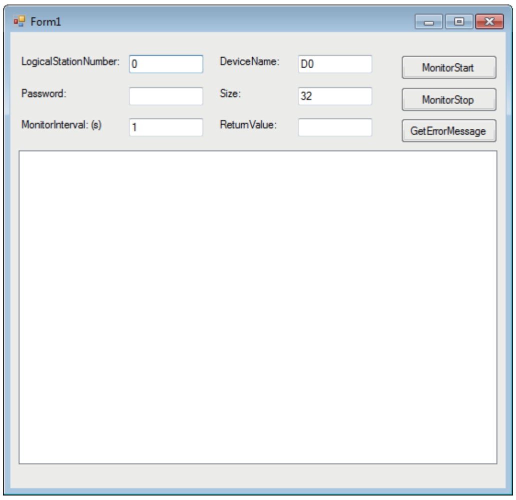
(3) Sample file

The sample program is installed in the following folder when the default pass is selected.

Folder name	File name
[User specified folder] - [Act] - [Samples] - [Vb.NET] - [ModemSample]	ModemSample.sln

(4) Screen

The following explains the settings on the sample program screen.



LogicalStationNumber: 0

DeviceName: D0

Password:

Size: 32

MonitorInterval: (s) 1

ReturnValue:

Item	Description
LogicalStationNumber	Enter the logical station number set in Communication Setup Utility.
Password	Enter the password when it is required.
MonitorInterval	Set the monitoring interval. (In seconds)
DeviceName	Enter the device name to be monitored.
Size	Enter the number of read points.
ReturnValue	Display the returned value of the executed method.
MonitorStart button	Open the communication line and start monitoring after connecting the telephone line.
MonitorStop button	Disconnect the telephone line and stop monitoring after closing the communication line.
GetErrorMessage button	Acquire and display the error description and corrective action for the error code of "ReturnValue".

6.4.2 Read/Write sample program

This sample program is a program to read/write device values of a programmable controller CPU and monitor device status using any of the ActUtlType control, the DotUtlType control, or the ActProgType control.

This sample program was created on Visual Basic® .NET (Visual Studio® 2005).

(1) How to use the sample program

1. **Load the form and select the control to be used.**
2. **When using the ActUtlType control, enter the logical number, which was specified for the communication setting in Communication Setup Utility, into the "LogicalStationNumber" text box before clicking the** Open **button.**
3. **Click the** Open **button to open the communication line.**
4. **For reading devices randomly, enter the device name and the number of points to be read into the "DeviceName" and "DeviceSize" text boxes in the "Random Read/Write" frame, and click the** ReadDeviceRandom2 **button to display the device data on the "Data" text box in the "Output" frame.**
5. **For writing devices randomly, enter the device name and the number of points to be written in the "DeviceName" and "DeviceSize" text boxes in the "Random Read/Write" frame, enter the device value to be written into the "DeviceData" text box inside the same frame, and click the** WriteDeviceRandom2 **button to write the device value to the programmable controller CPU.**
6. **For reading devices in bulk, enter the device name and the number of points to be read into the "DeviceName" and "DeviceSize" text boxes in the "Block Read/Write" frame, and click the** ReadDeviceBlock2 **button to display the device data on the "Data" text box in the "Output" frame.**
7. **For writing devices in bulk, enter the device name and the number of points to be written into the "DeviceName" and "DeviceSize" text boxes in the "Block Read/Write" frame, enter the device value to be written into the "DeviceData" text box in the same frame, and click the** WriteDeviceBlock2 **button to write the device value to the programmable controller CPU.**
8. **For registering devices for status monitoring, enter the device name, number of points, device value and status monitoring interval into the "DeviceName", "DeviceSize", "DeviceData", and "MonitorCycle" text boxes in the "Status Entry/Free" frame as the event occurrence conditions, and click the** EntryDeviceStatus **button.**
   When the registered event occurrence conditions are satisfied, the event data are displayed on the "Data" text box in the "Output" frame.
9. **For deregistering registered devices for status monitoring, click the** FreeDeviceStatus **button.**
10. **If an error occurs during the process, an error code is displayed on the "Return Code" box in the "Output" frame.**
11. **When an error occurs, perform the troubleshooting corresponds to the error code. (☞ Page 378, CHAPTER 7 ERROR CODES)**
12. **Click the** Close **button to close the communication line.**

(2) Considerations for using the sample program

- When using the ActUtlType control, set the communication settings in Communication Setup Utility before executing the sample program.
- When using the ActProgType control in this sample program, "Q02(H)" is set for the programmable controller CPU, "COM1" is set for the COM port, and 19200bps is set for the transmission speed.
  The specification can be changed by changing values of the corresponding property.
- When changing the control to be used, close the communication line by clicking the Close button, change the control, and click the Open button to reopen the line.

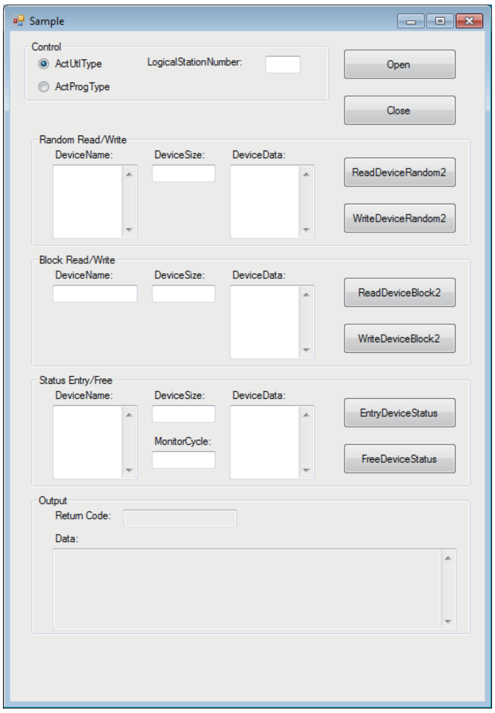
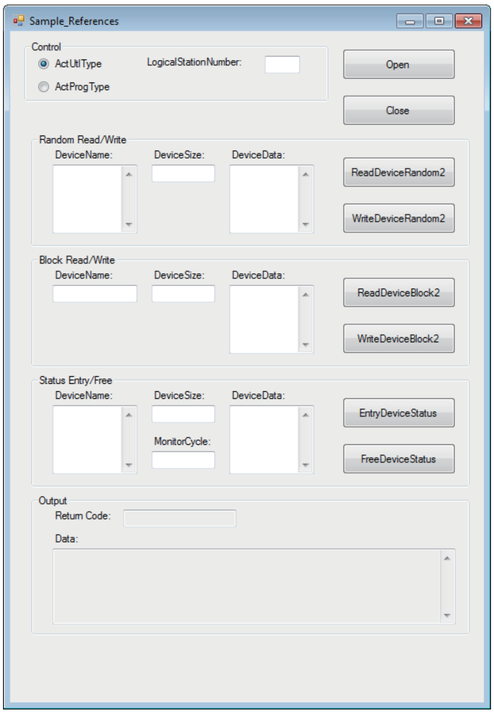
(3) Sample file list

The sample program is installed in the following folder when the default pass is selected.

Folder name	File name
[User specified folder] - [Act] - [Samples] - [Vb.NET] - [Sample]	Sample.sln
[User specified folder] - [Act] - [Samples] - [Vb.NET] - [Sample_References]	Sample_References.sln
[User specified folder] - [Act] - [Samples] - [Vb.NET] - [SampleDot]	Sample.sln
[User specified folder] - [Act] - [Samples] - [Vb.NET] - [Sample_ReferencesDot]	Sample_References.sln

(4) Screen

The following explains the settings on the sample program screen.

<Sample>	<Sample_References>
	

Item		Description
"Control" frame	Any of ActUtlType, ActProgType, or DotUtlType	Select the control to be used.
	LogicalStationNumber	Enter the logical station number which was specified for communication setting in Communication Setup Utility.
Open button		Open the communication line.
Close button		Close the communication line.
"Random Read/Write" frame	DeviceName	Enter the device name for reading/writing devices randomly.
	DeviceSize	Enter the number of device points for reading/writing devices randomly.
	DeviceData	Enter the device value for writing devices randomly.
	ReadDeviceRandom2 button	Read device data randomly from the programmable controller CPU to the data source entered into the "DeviceName" and "DeviceSize" text boxes in the same frame, and display the data on the "Data" text box in the "Output" frame.
	WriteDeviceRandom2 button	Write device data randomly to the programmable controller CPU from the data source entered into the "DeviceName", "DeviceSize", and "DeviceData" text boxes in the same frame.
"Block Read/Write" frame	DeviceName	Enter the device name for reading/writing devices in bulk.
	DeviceSize	Enter the number of device points for reading/writing devices in bulk.
	DeviceData	Enter the device values for writing devices in bulk.
	ReadDeviceBlock2 button	Read device data in bulk from the programmable controller CPU to the data source entered into the "DeviceName" and "DeviceSize" text boxes in the same frame, and display the data on the "Data" text box in the "Output" frame.
	WriteDeviceBlock2 button	Write device data in bulk to the programmable controller CPU from the data source entered into the "DeviceName", "DeviceSize", and "DeviceData" text boxes in the same frame.
"Status Entry/Free" frame	DeviceName	Enter the device name as an event occurrence condition.
	DeviceSize	Enter the number of device points as an event occurrence condition.
	MonitorCycle	Enter the event monitoring interval.
	DeviceData	Enter the device value as an event occurrence condition.
	EntryDeviceStatus button	Register the on-device status/event to the data source entered into the "DeviceName", "DeviceSize", "MonitorCycle", and "DeviceData" text boxes in the same frame. When the registered event occurrence condition is satisfied, the event data is displayed on the "Data" text box of the "Output" frame.
	FreeDeviceStatus button	Delete the registered on-device status/event.
"Output" frame	Return Code	Display the method execution result.
	Data	Display the read device values.

6.4.3 Type conversion sample program

This sample program is a program to read/write ASCII character string, 32-bit integer, or real number data from/to a programmable controller CPU devices in the corresponding data format using the ActUtlType control, and display the error message using the ActSupportMsg control when an error occurs during the process.

This sample program was created on Visual Basic® .NET (Visual Studio® 2005).

(1) How to use the sample program

1. **Load the form, enter the logical number, which was specified for the communication setting in Communication Setup Utility, into the "LogicalStationNumber" text box, and click the Open button to open the communication line.**
2. **For writing ASCII character data to the programmable controller CPU, enter the ASCII character data into the upper text box in the "ASCII character" frame, and click the Write button in the same frame.(The data write function is executed within the range of D0 to D9, regardless of the number of entered characters. When the number of characters is insufficient, the blanks are filled with 0 (Hex).)**
3. **For reading ASCII character data, click the Read button in the "ASCII character" frame. Data from the programmable controller CPU is displayed on the lower text box in the same frame. (The data read function is executed within the range of D0 to D9.)**
4. **For writing 32-bit integer data to the programmable controller CPU, enter the 32-bit integer data into the upper text box in the "32bit integer" frame, and click the Write button in the same frame. (The data write function is executed within the range of D10 to D11.)**
5. **For reading 32-bit integer data, click the Read button in the "32bit integer" frame. Data from the programmable controller CPU is displayed on the lower text box in the same frame. (The data read function is executed within the range of D10 to D11.)**
6. **For writing real number data to the programmable controller CPU, enter the real number data into the upper text box in the "Real number" frame, and click the Write button in the same frame. (The data write function is executed within the range of D12 to D13.)**
7. **For reading real number data, click the Read button in the "Real number" frame. Data from the programmable controller CPU is displayed on the lower text box in the same frame. (The data read function is executed within the range of D12 to D13.)**
8. **When an error occurs, perform the troubleshooting corresponds to the error code. (☞ Page 378, CHAPTER 7 ERROR CODES)**
9. **Click the Close button to close the communication line.**

(2) Considerations for using the sample program

- When using the ActUtlType control, set the communication settings in Communication Setup Utility before executing the sample program.
- If an error which does not relate to the ActUtlType control occurs (for example, a character string or out-of-range value is entered for 32-bit integer or real number), the corresponding error message is displayed on the message box, and the program is terminated.

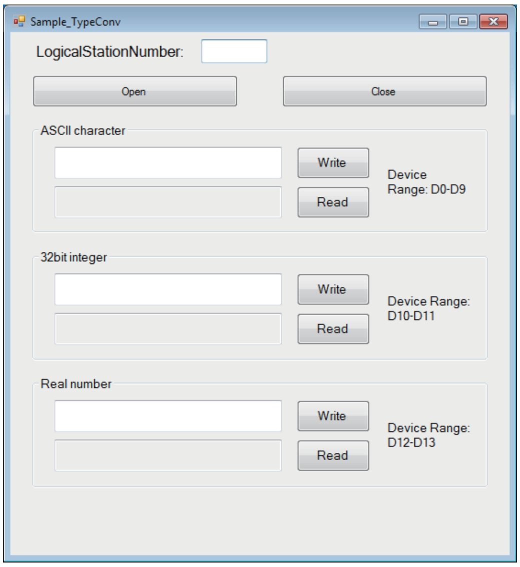
(3) Sample file

The sample program is installed in the following folder when the default pass is selected.

Folder name	File name
[User specified folder] - [Act] - [Samples] - [Vb.NET] - [Sample_TypeConv]	Sample_TypeConv.sln

(4) Screen

The following explains the settings on the sample program screen.



Item		Description
LogicalStationNumber		Enter the logical station number which was specified for the communication setting in Communication Setup Utility.
Open button		Open the communication line.
Close button		Close the communication line.
"ASII character" frame	Write button	Write the ASCII character data, which was entered into the upper text box in the frame, to the programmable controller CPU.
	Read button	Display the ASCII character data, which was read from the programmable controller CPU, on the lower text box in the frame.
"32bit integer" frame	Write button	Write the 32-bit integer data, which was entered into the upper text box in the frame, to the programmable controller CPU.
	Read button	Display the 32-bit integer data, which was read from the programmable controller CPU, on the lower text box in the frame.
"Real number" frame	Write button	Write the real number data, which was entered into the upper text box in the frame, to the programmable controller CPU.
	Read button	Display the real number data, which was read from the programmable controller CPU, on the lower text box in the frame.

6.5 Visual C++® .NET Sample Programs

6.5.1 Read/Write sample program

This sample program is a program to read/write device values of a programmable controller CPU and monitor device status using any of the ActUtlType control, the DotUtlType control, or the ActProgType control.

This sample program was created on Visual C++® .NET (Visual Studio® 2005) and Visual C++® .NET (Visual Studio® 2010).

(1) How to use the sample program

The operation is the same as that for Visual Basic® .NET (Visual Studio® 2005). (☞ Page 365, Section 6.4.2 (1) How to use the sample program)

(2) Considerations for using the sample program

The considerations are the same as those for Visual Basic® .NET (Visual Studio® 2005). (☞ Page 366, Section 6.4.2 (2) Considerations for using the sample program)

(3) Sample file list

The sample program is installed in the following folder when the default pass is selected.

(a) For Visual C++® .NET (Visual Studio® 2005)

Folder name	File name
[User specified folder] - [Act] - [Samples] - [Vc.NET] - [Sample2005]	Sample2005.sln
[User specified folder] - [Act] - [Samples] - [Vc.NET] - [Sample_References2005]	Sample_References2005.sln
[User specified folder] - [Act] - [Samples] - [Vc.NET] - [SampleDot2005]	Sample2005.sln
[User specified folder] - [Act] - [Samples] - [Vc.NET] - [Sample_ReferencesDot2005]	Sample_References2005.sln

(b) For Visual C++® .NET (Visual Studio® 2010)

Folder name	File name
[User specified folder] - [Act] - [Samples] - [Vc.NET] - [Sample2010]	Sample2010.sln
[User specified folder] - [Act] - [Samples] - [Vc.NET] - [Sample_References2010]	Sample_References2010.sln
[User specified folder] - [Act] - [Samples] - [Vc.NET] - [SampleDot2010]	Sample2010.sln
[User specified folder] - [Act] - [Samples] - [Vc.NET] - [Sample_ReferencesDot2010]	Sample_References2010.sln

(4) Screen

The settings on the screen are the same as those for Visual Basic® .NET (Visual Studio® 2005).
(☞ Page 366, Section 6.4.2 (4) Screen)

6.5.2 Troubleshooting function sample program

This sample program is a program to perform the troubleshooting function using the DotSupportMsg control.

(1) How to use the sample program

1. **Load the form.**

2. **Enter the error code into the "ErrorCode" text box to display the error description.**

3. **Click the** GetErrorMessage **button to display the error description and corrective action on the text box for displaying the error description.**

4. **Displayed error descriptions and corrective actions are the same as the descriptions on ☞ Page 378, CHAPTER 7 ERROR CODES.**

5. **Click the** Exit **button to exit the sample program.**

(2) Considerations for using the sample program

For error codes to be entered into the "ErrorCode" text box, enter the error code described on ☞ Page 378, Section 7.1 Error Codes Returned by Controls (0x********), or enter it in decimal number.

