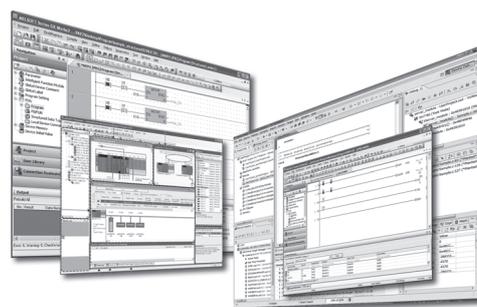


MELSOFT

Engineering Software

MX Component Version 5 Reference Manual

-SW5DND-ACT-E



SAFETY PRECAUTIONS

(Read these precautions before using this product.)

Before using this product, please read this manual carefully and pay full attention to safety to handle the product correctly. If the equipment is used in a manner not specified by the manufacturer, the protection provided by the equipment may be impaired.

The precautions given in this manual are concerned with this product. For the safety precautions of the programmable controller system, refer to the user's manual for the module used and MELSEC iQ-R Module Configuration Manual.

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



WARNING

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



CAUTION

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

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

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

- When data change, program change, or status control is performed from a personal computer to a running programmable controller, create an interlock circuit outside the programmable controller to ensure that the whole system always operates safely.

Furthermore, for the online operations performed from a personal computer to a CPU module, the corrective actions against a communication error due to such as a cable connection fault should be predetermined as a system.

[Design Precautions]



CAUTION

- The online operations performed from a personal computer to a running CPU module (forced output and operating status changes) must be executed after the manual has been carefully read and the safety has been ensured.

The operation failure may cause the injury or machine damage.

[Security Precautions]



WARNING

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

CONDITIONS OF USE FOR THE PRODUCT

- (1) MELSEC programmable controller ("the PRODUCT") shall be used in conditions;
- i) where any problem, fault or failure occurring in the PRODUCT, if any, shall not lead to any major or serious accident; and
 - ii) where the backup and fail-safe function are systematically or automatically provided outside of the PRODUCT for the case of any problem, fault or failure occurring in the PRODUCT.
- (2) The PRODUCT has been designed and manufactured for the purpose of being used in general industries. MITSUBISHI ELECTRIC SHALL HAVE NO RESPONSIBILITY OR LIABILITY (INCLUDING, BUT NOT LIMITED TO ANY AND ALL RESPONSIBILITY OR LIABILITY BASED ON CONTRACT, WARRANTY, TORT, PRODUCT LIABILITY) FOR ANY INJURY OR DEATH TO PERSONS OR LOSS OR DAMAGE TO PROPERTY CAUSED BY the PRODUCT THAT ARE OPERATED OR USED IN APPLICATION NOT INTENDED OR EXCLUDED BY INSTRUCTIONS, PRECAUTIONS, OR WARNING CONTAINED IN MITSUBISHI ELECTRIC USER'S, INSTRUCTION AND/OR SAFETY MANUALS, TECHNICAL BULLETINS AND GUIDELINES FOR the PRODUCT.
- ("Prohibited Application")
- Prohibited Applications include, but not limited to, the use of the PRODUCT in;
- Nuclear Power Plants and any other power plants operated by Power companies, and/or any other cases in which the public could be affected if any problem or fault occurs in the PRODUCT.
 - Railway companies or Public service purposes, and/or any other cases in which establishment of a special quality assurance system is required by the Purchaser or End User.
 - Aircraft or Aerospace, Medical applications, Train equipment, transport equipment such as Elevator and Escalator, Incineration and Fuel devices, Vehicles, Manned transportation, Equipment for Recreation and Amusement, and Safety devices, handling of Nuclear or Hazardous Materials or Chemicals, Mining and Drilling, and/or other applications where there is a significant risk of injury to the public or property.
- Notwithstanding the above restrictions, Mitsubishi Electric may in its sole discretion, authorize use of the PRODUCT in one or more of the Prohibited Applications, provided that the usage of the PRODUCT is limited only for the specific applications agreed to by Mitsubishi Electric and provided further that no special quality assurance or fail-safe, redundant or other safety features which exceed the general specifications of the PRODUCTS are required. For details, please contact the Mitsubishi Electric representative in your region.
- (3) Mitsubishi Electric shall have no responsibility or liability for any problems involving programmable controller trouble and system trouble caused by DoS attacks, unauthorized access, computer viruses, and other cyberattacks.

INTRODUCTION

Thank you for purchasing the engineering software, MELSOFT series.

This manual is designed for users to understand operations of MX Component.

Before using the product, thoroughly read this manual and related manuals to develop full familiarity with the functions and performance of MX Component and supported modules to ensure correct use.

Note that the menu names and operating procedures may differ depending on an operating system in use and its version.

When reading this manual, replace the names and procedures with the applicable ones as necessary.

CONTENTS

SAFETY PRECAUTIONS	1
CONDITIONS OF USE FOR THE PRODUCT	2
INTRODUCTION	3
RELEVANT MANUALS	12
TERMS	12
GENERIC TERMS AND ABBREVIATIONS	13

PART 1 FUNDAMENTALS AND CREATING PROCEDURE

CHAPTER 1 FUNDAMENTALS OF MX Component 16

1.1 Before Using this Product	16
Using MX Component	16
1.2 Types of Control	21
Control list	21
1.3 Types of Utility	22

CHAPTER 2 SYSTEM CONFIGURATIONS 24

2.1 Operating Environment	24
2.2 Supported Communication Routes	26
2.3 System Configuration List	27
System configuration examples of each communication route	27
2.4 Available Module Lists	29
CPU module list	29
Module list	31
2.5 Details on the Communication Routes	32

CHAPTER 3 USER APPLICATION CREATING PROCEDURES 39

3.1 Installing MX Component	39
Checking the Version	39
Opening the manual	39
3.2 Grasping a Control to be Used	40
3.3 Selecting a Communication Setting Method	40
3.4 Adding a Control to a User Program	41
3.5 Creating a New User Program	41
3.6 Communication Setting	41
3.7 Programming	41

CHAPTER 4 ADDING CONTROLS TO USER PROGRAM 42

4.1 When using VBA	42
4.2 When using Visual Basic .NET, Visual C#	46
4.3 When using Visual C++	49

CHAPTER 5 COMMUNICATION SETTING 54

5.1 Setting with Utilities	54
Starting Communication Settings Utility	54
[Target setting] tab	55
[List view] tab	60

	[Communication test] tab	61
	Exporting/importing the communication settings	62
	Specifying a created logical number in a program	64
5.2	Setting in Programs	65
	Checking a property with Communication Settings Utility	65
	Setting a required property in a user program	67
CHAPTER 6 COMMUNICATION SETTING EXAMPLES (UTILITY SETTING TYPE)		68
6.1	Serial Communication	68
	Access procedure	68
	Logical station number setting example	69
	Settings of serial communication modules	70
6.2	Ethernet Communication (When Using Ethernet Modules)	75
	Access procedure	75
	Logical station number setting example	76
	Settings of modules	77
6.3	Ethernet Communication (When Using Built-in Ethernet CPUs)	79
	Access procedure	79
	Logical station number setting example	80
	Settings of modules	81
6.4	Ethernet Communication (When Using Ethernet Adapter Modules)	82
	Access procedure	82
	Logical station number setting example	82
6.5	Ethernet Communication (When Using Ethernet Adapter)	83
	Access procedure	83
	Logical station number setting example	84
6.6	CPU COM Communication	85
	Access procedure	85
	Logical station number setting example	85
6.7	CPU USB Communication	86
	Access procedure	86
	Logical station number setting example	86
6.8	CC-Link Communication	87
	Access procedure	87
	Logical station number setting example	88
	Settings of modules	89
6.9	CC-Link system RS-232 interface Communication	90
	Access procedure	90
	Logical station number setting example	91
6.10	GX Simulator2 Communication	92
	Access procedure	92
	Logical station number setting example	92
6.11	GX Simulator3 Communication	93
	Access procedure	93
	Logical station number setting example	94
6.12	MT Simulator2 Communication	95
	Access procedure	95
	Logical station number setting example	95
6.13	MELSECNET/H Communication	96
	Access procedure	96

	Logical station number setting example	96
6.14	CC-Link IE Controller Network Communication	98
	Access procedure	98
	Logical station number setting example	98
6.15	CC-Link IE Field Network Communication	101
	Access procedure	101
	Logical station number setting example	102
6.16	CC-Link IE TSN Communication	104
	Access procedure	104
	Logical station number setting example	105
6.17	GOT Gateway Device Communication	106
	Access procedure	106
	Logical station number setting example	107
6.18	GOT Transparent Communication	108
	Access procedure	108
	Logical station number setting example	109
6.19	Inverter COM Communication	110
	Access procedure	110
	Logical station number setting example	110
6.20	Inverter USB Communication	111
	Access procedure	111
	Logical station number setting example	111
6.21	Robot controller COM Communication	112
	Access procedure	112
	Logical station number setting example	112
6.22	Robot controller USB Communication	113
	Access procedure	113
	Logical station number setting example	113
6.23	Robot controller Ethernet Communication	114
	Access procedure	114
	Logical station number setting example	114
CHAPTER 7 PROGRAMMING		115
7.1	Creating a Label With Label Utility	115
	Starting Label Utility	115
	Label screen	115
	Registering/deleting a logical station number	117
	System label list	118
	Structure setting	123
	Referring and registering/canceling Workspaces	124
	Change notification	125
	Updating system label data	126
	Exporting/Importing LabelSpace	127
	System label	128
7.2	Coding	142
7.3	Debugging	143
	Checking a value using PLC Monitor Utility	143

PART 2 DETAILED SPECIFICATIONS OF PROGRAMS

CHAPTER 8	ACCESSIBLE DEVICES AND DEVICE TYPES	154
8.1	Accessible Devices	154
	When the access target is a CPU module (other than motion CPU and MELSECWinCPU)	155
	When the access target is a motion CPU	157
	When the access target is a MELSECWinCPU	158
	When the access target is an RnSFCPU or RnPSFCPU (safety device)	158
	When the access target is an own board for CC-Link communication	159
	When the access target is an own board for MELSECNET/H communication	159
	When the access target is an own board for CC-Link IE Controller Network communication	159
	When the access target is an own board for CC-Link IE Field Network communication	159
	When the access target is an own board for CC-Link IE TSN communication	160
	When the access target is an inverter	161
	When the access target is a robot controller	162
8.2	Device Types	163
CHAPTER 9	PROPERTIES OF CONTROLS	170
9.1	Property List	170
9.2	Details of Control Properties	172
CHAPTER 10	PROPERTY SETTINGS OF COMMUNICATION ROUTES	181
10.1	Serial Communication	181
	Serial communication when the connected station is MELSEC iQ-R series-compatible C24 (1)	181
	Serial communication when the connected station is MELSEC iQ-R series-compatible C24 (2)	184
	Serial communication when the connected station is Q series-compatible C24 (1)	187
	Serial communication when the connected station is Q series-compatible C24 (2)	190
	Serial communication when the connected station is L series-compatible C24 (1)	192
	Serial communication when the connected station is L series-compatible C24 (2)	195
	Serial communication when the connected station is FX extended port	197
10.2	Ethernet Communication	199
	Ethernet communication when the connected station is MELSEC iQ-R series-compatible E71 (TCP)	199
	Ethernet communication when the connected station is MELSEC iQ-R series-compatible E71 (UDP)	202
	Ethernet communication when the connected station is Q series-compatible E71 (TCP)	205
	Ethernet communication when the connected station is Q series-compatible E71 (UDP)	208
	Ethernet communication when the connected station is L series-compatible E71 (TCP)	211
	Ethernet communication when the connected station is L series-compatible E71 (UDP)	214
	Ethernet communication when the connected station is a MELSEC iQ-R series CPU module (TCP)	217
	Ethernet communication when the connected station is a MELSEC iQ-R series CPU module (UDP)	221
	Ethernet communication when the connected station is an LHCPU (TCP)	225
	Ethernet communication when the connected station is an LHCPU (UDP)	227
	Ethernet communication when the connected station is an FX5CPU (TCP)	229
	Ethernet communication when the connected station is an FX5CPU (UDP)	231
	Ethernet communication when the connected station is a built-in Ethernet port QCPU (TCP)	233
	Ethernet communication when the connected station is a built-in Ethernet port QCPU (UDP)	237
	Ethernet communication when the connected station is a built-in Ethernet port LCPU (TCP)	241
	Ethernet communication when the connected station is a built-in Ethernet port LCPU (UDP)	244
	Ethernet communication when the connected station is a Ethernet adapter module (TCP)	247
	Ethernet communication when the connected station is a Ethernet adapter module (UDP)	251
	Ethernet communication when the connected station is an Ethernet adapter (TCP) (1)	255
	Ethernet communication when the connected station is an Ethernet adapter (TCP) (2)	256
	Ethernet communication when the connected station is an Ethernet adapter (UDP) (1)	258

	Ethernet communication when the connected station is an Ethernet adapter (UDP) (2)	259
	Ethernet communication when the connected station is an Ethernet adapter (UDP) (3)	261
	Ethernet communication when the connected station is a robot controller	263
	Ethernet communication when the connected station is a CC-Link IE TSN module (TCP)	264
	Ethernet communication when the connected station is a CC-Link IE TSN module (UDP)	267
10.3	CPU COM Communication	270
	CPU COM communication when the connected station is an FX5CPU	270
	CPU COM communication when the connected station is a QCPU (Q mode)	272
	CPU COM communication when the connected station is an LCPU	275
	CPU COM communication when the connected station is a Q motion CPU	278
	CPU COM communication when the connected station is an FXCPU	280
	CPU COM communication when the connected station is an inverter	282
	CPU COM communication when the connected station is a robot controller	283
10.4	CPU USB Communication	284
	CPU USB communication when the connected station is an RCPU	284
	CPU USB communication when the connected station is an R motion CPU	287
	CPU USB communication when the connected station is an LHCPU	289
	CPU USB communication when the connected station is an FX5CPU	291
	CPU USB communication when the connected station is a QCPU (Q mode)	293
	CPU USB communication when the connected station is an LCPU	297
	CPU USB communication when the connected station is a QSCPU	300
	CPU USB communication when the connected station is a Q motion CPU	301
	CPU USB communication when the connected station is an FXCPU	303
	CPU USB communication when the connected station is an inverter	304
	CPU USB communication when the connected station is a robot controller	305
10.5	MELSECNET/H Communication	306
10.6	CC-Link IE Controller Network Communication	309
	CC-Link IE Controller Network communication when the connected station is an RCPU	309
	CC-Link IE Controller Network communication when the connected station is a module supporting Q series	311
10.7	CC-Link IE Field Network Communication	314
	CC-Link IE Field Network communication when the connected station is an RCPU	314
	CC-Link IE Field Network communication when the connected station is a module supporting QCPU (Q mode) or LCPU	316
10.8	CC-Link IE TSN Communication	319
	CC-Link IE TSN communication when the connected station is an RCPU	319
10.9	CC-Link Communication	321
	CC-Link communication when the connected station is an RCPU	321
	CC-Link communication when the connected station is an LCPU	323
	CC-Link communication when the connected station is a module supporting Q series	325
10.10	CC-Link System RS-232 Interface Communication	328
10.11	GX Simulator2 Communication	331
10.12	GX Simulator3 Communication	331
10.13	MT Simulator2 Communication	332
10.14	GOT Gateway Device Communication	332
10.15	GOT Transparent Communication	333
	Personal computer side port: USB, GOT2000 side port: USB, CPU side port: Direct connection (1)	333
	Personal computer side port: USB, GOT2000 side port: USB, CPU side port: Direct connection (2)	335
	Personal computer side port: USB, GOT2000 side port: USB, CPU side port: Direct connection (3)	337
	Personal computer side port: USB, GOT2000 side port: USB, CPU side port: Direct connection (4)	340
	Personal computer side port: USB, GOT2000 side port: USB, CPU side port: Bus connection	342

Personal computer side port: USB, GOT2000 side port: USB, CPU side port: Serial communication module (1) . 345

Personal computer side port: USB, GOT2000 side port: USB, CPU side port: Serial communication module (2) . 348

Personal computer side port: USB, GOT2000 side port: USB, CPU side port: Serial communication module (3) . 350

Personal computer side port: USB, GOT2000 side port: USB, CPU side port: Serial communication module (4) . 352

Personal computer side port: USB, GOT2000 side port: USB, CPU side port: Serial communication module (5) . 354

Personal computer side port: USB, GOT2000 side port: USB, CPU side port: Serial communication module (6) . 357

