

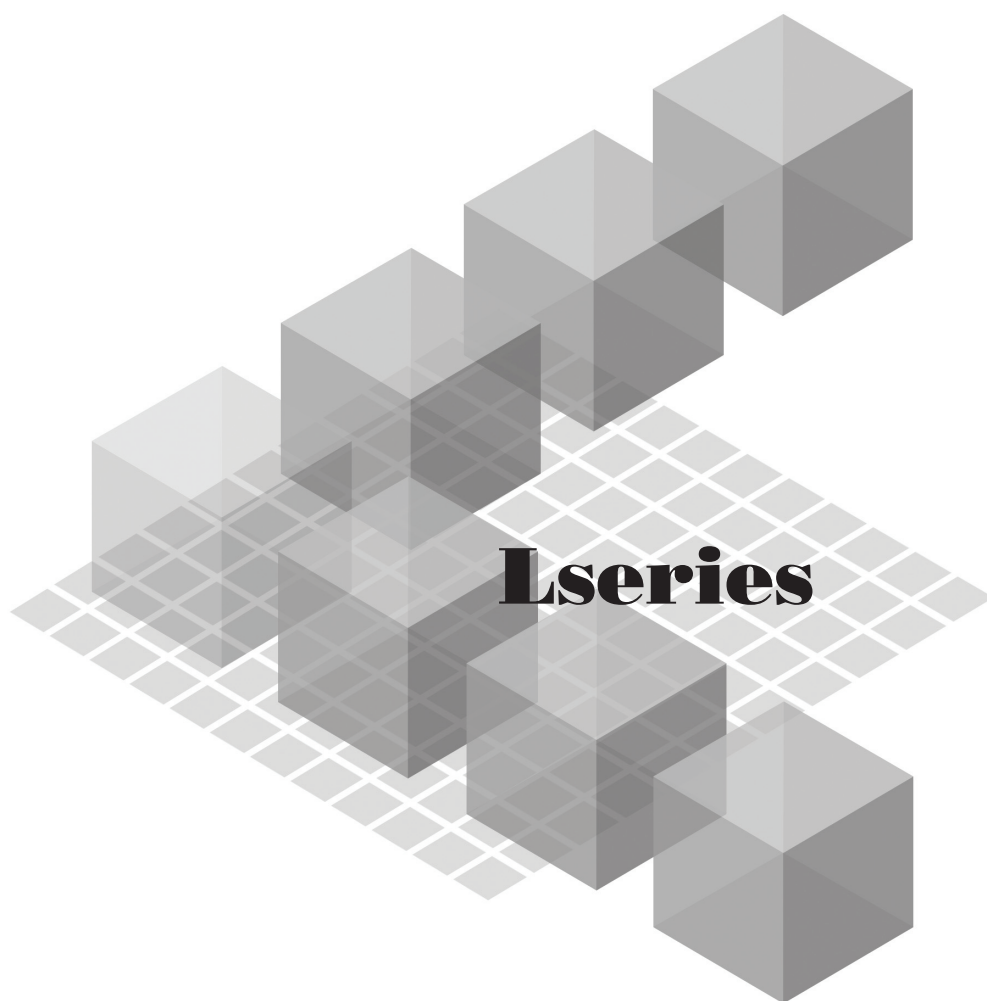
MITSUBISHI

Mitsubishi Programmable Controller

MELSEC *L*series

MELSEC-L Ethernet Interface Module User's Manual

Basic



-LJ71E71-100

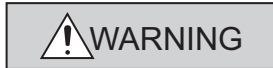
MODEL

● SAFETY PRECAUTIONS ●

(Read these precautions before using this product.)

Before using this product, please read this manual and the relevant manuals carefully and pay full attention to safety to handle the product correctly.

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



Indicates that incorrect handling may cause hazardous conditions, resulting in death or severe injury.



Indicates that incorrect handling may cause hazardous conditions, resulting in minor or moderate injury or property damage.

Under some circumstances, failure to observe the precautions given under "⚠ CAUTION" may lead to serious consequences.

Observe the precautions of both levels because they are important for personal and system safety. Make sure that the end users read this manual and then keep the manual in a safe place for future reference.

[Design Precautions]

WARNING

- For the operating status of each station after a communication failure, refer to relevant manuals for each network. Incorrect output or malfunction due to a communication failure may result in an accident.
- To prevent the malfunction of the programmable controller system due to harmful e-mails, take preventive measures (such as antivirus measures) so that the mail server for this module does not receive harmful e-mails.
- To maintain the safety of the programmable controller system against unauthorized access from external devices via the Internet, take appropriate measures.
- When connecting a peripheral with the CPU module or connecting an external device, such as a personal computer, with an intelligent function module to modify data of a running programmable controller, configure an interlock circuit in the program to ensure that the entire system will always operate safely. For other forms of control (such as program modification or operating status change) of a running programmable controller, read the relevant manuals carefully and ensure that the operation is safe before proceeding. Especially, when a remote programmable controller is controlled by an external device, immediate action cannot be taken if a problem occurs in the programmable controller due to a communication failure. To prevent this, configure an interlock circuit in the program, and determine corrective actions to be taken between the external device and CPU module in case of a communication failure.
- Do not write any data to the "system area" of the buffer memory in the intelligent function module. Also, do not use any "use prohibited" signal as an output signal from the CPU module to the intelligent function module. Doing so may cause malfunction of the programmable controller system.

[Design Precautions]

CAUTION

- Do not install the control lines or communication cables together with the main circuit lines or power cables. Keep a distance of 100mm or more between them. Failure to do so may result in malfunction due to noise.
- When changing the operating status of the CPU module (such as remote RUN/STOP) from the external device, select "Always wait for OPEN (Communication possible at STOP time)" for the "Initial timing" setting in the network parameter. The communication line will be closed when "Do not wait for OPEN (Communications impossible at STOP time)" is selected and the remote STOP is executed from the external device. Consequently, the CPU module cannot reopen the communication line, and the external device cannot execute the remote RUN.

[Installation Precautions]

CAUTION

- Use the programmable controller in an environment that meets the general specifications in the Safety Guidelines included with the MELSEC-L series CPU module. Failure to do so may result in electric shock, fire, malfunction, or damage to or deterioration of the product.
- To interconnect modules, engage the respective connectors and securely lock the module joint levers until they click. Incorrect interconnection may cause malfunction, failure, or drop of the module.
- Shut off the external power supply (all phases) used in the system before mounting or removing a module. Failure to do so may result in damage to the product.
- Do not directly touch any conductive parts and electronic components of the module. Doing so can cause malfunction or failure of the module.

[Wiring Instructions]

CAUTION

- When connecting a cable with connector to the module, connect the connector part to the module securely.
- Place the cables in a duct or clamp them. If not, dangling cable may swing or inadvertently be pulled, resulting in damage to the module or cables or malfunction due to poor contact.
- When disconnecting the communication and power cables from the module, do not pull the cables by hand. For the cable with connector, hold the connector part of the cable. Pulling the cable connected to the module may result in malfunction or damage to the module or cable.
- Prevent foreign matter such as dust or wire chips from entering the module. Such foreign matter can cause a fire, failure, or malfunction.
- A protective film is attached to the top of the module to prevent foreign matter, such as wire chips, from entering the module during wiring. Do not remove the film during wiring. Remove it for heat dissipation before system operation.

[Setup and Maintenance Precautions]

WARNING

- Shut off the external power supply (all phases) used in the system before cleaning the module. Failure to do so may result in electric shock.

[Setup and Maintenance Precautions]

CAUTION

- Do not disassemble or modify the module. Doing so may cause failure, malfunction, injury, or a fire.
- Shut off the external power supply (all phases) used in the system before connecting or disconnecting a module. Failure to do so may cause the module to fail or malfunction.
- After the first use of the product (module and display unit), the number of connections/disconnections is limited to 50 times (in accordance with IEC 61131-2). Exceeding the limit may cause malfunction.
- Before handling the module, touch a conducting object such as a grounded metal to discharge the static electricity from the human body. Failure to do so may cause the module to fail or malfunction.

[Operating Precautions]

CAUTION

- When changing data and operating status, and modifying program of the running programmable controller from a personal computer connected to an intelligent function module, read relevant manuals carefully and ensure the safety before operation. Incorrect change or modification may cause system malfunction, damage to the machines, or accidents.

[Disposal Precautions]

CAUTION

- When disposing of this product, treat it as industrial waste.

● CONDITIONS OF USE FOR THE PRODUCT ●

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

MITSUBISHI SHALL HAVE NO RESPONSIBILITY OR LIABILITY (INCLUDING, BUT NOT LIMITED TO ANY AND ALL RESPONSIBILITY OR LIABILITY BASED ON CONTRACT, WARRANTY, TORT, PRODUCT LIABILITY) FOR ANY INJURY OR DEATH TO PERSONS OR LOSS OR DAMAGE TO PROPERTY CAUSED BY the PRODUCT THAT ARE OPERATED OR USED IN APPLICATION NOT INTENDED OR EXCLUDED BY INSTRUCTIONS, PRECAUTIONS, OR WARNING CONTAINED IN MITSUBISHI'S USER, INSTRUCTION AND/OR SAFETY MANUALS, TECHNICAL BULLETINS AND GUIDELINES FOR the PRODUCT.

("Prohibited Application")

Prohibited Applications include, but not limited to, the use of the PRODUCT in;

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

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

INTRODUCTION

Thank you for purchasing the Mitsubishi MELSEC-L series programmable controllers. This manual describes the operating procedure, system configuration, parameter settings, functions, programming, and troubleshooting of the LJ71E71-100 Ethernet interface module (hereafter referred to as E71).


Before using this product, please read this manual and the relevant manuals carefully and develop familiarity with the functions and performance of the MELSEC-L series programmable controller to handle the product correctly. When applying the program examples introduced in this manual to the actual system, ensure the applicability and confirm that it will not cause system control problems.

Please make sure that the end users read this manual.

Remark

Unless otherwise specified, this manual describes the program examples in which the I/O numbers of X/Y00 to X/Y1F are assigned for an E71.

For I/O number assignment, refer to the following.

 MELSEC-L CPU Module User's Manual (Function Explanation, Program Fundamentals)

COMPLIANCE WITH EMC AND LOW VOLTAGE DIRECTIVES

(1) Method of ensuring compliance

To ensure that Mitsubishi programmable controllers maintain EMC and Low Voltage Directives when incorporated into other machinery or equipment, certain measures may be necessary. Please refer to one of the following manuals.

- MELSEC-L CPU Module User's Manual (Hardware Design, Maintenance and Inspection)
- Safety Guidelines (This manual is included with the CPU module.)

The CE mark on the side of the programmable controller indicates compliance with EMC and Low Voltage Directives.

(2) Additional measures

To ensure that this product maintains EMC and Low Voltage Directives, please refer to one of the manuals listed under (1).

RELEVANT MANUALS

The manuals related to this product are listed below. Please place an order as needed.

(1) E71 (relevant) manuals

Manual name <manual number, model code>	Description
MELSEC-Q/L Ethernet Interface Module User's Manual (Application) <SH-080010, 13JL89>	The e-mail function, CPU module monitoring function, communications over CC-Link IE Controller Network, CC-Link IE Field Network, MELSECNET/H, or MELSECNET/10, communications using data link instructions, and file transfer function (FTP server) of the E71
MELSEC-Q/L Ethernet Interface Module User's Manual (Web function) <SH-080180, 13JR40>	The Web function of the E71
MELSEC-Q/L MELSEC Communication Protocol Reference Manual <SH-080008, 13JF89>	Communication methods and control procedures through the MC protocol used for data communications between a connected device and a CPU module using the C24/E71

(2) CPU module user's manuals

Manual name <manual number, model code>	Description
MELSEC-L CPU Module User's Manual (Hardware Design, Maintenance and Inspection) <SH-080890ENG, 13JZ36>	Specifications of the CPU modules, power supply modules, display unit, branch module, extension module, SD memory cards, and batteries, information on how to establish a system, maintenance and inspection, and troubleshooting
MELSEC-L CPU Module User's Manual (Function Explanation, Program Fundamentals) <SH-080889ENG, 13JZ35>	Functions, programming methods, and devices of the CPU module

(3) Operating manual

Manual name <manual number, model code>	Description
GX Works2 Version 1 Operating Manual (Common) <SH-080779ENG, 13JU63>	System configuration, parameter settings, and online operations (common to Simple project and Structured project) of GX Works2

CONTENTS

SAFETY PRECAUTIONS	1
CONDITIONS OF USE FOR THE PRODUCT	5
INTRODUCTION	6
COMPLIANCE WITH EMC AND LOW VOLTAGE DIRECTIVES	6
RELEVANT MANUALS	7
MANUAL PAGE ORGANIZATION	13
TERM.	16
PACKING LIST	18
<hr/>	
CHAPTER 1 FEATURES	19
<hr/>	
CHAPTER 2 PART NAMES	24
<hr/>	
CHAPTER 3 SPECIFICATIONS	26
<hr/>	
3.1 General Specifications	26
3.2 Performance Specifications	26
3.3 Function List	28
3.3.1 Function list	28
3.3.2 Use with other functions	29
3.4 List of I/O Signals	30
3.5 Buffer Memory	32
3.5.1 Configuration of the buffer memory	32
3.5.2 List of buffer memory addresses	33
<hr/>	
CHAPTER 4 PROCEDURES BEFORE OPERATION	54
<hr/>	
CHAPTER 5 SYSTEM CONFIGURATION	56
<hr/>	
5.1 Overall System Configuration	56
5.2 Network Components	57
5.3 Applicable Systems	58
<hr/>	
CHAPTER 6 INSTALLATION AND WIRING	59
<hr/>	
6.1 Installation	59
6.2 Wiring	60
<hr/>	
CHAPTER 7 COMMUNICATION PROCEDURE	62
<hr/>	
7.1 Setting Parameters Required for Communications	63
7.1.1 Parameter list	63
7.1.2 Basic setting	64
7.1.3 Ethernet Operation Setting	65
7.1.4 Open Setting	67
7.2 TCP/IP Communications	70
7.2.1 Establishing a connection	70
7.2.2 Communication process	71
7.2.3 Active open procedure	73

7.2.4	Passive open procedure	75
7.3	UDP/IP Communications	78
7.3.1	Communication process	78
7.3.2	Open procedure	79

CHAPTER 8	CONNECTING MELSOFT PRODUCTS AND A GOT	81
------------------	--	-----------

8.1	Applications	81
8.2	Data Communication Procedure	82

CHAPTER 9	MC PROTOCOL COMMUNICATIONS	84
------------------	-----------------------------------	-----------

9.1	Applications	84
9.2	Communication Structure	85
9.3	Data Communication Procedure	85
9.4	Parameter Setting	86

CHAPTER 10	COMMUNICATIONS USING A FIXED BUFFER	87
-------------------	--	-----------

10.1	Applications	87
10.1.1	Differences between the "Procedure Exist" and "No Procedure" control methods	87
10.2	Communication Structure	88
10.3	Data Sending Procedure	90
10.4	Data Receiving Procedure	92
10.4.1	Data receiving using the main program (BUFRCV instruction)	92
10.4.2	Data receiving using an interrupt program (BUFRCVS instruction)	95
10.5	Parameter Setting	97
10.5.1	Parameter setting when using an interrupt program	98
10.6	Data Format	100
10.6.1	Header	100
10.6.2	Application data	100
10.7	Pairing Open	105
10.7.1	Applications	105
10.7.2	Parameter setting	106
10.8	Broadcast Communications	107
10.8.1	Sending/receiving procedures	107
10.8.2	Parameter setting	109
10.8.3	Precautions	111
10.9	Example of Communications Using a Fixed Buffer	112
10.9.1	System configuration	112
10.9.2	Parameter setting	112
10.9.3	Program	115

CHAPTER 11	COMMUNICATIONS USING A RANDOM ACCESS BUFFER	120
-------------------	--	------------

11.1	Applications	120
11.2	Communication Structure	121
11.2.1	How data is read from a connected device	122

11.2.2	How a connected device writes data	122
11.3	Parameter Setting	123
11.4	Data Format	124
11.4.1	Header	124
11.4.2	Application data	125
11.4.3	Examples of command and response formats	129
11.5	Precautions when Creating Programs	133
11.6	Physical and Logical Addresses of a Random Access Buffer	133
11.7	Example of Communications Using a Random Access Buffer	134

CHAPTER 12 OTHER FUNCTIONS	135
-----------------------------------	------------

12.1	Router Relay Function	135
12.1.1	Applications	135
12.1.2	Parameter settings	135
12.2	Communications Using an Auto-open UDP Port	139
12.2.1	Application	139
12.3	Remote Password	140
12.3.1	Application	140
12.3.2	Remote password setting processes (unlock and lock processes)	141
12.3.3	Remote password check procedure	142
12.3.4	Differences in functions according to the remote password check status (enabled/disabled)	145
12.3.5	Precautions	146
12.3.6	Parameter settings	148
12.4	Hub Connection Status Monitor Function	150
12.5	IP Address in Use Detection Function	151
12.6	Alive Check Function	153

CHAPTER 13 DEDICATED INSTRUCTIONS	155
--	------------

13.1	List of Dedicated Instructions	155
13.2	Parameter Settings for Using Dedicated Instructions	156
13.2.1	When using data link instructions	156
13.3	Precautions for Dedicated Instructions	156
13.4	Organization of the Dedicated Instruction Sections	157
13.5	ZP.OPEN	158
13.6	ZP.CLOSE	162
13.7	ZP.BUFSND	165
13.8	ZP.BUFRVCV	169
13.9	Z.BUFRCVS	173
13.10	ZP.ERRCLR	175
13.11	ZP.ERRRD	179
13.12	ZP.UINI	182

CHAPTER 14 TROUBLESHOOTING **189**

14.1	Before Troubleshooting	189
14.2	Troubleshooting Procedure	189
14.3	Checking with the Module Error Collection Function	190
14.4	Checking the LEDs	191
14.4.1	If the RUN LED turns off	191
14.4.2	If the ERR. LED or COM.ERR. LED turns on	191
14.4.3	If the SD/RD LED does not flash when data is sent.	192
14.5	Checking with a Display Unit	193
14.6	Troubleshooting by Symptom	194
14.6.1	Communications cannot be performed with the connected device.	194
14.6.2	The E71 frequently fails to receive a message sent from the connected device.	195
14.6.3	A dedicated instruction is not completed.	195
14.6.4	MC protocol communications cannot be performed.	196
14.6.5	Data cannot be sent with communications using a fixed buffer.	197
14.6.6	Data cannot be received with communications using a fixed buffer.	198
14.6.7	Communications using a random access buffer cannot be performed.	199
14.6.8	An e-mail cannot be sent.	200
14.6.9	An e-mail cannot be received.	201
14.6.10	Communications using data link instructions cannot be performed.	202
14.7	Error Code List	203
14.7.1	End codes returned to a connected device during data communications	210
14.7.2	Abnormal codes returned during communications using an A-compatible 1E frame.	213
14.7.3	Error codes stored in the buffer memory	214
14.8	Ethernet Diagnostics	237
14.9	Checking the Status of the E71 with the System Monitor	238
14.10	How to Turn Off the COM.ERR. LED	240

APPENDICES **241**

Appendix 1	Processing Time	241
Appendix 2	Port Numbers Used for the E71	245
Appendix 3	Initial Process	246
Appendix 3.1	Setting the initial process.	246
Appendix 3.2	Reinitialization process	250
Appendix 4	Line Status Check	256
Appendix 4.1	PING test	256
Appendix 4.2	Loopback test	261
Appendix 5	Self-Diagnostic Tests	265
Appendix 5.1	Self-loopback test	265
Appendix 5.2	Hardware test (H/W Test)	266
Appendix 6	Differences from Ethernet Modules of Other Series	267
Appendix 6.1	Differences from a built-in Ethernet port LCPU	267
Appendix 6.2	Differences from Q series modules	267
Appendix 6.3	Differences from QnA/A series modules	267

Appendix 7 Sample Program on the Connected Device Side	271
Appendix 7.1 When Visual C++®.NET is used	273
Appendix 7.2 When Visual Basic®.NET is used	282
Appendix 8 Checking the Serial Number, Function Version, and MAC address	291
Appendix 9 External Dimension Diagram	292
Appendix 10 ASCII Code List	293

INDEX	294
REVISIONS	296
WARRANTY	297

MANUAL PAGE ORGANIZATION

In this manual, pages are organized and the symbols are used as shown below.

The following illustration is for explanation purpose only, and should not be referred to as an actual documentation.

Annotations for the manual page screenshot:

- ""** is used for window names and items.
- 1.** shows operating procedures.
- shows mouse operations.*1
- []** is used for items in the menu bar and the project window.
- [Ex.]** shows setting or operating examples.
- shows reference manuals.
- shows reference pages.
- Point** shows notes that requires attention.
- Remark** shows useful information.

Page content details:

CHAPTER 7. VARIOUS SETTINGS

7.1.1 Setting method

(1) Setting parameters

(a) Operating procedure

1. Open the "PLC Parameter" dialog box.
Project window → [Parameter] → [PLC parameter]
2. Select the "IO Assignment" tab.

Item	Description	Reference
Type	Select the type of the connected module.	Page 74, Section 7.1.2
Model Name	Select the model name of the connected module.	Page 74, Section 7.1.3
Points	Set the number of points assigned to each slot.	Page 74, Section 7.1.4
Start XY	Specify a start I/O number for each slot.	Page 74, Section 7.1.5
Switch setting	Configure the switch setting of the built-in I/O or intelligent function modules.	Page 74, Section 7.1.6
Default setting	Set the following: - Error Time Output Mode - PLC Operation Mode at HW Error - I/O Response Time	Page 75, Section 7.1.7

Setting "Start XY" enables modification on the start I/O numbers assigned to connected modules.
[Ex.] When "1000" is specified in "Start XY" to the slot where a 16-point module is connected, the assignment range of an input module is changed to X1000 to X100F.
 For details, refer to the following:
 MELSEC-L CPU Module User's Manual (Function Explanation, Program Fundamentals)

Point
 Set the type of the connected module in "Type". Setting a different type results in "SP.LIMIT LAV ERR."
 For the intelligent function module, the I/O points must also be the same in addition to the I/O assignment setting.
 (Page 30, Section 4.2.2)

Remark
 When an intelligent module is connected, I/O assignment can be omitted by selecting connected modules from "Intelligent Function Module" in the Project window.

*1 The mouse operation example is provided below.

Annotations for the software interface screenshot:

- Menu bar**
[Ex.] [Online] → [Write to PLC...]
 Select [Online] on the menu bar, and then select [Write to PLC...].
- A window selected in the view selection area is displayed.**
[Ex.] Project window → [Parameter] → [PLC Parameter]
 Select [Project] from the view selection area to open the Project window. In the Project window, expand [Parameter] and select [PLC Parameter].
- View selection area**

Pages describing instructions are organized as shown below.

The following illustration is for explanation purpose only, and should not be referred to as an actual documentation.

10.4 JP/GP.READ (Reading Data from Another Station's Programmable Controller)

Reads data from the device of another station programmable controller, (in units of word)

Execution condition of the instruction

Structure of the instruction in the ladder mode

A device with ○ is applicable to the instruction.

Descriptions of setting data and data type

Setting side

- User: A device value is set by the user.
- System: A device value is set by the CPU module.

Setting data ^{*1}	Internal device (System, user)		File register	Link direct device		Intelligent function module device U□/G□	Index register Zn	Constant			Others
	Bit	Word		Bit	Word			K, H	\$		
(S1)	—	○									
(S2)	○ ^{*2}	○ ^{*2}									
(D1)	—	○									
(D2)		○									

^{*1} Local devices and file registers set for each program cannot be used.
^{*2} T, C, D, W, ST, SD, and SW can be used.
^{*3} When the target station is a Universal model QCPU or a Basic model QCPU, a digit-specified bit device can be used.
 (Ex) K4M16
 When using digit specification, the following conditions must be satisfied:
 • Device number should be a multiple of 16 (16j).
 • Number of digits (points) should be 4 (K4).

(1) Setting data		Set by	Data type
Setting data	Description		
Jn	Own station's network No. (1 to 239, 254) 254: Network specified in Valid Module During Other Station Access	User	BIN16-bit
Un	Start I/O number of the master/local module of the own station (00 to FE _h ; Upper 2 digits of the I/O number expressed in 3 digits)		
(S1)	Start device of the own station where control data is stored (Page 157, Section 10.4 (1)(a))	System	Device name
(S2)	Start device of the target station where data to be read is stored (Page 159, Section 10.4 (1)(b))		
(D1)	Start device of other station where read data is stored A continuous area for the read data length is required (Page 159, Section 10.4 (1)(b))		
(D2)	Device of the own station, which is turned ON for 1 scan upon completion of the instruction. (D2)+1 is turned ON as well when the instruction fails.		

156

- Instructions can be executed under the following conditions.

Any time	During on	On the rising edge	During off	On the falling edge
No symbol				

- The following devices can be used.

Internal device		File register	Constant	
Bit ^{*1}	Word		K, H	\$
X, Y, M, L, F, V, B	T, ST, C, D, W	R, ZR	K□, H□	\$□

*1 For bit data, a bit-specified word device can be used. For example, Word device.Bit No. is used for the bit specification of a word device. (The bit number is specified in hexadecimal.)

For example, the bit 10 of D0 is specified by D0.A. However, bit specification cannot be used for timer (T), retentive timer (ST), and counter (C).

For details on each device, refer to the following.

 User's manual (function explanation, program fundamentals) for the CPU module used

The following data types can be used.

Data type	Description
Bit	Bit data or the start number of bit data
BIN 16-bit	16-bit binary data or the start number of word device
BIN 32-bit	32-bit binary data or the start number of double-word device
BCD 4-digit	Four-digit binary-coded decimal data
BCD 8-digit	Eight-digit binary-coded decimal data
Real number	Floating-point data
Character string	Character string data
Device name	Device name data

TERM

Unless otherwise specified, this manual uses the following terms.

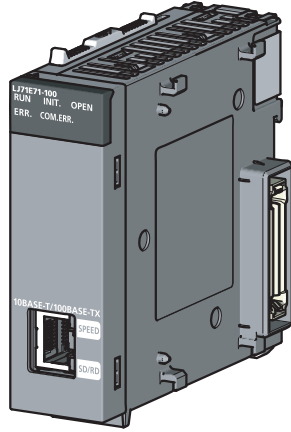
Term	Description
ARP	The abbreviation for Address Resolution Protocol. This protocol is used to obtain the MAC address of Ethernet from an IP address.
BUFRCV	The abbreviation for ZP.BUFRCV
BUFRCVS	The abbreviation for Z.BUFRCVS
BUFSND	The abbreviation for ZP.BUFSND
CLOSE	The abbreviation for ZP.CLOSE
C24	Another term for the L series serial communication module
DNS	The abbreviation for Domain Name System. This system is mainly used to convert host names on the Internet or domain names used for e-mails to IP addresses.
ERRCLR	The abbreviation for ZP.ERRCLR
ERRRD	The abbreviation for ZP.ERRRD
E71	Another term for the LJ71E71-100
E71-connected station	The abbreviation for the station where the E71 is connected
FTP	The abbreviation for File Transfer Protocol. This protocol is used to transfer data files over a network.
GX Developer	The product name of the software package for the MELSEC programmable controllers
GX Works2	
HTTP	The abbreviation for Hyper Text Transfer Protocol. This protocol is used to send and receive content, such as HTML files, between a Web browser and a Web server.
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 an Ethernet network.
IP	The abbreviation for Internet Protocol
MAC address	A unique identifier assigned to each external device on a network. This address is also known as an Ethernet hardware address.
MC protocol	The abbreviation for MELSEC Communication Protocol. This protocol is used to access MC protocol supporting modules, such as the C24 and E71, or programmable controllers connected to MC protocol supporting modules from external devices.
MELSECNET/H	The abbreviation for a MELSECNET/H network system
MELSECNET/H remote I/O station	A generic term for the QJ72LP25-25, QJ72LP25G, and QJ72BR15
MELSECNET/10	The abbreviation for a MELSECNET/10 network system
MRECV	The abbreviation for ZP.MRECV
MSEND	The abbreviation for ZP.MSEND
MX Component	The abbreviation for MX Component (SW0D5C-ACT-E or later)
OPEN	The abbreviation for ZP.OPEN
POP3	The abbreviation for Post Office Protocol Ver.3. This protocol is used to transfer e-mails from a mail server to a local computer.
READ	The abbreviation for JP.READ and GP.READ
RECV	The abbreviation for JP.RECV and GP.RECV
RECVS	The abbreviation for Z.RECVS
REQ	The abbreviation for J.REQ, JP.REQ, G.REQ, and GP.REQ
SEND	The abbreviation for JP.SEND and GP.SEND
SMTP	The abbreviation for Simple Mail Transfer Protocol. This protocol is used to transfer e-mails over the Internet.
SREAD	The abbreviation for JP.SREAD and GP.SREAD
SWRITE	The abbreviation for JP.SWRITE and GP.SWRITE

Term	Description
UINI	The abbreviation for ZP.UINI
WRITE	The abbreviation for JP.WRITE and GP.WRITE
ZNRD	The abbreviation for J.ZNRD and JP.ZNRD
ZNWR	The abbreviation for J.ZNWR and JP.ZNWR
Intelligent function module	A module that has functions other than an input or output, such as an A/D converter module and D/A converter module
Subnet mask	<p>A number used to logically divide one network into multiple subnetworks and manage them easily. The following Ethernet network systems can be configured:</p> <ul style="list-style-type: none"> • 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
Device	A device (X, Y, M, D, or others) in a CPU module
Buffer memory	A memory in an intelligent function module, where data (such as setting values and monitoring values) exchanged with a CPU module are stored
Buffer memory address	An address that indicates the storage location of data assigned to the buffer memory in an intelligent function module
Programming tool	Another term for GX Works2
Dedicated Instruction	An instruction that simplifies programming for using functions of intelligent function modules

PACKING LIST

The following items are included in the package of this product. Before use, check that all the items are included.

LJ71E71-100



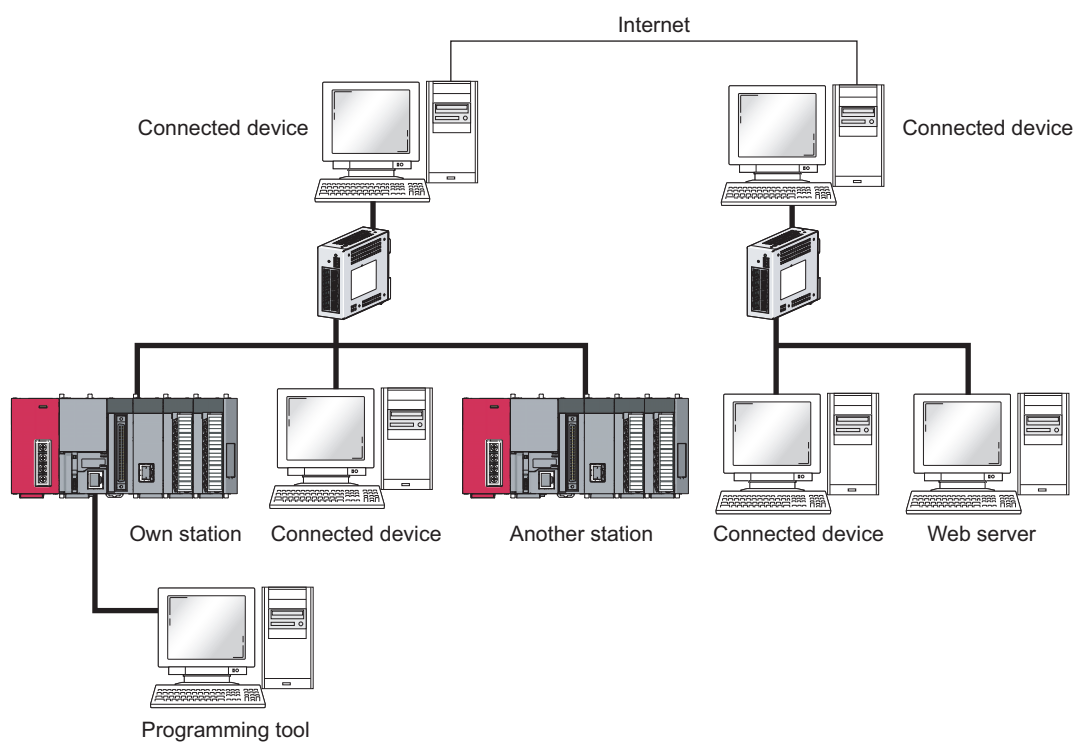
LJ71E71-100



Before Using the Product

CHAPTER 1 FEATURES

An Ethernet module (hereafter abbreviated as E71) is an interface module on the programmable controller side for connecting a programmable controller system to the host system, such as a personal computer and a workstation, over Ethernet. The module can collect and modify programmable controller data, monitor and control CPU operating status, and exchange data in TCP/IP or UDP/IP.

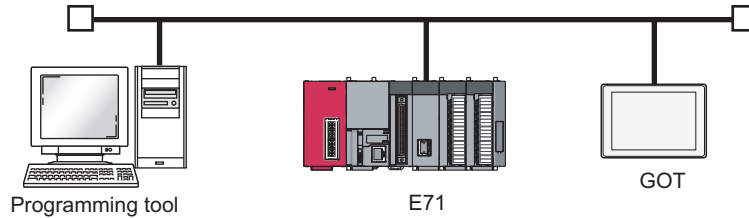


The E71 has the following basic functions.

- Connection with MELSOFT products and a GOT
- Collection and modification of CPU module data from connected devices (MC protocol communications)
- Exchange of data with connected devices (communications using a fixed buffer and random access buffer)
- Prevention of unauthorized access through a remote password
- E-mail sending/receiving (e-mail function)
- Data sending/receiving using the Web function

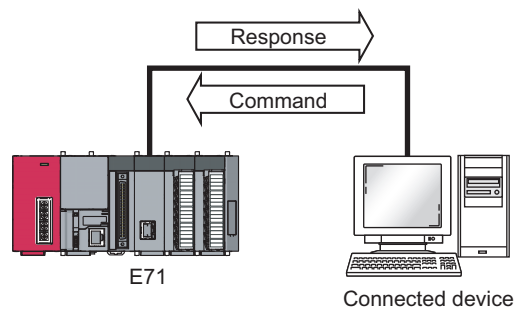
(1) Connection with MELSOFT products and a GOT

In Ethernet, a programming tool can create programming of a programmable controller and monitor a programmable controller (MELSOFT connection), and the GOT can monitor and test a programmable controller. Remote operations making full use of the Ethernet capability, long-distance connectivity and high-speed communications, are achieved.



(2) Collection and modification of CPU module data from connected devices (MC protocol communications)

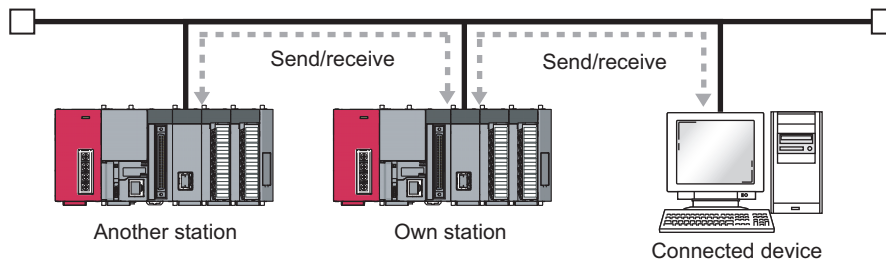
The MC protocol enables connected devices to access MC protocol supporting modules over Ethernet. The E71 can communicate with a personal computer and HMI (Human Machine Interface) as long as the connected devices can receive/send messages in the MC protocol control procedure. By using a separately sold communication support tool (MX Component), a communication program for the host system can be created without considering detailed protocols (communication procedures). (☞ Page 84, CHAPTER 9)



(3) Exchange of data with connected devices (communications using a fixed buffer and random access buffer)

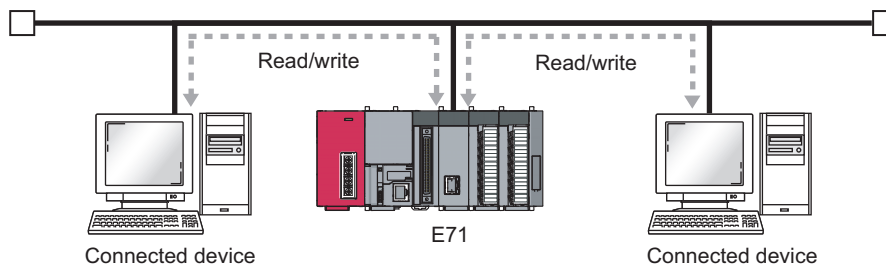
(a) Communications using a fixed buffer

Up to 1K-word data can be exchanged among programmable controllers or between a programmable controller and the host system. While MC protocol communications are passive, communications using a fixed buffer are an active protocol. If an error occurs in equipment or certain conditions are met, the programmable controller can send data to the host system. Using an interrupt program allows the CPU module to quickly read received data. (☞ Page 87, CHAPTER 10)



(b) Communications using a random access buffer

Up to 6K-word data can be communicated. This buffer is useful when the data size is too large for communications using a fixed buffer (capacity: 1K word). (☞ Page 120, CHAPTER 11)




(4) Prevention of unauthorized access through a remote password

This function prevents unauthorized remote access to the CPU module. The E71 checks an entered remote password in data communications from a connected device using remote password-protected connection.

(☞ Page 140, Section 12.3)

(5) E-mail sending/receiving (e-mail function)

This function sends and receives e-mails to and from a connected device in a remote location via the Internet. For details, refer to the following.

 MELSEC-Q/L Ethernet Interface Module User's Manual (Application)

(a) E-mail sending/receiving through the CPU module

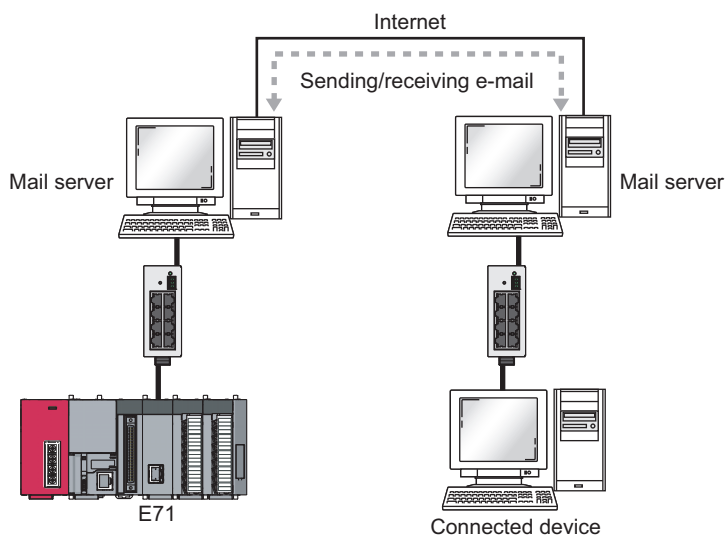
The following data can be sent and received using the MSEND/MRECV instructions.

- The CPU module can receive/send up to 6K-word data from/to a personal computer or other E71 modules as an e-mail attachment.
- The CPU module can send up to 960-word data to a personal computer or portable terminal as the main text of an e-mail.

(b) E-mail sending using the programmable controller CPU monitoring function


Notification conditions (CPU module status or device values) that have been set using parameters are regularly monitored. When the conditions are met, up to 960-word data can be sent by either of the following data formats.

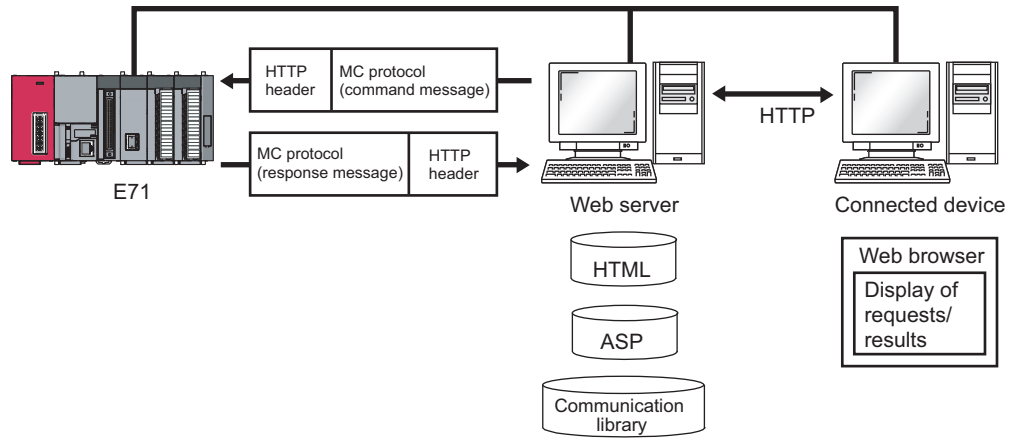
- Attachment
- Main text



(6) Data sending/receiving using the Web function

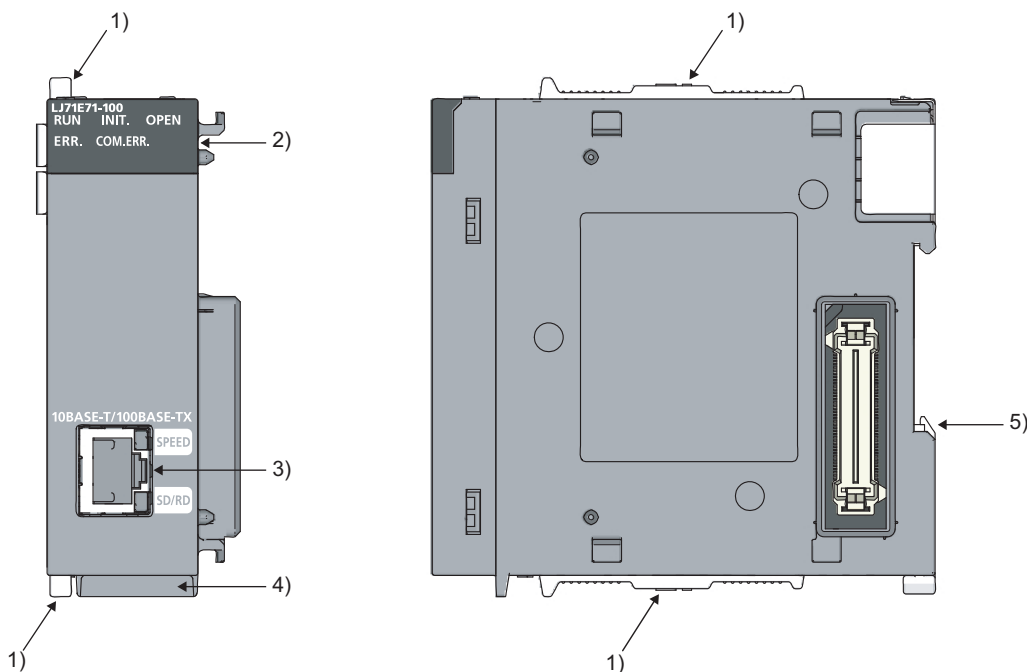
The system administrator can monitor a CPU module in a remote location via the Internet using a Web browser. For details, refer to the following.

 MELSEC-Q/L Ethernet Interface Module User's Manual (Web function)



CHAPTER 2 PART NAMES

This chapter describes the E71 parts.



No.	Name	Application	
1)	Module joint lever	A lever to fix modules	
2)	RUN	Indicates operating status.	
		ON	In normal operation
		OFF	An error has occurred. (☞ Page 191, Section 14.4.1)
	INIT.	Indicates initial process status.	
		ON	Normal completion
		OFF	Not processed
	OPEN* ¹	Indicates open process status.	
		ON	An open process normally completed (connection open)
		OFF	An open process normally completed (no connection)
	ERR.	Indicates hardware failure status.	
		ON	A hardware failure has occurred.
		OFF	Normal
COM.ERR.* ²	Indicates whether a communication failure has occurred.		
	ON	A communication failure has occurred. (☞ Page 191, Section 14.4.2)	
	OFF	Normal communications in progress	

No.	Name	Application	
3)	SPEED	Indicates a transmission speed.	
		ON	100Mbps
		OFF	10Mbps or a cable not connected
	SD/RD	Indicates data communication status.	
		ON	Data communications in progress
		OFF	Data not sent or not received (☞ Page 192, Section 14.4.3)
	10BASE-T/100BASE-TX communication connector (RJ45)	A connector to connect an E71 to the 10BASE-T or 100BASE-TX network (An E71 judges whether to connect to the 10BASE-T or 100BASE-TX network based on the hub.)	
4)	Serial number display	A display indicating the serial number printed on the rating plate	
5)	DIN rail hook	A hook to install the module on a DIN rail	


- *1 The OPEN LED turns on and off depending on the open status of user connections 1 to 16. (The open status of the system connections (e.g. automatic open UDP port) is not included.)
- *2 If the COM.ERR. LED is on, it does not turn off even if the error cause is eliminated. For how to turn off the LED, refer to "How to Turn Off the COM.ERR. LED". (☞ Page 240, Section 14.10)

CHAPTER 3 SPECIFICATIONS

This chapter describes the performance specifications, functions, CPU module I/O signals, and buffer memory areas of an E71.

3.1 General Specifications

For the general specifications of an E71, refer to the following.

 "Safety Guidelines", the manual included with the CPU module

3.2 Performance Specifications

The following table lists the performance specifications of an E71.

Item		Specifications	
		LJ71E71-100	
		100BASE-TX	10BASE-T
Transmission specifications	Data transmission speed	100Mbps	10Mbps
	Interface	RJ45 (AUTO MDI/MDI-X)	
	Communication mode	Full-duplex/Half-duplex	Half-duplex
	Transmission method	Base band	
	Maximum segment length	100m (length between a hub and a node)*1	
	Maximum number of cascade connections	Cascade connection (maximum of 2 levels*2)	Cascade connection (maximum of 4 levels*2)
Sending/receiving data storage memory	Number of simultaneous open connections	16 connections (Connections usable on a program)	
	Fixed buffer	1k word × 16	
	Random access buffer	6k words × 1	
	E-mail	Attachment	6k words × 1
Main text		960k words × 1	
Number of occupied I/O points		32 points (I/O assignment: Intelligent 32 points)	
Internal current consumption (5VDC)		0.60A	
External dimensions		95 (H) × 28.5 (W) × 90 (D) [mm]	
Weight		0.18kg	

Item			Specifications	
			LJ71E71-100	
			100BASE-TX	10BASE-T
Transmission specifications sending/receiving data	Data size	Attachment	6k words × 1	
		Main text	960k words × 1	
	Data transfer method		When sending: Send either a file as attachment or main text (select one). When receiving: Receive a file as attachment.	
	Subject		Us-ASCII format or ISO-2022-JP (Base64)	
	Attachment format		MIME format	
	MIME		Version 1.0	
	Data of attachment format		Can be selected from binary, ASCII, and CSV. File name: XXXX.bin (binary), XXXX.asc (ASCII), XXXX.csv (CSV) (CSV: Comma Separated Value)	
	Division of attachment		Cannot be divided (Only one file can be sent/received.)* ³	
	When sending (encode)		Subject: Base64/7 bits Main text: 7 bits Attachment: Base64	
	When receiving (decode)		Subject: (Does not decode) Main text: (Cannot be received) Attached file: Base64/7 bits/8 bits/Quoted Printable* ⁴	
	Encryption		None	
	Compression		None	
	Communications with a mail server		SMTP (sending server) Port number = 25, POP3 (receiving server) Port number = 110	
	Operation check mailer		Microsoft® Corporation Microsoft Office Outlook 2003	

- *1 For the maximum segment length (a length between hubs), consult with the manufacturer of the switching hub used.
- *2 This applies when a repeater hub is used. For the number of levels that can be constructed when a switching hub is used, consult with the manufacturer of the switching hub used.
- *3 If divided files are received, only the first file is received and the remaining files are discarded.
- *4 If an e-mail is sent from a connected device to the programmable controller side, specify the encoding method (Base64/7 bits/8 bits/Quoted Printable) of the attachment.

3.3 Function List

This section lists the E71 functions.

3.3.1 Function list

The following table lists the functions of the E71.


(1) Basic functions

The following table lists the basic E71 functions explained in this manual.

Function		Description	Reference
Connecting with MELSOFT products and a GOT		An E71 can be connected with MELSOFT products, such as a programming tool and MX Component, and a GOT.	Page 81, CHAPTER 8
MC protocol communications		CPU module data can be read/written from/to connected devices. Access to files can be also performed.	Page 84, CHAPTER 9
Communications using a fixed buffer	Procedure exists	Any data is sent/received between a CPU module and connected devices using the fixed buffer of an E71.	Page 87, CHAPTER 10
	No procedure		
	Pairing open	Paring receiving/sending connections enables data communications with two connections by performing the open process for one port.	Page 105, Section 10.7
	Broadcast communications	Broadcast communications are enabled with all E71-connected stations in the same Ethernet network that is connected to the E71 when "No Procedure" communications using a fixed buffer are performed using UDP/IP.	Page 107, Section 10.8
Communications using a random access buffer		Data is read/written from multiple connected devices to the random access buffer of an E71.	Page 120, CHAPTER 11
Router relay function		Data communications are performed through a router and a gateway. This function is not the function where an E71 operates as a router.	Page 135, Section 12.1
Communications using an auto-open UDP port		Communications are enabled without the open/close processes after an E71-connected station is started up.	Page 139, Section 12.2
Remote password		Unauthorized remote access to a CPU module is prevented.	Page 140, Section 12.3
Hub connection status monitor function		The current connection status and transmission speed of an E71 and a hub and the number of times that the E71 detected disconnection can be checked.	Page 150, Section 12.4
IP address in use detection function		If different stations in the same network use the same IP address, the address in use can be detected.	Page 151, Section 12.5
Alive check function		Whether a connected device is normally operating after a connection is established (open process) can be checked.	Page 153, Section 12.6
Module error collection function		An error that has occurred in an E71 can be stored in the CPU module as error history. The history data can be stored on a memory with the backup power feature; therefore, error details are held even if the CPU module is reset or the system is powered off.	Page 190, Section 14.3

(2) Special functions


The following special functions are also available. For the functions, refer to the following.

 MELSEC-Q/L Ethernet Interface Module User's Manual (Application)

Function	Description
E-mail function	Data are sent/received using an e-mail. • Data sent/received by a CPU module • Data sent using the programmable controller CPU monitoring function (automatic notification function)
CC-Link IE Controller Network, CC-Link IE Field Network, MELSECNET/H, MELSECNET/10 relay communications	Data are communicated over multiple network systems where Ethernet and other networks exist together or network systems that relay multiple Ethernet networks.
Communications using data link instructions	Data of a CPU module in other stations can be read/written over Ethernet using data link instructions.
File transfer (FTP server) function	Data can be read/written in files from connected devices using an exclusive FTP command.

(3) Web function

This function allows data to be read/written from/to a remote CPU module over the Internet using a Web browser. For the function, refer to the following.

 MELSEC-Q/L Ethernet Interface Module User's Manual (Web function)

3.3.2 Use with other functions

The following table lists the relationships between functions that can be used together.

○: Available, ×: Not available or this function does not correspond to the functions in the "Communication function" column.

Communication function	CC-Link IE Controller Network, CC-Link IE Field Network, MELSECNET/H, MELSECNET/10 relay communications	Router relay function	Alive check function	Pairing open	Communications using an auto-open UDP port	Remote password	Broadcast communications	Communication method	
								TCP/IP	UDP/IP
MC protocol communications	○*1	○	○*2	×	○*1	○	○*1 *3	○	○
Communications using a fixed buffer	×	○	○	○	×	○	○*3 *4	○	○
Communications using a random access buffer	×	○	○	×	×	○	×	○	○
E-mail function	×	×	×	×	×	×	×	○	×
Communications using data link instructions	○	○	×	×	○	×	○	×	○
File transfer (FTP server) function	×	○	×	×	×	○	×	○	×
Web function	○	○	×	×	×	○	×	○	×

*1 These functions cannot be used with an A-compatible 1E frame.

*2 The auto-open UDP port is excluded.

*3 These functions can be used only for UDP/IP communication.

*4 These functions cannot be used for communications using a fixed buffer in the "Procedure Exist" control method.

3.4 List of I/O Signals

The following table lists the I/O signals for an E71. The I/O signal assignment of when the start I/O number of an E71 is 0000 is listed below.

Device number	Signal name	Device number	Signal name
X0	For fixed buffer communication of connection No.1 (ON: Sending normal completion or reception completion, OFF: -)	Y0	Connection No.1 (ON: At sending request or reception complete confirmation signal, OFF: -)
X1	For fixed buffer communication of connection No.1 (ON: Detection of sending error or reception error, OFF: -)	Y1	Connection No.2 (ON: At sending request or reception complete confirmation signal, OFF: -)
X2	For fixed buffer communication of connection No.2 (ON: Sending normal completion or reception completion, OFF: -)	Y2	Connection No.3 (ON: At sending request or reception complete confirmation signal, OFF: -)
X3	For fixed buffer communication of connection No.2 (ON: Detection of sending error or reception error, OFF: -)	Y3	Connection No.4 (ON: At sending request or reception complete confirmation signal, OFF: -)
X4	For fixed buffer communication of connection No.3 (ON: Sending normal completion or reception completion, OFF: -)	Y4	Connection No.5 (ON: At sending request or reception complete confirmation signal, OFF: -)
X5	For fixed buffer communication of connection No.3 (ON: Detection of sending error or reception error, OFF: -)	Y5	Connection No.6 (ON: At sending request or reception complete confirmation signal, OFF: -)
X6	For fixed buffer communication of connection No.4 (ON: Sending normal completion or reception completion, OFF: -)	Y6	Connection No.7 (ON: At sending request or reception complete confirmation signal, OFF: -)
X7	For fixed buffer communication of connection No.4 (ON: Detection of sending error or reception error, OFF: -)	Y7	Connection No.8 (ON: At sending request or reception complete confirmation signal, OFF: -)
X8	For fixed buffer communication of connection No.5 (ON: Sending normal completion or reception completion, OFF: -)	Y8	Connection No.1 (ON: Open request, OFF: -)
X9	For fixed buffer communication of connection No.5 (ON: Detection of sending error or reception error, OFF: -)	Y9	Connection No.2 (ON: Open request, OFF: -)
XA	For fixed buffer communication of connection No.6 (ON: Sending normal completion or reception completion, OFF: -)	YA	Connection No.3 (ON: Open request, OFF: -)
XB	For fixed buffer communication of connection No.6 (ON: Detection of sending error or reception error, OFF: -)	YB	Connection No.4 (ON: Open request, OFF: -)
XC	For fixed buffer communication of connection No.7 (ON: Sending normal completion or reception completion, OFF: -)	YC	Connection No.5 (ON: Open request, OFF: -)
XD	For fixed buffer communication of connection No.7 (ON: Detection of sending error or reception error, OFF: -)	YD	Connection No.6 (ON: Open request, OFF: -)
XE	For fixed buffer communication of connection No.8 (ON: Sending normal completion or reception completion, OFF: -)	YE	Connection No.7 (ON: Open request, OFF: -)
XF	For fixed buffer communication of connection No.8 (ON: Detection of sending error or reception error, OFF: -)	YF	Connection No.8 (ON: Open request, OFF: -)
X10	Open completed for connection No.1 (ON: Open completion signal, OFF: -)	Y10	Use prohibited
X11	Open completed for connection No.2 (ON: Open completion signal, OFF: -)	Y11	Use prohibited

Device number	Signal name	Device number	Signal name
X12	Open completed for connection No.3 (ON: Open completion signal, OFF: -)	Y12	Use prohibited
X13	Open completed for connection No.4 (ON: Open completion signal, OFF: -)	Y13	Use prohibited
X14	Open completed for connection No.5 (ON: Open completion signal, OFF: -)	Y14	Use prohibited
X15	Open completed for connection No.6 (ON: Open completion signal, OFF: -)	Y15	Use prohibited
X16	Open completed for connection No.7 (ON: Open completion signal, OFF: -)	Y16	Use prohibited
X17	Open completed for connection No.8 (ON: Open completion signal, OFF: -)	Y17	COM.ERR. LED Off request (ON: At off request, OFF: -)
X18	Open abnormal detection signal (ON: At off request, OFF: -)	Y18	Use prohibited
X19	Open abnormal detection signal (ON: Normal completion, OFF: -)	Y19	Initial request signal (ON: At request, OFF: -)
X1A	Open abnormal detection signal (ON: Abnormal end, OFF: -)	Y1A	Use prohibited
X1B	Use prohibited	Y1B	Use prohibited
X1C	COM.ERR. LED lit confirmation (ON: lit, OFF: off)	Y1C	Use prohibited
X1D	Use prohibited	Y1D	Use prohibited
X1E	Use prohibited	Y1E	Use prohibited
X1F	Watchdog timer error detection (ON: Watchdog timer error, OFF: -)	Y1F	Use prohibited

Point

Do not use any use prohibited signal as an I/O signal to the CPU module. Doing so may cause malfunction of the programmable controller system.

Remark

The I/O signals listed in this section are mainly applied when QnA series module programs are used. For the L series, the I/O signals for intelligent function modules are turned on and off with dedicated instructions. When QnA series module programs are used, it is recommended to replace the signals with the dedicated instructions described in the section that describes the corresponding functions.

3.5 Buffer Memory

This section describes the E71 buffer memory.

3.5.1 Configuration of the buffer memory

This section describes a buffer memory configuration.

(1) Buffer memory address configuration

A buffer memory area consists of 16 bits per address.

b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
-----	-----	-----	-----	-----	-----	----	----	----	----	----	----	----	----	----	----

(2) Buffer memory area configuration

Buffer memory consists of user areas and system areas.

(a) User areas

A user area is an area where a user writes or reads data. A user area consists of a parameter area for the initial process and data communications, an area for data communications, and an area to store communication status and communication error data. Data communications may take long if user areas are used continually; therefore, use them only when needed.

(b) System areas

A system area is an area used by a system.

Point

Do not write any data to the "system area". Doing so may cause malfunction of the programmable controller system.

3.5.2 List of buffer memory addresses

The following table lists the buffer memory addresses of an E71.

Address	Application	Name	Initial value	Programming tool setting applicability	
Decimal (Hexadecimal)					
0 and 1 (0 _H and 1 _H)	Initial process parameter setting area	Own station E71 IP address	C00001FE _H	○	
2 and 3 (2 _H and 3 _H)		System area	-	-	
4 (4 _H)		Special function settings Router relay function (b5, b4) • 00: Do not use • 01: Use Conversion system setting for CC-Link IE Controller Network, CC-Link IE Field Network, MELSECNET/H, MELSECNET/10 relay function (b7, b6) • 00: Automatic response system • 01: IP address computation system • 10: Table exchange system • 11: Use-together system FTP function setting (b9, b8) • 00: Do not use • 01: Use Bits other than those described above are reserved for system use.	0100 _H	○	
5 to 10 (5 _H to A _H)		System area	-	-	
11 (B _H)		Monitoring timer	TCP ULP timer value (Setting time = setting value × 500 ms)	3C _H (60)	○
12 (C _H)			TCP zero window timer value (Setting time = setting value × 500 ms)	14 _H (20)	○
13 (D _H)			TCP resend timer value (Setting time = setting value × 500 ms)	14 _H (20)	○
14 (E _H)			TCP end timer value (Setting time = setting value × 500 ms)	28 _H (40)	○
15 (F _H)			IP assembly timer value (Setting time = setting value × 500 ms)	A _H (10)	○
16 (10 _H)			Response monitoring timer value (Setting time = setting value × 500 ms)	3C _H (60)	○
17 (11 _H)	Destination existence confirmation starting interval timer value (Setting time = setting value × 500 ms)		4B0 _H (1200)	○	
18 (12 _H)	Destination existence confirmation interval timer value (Setting time = setting value × 500 ms)		14 _H (20)	○	
19 (13 _H)	Destination existence confirmation resending time		3 _H (3)	○	

Address	Application	Name	Initial value	Programming tool setting applicability
Decimal (Hexadecimal)				
20 (14 _H)	Initial process parameter setting area	Auto-open UDP port number	1388 _H	×
21 to 29 (15 _H to 1D _H)		System area	-	-
30 (1E _H)	Initial process parameter setting area (For reinitialization)	TCP Maximum Segment Transmission setting area <ul style="list-style-type: none"> • 0_H: Enable TCP Maximum Segment Size Option transmission • 8000_H: Disable TCP Maximum Segment Size Option transmission <p>Re-initialization makes the set value effective.</p>	8000 _H	×
31 (1F _H)		Communication condition setting (Ethernet Operation Setting) area Communication data code setting (b1) <ul style="list-style-type: none"> • 0: Binary Code • 1: ASCII Code TCP existence confirmation setting (b4) <ul style="list-style-type: none"> • 0: Use the Ping • 1: Use the KeepAlive Send frame setting (b5) <ul style="list-style-type: none"> • 0: Ethernet • 1: IEEE 802.3 Setting of write enable/disable at RUN time (b6) <ul style="list-style-type: none"> • 0: Disable • 1: Enable Initial timing setting (b8) <ul style="list-style-type: none"> • 0: Do not wait for OPEN (Communications impossible at STOP time) • 1: Always wait for OPEN (Communication possible at STOP time) Reinitialization specification (b15) <ul style="list-style-type: none"> • 0: Reinitialization process complete (reset by the system) • 1: Reinitialization process request (set by the user) <p>Bits other than those described above are reserved for system use.</p>	0 _H	○

Address	Application	Name	Initial value	Programming tool setting applicability
Decimal (Hexadecimal)				
32 (20 _H)	Communication parameter setting area	Connection usage setting area	0 _H	○
33 (21 _H)				
34 (22 _H)				
35 (23 _H)				
36 (24 _H)				
37 (25 _H)				
38 (26 _H)				
39 (27 _H)				

Connection No.1
Usage of fixed buffer (b0)
• 0: For sending, or communications using a fixed buffer are not performed
• 1: For receiving
Destination existence confirmation (b1)
• 0: No confirm
• 1: Confirm
Pairing open (b7)
• 0: Disable
• 1: Enable
Communication method (protocol) (b8)
• 0: TCP/IP
• 1: UDP/IP
Communications using a fixed buffer (b9)
• 0: Procedure exists
• 1: No procedure
Open system (b15, b14)
• 00: Active open or UDP/IP
• 10: Unpassive open
• 11: Fullpassive open
Bits other than those described above are reserved for system use.

Connection No.2 (The bit configuration is the same as Connection No.1.)

Connection No.3 (The bit configuration is the same as Connection No.1.)

Connection No.4 (The bit configuration is the same as Connection No.1.)

Connection No.5 (The bit configuration is the same as Connection No.1.)

Connection No.6 (The bit configuration is the same as Connection No.1.)

Connection No.7 (The bit configuration is the same as Connection No.1.)

Connection No.8 (The bit configuration is the same as Connection No.1.)

Address	Application	Name		Initial value	Programming tool setting applicability	
Decimal (Hexadecimal)						
40 (28 _H)	Communication parameter setting area	Communication address setting area	Connection No.1	Own station port No.	0 _H	○
41 and 42 (29 _H and 2A _H)				Destination IP address	0 _H	○
43 (2B _H)				Destination Port No.	0 _H	○
44 to 46 (2C _H to 2E _H)				Destination MAC address	FFFFFFFF FFF _H	×
47 to 53 (2F _H to 35 _H)		Connection No.2 (The bit configuration is the same as Connection No.1.)				
54 to 60 (36 _H to 3C _H)		Connection No.3 (The bit configuration is the same as Connection No.1.)				
61 to 67 (3D _H to 43 _H)		Connection No.4 (The bit configuration is the same as Connection No.1.)				
68 to 74 (44 _H to 4A _H)		Connection No.5 (The bit configuration is the same as Connection No.1.)				
75 to 81 (4B _H to 51 _H)		Connection No.6 (The bit configuration is the same as Connection No.1.)				
82 to 88 (52 _H to 58 _H)		Connection No.7 (The bit configuration is the same as Connection No.1.)				
89 to 95 (59 _H to 5F _H)		Connection No.8 (The bit configuration is the same as Connection No.1.)				
96 to 102 (60 _H to 66 _H)		System area			-	-
103 and 104 (67 _H and 68 _H)		System area			-	-
105 (69 _H)	Communication status storage area	Area for the initial process	Initial error code	0 _H	×	
106 and 107 (6A _H and 6B _H)			Own station IP address	0 _H	×	
108 to 110 (6C _H to 6E _H)			Own station MAC address	0 _H	×	
111 to 115 (6F _H to 73 _H)			System area	-	-	
116 (74 _H)			Auto-open UDP port number	0 _H	×	
117 (75 _H)			System area	-	-	
118 (76 _H)			Station number (b0 to b7) Network number of the own station (b8 to b15)	0 _H	×	
119 (77 _H)			Own station group number	0 _H	×	

Address	Application	Name	Initial value	Programming tool setting applicability		
Decimal (Hexadecimal)						
120 (78 _H)	Communication status storage area	Connection information area	Own station port No.	0 _H	x	
121 and 122 (79 _H and 7A _H)			Destination IP address	0 _H	x	
123 (7B _H)			Destination Port No.	0 _H	x	
124 (7C _H)			Open error code	0 _H	x	
125 (7D _H)			Fixed buffer sending error code	0 _H	x	
126 (7E _H)			Connection end code	0 _H	x	
127 (7F _H)			Fixed buffer communication time (Maximum value)	0 _H	x	
128 (80 _H)			Fixed buffer communication time (Minimum value)	0 _H	x	
129 (81 _H)			Fixed buffer communication time (Current value)	0 _H	x	
130 to 139 (82 _H to 8B _H)			Connection No.2 (The bit configuration is the same as Connection No.1.)			
140 to 149 (8C _H to 95 _H)			Connection No.3 (The bit configuration is the same as Connection No.1.)			
150 to 159 (96 _H to 9F _H)			Connection No.4 (The bit configuration is the same as Connection No.1.)			
160 to 169 (A0 _H to A9 _H)			Connection No.5 (The bit configuration is the same as Connection No.1.)			
170 to 179 (AA _H to B3 _H)			Connection No.6 (The bit configuration is the same as Connection No.1.)			
180 to 189 (B4 _H to BD _H)			Connection No.7 (The bit configuration is the same as Connection No.1.)			
190 to 199 (BE _H to C7 _H)			Connection No.8 (The bit configuration is the same as Connection No.1.)			

Address		Application	Name	Initial value	Programming tool setting applicability	
Decimal (Hexadecimal)						
200 (C8 _H)		Communication status storage area	<p>LED on/off status (Stores the on/off status of the LEDs on the front of the Ethernet module)</p> <p>INIT. LED (b0)</p> <ul style="list-style-type: none"> • 0: OFF • 1: ON (initial process completed) <p>OPEN LED (b1)</p> <ul style="list-style-type: none"> • 0: OFF • 1: ON (connection open processing completed) <p>ERR. LED (b3)</p> <ul style="list-style-type: none"> • 0: OFF • 1: ON (hardware failure) <p>COM.ERR. LED (b4)</p> <ul style="list-style-type: none"> • 0: OFF • 1: ON (communication error) <p>Bits other than those described above are reserved for system use.</p>	0 _H	×	
201 (C9 _H)			Area for module status	<p>Hub connection status area</p> <p>Communication mode (b9)</p> <ul style="list-style-type: none"> • 0: Half duplex • 1: Full duplex <p>Hub connection status (b10)</p> <ul style="list-style-type: none"> • 0: Hub not connected/disconnected • 1: Hub connected <p>Data transmission speed (b15, b14)</p> <ul style="list-style-type: none"> • 00: Operating at 10BASE-T • 01: Operating at 100BASE-TX <p>Bits other than those described above are reserved for system use.</p>	0 _H	×
202 (CA _H)				<p>Switch status (operational mode setting)</p> <p>Communication data code setting</p> <ul style="list-style-type: none"> • 0: Online • 1: Offline • 2: Self-loopback test • 3: Hardware test 	0 _H	○

Address	Application	Name		Initial value	Programming tool setting applicability	
Decimal (Hexadecimal)						
203 (CB _H)	Communication status storage area	Module status area	Status of settings with a programming tool		0 _H	○
			Communication data code setting (b1)			
			• 0: Communications in a binary code			
			• 1: Communications in an ASCII code			
			Initial/open method setting (b2)			
			• 0: No parameter setting (start up according to the program)			
			• 1: Parameter setting (start up according to the parameters)			
			TCP Existence confirmation setting (b4)			
	• 0: Use the Ping					
	• 1: Use the KeepAlive					
	Send frame setting (b5)					
	• 0: Ethernet					
	• 1: IEEE 802.3					
	Setting of write enable/disable at RUN time (b6)					
	• 0: Disable					
	• 1: Enable					
	Initial timing setting (b8)					
	• 0: Do not wait for OPEN (Communications impossible at STOP time)					
	• 1: Always wait for OPEN (Communication possible at STOP time)					
	Bits other than those described above are reserved for system use.					
204 (CC _H)	Area for sending/receiving instructions	System area		-	-	
205 (CD _H)		RECV instruction execution request		0 _H	×	
206 (CE _H)		System area		-	-	
207 (CF _H)		Data link instruction execution result	ZNRD instruction		0 _H	×
208 (D0 _H)			System area		-	-
209 (D1 _H)			ZNWR instruction		0 _H	×
210 to 223 (D2 _H to DF _H)			System area		-	-
224 to 226 (E0 _H to E2 _H)		Error log area	System area		-	-
227 (E3 _H)	Number of errors		0 _H	×		
228 (E4 _H)	Error log write pointer		0 _H	×		

Address	Application	Name	Initial value	Programming tool setting applicability		
Decimal (Hexadecimal)						
229 (E5 _H)	Error log area	Error log block 1	Error code/end code	0 _H	×	
230 (E6 _H)			Subheader	0 _H	×	
231 (E7 _H)			Command code	0 _H	×	
232 (E8 _H)			Connection No.	0 _H	×	
233 (E9 _H)			Own station port No.	0 _H	×	
234 and 235 (EA _H and EB _H)			Destination IP address	0 _H	×	
236 (EC _H)			Destination Port No.	0 _H	×	
237 (ED _H)			System area	-	-	
238 to 246 (EE _H to F6 _H)			Error log block 2 (The bit configuration is the same as Error log block 1.)			
247 to 255 (F7 _H to FF _H)		Error log block 3 (The bit configuration is the same as Error log block 1.)				
256 to 264 (100 _H to 108 _H)		Error log block 4 (The bit configuration is the same as Error log block 1.)				
265 to 273 (109 _H to 111 _H)		Error log block area	Error log block 5 (The bit configuration is the same as Error log block 1.)			
274 to 282 (112 _H to 11A _H)			Error log block 6 (The bit configuration is the same as Error log block 1.)			
283 to 291 (11B _H to 123 _H)			Error log block 7 (The bit configuration is the same as Error log block 1.)			
292 to 300 (124 _H to 12C _H)			Error log block 8 (The bit configuration is the same as Error log block 1.)			
301 to 309 (12D _H to 135 _H)			Error log block 9 (The bit configuration is the same as Error log block 1.)			
310 to 318 (136 _H to 13E _H)	Error log block 10 (The bit configuration is the same as Error log block 1.)					
319 to 327 (13F _H to 147 _H)	Error log block 11 (The bit configuration is the same as Error log block 1.)					
328 to 336 (148 _H to 150 _H)	Error log block 12 (The bit configuration is the same as Error log block 1.)					
337 to 345 (151 _H to 159 _H)	Error log block 13 (The bit configuration is the same as Error log block 1.)					
346 to 354 (15A _H to 162 _H)	Error log block 14 (The bit configuration is the same as Error log block 1.)					
355 to 363 (163 _H to 16B _H)	Error log block 15 (The bit configuration is the same as Error log block 1.)					
364 to 372 (16C _H to 174 _H)	Error log block 16 (The bit configuration is the same as Error log block 1.)					

Address	Application	Name	Initial value	Programming tool setting applicability			
Decimal (Hexadecimal)							
373 to 375 (175 _H to 177 _H)	System area		-	-			
376 and 377 (178 _H and 179 _H)	Error log area	Status for each protocol	IP	Received IP packet count	0 _H	×	
378 and 379 (17A _H and 17B _H)				Received IP packet count discarded due to sumcheck error	0 _H	×	
380 and 381 (17C _H and 17D _H)				Sent IP packet total count	0 _H	×	
382 to 407 (17E _H to 197 _H)				System area	-	-	
408 and 409 (198 _H and 199 _H)			ICMP	Received ICMP packet count	0 _H	×	
410 and 411 (19A _H and 19B _H)				Received ICMP packet count discarded due to sum check error	0 _H	×	
412 and 413 (19C _H and 19D _H)				Sent ICMP packet total count	0 _H	×	
414 and 415 (19E _H and 19F _H)				Echo request total count of received ICMP packets	0 _H	×	
416 and 417 (1A0 _H and 1A1 _H)				Echo reply total count of sent ICMP packets	0 _H	×	
418 and 419 (1A2 _H and 1A3 _H)				Echo request total count of sent ICMP packets	0 _H	×	
420 and 421 (1A4 _H and 1A5 _H)				Echo reply total count of received ICMP packets	0 _H	×	
422 to 439 (1A6 _H to 1B7 _H)				System area	-	-	
440 and 441 (1B8 _H and 1B9 _H)				TCP	Received TCP packet count	0 _H	×
442 and 443 (1BA _H and 1BB _H)					Received TCP packet count discarded due to sum check error	0 _H	×
444 and 445 (1BC _H and 1BD _H)			Sent TCP packet total count		0 _H	×	
446 to 471 (1BE _H to 1D7 _H)			System area		-	-	
472 and 473 (1D8 _H and 1D9 _H)			UDP	Received UDP packet count	0 _H	×	
474 and 475 (1DA _H and 1DB _H)				Received UDP packet count discarded due to sum check error	0 _H	×	
476 and 477 (1DC _H and 1DD _H)				Sent UDP packet total count	0 _H	×	
478 to 481 (1DE _H to 1E1 _H)				System area	-	-	

Address	Application	Name		Initial value	Programming tool setting applicability	
Decimal (Hexadecimal)						
482 to 511 (1E2 _H to 1FF _H)	Error log area	Status for each protocol	System area	-	-	
512 and 513 (200 _H and 201 _H)	Router relay parameter setting area	Sub-net mask		0 _H	○	
514 and 515 (202 _H and 203 _H)		Default router IP address		0 _H	○	
516 (204 _H)		Number of registered routers		0 _H	○	
517 and 518 (205 _H and 206 _H)		Router 1	Sub-net address		0 _H	○
519 and 520 (207 _H and 208 _H)			Router IP address		0 _H	○
521 to 524 (209 _H to 20C _H)		Router 2 (The bit configuration is the same as Router 1.)				
525 to 528 (20D _H to 210 _H)		Router 3 (The bit configuration is the same as Router 1.)				
529 to 532 (211 _H to 214 _H)		Router 4 (The bit configuration is the same as Router 1.)				
533 to 536 (215 _H to 218 _H)		Router 5 (The bit configuration is the same as Router 1.)				
537 to 540 (219 _H to 21C _H)		Router 6 (The bit configuration is the same as Router 1.)				
541 to 544 (21D _H to 220 _H)		Router 7 (The bit configuration is the same as Router 1.)				
545 to 548 (221 _H to 224 _H)		Router 8 (The bit configuration is the same as Router 1.)				
549 (225 _H)		System area		-	-	

Address	Application	Name		Initial value	Programming tool setting applicability	
Decimal (Hexadecimal)						
550 and 551 (226 _H and 227 _H)	Station No. <-> IP information setting area	System area		-	-	
552 (228 _H)		Number of conversion table data		0 _H	○	
553 and 554 (229 _H and 22A _H)		Conversion information No.1	Communication request destination/source stations network number and station number		0 _H	○
555 and 556 (22B _H and 22C _H)			External station Ethernet module IP address		0 _H	○
557 and 558 (22D _H and 22E _H)			System area		-	-
559 to 564 (22F _H to 234 _H)		Conversion information No.2 (The bit configuration is the same as Conversion information No.1.)				
to		to				
931 to 936 (3A3 _H to 3A8 _H)		Conversion information No.64 (The bit configuration is the same as Conversion information No.1.)				
937 and 938 (3A9 _H and 3AA _H)		Subnet mask pattern for CC-Link IE Controller Network, CC-Link IE Field Network, MELSECNET/H, and MELSECNET/10 routing		0 _H	○	
939 to 943 (3AB _H to 3AF _H)		System area		-	-	
944 to 949 (3B0 _H to 3B5 _H)	FTP setting area	FTP login name		"LJ71E71"	○	
950 to 953 (3B6 _H to 3B9 _H)		Password		"LJ71E71"	○	
954 (3BA _H)		Command input monitoring timer		708 _H	○	
955 (3BB _H)		Programmable controller CPU monitoring timer		A _H	○	
956 to 1663 (3BC _H to 67F _H)		System area		-	-	

Address	Application	Name		Initial value	Programming tool setting applicability
Decimal (Hexadecimal)					
1664 (680 _H)	Fixed buffer data area	Fixed buffer No.1	Data length	0 _H	×
1665 to 2687 (681 _H to A7F _H)			Fixed buffer data	0 _H	×
2688 to 3711 (A80 _H to E7F _H)		Fixed buffer No.2 (The bit configuration is the same as Fixed buffer No.1.)			
3712 to 4735 (E80 _H to 127F _H)		Fixed buffer No.3 (The bit configuration is the same as Fixed buffer No.1.)			
4736 to 5759 (1280 _H to 167F _H)		Fixed buffer No.4 (The bit configuration is the same as Fixed buffer No.1.)			
5760 to 6783 (1680 _H to 1A7F _H)		Fixed buffer No.5 (The bit configuration is the same as Fixed buffer No.1.)			
6784 to 7807 (1A80 _H to 1E7F _H)		Fixed buffer No.6 (The bit configuration is the same as Fixed buffer No.1.)			
7808 to 8831 (1E80 _H to 227F _H)		Fixed buffer No.7 (The bit configuration is the same as Fixed buffer No.1.)			
8832 to 9855 (2280 _H to 267F _H)		Fixed buffer No.8 (The bit configuration is the same as Fixed buffer No.1.)			
9856 to 16383 (2680 _H to 3FFF _H)		Shared area for random access buffers and e-mail buffers	Shared area for random access buffers and e-mail buffers		0 _H
16384 to 20479 (4000 _H to 4FFF _H)	System area		-	-	

Address	Application	Name	Initial value	Programming tool setting applicability
Decimal (Hexadecimal)				
20480 (5000 _H)	Connection status storage area	Connection status information area	Open completion signal • 0: Open incomplete • 1: Open completed Connection No.1 (b0) Connection No.2 (b1) to Connection No.16 (b15)	0 _H x
20481 (5001 _H)			System area	- -
20482 (5002 _H)			Open request signal • 0: No open request • 1: Open being requested Connection No.1 (b0) Connection No.2 (b1) to Connection No.16 (b15)	0 _H x
20483 and 20484 (5003 _H and 5004 _H)		System area	- -	
20485 (5005 _H)		Fixed buffer information area	Fixed buffer reception status signal • 0: Data not received • 1: Data being received Connection No.1 (b0) Connection No.2 (b1) to Connection No.16 (b15)	0 _H x
20486 (5006 _H)	Remote password status storage area	Remote password status • 0: Unlock status/no remote password setting • 1: Lock status Connection No.1 (b0) Connection No.2 (b1) to Connection No.16 (b15)	0 _H x	
20487 (5007 _H)	System port information area	Remote password status • 0: Unlock status/no remote password setting • 1: Lock status • Auto-open UDP port (b0) • MELSOFT application transmission port (UDP) (b1) • MELSOFT application transmission port (TCP) (b2) • FTP transmission port (b3)	0 _H x	

Address	Application	Name		Initial value	Programming tool setting applicability	
Decimal (Hexadecimal)						
20488 (5008 _H)	System port information area	System port use prohibited designation area	System port use prohibited designation		×	
20489 to 20591 (5009 _H to 506F _H)			System area			-
20592 (5070 _H)	Monitoring area	Remote password function monitoring area	Remote password mismatch notification accumulated count designation (For user open port)		×	
20593 (5071 _H)			Remote password mismatch notification accumulated count designation (For auto-open UDP port, MELSOFT application transmission port (TCP/UDP) and FTP transmission port)		×	
20594 (5072 _H)			Connection No.1	Accumulated count of unlock process normal completion		×
20595 (5073 _H)				Accumulated count of unlock process abnormal end		×
20596 (5074 _H)				Accumulated count of lock process normal completion		×
20597 (5075 _H)				Accumulated count of lock process abnormal end		×
20598 (5076 _H)				Accumulated count of lock process based on close		×
20599 to 20603 (5077 _H to 507B _H)				Connection No.2 (The bit configuration is the same as Connection No.1.)		
20604 to 20608 (507C _H to 5080 _H)			Connection No.3 (The bit configuration is the same as Connection No.1.)			
20609 to 20613 (5081 _H to 5085 _H)			Connection No.4 (The bit configuration is the same as Connection No.1.)			
20614 to 20618 (5086 _H to 508A _H)			Connection No.5 (The bit configuration is the same as Connection No.1.)			
20619 to 20623 (508B _H to 508F _H)			Connection No.6 (The bit configuration is the same as Connection No.1.)			
20624 to 20628 (5090 _H to 5094 _H)			Connection No.7 (The bit configuration is the same as Connection No.1.)			