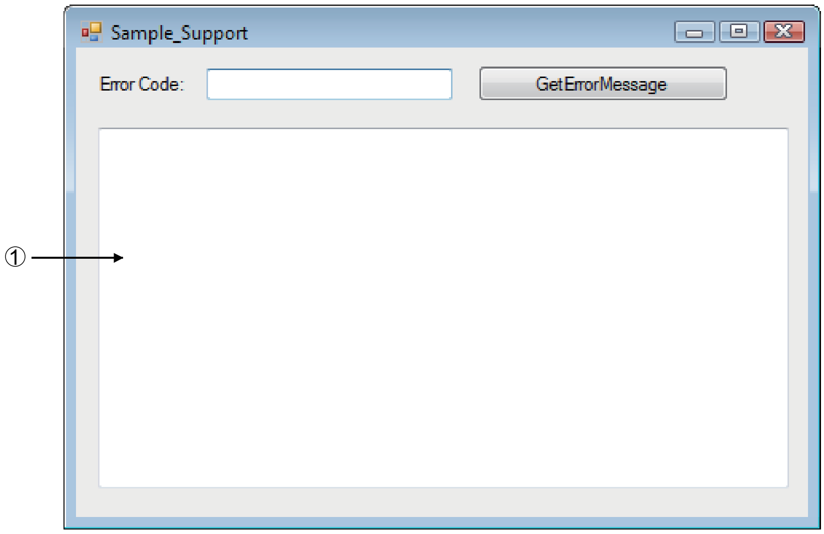
(3) Sample file

The sample program is installed in the following folder when the default pass is selected.

Folder name	File name
[User specified folder] - [Act] - [Samples] - [Vc.Net] - [Sample_Support]	Sample_Support.sln

(4) Screen

The following explains the settings on the sample program screen.



Item	Description
ErrorCode	Enter the error code.
GetErrorMessage button	Read the error description and corrective action of the error code entered into the "ErrorCode" text box.
Exit button	Exit the sample program.
① (Error definition displaying text box)	Display the error description and corrective action of the error code entered into the "ErrorCode" text box.

6.6 Visual C++ ® .NET (MFC) Sample Programs

This section explains the sample programs for Visual C++ ® .NET (MFC) which were created using the dispatch interface and the custom interface.

6.6.1 Dispatch interface

This sample program is a program to read a model of connected CPU and read/write device values using the ActUtlType control or the ActProgType control with the dispatch interface.

(1) How to use the sample program

1. **Load the form and select the control to be used.**

2. **Click the** Open Communication **button to open the communication line through Ethernet communication.**

3. **Click the** GetCpuType **button to display the model code and the CPU model of the currently-connected programmable controller CPU on the upper text box and the lower text box of "Output Data" respectively.**

4. **Enter the device to be read into the "Device Name" text box, and click the** GetDevice **button to display the device data on the upper text box of "Output Data".**

5. **Enter the device to be written into the "Device Name" text box, enter the device value to be written into the "Device Value" text box, and click the** SetDevice **button to write the device value.**

6. **Click the** Close Communication **button to close the communication line.**

7. **If an error occurs during the process, an error code is displayed on the "Return Value" text box.**

8. **When an error occurs, perform the troubleshooting corresponds to the error code. (☞ Page 378, CHAPTER 7 ERROR CODES)**

(2) Considerations for using the sample program

- When using the ActUtlType control, set the logical station number "1" for the Ethernet communication information in Communication Setup Utility before executing the sample program.
- When changing the control to be used, close the communication line by clicking the Close Communication button, change the control, and reopen the line.

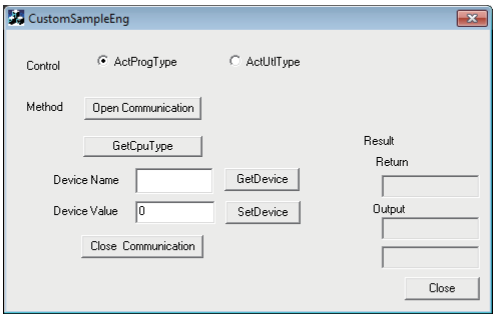
(3) Sample file

The sample program is installed in the following folder when the default pass is selected.

Folder name	File name
[User specified folder] - [Act] - [Samples] - [Vc] - [SampleENG]	sampleENG.sln

(4) Screen

The following explains the settings on the sample program screen.



Control ● ActProgType ○ ActUtlType

Device Value 0

Item		Description
Control		Select the control to be used.
Open Communication button		Open the communication line.
GetCpuType button		Read the programmable controller CPU model.
Device Name		Enter the device to be read/written.
Device Value		Enter the device value to be written.
Close Communication button		Close the communication line.
GetDevice button		Read the data of the device entered into the "Device Name" text box.
SetDevice button		Write the data of the device entered into the "Device Name" text box.
Return		Display the function execution result.
Output	Upper	Display the CPU model code or the read device value.
	Lower	Display the CPU model.

6.6.2 Custom interface

This sample program is a program to read a model of connected CPU and read/write device values using the ActUtlType control or ActProgType control with the custom interface.

(1) How to use the sample program

The operation is the same as that for the dispatch interface.(☞ Page 373, Section 6.6.1 (1) How to use the sample program)

(2) Considerations for using the sample program

- When using the ActUtlType control, set the logical station number "2" for the Ethernet communication information in Communication Setup Utility before executing the sample program.
- When changing the control to be used, close the communication line by clicking the Close Communication button, change the control, and reopen the line.

(3) Sample file

The sample files are installed into the following folders at default installation.

Folder name	File name
[User specified folder] - [Act] - [Samples] - [Vc] - [CustomSampleENG]	CustomSampleENG.sln

(4) Screen

The settings on the screen are the same as those for the dispatch interface. (☞ Page 374, Section 6.6.1 (4) Screen)

6.6.3 Troubleshooting function sample program

This sample program is a program to perform the troubleshooting function using the ActSupportMsg control.

(1) How to use the sample program

1. **Load the form.**

2. **Enter the error code into the "ErrorCode" text box to display the error description.**

3. **Click the** GetErrorMessage **button to display the error description and corrective action on the text box for displaying the error description.**

4. **Displayed error descriptions and corrective actions are the same as the descriptions on ☞ Page 378, CHAPTER 7 ERROR CODES.**

5. **Click the** Exit **button to exit the sample program.**

(2) Considerations for using the sample program

For error codes to be entered into the "ErrorCode" text box, enter the error code described on ☞ Page 378, Section 7.1 Error Codes Returned by Controls (0x********), or enter it in decimal number.

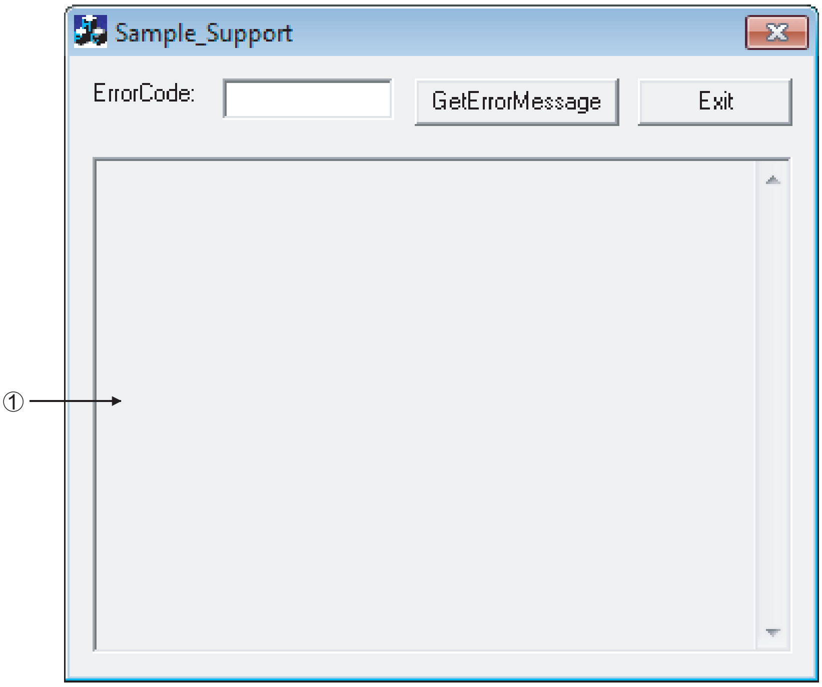
(3) Sample file

The sample program is installed in the following folder when the default pass is selected.

Folder name	File name
[User specified folder] - [Act] - [Samples] - [Vc] - [Sample_SupportENG]	Sample_SupportENG.sln

(4) Screen

The following explains the settings on the sample program screen.



Item	Description
ErrorCode	Enter the error code.
GetErrorMessage button	Read the error description and corrective action of the error code entered into the "ErrorCode" text box.
Exit button	Exit the sample program.
① (Error definition displaying text box)	Display the error description and corrective action of the error code entered into the "ErrorCode" text box.

CHAPTER 7 ERROR CODES

This chapter explains the error codes returned by controls and the error codes returned by CPUs, modules, and network boards.

7.1 Error Codes Returned by Controls

The following table shows the error codes returned by controls.

Error code	Error description	Corrective action
0x00000000	Normal end	—
0x01010002	Timeout error	• Check the property timeout value. • Check the settings in the communication settings utility. • Check the programmable controller, Unit settings, state of the cable, etc. • Close and Open again. • Exit the program and restart the personal computer.
0x01010005	Message error	• Check the system noise. • Check the property timeout value. • Check the settings in the communication settings utility. • Check the programmable controller, unit settings, status of the cable, etc. • Close and Open again. • Exit the program, restart the personal computer.
0x01010010	Programmable controller No. error Communication could not be made with the specified station number.	• Check the station number set on the communication setup utility. • Check the station number set to ActStationNumber.
0x01010011	Mode error Command not supported.	• Check if the correct CPU type setting is done. • Check the programmable controller, Unit settings, status of the cable, etc. • Exit the program and restart the personal computer. • Reinstall MX Component.
0x01010012	Special Unit Specification error	• Check the specified address of the special unit.
0x01010013	Other data error Communication cannot be made for some cause.	• Check that the system configuration is not an unsupported configuration. • Check that the CPU type setting is correct. • Exit the program and restart the personal computer.
0x01010018	Remote request error Remote operation is being performed in the path different from the communicating path.	• Cancel the remote operation being performed in the other path.
0x01010020	Link error Link communications could not be made.	• Check that reset operation is not performed for the other end of communication, the control station (master station) or the station passed through by routing. • Check that the network parameter setting is correct.
0x01010021	Special Unit Bus error There is no response from the special unit under consideration.	• Repair or exchange the special unit under consideration. • Exit the program and restart the personal computer. • Reinstall MX Component.
0x01800001	No command error The method does not support.	• The corresponding method does not support.
0x01800002	Memory lock error	• Exit the program and restart the personal computer. *1
0x01800003	Memory securing error	• Exit the program and restart the personal computer.*1 • Exit other programs and secure free memory area.

Error code	Error description	Corrective action
0x01800004	DLL load error	• Exit the program and restart the personal computer. • Exit other programs and secure free memory area. • Reinstall MX Component.
0x01800005	Resource securing error	• Exit the program and restart the personal computer. • Exit other programs and secure free memory area.
0x01801001	Resource Timeout error The resource could not be retrieved within the specified time.	• Execute again after the other object completes the communication. • Execute again after increasing the timeout value. • Exit the program and restart the personal computer. • Reinstall MX Component.
0x01801002	Multi-line open error	• Exit the program and restart the personal computer.
0x01801003	Open not yet executed	
0x01801004	Open Type error	• Exit the program and restart the personal computer. • Reinstall MX Component.
0x01801005	Specified port error	
0x01801006	Specified module error	• Check that the actual system configuration matches to the settings in the communication settings utility or the values of the properties. • Exit the program and restart the personal computer. • Reinstall MX Component.
0x01801007	Specified CPU error	• Check the CPU type set to ActCpuType. • Check that the system configuration is not an unsupported configuration. • Exit the program and restart the personal computer. • Reinstall MX Component. • Check the packet type set to ActPacketType.
0x01801008	Target station access error	• Review the target station.
0x01801009	Registry open failure Failed while opening data key of the registry.	• Exit the program and restart the personal computer. • Reinstall MX Component.
0x0180100A	Packet Type error The packet type specified is incorrect.	• Recheck the ActPacketType. • Exit the program and restart the personal computer. • Reinstall MX Component.
0x0180100B	Protocol Type error The protocol specified is incorrect.	• Exit the program and restart the personal computer. • Reinstall MX Component.
0x0180100C	Registry search failure	
0x0180100D	GetProcAddress failure	
0x0180100E	DLL non-load error	
0x0180100F	Another Object in execution Method cannot be executed because of exclusive control in progress.	• Execute again after some time.
0x01802001	Device error The device character string specified in the method is an unauthorized device character string.	• Review the device name.
0x01802002	Device number error The device character string number specified in the method is an unauthorized device number.	• Review the device number.
0x01802003	Program Type error	• Exit the program and restart the personal computer. • Reinstall MX Component.

*1 : When using an interface board for personal computer, refer to the following appendix.
(☞ Page 424, Appendix 4 Corrective Actions for Error Codes 0x01800002 and 0x018000003)

Error code	Error description	Corrective action
0x01802004	Sumcheck error The sumcheck value of the received data is abnormal.	• Check the module side sumcheck setting. • Check the sumcheck property of the control. • Check the cable. • Exit the program and restart the personal computer. • Reinstall MX Component.
0x01802005	Size error The number of points specified in the method is unauthorized.	• Check the number of points specified in the method. • Review the system, e.g. programmable controller CPU, module setting and cable status. • Exit the program and restart the personal computer. • Reinstall MX Component.
0x01802006	Block number error The block specifying number in the device character string specified in the method is unauthorized.	• Review the block specifying number in the device character string specified in the method.
0x01802007	Receive data error The data received is abnormal.	• Review the system, e.g. programmable controller CPU, module setting and cable status. • Check the cable. • Exit the program and restart the personal computer.
0x01802008	Write Protect error	• Exit the program and restart the personal computer. • Reinstall MX Component.
0x01802009	Reading Parameters error	
0x0180200A	Writing Parameters error	
0x0180200B	Programmable controller type mismatch The CPU type set to the property and the CPU type set on the communication settings utility do not match the CPU type on the other end of communication.	• Set the correct CPU type as the CPU type of the property. • Set the correct CPU type on the communication settings utility. • Review the system, e.g. programmable controller CPU, module setting and cable status.
0x0180200C	Request Cancel error The request was cancelled while being processed.	• Exit the program and restart the personal computer. • Reinstall MX Component.
0x0180200D	Drive Name error The specified drive name is incorrect.	
0x0180200E	Beginning Step error The beginning step specified is incorrect.	
0x0180200F	Parameter Type error The parameter type is incorrect.	
0x01802010	File Name error The file name is incorrect.	
0x01802011	Status error The status of Registration/Cancellation/Setting is incorrect.	
0x01802012	Detailed Condition Field error	
0x01802013	Step Condition error	
0x01802014	Bit Device Condition error	
0x01802015	Parameter Settings error	
0x01802016	Error in specifying telephone exchange number. Method does not support the operations corresponding to the specified telephone exchange number.	• Check the telephone exchange number. • Check if the method being executed is supported or not. • Check the system configuration such as programmable controller, unit, etc.
0x01802017	Keyword error	• Exit the program and restart the personal computer. • Reinstall MX Component.
0x01802018	Read/Write Flag error	
0x01802019	Refresh Method error	
0x0180201A	Buffer Access Method error	
0x0180201B	Start Mode/Stop Mode error	

Error code	Error description	Corrective action
0x0180201C	Written clock data error Clock data specified for write cannot be written properly since that data is in error.	• Review the clock data to be written.
0x0180201D	Online clock data write error Write of clock data failed. Clock data cannot be written since the programmable controller CPU is during RUN.	• Place the programmable controller CPU in the STOP status.
0x0180201E	ROM drive error	• Exit the program and restart the personal computer. • Reinstall MX Component.
0x0180201F	While Tracing error Invalid operation was carried out during trace.	
0x01802020	First I/O number error The first I/O number specified in the method is an unauthorized value.	• Check the value of the first I/O number specified in the method. • Using the GPP function, check the programmable controller CPU parameters (I/O assignment). • Exit the program and restart the personal computer.
0x01802021	First address error The buffer address specified in the method is an unauthorized value.	• Check the value of the buffer address specified in the method. • Exit the program and restart the personal computer.
0x01802022	Pattern error	• Exit the program and restart the personal computer. • Reinstall MX Component.
0x01802023	SFC Block No. error	
0x01802024	SFC Step No. error	
0x01802025	Step No. error	
0x01802026	Data error	
0x01802027	System Data error	
0x01802028	Error in number of TC settings Value	
0x01802029	Clear Mode error	
0x0180202A	Signal Flow error	
0x0180202B	Version Control error	
0x0180202C	Monitor Not Registered error	
0x0180202D	PI Type error	
0x0180202E	PI No error	
0x0180202F	Error in Number of PIs	
0x01802030	Shift error	
0x01802031	File Type error	
0x01802032	Specified Unit error	
0x01802033	Error check flag error	
0x01802034	Step RUN operation error	
0x01802035	Step RUN data error	
0x01802036	During Step RUN error	
0x01802037	Write error while running program corresponding to E^{2}PROM	
0x01802038	Clock data read/write error The clock data read/write method was executed for the programmable controller CPU which does not have the clock devices.	• Do not execute clock data read/write.
0x01802039	Trace not completed error	• Exit the program and restart the personal computer. • Reinstall MX Component.
0x0180203A	Registration Clear Flag error	
0x0180203B	Operation error	
0x0180203C	Error in the number of exchanges	
0x0180203D	Error in number of loops specified	

Error code	Error description	Corrective action
0x0180203E	Retrieve data selection	• Exit the program and restart the personal computer. • Reinstall MX Component.
0x0180203F	Error in number of SFC cycles	
0x01802040	Motion programmable controller error	
0x01802041	Motion programmable controller Communication error	
0x01802042	Fixed execution time setting error	
0x01802043	Error in number of functions	
0x01802044	System information specification error	
0x01802045	Registration Condition Not Formed error	
0x01802046	Function No. error	
0x01802047	RAM drive error	
0x01802048	ROM drive error at the booting side	
0x01802049	Transfer mode specification error at the booting side	
0x0180204A	Insufficient memory error	
0x0180204B	Back up drive ROM error	
0x0180204C	Block size error	
0x0180204D	Detached during RUN state error	
0x0180204E	Unit Already Registered error	
0x0180204F	Password Registration Data Full error	
0x01802050	Password Not Registered error	
0x01802051	Remote Password error	
0x01802052	IP Address error	
0x01802053	Timeout value out of range error	
0x01802054	Command not detected error	
0x01802055	Trace execution type error	
0x01802056	Version error	
0x01802057	Tracking cable error The tracking cable is faulty. The programmable controller CPU status is error.	• Reexamine the system such as the programmable controller CPU, module setting and cable status.
0x0180205C	Keyword protection error Programmable controller is protected by the key word.	• Disable the keyword and execute again.
0x0180205D	Keyword disable error The inputted keyword is wrong.	• Input a correct keyword.
0x0180205E	Keyword protecting error Programmable controller did not accept the protecting command.	• Execute again or re-switch the power of the programmable controller.
0x0180205F	Keyword entry error An illegal character is included in the inputted keyword.	• Input a correct keyword.
0x01802060	Keyword deletion error The inputted keyword is wrong.	
0x01802062	Received packet CRC check error An error occurred in CRC check for receive packet data.	• Execute the communication process again.
0x01802063	Received packet CRC check error An error occurred in CRC check for whole data file of receive packet.	
0x01802064	FX Series programmable controller connection error	• Please contact out telephone center.
0x01802070	Online change program error No target program for online change exists in the programmable controller CPU.	• Execute the online change after turning the programmable controller CPU to STOP.