Personal computer side port: Serial, GOT2000 side port: Serial, CPU side port: MELSEC iQ-R series-compatible E71 359

Personal computer side port: USB, GOT2000 side port: USB, CPU side port: MELSEC iQ-R series-compatible E71 362

Personal computer side port: USB, GOT2000 side port: USB, CPU side port: Q series-compatible E71 365

Personal computer side port: USB, GOT2000 side port: USB, CPU side port: L series-compatible E71 368

Personal computer side port: USB, GOT2000 side port: USB, CPU side port: Ethernet port (1) 371

Personal computer side port: USB, GOT2000 side port: USB, CPU side port: Ethernet port (2) 374

Personal computer side port: USB, GOT2000 side port: USB, CPU side port: Ethernet port (3) 376

Personal computer side port: USB, GOT2000 side port: USB, CPU side port: Ethernet port (4) 380

Personal computer side port: USB, GOT2000 side port: USB, CPU side port: Ethernet adapter module 383

Personal computer side port: USB, GOT2000 side port: USB, CPU side port: Ethernet adapter/module 387

Personal computer side port: USB, GOT2000 side port: USB, CPU side port: Ethernet adapter 389

Personal computer side port: Ethernet board, GOT2000 port: Ethernet port, CPU side port: Serial (1) 391

Personal computer side port: Ethernet board, GOT2000 port: Ethernet port, CPU side port: Serial (2) 393

Personal computer side port: Ethernet board, GOT2000 port: Ethernet port, CPU side port: Serial (3) 396

Personal computer side port: Ethernet board, GOT2000 port: Ethernet port, CPU side port: Serial (4) 399

Personal computer side port: Ethernet board, GOT2000 side port: Ethernet port, CPU side port: Serial communication module (1) 400

Personal computer side port: Ethernet board, GOT2000 side port: Ethernet port, CPU side port: Serial communication module (2) 403

Personal computer side port: Ethernet board, GOT2000 side port: Ethernet port, CPU side port: Serial communication module (3) 406

Personal computer side port: Ethernet board, GOT2000 side port: Ethernet port, CPU side port: Bus connection 409

Personal computer side port: Ethernet board, GOT2000 port: Ethernet port, CPU side port: Ethernet port . . . 412

Personal computer side port: Ethernet board, GOT2000 port: Ethernet port, CPU side port: Q series-compatible E71 414

CHAPTER 11 FUNCTIONS 416

11.1 Programming Considerations 416

11.2 Function List 420

11.3 Details of Functions (For Act Control) 422

Open (Opening communication line) 422

Close (Closing communication line) 424

ReadDeviceBlock (Reading devices in batch) 425

WriteDeviceBlock (Writing devices in batch) 428

ReadDeviceRandom (Reading devices randomly) 432

WriteDeviceRandom (Writing devices randomly)	436
SetDevice (Setting device data)	439
GetDevice (Acquiring device data)	441
ReadBuffer (Reading data from buffer memory)	444
WriteBuffer (Writing data to buffer memory)	448
GetClockData (Reading clock data)	452
SetClockData (Writing clock data)	456
GetCpuType (Reading CPU module model)	460
SetCpuStatus (Remote control)	462
EntryDeviceStatus (Registering devices for status monitoring)	465
FreeDeviceStatus (Deregistering devices for status monitoring)	470
OnDeviceStatus (Event notification)	471
ReadDeviceBlock2 (Reading devices in batch)	474
WriteDeviceBlock2 (Writing devices in batch)	477
ReadDeviceRandom2 (Reading devices randomly)	480
WriteDeviceRandom2 (Writing devices randomly)	483
SetDevice2 (Setting device data)	487
GetDevice2 (Acquiring device data)	490
GetErrorMessage(Acquiring error description and corrective action)	492
11.4 Details of Functions (For Act Control (Logging File Transfer)).	494
Open (Opening communication line)	494
Close (Closing communication line)	496
ReadFirstFile (Searching for a file/directory)	497
ReadNextFile (Searching for a file/directory)	499
ReadClose (Ending the search)	501
GetFile (Transferring logging files)	502
11.5 Details of Functions (For .Net Control)	504
Open (Opening communication line)	504
Close (Closing communication line)	505
ReadDeviceBlock (Reading devices in batch)	506
WriteDeviceBlock (Writing devices in batch)	512
ReadDeviceRandom (Reading devices randomly)	518
WriteDeviceRandom (Writing devices randomly)	529
SetDevice (Setting device data)	540
GetDevice (Acquiring device data)	544
ReadBuffer (Reading data from buffer memory)	548
WriteBuffer (Writing data to buffer memory)	550
GetClockData (Reading clock data)	552
SetClockData (Writing clock data)	554
GetCpuType (Reading CPU module model)	556
SetCpuStatus (Remote control)	557
EntryDeviceStatus (Registering devices for status monitoring)	558
FreeDeviceStatus (Deregistering devices for status monitoring)	562
OnDeviceStatus (Event notification)	563
ReadDeviceBlock2 (Reading devices in batch)	565
WriteDeviceBlock2 (Writing devices in batch)	570
ReadDeviceRandom2 (Reading devices randomly)	575
WriteDeviceRandom2 (Writing devices randomly)	584
SetDevice2 (Setting device data)	595
GetDevice2 (Acquiring device data)	599
Dispose (release memory)	602

CHAPTER 12 SAMPLE PROGRAMS	603
12.1 Visual Basic.NET, Visual C#, Visual C++	605
Act_2019.sln, Dot_2019.sln	605
ActDatalogging_2019.sln	608
12.2 VBA	609
For Access (ActTest_EarlyBind.accdb, ActTest_LateBind.accdb, ActTest64.accdb)	609
For Excel (ActUtilType.xlsm, ActUtilType64.xlsm, ActProgType.xlsm, and ActProgType64.xlsm)	613
CHAPTER 13 ERROR CODES	616
13.1 Error Code List	616
13.2 Error Codes Returned by CPUs, Modules, and Network Boards	644
13.3 HRESULT Type Error Codes	645
13.4 Error Codes Displayed on Event Viewer	645
APPENDIX	646
Appendix 1 Changes from Previous Version	646
Appendix 2 Concept of Routing Parameters	647
Appendix 3 Multi-CPU System	650
Appendix 4 The Number of Mountable Network Modules for Q00UJCPU, Q00UCPU, and Q01UCPU	651
Appendix 5 Compatibility with Redundant CPU	652
For RnPCPU (redundant mode) and RnPSFCPU	652
For QnPRH	655
Appendix 6 Character Strings that cannot be Used for Label Names	663
Appendix 7 Performance	665
Performance of MX Component (Control)	665
Performance of the Label Service	666
Performance of the Label Utility	667
Appendix 8 Installing a USB Driver	669
Appendix 9 Programing Examples for Monitoring Word Device Status	670
Appendix 10 Time-Out Periods	673
Communication retries at time-out error occurrence	673
Communication retries at receive data error occurrence	673
Time-out errors at fixed time in the Act control	673
Appendix 11 Troubleshooting	674
When a header file is not created properly at the time of pasting a control to a form with Visual C++	677
When an error occurs in the setting for using controls in Visual Studio .NET	678
When a link error occurs in creating a user program in Visual C++	679
When an unstart error occurs during communication with GX Simulator3	680
Appendix 12 Differences with previous version of MX Component	681
Comparison with MX Component Version 4	681
Compatibility	684
FUNCTION INDEX	686
REVISIONS	688
WARRANTY	689
INFORMATION AND SERVICES	690
TRADEMARKS	690
COPYRIGHTS	690

RELEVANT MANUALS

Manual name [manual number]	Description	Available form
MX Component Version 5 Reference Manual [SH-082395ENG] (this manual)	Operation method, programming procedures, and error codes of MX Component Version 5	e-Manual PDF



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

e-Manual has the following features:

- Required information can be cross-searched in multiple manuals.
- Other manuals can be accessed from the links in the manual.
- Hardware specifications of each part can be found from the product figures.
- Pages that users often browse can be bookmarked.
- Sample programs can be copied to an engineering tool.

TERMS

Unless otherwise specified, this manual uses the following terms.

Term	Description
Control CPU	A CPU module that controls connected I/O modules and intelligent function modules. In a multiple CPU system, a control CPU can be set for each module.
Control station	Only one station that controls the network to which it is connected. Each station's send range for cyclic transmission is assigned to the control station.
Control system	A system that controls a redundant system and performs network communications in a redundant system
Engineering tool	A tool used for setting up programmable controllers, programming, debugging, and maintenance
Ethernet port direct connection	A direct connection between a programmable controller and personal computer without using a hub
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
Master station	A station that controls the entire network. This station can perform cyclic transmission and transient transmission with all stations. Only one master station can be used in a network.
MELSOFT Navigator	An integrated development environment included in MELSOFT iQ Works
Multiple CPU system	A system where two to four CPU modules separately control I/O modules and intelligent function modules
Other station	A station connected to the connected station (host station) on the network
Own system	A system that contains the CPU module that takes control in a redundant system
Process CPU	A CPU module that performs process control and sequence control. Process control function blocks and the online module change function can be executed. This module is also used with a redundant function module as a pair and configures a redundant system.
Redundant system	A system consisting of two systems that have same configuration (CPU module, power supply module, network module, and other modules). Even after an error occurs in one of the two system, the other system takes over the control of the entire system.
Relay station	A station that relays data link to other station with mounting more than one network modules on one programmable controller
Safety CPU	A module that performs both standard control and safety control and is used with a safety function module
Sequence program	Programming system devised to make a contact type sequence compatible with the programmable controller language as-is. Draw two vertical control buses and describe contacts, etc. between the buses to perform programming.
Standby system	A backup system in a redundant system
System A	A system that is set as system A to distinguish two systems in a redundant system. It is used to distinguish connected two systems (system A and system B).
System B	A system that is set as system B to distinguish two systems in a redundant system. It is used to distinguish connected two systems (system A and system B).
System switching	A function which switches the systems between the control system and the standby system to continue operation of the redundant system when a failure or an error occurs in the control system
The other system	The other system that has been connected to own system with tracking cables. When the system A is the own system, the system B is the other system, and vice versa.

GENERIC TERMS AND ABBREVIATIONS

Unless otherwise specified, this manual uses the following generic terms and abbreviations.

Generic term/abbreviation	Description
CC-Link IE TSN board	CC-Link IE TSN interface board
CC-Link IE Controller Network board	CC-Link IE Controller Network interface board
CC-Link IE Field Network board	CC-Link IE Field Network interface board
CC-Link Ver.2 board	CC-Link system master/local interface board
C Controller module	R12CCPU-V, Q12DCCPU-V (Basic mode), Q12DCCPU-V (Extended mode), Q24DHCCPU-V, Q24DHCCPU-LS, Q24DHCCPU-VG, Q26DHCCPU-LS
Ethernet adapter module	NZ2GF-ETB CC-Link IE Field Network Ethernet adapter module
Built-in Ethernet port QCPU	A Q series CPU module with an Ethernet port
Ethernet module	MELSEC iQ-R series-compatible E71, Q series-compatible E71, L series-compatible E71
Built-in Ethernet CPU	A CPU module with an Ethernet port
GOT	Mitsubishi Electric Graphic Operation Terminal GOT1000 series, GOT2000 series
GT SoftGOT	GT SoftGOT2000 Version1, GT SoftGOT1000 Version2, GT SoftGOT1000 Version3
MELSECNET/H board	MELSECNET/H interface board
Q12DCCPU-V (Extended mode)	Status that Q12DCCPU-V is initialized with the extended mode. For Q12DCCPU-V (Extended mode), refer to the following:  MELSEC-Q C Controller Module User's Manual
Q12DCCPU-V (Basic mode)	Status that Q12DCCPU-V is initialized with the basic mode. For Q12DCCPU-V (Basic mode), refer to the following:  C Controller Module User's Manual (Hardware Design, Function Explanation)
QnPRH	Q12PRHCPU, Q25PRHCPU
Serial communication module	MELSEC iQ-R series-compatible C24, Q series-compatible C24, L series-compatible C24, FX extended port
Redundant CPU	QnPRH, RnPCPU, RnPSFCPU
Redundant type extension base unit	Q65WRB extension base unit for redundant system
High performance model QCPU	Q02CPU, Q02HCPU, Q06HCPU, Q12HCPU, Q25HCPU
Basic model QCPU	Q00JCPU, Q00CPU, Q01CPU
Robot controller	CR750-D series, CRnD-700 series

MEMO

This part explains the fundamentals of MX Component and creating procedure.

1 FUNDAMENTALS OF MX Component

2 SYSTEM CONFIGURATIONS

3 USER APPLICATION CREATING PROCEDURES

4 ADDING CONTROLS TO USER PROGRAM

5 COMMUNICATION SETTING

6 COMMUNICATION SETTING EXAMPLES (UTILITY SETTING TYPE)

7 PROGRAMMING

1 FUNDAMENTALS OF MX Component

1.1 Before Using this Product

MX Component is a library that realizes easy communication from a personal computer to a programmable controller. By using the controls provided by MX Component, device information can be acquired from a programmable controller and an application can be developed.

In addition, a communication route to a programmable controller can be set in detail by using the utilities.

When applying any of the following program examples to the actual system, examine the applicability and confirm that no problem will occur in the system control.

Using MX Component

Support of a wide range of communication routes for programmable controller

A wide range of communication routes to a programmable controller are supported to enable the user to configure a system as desired.

Substantial improvement in user's development efficiency

When creating a user program, select either the utility setting type or program setting type. (☞ Page 40 Selecting a Communication Setting Method)

In case of utility settings type, the wizard type communication settings utility is provided.

By simply setting dialog-based communication settings on the screen, communication settings to access a CPU module to communicate with can be realized.

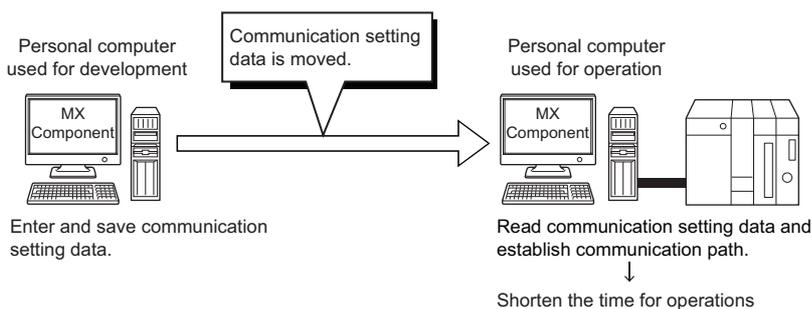
Once the communication settings are set, stations can be accessed by only specifying the logical station number of the CPU module stored in Communication Settings Utility.

Save and read of communication settings

MX Component features the functions to save and read the communication settings set in Communication Settings Utility.

The set data can easily be moved from a personal computer used for development to that used for operation.

MX Component must be installed in both of the computers.



Create programs with labels

Programs can be created without considering device numbers by using the label function.

Devices can be read/written by using the label names.

'Execute the processing of function "ReadDeviceRandom2".

```
iReturnCode=
DotUtilType.ReadDeviceRandom2(
"Amount of production",
3,
objData)
```

Structured data type label

D0	Word
CN200	Word
D1	Word

Specify the label name.

'Execute the processing of function "ReadDeviceRandom2".

```
iReturnCode=
DotUtilType.ReadDeviceRandom2(
"AlarmArray",
3,
objectValue)
```

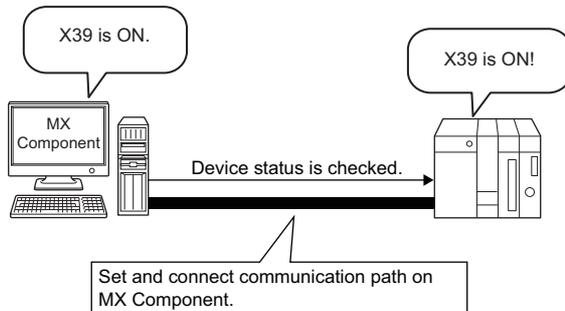
Array type label

[0] : D0	Word
[1] : D1	Word
[2] : D2	Word

Free from complex use of data-type-dedicated methods.

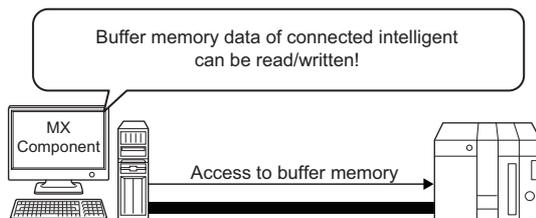
Monitor devices and change their data

By utilizing PLC Monitor Utility, the status of a specified device can be monitored and its data can be changed.