Address	Application	Name	Initial value	Programming tool setting applicability
Decimal (Hexadecimal)				
20629 to 20633 (5095 _H to 5099 _H)	Monitoring area	Remote password function monitoring area		Connection No.8 (The bit configuration is the same as Connection No.1.)
20634 to 20638 (509A _H to 509E _H)				Connection No.9 (The bit configuration is the same as Connection No.1.)
20639 to 20643 (509F _H to 50A3 _H)				Connection No.10 (The bit configuration is the same as Connection No.1.)
20644 to 20648 (50A4 _H to 50A8 _H)				Connection No.11 (The bit configuration is the same as Connection No.1.)
20649 to 20653 (50A9 _H to 50AD _H)				Connection No.12 (The bit configuration is the same as Connection No.1.)
20654 to 20658 (50AE _H to 50B2 _H)				Connection No.13 (The bit configuration is the same as Connection No.1.)
20659 to 20663 (50B3 _H to 50B7 _H)				Connection No.14 (The bit configuration is the same as Connection No.1.)
20664 to 20668 (50B8 _H to 50BC _H)				Connection No.15 (The bit configuration is the same as Connection No.1.)
20669 to 20673 (50BD _H to 50C1 _H)				Connection No.16 (The bit configuration is the same as Connection No.1.)
20674 to 20678 (50C2 _H to 50C6 _H)				Auto-open UDP port (The bit configuration is the same as Connection No.1.)
20679 to 20683 (50C7 _H to 50CB _H)				MELSOFT application transmission port (UDP) (The bit configuration is the same as Connection No.1.)
20684 to 20688 (50CC _H to 50D0 _H)				MELSOFT application transmission port (TCP) (The bit configuration is the same as Connection No.1.)
20689 to 20693 (50D1 _H to 50D5 _H)				FTP transmission port (The bit configuration is the same as Connection No.1.)

Address	Application	Name	Initial value	Programming tool setting applicability	
Decimal (Hexadecimal)					
20694 to 20736 (50D6 _H to 5100 _H)	HTTP status storage area	System area	-	-	
20737 (5101 _H)		Error log pointer	0 _H	×	
20738 (5102 _H)		Log counter (HTTP response code 100 to 199)	0 _H	×	
20739 (5103 _H)		Log counter (HTTP response code 200 to 299)	0 _H	×	
20740 (5104 _H)		Log counter (HTTP response code 300 to 399)	0 _H	×	
20741 (5105 _H)		Log counter (HTTP response code 400 to 499)	0 _H	×	
20742 (5106 _H)		Log counter (HTTP response code 500 to 599)	0 _H	×	
20743 (5107 _H)		System area	-	-	
20744 (5108 _H)		Error log block 1	HTTP response code	0 _H	×
20745 and 20746 (5109 _H and 510A _H)			Destination IP address	0 _H	×
20747 to 20750 (510B _H to 510E _H)			Error time	0 _H	×
20751 to 20757 (510F _H to 5115 _H)		Error log block 2 (The bit configuration is the same as Error log block 1.)			
20758 to 20764 (5116 _H to 511C _H)		Error log block 3 (The bit configuration is the same as Error log block 1.)			
20765 to 20771 (511D _H to 5123 _H)		Error log block 4 (The bit configuration is the same as Error log block 1.)			
20772 to 20778 (5124 _H to 512A _H)		Error log block 5 (The bit configuration is the same as Error log block 1.)			
20779 to 20785 (512B _H to 5131 _H)		Error log block 6 (The bit configuration is the same as Error log block 1.)			
20786 to 20792 (5132 _H to 5138 _H)		Error log block 7 (The bit configuration is the same as Error log block 1.)			
20793 to 20799 (5139 _H to 513F _H)		Error log block 8 (The bit configuration is the same as Error log block 1.)			
20800 to 20806 (5140 _H to 5146 _H)		Error log block 9 (The bit configuration is the same as Error log block 1.)			
20807 to 20813 (5147 _H to 514D _H)		Error log block 10 (The bit configuration is the same as Error log block 1.)			
20814 to 20820 (514E _H to 5154 _H)	Error log block 11 (The bit configuration is the same as Error log block 1.)				
20821 to 20827 (5155 _H to 515B _H)	Error log block 12 (The bit configuration is the same as Error log block 1.)				
20828 to 20834 (515C _H to 5162 _H)	Error log block 13 (The bit configuration is the same as Error log block 1.)				

Address	Application	Name	Initial value	Programming tool setting applicability
Decimal (Hexadecimal)				
20835 to 20841 (5163 _H to 5169 _H)	HTTP status storage area	Error log block 14 (The bit configuration is the same as Error log block 1.)		
20842 to 20848 (516A _H to 5170 _H)		Error log block 15 (The bit configuration is the same as Error log block 1.)		
20849 to 20855 (5171 _H to 5177 _H)		Error log block 16 (The bit configuration is the same as Error log block 1.)		
20856 to 20994 (5178 _H to 5202 _H)	Use prohibited	System area	-	-
20995 (5203 _H)	Disconnection detection storage area	Disconnection detection count	0 _H	×
20996 to 21119 (5204 _H to 527F _H)	System area		-	-
21120 (5280 _H)	IP address status storage area	Same IP address detection flag • 0: Unique IP addresses • 1: Same IP address for multiple stations	0	×
21121 to 21123 (5281 _H to 5283 _H)		The MAC address of the station that has been already connected to the network (The address is stored in the station with the same IP address.)	FFFFFFFF FFF _H	×
21124 to 21126 (5284 _H to 5286 _H)		The MAC address of the station with the IP address used for another station (The address is stored in the station that has been already connected to the network.)	FFFFFFFF FFF _H	×
21127 to 22559 (5287 _H to 581F _H)	System area		-	-

Address	Application	Name		Initial value	Programming tool setting applicability	
Decimal (Hexadecimal)						
22560 (5820 _H)	Communication status storage area	Connection information area	Connection No.9	Own station port No.	0 _H	×
22561 and 22562 (5821 _H and 5822 _H)				Destination IP address	0 _H	×
22563 (5823 _H)				Destination Port No.	0 _H	×
22564 (5824 _H)				Open error code	0 _H	×
22565 (5825 _H)				Fixed buffer sending error code	0 _H	×
22566 (5826 _H)				Connection end code	0 _H	×
22567 (5827 _H)				Fixed buffer communication time (Maximum value)	0 _H	×
22568 (5828 _H)				Fixed buffer communication time (Minimum value)	0 _H	×
22569 (5829 _H)				Fixed buffer communication time (Current value)	0 _H	×
22570 to 22579 (582A _H to 5833 _H)			Connection No.10 (The bit configuration is the same as Connection No.9.)			
22580 to 22589 (5834 _H to 583D _H)			Connection No.11 (The bit configuration is the same as Connection No.9.)			
22590 to 22599 (583E _H to 5847 _H)			Connection No.12 (The bit configuration is the same as Connection No.9.)			
22600 to 22609 (5848 _H to 5851 _H)			Connection No.13 (The bit configuration is the same as Connection No.9.)			
22610 to 22619 (5852 _H to 585B _H)			Connection No.14 (The bit configuration is the same as Connection No.9.)			
22620 to 22629 (585C _H to 5865 _H)			Connection No.15 (The bit configuration is the same as Connection No.9.)			
22630 to 22639 (5866 _H to 586F _H)			Connection No.16 (The bit configuration is the same as Connection No.9.)			

Address	Application	Name	Initial value	Programming tool setting applicability		
Decimal (Hexadecimal)						
22640 (5870 _H)	E-mail status storage area	Receive	Number of mails remaining on the server	0 _H	x	
22641 (5871 _H)			Dedicated instruction normal completion count	0 _H	x	
22642 (5872 _H)			Dedicated instruction abnormal end count	0 _H	x	
22643 (5873 _H)			Normal receiving count	0 _H	x	
22644 (5874 _H)			Attached file receiving count	0 _H	x	
22645 (5875 _H)			Server inquiry count	0 _H	x	
22646 (5876 _H)			Server communication error count	0 _H	x	
22647 (5877 _H)			Error log write count	0 _H	x	
22648 (5878 _H)			Receiving error log write pointer	0 _H	x	
22649 (5879 _H)			Error log block 1	Error code	0 _H	x
22650 (587A _H)				Command code	0 _H	x
22651 to 22658 (587B _H to 5882 _H)				From	0 _H	x
22659 to 22662 (5883 _H to 5886 _H)				Date	0 _H	x
22663 to 22692 (5887 _H to 58A4 _H)				Subject	0 _H	x

Address	Application	Name	Initial value	Programming tool setting applicability	
Decimal (Hexadecimal)					
22693 to 22736 (58A5 _H to 58D0 _H)	E-mail status storage area	Receive	Error log block 2 (The bit configuration is the same as Error log block 1.)		
22737 to 22780 (58D1 _H to 58FC _H)			Error log block 3 (The bit configuration is the same as Error log block 1.)		
22781 to 22824 (58FD _H to 5928 _H)			Error log block 4 (The bit configuration is the same as Error log block 1.)		
22825 to 22868 (5929 _H to 5954 _H)			Error log block 5 (The bit configuration is the same as Error log block 1.)		
22869 to 22912 (5955 _H to 5980 _H)			Error log block 6 (The bit configuration is the same as Error log block 1.)		
22913 to 22956 (5981 _H to 59AC _H)			Error log block 7 (The bit configuration is the same as Error log block 1.)		
22957 to 23000 (59AD _H to 59D8 _H)			Error log block 8 (The bit configuration is the same as Error log block 1.)		
23001 to 23044 (59D9 _H to 5A04 _H)			Error log block 9 (The bit configuration is the same as Error log block 1.)		
23045 to 23088 (5A05 _H to 5A30 _H)			Error log block 10 (The bit configuration is the same as Error log block 1.)		
23089 to 23132 (5A31 _H to 5A5C _H)			Error log block 11 (The bit configuration is the same as Error log block 1.)		
23133 to 23176 (5A5D _H to 5A88 _H)			Error log block 12 (The bit configuration is the same as Error log block 1.)		
23177 to 23220 (5A89 _H to 5AB4 _H)			Error log block 13 (The bit configuration is the same as Error log block 1.)		
23221 to 23264 (5AB5 _H to 5AE0 _H)			Error log block 14 (The bit configuration is the same as Error log block 1.)		
23265 to 23308 (5AE1 _H to 5B0C _H)			Error log block 15 (The bit configuration is the same as Error log block 1.)		
23309 to 23352 (5B0D _H to 5B38 _H)			Error log block 16 (The bit configuration is the same as Error log block 1.)		
23353 (5B39 _H)			Send	Dedicated instruction normal completion count	
23354 (5B3A _H)	Dedicated instruction abnormal end count			0 _H	×
23355 (5B3B _H)	Number of mails normally completed			0 _H	×
23356 (5B3C _H)	Attached file sending count			0 _H	×
23357 (5B3D _H)	Sending to the server count			0 _H	×
23358 (5B3E _H)	Number of mails abnormally completed			0 _H	×
23359 (5B3F _H)	Error log write count			0 _H	×
23360 (5B40 _H)	Sending error log write pointer			0 _H	×

Address	Application	Name			Initial value	Programming tool setting applicability
Decimal (Hexadecimal)						
23361 (5B41 _H)	E-mail status storage area	Send	Error log block 1	Error code	0 _H	×
23362 (5B42 _H)				Command code	0 _H	×
23363 to 23370 (5B43 _H to 5B4A _H)				To	0 _H	×
23371 to 23374 (5B4B _H to 5B4E _H)				Date	0 _H	×
23375 to 23404 (5B4F _H to 5B6C _H)				Subject	0 _H	×
23405 to 23448 (5B6D _H to 5B98 _H)		Error log block 2 (The bit configuration is the same as Error log block 1.)				
23449 to 23492 (5B99 _H to 5BC4 _H)		Error log block 3 (The bit configuration is the same as Error log block 1.)				
23493 to 23536 (5BC5 _H to 5BF0 _H)		Error log block 4 (The bit configuration is the same as Error log block 1.)				
23537 to 23580 (5BF1 _H to 5C1C _H)		Error log block 5 (The bit configuration is the same as Error log block 1.)				
23581 to 23624 (5C1D _H to 5C48 _H)		Error log block 6 (The bit configuration is the same as Error log block 1.)				
23625 to 23668 (5C49 _H to 5C74 _H)		Error log block 7 (The bit configuration is the same as Error log block 1.)				
23669 to 23712 (5C75 _H to 5CA0 _H)		Error log block 8 (The bit configuration is the same as Error log block 1.)				
23713 to 24575 (5CA1 _H to 5FFF _H)		System area			-	-
24576 (6000 _H)	Fixed buffer data area	Fixed buffer No.9	Data length	0 _H	×	
24577 to 25599 (6001 _H to 63FF _H)			Fixed buffer data	0 _H	×	
25600 to 26623 (6400 _H to 67FF _H)		Fixed buffer No.10 (The bit configuration is the same as Fixed buffer No.9.)				
26624 to 27647 (6800 _H to 6BFF _H)		Fixed buffer No.11 (The bit configuration is the same as Fixed buffer No.9.)				
27648 to 28671 (6C00 _H to 6FFF _H)		Fixed buffer No.12 (The bit configuration is the same as Fixed buffer No.9.)				
28672 to 29695 (7000 _H to 73FF _H)		Fixed buffer No.13 (The bit configuration is the same as Fixed buffer No.9.)				
29696 to 30719 (7400 _H to 77FF _H)		Fixed buffer No.14 (The bit configuration is the same as Fixed buffer No.9.)				
30720 to 31743 (7800 _H to 7BFF _H)		Fixed buffer No.15 (The bit configuration is the same as Fixed buffer No.9.)				
31744 to 32767 (7C00 _H to 7FFF _H)		Fixed buffer No.16 (The bit configuration is the same as Fixed buffer No.9.)				

CHAPTER 4 PROCEDURES BEFORE OPERATION

This chapter describes how to connect an E71 to Ethernet. For how to communicate with connected devices after the E71 is connected, refer to the communication procedure. (☞ Page 62, CHAPTER 7)

Checkbox

System consideration

Design the system configuration of the E71.

☞ Page 26, CHAPTER 3,
Page 56, CHAPTER 5



Installation

Connect the E71.

☞ Page 24, CHAPTER 2,
Page 59, Section 6.1



Wiring

Connect an Ethernet cable to the E71.

☞ Page 57, Section 5.2,
Page 60, Section 6.2



Setting parameters

Set the following parameters, which are required to start up the E71, from the programming tool.

- Basic setting
- Ethernet operation setting

☞ Page 64, Section 7.1.2,
Page 65, Section 7.1.3



Restarting the CPU module

Reflect the parameter settings by turning off and on the CPU module or resetting the CPU module.

📖 MELSEC-L CPU Module User's Manual
(Hardware Design, Maintenance and
Inspection)



Network diagnostics

Check that the cables are connected properly or that communications are performed properly with the set parameters. (PING test, loopback test)

☞ Page 256, Appendix 4

Continued to COMMUNICATION PROCEDURE

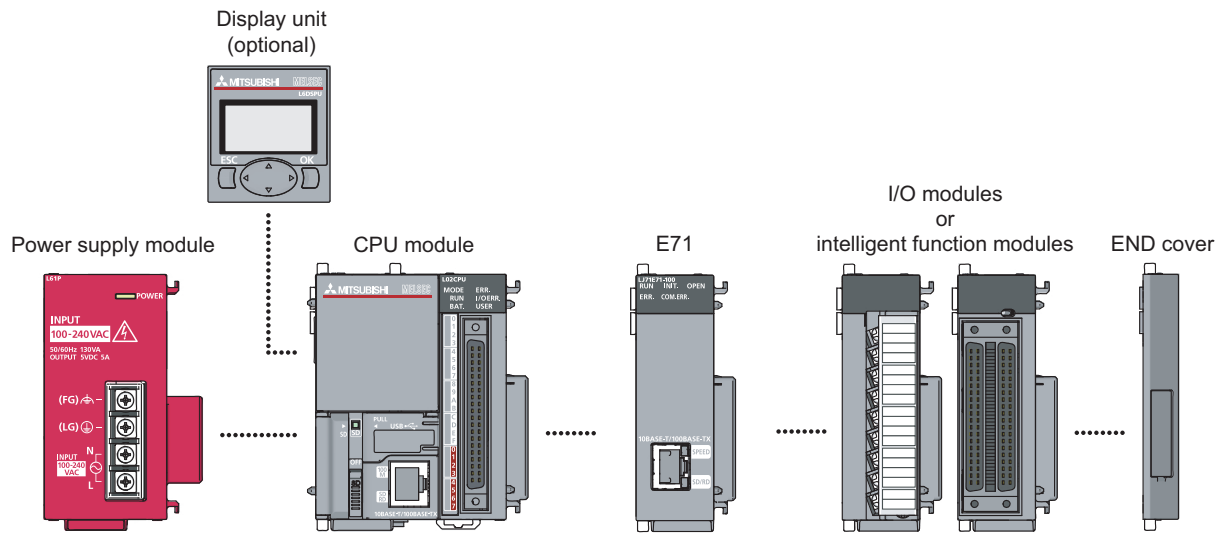
☞ Page 62, CHAPTER 7

Memo

CHAPTER 5 SYSTEM CONFIGURATION

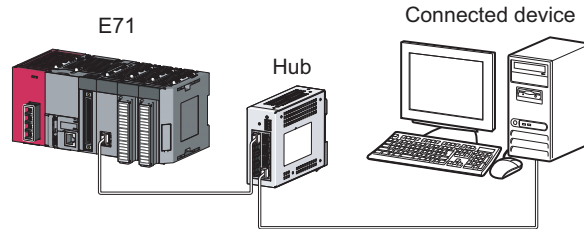
5.1 Overall System Configuration

The following is a system configuration example of when using an E71.



5.2 Network Components

This section describes the devices used for Ethernet.



Point

An E71 determines whether to use 100BASE-TX or 10BASE-T and the full-duplex or half-duplex transmission mode according to the hub. For connection to a hub without the automatic negotiation function, set the half-duplex mode on the hub side.

(1) 100BASE-TX connection

To configure a system for 100BASE-TX connection, use devices that meet the IEEE 802.3 100 BASE-TX standards.

Configuration device		Description
Shielded twisted pair cable (STP) ^{*2}	Straight cable	A Category 5 or higher straight cable is usable.
	Cross cable	A Category 5 or 5e cross cable is usable.
Connector	RJ45 connector	-
Hub	100Mbps hub	Up to two levels of cascade connection can be configured. ^{*1}

*1 This applies when a repeater hub is used. For the number of levels that can be constructed when a switching hub is used, consult with the manufacturer of the switching hub used.

*2 Data can be communicated with an E71 (between LJ71E71-100 modules). In addition, an E71 can be directly connected to a GOT.

Point

During high-speed communications (100Mbps) using 100BASE-TX connection, a communication error may occur due to high-frequency noise from devices other than a programmable controller in a given installation environment. The following lists countermeasures to be taken on the E71 side to avoid high-frequency noise influence.

- Wiring connection
 - Do not bundle the cable with the main circuit or power cable or do not place it near those lines.
 - Put the cable in a duct.
- Communication method
 - Use TCP/IP for data communications with connected devices.
 - Increase the number of retries of communications if needed.
- 10Mbps communications
 - Use a 10Mbps hub for the E71 and set the data transmission speed to 10Mbps.
 - Change the transmission speed of the E71 to 10Mbps using the UINI instruction. (☞ Page 182, Section 13.12)

Remark

Consult a network specialist for required devices.

(2) 10BASE-T connection

To configure a system for 10BASE-T connection, use devices that meet the IEEE 802.3 10BASE-T standards.

Configuration device		Description
Unshielded twisted pair cable (UTP) or shielded twisted pair cable (STP)* ²	Straight cable	A Category 3, 4, or 5 straight cable or cross cable is usable.
	Cross cable	
Connector	RJ45 connector	-
Hub	10Mbps hub	Up to four levels of cascade connection can be configured.* ¹

*1 This applies when a repeater hub is used. For the number of levels that can be constructed when a switching hub is used, consult with the manufacturer of the switching hub used.

*2 Data can be communicated with an E71 (between LJ71E71-100 modules). In addition, an E71 can be directly connected to a GOT.

Remark


Consult a network specialist for required devices.

5.3 Applicable Systems

This section describes MELSEC-L series systems that include an E71.

(1) Applicable CPU modules and the number of connectable modules

An E71 can be connected to an LCPU with an serial number (first five digits) of 14112 or later. For the number of connectable modules, refer to the following.

 MELSEC-L CPU Module User's Manual (Hardware Design, Maintenance and Inspection)

(2) Applicable software

GX Works2 is required for setting and diagnosing an E71. GX Developer cannot be used.

Software	Version
GX Works2	1.95Z or later

CHAPTER 6 INSTALLATION AND WIRING


This chapter describes installation and wiring of the E71.

6.1 Installation

This section describes installation of the E71.

(1) Installation method

For the module installation environment and position, refer to the following.

 MELSEC-L CPU Module User's Manual (Hardware Design, Maintenance and Inspection)

(2) Handling precautions

Do not drop or apply strong shock to the E71 case since it is made from resin.

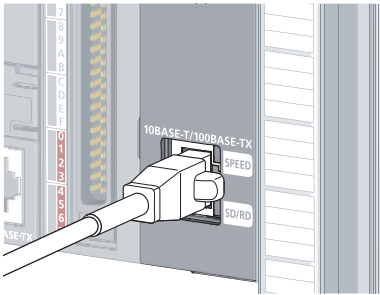
6.2 Wiring

This section describes Ethernet cable wiring and wiring precautions. For a network configuration and cables and hubs used, refer to "SYSTEM CONFIGURATION". (☞ Page 56, CHAPTER 5)

(1) Ethernet cable connection

The following is how to connect and disconnect an Ethernet cable.

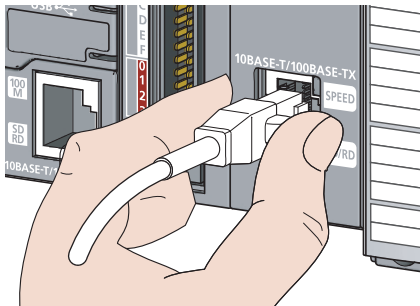
(a) Connecting the cable



1. Power off the E71 and the connected device.
2. Push the Ethernet cable connector into the E71 until it clicks. Pay attention to the connector's direction.

3. Power on the E71 and the connected device.

(b) Disconnecting the cable



1. Power off the system.
2. Press the latch down and unplug the Ethernet cable.

(2) Precautions

This section describes wiring precautions.

(a) Handling

Place the Ethernet cable in a duct or clamp them. If not, dangling cable may swing or inadvertently be pulled, resulting in damage to the module or cables or 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, arising a problem in data link.

Check the following:

- Is any Ethernet cable disconnected?
- Is any Ethernet cable shorted?
- Are the connectors securely connected?

(b) Broken cable latch

Do not use Ethernet cables with broken latches. Doing so may cause the cable to unplug or malfunction.

(c) Connecting and disconnecting the Ethernet cable

Hold the connector part when connecting and disconnecting the Ethernet cable. Pulling a cable connected to the module may damage the module or cable or result in malfunction due to poor contact.

(d) Maximum station-to-station distance (maximum cable length)

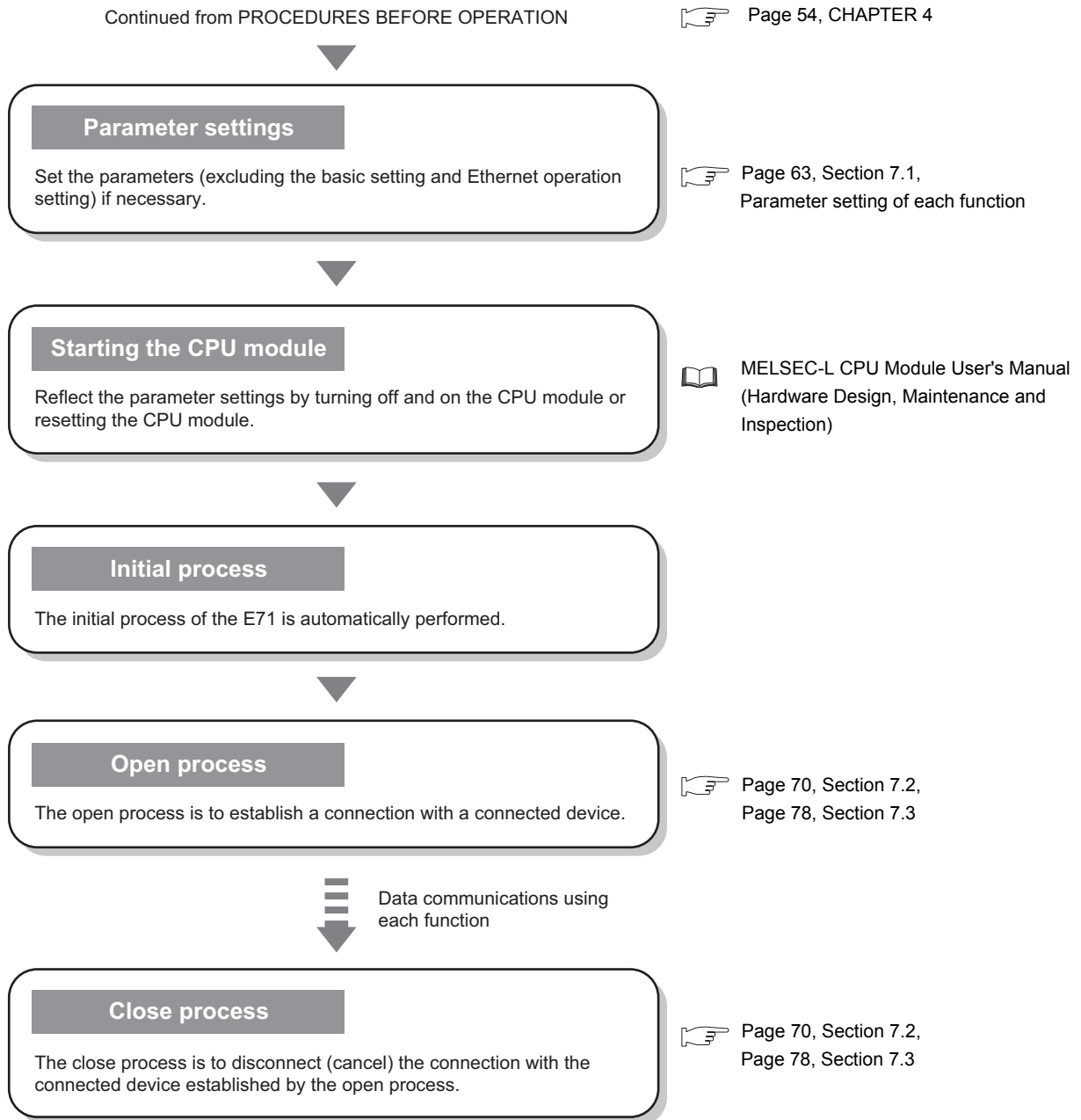
The maximum station-to-station distance is 100m. However, the distance may be shorter depending on the operating environment of the cable. For details, contact your cable manufacturer.

(e) Bend radius

The bend radius of the Ethernet cable is limited. For details, see the specifications of the Ethernet cable to be used.

CHAPTER 7 COMMUNICATION PROCEDURE

This chapter describes the procedure for communicating with connected devices after the E71 is connected to Ethernet. For the procedure for connecting the E71 to Ethernet, refer to "PROCEDURES BEFORE OPERATION".



7.1 Setting Parameters Required for Communications

This section describes how to set parameters to communicate between the E71 and connected devices.

7.1.1 Parameter list

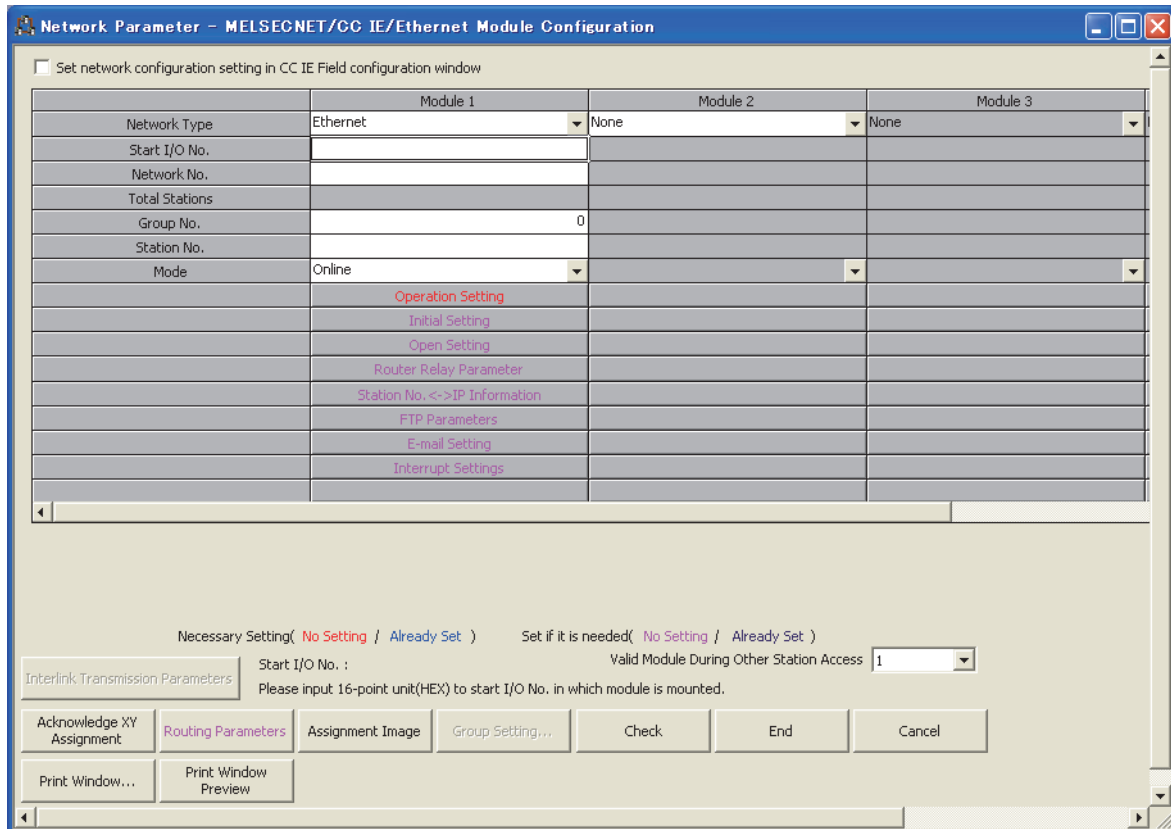
The following table lists parameters set through a programming tool.

Item		Description	Reference
Network Parameter	Basic setting	Network Type Start I/O No. Network No. Group No. Station No. Mode	Configure settings to use the E71 as a network module. Page 64, Section 7.1.2
	Operation Setting	Configure the settings, such as an IP address, to connect the E71 to Ethernet.	Page 65, Section 7.1.3
	Initial Setting	Set a data communication timer value.	Page 246, Appendix 3.1
	Open Setting	Configure connection settings.	Page 67, Section 7.1.4, Parameter setting of each function
	Router Relay Parameter	Configure settings to communicate with connected devices on Ethernet via a router and a gateway.	MELSEC-Q/L Ethernet Interface Module User's Manual (Application)
	Station No. <-> IP Information	Configure settings (e.g. network number and station number) to communicate with other network modules.	
	FTP Parameters	Configure the settings of the file transfer (FTP) function.	
	E-mail Setting	Configure the settings of the e-mail function and the automatic notification function.	
	Interrupt Settings	Set the control number (SI) on the E71 side to request an interrupt to the CPU module.	Page 98, Section 10.5.1
	Routing Parameters	Set the communication path to communicate with a station having a different network number.	MELSEC-Q/L Ethernet Interface Module User's Manual (Application)
	Valid Module During Other Station Access	Select a relay module when an access request with no network number specified is issued from another station.	-
	Remote Password	Remote Password	Select the connection protected by a remote password. Page 140, Section 12.3

7.1.2 Basic setting

Set items, such as a network number and station number.


Project window ⇨ [Parameter] ⇨ [Network Parameter] ⇨ [Ethernet/CC IE Field] ⇨ Select "Ethernet" under "Network Type".

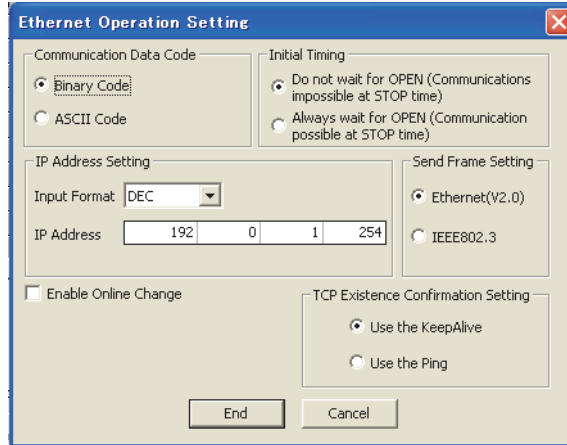


Item	Description	Setting range
Network Type	Select "Ethernet".	-
Start I/O No.	Set the start I/O number of the E71 in units of 16 points.	Within the number of I/O points of the CPU module
Network No.	Set the network number of the E71.	1 to 239
Group No.	Set the group number of the E71.	0 to 32
Station No.	Set the station number of the E71.	1 to 64
Mode	Select the operation mode of the E71.	<ul style="list-style-type: none"> • Online • Offline • Self-Loopback Test • H/W Test

7.1.3 Ethernet Operation Setting

Configure the settings, such as an IP address, to connect the E71 to Ethernet.

Project window ⇨ [Parameter] ⇨ [Network Parameter] ⇨ [Ethernet/CC IE Field] ⇨ Select "Ethernet" under "Network Type". ⇨  button



Item	Description	Setting range
Communication Data Code	Select the communication data code for the connected device.	<ul style="list-style-type: none"> Binary Code ASCII Code
Initial Timing	Refer to (1) in this section.	<ul style="list-style-type: none"> Do not wait for OPEN (Communications impossible at STOP time) Always wait for OPEN (Communication possible at STOP time)
IP Address Setting	Input Format	<ul style="list-style-type: none"> DEC HEX
	IP Address	<ul style="list-style-type: none"> DEC: 0.0.0.1 to 255.255.255.254 HEX: 00.00.00.01 to FF.FF.FF.FE
Send Frame Setting	Select the frame of the Ethernet header for the data link layer to be sent by the E71. An E71 receives data from the connected device both with the Ethernet frame and the IEEE802.3 frame.	<ul style="list-style-type: none"> Ethernet(V2.0) IEEE802.3*1
Enable Online Change	Select whether to enable connected devices to write data in MC protocol communications while the CPU module is in RUN.	<ul style="list-style-type: none"> Selected: Enable Not selected: Disable
TCP Existence Confirmation Setting	Select an alive check method in TCP/IP communications.	<ul style="list-style-type: none"> Use the KeepAlive Use the Ping

*1 The IEEE802.3 frame format can be used only when the connected device is compatible with the frame. Generally, using the Ethernet frame is recommended.

(1) Initial Timing

This setting configures the open timing of the connection where "TCP" (Passive open) or "UDP" has been selected under "Open System" in the open setting. (Page 67, Section 7.1.4)

(a) Do not wait for OPEN (Communications impossible at STOP time)

Connections are opened or closed using a program. Communications cannot be performed while the switch on the CPU module is set to STOP.

(b) Always wait for OPEN (Communication possible at STOP time)

Connections always wait for the open status according to the "Open System" parameter. (This eliminates the need to open/close connections using a program. *1) Communications can be performed while the switch on the CPU module is set to STOP.


*1 If the program of the CPU module on the own station closes a connection, the station does not enter the OPEN request wait status after the connection is disconnected.

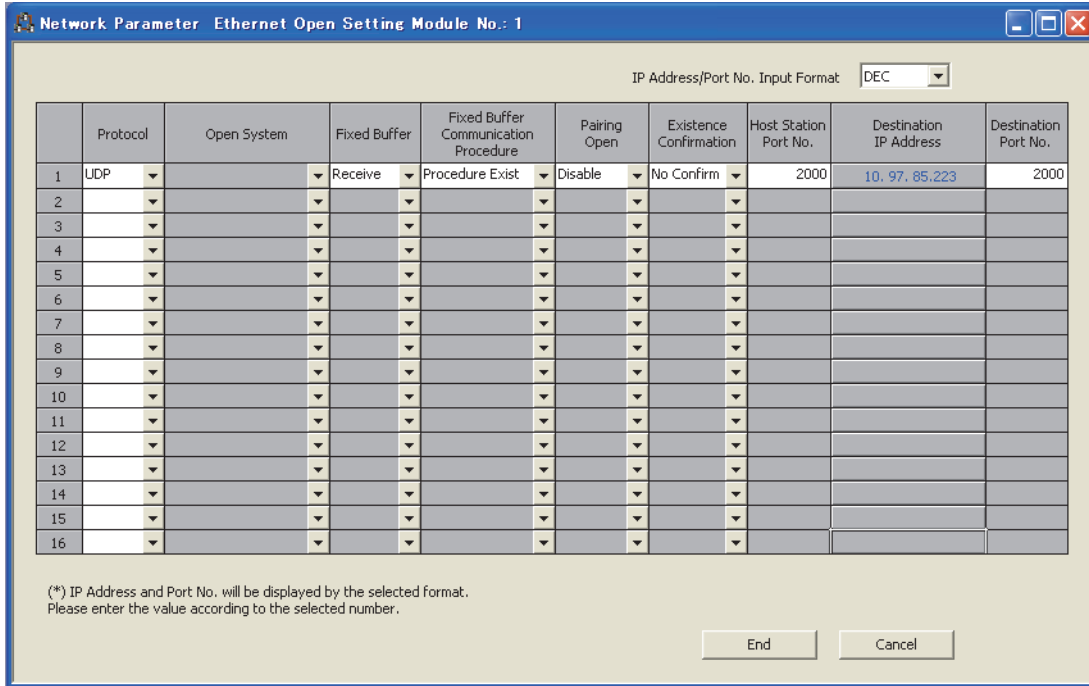
Point

When remotely controlling the CPU module from a connected device, select "Always wait for OPEN (Communication possible at STOP time)". If "Do not wait for OPEN (Communications impossible at STOP time)" is selected, the communication line is closed during remote STOP. After the line is closed, the line cannot be reopened on the CPU module side, and remote RUN cannot be performed from the connected device.

7.1.4 Open Setting

Configure settings to open connections for data communications with connected devices.

Project window ⇨ [Parameter] ⇨ [Network Parameter] ⇨ [Ethernet/CC IE Field] ⇨ Select "Ethernet" under "Network Type". ⇨  button



Item	Description	Setting range
IP Address/Port No. Input Format	Select the input format for the IP address and port number.	<ul style="list-style-type: none"> • DEC • HEX
Protocol	Select the communication protocol for the connected device. (☞ Page 70, Section 7.2, Page 78, Section 7.3)	<ul style="list-style-type: none"> • TCP • UDP
Open System	Set this item when "TCP" is selected under "Protocol". (☞ Page 71, Section 7.2.2, Page 73, Section 7.2.3, Page 75, Section 7.2.4)	<ul style="list-style-type: none"> • Active • Unpassive • Fullpassive • MELSOFT Connection*¹
Fixed Buffer	For communications using a fixed buffer, select whether to use the buffer for sending or for receiving in a connection to the connected device. When not performing communications using a fixed buffer, select "Send". (☞ Page 87, CHAPTER 10)	<ul style="list-style-type: none"> • Send • Receive
Fixed Buffer Communication Procedure	Select a communication method for communications using a fixed buffer. (☞ Page 87, CHAPTER 10)	<ul style="list-style-type: none"> • Procedure Exist • No Procedure
Pairing Open	Configure the pairing open setting for communications using a fixed buffer. (☞ Page 105, Section 10.7)	<ul style="list-style-type: none"> • Disable • Enable
Existence Confirmation	Select whether to use the alive check function. (☞ Page 153, Section 12.6)	<ul style="list-style-type: none"> • No Confirm • Confirm

Item	Description	Setting range
Host Station Port No.	Set the E71 connection port numbers.	<ul style="list-style-type: none"> • DEC: 1025 to 4999, 5003 to 65534^{*2} • HEX: 401_H to 1387_H, 138B_H to FFFE_H^{*2}
Destination IP Address	Set the IP address of the connected device.	<ul style="list-style-type: none"> • DEC: 0.0.0.0 to 255.255.255.255 • HEX: 00.00.00.00 to FF.FF.FF.FF
Destination Port No.	Set port numbers for connections of connected devices.	<ul style="list-style-type: none"> • DEC: 1025 to 65535 • HEX: 401_H to FFFF_H

*1 An E71 is always waiting for an open request from a connected device (communications can be performed during STOP), regardless of the initial timing setting in the Ethernet operation setting. (☞ Page 65, Section 7.1.3)

*2 Do not enter 5000 (1388_H) to 5002 (138A_H). (☞ Page 245, Appendix 2)



- Note the following points when setting port numbers.

○: Can be set, ×: Cannot be set

Connection status	Description	Communication protocol	
		TCP/IP	UDP/IP
	When connecting multiple connected devices, set multiple own station port numbers.	○	○
	When connecting multiple connected devices, set a single own station port number. (However, connections need to be opened by the number of connections.) This setting cannot be configured when the own station is Unpassive.	○	×
	When connecting multiple ports on a connected device, set multiple port numbers of the E71.	○	○
	When connecting multiple ports on a connected device, set a single port number of the E71. (However, connections need to be opened by the number of connections.) This setting cannot be configured when the own station is Unpassive.	○	×
	When connecting the same port on a connected device, set multiple port numbers of the E71. (However, connections need to be opened by the number of connections.)	○	○
	Multiple ports can be set for the same port on a connected device and on the E71 only when paring open has been set.	○	○

- Consult with the network administrator for setting the port numbers of the E71 and the connected device and the IP address of the connected device.

7.2 TCP/IP Communications

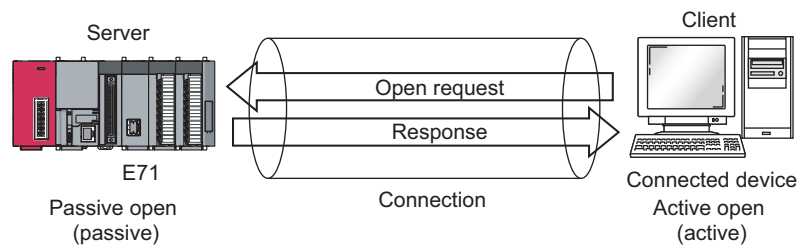
This section describes TCP/IP communications.

7.2.1 Establishing a connection

TCP/IP communications require establishing connections between communication devices. When the server-side device is in standby status after performing a Passive open process, a connection is established after the client-side device issues an open request (Active open process) to the server and a response is received.

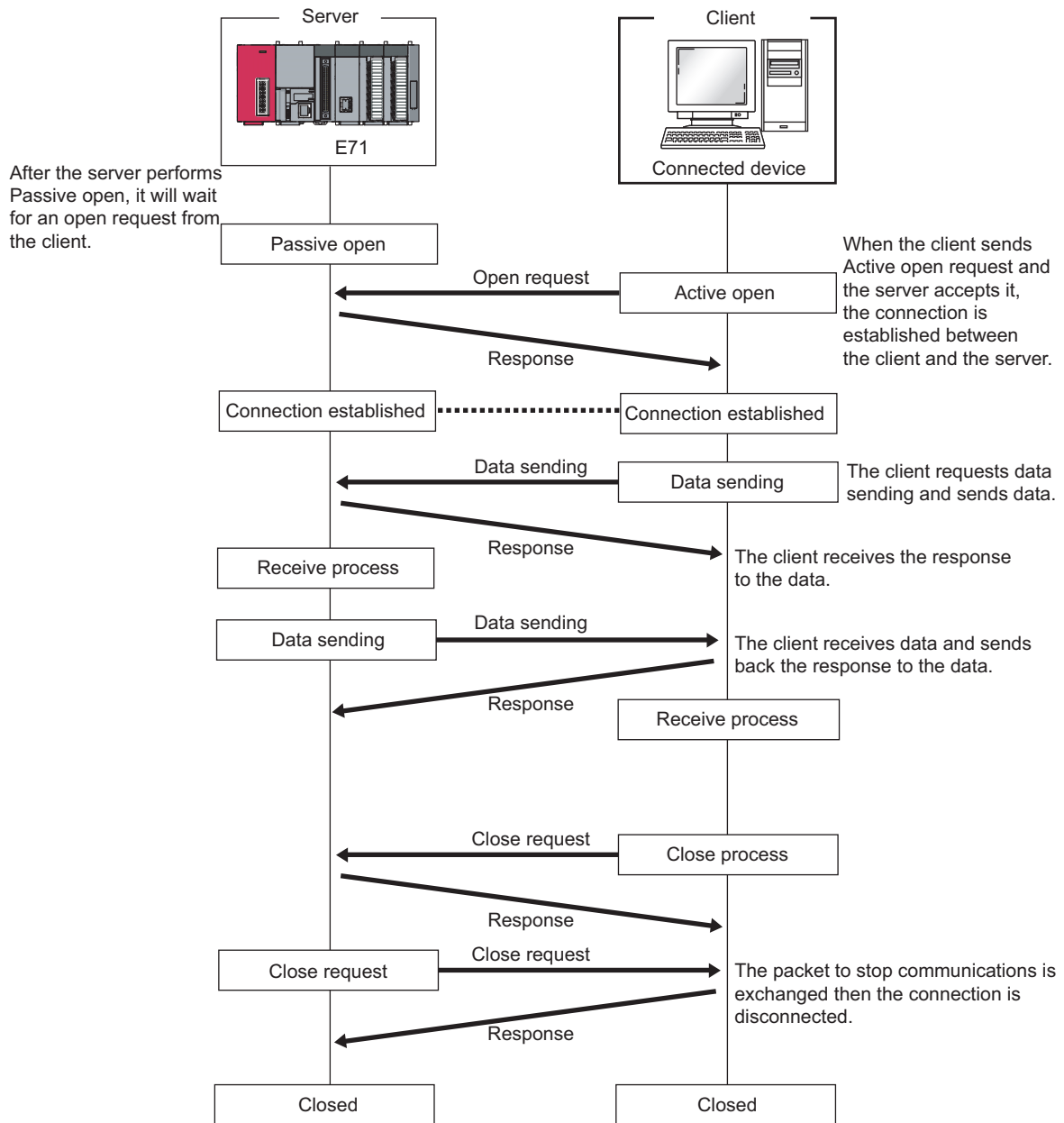
In TCP/IP communications, the system establishes a connection and checks whether each data has arrived at the recipient normally, thereby ensuring data reliability. Compared to UDP/IP communications, however, TCP/IP places a greater load on the line.

Ex. When the E71 is in Passive open status



7.2.2 Communication process

This section describes the process from establishing a connection to terminating communications.

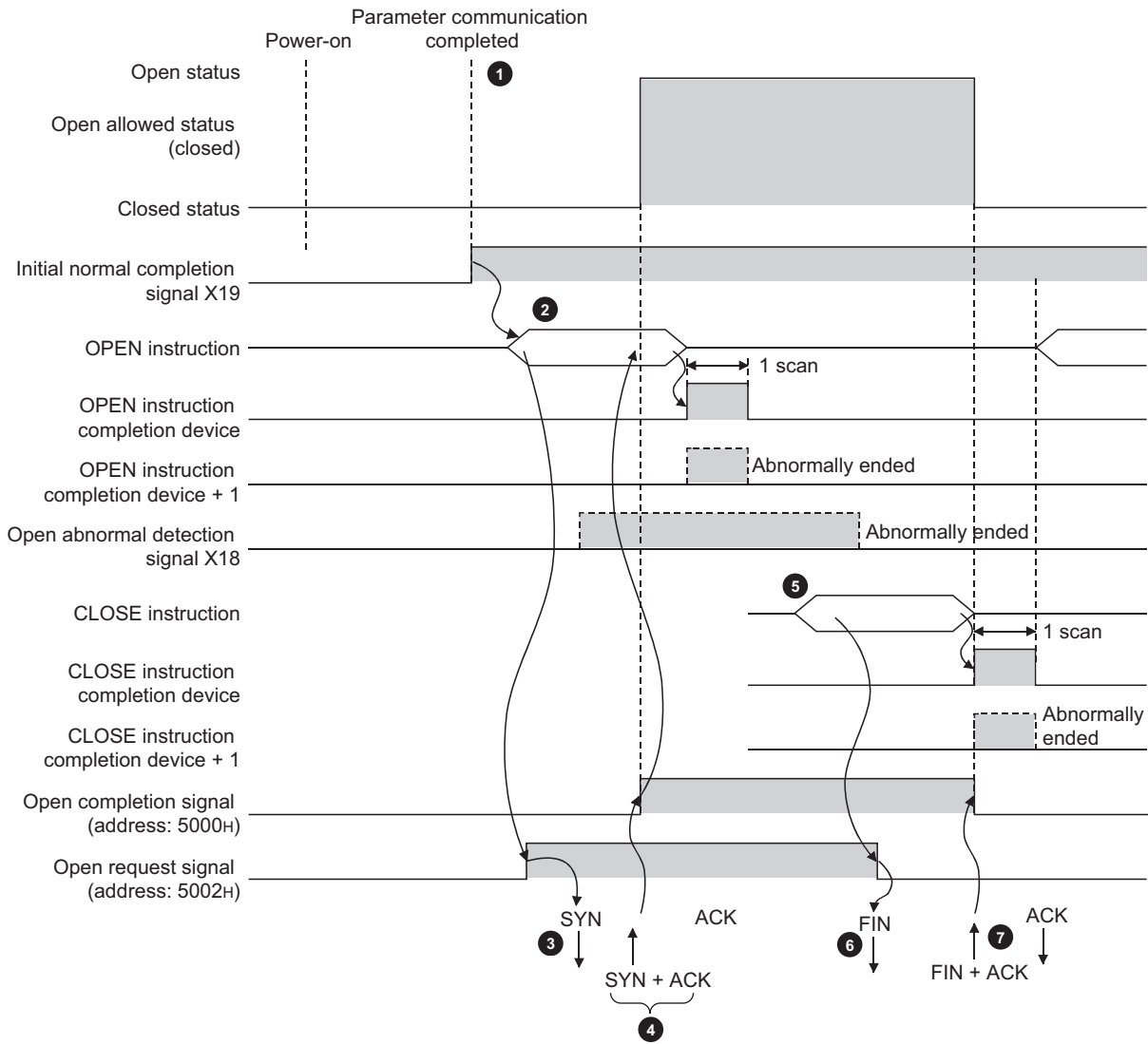


- The corresponding bit of Open completion signal (address: 5000_H) automatically turns off and the communication line is closed in the following situations:
 - A timeout has occurred in the alive check function. (☞ Page 153, Section 12.6)
 - An E71 has received a close request and the ABORT (RST) instruction from a connected device.
 - An E71 has received Alive open request from a connected device again with open completed in TCP.
After an E71 sends an ACK to a connected device, the E71 closes the connection when receiving the RST command from the connected device. If the E71 with a different IP address or port number receives Active open request from the connected device again, the E71 only sends the RST command. (It does not close the connection.)
 - A timeout has occurred upon transmission in TCP.
 - After the connected device sends a close request to the E71, wait for 500ms or more before performing an open process again.
-

7.2.3 Active open procedure

Active open is a connection method that performs an active open process on a connected device in Passive open wait status. The procedure that the E71 performs an Active open process is as follows. For the OPEN/CLOSE instructions, refer to "DEDICATED INSTRUCTIONS". (☞ Page 155, CHAPTER 13)

Ex. Open/close processes for the connection number 1



- ① After parameter communications, check that the initial process of the E71 is normally completed.
(Initial normal completion signal (X19): ON)
 - ② Start the open process using the OPEN instruction. (Open request signal (address: 5002_H (b0)): ON)
 - ③ The E71 performs the open process. (The E71 sends an open request (SYN) to the connected device.)
 - ④ When the open process is normally completed, data communications are enabled.*¹
 - ⑤ Start a close process using the CLOSE instruction. (Open request signal: OFF)
 - ⑥ The E71 performs the close process. (The E71 sends a close request (FIN) to the connected device.)
 - ⑦ When the close process is normally completed, data communications terminate.*²
- *1 If the E71 sends an SYN and the connected device returns a RST, Open abnormal detection signal (X18) turns on immediately and the open process terminates. (Open abnormal end)
- *2 If an ACK and a FIN are not returned within the time specified by the TCP end timer value, the E71 forcibly closes the connection (sends a RST). (Close abnormal end)

7.2.4 Passive open procedure

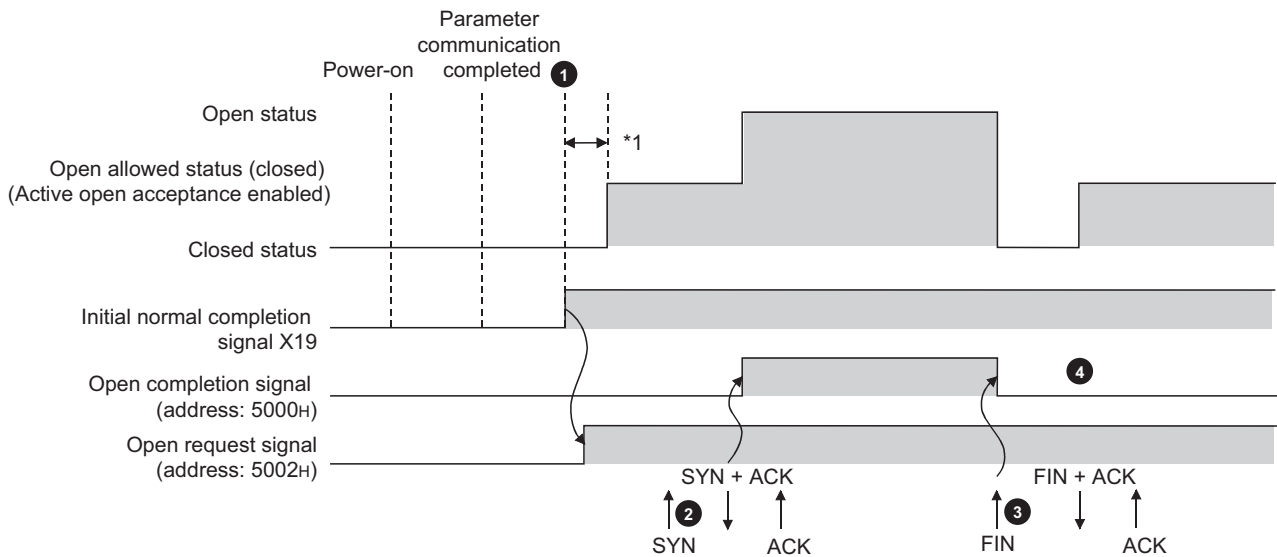
The E71 has the following two connection methods for Passive open.

- Unpassive: Performs a passive open process on connections for all devices connected to the network, regardless of the IP address and port number of the destination device.
- Fullpassive: Performs a passive open process on connections for the connected device specified by the IP address and port number.

Under the Passive open method, the open/close processes are performed using the procedures described below, according to the Ethernet operation setting.

(1) When "Always wait for OPEN (Communication possible at STOP time)" is selected for the Ethernet operation setting

As the E71 is set to always wait for open, a connection is established when an Active open is requested from the connected device. This eliminates the need for open/close processing programs on the E71 side.



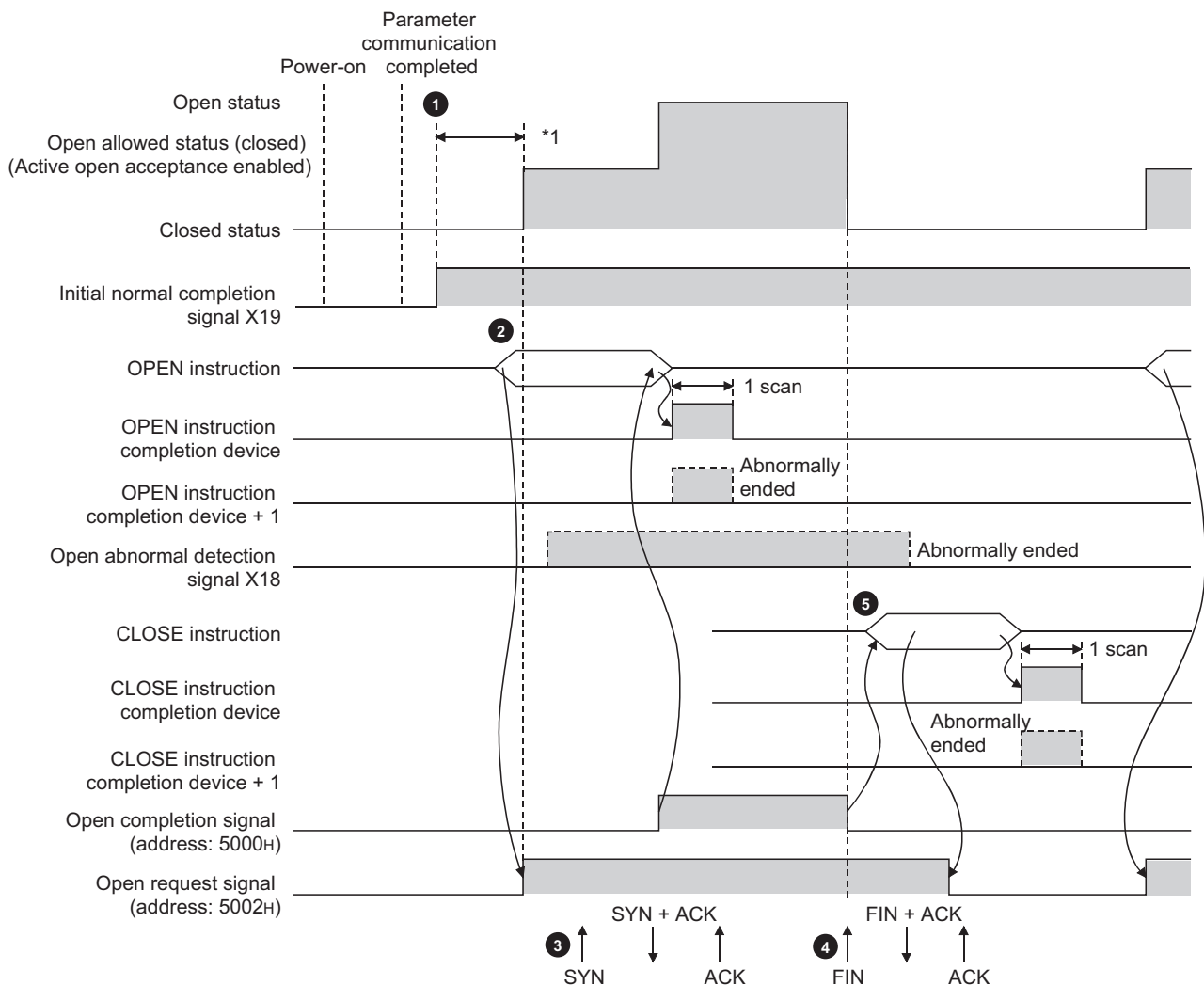
- ➊ After parameter communications, check that the initial process of the E71 is normally completed. (Initial normal completion signal (X19): ON)
After the initial process is normally completed, the connection enters open enable status and the E71 waits for an open request from the connected device.
 - ➋ The E71 starts the open process upon receiving the open request (SYN) from the connected device. When the open process is normally completed, Open completion signal (address: 5000_H (b0)) turns on and data communications are enabled.
 - ➌ The E71 starts the close process upon receiving the close request (FIN) from the connected device. When the close process is completed, Open completion signal (address: 5000_H (b0)) turns off and data communications are disabled.
 - ➍ After the internal process of the E71 is completed, the connection returns to the OPEN request wait status.
- *1 An open request (SYN) received after the normal completion of an initial process and before the E71 is in the OPEN request wait status will cause an error, and the E71 sends a connection forced close (RST).

When the open/close processes are performed from the E71 side using a dedicated instruction, even if "Always wait for OPEN" has been selected for the Ethernet operation setting, the connection will not return to the open acceptance enable status after the close process is completed.

(2) When "Do not wait for OPEN (Communications impossible at STOP time)" is selected for the Ethernet operation setting

Executing the OPEN/CLOSE instructions is required on the E71 to make the E71 enter open/close wait status before receiving an open/close request from the connected device. When the open process is normally completed, data sending and receiving are enabled. For the OPEN/CLOSE instructions, refer to "DEDICATED INSTRUCTIONS". (Page 155, CHAPTER 13)

Ex. Open/close processes for the connection number 1



- ① After parameter communications, check that the initial process of the E71 is normally completed.
(Initial normal completion signal (X19): ON)
- ② Start the open process using the OPEN instruction. (Open request signal (address: 5002_H (b0)): ON)
- ③ The E71 starts the open process upon receiving the open request (SYN) from the connected device. When the open process is normally completed, Open completion signal (address: 5000_H (b0)) turns on and data communications are enabled.
- ④ The E71 starts the close process upon receiving the close request (FIN) from the connected device. When the close process is completed, Open completion signal (address: 5000_H (b0)) turns off and data communications are disabled.
- ⑤ Start the close process using the CLOSE instruction. (Open request signal: OFF)

*1 An open request (SYN) received after the normal completion of an initial process and before the E71 is in the OPEN request wait status will cause an error, and the E71 sends a connection forced close (RST).

Point

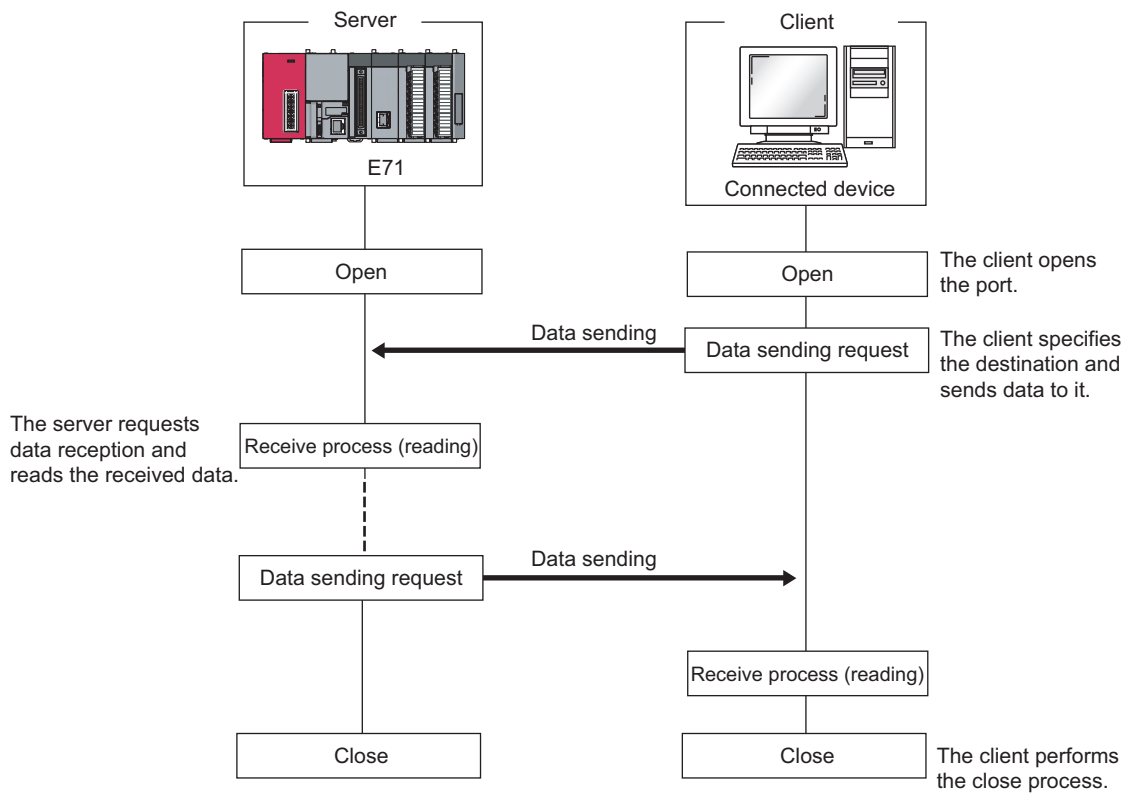
- Change connection settings before executing the OPEN instruction.
- Once an open process is performed, an open request cannot be canceled before the open process is completed. Perform a close process (Execute the CLOSE instruction.) after the open process is completed.

7.3 UDP/IP Communications

This section describes UDP/IP communications. In UDP/IP communications, the system does not establish a connection and does not check whether each data has arrived at the destination normally, thereby reducing the load on the line. However, UDP/IP communications do not guarantee data reliability as the TCP/IP communications do.

7.3.1 Communication process

Unlike TCP/IP communications, UDP/IP communications do not require connections to be established with connected devices.



Point

- If a timeout has occurred in the alive check function, the corresponding bit of Open completion signal (address: 5000_H) automatically turns off and the communication line is closed.
- After the connected device sends a close request to the E71, wait for 500ms or more before performing an open process again.

7.3.2 Open procedure

The open/close processes are performed by the procedures described below, according to the Ethernet operation setting.

(1) When "Always wait for OPEN (Communication possible at STOP time)" is selected for the Ethernet operation setting

After the E71-connected station is started up, the connection in UDP/IP communications automatically opens and data sending/receiving are enabled. Creating open/close processing programs is not required.

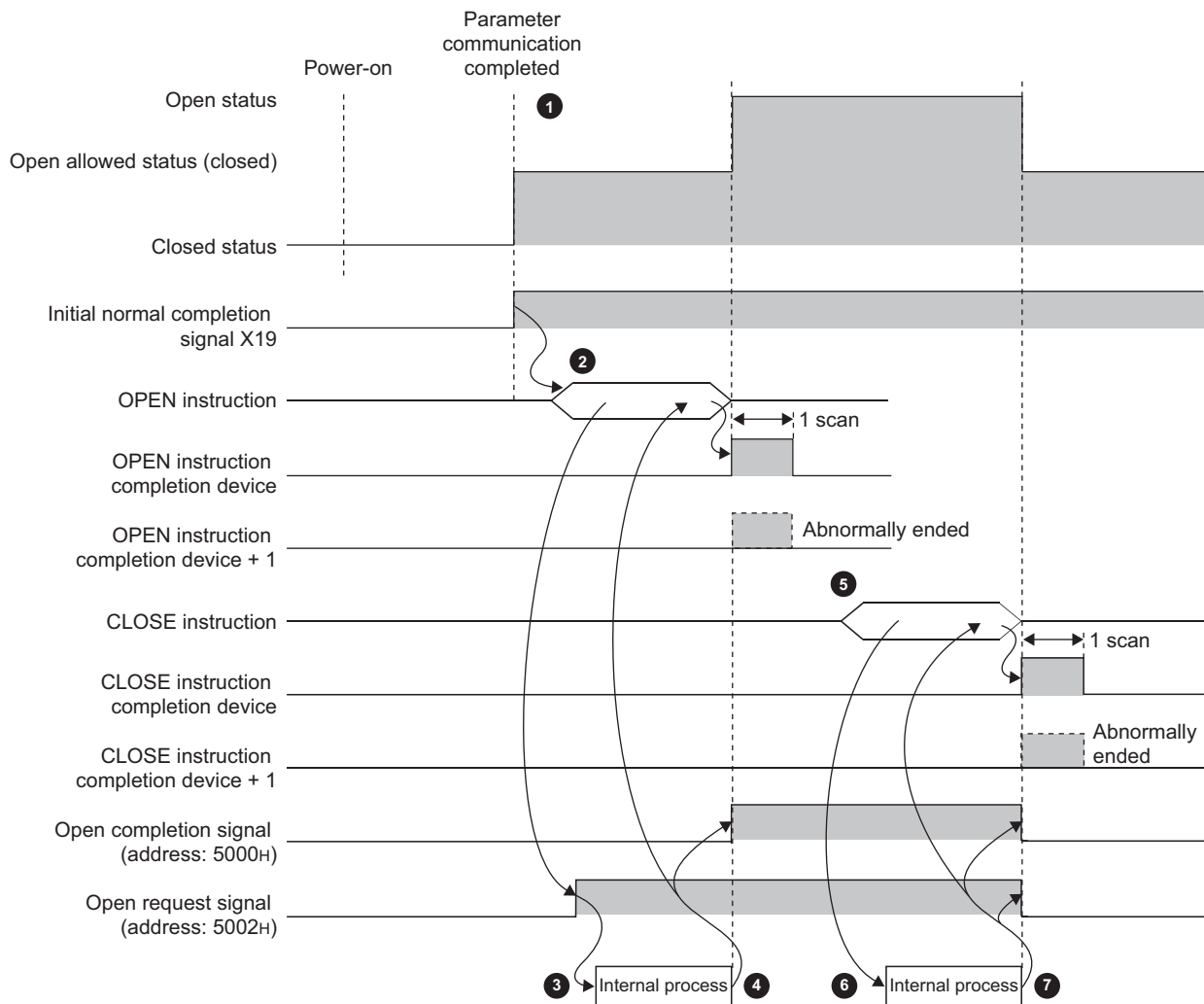
Point

When open/close processes are performed from the E71 side using a dedicated instruction with "Always wait for OPEN (Communication possible at STOP time)" set in the Ethernet operation setting, all open/close processes with the subsequent connected devices must be performed in a program.

(2) When "Do not wait for OPEN (Communications impossible at STOP time)" is selected for the Ethernet operation setting

Executing the OPEN/CLOSE instructions is required on the E71 to make the E71 enter open/close wait status open/close wait status before receiving an open/close request from the connected device. When the open process is normally completed, data sending and receiving are enabled. For the OPEN/CLOSE instructions, refer to "DEDICATED INSTRUCTIONS". (☞ Page 155, CHAPTER 13)

Ex. Open/close processes for the connection number 1



- ① After parameter communications, check that the initial process of the E71 is normally completed. (Initial normal completion signal (X19): ON)
- ② Start the open process using the OPEN instruction. (Open request signal (address: 5002_H (b0)): ON)
- ③ The E71 performs the open process. (Internal process only)
- ④ Data communications are enabled when the open process is normally completed.
- ⑤ Start the close process using the CLOSE instruction. (Open request signal: OFF)
- ⑥ The E71 performs the close process. (Internal process only)
- ⑦ When the close process is normally completed, data communications terminate.

CHAPTER 8 CONNECTING MELSOFT PRODUCTS AND A GOT

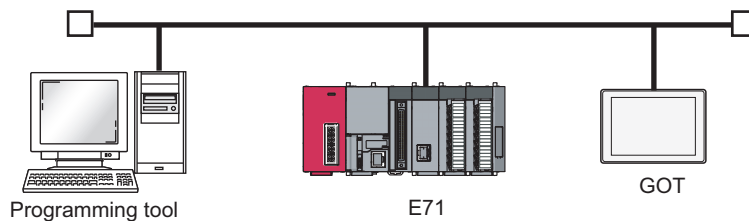
This chapter describes the connection of the E71 with MELSOFT products (such as a programming tool and MX Component) and the GOT.

8.1 Applications

This section describes applications according to a connection type.

(1) Programming and monitoring over Ethernet

In Ethernet, a programming tool can create programming of a programmable controller and monitor a programmable controller (MELSOFT connection), and the GOT can monitor and test a programmable controller. Remote operations making full use of the Ethernet capability, long-distance connectivity and high-speed communications, are achieved.



(2) Connecting multiple products

Multiple MELSOFT products and GOTs can be connected using MELSOFT application communication ports.

Point

The connection used in the connection with MELSOFT products is only for data communications with the MELSOFT products and cannot be used for data communications with connected devices other than MELSOFT products.

(a) Connection using the TCP/IP communications

The E71 can connect to up to 17 MELSOFT products using one dedicated system connection and 16 user connections. GOTs cannot be connected using the TCP/IP communications. (Use the UDP/IP communications.)

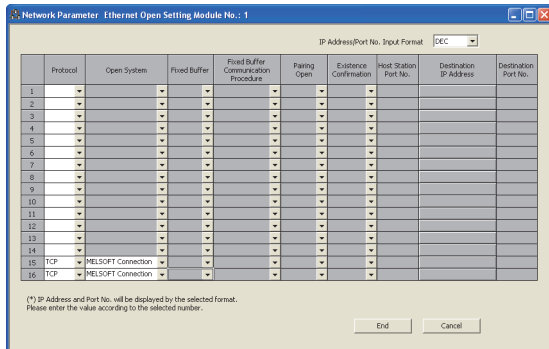
(b) Connection using the UDP/IP communications

The E71 can connect a MELSOFT product and a GOT using one dedicated system connection.

8.2 Data Communication Procedure

This section describes the data communication procedure in the MELSOFT connection.

Ex. Connecting the E71 and a personal computer (programming tool)

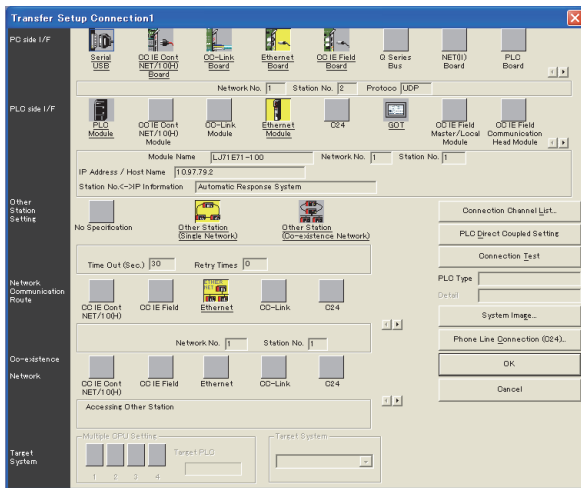


1. Write the parameter settings (see Page 67, Section 7.1.4) in the CPU module. Configure connections by the number of connected products in the open setting only when connecting multiple MELSOFT products. In the following cases, the open setting is unnecessary.

- Only one product is connected in the TCP/IP communications.
- The network is running in the UDP/IP communications.

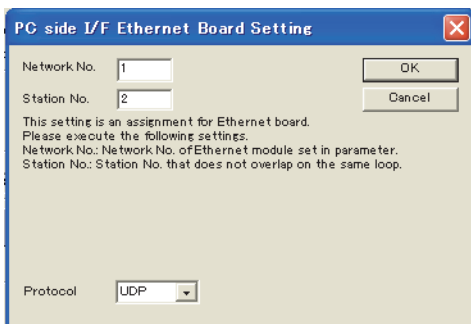
2. Set the connection target in the programming tool.

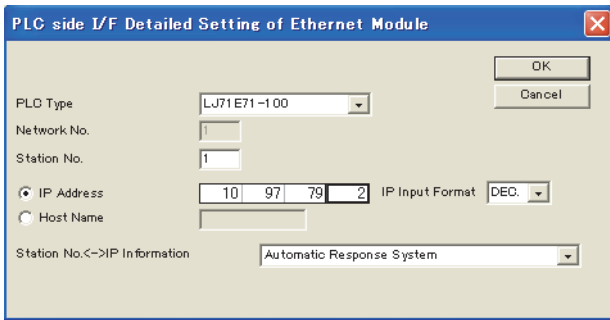
Connection destination window ⇒ [Connection1]



3. Select "Ethernet Board" under "PC side I/F" and double-click the item. The window shown to left appears.

On the window, enter the network number and station number of the personal computer, not of the E71. Determine the network number and station number of the personal computer as follows:
The network number must be the same as that of the E71.
The station number must be different from that of the E71 on the same network.






4. Select "Ethernet Module" under "PLC side I/F" and double-click the item. The window shown to left appears. Set the station number and the IP address according to the network parameters.
5. Set "Other Station Setting" and "Network Communication Route" when required.
6. A Passive open process is performed on all the MELSOFT products connected to the network. (The E71 will wait for an Active open request to itself.)

CHAPTER 9 MC PROTOCOL COMMUNICATIONS

Using MC protocol communications, connected devices that can send and receive data in accordance with the MC protocol can access a CPU module. Because an E71 processes and sends/receives data based on commands from connected devices, no programs for data communications are required on the programmable controller side.

For MC protocol communications, refer to the following.

 MELSEC-Q/L MELSEC Communication Protocol Reference Manual

9.1 Applications

Applications of MC protocol communications are as follows.

(1) Data read/write

Data can be read/written from/to a CPU module device and intelligent function module buffer memory. This allows a connected device to monitor the CPU module operation, perform data analysis, and manage production control.

(2) File read/ write

Files such as a program and parameter stored in a CPU module can be read/written. This allows a connected device to manage CPU module files and switch execution programs.

(3) Remote control of a CPU module

The remote operation allows a connected device to remotely operate a CPU module.

(4) Turning off the COM.ERR. LED

The COM.ERR. LED on an E71 can be turned off from a connected device.

(5) Remote password lock/unlock

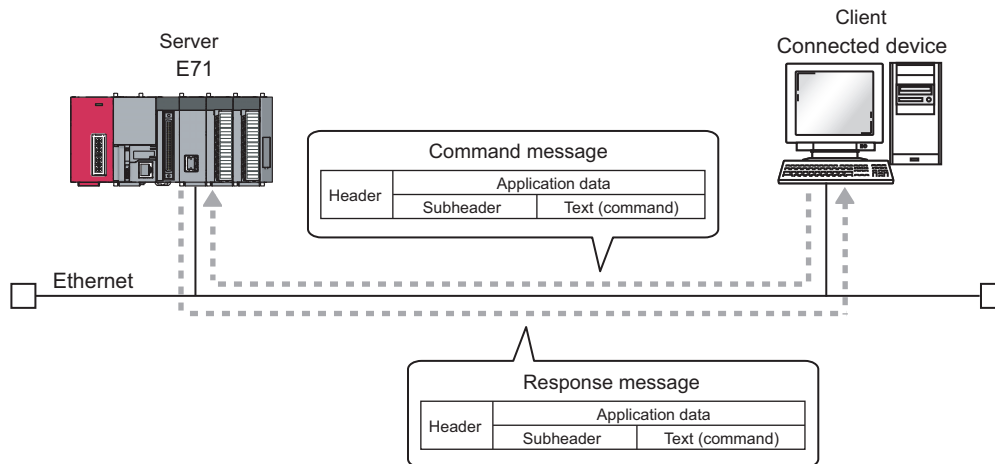
A remote password can be locked/unlocked from a connected device.

(6) Access to a programmable controller in other stations over other networks

In a system containing CC-Link IE Controller Network, CC-Link IE Field Network, MELSECNET/H, MELSECNET/10, and Ethernet, access to a programmable controller in other stations can be performed over each network from a connected device.

9.2 Communication Structure

When a connected device sends a message to an E71 in the MC protocol message format, the E71 performs the process according to the message. During communications, a CPU module including the E71 operates as a server, and the connected device (a personal computer or other terminals) operates as a client. Based on the command message received from the client, the server (the E71) automatically sends an appropriate response message.



The frames that can be used with an E71 are as follows.

- 4E frame
- QnA-compatible 3E frame
- A-compatible 1E frame

9.3 Data Communication Procedure

The following is a data communication procedure for MC protocol communications.

1. Set the parameters. (☞ Page 86, Section 9.4)
2. Perform the open process and establish a connection between the E71 and the connected device. (☞ Page 70, Section 7.2, Page 78, Section 7.3)
3. Once the connection is established, the connected device sends an MC protocol message.
4. The connection is closed after the communications are completed.

Point

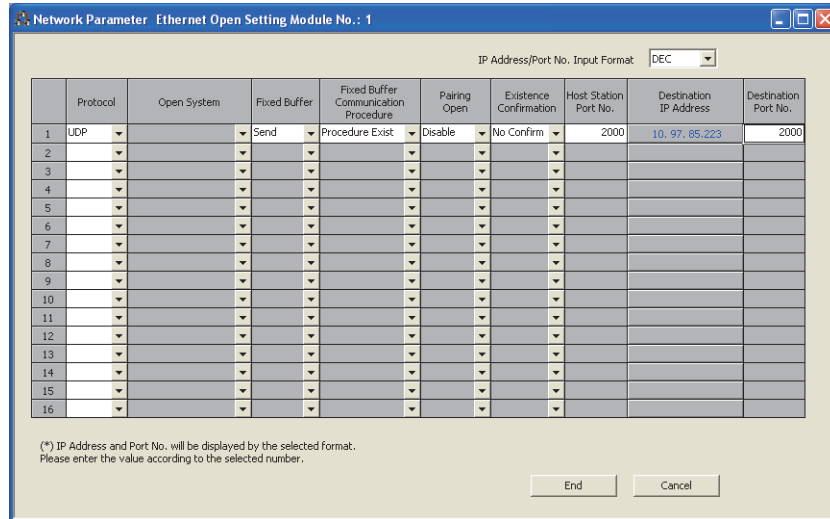
In the cases described below, an E71 performs a remote password check when the connected device accesses the programmable controller. If communications cannot be performed, unlock the remote password. (☞ Page 141, Section 12.3.2)

- When a remote password has been set in the CPU module
- When the connection for data communications with the connected device is to be subjected to the remote password check

9.4 Parameter Setting

Set the following parameters for MC protocol communications.