Error code	Error description	Corrective action
0x01802071	Ether direct communication multiple response receive error Multiple responses were received during Ether direct communication.	• Check that the personal computer and the programmable controller CPU are in a one-to-one connection.
0x01802072	Ether direct communication error Cannot communicate because the programmable controller CPU is being accessed by another personal computer during Ether direct communication.	
0x01802073	Programmable controller CPU search response error The number of responses in the programmable controller CPU search exceeded the maximum number to be searched.	• Reduce the number of programmable controllers on the network to 1024 or less.
0x01802074	Redundant system other system connection diagnostics error	• Disconnect the cable and connect it to the currently disconnected programmable controller CPU. • Or, change the redundant CPU specification to the self system.
0x01808001	Multiple Open error Open method was executed while it was open.	• Exit the program and restart the personal computer. • Execute any method other than Open.
0x01808002	Channel number specifying error The port number set to the property and the port number set on the communication settings utility are unauthorized values.	• Set the correct value to the port number of the property. • Make communication settings again on the communication settings utility.
0x01808003	Driver not yet started The network board driver is not started.	• Start the driver.
0x01808004	Error in overlap event generation	• Exit the program and restart the personal computer. • Reinstall MX Component.
0x01808005	MUTEX generation error Creation of MUTEX to exercise exclusive control failed.	
0x01808006	Error in socket object generation Socket object could not be created	
0x01808007	Socket object generation error Creation of the Socket object failed.	• Check for a running application which uses the same port number. • Retry after changing the port number value of the property. • Retry after changing the port number value on the communication settings utility. • Make Ethernet board and protocol settings on the control panel of the OS. • Exit the program and restart the personal computer.
0x01808008	Port connection error Establishment of connection failed. The other end does not respond.	• Review the IP address and port number values of the properties. • Review the port number value on the communication settings utility. • Review the system, e.g. programmable controller CPU, module setting and cable status. • Exit the program and restart the personal computer.
0x01808009	COM port handle error The handle of the COM port cannot be acquired. The COM port object cannot be copied. The SOCKET object cannot be copied.	• Check for an application which uses the COM port. • Exit the program and restart the personal computer.
0x0180800A	Buffer size setting error Setting of the COM port buffer size failed.	• Check for an application which uses the COM port. • Make COM port setting on the control panel of the OS. • Exit the program and restart the personal computer.
0x0180800B	DCB value acquisition error Acquisition of the COM port DCB value failed.	• Check for an application which uses the COM port. • Make COM port setting on the control panel of the OS. • Exit the program and restart the personal computer.

Error code	Error description	Corrective action
0x0180800C	DCB setting error Setting of the COM port DCB value failed.	• Check for an application which uses the COM port. • Make COM port setting on the control panel of the OS. • Exit the program and restart the personal computer.
0x0180800D	Time-out value setting error Setting of the COM port time-out value failed.	• Review the time-out value of the property. • Review the time-out value on the communication settings utility. • Check for an application which uses the COM port. • Make COM port setting on the control panel of the OS. • Exit the program and restart the personal computer.
0x0180800E	Shared memory open error Open processing of shared memory failed.	• Check whether the GX Simulator has started. • Exit the program and restart the personal computer.
0x01808101	Duplex close error	• Exit the program and restart the personal computer.
0x01808102	Handle close error Closing of the COM port handle failed.	
0x01808103	Driver close error Closing of the driver handle failed.	
0x01808104	Overlap Event Close error	• Exit the program and restart the personal computer. • Reinstall MX Component.
0x01808105	Mutex Handle Close error	
0x01808106	COM Port Handle Close error	
0x01808201	Send error Data send failed.	• Review the system, e.g. programmable controller CPU, module setting and cable status. • Make COM port setting on the control panel of the OS. • Make Ethernet board and protocol settings on the control panel. • Retry the method. • Exit the program and restart the personal computer.
0x01808202	Send data size error Data send failed.	• Exit the program and restart the personal computer.
0x01808203	Queue clear error Clearing of the COM port queue failed.	• Exit the program and restart the personal computer. • Perform Close once and execute Open again.
0x01808301	Receive error Data receive failed.	• Review the system, e.g. programmable controller CPU, module setting and cable status. • Review the time-out value of the property. • Review the time-out value on the communication settings utility. • Retry the method. • Exit the program and restart the personal computer.
0x01808302	Not Sent error	• Exit the program and restart the personal computer. • Reinstall MX Component.
0x01808303	Error in retrieving Overlap Event	
0x01808304	Receive buffer size shortage Receive data was larger than the receive buffer size prepared for the system.	• Exit the program and restart the personal computer.
0x01808401	Control error Changing of the COM port communication control failed.	
0x01808402	Signal Line Control error	• Exit the program and restart the personal computer. • Reinstall MX Component.
0x01808403	Signal line specifying error Changing of the COM port communication control failed.	• Exit the program and restart the personal computer.
0x01808404	Open not yet executed	• Execute Open. • Exit the program and restart the personal computer.
0x01808405	Communication parameter error The data bit and stop bit combination of the properties is unauthorized.	• Review the data bit and stop bit values of the properties. • Set them again on the communication settings utility.

Error code	Error description	Corrective action
0x01808406	Transmission speed value specifying error The transmission speed of the property is unauthorized.	• Review the transmission speed value of the property. • Set it again on the communication settings utility.
0x01808407	Data length error The data bit value of the property is unauthorized.	• Review the data bit value of the property. • Set it again on the communication settings utility.
0x01808408	Parity specifying error The parity value of the property is unauthorized.	• Review the parity value of the property. • Set it again on the communication settings utility.
0x01808409	Stop bit specifying error The stop bit value of the property is unauthorized.	• Review the stop bit value of the property. • Set it again on the communication settings utility.
0x0180840A	Communication control setting error The control value of the property is unauthorized.	• Review the control value of the property. • Set it again on the communication settings utility.
0x0180840B	Time-out error Though the time-out period had elapsed, data could not be received.	• Review the time-out value of the property. • Set it again on the communication settings utility. • Review the system, e.g. programmable controller CPU, module setting and cable status. • Retry the method. • Perform Close once and execute Open again. • Exit the program and restart the personal computer.
0x0180840C	Connect error	• Exit the program and restart the personal computer.
0x0180840D	Duplex connect error	
0x0180840E	Attach failure Attaching of the socket object failed.	
0x0180840F	Signal line status acquisition failure Acquisition of the COM port signal line status failed.	
0x01808410	CD signal line OFF The CD signal on the other end of communication is in the OFF status.	• Review the system, e.g. programmable controller CPU, module setting and cable status. • Exit the program and restart the personal computer.
0x01808411	Password mismatch error	• Check the remote password of the property.
0x01808412	TEL Communication error	• Exit the program and restart the personal computer. • Reinstall MX Component.
0x01808501	USB driver load error Loading of the USB driver failed.	• Exit the program and restart the personal computer. • Reinstall MX Component. • Check USB driver installation.
0x01808502	USB driver connect error Connection of the USB driver failed.	
0x01808503	USB driver send error Data send failed.	• Review the system, e.g. programmable controller CPU, module setting and cable status. • Make USB setting on the control panel (device manger) of the OS. • Retry the method. • Exit the program and restart the personal computer.
0x01808504	USB driver receive error Data receive failed.	
0x01808505	USB Driver Timeout error	• Recheck the timeout value. • Exit the program and restart the personal computer. • Reinstall MX Component.
0x01808506	USB driver initialization error Initialization of the USB driver failed.	• Make USB setting on the control panel (device manger) of the OS. • Exit the program and restart the personal computer.
0x01808507	Other USB error Error related to data send/receive occurred.	• Disconnect the cable once, then reconnect. • Exit the program and restart the personal computer. • Reinstall MX Component.
0x01809000	GX Works2 uninstallation error The error occurred by retrieving the installation passing of GX Simulator2.	• Reinstall GX Works2.
0x01809001	GX Simulator2 unstart error GX Simulator2 did not start.	• Start GX Simulator2.

Error code	Error description	Corrective action
0x01809002	GX Simulator2 start error	• Exit the program and restart the personal computer. • Reinstall MX Component.
0x01809003	GX Simulator2 start time-out error	
0x01809004	GX Simulator2 stop error	
0x01809005	GX Simulator2 start error	
0x01809007	GX Simulator2 stop error	
0x01809008	GX Simulator2 start error Because it had reached upper bounds of the number of simulations that was able to be started at the same time, it was not possible to start.	
0x01809009	GX Simulator2 start error The simulation of only one project that can be started has started.	
0x01809010	GX Simulator2 start information illegal error The error occurred because it was not able to secure the memory area to allocate GX Simulator2 start information.	
0x01809021	GX Simulator2 start error Because it had reached upper bounds of the number of simulations that was able to be started at the same time, it was not possible to start.	
0x01809022	GX Simulator2 start error The simulation of other CPU was not able to begin because the simulation of the project of FXCPU had already been begun.	
0x02000001	Points Exceeded error The number of points registered in the monitoring server is very high.	• Reduce the no. of points registered by the monitor. • Exit the program and restart the personal computer. • Reinstall MX Component.
0x02000002	Shared memory creation error Failed in creating shared memory.	• Exit the program and restart the personal computer. • Reinstall MX Component.
0x02000003	Shared memory access error	
0x02000004	Memory Secure error Failed in securing memory for the monitoring server.	• Close the other applications. • Increase the system memory. • Exit the program and restart the personal computer. • Reinstall MX Component.
0x02000005	Device Not Registered error Monitor has not been registered	• Register the monitor in the monitoring server. • Exit the program and restart the personal computer. • Reinstall MX Component.
0x02000006	Monitoring Server Startup error Monitoring Server is not started.	• Start the Monitoring Server. • Exit the program and restart the personal computer. • Reinstall MX Component.
0x02000010	Yet to retrieve Device Value error Monitoring is not yet completed.	• Try to retrieve the value again after waiting for a fixed amount of time. • Exit the program and restart the personal computer. • Reinstall MX Component.
0x03000001	Command not Supported. Command is not supported.	• Exit the program and restart the personal computer. • Reinstall MX Component.
0x03000002	Memory Lock error Failed while locking memory.	
0x03000003	Error Securing Memory Failed in securing the memory.	
0x03000004	DLL read error Failed in reading DLL.	
0x03000005	Error in securing Resources. Failed in securing the resources.	

Error code	Error description	Corrective action
0x03010001	File Creation error Failed in creating the file.	• Check if there is enough space on the hard disk. • Exit the program and restart the personal computer. • Reinstall MX Component.
0x03010002	File Open error Failed to open the file.	• Exit the program and restart the personal computer. • Reinstall MX Component.
0x03010003	Buffer Size error The buffer size specified is either incorrect or not enough.	
0x03010004	SIL Sentence formation error SIL sentence formation is incorrect.	
0x03010005	Filename error The specified filename is too long.	• Specify a shorter filename. • Exit the program and restart the personal computer. • Reinstall MX Component.
0x03010006	File does not exist error The specified file does not exist.	• Check the filename. • Check if the file exists or not. • Exit the program and restart the personal computer. • Reinstall MX Component.
0x03010007	File Structure error The data structure in the specified file is incorrect.	• Exit the program and restart the personal computer. • Reinstall MX Component.
0x03010008	File already exists error The specified file already exists.	• Check the filename. • Exit the program and restart the personal computer. • Reinstall MX Component.
0x03010009	File does not exist error The specified file does not exist.	
0x0301000A	File Deletion error The specified file could not be deleted.	• Exit the program and restart the personal computer. • Reinstall MX Component.
0x0301000B	Multiple Open error The specified project has been opened twice.	
0x0301000C	Filename error The specified filename is incorrect.	• Check the filename. • Exit the program and restart the personal computer. • Reinstall MX Component.
0x0301000D	File Read error Falied in reading the file.	• Exit the program and restart the personal computer. • Reinstall MX Component.
0x0301000E	File Write error Failed in writing the file.	
0x0301000F	File Seek error File seek failed.	
0x03010010	File Close error Failed while closing the file.	
0x03010011	Folder Creation error Failed while creating the folder.	
0x03010012	File Copy error Failed while copying the file.	
0x03010013	Project Path error The length of the project path is incorrect.	
0x03010014	Project Type error The project type is incorrect.	
0x03010015	File Type error The file type is incorrect.	
0x03010016	Sub-File Type error The sub-file type is incorrect.	
0x03010017	Insufficient Disk space error The disk space is insufficient.	

Error code	Error description	Corrective action
0x03020002	Multiple Open error Tried to open DBProduct more than once.	• Exit the program and restart the personal computer. • Reinstall MX Component.
0x03020003	Not Opened error DBProduct is not opened.	
0x03020004	Extract error DBProduct is not extracted.	
0x03020010	Parameter error The parameters of DBProduct are incorrect.	
0x03020011	Language error The language parameter is incorrect.	
0x03020012	Error in specifying Maker The maker parameter is incorrect.	
0x03020013	Error in specifying Unit The unit parameter is incorrect.	
0x03020014	SQL Parameter error SIL, SQL Parameter of DBProduct is incorrect.	
0x03020015	SIL Sentence formation error SIL sentence formation is incorrect.	
0x03020016	Field Key Input error The field key entered is incorrect.	
0x03020050	Record Data Construction error. Failed in reconstructing the record data of DBProduct.	
0x03020060	Error Retrieving Record Data Failed while retrieving DBProduct record data.	
0x03020061	Last Record error Cannot retrieve the next record since the current record is the last record.	
0x03FF0000	Initialization error	
0x03FF0001	Not Initialized error	
0x03FF0002	Multiple Initialization error	
0x03FF0003	Workspace Initialization error	
0x03FF0004	Database Initialization error	
0x03FF0005	Recordset Initialization error	
0x03FF0006	Error Closing Database	
0x03FF0007	Error Closing Recordset	
0x03FF0008	Database Not Opened error Database is not opened.	
0x03FF0009	Recordset Not Opened error Recordset is not opened.	
0x03FF000A	Table Initialization error Failed in initializing TtableInformation table.	
0x03FF000B	Table Initialization error Failed in initializing TfieldInformation table.	
0x03FF000C	Table Initialization error Failed in initializing TrelationInformation table.	
0x03FF000D	Table Initialization error Failed in initializing Tlanguage table.	
0x03FF000E	Table Initialization error Failed in initializing Tmaker table.	
0x03FF000F	Table Initialization error Failed in initializing TOpenDatabase table.	