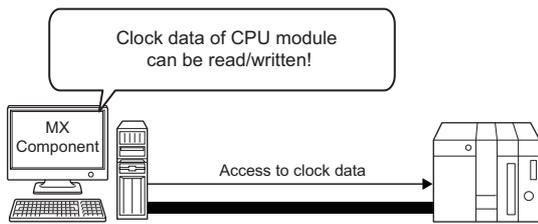
Access to buffer memory of intelligent function module

Not only devices of a CPU module but also the buffer memory of an intelligent function module can be accessed.



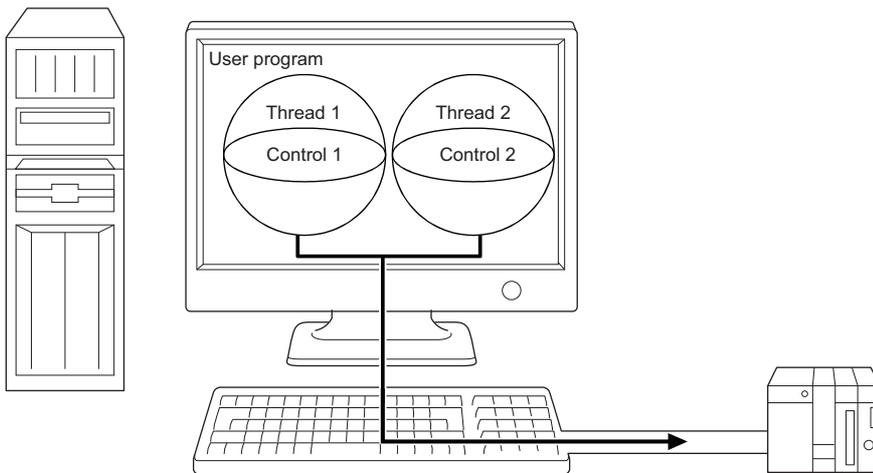
Read/write of CPU module clock data

The clock data of a CPU module connected to a personal computer can be read/written.



Multi-thread communication

The same communication route can be accessed from multiple threads at the same time.

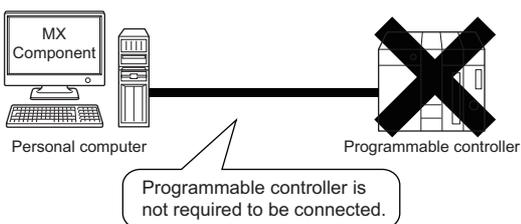


The simulator function for offline debugging

Communication with various simulators (GX Simulator2, GX Simulator3, and MT Simulator2) can be performed.

By using the simulation function, debugging can be performed on a single personal computer without connecting with a programmable controller.

This function is not supported by QSCPU.



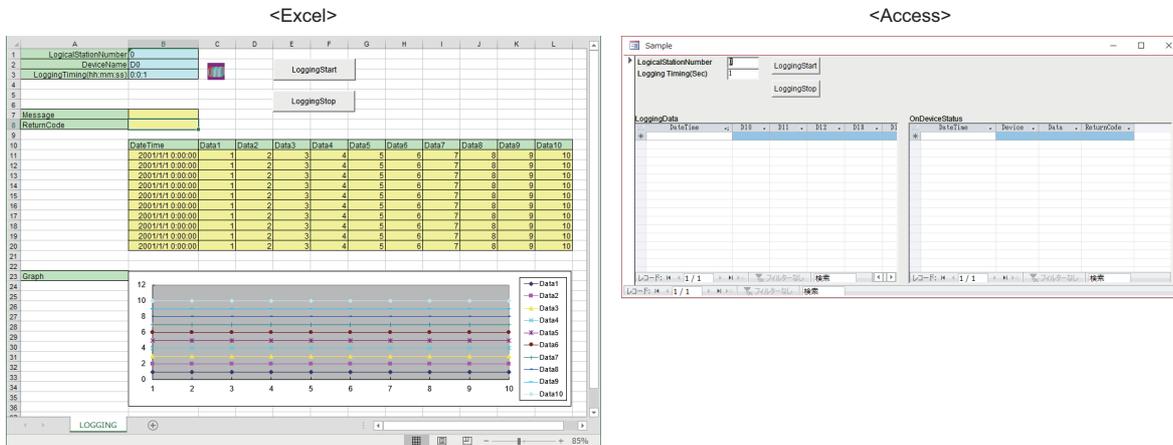
A wide variety of programming languages supported

MX Component supports Visual Basic .NET, Visual C++, Visual C#, and VBA.

■VBA-driven data collection and monitoring function

Programming using VBA allows Excel or Access functions to be utilized to create an application for providing a real-time graph display.

Device data of a programmable controller can be logged and device data can be sampled/saved in real time.



Support multi-CPU system of QCPU (Q mode)

Setting Communication Settings Utility or control properties enables access to the multi-CPU system of QCPU (Q mode).

Transfer logging files to a personal computer

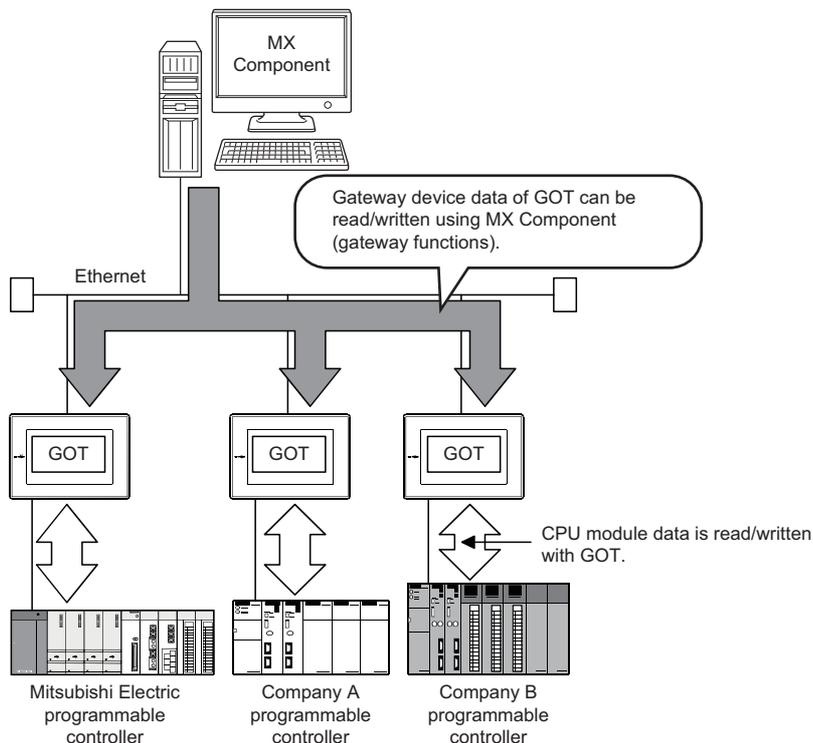
A logging file in an SD memory card inserted in an RCPUCPU can be transferred to a personal computer.

For controls for transferring logging files, refer to the following:

📄 Page 21 Act Control (logging file transfer)

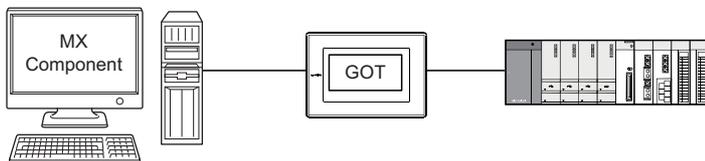
Accessibility to gateway devices of GOT

The device data of a CPU module that is being monitored by GOT can be read/written by reading from/writing to the gateway device data of GOT.



Support GOT transparent communication

Using the GOT transparent function, a CPU module can be accessed via GOT.

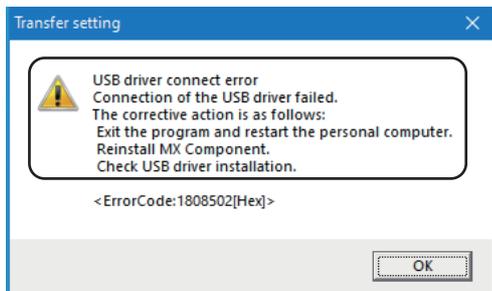


Reduction of error description search time

By using the 'ActSupportMsg' which is a control for the troubleshooting function, the error description and corrective action can be displayed within a user program by only specifying the error code.

Therefore, checking error description or corrective action on a manual is not required when an error occurs in a control.

- Example of displaying error description and corrective action in a message box



1.2 Types of Control

This section explains controls provided by MX Component.

Controls shown in this section are used to create user programs to communicate with a CPU module. By using these controls, communication with a CPU module can be realized without being aware of the hardware and the communication protocol.

There are the following three types of controls provided by MX Component.

- Act Control
- Act Control (logging file transfer)
- .Net Control

The following table shows each control.

Control list

Act control

The following table shows the ActiveX Controls provided by MX Component.

Data can be accessed using devices.

Control name		Application	Supported language
32-bit	ActUtilType	A utility setting type control.	Visual C# Visual Basic .NET Visual C++ VBA
64-bit	ActUtilType64	It is used to create a user program by using Communication Settings Utility.	
32-bit	ActProgType	A program setting type control.*1	
64-bit	ActProgType64	It is used to create a user program without using Communication Settings Utility.	
32-bit	ActSupportMsg	Common control for a utility setting type and program setting type.	
64-bit	ActSupportMsg64	Used for the troubleshooting function.	

*1 Not applicable to inverter communication/robot controller communication.

Act Control (logging file transfer)

The following table shows the ActiveX Controls for transferring logging files provided by MX Component.

Do not use the controls while the logging function of a CPU module is in operation.

Control name		Application	Supported language
32-bit	ActUtilDataLogging	A utility setting type control.	Visual C# Visual Basic .NET Visual C++
64-bit	ActUtilDataLogging64	It is used to transfer a logging file. It operates with a logical station number created by using Communication Settings Utility.	
32-bit	ActProgDataLogging	A program setting type control.	
64-bit	ActProgDataLogging64	It is used to transfer a logging file. It operates with the property of the control.	

.Net control

The following table shows the .Net Controls provided by MX Component.

Data can be accessed using labels.

Control name		Application	Supported language
32-bit	DotUtilType	A utility setting type control.	Visual C# Visual Basic .NET
64-bit	DotUtilType64	It is used to create a user program by using Communication Settings Utility.	

1.3 Types of Utility

The following are the utilities used in MX Component.

Item	Description	Reference
Communication Settings Utility	Sets the communication settings in the wizard format.	Page 54 Setting with Utilities
Label Utility	Registers/utilizes the system labels.	Page 115 Creating a Label With Label Utility
PLC Monitor Utility	Sets a connection destination by using a logical station number, and monitors the devices and buffer memories of a programmable controller.	Page 143 Checking a value using PLC Monitor Utility

Administrator authority

When Label Utility is executed, the status is switched to the administrator authority automatically.

When executing Communication Settings Utility and PLC Monitor Utility, the performance differs according to the user account control (UAC) settings.

■When user account control (UAC) is enabled

All users including administrator operate as "standard user" in the default setting.

To execute programs with administrator authority, specify "Run as administrator."

■When user account control (UAC) is disabled

Programs can be executed by login user.

■Warning messages in Windows

When executing a utility as an administrator, a warning message appears.



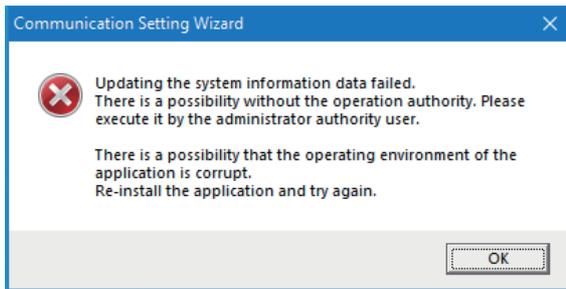
To execute a utility as an administrator at all times, select the checkbox of "Run this program as an administrator" in the [Compatibility] tab of [Property].

■When MX Component is operated by a user without Administrator's authority.

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

Item	Restrictions
Communication Settings Utility	The logical station number cannot be created, changed, or deleted. Communication settings cannot be imported. This utility cannot be started up if the communication settings are set using MX Component earlier than Version 3.00A.*1
PLC Monitor Utility	This utility cannot be started up if the communication settings are set using MX Component earlier than Version 3.00A.*1

- *1 If the following error message appears, start up or close the utility as a user with Administrator's authority.
This operation enables a user without Administrator's authority to start up the utility.



2 SYSTEM CONFIGURATIONS

This chapter explains the system configuration of MX Component such as the operating environment, communication routes, and available devices.

2.1 Operating Environment

Item	Description	
Personal computer	Personal computer where Microsoft® Windows® operates	
CPU	Windows 11	2 or more cores on a compatible 64-bit processor or System on a Chip (SoC)
	Windows 10	Processor 1.6 GHz with 2 or more cores
Required memory	Windows 11	4 GB or more recommended
	Windows 10	2 GB or more recommended
Available hard disk capacity	<ul style="list-style-type: none"> For installation: 2.5 GB or more free hard disk capacity For operation: 512 MB or more free virtual memory capacity 	
Monitor	Windows 11	Resolution WXGA (1280 × 720 pixels) or higher
	Windows 10	Resolution 1024 × 768 pixels or higher
OS (English version) (32-bit version/64-bit version)	<ul style="list-style-type: none"> Windows 11 (Home, Pro, Enterprise, Education)^{*1} Windows 10 (Home, Pro, Enterprise, Education, IoT Enterprise 2016 LTSC^{*1}, IoT Enterprise 2019 LTSC^{*1}) 	
Programming language ^{*2*3} (English version) (32-bit version/64-bit version)	Visual Basic.Net	<ul style="list-style-type: none"> Microsoft Visual Studio® 2019 Microsoft Visual Studio 2022
	Visual C#	<ul style="list-style-type: none"> Microsoft Visual Studio 2019 Microsoft Visual Studio 2022
	Visual C++	<ul style="list-style-type: none"> Microsoft Visual Studio 2019 Microsoft Visual Studio 2022
	VBA	<ul style="list-style-type: none"> Microsoft Excel® 2019 Microsoft Excel 2021^{*5} Microsoft Excel for Microsoft 365^{*1*6} Microsoft Access® 2019 Microsoft Access 2021^{*5} Microsoft Access for Microsoft 365^{*1*6}
Framework ^{*4}	.NET Framework 3.5/4.5.2/4.6/4.6.1/4.6.2/4.7/4.7.1/4.7.2/4.8	

*1 64-bit version only

*2 User programs created in the English environment are applicable to the English environment only, cannot be used in the other environment.

*3 Visual C++ and VBA do not support access with labels.

*4 .Net Core and .Net 5 are not supported.

*5 The operation of this product has been confirmed with the version 2206 (Build 15330.20196). This product does not run with a version earlier than this version.

*6 The operation of this product has been confirmed with the version 2201 (Build 14827.20192). This product does not run with a version earlier than this version.

.NET Framework 3.5 (including .NET 2.0 and 3.0), .NET Framework 4.6 Advanced Services, and .NET Framework 4.8 Advanced Services need to be enabled. For details, refer to TECHNICAL BULLETIN No. FA-A-0207.

When the following functions are used, this product may not run properly.

- Application start-up in Windows compatibility mode
- Fast user switching
- Remote desktop
- Touch function
- Virtual environment such as Client Hyper-V
- Tablet mode
- Virtual Desktops
- Windows hibernate or standby
- Unified Write Filter
- Text cursor indicator

In the following cases, the screen of this product may not work properly.

- The size of the text and other items in the screen is other than 100% (96 DPI, 9 pt etc.).
- The resolution of the screen is changed in operation.
- The Windows theme is changed in operation.
- The multi-display is set.

If the Windows firewall setting is enabled, the "Find Module function" and "Direct connection function" may not operate correctly. Disable the Windows firewall setting.

In this product, the period (.) is used as the symbol of the decimal point. Regardless of the setting of "Decimal symbol" on the control panel, use the period when entering the decimal point.

Precautions

- A communication error may occur when communicating with a CPU module after setting the resume function, suspend setting, power-saving function, and/or standby mode of a personal computer.
Therefore, when communicating with the CPU module, do not set the above functions.
- Surrogate pair characters and environment dependent characters are not available in MX Component utilities.
- MX Component cannot be used from a Windows service application.

2.2 Supported Communication Routes

This section shows the supported communication routes by MX Component.