- Basic setting (☞ Page 64, Section 7.1.2)
- Ethernet operation setting (☞ Page 65, Section 7.1.3)
- Open setting (☞ Page 67, Section 7.1.4)



Item	Description	Setting range
Fixed Buffer Communication Procedure	Select the communication method for communications using a fixed buffer. For MC protocol communications, select "Procedure Exist".	Procedure Exist

CHAPTER 10 COMMUNICATIONS USING A FIXED BUFFER

This chapter describes communications using a fixed buffer.

10.1 Applications

In communications using a fixed buffer, a programmable controller can actively send data; therefore, data can be sent from the programmable controller to the host system when an error occurs in machine equipment or some other conditions are satisfied. A maximum of 1K word of data can be sent or received between programmable controllers or between a programmable controller and the host system.

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

There are two control methods for communications using a fixed buffer; "Procedure Exist" control method and "No Procedure" control method. This section describes the differences between the "Procedure Exist" and "No Procedure" control methods.

Item	Difference	
	Procedure Exist	No Procedure
Message format	Data is sent/received in a fixed data format.	Data is sent/received in the message format of the connected device.
A response to data receiving	A response to data receiving is sent.	A response to data receiving is not sent.
Data code	Communications can be performed using a binary code or ASCII code.	Communications are performed using only a binary code.
Data length specified using a dedicated instruction	Specified by the number of words.	Specified by the number of bytes.
Amount of application data per communication	Maximum of 1017 words	Maximum of 2046 bytes

Point

A connection with a connected device is dedicated for "No Procedure" communications using a fixed buffer. MC protocol communications and "Procedure Exist" communications using a fixed buffer and random access buffer cannot be performed at the same time as "No Procedure" communications using a fixed buffer.

10.2 Communication Structure

This section describes the structure of communications using a fixed buffer.

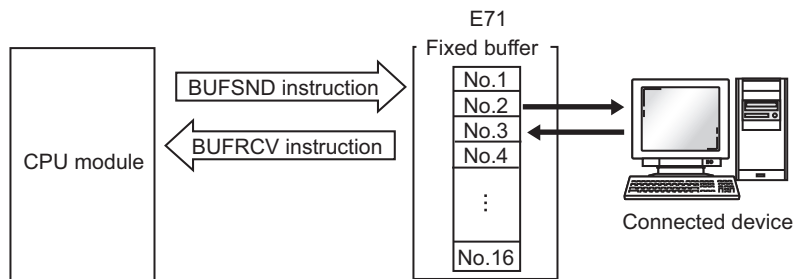
(1) Data flow

In communications using a fixed buffer, data is sent/received using dedicated instructions.

- Sending data: BUFSND instruction
- Receiving data: BUFRCV instruction or BUFRCVS instruction

When the "Procedure Exist" control method is used, the CPU module and the connected device communicate on a 1:1 basis. The CPU module sends/receives data to/from the connected device while handshaking with the connected device.

When the "No Procedure" control method is used, the CPU module sends/receives data to/from the connected device in a non-procedural manner.

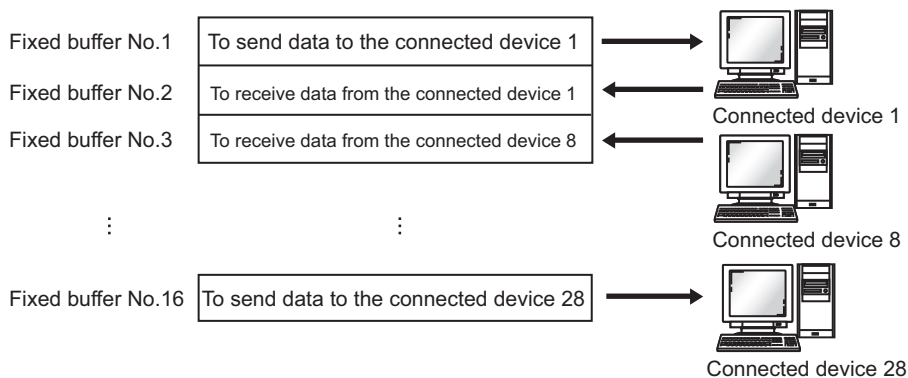


(2) Connected devices where data communications can be performed

Data communications can be performed with the following connected devices.

- Devices in Ethernet where an E71 is connected
- Devices connected with the router relay function

When using each fixed buffer (numbers 1 to 16) as shown in the figure below, set the destination devices and usage conditions (e.g. for sending/receiving and "Procedure Exist"/"No Procedure") in the open setting to fix a connected device to each buffer.



Note the following points when changing connected devices.

- During TCP/IP communications, a connected device can be changed only when the connection with the connected device is not established (when Open completion signal is off).
- During UDP/IP communications, a connected device can be changed regardless of the status of connection with the connected device.

Point

- When changing connected devices, do not use the pairing open or alive check function.
- When "No Procedure" is selected for a connection, the connection is dedicated to "No Procedure" communications using a fixed buffer after the open process is completed. When "Procedure Exist" is selected for a connection, the following data communications can be performed after the open process is completed.
 - MC protocol communications
 - Communications using a fixed buffer using the "Procedure Exist" control method
 - Communications using a random access buffer

(3) Processes during data sending/receiving**(a) When data is sent**

When a CPU module executes the BUFSND instruction, the E71 sends data of the applicable fixed buffer number n to the connected device that is specified in Communication address setting area (addresses: 28_H to $5F_H$) corresponding to the fixed buffer number n .

(b) When data is received

An E71 processes the data that is received from the connected device set in the communication address setting area corresponding to the fixed buffer number n .^{*1}

In addition, when an E71 stores the received data in the corresponding fixed buffer in the receive process, it updates the IP address and destination port number of the connected device in Connection information area (addresses: 78_H to $C7_H$ and 5820_H to $586F_H$) corresponding to the fixed buffer number n .

If data is received from a connected device not set in the connection information area of the buffer memory, an E71 ignores the received data.

*1 With the TCP/IP Unpassive open, data is sent to or received from a connected device stored in the connection information area of the buffer memory.

Point

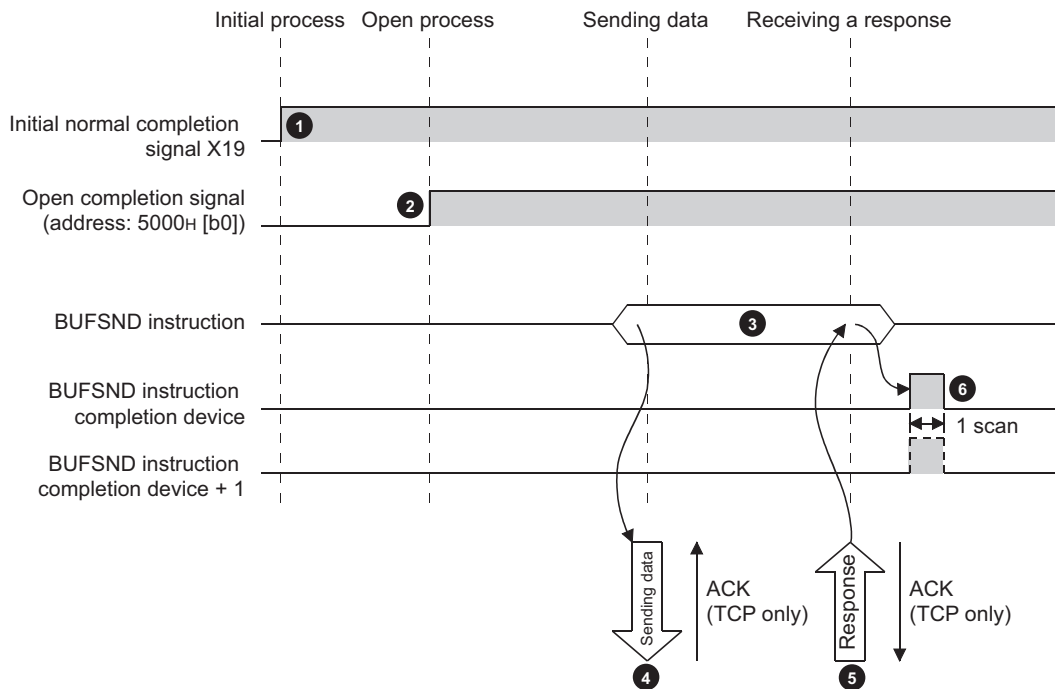
When received data is stored in the corresponding fixed buffer in the receive process of broadcast communications, an E71 updates the destination IP address and port number in Connection information area (address: 78_H to $C7_H$ and 5820_H to $586F_H$) corresponding to the fixed buffer number n .

10.3 Data Sending Procedure

This section describes how an E71 sends data to the connected device.

(1) Procedure exists

The following is the data send process performed in the fixed buffer number 1 area for the connection number 1.



- ① Confirm the normal completion of the initial process. (Initial normal completion signal (X19): ON)
- ② Establish a connection between the E71 and the connected device and confirm the normal completion of the open process of the connection number 1.
- ③ Execute the BUFSND instruction. (The data is sent.)
- ④ The send data in the fixed buffer number 1 is sent to the connected device by the send data length.
- ⑤ When the connected device receives the data from the E71, the connected device sends a response to the E71.
- ⑥ When the E71 receives the response from the connected device, the E71 stops sending data. If a response is not sent within the response monitoring timer value, a data send error occurs.*1

After the data send process abnormally ends, execute the BUFSND instruction again to repeat the send process.

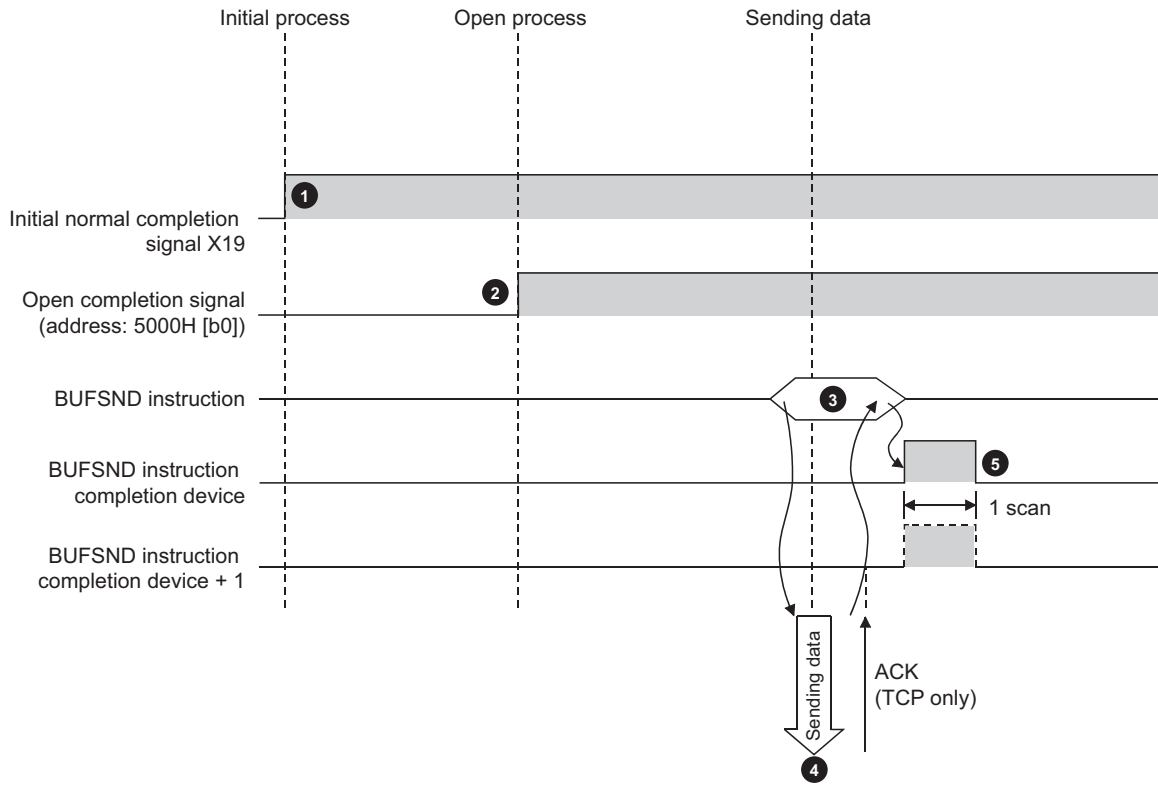
*1 Refer to the section describing the initial setting when adjusting the monitoring timer value. (👉 Page 246, Appendix 3.1)

Point

- The items configured in the open setting become enabled when Open completion signal of an E71 is started up.
- Do not send data (command) before data communications are completed (or a response is received) in response to the previous data (command).
- When an E71 communicates data with multiple connected devices, data can be sent sequentially; however, switching the connected devices before sending/receiving data is recommended to prevent communication problems. When a connection opened using the UDP/IP protocol is used, change the setting values in the communication address setting area before sending or receiving data to switch the connected devices.

(2) No procedure

The following is the data send process performed in the fixed buffer number 1 area for the connection number 1.



- ① Confirm the normal completion of the initial process. (Initial normal completion signal (X19): ON)
- ② Establish a connection between the E71 and the connected device and confirm the normal completion of the open process of the connection number 1.
- ③ Execute the BUF SND instruction. (The data is sent.)
- ④ The send data in the fixed buffer number 1 is sent to the connected device by the send data length.
- ⑤ The E71 stops sending data. After the data send process abnormally ends, execute the BUF SND instruction again to repeat the send process.

Point

During UDP/IP communications, when the internal process of an E71 is normally completed, the data send process ends even if the communication line between a CPU module and the connected device is not connected due to cable disconnection or other causes. It is recommended to set an original communication procedure to send/receive data.

10.4 Data Receiving Procedure

This section describes how an E71 receives data from the connected device. The following methods for receiving data are offered:

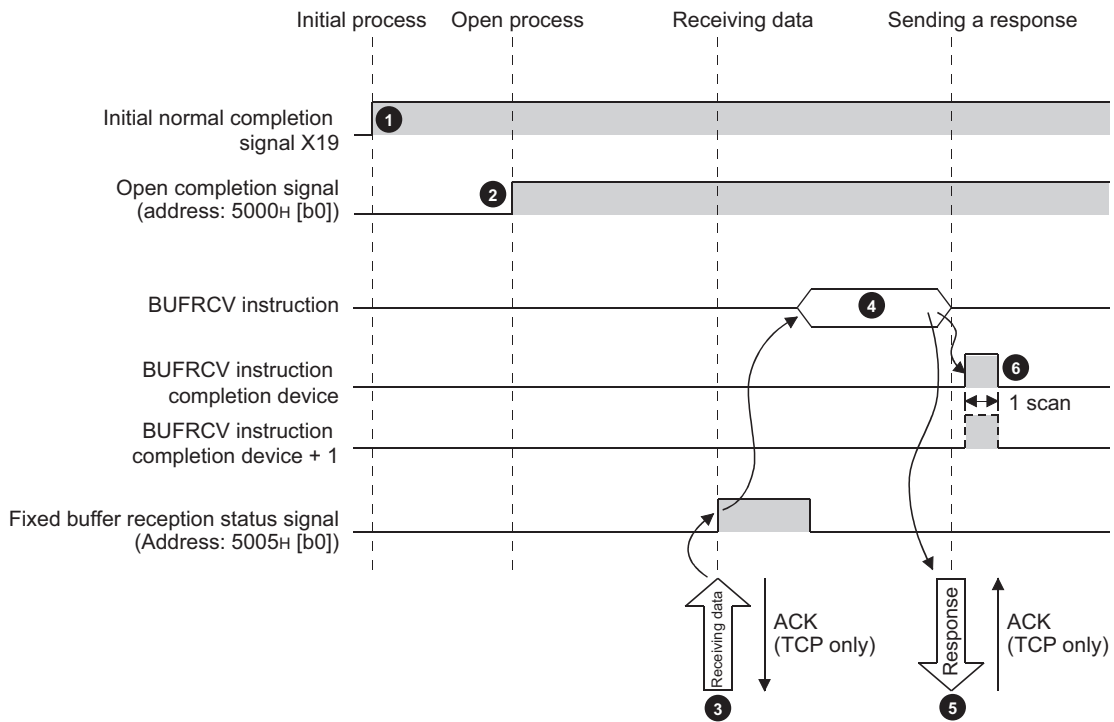
- Data receiving using the main program (BUFRCV instruction)
- Data receiving using an interrupt program (BUFRCVS instruction)

10.4.1 Data receiving using the main program (BUFRCV instruction)

A receive process in the main program is performed using the BUFRCV instruction.

(1) Procedure exists

The following is the data receive process performed in the fixed buffer number 1 area for the connection number 1.



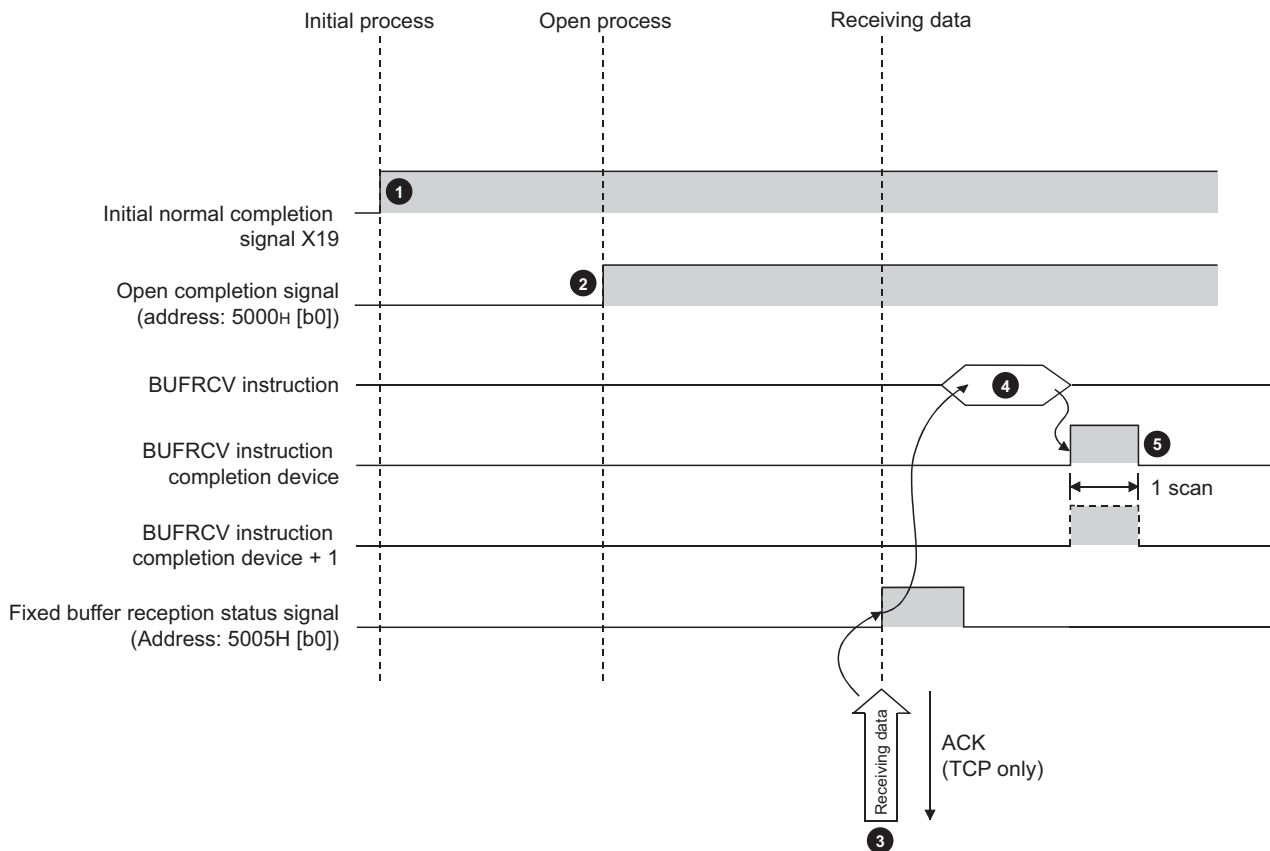
- 1 Confirm the normal completion of the initial process. (Initial normal completion signal (X19): ON)
- 2 Establish a connection between the E71 and the connected device and confirm the normal completion of the open process of the connection number 1.
- 3 Receive data from the destination.
 - Fixed buffer reception status signal (address: 5005_H (b0)): ON
- 4 Execute the BUFRCV instruction to read the receive data length and receive data from the fixed buffer number 1.
 - Fixed buffer reception status signal (address: 5005_H (b0)): OFF
- 5 When the receive data length and the receive data are completely read out, a response is sent to the destination.
- 6 End the receive process. If the data receive process is abnormally ended, execute the BUFRCV instruction again to repeat the receive process.

Point

- The items configured in the open setting become enabled when Open completion signal of an E71 is started up.
- When the buffer memory area that stores Fixed buffer reception status signal is turned off and on, execute the BUFRCV instruction.
- When abnormal data is received, Fixed buffer reception status signal does not turn on. In addition, data is not stored in the fixed buffer number 1 area.

(2) No procedure

The following is the receive process performed in the fixed buffer number 1 area for the connection number 1.



- 1 Confirm the normal completion of the initial process. (Initial normal completion signal (X19): ON)
- 2 Establish a connection between the E71 and the connected device and confirm the normal completion of the open process of the connection number 1.
- 3 Receive data from the connected device.
 - Fixed buffer reception status signal (address: 5005_H (b0)): ON
- 4 Execute the BUFRCV instruction to read the receive data length and receive data from the fixed buffer number 1.
 - Fixed buffer reception status signal (address: 5005_H (b0)): OFF
- 5 End the receive process. If the data receive process is abnormally ended, execute the BUFRCV instruction again to repeat the receive process.

Point 

- The items configured in the open setting become enabled when Open completion signal of an E71 is started up.
 - When the buffer memory area that stores Fixed buffer reception status signal is turned off and on, execute the BUFRCV instruction.
 - When abnormal data is received, Fixed buffer reception status signal does not turn on. In addition, data is not stored in the fixed buffer number 1 area.
-

10.4.2 Data receiving using an interrupt program (BUFRCVS instruction)

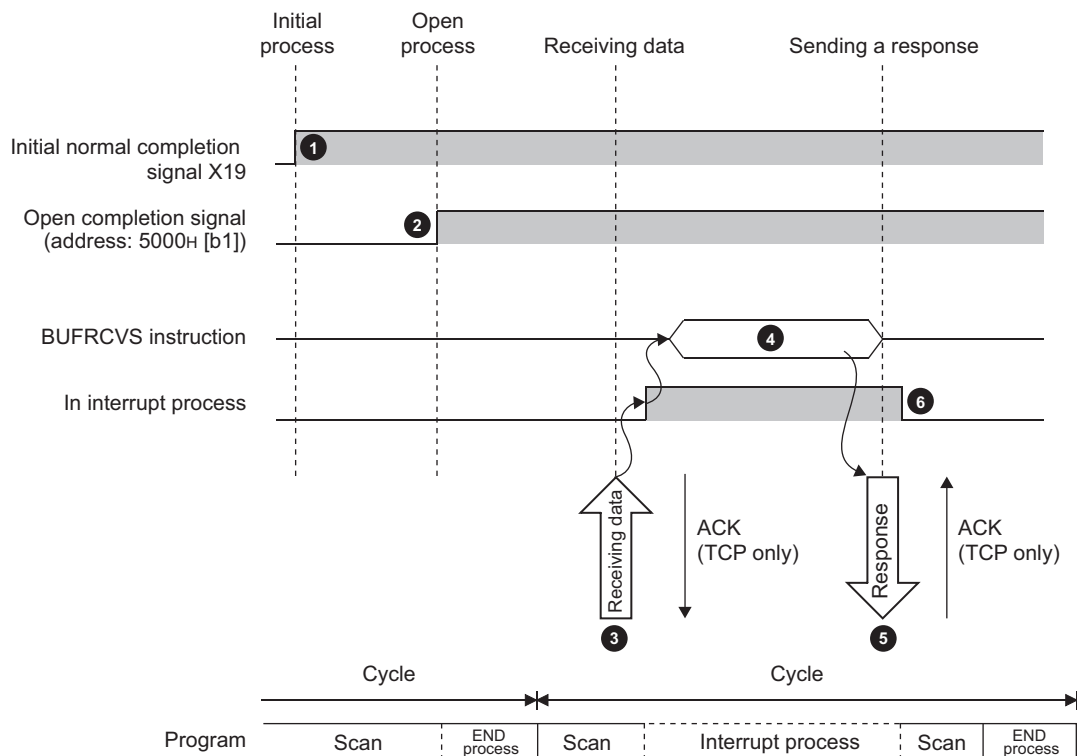
A receive process in an interrupt program is performed using the BUFRCVS instruction. An interrupt program is started up when data is received from a connected device, and receive data for a CPU module can be read.

The following settings are required to use an interrupt program: (☞ Page 98, Section 10.5.1)

- Interrupt pointer setting
- Interrupt settings

(1) Procedure exists

The following is the receive process performed in the fixed buffer number 2 area for the connection number 2.

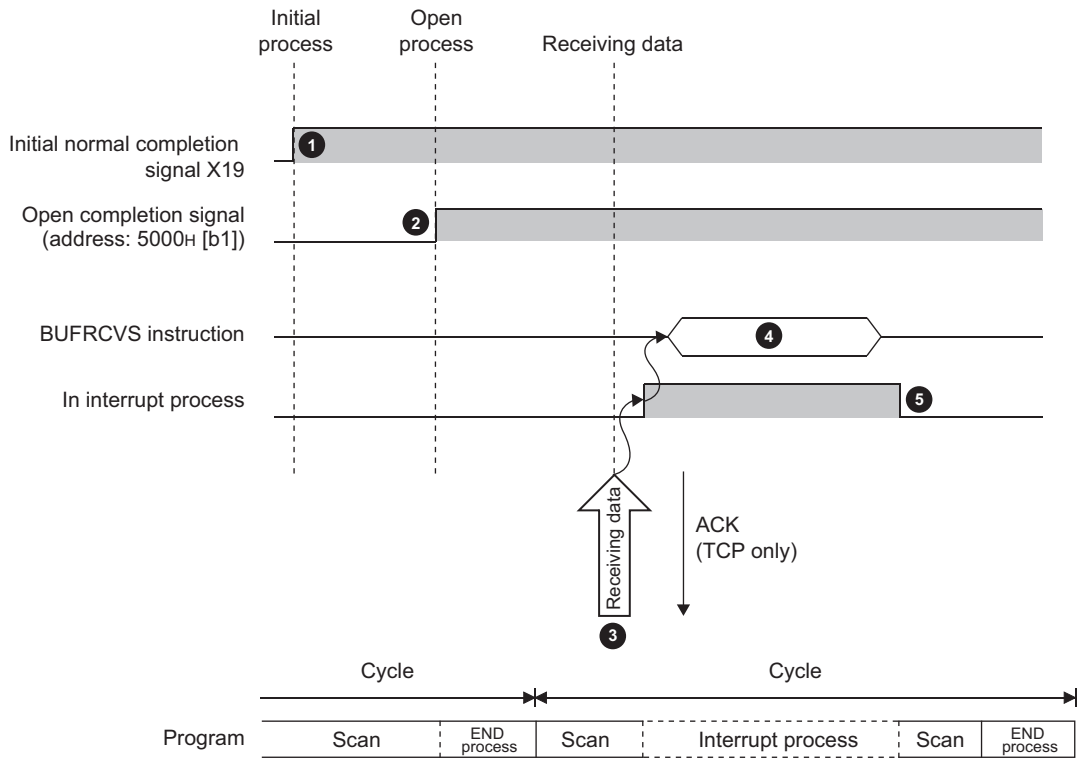


- 1 Confirm the normal completion of the initial process. (Initial normal completion signal (X19): ON)
- 2 Establish a connection between the E71 and the connected device and confirm the normal completion of the open process of the connection number 2.
- 3 Receive data from the destination.
 - Fixed buffer reception status signal (address: 5005_H (b1)): ON
 - Request the CPU module to start up the interrupt program.
- 4 The interrupt program starts up. Execute the BUFRCVS instruction to read the receive data length and receive data from the fixed buffer number 2.
- 5 When the receive data length and the receive data are completely read out, a response is sent to the destination.*1
- 6 End the interrupt program to restart the main program.

*1 If the data receive process is abnormally ended, the E71 does not return a response.

(2) No procedure

The following is the receive process performed in the fixed buffer number 1 area for the connection number 1.



- ① Confirm the normal completion of the initial process. (Initial normal completion signal (X19): ON)
- ② Establish a connection between the E71 and the connected device and confirm the normal completion of the open process of the connection number 1.
- ③ Receive data from the destination.
 - Request the CPU module to start up the interrupt program.
- ④ The interrupt program starts up. Execute the BUFRCVS instruction to read the receive data length and receive data from the fixed buffer number 1.
- ⑤ End the interrupt program to restart the main program.

10.5 Parameter Setting

Set the following parameters to perform communications using a fixed buffer.

- Basic setting (☞ Page 64, Section 7.1.2)
- Ethernet operation setting (☞ Page 65, Section 7.1.3)
- Open setting (☞ Page 67, Section 7.1.4)

	Protocol	Open System	Fixed Buffer	Fixed Buffer Communication Procedure	Pairing Open	Existence Confirmation	Host Station Port No.	Destination IP Address	Destination Port No.
1	UDP		Receive	Procedure Exist	Disable	No Confirm	2000	10.97.85.223	2000
2									
3									
4									
5									
6									
7									
8									
9									
10									
11									
12									
13									
14									
15									
16									

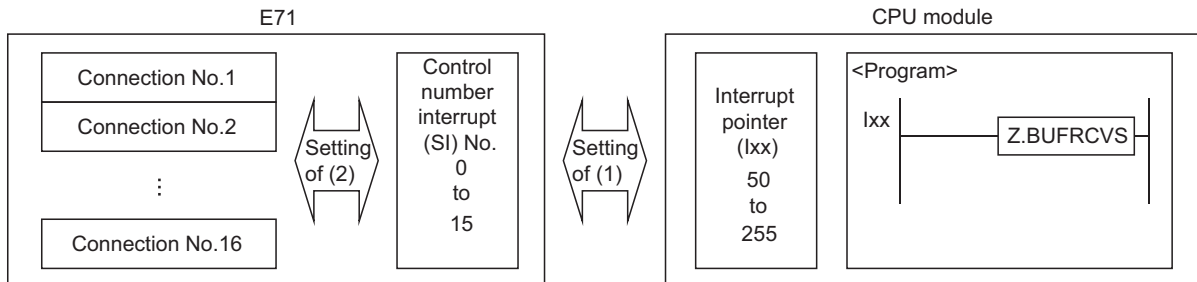
(*) IP Address and Port No. will be displayed by the selected format.
Please enter the value according to the selected number.

End Cancel

Item	Description	Setting range
Fixed Buffer	Select whether to use the fixed buffer for connection with the connected device for sending or receiving.	<ul style="list-style-type: none"> • Send • Receive
Fixed Buffer Communication Procedure	Select the communication method for communications using a fixed buffer.	<ul style="list-style-type: none"> • Procedure Exist • No Procedure
Pairing Open	Select whether to disable or enable the pairing open for communications using a fixed buffer. (☞ Page 105, Section 10.7)	<ul style="list-style-type: none"> • Disable • Enable
Existence Confirmation	Select whether or not to use the alive check function. (☞ Page 153, Section 12.6) Select "No Confirm" to send data by broadcast communications during "No Procedure" communications using a fixed buffer.	<ul style="list-style-type: none"> • No Confirm • Confirm

10.5.1 Parameter setting when using an interrupt program

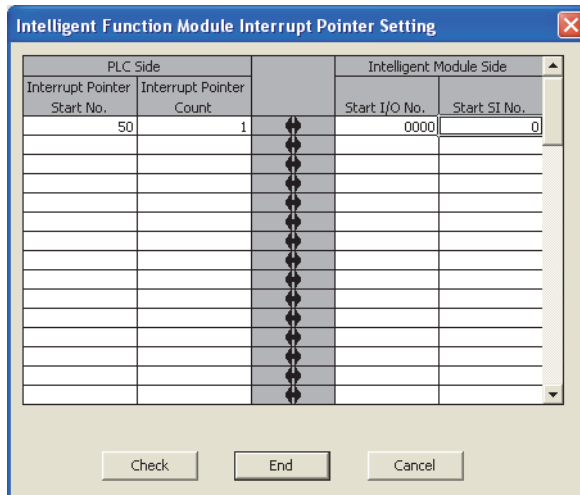
Configure the parameter setting on a programming tool to start up an interrupt program.



(1) Interrupt pointer setting

The following is a setting example.


Project window ⇨ [Parameter] ⇨ [PLC Parameter] ⇨ [PLC System] ⇨ button

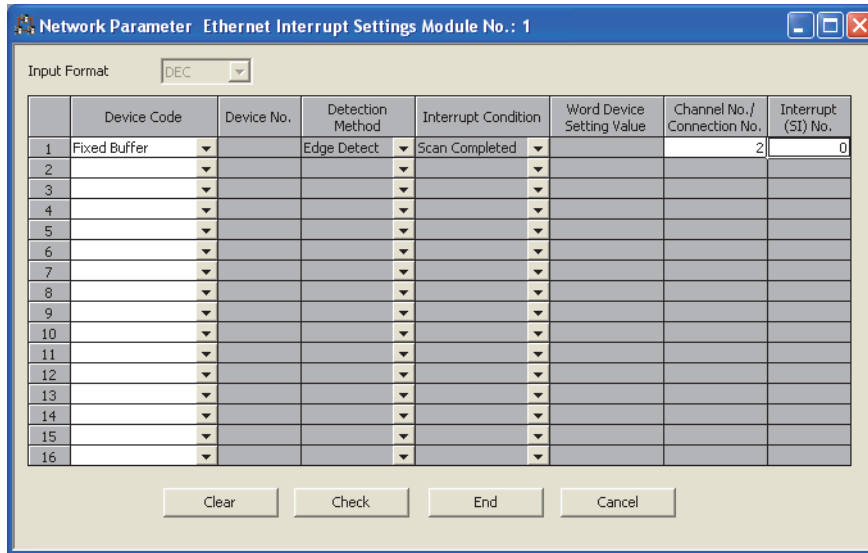


Item		Description	Setting range
PLC Side	Interrupt Pointer Start No.	Set the start number of the interrupt pointer used in a program.	50 to 255
	Interrupt Pointer Count	Set the number of interrupt pointers used in a program.	1 to 16
Intelligent Module Side	Start I/O No.	Set the start I/O number of an E71.	0000 _H to 0FE0 _H
	Start SI No.	Among maximum 16 interrupt (SI) numbers (control numbers on the Ethernet module side) to be set in the interrupt program, set the smallest numbers 0 to 15.	0 to 15

(2) Interrupt settings

The following is a setting example.

Project window ⇒ [Parameter] ⇒ [Network Parameter] ⇒ [Ethernet/CC IE Field] ⇒ Select "Ethernet" under "Network Type". ⇒  button



Item	Description	Setting range
Input Format	The input format of each data is decimal number only.	-
Device Code	Select a device code. Select "Fixed Buffer" in this setting.	Fixed Buffer
Channel No./Connection No.	Set the connection number of the fixed buffer to trigger the interrupt program.	1 to 16
Interrupt (SI) No.	Set the interrupt control number for the E71 for when an interrupt request is issued from the E71 to the CPU module.*1	0 to 15

*1 Do not set the same interrupt (SI) numbers as those to be set for an interrupt of other communications using a fixed buffer or the RECV instruction.

Point

Interrupt (SI) numbers can be assigned by the user to receive up to 16 communications using a fixed buffer and communications using the RECV instruction in an interrupt program. The user must manage the interrupt (SI) numbers assigned for receiving data.

10.5 Parameter Setting
10.5.1 Parameter setting when using an interrupt program

10.6 Data Format

Communication data consists of a header and application data.

10.6.1 Header

The header for TCP/IP or UDP/IP is used. Because an E71 automatically adds and deletes a header, the user setting is not required.

(1) Detailed header sizes

(a) TCP/IP

Ethernet (14 bytes)	IP (20 bytes)	TCP (20 bytes)
------------------------	------------------	-------------------

(b) UDP/IP

Ethernet (14 bytes)	IP (20 bytes)	UDP (8 bytes)
------------------------	------------------	------------------

10.6.2 Application data

This section describes the application data configuration.

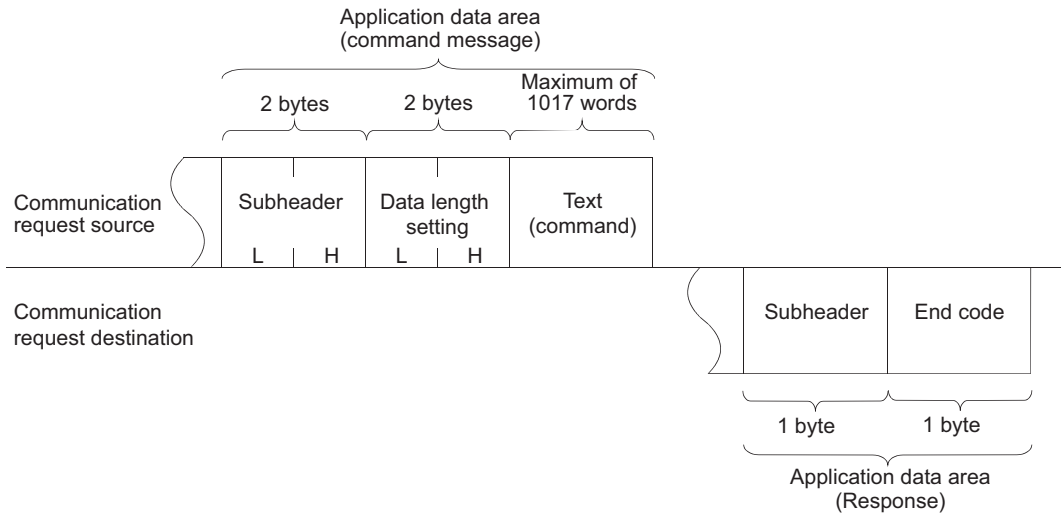
(1) Procedure exists

In the application data for communications with the "Procedure Exist" control method, the following data code is expressed in a binary code or an ASCII code. To switch between a binary code and an ASCII code, use the Ethernet operation setting.

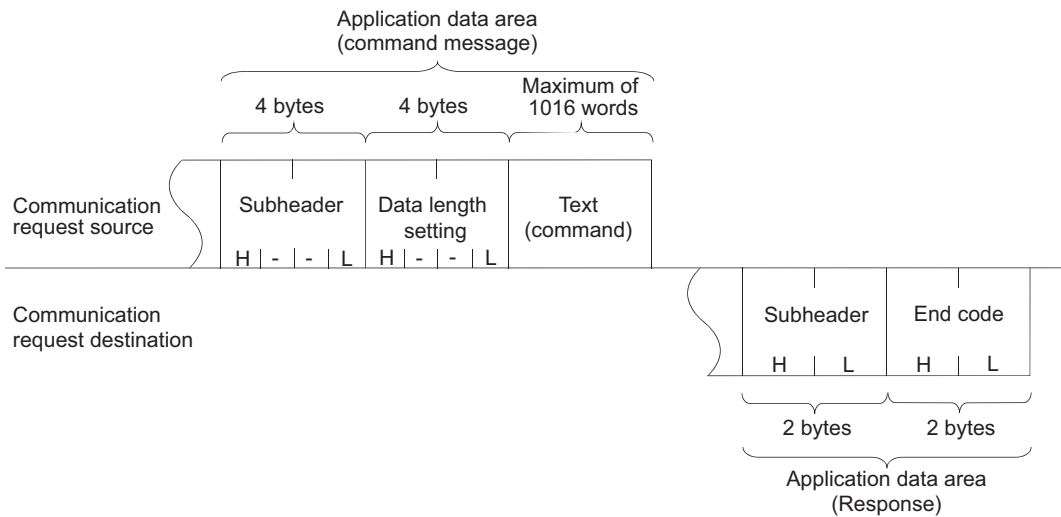
 Project window ⇨ [Parameter] ⇨ [Network Parameter] ⇨ [Ethernet/CC IE Field] ⇨ Select "Ethernet" under "Network Type". ⇨  button

(a) Format

- Communications using a binary code

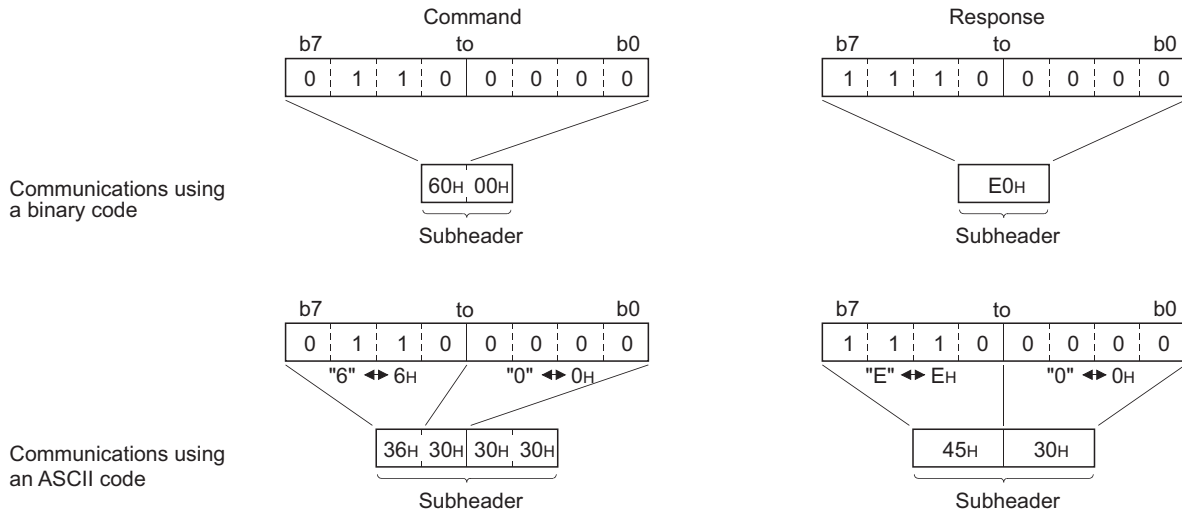


- Communications using an ASCII code



(b) Subheader

Because an E71 automatically adds and deletes a header, the user setting is not required.



(c) Data length setting

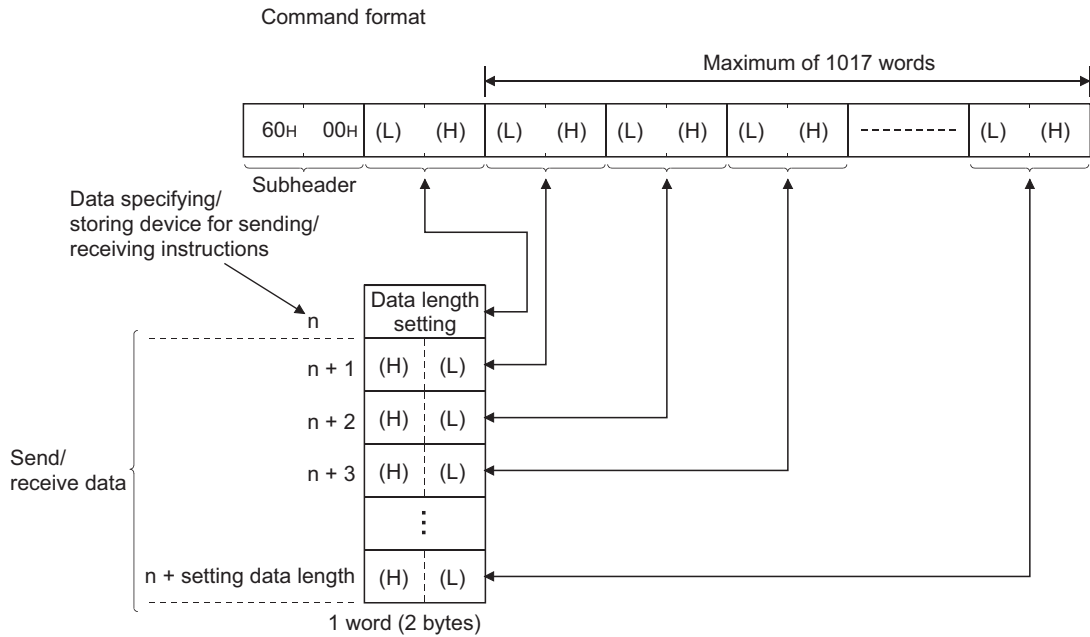
The following are the data sizes of a text (command).

- Communications using a binary code: Maximum of 1017 words
- Communications using an ASCII code: Maximum of 508 words

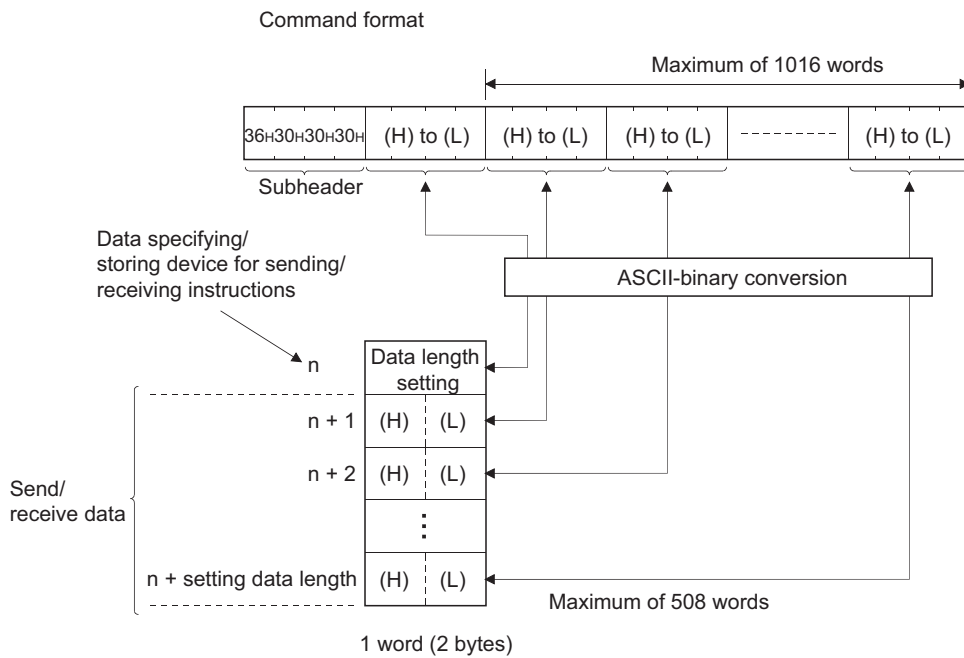
(d) Text (command)

The following is the format of a command/response.

- Communications using a binary code



- Communications using an ASCII code



(e) End codes

An error code is stored in the end code added to a response. For the error codes, refer to the error code list.

(☞ Page 210, Section 14.7.1)

End codes are stored in the complete status area (in the control data) of the BUFSND and BUFRCV instructions, as well as the communication status storage area of the buffer memory.

Point

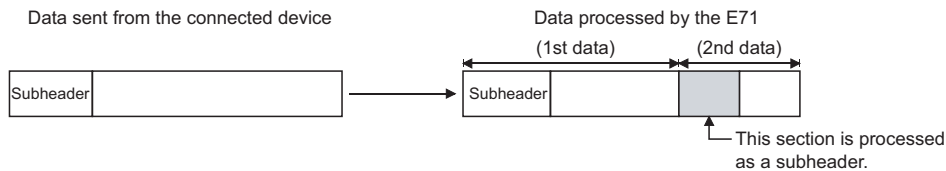
If an error code related to MC protocol communications or communications using a random access buffer is stored during communications using a fixed buffer, it may be caused by the following.

Cause	Action
The data length value specified in the application data of the message sent from a connected device to the E71 is different from the actual text data size.	Specify the actual text data size for the data length in the application data. (Refer to "Remark" in this page.)
The subheader of the message sent from a connected device to the E71 is incorrect.	Correct the subheader specified in the application data.

Remark

Depending on the restrictions of the buffers of the own station and destination station, data may be divided for communications. Data received separately is restored (reassembled) by the E71 for communications. The received data is restored (reassembled) based on the data length in the communication data. The E71 performs the following processes if data in the communication data is incorrect.

- If the data length specified immediately after the subheader is smaller than the amount of text data received
 - 1) The data immediately after the text data equivalent to the data length specified immediately after the subheader is regarded as the next message.
 - 2) Because the header of each message is a subheader, an E71 processes data according to the code in the subheader.
 - 3) If the subheader contains a code that is not recognized by an E71, the E71 sends a response notifying about an abnormal end to the connected device.

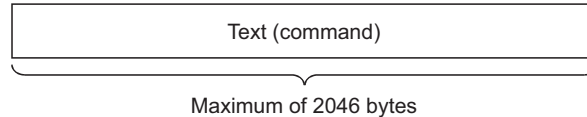


In the figure above, the E71 returns a response containing a code obtained by changing the most significant bit of the code processed as a subheader to 1. For example, if the subheader of a command is 65_H, the subheader of the response is E5_H.

- If the data length specified immediately after the subheader is larger than the amount of text data received
 - 1) The E71 waits for the remaining data.
 - 2) When the remaining data is received within the response monitoring timer value, the E71 processes data according to the code in the subheader.
 - 3) If the remaining data could not be received within the response monitoring timer value, the E71 performs the following processes.
 - Sends the ABORT (RST) instruction to the connected device and closes the line.
 - Notifies the CPU module that an open error has occurred by turning on Open abnormal detection signal (X18)
 - Stores the error code in the open error code storage area. (The error code is not stored in the error log storage area.)

(2) No procedure

In the application data for communications in the "No Procedure" control method, the following data code is expressed in a binary code. Communications are performed using a binary code regardless of the communication data code setting.



Point

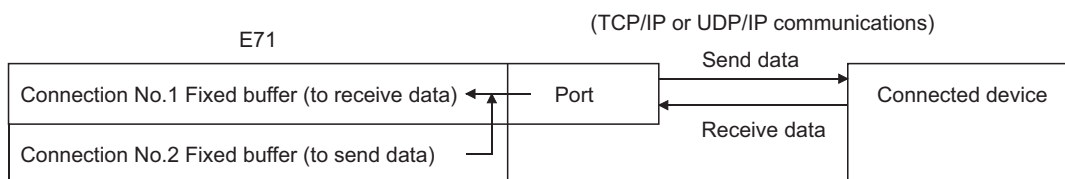
In communications in the "No Procedure" control method, a subheader or data length setting is not added unlike the "Procedure Exist" control method; therefore, all data are handled as a valid text. In addition, the E71 turns on Fixed buffer reception status signal after storing the size of the received message (packet) in the data length storage area. It is recommended to set a checking method. For example, include data length and a data type code in the application data of a message so that the number of bytes and data type of the application data can be identified on the receiving side.

10.7 Pairing Open

Pairing open is an opening method that connects the own station with the connected device using only one port by pairing the receive and send connections for communications using a fixed buffer.

10.7.1 Applications

Enabling the pairing open allows data communications to be performed with two connections by performing the open process for one port. MC protocol communications and communications using a random access buffer can be also performed using pairing-opened connections.

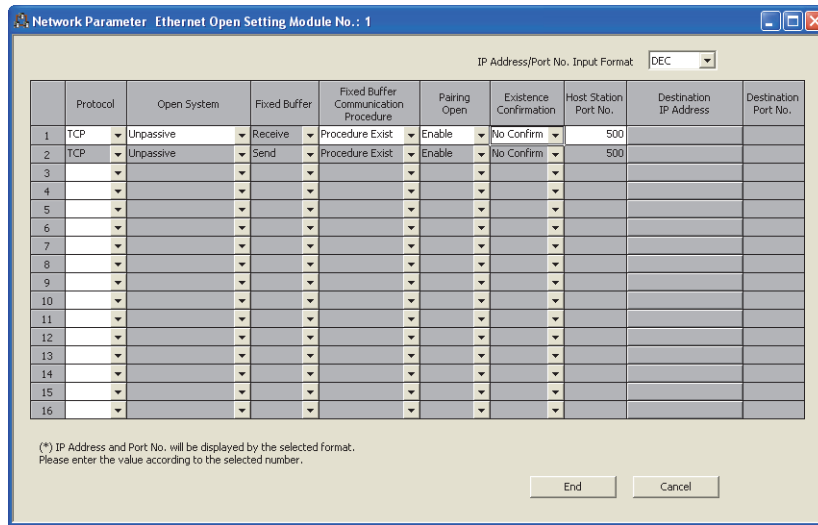


Point

- Connected devices where communications can be performed using the pairing open method are those in Ethernet where an E71 is connected and those connected using the router relay function.
- The open/close processes in the receive connection where the pairing open has been enabled automatically perform the open/close processes in the next connection (send connection).

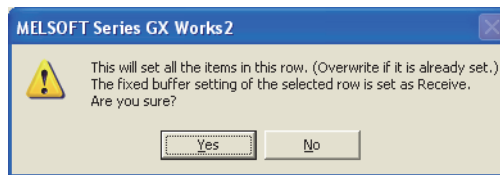
10.7.2 Parameter setting

The following is the pairing open setting.



Item	Description	Setting range
Pairing Open	Set "Pairing Open" of the receive connection to "Enable". The next connection is automatically set as a send connection.*1	-

*1 When "Enable" has been set in "Pairing Open" for a send connection, the following window appears.



When the button is selected, the applicable connection becomes a receive connection, and the next connection is automatically set as a send connection. (When "Enable" has already been set in "Pairing Open", the setting is overwritten.)

Point

Because the fixed buffer of the connection with the connected device (for receiving) and the fixed buffer of the next connection (for sending) are paired, set connection numbers 1 to 7 and 9 to 15 as receive connections. (Connection numbers 8 and 16 cannot be set.)

10.8 Broadcast Communications

Broadcast communications mean that the same data is sent to all E71-connected stations in the same Ethernet network and to the connected devices without specifying destinations. Broadcast communications can be performed only when the UDP/IP protocol is used and during "No Procedure" communications using a fixed buffer.

Point/

- When connected devices in the same Ethernet network do not need to receive communication messages by broadcast communications, a discarding process is required.
 - The user needs to determine the port numbers dedicated for data sending/receiving for broadcast communications.
-

10.8.1 Sending/receiving procedures

(1) Sending procedure

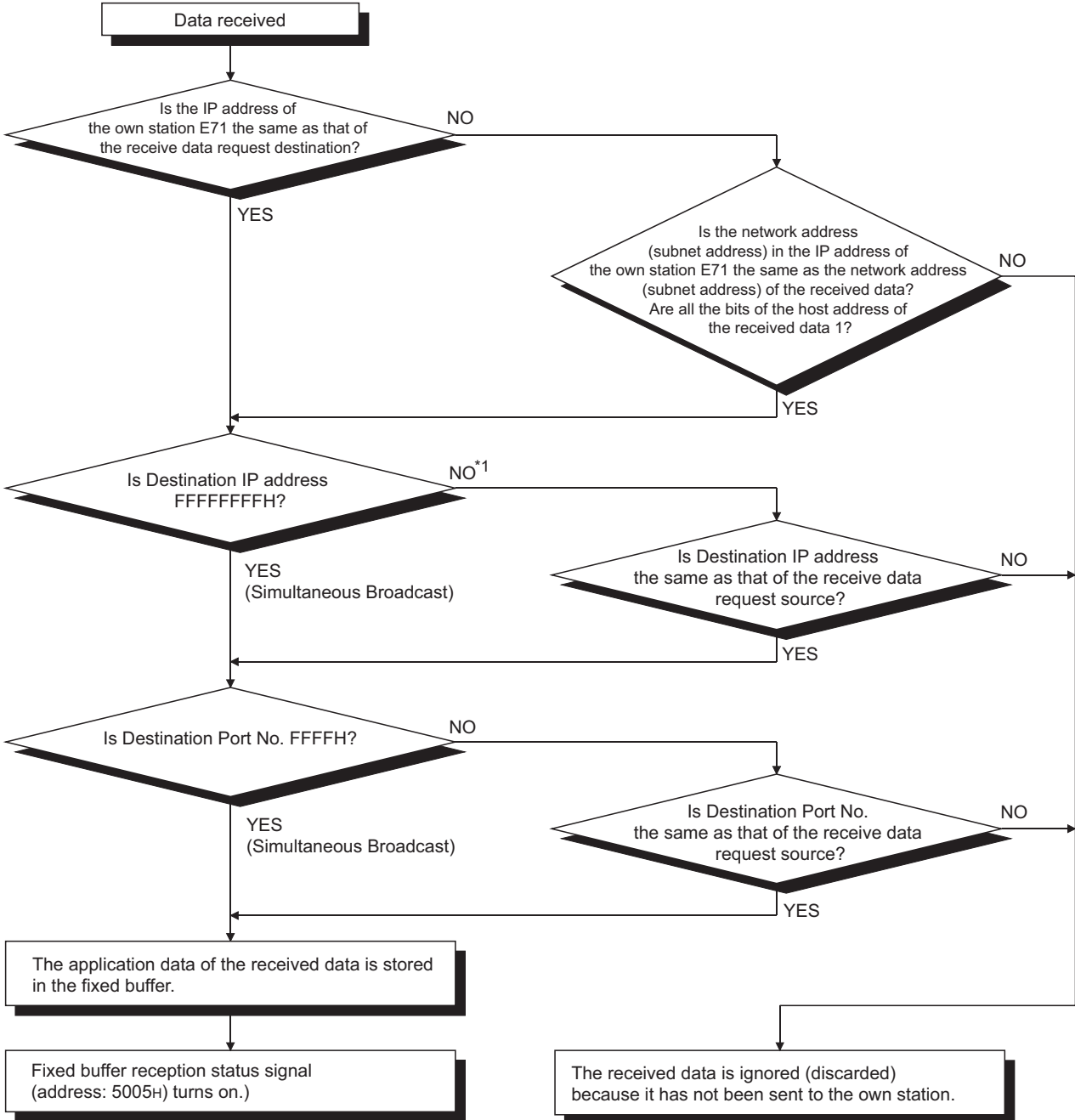
To send data by broadcast communications, set the IP addresses of the connected devices where the E71 sends data to FFFFFFFF_H and perform the open process. The E71 sets all the host address bits to 1 and sends data over the Ethernet network.

(2) Receiving procedure

To receive data as data for broadcast communications, set the IP addresses of the connected devices that send data to the E71 to FFFFFFFF_H and the port numbers to FFFF_H and perform the open process.

Remark

The following is the outline of the internal process of an E71 when data is received in the "No Procedure" control method or by broadcast communications. (The values specified in the buffer memory addresses 0_H, 1_H and 28_H to 5F_H are used for the IP addresses and port numbers of the E71 in the own station as well as those of connected devices.)



*1 When all the bits of the area expressing the host address of the IP address of the receive data request destination are 1, proceed to "YES".

10.8.2 Parameter setting

The following is the parameter setting for broadcast communications.

(1) When sending data

The following is the parameter setting for data sending.

	Protocol	Open System	Fixed Buffer	Fixed Buffer Communication Procedure	Pairing Open	Existence Confirmation	Host Station Port No.	Destination IP Address	Destination Port No.
1	UDP		Send	No Procedure	Disable	No Confirm	0800	Broadcast Together	0801
2									
3									
4									
5									
6									
7									
8									
9									
10									
11									
12									
13									
14									
15									
16									

(*) IP Address and Port No. will be displayed by the selected format.
Please enter the value according to the selected number.

End Cancel

Item	Description	Setting range
Protocol	Select "UDP".	-
Fixed Buffer	Select "Send".	-
Fixed Buffer Communication Procedure	Select "No Procedure".	-
Existence Confirmation	Select "No Confirm".	-
Destination IP Address	Set FFFFFFFFH.	-

(2) When receiving data

The following is the parameter setting for data receiving.

	Protocol	Open System	Fixed Buffer	Fixed Buffer Communication Procedure	Pairing Open	Existence Confirmation	Host Station Port No.	Destination IP Address	Destination Port No.
1	UDP		Receive	No Procedure	Disable	No Confirm	0801	Broadcast Together	FFFF
2									
3									
4									
5									
6									
7									
8									
9									
10									
11									
12									
13									
14									
15									
16									

(*) IP Address and Port No. will be displayed by the selected format.
Please enter the value according to the selected number.

End Cancel

Item	Description	Setting range
Protocol	Select "UDP".	-
Fixed Buffer	Select "Receive".	-
Fixed Buffer Communication Procedure	Select "No Procedure".	-
Existence Confirmation	Select "No Confirm".	-
Destination IP Address	Set FFFFFFFFH.	-
Destination Port No.	Set FFFFH.	-

10.8.3 Precautions

This section describes the precautions for broadcast communications.

(1) Port number

To perform broadcast communications, the user needs to determine the port numbers dedicated for data sending/receiving for broadcast communications and specify these port numbers.

(2) Size of send/receive data per sending/receiving

A maximum of 2046 bytes of data in the application data can be processed per sending/receiving. To send data with a size exceeding 2047 bytes, divide the data into smaller chunks at the send source.

Remark

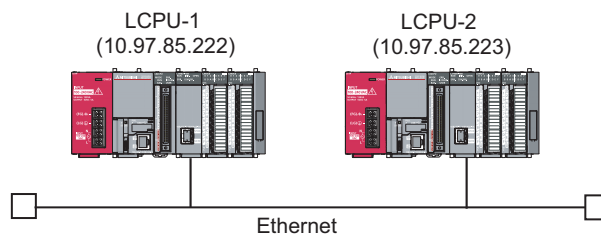
An E71 temporarily stores a subsequent receive data in its internal buffer for the operating system until the current receive process is completed. If data with a size exceeding the capacity of the internal buffer (approximately 40KB) is received by broadcast communications, the data is discarded. In communications using a fixed buffer in the "Procedure Exist" control method, the E71 sends a command message to the connected device, waits for a response message, then sends the next command message. Therefore, the user need not take account of the above-mentioned internal buffer for the operating system.

10.9 Example of Communications Using a Fixed Buffer

This section describes an example of communications using a fixed buffer in the "Procedure Exist" control method between an E71 and the connected device.

10.9.1 System configuration

The following system configuration is used for explanation purpose.



10.9.2 Parameter setting

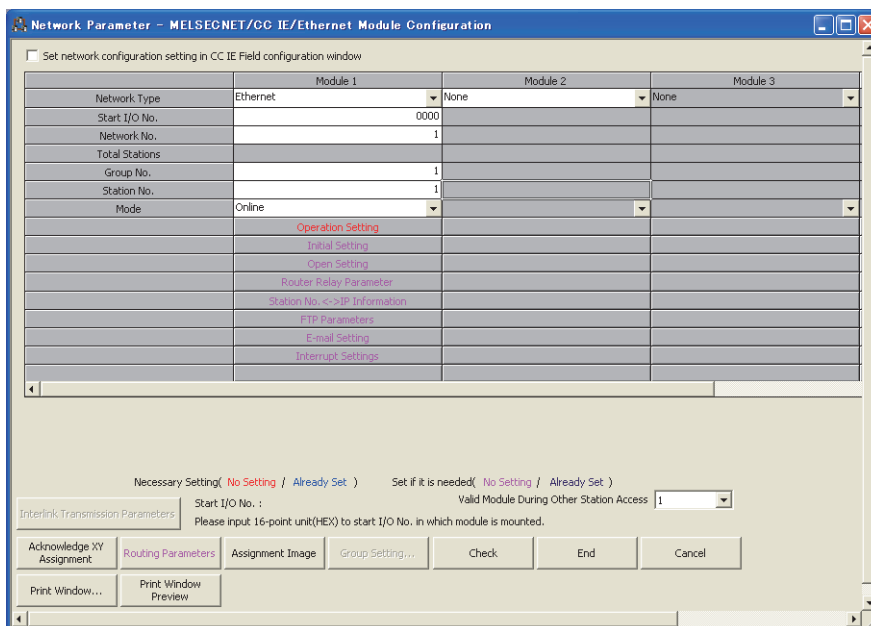
This section describes a parameter setting example.

(1) Sending side (LCPU-1 side)

The following is an example of the parameter settings of the sending side (LCPU-1 side).

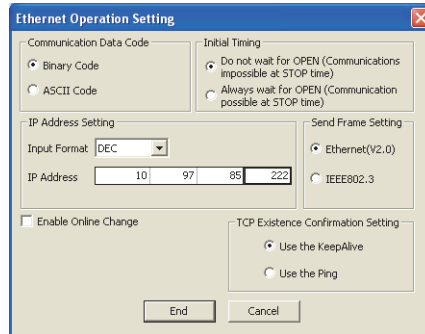
(a) Basic setting

The following is an example of the basic setting.



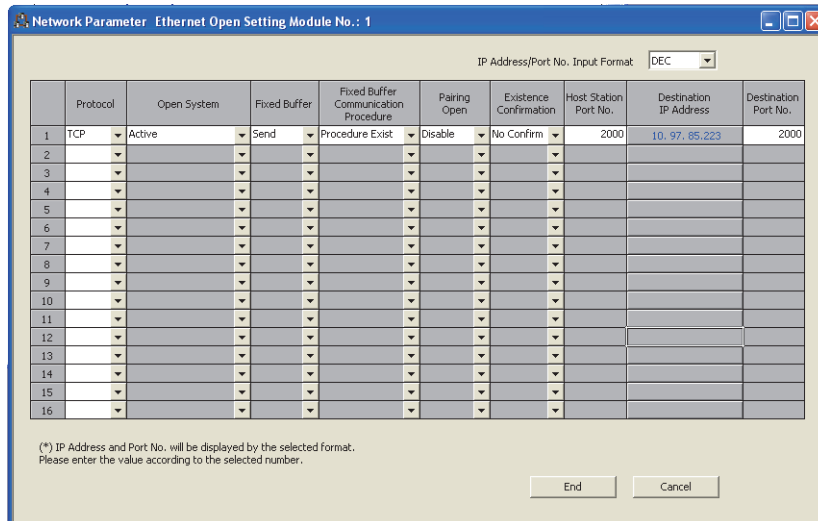
(b) Ethernet operation setting

The following is an example of the Ethernet operation setting.



(c) Open setting

The following is an example of the open setting.

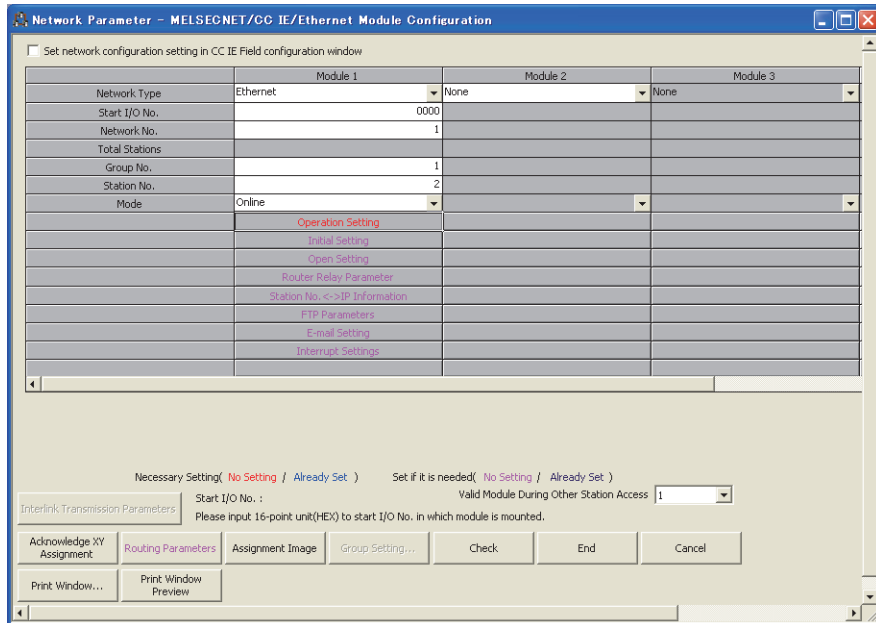


(2) Receiving side (LCPU-2 side)

The following is an example of the parameter settings of the receiving side (LCPU-2 side).

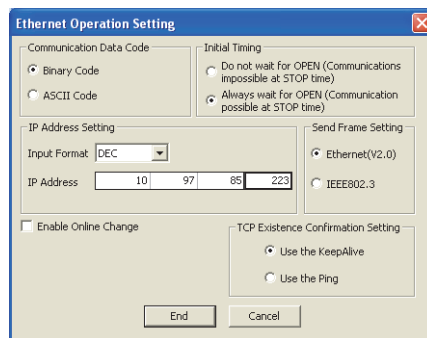
(a) Basic setting

The following is an example of the basic setting.



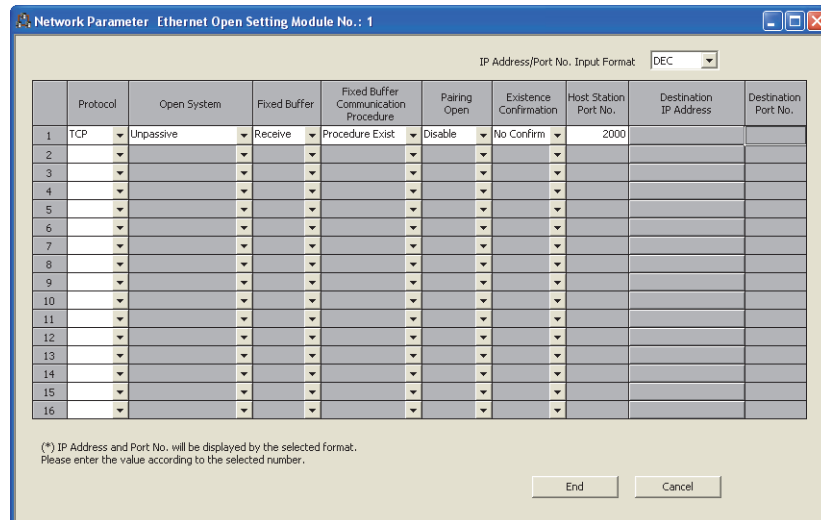
(b) Ethernet operation setting

The following is an example of the Ethernet operation setting.



(c) Open setting

The following is an example of the open setting.



10.9.3 Program

(1) Sample program procedures

This section describes the procedures for a sample program.

(a) Sending side (LCPU-1 side)

The following describes how communications are processed on the sending side (LCPU-1 side).

- 1. Set each parameter on a programming tool and write the set parameters to the CPU module. Then reset the CPU module and check that the initial process is completed.**
- 2. Perform the open process (Active open) of the connection number 1.**
- 3. Perform communications using a fixed buffer from the CPU module using the BUFSND instruction.**
- 4. After the data is sent, perform the close process of the connection number 1.**

(b) Receiving side (LCPU-2 side)

The following describes how communications are processed on the receiving side (LCPU-2 side).

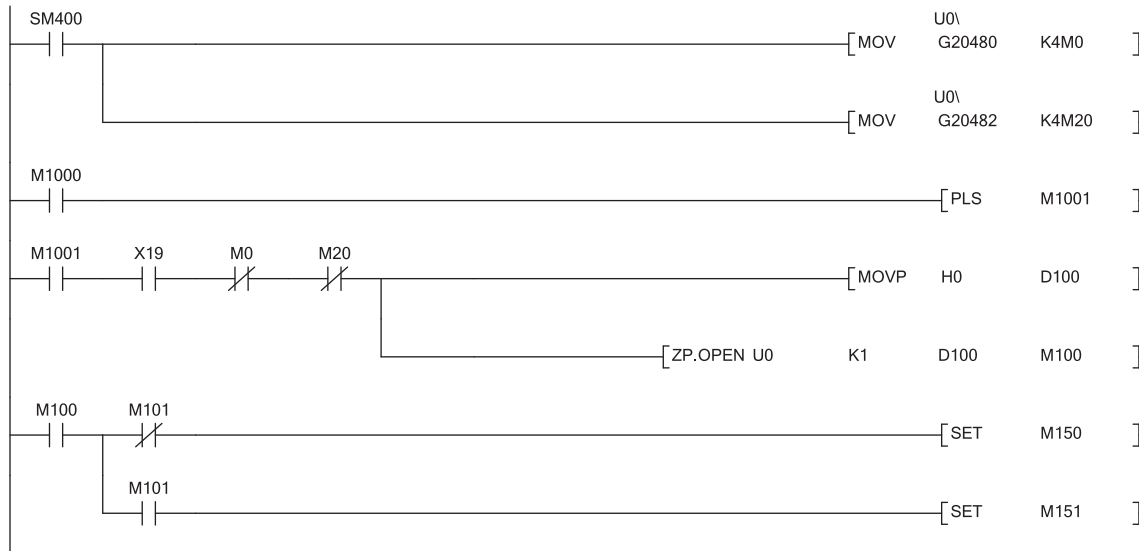
- 1. Set each parameter on a programming tool and write the set parameters to the CPU module. Then reset the CPU module and check that the initial process is completed. When the initial process is normally completed, the connection number 1 waits for an Active open request from the connected device.**
- 2. Perform communications using a fixed buffer from the connected device.**
- 3. Use the BUFRVC instruction to read the data that have been received by the corresponding fixed buffer data area of the E71 into the CPU module.**

(2) Sample program

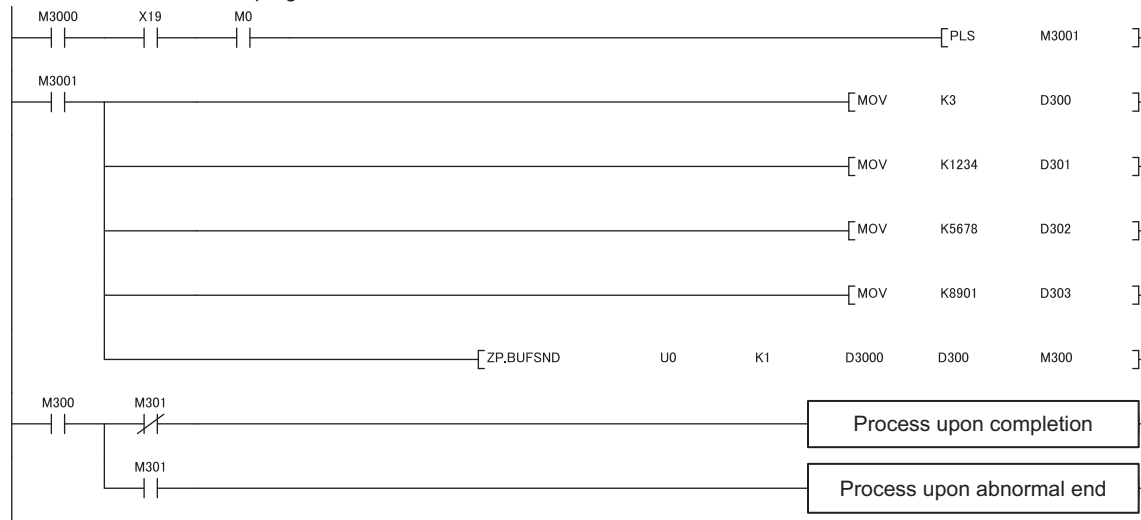
(a) Program on the sending side (LCP-1 side)

The following is a program on the sending side (LCP-1 side).

<<Open processing program>>



<<Fixed buffer No.1 send program>>

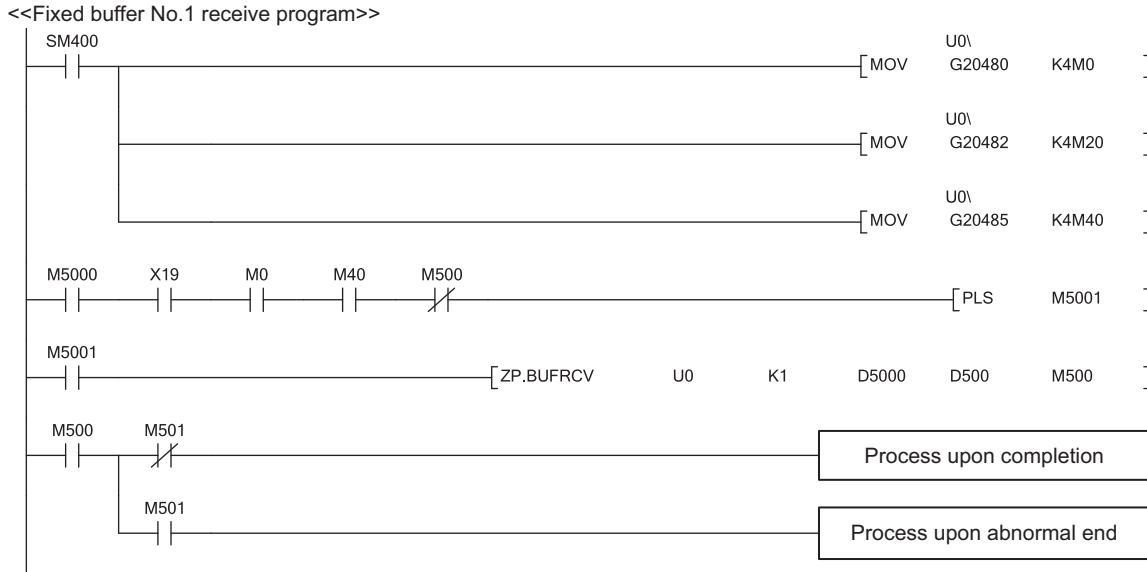


<<Close processing program>>



(b) Receiving side (LCPU-2 side)

The following is a program on the receiving side (LCPU-2 side).



Point

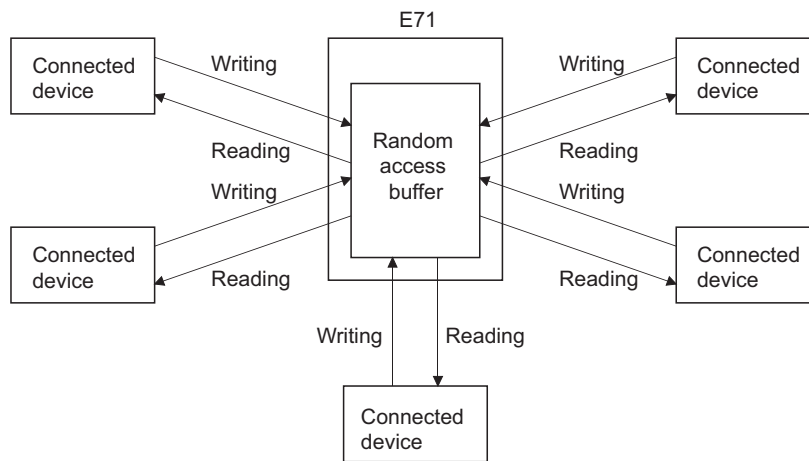
Secure sufficient device areas according to the maximum length of data sent from the send source to prevent the device areas used for other purposes from being overwritten by the receive data.

CHAPTER 11 COMMUNICATIONS USING A RANDOM ACCESS BUFFER

This section describes communications using a random access buffer.

11.1 Applications

In communications using a random access buffer, data can be freely read from and written to any connected device (not including an E71). A random access buffer is used as a common buffer area for all the connected devices in the Ethernet network.



Point

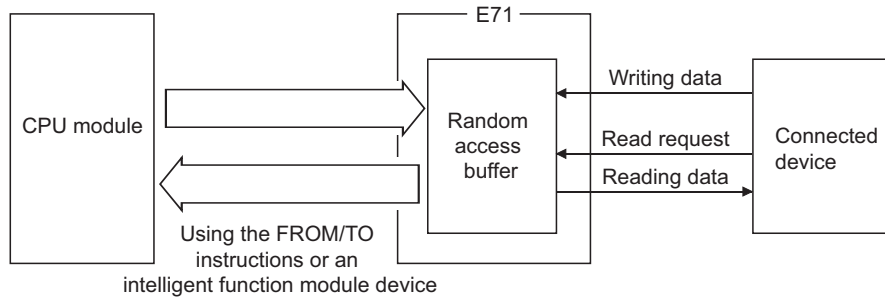
Data communications (sending/receiving) by e-mail using a CPU module cannot be used together with communications using a random access buffer. Only one of the functions can be used at one time. (The e-mail sending function using the E71 programmable controller CPU monitoring function can be used together with communications using a random access buffer.)

11.2 Communication Structure

This section describes the structure of communications using a random access buffer.

(1) Data flow

The data flow in communications using a random access buffer is as follows. The FROM/TO instructions or intelligent function module devices are used to access a random access buffer from a CPU module.



