MITSUBISHI ELECTRIC
Changes for the Better

for a greener tomorrow eco Changes

FATEC

Mitsubishi Programmable Controllers
Training Manual
Ethernet (for GX Works3)

SAFETY PRECAUTIONS

(Always read these instructions before using the products.)

When designing the system, always read the relevant manuals and give sufficient consideration to safety.
During the exercise, pay full attention to the following points and handle the product correctly.

[EXERCISE PRECAUTIONS]

⚠WARNING

- Do not touch the terminals while the power is on to prevent electric shock.
- Before opening the safety cover, turn off the power or ensure the safety.

⚠CAUTION

- Follow the instructor's direction during the exercise.
- Do not remove the module of the demonstration machine or change wirings without permission.
  Doing so may cause failures, malfunctions, personal injuries and/or a fire.
- Turn off the power before mounting or removing the module.
  Failure to do so may result in malfunctions of the module or electric shock.
- When the demonstration machine (such as X/Y table) emits abnormal odor/sound, press the "Power switch" or "Emergency switch" to turn off.
- When a problem occurs, notify the instructor as soon as possible.

MEMO

REVISIONS

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

Revision date	*Manual number	Description
March 2017	SH(NA)-081820ENG-A	First edition

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

© 2017 MITSUBISHI ELECTRIC CORPORATION

TRADEMARKS

Ethernet is a registered trademark of Fuji Xerox Co., Ltd. in Japan.

Microsoft, Microsoft Access, Excel, SQL Server, Visual Basic, Visual C++, Visual Studio, Windows, Windows NT, Windows Server, Windows Vista, and Windows XP are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

CONTENTS

CHAPTER 5 EXERCISE 1 (COMMUNICATIONS USING MX COMPONENT BETWEEN A PERSONAL COMPUTER AND CPU MODULE) 5 - 1

CHAPTER 6 EXERCISE 2 (COMMUNICATIONS USING A FIXED BUFFER (PROCEDURE EXIST) BETWEEN CPU MODULES) 6 - 1

CHAPTER 7 EXERCISE 3 (COMMUNICATIONS USING A FIXED BUFFER (NO PROCEDURE) BETWEEN CPU MODULES) 7 - 1

CHAPTER 8 EXERCISE 4 (COMMUNICATIONS USING A FIXED BUFFER (NO PROCEDURE) VIA A ROUTER) 8 - 1

APPENDICES App. - 1

INTRODUCTION

This school textbook helps users to understand how to use the MELSEC iQ-R series Ethernet-equipped module RJ71EN71 and how to create programs for the module.
Use a personal computer that supports Microsoft® Visual Basic® 2012 as an external device for exercises.

RELEVANT MANUALS

Manual name [manual number]	Description	Available form
MELSEC iQ-R Ethernet/CC-Link IE User's Manual (Startup) [SH-081256ENG]	Specifications, procedures before operation, system configuration, wiring, and communication examples of Ethernet, CC-Link IE Controller Network, and CC-Link IE Field Network	e-Manual PDF
MELSEC iQ-R Ethernet User's Manual (Application) [SH-081257ENG]	Functions, parameter settings, programming, troubleshooting, I/O signals, and buffer memory of Ethernet	e-Manual PDF
MELSEC iQ-R Programming Manual (Instructions, Standard Functions/ Function Blocks) [SH-081266ENG]	Instructions for the CPU module, instructions dedicated for intelligent function modules, and standard functions/standard function blocks	e-Manual PDF
GX Works3 Operating Manual [SH-081215ENG]	System configuration of GX Works3, parameter settings, and operation method of the online function	e-Manual PDF
SLMP Reference Manual [SH-080956ENG]	A protocol (SLMP) for accessing an Ethernet-equipped module from an external device	e-Manual PDF
iQ Sensor Solution Reference Manual [SH-081133ENG]	Operation method of the online function with iQ Sensor Solution	e-Manual PDF
MX Component Version 4 Operating Manual [SH-081084ENG]	Settings and operation method of each utility for MX Component	PDF
MX Component Version 4 Programming Manual [SH-081085ENG]	Programming procedures, detailed explanations, and error codes of the ACT control	PDF
MELSEC iQ-R CC-Link IE Field Network User's Manual (Application) [SH-081259ENG]	Functions, parameter settings, programming, troubleshooting, I/O signals, and buffer memory of CC-Link IE Field Network	e-Manual PDF
MELSEC iQ-R CC-Link IE Controller Network User's Manual (Application) [SH-081258ENG]	Functions, parameter settings, troubleshooting, and buffer memory of CC-Link IE Controller Network	e-Manual PDF

Point

e-Manual refers to the Mitsubishi FA electronic book manuals that can be browsed using a dedicated tool.
e-Manual has the following features:

- Required information can be cross-searched in multiple manuals.
- Other manuals can be accessed from the links in the manual.
- The hardware specifications of each part can be found from the product figures.
- Pages that users often browse can be bookmarked.

ABBREVIATIONS AND TERMS

The following table lists the abbreviations and terms used in this textbook.

Term	Description
System A	A system that is set as system A to distinguish two systems, which are connected with two tracking cables. When the two systems start up at the same time, this system will be a control system. System switching does not affect the system A/B setting.
BUFRCV	A generic term for the GP.BUFRCV and ZP.BUFRCV
BUFRCVS	A generic term for the G.BUFRCVS and Z.BUFRCVS
BUFSND	A generic term for the GP.BUFSND and ZP.BUFSND
System B	A system that is set as system B to distinguish two systems, which are connected with two tracking cables. When the two systems start up at the same time, this system will be a standby system. System switching does not affect the system A/B setting.
CC-Link IE	A generic term for CC-Link IE Controller Network and CC-Link IE Field Network
CLOSE	A generic term for the GP.CLOSE and ZP.CLOSE
CPU module	A generic term for the MELSEC iQ-R series CPU modules
CPU module (built-in Ethernet port part)	A built-in Ethernet port part of the CPU module (CPU part for the RnENCPU) (📖 MELSEC iQ-R Ethernet/CC-Link IE User's Manual (Startup))
ERRCLEAR	A generic term for the GP.ERRCLEAR and ZP.ERRCLEAR
ERRRD	A generic term for the GP.ERRRD and ZP.ERRRD
Ethernet-equipped module	A generic term for the following modules when the Ethernet function is used: • RJ71EN71 • CPU module
FTP	The abbreviation for File Transfer Protocol. This protocol is used to transfer data files over a network.
ICMP	The abbreviation for Internet Control Message Protocol. This protocol is used to exchange messages of errors in an IP network or other information related to the network.
Device supporting iQSS	The abbreviation for a device which supports iQ Sensor Solution. For details on iQ Sensor Solution, refer to the following. 📖 iQ Sensor Solution Reference Manual
MELSECNET/10	The abbreviation for the MELSECNET/10 network system
MELSECNET/H	The abbreviation for the MELSECNET/H network system
OPEN	A generic term for the GP.OPEN and ZP.OPEN
OPS	A generic term for the partner products with built-in EZSocket that supports a redundant system. For communications with an OPS, use "OPS Connection Module" of "Module List" in "External Device Configuration" under "Basic Settings".
READ	A generic term for the JP.READ and GP.READ
RECV	A generic term for the JP.RECV and GP.RECV
RECVS	A generic term for the G.RECVS and Z.RECVS
REQ	A generic term for the J.REQ, JP.REQ, G.REQ, and GP.REQ
RnENCPU	A generic term for the R04ENCPU, R08ENCPU, R16ENCPU, R32ENCPU, and R120ENCPU
RnENCPU (CPU part)	A module on the left side of the RnENCPU (CPU part) (📖 MELSEC iQ-R Ethernet/CC-Link IE User's Manual (Startup))
RnENCPU (network part)	A module on the right side of the RnENCPU (network part) (📖 MELSEC iQ-R Ethernet/CC-Link IE User's Manual (Startup))
SEND	A generic term for the JP.SEND and GP.SEND
SLMP	The abbreviation for Seamless Message Protocol. This protocol is used to access an SLMP-compatible device or a programmable controller connected to an SLMP-compatible device from an external device.
SREAD	A generic term for the JP.SREAD and GP.SREAD
SWRITE	A generic term for the JP.SWRITE and GP.SWRITE
UINI	A generic term for the G.UINI, GP.UINI, Z.UINI, and ZP.UINI
WRITE	A generic term for the JP.WRITE and GP.WRITE
ZNRD	A generic term for the J.ZNRD and JP.ZNRD
ZNWR	A generic term for the J.ZNWR and JP.ZNWR
Intelligent device station	A station that exchanges I/O signals (bit data) and I/O data (word data) over CC-Link IE Field Network by cyclic transmission. This station responds to a transient transmission request from another station and also issues a transient transmission request to another station.
Intelligent function module	A module that has functions other than input and output, such as an A/D converter module and D/A converter module
Engineering tool	Another term for the software package for the MELSEC programmable controllers

Term	Description
Global label	A label that is enabled for all program data when multiple program data is created in the project. There are two types of global label: a module specific label (module label), which is generated automatically by GX Works3, and an optional label, which can be created for any specified device.
Subnet mask	A number used to logically divide one network into multiple subnetworks and manage them easily. The following Ethernet network systems can be configured: • A small-scale Ethernet network system in which multiple network devices are connected • A medium- or large-scale network system in which multiple small-scale network systems are connected via routers or other network communication devices
Submaster station	A station that serves as a master station to control the entire network if the master station of CC-Link IE Field Network is disconnected. Only one submaster station can be used in a network.
Data link	A generic term for cyclic transmission and transient transmission
Device	A device (X, Y, M, D, or others) in a CPU module
Transient transmission	A function of communications with another station, which is used when requested by a dedicated instruction or the engineering tool
Transient transmission group number	A number that is assigned for transient transmission to any given stations. By specifying a group of stations as transient transmission target, data can be sent to the stations of the same group number.
Network module	A generic term for the following modules: • Ethernet interface module • CC-Link IE Controller Network module • Module on CC-Link IE Field Network • MELSECNET/H network module • MELSECNET/10 network module • RnENCPU (network part)
Buffer memory	A memory in an intelligent function module, where data (such as setting values and monitoring values) is stored. When using the CPU module, the memory is indicated for storing data (such as setting values and monitored values) of the Ethernet function and data used for data communications of the multiple CPU function.
Head module	The abbreviation for the LJ72GF15-T2 CC-Link IE Field Network head module
Master station	A station that controls the entire network on CC-Link IE Field Network. This station can perform cyclic transmission and transient transmission with all stations. Only one master station can be used in a network.
Master operating station	A station that controls the entire network when the submaster function of CC-Link IE Field Network is used. Only one master operating station can be used in a network.
Module label	A label that represents one of memory areas (I/O signals and buffer memory areas) specific to each module in a given character string. GX Works3 automatically generates this label, which can be used as a global label.
Label	A label that represents a device in a given character string
Remote head module	The abbreviation for the RJ72GF15-T2 CC-Link IE Field Network remote head module
Link device	A device in a CC-Link IE module
Routing	A process of selecting paths for communications with other networks. There are two types of routing: dynamic routing that auto-selects the communication routes, and static routing where communication routes are arbitrarily set.
Local station	A station that performs cyclic transmission and transient transmission with the master station and other local stations on CC-Link IE Field Network
Control CPU	A CPU module that controls connected I/O modules and intelligent function modules. In a multiple CPU system, there are multiple CPU modules and each connected module can be controlled by a different CPU module.
Control station	A station that controls the entire network on CC-Link IE Controller Network. This station can perform cyclic transmission and transient transmission with all stations. Only one control station can be used in a network.
Control system	A system that takes control and performs network communications in a redundant system
Dedicated instruction	An instruction for using functions of the module
External device	A generic term for the personal computer and other Ethernet-equipped modules connected over Ethernet for data communications
Standby system	A backup system in a redundant system
Relay station	A station that includes two or more network modules. Transient transmission is performed through this station to stations on other networks.
Normal station	A station that performs cyclic transmission and transient transmission with the control station and other normal stations on CC-Link IE Controller Network
Predefined protocol support function	A function of GX Works3. This function sets protocols appropriate to each external device and reads/writes protocol setting data.
Redundant system	A system consisting of two systems that have the same configuration (CPU module, power supply module, network module, and other modules). Even after an error occurs in one of the two system, the other system takes over the control of the entire system. For details, refer to "Redundant system" of the following manual. 📖 MELSEC iQ-R Module Configuration Manual

MEMO

1 OVERVIEW

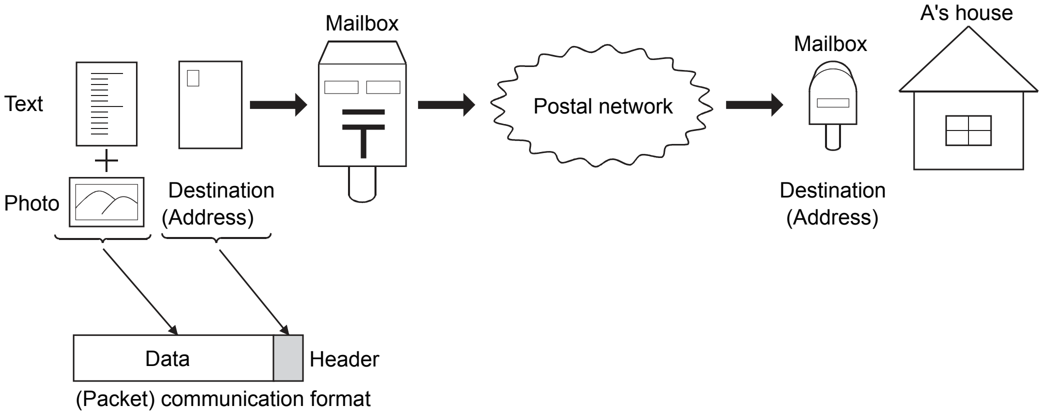
This textbook mainly describes the basic usage of the MELSEC iQ-R series Ethernet-equipped module.
For details on general computer network technology (such as TCP/IP communications), refer to commercially-available textbooks.
For details on the Ethernet-equipped module, refer to the product manuals.

1.1 Ethernet

The development of Ethernet was started by Xerox Palo Alto Research Center in the U.S. in 1973, and Ethernet was approved as a network technology by ANSI/IEEE standards and ISO international standards. Lately, Ethernet has a wider meaning such as networking equipment and communication paths. In addition to hardware technique including transmission lines, which are defined by Ethernet (in a narrow sense), actual network operation requires technology of TCP/IP or other protocols (protocol: a set of rules) for communications.

1.2 Addresses

The devices and computers connected to Ethernet must have their own addresses to communicate on the network.
The addresses which the user need to consider are basically IP addresses for Ethernet.



1.2.1 MAC addresses (Ethernet addresses)

A media access control address (MAC address) is a unique physical address which is assigned to each network device. (No other devices have the same MAC address.)
For Ethernet, a MAC address is shown as a 6-byte code: a vender code of 3 bytes (managed by IEEE) indicating an equipment manufacturer and a node number of 3 bytes (managed by each manufacturer).[*1]
As each Ethernet-connected device automatically gets a MAC address from the IP address specified by the user, the user need not consider the MAC address.
Although a MAC address may be called an Ethernet address or Internet address, it differs from the IP address described in the following section.

[*1] The MAC address of the Ethernet-equipped module is shown in the MAC ADD. field of the rating plate on the side of the module.

1.2.2 IP addresses

An internet protocol address (IP address) is an identification number assigned to identify each device or computer connected to an IP network such as the Internet and an intranet. (It corresponds to a mail address or a telephone number.)
Unique addresses managed internationally by each country are used on the Internet, where a network is connected on a global scale. An IP address is expressed by a 32-bit number for IPv4, which is now commonly used.
In general, an IP address is divided into four parts of 8 bits like 192.168.1.1 and represented in decimal.
The 32-bit value consists of a network part for identifying each network and the host part for identifying each connected device in the network (such as a computer).

- (IP address) = ((Class) + Network part) + (Host part)

Representation method of IP address

An IP address (IPv4) is expressed by a 32-bit number.

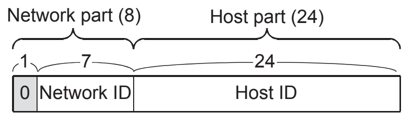
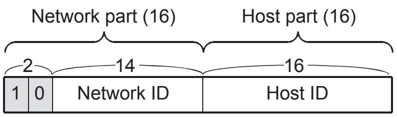
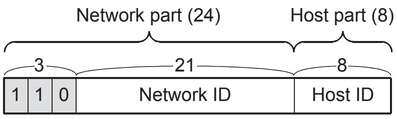
Binary 00000000000000000000000000000000 to 11111111111111111111111111111111
Decimal 0 to 4294967295
Hexadecimal 0 to FFFFFFFF

The number is divided every 8 bits for clarification.

Binary 00000000.00000000.00000000.00000000 to 11111111.11111111.11111111.11111111
Decimal 0. 0. 0. 0 to 255. 255. 255. 255
Hexadecimal 0. 0. 0. 0 to FF. FF. FF. FF

Classification by class

The classification system called class has been traditionally used to fixedly handle the boundary between the network part and the host part of an IP address.

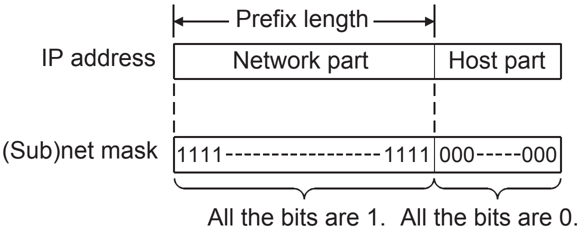
Class	Bit assignment[*1]		Upper 8 bits	Boundary of network address	Private IP address range
	Upper bits	Lower bits			
Class A	 Network part (8) Host part (24) 1 7 24 0 Network ID Host ID		0******* (0 to 127)	Upper 8 bits	10.0.0.0 to 10.255.255.255
Class B	 Network part (16) Host part (16) 2 14 16 1 0 Network ID Host ID		10****** (128 to 191)	Upper 16 bits	172.16.0.0 to 172.31.255.255
Class C	 Network part (24) Host part (8) 3 21 8 1 1 0 Network ID Host ID		110***** (192 to 223)	Upper 24 bits	192.168.0.0 to 192.168.255.255

*1 The □ part in the upper bits of each IP address indicates the class.

The classes A to C are the addresses for public use.
Among these addresses, the usable addresses for devices that are not directly connected to the Internet are called private IP addresses.

Management by class address

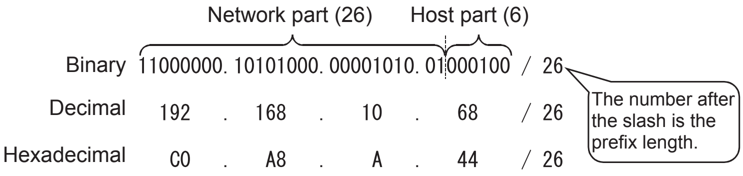
IP addresses were once managed by using the classes. Nowadays, due to the shortage of IP addresses, the class addresses with a variable-length network part are used to ensure the effective use of the address space.



The IP address whose host part bits are all 0 is a network address. For specifying the network address length (prefix length) clearly, "/" may be added to the end of the IP address and the bit length of the network address part may be added after "/".
A netmask is commonly used for a historical reason. All the bits in the network part are 1 and all the bits in the host part are 0 for the netmask.
As the netmask is used to divide (subnet) a network, the netmask is also called subnet mask.

Ex.

For the IP address of 192.168.10.68 with the prefix length of 26 bits



Binary	11000000.	10101000.	00001010.	01¦000100	/ 26
Decimal	192 .	168 .	10 .	68	/ 26
Hexadecimal	C0 .	A8 .	A .	44	/ 26

Information examples of when an IP address is set to the host

Information	Address	Address value (Binary)				Remarks
IP address	192.168. 10. 68/26	1100 0000	1010 1000	0000 1010	0100 0100	—
(Sub)net mask	255.255.255.192	1111 1111	1111 1111	1111 1111	1100 0000	
Network address	192.168. 10. 64/26	1100 0000	1010 1000	0000 1010	0100 0000	
Broadcast address	192.168. 10.127	1100 0000	1010 1000	0000 1010	0111 1111	An address whose host part bits are all 1

Point

Special IP addresses

- Address whose bits are all 0 or 1

  The IP address 0.0.0.0 is used when the own IP address is unknown or informing the communication destination device of the IP address is unnecessary.

  The IP address 255.255.255.255 indicates a broadcast address (destination: all the devices and computers connected to the same network).
- Loopback address

  An address which is used among programs executed on the same device (computer).

  The range is from 127.0.0.0 to 127.255.255.255.
- Multicast address

  An address which is used for communications among specific groups.

  The range is from 224.0.0.0 to 239.255.255.255.
- Private address

  It seems that any IP address can be used when devices are not connected to the Internet. However, addresses which can be freely assigned are specified to avoid troubles.

  This type of address is called a private address.

Network address*1	IP address range*2
10/8	10.0.0.0 to 10.255.255.255
172.16/12	172.16.0.0 to 172.31.255.255
192.168/16	192.168.0.0 to 192.168.255.255

*1 The numerical value on the right of "/" indicates the number of bits of the network address in the upper part of each IP address (the number of bits from the most significant bit).

*2 Broadcast IP addresses are included.

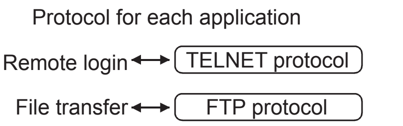
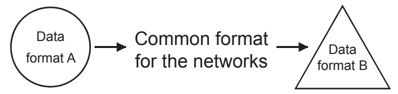
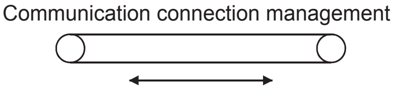
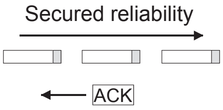
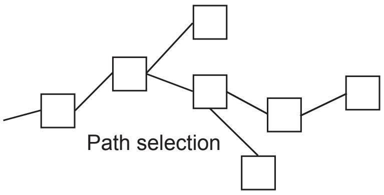
1.3 Predefined Protocols

The MELSEC iQ-R Ethernet-equipped module supports two protocols (rules) for communications: TCP/IP and UDP/IP.

1.3.1 Communication model

The OSI reference model is a famous communication mechanism model defined by the International Organization for Standardization (ISO).

This model classifies the functions required for communications into seven layers.

Layer		Function	Image	Protocol example
7	Application layer	"What the users want to do" • Agreement on contents of actual services • What the users can see • Available services	 Protocol for each application Remote login TELNET protocol File transfer FTP protocol	HTTP TELNET FTP SMTP
6	Presentation layer	"Defining and converting the data format" • Defining a data representation system • Encrypting/decrypting and compressing/decompressing data • Character code, data format	 Data format A Common format for the networks Data format B Differences in data representation are resolved.	MIME HTML XML
5	Session layer	"Establishing a communication connection" • Establishing/disconnecting a connection • Authenticating a connection • Synchronizing data communications		RPC
4	Transport layer	"Delivering data to the communication destination device correctly" • Securing reliable data transfer between the source and destination • Correcting errors (correcting the arrival sequence, requesting retransmission) • Flow control of communications		TCP UDP
3	Network layer	"Communication procedure with non-adjacent communication destination devices" • Defining a path control (routing) • Defining a communication path decision • Establishing a virtual connection using addresses		IP
2	Data link layer	"Delivering data to the adjacent device" • Communication procedure between adjacent devices • Defining the format of send/receive data • Detecting data errors among devices and defining a correction method	 Conversion of frames/bit strings and data transfer among adjacent devices	Ethernet PPP
1	Physical layer	"Physical connection" • Conditions for electrical connection at the lowest level • ON/OFF definition of data signals • Connector shape, pin layout for each signal, and others		Ethernet ISDN Telephone line

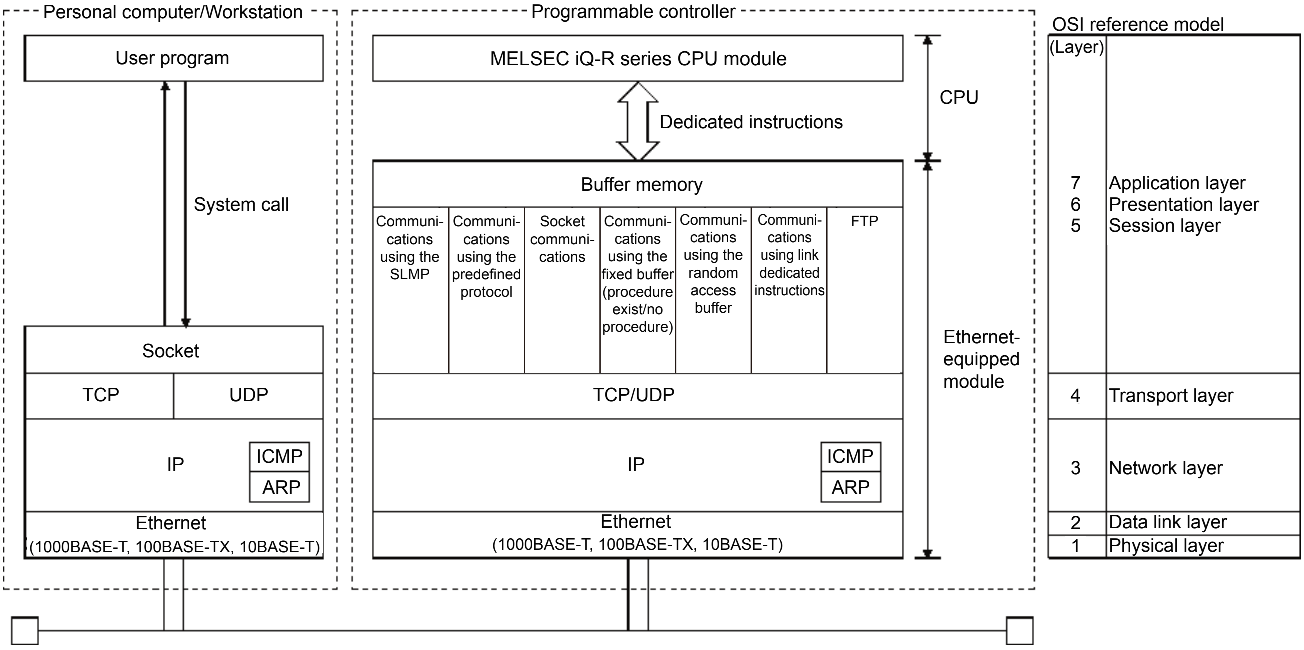
Bigger numbers are higher (logical) layers and smaller numbers are lower (physical) layers.

IP belongs to the network layer, and both TCP and UDP belong to the transport layer.

Ethernet belongs to the data link layer and physical layer.

Ethernet-equipped module and communication model

- The following figure shows the correspondence between the software configuration of the Ethernet-equipped module and the OSI reference model.
- The physical and data link layers are the Ethernet part of the Ethernet-equipped module.
- The network and the transport layers are the IP and TCP/UDP parts of the Ethernet-equipped module.
- The higher layers (session, presentation, and application layers) are the software part of various functions designed specific to the MELSEC iQ-R series modules, which have been implemented by combining the CPU module and Ethernet-equipped module.



Personal computer/Workstation
User program
System call
Socket
TCP
UDP
IP
ICMP
ARP
Ethernet
(1000BASE-T, 100BASE-TX, 10BASE-T)
Programmable controller
MELSEC iQ-R series CPU module
Dedicated instructions
Buffer memory
Communi-cations using the SLMP
Communi-cations using the predefined protocol
Socket communi-cations
Communi-cations using the fixed buffer (procedure exist/no procedure)
Communi-cations using the random access buffer
Communi-cations using link dedicated instructions
FTP
TCP/UDP
IP
ICMP
ARP
Ethernet
(1000BASE-T, 100BASE-TX, 10BASE-T)
CPU
Ethernet-equipped module
OSI reference model
(Layer)
7 Application layer
6 Presentation layer
5 Session layer
4 Transport layer
3 Network layer
2 Data link layer
1 Physical layer

1.3.2 IP protocol

Role of IP

The internet protocol (IP) is a network-layer protocol which is processed by all the devices connected to an IP network.

The most important role of the TCP/IP (UDP/IP) network is to transfer data to a device or computer with the destination address.

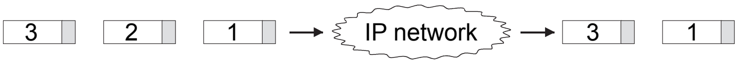
This role is achieved by IP.

Data (packets) are delivered with the information called an IP header (a tag) attached.

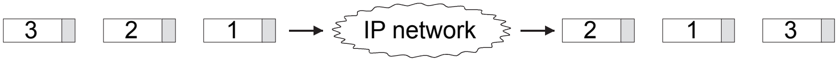
Restrictions on IP

Although IP delivers data to a destination computer, some restrictions are applied.

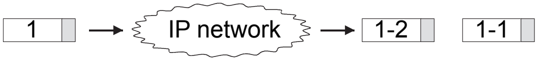
(a) There is no guarantee that data reaches the destination computer.



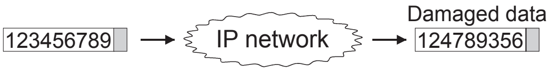
(b) There is no guarantee that data reaches in the order of sending.



(c) As the delivery size at one time is limited, a packet may be divided into several pieces at transmission.[*1]



(d) There is no guarantee of no data damage.



Shortly, IP makes every effort to deliver data (packets) to the communication destination device or computer, however, it does not guarantee delivery of the packets (best effort).

Note that these restrictions does not need to be considered when TCP is used as a higher layer of IP.

When using UDP as a higher layer, note that the above restrictions are applied except data damage detection.

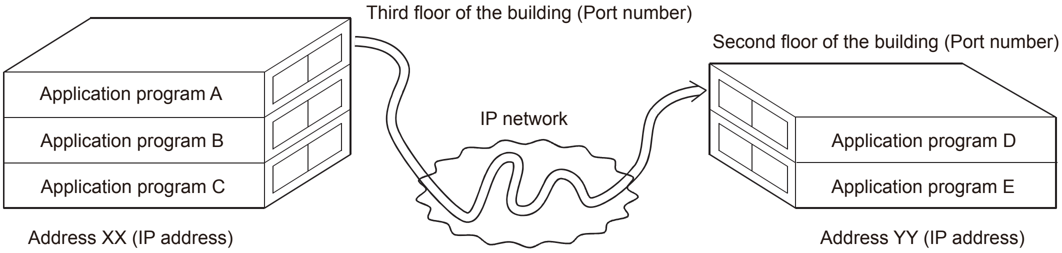
[*1] The size of one message (one packet) that can be sent/received by the Ethernet-equipped module is up to 1500 bytes (including the IP header).
Data exceeding 1500 bytes are divided when sent in either of the TCP/IP or UDP/IP communications. The divided data are reassembled into one data on the receiving side and handed to the application program.

1.3.3 TCP and UDP

TCP and UDP are transport-layer protocols which are processed by the devices or computers on the both ends of communications.

Port numbers

Actual communications are performed between the application programs operated on the devices or computers.
TCP and UDP identify which application programs are communicating with each other using the port numbers.
If an IP address is regarded as an address, a port number is a floor of a building.



Combinations of the following five items identify individual communication.

- Destination IP address
- Source IP address
- Destination port number
- Source port number
- Protocol number (TCP = 6H, UDP = 17H)

Point

Consult with the network manager before setting the port number of the Ethernet-equipped module.
Specify a value not used for other ports within the range of 401H to 1387H and 1392H to FFFEH.
(The port numbers 1388H to 1391H cannot be specified because they are used by the Ethernet-equipped module system.)

Comparison between TCP and UDP

The request level for a network differs depending on user applications.

However, it is difficult to create each unique protocol for many requests.

Therefore, the transmission control protocol (TCP) and user datagram protocol (UDP) have been developed as minimally required basic services.

- TCP: Fixes connection to the destination first and performs bidirectional 1:1 communications with high reliability.
- UDP: Performs unidirectional communications to transfer data given from an application to a specified destination.
  The communications are performed at a high speed because the data is directly sent using IP.

TCP is suitable for assured data transfer.

UDP is suitable for real-time monitoring on the display of a personal computer.

The following table compares the characteristics between TCP and UDP.

Item	TCP	UDP	Remarks
Reliability	High	Low	
(Processing) speed	Low	High	
Number of external devices	1:1	1:1 or 1:n	Unicast (1:1 communications) Multicast (1:n communications)[*1]
Guarantee to reach the destination	Guaranteed	Not guaranteed	
Operation when a send error occurs	Resends the data automatically. (The operation depends on the setting.)	Not resend the data. (The packet is discarded.)	
Communication connection establishment	Required	Not required	Data reaches the destination in order of sending packets for TCP.
Transfer type	Stream type (Instructions and data are sent with strings.)	Datagram type (Instructions and data are sent in a fixed format.)	Datagram transfer is also available for TCP in the application level.
Flow control	Available	Not available	The sending side controls the send data amount depending on the buffer size of the receiving side.
Congestion control (resend control)[*2]	Available	Not available	Send packet amount is controlled depending on the congestion degree of the network.
External device change in an open connection	Not allowed	Allowed[*3]	☞ Page 2 - 21 Open/Close Processing

[*1] The "n" of multicast (1:n communications) represents multiple devices belonging to one group on the same Ethernet network.

[*2] A buildup of communication packets on the network is called congestion.

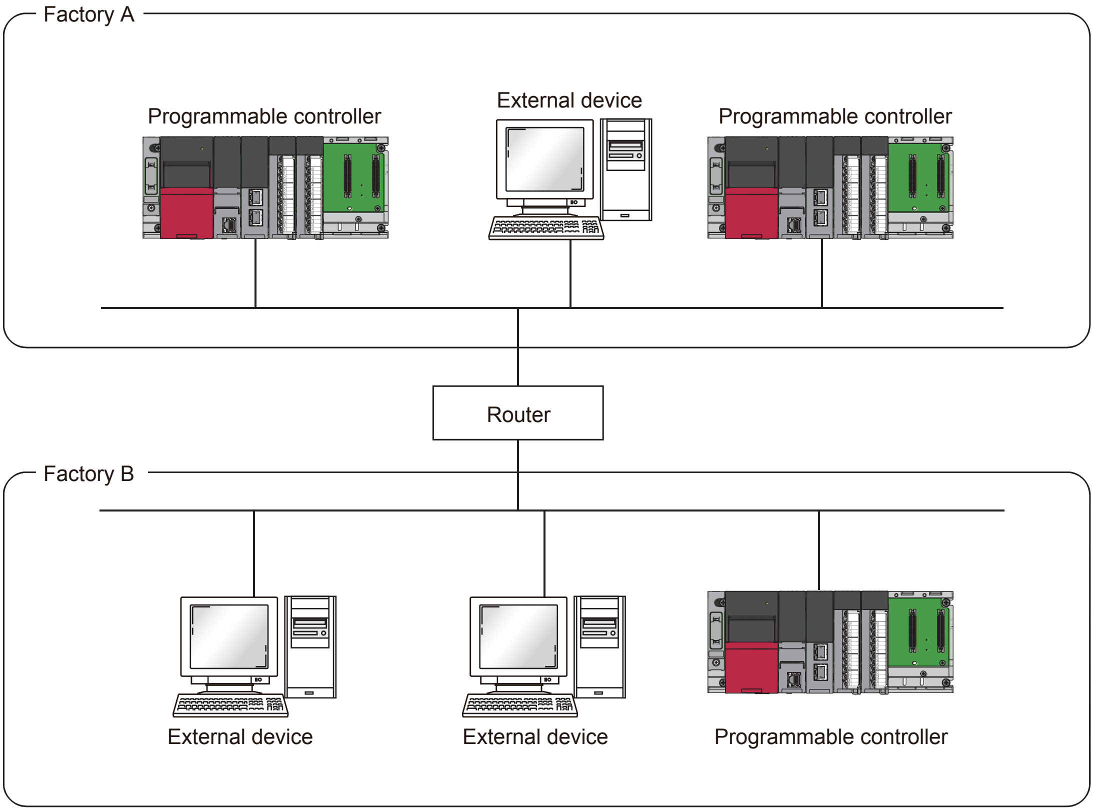
[*3] External device change while the connection is open may cause communication troubles.
Do not change the external device while the connection is open.

1.4 MELSEC iQ-R Ethernet-equipped Module

1.4.1 Role of Ethernet-equipped module

The Ethernet-equipped module supports data communications between an external device and a CPU module or between CPU modules connected to Ethernet, and sends/receives data to/from the external device in the TCP/IP or UDP/IP communications.

Therefore, the CPU module status can be checked remotely from an external device.



All the external devices can communicate with programmable controllers in factory A and B over Ethernet.

The programmable controllers in factory A and B can also communicate with each other.

1.4.2 Overview of Ethernet-equipped module

■Supporting the TCP/IP and UDP/IP communications as communication methods

The Ethernet-equipped module supports the TCP/IP and UDP/IP communications.

A communication method suitable for the external device can be selected.

■Selectable data code

The Ethernet-equipped module uses binary-code data or ASCII-code data for communications.

For details on the data codes, refer to Page 2 - 1 Two Data Codes.

■Communication functions provided for various purposes

The Ethernet-equipped module has the following communication functions.

Any one of these functions can be used for data communications according to user's communication purposes.

For details, refer to Page 2 - 3 Data Communication Functions.

- Communications using the SLMP
- Communications using the predefined protocol
- Socket communications
- Communications using the fixed buffer (procedure exist/no procedure)
- Communications using the random access buffer
- Communications using link dedicated instructions

Point

The communication method and the data code of the data to be sent/received must be the same between the communicating devices.

MEMO

2 BEFORE USING AN ETHERNET-EQUIPPED MODULE

2.1 Two Data Codes

The Ethernet-equipped module can communicate with external devices by using binary code data or ASCII-code data.
Switch the code setting between the binary code data or ASCII-code data with GX Works3.
For details, refer to Page 4 - 13 Settings with GX Works3.

Communications using the binary code

The Ethernet-equipped module sends/receives 1-byte data without converting it.

■Advantage

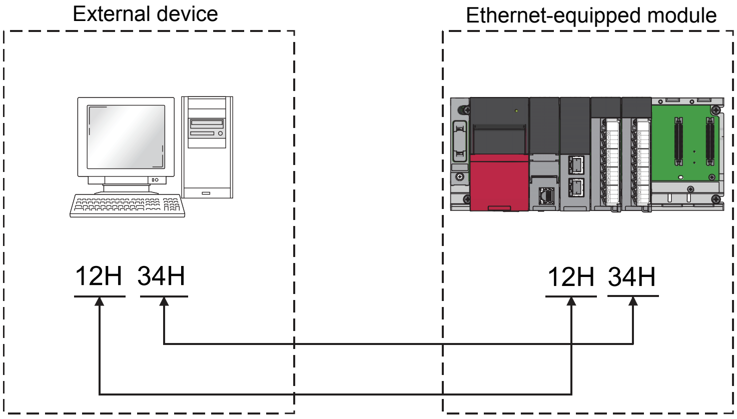
- A capacity of binary code data to be sent/received is half as large as that of ASCII-code data, and therefore a load on a communication line is reduced.
- Data of 00H to FFH can be processed.

■Disadvantage

- Binary code data must be converted to ASCII-code data to display numerical value data.

Ex.

Sending/receiving data of 1234H



• 1-byte data is sent/received without being converted.

Communications using the ASCII code

The Ethernet-equipped module sends/receives 1-byte data as data equivalent to two characters in the ASCII code.

■Advantage

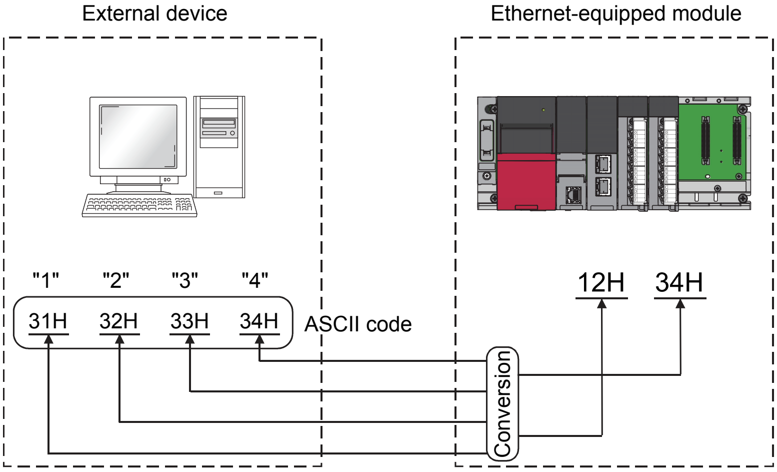
- An external device can display data without converting it.

■Disadvantage

- A capacity of ASCII-code data to be sent/received is twice as large as that of binary code data, and therefore a load on a communication line is increased.
- ASCII-code data must be converted to binary code data to process numerical value data.
  (The Ethernet-equipped module side automatically converts the data.)

Ex.

Sending/receiving data of "1234"



- 1-byte data from the programmable controller side is sent/received as data equivalent to two characters.

Relationship between communication methods and data codes

The following table lists the availability of data codes in each communication method.

○: Selectable, —: Not selectable

Data communication function		Communication data code setting	
		Binary code	ASCII code
Communications using the fixed buffer	Procedure exist	○	○
	No procedure	○[*1]	—
Communications using the random access buffer		○	○

[*1] The Ethernet-equipped module uses binary code data for communications regardless of the communication data code setting of GX Works3.

2.2 Data Communication Functions

The Ethernet-equipped module has six communication functions: communications using the SLMP, communications using the predefined protocol, socket communications, communications using the fixed buffer, communications using the random access buffer, and communications using link dedicated instructions. For details, refer to the following.

📖 MELSEC iQ-R Ethernet User's Manual (Application)

The following sections describe the outline of each communication function.

2.2.1 Communications using the SLMP

SLMP is a protocol used by external devices to access SLMP-compatible devices over Ethernet.

Communications using the SLMP are available among devices that can receive/send messages with the SLMP control procedure.

The Ethernet-equipped module processes and transfers data following instructions (command) from the external device. Thus, the programmable controller only needs the open/close processing and does not require a program for data communications.

For the communications using the SLMP, refer to the following.

📖 SLMP Reference Manual

Applications

This section describes the applications of communications using the SLMP.

■Data read/write

Data read/write can be executed for the following data. With this function, the external device can monitor the operation of the Ethernet-equipped module, analyze data, and control production.

- Device or global label of the CPU module connected with the RJ71EN71 (When the Ethernet function of the RJ71EN71 or the RnENCPU (network part) is used)
- Device or global label of the CPU module (When the Ethernet function of the CPU module (CPU part for the RnENCPU) is used)
- Buffer memory of the intelligent function module

■File read/write

Files such as parameter files stored in a CPU module can be read/written. Files in a CPU module can be managed on an external device.

■Remote control of a CPU module

A CPU module can be remotely controlled from the external device by using remote operations.

■Remote password lock/unlock

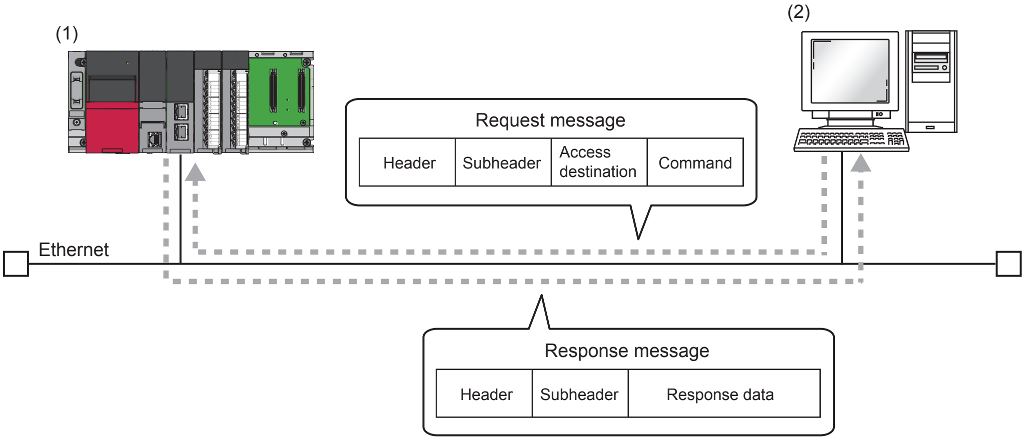
The remote password can be locked and unlocked from the external device.

■Access to the programmable controller on another station over other network

In systems over CC-Link IE Controller Network, CC-Link IE Field Network, MELSECNET/H, MELSECNET/10, or Ethernet, the programmable controller on another station can be accessed from the external device over a network. However, when the external device is connected to the CPU module (built-in Ethernet port part), other stations cannot be accessed over a network such as CC-Link IE Controller Network and CC-Link IE Field Network.

Communication process

When a message is sent from the external device to the Ethernet-equipped module by using the SLMP message format, the Ethernet-equipped module performs processing corresponding to the received message. During communications, the Ethernet-equipped module functions as a server and the external device (terminals such as a personal computer) functions as the client. The server (Ethernet-equipped module) automatically returns a response message suitable for the request message received from the client.



(1) Server side: Ethernet-equipped module
(2) Client side: External device

Data communication procedure

This section describes a procedure for communications using the SLMP.

1. After setting the module parameters, check that the initial processing of the Ethernet-equipped module has completed successfully. ('Initial status' (Un\G1900024.0): On)
2. Perform the open processing to establish a connection between the Ethernet-equipped module and external device.
3. After the connection is established, the SLMP messages are sent from the external device.
4. Close the connection when communications end.

Point

In the following cases, the Ethernet-equipped module performs a remote password check when the external device is accessing the programmable controller. If communications are not performed, unlock the remote password.

- When a remote password is set for the CPU module
- When a connection for exchanging data with external device is set as a remote password check target

■Setting procedure

Set "External Device Configuration" under "Basic Settings".

1. Select "SLMP Connection Module" in "Module List", and drag and drop it to "List of devices" or "Device map area".
2. Set the other items to the connection if required.

■Communications using an auto-open UDP port

The auto-open UDP port is used for communications using the SLMP.

The auto-open UDP port is a UDP/IP port that automatically opens and closes at the following timing. When this port is used, communications are enabled when the initial processing completes. Communications can be performed without a program regardless of the connection's open status.

- Open/close timing

  After the Ethernet-equipped module initial processing completes, the port automatically opens according to the registered parameter settings. The port automatically closes when the Ethernet-equipped module station is powered off or reset.

Point

- When the initial processing completes successfully, the Ethernet-equipped module enables communications using an automatic open UDP port. The module waits for a communication request to the Ethernet-equipped module on the own station (Automatic open).
- The Ethernet-equipped module accepts and processes requests from anywhere as long as they are addressed to the Ethernet-equipped module itself.
- If a communication request is received from an external device, the corresponding port number is occupied until that processing ends. Even if another communication request is accepted during this time, the communication processing will be waited.

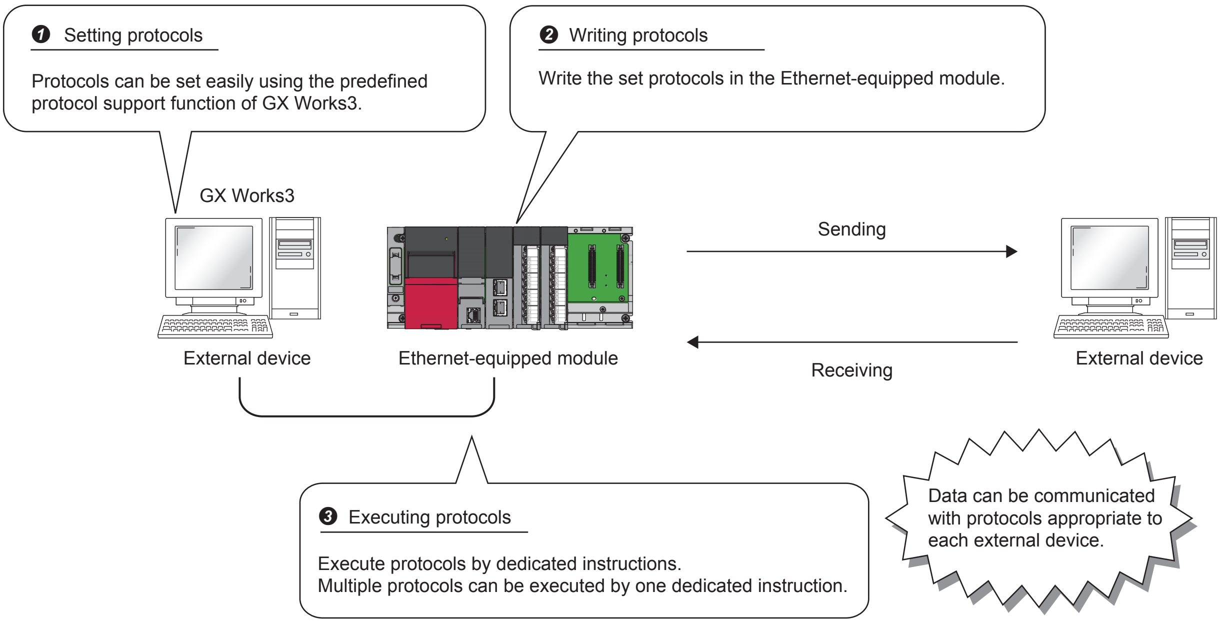
2.2.2 Communications using the predefined protocol

Data can be sent/received between an external device (such as measuring instrument and bar code reader) and a CPU module following a protocol of the external device.

Data that varies according to communication sessions can be handled by incorporating a device or buffer memory into a communication packet.

Set a protocol required for communications with external devices by using the engineering tool.

The protocol can be set by selecting from the predefined protocol library, or it can be created and edited.



❶ Setting protocols

Protocols can be set easily using the predefined protocol support function of GX Works3.

❷ Writing protocols

Write the set protocols in the Ethernet-equipped module.

❸ Executing protocols

Execute protocols by dedicated instructions.
Multiple protocols can be executed by one dedicated instruction.

Data can be communicated with protocols appropriate to each external device.

Point

The number of protocols and packets that can be registered is as follows.

- Protocols: 128 maximum
- Packets: 256 maximum
- Packet data area size: 12288 bytes maximum

When the number of packets reaches the upper limit, protocols cannot be added even if the number of protocols has not reached the upper limit. If the packet data area size reaches the upper limit, protocols and packets cannot be added even if the number of protocols and packets has not reached the upper limit.

Applicable connections

The connections No.1 to No.16 of the P1 connector can be used for the communications using the predefined protocol.

The communications using the predefined protocol cannot be used with the P2 connector.

Data communication procedure

When the predefined protocol support function is used, data can be communicated with an external device by using the following procedure.

1. Select, create, or edit a protocol with the predefined protocol support function, and write protocol setting data.
2. Set the module parameters.
3. Write the parameters to the CPU module, and check that initial processing of the Ethernet-equipped module completed successfully. ('Initial status' (Un\G1900024.0): On)
4. Perform the open processing to establish a connection between the Ethernet-equipped module and external device.
5. Execute the protocol with the dedicated instruction (SP.ECPRTCL instruction or GP.ECPRTCL instruction).
6. Close the connection when communications end.

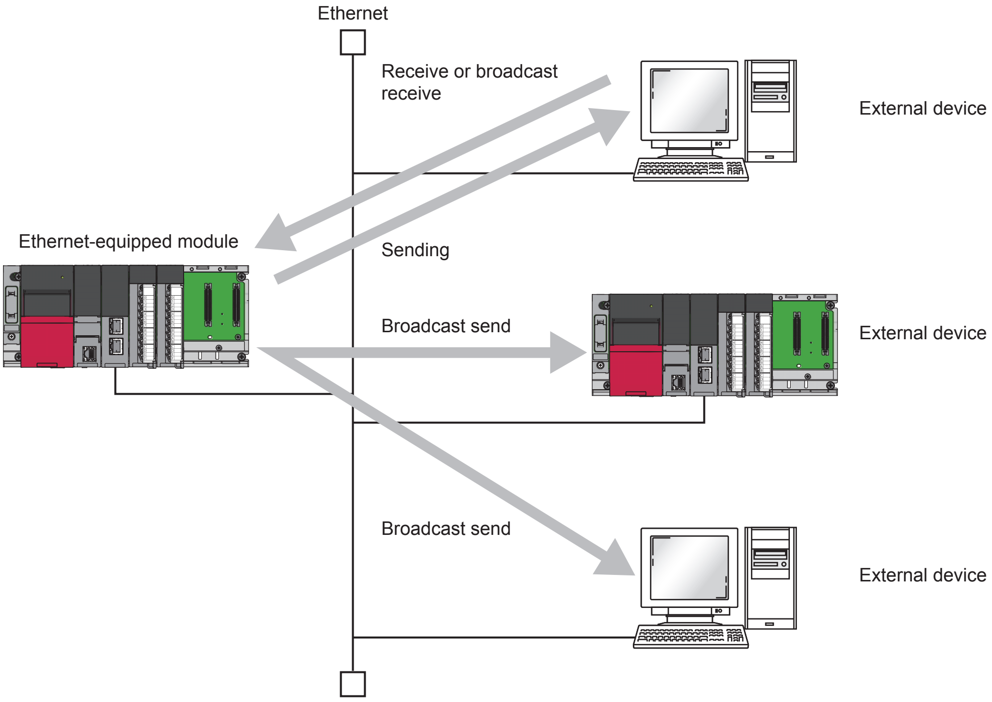
Point

The communication data code is binary code communication regardless of the selected settings.

2.2.3 Socket communications

By using dedicated instructions, arbitrary data can be sent/received with an external device connected to Ethernet over TCP/IP or UDP/IP.

Use this communication function for bidirectional communications one-on-one with an external device.



Point

For examples of socket communications, refer to the following.

📖 MELSEC iQ-R Ethernet/CC-Link IE User's Manual (Startup)

■Precautions

The socket communications cannot be used when the RJ71EN71 network type is set to "Q Compatible Ethernet".

Setting procedure

Set "External Device Configuration" under "Basic Settings".

1. Select an external device to be connected in "Module List", and drag and drop it to "List of devices" or "Device map area".

External device name	Description
UDP Connection Module	Select to communicate with the external device using UDP/IP.
Active Connection Module	Select to perform the open processing to the external device from the Ethernet-equipped module (Active open) and communicate using TCP/IP.
Unpassive Connection Module	Select to receive the open processing from the unspecified external device (Unpassive open) and communicate using TCP/IP.
Fullpassive Connection Module	Select to receive the open processing from the specified external device (Fullpassive open) and communicate using TCP/IP.

2. Set "Communication Method" for the external device to "Socket Communication".

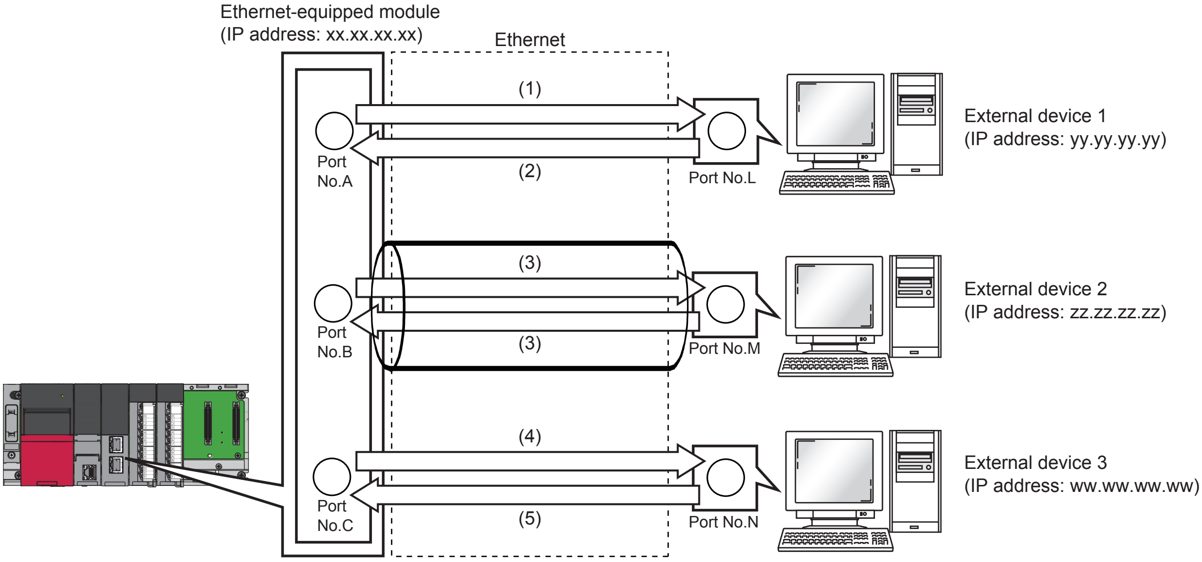
3. Set the other parameters required for communications in the connection.

Communication process

In the socket communications, port numbers that distinguish communications are used to enable multiple communication sessions with the external device. These port numbers are used for both TCP/IP and UDP/IP.

For send: Specify the send source Ethernet-equipped module's port number and the send destination external device's port number.

For receive: Specify the Ethernet-equipped module's port number, and read the data sent to that port.



(1) Sending UDP data from Ethernet-equipped module's port number A to external device 1's port number L
(2) Sending UDP data from external device 1's port number L to Ethernet-equipped module's port number A
(3) Sending data with TCP/IP connection
(4) Sending UDP data from Ethernet-equipped module's port number C to external device 3's port number N
(5) Sending UDP data from external device 3's port number N to Ethernet-equipped module's port number C

2.2.4 Communications using the fixed buffer

Communications using the fixed buffer use TCP/IP and UDP/IP to send and receive arbitrary data with an external device connected to Ethernet with dedicated instructions in the same manner as communicating data in socket communications. Arbitrary data can be sent/received between a CPU module and an external device by using the fixed buffer of the RJ71EN71 and the RnENCPU (network part).

The following table lists the differences with socket communications.

Item	Difference	
	Socket communications	Communications using the fixed buffer
Connection send/receive	Allows send and receive with one connection.	Specifies send or receive for one connection. (Two connections are required for send and receive.)

■Precautions

The CPU module (CPU part for the RnENCPU) cannot communicate data with the fixed buffer.

Differences between the "Procedure Exist" and "No Procedure" control methods

"Procedure Exist" and "No Procedure" control methods can be used for the communications using a fixed buffer. The following table lists the differences between "Procedure Exist" and "No Procedure".

Item	Difference	
	Procedure exist	No procedure
Message format	Data is sent and received with the predetermined data format.	Data is sent and received according to the message format of the external device.
Response for received data	A response is sent for the received data.	No response is sent for the received data.
Data code	Data can be communicated with binary code or ASCII code.	Data is communicated only with binary codes.
Data length specified with dedicated instructions	Specify with a number of words.	Specify with a number of bytes.
Amount of application data per data communication session[*1]	Maximum 5113 words (binary code) Maximum 2556 words (ASCII code)	Maximum 10238 bytes

[*1] The following value is used if the RJ71EN71 network type is set to "Q Compatible Ethernet".
Procedure Exist: Maximum 1017 words (binary code), maximum 508 words (ASCII code)
No Procedure: Maximum 2046 bytes

Setting procedure

Set "External Device Configuration" under "Basic Settings".

1. Select an external device to be connected in "Module List", and drag and drop it to "List of devices" or "Device map area".

External device name	Description
UDP Connection Module	Select to communicate with the external device using UDP/IP.
Active Connection Module	Select to perform the open processing to the external device from the Ethernet-equipped module (Active open) and communicate using TCP/IP.
Unpassive Connection Module	Select to receive the open processing from the unspecified external device (Unpassive open) and communicate using TCP/IP.
Fullpassive Connection Module	Select to receive the open processing from the specified external device (Fullpassive open) and communicate using TCP/IP.

2. Set "Communication Method" for the external device to "Fixed Buffer (Procedure Exist)" or "Fixed Buffer (No Procedure)".

3. Set "Fixed Buffer Send/Receive Setting".

4. Set the other parameters required for communications in the connection.

Applicable dedicated instructions

The following table lists the dedicated instructions used for the communications using the fixed buffer.

For details on the dedicated instructions, refer to the following.

📖 MELSEC iQ-R Programming Manual (Instructions, Standard Functions/Function Blocks)

■List of dedicated instructions

The following table lists the dedicated instructions used by each module.

Instruction	Description
GP.CONOPEN[*1]	Establishes a connection.
OPEN	
GP.CONCLOSE[*1]	Closes the connection.
CLOSE	
BUFRCV	Reads the receive data from the external device.
BUFRCVS	Reads the receive data with an interrupt program.
BUFSND	Sends data to the external device.

[*1] This instruction cannot be used when "Q Compatible Ethernet" is selected in the network type.

Communication process

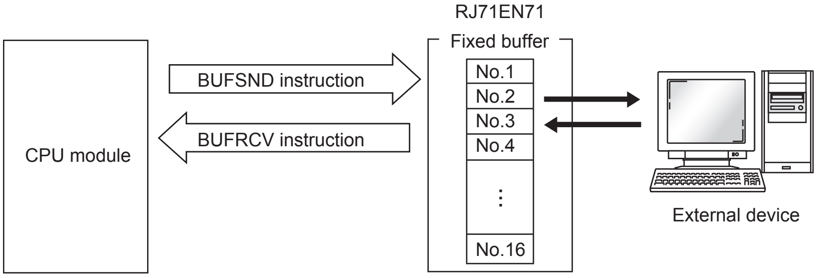
This section describes a process of communications with fixed buffer.

■Data flow

The dedicated instructions are used to send and receive data in communications using a fixed buffer. (☞ Page 2 - 11 Applicable dedicated instructions)

With "Procedure Exist", the CPU module and external device communicate data one-on-one. A handshake is established with the external device when the CPU module receives/sends data from/to the external device.

With "No Procedure", data is sent from the CPU module and received from the external device without a procedure.

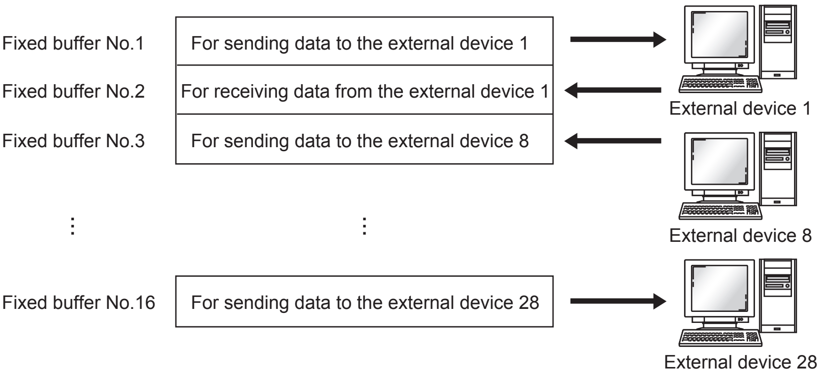


■External devices capable of data communications

Data can be communicated with the following external devices.

- Devices in Ethernet to which the RJ71EN71 is connected
- Devices in Ethernet to which the RnENCPU (network part) is connected
- Devices connected via a router

As shown in the following figure, the external devices for communications and the applications (for send/receive and "Procedure Exist"/"No Procedure") are set in "External Device Configuration" under "Basic Settings" by using each fixed buffer (No.1 to No.16) to fix the external device for each buffer.


Fixed buffer No.1
For sending data to the external device 1
Fixed buffer No.2
For receiving data from the external device 1
Fixed buffer No.3
For sending data to the external device 8
⋮
⋮
Fixed buffer No.16
For sending data to the external device 28
External device 1
External device 8
External device 28

Pay attention to the following points when changing the external device.

- During TCP/IP communications, the external device can be changed only when a connection is not established with the external device (when the open completion signal is off).
- During UDP/IP communications, the external device can be changed regardless of the connection status with the external device.
- When changing the external device, do not use the pairing open or alive check function.

■Processing during data send/receive

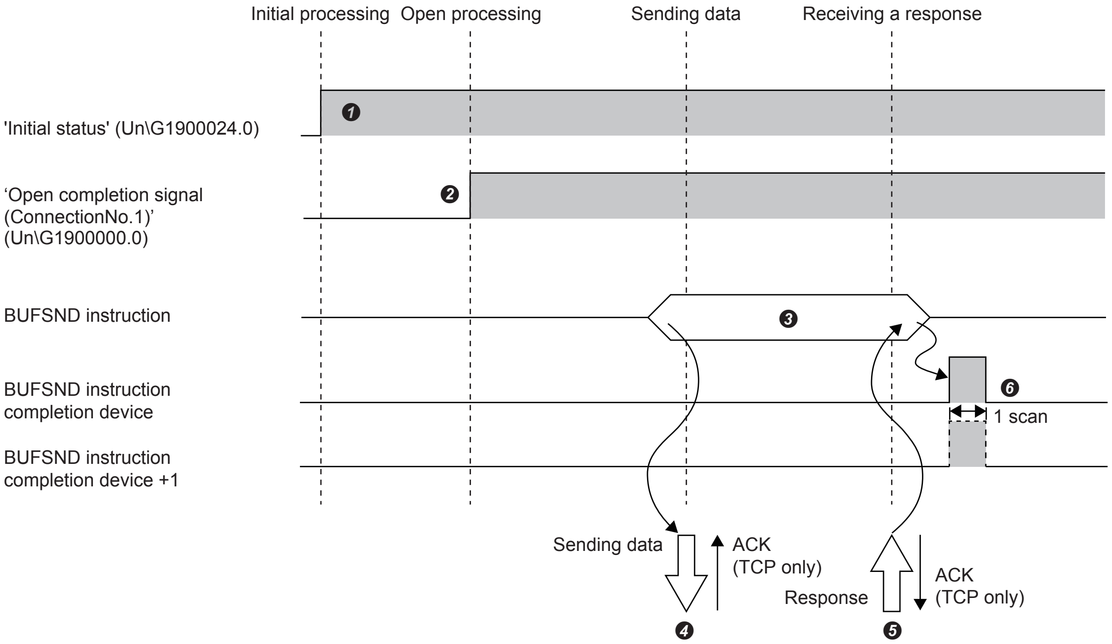
- During data send
  When the BUFSND instruction is executed, the RJ71EN71 and the RnENCPU (network part) send data from the corresponding fixed buffer to the external device set in the specified connection.
- During data receive
  If data is being received from an external device set in the specified connection, the RJ71EN71 and the RnENCPU (network part) will receive the data.
  If data is being received from an external device that is not set in the specified connection, the RJ71EN71 and the RnENCPU (network part) will ignore the received data.

Send procedure

This section describes the processing of when data is sent from the RJ71EN71 or the RnENCPU (network part) to an external device.

■Procedure exist

The following figure shows the send processing for the fixed buffer No.1 area corresponding to the connection No.1.



Initial processing
Open processing
Sending data
Receiving a response
'Initial status' (Un\G1900024.0)
❶
'Open completion signal (ConnectionNo.1)' (Un\G1900000.0)
❷
BUFSND instruction
❸
BUFSND instruction completion device
❻
1 scan
BUFSND instruction completion device +1
Sending data
ACK (TCP only)
Response
ACK (TCP only)
❹
❺

❶ Normal completion of the initial processing is checked. ('Initial status' (Un\G1900024.0): On)

❷ A connection is established between the external device and the RJ71EN71 or the RnENCPU (network part), and normal completion of the connection No.1 open processing is checked. (☞ Page 2 - 21 Open/Close Processing)

❸ The BUFSND instruction is executed. (Data is sent.)

❹ The send data of data length is sent from the fixed buffer No.1 area to the external device.

❺ When the external device receives the data from the RJ71EN71 or the RnENCPU (network part), it returns a response to the RJ71EN71 or the RnENCPU (network part).

❻ When the RJ71EN71 or the RnENCPU (network part) receives response the external device, it finishes data send. If the response is not returned within the response monitor timer value, a data send error occurs.[*1] If the data send completes abnormally, execute the BUFSND instruction again and start the send processing.

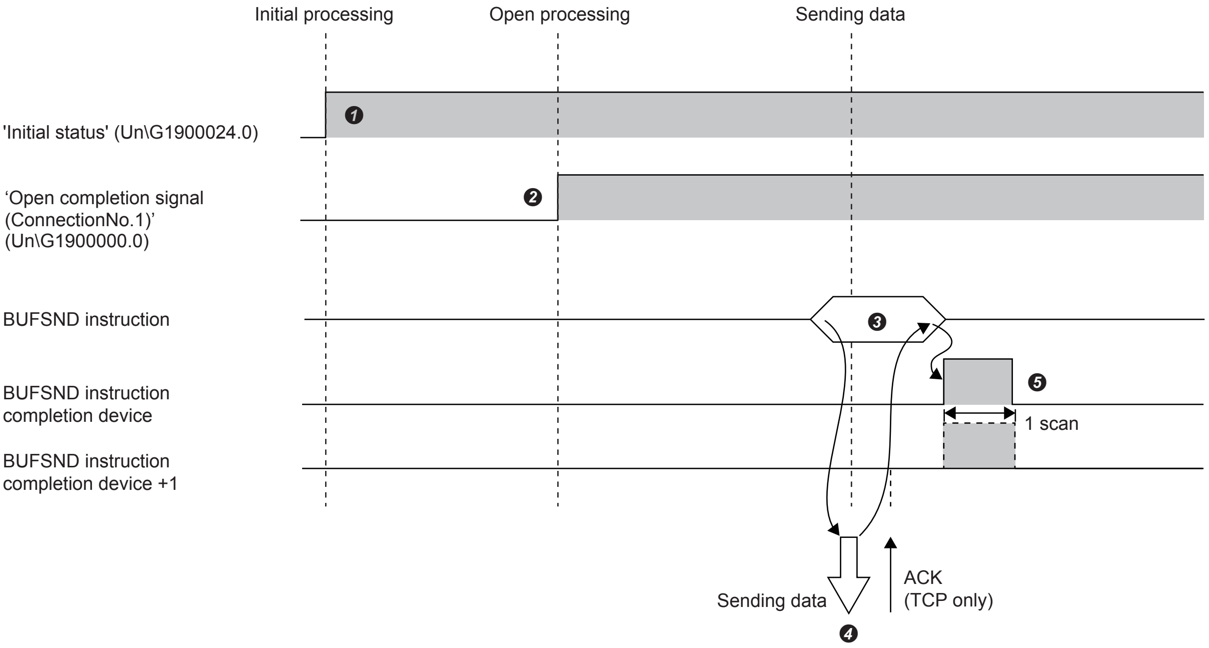
*1 Adjust the monitor timer value in the parameters.

Point

- The details of the open setting are enabled at the rising edge of the RJ71EN71 and the RnENCPU (network part) open completion signal.
- Send the next data (command) after the data communications have completed (a response has been received) for the previous data (command) send.
- When data is received/sent from/to multiple external devices, the data can be sent sequentially. However, to avoid communication troubles, it is recommended to switch the external device and send/receive the data. When a connection opened with UDP/IP is used, the setting value in the communication address setting area can be changed before sending or receiving to switch the external device.

■No procedure

The following figure shows the send processing for the fixed buffer No.1 area corresponding to the connection No.1.



Initial processing
Open processing
Sending data
'Initial status' (Un\G1900024.0)
‘Open completion signal (ConnectionNo.1)’ (Un\G1900000.0)
BUFSND instruction
BUFSND instruction completion device
BUFSND instruction completion device +1
❶
❷
❸
❹
❺
1 scan
Sending data
ACK (TCP only)

❶ Normal completion of the initial processing is checked. ('Initial status' (Un\G1900024.0): On)

❷ A connection is established between the external device and the RJ71EN71 or the RnENCPU (network part), and normal completion of the connection No.1 open processing is checked. (☞ Page 2 - 21 Open/Close Processing)

❸ The BUFSND instruction is executed. (Data is sent.)

❹ The send data of data length is sent from the fixed buffer No.1 area to the external device.

❺ The RJ71EN71 or the RnENCPU (network part) finishes data send. If the data send completes abnormally, execute the BUFSND instruction again and start the send processing.

Point

During UDP/IP communications, if the internal processing of the RJ71EN71 and the RnENCPU (network part) completes successfully, the data send processing may complete successfully even if the communication line between the CPU module and the external device is disconnected because of a connection cable disconnection or other causes. Thus, providing a communication procedure for data send/receive is recommended.

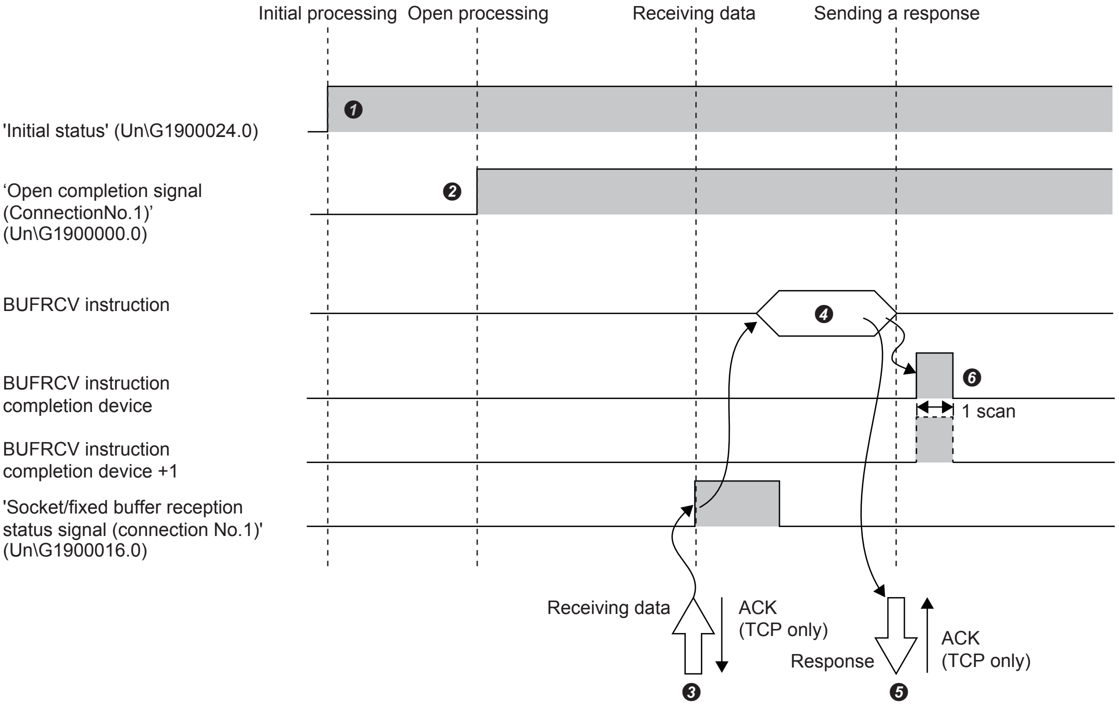
Receive procedure

This section describes the processing for the RJ71EN71 or the RnENCPU (network part) to receive data from an external device. The following receive methods can be used.

- Receiving with the main program (BUFRCV instruction)
- Receiving with an interrupt program (BUFRCVS instruction)

■Receiving with the main program (procedure exist)

The following figure shows the receive processing for the fixed buffer No.1 area corresponding to the connection No.1.


Initial processing
Open processing
Receiving data
Sending a response
'Initial status' (Un\G1900024.0)
'Open completion signal (ConnectionNo.1)' (Un\G1900000.0)
BUFRCV instruction
BUFRCV instruction completion device
BUFRCV instruction completion device +1
'Socket/fixed buffer reception status signal (connection No.1)' (Un\G1900016.0)
1 scan
Receiving data
ACK (TCP only)
Response
ACK (TCP only)
❶
❷
❸
❹
❺
❻

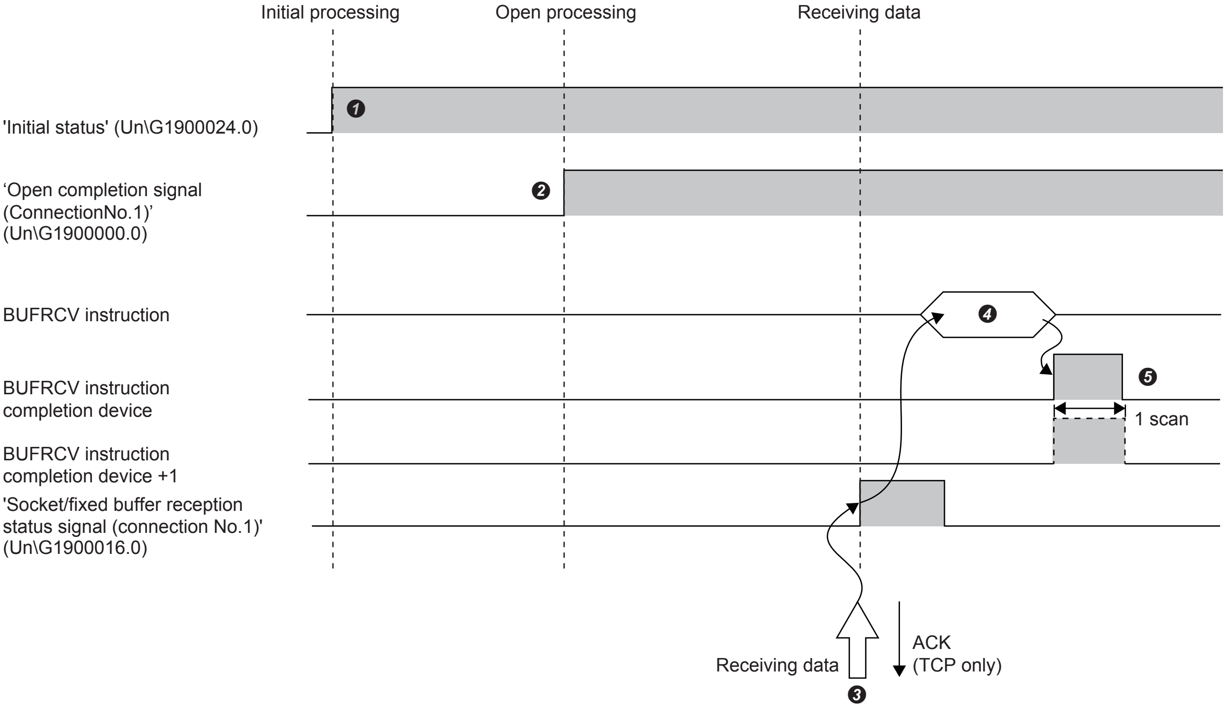
❶ Normal completion of the initial processing is checked. ('Initial status' (Un\G1900024.0): On)
❷ A connection is established between the external device and the RJ71EN71 or the RnENCPU (network part), and normal completion of the connection No.1 open processing is checked. (☞ Page 2 - 21 Open/Close Processing)
❸ Data is received from the external device. ('Socket/fixed buffer reception status signal (connection No.1)' (Un\G1900016.0): On)
❹ The BUFRCV instruction is executed, and the receive data length and receive data are read from the fixed buffer No.1. ('Socket/fixed buffer reception status signal (connection No.1)' (Un\G1900016.0): Off)
❺ When reading of the receive data length and receive data completes, a response is returned to the external device.
❻ The receive processing ends. If the data reception completes abnormally, execute the BUFRCV instruction again and start the receive processing.

Point

- The details of the open setting are enabled at the rising edge of the RJ71EN71 and the RnENCPU (network part) open completion signal.
- Execute the BUFRCV instruction when the socket/fixed buffer reception status signal turns on.
- The socket/fixed buffer receive status signal does not turn on when abnormal data is received. In addition, data is not stored in the fixed buffer No.1 area.

■Receiving with the main program (no procedure)

The following figure shows the receive processing for the fixed buffer No.1 area corresponding to the connection No.1.


Initial processing
Open processing
Receiving data
'Initial status' (Un\G1900024.0)
❶
'Open completion signal (ConnectionNo.1)' (Un\G1900000.0)
❷
BUFRCV instruction
❹
BUFRCV instruction completion device
❺
1 scan
BUFRCV instruction completion device +1
'Socket/fixed buffer reception status signal (connection No.1)' (Un\G1900016.0)
Receiving data
ACK (TCP only)
❸

❶ Normal completion of the initial processing is checked. ('Initial status' (Un\G1900024.0): On)

❷ A connection is established between the external device and the RJ71EN71 or the RnENCPU (network part), and normal completion of the connection No.1 open processing is checked. (☞ Page 2 - 21 Open/Close Processing)

❸ Data is received from the external device. ('Socket/fixed buffer reception status signal (connection No.1)' (Un\G1900016.0): On)

❹ The BUFRCV instruction is executed, and the receive data length and receive data are read from the fixed buffer No.1. ('Socket/fixed buffer reception status signal (connection No.1)' (Un\G1900016.0): Off)

❺ The receive processing ends. If the data reception completes abnormally, execute the BUFRCV instruction again and start the receive processing.

Point

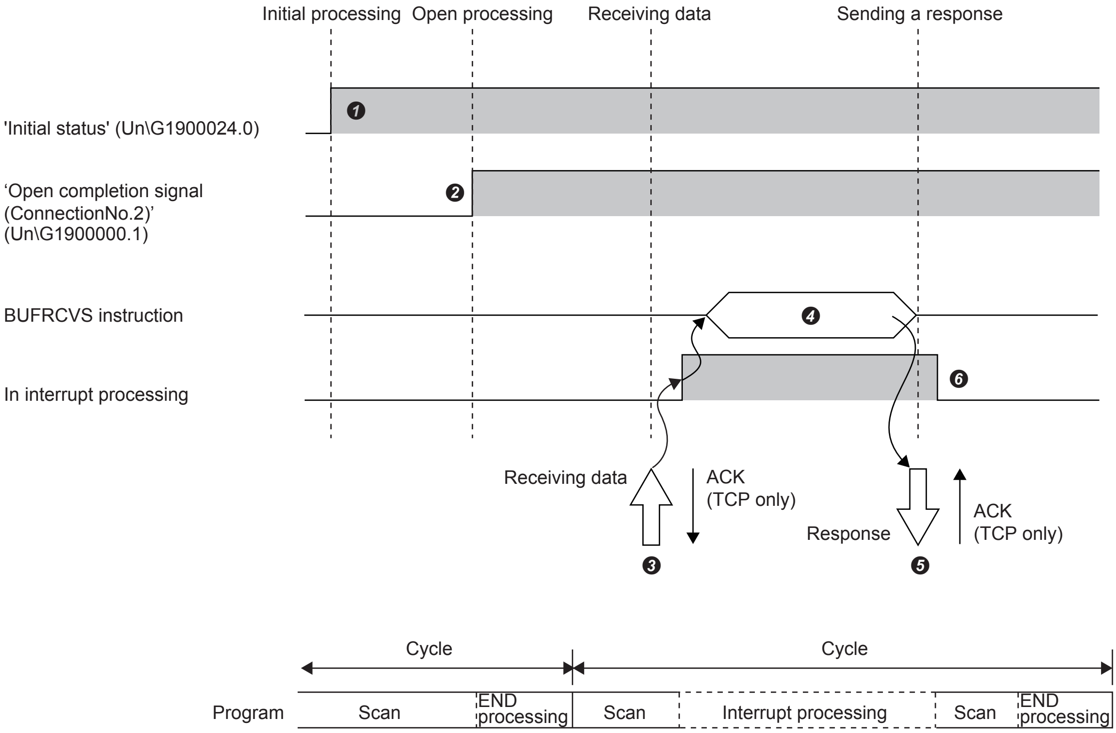
- The details of the open setting are enabled at the rising edge of the RJ71EN71 and the RnENCPU (network part) open completion signal.
- Execute the BUFRCV instruction when the socket/fixed buffer reception status signal turns on.
- The socket/fixed buffer receive status signal does not turn on when abnormal data is received. In addition, data is not stored in the fixed buffer No.1 area.

■Receiving with an interrupt program (procedure exist)

Use the BUFRCVS instruction to receive data with an interrupt program. The interrupt program is started when data is received from an external device. It enables reading the receive data from the CPU module.

The interrupt settings are required to use the interrupt program.

The following figure shows the receive processing for the fixed buffer No.2 area corresponding to the connection No.2.



❶ Normal completion of the initial processing is checked. ('Initial status' (Un\G1900024.0): On)

❷ A connection is established between the external device and the RJ71EN71 or the RnENCPU (network part), and normal completion of the connection No.2 open processing is checked. (☞ Page 2 - 21 Open/Close Processing)

❸ The CPU module is requested to start the interrupt program, and data is received from the external device. ('Socket/fixed buffer reception status signal (connection No.2)' (Un\G1900016.1): On)

❹ The interrupt program starts. The BUFRCVS instruction is executed, and the receive data length and receive data are read from the fixed buffer No.2. ('Socket/fixed buffer reception status signal (connection No.2)' (Un\G1900016.1): Off)

❺ When reading of the receive data length and receive data is completes, a response is returned to the external device.[*1]

❻ The interrupt program ends, and the main program starts.

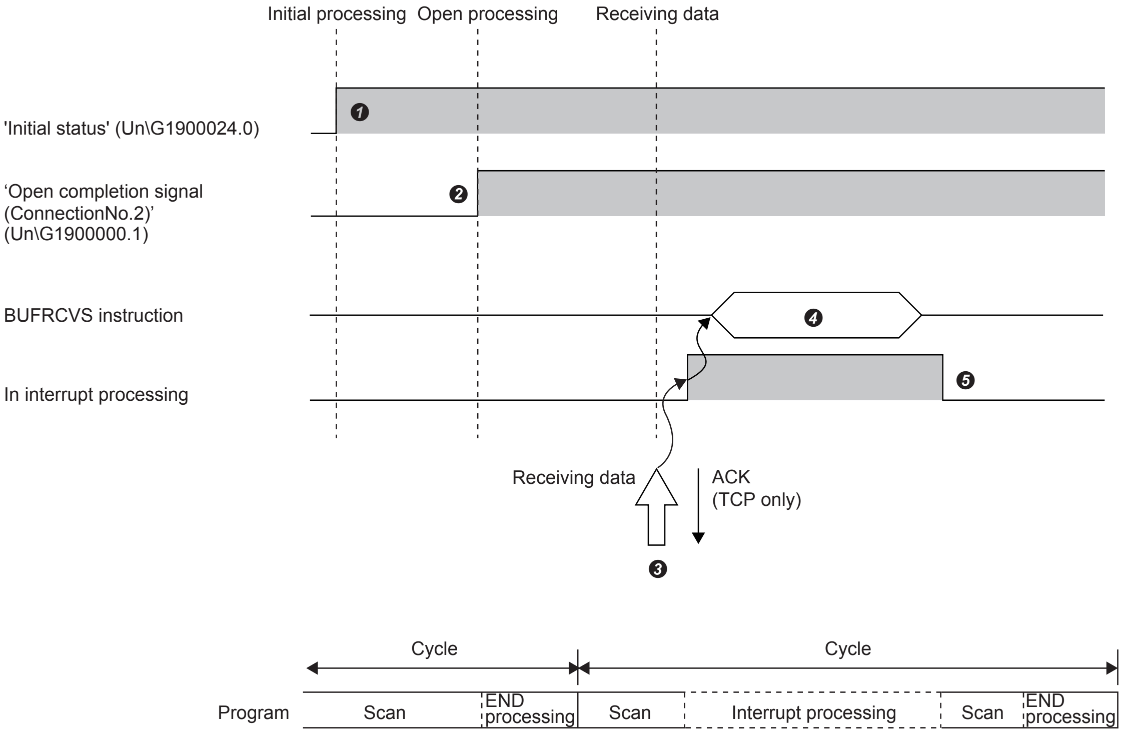
*1 A response is not returned when reading the receive data completes abnormally.

■Receiving with an interrupt program (no procedure)

Use the BUFRCVS instruction to receive data with an interrupt program. The interrupt program is started when data is received from an external device. It enables reading the receive data from the CPU module.

The interrupt settings are required to use the interrupt program.

The following figure shows the receive processing for the fixed buffer No.2 area corresponding to the connection No.2.



❶ Normal completion of the initial processing is checked. ('Initial status' (Un\G1900024.0): On)

❷ A connection is established between the external device and the RJ71EN71 or the RnENCPU (network part), and normal completion of the connection No.2 open processing is checked. (☞ Page 2 - 21 Open/Close Processing)

❸ The CPU module is requested to start the interrupt program, and data is received from the external device. ('Socket/fixed buffer reception status signal (connection No.2)' (Un\G1900016.1): On)

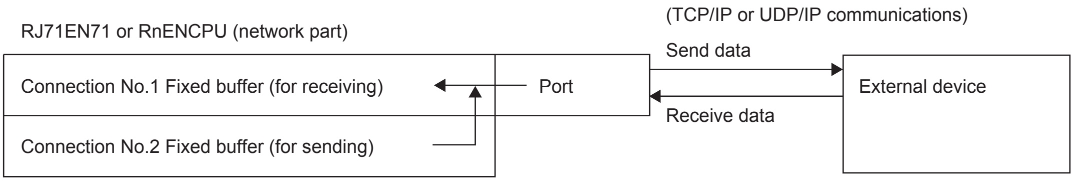
❹ The interrupt program starts. The BUFRCVS instruction is executed, and the receive data length and receive data are read from the fixed buffer No.2. ('Socket/fixed buffer reception status signal (connection No.2)' (Un\G1900016.1): Off)

❺ The interrupt program ends, and the main program starts.

Pairing open

The pairing open is an opening method that pairs a reception connection and a send connection for communications using a fixed buffer and establishes a connection between the own station and an external device in each port.

When the pairing open is specified, data can be communicated with two connections by performing the open processing for one port.


RJ71EN71 or RnENCPU (network part)
(TCP/IP or UDP/IP communications)
Send data
Connection No.1 Fixed buffer (for receiving)
Port
External device
Receive data
Connection No.2 Fixed buffer (for sending)

Point

- Only the external devices in Ethernet to which the RJ71EN71 or the RnENCPU (network part) is connected and the devices connected via a router can communicate data with the pairing open.
- The open/close processing for the next connection (send connection) is performed automatically by the open/close processing for the receive connection side set to the pairing open.

2.2.5 Communications using link dedicated instructions

Link dedicated instructions are used to communicate random data, read/write arbitrary data, and perform remote RUN/STOP to a CPU module in another network or Ethernet. If the initial processing completed successfully on the RJ71EN71 or the RnENCPU (network part) of the own station, relay station, and access station, it enables access to other stations with link dedicated instructions.

This section describes a procedure to access a CPU module in another station in the same Ethernet network.

To communicate data with the CPU module in another station in a different network, refer to the following.

📖 MELSEC iQ-R Ethernet User's Manual (Application)

■Precautions

The CPU module (CPU part for the RnENCPU) cannot communicate data with the link dedicated instructions.

Applicable dedicated instructions

The following table lists the link dedicated instructions used for data communications.

Instruction	Description
SEND	Sends data to another station.
RECV	Reads the receive data from another station (for main programs).
RECVS	Reads the receive data from another station (for interrupt programs).
READ	Reads data from the word device of another station.
SREAD	Reads data from the word device of another station (with a completion device).
WRITE	Writes data to the word device of another station.
SWRITE	Writes data to the word device of another station (with a completion device).
REQ	Requests the remote RUN/STOP to the CPU module on another station.
	Reads/writes clock data from/to another station.
ZNRD	Reads data from the word device of another station (ACPU).
ZNWR	Writes data to the word device of another station (ACPU).

Point

For details on link dedicated instructions, refer to the following.

📖 MELSEC iQ-R Programming Manual (Instructions, Standard Functions/Function Blocks)

Data communication procedure

This section describes a procedure for communications using link dedicated instructions.

1. After the module parameters are set, normal completion of the initial processing on the RJ71EN71 or the RnENCPU (network part) is checked. ('Initial status' (Un\G1900024.0): On)
2. Data is communicated with link dedicated instructions.

Point

The open/close processing is not required because communications using link dedicated instructions use the MELSOFT transmission port (UDP/IP).

2.2.6 Connection with MELSOFT products and GOTs

Programming and monitoring of the programmable controller with the engineering tool, and monitoring and testing of the programmable controller from the GOT can be performed over Ethernet. This function enables remote operations by using a long-distance connection and high-speed communications of Ethernet.

The following table lists methods to connect the Ethernet-equipped module with MELSOFT products (such as engineering tool and MX Component) and GOTs.

○: Connection available, ×: Connection not available

Connection method	Purpose	Availability			
		MELSOFT product		GOT	
		RJ71EN71, RnENCPU (network part)	CPU module (built-in Ethernet port part)	RJ71EN71, RnENCPU (network part)	CPU module (built-in Ethernet port part)
Connection via a hub (Connection by specifying an IP address)	• To connect to the Ethernet-equipped module that has no network number and station number • To connect multiple MELSOFT products	○	○	×	○
Connection via a hub (Connection by specifying a network number and a station number)	• To connect by using a network number and a station number • To connect multiple MELSOFT products and GOTs	○	×	○	×
Direct connection (Connection without specifying an IP address, a network number, or a station number)[*1]	• To connect without a hub using one Ethernet cable for one-on-one communications with an external device • To connect to the Ethernet-equipped module whose IP address is unknown	○	○	×	×

[*1] This connection method is not available when the RJ71EN71 network type is set to "Q Compatible Ethernet".

Point

For the procedure to connect the Ethernet-equipped module and GOT, refer to the following.

📖 Manual for the GOT used

2.3 Open/Close Processing

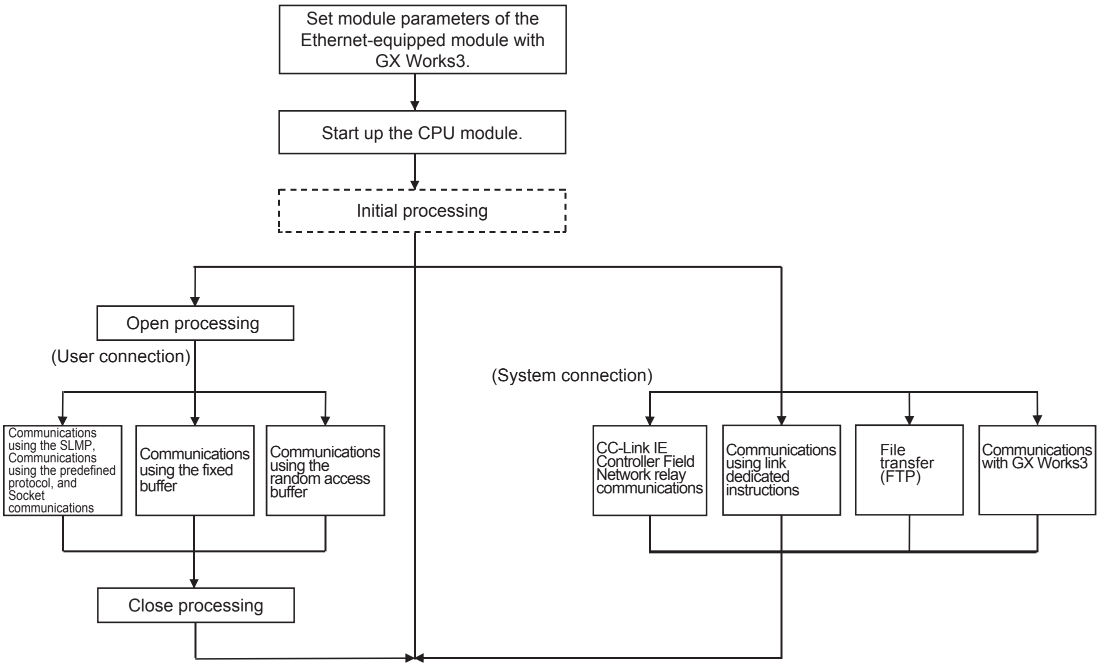
To perform data communications with an external device, a connection must be established between the external device and an Ethernet-equipped module (connection of a logic circuit) after completion of the initial processing performed automatically at start-up of the Ethernet-equipped module.

For the Ethernet-equipped module, there are two types of connections: system connection and user connection. Performing the open processing on all connections enables data communications with external devices.

After the communications complete, the close processing is performed on the established connection.

This section describes the connections, open processing, and close processing of the Ethernet-equipped module.

(Data communication procedure)


Set module parameters of the Ethernet-equipped module with GX Works3.
Start up the CPU module.
Initial processing
Open processing
(User connection)
(System connection)
Communications using the SLMP, Communications using the predefined protocol, and Socket communications
Communications using the fixed buffer
Communications using the random access buffer
CC-Link IE Controller Field Network relay communications
Communications using link dedicated instructions
File transfer (FTP)
Communications with GX Works3
Close processing

Connections of the Ethernet-equipped module

■System connection

- This connection is used to perform data communications using the special functions of the Ethernet-equipped module.
- The open processing is automatically performed at start-up of the Ethernet-equipped module.
  This connection does not require the open/close processing by a user.

■User connection

- This connection is used to perform data communications using the basic functions of the Ethernet-equipped module.
- A user performs the open processing when starting data communications with external devices and the close processing when finishing the data communications.
- A connection is established according to the communication method for data communications with external devices.

[TCP/IP communications]

- A connection is established when the open processing completes successfully.
- A connection is disconnected when the close processing completes successfully after completion of data communications.
- There are two methods to establish a connection: Active open and Passive open.

(Active open)

- The Ethernet-equipped module side issues a connection establishment request to a specified external device in the standby state.
  Compared to the telephone line, the Ethernet-equipped module side is a caller.

(Passive open)

- There are two types of Passive open: Fullpassive open and Unpassive open.
  Compared to the telephone line, the Ethernet-equipped module side is a receiver.

- There are two methods to perform the open/close processing by Passive open on the Ethernet-equipped module side: Processing by a CPU module and Processing by an Ethernet-equipped module system.*1

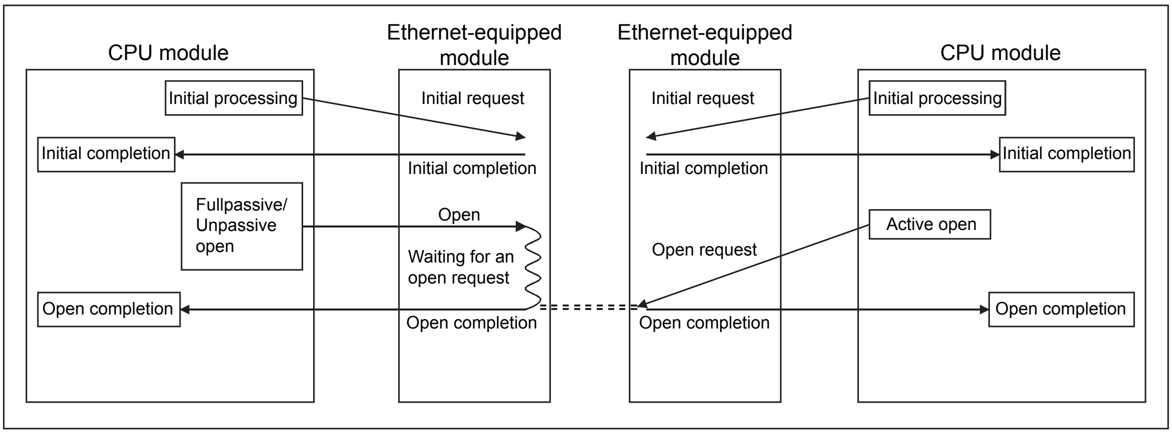
(Fullpassive open)

- The Ethernet-equipped module side waits for a connection establishment request (Active open) addressed to an own station from a specified external device.

(Unpassive open)

- The Ethernet-equipped module side waits for a connection establishment request (Active open) addressed to an own station from an external device.

(Connection establishment procedure)



[UDP/IP communications]

- The open/close processing is performed in the Ethernet-equipped module to enable data communications with external devices.
- A connection is established during data communications after the open processing completes successfully.
- There are two methods to perform the open/close processing on the Ethernet-equipped module side in the UDP/IP communications: Processing by a CPU module and Processing by an Ethernet-equipped module system.*1

*1 The module parameter setting for the Ethernet-equipped module of GX Works3 determines the used method.
• The open/close processing by Passive open in the TCP/IP communications
• The open/close processing in the UDP/IP communications

(1) The Ethernet-equipped module system performs the open/close processing on a connection which has the following settings 1) and 2) in the module parameter setting for the Ethernet-equipped module of GX Works3.
The open/close processing by a sequence program is not required for a connection used for data communications with an external device.
1) "Do Not Open by Program" is set to "Opening Method" of the module parameter. (This setting is common to all user connections.)
2) For a user connection, "Protocol" and an Ethernet device are set as follows in "External Device Configuration".
• "Protocol" = TCP, Model name = Unpassive Connection Module
• "Protocol" = TCP, Model name = Fullpassive Connection Module
• "Protocol" = UDP, Model name = UDP Connection Module

Point

When the Ethernet-equipped module is started up with the above settings

(Connections where TCP is set to "Protocol")

- The open processing (Active open) from an external device opens the connection and enables data communications.
- When the data communications end, the close processing from an external device disconnects the connection.
- Perform the open/close processing from external devices and communicate data if needed.
  However, when closing a connection from the CPU module side due to a communication error while the connection is open, perform the close processing by using the CLOSE instruction in a sequence program.
  In this case, the Ethernet-equipped module system does not set the connection in an open standby state.
  All the subsequent open processing (using the OPEN instruction) and close processing on the Ethernet-equipped module side must be performed using a sequence program.

(Connections where UDP is set to "Protocol")

Starting up the Ethernet-equipped module enables data communications with external devices.

(2) A sequence program performs the open/close processing on a connection which has either of the following settings 1) and 2) in the module parameter setting for the Ethernet-equipped module of GX Works3.
The Ethernet-equipped module system does not perform the open/close processing.
1) "Open by Program" is set to "Opening Method" of the module parameter. (This setting is common to all user connections.)
2) For a user connection, "Protocol" and an Ethernet device are set as follows in "External Device Configuration".
- "Protocol" = TCP, Model name = Active Connection Module

Open processing

The open processing enables data communications with external devices.

■TCP/IP communications

- The Ethernet-equipped module system communicates with an external device by using a connection used for data communications and establishes a connection (connection of a logic circuit).
- After the open processing completes successfully, a user can communicate data through the connection.

■UDP/IP communications

- The Ethernet-equipped module system performs internal processing.
- After the open processing completes successfully, a user can communicate data through the connection opened.

The initial processing must have completed successfully before the open processing.

The open processing can be performed on up to 128 external devices.

To communicate data with the same external device in communications using a fixed buffer, two fixed buffers are required. Therefore, the number of external devices for data communications is reduced.

Point

Pay attention to the following point when performing communications using the random access buffer.

- To continue data communications even after the operating status of the CPU module on the station where the Ethernet-equipped module is mounted is changed to STOP, set "Opening Method" to "Do Not Open by Program".

Close processing

The close processing disables data communications with external devices.

■TCP/IP communications

- The Ethernet-equipped module system communicates with an external device by using a connection used for data communications and disconnects the connection (disconnection of a logic circuit).
- After the close processing completes successfully, a user can change an external device used for data communications thorough the connection.

■UDP/IP communications

- The Ethernet-equipped module system performs internal processing.
- After the close processing completes successfully, a user can change an external device used for data communications thorough the connection.

The close processing is performed mainly in the following cases.

- Finishing communications with an external device
- Changing an external device to communicate
- Changing communication conditions

For a connection on which a sequence program performs the open processing, perform the close processing by using a sequence program.

Determine a timing for the close processing with the external device.

Point

Even if the close processing is not requested, the open completion signal (address: corresponding bit of 1900000) automatically turns off and the communication line is closed in the following cases:

(1) A timeout of the alive check function occurs.

(2) The CLOSE instruction or the ABORT(RST) instruction is received from an external device.

(3) The Ethernet-equipped module is in the open completion state and receives the Active open request again from an external device in TCP/IP communications.

(4) A timeout occurs at TCP send.

2.3.1 TCP/IP communications

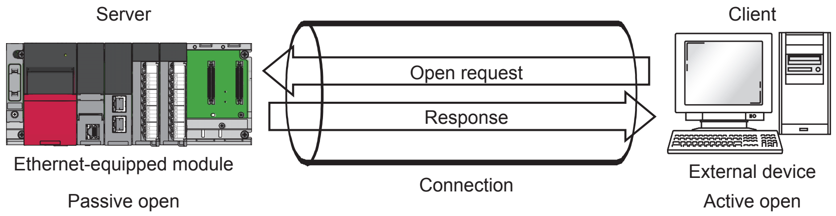
This section describes TCP/IP communications.

Establishing a connection

In TCP/IP communications, a connection must be established between communicating devices. If the server side device performs the Passive open processing and is in the standby state, the client side device issues an open request (Active open processing) to the server. When a response is returned, a connection is established.
In TCP/IP communications, a connection is established during communications. Since the communicating devices communicate data while checking that the data has correctly reached the communication destination, the data reliability can be ensured. Note that the line load is larger than that in UDP/IP communications.

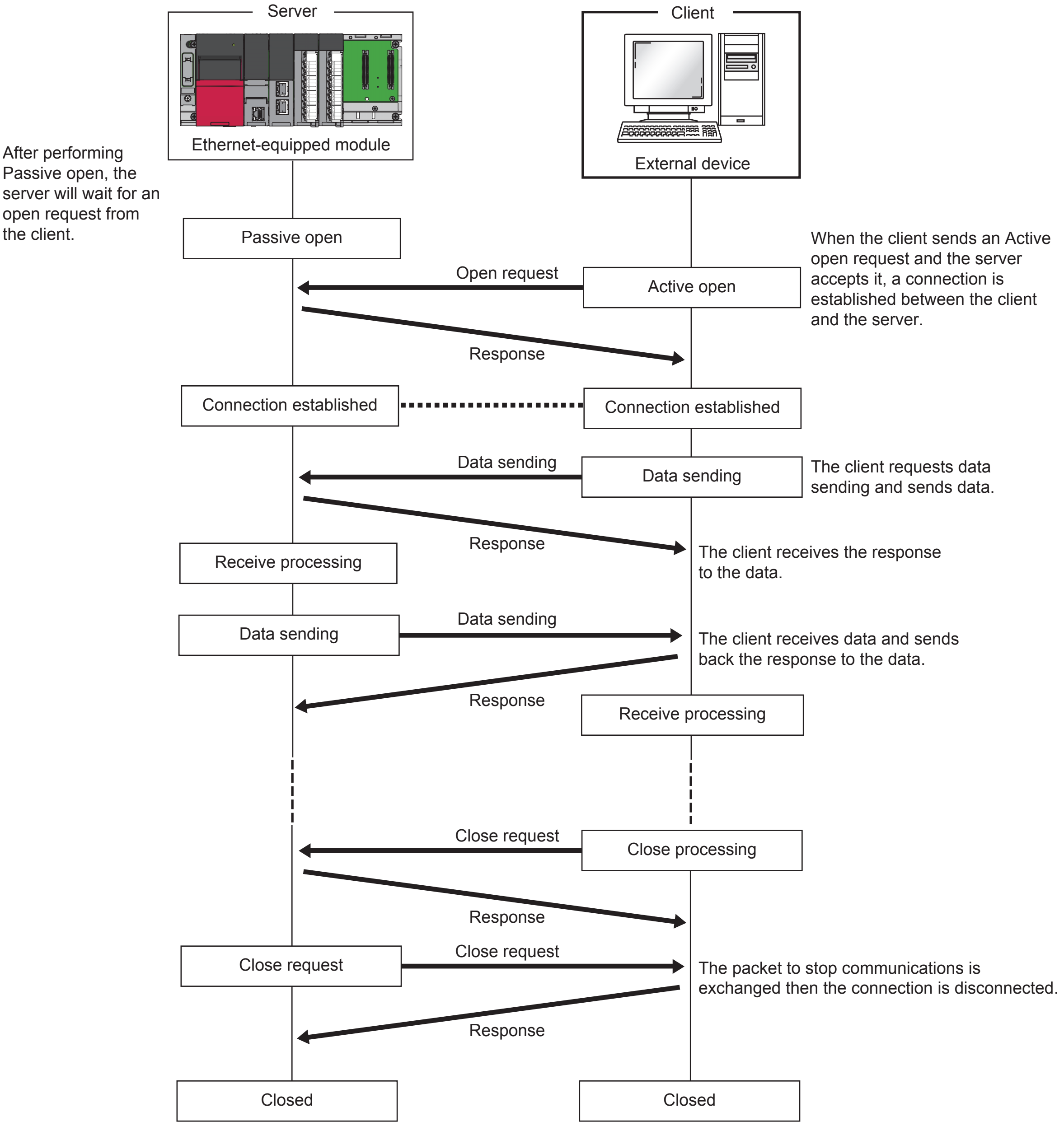
Ex.

When the Ethernet-equipped module is Passive open



Communication process

The following figure shows the process from the establishment of connection to the end of communication.


Server
Ethernet-equipped module
Client
External device
After performing Passive open, the server will wait for an open request from the client.
Passive open
Open request
Active open
When the client sends an Active open request and the server accepts it, a connection is established between the client and the server.
Response
Connection established
Connection established
Data sending
Data sending
The client requests data sending and sends data.
Response
Receive processing
The client receives the response to the data.
Data sending
Data sending
The client receives data and sends back the response to the data.
Response
Receive processing
Close request
Close processing
Response
Close request
Close request
The packet to stop communications is exchanged then the connection is disconnected.
Response
Closed
Closed

Point

Wait for at least 500ms or more before performing the open processing again after the close request is sent from the external device to the Ethernet-equipped module.

Active open procedure

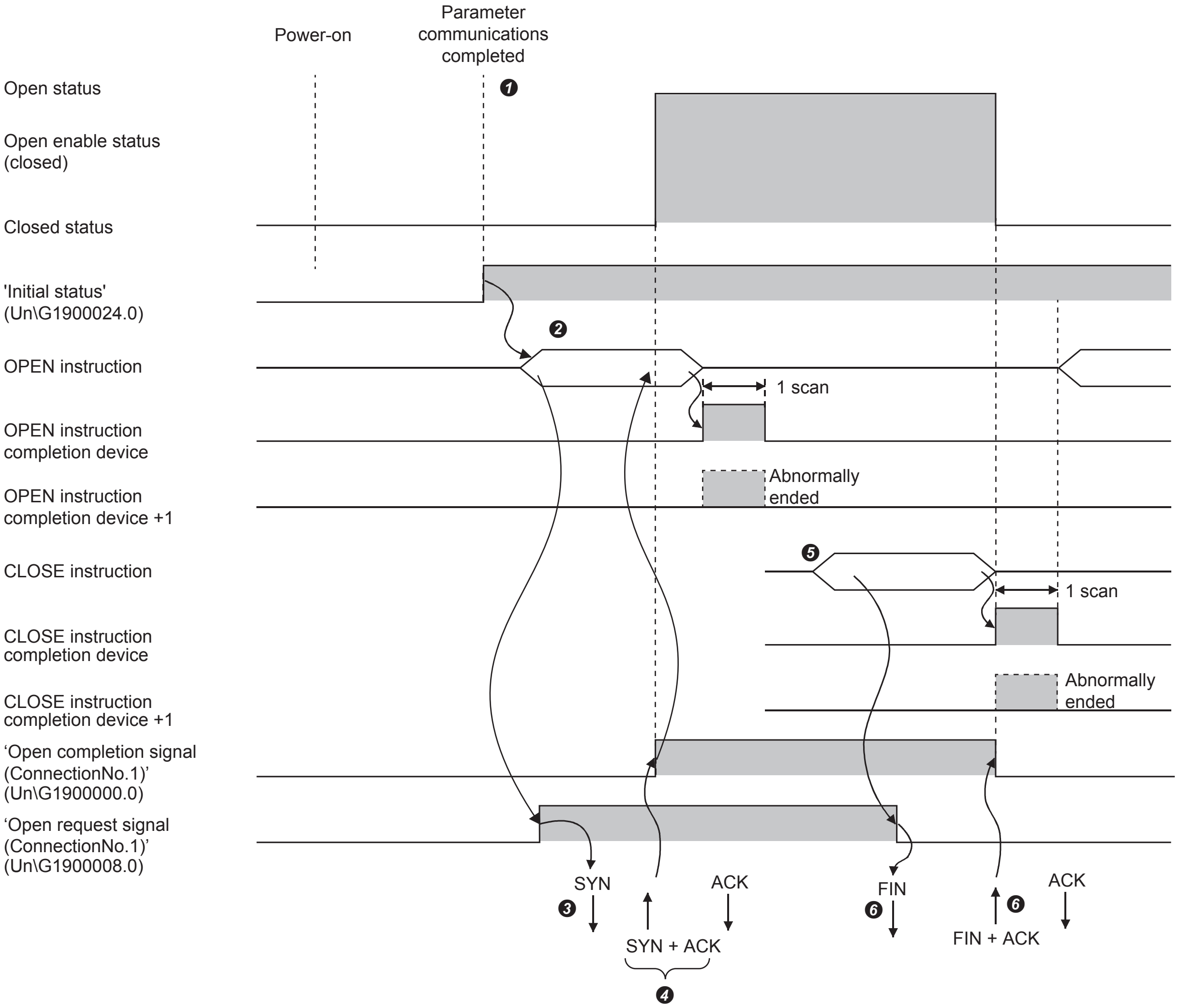
Active open is a connection method that performs the active open processing to an external device (Passive open) that is in a passive open standby state for a connection. The following figure shows a process that the Ethernet-equipped module performs the active open processing.

For the OPEN/CLOSE instruction, refer to the following.

📖 MELSEC iQ-R Programming Manual (Instructions, Standard Functions/Function Blocks)

Ex.

Open/close processing for the connection No.1


Power-on
Parameter communications completed
❶
Open status
Open enable status (closed)
Closed status
'Initial status' (Un\G1900024.0)
❷
OPEN instruction
1 scan
OPEN instruction completion device
Abnormally ended
OPEN instruction completion device +1
❺
CLOSE instruction
1 scan
CLOSE instruction completion device
Abnormally ended
CLOSE instruction completion device +1
'Open completion signal (ConnectionNo.1)' (Un\G1900000.0)
'Open request signal (ConnectionNo.1)' (Un\G1900008.0)
SYN
❸
SYN + ACK
ACK
❹
FIN
❻
FIN + ACK
❻
ACK

❶ After setting the module parameters, check that the initial processing for the Ethernet-equipped module has completed successfully. ('Initial status' (Un\G1900024.0): On)

❷ Start the open processing by using the OPEN instruction.[*3] ('Open request signal (connection No.1)' (Un\G1900008.0): On)

❸ The Ethernet-equipped module performs the open processing. (The module sends the open request (SYN) to the external device.)

❹ Data can be communicated after the open processing completes successfully.[*1]

❺ The close processing is started by using the CLOSE instruction. ('Open request signal (connection No.1)' (Un\G1900008.0): Off)

❻ The Ethernet-equipped module performs the close processing. (The module sends the close request (FIN) to external device.)

❼ Data communications end when the close processing completes successfully.[*2]

*1 If RST is returned from the external device after SYN is sent from the Ethernet-equipped module, the open processing completes with an error immediately.

*2 If ACK or FIN is not returned even after the TCP end timer time, the Ethernet-equipped module forcibly disconnects the connection (sends RST). (The close processing completes with an error.)

*3 If the open processing target port has not been linked up, the OPEN instruction will complete with an error. Perform the open processing again after link-up, or check that 'Connection status' (Un\G5192) is set to 1 before starting the open processing. If auto-negotiation fails, the open processing will complete with an error. Perform the open processing again after a while.
If the open processing completes with an error again, check the Ethernet cable connection or the operation of the external device and switching hub.

Passive open procedure

The following two types of connection methods can be used for Passive open of an Ethernet-equipped module.

Connection method	Description
Unpassive	Performs the passive open processing for the connection to all devices connected to the network without restriction to the IP address or port number of the communication destination.
Fullpassive	Performs the passive open processing for the connection of a specific external device when the IP address and port number of the communication destination are specified.

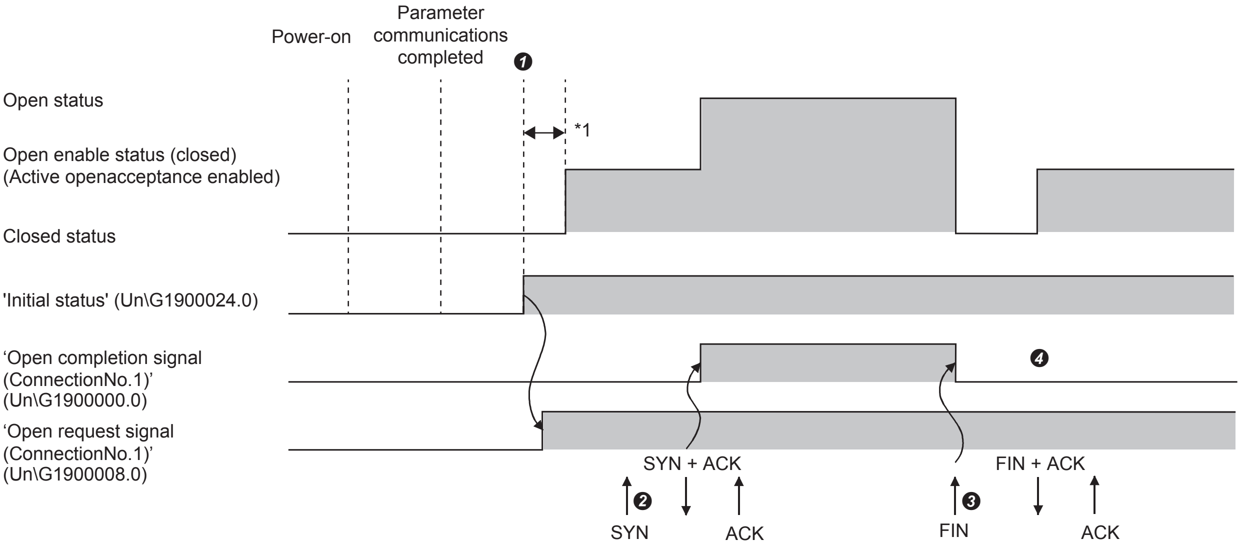
The open/close processing procedure for Passive open follows the setting of "Opening Method" under "Own Node Settings" in "Basic Settings". (☞ Page 4 - 13 Basic Settings)

■When "Do Not Open by Program" is set

Since the Ethernet-equipped module is constantly in the open standby state, a connection is established when Active open is initiated by an external device. This operation eliminates the need for an open/close processing program on the Ethernet-equipped module side.

Ex.

Open/close processing for the connection No.1


Power-on
Parameter communications completed
❶
Open status
*1
Open enable status (closed)
(Active openacceptance enabled)
Closed status
'Initial status' (Un\G1900024.0)
‘Open completion signal (ConnectionNo.1)’ (Un\G1900000.0)
❹
‘Open request signal (ConnectionNo.1)’ (Un\G1900008.0)
SYN + ACK
FIN + ACK
❷
❸
SYN
ACK
FIN
ACK

❶ After setting the module parameters, check that the initial processing for the Ethernet-equipped module has completed successfully. ('Initial status' (Un\G1900024.0): On) When the initial processing completes successfully, the connection enters the open enable state, and the module waits for the open request from the external device.

❷ The Ethernet-equipped module performs the open processing when the open request (SYN) is received from the external device. When the open processing completes successfully, 'Open completion signal (connection No.1)' (Un\G1900000.0) turns on and data communications are enabled.

❸ The Ethernet-equipped module performs the close processing when the close request (FIN) is received from the external device. When the close processing completes successfully, the open completion signal turns off and data communications are disabled.

❹ After the internal processing in the Ethernet-equipped module completes, the connection waits for the open request again.

*1 The open request (SYN) received between the initial processing normal completion and the open request standby state is handled as an error, and the Ethernet-equipped module sends a connection forced close (RST) (to the external device that sent the open request (SYN)).

Point

When the open/close processing is performed by a dedicated instruction from the Ethernet-equipped module, even if "Do Not Open by Program" is set in "Opening Method" under "Own Node Settings" in "Basic Settings", the connection will not return to the open request standby state after the close processing completes.

■When "Open by Program" is set

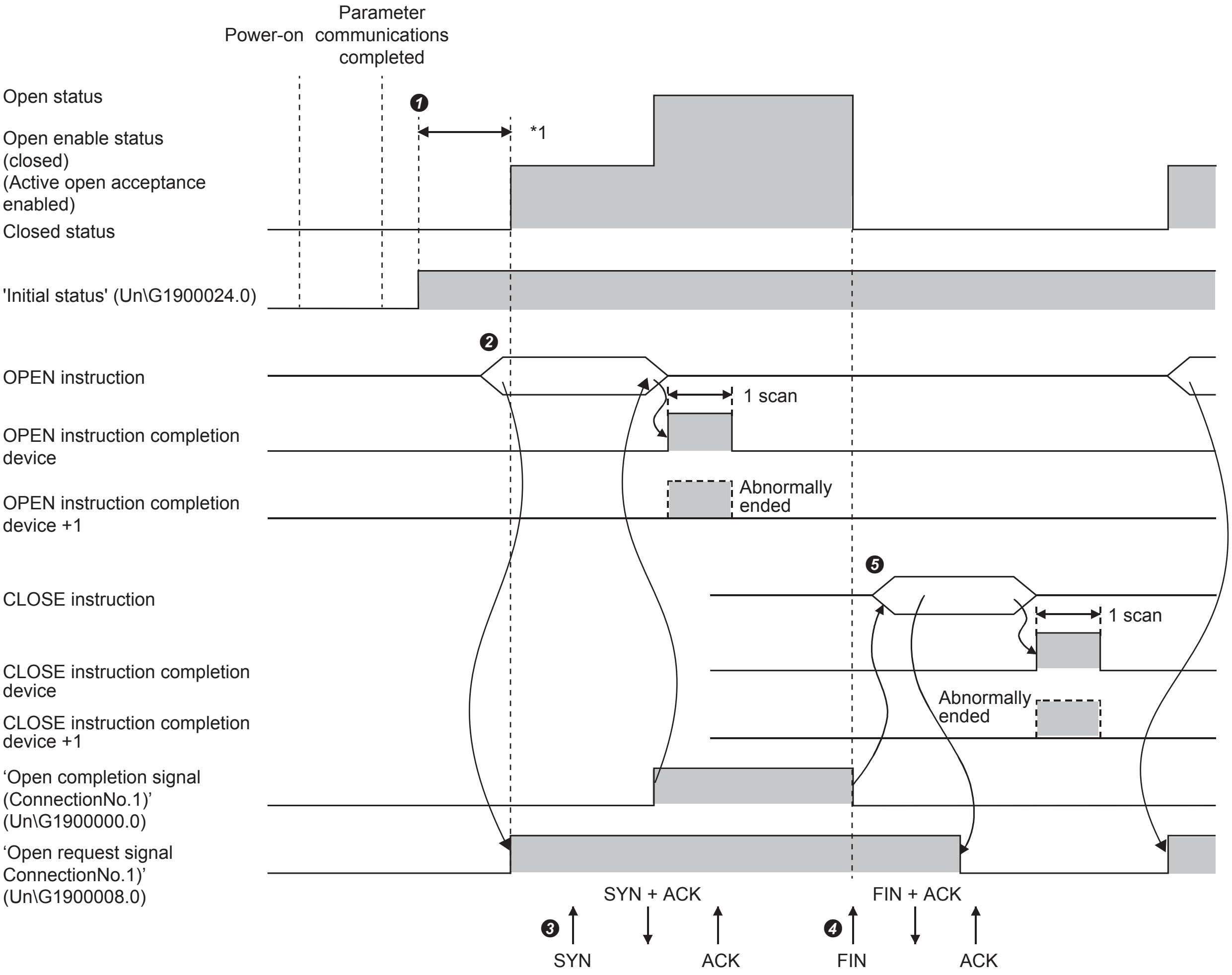
Before the open/close request is received from an external device, the Ethernet-equipped module must execute the OPEN/ CLOSE instruction and enter the open/close standby state. Data can be sent and received after the open processing is completes successfully.