Communication route	Description	Reference
Serial communication	To perform communication with a CPU module by using a serial communication module.	Page 32 Serial communication
Ethernet communication	To perform communication by connecting a personal computer to an Ethernet module or a built-in Ethernet CPU.	Page 33 Ethernet communication
CPU COM communication	To perform communication by connecting a personal computer to an RS-232 or RS-422 connector of a CPU module.	Page 33 CPU COM communication
CPU USB communication	To perform communication by connecting a personal computer to a USB connector of a CPU module.	Page 36 CPU USB communication
MELSECNET/H communication*1	To perform communication with a CPU module by using a MELSECNET/H board.	Page 36 MELSECNET/H communication
CC-Link IE Controller Network communication*1	To perform communication with a CPU module by using a CC-Link IE Controller Network board.	Page 36 CC-Link IE Controller Network communication
CC-Link IE Field Network communication*1	To perform communication with a CPU module by using a CC-Link IE Field Network board.	Page 36 CC-Link IE Field Network communication
CC-Link IE TSN communication*1	To perform communication with a CPU module by using a CC-Link IE TSN board.	Page 36 CC-Link IE TSN communication
CC-Link communication*1	To perform communication with a CPU module by using a CC-Link Ver.2 board.	Page 37 CC-Link communication
CC-Link system RS-232 interface communication	To perform communication with a CPU module by using an AJ65BT-R2N module.	Page 37 CC-Link system RS-232 interface communication
GX Simulator2 communication	To perform communication using the simulation function of GX Works2.	Page 37 GX Simulator2 communication
GX Simulator3 communication	To perform communication using the simulation function of GX Works3.	Page 37 GX Simulator3 communication
MT Simulator2 communication	To perform communication using the simulation function of MT Developer2.	Page 37 MT Simulator2 communication
GOT gateway device communication	To perform communication with a CPU module and a third-party programmable controller by using the gateway functions of GOT.	Page 37 GOT gateway device communication
GOT transparent communication	To perform communication with a CPU module by using the GOT transparent function.	Page 37 GOT transparent communication
Inverter COM communication	To perform communication by connecting an inverter to a COM port of a personal computer.	Page 37 Inverter (COM/USB) communication
Inverter USB communication	To perform communication by connecting an inverter to a USB port of a personal computer.	
Robot controller COM communication	To perform communication by connecting a robot controller to a COM port of a personal computer.	Page 38 Robot controller(COM/USB/Ethernet) communication
Robot controller USB communication	To perform communication by connecting a robot controller to a USB port of a personal computer.	
Robot controller Ethernet communication	To perform communication by connecting a robot controller and a personal computer to Ethernet.	

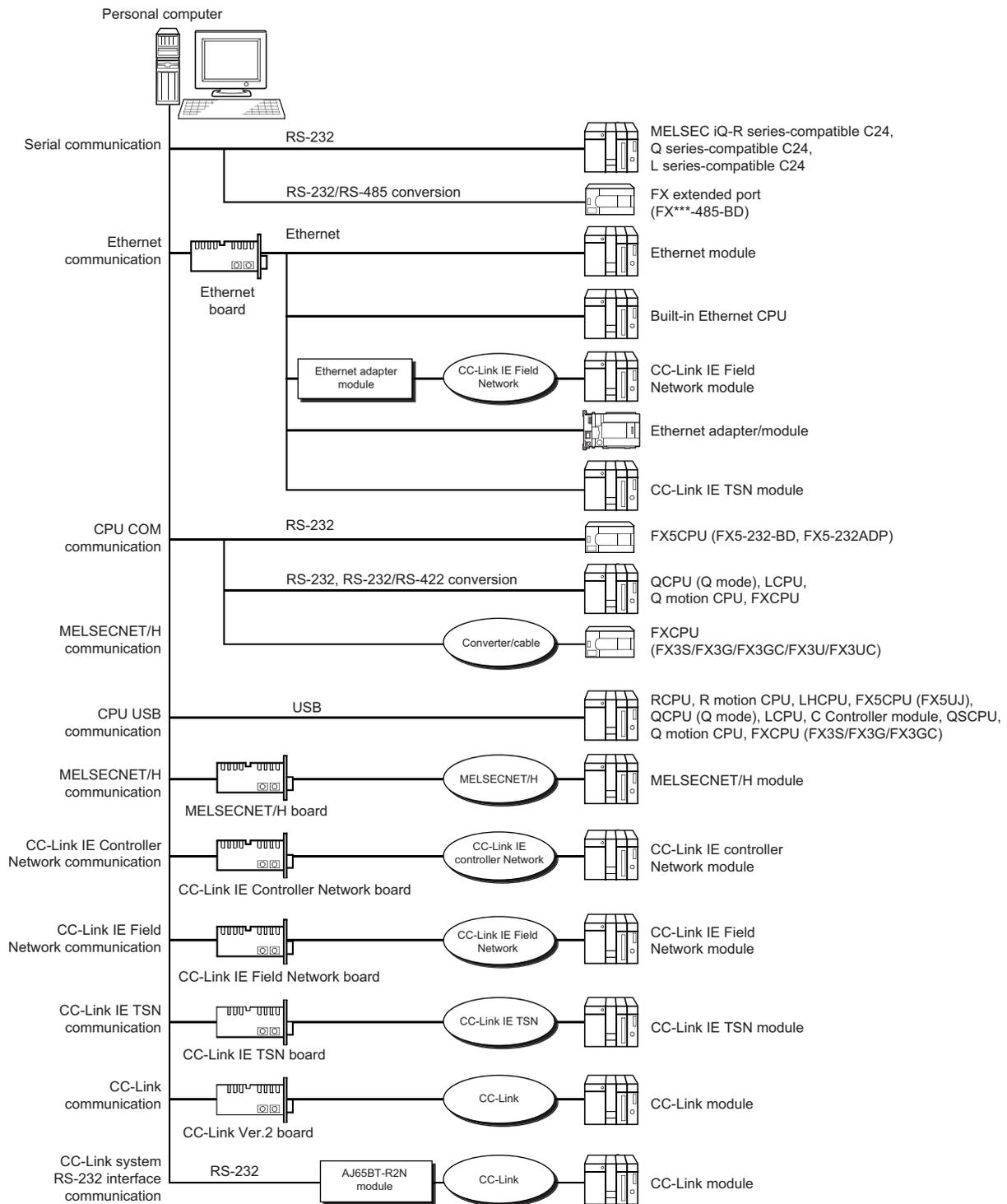
*1 Not supported when using Windows 10 Education.

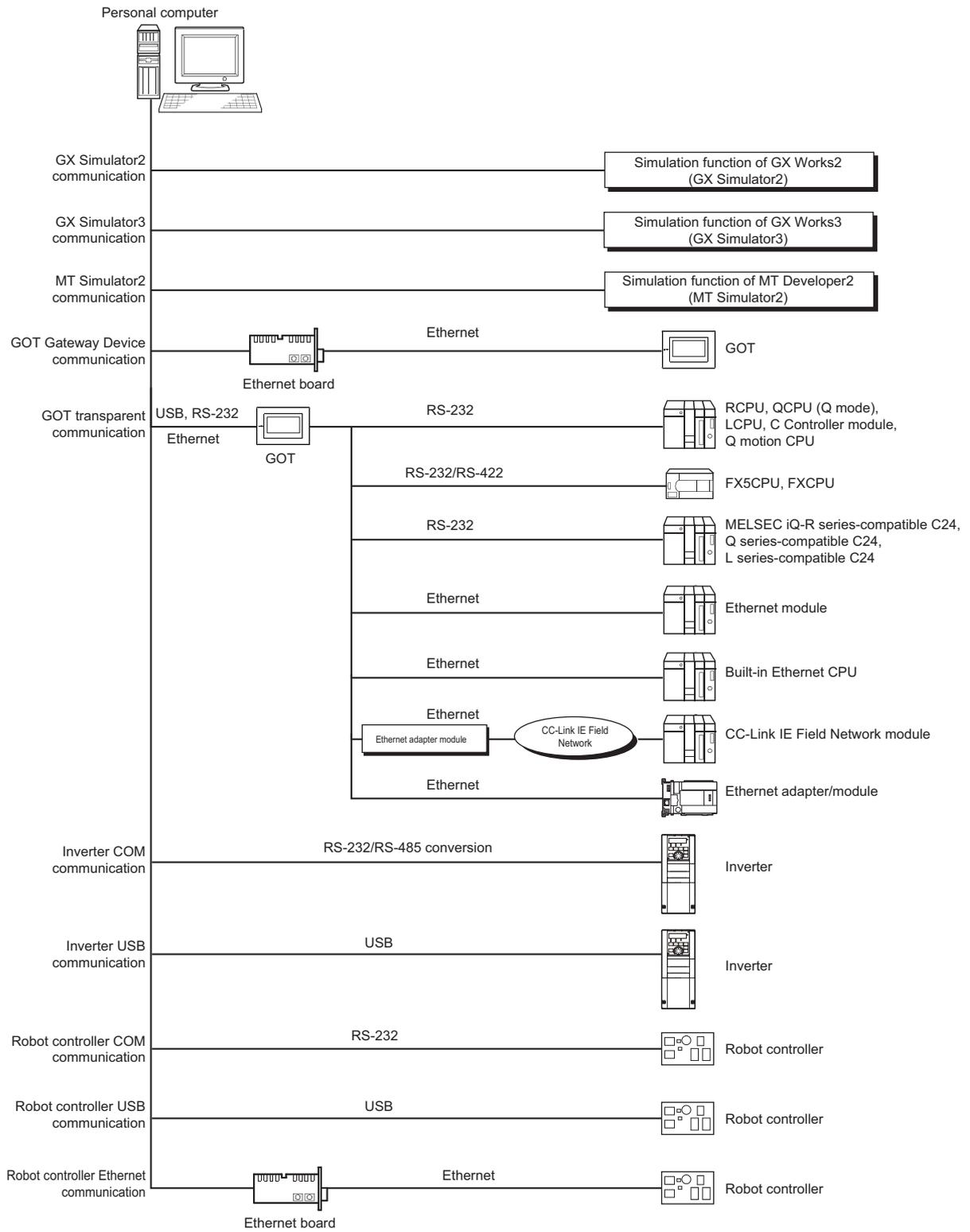
2.3 System Configuration List



For details on the available cables and considerations for communication, refer to the manual of each module.

System configuration examples of each communication route





2.4 Available Module Lists

The following shows available models in MX Component.

CPU module list

The following table shows available CPU modules.

Series	Module model	
RCPU	RnCPU	R00CPU, R01CPU, R02CPU, R04CPU, R04ENCPU, R08CPU, R08ENCPU, R16CPU, R16ENCPU, R32CPU, R32ENCPU, R120CPU, R120ENCPU
	RnPCPU	R08PCPU, R16PCPU, R32PCPU, R120PCPU
	RnSFPCPU	R08SFPCPU, R16SFPCPU, R32SFPCPU, R120SFPCPU
	RnPSFPCPU	R08PSFPCPU, R16PSFPCPU, R32PSFPCPU, R120PSFPCPU
RCCPU	R12CCPU-V	
R motion CPU	R16MTCPU, R32MTCPU, R64MTCPU	
MELSECWinCPU	R102WCPU-W	
LHCPU	L04HCPU, L08HCPU, L16HCPU, L32HCPU	
FX5CPU	FX5UCPU, FX5UCCPU, FX5UJCPU, FX5SCPU	
QCPU (Q mode)	Q00UJCPU, Q00UCPU, Q01UCPU, Q02UCPU, Q03UDCPU, Q03UDECPU, Q03UDVCPU, Q04UDHCPU, Q04UDEHCPU, Q04UDVCPU, Q04UDPVCPU, Q06UDHCPU, Q06UDEHCPU, Q06UDVCPU, Q06UDPVCPU, Q10UDHCPU, Q10UDEHCPU, Q12PRHCPU, Q13UDHCPU, Q13UDEHCPU, Q13UDVCPU, Q13UDPVCPU, Q20UDHCPU, Q20UDEHCPU, Q25PRHCPU, Q26UDHCPU, Q26UDEHCPU, Q26UDVCPU, Q26UDPVCPU, Q50UDEHCPU, Q100UDEHCPU	
LCPU	L02SCPU, L02CPU, L06CPU, L26CPU, L26CPU-BT	
QCCPU	Q12DCCPU-V (Basic mode) ^{*1} , Q12DCCPU-V (Extended mode), Q24DHCCPU-V, Q24DHCCPU-LS, Q24DHCCPU-VG, Q26DHCCPU-LS	
QSCPU	QS001CPU	
Q motion CPU	Q172DCPU, Q173DCPU, Q172DSCPU, Q173DSCPU	
FXCPU	FX3SCPU, FX3GCPU, FX3GCCPU, FX3UCPU, FX3UCCPU	

*1 The first five digits of the serial number is 12042 or higher are supported.

Considerations

■RnSFCPU

In order to protect the safety programmable controller system, a function to write data to the buffer memory or safety devices in safety mode cannot be executed.

■FX5CPU

- Multiple simultaneous communications cannot be established via the route specified by a different adapter when an adapter to be used for Ethernet direct connection is specified to establish a communication.
- Personal computer side adapter information of communication settings may need to reset depending on the change in personal computer environment or other MELSOFT application settings.

■Q00UJ/Q00U/Q01U/Q02UCPU

"Serial communication function compatible CPU" indicates Q00UJ/Q00U/Q01U/Q02UCPU.

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

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

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

■QSCPU

In order to protect the safety programmable controller system, a function to write data to the buffer memory or safety devices in safety mode cannot be executed.

■Built-in Ethernet CPU

- When resetting a CPU module during TCP/IP connection establishment (during opening) by using MX Component, a communication error or receive error occurs at subsequent communication.
In this case, perform the close processing in the application that uses MX Component, and perform the open processing again.
- The communication error may occur while establishing the Ethernet direct connection for the CPU, which communicates using Ethernet direct connection with another personal computer. In such case, retry the communication by resetting the CPU.

■FXCPU

- When an FXCPU is used, the TN devices (timer present values) or CN devices (counter present values) cannot be accessed if the device numbers specified are split across 199 or lower and 200 or higher.
- Since FXCPUs do not feature the PAUSE switch as CPU modules, an error is returned when remote pause is specified in SetCpuStatus.
- Note that specifying the first I/O number of a non-existing module and executing the WriteBuffer() method will not return an error.
- For the index registers (Z, V) of FXCPUs, data cannot be written to 2 or more consecutive points using WriteDeviceBlock(). (Data may be written to only one point.)

Module list

The following table shows the modules and boards that are available for communication with a CPU module in each communication type.

Communication type	Category	Module model
Serial communication	MELSEC iQ-R series-compatible C24	RJ71C24, RJ71C24-R2, RJ71C24-R4
	Q series-compatible C24	QJ71C24, QJ71C24N, QJ71C24N-R2, QJ71C24N-R4
	L series-compatible C24	LJ71C24, LJ71C24-R2
	FX extended port	FX3G-485-BD, FX3U-485-BD
Ethernet communication	MELSEC iQ-R series-compatible E71	RJ71EN71
	Q series-compatible E71	QJ71E71, QJ71E71-B2, QJ71E71-B5, QJ71E71-100
	L series-compatible E71	LJ71E71
	Ethernet adapter module	NZ2GF-ETB
	Ethernet adapter/module	FX3U-ENET-ADP, FX3U-ENET, FX5-CCLGN-MS, FX5-ENET, FX5-ENET/IP
MELSECNET/H communication	MELSECNET/H module	RJ71LP21 QJ71LP21
	MELSECNET/H board*1	Q80BD-J71LP21-25, Q80BD-J71LP21S-25, Q81BD-J71LP21-25, Q80BD-J71LP21G, Q80BD-J71BR11
CC-Link communication	CC-Link module	RJ61BT11
		QJ61BT11N
		LJ61BT11
		FX3U-64CCL
	CC-Link Ver.2 board*2	Q80BD-J61BT11N, Q81BD-J61BT11
CC-Link system RS-232 interface communication	AJ65BT-R2N module	AJ65BT-R2N
CC-Link IE Controller Network communication	CC-Link IE Controller Network module	RJ71GP21-SX
		QJ71GP21S-SX, QJ71GP21-SX
		LJ71GP21-SX
	CC-Link IE Controller Network board*3	Q80BD-J71GP21-SX, Q80BD-J71GP21S-SX, NZ81GP21-SX
CC-Link IE Field Network communication	CC-Link IE Field Network module	RJ71GF11-T2
		QJ71GF11-T2
		LJ71GF11-T2
		CC-Link IE Field Network board*4
CC-Link IE TSN communication	CC-Link IE TSN module	RJ71GN11-T2, RJ71GN11-SX
	CC-Link IE TSN board*5	NZ81GN11-T2, NZ81GN11-SX
GOT transparent communication	GOT2000	GT27, GT25
Inverter communication	A800	—
Robot controller communication	CR750	CR750-DCR751-D
	CRnD-700	CR1DA-7xx, CR2DA-7xx, CR3DA-7xx

*1 Applicable driver: SW0DNC-MNETH-B (Version 30G) or later

*2 Applicable driver: SW1DNC-CCBD2-B (Version 1.19V) or later

*3 Applicable driver: SW1DNC-MNETG-B (Version 1.22Y) or later

*4 Applicable driver: SW1DNC-CCIEF-B (Version 1.10L) or later

*5 Applicable driver: SW1DNN-CCIETBDM-B (Version 1.000A) or later

2.5 Details on the Communication Routes

This section explains details and the considerations for each communication route.