Point

- Communications can be performed only with connected devices where communications using a fixed buffer in the "Procedure Exist" control method are set using the connection with Open completion signal of an E71 on.
- Communications are performed asynchronously with the program. If synchronization is required, use communications using a fixed buffer.

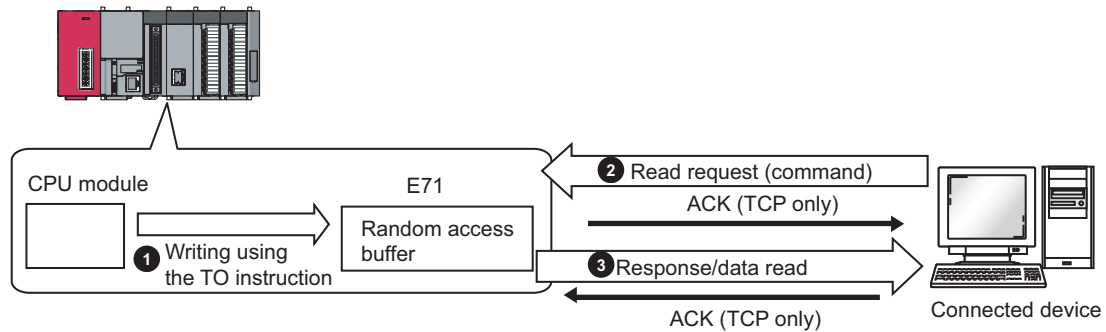
(2) Connected devices where data communications can be performed

Data communications can be performed with the following connected devices:

- Devices in Ethernet where an E71 is connected
- Devices connected using the router relay function

11.2.1 How data is read from a connected device

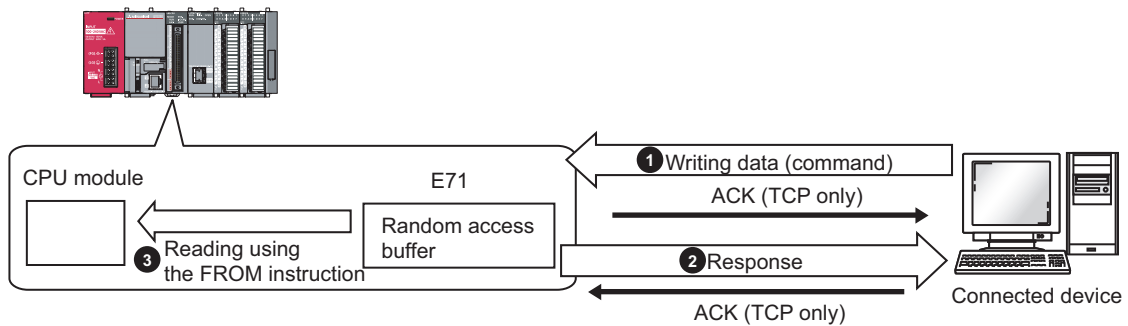
The following figure shows how an E71 sends data in response to a read request sent from a connected device.



- 1 Write data in the random access buffer of the E71 using the TO instruction.
- 2 Asynchronously with the process 1 above, the connected device sends a read request to the E71. (The E71 side: command receiving)
- 3 Upon receiving the read request from the connected device, the E71 sends the data written in the random access buffer to the connected device that sent the read request. (The E71 side: response sending)

11.2.2 How a connected device writes data

The following figure shows how a connected device writes data to the E71 random access buffer.

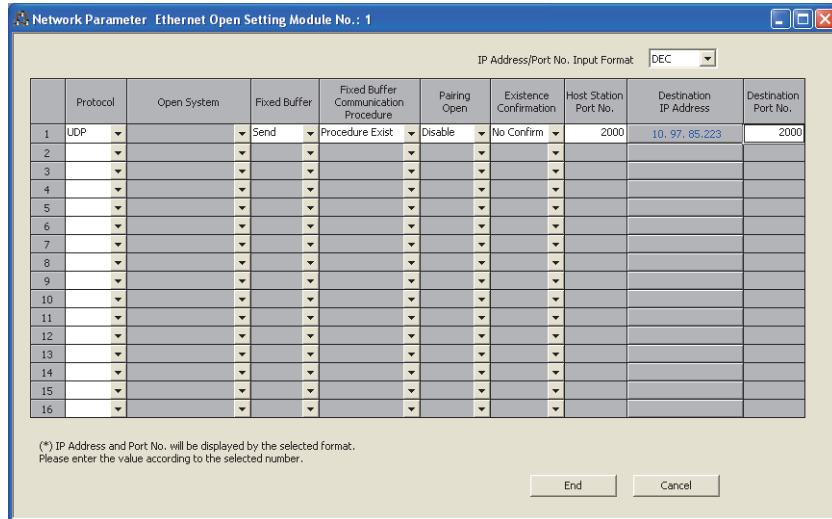


- 1 The connected device writes data to the random access buffer of the E71. (The E71 side: command receiving)
- 2 The E71 processes the write request from the connected device and returns the writing result to the connected device that sent the write request. (The E71 side: response sending)
- 3 Asynchronously with the processes 1 and 2 above, the data written in the random access buffer is read using the FROM instruction.

11.3 Parameter Setting

Set the following parameters to perform communications using a random access buffer.

- Basic setting (☞ Page 64, Section 7.1.2)
- Ethernet operation setting (☞ Page 65, Section 7.1.3)
- Open setting (☞ Page 67, Section 7.1.4)



Item	Description	Setting range
Fixed Buffer Communication Procedure	Select the communication method for communications using a fixed buffer. Select "Procedure Exist" for communications using a random access buffer.	Procedure Exist

11.4 Data Format

Communication data consists of a header and application data.

11.4.1 Header

The header for TCP/IP or UDP/IP is used. Because an E71 automatically adds and deletes a header, the user setting is not required.

(1) Detailed header sizes

(a) TCP/IP

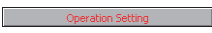
Ethernet (14 bytes)	IP (20 bytes)	TCP (20 bytes)
------------------------	------------------	-------------------

(b) UDP/IP

Ethernet (14 bytes)	IP (20 bytes)	UDP (8 bytes)
------------------------	------------------	------------------

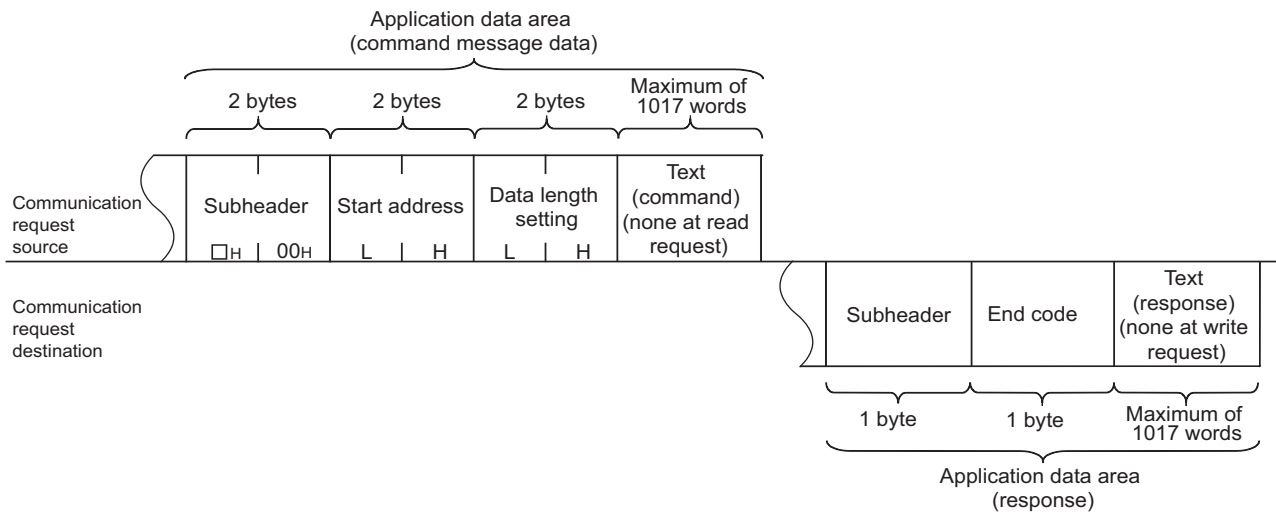
11.4.2 Application data

In the application data for communications with the "Procedure Exist" control method, the following data code is expressed in a binary code or an ASCII code. To switch between a binary code and an ASCII code, use the Ethernet operation setting.

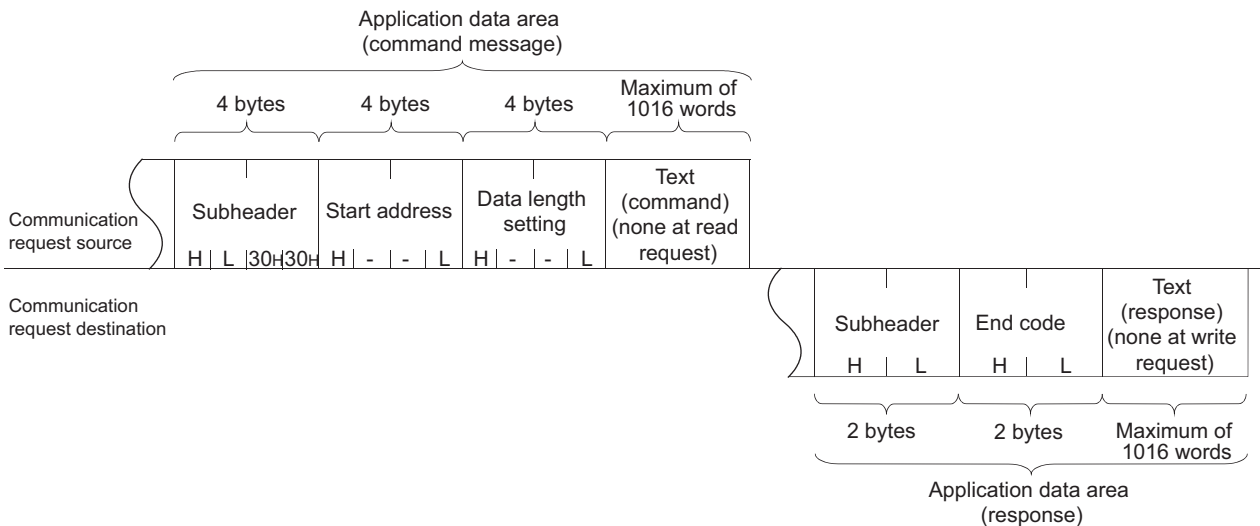
Project window ⇒ [Parameter] ⇒ [Network Parameter] ⇒ [Ethernet/CC IE Field] ⇒ Select "Ethernet" under "Network Type". ⇒  button

(1) Format

- Communications using a binary code

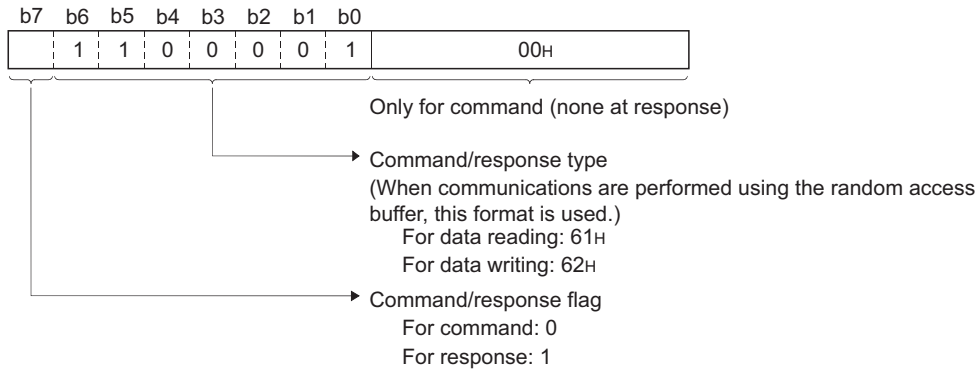


- Communications using an ASCII code

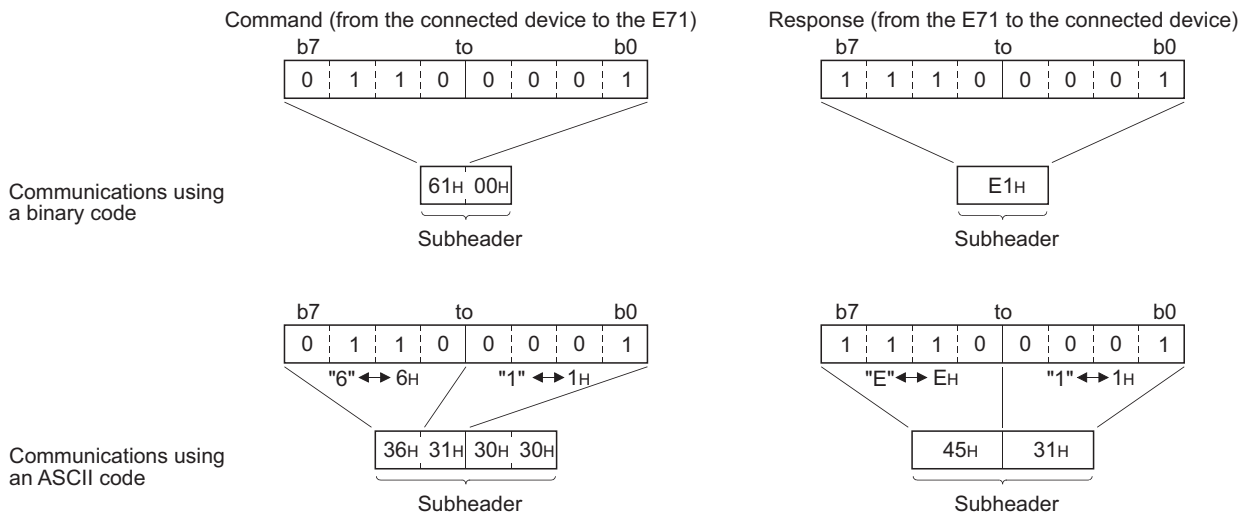


(2) Subheader

Because an E71 automatically adds and deletes a header, the user setting is not required.



(a) When reading

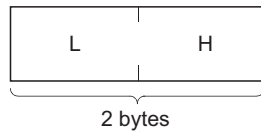


(3) Start address

Use a logical address to set the start address of a random access buffer that reads/writes data. (☞ Page 133, Section 11.6)

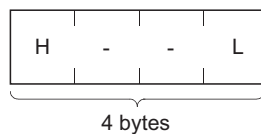
(a) Communications using a binary code

Specify the start address using a binary value.



(b) Communications using an ASCII code

Specify the start address using an ASCII code value expressed in hexadecimal.

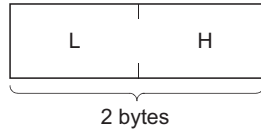


(4) Data length setting

Set the number of words of the read/written data in the random access buffer range.

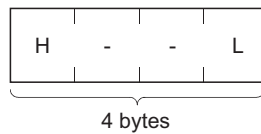
(a) Communications using a binary code

Specify the number of words using a binary value. (Maximum of 1017 words)



(b) Communications using an ASCII code

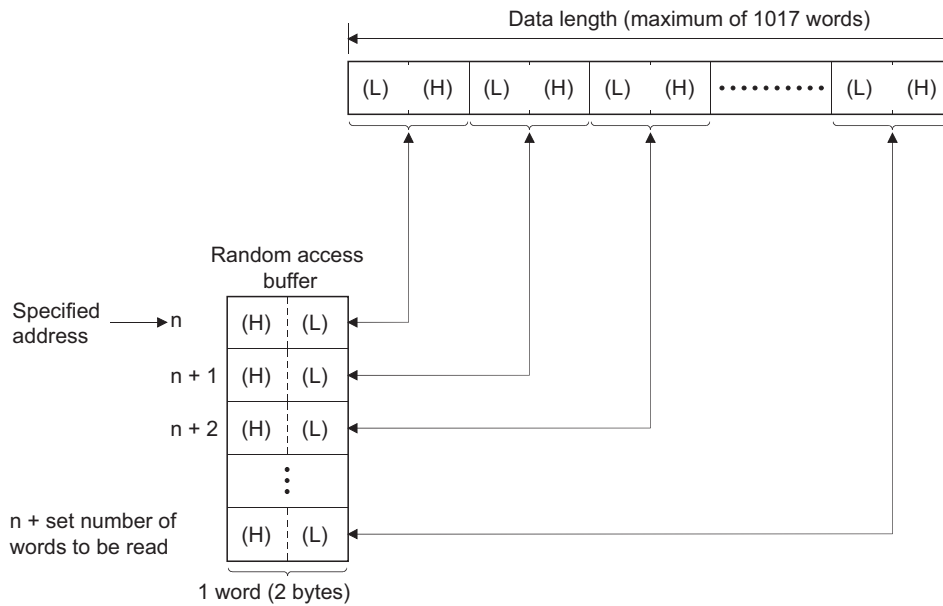
Specify the number of words using an ASCII code value expressed in hexadecimal. (Maximum of 508 words)



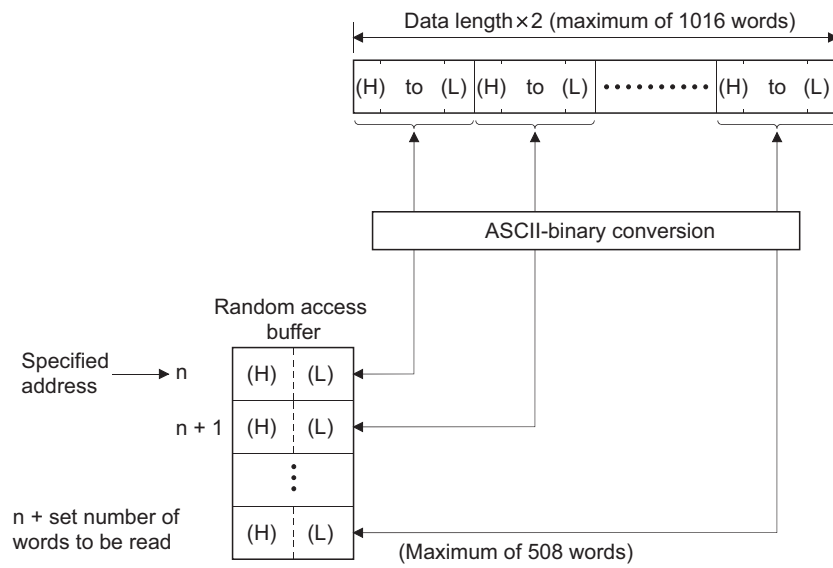
(5) Text

Text is a data written to and read from a random access buffer.

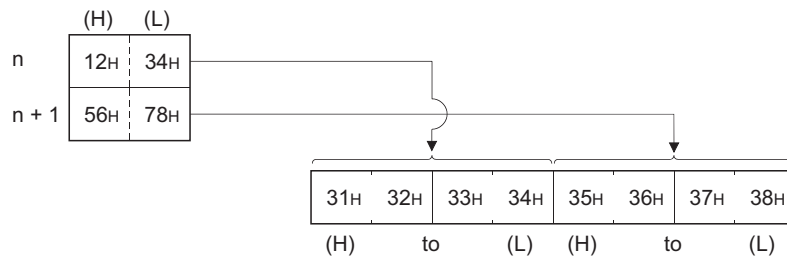
- Communications using a binary code



- Communications using an ASCII code



Ex.



(a) End code

An error code is stored in the end code added to a response. For the error codes, refer to the error code list.

(👉 Page 210, Section 14.7.1)

End codes are stored in the communication status storage area of the buffer memory.

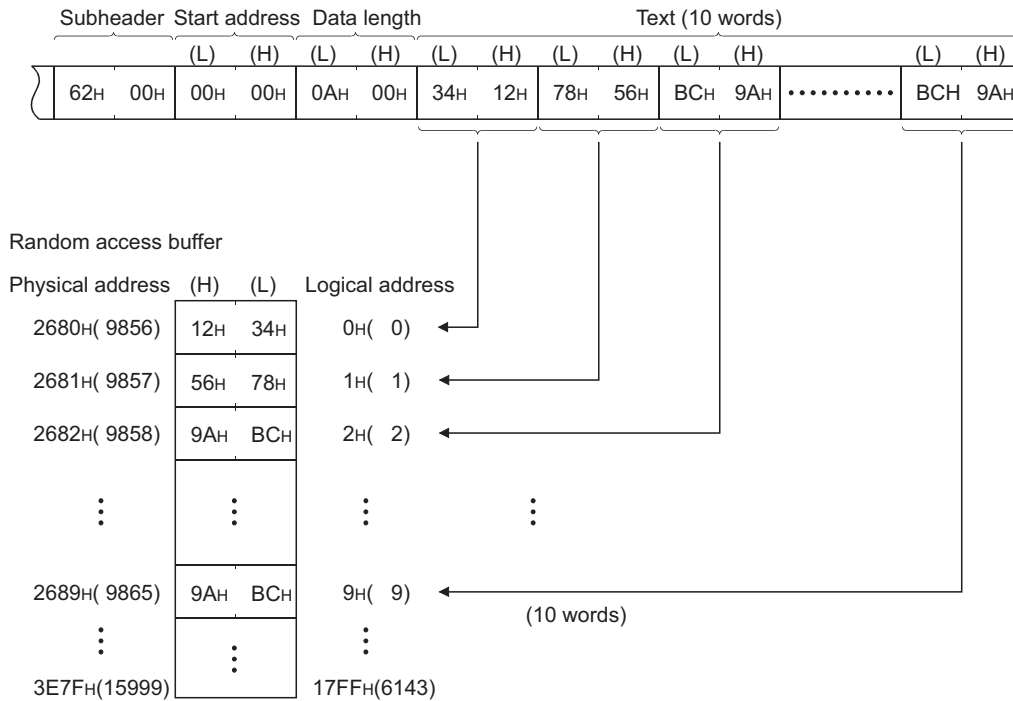
11.4.3 Examples of command and response formats

This section describes examples of command and response formats.

(1) Writing data to a random access buffer upon a write request from a connected device

(a) Communications using a binary code

- Command format (from the connected device to the E71)

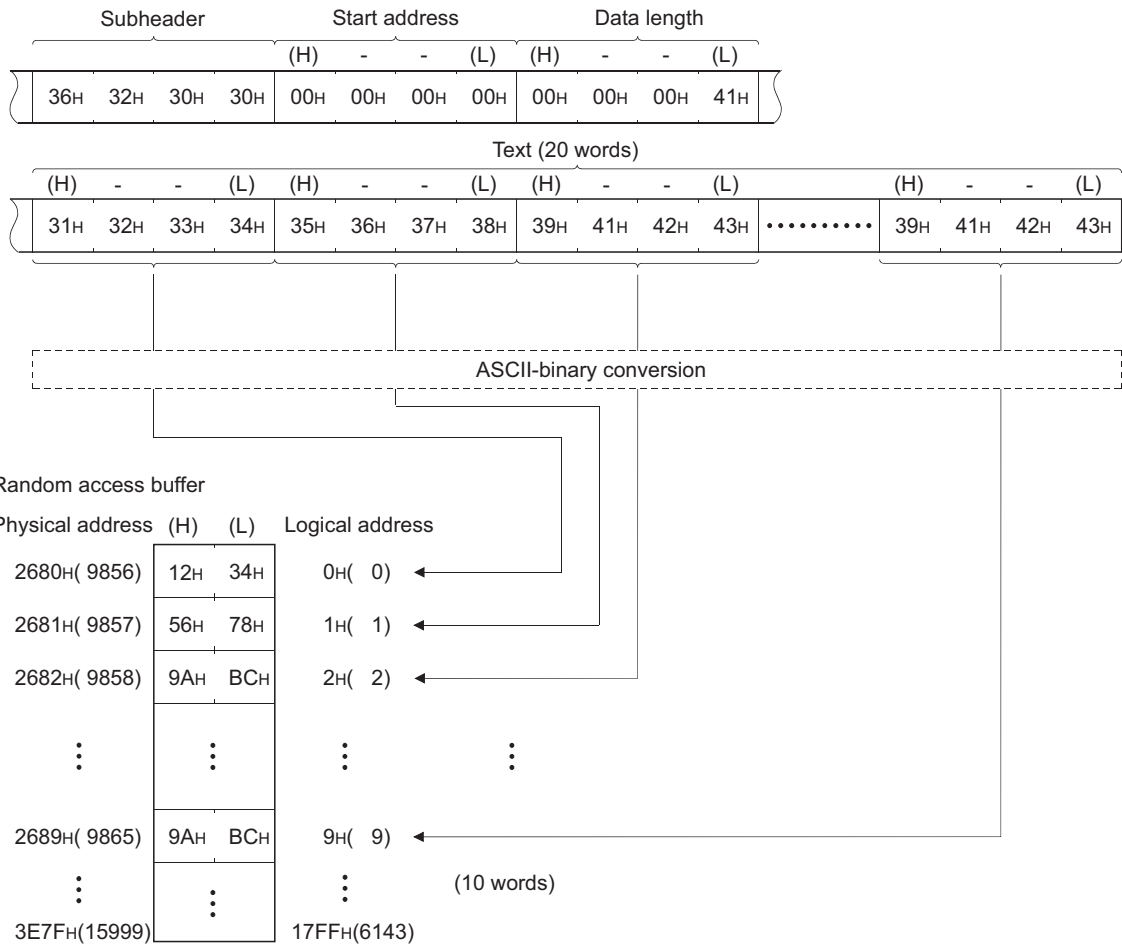


- Response format (from the E71 to the connected device)

Subheader	End code
E2H	00H

(b) Communications using an ASCII code

- Command format (from the connected device to the E71)



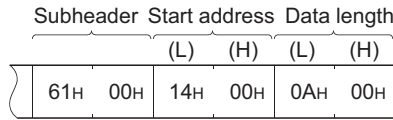
- Response format (from the E71 to the connected device)

Subheader		End code	
45H	32H	30H	30H

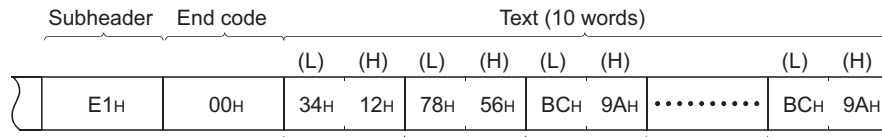
(2) Reading data from a random access buffer upon a read request from a connected device

(a) Communications using a binary code

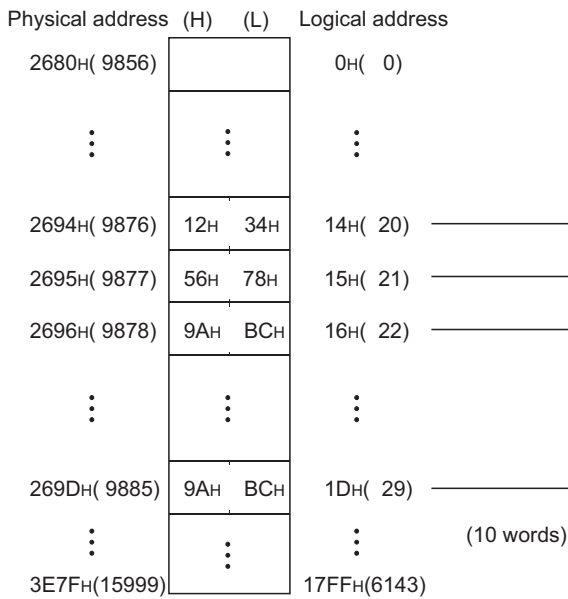
- Command format (from the connected device to the E71)



- Response format (from the E71 to the connected device)



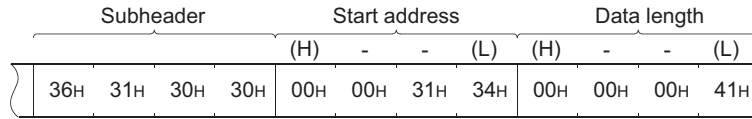
Random access buffer



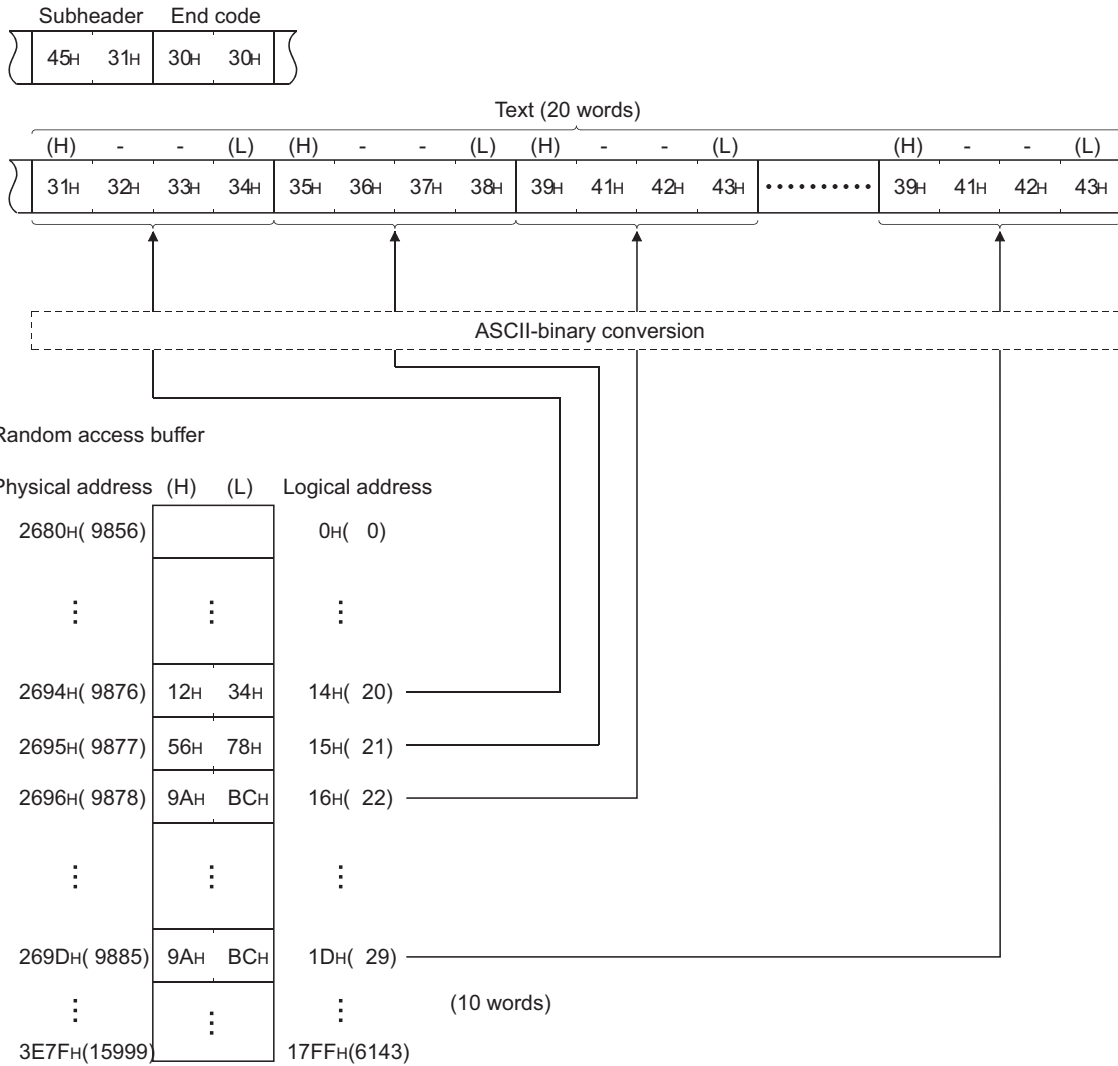
11.4 Data Format
11.4.3 Examples of command and response formats

(b) Communications using an ASCII code

- Command format (from the connected device to the E71)



- Command format (from the connected device to the E71)



11.5 Precautions when Creating Programs

This section describes the precautions when creating programs for communications using a random access buffer.

(1) Completion of the initial and open processes

The initial process and the connection open process need to be completed.

(2) Send request from a CPU module

A CPU module cannot issue send requests. In addition, reception completion in a CPU module is not confirmed. When it is necessary to synchronize data sending/receiving between the CPU module and the connected device, use communications using a fixed buffer.

(3) Random access buffer address

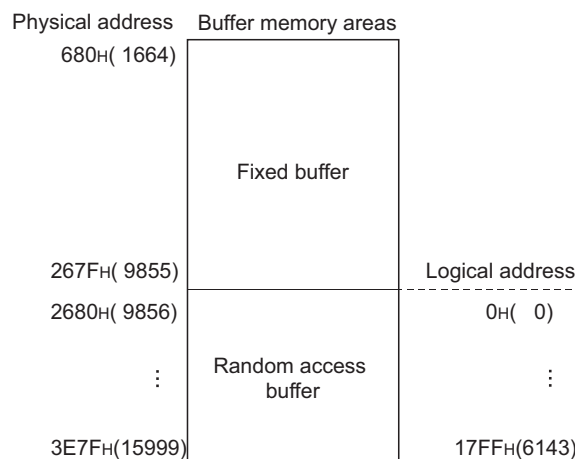
The address specified by the connected device and the address specified using the FROM/TO instructions are different. For details, refer to the physical and logical addresses of a random access buffer. (Page 133, Section 11.6)

11.6 Physical and Logical Addresses of a Random Access Buffer

This section describes the start address of the E71 random access buffer (without battery backup), which is specified in commands.

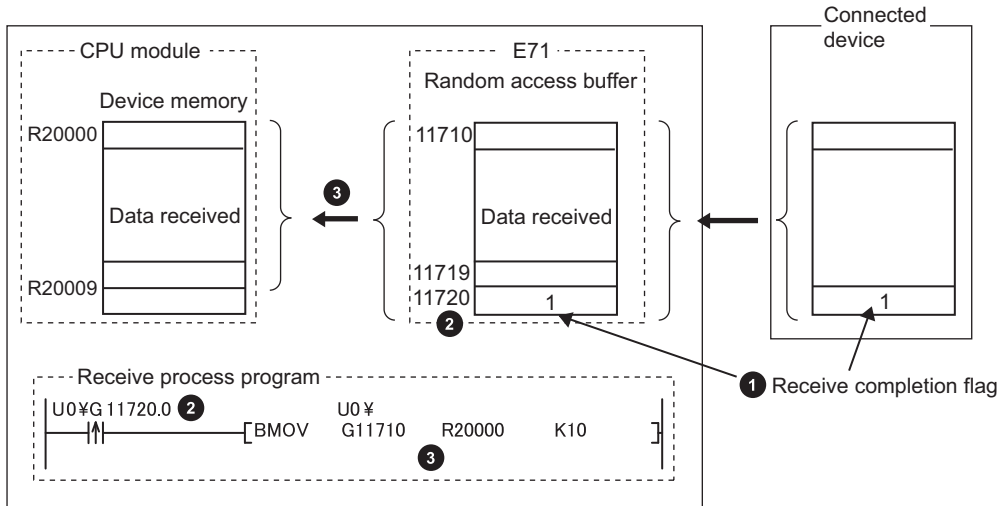
Note that when designating the start address of a random access buffer, the address specified by a connected device is different from that specified using the FROM/TO instructions.

- Physical address: Address specified with the FROM/TO instructions of a program
- Logical address: Address specified by a connected device as the start address in the command



11.7 Example of Communications Using a Random Access Buffer

The following figure shows an example of a write operation from a connected device.



- ① The receive completion flag is provided at the end of the received data.
- ② The receive completion flag is monitored.
- ③ When the receive completion flag turns on, the data is transmitted from the random access buffer to the CPU module.

CHAPTER 12 OTHER FUNCTIONS

This chapter describes basic functions of the E71 other than those described in the preceding chapters.

12.1 Router Relay Function

This section describes the router relay function.

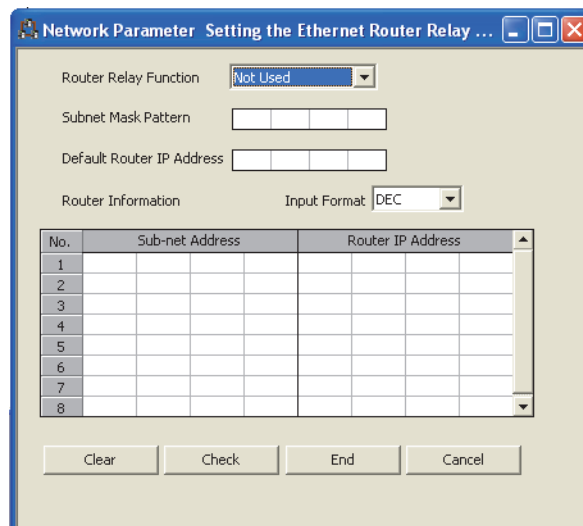
12.1.1 Applications

This function allows the E71 to communicate with connected devices on other Ethernet networks via a router and gateway. This function does not allow the E71 to operate as a router. One default router and up to eight routers can be set.

12.1.2 Parameter settings

Configure the settings in the following window.

Project window ⇒ [Parameter] ⇒ [Network Parameter] ⇒ [Ethernet/CC IE Field] ⇒ Select "Ethernet" under "Network Type". ⇒  button



Router Relay Function:

Subnet Mask Pattern:

Default Router IP Address:

Router Information: Input Format:

No.	Sub-net Address	Router IP Address
1		
2		
3		
4		
5		
6		
7		
8		

Buttons: Clear, Check, End, Cancel

Item	Description	Setting range
Router Relay Function	Select whether to use the router relay function.	Not Used/Use
Subnet Mask Pattern	Refer to (1) in this section.	C0000000 _H to FFFFFFFC _H
Default Router IP Address	Refer to (2) in this section.	A value other than 00000000 _H and FFFFFFFF _H
Input Format	Select the input format of router information.	DEC/HEX
Router Information	Sub-net Address	Refer to (3) (a) in this section. A value other than 00000000 _H and FFFFFFFF _H
	Router IP Address	Refer to (3) (b) in this section. A value other than 00000000 _H and FFFFFFFF _H

(1) Subnet mask pattern

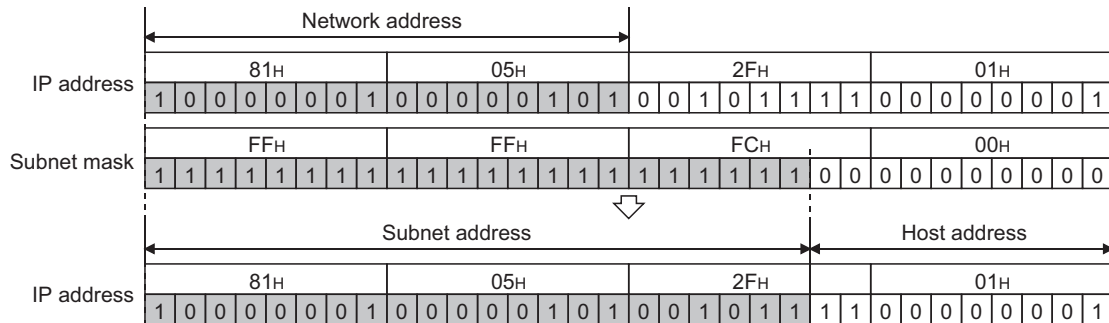
Set the subnet mask. (Consult with the network administrator for the setting.) When not using the subnet mask, set any of the following values according to the class.

Class	Mask value
Class A	FF000000 _H
Class B	FFFF0000 _H
Class C	FFFFFF00 _H

(a) Setting example

The following shows a Class B setting example.

Ex. Class B



Point

- All devices on the same subnetwork must have the common subnet mask.
- When not managed by the subnetwork, the connected devices need not have subnet masks. (Set the network address of the corresponding class.)

(2) Default router IP address

Set the IP address of the router (default router) when the E71 communicates with the connected devices on other Ethernet networks via a router other than the one specified in the router information. Set the value that satisfies the following conditions.

- Condition 1: The IP address class is any of A, B, and C.
- Condition 2: The subnet address of the default router is the same as that of the E71 on the own station.
- Condition 3: The host address bits are not all "0" or all "1".

Point

If the corresponding subnet address does not exist in the router information when the connection is opened or data communications are performed, communications are performed via the default router.

(3) Router information

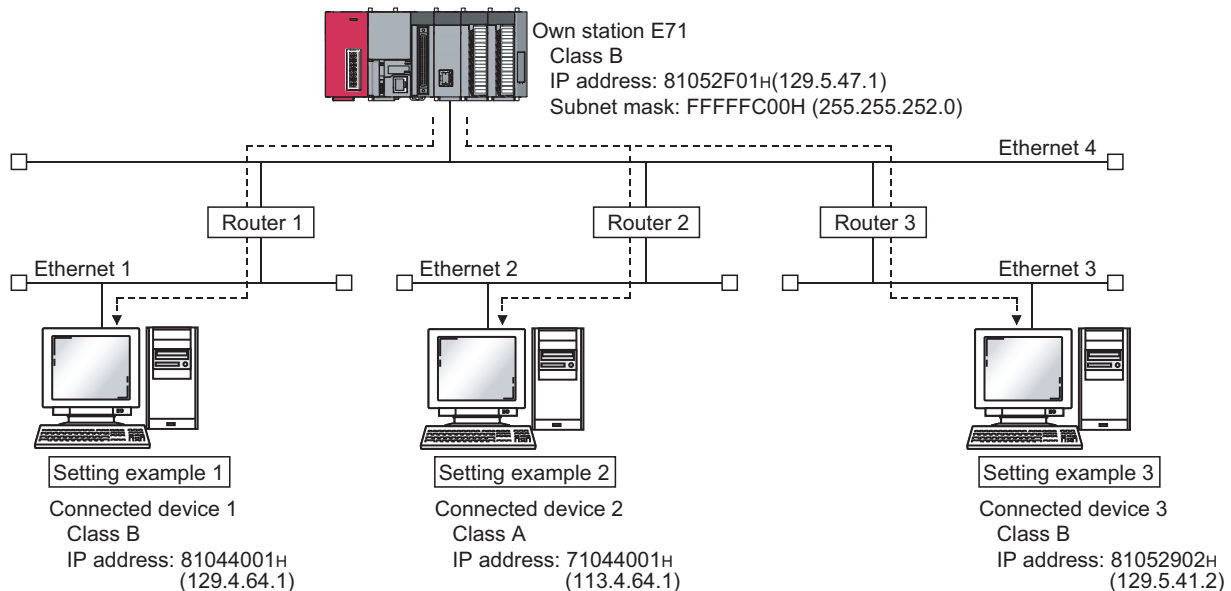
(a) Subnet address

Set the network address^{*1} or subnet address^{*2} of the connected device when the E71 communicates with the connected devices on other Ethernet networks via a router other than the default router. Set the value that satisfies the following conditions.

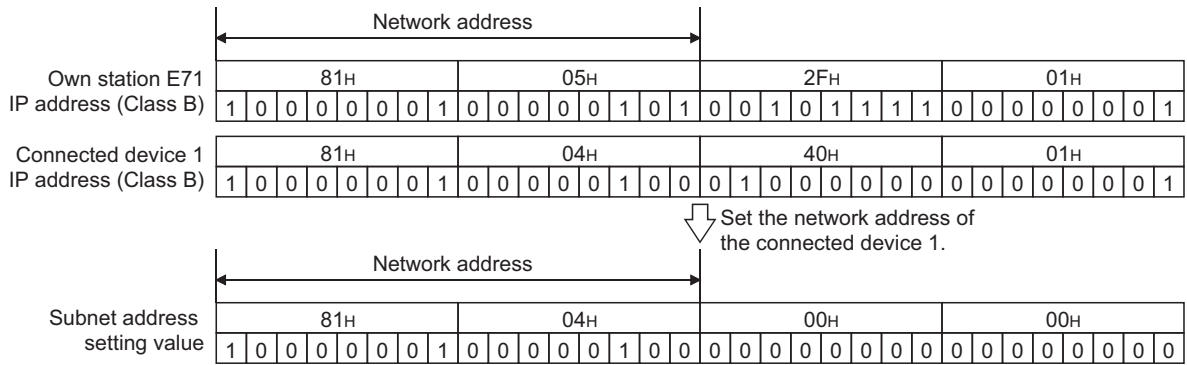
- Condition 1: The IP address class is any of A, B, and C.
- Condition 2: The host address bits are all "0".

*1 If the class (network address) of the E71 on the own station differs from that of the connected device, set the network address of the connected device.

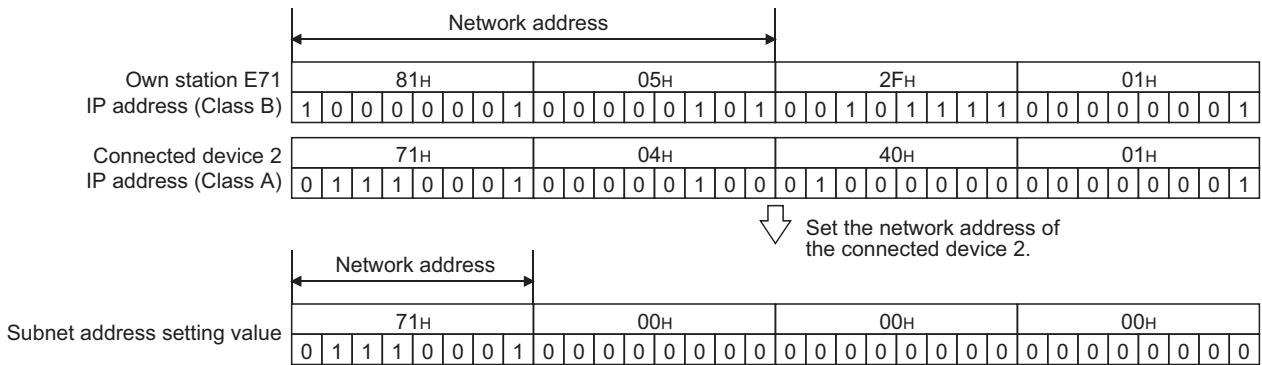
*2 If the class (network address) of the E71 on the own station is the same as that of the connected device, set the subnet address of the connected device.



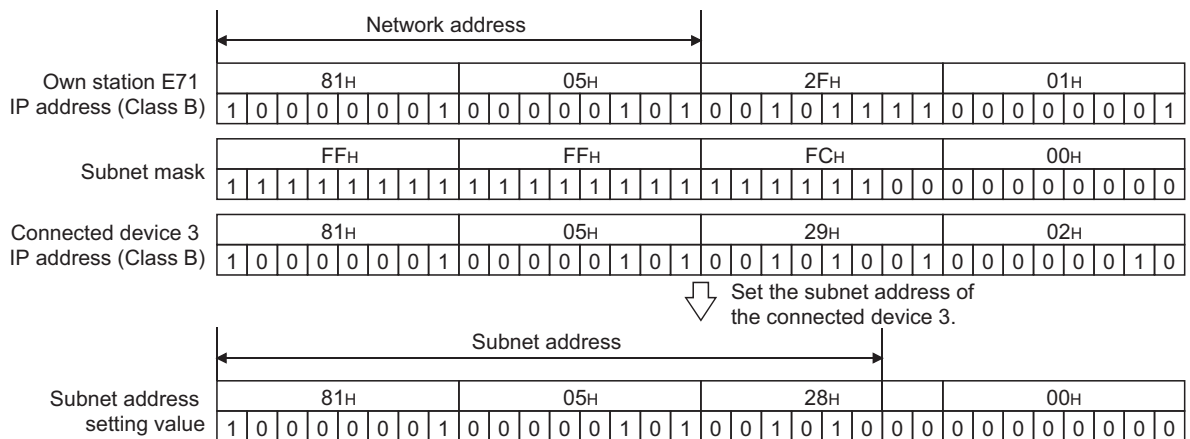
Ex. When the network addresses differ between the E71 on the own station and the connected device



Ex. When the classes differ between the E71 on the own station and the connected device



Ex. When the network address of the E71 on the own station is the same as that of the connected device



(b) Router IP address

Set the IP addresses of the routers when the E71 communicates with the connected devices on other Ethernet networks via a router other than the default router. Set the value that satisfies the following conditions.

- Condition 1: The IP address class is any of A, B, and C.
- Condition 2: The subnet address of the router is the same as that of the E71 on the own station.
- Condition 3: The host address bits are not all "0" or all "1".

Point

- When the E71 communicates with the connected devices via a router in Passive open status, communications can be performed without using the router relay function.
- The router relay function is not needed in a system that uses the Proxy router.

12.2 Communications Using an Auto-open UDP Port

This section describes communications using the auto-open UDP port.

12.2.1 Application

The auto-open UDP port is a UDP/IP port that automatically opens and closes at the following timing. Using this port makes the E71 ready for communications upon completion of an initial process, thus enabling communications without using any programs regardless of the open status of connection numbers 1 to 16.

(1) Open and close timing

The port automatically opens according to the registered parameter settings upon completion of the initial process of the E71. In addition, it automatically closes when the E71-connected station is powered off or reset.

Point

- The E71 enables communications using an auto-open UDP port after the initial process is normally completed, and the E71 on the own station waits for communication requests to itself (auto open).
- The E71 accepts requests from anywhere as long as they are addressed to the E71 itself.
- When a communication request is accepted from the connected device, the corresponding port number is occupied until the process is completed. Even if the next communication request is accepted during this time, the communication process will be waited.
- To change the auto-open UDP port number, a reinitialization process is required. (Page 250, Appendix 3.2)

12.3 Remote Password

The CPU module can be protected by a remote password.

Point

A remote password is to prevent unauthorized access (e.g. Corruption of programs and data) from connected devices. However, this function does not guarantee prevention of all unauthorized access. To have the programmable controller system fully secured against unauthorized access from the connected devices, take additional measures. Mitsubishi Electric Corporation does not hold any responsibility for any system problems caused by unauthorized access. The following are examples of measures against unauthorized access.

- Install a firewall.
 - Install a personal computer as a relay station and control the relay of communication data using an application program.
 - Install the connected device that can control access rights (Please consult with your network service provider or networking equipment manufacturer.) as a relay station.
-

12.3.1 Application

This function allows or prohibits access from connected devices to the CPU module through the following modules. This prevents remote unauthorized access to the CPU module.

- E71
- Built-in Ethernet port LCPU
- C24

12.3.2 Remote password setting processes (unlock and lock processes)

This section describes the processes that enable/disable access from the connected devices to the programmable controller.

(1) Access permission process (unlock process)

To access the specified CPU module, the connected device performs a remote password unlock process for the remote password-protected E71 on the station in direct connection (own station). The following are unlock process methods.

- Dedicated command of the MC protocol (remote password (unlock): 1630)
- When the file transfer (FTP server) function is in use: Dedicated FTP command (password-unlock)
- Programming tool: Input of the remote password in the window
- When the Web function is in use: Input of a remote password in the window displayed on the Web browser

When the unlock process is not performed, the remote password-set E71 that has accepted a communication request checks an entered remote password and disables access to the specified station.

All data received prior to the unlock process is treated as an error.

(2) Access process

Access to the specified station is enabled when the remote password unlock process is completed normally. Specify the station to be accessed and access the station.

(3) Access prohibition process (lock process)

To terminate the access to the specified station, the connected device performs the remote password lock process to disable subsequent access. The following are lock process methods.

- Dedicated command of the MC protocol (remote password (lock): 1631)
- When the file transfer (FTP server) function is in use: Dedicated FTP command (password-lock)
- Programming tool: Automatically executes the lock process.
- When the Web function is in use: Automatically performs a lock process when the Web browser is closed.

12.3.3 Remote password check procedure

This section describes the procedure of the remote password check performed by the E71.

(1) Communications in which an entered remote password is checked

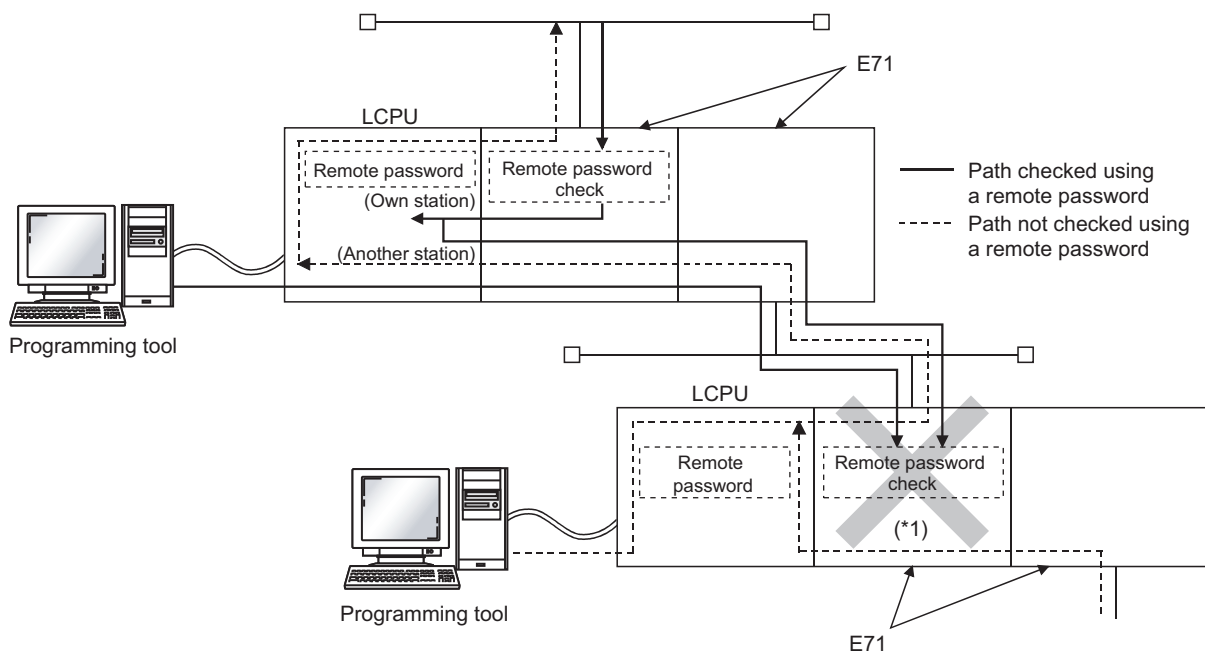
When the following parameters have been set for the E71 on the LCPU station, the E71 checks a remote password for communication requests listed below.

- When a remote password has been set in the CPU module
- When the connection where the E71 communicates with a connected device has been set as the target for the remote password check

The E71 checks an entered remote password for a communication request addressed to the own station/another station received from a connected device.

The E71 performs a send process upon the following send requests without checking an entered remote password.

- Send request from the CPU module on the own station (such as communications using a fixed buffer)
- Communication request from the connected device (including the programming tool connected to the CPU module on the own station) to send data to another station upon request from the CPU module



*1 A communication request from the connected device cannot be accepted because a remote password check has been set. If the remote password check has not been set, the communication request is accepted, and the E71 can communicate with the connected devices.

(2) Selecting a connection for which the remote password check is performed

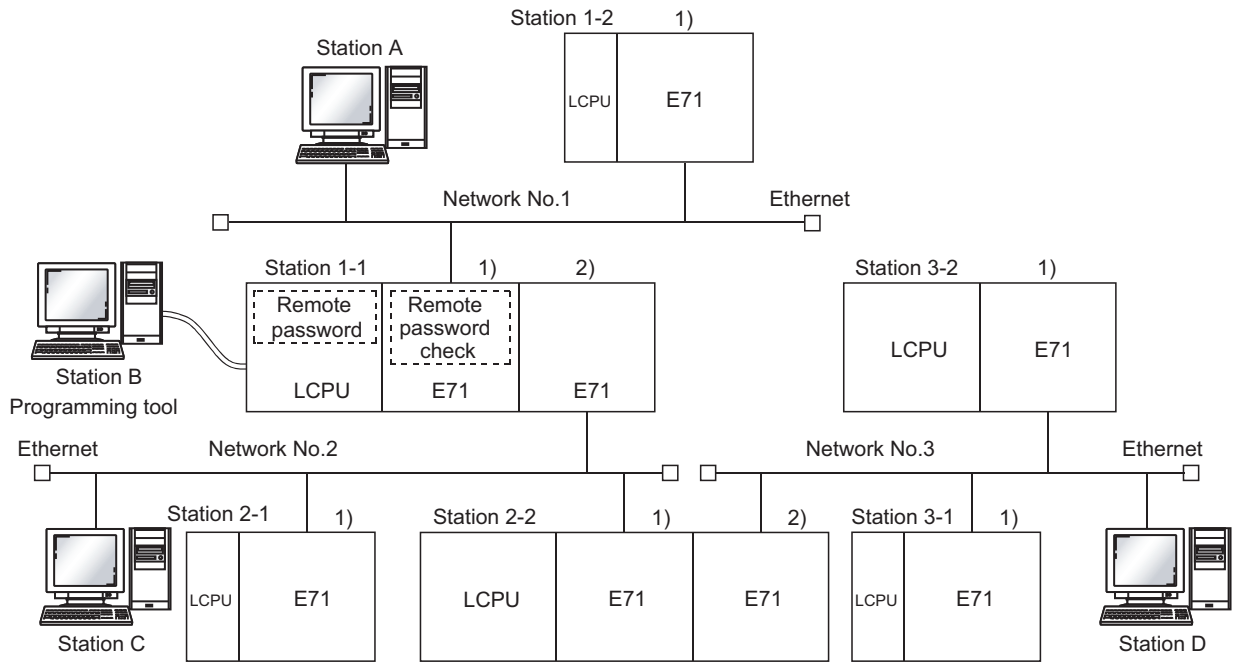
A connection for which an entered remote password is checked can be selected using a parameter.

(☞ Page 148, Section 12.3.6)

(3) Stations that can be accessed when the remote password check is performed

When the CPU module is protected with a remote password, the stations accessible by the connected device and the LCPU stations that can perform the remote password unlock/lock processes are limited to those in the same network. The following shows an example of accessible stations.

Ex. When a remote password has been set in the LCPU on the station 1-1, and the remote password check has been set in 1) on the station 1-1



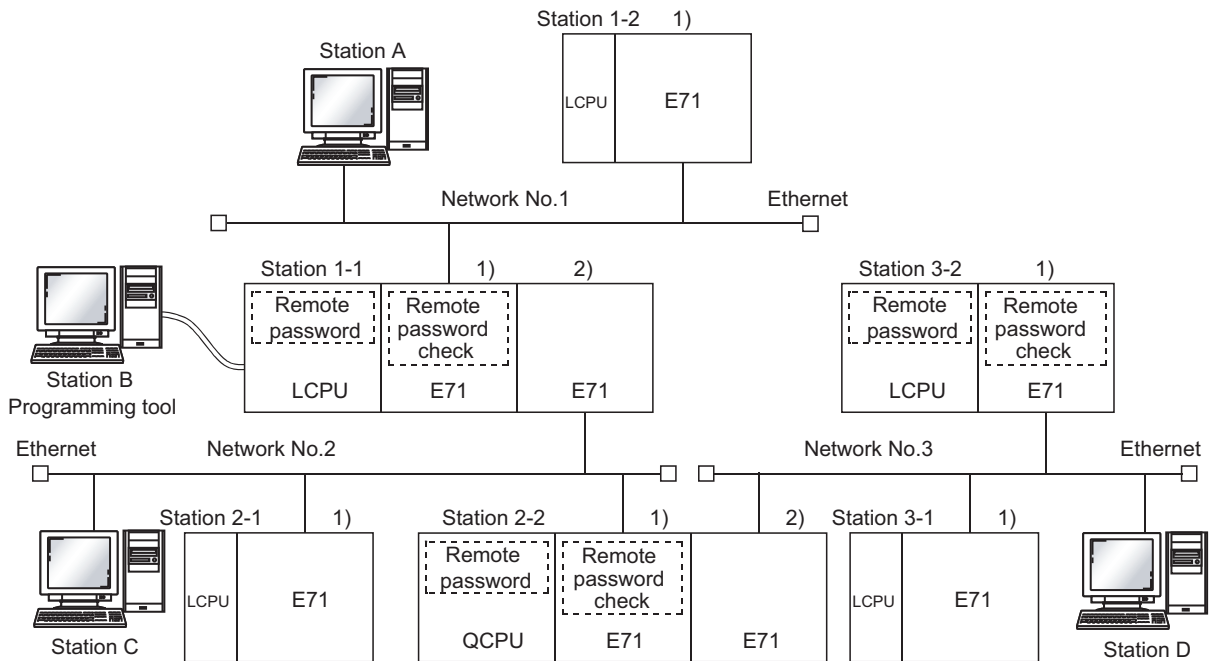
The station A is the only connected device that can perform an unlock/lock process for 1) of the station 1-1.

- : Stations that can be accessed from the connected device after the remote password unlock process
- : Stations that can be accessed from the connected device without performing the remote password unlock process

Connected device ^{*1} (request source)	Target programmable controller station (request destination)					
	Station 1-1 LCPU	Station 1-2 LCPU	Station 2-1 LCPU	Station 2-2 LCPU	Station 3-1 LCPU	Station 3-2 LCPU
Station A	●	○	●	●	●	●
Station B	○	○	○	○	○	○
Station C	○	○	○	○	○	○
Station D	○	○	○	○	○	○

*1 The station A can access the stations marked with ● after the remote password unlock process is completed for 1) of the station 1-1. It can access the stations marked with ○ when the communication lines are open. Stations B, C, and D can access the stations marked with ○ when the communication lines are open.

Ex. When a remote password and the remote password check have been set in multiple LCPU stations



The unlock/lock process can be performed for the following connected devices.

- 1) of the station 1-1: By the station A only
- 1) of the station 2-2: By the station C only
- ×1) of the station 3-2: By the station D only

- : Stations that can be accessed from the connected device after the remote password unlock process
- : Stations that can be accessed from the connected device without performing the remote password unlock process
- ×: Stations that cannot be accessed from the connected device

Connected device ^{*1} (request source)	Target programmable controller station (request destination)					
	Station 1-1 LCPU	Station 1-2 LCPU	Station 2-1 LCPU	Station 2-2 LCPU	Station 3-1 LCPU	Station 3-2 LCPU
Station A	●	○	●	×	×	×
Station B	○	○	○	×	×	×
Station C	○	○	○	●	●	×
Station D	○	○	○	○	○	●

*1 The station A can access the stations marked with ● after the remote password unlock process is completed for 1) of the station 1-1, and can access the stations marked with ○ when the communication lines are open. The station B can access the stations marked with ○ when the communication lines are open. The station C can access the stations marked with ● after the remote password unlock process is completed for 1) of the station 2-2, and can access the stations marked with ○ when the communication lines are open. The station D can access the stations marked with ● after the remote password unlock process is completed for 1) of the station 3-2, and can access the stations marked with ○ when the communication lines are open.

Point

To disable access from the connected devices to other stations through CC-Link IE Controller Network, CC-Link IE Field Network, MELSECNET/H, and MELSECNET/10 relay communication functions, select "MELSOFT Application Transmission port (UDP/IP), Dedicated Instruction, CC-Link IE, NET10(H) Relay Transmission Port" in the remote password setting of the relay station or a station to be accessed.

12.3.4 Differences in functions according to the remote password check status (enabled/disabled)

The following table lists the behavior of the functions according to the remote password check status (enabled/disabled).

Function		Remote password check setting	
		Disabled	Enabled
Connection with MELSOFT products and a GOT		After an initial process is completed, communications are enabled by establishing a connection in the programming tool.	After the remote password is entered, communications are enabled. The remote password lock process is automatically performed when the project is closed.
MC protocol communications	User open port	After an open process is completed, communications are enabled.	After an open process is completed, communications are enabled from the time that the unlock command is received until the lock command is received.
	Auto-open UDP port	After an initial process is completed, communications are enabled.	After an initial process is completed, communications are enabled from the time that the unlock command is received until the lock command is received.
Communications using a fixed buffer	Procedure exists	After an open process is completed, communications are enabled.	After an open process is completed, communications are enabled from the time that the unlock command is received until the lock command is received.
	No procedure		After an open process is completed, communications are enabled. ^{*1}
Communications using a random access buffer			After an open process is completed, communications are enabled from the time that the unlock command is received until the lock command is received.
E-mail function		After an initial process is completed, data sending and receiving are enabled. ^{*2}	After an initial process is completed, data sending and receiving are enabled. ^{*2}
Web function		After an initial process is completed, communications are enabled.	After the remote password is entered, communications are enabled. The remote password lock process is automatically performed when the Web browser is closed.
CC-Link IE Controller Network, CC-Link IE Field Network, MELSECNET/H, and MELSECNET/10 relay communications			After an initial process is completed, communications are enabled. ^{*3}
Communications using data link instructions			
File transfer (FTP server) function		Within Ethernet, communications are enabled with the connected device that has completed an open process.	After an open process is completed, communications are enabled from the time that the unlock command is received until the lock command is received.

*1 A dedicated connection is used for "No Procedure" communications using a fixed buffer. Do not set the remote password for the connection with the connected device.

*2 Remote password check is not performed against the e-mail function.

*3 If a remote password has been set to the CPU module on the relay station or station to be accessed where the E71 is connected, access to another station may not be performed.

12.3.5 Precautions

The precautions for using the remote password function of the E71 are provided below.

(1) Timing of activating a remote password

To enable the remote password setting, power off or reset the CPU module. After a remote password has been set, restart the CPU module.

(2) Target connections

Set a remote password check only for the connections used for data communications with the connected devices that can perform unlock and lock processes.

Ex. In communications using a fixed buffer, do not set the remote password check for the connections through which the connected device receives data sent from the CPU module.

(3) Connections where "No Procedure" communications using a fixed buffer are performed

Do not set a remote password check for connections where "No Procedure" communications using a fixed buffer are performed. The remote password check is not performed for such connections.

(4) Access to the programmable controller on another station

When the connected device accesses the programmable controller on another station through the E71, it may fail to access the remote-password-protected CPU modules on the relay station or the station to be accessed.

(5) For UDP/IP communications

(a) Connected device for data communications

Do not perform data communications with unspecified connected device. Set the destination device.

(b) Alive check function

Use the alive check function of the E71.^{*1} Also, when terminating data communications, perform the remote password lock process. If the process is not performed, data communications from other devices are enabled until a timeout occurs by the alive check function of the E71. For this reason, when setting a connection with the connected device using a programming tool, set the parameters as follows.

- When configuring the initial setting, set the start interval timer value and the interval timer value for the alive check function as small as possible.
- When configuring the open setting, select "Confirm" under "Existence Confirmation".

^{*1} The alive check is automatically performed when the connection for data communications through the auto-open UDP port is set as the target for the remote password check.

(6) Communications from the programming tool in the Ethernet connection

For communications from a programming tool in the Ethernet connection, using TCP/IP communications is recommended.

(7) When the unlock process or lock process fails

If the remote password unlock/lock process fails, check the remote password of the CPU module then perform the unlock/lock process again.

(a) E71 operation in case of a process failure

If the number of process failures exceeds the notification accumulated count^{*1} set in the buffer memory area, the E71 performs the following operations.

- The E71 turns on the COM.ERR. LED.
- The E71 stores C200_H in the error code and end code storage area in the buffer memory area, Error log storage area (address: 227 (E3_H) to 372 (174_H)).

*1 This is the number of times stored in Remote password mismatch notification accumulated count designation (address: 20592 (5070_H) and 20593 (5071_H)) by the CPU module (using an instruction such as the TO instruction) when the E71 starts up.

In the above case, identify the connection where the unlock/lock process has failed using the following buffer memory areas.

- Accumulated count of unlock process abnormal end (address of Connection No.1: 20595 (5073_H))
- Accumulated count of lock process abnormal end (address of Connection No.1: 20597 (5075_H))

(b) Corrective action against a process failure

Take the following actions if required.


- Close the connection with the connected device.
- Write "0" in the buffer memory areas, Accumulated count of unlock process abnormal end (address: 20595 (5073_H)) and Accumulated count of lock process abnormal end (address: 20597 (5075_H)). If this action is not taken, the process (a) above will be performed each time a process failure occurs exceeding the notification accumulated count.
- If the number of unlock/lock process failures for the connection to the connected device is greater than the above accumulated notification count, the possible cause is unauthorized access from the connected device. Disable the connection using the buffer memory area, System port use prohibited designation area (address: 20488 (5008_H)). (After this, the unlock process cannot be performed for the connection until "Use allowed" is set.)
- Inform the system manager that the number of unlock/lock process failures is greater than the accumulated notification count, and take corrective actions.

Point

- For the method to turn off the COM.ERR. LED, refer to the method using the Ethernet diagnostics. (Page 240, Section 14.10)
- The accumulated counts of unlock/lock process abnormal end for Connection Nos.1 to 16 stored in the buffer memory can be cleared.
 - Accumulated count of unlock process abnormal end (address of Connection No.1: 20595 (5073_H))
 - Accumulated count of lock process abnormal end (address of Connection No.1: 20597 (5075_H))

12.3.6 Parameter settings

The remote password setting of the E71 is described.

 Project window ⇨ [Parameter] ⇨ [Remote Password]

Item		Description	Setting range	
Remote Password Setting	Password Setting	Enter a password set for the CPU module.	-	
	Password Active Module Setting	Model Name	Select the module model that checks an entered remote password against the remote password set for the CPU module.	
		Start XY	Set the start address of the module that checks an entered remote password.	0000 _H to 0FE0 _H
		Condition	Opens the "Remote Password Detail Setting" window.	-
Remote Password Detail Setting	User Connection No. *1*3	Connection 1 to Connection 16	Specify a connection where an entered remote password is checked.	
	System Connection *2	Auto Open UDP Port		
		FTP Transmission Port (TCP/IP)		
		MELSOFT Application Transmission Port (TCP/IP)		
		MELSOFT Application Transmission Port (UDP/IP), Dedicated Instruction, CC-Link IE, NET/10(H) Relay Transmission Port		
HTTP Port, HTTP Protocol				

*1 The user connection is used to perform MC protocol communications and fixed buffer communications.

*2 The system connection is used in the system to perform FTP communications and MELSOFT communications (TCP/IP, UDP/IP).

*3 When "MELSOFT Connection" has been selected in the open setting, select the MELSOFT application communication port in "System Connection". MELSOFT connection is performed to the MELSOFT application communication port of the system connection, not of the user connection. If the MELSOFT application communication port is not selected in "System Connection", a window for a password does not appear.

Point 

When determining a remote password, take care of the following.

- Avoid a simple character string (e.g. Character string consisting of alphanumeric characters only).
- Combine alphanumeric characters and special characters (e.g. "?", "!", "&", and "%").
- Avoid using a character string that represents the user's name or the date of birth.

12.4 Hub Connection Status Monitor Function

The current connection status and transmission speed of an E71 and a hub and the number of times that the E71 detected disconnection can be checked using the following buffer memory areas. For details on buffer memory areas, refer to the list of buffer memory addresses. (☞ Page 33, Section 3.5.2)

Buffer memory area	Description
Hub connection status area (address: 201 (C9 _H))	Stores the current connection status and transmission speed of an E71 and a hub.
Disconnection detection count (address: 20995 (5203 _H))	Stores the number of times that the E71 detected disconnection after an initial process. Disconnection is detected in any of the following cases. <ul style="list-style-type: none">• Disconnection between the E71 and a hub• Cable removal from the hub side connector• Hub power-off• Cable removal from the E71 side connector

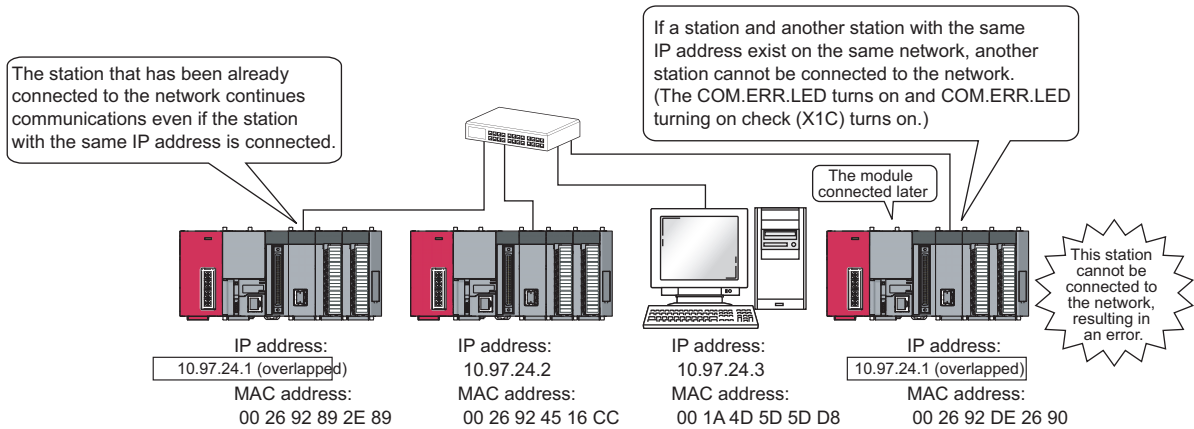
Point

If an error has occurred 65536 times or more, the count stops at 65535 (FFFF_H). Write "0" to this area using a program to clear the stored value.

12.5 IP Address in Use Detection Function

If different stations in the same network use the same IP address, the address in use can be detected. This prevents a network from stopping due to incorrect IP address.

When the connected device with the same IP address does not support the IP address in use detection function, the error is not detected.



(1) Checking the IP address already in use

The following two methods are available:

- Checking buffer memory areas
- Checking using the Ethernet diagnostics

(a) Checking buffer memory areas

Both the station already connected to the network and the station with the same IP address (station connected to the network later), and the MAC addresses of the stations with the same IP address can be checked. The destination IP address can be checked only in the station with the same IP address (station connected to the network later).

Buffer memory area		Buffer memory address	Description
Destination IP address (stored in the station with the same IP address connected to the network later.)		234 (EA _H)	Lower first word of the IP address
		235 (EB _H)	Lower second word of the IP address
IP address status storage area	The MAC address of the station that has been already connected to the network (stored in the station with the same IP address connected to the network later).	21121 (5281 _H)	Lower first word of the MAC address
		21122 (5282 _H)	Lower second word of the MAC address
		21123 (5283 _H)	Lower third word of the MAC address
	The MAC address of the station with the same IP address connected to the network later (stored in the station that has been already connected to the network).	21124 (5284 _H)	Lower first word of the MAC address
		21125 (5285 _H)	Lower second word of the MAC address
		21126 (5286 _H)	Lower third word of the MAC address


Remark

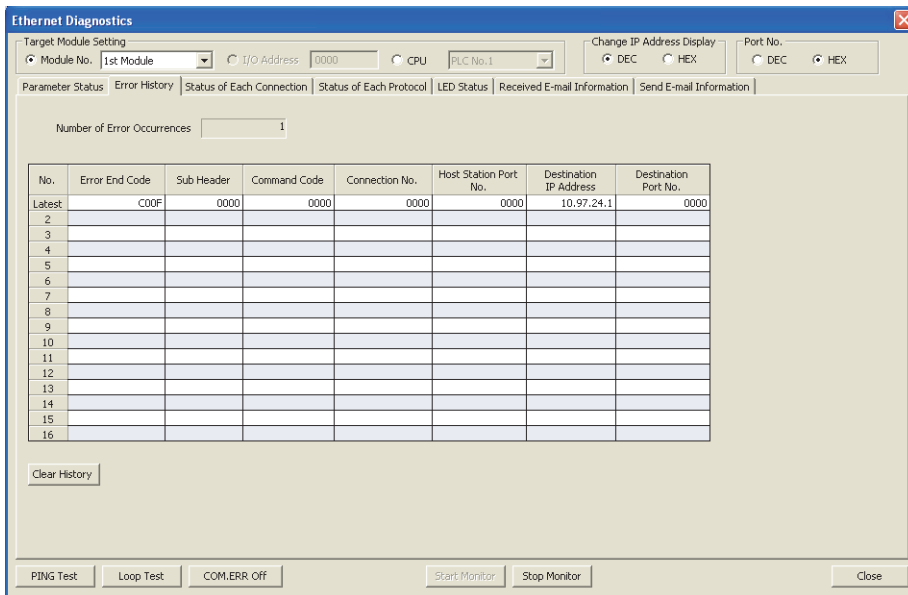
When the MAC address of the station that has been already connected to the network is 00.26.92.89.2E.89, the IP address already in use is 10.97.24.01, and the MAC address of the station connected to the network later is 00.26.92.DE.26.90, the value stored in each buffer memory area is as follows.

- 234 (EA_H): 2401 (1801_H)
- 235 (EB_H): 1097 (0A61_H)
- 21121 (5281_H): 2E89_H
- 21122 (5282_H): 9289_H
- 21123 (5283_H): 0026_H
- 21124 (5284_H): 2690_H
- 21125 (5285_H): 92DE_H
- 21126 (5286_H): 0026_H

(b) Checking using the Ethernet diagnostics

An error code (C00F_H) is displayed in the "Ethernet Diagnostics" window, and the IP address used in multiple stations is displayed in the "Destination IP Address" column. The destination IP address can be checked only in the station with the same IP address (station connected to the network later).

 [Diagnostics] ⇄ [Ethernet Diagnostics...]





12.6 Alive Check Function

When the E71 has not communicated with the connected device for a certain period of time while the connection is open, this function checks whether the connected device is alive by sending an alive check message to the connected device and waiting for the response.

(1) Parameter setting

To enable the alive check function, select "Confirm" under "Existence Confirmation" in the open setting.

( Page 67, Section 7.1.4)

In TCP/IP communications, the following methods can be selected in the Ethernet operation setting. ( Page 65, Section 7.1.3)

(2) Alive check method

The following alive check methods are available.

(a) Check using the PING command

This method is used for a connection opened using TCP/IP or UDP/IP. The E71 performs an alive check by sending the PING command (ICMP echo request/response function) to the connected device with which communications have not been performed for a certain period of time and waiting to see whether the response is received.*1

*1 The E71 automatically sends an echo response packet when it receives a PING echo request. (It sends a response to the received PING command even if the connection used in the data communications with the connected device is closed.)

(b) Check using the KeepAlive function

This method is used for a connection opened using TCP/IP. The E71 performs an alive check by sending a KeepAlive ACK message to the connected device with which communications have not been performed for a certain period of time and waiting to see whether the response is received.*2

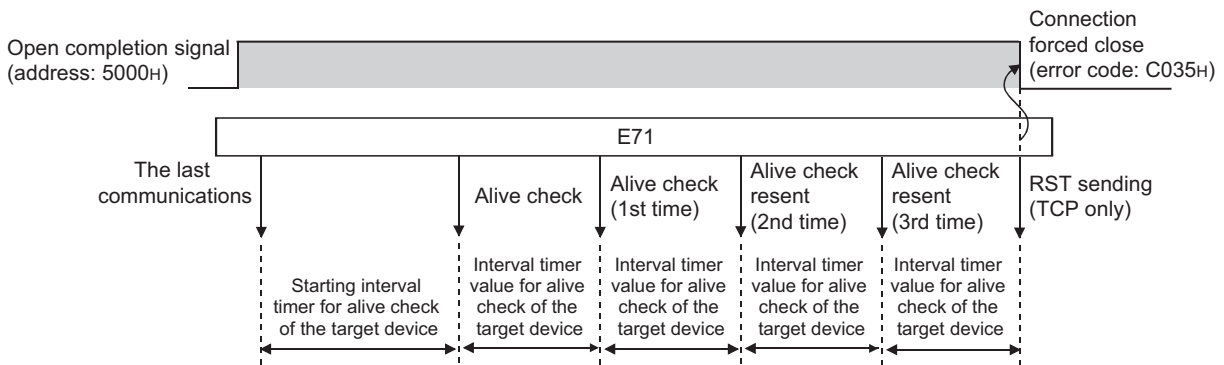
*2 The connection may be disconnected if the connected device does not support the TCP KeepAlive function (response to a KeepAlive ACK message).

(3) If a response message cannot be received from the connected device

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

- The corresponding connection will be forcibly closed. (The line is disconnected.) Open the connection using a user program again.
- Open completion signal is turned off, and the error code (C035_H) is stored in the open error code storage area.

Ex. Assuming that the values ^{*1} are set so that the number of retries may be three, the E71 performs an alive check at the timing shown in the figure below (an example of alive check using the PING command).



*1 The values of the number of retries and timers can be changed in the initial setting. (☞ Page 246, Appendix 3.1)

CHAPTER 13 DEDICATED INSTRUCTIONS

Dedicated instructions facilitate programming for using intelligent function modules. This chapter describes dedicated instructions that can be used in the E71.

13.1 List of Dedicated Instructions


(1) Dedicated instructions for using basic functions

The following table lists dedicated instructions for using the functions explained in this manual.