Error code	Error description	Corrective action
0x03FF0010	Field Value error	• Exit the program and restart the personal computer. • Reinstall MX Component.
0x03FF0011	Field Value error	
0x03FF0012	Exit error Failed to exit the database.	
0x03FF0100	Moving Record error Failed while moving the record.	
0x03FF0101	Retrieving Record Count error Failed to retrieve the record count.	
0x03FF0110	Retrieving Field Value error Failed in retrieving the field value.	
0x03FF0111	Setting Field Value error Failed in setting the field value.	
0x03FFFFFF	Other errors	
0x04000001	No command error The specified CPU type cannot be used to perform processing.	• Check the CPU type set to ActCpuType. • Check whether the system configuration is supported or not. • Exist the program and restart the personal computer. • Reinstall MX Component.
0x04000002	Memory lock error Failed in locking memory.	• Exit the program and restart the personal computer. • Reinstall MX Component.
0x04000003	Securing Memory error Failed in securing the memory.	
0x04000004	Internal server DLL load error Start of the internal server failed.	• Check for the deleted or moved installation file of MX Component. • Exit the program and restart the personal computer. • Reinstall MX Component.
0x04000005	Securing Resources error Failed in securing the resources.	• Exit the program and restart the personal computer. • Reinstall MX Component.
0x04000006	Error Loading Main Object Failed in reading the file.	
0x04000007	Error Loading Conversion Table Failed in reading table data.	
0x04000100	Incorrect Intermediate Code Size error	
0x04010001	Intermediate Code Not Converted error The converted machine code for one command is more than 256 bytes.	
0x04010002	Intermediate Code Completion error Intermediate code area of the code to be converted ended abruptly.	
0x04010003	Insufficient Intermediate Code error The intermediate code of the code to be converted was insufficient.	
0x04010004	Intermediate Code Data error The intermediate code to be converted is incorrect.	
0x04010005	Intermediate Code Structure error The number of steps in the intermediate code is incorrect.	
0x04010006	Error in Number of Steps The number of steps in comment intermediate code is incorrect.	
0x04010007	Insufficient Storage Space for Machine Code error The storage space for machine code is insufficient.	

Error code	Error description	Corrective action
0x04010008	Other errors (Other errors generated during the conversion of Intermediate code to machine code.)	• Exit the program and restart the personal computer. • Reinstall MX Component.
0x04011001	Machine Code Not Converted error The converted intermediate code for one command is more than 256 bytes.	
0x04011002	Machine Code Completion error The machine code area to be converted ended abruptly.	
0x04011003	Abnormal Machine Code Could not convert since the machine code to be converted was abnormal.	
0x04011004	Insufficient Storage Space for Intermediate Code error The storage area for intermediate code is insufficient.	
0x04011005	Other errors Other errors generated while converting machine code to Intermediate code.	
0x04020001	Text Code Not Converted error The converted intermediate code for one command is more than 256 bytes.	
0x04020002	No Input error The input list code is insufficient.	
0x04020003	Command error The command name of list code to be converted is incorrect.	
0x04020004	Device error The device name of list code to be converted is incorrect.	
0x04020005	Device Number error The device number of the list code to be converted is out of range.	
0x04020006	Conversion error The list code to be converted conversion could not be identified.	
0x04020007	Text Data error The list code to be converted is incorrect.	
0x04020008	Error in SFC Operation Output The output command of SFC operation is incorrect.	
0x04020009	SFC Shift Condition error SFC shift condition command is incorrect.	
0x0402000A	Error in Statements between lines The statements entered between lines are incorrect.	
0x0402000B	P.I Statement error The P.I statement entered is incorrect.	
0x0402000C	Note error The Note entered is incorrect.	
0x0402000D	Comment error The comment entered is incorrect.	
0x0402000E	Other errors (Other errors generated during the conversion of list to Intermediate code)	
0x04021001	Intermediate Code Not Converted error The converted list code for one command has exceeded 256 bytes.	

Error code	Error description	Corrective action
0x04021002	Intermediate Code Area Full error Intermediate code area to be converted is full.	• Exit the program and restart the personal computer. • Reinstall MX Component.
0x04021003	Command error The command specified by the intermediate code to be converted is incorrect.	
0x04021004	Device error The device specified in the intermediate code to be converted is incorrect.	
0x04021005	Intermediate Code error The structure of intermediate code to be converted is incorrect.	
0x04021006	Insufficient List Storage Space error The space for storing the converted list code is insufficient.	
0x04021007	Other errors (Other errors generated during the conversion of intermediate code to list)	
0x04030001	Not Converted error The storage space for converted intermediate code is insufficient.	
0x04030002	Bad Circuit Creation error The character memory circuit is not completed in a sequence.	
0x04030003	Specified Circuit Size Exceeded Specified circuit size is too big.	
0x04030004	Incorrect Return Circuit error There is no consistency before and after the return circuit. The setting for the return circuit is too high.	
0x04030005	Other errors (Other errors generated while converting from Character Memory to Intermediate Code)	
0x04031001	Not Converted error The size (vertical/horizontal) of the character memory specified is incorrect.	
0x04031002	Abnormal Command Code error The command intermediate code to be converted is incorrect.	
0x04031003	Bad Circuit Creation error Could not be converted to Sequence Circuit. There is no END command.	
0x04031004	Specified Circuit Size exceeded error Specified circuit size is too big.	
0x04031005	Fatal error Fatal error has occurred.	
0x04031006	Insufficient number of storage blocks error The space to store the converted character memory circuit blocks is not sufficient.	
0x04031007	Circuit Block Search error Data is broken off in the circuit block.	
0x04031008	Other errors (Other errors generated during the conversion of intermediate code to character memory)	

Error code	Error description	Corrective action
0x04040001	CAD Data error There is no CAD data to be converted. The CAD data format is incorrect.	• Exit the program and restart the personal computer. • Reinstall MX Component.
0x04040002	Output Data error The input CAD data type and the output CAD data type are not matching.	
0x04040003	Library Load error Failed to load the library.	
0x04040004	Storage Space Secure error The space secured to store the converted data is not sufficient.	
0x04040005	No END Command error There is no END command in the CAD data to be converted.	
0x04040006	Abnormal Command Code There is abnormal command code in the CAD data to be converted.	
0x04040007	Device No. error The device number is out of range.	
0x04040008	Step No. error The step number is out of range.	
0x04040009	The specified circuit size exceeded error. 1 circuit block is too big.	
0x0404000A	Return Circuit Error The return circuit is incorrect.	
0x0404000B	Bad Circuit Creation error The circuit data is incorrect.	
0x0404000C	SFC Data error The SFC data to be converted is incorrect.	
0x0404000D	List Data error The list data to be converted is incorrect.	
0x0404000E	Comment Data error The comment data to be converted is incorrect.	
0x0404000F	Statement error The statement data to be converted is incorrect.	
0x04040010	Other errors (Other errors generated during the conversion of CAD code to Intermediate code.)	
0x04041001	Intermediate Code Data error There is no intermediate code to be converted. The format of the intermediate code is incorrect.	
0x04041002	CAD Data Type error The input CAD data type and the output CAD data type are not matching.	
0x04041003	Library error Failed to load the library.	
0x04041004	Insufficient Input Data error Data to be converted is insufficient.	
0x04041005	Insufficient Storage Space error There is not enough space to store the CAD data to be converted.	

Error code	Error description	Corrective action
0x04041006	No END Command error There is no END command in the CAD data to be converted.	• Exit the program and restart the personal computer. • Reinstall MX Component.
0x04041007	Abnormal Command Code There is abnormal command code in the CAD data to be converted.	
0x04041008	Device No. error The device number is out of range.	
0x04041009	Step No. error The step number is out of range.	
0x0404100A	The specified circuit size exceeded error 1 circuit block is too big.	
0x0404100B	Return Circuit error The return circuit is incorrect.	
0x0404100C	Bad Circuit Creation error The circuit data is incorrect.	
0x0404100D	SFC Data error The SFC data to be converted is incorrect.	
0x0404100E	List Data error The list data to be converted is incorrect.	
0x0404100F	Comment Data error The comment data to be converted is incorrect.	
0x04041010	Statement error The statement data to be converted is incorrect.	
0x04041011	Other errors (Other errors generated during the conversion of Intermediate code to CAD code.)	
0x040A0001	Insufficient Intermediate Code Storage Space The space to store the data after conversion is insufficient.	
0x040A0002	The space to store addition SFC information is not sufficient	
0x040A0003	Conversion error	
0x040A0004	Non-SFC Program error	
0x040A1001	Step Not Used / No Output error	
0x040A1002	Step No out of range error	
0x040A1003	Step Not Used / No Output error	
0x040A1004	Transition No out of range.	
0x040A1005	Maximum Number Exceeded error	
0x040A1006	Microcontroller Program space error	
0x040A1007	Non-SFC Program error	
0x040B0001	Insufficient Intermediate Code Storage Space. The space to store the data after conversion is insufficient.	
0x040B0002	Conversion error	
0x040B1001	Failed in creating Step Start position table	
0x040B1002	Error Reading Step Information	
0x040B1003	Step No. error	
0x040B1004	Failed in reading the output of operation/Transition condition intermediate code error.	
0x040B1005	Securing Internal Work Area Failed error	
0x040B1006	Error in setting the maximum value of X direction for character memory	

Error code	Error description	Corrective action
0x040B1007	Insufficient Internal Work Area error	• Exit the program and restart the personal computer. • Reinstall MX Component.
0x040B1008	Stack Overflow, Abnormal Character Memory	
0x040B1009	Insufficient No of Storage Blocks error	
0x040B100A	Non-SFC Program error	
0x04050001	Abnormal Character String Specified error Device character string specified is incorrect.	
0x04050002	Device Points error Device points are out of range.	
0x04050003	Other errors (The errors generated during the conversion of the Device Character String to Device Intermediate Code)	
0x04051001	Device Name error The classification specified for the device intermediate code is incorrect.	
0x04051002	Device Name error The classification specified for the extended specification device intermediate code is incorrect.	
0x04051003	Other errors (The errors generated during the conversion of the Device Intermediate Code to Device Character String)	
0x04052001	Abnormal Character String Specified error Device character string specified is incorrect.	
0x04052002	Device Points error Device points are out of range.	
0x04052003	Other errors (The errors generated during the conversion of the Device Character String to Device Representation Code)	
0x04053001	Device Representation error The classification specified for the device intermediate code is incorrect.	
0x04053002	Device Representation error The classification specified for the extended specification device intermediate code is incorrect.	
0x04053003	Device Representation error The rectification part specified for the device is incorrect.	
0x04053004	Device Representation error The rectification part specified for the extended device is incorrect.	
0x04053005	Other errors (The errors generated during the conversion of the Device Representation Code to Device Character String)	
0x04064001	Abnormal Device Intermediate Code error The intermediate code for the device is incorrect.	
0x04064002	Other errors (Other errors generated during the conversion of the Intermediate code for the Device to Device Name)	
0x04065001	Abnormal Device Name error The classification specified for the intermediate code of the device is incorrect.	
0x04065002	Abnormal Device Name error The classification for the intermediate code of the extended specification device is incorrect.	

Error code	Error description	Corrective action
0x04065003	Other errors (Other errors generated during the conversion of the device name to Intermediate code)	• Exit the program and restart the personal computer. • Reinstall MX Component.
0x04066001	Device Intermediate Code error The intermediate code for the device is incorrect.	
0x04066002	Other errors (Other errors generated during the conversion of the device intermediate code to device representation code.)	
0x04067001	Device Representation error The classification specified for the intermediate code of the device is incorrect.	
0x04067002	Device Representation error The classification for the intermediate code of the extended specification device is incorrect.	
0x04067003	Device Representation error The rectification part specified for the device is incorrect.	
0x04067004	Device Representation error The rectification part specified for the extended device is incorrect.	
0x04067005	Other errors (Other errors generated during the conversion of device representation code to the device intermediate code)	
0x04070001	Common Data Conversion error The input data of the device comment conversion is incorrect.	
0x04070002	Insufficient Common Data The data to be converted is insufficient.	
0x04070003	Insufficient Storage Area The area where the conversion data is stored is insufficient.	
0x04071001	Error in CPU Data Conversion The input data of the device comment conversion is incorrect.	
0x04071002	Insufficient CPU Data error The data to be converted is insufficient.	
0x04071003	Insufficient Storage Area The area where the conversion data is stored is insufficient.	
0x04072001	Open error Failed in creating conversion object.	
0x04072002	CPU Type error The specified CPU type does not exist.	
0x04072003	Not Converted error Converted object does not exist.	
0x04072004	Input Data error The input data is incorrect.	
0x04073001	Program Common Data Conversion error	
0x04073002	Program Common Data Conversion error	
0x04073101	Program CPU Data Conversion error	
0x04074001	Common Data Parameter error	
0x04074002	Network Parameter Common Data error The parameter block exists, but the data inside is not set.	

Error code	Error description	Corrective action
0x04074101	Parameter CPU Data error	• Exit the program and restart the personal computer. • Reinstall MX Component.
0x04074102	Network Parameter CPU Data error The parameter block exists, but the data inside is not set.	
0x04074103	Offset error	
0x04074201	Error in Specifying Network Type The CPU specified does not support the network type.	
0x04074202	Parameter Block Number error The Block corresponding to the parameter block number specified does not exist.	
0x04074203	Parameter Block Content error It is different from the content supported by the specified.	
0x04074204	Parameter Block Information error The specified block number does not exist.	
0x04074205	Default Parameter Block is Abnormal The specified block number does not exist.	
0x04074301	Error in Conversion of the Common Parameter Block	
0x04074302	Error in Common Parameter Block No. 1001 The value of the RUN-PAUSE settings existence flag is incorrect.	
0x04074303	Error in Common Parameter Block No. 1003	
0x04074304	Error in Common Parameter Block No. 1008	
0x04074305	Error in Common Parameter Block No. 1100	
0x04074306	Error in Common Parameter Block No. 2001 The device intermediate code specified does not exist.	
0x04074307	Error in Common Parameter Block No. 3000	
0x04074308	Error in Common Parameter Block No. 3002	
0x04074309	Error in Common Parameter Block No. 3004 The settings for the annunciator display mode is incorrect.	
0x0407430A	Error in Common Parameter Block No. 4000 I/O Allotment Data is not created.	
0x0407430B	Error in Common Parameter Block No. 5000 The specified network is not supported.	
0x0407430C	Error in Common Parameter Block No. 5001 Valid unit No is not set while accessing other exchange.	
0x0407430D	Error in Common Parameter Block No. 5002	
0x0407430E	Error in Common Parameter Block No. 5003	
0x0407430F	Error in Common Parameter Block No. 5NM0	
0x04074310	Error in Common Parameter Block No. 5NM1	
0x04074311	Error in Common Parameter Block No. 5NM2	
0x04074312	Error in Common Parameter Block No. 5NM3	
0x04074313	Error in Common Parameter Block No. 6000	
0x04074314	Error in Common Parameter Block No. FF18 Link parameter Capacity is not set.	
0x04074315	Error in Common Parameter Block No. FF25 Calculation circuit check is not set.	
0x04074316	Error in Common Parameter Block No. FF30 Sampling Trace Data is not created.	
0x04074317	Error in Common Parameter Block No. FF31 Status latch data is not created.	
0x04074318	Error in Common Parameter Block No. FF42 Timer processing points are not set.	

Error code	Error description	Corrective action
0x04074319	Error in Common Parameter Block No. FF30 Setting value device for specified extended timer does not exist.	• Exit the program and restart the personal computer. • Reinstall MX Component.
0x0407431A	Error in Common Parameter Block No. FF44	
0x0407431B	Error in Common Parameter Block No. FF45	
0x0407431C	Error in Common Parameter Block No. FF60 Terminal Settings are not set.	
0x0407431D	Error in Common Parameter Block No. FF70 User Release area is not set.	
0x04074401	Error in Conversion of CPU Parameter Block	
0x04074402	Error in CPU Parameter Block No.1001	
0x04074403	Error in CPU Parameter Block No.1003	
0x04074404	Error in CPU Parameter Block No.1008	
0x04074405	Error in CPU Parameter Block No.1100	
0x04074406	Error in CPU Parameter Block No.2001	
0x04074407	Error in CPU Parameter Block No.3000	
0x04074408	Error in CPU Parameter Block No.3002	
0x04074409	Error in CPU Parameter Block No.3004	
0x0407440A	Error in CPU Parameter Block No.4000	
0x0407440B	Error in CPU Parameter Block No.5000 The specified network type is not supported.	
0x0407440C	Error in CPU Parameter Block No.5001	
0x0407440D	Error in CPU Parameter Block No.5002	
0x0407440E	Error in CPU Parameter Block No.5003	
0x0407440F	Error in CPU Parameter Block No. 5NM0 The specified network type is not supported.	
0x04074410	Error in CPU Parameter Block No. 5NM1	
0x04074411	Error in CPU Parameter Block No. 5NM2 The specified network type is not supported.	
0x04074412	Error in CPU Parameter Block No. 5NM3	
0x04074413	Error in CPU Parameter Block No. 6000	
0x04074414	Error in CPU Parameter Block No. FF18	
0x04074415	Error in CPU Parameter Block No. FF25	
0x04074416	Error in CPU Parameter Block No. FF30	
0x04074417	Error in CPU Parameter Block No. FF31	
0x04074418	Error in CPU Parameter Block No. FF42	
0x04074419	Error in CPU Parameter Block No. FF43	
0x0407441A	Error in CPU Parameter Block No. FF44	
0x0407441B	Error in CPU Parameter Block No. FF45	
0x0407441C	Error in CPU Parameter Block No. FF60	
0x0407441D	Error in CPU Parameter Block No. FF70	
0x04075001	Common Data Conversion error Failed while converting the device memory settings portion.	
0x04075002	Common Data Conversion error Failed while converting the device memory data portion.	
0x04075003	Common Data Conversion error Device memory data portion did not exist.	