Serial communication

For the method for connecting to a serial communication module, refer to the manual of the serial communication module.

■Serial communication

- On any serial communication modules, remote "PAUSE" operation will result in an error for all connections.
- An FX extended port is required when performing the serial communication using FX3S, FX3G(C), or FX3U(C)CPU.

■About connection of applicable modules

When accessing a CPU module from a personal computer via serial communication modules, note that modules that can be connected to the personal computer directly are limited.

Even if a module cannot be directly connected to the personal computer, it may be applicable as the nth module in multi-dropped connection.

○: Applicable, ×: Not applicable

Module model	Interface	1:1 system configuration	Multi-dropped connection	
			1st module	nth module
RJ71C24	RS-232(CH1)	○	○	×
	RS-422/485(CH2)	×	×	○
RJ71C24-R2	RS-232(CH1)	○	×	×
	RS-232(CH2)	○	×	×
RJ71C24-R4	RS-422/485(CH1)	×	×	○
	RS-422/485(CH2)	×	×	○
QJ71C24	RS-232(CH1)	○	○	×
QJ71C24N	RS-422/485(CH2)	×	×	○
QJ71C24N-R2	RS-232(CH1)	○	×	×
	RS-232(CH2)	○ (Function version B or later)	×	×
QJ71C24N-R4	RS-422/485(CH1)	×	×	○
	RS-422/485(CH2)	×	×	○
LJ71C24	RS-232(CH1)	○	○	×
	RS-422/485(CH2)	×	×	○
LJ71C24-R2	RS-232(CH1)	○	×	×
	RS-232(CH2)	○	×	×
FX3G-485-BD FX3U-485-BD	RS-422/485	○	○	○

Ethernet communication

- For the method for connecting to an Ethernet module, refer to the manual of the Ethernet communication module.
- For the FX series Ethernet module, refer to the user's manuals for the FX series.
- For the method for connecting to an Ethernet Built-in CPU, refer to the manual of the Ethernet Built-in CPU.
- For the method for connecting to an Ethernet adapter module, refer to the manual of the Ethernet adapter module.
- For the method for connecting to an Ethernet adapter, refer to the manual of the Ethernet adapter.
- For the method for connecting to a CC-Link IE TSN module, refer to the manual of the CC-Link IE TSN module.

CPU COM communication

■Cable for connection of an QCPU (Q mode) and LCPU

The following cable is required for communications between the personal computer and of QCPU (Q mode).

RS-232 cable

QC30R2 (Personal computer connector: 9-pin D-sub connector)

RS-232 adaptor (L6ADP-R2) is required when connecting to the LCPU.

■Cable for connection of an FX5CPU

The following cable is required for communications between the personal computer and of FX5CPU.

Function expansion board and special adapter	RS-232 cable
<ul style="list-style-type: none"> • FX5-232ADP • FX5-232-BD 	FX-232CAB-1

For details, refer to the following:

 MELSEC iQ-F FX5U User's Manual (Hardware)

■Cables for connection of an FXCPU

The following cable is required for communications between the personal computer and of FXCPU.

Personal computer side (RS-232 cable)	RS-232/RS-422 converter	CPU module side (RS-422 cable)
<ul style="list-style-type: none"> • F2-232CAB (25-pin D-sub ↔ 25-pin D-sub) • F2-232CAB-1 (9-pin D-sub ↔ 25-pin D-sub) • F2-232CAB-2 (Half pitch ↔ 25-pin D-sub) • AC30N2A (25-pin ↔ 25-pin) 	<ul style="list-style-type: none"> • FX-232AW • FX-232AWC • FX-232AWC-H 	FX-422CAB0 (1.5m)

■Cables for connection of Q motion CPU

For communications between the personal computer and Q motion CPU, use the cables as indicated below.

 Page 33 Cable for connection of an QCPU (Q mode) and LCPU

■Converter/Cable (FXCPU compatible) for connecting to a USB on personal computer

System configuration

- FX-USB-AW
- USB cable (included)

Point

- Using a USB cable for the first time
Install the driver from the CD included with FX-USB-AW and FX3U-USB-BD.
- Considerations and restrictions
For the considerations and restrictions relating to FX-USB-AW, refer to the user's manual included.

■USB cables and function expansion boards (compatible with FX3U, FX3UCCPU)

System configuration

- FX3U-USB-BD
- USB cable (included)

Point

- Using a USB cable for the first time
Install the driver from the CD included with FX-USB-AW and FX3U-USB-BD.
- Considerations and restrictions
For the considerations and restrictions relating to FX3U-USB-BD, refer to the user's manual included.
- PLC parameter
When "Operate Communication Setting" is selected on the [PLC System (2)] tab of PLC parameter in GX Works2, the corresponding port cannot communicate with the CPU module. Write the setting in which "Operate Communication Setting" is cleared from the built-in programming port of the CPU module.
For the method of PLC parameter settings, refer to the following:
 GX Works2 Version 1 Operating Manual (Common)

■USB cables (compatible with FX3S, FX3G, and FX3GCCPU)

For details on cables, refer to the following:

-  FX3S Series User's Manual- Hardware Edition
-  FX3G Series User's Manual- Hardware Edition
-  FX3GC Series User's Manual- Hardware Edition

Point

- Using a USB cable for the first time
Install the USB driver. ( Page 669 Installing a USB Driver)

■RS-422 function expansion board for FXCPUs

Series	Function expansion board
FX3U, FX3UC (FX3UC-32MT-LT and FX3UC-32MT-LT-2 only)	FX3U-422-BD
FX3S, FX3G	FX3G-422-BD

Point

- PLC parameter
When "Operate Communication Setting" is selected on the [PLC System (2)] tab of PLC parameter in GX Works2, the corresponding port cannot communicate with the CPU module. Write the setting in which "Operate Communication Setting" is cleared from the built-in programming port of the CPU module.
For the method of PLC parameter settings, refer to the following:
 GX Works2 Version 1 Operating Manual (Common)

■RS-232 cable and function expansion board (special adaptor) for FXCPUs

Serial port shape of personal computer	Series	Function expansion board and special adapter	RS-232 cable
9-pin D-sub	FX3U, FX3UC	FX3U-232-BD ^{*1}	FX-232CAB-1
		Function expansion board (FX3U-***-BD) ^{*2} + FX3U-232ADP	
	FX3G, FX3GC	FX3G-232-BD ^{*3}	FX-232CAB-1
		FX3G-CNV-ADP ^{*4} + FX3U-232ADP	
	FX3S	FX3G-232-BD	FX-232CAB-1
		FX3S-CNV-ADP + FX3U-232ADP	
Half pitch 14-pin	FX3U, FX3UC	FX3U-232-BD	FX-232CAB-2
		Function expansion board (FX3U-***-BD) ^{*2} + FX3U-232ADP	
	FX3G, FX3GC	FX3G-232-BD ^{*3}	FX-232CAB-2
		FX3G-CNV-ADP ^{*4} + FX3U-232ADP	
	FX3S	FX3G-232-BD	FX-232CAB-2
		FX3S-CNV-ADP + FX3U-232ADP	
25-pin D-sub	FX3U, FX3UC	FX3U-232-BD	F2-232CAB-1
		Function expansion board (FX3U-***-BD) ^{*2} + FX3U-232ADP	
	FX3G, FX3GC	FX3G-232-BD ^{*3}	F2-232CAB-1
		FX3G-CNV-ADP ^{*4} + FX3U-232ADP	
	FX3S	FX3G-232-BD	F2-232CAB-1
		FX3S-CNV-ADP + FX3U-232ADP	

*1 For FX3UC series, only FX3UC-32MT-LT and FX3UC-32MT-LT-2 can be used.

*2 *** of the function expansion board (FX3U-***-BD) indicates 232, 485, 422, USB, CNV, or 8AV.

Function expansion board (FX3U-***-BD) is not required for FX3UC (D, DS, DSS) series.

The serial communication is also available for FX3U-485-BD. For setting for the serial communication, refer to the following:

 Page 72 FX extended port

*3 FX3GC series cannot be used.

*4 ADP (FX3G-CNV-ADP) for FX3U adapter connection is not required for FX3GC series.

Point

- PLC parameter

When "Operate Communication Setting" is selected on the [PLC System (2)] tab of PLC parameter in GX Works2, the corresponding port cannot communicate with the CPU module. Write the setting in which "Operate Communication Setting" is cleared from the built-in programming port of the CPU module.

For the method of PLC parameter settings, refer to the following:

 GX Works2 Version 1 Operating Manual (Common)

CPU USB communication

The following table shows the available USB cables.

Product name	Module model
USB cable (USB A type — USB B type)	AU230
USB cable (USB A type — USB miniB type)	KU-AMB530
	U2C-M30BK
	MR-J3USBCBL3M
	GT09-C30USB-5P

Point

- Using a USB cable for the first time
Install a USB driver. For the installation method, refer to the following:
 Page 669 Installing a USB Driver
- Considerations and restrictions
For the considerations and restrictions when accessing a CPU module, refer to the following:
 Page 674 For CPU modules

MELSECNET/H communication

- A MELSECNET/H board can be used.
- Use the communication driver SW0DNC-MNETH-B or later.
The other communication drivers cannot be used.
- For details on the supported operating system of the network board to be used for communication, refer to the manual of each network board.

CC-Link IE Controller Network communication

- A CC-Link IE Controller Network board can be used.
- Use the communication driver SW1DNC-MNETG-B or later.
The other communication drivers cannot be used.
- For details on the supported operating system of the network board to be used for communication, refer to the manual of each network board.

CC-Link IE Field Network communication

- A CC-Link IE Field Network board can be used.
- Use the communication driver SW1DNC-CCIEF-B or later.
The other communication drivers cannot be used.
- For details on the supported operating system of the network board to be used for communication, refer to the manual of each network board.

CC-Link IE TSN communication

- A CC-Link IE TSN board can be used.
- Use the communication driver SW1DNN-CCIETBDM-B or later.
The other communication drivers cannot be used.
- For details on the supported operating system of the network board to be used for communication, refer to the manual of each network board.

CC-Link communication

- A CC-Link Ver.2 board can be used.
- Use the communication driver SW1DNC-CCBD2-B or later.
The other communication drivers cannot be used.
- For details on the supported operating system of the network board to be used for communication, refer to the manual of each network board.
- Use the CC-Link master station module the software version of which is "N" or later.
Modules of software version "M" or earlier do not operate normally.

CC-Link system RS-232 interface communication

- Use the CC-Link master station module the software version of which is "N" or later.
Modules of software version "M" or earlier do not operate normally.
- For details on the communication with an AJ65BT-R2N module, refer to the manual of a CC-Link system RS-232 interface module.

GX Simulator2 communication

When performing GX Simulator2 communication, use GX Works2.

GX Simulator3 communication

When performing GX Simulator3 communication, use GX Works3.

MT Simulator2 communication

When performing MT Simulator2 communication, use MT Developer2.

This function is supported by a Q motion CPU only.

■ Considerations for performing MT Simulator2 communication

After installing MX Component, install MT Developer2.

GOT gateway device communication

For details on the GOTs which support the gateway function and the settings, refer to the manual of the gateway function for GOT2000 series used.

GOT transparent communication

For details on the connection and settings, refer to the connection manuals for GOT2000 series.

(Mitsubishi Products), (Non-Mitsubishi Products 1), (Non-Mitsubishi Products 2), (Microcomputer, MODBUS/Fieldbus Products, Peripherals)

Inverter (COM/USB) communication

For the connection cables, refer to the manual of inverter used.

Robot controller(COM/USB/Ethernet) communication

For the connection cables, refer to the following:

 CR750/700/500 series RT ToolBox2 / RT ToolBox2 mini User's Manual

■Considerations when connecting a robot controller with a USB

For the considerations when connecting a robot controller with a USB, refer to the following:

 CR750/700/500 series RT ToolBox2 / RT ToolBox2 mini User's Manual

When connecting a robot controller with a USB, an error does not occur for Open method of the control even when the robot controller cannot be used.

If an error code 106 (connection is disconnected) occurred in the method after executing Open, execute Open again after executing Close.

■Multiplex communication

Do not perform multiplex communication for one robot controller.

3 USER APPLICATION CREATING PROCEDURES

This chapter shows the procedure for creating a user program.

1. Install MX Component
2. Grasp a control to be used.
3. Select the communication setting method
4. Create a new user program
5. Add the controls to a user program
6. Set the communication settings
7. Create a program (creating a label, coding, and debugging)

3

3.1 Installing MX Component

Install MX Component on a personal computer. For the installation method, refer to the following:

 MX Component Version 5 Installation Instructions

Precautions

- When performing overwrite installation, install the software in the same folder where it is installed previously.
- When downloading the installer, save the installer to a directory which does not include any space and execute it.
- When uninstalling MX Component, the item may remain in the start menu. In this case, restart the personal computer.

Checking the Version

The version of MX Component can be checked by using the [Help] menu of each utility.

Operating procedure

Select [Help] ⇒ [About].

Opening the manual

e-Manual can be opened by using the [Help] menu of each utility.

Operating procedure

Select [Help] ⇒ [Help].

3.2 Grasping a Control to be Used

When developing a user program by using MX Component, controls to be used differs depending on the content of the processing and communication setting method.

Check if the controls are suitable for the contents of development.

Contents of processing	Supported language	Communication setting	Types of control	Control name (Upper: for 32-bit, lower: for 64-bit)
Reading/writing device data	<ul style="list-style-type: none"> • Visual Basic.Net • Visual C# • Visual C++ • VBA 	Set in Communication Settings Utility.	Act control	ActUtilType ActUtilType64
		Set in a program.		ActProgType ActProgType64
		Use the same control regardless of the communication setting method.		ActSupportMsg ActSupportMsg64
Troubleshooting				
Transferring a logging file		Set in Communication Settings Utility.	Act Control (logging file transfer)	ActUtilDataLogging ActUtilDataLogging64
		Set in a program.		ActProgDataLogging ActProgDataLogging64
Reading/writing label data	<ul style="list-style-type: none"> • Visual Basic.Net • Visual C# 	Set in Communication Settings Utility only.	.Net control	DotUtilType DotUtilType64

For the difference between using Communication Settings Utility and the program as the "communication setting," refer to the following:

☞ Page 40 Selecting a Communication Setting Method

For the method to incorporate the Act control and .Net control into the development environment, refer to the following:

☞ Page 42 ADDING CONTROLS TO USER PROGRAM

3.3 Selecting a Communication Setting Method

Select either type for the communication setting method (development type), a method to use Communication Settings Utility (utility setting type) or to use a program (program setting type).

The following table shows the differences between the utility setting type and program setting type.

Development type	Feature	Specification method in a user program	Applicable control
Utility setting type	A user program can be created with Communication Settings Utility without being aware of the complicated parameters of each communication. Set the communication settings using Communication Setting Wizard. Each communication setting is saved in a personal computer by adding a logical station number.	Specify the logical station number set with Communication Setting Wizard in the property of the Act control and .Net control, or within the user program.	<ul style="list-style-type: none"> • ActUtilType • ActUtilDataLogging • ActSupportMsg • DotUtilType • ActUtilType64 • ActUtilDataLogging64 • ActSupportMsg64 • DotUtilType64
Program setting type	A user program can be created without using Communication Settings Utility. Communication settings can be changed flexibly in the user program.	Set this type by entering the properties on the property window of the Act control directly or within the user program. The required settings for the properties differ depending on the Act control.	<ul style="list-style-type: none"> • ActProgType • ActProgDataLogging • ActSupportMsg • ActProgType64 • ActProgDataLogging64 • ActSupportMsg64

3.4 Adding a Control to a User Program

To develop a user program that uses MX Component, a control of MX Component needs to be added to the user program. The method to add a control differs depending on the development language and types of control to be used. For details, refer to the following:

☞ Page 42 ADDING CONTROLS TO USER PROGRAM

3.5 Creating a New User Program

Create a user program according to the development language to be used.