For the OPEN/CLOSE instruction, refer to the following.

📖 MELSEC iQ-R Programming Manual (Instructions, Standard Functions/Function Blocks)

Ex.

Open/close processing for the connection No.1



Power-on
Parameter communications completed
Open status
Open enable status (closed) (Active open acceptance enabled)
Closed status
'Initial status' (Un\G1900024.0)
OPEN instruction
OPEN instruction completion device
OPEN instruction completion device +1
CLOSE instruction
CLOSE instruction completion device
CLOSE instruction completion device +1
'Open completion signal (ConnectionNo.1)' (Un\G1900000.0)
'Open request signal ConnectionNo.1)' (Un\G1900008.0)
❶
❷
❸
❹
❺
*1
1 scan
Abnormally ended
1 scan
Abnormally ended
SYN + ACK
FIN + ACK
SYN
ACK
FIN
ACK

❶ After setting the module parameters, check that the initial processing for the Ethernet-equipped module has completed successfully. ('Initial status' (Un\G1900024.0): On)

❷ The open processing is started by using the OPEN instruction. ('Open request signal (connection No.1)' (Un\G1900008.0): On)

❸ The Ethernet-equipped module performs the open processing when the open request (SYN) is received from the external device. When the open processing completes successfully, 'Open completion signal (connection No.1)' (Un\G1900000.0) turns on and data communications are enabled.

❹ The Ethernet-equipped module performs the close processing when the close request (FIN) is received from the external device. When the close processing completes successfully, the open completion signal turns off and data communications are disabled.

*1 The open request (SYN) received between the initial processing normal completion and the open request standby state is handled as an error, and the Ethernet-equipped module sends a connection forced close (RST) (to the external device that sent the open request (SYN)).

Point

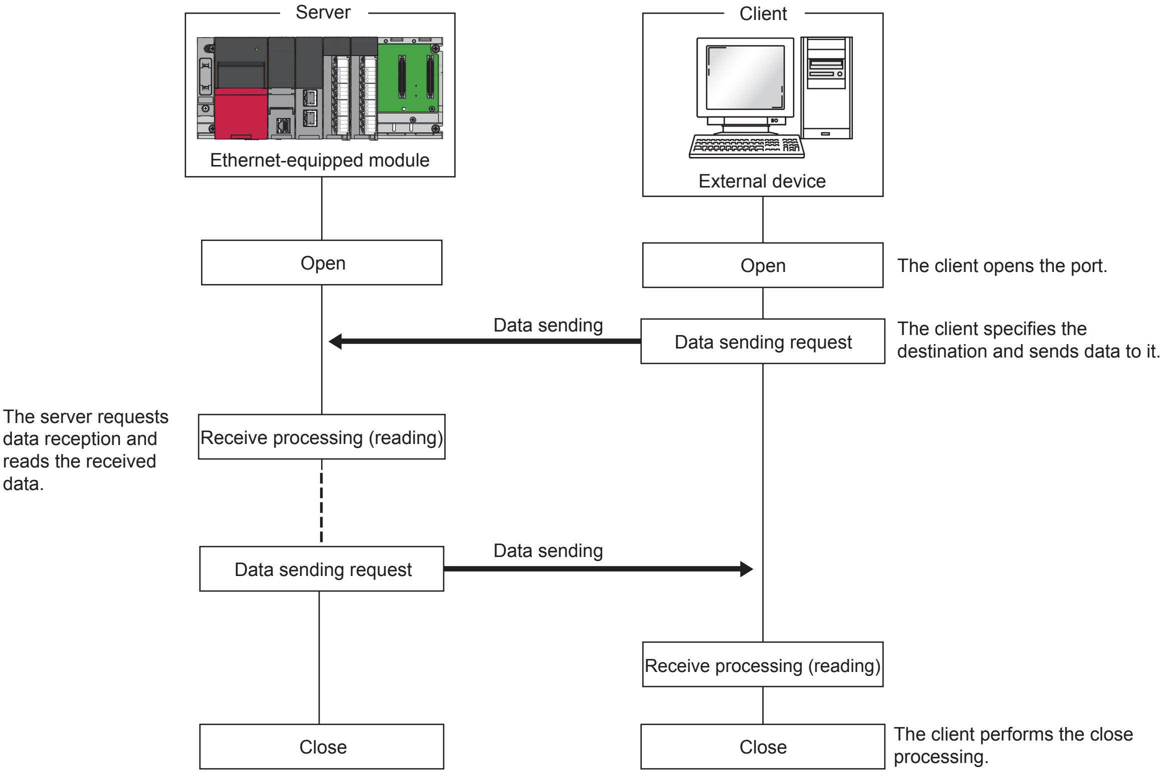
- Change the connection setting before executing the OPEN instruction.
- After the open processing is performed, the open request cannot be canceled until the open processing completes. Perform the close processing (the CLOSE instruction) after the open processing completes.

2.3.2 UDP/IP communications

This section describes UDP/IP communications. In UDP/IP communications, a connection is not established during communications. Since the communicating devices do not check that the communication destination has correctly received the data, the line load becomes smaller. Note that the data reliability is smaller than that in TCP/IP communications.

Communication process

UDP/IP communications do not require a process to establish a connection with the external device as is required with TCP/IP communications.


Server
Ethernet-equipped module
Client
External device
Open
Open
The client opens the port.
Data sending
Data sending request
The client specifies the destination and sends data to it.
The server requests data reception and reads the received data.
Receive processing (reading)
Data sending request
Data sending
Receive processing (reading)
Close
Close
The client performs the close processing.

Point

Wait for at least 500ms or more before performing the open processing again after the close request is sent from the external device to the Ethernet-equipped module.

Open procedure

According to the setting of "Opening Method" under "Own Node Settings" in "Basic Settings", the open/close processing procedure is as follows. (☞ Page 4 - 13 Basic Settings)

■When "Do Not Open by Program" is set

After the Ethernet-equipped module mounted station starts up, the UDP/IP communication setting connection automatically opens, and data send/receive is enabled. Programs for open/close processing are not required.

Point

When the open/close processing is performed by a dedicated instruction from the Ethernet-equipped module, even if "Do Not Open by Program" is set in "Opening Method" under "Own Node Settings" in "Basic Settings", all the subsequent open/close processing must be performed by using a program after a connection with the external device is established.

■When "Open by Program" is set

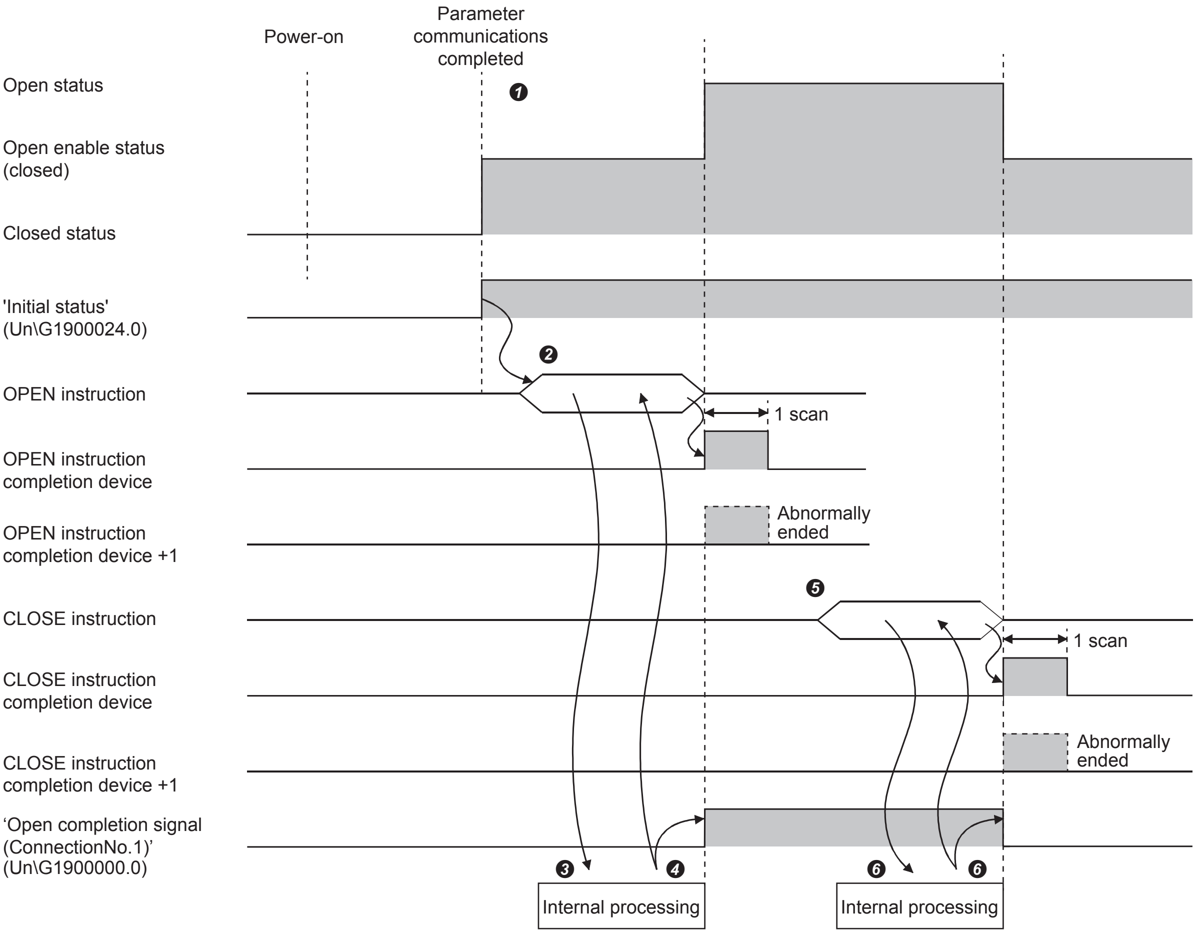
Before the open/close request is received from an external device, the Ethernet-equipped module must execute the OPEN/CLOSE instruction and enter the open/close standby state. Data can be sent and received after the open processing is completes successfully.

For the OPEN/CLOSE instruction, refer to the following.

📖 MELSEC iQ-R Programming Manual (Instructions, Standard Functions/Function Blocks)

Ex.

Open/close processing for the connection No.1


Power-on
Parameter communications completed
Open status
Open enable status (closed)
Closed status
'Initial status' (Un\G1900024.0)
OPEN instruction
OPEN instruction completion device
OPEN instruction completion device +1
CLOSE instruction
CLOSE instruction completion device
CLOSE instruction completion device +1
'Open completion signal (ConnectionNo.1)' (Un\G1900000.0)
1 scan
Abnormally ended
1 scan
Abnormally ended
Internal processing
Internal processing
❶
❷
❸
❹
❺
❻
❻

❶ After setting the module parameters, check that the initial processing for the Ethernet-equipped module has completed successfully. ('Initial status' (Un\G1900024.0): On)

❷ The open processing is started by using the OPEN instruction. ('Open request signal (connection No.1)' (Un\G1900008.0): On)

❸ The Ethernet-equipped module performs the open processing (only internal processing).

❹ Data can be communicate after the open processing completes successfully.

❺ The close processing is started by using the CLOSE instruction. ('Open request signal (connection No.1)' (Un\G1900008.0): Off)

❻ The Ethernet-equipped module performs the close processing (only internal processing).

❼ The data communications end when the close processing completes successfully.

MEMO

3 SYSTEM CONFIGURATION

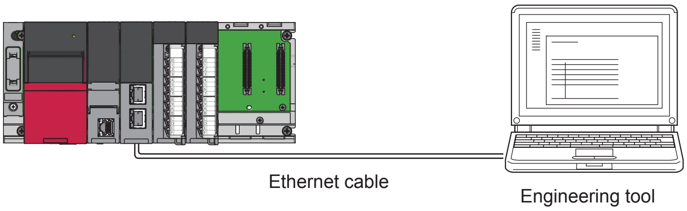
3.1 Ethernet Configuration

3

This section describes the system configuration of Ethernet.

Direct connection to an engineering tool

The engineering tool can be directly connected only with a single Ethernet cable, without using a hub.
In direct connection, communications can be performed simply by the transfer setup, without setting an IP address.

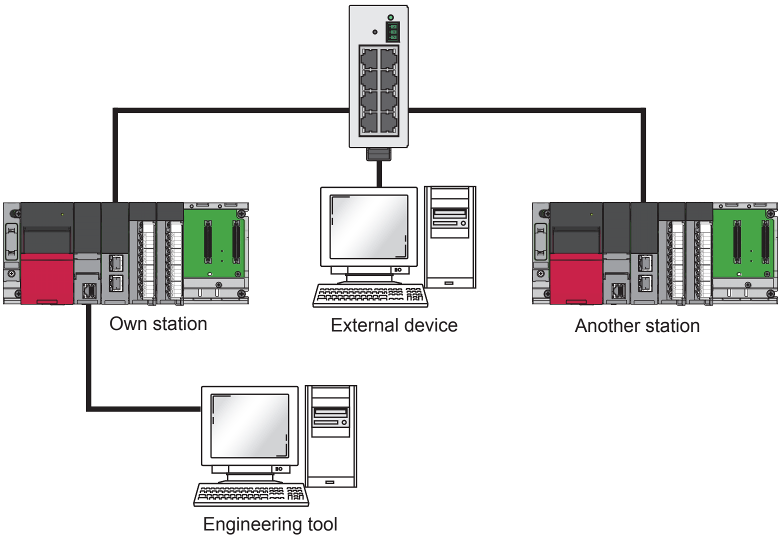


Direct connection can cause an unauthorized connection from a remote location as an Ethernet cable is longer than a USB cable.
Unauthorized connection can be prevented by selecting "Disable" in the following setting.

[Navigation window] ⇨ [Parameter] ⇨ Target module ⇨ [Module Parameter] ⇨ [Application Settings] ⇨ [Security] ⇨ [Disable Direct Connection with MELSOFT]

Connection with an external device and another station

Connecting the module with an external device and another station over Ethernet enables the collection or modification of programmable controller data, monitoring of CPU module operation, status control, and data communications.



MEMO

4 ETHERNET-EQUIPPED MODULE SPECIFICATIONS/SETTINGS AND PROCEDURE BEFORE OPERATION

4.1 Performance Specifications

The following table lists the performance specifications of Ethernet.

Item			RJ71EN71		CPU module		
			Ethernet	Q-compatible Ethernet	Built-in Ethernet port part	RnENCPU (CPU part)	RnENCPU (network part)
Transmission specifications	Data transmission speed		1Gbps/100Mbps/10Mbps	1Gbps[1]/100Mbps/10Mbps	100Mbps/10Mbps		1Gbps/100Mbps/10Mbps
	Communication mode	1000BASE-T	Full-duplex	—			Full-duplex
		100BASE-TX	Full-duplex/half-duplex				
		10BASE-T	Full-duplex/half-duplex				
	Interface		RJ45 connector (Auto MDI/MDI-X)				
	Transmission method		Base band				
	Maximum frame size		• 1518 bytes • 9022 bytes (when jumbo frames are used)	1518 bytes			• 1518 bytes • 9022 bytes (when jumbo frames are used)
	Jumbo frame		Available	Not available			Available
	Maximum segment length		100m (length between hub and node)[2]				
	Number of cascade connections	1000BASE-T	[3]		—		[3]
		100BASE-TX	2 levels maximum[4]				
		10BASE-T	4 levels maximum[4]				
	IP version		Compatible with IPv4				
Send/receive data storage memory	Number of simultaneous open connections		128 connections (connections usable on a program)	16 connections (connections usable on a program)			64 connections (connections usable on a program)
	Fixed buffer		5K words × 16	1K word × 16	—		5K words × 16
	Socket communications		• 5K words × 48 (when only P1 is used) • 5K words × 112 (when P1 and P2 are used)	—	5K words × 16		5K words × 48
	Random access buffer		6K words × 1		—		6K words × 1

[1] When using the module at 1Gbps, set "Communication Speed" under "Application Settings" to "Automatic Negotiation". ("1Gbps/Full Duplex" cannot be selected.)

[2] For the maximum segment length (length between hubs), consult the manufacturer of the hub used.

[3] Consult the manufacturer of the switching hub used.

[4] This number applies when a repeater hub is used. For the number of levels that can be constructed when a switching hub is used, consult the manufacturer of the switching hub used.

Point

The operation of commercial devices used for the following applications is not guaranteed. Check the operation before using the module.

- Internet (general public line) (Internet-access service offered by an Internet service provider or a telecommunications carrier)
- Firewall device(s)
- Broadband router(s)
- Wireless LAN

4.2 Function List

The following table lists the functions of Ethernet. For details on the functions, refer to the following.

📖 MELSEC iQ-R Ethernet User's Manual (Application)

○: Available, △: Partially available, ×: Not available

Function	Description	Availability	
		RJ71EN71, RnENCPU (network part)	CPU module (built-in Ethernet port part)
Connection with MELSOFT products and GOTs	Allows the programming and monitoring using the engineering tool, and monitoring and testing a programmable controller from the GOT with Ethernet connection.	○	○
Communications using the SLMP	Allows read/write of the buffer memory/device and other operations from the external device to an SLMP-compatible device connected in the same network as the Ethernet-equipped module. Read/write can also be executed to a CPU module device.	○	○
Communications using the predefined protocol	Allows the communication processing only with the program of the start instruction when the protocol data for communications with the external device is registered using the engineering tool. Allows easy setting of the protocol required for communications with the external device (such as a temperature controller and bar code reader) using the engineering tool's predefined protocol support function.	○	○
Socket communications	Exchanges arbitrary data with an external device connected by Ethernet over TCP/IP or UDP/IP using dedicated instructions.	△[*1]	○
Communications using the fixed buffer	Exchanges arbitrary data between the CPU module and external device using the fixed buffer of the RJ71EN71 and the RnENCPU (network part).	○	×
Communications using the random access buffer	Reads/writes data from multiple external devices to the random access buffer of the RJ71EN71 and the RnENCPU (network part).	○	×
Communications using link dedicated instructions	Reads/writes data from/to other station CPU modules on other networks by Ethernet using link dedicated instructions.	○	×
File transfer function (FTP server)	Reads/writes data per file using the dedicated FTP commands from an external device.	○	○
File transfer function (FTP client)	Enables the CPU module, as an FTP client, to transfer files to an FTP server connected over Ethernet by using the file transfer function instructions.	×	△[*3]
Time setting function (SNTP client)	Automatically sets the time setting of the CPU module by collecting clock data from the time information server (SNTP server) on the LAN.	△[*1]	○
IP filter function	Identifies the IP address of the access source to limit access to the Ethernet-equipped module.	△[*1]	○
Remote password	Prevents illegal access to the CPU module from a remote location.	○	○
IP address change function	Changes the CPU module's IP address without changing the parameter settings.	△[*2]	○
Ethernet diagnostics	Checks the connection status, protocol status, and line status using the engineering tool.	○	○
Automatic detection of connected device	Detects devices supporting iQSS which are connected to the CPU module (built-in Ethernet port part), and automatically displays them on "List of devices" and "Device map area" using the engineering tool. For details, refer to the following. 📖 iQ Sensor Solution Reference Manual	×	○
Communication setting reflection	Reflects the communication settings (such as IP addresses) in devices supporting iQSS in "Device map area" which are connected over Ethernet. For details, refer to the following. 📖 iQ Sensor Solution Reference Manual	×	○
Sensor parameter read/write	Reads/writes parameters from/to iQSS-compatible devices. For details, refer to the following. 📖 iQ Sensor Solution Reference Manual	×	○
Redundant system function[*2]	Allows the network to be configured in a redundant system.	△	△

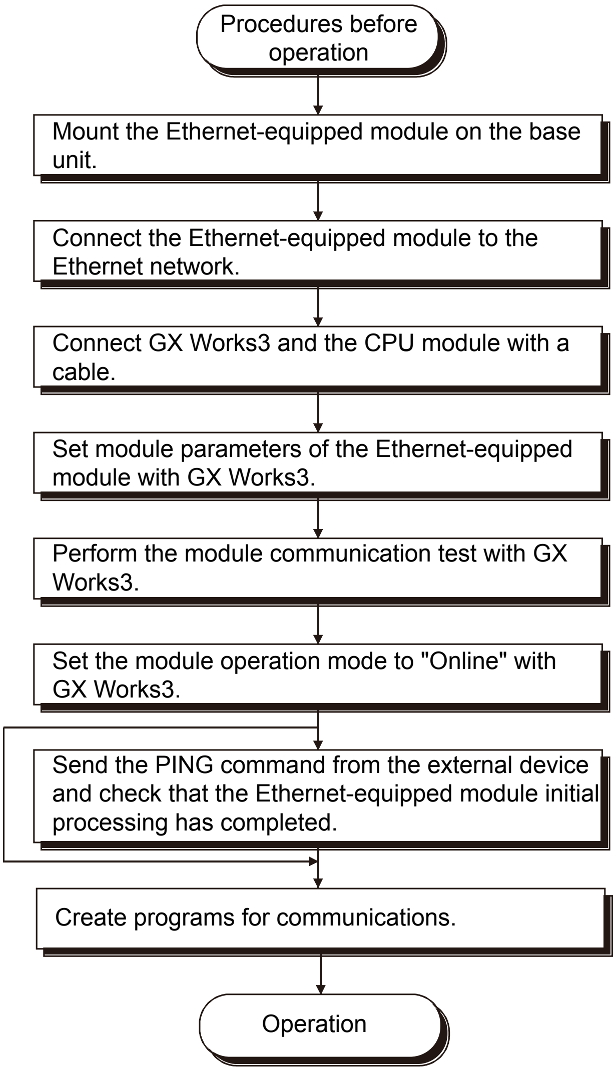
[*1] This function cannot be used when "Q Compatible Ethernet" is selected for the network type.

[*2] Availability depends on the network type and firmware version of the Ethernet-equipped module.
📖 MELSEC iQ-R Ethernet User's Manual (Application)

[*3] Availability depends on the firmware version.
📖 MELSEC iQ-R Ethernet User's Manual (Application)

4.3 Settings and Procedure before Operation

This section describes the settings and operating procedure of the Ethernet-equipped module.



Procedures before operation

Mount the Ethernet-equipped module on the base unit.

Connect the Ethernet-equipped module to the Ethernet network.

Connect GX Works3 and the CPU module with a cable.

Set module parameters of the Ethernet-equipped module with GX Works3.

Perform the module communication test with GX Works3.

Set the module operation mode to "Online" with GX Works3.

Send the PING command from the external device and check that the Ethernet-equipped module initial processing has completed.

Create programs for communications.

Operation

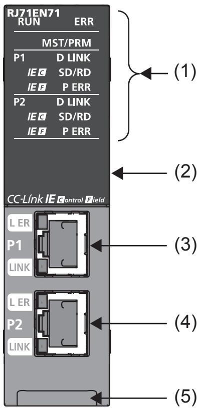
Point

- To operate the Ethernet-equipped module with parameters which are added/changed using GX Works3, write the parameters to the CPU module and reset the CPU module.
- Do not write data to the system areas in the buffer memory of the module.
- Do not use (turn on) any "use prohibited" signal as an input or output signal.
- When the Ethernet-equipped module is replaced, reset the external device.
  (When the external device holds the MAC address of the communication destination, communications may not be continued because the MAC address is changed by the Ethernet-equipped module replacement.)
  For the same reason, restart the Ethernet-equipped module after replacing an external device (such as a personal computer).

4.4 Part Names

4.4.1 RJ71EN71

This section describes the names of each part of the RJ71EN71.



No.	Name	Description
(1)	Operation status indicator LED	Indicates the operating status of the module.
(2)	Dot matrix LED	Indicates the station number set in the module and the module communication test result. For indication of the module communication test result, refer to the following. 📖 MELSEC iQ-R Ethernet User's Manual (Application)
(3)	Ethernet port (P1)	PORT1 connector for network. Connect an Ethernet cable.
	L ER LED	Indicates the port status.
	LINK LED	Indicates the link status.
(4)	Ethernet port (P2)	PORT2 connector for network. Connect an Ethernet cable.
	L ER LED	Same as the P1 connector.
	LINK LED	
(5)	Production information marking	Shows the production information (16 digits) of the module.

Available combinations of networks

The Ethernet port (P1 and P2) of the RJ71EN71 can be used in the following networks depending on the engineering tool setting. (📖 GX Works3 Operating Manual)

- Ethernet
- CC-Link IE Controller Network
- CC-Link IE Field Network
- Q-compatible Ethernet

The following table lists the available combinations of networks and the settings with GX Works3.

Network combination	Settings with GX Works3			Description	Remarks
	Model[*1]	Port 1 network type	Port 2 network type		
Ethernet only	RJ71EN71(E+E)	Ethernet	Ethernet	P1 and P2 can be connected to different Ethernet networks.	—
CC-Link IE Controller Network only	RJ71EN71(CCIEC)	CC-Link IE Control	CC-Link IE Control	P1 and P2 can be connected to CC-Link IE Controller Network.	P1 and P2 cannot be connected to CC-Link IE Controller Network with different network numbers.
CC-Link IE Field Network only	RJ71EN71(CCIEF)	CC-Link IE Field	CC-Link IE Field	P1 and P2 can be connected to CC-Link IE Field Network.	P1 and P2 cannot be connected to CC-Link IE Field Network with different network numbers.
Ethernet + CC-Link IE Controller Network	RJ71EN71(E+CCIEC)	Ethernet	CC-Link IE Control	P1 can be connected to Ethernet and P2 can be connected to CC-Link IE Controller Network.	CC-Link IE Controller Network cannot be configured in ring topology.
Ethernet + CC-Link IE Field Network	RJ71EN71(E+CCIEF)	Ethernet	CC-Link IE Field	P1 can be connected to Ethernet and P2 can be connected to CC-Link IE Field Network.	CC-Link IE Field Network cannot be configured in ring topology.
Q-compatible Ethernet	RJ71EN71(Q)	Q-compatible Ethernet	—	A setting for replacement from the MELSEC-Q series Ethernet interface module. This setting enables the module to be connected to Ethernet without changing the I/O signals and buffer memory from those of the MELSEC-Q series Ethernet interface module.	• Some parameters cannot be set. • P2 cannot be used.

[*1] The name in parentheses is the abbreviation of the network type.

LED indication when Ethernet or Q-compatible Ethernet is used

The LED indication of the RJ71EN71 differs depending on the network used.

The following table lists the LED indications when Ethernet or Q-compatible Ethernet is used.

LED name	Description
RUN LED	Indicates the operating status. On: Normal operation Off: Error (📖 MELSEC iQ-R Ethernet User's Manual (Application))
ERR LED*1	Indicates the error status of the module. On, flashing: Error (📖 MELSEC iQ-R Ethernet User's Manual (Application)) Off: Normal operation
MST/PRM LED*1	Indicates the operating status of CC-Link IE Controller Network when P2 is used in CC-Link IE Controller Network. (The MST/PRM LED is always off when CC-Link IE Controller Network is not used.)
	Indicates the operating status of CC-Link IE Field Network when P2 is used in CC-Link IE Field Network. (The MST/PRM LED is always off when CC-Link IE Field Network is not used.)
D LINK LED*1	Indicates the data link status of P2 when P2 is used in CC-Link IE Controller Network. (The D LINK LED of P1 is always off.)
	Indicates the data link status of P2 when P2 is used in CC-Link IE Field Network. (The D LINK LED of P1 is always off.)
SD/RD LED	Indicates the data communication status. On: Data being sent or received Off: Data not sent nor received
P ERR LED*1	Indicates the error status of P1 and P2. On, flashing: Error (📖 MELSEC iQ-R Ethernet User's Manual (Application)) Off: Normal operation
IE C LED*1*2	Indicates the network type setting status when P2 is used in CC-Link IE Controller Network. (The IE C LED of P1 is always off.)
IE F LED*1	Indicates the network type setting status when P2 is used in CC-Link IE Field Network. (The IE F LED of P1 is always off.)
L ER LED	Indicates the port status when P2 is used in CC-Link IE Controller Network. (The L ER LED of P1 is always off.)
	Indicates the port status when P2 is used in CC-Link IE Field Network. (The L ER LED of P1 is always off.)
LINK LED	Indicates the link status. On (green): Link-up (1Gbps) On (yellow): Link-up (100Mbps) Off: Link-down, link-up (10Mbps)

*1 The LED is always off in offline mode.

*2 LED indications differ depending on the version of the RJ71EN71.
(📖 MELSEC iQ-R CC-Link IE Controller Network User's Manual (Application))

Dot matrix LED indication

The following table describes the station number indicated on the dot matrix LED.

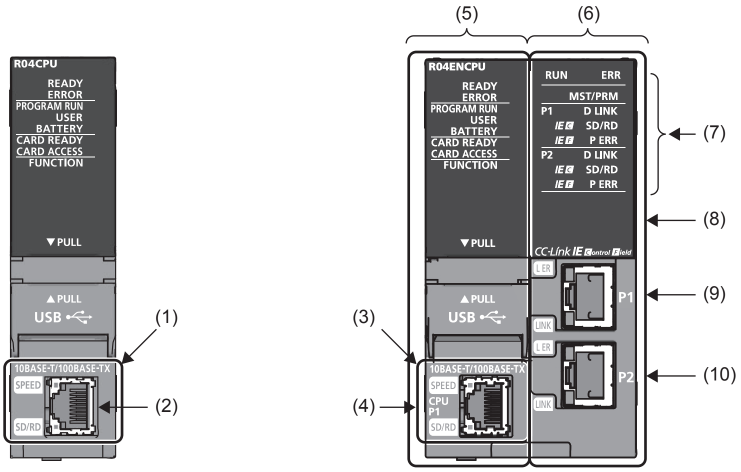
Network combination	Settings with GX Works3			Indication
	Model[*1]	Port 1 network type	Port 2 network type	
Ethernet only	RJ71EN71(E+E)	Ethernet	Ethernet	Always off
CC-Link IE Controller Network only	RJ71EN71(CCIEC)	CC-Link IE Control	CC-Link IE Control	Indicates the current station number of CC-Link IE Controller Network. Station number not set: "— — —" Control station, normal station: 1 to 120
CC-Link IE Field Network only	RJ71EN71(CCIEF)	CC-Link IE Field	CC-Link IE Field	Indicates the current station number of CC-Link IE Field Network. Station number not set: "— — —" Master station: 0 Submaster station, local station: 1 to 120
Ethernet + CC-Link IE Controller Network	RJ71EN71(E+CCIEC)	Ethernet	CC-Link IE Control	Indicates the current station number of CC-Link IE Controller Network. Station number not set: "— — —" Control station, normal station: 1 to 120
Ethernet + CC-Link IE Field Network	RJ71EN71(E+CCIEF)	Ethernet	CC-Link IE Field	Indicates the current station number of CC-Link IE Field Network. Station number not set: "— — —" Master station: 0 Submaster station, local station: 1 to 120
Q-compatible Ethernet	RJ71EN71(Q)	Q-compatible Ethernet	—	Always off
In offline mode				Indicates "···" in offline mode.
At major error				Undefined

*1 The name in parentheses is the abbreviation of the network type.

4.4.2 CPU module

This section describes the part names of the CPU module related to the Ethernet function. For the names of the other parts, refer to the following.

🕮 MELSEC iQ-R CPU Module User's Manual (Startup)



No.	Name	Description
(1)	Built-in Ethernet port part	A part to connect the CPU module to Ethernet
(2)	Ethernet port	A connector to connect the CPU module to the 10BASE-T/100BASE-TX. (RJ45 connector) The CPU module determines whether to use the 10BASE-T or 100BASE-TX according to the hub.
	SPEED LED	Indicates the link status. On: Link-up (100Mbps) Off: Link-down or link-up (10Mbps)
	SD/RD LED	Indicates the data communication status. On: Data being sent or received Off: Data not sent nor received
(3)	Built-in Ethernet port part	A part to connect the CPU module to Ethernet
(4)	Ethernet port (CPU P1)	A connector to connect the CPU module to the 10BASE-T/100BASE-TX. (RJ45 connector) The CPU module determines whether to use the 10BASE-T or 100BASE-TX according to the hub.
	SPEED LED	Indicates the link status. On: Link-up (100Mbps) Off: Link-down or link-up (10Mbps)
	SD/RD LED	Indicates the data communication status. On: Data being sent or received Off: Data not sent nor received
(5)	CPU part	A part that works as the CPU module
(6)	Network part	A part that has the functions of Ethernet, CC-Link IE Controller Network, and CC-Link IE Field Network
(7)	Operation status indicator LED	Indicates the operating status of the module.
(8)	Dot matrix LED	Indicates the station number set in the module and the module communication test result. For indication of the module communication test result, refer to the following. 🕮 MELSEC iQ-R Ethernet User's Manual (Application)
(9)	Ethernet port (P1)	PORT1 connector for network. Connect an Ethernet cable.
	L ER LED	Indicates the port status.
	LINK LED	Indicates the link status.
(10)	Ethernet port (P2)	PORT2 connector for network. Connect an Ethernet cable.
	L ER LED	Same as the P1 connector.
	LINK LED	

Available combinations of networks

The Ethernet port (P1 and P2) of the RnENCPU can be used in the following networks depending on the engineering tool setting. (📖 GX Works3 Operating Manual)

- Ethernet
- CC-Link IE Controller Network
- CC-Link IE Field Network

The following table lists the available combinations of networks and the settings with GX Works3.

Network combination	Settings with GX Works3			Description	Remarks
	Model[*1]	Port 1 network type	Port 2 network type		
CC-Link IE Controller Network only	_RJ71EN71(CCIEC)	CC-Link IE Control	CC-Link IE Control	P1 and P2 can be connected to CC-Link IE Controller Network.	P1 and P2 cannot be connected to CC-Link IE Controller Network with different network numbers.
CC-Link IE Field Network only	_RJ71EN71(CCIEF)	CC-Link IE Field	CC-Link IE Field	P1 and P2 can be connected to CC-Link IE Field Network.	P1 and P2 cannot be connected to CC-Link IE Field Network with different network numbers.
Ethernet + CC-Link IE Controller Network	_RJ71EN71(E+IEC)	Ethernet	CC-Link IE Control	P1 can be connected to Ethernet and P2 can be connected to CC-Link IE Controller Network.	CC-Link IE Controller Network cannot be configured in ring topology.
Ethernet + CC-Link IE Field Network	_RJ71EN71(E+IEF)	Ethernet	CC-Link IE Field	P1 can be connected to Ethernet and P2 can be connected to CC-Link IE Field Network.	CC-Link IE Field Network cannot be configured in ring topology.

[*1] The name in parentheses is the abbreviation of the network type.

■Precautions

The following items cannot be set for the RnENCPU.

- Ethernet only (Port 1 network type: Ethernet, Port 2 network type: Ethernet)
- Q-compatible Ethernet

Network used and LED indication

The LED indication of the RnENCPU differs depending on the network used.

For the LED indication when Ethernet is used, refer to Page 4 - 6 LED indication when Ethernet or Q-compatible Ethernet is used.

Dot matrix LED indication

The following table describes the station number indicated on the dot matrix LED.

Network combination	Settings with GX Works3			Indication
	Model[*1]	Port 1 network type	Port 2 network type	
CC-Link IE Controller Network only	_RJ71EN71(CCIEC)	CC-Link IE Control	CC-Link IE Control	Indicates the current station number of CC-Link IE Controller Network. Station number not set: "— — —" Control station, normal station: 1 to 120
CC-Link IE Field Network only	_RJ71EN71(CCIEF)	CC-Link IE Field	CC-Link IE Field	Indicates the current station number of CC-Link IE Field Network. Station number not set: "— — —" Master station: 0 Submaster station, local station: 1 to 120
Ethernet + CC-Link IE Controller Network	_RJ71EN71(E+IEC)	Ethernet	CC-Link IE Control	Indicates the current station number of CC-Link IE Controller Network. Station number not set: "— — —" Control station, normal station: 1 to 120
Ethernet + CC-Link IE Field Network	_RJ71EN71(E+IEF)	Ethernet	CC-Link IE Field	Indicates the current station number of CC-Link IE Field Network. Station number not set: "— — —" Master station: 0 Submaster station, local station: 1 to 120
In offline mode				Indicates "···" in offline mode.
At major error				Undefined

[*1] The name in parentheses is the abbreviation of the network type.

4.5 Ethernet Wiring

This section describes the wiring when Ethernet is used.

Wiring methods

The following describes connection and disconnection of an Ethernet cable.

■Connecting a cable

1. Push the Ethernet cable connector into the Ethernet-equipped module until it clicks. Pay attention to the connector's direction.
2. Lightly pull it to check that it is securely connected.
3. Check whether the LINK LED of the port connected with an Ethernet cable is on.[*1]

[*1] The time between the cable connection and the LINK LED turning on may vary. The LINK LED usually turns on in a few seconds. Note, however, that the time may be extended further if the link-up processing is repeated depending on the status of the device on the line. Check that the cable is connected properly if the LINK LED does not turn on.

■Disconnecting a cable

1. Unplug the Ethernet cable while pressing the down latch.

■Precautions

- Place the Ethernet cable in a duct or clamp it. If not, dangling cable may swing or inadvertently be pulled, resulting in damage to the module or cables or malfunction due to poor contact.
- Do not touch the core of the cable-side or module-side connector, and protect it from dirt or dust. If oil from your hand, dirt, or dust is attached to the core, it can increase transmission loss, resulting in failure of data link.
- Check that the Ethernet cable is not disconnected or not shorted and there is no problem with the connector connection.
- Do not use Ethernet cables with broken latches. Doing so may cause the cable to unplug or malfunction.
- Hold the connector part when connecting and disconnecting the Ethernet cable. Pulling the cable connected to the module may result in malfunction or damage to the module or cable.
- For connectors without Ethernet cable, attached connector cover should be placed to prevent foreign matter such as dirt and dust.
- The maximum segment length of the Ethernet cable is 100m. However, the length may be shorter depending on the operating environment of the cable. For details, contact the manufacturer of the cable used.
- The bend radius of the Ethernet cable is limited. For details, check the specifications of the Ethernet cable used.

Wiring products

The following describes the devices used for Ethernet.

Point

An Ethernet-equipped module determines whether to use 1000BASE-T/100BASE-TX/10BASE-T and the fullduplex/half-duplex communication mode based on the hub. For connection to the hub with no automatic negotiation function, set the communication mode on the hub side to meet the mode of the Ethernet-equipped module.

■Ethernet cables

Use Ethernet cables that meet the following standards.

Communication speed	Ethernet cable	Connector	Type
1Gbps[*1]	Category 5e or higher, straight cable (shielded, STP)	RJ45 connector	1000BASE-T
	Category 5e or higher, crossing cable (shielded, STP)		
100Mbps	Category 5 or higher, straight cable (shielded, STP)		100BASE-TX
	Category 5 or higher, crossing cable (shielded, STP)		
10Mbps	Category 3 or higher, straight cable (shielded, STP)		10BASE-T
	Category 3 or higher, straight cable (UTP)		
	Category 3 or higher, crossing cable (shielded, STP)		
	Category 3 or higher, crossing cable (UTP)		

[*1] The CPU module (built-in Ethernet port part) does not support the communication speed.

Point

A communication error may occur due to high-frequency noise from devices other than a programmable controller in a given connection environment. The following describes countermeasures to be taken on the Ethernet-equipped module side to avoid high-frequency noise influence.

Wiring

- Do not bundle the cable with the main circuit or power cable or do not place it near those lines.
- Place the cable in a duct.
- When using a UTP cable, use STP cables.

Communication method

- Use TCP/IP for data communications with external devices.
- Increase the number of retries of communications if needed.

Transmission speed

- Change the communication speed to be slower than the speed of communications connected with an Ethernet cable in "Communication Speed" of "Application Settings". (📖 MELSEC iQ-R Ethernet User's Manual (Application))

■Hubs

The hubs used in Ethernet must support the transmission speed of communications.

4.6 Settings with GX Works3

To use the Ethernet-equipped module, set parameters with GX Works3 in advance.

For the parameter settings used for the exercises in this textbook, refer to the parameter settings described in each exercise (Chapter 5 to 8).

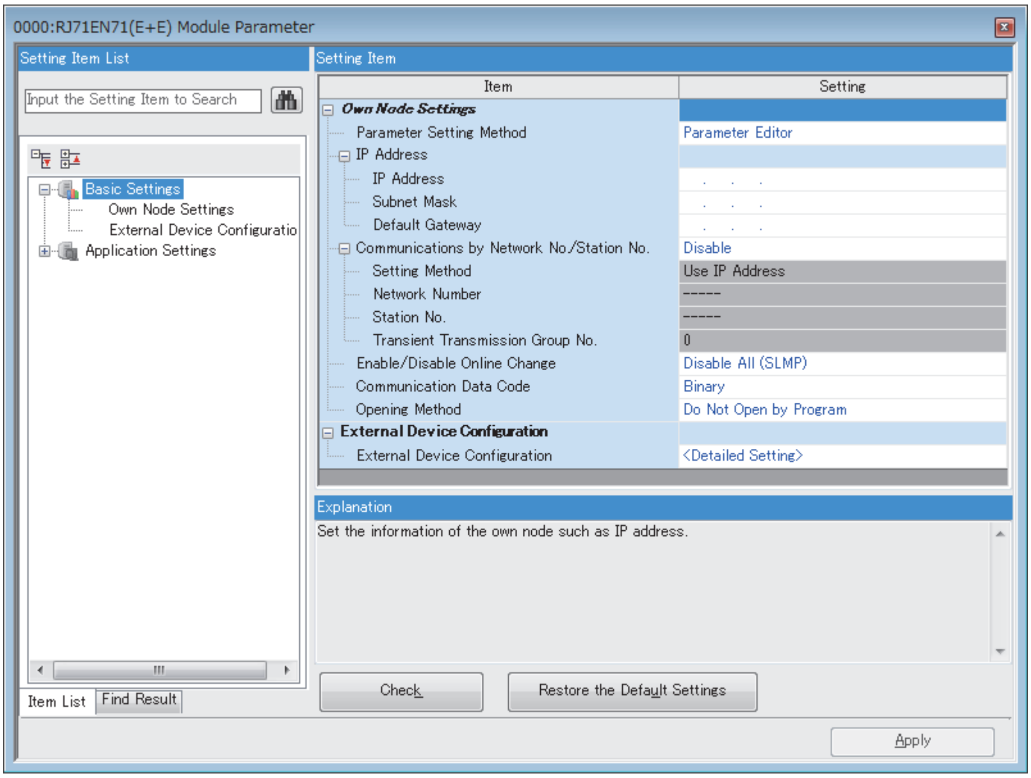
For details on each window, refer to the following.

📖 GX Works3 Operating Manual

📖 MELSEC iQ-R Ethernet User's Manual (Application)

4.6.1 Basic Settings

Set values for the own node settings, external device configuration, and other settings of the Ethernet-equipped module.



0000:RJ71EN71(E+E) Module Parameter

Item	Setting
Own Node Settings	
Parameter Setting Method	Parameter Editor
IP Address	
IP Address	. . .
Subnet Mask	. . .
Default Gateway	. . .
Communications by Network No./Station No.	Disable
Setting Method	Use IP Address
Network Number	-----
Station No.	-----
Transient Transmission Group No.	0
Enable/Disable Online Change	Disable All (SLMP)
Communication Data Code	Binary
Opening Method	Do Not Open by Program
External Device Configuration	
External Device Configuration	<Detailed Setting>

Explanation

Set the information of the own node such as IP address.

Setting items	Description
Own Node Settings	Set the IP address and communication data code of the Ethernet-equipped module.
External Device Configuration	Set the method and protocol used to communicate with external devices.

Own Node Settings

Set the IP address and communication data code of the Ethernet-equipped module.

Item		Description	Setting range
Parameter Setting Method		Set the own station settings in parameters.	Parameter Editor (fixed)
IP Address	IP Address	Set the IP address of the own station. Ensure that the Ethernet-equipped module on the own station and the external device to be communicated with have the same class and subnet address. Consult with the network manager for the IP address setting. When using the IP address in a redundant system, use it as the system A IP address.	• Blank • 0.0.0.1 to 223.255.255.254 (Default: Blank)
	Subnet Mask	Set the subnet mask pattern of the default gateway when setting the IP address of the default gateway and communicating with an external device on another network via a router. All the devices on a subnetwork must have the same subnet mask. This setting is not required when communicating in a single network.	• Blank • 0.0.0.1 to 255.255.255.255 (Default: Blank)
	Default Gateway	Set the IP address of the default gateway (the device which the own node passes through to access a device of another network). Set a value that satisfies the following conditions. • The IP address class is any of A, B, and C. • The subnet address of the default gateway is the same as that of the Ethernet-equipped module on the own station. • The host address bits are not all "0" or all "1".	• Blank • 0.0.0.1 to 223.255.255.254 (Default: Blank)
Communications by Network No./Station No.[*1]	—	Select "Enable" to set the network number, station number, and transient transmission group number. This setting is not required when the following functions are not used. • Connection with MELSOFT products and GOTs (when connecting by specifying a network number and a station number) • Communications using the SLMP (when specifying a target station with its network number and station number) • Communications using link dedicated instructions • Communications with different networks	• Disable • Enable (Default: Disable)
	Setting Method	Select the method for setting the network number and station number.	• Use IP Address • Not Use IP Address (Default: Use IP Address)
	Network Number	Enter the network number of the own station when selecting "Not Use IP Address" in "Setting Method".	1 to 239 (Default: 1)
	Station No.	Enter the station number of the own station when selecting "Not Use IP Address" in "Setting Method".	1 to 120 (Default: 1)
	Transient Transmission Group No.	Set the transient transmission group number of the own station.	0 to 32 (Default: 0)
Enable/Disable Online Change		Select whether to enable external devices to write data in communications using the SLMP while the CPU module is in RUN state.	• Disable All (SLMP) • Enable All (SLMP) (Default: Disable All (SLMP))
Communication Data Code		Select the communication data code used for communications.	• Binary • ASCII (Default: Binary)
Opening Method		Select how to open a connection when using UDP/IP communications or Passive open of TCP/IP communications. • When "Do Not Open by Program" is selected, a connection is open when the system receives the Active request. The program for open/close processing is not required. • When "Open by Program" is selected, the open/close processing is performed by a program. The module cannot communicate when the CPU module is in STOP state.	• Do Not Open by Program • Open by Program (Default: Do Not Open by Program)

*1 This setting is not available for the CPU module (built-in Ethernet port part).

■IP Address

When the parameters are written without setting the IP addresses, the following addresses are set.

Module		IP address
RJ71EN71	P1 connector	192.168.3.40
	P2 connector	192.168.4.40
RnENCPU (network part)	P1 connector	192.168.3.40
CPU module (built-in Ethernet port part)		192.168.3.39

■Setting Method

When "Use IP Address" is selected, the network number and station number will be set from the third octet and fourth octet of the IP address.

For example, when the IP address is set to 192.168.1.10, the network number is set to 1 and the station number is set to 10.

When an IP address is used, a value out of the range of the network number and station number cannot be set in the third octet and fourth octet.

When "Not Use IP Address" is selected, set the network number and station number.

Point

When "Port 1 Network Type" and "Port 2 Network Type" are set to "Ethernet" for the RJ71EN71 and the same network number is set for both of P1 and P2, P1 is always used for relay to other networks.

■Enable/Disable Online Change

Select whether to enable an SLMP data write command which is received by the Ethernet-equipped module while the CPU module is in RUN state.

This setting is effective for not only the CPU module on the own station but also the ones on the other stations.

When "Disable All (SLMP)" is selected, data writing using SLMP is disabled while the CPU module is in RUN state.

When "Enable All (SLMP)" is selected, data writing using SLMP is enabled even when the CPU module is in RUN state.

To write data to the FTP server when the CPU module is in RUN state, set "Allow Online Change" under "FTP Server Settings" in "Application Settings".

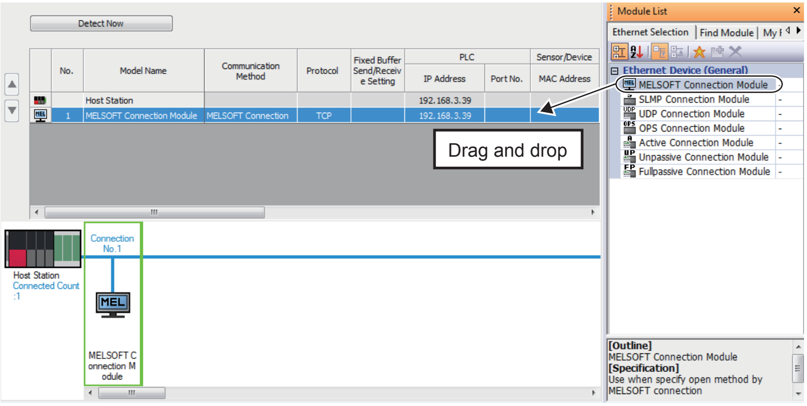
4.6.2 External Device Configuration

Set the method and protocol used to communicate with external devices.

Setting procedure

The procedure for setting the external device to be connected is shown below.

1. Select the external device to be connected in "Module List" and drag and drop it to "List of devices" or "Device map area".



	No.	Model Name	Communication Method	Protocol	Fixed Buffer Send/Receive Setting	PLC		Sensor/Device
						IP Address	Port No.	MAC Address
		Host Station				192.168.3.39		
	1	MELSOFT Connection Module	MELSOFT Connection	TCP		192.168.3.39		

Drag and drop

[Outline]
MELSOFT Connection Module
[Specification]
Use when specify open method by MELSOFT connection

2. Set the required items. (The required items vary depending on the selected external device.)

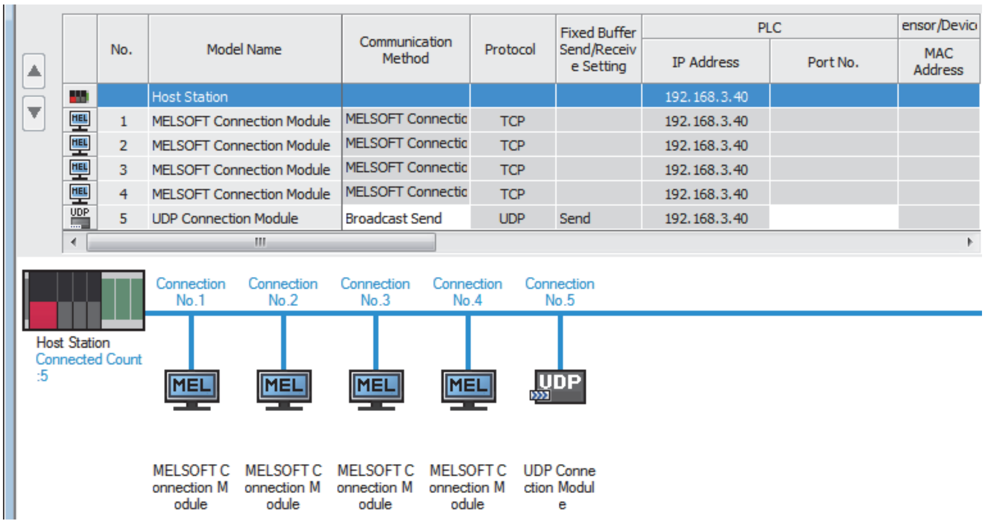
3. Select [Close with Reflecting the Setting] to end the external device configuration settings.

Restriction

External devices must be set from the connection No.1.

To use a specific connection number, set "MELSOFT Connection Module" in the connection number not used.

- When only the connection No.5 is used.



	No.	Model Name	Communication Method	Protocol	Fixed Buffer Send/Receive Setting	PLC		ensor/Devic
						IP Address	Port No.	MAC Address
		Host Station				192.168.3.40		
	1	MELSOFT Connection Module	MELSOFT Connectic	TCP		192.168.3.40		
	2	MELSOFT Connection Module	MELSOFT Connectic	TCP		192.168.3.40		
	3	MELSOFT Connection Module	MELSOFT Connectic	TCP		192.168.3.40		
	4	MELSOFT Connection Module	MELSOFT Connectic	TCP		192.168.3.40		
	5	UDP Connection Module	Broadcast Send	UDP	Send	192.168.3.40		

Setting items

Item		Description	Setting range
[Detect Now] button		Executes the automatic detection of connected devices. The [Detect Now] button is displayed only when the CPU module (built-in Ethernet port part) is used. For details, refer to the following. 📖 iQ Sensor Solution Reference Manual	—
No.		Connection number for distinguishing settings for each user connection	The number is set in the following range starting with 1. • RJ71EN71 and RnENCPU (network part): 1 to 64[*1] • CPU module (built-in Ethernet port part): 1 to 16
Model Name		The name of the external device to be connected is displayed.	—
Communication Method		Set the method for communications with the external device.	• Broadcast Send • Broadcast Receive • Fixed Buffer (Procedure Exist)[*2] • Fixed Buffer (No Procedure)[*2] • Random Access Buffer[*2] • Predefined Protocol • Socket Communication[*3] • MELSOFT Connection • SLMP • OPS Connection
Protocol		Select the predefined protocol for the external device.	• TCP[*4] • UDP[*4]
Fixed Buffer Send/Receive Setting[*2]		For communications using the fixed buffer, select whether to use the buffer for sending or for receiving in a connection to the external device.	• Send • Receive • Pairing (Receive) • Pairing (Send)
PLC	IP Address	The IP address of the own node, which is set in "IP Address" under "Own Node Settings" of "Basic Settings", is displayed. Or the IP address of the communication destination set in the device supporting iQSS is displayed when the automatic detection of connected devices is used.	—
	Port No.	Set the port number for each connection of the Ethernet-equipped module.	1 to 4999, 5010 to 65534 (Default: Blank)[*4]
Sensor/Device	MAC Address	The MAC address of the device supporting iQSS is displayed when the automatic detection of connected devices is used.	—
	Host Name	Set the name for identifying the device supporting iQSS. This setting is available only for the devices supporting iQSS which were detected by automatic detection of connected devices.	63 characters maximum The following one-byte characters can be used. • Number (0 to 9) • Alphabetical character (a to z, A to Z) • Hyphen (-) • Period (.) • Colon (:) • Underscore (_) (Default: Blank)[*4]
	IP Address	Set the IP address of the external device.	0.0 0.1 to 223.255.255.254, 255.255.255.255[*5] (Default: Blank)[*4]
	Port No.	Set the port number of the external device. Set 65535 to set all the port numbers as the target of data receive.	1 to 65534, 65535 (Default: Blank)[*4]
	Subnet Mask	Set the subnet mask of the device supporting iQSS. This setting is available only for the devices supporting iQSS which were detected by automatic detection of connected devices.	192.0.0.0 to 255.255.255.252 (Default: Blank)[*4]
	Default Gateway	Set the default gateway of the device supporting iQSS. This setting is available only for the devices supporting iQSS which were detected by automatic detection of connected devices.	0.0.0.1 to 223.255.255.254 (Default: Blank)[*4]
Existence Confirmation		Select the method of alive check which is performed when the Ethernet-equipped module has not communicated with the external device for a certain period of time. When the module cannot communicate with the external device, the connection will be closed.	• KeepAlive • UDP • Do not confirm existence

[1] When "Q Compatible Ethernet" is selected in the network type, the setting range is 1 to 16.
[2] This setting is not available for the CPU module (built-in Ethernet port part).
[3] This setting is not available when the RJ71EN71 network type is set to "Q Compatible Ethernet".
[4] When the automatic detection of connected devices is executed, the values read from the connected devices will be the default.
[5] When "Communication Method" is set to "OPS Connection", 255.255.255.255 can be set.

Point

Comments can be set on the "Properties" window displayed by right-clicking the module in "List of devices" or "Device map area" and selecting "Properties". The following settings are available depending on the selected device.

- Changing the image
- Creating association with a file or application

■Existence Confirmation

When the Ethernet-equipped module has not communicated with the external device for a certain period of time while the connection is open, this function checks whether the external device is alive by sending an alive check message to the device and waiting for the response.

The following table lists the details on alive check.

Item	Applicable protocol	Description
KeepAlive	TCP/IP	This method is used for a connection opened using TCP/IP. The Ethernet-equipped module performs an alive check by sending an alive check ACK message to the external device with which communications have not been performed for a certain period of time and waiting to see whether the response is received. The connection will be automatically closed when the open state is not continued.[1]
UDP	UDP/IP	This method is used for a connection opened using UDP/IP. The Ethernet-equipped module performs an alive check by sending the PING command (ICMP echo request/response function) to the external device with which communications have not been performed for a certain period of time and waiting to see whether the response is received.[2]
Do not confirm existence	TCP/IP, UDP/IP	Alive check is not performed.

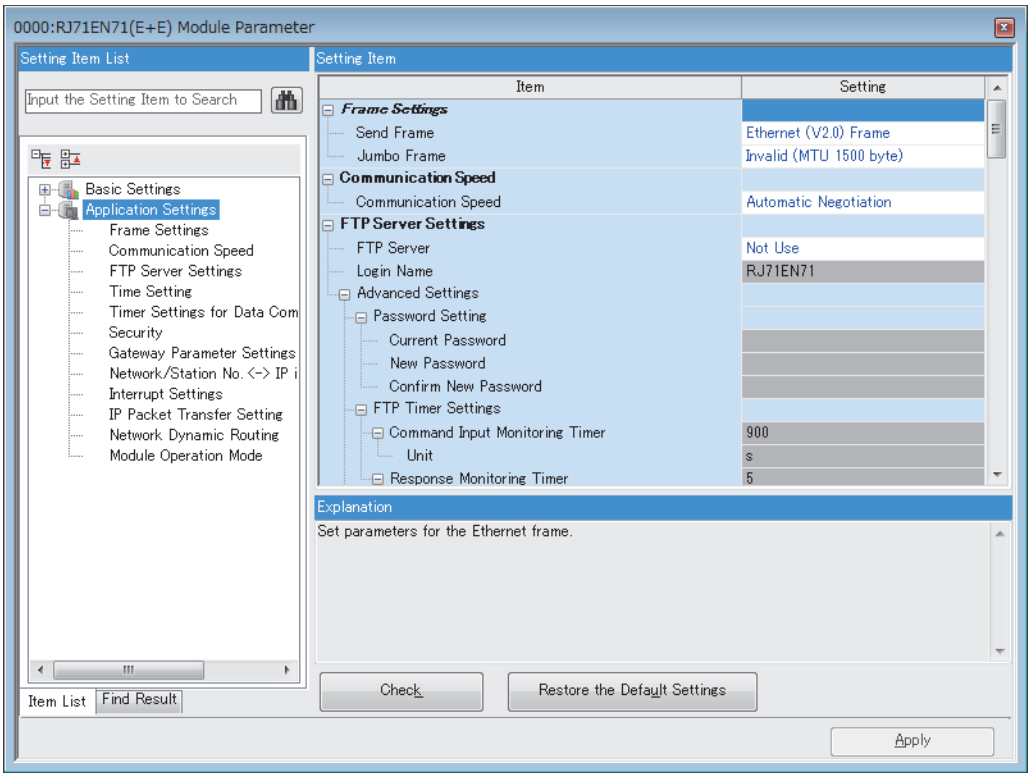
[1] The connection may be disconnected if the external device does not support the TCP KeepAlive function (response to a KeepAlive ACK message).
[2] The Ethernet-equipped module automatically sends an echo response packet when it receives a PING echo request command. (It sends a response to the received PING command even if the connection used in the data communications with the external device is closed.)

If a response message cannot be received from the external device (or if an error has been detected) using the alive check function, the following are performed.

- The corresponding connection will be forcibly closed. (The line is disconnected.) Open the connection again using a user program.
- The open completion signal turns off, and the error code is stored in the buffer memory areas.

4.6.3 Application Settings

Set values for the frame setting, the communication speed setting, and other settings of the Ethernet-equipped module.



Item	Setting
Frame Settings	
Send Frame	Ethernet (V2.0) Frame
Jumbo Frame	Invalid (MTU 1500 byte)
Communication Speed	
Communication Speed	Automatic Negotiation
FTP Server Settings	
FTP Server	Not Use
Login Name	RJ71EN71
Advanced Settings	
Password Setting	
Current Password	
New Password	
Confirm New Password	
FTP Timer Settings	
Command Input Monitoring Timer	900
Unit	s
Response Monitoring Timer	5

Set parameters for the Ethernet frame.

Setting items	Description
Frame Settings*1	Set parameters for the Ethernet frame.
Communication Speed*1	Set the communication speed between the module and the external device.
FTP Server Settings	Set the file transfer function (FTP server).
FTP Client Settings*3	Set the file transfer function (FTP client).
DNS Settings*3	Set the DNS setting.
Time Setting*2	Set the time setting function (SNTP client).
Timer Settings for Data Communication	Set the timer used for the following communications. • Connection with MELSOFT products and GOTs • Communications using the SLMP • Communications using the predefined protocol • Socket communications/Communications using the fixed buffer • Communications using the random access buffer • File transfer function (FTP server) • File transfer function (FTP client)
Security*2	Set the security measures for access to the Ethernet-equipped module.
Gateway Parameter Settings	Set this item to communicate with an external device on Ethernet via a router and gateway.
Network/Station No. <-> IP information setting*1	Set this item to communicate with another network module by a network number and station number.
Interrupt Settings*1	Set this item to start up an interrupt program.
IP Packet Transfer Setting*2	Set the IP packet transfer function. For details on the IP packet transfer function, refer to the following. • 🕮 MELSEC iQ-R CC-Link IE Controller Network User's Manual (Application) • 🕮 MELSEC iQ-R CC-Link IE Field Network User's Manual (Application)
Network Dynamic Routing*1*2	Set the dynamic routing.
Module Operation Mode*1	Set the module operation mode.
Redundant System Settings	Set the redundant system function. Only functions that meet usage conditions with a redundant system can be set.

*1 This setting is not available for the CPU module (built-in Ethernet port part).

*2 This setting is not available when the RJ71EN71 network type is set to "Q Compatible Ethernet".

*3 This setting is available only for the CPU module (built-in Ethernet port part).

Gateway Parameter Settings

With gateway parameter settings, the Ethernet-equipped module can communicate with external devices on other Ethernet networks via a router and gateway. One default router and up to eight routers can be set.

Item		Description	Setting range
Gateway Other Than Default Gateway		Set this item to communicate with an external device on another Ethernet network via a router.	• Use • Not Use (Default: Not Use)
Gateway Information	No.1 to No.8	Set the information of the gateway other than the default gateway.	—

Point

Set the default gateway when communicating via the default gateway.

■Gateway IP Address

When communicating with an external device on another Ethernet network through a gateway other than the default gateway, set the IP address of the gateway. (Setting range: 0.0.0.1 to 223.255.255.254)

Set a value that satisfies the following conditions.

- The IP address class is any of A, B, and C.
- The subnet address of the gateway is the same as that of the Ethernet-equipped module on the own station.
- The host address bits are not all "0" or all "1".

Point

- When the Ethernet-equipped module communicates with an external device on another Ethernet network by Passive open, communications can be performed without gateway parameter settings.
- In a system where the Proxy router is used, the gateway parameter settings are not required.

■Subnet Address

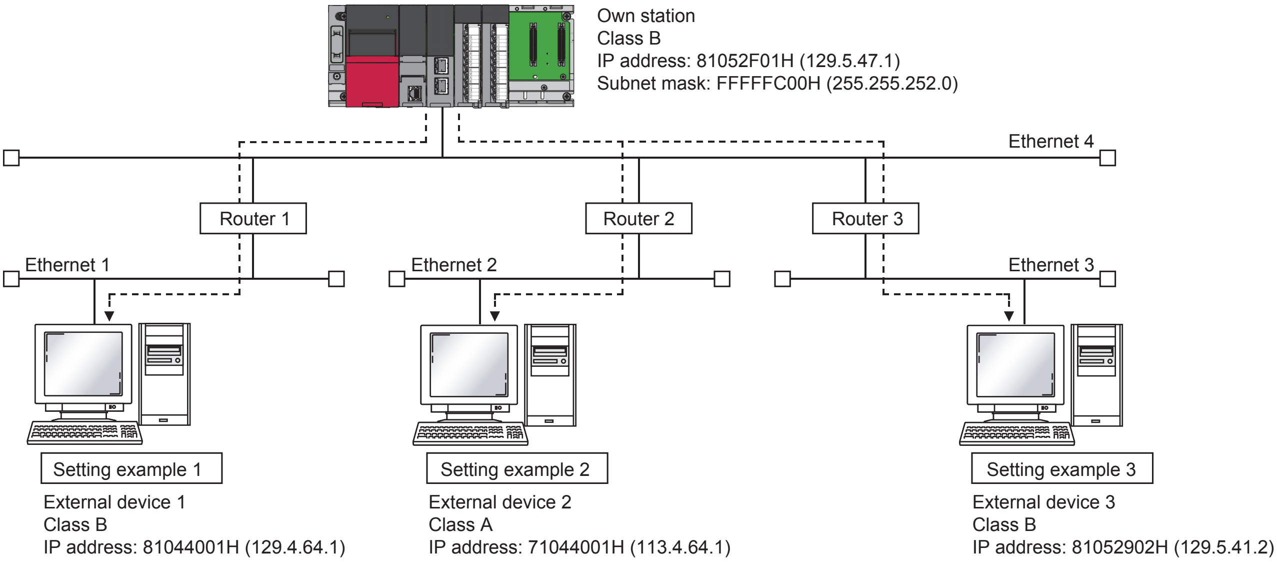
When communicating with an external device on another Ethernet network through a gateway other than the default gateway, set the network address[*1] or subnet address[*2] of the external device. (Setting range: 0.0.0.1 to 255.255.255.254)

Set a value that satisfies the following conditions.

- The IP address class is any of A, B, and C.
- The host address bits are all "0".

[*1] Set the network address of the external device when its class (network address) is different from that of the Ethernet-equipped module on the own station.

[*2] Set the subnet address of the external device when its class (network address) is the same as that of the Ethernet-equipped module on the own station.



Ex.

When the network addresses differ between the Ethernet-equipped module on the own station and the external device

Network address

Own station IP address (Class B)	81H 1 0 0 0 0 0 0 1	05H 0 0 0 0 0 1 0 1	2FH 0 0 1 0 1 1 1 1	01H 0 0 0 0 0 0 0 1
External device 1 IP address (Class B)	81H 1 0 0 0 0 0 0 1	04H 0 0 0 0 0 1 0 0	40H 0 1 0 0 0 0 0 0	01H 0 0 0 0 0 0 0 1

⇩ Set the network address of external device 1.

Network address

Subnet address setting value	81H 1 0 0 0 0 0 0 1	04H 0 0 0 0 0 1 0 0	00H 0 0 0 0 0 0 0 0	00H 0 0 0 0 0 0 0 0

Ex.

When the classes differ between the Ethernet-equipped module on the own station and the external device

Network address

Own station IP address (Class B)	81H 1 0 0 0 0 0 0 1	05H 0 0 0 0 0 1 0 1	2FH 0 0 1 0 1 1 1 1	01H 0 0 0 0 0 0 0 1
External device 2 IP address (Class A)	71H 0 1 1 1 0 0 0 1	04H 0 0 0 0 0 1 0 0	40H 0 1 0 0 0 0 0 0	01H 0 0 0 0 0 0 0 1

⇩ Set the network address of external device 2.

Network address

Subnet address setting value	71H 0 1 1 1 0 0 0 1	00H 0 0 0 0 0 0 0 0	00H 0 0 0 0 0 0 0 0	00H 0 0 0 0 0 0 0 0

Ex.

When the network address of the Ethernet-equipped module on the own station is the same as that of the external device

Network address

Own station IP address (Class B)	81H 1 0 0 0 0 0 0 1	05H 0 0 0 0 0 1 0 1	2FH 0 0 1 0 1 1 1 1	01H 0 0 0 0 0 0 0 1
Subnet mask	FFH 1 1 1 1 1 1 1 1	FFH 1 1 1 1 1 1 1 1	FCH 1 1 1 1 1 1 0 0	00H 0 0 0 0 0 0 0 0
External device 3 IP address (Class B)	81H 1 0 0 0 0 0 0 1	05H 0 0 0 0 0 1 0 1	29H 0 0 1 0 1 0 0 1	02H 0 0 0 0 0 0 1 0

⇩ Set the subnet address of external device 3.

Subnet address

Subnet address setting value	81H 1 0 0 0 0 0 0 1	05H 0 0 0 0 0 1 0 1	28H 0 0 1 0 1 0 0 0	00H 0 0 0 0 0 0 0 0

4.7 Module Communication Test

The module communication test checks the hardware of the RJ71EN71 or the RnENCPU (network part). When the communications using the RJ71EN71 or the RnENCPU (network part) are unstable, whether a hardware failure occurs or not can be checked.

The following table lists the tests performed.

Test item	Description
Internal self-loopback test	Checks whether the communication function of the module can be executed normally.
External self-loopback test	Checks whether the communications can be performed normally with the cable connected between two connectors.

Creation of module configurations is described on Page 5 - 8 Setting parameters with GX Works3 or later. Therefore, detailed explanations are omitted here.

■Procedure

1. Select program elements (objects) from the "Element Selection" window and arrange them in the "Module Configuration" window.

🖱 Double-click [Module Configuration] in the "Navigation" window.

2. Set the module operation mode for P1 and P2 each to module communication test mode in the following item.

🖱 [Navigation] window ⇨ [Parameter] ⇨ [Module Information] ⇨ [RJ71EN71] ⇨ [Module Parameter] ⇨ [Application Settings] ⇨ [Module Operation Mode]

3. Connect the P1 and P2 of the RJ71EN71 or the RnENCPU (network part) with an Ethernet cable.

4. Write the module parameters to the CPU module.

5. Reset or power off and on the CPU module to start the module communication test.

Point

Do not perform a module communication test while the module is connected to another station. The operation of another station may fail.

■Checking the status and result of the module communication test

The test status and result can be checked with the dot matrix LED of the module.

Test status	LED indication
Test in progress	The dot matrix LED displays "UCT".
Normal completion	The dot matrix LED displays "OK".
Abnormal end	The ERR LED turns on and the dot matrix LED indicates "ERR" and the error number alternately at intervals of one second.

■Error number when the test completes with an error

The dot matrix LED indicates the error number with the form of "Target Ethernet port Error number".

For example, "1 3" is displayed when error No.3 occurs in P1.

If the module communication test fails, take the following actions.

Error number	Description	Action
1	Internal self-loopback test failure	Please consult your local Mitsubishi representative.
2	External self-loopback test connection error	Check the Ethernet cable connection or replace the Ethernet cable, and perform the test again. If the test fails again, please consult your local Mitsubishi representative.
3	External self-loopback test communication error	Replace the Ethernet cable and perform the test again. If the test fails again, please consult your local Mitsubishi representative.

4.8 Mounting and Removing a Module

This section describes the procedure for mounting/removing an Ethernet-equipped module.

■Procedure for replacing an Ethernet-equipped module

1. Power off the station where an Ethernet-equipped module is mounted.
2. Disconnect the Ethernet cable and remove the Ethernet-equipped module.
3. Start up the new Ethernet-equipped module according to Page 4 - 3 Settings and Procedure before Operation.
4. Reset the external device.

Point

When the Ethernet-equipped module is replaced, reset the external device.

When the external device holds the Ethernet address of the communication destination, communications may not be continued because the Ethernet address is changed by the Ethernet-equipped module replacement. For the same reason, restart the Ethernet-equipped module after replacing an external device (such as a personal computer).

■Procedure for replacing a CPU module

1. Read the parameters for the Ethernet-equipped module from the CPU module and save them in GX Works3.*1
2. Replace the CPU module.

📖 MELSEC iQ-R CPU Module User's Manual (Startup)

3. Write the parameters for the Ethernet-equipped module that have been saved in GX Works3 to the new CPU module.
4. Reset the external device.

*1 When parameters for the Ethernet-equipped module are created and changed, recording and saving them is recommended even if the CPU module is not replaced.

MEMO

5 EXERCISE 1 (COMMUNICATIONS USING MX COMPONENT BETWEEN A PERSONAL COMPUTER AND CPU MODULE)

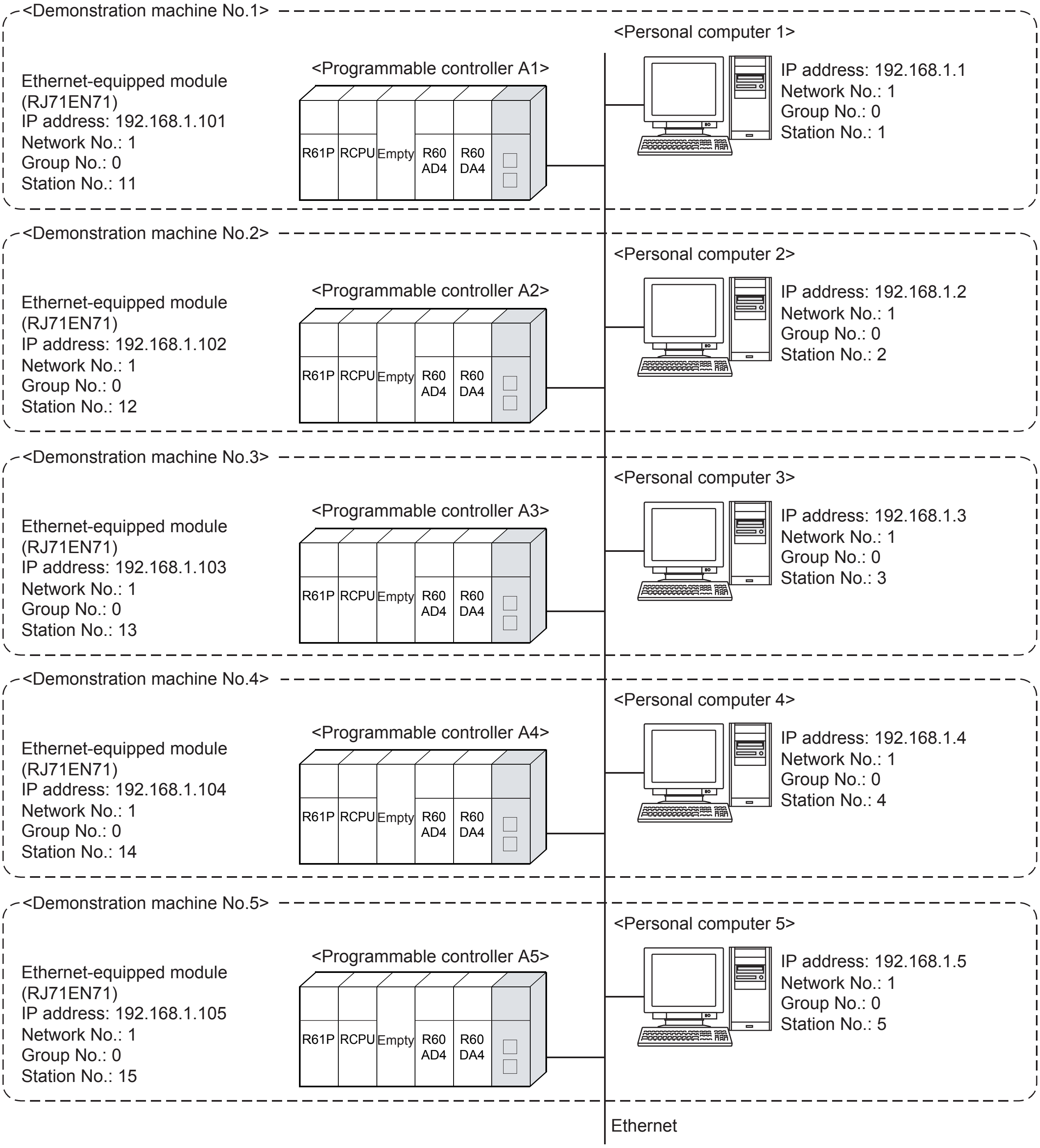
This exercise is for the communication function using MX Component. The personal computer (external device) reads/writes information to/from the CPU module using MX Component.

5.1 System Configuration

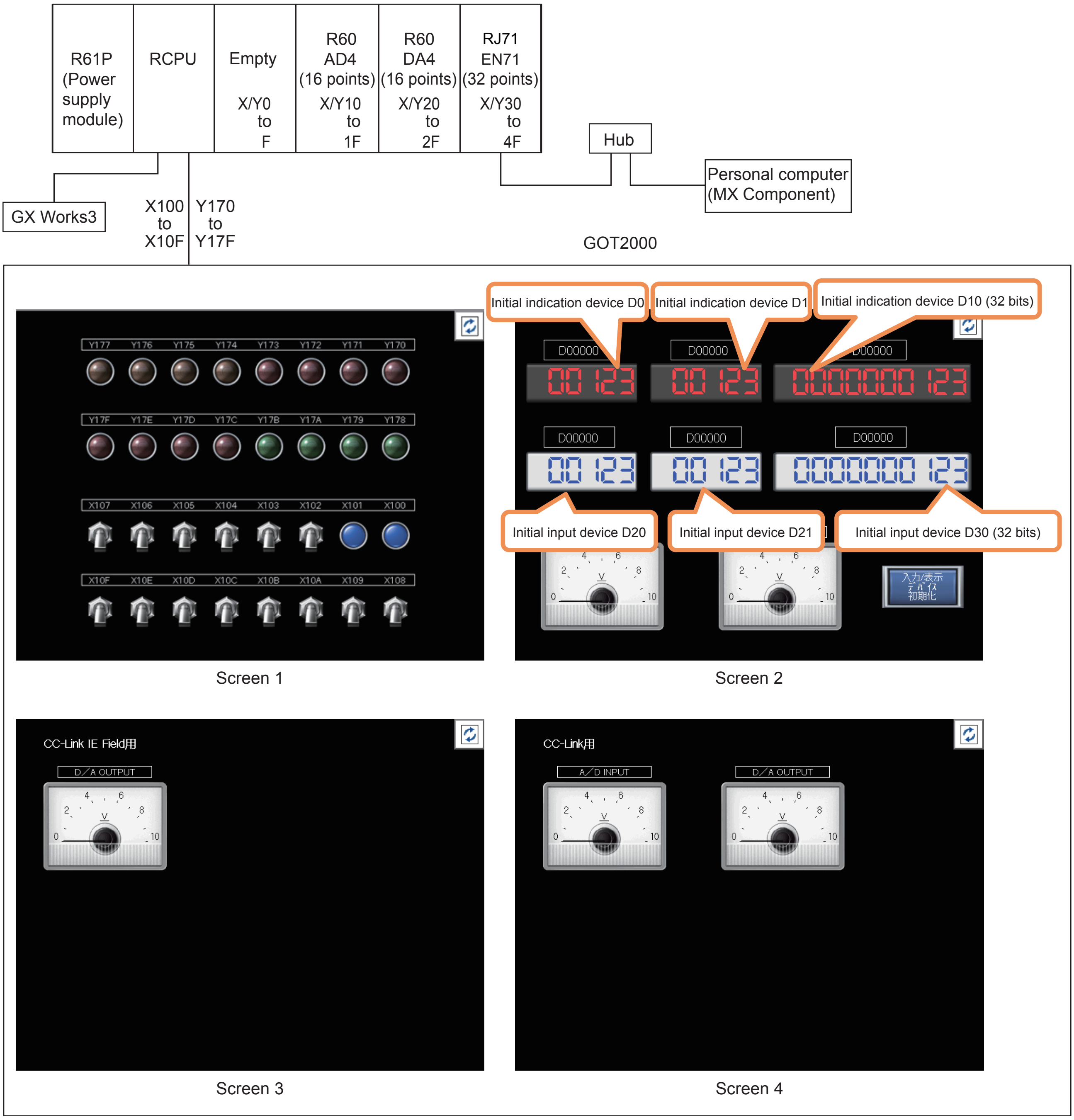
The following figure shows the system configuration of Exercise 1.

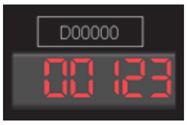
Although 10 devices are connected on the same Ethernet network, perform communications only between the devices with the same demonstration machine number in Exercise 1.

Module configuration



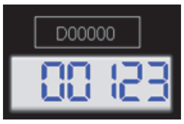
I/O assignment


R61P (Power supply module)
RCPU
Empty
X/Y0 to F
R60 AD4 (16 points)
X/Y10 to 1F
R60 DA4 (16 points)
X/Y20 to 2F
RJ71 EN71 (32 points)
X/Y30 to 4F
Hub
Personal computer (MX Component)
GX Works3
X100 to X10F
Y170 to Y17F
GOT2000
Y177 Y176 Y175 Y174 Y173 Y172 Y171 Y170
Y17F Y17E Y17D Y17C Y17B Y17A Y179 Y178
X107 X106 X105 X104 X103 X102 X101 X100
X10F X10E X10D X10C X10B X10A X109 X108
Initial indication device D0
Initial indication device D1
Initial indication device D10 (32 bits)
D00000
00 123
0000000 123
Initial input device D20
Initial input device D21
Initial input device D30 (32 bits)
入力/表示デバイス初期化
Screen 1
Screen 2
CC-Link IE Field用
D／A OUTPUT
CC-Link用
A／D INPUT
D／A OUTPUT
Screen 3
Screen 4



Upper section: The indication device can be changed.

Lower section: Data is displayed.



Upper section: The input device can be changed.

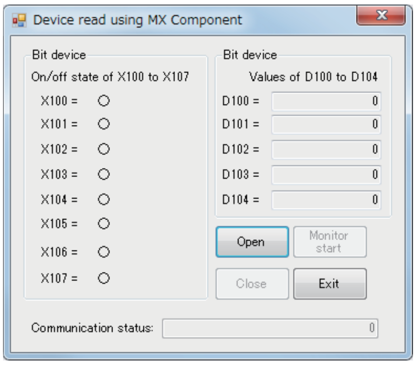
Lower section: The input data can be set and displayed.

- Touching 🔄 switches the screen.
- The initial value is automatically set to a device number in the upper section (trigger action function).
- Touching the "Initialize Input/Indication Device" button also initializes the device number.

Exercise

Monitor the devices and write data to the device using a program created with MX Component.

<Device read>



Device read using MX Component

Bit device
On/off state of X100 to X107

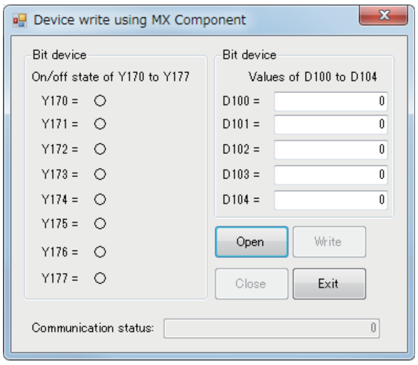
X100 = ○
X101 = ○
X102 = ○
X103 = ○
X104 = ○
X105 = ○
X106 = ○
X107 = ○

Bit device
Values of D100 to D104

D100 =	0
D101 =	0
D102 =	0
D103 =	0
D104 =	0

Communication status: 0

<Device write>



Device write using MX Component

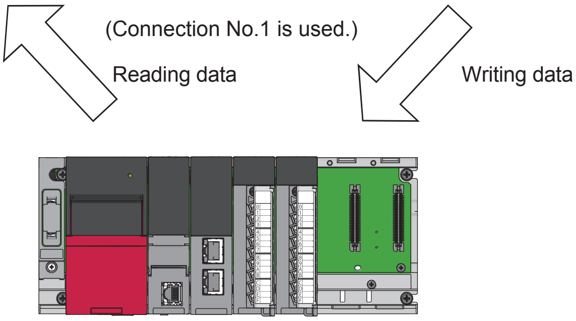
Bit device
On/off state of Y170 to Y177

Y170 = ○
Y171 = ○
Y172 = ○
Y173 = ○
Y174 = ○
Y175 = ○
Y176 = ○
Y177 = ○

Bit device
Values of D100 to D104

D100 =	0
D101 =	0
D102 =	0
D103 =	0
D104 =	0

Communication status: 0



CPU module

5.2 Parameter Settings and TCP/IP Settings for the Personal Computer

Before starting communications with the personal computer, set the parameters of the Ethernet-equipped module with GX Works3.

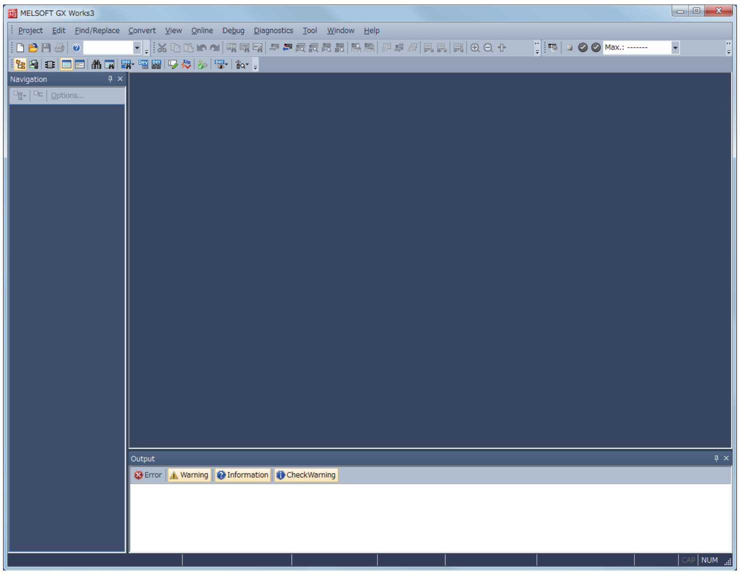
Set TCP/IP of the personal computer where MX Component has been installed.

Set them as described on Page 5 - 2 Module configuration.

5.2.1 Starting GX Works3

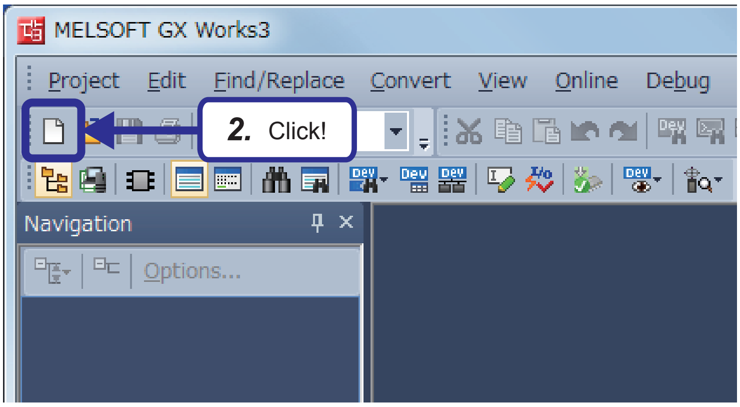
Start GX Works3 to set parameters.

Operating procedure


MELSOFT GX Works3
Project Edit Find/Replace Convert View Online Debug Diagnostics Tool Window Help
Navigation
Options...
Output
Error Warning Information CheckWarning
NUM

1. Click [MELSOFT] ⇨ [GX Works3] ⇨ [GX Works3] from the Windows® Start menu[*1].

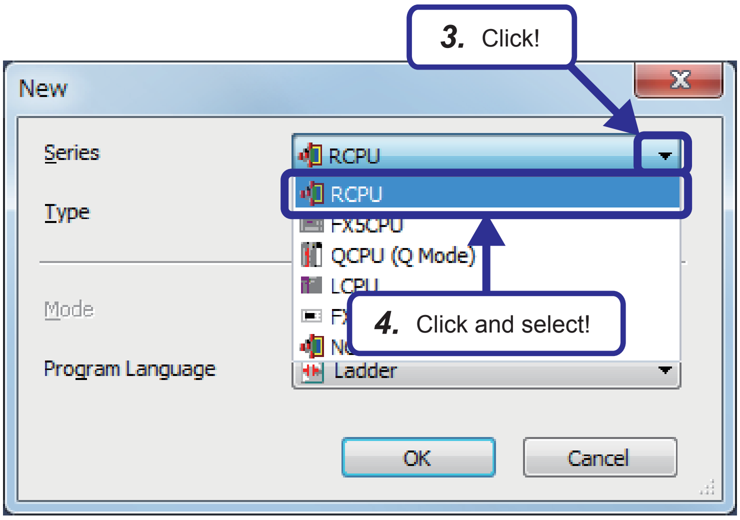
[*1] Select [Start] ⇨ [All apps] or [Start]⇨ [All Programs].



2. Click 🗋 on the toolbar or select [Project] ⇨ [New] (Ctrl + N) from the menu.

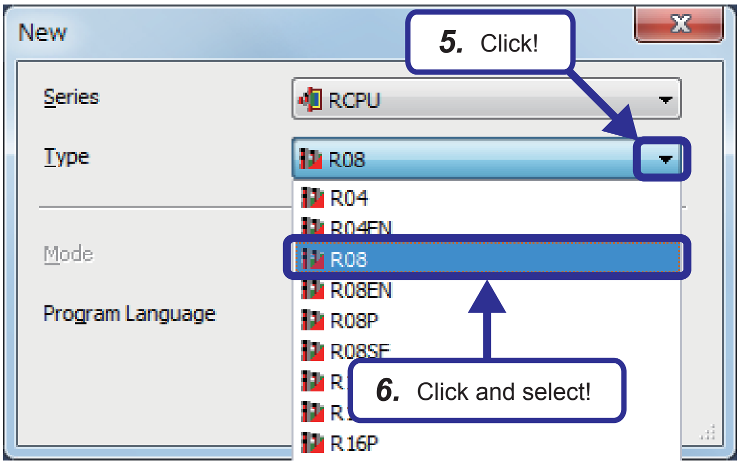
(To the next page)

(From the previous page)



3. Click the list button of "Series".

4. Select "RCPU" from the drop-down menu.

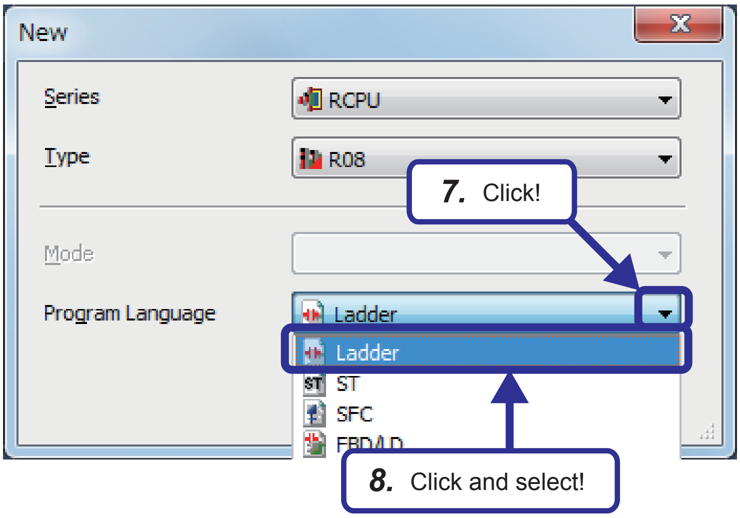


5. Click the list button of "Type".

6. Select "R08" from the drop-down menu.

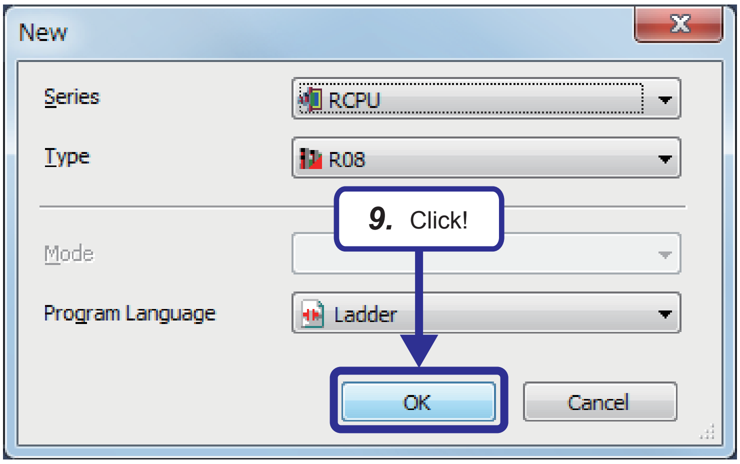
(To the next page)

(From the previous page)

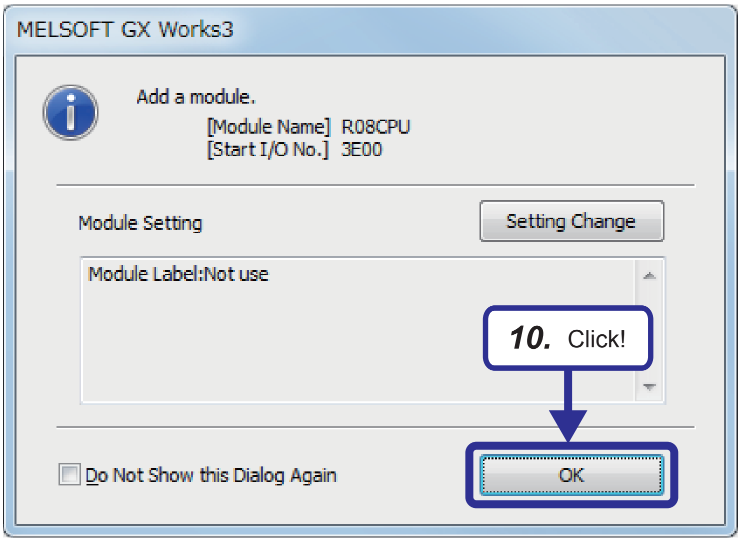


7. Click the list button of "Program Language".

8. Select "Ladder" from the drop-down menu.



9. Click the [OK] button.


MELSOFT GX Works3

Add a module.

[Module Name] R08CPU

[Start I/O No.] 3E00

Module Setting

Module Label:Not use

□ Do Not Show this Dialog Again

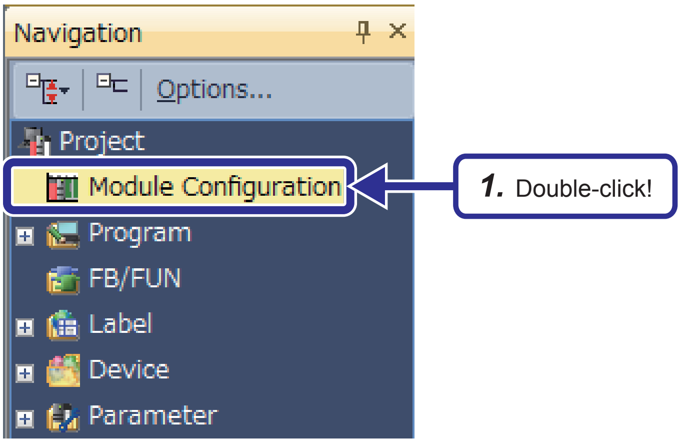
10. Click the [OK] button.

5.2.2 Setting parameters with GX Works3

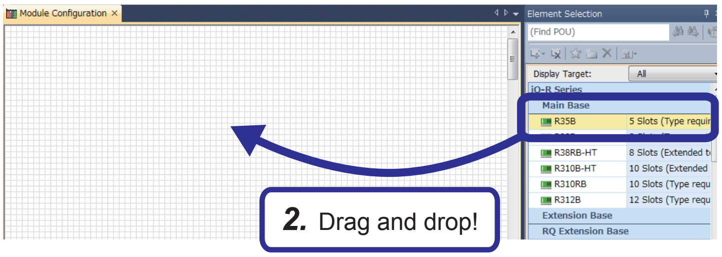
Adding Ethernet-equipped module data

Adding an Ethernet-equipped module in the "Module Configuration" window enables users to set parameters of the Ethernet-equipped module.

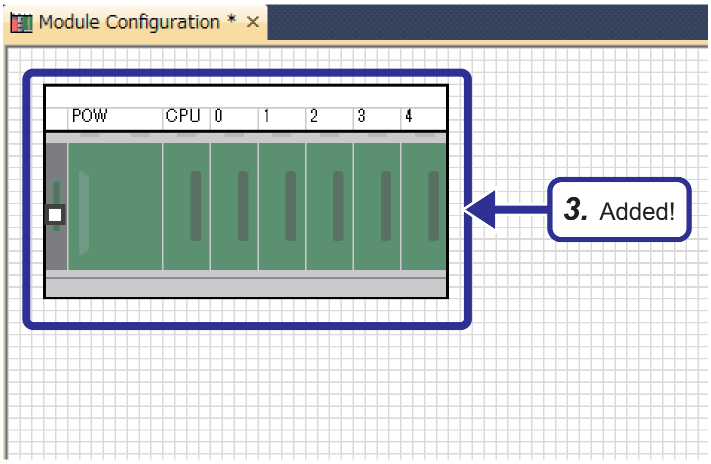
Operating procedure



1. Double-click [Module Configuration] in the "Navigation" window.
When the dialog box about parameter information appears, click the [OK] button.



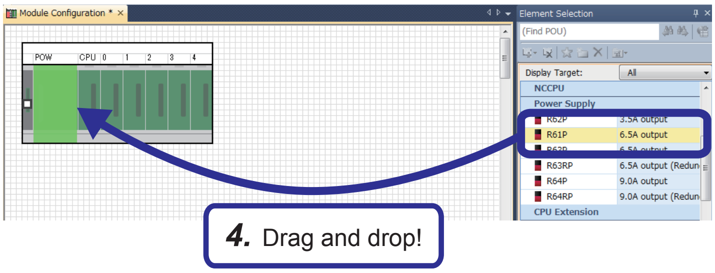
2. The "Module Configuration" window appears. Select "R35B" from "Main Base" in the "Element Selection" window, and drag and drop it to the "Module Configuration" window.



3. "R35B" is added on the "Module Configuration" window.

(To the next page)

(From the previous page)



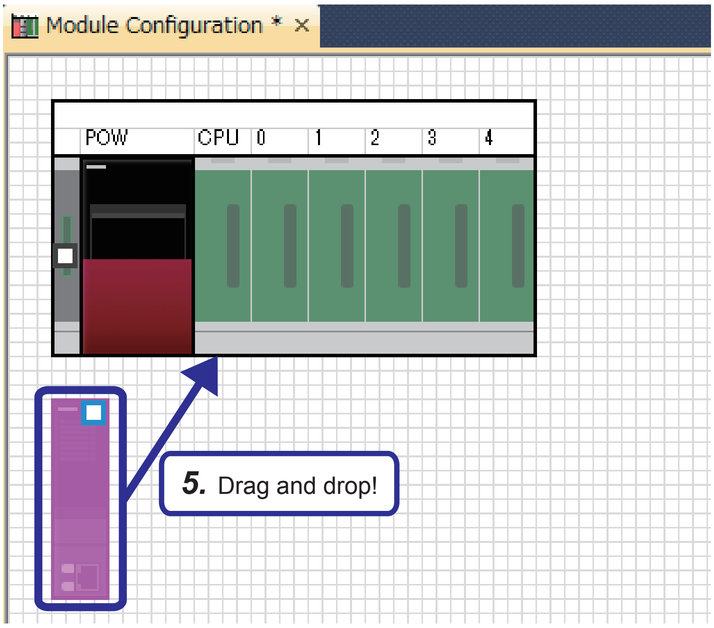
Element Selection

(Find POU)

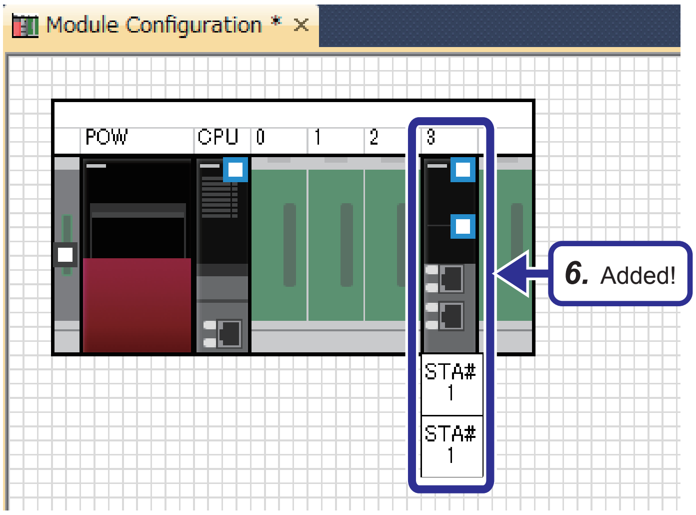
Display Target: All

NCCPU	
Power Supply	
R62P	3.5A output
R61P	6.5A output
R63P	6.5A output
R63RP	6.5A output (Redun
R64P	9.0A output
R64RP	9.0A output (Redun
CPU Extension	

4. Select "R61P" from "Power Supply" in the "Element Selection" window, and drag and drop it to the power supply slot of the R35B on the "Module Configuration" window.
While the power supply module is being dragged and dropped, the slot where the power supply module can be arranged is highlighted.



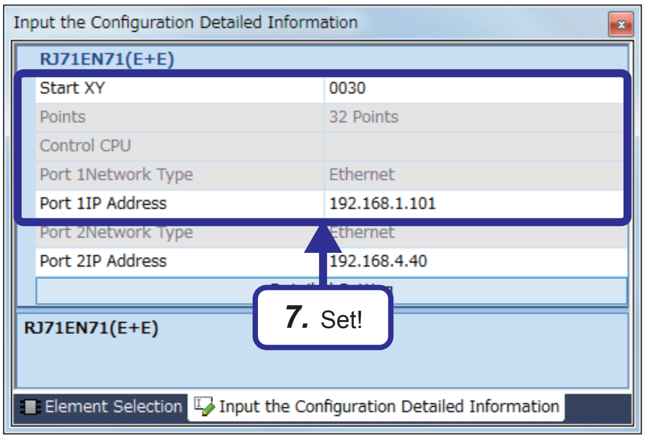
5. Add the R08CPU that has already been arranged in the "Module Configuration" window to the CPU slot of the R35B.
When the R08CPU has not been arranged in the "Module Configuration" window, add the R08CPU from the "Element Selection" window in the same way as for the power supply module.



6. Add "RJ71EN71(E+E)" from "Information Module" in the "Element Selection" window to the slot No.3 of the R35B.

(To the next page)

(From the previous page)

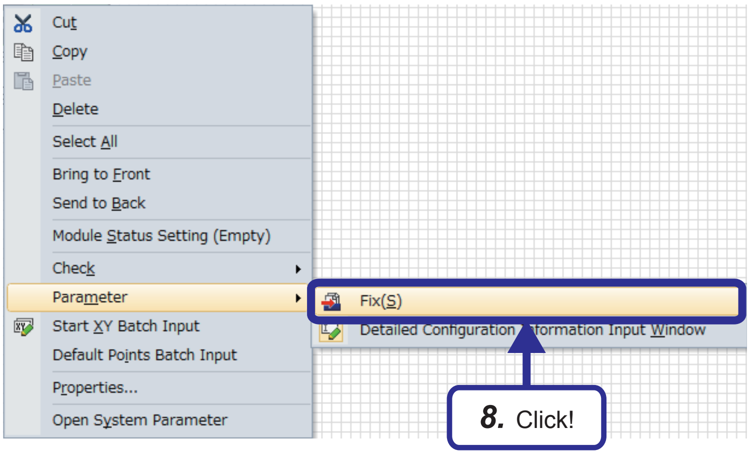

Input the Configuration Detailed Information

RJ71EN71(E+E)

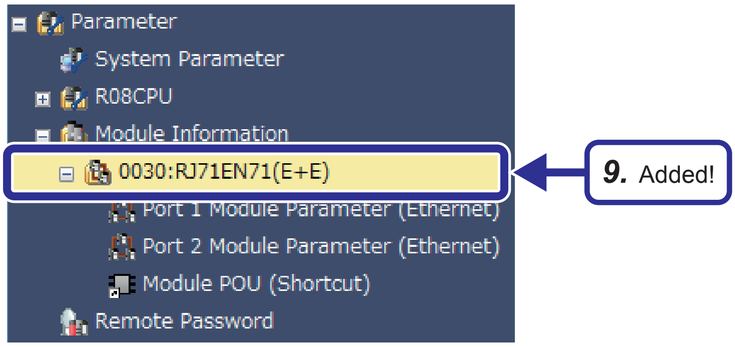
Start XY	0030
Points	32 Points
Control CPU	
Port 1Network Type	Ethernet
Port 1IP Address	192.168.1.101
Port 2Network Type	Ethernet
Port 2IP Address	192.168.4.40

7. Right-click the RJ71EN71(E+E), and click [Parameter] ⇨ [Detailed Configuration Information Input Window] from the menu to display the "Input the Configuration Detailed Information" window. Set parameters as follows.

[Parameters to be set]
Start XY: 0030
Port 1IP address: Set the IP address (192.168.1.101 to 192.168.1.105) of the demonstration machine used.



8. After setting the parameters, right-click the RJ71EN71(E+E) and click [Parameter] ⇨ [Fix] from the menu to fix the parameters.
(Click the [OK] button when the confirmation window for adding the module label appears.)



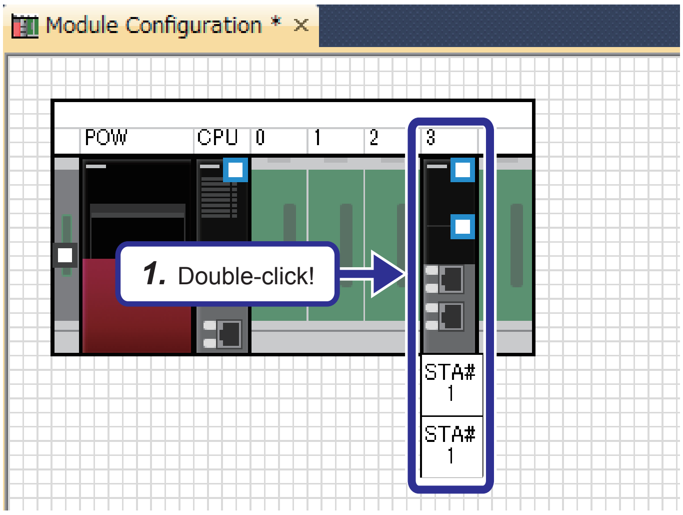
9. The data of the specified Ethernet-equipped module is added to the "Navigation" window.

Setting parameters

Set parameters of the Ethernet-equipped module.

Operating procedure

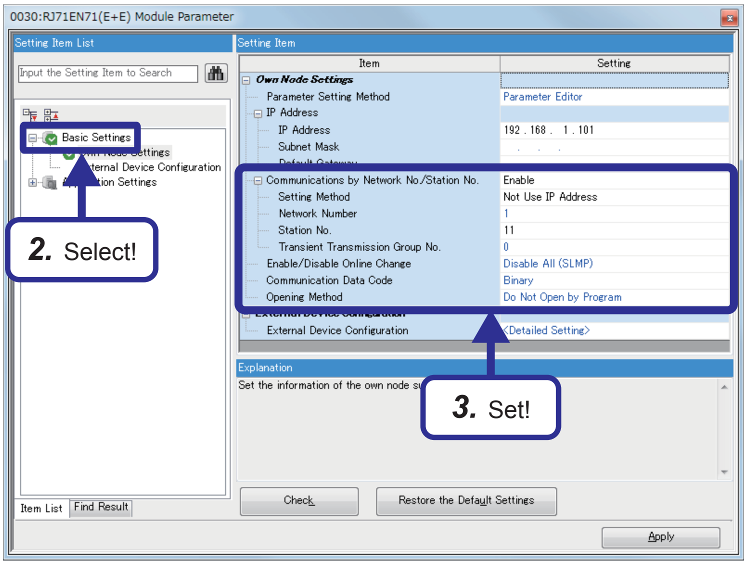
1. Double-click the RJ71EN71(E+E) module in the "Module Configuration" window.



2. Select "Basic Settings" in "Setting Item List".

3. Set "Communications by Network No./Station No." as follows.

[Parameters to be set]
Communications by Network No./Station No.: Enable
Setting Method: Not Use IP Address
Network Number: 1
Station No.: Set the station number (11 to 15) of the demonstration machine used.

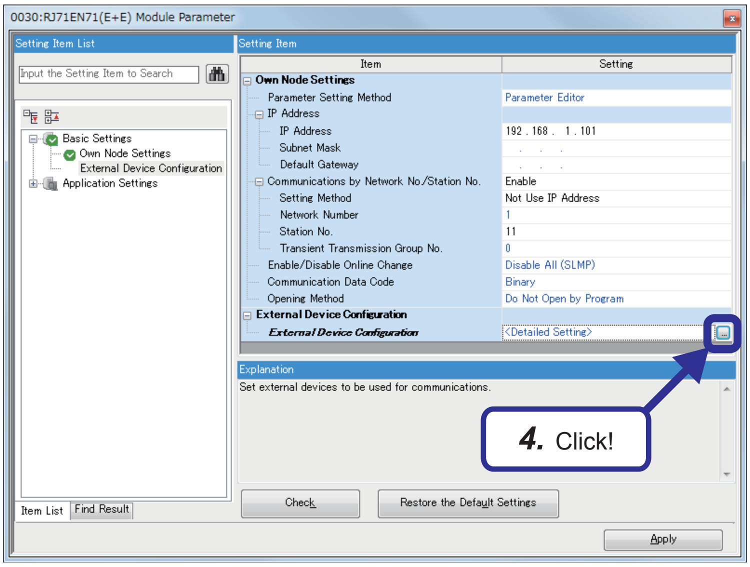

0030:RJ71EN71(E+E) Module Parameter

Item	Setting
Own Node Settings	
Parameter Setting Method	Parameter Editor
IP Address	
IP Address	192 . 168 . 1 . 101
Subnet Mask	. . .
Communications by Network No./Station No.	Enable
Setting Method	Not Use IP Address
Network Number	1
Station No.	11
Transient Transmission Group No.	0
Enable/Disable Online Change	Disable All (SLMP)
Communication Data Code	Binary
Opening Method	Do Not Open by Program
External Device Configuration	<Detailed Setting>

Explanation
Set the information of the own node s

(To the next page)

(From the previous page)

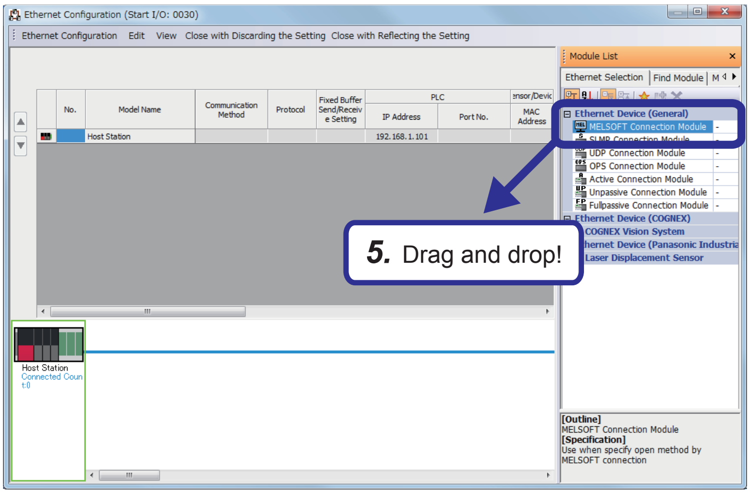

0030:RJ71EN71(E+E) Module Parameter
Setting Item List
Input the Setting Item to Search
Basic Settings
Own Node Settings
External Device Configuration
Application Settings
Item List
Find Result
Setting Item
Explanation
Check
Restore the Default Settings
Apply

Item	Setting
Own Node Settings	
Parameter Setting Method	Parameter Editor
IP Address	
IP Address	192 . 168 . 1 . 101
Subnet Mask	. . .
Default Gateway	. . .
Communications by Network No./Station No.	Enable
Setting Method	Not Use IP Address
Network Number	1
Station No.	11
Transient Transmission Group No.	0
Enable/Disable Online Change	Disable All (SLMP)
Communication Data Code	Binary
Opening Method	Do Not Open by Program
External Device Configuration	
External Device Configuration	<Detailed Setting>

Set external devices to be used for communications.

4. Click!

4. Click the ▢ button of "Detailed Setting" of "External Device Configuration".


Ethernet Configuration (Start I/O: 0030)
Ethernet Configuration
Edit
View
Close with Discarding the Setting
Close with Reflecting the Setting
Module List
Ethernet Selection
Find Module
M
Ethernet Device (General)
MELSOFT Connection Module
SLMP Connection Module
UDP Connection Module
OPS Connection Module
Active Connection Module
Unpassive Connection Module
Fullpassive Connection Module
Ethernet Device (COGNEX)
COGNEX Vision System
hernet Device (Panasonic Industria
Laser Displacement Sensor
Host Station
Connected Coun
t:0

	No.	Model Name	Communication Method	Protocol	Fixed Buffer Send/Receive Setting	PLC IP Address	PLC Port No.	ensor/Devic MAC Address
		Host Station				192.168.1.101		

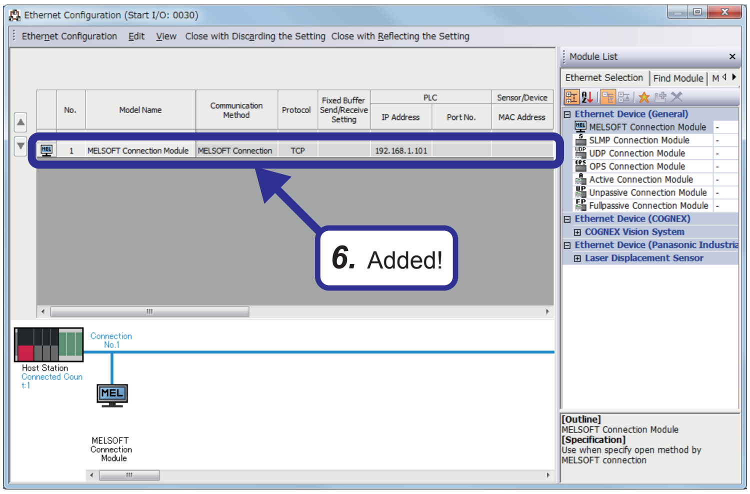
[Outline]
MELSOFT Connection Module
[Specification]
Use when specify open method by MELSOFT connection

5. Drag and drop!

5. The "Ethernet Configuration" dialog box appears. Select "MELSOFT Connection Module" from "Ethernet Device (General)" in "Module List" and drag and drop it to the list of devices or device map area.

(To the next page)

(From the previous page)



Ethernet Configuration (Start I/O: 0030)

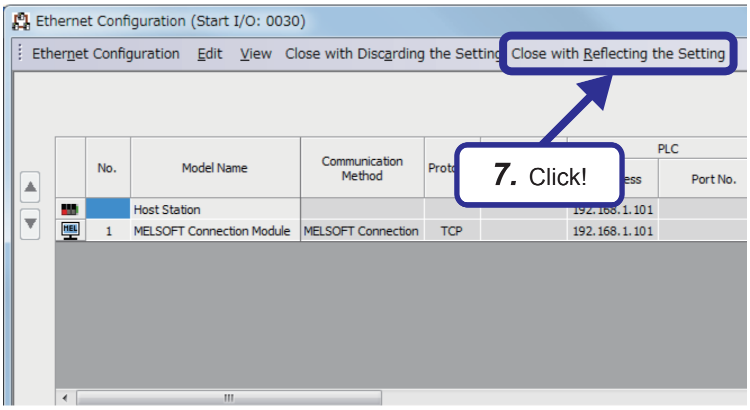
No.	Model Name	Communication Method	Protocol	Fixed Buffer Send/Receive Setting	PLC IP Address	PLC Port No.	Sensor/Device MAC Address
1	MELSOFT Connection Module	MELSOFT Connection	TCP		192.168.1.101		

6. Added!

[Outline]
MELSOFT Connection Module
[Specification]
Use when specify open method by MELSOFT connection

Ethernet Configuration
Edit
View
Close with Discarding the Setting
Close with Reflecting the Setting
Module List
Ethernet Selection
Find Module
Ethernet Device (General)
MELSOFT Connection Module
SLMP Connection Module
UDP Connection Module
OPS Connection Module
Active Connection Module
Unpassive Connection Module
Fullpassive Connection Module
Ethernet Device (COGNEX)
COGNEX Vision System
Ethernet Device (Panasonic Industria
Laser Displacement Sensor
Connection No.1
Host Station
Connected Coun
1:1
MEL
MELSOFT Connection Module

6. "MELSOFT Connection Module" is added to the list of devices.



Ethernet Configuration (Start I/O: 0030)

	No.	Model Name	Communication Method	Prot	ess	Port No.
		Host Station			192.168.1.101	
	1	MELSOFT Connection Module	MELSOFT Connection	TCP	192.168.1.101	

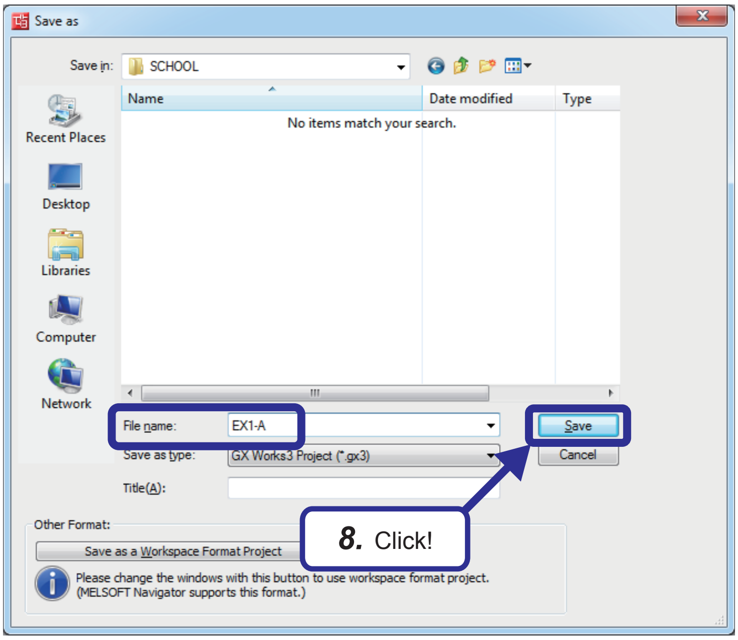
7. Click!

Ethernet Configuration
Edit
View
Close with Discarding the Setting
Close with Reflecting the Setting
PLC

7. After setting the parameters, click "Close with Reflecting the Setting" in the menu to close the "Ethernet Configuration" dialog box.

(To the next page)

(From the previous page)



Save as

Save in: SCHOOL

No items match your search.

File name: EX1-A

Save as type: GX Works3 Project (*.gx3)

Title(A):

Please change the windows with this button to use workspace format project.
(MELSOFT Navigator supports this format.)

8. Click [Project] ⇨ [Save As] from the menu to save the project.

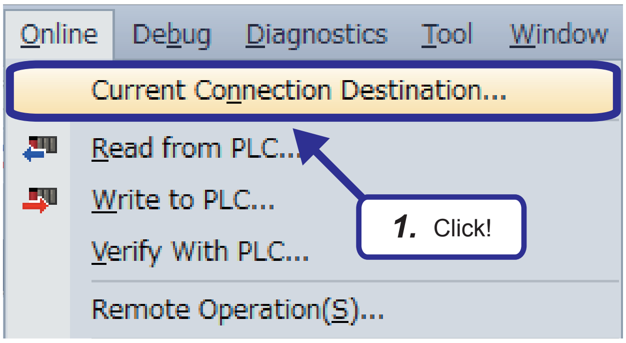
Save destination: Desired location
File name: EX1-A
Title: Blank

5.2.3 Specifying a connection destination

Specify a connection destination.

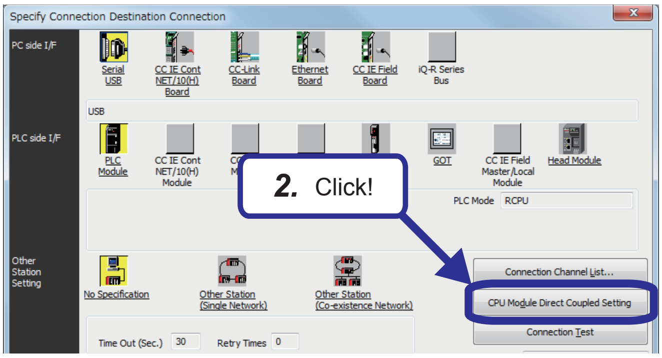
Operating procedure

1. Select [Online] ⇨ [Current Connection Destination] from the menu of the engineering tool.



2. Click the [CPU Module Direct Coupled Setting] button on the "Specify Connection Destination Connection" window.

The "CPU Module Direct Coupled Setting" dialog box appears.

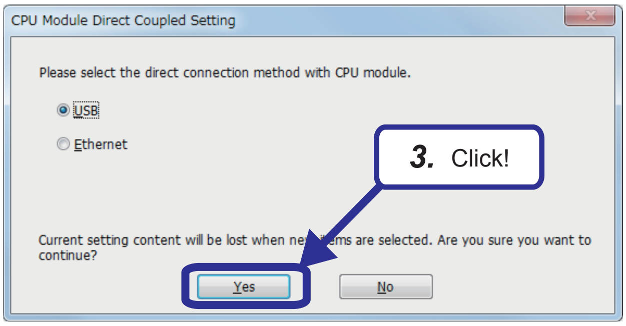

Specify Connection Destination Connection
PC side I/F
Serial USB
CC IE Cont NET/10(H) Board
CC-Link Board
Ethernet Board
CC IE Field Board
iQ-R Series Bus
PLC side I/F
PLC Module
CC IE Cont NET/10(H) Module
GOT
CC IE Field Master/Local Module
Head Module
Other Station Setting
No Specification
Other Station (Single Network)
Other Station (Co-existence Network)
Connection Channel List...
CPU Module Direct Coupled Setting
Connection Test
2. Click!

USB

PLC Mode RCPU

Time Out (Sec.) 30 Retry Times 0

3. Select the connection method, and click the [Yes] button.



Please select the direct connection method with CPU module.