Error code	Error description	Corrective action
0x04075101	CPU Data Conversion error Failed while converting the settings portion of the device memory.	• Exit the program and restart the personal computer. • Reinstall MX Component.
0x04075102	CPU Data Conversion error Failed while converting the data portion of the device memory.	
0x04076001	Common Data Conversion error Failed while converting the settings portion of the device comments.	
0x04076002	Common Data Conversion error Failed while converting the data portion of the device comments.	
0x04076101	CPU Data Conversion error Failed while converting the settings portion of the device comments.	
0x04076102	CPU Data Conversion error Failed while converting the settings portion of the device comments.	
0x04077001	Common Data Conversion error Failed during the conversion of sampling trace settings portion.	
0x04077002	Common Data Conversion error Failed during the conversion of sampling trace data portion.	
0x04077101	CPU Data Conversion error Failed during the conversion of sampling trace settings portion.	
0x04077102	CPU Data Conversion error Failed during the conversion of sampling trace data portion.	
0x04078001	Common Data Conversion error Failed in the conversion of the status latch settings portion.	
0x04078002	Common Data Conversion error Failed in the conversion of the status latch data portion.	
0x04078101	CPU Data Conversion error Failed in the conversion of the status latch settings portion.	
0x04078102	CPU Data Conversion error Failed in the conversion of the status latch data portion.	
0x04079101	Failure history CPU Data Conversion error	
0x0407A101	File List CPU Data Conversion error	
0x0407B101	Error Information CPU Data Conversion error	
0x0407C001	Error in Conversion of Indirect Address to Device Name The device name storage area is not secured.	
0x0407C002	Error in Conversion of Device Name to Indirect Address Indirect Address storage area is not secured.	
0x0407C003	Error in Conversion of Indirect Address to Device Representation The device representation storage area is not secured.	
0x0407C004	Error in Conversion of Device Representation to Indirect Address Indirect Address storage area is not secured.	

Error code	Error description	Corrective action
0x0407C005	Error in Conversion of Indirect Address to Device Character String Device Character String storage area is not secured.	• Exit the program and restart the personal computer. • Reinstall MX Component.
0x0407C006	Error in Conversion of Device Character String to Indirect Address Indirect Address storage area is not secured.	
0x0407C007	Error in Conversion of Intermediate Code to Device Name Device Name storage area is not secured.	
0x0407C008	Error in Conversion of Device Name to Intermediate Code Intermediate Code storage area is not secured.	
0x0407C009	Error in Conversion of Intermediate Code to Device representation Device Representation storage area is not secured.	
0x0407C00A	Error in Conversion of Device Representation to Intermediate Code Intermediate Code storage area is not secured.	
0x0407C00B	Error in Conversion of Intermediate Code to Indirect Address Indirect Address storage area is not secured.	
0x0407C00C	Error in Conversion of Indirect Address to Intermediate Code Intermediate Code storage area is not secured.	
0x0407C00D	CPU Type error The specified CPU type is not supported.	
0x0407C00E	Device Character String error The specified device is not supported.	
0x0407C00F	Device Character String error The specified device character string, type is incorrect.	
0x0407C010	Device error The specified device is not supported by the specified CPU.	
0x0407C011	CPU Type error The specified CPU is not supported.	
0x0407C012	Device out of Range error	
0x0407D001	Common Data Conversion error Error in Conversion of SFC trace condition settings portion.	
0x0407D002	Common Data Conversion error Error in Conversion of SFC trace condition data portion.	
0x0407D101	CPU Data Conversion error Error in Conversion of SFC trace condition settings portion.	
0x0407D102	CPU Data Conversion error Error in Conversion of SFC trace condition data portion.	
0x04080001	Intermediate Code classification out of range error The intermediate code classification specified is out of range.	
0x04080002	Extended specification Intermediate Code classification out of range error The extended specification intermediate code specified is out of range.	
0x04080003	Device Points check absent error The device does not check the device points.	

Error code	Error description	Corrective action
0x04090001	GPP Project error The specified CPU type and GPP project type are not matching.	• Exit the program and restart the personal computer. • Reinstall MX Component.
0x04090002	File Type error The specified GPP project type and file type are not matching.	
0x04090010	Insufficient GPP Data to be converted There is no data to be converted. The data size specified is incorrect.	
0x04090011	Insufficient Storage Space for Converted Data The space for storing converted data is insufficient.	
0x04090012	Error in GPP Data to be converted The GPP data to be converted is incorrect.	
0x04090110	Insufficient Data to be converted error There is no data to be converted. The data size specified is insufficient.	
0x04090111	Insufficient Storage Space for Converted Data error. The storage space for converted data is insufficient.	
0x04090112	Error in data to be converted The data to be converted is incorrect.	
0x04FFFFFF	Other errors	
0x10000001	No Command error	
0x10000002	Start of communication DLL of MX Component failed.	
0x10000003	Open failed. (DiskDrive)	
0x10000004	Duplex open error	• Exit the program and restart the personal computer.
0x10000005	File Access error	• Exit the program and restart the personal computer. • Reinstall MX Component.
0x10000006	Incorrect Folder Name error	
0x10000007	File Access Denied error	
0x10000008	Disk Full error	
0x10000009	File Delete error	
0x1000000A	Incorrect File Name error	
0x1000000C	Execution failed since another application or thread is making a request.	• Execute again after some time. • Perform programming according to the multithread rules of COM and ActiveX. • Exit the program and restart the personal computer.
0x1000000D	Folder Creation error	• Exit the program and restart the personal computer. • Reinstall MX Component.
0x1000000E	Folder/ File Type error	
0x1000000F	Offset Address error	
0x10000010	Request Cancel Cancel Process has occurred.	
0x10000011	Memory securing error	• Exit the program and restart the personal computer. • Reinstall MX Component.
0x10000012	Open not yet executed	• Exit the program and restart the personal computer.
0x10000013	Attach Not Executed error	• Exit the program and restart the personal computer. • Reinstall MX Component.
0x10000014	Object Invalid error	
0x10000015	Request Cancel Failed error	
0x10000016	Failed in Reading Status error	
0x10000017	The specified size (number of devices) is unauthorized.	• Check the number of points specified in the method. • Exit the program and restart the personal computer.
0x10000018	There is no registered device.	• Exit the program and restart the personal computer.

Error code	Error description	Corrective action
0x10000019	Dataset Not Executed	• Exit the program and restart the personal computer. • Reinstall MX Component.
0x1000001A	Read Not Executed error	
0x1000001B	Incorrect Create Flag error	
0x1000001C	Operation Over Access	
0x1000001D	Redundant Device error	
0x1000001E	Registry search failed.	• Exit the program and restart the personal computer. • Exit other programs and secure free memory area. • Reinstall MX Component.
0x1000001F	File Type error	• Exit the program and restart the personal computer. • Reinstall MX Component.
0x10000020	Device Memory Type error	
0x10000021	Program Range error	
0x10000022	TEL Type error	
0x10000023	TEL Access error	
0x10000024	Cancel Flag Type error	
0x10000030	Multiple Device Registration error	
0x10000031	Device Not Registered error	
0x10000032	Specified device error	• Review the specified device data. • Exit the program and restart the personal computer. • Exit other programs and secure free memory area.
0x10000033	Specified device range error	
0x10000034	File Write error	• Exit the program and restart the personal computer. • Reinstall MX Component.
0x10000040	Server start failed.	
0x10000041	Server Stop error Failed while stopping the server.	
0x10000042	Server Started Twice error	
0x10000043	Server Not Started error	
0x10000044	Resource Timeout error	
0x10000045	Server Type error	
0x10000046	Failed to Access Server error	
0x10000047	Server Already Accessed error	
0x10000048	Failed in Simulator Startup	
0x10000049	Failed in exiting Simulator	
0x1000004A	Simulator Not Started error	
0x1000004B	Simulator Type error	
0x1000004C	Simulator Not Supported error	
0x1000004D	Simulator Started Twice error	
0x1000004E	Shared Memory Not Started error	
0x80200107	Communication error	• Try the same method again. • Exit the program and restart the personal computer.
0x80200203	Memory Secure error	• Exit the program and restart the personal computer. • Exit other programs and secure free memory area.
0x80201001	Specified CPU error	• Connect to the supported motion CPU.
0x80201101	Already open error The Open method was executed in the open status.	• When changing the communication target CPU, execute the Open method after performing Close.
0x80201104	DLL load error	• Exit the program and restart the personal computer. • Exit other programs and secure free memory area. • Reinstall MX Component.
0x80201106	Error in Communication object generation	• Exit the program and restart the personal computer. • Reinstall MX Component.
0x80201201	Not yet open error The Open method is not yet executed.	• After executing the Open method, execute the corresponding method.

Error code	Error description	Corrective action
0x80201203	Memory Secure error	• Exit the program and restart the personal computer. • Exit other programs and secure free memory area.
0x80204203	Memory Secure error	• Exit the program and restart the personal computer. • Exit other programs and secure free memory area.
0x80205203	Memory Secure error	
0x80205001	Specified CPU error	• Check the CPU type set to ActCpuType. • Check that the system configuration is not an unsupported configuration.
0x80206004	Remote request error	• Change the switch on the module to RUN, exit the test mode, and try again.
0x8020A104	DLL load error	• Exit the program and restart the personal computer. • Exit other programs and secure free memory area. • Reinstall MX Component.
0x8020A203	Memory Secure error	• Exit the program and restart the personal computer. • Exit other programs and secure free memory area.
0x80A00101	Not yet open error The Open method is not yet executed.	• After executing the Open method, execute the corresponding method.
0x80A00104	Already open error The Open method was executed in the open status.	• When changing the communication target CPU, execute the Open method after performing Close.
0x80A00105	Incorrect data type of the argument	• Check the data type being used for the argument.
0x80A00106	Data range of the argument is invalid	• Check the value being used for the argument.
0x80A00107	No command error	• Not supported by the method.
0xF0000001	No-license error The license is not given to the personal computer.	• Using the license FD, give the license to the personal computer.
0xF0000002	Set data read error Reading of the set data of the logical station number failed.	• Specify the correct logical station number. • Set the logical station number on the communication settings utility.
0xF0000003	Already open error The Open method was executed in the open status.	• When changing the communication target CPU, execute the Open method after performing Close.
0xF0000004	Not yet open error The Open method is not yet executed.	• After executing the Open method, execute the corresponding method.
0xF0000005	Initialization error Initialization of the object possessed internally in MX Component failed.	• Exit the program and restart the personal computer. • Reinstall MX Component.
0xF0000006	Memory securing error Securing of MX Component internal memory failed.	• Exit the program and restart the personal computer. • Exit other programs and secure free memory area.
0xF0000007	Function non-support error The method does not support.	• Can not use because the corresponding method is not supported.
0xF1000001	Character code conversion error Character code conversion (UNICODE ASCII code or ASCII code UNICODE) failed.	• Check the character string specified in the method. • The ASCII character string acquired from the programmable controller CPU is abnormal. • Review the system, e.g. programmable controller CPU, module setting and cable status. • Exit the program and restart the personal computer. • Retry the GetCpuType method.
0xF1000002	First I/O number error The first I/O number specified is an unauthorized value. A matching first I/O number does not exist.	• Check the value of the first I/O number specified in the method. • Using the GPP function, check the programmable controller CPU parameters (I/O assignment).
0xF1000003	Buffer address error The buffer address specified is an unauthorized value. The buffer address is outside the range.	• Check the value of the buffer address specified in the method.

Error code	Error description	Corrective action
0xF1000004	Buffer read size error As a result of buffer read, the specified size could not be acquired.	• Perform reopen processing. • Review the system, e.g. programmable controller CPU, module setting and cable status. • Retry. • Exit the program.
0xF1000005	Size error The size specified in the read/write method is abnormal. The read/write first number plus size exceeds the device or buffer area.	• Check the size specified in the method.
0xF1000006	Operation error The operation specified for remote operation is an abnormal value.	• Check the operation specifying value specified in the method.
0xF1000007	Clock data error The clock data is abnormal.	• Check the clock data specified in the method. • Set the correct clock data to the clock data of the programmable controller CPU.
0xF1000008	Monitored device registration count excess The number of device points registered in the EntryDeviceStatus method was 0 or less. The number of device points registered in the EntryDeviceStatus method was more than 20.	• Register the device points between 1 and 20 in the EntryDeviceStaus method.
0xF1000009	Monitored device data registration error	• After making deregistration in the FreeDeviceStatus method, execute the EntryDeviceStatus method again.
0xF1000010	Device status monitor processing failed to start. Device status monitor processing failed to end.	• Start/end the device status monitor processing again in the EntryDeviceStatus method.
0xF1000011	The VARIANT argument data type is wrong.	• Reexamine the data type specified for the VARIANT argument. • Check whether the array variable size is large enough. • Check whether the data type specified in the corresponding method has been set.
0xF1000012	The device status monitoring time interval is a value outside the range 1 second to 1 hour (1 to 3600).	• Specify the device status monitoring time between 1 and 3600.
0xF1000013	Already Connected error. Connect was executed again after it was executed for the same object.	• Execute the Connect method after executing the Disconnect method.
0xF1000014	Invalid Telephone Number error. Characters other than "0123456789-*#" that are allowed for telephone numbers are included.	• Rectify the Telephone number and try to Connect again.
0xF1000015	Exclusive Control Failure error. There was failure in the exclusive control process while executing the Connect and Disconnect method.	• In case if Connect/Disconnect method is being executed for any other object, execute the failed method (Connect/ Disconnect) again after the completion of the Connect/ Disconnect method of that object. • If the Connect/Disconnect process is in progress only for the self object, perform the following. Exit the program. Restart the personal computer. Reinstall MX Component.
0xF1000016	While connecting to the telephone line error. The telephone line is connected to some other application, other than the one using MX Component.	• Try Connecting again after disconnecting the application that is using the telephone line.

Error code	Error description	Corrective action
0xF1000017	Telephone line not connected error. Telephone line is not connected. Connect was executed and the telephone line was connected, but it got disconnected due to some reason.	• (When Connect method has failed) Execute Connect again after executing Disconnect method. • (When method other than Connect has failed) Execute Disconnect method, Execute Connect and connect to the telephone line. After connecting, execute the method that failed once again.
0xF1000018	No Telephone number error. The telephone No. is not set. The telephone No. or call back No. is not set, if the connection method is Automatic (when specifying the call back No.), call back connection (when specifying the number), or call back Request(when specifying the number).	• In case of program settings type, set the telephone No. to the property ActDialNumber. (Set the telephone No. to the properties ActDialNumber and ActCallbackNumber, if the connection method is automatic (when specifying the call back No.), call back connection (when specifying the telephone No.), or call back request (when specifying the number).) • In case of utility settings type, set the telephone No. using the wizard. (Set the telephone No. and call back No. , if the connection method is automatic (when specifying the call back No.), call back connection (when specifying the telephone No.), or call back request (when specifying the number).)
0xF1000019	Not Closed error. Disconnect was executed while in Open state.	• Try Disconnect again after executing Close.
0xF100001A	Target telephone line connection mismatch error. Connect was tried for a different telephone number using the port which is already connected to a telephone line. (When the method of connection is a callback reception, it is considered that the telephone number is different from methods of connection in other than the callback reception.)	• If you want to connect to a different telephone number, Execute Disconnect with respect to the telephone line that is already connected and executes Connect after it gets disconnected. • In case of connecting the telephone line with callback reception, use the Connect of the connection method that is executed at the earliest in the same port as callback reception.
0xF100001B	Control Type Mismatch error. An object, whose control type is different from that of the object already connected to the telephone line, tried to Connect.	• Execute Disconnect for the object currently connected to the telephone line and execute Connect once again after the telephone line gets disconnected.
0xF100001C	Not Disconnected error. When Disconnect method is executed for the object connected to the telephone line, it is found that other objects are in connected state.	• Execute Disconnect for all the Connected objects. Try Disconnect again for the object that actually performed the telephone line connection.
0xF100001D	Not Connected error. Open was executed before Connect. Or, Disconnect was executed.	• Execute Open again after executing Connect. • Or execute Disconnect again after executing Connect.
0xF100001E	Fatal error.	• Exit the program. • Restart the personal computer. • Reinstall MX Component.
0xF100001F	Open time setting error There is some difference in telephone number and the port number settings used during Connect and Open. There is some error in Connect way.	• Check the telephone number and the port number. • Check the Connect way.
0xF2000002	There is an error response from the target telephone. Causes can be the following. • Communication error has occurred.	• Check the value of the properties set in case of program settings type and check the contents of the detailed settings that were set using the wizard in case of utility settings type.