Development language	Creation method
VBA	Create a file in Excel or Access, and start Visual Basic Editor.
Visual Basic .NET Visual C# Visual C++	Start Visual Studio, and create a project file.

3.6 Communication Setting

For setting with the utility setting type, refer to the following:

☞ Page 54 Setting with Utilities

For setting with the program setting type, refer to the following:

☞ Page 65 Setting in Programs

3.7 Programming

Create a program to communicate with a program controller by using a property and function of a control.

In addition, create a label used in the program by using a label utility.

When debugging the created user program, a device and buffer memory can be checked easily by using PLC Monitor Utility.

For details, refer to the following:

☞ Page 115 PROGRAMMING

4 ADDING CONTROLS TO USER PROGRAM

When using the functions and properties of MX Component in a user program, controls need to be added to the user program. The method to add controls differs depending on the development language and type of the controls.

Development language	Type of control	Using control	Details
VBA	Act control (for 32-bit, for 64-bit)	Use a control by creating an object.	Page 42 When using a control by creating an object
	Act control (for 32-bit)* ¹	Use a control by pasting the control to a form.	Page 43 When using a control by pasting a control to a form
Visual Basic .NET Visual C#	Act control (for 32-bit, for 64-bit)	Use a control by creating an object.	Page 46 When using a control by creating an object
	Act control (for 32-bit)* ¹	Use a control by pasting the control to a form.	Page 48 When using a control by pasting a control to a form
	.Net control (for 32-bit, for 64-bit)	Use a control by creating an object.	Page 46 When using a control by creating an object
Visual C++	Act control (for 32-bit, for 64-bit)	Use a control by creating an object.	Page 49 When using a control by creating an object
	Act control (for 32-bit)* ¹	Use a control by pasting the control to a form.	Page 51 When using a control by pasting a control to a form

*1 Only Visual Studio 2019 is supported.

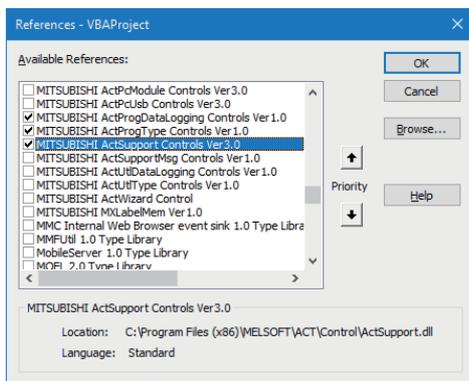
4.1 When using VBA

The following shows the setting operation when using VBA.

When using a control by creating an object

Operating procedure

1. Start Visual Basic Editor.
 - When using Microsoft Excel 2019: Select [Visual Basic] on the [Developer] tab.
 - When using Microsoft Access 2019: Select [Visual Basic] on the [Database Tools] tab.
2. Select "References" on the [Tools] tab of Visual Basic Editor.
3. Select a control of "Development type" to be used (☞ Page 43 Applicable Controls (When using a control by creating an object)) in the "Available References" screen, and click the [OK] button.



■Applicable Controls (When using a control by creating an object)

The following table shows applicable controls by the development type.

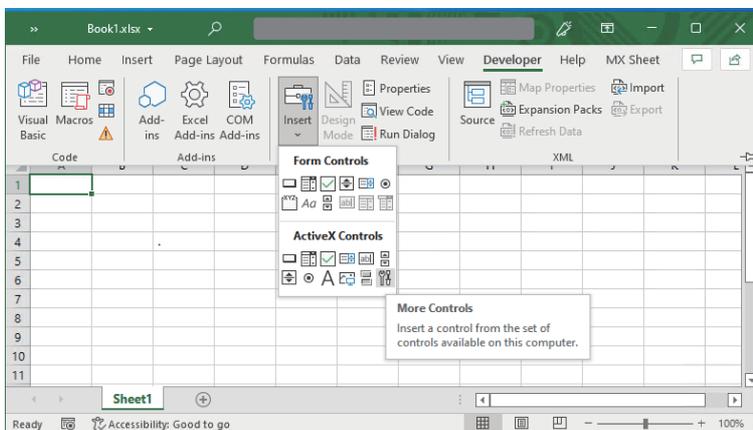
Development type	Applicable control
Utility setting type (for 32-bit)	MITSUBISHI ActUtilType Controls Ver1.0 MITSUBISHI ActSupportMsg Controls Ver1.0
Utility setting type (for 64-bit)	ActUtilType64 Control ActSupportMsg64 Control
Program setting type (for 32-bit)	MITSUBISHI ActProgType Controls Ver1.0 MITSUBISHI ActSupportMsg Controls Ver1.0
Program setting type (for 64-bit)	ActProgType64 Control ActSupportMsg64 Control

When using a control by pasting a control to a form

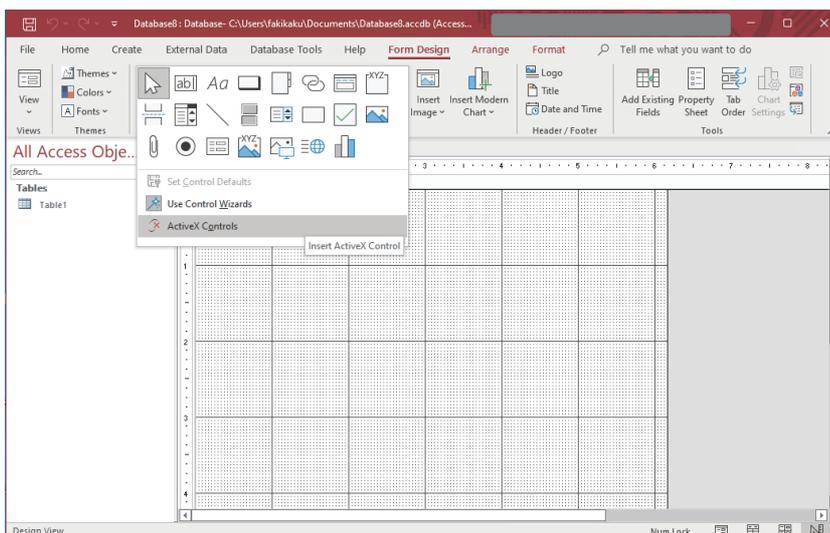
Operating procedure

1. Add ActiveX control to the screen.

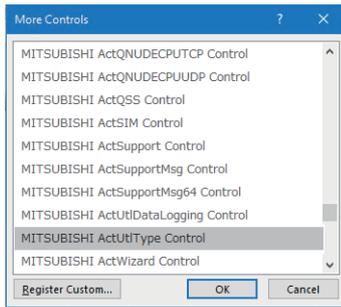
- When using Microsoft Excel 2019: Select  in [Insert] on the [Developer] tab.



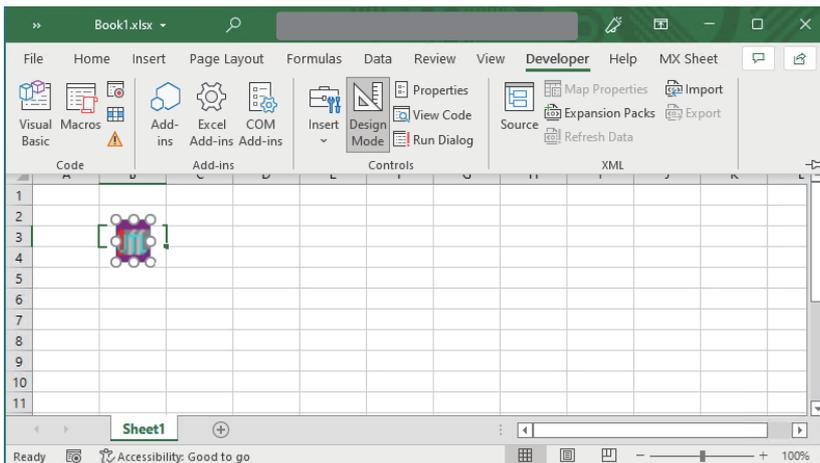
- When using Microsoft Access 2019: Select [Controls] on the [Form Design] tab ⇨ [ActiveX Controls].



2. Select a control of "Development type" to be used ( Page 44 Applicable Controls (When pasting a control to a form)) in the "More Controls" screen, and click the [OK] button.



3. Paste the selected Act control to the sheet.



4. Start Visual Basic Editor.

- When using Microsoft Excel 2019: Select [Visual Basic] on the [Developer] tab.
- When using Microsoft Access 2019: Select [View Code] on the [Design] tab.

■Applicable Controls (When pasting a control to a form)

The following table shows applicable controls by the development type.

Development type	Applicable control
Utility setting type (for 32-bit)	MITSUBISHI ActUiType Control MITSUBISHI ActSupportMsg Control
Program setting type (for 32-bit)	MITSUBISHI ActProgType Control MITSUBISHI ActSupportMsg Control

Considerations for using Microsoft Excel

■When controls cannot be pasted to Excel

This symptom occurs if the cache file (temporary file) of Excel remains.
In such a case, perform the operation in the following procedure.

Operating procedure

1. Close Excel.
2. Delete "*.exd" in the Excel 8.0 folder of the temp folders.
 - The location of the temp folder differs according to the operating system.
 - When the corresponding folder and file are not displayed, set the settings in the folder option setting to display all files and folders.
3. Restart Excel.

■Resizing of Act control

Act control can be resized in Excel, however this does not affect the operation of MX Component.
To restore the size, set the Height and Width properties of Act control to "24."

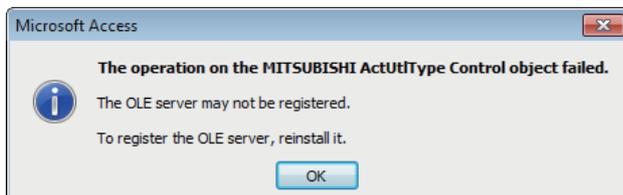
■Considerations for using Excel VBA

Do not set the page feed preview function in the application that uses Excel VBA.
A memory leak or operating system basic operation (file operation, printing, or the like) failure may occur.

Considerations for using Microsoft Access

■When a following error message appears after pasting the Act control to an Access form and double-clicking the Act control or selecting the custom control

The following error message appears. However, this does not affect the operation of Act control.
(An error message other than the following message may be displayed.)



■When the displayed property name is collapsed after pasting Act control and displaying property

This symptom only occurs on the display, and this does not affect the functions of the property.

■Resizing of Act control

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

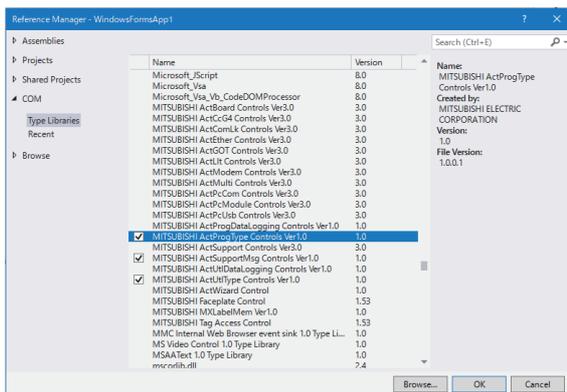
4.2 When using Visual Basic .NET, Visual C#

The following shows the setting operation when using Visual Basic .NET, Visual C#.

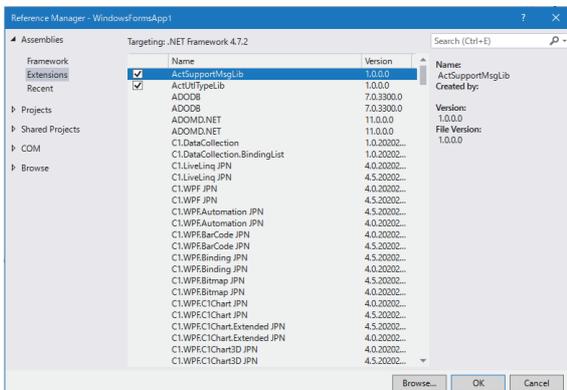
When using a control by creating an object

Operating procedure

1. Start Visual Studio.
2. Select [Project] ⇒ [Add Reference].
3. Add a control of "Development type" to be used ( Page 47 Applicable Controls (When using a control by creating an object)).
 - When using Act control: Select a control from "COM" in the "Reference Manager" screen.



- When using .Net control: Select a control from "Extensions" in "Assemblies" in the "Reference Manager" screen.



4. When developing a program by using the program setting type, select [Project] ⇒ [Add Existing Item], and add include files ( Page 47 Include file (When using Visual Basic .NET, Visual C#)). The include files are stored in the following folder at the time of installation.
(User-specified folder)\Act\Include
5. Create an instance of class for each control to be used. ( Page 47 Class created for each control)

■Applicable Controls (When using a control by creating an object)

The following table shows applicable controls by the development type.

Development type	Type of control	Applicable control
Utility setting type (for 32-bit)	Act control	MITSUBISHI ActUtilType Controls Ver1.0 MITSUBISHI ActUtilDataLogging Controls Ver1.0 MITSUBISHI ActSupportMsg Controls Ver1.0
	.Net control	MITSUBISHI DotUtilType Component
Utility setting type (for 64-bit)	Act control	ActUtilType64 Control ActUtilDataLogging64 Control ActSupportMsg64 Control
	.Net control	DotUtilType64 Control
Program setting type (for 32-bit)	Act control	MITSUBISHI ActProgType Controls Ver1.0 MITSUBISHI ActProgDataLogging Controls Ver1.0 MITSUBISHI ActSupportMsg Controls Ver1.0
	Act control	ActProgType64 Control ActProgDataLogging64 Control ActSupportMsg64 Control

■Class created for each control

Control	Class
ActUtilType Control	ActUtilTypeLib.ActUtilTypeClass
ActUtilDataLogging Control	ActUtilDataLoggingLib.ActUtilDataLoggingClass
ActSupportMsg Control	ActSupportMsgLib.ActSupportMsgClass
MITSUBISHI DotUtilType Component	MITSUBISHI.Component.DotUtilType
ActUtilType64 Control	ActUtilType64Lib.ActUtilType64Class
ActUtilDataLogging64 Control	ActUtilDataLogging64Lib.ActUtilDataLogging64Class
ActSupportMsg64 Control	ActSupportMsg64Lib.ActSupportMsg64Class
DotUtilType64 Control	DotUtilType64.DotUtilType64
ActProgType Control	ActProgTypeLib.ActProgTypeClass
ActProgDataLogging Control	ActProgDataLoggingLib.ActProgDataLoggingClass
ActSupportMsg Control	ActSupportMsgLib.ActSupportMsgClass
ActProgType64	ActProgType64Lib.ActProgType64Class
ActProgDataLogging64	ActProgDataLogging64Lib.ActProgDataLogging64Class
ActSupportMsg64	ActSupportMsg64Lib.ActSupportMsg64Class

■Include file (When using Visual Basic .NET, Visual C#)

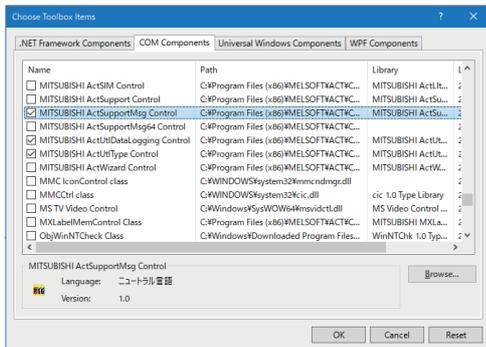
Development language	Include file
Visual Basic .NET	ActDefine.vb
Visual C#	ActDefine.cs

When using a control by pasting a control to a form

This function only supports Window Form. WPF and UWP are not supported.

Operating procedure

1. Start Visual Studio.
2. Right-click in "Toolbox" and select [Choose Items].
3. Select a control of "Development type" to be used ( Page 48 Applicable Controls (When pasting a control to a form)) from the [COM Components] tab in the "Choose Toolbox Items" screen, click the [OK] button.



4. Paste the control added to the toolbox to a form.
5. Select [Project] ⇒ [Add Existing Item], and add included files ( Page 47 Include file (When using Visual Basic .NET, Visual C#)). The include files are stored in the following folder at the time of installation.
(User-specified folder)\Act\Include

■Applicable Controls (When pasting a control to a form)

The following table shows applicable controls by the development type.