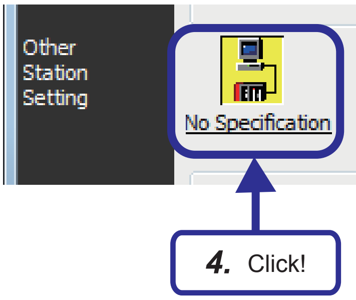
● USB

○ Ethernet

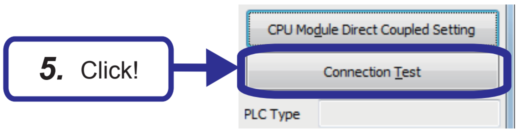
Current setting content will be lost when new items are selected. Are you sure you want to continue?

(To the next page)

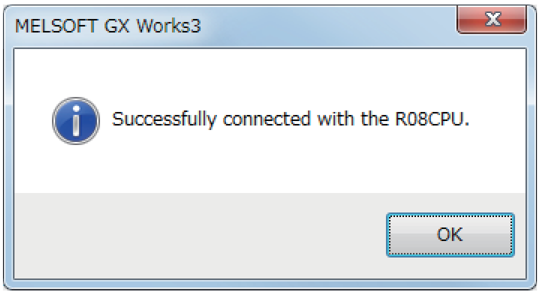
(From the previous page)



4. Click "No Specification" of "Other Station Setting".

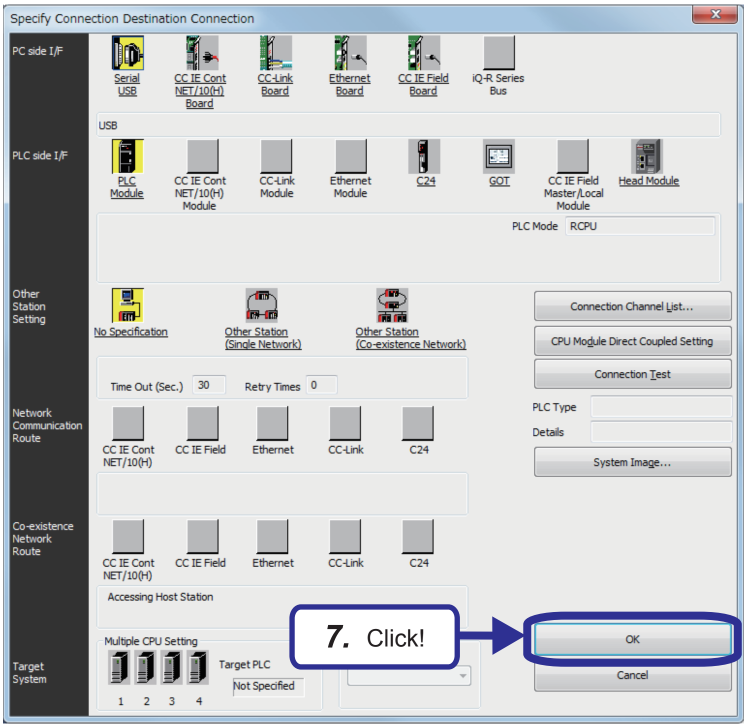


5. Click the [Connection Test] button.



Successfully connected with the R08CPU.

6. Check that the connection between the Ethernet-equipped module and CPU module is successfully established.



USB

PLC Mode RCPU

Time Out (Sec.) 30

Retry Times 0

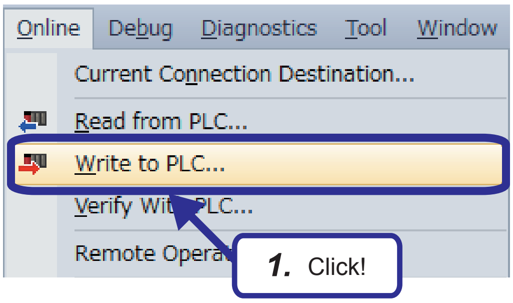
Target PLC Not Specified

7. Click the [OK] button.

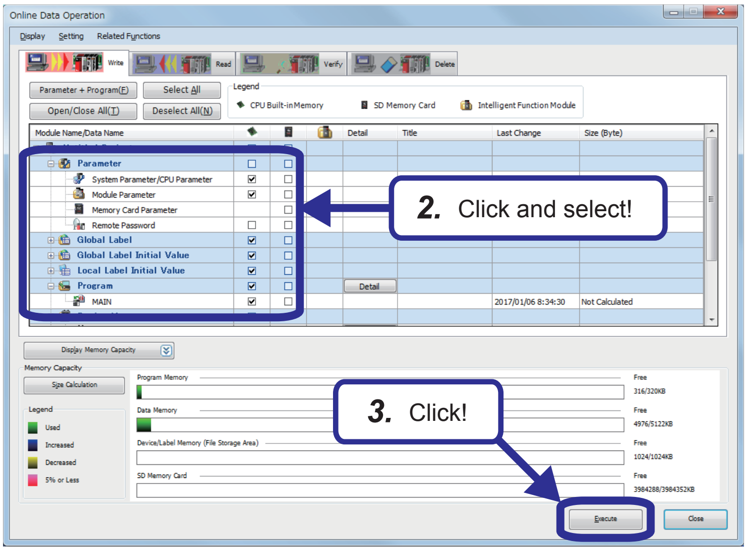
5.2.4 Writing parameters

Write the set parameters to the CPU module.

Operating procedure



1. Select [Online] ⇨ [Write to PLC] from the menu of the engineering tool.



Module Name/Data Name				Detail	Title	Last Change	Size (Byte)
Parameter	□	□					
System Parameter/CPU Parameter	☑	□					
Module Parameter	☑	□					
Memory Card Parameter		□					
Remote Password	□	□					
Global Label	☑	□					
Global Label Initial Value	☑	□					
Local Label Initial Value	☑	□					
Program	☑	□		Detail			
MAIN	☑	□				2017/01/06 8:34:30	Not Calculated

Program Memory — Free 316/320KB

Data Memory — Free 4976/5122KB

Device/Label Memory (File Storage Area) — Free 1024/1024KB

SD Memory Card — Free 3984288/3984352KB

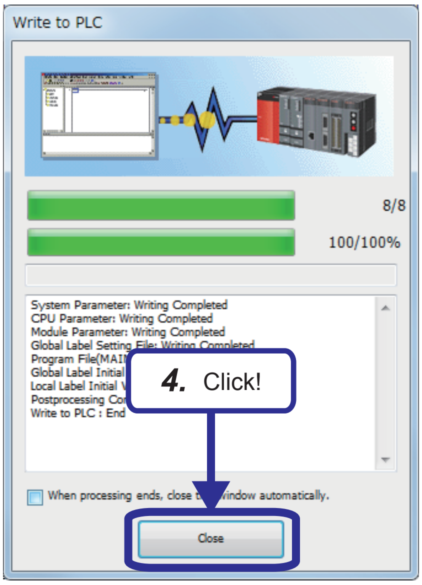
2. The "Online Data Operation" dialog box appears. Select the following items.

- System Parameter/CPU Parameter
- Module Parameter

3. Click the [Execute] button.

(To the next page)

(From the previous page)



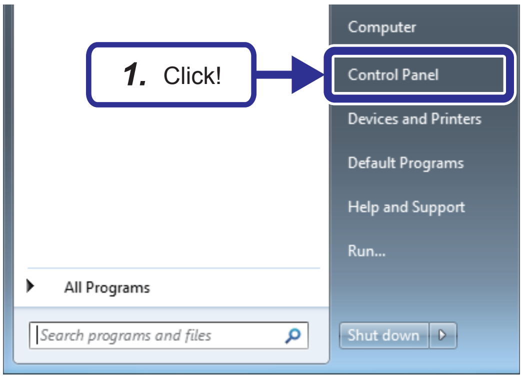
```
System Parameter: Writing Completed
CPU Parameter: Writing Completed
Module Parameter: Writing Completed
Global Label Setting File: Writing Completed
Program File(MAI
Global Label Initial
Local Label Initial V
Postprocessing Co
Write to PLC : End
```

4. The "Write to PLC" dialog box appears. When the write processing has completed, the message "Completed" is displayed. Click the [Close] button.

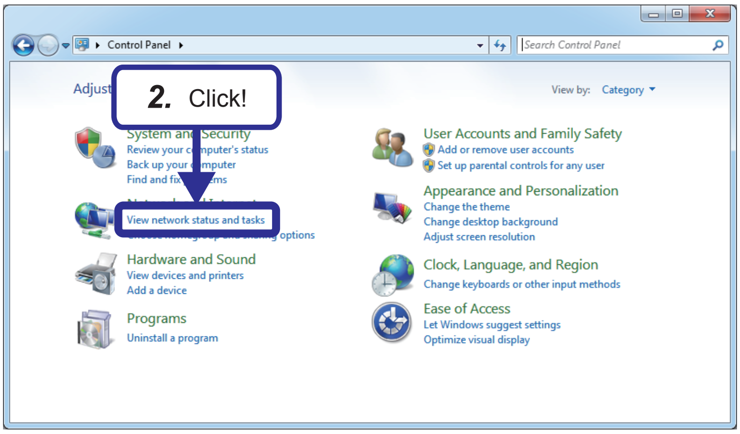
5.2.5 TCP/IP settings for the personal computer

Set TCP/IP for the personal computer.
This textbook describes the operating procedure for the personal computer that operates with the Microsoft® Windows® 7 Operating System.

Operating procedure



1. Click [Control Panel] from the Windows® Start menu.

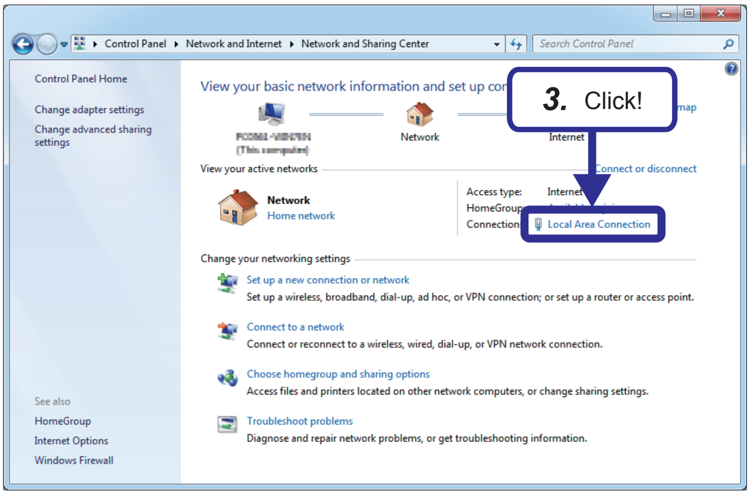

Control Panel
Search Control Panel
Adjust
2. Click!
View by: Category
System and Security
Review your computer's status
Back up your computer
Find and fix problems
View network status and tasks
options
Hardware and Sound
View devices and printers
Add a device
Programs
Uninstall a program
User Accounts and Family Safety
Add or remove user accounts
Set up parental controls for any user
Appearance and Personalization
Change the theme
Change desktop background
Adjust screen resolution
Clock, Language, and Region
Change keyboards or other input methods
Ease of Access
Let Windows suggest settings
Optimize visual display

2. The "Control Panel" dialog box appears. Click [View network status and tasks].

(To the next page)

(From the previous page)

3. Click [Local Area Connection].



Control Panel ▸ Network and Internet ▸ Network and Sharing Center

Search Control Panel

Control Panel Home

Change adapter settings

Change advanced sharing settings

View your basic network information and set up con... map

[redacted] — Network — Internet

View your active networks — Connect or disconnect

Network
Home network

Access type: Internet
HomeGroup:
Connections: Local Area Connection

Change your networking settings

Set up a new connection or network
Set up a wireless, broadband, dial-up, ad hoc, or VPN connection; or set up a router or access point.

Connect to a network
Connect or reconnect to a wireless, wired, dial-up, or VPN network connection.

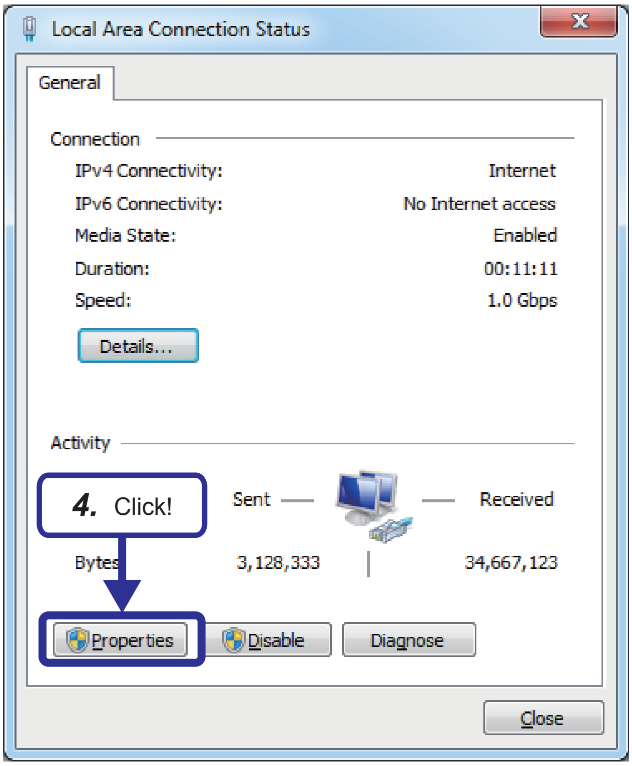
Choose homegroup and sharing options
Access files and printers located on other network computers, or change sharing settings.

Troubleshoot problems
Diagnose and repair network problems, or get troubleshooting information.

See also
HomeGroup
Internet Options
Windows Firewall

3. Click!

4. Click the [Properties] button.



Local Area Connection Status

General

Connection

IPv4 Connectivity:	Internet
IPv6 Connectivity:	No Internet access
Media State:	Enabled
Duration:	00:11:11
Speed:	1.0 Gbps

Details...

Activity

	Sent	Received
Bytes:	3,128,333	34,667,123

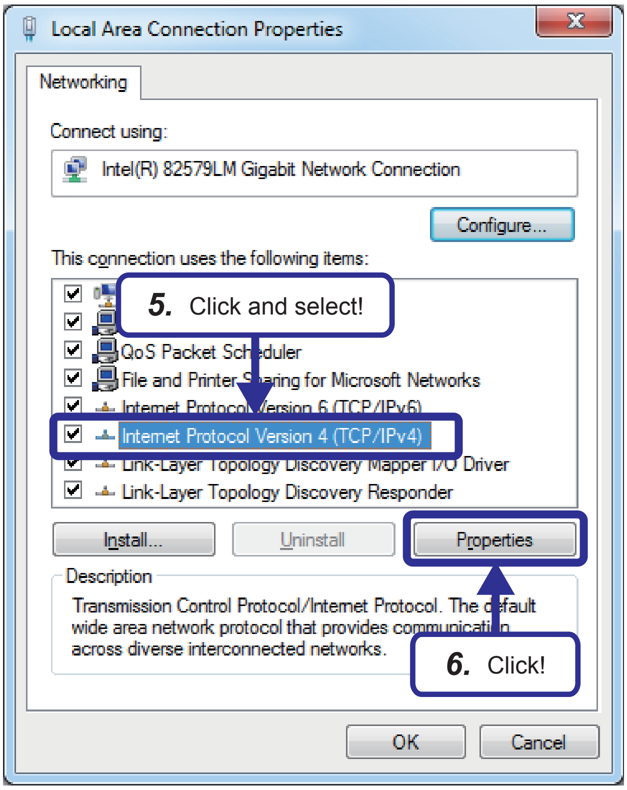
Properties Disable Diagnose

Close

4. Click!

(To the next page)

(From the previous page)



Local Area Connection Properties

Networking

Connect using:

Intel(R) 82579LM Gigabit Network Connection

Configure...

This connection uses the following items:

- ☑
- ☑
- ☑ QoS Packet Scheduler
- ☑ File and Printer Sharing for Microsoft Networks
- ☑ Internet Protocol Version 6 (TCP/IPv6)
- ☑ Internet Protocol Version 4 (TCP/IPv4)
- ☑ Link-Layer Topology Discovery Mapper I/O Driver
- ☑ Link-Layer Topology Discovery Responder

Install... Uninstall Properties

Description

Transmission Control Protocol/Internet Protocol. The default wide area network protocol that provides communication across diverse interconnected networks.

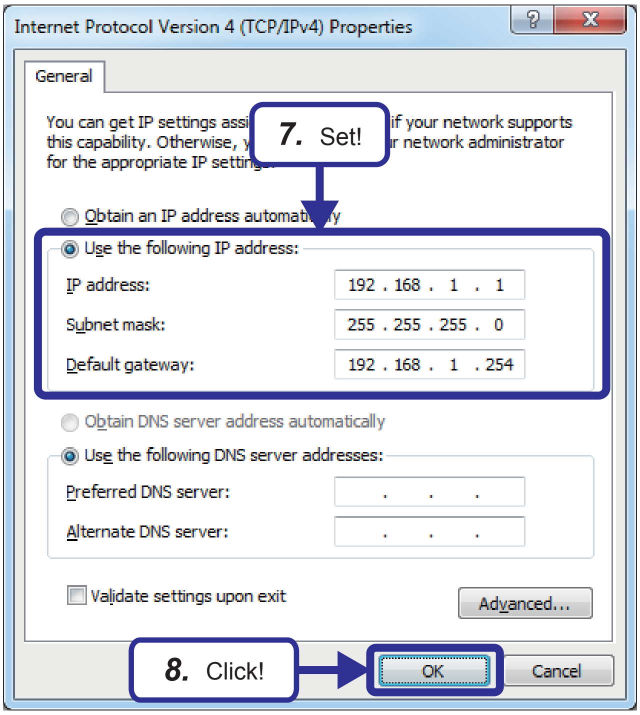
OK Cancel

5. Click and select!

6. Click!

5. Select "Internet Protocol Version 4 (TCP/IPv4)".

6. Click the [Properties] button.



Internet Protocol Version 4 (TCP/IPv4) Properties

General

You can get IP settings assi … if your network supports this capability. Otherwise, y … r network administrator for the appropriate IP settin…

○ Obtain an IP address automatically

● Use the following IP address:

IP address: 192 . 168 . 1 . 1

Subnet mask: 255 . 255 . 255 . 0

Default gateway: 192 . 168 . 1 . 254

○ Obtain DNS server address automatically

● Use the following DNS server addresses:

Preferred DNS server: . . .

Alternate DNS server: . . .

☐ Validate settings upon exit

Advanced...

OK Cancel

7. Set!

8. Click!

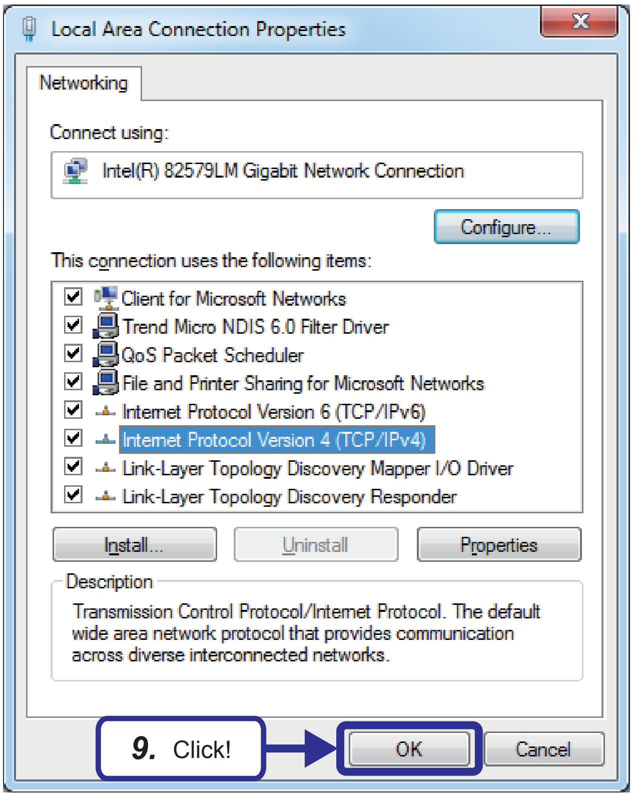
7. Select "Use the following IP address" and set the following items.

[Parameters to be set]
IP address: IP address (192.168.1.1 to 192.168.1.5) of the personal computer used
Subnet mask: 255.255.255.0 (Class C)
Default gateway: IP address of the router (192.168.1.254)

8. Click the [OK] button.

(To the next page)

(From the previous page)



Connect using:

Intel(R) 82579LM Gigabit Network Connection

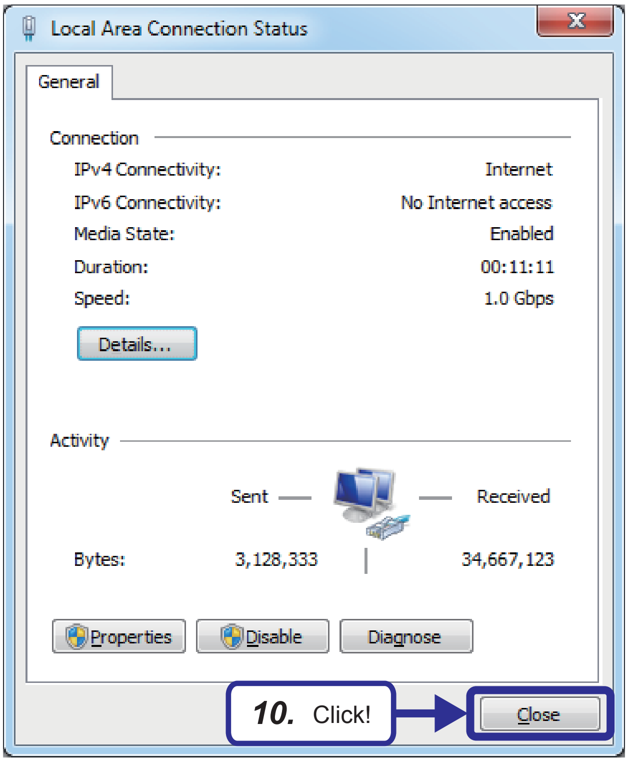
This connection uses the following items:

- ☑ Client for Microsoft Networks
- ☑ Trend Micro NDIS 6.0 Filter Driver
- ☑ QoS Packet Scheduler
- ☑ File and Printer Sharing for Microsoft Networks
- ☑ Internet Protocol Version 6 (TCP/IPv6)
- ☑ Internet Protocol Version 4 (TCP/IPv4)
- ☑ Link-Layer Topology Discovery Mapper I/O Driver
- ☑ Link-Layer Topology Discovery Responder

Description

Transmission Control Protocol/Internet Protocol. The default wide area network protocol that provides communication across diverse interconnected networks.

9. Click the [OK] button.



Connection

IPv4 Connectivity:	Internet
IPv6 Connectivity:	No Internet access
Media State:	Enabled
Duration:	00:11:11
Speed:	1.0 Gbps

Activity

	Sent	Received
Bytes:	3,128,333	34,667,123

10. Click the [Close] button.

5.3 Operating MX Component

This section describes how to operate MX Component.

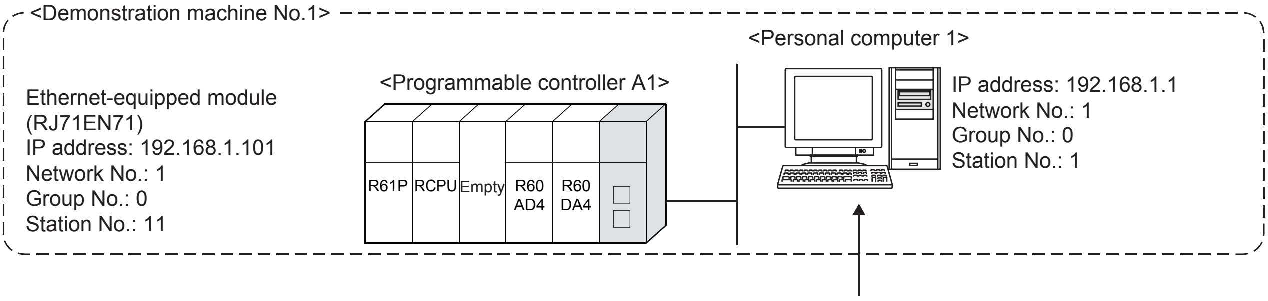
In this textbook, create an application with Visual Basic® 2012 and access the CPU module over Ethernet.

5.3.1 Setting the logical station number

The pieces of connection target information required to open a communication line are collected into one data by using the communication setup utility. A logical station number 0 to 1023 (0H to 3FFH) is assigned to the data.

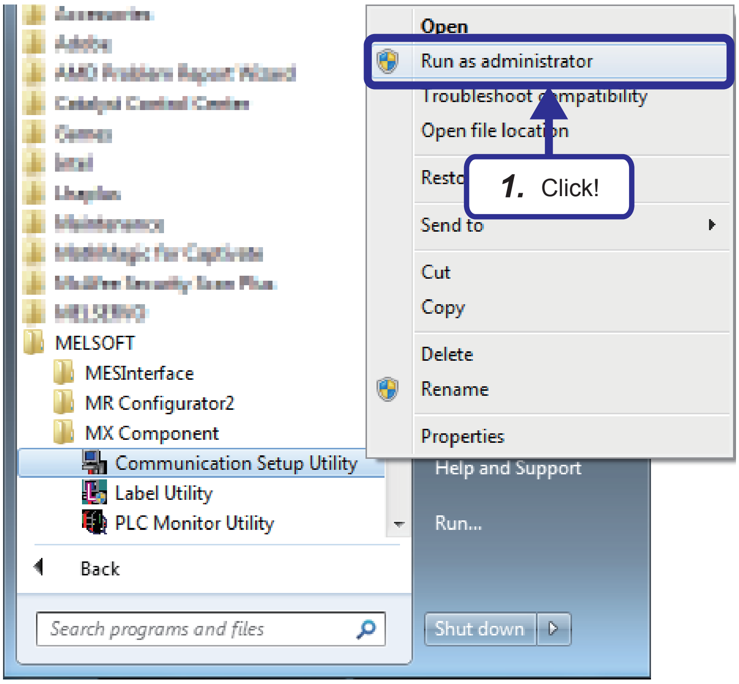
Set the logical station number to access the CPU module.

(Example) Ethernet communications



The pieces of connection target information on the RCPU are collected into one data, and a logical station number is assigned to the data.

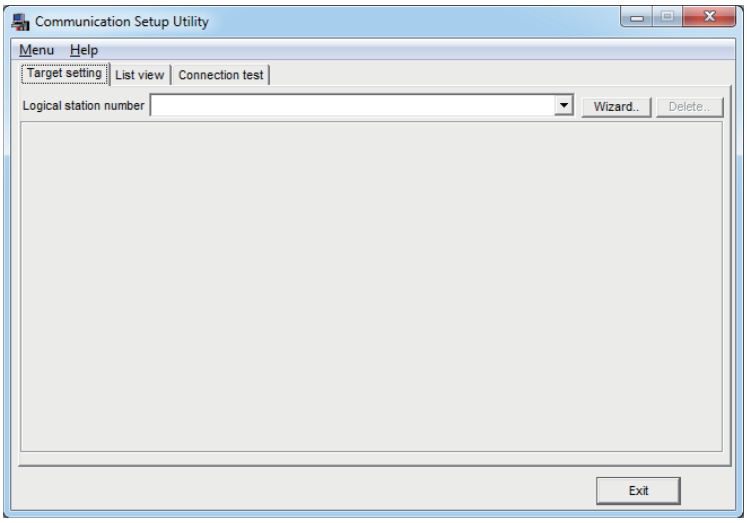
Operating procedure



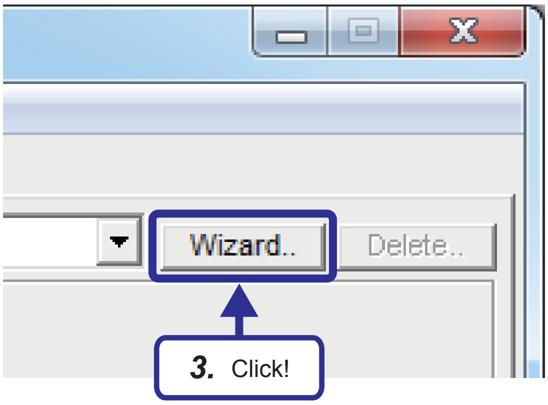
1. Click [All Programs] ⇨ [MELSOFT] ⇨ [MX Component], right-click "Communication Setup Utility", and click [Run as administrator] from the Windows® Start menu.

(To the next page)

(From the previous page)



2. The communication setup utility starts up.

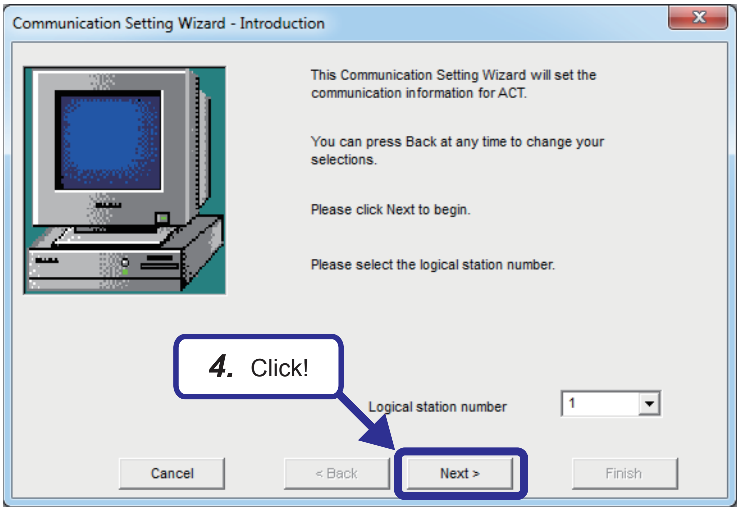


3. Click!

3. Click the [Wizard] button.

(To the next page)

(From the previous page)


Communication Setting Wizard - Introduction

This Communication Setting Wizard will set the communication information for ACT.

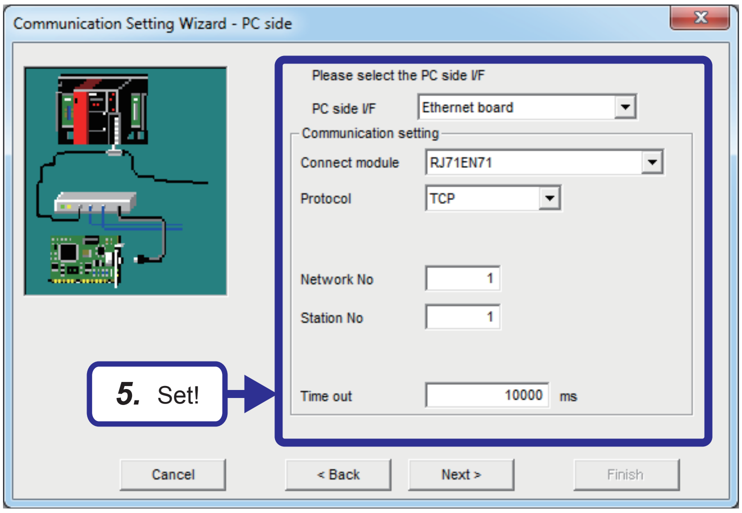
You can press Back at any time to change your selections.

Please click Next to begin.

Please select the logical station number.

Logical station number: 1

4. The window shown on the left appears. Set "Logical station number" to "1" and click the [Next] button.


Communication Setting Wizard - PC side

Please select the PC side I/F

PC side I/F: Ethernet board

Communication setting

Connect module: RJ71EN71

Protocol: TCP

Network No: 1

Station No: 1

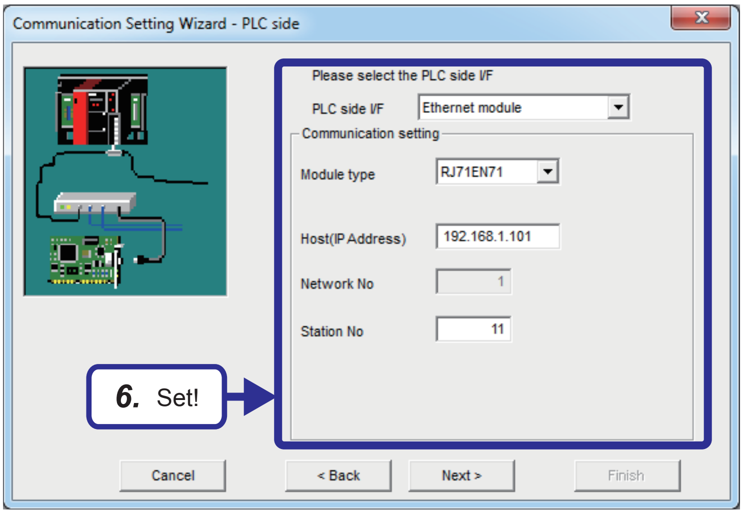
Time out: 10000 ms

5. Set the personal computer side as follows and click the [Next] button.

[Parameters to be set]
PC side I/F: Ethernet board
Connect module: RJ71EN71
Protocol: TCP
Network No: 1
Station No: Set the station number (1 to 5) of the demonstration machine used.
Time out: 10000ms (for monitoring the time to receive a response)

(To the next page)

(From the previous page)



Communication Setting Wizard - PLC side

Please select the PLC side I/F

PLC side I/F: Ethernet module

Communication setting

Module type: RJ71EN71

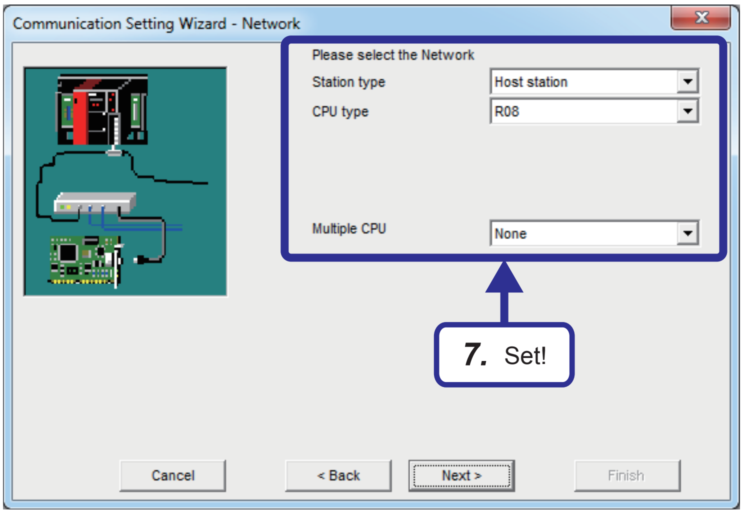
Host(IP Address): 192.168.1.101

Network No: 1

Station No: 11

6. Set the programmable controller side as follows and click the [Next] button.

[Parameters to be set]
PLC side I/F: Ethernet-equipped module
Module type: RJ71EN71
Host(IP address): Set the IP address (192.168.1.101 to 192.168.1.105) of the demonstration machine used.
Network No: (1)
Station No: Set the station number (11 to 15) of the demonstration machine used.



Communication Setting Wizard - Network

Please select the Network

Station type: Host station

CPU type: R08

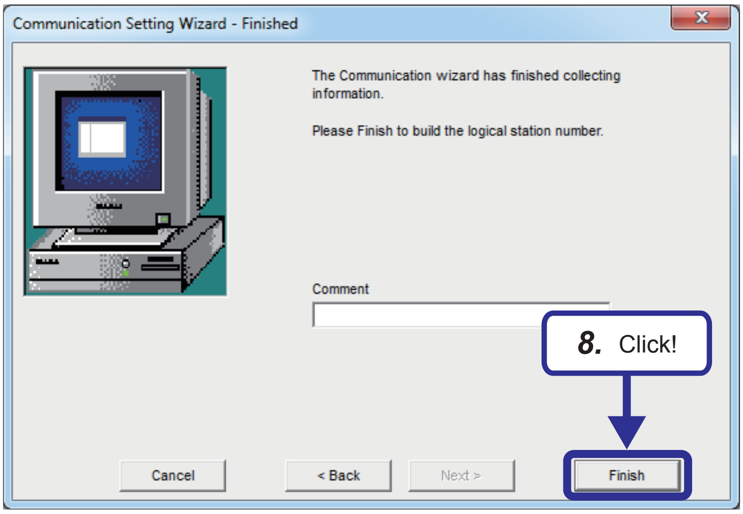
Multiple CPU: None

7. Set communications as follows and click the [Next] button.

[Parameters to be set]
Station type: Host station
CPU type: R08
Multiple CPU: None

(To the next page)

(From the previous page)

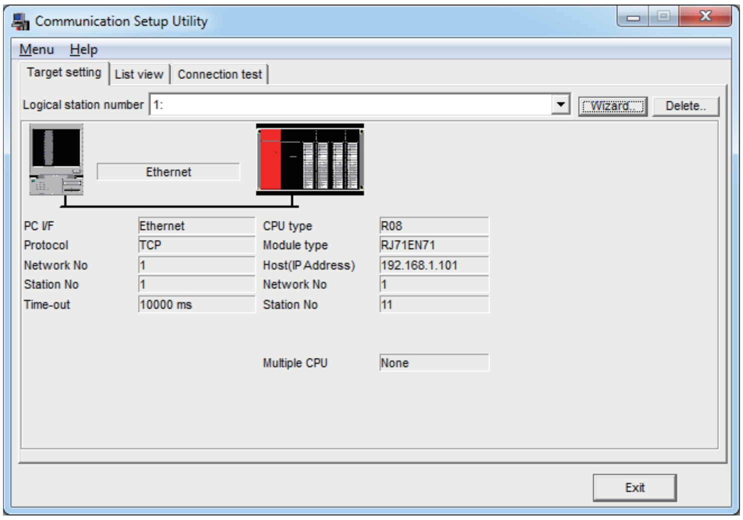

Communication Setting Wizard - Finished

The Communication wizard has finished collecting information.

Please Finish to build the logical station number.

Comment

8. "Comment" is not set in the exercise described in this textbook. Click the [Finish] button. (To add a comment to the logical station number, set it here.)



Logical station number 1:

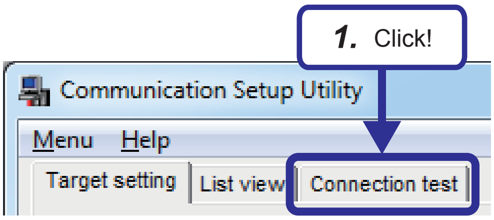
PC I/F	Ethernet	CPU type	R08
Protocol	TCP	Module type	RJ71EN71
Network No	1	Host(IP Address)	192.168.1.101
Station No	1	Network No	1
Time-out	10000 ms	Station No	11
		Multiple CPU	None

9. When the parameters have been set, the "Target setting" window appears. Check the settings.

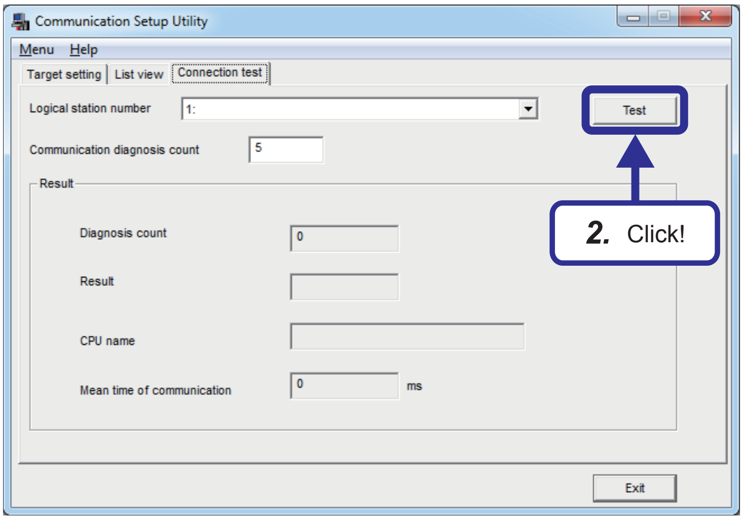
5.3.2 Performing the communication diagnostics

Perform the communication diagnostics to check whether the logical station number is set correctly.

Operating procedure



1. Click the [Connection test] tab.



Logical station number: 1:

Communication diagnosis count: 5

Diagnosis count: 0

Result:

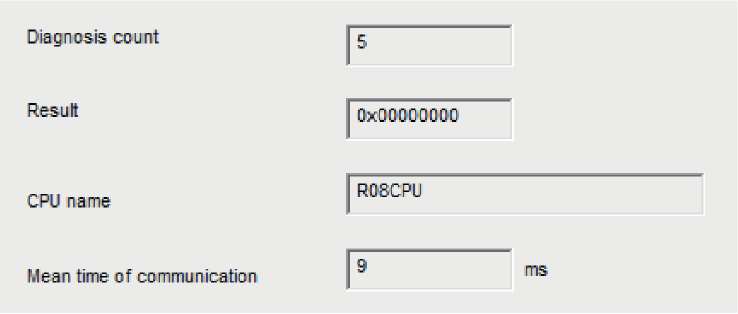
CPU name:

Mean time of communication: 0 ms

2. The "Connection test" window appears.
Specify a test-target logical station number.
(Specify 1 in the exercise described in this textbook.)
Click the [Test] button to perform the connection test.

(To the next page)

(From the previous page)



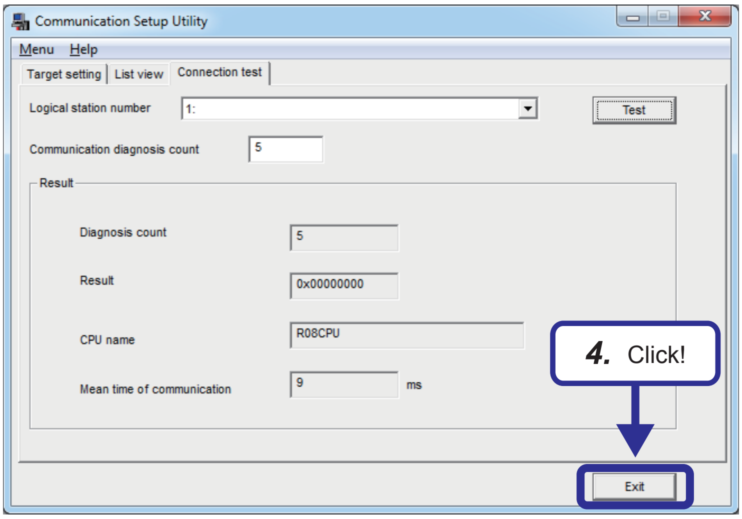
Diagnosis count 5

Result 0x00000000

CPU name R08CPU

Mean time of communication 9 ms

3. When data is successfully communicated with the CPU module, "Communication test is successful." is displayed.
At the same time, "0x00000000" is displayed in "Result".
When an error occurs, check the displayed error code.
For details on error codes, refer to the following.
📖 MX Component Version 4 Programming Manual



Logical station number 1:

Communication diagnosis count 5

Result

Diagnosis count 5

Result 0x00000000

CPU name R08CPU

Mean time of communication 9 ms

4. Click the [Exit] button.

5.4 Visual Basic 2012 Program (Reading Values from the Device)

The following program reads the on/off states of the bit device areas X100 to X107 (8 points) and the values of the word device areas D100 to D104 (5 words) of the CPU module.

5.4.1 Visual Basic 2012 program

Solution file name	E71-1.sln

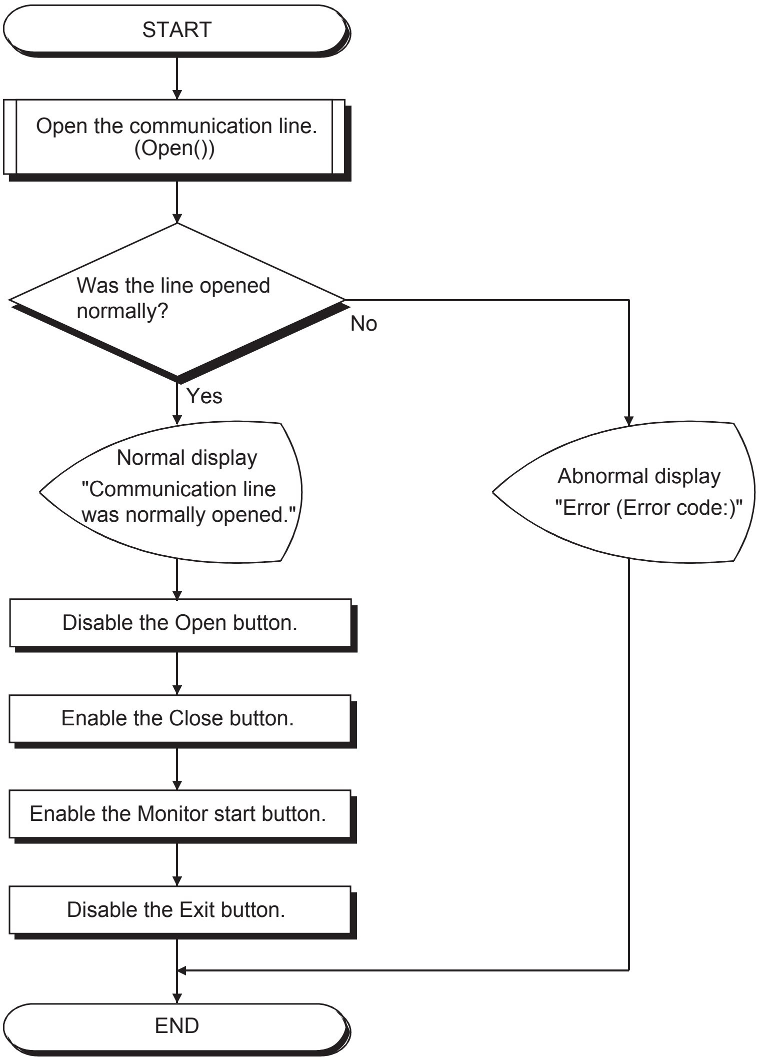
```
' E71-1.vb
Option Strict On              ' Limit the implicit data type conversion to "Expand" conversion.
Option Explicit On            ' Force the variable declaration.

Public Class frmE71_1
    ' Device monitor interval (Millisecond)
    Private Const MONITOR_INTERVAL As Integer = 100

    ' Sequence data of the monitor device name list
    Private DEVICE_NUMBERS As String() = { _
        "X100", "X101", "X102", "X103", "X104", "X105", "X106", "X107", "D100", "D101", "D102", "D103", "D104" _
    }

    Private Sub frmE71_1_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        timMONITOR.Interval = MONITOR_INTERVAL                                  ' Timer interval setting
    End Sub

    Private Sub btnOPEN_Click(sender As Object, e As EventArgs) Handles btnOPEN.Click
        Dim nRet As Integer                                                     ' Return value

        nRet = AxActUtlType1.Open                                               ' Open the communication line.
        If nRet = 0 Then
            txtSTATUS.Text = "Communication line was normally opened."          ' Message at normal open
            btnOPEN.Enabled = False                                             ' Disable the Open button.
            btnCLOSE.Enabled = True                                             ' Enable the Close button.
            btnMONITOR.Enabled = True                                           ' Enable the Monitor start button.
            btnEXIT.Enabled = False                                             ' Disable the Exit button.
        Else
            txtSTATUS.Text = "Error (Error code:" + nRet.ToString("X") + ")"    ' Message at abnormal open
        End If
    End Sub

    Private Sub btnCLOSE_Click(sender As Object, e As EventArgs) Handles btnCLOSE.Click
        Dim nRet As Integer                                                     ' Return value

        nRet = AxActUtlType1.Close                                              ' Close the communication line.
        If nRet = 0 Then
            txtSTATUS.Text = "Communication line was normally closed."          ' Message at normal close
            btnOPEN.Enabled = True                                              ' Enable the Open button.
            btnCLOSE.Enabled = False                                            ' Disable the Close button.
            btnMONITOR.Enabled = False                                          ' Disable the Monitor start button.
            btnEXIT.Enabled = True                                              ' Enable the Exit button.
        Else
            txtSTATUS.Text = "Error (Error code:" + nRet.ToString("X") + ")"    ' Message at abnormal close
        End If
    End Sub
```

```
Private Sub btnMONITOR_Click(sender As Object, e As EventArgs) Handles btnMONITOR.Click
    If Not timMONITOR.Enabled Then
        timMONITOR.Enabled = True
        txtSTATUS.Text = "Monitoring"                                     ' Display the monitoring message.
        btnMONITOR.Text = "Monitor stop"                                  ' Change the button display.
        btnCLOSE.Enabled = False                                          ' Disable the Close button.
    Else
        timMONITOR.Enabled = False
        txtSTATUS.Text = "Monitoring was stopped."                        ' Display the monitoring stop message.
        btnMONITOR.Text = "Monitor start"                                 ' Change the button display.
        btnCLOSE.Enabled = True                                           ' Enable the Close button.
    End If
End Sub

Private Sub btnEXIT_Click(sender As Object, e As EventArgs) Handles btnEXIT.Click
    Close()
End Sub

Private Sub timMONITOR_Tick(sender As Object, e As EventArgs) Handles timMONITOR.Tick
    Dim ni As Integer                                                     ' Variable 1 for loop
    Dim nj As Integer                                                     ' Variable 2 for loop
    Dim sDeviceList As String = ""                                        ' For storing the monitor device
    Dim nData(12) As Integer                                              ' For storing the monitor device value
    Dim nRet As Integer                                                   ' Return value

    'Set the device to be monitored.
    For Each sDevice As String In DEVICE_NUMBERS
        sDeviceList = sDeviceList + sDevice + vbLf
    Next
    ' Execute the random read.
    nRet = AxActUtlType1.ReadDeviceRandom(sDeviceList, DEVICE_NUMBERS.GetUpperBound(0) + 1, nData(0))
    If (nRet <> 0) Then
        txtSTATUS.Text = "Error (Error code:" + nRet.ToString("X") + ")"  ' Display the error code when an error occurs.
    End If
    ' Display the bit device (X100 to X107).
    For ni = 0 To 7
        If nData(ni) = 1 Then
            SetBitDevice(ni, "●")                                         ' Display ● when the bit device is on.
        Else
            SetBitDevice(ni, "○")                                         ' Display ○ when the bit device is off.
        End If
    Next ni
    ' Display the word device (D100 to D104).
    For nj = 0 To 4
        SetWordDevice(nj, nData(nj + 8))
    Next nj
End Sub
```

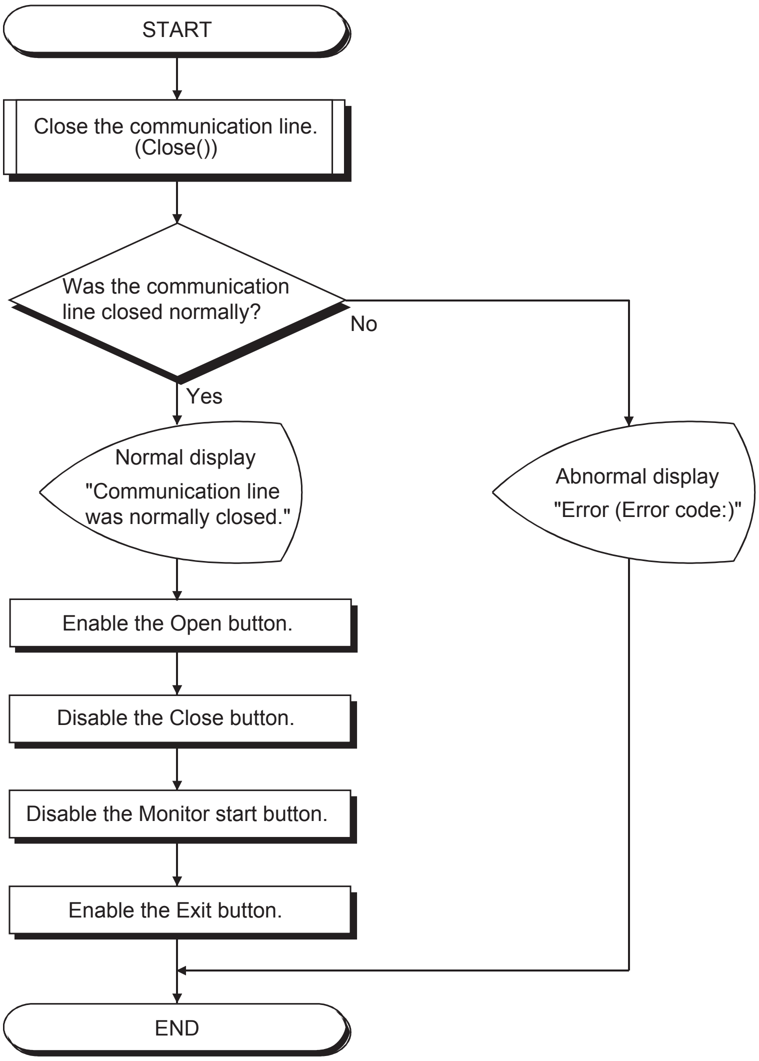
```
    ' A function for setting string values of the bit device (X100 to X107) by specifying the index
    Private Sub SetBitDevice(ByVal Aindex As Integer, ByVal AText As String)
        ' Set the string data type to each control corresponding to the index.
        Select Case Aindex
            Case 0
                lblVAL_X100.Text = AText
            Case 1
                lblVAL_X101.Text = AText
            Case 2
                lblVAL_X102.Text = AText
            Case 3
                lblVAL_X103.Text = AText
            Case 4
                lblVAL_X104.Text = AText
            Case 5
                lblVAL_X105.Text = AText
            Case 6
                lblVAL_X106.Text = AText
            Case 7
                lblVAL_X107.Text = AText
        End Select
    End Sub

    ' A function for setting integer values of the word device (D100 to D104) by specifying the index
    Private Sub SetWordDevice(ByVal AIndex As Integer, ByVal AValue As Integer)
        ' Set the string data type to each control corresponding to the index.
        Select Case AIndex
            Case 0
                txtVAL_D100.Text = AValue.ToString()
            Case 1
                txtVAL_D101.Text = AValue.ToString()
            Case 2
                txtVAL_D102.Text = AValue.ToString()
            Case 3
                txtVAL_D103.Text = AValue.ToString()
            Case 4
                txtVAL_D104.Text = AValue.ToString()
        End Select
    End Sub
End Class
```

Flow chart

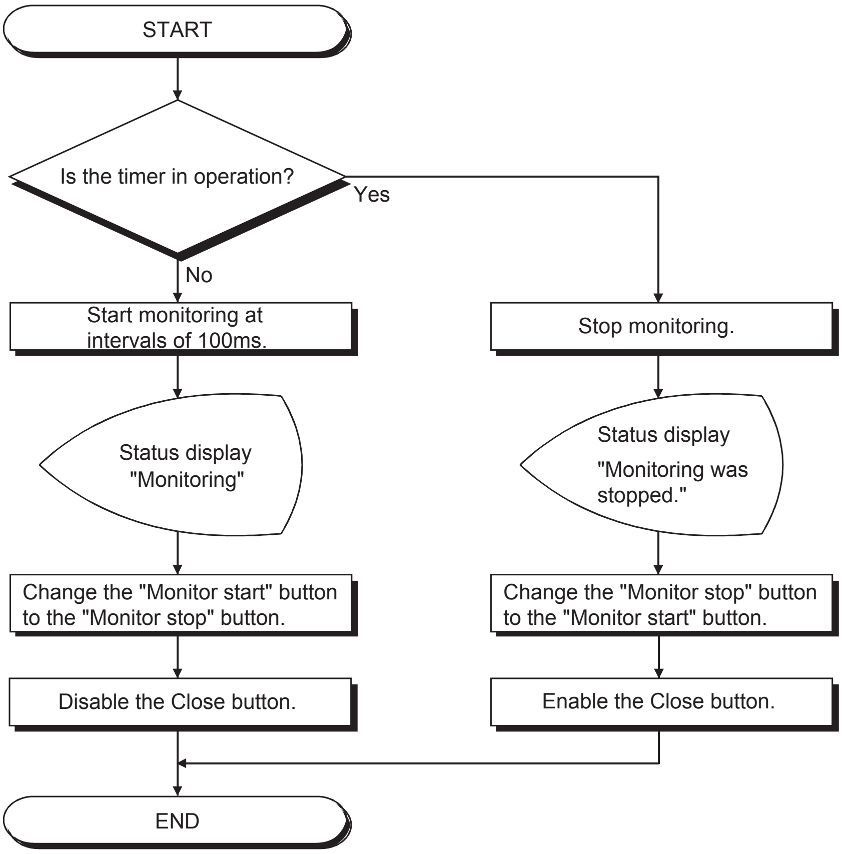
■ btnOPEN_Click() (Operation by clicking the Open button)



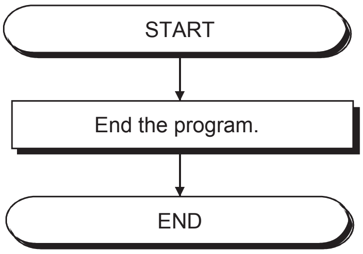
■ btnCLOSE_Click() (Operation by clicking the Close button)


START
Close the communication line.
(Close())
Was the communication line closed normally?
No
Yes
Normal display
"Communication line was normally closed."
Abnormal display
"Error (Error code:)"
Enable the Open button.
Disable the Close button.
Disable the Monitor start button.
Enable the Exit button.
END

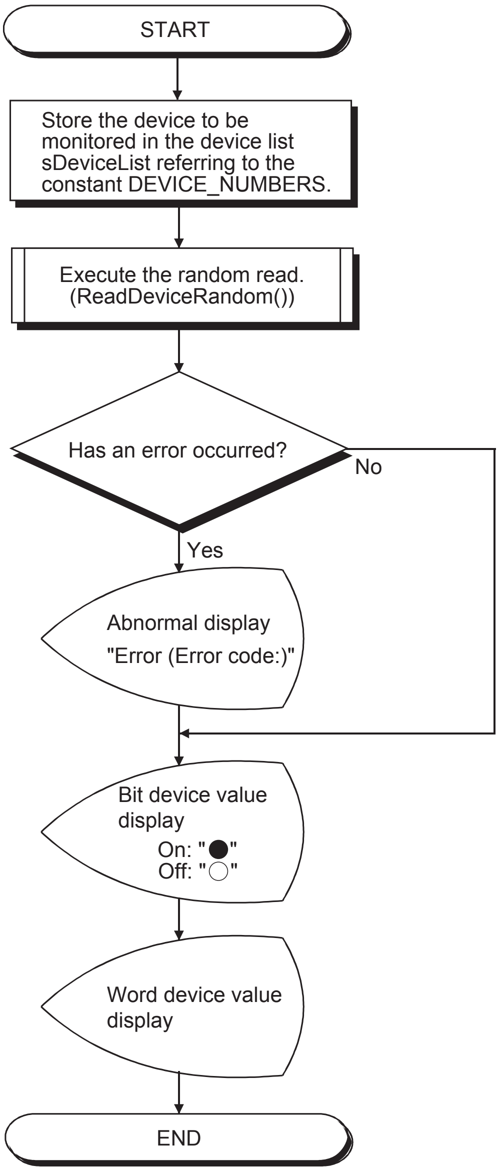
■btnMONITOR_Click() (Operation by clicking the Monitor start button)


START
Is the timer in operation?
Yes
No
Start monitoring at intervals of 100ms.
Stop monitoring.
Status display "Monitoring"
Status display "Monitoring was stopped."
Change the "Monitor start" button to the "Monitor stop" button.
Change the "Monitor stop" button to the "Monitor start" button.
Disable the Close button.
Enable the Close button.
END

■ btnEXIT_Click() (Operation by clicking the Exit button)



■timMONITOR_Tick() (Operation during monitoring)

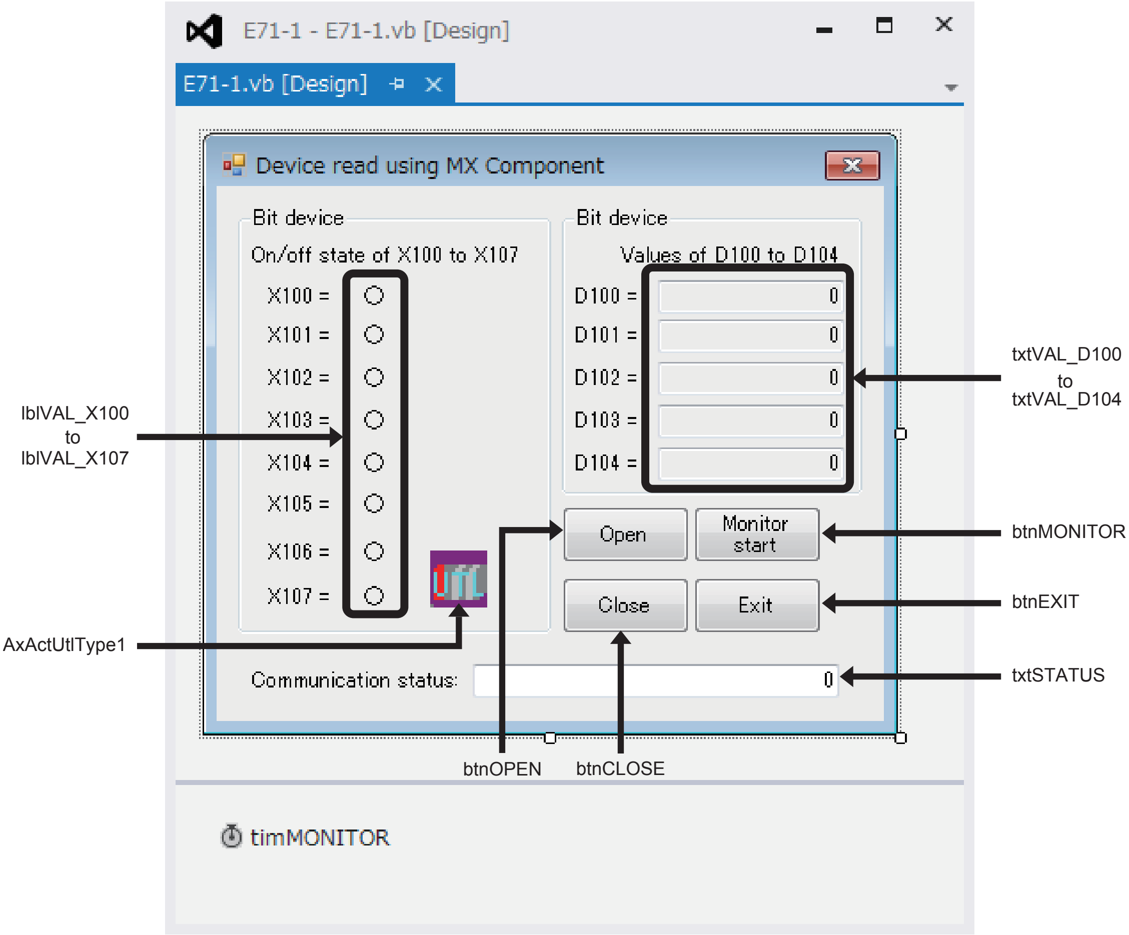

START
Store the device to be monitored in the device list sDeviceList referring to the constant DEVICE_NUMBERS.
Execute the random read. (ReadDeviceRandom())
Has an error occurred?
No
Yes
Abnormal display
"Error (Error code:)"
Bit device value display
On: "●"
Off: "○"
Word device value display
END

5.4.2 Operation of the demonstration machine

Execute the program to monitor the devices of the CPU module.

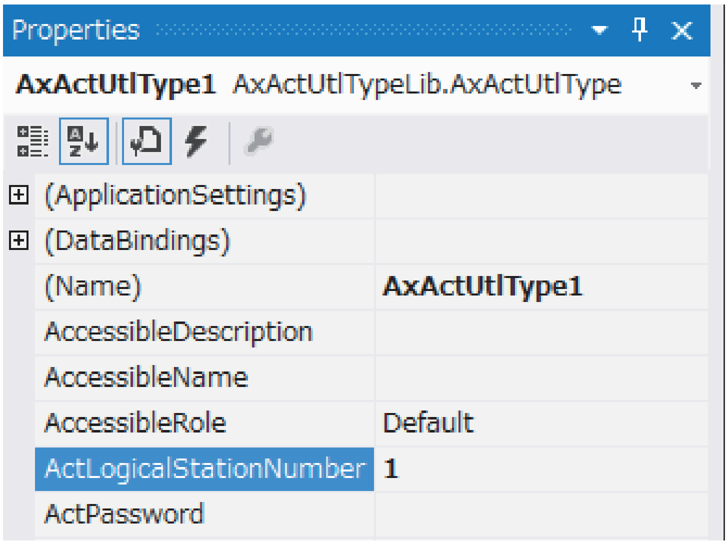
Program window

The following figure shows the object names in the program window.



Checking the device status of X100 to X107

Operating procedure



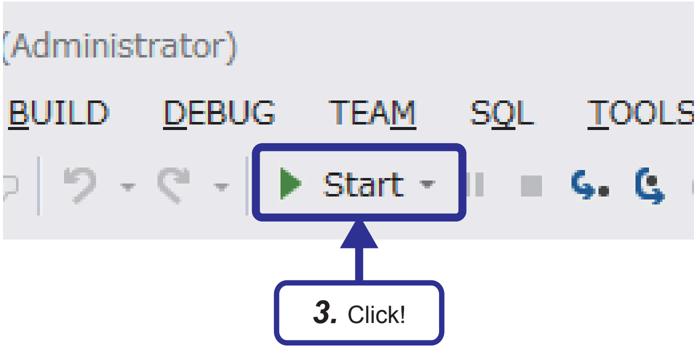
Properties

AxActUtlType1 AxActUtlTypeLib.AxActUtlType

⊞ (ApplicationSettings)	
⊞ (DataBindings)	
(Name)	AxActUtlType1
AccessibleDescription	
AccessibleName	
AccessibleRole	Default
ActLogicalStationNumber	1
ActPassword	

1. Start Visual Basic® 2012 and load the solution file "E71-1.sln".

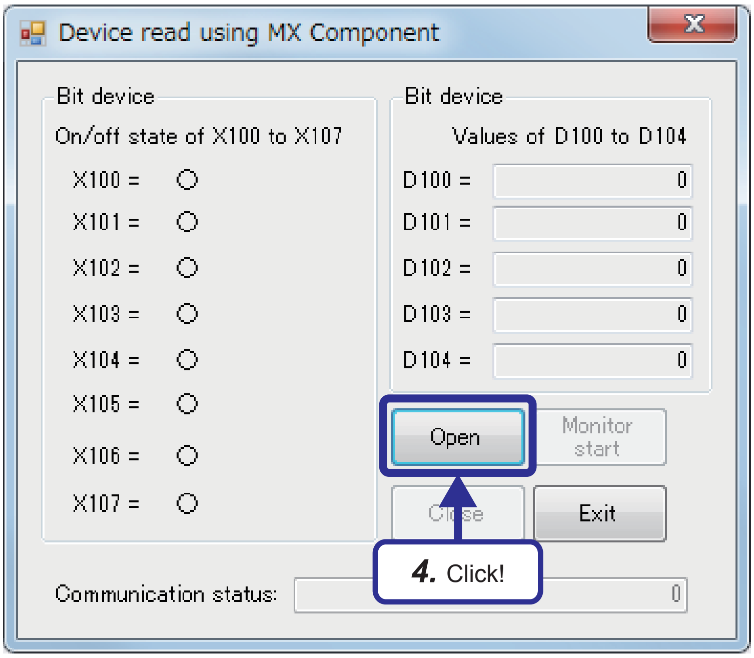
2. Check that ActLogicalStationNumber in the properties of AxActUtlType1 is set to "1".
This is the logical station number set on Page 5 - 23 Setting the logical station number.
Using the logical station number allows communications with the path set in the logical station number.



3. Click the [Start] button to execute the program.

(To the next page)

(From the previous page)


Device read using MX Component

Bit device

On/off state of X100 to X107

X100 = ○
X101 = ○
X102 = ○
X103 = ○
X104 = ○
X105 = ○
X106 = ○
X107 = ○

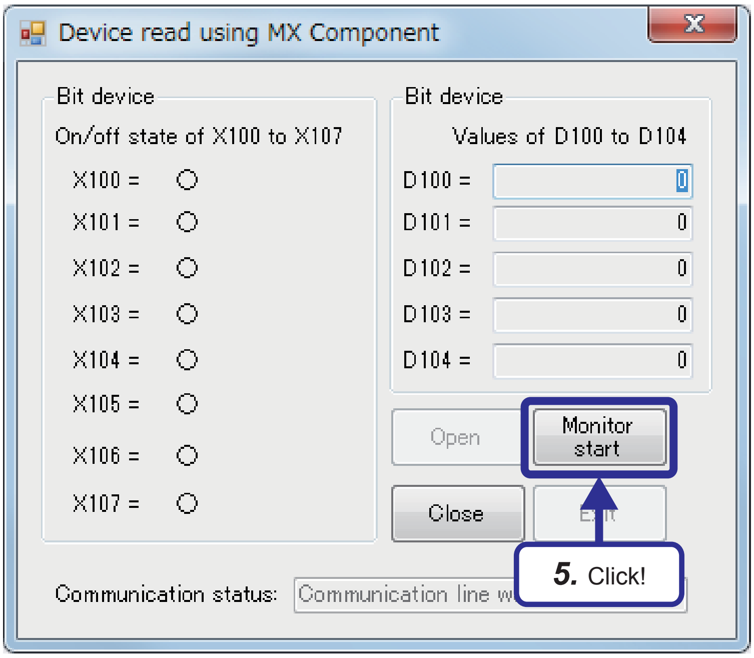
Bit device

Values of D100 to D104

D100 = 0
D101 = 0
D102 = 0
D103 = 0
D104 = 0

Communication status: 0

4. Click the [Open] button to open the communication line.
When the communication line is normally opened, "Communication line was normally opened." is displayed in "Communication status".
When an error occurs, the error code is displayed.
For details on error codes, refer to the following.
📖 MX Component Version 4 Programming Manual


Device read using MX Component

Bit device

On/off state of X100 to X107

X100 = ○
X101 = ○
X102 = ○
X103 = ○
X104 = ○
X105 = ○
X106 = ○
X107 = ○

Bit device

Values of D100 to D104

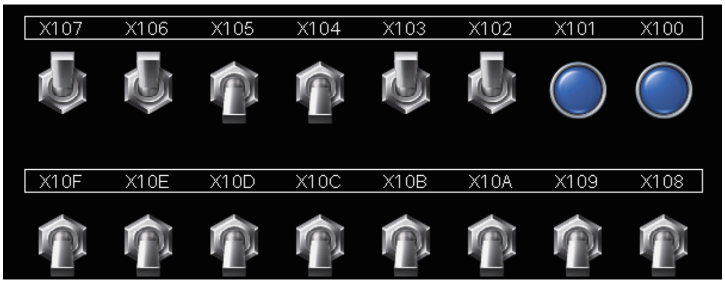
D100 = 0
D101 = 0
D102 = 0
D103 = 0
D104 = 0

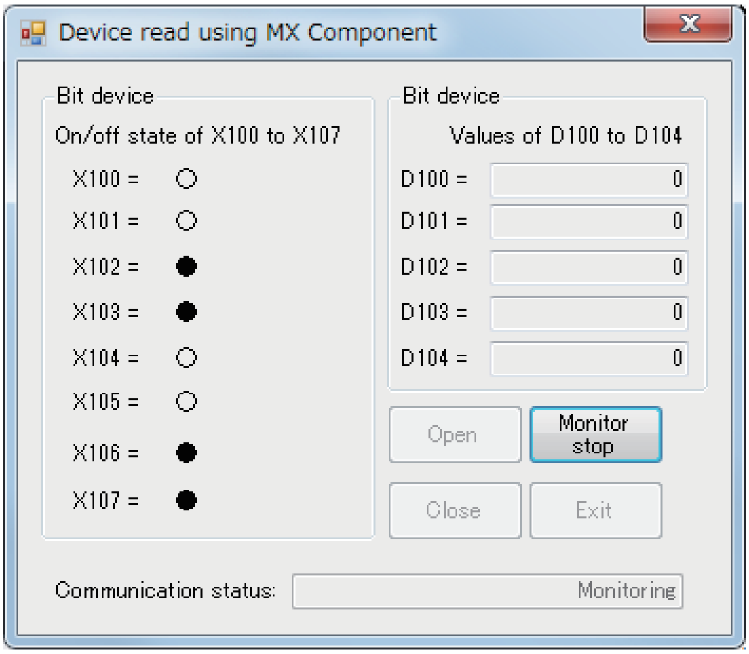
Communication status: Communication line w

5. When the communication line is normally opened, the [Monitor start] button becomes active. Click it to start monitoring.

(To the next page)

(From the previous page)




Device read using MX Component

Bit device

On/off state of X100 to X107

X100 = ○
X101 = ○
X102 = ●
X103 = ●
X104 = ○
X105 = ○
X106 = ●
X107 = ●

Bit device

Values of D100 to D104

D100 =	0
D101 =	0
D102 =	0
D103 =	0
D104 =	0

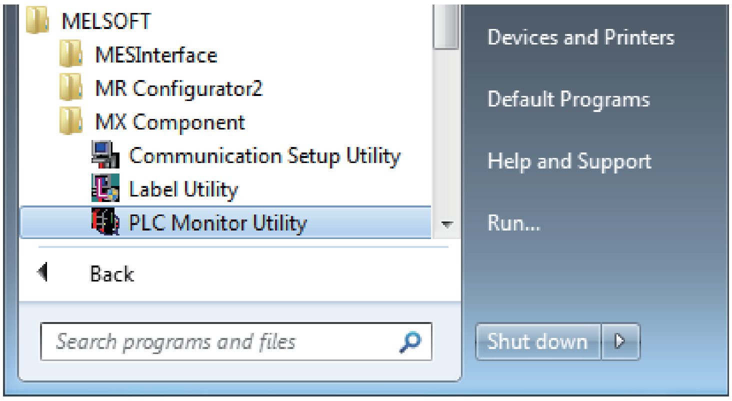
Communication status: Monitoring

6. Check that the status of the bit device areas in the program is changed by turning on/off the switches of X100 to X107 on the GOT.
After that, check the values in the word device areas (D100 to D104).

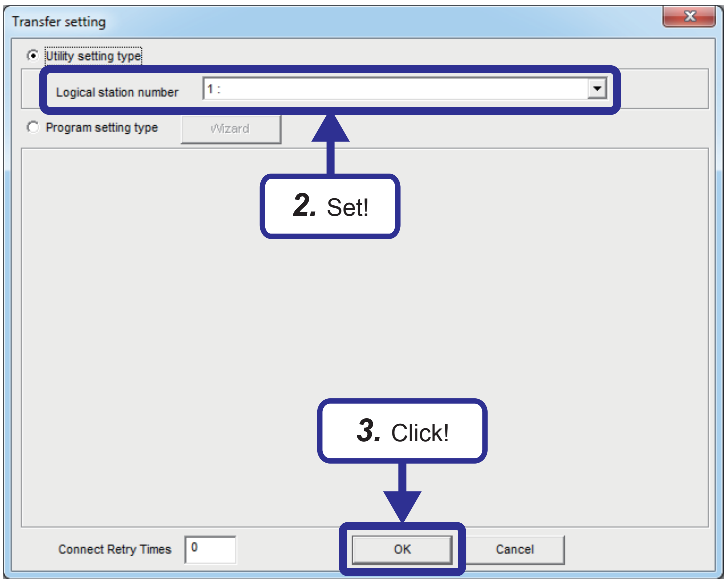
Checking the device status of D100 to D104

Check the values changed with MX Component in the program.

Operating procedure



1. Click [All Programs] ⇨ [MELSOFT] ⇨ [MX Component] ⇨ [PLC Monitor Utility] from the Windows® Start menu.

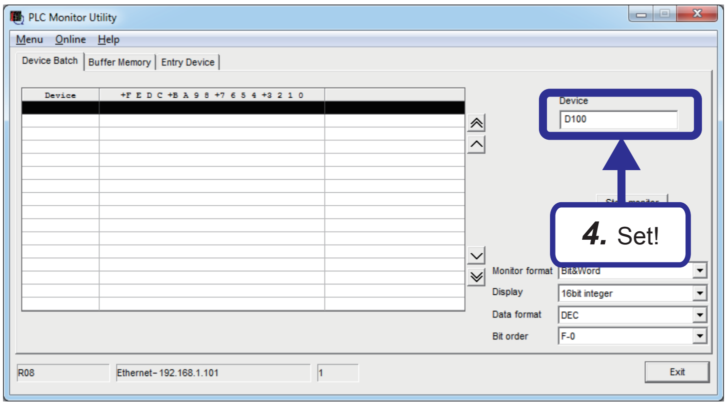


2. The "Transfer setting" window appears. Set "Logical station number" to "1".

3. Click the [OK] button.

(To the next page)

(From the previous page)



Device: D100

Monitor format: Bit&Word

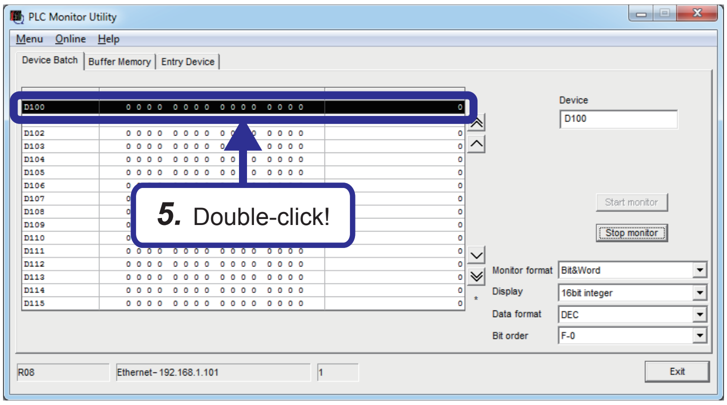
Display: 16bit integer

Data format: DEC

Bit order: F-0

R08 | Ethernet- 192.168.1.101 | 1

4. The PLC monitor utility starts up. Set "Device" to "D100" and click the [Start monitor] button.



Device: D100

Monitor format: Bit&Word

Display: 16bit integer

Data format: DEC

Bit order: F-0

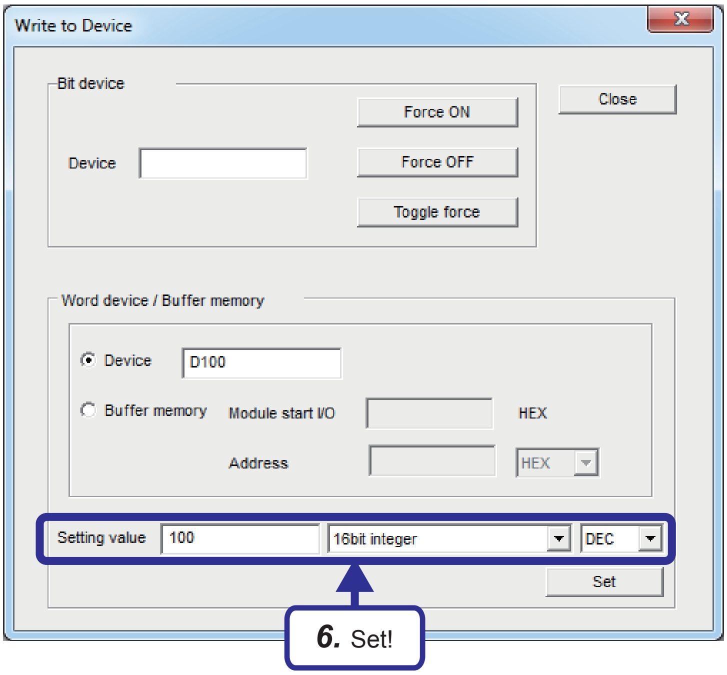
R08 | Ethernet- 192.168.1.101 | 1

5. The values of the device areas are displayed. Double-click a device area to change the value.
When a value of D100 to D104 is changed*1, the display of the corresponding device area changes in the program.

*1 Input a value in the format specified in the window.

(To the next page)

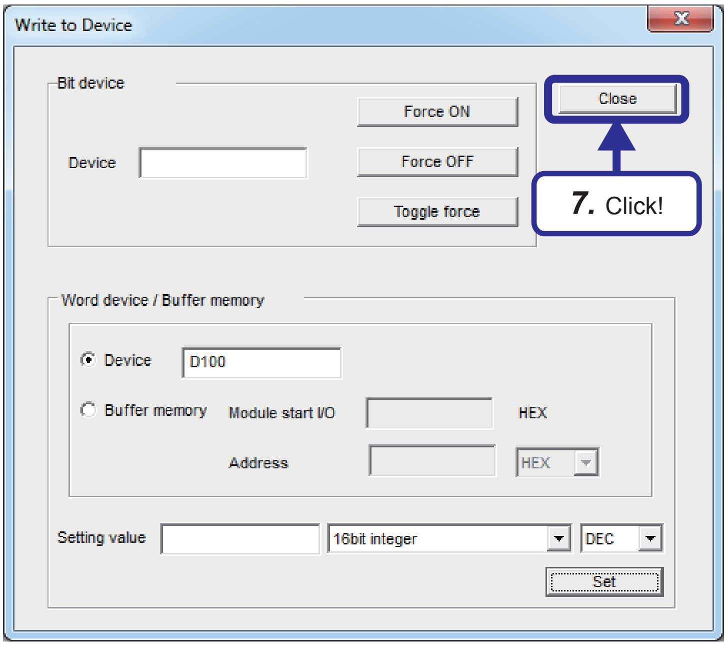
(From the previous page)



Device ● D100

Setting value 100 16bit integer DEC

6. The "Write to Device" window appears. Input a numerical value in "Setting value" and click the [Set] button.
Set 100 in D100 here.



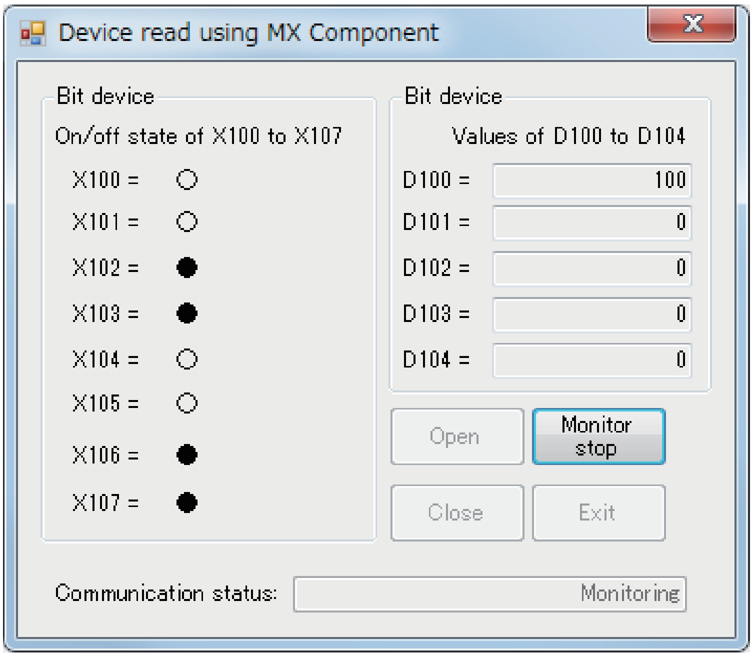
Device ● D100

Setting value 16bit integer DEC

7. Click the [Close] button in the "Write to Device" window to close the window.

(To the next page)

(From the previous page)



Device read using MX Component

Bit device

On/off state of X100 to X107

X100 = ○

X101 = ○

X102 = ●

X103 = ●

X104 = ○

X105 = ○

X106 = ●

X107 = ●

Bit device

Values of D100 to D104

D100 = 100

D101 = 0

D102 = 0

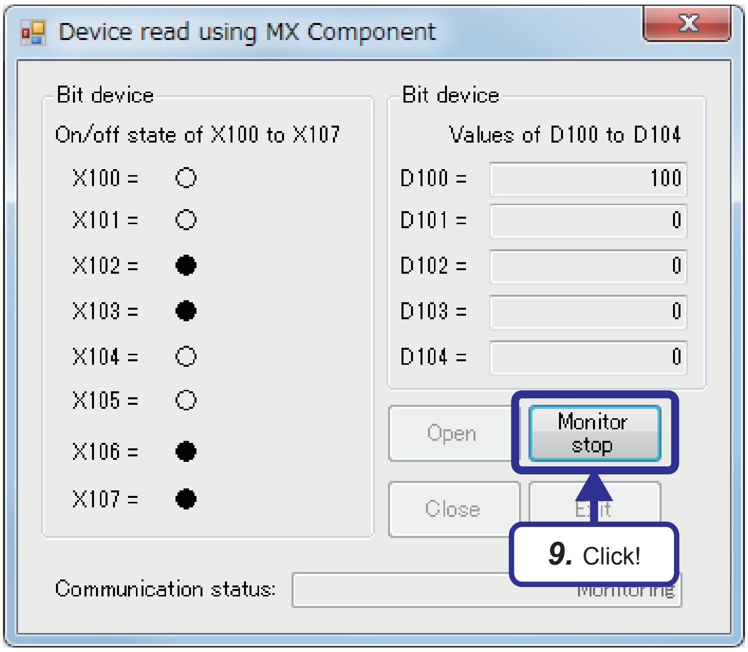
D103 = 0

D104 = 0

Communication status: Monitoring

8. Activate the program[*1] and check that the value of the word device has been changed. When steps 4 and 5 are repeated to change values of D100 to D104, the values of the corresponding device areas also change in the program.

[*1] Bring the window to the front to activate.



Device read using MX Component

Bit device

On/off state of X100 to X107

X100 = ○

X101 = ○

X102 = ●

X103 = ●

X104 = ○

X105 = ○

X106 = ●

X107 = ●

Bit device

Values of D100 to D104

D100 = 100

D101 = 0

D102 = 0

D103 = 0

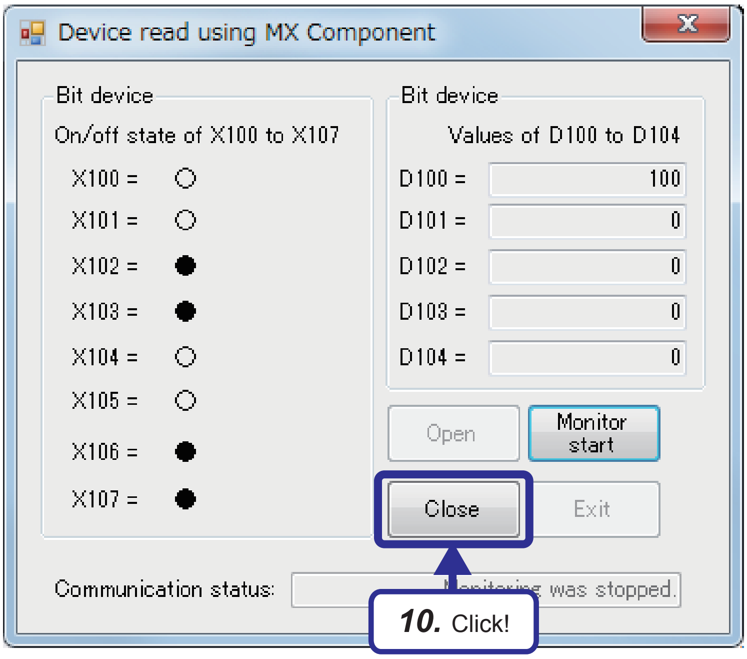
D104 = 0

Communication status: Monitoring

9. To stop monitoring, click the [Monitor stop] button.

(To the next page)

(From the previous page)



Device read using MX Component

Bit device
On/off state of X100 to X107

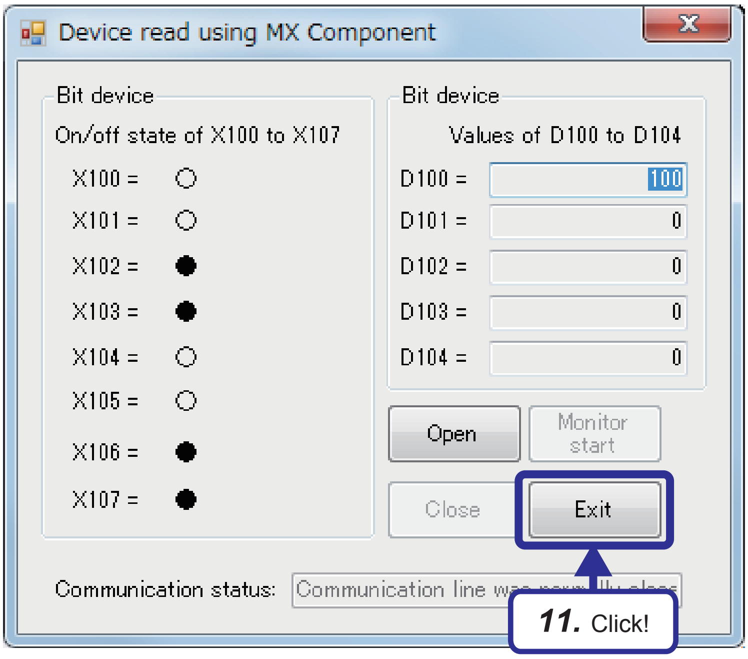
X100 =	○
X101 =	○
X102 =	●
X103 =	●
X104 =	○
X105 =	○
X106 =	●
X107 =	●

Bit device
Values of D100 to D104

D100 =	100
D101 =	0
D102 =	0
D103 =	0
D104 =	0

Communication status: Monitoring was stopped.

10. Click the [Close] button to close the communication line.
When the communication line is normally closed, "Communication line was normally closed." is displayed in "Communication status".
When an error occurs, the error code is displayed.
For details on error codes, refer to the following.
📖 MX Component Version 4 Programming Manual



Device read using MX Component

Bit device
On/off state of X100 to X107

X100 =	○
X101 =	○
X102 =	●
X103 =	●
X104 =	○
X105 =	○
X106 =	●
X107 =	●

Bit device
Values of D100 to D104

D100 =	100
D101 =	0
D102 =	0
D103 =	0
D104 =	0

Communication status: Communication line was normally closed.

11. Click the [Exit] button to exit the program.
The next solution file can be loaded.

5.5 Visual Basic 2012 Program (Writing Values in the Device)

The following program changes on/off states of the bit device areas Y170 to Y177 (8 points) and the values of the word device areas D100 to D104 (5 words) of the CPU module.

5.5.1 Visual Basic 2012 program

Solution file name	E71-2.sln

```
' E71-2.vb
Option Strict On                ' Limit the implicit data type conversion to "Expand" conversion.
Option Explicit On              ' Force the variable declaration.

Public Class frmE71_2
    ' Sequence data of the monitor device name list
    Private DEVICE_NUMBERS As String() = { _
        "Y170", "Y171", "Y172", "Y173", "Y174", "Y175", "Y176", "Y177", "D100", "D101", "D102", "D103", "D104" _
    }

    Private Sub frmE71_2_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        ' Add double-click events of 8 bit devices to a procedure as events.
        AddHandler lblVAL_Y170.DoubleClick, AddressOf lblBITVALUES_DoubleClick
        AddHandler lblVAL_Y171.DoubleClick, AddressOf lblBITVALUES_DoubleClick
        AddHandler lblVAL_Y172.DoubleClick, AddressOf lblBITVALUES_DoubleClick
        AddHandler lblVAL_Y173.DoubleClick, AddressOf lblBITVALUES_DoubleClick
        AddHandler lblVAL_Y174.DoubleClick, AddressOf lblBITVALUES_DoubleClick
        AddHandler lblVAL_Y175.DoubleClick, AddressOf lblBITVALUES_DoubleClick
        AddHandler lblVAL_Y176.DoubleClick, AddressOf lblBITVALUES_DoubleClick
        AddHandler lblVAL_Y177.DoubleClick, AddressOf lblBITVALUES_DoubleClick
    End Sub

    Private Sub btnOPEN_Click(sender As Object, e As EventArgs) Handles btnOPEN.Click
        Dim nRet As Integer                                                 ' Return value

        nRet = AxActUtlType1.Open                                           ' Open the communication line.
        If nRet = 0 Then
            txtSTATUS.Text = "Communication line was normally opened."      ' Message at normal open
            btnOPEN.Enabled = False                                         ' Disable the Open button.
            btnCLOSE.Enabled = True                                         ' Enable the Close button.
            btnWRITEDEVICE.Enabled = True                                   ' Enable the Write button.
            btnEXIT.Enabled = False                                         ' Disable the Exit button.
        Else
            txtSTATUS.Text = "Error (Error code:" + nRet.ToString("X") + ")"  ' Message at abnormal open
        End If
    End Sub

    Private Sub btnCLOSE_Click(sender As Object, e As EventArgs) Handles btnCLOSE.Click
        Dim nRet As Integer                                                 ' Return value

        nRet = AxActUtlType1.Close                                          ' Close the communication line.
        If nRet = 0 Then
            txtSTATUS.Text = "Communication line was normally closed."      ' Message at normal close
            btnOPEN.Enabled = True                                          ' Enable the Open button.
            btnCLOSE.Enabled = False                                        ' Disable the Close button.
            btnWRITEDEVICE.Enabled = False                                  ' Disable the Write button.
            btnEXIT.Enabled = True                                          ' Enable the Exit button.
        Else
            txtSTATUS.Text = "Error (Error code:" + nRet.ToString("X") + ")"  ' Message at abnormal close
        End If
    End Sub
```

```
Private Sub btnWRITEDEVICE_Click(sender As Object, e As EventArgs) Handles btnWRITEDEVICE.Click
    Dim ni As Short                                                          ' Variable for loop
    Dim nj As Short                                                          ' Variable 2 for loop
    Dim sDeviceList As String = ""                                           ' For storing the monitor device
    Dim nData(12) As Integer                                                 ' For storing the monitor device value
    Dim nRet As Integer                                                      ' Return value

    ' Set the device to be monitored.
    For Each sDevice As String In DEVICE_NUMBERS
        sDeviceList = sDeviceList + sDevice + vbLf
    Next
    ' Set values of the bit device (Y170 to Y177).
    For ni = 0 To 7
        If GetBitDevice(ni) = "●" Then
            nData(ni) = 1                                                    ' To set ● , store "1".
        Else
            nData(ni) = 0                                                    ' To set ○ , store  "0".
        End If
    Next
    ' Set values of the word device (D100 to D104).
    For nj = 0 To 4
        nData(nj + 8) = GetWordDevice(nj)
    Next
    ' Execute the random write.
    nRet = AxActUtlType1.WriteDeviceRandom(sDeviceList, DEVICE_NUMBERS.GetUpperBound(0) + 1, nData(0))
    If (nRet = 0) Then
        txtSTATUS.Text = "The value was normally written."                   ' Message at normal write
    Else
        txtSTATUS.Text = "Error (Error code:" + nRet.ToString("X") + ")"     ' Display the error code when an error occurs.
    End If
End Sub

Private Sub btnEXIT_Click(sender As Object, e As EventArgs) Handles btnEXIT.Click
    Close()
End Sub

Private Sub lblBITVALUES_DoubleClick(ByVal sender As Object, ByVal e As System.EventArgs)
    Dim objDevLabel As Label = CType(sender, Label)
    ' Double-click● and○ to reverse their display.
    If objDevLabel.Text = "●" Then
        objDevLabel.Text = "○"
    Else
        objDevLabel.Text = "●"
    End If
End Sub

' A function for getting string values of the bit device (Y170 to Y177) by specifying the index
Private Function GetBitDevice(ByVal Aindex As Integer) As String
    Dim sBit As String = ""
    ' Get information of each control corresponding to the index.
    Select Case Aindex
        Case 0
            sBit = lblVAL_Y170.Text
        Case 1
            sBit = lblVAL_Y171.Text
        Case 2
            sBit = lblVAL_Y172.Text
        Case 3
            sBit = lblVAL_Y173.Text
        Case 4
            sBit = lblVAL_Y174.Text
        Case 5
            sBit = lblVAL_Y175.Text
        Case 6
            sBit = lblVAL_Y176.Text
        Case 7
            sBit = lblVAL_Y177.Text
    End Select
    Return sBit
End Function
```

```
    ' A function for getting integer values of the word device (D100 to D104) by specifying the index
    Private Function getWordDevice(ByVal AIndex As Integer) As Integer
        Dim sValue As String = "0"
        Dim nValue As Integer
        ' Get information of each control corresponding to the index.
        Select Case AIndex
            Case 0
                sValue = txtVAL_D100.Text
            Case 1
                sValue = txtVAL_D101.Text
            Case 2
                sValue = txtVAL_D102.Text
            Case 3
                sValue = txtVAL_D103.Text
            Case 4
                sValue = txtVAL_D104.Text
        End Select
        ' Convert decimal string type data into numerical value type data (Exception handling).
        Try
            nValue = System.Convert.ToInt32(sValue.Trim())
        Catch ex As Exception
            nValue = 0
        End Try
        Return nValue
    End Function
End Class
```

Flow chart

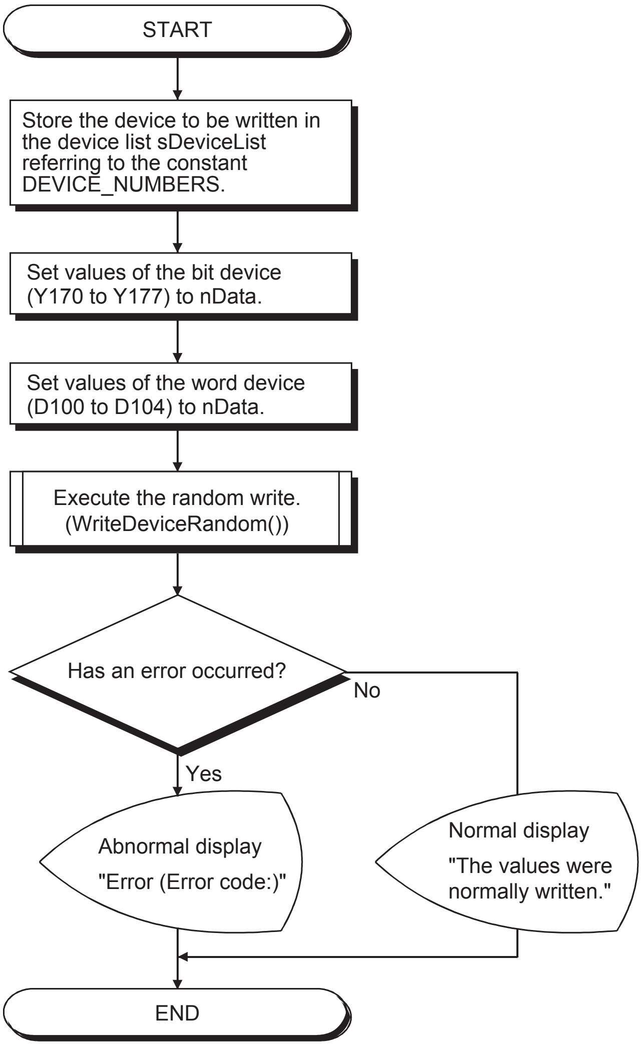
■btnOPEN_Click() (Operation by clicking the Open button)

Refer to Page 33 btnOPEN_Click() (Operation by clicking the Open button) because the flow chart is the same.

■btnCLOSE_Click() (Operation by clicking the Close button)

Refer to Page 34 btnCLOSE_Click() (Operation by clicking the Close button) because the flow chart is the same.

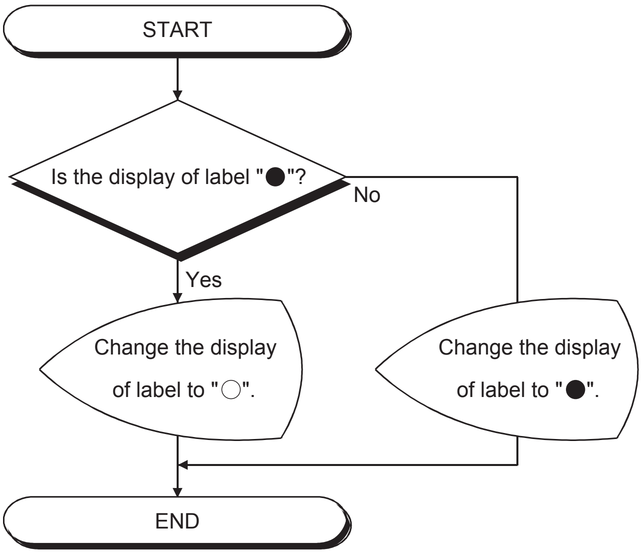
■btnMONITOR_Click() (Operation by clicking the Write button)


START
Store the device to be written in the device list sDeviceList referring to the constant DEVICE_NUMBERS.
Set values of the bit device (Y170 to Y177) to nData.
Set values of the word device (D100 to D104) to nData.
Execute the random write. (WriteDeviceRandom())
Has an error occurred?
No
Yes
Abnormal display
"Error (Error code:)"
Normal display
"The values were normally written."
END

■btnEXIT_Click() (Operation by clicking the Exit button)

Refer to Page 35 btnEXIT_Click() (Operation by clicking the Exit button) because the flow chart is the same.

■lblBITVALUES_DoubleClick() (Operation by double-clicking a label)

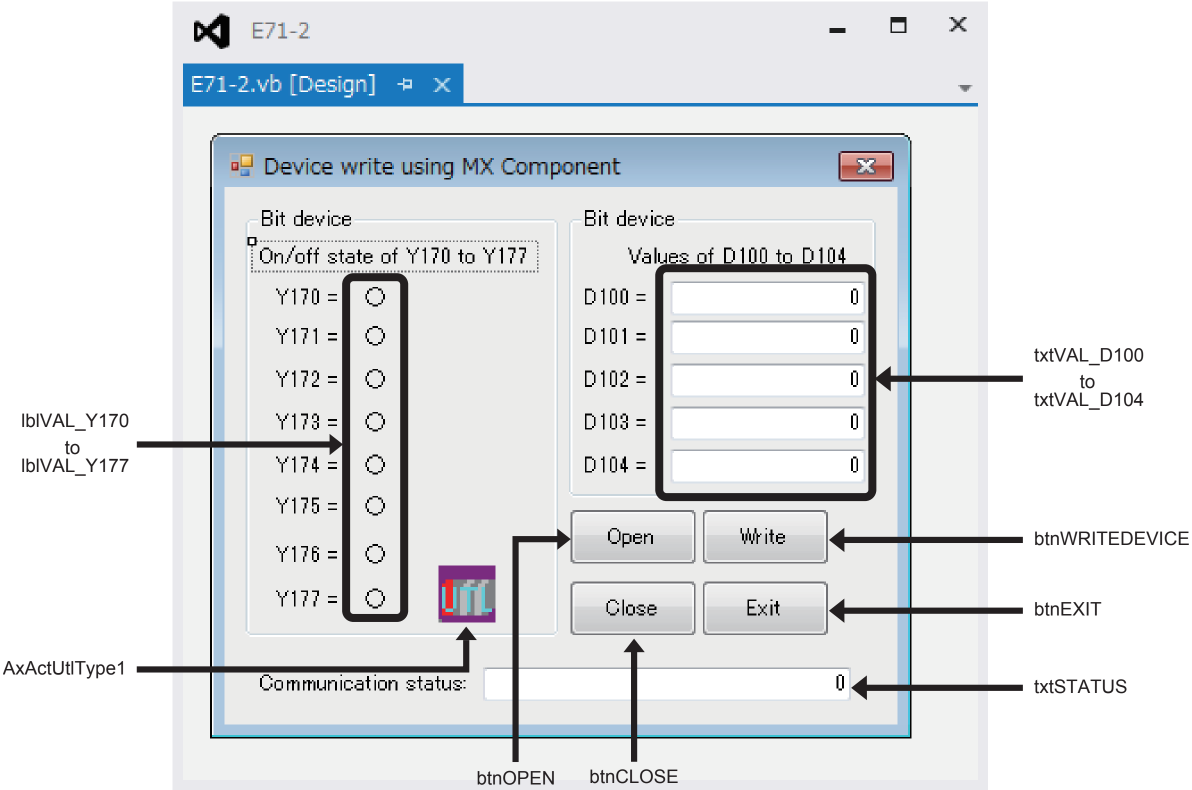

START
Is the display of label "●"?
No
Yes
Change the display of label to "○".
Change the display of label to "●".
END

5.5.2 Operation of the demonstration machine

Execute the program to change device values of the CPU module.

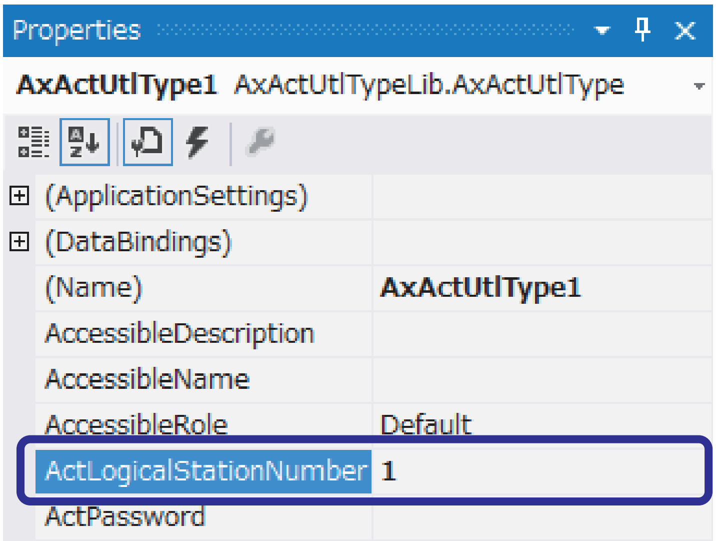
Program window

The following figure shows the object names in the program window.



Checking the device status of Y170 to Y177

Operating procedure

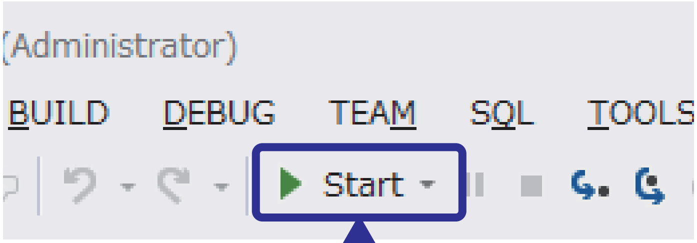


AxActUtlType1 AxActUtlTypeLib.AxActUtlType

⊞ (ApplicationSettings)	
⊞ (DataBindings)	
(Name)	AxActUtlType1
AccessibleDescription	
AccessibleName	
AccessibleRole	Default
ActLogicalStationNumber	1
ActPassword	

1. Start Visual Basic® 2012 and load the solution file "E71-2.sln".

2. Check that ActLogicalStationNumber in the properties of AxActUtlType1 is set to "1".
   This is the logical station number set on Page 5 - 23 Setting the logical station number.
   Using the logical station number allows communications with the path set in the logical station number.

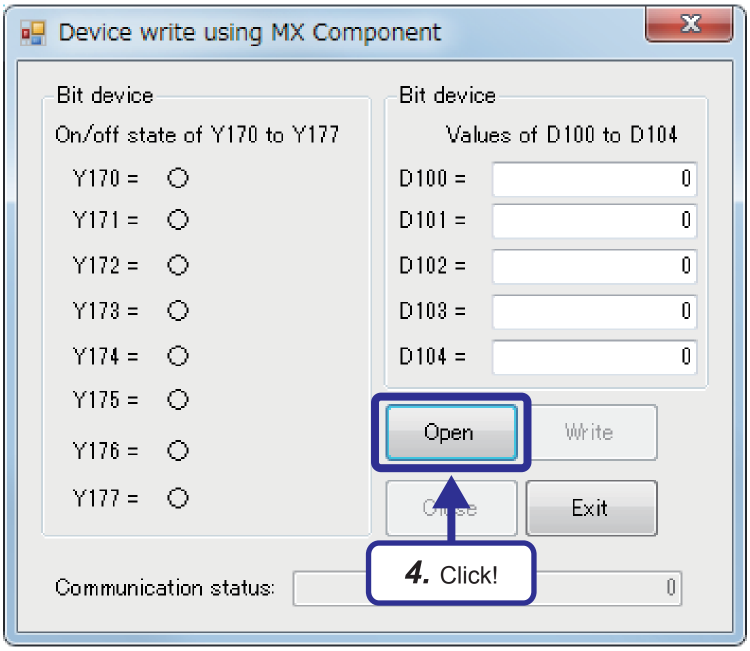


3. Click!

3. Click the [Start] button to execute the program.

(To the next page)

(From the previous page)



Device write using MX Component

Bit device

On/off state of Y170 to Y177

Y170 = ○
Y171 = ○
Y172 = ○
Y173 = ○
Y174 = ○
Y175 = ○
Y176 = ○
Y177 = ○

Bit device

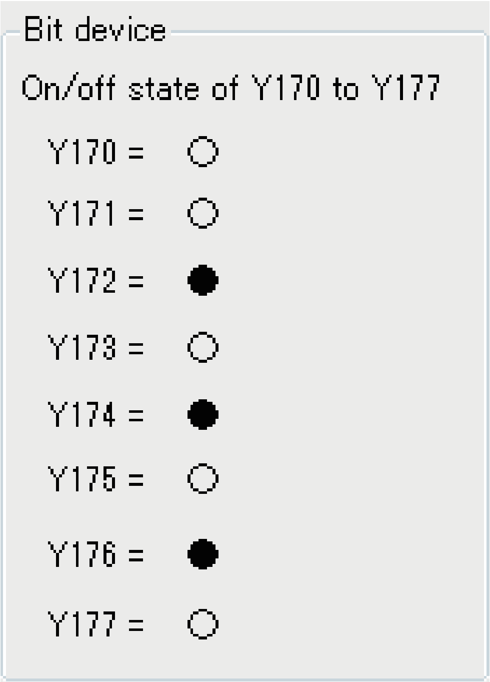
Values of D100 to D104

D100 = 0
D101 = 0
D102 = 0
D103 = 0
D104 = 0

Communication status: 0

4. Click!

4. Click the [Open] button to open the communication line.
When the communication line is normally opened, "Communication line was normally opened." is displayed in "Communication status".
When an error occurs, the error code is displayed. For details on error codes, refer to the following.
📖 MX Component Version 4 Programming Manual



Bit device

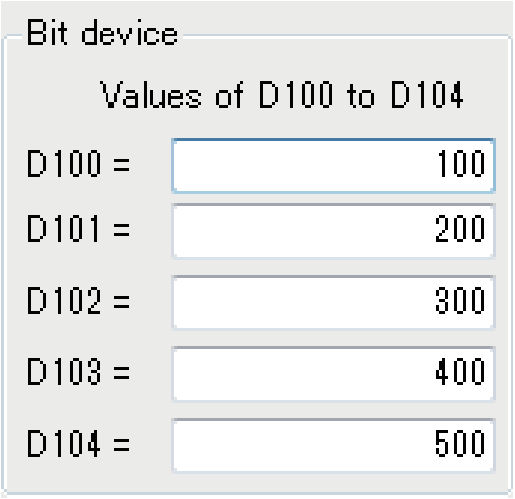
On/off state of Y170 to Y177

Y170 = ○
Y171 = ○
Y172 = ●
Y173 = ○
Y174 = ●
Y175 = ○
Y176 = ●
Y177 = ○

5. Double-click "○" of the change-target bit device to set the on/off state.
"○" is off and "●" is on.

(To the next page)

(From the previous page)


Bit device

Values of D100 to D104

D100 = 100

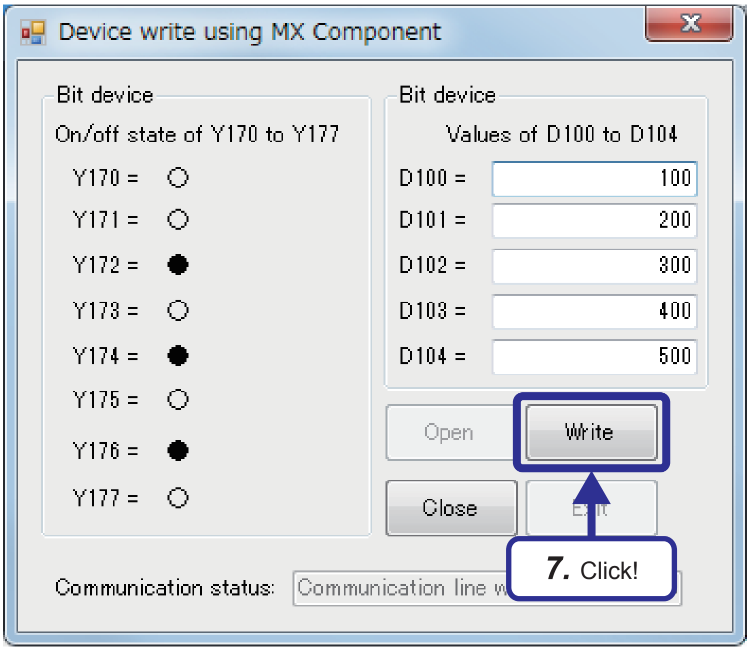
D101 = 200

D102 = 300

D103 = 400

D104 = 500

6. Input a numerical value in the text box of the change-target word device.
The input value is handled as a decimal number.


Device write using MX Component

Bit device

On/off state of Y170 to Y177

Y170 = ○

Y171 = ○

Y172 = ●

Y173 = ○

Y174 = ●

Y175 = ○

Y176 = ●

Y177 = ○

Bit device

Values of D100 to D104

D100 = 100

D101 = 200

D102 = 300

D103 = 400

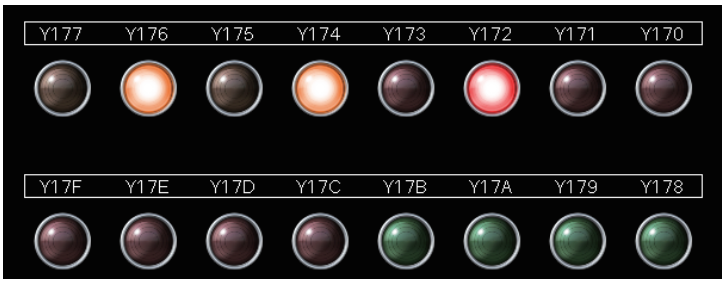
D104 = 500

Communication status: Communication line w

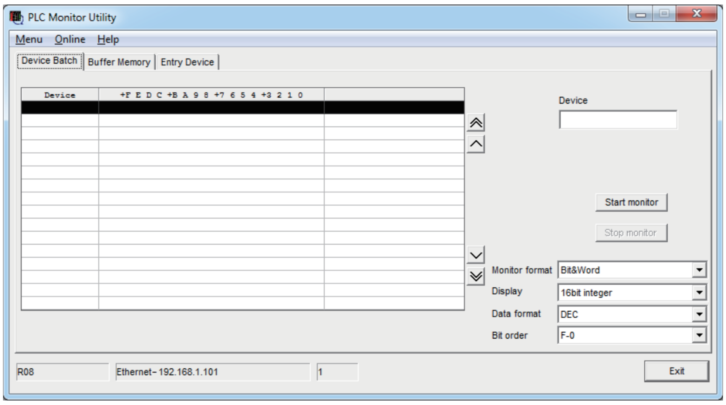
7. Click the [Write] button to write the values.

(To the next page)

(From the previous page)



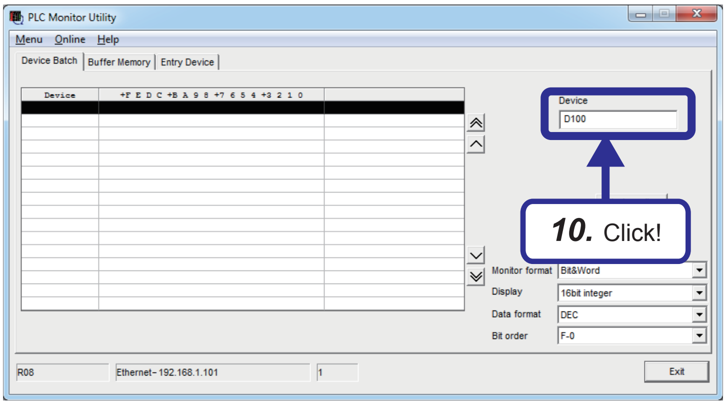
8. When the values are normally written, "The values were normally written." is displayed. Check the LEDs of Y170 to Y177 on the GOT. Check that the LEDs of the bit device areas that have turned on in the program are on.



Monitor format: Bit&Word
Display: 16bit integer
Data format: DEC
Bit order: F-0

R08 | Ethernet-192.168.1.101 | 1

9. Check the written values with MX Component. Start up the PLC monitor utility and set the logical station number to "1" referring to Page 5 - 41 Checking the device status of D100 to D104.
Click [All Programs] ⇨ [MELSOFT] ⇨ [MX Component] ⇨ [PLC Monitor Utility] from the Windows® Start menu.
Set "Logical station number" to "1" and click the [OK] button in the "Transfer setting" window.



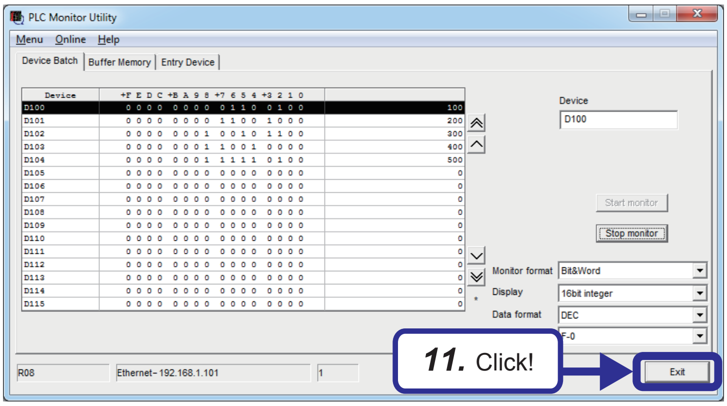
Device: D100
Monitor format: Bit&Word
Display: 16bit integer
Data format: DEC
Bit order: F-0

R08 | Ethernet-192.168.1.101 | 1

10. Set "Device" to "D100" and click the [Start monitor] button.

(To the next page)

(From the previous page)



Device	+F E D C +B A 9 8 +7 6 5 4 +3 2 1 0	
D100	0 0 0 0 0 0 0 0 0 1 1 0 0 1 0 0	100
D101	0 0 0 0 0 0 0 0 1 1 0 0 1 0 0 0	200
D102	0 0 0 0 0 0 0 1 0 0 1 0 1 1 0 0	300
D103	0 0 0 0 0 0 0 1 1 0 0 1 0 0 0 0	400
D104	0 0 0 0 0 0 0 1 1 1 1 1 0 1 0 0	500
D105	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0
D106	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0
D107	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0
D108	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0
D109	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0
D110	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0
D111	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0
D112	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0
D113	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0
D114	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0
D115	0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0	0

Device: D100

Monitor format: Bit&Word

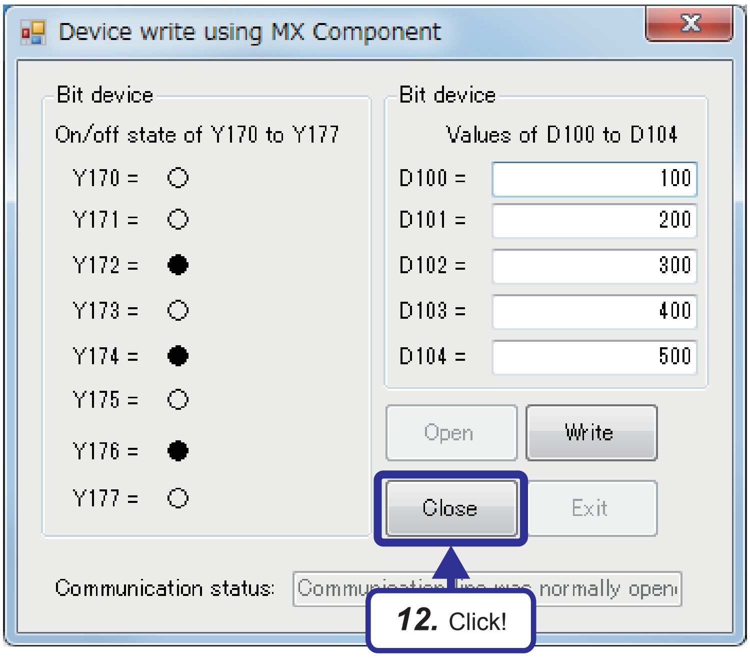
Display: 16bit integer

Data format: DEC

F-0

R08 | Ethernet- 192.168.1.101 | 1

11. Check that the written device values are displayed.
After checking the values, click the [Exit] button.



Bit device

On/off state of Y170 to Y177

Y170 = ○

Y171 = ○

Y172 = ●

Y173 = ○

Y174 = ●

Y175 = ○

Y176 = ●

Y177 = ○

Bit device

Values of D100 to D104

D100 = 100

D101 = 200

D102 = 300

D103 = 400

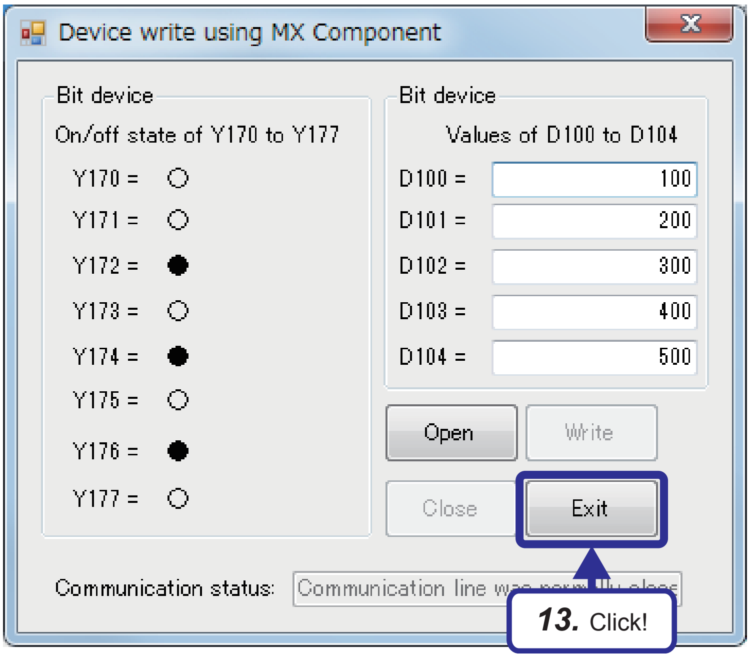
D104 = 500

Communication status: Communication line was normally open

12. Click the [Close] button to close the communication line.
When the communication line is normally closed, "Communication line was normally closed." is displayed in "Communication status".
When an error occurs, the error code is displayed.
For details on error codes, refer to the following.
📖 MX Component Version 4 Programming Manual

(To the next page)

(From the previous page)



Device write using MX Component

Bit device

On/off state of Y170 to Y177

Y170 = ○

Y171 = ○

Y172 = ●

Y173 = ○

Y174 = ●

Y175 = ○

Y176 = ●

Y177 = ○

Bit device

Values of D100 to D104

D100 =	100
D101 =	200
D102 =	300
D103 =	400
D104 =	500

Communication status: Communication line was normally closed

13. Click the [Exit] button to exit the program.
The next solution file can be loaded.

MEMO

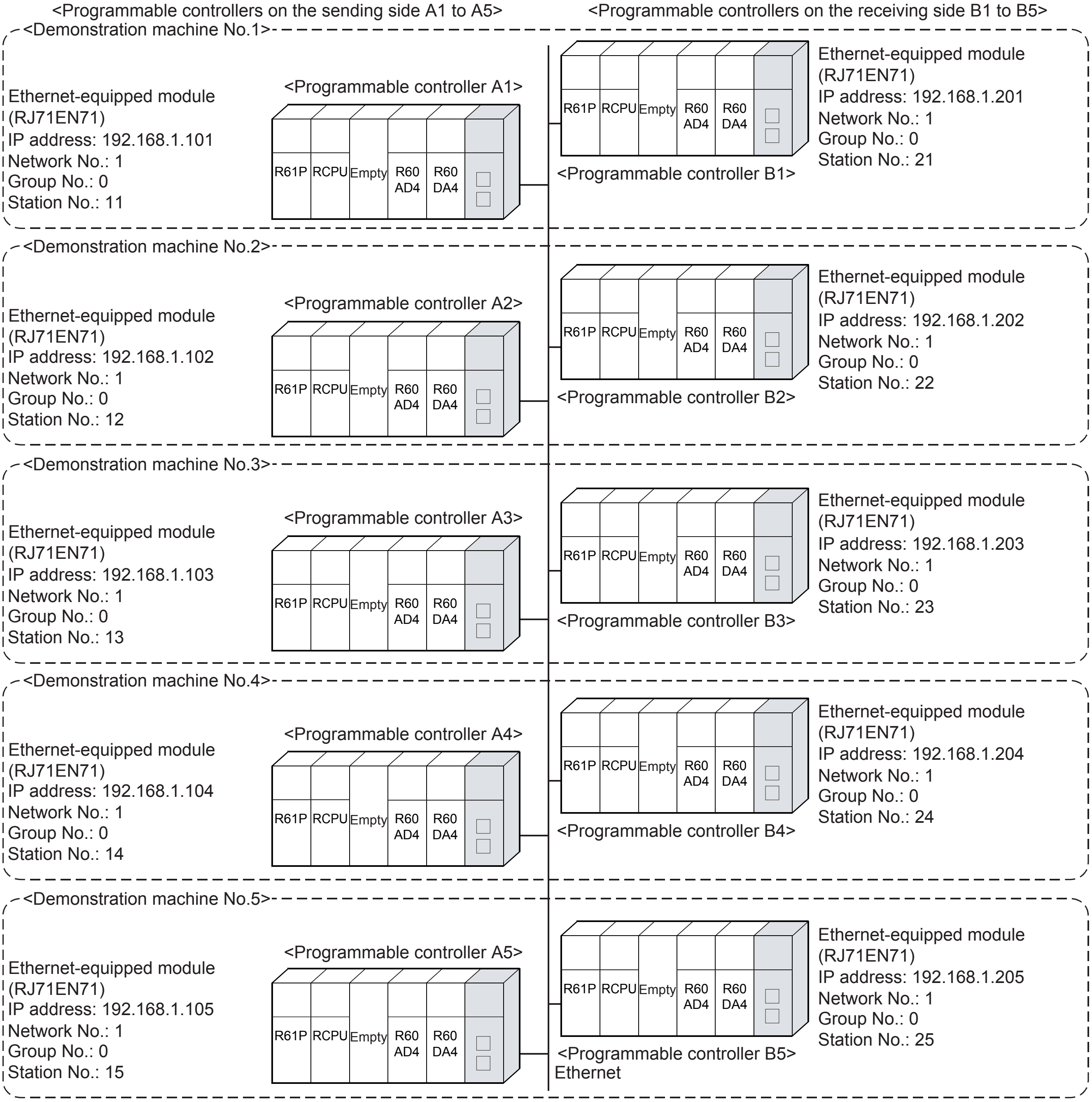
6 EXERCISE 2 (COMMUNICATIONS USING A FIXED BUFFER (PROCEDURE EXIST) BETWEEN CPU MODULES)

This exercise is for communications using a fixed buffer (procedure exist).
The programmable controllers A send data and the programmable controllers B receive data.