Error code	Error description	Corrective action
0xF2000003	Invalid data was received. Causes can be the following. • Incorrect data packet received due to noise.	• Retry. • Check the communication device used at the other end.
0xF2000004	There is no response from the modem. Causes can be the following. • Abnormality in the modem. • Telephone number setting mistake.	• Check the status of the modem. • Check the telephone number. • If the problem persists even after checking the above points, change the value of the properties set (Properties such as ActConnectionCDWaitTime etc. , which set the timings) in case of program settings type and change the contents of the detailed settings that were set using the wizard in case of utility settings type.
0xF2000005	There are chances that the line is not disconnected.	• Check the line.
0xF2000006	The PC modem did not receive the AT command. Causes can be the following. • Invalid AT command was specified. • Abnormality in the modem.	• Check the contents of the AT command. • Check the status of the modem.
0xF2000007	Modem did not respond properly to the standard escape command.	• Check the modem. • Confirm whether the value of the time-out is too small. (5000ms or more is recommended.)
0xF2000009	Modem does not respond properly to the line Disconnect command.	• Check the modem.
0xF200000A	Target did not receive the signal. • The Receive settings of the modem at the other end may be incorrect. • The other end may be busy. • The telephone number may be incorrect.	• Check the Receive settings of the modem at the other end. • Check if the other end is busy. • Check the telephone number.
0xF200000B	Timeout reached for the call back receive waiting time.	• Increase the call back receive waiting time ActCallbackReveptionWaitingTimeOut and execute connect again.
0xF200000C	Password of QJ71C24 units could not be resolved.	• Set the password to ActPassword property and execute the failed method again.
0xF2010001	The callback line disconnect wait time is other than 0 -180 Seconds. The callback execution delay time is other than 0 -1800 Seconds. The telephone number is more than 62 characters.	• Check whether the callback line disconnect wait time is with in 0 - 180 Seconds. • Check whether the callback execution delay time is with in 0 - 1800 Seconds. • Check whether the telephone number is less than or equal to 62 characters. • Exit the program and restart the personal computer. • Reinstall MX Component.
0xF2010002	QJ71C24 did not receive the specified connection method. Causes can be the following. • Incorrect Connection method. • Incorrect telephone number for Call back.	• Check whether the settings of QJ71C24 and the MX Component are matching.
0xF2010003	QJ71C24 does not permit the automatic connection (during fixed Call back or when the number is specified.)	• Check the settings of QJ71C24.
0xF2100005	There are chances that the line is not disconnected.	• If there is no problem with the modem or the telephone line, change the value of the properties set (Properties like ActConnectionCDWaitTime etc. , which set the timings) in case of program settings type and change the contents of the detailed settings that were set using the wizard in case of utility settings type.
0xF2100006	Modem did not receive the startup command AT.	• Change the settings of the property ActATCommand. in case of program settings type and change the command AT that were set using the wizard in case of utility settings type.

Error code	Error description	Corrective action
0xF2100007	The PC modem does not respond to the Escape command.	• If there is no problem with the modem or the telephone line, change the value of the properties set (Properties like ActConnectionCDWaitTime etc. , which set the timings) in case of program settings type and change the contents of the detailed settings that were set using the wizard in case of utility settings type.
0xF2100008	There was no response from the modem for the data sent from the PC.	• Change the value of the properties set (Properties such as ActConnectionCDWaitTime etc. , which set the timings) in case of program settings type and change the contents of the detailed settings that were set using the wizard in case of utility settings type.
0xF21000**	There is no response from the modem. Causes can be the following. • Abnormality in the modem. • Telephone number setting mistake.	• Check the status of the modem. • Check the telephone number. • If the problem persists even after checking the above points, change the value of the properties set (Properties such as ActConnectionCDWaitTime etc. , which set the timings) in case of program settings type and change the contents of the detailed settings that were set using the wizard in case of utility settings type.
0xF202****	There was a communication failure. Following causes can be considered depending on the status. • Communication time over (Break in cable, the specified port not supported, mistake in specifying the COM port) • Modem's power is switched OFF.	• Check whether the cable is broken. • Check whether the specified port is not supported. • Check whether correct COM port is set. • Check if the modem power is switched OFF. • For detailed troubleshooting, please refer to the details of the error code got after replacing the first four digits with "0x0180". eg In case of "0xF202480B", please refer to the code "0x0180480B".
0xF3000001	Label error The label character string specified in the method is an unauthorized label character string.	• Review the label name.
0xF3000002	Label Service not started	• Exit the program and restart the personal computer. • Reinstall MX Component.
0xF3000003	MMS Service not started	
0xF3000004	The specified LabelSpace is currently being used and cannot be opened.	• Please Retly.
0xFF000001	MX Component trial version error Expired MX Component trial version	• Uninstall the trial version and install the commercial version, or reinstall the trial version.

7.2 Error Codes Returned by CPUs, Modules, and Network Boards

This section explains the error codes returned by CPUs, modules, and network boards.

Point

Error codes may not be returned as described in the following (1).
Check the considerations described in the (2) first, and refer to the (1).

(1) Error code list

If a CPU, module, or network board detected an error, any of the error codes indicated in the following table is returned.

The two upper bytes denote the error detection module, and the two lower bytes denote the error code returned by the error detection module.

For error details, refer to the manual of CPU, module, or network board corresponds to the error code.

Error code	Error detection module
0x01010000 to 0x0101FFFF[*1,*2]	Motion controller CPU
0x01070000 to 0x0107FFFF[*1]	CC-Link IE Controller Network board, MELSECNET/H board, CC-Link board
0x01090000 to 0x0109FFFF[*1]	FXCPU
0x010A0000 to 0x010AFFFF[*1]	QCPU (Q mode), QSCPU
0x010B0000 to 0x010BFFFF[*1]	Q series-compatible C24
0x010C0000 to 0x010CFFFF[*1]	Q series-compatible E71
0x010D0000 to 0x010DFFFF[*1]	PC CPU module
0x010F0000 to 0x010FFFFF[*1]	GOT

*1 : Refer to "Point"

*2 : The error codes are also described in the following section.
Page 378, Section 7.1 Error Codes Returned by Controls

(2) Considerations for confirming error codes

The following are the considerations for confirming the error codes returned by CPU, module, and network board.

(a) Property setting error

If the used system configuration does not match the preset property values, the two upper bytes do not indicate the correct error detection module.

For example, when the property values of FXCPU are set to ActCpuType for QCPU (Q mode), the two upper bytes may indicate that the error detection module is FXCPU.

In such a case, perform the communication again after checking the system configuration and all preset property values.

When the ActEasyIF or ActMLEasyIF control is used, check the settings of Communication Setup Utility.

(b) When accessing another station

When accessing another station, the error code of the used relayed module (CC-Link IE Controller Network, MELSECNET/H, CC-Link, Serial communication, Ethernet module) may be entered to the two lower bytes.

In such a case, the two upper bytes that indicate the error detection module may not always match the faulty module. Check the system configuration, and check the manuals of the used CPU, relayed network module, and network board.

7.3 HRESULT Type Error Codes

Normally, the ActiveX control and the ACT control return HRESULT type returned values.
When the custom interface is used, the returned value is equivalent to the returned value of method API.
When the dispatch interface is used, HRESULT type returned values can be acquired by performing the exception processing.
The following table shows HRESULT type returned values of ACT controls.

Returned value	Termination status	Description
S_OK	Normal termination	The function processing is normally terminated.
S_FALSE	Normal termination	The function processing (as ActiveX control) is normally terminated, but the operation (access to programmable controller) failed.
E_POINTER	Abnormal termination	The pointer passed to the function is abnormal.
E_OUTOFMEMORY	Abnormal termination	Memory reservation or object creation failed.
E_FAIL	Abnormal termination	An indefinite error occurred.

Point

If the exception processing to acquire the HRESULT type returned value is not performed, the dispatch interface displays the error dialog box on the operating system level when E_POINTER (E_XXXXX defined returned value) or the like is returned from the ACT control.

7.4 Error Codes Displayed on Event Viewer

This section explains the error codes displayed on Windows® Event Viewer by MX Component.

MX Component uses MMS (system label database) Service and Label Service to manage label information.
When an error occurs in Label Service, an error description is displayed on the system log in Event Viewer.
For errors regarding Label Service of MX Component, "MXLabelService" is displayed on the source field in Event Viewer.

The following table shows the errors and their corrective actions.

Event ID	Error message	Corrective action
3	MMS Service not started.	• Wait for the completion of MMS (system label database) Service startup. If MMS Service startup does not complete, exit the program and restart the personal computer. • Reinstall MX Component.
5	MXLabelService error (%1,%2) Please perform one of the following operations. - Restart the personal computer. - Reinstall the application and try again.	• Exit the program and restart the personal computer. • Reinstall MX Component.

APPENDIX

Appendix 1 Connection System of Callback Function

This section explains the connection system of the callback function for modem communication using Q series-compatible C24.

The callback function enables access from MX Component to a programmable controller CPU by the line reconnection (callback) performed from Q series-compatible C24 after the line connection from MX Component.

- For (1) to (3), the telephone use fees are charged on the personal computer side.
- For (4) to (8), the telephone use fees are charged on Q series-compatible C24 side.

For details of the callback function, refer to the following manual.

📖Q Corresponding Serial Communication Module User's Manual (Application)

(1) Auto line connect

Select this when the callback function is not set to Q series-compatible C24.

(2) Auto line connect (Callback fixation)

Connect the line without using the callback function when the callback function is set to Q series-compatible C24. Only personal computers whose telephone number is registered in the buffer memory (2101H) of Q series-compatible C24 can be connected.

(3) Auto line connect (Callback number specification)

Connect the line without using the callback function when the callback function is set to Q series-compatible C24. Only personal computers whose telephone number is specified by MX Component can be connected.

(4) Callback connect (Fixation)

Callback only personal computers whose telephone number is registered in the buffer memory (2101H) of Q series-compatible C24.

(5) Callback connect (Number specification)

Callback only personal computers whose telephone number is specified by MX Component.

(6) Callback request (Fixation)

Send a callback request from the selected personal computer to callback personal computers whose telephone number is registered in the buffer memory (2101H) of Q series-compatible C24. (The callback is not performed on the personal computer which is connected first.)

(7) Callback request (Number specification)

Send a callback request from the selected personal computer to callback personal computers whose telephone number is specified by MX Component. (The callback is not performed on the personal computer which is connected first.)

(8) Callback reception waiting

When connecting a line with callback request (fixation, number specification), "Callback reception waiting" is selected on the callback target personal computer to connect the line.

Appendix 2 Programming Examples for Monitoring Word Device Status

This section explains the programming examples to monitor word devices for negative values using the EntryDeviceStatus function.

(1) When using Visual Basic® .NET

The following is a programming example to monitor D0 for -10, D1 for 0, and D2 for 10 using Visual Basic® 6.0.

```
Dim szDevice As String                    'Checked device list
Dim lInputData(2) As Long                 'Set value
Dim lEntryData(2) As Long                 'Value set to argument of EntryDeviceStatus
Dim lReturnCode As Long                   'Returned value to EntryDeviceStatus
Dim lCount As Long                        'Loop counter
'Sets D0, D1 and D2 to the checked device list.
szDevice = "D0" + vbLf + "D1" + vbLf + "D2"
'Sets the checked device value "-10" for D0.
lInputData(0) = -10
'Sets the checked device value "0" for D1.
 lInputData(1) = 0
'Sets the checked device value "10" for D2.
lInputData(2) = 10
'If the set value is negative, stores "0"s into the upper 2 bytes
'for conversion into the value to be set to EntryDeviceStatus.
'Loops through the number of device points.
 For lCount = 0 To 2
     'If the set value is negative
        If lInputData(lCount) < 0 Then
           'Masks with 65535 (0000FFFF[hex]) to store "0"s into upper 2 bytes.
           lEntryData(lCount) = lInputData(lCount) And 65535
        Else
           'If the set value is positive, assigns the value as-is to lEntryData.
           lEntryData(lCount) = lInputData(lCount)
        End If
 Next
'Executes EntryDeviceStatus.
lReturnCode = AxActUtlType1.EntryDeviceStatus(szDevice, 3, 5, lEntryData(0))
```

<When ldata = -1>

```
Private Sub AxActUtlType1_OnDeviceStatus(ByVal szDevice As String, ByVal lData As Long, ByVal lReturnCode As Long)
Dim lCheckData As Long              'Value set to EntryDeviceStatus (value before 0s are stored into the upper 2 bytes)
'If the device value whose condition was established is a WORD type negative value (greater than 32767 (7FFF[Hex])
     If lData > 32767 Then
          'Since "0"s are stored in the upper 2 bytes, the device value is ORed with FFFF0000[Hex] to convert it into a LONG
           'type negative value.
           lCheckData = lData Or &HFFFF0000
     Else
           'If the device value whose condition was established is positive, assigns the value as-is to lCheckData.
           lCheckData = lData
     End If
End Sub
```

(2) When using Visual C++® .NET

The following is a programming example to monitor D0 for -10, D1 for 0, and D2 for 10 using Visual C++® .NET.

```
CString szDevice;                    //Checked device list
LONG lInputData[3];                  //Set value
LONG lEntryData[3];                  //Value set to argument of EntryDeviceStatus
LONG lReturnCode;                    //Returned value to EntryDeviceStatus
LONG lCount;                         //Loop counter
//Sets D0, D1 and D2 to the checked device list.
szDevice = "D0\nD1\nD2";
// Sets the checked device value "-10" for D0.
lInputData[0] = -10;
// Sets the checked device value "0" for D1.
lInputData[1] = 0;
// Sets the checked device value "10" for D2.
lInputData[2] = 10;

//If the set value is negative, stores "0"s into the upper 2 bytes
//for conversion into the value to be set to EntryDeviceStatus.
//Loops through the number of device points.
for(lCount = 0;lCount<=2; lCount++) {
    //If the set value is negative
    if (lInputData[lCount] < 0 ){
        //Masks with 65535 (0000FFFF[hex]) to store "0"s into upper 2 bytes.
        lEntryData[lCount] = lInputData[lCount] & 0x0000FFFF;
    }else{
        //If the set value is positive, assigns the value as-is to lEntryData.
        lEntryData[lCount] = lInputData[lCount];
    }
}

//Executes EntryDeviceStatus.
lReturnCode = m_Actutltype.EntryDeviceStatus(szDevice,3,5,lEntryData);
```

<When lData = -1>

```
void CSampleDlg::OnDeviceStatusActutltype1(LPCTSTR szDevice, long lData, long lReturnCode)
{
LONG lCheckData;                     //Value set to EntryDeviceStatus
    //If the device value whose condition was established is a WORD type negative value
    //greater than 32767 (7FFF[Hex])
    if(lData > 0x7FFF){
        //Since "0"s are stored in the upper 2 bytes, the device value is ORed
        //with FFFF0000[Hex] to convert it into a LONG type negative value.
        lCheckData = lData | 0xFFFF0000;
    }else{
    //If the device value whose condition was established is positive,
    //assigns the value as-is to lCheckData.
    lCheckData = lData;
    }
}
```

Appendix 3 Time-Out Periods

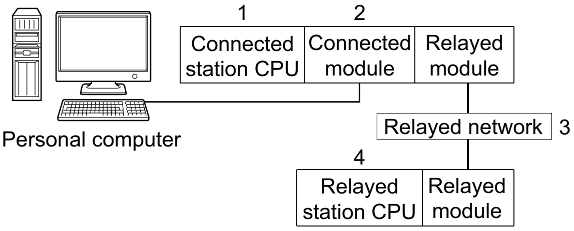
In MX Component, a time-out may occur at the period different from the value set to the ActTimeOut property in the ACT control.
This section explains the time-out periods in various status.

Appendix 3.1 Communication retries at time-out error occurrence

If a time-out error occurs during communication, the time-out processing may be repeated for a maximum of three times in the ACT control.
At that time, the process takes a maximum of three times longer period than the set time-out value which is a period until the time-out occurrence.
The following shows communication paths for retries at a time-out error occurrence.

(1) Serial communication

(a) Configuration



(b) Target/non-target communication path

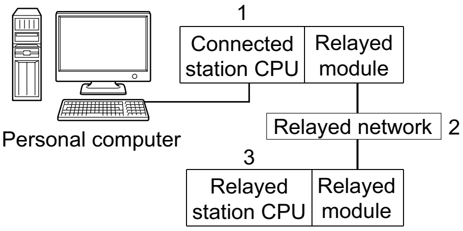
The following table shows the communication paths for retries at a time-out error occurrence.
All the connected station CPUs can be the targets.
Target/non-target relayed CPUs are indicated by ○ (target) or × (non-target).

Connected station		3. Relayed network	4. Relayed station CPU					
1. CPU	2. Connected module		QCPU (Q mode)	Q12DC CPU-V	LCPU	QS CPU	Q motion CPU	FX CPU
QCPU (Q mode)	Q series-compatible C24	CC IE Field	○	○	○	○	×	×
		CC IE Control	○	○	×	○	×	×
		MELSECNET/H	○	○	×	○	×	×
		Ethernet	○	×	×	○	×	×
		Serial communication	○	×	○	×	×	×
		CC-Link	○	○	○	×	○	○
		Multi-drop connection	○	×	○	×	×	×

Connected station		3. Relayed network	4. Relayed station CPU					
1. CPU	2. Connected module		QCPU (Q mode)	Q12DC CPU-V	LCPU	QS CPU	Q motion CPU	FX CPU
LCPU	L series-compatible C24	CC IE Field	○	×	○	×	×	×
		CC IE Control	×	×	×	×	×	×
		MELSECNET/H	×	×	×	×	×	×
		Ethernet	×	×	×	×	×	×
		Serial communication	○	×	○	×	×	×
		CC-Link	○	○	○	×	×	×
		Multi-drop connection	○	×	○	×	×	×
FXCPU	FX extended port	CC IE Field	×	×	×	×	×	×
		CC IE Control	×	×	×	×	×	×
		MELSECNET/H	×	×	×	×	×	×
		Ethernet	×	×	×	×	×	×
		Serial communication	×	×	×	×	×	○
		CC-Link	×	×	×	×	×	×
		Multi-drop connection	×	×	×	×	×	○

(2) CPU COM communication

(a) Configuration



(b) Target/non-target communication path

The following table shows the communication paths for retries at a time-out error occurrence.