Development type	Applicable control
Utility setting type (for 32-bit)	MITSUBISHI ActUtilType Controls MITSUBISHI ActUtilDataLogging Controls MITSUBISHI ActSupportMsg Controls
Program setting type (for 32-bit)	MITSUBISHI ActProgType Controls MITSUBISHI ActProgDataLogging Controls MITSUBISHI ActSupportMsg Controls

4.3 When using Visual C++

When using a control with Visual C++, MFC must be installed in advance.
The following shows the setting operation when using Visual C++.

When using a control by creating an object

Operating procedure

1. Start Visual Studio.
2. Set required include files for "Development type" to be used by referring the following table. (☞ Page 50 Setting the include file (for Visual C++)) The include files are stored in the following folder at the time of installation.
(User-specified folder)\Act\Include

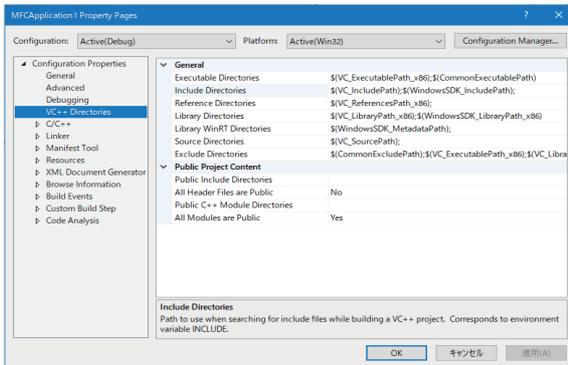
Development type	Required include file
Utility setting type (for 32-bit)	ActUtilType_i.c ActUtilType_i.h ActUtilDataLogging_i.c ActUtilDataLogging_i.h ActSupportMsg_i.c ActSupportMsg_i.h
Utility setting type (for 64-bit)	ActUtilType64_i.c ActUtilType64_i.h ActUtilDataLogging64_i.c ActUtilDataLogging64_i.h ActSupportMsg64_i.c ActSupportMsg64_i.h
Program setting type (for 32-bit)	ActProgType_i.c ActProgType_i.h ActProgDataLogging_i.h ActProgDataLogging_i.c ActSupportMsg_i.c ActSupportMsg_i.h ActDefine.h
Program setting type (for 64-bit)	ActProgType64_i.c ActProgType64_i.h ActProgDataLogging64_i.h ActProgDataLogging64_i.c ActSupportMsg64_i.c ActSupportMsg64_i.h ActDefine.h

3. Create instances for each control to be used. For the creation method of instance, refer to the sample program for Visual C++.

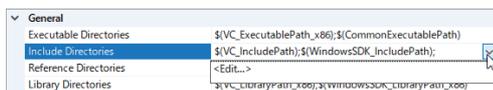
■Setting the include file (for Visual C++)

Operating procedure

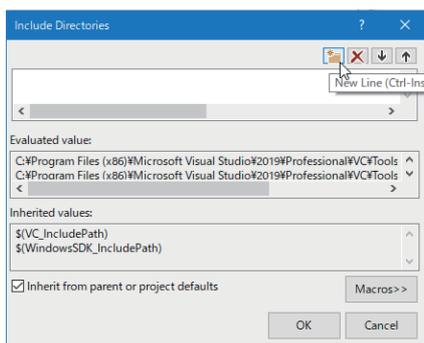
1. Start Visual Studio and select [Project] ⇒ [Properties].
2. Select [Configuration Properties] ⇒ [VC++ Directories] on the navigation pane displayed on the left side of the screen.



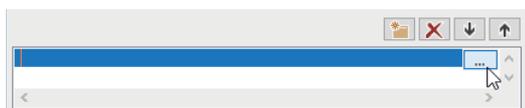
3. Select "Include Directories" displayed on the right side of the screen and click the [▼] button, then select <Edit...>.



4. Click the [New Line] button.



5. Click the [...] button.

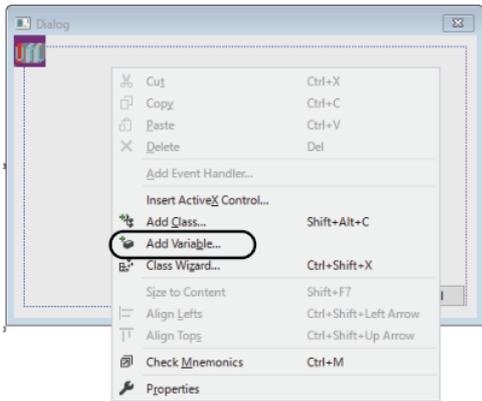


6. On the "Select Directory" screen, select the folder in which the include files are stored.
The include files are stored in the following folder at the time of installation.
(User-specified folder)\Act\Include

When using a control by pasting a control to a form

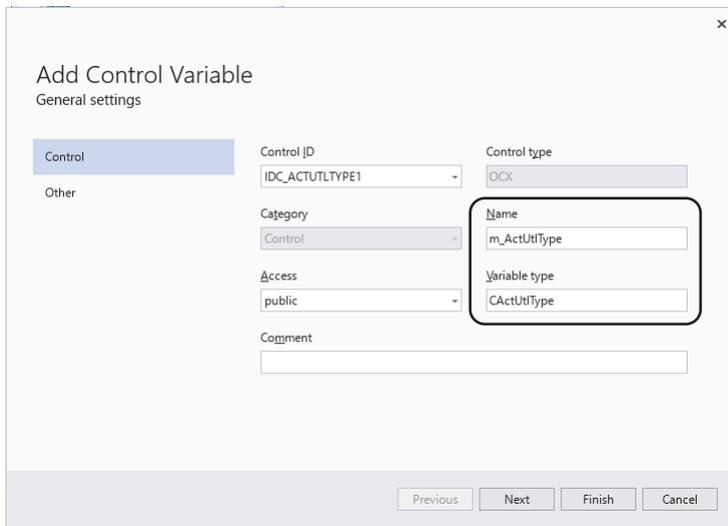
Operating procedure

1. Start Visual Studio.
2. Right-click the form and select "Insert ActiveX Control."
3. Select a control of "Development type" to be used ( Page 53 Applicable Controls (When pasting a control to a form)), and click the [OK] button.
4. Right-click the form and select "Add Variable."

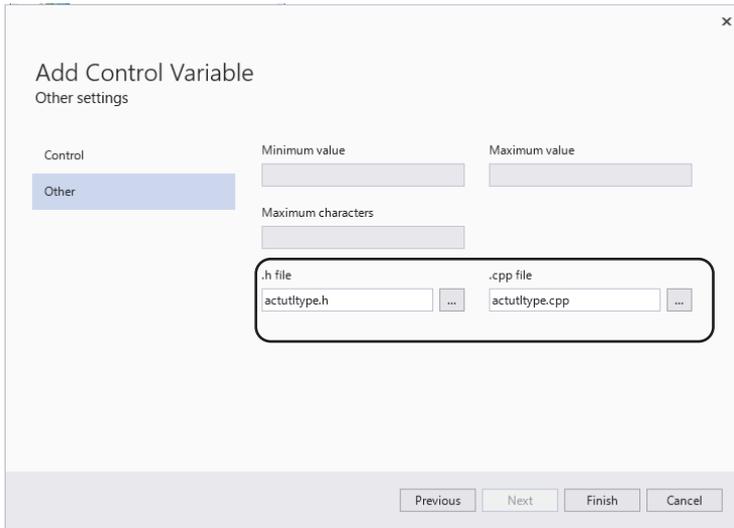


5. Enter "Name" and "Variable type" ( Page 53 Variable type) in "General settings" of the "Add Control Variable" screen, and click the [Next] button.

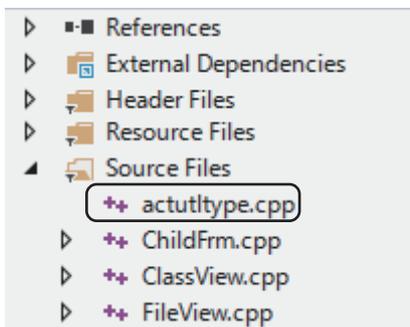
For "Name," enter the name of an object to be used in the program.



6. Enter ".h file" and ".cpp file" (☞ Page 53 .h file and .cpp file) in "Other settings" of the "Add Control Variable " screen, and click the [Finish] button.



7. Make sure that ".h file" and ".cpp file" set in the step 6 are created in Solution Explorer.



8. When using ActProgType or ActProgDataLogging, select [Project] ⇨ [Add Existing Item], and add an include file "ActDefine.h." The include file is stored in the following folder at the time of installation.
(User-specified folder)\Act\Include

Point 

".h file" created in the step 6 from 'When pasting a control to the form (☞ Page 51 When using a control by pasting a control to a form)' may not be created properly. (☞ Page 677 When a header file is not created properly at the time of pasting a control to a form with Visual C++)

■Applicable Controls (When pasting a control to a form)

The following table shows applicable controls by the development type.

Development type	Applicable control
Utility setting type (for 32-bit)	MITSUBISHI ActUtilType Controls MITSUBISHI ActUtilDataLogging Controls MITSUBISHI ActSupportMsg Controls
Program setting type (for 32-bit)	MITSUBISHI ActProgType Controls MITSUBISHI ActProgDataLogging Controls MITSUBISHI ActSupportMsg Controls

■Variable type

The following list shows "Variable type" for each control to be entered in the "Add Control Variable" screen.

Control	Variable type
ActUtilType	CActUtilType
ActUtilDataLogging	CActUtilDataLogging
ActSupportMsg	CActSupportMsg
ActProgType	CActProgType
ActProgDataLogging	CActProgDataLogging

■.h file and .cpp file

The following table shows ".hfile" and ".cpp file" for each control to be entered in the "Add Control Variable" screen.

Control	.h file	.cpp file
ActUtilType	actutltype.h	actutltype.cpp
ActUtilDataLogging	actutldatalogging.h	actutldatalogging.cpp
ActSupportMsg	actsupportmsg.h	actsupportmsg.cpp
ActProgType	actprogtype.h	actprogtype.cpp
ActProgDataLogging	actprogdatalogging.h	actprogdatalogging.cpp

5 COMMUNICATION SETTING

5.1 Setting with Utilities

This section explains the procedure for communication by using the Communication Settings Utility, and how to operate and set Communication Settings Utility.

Operation procedure

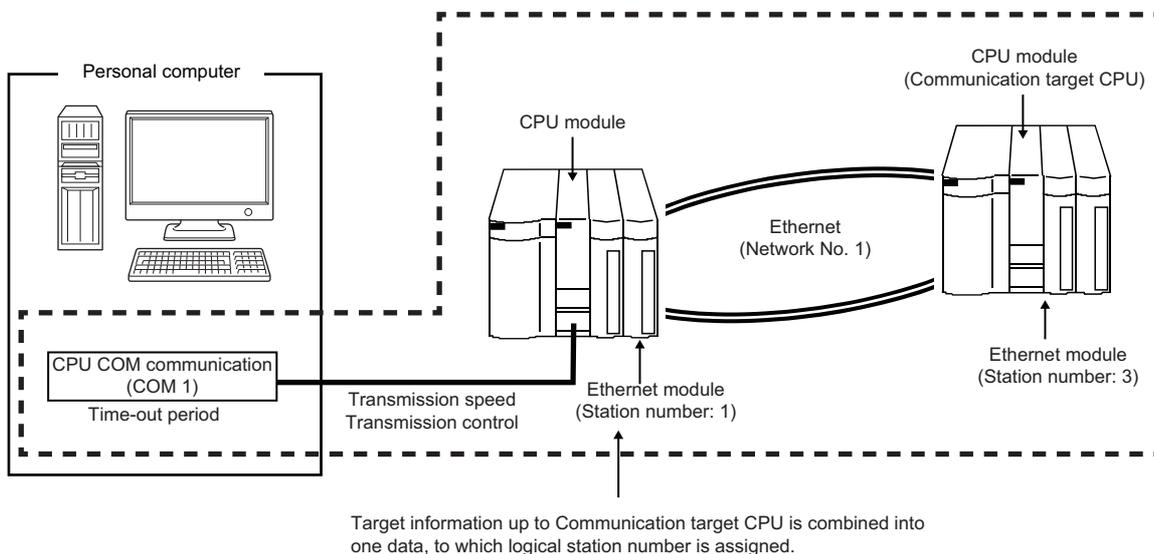
1. Create settings for each logical station number by using Communication Settings Utility.
2. Specify the logical number to the ActLogicalStationNumber property in a user program.

Logical station number

The logical station number is the logical number assigned to the data that is the compilation of the connection-destination information necessary for opening a communication line into a single data by using Communication Settings Utility.

Ex.

For CPU COM communication



Starting Communication Settings Utility

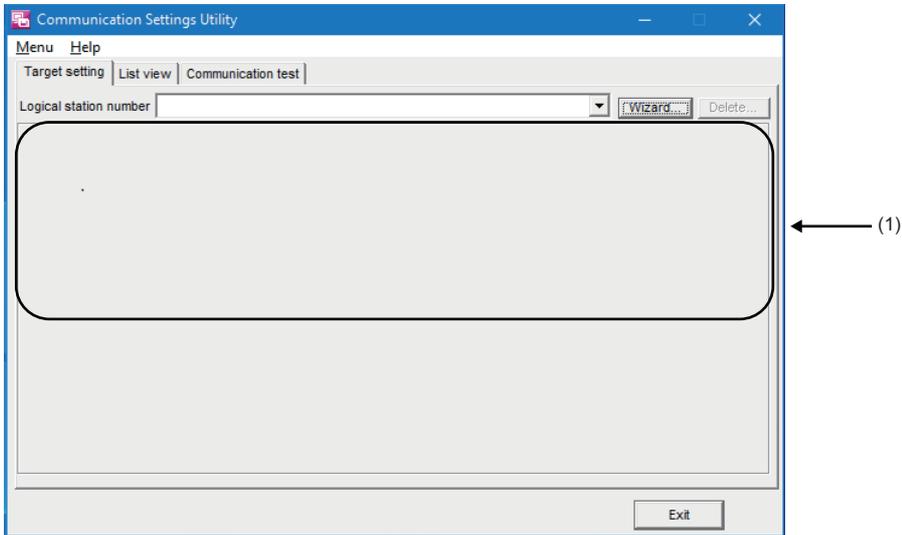
Operating procedure

Start Communication Settings Utility from "MELSOFT" in Windows Start.

[Target setting] tab

This tab is used to display the setting details of the logical station number set in Communication Setting Wizard and to edit it.

Window



Item	Description
Logical station number	Select a logical station number to display and edit the setting details set in Communication Setting Wizard.
[Wizard] button	Click this to start Communication Setting Wizard and set the logical station number.
[Delete] button	Click this to delete the preset logical station number.
(1) Connection image diagram	Displays the connection image diagram of the selected logical station number. By clicking any sketch (personal computer or module) in the connection image diagram, Communication Setting Wizard starts and the settings can be changed.

Point

For details on Communication Setting Wizard, refer to the following:

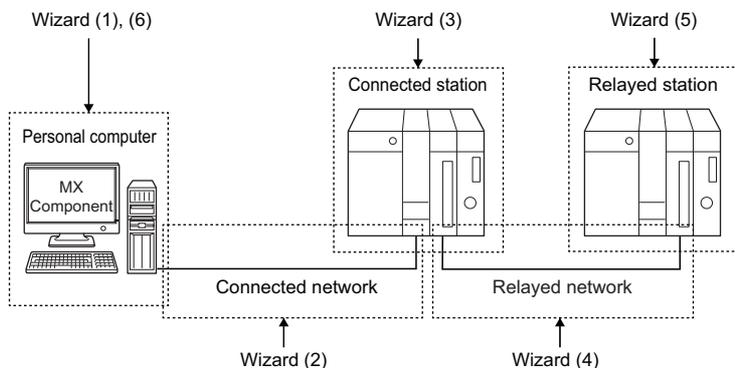
☞ Page 56 Operations on Communication Setting Wizard screens

Operations on Communication Setting Wizard screens

Set a logical station number which is required to perform communication with the utility setting type.

■ Outline of Communication Setting Wizard

The logical station number which is required to perform communication with the utility setting type is set in the wizard format.