6.1 System Configuration

The following figure shows the system configuration of Exercise 2.
In Exercise 2, perform data communications using TCP/IP and monitor the device (X) of the CPU modules on the sending side with the demonstration machines on the receiving side.
Although 10 CPU modules are connected on the same Ethernet network, perform communications only between the CPU modules with the same demonstration machine number in Exercise 2.

Module configuration

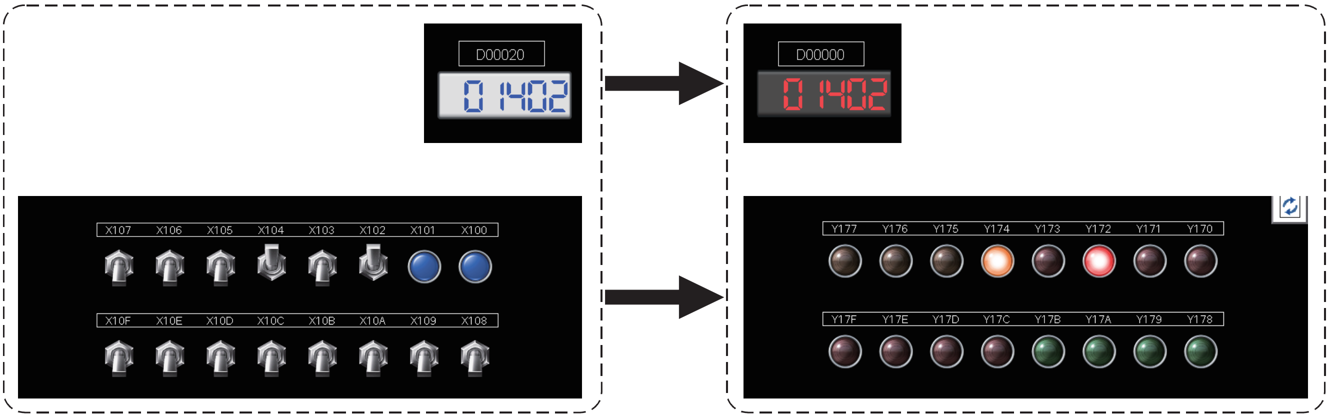


Exercise

Display the information of X100 to X107 and D20 of the CPU module on the sending side on Y170 to Y177 and D0 of the CPU module on the receiving side.

<CPU module on the sending side (Active open)>

<CPU module on the receiving side (Passive open)>


D00020
0 1402
X107 X106 X105 X104 X103 X102 X101 X100
X10F X10E X10D X10C X10B X10A X109 X108
D00000
0 1402
Y177 Y176 Y175 Y174 Y173 Y172 Y171 Y170
Y17F Y17E Y17D Y17C Y17B Y17A Y179 Y178

6.2 Setting the Ethernet-equipped Module on the Sending Side

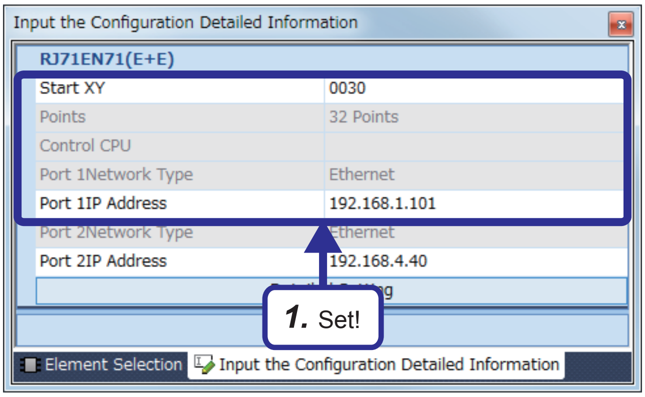
Set the Ethernet-equipped module on the sending side.

6.2.1 Setting parameters with GX Works3

Adding Ethernet-equipped module data

Set parameters of the Ethernet-equipped module with GX Works3.
Set them as described on Page 6 - 1 System Configuration.

Operating procedure

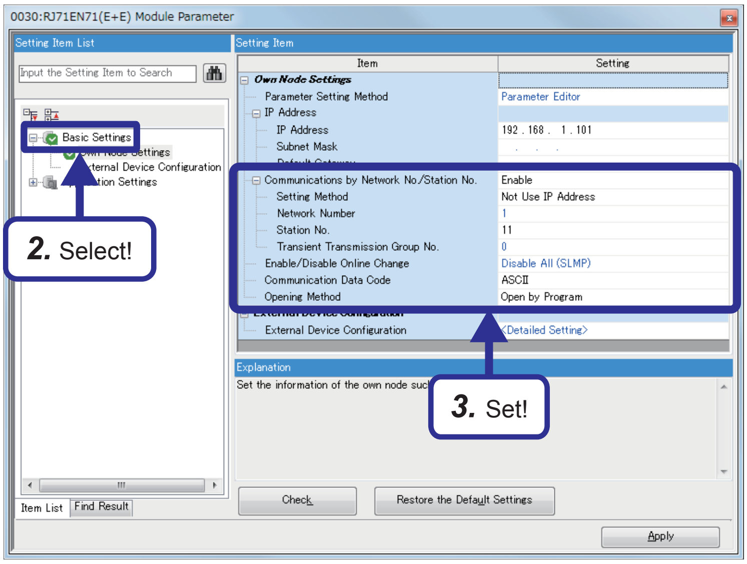


Input the Configuration Detailed Information

RJ71EN71(E+E)	
Start XY	0030
Points	32 Points
Control CPU	
Port 1Network Type	Ethernet
Port 1IP Address	192.168.1.101
Port 2Network Type	Ethernet
Port 2IP Address	192.168.4.40

1. Create a new project and add "RJ71EN71(E+E)" from "Information Module" in the "Element Selection" window to the slot No.3 of the R35B by following the procedure described on Page 5 - 8 Adding Ethernet-equipped module data. After setting the parameters as follows, fix them.
(Click the [OK] button when the confirmation window for adding the module appears.)

[Parameters to be set]
Start XY: 0030
Port 1IP address: Set the IP address (192.168.1.101 to 192.168.1.105) of the demonstration machine used.



0030:RJ71EN71(E+E) Module Parameter

Item	Setting
Own Node Settings	
Parameter Setting Method	Parameter Editor
IP Address	
IP Address	192 . 168 . 1 . 101
Subnet Mask	. . .
Communications by Network No./Station No.	Enable
Setting Method	Not Use IP Address
Network Number	1
Station No.	11
Transient Transmission Group No.	0
Enable/Disable Online Change	Disable All (SLMP)
Communication Data Code	ASCII
Opening Method	Open by Program
External Device Configuration	
External Device Configuration	<Detailed Setting>

Explanation
Set the information of the own node suc

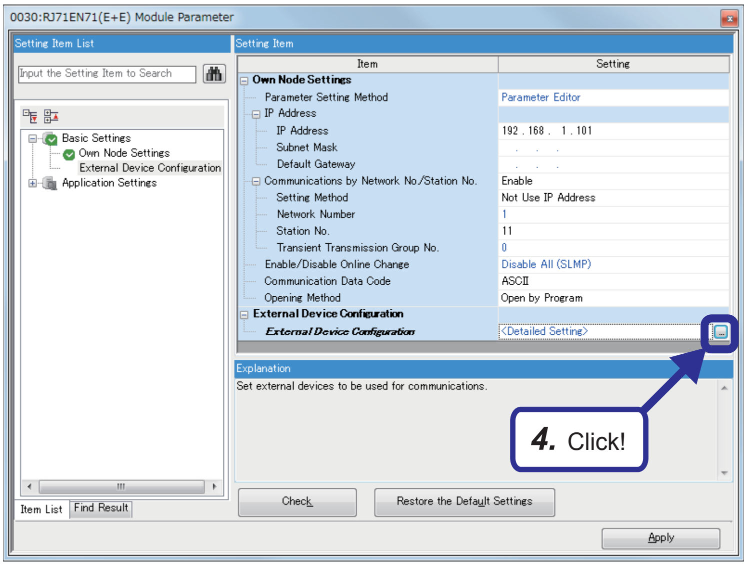
2. Double-click the RJ71EN71(E+E) module in the "Module Configuration" window and select "Basic Settings" in "Setting Item List".

3. Set "Communications by Network No./Station No." as follows.

[Parameters to be set]
Communications by Network No./Station No.: Enable
Setting Method: Not Use IP Address
Network Number: 1
Station No.: Set the station number (11 to 15) of the demonstration machine used.
Communication Data Code: ASCII
Opening Method: Open by Program

(To the next page)

(From the previous page)



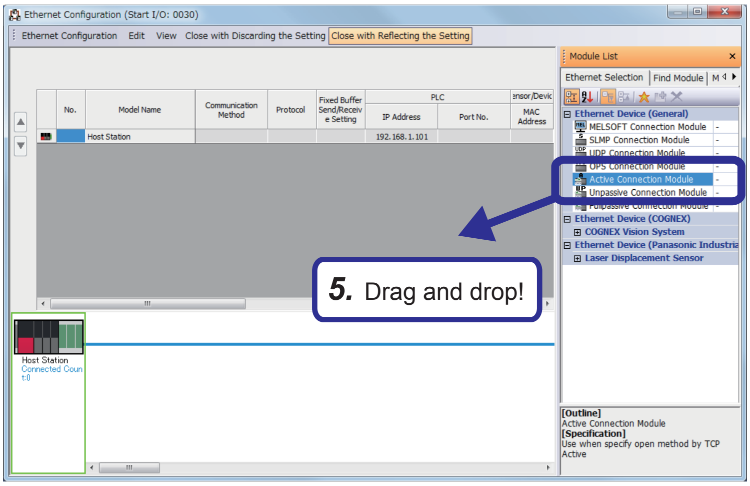
0030:RJ71EN71(E+E) Module Parameter

Item	Setting
Own Node Settings	
Parameter Setting Method	Parameter Editor
IP Address	
IP Address	192 . 168 . 1 . 101
Subnet Mask	. . .
Default Gateway	. . .
Communications by Network No./Station No.	Enable
Setting Method	Not Use IP Address
Network Number	1
Station No.	11
Transient Transmission Group No.	0
Enable/Disable Online Change	Disable All (SLMP)
Communication Data Code	ASCII
Opening Method	Open by Program
External Device Configuration	
External Device Configuration	<Detailed Setting>

Explanation

Set external devices to be used for communications.

4. Click the button of "Detailed Setting" of "External Device Configuration".



Ethernet Configuration (Start I/O: 0030)

	No.	Model Name	Communication Method	Protocol	Fixed Buffer Send/Receive Setting	PLC IP Address	PLC Port No.	Sensor/Device MAC Address
		Host Station				192.168.1.101		

Module List

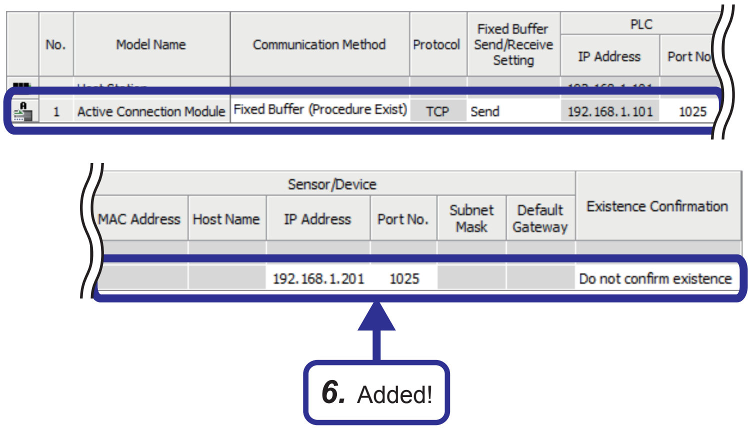
- Ethernet Device (General)
  - MELSOFT Connection Module
  - SLMP Connection Module
  - UDP Connection Module
  - OPS Connection Module
  - Active Connection Module
  - Unpassive Connection Module
  - Fullpassive Connection Module
- Ethernet Device (COGNEX)
  - COGNEX Vision System
- Ethernet Device (Panasonic Industria
  - Laser Displacement Sensor

[Outline]
Active Connection Module
[Specification]
Use when specify open method by TCP Active

5. The "Ethernet Configuration" dialog box appears. Select "Active Connection Module" from "Ethernet Device (General)" in "Module List" and drag and drop it to the list of devices or device map area.

(To the next page)

(From the previous page)

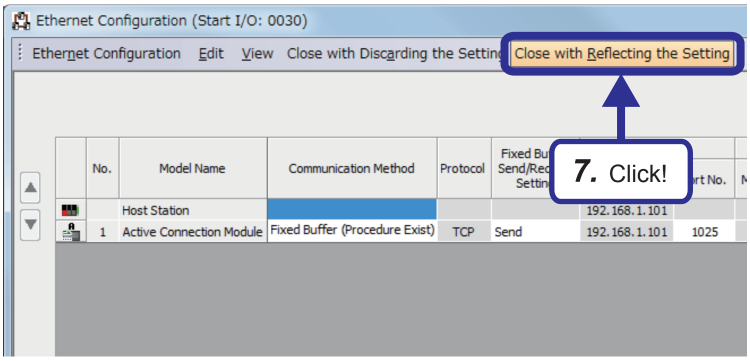


	No.	Model Name	Communication Method	Protocol	Fixed Buffer Send/Receive Setting	PLC IP Address	 Port No
	1	Active Connection Module	Fixed Buffer (Procedure Exist)	TCP	Send	192.168.1.101	1025

Sensor/Device MAC Address	 Host Name	 IP Address	 Port No.	 Subnet Mask	 Default Gateway	Existence Confirmation
		192.168.1.201	1025			Do not confirm existence

6. "Active Connection Module" is added to the list of devices.
Set "Active Connection Module" as follows.

[Parameters to be set]
Communication Method: Fixed Buffer (Procedure Exist)
Fixed Buffer Send/Receive Setting: Send
Port No. (PLC): 1025
IP Address: Set the IP address (192.168.1.201 to 192.168.1.205) of the demonstration machine used.
Port No. (Sensor/Device): 1025
Existence Confirmation: Do not confirm existence



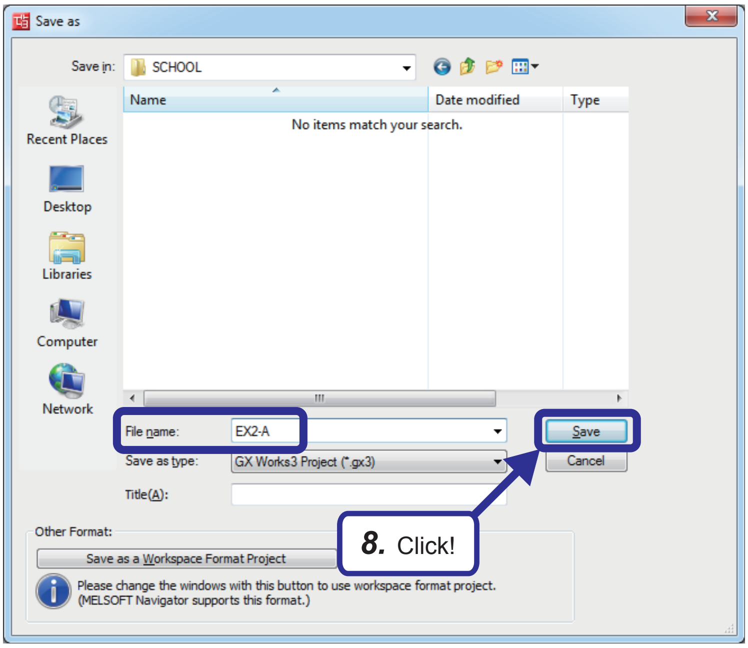
Ethernet Configuration (Start I/O: 0030)
Ethernet Configuration
Edit
View
Close with Discarding the Setting
Close with Reflecting the Setting
7. Click!

	No.	Model Name	Communication Method	Protocol	Fixed Bu Send/Rec Settin		rt No.
		Host Station				192.168.1.101	
	1	Active Connection Module	Fixed Buffer (Procedure Exist)	TCP	Send	192.168.1.101	1025

7. After setting the parameters, click "Close with Reflecting the Setting" in the menu to close the "Ethernet Configuration" dialog box.

(To the next page)

(From the previous page)



File name: EX2-A

Save as type: GX Works3 Project (*.gx3)

Title(A):

Please change the windows with this button to use workspace format project. (MELSOFT Navigator supports this format.)

8. Click [Project] ⇨ [Save As] from the menu to save the project.

Save destination: Desired location
File name: EX2-A
Title: Blank

6.2.2 Sequence program

Create a program that sends data to the CPU module.

Open the connection No.1 and send data.

After sending the data, close the connection No.1.

Point

After creating the sequence program, perform the following operations with GX Works3.

- Saving the project
  Project name: EX2-A
- Specifying a connection destination (Refer to Page 5 - 15 Specifying a connection destination.)
- Writing data to the CPU module (Refer to Page 5 - 17 Writing parameters.)

List of devices used

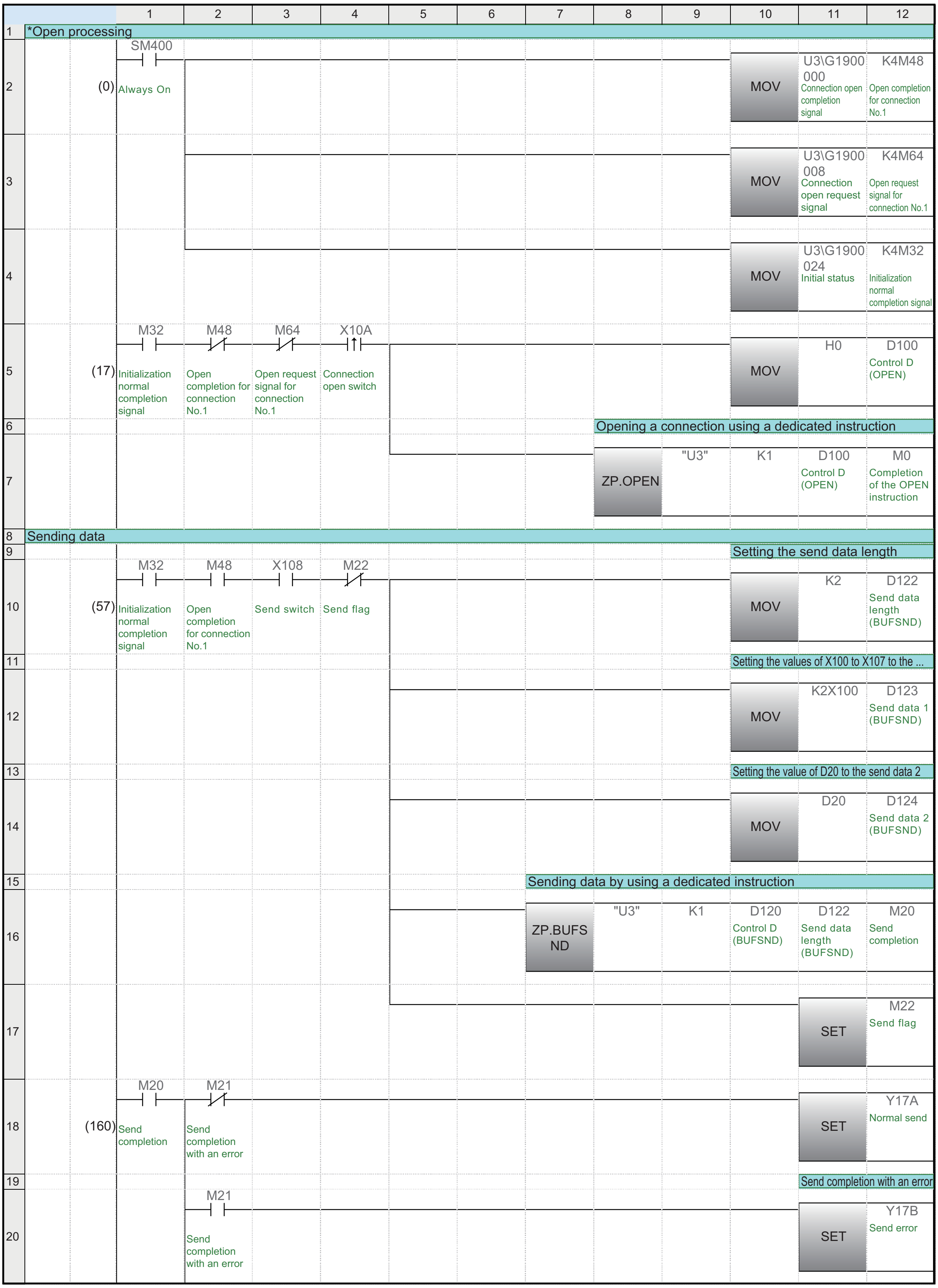
Device name	Description	Device name	Description
SM400	Always On	X108	A command switch to send data
M0	Turns on only for one scan after the open processing for the connection No.1 has completed.	X109	A command switch to clear an error display of the indicator LED
M1	Turns on only for one scan after the open processing for the connection No.1 has completed with an error.	X10A	A command switch to open a connection
M10	Turns on only for one scan after the close processing for the connection No.1 has completed.	X10B	A command switch to close a connection
M11	Turns on only for one scan after the close processing for the connection No.1 has completed with an error.	Y17A	Turns on after the send processing has completed successfully.
M20	Turns on only for one scan after the send processing has completed.	Y17B	Turns on after the send processing has completed with an error.
M21	Turns on only for one scan after the send processing has completed with an error.	D20	Data to be sent to the receiving side
M22	Turns on when data is being sent. Turns off after the send processing has completed.	D100 to D109	Store the control data of the OPEN instruction.
M32	Turns on after the initial processing of the Ethernet-equipped module has completed successfully.	D110 and D111	Store the control data of the CLOSE instruction.
M48	Turns on when the connection No.1 is open.	D120 and D121	Store the control data of the BUFSND instruction.
M64	Turns on when the open request for the connection No.1 is being issued or the connection No.1 is open.	D122	Stores the send data length.
M100	Turns on only for one scan after an error has cleared.	D123	Stores send data (X100 to X107).
M110	Turns on when an error is cleared. Turns off when an error has cleared.	D124	Stores send data (D20).
X100 to X107	Data to be sent to the receiving side	D140 to D144	Store the control data of the ERRCLEAR instruction.

Sequence program

For details on the dedicated instructions, refer to the following.

📖 MELSEC iQ-R Programming Manual (Instructions, Standard Functions/Function Blocks)

Project name	EX2-A



1 *Open processing

2 (0) SM400 Always On — MOV U3\G1900000 Connection open completion signal, K4M48 Open completion for connection No.1

3 MOV U3\G1900008 Connection open request signal, K4M64 Open request signal for connection No.1

4 MOV U3\G1900024 Initial status, K4M32 Initialization normal completion signal

5 (17) M32 Initialization normal completion signal, M48 Open completion for connection No.1, M64 Open request signal for connection No.1, X10A Connection open switch — MOV H0 D100 Control D (OPEN)

6 Opening a connection using a dedicated instruction

7 ZP.OPEN "U3" K1 D100 Control D (OPEN), M0 Completion of the OPEN instruction

8 Sending data

9 Setting the send data length

10 (57) M32 Initialization normal completion signal, M48 Open completion for connection No.1, X108 Send switch, M22 Send flag — MOV K2 D122 Send data length (BUFSND)

11 Setting the values of X100 to X107 to the ...

12 MOV K2X100 D123 Send data 1 (BUFSND)

13 Setting the value of D20 to the send data 2

14 MOV D20 D124 Send data 2 (BUFSND)

15 Sending data by using a dedicated instruction

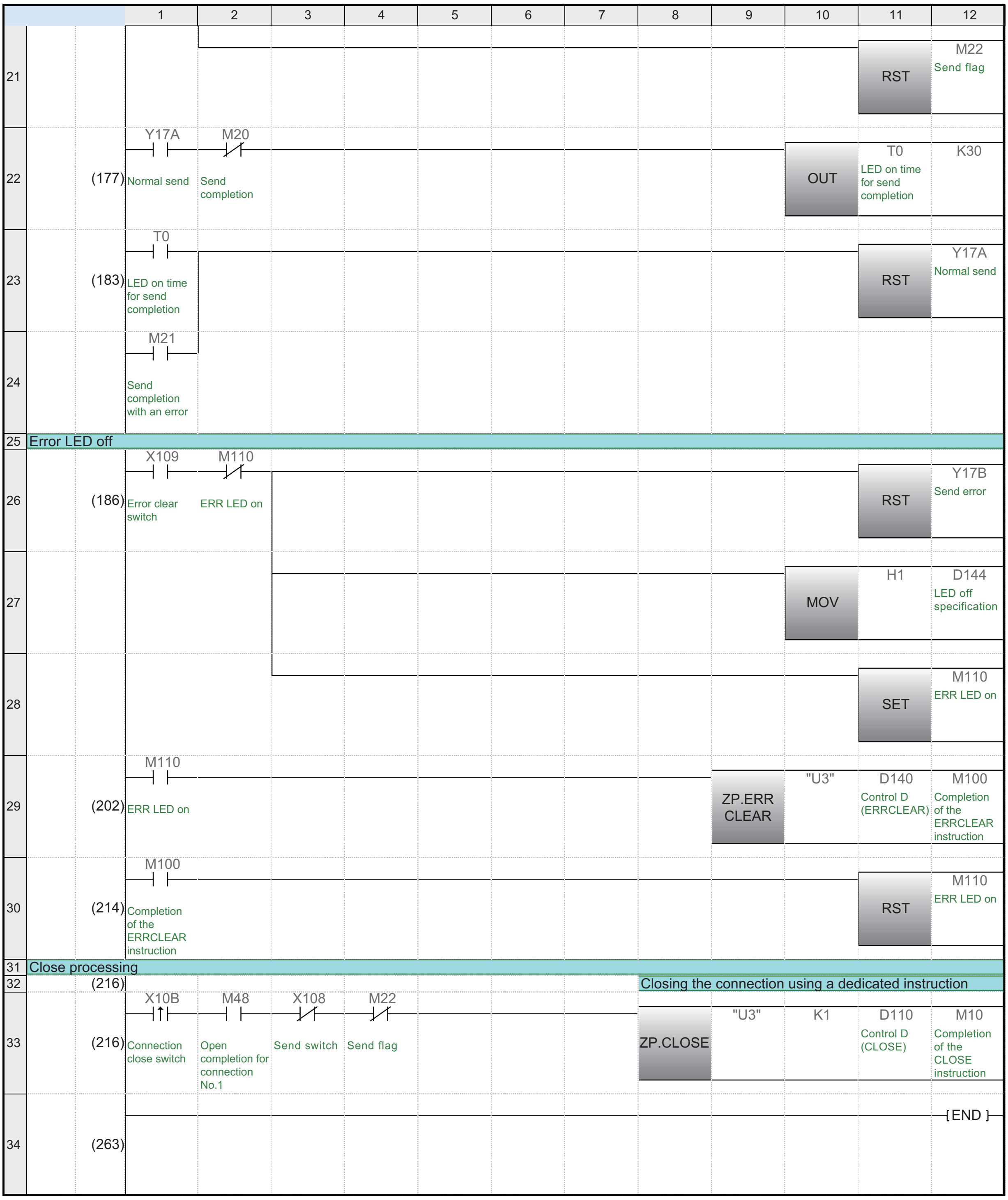
16 ZP.BUFSND "U3" K1 D120 Control D (BUFSND), D122 Send data length (BUFSND), M20 Send completion

17 SET M22 Send flag

18 (160) M20 Send completion, M21 Send completion with an error — SET Y17A Normal send

19 Send completion with an error

20 M21 Send completion with an error — SET Y17B Send error



```
21                                                           RST   M22 Send flag
22  (177) Y17A Normal send   M20 Send completion              OUT   T0 LED on time for send completion   K30
23  (183) T0 LED on time for send completion                  RST   Y17A Normal send
24        M21 Send completion with an error
25  Error LED off
26  (186) X109 Error clear switch   M110 ERR LED on           RST   Y17B Send error
27                                                           MOV   H1   D144 LED off specification
28                                                           SET   M110 ERR LED on
29  (202) M110 ERR LED on                   ZP.ERR CLEAR   "U3"   D140 Control D (ERRCLEAR)   M100 Completion of the ERRCLEAR instruction
30  (214) M100 Completion of the ERRCLEAR instruction         RST   M110 ERR LED on
31  Close processing
32  (216)                                   Closing the connection using a dedicated instruction
33  (216) X10B Connection close switch   M48 Open completion for connection No.1   X108 Send switch   M22 Send flag
                                            ZP.CLOSE   "U3"   K1   D110 Control D (CLOSE)   M10 Completion of the CLOSE instruction
34  (263)                                                                  END
```

6.3 Setting the Ethernet-equipped Module on the Receiving Side

Set the Ethernet-equipped module on the receiving side.

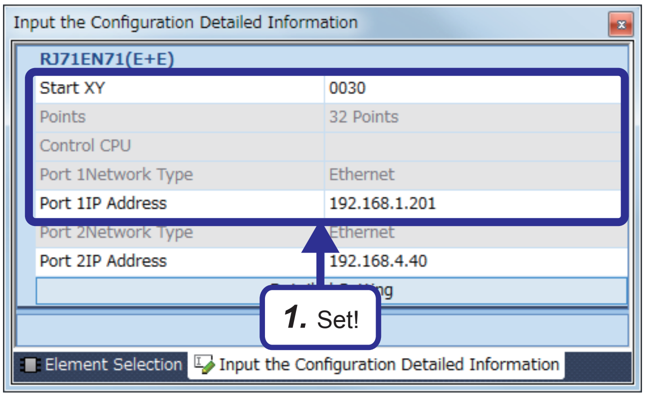
6.3.1 Setting parameters with GX Works3

Adding Ethernet-equipped module data

Set parameters of the Ethernet-equipped module with GX Works3.

Set them as described on Page 6 - 1 System Configuration.

Operating procedure



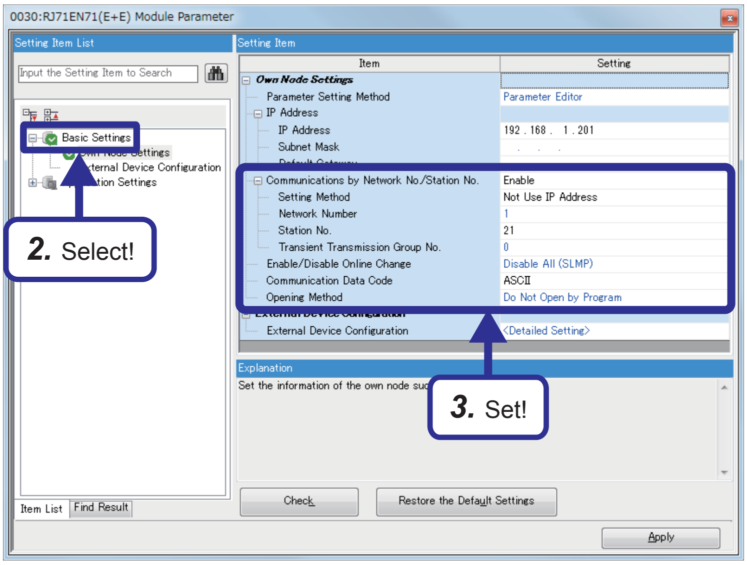
Input the Configuration Detailed Information

RJ71EN71(E+E)

Start XY	0030
Points	32 Points
Control CPU	
Port 1Network Type	Ethernet
Port 1IP Address	192.168.1.201
Port 2Network Type	Ethernet
Port 2IP Address	192.168.4.40

1. Create a new project and add "RJ71EN71(E+E)" from "Information Module" in the "Element Selection" window to the slot No.3 of the R35B by following the procedure described on Page 5 - 8 Adding Ethernet-equipped module data. After setting the parameters as follows, fix them.
(Click the [OK] button when the confirmation window for adding the module appears.)

[Parameters to be set]
Start XY: 0030
Port 1IP address: Set the IP address (192.168.1.201 to 192.168.1.205) of the demonstration machine used.



0030:RJ71EN71(E+E) Module Parameter

Item	Setting
Own Node Settings	
Parameter Setting Method	Parameter Editor
IP Address	
IP Address	192 . 168 . 1 . 201
Subnet Mask	. . .
Default Gateway	
Communications by Network No./Station No.	Enable
Setting Method	Not Use IP Address
Network Number	1
Station No.	21
Transient Transmission Group No.	0
Enable/Disable Online Change	Disable All (SLMP)
Communication Data Code	ASCII
Opening Method	Do Not Open by Program
External Device Configuration	
External Device Configuration	<Detailed Setting>

Explanation
Set the information of the own node suc

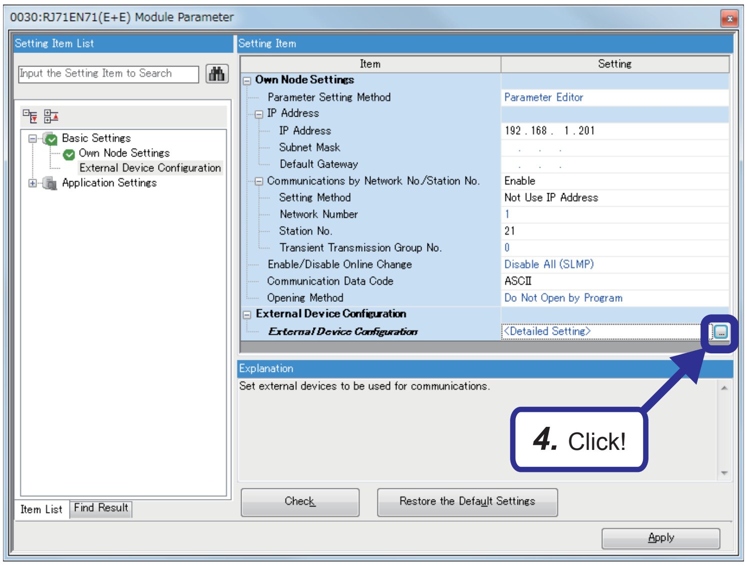
2. Double-click the RJ71EN71(E+E) module in the "Module Configuration" window and select "Basic Settings" in "Setting Item List".

3. Set "Communications by Network No./Station No." as follows.

[Parameters to be set]
Communications by Network No./Station No.: Enable
Setting Method: Not Use IP Address
Network Number: 1
Station No.: Set the station number (21 to 25) of the demonstration machine used.
Communication Data Code: ASCII
Opening Method: Do Not Open by Program

(To the next page)

(From the previous page)



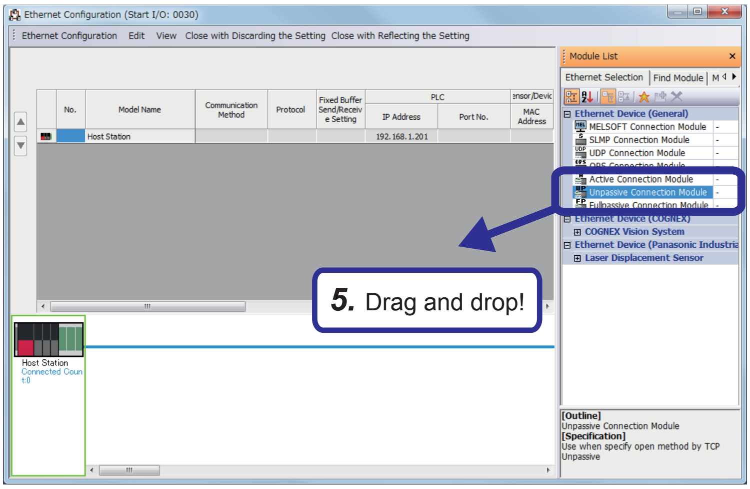
0030:RJ71EN71(E+E) Module Parameter

Item	Setting
Own Node Settings	
Parameter Setting Method	Parameter Editor
IP Address	
IP Address	192 . 168 . 1 . 201
Subnet Mask	. . .
Default Gateway	. . .
Communications by Network No./Station No.	Enable
Setting Method	Not Use IP Address
Network Number	1
Station No.	21
Transient Transmission Group No.	0
Enable/Disable Online Change	Disable All (SLMP)
Communication Data Code	ASCII
Opening Method	Do Not Open by Program
External Device Configuration	
External Device Configuration	<Detailed Setting>

Explanation

Set external devices to be used for communications.

4. Click the … button of "Detailed Setting" of "External Device Configuration".



Ethernet Configuration (Start I/O: 0030)

Ethernet Configuration
Edit
View
Close with Discarding the Setting
Close with Reflecting the Setting

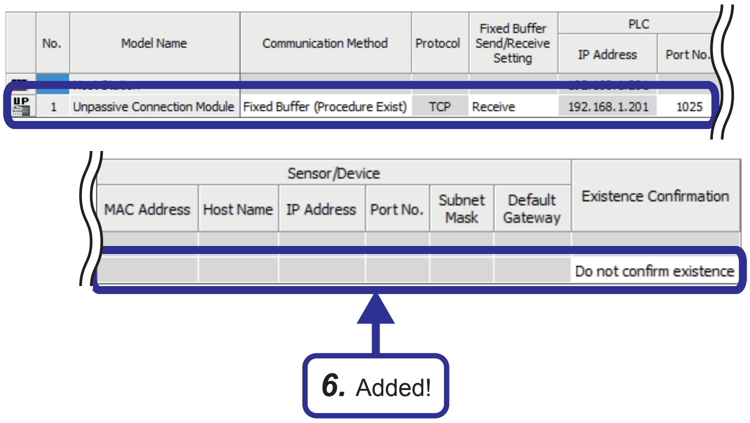
No.	Model Name	Communication Method	Protocol	Fixed Buffer Send/Receive Setting	PLC IP Address	PLC Port No.	Sensor/Device MAC Address
	Host Station				192.168.1.201		

[Outline]
Unpassive Connection Module
[Specification]
Use when specify open method by TCP Unpassive

5. The "Ethernet Configuration" dialog box appears. Select "Unpassive Connection Module" from "Ethernet Device (General)" in "Module List" and drag and drop it to the list of devices or device map area.

(To the next page)

(From the previous page)

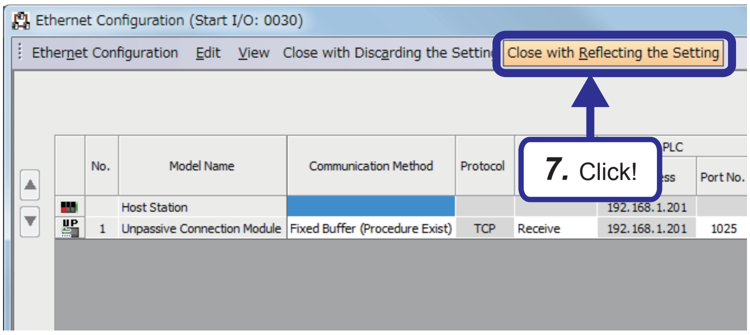


No.	Model Name	Communication Method	Protocol	Fixed Buffer Send/Receive Setting	PLC	
					IP Address	Port No.
1	Unpassive Connection Module	Fixed Buffer (Procedure Exist)	TCP	Receive	192.168.1.201	1025

Sensor/Device						Existence Confirmation
MAC Address	Host Name	IP Address	Port No.	Subnet Mask	Default Gateway	
						Do not confirm existence

6. "Unpassive Connection Module" is added to the list of devices.
Set "Unpassive Connection Module" as follows.

[Parameters to be set]
Communication Method: Fixed Buffer (Procedure Exist)
Fixed Buffer Send/Receive Setting: Receive
Port No.: 1025
Existence Confirmation: Do not confirm existence

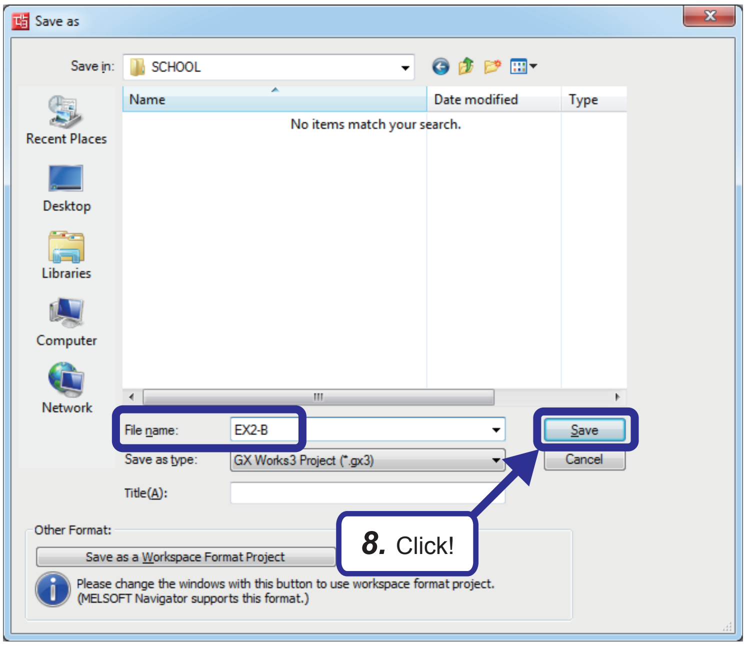


Ethernet Configuration (Start I/O: 0030)

Ethernet Configuration
Edit
View
Close with Discarding the Settin
Close with Reflecting the Setting
7. Click!

No.	Model Name	Communication Method	Protocol	PLC		
					ss	Port No.
	Host Station				192.168.1.201	
1	Unpassive Connection Module	Fixed Buffer (Procedure Exist)	TCP	Receive	192.168.1.201	1025

7. After setting the parameters, click "Close with Reflecting the Setting" in the menu to close the "Ethernet Configuration" dialog box.



Save as

Save in: SCHOOL

Name	Date modified	Type

No items match your search.

File name: EX2-B
Save as type: GX Works3 Project (*.gx3)
Title(A):

Other Format:

Please change the windows with this button to use workspace format project.
(MELSOFT Navigator supports this format.)

8. Click [Project] ⇨ [Save As] from the menu to save the project.

Save destination: Desired location
File name: EX2-B
Title: Blank

6.3.2 Sequence program

Create a program that receives the data sent from the external devices.
Output the sent data (X100 to X107 and D20) to Y170 to Y177 and D0 after receive processing has completed successfully.

Point

After creating the sequence program, perform the following operations with GX Works3.

- Saving the project
  Project name: EX2-B
- Specifying a connection destination (Refer to Page 5 - 15 Specifying a connection destination.)
- Writing data to the CPU module (Refer to Page 5 - 17 Writing parameters.)

List of devices used

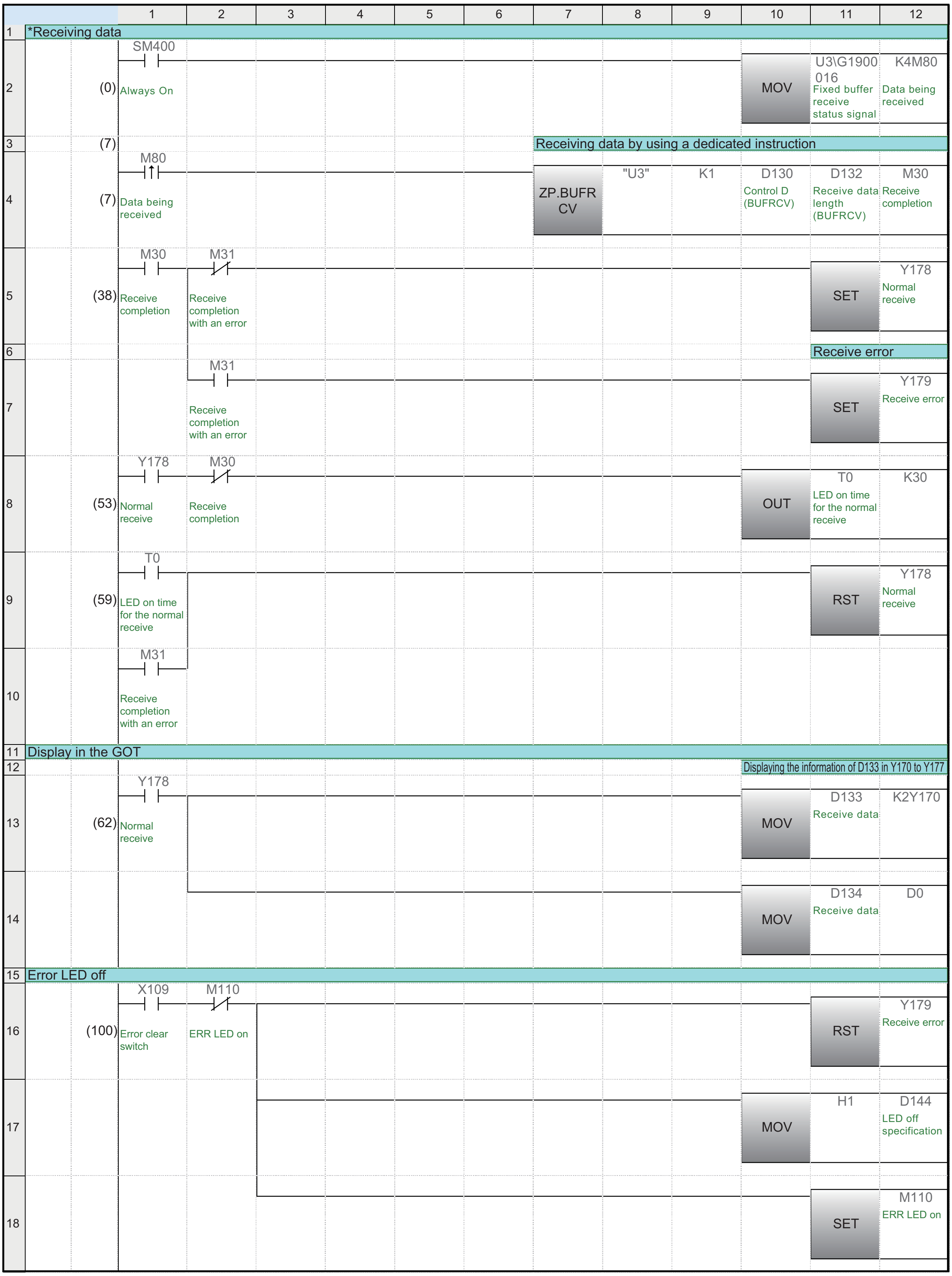
Device name	Description	Device name	Description
SM400	Always On	Y178	Turns on after the receive processing has completed successfully.
M30	Turns on only for one scan after the receive processing has completed.	Y179	Turns on after the receive processing has completed with an error.
M31	Turns on only for one scan after the receive processing has completed with an error.	D0	Stores the information of D20 sent from the sending side.
M80	Turns on while the connection No.1 of the Ethernet-equipped module is receiving data.	D130 and D131	Store the control data of the BUFRCV instruction.
M100	Turns on only for one scan after an error has cleared.	D132	Stores the receive data length.
M110	Turns on when an error is cleared. Turns off when an error has cleared.	D133	Stores receive data (X100 to X107).
X109	A command switch to clear an error display of the indicator LED	D134	Stores receive data (D20).
Y170 to Y177	Store the information of X100 to X107 sent from the sending side.	D140 to D144	Store the control data of the ERRCLEAR instruction.

Sequence program

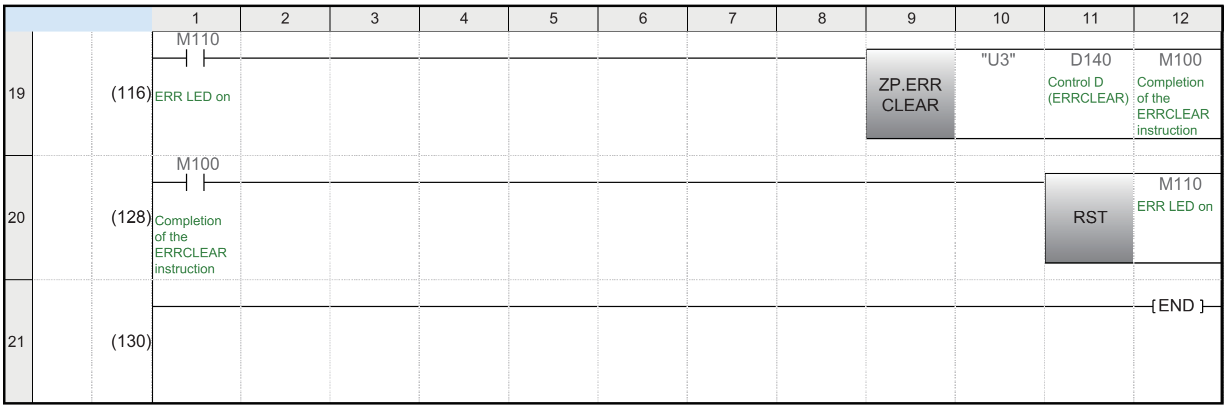
For details on the dedicated instructions, refer to the following.

📖 MELSEC iQ-R Programming Manual (Instructions, Standard Functions/Function Blocks)

Project name	EX2-B



1 2 3 4 5 6 7 8 9 10 11 12
1 *Receiving data
2 (0) SM400 Always On — MOV U3\G1900016 Fixed buffer receive status signal K4M80 Data being received
3 (7) Receiving data by using a dedicated instruction
4 (7) M80 Data being received — ZP.BUFRCV "U3" K1 D130 Control D (BUFRCV) D132 Receive data length (BUFRCV) M30 Receive completion
5 (38) M30 Receive completion, M31 Receive completion with an error — SET Y178 Normal receive
6 Receive error
7 M31 Receive completion with an error — SET Y179 Receive error
8 (53) Y178 Normal receive, M30 Receive completion — OUT T0 LED on time for the normal receive K30
9 (59) T0 LED on time for the normal receive — RST Y178 Normal receive
10 M31 Receive completion with an error
11 Display in the GOT
12 Displaying the information of D133 in Y170 to Y177
13 (62) Y178 Normal receive — MOV D133 Receive data K2Y170
14 MOV D134 Receive data D0
15 Error LED off
16 (100) X109 Error clear switch, M110 ERR LED on — RST Y179 Receive error
17 MOV H1 D144 LED off specification
18 SET M110 ERR LED on



19 (116) M110 ERR LED on — ZP.ERRCLEAR "U3" D140 Control D (ERRCLEAR) M100 Completion of the ERRCLEAR instruction

20 (128) M100 Completion of the ERRCLEAR instruction — RST M110 ERR LED on

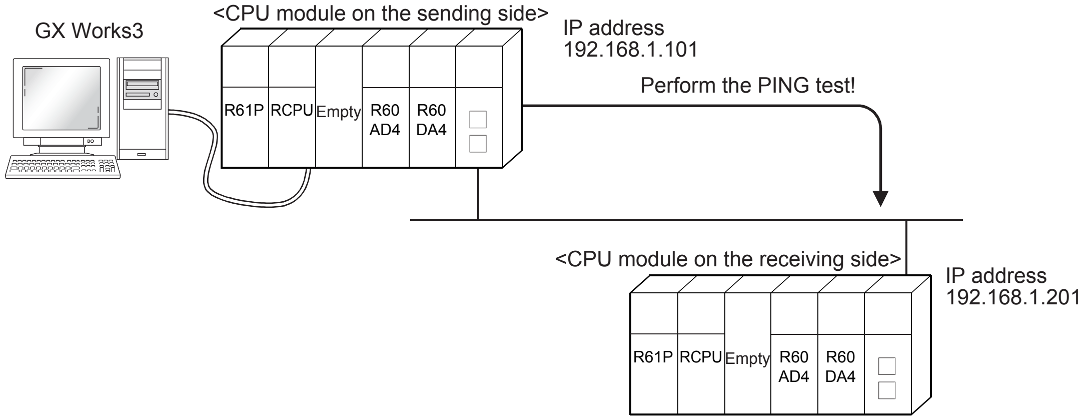
21 (130) END

6.4 PING Test with GX Works3 (via CPU Modules)

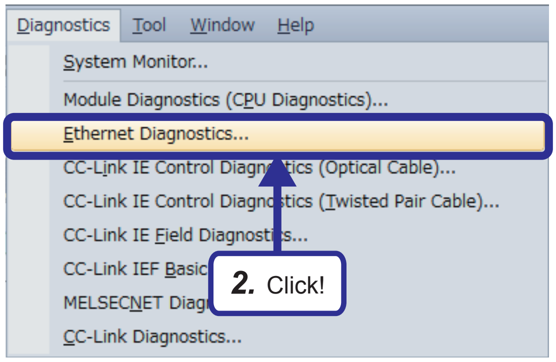
Perform the PING test with GX Works3.

The PING test checks whether the initial processing of the Ethernet-equipped module on the own station has completed successfully and the Ethernet-equipped module is properly connected to the external device on the same Ethernet network.

Power on the CPU module on the sending and receiving sides, and set the CPU module to the STOP state before performing the PING test.

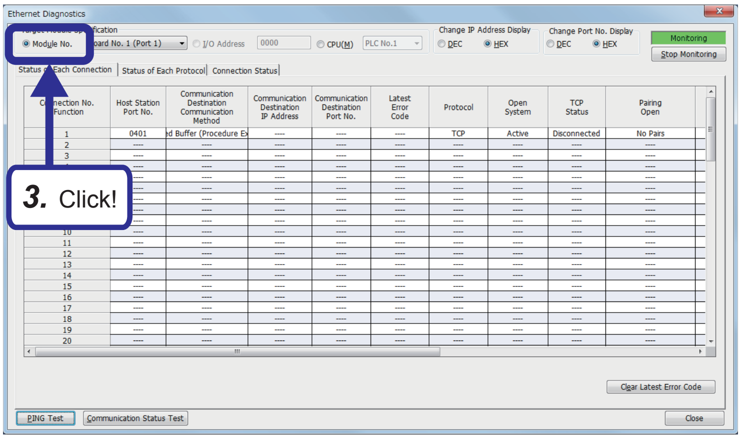


Operating procedure


Diagnostics
Tool
Window
Help
System Monitor...
Module Diagnostics (CPU Diagnostics)...
Ethernet Diagnostics...
CC-Link IE Control Diagnostics (Optical Cable)...
CC-Link IE Control Diagnostics (Twisted Pair Cable)...
CC-Link IE Field Diagnostics...
CC-Link IEF Basic
MELSECNET Diag
CC-Link Diagnostics...
2. Click!

1. Open the project file of the CPU module on the sending side.

2. Click [Diagnostics] → [Ethernet Diagnostics] from the menu.

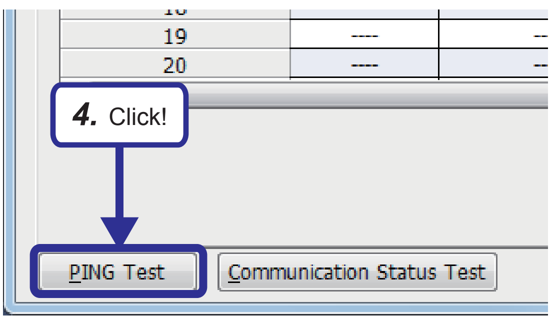


Connection No. Function	Host Station Port No.	Communication Destination Communication Method	Communication Destination IP Address	Communication Destination Port No.	Latest Error Code	Protocol	Open System	TCP Status	Pairing Open
1	0401	ed Buffer (Procedure Ex	----	----	----	TCP	Active	Disconnected	No Pairs
2	----	----	----	----	----	----	----	----	----
3	----	----	----	----	----	----	----	----	----
	----	----	----	----	----	----	----	----	----
	----	----	----	----	----	----	----	----	----
	----	----	----	----	----	----	----	----	----
	----	----	----	----	----	----	----	----	----
	----	----	----	----	----	----	----	----	----
	----	----	----	----	----	----	----	----	----
10	----	----	----	----	----	----	----	----	----
11	----	----	----	----	----	----	----	----	----
12	----	----	----	----	----	----	----	----	----
13	----	----	----	----	----	----	----	----	----
14	----	----	----	----	----	----	----	----	----
15	----	----	----	----	----	----	----	----	----
16	----	----	----	----	----	----	----	----	----
17	----	----	----	----	----	----	----	----	----
18	----	----	----	----	----	----	----	----	----
19	----	----	----	----	----	----	----	----	----
20	----	----	----	----	----	----	----	----	----

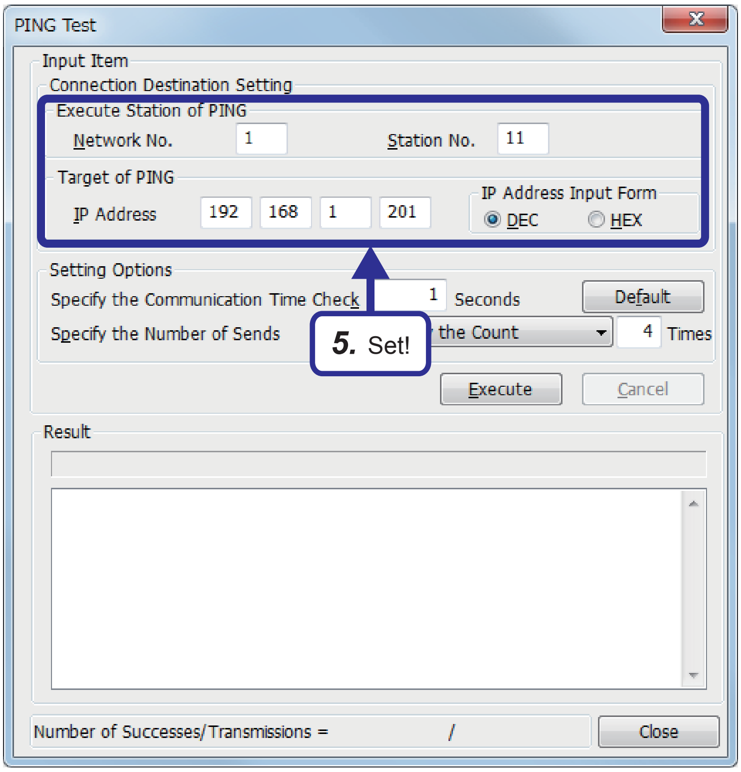
3. The "Ethernet Diagnostics" window appears. Specify the target Ethernet-equipped module on the own station in "Module No.".

(To the next page)

(From the previous page)



4. Click the [PING Test] button.



PING Test

Input Item

Connection Destination Setting

Execute Station of PING

Network No. 1   Station No. 11

Target of PING

IP Address 192 168 1 201

IP Address Input Form

● DEC ○ HEX

Setting Options

Specify the Communication Time Check 1 Seconds

Specify the Number of Sends ... the Count 4 Times

Result

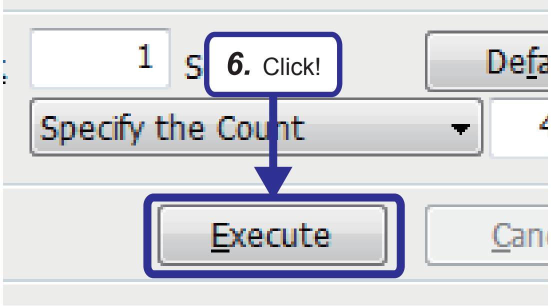
Number of Successes/Transmissions = /

5. The "PING Test" window appears. Set the parameters as follows.

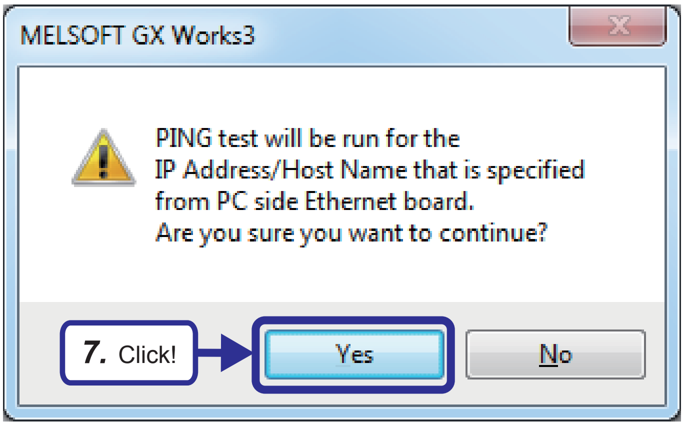
[Parameters to be set]
Network No.: 1
Station No.: 11
IP Address: Set the IP address (192.168.1.201 to 192.168.1.205) of the test-target communication destination demonstration machine.

(To the next page)

(From the previous page)

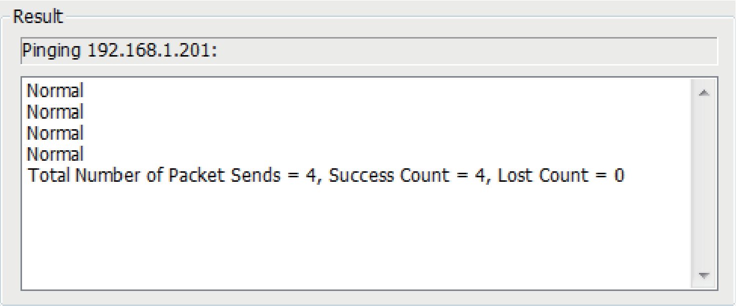


6. Click the [Execute] button.



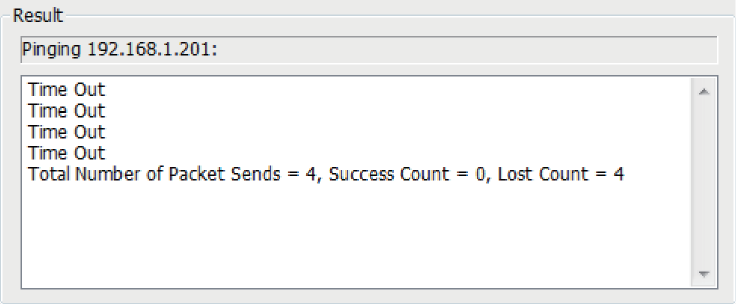
PING test will be run for the
IP Address/Host Name that is specified
from PC side Ethernet board.
Are you sure you want to continue?

7. Click the [Yes] button to perform the PING test.



```
Pinging 192.168.1.201:
Normal
Normal
Normal
Normal
Total Number of Packet Sends = 4, Success Count = 4, Lost Count = 0
```

8. The execution result of the PING test is displayed.



```
Pinging 192.168.1.201:
Time Out
Time Out
Time Out
Time Out
Total Number of Packet Sends = 4, Success Count = 0, Lost Count = 4
```

When "Time Out" is displayed, the IP address of the communication destination demonstration device may be incorrect or the communications may be disconnected. Check the IP address and the connection.

Point

The following provides an example for checking completion of the initial processing by issuing the PING command from the external device connected on the same Ethernet network to the Ethernet-equipped module on the own station (IBM-PC/AT-compatible personal computer → Ethernet-equipped module).

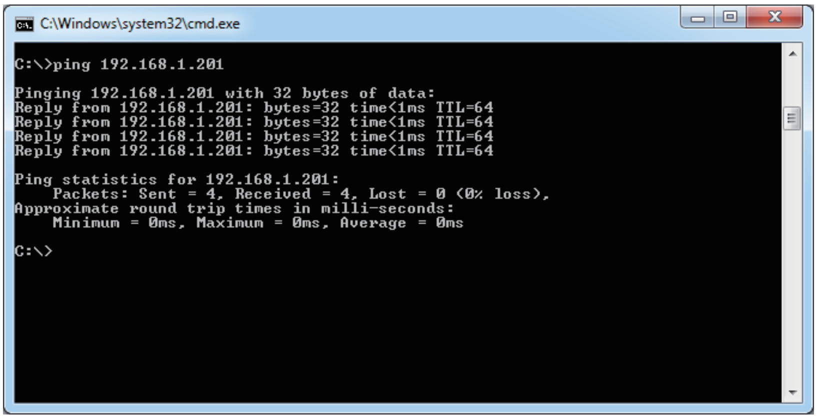
(An example between the devices that have the same class and network ID in their IP addresses)

[Specification method]

Ping IP address

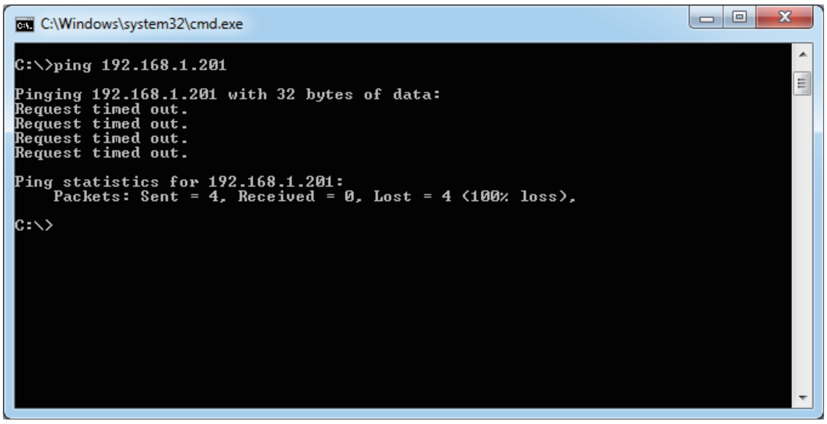
[Program example]

IP address of the Ethernet-equipped module: 192.168.1.201



C:\Windows\system32\cmd.exe

```
C:\>ping 192.168.1.201

Pinging 192.168.1.201 with 32 bytes of data:
Reply from 192.168.1.201: bytes=32 time<1ms TTL=64
Reply from 192.168.1.201: bytes=32 time<1ms TTL=64
Reply from 192.168.1.201: bytes=32 time<1ms TTL=64
Reply from 192.168.1.201: bytes=32 time<1ms TTL=64

Ping statistics for 192.168.1.201:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\>
```



C:\Windows\system32\cmd.exe

```
C:\>ping 192.168.1.201

Pinging 192.168.1.201 with 32 bytes of data:
Request timed out.
Request timed out.
Request timed out.
Request timed out.

Ping statistics for 192.168.1.201:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

C:\>
```

<When the processing has completed with an error>

Check the following items and send the PING command again.

- Mounting status of the Ethernet-equipped module to the base unit
- Connection to the Ethernet network
- Parameter settings written to the CPU module
- Operating status of the CPU module (whether or not an error has occurred)
- IP address of the send destination Ethernet-equipped module specified by the PING command

6.5 Operation of the Demonstration Machine

Operate the demonstration machine to check that data is sent from the CPU module A (sending side) to the CPU module B (receiving side) over Ethernet.

Operation method

1. Write parameters and programs to the CPU module A and CPU module B.

2. Set the RUN/STOP/RESET switch of the CPU module A (sending side) and CPU module B (receiving side) to the "RESET" position (for approximately one second) to reset the CPU modules. Then, move it to the "RUN" position.

3. Turn on and off X10A of the CPU module A to open the connection.
   The CPU module B is waiting for the connection to be open (Unpassive open) because the following value is set in the operation setting parameter.
   Opening Method: Do Not Open by Program

4. Turn on/off X100 to X107 of the CPU module A. Input a numerical value in the initial input device (D20).
   Turn on X108 of the CPU module A to send data.

5. Y170 to Y177 of the CPU module B turn on according to the status of X100 to X107 of the CPU module A.
   The numerical value of the initial input device (D20) of the CPU module A is displayed in the initial indication device (D0) of the CPU module B.

6. After checking the value, turn off X108 of the CPU module A and end the send processing.
   To send the data again, start from Step 4.

7. Turn on and off X10B of the CPU module A to close the connection.

6.6 Additional Exercises

6.6.1 Additional exercise 1

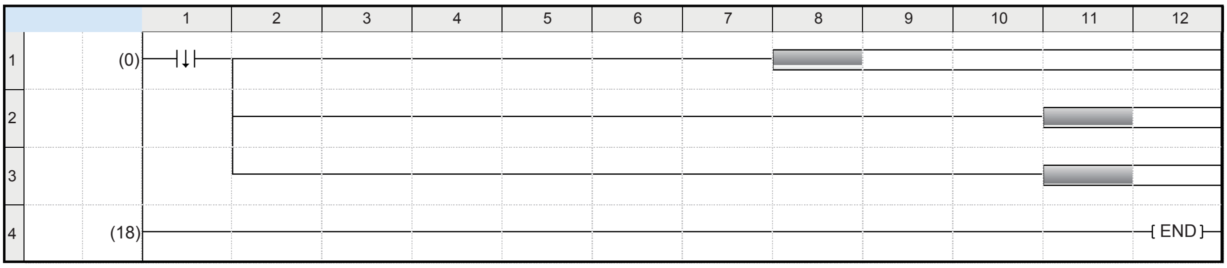
Operation method

1. On Page 6 - 21 Operation of the Demonstration Machine, disconnect the Ethernet cable from the Ethernet-equipped module on the demonstration machine A side while data is being sent from the CPU module A to the CPU module B by using the BUFSND instruction.

2. The open processing for the connection of the Ethernet-equipped module stops and the ERR LED and P ERR LED turn on.

3. Even when the Ethernet cable is connected again and the open instruction is executed, the open processing will not be performed.

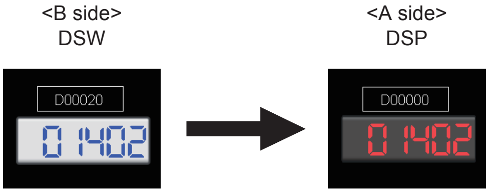
4. Identify the cause by executing the read monitor function or other functions of GX Works3.
Add the following sequence program to open the connection again without resetting the CPU module.
(For the answer, refer to Page App. - 161 Answers for the Additional Exercise 1 (Section 6.6.1).)



5. After writing the sequence program, disconnect and connect the Ethernet cable and check that the open processing is performed.

6.6.2 Additional exercise 2

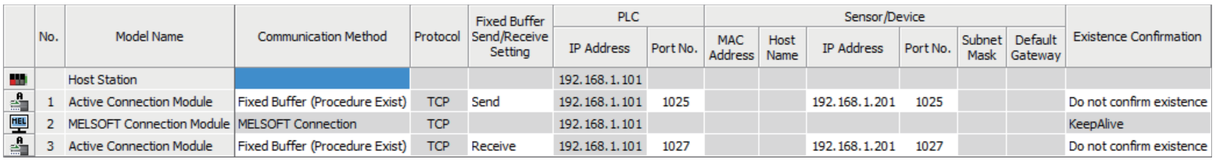
Display the information of D20 of the CPU module B on D0 of the CPU module A using the fixed buffer No.3.



Module Parameter (External Device Configuration)

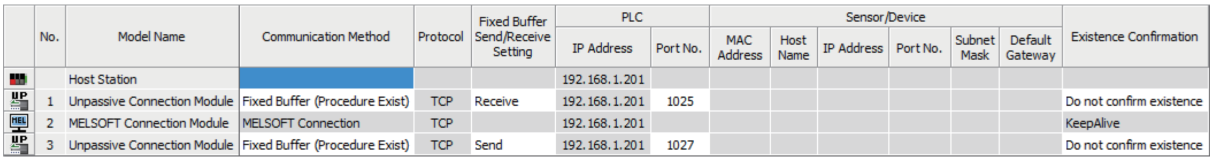
Change the settings of "Module Parameter (External Device Configuration)" of the CPU module A and B as follows.

CPU module B



	No.	Model Name	Communication Method	Protocol	Fixed Buffer Send/Receive Setting	PLC		Sensor/Device						Existence Confirmation
						IP Address	Port No.	MAC Address	Host Name	IP Address	Port No.	Subnet Mask	Default Gateway	
		Host Station				192.168.1.101								
A	1	Active Connection Module	Fixed Buffer (Procedure Exist)	TCP	Send	192.168.1.101	1025			192.168.1.201	1025			Do not confirm existence
MEL	2	MELSOFT Connection Module	MELSOFT Connection	TCP		192.168.1.101								KeepAlive
A	3	Active Connection Module	Fixed Buffer (Procedure Exist)	TCP	Receive	192.168.1.101	1027			192.168.1.201	1027			Do not confirm existence

CPU module A



	No.	Model Name	Communication Method	Protocol	Fixed Buffer Send/Receive Setting	PLC		Sensor/Device						Existence Confirmation
						IP Address	Port No.	MAC Address	Host Name	IP Address	Port No.	Subnet Mask	Default Gateway	
		Host Station				192.168.1.201								
UP	1	Unpassive Connection Module	Fixed Buffer (Procedure Exist)	TCP	Receive	192.168.1.201	1025							Do not confirm existence
MEL	2	MELSOFT Connection Module	MELSOFT Connection	TCP		192.168.1.201								KeepAlive
UP	3	Unpassive Connection Module	Fixed Buffer (Procedure Exist)	TCP	Send	192.168.1.201	1027							Do not confirm existence

* The IP address of the communication destination varies depending on the demonstration machine number.

Sequence program

CPU module A: Use the receive program on Page 6 - 14 Sequence program and add a program that opens/closes the fixed buffer No.3 (connection No.3).

CPU module B: Use the send program on Page 6 - 8 Sequence program.

Operation of the demonstration machine

Check that the data is sent referring to Page 6 - 21 Operation of the Demonstration Machine.

MEMO

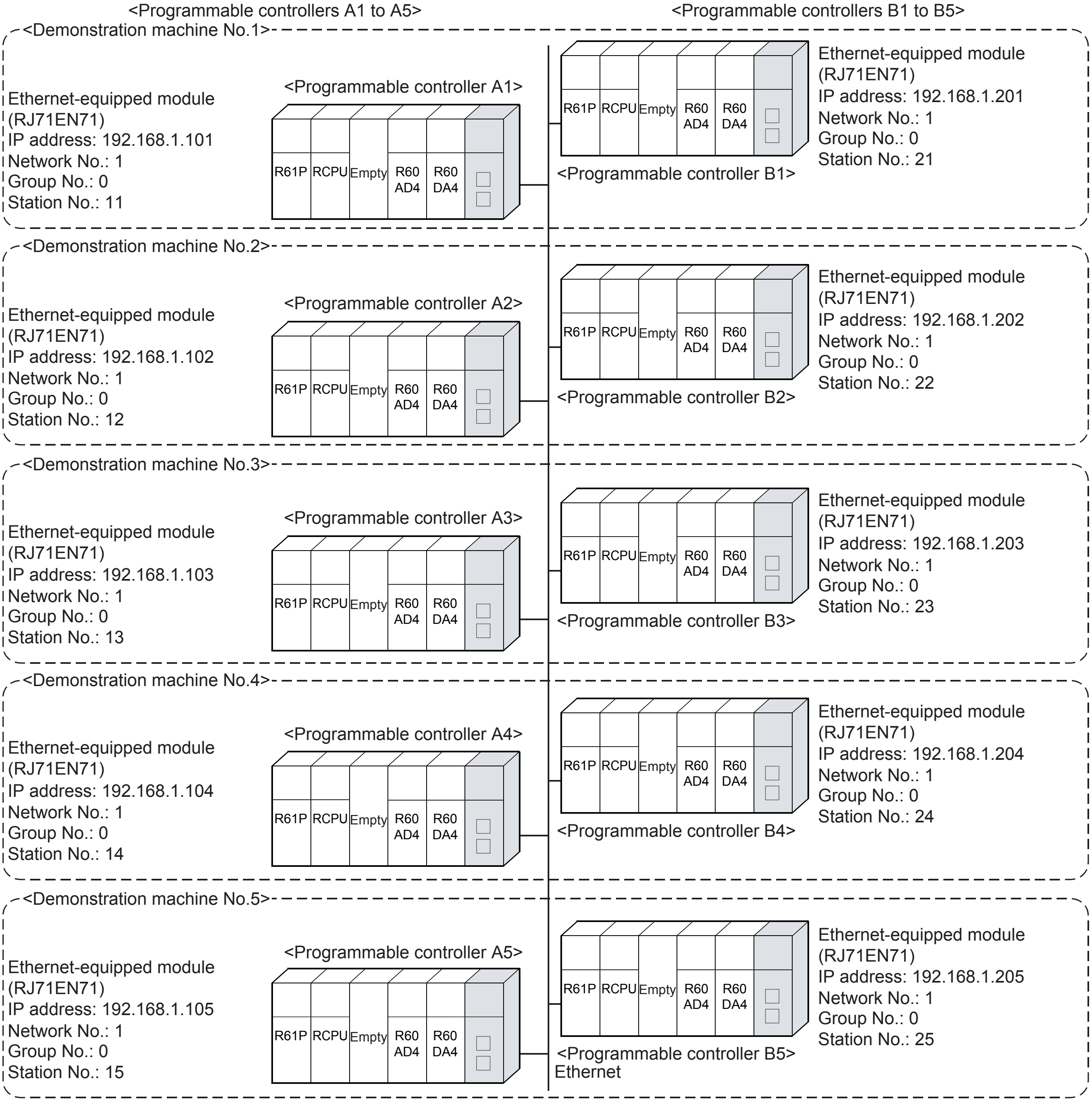
7 EXERCISE 3 (COMMUNICATIONS USING A FIXED BUFFER (NO PROCEDURE) BETWEEN CPU MODULES)

This exercise is for communications using a fixed buffer (no procedure).
The programmable controllers A and B send and receive data.

7.1 System Configuration

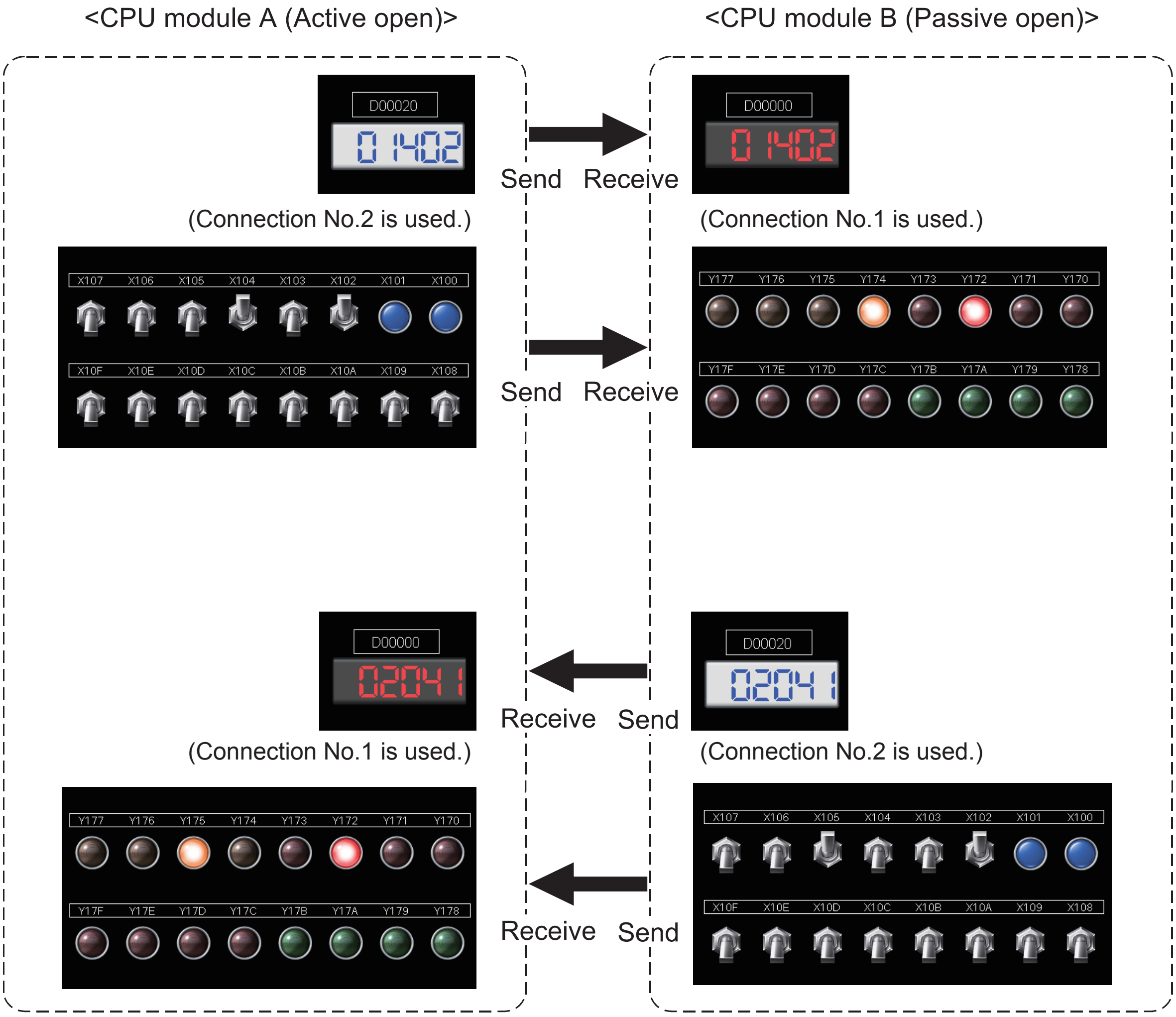
The following figure shows the system configuration of Exercise 3.
In Exercise 3, perform data communications using TCP/IP and monitor the device (X) of the CPU modules on the sending side with the demonstration machines on the receiving side.
Although 10 CPU modules are connected on the same Ethernet network, perform communications only between the CPU modules with the same demonstration machine number in Exercise 3.

Module configuration



Exercise

Display the information of X100 to X107 and D20 of the CPU module A on Y170 to Y177 and D0 of the CPU module B.
Display the information of X100 to X107 and D20 of the CPU module B on Y170 to Y177 and D0 of the CPU module A.


<CPU module A (Active open)>
<CPU module B (Passive open)>
D00020
01402
Send
Receive
D00000
01402
(Connection No.2 is used.)
(Connection No.1 is used.)
X107 X106 X105 X104 X103 X102 X101 X100
X10F X10E X10D X10C X10B X10A X109 X108
Send
Receive
Y177 Y176 Y175 Y174 Y173 Y172 Y171 Y170
Y17F Y17E Y17D Y17C Y17B Y17A Y179 Y178
D00000
02041
Receive
Send
D00020
02041
(Connection No.1 is used.)
(Connection No.2 is used.)
Y177 Y176 Y175 Y174 Y173 Y172 Y171 Y170
Y17F Y17E Y17D Y17C Y17B Y17A Y179 Y178
Receive
Send
X107 X106 X105 X104 X103 X102 X101 X100
X10F X10E X10D X10C X10B X10A X109 X108

7.2 Setting the Ethernet-equipped Module (Programmable Controllers A1 to A5)

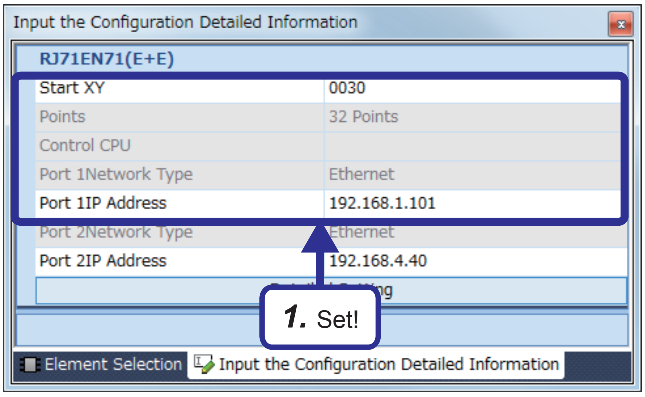
Set the Ethernet-equipped module (programmable controllers A1 to A5).

7.2.1 Setting parameters with GX Works3

Adding Ethernet-equipped module data

Set parameters of the Ethernet-equipped module with GX Works3.
Set them as described on Page 7 - 1 System Configuration.

Operating procedure

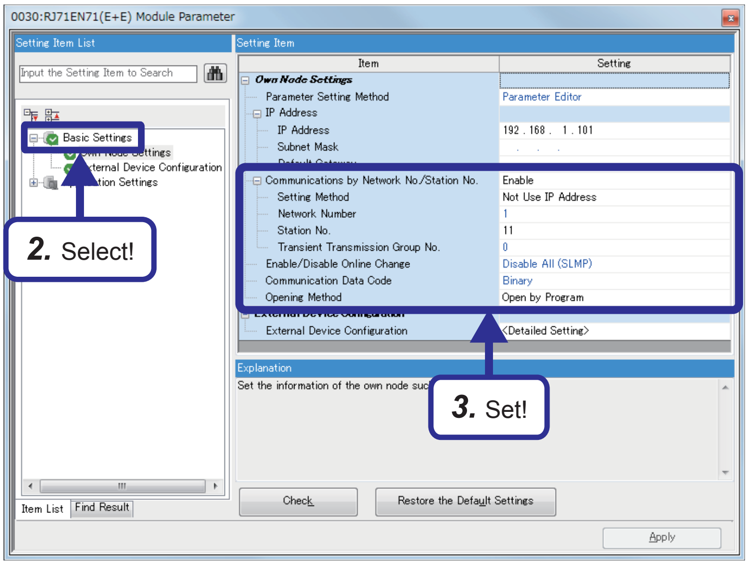


Input the Configuration Detailed Information

RJ71EN71(E+E)	
Start XY	0030
Points	32 Points
Control CPU	
Port 1Network Type	Ethernet
Port 1IP Address	192.168.1.101
Port 2Network Type	Ethernet
Port 2IP Address	192.168.4.40

1. Create a new project and add "RJ71EN71(E+E)" from "Information Module" in the "Element Selection" window to the slot No.3 of the R35B by following the procedure described on Page 5 - 8 Adding Ethernet-equipped module data. After setting the parameters as follows, fix them.
(Click the [OK] button when the confirmation window for adding the module appears.)

[Parameters to be set]
Start XY: 0030
Port 1IP address: Set the IP address (192.168.1.101 to 192.168.1.105) of the demonstration machine used.



0030:RJ71EN71(E+E) Module Parameter

Item	Setting
Own Node Settings	
Parameter Setting Method	Parameter Editor
IP Address	
IP Address	192 . 168 . 1 . 101
Subnet Mask	. . .
Communications by Network No./Station No.	Enable
Setting Method	Not Use IP Address
Network Number	1
Station No.	11
Transient Transmission Group No.	0
Enable/Disable Online Change	Disable All (SLMP)
Communication Data Code	Binary
Opening Method	Open by Program
External Device Configuration	<Detailed Setting>

Explanation: Set the information of the own node suc

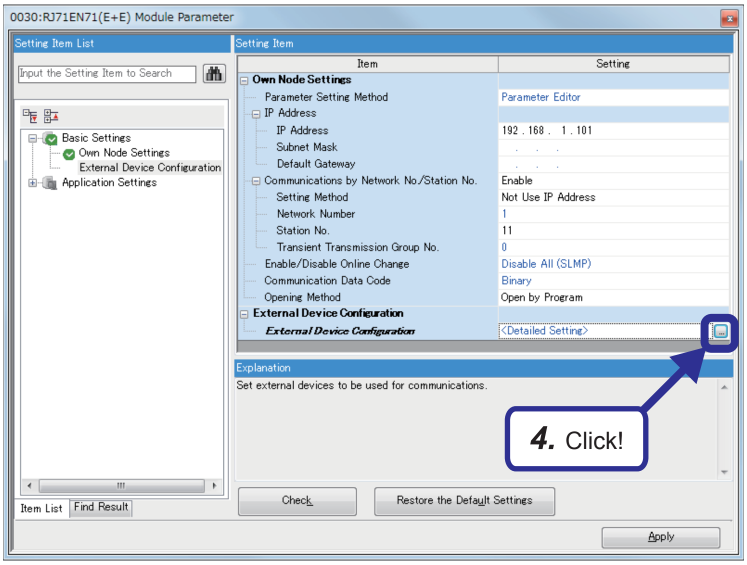
2. Double-click the RJ71EN71(E+E) module in the "Module Configuration" window and select "Basic Settings" in "Setting Item List".

3. Set "Communications by Network No./Station No." as follows.

[Parameters to be set]
Communications by Network No./Station No.: Enable
Setting Method: Not Use IP Address
Network Number: 1
Station No.: Set the station number (11 to 15) of the demonstration machine used.
Communication Data Code: Optional (The binary code is used for the communications regardless of the setting.)
Opening Method: Open by Program

(To the next page)

(From the previous page)

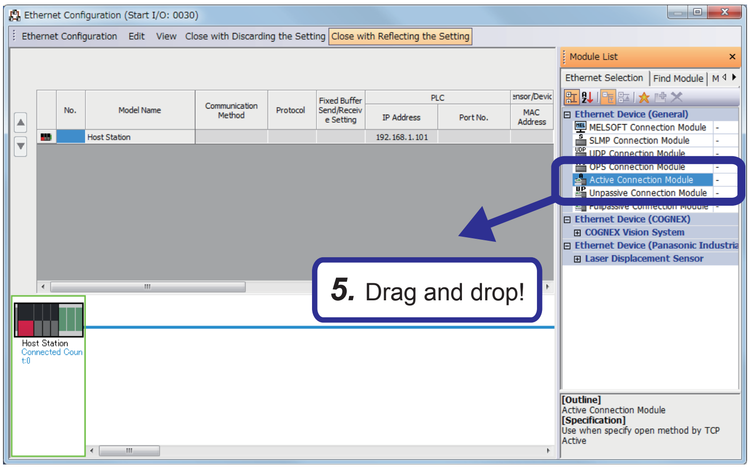

0030:RJ71EN71(E+E) Module Parameter
Setting Item List
Input the Setting Item to Search
Basic Settings
Own Node Settings
External Device Configuration
Application Settings
Item List
Find Result
Setting Item
Explanation
Check
Restore the Default Settings
Apply

Item	Setting
Own Node Settings	
Parameter Setting Method	Parameter Editor
IP Address	
IP Address	192 . 168 . 1 . 101
Subnet Mask	. . .
Default Gateway	. . .
Communications by Network No./Station No.	Enable
Setting Method	Not Use IP Address
Network Number	1
Station No.	11
Transient Transmission Group No.	0
Enable/Disable Online Change	Disable All (SLMP)
Communication Data Code	Binary
Opening Method	Open by Program
External Device Configuration	
External Device Configuration	<Detailed Setting>

Set external devices to be used for communications.

4. Click!

4. Click the ▭ button of "Detailed Setting" of "External Device Configuration".


Ethernet Configuration (Start I/O: 0030)
Ethernet Configuration
Edit
View
Close with Discarding the Setting
Close with Reflecting the Setting
Module List
Ethernet Selection
Find Module
M
Ethernet Device (General)
MELSOFT Connection Module
SLMP Connection Module
UDP Connection Module
OPS Connection Module
Active Connection Module
Unpassive Connection Module
Fullpassive Connection Module
Ethernet Device (COGNEX)
COGNEX Vision System
Ethernet Device (Panasonic Industria
Laser Displacement Sensor
Host Station
Connected Count:0

	No.	Model Name	Communication Method	Protocol	Fixed Buffer Send/Receive Setting	PLC IP Address	PLC Port No.	Sensor/Device MAC Address
		Host Station				192. 168. 1. 101		

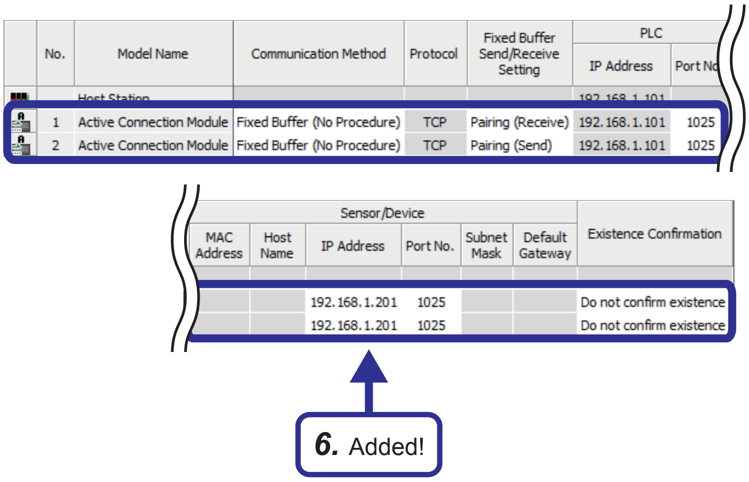
[Outline]
Active Connection Module
[Specification]
Use when specify open method by TCP Active

5. Drag and drop!

5. The "Ethernet Configuration" dialog box appears. Select "Active Connection Module" from "Ethernet Device (General)" in "Module List", and drag and drop it to the list of devices or device map area.
Select "Active Connection Module" again, and drag and drop it to the list of devices or device map area.

(To the next page)

(From the previous page)



	No.	Model Name	Communication Method	Protocol	Fixed Buffer Send/Receive Setting	PLC	
						IP Address	Port No.
		Host Station				192.168.1.101	
	1	Active Connection Module	Fixed Buffer (No Procedure)	TCP	Pairing (Receive)	192.168.1.101	1025
	2	Active Connection Module	Fixed Buffer (No Procedure)	TCP	Pairing (Send)	192.168.1.101	1025

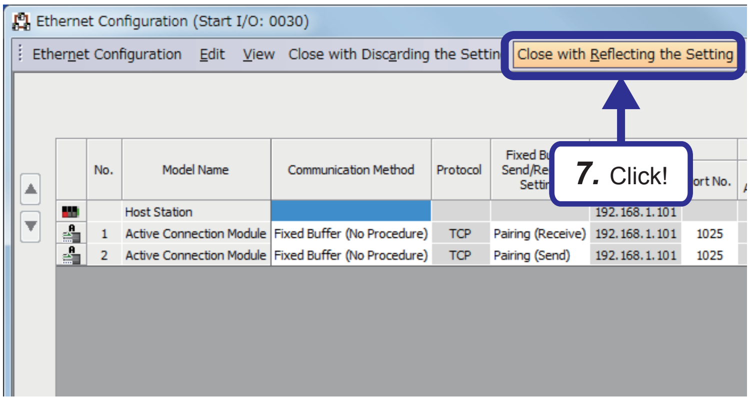
Sensor/Device						Existence Confirmation
MAC Address	Host Name	IP Address	Port No.	Subnet Mask	Default Gateway	
		192.168.1.201	1025			Do not confirm existence
		192.168.1.201	1025			Do not confirm existence

6. Added!

6. "Active Connection Module" is added to the list of devices.
Set "Active Connection Module" as follows.

[Parameters to be set: 1]
Communication Method: Fixed Buffer (No Procedure)
Fixed Buffer Send/Receive Setting: Pairing (Receive)
Port No. (PLC): 1025
IP Address: Set the IP address (192.168.1.201 to 192.168.1.205) of the demonstration machine used.
Port No. (Sensor/Device): 1025
Existence Confirmation: Do not confirm existence

[Parameters to be set: 2]
Communication Method: Fixed Buffer (No Procedure)
Fixed Buffer Send/Receive Setting: Pairing (Send)
Port No. (PLC): 1025
IP Address: Set the IP address (192.168.1.201 to 192.168.1.205) of the demonstration machine used.
Port No. (Sensor/Device): 1025
Existence Confirmation: Do not confirm existence


Ethernet Configuration (Start I/O: 0030)
Ethernet Configuration
Edit
View
Close with Discarding the Settin
Close with Reflecting the Setting

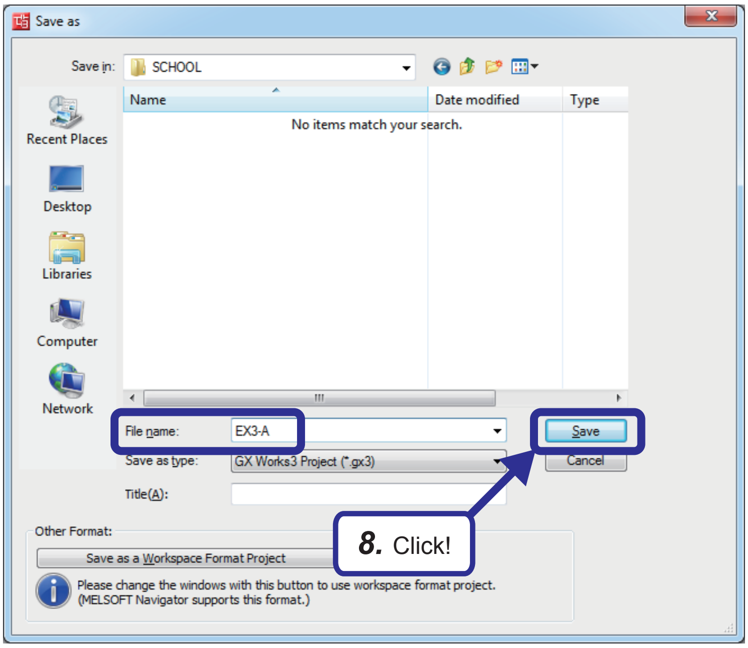
	No.	Model Name	Communication Method	Protocol	Fixed Buffer Send/Receive Setting	IP Address	Port No.
		Host Station				192.168.1.101	
	1	Active Connection Module	Fixed Buffer (No Procedure)	TCP	Pairing (Receive)	192.168.1.101	1025
	2	Active Connection Module	Fixed Buffer (No Procedure)	TCP	Pairing (Send)	192.168.1.101	1025

7. Click!

7. After setting the parameters, click "Close with Reflecting the Setting" in the menu to close the "Ethernet Configuration" dialog box.

(To the next page)

(From the previous page)


Save as
Save in: SCHOOL
Name
Date modified
Type
No items match your search.
Recent Places
Desktop
Libraries
Computer
Network
File name: EX3-A
Save as type: GX Works3 Project (*.gx3)
Title(A):
Save
Cancel
Other Format:
Save as a Workspace Format Project
Please change the windows with this button to use workspace format project.
(MELSOFT Navigator supports this format.)
8. Click!

8. Click [Project] ⇨ [Save As] from the menu to save the project.

Save destination: Desired location
File name: EX3-A
Title: Blank

7.2.2 Sequence program

Create a sequence program.

Open the connections No.1 and No.2 (specify the paring open) by setting "Fixed Buffer Send/Receive Setting" in the "Ethernet Configuration" dialog box, and send and receive data.

After sending/receiving the data, close the connections No.1 and No.2.

Point

When communications are performed with no procedure, the data length is specified in bytes.

When a procedure exists, the data length is specified in words.

After creating the sequence program, perform the following operations with GX Works3.

- Saving the project
  Project name: EX3-A
- Specifying a connection destination (Refer to Page 5 - 15 Specifying a connection destination.)
- Writing data to the CPU module (Refer to Page 5 - 17 Writing parameters.)

List of devices used

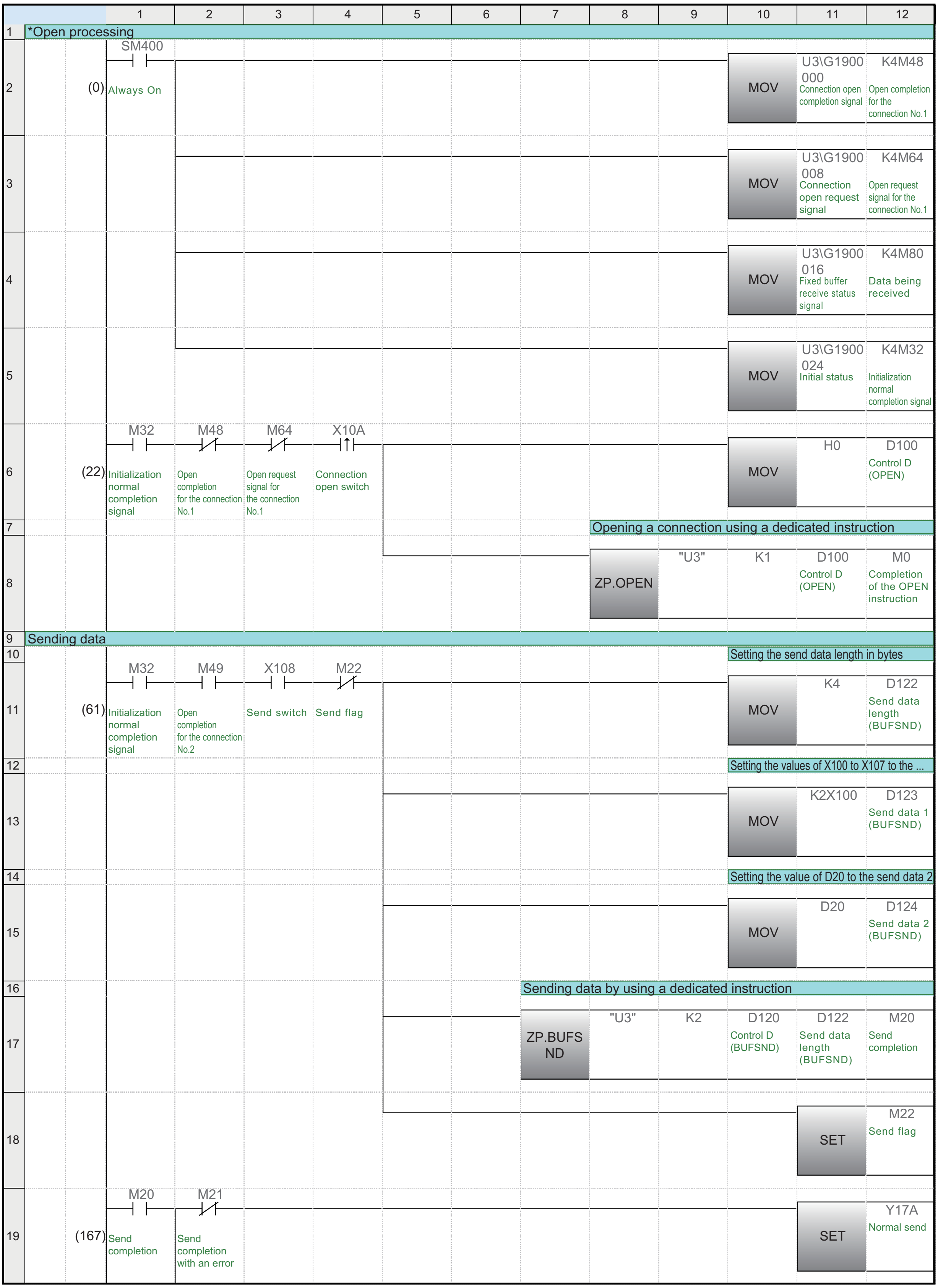
Device name	Description	Device name	Description
SM400	Always On	X10A	A command switch to open a connection
M0	Turns on only for one scan after the open processing for the connection No.1 has completed.	X10B	A command switch to close a connection
M1	Turns on only for one scan after the open processing for the connection No.1 has completed with an error.	Y170 to Y177	Store the information of X100 to X107 sent from the CPU module B.
M10	Turns on only for one scan after the close processing for the connection No.1 has completed.	Y178	Turns on after the receive processing has completed successfully.
M11	Turns on only for one scan after the close processing for the connection No.1 has completed with an error.	Y179	Turns on after the receive processing has completed with an error.
M20	Turns on only for one scan after the send processing has completed.	Y17A	Turns on after the send processing has completed successfully.
M21	Turns on only for one scan after the send processing has completed with an error.	Y17B	Turns on after the send processing has completed with an error.
M22	Turns on when data is being sent. Turns off after the send processing has completed.	D0	Stores the information of D20 sent from the CPU module B.
M30	Turns on only for one scan after the receive processing has completed.	D20	Data to be sent to the CPU module B
M31	Turns on only for one scan after the receive processing has completed with an error.	D100 to D109	Store the control data of the OPEN instruction.
M32	Turns on after the initial processing of the Ethernet-equipped module has completed successfully.	D110 and D111	Store the control data of the CLOSE instruction.
M48	Turns on when the connection No.1 is open.	D120 and D121	Store the control data of the BUFSND instruction.
M49	Turns on when the connection No.2 is open.	D122	Stores the send data length.
M64	Turns on when the open request for the connection No.1 is being issued or the connection No.1 is open.	D123	Stores send data (X100 to X107).
M65	Turns on when the open request for the connection No.2 is being issued or the connection No.2 is open.	D124	Stores send data (D20).
M80	Turns on while the connection No.1 of the Ethernet-equipped module is receiving data.	D130 and D131	Store the control data of the BUFRCV instruction.
M100	Turns on only for one scan after an error has cleared.	D132	Stores the receive data length.
M110	Turns on when an error is cleared. Turns off when an error has cleared.	D133	Stores receive data (X100 to X107).
X100 to X107	Data to be sent to the CPU module B	D134	Stores receive data (D20).
X108	A command switch to send data	D140 to D144	Store the control data of the ERRCLEAR instruction.
X109	A command switch to clear an error display of the indicator LED	—	—

Sequence program

For details on the dedicated instructions, refer to the following.

MELSEC iQ-R Programming Manual (Instructions, Standard Functions/Function Blocks)

Project name	EX3-A



1 *Open processing

2 (0) SM400 Always On — MOV U3\G1900000 (Connection open completion signal) K4M48 (Open completion for the connection No.1)

3 MOV U3\G1900008 (Connection open request signal) K4M64 (Open request signal for the connection No.1)

4 MOV U3\G1900016 (Fixed buffer receive status signal) K4M80 (Data being received)

5 MOV U3\G1900024 (Initial status) K4M32 (Initialization normal completion signal)

6 (22) M32 (Initialization normal completion signal) M48 (Open completion for the connection No.1) M64 (Open request signal for the connection No.1) X10A (Connection open switch) — MOV H0 D100 (Control D (OPEN))

7 Opening a connection using a dedicated instruction

8 ZP.OPEN "U3" K1 D100 (Control D (OPEN)) M0 (Completion of the OPEN instruction)

9 Sending data

10 Setting the send data length in bytes

11 (61) M32 (Initialization normal completion signal) M49 (Open completion for the connection No.2) X108 (Send switch) M22 (Send flag) — MOV K4 D122 (Send data length (BUFSND))

12 Setting the values of X100 to X107 to the ...

13 MOV K2X100 D123 (Send data 1 (BUFSND))

14 Setting the value of D20 to the send data 2

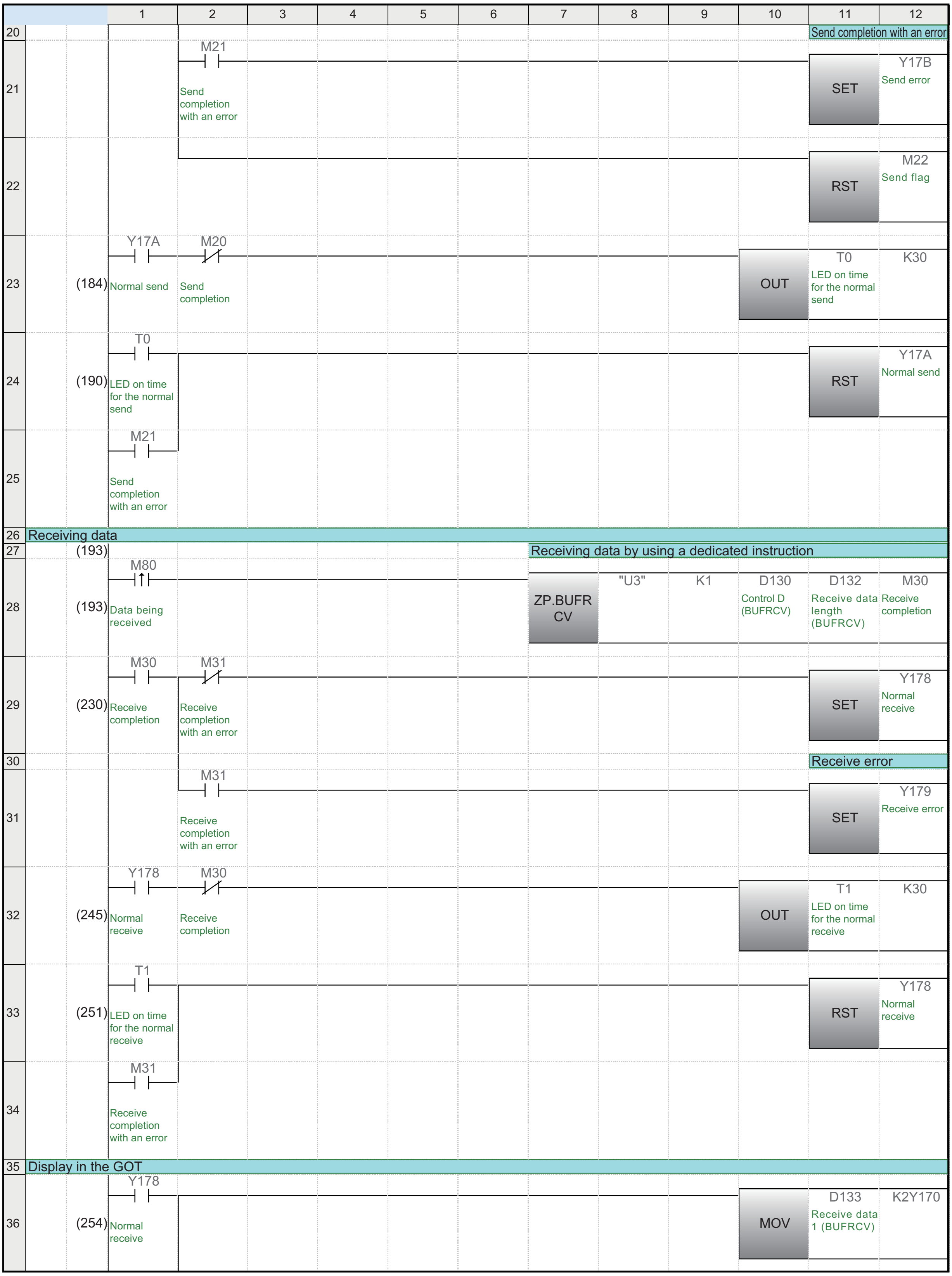
15 MOV D20 D124 (Send data 2 (BUFSND))

16 Sending data by using a dedicated instruction

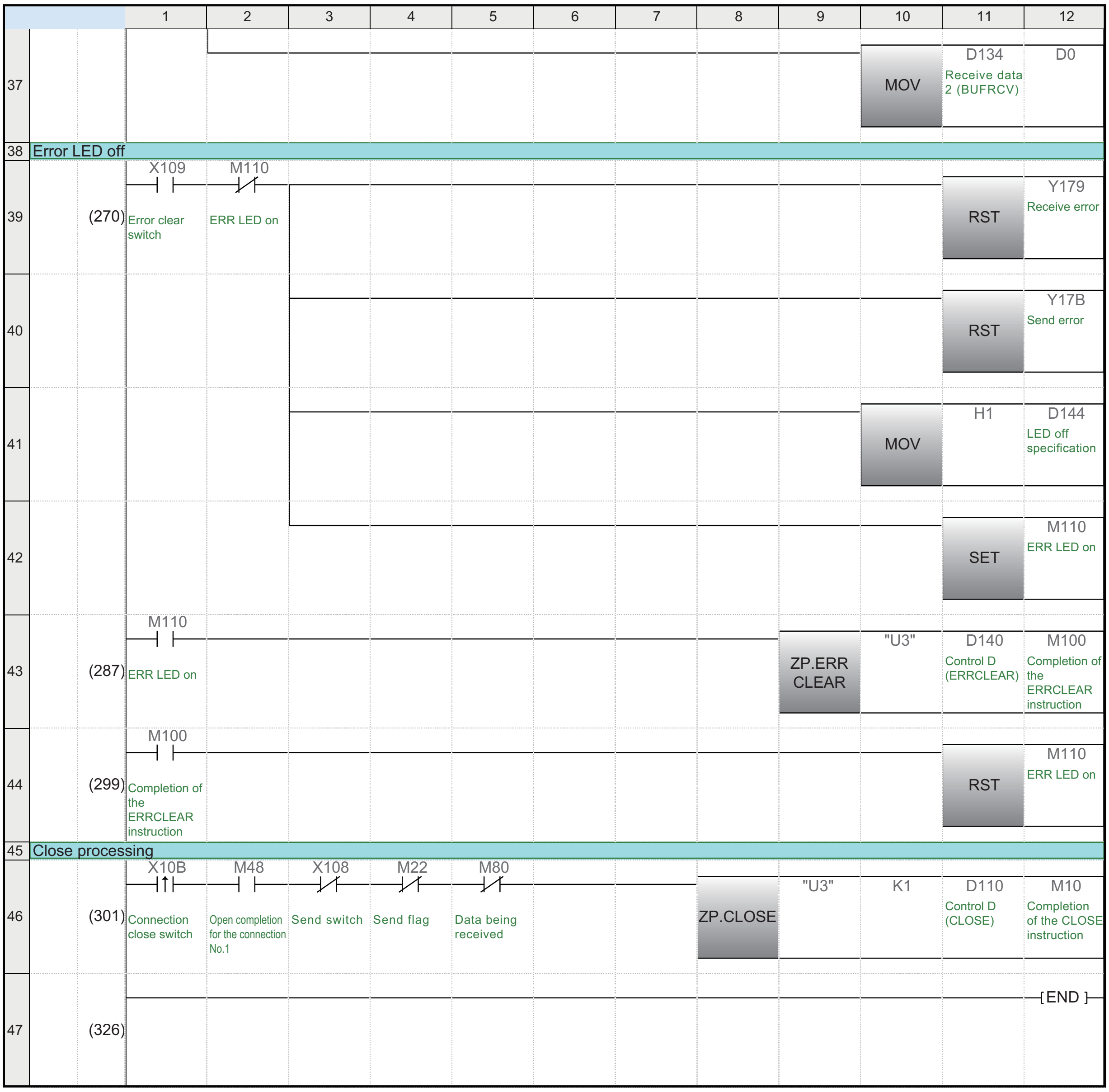
17 ZP.BUFSND "U3" K2 D120 (Control D (BUFSND)) D122 (Send data length (BUFSND)) M20 (Send completion)

18 SET M22 (Send flag)

19 (167) M20 (Send completion) M21 (Send completion with an error) — SET Y17A (Normal send)



		1	2	3	4	5	6	7	8	9	10	11	12
20												Send completion with an error	
21			M21 Send completion with an error									SET	Y17B Send error
22												RST	M22 Send flag
23	(184)	Y17A Normal send	M20 Send completion								OUT	T0 LED on time for the normal send	K30
24	(190)	T0 LED on time for the normal send										RST	Y17A Normal send
25		M21 Send completion with an error											
26	Receiving data												
27	(193)							Receiving data by using a dedicated instruction					
28	(193)	M80 Data being received						ZP.BUFRCV	"U3"	K1	D130 Control D (BUFRCV)	D132 Receive data length (BUFRCV)	M30 Receive completion
29	(230)	M30 Receive completion	M31 Receive completion with an error									SET	Y178 Normal receive
30												Receive error	
31			M31 Receive completion with an error									SET	Y179 Receive error
32	(245)	Y178 Normal receive	M30 Receive completion								OUT	T1 LED on time for the normal receive	K30
33	(251)	T1 LED on time for the normal receive										RST	Y178 Normal receive
34		M31 Receive completion with an error											
35	Display in the GOT												
36	(254)	Y178 Normal receive									MOV	D133 Receive data 1 (BUFRCV)	K2Y170


1
2
3
4
5
6
7
8
9
10
11
12
37
MOV
D134
Receive data 2 (BUFRCV)
D0
38 Error LED off
39
(270)
X109
Error clear switch
M110
ERR LED on
RST
Y179
Receive error
40
RST
Y17B
Send error
41
MOV
H1
D144
LED off specification
42
SET
M110
ERR LED on
43
(287)
M110
ERR LED on
ZP.ERR CLEAR
"U3"
D140
Control D (ERRCLEAR)
M100
Completion of the ERRCLEAR instruction
44
(299)
M100
Completion of the ERRCLEAR instruction
RST
M110
ERR LED on
45 Close processing
46
(301)
X10B
Connection close switch
M48
Open completion for the connection No.1
X108
Send switch
M22
Send flag
M80
Data being received
ZP.CLOSE
"U3"
K1
D110
Control D (CLOSE)
M10
Completion of the CLOSE instruction
47
(326)
END

7.3 Setting the Ethernet-equipped Module (Programmable Controllers B1 to B5)

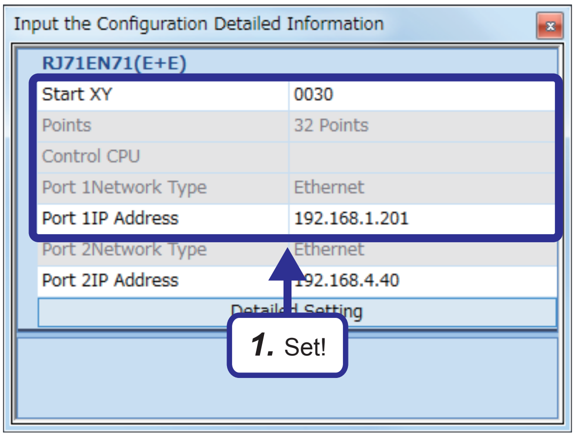
Set the Ethernet-equipped module (programmable controllers B1 to B5).

7.3.1 Setting parameters with GX Works3

Adding Ethernet-equipped module data

Set parameters of the Ethernet-equipped module with GX Works3.
Set them as described on Page 7 - 1 System Configuration.

Operating procedure



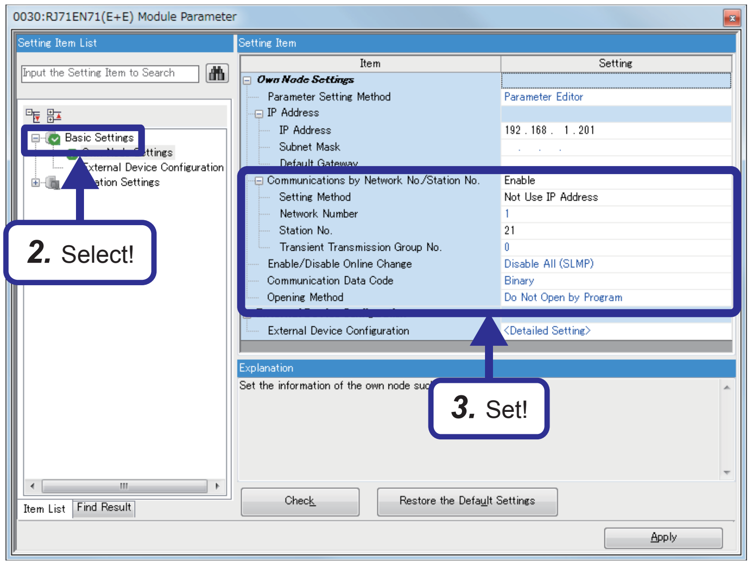
Input the Configuration Detailed Information

RJ71EN71(E+E)

Start XY	0030
Points	32 Points
Control CPU	
Port 1Network Type	Ethernet
Port 1IP Address	192.168.1.201
Port 2Network Type	Ethernet
Port 2IP Address	192.168.4.40

1. Create a new project and add "RJ71EN71(E+E)" from "Information Module" in the "Element Selection" window to the slot No.3 of the R35B by following the procedure described on Page 5 - 8 Adding Ethernet-equipped module data. After setting the parameters as follows, fix them.
(Click the [OK] button when the confirmation window for adding the module appears.)