Instruction	Description	Reference
OPEN	Establishes (opens) a connection with the connected device to perform data communications.	Page 158, Section 13.5
CLOSE	Disconnects (closes) a connection with the connected device performing data communications.	Page 162, Section 13.6
BUFSND	Sends data to the connected device through communications using a fixed buffer.	Page 165, Section 13.7
BUFRCV	Reads data received from the connected device through communications using a fixed buffer (used in the main program).	Page 169, Section 13.8
BUFRCVS	Reads data received from the connected device through communications using a fixed buffer (used in an interrupt program).	Page 173, Section 13.9
ERRCLR	Turns off LEDs of the E71 or clears error information stored in the buffer memory.	Page 175, Section 13.10
ERRRD	Reads error information stored in the buffer memory of the E71.	Page 179, Section 13.11
UINI	Re-initializes the E71.	Page 182, Section 13.12

(2) Dedicated instructions for using special functions

The following table lists dedicated instructions for using special functions. For details on the dedicated instructions, refer to the following.

 MELSEC-Q/L Ethernet Interface Module User's Manual (Application)

Function	Instruction	Description	
E-mail function	MRECV	Reads the received e-mails.	
	MSEND	Sends e-mails.	
Communications using data link instructions	READ	Reads word device data from other stations.	
	RECV	Reads data received from other stations. (Used in the main program.)	
	RECVS	Reads data received from other stations. (Used in an interrupt program.)	
	REQ		Performs remote RUN/STOP to CPU modules on other stations.
			Reads/writes the clock data of other stations.
	SEND	Sends data to other stations.	
	SREAD	Reads word device data from other stations (with a completion device).	
	SWRITE	Writes word device data to other stations (with a completion device).	
	WRITE	Writes word device data to other stations.	
	ZNRD	Reads word device data from other stations (ACPU).	
ZNWR	Writes word device data to other stations (ACPU).		

13.2 Parameter Settings for Using Dedicated Instructions


When using dedicated instructions, set the parameters of each function.

13.2.1 When using data link instructions

When using data link instructions, set the following parameters.

- Station No. <-> IP information setting
- Routing parameters setting

For details on the settings, refer to the following.

 MELSEC-Q/L Ethernet Interface Module User's Manual (Application)

13.3 Precautions for Dedicated Instructions

This section describes precautions for using dedicated instructions.

(1) When changing data specified by dedicated instructions

Do not change any data (e.g. control data) until execution of the dedicated instruction is completed.


(2) When the dedicated instruction is not completed

Check that "Online" has been set under "Mode" in the network parameter window. Dedicated instructions cannot be executed during offline.

(3) Securing service process time

When accessing a programmable controller on another station using a data link instruction during the Ethernet diagnostics, execution of the instruction may be delayed. Take the following measures to execute the data link instruction after the Ethernet diagnostics.

- Execute the COM instruction.
- Reserve time for a communication process for 2 to 3 ms. Use a programming tool for the setting.

 Project window ⇨ [Parameter] ⇨ [PLC Parameter] ⇨ "Service Processing Setting" under the "PLC System" tab

(4) Specifying the first argument of a dedicated instruction (Un)

The first argument of a dedicated instruction (Un) can be also specified with double quotation marks (" ").

13.4 Organization of the Dedicated Instruction Sections

For the organization of the dedicated instruction sections, refer to Page 13.

13.5 ZP.OPEN

This instruction establishes (opens) a connection with the connected device to perform data communications.



Setting data ^{*1}	Available device									
	Internal device (system, user)		File register	Link direct device J□□		Intelligent function module device U□\G□	Index register Zn	Constant		Others
	Bit	Word		Bit	Word			K, H	\$	
(S1)	-	○	○			-		○	-	-
(S2)	-	○	○			-		-	-	-
(D1)	○	○	○			-		-	-	-

*1 The file registers set for each local device and program cannot be used.

(1) Setting data

Setting data	Description	Set by	Data type
Un/"Un"	Start I/O number of the E71 (00 to FE _H : The first two digits of the three-digit I/O number)	User	BIN 16-bit/character string
(S1)	Connection number (1 to 16)		BIN 16-bit
(S2)	Start number of the device that stores control data	User, system	Device name
(D1)	The start number of the bit device in the own station that turns on for one scan upon completion of the instruction. (D1)+1 also turns on if the instruction ends abnormally.	System	Bit

(2) Control data

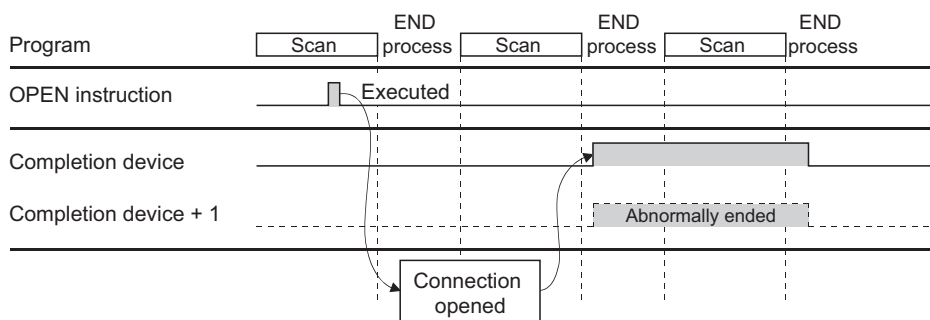
Device	Item	Setting data	Setting range	Set by
(S2)+0	Execution type/completion type	To open a connection, specify whether to use the parameter settings of a programming tool or to use the settings stored in the following control data starting from (S2)+2. <ul style="list-style-type: none"> • 0000_H: Use the parameter settings of the programming tool. • 8000_H: Use the settings of control data starting from (S2)+2 to (S2)+9. 	0000 _H , 8000 _H	User
(S2)+1	Completion status	Stores the status at completion. <ul style="list-style-type: none"> • 0000_H: Normal completion • Values other than 0000_H: Abnormal end (error code) 	-	System
(S2)+2	Usage setting area	Specify the application of a connection. <ul style="list-style-type: none"> • Usage of fixed buffer (b0) 0: For sending or fixed buffer communications are not performed 1: For receiving • Destination existence confirmation (b1) 0: No confirm 1: Confirm • Pairing open (b7) 0: Disable 1: Enable • Communication method (protocol) (b8) 0: TCP/IP 1: UDP/IP • Fixed buffer communications (b9) 0: Procedure exists 1: No procedure • Open system (b15, b14) 00: Active open or UDP/IP 10: Unpassive open 11: Fullpassive open 	As described in the left	User
(S2)+3	Own station port No.	Specify the port number of the own station.	401 _H to 1387 _H , 138B _H to FFFE _H	User
(S2)+4 (S2)+5	Destination IP address	Specify the IP address of the connected device.	1 _H to FFFFFFFF _H (FFFFFFF _H : broadcast)	User
(S2)+6	Destination port No.	Specify the port number of the connected device.	401 _H to FFFF _H (FFF _H : broadcast)	User
(S2)+7 (S2)+8 (S2)+9	Destination MAC address	Specify the MAC address of the connected device.	n 000000000000 _H FFFFFFFFFFFF _H	User

(3) Functions

- This instruction performs the open process for a connection specified by (S1) for the module specified by Un. The setting value used for the open process is selected by (S2)+0.
- Completion of the OPEN instruction can be checked with Completion device (D1)+0 and (D1)+1.

Item	Description
Completion device (D1)+0:	Turns on at the END process of the scan where the OPEN instruction is completed and turns off at the next END process.
Completion device (D1)+1:	Turns on and off depending on the completion status of the OPEN instruction. Normal completion: Stays off and does not change. Abnormal end: Turns on at the END process of the scan where the OPEN instruction is completed and turns off at the next END process.

[Operation while the OPEN instruction is being executed]



- The ZP.OPEN instruction is executed on the rising edge (OFF → ON) of the open command.

Point

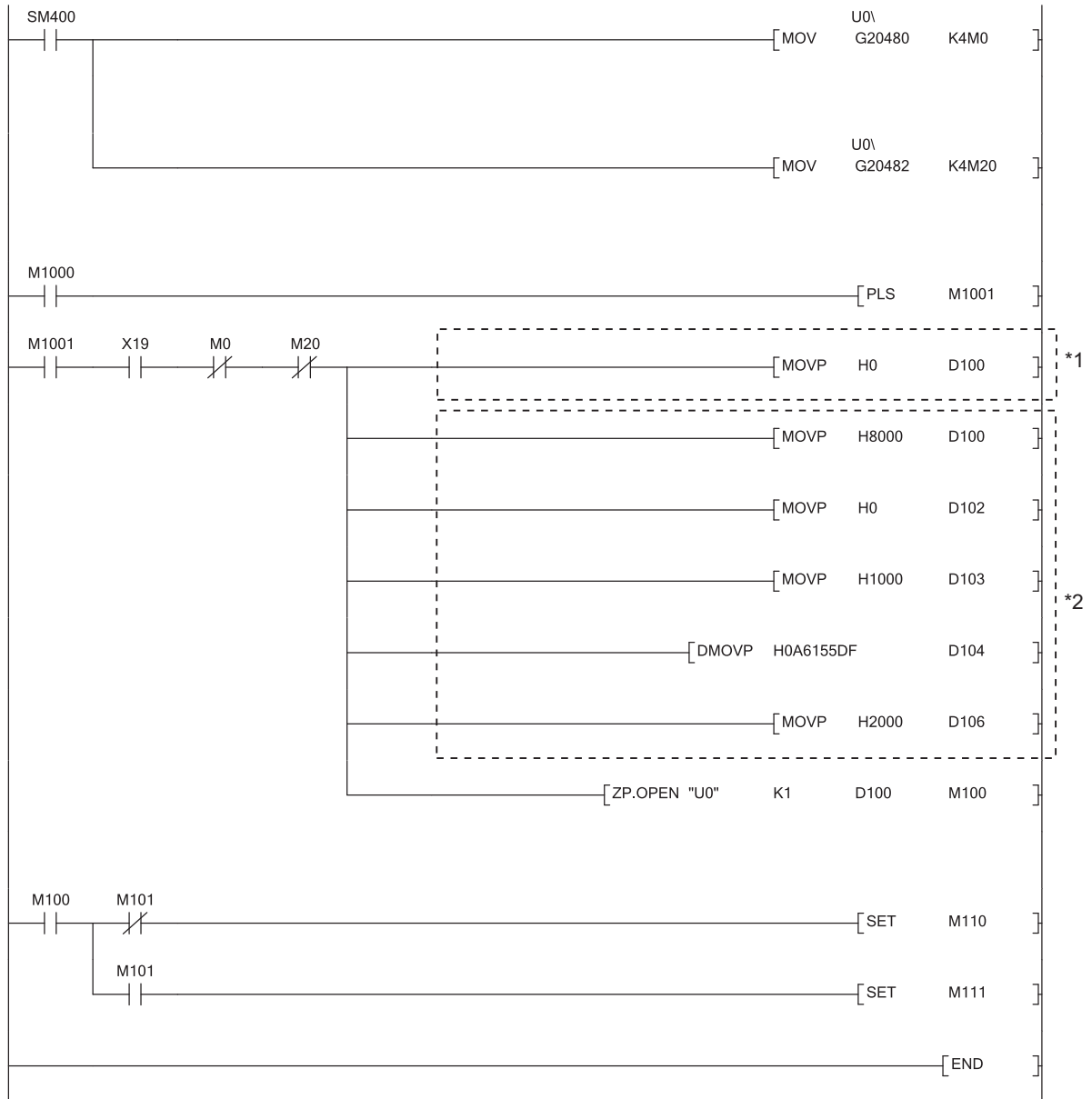
Do not perform an open or close process using I/O signals and the dedicated instruction together for the same connection. Doing so will result in malfunction.

(4) Errors

If a dedicated instruction ends with an error, Completion device (D1)+1 turns on, and the error code is stored in Completion status (S2)+1.

(5) Program example

The following shows a sample program that performs an Active open process on the connection number 1 for TCP/IP communications (when the I/O signals of the E71 are X/Y00 to X/Y1F).



- *1 Necessary when the open setting of the programming tool is used. (In this case, the program marked with *2 is unnecessary.)
- *2 Necessary when the open setting of the program is used. (In this case, the program marked with *1 is unnecessary.)

13.6 ZP.CLOSE

This instruction disconnects (closes) a connection with the connected device performing data communications.



Setting data ^{*1}	Available device									
	Internal device (system, user)		File register	Link direct device J□□		Intelligent function module device U□\G□	Index register Zn	Constant		Others
	Bit	Word		Bit	Word			K, H	\$	
(S1)	-	○	○			-		○	-	-
(S2)	-	○	○			-		-	-	-
(D1)	○	○	○			-		-	-	-

*1 The file registers set for each local device and program cannot be used.

(1) Setting data

Setting data	Description	Set by	Data type
Un/"Un"	Start I/O number of the E71 (00 to FE _H : The first two digits of the three-digit I/O number)	User	BIN 16-bit/character string
(S1)	Connection number (1 to 16)		BIN 16-bit
(S2)	Start number of the device that stores control data	System	Device name
(D1)	The start number of the bit device in the own station that turns on for one scan upon completion of the instruction. (D1)+1 also turns on if the instruction ends abnormally.		Bit

(2) Control data

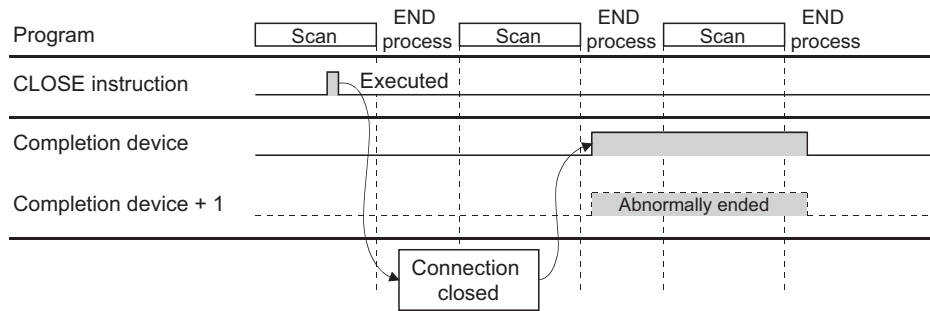
Device	Item	Setting data	Setting range	Set by
(S2)+0	System area	-	-	-
(S2)+1	Completion status	Stores the status at completion. • 0000 _H : Normal completion • Values other than 0000 _H : Abnormal end (error code)	-	System

(3) Functions

- This instruction performs the close process for a connection specified by (S1) for the module specified by Un (connection closed).
- Completion of the CLOSE instruction can be checked with Completion device (D1)+0 and (D1)+1.

Item	Description
Completion device (D1)+0:	Turns on at the END process of the scan where the CLOSE instruction is completed and turns off at the next END process.
Completion device (D1)+1:	Turns on and off depending on the completion status of the CLOSE instruction. Normal completion: Stays off and does not change. Abnormal end: Turns on at the END process of the scan where the CLOSE instruction is completed and turns off at the next END process.

[Operation while the CLOSE instruction is being executed]



- The ZP.CLOSE instruction is executed on the rising edge (OFF → ON) of the close command.

Point

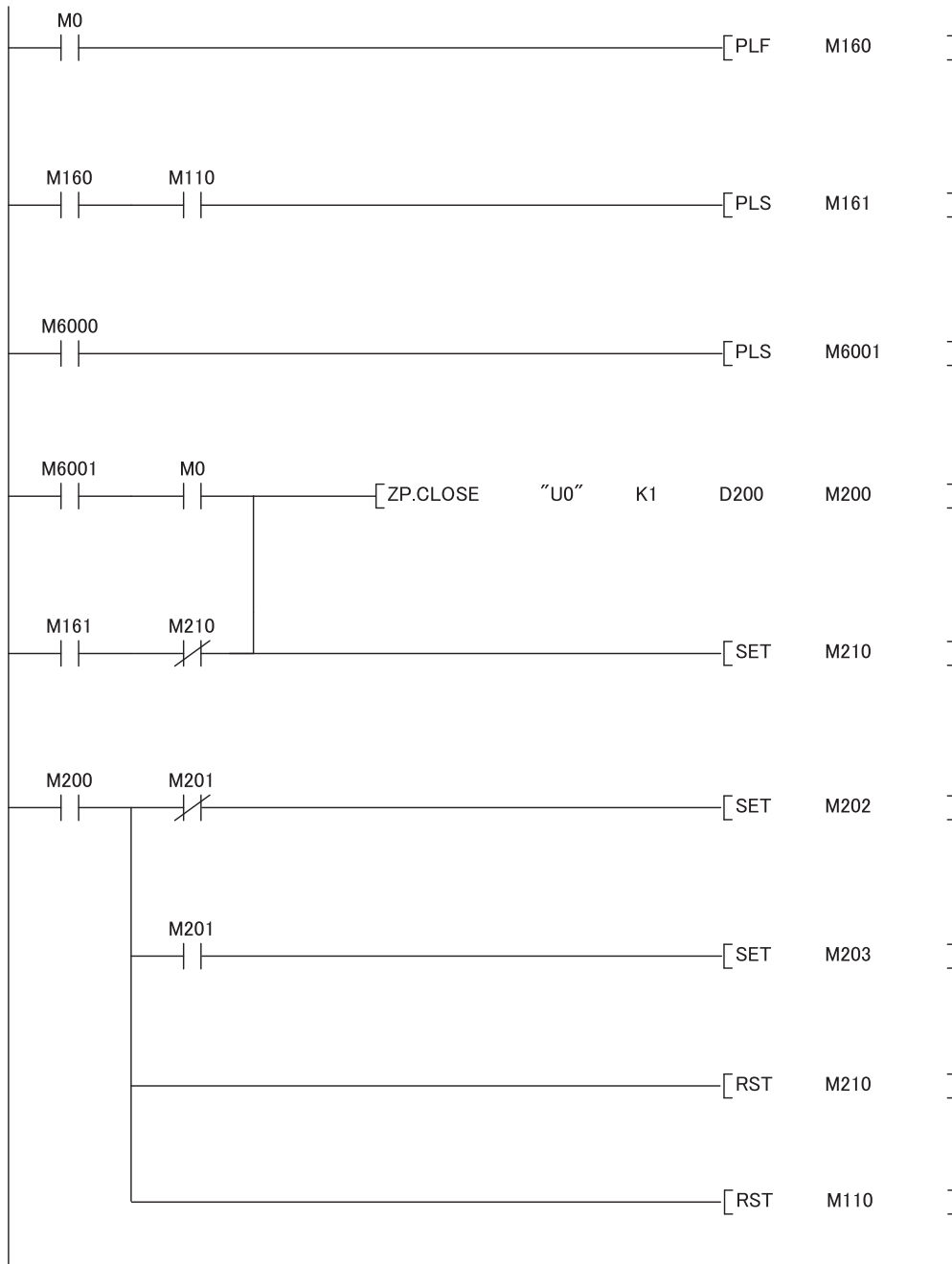
Do not perform a open/close process using I/O signals and using the dedicated instruction together for the same connection. Doing so will result in malfunction.

(4) Errors

If a dedicated instruction ends with an error, Completion device (D1)+1 turns on, and the error code is stored in Completion status (S2)+1.

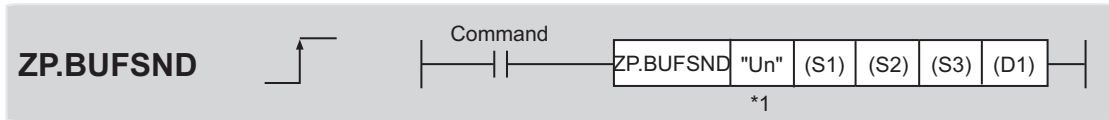
(5) Program example

The following shows a sample program that closes the connection number 1 (when the I/O signals of the E71 are X/Y00 to X/Y1F).



13.7 ZP.BUFSND

This instruction sends data to the connected device through communications using a fixed buffer.



Setting data ^{*1}	Available device									
	Internal device (system, user)		File register	Link direct device J□\□		Intelligent function module device U□\G□	Index register Zn	Constant		Others
	Bit	Word		Bit	Word			K, H	\$	
(S1)	-	○	○			-		○	-	-
(S2)	-	○	○			-		-	-	-
(S3)	-	○	○			-		-	-	-
(D1)	○	○	○			-		-	-	-

*1 The file registers set for each local device and program cannot be used.

(1) Setting data

Setting data	Description	Set by	Data type
Un/"Un"	Start I/O number of the E71 (00 to FE _H ; The first two digits of the three-digit I/O number)	User	BIN 16-bit/character string
(S1)	Connection number (1 to 16)		BIN 16-bit
(S2)	Start number of the device that stores control data	System	Device name
(S3)	Start number of the device that stores send data	User	Device name
(D1)	The start number of the bit device in the own station that turns on for one scan upon completion of the instruction. (D1)+1 also turns on if the instruction ends abnormally.	System	Bit

(2) Control data

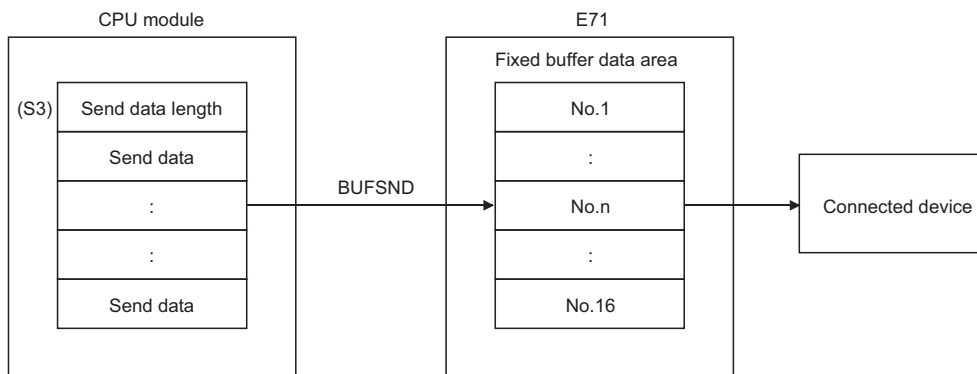
Device	Item	Setting data	Setting range	Set by
(S2)+0	System area	-	-	-
(S2)+1	Completion status	Stores the status at completion. • 0000 _H : Normal completion • Values other than 0000 _H : Abnormal end (error code)	-	System

(3) Send data

Device	Item	Setting data	Setting range	Set by
(S3)+0	Send data length	Specify send data length. (Depending on the procedure of fixed buffer communications, the data length is the number of words or the number of bytes.)	-	User
		Procedure exists (communications using a binary code): Number of words	1 to 1017	
		Procedure exists (communications using an ASCII code): Number of words	1 to 508	
		No procedure (communications using a binary code): Number of bytes	1 to 2046	
(S3)+1 to (S3)+n	Send data	Specify the send data.	-	User

(4) Functions

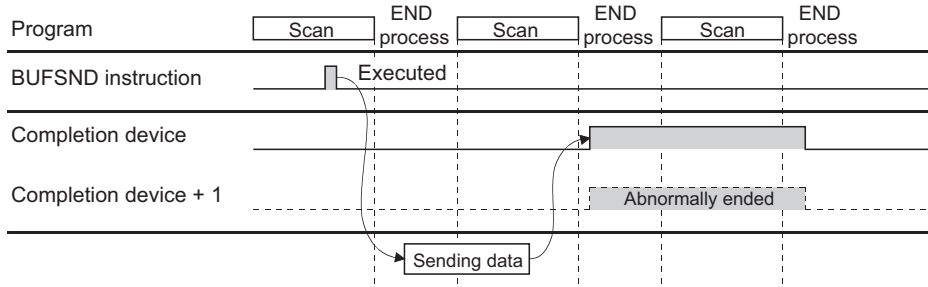
- This instruction sends the data specified by (S3) to the connected device of the connection specified by (S1) for the module specified by Un.



- Completion of the BUFSND instruction can be checked with Completion device (D1)+0 and (D1)+1.

Item	Description
Completion device (D1)+0:	Turns on at the END process of the scan where the BUFSND instruction is completed and turns off at the next END process.
Completion device (D1)+1:	Turns on and off depending on the completion status of the BUFSND instruction. Normal completion: Stays off and does not change. Abnormal end: Turns on at the END process of the scan where the BUFSND instruction is completed and turns off at the next END process.

[Operation while the BUFSND instruction is being executed]



- The ZP.BUFSND instruction is executed on the rising edge (OFF → ON) of the sending command.

Point

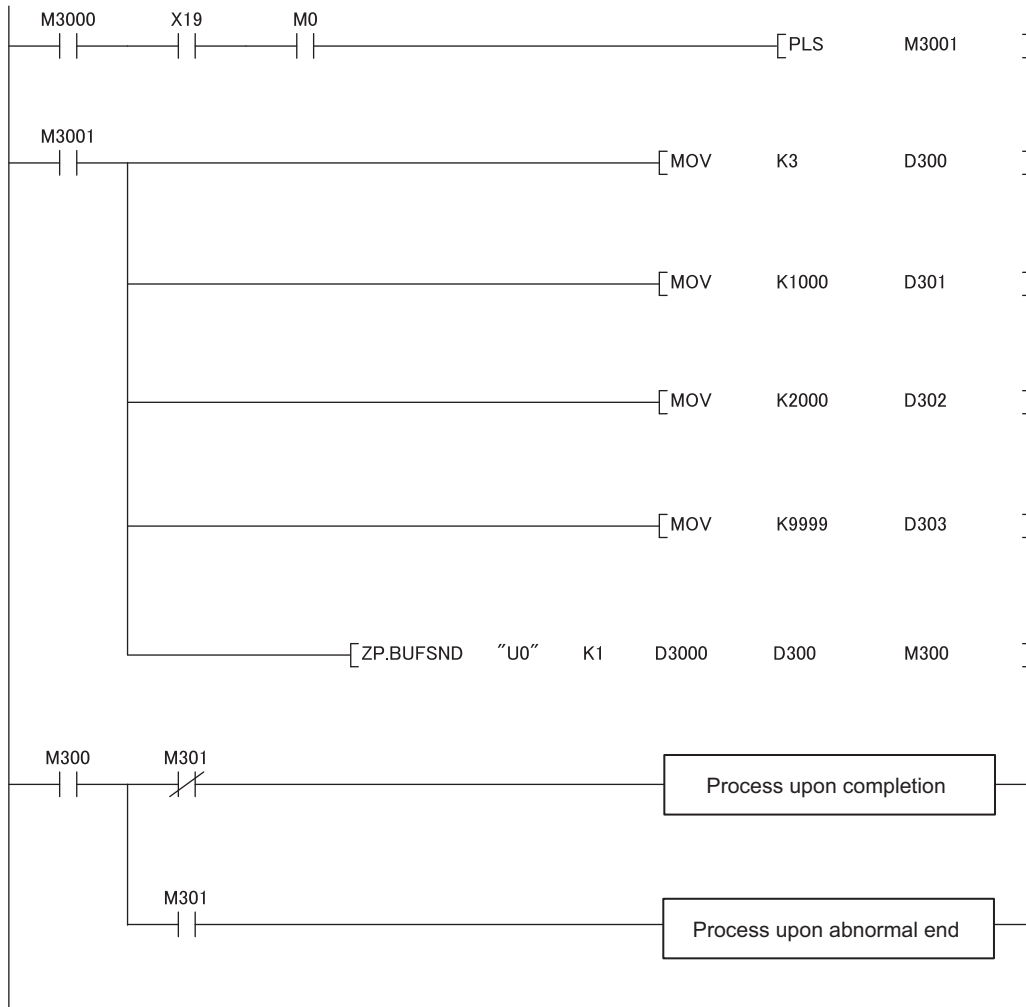
Do not perform a send process using I/O signals and using the dedicated instruction together for the same connection. Doing so will result in malfunction.

(5) Errors

- If a dedicated instruction ends with an error, Completion device (D1)+1 turns on, and the error code is stored in Completion status (S2)+1.

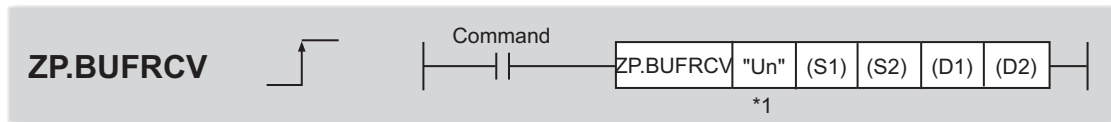
(6) Program example

The following shows a sample program that sends data stored on the fixed buffer of the connection number 1 (when the I/O signals of the E71 are X/Y00 to X/Y1F).



13.8 ZP.BUFRCV

This instruction reads data received from the connected device through communications using a fixed buffer (used in the main program).



Setting data ^{*1}	Available device									
	Internal device (system, user)		File register	Link direct device J□\□		Intelligent function module device U□\G□	Index register Zn	Constant		Others
	Bit	Word		Bit	Word			K, H	\$	
(S1)	-	○	○			-		○	-	-
(S2)	-	○	○			-		-	-	-
(D1)	-	○	○			-		-	-	-
(D2)	○	○	○			-		-	-	-

*1 The file registers set for each local device and program cannot be used.

(1) Setting data

Setting data	Description	Set by	Data type
Un/"Un"	Start I/O number of the E71 (00 to FE _H ; The first two digits of the three-digit I/O number)	User	BIN 16-bit/character string
(S1)	Connection number (1 to 16)		BIN 16-bit
(S2)	Start number of the device that stores control data	System	Device name
(D1)	Start number of the device that stores receive data		Device name
(D2)	The start number of the bit device in the own station that turns on for one scan upon completion of the instruction. (D2)+1 also turns on if the instruction ends abnormally.		Bit

(2) Control data

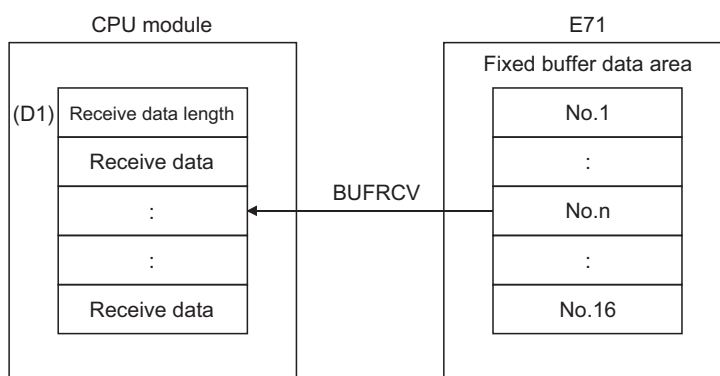
Device	Item	Setting data	Setting range	Set by
(S2)+0	System area	-	-	-
(S2)+1	Completion status	Stores the status at completion. • 0000 _H : Normal completion • Values other than 0000 _H : Abnormal end (error code)	-	System

(3) Receive data

Device	Item	Setting data	Setting range	Set by
(D1)+0	Receive data length	Stores the data length of the data read from the fixed buffer data area. (Depending on the procedure of fixed buffer communications, the data length is the number of words or the number of bytes.)	-	System
		Procedure exists (communications using a binary code): Number of words	1 to 1017	
		Procedure exists (communications using an ASCII code): Number of words	1 to 508	
		No procedure (communications using a binary code): Number of bytes	1 to 2046	
(D1)+1 to (D1)+n	Receive data	Stores the data read from the fixed buffer data area starting from the smallest address.	-	System

(4) Functions

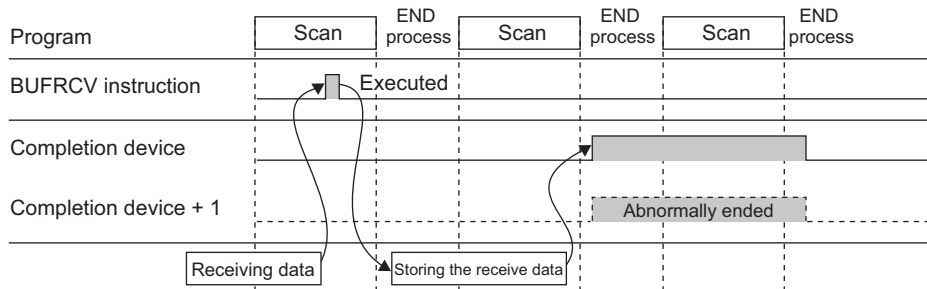
- This instruction reads the receive data (for fixed buffer communications) of the connection specified by (S1) for the module specified by Un.



- Completion of the BUFRCV instruction can be checked with Completion device (D2)+0 and (D2)+1.

Item	Description
Completion device (D2)+0:	Turns on at the END process of the scan where the BUFRCV instruction is completed and turns off at the next END process.
Completion device (D2)+1:	Turns on and off depending on the completion status of the BUFRCV instruction. Normal completion: Stays off and does not change. Abnormal end: Turns on at the END process of the scan where the BUFRCV instruction is completed and turns off at the next END process.

[Operation while the BUFRCV instruction is being executed]



- The ZP.BUFRCV instruction is executed on the rising edge (OFF → ON) of the read command (bit for a connection with the connected device set in Fixed buffer reception status signal (address: 5005H)).

Point

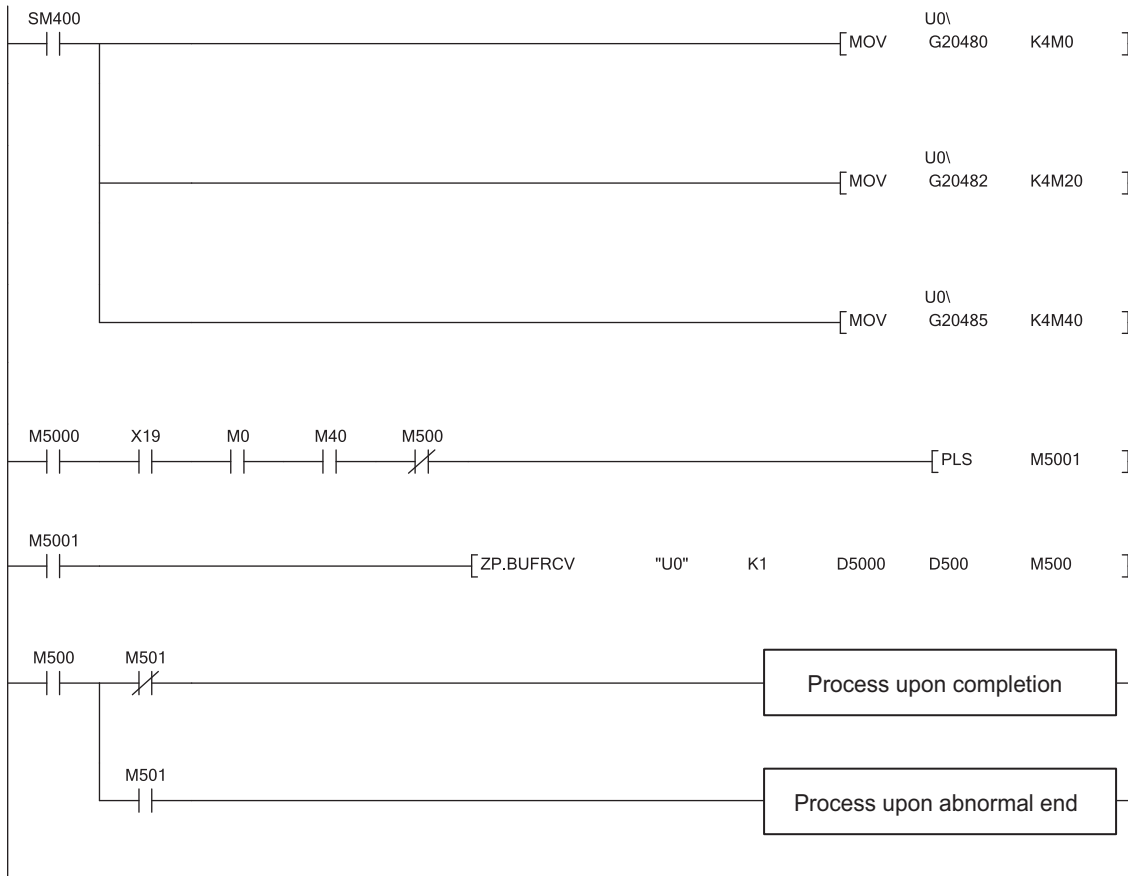
- Do not perform a receive process using I/O signals and using the dedicated instruction together for the same connection. Doing so will result in malfunction.
- When receive data is read from the same connection, the BUFRCVS instruction (for an interrupt program) cannot be used together.

(5) Errors

If a dedicated instruction ends with an error, Completion device (D2)+1 turns on, and the error code is stored in Completion status (S2)+1.

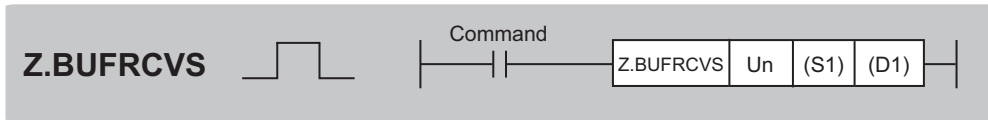
(6) Program example

The following shows a sample program that reads the receive data from the fixed buffer of connection number 1 (when the I/O signals of the E71 are X/Y00 to X/Y1F).



13.9 Z.BUFRCVS

This instruction reads data received from the connected device through communications using a fixed buffer (used in an interrupt program).



Setting data ^{*1}	Available device									
	Internal device (system, user)		File register	Link direct device J□\□		Intelligent function module device U□\G□	Index register Zn	Constant		Others
	Bit	Word		Bit	Word			K, H	\$	
(S1)	-	○	○			-		○	-	-
(D1)	-	○	○			-		-	-	-

*1 The file registers set for each local device and program cannot be used.

(1) Setting data

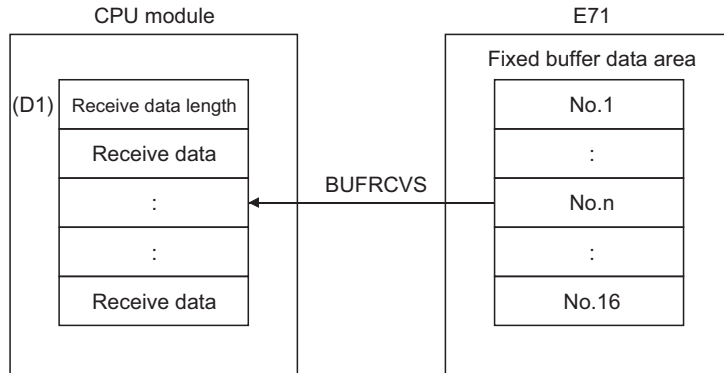
Setting data	Description	Set by	Data type
Un/"Un"	Start I/O number of the E71 (00 to FE _H ; The first two digits of the three-digit I/O number)	User	BIN 16-bit/character string
(S1)	Connection number (1 to 16)		BIN 16-bit
(D1)	Start number of the device that stores receive data	System	Device name

(2) Receive data

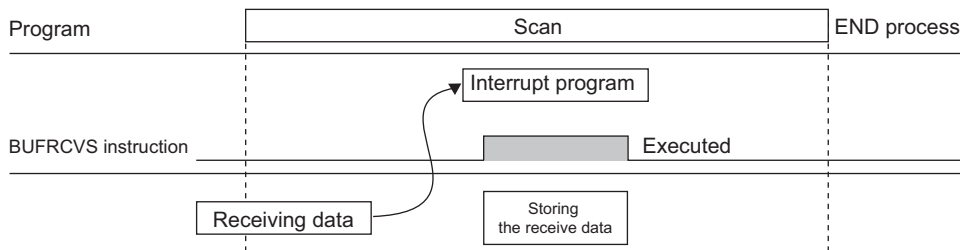
Device	Item	Setting data	Setting range	Set by
(D1)+0	Receive data length	Stores the data length of the data read from the fixed buffer data area. (Depending on the procedure of fixed buffer communications, the data length is the number of words or the number of bytes.)	-	System
		Procedure exists (communications using a binary code): Number of words	1 to 1017	
		Procedure exists (communications using an ASCII code): Number of words	1 to 508	
		No procedure (communications using a binary code): Number of bytes	1 to 2046	
(D1)+1 to (D1)+n	Receive data	Stores the data read from the fixed buffer data area starting from the smallest address.	-	System

(3) Functions

- This instruction reads the receive data (for fixed buffer communications) of the connection specified by (S1) for the module specified by Un.



[Operation while the BUFRCVS instruction is being executed]



- The Z.BUFRCVS instruction is executed in an interrupt program, and the process is completed within one scan.

Point

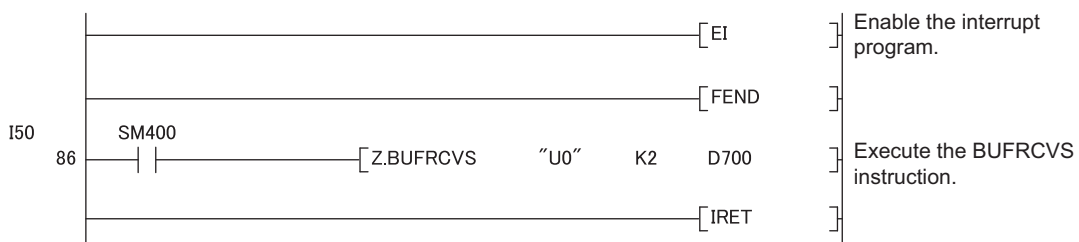
- To read receive data using an interrupt program, configure both the interrupt settings and interrupt pointer setting parameters of the programming tool. (☞ Page 98, Section 10.5.1)
- When receive data is read from the same connection, the BUFRCV instruction (for the main program) cannot be used together.

(4) Errors

If a dedicated instruction ends with an error, Diagnostic errors (SM0) turns on, and the error code is stored in SD0.

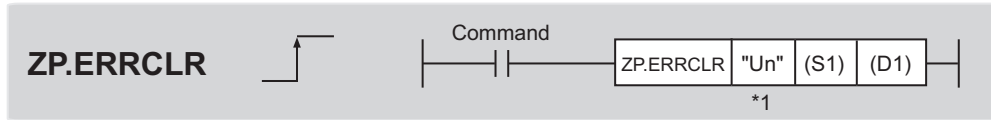
(5) Program example

The following shows a sample program that reads the receive data from the fixed buffer of connection number 2 (when the I/O signals of the E71 are X/Y00 to X/Y1F).



13.10 ZP.ERRCLR

This instruction turns off LEDs of the E71 or clears error information stored in the buffer memory.



Setting data ^{*1}	Available device									
	Internal device (system, user)		File register	Link direct device J□\□		Intelligent function module device U□\G□	Index register Zn	Constant		Others
	Bit	Word		Bit	Word			K, H	\$	
(S1)	-	○	○			-		-	-	-
(D1)	○	○	○			-		-	-	-

*1 The file registers set for each local device and program cannot be used.

(1) Setting data

Setting data	Description	Set by	Data type
Un/"Un"	Start I/O number of the E71 (00 to FE _H ; The first two digits of the three-digit I/O number)	User	BIN 16-bit/character string
(S1)	Start number of the device that stores control data	User, system	Device name
(D1)	The start number of the bit device in the own station that turns on for one scan upon completion of the instruction. (D1)+1 also turns on if the instruction ends abnormally.	System	Bit

(2) Control data

Device	Item	Setting data	Setting range	Set by
(S1)+0	System area	-	-	-
(S1)+1	Completion status	Stores the status at completion. • 0000 _H : Normal completion • Values other than 0000 _H : Abnormal end (error code)	-	System
(S1)+2	Clear target specification	Specify error information to be cleared. • 0000 _H : Initial error code • 0001 _H to 0010 _H : Open error code for a connection with the connected device • 0100 _H : Error log block area • 0101 _H : Communication status - status for each protocol • 0102 _H : Communication status - e-mail receiving status • 0103 _H : Communication status - e-mail sending status • FFFF _H : Clears all of the above	As listed in the left	User
(S1)+3	Clear function specification	Specify the function to be cleared. • 0000 _H : COM.ERR LED off, error code clear • FFFF _H : Error log clear	0000 _H , FFFF _H	User
(S1)+4 to (S1)+7	System area	-	-	-

(3) Functions

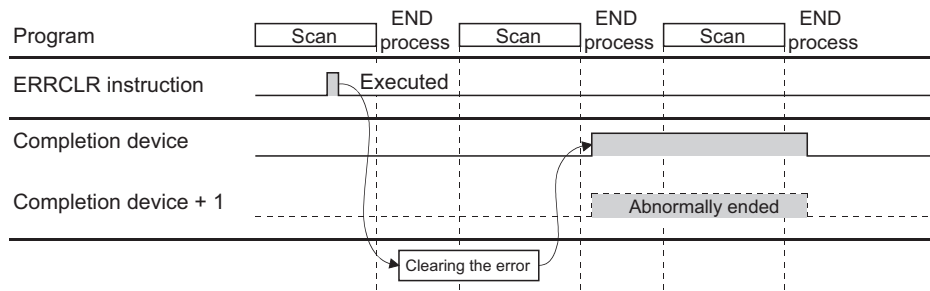
- This instruction turns off the COM.ERR. LED and clears error information listed below for the module specified by Un.

Item	Target specification (S1)+2	Function specification (S1)+3	Error information to be cleared (buffer memory)	
Initial error	0000 _H	0000 _H	• Initial error code (address: 69 _H) • COM.ERR. LED off	
Open error	0001 _H to 0010 _H	0000 _H	Open error code for a connection with the connected device (address: 7C _H , 86 _H ...)	
Error log	0100 _H	FFFF _H	Error log (address: E3 _H to 174 _H)	
Communication status	Status for each protocol	0101 _H	FFFF _H	Clears communication status. (address: 178 _H to 1FF _H)
	E-mail receiving status	0102 _H	FFFF _H	E-mail receiving (address: 5871 _H to 5B38 _H)
	E-mail sending status	0103 _H	FFFF _H	E-mail sending (address: 5B39 _H to 5CA0 _H)
All	FFFF _H	FFFF _H	Clears all of the above.	

- Completion of the ERRCLR instruction can be checked with Completion device (D1)+0 and (D1)+1.

Item	Description
Completion device (D1)+0:	Turns on at the END process of the scan where the ERRCLR instruction is completed and turns off at the next END process.
Completion device (D1)+1:	Turns on and off depending on the completion status of the ERRCLR instruction. Normal completion: Stays off and does not change. Abnormal end: Turns on at the END process of the scan where the ERRCLR instruction is completed and turns off at the next END process.

[Operation while the ERRCLR instruction is being executed]



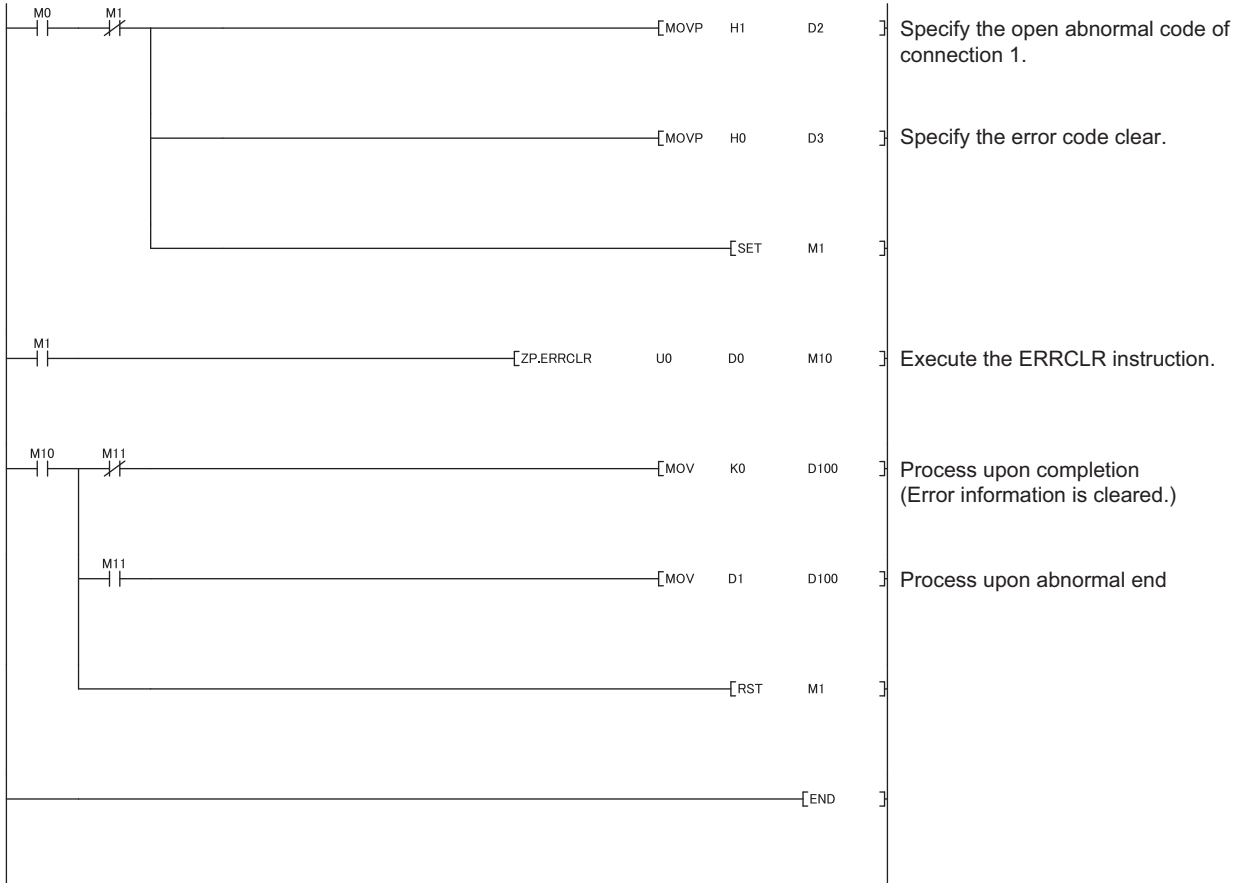
- The ZP.ERRCLR instruction is executed on the rising edge (OFF → ON) of the clear command.

(4) Errors

If a dedicated instruction ends with an error, Completion device (D1)+1 turns on, and the error code is stored in Completion status (S1)+1.

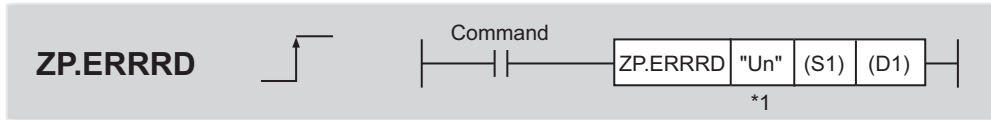
(5) Program example

The following shows a sample program that clears the open error code of the connection number 1 (when the I/O signals of the E71 are X/Y00 to X/Y1F).



13.11 ZP.ERRRD

This instruction reads error information stored in the buffer memory of the E71.



Setting data ^{*1}	Available device									
	Internal device (system, user)		File register	Link direct device J□\□		Intelligent function module device U□\G□	Index register Zn	Constant		Others
	Bit	Word		Bit	Word			K, H	\$	
(S1)	-	○	○			-		-	-	-
(D1)	○	○	○			-		-	-	-

*1 The file registers set for each local device and program cannot be used.

(1) Setting data

Setting data	Description	Set by	Data type
Un/"Un"	Start I/O number of the E71 (00 to FE _H ; The first two digits of the three-digit I/O number)	User	BIN 16-bit/character string
(S1)	Start number of the device that stores control data	User, system	Device name
(D1)	The start number of the bit device in the own station that turns on for one scan upon completion of the instruction. (D1)+1 also turns on if the instruction ends abnormally.	System	Bit

(2) Control data

Device	Item	Setting data	Setting range	Set by
(S1)+0	System area	-	-	-
(S1)+1	Completion status	Stores the status at completion. • 0000 _H : Normal completion • Values other than 0000 _H : Abnormal end (error code)	-	System
(S1)+2	Read information specification	Specify error information to be read. • 0000 _H : Initial error code • 0001 _H to 0010 _H : Open error code for a connection with the connected device	0000 _H , 0001 _H to 0010 _H	User
(S1)+3	Read target information specification	Specify whether to read the latest error information. • 0000 _H : Reads the latest error information.	0000 _H	User
(S1)+4	Error information	Stores the error information that has been read. • 0000 _H : No error • Values other than 0000 _H (error code): Abnormal end	-	System
(S1)+5 to (S1)+7	System area	-	-	-

(3) Functions

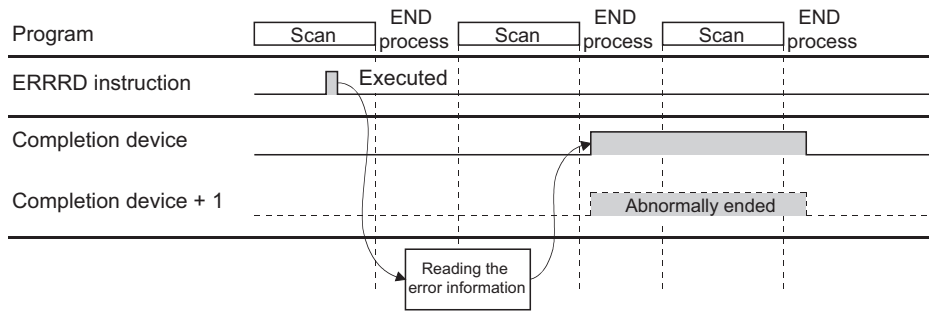
- This instruction reads the error information of the module specified for Un.

Item	Target specification (S1)+2	Function specification (S1)+3	Error information to be read (buffer memory)
Initial error	0000 _H	0000 _H	Initial error code (address: 69 _H)
Open error	0001 _H to 0010 _H	0000 _H	Open error code for a connection with the connected device (address: 7C _H , 86 _H ...)

- Completion of the ERRRD instruction can be checked with Completion device (D1)+0 and (D1)+1.

Item	Description
Completion device (D1)+0:	Turns on at the END process of the scan where the ERRRD instruction is completed and turns off at the next END process.
Completion device (D1)+1:	Turns on and off depending on the completion status of the ERRRD instruction. Normal completion: Stays off and does not change. Abnormal end: Turns on at the END process of the scan where the ERRRD instruction is completed and turns off at the next END process.

[Operation while the ERRRD instruction is being executed]



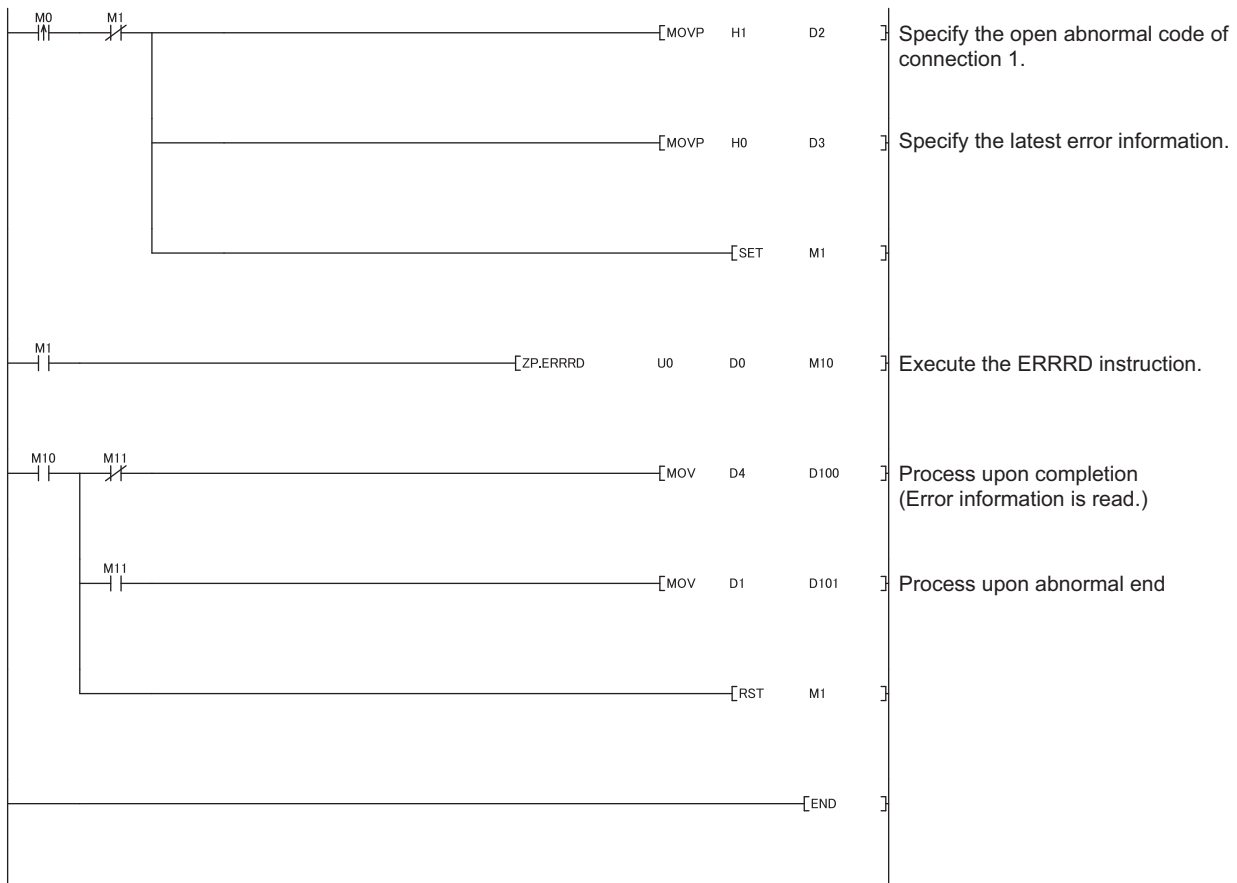
- The ZP.ERRRD instruction is executed on the rising edge (OFF → ON) of the read command.

(4) Errors

If a dedicated instruction ends with an error, Completion device (D1)+1 turns on, and the error code is stored in Completion status (S1)+1.

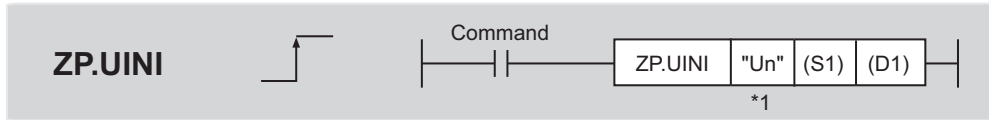
(5) Program example

The following shows a sample program that reads the open error code of the connection number 1 (when the I/O signals of the E71 are X/Y00 to X/Y1F).



13.12 ZP.UINI

This instruction changes the setting, such as the Ethernet operation setting, and re-initializes the E71.



Setting data ^{*1}	Available device									
	Internal device (system, user)		File register	Link direct device J□\□		Intelligent function module device U□\G□	Index register Zn	Constant		Others
	Bit	Word		Bit	Word			K, H	\$	
(S1)	-	○	○			-		-	-	-
(D1)	○	○	○			-		-	-	-

*1 The file registers set for each local device and program cannot be used.

(1) Setting data

Setting data	Description	Set by	Data type
Un/"Un"	Start I/O number of the E71 (00 to FE _H : The first two digits of the three-digit I/O number)	User	BIN 16-bit/character string
(S1)	Start number of the device that stores control data	User, system	Device name
(D1)	The start number of the bit device in the own station that turns on for one scan upon completion of the instruction. (D1)+1 also turns on if the instruction ends abnormally.	System	Bit

(2) Control data

Device	Item	Setting data	Setting range	Set by
(S1)+0	System area	-	-	-
(S1)+1	Completion status	Stores the status at completion. <ul style="list-style-type: none"> • 0000_H: Normal completion • Values other than 0000_H: Abnormal end (error code) 	-	System
(S1)+2	Specification of changed item	Specify 0000 _H to update the address information on the connected devices retained by the E71. Specify the target for setting change among the own station IP address, Ethernet operation setting, transmission speed, and communication mode. Do not specify the transmission speed and communication mode together with the own station IP address and Ethernet operation setting. If doing so, only the specification of the own station IP address and Ethernet operation setting is updated. <ul style="list-style-type: none"> • Specification of own station IP address change (b0): Specify whether to change the own station IP address. (Set the address in (S1)+3 and (S1)+4.) 0: Do not change 1: Change • Specification of the Ethernet operation setting change (b1): Specify whether to change the Ethernet operation setting. (Set the parameters in (S1)+5.) 0: Do not change 1: Change • Specification of transmission speed and communication mode changes (b12 to b15): Specify the transmission speed and the communicate mode. 0: Do not change 1: Auto negotiation 2: 100Mbps/full-duplex 3: 100Mbps/half-duplex 4: 10Mbps/full-duplex 5: 10Mbps/half-duplex 	0000 _H to 5000 _H	User
(S1)+3 (S1)+4	Own station IP address	Specify the IP address of the own station.	00000001 _H to FFFFFFFE _H	User

Device	Item	Setting data	Setting range	Set by
(S1)+5	Ethernet operation setting	Specify the Ethernet operation setting. <ul style="list-style-type: none"> • Communication data code setting (b1) 0: Binary Code 1: ASCII Code • TCP existence confirmation setting (b4) 0: Use the Ping 1: Use the KeepAlive • Send frame setting (b5) 0: Ethernet 1: IEEE 802.3 • Setting of write enable/disable at RUN time (b6) 0: Disable 1: Enable • Initial timing setting (b8) 0: Do not wait for OPEN (Communications impossible at STOP time) 1: Always wait for OPEN (Communications possible at STOP time) 	As listed in the left	User
(S1)+6 (S1)+7	System area	-	-	-

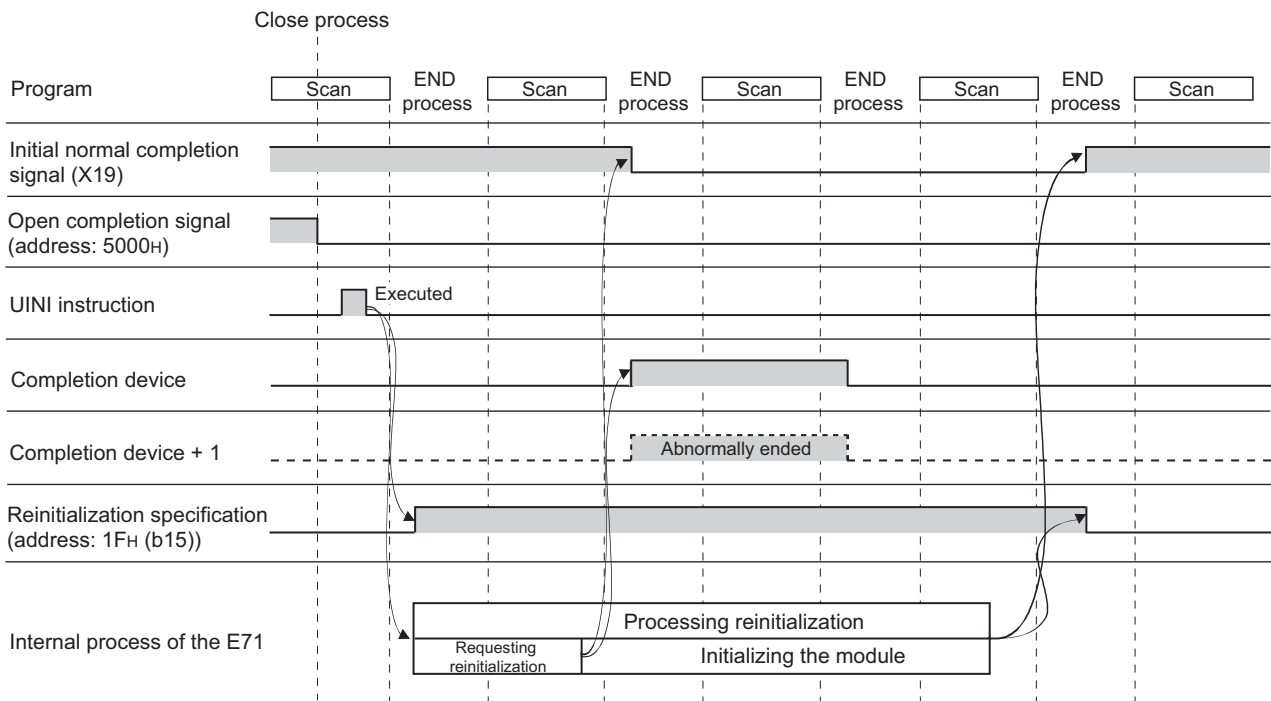
*1 The E71 enables data exchange to be resumed by clearing the address information of the connected device retained in the E71 and performing a reinitialization process. (Initial normal completion signal (X19) turns on.)

(3) Functions

- This instruction re-initializes the module specified by Un.
- Completion of the UINI instruction can be checked with Completion device (D1)+0 and (D1)+1.

Item	Description
Completion device (D1)+0:	Turns on at the END process of the scan where the UINI instruction is completed and turns off at the next END process.
Completion device (D1)+1:	Turns on and off depending on the completion status of the UINI instruction. Normal completion: Stays off and does not change. Abnormal end: Turns on at the END process of the scan where the UINI instruction is completed and turns off at the next END process.

[Operation while the UINI instruction is being executed]



- The ZP.UINI instruction is executed on the rising edge (OFF → ON) of the reinitialization command.

Point

To re-initialize the E71, note the following.

- Terminate all data communications with connected devices, check that all connections are closed, then re-initialize the E71.
- Do not perform a reinitialization process using both direct writing to the buffer memory and the UINI instruction. Also, do not request another reinitialization process while reinitialization is being performed.
- When the IP address of the E71 has been changed, reset that of the connected device as well. (If the connected device retains the MAC address of the destination device, the communications may not be continued after the IP address of the E71 has been changed.)

(4) Errors

If a dedicated instruction ends with an error, Completion device (D1)+1 turns on, and the error code is stored in Completion status (S1)+1.

(5) Program example

The following pages show sample programs that change the following setting items using the UINI instruction.

- Ethernet operation setting
- Transmission speed and communication mode

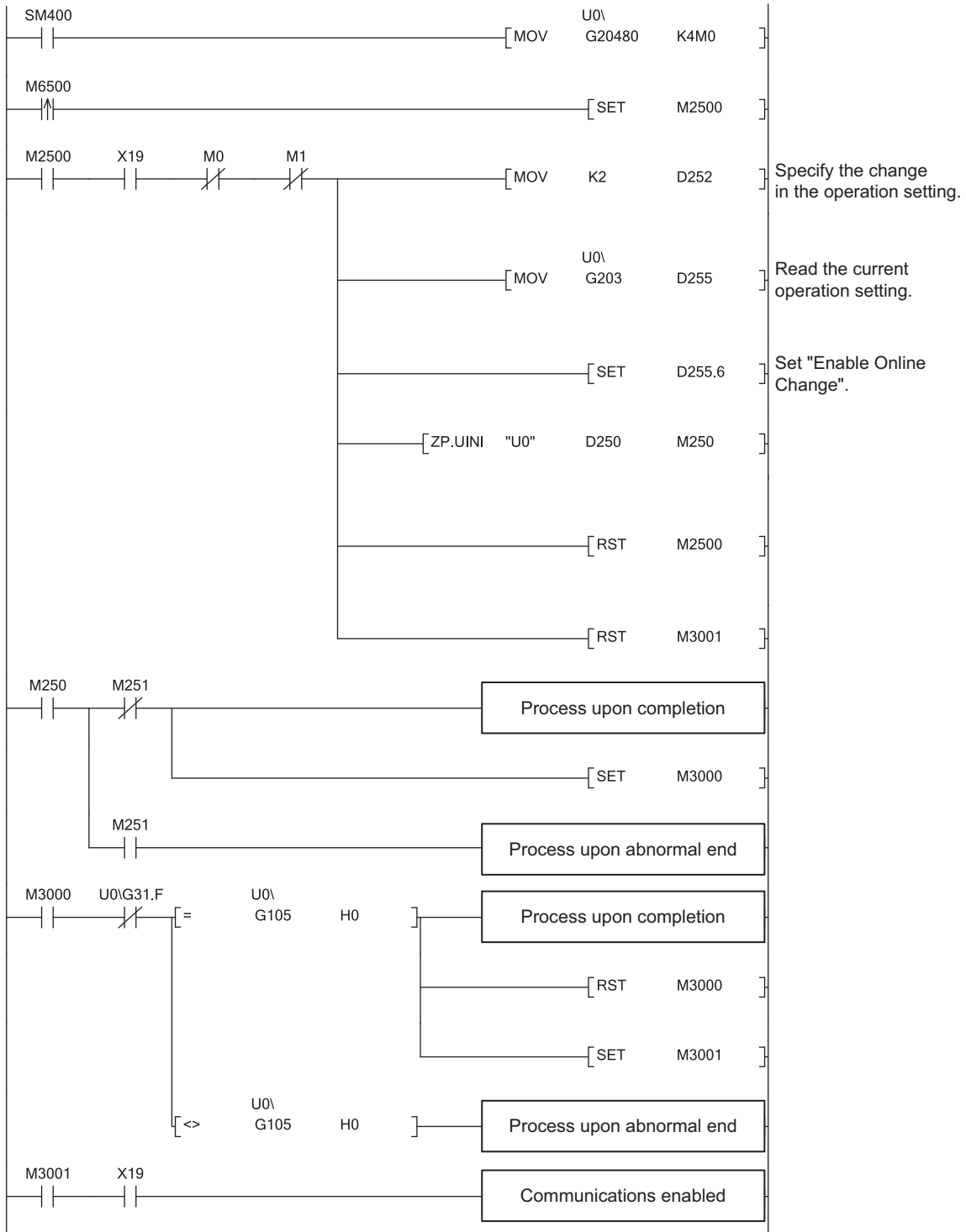
Remark

The following are sample programs for communications using connection numbers 1 and 2. When using other connections, specify the corresponding signals and bits for each connection.

The UINI instruction is also used in the program for a reinitialization process. When performing a reinitialization process using the UINI instruction, refer to the program for a reinitialization process. (☞ Page 250, Appendix 3.2)

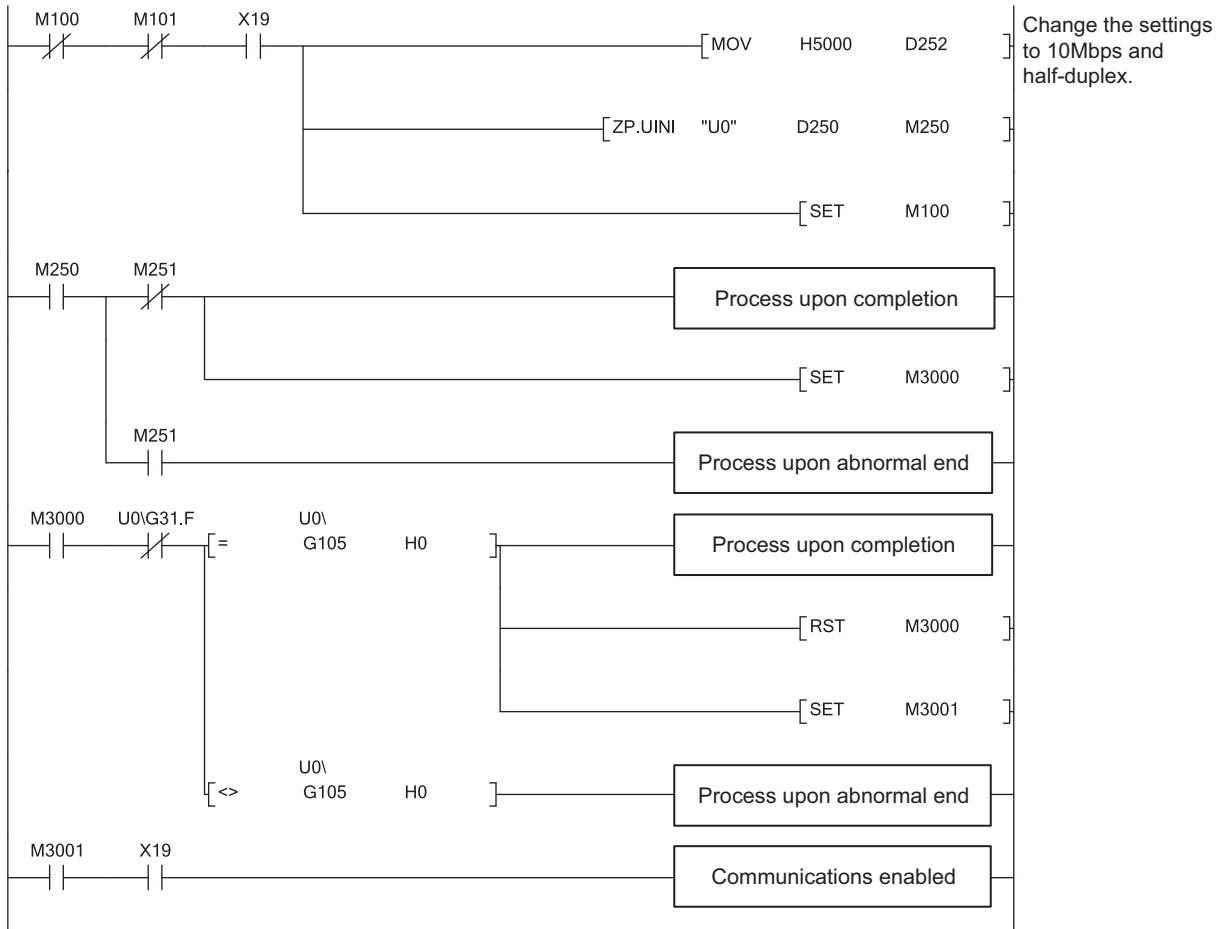
(a) Changing the Ethernet operation setting

The following page provides a sample program that changes the Ethernet operation setting (enables writing during RUN) (when the I/O signals of the E71 are X/Y00 to X/Y1F).



(b) Changing the transmission speed and communication mode

The following shows a sample program that changes the transmission speed to 10Mbps and the communication mode to half-duplex (when the I/O signals of the E71 are X/Y00 to X/Y1F).



CHAPTER 14 TROUBLESHOOTING

This chapter describes how to identify causes of problems that occur when an E71 is used and take corrective action.

14.1 Before Troubleshooting

Check that the POWER LED of the power supply module and the MODE LED of the CPU module are on. If either or both are off, troubleshoot the CPU module.

User's Manual (hardware design, maintenance and inspection) for the CPU module used

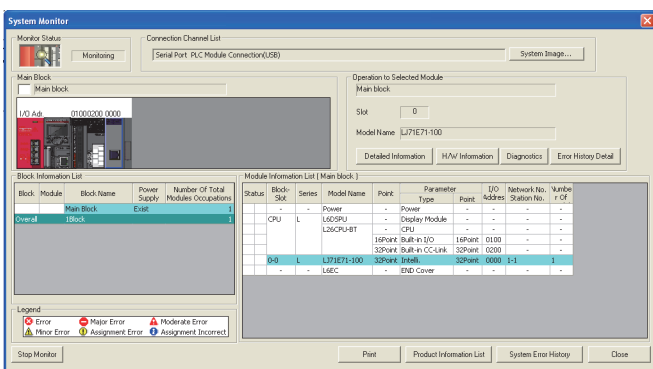
14.2 Troubleshooting Procedure

How to identify the cause of a problem and take corrective action is explained below. Use a programming tool to identify the cause and take corrective action.

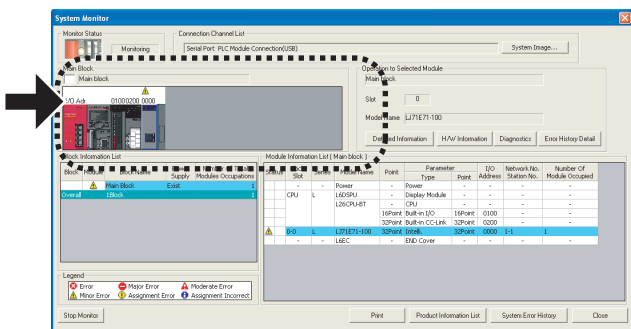
(1) Procedure

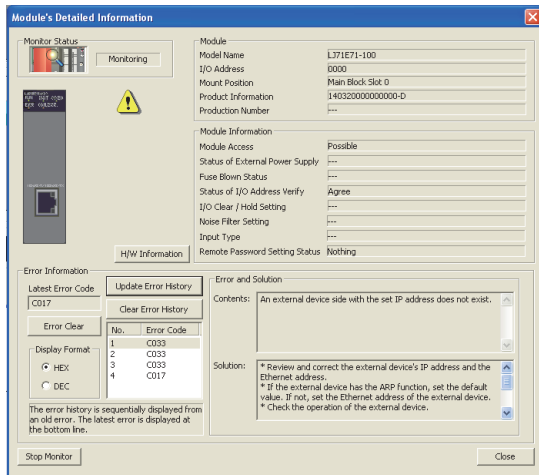
1. Connect the CPU module to a programming tool, and open the "System Monitor" window.

[Diagnostics] ⇒ [System Monitor]



2. When an error in the E71 is reported, select the E71 and click the **Detailed Information** button. When an error in a module other than the E71 is reported, refer to the manual for that module.





3. The "Module's Detailed Information" window is displayed. Clicking the **Update Error History** button displays the error details and corrective action.

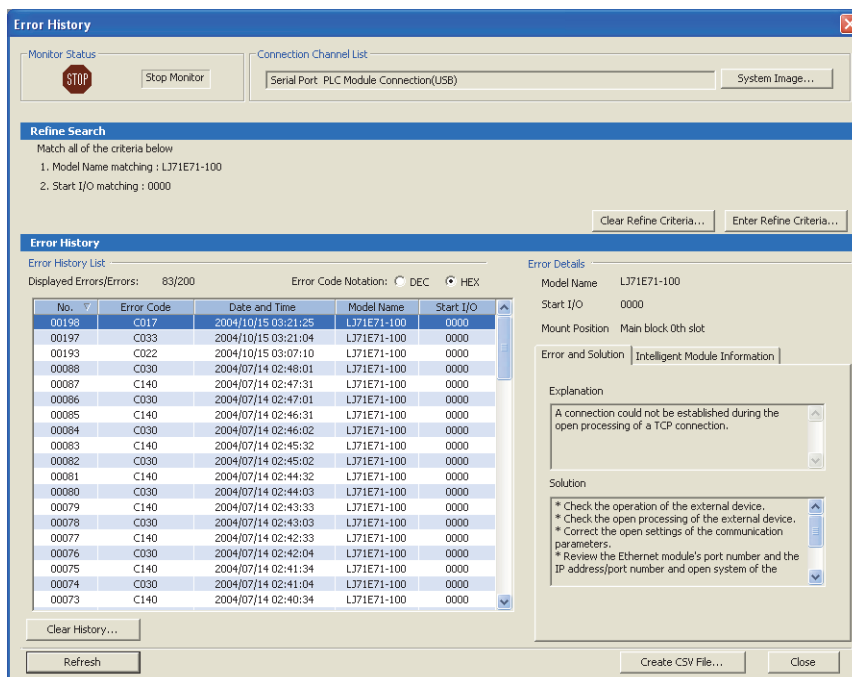
4. If the error details cannot be obtained in the step above, perform the troubleshooting below.

- Checking the LEDs (👉 Page 191, Section 14.4)
- Troubleshooting by symptom (👉 Page 194, Section 14.6)
- Ethernet Diagnostics (👉 Page 237, Section 14.8)

14.3 Checking with the Module Error Collection Function

With the module error collection function, the errors that have occurred in the E71 are saved into the CPU module. This allows error details to be held even if the power is turned off or the CPU module is reset. Check the E71's errors collected by the CPU module on the "Error History" window.

👉 [Diagnostics] ⇨ [System Monitor] ⇨ **Error History Detail** button



14.4 Checking the LEDs

The following describes how to troubleshoot the problem by checking the LEDs.

Point

The on/off status of the INIT. LED, OPEN LED, ERR. LED, and COM.ERR. LED can be also checked with a programming tool and in buffer memory.

- With a programming tool: "H/W Information" window (☞ Page 238, Section 14.9)
- In buffer memory: Area for module status (address: C8_H) (☞ Page 33, Section 3.5.2)

14.4.1 If the RUN LED turns off

The following table lists the items to be checked and the actions to be taken if the RUN LED turns off after the E71 is powered on.

Check item	Action
Is it a watchdog timer error?	Reset the CPU module and check if the RUN LED turns on. If the RUN LED still does not turn on, the E71 may be faulty. Please consult your local Mitsubishi representative.
Is the E71 connected properly?	Power off the E71, and connect the module properly.
Is the power capacity of the power supply module sufficient?	Check if the power supply module has enough power capacity.

If the action above does not solve the problem, perform the following tests on the E71 to check for hardware error.

- Hardware test (☞ Page 266, Appendix 5.2)
- Self-loopback test (☞ Page 265, Appendix 5.1)

14.4.2 If the ERR. LED or COM.ERR. LED turns on

Connect the CPU module connected to the E71 with the ERR. LED or COM.ERR. LED on to a programming tool to identify the cause. (☞ Page 189, Section 14.2, Page 237, Section 14.8)


If the action above does not solve the problem, perform the following tests on the E71 to check for hardware error.

- Hardware test (☞ Page 266, Appendix 5.2)
- Self-loopback test (☞ Page 265, Appendix 5.1)

The COM.ERR. LED does not turn off even after the error cause is removed. To turn it off, refer to "How to Turn Off the COM.ERR. LED". (☞ Page 240, Section 14.10)

14.4.3 If the SD/RD LED does not flash when data is sent

The following table lists the items to be checked and the actions to be taken if the SD LED does not flash when data is sent.