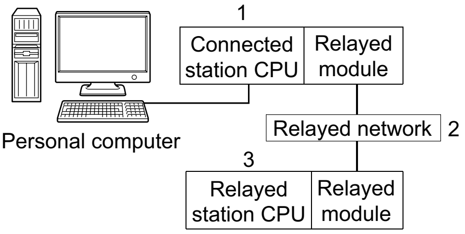
All the connected station CPUs can be the targets.

Target/non-target relayed CPUs are indicated by ○ (target) or × (non-target).

1. Connected module	2. Relayed network	3. Relayed station CPU					
		QCPU (Q mode)	Q12DC CPU-V	LCPU	QS CPU	Q motion CPU	FX CPU
QCPU (Q mode)	CC IE Field	○	○	○	○	×	×
	CC IE Control	○	○	×	○	×	×
	MELSECNET/H	○	○	×	○	×	×
	Ethernet	○	×	×	○	×	×
	Serial communication	○	×	○	×	×	×
	CC-Link	○	○	○	×	○	○
LCPU	CC IE Field	○	×	○	×	×	×
	CC IE Control	×	×	×	×	×	×
	MELSECNET/H	×	×	×	×	×	×
	Ethernet	×	×	×	×	×	×
	Serial communication	○	×	○	×	×	×
	CC-Link	○	○	○	×	×	×
Q motion CPU	CC IE Field	×	×	×	×	×	×
	CC IE Control	×	×	×	×	×	×
	MELSECNET/H	×	×	×	×	×	×
	Ethernet	×	×	×	×	×	×
	Serial communication	×	×	×	×	×	×
	CC-Link	×	×	×	×	×	×
FXCPU	CC IE Field	×	×	×	×	×	×
	CC IE Control	×	×	×	×	×	×
	MELSECNET/H	×	×	×	×	×	×
	Ethernet	×	×	×	×	×	×
	Serial communication	×	×	×	×	×	×
	CC-Link	×	×	×	×	×	×

(3) CPU USB communication

(a) Configuration



(b) Target/non-target communication path

The following table shows the communication paths for retries at a time-out error occurrence.

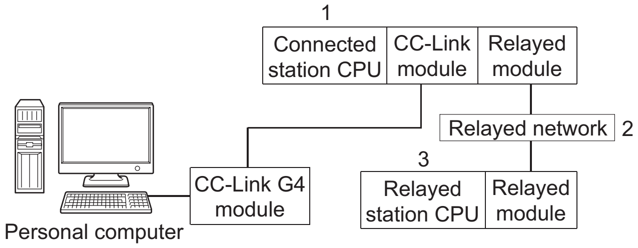
All the connected station CPUs can be the targets.

Target/non-target relayed CPUs are indicated by ○ (target) or × (non-target).

1. Connected module	2. Relayed network	3. Relayed station CPU					
		QCPU (Q mode)	Q12DC CPU-V	LCPU	QS CPU	Q motion CPU	FX CPU
QCPU (Q mode)	CC IE Field	○	○	○	○	×	×
	CC IE Control	○	○	×	○	×	×
	MELSECNET/H	○	○	×	○	×	×
	Ethernet	○	×	×	○	×	×
	Serial communication	○	×	○	×	×	×
	CC-Link	○	○	○	×	○	×
Q12DCCPU-V	CC IE Field	○	○	○	○	×	×
	CC IE Control	○	○	×	○	×	×
	MELSECNET/H	○	○	×	○	×	×
	Ethernet	×	×	×	×	×	×
	Serial communication	×	×	×	×	×	×
	CC-Link	○	○	○	×	○	×
LCPU	CC IE Field	○	×	○	×	×	×
	CC IE Control	×	×	×	×	×	×
	MELSECNET/H	×	×	×	×	×	×
	Ethernet	×	×	×	×	×	×
	Serial communication	○	×	○	×	×	×
	CC-Link	○	○	○	×	×	×
FXCPU	CC IE Field	×	×	×	×	×	×
	CC IE Control	×	×	×	×	×	×
	MELSECNET/H	×	×	×	×	×	×
	Ethernet	×	×	×	×	×	×
	Serial communication	×	×	×	×	×	×
	CC-Link	×	×	×	×	×	×

(4) CC-Link G4 communication

(a) Configuration



(b) Target/non-target communication path

The following tables show the communication paths for retries at a time-out error occurrence.

All the connected station CPUs can be the targets.

Target/non-target relayed CPUs are indicated by ○ (target) or × (non-target).

①When CC-Link G4-S3 module is Q mode and connected station CPU is QCPU (Q mode) or Q12DCCPU-V

1. Connected module	2. Relayed network	3. Relayed station CPU					
		QCPU (Q mode)	Q12DC CPU-V	LCPU	QS CPU	Q motion CPU	FX CPU
QCPU (Q mode)	CC IE Field	○	○	○	○	×	×
	CC IE Control	○	○	×	○	×	×
	MELSECNET/H	○	○	×	○	×	×
	Ethernet	○	×	×	○	×	×
	Serial communication	○	×	×	×	×	×
	CC-Link	○	×	×	×	○	×
Q12DCCPU-V	CC IE Field	○	○	○	○	×	×
	CC IE Control	○	○	×	○	×	×
	MELSECNET/H	○	○	×	○	×	×
	Ethernet	×	×	×	×	×	×
	Serial communication	×	×	×	×	×	×
	CC-Link	×	×	×	×	×	×

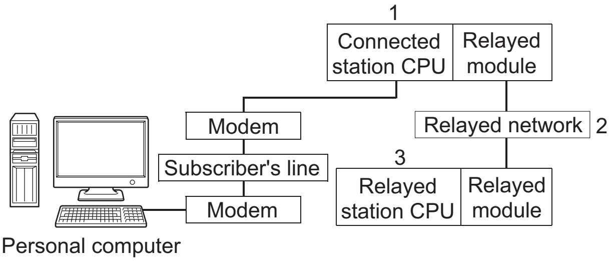
②When CC-Link G4-S3 module is Q mode and connected station CPU is LCPU

1. Connected module	2. Relayed network	3. Relayed station CPU					
		QCPU (Q mode)	Q12DC CPU-V	LCPU	QS CPU	Q motion CPU	FX CPU
LCPU	CC IE Field	×	×	×	×	×	×
	CC IE Control	×	×	×	×	×	×
	MELSECNET/H	×	×	×	×	×	×
	Ethernet	×	×	×	×	×	×
	Serial communication	×	×	×	×	×	×
	CC-Link	×	×	×	×	×	×

(5) Modem communication

(a) When using FXCPU

① Configuration



② Target/non-target communication path

The following table shows the communication paths for retries at a time-out error occurrence.

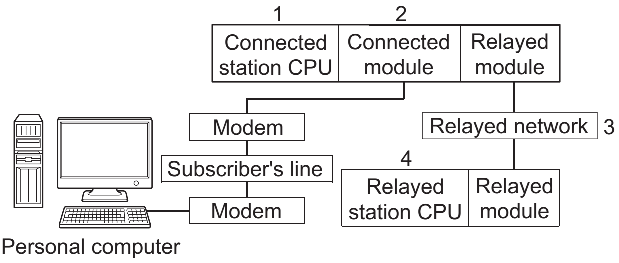
All the connected station CPUs can be the targets.

Target/non-target relayed CPUs are indicated by ○ (target) or × (non-target).

1. Connected module	2. Relayed network	3. Relayed station CPU					
		QCPU (Q mode)	Q12DC CPU-V	LCPU	QS CPU	Q motion CPU	FX CPU
FXCPU	CC IE Field	×	×	×	×	×	×
	CC IE Control	×	×	×	×	×	×
	MELSECNET/H	×	×	×	×	×	×
	Ethernet	×	×	×	×	×	×
	Serial communication	×	×	×	×	×	×
	CC-Link	×	×	×	×	×	×

(b) When using Q series-compatible C24 or QC24N

① Configuration



② Target/non-target communication path

The following table shows the communication paths for retries at a time-out error occurrence.

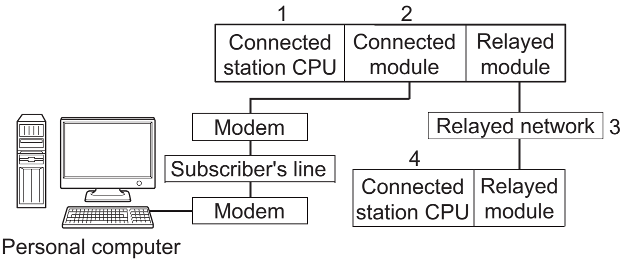
All the connected station CPUs can be the targets.

Target/non-target relayed CPUs are indicated by ○ (target) or × (non-target).

Connected station		3. Relayed network	4. Relayed station CPU					
1. CPU	2. Connected module		QCPU (Q mode)	Q12DC CPU-V	LCPU	QS CPU	Q motion CPU	FX CPU
QCPU (Q mode)	Q series-compatible C24	CC IE Control CC IE Field	○	○	○	○	×	×
		CC IE Control	○	○	×	○	×	×
		MELSECNET/H	○	○	×	○	×	×
		Ethernet	○	×	×	○	×	×
		Serial communication	○	×	○	×	×	×
		CC-Link	○	○	○	×	○	×
		Multi-drop connection (Independent mode)	○	×	○	×	×	×

(c) When using L series-compatible C24

① Configuration



② Target/non-target communication path

The following table shows the communication paths for retries at a time-out error occurrence.

All the connected station CPUs can be the targets.

Target/non-target relayed CPUs are indicated by ○ (target) or × (non-target).

Connected station		3. Relayed network	4. Relayed station CPU					
1. CPU	2. Connected module		QCPU (Q mode)	Q12DC CPU-V	LCPU	QS CPU	Q motion CPU	FX CPU
LCPU	L series-compatible C24	CC IE Field	○	×	○	×	×	×
		CC IE Control	×	×	×	×	×	×
		MELSECNET/H	×	×	×	×	×	×
		Ethernet	×	×	×	×	×	×
		Serial communication	○	×	○	×	×	×
		CC-Link	○	○	○	×	×	×
		Multi-drop connection (Independent mode)	○	×	○	×	×	×

Appendix 3.2 Communication retries at receive data error occurrence

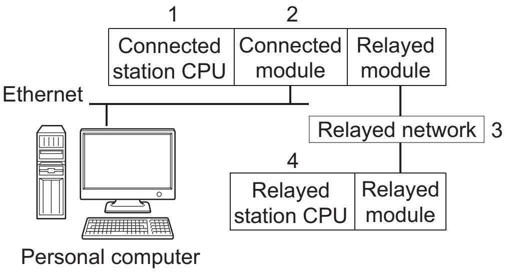
If a receive data error occurs during communication, send/receive retry processing may be repeated for a maximum of three times in the ACT control.

At that time, the process takes a maximum of three times longer period than the set time-out value which is a period until the normal or abnormal termination of the function.

The following shows communication paths for retries at a receive data error occurrence.

(1) Ethernet communication

(a) Configuration



(b) Target/non-target communication path

The following table shows the communication paths for retries at a data error occurrence.

All the connected station CPUs can be the targets.

Target/non-target relayed CPUs are indicated by ○ (target) or × (non-target).

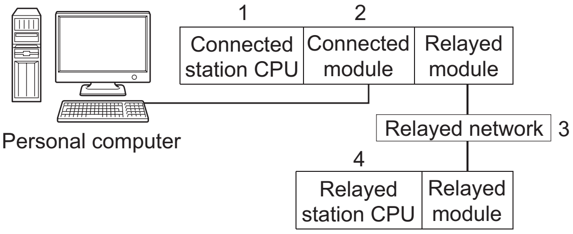
Connected station		3. Relayed network	4. Relayed station CPU					
1. CPU	2. Connected module		QCPU (Q mode)	Q12DC CPU-V	LCPU	QS CPU	Q motion CPU	FX CPU
QCPU (Q mode)	Q series-compatible E71, Built-in Ethernet port QCPU	CC IE Field	○	○	○	○	×	×
		CC IE Control	○	○	×	○	×	×
		MELSECNET/H	○	○	×	○	×	×
		Ethernet	○	×	×	○	×	×
		Serial communication	○	×	○	×	×	×
		CC-Link	○	○	○	×	×	×
LCPU	LCPU	CC IE Field	○	×	○	×	×	×
		CC IE Control	×	×	×	×	×	×
		MELSECNET/H	×	×	×	×	×	×
		Ethernet	×	×	×	×	×	×
		Serial communication	○	×	○	×	×	×
		CC-Link	○	○	○	×	×	×

Appendix 3.3 Time-out errors at fixed time in ACT control

MX Component performs communication to check whether a personal computer and a programmable controller system are connected normally before executing the Open function.
When performing the above communication, the fixed time-out period (1000ms to 4500ms) in the ACT control is used.
Note that if an error occurs during the above communication, an error other than the time-out error may occur.
The following shows communication paths for a time-out error occurrence at the fixed time in the ACT control.

(1) Serial communication

(a) Configuration



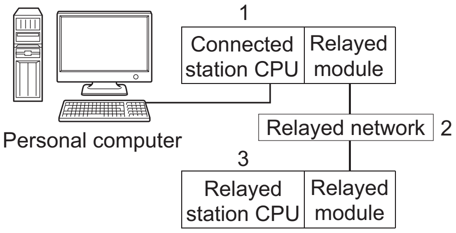
(b) Target/non-target communication path

The following table shows the communication paths for a time-out error occurrence at the fixed time in the ACT control.
All the connected station CPUs can be the targets.
Target/non-target relayed CPUs are indicated by ○ (target) or × (non-target).

Connected station		3. Relayed network	4. Relayed station CPU					
1. CPU	2. Connected module		QCPU (Q mode)	Q12DC CPU-V	LCPU	QS CPU	Q motion CPU	FX CPU
QCPU (Q mode)	Q series-compatible C24	CC IE Field	○	○	○	○	×	×
		CC IE Control	○	○	×	○	×	×
		MELSECNET/H	○	○	×	○	×	×
		Ethernet	○	×	×	○	×	×
		Serial communication	○	×	○	×	×	×
		CC-Link	○	○	○	×	○	○
		Multi-drop connection	○	×	○	×	×	×
LCPU	L series-compatible C24	CC IE Field	○	×	○	×	×	×
		CC IE Control	×	×	×	×	×	×
		MELSECNET/H	×	×	×	×	×	×
		Ethernet	×	×	×	×	×	×
		Serial communication	○	×	○	×	×	×
		CC-Link	○	○	○	×	×	×
		Multi-drop connection	○	×	○	×	×	×

(2) CPU COM communication

(a) Configuration



(b) Target/non-target communication path

The following table shows the communication paths for a time-out error occurrence at the fixed time in the ACT control.

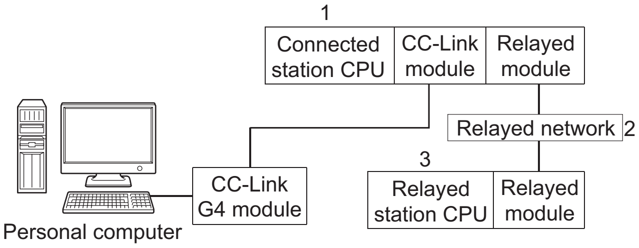
All the connected station CPUs can be the targets.

Target/non-target relayed CPUs are indicated by ○ (target) or × (non-target).

1. Connected module	2. Relayed network	3. Relayed station CPU					
		QCPU (Q mode)	Q12DC CPU-V	LCPU	QS CPU	Q motion CPU	FX CPU
QCPU (Q mode), Q12DCCPU-V	CC IE Field	○	○	○	○	×	×
	CC IE Control	○	○	×	○	×	×
	MELSECNET/H	○	○	×	○	×	×
	Ethernet	○	×	×	○	×	×
	Serial communication	○	×	○	×	×	×
	CC-Link	○	○	○	×	○	○
LCPU	CC IE Field	○	×	○	×	×	×
	CC IE Control	×	×	×	×	×	×
	MELSECNET/H	×	×	×	×	×	×
	Ethernet	×	×	×	×	×	×
	Serial communication	○	×	○	×	×	×
	CC-Link	○	○	○	×	×	×
FXCPU	CC IE Field	×	×	×	×	×	×
	CC IE Control	×	×	×	×	×	×
	MELSECNET/H	×	×	×	×	×	×
	Ethernet	×	×	×	×	×	×
	Serial communication	×	×	×	×	×	×
	CC-Link	×	×	×	×	×	×

(3) CC-Link G4 communication

(a) Configuration



(b) Target/non-target communication path

The following table shows the communication paths for a time-out error occurrence at the fixed time in the ACT control.