[Parameters to be set]
Start XY: 0030
Port 1IP address: Set the IP address (192.168.1.201 to 192.168.1.205) of the demonstration machine used.



0030:RJ71EN71(E+E) Module Parameter

Setting Item

Item	Setting
Own Node Settings	
Parameter Setting Method	Parameter Editor
IP Address	
IP Address	192 . 168 . 1 . 201
Subnet Mask	. . .
Default Gateway	
Communications by Network No./Station No.	Enable
Setting Method	Not Use IP Address
Network Number	1
Station No.	21
Transient Transmission Group No.	0
Enable/Disable Online Change	Disable All (SLMP)
Communication Data Code	Binary
Opening Method	Do Not Open by Program
External Device Configuration	
External Device Configuration	<Detailed Setting>

Explanation
Set the information of the own node suc

2. Double-click the RJ71EN71(E+E) module in the "Module Configuration" window and select "Basic Settings" in "Setting Item List".

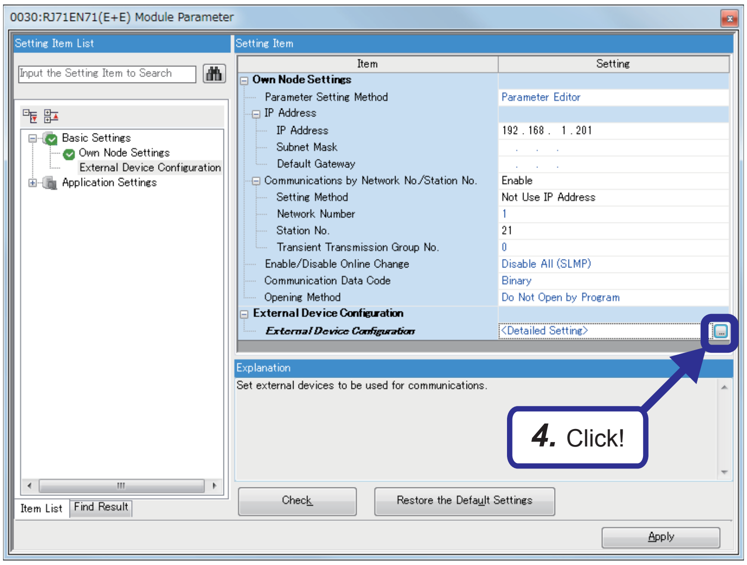
3. Set "Communications by Network No./Station No." as follows.

[Parameters to be set]
Communications by Network No./Station No.: Enable
Setting Method: Not Use IP Address
Network Number: 1
Station No.: Set the station number (21 to 25) of the demonstration machine used.
Communication Data Code: Optional (The binary code is used for the communications regardless of the setting.)
Opening Method: Do Not Open by Program

(To the next page)

(From the previous page)

4. Click the ▢ button of "Detailed Setting" of "External Device Configuration".



0030:RJ71EN71(E+E) Module Parameter

Setting Item List

Input the Setting Item to Search

- Basic Settings
  - Own Node Settings
  - External Device Configuration
- Application Settings

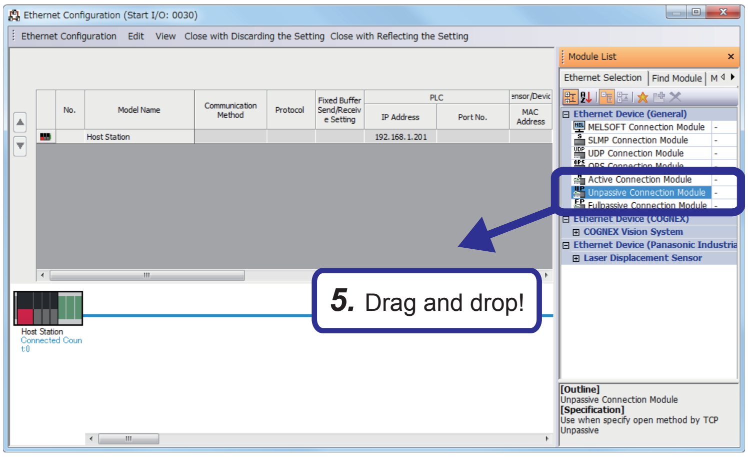
Setting Item

Item	Setting
Own Node Settings	
Parameter Setting Method	Parameter Editor
IP Address	
IP Address	192 . 168 . 1 . 201
Subnet Mask	. . .
Default Gateway	. . .
Communications by Network No./Station No.	Enable
Setting Method	Not Use IP Address
Network Number	1
Station No.	21
Transient Transmission Group No.	0
Enable/Disable Online Change	Disable All (SLMP)
Communication Data Code	Binary
Opening Method	Do Not Open by Program
External Device Configuration	
External Device Configuration	<Detailed Setting>

Explanation

Set external devices to be used for communications.

5. The "Ethernet Configuration" dialog box appears. Select "Unpassive Connection Module" from "Ethernet Device (General)" in "Module List" and drag and drop it to the list of devices or device map area.
Select "Unpassive Connection Module" again, and drag and drop it to the list of devices or device map area.



Ethernet Configuration (Start I/O: 0030)

Ethernet Configuration
Edit
View
Close with Discarding the Setting
Close with Reflecting the Setting

	No.	Model Name	Communication Method	Protocol	Fixed Buffer Send/Receive Setting	PLC		ensor/Devic
						IP Address	Port No.	MAC Address
		Host Station				192.168.1.201		

Host Station
Connected Count:0

Module List

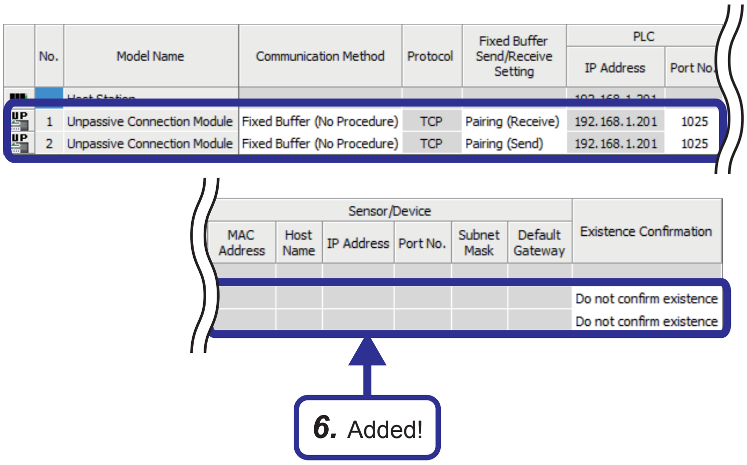
Ethernet Selection
Find Module
M

- Ethernet Device (General)
  - MELSOFT Connection Module -
  - SLMP Connection Module -
  - UDP Connection Module -
  - OPS Connection Module -
  - Active Connection Module -
  - Unpassive Connection Module -
  - Fullpassive Connection Module -
- Ethernet Device (COGNEX)
  - COGNEX Vision System
- Ethernet Device (Panasonic Industria
  - Laser Displacement Sensor

[Outline]
Unpassive Connection Module
[Specification]
Use when specify open method by TCP Unpassive

(To the next page)

(From the previous page)



	No.	Model Name	Communication Method	Protocol	Fixed Buffer Send/Receive Setting	PLC IP Address	PLC Port No.
		Host Station				192.168.1.201	
UP	1	Unpassive Connection Module	Fixed Buffer (No Procedure)	TCP	Pairing (Receive)	192.168.1.201	1025
UP	2	Unpassive Connection Module	Fixed Buffer (No Procedure)	TCP	Pairing (Send)	192.168.1.201	1025

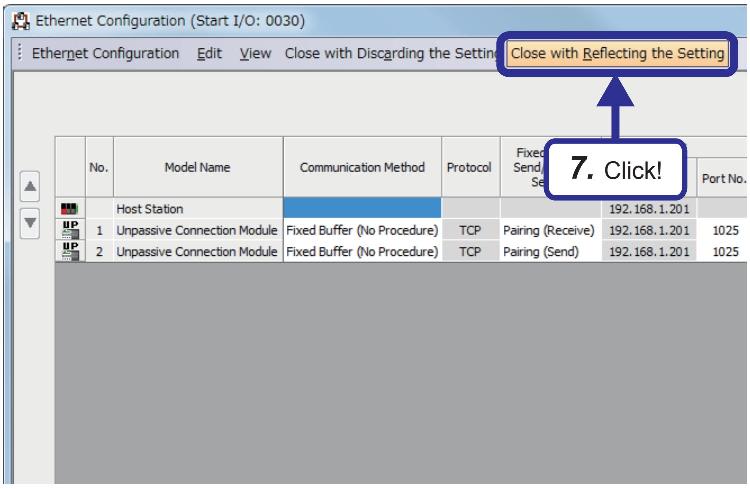
Sensor/Device MAC Address	Host Name	IP Address	Port No.	Subnet Mask	Default Gateway	Existence Confirmation
						Do not confirm existence
						Do not confirm existence

6. Added!

6. "Unpassive Connection Module" is added to the list of devices.
Set "Unpassive Connection Module" as follows.

[Parameters to be set: 1]
Communication Method: Fixed Buffer (No Procedure)
Fixed Buffer Send/Receive Setting: Pairing (Receive)
Port No.: 1025
Existence Confirmation: Do not confirm existence

[Parameters to be set: 2]
Communication Method: Fixed Buffer (No Procedure)
Fixed Buffer Send/Receive Setting: Pairing (Send)
Port No.: 1025
Existence Confirmation: Do not confirm existence


Ethernet Configuration (Start I/O: 0030)
Ethernet Configuration
Edit
View
Close with Discarding the Setting
Close with Reflecting the Setting

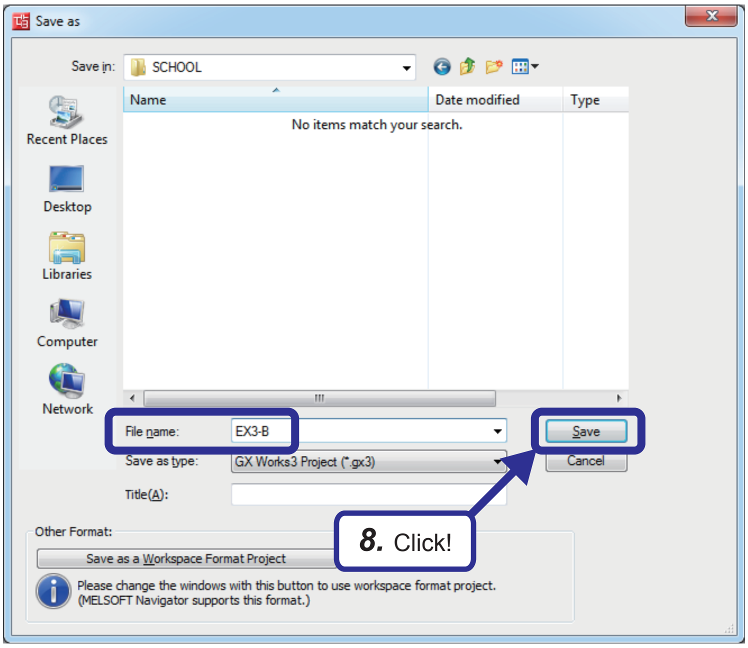
	No.	Model Name	Communication Method	Protocol	Fixed Buffer Send/Receive Setting	IP Address	Port No.
		Host Station				192.168.1.201	
UP	1	Unpassive Connection Module	Fixed Buffer (No Procedure)	TCP	Pairing (Receive)	192.168.1.201	1025
UP	2	Unpassive Connection Module	Fixed Buffer (No Procedure)	TCP	Pairing (Send)	192.168.1.201	1025

7. Click!

7. After setting the parameters, click "Close with Reflecting the Setting" in the menu to close the "Ethernet Configuration" dialog box.

(To the next page)

(From the previous page)



File name: EX3-B

Save as type: GX Works3 Project (*.gx3)

Title(A):

Please change the windows with this button to use workspace format project. (MELSOFT Navigator supports this format.)

8. Click [Project] ⇨ [Save As] from the menu to save the project.

Save destination: Desired location
File name: EX3-B
Title: Blank

7.3.2 Sequence program

Create a sequence program.

Establish a connection with Active open on the side of the CPU modules A1 to A5, and send and receive data.

Point

When communications are performed with no procedure, the data length is specified in bytes.

When a procedure exists, the data length is specified in words.

After creating the sequence program, perform the following operations with GX Works3.

- Saving the project
  Project name: EX3-B
- Specifying a connection destination (Refer to Page 5 - 15 Specifying a connection destination.)
- Writing data to the CPU module (Refer to Page 5 - 17 Writing parameters.)

List of devices used

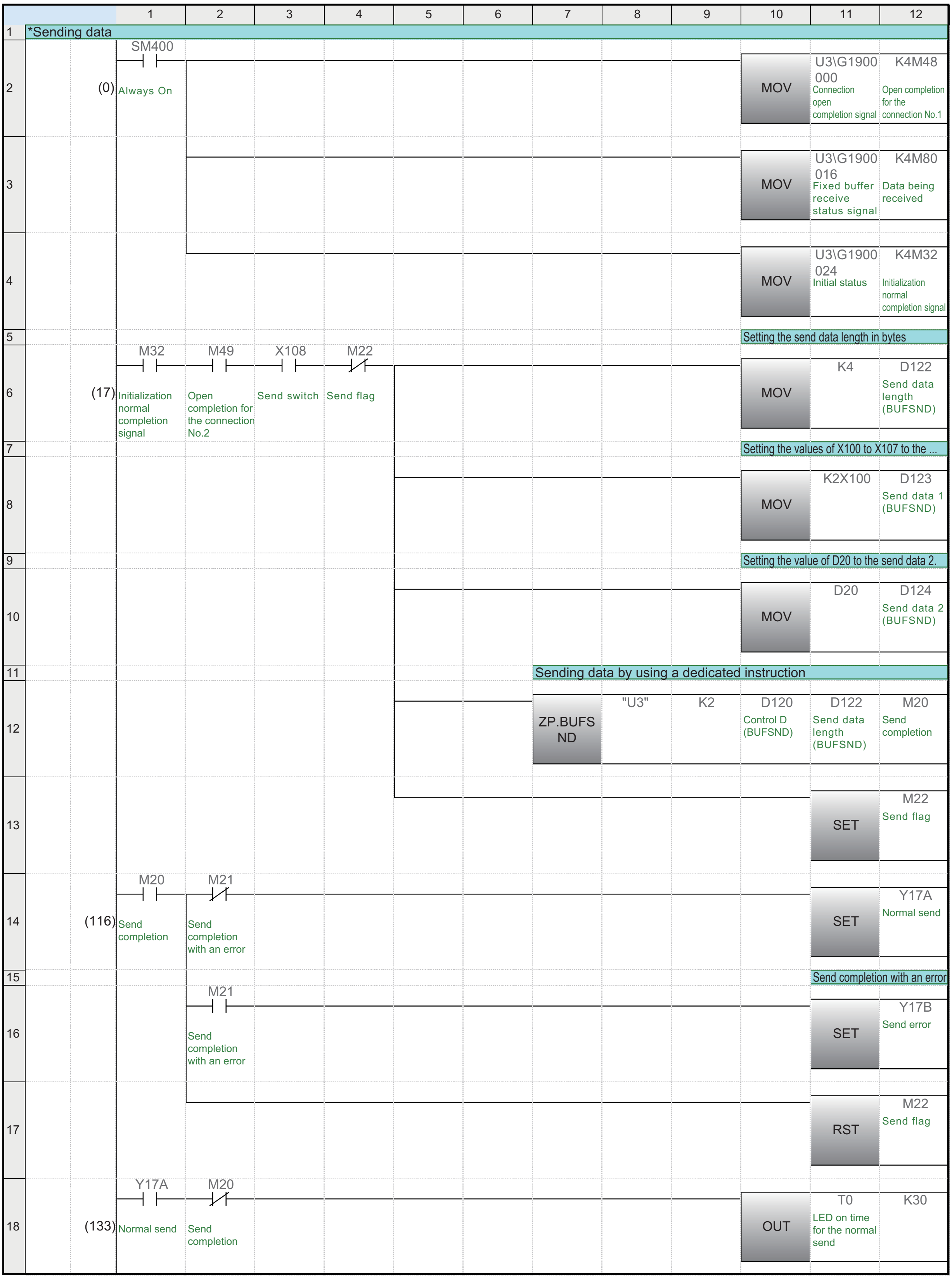
Device name	Description	Device name	Description
SM400	Always On	Y178	Turns on after the receive processing has completed successfully.
M20	Turns on only for one scan after the send processing has completed.	Y179	Turns on after the receive processing has completed with an error.
M21	Turns on only for one scan after the send processing has completed with an error.	Y17A	Turns on after the send processing has completed successfully.
M22	Turns on when data is being sent. Turns off after the send processing has completed.	Y17B	Turns on after the send processing has completed with an error.
M30	Turns on only for one scan after the receive processing has completed.	D0	Stores the information of D20 sent from the CPU module A.
M31	Turns on only for one scan after the receive processing has completed with an error.	D20	Data to be sent to the CPU module A
M32	Turns on after the initial processing of the Ethernet-equipped module has completed successfully.	D120 and D121	Store the control data of the BUFSND instruction.
M48	Turns on when the connection No.1 is open.	D122	Stores the send data length.
M80	Turns on while the connection No.1 of the Ethernet-equipped module is receiving data.	D123	Stores send data (X100 to X107).
M100	Turns on only for one scan after an error has cleared.	D124	Stores send data (D20).
M110	Turns on when an error is cleared. Turns off when an error has cleared.	D130 and D131	Store the control data of the BUFRCV instruction.
X100 to X107	Data to be sent to the CPU module A	D132	Stores the receive data length.
X108	A command switch to send data	D133	Stores receive data (X100 to X107).
X109	A command switch to clear an error display of the indicator LED	D134	Stores receive data (D20).
Y170 to Y177	Store the information of X100 to X107 sent from the CPU module A.	D140 to D144	Store the control data of the ERRCLEAR instruction.

Sequence program

For details on the dedicated instructions, refer to the following.

📖 MELSEC iQ-R Programming Manual (Instructions, Standard Functions/Function Blocks)

Project name	EX3-B



1 *Sending data

2 (0) SM400 Always On — MOV U3\G1900000 Connection open completion signal, K4M48 Open completion for the connection No.1

3 MOV U3\G1900016 Fixed buffer receive status signal, K4M80 Data being received

4 MOV U3\G1900024 Initial status, K4M32 Initialization normal completion signal

5 Setting the send data length in bytes

6 (17) M32 Initialization normal completion signal, M49 Open completion for the connection No.2, X108 Send switch, M22 Send flag — MOV K4 D122 Send data length (BUFSND)

7 Setting the values of X100 to X107 to the ...

8 MOV K2X100 D123 Send data 1 (BUFSND)

9 Setting the value of D20 to the send data 2.

10 MOV D20 D124 Send data 2 (BUFSND)

11 Sending data by using a dedicated instruction

12 ZP.BUFSND "U3" K2 D120 Control D (BUFSND) D122 Send data length (BUFSND) M20 Send completion

13 SET M22 Send flag

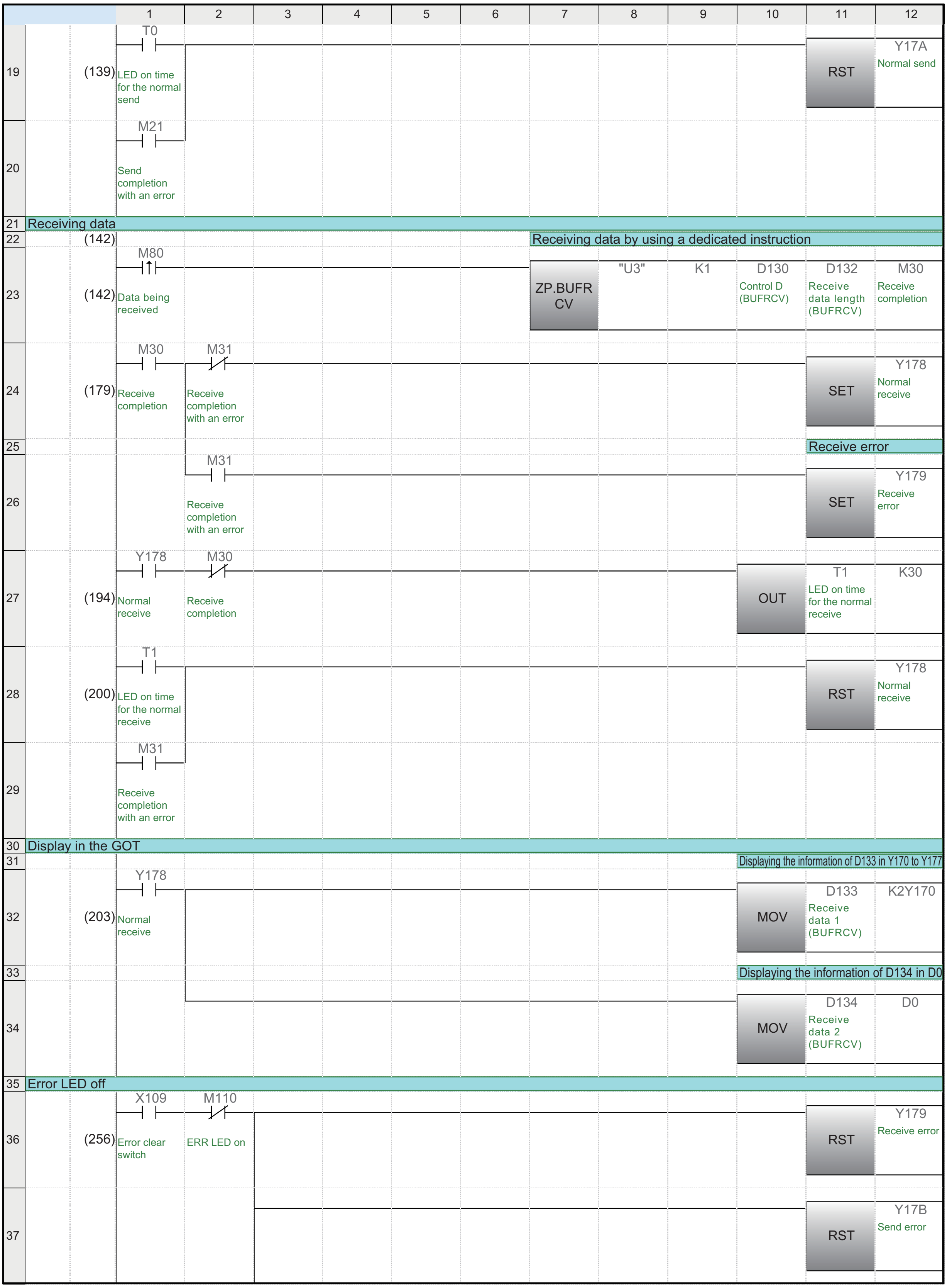
14 (116) M20 Send completion, M21 Send completion with an error — SET Y17A Normal send

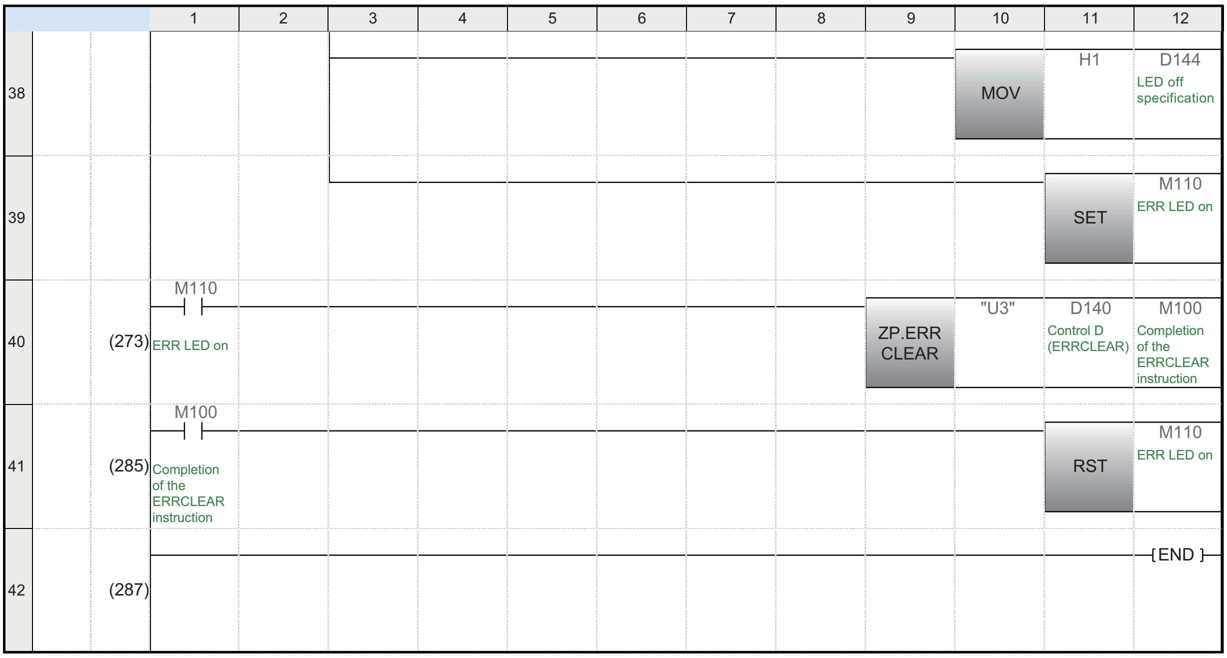
15 Send completion with an error

16 M21 Send completion with an error — SET Y17B Send error

17 RST M22 Send flag

18 (133) Y17A Normal send, M20 Send completion — OUT T0 LED on time for the normal send K30


1
2
3
4
5
6
7
8
9
10
11
12
19
(139)
T0
LED on time for the normal send
RST
Y17A
Normal send
20
M21
Send completion with an error
21
Receiving data
22
(142)
Receiving data by using a dedicated instruction
23
(142)
M80
Data being received
ZP.BUFRCV
"U3"
K1
D130
Control D (BUFRCV)
D132
Receive data length (BUFRCV)
M30
Receive completion
24
(179)
M30
Receive completion
M31
Receive completion with an error
SET
Y178
Normal receive
25
Receive error
26
M31
Receive completion with an error
SET
Y179
Receive error
27
(194)
Y178
Normal receive
M30
Receive completion
OUT
T1
LED on time for the normal receive
K30
28
(200)
T1
LED on time for the normal receive
RST
Y178
Normal receive
29
M31
Receive completion with an error
30
Display in the GOT
31
Displaying the information of D133 in Y170 to Y177
32
(203)
Y178
Normal receive
MOV
D133
Receive data 1 (BUFRCV)
K2Y170
33
Displaying the information of D134 in D0
34
MOV
D134
Receive data 2 (BUFRCV)
D0
35
Error LED off
36
(256)
X109
Error clear switch
M110
ERR LED on
RST
Y179
Receive error
37
RST
Y17B
Send error



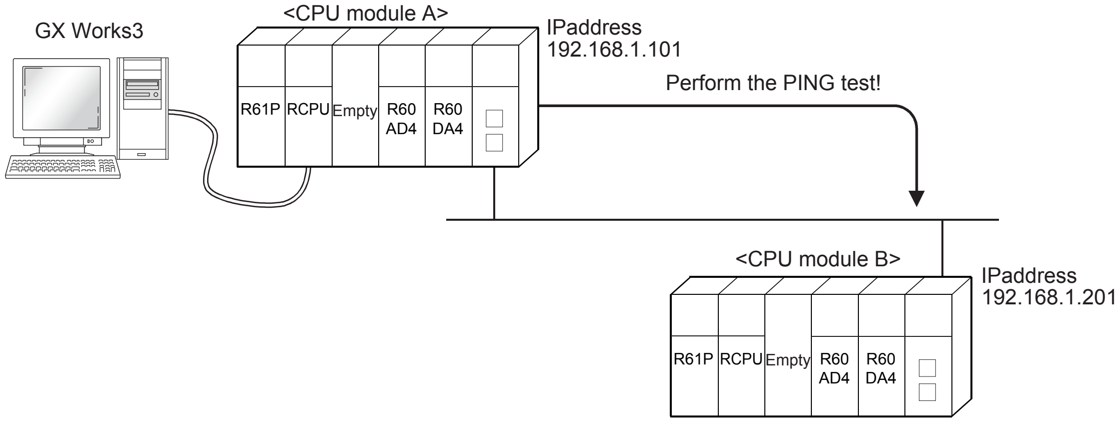
```
   1  2  3  4  5  6  7  8  9  10  11  12
38                               MOV   H1    D144
                                              LED off specification
39                                     SET   M110
                                              ERR LED on
40 (273) M110                    ZP.ERR  "U3"  D140  M100
         ERR LED on              CLEAR         Control D  Completion of the
                                               (ERRCLEAR) ERRCLEAR instruction
41 (285) M100                          RST   M110
         Completion of the                    ERR LED on
         ERRCLEAR instruction
42 (287)                                     {END}
```

7.4 PING Test with GX Works3 (via CPU Modules)

Perform the PING test with GX Works3.

The PING test checks whether the sequence program of the Ethernet-equipped module on the own station has completed successfully and the Ethernet-equipped module is properly connected to the external device on the same Ethernet network.

Power on the CPU modules A1 to A5 and B1 to B5, and set the CPU modules to the STOP state before performing the PING test.



Refer to Page 6 - 17 PING Test with GX Works3 (via CPU Modules) because the operating procedure is the same.

7.5 Operation of the Demonstration Machine

Operate the demonstration machine to check that data is sent/received between the CPU module A and CPU module B over Ethernet.

Operation method

1. Write parameters and programs to the CPU module A and CPU module B.
2. Set the RUN/STOP/RESET switch of the CPU module A (sending side) and CPU module B (receiving side) to the "RESET" position (for approximately one second) to reset the CPU modules. Then, move it to the "RUN" position.
3. Turn on and off X10A of the CPU module A to open the connection.
   The CPU module B is waiting for the connection to be open (Unpassive open) because the following value is set in the operation setting parameter.
   Opening Method: Do Not Open by Program
4. Turn on/off X100 to X107 of the CPU module A. Input a numerical value in the initial input device (D20).
   Turn on X108 of the CPU module A to send data.
5. Y170 to Y177 of the CPU module B turn on according to the status of X100 to X107 of the CPU module A.
   The numerical value of the initial input device (D20) of the CPU module A is displayed in the initial indication device (D0) of the CPU module B.
6. After checking the value, turn off X108 of the CPU module A and end the send processing.
   To send the data again, start from Step 4.
7. Turn on/off X100 to X107 of the CPU module B. Input a numerical value in the initial input device (D20).
   Turn on X108 of the CPU module B to send data.
8. Y170 to Y177 of the CPU module A turn on depending on the status of X100 to X107 of the CPU module B.
   The numerical value of the initial input device (D20) of the CPU module B is displayed in the initial indication device (D0) of the CPU module A.
9. After checking the value, turn off X108 of the CPU module B and end the send processing.
   To send the data again, start from Step 7.
10. Turn on and off X10B of the CPU module A to close the connection.

MEMO

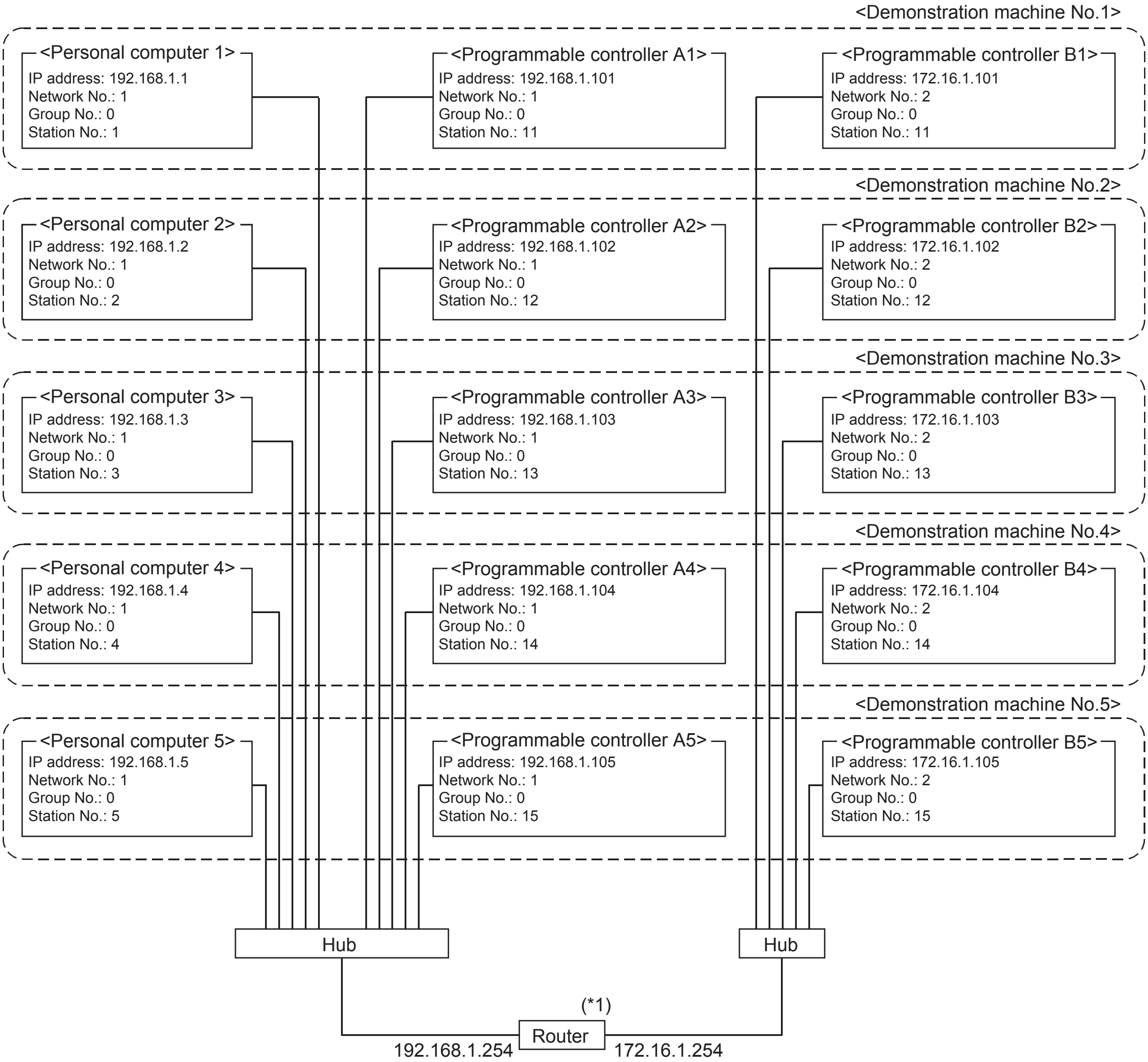
8 EXERCISE 4 (COMMUNICATIONS USING A FIXED BUFFER (NO PROCEDURE) VIA A ROUTER)

This exercise is for data communications via a router.
Perform the communications using a fixed buffer (no procedure), which is described in Exercise 3 in Chapter 7, via a router.

8.1 System Configuration

The following figure shows the system configuration of Exercise 4.
Although 10 CPU modules are connected on the same Ethernet network, perform communications only between the CPU modules with the same demonstration machine number in Exercise 4.

Module configuration



*1 Set the router referring to the manual of the router used.

Exercise

Refer to Page 7 - 3 Exercise because the exercise is the same as Exercise 3.

8.2 Setting the Ethernet-equipped Module (Programmable Controllers A1 to A5)

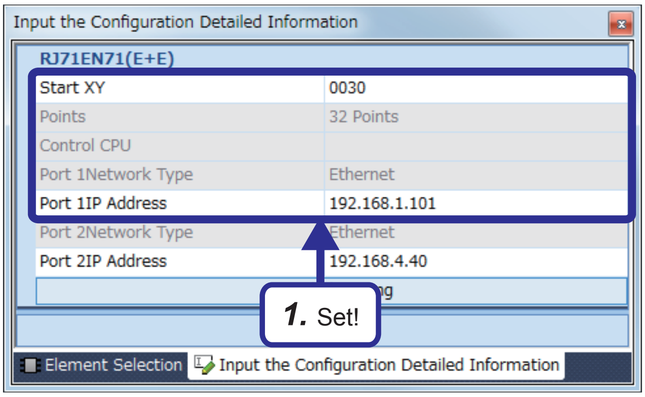
Set the Ethernet-equipped module (programmable controllers A1 to A5).

8.2.1 Setting parameters with GX Works3

Adding Ethernet-equipped module data

Set parameters of the Ethernet-equipped module with GX Works3.
Set them as described on Page 8 - 1 System Configuration.

Operating procedure



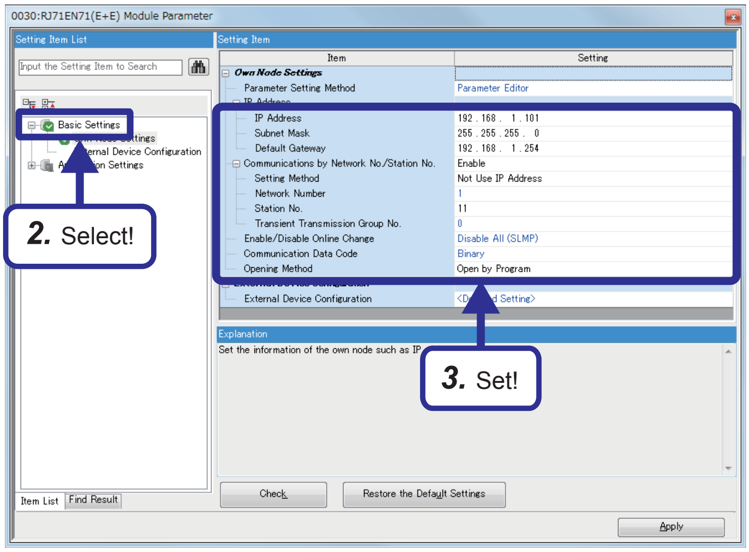
Input the Configuration Detailed Information

RJ71EN71(E+E)

Start XY	0030
Points	32 Points
Control CPU	
Port 1Network Type	Ethernet
Port 1IP Address	192.168.1.101
Port 2Network Type	Ethernet
Port 2IP Address	192.168.4.40

1. Create a new project and add "RJ71EN71(E+E)" from "Information Module" in the "Element Selection" window to the slot No.3 of the R35B by following the procedure described on Page 5 - 8 Adding Ethernet-equipped module data. After setting the parameters as follows, fix them.
(Click the [OK] button when the confirmation window for adding the module appears.)

[Parameters to be set]
Start XY: 0030
Port 1IP address: Set the IP address (192.168.1.101 to 192.168.1.105) of the demonstration machine used.



0030:RJ71EN71(E+E) Module Parameter

Item	Setting
Own Node Settings	
Parameter Setting Method	Parameter Editor
IP Address	192 . 168 . 1 . 101
Subnet Mask	255 . 255 . 255 . 0
Default Gateway	192 . 168 . 1 . 254
Communications by Network No./Station No.	Enable
Setting Method	Not Use IP Address
Network Number	1
Station No.	11
Transient Transmission Group No.	0
Enable/Disable Online Change	Disable All (SLMP)
Communication Data Code	Binary
Opening Method	Open by Program
External Device Configuration	<D[illegible]d Setting>

Explanation
Set the information of the own node such as IP

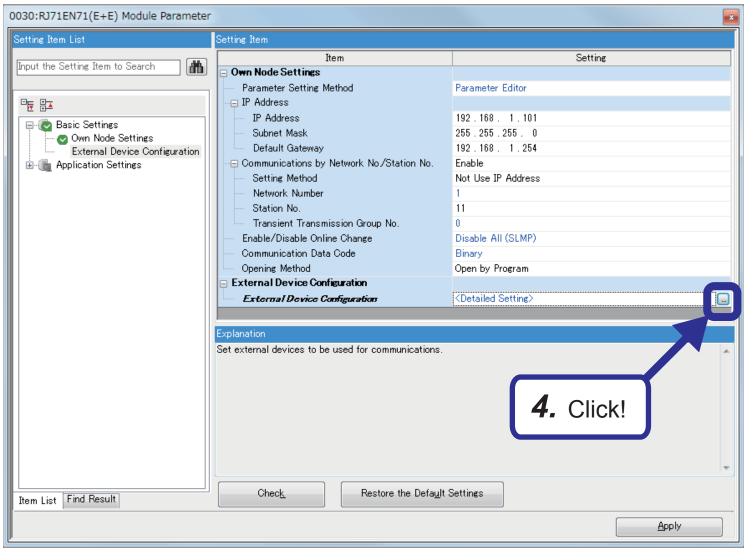
2. Double-click the RJ71EN71(E+E) module in the "Module Configuration" window and select "Basic Settings" in "Setting Item List".

3. Set "Communications by Network No./Station No." as follows.

[Parameters to be set]
Subnet Mask: 255.255.255.0
Default Gateway: 192.168.1.254
Communications by Network No./Station No.: Enable
Setting Method: Not Use IP Address
Network Number: 1
Station No.: Set the station number (11 to 15) of the demonstration machine used.
Communication Data Code: Optional (The binary code is used for the communications regardless of the setting.)
Opening Method: Open by Program

(To the next page)

(From the previous page)

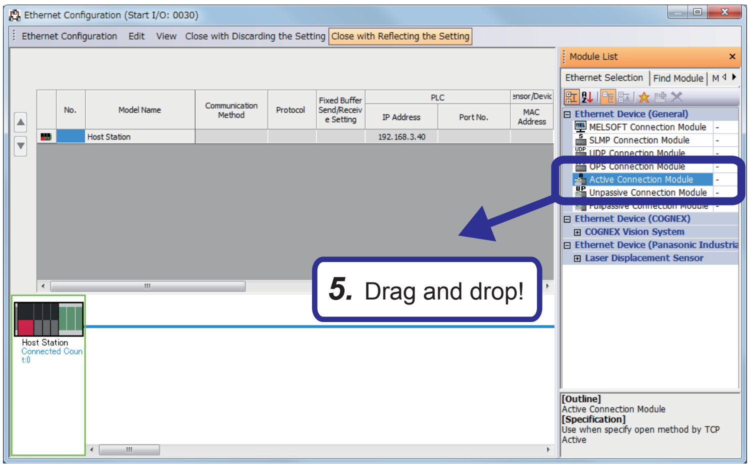

0030:RJ71EN71(E+E) Module Parameter
Setting Item List
Input the Setting Item to Search
Basic Settings
Own Node Settings
External Device Configuration
Application Settings
Setting Item
Check
Restore the Default Settings
Item List
Find Result
Apply
4. Click!

Item	Setting
Own Node Settings	
Parameter Setting Method	Parameter Editor
IP Address	
IP Address	192 . 168 . 1 . 101
Subnet Mask	255 . 255 . 255 . 0
Default Gateway	192 . 168 . 1 . 254
Communications by Network No./Station No.	Enable
Setting Method	Not Use IP Address
Network Number	1
Station No.	11
Transient Transmission Group No.	0
Enable/Disable Online Change	Disable All (SLMP)
Communication Data Code	Binary
Opening Method	Open by Program
External Device Configuration	
External Device Configuration	<Detailed Setting>

Explanation

Set external devices to be used for communications.

4. Click the button of "Detailed Setting" of "External Device Configuration".


Ethernet Configuration (Start I/O: 0030)
Ethernet Configuration
Edit
View
Close with Discarding the Setting
Close with Reflecting the Setting
Module List
Ethernet Selection
Find Module
M
Ethernet Device (General)
MELSOFT Connection Module
SLMP Connection Module
UDP Connection Module
OPS Connection Module
Active Connection Module
Unpassive Connection Module
Fullpassive Connection Module
Ethernet Device (COGNEX)
COGNEX Vision System
Ethernet Device (Panasonic Industria
Laser Displacement Sensor
Host Station
Connected Count:0
5. Drag and drop!

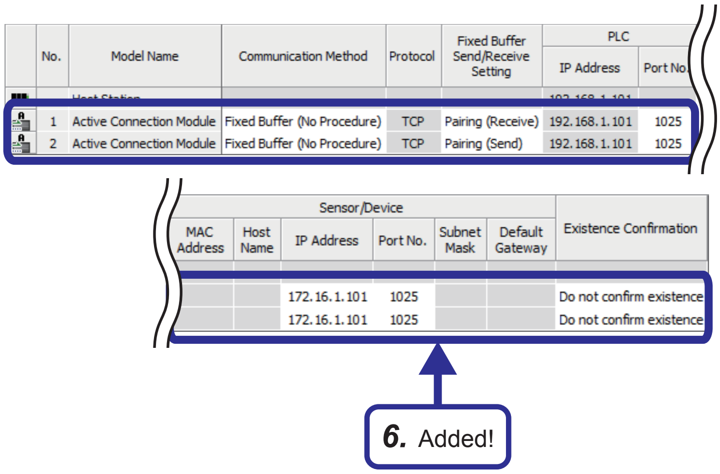
	No.	Model Name	Communication Method	Protocol	Fixed Buffer Send/Receive Setting	PLC		ensor/Devic
						IP Address	Port No.	MAC Address
		Host Station				192.168.3.40		

[Outline]
Active Connection Module
[Specification]
Use when specify open method by TCP Active

5. The "Ethernet Configuration" dialog box appears. Select "Active Connection Module" from "Ethernet Device (General)" in "Module List", and drag and drop it to the list of devices or device map area.
Select "Active Connection Module" again, and drag and drop it to the list of devices or device map area.

(To the next page)

(From the previous page)



No.	Model Name	Communication Method	Protocol	Fixed Buffer Send/Receive Setting	PLC IP Address	PLC Port No.
	Host Station				192.168.1.101	
1	Active Connection Module	Fixed Buffer (No Procedure)	TCP	Pairing (Receive)	192.168.1.101	1025
2	Active Connection Module	Fixed Buffer (No Procedure)	TCP	Pairing (Send)	192.168.1.101	1025

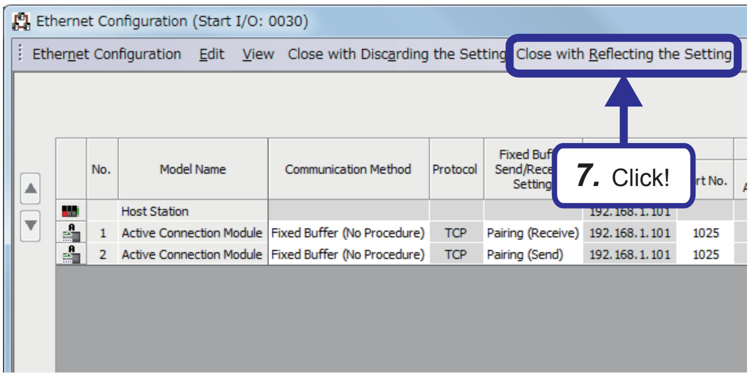
Sensor/Device MAC Address	Sensor/Device Host Name	Sensor/Device IP Address	Sensor/Device Port No.	Sensor/Device Subnet Mask	Sensor/Device Default Gateway	Existence Confirmation
		172.16.1.101	1025			Do not confirm existence
		172.16.1.101	1025			Do not confirm existence

6. Added!

6. "Active Connection Module" is added to the list of devices.
Set "Active Connection Module" as follows.

[Parameters to be set: 1]
Communication Method: Fixed Buffer (No Procedure)
Fixed Buffer Send/Receive Setting: Pairing (Receive)
Port No. (PLC): 1025
IP Address: Set the IP address (172.16.1.101 to 172.16.1.105) of the demonstration machine used.
Port No. (Sensor/Device): 1025
Existence Confirmation: Do not confirm existence

[Parameters to be set: 2]
Communication Method: Fixed Buffer (No Procedure)
Fixed Buffer Send/Receive Setting: Pairing (Send)
Port No. (PLC): 1025
IP Address: Set the IP address (172.16.1.101 to 172.16.1.105) of the demonstration machine used.
Port No. (Sensor/Device): 1025
Existence Confirmation: Do not confirm existence


Ethernet Configuration (Start I/O: 0030)
Ethernet Configuration
Edit
View
Close with Discarding the Setting
Close with Reflecting the Setting

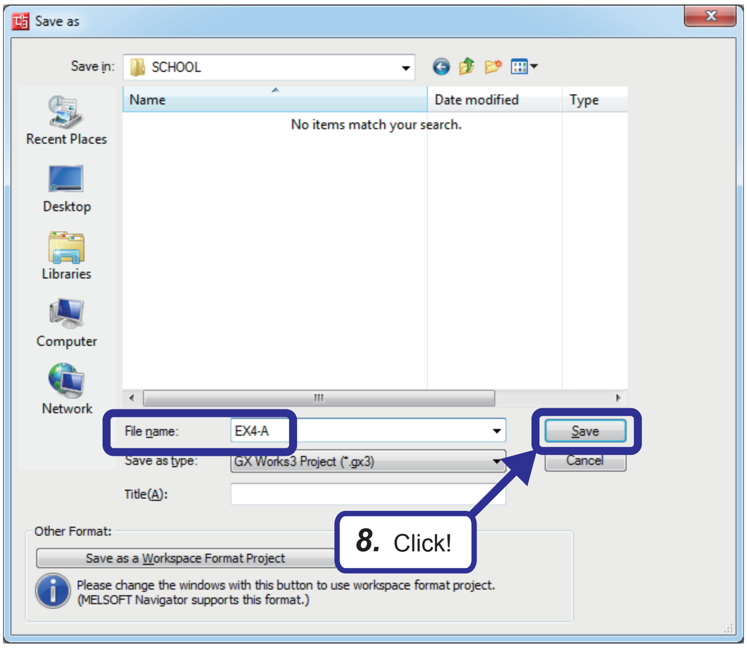
No.	Model Name	Communication Method	Protocol	Fixed Buf Send/Rece Setting		rt No.
	Host Station				192.168.1.101	
1	Active Connection Module	Fixed Buffer (No Procedure)	TCP	Pairing (Receive)	192.168.1.101	1025
2	Active Connection Module	Fixed Buffer (No Procedure)	TCP	Pairing (Send)	192.168.1.101	1025

7. Click!

7. After setting the parameters, click "Close with Reflecting the Setting" in the menu to close the "Ethernet Configuration" dialog box.

(To the next page)

(From the previous page)


Save as
Save in: SCHOOL
Name
Date modified
Type
No items match your search.
Recent Places
Desktop
Libraries
Computer
Network
File name: EX4-A
Save as type: GX Works3 Project (*.gx3)
Title(A):
Save
Cancel
Other Format:
Save as a Workspace Format Project
Please change the windows with this button to use workspace format project. (MELSOFT Navigator supports this format.)
8. Click!

8. Click [Project] ⇨ [Save As] from the menu to save the project.

Save destination: Desired location
File name: EX4-A
Title: Blank

8.2.2 Sequence program

Use the same sequence program as Page 7 - 8 Sequence program.

Point

After creating the sequence program, perform the following operations with GX Works3.

- Saving the project
  Project name: EX4-A
- Specifying a connection destination (Refer to Page 5 - 15 Specifying a connection destination.)
- Writing data to the CPU module (Refer to Page 5 - 17 Writing parameters.)

8.3 Setting the Ethernet-equipped Module (Programmable Controllers B1 to B5)

Set the Ethernet-equipped module (programmable controllers B1 to B5).

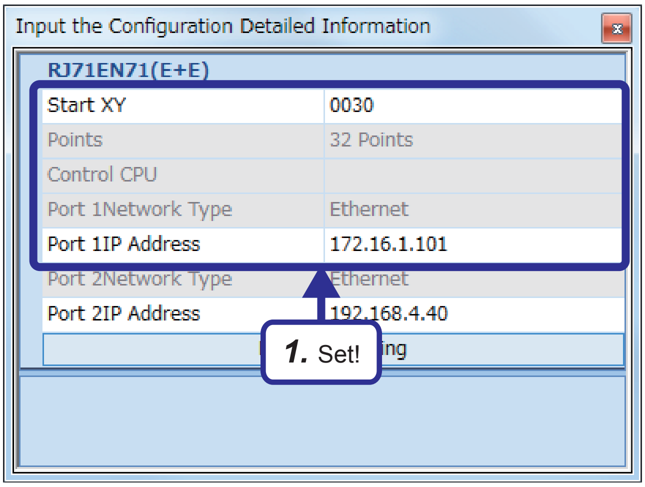
8.3.1 Setting parameters with GX Works3

Adding Ethernet-equipped module data

Set parameters of the Ethernet-equipped module with GX Works3.

Set them as described on Page 8 - 1 System Configuration.

Operating procedure



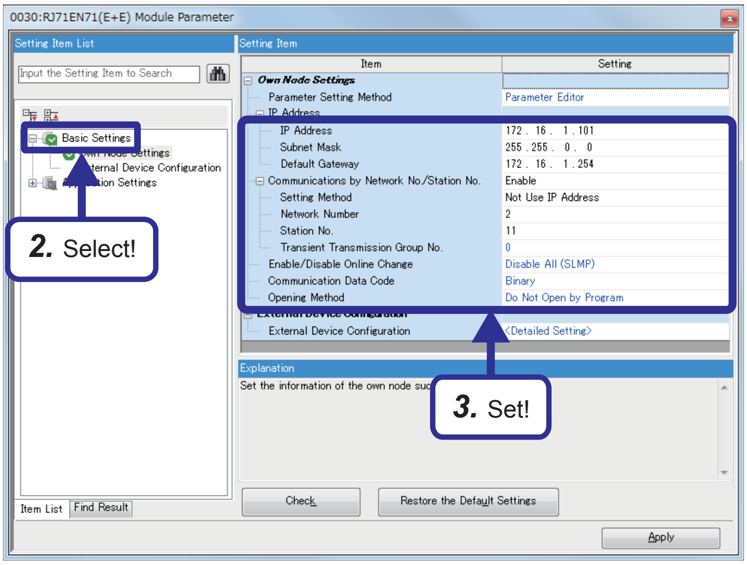
Input the Configuration Detailed Information

RJ71EN71(E+E)

Start XY	0030
Points	32 Points
Control CPU	
Port 1Network Type	Ethernet
Port 1IP Address	172.16.1.101
Port 2Network Type	Ethernet
Port 2IP Address	192.168.4.40

1. Create a new project and add "RJ71EN71(E+E)" from "Information Module" in the "Element Selection" window to the slot No.3 of the R35B by following the procedure described on Page 5 - 8 Adding Ethernet-equipped module data. After setting the parameters as follows, fix them.
(Click the [OK] button when the confirmation window for adding the module appears.)

[Parameters to be set]
Start XY: 0030
Port 1IP address: Set the IP address (172.16.1.101 to 172.16.1.105) of the demonstration machine used.



0030:RJ71EN71(E+E) Module Parameter

Item	Setting
Own Node Settings	
Parameter Setting Method	Parameter Editor
IP Address	
IP Address	172 . 16 . 1 . 101
Subnet Mask	255 . 255 . 0 . 0
Default Gateway	172 . 16 . 1 . 254
Communications by Network No./Station No.	Enable
Setting Method	Not Use IP Address
Network Number	2
Station No.	11
Transient Transmission Group No.	0
Enable/Disable Online Change	Disable All (SLMP)
Communication Data Code	Binary
Opening Method	Do Not Open by Program
External Device Configuration	
External Device Configuration	<Detailed Setting>

Set the information of the own node suc

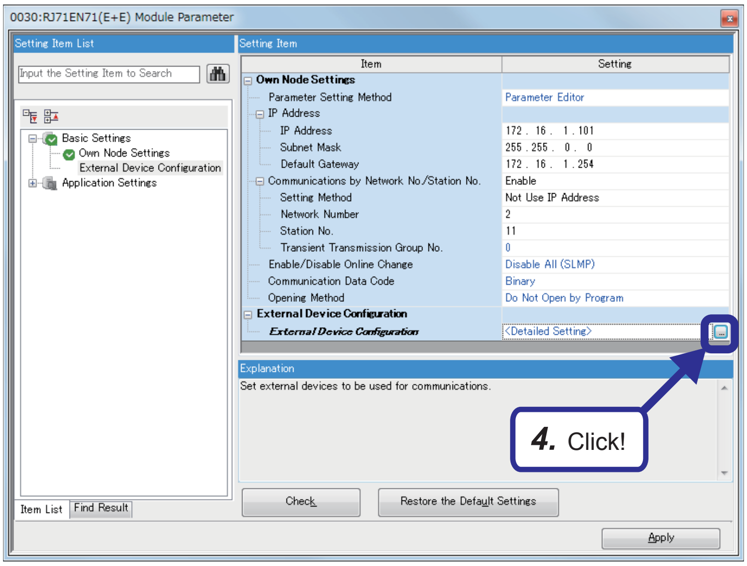
2. Double-click the RJ71EN71(E+E) module in the "Module Configuration" window and select "Basic Settings" in "Setting Item List".

3. Set "Communications by Network No./Station No." as follows.

[Parameters to be set]
Subnet Mask: 255.255.0.0
Default Gateway: 172.16.1.254
Communications by Network No./Station No.: Enable
Setting Method: Not Use IP Address
Network Number: 2
Station No.: Set the station number (11 to 15) of the demonstration machine used.
Communication Data Code: Optional (The binary code is used for the communications regardless of the setting.)
Opening Method: Do Not Open by Program

(To the next page)

(From the previous page)



0030:RJ71EN71(E+E) Module Parameter

Setting Item List

Basic Settings
- Own Node Settings
- External Device Configuration

Application Settings

Setting Item

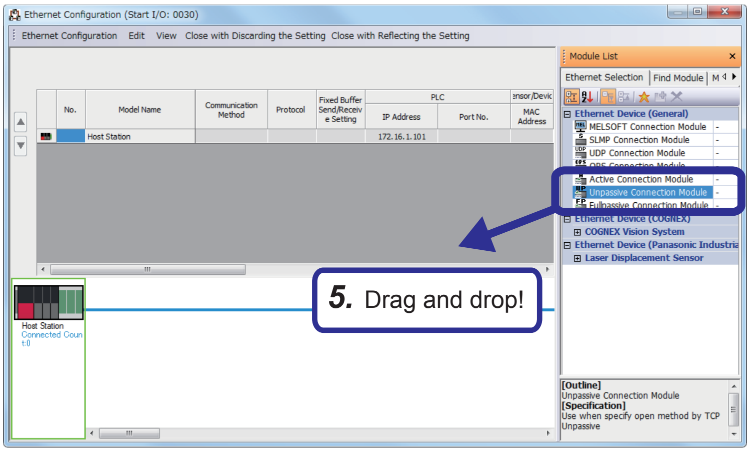
Item	Setting
Own Node Settings	
Parameter Setting Method	Parameter Editor
IP Address	
IP Address	172 . 16 . 1 . 101
Subnet Mask	255 . 255 . 0 . 0
Default Gateway	172 . 16 . 1 . 254
Communications by Network No./Station No.	Enable
Setting Method	Not Use IP Address
Network Number	2
Station No.	11
Transient Transmission Group No.	0
Enable/Disable Online Change	Disable All (SLMP)
Communication Data Code	Binary
Opening Method	Do Not Open by Program
External Device Configuration	
External Device Configuration	<Detailed Setting>

Explanation

Set external devices to be used for communications.

Input the Setting Item to Search
Item List
Find Result
Check
Restore the Default Settings
Apply
4. Click!

4. Click the ▭ button of "Detailed Setting" of "External Device Configuration".



Ethernet Configuration (Start I/O: 0030)

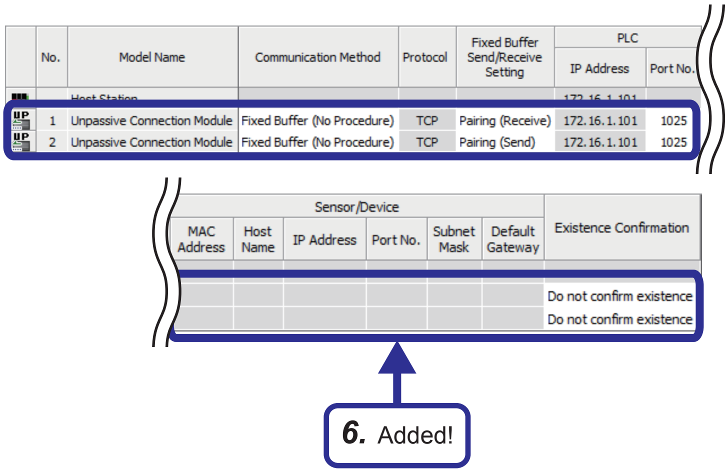
	No.	Model Name	Communication Method	Protocol	Fixed Buffer Send/Receive Setting	PLC		ensor/Devic
						IP Address	Port No.	MAC Address
		Host Station				172.16.1.101		

[Outline]
Unpassive Connection Module
[Specification]
Use when specify open method by TCP Unpassive

5. The "Ethernet Configuration" dialog box appears. Select "Unpassive Connection Module" from "Ethernet Device (General)" in "Module List" and drag and drop it to the list of devices or device map area.
Select "Unpassive Connection Module" again, and drag and drop it to the list of devices or device map area.

(To the next page)

(From the previous page)



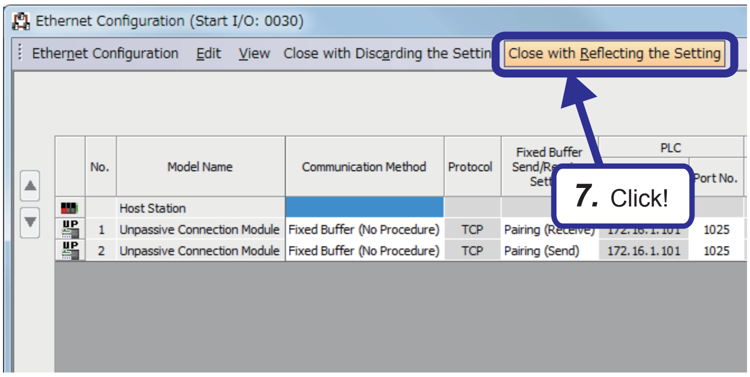
	No.	Model Name	Communication Method	Protocol	Fixed Buffer Send/Receive Setting	PLC	
						IP Address	Port No.
		Host Station				172.16.1.101	
UP	1	Unpassive Connection Module	Fixed Buffer (No Procedure)	TCP	Pairing (Receive)	172.16.1.101	1025
UP	2	Unpassive Connection Module	Fixed Buffer (No Procedure)	TCP	Pairing (Send)	172.16.1.101	1025

Sensor/Device						Existence Confirmation
MAC Address	Host Name	IP Address	Port No.	Subnet Mask	Default Gateway	
						Do not confirm existence
						Do not confirm existence

6. "Unpassive Connection Module" is added to the list of devices.
Set "Unpassive Connection Module" as follows.

[Parameters to be set: 1]
Communication Method: Fixed Buffer (No Procedure)
Fixed Buffer Send/Receive Setting: Pairing (Receive)
Port No. (PLC): 1025
Existence Confirmation: Do not confirm existence

[Parameters to be set: 2]
Communication Method: Fixed Buffer (No Procedure)
Fixed Buffer Send/Receive Setting: Pairing (Send)
Port No. (PLC): 1025
Existence Confirmation: Do not confirm existence



Ethernet Configuration (Start I/O: 0030)

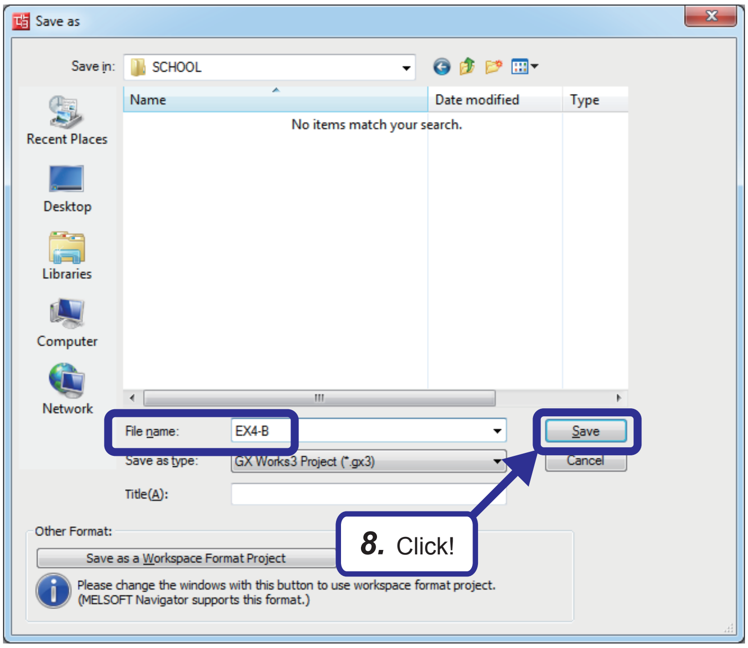
Ethernet Configuration
Edit
View
Close with Discarding the Settin
Close with Reflecting the Setting
7. Click!

	No.	Model Name	Communication Method	Protocol	Fixed Buffer Send/Receive Setting	PLC	
						IP Address	Port No.
		Host Station					
UP	1	Unpassive Connection Module	Fixed Buffer (No Procedure)	TCP	Pairing (Receive)	172.16.1.101	1025
UP	2	Unpassive Connection Module	Fixed Buffer (No Procedure)	TCP	Pairing (Send)	172.16.1.101	1025

7. After setting the parameters, click "Close with Reflecting the Setting" in the menu to close the "Ethernet Configuration" dialog box.

(To the next page)

(From the previous page)


Save as
Save in: SCHOOL
Recent Places
Desktop
Libraries
Computer
Network
Name
Date modified
Type
No items match your search.
File name: EX4-B
Save as type: GX Works3 Project (*.gx3)
Title(A):
Save
Cancel
Other Format:
Save as a Workspace Format Project
Please change the windows with this button to use workspace format project.
(MELSOFT Navigator supports this format.)
8. Click!

8. Click [Project] ⇨ [Save As] from the menu to save the project.

Save destination: Desired location
File name: EX4-B
Title: Blank

8.3.2 Sequence program

Use the same sequence program as Page 7 - 16 Sequence program.

Point

After creating the sequence program, perform the following operations with GX Works3.

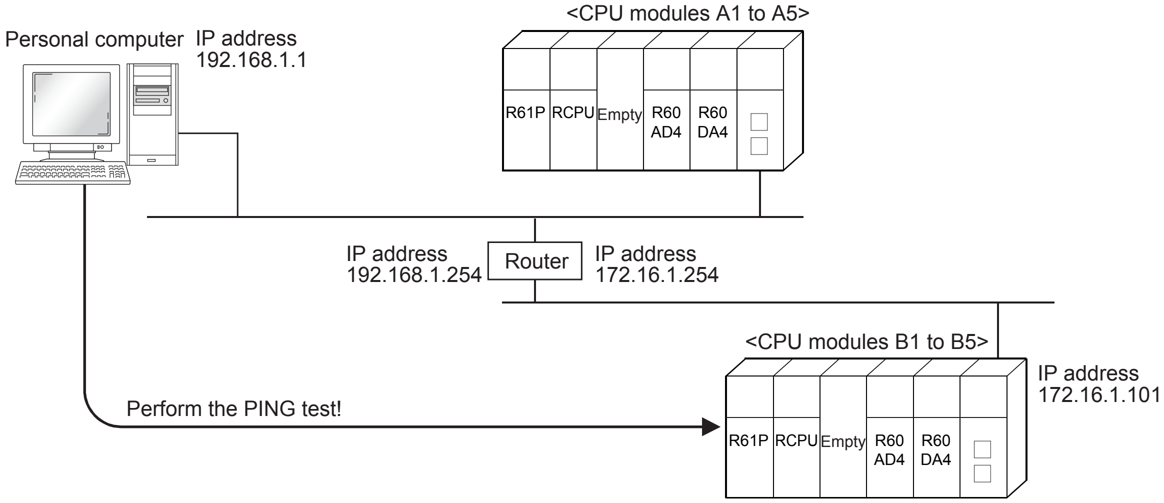
- Saving the project
  Project name: EX4-B
- Specifying a connection destination (Refer to Page 5 - 15 Specifying a connection destination.)
- Writing data to the CPU module (Refer to Page 5 - 17 Writing parameters.)

8.4 PING Test from the Personal Computer

Perform the PING test from the personal computer, which was used for the exercise of MX Component, to the programmable controllers B.

The PING test checks whether the initial processing of the Ethernet-equipped module on the side of the programmable controllers B has completed successfully and the personal computer is properly connected to the programmable controllers B.

Power on the CPU modules A1 to A5 and B1 to B5, and set the CPU modules to the STOP state before performing the PING test.



For an operating procedure example on the personal computer side, refer to Page 6 - 17 PING Test with GX Works3 (via CPU Modules).

8.5 Operation of the Demonstration Machine

Refer to Page 7 - 21 Operation of the Demonstration Machine because the operating procedure is the same.

APPENDICES

Appendix 1 Microsoft® Visual Basic® 2012

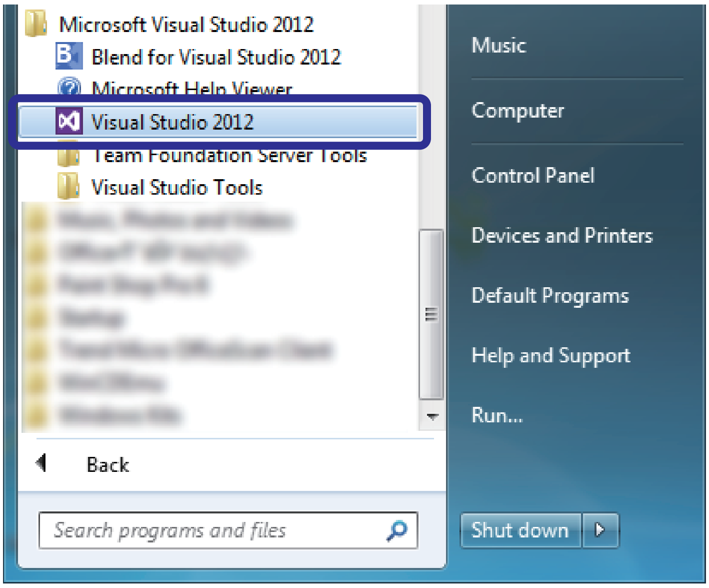
Appendix 1.1 Starting Visual Basic® 2012

This textbook describes how to start Visual Basic 2012 of Microsoft® Visual Studio® 2012. For details, refer to the product manual.

■Starting Visual Basic®

1. Click [Microsoft Visual Studio 2012] ⇨ [Visual Studio 2012] from the Windows® Start menu[*1].

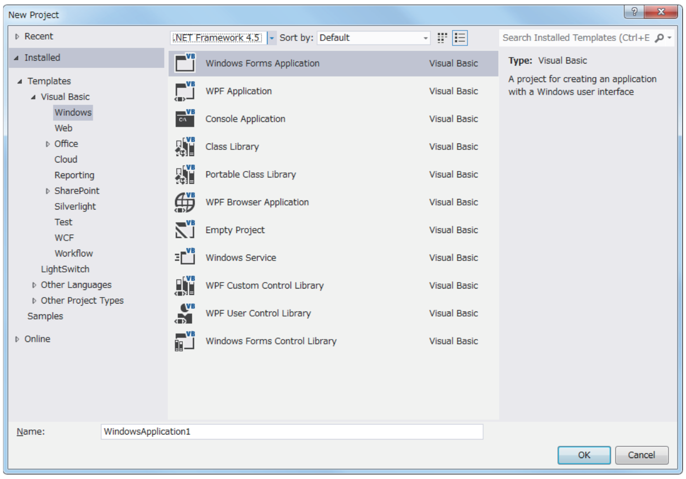
[*1] Select [Start] ⇨ [All apps] or [Start] ⇨ [All Programs].



■Creating a new Visual Basic® project

1. Select [New Project].

2. Select [Visual Basic] ⇨ [Windows Forms Application].

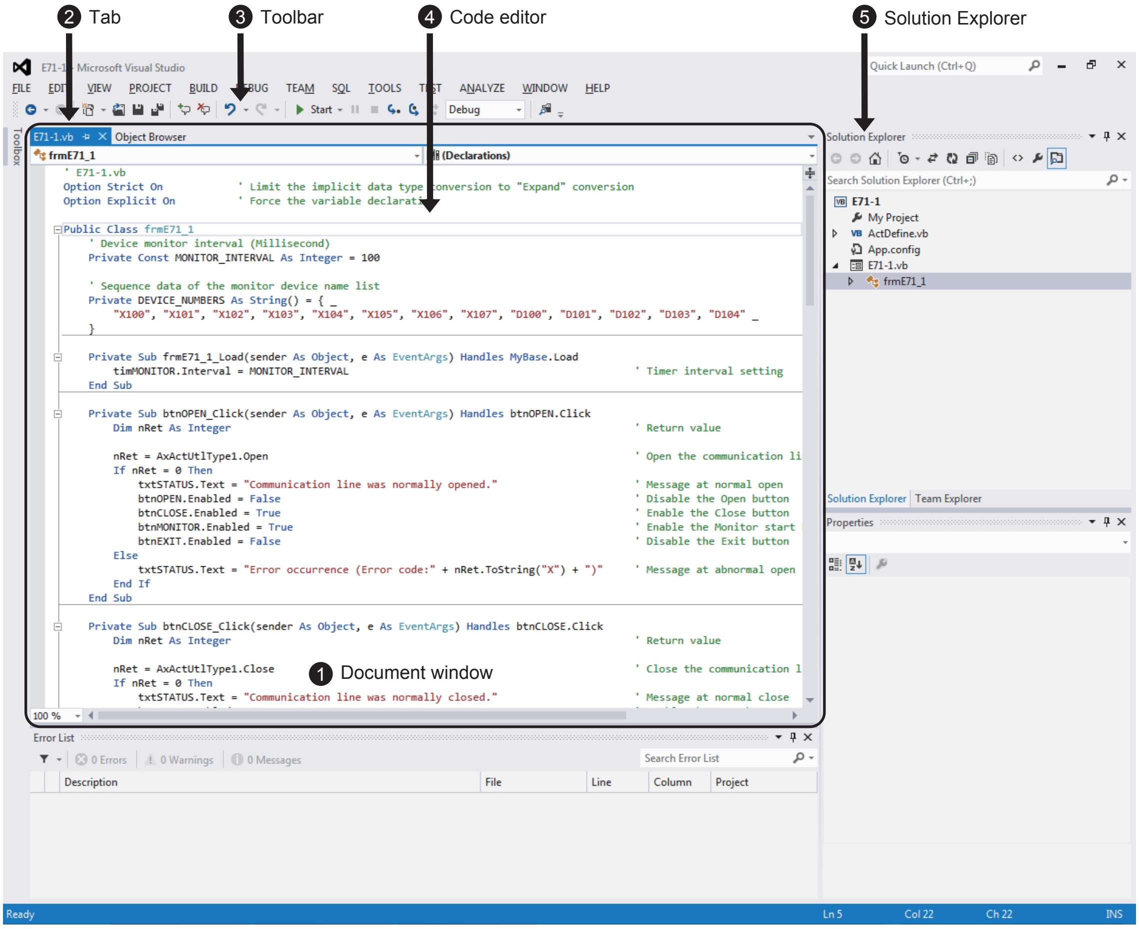

New Project
Recent
Installed
Templates
Visual Basic
Windows
Web
Office
Cloud
Reporting
SharePoint
Silverlight
Test
WCF
Workflow
LightSwitch
Other Languages
Other Project Types
Samples
Online
.NET Framework 4.5
Sort by: Default
Search Installed Templates (Ctrl+E
Windows Forms Application Visual Basic
WPF Application Visual Basic
Console Application Visual Basic
Class Library Visual Basic
Portable Class Library Visual Basic
WPF Browser Application Visual Basic
Empty Project Visual Basic
Windows Service Visual Basic
WPF Custom Control Library Visual Basic
WPF User Control Library Visual Basic
Windows Forms Control Library Visual Basic
OK
Cancel

Type: Visual Basic

A project for creating an application with a Windows user interface

Name: WindowsApplication1

■Visual Basic® 2012 window



```
' E71-1.vb
Option Strict On            ' Limit the implicit data type conversion to "Expand" conversion
Option Explicit On          ' Force the variable declarati

Public Class frmE71_1
    ' Device monitor interval (Millisecond)
    Private Const MONITOR_INTERVAL As Integer = 100

    ' Sequence data of the monitor device name list
    Private DEVICE_NUMBERS As String() = { _
        "X100", "X101", "X102", "X103", "X104", "X105", "X106", "X107", "D100", "D101", "D102", "D103", "D104" _
    }

    Private Sub frmE71_1_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        timMONITOR.Interval = MONITOR_INTERVAL                                         ' Timer interval setting
    End Sub

    Private Sub btnOPEN_Click(sender As Object, e As EventArgs) Handles btnOPEN.Click
        Dim nRet As Integer                                                            ' Return value

        nRet = AxActUtlType1.Open                                                      ' Open the communication li
        If nRet = 0 Then
            txtSTATUS.Text = "Communication line was normally opened."                 ' Message at normal open
            btnOPEN.Enabled = False                                                    ' Disable the Open button
            btnCLOSE.Enabled = True                                                    ' Enable the Close button
            btnMONITOR.Enabled = True                                                  ' Enable the Monitor start
            btnEXIT.Enabled = False                                                    ' Disable the Exit button
        Else
            txtSTATUS.Text = "Error occurrence (Error code:" + nRet.ToString("X") + ")"  ' Message at abnormal open
        End If
    End Sub

    Private Sub btnCLOSE_Click(sender As Object, e As EventArgs) Handles btnCLOSE.Click
        Dim nRet As Integer                                                            ' Return value

        nRet = AxActUtlType1.Close                                                     ' Close the communication l
        If nRet = 0 Then
            txtSTATUS.Text = "Communication line was normally closed."                 ' Message at normal close
```

❶ Document window

This window is created dynamically when a file is opened or items are edited. The document window is also called workspace.

❷ Tab

This part is used to switch windows displayed in a document window. The current window can be identified by the front tab. To switch the window, click another tab.

❸ Toolbar

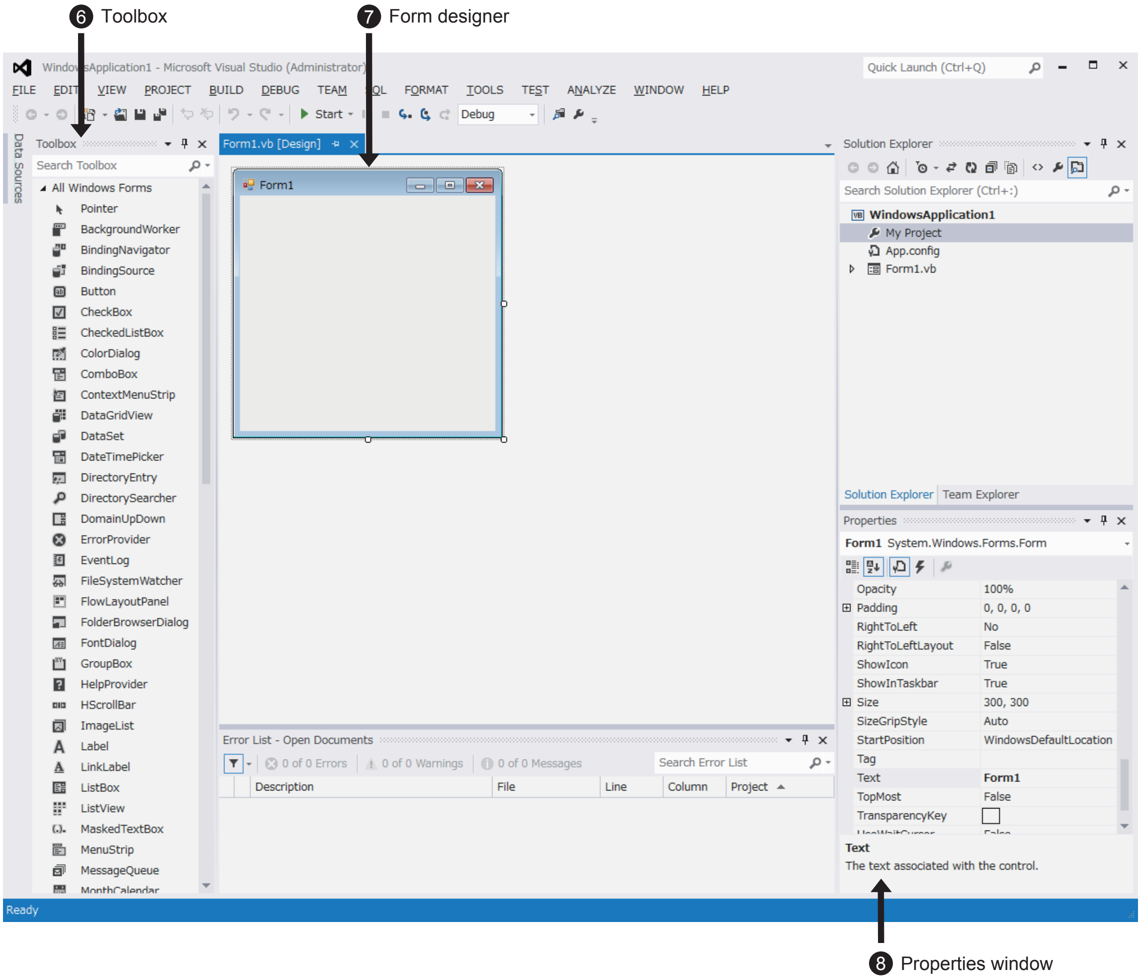
Frequently used commands can be selected.

❹ Code editor

Program codes are displayed and edited in this window. This window appears when an appropriate file is selected in "Solution Explorer" described below.

❺ Solution Explorer

This area displays solutions and file structures of projects. Select a file to be displayed and edited in the document window.



Opacity	100%
Padding	0, 0, 0, 0
RightToLeft	No
RightToLeftLayout	False
ShowIcon	True
ShowInTaskbar	True
Size	300, 300
SizeGripStyle	Auto
StartPosition	WindowsDefaultLocation
Tag	
Text	Form1
TopMost	False
TransparencyKey	□
UseWaitCursor	False

Text
The text associated with the control.

❻ Toolbox

Controls to be arranged in the form can be selected in this window.

❼ Form designer

This window is the basis of an application.

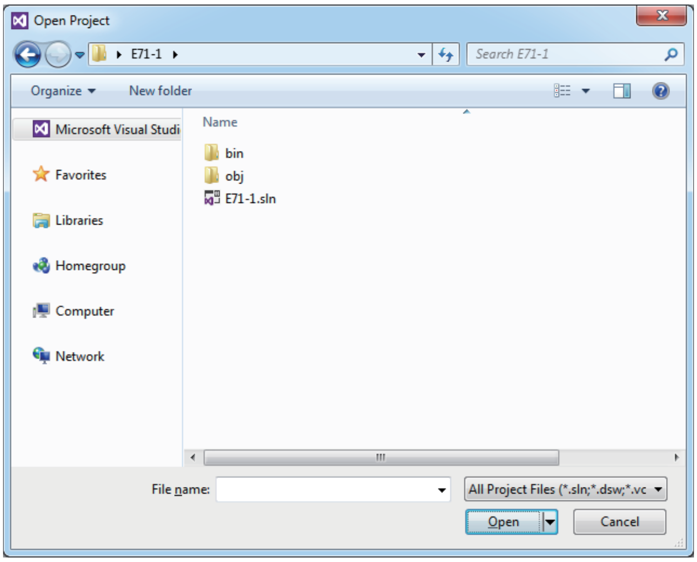
A desired interface can be created by arranging controls in the form.

❽ Properties window

Attributes (properties) such as a shape and color can be set to the control in this window.

■Loading a solution

1. Select [FILE] ⇨ [Open] ⇨ [Open Project].
2. Select the file location.
3. Select a solution file name (*.sln) and click the [Open] button.



All the projects and files in the solution are loaded.

■Saving a solution

Select [FILE] ⇨ [Save All].

■Closing a solution

Select [FILE] ⇨ [Close Project].

Point

In Visual Basic .NET, a project contains files composing an execution file and a solution organizes multiple projects.

Although a solution file or a project file can be specified, projects are always included in the solution after loading an application. Therefore, specify a solution file when loading an application.

In Visual Basic .NET, multiple files are required to complete an application. Even when the name or save destination of the application is changed and saved, all the files will not be saved in the changed save destination. Therefore, if you do not understand the file structure, create an application with the specified name and save destination, and always overwrite the application.

■Displaying the form designer

Right-click the form module file in the Solution Explorer window and click [View Designer].

■Displaying the code editor

Double-click a control in the form.

Or, right-click the form module file in the Solution Explorer window and click [View Code].

■Displaying the properties window

Select [VIEW] ⇨ [Properties Window].

■Execution

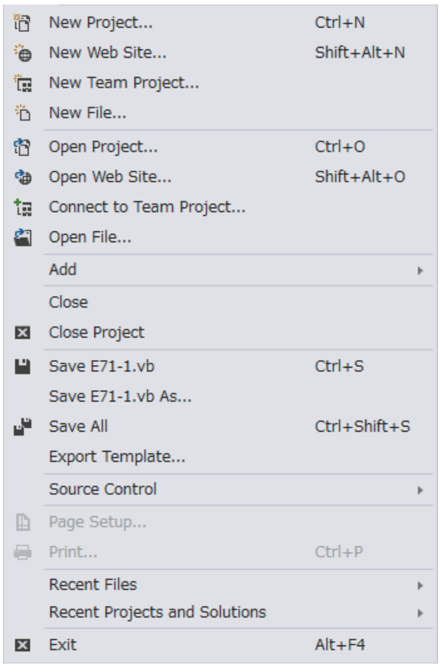
Select [DEBUG] ⇨ [Start Debugging] or ▶ Start ▾ on the toolbar.

■Ending the application

Select [DEBUG] ⇨ [Stop Debugging] or ■ on the toolbar.

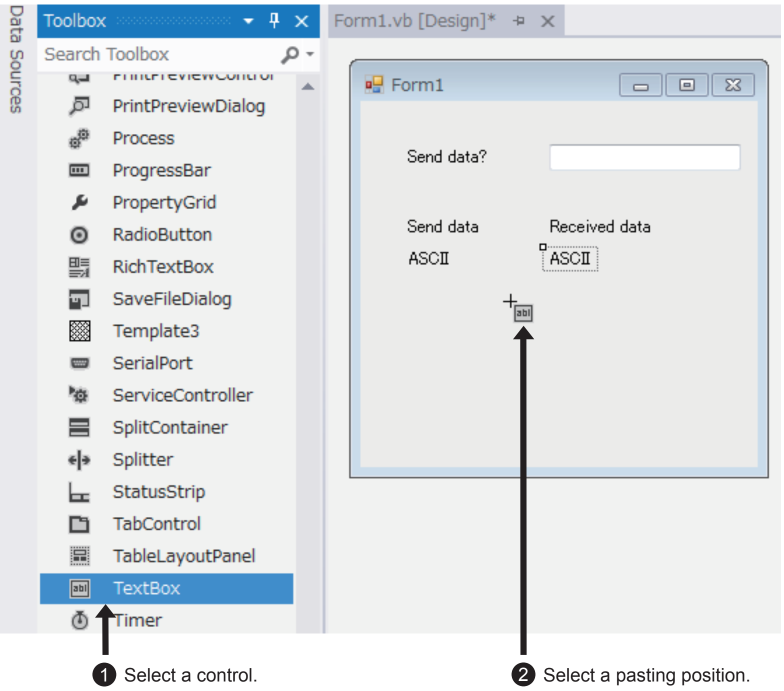
■Ending Visual Basic®

Select [FILE] ⇨ [Exit].



Appendix 1.2 Editing the form

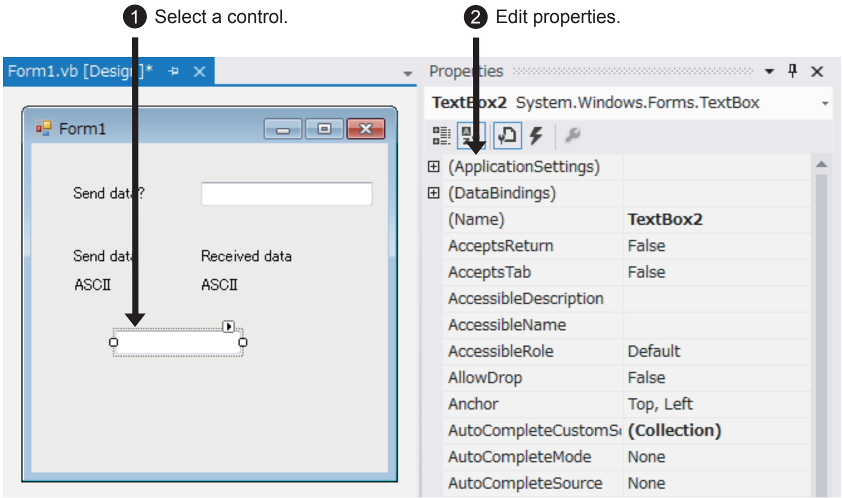
■Pasting a control



❶ Select a control to be pasted in the toolbox.

❷ Click a position to past the control.

■Editing properties



⊞ (ApplicationSettings)	
⊞ (DataBindings)	
(Name)	**TextBox2**
AcceptsReturn	False
AcceptsTab	False
AccessibleDescription	
AccessibleName	
AccessibleRole	Default
AllowDrop	False
Anchor	Top, Left
AutoCompleteCustomS	**(Collection)**
AutoCompleteMode	None
AutoCompleteSource	None

❶ Select a control to edit its properties.

❷ Edit the properties.

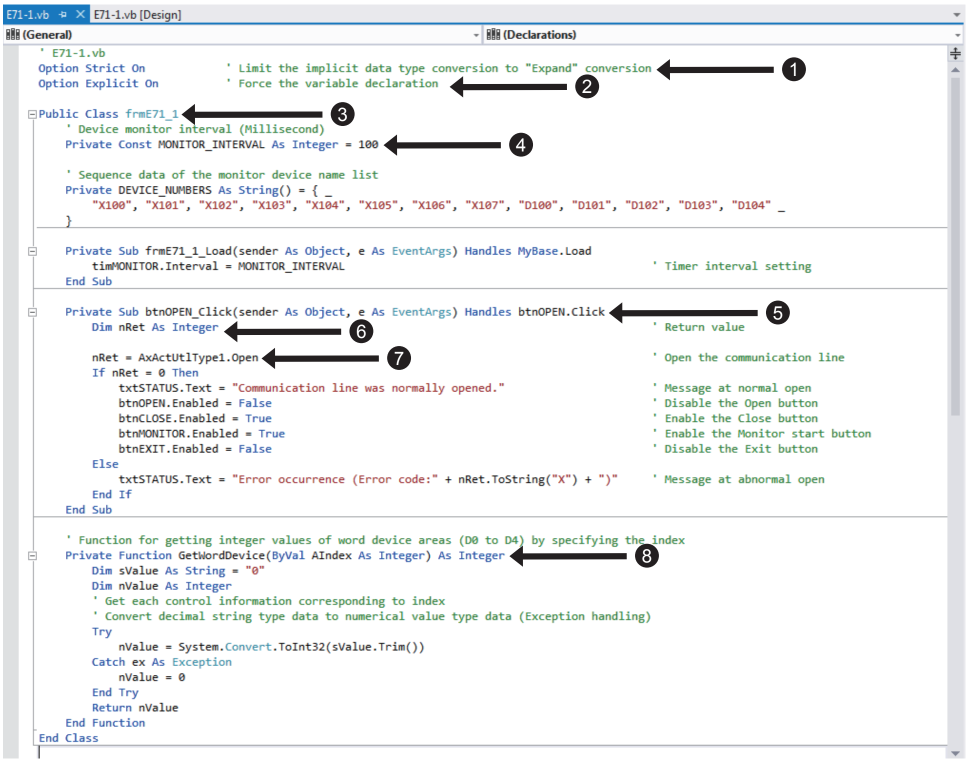
■Locking a control

The size or position of a control may be changed accidentally while the form or control is being edited. Controls can be fixed by the "Lock Controls" function to prevent them from being changed accidentally. With right-click on the form or control, controls can be locked or unlocked.

The locked state of the control can be checked and changed from "Locked" in the "Properties" window.

Appendix 1.3 How to write a program

■Sample code/basic statements



```
' E71-1.vb
Option Strict On            ' Limit the implicit data type conversion to "Expand" conversion
Option Explicit On          ' Force the variable declaration

Public Class frmE71_1
    ' Device monitor interval (Millisecond)
    Private Const MONITOR_INTERVAL As Integer = 100

    ' Sequence data of the monitor device name list
    Private DEVICE_NUMBERS As String() = { _
        "X100", "X101", "X102", "X103", "X104", "X105", "X106", "X107", "D100", "D101", "D102", "D103", "D104" _
    }

    Private Sub frmE71_1_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        timMONITOR.Interval = MONITOR_INTERVAL                                              ' Timer interval setting
    End Sub

    Private Sub btnOPEN_Click(sender As Object, e As EventArgs) Handles btnOPEN.Click
        Dim nRet As Integer                                                                 ' Return value

        nRet = AxActUtlType1.Open                                                           ' Open the communication line
        If nRet = 0 Then
            txtSTATUS.Text = "Communication line was normally opened."                      ' Message at normal open
            btnOPEN.Enabled = False                                                         ' Disable the Open button
            btnCLOSE.Enabled = True                                                         ' Enable the Close button
            btnMONITOR.Enabled = True                                                       ' Enable the Monitor start button
            btnEXIT.Enabled = False                                                         ' Disable the Exit button
        Else
            txtSTATUS.Text = "Error occurrence (Error code:" + nRet.ToString("X") + ")"     ' Message at abnormal open
        End If
    End Sub

    ' Function for getting integer values of word device areas (D0 to D4) by specifying the index
    Private Function GetWordDevice(ByVal AIndex As Integer) As Integer
        Dim sValue As String = "0"
        Dim nValue As Integer
        ' Get each control information corresponding to index
        ' Convert decimal string type data to numerical value type data (Exception handling)
        Try
            nValue = System.Convert.ToInt32(sValue.Trim())
        Catch ex As Exception
            nValue = 0
        End Try
        Return nValue
    End Function
End Class
```

❶ Option Strict {On|Off*}

This statement sets whether to enable all type conversions implicitly without specifying the data type conversion by using program codes (Off) or to enable only safe conversions (On). The default value is Off. Specify On when creating a high-reliable program.

❷ Option Explicit {On*|Off}

This statement sets whether to declare the variable implicitly (Off) or not (On) when a local variable is not declared. The default value is On and variable declaration is required.

❸[accessibility] **Class** classname [**Inherits** inheritsclassname]

Describe the class structure and program here.

End Class

This statement defines a class.

accessibility: Set the access level of the class.

classname: Set the name of the class.

inheritsclassname: Optional. Specify another class to be inherited.

❹[accessibility] **Const** constname **As** type = initexpr

This statement defines a constant.

accessibility: Set the access level of the constant.

constname: Set the name of the constant.

type: Specify the constant type.

initexpr: Specify the constant value.

❺[accessibility] **Sub** subname[(arglist)] [**Handles** event]

Describe a program code of a procedure here.

End Sub

This statement defines a procedure with no return value.
accessibility: Set the access level of the procedure.
funcname: Set the name of the procedure.
arglist: Specify the argument list.
event: Specify the events to be processed in this procedure.

❻[accessibility] **Dim** varname **As** type [= initexpr]

This statement defines a variable.
accessibility: Set the access level of the variable.
varname: Set the name of the variable.
type: Specify the variable type.
initexpr: Specify an initial value of the variable.
When accessibility is specified, the Dim keyword is optional.

❼varname = expr

This statement assigns a value to the variable.
varname: Set the name of the variable.
expr: Specify a value to be assigned.

❽[accessibility] **Function** funcname[(arglist)] **As** type [**Handles** event]

Describe a program code of a procedure here.

End Function

This statement defines a procedure with a return value.
accessibility: Set the access level of the procedure.
funcname: Set the name of the procedure.
arglist: Specify the argument list.
type: Specify the return value type.
event: Specify the events to be processed in this procedure.

- accessibility
  Specify the access level for the elements such as modules, classes, constants, and variables.
  Private: The elements can be accessed only from within the same class or module.
  Public: The elements can be accessed from out of their class and module.
  There are other access levels such as Friend and Protected.
- arglist
  Specify the argument to be passed to a procedure.
  [**Optional**] [{**ByVal***|**ByRef**}] argname **As** type [= defaultvalue]
  Optional: An argument can be omitted.
  ByVal: The procedure cannot change argument values.
  ByRef: The procedure can change argument values in the same way as variables.
  argname: Specify the name of the argument called formal argument.
  type: Specify the argument type.
  defaultvalue: Specify the default value of the argument. When Optional is specified, always specify the default value.
  When ByVal or ByRef is not specified, ByVal is considered to be specified.

Appendix 1.4 List of data types

Data type	Function	
Boolean	Description	Boolean type
	Value range	True or False
	Definition example	**Dim** variable name **As Boolean**
Byte	Description	Byte type (1-byte unsigned integer type)
	Value range	0 to 255 (Unsigned)
	Definition example	**Dim** variable name **As Byte**
Char	Description	Character type (2-byte unsigned integer type)
	Value range	0 to 65536 (Unsigned)
	Definition example	**Dim** variable name **As Char**
Date	Description	Date type
	Value range	January 1, 0001, 0:00:00 to December 31, 9999, 11:59:59 PM
	Definition example	**Dim** variable name **As Date**
Decimal	Description	Decimal type (16-byte decimal floating-point type)
	Value range	-79,228,162,514,264,337,593,543,950,335 to 79,228,162,514,264,337,593,543,950,335
	Definition example	**Dim** variable name **As Decimal**
Double	Description	Double-precision floating-point type (8-byte floating-point type)
	Value range	-1.79769313486231570E+308 to 1.79769313486231570E+308
	Definition example	**Dim** variable name **As Double**
Integer	Description	Integer type (4-byte signed integer type)
	Value range	-2,147,483,648 to 2,147,483,647
	Definition example	**Dim** variable name **As Integer**
Long	Description	Long integer type (8-byte signed integer type)
	Value range	-9,223,372,036,854,775,808 to 9,223,372,036,854,775,807
	Definition example	**Dim** variable name **As Long**
Object	Description	Object type
	Value range	(Any type data can be stored.)
	Definition example	**Dim** variable name **As TextBox**
Short	Description	Short integer type (2-byte signed integer type)
	Value range	-32,768 to 32,767
	Definition example	**Dim** variable name **As Short**
Single	Description	Single-precision floating-point type (4-byte floating-point type)
	Value range	-3.4028235E+38 to 3.4028235E+38
	Definition example	**Dim** variable name **As Single**
String	Description	String type
	Value range	0 to 2 billion of Unicode characters
	Definition example	**Dim** variable name **As String**

Appendix 1.5 List of classes

Class	Function
System.Convert	Converts a base data type to another base data type. ToInt32(method): Converts a specified value to a 32-bit signed integer.
System.Data (Namespace)	Provides access to classes that represent the ADO.NET architecture.
System.Drawing.Pens	Defines an object used to draw lines and curves. This class cannot be inherited.
System.Drawing.Brushs	Defines objects used to fill the interiors of graphical shapes such as rectangles, ellipses, pies, polygons, and paths.
System.Drawing.Graphics	Encapsulates a GDI+ drawing surface. This class cannot be inherited. DrawLine(method): Draws a line connecting the two points specified by the coordinate pairs. DrawString(method): Draws the specified text string at the specified location with the specified Brush and Font objects. FillRectangle(method): Fills the interior of a rectangle specified by a pair of coordinates, a width, and a height.
System.IO.File	Provides static methods for the creation, copying, deletion, moving, and opening of a single file, and aids in the creation of FileStream objects. Copy(method): Copies an existing file to a new file. Delete(method): Deletes the specified file. Exists(method): Determines whether the specified file exists. Open(method): Opens the specified file.
System.IO.Path	Performs operations on String instances that contain file or directory path information. GetDirectoryName(method): Returns the directory information for the specified path string. GetExtension(method): Returns the extension of the specified path string. GetFileName(method): Returns the file name and extension of the specified path string. GetTempFileName(method): Creates a uniquely named, zero-byte temporary file on disk and returns the full path of that file.
System.Math	Provides constants and static methods for trigonometric, logarithmic, and other common mathematical functions. E(constant): Represents the natural logarithmic base, specified by the constant, e. PI(constant): Represents the ratio of the circumference of a circle to its diameter, specified by the constant, π. Abs(method): Returns the absolute value of a specified value. Cos(method): Returns the cosine of the specified angle. Log(method): Returns the natural logarithm of a specified number. Sin(method): Returns the sine of the specified angle. Sqrt(method): Returns the square root of a specified number. Tan(method): Returns the tangent of the specified angle.
System.String	Represents text as a sequence of UTF-16 code units. Length(property): Gets the number of characters in the current String object. IndexOf(method): Searches for the specified string in a particular string and get the index. Replace(method): Replaces the specified character in a string with another character. Substring(method): Retrieves a substring from the instance. ToLower(method): Returns a copy of the string converted to lowercase. ToUpper(method): Returns a copy of the string converted to uppercase.
Socket	Implements the Berkeley sockets interface. Close(method): Closes the Socket connection and releases all associated resources. Connect(method): Establishes a connection to a remote host. Receive(method): Receives data from a bound Socket. Send(method): Sends data to a connected Socket. Shutdown(method): Disables sends and receives on a Socket.
Microsoft.VisualBasic (Namespace)	Contains types that support the Visual Basic Runtime in Visual Basic. (Function example) Division method of a string by using the Strings module Strings.Left(str, length) Strings.Mid(str, start, length) Strings.Right(str, length) str: A string to be divided start: Starting position of the characters to divide (Numerical value beginning with 1) length: The number of characters to be divided *Note that the pure .NET programming uses String.Substring.

Appendix 1.6 Brief commands

For...Next statement

■Function

This is a flow control statement which repeats a sequential statement for the specified number of times.

■Format

For counter = start **To** end [**Step** step]

[statements]

[**Exit For**]

[statements]

Next [counter]

The following table lists the parts of the **For...Next** statement.

Part	Description
counter	Required in the For statement. Numeric variable. The control variable for the loop.
start	Required. Numeric expression. The initial value of counter.
end	Required. Numeric expression. The final value of counter.
step	Optional. Numeric expression. The amount by which counter is incremented each time through the loop.
statements	Optional. One or more statements between For and Next that run the specified number of times.

If...Then...Else statement

■Function

This is a flow control statement which performs the execution with conditions depending on the value of an expression.

■Format

If condition **Then** statements[**Else** elsestatements]

The following block type statement can also be used.

If condition **Then**

[statements]

[**ElseIf** condition-n **Then**

[elseifstatements]]...

[**Else**

[elsestatements]]

End If

The following table lists the parts of the **If**...**Then**...**Else** statement.

Part	Description
condition	Required. Expression. Must evaluate to True or False, or to a data type that is implicitly convertible to Boolean.
statements	Optional. One or more statements following If...Then that are executed if condition evaluates to True.
condition-n	Same as the condition argument
elseifstatements	Optional. One or more statements following ElseIf...Then that are executed if elseifcondition evaluates to True.
elsestatements	Optional. One or more statements that are executed if no previous condition or elseifcondition expression evaluates to True.

Exception handling

■Function

This is a statement which handles exceptions in statements that may cause errors.

■Format

Try: Describe a code that may cause an error. (This statement is always executed.)

[**Catch** value **As** anyexceptionclass]: Describe a code to be executed when a specific error occurs. (This statement is executed only when a specific error occurs.)

Catch ex As Exception: Describe a code which is executed when an error occurs. (This statement is executed when an error occurs.)

[**Finally**]: Describe a code which is executed finally regardless of whether an error occurs or not in the Try-Catch statement. (This statement is always executed.)

End Try: To execute different programs depending on the error type, specify the exception class for anyexceptionclass. Specify Exception to perform the exception handling for any errors.

■Note

Perform the exception handling within a narrow range.

The exception handling for many functions and the whole statement in a wide range prevents users from identifying the problems in programming and creating a stable system.

Type conversion functions (Convert)

■Function

This is a statement which converts a specified value to the specified type.

■Format

Convert.ToString(value, tobase): Specify a mathematical expression for the value argument and a base (2, 8, 10, or 16) for the tobase argument.

Convert.ToByte(value): This function converts the value specified by the value argument to the Byte type.

Convert.ToInt16(value): This function converts the value specified by the value argument to the Short type.

Convert.ToInt32(value): This function converts the value specified by the value argument to the Integer type.

Convert.ToInt64(value): This function converts the value specified by the value argument to the Long type.

■Reference

To return a hexadecimal string of the integer variable type, the method of the variable can also be used.

Variable name.ToString("X") ••• This function returns a hexadecimal string in uppercase, for example "10AD78FC".

Variable name.ToString("x") ••• This function returns a hexadecimal string in lowercase, for example "10ad78fc".

■Note

When a conversion has failed, an error occurs. An exception handling is required for the conversion part.

```
Try
sHexCode = Convert.ToString(nValue, "16")
Catch ex As Exception
sHexCode = "0"
End Try
```

Appendix 1.7 List of controls

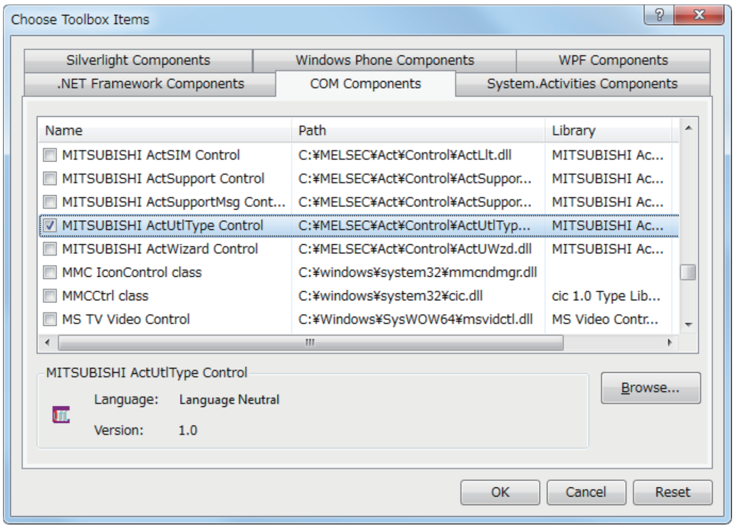
When arranging controls in the form, use the buttons in the toolbox.

Control	Function
Label	Used to display text and cannot be edited by the user. Text(property): Gets or sets the text content of the Label control.
TextBox	Used to display information entered by the user at run time, or assigned to the Text property of the control at design or run time. Text(property): Text entered into the text box control is contained. ReadOnly(property): Prevents users from editing text box contents.
GroupBox	Used to provide an identifiable grouping for other controls.
Button	Used to begin, interrupt, or end a process. Click(event): Occurs when a Button is clicked.
CheckBox	Displays a check mark when it is selected. It is commonly used to present a Yes/No or True/False selection to the user. You can use check box controls in groups to display multiple choices from which the user can select one or more. Checked(property): Gets or set a value indicating whether the CheckBox is in the checked state. CheckedChanged(event): Occurs when the value of the Checked property changes.
RadioButton	Used to display options, usually in option button groups, from which the user can choose one. Checked(property): Gets or sets a value indicating whether the control is checked. CheckedChanged(event): Occurs when the value of the Checked property changes.
ListBox	Displays a list of items from which the user can select one or more. MultiColumn(property): Gets or sets a value indicating whether the ListBox supports multiple columns. SelectedItem(property): Gets or sets the currently selected item in the ListBox. Items.Add(method): Adds an item to the list of items for a ListBox. SelectedIndexChanged(event): Occurs when the SelectedIndex property or the SelectedIndices collection has changed.
ComboBox	Allows the user to select an item either by typing text into the combo box, or by selecting it from the list. DropDownStyle(property): Gets or sets a value specifying the style of the combo box. SelectedItem(property): Gets or sets currently selected item in the ComboBox. SelectedIndex(property): Gets or sets the index specifying the currently selected item. Items.Add(method): Adds an item to the list of items for a ComboBox. SelectedIndexChanged(event): Occurs when the SelectedIndex property has changed.

■Adding a control

Controls not prepared in the standard toolbox can be added or deleted by selecting [Choose Toolbox Items] from [TOOLS] menu.

1. Select [TOOLS] ⇨ [Choose Toolbox Items].
2. Check the check box of the control to be added and select [OK].
3. The control is added in the lowest part of the tab selected in "Toolbox".



Name	Path	Library
□ MITSUBISHI ActSIM Control	C:¥MELSEC¥Act¥Control¥ActLlt.dll	MITSUBISHI Ac...
□ MITSUBISHI ActSupport Control	C:¥MELSEC¥Act¥Control¥ActSuppor...	MITSUBISHI Ac...
□ MITSUBISHI ActSupportMsg Cont...	C:¥MELSEC¥Act¥Control¥ActSuppor...	MITSUBISHI Ac...
☑ MITSUBISHI ActUtlType Control	C:¥MELSEC¥Act¥Control¥ActUtlTyp...	MITSUBISHI Ac...
□ MITSUBISHI ActWizard Control	C:¥MELSEC¥Act¥Control¥ActUWzd.dll	MITSUBISHI Ac...
□ MMC IconControl class	C:¥windows¥system32¥mmcndmgr.dll	
□ MMCCtrl class	C:¥windows¥system32¥cic.dll	cic 1.0 Type Lib...
□ MS TV Video Control	C:¥Windows¥SysWOW64¥msvidctl.dll	MS Video Contr...

MITSUBISHI ActUtlType Control

Language: Language Neutral

Version: 1.0

Point

- Add the control of "MITSUBISHI Act ..." when using MX Component.
- Use "MITSUBISHI ActUtlType Control" in the exercises.

Appendix 1.8 Terms

Term	Description
Form	A window that forms the basis of the interface of the application. An application consists of at least one form.
Control	An object that is contained within form objects (such as a button and text box)
Object	A generic term for the objects operated by users, such as forms and controls
Property	An attribute of the object. Setting a value to the properties allows users to define features of appearance of an object, such as the size, color, display position in the window, and status whether the size can be changed.
Event	An operation such as clicking the mouse and pressing keys. Available events depend on the object type. In an application, processing corresponding to the event is described in the Visual Basic code. Events occur when users operate an object. Some events occur from the program or the system side.
Method	A keyword used to control the objects. The object (a noun) is followed by the method (a verb).
Solution	A conceptual container that manages multiple programs
Class	A reference type that defines the operations that an object can perform (methods, events, or properties) and the data that the object contains. It can also inherit and reuse code defined in other classes.
Statement	A complete instruction. A declaration statement names a variable, constant, or procedure and an executable statement performs actions and can call a method or function.
Reference	To use Public elements defined in another project, a reference needs to be set first to that project's assembly or type library.
Assembly	A fundamental unit of physical code grouping used for several purposes: deployment, type identity, versioning, reference scope, and security
Startup project	A project that Visual Studio will build and debug first
Configuration manager	A dialog box used to create and edit solution build configurations and project configurations
Debugging	The process of finding and fixing errors in a program. Processing of the program can be stopped (broken) during debugging as needed.
Step Into	A command that executes code one statement at a time. If the statement is a call to a procedure, the next statement displayed is the first statement in the procedure.
Step Over	A command that executes code one statement at a time. The next statement displayed is the next statement in the current procedure regardless of whether the current statement is a call to another procedure.
Step Out	A command that executes the remaining lines of a function in which the current execution point lies. All of the code is executed between the current and the final execution points.
Namespace	A fundamental unit of logical code grouping. Namespaces are a way of grouping type names and reducing the chance of name collisions.
Build	A process to create an execution program by compiling the source program and linking with libraries.

Appendix 2 MX Component

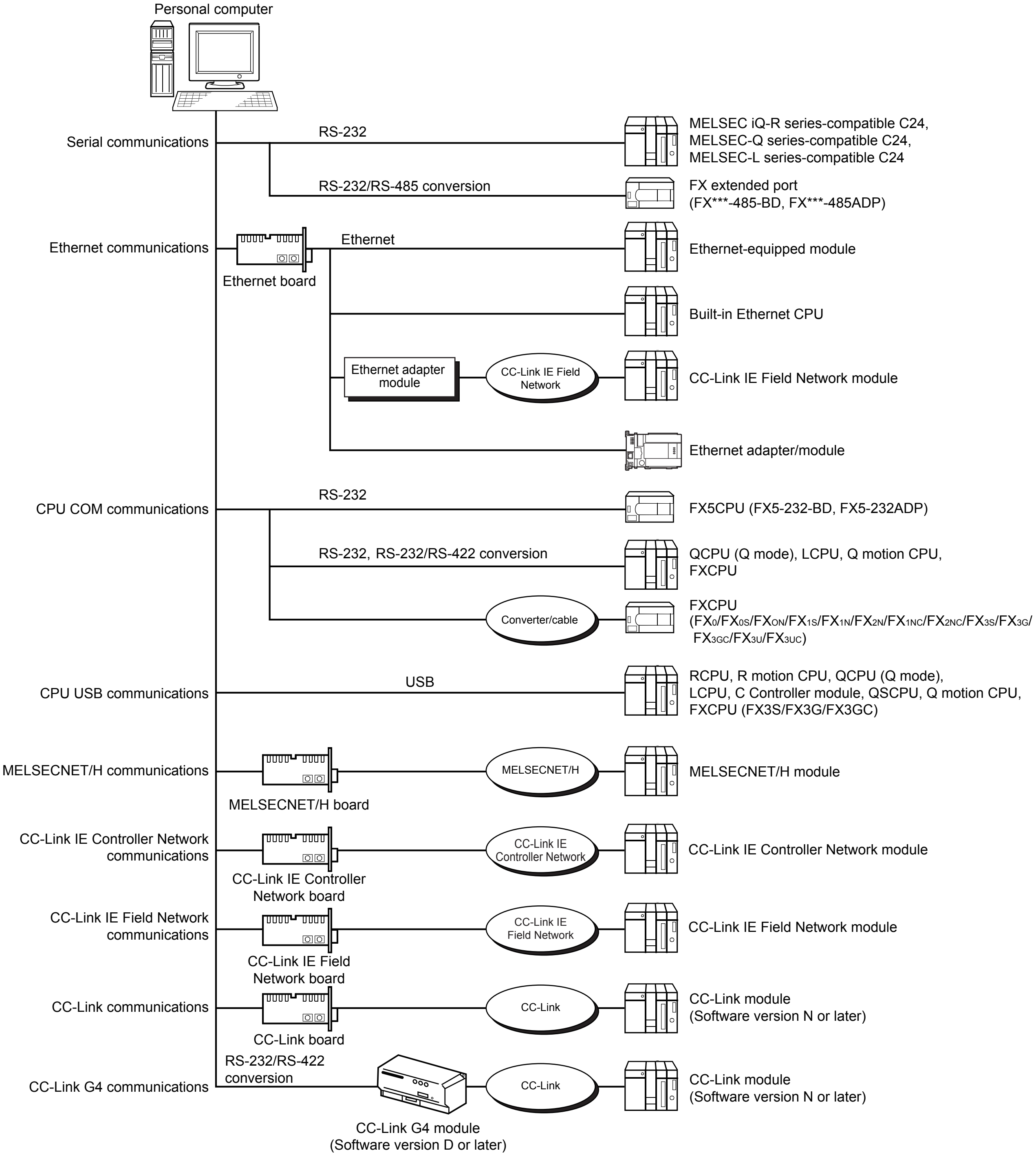
Appendix 2.1 Functions of MX Component

Using MX Component with an application created with Visual Basic®, Visual C++®, VBScript, or VBA allows to access the CPU module over various communication paths.

In this textbook, create an application with Visual Basic® 2012 and access the CPU module over Ethernet.

Support of a wide range of communication paths for programmable controller

A wide range of communication paths to the programmable controller are supported to enable the user to configure up a system as desired.


Personal computer
Serial communications
RS-232
MELSEC iQ-R series-compatible C24,
MELSEC-Q series-compatible C24,
MELSEC-L series-compatible C24
RS-232/RS-485 conversion
FX extended port
(FX***-485-BD, FX***-485ADP)
Ethernet communications
Ethernet board
Ethernet
Ethernet-equipped module
Built-in Ethernet CPU
Ethernet adapter module
CC-Link IE Field Network
CC-Link IE Field Network module
Ethernet adapter/module
CPU COM communications
RS-232
FX5CPU (FX5-232-BD, FX5-232ADP)
RS-232, RS-232/RS-422 conversion
QCPU (Q mode), LCPU, Q motion CPU,
FXCPU
Converter/cable
FXCPU
(FX0/FX0S/FX0N/FX1S/FX1N/FX2N/FX1NC/FX2NC/FX3S/FX3G/
FX3GC/FX3U/FX3UC)
CPU USB communications
USB
RCPU, R motion CPU, QCPU (Q mode),
LCPU, C Controller module, QSCPU, Q motion CPU,
FXCPU (FX3S/FX3G/FX3GC)
MELSECNET/H communications
MELSECNET/H board
MELSECNET/H
MELSECNET/H module
CC-Link IE Controller Network communications
CC-Link IE Controller Network board
CC-Link IE Controller Network
CC-Link IE Controller Network module
CC-Link IE Field Network communications
CC-Link IE Field Network board
CC-Link IE Field Network
CC-Link IE Field Network module
CC-Link communications
CC-Link board
CC-Link
CC-Link module
(Software version N or later)
CC-Link G4 communications
RS-232/RS-422 conversion
CC-Link G4 module
(Software version D or later)
CC-Link
CC-Link module
(Software version N or later)

Substantial improvement in user's development efficiency

MX Component comes with the wizard type communication setup utility.

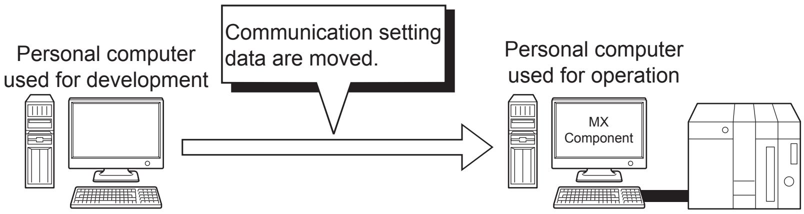
By simply setting dialog-based communication settings on the screen, the user can achieve communication settings to access the programmable controller CPU to communicate with.

Once the communication settings are set, stations can be accessed by merely specifying the logical station number of the programmable controller CPU stored on the communication setup utility.

Save and read of communication settings

MX Component features the functions to save and read the communication settings set on the communication setup utility.

The set data can easily be moved from the personal computer used for development to that used for operation.[*1]


Personal computer used for development
Communication setting data are moved.
Personal computer used for operation
MX Component

Enter and save communication setting data.

Read communication setting data and establish communication path.

↓

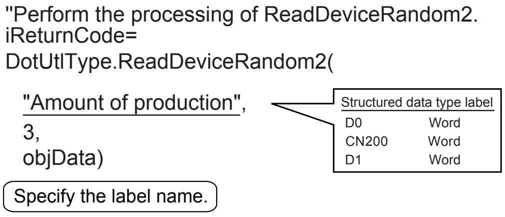
Shorten the time for operations

[*1] MX Component must be installed in both the personal computer used for development and the actually used personal computer.

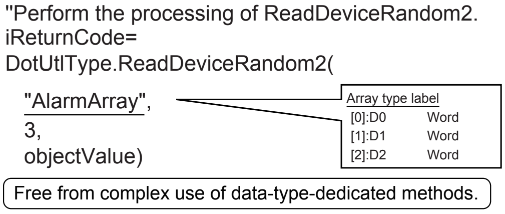
Label function

This function allows users to create a program with labels.

Programs can be created without considering device numbers, and devices can be read or written by using labels names.



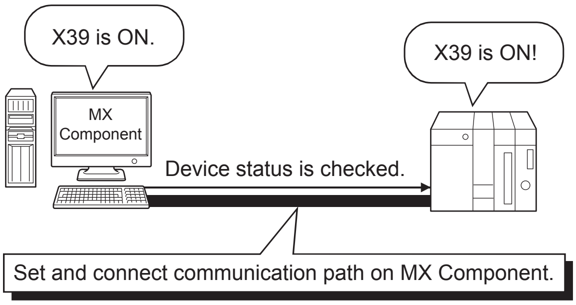
```
"Perform the processing of ReadDeviceRandom2.
iReturnCode=
DotUtlType.ReadDeviceRandom2(
    "Amount of production",
    3,
    objData)
```



```
"Perform the processing of ReadDeviceRandom2.
iReturnCode=
DotUtlType.ReadDeviceRandom2(
    "AlarmArray",
    3,
    objectValue)
```

Device monitor function

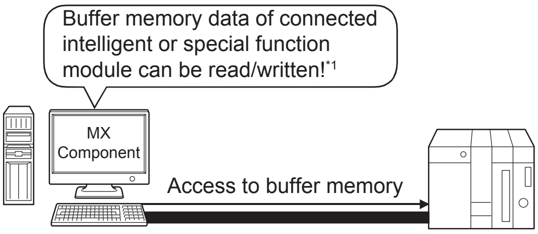
Utilizing the PLC monitor utility enables users to monitor the status of the specified device and change its data.[*1]


X39 is ON.
X39 is ON!
MX Component
Device status is checked.
Set and connect communication path on MX Component.

[*1] Device data of QSCPU cannot be changed.

Access to buffer memory of special function module

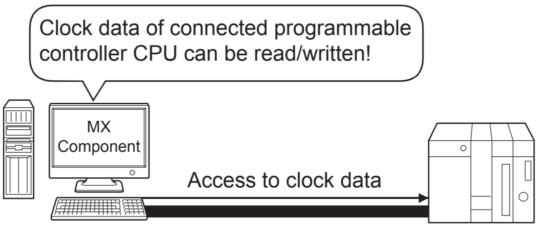
Not only the devices of the programmable controller CPU but also the buffer memory of an intelligent function or special function module can be accessed.


Buffer memory data of connected intelligent or special function module can be read/written!*1
MX Component
Access to buffer memory

*1 Buffer memory data of QSCPU cannot be written.