Check item	Action
Is the ERR. LED or COM.ERR. LED on?	Remove the cause that turned on the ERR. LED or COM.ERR. LED.
Are the cables properly connected?	Check the cable connection. In addition, perform a line test to check if there is a problem with the cable connection and the Ethernet line. ( Page 256, Appendix 4)
Are the parameter settings correct?	Correct the IP addresses of the own station and the communication destination, router setting, and subnet mask setting.
Is the program correct?	Correct the send program of the connected device.


If the action above does not solve the problem, perform the following tests on the E71 to check for hardware error.

- Hardware test ( Page 266, Appendix 5.2)
- Self-loopback test ( Page 265, Appendix 5.1)

14.5 Checking with a Display Unit

With the buffer memory monitor/test of a display unit, a value in the buffer memory can be checked without a programming tool, and a problem can be troubleshot.

For how to monitor buffer memory with a display unit, refer to the following.

 MELSEC-L CPU Module User's Manual (Function Explanation, Program Fundamentals)

14.6 Troubleshooting by Symptom

The following describes how to troubleshoot for each phenomenon. When an error occurs in the E71, identify the cause with the programming tool. (☞ Page 189, Section 14.2)

14.6.1 Communications cannot be performed with the connected device.

The following table lists how to troubleshoot the problem when the E71 cannot communicate with the connected device.

Check item	Action
Is the RUN LED of the E71 on?	Reset the CPU module. If resetting the CPU module does not turn on the RUN LED on the E71, the hardware of the E71 may be faulty. Replace the E71 and restart all the connected devices that were performing communications with the own station. *1
Does the connected device conform to the Ethernet standard?	Replace the device with one conforming to the Ethernet standard. (☞ Page 56, CHAPTER 5)
Is the cable securely connected?	<ul style="list-style-type: none"> Securely lock the cable. Check the wiring. (☞ Page 60, Section 6.2)
Is the network parameter mode set to "Online"?	Set the network parameter mode to "Online". (☞ Page 64, Section 7.1.2)
Does the communication data code (ASCII or binary) setting match with that of the connected device?	If the data code settings do not match, data cannot be sent or received because the connected device cannot normally decode commands. *2 Set the same data code as that of the connected device in "Communication Data Code" of the Ethernet operation setting. (☞ Page 65, Section 7.1.3)
Has the initial process been completed normally? (Is Initial normal completion signal (X19) on?)	<ul style="list-style-type: none"> If Initial normal completion signal (X19) is off, perform the initial process. (☞ Page 246, Appendix 3) If the initial setting of the programming tool is same as that of the initial process program, delete the initial process program.
Was a device on the line (such as an E71, connected device, hub, and router) replaced by the one with the same IP address?	Restart all the devices on the line. *1

*1 Devices on an Ethernet network contain the IP to Mac address mapping, called an "ARP cache". If a device on a line is replaced by another device with the same IP address, the MAC address contained in the "ARP cache" and the MAC address of the replaced device do not match; therefore, communications may not be performed normally. The "ARP cache" is updated when the device is reset or after a certain period of time. The time varies depending on the devices.

*2 If the communication data code setting is different between the E71 and the connected device, error codes that are not found in the error code list may be returned to the connected device. When receiving data with different data codes, the E71 cannot decode commands normally. The E71 returns an error response according to the communication data code setting. (☞ Page 104, Section 10.6.2 (1) (e), Remark)

*3 If the LG and FG terminals of the power supply module on the E71-connected station are not connected, the communication line may be closed (disconnected) due to noise; therefore, communications with the connected device may not be performed.

If the action above does not solve the problem, check for an error in each communication process and take corrective action. (☞ Page 196, Section 14.6.4 to Page 202, Section 14.6.10)

14.6.2 The E71 frequently fails to receive a message sent from the connected device.

The following table lists how to troubleshoot the problem when the E71 frequently fails to receive a message sent from the connected device.

Check item	Action
Has the error code C0C7 _H been stored in Error code/end code (address: E5 _H) in the error log block?	There may be a heavy load on the Ethernet line due to data sending and receiving between connected devices. <ul style="list-style-type: none"> • Taking action such as separating networks and decreasing the number of data sending reduces the load on the Ethernet line. • Consult the network administrator and reduce the load on the Ethernet line.
Has the number in Received TCP packet count (address: 1B8 _H and 1B9 _H) increased?	If the packets cannot be received even though the number of received TCP packets has increased, set the value of TCP Maximum Segment Transmission setting area (address: 1E _H) to 8000 _H . Then perform the initial process again. (☞ Page 246, Appendix 3)

If the action above does not solve the problem, perform the following tests on the E71 to check for hardware error.

- Hardware test (☞ Page 266, Appendix 5.2)
- Self-loopback test (☞ Page 265, Appendix 5.1)

14.6.3 A dedicated instruction is not completed.

The following table lists how to troubleshoot the problem when a dedicated instruction is not completed.


Check item	Action
Is the network parameter mode set to "Online"?	Set the network parameter mode to "Online". (☞ Page 64, Section 7.1.2) (If a dedicated instruction is executed in the "Offline" mode, no error occurs but the instruction is not completed.)

If the action above does not solve the problem, perform the following tests on the E71 to check for hardware error.


- Hardware test (☞ Page 266, Appendix 5.2)
- Self-loopback test (☞ Page 265, Appendix 5.1)

14.6.4 MC protocol communications cannot be performed.

The following table lists how to troubleshoot the problem when the E71 cannot perform MC protocol communications.

Check item	Action
Has the connection with the connected device been opened? (Check the corresponding bit of 5000 _H in the buffer memory.) ^{*1}	<ul style="list-style-type: none"> • Open the connection with the connected device. • Check if the open/close process for I/O signals and the OPEN/CLOSE instructions are not used together for the same connection. If so, correct the program.
Did the connected device send a command?	Send a command to the E71.
Was a response returned to the device that had sent the command?	<ul style="list-style-type: none"> • Check if the correct IP address was specified in the command. If not, correct the IP address and send the command again. • Check if the communication protocol (TCP/IP or UDP/IP) matches with that of the connected device. If not, correct it.
Does the communication protocol (TCP/IP or UDP/IP) match with that of the connected device?	Use the same communication protocol (TCP/IP or UDP/IP) as that of the connected device.
Does the communication data code (ASCII or binary) setting match with that of the connected device?	<p>If the data code settings do not match, data cannot be sent or received because the connected device cannot normally decode commands.^{*2} Use the same setting of "Communication Data Code" in the Ethernet operation setting as that of the connected device.</p> <p>( Page 65, Section 7.1.3)</p>
Is the end code of the response 0?	Check the end and error codes to correct the error.
Is the correct command format used for the command type, device, address, and others?	Correct the command format.
Is the write operation during RUN enabled?	Check the checkbox next to "Enable Online Change" in the Ethernet operation setting.
Has an open or initial error occurred in the error log area?	Check and correct the error.

*1 If the connection of only the connected device is closed due to cable disconnection, personal computer restart, or other reasons, reopen the connection using the same port used before the error occurred. The E71 does not close a connection if it receives an Active open request again from the connected device with a different IP address or a port number.

*2 If the communication data code setting is different between the E71 and the connected device, error codes that are not found in the error code list may be returned to the connected device. When receiving data with different data codes, the E71 cannot decode commands normally. The E71 returns an error response according to the communication data code setting. ( Page 104, Section 10.6.2 (1) (e), Remarks)

If the action above does not solve the problem, perform the following tests on the E71 to check for hardware error.

- Hardware test ( Page 266, Appendix 5.2)
- Self-loopback test ( Page 265, Appendix 5.1)

14.6.5 Data cannot be sent with communications using a fixed buffer.

The following table lists how to troubleshoot the problem when an E71 cannot send data with communications using a fixed buffer.

Check item	Action
Has the connection with the connected device been opened? (Check the corresponding bit of 5000 _H in the buffer memory.)	<ul style="list-style-type: none"> • Open the connection with the connected device. • Check if the open/close processes for I/O signals and the OPEN/CLOSE instructions are not used together for the same connection. If so, correct the program. • Read Open error code (address: 124 (7C_H)) in the communication status storage area. Then check the error details to correct the error.
Is the IP address setting of the connected device correct?	Check and correct the IP address of the connected device.
Does the communication protocol (TCP/IP or UDP/IP) match with that of the connected device?	Use the same communication protocol (TCP/IP or UDP/IP) as that of the connected device.
Was the BUFSND instruction executed?	<ul style="list-style-type: none"> • Correct the execution conditions of the BUFSND instruction. • Check if the send/receive process for I/O signals and the BUFSND/BUFRCV instructions are used together for the same connection. If so, correct the program.
Was the BUFSND instruction completed normally?	Check the error code in the BUFSND instruction completion status area to correct the error.
Has the data length of the send data been set in the control data?	Write the data length.
Is the communication status of the connected device normal?	Correct the error in the connected device.
Has an open or initial error occurred in the error log area?	Check and correct the error.

If the action above does not solve the problem, perform the following tests on the E71 to check for hardware error.

- Hardware test (☞ Page 266, Appendix 5.2)
- Self-loopback test (☞ Page 265, Appendix 5.1)

14.6.6 Data cannot be received with communications using a fixed buffer.

The following table lists how to troubleshoot the problem when an E71 cannot receive data with communications using a fixed buffer.

Check item	Action
Does the receive data length match with the amount of data actually received?	If the amount of the actually received data is smaller than the receive data length, receive the remaining data. If it is larger than the receive data length, receive the excess data as the next request message or discard it.
Has the connection with the connected device been opened? (Check the corresponding bit of 5000 _H in the buffer memory.)	<ul style="list-style-type: none"> • Open the connection with the connected device. • Check if the open/close processes for I/O signals and the OPEN/CLOSE instructions are not used together for the same connection. If so, correct the program. • Read Open error code (address: 124 (7C_H)) in the communication status storage area. Then check the error details to correct the error.
Is the IP address setting of the connected device correct?	Check and correct the IP address of the connected device.
Does the communication protocol (TCP/IP or UDP/IP) match with that of the connected device?	Use the same communication protocol (TCP/IP or UDP/IP) as that of the connected device.
Is the reception completion signal of the corresponding fixed buffer on? (Check the corresponding bit of 5005 _H in the buffer memory.)	Data was not sent from the connected device. Check the connected device on the sending side and correct the error.
Was the BUFRCV instruction executed?	<ul style="list-style-type: none"> • Correct the execution conditions of the BUFRCV instruction. • Check if the send/receive processes for I/O signals and the BUFSND/BUFRCV instructions are not used together for the same connection. If so, correct the program.
Was the BUFRCV instruction completed normally?	Check the error code in the BUFSND instruction completion status area to correct the error.
Was the BUFRCVS instruction executed?	<ul style="list-style-type: none"> • Correct the interrupt setting of the programming tool. • Check if the send/receive processes for I/O signals and the BUFSND/BUFRCV instructions are not used for the same connection. If so, correct the program.
Is the control data of the BUFRCVS instruction correct?	Correct the control data.
Has an open or initial error occurred in the error log area?	Check and correct the error.

If the action above does not solve the problem, perform the following tests on the E71 to check for hardware error.

- Hardware test (☞ Page 266, Appendix 5.2)
- Self-loopback test (☞ Page 265, Appendix 5.1)

14.6.7 Communications using a random access buffer cannot be performed.

The following table lists how to troubleshoot the problem when an E71 cannot perform communications using a random access buffer.

Check item	Action
Has the connection with the connected device been opened? (Check the corresponding bit of 5000 _H in the buffer memory.)	<ul style="list-style-type: none"> • Open the connection with the connected device. • Check if the open/close process for I/O signals and the OPEN/CLOSE instructions are used together for the same connection. If so, correct the program.
Did the connected device send a command?	Send a command to the E71.
Was a response returned to the device that had sent the command?	<ul style="list-style-type: none"> • Check if the correct IP address was specified in the command. If not, correct the IP address and send the command again. • Check if the communication protocol (TCP/IP or UDP/IP) matches with that of the connected device. If not, correct it.
Is the end code of the response 0?	Check the end and error codes to correct the error.
Is the correct buffer memory address specified for the command?	Correct the buffer memory address and send the command again.
Has data been set in the specified address of the random access buffer?	Write the data.
Has the write data been set in the connected device?	Set the data.
Has an open or initial error occurred in the error log area?	Check and correct the error.

If the action above does not solve the problem, perform the following tests on the E71 to check for hardware error.

- Hardware test (☞ Page 266, Appendix 5.2)
- Self-loopback test (☞ Page 265, Appendix 5.1)

14.6.8 An e-mail cannot be sent.

The following table lists how to troubleshoot the problem when an E71 cannot send an e-mail.

(1) When sending an e-mail with a program

Check item	Action
Was the MSEND instruction executed?	Correct the execution conditions of the MSEND instruction.
Was the MSEND instruction completed normally?	Check the error code in the MSEND instruction completion status area and the e-mail sending error log to correct the error.
Is the error response received from the mail server using the MRECV instruction?	Check with the system administrator that the following settings are correct: <ul style="list-style-type: none"> • DNS setting • E-mail setting

If the action above does not solve the problem, check the following.

- Is the e-mail parameter setting of the E71 correct?
- Is the mail server system down?
- Is there any error in the communication path?

(2) When sending an e-mail with the programmable controller CPU monitoring function

Check item	Action
Are the monitoring conditions satisfied?	Check the notification setting with the programming tool.
Was the first sending performed using the programmable controller CPU monitoring function?	Check if there is any device outside the range of the device settings among the condition devices and correct the condition device.
Were the second and subsequent sendings performed using the programmable controller CPU monitoring function?	<ul style="list-style-type: none"> • Correct the following so that the monitoring condition disable time becomes longer than the value set in "PLC Inquiry Interval". <ul style="list-style-type: none"> • PLC Inquiry Interval • Condition Device • Correct the following so that the sending time of the SMTP server and the E71 becomes shorter than the value set in "PLC Inquiry Interval". <ul style="list-style-type: none"> • Reduce the load on the SMTP server. • When the SMTP server and the E71 are connected via routers or other devices, connect the E71 to the same main line as the SMTP server.

If the action above does not solve the problem, check the following.

- Is the e-mail parameter setting of the E71 correct?
- Is the mail server system down?
- Is there any error in the communication path?


14.6.9 An e-mail cannot be received.

The following table lists how to troubleshoot the problem when an E71 cannot receive an e-mail.

Check item	Action
Was the MRECV instruction executed?	Correct the execution conditions of the MSEND instruction.
Was the MRECV instruction completed normally?	Check the error code in the MRECV instruction completion status area and correct the error.
Does the receive data exist in the device?	<ul style="list-style-type: none"> • Shorten the inquiry interval. • Check if there is any error in the communication path. • Check the number of e-mails remaining on the mail server.
Is the receive data value incorrect? (Are the characters in the subject garbled?)	<ul style="list-style-type: none"> • Check and correct the e-mail parameter settings. • Check if the mail server system is down. • Check if there is any error in the communication path.
Is a device other than the E71 sending the data?	<p>If EUC or SHIFT-JIS is used, perform the following:</p> <ul style="list-style-type: none"> • Send an e-mail from a personal computer to the own station. • Check the header information of the received e-mail sent by the own station. <p>For Outlook Express, select "Properties" → "Details". In the header information, check that the setting is "Content-Type:text-plain;charset=iso-2022jp". If it is "charset=EUC-jp" or "charset=SHIFT-JIS", correct it.</p>
Is the E71 sending the data?	Because the E71 does not decode received data, it cannot receive an e-mail normally. Try to send an e-mail from a device other than the E71.
Did the sending device send the e-mail in the ASCII code or convert the ASCII code further into the ASCII code?	<p>Check the following.</p> <ul style="list-style-type: none"> • Convert the e-mail attachment into the binary code and send it. • Check if the sent data (e-mail attachment) is not in the ASCII code. • When the destination is a personal computer, use another mailer. If the received e-mails differ depending on the mailer used, check the mailer settings. • Check if the encoding and decoding methods of the mail server differs from those of Ethernet.

14.6.10 Communications using data link instructions cannot be performed.

The following table lists how to troubleshoot the problem when an E71 cannot perform communications using data link instructions.

Check item	Action
Is the communication status of the connected device normal?	Correct the error of the connected device.
Is the number of resends set upon every instruction execution?	Modify the program to set the number of resends upon every instruction execution.
Does an error occur even after the arrival monitoring time is increased?	Check and correct the arrival monitoring time.
Does an error occur even after the setting value in "TCP Resend Timer" in the initial setting is increased?	Check and correct the setting value in "TCP Resend Timer".
Are the routing parameter settings correct?	Check the routing parameters and correct the error.
Is the Station No. <->IP information setting configured?	Set the "Station No.<->IP Information" setting in the network parameter window. ( MELSEC-Q/L Ethernet Interface Module User's Manual (Application))


If the action above does not solve the problem, perform the following tests on the E71 to check for hardware error.

- Hardware test ( Page 266, Appendix 5.2)
- Self-loopback test ( Page 265, Appendix 5.1)

14.7 Error Code List

This section describes the error codes (abnormal codes) of the errors that may occur in each process for performing communications between the E71 and a connected device as well as errors caused by process requests issued from the CPU module of the own station, the error details, and the actions to be taken.

The following table lists the types of errors that may occur.

Error type	Description	Buffer memory area that stores the error code
Errors occurring in the initial process	<ul style="list-style-type: none"> Setting value error Initial process error 	Page 204, Section 14.7 (1)
Errors occurring in the open process	<ul style="list-style-type: none"> Setting value error Open process error 	Page 204, Section 14.7 (2)
Errors occurring in communications using a fixed buffer (sending) with the connected device	<ul style="list-style-type: none"> Specified data error Sending error 	Page 204, Section 14.7 (3), Page 204, Section 14.7 (4)
Errors occurring in communications using a fixed buffer with the connected device	<ul style="list-style-type: none"> Specified data error Communication error (excluding errors occurring in communications using a fixed buffer (sending) with the connected device) 	Page 204, Section 14.7 (4)
Errors returned to the connected device when communicating with it	<ul style="list-style-type: none"> Errors returned in communications using a fixed buffer (end code) Errors returned in communications using a random access buffer (end code) 	-
	Errors returned in communications using the MC protocol	End code or error code for each command
Errors occurring while communicating with the destination (including the causes in the "Description" column), and whose error codes are stored in the error log area	<ul style="list-style-type: none"> Specified data error Errors where the source cannot be found Errors that have occurred in MC protocol communications Errors that have occurred in communications using a random access buffer 	Page 205, Section 14.7 (5)
Errors (response commands) occurring in communications with the destination using the file transfer (FTP server) function	<ul style="list-style-type: none"> Specified data error Communication error 	 MELSEC-L CPU Module User's Manual (Built-In Ethernet Function)
Errors occurring in communications using the Web function	Communication error	Page 206, Section 14.7 (6)
Errors occurring when an e-mail is received	<ul style="list-style-type: none"> Setting data error Receiving error 	Page 207, Section 14.7 (7) (a)
Errors occurring when an e-mail is sent	<ul style="list-style-type: none"> Setting data error Sending error 	Page 209, Section 14.7 (7) (b)
Errors occurring in communications using data link instructions from the CPU module of the own station	<ul style="list-style-type: none"> Specified data error Communication error 	Not stored (stored in the complete status area of the instructions)
Errors occurring in communications using the dedicated instructions from the CPU module of the own station	<ul style="list-style-type: none"> Specified data error Communication data error 	Not stored (stored in the complete status area of the instructions)

(1) Initial error code (address: 69_H)

The error codes generated when the initial process is executed are stored. Error codes are stored in binary when Initial abnormal completion signal (X1A) is on. The error codes are cleared when Initial normal completion signal (X19) turns on, but can be also cleared by the following processes:

- Resetting or powering off the CPU module
- Writing 0 to the initial error code storage area using a program

(2) Open error code (address: 7C_H to C1_H, 5824_H to 5869_H)

The result of the open process for connection with the connected device is stored in binary.

- 0: Normal completion
- Values other than 0: Abnormal end (Open abnormal detection signal (X18): ON)

The error codes are cleared by the following operations:

- Reopening the connection that caused an open error
- Powering off and on or resetting the CPU module

(3) Fixed buffer sending error code (address: 7D_H to C2_H, 5825_H to 586A_H)

The error codes generated when an error has occurred in data sending to the destination during communications using a fixed buffer in the connection with the connected device are stored. A sending error code is cleared when the next data sending is completed normally.

(4) Connection end code (address: 7E_H to C3_H, 5826_H to 586B_H)

The codes returned in the responses from the destination during communications using a fixed buffer in the connection with the connected device are stored in binary. Process the end codes in the responses according to the destination.

(5) Error log area (address: E0_H to 1FF_H)

The following table lists the details of the error log area. The buffer memory addresses of the error log block are those of the error log block 1. For the addresses of the error log block 2 and later, refer to the list of buffer memory addresses. (☞ Page 33, Section 3.5.2)

Buffer memory name	Address	Description																	
Number of errors	E3 _H	The number of errors registered in the error log block area is stored. When the number of errors reaches 65536, the count stops at FFFF _H (65535). ^{*1}																	
Error log write pointer	E4 _H	The number of the error log block where the latest error log is registered is stored. <ul style="list-style-type: none"> • 0: No error (no registration of error log) • 1 or more: The number of the error log block where the latest error log is registered. If the pointer value is 16, it means that the latest error log is registered in the error log block 16 area. When the number of errors reaches 17, the error log is then registered in the error log block 1 area again. ^{*2}																	
Error log block	Error code/end code	E5 _H	An error code indicating error details is stored. The subheader code of the error message is stored in bits 0 to 7 of the corresponding area. (Bits 8 to 15 store 0.) For errors below the TCP/IP or UDP/IP level, 0 is stored.																
	Command code	E7 _H	The command code of the error message or the values of the lower bytes of the request type and subrequest type of the data link instructions are stored. <div style="text-align: center;"> <table border="1" style="display: inline-table; margin-right: 20px;"> <tr> <td style="text-align: center;">b15</td> <td style="text-align: center;">to</td> <td style="text-align: center;">b0</td> </tr> <tr> <td colspan="3" style="text-align: center;">Command code</td> </tr> </table> or <table border="1" style="display: inline-table;"> <tr> <td style="text-align: center;">b15</td> <td style="text-align: center;">to</td> <td style="text-align: center;">b8</td> <td style="text-align: center;">b7</td> <td style="text-align: center;">to</td> <td style="text-align: center;">b0</td> </tr> <tr> <td colspan="2" style="text-align: center;">Subrequest type</td> <td colspan="2" style="text-align: center;">Request type</td> </tr> </table> </div> 0 is stored in the following cases: <ul style="list-style-type: none"> • For messages not containing a command code • For errors below the TCP/IP or UDP/IP level (because their commands are unknown) 	b15	to	b0	Command code			b15	to	b8	b7	to	b0	Subrequest type		Request type	
	b15	to	b0																
	Command code																		
	b15	to	b8	b7	to	b0													
	Subrequest type		Request type																
Connection No.	E8 _H	The number of the error connection is stored in bits 0 to 7 of the corresponding area. (Bits 8 to 15 store 0.) For errors below the TCP/IP or UDP/IP level, 0 is stored.																	
Own station port No.	E9 _H	The port number of the error own station is stored. For errors below the TCP/IP or UDP/IP level, 0 is stored.																	
Destination IP address	EA _H and EB _H	The IP address of the error connected device is stored. 0 is stored in the following cases: <ul style="list-style-type: none"> • For errors below the IP level • When an error response was sent through the CPU module 																	
Destination Port No.	EC _H	The IP address of the error connected device is stored. For errors below the TCP/IP or UDP/IP level, 0 is stored.																	
Status for each protocol	178 _H to 1FF _H	The number of occurrences of the corresponding status of each protocol is stored. When the count value by the E71 exceeds two words, the count stops at FFFFFFFF _H (4294967295).																	

- *1 The error information is continued to be stored into the following areas even if the count of the errors is stopped:
- Error log write pointer storage area
 - Error log block

- *2 An error log block area consists of 16 error log blocks that have the same data order.

Point

Values stored in the buffer memory are cleared when an E71-connected station is powered on or reset. (They are not cleared during the initial process.) Although this area does not normally need to be read, read it when necessary for maintenance or other purposes.

(6) HTTP status storage area (address: 5101_H to 5177_H)


The following table lists the HTTP status storage areas. The buffer memory addresses of the error log block are those of the error log block 1. For the addresses of the error log block 2 and later, refer to the list of buffer memory addresses. (☞ Page 33, Section 3.5.2)

Buffer memory name	Address	Description																																								
Error log pointer	5101 _H	The number of the error log block where the latest error log is registered is stored. <ul style="list-style-type: none"> • 0: No error (no registration of error log) • 1 or more: The number of the error log block where the latest error log is registered. If the pointer value is 16, it means that the latest error log is registered in the error log block 16 area. When the number of errors reaches 17, the error log is then registered in the error log block 1 area again.*1																																								
Log counter	5102 _H to 5106 _H	The number of times the HTTP response code was returned from the E71 to the Web browser is stored.																																								
Error log block	HTTP response code	5108 _H	The HTTP response code in case of an error is stored.																																							
	Destination IP address	5109 _H to 510A _H	The IP address of the server in case of an error is stored.																																							
	Error time	510B _H to 510E _H	The time when the error occurred is stored in the BCD code. <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="text-align: center;">b15</td> <td style="text-align: center;">to</td> <td style="text-align: center;">b8 b7</td> <td style="text-align: center;">to</td> <td style="text-align: center;">b0</td> </tr> <tr> <td colspan="2" style="text-align: center;">Month (01_H to 12_H)</td> <td colspan="3" style="text-align: center;">Year (00_H to 99_H): Last two digits of the year</td> </tr> <tr> <td style="text-align: center;">b15</td> <td style="text-align: center;">to</td> <td style="text-align: center;">b8 b7</td> <td style="text-align: center;">to</td> <td style="text-align: center;">b0</td> </tr> <tr> <td colspan="2" style="text-align: center;">Hour (00_H to 23_H)</td> <td colspan="3" style="text-align: center;">Day (01_H to 31_H)</td> </tr> <tr> <td style="text-align: center;">b15</td> <td style="text-align: center;">to</td> <td style="text-align: center;">b8 b7</td> <td style="text-align: center;">to</td> <td style="text-align: center;">b0</td> </tr> <tr> <td colspan="2" style="text-align: center;">Second (00_H to 59_H)</td> <td colspan="3" style="text-align: center;">Minute (00_H to 59_H)</td> </tr> <tr> <td style="text-align: center;">b15</td> <td style="text-align: center;">to</td> <td style="text-align: center;">b8 b7</td> <td style="text-align: center;">to</td> <td style="text-align: center;">b0</td> </tr> <tr> <td colspan="2" style="text-align: center;">Year (00_H to 99_H): First two digits of the year</td> <td colspan="3" style="text-align: center;">Day of the week (0 to 6)</td> </tr> </table>	b15	to	b8 b7	to	b0	Month (01 _H to 12 _H)		Year (00 _H to 99 _H): Last two digits of the year			b15	to	b8 b7	to	b0	Hour (00 _H to 23 _H)		Day (01 _H to 31 _H)			b15	to	b8 b7	to	b0	Second (00 _H to 59 _H)		Minute (00 _H to 59 _H)			b15	to	b8 b7	to	b0	Year (00 _H to 99 _H): First two digits of the year		Day of the week (0 to 6)	
b15	to	b8 b7	to	b0																																						
Month (01 _H to 12 _H)		Year (00 _H to 99 _H): Last two digits of the year																																								
b15	to	b8 b7	to	b0																																						
Hour (00 _H to 23 _H)		Day (01 _H to 31 _H)																																								
b15	to	b8 b7	to	b0																																						
Second (00 _H to 59 _H)		Minute (00 _H to 59 _H)																																								
b15	to	b8 b7	to	b0																																						
Year (00 _H to 99 _H): First two digits of the year		Day of the week (0 to 6)																																								

*1 An error log block area consists of 16 error log blocks that have the same data order.

(7) E-mail status storage area (address: 5870_H to 5FFF_H)

When the stored count exceeds FFFF_H, the count starts from 0_H again. The following table lists the details of the e-mail status storage area. The buffer memory addresses of the error log block are those of the error log block 1. For the addresses of the error log block 2 and later, refer to the list of buffer memory addresses.

( Page 33, Section 3.5.2)

(a) Receive area

Buffer memory name	Address	Description
Number of mails remaining on the server	5870 _H	The number of remaining e-mails returned when the E71 inquired the receive mail server is stored. <ul style="list-style-type: none"> • 0: No received mail in the server • 1 to 15: The number of mails remaining in the server • 16: The number of mails in the server is 16 or more.
Dedicated instruction normal completion count	5871 _H	The cumulative number of normal completions of the dedicated instruction (MRECV) is stored. <ul style="list-style-type: none"> • 0: The MRECV instruction has not been executed or no executions have resulted in normal completion. • 1 or more: The cumulative number of normal completions of the MRECV instruction
Dedicated instruction abnormal completion count	5872 _H	The cumulative number of abnormal ends of the dedicated instruction (MRECV) is stored. <ul style="list-style-type: none"> • 0: The MRECV instruction has not been executed or no executions have resulted in abnormal end. • 1 or more: The cumulative number of abnormal ends of the MRECV instruction
Normal receiving count	5873 _H	The cumulative number of received e-mails forwarded by the E71 to the mail buffer data area is stored. <ul style="list-style-type: none"> • 0: No mail has been transferred. • 1 or more: The number of normal mail transfer completions
Attached file receiving count	5874 _H	The cumulative number of e-mails with an attachment received by the E71 is stored. <ul style="list-style-type: none"> • 0: No e-mail with an attachment has been received. • 1 or more: The number of normal reception of e-mails with an attachment
Server inquiry count	5875 _H	The cumulative number of inquiries to the receive mail server according to the parameter setting is stored. <ul style="list-style-type: none"> • 0: No inquiry has been made to the server. • 1 or more: The cumulative number of inquiries to the server
Server communication error count	5876 _H	The cumulative number of communication errors returned in response to inquiries to the receive mail server is stored. <ul style="list-style-type: none"> • 0: No communication error between servers or no inquiry • 1 or more: The cumulative number of communication errors
Error log write count	5877 _H	The cumulative number of registrations in the receiving error log block area is stored. <ul style="list-style-type: none"> • 0: No error or no inquiry to the server • 1 or more: The cumulative number of writes to the error log block area
Receiving error log write pointer	5878 _H	The number of the error log block where the latest receiving error log is registered is stored. <ul style="list-style-type: none"> • 0: No error (no registration of error log) • 1 or more: The number of the error log block where the latest error log is registered. <p>If the pointer value is 16, it means that the latest error log is registered in the receiving error log block 16 area. When the number of errors reaches 17, the error log is then registered in the error log block 1 area again.*1</p>

Buffer memory name	Address	Description																																								
Error log block	Error code	5879 _H An error code indicating the error details is stored.																																								
	Command code	587A _H The system command code of the error message is stored.																																								
	From	587B _H The first eight words of the sender's e-mail address of the error e-mail in the communications with the mail server are stored in the ASCII code. (Example) If the sender's e-mail address is "use@from.add.sample.co.jp", "use@from.add.sam" is stored in the ASCII code.																																								
	Date	5883 _H The date and time when the e-mail was received are stored in the BCD code. <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 25%; text-align: center;">b15</td> <td style="width: 25%; text-align: center;">to</td> <td style="width: 25%; text-align: center;">b8 b7</td> <td style="width: 25%; text-align: center;">to</td> <td style="width: 25%; text-align: center;">b0</td> </tr> <tr> <td colspan="2" style="text-align: center;">Month (01_H to 12_H)</td> <td colspan="3" style="text-align: center;">Year (00_H to 99_H): Last two digits of the year</td> </tr> <tr> <td style="text-align: center;">b15</td> <td style="text-align: center;">to</td> <td style="text-align: center;">b8 b7</td> <td style="text-align: center;">to</td> <td style="text-align: center;">b0</td> </tr> <tr> <td colspan="2" style="text-align: center;">Hour (00_H to 23_H)</td> <td colspan="3" style="text-align: center;">Day (01_H to 31_H)</td> </tr> <tr> <td style="text-align: center;">b15</td> <td style="text-align: center;">to</td> <td style="text-align: center;">b8 b7</td> <td style="text-align: center;">to</td> <td style="text-align: center;">b0</td> </tr> <tr> <td colspan="2" style="text-align: center;">Second (00_H to 59_H)</td> <td colspan="3" style="text-align: center;">Minute (00_H to 59_H)</td> </tr> <tr> <td style="text-align: center;">b15</td> <td style="text-align: center;">to</td> <td style="text-align: center;">b8 b7</td> <td style="text-align: center;">to</td> <td style="text-align: center;">b0</td> </tr> <tr> <td colspan="2" style="text-align: center;">Year (00_H to 99_H): First two digits of the year</td> <td colspan="3" style="text-align: center;">Day of the week (0 to 6)</td> </tr> </table>	b15	to	b8 b7	to	b0	Month (01 _H to 12 _H)		Year (00 _H to 99 _H): Last two digits of the year			b15	to	b8 b7	to	b0	Hour (00 _H to 23 _H)		Day (01 _H to 31 _H)			b15	to	b8 b7	to	b0	Second (00 _H to 59 _H)		Minute (00 _H to 59 _H)			b15	to	b8 b7	to	b0	Year (00 _H to 99 _H): First two digits of the year		Day of the week (0 to 6)		
	b15	to	b8 b7	to	b0																																					
Month (01 _H to 12 _H)		Year (00 _H to 99 _H): Last two digits of the year																																								
b15	to	b8 b7	to	b0																																						
Hour (00 _H to 23 _H)		Day (01 _H to 31 _H)																																								
b15	to	b8 b7	to	b0																																						
Second (00 _H to 59 _H)		Minute (00 _H to 59 _H)																																								
b15	to	b8 b7	to	b0																																						
Year (00 _H to 99 _H): First two digits of the year		Day of the week (0 to 6)																																								
Subject	5887 _H The first 30 words of the e-mail subject are stored. A subject is not stored successfully if it contains characters other than alphanumeric and ASCII code characters.																																									

*1 A receiving error log block area consists of 16 error log blocks that have the same data order.

(b) Send area

Buffer memory name	Address	Description	
Dedicated instruction normal completion count	5B39 _H	The cumulative number of normal completions of the dedicated instruction (MSEND) is stored. • 0: The MSEND instruction has not been executed or no executions have resulted in normal completion. • 1 or more: The cumulative number of normal completions of the MSEND instruction	
Dedicated instruction abnormal completion count	5B3A _H	The cumulative number of abnormal ends of the dedicated instruction (MSEND) is stored. • 0: The MSEND instruction has not been executed or no executions have resulted in abnormal end. • 1 or more: The cumulative number of abnormal ends of the MSEND instruction	
Number of mails normally completed	5B3B _H	The cumulative number of sent e-mails forwarded by the E71 to the send mail server is stored. • 0: No mail has been sent. • 1 or more: The number of normal completions of mail sending	
Attached file sending count	5B3C _H	The cumulative number of e-mails with an attachment sent by the E71 is stored. • 0: No e-mail with an attachment has been sent. • 1 or more: The number of normal sending of e-mails with an attachment	
Sending to the server count	5B3D _H	The cumulative number of e-mails sent by the E71 to the send mail server is stored. • 0: No e-mail has been sent to the server. • 1 or more: The cumulative number of sending e-mails to the server	
Number of mails abnormally completed	5B3E _H	The cumulative number of communication errors returned in response to send requests to the send mail server is stored. • 0: No communication error between servers or no e-mail sent • 1 or more: The cumulative number of communication errors	
Error log write count	5B3F _H	The cumulative number of registrations in the send error log block area is stored. • 0: No error or no inquiry to the server • 1 or more: The cumulative number of writes to the error log block area	
Sending error log write pointer	5B40 _H	The number of the error log block where the latest send error log is registered is stored. • 0: No error (no registration of sending error log) • 1 or more: The number of the error log block where the latest sending error log is registered. If the pointer value is 8, it means that the latest error log is registered in the receiving error log block 8 area. When the number of receiving errors reaches nine, the error log is then registered in the error log block 1 area again.*1	
Error log block	Error code	5B41 _H	An error code indicating the error details is stored.
	Command code	5B42 _H	The system command code of the error message is stored.
	To	5B43 _H	The first eight words of the receiver's e-mail address of the error e-mail in the communications with the mail server are stored in the ASCII code. (Example) If the sender's e-mail address is "use@from.add.sample.co.jp", "use@from.add.sam" is stored in the ASCII code.
	Date	5B4B _H	The date and time when the e-mail was sent are stored in the BCD code (similar to 5883 _H of (a)).
	Subject	5B4F _H	The first 15 words of the e-mail subject are stored. A subject is not stored successfully if it contains characters other than alphanumeric and ASCII code characters.

*1 A sending error log block area consists of 16 error log blocks that have the same data order.

14.7.1 End codes returned to a connected device during data communications

The following table lists the error codes stored in the end code appended to a response during data communications.

○: Stored in the end code appended to a response

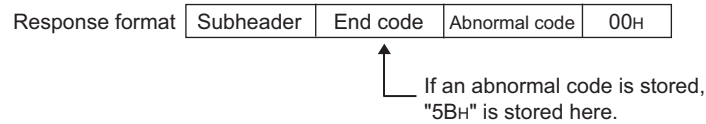
Error code	Error description	Action	Data communication type		
			MC protocol communications	Communications using a fixed buffer	Communications using a random access buffer
00 _H	<ul style="list-style-type: none"> Normal completion 	<ul style="list-style-type: none"> When each communication is normally completed, the error code 00_H is stored. 	○	○	○
02 _H	<ul style="list-style-type: none"> The device range destination of devices to be read/written from/to is incorrect. 	<ul style="list-style-type: none"> Check and correct the specified start device and the number of points. 	○		
50 _H	<ul style="list-style-type: none"> Codes for the command/response type of a subheader are not within the specifications. MC protocol communications: 00_H to 3C_H Communications using a fixed buffer: 60_H Communications using a random access buffer: 61_H, 62_H In communications using a fixed buffer, if the data length setting is smaller than the actual data amount, the remaining data is processed as the second data. In this case, a subheader undefined command type error may occur. 	<ul style="list-style-type: none"> Check and correct the command/response type set for the connected device. (Because the E71 automatically adds the command/response type, the user setting is not required.) Check and correct the data length. 	○	○	○
51 _H	<ul style="list-style-type: none"> In communications using a random access buffer, the start address specified by a connected device has been set outside the range from 0 to 6143. 	<ul style="list-style-type: none"> Check and correct the specified start address. 			○
52 _H	<ul style="list-style-type: none"> In communications using a random access buffer, the start address and the number of data words (depending on the setting when reading data) specified by a connected device exceeds the range from 0 to 6143. Data by the number of words specified (text) cannot be sent in one frame. (The data length value or the amount of the text sent/received are not in the allowable range.) 	<ul style="list-style-type: none"> Check and correct the start address and the number of data words. Correct the number of read/write points. 		○	
54 _H	<ul style="list-style-type: none"> When "ASCII Code" has been selected in the communication data code setting in the Ethernet operation setting, ASCII code data which cannot be converted into binary code data has been sent from the connected device. 	<ul style="list-style-type: none"> Check and correct the data sent from the connected device. 	○	○	○
55 _H	<ul style="list-style-type: none"> When "Enable Online Change" was disabled (or not selected) in the Ethernet operation setting, the connected device requested a data write while the CPU module was running. While the CPU module was running, the connected device requested for writing a parameter, program, or microcomputer program. 	<ul style="list-style-type: none"> Enable (select) "Enable Online Change" in the Ethernet operation setting and write data. (However, a parameter, program, or microcomputer program cannot be written while the CPU module is running.) Set the switch on the CPU module to STOP and write the data. 	○		
56 _H	<ul style="list-style-type: none"> The connected device specified a wrong device. 	<ul style="list-style-type: none"> Correct the device specification. 	○		

Error code	Error description	Action	Data communication type		
			MC protocol communications	Communications using a fixed buffer	Communications using a random access buffer
57 _H	<ul style="list-style-type: none"> The number of points for a command specified by a connected device exceeds the maximum number of processing points (number of processes that can be executed per communication) for each process. Addresses from the start address (start device number and start step number) to the specified number of points exceed the largest addresses (device number and step number) for each process. 	<ul style="list-style-type: none"> Correct the specified points or the start address (device number and step number). 	○		
	<ul style="list-style-type: none"> The byte length of a command is not within the range defined by the specifications. When data is written, the set number of write data points is different from the specified number of points. 	<ul style="list-style-type: none"> Check the data length of the command and set the data again. 	○		
	<ul style="list-style-type: none"> A monitor request was issued even though monitoring data is not registered. 	<ul style="list-style-type: none"> Register the monitoring data. 	○		
	<ul style="list-style-type: none"> For data read/write in a microcomputer program, an address after the last address that can be set in the parameter setting has been specified. 	<ul style="list-style-type: none"> Data cannot be read from/written to an area with an address after the last address. Correct the specified address. 	○		
	<ul style="list-style-type: none"> In the block number specification of the extension file register, a block number exceeding the range of the corresponding memory cassette size has been specified. 	<ul style="list-style-type: none"> Correct the block number. 	○		
58 _H	<ul style="list-style-type: none"> The start address (start device number and start step number) of a command specified by a connected device has been set outside the range that can be specified. For data read/write in a microcomputer program or file register, values exceeding the CPU module parameter setting range has been specified. 	<ul style="list-style-type: none"> Correct the value to the one within the range that can be specified for each process. 	○		
	<ul style="list-style-type: none"> A block number specified for an extension file register does not exist. 	<ul style="list-style-type: none"> Correct the block number. 	○		
	<ul style="list-style-type: none"> A file register cannot be specified. 	<ul style="list-style-type: none"> Correct the device specification. 	○		
	<ul style="list-style-type: none"> A word device has been specified for the command for bit devices. The start number of bit devices has been specified using a value other than a multiple of 16 in the command for word devices. 	<ul style="list-style-type: none"> Correct the command or specified device. 	○		
59 _H	<ul style="list-style-type: none"> An extension file register cannot be specified. 	<ul style="list-style-type: none"> Correct the device specification. 	○		
5B _H	<ul style="list-style-type: none"> The CPU module and the E71 cannot communicate. The CPU module cannot process a request from a connected device. 	<ul style="list-style-type: none"> Check the error code appended to the end code and correct the error. 	○		
60 _H	<ul style="list-style-type: none"> The communication time between the CPU module and the E71 exceeded the CPU monitoring timer value. 	<ul style="list-style-type: none"> Increase the CPU monitoring timer value. 	○		

Error code	Error description	Action	Data communication type		
			MC protocol communications	Communications using a fixed buffer	Communications using a random access buffer
63 _H	<ul style="list-style-type: none"> In communications using a fixed buffer, the remote password of the port for the destination E71 is in the lock status. 	<ul style="list-style-type: none"> After unlocking the remote password using the MC protocol, perform communications using a fixed buffer. Exclude ports used for communications using a fixed buffer from the targets of the remote password check. 		○	○
A0 _H to FFFF _H	<ul style="list-style-type: none"> The error details and how to troubleshoot the problem are the same as those for the error code stored in the buffer memory. (☞ Page 214, Section 14.7.3) 				

14.7.2 Abnormal codes returned during communications using an A-compatible 1E frame


The following table lists the error codes stored in the abnormal code appended to a response in the A-compatible 1E frame in MC protocol communications. (An abnormal code is appended only when an end code is 5B_H.)




Error code	Error	Error description	Action
10 _H	PC number error	The PC number specified with a command is not FF of the own station or not the station number set with the MELSECNET link parameters.	Change the PC number to FF _H of the own station or not the station number set using the link parameter to perform communications again.
11 _H	Mode error	After the E71 receives a request from the connected device normally, communications cannot be normally performed between the E71 and the CPU module due to some causes such as noise.	Perform communications again. If the error occurs again, check for noise and other causes and/or replace the E71.
12 _H	Intelligent function module designation error	The specified intelligent function module number has not been assigned to an intelligent function module that has the buffer memory and where communications can be performed. (For example, the specified module is an I/O module or empty.)	Change the specified data contents in the control procedure or change the connecting position of the intelligent function module. Then perform communications again.
18 _H	Remote error	The remote RUN/STOP cannot be executed. (The remote STOP/PAUSE has been already executed from another module.)	Check if the remote STOP/PAUSE has been already executed from another module. Then cancel the operation to perform communications again.
1F _H	Device error	Invalid device specification	<ul style="list-style-type: none"> • Correct the specified device. • Do not access a device which does not exist.
20 _H	Link error	The CPU module of the request destination has been disconnected from the data link.	Check if the CPU module of the station number set for the PC number has been disconnected. Then remove the cause of the disconnection to perform communications again.
21 _H	Intelligent function module bus error	Memory access to the intelligent function module cannot be performed due to the following causes: <ul style="list-style-type: none"> • The control bus to the intelligent function module is faulty. • The intelligent function module is in failure. 	There is a hardware problem with the CPU module, intelligent function module, or the E71. Please consult your local Mitsubishi representative.

14.7.3 Error codes stored in the buffer memory

The following table lists the error codes stored in each buffer memory area when an error occurs. For the buffer memory areas where these error codes are stored, refer to the descriptions of the corresponding buffer memory area.

( Page 33, Section 3.5.2)

Error code	Error description	Action
00 _H	The error details and how to troubleshoot the problem are the same as those for the end code returned to the connected device in data communications. ( Page 210, Section 14.7.1)	
02 _H		
0050 _H		
0051 _H		
0052 _H		
0054 _H		
0055 _H		
0056 _H		
0057 _H		
0058 _H		
0059 _H		
005B _H	<ul style="list-style-type: none"> The CPU module and the Ethernet module cannot communicate. The programmable controller CPU cannot process the request from the connected device. 	<ul style="list-style-type: none"> Check the error code appended to the end code and correct the error.
0060 _H	<ul style="list-style-type: none"> The communication time between the CPU module and the Ethernet module exceeded the CPU monitoring timer value. 	<ul style="list-style-type: none"> Increase the CPU monitoring timer value.
0063 _H	<ul style="list-style-type: none"> In communications using a fixed buffer, the remote password of the port for the destination Ethernet module is in the lock status. 	<ul style="list-style-type: none"> After unlocking the remote password using the MC protocol, perform communications using a fixed buffer. Exclude ports used for communications using a fixed buffer from the targets of the remote password check.
00A0 _H	This request cannot be specified in connection with the connected device.	<ul style="list-style-type: none"> Correct the request details. Correct the open setting.
00A1 _H	Contents of the request cannot be analyzed because the text length or request data length is too short.	<ul style="list-style-type: none"> Check and correct the text length or request data length of the Qn header. Then send data to the Ethernet module again.
00A2 _H	This request cannot be processed.	<ul style="list-style-type: none"> Correct the request details and command.
3E8 _H to 4FFF _H	(Errors detected by a programmable controller CPU)	<ul style="list-style-type: none"> Refer to the troubleshooting section of the user's manual (hardware design, maintenance and inspection) for the CPU module used, and take corrective action.
7000 _H to 7FFF _H	(Errors detected by a module such as a serial communication module)	<ul style="list-style-type: none"> Refer to a manual such as a serial communication module user's manual, and take corrective action.
B000 _H to BFFF _H	(Errors detected by a CC-Link module)	<ul style="list-style-type: none"> Refer to the CC-Link System Master/Local Module User's Manual, and take corrective action.
C001 _H	<ul style="list-style-type: none"> The IP address setting value of the E71 for the initial process is incorrect. The setting value of the subnet mask field for the router relay function is incorrect. 	<ul style="list-style-type: none"> Correct the IP address. Set the class to A/B/C. Correct the subnet mask.

Error code	Error description	Action
C002 _H	Some of the various timer setting values for the initial process are outside the allowable range.	<ul style="list-style-type: none"> Check and correct each timer setting value for the initial process.
C003 _H	The setting value of the auto-open UDP port number for the initial process is outside the allowable range.	<ul style="list-style-type: none"> Check and correct the auto-open UDP port number.
C004 _H	The setting value of the subnet mask field is incorrect.	<ul style="list-style-type: none"> Correct the subnet mask and perform the initial process again.
C005 _H	<ul style="list-style-type: none"> The setting value of the default router IP address for the router relay function is incorrect. The network address (network address after the subnet mask) of the default router IP address is different from that of the IP address of the Ethernet module in the own station. 	<ul style="list-style-type: none"> Correct the default router IP address and perform the initial process again. Set the same network address as that of the IP address of the Ethernet module in the own station.
C006 _H	The setting value of the subnet address for the router relay function is incorrect.	<ul style="list-style-type: none"> Correct the subnet address and perform the initial process again.
C007 _H	<ul style="list-style-type: none"> The setting value of the router IP address for the router relay function is incorrect. The network address (network address after the subnet mask) of the router IP address is different from that of the IP address of the Ethernet module in the own station. 	<ul style="list-style-type: none"> Correct the router IP address and perform the initial process again. Change the IP address of the Ethernet module in the own station to the same address as the network address.
C00E _H	The send/receive process has been executed while the IP address in use detection function is being executed.	Do not execute the send/receive process while executing the IP address in use detection function.
C00F _H	One IP address is used for two or more modules.	<ul style="list-style-type: none"> Set different IP addresses. <p>The MAC addresses of the Ethernet modules in the stations with the same IP address can be checked in IP address duplication status storage area (5281_H to 5286_H).</p>
C010 _H	The setting value of the Ethernet module port number for the open process is incorrect.	<ul style="list-style-type: none"> Correct the port number.
C011 _H	The setting value of the port number of the connected device for the open process is incorrect.	<ul style="list-style-type: none"> Correct the port number.
C012 _H	The port number set is used in a connection already opened in TCP/IP.	<ul style="list-style-type: none"> Correct the port numbers of the Ethernet module and the connected device.
C013 _H	The port number used in a connection already opened is set for the open process in UDP/IP.	<ul style="list-style-type: none"> Correct the port number of the Ethernet module.
C014 _H	The initial process or open process of the Ethernet module is not completed.	<ul style="list-style-type: none"> Perform the initial process or open process.
C015 _H	The setting value of the IP address of the connected device for the open process is incorrect.	<ul style="list-style-type: none"> Correct the IP address. Set the class to A/B/C.
C016 _H	The open process of the connection (or the next connection) specified for pairing open has been already completed.	<ul style="list-style-type: none"> Check that none of the connections targeted for pairing open has been opened. Correct the combination of modules set for pairing open.
C017 _H	A connection could not be established in the open process of the TCP connection.	<ul style="list-style-type: none"> Check the operation of the connected device. Check the open process of the connected device. Correct the open setting of the communication parameters. Correct the port number of the Ethernet module, the IP address and port number of the connected device, and the open setting. Check if the cable is disconnected.
C018 _H	The setting value of the IP address of the connected device is incorrect.	<ul style="list-style-type: none"> Correct the IP address.
C020 _H	The data length exceeds the allowable range.	<ul style="list-style-type: none"> Correct the data length. When the amount of data to be sent exceeds the limit, divide the data into smaller chunks to send it.

Error code	Error description	Action
C021 _H	An abnormal end response was received for communications using a fixed buffer.	<ul style="list-style-type: none"> • Read the response end code from the connection end code or error log area and perform the required action.
C022 _H	<ul style="list-style-type: none"> • A response could not be received within the response monitoring timer value. • The connection with the connected device was closed while waiting for a response. 	<ul style="list-style-type: none"> • Check the operation of the connected device. • Correct the response monitoring timer value. • Check the open status of the connection with the connected device.
C023 _H	<ul style="list-style-type: none"> • The connection with the connected device has not been opened. • The connection with the connected device has been closed. 	<ul style="list-style-type: none"> • Open the connection with the connected device.
C030 _H	A sending error has occurred.	<ul style="list-style-type: none"> • Because there may be congestion of packets on the line, send data after a certain period of time. • Check if the cable is disconnected. • Perform the self-diagnostics test to check for an error in the Ethernet module.
C031 _H	A sending error has occurred.	<ul style="list-style-type: none"> • Because there may be congestion of packets on the line, send data after a certain period of time. • Check if the cable is disconnected. • Perform the self-diagnostics test to check for an error in the Ethernet module.
C032 _H	A TCP ULP timeout error has occurred in the TCP/IP communication. (The connected device does not send an ACK response.)	<ul style="list-style-type: none"> • Check the operation of the connected device. • Correct the TCP ULP timeout value and perform the initial process again. • Because there may be congestion of packets on the line, send data after a certain period of time. • Check if the cable is disconnected.
C033 _H	There is no connected device with the IP address that has been set.	<ul style="list-style-type: none"> • Correct the IP address of the connected device and the Ethernet address. • Use the default value (FFFFFFFF_H) if the connected device has the ARP function. If it does not have the function, set the MAC address. • Check the operation of the connected device. • Because there may be congestion of packets on the line, send data after a certain period of time. • Check if the cable is disconnected.
C034 _H	On the ARP table, there is no connected device with the IP address that has been set.	<ul style="list-style-type: none"> • Decrease the number of communication destinations.
C035 _H	The existence of the connected device could not be checked within the response monitoring timer value.	<ul style="list-style-type: none"> • Check the operation of the connected device. • Correct the settings of the alive check. • Check if the cable is disconnected.
C036 _H	Because a cable is not connected or disconnected, the send process cannot be executed.	<ul style="list-style-type: none"> • Check if the cable is disconnected. • Perform the loopback test to check for an error in the line. • Perform the self-diagnostics test to check for an error in the Ethernet module.
C040 _H	<ul style="list-style-type: none"> • 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. 	<ul style="list-style-type: none"> • Correct the data length of the communication data. • Because there may be congestion of packets on the line, correct the settings of the initial process. • Send the same data from the connected device again.

Error code	Error description	Action
C041 _H	There is an error in the checksum of the receive data when TCP is used.	<ul style="list-style-type: none"> • Correct the checksum value sent from the connected device and send the correct value. • Check the situation of the line, such as noise, the distance between the line and power cable, and the grounding of each device.
C042 _H	There is an error in the checksum of the receive data when UDP is used.	<ul style="list-style-type: none"> • Correct the checksum value sent from the connected device and send the correct value. • Check the situation of the line, such as noise, the distance between the line and power cable, and the grounding of each device.
C043 _H	The checksum in the header of the IP packet received is incorrect.	<ul style="list-style-type: none"> • Correct the checksum value sent from the connected device and send the correct value. • Check the situation of the line, such as noise, the distance between the line and power cable, and the grounding of each device.
C044 _H	An error packet of ICMP was received.	<ul style="list-style-type: none"> • Check the operation of the connected device. • Check if the cable is disconnected.
C045 _H	An error packet of ICMP was received.	<ul style="list-style-type: none"> • Check the operation of the connected device. • Check if the cable is disconnected.
C046 _H	An error packet of ICMP was received.	<ul style="list-style-type: none"> • Check the operation of the connected device. • Check if the cable is disconnected.
C047 _H	An error packet of ICMP was received.	<ul style="list-style-type: none"> • Check the operation of the connected device. • Check if the cable is disconnected.
C048 _H	An error packet of ICMP was received.	<ul style="list-style-type: none"> • Check the operation of the connected device. • Check if the cable is disconnected.
C049 _H	An error packet of ICMP was received.	<ul style="list-style-type: none"> • Check the operation of the connected device. • Because there may be congestion of packets on the line, send data after a certain period of time. • Check if the cable is disconnected. • If an alive check timeout occurs, correct the IP assembly timer value of the connected device.
C04A _H	An error packet of ICMP was received. (An IP assembly timeout error has occurred in the connected device.)	<ul style="list-style-type: none"> • Check the operation of the connected device. • Because there may be congestion of packets on the line, send data after a certain period of time. • Check if the cable is disconnected. • If an alive check timeout occurs, correct the IP assembly timer value of the connected device.
C04B _H	An IP assembly timeout error has occurred. (The remaining part of the divided data could not be received and a timeout has occurred.)	<ul style="list-style-type: none"> • Check the operation of the connected device. • Because there may be congestion of packets on the line, send data after a certain period of time. • Check if the cable is disconnected. • Correct the IP assembly timer value and perform the initial process again.
C04C _H	Because there is no space in the internal buffer, such as the IP header buffer, data cannot be sent.	<ul style="list-style-type: none"> • Send the same data again and check the receipt of a response.
C04D _H	<ul style="list-style-type: none"> • In communications using an auto-open UDP port or communications using a fixed buffer in the "No Procedure" control method, there is an error in the data length specified in the application data of the message received by the Ethernet module. • Not all the receive data can be stored. 	<ul style="list-style-type: none"> • Correct the data length. • Correct the text size so that the text data size is within the receive buffer memory size.

Error code	Error description	Action
C050 _H	When "ASCII Code" has been selected in the communication data code setting in the Ethernet operation setting, ASCII code data which cannot be converted into binary code data has been sent from the connected device.	<ul style="list-style-type: none"> • Select "Binary Code" in the Ethernet operation setting. Then restart the Ethernet module to perform communications again. • Check and correct the data sent from the connected device and send it again.
C051 _H	The number of read/write points is outside the allowable range.	<ul style="list-style-type: none"> • Correct the number of read/write points and send the data to the Ethernet module again.
C052 _H	The number of read/write points is outside the allowable range.	<ul style="list-style-type: none"> • Correct the number of read/write points and send the data to the Ethernet module again.
C053 _H	The number of read/write points is outside the allowable range.	<ul style="list-style-type: none"> • Correct the number of read/write points and send the data to the Ethernet module again.
C054 _H	The number of read/write points is outside the allowable range.	<ul style="list-style-type: none"> • Correct the number of read/write points and send the data to the Ethernet module again.
C055 _H	The number of file data read/write points is outside the allowable range.	<ul style="list-style-type: none"> • Correct the number of read/write points (or the number of bytes) and send the data to the Ethernet module again.
C056 _H	<ul style="list-style-type: none"> • The read/write request exceeds the largest address. • The address is 0. 	<ul style="list-style-type: none"> • Correct the start address or the number of read/write points and send the data to the Ethernet module again. (Make sure that the request does not exceed the largest address.)
C057 _H	The request data length does not match with the number of data in the character (a part of text).	<ul style="list-style-type: none"> • Check and correct the text or requested data length of the Qn header. Then send the data to the Ethernet module again.
C058 _H	The request data length after the ASCII/binary conversion does not match with the number of data in the character (a part of text).	<ul style="list-style-type: none"> • Check and correct the text or requested data length of the Qn header. Then send the data to the Ethernet module again.
C059 _H	Incorrect designation of the command and subcommand	<ul style="list-style-type: none"> • Correct the request details.
C05A _H	The Ethernet module cannot read/write data from/to the specified device.	<ul style="list-style-type: none"> • Check the device for read/write.
C05B _H	The Ethernet module cannot read/write data from/to the specified device.	<ul style="list-style-type: none"> • Check the device for read/write.
C05C _H	Incorrect request (For example, a request for data read/write in bit units has been issued to a word device.)	<ul style="list-style-type: none"> • Correct the request details and send it to the Ethernet module again. (For example, change the subcommand.)
C05D _H	The monitor registration is not performed.	<ul style="list-style-type: none"> • Register the monitoring data and perform monitoring.
C05E _H	The communication time between the Ethernet module and the programmable controller CPU exceeded the CPU monitoring timer value.	<ul style="list-style-type: none"> • Increase the CPU monitoring timer value. • Check that the programmable controller CPU is operating normally. • Correct the network number or PC number. • If the destination is a station with a different network number, correct the routing parameter setting. • If the destination is a station with a different network number, check if the network number is not in use.
C05F _H	This request cannot be executed to the target programmable controller.	<ul style="list-style-type: none"> • Correct the network number or PC number. • Correct the read/write request details.
C060 _H	Incorrect request (For example, incorrect data was specified for bit devices.)	<ul style="list-style-type: none"> • Correct the request details and send it to the Ethernet module again. (For example, correct the data.)
C061 _H	The request data length does not match with the number of data in the character (a part of text).	<ul style="list-style-type: none"> • Check and correct the text or requested data length of the Qn header. Then send the data to the Ethernet module again.
C062 _H	When writing was prohibited while the CPU was running, data has been written to the remote I/O station using the MC protocol (with a QnA-compatible 3E frame or 4E frame).	<ul style="list-style-type: none"> • To write data to the remote I/O station using the MC protocol (with a QnA-compatible 3E frame or 4E frame), select "Enable Write at RUN time" in the Ethernet operation setting.

Error code	Error description	Action
C070 _H	The device memory cannot be extended for the target station.	<ul style="list-style-type: none"> Read/write data without the device memory set for extension. Device memory can be extended only in an Ethernet module-connected station and a Q/QnACPU via CC-Link IE Controller Network, MELSECNET/H, or MELSECNET/10.
C071 _H	The number of device points for data read/write set for modules other than a Q/QnACPU is too large.	<ul style="list-style-type: none"> Correct the number of device points for data read/write and send the data to the Ethernet module again.
C072 _H	Incorrect request (For example, a request for data read/write in bit units has been issued to a word device.)	<ul style="list-style-type: none"> Check whether the data can be requested to the target programmable controller CPU. Correct the request details and send it to the Ethernet module again. (For example, change the subcommand.)
C073 _H	The Ethernet module cannot issue this request to the target programmable controller CPU. (For example, the number of double word access points cannot be specified for modules other than a Q/QnACPU.)	<ul style="list-style-type: none"> Correct the request details.
C074 _H	This request cannot be executed to the target programmable controller CPU.	<ul style="list-style-type: none"> Correct the network number or PC number. Correct the read/write request details.
C080 _H	The destination IP address could not be obtained during CC-Link IE Controller Network, CC-Link IE Field Network, MELSECNET/H, MELSECNET/10 relay communications or communications using data link instructions.	<ul style="list-style-type: none"> Set the Station No. <-> IP information in the Ethernet module. Change the conversion method for the CC-Link IE Controller Network, CC-Link IE Field Network, MELSECNET/H, and MELSECNET/10 relay communication parameters.
C081 _H	The termination process for the Ethernet module is in process, and whether communications using data link instructions have reached cannot be checked.	<ul style="list-style-type: none"> Finish all the communications to perform the termination process of the Ethernet module.
C082 _H	<p>The communication process was abnormally ended in the following communications.</p> <ul style="list-style-type: none"> Communications with a programming tool (UDP/IP) CC-Link IE Controller Network, CC-Link IE Field Network, MELSECNET/H, MELSECNET/10 relay communications 	<ul style="list-style-type: none"> Check if the relay station/connected station is operating normally. (If the communication is alive, no action for this error is required.) Check if there is an error with the cable connection between the own station and the connected station. If the line is heavily loaded, reduce the load. Increase the number of resends for the data link instructions. Increase the time reserved for the communication process in the service processing setting in "PLC System" under "PLC Parameter".
C083 _H	The communication process was abnormally ended in communications using data link instructions.	<ul style="list-style-type: none"> Check if the relay station/connected station is operating normally. (If the communication is alive, no action for this error is required.) Check if there is an error with the cable connection between the own station and the connected station. If the line is heavily loaded, reduce the load. Increase the time reserved for the communication process in the service processing setting in "PLC System" under "PLC Parameter".
C084 _H	The communication process was abnormally ended in communications using data link instructions.	<ul style="list-style-type: none"> Check if the own station/relay station/connected station is running normally. Check if there is a problem with the cable connection between the own station and the connected station. Increase the TCP resend timer value. Increase the time reserved for the communication process in the service processing setting in "PLC System" under "PLC Parameter".
C085 _H	The own station's channel specified by another station in communications using data link instructions is currently in use.	<ul style="list-style-type: none"> Execute the request from another station again.
C086 _H	A message exceeding the receive message size was received.	<ul style="list-style-type: none"> Correct the send message size of the request source.

Error code	Error description	Action
C087 _H	There is an error in the IP address set in the Station No. <-> IP information setting for CC-Link IE Controller Network, CC-Link IE Field Network, MELSECNET/H, MELSECNET/10 relay communications.	<ul style="list-style-type: none"> • In the Station No.<->IP information setting, set the IP addresses for the CC-Link IE Controller Network, CC-Link IE Field Network, MELSECNET/H, and MELSECNET/10 relay communication target devices.
C0B2 _H	There is no insufficient space in the receive buffer of the relay station PC number for MELSOFT connection or data link instructions, or the communication request destination station. (Receive buffer full error)	<ul style="list-style-type: none"> • 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.
C0B3 _H	A request that cannot be processed was issued from the programmable controller CPU.	<ul style="list-style-type: none"> • Correct the request details. • Correct the network number or PC number.
C0B5 _H	Data that cannot be processed by the programmable controller CPU or Ethernet module was specified.	<ul style="list-style-type: none"> • Correct the request details. • Cancel the current request.
C0B6 _H	The channel number is outside the allowable range.	<ul style="list-style-type: none"> • Specify the channel number from 1 to 8.
C0B7 _H	A channel number currently in use was specified.	<ul style="list-style-type: none"> • Change the channel number. • Perform communications again after the current communications are completed.
C0B8 _H	<ul style="list-style-type: none"> • The network number or PC number is outside the allowable range. • A response from the programmable controller CPU is faulty. 	<ul style="list-style-type: none"> • Correct the network number or PC number. • Check the programmable controller CPU operation.
C0B9 _H	The open process of the TCP connection has not been completed.	<ul style="list-style-type: none"> • Perform the open process. • Check the operation of the connected device. • After sending a close request (FIN) from the connected device to the E71, perform the open process again and wait for 500ms or more.
C0BA _H	Because the close process is in process using the CLOSE instruction, a send request cannot be accepted.	<ul style="list-style-type: none"> • Perform the open process and send the request.
C0BB _H	System error (The OS detected an error.)	<ul style="list-style-type: none"> • Follow the procedure below. (1) Check if the Ethernet module, the power supply module, and the CPU module are properly connected. (2) Check if the operating environment of the Ethernet module conforms to the general specifications of the CPU module. (3) Check if the power capacity is sufficient. (4) Check if the hardware is normal according to the manual for each module. <p>If the actions above do not solve the problem, please consult your local Mitsubishi representative.</p>
C0BC _H	The specified communication line is closed.	<ul style="list-style-type: none"> • Open the communication line. • Correct the target connection number.
C0BD _H	Requests have been accepted successively and cannot be sent.	<ul style="list-style-type: none"> • Check if requests have been sent successively without waiting for a response.
C0BE _H	System error (The OS detected an error.)	<ul style="list-style-type: none"> • Follow the procedure below. (1) Check if the Ethernet module, the power supply module, and the CPU module are properly connected. (2) Check if the operating environment of the Ethernet module conforms to the general specifications of the CPU module. (3) Check if the power capacity is sufficient. (4) Check if the hardware is normal according to the manual for each module. <p>If the actions above do not solve the problem, please consult your local Mitsubishi representative.</p>

Error code	Error description	Action
C0BF _H	System error (The OS detected an error.)	<ul style="list-style-type: none"> Follow the procedure below. (1) Check if the Ethernet module, the power supply module, and the CPU module are properly connected. (2) Check if the operating environment of the Ethernet module conforms to the general specifications of the CPU module. (3) Check if the power capacity is sufficient. (4) Check if the hardware is normal according to the manual for each module. <p>If the actions above do not solve the problem, please consult your local Mitsubishi representative.</p>
C0C0 _H	The open process of the UDP connection has not been completed.	<ul style="list-style-type: none"> Perform the open process. Check the operation of the connected device.
C0C1 _H	The transmission interval of UDP is too short.	<ul style="list-style-type: none"> Check if a send request is being repeatedly issued. Increase the send interval.
C0C2 _H	System error (The OS detected an error.)	<ul style="list-style-type: none"> Follow the procedure below. (1) Check if the Ethernet module, the power supply module, and the CPU module are properly connected. (2) Check if the operating environment of the Ethernet module conforms to the general specifications of the CPU module. (3) Check if the power capacity is sufficient. (4) Check if the hardware is normal according to the manual for each module. <p>If the actions above do not solve the problem, please consult your local Mitsubishi representative.</p>
C0C3 _H	System error (The OS detected an error.)	<ul style="list-style-type: none"> Follow the procedure below. (1) Check if the Ethernet module, the power supply module, and the CPU module are properly connected. (2) Check if the operating environment of the Ethernet module conforms to the general specifications of the CPU module. (3) Check if the power capacity is sufficient. (4) Check if the hardware is normal according to the manual for each module. <p>If the actions above do not solve the problem, please consult your local Mitsubishi representative.</p>
C0C4 _H	The UINI instruction has been executed during communications.	<ul style="list-style-type: none"> Execute the UINI instruction after closing all connections.
C0C5 _H	<ul style="list-style-type: none"> A send request was issued to the connected device with the class/network address different from those of the own station when the router relay function is not used. The setting of the router relay parameter is incorrect. 	<ul style="list-style-type: none"> Enable the router relay function and perform the initial process. Set the correct data in the router relay parameter. Correct the IP address of the connected device and perform the open process. Check if the network address is correct. After changing the network address, perform the initial process again.
C0C6 _H	System error (The OS detected an error.)	<ul style="list-style-type: none"> Follow the procedure below. (1) Check if the Ethernet module, the power supply module, and the CPU module are properly connected. (2) Check if the operating environment of the Ethernet module conforms to the general specifications of the CPU module. (3) Check if the power capacity is sufficient. (4) Check if the hardware is normal according to the manual for each module. <p>If the actions above do not solve the problem, please consult your local Mitsubishi representative.</p>
C0C7 _H	An Ethernet module system error has occurred.	<ul style="list-style-type: none"> Perform the initial process again.

Error code	Error description	Action
C0C8 _H	System error (The OS detected an error.)	<ul style="list-style-type: none"> • Follow the procedure below. (1) Check if the Ethernet module, the power supply module, and the CPU module are properly connected. (2) Check if the operating environment of the Ethernet module conforms to the general specifications of the CPU module. (3) Check if the power capacity is sufficient. (4) Check if the hardware is normal according to the manual for each module. <p>If the actions above do not solve the problem, please consult your local Mitsubishi representative.</p>
C0C9 _H	System error (The OS detected an error.)	<ul style="list-style-type: none"> • Follow the procedure below. (1) Check if the Ethernet module, the power supply module, and the CPU module are properly connected. (2) Check if the operating environment of the Ethernet module conforms to the general specifications of the CPU module. (3) Check if the power capacity is sufficient. (4) Check if the hardware is normal according to the manual for each module. <p>If the actions above do not solve the problem, please consult your local Mitsubishi representative.</p>
C0CA _H	System error (The OS detected an error.)	<ul style="list-style-type: none"> • Follow the procedure below. (1) Check if the Ethernet module, the power supply module, and the CPU module are properly connected. (2) Check if the operating environment of the Ethernet module conforms to the general specifications of the CPU module. (3) Check if the power capacity is sufficient. (4) Check if the hardware is normal according to the manual for each module. <p>If the actions above do not solve the problem, please consult your local Mitsubishi representative.</p>
C0CB _H	Another send request was issued before the send process has not been completed.	<ul style="list-style-type: none"> • Wait for completion of transmission before sending the next send request.
C0CC _H	System error (The OS detected an error.)	<ul style="list-style-type: none"> • Follow the procedure below. (1) Check if the Ethernet module, the power supply module, and the CPU module are properly connected. (2) Check if the operating environment of the Ethernet module conforms to the general specifications of the CPU module. (3) Check if the power capacity is sufficient. (4) Check if the hardware is normal according to the manual for each module. <p>If the actions above do not solve the problem, please consult your local Mitsubishi representative.</p>
C0CF _H	System error (The OS detected an error.)	<ul style="list-style-type: none"> • Follow the procedure below. (1) Check if the Ethernet module, the power supply module, and the CPU module are properly connected. (2) Check if the operating environment of the Ethernet module conforms to the general specifications of the CPU module. (3) Check if the power capacity is sufficient. (4) Check if the hardware is normal according to the manual for each module. <p>If the actions above do not solve the problem, please consult your local Mitsubishi representative.</p>
C0D0 _H	Incorrect data length was specified.	<ul style="list-style-type: none"> • Correct the specified data length.
C0D1 _H	The number of resends is incorrect.	<ul style="list-style-type: none"> • Correct the number of resends.

Error code	Error description	Action
C0D2 _H	The arrival monitoring time is incorrect.	<ul style="list-style-type: none"> • Correct the arrival monitoring time.
C0D3 _H	The number of relay stations in CC-Link IE Controller Network, CC-Link IE Field Network, MELSECNET/H, and MELSECNET/10 exceeded the allowable range.	<ul style="list-style-type: none"> • Check the specified value for the communication destination. • Correct the settings in the Station No. <-> IP information for the stations between the own station and the communication destination.
C0D4 _H	The number of relay stations in CC-Link IE Controller Network, CC-Link IE Field Network, MELSECNET/H, and MELSECNET/10 exceeded the allowable range.	<ul style="list-style-type: none"> • Check the specified value for the communication destination. • Correct the settings in the Station No. <-> IP information for the stations between the own station and the communication destination.
C0D5 _H	The number of retries is incorrect.	<ul style="list-style-type: none"> • Correct the number of retries.
C0D6 _H	The network number or station number is incorrect.	<ul style="list-style-type: none"> • Correct the settings for the communication destination. • Correct the specified value for the communication destination.
C0D7 _H	Data were sent without the initial process completed.	<ul style="list-style-type: none"> • Set the parameters with the programming tool and write them to the CPU module. Then perform communications with the connected device. • Wait for normal completion of the initial process before performing communications with the connected device.
C0D8 _H	The number of blocks exceeded the range.	<ul style="list-style-type: none"> • Correct the specified value for the number of blocks.
C0D9 _H	The subcommand value is incorrect.	<ul style="list-style-type: none"> • Correct the specified value for the subcommand.
C0DA _H	A response to the PING test could not be received within the time of the communication time check.	<ul style="list-style-type: none"> • Correct the IP address and host name of the Ethernet module where the PING test is executed. • Enable the Ethernet module where the PING test is executed to perform communications. (Complete the initial process.)
C0DB _H	The IP address and host name of the Ethernet module for the PING test are incorrect.	<ul style="list-style-type: none"> • Correct the IP address and host name of the Ethernet module to perform the PING test.
C0DC _H	System error (The OS detected an error.)	<ul style="list-style-type: none"> • Follow the procedure below. <ol style="list-style-type: none"> (1) Check if the Ethernet module, the power supply module, and the CPU module are properly connected. (2) Check if the operating environment of the Ethernet module conforms to the general specifications of the CPU module. (3) Check if the power capacity is sufficient. (4) Check if the hardware is normal according to the manual for each module. <p>If the actions above do not solve the problem, please consult your local Mitsubishi representative.</p>
C0DD _H	System error (The OS detected an error.)	<ul style="list-style-type: none"> • Follow the procedure below. <ol style="list-style-type: none"> (1) Check if the Ethernet module, the power supply module, and the CPU module are properly connected. (2) Check if the operating environment of the Ethernet module conforms to the general specifications of the CPU module. (3) Check if the power capacity is sufficient. (4) Check if the hardware is normal according to the manual for each module. <p>If the actions above do not solve the problem, please consult your local Mitsubishi representative.</p>
C0DE _H	Data could not be received within the specified arrival monitoring time.	<ul style="list-style-type: none"> • Correct the specified arrival monitoring time. • Correct the specified channel number. • Check the status of the send source station and relay station.

Error code	Error description	Action
C0DF _H	System error (The OS detected an error.)	<ul style="list-style-type: none"> • Follow the procedure below. (1) Check if the Ethernet module, the power supply module, and the CPU module are properly connected. (2) Check if the operating environment of the Ethernet module conforms to the general specifications of the CPU module. (3) Check if the power capacity is sufficient. (4) Check if the hardware is normal according to the manual for each module. <p>If the actions above do not solve the problem, please consult your local Mitsubishi representative.</p>
C0E0 _H	An error was detected in the programmable controller CPU.	<ul style="list-style-type: none"> • Check that the programmable controller CPU and each intelligent function module are properly connected. • Check that the programmable controller CPU is not in the reset status. • Check if an error exists in the programmable controller CPU. If an error exists, take the action required to correct it. • Replace the power supply module, programmable controller CPU, or intelligent function module.
C0E1 _H	An error was detected in the programmable controller CPU.	<ul style="list-style-type: none"> • Check that the programmable controller CPU and each intelligent function module are properly connected. • Check that the programmable controller CPU is not in the reset status. • Check if an error exists in the programmable controller CPU. If an error exists, take the action required to correct it. • Replace the power supply module, programmable controller CPU, or intelligent function module.
C0E2 _H	An error was detected in the programmable controller CPU.	<ul style="list-style-type: none"> • Check that the programmable controller CPU and each intelligent function module are properly connected. • Check that the programmable controller CPU is not in the reset status. • Check if an error exists in the programmable controller CPU. If an error exists, take the action required to correct it. • Replace the power supply module, programmable controller CPU, or intelligent function module.
C0E3 _H	An error was detected in the programmable controller CPU.	<ul style="list-style-type: none"> • Check that the programmable controller CPU and each intelligent function module are properly connected. • Check that the programmable controller CPU is not in the reset status. • Check if an error exists in the programmable controller CPU. If an error exists, take the action required to correct it. • Replace the power supply module, programmable controller CPU, or intelligent function module.
C0E4 _H	An error was detected in the programmable controller CPU.	<ul style="list-style-type: none"> • Check that the programmable controller CPU and each intelligent function module are properly connected. • Check that the programmable controller CPU is not in the reset status. • Check if an error exists in the programmable controller CPU. If an error exists, take the action required to correct it. • Replace the power supply module, programmable controller CPU, or intelligent function module.

Error code	Error description	Action
C0E5 _H	An error was detected in the programmable controller CPU.	<ul style="list-style-type: none"> • Check that the programmable controller CPU and each intelligent function module are properly connected. • Check that the programmable controller CPU is not in the reset status. • Check if an error exists in the programmable controller CPU. If an error exists, take the action required to correct it. • Replace the power supply module, programmable controller CPU, or intelligent function module.
C0E6 _H	An error was detected in the programmable controller CPU.	<ul style="list-style-type: none"> • Check that the programmable controller CPU and each intelligent function module are properly connected. • Check that the programmable controller CPU is not in the reset status. • Check if an error exists in the programmable controller CPU. If an error exists, take the action required to correct it. • Replace the power supply module, programmable controller CPU, or intelligent function module.
C0E7 _H	An error was detected in the programmable controller CPU.	<ul style="list-style-type: none"> • Check that the programmable controller CPU and each intelligent function module are properly connected. • Check that the programmable controller CPU is not in the reset status. • Check if an error exists in the programmable controller CPU. If an error exists, take the action required to correct it. • Replace the power supply module, programmable controller CPU, or intelligent function module.
C0E8 _H	An error was detected in the programmable controller CPU.	<ul style="list-style-type: none"> • Check that the programmable controller CPU and each intelligent function module are properly connected. • Check that the programmable controller CPU is not in the reset status. • Check if an error exists in the programmable controller CPU. If an error exists, take the action required to correct it. • Replace the power supply module, programmable controller CPU, or intelligent function module.
C0E9 _H	An error was detected in the programmable controller CPU.	<ul style="list-style-type: none"> • Check that the programmable controller CPU and each intelligent function module are properly connected. • Check that the programmable controller CPU is not in the reset status. • Check if an error exists in the programmable controller CPU. If an error exists, take the action required to correct it. • Replace the power supply module, programmable controller CPU, or intelligent function module.
C0EA _H	An error was detected in the programmable controller CPU.	<ul style="list-style-type: none"> • Check that the programmable controller CPU and each intelligent function module are properly connected. • Check that the programmable controller CPU is not in the reset status. • Check if an error exists in the programmable controller CPU. If an error exists, take the action required to correct it. • Replace the power supply module, programmable controller CPU, or intelligent function module.

Error code	Error description	Action
C0EB _H	An error was detected in the programmable controller CPU.	<ul style="list-style-type: none"> • Check that the programmable controller CPU and each intelligent function module are properly connected. • Check that the programmable controller CPU is not in the reset status. • Check if an error exists in the programmable controller CPU. If an error exists, take the action required to correct it. • Replace the power supply module, programmable controller CPU, or intelligent function module.
C0EC _H	An error was detected in the programmable controller CPU.	<ul style="list-style-type: none"> • Check that the programmable controller CPU and each intelligent function module are properly connected. • Check that the programmable controller CPU is not in the reset status. • Check if an error exists in the programmable controller CPU. If an error exists, take the action required to correct it. • Replace the power supply module, programmable controller CPU, or intelligent function module.
C0ED _H	An error was detected in the programmable controller CPU.	<ul style="list-style-type: none"> • Check that the programmable controller CPU and each intelligent function module are properly connected. • Check that the programmable controller CPU is not in the reset status. • Check if an error exists in the programmable controller CPU. If an error exists, take the action required to correct it. • Replace the power supply module, programmable controller CPU, or intelligent function module.
C0EE _H	An error was detected in the programmable controller CPU.	<ul style="list-style-type: none"> • Check that the programmable controller CPU and each intelligent function module are properly connected. • Check that the programmable controller CPU is not in the reset status. • Check if an error exists in the programmable controller CPU. If an error exists, take the action required to correct it. • Replace the power supply module, programmable controller CPU, or intelligent function module.
C0EF _H	An error was detected in the programmable controller CPU.	<ul style="list-style-type: none"> • Check that the programmable controller CPU and each intelligent function module are properly connected. • Check that the programmable controller CPU is not in the reset status. • Check if an error exists in the programmable controller CPU. If an error exists, take the action required to correct it. • Replace the power supply module, programmable controller CPU, or intelligent function module.
C0F0 _H	An Ethernet module RAM error was detected in the hardware test (H/W test).	<ul style="list-style-type: none"> • Perform the hardware test again. If an error occurs again, the Ethernet module may have a hardware error. Please consult your local Mitsubishi representative.
C0F1 _H	An Ethernet module ROM error was detected in the hardware test (H/W test).	<ul style="list-style-type: none"> • Perform the hardware test again. If an error occurs again, the Ethernet module may have a hardware error. Please consult your local Mitsubishi representative.
C0F3 _H	A system error (major error) was detected in the CPU module.	<ul style="list-style-type: none"> • Remove the error cause of the CPU module in the own station.