All the connected station CPUs can be the targets.

Target/non-target relayed CPUs are indicated by ○ (target) or × (non-target).

1. Connected module	2. Relayed network	3. Relayed station CPU					
		QCPU (Q mode)	Q12DC CPU-V	LCPU	QS CPU	Q motion CPU	FX CPU
QCPU (Q mode)	CC IE Field	○	○	○	○	×	×
	CC IE Control	○	○	×	○	×	×
	MELSECNET/H	○	○	×	○	×	×
	Ethernet	○	×	×	○	×	×
	Serial communication	×	×	×	×	×	×
	CC-Link	×	×	×	×	×	×
Q12DCCPU-V	CC IE Field	○	○	○	○	×	×
	CC IE Control	○	○	×	○	×	×
	MELSECNET/H	○	×	○	×	○	×
	Ethernet	×	×	×	×	×	×
	Serial communication	×	×	×	×	×	×
	CC-Link	×	×	×	×	×	×
LCPU	CC IE Field	×	×	×	×	×	×
	CC IE Control	×	×	×	×	×	×
	MELSECNET/H	×	×	×	×	×	×
	Ethernet	×	×	×	×	×	×
	Serial communication	×	×	×	×	×	×
	CC-Link	×	×	×	×	×	×

Appendix 4 Corrective Actions for Error Codes 0x01800002 and 0x018000003

If a memory lock error (0x01800002) or a memory securing error (0x01800003) occurs in ACT control when an interface board for personal computer is used, increase the minimum working set size of the personal computer.

The personal computer board driver runs using the minimum working set size in the memory area reserved in the user program.

Some user programs require a larger size for the minimum working set.

Therefore, if the minimum working set size for the personal computer board driver cannot be reserved, the memory lock error (0x01800002) or the memory securing error (0x01800003) is returned.

In this situation, increase the minimum working set size in the user program before executing the ACT control function. (☞Page 425, (2) in this section, Page 426, (3) in this section)

The minimum working set size of 200KB is reserved at startup of the personal computer.

(1) Processing overview of sample program

1. **Acquire a user program ID using the GetCurrentProcessID function.**
2. **Acquire a user program handle with the acquired user program ID using the OpenProcess function.**
3. **The current minimum and maximum working set sizes can be acquired by executing the GetProcessWorkingSetSize function.**
4. **Set the value larger than the acquired minimum working set size, and execute the SetProcessWorkingSetSize function.**
5. **Release the user program handle using the CloseHandle function.**

(2) Sample program: Setting working set sizes using Visual Basic® .NET

(Example: when a minimum working set size is 1MB and maximum working set size is 3MB)

Define the variables id, ph, wkmin, and wkmax as 32-bit integer type.

```
Dim id As Integer                   'User program ID variable
Dim ph As Integer                   'User program handle variable
Dim wkmin As Integer                'Minimum working set variable
Dim wkmax As Integer                'Maximum working set variable
Dim bret As Boolean                 'Return value
'Obtain the user program ID
id = GetCurrentProcessID()
'Open the user program handle
'PROCESS_SET_QUOTA = 256,PROCESS_QUERY_INFORMATION = 1024
ph = OpenProcess(256 + 1024,False,id)
'Obtain the maximum working set size and minimum working set size for the user program
bret = GetProcessWorkingSetSize(ph,wkmin,wkmax)
'Set the minimum working set size to 1MB
wkmin = 1 * 1024 * 1024
'Set the maximum working set size to 3MB
wkmax = 3 * 1024 * 1024
'Change the maximum working set size and minimum working set size for the user program
bret = SetProcessWorkingSetSize(ph,wkmin,wkmax)
'Close the user program handle
bret = CloseHandle(ph)
```

The set sizes shown above are reference sizes.

Adjust the sizes according to the system.

(3) Sample program: Setting working set sizes using Visual C++® .NET

(Example: when a minimum working set size is 1MB and maximum working set size is 3MB)

```
#define     ERROR   -1
short       ChangeWorkingSetSize()
{
DWORD       dwProcessId;                    /*User program ID variable*/
HANDLE      hProcess;                       /*User program handle variable*/
DWORD       dwMinimumWorkingSetSize;        /*Minimum working set variable*/
DWORD       dwMaximumWorkingSetSize;        /*Maximum working set variable*/
/*Obtain the user program ID*/
dwProcessId = GetCurrentProcessId();
/*Open the user program handle*/
hProcess = OpenProcess(PROCESS_SET_QUOTA+PROCESS_QUERY_INFORMATION,FALSE,dwProcessId);
if(hProcess == NULL){
            /*Error end*/
            return(ERROR);
}
/*Obtain the maximum working set size and minimum working set size for the user program*/
if(GetProcessWorkingSetSize(hProcess,&dwMinimumWorkingSetSize,&dwMaximumWorkingSetSize)==0){
            /*Error end*/
            CloseHandle(hProcess);
            return(ERROR);
}
/*Set the minimum working set size to 1MB*/
dwMinimumWorkingSetSize = 1 * 1024 * 1024;
/*Set the maximum working set size to 3MB*/
dwMaximumWorkingSetSize = 3 * 1024 * 1024;
/*Change the maximum working set size and minimum working set size for the user program*/
if(SetProcessWorkingSetSize(hProcess,dwMinimumWorkingSetSize,dwMaximumWorkingSetSize)==0){
            /*Error end*/
            CloseHandle(hProcess);
            return(ERROR);
}
/*Close the user program handle*/
CloseHandle(hProcess);
/*Normal return*/
return(0);
}
```

The set sizes shown above are reference sizes.
Adjust the sizes according to the system.

REVISIONS

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

Print date	*Manual number	Revision
Jul., 2012	SH-081085ENG-A	First edition

Japanese Manual Version SH-081083-A

Memo

WARRANTY

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

1. Gratis Warranty Term and Gratis Warranty Range

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

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

[Gratis Warranty Term]

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

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

[Gratis Warranty Range]

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

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

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

2. Onerous repair term after discontinuation of production

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

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

3. Overseas service

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

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

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

5. Changes in product specifications

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

MX Component Version 4

Programming Manual

MODEL	SW4DNC-ACT-P-E
MODEL CODE	13JW12
SH(NA)-081085ENG-A(1207)KWIX	

MITSUBISHI ELECTRIC CORPORATION

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

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

Specifications subject to change without notice.

MITSUBISHI ELECTRIC

HEADQUARTERS

MITSUBISHI ELECTRIC EUROPE B.V. **EUROPE**
German Branch
Gothaer Straße 8
D-40880 Ratingen
Phone: +49 (0)2102 / 486-0
Fax: +49 (0)2102 / 486-1120

MITSUBISHI ELECTRIC EUROPE B.V.-org.sl. **CZECH REP.**
Czech Branch
Avenir Business Park, Radlická 714/113a
CZ-158 00 Praha 5
Phone: +420 - 251 551 470
Fax: +420 - 251-551-471

MITSUBISHI ELECTRIC EUROPE B.V. **FRANCE**
French Branch
25, Boulevard des Bouvets
F-92741 Nanterre Cedex
Phone: +33 (0)1 / 55 68 55 68
Fax: +33 (0)1 / 55 68 57 57

MITSUBISHI ELECTRIC EUROPE B.V. **IRELAND**
Irish Branch
Westgate Business Park, Ballymount
IRL-Dublin 24
Phone: +353 (0)1 4198800
Fax: +353 (0)1 4198890

MITSUBISHI ELECTRIC EUROPE B.V. **ITALY**
Italian Branch
Viale Colleoni 7
I-20864 Agrate Brianza (MB)
Phone: +39 039 / 60 53 1
Fax: +39 039 / 60 53 312

MITSUBISHI ELECTRIC EUROPE B.V. **POLAND**
Poland Branch
Krakowska 50
PL-32-083 Balice
Phone: +48 (0)12 / 630 47 00
Fax: +48 (0)12 / 630 47 01

MITSUBISHI ELECTRIC EUROPE B.V. **RUSSIA**
52, bld. 3 Kosmodamianskaya nab 8 floor
RU-115054 Moscow
Phone: +7 495 721-2070
Fax: +7 495 721-2071

MITSUBISHI ELECTRIC EUROPE B.V. **SPAIN**
Spanish Branch
Carretera de Rubí 76-80
E-08190 Sant Cugat del Vallés (Barcelona)
Phone: 902 131121 // +34 935653131
Fax: +34 935891579

MITSUBISHI ELECTRIC EUROPE B.V. **UK**
UK Branch
Travellers Lane
UK-Hatfield, Herts. AL10 8XB
Phone: +44 (0)1707 / 27 61 00
Fax: +44 (0)1707 / 27 86 95

MITSUBISHI ELECTRIC CORPORATION **JAPAN**
Office Tower "Z" 14 F
8-12,1 chome, Harumi Chuo-Ku
Tokyo 104-6212
Phone: +81 3 622 160 60
Fax: +81 3 622 160 75

MITSUBISHI ELECTRIC AUTOMATION, Inc. **USA**
500 Corporate Woods Parkway
Vernon Hills, IL 60061
Phone: +1 847 478 21 00
Fax: +1 847 478 22 53

EUROPEAN REPRESENTATIVES

GEVA **AUSTRIA**
Wiener Straße 89
AT-2500 Baden
Phone: +43 (0)2252 / 85 55 20
Fax: +43 (0)2252 / 488 60

TECHNIKON **BELARUS**
Oktyabrskaya 19, Off. 705
BY-220030 Minsk
Phone: +375 (0)17 / 210 46 26
Fax: +375 (0)17 / 210 46 26

ESCO DRIVES & AUTOMATION **BELGIUM**
Culliganlaan 3
BE-1831 Diegem
Phone: +32 (0)2 / 717 64 30
Fax: +32 (0)2 / 717 64 31

Koning & Hartman b.v. **BELGIUM**
Woluwelaan 31
BE-1800 Vilvoorde
Phone: +32 (0)2 / 257 02 40
Fax: +32 (0)2 / 257 02 49

INEA RBT d.o.o. **BOSNIA AND HERZEGOVINA**
Aleja Lipa 56
BA-71000 Sarajevo
Phone: +387 (0)33 / 921 164
Fax: +387 (0)33/ 524 539

AKHNATON **BULGARIA**
4, Andrei Ljapchev Blvd., PO Box 21
BG-1756 Sofia
Phone: +359 (0)2 / 817 6000
Fax: +359 (0)2 / 97 44 06 1

INEA RBT d.o.o. **CROATIA**
Losinjska 4 a
HR-10000 Zagreb
Phone: +385 (0)1 / 36 940 - 01/ -02/ -03
Fax: +385 (0)1 / 36 940 - 03

AutoCont C.S. s.r.o. **CZECH REPUBLIC**
Technologická 374/6
CZ-708 00 Ostrava-Pustkovec
Phone: +420 595 691 150
Fax: +420 595 691 199

Beijer Electronics A/S **DENMARK**
Lykkegårdsvej 17
DK-4000 Roskilde
Phone: +45 (0)46/ 75 76 66
Fax: +45 (0)46 / 75 56 26

Beijer Electronics Eesti OÜ **ESTONIA**
Pärnu mnt.160i
EE-11317 Tallinn
Phone: +372 (0)6 / 51 81 40
Fax: +372 (0)6 / 51 81 49

Beijer Electronics OY **FINLAND**
Peltoie 37
FIN-28400 Ulvila
Phone: +358 (0)207 / 463 540
Fax: +358 (0)207 / 463 541

UTECO **GREECE**
5, Mavrogenous Str.
GR-18542 Piraeus
Phone: +30 211 / 1206 900
Fax: +30 211 / 1206 999

MELTRADE Kft. **HUNGARY**
Fertő utca 14.
HU-1107 Budapest
Phone: +36 (0)1 / 431-9726
Fax: +36 (0)1 / 431-9727

Beijer Electronics SIA **LATVIA**
Ritausmas iela 23
LV-1058 Riga
Phone: +371 (0)784 / 2280
Fax: +371 (0)784 / 2281

Beijer Electronics UAB **LITHUANIA**
Savanoriu Pr. 187
LT-02300 Vilnius
Phone: +370 (0)5 / 232 3101
Fax: +370 (0)5 / 232 2980

EUROPEAN REPRESENTATIVES

ALFATRADE Ltd. **MALTA**
99, Paola Hill
Malta- Paola PLA 1702
Phone: +356 (0)21 / 697 816
Fax: +356 (0)21 / 697 817

INTEHSIS srl **MOLDOVA**
bld. Traian 23/1
MD-2060 Kishinev
Phone: +373 (0)22 / 66 4242
Fax: +373 (0)22 / 66 4280

HIFLEX AUTOM.TECHNIEK B.V. **NETHERLANDS**
Wolweverstraat 22
NL-2984 CD Ridderkerk
Phone: +31 (0)180 – 46 60 04
Fax: +31 (0)180 – 44 23 55

Koning & Hartman b.v. **NETHERLANDS**
Haarlerbergweg 21-23
NL-1101 CH Amsterdam
Phone: +31 (0)20 / 587 76 00
Fax: +31 (0)20 / 587 76 05

Beijer Electronics AS **NORWAY**
Postboks 487
NO-3002 Drammen
Phone: +47 (0)32 / 24 30 00
Fax: +47 (0)32 / 84 85 77

Fonseca S.A. **PORTUGAL**
R. João Francisco do Casal 87/89
PT - 3801-997 Aveiro, Esgueira
Phone: +351 (0)234 / 303 900
Fax: +351 (0)234 / 303 910

Sirius Trading & Services srl **ROMANIA**
Aleea Lacul Morii Nr. 3
RO-060841 Bucuresti, Sector 6
Phone: +40 (0)21 / 430 40 06
Fax: +40 (0)21 / 430 40 02

INEA RBT d.o.o. **SERBIA**
Izletnicka 10
SER-113000 Smederevo
Phone: +381 (0)26 / 615 401
Fax: +381 (0)26 / 615 401

SIMAP s.r.o. **SLOVAKIA**
Jána Derku 1671
SK-911 01 Trencín
Phone: +421 (0)32 743 04 72
Fax: +421 (0)32 743 75 20

PROCONT, spol. s r.o. Prešov **SLOVAKIA**
Kúpelná 1/A
SK-080 01 Prešov
Phone: +421 (0)51 7580 611
Fax: +421 (0)51 7580 650

INEA RBT d.o.o. **SLOVENIA**
Stegne 11
SI-1000 Ljubljana
Phone: +386 (0)1 / 513 8116
Fax: +386 (0)1 / 513 8170

Beijer Electronics AB **SWEDEN**
Box 426
SE-20124 Malmö
Phone: +46 (0)40 / 35 86 00
Fax: +46 (0)40 / 93 23 01

Omni Ray AG **SWITZERLAND**
Im Schörli 5
CH-8600 Dübendorf
Phone: +41 (0)44 / 802 28 80
Fax: +41 (0)44 / 802 28 28

GTS **TURKEY**
Bayraktar Bulvari Nutuk Sok. No:5
TR-34775 Yukarı Dudullu-Ümraniye-İSTANBUL
Phone: +90 (0)216 526 39 90
Fax: +90 (0)216 526 3995

CSC Automation Ltd. **UKRAINE**
4-B, M. Raskovoyi St.
UA-02660 Kiev
Phone: +380 (0)44 / 494 33 55
Fax: +380 (0)44 / 494-33-66

EURASIAN REPRESENTATIVES

TOO Kazpromavtomatika **KAZAKHSTAN**
Ul. Zhambyla 28
KAZ-100017 Karaganda
Phone: +7 7212 / 50 10 00
Fax: +7 7212 / 50 11 50

MIDDLE EAST REPRESENTATIVES

I.C. SYSTEMS LTD. **EGYPT**
23 Al-Saad-Al-Alee St.
EG-Sarayat, Maadi, Cairo
Phone: +20 (0) 2 / 235 98 548
Fax: +20 (0) 2 / 235 96 625

ILAN & GAVISH Ltd. **ISRAEL**
24 Shenkar St., Kiryat Arie
IL-49001 Petah-Tiqva
Phone: +972 (0)3 / 922 18 24
Fax: +972 (0)3 / 924 0761

GIRIT CELADON LTD **ISRAEL**
12 H'aomanut Street
IL-42505 Netanya
Phone: +972 (0)9 / 863 39 80
Fax: +972 (0)9 / 885 24 30

CEG INTERNATIONAL **LEBANON**
Cebaco Center/Block A Autostrade DORA
Lebanon - Beirut
Phone: +961 (0)1 / 240 430
Fax: +961 (0)1 / 240 438

AFRICAN REPRESENTATIVE

CBI Ltd. **SOUTH AFRICA**
Private Bag 2016
ZA-1600 Isando
Phone: + 27 (0)11 / 977 0770
Fax: + 27 (0)11 / 977 0761

MITSUBISHI ELECTRIC
FACTORY AUTOMATION

Mitsubishi Electric Europe B.V. /// FA - European Business Group /// Gothaer Straße 8 /// D-40880 Ratingen /// Germany
Tel.: +49(0)2102-4860 /// Fax: +49(0)2102-4861120 /// info@mitsubishi-automation.com /// www.mitsubishi-automation.com