Read/write of programmable controller CPU clock data

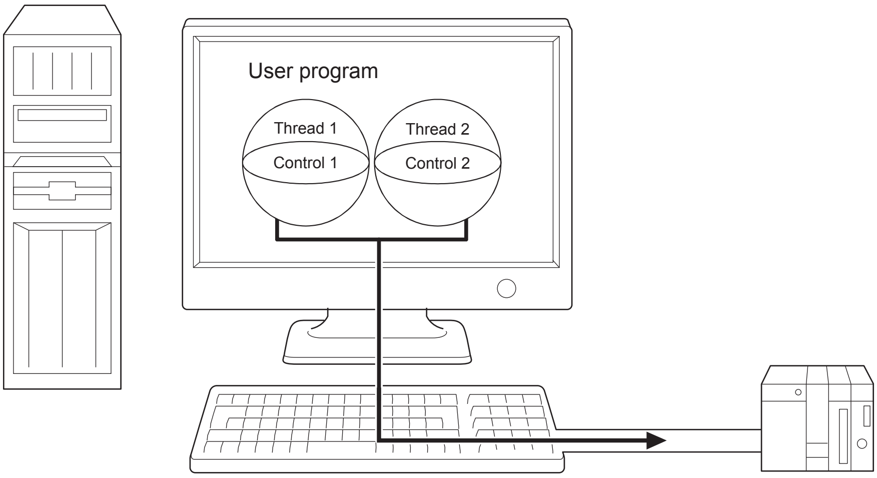
The clock data of the programmable controller CPU connected to the personal computer can be read/written.*1


Clock data of connected programmable controller CPU can be read/written!
MX Component
Access to clock data

*1 Clock data of QSCPU cannot be written.

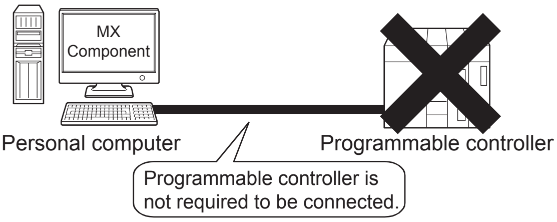
Multi-thread communications

The same communication path can be accessed from multiple threads at the same time.



The simulator function for offline debugging

By using the simulation function, debugging can be performed on a single personal computer without connecting the programmable controller.



■Software required for the simulation function

- GX Developer and GX Simulator are separately required to use GX Simulator[*1].
- GX Works2 is required to use GX Simulator2[*1].
  The maximum of four projects can be simulated simultaneously.
- GX Works3 is required to use GX Simulator3.
  The number of projects of GX Simulator3 can simultaneously simulate depends on the number of starts of GX Works3.

[*1] This function is not supported by QSCPU.

■When using the simulation function (MT Simulator2) of MT Developer2[*1]

- MT Developer2 is required to use MT Simulator2.
- The maximum of three projects can be simulated simultaneously.

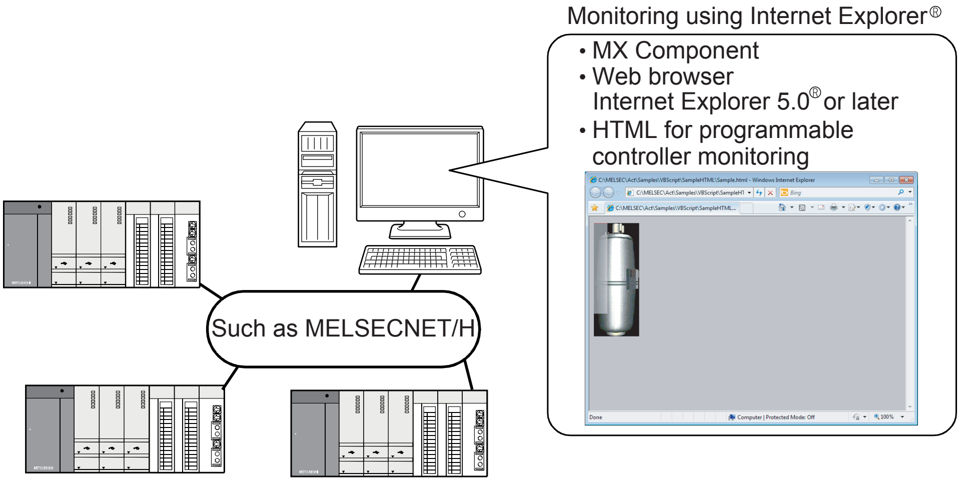
[*1] This function is supported by Q motion CPU only.

A wide variety of programming languages supported

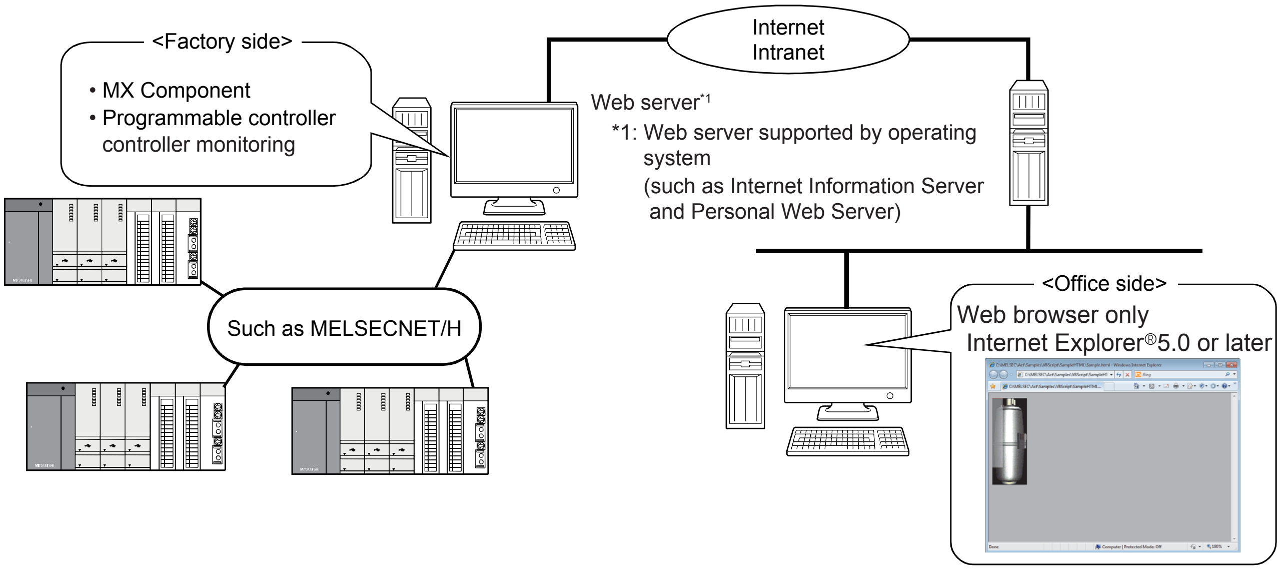
MX Component supports VBScript and VBA as well as Visual Basic®, Visual C++®, and Visual C#®.

■Creation of monitoring page by using VBScript

- Monitoring page can be created in HTML format
  A graphical monitoring home page (HTML format) can be created by using the text editor.
  Visual Basic®, Visual C++®, and Visual C#® are not required to be purchased.


Monitoring using Internet Explorer®
• MX Component
• Web browser
Internet Explorer 5.0® or later
• HTML for programmable controller monitoring
Such as MELSECNET/H

- Using ASP function for monitoring over Internet/intranet
  Using the ASP function of VBScript and releasing the Web pages on the factory side (side which monitors data using MX Component) enables the programmable controller device status or remote operation for an error to be performed from a remote location or during business trip over the Internet/intranet by merely specifying the factory side URL on Internet Explorer®.

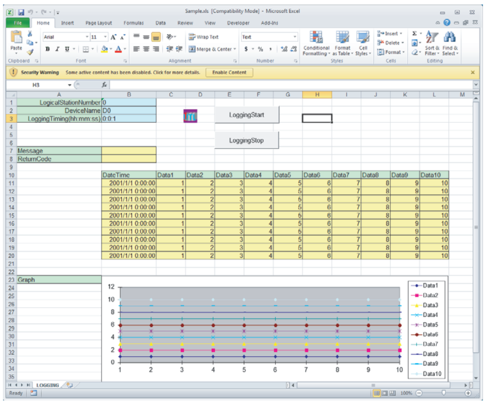

<Factory side>
• MX Component
• Programmable controller
controller monitoring
Internet
Intranet
Web server*1
*1: Web server supported by operating system
(such as Internet Information Server and Personal Web Server)
Such as MELSECNET/H
<Office side>
Web browser only
Internet Explorer®5.0 or later

■VBA-driven data collection and monitoring function

Programming using VBA allows Excel or Access functions to be utilized to create an application for providing a real-time graph display.

Device data of the programmable controller can be logged and device data can be sampled/saved in real time.

<Excel>

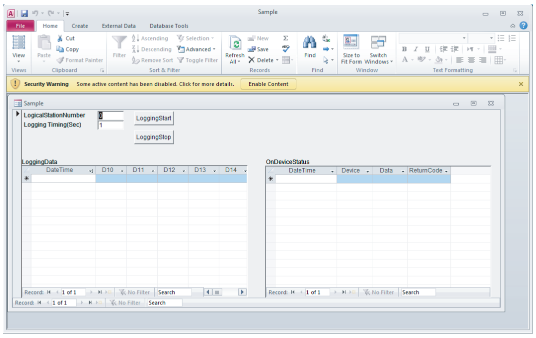


Security Warning Some active content has been disabled. Click for more details. Enable Content

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

<Access>



Security Warning Some active content has been disabled. Click for more details. Enable Content

LogicalStationNumber 0
Logging Timing(Sec) 1

DateTime	D10	D11	D12	D13	D14

DateTime	Device	Data	ReturnCode

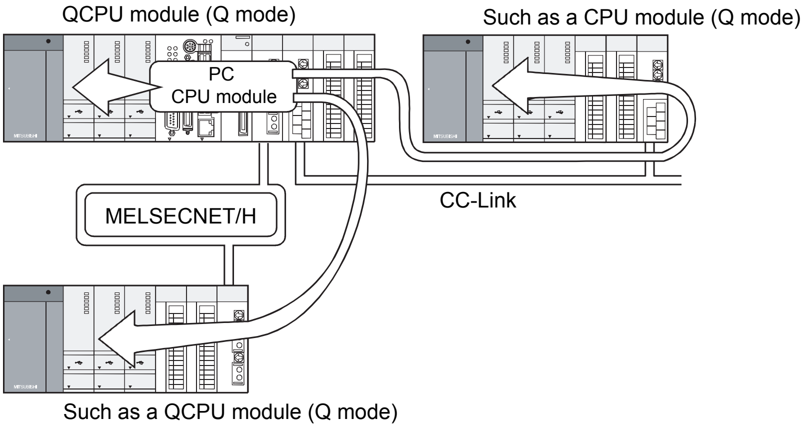
Compatibility with multi-CPU system of QCPU (Q mode)

Setting the communication setup utility or control properties enables access to the multi-CPU system.

Operability on personal computer CPU module

MELSEC-Q series bus communications from the personal computer CPU module enables access to the QCPU module (Q mode) on the same base.

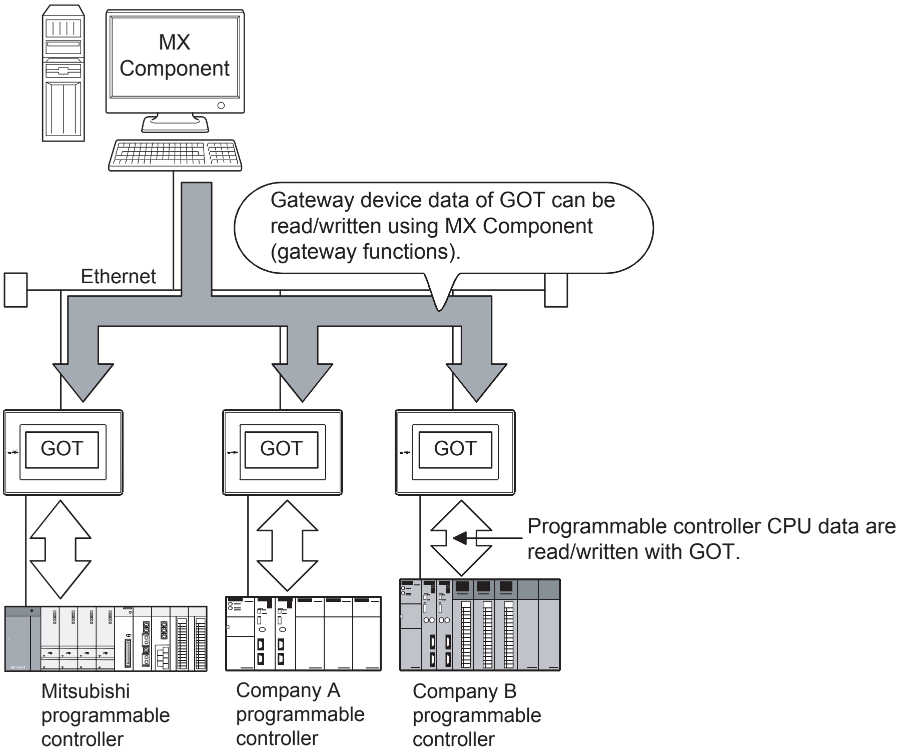
Using the MELSECNET/H communication control and CC-Link communication control enables access to other stations via the MELSECNET/H module and CC-Link module controlled by the PC CPU module.



Accessibility to gateway devices of GOT

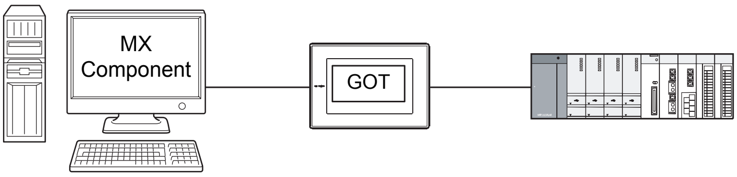
The gateway device data of GOT can be read/written by using the gateway function communications of MX Component.

The device data of the programmable controller CPU that is being monitored by GOT can be read/written by reading/writing the gateway device data of GOT.


MX Component
Gateway device data of GOT can be read/written using MX Component (gateway functions).
Ethernet
GOT
GOT
GOT
Programmable controller CPU data are read/written with GOT.
Mitsubishi programmable controller
Company A programmable controller
Company B programmable controller

Compatibility with GOT transparent function

Using the GOT transparent function, the programmable controller CPU can be accessed via GOT.



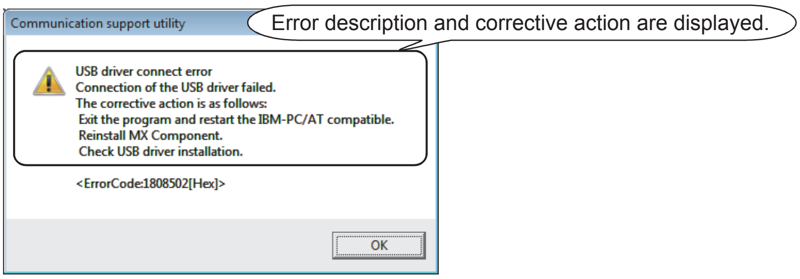
Reduction of error description search time

The ActSupport control for troubleshooting function is supported.

The error description and corrective action are displayed within the user application by only specifying the error code.

Therefore, checking error description or corrective action on the programming manual is not required when an error occurs in a control.

<Example of displaying error description on message box>



Communication support utility

USB driver connect error
Connection of the USB driver failed.
The corrective action is as follows:
Exit the program and restart the IBM-PC/AT compatible.
Reinstall MX Component.
Check USB driver installation.

<ErrorCode:1808502[Hex]>

Error description and corrective action are displayed.

Appendix 2.2 Operation procedure

This section describes the selection of the MX Component development type and the procedure for creating user applications.

For details, refer to the following.

📖 MX Component Version 4 Operating Manual

Selecting development type

When using MX Component to create user applications, select the utility setting type or program setting type before creating a user application.

The following describes the utility setting type and program setting type.

■Utility setting type

Set the communication settings by using the communication setting wizard.

With the communication setup utility, a user program can be created without being aware of the complicated parameters of any communication.

In the user program, the communication line can be connected by simply setting the logical station number set on the communication setting wizard for the ACT control or the .NET control property or into a user program.

■Program setting type

A user program is created without using the communication setup utility.

Set the ACT control settings for the corresponding communications on the property window directly or within the user program.

The required settings for the properties differ depending on the ACT control.

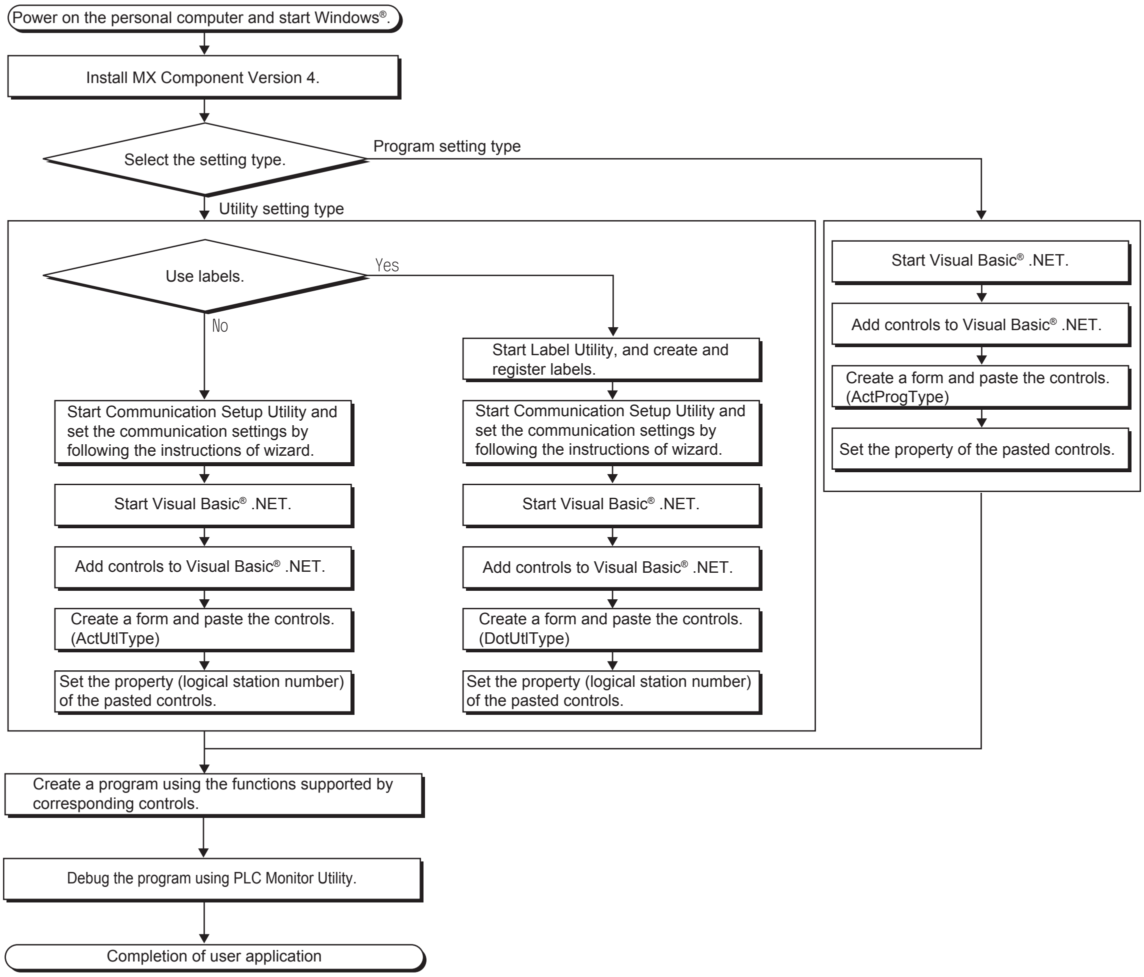
■Comparison

The following table compares the utility setting type and program setting type.

Item	Utility setting type	Program setting type
Feature	Communication settings can be set easily by using the communication setting wizard. In program creation, communications can be performed by merely setting the parameter (logical station number) on the communication setting wizard. (The number of development processes can be reduced.)	All communication settings can be set in the user program. Communication settings can be changed flexibly in the user program.
Used ACT control	ActUtlType, ActMLUtlType	ActProgType, ActMLProgType
Used .NET control	DotUtlType (when labels are used)	-
Communication setup utility	Used	Not used
How to connect the PLC monitor utility	Select the logical station number.	Change the settings every connection. (Use the wizard.)

Procedure for creating a user application (when using Visual Basic® .NET)

The following figure shows the creation procedure by using Visual Basic® .NET.


Power on the personal computer and start Windows®.
Install MX Component Version 4.
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.
Start Communication Setup Utility and set the communication settings by following the instructions of wizard.
Start Visual Basic® .NET.
Start Visual Basic® .NET.
Add controls to Visual Basic® .NET.
Add controls to Visual Basic® .NET.
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.
Set the property (logical station number) of the pasted controls.
Start Visual Basic® .NET.
Add controls to Visual Basic® .NET.
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.
Debug the program using PLC Monitor Utility.
Completion of user application

Appendix 2.3 List of controls

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

■ACT control

The following table lists the ActiveX controls supported by MX Component.

Data can be accessed by using devices.

DLL name	Included control name		Application
	For VB, VC++, VC#, VBA	For VBScript	
ActUtlType.dll	ActUtlType	ActMLUtlType[*1]	The utility setting type control which is used to create a user program by using the communication setup utility.
ActProgType.dll[*2]	ActProgType	ActMLProgType[*1]	The program setting type control which is used to create a user program without using the communication setup utility.
ActSupportMsg.dll	ActSupportMsg	ActMLSupportMsg	This control is used for the troubleshooting function.

*1 Communications are disabled if the communication path is a modem.

*2 This function is not applicable to inverter communications/robot controller communications.

■.NET control

The following table lists the .NET controls supported by MX Component.

Data can be accessed by using labels.

DLL name	Included control name	Application
	For VB, VC++, VC#	
DotUtlType.dll	DotUtlType	The utility setting type control which is used to create a user program by using the communication setup utility.
DotSupportMsg.dll	DotSupportMsg	This control is used for the troubleshooting function.

Appendix 2.4 List of functions

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

Function name	Feature
Open	Opens a communication line.
Close	Closes a communication line.
ReadDeviceBlock	Reads devices in bulk. (4-byte data)
WriteDeviceBlock	Writes devices in bulk. (4-byte data)
ReadDeviceRandom	Reads devices randomly. (4-byte data)
WriteDeviceRandom	Writes devices randomly. (4-byte data)
SetDevice	Sets one point of the device. (4-byte data)
GetDevice	Gets data of one point of the device. (4-byte data)
ReadBuffer	Reads data from the buffer memory.
WriteBuffer	Writes data to the buffer memory.
GetClockData	Reads clock data from the programmable controller CPU.
SetClockData	Writes clock data to the programmable controller CPU.
GetCpuType	Reads a programmable controller CPU model.
SetCpuStatus	Sets the remote RUN/STOP/PAUSE of the programmable controller CPU.
EntryDeviceStatus	Registers a device status monitor.
FreeDeviceStatus	Deregisters a device status monitor.
OnDeviceStatus	Announces an event.
ReadDeviceBlock2	Reads devices in bulk. (2-byte data)
WriteDeviceBlock2	Writes devices in bulk. (2-byte data)
ReadDeviceRandom2	Reads devices randomly. (2-byte data)
WriteDeviceRandom2	Writes devices randomly. (2-byte data)
SetDevice2	Sets one point of the device. (2-byte data)
GetDevice2	Gets data of one point of the device. (2-byte data)
Connect	Connects a telephone line.
Disconnect	Disconnects a telephone line.
GetErrorMessage	Displays an error definition and corrective action.

Point

- Consideration for using RnSFSCPU
  To protect the safety programmable controller system, the functions to write data to buffer memory and write data to safety devices in safety mode cannot be executed.
- Consideration for using QSCPU
  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.

Appendix 2.5 Details of functions (For ACT control)

Open (Opening communication line)

■Applicable controls

This function is applicable to the Act(ML)ProgType control and the Act(ML)UtlType control.

■Feature

This function opens the communication line.

■Format (Dispatch interface)

- Visual C++® .NET(MFC), VBA

lRet = object.Open()

Long	lRet	Return value	Output

- VBScript

varRet = object.Open()

VARIANT	varRet	Return value (LONG type)	Output

- Visual Basic® .NET

lRet = object.Open()

Integer	lRet	Return value	Output

- Visual C++® .NET

iRet = object.Open()

int	iRet	Return value	Output

- Visual C#® .NET

iRet = object.Open()

int	iRet	Return value	Output

■Format (Custom interface)

- Visual C++® .NET(MFC)

hResult = object.Open(*lplRetCode)

HRESULT	hResult	Return value of COM	Output
LONG	*lplRetCode	Return value of communication function	Output

■Description

Lines are connected according to the setting value of the Open function property.

■Return value

Normal end: 0 is returned.

Abnormal end: Any value other than 0 is returned.

Point

- When modem communications are 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 communications is set to the ActCpuType property, the Open function may complete successfully.
  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.

Close (Closing communication line)

■Applicable controls

This function is applicable to the Act(ML)ProgType control and the Act(ML)UtlType control.

■Feature

This function closes the communication line.

■Format (Dispatch interface)

- Visual C++® .NET(MFC), VBA

lRet = object.Close()

Long	lRet	Return value	Output

- VBScript

varRet = object.Close()

VARIANT	varRet	Return value (LONG type)	Output

- Visual Basic® .NET

lRet = object.Close()

Integer	lRet	Return value	Output

- Visual C++® .NET

iRet = object.Close()

int	iRet	Return value	Output

- Visual C#® .NET

iRet = object.Close()

int	iRet	Return value	Output

■Format (Custom interface)

- Visual C++® .NET(MFC)

hResult = object.Close(*lplRetCode)

HRESULT	hResult	Return value of COM	Output
LONG	*lplRetCode	Return value of communication function	Output

■Description

The line connected using the Open function is closed.

■Return value

Normal end: 0 is returned.

Abnormal end: Any value other than 0 is returned.

ReadDeviceRandom (Reading devices randomly)

■Applicable controls

This function is applicable to the Act(ML)ProgType control and the Act(ML)UtlType control.

■Feature

This function reads devices randomly.

It reads monitor types of inverters/robot controllers randomly.

■Format (Dispatch interface)

- VBA

lRet = object.ReadDeviceRandom(szDeviceList, lSize, lData(0))

Long	lRet	Return value	Output
String	szDeviceList	Device name/monitor type	Input
Long	lSize	Number of read points	Input
Long	lData(n)	Read device value	Output

- Visual C++® .NET(MFC)

lRet = object.ReadDeviceRandom(szDeviceList, lSize, *lplData)

Long	lRet	Return value	Output
CString	szDeviceList	Device name/monitor type	Input
Long	lSize	Number of read points	Input
Long	*lplData	Read device value	Output

- VBScript

varRet = object.ReadDeviceRandom(varDeviceList, varSize, lpvarData)

VARIANT	varRet	Return value (LONG type)	Output
VARIANT	varDeviceList	Device name/monitor type (string type)	Input
VARIANT	varSize	Number of read points (LONG type)	Input
VARIANT	lpvarData	Read device value (LONG array type)	Output

- Visual Basic® .NET

lRet = object.ReadDeviceRandom(szDeviceList, iSize, iData(0))

Integer	lRet	Return value	Output
String	szDeviceList	Device name/monitor type	Input
Integer	iSize	Number of read points	Input
Integer	iData(n)	Read device value	Output

- Visual C++® .NET

iRet = object.ReadDeviceRandom(*szDeviceList, iSize, *iplData)

int	iRet	Return value	Output
String	*szDeviceList	Device name/monitor type	Input
int	iSize	Number of read points	Input
int	*iplData	Read device value	Output

- Visual C#® .NET

iRet = object.ReadDeviceRandom(szDevice, iSize, out iData[0])

int	iRet	Return value	Output
String	szDeviceList	Device name/monitor type	Input
int	iSize	Number of read points	Input
int[n]	iData	Read device value	Output

■Format (Custom interface)

- Visual C++® .NET(MFC)

hResult = object.ReadDeviceRandom(szDevice, lSize, *lplData, *lplRetCode)

HRESULT	hResult	Return value of COM	Output
BSTR	szDevice	Device name/monitor type	Input
LONG	lSize	Number of read points	Input
LONG	*lplData	Read device value	Output
LONG	*lplRetCode	Return value of communication function	Output

■Description

- Data of a device group/monitor type 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 string specified for the device list. The line feed symbol is not necessary to be suffixed to the last device.

Ex.

When using Visual Basic® .NET, VBA, VBScript: "D0" & vbLf & "D1" & vbLf & "D2"
When using Visual C++® .NET: D0\nD1\nD2
When using Visual C#® .NET: D0\nD1\nD2

- For lData (lplData or lpvarData), reserve arrays for more than the amount specified for lSize (varSize).
- For details on the items (monitor types) that can be monitored, refer to the following.
  📖 MX Component Version 4 Operating Manual

■How to specify devices

The following describes how to specify the 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
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 double word devices are specified

When using Visual Basic® .NET, VBA, VBScript: "LTN0" & vbLf & "LTN1" & vbLf & "LTN2"
When using Visual C++® .NET: LTN0\nLTN1\nLTN2
When using Visual C#® .NET: LTN0\nLTN1\nLTN2

Upper 2 bytes	Lower 2 bytes
H of LTN0	L of LTN0
H of LTN1	L of LTN1
H of LTN2	L of LTN2

Example 3: 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
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 4: 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

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 5: When 8-bit devices including EG are specified (Total number of points: 3 points)

The following is the example when 8-bit device areas (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

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>

Example 6: When monitor types (1, 2, and 5) of inverter are specified (Total number of points: 3 points)

When using Visual Basic® .NET, VBA, VBScript: "1" & vbLf & "2" & vbLf & "5"

When using Visual C++® .NET: 1\n2\n5

When using Visual C#® .NET: 1\n2\n5

Upper 2 bytes	Lower 2 bytes
H of 1	L of 1
H of 2	L of 2
H of 5	L of 5

Example 7: When monitor types (223.102.A and 223.103.B) of robot controller are specified (Total number of points: 2 points)

Format: (Request ID).(Data type).(Argument)

When using Visual Basic® .NET, VBA, VBScript: "223.102.A" & vbLf & "223.103.B"

When using Visual C++® .NET: 223.102.A\n223.103.B

When using Visual C#® .NET: 223.102.A\n223.103.B

Upper 2 bytes	Lower 2 bytes
H of 223.102.A	L of 223.102.A
H of 223.103.B	L of 223.103.B

*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 devices are read randomly.

■Return value

Normal end: 0 is returned.

Abnormal end: Any value other than 0 is returned.

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, such as an application error, may occur.

WriteDeviceRandom (Writing devices randomly)

■Applicable controls

This function is applicable to the Act(ML)ProgType control and the Act(ML)UtlType control.[*1]

[*1] This function is not applicable to inverter communications/robot controller communications.

■Feature

This function writes devices randomly.

■Format (Dispatch interface)

- VBA

lRet = object.WriteDeviceRandom(szDeviceList, lSize, lData(0))

Long	lRet	Return value	Output
String	szDeviceList	Device name	Input
Long	lSize	Number of write points	Input
Long	lData(n)	Device value to be written	Input

- Visual C++® .NET(MFC)

lRet = object.WriteDeviceRandom(szDeviceList, lSize, *lplData)

Long	lRet	Return value	Output
CString	szDeviceList	Device name	Input
Long	lSize	Number of write points	Input
Long	*lplData	Device value to be written	Input

- VBScript

varRet = object.WriteDeviceRandom(varDeviceList, varSize,varData)

VARIANT	varRet	Return value (LONG type)	Output
VARIANT	varDeviceList	Device name (string type)	Input
VARIANT	varSize	Number of write points (LONG type)	Input
VARIANT	varData	Device value to be written (LONG array type)	Input

- Visual Basic® .NET

lRet = object.WriteDeviceRandom(szDeviceList, iSize, iData(0))

Integer	lRet	Return value	Output
String	szDeviceList	Device name	Input
Integer	iSize	Number of write points	Input
Integer	iData(n)	Device value to be written	Input

- Visual C++® .NET

iRet = object.WriteDeviceRandom(*szDeviceList, iSize, *iplData)

int	iRet	Return value	Output
String	*szDeviceList	Device name	Input
int	iSize	Number of write points	Input
int	*iplData	Device value to be written	Input

- Visual C#® .NET

iRet = object.WriteDeviceRandom(szDevice, iSize, out iData[0])

int	iRet	Return value	Output
String	szDevice	Device name	Input
int	iSize	Number of write points	Input
int[n]	iData	Device value to be written	Input

■Format (Custom interface)

- Visual C++® .NET(MFC)

hResult = object.WriteDeviceRandom(szDevice, lSize, *lplData, *lplRetCode)

HRESULT	hResult	Return 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	Return value of communication function	Output

■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 string specified for the device list.
  The line feed symbol is not necessary to be suffixed to the last device.

Ex.

When using Visual Basic® .NET, VBA, VBScript: "D0" & vbLf & "D1" & vbLf & "D2"
When using Visual C++® .NET: D0\nD1\nD2
When using Visual C#® .NET: D0\nD1\nD2

- For lData (lplData or varData), reserve arrays for more than the amount specified for lSize (varSize).

■How to specify devices

The following describes how to specify the 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
When using Visual C#® .NET: M0\nD0\nK8M0

Upper 2 bytes	Lower 2 bytes
Not used	M0
	D0
M16 to M31[*1]	M0 to M15[*1]

Example 2: When double word devices are specified

When using Visual Basic® .NET, VBA, VBScript: "LTN0" & vbLf & "LTN1" & vbLf & "LTN2"
When using Visual C++® .NET: LTN0\nLTN1\nLTN2
When using Visual C#® .NET: LTN0\nLTN1\nLTN2

Upper 2 bytes	Lower 2 bytes
H of LTN0	L of LTN0
H of LTN1	L of LTN1
H of LTN2	L of LTN2

Example 3: 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
When using Visual C#® .NET: D0\nCN200\nD1

Upper 2 bytes	Lower 2 bytes
Not used	D0
H of CN200	L of CN200
Not used	D1

Example 4: When devices including FD are specified (Total number of points: 3 points)

Upper 2 bytes	Lower 2 bytes
Not used	D0
	LL of FD0
	D1

Example 5: When 8-bit devices including EG are specified (Total number of points: 3 points)

The following is the example when 8-bit device areas (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

When using Visual C#® .NET: D0\nEG0\nD1

Upper 2 bytes	Lower 2 bytes	
Not used	D0	
	EG0	
	(E0001)	(E0000)
	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 devices are read randomly.

■Return value

Normal end: 0 is returned.

Abnormal end: Any value other than 0 is returned.

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, such as an application error, may occur.
- If a Q motion CPU is accessed, an error is returned.
- If the function is run against the safety device in safety mode of RnSFCPU, an error code 0x010A42A5 (an operation that cannot be performed in safety mode was performed) is returned.

Appendix 2.6 Details of functions (For .NET control)

Open (Opening communication line)

■Applicable control

This function is applicable to the DotUtlType control.

■Feature

This function opens the communication line.

■Format

- Visual C++® .NET

IRet = object.Open()

Integer	IRet	Return value	Output

- Visual C++® .NET

iRet = object.Open()

int	iRet	Return value	Output

- Visual C#® .NET

iRet = object.Open()

int	iRet	Return value	Output

■Description

Lines are connected according to the setting value of the Open function property.

■Return value

Normal end: 0 is returned.

Abnormal end: Any value other than 0 is returned.

Point

- When modem communications are 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 communications is set to the ActCpuType property, the Open function may complete successfully.
  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.

Close (Closing communication line)

■Applicable control

This function is applicable to the DotUtlType control.

■Feature

This function closes the communication line.

■Format

- Visual Basic® .NET

IRet = object.Close()

Integer	IRet	Return value	Output

- Visual C++® .NET

iRet = object.Close()

int	iRet	Return value	Output

- Visual C#® .NET

iRet = object.Close()

int	iRet	Return value	Output

■Description

The line connected using the Open function is closed.

■Return value

Normal end: 0 is returned.

Abnormal end: Any value other than 0 is returned.

ReadDeviceRandom (Reading devices randomly)

■Applicable control

This function is applicable to the DotUtlType control.

■Feature

This function reads devices randomly.

■Format

- Visual Basic® .NET

lRet = object.ReadDeviceRandom(szLabel, iSize, iData(0))

Integer	lRet	Return value	Output
String	szLabel	Label name	Input
Integer	iSize	Number of read points	Input
Integer	iData(n)	Read device value	Output

lRet = object.ReadDeviceRandom(szLabelList, iSize, iData(0))

Integer	lRet	Return value	Output
String	szLabelList(n)	Label list	Input
Integer	iSize	Number of read points	Input
Integer	iData(n)	Read device value	Output

- Visual C++® .NET

iRet = object.ReadDeviceRandom(*szLabel, iSize, *ipiData)

int	iRet	Return value	Output
String*	szLabel	Label name	Input
int	iSize	Number of read points	Input
int	iplData	Read device value	Output

iRet = object.ReadDeviceRandom(**szLabelList, iSize, *ipiData)

int	lRet	Return value	Output
String**	szLabelList	Label list	Input
int	iSize	Number of read points	Input
int	iData	Read device value	Output

- Visual C#® .NET

iRet = object.ReadDeviceRandom(ref szLabel, iSize, ref iData)

int	iRet	Return value	Output
String	szLabel	Label name	Input
int	iSize	Number of read points	Input
int[n]	iData	Read device value	Output

iRet = object.ReadDeviceRandom(ref szLabelList, iSize, ref iData)

int	iRet	Return value	Output
System.String[]	szLabelList	Label list	Input
int	iSize	Number of read points	Input
int[n]	iData	Read device value	Output

■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.

■How to specify devices

The following describes how to specify label names and device values to be read.

- The following data types can be specified for the label name.

Type class		Label data type	Label name format
Basic type		Bit, Word, Double Word, Float (Single Precision), Float (Double Precision), String[*1], String (Unicode)[*1], Time, Timer, Long timer, Counter, Long counter, Retentive Timer, Long 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
String (Unicode)	33	Number of label array elements multiplied by 33
Time	2	Number of label elements multiplied by 2
Timer, Long timer, Counter, Long counter, Retentive Timer, Long retentive timer	1	Number of label elements

- For the number of read points, specify the sum of the following values correspond to the elements according to the label data type.

<When the bit device and word device are specified>

Ex.

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 value

Upper 2 bytes	Lower 2 bytes	Applicable label
Not used (0 is stored.)	M0[*1]	LABEL1
	D0	LABEL2

[*1] 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>

Ex.

Read 3 points of data from the devices including CN200.[*2]

- Label setting

szLabelList		Data type	Device
[0]	LABEL1	Word	D0
[1]	LABEL2	Word	CN200
[2]	LABEL3	Word	D1

- Number of read points: 3
- Read device value

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

[*2] For FXCPU devices of CN200 and later, 4 bytes are read as 1 read point.

<When FD devices are specified (4-word device)>

Ex.

Read 3 points of data from the devices including FD0.

- Label setting

szLabelList		Data type	Device
[0]	LABEL1	Word	D0
[1]	LABEL2	Word	FD0
[2]	LABEL3	Word	D1

- Number of read points: 3
- Read device value

Upper 2 bytes	Lower 2 bytes	Applicable label
Not used (0 is stored.)	D0	LABEL1
	LL of FD0[*3]	LABEL2
	D1	LABEL3

[*3] Only lower 2 bytes are read. Data is not read from HH, HL, and LH (upper 6 bytes) of the specified devices.

<When the data type equivalent to 2 words or more is specified for label>

Ex.

Read 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 read points: 27
- Read device value

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[*4]	LABEL4
	D400	LABEL5
	D401	

*4 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 labels are specified>

Ex.

Read data from the devices by specifying array type labels.

- Label setting

szLabelList		Data type	Device
[0]	LABEL1	Bit (0..1)	M0
[1]	LABEL2	Word (0..1)	CN200
[2]	LABEL3	Double Word (0..1)	D0

- Number of read points: 8
- Read device value

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 Long timer, Long counter, or Long retentive timer type is specified>

Ex.

Read 3 points of data from LT0.

- Label setting

szLabelList		Data type	Device
[0]	LABEL	Long timer	LT0
[1]	LABEL	Long timer	LT1
[2]	LABEL	Long timer	LT2

- Number of read points: 3
- Read device value

Upper 2 bytes	Lower 2 bytes	Applicable label
H of LT0	L of LT0	LABEL[1]
H of LT1	L of LT1	LABEL[2]
H of LT2	L of LT2	LABEL[3]

<When structure type labels are specified>

Ex.

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 value

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[*5]	

[*5] 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>

Ex.

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 value

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	

■Return value

Normal end: 0 is returned.

Abnormal end: Any value other than 0 is returned.

Point

- The maximum number of read points that can be specified 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, such as an application error, 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.
- When system label Ver.2 is used, the data type defined in the label utility of MX Component must match the data type managed by MELSOFT Navigator.

  If the data type does not match, the read data length may be incorrect, or the array of the read device value may not correspond to the label name when multiple labels are specified.

WriteDeviceRandom (Writing devices randomly)

■Applicable control

This function is applicable to the DotUtlType control.

■Function

This function writes devices randomly.

■Format

- Visual Basic® .NET

Ret = object.WriteDeviceRandom(szLabel, iSize, iData(0))

Integer	lRet	Return value	Output
String	szLabel	Label name	Input
Integer	iSize	Number of write points	Input
Integer	iData(n)	Device value to be written	Input

Ret = object.WriteDeviceRandom(szLabelList, iSize, iData(0))

Integer	lRet	Return value	Output
String	szLabelList(n)	Label list	Input
Integer	iSize	Number of write points	Input
Integer	iData(n)	Device value to be written	Input

- Visual C++® .NET

iRet = object.WriteDeviceRandom(*szLabel, iSize, *ipiData)

int	iRet	Return value	Output
String*	szLabel	Label name	Input
int	iSize	Number of write points	Input
int*	iplData	Device value to be written	Input

iRet = object.WriteDeviceRandom(**szLabelList, iSize, *ipiData)

int	iRet	Return value	Output
String**	szLabelList	Label list	Input
int	iSize	Number of write points	Input
int*	iplData	Device value to be written	Input

- Visual C++® .NET

iRet = object.WriteDeviceRandom(ref szLabel, iSize, iData)

int	iRet	Return value	Output
String	szLabel	Label name	Input
int	iSize	Number of write points	Input
int[n]	iData	Device value to be written	Input

iRet = object.WriteDeviceRandom(ref szLabelList, iSize, iData)

int	iRet	Return value	Output
System.String[]	szLabelList	Label list	Input
int	iSize	Number of write points	Input
int[n]	iData	Device value to be written	Input

■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.

■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 the label name.

Type class		Label data type	Label name format
Basic type		Bit, Word, Double Word, Float (Single Precision), Float (Double Precision), String*1, String (Unicode)*1, Time, Timer, Long timer, Counter, Long counter, Retentive Timer, Long 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
String (Unicode)	33	Number of label array elements multiplied by 33
Time	2	Number of label elements multiplied by 2
Timer, Long timer, Counter, Long counter, Retentive Timer, Long retentive timer	1	Number of label elements

- Set the device values to be written as follows.

<When the bit device and word device are specified>

Ex.

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 value 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>

Ex.

Write 3 points of data to the devices including CN200.[*1]

- Label setting

szLabelList		Data type	Device
[0]	LABEL1	Word	D0
[1]	LABEL2	Word	CN200
[2]	LABEL3	Word	D1

- Number of write points: 3
- Device value 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

*1 For FXCPU devices of CN200 and later, 4 bytes are written as 1 write point.

<When FD devices are specified (4-word device)>

Ex.

Write 3 points of data to the devices including FD0.

- Label setting

szLabelList		Data type	Device
[0]	LABEL1	Word	D0
[1]	LABEL2	Word	FD0
[2]	LABEL3	Word	D1

- Number of write points: 3
- Device value to be written

Upper 2 bytes	Lower 2 bytes	Applicable label
Not used	D0	LABEL1
	LL of FD0[*2]	LABEL2
	D1	LABEL3

*2 Only lower 2 bytes can be set. "0" is written to HH, HL, and LH (upper 6 bytes) of the specified devices.

<When the data type equivalent to 2 words or more is specified for label>

Ex.

Write data by specifying labels of Double Word, Float (Single Precision), Float (Double Precision), String*3, 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 value 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*3	LABEL4
	D400	LABEL5
	D401	

*3 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 labels are specified>

Ex.

Write data to the devices by specifying array type labels.

- Label setting

szLabelList		Data type	Device
[0]	LABEL1	Bit (0..1)	M0
[1]	LABEL2	Word (0..1)	CN200
[2]	LABEL3	Double Word (0..1)	D0

- Number of write points: 8
- Device value 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 Long timer, Long counter, or Long retentive timer type is specified>

Ex.

Write 3 points of data from LT0.

- Label setting

szLabelList		Data type	Device
[0]	LABEL	Long timer	LT0
[1]	LABEL	Long timer	LT1
[2]	LABEL	Long timer	LT2

- Number of write points: 3
- Device value to be written

Upper 2 bytes	Lower 2 bytes	Applicable label
H of LT0	L of LT0	LABEL[1]
H of LT1	L of LT1	LABEL[2]
H of LT2	L of LT2	LABEL[3]

<When structure type labels are specified>

Ex.

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	Device	Device
[0]	LABEL1	STRUCT	L1	D0.0
			L2	D0
[1]	LABEL2	STRUCT	L1	M10
			L2	CN200

- Number of write points: 5
- Device value 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>

Ex.

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	Device	Device
[0]	LABEL1	STRUCT1 (0..1)	L1	X0
			L2	D0
[1]	LABEL2	STRUCT2	L1	M0
			L2	D100

- Number of write points: 9
- Device value 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	

■Return value

Normal end: 0 is returned.

Abnormal end: Any value other than 0 is returned.

Point

- The maximum number of write points that can be specified 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, such as an application error, 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.
- If a Q motion CPU is accessed, an error is returned.
- When system label Ver.2 is used, the data type defined in the label utility of MX Component must match the data type managed by MELSOFT Navigator.
  If the data type does not match, the read data length may be incorrect, or the array of the read device value may not correspond to the label name when multiple labels are specified.
- If the function is run against the safety device in safety mode of RnSFCPU, an error code 0x010A42A5 (an operation that cannot be performed in safety mode was performed) is returned.

Appendix 2.7 List of error codes

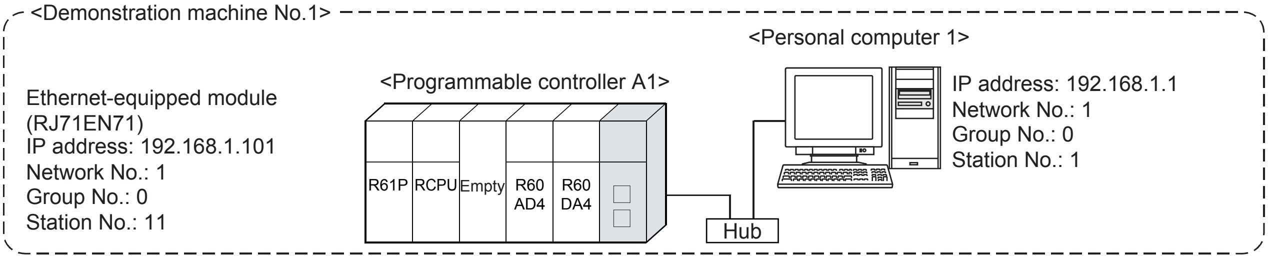
For error codes that are returned from the controls, CPU module, module, and network board, refer to the following.

📖 MX Component Version 4 Programming Manual

Appendix 3 Connecting GX Works3 and a CPU Module over Ethernet

This section describes how to connect GX Works3 directly to the Ethernet-equipped module (Ethernet connection) and access the CPU module from GX Works3 via the Ethernet-equipped module.

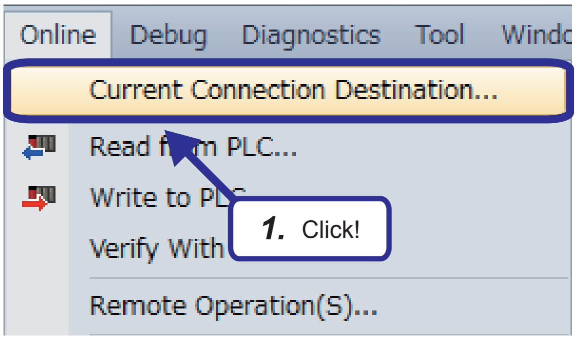
The demonstration machine No.1 of Exercise 1 is used for an example of how to change the connection path to the Ethernet connection.



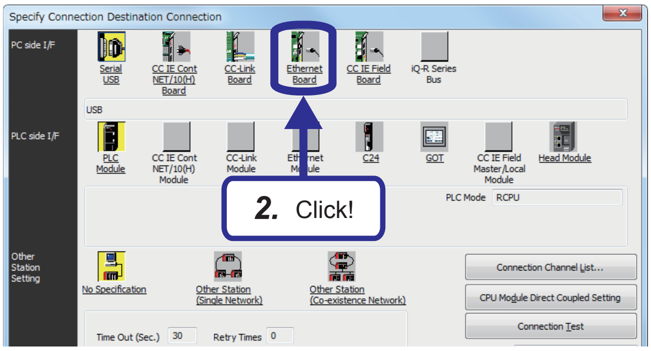
Point

- Before directly connecting GX Works3 to the Ethernet-equipped module, Ethernet parameters must be registered (written) to the CPU module on the station where the connection-target Ethernet-equipped module is mounted.
  Refer to Page 5 - 5 Starting GX Works3 to Page 5 - 17 Writing parameters and write the parameters to the CPU module in advance.

Operating procedure



1. Select [Online] ⇨ [Current Connection Destination] from the menu of the engineering tool.


Specify Connection Destination Connection

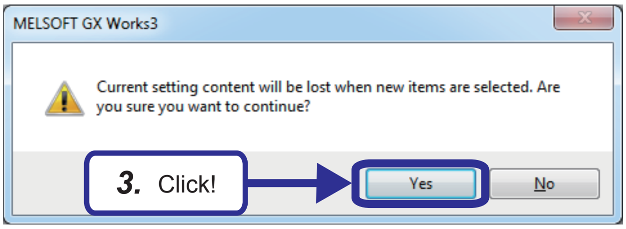
PLC Mode RCPU

Time Out (Sec.) 30 Retry Times 0

2. Click the [Ethernet Board] button on the "Specify Connection Destination Connection" window.

(To the next page)

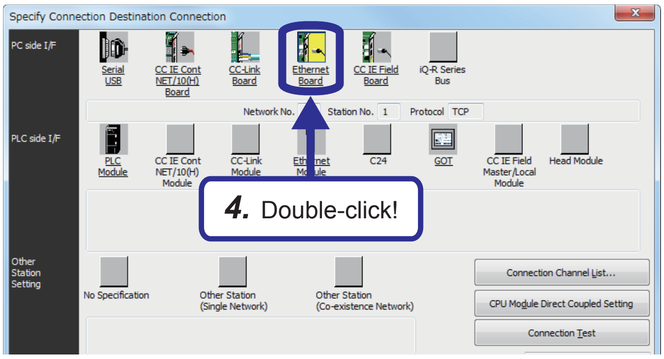
(From the previous page)



MELSOFT GX Works3

Current setting content will be lost when new items are selected. Are you sure you want to continue?

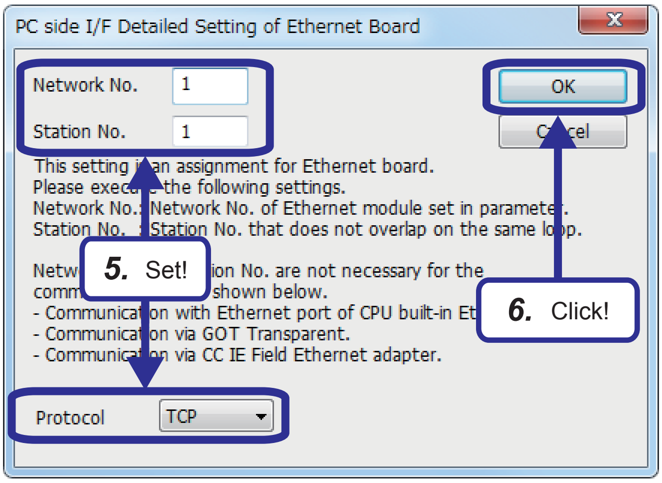
3. Click the [Yes] button.



Specify Connection Destination Connection

Network No. Station No. 1 Protocol TCP

4. Double-click the [Ethernet Board] button on the "Specify Connection Destination Connection" window.



PC side I/F Detailed Setting of Ethernet Board

Network No. 1

Station No. 1

This setting i an assignment for Ethernet board.
Please exec the following settings.
Network No.: Network No. of Ethernet module set in paramete.
Station No. : Station No. that does not overlap on the same lo p.

Netw ion No. are not necessary for the
comm shown below.
- Communicat on with Ethernet port of CPU built-in Et
- Communicat on via GOT Transparent.
- Communicat n via CC IE Field Ethernet adapter.

Protocol TCP

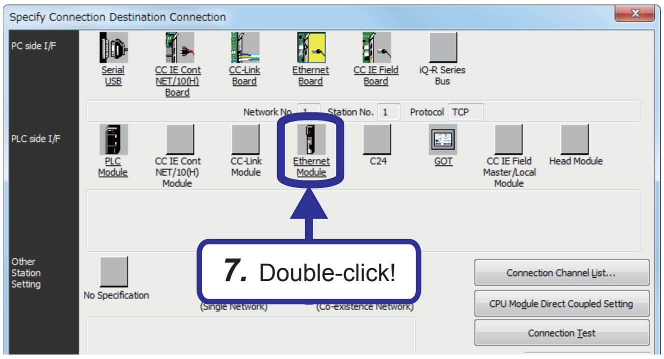
5. Set the parameters as follows on the "PC side I/F Detailed Setting of Ethernet Board" window.

[Parameters to be set]
Network No.: 1
Station No.: A station number of a personal computer (1 to 5)
Protocol: TCP

6. Click the [OK] button.

(To the next page)

(From the previous page)



Specify Connection Destination Connection

PC side I/F: Serial USB, CC IE Cont NET/10(H) Board, CC-Link Board, Ethernet Board, CC IE Field Board, iQ-R Series Bus

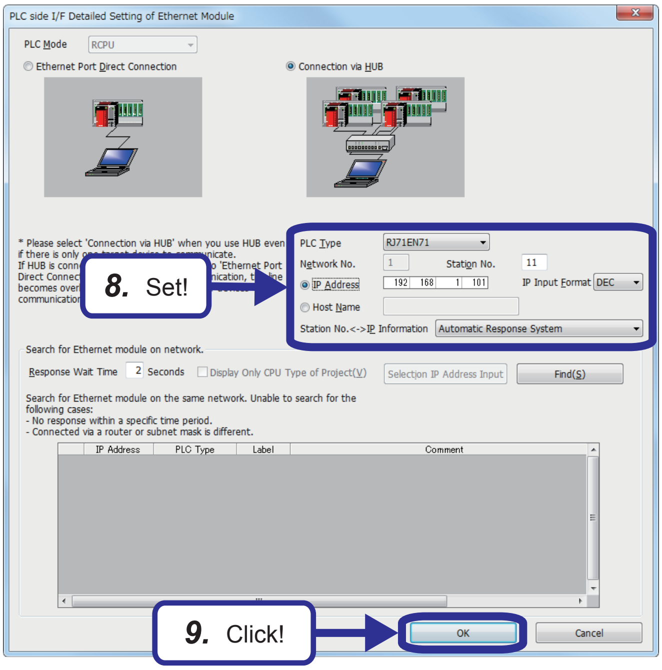
Network No. 1 | Station No. 1 | Protocol TCP

PLC side I/F: PLC Module, CC IE Cont NET/10(H) Module, CC-Link Module, Ethernet Module, C24, GOT, CC IE Field Master/Local Module, Head Module

Other Station Setting: No Specification, (Single Network), (Co-existence Network)

Connection Channel List... | CPU Module Direct Coupled Setting | Connection Test

7. Double-click the [Ethernet Module] button.



PLC side I/F Detailed Setting of Ethernet Module

PLC Mode: RCPU

○ Ethernet Port Direct Connection | ● Connection via HUB

* Please select 'Connection via HUB' when you use HUB even if there is only one target device to communicate. If HUB is connected to 'Ethernet Port Direct Connection' communication, the line becomes overloaded and communication...

PLC Type: RJ71EN71

Network No.: 1 | Station No.: 11

● IP Address: 192 168 1 101 | IP Input Format: DEC

○ Host Name

Station No.<->IP Information: Automatic Response System

Search for Ethernet module on network.

Response Wait Time 2 Seconds | □ Display Only CPU Type of Project(V) | Selection IP Address Input | Find(S)

Search for Ethernet module on the same network. Unable to search for the following cases:
- No response within a specific time period.
- Connected via a router or subnet mask is different.

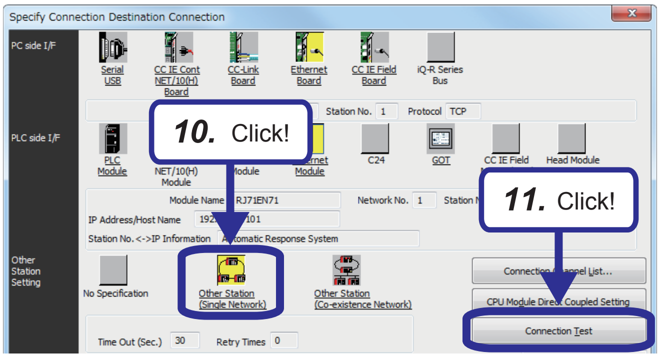
IP Address	PLC Type	Label	Comment

OK | Cancel

8. Set the parameters as follows on the "PLC side I/F Detailed Setting of Ethernet Module" window.

[Parameters to be set]
PLC Type: RJ71EN71
Station No.: A station number of a demonstration machine (11 to 15)
IP Address: An IP address of a demonstration machine (192.168.1.101 to 192.168.1.105)
Station No.<->IP Information: Automatic Response System

9. Click the [OK] button.



Specify Connection Destination Connection

PC side I/F: Serial USB, CC IE Cont NET/10(H) Board, CC-Link Board, Ethernet Board, CC IE Field Board, iQ-R Series Bus

Station No. 1 | Protocol TCP

PLC side I/F: PLC Module, NET/10(H) Module, Module, Ethernet Module, C24, GOT, CC IE Field, Head Module

Module Name: RJ71EN71 | Network No.: 1 | Station No.

IP Address/Host Name: 192.168.1.101

Station No.<->IP Information: Automatic Response System

Other Station Setting: No Specification, Other Station (Single Network), Other Station (Co-existence Network)

Time Out (Sec.) 30 | Retry Times 0

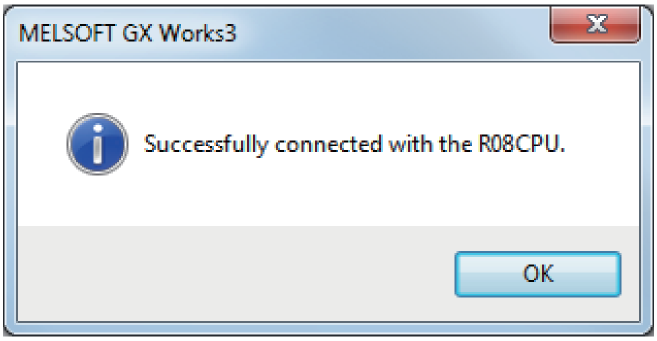
Connection Channel List... | CPU Module Direct Coupled Setting | Connection Test

10. Click the [Other Station (Single Network)] button.

11. Click the [Connection Test] button.

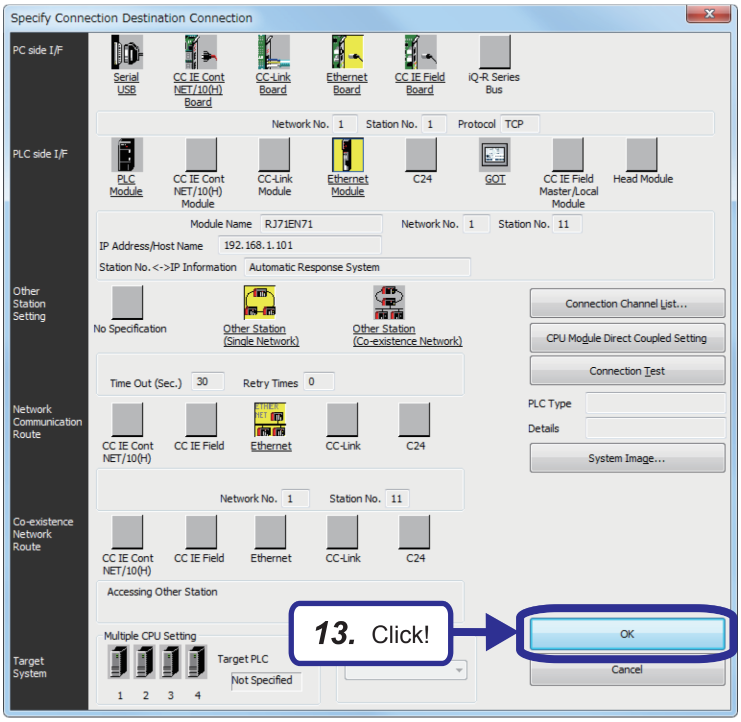
(To the next page)

(From the previous page)


MELSOFT GX Works3

Successfully connected with the R08CPU.

12. Check that the connection between the Ethernet-equipped module and CPU module is successfully established.


Specify Connection Destination Connection

PC side I/F: Ethernet Board — Network No. 1, Station No. 1, Protocol TCP

PLC side I/F: Ethernet Module — Module Name RJ71EN71, Network No. 1, Station No. 11, IP Address/Host Name 192.168.1.101, Station No.<->IP Information Automatic Response System

Other Station Setting: Time Out (Sec.) 30, Retry Times 0

Network Communication Route: Ethernet — Network No. 1, Station No. 11

Target System: Target PLC Not Specified

13. Click the [OK] button.

Appendix 4 Troubleshooting

This section describes troubleshooting of when the Ethernet function is used.

Appendix 4.1 Checking with LED

This section describes troubleshooting using LED.
Error status can be determined by status of the RUN LED and the ERR LED.

RUN LED	ERR LED	Error status[*1]	Description
Off	On, flashing	Major error	An error such as hardware failure or memory failure. The module stops operating.
On	Flashing	Moderate error	An error, such as parameter error, which affect module operation. The module stops operating.
On	On	Minor error	An error such as communication failure. The module continues operating.

[*1] When multiple errors occur, the error status is displayed in the order of major, moderate, and minor.

Point

For the RJ71EN71 or RnENCPU, whether the error occurs in the P1 or P2 can be checked with the P ERR LED.

When the RUN LED or READY LED turns off

When the READY LED of the CPU module turns off after power-on, refer to the troubleshooting of the CPU module. (📖 MELSEC iQ-R CPU Module User's Manual (Application))

When the RUN LED of the RJ71EN71 or RnENCPU turns off, check the following.

Check item	Action
Is the Ethernet-equipped module mounted correctly?	Securely mount the Ethernet-equipped module on the base unit.

If the above action does not solve the problem, perform the module communication test to check for hardware failure. (☞ Page 4 - 22 Module Communication Test)

When the ERROR LED or ERR LED turns on or flashes

When the ERROR LED of the CPU module turns on or flashes, perform the module diagnostics of the CPU module.

🖱 [Diagnostics] ⇨ [Module Diagnostics (CPU Diagnostics)]

When the ERR LED of the RJ71EN71 or RnENCPU turns on or flashes, identify the error cause using the engineering tool. (☞ Page App. - 55 Checking the module status)

If the above action does not solve the problem, perform the module communication test to check for hardware failure. (☞ Page 4 - 22 Module Communication Test)

When the SD/RD LED does not turn on at data sending

When the SD/RD LED does not turn on at data sending, check the following items.

Check item	Action
Is the ERROR LED or ERR LED on or flashing?	Identify the error cause using the engineering tool.
Are the cables properly connected?	Connect the cable properly.
	Perform the following tests to check for the status of the cable connection and line. • PING test • Communication status test
Is the program correct?	Check and correct the send program of the Ethernet-equipped module.

If the above action does not solve the problem, perform the module communication test to check for hardware failure.[*1]

*1 The module communication test cannot be performed for the CPU module (built-in Ethernet port part). Perform the troubleshooting of the CPU module and check for hardware failure. (📖 MELSEC iQ-R CPU Module User's Manual (Startup))

When data cannot be received with the SD/RD LED off

When data cannot be received with the SD/RD LED off, check the following.

Check item	Action
Is the ERROR LED or ERR LED on or flashing?	Identify the error cause using the engineering tool.
Are the cables properly connected?	Connect the cable properly.
	Perform the following tests to check for the status of the cable connection and line. • PING test • Communication status test
Are the parameter settings correct?	Check the module parameters of the Ethernet-equipped module. Correct the value for the following setting if it is wrongly set. • "IP Address" under "Own Node Settings" of "Basic Settings" • "Gateway Parameter Settings" of "Application Settings"
Is the program correct?	Check and correct the send program of the external device.

If the above action does not solve the problem, perform the module communication test to check for hardware failure.[*1]

*1 The module communication test cannot be performed for the CPU module (built-in Ethernet port part). Perform the troubleshooting of the CPU module and check for hardware failure. (📖 MELSEC iQ-R CPU Module User's Manual (Startup))

When the L ER LED turns on

When the L ER LED turns on, check the following.

Check item	Action
Are the Ethernet cables used normally?	• Check if the Ethernet cable which conforms the standard is used. (📖 MELSEC iQ-R Ethernet/CC-Link IE User's Manual (Startup)) • Check if the station-to-station distance is set within range. (📖 MELSEC iQ-R Ethernet/CC-Link IE User's Manual (Startup)) • Check if the Ethernet cables are not disconnected.
Does the cabling condition (bending radius) meet the specifications?	Refer to the manual for the Ethernet cable, and correct the bending radius.
Is the hub used operating normally?	• Check if the hub which conforms the standard is used. (📖 MELSEC iQ-R Ethernet/CC-Link IE User's Manual (Startup)) • Power off and on the hub.
Is there any source of noise near the module or cables?	Change the location of the module or cables.

If the above action does not solve the problem, perform the module communication test to check for hardware failure.

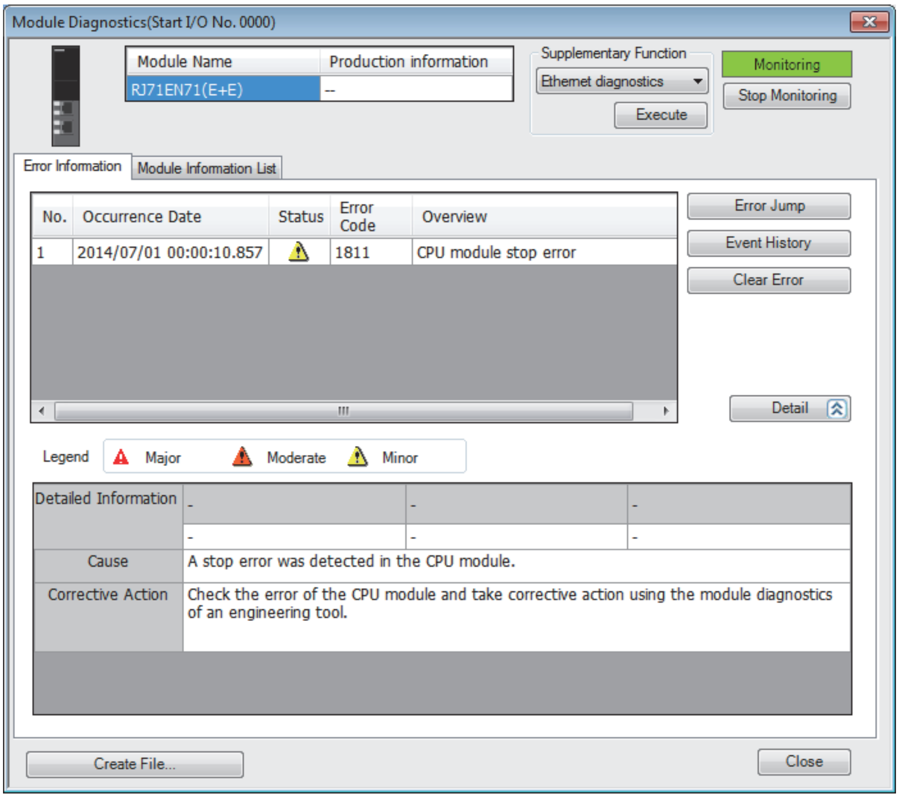
Appendix 4.2 Checking the module status

The following table lists the functions which can be used in the "Module Diagnostics" window of the Ethernet-equipped module.

Function		Application
Error Information		Displays the details of the errors currently occurring. Click the [Event History] button to check the history of errors that have occurred on the network, errors detected for each module, and operations that have been executed.
Module Information List		Displays various status information of the Ethernet-equipped module.
Supplementary Function	Ethernet diagnostics	Enables checking the cause to resolve the problem when an error occurs in the Ethernet system.

Error Information

Check the details of the error currently occurring and action to remove the error.



Module Name	Production information
RJ71EN71(E+E)	--

Supplementary Function: Ethernet diagnostics

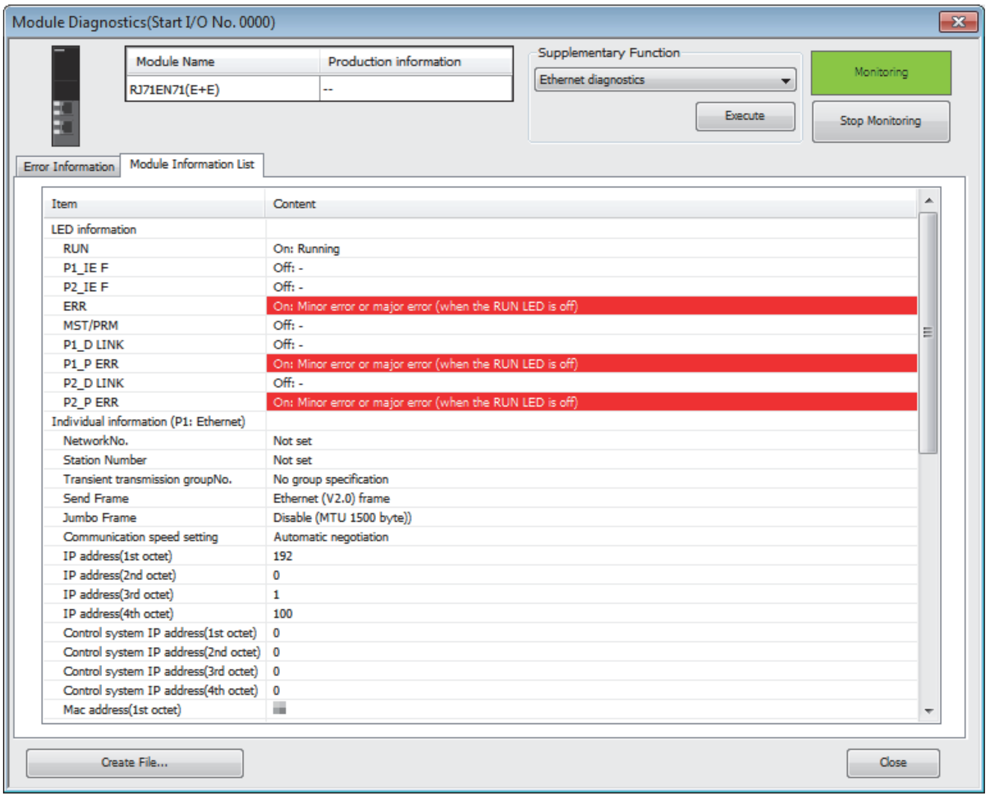
No.	Occurrence Date	Status	Error Code	Overview
1	2014/07/01 00:00:10.857	⚠	1811	CPU module stop error

Detailed Information	-	-	-
	-	-	-
Cause	A stop error was detected in the CPU module.		
Corrective Action	Check the error of the CPU module and take corrective action using the module diagnostics of an engineering tool.		

Item	Description
Status	Major: An error such as hardware failure or memory failure. The module stops operating.
	Moderate: An error, such as parameter error, which affect module operation. The module stops operating.
	Minor: An error such as communication failure. The module continues operating.
Detailed Information	Displays detailed information about each error (maximum of 3 pieces).
Cause	Displays the detailed error causes.
Corrective Action	Displays the actions to eliminate the error causes.

Module Information List

Switch to the [Module Information List] tab to check various status information of the Ethernet-equipped module.



Module Name	Production information
RJ71EN71(E+E)	--

Supplementary Function: Ethernet diagnostics

Item	Content
LED information	
RUN	On: Running
P1_IE F	Off: -
P2_IE F	Off: -
ERR	On: Minor error or major error (when the RUN LED is off)
MST/PRM	Off: -
P1_D LINK	Off: -
P1_P ERR	On: Minor error or major error (when the RUN LED is off)
P2_D LINK	Off: -
P2_P ERR	On: Minor error or major error (when the RUN LED is off)
Individual information (P1: Ethernet)	
NetworkNo.	Not set
Station Number	Not set
Transient transmission groupNo.	No group specification
Send Frame	Ethernet (V2.0) frame
Jumbo Frame	Disable (MTU 1500 byte))
Communication speed setting	Automatic negotiation
IP address(1st octet)	192
IP address(2nd octet)	0
IP address(3rd octet)	1
IP address(4th octet)	100
Control system IP address(1st octet)	0
Control system IP address(2nd octet)	0
Control system IP address(3rd octet)	0
Control system IP address(4th octet)	0
Mac address(1st octet)	▨

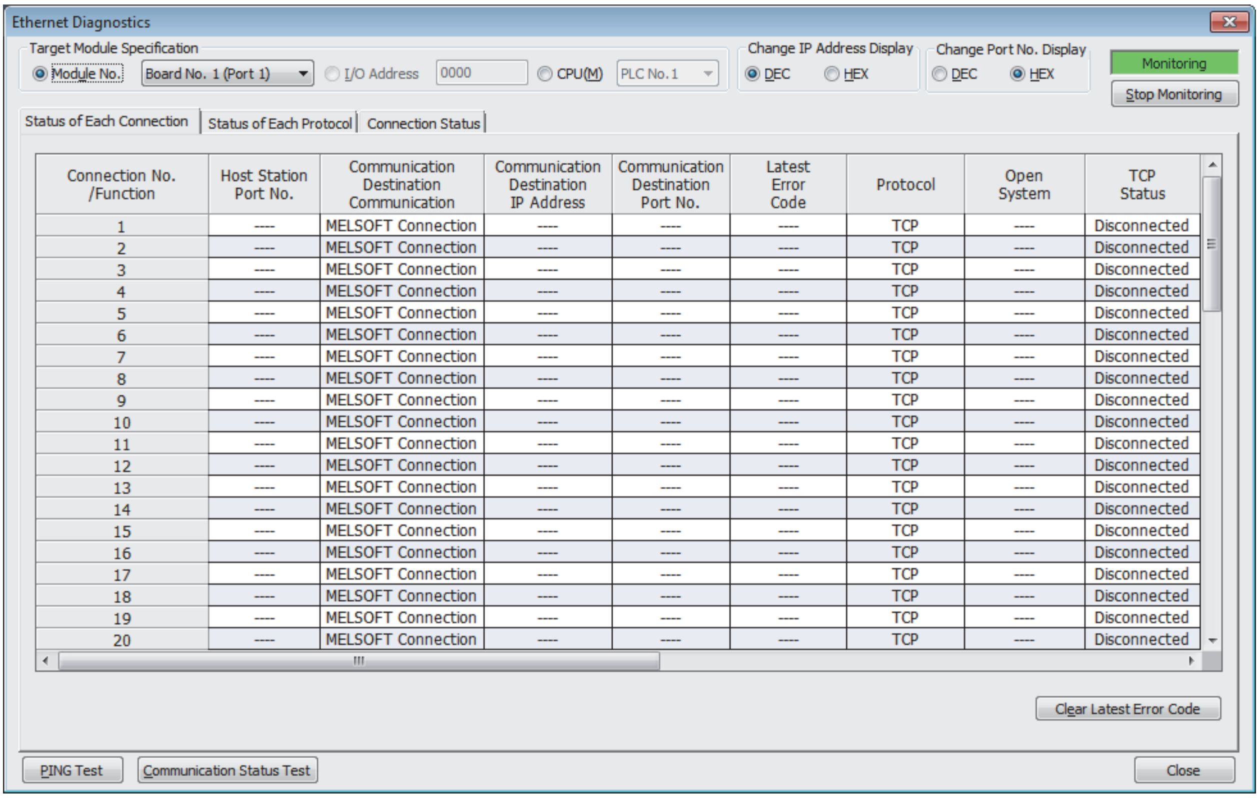
Item		Description
LED information		Displays the LED status of the Ethernet-equipped module.
Individual information (P1: Ethernet)[*1]	Network No.	Displays the network number set for the selected module.
	Station Number	Displays the station number set for the selected module.
	Transient transmission group No.	Displays the transient transmission group number set for the selected module.
	Send Frame	Displays the send frame setting set for the selected module.
	Jumbo Frame	Displays the jumbo frame setting set for the selected module.
	Communication speed setting	Displays the communication mode set for the selected module.
	IP address (1st octet)	Displays the IP address set for the selected module.
	IP address (2nd octet)	
	IP address (3rd octet)	
	IP address (4th octet)	
	Control system IP address (1st octet)	Displays the control system IP address set for the selected module.
	Control system IP address (2nd octet)	
	Control system IP address (3rd octet)	
	Control system IP address (4th octet)	
	MAC address (1st octet)	Displays the MAC address of the selected module.
	MAC address (2nd octet)	
	MAC address (3rd octet)	
	MAC address (4th octet)	
	MAC address (5th octet)	
	MAC address (6th octet)	
Individual information (P2: Ethernet)[*1]		(Same as the P1 connector)

[*1] This item is displayed when the network type is set to "Ethernet".
For when the network type is set to "CC-Link IE Control", refer to the following.
📖 MELSEC iQ-R CC-Link IE Controller Network User's Manual (Application)
For when the network type is set to "CC-Link IE Field", refer to the following.
📖 MELSEC iQ-R CC-Link IE Field Network User's Manual (Application)

Appendix 4.3 Checking the network status

The communication status of the Ethernet-equipped module and external device can be checked with Ethernet diagnostics.

[Diagnostics] ⇨ [Ethernet Diagnostics]



Target Module Specification: ● Module No. Board No. 1 (Port 1) ○ I/O Address 0000 ○ CPU(M) PLC No.1

Change IP Address Display: ● DEC ○ HEX

Change Port No. Display: ○ DEC ● HEX

Connection No. /Function	Host Station Port No.	Communication Destination Communication	Communication Destination IP Address	Communication Destination Port No.	Latest Error Code	Protocol	Open System	TCP Status
1	----	MELSOFT Connection	----	----	----	TCP	----	Disconnected
2	----	MELSOFT Connection	----	----	----	TCP	----	Disconnected
3	----	MELSOFT Connection	----	----	----	TCP	----	Disconnected
4	----	MELSOFT Connection	----	----	----	TCP	----	Disconnected
5	----	MELSOFT Connection	----	----	----	TCP	----	Disconnected
6	----	MELSOFT Connection	----	----	----	TCP	----	Disconnected
7	----	MELSOFT Connection	----	----	----	TCP	----	Disconnected
8	----	MELSOFT Connection	----	----	----	TCP	----	Disconnected
9	----	MELSOFT Connection	----	----	----	TCP	----	Disconnected
10	----	MELSOFT Connection	----	----	----	TCP	----	Disconnected
11	----	MELSOFT Connection	----	----	----	TCP	----	Disconnected
12	----	MELSOFT Connection	----	----	----	TCP	----	Disconnected
13	----	MELSOFT Connection	----	----	----	TCP	----	Disconnected
14	----	MELSOFT Connection	----	----	----	TCP	----	Disconnected
15	----	MELSOFT Connection	----	----	----	TCP	----	Disconnected
16	----	MELSOFT Connection	----	----	----	TCP	----	Disconnected
17	----	MELSOFT Connection	----	----	----	TCP	----	Disconnected
18	----	MELSOFT Connection	----	----	----	TCP	----	Disconnected
19	----	MELSOFT Connection	----	----	----	TCP	----	Disconnected
20	----	MELSOFT Connection	----	----	----	TCP	----	Disconnected

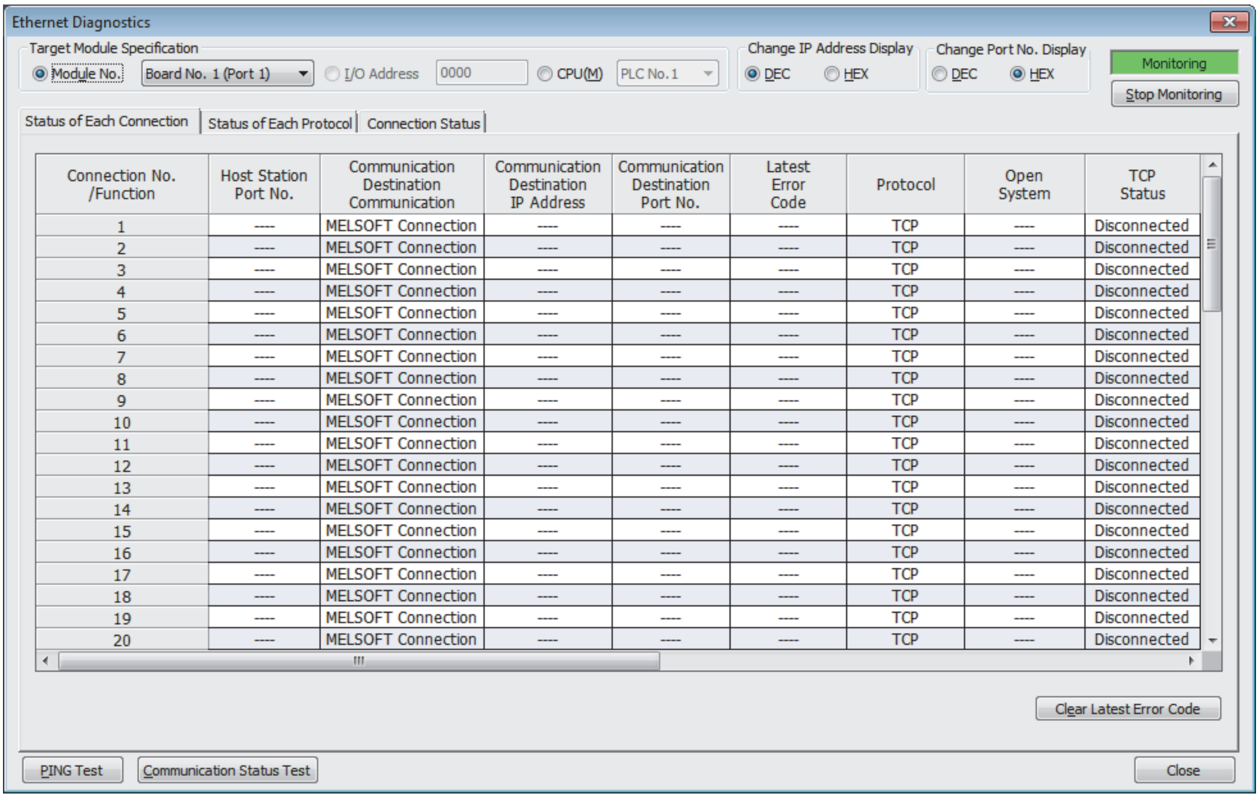
Set the Ethernet-equipped module to be diagnosed in "Target Module Specification".

Point

- The Ethernet diagnostics cannot be started when another station has been specified in "Other Station Setting" on the "Specify Connection Destination Connection" window. Directly connect the engineering tool to the station to be diagnosed, and start the Ethernet diagnostics.
- In a redundant system configuration of the remote head module, the Ethernet diagnostics cannot be started when the engineering tool is connected to the remote head module of the standby system. Connect it to the remote head module of the control system, and start the diagnostics.

Status of Each Connection

The status of each connection of the Ethernet-equipped module selected.



Target Module Specification: ● Module No. Board No. 1 (Port 1) ○ I/O Address 0000 ○ CPU(M) PLC No.1

Change IP Address Display: ● DEC ○ HEX

Change Port No. Display: ○ DEC ● HEX

Connection No. /Function	Host Station Port No.	Communication Destination Communication	Communication Destination IP Address	Communication Destination Port No.	Latest Error Code	Protocol	Open System	TCP Status
1	----	MELSOFT Connection	----	----	----	TCP	----	Disconnected
2	----	MELSOFT Connection	----	----	----	TCP	----	Disconnected
3	----	MELSOFT Connection	----	----	----	TCP	----	Disconnected
4	----	MELSOFT Connection	----	----	----	TCP	----	Disconnected
5	----	MELSOFT Connection	----	----	----	TCP	----	Disconnected
6	----	MELSOFT Connection	----	----	----	TCP	----	Disconnected
7	----	MELSOFT Connection	----	----	----	TCP	----	Disconnected
8	----	MELSOFT Connection	----	----	----	TCP	----	Disconnected
9	----	MELSOFT Connection	----	----	----	TCP	----	Disconnected
10	----	MELSOFT Connection	----	----	----	TCP	----	Disconnected
11	----	MELSOFT Connection	----	----	----	TCP	----	Disconnected
12	----	MELSOFT Connection	----	----	----	TCP	----	Disconnected
13	----	MELSOFT Connection	----	----	----	TCP	----	Disconnected
14	----	MELSOFT Connection	----	----	----	TCP	----	Disconnected
15	----	MELSOFT Connection	----	----	----	TCP	----	Disconnected
16	----	MELSOFT Connection	----	----	----	TCP	----	Disconnected
17	----	MELSOFT Connection	----	----	----	TCP	----	Disconnected
18	----	MELSOFT Connection	----	----	----	TCP	----	Disconnected
19	----	MELSOFT Connection	----	----	----	TCP	----	Disconnected
20	----	MELSOFT Connection	----	----	----	TCP	----	Disconnected

The following table lists the displayed items in "Status of Each Connection" tab.

Item	Description
Connection No./Function	Displays the connection number and functions (FTP server, FTP client[*3], MELSOFT direct connection).
Host Station Port No.	Displays the own station port number used.
Communication Destination Communication Method[*2]	Displays the communication method.
Communication Destination IP Address	Displays the IP address of the sensor/device to be connected, which is set in the parameter settings.
Communication Destination Port No.	Displays the port number of the sensor/device to be connected, which is set in the parameter settings.
Latest Error Code	Displays the error code that indicates the definition of latest error occurred.
Protocol	Displays the protocol (TCP/IP or UDP/IP).
Open System	Displays the open method (Active, Unpassive, or Fullpassive) when the protocol of the connection is TCP/IP.
TCP Status	Displays the status (open status) of connection with the sensor/device when the protocol of the connection status is TCP/IP.
Pairing Open[*1]	Displays the setting status of pairing open when the communication method of the connection is the fixed buffer.
Existence Confirmation[*1]	Displays the alive check method set in the parameter settings.
Remote Password Status	Displays the remote password setting status.
Consecutive failed unlock attempts	Displays the total number of continuous failure of remote password unlock.
Force Deactivation Status	Displays the status of forced invalidation specification.

[*1] This item is displayed only for the RJ71EN71 and the RnENCPU (network part).

[*2] This item is displayed only when the network type of the RJ71EN71 is set to "Q Compatible Ethernet" and "MELSOFT Connection Module" is set as an external device in "External Device Configuration" of "Basic Settings".

[*3] This item is displayed only for the CPU module (built-in Ethernet port part).

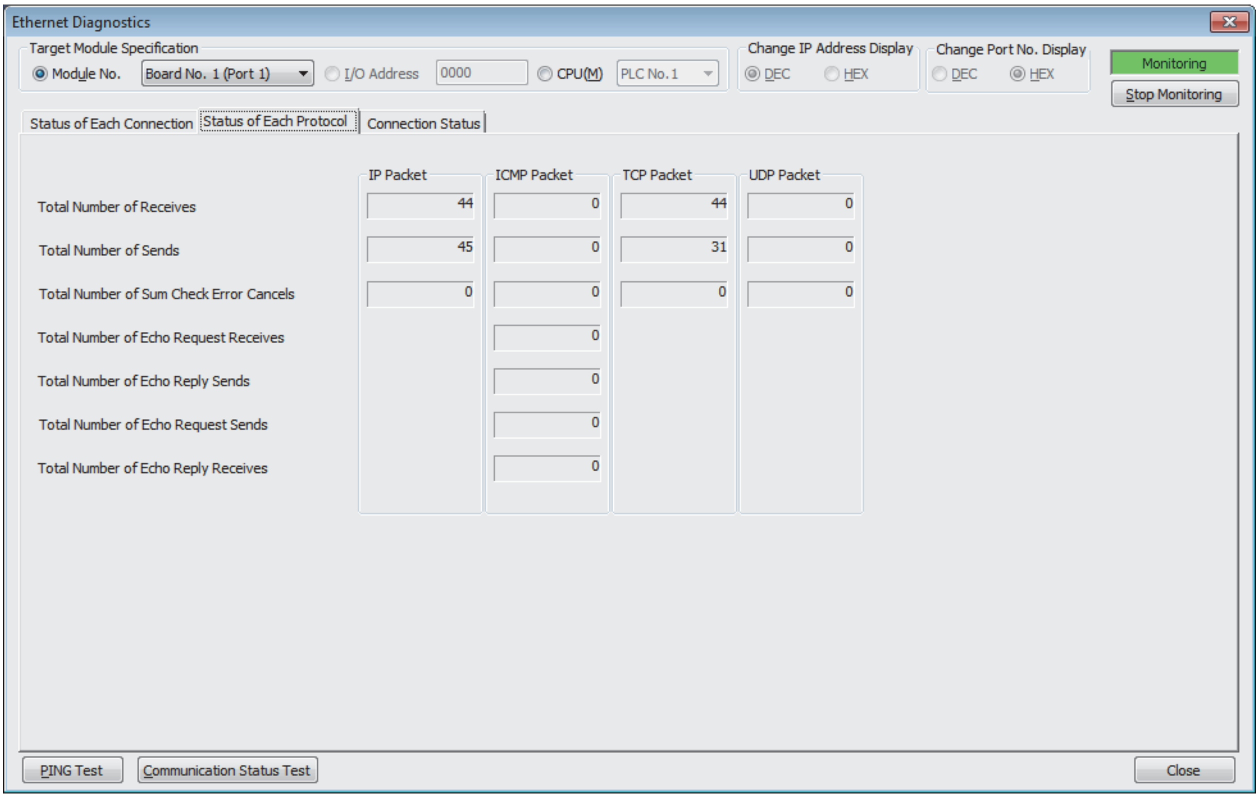
Click the [Clear Latest Error Code] button to clear all the errors displayed in "Latest Error Code" of each connection.

Point

Information about connections No.17 to 64, FTP server, and MELSOFT direct connection are not displayed when "Q Compatible Ethernet" is set in the network type of the RJ71EN71.

Status of Each Protocol

The total number of packets sent/received by each protocol of the selected Ethernet-equipped module can be checked.



Ethernet Diagnostics

Target Module Specification: ● Module No. Board No. 1 (Port 1) ○ I/O Address 0000 ○ CPU(M) PLC No.1

Change IP Address Display: ● DEC ○ HEX

Change Port No. Display: ○ DEC ● HEX

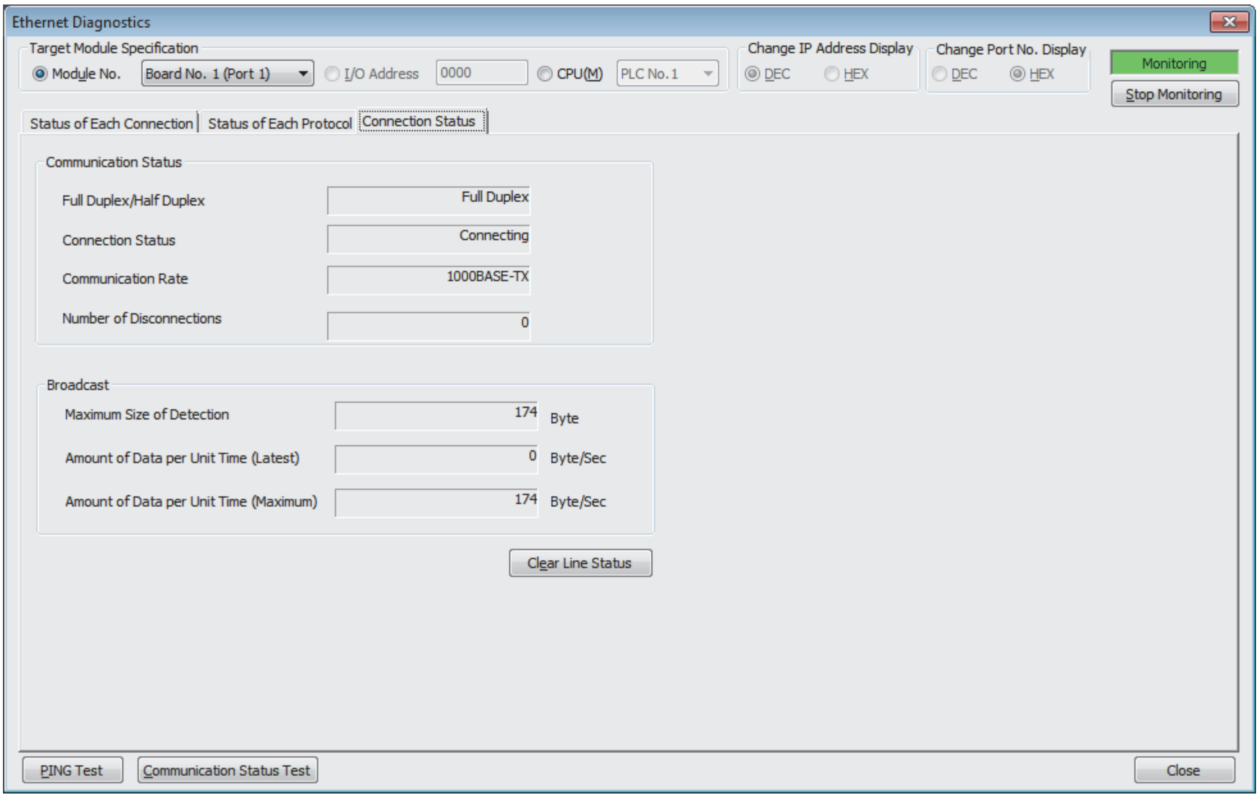
	IP Packet	ICMP Packet	TCP Packet	UDP Packet
Total Number of Receives	44	0	44	0
Total Number of Sends	45	0	31	0
Total Number of Sum Check Error Cancels	0	0	0	0
Total Number of Echo Request Receives		0		
Total Number of Echo Reply Sends		0		
Total Number of Echo Request Sends		0		
Total Number of Echo Reply Receives		0		

The following table lists the displayed items in "Status of Each Protocol" tab.

Item	Description	Display range
Total Number of Receives	Displays the total number of received packets.	0 to 4294967295
Total Number of Sends	Displays the total number of sent packets.	0 to 4294967295
Total Number of Sum Check Error Cancels	Displays the number of times the received packet was discarded due to checksum error.	0 to 4294967295
Total Number of Echo Request Receives	Displays the total number of received ICMP echo request packets.	0 to 4294967295
Total Number of Echo Reply Sends	Displays the total number of sent ICMP echo replay packets.	0 to 4294967295
Total Number of Echo Request Sends	Displays the total number of sent ICMP echo request packets.	0 to 4294967295
Total Number of Echo Reply Receives	Displays the total number of received ICMP echo reply packets.	0 to 4294967295

Connection Status

The communication status of the Ethernet-equipped module.



Target Module Specification
● Module No. Board No. 1 (Port 1) ○ I/O Address 0000 ○ CPU(M) PLC No.1

Change IP Address Display ● DEC ○ HEX

Change Port No. Display ○ DEC ● HEX

Communication Status

Full Duplex/Half Duplex: Full Duplex

Connection Status: Connecting

Communication Rate: 1000BASE-TX

Number of Disconnections: 0

Broadcast

Maximum Size of Detection: 174 Byte

Amount of Data per Unit Time (Latest): 0 Byte/Sec

Amount of Data per Unit Time (Maximum): 174 Byte/Sec

The following table lists the displayed items in "Connection Status" tab.

Item		Description	Display range
Communication Status	Full Duplex/Half Duplex	Displays whether the line is full-duplex or half-duplex.	—
	Connection Status	Displays the cable connection status.	—
	Communication Rate	Displays the communication speed.	—
	Number of Disconnections	Displays the number of times the line entered a state where communication cannot be performed.	0 to 65535
Broadcast	Maximum Size of Detection	Displays the maximum size of discarded broadcast messages.	0 to 65535
	Amount of Data per Unit Time (Latest)	Displays the size (latest value) per second of discarded broadcast messages.	0 to 4294967295
	Amount of Data per Unit Time (Maximum)	Displays the size (maximum value) per second of discarded broadcast messages.	0 to 4294967295

Click the [Clear Line Status] button to clear all the data in "Broadcast".

Point

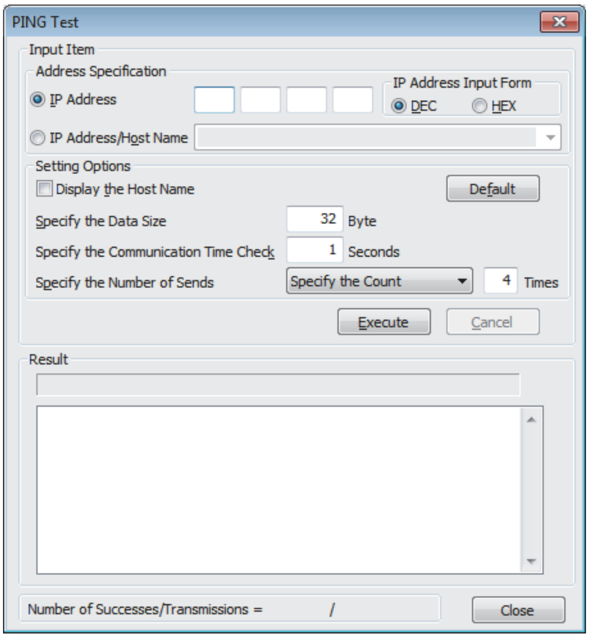
Information in "Broadcast" is not displayed when "Q Compatible Ethernet" is selected for the network type of the RJ71EN71.

PING Test

The PING test checks existence of an Ethernet device on the same Ethernet network.

This test is performed on the network of stations connected to the engineering tool by sending packets for check. If a response returns, the communications can be performed.

"Ethernet Diagnostics" window ⇨ [PING Test] button



- ● IP Address
- ○ IP Address/Host Name
- IP Address Input Form: ● DEC ○ HEX
- □ Display the Host Name
- Specify the Data Size: 32 Byte
- Specify the Communication Time Check: 1 Seconds
- Specify the Number of Sends: Specify the Count, 4 Times
- Number of Successes/Transmissions = /

■Procedure

Set the required items in "Input Item" and click the [Execute] button to execute the PING test. The test results are displayed in the "Result" box.

Point

When executing the PING test from the RJ71EN71 and the RnENCPU (network part) by connecting the engineering tool and CPU module with an USB cable, network number and station number setting is required to specify the PING executing station.

■Action for abnormal end

If the test fails, check the following and perform the test again.

- Whether the Ethernet-equipped module is properly mounted on the base unit
- Connection to the Ethernet network
- Parameter settings written to the CPU module
- Operating status of the CPU module (whether or not an error has occurred)
- IP addresses set in the engineering tool and the PING test target station
- Whether the external device has been reset after the Ethernet-equipped module was replaced

Communication Status Test

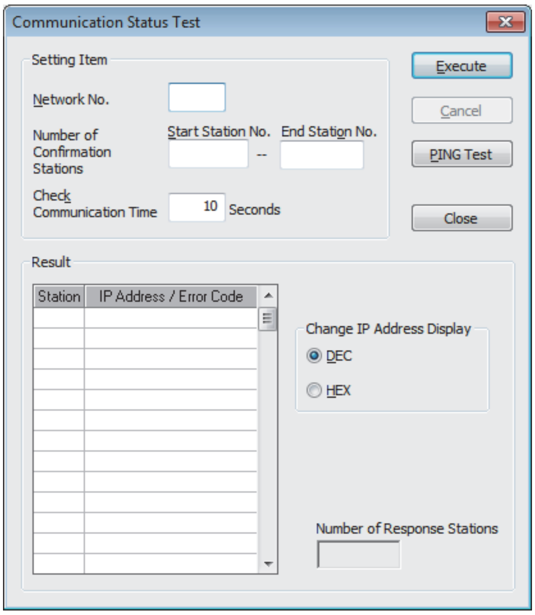
The communication status test checks whether the communications between the own station and the external device on the same Ethernet are operated normally.

The following external devices are the target of communication status test.

- RJ71EN71 (when the Ethernet function is used)
- RnENCPU (network part) (when the Ethernet function is used)
- MELSEC-Q series Ethernet interface module (function version B or later)
- MELSEC-L series Ethernet interface module

This test is performed on the network of stations connected to the engineering tool, by sending the test messages sequentially to specified network and station number range. If the module that received the message returns a response, communications are normal.

"Ethernet Diagnostics" window ⇨ [Communication Status Test] button



Restriction

The communication status test cannot be performed when the Ethernet function of the CPU module (CPU part for the RnENCPU) is used.

■Procedure

Set the required items in "Setting Item" and click the [Execute] button to execute the communication status test. The test results are displayed in the "Result" box.

■Action for abnormal end

If the communication status test fails, "No Response" or an error code is displayed in the "IP Address / Error Code" field in "Result".

Displayed test result	Status of the external device	Cause	Action
No Response	No error	The initial process for the Ethernet-equipped module has not completed successfully.	Correct the module parameters.
		There is an error in the line connection to the Ethernet-equipped module. (Cable disconnection, line disconnection, or other errors)	• Check the cable. • Check the hub and gateway.
		The IP address of the Ethernet-equipped module is incorrect. (The class or subnet address differs from that of the Ethernet-equipped module.)	Correct the module parameters.
		The same IP address has been set to multiple Ethernet-equipped modules.	
		The same network number or station number has been set to multiple Ethernet-equipped modules.	
	No error/Error exists	The Ethernet line is heavily loaded.	Perform the test again when the Ethernet line is not heavily loaded.
	Error exists	The routing settings are not configured.	Correct the setting value for "Routing Setting" of "CPU Parameter".
Error code	No error	The "MELSOFT Transmission Port (UDP/IP)" for the Ethernet-equipped module is locked with the remote password.	Disable the remote password setting and write the parameters to the CPU module.
		The target module does not support the communication status test.	Check the module name and function version.
	No error/Error exists	The Ethernet line is heavily loaded.	Perform the test again when the Ethernet line is not heavily loaded.

■Precautions

- The communication status test cannot be executed when "MELSOFT Transmission Port (UDP/IP)" for the target Ethernet-equipped module is locked with the remote password.
- When executing the communication status test, set "Gateway Other Than Default Gateway" under "Gateway Parameter Settings" in "Application Settings" to "Not Use".

Appendix 5 Dedicated Instructions

Dedicated instructions facilitate programming for using functions of intelligent function modules.

This section describes the Ethernet-equipped module dedicated instructions available on the CPU module and the instructions used in sequence programs on this textbook.

For details, refer to the following.

📖 MELSEC iQ-R Programming Manual (Instructions, Standard Functions/Function Blocks)

Appendix 5.1 List of dedicated instructions

This section describes the dedicated instructions that can be used in Ethernet.

Point

For details on dedicated instructions, refer to the following.

📖 MELSEC iQ-R Programming Manual (Instructions, Standard Functions/Function Blocks)

Open/close instructions

The following table lists the dedicated instructions used for open or close processing.

○: Available, ×: Not available

Instruction	Description	Availability	
		RJ71EN71, RnENCPU (network part)	CPU module (built-in Ethernet port part)
GP.CONOPEN	Establishes a connection.	○	×
SP.SOCOPEN		×	○
OPEN		○	×
GP.CONCLOSE	Closes the connection.	○	×
SP.SOCCLOSE		×	○
CLOSE		○	×

Instructions for predefined protocol communications

The following table lists the dedicated instructions used for communications using the predefined protocol.

○: Available, ×: Not available

Instruction	Description	Availability	
		RJ71EN71, RnENCPU (network part)	CPU module (built-in Ethernet port part)
GP.ECPRTCL	Executes the protocol registered with the engineering tool's predefined protocol support function.	○	×
SP.ECPRTCL		×	○

Socket communications instructions

The following table lists the dedicated instructions used for socket communications.

○: Available, ×: Not available

Instruction	Description	Availability	
		RJ71EN71, RnENCPU (network part)	CPU module (built-in Ethernet port part)
GP.SOCRCV	Reads the receive data from the external device.	○	×
SP.SOCRCV		×	○
G.SOCRCVS		○	○
S.SOCRCVS		×	○
GP.SOCSND	Sends data to the external device.	○	○
SP.SOCSND		×	○
SP.SOCCINF	Reads connection information.	×	○
SP.SOCCSET	Changes the communication target.	×	○
SP.SOCRMODE	Changes the connection receive mode.	×	○
S(P).SOCRDATA	Reads the specified size of data from the socket communication receive data area.	×	○

Instruction for SLMP communications

The following table lists the dedicated instruction used for communications using the SLMP.

○: Available, ×: Not available

Instruction	Description	Availability	
		RJ71EN71, RnENCPU (network part)	CPU module (built-in Ethernet port part)
SP.SLMPSND	Sends SLMP message to the SLMP-compatible device.	×	○

File transfer function instructions

The following table lists the dedicated instructions used for the file transfer function (FTP client).

○: Available, ×: Not available

Instruction	Description	Availability	
		RJ71EN71, RnENCPU (network part)	CPU module (built-in Ethernet port part)
SP.FTPPUT	This instruction sends files in the CPU module (FTP client) to the folder path of the specified FTP server.	×	○
SP.FTPGET	This instruction retrieves files on the FTP server to the folder path of the specified CPU module (FTP client).	×	○

Instructions for communications using the fixed buffer

The following table lists the dedicated instructions used in the RJ71EN71 and the RnENCPU (network part) for communications using the fixed buffer.

Instruction	Description
BUFRCV	Reads the receive data from the external device.
BUFRCVS	Reads the receive data with an interrupt program.
BUFSND	Sends data to the external device.

Link dedicated instructions

The following table lists the dedicated instructions used in the RJ71EN71 and the RnENCPU (network part) for transient transmission with programmable controllers on other stations. Each link dedicated instruction allows access to a station on a network other than Ethernet.

Instruction	Description
SEND	Sends data to another station.
RECV	Reads the receive data from another station (for main program).
RECVS	Reads the receive data from another station (for interrupt program).
READ	Reads data from the word device of another station.
SREAD	Reads data from the word device of another station (with completion device).
WRITE	Writes data to the word device of another station.
SWRITE	Writes data to the word device of another station (with completion device).
REQ	Requests the remote RUN/STOP to the CPU module on another station.
	Reads/writes clock data from/to another station.
ZNRD	Reads data from the word device in another station (ACPU).
ZNWR	Writes data to the word device in another station (ACPU).

Other dedicated instructions

Other instructions used by the RJ71EN71 and the RnENCPU (network part)

Instruction	Description
ERRCLEAR	Turns off the LED and clears error information.[*1]
ERRRD	Reads error information.
UINI	Performs re-initial processing.

[*1] Availability for turning off the LED depends on the firmware version of the RJ71EN71.

Appendix 5.1.1 Precautions for dedicated instructions

This section describes precautions when using the dedicated instructions.

Precautions for dedicated instructions (common)

■When changing data specified by dedicated instructions

Do not change any data (such as control data) until execution of the dedicated instruction completes.

■When the dedicated instruction does not complete

Check whether "Module Operation Mode" in "Application Settings" of the RJ71EN71 and the RnENCPU (network part) is "Online".

A dedicated instruction cannot be executed when the mode is "Offline" or "Module Communication Test".

Precautions for link dedicated instructions

The following describes precautions when executing multiple link dedicated instructions simultaneously.

■Channel of the link dedicated instructions

When executing multiple link dedicated instructions simultaneously, check that the instructions do not use the same channel number. Link dedicated instructions specifying the same channel number cannot be executed simultaneously. To use the same channel for multiple link dedicated instructions, configure an interlock so that an instruction is executed after completion of another.

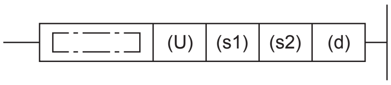
Appendix 5.2 Opening a connection

GP.OPEN, ZP.OPEN

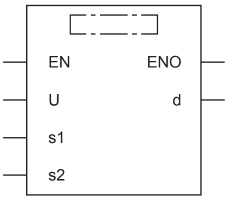
RnCPU RnENCPU RnPCPU (Process) RnPCPU (Redundant) RnSFCPU (Standard) ☒ RnSFCPU (Safety)

These instructions establish (open) a connection with an external device for data communications.

Ladder	ST

	ENO: =GP_OPEN(EN,U,s1,s2,d);
	ENO: =ZP_OPEN(EN,U,s1,s2,d);

FBD/LD



■Execution condition

Instruction	Execution condition
GP.OPEN ZP.OPEN	↑

Setting data

■Description, range, data type

Operand		Description	Range	Data type	Data type (label)
(U)	GP.OPEN	Start I/O number (first three digits in four-digit hexadecimal representation) of own station or own node	00H to FEH	16-bit unsigned binary	ANY16
	ZP.OPEN	Start I/O number (first three digits in four-digit hexadecimal representation) of own station or own node	00H to FEH	String	ANY16_OR_STRING_SINGLE
(s1)		Connection No.	1 to 16	16-bit unsigned binary	ANY16
(s2)		Own station start device where control data is stored	Refer to the control data.	Device name	ANY16*1
(d)		Device of the own station, which turns on for one scan upon completion of the instruction. When the instruction completes with an error, (d)+1 also turns on.	—	Bit	ANYBIT_ARRAY (Number of elements: 2)
EN		Execution condition	—	Bit	BOOL
ENO		Execution result	—	Bit	BOOL

*1 When specifying setting data by using a label, define an array to secure enough operation area and specify an element of the array label.

■Applicable devices

Operand		Bit		Word			Double word		Indirect specification	Constant			Others (U)
		X, Y, M, L, SM, F, B, SB, FX, FY	J□\□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□\G□, J□\□, U3E□\(H)G□	Z	LT, LST, LC	LZ		K, H	E	$	
(U)	GP.OPEN	—	—	○	—	—	—	—	○	○	—	—	○
	ZP.OPEN	—	—	○	—	—	—	—	○	—	—	○	○
(s1)		○*1	—	○*2	—	—	—	—	○	○	—	—	—
(s2)		—	—	○*2	—	—	—	—	○	—	—	—	—
(d)		○*1	—	○*3	—	—	—	—	—	—	—	—	—

*1 FX and FY cannot be used.

*2 FD cannot be used.

*3 T, ST, C, and FD cannot be used.

■Control data

Operand: (s2)				
Device	**Item**	**Description**	**Setting range**	**Set by**
+0	Execution type/End type	Specify whether to use the parameter value set by the engineering tool or the value set in (s2)+2 to (s2)+6 of control data for opening a connection. • 0000H: The open processing is performed according to the setting in "External Device Configuration" of the engineering tool. • 8000H: The open processing is performed according to the setting in (s2)+2 to (s2)+6 of control data.	0000H 8000H	User
+1	Completion status	The completion status is stored upon completion of the instruction. • 0: Completed successfully • Other than 0: Completed with an error (error code)	—	System
+2	Application setting area	Specify the application of a connection. b15 b14 b13 ··· b11 b10 b9 b8 b7 b6 ··· b2 b1 b0 (6) | 0 | (5) | (4) | (3) | 0 | (2) | (1) (1) Application of fixed buffer (b0) • 0: For sending • 1: For receiving (2) Target station alive check (b1) • 0: Disable the alive check • 1: Alive check[*1] (3) Pairing open (b7) • 0: Disable pairing open • 1: Enable pairing open[*2] (4) Protocol (b8) • 0: TCP/IP • 1: UDP/IP (5) Fixed buffer communications procedure (b9 and b10) • 00: Procedure exist • 01: No procedure • 10: Predefined protocol (6) Opening method (b14 and b15) • 00: Active open or UDP/IP • 10: Unpassive open • 11: Fullpassive open	Left	User
+3	Own station port number	Specify the port number of the own station. (Port numbers 5000 to 5009 is reserved for the system and cannot be used.)	1024 to 4999, 5010 to 65534 (0400H to 1387H, 1392H to FFFEH)	User
+4 to +5	IP address of external device[*3]	Specify the IP address (IPv4) of an external device. • The IP address is stored in (s2)+4 and (s2)+5. • To enable broadcast, specify FFFFFFFFH.	00000001H to FFFFFFFFH	User
+6	Destination port number[*3]	Specify the destination port number. When receiving data from all port numbers, specify FFFFH.	1 to 65534, 65535 (0001H to FFFEH, FFFFH)	User
+7 to +9	System area	—	—	—

[*1] When the TCP/IP protocol is used, the alive check method is fixed to KeepAlive. (When UDP/IP is used, it is fixed to Ping.)

[*2] This item can be set when the connection number set in (s1) is one from 1 to 7 and 9 to 15.

[*3] Settings are ignored if the open method (bits 14 and 15) specified by (s2)+2 is "10: Unpassive open".

Processing details

- These instructions open the connection specified by (s1) of the module specified by (U).
- The selection of the setting value used for open processing is specified by (s2)+0. (Specify whether to use the parameter value set by the engineering tool or the value set in (s2)+2 to (s2)+16 of control data.)
- The execution status and the completion status of the OPEN instruction can be checked with the completion device (d) and the completion status indication device (d)+1.

- Completion device (d)

This device turns on during END processing of the scan where the OPEN instruction completes, and turns off during the next END processing.

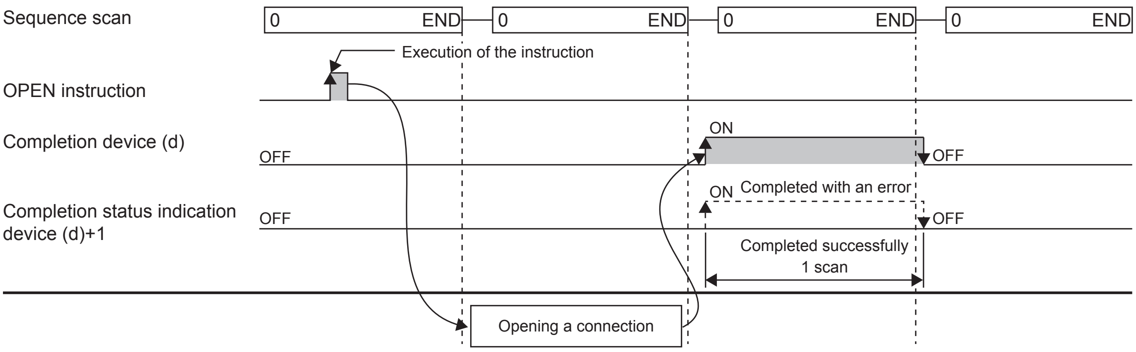
- Completion status indication device (d)+1

This device turns on or off depending on the completion status of the OPEN instruction.

When completed successfully: The device remains off.

When completed with an error: The device turns on during END processing of the scan where the OPEN instruction completes, and turns off during the next END processing.

- The following figure shows the operation at completion of the OPEN instruction.



- The OPEN instruction is executed on the rising edge (OFF to ON) of the open command.

Precautions

For the same connection, do not perform open and close processing using the OPEN and CLOSE instructions simultaneously with open and close processing using other means. Simultaneous use results in a malfunction.

Operation error

Error code ((s2)+1)	Description
C000H to CFFFH	📖 MELSEC iQ-R Ethernet User's Manual (Application)

Restriction

- The communication method of the target connection is fixed buffer communications (procedure exist), fixed buffer communications (no procedure), or predefined protocol. Make settings in (b9, b10) (fixed buffer communications procedure) of (s2)+2 in control data.
- When the protocol is set to TCP/IP, the alive check method is fixed to KeepAlive.
- The connection numbers that can be specified range from 1 to 16. Connection number 17 and after cannot be specified.
- If no parameter data is set in "External Device Configuration" of the engineering tool, the communication data code becomes "Binary". If one or more parameter data are set, the instruction follows the value set in "Communication Data Code".
- If no parameter data is set in "External Device Configuration" of the engineering tool, the opening method becomes "Do Not Open by Program". If one or more parameter data are set, the instruction follows the value set in "Opening Method".

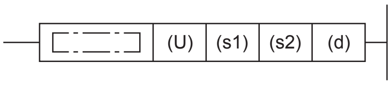
Appendix 5.3 Closing a connection

GP.CLOSE, ZP.CLOSE

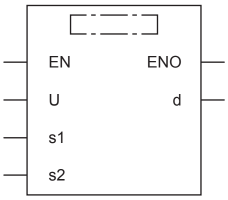
RnCPU RnENCPU RnPCPU (Process) RnPCPU (Redundant) RnSFCPU (Standard) ~~RnSFCPU (Safety)~~

These instructions disconnect (close) the connection from the external device during data communications.

Ladder	ST

	ENO: =GP_CLOSE(EN,U,s1,s2,d);
	ENO: =ZP_CLOSE(EN,U,s1,s2,d);

FBD/LD



■Execution condition

Instruction	Execution condition
GP.CLOSE ZP.CLOSE	↑

Setting data

■Description, range, data type

Operand		Description	Range	Data type	Data type (label)
(U)	GP.CLOSE	Start I/O number (first three digits in four-digit hexadecimal representation) of own station or own node	00H to FEH	16-bit unsigned binary	ANY16
	ZP.CLOSE	Start I/O number (first three digits in four-digit hexadecimal representation) of own station or own node	00H to FEH	String	ANY16_OR_STRING_SINGLE
(s1)		Connection No.	1 to 16	16-bit unsigned binary	ANY16
(s2)		Own station start device where control data is stored	Refer to the control data.	Device name	ANY16*1
(d)		Device of the own station, which turns on for one scan upon completion of the instruction. When the instruction completes with an error, (d)+1 also turns on.	—	Bit	ANYBIT_ARRAY (Number of elements: 2)
EN		Execution condition	—	Bit	BOOL
ENO		Execution result	—	Bit	BOOL

*1 When specifying setting data by using a label, define an array to secure enough operation area and specify an element of the array label.

■Applicable devices

Operand		Bit		Word			Double word		Indirect specification	Constant			Others (U)
		X, Y, M, L, SM, F, B, SB, FX, FY	J□\□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□\G□, J□\□, U3E□\(H)G□	Z	LT, LST, LC	LZ		K, H	E	$	
(U)	GP.CLOSE	—	—	○	—	—	—	—	○	○	—	—	○
	ZP.CLOSE	—	—	○	—	—	—	—	○	—	—	○	○
(s1)		○[1]	—	○[2]	—	—	—	—	○	○	—	—	—
(s2)		—	—	○[2]	—	—	—	—	○	—	—	—	—
(d1)		○[1]	—	○[3]	—	—	—	—	—	—	—	—	—

[1] FX and FY cannot be used.

[2] FD cannot be used.

[3] T, ST, C, and FD cannot be used.

■Control data

Operand: (s2)				
Device	**Item**	**Description**	**Setting range**	**Set by**
+0	System area	—	—	—
+1	Completion status	The completion status is stored upon completion of the instruction. • 0: Completed successfully • Other than 0: Completed with an error (error code)	—	System

Processing details

- These instructions close the connection specified by (s1) of the module specified by (U).
- The execution status and the completion status of the CLOSE instruction can be checked with the completion device (d) and the completion status indication device (d)+1.

• Completion device (d)

This device turns on during END processing of the scan where the CLOSE instruction completes, and turns off during the next END processing.

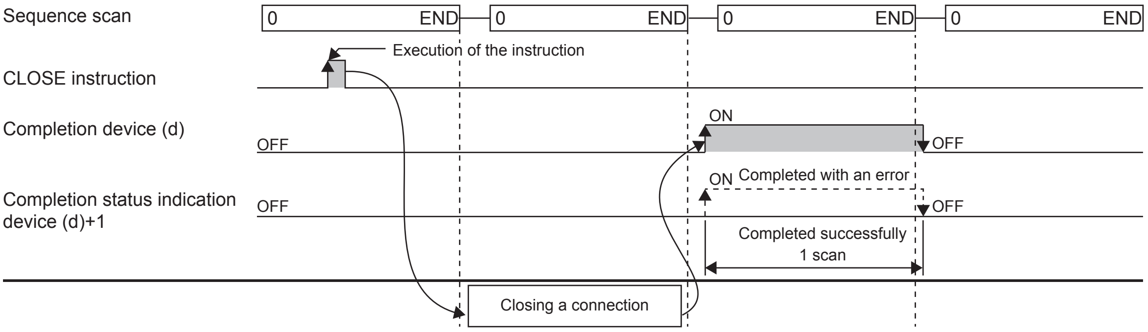
• Completion status indication device (d)+1

This device turns on or off depending on the completion status of the CLOSE instruction.

When completed successfully: The device remains off.

When completed with an error: The device turns on during END processing of the scan where the CLOSE instruction completes, and turns off during the next END processing.

- The following figure shows the operation at completion of the CLOSE instruction.



- The CLOSE instruction is executed on the rising edge (OFF to ON) of the close command.

Precautions

- For the same connection, do not perform open and close processing using the OPEN and CLOSE instructions simultaneously with open and close processing using other means. Simultaneous use results in a malfunction.
- If a connection for which the OPEN instruction is in execution is specified in TCP Unpassive/Fullpassive open mode, an error (C1B2H: OPEN/CLOSE instruction is in execution for the specified connection) occurs.

Operation error

Error code ((s2)+1)	Description
C000H to CFFFH	🕮 MELSEC iQ-R Ethernet User's Manual (Application)

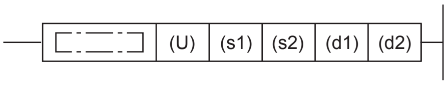
Appendix 5.4 Reading receive data

GP.BUFRCV, ZP.BUFRCV

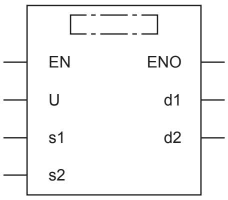
RnCPU RnENCPU RnPCPU (Process) RnPCPU (Redundant) RnSFCPU (Standard) ~~RnSFCPU (Safety)~~

These instructions read receive data from the external device through fixed buffer communications.