Error code	Error description	Action
C0F4 _H	System error (The OS detected an error.)	<ul style="list-style-type: none"> • Follow the procedure below. (1) Check if the Ethernet module, the power supply module, and the CPU module are properly connected. (2) Check if the operating environment of the Ethernet module conforms to the general specifications of the CPU module. (3) Check if the power capacity is sufficient. (4) Check if the hardware is normal according to the manual for each module. <p>If the actions above do not solve the problem, please consult your local Mitsubishi representative.</p>
C0F5 _H	System error (The OS detected an error.)	<ul style="list-style-type: none"> • Follow the procedure below. (1) Check if the Ethernet module, the power supply module, and the CPU module are properly connected. (2) Check if the operating environment of the Ethernet module conforms to the general specifications of the CPU module. (3) Check if the power capacity is sufficient. (4) Check if the hardware is normal according to the manual for each module. <p>If the actions above do not solve the problem, please consult your local Mitsubishi representative.</p>
C0F6 _H	System error (The OS detected an error.)	<ul style="list-style-type: none"> • Follow the procedure below. (1) Check if the Ethernet module, the power supply module, and the CPU module are properly connected. (2) Check if the operating environment of the Ethernet module conforms to the general specifications of the CPU module. (3) Check if the power capacity is sufficient. (4) Check if the hardware is normal according to the manual for each module. <p>If the actions above do not solve the problem, please consult your local Mitsubishi representative.</p>
C0F7 _H	An error has occurred in the self-loopback test.	<ul style="list-style-type: none"> • Because there may be congestion of packets on the line, send data from the connected device after a certain period of time. • Check if the cable is disconnected. <p>If the actions above do not solve the problem, please consult your local Mitsubishi representative.</p>
C0F8 _H	System error (The OS detected an error.)	<ul style="list-style-type: none"> • Follow the procedure below. (1) Check if the Ethernet module, the power supply module, and the CPU module are properly connected. (2) Check if the operating environment of the Ethernet module conforms to the general specifications of the CPU module. (3) Check if the power capacity is sufficient. (4) Check if the hardware is normal according to the manual for each module. <p>If the actions above do not solve the problem, please consult your local Mitsubishi representative.</p>
C100 _H	System error (The OS detected an error.)	<ul style="list-style-type: none"> • Follow the procedure below. (1) Check if the Ethernet module, the power supply module, and the CPU module are properly connected. (2) Check if the operating environment of the Ethernet module conforms to the general specifications of the CPU module. (3) Check if the power capacity is sufficient. (4) Check if the hardware is normal according to the manual for each module. <p>If the actions above do not solve the problem, please consult your local Mitsubishi representative.</p>

Error code	Error description	Action
C101 _H	A response could not be received from the DNS client.	<ul style="list-style-type: none"> • Check the address of the DNS server. • Check if communications with the DNS server can be performed using the PING command. • Check if the IP addresses of the own station and the DNS server belong to the same class. (If they belong to different classes, check the router setting.)
C102 _H	A response from the SMTP layer could not be received.	<ul style="list-style-type: none"> • Check if the SMTP server name has been registered in the DNS. • Delete the SMTP server name and change the setting to the IP address setting. Then check the operation. • Check if communications with the SMTP server can be performed using the PING command.
C103 _H	System error (The OS detected an error.)	<ul style="list-style-type: none"> • Follow the procedure below. (1) Check if the Ethernet module, the power supply module, and the CPU module are properly connected. (2) Check if the operating environment of the Ethernet module conforms to the general specifications of the CPU module. (3) Check if the power capacity is sufficient. (4) Check if the hardware is normal according to the manual for each module. • If the actions above do not solve the problem, please consult your local Mitsubishi representative.
C104 _H	System error (The OS detected an error.)	<ul style="list-style-type: none"> • Follow the procedure below. (1) Check if the Ethernet module, the power supply module, and the CPU module are properly connected. (2) Check if the operating environment of the Ethernet module conforms to the general specifications of the CPU module. (3) Check if the power capacity is sufficient. (4) Check if the hardware is normal according to the manual for each module. If the actions above do not solve the problem, please consult your local Mitsubishi representative.
C105 _H	System error (The OS detected an error.)	<ul style="list-style-type: none"> • Follow the procedure below. (1) Check if the Ethernet module, the power supply module, and the CPU module are properly connected. (2) Check if the operating environment of the Ethernet module conforms to the general specifications of the CPU module. (3) Check if the power capacity is sufficient. (4) Check if the hardware is normal according to the manual for each module. If the actions above do not solve the problem, please consult your local Mitsubishi representative.
C106 _H	System error (The OS detected an error.)	<ul style="list-style-type: none"> • Follow the procedure below. (1) Check if the Ethernet module, the power supply module, and the CPU module are properly connected. (2) Check if the operating environment of the Ethernet module conforms to the general specifications of the CPU module. (3) Check if the power capacity is sufficient. (4) Check if the hardware is normal according to the manual for each module. If the actions above do not solve the problem, please consult your local Mitsubishi representative.

Error code	Error description	Action
C110 _H	System error (The OS detected an error.)	<ul style="list-style-type: none"> • Follow the procedure below. (1) Check if the Ethernet module, the power supply module, and the CPU module are properly connected. (2) Check if the operating environment of the Ethernet module conforms to the general specifications of the CPU module. (3) Check if the power capacity is sufficient. (4) Check if the hardware is normal according to the manual for each module. <p>If the actions above do not solve the problem, please consult your local Mitsubishi representative.</p>
C111 _H	A response could not be received from the DNS client.	<ul style="list-style-type: none"> • Check the cables and hubs. • Check if communications with the DNS server can be performed using the PING command.
C112 _H	A response from the POP3 layer could not be received.	<ul style="list-style-type: none"> • Check if the POP3 server name has been registered in the DNS. • Delete the POP3 server name and change the setting to the IP address setting. Then check the operation. • Check if communications with the POP3 server can be performed using the PING command.
C113 _H	An e-mail without an attachment was received. (This error occurs when an attachment is not read normally.)	<ul style="list-style-type: none"> • Specify the attachment on the sending side. • Check the program on the sending side. • If data sending using the MSEND instruction previously failed with the sending source of the mail server, check the destination of the MSEND instruction. • Check if the e-mail specifications, such as the encoding/decoding style and file format, on the sending side are the same as those for the Ethernet module. • Receive an e-mail with an unknown destination and unknown destination server from the SMTP server.
C114 _H	An e-mail that was received has an attachment with an invalid name.	<ul style="list-style-type: none"> • Check if the file extension of the attachment is "bin" or "asc" on the sending side. • Check that the e-mail is not compressed or encrypted. • Check the destination of the MSEND instruction. • Receive an e-mail with an unknown destination and unknown destination server from the SMTP server.
C115 _H	System error (The OS detected an error.)	<ul style="list-style-type: none"> • Follow the procedure below. (1) Check if the Ethernet module, the power supply module, and the CPU module are properly connected. (2) Check if the operating environment of the Ethernet module conforms to the general specifications of the CPU module. (3) Check if the power capacity is sufficient. (4) Check if the hardware is normal according to the manual for each module. <p>If the actions above do not solve the problem, please consult your local Mitsubishi representative.</p>
C116 _H	System error (The OS detected an error.)	<ul style="list-style-type: none"> • Follow the procedure below. (1) Check if the Ethernet module, the power supply module, and the CPU module are properly connected. (2) Check if the operating environment of the Ethernet module conforms to the general specifications of the CPU module. (3) Check if the power capacity is sufficient. (4) Check if the hardware is normal according to the manual for each module. <p>If the actions above do not solve the problem, please consult your local Mitsubishi representative.</p>

Error code	Error description	Action
C117 _H	System error (The OS detected an error.)	<ul style="list-style-type: none"> • Follow the procedure below. (1) Check if the Ethernet module, the power supply module, and the CPU module are properly connected. (2) Check if the operating environment of the Ethernet module conforms to the general specifications of the CPU module. (3) Check if the power capacity is sufficient. (4) Check if the hardware is normal according to the manual for each module. <p>If the actions above do not solve the problem, please consult your local Mitsubishi representative.</p>
C118 _H	System error (The OS detected an error.)	<ul style="list-style-type: none"> • Follow the procedure below. (1) Check if the Ethernet module, the power supply module, and the CPU module are properly connected. (2) Check if the operating environment of the Ethernet module conforms to the general specifications of the CPU module. (3) Check if the power capacity is sufficient. (4) Check if the hardware is normal according to the manual for each module. <p>If the actions above do not solve the problem, please consult your local Mitsubishi representative.</p>
C119 _H	There is no received mail.	<ul style="list-style-type: none"> • Read Mail information storage area (address: 2682_H) of the buffer memory and read received e-mails remaining on the server.
C11A _H	The received mail failed to be converted.	<ul style="list-style-type: none"> • Check that the e-mail is not compressed or encrypted. • Check if the e-mail specifications, such as the encoding/decoding style and file format, on the sending side are the same as those for the Ethernet module. • Check if the sending side has not divided the file.
C11B _H	An e-mail was sent and an error mail was received from the mail server of the destination.	<ul style="list-style-type: none"> • Receive an e-mail with an unknown destination and unknown destination server from the SMTP server. (The received mail is stored in the mail buffer.) • Check if the portion before "@" is correct in the mail address setting in the parameter settings. • Check if the portion before "@" has been registered on the destination mail server.
C11D _H	The attachment size exceeded the allowable size.	<ul style="list-style-type: none"> • Check if the attachment size is 6K words or less. • Check that the sending side has not divided the attachment.
C120 _H	The SMTP server could not be opened.	<ul style="list-style-type: none"> • Check that the port number of the SMTP server is 25. • Check if communications with the SMTP server can be performed using the PING command.
C121 _H	Communications cannot be performed with the SMTP server. (Error response)	<ul style="list-style-type: none"> • Check if the SMTP server is busy.
C122 _H	Communications cannot be performed with the SMTP server. (Abort)	<ul style="list-style-type: none"> • Check if the SMTP server is busy.
C123 _H	Communications cannot be performed with the SMTP server. (Reset response)	<ul style="list-style-type: none"> • Check if the SMTP server is busy.
C124 _H	A response from the SMTP server timed out.	<ul style="list-style-type: none"> • Check for an error in the SMTP server. • Check if the network is heavily loaded.
C125 _H	Forcefully disconnected from the SMTP server	<ul style="list-style-type: none"> • Check for an error in the SMTP server. • Check if the network is heavily loaded.
C126 _H	The SMTP server could not be closed.	<ul style="list-style-type: none"> • Check for an error in the SMTP server. • Check if the network is heavily loaded.
C127 _H	Closing the SMTP server gave an error response.	<ul style="list-style-type: none"> • Check for an error in the SMTP server. • Check if the network is heavily loaded.

Error code	Error description	Action
C130 _H	The communication channel is closed because the service is not available.	<ul style="list-style-type: none"> • Check the status of the SMTP server.
C131 _H	The SMTP server is in process, and an error response was received.	<ul style="list-style-type: none"> • Check that a user name not registered in the server has not been specified. • Send the e-mail again after a certain period of time.
C132 _H	The SMTP server is in process, and an error response was received. (Local error)	<ul style="list-style-type: none"> • Check the status of the SMTP server.
C133 _H	The SMTP server is in process, and an error response was received. (Insufficient memory area)	<ul style="list-style-type: none"> • Check the status of the SMTP server.
C134 _H	System error (The OS detected an error.)	<ul style="list-style-type: none"> • Follow the procedure below. (1) Check if the Ethernet module, the power supply module, and the CPU module are properly connected. (2) Check if the operating environment of the Ethernet module conforms to the general specifications of the CPU module. (3) Check if the power capacity is sufficient. (4) Check if the hardware is normal according to the manual for each module. <p>If the actions above do not solve the problem, please consult your local Mitsubishi representative.</p>
C135 _H	System error (The OS detected an error.)	<ul style="list-style-type: none"> • Follow the procedure below. (1) Check if the Ethernet module, the power supply module, and the CPU module are properly connected. (2) Check if the operating environment of the Ethernet module conforms to the general specifications of the CPU module. (3) Check if the power capacity is sufficient. (4) Check if the hardware is normal according to the manual for each module. <p>If the actions above do not solve the problem, please consult your local Mitsubishi representative.</p>
C136 _H	System error (The OS detected an error.)	<ul style="list-style-type: none"> • Follow the procedure below. (1) Check if the Ethernet module, the power supply module, and the CPU module are properly connected. (2) Check if the operating environment of the Ethernet module conforms to the general specifications of the CPU module. (3) Check if the power capacity is sufficient. (4) Check if the hardware is normal according to the manual for each module. <p>If the actions above do not solve the problem, please consult your local Mitsubishi representative.</p>
C137 _H	System error (The OS detected an error.)	<ul style="list-style-type: none"> • Follow the procedure below. (1) Check if the Ethernet module, the power supply module, and the CPU module are properly connected. (2) Check if the operating environment of the Ethernet module conforms to the general specifications of the CPU module. (3) Check if the power capacity is sufficient. (4) Check if the hardware is normal according to the manual for each module. <p>If the actions above do not solve the problem, please consult your local Mitsubishi representative.</p>
C138 _H	The SMTP server is in process, and an error response was received. (Mailbox not found)	<ul style="list-style-type: none"> • Check if the mail address of the Ethernet module has been correctly set.

Error code	Error description	Action
C139 _H	System error (The OS detected an error.)	<ul style="list-style-type: none"> • Follow the procedure below. (1) Check if the Ethernet module, the power supply module, and the CPU module are properly connected. (2) Check if the operating environment of the Ethernet module conforms to the general specifications of the CPU module. (3) Check if the power capacity is sufficient. (4) Check if the hardware is normal according to the manual for each module. <p>If the actions above do not solve the problem, please consult your local Mitsubishi representative.</p>
C13A _H	The SMTP server is in process, and an error response was received. (Exceeded the allocation of the memory area)	<ul style="list-style-type: none"> • Check the status of the SMTP server.
C13B _H	The SMTP server is in process, and an error response was received. (Illegal mailbox name)	<ul style="list-style-type: none"> • Check if the mail address of the Ethernet module has been correctly set.
C13C _H	System error (The OS detected an error.)	<ul style="list-style-type: none"> • Follow the procedure below. (1) Check if the Ethernet module, the power supply module, and the CPU module are properly connected. (2) Check if the operating environment of the Ethernet module conforms to the general specifications of the CPU module. (3) Check if the power capacity is sufficient. (4) Check if the hardware is normal according to the manual for each module. <p>If the actions above do not solve the problem, please consult your local Mitsubishi representative.</p>
C140 _H	The POP3 server could not be opened.	<ul style="list-style-type: none"> • Check that the port number of the POP3 server is 110. (The port number of the Ethernet module is fixed to 110.) • Check if communications with the POP3 server can be performed using the PING command.
C141 _H	Communications cannot be performed with the POP3 server. (Error response)	<ul style="list-style-type: none"> • Check if the POP3 server is busy.
C142 _H	Communications cannot be performed with the POP3 server. (Abort)	<ul style="list-style-type: none"> • Check if the POP3 server is busy.
C143 _H	Communications cannot be performed with the POP3 server. (Reset response)	<ul style="list-style-type: none"> • Check if the POP3 server is busy.
C144 _H	A response from the POP3 server could not be received.	<ul style="list-style-type: none"> • Check for an error in the POP3 server. • Check if the network is heavily loaded.
C145 _H	Forcefully disconnected from the POP3 server	<ul style="list-style-type: none"> • Check for an error in the POP3 server.
C146 _H	The POP3 server could not be closed.	<ul style="list-style-type: none"> • Check for an error in the POP3 server. • Check if the network is heavily loaded.
C147 _H	Closing the POP3 server gave an error response.	<ul style="list-style-type: none"> • Check for an error in the POP3 server.
C150 _H	POP3 server verification error	<ul style="list-style-type: none"> • Check the status of the POP3 server.
C151 _H	The Ethernet module's mail address (e-mail setting parameter) is different from the account name in the mailbox on the server side.	<ul style="list-style-type: none"> • Check the account name of the mailbox on the server side and correct the account of the mailbox set in the Ethernet module.
C152 _H	The Ethernet module's password (e-mail setting parameter) is different from the password on the server side.	<ul style="list-style-type: none"> • Check the password on the server side and correct the password of the Ethernet module.
C153 _H	An error has occurred in acquisition of the received mail list. (The list of the mail that arrived at the POP3 server failed to be acquired.)	<ul style="list-style-type: none"> • Reset the server inquiry time to the default value and restart the CPU module of the own station.

Error code	Error description	Action
C154 _H	An error has occurred upon receiving a mail. (An e-mail cannot be read from the POP3 server.)	<ul style="list-style-type: none"> • Check that the e-mail is not compressed or encrypted. • Check if the e-mail specifications, such as the encoding/decoding style and file format, on the sending side are the same as those for the Ethernet module.
C160 _H	A response was received from the DNS server after timeout.	<ul style="list-style-type: none"> • Check if the network is heavily loaded. • Check the status of the DNS server.
C161 _H	A response from the DNS server could not be received.	<ul style="list-style-type: none"> • Check if the network is heavily loaded. • Check the status of the DNS server.
C162 _H	System error (The OS detected an error.)	<ul style="list-style-type: none"> • Follow the procedure below. (1) Check if the Ethernet module, the power supply module, and the CPU module are properly connected. (2) Check if the operating environment of the Ethernet module conforms to the general specifications of the CPU module. (3) Check if the power capacity is sufficient. (4) Check if the hardware is normal according to the manual for each module. <p>If the actions above do not solve the problem, please consult your local Mitsubishi representative.</p>
C163 _H	System error (The OS detected an error.)	<ul style="list-style-type: none"> • Follow the procedure below. (1) Check if the Ethernet module, the power supply module, and the CPU module are properly connected. (2) Check if the operating environment of the Ethernet module conforms to the general specifications of the CPU module. (3) Check if the power capacity is sufficient. (4) Check if the hardware is normal according to the manual for each module. <p>If the actions above do not solve the problem, please consult your local Mitsubishi representative.</p>
C171 _H	An error was returned from the DNS server.	<ul style="list-style-type: none"> • Check if the IP address of the DNS server is correct. • Check if the mail server names (SMTP server names and POP server names) are correct. • Check with the network administrator if the DNS function of the server specified in the DNS setting is properly operating.
C172 _H	An error was returned from the DNS server.	<ul style="list-style-type: none"> • Check if the IP address of the DNS server is correct. • Check if the mail server names (SMTP server names and POP server names) are correct. • Check with the network administrator if the DNS function of the server specified in the DNS setting is properly operating.
C173 _H	An error was returned from the DNS server.	<ul style="list-style-type: none"> • Check if the IP address of the DNS server is correct. • Check if the mail server names (SMTP server names and POP server names) are correct. • Check with the network administrator if the DNS function of the server specified in the DNS setting is properly operating.
C174 _H	An error was returned from the DNS server.	<ul style="list-style-type: none"> • Check if the IP address of the DNS server is correct. • Check if the mail server names (SMTP server names and POP server names) are correct. • Check with the network administrator if the DNS function of the server specified in the DNS setting is properly operating.
C175 _H	An error was returned from the DNS server.	<ul style="list-style-type: none"> • Check if the IP address of the DNS server is correct. • Check if the mail server names (SMTP server names and POP server names) are correct. • Check with the network administrator if the DNS function of the server specified in the DNS setting is properly operating.

Error code	Error description	Action
C176 _H	An error was returned from the DNS server.	<ul style="list-style-type: none"> • Check if the IP address of the DNS server is correct. • Check if the mail server names (SMTP server names and POP server names) are correct. • Check with the network administrator if the DNS function of the server specified in the DNS setting is properly operating.
C177 _H	An error was returned from the DNS server.	<ul style="list-style-type: none"> • Check if the IP address of the DNS server is correct. • Check if the mail server names (SMTP server names and POP server names) are correct. • Check with the network administrator if the DNS function of the server specified in the DNS setting is properly operating.
C178 _H	An error was returned from the DNS server.	<ul style="list-style-type: none"> • Check if the IP address of the DNS server is correct. • Check if the mail server names (SMTP server names and POP server names) are correct. • Check with the network administrator if the DNS function of the server specified in the DNS setting is properly operating.
C179 _H	An error was returned from the DNS server.	<ul style="list-style-type: none"> • Check if the IP address of the DNS server is correct. • Check if the mail server names (SMTP server names and POP server names) are correct. • Check with the network administrator if the DNS function of the server specified in the DNS setting is properly operating.
C17A _H	An error was returned from the DNS server.	<ul style="list-style-type: none"> • Check if the IP address of the DNS server is correct. • Check if the mail server names (SMTP server names and POP server names) are correct. • Check with the network administrator if the DNS function of the server specified in the DNS setting is properly operating.
C17B _H	An error was returned from the DNS server.	<ul style="list-style-type: none"> • Check if the IP address of the DNS server is correct. • Check if the mail server names (SMTP server names and POP server names) are correct. • Check with the network administrator if the DNS function of the server specified in the DNS setting is properly operating.
C17C _H	An error was returned from the DNS server.	<ul style="list-style-type: none"> • Check if the IP address of the DNS server is correct. • Check if the mail server names (SMTP server names and POP server names) are correct. • Check with the network administrator if the DNS function of the server specified in the DNS setting is properly operating.
C17D _H	An error was returned from the DNS server.	<ul style="list-style-type: none"> • Check if the IP address of the DNS server is correct. • Check if the mail server names (SMTP server names and POP server names) are correct. • Check with the network administrator if the DNS function of the server specified in the DNS setting is properly operating.
C17E _H	An error was returned from the DNS server.	<ul style="list-style-type: none"> • Check if the IP address of the DNS server is correct. • Check if the mail server names (SMTP server names and POP server names) are correct. • Check with the network administrator if the DNS function of the server specified in the DNS setting is properly operating.
C17F _H	An error was returned from the DNS server.	<ul style="list-style-type: none"> • Check if the IP address of the DNS server is correct. • Check if the mail server names (SMTP server names and POP server names) are correct. • Check with the network administrator if the DNS function of the server specified in the DNS setting is properly operating.
C180 _H	The device specified in "Condition Device" in "News Setting" is outside the range set in the [Device] tab of the PLC parameter window.	<ul style="list-style-type: none"> • Correct the device settings in the PLC parameter window and specify the condition devices within the range of the device settings.

Error code	Error description	Action
C1A0 _H	An illegal request was made.	Try again. If the same error occurs, the Ethernet module may have a hardware error. Please consult your local Mitsubishi representative.
C1A2 _H	A response to the request could not be received.	• Check and correct the response waiting time.
C1A4 _H	A request or subrequest is incorrect.	• Correct the request and subrequest.
C1A5 _H	The specified target station or clear target is incorrect.	• Correct the specified target station or clear target.
C1A6 _H	The specified connection number is incorrect.	• Specify the connection number within the range of 1 to 16. • Do not select the connection number 8 or 16 when using the pairing open method.
C1A7 _H	The specified network number is incorrect.	• Correct the specified network number.
C1A8 _H	The specified station number is incorrect.	• Correct the specified station number.
C1A9 _H	The specified device number is incorrect.	• Correct the specified device number.
C1AA _H	The specified device name is incorrect.	• Correct the specified device name.
C1AC _H	The specified number of resends is incorrect.	• Correct the number of resends.
C1AD _H	The specified data length is incorrect.	• Correct the specified data length.
C1AE _H	The send/receive data length and header length are incorrect.	• Correct the specified send/receive data length and header length to values within the allowable ranges. • The send/receive data length needs to be equal to or longer than the header length.
C1AF _H	The specified port number is incorrect.	• Correct the specified port number.
C1B0 _H	The open process of the specified connection has been already completed.	• Perform the close process before the open process.
C1B1 _H	The open process of the specified connection has not been completed.	• Perform the open process.
C1B2 _H	The OPEN or CLOSE instruction is being executed in the specified connection.	• Try again after the OPEN or CLOSE instruction is completed.
C1B3 _H	Another send or receive instruction is being executed in the specified channel.	• Change the channel number. • Try again after the send or receive instruction is completed.
C1B4 _H	The specified arrival monitoring time is incorrect.	• Set the arrival monitoring time to a value within the allowable range.
C1B5 _H	Data could not be received within the specified arrival monitoring time.	• Correct the specified arrival monitoring time.
C1B6 _H	The specified send mail address number is incorrect.	• Correct the specified send mail address number. • Correct the send mail address setting in the parameter setting.
C1B7 _H	A reading operation was executed while no receiving e-mail was stored in the mail buffer data area.	• If there is an mail received in the mail information, execute the MRECV instruction.
C1B8 _H	The RECV instruction was executed for the channel that had not received data.	• Correct the execution condition of the RECV instruction. • Correct the channel number.
C1B9 _H	The OPEN instruction cannot be executed for the specified connection.	• Correct the connection number.
C1BA _H	The dedicated instruction was executed with the initialization not completed.	• Execute the dedicated instruction after the initial process is completed.
C1BB _H	The target station CPU type is incorrect.	• Correct the specified target station CPU type.
C200 _H	The remote password is incorrect.	• Correct the remote password, and unlock/lock the remote password again.
C201 _H	The remote password status of the port used for communications is in the lock status.	• After unlocking the remote password, perform communications.


Error code	Error description	Action
C202 _H	When another station was accessed, the remote password could not be unlocked.	<ul style="list-style-type: none"> 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.
C203 _H	System error (The OS detected an error.)	<ul style="list-style-type: none"> Follow the procedure below. <ol style="list-style-type: none"> Check if the Ethernet module, the power supply module, and the CPU module are properly connected. Check if the operating environment of the Ethernet module conforms to the general specifications of the CPU module. Check if the power capacity is sufficient. Check if the hardware is normal according to the manual for each module. If the actions above do not solve the problem, please consult your local Mitsubishi representative.
C204 _H	The device is different from the one requesting the remote password unlock.	<ul style="list-style-type: none"> Request the lock process of the remote password from the connected device that requested the unlock process of the remote password.
C205 _H	When another station was accessed, the remote password could not be unlocked.	<ul style="list-style-type: none"> 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.
C206 _H	System error (The OS detected an error.)	<ul style="list-style-type: none"> Follow the procedure below. <ol style="list-style-type: none"> Check if the Ethernet module, the power supply module, and the CPU module are properly connected. Check if the operating environment of the Ethernet module conforms to the general specifications of the CPU module. Check if the power capacity is sufficient. Check if the hardware is normal according to the manual for each module. If the actions above do not solve the problem, please consult your local Mitsubishi representative.
C207 _H	The file name has too many characters.	<ul style="list-style-type: none"> Name the file with 255 characters or less.
C300 _H	A response could not be received within the response monitoring timer value.	<ul style="list-style-type: none"> Check the operation of the connected device. Correct the response monitoring timer value.
D000 _H to DFFF _H	(Errors detected by CC-Link IE Field Network)	<ul style="list-style-type: none"> Refer to the manual for each module.
E000 _H to EFFF _H	(Errors detected by CC-Link IE Controller Network)	<ul style="list-style-type: none"> Refer to the manual for each module.
F000 _H to FFFF _H	(Errors detected by the MELSECNET/H and MELSECNET/10 network system)	<ul style="list-style-type: none"> Refer to the MELSECNET/H or MELSECNET/10 network system reference manual, and take corrective action.

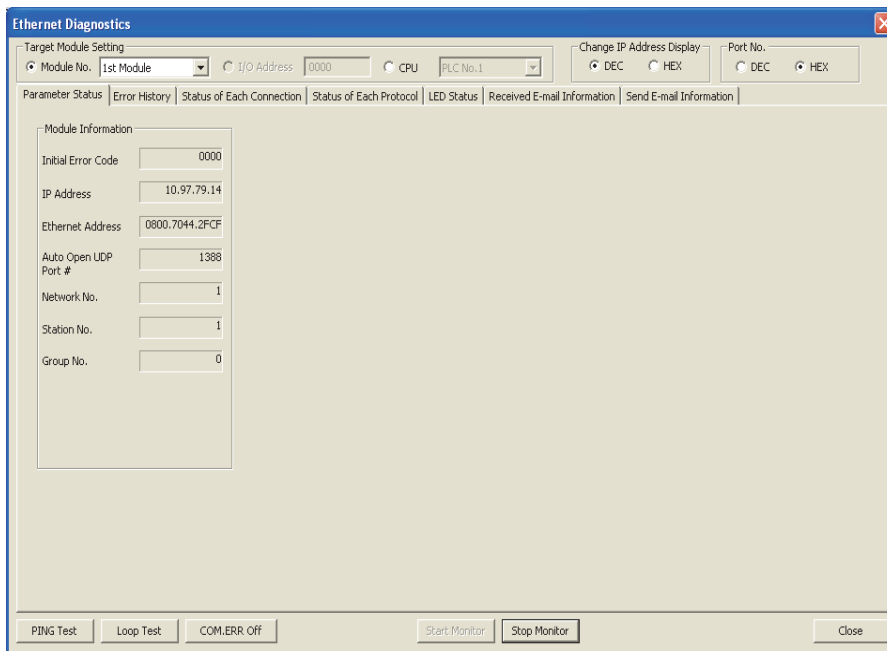
Remark

The error codes listed in the table include those returned to the connected device. Each buffer memory may also store the error numbers of the response messages returned from the connected device. If an error code other than those listed is stored, check the manual for the connected device and the response message from the connected device.





14.8 Ethernet Diagnostics

Items such as the module status, parameter settings, communication status, and error history of an E71 can be checked. Perform the diagnostics in the "Ethernet Diagnostics" window of the programming tool.

 [Diagnostics] ⇔ [Ethernet Diagnostics...]




14

Item	Description
Target Module Setting (Module No.)	Specify the E71 to be monitored. The number of other network modules is not included.
Parameter Status	Module information (such as the IP address and basic settings) is displayed.
Error History	The number of errors and their details (such as the error code and information about the connection where the error has occurred) is displayed.
Status of Each Connection	The status of each connection (such as error codes and the open setting) is displayed.
Status of Each Protocol	Information such as the total number of data sending/receiving times for each protocol is displayed.
LED Status	The operation mode and the LED status of the E71 are displayed.
Received E-mail Information	Received mail information and the error log are displayed.
Send E-mail Information	Sent mail information and the error log are displayed.
 button	The "PING Test" window is displayed.
 button	The "Loopback Test" window is displayed.
 button	The COM.ERR. LED is turned off. ( Page 240, Section 14.10)

14.8 Ethernet Diagnostics


Remark

For each window and detailed explanations, refer to the following.
 Operating manual for the programming tool used

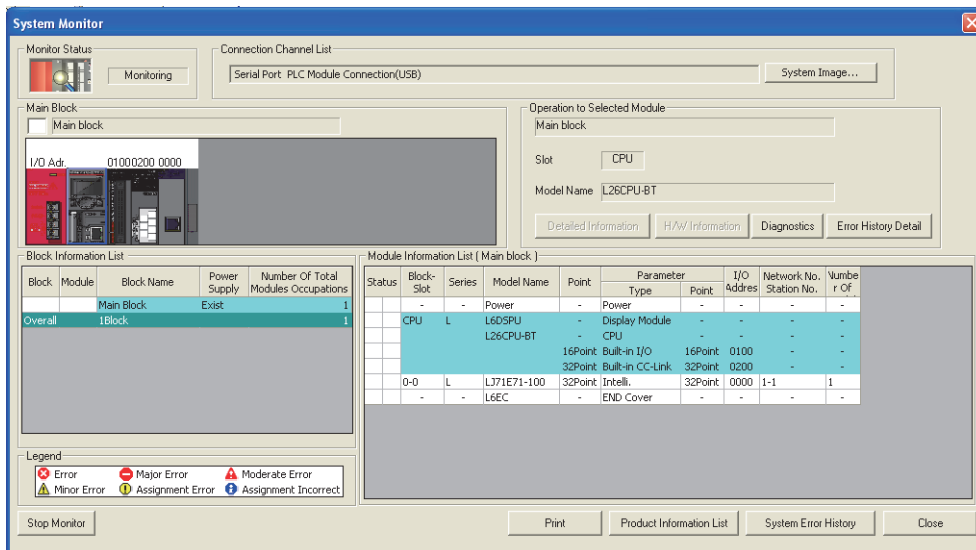
14.9 Checking the Status of the E71 with the System Monitor

Open the system monitor window of a programming tool to check the LED status, connection status, and operating status of the E71.

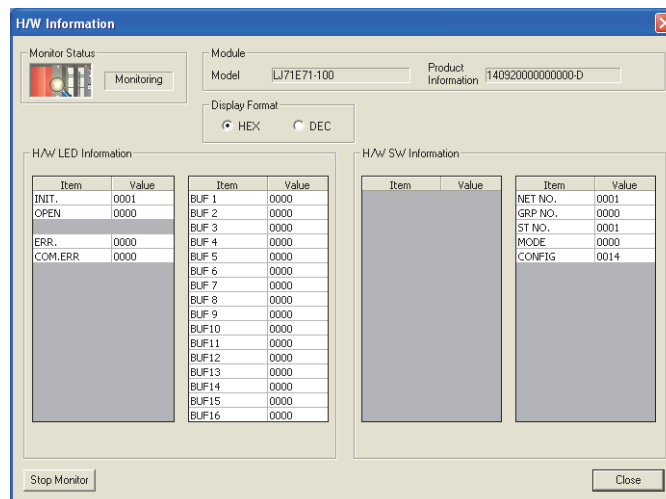
1. Open the "System Monitor" window.

 [Diagnostics] => [System Monitor]

2. Select the module to diagnose in the "Main Block" field in the upper left of the window. Then click the **H/W Information** button.



3. The "H/W Information" is displayed.



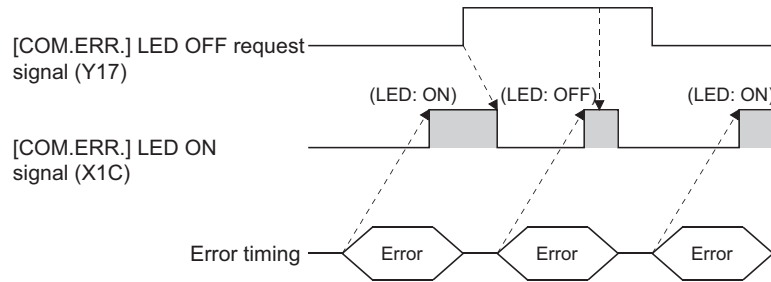
Item		Description																																															
Display Format		Switches the input format of "H/W LED Information" and "H/W SW Information".																																															
H/W LED Information	INIT.	Indicates the statuses of the LEDs on the E71. 0000: Off 0001: On																																															
	OPEN																																																
	ERR.																																																
	COM. ERR.																																																
	BUF1 to BUF16	Indicates the connection statuses of the connection numbers 1 to 16. 0000: Open not completed 0001: Open completed																																															
H/W SW Information	NET NO.	Indicates the network No. of the E71.																																															
	GRP NO.	Indicates the group No. of the E71.																																															
	ST NO.	Indicates the station number of the E71.																																															
	MODE	Indicates the mode of the E71. 0000: Online 0001: Offline 0002: Self-loopback test 0003: H/W test																																															
	CONFIG	<p>Indicates the operating status of the E71, which has been set with a programming tool.</p> <table border="1" style="margin-left: auto; margin-right: auto;"> <tr> <td>b15</td><td>to</td><td>b9</td><td>b8</td><td>b7</td><td>b6</td><td>b5</td><td>b4</td><td>b3</td><td>b2</td><td>b1</td><td>b0</td> </tr> <tr> <td>-</td><td></td><td>1/0</td><td>-</td><td>1/0</td><td>1/0</td><td>1/0</td><td>-</td><td>1/0</td><td>1/0</td><td>-</td><td></td> </tr> <tr> <td></td><td></td><td>↑</td><td></td><td>↑</td><td>↑</td><td>↑</td><td></td><td>↑</td><td>↑</td><td></td><td></td> </tr> <tr> <td></td><td></td><td>6)</td><td></td><td>5)</td><td>4)</td><td>3)</td><td></td><td>2)</td><td>1)</td><td></td><td></td> </tr> </table> <div style="display: flex; justify-content: space-between;"> <div style="width: 48%;"> <p>1) Communication data code setting 0: Communications in a binary code 1: Communications in an ASCII code</p> <p>2) Initial/open method setting 0: No parameter setting (start up according to the sequence program) 1: Parameter setting (start up according to the parameters)</p> <p>3) TCP Existence confirmation setting 0: Use the Ping 1: Use the KeepAlive</p> </div> <div style="width: 48%;"> <p>4) Send frame setting 0: Ethernet frame 1: IEEE 802.3 frame</p> <p>5) Setting of write enable/disable at RUN time 0: Disable 1: Enable</p> <p>6) Initial timing setting 0: Do not wait for OPEN (Communications impossible at STOP time) 1: Always wait for OPEN (Communication possible at STOP time)</p> </div> </div>	b15	to	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0	-		1/0	-	1/0	1/0	1/0	-	1/0	1/0	-				↑		↑	↑	↑		↑	↑					6)		5)	4)	3)		2)	1)	
b15	to	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0																																						
-		1/0	-	1/0	1/0	1/0	-	1/0	1/0	-																																							
		↑		↑	↑	↑		↑	↑																																								
		6)		5)	4)	3)		2)	1)																																								

14.10 How to Turn Off the COM.ERR. LED

The COM.ERR. LED turns on when an error occurs during communications with the connected device. The COM.ERR. LED does not turn off even after the error cause is removed. Turn it off using the following methods.

(1) Using an I/O signal

Turn off the COM.ERR. LED by turning on COM.ERR. LED OFF request (Y17). An off request continues to be issued while COM.ERR. LED OFF request (Y17) is on. However, this method does not clear the error information in the error log area in the buffer memory.



(2) Using a dedicated instruction

Turn off the COM.ERR. LED using the ERRCLR instruction. (👉 Page 175, Section 13.10) This method can also clear the error information (initial error code, open error code) or the error log.

(3) In the "Ethernet Diagnostics" window

Turn off the COM.ERR. LED by clicking the button in the "Ethernet Diagnostics" window. However, this method does not clear the error information in the error log area in the buffer memory. For the window, refer to "Ethernet Diagnostics". (👉 Page 237, Section 14.8)

(4) Using the MC protocol

Turn off the COM.ERR. LED using the dedicated command for the MC protocol (LED off, Error code initialization: 1617). For details, refer to the following.

📖 MELSEC-Q/L MELSEC Communication Protocol Reference Manual

APPENDICES

Appendix 1 Processing Time

Calculate the minimum processing time for each function using the expressions below. Note that the processing time may become longer depending on the load factor on the network (how congested the line is), the window size of each connected device, the number of connections, and system configuration. Use the values obtained from the expressions as a guideline for the processing time when communications are performed in only one connection.

(1) Minimum processing time of communications using a fixed buffer (communications between the E71s)

(a) Procedure exists

$$Tfs = St + Ke + (Kdf \times Df) + Sr$$

- Tfs : The time from the start of sending to the completion of sending (unit: ms)
- St : Sending station scan time
- Ke, Kdf : Constant (Refer to the table below.)
- Df : Number of send data words
- Sr : Receiving station scan time

Item	LJ71E71-100			
	TCP/IP communications		UDP/IP communications	
	Ke	Kdf	Ke	Kdf
Communications using a binary code	7.3	0.0033	6.2	0.0040
Communications using an ASCII code	7.3	0.0082	6.2	0.0081

(b) No procedures

$$Tfs = St + Ke + (Kdf \times Df)$$

- Tfs : The time from the start of sending to the completion of sending (unit: ms)
- St : Sending station scan time
- Ke, Kdf : Constant (Refer to the table below.)
- Df : Number of send data bytes

Item	LJ71E71-100			
	TCP/IP communications		UDP/IP communications	
	Ke	Kdf	Ke	Kdf
Communications using a binary code	5.4	0.0010	3.8	0.0009

[Calculation example]

The time between the E71 sending 1017-word data of a binary code and it completing the sending when the E71 communicates with another E71 in TCP/IP with the "Procedure Exist" method (unit: ms)

Assume that the scan time on the sending side is 10ms, and the scan time on the receiving side is 8ms.

- $28.6561 \text{ (ms)} \doteq 10 + 7.3 + (0.0033 \times 1017) + 8$

(2) Minimum processing time of communications using a random access buffer

$Trs = Kr + (Kdr \times Df) + \text{ACK processing time of a connected device (added only for TCP/IP communications)}$

- Trs : The time between the E71 receiving request data from a personal computer and it completing the process (unit: ms)
- Kr, Kdr : Constant (Refer to the table below.)
- Df : Number of request data words
- ACK processing time of the connected device : The time between a random access buffer reading/writing completed and the connected device returning ACK

Item		LJ71E71-100			
		TCP/IP communications		UDP/IP communications	
		Kr	Kdr	Kr	Kdr
Reading	Communications using a binary code	2.0	0.0010	1.4	0.0012
	Communications using an ASCII code	2.0	0.0032	1.4	0.0033
Writing	Communications using a binary code	2.0	0.0016	1.4	0.0018
	Communications using an ASCII code	2.0	0.0056	1.4	0.0048

[Calculation example 1]

The time between the E71 receiving request data from a personal computer and it completing reading when the E71 communicates with the personal computer in TCP/IP and reads 508-word data of a binary code from a random access buffer (unit: ms)

- $2.508 + \text{ACK processing time of the connected device (ms)} \doteq 2.0 + (0.0010 \times 508) + \text{ACK processing time of the connected device}$

[Calculation example 2]

The time between the E71 receiving request data from a personal computer and it completing writing when the E71 communicates with the personal computer in TCP/IP and writes 508-word data of a binary code to a random access buffer (unit: ms)

- $2.8128 + \text{ACK processing time of the connected device (ms)} \doteq 2.0 + (0.0016 \times 508) + \text{ACK processing time of the connected device}$

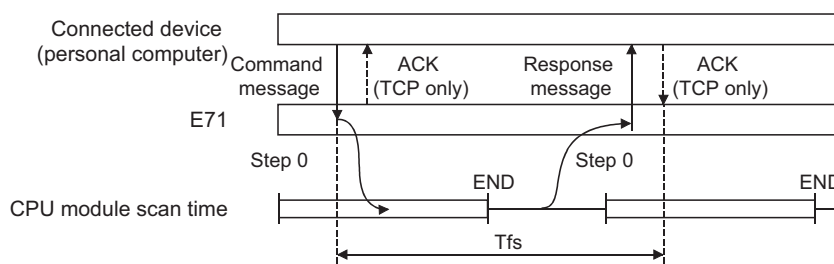
(3) Minimum processing time of MC protocol communications (batch reading and batch writing)

$T_{fs} = K_e + (K_{dt} \times D_f) + S_{cr} \times \text{number of scans required for the process} + \text{ACK processing time of the connected device}$

- T_{fs} : The time between the E71 receiving request data from a personal computer and it completing the process (unit: ms)^{*1}
- K_e, K_{dt} : Constant (Refer to the table below.)
- D_f : Number of request data words + number of response data words (Application data part)
- S_{cr} : CPU module processing time
 - Own station access: E71-connected station scan time
 - Another station access over MELSECNET/10: Transmission delay time + E71-connected station scan time

Item		LJ71E71-100			
		TCP/IP communications		UDP/IP communications	
		K_e	K_{dt}	K_e	K_{dt}
Batch reading	Communications using a binary code	3.3	0.0022	2.6	0.0018
	Communications using an ASCII code	3.3	0.0046	2.7	0.0046
Batch writing	Communications using a binary code	3.3	0.0024	2.6	0.0033
	Communications using an ASCII code	3.3	0.0073	2.7	0.0070

*1 The time from when the E71 receives the request data from the personal computer until it completes the process is shown below.



[Calculation example 1]

The time between the E71 receiving request data from a personal computer and it completing reading when the E71 communicates with the personal computer in TCP/IP and reads 100-point data of an ASCII code from the data register (D) in the own station (unit: ms)

Assume that the scan time of the E71-connected station is 10ms, the command data length is 21 words, and the response data length is 211 words.

- $14.37 + \text{ACK processing time of the connected device (ms)} \doteq 3.3 + (0.0046 \times (21 + 211)) + 10 \times 1 + \text{ACK processing time of the connected device}$

[Calculation example 2]

The time between the E71 receiving request data from a personal computer and it completing writing when the E71 communicates with the personal computer in TCP/IP and writes 100-point data of an ASCII code to the data register (D) in the own station (unit: ms)

Assume that the scan time of the E71-connected station is 10ms, the command data length is 221 words, the response data length is 11 words, and writing during RUN is enabled.

- $14.99 \text{ (ms)} + \text{ACK processing time of the connected device} \doteq 3.3 + (0.0073 \times (221 + 11)) + 10 \times 1 + \text{ACK processing time of the connected device}$

A

Appendix 1 Processing Time

(4) Processing time of the dedicated instructions

The following table lists approximate operation processing time of each dedicated instruction. The operation processing time differs depending on the system configuration and the scan time on the sending/receiving stations.

Instruction name	Number of access points		Processing time (unit: ms)		Instruction execution condition
			LCPU		
	Condition 1)	Condition 2)	For 1)	For 2)	
BUFRVCV	1 word	1017 words	0.80	1.16	TCP/IP communications, binary code communications, and fixed buffer communications (procedure exists)
BUFRCVS			0.23	0.61	
BUFSND			7.44	10.80	
CLOSE	1 port		2.36		A UDP/IP transmission port is closed.
ERRCLR	All error information is cleared.		1.06		-
ERRRD	An initial error code is read.		1.04		-
OPEN	1 port		2.12		A UDP/IP transmission port is open.
RECVS	1 word	960 words	0.23	0.62	Communications between E71-connected stations
		480 words		0.42	
READ		960 words	7.20	12.50	
		480 words		9.84	
SREAD		960 words	7.28	12.60	
		480 words		9.92	
RECV		960 words	1.38	2.00	
		480 words		1.86	
SEND		960 words	4.64	8.32	
		480 words		6.32	
WRITE		960 words	7.28	13.00	
		480 words		9.84	
SWRITE		960 words	7.36	13.00	
		480 words		10.20	
ZNRD		230 words	7.12	7.84	
ZNWR			7.04	8.48	
UINI	-		11.20		Time between the UINI instruction accepted and reinitialization completed (X19 turning on)

Appendix 2 Port Numbers Used for the E71

The following table lists port numbers reserved for an E71 system.

Port number	Application
1388 _H (5000)	Auto-open UDP port (Default value) ^{*1}
1389 _H (5001)	MELSOFT application transmission port (UDP)
138A _H (5002)	MELSOFT application transmission port (TCP)

*1 The port number can be changed.

Ex. The port number can be changed by overwriting the buffer memory address, Automatic open UDP port number (address: 20 (14_H)).

Appendix 3 Initial Process

The initial process is for enabling data communications with connected devices by setting the minimum number of parameters for the E71.

In the MELSEC-L series, the initial process is performed automatically.

(1) Checking the initial process result


The results of the initial process can be checked using the LEDs and I/O signals.

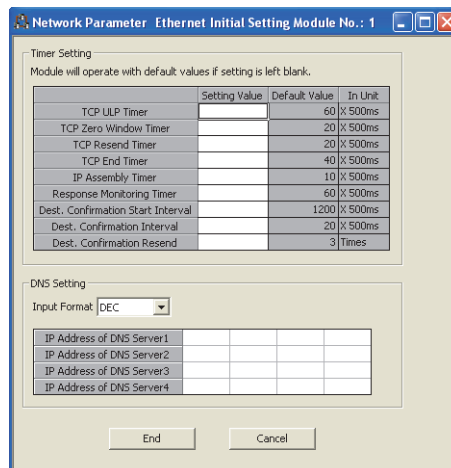
Initial process	INIT. LED	Initial normal completion signal (X19)	Initial abnormal completion signal (X1A)
Upon normal completion	ON	ON	OFF
Upon abnormal end	OFF	OFF	ON

If the initial process is not normally completed, correct the set values of parameters listed in Chapter 7 and those in the initial process setting parameter described in this chapter. After correcting the parameters, write them to the CPU module again, power off and on the CPU module or reset the CPU module.

Appendix 3.1 Setting the initial process


Configure the setting in the initial setting window. Each setting time is calculated by "setting value × 500ms".

Project window ⇨ [Parameter] ⇨ [Network Parameter] ⇨ [Ethernet/CC IE Field] ⇨ Select "Ethernet" under "Network Type". ⇨  button



	Item	Description	Setting range
Timer Setting	TCP ULP Timer	Set Time to Live during packet sending in TCP/IP.	2 to 32767
	TCP Zero Window Timer	Set the receive buffer value on the recipient. If the receive buffer on the recipient has no free space (window size = 0), data sending is waited until the recipient has free receive buffer space. At this time, the sender sends packets for checking the sending window to the recipient according to the TCP zero window timer value to check the receiving status.	2 to 32767
	TCP Resend Timer	Set the time to resend ACK if it is not returned when a connection is opened or data is sent in TCP/IP. This timer is also used for Time to Live for the ARP function. (ARP is resent by the time of "TCP resend timer value/2" if no response to an ARP request is returned.) This timer is also used as the minimum set time for arrival monitoring time for data link instructions.	2 to 32767
	TCP End Timer	When closing the TCP/IP connection by the own station, set the monitoring time for waiting for a FIN from the connected device after the own station sends a FIN and an ACK is returned from the connected device. If a FIN is not received from the connected device within the time specified by the TCP end timer value, a RST is sent to the connected device and the connection is forcibly closed.	2 to 32767
	IP Assembly Timer	The communication data may be divided at the IP level before being sent due to the buffer limitation of the sending station or the receiving station. Set the waiting time for the divided data in such a case.	1 to 32766
	Response Monitoring Timer	Set the following waiting time. <ul style="list-style-type: none"> • Time for waiting a response after a command is sent • For receiving divided messages, time until the last message is received after the first message is received 	2 to 32767
	Dest. Confirmation Start Interval	Set the time until starting the alive check after communications with the connected device of the connection that has been opened where the alive check target exists is lost.	1 to 32767
	Dest. Confirmation Interval	For the connection that is opened where the alive check target exists, set the interval to perform the alive check again when a response from the connected device as the alive check target is not received.	1 to 32767
	Dest. Confirmation Resend	For the connection that has been opened where the alive check target exists, set the number of times that the alive check is performed again when a response from the connected device as the alive check target is not received.	1 to 32767
DNS Setting*1	Input Format	Select an input format.	DEC/HEX
	IP Address of DNS Server1	Set the IP address of each DNS server.	-
	IP Address of DNS Server2		
	IP Address of DNS Server3		
	IP Address of DNS Server4		

*1 The DNS setting is configured for using the e-mail function. For details, refer to the following.

 MELSEC-Q/L Ethernet Interface Module User's Manual (Application)

(1) Precautions for settings

- Specify the setting value of each timer on the E71 side so that the following formula is met.

$$\left(\begin{array}{c} \text{Response} \\ \text{monitoring} \\ \text{timer value} \end{array} \right) \geq \left(\begin{array}{c} \text{TCP ULP} \\ \text{timer value} \end{array} \right) \geq \left(\begin{array}{c} \text{TCP end} \\ \text{timer value} \end{array} \right) \geq \left(\begin{array}{c} \text{TCP} \\ \text{resend} \\ \text{timer value} \end{array} \right) > \left(\begin{array}{c} \text{IP} \\ \text{assembly} \\ \text{timer value} \end{array} \right)$$

$$\left(\begin{array}{c} \text{TCP} \\ \text{resend} \\ \text{timer value} \end{array} \right) = \left(\begin{array}{c} \text{TCP zero window} \\ \text{timer value} \end{array} \right)$$

When connecting Mitsubishi products to the line, configure the same settings for both modules.

- Specify the setting value of each timer on the connected device side so that the following formula is met. The frequency of a communication error, such as a transmission timeout, may be higher if the timer values do not meet the formula.

$$\left(\begin{array}{c} \text{TCP resend} \\ \text{timer value on} \\ \text{the connected device} \end{array} \right) > \left(\begin{array}{c} \text{TCP resend timer value} \\ \text{on the E71} \end{array} \right)$$

$$\left(\begin{array}{c} \text{Monitoring timer value of} \\ \text{the application software on} \\ \text{the connected device} \end{array} \right) > \left\{ \left(\begin{array}{c} \text{TCP ULP timer value} \\ \text{on the E71} \end{array} \right) \times n + 1 \right\}$$

*1 "n" is the number of TCP segment transmission and is calculated by the following formula.

$$n = \text{A value that } \left(\frac{\text{Message size sent by the E71}}{\text{Maximum Segment Size}} \right) \text{ is rounded up to the nearest integer}$$

Ex. The number of TCP segment transmission when communications are performed on the same line
The Maximum Segment Size is 1460 bytes on the same line (without a router) and the number of TCP Segment transmission is as follows.

- n = 1 when the size of the message sent by the E71 is 1460 bytes or less
- n = 2 when the size of the message sent by the E71 is greater than 1460 bytes

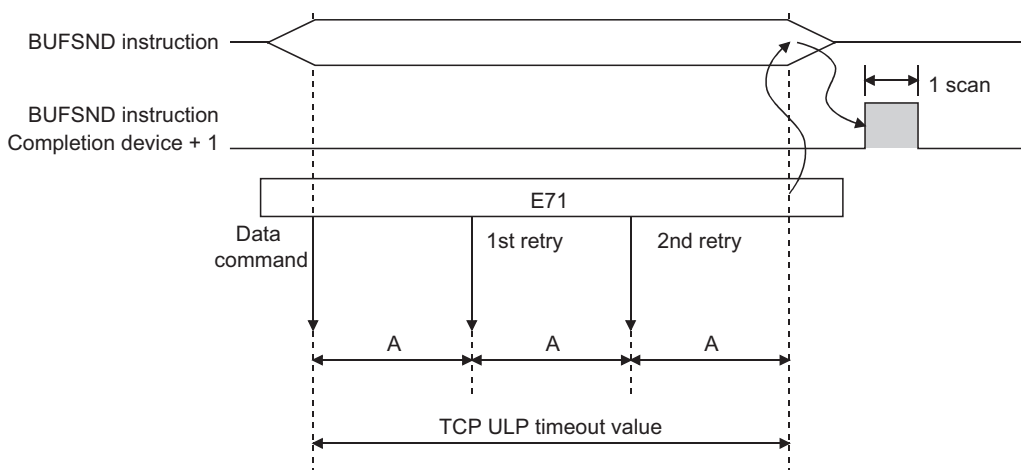
Ex. The number of TCP segment transmission when communications are performed on a different line
The Maximum Segment Size is at least 536 bytes on another line (via a dialup router or other communication device) and the number of TCP Segment transmission is as follows.

- n = 1 when the size of the message sent by the E71 is 536 bytes or less
- n = 2 when the size of the message sent by the E71 is greater than 536 bytes and no more than 1072 bytes
- n = 3 when the size of the message sent by the E71 is greater than 1072 bytes and no more than 1608 bytes

- When a communication failure occurs due to a problem such as noise, change the value so that the number of retries may increase. The number of retries is obtained using the following formula. (When using the default values, $2 = (60 \div 20) - 1$)

$$\left(\text{Number of retries} \right) = \left\{ \left(\frac{\text{TCP ULP timer value}}{\text{TCP resend timer value}} \right) - 1 \right\}$$

Ex. When the number of retries is two and data sending fails, a data sending error occurs at the timing shown in the figure below (in communications using the fixed buffer).



A: TCP resend timer value
(The time at which data are sent when ACK is not returned after data transmission)

- When not performing the above retry process (set 0 times), configure the setting so that the following formula is met.

$$\left(\text{TCP ULP timer value} \right) = \left(\text{TCP end timer value} \right) = \left(\text{TCP resend timer value} \right)$$

(Set the same value for the timer values.)

A

Appendix 3 Initial Process
Appendix 3.1 Setting the initial process

Appendix 3.2 Reinitialization process

The reinitialization process sets the E71 to the status after startup without restarting the programmable controller (or resetting the CPU module). Perform the reinitialization process using a program.

(1) The purpose of a reinitialization process

The purpose of a reinitialization process is described.

(a) To update address information of a connected device held by the E71

To prevent other devices from illegally accessing the programmable controller using the IP address of the connected device with which communications are normally performed, the E71 holds the IP address of the connected device and the corresponding MAC address.*1 If a module or the board of the connected device is replaced due to a failure, the address information of the connected device held by the E71 must be cleared by the reinitialization process.

*1 A MAC address is unique to a device. No devices share the same MAC address.

(b) To change the IP address of the E71 on the own station

When a system is changed, by only changing the own station IP address set by the programming tool, communications with the connected device can be resumed.

(c) To change the Ethernet operation setting

By changing the communication conditions in the Ethernet operation setting set using the programming tool, communications with the connected device can be resumed.

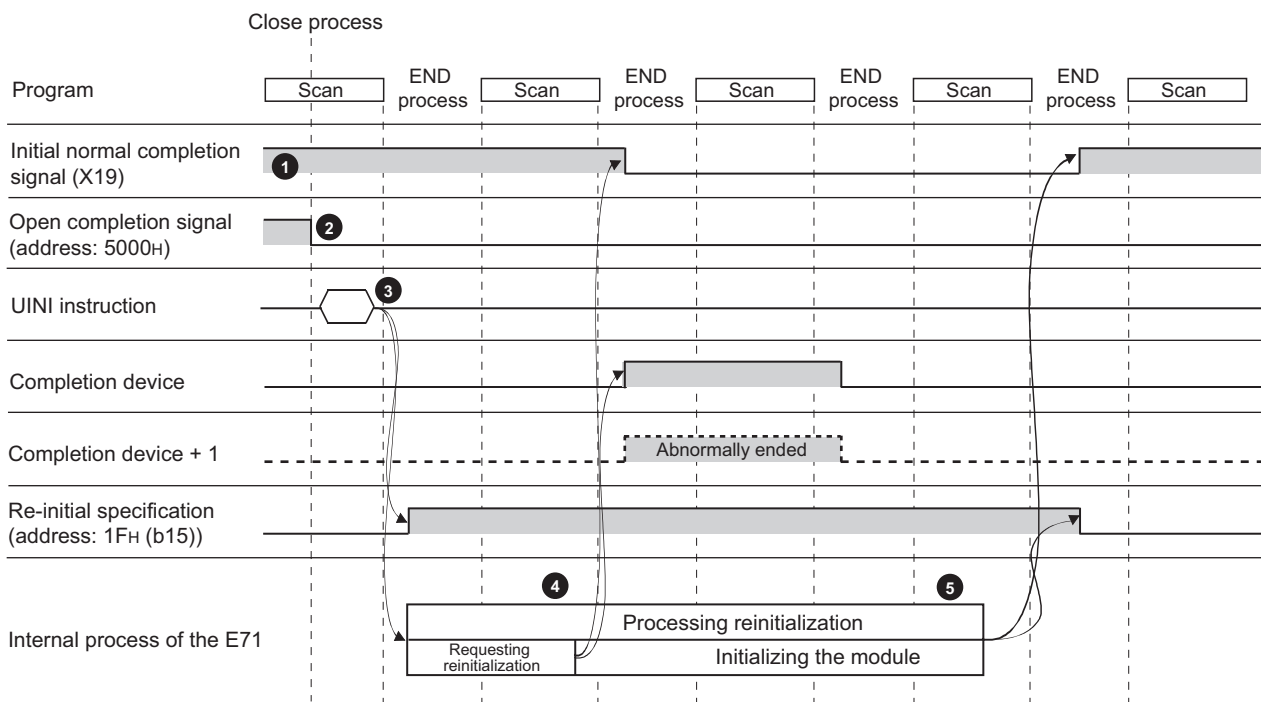
(d) To change the transmission speed and communication mode

By changing the transmission speed and communication mode, communications with the connected device can be resumed.

(2) Program for the reinitialization process

Perform the reinitialization process using a program. The following shows the process timing and sample programs.

(a) Process timing



- ① Check the normal completion of the initial process. (Initial normal completion signal (X19): ON)
- ② Terminate all data communications with the connected devices and perform a close process on all connections.
- ③ Execute the UINI instruction. Specify the parameters (such as the IP address of the own station and Ethernet operation setting) using the control data of the dedicated instruction and initialize the E71.
- ④ When the E71 reinitialization request is completed, Initial normal completion signal (X19) turns off.
- ⑤ When the reinitialization processes are all completed, Reinitialization specification (address: 1FH (b15)) turns "0" and Initial normal completion signal (X19) turns on. If the reinitialization process fails, an error code is stored in the initial error code storage area.

(b) Sample program

The following two methods are available:

- Using the dedicated instruction (UINI instruction)
- Writing directly to the buffer memory

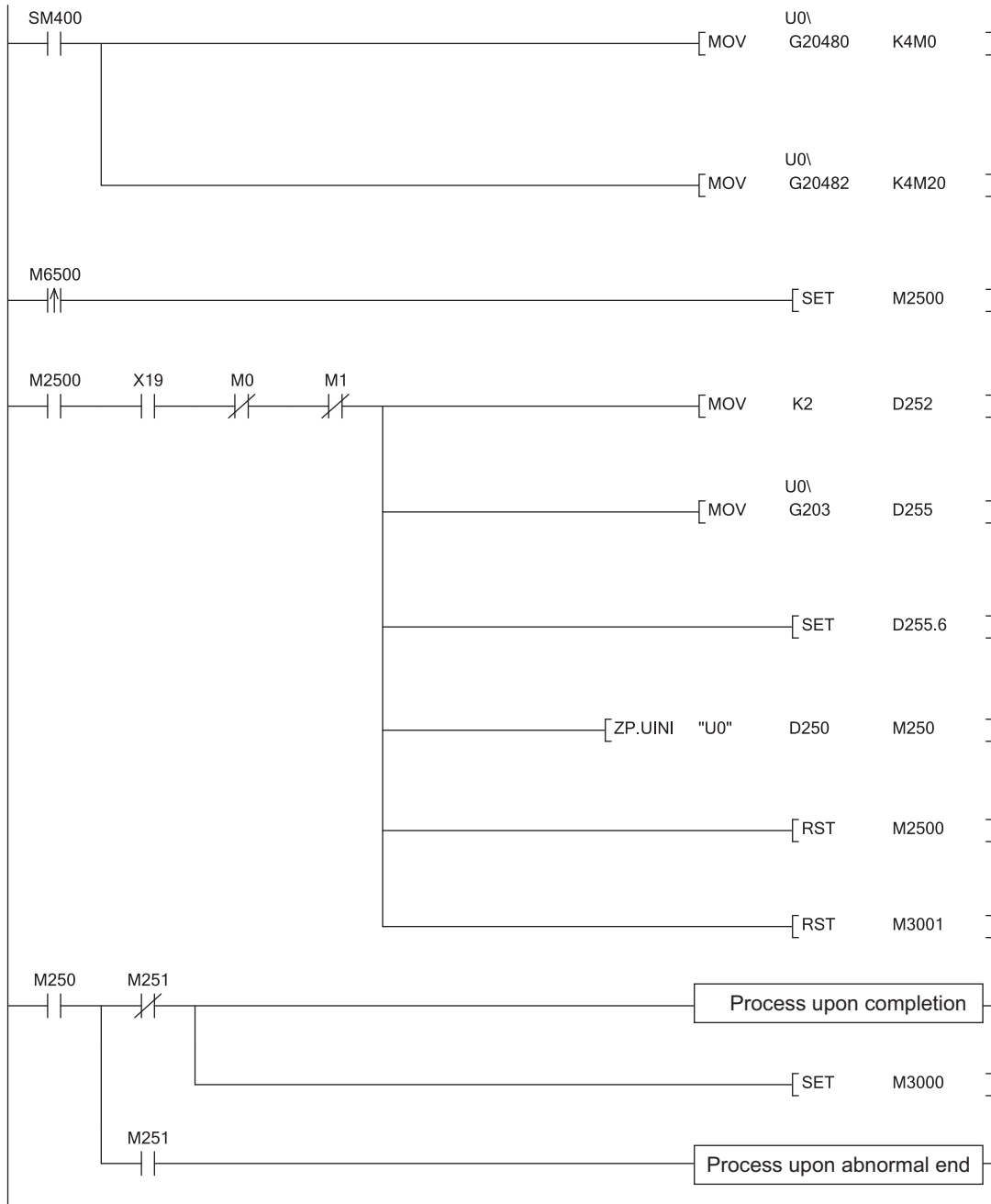


Do not use the method using the dedicated instruction (UINI instruction) and the method by writing directly to the buffer memory together.

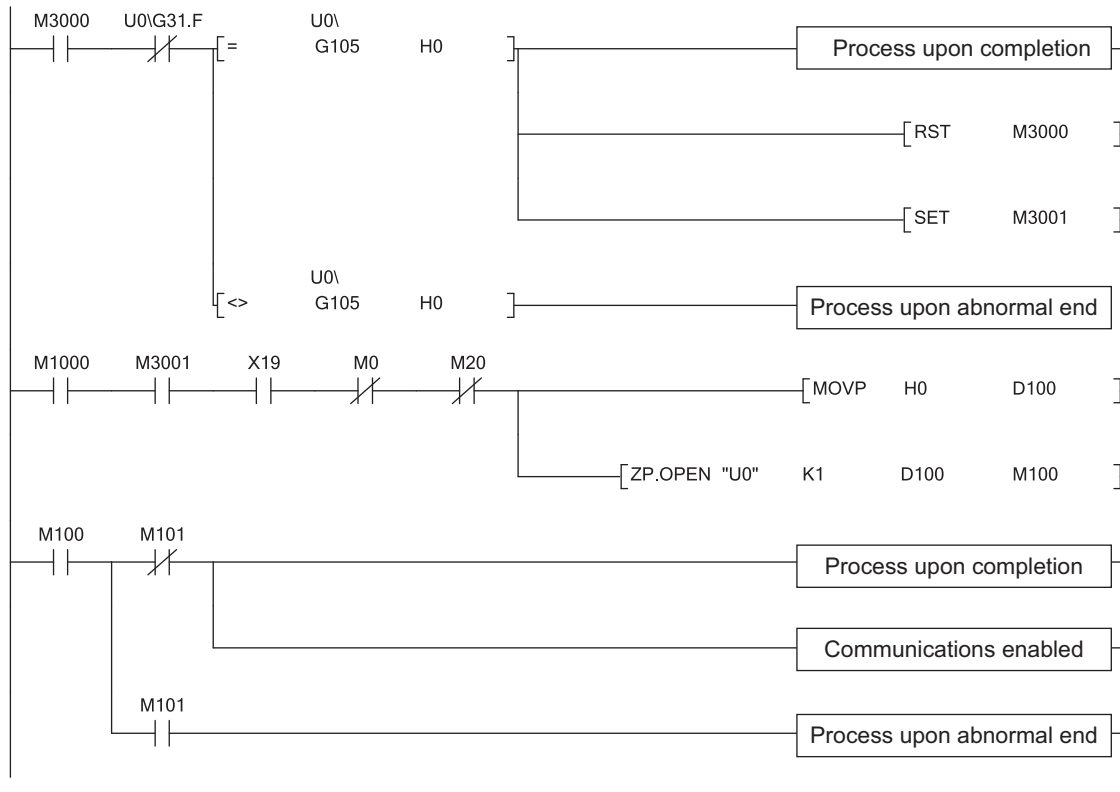
A

(c) Using the dedicated instruction (UINI instruction)

The following shows an example where the reinitialization process is performed using the UINI instruction and the OPEN instruction is executed after the process is completed. (When the I/O signals of the E71 are X/Y00 to X/Y1F)



(To the next page)

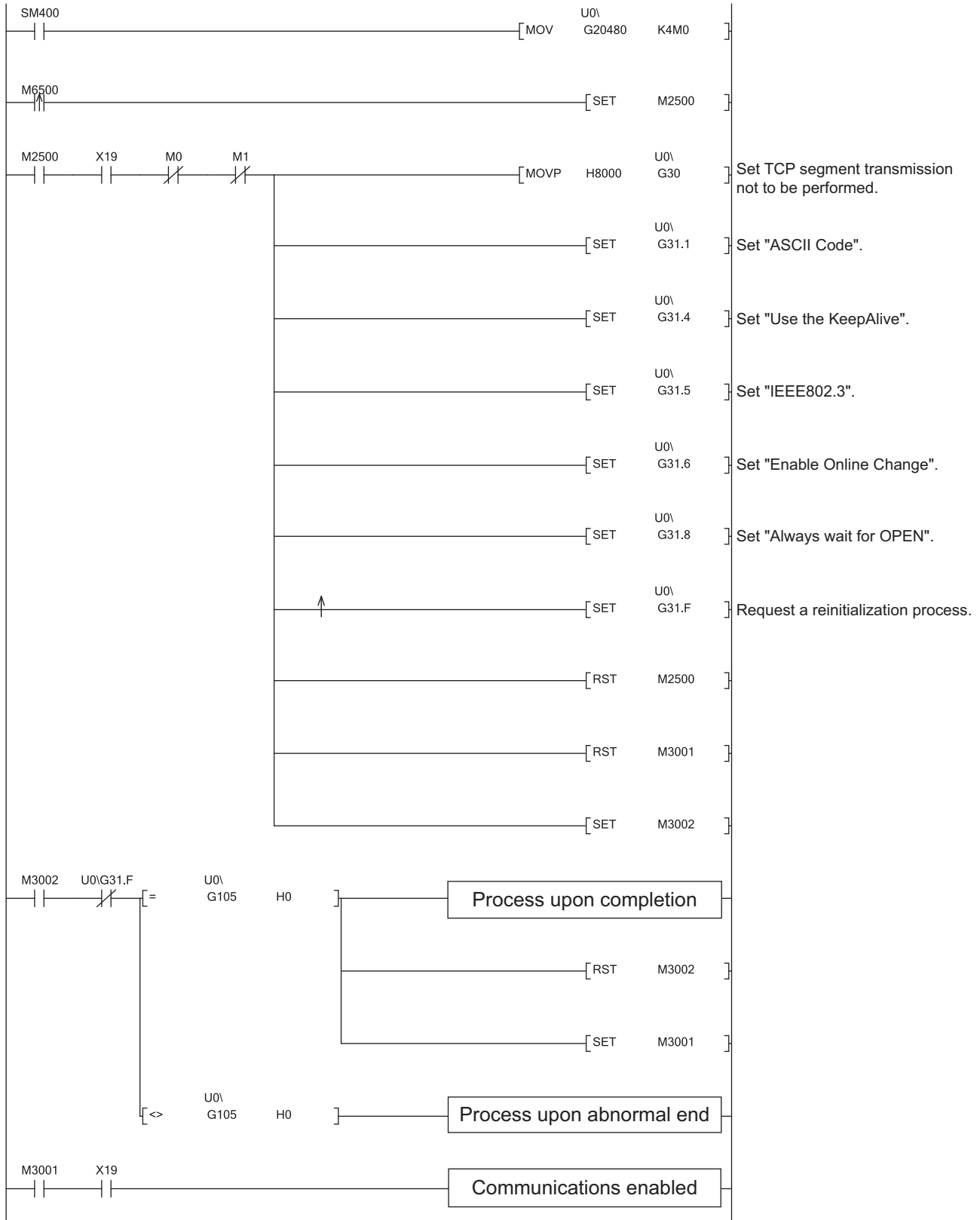


A

Appendix 3 Initial Process
 Appendix 3.2 Reinitialization process

(d) Writing directly to the buffer memory

The following shows an example where the stored values in the TCP Maximum Segment Transmission setting area and Communication condition setting (Ethernet Operation Setting) area is changed (when the I/O signals of the E71 are X/Y00 to X/Y1F). Create a program with a contact (such as a flag for user) indicating the status of the open completion signal storage area.



Remark

This is a sample program for communications using connection numbers 1 and 2. When using another connection number, specify the corresponding signals and bits.

(3) Precautions**(a) Buffer memory area settings**

The following parameters operate according to the settings of the corresponding buffer memory areas configured when a reinitialization process is performed. Therefore, do not change the settings of the buffer memory areas.

- Timer setting of the initial setting
- Open setting
- Router relay parameter setting
- Station number <-> IP information setting
- FTP parameters setting

(b) Reinitialization process request

Do not request another reinitialization process while a reinitialization process is in progress.

A

Appendix 4 Line Status Check

The E71 line status, parameter settings, and progress of the initial process can be checked. The following two methods can be used to check the line status.

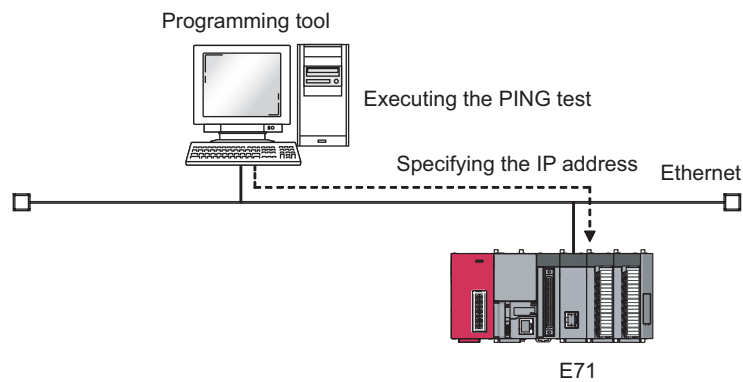
- PING test
- Loopback test

Appendix 4.1 PING test

This section describes the PING test.

(1) Performing a test with direct connection to the E71

This test is to check the existence of an E71 that has completed the initial process on the same Ethernet network using the programming tool or to check the existence of a connected device, such as a personal computer, having a specified IP address.



(a) Applications

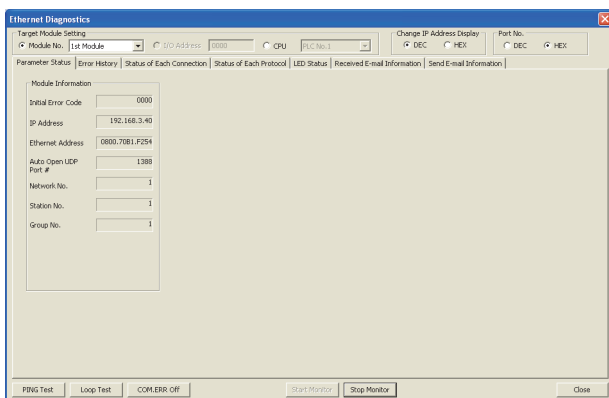
The line status between the programming tool (personal computer) and the E71 can be checked.

Point

When the connected device subjected to the PING test is an E71, do not specify the MELSOFT application transmission port (UDP/IP) as a valid port for the remote password check. The PING test cannot be executed.

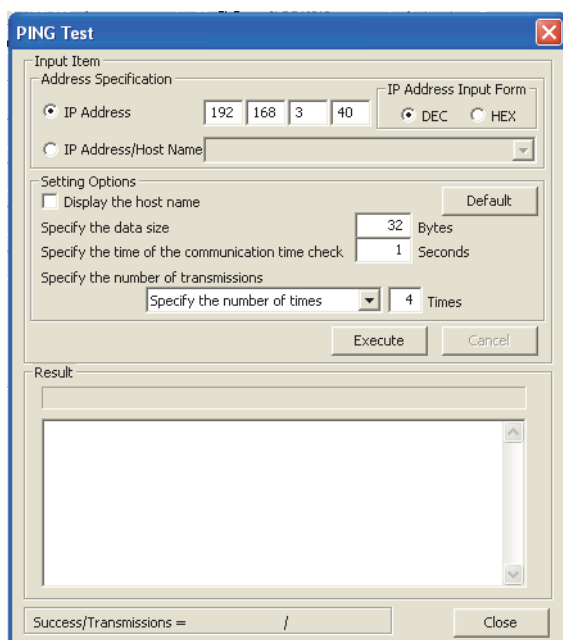
(b) Operating procedure

Use the following procedure.

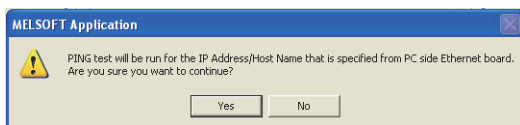


1. Click the **PING Test** button in the "Ethernet Diagnostics" window.

[Diagnostics] ⇒ [Ethernet Diagnostics] ⇒ "Target Module Setting" under "Module No."

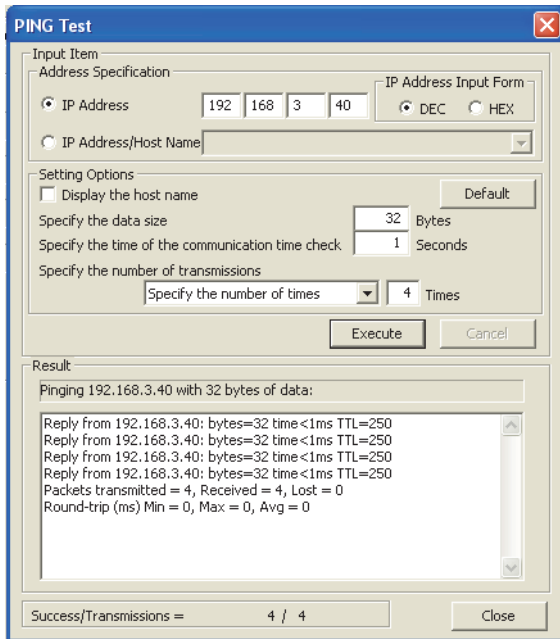


2. Configure the settings and click the **Execute** button.



3. Click the **Yes** button.

A



4. The test results are displayed.

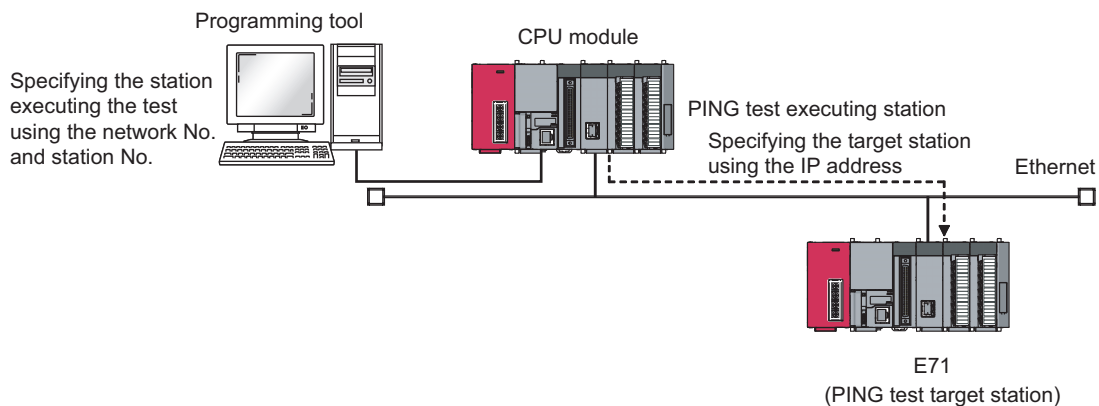
(c) Action for abnormal end

If the test fails, check the following and perform the test again.

- Whether the E71 is properly connected
- Connection to the Ethernet network
- Parameter settings written in the CPU module
- Operating status of the CPU module (whether or not an error has occurred)
- IP addresses set in the programming tool and the PING test target station
- Whether the connected device has been reset after the E71 was replaced

(2) Performing a test with direct connection to the CPU module

This test is to check the existence of an E71^{*1, *2} that has completed the initial process on the same Ethernet network where a station connected to the programming tool exists or to check the existence of a connected device, such as a personal computer, having a specified IP address, by directly connecting the programming tool and the CPU module.



*1 This also applies to Q/QnA/A series modules. However, for an AJ71E71, AJ71E71-B2, and AJ71E71-B5, software versions S and later are required.

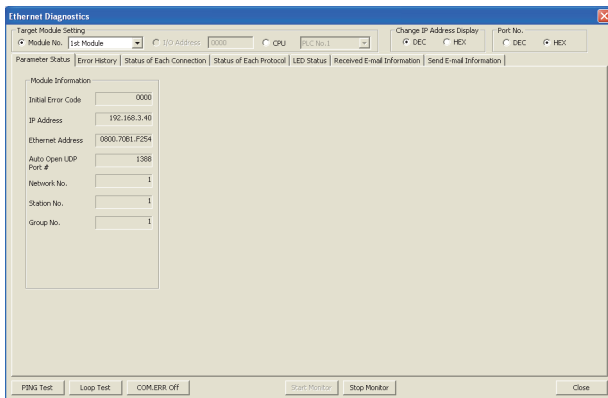
*2 The PING test cannot be executed on the own station.

(a) Applications

By specifying the execution station and the target station, the PING test can be executed from a remote device.

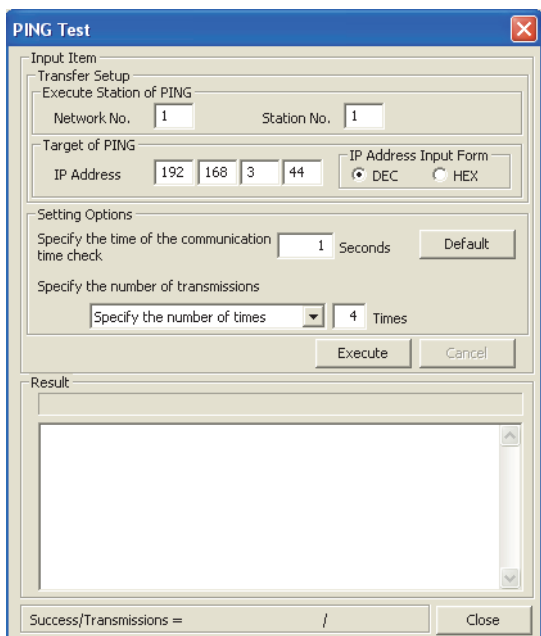
(b) Operating procedure

Use the following procedure.

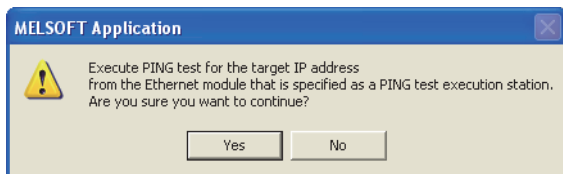


1. Click the **PING Test** button in the "Ethernet Diagnostics" window.

[Diagnostics] ⇨ [Ethernet Diagnostics] ⇨ "Target Module Setting" under "Module No."

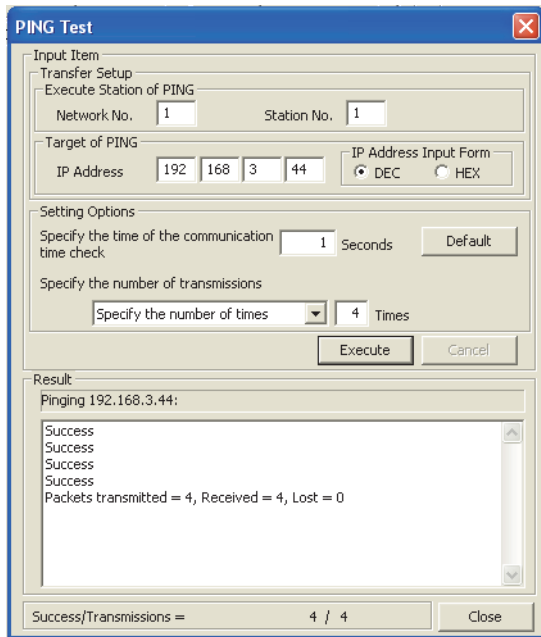


2. Configure the settings and click the **Execute** button.



3. Click the **Yes** button.





4. The test results are displayed.

(c) Action for abnormal end

If the test fails, take the same action as that for the method using direct connection to the E71. (Page 256, Appendix 4.1 (1))

(3) Performing a test with the PING command

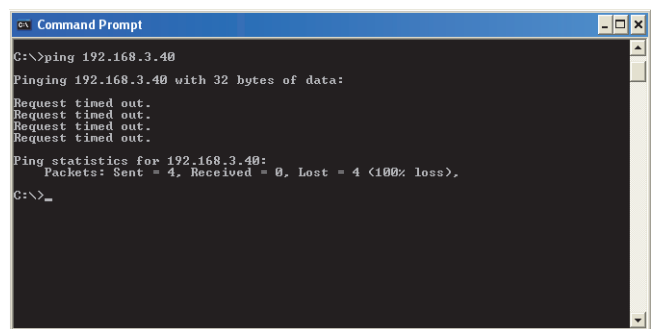
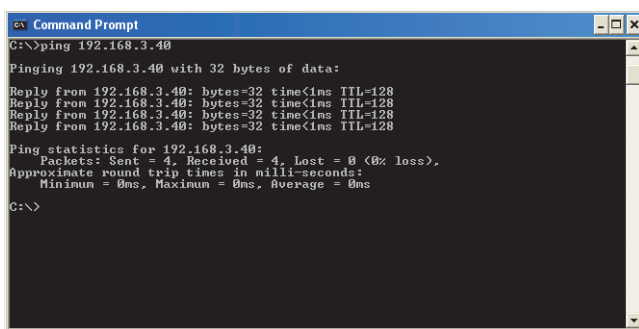
The PING test can be also executed using the PING command. The following shows an example of checking whether the initial process has been completed by issuing a PING command to the E71 of the own station from a connected device in the same Ethernet network.

[Specification method] ping IP Address

When the IP address of an E71 is 192.168.3.40

(Window upon normal completion)

(Window upon abnormal end)



(a) Action for abnormal end

If the test fails, check the following and perform the test again.

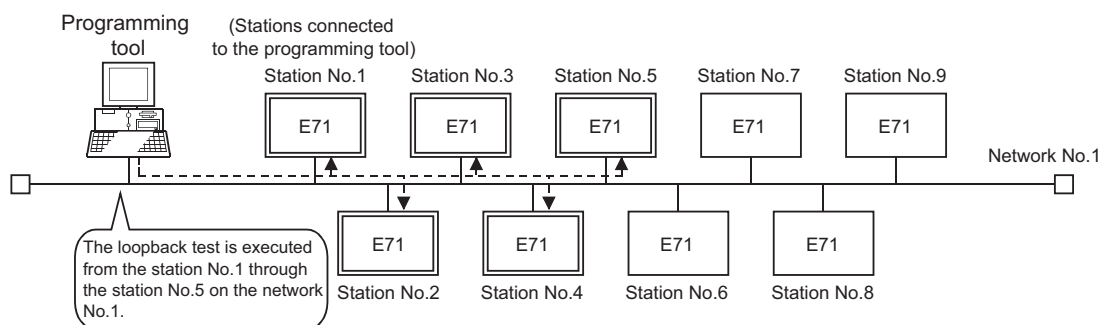
- Whether the E71 is properly connected
- Connection to the Ethernet network
- Parameter settings written in the CPU module
- Operating status of the CPU module (whether or not an error has occurred)
- IP address of the destination E71 specified using the PING command

Appendix 4.2 Loopback test

This section describes the loopback test.

(1) Performing a test using a programming tool

This test is to check whether the initial process for each module has been completed. The test is executed on the network of stations connected to the programming tool, and loopback test messages are sent sequentially to each E71*¹ on the network and in the station number range specified for the loopback test request designation.



*1 Because the QJ71E71-100 of function version A and QnA/A series modules do not have a function for responding to this request, the test results cannot be checked.

(a) Applications

The test can be executed only by specifying the network number for the test and target station number range.

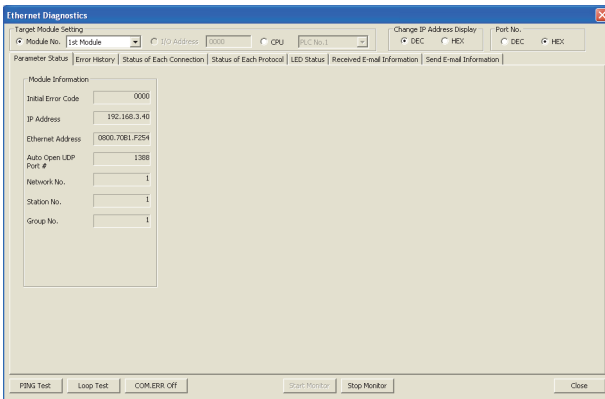
Point

For the E71-connected station subjected to the loopback test, do not specify the MELSOFT application transmission port (UDP/IP) as a valid port for the password check. The loopback test cannot be executed.


A

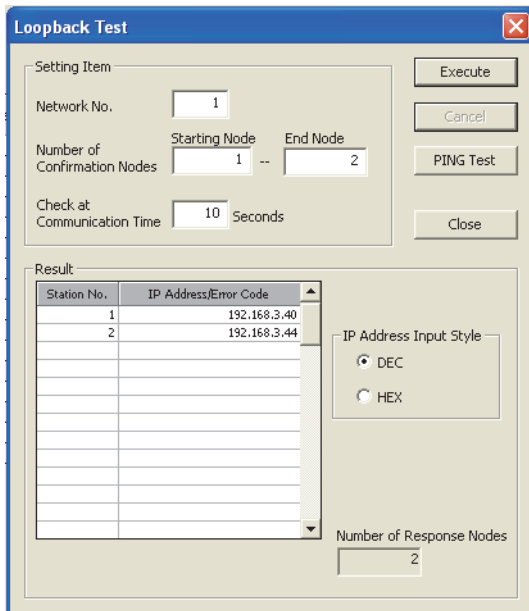
(b) Operating procedure

Use the following procedure.



1. Click the **Loop Test** button in the "Ethernet Diagnostics" window.

 [Diagnostics] ⇨ [Ethernet Diagnostics] ⇨ "Target Module Setting" under "Module No."



2. Configure the settings and click the **Execute** button to display the test results.

Point

If the same IP address or station number is assigned to multiple stations, only the results of the station that responds first are displayed.

(c) Action for abnormal end

If the loopback test fails, "No response" or an error code is displayed.

Displayed test result	Status of the target E71	Cause	Action
IP address	Initial processing normal completion status (INIT. LED turns on.)	Normal completion	-
No response	No error	The initial process for the target E71 has not been completed normally.	Correct the following parameter settings. <ul style="list-style-type: none"> • Basic setting • Ethernet operation setting • Initial setting
		There is an error, such as cable disconnection and line disconnection, in the line connection to the target E71.	<ul style="list-style-type: none"> • Check the cable.
		The IP address of the target E71 is incorrect. (The class or subnet address differs from that of the E71.)	Correct the set values of the Ethernet operation settings.
		The same IP address has been set to multiple target E71 modules.	
	The same network number or station number has been set to multiple target E71 modules.	Perform the PING test on the "No response" module. When the test is completed normally, correct the set values in the basic settings.	
	No error/Error exists	The Ethernet line is heavily loaded (including a case where an error equivalent to the error codes C030 _H and C031 _H is occurring).	When the Ethernet line is not heavily loaded, perform the test again.
Error exists	The routing parameter is not set (an error equivalent to the error code C080 _H is occurring).	Correct the set value of the routing parameter.	
Error code	No error	The MELSOFT application transmission port (UDP/IP) for the target E71 is locked with the remote password.	Disable the remote password setting and write the parameter to the CPU module.
		The target E71 is a function version A module.	Check the module name and function version.
	No error/Error exists	The Ethernet line is heavily loaded (including a case where an error equivalent to the error codes C030 _H and C031 _H is occurring).	When the Ethernet line is not heavily loaded, perform the test again.




For an E71 where the loopback test failed, check the error and take corrective action. Then restart the E71-connected station. Restarting the E71 executes the initial process. To check whether the initial process has been completed in the E71, perform the PING test. The PING test can be also performed on the "Loopback Test" window.

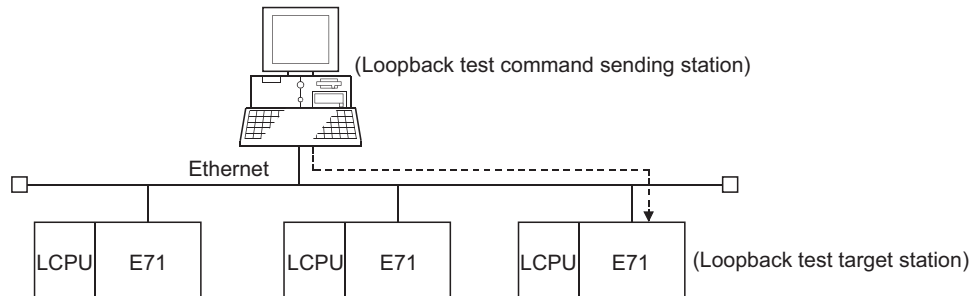
A

Appendix 4 Line Status Check
Appendix 4.2 Loopback test

(2) Performing a test using the MC protocol

The loopback test can be also executed using MC protocol communications. Note, however, that only the E71 in the own station can be used. The test cannot be executed on E71 modules in other stations over the network. For details, refer to the following.

 MELSEC-Q/L MELSEC Communication Protocol Reference Manual



(a) Execution method

Execute the test using the MC protocol dedicated command (Loopback test: 0619).

Point

When the user port on the E71 side is used to perform MC protocol communications, the line connection process is necessary. Execute the open process for the connection used for the E71 side.

Appendix 5 Self-Diagnostic Tests

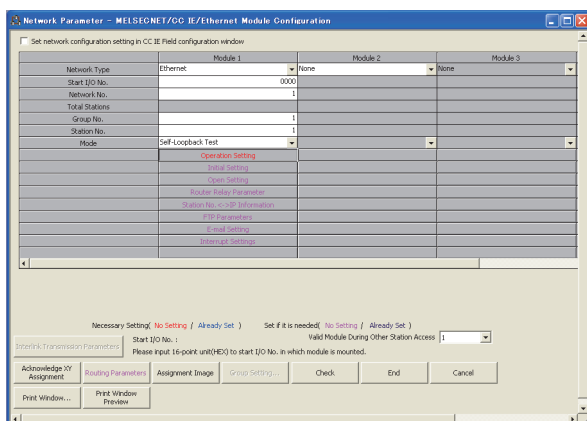
This section describes the self-diagnostic tests to check data communications and hardware of an E71.

Appendix 5.1 Self-loopback test

Hardware including the E71 communication circuit is checked. Send the test message destined to the own station of the E71 to the line. Then check whether the same message can be received over the network.

(1) Operating procedure

Use the following procedure.



1. Connect the E71 to the line.*1
2. Set the switch on the CPU module to "STOP".
3. Select "Self-Loopback Test" under "Mode" and write the parameter to the CPU module.

Project window ⇔ [Parameter] ⇔
[Network Parameter] ⇔ [Ethernet/CC IE
Field] ⇔ Select "Ethernet" under "Network Type".

4. Resetting the CPU module starts the test. Test time is approximately five seconds. The RUN and OPEN LEDs turn on during the test.
5. Check the LED of the E71 after five seconds. When the test is running normally, the RUN LED turns on. If the test fails, the RUN and ERR. LEDs turn on.

*1 When the line is not connected, the self-loopback test is not performed and ends normally.

If an error is detected in the self-loopback test, the following causes are possible. The error information is stored in Error log area (address: E5_H) in the buffer memory.

- E71 hardware error
- Ethernet line error

Point

There is no hardware-related problem even when the self-loopback test is executed while the destination is online. Because a packet collision occurs when there are many packets on the line, the test may fail or may not be completed in five seconds. In this case, execute the test after stopping data communications between other devices.

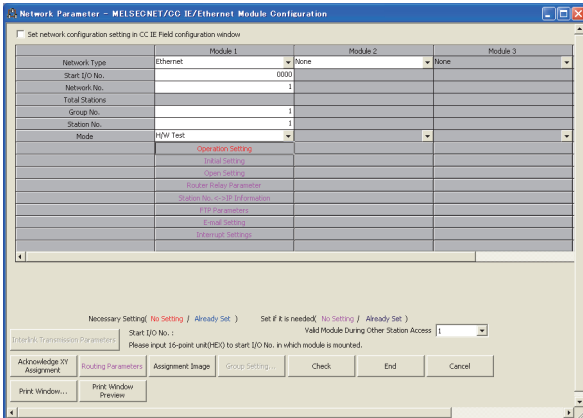
A

Appendix 5.2 Hardware test (H/W Test)

The RAM and ROM of an E71 are checked.

(1) Operating procedure

Use the following procedure.



1. Set the switch on the CPU module to "STOP".
2. Select "H/W Test" under "Mode" and write the parameter to the CPU module.

Project window ⇨ [Parameter] ⇨
[Network Parameter] ⇨ [Ethernet/CC IE
Field] ⇨ Select "Ethernet" under "Network Type".

3. Resetting the CPU module starts the test. Test time is approximately five seconds. The RUN and OPEN LEDs turn on during the test.
4. Check the LED of the E71 after five seconds. When the test is running normally, the RUN LED turns on. If the test fails, the RUN and ERR. LEDs turn on.

If an error is detected in the hardware test (H/W test), the following cause is possible. The error information is stored in Error log area (address: E5_H) in the buffer memory.

- E71 RAM/ROM error

Point

If the test fails, perform it again. If it fails again, there may be a hardware problem with the E71. Please consult your local Mitsubishi representative.

Appendix 6 Differences from Ethernet Modules of Other Series

This section describes the differences between an E71 and Ethernet modules of other series.

Appendix 6.1 Differences from a built-in Ethernet port LCPU

For comparison of the specifications and functions between an LJ71E71-100 and a Built-in Ethernet port LCPU, refer to the following.

 MELSEC-L CPU Module User's Manual (Built-In Ethernet Function)

Appendix 6.2 Differences from Q series modules

The functions of LJ71E71-100 are the same as those of QJ71E71-100, except for those listed below. A program of QJ71E71-100 can be also used.

○: Supported, ×: Not supported

Item	LJ71E71-100	QJ71E71-100
AUTO MDI/MDI-X compatibility (automatic detection of a straight cable and a cross cable)	○	×
Module error collection function	○	×
File password 32	○	×
Redundant system supporting function	×	○

Appendix 6.3 Differences from QnA/A series modules

This section provides the functional comparison between an E71 and QnA/A series modules and precautions for using the same program. The QnA/A series modules are the following products.

Series	Model name	Product name
A series	AJ71E71	AJ71E71, A1SJ71E71-B2, A1SJ71E71-B5
	AJ71E71-S3	AJ71E71-S3, A1SJ71E71-B2-S3, A1SJ71E71-B5-S3
	AJ71E71N	AJ71E71N3-T, AJ71E71N-B5, AJ71E71N-B2, AJ71E71N-T, AJ71E71N-B5T, A1SJ71E71N3-T, A1SJ71E71N-B5, A1SJ71E71N-B2, A1SJ71E71N-T, A1SJ71E71-B5T
QnA series	QE71(N)	AJ71QE71, AJ71QE71-B5, A1SJ71QE71-B2, A1SJ71QE71-B5, AJ71QE71N3-T, AJ71QE71N-B5, AJ71QE71N-B2, AJ71QE71N-T, AJ71QE71N-B5T, A1SJ71QE71N3-T, A1SJ71QE71N-B5, A1SJ71QE71N-B2, A1SJ71QE71N-T, A1SJ71QE71N-B5T

A

Appendix 6 Differences from Ethernet Modules of Other Series
Appendix 6.1 Differences from a built-in Ethernet port LCPU

(1) Functional comparison

The following table lists the functional comparison.

○: Supported, ×: Not supported

Function		AJ71E71	AJ71E71-S3, AJ71E71N	QE71(N)		E71
				9706 or earlier	9706B or later	
Initial process	Initial process with the program	○	○	○	○	○
	Initial process with parameter settings	×	×	×	○	○
Open process *1	Open process with the program	○	○	○	○	○
	Open process with parameter settings	×	×	×	×	○
Communications using a fixed buffer	Procedure Exist	○	○	○	○	○*2
	No Procedure	×	○	○	○	○*2
Communications using a random access buffer		○	○	○	○	○
MC protocol communications		○	○	○	○*3	○*3
Communications using data link instructions		×	×	×	○*4	○
Interrupt process (upon data receiving)	BUFRCVS instruction	×	×	×	×	○
	RCVSV instruction	×	×	×	×	○
E-mail function	Send/receive with the program	×	×	×	×	○
	Send with the auto notification function	×	×	×	×	○
File transfer (FTP server) function		×	×	×	○	○
Web function		×	×	×	×	○
Broadcast communications		×	○	○	○	○
Communications with the switch on the CPU module set to STOP		×	○	×	○	○
Selection of the communication data code (ASCII/binary)		○	○	○	○	○
CC-Link IE Controller Network, CC-Link IE Field Network, MELSECNET/H, MELSECNET/10 relay communications		×	×	×	○*4	○
Router relay function		×	○	○	○	○
Connected device alive check	Ping	×	○	○	○	○
	KeepAlive	×	×	×	×	○
Pairing open		×	○	○	○	○
Unit of each timer setting value for data communications	500ms	×*5	○	○ (Fixed)	○ (Fixed)	○ (Fixed)
	2s	○ (Fixed)	○	×	×	×
Connection to MELSOFT products	TCP/IP	×	×	×	×	○
	UDP/IP	×	○	○	○	○
Installation of EEPROM		×	×	○	○	×*6
TCP Maximum Segment Size Option transmission		×	○*7	×	○*7	○*8

*1 For the E71, the number of connections for the open process from the CPU module has increased to 16 connections.

*2 Compatible with the I/O signal and QE71(N) buffer memory

*3 Data of up to 960 words can be read from/written to an E71, and up to 480 words in a QE71(N).

*4 Usage depends on the CPU module and programming tool version.

*5 This applies to a module with the software version of Q or earlier.

*6 No EEPROM is installed. Items registered in the EEPROM of a QE71(N) are set using the parameter settings of the programming tool.

*7 This applies to a module with the software version of E or later.

*8 If the E71 cannot communicate data when combined with a certain connected device, the setting can be changed in the buffer memory (address: 30(1E_H)). (Page 33, Section 3.5.2) After changing the setting, execute the reinitialization process.



The response performance of the E71 to connected devices is faster than that of the A/QnA series modules. When the E71 is used, the compatibility with the A/QnA series modules cannot be precisely maintained. If there is a problem due to the performance of the connected device, create a timing similar to that of the existing system using the CPU module constant scan setting.

(2) Precautions for using the same program

The QnA/A series modules used for data communications between a CPU module and the connected device on the Ethernet network can be replaced by an E71. This section describes precautions for using the same program used in QnA/A series modules.

(a) AJ71E71(-S3) and AJ71E71N

- Program of connected devices

The following communication function programs of connected devices for an AJ71E71(-S3) and AJ71E71N can be used for communications with an E71. However, because the response speed of the AJ71E71(-S3) and AJ71E71N is different from that of the E71, the program may not be used as is. Before using the same program, check the operation.

- : Communications allowed by using the program for an AJ71E71(-S3) and AJ71E71N on the connected device side
 -: Not allowed to use

Function	Connected device →E71	E71→ connected device	AJ71E71(-S3), AJ71E71N→E71	E71→AJ71E71(-S3), AJ71E71N
Communications using a fixed buffer ("Procedure Exist")	○	○	○	○
Communications using a random access buffer	○	-	-	-
Data read/write in the CPU module ^{*1}	○	-	-	-

*1 Only A-compatible 1E frame commands can be used for data communications. To perform data communications using a command other than the A-compatible 1E frame commands, create a new program.

- Program for an AJ71E71(-S3) and AJ71E71N in the own station
 Because the assignment of the buffer memory areas for an E71 is different from that for an AJ71E71(-S3) and AJ71E71N, a program for the AJ71E71(-S3) and AJ71E71N cannot be used for an E71. Create a new program by referring to chapters that explain each function.
- Program using an I/O signal
 Programs cannot be used together with parameter settings on the programming tool.

(b) QE71(N)

Program of connected devices

Programs of the connected devices for a QE71(N) can be used for communications with the E71 except for the following programs.

- Program for file operation-related commands
- Program to access the data link system (A QCPU (Q Mode) cannot be connected to MELSECNET (II).)

However, because the response speed of an E71 is different from that of a QE71(N), the program may not be used as is. Before using the same program, check the operation.

Program for a QE71(N) in the own station

When using a program for the initial process and the termination process, do not write the parameters of the E71 (network parameters) set on the programming tool to the LCPU. Note the following when not using the parameters of an E71 set on the programming tool.

- Because a QE71(N) operates with all of its communication condition setting switches off, set the communication conditions using the reinitialization process.
- When MELSOFT products (such as a programming tool) are directly connected to an E71, the MELSOFT products (such as a programming tool) cannot access an LCPU.

To set the parameters of an E71 using the programming tool, delete the programs for the initial process and the termination process.

Programs for the QE71(N) in the own station can be used for communications with an E71 except for the following programs.

- Program to access the data link system
- Program related to the EEPROM
- Connection number 8 pairing open setting
- Parameter setting program using the EPRSET instruction

However, because the response speed of an E71 is different from that of a QE71(N), the program may not be used as is. Before using the same program, check the operation.

Program using an I/O signal

Programs cannot be used together with parameter settings on the programming tool.

Point

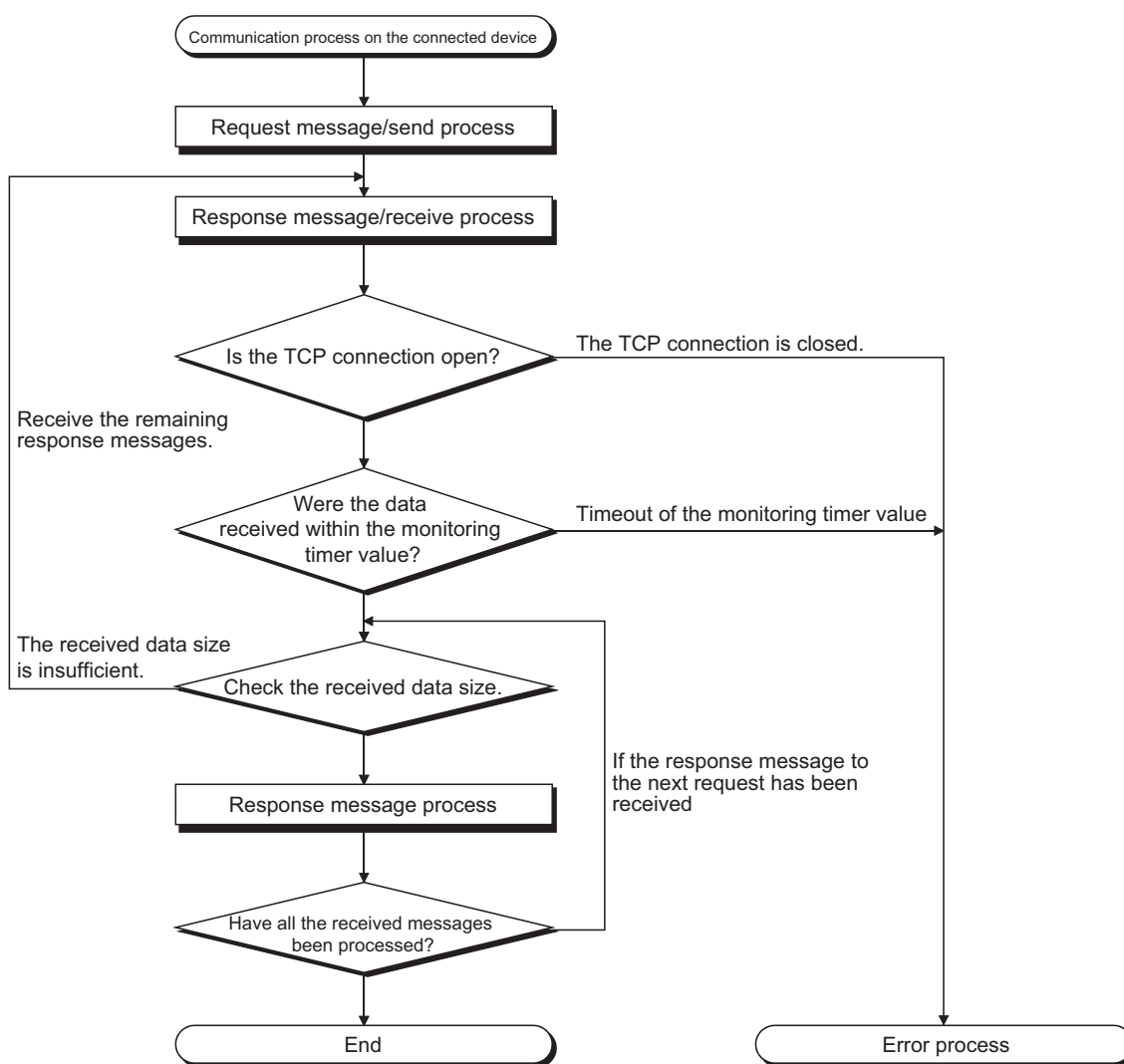
- For an E71, set the operation mode and communication conditions using the programming tool. Settings cannot be configured using a switch unlike QnA/A series modules. (There is no switch for settings.)
 - For an E71, after the Passive open process is executed, the open request cannot be cancelled before open completion. After the open completion, execute the close process.
-

Appendix 7 Sample Program on the Connected Device Side

The following shows a sample program for the connected device on the same Ethernet network as the E71. Each program example describes only the minimum programming that is required to perform a communication test. Change the IP address, port number and other values according to the system configuration used. Other processes such as a process performed upon an error can be added.

(1) Receive process of the connected device

A receive processing example on the connected device side is shown below.



Point

For Ethernet communications, the TCP socket functions are used inside the personal computer. However, these functions do not have any limit. Therefore, when the "send" function is executed once to transmit data, the receiving side needs to execute the "recv" function once or more to read the data ("send" and "recv" are not proportional to 1:1 execution). For this reason, the receive process above is required.

(2) When the receive process of the connected device is not supported

When the receive process of the connected device is not the one shown in (1), the following may occur when "Enable TCP Maximum Segment Size Option transmission" is set for communications.

- When batch reading is executed from the connected device using the MC protocol, data cannot be normally read.
- After replacing the E71 (which does not support the TCP Maximum Segment Size Option transmission function) with the alternative module supporting the function, data cannot be normally read.
- Even though the value in Received TCP packet count (Address: 1B8_H and 1B9_H) in the buffer memory was changed, data cannot be received.

If these errors occur, select "Disable TCP Maximum Segment Size Option transmission".

Appendix 7.1 When Visual C++[®].NET is used

This section describes the program execution environment, data communication details, and a sample program for when Visual C++[®].NET is used on the program of a connected device.

(1) Execution environment of the program example

(a) CPU module side

Item		Description
A model name of the LCPU in the E71-connected station		L02CPU
E71 I/O signal		X/Y00 to X/Y1F
E71 IP address		C0.00.01.FD _H (192.00.01.253)
E71 port number		2000 _H
Programming tool setting	Ethernet operation setting	Refer to Page 274, Appendix 7.1 (3) (a).
	Open setting	Refer to Page 274, Appendix 7.1 (3) (b).

(b) Connected device side

Item	Description
Operation environment	Microsoft [®] Windows [®] XP Professional Operating System Ver.2002 Service Pack2
Ethernet interface board model name	WINSOCK compatible board
Library to link	WSOCK32.LIB
Software development environment	Visual C++ [®] .NET 2003 manufactured by Microsoft [®] Corporation
MAC address	This setting is not required because the ARP function can be used.
IP address	Received upon an Active open
Port number	Received upon an Active open

(c) Communication protocol

TCP/IP is used.

(2) Outline of the program example

(a) Program on the CPU module side

Because parameters are set on the programming tool, no program is required.

(b) Program on the connected device side

Read/write data from/to the CPU module using the library described above.

- Write in word units (for five points from D0 to D4)
- Read in word units (for five points from D0 to D4)

A

(3) Programming tool setting

Set the parameters using the programming tool as shown below.

(a) Ethernet operation setting

Ethernet Operation Setting

Communication Data Code
 Binary Code
 ASCII Code

Initial Timing
 Do not wait for OPEN (Communications impossible at STOP time)
 Always wait for OPEN (Communication possible at STOP time)

IP Address Setting
 Input Format: DEC
 IP Address: 192 0 1 253

Send Frame Setting
 Ethernet(V2.0)
 IEEE802.3

Enable Online Change

TCP Existence Confirmation Setting
 Use the KeepAlive
 Use the Ping

End Cancel

(b) Open setting

Network Parameter Ethernet Open Setting Module No.: 1

IP Address/Port No. Input Format: HEX

	Protocol	Open System	Fixed Buffer	Fixed Buffer Communication Procedure	Pairing Open	Existence Confirmation	Host Station Port No.	Destination IP Address	Destination Port No.
1	TCP	Unpassive	Send	Procedure Exist	Disable	No Confirm	2000		
2									
3									
4									
5									
6									
7									
8									
9									
10									
11									
12									
13									
14									
15									
16									

(*) IP Address and Port No. will be displayed by the selected format.
 Please enter the value according to the selected number.

End Cancel

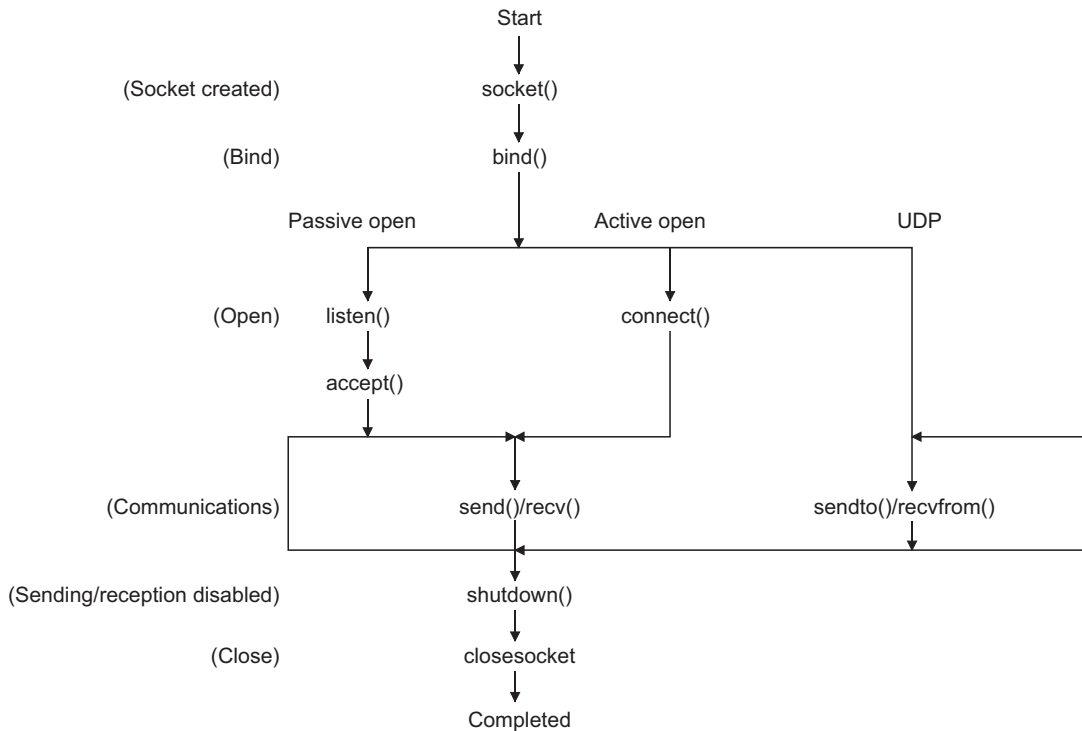
(4) Sample program on the connected device side

The program example of the connected device accessing the L02CPU in the E71-connected station is described. When this program is executed, the contents of the following communication messages are displayed in order:

- Batch write command message in word units
- Batch write response message in word units
- Batch read command message in word units
- Batch read response message in word units

Remark

- The following is an outline of how to compile a program created using Microsoft® Corporation Visual C++®.NET.
 - Start Visual C++®.NET.
 - Create a project. From [File]→[New]→[Project], select ".NET" in "Project Types" and "Empty Project" in "Templates", and set the project name (e.g. AJSAMP) and location.
 - Create a source file. Open Solution Explorer. Then right-click Source Files and select [Add]→[Add New Item]. Set the file name (e.g. AJSAMP.cpp) and location, and create a program according to the program example.
 - From the project setting window, get WSOCK32.LIB linked. Open Solution Explorer. Then right-click the project name (AJSAMP) and select [Properties]→[Configuration Properties]→[Linker]→[Command Line]. Type WSOCK32.LIB in Additional Options and press the OK button.
 - On the Build menu, click Build Solution to create an execution file (AJSAMP.EXE).
 - Exit Visual C++®.NET.
 - Execute AJSAMP.EXE.
- The following is an outline of how to implement a socket routine call.



A

Appendix 7 Sample Program on the Connected Device Side
Appendix 7.1 When Visual C++®.NET is used

```

/*****/
/**   Sample program (program name: AJSAMP.CPP)   **/
/**                                           **/
/**   This program is a sample program to conduct a   **/
/**   connection test between the E71 and target device. **/
/**   This program accesses the data register (D) of   **/
/**   the PLC CPU installed together with the E71     **/
/**                                           **/
/**   Copyright(C) 2005 Mitsubishi Electric Corporation **/
/**   All Rights Reserved                         **/
/*****/

#include <stdio.h>
#include <winsock.h>

#define FLAG_OFF          0           // Completion flag OFF
#define FLAG_ON           1           // Completion flag ON
#define SOCK_OK           0           // Normal completion
#define SOCK_NG           -1          // Abnormal completion
#define BUF_SIZE         4096        // Receive buffer size

#define ERROR_INITIAL     0           // Initial error
#define ERROR_SOCKET      1           // Socket creation error
#define ERROR_BIND        2           // Bind error
#define ERROR_CONNECT     3           // Connection error
#define ERROR_SEND        4           // Send error
#define ERROR_RECEIVE     5           // Receive error
#define ERROR_SHUTDOWN    6           // Shutdown error
#define ERROR_CLOSE       7           // Line close error

//Definitions for checking the receiving sizes
//#define RECV_ANS_1  4 // Receiving size of response message in reply to device write (1E frame)
#define RECV_ANS_1  22 // Receiving size of response message in reply to device write (3E frame)
//#define RECV_ANS_2  24 // Receiving size of response message in reply to device read (1E frame)
#define RECV_ANS_2  42 // Receiving size of response message in reply to device read (3E frame)

typedef struct sck_inf{
    struct in_addr my_addr;
    unsigned short my_port;
    struct in_addr aj_addr;
    unsigned short aj_port;
}sck_inf;

```

(To the next page)

```

int nErrorStatus;           // Error information storage variable
int Dmykeyin;              // Dummy key input
int Closeflag;            // Connection completion flag
int socketno;

int main()
{
    WORD wVersionRequested=MAKEWORD(1,1);           // Winsock Ver 1.1 request
    WSADATA wsaData;
    int length;                                     // Communication data length
    unsigned char s_buf[BUF_SIZE];                 // Send buffer
    unsigned char r_buf[BUF_SIZE];                 // Receive buffer
    int rbuf_idx;                                   // Receive data storage head index
    int rcv_size;                                   // Number of receive data
    struct sck_inf sc;
    struct sockaddr_in hostdata;                    // External device side data
    struct sockaddr_in aj71e71;                     // E71 side data
    void Sockerror(int);                            // Error handling function

    unsigned long ulCmdArg ;                        // Non-blocking mode setting flag

    sc.my_addr.s_addr=htonl(INADDR_ANY);            // External device side IP address
    sc.my_port=htons(0);                            // External device side port number
    sc.aj_addr.s_addr=inet_addr("192.0.1.253");     // E71 side IP address (C00001FDH)
    sc.aj_port=htons(0x2000);                       // E71 side port number

    Closeflag=FLAG_OFF;                             // Connection completion flag off

    nErrorStatus=WSAStartup(wVersionRequested,&wsaData); // Winsock Initial processing

    if(nErrorStatus!=SOCK_OK) {
        Sockerror(ERROR_INITIAL);                   // Error handling
        return(SOCK_NG);
    }

    printf("Winsock Version is %ld.%ld\n",HIBYTE(wsaData.wVersion), LOBYTE(wsaData.wVersion));
    printf("AJ_test Start\n");

    socketno=socket(AF_INET,SOCK_STREAM,0);         // Create socket for TCP/IP

    if(socketno==INVALID_SOCKET){
        Sockerror(ERROR_SOCKET);                     // Error handling
        return(SOCK_NG);
    }
}

```

(To the next page)

A

```

hostdata.sin_family=AF_INET;
hostdata.sin_port=sc.my_port;
hostdata.sin_addr.s_addr=sc.my_addr.s_addr;

if(bind(socketno,(LPSOCKADDR)&hostdata,sizeof(hostdata))!=SOCK_OK){
    // Bind
    Sockerror(ERROR_BIND); // Error handling
    return(SOCK_NG);
}

aj71e71.sin_family=AF_INET;
aj71e71.sin_port=sc.aj_port;
aj71e71.sin_addr.s_addr=sc.aj_addr.s_addr;

if(connect(socketno,(LPSOCKADDR)&aj71e71,sizeof(aj71e71))!=SOCK_OK){
    // Connection (Active open)
    Sockerror(ERROR_CONNECT); // Error handling
    return(SOCK_NG);
}

Closeflag=FLAG_ON; // Connection completion flag ON

// Set to non-blocking mode
ulCmdArg = 1;
ioctlsocket(socketno, FIONBIO, &ulCmdArg); // Set to non-blocking mode

// strcpy((char*)(s_buf), "03FF000A4420000000000500112233445566778899AA");
// D0 to D4 batch write request (1E frame)
strcpy((char*)(s_buf),"500000FF03FF00002C000A1401000D
*0000000005112233445566778899AA");
// D0 to D4 batch write request (3E frame)

length = strlen((char*)(s_buf));

if(send(socketno, (char*)(s_buf), length, 0) == SOCKET_ERROR) {
    // Data sending
    Sockerror(ERROR_SEND); // Error handling
    return (SOCK_NG);
}
printf("\n send data\n%s\n",s_buf);

```

(To the next page)


```

// Perform receiving size check and receiving processing simultaneously
rbuf_idx = 0; // Receive data storage head index initialization
recv_size = 0; // Initialize the number of receive data
while(1) {
    length = recv(socketno, (char*) (&r_buf[rbuf_idx]), (BUF_SIZE - rbuf_idx), 0);
                                                    // Response data receiving

    if(length == 0) { // Is connection cut off?
        Sockerror(ERROR_RECEIVE); // Error handling
        return (SOCK_NG);
    }

    if(length == SOCKET_ERROR) {
        nErrorStatus = WSAGetLastError();
        if(nErrorStatus != WSAEWOULDBLOCK) {
            Sockerror(ERROR_RECEIVE); // Error handling
            return (SOCK_NG);
        } else {
            continue; // Repeat until messages are received
        }
    } else {
        rbuf_idx += length; // Update the receive data storage
                            // position

        recv_size += length; // Update the number of receive data
        if(recv_size >= RECV_ANS_1) // Have all response messages been
                                    // received?
            break; // Stop repeating as messages have
                    // been received
    }
}
r_buf[rbuf_idx] = '\0'; // Set NULL at the end of receive data

printf("\n receive data\n%s\n",r_buf);

// strcpy((char *) (s_buf), "01FF000A442000000000500"); // D0 to D4 batch read request
// (1E frame)
strcpy((char *) (s_buf), "500000FF03FF000018000A04010000D*0000000005");
// D0 to D4 batch read request
// (3E frame)

length = strlen((char *) (s_buf));

if(send(socketno, (char *) (s_buf), length, 0) == SOCKET_ERROR) {
    // Data sending
    Sockerror(ERROR_SEND); // Error handling
    return (SOCK_NG);
}
printf("\n send data\n%s\n",s_buf);

```

(To the next page)

```

// Perform receiving size check and receiving processing simultaneously
rbuf_idx = 0; // Receive data storage head index
// initialization
recv_size = 0; // Initialize the number of receive data
while(1) {
    length = recv(socketno, (char*)&r_buf[rbuf_idx], (BUF_SIZE - rbuf_idx), 0);
    // Response data receiving
    if(length == 0) { // Is connection cut off?
        Sockerror(ERROR_RECEIVE); // Error handling
        return (SOCK_NG);
    }

    if(length == SOCKET_ERROR) {
        nErrorStatus = WSAGetLastError();
        if(nErrorStatus != WSAEWOULDBLOCK) {
            Sockerror(ERROR_RECEIVE); // Error handling
            return (SOCK_NG);
        } else {
            continue; // Repeat until messages are received
        }
    } else {
        rbuf_idx += length; // Update the receive data storage
        // position
        recv_size += length; // Update the number of receive data
        if(recv_size >= RECV_ANS_2) // Have all response messages been
        // received?
            break; // Stop repeating as messages have
        // been received
    }
}
r_buf[rbuf_idx] = '\0'; // Set NULL at the end of receive data

printf("receive data\n%s\n", r_buf);

if(shutdown(socketno,2)!=SOCK_OK){ // Processing to disable
    // sending/receiving
    Sockerror(ERROR_SHUTDOWN); // Error handling
    return(SOCK_NG);
}
if(closesocket(socketno)!=SOCK_OK){ // Close processing
    // Error handling
    Sockerror(ERROR_CLOSE);
    return(SOCK_NG);
}

Closeflag=FLAG_OFF; // Connection completion flag off
WSACleanup(); // Release Winsock.DLL

```

(To the next page)

```

printf("\nAJ_test End.\n\n Normally completed. \n");
printf("Press any key to exit the program.\n");
Dmykeyin=getchar(); // Wait for key input

return(SOCK_OK);
}

void Sockerror(int error_kind) // Error handling function
{
    if(error_kind==ERROR_INITIAL){
        printf("Initial processing is abnormal.");
    }
    else{
        nErrorStatus=WSAGetLastError();
        switch(error_kind){
            case ERROR_SOCKET:
                printf("Failed to create socket.");
                break;
            case ERROR_BIND:
                printf("Failed to bind.");
                break;
            case ERROR_CONNECT:
                printf("Failed to establish connection.");
                break;
            case ERROR_SEND:
                printf("Sending failed.");
                break;
            case ERROR_RECEIVE:
                printf("Receiving failed.");
                break;
            case ERROR_SHUTDOWN:
                printf("Failed to shutdown.");
                break;
            case ERROR_CLOSE:
                printf("Failed to close normally.");
                break;
        }
    }
    printf("Error code is %d.\n", nErrorStatus);
    if(Closeflag==FLAG_ON){
        nErrorStatus=shutdown(socketno,2);           // Shutdown processing
        nErrorStatus=closesocket(socketno);         // Close processing
        Closeflag=FLAG_OFF;                         // Connection completion flag off
    }

    printf("Press any key to exit the program.\n");
    Dmykeyin=getchar();                             // Wait for a key input
    WSACleanup();                                   // Release Winsock.DLL
    return;
}

```

Appendix 7.2 When Visual Basic®.NET is used

This section describes the program execution environment, data communication details, and a sample program for when Visual Basic®.NET is used on the program of the connected device.

(1) Execution environment of the program example

(a) CPU module side

Item		Description
A model name of the LCPU in the E71-connected station		L02CPU
E71 I/O signal		X/Y00 to X/Y1F
E71 IP address		C0.00.01.FD _H (192.00.01.253)
E71 port number		2000 _H
Programming tool setting	Ethernet operation setting	Refer to Page 283, Appendix 7.2 (3) (a).
	Open setting	Refer to Page 283, Appendix 7.2 (3) (b).

(b) Connected device side

Item	Description
Operation environment	Microsoft® Windows® XP Professional Operating System Ver.2002 Service Pack2
Ethernet interface board model name	WINSOCK compatible board
Software development environment	Visual Basic®.NET 2003 manufactured by Microsoft® Corporation
MAC address	This setting is not required because the ARP function can be used.
IP address	An arbitrary number is assigned.
Port number	An arbitrary number is assigned.

(c) Communication protocol

TCP/IP is used.

(2) Outline of the program example

(a) Program on the CPU module side

Because parameters are set on the programming tool, no program is required.

(b) Program on the connected device side

Data (D0 to D4) in the CPU module are read out.

(3) Programming tool setting

Set the parameters using the programming tool as shown below.

(a) Ethernet operation setting

(b) Open setting

	Protocol	Open System	Fixed Buffer	Fixed Buffer Communication Procedure	Pairing Open	Existence Confirmation	Host Station Port No.	Destination IP Address	Destination Port No.
1	TCP	Unpassive	Send	Procedure Exist	Disable	No Confirm	2000		
2									
3									
4									
5									
6									
7									
8									
9									
10									
11									
12									
13									
14									
15									
16									

(*) IP Address and Port No. will be displayed by the selected format.
Please enter the value according to the selected number.

A

Appendix 7 Sample Program on the Connected Device Side
Appendix 7.2 When Visual Basic® NET is used

(4) Sample program on the connected device side

The program example of the connected device accessing the L02CPU in the E71-connected station is described. In this program, data of D0 to D4 (five points) in the L02CPU in the E71-connected station are read with an A-compatible 1E frame command (01: Batch read in word units). The following are basic operation procedures:

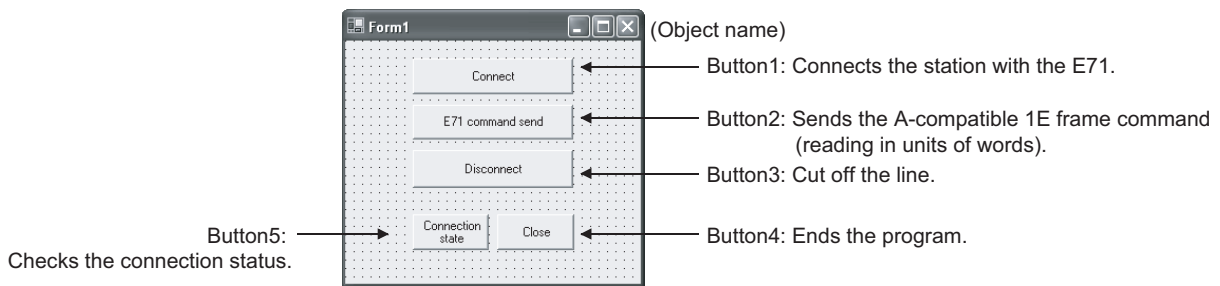
- Sending a command with the line disconnected (Connect the line. After completing the connection, the E71 sends a command.)
- Reconnecting the line with it connected (Disconnect the line. After the disconnection is completed, reconnect it.)

Remark

The following is an outline of how to compile a program created using Microsoft® Corporation Visual Basic®.NET.

- Start Visual Basic®.NET.
- Create a project. From [File]→[New]→[Project], select "Visual Basic Project" in "Project Types" and "Windows Application" in "Templates", and set the project name (e.g. AJSAMP) and location.
- Create a form and a program. Use the toolbox "Button" to create the sample window shown below (Form1.vb) and create a program referring to the sample program.
- On the Build menu, click Build Solution to create an execution file (AJSAMP.EXE).
- Exit the Visual Basic®.NET.
- Execute the AJSAMP.EXE.

[Window example (Form 1.vb)]



[Sample program (Form 1.vb)]

```
Option Strict Off
Option Explicit ON
```

```
Imports System
Imports System.Text
Imports System.Net
```

```
Friend Class Form1
```

```
    Inherits System.Windows.Forms.Form
```

```
#Region "Windows Form Designer generated code"
```

```
    Public Sub New()
```

```
        MyBase.New()
```

```
        If m_vb6FormDefInstance Is Nothing Then
```

```
            If m_InitializingDefInstance Then
```

```
                m_vb6FormDefInstance = Me
```

```
            Else
```

```
                Try
```

```
                    'For the start-up form, the first instance created is the default instance.
```

```
                    If System.Reflection.Assembly.GetExecutingAssembly().EntryPoint
```

```
                        .DeclaringType Is Me.GetType Then
```

```
                            m_vb6FormDefInstance = Me
```

```
                        EndIf
```

```
                    Catch
```

```
                    End Try
```

```
            End If
```

```
        End If
```

```
        ' This call is required by the Windows form designer.
```

```
        InitializeComponent()
```

```
    End Sub
```

(To the next page)

A

```

'Form overrides dispose to clean up the component list.
Protected Overloads Overrides Sub Dispose(ByVal Disposing As Boolean)
    If Disposing Then
        If Not components Is Nothing Then
            components.Dispose()
        End If
    End If
    MyBase.Dispose(Disposing)
End Sub

```

```

'Required by the Windows Form Designer.
Private components As System.ComponentModel.IContainer
Public WithEvents Command5 As System.Windows.Forms.Button
Public WithEvents Command4 As System.Windows.Forms.Button
Public WithEvents Command3 As System.Windows.Forms.Button
Public WithEvents Command2 As System.Windows.Forms.Button
Public WithEvents Command1 As System.Windows.Forms.Button
Dim Ajsoc As Sockets.Socket
Private State As Boolean = False

```

```

'NOTE: The following procedure is required by the Windows Form Designer.
'It can be modified using the Windows Form Designer.
'Do not modify it using the code editor.

```

```

<System.Diagnostics.DebuggerStepThrough(> Private Sub InitializeComponent()
    Me.Command5 = New System.Windows.Forms.Button
    Me.Command4 = New System.Windows.Forms.Button
    Me.Command3 = New System.Windows.Forms.Button
    Me.Command2 = New System.Windows.Forms.Button
    Me.Command1 = New System.Windows.Forms.Button
    Me.SuspendLayout()
    '
    'Command5
    '
    Me.Command5.BackColor = System.Drawing.SystemColors.Control
    Me.Command5.Cursor = System.Windows.Forms.Cursors.Default
    Me.Command5.ForeColor = System.Drawing.SystemColors.ControlText
    Me.Command5.Location = New System.Drawing.Point(64, 152)
    Me.Command5.Name = "Command5"
    Me.Command5.RightToLeft = System.Windows.Forms.RightToLeft.No
    Me.Command5.Size = New System.Drawing.Size(72, 32)
    Me.Command5.TabIndex = 4
    Me.Command5.Text = "Connection status"
    '
    'Command4
    '

```

(To the next page)


```

Me.Command4.BackColor = System.Drawing.SystemColors.Control
Me.Command4.Cursor = System.Windows.Forms.Cursors.Default
Me.Command4.ForeColor = System.Drawing.SystemColors.ControlText
Me.Command4.Location = New System.Drawing.Point(144, 152)
Me.Command4.Name = "Command4"
Me.Command4.RightToLeft = System.Windows.Forms.RightToLeft.No
Me.Command4.Size = New System.Drawing.Size(73, 32)
Me.Command4.TabIndex = 3
Me.Command4.Text = "Close"
'
'Command3
'
Me.Command3.BackColor = System.Drawing.SystemColors.Control
Me.Command3.Cursor = System.Windows.Forms.Cursors.Default
Me.Command3.ForeColor = System.Drawing.SystemColors.ControlText
Me.Command3.Location = New System.Drawing.Point(64, 96)
Me.Command3.Name = "Command3"
Me.Command3.RightToLeft = System.Windows.Forms.RightToLeft.No
Me.Command3.Size = New System.Drawing.Size(152, 33)
Me.Command3.TabIndex = 2
Me.Command3.Text = "disconnect"
'
'Command2
'
Me.Command2.BackColor = System.Drawing.SystemColors.Control
Me.Command2.Cursor = System.Windows.Forms.Cursors.Default
Me.Command2.ForeColor = System.Drawing.SystemColors.ControlText
Me.Command2.Location = New System.Drawing.Point(64, 56)
Me.Command2.Name = "Command2"
Me.Command2.RightToLeft = System.Windows.Forms.RightToLeft.No
Me.Command2.Size = New System.Drawing.Size(152, 31)
Me.Command2.TabIndex = 1
Me.Command2.Text = "Sending a E71 command"
'
'Command1
'
Me.Command1.BackColor = System.Drawing.SystemColors.Control
Me.Command1.Cursor = System.Windows.Forms.Cursors.Default
Me.Command1.ForeColor = System.Drawing.SystemColors.ControlText
Me.Command1.Location = New System.Drawing.Point(64, 16)
Me.Command1.Name = "Command1"
Me.Command1.RightToLeft = System.Windows.Forms.RightToLeft.No
Me.Command1.Size = New System.Drawing.Size(152, 31)
Me.Command1.TabIndex = 0
Me.Command1.Text = "connect"

```

(To the next page)

```

    '
    'Form1
    '
    Me.AutoScaleBaseSize = New System.Drawing.Size(5, 12)
    Me.BackColor = System.Drawing.SystemColors.Control
    Me.ClientSize = New System.Drawing.Size(280, 214)
    Me.Controls.Add(Me.Command5)
    Me.Controls.Add(Me.Command4)
    Me.Controls.Add(Me.Command3)
    Me.Controls.Add(Me.Command2)
    Me.Controls.Add(Me.Command1)
    Me.Cursor = System.Windows.Forms.Cursors.Default
    Me.Location = New System.Drawing.Point(329, 189)
    Me.Name = "Form1"
    Me.RightToLeft = System.Windows.Forms.RightToLeft.No
    Me.StartPosition = System.Windows.Forms.FormStartPosition.Manual
    Me.Text = "Form1"
    Me.ResumeLayout(False)
End Sub
#End Region

#Region "Upgrade Wizard support code"
Private Shared m_vb6FormDefInstance As Form1
Private Shared m_InitializingDefInstance As Boolean
Public Shared Property DefInstance() As Form1
    Get
        If m_vb6FormDefInstance Is Nothing OrElse m_vb6FormDefInstance.IsDisposed Then
            m_InitializingDefInstance = True
            m_vb6FormDefInstance = New Form1()
            m_InitializingDefInstance = False
        End If
        DefInstance = m_vb6FormDefInstance
    End Get
    Set
        m_vb6FormDefInstance = Value
    End Set
End Property
#End Region

```

(To the next page)

```

Private Sub Command1_Click(ByVal eventSender As System.Object, ByVal eventArgs
As System.EventArgs) Handles Command1.Click
    'Connect to the Ethernet interface module.
    Dim sock As New Sockets.Socket(Sockets.AddressFamily.InterNetwork, _
Sockets.SocketType.Stream, Sockets.ProtocolType.Tcp)
    Asock = sock
    Dim ip As IPAddress = Dns.Resolve("192.0.1.253").AddressList(0)
    Dim ipend As IPEndPoint = New IPEndPoint(ip, "8192")

    Me.Asock.Connect(ipend)
    MsgBox("Connection Completed")
    State = Me.Asock.Connected()

End Sub

Private Sub Command2_Click(ByVal eventSender As System.Object, ByVal eventArgs
As System.EventArgs) Handles Command2.Click
    Dim SData As Byte()
    Dim RData(256) As Byte

    'Rend D0 to D4 (5 points) with the A-compatible 1E frame command.
    SData = Encoding.ASCII.GetBytes("01FF000A4420000000000500")
    'Read D0 to D4 (5 points) with the QnA-compatible 3E frame command.
    'SData = Encoding.ASCII.GetBytes("500000FF03FF000018000A04010000D
*0000000005")

    'Send the data.
    Me.Asock.Send(SData)
    MsgBox("Send completion", MsgBoxStyle.Information)

    'Read the response from the PLC CPU.
    Me.Asock.Receive(RData)
    MsgBox(Encoding.ASCII.GetString(RData), MsgBoxStyle.Information)

End Sub

```

(To the next page)

```
Private Sub Command3_Click(ByVal eventSender As System.Object, ByVal eventArgs
As System.EventArgs) Handles Command3.Click
    'Close the TCP (UDP) connection socket (disconnect the line).
    Me.Ajsock.Shutdown(Net.Sockets.SocketShutdown.Both)
    Me.Ajsock.Close()
    MsgBox("The disconnection was successful", MsgBoxStyle.Information)
    State = Me.Ajsock.Connected()
End Sub
```

```
Private Sub Command4_Click(ByVal eventSender As System.Object, ByVal eventArgs
As System.EventArgs) Handles Command4.Click
    'End the program.
    End
End Sub
```


```
Private Sub Command5_Click(ByVal eventSender As System.Object, ByVal eventArgs
As System.EventArgs) Handles Command5.Click
    'Check the connection state.
    If State Then
        MsgBox("Connected")
    Else
        MsgBox("Closed")
    End If
End Sub
```

```
End Class
```

Appendix 8 Checking the Serial Number, Function Version, and MAC address


(1) Checking the serial number and function version

The serial number and function version can be checked on the rating plate or with the system monitor. Refer to the following.

 MELSEC-L CPU Module User's Manual (Hardware Design, Maintenance and Inspection)

(2) Checking the MAC address

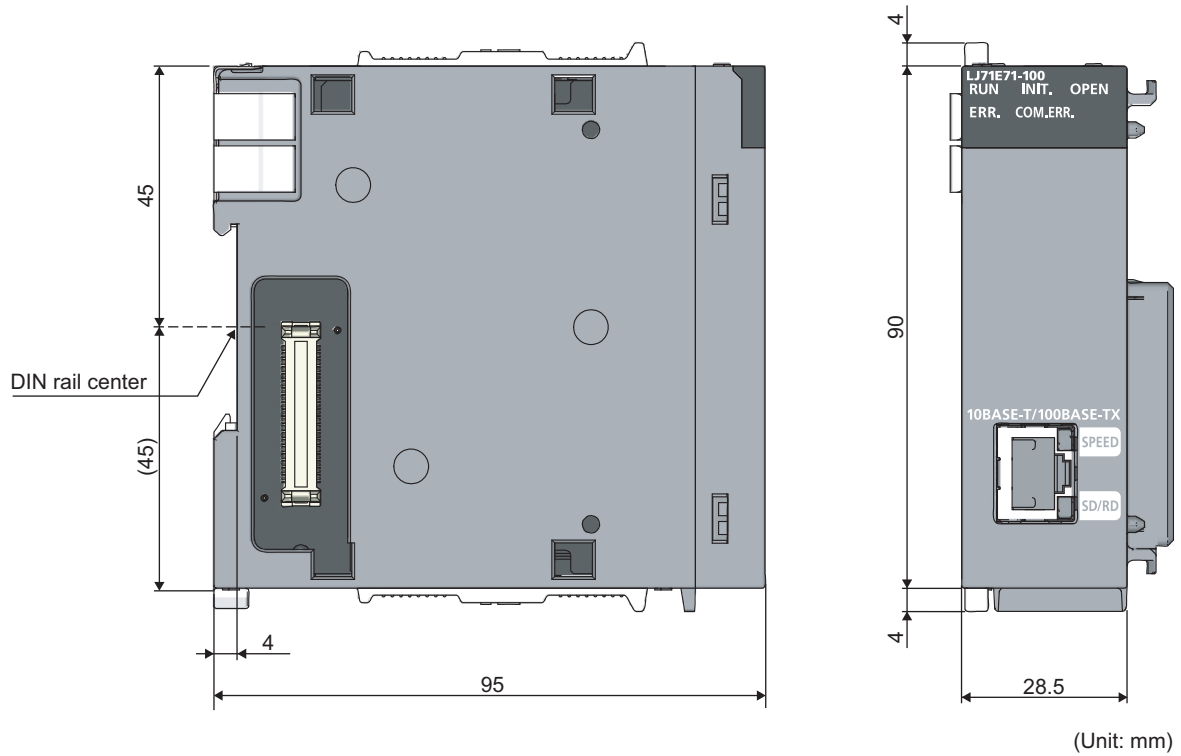
The MAC address can be checked on the rating plate or with the Ethernet diagnostics. For how to check the MAC address on the rating plate, refer to the following.

 MELSEC-L CPU Module User's Manual (Hardware Design, Maintenance and Inspection)

To use the Ethernet diagnostics, refer to Page 237, Section 14.8 in this manual.

Appendix 9 External Dimension Diagram

This appendix provides the external dimension diagrams of an E71.



Appendix 10 ASCII Code List

The following is the ASCII code table.

		Higher 4 bits									
Hexadecimal		0	1	2	3	4	5	6	7	8-F	
Binary		0000	0001	0010	0111	0100	0101	0110	0111	1000-1111	
Lower 4 bits	0	0000	NUL	DLE	(SP)	0	@	P	`	p	
	1	0001	SOH	DC1	!	1	A	Q	a	q	
	2	0010	STX	DC2	"	2	B	R	b	r	
	3	0011	ETX	DC3	#	3	C	S	c	s	
	4	0100	EOT	DC4	\$	4	D	T	d	t	
	5	0101	ENQ	NAK	%	5	E	U	e	u	
	6	0110	ACK	SYN	&	6	F	V	f	v	
	7	0111	BEL	ETB	'	7	G	W	g	w	
	8	1000	BS	CAN	(8	H	X	h	x	
	9	1001	HT	EM)	9	I	Y	i	y	
	A	1010	LF	SUB	*	:	J	Z	j	z	
	B	1011	VT	ESC	+	;	K	[k	{	
	C	1100	FF	FS	,	<	L	\(¥)	l		
	D	1101	CR	GS	-	=	M]	m	}	
	E	1110	SO	RS	.	>	N	^	n	~	
	F	1111	SI	US	/	?	O	_	o	DEL	

A

INDEX

0 to 9

100BASE-TX connection	57
100Mbps hub	57
10BASE-T connection	58
10Mbps hub	58
4E frame	85

A

Abnormal codes	213
A-compatible 1E frame	85
Active open	73
Always wait for OPEN (Communication possible at STOP time)	66
ARP	16
ASCII Code	293
Auto-open UDP port	245

B

Basic functions	28
Basic setting	64
Broadcast Communications	107
Buffer Memory	32
BUFRCV	155
BUFRCVS	155
BUFSND	155

C

CC-Link IE Controller Network, CC-Link IE Field Network, MELSECNET/H, MELSECNET/10 relay communications	29
CLOSE	155
Communication Data Code	65
COMMUNICATIONS USING A FIXED BUFFER	87
COMMUNICATIONS USING A RANDOM ACCESS BUFFER	120
Communications Using an Auto-open UDP Port	139
Communications using data link instructions	29
COMPLIANCE WITH THE EMC AND LOW VOLTAGE DIRECTIVES	6
CONNECTING MELSOFT PRODUCTS AND A GOT	81
Connection target	82
Cross cable	57

D

Data receiving using an interrupt program	95
Data receiving using the main program	92
Data transmission speed	26
DEDICATED INSTRUCTIONS	155
Default router IP address	137
Device	17
Do not wait for OPEN (Communications impossible at STOP time)	66

E

E-mail function	29
Enable Online Change	65
End codes	210
ERRCLR	155
ERRRD	155
Ethernet Diagnostics	237
Ethernet Operation Setting	65
External Dimension Diagram	292

F

File transfer (FTP server) function	29
Fixed Buffer	97
Fixed Buffer Communication Procedure	97
FTP	16
Fullpassive	75

G

General Specifications	26
Group No.	64
GX Developer	16
GX Works2	16

H

H/W Test	64
Hardware test	266
How to Turn Off the COM.ERR. LED	240
HTTP	16
Hub Connection Status Monitor Function	150

I

ICMP	16
Initial Process	246
Initial Timing	65,66
Input Format	65
Installation	59
Intelligent function module	17
Interface	26
Internal current consumption (5VDC)	26
Interrupt pointer setting	98
Interrupt settings	99
IP	16
IP Address	65
IP Address in Use Detection Function	151
IP Address Setting	65

L

List of buffer memory addresses	33
List of Dedicated Instructions	155
List of I/O Signals	30
Logical Addresses	133
Loopback test	261

M

MAC address	16
MC protocol	16
MC PROTOCOL COMMUNICATIONS	84
MELSECNET/10	16
MELSECNET/H	16
MELSECNET/H remote I/O station	16
MELSOFT application transmission port (TCP)	245
MELSOFT application transmission port (UDP)	245
MELSOFT connection	82
Mode	64
MRECV	155
MSEND	155
MX Component	16

N

Network Components	56
Network No.	64
Network Parameter	63
Network Type	64
No Procedure	87
Number of occupied I/O points	26

O

Offline	64
Online	64
OPEN	155
Open Setting	67
Open System	67

P

PACKING LIST	18
Pairing Open	105
Parameter list	63
PART NAMES	24
Passive open	75
Performance Specifications	26
Physical Addresses	133
PING test	256
POP3	16
Port Numbers Used for the E71	245
Procedure Exist	87
PROCEDURES BEFORE OPERATION	54
Programming tool	17

Q

QnA-compatible 3E frame	85
-------------------------	----

R

READ	155
RECV	155
RCVSV	155
Reinitialization process	250
Remote Password	140
Remote password check	142
REQ	155
RJ45 connector	57

Router information	137
Router IP address	139
Router Relay Function	135

S

Self-Diagnostic Tests	265
Self-loopback test	64,265
SEND	155
Send Frame Setting	65
Shielded twisted pair cable (STP)	57
SMTP	16
Special functions	29
SREAD	155
Start I/O No.	64
Station No.	64
Straight cable	57
Subnet address	137
Subnet mask	17
Subnet mask pattern	136
SWRITE	155
SYSTEM CONFIGURATION	56

T

TCP Existence Confirmation Setting	65
TCP/IP Communications	70
Transmission specifications	26
TROUBLESHOOTING	189

U

UDP/IP Communications	78
UINI	155
Unlock process	141
Unpassive	75
Unshielded twisted pair cable (UTP)	58

W

Web function	29
When using data link instructions	156
Wiring	60
WRITE	155

Z

Z.BUFRCSV	173
ZNRD	155
ZNWR	155
ZP.BUFRCSV	169
ZP.BUFSND	165
ZP.CLOSE	162
ZP.ERRCLR	175
ZP.ERRRD	179
ZP.OPEN	158
ZP.UINI	182

REVISIONS

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

Print Date	*Manual Number	Revision
February 2013	SH(NA)-081105ENG-A	First Edition

Japanese Manual Version SH-081104-A

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

©2013 MITSUBISHI ELECTRIC CORPORATION

WARRANTY

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

1. Gratis Warranty Term and Gratis Warranty Range

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

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

[Gratis Warranty Term]

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

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

[Gratis Warranty Range]

- (1) The range shall be limited to normal use within the usage state, usage methods and usage environment, etc., which follow the conditions and precautions, etc., given in the instruction manual, user's manual and caution labels on the product.
- (2) Even within the gratis warranty term, repairs shall be charged for in the following cases.
 1. Failure occurring from inappropriate storage or handling, carelessness or negligence by the user. Failure caused by the user's hardware or software design.
 2. Failure caused by unapproved modifications, etc., to the product by the user.
 3. When the Mitsubishi product is assembled into a user's device, Failure that could have been avoided if functions or structures, judged as necessary in the legal safety measures the user's device is subject to or as necessary by industry standards, had been provided.
 4. Failure that could have been avoided if consumable parts (battery, backlight, fuse, etc.) designated in the instruction manual had been correctly serviced or replaced.
 5. Failure caused by external irresistible forces such as fires or abnormal voltages, and Failure caused by force majeure such as earthquakes, lightning, wind and water damage.
 6. Failure caused by reasons unpredictable by scientific technology standards at time of shipment from Mitsubishi.
 7. Any other failure found not to be the responsibility of Mitsubishi or that admitted not to be so by the user.

2. Onerous repair term after discontinuation of production

- (1) Mitsubishi shall accept onerous product repairs for seven (7) years after production of the product is discontinued.
Discontinuation of production shall be notified with Mitsubishi Technical Bulletins, etc.
- (2) Product supply (including repair parts) is not available after production is discontinued.

3. Overseas service

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

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

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

5. Changes in product specifications

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

Microsoft, Windows, Windows NT, and Windows Vista are registered trademarks of Microsoft Corporation in the United States and other countries.

Pentium is a trademark of Intel Corporation in the United States and other countries.

Ethernet is a trademark of Xerox Corporation.

All other company names and product names used in this manual are trademarks or registered trademarks of their respective companies.

MELSEC-L Ethernet Interface Module User's Manual

Basic

MODEL	LJ71E71-U-KI-E
MODEL CODE	13JZ73
SH(NA)-081105ENG-A(1302)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.

HEADQUARTERS		EUROPEAN REPRESENTATIVES		EUROPEAN REPRESENTATIVES		EUROPEAN REPRESENTATIVES	
MITSUBISHI ELECTRIC EUROPE B.V. German Branch Gothaer Straße 8 D-40880 Ratingen Phone: +49 (0)2102 / 486-0 Fax: +49 (0)2102 / 486-1120	EUROPE	GEVA Wiener Straße 89 AT-2500 Baden Phone: +43 (0)2252 / 85 55 20 Fax: +43 (0)2252 / 488 60	AUSTRIA	ALFATRADE Ltd. 99, Paola Hill Malta- Paola PLA 1702 Phone: +356 (0)21 / 697 816 Fax: +356 (0)21 / 697 817	MALTA	TOO Kazpromavtomatika Ul. Zhambyla 28 KAZ-100017 Karaganda Phone: +7 7212 / 50 10 00 Fax: +7 7212 / 50 11 50	KAZAKHSTAN
MITSUBISHI ELECTRIC EUROPE B.V.-org.sl. Czech Branch Avenir Business Park, Radlická 714/113a CZ-158 00 Praha 5 Phone: +420 - 251 551 470 Fax: +420 - 251-551-471	CZECH REP.	TECHNIKON Oktyabrskaya 19, Off. 705 BY-220030 Minsk Phone: +375 (0)17 / 210 46 26 Fax: +375 (0)17 / 210 46 26	BELARUS	INTEHSIS srl bld. Traian 23/1 MD-2060 Kishinev Phone: +373 (0)22 / 66 4242 Fax: +373 (0)22 / 66 4280	MOLDOVA	MIDDLE EAST REPRESENTATIVES	
MITSUBISHI ELECTRIC EUROPE B.V. French Branch 25, Boulevard des Bouvets F-92741 Nanterre Cedex Phone: +33 (0)1 / 55 68 55 68 Fax: +33 (0)1 / 55 68 57 57	FRANCE	ESCO DRIVES & AUTOMATION Culliganlaan 3 BE-1831 Diegem Phone: +32 (0)2 / 717 64 30 Fax: +32 (0)2 / 717 64 31	BELGIUM	HIFLEX AUTOM. TECHNIEK B.V. Wolweverstraat 22 NL-2984 CD Ridderkerk Phone: +31 (0)180 - 46 60 04 Fax: +31 (0)180 - 44 23 55	NETHERLANDS	EGYPT	
MITSUBISHI ELECTRIC EUROPE B.V. Irish Branch Westgate Business Park, Ballymount IRL-Dublin 24 Phone: +353 (0)1 4198800 Fax: +353 (0)1 4198890	IRELAND	Koning & Hartman b.v. Woluwelaan 31 BE-1800 Vilvoorde Phone: +32 (0)2 / 257 02 40 Fax: +32 (0)2 / 257 02 49	BELGIUM	Koning & Hartman b.v. Haarlerbergweg 21-23 NL-1101 CH Amsterdam Phone: +31 (0)20 / 587 76 00 Fax: +31 (0)20 / 587 76 05	NETHERLANDS	ISRAEL	
MITSUBISHI ELECTRIC EUROPE B.V. Italian Branch Viale Colleoni 7 I-20864 Agrate Brianza (MB) Phone: +39 039 / 60 53 1 Fax: +39 039 / 60 53 312	ITALY	INEA RBT d.o.o. Aleja Lipa 56 BA-71000 Sarajevo Phone: +387 (0)33 / 921 164 Fax: +387 (0)33 / 524 539	BOSNIA AND HERZEGOVINA	Beijer Electronics AS Postboks 487 NO-3002 Drammen Phone: +47 (0)32 / 24 30 00 Fax: +47 (0)32 / 84 85 77	NORWAY	ISRAEL	
MITSUBISHI ELECTRIC EUROPE B.V. Poland Branch Krakowska 50 PL-32-083 Balice Phone: +48 (0)12 / 630 47 00 Fax: +48 (0)12 / 630 47 01	POLAND	AKHNATON 4, Andrej Ljapchev Blvd., PO Box 21 BG-1756 Sofia Phone: +359 (0)2 / 817 6000 Fax: +359 (0)2 / 97 44 06 1	BULGARIA	Fonseca S.A. R. João Francisco do Casal 87/89 PT - 3801-997 Aveiro, Esgueira Phone: +351 (0)234 / 303 900 Fax: +351 (0)234 / 303 910	PORTUGAL	LEBANON	
MITSUBISHI ELECTRIC EUROPE B.V. Russian Branch 52, bld. 3 Kosmodamienskaya nab 8 floor RU-115054 Moscow Phone: +7 495 721-2070 Fax: +7 495 721-2071	RUSSIA	INEA RBT d.o.o. Losinjska 4 a HR-10000 Zagreb Phone: +385 (0)1 / 36 940 -01 / -02 / -03 Fax: +385 (0)1 / 36 940 - 03	CROATIA	Sirius Trading & Services srl Aleea Lacul Morii Nr. 3 RO-060841 Bucuresti, Sector 6 Phone: +40 (0)21 / 430 40 06 Fax: +40 (0)21 / 430 40 02	ROMANIA	LEBANON	
MITSUBISHI ELECTRIC EUROPE B.V. Spanish Branch Carretera de Rubí 76-80 E-08190 Sant Cugat del Vallés (Barcelona) Phone: 902 131121 // +34 935653131 Fax: +34 935891579	SPAIN	AutoCont C.S. s.r.o. Technologická 374/6 CZ-708 00 Ostrava-Pustkovec Phone: +420 595 691 150 Fax: +420 595 691 199	CZECH REPUBLIC	INEA RBT d.o.o. Izletnička 10 SER-113000 Smederevo Phone: +381 (0)26 / 615 401 Fax: +381 (0)26 / 615 401	SERBIA	LEBANON	
MITSUBISHI ELECTRIC EUROPE B.V. UK Branch Travellers Lane UK-Hatfield, Herts. AL10 8XB Phone: +44 (0)1707 / 27 61 00 Fax: +44 (0)1707 / 27 86 95	UK	Beijer Electronics A/S Lykkegårdsvej 17 DK-4000 Roskilde Phone: +45 (0)46 / 75 76 66 Fax: +45 (0)46 / 75 56 26	DENMARK	SIMAP s.r.o. Jána Derku 1671 SK-911 01 Trenčín Phone: +421 (0)32 743 04 72 Fax: +421 (0)32 743 75 20	SLOVAKIA	LEBANON	
MITSUBISHI ELECTRIC EUROPE B.V. USA Branch 500 Corporate Woods Parkway Vernon Hills, IL 60061 Phone: +1 847 478 21 00 Fax: +1 847 478 22 53	USA	Beijer Electronics Eesti OÜ Pärnu mnt.160i EE-11317 Tallinn Phone: +372 (0)6 / 51 81 40 Fax: +372 (0)6 / 51 81 49	ESTONIA	PROCONT, spol. s r.o. Prešov Kúpeľná 1/A SK-080 01 Prešov Phone: +421 (0)51 7580 611 Fax: +421 (0)51 7580 650	SLOVAKIA	LEBANON	
		Beijer Electronics OY Peltoie 37 FIN-28400 Ulvila Phone: +358 (0)207 / 463 540 Fax: +358 (0)207 / 463 541	FINLAND	INEA RBT d.o.o. Stegne 11 SI-1000 Ljubljana Phone: +386 (0)1 / 513 8116 Fax: +386 (0)1 / 513 8170	SLOVENIA	LEBANON	
		UTEKO 5, Mavrogenou Str. GR-18542 Piraeus Phone: +30 211 / 1206 900 Fax: +30 211 / 1206 999	GREECE	Beijer Electronics AB Box 426 SE-20124 Malmö Phone: +46 (0)40 / 35 86 00 Fax: +46 (0)40 / 93 23 01	SWEDEN	LEBANON	
		MELTRADE Kft. Fertő utca 14. HU-1107 Budapest Phone: +36 (0)1 / 431-9726 Fax: +36 (0)1 / 431-9727	HUNGARY	Omni Ray AG Im Schörl 5 CH-8600 Dübendorf Phone: +41 (0)44 / 802 28 80 Fax: +41 (0)44 / 802 28 28	SWITZERLAND	LEBANON	
		Beijer Electronics SIA Ritausmas iela 23 LV-1058 Riga Phone: +371 (0)784 / 2280 Fax: +371 (0)784 / 2281	LATVIA	GTS Bayraktar Bulvari Nutuk Sok. No:5 TR-34775 Yukarı Dudullu-Ümraniye-İSTANBUL Phone: +90 (0)216 526 39 90 Fax: +90 (0)216 526 3995	TURKEY	LEBANON	
		Beijer Electronics UAB Savanoriu Pr. 187 LT-02300 Vilnius Phone: +370 (0)5 / 232 3101 Fax: +370 (0)5 / 232 2980	LITHUANIA	CSC Automation Ltd. 4-B, M. Raskovoyi St. UA-02660 Kiev Phone: +380 (0)44 / 494 33 55 Fax: +380 (0)44 / 494-33-66	UKRAINE	LEBANON	
						AFRICAN REPRESENTATIVE	
						SOUTH AFRICA	
						CBI Ltd. Private Bag 2016 ZA-1600 Isando Phone: +27 (0)11 / 977 0770 Fax: +27 (0)11 / 977 0761	