Ladder	ST

	ENO: =GP_BUFRCV(EN,U,s1,s2,d1,d2);
	ENO: =ZP_BUFRCV(EN,U,s1,s2,d1,d2);

FBD/LD



■Execution condition

Instruction	Execution condition
GP.BUFRCV ZP.BUFRCV	↑

Setting data

■Description, range, data type

Operand		Description	Range	Data type	Data type (label)
(U)	GP.BUFRCV	Start I/O number (first three digits in four-digit hexadecimal representation) of own station or own node	00H to FEH	16-bit unsigned binary	ANY16
	ZP.BUFRCV	Start I/O number (first three digits in four-digit hexadecimal representation) of own station or own node	00H to FEH	String	ANY16_OR_STRING_SINGLE
(s1)		Connection No.	1 to 16	16-bit unsigned binary	ANY16
(s2)		Own station start device where control data is stored	Refer to the control data.	Device name	ANY16[*1]
(d1)		Own station start device for storing the receive data	—	Device name	ANY16[*1]
(d2)		Device of the own station, which turns on for one scan upon completion of the instruction. When the instruction completes with an error, (d2)+1 also turns on.	—	Bit	ANYBIT_ARRAY (Number of elements: 2)
EN		Execution condition	—	Bit	BOOL
ENO		Execution result	—	Bit	BOOL

[*1] When specifying setting data by using a label, define an array to secure enough operation area and specify an element of the array label.

■Applicable devices

Operand		Bit		Word			Double word		Indirect specification	Constant			Others (U)
		X, Y, M, L, SM, F, B, SB, FX, FY	J□\□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□\G□, J□\□, U3E□\(H)G□	Z	LT, LST, LC	LZ		K, H	E	$	
(U)	GP.B UFR CV	—	—	○	—	—	—	—	○	○	—	—	○
	ZP.B UFR CV	—	—	○	—	—	—	—	○	—	—	○	○
(s1)		○*1	—	○*2	—	—	—	—	○	○	—	—	—
(s2)		—	—	○*2	—	—	—	—	○	—	—	—	—
(d1)		—	—	○*2	—	—	—	—	○	—	—	—	—
(d2)		○*1	—	○*3	—	—	—	—	—	—	—	—	—

*1 FX and FY cannot be used.

*2 FD cannot be used.

*3 T, ST, C, and FD cannot be used.

■Control data

Operand: (s2)				
Device	Item	Description	Setting range	Set by
+0	System area	—	—	—
+1	Completion status	The completion status is stored upon completion of the instruction. • 0: Completed successfully • Other than 0: Completed with an error (error code)	—	System

■Receive data

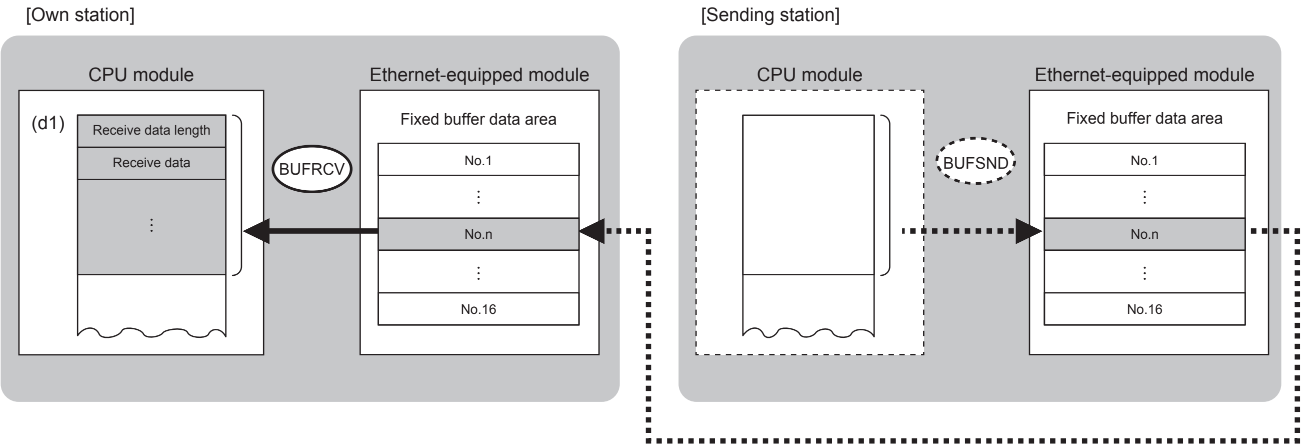
Operand: (d1)					
Device	Item	Description		Setting range	Set by
+0	Receive data length	The length of the data read from the fixed buffer data area is stored. (The data length is represented in words or bytes according to the procedure of fixed buffer communications.)	Procedure exist (binary): Number of words	1 to 5113 (1 to 1017)*1	System
			Procedure exist (ASCII): Number of words	1 to 2556 (1 to 508)*1	
			No procedure (binary): Number of bytes	1 to 10238 (1 to 2046)*1	
+1 to +□	Receive data	The data read from the fixed buffer data areas is stored sequentially in ascending order of addresses.		—	System

*1 This setting range is applicable when the network type is "Q-compatible Ethernet".

Processing details

- These instructions read the receive data (fixed buffer communications area) of the connection specified by (s1) of the module specified by (U). These instructions can be used only for the connections for which "Fixed Buffer (Procedure Exist/ No Procedure)" is set as the communication method.

[Reading receive data from the sending station to the own station]



No.: Connection No.

- The execution status and the completion status of the BUFRCV instruction can be checked with the completion device (d2) and the completion status indication device (d2)+1.
  - Completion device (d2)

This device turns on during END processing of the scan where the BUFRCV instruction completes, and turns off during the next END processing.

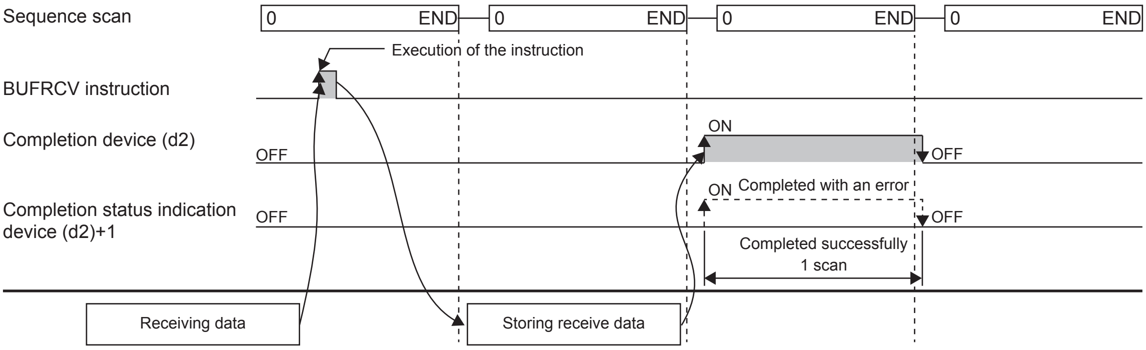
  - Completion status indication device (d2)+1

This device turns on or off depending on the completion status of the BUFRCV instruction.

When completed successfully: The device remains off.

When completed with an error: The device turns on during END processing of the scan where the BUFRCV instruction completes, and turns off during the next END processing.

- The following figure shows the operation at completion of the BUFRCV instruction.



- The BUFRCV instruction is executed on the rising edge (OFF to ON) of the socket/fixed buffer receive status signal (Un\G1900016 to Un\G1900023).

Restriction

When the BUFRCV instruction reads receive data from the same connection, it cannot be used in combination with the BUFRCVS instruction (for interrupt programs).

Operation error

Error code ((s2)+1)	Description
C000H to CFFFH	🕮 MELSEC iQ-R Ethernet User's Manual (Application)

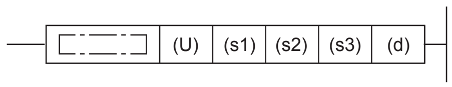
Appendix 5.5 Sending data

GP.BUFSND, ZP.BUFSND

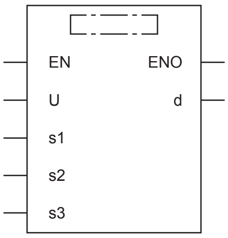
RnCPU RnENCPU RnPCPU (Process) RnPCPU (Redundant) RnSFCPU (Standard) ~~RnSFCPU (Safety)~~

These instructions send data to the external device through fixed buffer communications.

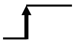
Ladder	ST

	ENO: =GP_BUFSND(EN,U,s1,s2,s3,d);
	ENO: =ZP_BUFSND(EN,U,s1,s2,s3,d);

FBD/LD



■Execution condition

Instruction	Execution condition
GP.BUFSND ZP.BUFSND	

Setting data

■Description, range, data type

Operand		Description	Range	Data type	Data type (label)
(U)	GP.BUFSND	Start I/O number (first three digits in four-digit hexadecimal representation) of own station or own node	00H to FEH	16-bit unsigned binary	ANY16
	ZP.BUFSND	Start I/O number (first three digits in four-digit hexadecimal representation) of own station or own node	00H to FEH	String	ANY16_OR_STRING_SINGLE
(s1)		Connection No.	1 to 16	16-bit unsigned binary	ANY16
(s2)		Own station start device where control data is stored	Refer to the control data.	Device name	ANY16[*1]
(s3)		Own station start device where send data is stored	—	Device name	ANY16[*1]
(d)		Device of the own station, which turns on for one scan upon completion of the instruction. When the instruction completes with an error, (d)+1 also turns on.	—	Bit	ANYBIT_ARRAY (Number of elements: 2)
EN		Execution condition	—	Bit	BOOL
ENO		Execution result	—	Bit	BOOL

[*1] When specifying setting data by using a label, define an array to secure enough operation area and specify an element of the array label.

■Applicable devices

Operand		Bit		Word			Double word		Indirect specification	Constant			Others (U)
		X, Y, M, L, SM, F, B, SB, FX, FY	J□\□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□\G□, J□\□, U3E□\(H)G□	Z	LT, LST, LC	LZ		K, H	E	$	
(U)	GP.B UFS ND	—	—	○	—	—	—	—	○	○	—	—	○
	ZP.B UFS ND	—	—	○	—	—	—	—	○	—	—	○	○
(s1)		○*1	—	○*2	—	—	—	—	○	○	—	—	—
(s2)		—	—	○*2	—	—	—	—	○	—	—	—	—
(s3)		—	—	○*2	—	—	—	—	○	—	—	—	—
(d)		○*1	—	○*3	—	—	—	—	—	—	—	—	—

*1 FX and FY cannot be used.

*2 FD cannot be used.

*3 T, ST, C, and FD cannot be used.

■Control data

Operand: (s2)				
Device	**Item**	**Description**	**Setting range**	**Set by**
+0	System area	—	—	—
+1	Completion status	The completion status is stored upon completion of the instruction. • 0: Completed successfully • Other than 0: Completed with an error (error code)	—	System

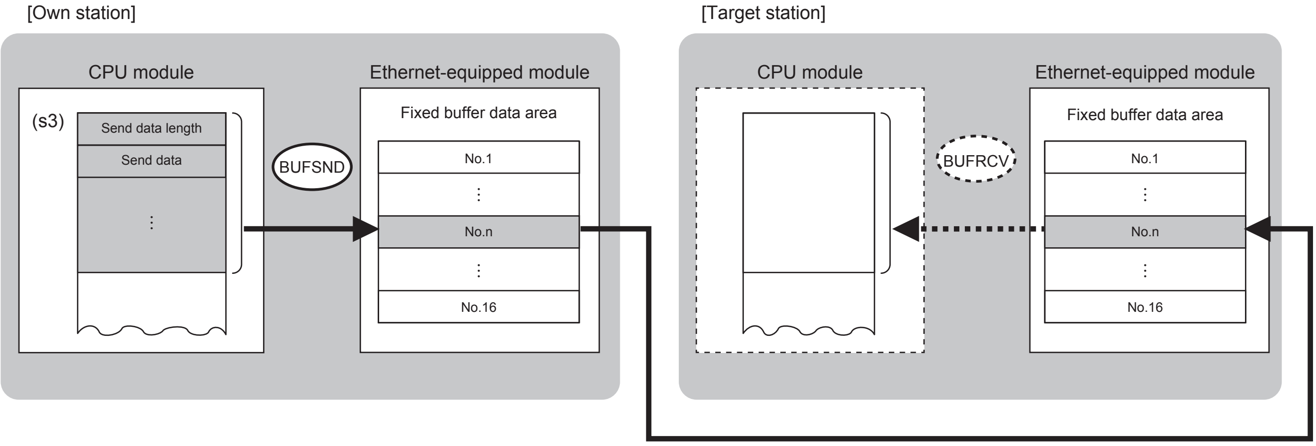
■Send data

Operand: (s3)					
Device	**Item**	**Description**		**Setting range**	**Set by**
+0	Send data length	Specify the send data length. (Specify the data length in words or bytes according to the procedure of fixed buffer communications.)	Procedure exist (binary): Number of words	1 to 5113 (1 to 1017)*1	User
			Procedure exist (ASCII): Number of words	1 to 2556 (1 to 508)*1	
			No procedure (binary): Number of bytes	1 to 10238 (1 to 2046)*1	
+1 to +□	Send data	Specify the send data.		—	User

*1 This setting range is applicable when the network type is "Q-compatible Ethernet".

Processing details

- These instruction send the data in the device specified by (s3) to the external device of the connection specified by (s1) of the module specified by (U). These instructions can be used only for the connections for which "Fixed Buffer (Procedure Exist/No Procedure)" is set as the communication method.



No.: Connection No.

- The execution status and the completion status of the BUFSND instruction can be checked with the completion device (d) and the completion status indication device (d)+1.

 • Completion device (d)

This device turns on during END processing of the scan where the BUFSND instruction completes, and turns off during the next END processing.

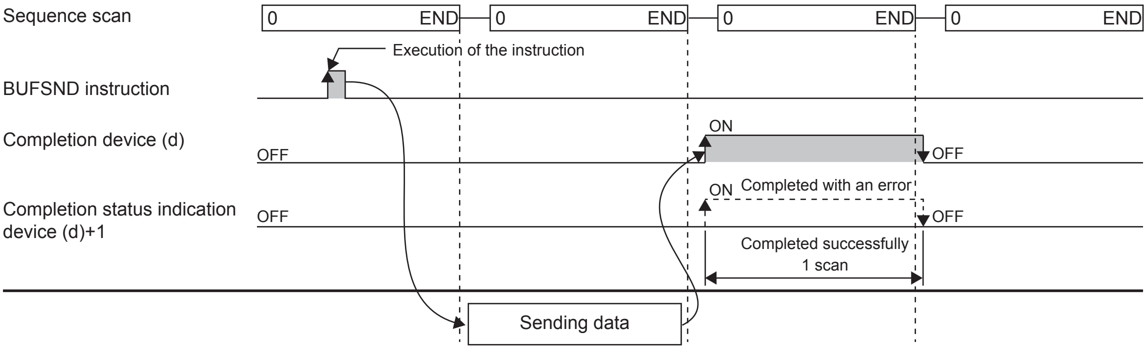
 • Completion status indication device (d)+1

This device turns on or off depending on the completion status of the BUFSND instruction.

When completed successfully: The device remains off.

When completed with an error: The device turns on during END processing of the scan where the BUFSND instruction completes, and turns off during the next END processing.

- The following figure shows the operation at completion of the BUFSND instruction.



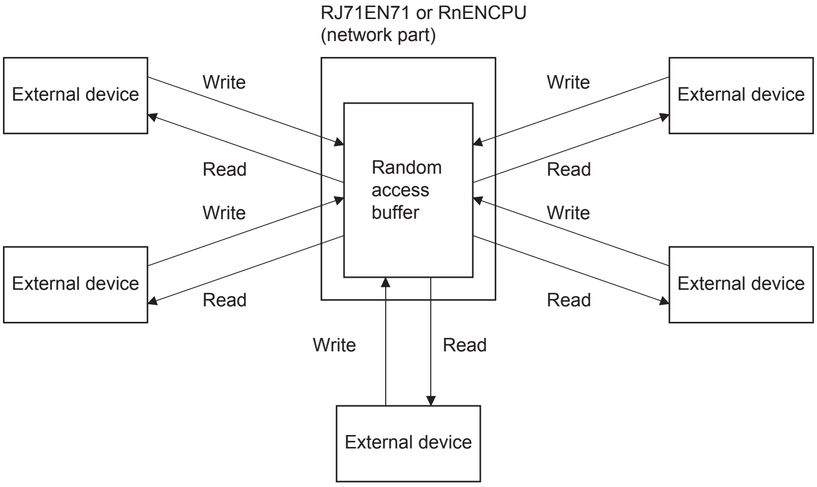
- The BUFSND instruction is executed on the rising edge (OFF to ON) of the send command.

Operation error

Error code ((s2)+1)	Description
C000H to CFFFH	🕮 MELSEC iQ-R Ethernet User's Manual (Application)

Appendix 6 Communications Using the Random Access Buffer

With communications using the random access buffer, data can be freely read and written between any external device (excluding Ethernet-equipped module) and the RJ71EN71 or the RnENCPU (network part). The external device does not need to be fixed. The random access buffer is used as the common buffer area for external devices connected to Ethernet.



Precautions

The CPU module (CPU part for the RnENCPU) cannot exchange data with the random access buffer.

Appendix 6.1 Setting procedure

Set "External Device Configuration" under "Basic Settings".

1. Select the external device to be connected in "Module List", and drag and drop it to the list of devices or device map.

External device name	Description
UDP Connection Module	Select to communicate with the external device using UDP/IP.
Active Connection Module	Select to perform the open processing to the external device from the Ethernet-equipped module (Active open) and communicate using TCP/IP.
Unpassive Connection Module	Select to receive the open processing from an unspecified external device (Unpassive open) and communicate using TCP/IP.
Fullpassive Connection Module	Select to receive the open processing from the specified external device (Fullpassive open) and communicate using TCP/IP.

2. Set "Communication Method" for external device to "Random Access Buffer".
3. Set the IP address of the external device.
4. Set the other parameters required for communications in the connections.

Appendix 6.2 Communication process

This section describes the mechanism of communications using the random access buffer.

Data flow

This section describes the flow of data communications using the random access buffer.

A dedicated packet is used to exchange data between the external device and the RJ71EN71 or the RnENCPU (network part).

A program is used to access the random access buffer from the CPU module.

Point

The process is executed asynchronously with the program, so if synchronization is required, use socket communications or communications using the fixed buffer.

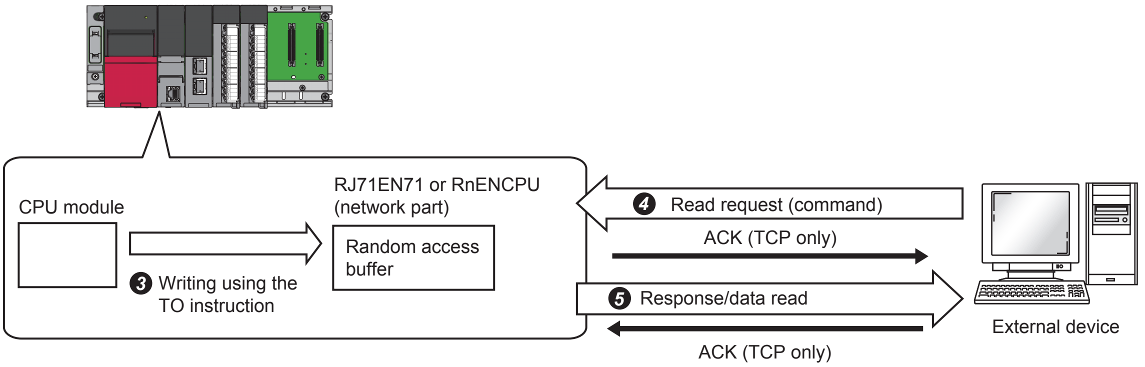
External devices capable of data communications

Data can be exchanged with the following external devices.

- Devices in Ethernet to which the RJ71EN71 is connected
- Devices in Ethernet to which the RnENCPU (network part) is connected
- Devices connected via a router

Appendix 6.3 Procedure for reading data from the external device

The following figure shows the processing when sending data from the RJ71EN71 or the RnENCPU (network part) in response to a read request from the external device.



❶ After the module parameters are set, check that the initial processing of the RJ71EN71 or the RnENCPU (network part) has completed normally. ('Initial status' (Un\G1900024.0): On)

❷ The open processing is performed to establish a connection between the RJ71EN71 or the RnENCPU (network part) and external device. (☞ Page 2 - 21 Open/Close Processing)

❸ The program writes the data to the random access buffer of the RJ71EN71 or the RnENCPU (network part).

❹ The read request is sent from the external device to the RJ71EN71 or the RnENCPU (network part) asynchronously from the above processing. (RJ71EN71 or RnENCPU (network part) side: Receives command)

❺ When the read request is received from the external device, the RJ71EN71 or the RnENCPU (network part) send the data written to the random access buffer to the external device. (RJ71EN71 or RnENCPU (network part) side: Sends response)

❻ Close the connection when communications are finished.

Appendix 6.4 Procedure for writing data from the external device

This section describes the procedure when writing data from the external device to the random access buffer of the RJ71EN71 or the RnENCPU (network part).

1. After the module parameters are set, check that the initial processing of the RJ71EN71 or the RnENCPU (network part) has completed normally. ('Initial status' (Un\G1900024.0): On)
2. The open processing is performed to establish a connection between the RJ71EN71 or the RnENCPU (network part) and external device.
3. Data is written from the external device to the random access buffer of the RJ71EN71 or the RnENCPU (network part). (RJ71EN71 or RnENCPU (network part) side: Receives command)
4. The RJ71EN71 or the RnENCPU (network part) performs the write processing requested by the external device, and returns the write results to the external device that sent the write request. (RJ71EN71 or RnENCPU (network part) side: Sends response)
5. The data written to the random access buffer is read asynchronously from the above processing by the program.
6. Close the connection when communications are finished.

Appendix 6.5 Data format

The communication data is configured of the header and application data.

Header

The header is for TCP/IP or UDP/IP. The RJ71EN71 or the RnENCPU (network part) automatically adds and deletes the header, the setting is not required.

■Details of header section size

The details of the header section data format and size are shown below.

- TCP/IP

Ethernet (14 bytes)	IP (20 bytes)	TCP (20 bytes)

- UDP/IP

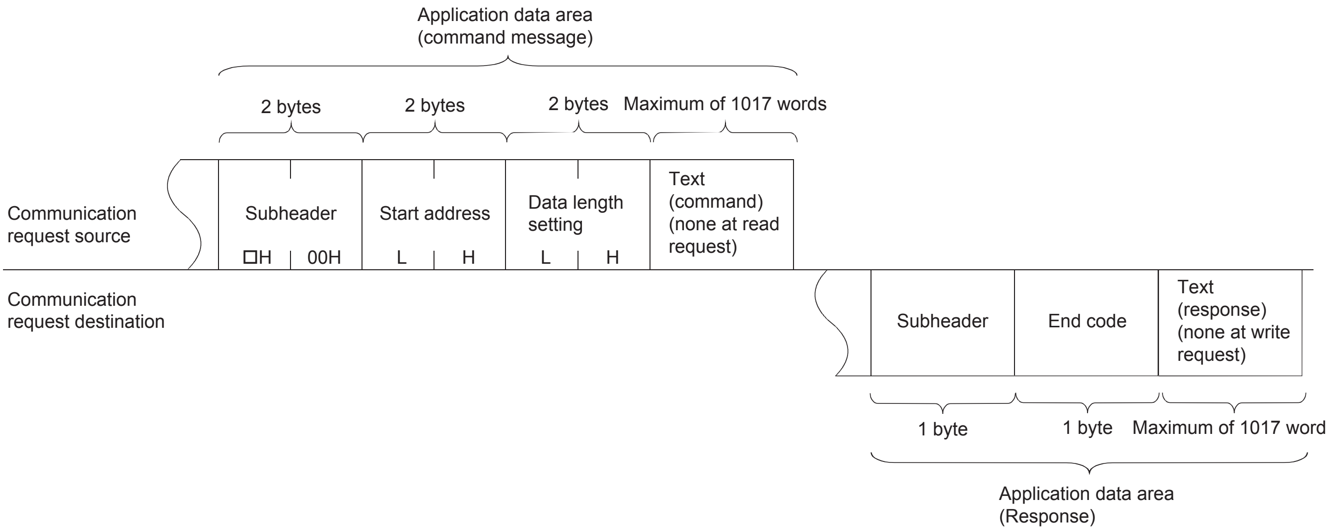
Ethernet (14 bytes)	IP (20 bytes)	UDP (8 bytes)

Application data

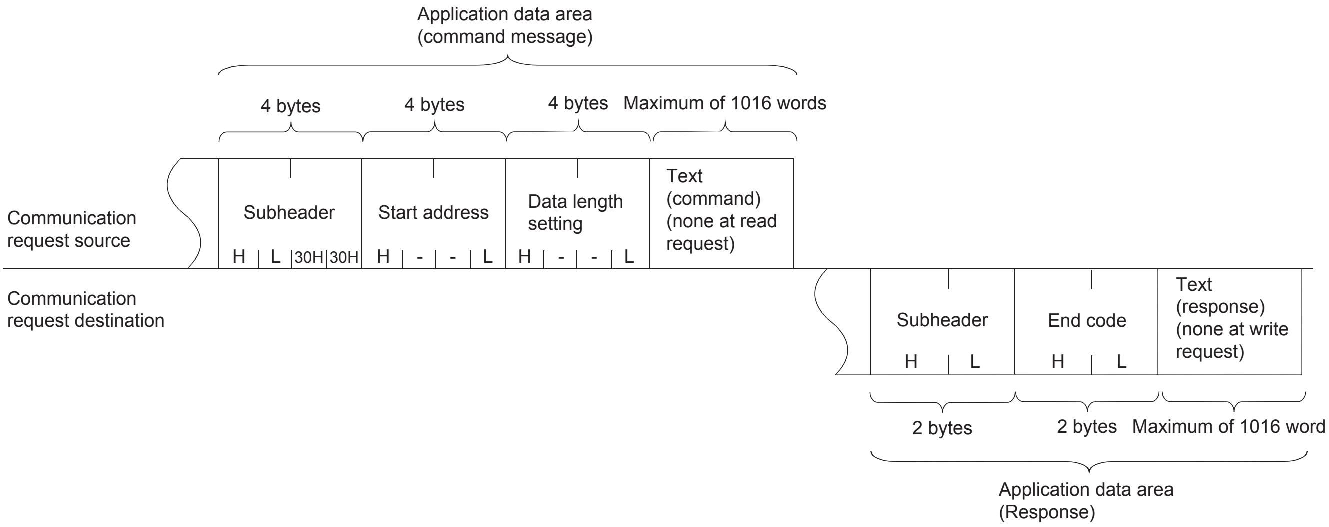
The application data expresses the following data code as binary code or ASCII code. Switch between the binary code and ASCII code with "Own Node Settings" under "Basic Settings".

■Format

- When exchanging data with binary codes

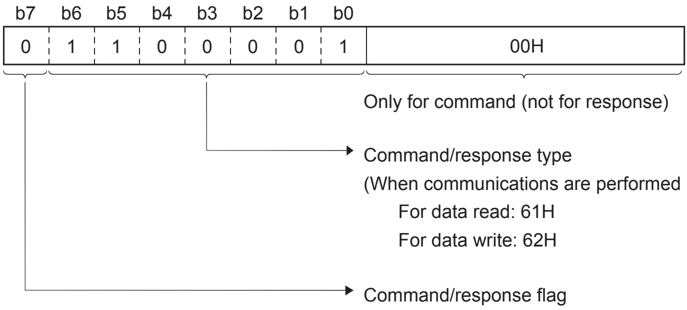


- When exchanging data with ASCII codes



■Subheader

The RJ71EN71 or the RnENCPU (network part) automatically adds and deletes the subheader, the setting is not required.



Only for command (not for response)

Command/response type

(When communications are performed using the random access buffer, this format is used.)

For data read: 61H

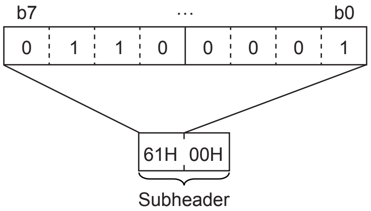
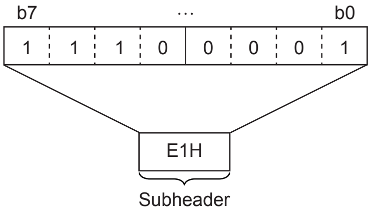
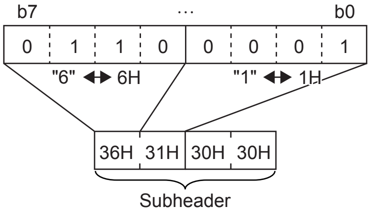
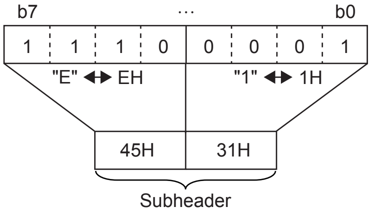
For data write: 62H

Command/response flag

For command: 0

For response: 1

- During read

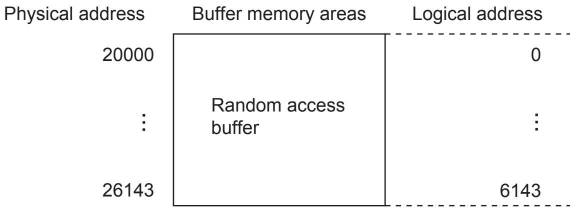
Data format	Command (external device → RJ71EN71 or RnENCPU (network part))	Response (RJ71EN71 or RnENCPU (network part) → external device)
Binary code		
ASCII code	 b7 … b0 0 1 1 0 0 0 0 1 "6" ◀▶ 6H "1" ◀▶ 1H 36H 31H 30H 30H Subheader	 b7 … b0 1 1 1 0 0 0 0 1 "E" ◀▶ EH "1" ◀▶ 1H 45H 31H Subheader

Appendix 6.6 Physical address and logical address of random access buffer

This section describes the start address of the random access buffer of the RJ71EN71 or the RnENCPU (network part) specified in the command.

The address specified for the random access buffer differs from the address specified by the external device and the address specified with the FROM/TO instruction.

- Physical address: Address specified with the FROM/TO instruction in the program
- Logical address: Address specified by the external device in start address item of command



Appendix 6.7 Precautions

This section describes the precautions for communications using the random access buffer.

Precautions for creating programs

■Initial processing and open processing completion

The initial processing and open processing for the connection must complete.

■Send request from the CPU module

Send cannot be requested from the CPU module. Receive completion of CPU module is not checked. If the send/receive processing must be synchronized between the CPU module and external device, use fixed buffer communications.

■Random access buffer address

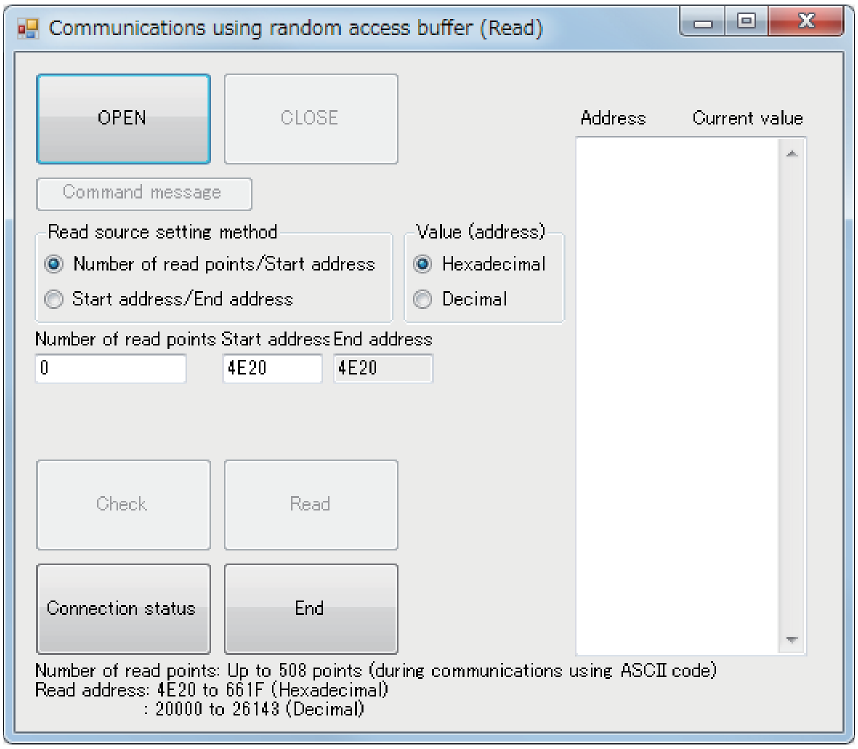
The address specified by the external device is different from the address specified with the FROM/TO instruction. For details, refer to the following.

(☞ Page App. - 84 Physical address and logical address of random access buffer)

Appendix 6.8 Communications using the random access buffer (Read)

Read data using the random access buffer.

■Execution window



Read source setting method
- ● Number of read points/Start address
- ○ Start address/End address

Value (address)
- ● Hexadecimal
- ○ Decimal

Number of read points	Start address	End address
0	4E20	4E20

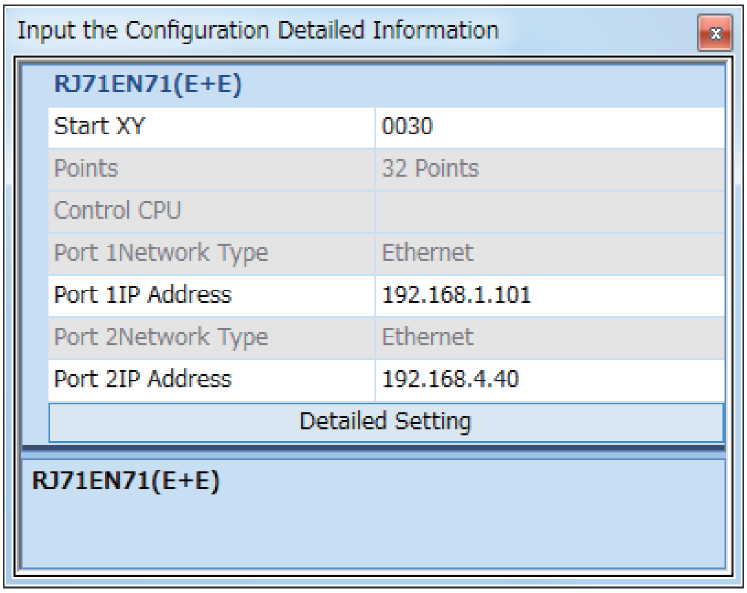
Number of read points: Up to 508 points (during communications using ASCII code)
Read address: 4E20 to 661F (Hexadecimal)
: 20000 to 26143 (Decimal)

■IP address setting on the personal computer

IP address on the personal computer: 192. 168. 1. 1

■Programmable controller setting

Configuration detailed information

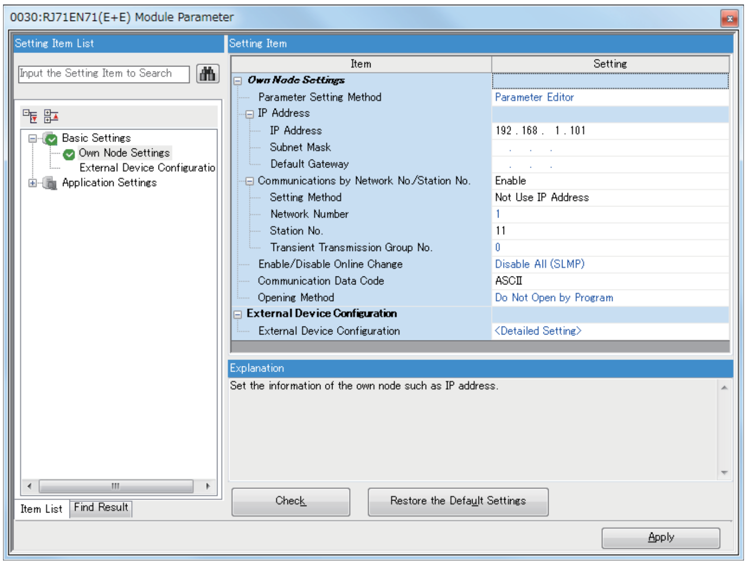


RJ71EN71(E+E)

Start XY	0030
Points	32 Points
Control CPU	
Port 1Network Type	Ethernet
Port 1IP Address	192.168.1.101
Port 2Network Type	Ethernet
Port 2IP Address	192.168.4.40

RJ71EN71(E+E)

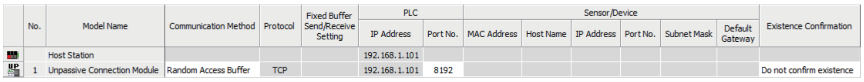
Setting the module parameter


0030:RJ71EN71(E+E) Module Parameter
Setting Item List
Input the Setting Item to Search
Basic Settings
Own Node Settings
External Device Configuratio
Application Settings
Item List
Find Result
Setting Item
Explanation
Check
Restore the Default Settings
Apply

Item	Setting
Own Node Settings	
Parameter Setting Method	Parameter Editor
IP Address	
IP Address	192 . 168 . 1 . 101
Subnet Mask	. . .
Default Gateway	. . .
Communications by Network No./Station No.	Enable
Setting Method	Not Use IP Address
Network Number	1
Station No.	11
Transient Transmission Group No.	0
Enable/Disable Online Change	Disable All (SLMP)
Communication Data Code	ASCII
Opening Method	Do Not Open by Program
External Device Configuration	
External Device Configuration	<Detailed Setting>

Set the information of the own node such as IP address.

Setting the external device configuration



	No.	Model Name	Communication Method	Protocol	Fixed Buffer Send/Receive Setting	PLC		Sensor/Device						Existence Confirmation
						IP Address	Port No.	MAC Address	Host Name	IP Address	Port No.	Subnet Mask	Default Gateway	
		Host Station				192.168.1.101								
	1	Unpassive Connection Module	Random Access Buffer	TCP		192.168.1.101	8192							Do not confirm existence

■Operation check procedure

1. Write data to the random access buffer memory areas.
2. Click the "OPEN" button to open the communications.
3. Set the number of read points and the target buffer memory address.
4. Click the "Check" button and check the settings.
5. Click the "Read" button to read data from the programmable controller.
6. Check the read data on the display of the personal computer.
7. Click the "CLOSE" button to close the communications.
8. Click the "End" button to end the processing.

■Visual Basic 2012 Program

```
Option Strict Off
Option Explicit On

Imports System
Imports System.Text
Imports System.Net

Public Class Form1
  Private state As Boolean
  Private Ajsock As Sockets.Socket

  Private Sub cmdOPEN_Click(sender As Object, e As EventArgs) Handles cmdOPEN.Click
    'Connect the programmable controller with the Ethernet-equipped module.
    Dim sock As New Sockets.Socket(Sockets.AddressFamily.InterNetwork, _
                                   Sockets.SocketType.Stream, Sockets.ProtocolType.Tcp)
    Ajsock = sock
    Dim ip As IPAddress
    Dim ipend As IPEndPoint

    Try
      ip = IPAddress.Parse("192.168.1.101")
      ipend = New IPEndPoint(ip, 8192)
      Ajsock.Connect(ipend)
    Catch ex As Exception
      MsgBox("Data communications not performed")
      Exit Sub
    End Try

    MsgBox("Connected")
    state = Ajsock.Connected

    cmdTEST.Enabled = True
    cmdCLOSE.Enabled = True
    cmdCHECK.Enabled = True
    cmdEND.Enabled = False
    cmdOPEN.Enabled = False

  End Sub

  Private Sub cmdTEST_Click(sender As Object, e As EventArgs) Handles cmdTEST.Click
    Dim sData As Byte()
    Dim rData(256) As Byte

    ' Read data of the buffer memory addresses H4E20 and H4E21.
    ' "6100" ...Subheader (61 is a read command.)
    ' "0000" ...Read address (specified by logical address)
    ' "0002" ...Number of read points
    sData = Encoding.ASCII.GetBytes("610000000002")
    ' Send data.
    Ajsock.Send(sData)

    MsgBox("Sending data completed", MsgBoxStyle.Information)

    ' Read the response message (response) from the CPU module.
    ' "E1" ...Subheader (E1 is a read response.)
    ' "00" ...End code (00 is a normal end.)
    Ajsock.Receive(rData)
    MsgBox(Encoding.ASCII.GetString(rData), MsgBoxStyle.Information)
  End Sub
```

```
Private Sub cmdREAD_Click(sender As Object, e As EventArgs) Handles cmdREAD.Click
    Dim sData As Byte()
    Dim rData(2048) As Byte
    Dim Address As String               ' Read start address
    Dim Number As String                ' Number of read points
    Dim rDataCheck1 As String           ' Read data
    Dim rDataCheck2 As String = ""      ' Read data
    Dim rDataCheck(508) As String       ' Read data
    Dim n As Integer
    Dim ni As Integer

    ' Read data from the random access buffer memory areas.
    ' Convert read addresses.
    If (rbHEX.Checked = True) Then
        Address = Convert.ToInt16(Convert.ToInt16(sAddress.Text, 16) - 20000).ToString("X4")
    Else
        Address = Convert.ToInt16(sAddress.Text - 20000).ToString("X4")
    End If

    ' Convert the number of read points.
    Number = Convert.ToInt16(wNumber.Text).ToString("X4")

    ' Encode the addresses and number of points as a command message.
    sData = Encoding.ASCII.GetBytes("6100" + Address + Number)

    ' Send data.
    Ajsock.Send(sData)
    MsgBox("Sending data completed", MsgBoxStyle.Information)

    ' Read the response message (response) from the CPU module.
    Ajsock.Receive(rData)
    MsgBox(Encoding.ASCII.GetString(rData), MsgBoxStyle.Information)

    ' Display the response in the text box.
    n = wNumber.Text
    rDataCheck1 = Encoding.ASCII.GetString(rData)
    For ni = 0 To (n - 1)
        rDataCheck(ni) = Convert.ToInt16(Strings.Mid(rDataCheck1, ni * 4 + 5, 4), 16)
    Next

    If (rbHEX.Checked = True) Then
        For ni = 0 To (n - 1)
            rDataCheck2 = rDataCheck2 & " H" & Hex(Convert.ToInt16(sAddress.Text, 16) + ni) & " = " _
                & Convert.ToInt16(rDataCheck(ni)) & vbCrLf
        Next
    Else
        For ni = 0 To (n - 1)
            rDataCheck2 = rDataCheck2 & "  " & Convert.ToInt16(sAddress.Text) + ni & " = " _
                & Convert.ToInt16(rDataCheck(ni)) & vbCrLf
        Next
    End If

    TextBox1.Text = rDataCheck2
    cmdREAD.Enabled = False
End Sub
```

```
Private Sub cmdCLOSE_Click(sender As Object, e As EventArgs) Handles cmdCLOSE.Click
    ' Close the sockets connected to TCP (UDP). (Disconnection)
    Ajsock.Shutdown(Net.Sockets.SocketShutdown.Both)
    Ajsock.Close()
    MsgBox("Disconnected", MsgBoxStyle.Information)
    state = Ajsock.Connected()

    cmdTEST.Enabled = False
    cmdOPEN.Enabled = True
    cmdCHECK.Enabled = False
    cmdEND.Enabled = True
    cmdCLOSE.Enabled = False
    cmdREAD.Enabled = False
End Sub

Private Sub cmdEND_Click(sender As Object, e As EventArgs) Handles cmdEND.Click
    End
End Sub

Private Sub cmdSTATUS_Click(sender As Object, e As EventArgs) Handles cmdSTATUS.Click
    ' Check the connection status.
    If state Then
        MsgBox("Connecting")
    Else
        MsgBox("Closed")
    End If
End Sub

Private Sub wNumber_sAddress_CheckedChanged(sender As Object, e As EventArgs) Handles wNumber_sAddress.CheckedChanged
    ' Change the operation when the radio buttons in the Read source setting method are operated.
    If (wNumber_sAddress.Checked = True) Then
        wNumber.ReadOnly = False
        lAddress.ReadOnly = True
    Else
        lAddress.ReadOnly = False
        wNumber.ReadOnly = True
    End If
End Sub

Private Sub rbHEX_CheckedChanged(sender As Object, e As EventArgs) Handles rbHEX.CheckedChanged
    ' Convert address numbers to decimal or hexadecimal.
    If ((sAddress.Text <> "") And (lAddress.Text <> "")) Then
        If (rbHEX.Checked = True) Then
            Try
                sAddress.Text = Hex(sAddress.Text)
                lAddress.Text = Hex(lAddress.Text)
            Catch ex As Exception                   ' Exception handling for when characters have been pasted from the right-click menu
                sAddress.Text = "4E20"
                lAddress.Text = "4E20"
            End Try
        Else
            Try
                sAddress.Text = Convert.ToString(Convert.ToInt32(sAddress.Text, 16))
                lAddress.Text = Convert.ToString(Convert.ToInt32(lAddress.Text, 16))
            Catch ex As Exception                   ' Exception handling for when characters have been pasted from the right-click menu
                sAddress.Text = 20000
                lAddress.Text = 20000
            End Try
        End If
    End If
End Sub
```

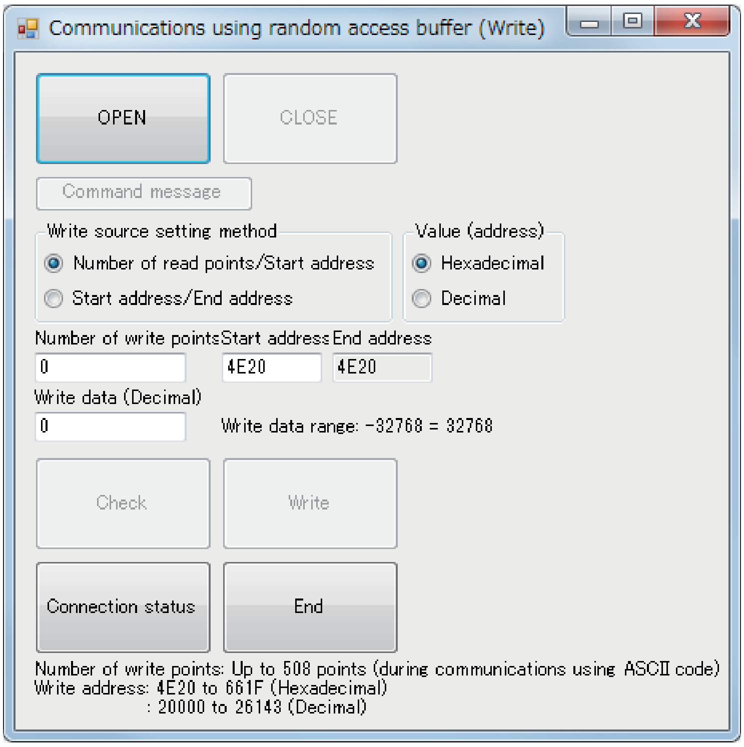
```
Private Sub cmdCHECK_Click(sender As Object, e As EventArgs) Handles cmdCHECK.Click
    ' Check the number of read points, read start address, and end address.
    ' Number of read points...Up to 508 words (during communications using ASCII code)
    ' Address range...H4E20 (20000) to H661F (26143) Physical address (buffer memory address)
    '                 H0 (0) to H17FF (6143) Physical address
    Dim a As Integer
    Dim b As Integer
    Dim c As Integer
    Dim d As Integer

    'Get the number of read points and address number.
    Try
        If (rbHEX.Checked = True) Then
            a = Convert.ToInt32(sAddress.Text, 16)
            b = wNumber.Text
            c = Convert.ToInt32(lAddress.Text, 16)
        Else
            a = sAddress.Text
            b = wNumber.Text
            c = lAddress.Text
        End If
    Catch ex As Exception                      ' Exception handling for when characters have been pasted from the right-click menu
        Exit Sub
    End Try

    ' Check the setting when Number of read points/Start address is selected in the Read source setting method.
    If (wNumber_sAddress.Checked = True) Then
        d = a + b - 1
        If ((b >= 1) And (b <= 508) And (a >= 20000) And (a <= 26143) And (d <= 26143)) Then
            If (rbHEX.Checked = True) Then
                lAddress.Text = Hex(d)
            Else
                lAddress.Text = d
            End If
            cmdREAD.Enabled = True
        Else
            MessageBox.Show("Review the setting.", "Check")
            If (rbHEX.Checked = True) Then
                lAddress.Text = Hex(d)
            Else
                lAddress.Text = d
            End If
            cmdREAD.Enabled = False
        End If
    Else
        ' Check the setting when Start address/End address is selected in the Read source setting method.
        d = c - a + 1
        If ((d <= 508) And (a >= 20000) And (a <= 26143) And (a <= c) And (c <= 26143)) Then
            wNumber.Text = d
            cmdREAD.Enabled = True
        Else
            MessageBox.Show("Review the setting.", "Check")
            wNumber.Text = d
            cmdREAD.Enabled = False
        End If
    End If
End Sub
```

```
    Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load
        ' Disable the IME mode in the text box.
        wNumber.ImeMode = Windows.Forms.ImeMode.Disable
        sAddress.ImeMode = Windows.Forms.ImeMode.Disable
        lAddress.ImeMode = Windows.Forms.ImeMode.Disable
    End Sub

    Private Sub wNumber_KeyPress(sender As Object, e As KeyPressEventArgs) Handles wNumber.KeyPress
        ' Prohibit non-numeric input in the text box for the number of read points.
        Dim intKey As Integer = Asc(e.KeyChar)

        If (intKey < 47) Or (intKey > 57) Then
            If (intKey <> 8) Then
                e.Handled = True
            End If
        End If
    End Sub

    Private Sub sAddress_KeyPress(sender As Object, e As KeyPressEventArgs) Handles sAddress.KeyPress
        ' Prohibit non-numeric input in the text box for the start address (for decimal).
        ' Prohibit input of other than numerical values, a to f, and A to F in the text box for the start address (for hexadecimal).
        Dim intKey As Integer = Asc(e.KeyChar)

        If (rbHEX.Checked = True) Then
            If ((intKey < 47) Or (intKey > 57)) And ((intKey < 64) Or (intKey > 70)) And ((intKey < 96)
               Or (intKey > 102)) Then
                If (intKey <> 8) Then
                    e.Handled = True
                End If
            End If
        Else
            If (intKey < 47) Or (intKey > 57) Then
                If (intKey <> 8) Then
                    e.Handled = True
                End If
            End If
        End If
    End Sub

    Private Sub lAddress_KeyPress(sender As Object, e As KeyPressEventArgs) Handles lAddress.KeyPress
        ' Prohibit non-numeric input in the text box for the end address (for decimal).
        ' Prohibit input of other than numerical values, a to f, and A to F in the text box for the end address (for hexadecimal).
        Dim intKey As Integer = Asc(e.KeyChar)

        If (rbHEX.Checked = True) Then
            If ((intKey < 47) Or (intKey > 57)) And ((intKey < 64) Or (intKey > 70))
               And ((intKey < 96) Or (intKey > 102)) Then
                If (intKey <> 8) Then
                    e.Handled = True
                End If
            End If
        Else
            If (intKey < 47) Or (intKey > 57) Then
                If (intKey <> 8) Then
                    e.Handled = True
                End If
            End If
        End If
    End Sub


End Class
```

Appendix 6.9 Communications using the random access buffer (Write)

Write data using the random access buffer.

■Execution window



Write source setting method

● Number of read points/Start address

○ Start address/End address

Value (address)

● Hexadecimal

○ Decimal

Number of write points	Start address	End address
0	4E20	4E20

Write data (Decimal): 0

Write data range: -32768 = 32768

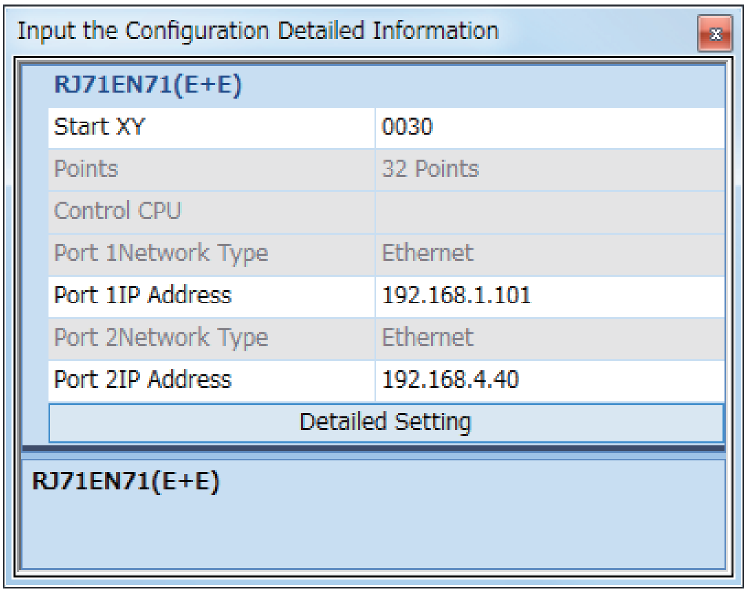
Number of write points: Up to 508 points (during communications using ASCII code)
Write address: 4E20 to 661F (Hexadecimal)
: 20000 to 26143 (Decimal)

■IP address setting on the personal computer

IP address on the personal computer: 192. 168. 1. 1

■Programmable controller setting

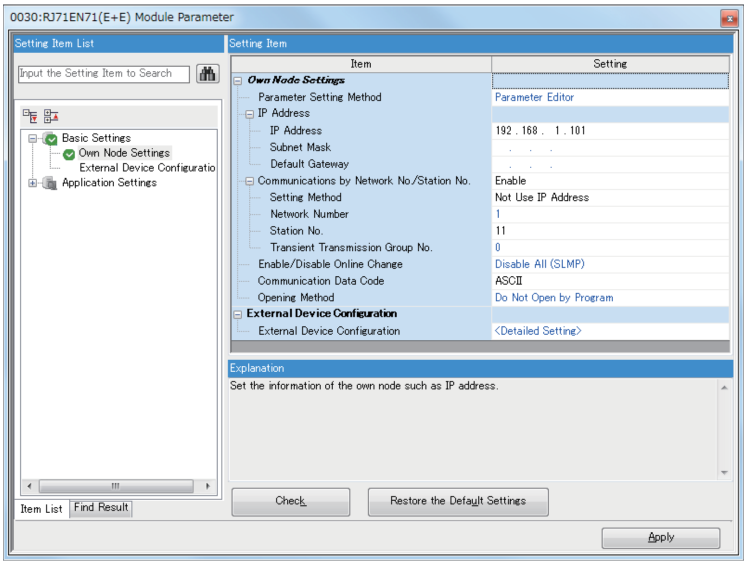
Configuration detailed information



RJ71EN71(E+E)	
Start XY	0030
Points	32 Points
Control CPU	
Port 1Network Type	Ethernet
Port 1IP Address	192.168.1.101
Port 2Network Type	Ethernet
Port 2IP Address	192.168.4.40

RJ71EN71(E+E)

Setting the module parameter

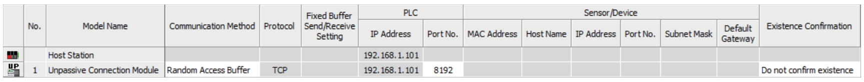

0030:RJ71EN71(E+E) Module Parameter

Item	Setting
Own Node Settings	
Parameter Setting Method	Parameter Editor
IP Address	
IP Address	192 . 168 . 1 . 101
Subnet Mask	. . .
Default Gateway	. . .
Communications by Network No./Station No.	Enable
Setting Method	Not Use IP Address
Network Number	1
Station No.	11
Transient Transmission Group No.	0
Enable/Disable Online Change	Disable All (SLMP)
Communication Data Code	ASCII
Opening Method	Do Not Open by Program
External Device Configuration	
External Device Configuration	<Detailed Setting>

Explanation

Set the information of the own node such as IP address.

Setting the external device configuration



	No.	Model Name	Communication Method	Protocol	Fixed Buffer Send/Receive Setting	PLC		Sensor/Device						Existence Confirmation
						IP Address	Port No.	MAC Address	Host Name	IP Address	Port No.	Subnet Mask	Default Gateway	
		Host Station				192.168.1.101								
	1	Unpassive Connection Module	Random Access Buffer	TCP		192.168.1.101	8192							Do not confirm existence

■Operation check procedure

1. Click the "OPEN" button to open the communications.

2. Set the number of read points, the target buffer memory address, and write data.

3. Click the "Check" button and check the settings.

4. Click the "Write" button to write data to the programmable controller.

5. Check the write processing on the "Device/Buffer Memory Batch Monitor" of GX Works3.

6. Click the "CLOSE" button to close the communications.

7. Click the "End" button to end the processing.

■Visual Basic 2012 Program

```
Option Strict Off
Option Explicit On

Imports System
Imports System.Text
Imports System.Net

Public Class Form1
    Private state As Boolean
    Private Ajsock As Sockets.Socket

    Private Sub cmdOPEN_Click(sender As Object, e As EventArgs) Handles cmdOPEN.Click
        'Connect the programmable controller with the Ethernet-equipped module.
        Dim sock As New Sockets.Socket(Sockets.AddressFamily.InterNetwork, _
                                Sockets.SocketType.Stream, Sockets.ProtocolType.Tcp)
        Ajsock = sock
        Dim ip As IPAddress
        Dim ipend As IPEndPoint

        Try
            ip = IPAddress.Parse("192.168.1.101")
            ipend = New IPEndPoint(ip, 8192)
            Ajsock.Connect(ipend)
        Catch ex As Exception
            MsgBox("Data communications not performed")
            Exit Sub
        End Try

        MsgBox("Connected")
        state = Ajsock.Connected

        cmdTEST.Enabled = True
        cmdCLOSE.Enabled = True
        cmdCHECK.Enabled = True
        cmdEND.Enabled = False
        cmdOPEN.Enabled = False
    End Sub

    Private Sub cmdTEST_Click(sender As Object, e As EventArgs) Handles cmdTEST.Click
        Dim sData As Byte()
        Dim rData(2048) As Byte

        ' Write H1234 and H5678 to the buffer memory addresses H4E20 and H4E21.
        ' "6200" ...Subheader (62 is a write command.)
        ' "0000" ...Write address (specified by logical address)
        ' "0002" ...Number of write points
        ' "1234" "5678" ...Write data
        sData = Encoding.ASCII.GetBytes("62000000000212345678")

        ' Send data.
        Ajsock.Send(sData)
        MsgBox("Sending data completed", MsgBoxStyle.Information)

       ' Read the response message (response) from the CPU module.
        ' "E2" ...Subheader (E2 is a write response.)
        ' "00" ...End code (00 is a normal end.)
        Ajsock.Receive(rData)
        MsgBox(Encoding.ASCII.GetString(rData), MsgBoxStyle.Information)
    End Sub
```

```
Private Sub cmdWRITE_Click(sender As Object, e As EventArgs) Handles cmdWRITE.Click
    Dim sData As Byte()
    Dim rData(256) As Byte
    Dim SD As String                                  ' Send data
    Dim Address As String                             ' Write start address
    Dim Number As String                              ' Number of write points
    Dim n As Integer
    Dim ni As Integer

    ' Write data to the random access buffer memory areas.
    ' Convert write addresses.
    If (rbHEX.Checked = True) Then
        Address = Convert.ToInt16(Convert.ToInt16(sAddress.Text, 16) - 20000).ToString("X4")
    Else
        Address = Convert.ToInt16(sAddress.Text - 20000).ToString("X4")
    End If

    ' Convert the number of write points.
    Number = Convert.ToInt16(wNumber.Text).ToString("X4")

    ' Convert write data.
    n = wNumber.Text
    SD = ""
    For ni = 0 To n - 1
        SD = SD + Convert.ToInt16(wData.Text).ToString("X4")
    Next

    ' Encode the addresses and number of points as a command message.
    sData = Encoding.ASCII.GetBytes("6200" + Address + Number + SD)
    ' Send data.
    Ajsock.Send(sData)
    MsgBox("Sending data completed", MsgBoxStyle.Information)

    ' Read the response message (response) from the CPU module.
    Ajsock.Receive(rData)
    MsgBox(Encoding.ASCII.GetString(rData), MsgBoxStyle.Information)
    cmdWRITE.Enabled = False
End Sub

Private Sub cmdCLOSE_Click(sender As Object, e As EventArgs) Handles cmdCLOSE.Click
    ' Close the sockets connected to TCP (UDP). (Disconnection)
    Ajsock.Shutdown(Net.Sockets.SocketShutdown.Both)
    Ajsock.Close()
    MsgBox("Disconnected", MsgBoxStyle.Information)
    state = Ajsock.Connected()

    cmdTEST.Enabled = False
    cmdOPEN.Enabled = True
    cmdCHECK.Enabled = False
    cmdEND.Enabled = True
    cmdCLOSE.Enabled = False
    cmdWRITE.Enabled = False
End Sub

Private Sub cmdEND_Click(sender As Object, e As EventArgs) Handles cmdEND.Click
    End
End Sub

Private Sub cmdSTATUS_Click(sender As Object, e As EventArgs) Handles cmdSTATUS.Click
    ' Check the connection status.
    If state Then
        MsgBox("Connecting")
    Else
        MsgBox("Closed")
    End If
End Sub
```

```
Private Sub wNumber_sAddress_CheckedChanged(sender As Object, e As EventArgs) Handles wNumber_sAddress.CheckedChanged
    ' Change the operation when the radio buttons in the Read source setting method are operated.
    If (wNumber_sAddress.Checked = True) Then
        wNumber.ReadOnly = False
        lAddress.ReadOnly = True
    Else
        lAddress.ReadOnly = False
        wNumber.ReadOnly = True
    End If
End Sub

Private Sub rbHEX_CheckedChanged(sender As Object, e As EventArgs) Handles rbHEX.CheckedChanged
    ' Convert address numbers to decimal or hexadecimal.
    If ((sAddress.Text <> "") And (lAddress.Text <> "")) Then
        If (rbHEX.Checked = True) Then
            Try
                sAddress.Text = Hex(sAddress.Text)
                lAddress.Text = Hex(lAddress.Text)
            Catch ex As Exception                    ' Exception handling for when characters have been pasted from the right-click menu
                sAddress.Text = "4E20"
                lAddress.Text = "4E20"
            End Try
        Else
            Try
                sAddress.Text = Convert.ToString(Convert.ToInt32(sAddress.Text, 16))
                lAddress.Text = Convert.ToString(Convert.ToInt32(lAddress.Text, 16))
            Catch ex As Exception                    ' Exception handling for when characters have been pasted from the right-click menu
                sAddress.Text = 20000
                lAddress.Text = 20000
            End Try
        End If
    End If
End Sub

Private Sub cmdCHECK_Click(sender As Object, e As EventArgs) Handles cmdCHECK.Click
    ' Check the number of write points, write start address, and write end address.
    ' Number of write points...Up to 508 words (during communications using ASCII code)
    ' Address range...H4E20 (20000) to H661F (26143) Physical address (buffer memory address)
    '                 H0 (0) to H17FF (6143) Logical address
    Dim a As Integer
    Dim b As Integer
    Dim c As Integer
    Dim d As Integer
    Dim f As Integer

    ' Get the number of write points and address number.
    Try
        If (rbHEX.Checked = True) Then
            a = Convert.ToInt32(sAddress.Text, 16)
            b = wNumber.Text
            c = Convert.ToInt32(lAddress.Text, 16)
            f = wData.Text
        Else
            a = sAddress.Text
            b = wNumber.Text
            c = lAddress.Text
            f = wData.Text
        End If
    Catch ex As Exception                            ' Exception handling for when characters have been pasted from the right-click menu
        MsgBox("Review the input.")
        Exit Sub
    End Try
```

```
   ' Check the setting when Number of write points/Start address is selected in the Write source setting method.
   If (wNumber_sAddress.Checked = True) Then
      d = a + b - 1
      If ((b >= 1) And (b <= 508) And (a >= 20000) And (a <= 26143) And (d <= 26143) And (f >= -32768)
        And (f <= 32768)) Then
         If (rbHEX.Checked = True) Then
            lAddress.Text = Hex(d)
         Else
            lAddress.Text = d
         End If
         cmdWRITE.Enabled = True
      Else
         MessageBox.Show("Review the setting.", "Check")
         If (rbHEX.Checked = True) Then
            lAddress.Text = Hex(d)
         Else
            lAddress.Text = d
         End If
         cmdWRITE.Enabled = False
      End If
   Else
      ' Check the setting when Start address/End address is selected in the Write source setting method.
      d = c - a + 1
      If ((d <= 508) And (a >= 20000) And (a <= 26143) And (a <= c) And (c <= 26143)
        And (f >= -32768) And (f <= 32768)) Then
         wNumber.Text = d
         cmdWRITE.Enabled = True
      Else
         MessageBox.Show("Review the setting.", "Check")
         wNumber.Text = d
         cmdWRITE.Enabled = False
      End If
   End If
End Sub

Private Sub Form1_Load(sender As Object, e As EventArgs) Handles MyBase.Load
   ' Disable the IME mode in the text box.
   wNumber.ImeMode = Windows.Forms.ImeMode.Disable
   sAddress.ImeMode = Windows.Forms.ImeMode.Disable
   lAddress.ImeMode = Windows.Forms.ImeMode.Disable
   wData.ImeMode = Windows.Forms.ImeMode.Disable
End Sub

Private Sub wData_KeyPress(sender As Object, e As KeyPressEventArgs) Handles wData.KeyPress
   ' Prohibit non-numeric input in the text box for the number of write points.
   Dim intKey As Integer = Asc(e.KeyChar)

   If ((intKey < 47) Or (intKey > 57)) And (intKey <> 45) Then
      If (intKey <> 8) Then
         e.Handled = True
      End If
   End If
End Sub

Private Sub wNumber_KeyPress(sender As Object, e As KeyPressEventArgs) Handles wNumber.KeyPress
   ' Prohibit non-numeric input in the text box for the number of read points.
   Dim intKey As Integer = Asc(e.KeyChar)

   If (intKey < 47) Or (intKey > 57) Then
      If (intKey <> 8) Then
         e.Handled = True
      End If
   End If
End Sub
```

A

```
    Private Sub sAddress_KeyPress(sender As Object, e As KeyPressEventArgs) Handles sAddress.KeyPress
        ' Prohibit non-numeric input in the text box for the start address (for decimal).
        ' Prohibit input of other than numerical values, a to f, and A to F in the text box for the start address (for hexadecimal).
        Dim intKey As Integer = Asc(e.KeyChar)

        If (rbHEX.Checked = True) Then
            If ((intKey < 47) Or (intKey > 57)) And ((intKey < 64) Or (intKey > 70))
              And ((intKey < 96) Or (intKey > 102)) Then
                If (intKey <> 8) Then
                    e.Handled = True
                End If
            End If
        Else
            If (intKey < 47) Or (intKey > 57) Then
                If (intKey <> 8) Then
                    e.Handled = True
                End If
            End If
        End If
    End Sub

    Private Sub lAddress_KeyPress(sender As Object, e As KeyPressEventArgs) Handles lAddress.KeyPress
        ' Prohibit non-numeric input in the text box for the end address (for decimal).
        ' Prohibit input of other than numerical values, a to f, and A to F in the text box for the end address (for hexadecimal).
        Dim intKey As Integer = Asc(e.KeyChar)

        If (rbHEX.Checked = True) Then
            If ((intKey < 47) Or (intKey > 57)) And ((intKey < 64) Or (intKey > 70))
              And ((intKey < 96) Or (intKey > 102)) Then
                If (intKey <> 8) Then
                    e.Handled = True
                End If
            End If
        Else
            If (intKey < 47) Or (intKey > 57) Then
                If (intKey <> 8) Then
                    e.Handled = True
                End If
            End If
        End If
    End Sub
End Class
```

Appendix 7 Ethernet-equipped Module Specifications

This section describes the I/O signals of the RJ71EN71 and the RnENCPU (network part) for the CPU module. The following tables show the I/O signal assignment of when the start I/O number of the RJ71EN71 or the RnENCPU (network part) is "0".

Appendix 7.1 List of I/O signals

The following tables list I/O signals. The device X is an input signal from the RJ71EN71 and the RnENCPU (network part) to the CPU module. The device Y is an output signal from the CPU module to the RJ71EN71 and the RnENCPU (network part). The I/O signals differ when the network type is Ethernet and Q-compatible Ethernet.

When network type is "Ethernet"

■Input signals

Device No.	Signal name
X0[*1]	Module failure (On: Module failure, Off: Module normal)
X1 to XE	Use prohibited
XF	Module ready (On: Available for module operation, Off: Not available for module operation)
X10 to X1F	Use prohibited

[*1] This signal is enabled when the network type for the P2 connector is set to "CC-Link IE Field". For details on operation of the signal, refer to the following.
📖 MELSEC iQ-R CC-Link IE Field Network User's Manual (Application)

■Output signals

Device No.	Signal name
Y0 to Y1F	Use prohibited

Point

- Do not use (turn on) any "use prohibited" signals as an input or output signal to the CPU module. Doing so may cause malfunction of the programmable controller system.
- For the I/O signals of the P2 connector when its network type is set to "CC-Link IE Control", refer to the following.

📖 MELSEC iQ-R CC-Link IE Controller Network User's Manual (Application)

- For the I/O signals of the P2 connector when its network type is set to "CC-Link IE Field", refer to the following.

📖 MELSEC iQ-R CC-Link IE Field Network User's Manual (Application)

When the network type is "Q Compatible Ethernet"

With Q-compatible Ethernet, the same I/O signals as the MELSEC-Q series Ethernet interface module can be used except in some cases.

The following table lists the differences of the RJ71EN71 and MELSEC-Q series Ethernet interface modules.

Device No.	RJ71EN71	QJ71E71-100
X1C	ERR LED lit confirmation[*1] (On: Lit, Off: —)	COM.ERR LED lit confirmation (On: Lit, Off: —)
Y17	ERR LED off request[*2] (On: At off request, Off: —)	COM.ERR LED off request (On: At off request, Off: —)
Y19	Use prohibited	Initial request signal (On: At request, Off: —)

[*1] This signal turns on when the ERR LED or the P ERR LED of P1 is on or flashing.

[*2] The details are listed below.

· The ERR LED and the P ERR LED of P1 turn off.

· An off request continues to be issued while this signal is on. (The LED does not turn on while the signal is on.)

· The event is registered in the event history during off processing.

· This signal cannot be turned off during major error.

· If the error occurs again after the device is turned on and off, the ERR LED or P ERR LED will turn on or flash according to the error details.

· A currently occurring error and error history information are not cleared.

For details on the other I/O signals, refer to the following.

📖 Q Corresponding Ethernet Interface Module User's Manual (Basic)

Point

Do not use (turn on) any "use prohibited" signals as an input or output signal to the CPU module. Doing so may cause malfunction of the programmable controller system.

Appendix 7.2 Buffer memory

The buffer memory is used for the following applications.

Module	Application
RJ71EN71, RnENCPU (network part)	Exchanges data with the CPU module.
CPU module (built-in Ethernet port part)	Stores data such as the setting values for the Ethernet function and monitor values, and data used for exchanging data with the multiple CPU function.

Buffer memory values are reset to default when the CPU module is reset or the system is powered off.

Appendix 7.2.1 List of buffer memory addresses

The following table lists the buffer memory addresses of the RJ71EN71 and the RnENCPU (network part) when the network type is set to "Ethernet".

P1 address (P2 address*1)		Application	Name
Decimal	Hexadecimal		
0 to 3 (2000000 to 2000003)	0H to 3H (1E8480H to 1E8483H)	System area	
4 to 5 (2000004 to 2000005)	4H to 5H (1E8484H to 1E8485H)	Own node setting status storage area	Own node IP address
6 to 13 (2000006 to 2000013)	6H to DH (1E8486H to 1E848DH)		System area
14 to 15 (2000014 to 2000015)	EH to FH (1E848EH to 1E848FH)		Subnet mask
16 to 17 (2000016 to 2000017)	10H to 11H (1E8490H to 1E8491H)		System area
18 to 19 (2000018 to 2000019)	12H to 13H (1E8492H to 1E8493H)		Default gateway IP address
20 to 27 (2000020 to 2000027)	14H to 1BH (1E8494H to 1E849BH)		System area
28 to 30 (2000028 to 2000030)	1CH to 1EH (1E849CH to 1E849EH)		Own node MAC address
31 (2000031)	1FH (1E849FH)		Own node network number
32 (2000032)	20H (1E84A0H)		Station number
33 (2000033)	21H (1E84A1H)		Transient transmission group number
34 (2000034)	22H (1E84A2H)		Send frame setting
35 (2000035)	23H (1E84A3H)		Jumbo frame setting
36 (2000036)	24H (1E84A4H)		Communication speed setting
37 (2000037)	25H (1E84A5H)		Auto-open UDP port number
38 (2000038)	26H (1E84A6H)		MELSOFT connection TCP port number
39 (2000039)	27H (1E84A7H)	Own node setting status storage area	MELSOFT connection UDP port number
40 (2000040)	28H (1E84A8H)		MELSOFT direct connection port number

P1 address (P2 address[*1])		Application	Name	
Decimal	Hexadecimal			
41 to 99 (2000041 to 2000099)	29H to 63H (1E84A9H to 1E84E3H)	System area		
100 to 163 (2000100 to 2000163)	64H to A3H (1E84E4H to 1E8523H)	Connection status storage area	Connection No.1 latest error code to Connection No.64 latest error code	
164 to 999 (2000164 to 2000999)	A4H to 3E7H (1E8524H to 1E8867H)	System area		
1000 (2001000)	3E8H (1E8868H)	System port latest error code storage area	FTP server latest error code	
1001 (2001001)	3E9H (1E8869H)		MELSOFT direct connection latest error code	
1002 to 4999 (2001002 to 2004999)	3EAH to 1387H (1E886AH to 1E9807H)	System area		
5000 to 5001 (2005000 to 2005001)	1388H to 1389H (1E9808H to 1E9809H)	Status for each protocol	IP packet	Total Number of Receives
5002 to 5003 (2005002 to 2005003)	138AH to 138BH (1E980AH to 1E980BH)			Total Number of Sum Check Error Cancels
5004 to 5005 (2005004 to 2005005)	138CH to 138DH (1E980CH to 1E980DH)			Total Number of Sends
5006 to 5021 (2005006 to 2005021)	138EH to 139DH (1E980EH to 1E981DH)			System area
5022 to 5023 (2005022 to 2005023)	139EH to 139FH (1E981EH to 1E981FH)			Simultaneous transmission error detection count (receive buffer full count)
5024 to 5039 (2005024 to 2005039)	13A0H to 13AFH (1E9820H to 1E982FH)		System area	
5040 to 5041 (2005040 to 2005041)	13B0H to 13B1H (1E9830H to 1E9831H)		ICMP packet	Total Number of Receives
5042 to 5043 (2005042 to 2005043)	13B2H to 13B3H (1E9832H to 1E9833H)			Total Number of Sum Check Error Cancels
5044 to 5045 (2005044 to 2005045)	13B4H to 13B5H (1E9834H to 1E9835H)			Total Number of Sends
5046 to 5047 (2005046 to 2005047)	13B6H to 13B7H (1E9836H to 1E9837H)			Total Number of Echo Request Receives
5048 to 5049 (2005048 to 2005049)	13B8H to 13B9H (1E9838H to 1E9839H)			Total Number of Echo Reply Sends
5050 to 5051 (2005050 to 2005051)	13BAH to 13BBH (1E983AH to 1E983BH)			Total Number of Echo Request Sends
5052 to 5053 (2005052 to 2005053)	13BCH to 13BDH (1E983CH to 1E983DH)			Total Number of Echo Reply Receives

P1 address (P2 address*1)		Application	Name	
Decimal	Hexadecimal			
5054 to 5079 (2005054 to 2005079)	13BEH to 13D7H (1E983EH to 1E9857H)	Status for each protocol	System area	
5080 to 5081 (2005080 to 2005081)	13D8H to 13D9H (1E9858H to 1E9859H)		TCP packet	Total Number of Receives
5082 to 5083 (2005082 to 2005083)	13DAH to 13DBH (1E985AH to 1E985BH)			Total Number of Sum Check Error Cancels
5084 to 5085 (2005084 to 2005085)	13DCH to 13DDH (1E985CH to 1E985DH)			Total Number of Sends
5086 to 5119 (2005086 to 2005119)	13DEH to 13FFH (1E985EH to 1E987FH)		System area	
5120 to 5121 (2005120 to 2005121)	1400H to 1401H (1E9880H to 1E9881H)		UDP packet	Total Number of Receives
5122 to 5123 (2005122 to 2005123)	1402H to 1403H (1E9882H to 1E9883H)			Total Number of Sum Check Error Cancels
5124 to 5125 (2005124 to 2005125)	1404H to 1405H (1E9884H to 1E9885H)			Total Number of Sends
5126 to 5159 (2005126 to 2005159)	1406H to 1427H (1E9886H to 1E98A7H)		System area	
5160 to 5161 (2005160 to 2005161)	1428H to 1429H (1E98A8H to 1E98A9H)		Receiving error	Framing error count
5162 to 5163 (2005162 to 2005163)	142AH to 142BH (1E98AAH to 1E98ABH)			Receive FIFO overflow count
5164 to 5165 (2005164 to 2005165)	142CH to 142DH (1E98ACH to 1E98ADH)			CRC error count
5166 to 5188 (2005166 to 2005188)	142EH to 1444H (1E98AEH to 1E98C4H)		System area	
5189 (2005189)	1445H (1E98C5H)	Own node operation status storage area	LED status	
5190 (2005190)	1446H (1E98C6H)		System area	
5191 (2005191)	1447H (1E98C7H)		Hub connection status area	Communication mode
5192 (2005192)	1448H (1E98C8H)			Connection status
5193 (2005193)	1449H (1E98C9H)			Communication speed
5194 (2005194)	144AH (1E98CAH)			Disconnection count
5195 to 5199 (2005195 to 2005199)	144BH to 144FH (1E98CBH to 1E98CFH)		System area	
5200 (2005200)	1450H (1E98D0H)		IP address duplication status storage area	Same IP address detection flag
5201 to 5203 (2005201 to 2005203)	1451H to 1453H (1E98D1H to 1E98D3H)			MAC address of the station already connected to the network
5204 to 5206 (2005204 to 2005206)	1454H to 1456H (1E98D4H to 1E98D6H)			MAC address of the station with the IP address already used

P1 address (P2 address[*1])		Application	Name	
Decimal	Hexadecimal			
5207 to 5300 (2005207 to 2005300)	1457H to 14B4H (1E98D7H to 1E9934H)	System area		
5301 (2005301)	14B5H (1E9935H)	Area for sending/ receiving instructions	RECV instruction execution request	
5302 to 5322 (2005302 to 2005322)	14B6H to 14CAH (1E9936H to 1E994AH)		System area	
5323 (2005323)	14CBH (1E994BH)		Link dedicated instruction	ZNRD instruction execution result
5324 (2005324)	14CCH (1E994CH)			System area
5325 (2005325)	14CDH (1E994DH)			ZNWR instruction execution result
5326 to 5625 (2005326 to 2005625)	14CEH to 15F9H (1E994EH to 1E9A79H)	System area		
5626 (2005626)	15FAH (1E9A7AH)	Remote password lock status storage area	Remote password lock status (connection No.1 to 16)	
5627 (2005627)	15FBH (1E9A7BH)		Remote password lock status (connection No.17 to 32)	
5628 (2005628)	15FCH (1E9A7CH)		Remote password lock status (connection No.33 to 48)	
5629 (2005629)	15FDH (1E9A7DH)		Remote password lock status (connection No.49 to 64)	
5630 (2005630)	15FEH (1E9A7EH)		Remote password lock status system port	
5631 to 5645 (2005631 to 2005645)	15FFH to 160DH (1E9A7FH to 1E9A8DH)	System area		
5646 (2005646)	160EH (1E9A8EH)	Forced connection invalidation setting area	Forced connection invalidation (connection No.1 to 16)	
5647 (2005647)	160FH (1E9A8FH)		Forced connection invalidation (connection No.17 to 32)	
5648 (2005648)	1610H (1E9A90H)		Forced connection invalidation (connection No.33 to 48)	
5649 (2005649)	1611H (1E9A91H)		Forced connection invalidation (connection No.49 to 64)	
5650 (2005650)	1612H (1E9A92H)		Forced connection invalidation system port	
5651 to 8299 (2005651 to 2008299)	1613H to 206BH (1E9A93H to 1EA4EBH)	System area		
8300 (2008300)	206CH (1EA4ECH)	Predefined protocol support function execution status check area	Connection No.1	Protocol execution status
8301 (2008301)	206DH (1EA4EDH)			System area
8302 to 8317 (2008302 to 2008317)	206EH to 207DH (1EA4EEH to 1EA4FDH)			Received data verification result (receive packet No.1 to 16)
8318 (2008318)	207EH (1EA4FEH)			Protocol execution count
8319(2008319)	207FH (1EA4FFH)			Protocol cancellation specification
8320 to 8335 (2008320 to 2008335)	2080H to 208FH (1EA500H to 1EA50FH)			System area
8336 to 8875 (2008336 to 2008875)	2090H to 22ABH (1EA510H to 1EA72BH)		Connection No.2 to Connection No.16	

P1 address (P2 address*1)		Application	Name	
Decimal	Hexadecimal			
8876 to 10999 (2008876 to 2010999)	22ACH to 2AF7H (1EA72CH to 1EAF77H)	System area		
11000 (2011000)	2AF8H (1EAF78H)	Time setting function (SNTP) area	Time setting function operation result	
11001 (2011001)	2AF9H (1EAF79H)		Time setting function execution time	(Year)
11002 (2011002)	2AFAH (1EAF7AH)			(Month)
11003 (2011003)	2AFBH (1EAF7BH)			(Day)
11004 (2011004)	2AFCH (1EAF7CH)			(Hour)
11005 (2011005)	2AFDH (1EAF7DH)			(Minute)
11006 (2011006)	2AFEH (1EAF7EH)			(Second)
11007 (2011007)	2AFFH (1EAF7FH)			(Day of the week)
11008 (2011008)	2B00H (1EAF80H)		Time setting function required response time	
11009 (2011009)	2B01H (1EAF81H)		Time setting function (SNTP client) execution	
11010 to 11049 (2011010 to 2011049)	2B02H to 2B29H (1EAF82H to 1EAFA9H)	System area		
11050 to 11051 (2011050 to 2011051)	2B2AH to 2B2BH (1EAFAAH to 1EAFABH)	IP packet transfer function area	IP packet transfer latest data volume	
11052 to 11053 (2011052 to 2011053)	2B2CH to 2B2DH (1EAFACH to 1EAFADH)		IP packet transfer maximum data volume	
11054 to 11499 (2011054 to 2011499)	2B2EH to 2CEBH (1EAFAEH to 1EB16BH)	System area		
11500 to 11514 (2011500 to 2011514)	2CECH to 2CFAH (1EB16CH to 1EB17AH)	Dynamic routing function area	Communication path determination status	
11515 to 11599 (2011515 to 2011599)	2CFBH to 2D4FH (1EB17BH to 1EB1CFH)	System area		
11600 to 11663 (2011600 to 2011663)	2D50H to 2D8FH (1EB1D0H to 1EB20FH)	Remote password function monitoring area	Connection No.1 continuous unlock failure count to Connection No.64 continuous unlock failure count	
11664 (2011664)	2D90H (1EB210H)		Auto-open UDP port continuous unlock failure count	
11665 (2011665)	2D91H (1EB211H)		MELSOFT transmission port (UDP/IP) continuous unlock failure count	
11666 (2011666)	2D92H (1EB212H)		MELSOFT transmission port (TCP/IP) continuous unlock failure count	
11667 (2011667)	2D93H (1EB213H)		FTP transmission port (TCP/IP) continuous unlock failure count	
11668 (2011668)	2D94H (1EB214H)		MELSOFT direct connection continuous unlock failure count	
11669 (2011669)	2D95H (1EB215H)	System area		
11670 to 11671 (2011670 to 2011671)	2D96H to 2D97H (1EB216H to 1EB217H)	Area for both systems identical IP address setting function	Control system IP address	

P1 address (P2 address*1)		Application	Name	
Decimal	Hexadecimal			
11672 to 11699 (2011672 to 2011699)	2D98H to 2DB3H (1EB218H to 1EB233H)	System area		
11700 to 11701 (2011700 to 2011701)	2DB4H to 2DB5H (1EB234H to 1EB235H)	IP address change function area	IP Address	IP address
11702 to 11703 (2011702 to 2011703)	2DB6H to 2DB7H (1EB236H to 1EB237H)			Subnet mask
11704 to 11705 (2011704 to 2011705)	2DB8H to 2DB9H (1EB238H to 1EB239H)			Default gateway
11706 (2011706)	2DBAH (1EB23AH)		Communications by Network No./Station No.	Network number
11707 (2011707)	2DBBH (1EB23BH)			Station number
11708 (2011708)	2DBCH (1EB23CH)			Transient transmission group number
11709 (2011709)	2DBDH (1EB23DH)		IP address storage area write request	
11710 (2011710)	2DBEH (1EB23EH)		IP address storage area write error	
11711 (2011711)	2DBFH (1EB23FH)		IP address storage area writing error cause	
11712 (2011712)	2DC0H (1EB240H)		IP address storage area clear request	
11713 (2011713)	2DC1H (1EB241H)		IP address storage area clear error	
11714 (2011714)	2DC2H (1EB242H)		IP address storage area clear error cause	
11715 (2011715)	2DC3H (1EB243H)		IP address change function operating status	
11716 to 19999 (2011716 to 2019999)	2DC4H to 4E1FH (1EB244H to 1ED29FH)	System area		
20000 to 26143 (2020000 to 2026143)	4E20H to 661FH (1ED2A0H to 1EEA9FH)	Random access buffer area	Random access buffer	
26144 to 65534 (2026144 to 2065534)	6620H to FFFEH (1EEAA0H to 1F847EH)	System area		
65535 (2065535)	FFFFH (1F847FH)	Network type information area	Network type information	
65536	10000H	Area for communication using a fixed buffer	Fixed buffer No.1	Data length
65537 to 70655	10001H to 113FFH			Fixed buffer data
70656 to 147455	11400H to 23FFFH		Fixed buffer No.2 to Fixed buffer No.16	(The bit configuration is the same as Fixed buffer No.1)
147456 to 1899999 (2147456 to 3899999)	24000H to 1CFDDFH (20C480H to 3B825FH)	System area		

*1 The P2 address cannot be used for the RnENCPU (network part).

■P1/P2 common area

Address		Application	Name	
Decimal	Hexadecimal			
1900000 to 1900007	1CFDE0H to 1CFDE7H	Ethernet PORT1/2 common information	Open completion signal	
1900008 to 1900015	1CFDE8H to 1CFDEFH		Open request signal	
1900016 to 1900023	1CFDF0H to 1CFDF7H		Socket/fixed buffer reception status signal	
1900024	1CFDF8H		Initial status	
1900025	1CFDF9H		Initial error code	
1900026 to 1900029	1CFDFAH to 1CFDFDH	System area		
1900030[*1]	1CFDFEH	Receive buffer status storage area	State of receive buffer	
1900031 to 1901001	1CFDFFH to 1D01C9H	System area		
1901002	1D01CAH	Predefined protocol support function check area	Predefined protocol ready	
1901003 to 1901019	1D01CBH to 1D01DBH	System area		
1901020	1D01DCH	Predefined protocol setting data check area	Predefined protocol setting data error information	Protocol number
1901021	1D01DDH			Setting type
1901022	1D01DEH			Packet number
1901023	1D01DFH			Element number
1901024	1D01E0H		Number of registered predefined protocols	
1901025 to 1901031	1D01E1H to 1D01E7H		System area	
1901032 to 1901047	1D01E8H to 1D01F7H		Predefined protocol registration	
1901048 to 1901999	1D01F8H to 1D05AFH	System area		
1902000 to 1904047	1D05B0H to 1D0DAFH	Send/receive area for predefined protocol support function	Send/receive area for predefined protocol support function	
1904048 to 1999999	1D0DB0H to 1E847FH	System area		

*1 Availability for the buffer memory depends on the firmware version.

Point

- Do not write any data to "System area". Doing so may cause malfunction of the programmable controller system.
- If the value in an area of one word in size becomes equal to or higher than 65536, the count stops at 65535 (FFFFH).

Appendix 7.2.2 Details of buffer memory areas

This section describes details of the buffer memory areas of the Ethernet-equipped module.

The buffer memory addresses used are for the P1 connector of the RJ71EN71.

Ethernet PORT1/2 common information

■Open completion signal (Un\G1900000 to Un\G1900007)

The open status of each connection is stored.

- 0: Closed or not open
- 1: Open completed

Address	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Un\G1900000	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
⋮																
Un\G1900007	128	127	126	125	124	123	122	121	120	119	118	117	116	115	114	113

The numbers in the table indicate connection numbers.

■Open request signal (Un\G1900008 to \G1900015)

The open processing status of each connection is stored.

- 0: No open request
- 1: Requesting open

Address	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Un\G1900008	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
⋮																
Un\G1900015	128	127	126	125	124	123	122	121	120	119	118	117	116	115	114	113

The numbers in the table indicate connection numbers.

■Socket/fixed buffer reception status signal (Un\G1900016 to Un\G1900023)

The reception status of each connection is stored.

- 0: Data not received
- 1: Data reception completed

Address	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
Un\G1900016	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1
⋮																
Un\G1900023	128	127	126	125	124	123	122	121	120	119	118	117	116	115	114	113

The numbers in the table indicate connection numbers.

■Initial status (Un\G1900024)

Address	Description
Un\G1900024	Stores the status of the initial processing of the RJ71EN71 or the RnENCPU (network part). Initial normal completion status (b0) 0: — 1: Initialization normal completion Initial abnormal completion status (b1) 0: — 1: Initialization abnormal completion b2 to b15: Use prohibited

■Initial error code (Un\G1900025)

Address	Description
Un\G1900025	Stores the information when the initial processing of the RJ71EN71 or the RnENCPU (network part) has completed abnormally. 0: In initial processing or initial normal completion Other than 0: Initial processing error code (An error code is stored.)

Appendix 7.3 Data format (Communications using the fixed buffer)

This section describes the data format used during communications using the fixed buffer.
The communication data is configured of the header and application data.

Header

The header is for TCP/IP or UDP/IP. The RJ71EN71 or the RnENCPU (network part) automatically adds and deletes the header, the setting is not required.
The contents of the header are shown below.

- TCP/IP

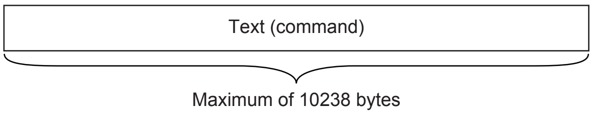
Ethernet (14 bytes)	IP (20 bytes)	TCP (20 bytes)

- UDP/IP

Ethernet (14 bytes)	IP (20 bytes)	UDP (8 bytes)

Application data

If the communication method is "Fixed Buffer (No Procedure)", the application data expresses the following data code with binary codes. Data is exchanged with binary codes regardless of the communication data code setting.



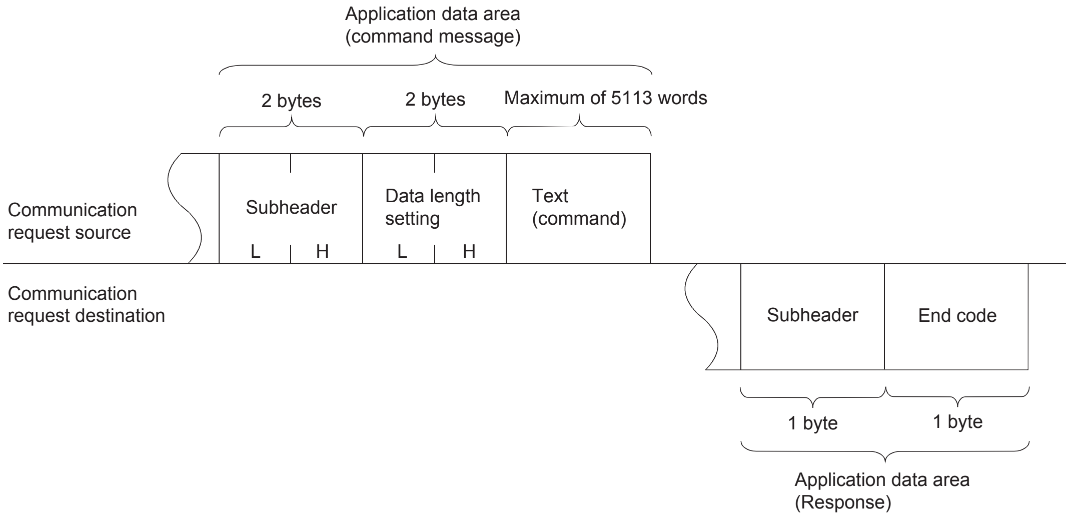
Point

With nonprocedural, the subheader and data length added for procedural are not used, so the data is all handled as valid text. The RJ71EN71 and the RnENCPU (network part) turn on the fixed buffer reception status signal after storing the size of the received message (packet) in the receive data length storage area. Providing a check procedure including the data length or data type code in the message's application data is recommended so that the application data's byte size and data type can be seen on the receiving side.

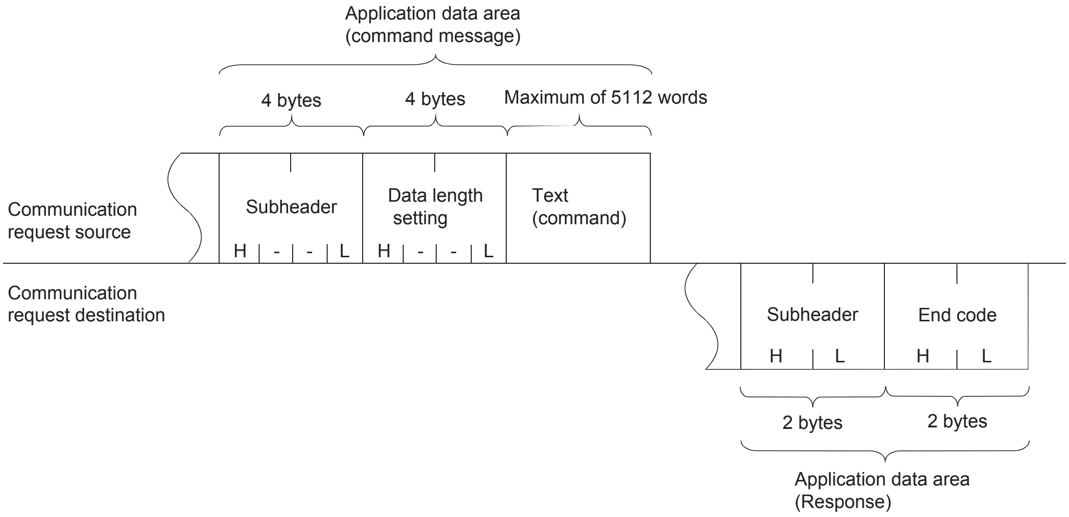
The following figures show the configuration of the application data when the communication method is set to "Fixed Buffer (Procedure Exist)".

■Format

- When exchanging data with binary codes

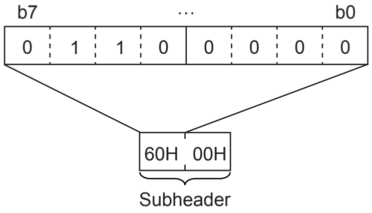
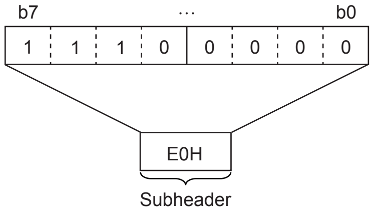
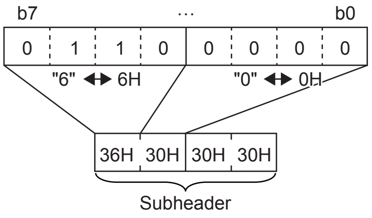
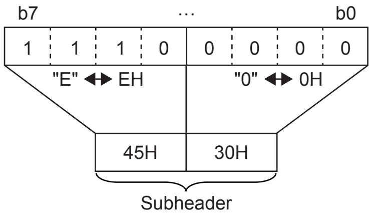

Application data area (command message)
2 bytes
2 bytes
Maximum of 5113 words
Communication request source
Subheader
L
H
Data length setting
L
H
Text (command)
Communication request destination
Subheader
End code
1 byte
1 byte
Application data area (Response)

- When exchanging data with ASCII codes


Application data area (command message)
4 bytes
4 bytes
Maximum of 5112 words
Communication request source
Subheader
H - - L
Data length setting
H - - L
Text (command)
Communication request destination
Subheader
H L
End code
H L
2 bytes
2 bytes
Application data area (Response)

■Subheader

The RJ71EN71 or the RnENCPU (network part) automatically adds and deletes the subheader, the setting is not required.

Data format	Command (external device → RJ71EN71 or RnENCPU (network part))	Response (RJ71EN71 or RnENCPU (network part) → external device)
Binary code		
ASCII code	 b7 … b0 0 1 1 0 0 0 0 0 "6" ◀▶ 6H "0" ◀▶ 0H 36H 30H 30H 30H Subheader	 b7 … b0 1 1 1 0 0 0 0 0 "E" ◀▶ EH "0" ◀▶ 0H 45H 30H Subheader

■Data length setting

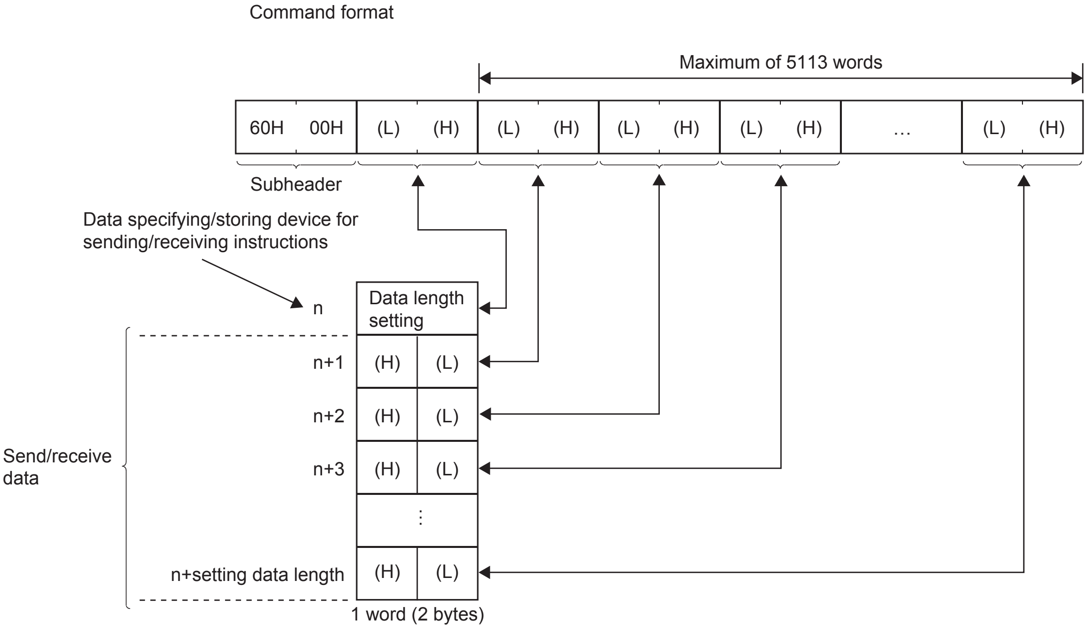
This setting shows the amount of data in the text (command) section.

- When exchanging data with binary codes: Maximum 5113 words
- When exchanging data with ASCII codes: Maximum 2556 words

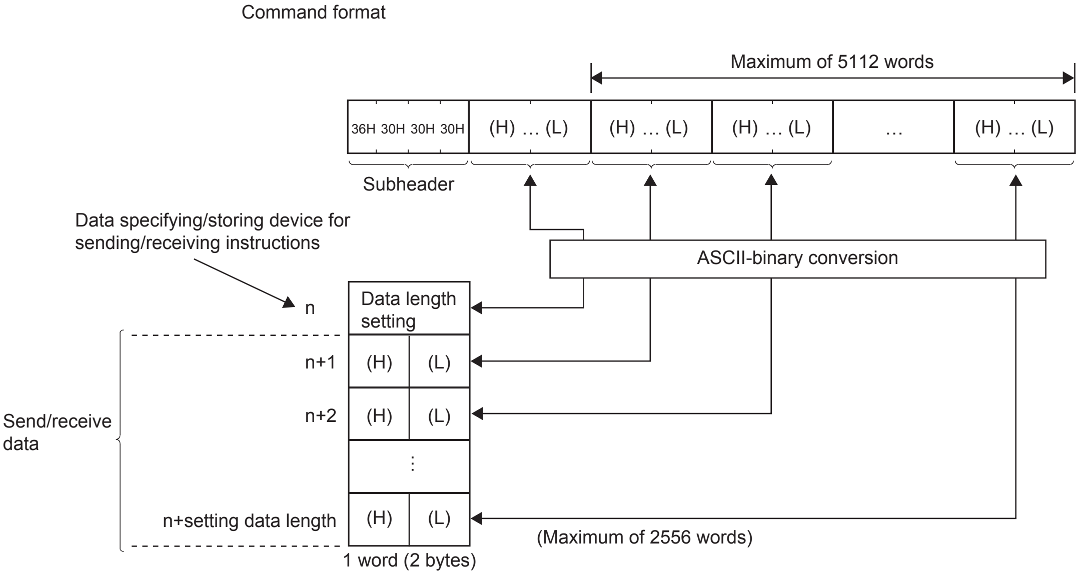
■Text (command)

The following figures show the format of the command/response.

- When exchanging data with binary codes



- When exchanging data with ASCII codes



■End code

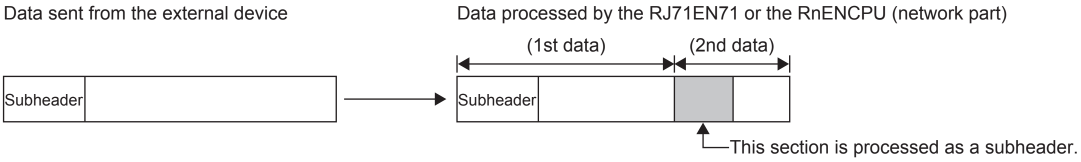
The error code is stored in the end command added to the response.

The end code is also stored in the BUFSND instruction and BUFRCV instruction completion status area (inside control data).

The following cases may apply if an error code for communications using the SLMP or random access buffer is stored even when communications using the fixed buffer is executed.

Description	Action
The data length specified in the application data section of the message sent from the external device to the RJ71EN71 or the RnENCPU (network part) differs from the actual data size in the text section.	Specify the actual data size in the text section as the data length in the application data section. (Refer to the following descriptions.)
The subheader of the message sent from the external device to the RJ71EN71 or the RnENCPU (network part) is incorrect.	Review the subheader specified in the application data section.

The communication data may be split and exchanged due to buffer limitations to the own station or external station. The data that is split and received is restored (reassembled) by the RJ71EN71 or the RnENCPU (network part) and exchanged. (The received data is restored (reassembled) based on the data length in the communication data.) The following table shows the processing of the RJ71EN71 and the RnENCPU (network part) for when the data in the communication data is incorrect.

Communication method	Description
Fixed Buffer (Procedure Exist), Random Access Buffer	When data length specified immediately after subheader < text data volume • The data immediately after the text corresponding to the data length specified immediately after the subheader is handled as the second message. • The start of each statement becomes the subheader, so the RJ71EN71 and the RnENCPU (network part) perform processing according to the subheader code. • If the subheader is not a code supported by the RJ71EN71 and the RnENCPU (network part), an abnormal completion response is sent to the external device.  Data sent from the external device Data processed by the RJ71EN71 or the RnENCPU (network part) (1st data) (2nd data) Subheader Subheader This section is processed as a subheader. In the above case, the code processed as the subheader with the uppermost bit set as 1 is returned as the response. For example, if the command's subheader section is 65H, the response's subheader is E5H.
	When data length specified immediately after subheader > text data volume The RJ71EN71 and the RnENCPU (network part) wait to receive the insufficient remaining data. If the remaining data is received within the response monitor timer value, the RJ71EN71 and the RnENCPU (network part) perform processing according to the subheader code. If the remaining data is not received within the response monitor timer value, the RJ71EN71 and the RnENCPU (network part) perform the following processing. • The ABORT(RST) instruction is sent to the external device, and the line is closed. • The error code is stored in 'Connection status storage area' (Un\G100 to Un\G163).
Fixed Buffer (No Procedure)	With nonprocedural, there is no message data length, so the received data is stored as is into the receive buffer area. Providing a check procedure including the data length or data type code in the message's application data is recommended so that the application data's byte size and data type can be seen on the receiving side.

Appendix 7.4 List of Error Codes

This section lists the error codes, error details and causes, and action for the errors occur in the processing for data communication between the Ethernet-equipped module and slave stations or caused by processing requests from the CPU module on the own station.

Error codes are classified into major error, moderate error, and minor error, and can be checked in the [Error Information] tab in the "Module Diagnostics" window of an Ethernet-equipped module.

For details, refer to the following.

📖MELSEC iQ-R Ethernet User's Manual (Application)

Error code	Error details and causes	Action	Detailed information
1080H	The number of writes to the flash ROM has exceeded 100000.	Replace the module.	—
1120H	Clock setting has failed when the system is powered on or the CPU module is reset.	• Check if the time settings are correctly set in parameter. • Check if the specified SNTP server is operating normally and there is no failure on the network accessing to the SNTP server computer.	—
1124H	• The default gateway is not set correctly. • The gateway IP address is not set correctly. • The default gateway/gateway IP address (network address after the subnet mask) is different from that of the IP address of the own node.	• Correct the default gateway IP address. • Set the same network address as that of the IP address.	Parameter information • Parameter type • I/O No. • Parameter No. • Network No. • Station No.
1128H	The port number is incorrect.	Correct the port number.	—
1129H	The port number of the external device is not set correctly.	Correct the port number of the external device.	—
112DH	The data was sent to the external device while the IP address setting of the device set in "External Device Configuration" under "Basic Settings" was incorrect.	• Correct the IP address of the external device in "External Device Configuration" under "Basic Settings". • Check that the IP address class of the external device is set to A/B/C in "External Device Configuration" under "Basic Settings".	—
112EH	A connection could not be established in the open processing.	• Check the operation of the external device. • Check if the open processing has been performed in the external device. • Correct the port number of the Ethernet-equipped module, IP address/port number of the external device, and opening method. • When the firewall is set in the external device, check if the access is permitted. • Check if the Ethernet cable is disconnected.	—
1133H	The response send failed during socket communications or communications using the fixed buffer.	• Check the operation of the external device or switching hub. • Since there may be congestion of packets on the line, send data after a certain period of time. • Check if the connection cable is disconnected. • Check that there is no connection failure with the switching hub. • Execute the communication status test, and if the test completed with an error, take the corrective action. • Execute the module communication test, and check that there is no failure in the module.	—
1134H	A TCP ULP timeout error has occurred in the TCP/IP communication. (The external device does not send an ACK response.)	• Check the operation of the external device. • Correct the TCP ULP timeout value of the Ethernet-equipped module. • Since there may be congestion of packets on the line, send data after a certain period of time. • Check if the connection cable is disconnected.	—
1152H	• The IP address is not set correctly. • The same IP address has been set to port 1 and port 2 of the Ethernet-equipped module.	• Correct the IP addresses. • Set different IP addresses for port 1 and port 2.	Parameter information • Parameter type • I/O No. • Parameter No. • Network No. • Station No.

Error code	Error details and causes	Action	Detailed information
1155H	• The specified connection was already closed in TCP/IP communications. • Open processing is not performed.	• Perform the open processing for the specified connection. • Check if the open processing has been performed in the external device.	—
1157H	• The specified connection was already closed in UDP/IP communications. • Open processing is not performed.	• Perform the open processing for the specified connection. • Check if the open processing has been performed in the external device.	—
1158H	• The receive buffer or send buffer is not sufficient. • The window size of the external device is not sufficient.	• Check the operation of the external device or switching hub. • When communications using a fixed buffer or socket communications is used, increase the execution frequency of the BUFRCV/SOCRCV instruction. • When the value of 'State of receive buffer' (Un\G1900030) is 0001H, reduce the frequency of data receive from the external device.	—
1165H	Data was not sent correctly with UDP/IP.	• Check the settings for connection with the external device. • Check the operation of the external device or switching hub. • Since there may be congestion of packets on the line, send data after a certain period of time. • Check if the connection cable is disconnected. • Check that there is no connection failure with the switching hub. • Execute the PING test and communication status test, and if the test completed with an error, take the corrective action.	—
1166H	Data was not sent correctly with TCP/IP.	• Check the settings for connection with the external device. • Check the operation of the external device or switching hub. • Since there may be congestion of packets on the line, send data after a certain period of time. • Check if the connection cable is disconnected. • Check that there is no connection failure with the switching hub. • Execute the PING test and communication status test, and if the test completed with an error, take the corrective action.	—
1167H	Unsent data found, but could not be sent.	• Check the settings for connection with the external device. • Check the operation of the external device or switching hub. • Since there may be congestion of packets on the line, send data after a certain period of time. • Check if the connection cable is disconnected. • Check that there is no connection failure with the switching hub. • Execute the PING test and communication status test, and if the test completed with an error, take the corrective action.	—
1180H	• The same IP address has been set as the system A IP address, system B IP address, and/or control system IP address. • Network addresses of the system A IP address, system B IP address, and control system IP address are different.	• Set different IP addresses for the system A IP address, system B IP address, and control system IP address. • Set the same network address for the system A IP address, system B IP address, and control system IP address.	Parameter information • Parameter type • I/O No. • Parameter No. • Network No. • Station No.
1801H	IP address of the external device could not be acquired.	• Correct the IP address in "Network/Station No. <-> IP information setting" under "Application Settings". • Check if the network or station number of the external device is correctly specified by using control data of the dedicated instruction.	Parameter information • Parameter type • I/O No. • Parameter No. • Network No. • Station No.
1811H	An error was detected in the CPU module.	Check the error of the CPU module and take action using the module diagnostics of the engineering tool.	—

Error code	Error details and causes	Action	Detailed information
1830H	Number of reception requests of transient transmission (link dedicated instruction) exceeded upper limit of simultaneously processable requests.	Lower the transient transmission usage frequency, and then perform again.	—
1845H	Too many processing of transient transmission (link dedicated instruction) and cannot perform transient transmission.	Correct the transient transmission execution count.	—
20E0H	The module cannot communicate with the CPU module.	The hardware failure of the CPU module may have been occurred. Please consult your local Mitsubishi representative.	—
2160H	Overlapping IP addresses were detected.	Check and correct the IP addresses.	—
2220H	The parameter setting is corrupted.	Check the detailed information of the error by executing module diagnostics using the engineering tool, and write the displayed parameter. If the error occurs again even after taking the above, the possible cause is a hardware failure of the module. Please consult your local Mitsubishi representative.	Parameter information • Parameter type
2221H	The set value is out of the range.	Check the detailed information of the error by executing module diagnostics using the engineering tool, and correct the parameter setting corresponding to the displayed number.	Parameter information • Parameter type • I/O No. • Parameter No. • Network No. • Station No.
2250H	The protocol setting data stored in the CPU module is not for the Ethernet-equipped module.	Write the protocol setting data for the Ethernet-equipped module to the CPU module.	Parameter information • Parameter type
24C0H to 24C3H	An error was detected on the system bus.	• Take measures to reduce noise. • Reset the CPU module, and run it again. If the error occurs again even after taking the above, the possible cause is a hardware failure of the module, base unit, or extension cable. Please consult your local Mitsubishi representative.	System configuration information • I/O No. • Base No. • Slot No. • CPU No.
24C6H	An error was detected on the system bus.	• Take measures to reduce noise. • Reset the CPU module, and run it again. If the error occurs again even after taking the above, the possible cause is a hardware failure of the module, base unit, or extension cable. Please consult your local Mitsubishi representative.	—
3020H	A value of the port number is out of range.	• Check the each system port number registered in the buffer memory. • If the error occurs again even after taking the above, the possible cause is a hardware failure of the error module. Please consult your local Mitsubishi representative.	—
3040H	Response data of the dedicated instruction cannot be created.	• Increase the request interval. • Decrease the number of request nodes. • Wait for a response to the previous request before sending the next request. • Correct the timeout value.	—
3060H	The send/receive data size exceeds the allowable range.	• Check and change the send data size of the Ethernet-equipped module or the external device. • If the same error code is displayed again, the possible cause is a hardware failure of the error module or CPU module. Please consult your local Mitsubishi representative.	—
3C00H to 3C03H	A hardware failure has been detected.	• Take measures to reduce noise. • Reset the CPU module, and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the error module. Please consult your local Mitsubishi representative.	—
3C0FH to 3C11H	A hardware failure has been detected.	• Take measures to reduce noise. • Reset the CPU module, and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the error module. Please consult your local Mitsubishi representative.	—

Error code	Error details and causes	Action	Detailed information
3C13H	A hardware failure has been detected.	Reset the CPU module, and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the error module. Please consult your local Mitsubishi representative.	—
3C14H	A hardware failure has been detected.	Reset the CPU module, and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the error module or CPU module. Please consult your local Mitsubishi representative.	—
3C2FH	An error was detected in the memory.	Reset the CPU module, and run it again. If the same error code is displayed again, the possible cause is a hardware failure of the error module. Please consult your local Mitsubishi representative.	—
3E01H	Network type of the own station is unexpected setting.	Rewrite the module parameter using the engineering tool. If the same error code is displayed again, the possible cause is a hardware failure of the error module. Please consult your local Mitsubishi representative.	—
4000H to 4FFFH	Errors detected by the CPU module (📖 MELSEC iQ-R CPU Module User's Manual (Application))		
C001H	• The IP address setting value of the E71 for the initial processing is incorrect. • The setting value of the subnet mask field for the router relay function is incorrect.	• Correct the IP address setting value for the initial processing. • Check if the class of the IP address is set to A/B/C. • Correct the subnet mask setting value for the initial processing.	—
C012H	The port number used in a connection already opened is set. (For TCP/IP)	Correct the port numbers of the Ethernet-equipped module and the external device.	—
C013H	The port number used in a connection already opened is set. (For UDP/IP)	Correct the port numbers of the Ethernet-equipped module and the external device.	—
C015H	• The specified IP address of the external device for the open processing is incorrect. • The specified IP address of the external device of the dedicated instruction is incorrect.	• Correct the specified IP address of the external device of the CONOPEN/OPEN instruction. Set the class to A/B/C. • Execute the dedicated instruction again after correcting the specified IP address of the external device.	—
C016H	The open processing of the connection specified for pairing open has already completed.	• Check that none of the connections targeted for pairing open has been opened. • Correct the combination of modules set for pairing open.	—
C018H	The specified IP address of the external device is incorrect.	Correct the specified IP address of the external device.	—
C020H	The send/receive data length exceeds the allowable range.	• Correct the data length to be sent. • When the amount of data to be sent exceeds the limit, divide the data into smaller chunks to send it.	—
C021H	An abnormal end response was received for communications using the fixed buffer and random access buffer.	• Check the command/response type of the subheader. • Check the data length setting to be sent. • Check if the communication data code setting of the Ethernet-equipped module meets the binary/ASCII of the message to be sent.	—
C022H	• A response could not be received within the response monitoring timer value. • The connection with the external device was closed while waiting for a response.	• Check the operation of the external device. • Correct the response monitoring timer value of the Ethernet-equipped module. • Check the open status of the connection with the external device.	—
C024H	• Communications using the fixed buffer or communications using a random access buffer were executed when communication method is set to the "Predefined Protocol" connection. • Predefined protocol was executed when communication method is set to "Fixed Buffer (Procedure Exist)" or "Fixed Buffer (No Procedure)" connection.	• Check that there is no error in the connection number specification of the dedicated instruction. • Correct the communication method of the connection with the external device.	—

Error code	Error details and causes	Action	Detailed information
C025H	There is an error in the usage setting area when starting the open processing by the CONOPEN/OPEN instruction or I/O signals.	• When starting the open by using the CONOPEN/OPEN instruction, correct the usage setting area of the control data. • When starting the open by I/O signals, correct the connection usage setting area of the buffer memory.	—
C026H	An error has occurred when reading/writing/verifying the predefined protocol setting data.	• Check that connection cable with the engineering tool is not disconnected and read/write/verify the predefined protocol setting data again. • Do not write data simultaneously when writing protocol setting data from multiple engineering tools.	—
C027H	Message send of the socket communications has failed.	• Check the operation of the external device or switching hub. • Since there may be congestion of packets on the line, send data after a certain period of time. • Check if the connection cable is disconnected. • Check that there is no connection failure with the switching hub. • Execute the communication status test, and if the test completed with an error, take the corrective action. • Execute the module communication test, and check that there is no failure in the module. • Check the IP address specified as the destination.	—
C028H	Message send of the fixed buffer has failed.	• Check the operation of the external device or switching hub. • Since there may be congestion of packets on the line, send data after a certain period of time. • Check if the connection cable is disconnected. • Check that there is no connection failure with the switching hub. • Execute the communication status test, and if the test completed with an error, take the corrective action. • Execute the module communication test, and check that there is no failure in the module. • Check the IP address specified as the destination.	—
C029H	• Description of control data is not correct. • Open instruction was executed through open settings parameter even though parameters are not set.	• Correct the descriptions of the control data. • Set the open settings parameters. Execute the OPEN instruction through control data.	—
C035H	The existence of the external device could not be checked within the response monitoring timer value.	• Check the operation of the external device. • Correct the settings in "Timer Settings for Data Communication" under "Application Settings" of the Ethernet-equipped module. • Check if the connection cable is disconnected.	—
C040H	• Not all the data could be received within the response monitoring timer value. • Sufficient data for the data length could not be received. • The remaining part of the message divided at the TCP/IP level could not be received within the response monitoring timer value.	• Correct the data length of the communication data. • The packets may be crowded in the line, so send the data again from the external device after a random time has passed.	—
C050H	When "ASCII" has been selected in the communication data code setting of the Ethernet-equipped module, ASCII code data which cannot be converted into binary code data has been received.	Check if ASCII code data which cannot be converted into binary code data has been sent from the external device.	—
C051H	The number of read/write points from/to the device of SLMP message is out of the allowable range in the CPU module (in units of words).	Correct the number of read/write points and send the SLMP message to the Ethernet-equipped module again.	—
C052H	The number of read/write points from/to the device of SLMP message is out of the allowable range in the CPU module (in units of bits).	Correct the number of read/write points and send the SLMP message to the Ethernet-equipped module again.	—
C053H	The number of read/write points from/to the random device of SLMP message is out of the allowable range in the CPU module (in units of bits).	Correct the number of read/write points and send the SLMP message to the Ethernet-equipped module again.	—
C054H	The number of read/write points from/to the random device of SLMP message is out of the allowable range in the CPU module (in units of words, double words).	Correct the number of read/write points and send the SLMP message to the Ethernet-equipped module again.	—

Error code	Error details and causes	Action	Detailed information
C055H	The read/write size from/to the file data of SLMP message is out of the allowable range.	Correct the read/write size and send the SLMP message to the Ethernet-equipped module again.	—
C056H	The read/write request exceeds the largest address.	Correct the start address or the number of read/write points so that the request does not exceed the largest address and send the data to the Ethernet-equipped module again.	—
C057H	The request data length of the SLMP message does not match with the number of data in the character (a part of text).	Check and correct the text or request data length, and send the SLMP message to the Ethernet-equipped module again.	—
C058H	The request data length of the SLMP message after the ASCII/binary conversion does not match with the number of data in the character (a part of text).	Check and correct the text or request data length, and send the SLMP message to the Ethernet-equipped module again.	—
C059H	• The specified command and subcommand of the SLMP message are incorrect • The function which is not supported by the target device was executed.	• Check that there is no error in the specification of the command and subcommand of the SLMP message. • Check whether the function executed is supported by the target device. • Check the version of the target device.	—
C05AH	The Ethernet-equipped module cannot read/write data from/to the device specified by the SLMP message.	Correct the specification of the device to be read/written and send the SLMP message to the Ethernet-equipped module again.	—
C05BH	The Ethernet-equipped module cannot read/write data from/to the device specified by the SLMP message.	Correct the specification of the device to be read/written and send the SLMP message to the Ethernet-equipped module again.	—
C05CH	• The received request data of the SLMP message is incorrect. • The setting value of the communication setting when the iQSS function is executed is out of range. • When the iQSS function is executed, the items of communication setting which cannot be set on the target device are set. • When the iQSS function is executed, the required setting items have not been set to the target device.	• Correct the request data and send the SLMP message to the Ethernet-equipped module again. • Correct the setting details of when the iQSS function is executed, and retry the operation.	—
C05DH	The "Monitor Request" command is received before the monitor registration is performed by "Monitor Registration/Clear" command of the SLMP message.	Register the monitoring data using "Monitor Registration/ Clear" command and perform monitoring.	—
C05EH	The time between received the SLMP message from the Ethernet-equipped module and returned response from the access destination exceeded the monitoring timer value set in the SLMP command.	• Increase the monitoring timer value. • Check if the access destination is operating normally. • Correct the network number or request destination station number. • When the access destination is a module with a different network number, check if "Routing Setting" of "CPU Parameter" is correctly set. • If the access destination is a module with a different network number, check if the network number is not in use.	—
C05FH	This request cannot be executed to the access destination specified by the SLMP message.	Correct the access destination.	—
C060H	The request details for bit devices of the SLMP message is incorrect.	Correct the request details and send the SLMP message to the Ethernet-equipped module again.	—
C061H	• The request data length of the SLMP message does not match with the number of data in the character (a part of text). • The write data length specified by the label write command is not even byte. • When the iQSS function is executed, incorrect frame is received.	• Check and correct the text or request data length, and send the SLMP message to the Ethernet-equipped module again. • Add the dummy data for one byte, and specify the length by even byte. • Check the operating status and connection status of the target device of when the iQSS function is executed. • Check the connection of an Ethernet cable and a hub of when the iQSS function is executed. • Check the line status of Ethernet of when the iQSS function is executed. • Reset the CPU module and device to be targets of the iQSS function, and retry the operation. For the error occurred when the iQSS function is executed, contact the manufacturer of the target device if the above actions do not solve the problem.	—

Error code	Error details and causes	Action	Detailed information
C070H	The device memory cannot be extended for the access destination specified by the SLMP message.	• Correct the SLMP message to read/write data without the device memory set for extension. • Specify the extension of the device memory only for an Ethernet-equipped module mounted station and a R/Q/QnACPU over CC-Link IE Controller Network, MELSECNET/H, or MELSECNET/10.	—
C071H	The number of device points for data read/write set for modules other than a R/Q/QnACPU is out of the range.	Correct the number of read/write points and send the SLMP message to the Ethernet-equipped module again.	—
C072H	The request details of the SLMP message is incorrect. (For example, a request for data read/write in bit units has been issued to a word device.)	• Check if the data can be requested to the access destination. • Correct the request details and send the SLMP message to the Ethernet-equipped module again.	—
C073H	The access destination of the SLMP message cannot issue this request. (For example, the number of double word access points cannot be specified for modules other than a R/Q/QnACPU.)	Correct the request details of the SLMP message.	—
C075H	The request data length for the label access is out of range.	• Correct the number of read/write points and send the SLMP message to the Ethernet-equipped module again. • Correct the label to shorten the label name and send the SLMP message to the Ethernet-equipped module again.	—
C081H	The termination processing for the Ethernet-equipped module that is involved with the reinitialization processing is being performed, and arrival of link dedicated instructions cannot be checked.	Finish all the communications to perform the reinitialization processing of the Ethernet-equipped module.	—
C083H	The communication processing was abnormally ended in the link dedicated instruction communications	• Check if the relay station and external station are operating normally. • Check if there is an error with the cable connection between the own station and the external station. • If the line is heavily loaded, reduce the load, and retry the operation. • Increase the number of resends for the link dedicated instructions and execute the operation.	—
C084H	The communication processing was abnormally ended in the link dedicated instruction communications	• Check if the own station/relay station/external station is operating normally. • Check if there is an error with the cable connection between the own station and the external station. • Increase the TCP resend timer value of the Ethernet-equipped module.	—
C085H	The target station's channel specified by the link dedicated instruction SEND is currently in use.	Correct the target station's channel of the SEND instruction.	—
C0B2H	(Receive buffer full error)	• Increase the request interval and execute the operation. • Do not access through one station using the MELSOFT connection, link dedicated instruction, or SLMP. • Wait for a response to the previous request before sending the next request. • Correct the set value for "Timer Settings for Data Communication" under "Application Settings" of the Ethernet-equipped module.	—
C0B3H	A request that cannot be processed was issued from the CPU module.	• Correct the request details. • Correct the network number or request destination station number.	—
C0B6H	The channel specified by the dedicated instruction is out of the range.	Correct the channel to a value within the allowable range of each dedicated instruction.	—
C0BAH	Since the close processing is being performed using the CONCLOSE/CLOSE instruction, a send request cannot be accepted.	Do no perform the send processing during the close processing.	—
C0C4H	The UINI instruction has been executed during communications.	Execute the UINI instruction after closing all connections.	—
C0D0H	The specified data length of the link dedicated instruction is incorrect.	Correct the data length.	—

Error code	Error details and causes	Action	Detailed information
C0D1H	The number of resends of the link dedicated instruction is incorrect.	Correct the number of resends.	—
C0D3H	The number of relay stations to communicate with other networks exceeds the allowable range.	• Check if the specification (network number/station number) for the communication destination is correct. • Check that number of relay stations accessing to the communication destination is 7 or less. • Correct the settings in the Network station number <-> IP information for the stations between the own station and the communication destination.	—
C0D4H	The number of relay stations to communicate with other networks exceeds the allowable range.	• Check if the specification (network number/station number) for the communication destination is correct. • Check that number of relay stations accessing to the communication destination is 7 or less. • Correct the settings in the Network station number <-> IP information for the stations between the own station and the communication destination.	—
C0D5H	The number of retries of the link dedicated instruction is incorrect.	Correct the number of retries.	—
C0D6H	The network number or station number of the link dedicated instruction is incorrect.	Check if the specification (network number/station number) for the communication destination is correct.	—
C0D7H	Data were sent without the initial processing completed.	After normal completion of the initial processing, perform the communications with the external device.	—
C0D8H	The number of specified blocks exceeded the range.	Correct the number of blocks.	—
C0D9H	The specified subcommand of the SLMP message is incorrect.	Correct the subcommand.	—
C0DAH	A response to the PING test could not be received within the time of the communication time check.	• Correct the IP address and host name of the target module where the PING test is executed. • Check that the initial processing of the Ethernet-equipped module where the PING test is executed has completed successfully.	—
C0DBH	The IP address and host name of the target module where the PING test is execute are incorrect.	Correct the IP address and host name of the target module where the PING test is executed.	—
C0DEH	Data could not be received within the specified arrival monitoring time.	• Correct the specified arrival monitoring time. • Correct the channel of the link dedicated instruction. • Check if the sending station and relay station are operating normally.	—
C101H	A response could not be received from the DNS server.	• Check the address of the DNS server. • Check that the data communications with the DNS server is possible by using the Ping command. • Check that the own station IP address and the DNS server IP address are in the same class. (If not, check the router setting.)	—
C1A2H	• A response to the request could not be received. • In transient transmission, the number of relay to other networks exceeded seven.	• Check and correct the response waiting time. • Change the system configuration so that the number of relay stations may be seven or less. • For the RECV instruction, execute again after correcting the channels used by own station in the control data. • For the RECV instruction, check that 'RECV execution request' (Un\G5301 b0 to b7) are on. • Check the operation of the external device. • Check if the external device supports the executed function.	—
C1A4H	• Any of the specified command, subcommand, or request destination module I/O number of the SLMP message is incorrect. • The specified clear function set by the ERRCLEAR instruction is incorrect. • The specified information to be read set by the ERRRD instruction is incorrect.	• Check that all of the specified command, subcommand, and request destination module I/O number of the SLMP message is correct. • Correct the specified value for the clear function set by the ERRCLEAR instruction. • Correct the specified value for the information to be read set by the ERRRD instruction.	—
C1A5H	The specified target station or clear target is incorrect.	Correct the specified the target station or clear target.	—
C1A6H	The specified connection number is incorrect.	Correct the setting value of the connection number.	—
C1A7H	The specified network number is incorrect.	Correct the specified network number.	—
C1A8H	The specified station number is incorrect.	Correct the specified station number.	—

Error code	Error details and causes	Action	Detailed information
C1A9H	The specified device number is incorrect.	Correct the specified device number.	—
C1AAH	The specified device name is incorrect.	Correct the specified device name.	—
C1ACH	The specified number of resends is incorrect.	Correct the number of resends.	—
C1ADH	The specified data length is incorrect.	Correct the specified data length.	—
C1AFH	The specified port number is incorrect.	Correct the specified port number	—
C1B0H	The open processing of the specified connection has already completed.	• Do not perform the open processing to a connection already opened. • When communications with the external device cannot be performed, perform the close processing before the open processing.	—
C1B1H	The open processing of the specified connection has not completed.	After completion of the open processing, perform the communication.	—
C1B2H	The open or close processing using CONOPEN/CONCLOSE/OPEN/CLOSE instruction is being executed in the specified connection.	Execute again after the CONOPEN/CONCLOSE/OPEN/CLOSE instruction has completed.	—
C1B3H	Another send or receive instruction is being executed in the specified channel.	• Change the channel number. • Execute again after the send or receive instruction has completed.	—
C1B4H	The specified arrival monitoring time is incorrect.	Set the arrival monitoring time to a value within the allowable range.	—
C1B8H	The RECV instruction was executed for the channel that had not received data.	• Correct the execution condition of the RECV instruction for the channel that had received data. • Correct the specified channel of the RECV instruction.	—
C1B9H	The CONOPEN/OPEN instruction cannot be executed for the specified connection.	Correct the specified connection.	—
C1BAH	The dedicated instruction was executed with the initialization not completed.	Execute the dedicated instruction after the initial processing has completed.	—
C1BBH	The target station CPU type of the link dedicated instruction is incorrect.	Correct the specified target station CPU type.	—
C1BCH	The target network number of the link dedicated instruction is incorrect.	• Execute the link dedicated instruction again after correcting the network number. • When specifying another network station, check if the settings in "Routing Setting" of "CPU Parameter" are made correctly.	—
C1BDH	The target station number of the link dedicated instruction is incorrect.	• Execute the link dedicated instruction again after correcting the target station number. • When specifying another network station, check if the settings in "Routing Setting" of "CPU Parameter" are made correctly.	—
C1BEH	The command code of the dedicated instruction is incorrect.	• Execute again after correcting the command code at the request source. • If the request source is on another network, check if the routing parameters are set correctly, and take action.	—
C1BFH	The channel used in the dedicated instruction is incorrect.	• Execute again after correcting the used channel within the allowable range at the request source. • If the request source is on another network, check if the routing parameters are set correctly, and take action.	—
C1C0H	The transient data is incorrect.	• Correct the transient data at the request source, and retry the operation. • If the error occurs again even after taking the above, please consult your local Mitsubishi representative.	—
C1C1H	The transient data is incorrect.	• Correct the transient data at the request source, and retry the operation. • If the error occurs again even after taking the above, please consult your local Mitsubishi representative.	—

Error code	Error details and causes	Action	Detailed information
C1C2H	When the dedicated instruction was executed, data was received twice.	• Check the network status and take corrective action using the Ethernet diagnostics of the engineering tool. • Check if the switching hub and the cables at the request source are connected properly. • If the request source is on another network, check if the routing parameters are set correctly, and take action.	—
C1C4H	The arrival check of the link dedicated instruction completed with an error.	• Execute link dedicated instruction again after changing the execution type in the control data to "No arrival acknowledgment". • For the REQ instruction, execute again after correcting request type.	—
C1C5H	A dedicated instruction which the target station does not support was executed.	• Change the target station at the station that executed the SEND instruction. • If the error occurs again even after taking the above, please consult your local Mitsubishi representative.	—
C1C6H	The execution or error completion type of the dedicated instruction is incorrect.	• Execute again after correcting the execution/error completion type in the control data. • If the problem cannot be resolved with the above actions, the possible cause is a hardware failure of the module. Please consult your local Mitsubishi representative.	—
C1C7H	The request type of the REQ instruction is incorrect.	• Execute again after correcting the request type of the REQ instruction. • If the problem cannot be resolved with the above actions, the possible cause is a hardware failure of the module. Please consult your local Mitsubishi representative.	—
C1C8H	The channel specified in the dedicated instruction is in use.	Change the channels used by own station or the target station's channel in the control data.	—
C1C9H	The device specification for the ZNRD/ZNWR instruction is not correct.	• Execute again after correcting the device specification for the ZNRD/ZNWR instruction. • If the problem cannot be resolved with the above actions, the possible cause is a hardware failure of the module. Please consult your local Mitsubishi representative.	—
C1CAH	The device specification for the ZNRD/ZNWR instruction is not correct.	• Execute again after correcting the device specification for the ZNRD/ZNWR instruction. • If the problem cannot be resolved with the above actions, the possible cause is a hardware failure of the module. Please consult your local Mitsubishi representative.	—
C1CBH	The transient data is incorrect.	• Correct the transient data at the request source, and retry the operation. • If the error occurs again even after taking the above, please consult your local Mitsubishi representative.	—
C1CCH	A response of the data length that exceeds the allowable range was received by the SLMPSND instruction.	• Execute again after correcting the request data to be within the range. • If the error occurs again even after taking the above, please consult your local Mitsubishi representative.	—
C1CDH	Message send of the SLMPSND instruction has failed.	• Check the operation of the external device or switching hub. • Since there may be congestion of packets on the line, send data after a certain period of time. • Check if the connection cable is disconnected. • Check that there is no connection failure with the switching hub. • Execute the communication status test, and if the test completed with an error, take the corrective action. • Execute the module communication test, and check that there is no failure in the module. • Check the IP address specified as the destination.	—

Error code	Error details and causes	Action	Detailed information
C1D0H	The requested module I/O No. of the dedicated instruction is incorrect.	• Execute again after correcting the requested module I/O No. at the request source of the dedicated instruction. • If the error occurs again even after taking the above, please consult your local Mitsubishi representative.	—
C1D2H	The target station IP address of the link dedicated instruction is incorrect.	Execute the link dedicated instruction again after correcting the IP address.	—
C1D3H	The dedicated instruction not supported by the communication method of the connection was executed.	• Check that the dedicated instruction can be executed by the specified communication method. Correct the program when the instruction cannot be executed. • Check that there is no error in the connection specification of the dedicated instruction.	—
C200H	The remote password is incorrect.	Correct the remote password, and unlock/lock the remote password again.	—
C201H	The remote password status of the port used for communications is in the lock status.	After unlocking the remote password, perform communications.	—
C202H	When another station was accessed, the remote password could not be unlocked.	When accessing another station, do not set the remote password on the relay station or access station, or do not execute the remote password check on them.	—
C203H	An error has occurred by checking the remote password.	Correct the remote password, and unlock/lock the remote password again.	—
C204H	The device is different from the one requesting the remote password unlock processing.	Request the lock processing of the remote password from the external device that requested the unlock processing of the remote password	—
C205H	When another station was accessed, the remote password could not be unlocked.	When accessing another station, do not set the remote password on the relay station or access station, or do not execute the remote password check on them.	—
C207H	The file name has too many characters.	Name the file with 255 characters or less.	—
C208H	The password length is out of range.	Set the password within 6 to 32 characters.	—
C400H	The ECPRTCL instruction was executed when Predefined protocol ready does not complete.	• Execute the ECPRTCL instruction after Predefined protocol ready has turned on. • Execute the ECPRTCL instruction after rewriting the protocol setting data to the Ethernet-equipped module. • If the error occurs again even after taking the above, the possible cause is a hardware failure of the module. Please consult your local Mitsubishi representative.	—
C401H	The protocol number specified by the ECPRTCL instruction is not registered in the Ethernet-equipped module.	• Correct the specified protocol number and execute the instruction again. • Register the protocol specified protocol number to the Ethernet-equipped module.	—
C402H	A error has occurred in the protocol setting data registered in the Ethernet-equipped module and the ECPRTCL instruction cannot be executed.	Correct the protocol setting data and register it again.	—
C403H	Multiple dedicated instructions was executed simultaneously.	• Do not execute the dedicated instructions which do not support simultaneous execution. • Correct the specified connection number and execute the dedicated instruction again.	—
C404H	The protocol being executed by the ECPRTCL instruction was canceled.	Check the canceled protocol in the control data of the ECPRTCL instruction (execution count result) and eliminate the cause of the cancellation.	—
C405H	The protocol number specified by the ECPRTCL instruction is incorrect.	Correct the specified protocol number.	—
C406H	The continuous protocol execution count of the ECPRTCL instruction is incorrect.	Correct the continuous protocol execution count.	—
C407H	The connection number specified by the ECPRTCL instruction is incorrect.	• Correct the specified connection number and execute the protocol again. • Correct the specified connection number in "External Device Configuration" under "Basic Settings" of the Ethernet-equipped module and execute the protocol again.	—

Error code	Error details and causes	Action	Detailed information
C408H	An error has occurred when the send processing of the predefined protocol using the ECPRTCL instruction was performed.	• Check the operation of the external device or switching hub. • Since there may be congestion of packets on the line, send data after a certain period of time. • Check if the connection cable is disconnected. • Check that there is no connection failure with the switching hub. • Execute the communication status test, and if the test completed with an error, take the corrective action. • Execute the module communication test, and check that there is no failure in the module.	—
C410H	Receive waiting time of the ECPRTCL instruction timed out.	• Check if the cable is disconnected. • Correct the specified connection number in "External Device Configuration" under "Basic Settings" of the Ethernet-equipped module and execute the protocol again. • Check that there is no error in the external device. • Check that the sending from the external device is not interrupted. • Check that there is no data lost due to a receive error. • Check that there is no error in the data (packet) sent by the external device.	—
C412H	The data which cannot be converted from ASCII to binary code was received.	• Check that there is no data lost due to a receive error. • Check that there is no error in the data (packet) sent by the external device.	—
C413H	The number of digits of the received data using the predefined protocol is not sufficient.	• Check that there is no data lost due to a receive error. • Check that there is no error in the data (packet) sent by the external device.	—
C414H	The number of digits of the received data using the predefined protocol is incorrect.	• Check that there is no data lost due to a receive error. • Check that there is no error in the data (packet) sent by the external device.	—
C417H	The data length or data quantity of the received data using the predefined protocol is out of range.	• Check the maximum allowable data length and specify the maximum length or less in the data length storage area. • Check the maximum allowable data quantity, and specify the maximum quantity or less in the data quantity storage area.	—
C420H	Protocol setting data write has failed.	• Write the data again. • If the error occurs again even after taking the above, the possible cause is a hardware failure of the specified module. Please consult your local Mitsubishi representative.	—
C421H	Writing was requested to the module whose flash ROM write count had exceeded the limit.	Replace the module because the number of writes exceeded the limit.	—
C430H	Protocol setting data was written during the ECPRTCL instruction execution.	Do not write the protocol setting data during the ECPRTCL instruction execution.	—
C431H	Close processing of the connection was performed during the ECPRTCL instruction execution.	• Check the operation of the external device. • Check the connection open status with the external device. • Open the connection with the external device again and execute the instruction.	—
C440H to C44FH	A communication error has occurred with an engineering tool when executing the Ethernet diagnostics.	• Execute the communication status test, and if the test completed with an error, take the corrective action. • Execute the module communication test, and check that there is no failure in the module.	—
C610H to C613H	The module processing completed with an error.	• Execute the communication status test, and if the test completed with an error, take the corrective action. • Execute the module communication test, and check that there is no failure in the module.	—
C614H	The module processing completed with an error.	• Correct the setting value of "Response Monitoring Timer" under "FTP Server Settings" because writing files may require a longer time. • Execute the communication status test, and if the test completed with an error, take the corrective action. • Execute the module communication test, and check that there is no failure in the module.	—

Error code	Error details and causes	Action	Detailed information
C615H	The module processing completed with an error.	• Execute the communication status test, and if the test completed with an error, take the corrective action. • Execute the module communication test, and check that there is no failure in the module.	—
C616H	Connection of the control port to the FTP server failed.	• Correct the IP address setting of the Ethernet-equipped module. • Correct the FTP server setting. • Check connection with the FTP server. • Disconnect the user session on the FTP server. • Data communications may not be ready. Wait for a while and perform the operation again. • Correct the TCP ULP timeout value because connecting to the FTP server may require a longer time.	—
C617H	Disconnection of the control port to the FTP server failed.	• Correct the FTP server setting. • Check connection with the FTP server.	—
C618H	Login to the FTP server failed.	• Correct the FTP server setting (login user name and login password). • Check the FTP server software settings (login user name and login password). • Check the data communication history of the FTP server software.	—
C619H	Execution of the FTP command to the FTP server failed.	• Correct the FTP server setting (folder path and connection method). • Check that the user has a right to access (read/write) the FTP server or the specified file. • Check that the specified folder path exists in the FTP server. • Check that the specified file exists in the FTP server. • Correct the FTP server software settings. • Check the data communication history of the FTP server software. • Check that the file access is being performed in the FTP server.	—
C620H	Connection of the data transfer port to the FTP server failed.	• Check connection with the FTP server. • Correct the FTP server setting (connection method). • If a firewall is active or the proxy server is on the connection path, consult a network administrator about the settings.	—
C621H	Disconnection of the data transfer port to the FTP server failed.	• Check connection with the FTP server. • Correct the FTP server setting (connection method). • If a firewall is active or the proxy server is on the connection path, consult a network administrator about the settings.	—
C622H	An error has occurred during file transfer to the FTP server.	• Delete unnecessary files on the FTP server to increase free space. • Check connection with the FTP server. • The specified file may be used in the other process. Wait for a while and perform the operation again. • The Ethernet line may be congested. Wait for a while and perform the operation again.	—
C623H	A response could not be received from the FTP server.	• Check that the FTP server name is registered in the DNS. • Change the FTP server name to the IP address, and check the operation. • Check that the data communications with the FTP server is possible by using the Ping command.	—
C700H	The module processing completed with an error.	• Execute the communication status test, and if the test completed with an error, take the corrective action. • Execute the module communication test, and check that there is no failure in the module.	—
C701H	The IP address (network number) setting is incorrect in communications using the IP packet transfer function.	Check the IP address (network number).	—
C702H	The IP address (station number) setting is incorrect in communications using the IP packet transfer function.	Check the IP addresses (station number).	—

Error code	Error details and causes	Action	Detailed information
C703H	The destination IP address (upper level) setting is incorrect in communications using the IP packet transfer function.	Check the destination IP address.	—
C704H	The destination IP address (lower level) setting is incorrect in communications using the IP packet transfer function.	Check the destination IP address.	—
C705H to C707H	The module processing completed with an error.	• Execute the communication status test, and if the test completed with an error, take the corrective action. • Execute the module communication test, and check that there is no failure in the module.	—
C708H	When communicating with the IP packet transfer function, "IP Packet Transfer Function" is set as "Not Use" in "IP Packet Transfer Setting" under "Application Settings" of the Ethernet-equipped module connected with the Ethernet devices.	When communicating with the IP packet transfer function, set "IP Packet Transfer Function" as "Not Use" in "IP Packet Transfer Setting" under "Application Settings".	—
C709H	A communication error has occurred with MELSOFT direct connection.	• Do no execute the specification of the direct connection when direct connection is not used. • Do not power off the system or reset the CPU module, or remove the cable during the communications when direct connection is used.	—
C810H	Remote password authentication has failed when required.	Set a correct password and perform password authentication again.	—
C811H	Remote password authentication has failed when required.	Set a correct password and perform password authentication again one minute later.	—
C812H	Remote password authentication has failed when required.	Set a correct password and perform password authentication again 5 minutes later.	—
C813H	Remote password authentication has failed when required.	Set a correct password and perform password authentication again 15 minutes later.	—
C814H	Remote password authentication has failed when required.	Set a correct password and perform password authentication again 60 minutes later.	—
C815H	Remote password authentication has failed when required.	Set a correct password and perform password authentication again 60 minutes later.	—
C816H	The security function was activated and remote password authentication cannot be performed.	Set a correct password and perform password authentication again after a certain period of time.	—
C840H	Number of transient request exceeded the upper limit of simultaneously processable requests.	• Pause the transient transmission temporarily, and retry the operation. • Lower the transient transmission usage frequency, and then perform again.	—
C842H	The routing setting is not set to reach to the destination network number.	• Execute the link dedicated instruction again after correcting the target network number and station number. • When the dynamic routing is used, check that communication path to the destination network number is set. • When the dynamic routing is not used, or a module of the series other than MELSEC iQ-R is included, retry the operation after correcting the settings in "Routing Setting" of "CPU Parameter".	—
C843H	Link dedicated instruction that cannot be executed on the network type were executed.	Check the network type of the Ethernet-equipped module.	—
C844H	Incorrect frame is received. • Unsupported pre-conversion protocol • Unsupported frame type • Application header variable part • Application header HDS • Application header RTP • Read command not requiring response	Correct the request data at the transient request source, and retry the operation.	—
CEE0H	The devices supporting iQSS which were detected by the other peripheral device, or other iQSS functions were executed while the automatic detection of connected devices is in process.	Execute the other function after the automatic detection of connected devices has completed.	—

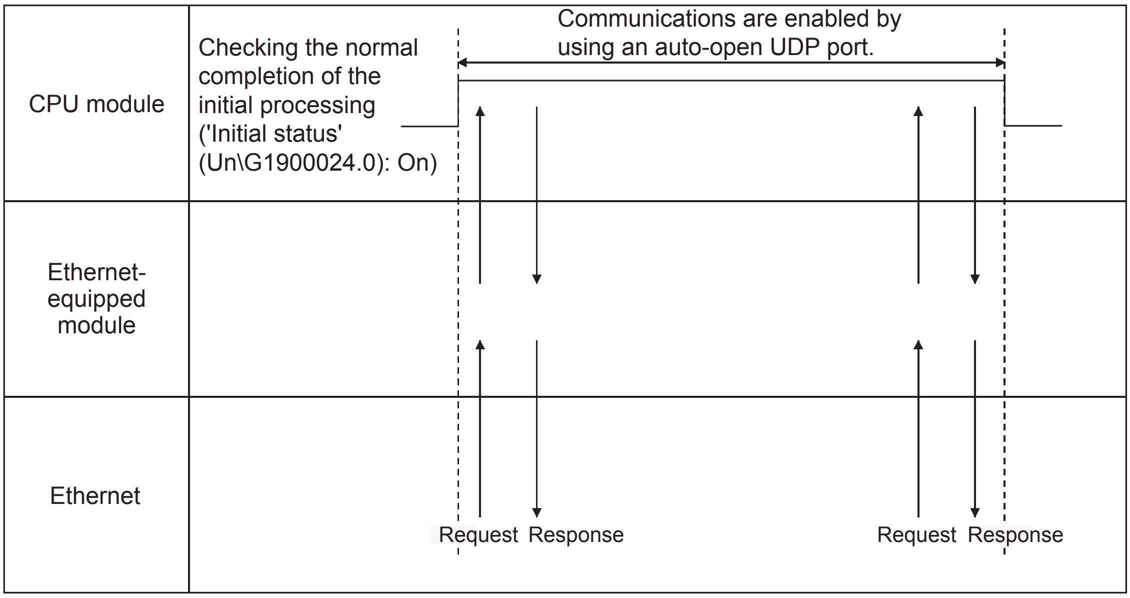
Error code	Error details and causes	Action	Detailed information
CEE1H	Incorrect frame is received.	• Check the operating status and connection status of the target device. • Check the connection of an Ethernet cable and a hub. • Check the line status of Ethernet. • Reset the CPU module and target device, and retry the operation. If the above actions do not solve the problem, contact the manufacturer of the target device.	—
CEE2H	Incorrect frame is received.	• Check the operating status and connection status of the target device. • Check the connection of an Ethernet cable and a hub. • Check the line status of Ethernet. • Reset the CPU module and target device, and retry the operation. If the above actions do not solve the problem, contact the manufacturer of the target device.	—
CF10H	Incorrect frame is received.	• Check the operating status and connection status of the target device. • Check the connection of an Ethernet cable and a hub. • Check the line status of Ethernet. • Reset the CPU module and target device, and retry the operation. If the above actions do not solve the problem, contact the manufacturer of the target device.	—
CF20H	• The setting value of the communication setting is out of range. • The items of communication setting which cannot be set on the target device are set. • The required setting items have not been set to the target device.	Correct the setting details, and retry the operation.	—
CF30H	The parameter which is not supported by the target device was specified.	Check the version of the target device.	—
CF31H	Incorrect frame is received.	• Check the operating status and connection status of the target device. • Check the connection of an Ethernet cable and a hub. • Check the line status of Ethernet. • Reset the CPU module and target device, and retry the operation. If the above actions do not solve the problem, contact the manufacturer of the target device.	—
CF70H	An error occurred on the Ethernet communication path.	• Check the operation of the target device. • Check if the connection cable is disconnected.	—
CF71H	A timeout error has occurred.	• Check the operation of the target device. Since there may be congestion of packets on the line, perform the operation after a while. • Correct the setting details of when the iQSS function is executed, and retry the operation. • Check the connection of an Ethernet cable and a hub.	—

Appendix 8 Procedure for Communications

Appendix 8.1 Auto-open UDP port

The auto-open UDP port is used for communications using the SLMP.

The auto-open UDP port is a UDP/IP port that automatically opens and closes at the following timing. When this port is used, communications are enabled when the initial processing completes. Communications can be performed without a program regardless of the connection's open status.


CPU module
Ethernet-equipped module
Ethernet
Checking the normal completion of the initial processing ('Initial status' (Un\G1900024.0): On)
Communications are enabled by using an auto-open UDP port.
Request
Response
Request
Response

Open/close timing

After the Ethernet-equipped module initial processing completes, the port automatically opens according to the registered parameter settings. The port automatically closes when the Ethernet-equipped module station is powered off or reset.

Point

- When the initial processing completes successfully, the Ethernet-equipped module enables communications using an automatic open UDP port. The module waits for a communication request to the Ethernet-equipped module on the own station. (Automatic open)
- The Ethernet-equipped module accepts and processes requests from anywhere as long as they are addressed to the Ethernet-equipped module itself.
- If a communication request is received from an external device, the corresponding port number is occupied until that processing ends. Even if another communication request is accepted during this time, the communication processing will be waited.

Appendix 9 Dedicated Instructions

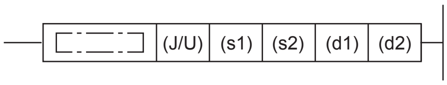
Appendix 9.1 Reading data from the programmable controller on another station

JP.READ, GP.READ

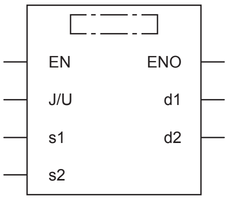
RnCPU RnENCPU RnPCPU (Process) RnPCPU (Redundant) RnSFCPU (Standard) ~~RnSFCPU (Safety)~~

These instructions read data from a device in the programmable controller of another station (in units of words).

Ladder	ST

	ENO: =JP_READ(EN,J,s1,s2,d1,d2);
	ENO: =GP_READ(EN,U,s1,s2,d1,d2);

FBD/LD



■Execution condition

Instruction	Execution condition
JP.READ GP.READ	↑

Setting data

■Description, range, data type

Operand		Description	Range	Data type	Data type (label)
(J/U)	JP.READ	(J): Own station network number	1 to 239	16-bit unsigned binary	ANY16
	GP.READ	(U): Start I/O number (first three digits in four-digit hexadecimal representation) of own station or own node	00H to FEH	16-bit unsigned binary	ANY16
(s1)		Own station start device where control data is stored	Refer to the control data.	Device name	ANY16[*1]
(s2)		Target station start device where the data to be read is stored	—	String[*2]	ANYSTRING_SINGLE[*2]
(d1)		Own station start device (a continuous area for the length of the read data) for storing the data that has been read	—	Device name	ANY16[*1]
(d2)		Device of the own station, which turns on for one scan upon completion of the instruction. When the instruction completes with an error, (d2)+1 also turns on.	—	Bit	ANYBIT_ARRAY (Number of elements: 2)
EN		Execution condition	—	Bit	BOOL
ENO		Execution result	—	Bit	BOOL

[*1] When specifying setting data by using a label, define an array to secure enough operation area and specify an element of the array label.

[*2] For the specifications of the string data to be specified, refer to the following.
📖 MELSEC iQ-R Programming Manual (Instructions, Standard Functions/Function Blocks)

■Applicable devices

Operand		Bit		Word			Double word		Indirect specification	Constant			Others	
		X, Y, M, L, SM, F, B, SB, FX, FY	J□\□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□\G□, J□\□, U3E□\(H)G□	Z	LT, LST, LC	LZ		K, H	E	$	J	U
(J/U)	JP.READ	—	—	—	—	—	—	—	—	—	—	—	○	—
	GP.READ	—	—	○	—	—	—	—	○	○	—	—	—	○
(s1)		—	—	○[2]	—	—	—	—	○	—	—	—	—	—
(s2)		—	—	—	—	—	—	—	—	—	—	○	—	—
(d1)		—	—	○[2]	—	—	—	—	○	—	—	—	—	—
(d2)		○[1]	—	○[3]	—	—	—	—	—	—	—	—	—	—

[1] FX and FY cannot be used.

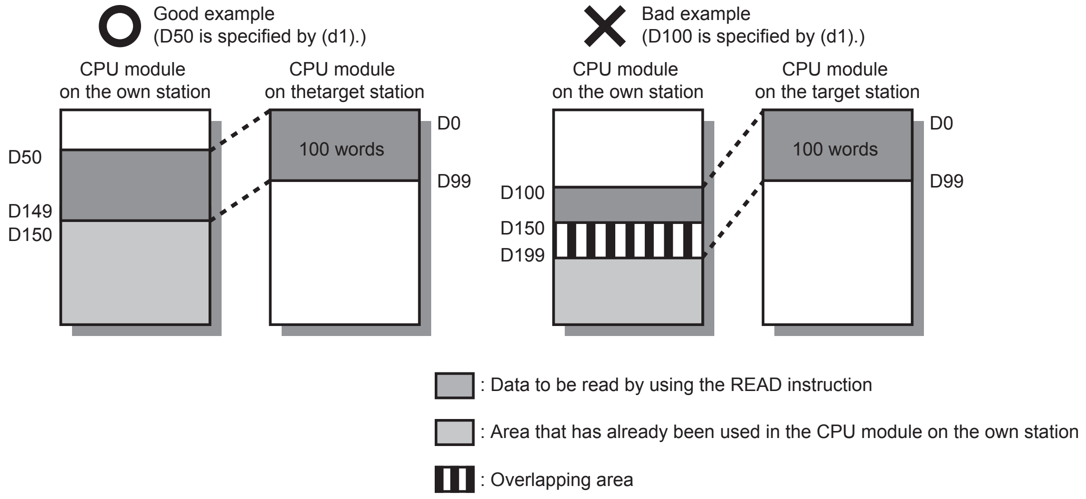
[2] FD cannot be used.

[3] T, ST, C, and FD cannot be used.

Point

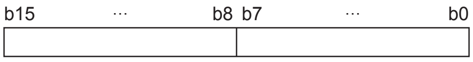
- The READ instruction cannot be executed when the target station is ACPU.
- Specify the own station start device (d1) for storing the data that has been read, by considering the range in which the data that has been read can be stored.

(Example: When areas D150 and later in the own station CPU module are already in use)



■Control data

Operand: (s1)				
Device	**Item**	**Description**	**Setting range**	**Set by**
+0	Error completion type	b15 ··· b8 b7 ··· b0 (3) 0 (2) (1) 0 (1) Error completion type (bit 7) Specify whether to set data at completion with an error. • 0: Dos not set data in (s1)+12 and later at completion with an error. • 1: Set data in (s1)+12 and later at completion with an error. (2) Arrival check time setting (bit 8) • 0: 1s units • 1: 100ms units (3) Target station address specification method (bit 15) • 0: Specify "network No." in (s1)+4 and "station number" in (s1)+5. • 1: Specify "IP address" in (s1)+4 and (s1)+5 (Ethernet only).	0000H 0080H 0100H 0180H 8000H 8080H 8100H 8180H	User
+1	Completion status	The completion status is stored upon completion of the instruction. • 0: Normal • Other than 0: Error (error code)	—	System
+2	Own station channel	Specify the channel used by the own station.	1 to 18	User
+3	Target station CPU type	Specify the CPU type of the target station. • 0000H: Addressed to target station CPU (control CPU) • 03D0H: Addressed to control system CPU • 03D1H: Addressed to standby system CPU • 03D2H: Addressed to system A CPU • 03D3H: Addressed to system B CPU • 03E0H: Addressed to multiple CPU No.1 • 03E1H: Addressed to multiple CPU No.2 • 03E2H: Addressed to multiple CPU No.3 • 03E3H: Addressed to multiple CPU No.4 • 03FFH: Addressed to target station CPU (control CPU)	0000H 03D0H to 03D3H 03E0H to 03E3H 03FFH	User
+4	Target network number	[CC-Link IE Controller Network or CC-Link IE Field Network] Specify the network number (1 to 239) of the target station. [Ethernet] ■When "0" is specified in bit 15 of (s1)+0 Specify the network number (1 to 239) of the target station. ■When "1" is specified in bit 15 of (s1)+0 Specify the IP address (third and fourth octets) of the target station. b15 ··· b8 b7 ··· b0 • b8 to b15: 3rd octet • b0 to b7: 4th octet	■(s1)+4 Network No.: 1 to 239 ■(s1)+5 Station No.: 1 to 120, 125, 126 ■(s1)+4, 5 IP address: 00000001H to FFFFFFFEH (1 to 4294967294)	User
+5	Target station number	[CC-Link IE Controller Network] Specify the station number (1 to 120) of the target station. [CC-Link IE Field Network] Specify the station number of the target station. • 125: Master station • 126: Master operating station • 1 to 120: Local station, intelligent device station, submaster station [Ethernet] ■When "0" is specified in bit 15 of (s1)+0 Specify the station number (1 to 120) of the target station. ■When "1" is specified in bit 15 of (s1)+0 Specify the IP address (first and second octets) of the target station. b15 ··· b8 b7 ··· b0 • b8 to b15: 1st octet • b0 to b7: 2nd octet		User
+6	Not used	—	—	—

Operand: (s1)				
Device	**Item**	**Description**	**Setting range**	**Set by**
+7	Number of resends	■Before instruction execution Specify the number of resends to be performed if the instruction does not complete within the monitoring time specified by (s1)+8. • 0 to 15 (times) ■At completion of instruction The number of resends performed (result) is stored. • 0 to 15 (times)	0 to 15	User/ system
+8	Arrival monitoring time	[CC-Link IE Controller Network or CC-Link IE Field Network] Specify the monitoring time until completion of processing. If the processing does not complete within the monitoring time, the request is resent by the number of resends specified by (s1)+7. ■When "0" is specified in bit 8 of (s1)+0 • 0: 10 seconds • 1 to 32767: 1 to 32767s ■When "1" is specified in bit 8 of (s1)+0 • 0: 10 seconds • 1 to 65535: 1 to 65535 × 100ms	0 to 65535	User
		[Ethernet] Specify the monitoring time until completion of processing. If the processing does not complete within the monitoring time, the request is resent by the number of resends specified by (s1)+7. ■When "0" is specified in bit 8 of (s1)+0 Specify the TCP resend timer value or greater for the monitoring time till completion of processing. • 0 to (TCP resend timer value): The TCP resend timer value is assumed as the monitoring time. • (TCP resend timer value + 1) to 16383: (TCP resend timer value + 1) to 16383s ■When "1" is specified in bit 8 of (s1)+0 • 0: 10 seconds • 1 to 65535: 1 to 65535 × 100ms	0 to 65535	User
+9	Read data length	Specify the number of words to be read. ■When reading data from RCPU, QCPU, or LCPU • Channels 1 to 8 are used: 1 to 960 (words) • Channels 9 and 10 are used: 1 to 8192 (words) ■When reading data from QnACPU • 1 to 480 (words)	1 to 8192	User
+10	Not used	—	—	—
+11	Clock setting flag	The validity status (valid or invalid) of the data in (s1)+12 and later is stored. Note that the data in (s1)+12 and later is not cleared even when the instruction has completed successfully. • 0: Invalid • 1: Valid	—	System
+12	Clock data (Set only in an abnormal state)	Upper 8 bits: Month (01H to 12H) Lower 8 bits: Year (00H to 99H: Upper two digits of the year)	—	System
+13		Upper 8 bits: Hour (00H to 23H) Lower 8 bits: Day (01H to 31H)	—	System
+14		Upper 8 bits: Second (00H to 59H) Lower 8 bits: Minute (00H to 59H)	—	System
+15		Upper 8 bits: Year (00H to 99H: Upper two digits of the year) Lower 8 bits: Day of the week (00H (Sun.) to 06H (Sat.))	—	System
+16	Error detection network number	■When "0" is specified in bit 15 of (s1)+0 The network number of the station in which an error was detected is stored. (No information is stored if an error is detected in the own station.) • 1 to 239 (Network number) ■When "1" is specified in bit 15 of (s1)+0 (Ethernet only) The IP address (third and fourth octets) of the station where an error was detected is stored.  • b8 to b15: 3rd octet • b0 to b7: 4th octet	—	System

Operand: (s1)

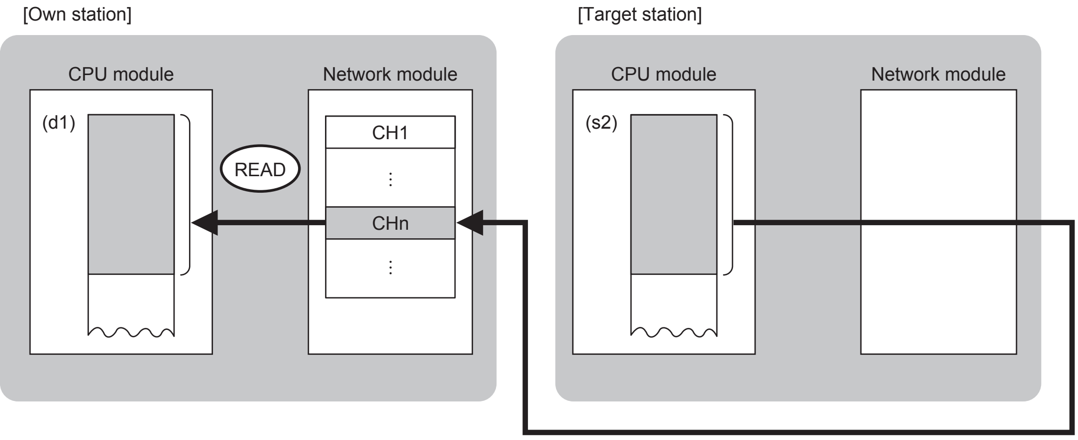
Device	Item	Description	Setting range	Set by
+17	Error-detected station number	■When "0" is specified in bit 15 of (s1)+0 The station number of the station in which an error was detected is stored. (No information is stored if an error is detected in the own station.) [Ethernet or CC-Link IE Controller Network] • 1 to 120: Station number [CC-Link IE Field Network] • 125: Master station • 1 to 120: Local station, intelligent device station, submaster station ■When "1" is specified in bit 15 of (s1)+0 (Ethernet only) The IP address (first and second octets) of the station where an error was detected is stored. b15 … b8 b7 … b0 • b8 to b15: 1st octet • b0 to b7: 2nd octet	—	System

Point

- The continuous area (a maximum of 8192 words) specified by the read data length (s1)+9 is required in the read data storage device (d1).
- The number of resends (s1)+7 must be set every time the instruction is executed.

Processing details

- These instructions read the data from the specified word device in the target station specified by the target network number and target station number of the control data or the target station specified by the IP address. Upon completion of reading the device data, the completion device specified by (d2) turns on.



CH: Channel

- When "network number" and "station number" are specified ("0" is specified in bit 15 of (s1)+0) by the target station address specification method, device data can be read also from stations connected to networks other than the stations connected to the own station network. (If "IP address" is specified ("1" is specified in bit 15 of (s1)+0), device data cannot be read from stations connected via a relay station.)
- When executing multiple link dedicated instructions simultaneously, check that the instructions do not use the same channel number. Link dedicated instructions specifying the same channel number cannot be executed simultaneously.
- The execution status and the completion status of the READ instruction can be checked with the completion device (d2) and the completion status indication device (d2)+1.

- Completion device (d2)

This device turns on during END processing of the scan where the READ instruction completes, and turns off during the next END processing.

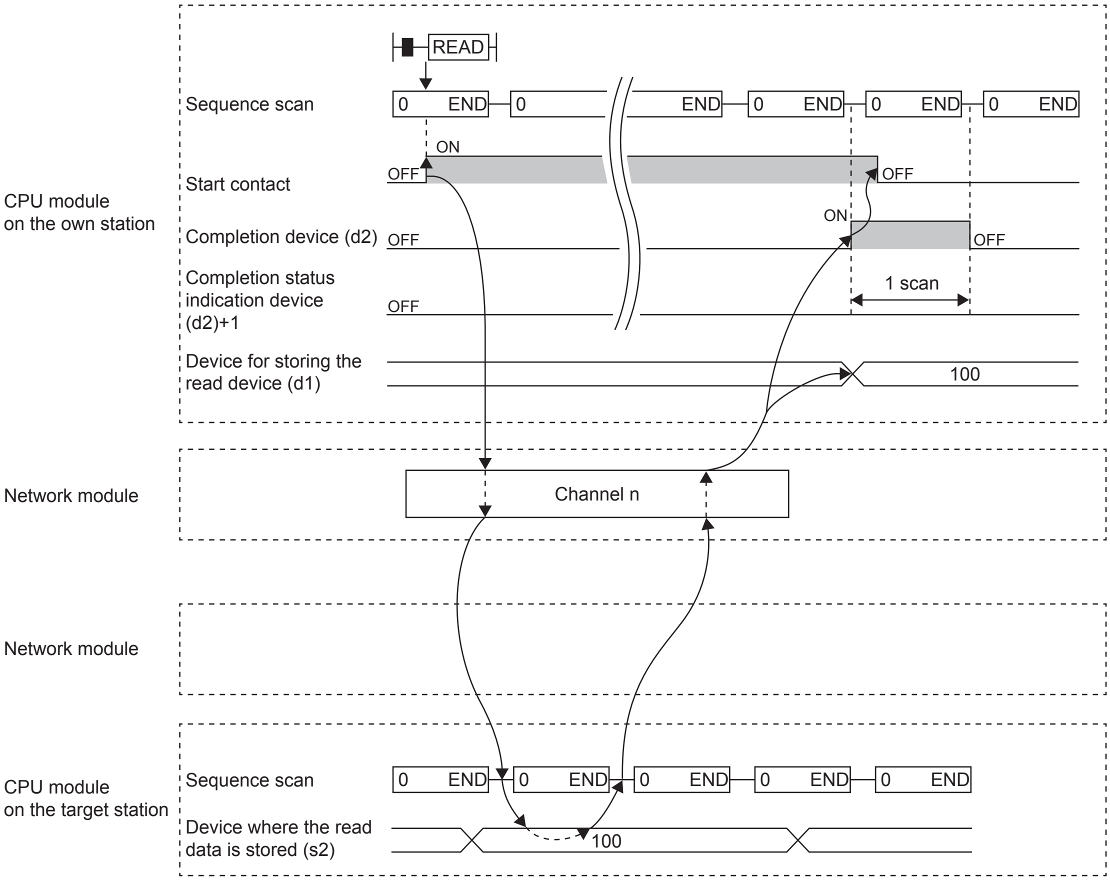
- Completion status indication device (d2)+1

This device turns on or off depending on the completion status of the READ instruction.

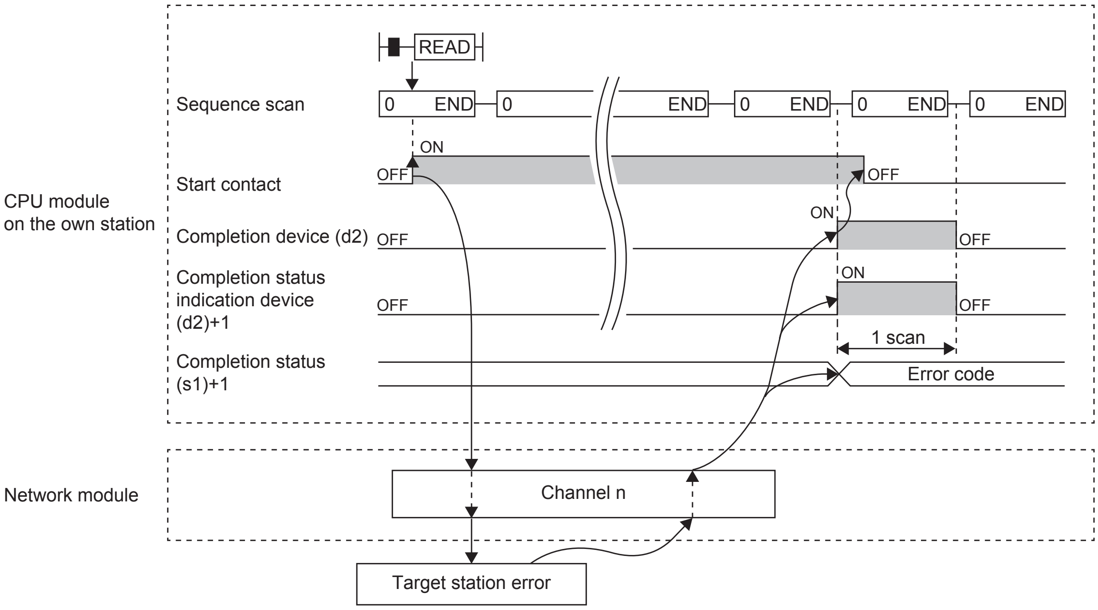
When completed successfully: The device remains off.

When completed with an error: The device turns on during END processing of the scan where the READ instruction completes, and turns off during the next END processing.

- The following figure shows the execution timing of the READ instruction.
  - When completed successfully


READ
CPU module on the own station
Sequence scan
0 END 0 END 0 END 0 END 0 END
ON
OFF
Start contact
OFF
ON
Completion device (d2)
OFF
OFF
Completion status indication device (d2)+1
OFF
1 scan
Device for storing the read device (d1)
100
Network module
Channel n
Network module
CPU module on the target station
Sequence scan
0 END 0 END 0 END 0 END 0 END
Device where the read data is stored (s2)
100

  - When completed with an error



- Read processing is performed only once on the rising edge when the read command turns on.

Operation error

Error code ((s1)+1)	Description
4000H to 4FFFH	📖 MELSEC iQ-R CPU Module User's Manual (Application)
6F00H to 6FFFH	📖 MELSEC iQ-R CPU Module User's Manual (Application)
C000H to CFFFH	📖 MELSEC iQ-R Ethernet User's Manual (Application)
D000H to DFFFH	📖 MELSEC iQ-R CC-Link IE Field Network User's Manual (Application)
E000H to EFFFH	📖 MELSEC iQ-R CC-Link IE Controller Network User's Manual (Application)

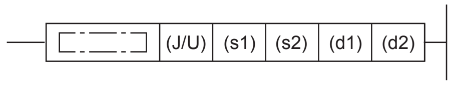
Appendix 9.2 Writing data to the programmable controller on another station

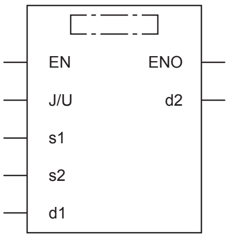
JP.WRITE, GP.WRITE

RnCPU RnENCPU RnPCPU (Process) RnPCPU (Redundant) RnSFCPU (Standard) RnSFCPU (Safety) ✗

These instructions write data to a device in the programmable controller of another station (in units of words).

Ladder	ST

	ENO: =JP_WRITE(EN,J,s1,s2,d1,d2);
	ENO: =GP_WRITE(EN,U,s1,s2,d1,d2);

FBD/LD


■Execution condition

Instruction	Execution condition
JP.WRITE GP.WRITE	↑

Setting data

■Description, range, data type

Operand		Description	Range	Data type	Data type (label)
(J/U)	JP.WRITE	(J): Own station network number	1 to 239	16-bit unsigned binary	ANY16
	GP.WRITE	(U): Start I/O number (first three digits in four-digit hexadecimal representation) of own station or own node	00H to FEH	16-bit unsigned binary	ANY16
(s1)		Own station start device where control data is stored	Refer to the control data.	Device name	ANY16[*1]
(s2)		Own station start device containing write data	—	Device name	ANY16[*1]
(d1)		Target station start device to which data is to be written	—	String[*2]	ANYSTRING_SINGLE[*2]
(d2)		Device of the own station, which turns on for one scan upon completion of the instruction. When the instruction completes with an error, (d2)+1 also turns on.	—	Bit	ANYBIT_ARRAY (Number of elements: 2)
EN		Execution condition	—	Bit	BOOL
ENO		Execution result	—	Bit	BOOL

*1 When specifying setting data by using a label, define an array to secure enough operation area and specify an element of the array label.

*2 For the specifications of the string data to be specified, refer to the following.
📖 MELSEC iQ-R Programming Manual (Instructions, Standard Functions/Function Blocks)

■Applicable devices

Operand		Bit		Word			Double word		Indirect specification	Constant			Others	
		X, Y, M, L, SM, F, B, SB, FX, FY	J□\□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□\G□, J□\□, U3E□\(H)G□	Z	LT, LST, LC	LZ		K, H	E	$	J	U
(J/U)	JP.WRITE	—	—	—	—	—	—	—	—	—	—	—	○	—
	GP.WRITE	—	—	○	—	—	—	—	○	○	—	—	—	○
(s1)		—	—	○*2	—	—	—	—	○	—	—	—	—	—
(s2)		—	—	○*2	—	—	—	—	○	—	—	—	—	—
(d1)		—	—	—	—	—	—	—	—	—	—	○	—	—
(d2)		○*1	—	○*3	—	—	—	—	—	—	—	—	—	—

*1 FX and FY cannot be used.

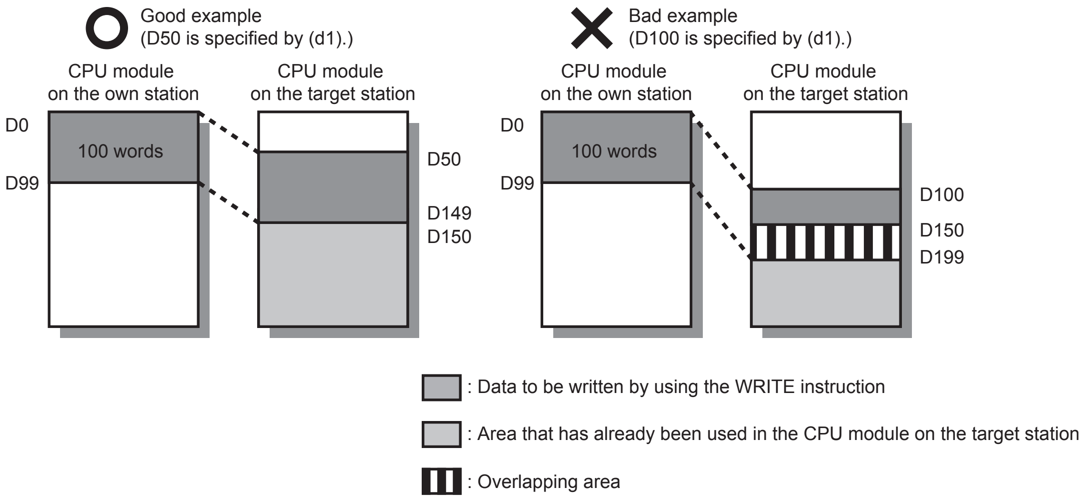
*2 FD cannot be used.

*3 T, ST, C, and FD cannot be used.

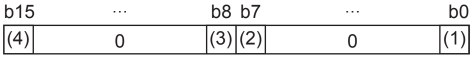
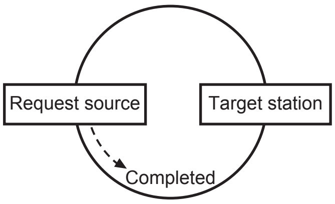
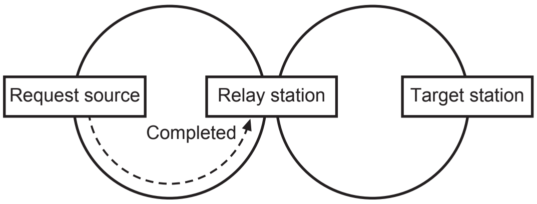
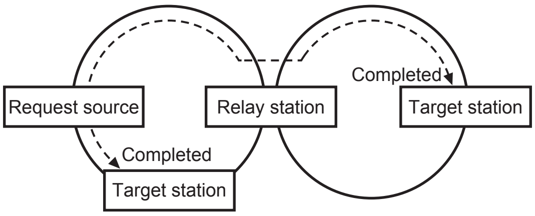
Point

- The WRITE instruction cannot be executed when the target station is ACPU.
- Specify the target station start device (d1), to which data is to be written, by considering the range in which the data that has been written can be stored.

(Example: When areas D150 and later in the target station CPU module are already in use)



■Control data

Operand: (s1)				
Device	**Item**	**Description**	**Setting range**	**Set by**
+0	Execution/error completion type	 b15 … b8 b7 … b0 (4) 0 (3) (2) 0 (1) (1) Execution type (bit 0) • 0: No arrival check When the target station is in the own network: Completed when data has been sent from the own station.  When the target station is in another network: Completed when data has arrived at the relay station of the own network.  Request source Relay station Target station Completed • 1: With arrival check Sending data completes when the data is written to the target station.  Completed Request source Relay station Target station Completed Target station (2) Error completion type (bit 7) Specify whether to set clock data after the instruction has completed with an error. • 0: Do not set data in (s1)+11 and later at completion with an error. • 1: Set data in (s1)+11 and later at completion with an error. (3) Arrival check time setting (bit 8) • 0: 1s units • 1: 100ms units (4) Target station address specification method (bit 15) • 0: Specify "network No." in (s1)+4 and "station number" in (s1)+5. • 1: Specify "IP address" in (s1)+4 and (s1)+5 (Ethernet only).	0000H 0001H 0080H 0081H 0100H 0101H 0180H 0181H 8000H 8001H 8080H 8081H 8100H 8101H 8180H 8181H	User
+1	Completion status	The completion status is stored upon completion of the instruction. • 0: Normal • Other than 0: Error (error code)	—	System
+2	Own station channel	Specify the channel used by the own station.	1 to 18	User
+3	Target station CPU type	Specify the CPU type of the target station. • 0000H: Addressed to target station CPU (control CPU) • 03D0H: Addressed to control system CPU • 03D1H: Addressed to standby system CPU • 03D2H: Addressed to system A CPU • 03D3H: Addressed to system B CPU • 03E0H: Addressed to multiple CPU No.1 • 03E1H: Addressed to multiple CPU No.2 • 03E2H: Addressed to multiple CPU No.3 • 03E3H: Addressed to multiple CPU No.4 • 03FFH: Addressed to target station CPU (control CPU)	0000H 03D0H to 03D3H 03E0H to 03E3H 03FFH	User

Operand: (s1)

Device	Item	Description	Setting range	Set by
+4	Target network number	[CC-Link IE Controller Network or CC-Link IE Field Network] Specify the network number (1 to 239) of the target station.	■(s1)+4 Network No.: 1 to 239 ■(s1)+5 Station No.: 1 to 120, 125, 126 Group number: 0081H to 00A0H All-station specification: 00FFH ■(s1)+4, 5 IP address: 00000001H to FFFFFFFEH (1 to 4294967294)	User
		[Ethernet] ■When "0" is specified in bit 15 of (s1)+0 Specify the network number (1 to 239) of the target station. ■When "1" is specified in bit 15 of (s1)+0 (Ethernet only) Specify the IP address (third and fourth octets) of the target station. b15 ··· b8 b7 ··· b0 • b8 to b15: 3rd octet • b0 to b7: 4th octet		
+5	Target station number	[CC-Link IE Controller Network] Specify the station number of the target station. (1) Station number specification • 1 to 120: Station number (2) Group number specification • 0081H to 00A0H: All stations with group numbers 0001H to 0020H (The number can be set when the execution type specified by (s1) is "0: Without arrival check".) (3) All-station specification • 00FFH: All stations of target network number (broadcast (excluding the own station)) (The number can be set when the execution type specified by (s1) is "0: Without arrival check".)		User
		[CC-Link IE Field Network] Specify the station number of the target station. (1) Station number specification • 125: Master station • 126: Master operating station • 1 to 120: Local station, intelligent device station, submaster station (2) All-station specification • 00FFH: All stations of target network number (broadcast (excluding the own station)) (The number can be set when the execution type specified by (s1) is "0: Without arrival check".)		
		[Ethernet] ■When "0" is specified in bit 15 of (s1)+0 Specify the station number of the target station. (1) Station number specification • 1 to 120: Station number (2) Group number specification • 0081H to 00A0H: All stations with group numbers 0001H to 0020H (The number can be set when the execution type specified by (s1) is "0: Without arrival check".) (3) All-station specification • 00FFH: All stations of target network number (broadcast (excluding the own station)) (The number can be set when the execution type specified by (s1) is "0: Without arrival check".) ■When "1" is specified in bit 15 of (s1)+0 Specify the IP address (first and second octets) of the target station. b15 ··· b8 b7 ··· b0 • b8 to b15: 1st octet • b0 to b7: 2nd octet		
+6	Not used	—	—	—
+7	Number of resends	Effective when the execution type specified by (s1) is "1: With arrival check". ■Before instruction execution Specify the number of resends to be performed if the instruction does not complete within the monitoring time specified by (s1)+8. • 0 to 15 (times) ■At completion of instruction The number of resends performed (result) is stored. • 0 to 15 (times)	0 to 15	User/ system

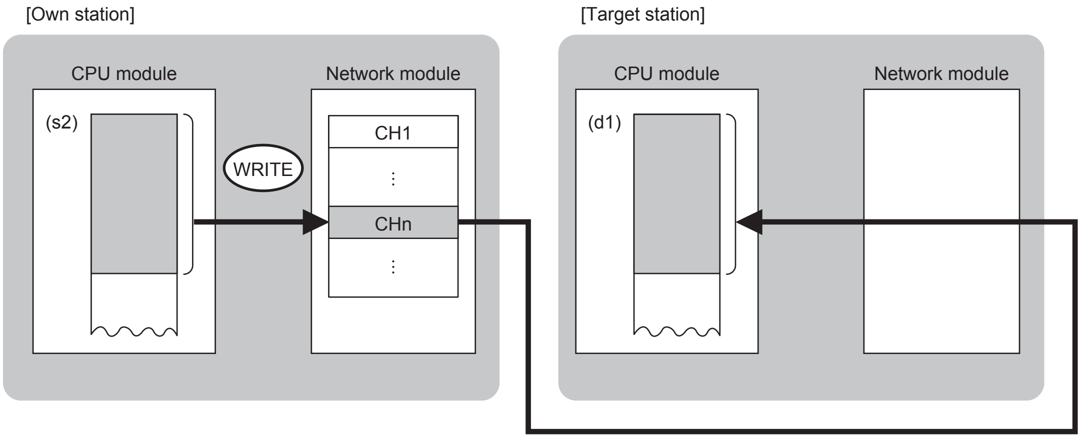
Operand: (s1)				
Device	**Item**	**Description**	**Setting range**	**Set by**
+8	Arrival monitoring time	[CC-Link IE Controller Network or CC-Link IE Field Network] Specify the monitoring time until completion of processing. If the processing does not complete within the monitoring time, the request is resent by the number of resends specified by (s1)+7. ■When "0" is specified in bit 8 of (s1)+0 • 0: 10 seconds • 1 to 32767: 1 to 32767 seconds ■When "1" is specified in bit 8 of (s1)+0 • 0: 10 seconds • 1 to 65535: 1 to 65535 × 100ms	0 to 65535	User
		[Ethernet] Specify the monitoring time until completion of processing. If the processing does not complete within the monitoring time, the request is resent by the number of resends specified by (s1)+7. ■When "0" is specified in bit 8 of (s1)+0 Specify the TCP resend timer value or greater for the monitoring time till completion of processing. • 0 to (TCP resend timer value): The TCP resend timer value is assumed as the monitoring time. • (TCP resend timer value + 1) to 16383: (TCP resend timer value + 1) to 16383s ■When "1" is specified in bit 8 of (s1)+0 • 0: 10 seconds • 1 to 65535: 1 to 65535 × 100ms	0 to 65535	User
+9	Write data length	Specify the number of words to be written. ■Writing to RCPU, QCPU, or LCPU • Channels 1 to 8 are used: 1 to 960 (words) • Channels 9 and 10 are used: 1 to 8192 (words) ■Writing to QnACPU • 1 to 480 (words)	1 to 8192	User
+10	Not used	—	—	—
+11	Clock setting flag	The validity status (valid or invalid) of the data in (s1)+12 and later is stored. Note that the data in (s1)+12 and later is not cleared even when the instruction has completed successfully. • 0: Invalid • 1: Valid	—	System
+12	Clock data (Set only in an abnormal state)	Upper 8 bits: Month (01H to 12H) Lower 8 bits: Year (00H to 99H: Upper two digits of the year)	—	System
+13		Upper 8 bits: Hour (00H to 23H) Lower 8 bits: Day (01H to 31H)	—	System
+14		Upper 8 bits: Second (00H to 59H) Lower 8 bits: Minute (00H to 59H)	—	System
+15		Upper 8 bits: Year (00H to 99H: Upper two digits of the year) Lower 8 bits: Day of the week (00H (Sun.) to 06H (Sat.))	—	System
+16	Error detection network number	The network number of the station in which an error was detected is stored. (No information is stored if an error is detected in the own station.) • 1 to 239 (Network number)	1 to 239	System
+17	Error-detected station number	The station number of the station in which an error was detected is stored. (No information is stored if an error is detected in the own station.) [Ethernet or CC-Link IE Controller Network] • 1 to 120: Station number [CC-Link IE Field Network] • 125: Master station • 1 to 120: Local station, intelligent device station, submaster station	1 to 120, 125	System

Point

- The continuous area (a maximum of 8192 words) for the write data length ((s1)+9) is required in the write data storage device (d1).
- When a number from 1 to 120 is specified for the target station number, the WRITE instruction should be executed with the execution type set to "With arrival check" to improve data reliability. When a number from 81H to A0H or FFH is specified for the target station number, the WRITE instruction should be executed with the execution type set to "Without arrival check".
- When performing device writing to the same station from multiple stations, do not overlap the write timing. When the execution type is set to "Without arrival check", successful completion results in the write source station if communications complete successfully even when the send data contains an error. Also, even when the send data is normal, a timeout results in the write source station if the WRITE instructions are executed for the same station from multiple stations.
- The number of resends (s1)+7 must be set every time the WRITE instruction is executed.

Processing details

- These instructions write the data in the device/label specified by (s2) in the own station to the word device in the target station specified by the target network number and target station number of the control data or the target station specified by the IP address. Upon completion of writing device data to another station number, the completion device specified by (d2) turns on.



CH: Channel

- When "network number" and "station number" are specified ("0" is specified in bit 15 of (s1)+0) by the target station address specification method, device data can also be written to the stations connected to networks other than the stations connected to the own station network. (If "IP address" is specified ("1" is specified in bit 15 of (s1)+0), device data cannot be read from stations connected via a relay station.)
- When executing multiple link dedicated instructions simultaneously, check that the instructions do not use the same channel number. Link dedicated instructions specifying the same channel number cannot be executed simultaneously.
- The execution status and the completion status of the WRITE instruction can be checked with the completion device (d2) and the completion status indication device (d2)+1.

- Completion device (d2)

This device turns on during END processing of the scan where the WRITE instruction completes, and turns off during the next END processing.

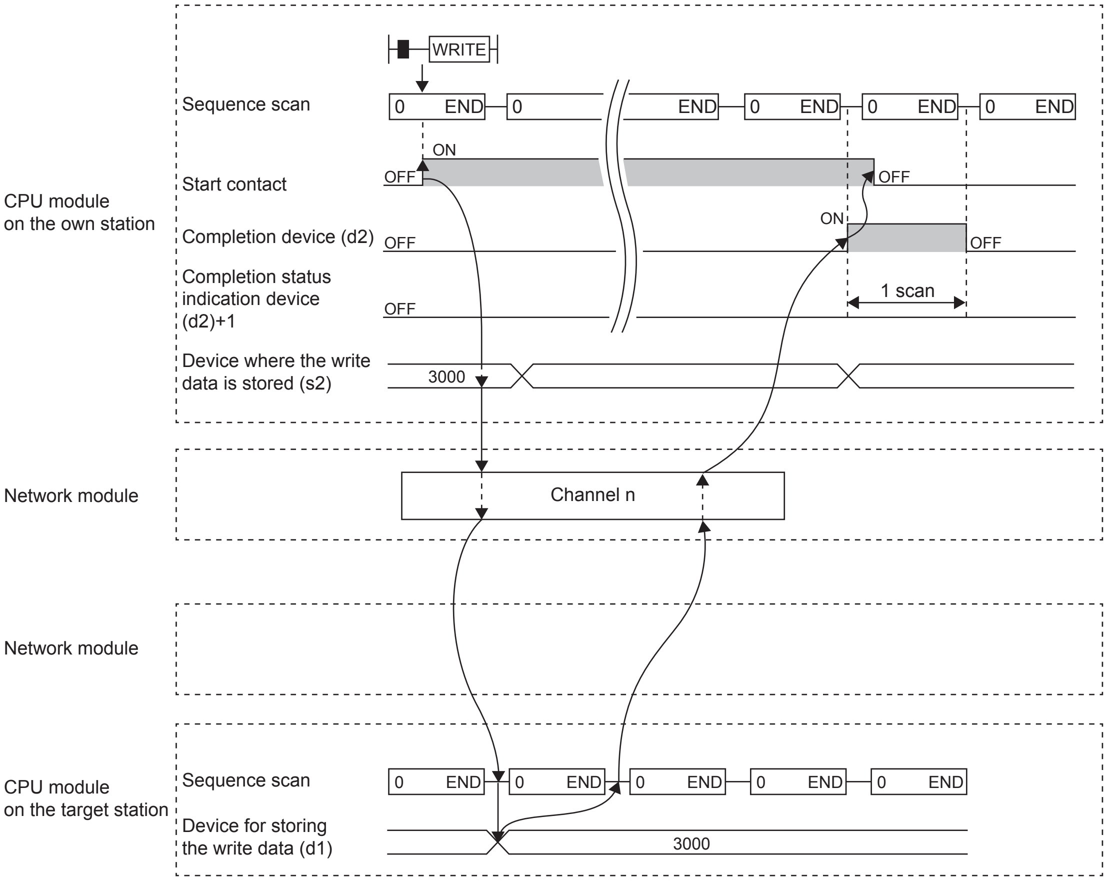
- Completion status indication device (d2)+1

This device turns on or off depending on the completion status of the WRITE instruction.

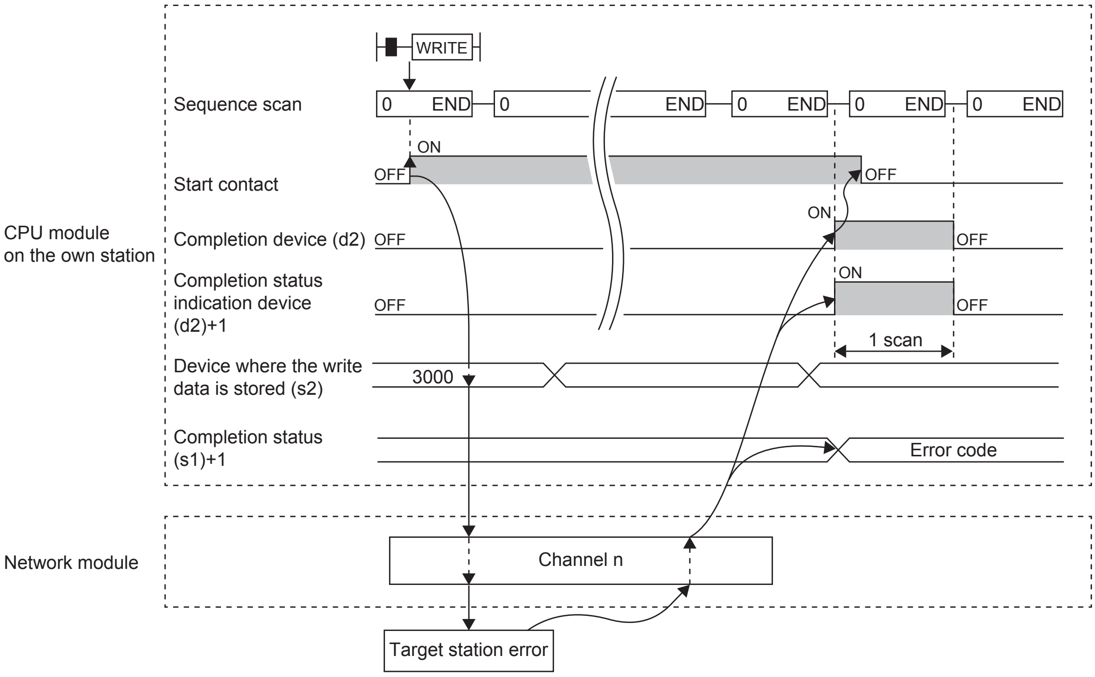
When completed successfully: The device remains off.

When completed with an error: The device turns on during END processing of the scan where the WRITE instruction completes, and turns off during the next END processing.

- The following figure shows the execution timing of the WRITE instruction.
  - When completed successfully



  - When completed with an error



- Write processing is performed only once on the rising edge when the write command turns on.

Operation error

Error code ((s1)+1)	Description
4000H to 4FFFH	📖 MELSEC iQ-R CPU Module User's Manual (Application)
6F00H to 6FFFH	📖 MELSEC iQ-R CPU Module User's Manual (Application)
C000H to CFFFH	📖 MELSEC iQ-R Ethernet User's Manual (Application)
D000H to DFFFH	📖 MELSEC iQ-R CC-Link IE Field Network User's Manual (Application)
E000H to EFFFH	📖 MELSEC iQ-R CC-Link IE Controller Network User's Manual (Application)

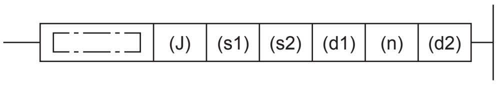
Appendix 9.3 Reading data from the programmable controller (Q series compatible)

J(P).ZNRD

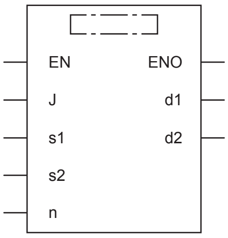
RnCPU RnENCPU RnPCPU (Process) RnPCPU (Redundant) RnSFCPU (Standard) RnSFCPU (Safety)

These instructions read data in units of words from a device in the programmable controller (MELSEC-A/QnA/Q/L series) of another station.

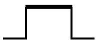
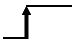
Ladder	ST

	ENO: =J_ZNRD(EN,J,s1,s2,n,d1,d2);
	ENO: =JP_ZNRD(EN,J,s1,s2,n,d1,d2);

FBD/LD



■Execution condition

Instruction	Execution condition
J.ZNRD	
JP.ZNRD	

Setting data

■Description, range, data type

Operand	Description	Range	Data type	Data type (label)
(J)	Target network number	1 to 239	16-bit unsigned binary	ANY16
(s1)	Target station number	1 to 64	16-bit unsigned binary	ANY16
(s2)	Target station start device where the data to be read is stored	—	String*1	ANYSTRING_SINGLE*1
(d1)	Own station start device for storing the read data (A continuous area for the read data length is required.)	—	Device name	ANY16*2
(n)	Read data length	■When the target station is AnUCPU/QnACPU/QCPU/LCPU 1 to 230 ■When the target station is a MELSEC-A series CPU module other than AnUCPU 1 to 32	16-bit unsigned binary	ANY16
(d2)	Device of the own station, which turns on for one scan upon completion of the instruction. When the instruction completes with an error, (d2)+1 also turns on.	—	Bit	ANYBIT_ARRAY (Number of elements: 2)
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

*1 For the specifications of the string data to be specified, refer to the following.
📖 MELSEC iQ-R Programming Manual (Instructions, Standard Functions/Function Blocks)

*2 When specifying setting data by using a label, define an array to secure enough operation area and specify an element of the array label.

■Applicable devices

Operand	Bit		Word			Double word		Indirect specification	Constant			Others (J)
	X, Y, M, L, SM, F, B, SB, FX, FY	J□\□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□\G□, J□\□, U3E□\(H)G□	Z	LT, LST, LC	LZ		K, H	E	$	
(J)	—	—	—	—	—	—	—	—	—	—	—	○
(s1)	○*1	—	○*2	—	—	—	—	○	○	—	—	—
(s2)	—	—	—	—	—	—	—	—	—	—	○	—
(d1)	—	—	○*2	—	—	—	—	○	—	—	—	—
(n)	○*1	—	○*2	—	—	—	—	○	○	—	—	—
(d2)	○*1	—	○*3	—	—	—	—	—	—	—	—	—

*1 FX and FY cannot be used.

*2 FD cannot be used.

*3 T, ST, C, and FD cannot be used.

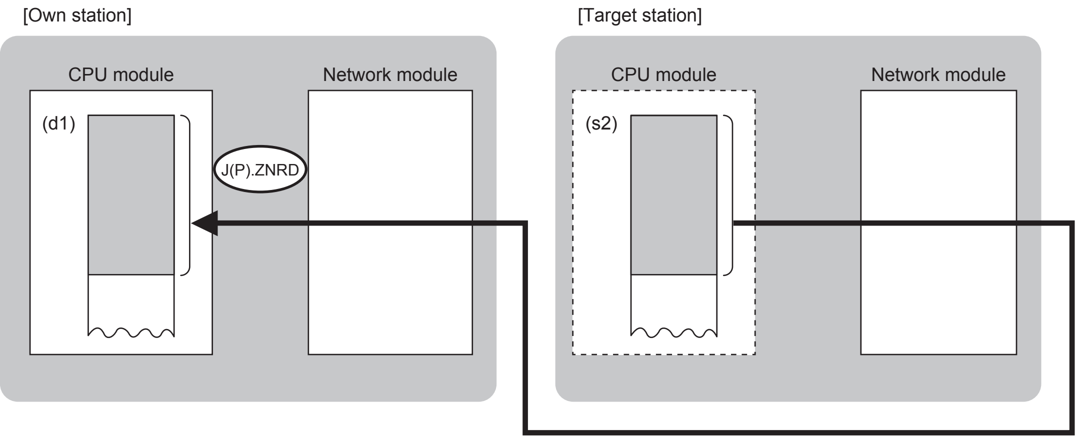
Point

The J(P).ZNRD instruction is executed by using the following fixed values in addition to setting data.

- Arrival monitoring time: 10s
- Number of resends: 5

Processing details

- These instructions read the data from the specified word device in the target station specified by the target network number (J) and target station number (s1). Upon completion of reading the device data, the completion device specified by (d2) turns on.



Point

- The J(P).ZNRD instruction cannot be executed for RCPU. If executed, the error code 4001H is stored in the completion status and the instruction completes with an error.
- The J(P).ZNRD instruction is an instruction (Q series compatible instruction) for replacement of the equivalent instruction used in the programs running on the MELSEC-Q series. When creating a new program, use the READ instruction.

- Device data can also be read from stations connected to networks other than the stations connected to the own station network.

- The execution status and the completion status of the J(P).ZNRD instruction can be checked with the completion device (d2) and the completion status indication device (d2)+1.
  - Completion device (d2)

This device turns on during END processing of the scan where the J(P).ZNRD instruction completes, and turns off during the next END processing.

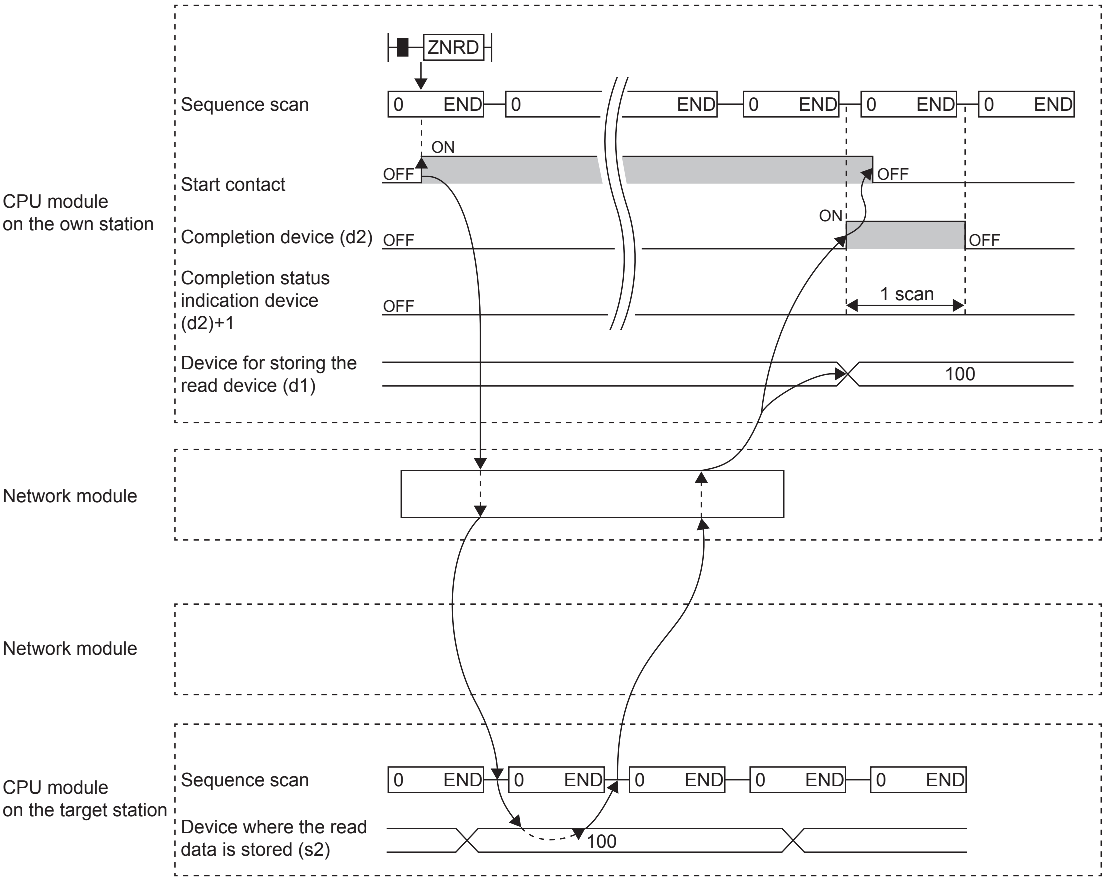
- Completion status indication device (d2)+1

This device turns on or off depending on the completion status of the J(P).ZNRD instruction.

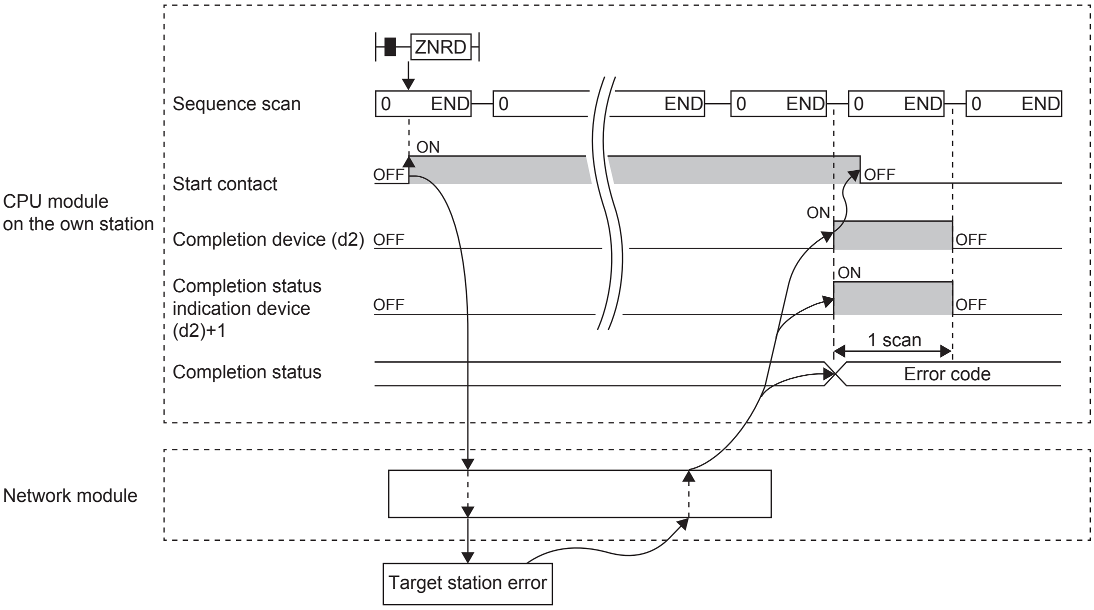
When completed successfully: The device remains off.

When completed with an error: The device turns on during END processing of the scan where the J(P).ZNRD instruction completes, and turns off during the next END processing.

- The following figure shows the execution timing of the J(P).ZNRD instruction.
  - When completed successfully


ZNRD
Sequence scan
0 END 0 END 0 END 0 END 0 END
ON
OFF
Start contact
OFF
CPU module on the own station
ON
Completion device (d2) OFF
OFF
Completion status indication device (d2)+1
OFF
1 scan
Device for storing the read device (d1)
100
Network module
Network module
CPU module on the target station
Sequence scan
0 END 0 END 0 END 0 END 0 END
Device where the read data is stored (s2)
100

- When completed with an error


ZNRD
Sequence scan
0 END 0 END 0 END 0 END 0 END
ON
OFF
Start contact
OFF
CPU module on the own station
ON
Completion device (d2) OFF
OFF
ON
Completion status indication device (d2)+1
OFF
OFF
1 scan
Completion status
Error code
Network module
Target station error

The completion status is stored in the special register (SW) when the CC-Link IE Controller Network is used or in the buffer memory when Ethernet is used.

- When the JP.ZNRD instruction is executed, read processing is performed only once on the rising edge when the read command turns on.

Precautions

The J(P).ZNRD instruction cannot be executed when the CPU module on the target station is one of the following:

- AnUCPU with the version AX (manufactured in July 1995) or earlier
- A2USCPU(-S1) with the version CN (manufactured in July 1995) or earlier

If executed, the dedicated instruction response timer causes a timeout in the instruction start source and the J(P).ZNRD instruction completes with an error. If the dedicated instruction response timer causes a timeout, use a CPU module satisfying one of the following versions.

- AnUCPU with the version AY (manufactured in July 1995) or later
- A2USCPU(-S1) with the version CP (manufactured in July 1995) or later

Operation error

Error code*4	**Description**
4000H to 4FFFH	📖 MELSEC iQ-R CPU Module User's Manual (Application)
C000H to CFFFH	📖 MELSEC iQ-R Ethernet User's Manual (Application)
E000H to EFFFH	📖 MELSEC iQ-R CC-Link IE Controller Network User's Manual (Application)

*4 The completion status in which an error code is stored is as follows.
CC-Link IE Controller Network: SW003A
Ethernet: Buffer memory address 5323 (14CBH)

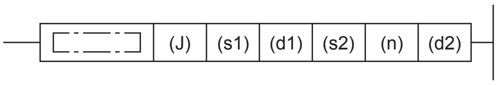
Appendix 9.4 Writing data to the programmable controller (Q series compatible)

J(P).ZNWR

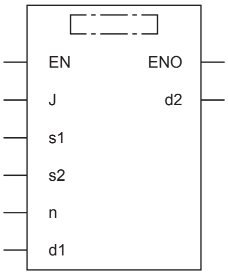
RnCPU RnENCPU RnPCPU (Process) RnPCPU (Redundant) RnSFCPU (Standard) ✕ RnSFCPU (Safety)

These instructions write data in units of words to a device in the programmable controller (MELSEC-A/QnA/Q/L series) of another station.

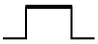
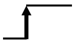
Ladder	ST

	ENO: =J_ZNWR(EN,J,s1,s2,n,d1,d2);
	ENO: =JP_ZNWR(EN,J,s1,s2,n,d1,d2);

FBD/LD



■Execution condition

Instruction	Execution condition
J.ZNWR	
JP.ZNWR	

Setting data

■Description, range, data type

Operand	Description	Range	Data type	Data type (label)
(J)	Target network number	1 to 239	16-bit unsigned binary	ANY16
(s1)	Target station number	■Station number specification 1 to 64: Station number station ■Group specification 0081H to 00A0H: All stations with group numbers 1 to 32 ■All-station specification 00FFH: All stations of target network number	16-bit unsigned binary	ANY16
(d1)	Target station start device to which data is to be written (A continuous area for the write data length is required.)	—	String*1	ANYSTRING_SINGLE*1
(s2)	Own station start device where write data is stored	—	Device name	ANY16*2
(n)	Write data length	■When the target station is AnUCPU/QnACPU/QCPU/LCPU 1 to 230 ■When the target station is a MELSEC-A series CPU module other than AnUCPU 1 to 32	16-bit unsigned binary	ANY16
(d2)	Device of the own station, which turns on for one scan upon completion of the instruction. When the instruction completes with an error, (d2)+1 also turns on.	—	Bit	ANYBIT_ARRAY (Number of elements: 2)
EN	Execution condition	—	Bit	BOOL
ENO	Execution result	—	Bit	BOOL

*1 For the specifications of the string data to be specified, refer to the following.
📖 MELSEC iQ-R Programming Manual (Instructions, Standard Functions/Function Blocks)

*2 When specifying setting data by using a label, define an array to secure enough operation area and specify an element of the array label.

■Applicable devices

Operand	Bit		Word			Double word		Indirect specification	Constant			Others (J)
	X, Y, M, L, SM, F, B, SB, FX, FY	J□\□	T, ST, C, D, W, SD, SW, FD, R, ZR, RD	U□\G□, J□\□, U3E□\(H)G□	Z	LT, LST, LC	LZ		K, H	E	$	
(J)	—	—	—	—	—	—	—	—	—	—	—	○
(s1)	○*1	—	○*2	—	—	—	—	○	○	—	—	—
(d1)	—	—	—	—	—	—	—	—	—	—	○	—
(s2)	—	—	○*2	—	—	—	—	○	—	—	—	—
(n)	○*1	—	○*2	—	—	—	—	○	○	—	—	—
(d2)	○*1	—	○*3	—	—	—	—	—	—	—	—	—

*1 FX and FY cannot be used.

*2 FD cannot be used.

*3 T, ST, C, and FD cannot be used.

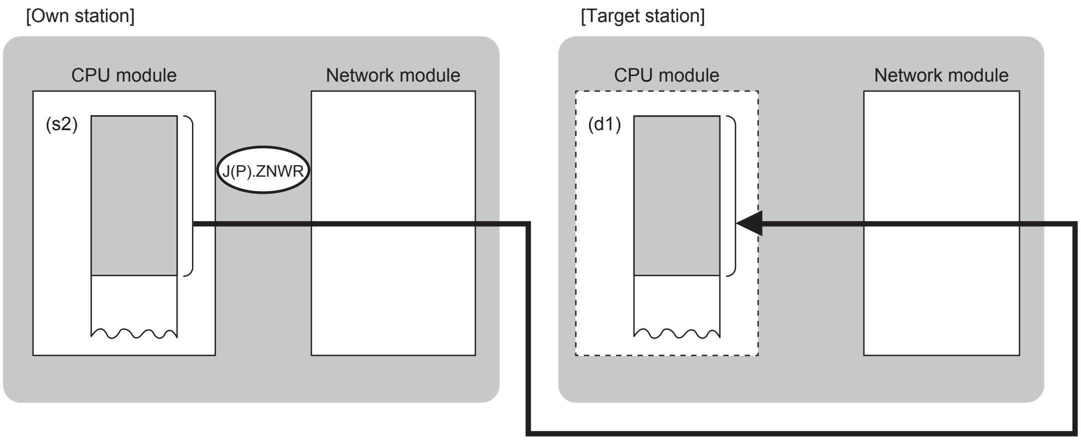
Point

The J(P).ZNWR instruction is executed by using the following fixed values in addition to setting data.

- Arrival monitoring time: 10s
- Number of resends: 5

Processing details

- These instructions write the data in the device/label specified by (s2) in the own station to the word device in the target station specified by the target network number and target station number of the control data. Upon completion of writing device data to another station number, the completion device specified by (d2) turns on.



Point

- The J(P).ZNWR instruction cannot be executed for RCPU. If executed, the error code 4001H is stored in the completion status and the instruction completes with an error.
- The J(P).ZNWR instruction is an instruction (Q series compatible instruction) for replacement of the equivalent instruction used in the programs running on the MELSEC-Q series. When creating a new program, use the WRITE instruction.

- Device data can also be written to stations connected to networks other than the stations connected to the own station network.
- The execution status and the completion status of the J(P).ZNWR instruction can be checked with the completion device (d2) and the completion status indication device (d2)+1.

- Completion device (d2)

This device turns on during END processing of the scan where the J(P).ZNWR instruction completes, and turns off during the next END processing.

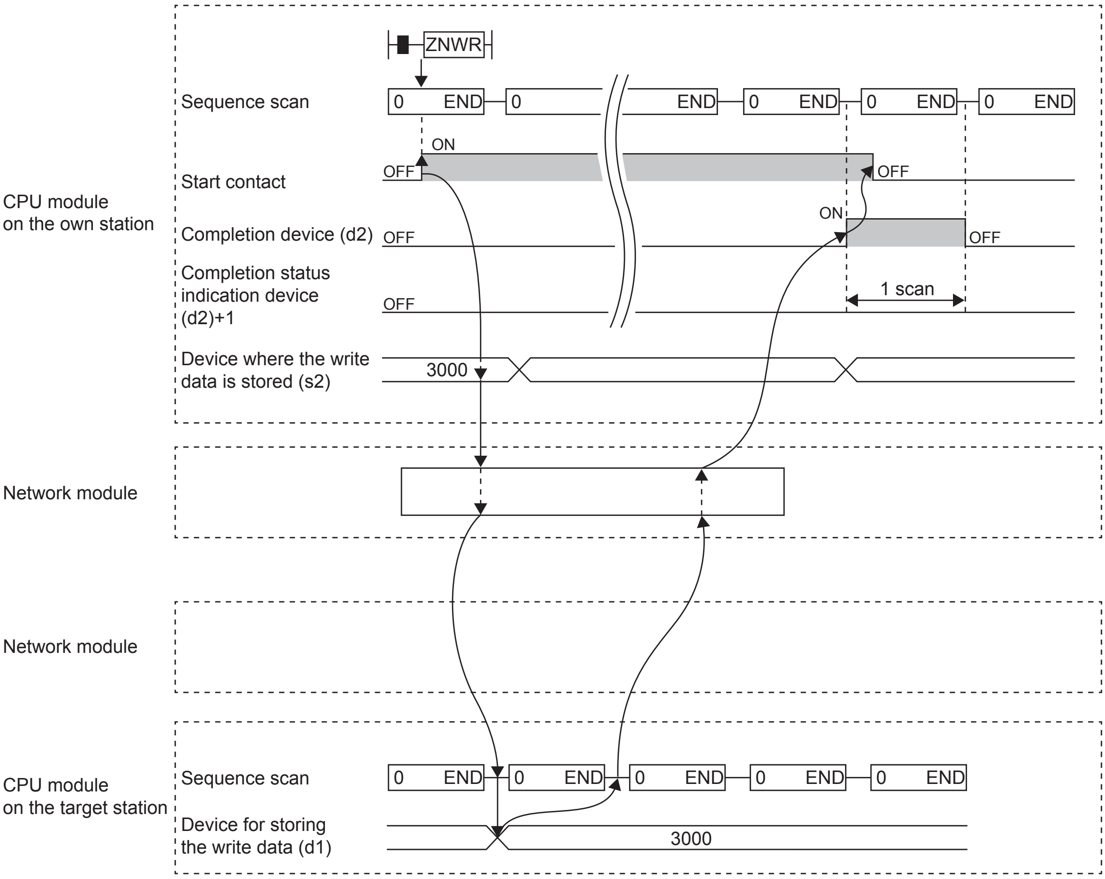
- Completion status indication device (d2)+1

This device turns on or off depending on the completion status of the J(P).ZNWR instruction.

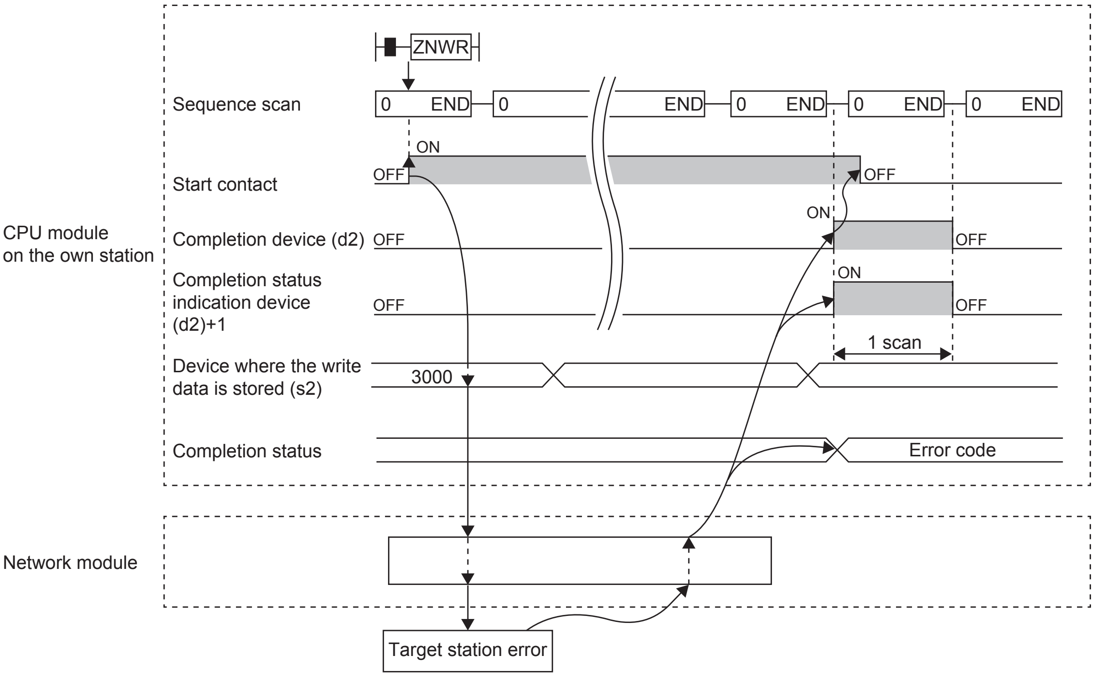
When completed successfully: The device remains off.

When completed with an error: The device turns on during END processing of the scan where the J(P).ZNWR instruction completes, and turns off during the next END processing.

- The following figure shows the execution timing of the J(P).ZNWR instruction.
  - When completed successfully



  - When completed with an error



The completion status is stored in the special register (SW) when the CC-Link IE Controller Network is used or in the buffer memory when Ethernet is used.

- When the J.ZNWR instruction is used for execution, one complete cycle of write processing is followed by another while the write command is on. When the JP.ZNWR instruction is executed, write processing is performed only once on the rising edge when the write command turns on.

Precautions

The J(P).ZNWR instruction cannot be executed for the following CPU modules.

- AnUCPU with the version AX (manufactured in July 1995) or earlier
- A2USCPU(-S1) with the version CN (manufactured in July 1995) or earlier

If executed, the dedicated instruction response timer causes a timeout in the instruction start source and the J(P).ZNWR instruction completes with an error. If the dedicated instruction response timer causes a timeout, use a CPU module satisfying one of the following versions.

- AnUCPU with the version AY (manufactured in July 1995) or later
- A2USCPU(-S1) with the version CP (manufactured in July 1995) or later

Operation error

Error code[*4]	Description
4000H to 4FFFH	📖 MELSEC iQ-R CPU Module User's Manual (Application)
C000H to CFFFH	📖 MELSEC iQ-R Ethernet User's Manual (Application)
E000H to EFFFH	📖 MELSEC iQ-R CC-Link IE Controller Network User's Manual (Application)

*4 The completion status in which an error code is stored is as follows.
CC-Link IE Controller Network: SW003B
Ethernet: Buffer memory address 5325 (14CDH)

Appendix 10 Examples

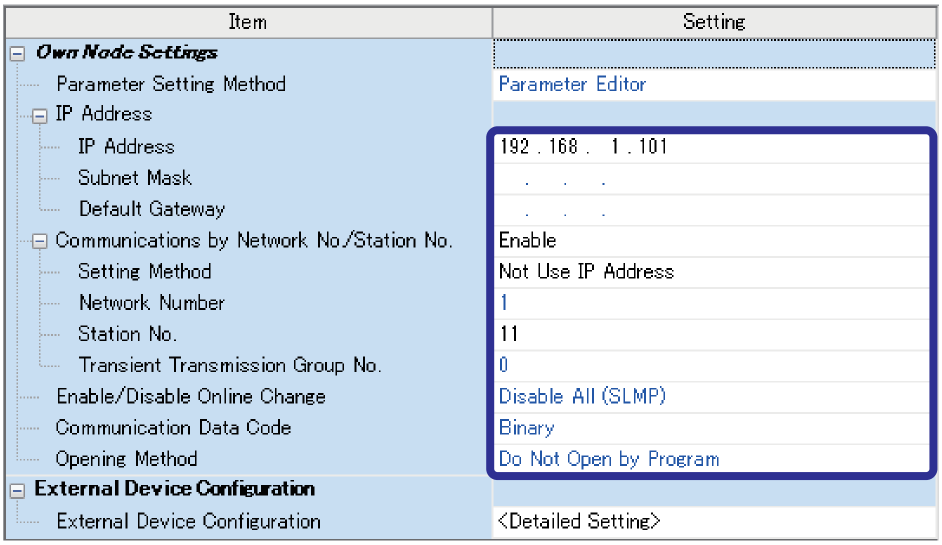
Appendix 10.1 Parameter settings

This section provides the module parameter setting examples.

For the setting operation, refer to Page 5 - 8 Setting parameters with GX Works3.

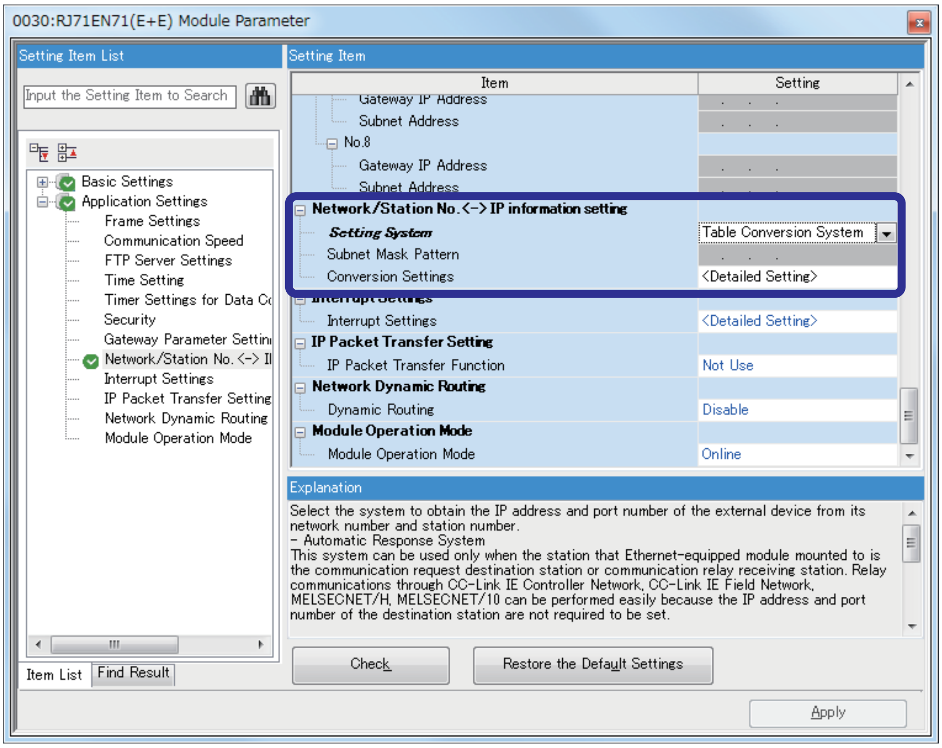
Parameter settings of the CPU module A

■Basic Settings



Item	Setting
Own Node Settings	
Parameter Setting Method	Parameter Editor
IP Address	
IP Address	192 . 168 . 1 . 101
Subnet Mask	. . .
Default Gateway	. . .
Communications by Network No./Station No.	Enable
Setting Method	Not Use IP Address
Network Number	1
Station No.	11
Transient Transmission Group No.	0
Enable/Disable Online Change	Disable All (SLMP)
Communication Data Code	Binary
Opening Method	Do Not Open by Program
External Device Configuration	
External Device Configuration	<Detailed Setting>

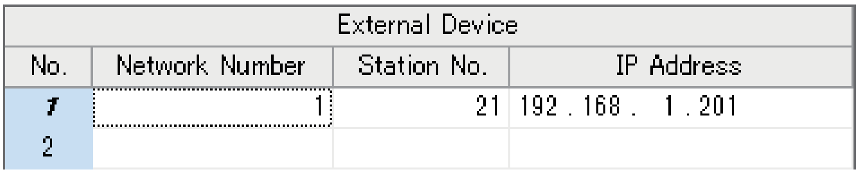
■Application Settings



Item	Setting
Gateway IP Address	. . .
Subnet Address	. . .
No.8	
Gateway IP Address	. . .
Subnet Address	. . .
Network/Station No.<-> IP information setting	
Setting System	Table Conversion System
Subnet Mask Pattern	. . .
Conversion Settings	<Detailed Setting>
Interrupt Settings	
Interrupt Settings	<Detailed Setting>
IP Packet Transfer Setting	
IP Packet Transfer Function	Not Use
Network Dynamic Routing	
Dynamic Routing	Disable
Module Operation Mode	
Module Operation Mode	Online

Select the system to obtain the IP address and port number of the external device from its network number and station number.
- Automatic Response System
This system can be used only when the station that Ethernet-equipped module mounted to is the communication request destination station or communication relay receiving station. Relay communications through CC-Link IE Controller Network, CC-Link IE Field Network, MELSECNET/H, MELSECNET/10 can be performed easily because the IP address and port number of the destination station are not required to be set.

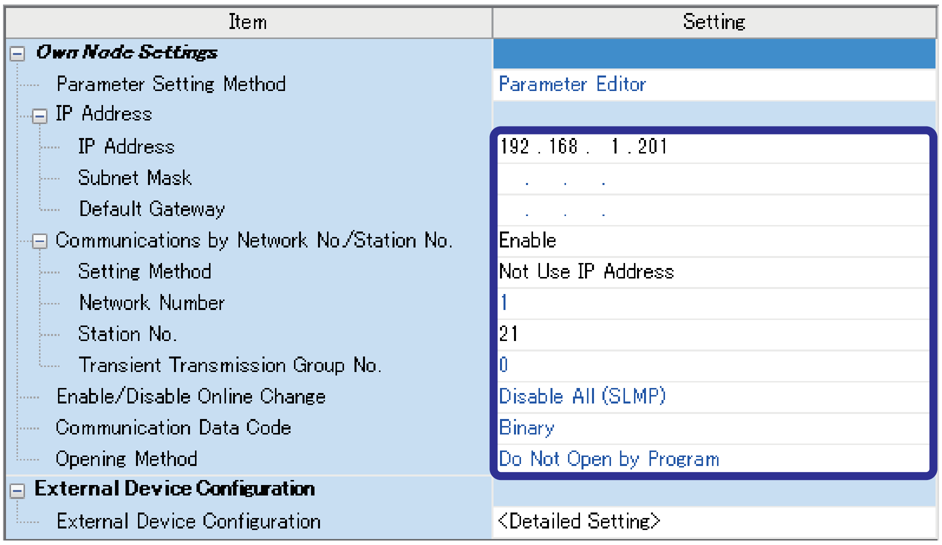
- Conversion settings



	External Device		
No.	Network Number	Station No.	IP Address
1	1	21	192 . 168 . 1 . 201
2			

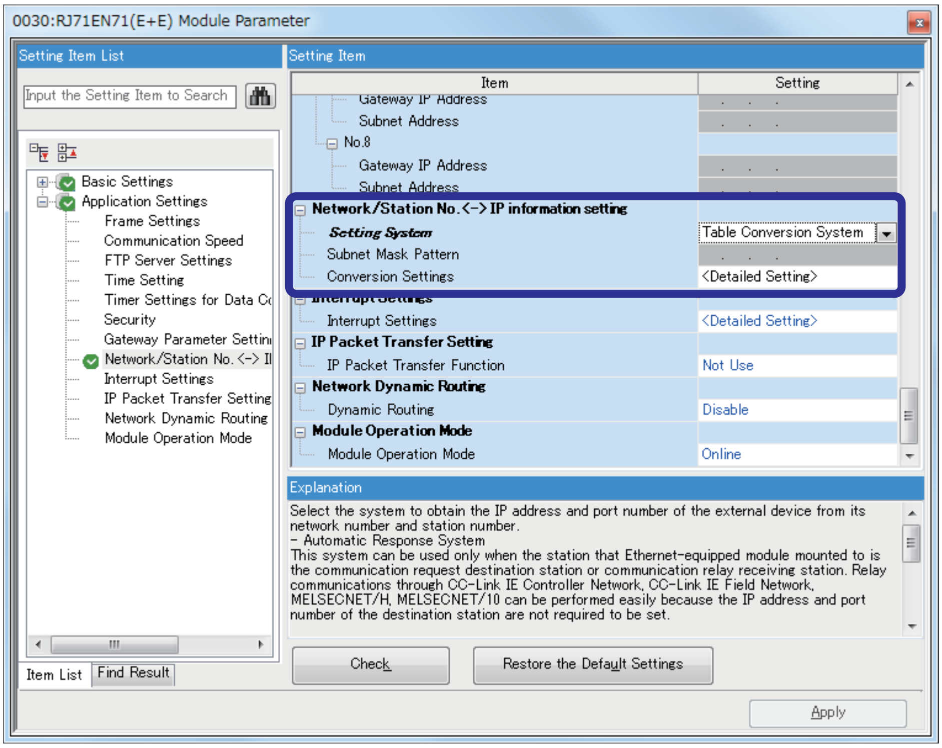
Parameter settings of the CPU module B

■Basic Settings



Item	Setting
Own Node Settings	
Parameter Setting Method	Parameter Editor
IP Address	
IP Address	192 . 168 . 1 . 201
Subnet Mask	. . .
Default Gateway	. . .
Communications by Network No./Station No.	Enable
Setting Method	Not Use IP Address
Network Number	1
Station No.	21
Transient Transmission Group No.	0
Enable/Disable Online Change	Disable All (SLMP)
Communication Data Code	Binary
Opening Method	Do Not Open by Program
External Device Configuration	
External Device Configuration	<Detailed Setting>

■Application Settings

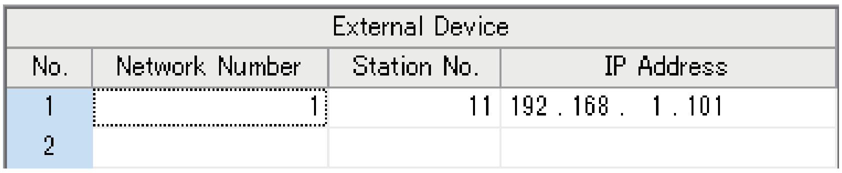


Item	Setting
Gateway IP Address	. . .
Subnet Address	. . .
No.8	
Gateway IP Address	. . .
Subnet Address	. . .
Network/Station No.<-> IP information setting	
Setting System	Table Conversion System
Subnet Mask Pattern	. . .
Conversion Settings	<Detailed Setting>
Interrupt Settings	
Interrupt Settings	<Detailed Setting>
IP Packet Transfer Setting	
IP Packet Transfer Function	Not Use
Network Dynamic Routing	
Dynamic Routing	Disable
Module Operation Mode	
Module Operation Mode	Online

Explanation

Select the system to obtain the IP address and port number of the external device from its network number and station number.
- Automatic Response System
This system can be used only when the station that Ethernet-equipped module mounted to is the communication request destination station or communication relay receiving station. Relay communications through CC-Link IE Controller Network, CC-Link IE Field Network, MELSECNET/H, MELSECNET/10 can be performed easily because the IP address and port number of the destination station are not required to be set.

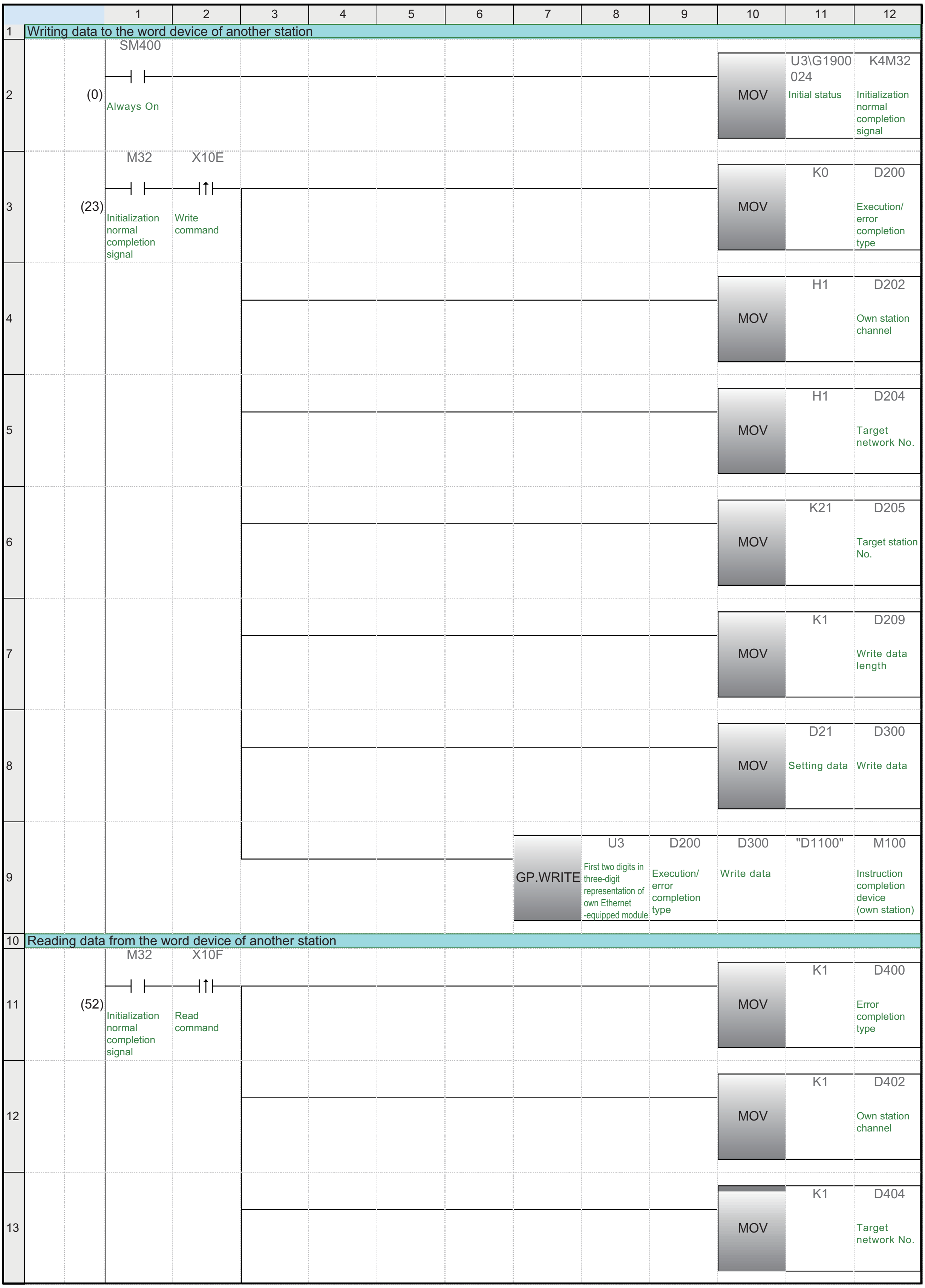
- Conversion settings



	External Device		
No.	Network Number	Station No.	IP Address
1	1	11	192 . 168 . 1 . 101
2			

Appendix 10.2 Sequence programs

Project name	EX5-A

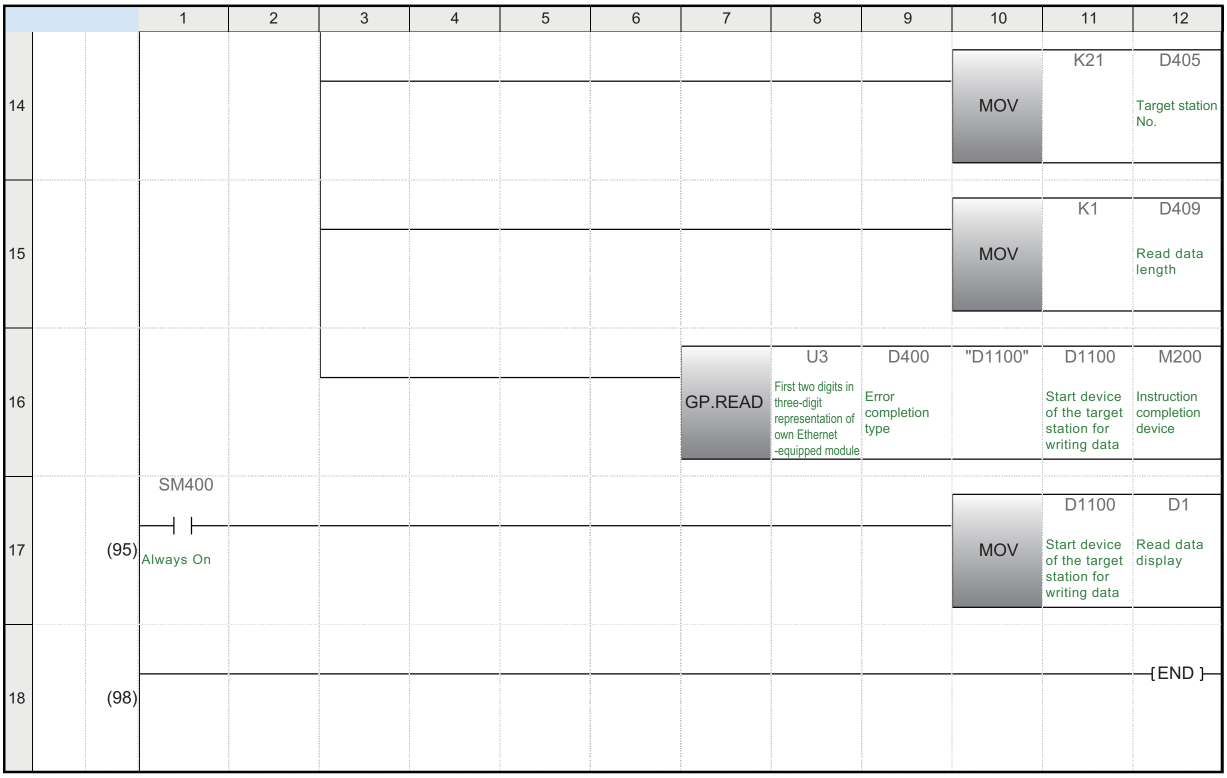


1 Writing data to the word device of another station

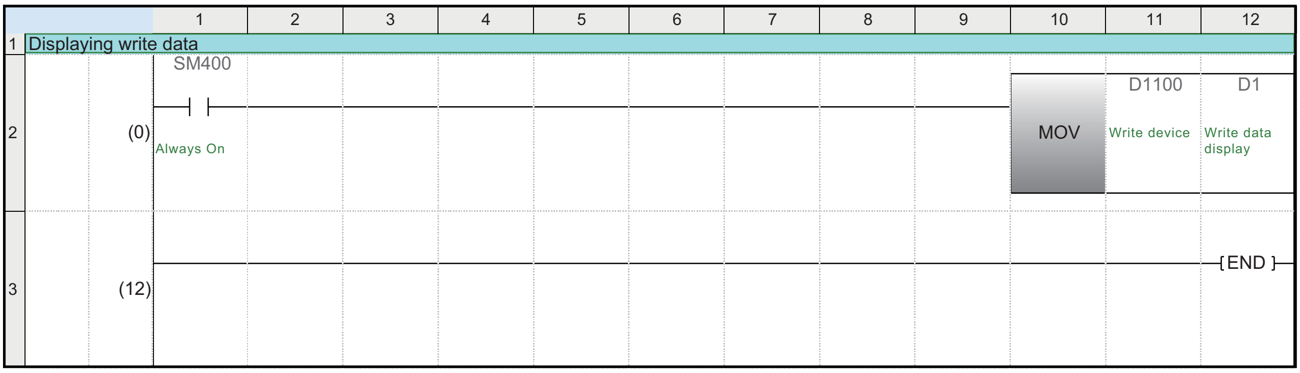
```
2   (0)   SM400 (Always On) ----------------------------- MOV U3\G1900024 (Initial status) K4M32 (Initialization normal completion signal)
3   (23)  M32 (Initialization normal completion signal)  X10E↑ (Write command)
                                                    ----- MOV K0   D200 (Execution/error completion type)
4                                                   ----- MOV H1   D202 (Own station channel)
5                                                   ----- MOV H1   D204 (Target network No.)
6                                                   ----- MOV K21  D205 (Target station No.)
7                                                   ----- MOV K1   D209 (Write data length)
8                                                   ----- MOV D21 (Setting data)  D300 (Write data)
9                                                   ----- GP.WRITE U3 (First two digits in three-digit representation of own Ethernet-equipped module) D200 (Execution/error completion type) D300 (Write data) "D1100" M100 (Instruction completion device (own station))
```

10 Reading data from the word device of another station

```
11  (52)  M32 (Initialization normal completion signal)  X10F↑ (Read command)
                                                    ----- MOV K1   D400 (Error completion type)
12                                                  ----- MOV K1   D402 (Own station channel)
13                                                  ----- MOV K1   D404 (Target network No.)
```


1
2
3
4
5
6
7
8
9
10
11
12
14
MOV
K21
D405
Target station No.
15
MOV
K1
D409
Read data length
16
GP.READ
U3
First two digits in three-digit representation of own Ethernet -equipped module
D400
Error completion type
"D1100"
D1100
Start device of the target station for writing data
M200
Instruction completion device
17
(95)
SM400
Always On
MOV
D1100
Start device of the target station for writing data
D1
Read data display
18
(98)
END

Project name	EX5-B



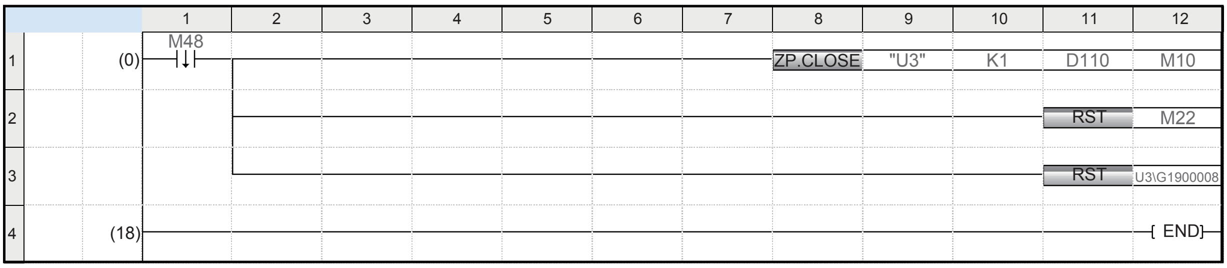
Appendix 10.3 Operation of the demonstration machine

Check that the CPU module A reads/writes data from/to the CPU module B by using dedicated instructions.

Operation method

1. Write parameters and programs to the CPU module A and CPU module B.
2. Set the RUN/STOP/RESET switch of the CPU module A and CPU module B to the "RESET" position (for approximately one second) to reset the CPU modules. Then, move it to the "RUN" position.
3. Input a numerical value in the initial input device (D21) of the CPU module A.
   Turn on X10E of the CPU module A to send data.
4. The numerical value input in the initial input device (D21) of the CPU module A is displayed in the initial indication device (D1) of the CPU module B.
5. After checking the value, turn off X10E of the CPU module A and end the send processing.
6. Receive data in the initial indication device (D1) of the CPU module B.
   Turn on X10F of the CPU module A to receive data.
7. The numerical value in the initial indication device (D1) of the CPU module B is displayed in the initial indication device (D1) of the CPU module A.
8. After checking the value, turn off X10F of the CPU module A and end the receive processing.

Appendix 11 Answers for the Additional Exercise 1 (Section 6.6.1)



```
  1  (0)   M48
           -|↓|-+---------------------------------- ZP.CLOSE "U3" K1 D110 M10
  2             |
                +---------------------------------- RST M22
  3             |
                +---------------------------------- RST U3\G1900008
  4  (18) ------------------------------------------[ END]
```

MEMO

Mitsubishi Programmable Controllers Training Manual Ethernet (for GX Works3)

MODEL	
MODEL CODE	
SH(NA)081820ENG-A (1704) MEE	

MITSUBISHI ELECTRIC CORPORATION

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

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

Specifications subject to change without notice.