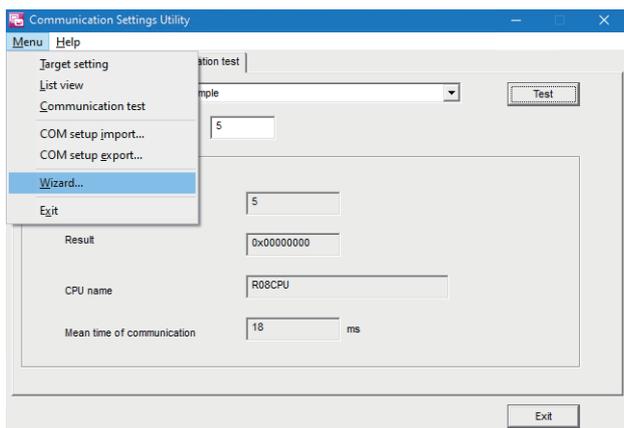
The following are the descriptions of the settings set in Communication Setting Wizard screens.

Communication Setting Wizard	Screen Name	Description
Wizard (1)	Introduction	Set the logical station number.
Wizard (2)	PC side	Set the connected network between the personal computer and connected station (CPU module and module).
Wizard (3)	PLC side	Set the connected station (CPU module and module).
Wizard (4)	Network	Set the relayed network between the connected station (CPU module and module) and relayed station (CPU module and module).
Wizard (5)	Other station	Set the relayed station CPU.
Wizard (6)	Finished	Add a comment to the logical station number.

■ Display method of Communication Setting Wizard screens.

Window

- Select [Menu] ⇒ [Wizard].
- Click the [Wizard] button on the utility screen.



■ Display procedure of Communication Setting Wizard screens.

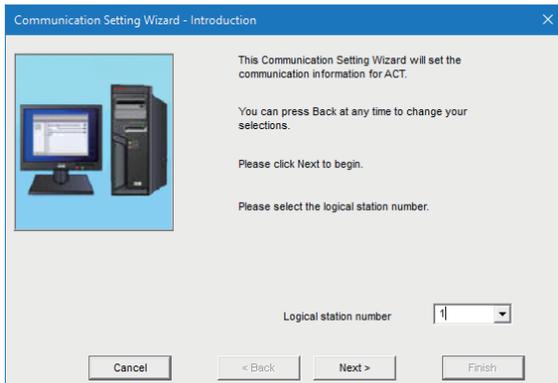
Communication Setting Wizard screens are displayed from wizard (1) to wizard (6) in order.

The following explains Communication Setting Wizard screens in displayed order.

Point

- The displays or available setting items of Communication Setting Wizard screens differ depending on the settings.
Set all the available setting items being displayed.
- Some of Communication Setting Wizard screens may not be displayed depending on the settings.

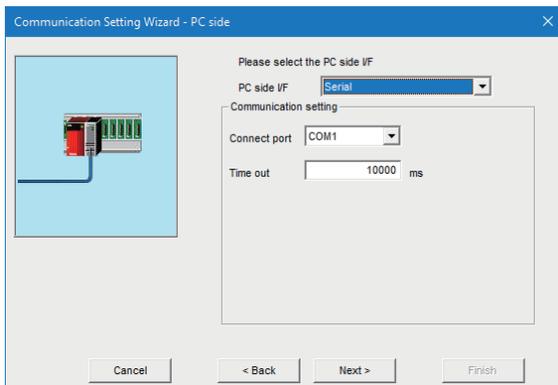
Operating procedure



Wizard (1) - Introduction

1. Enter or select a logical station number.

The logical station number can be registered from 0 to 1023.

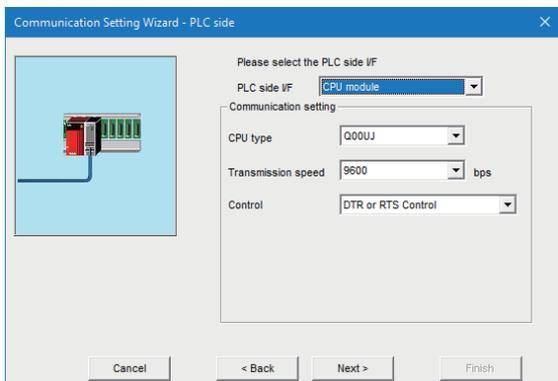


Wizard (2) - PC side

2. Select the personal computer side interface.

Items corresponding to the communications in "PC side I/F"

(Page 59 Items corresponding to the communications in "PC side I/F")

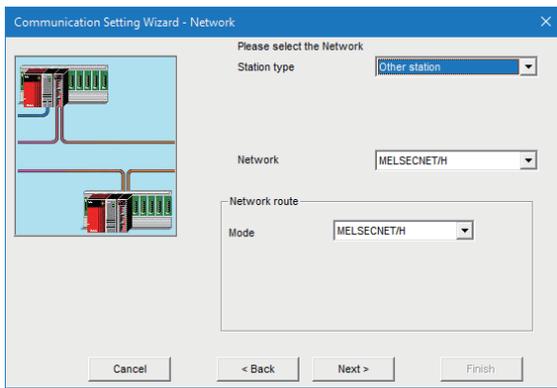


Wizard (3) - PLC side

3. Select the PLC side interface.

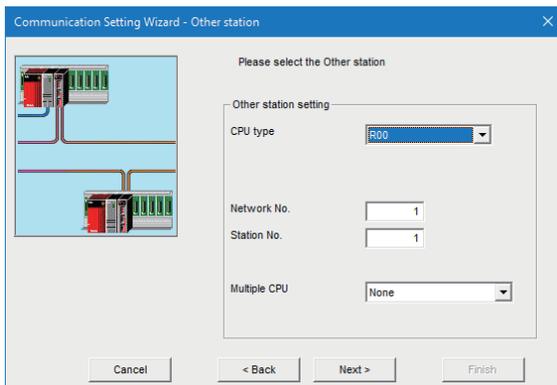
Communication via GOT (Page 58 Communication via GOT)





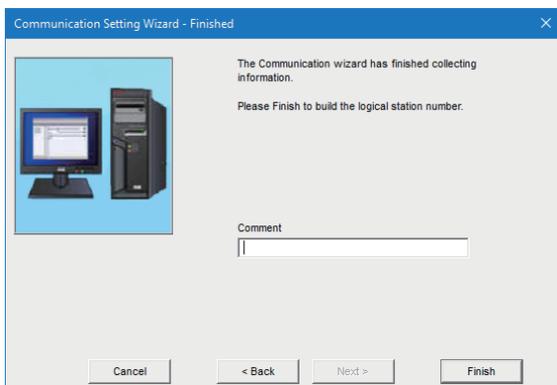
Wizard (4) - Network

4. Select a communication route.



Wizard (5) - Other station

5. Select other station.

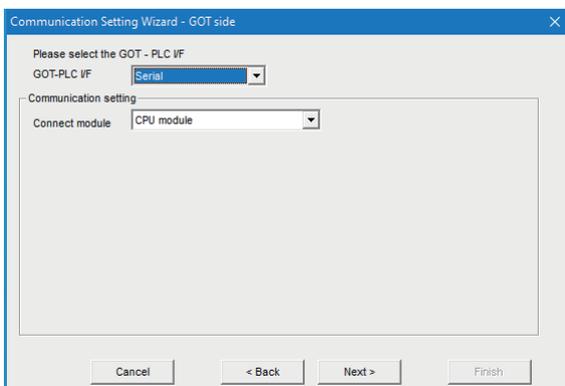


Wizard (6) - Finished

6. Enter a comment and click the [Finish] button.
Up to 32 alphanumeric characters can be entered in 'Comment.'
When the registration of the logical station number is completed in Communication Setting Wizard, the settings are displayed on the [Target setting] tab.

■ Communication via GOT

The following screen appears.



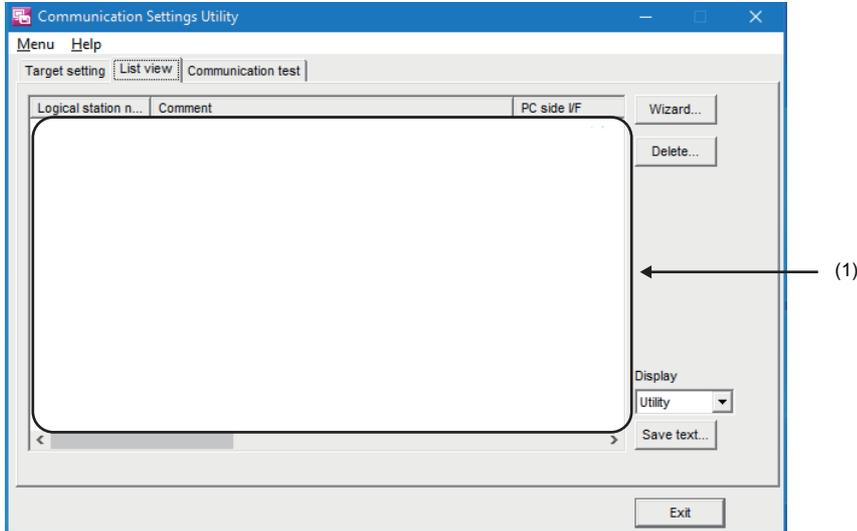
■Items corresponding to the communications in "PC side I/F"

Setting Item	Communication Name
USB	USB communication
USB (via GOT)	GOT transparent communication
Serial	Serial communication, CPU COM communication, CC-Link system RS-232 interface communication
Serial (via GOT)	GOT transparent communication
Ethernet board	Ethernet communication, GOT gateway device communication
Ethernet board (via GOT)	GOT transparent communication
CC IE Control board	CC-Link IE Controller Network communication
MELSECNET/H board	MELSECNET/H communication
CC IE TSN board	CC-Link IE TSN communication
CC IE Field board	CC-Link IE Field Network communication
CC-Link board	CC-Link communication
GX Simulator3	GX Simulator3 communication
GX Simulator2	GX Simulator2 communication
MT Simulator2	MT Simulator2 communication

[List view] tab

This tab is used to list and edit the registered logical station number, and list the properties required for the program setting type.

Window



Item	Description
Property list	Displays the settings of the registered logical station number. By double-clicking the logical station number, Communication Setting Wizard starts.
[Wizard] button	Click this to start Communication Setting Wizard and set the logical station number.
[Delete] button	Click this to delete the preset logical station number.
Display	Utility: Displays the settings set for the logical station number in the property list. Program: Displays the property list required for setting with the program setting type.
[Save text] button	Click this to save the settings of property list to a file in the .txt format.

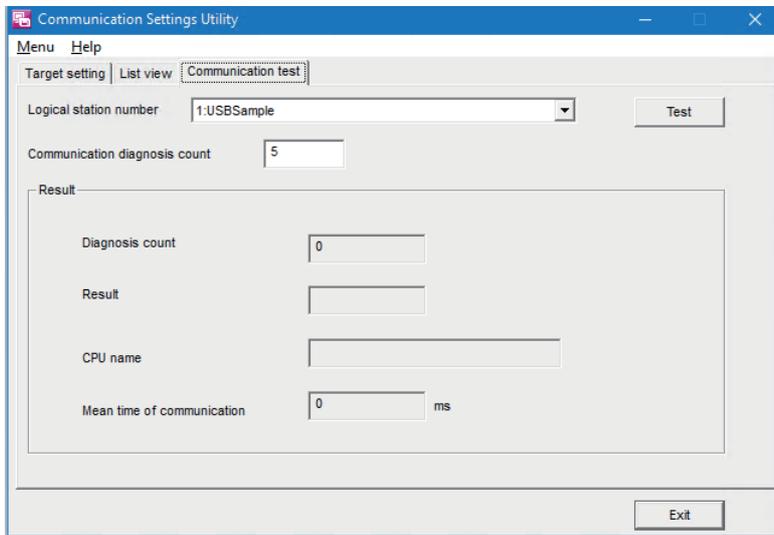
Point

- For details on Communication Setting Wizard, refer to the following:
 Page 56 Operations on Communication Setting Wizard screens
- For a communication route for selecting PC side adapter, the ActHostAddress property on the [List view] tab is displayed as follow:
 'Connection target IP address' + tab character + 'Personal computer side IP address.'
 To set the value of the ActHostAddress property by using the program setting type, use a tab character after replacing it to the character code of the correct tab.

[Communication test] tab

This tab is used to perform the communication test of a registered logical station number.

Window



Item	Description	
Logical station number	Select a logical station number to perform a communication test.	
Communication diagnosis count	Set how many times (1 to 32767) the communication test will be repeated for the specified logical station number.	
[Test] button ([Cancel] button)	Click this to start (stop) the communication test.	
Result	Diagnosis count	Displays the number of connections during the communication test.
	Result	Displays the test result. An error code is displayed at error occurrence. '0' is displayed at normal termination, or any value other than '0' is displayed at abnormal termination.
	CPU name	Displays the connected CPU type.
	Mean time of communication	Displays the average time taken until one communication test is established. (Unit: ms)

Operating procedure

1. Select a logical station number.
2. Click the [Test] button.
 - The error code is displayed in 'Result.' (At normal termination, "0x00000000" is displayed in 'Result.')
 - If an error occurs, check the error code and remove the error. (👉 Page 616 ERROR CODES)

Point

Create a user program and correct device date with PLC Monitor Utility by using a logical station number whose communication is conformed as normal.

Exporting/importing the communication settings

Exporting

Save the communication settings set to a personal computer in a file. (The file where data is saved is called the ACT registered file.)

This file is used to apply the communication settings to another personal computer.

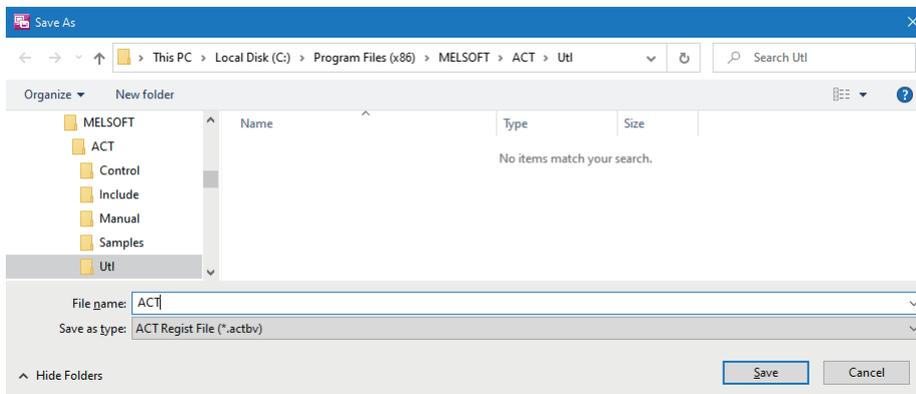
Point

- Uninstalling deletes all the settings within "Communication Settings Utility."
- To avoid this, export the file storing the settings.
- When importing the file using Communication Settings Utility with MX Component Version 4.11M or earlier, a file of which '*.actv' is selected when saving the file can be used.

Operating procedure

Select [Menu] ⇒ [COM setup export].

Window



Importing

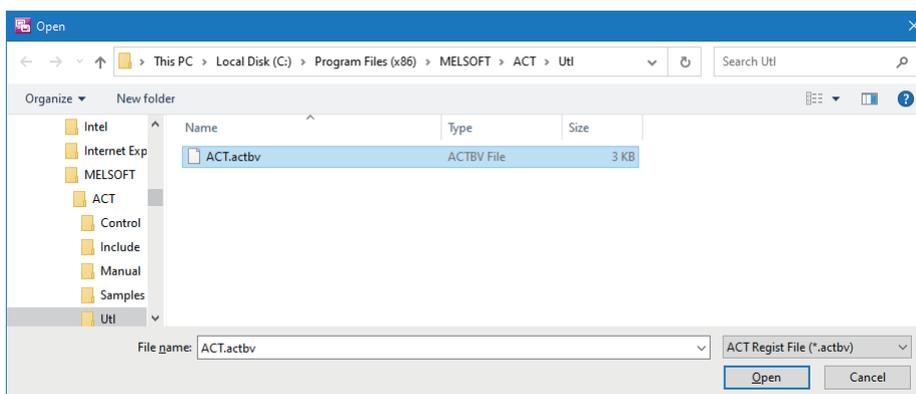
Apply the exported communication settings saved to the file (Page 62 Exporting) to the utility.

This screen is used when the communication settings set on another personal computer are to be applied to the personal computer being used.

Operating procedure

Select [Menu] ⇒ [COM setup import].

Window



Point 

To import the communication settings, use MX Component of a version of the one used for export, or later. Using a MX Component version earlier than that may cause incorrect import.

Specifying a created logical number in a program

Communication can be established by specifying the logical station number set with Communication Settings Utility to the ActLogicalStationNumber(LONG) which is the open function property, and executing the open function.

5.2 Setting in Programs

To perform communication using the program setting type, the properties of the corresponding Act controls is required to be set.

For the properties of the corresponding Act controls, directly enter them on the property window or change their settings in the user program.

For details on the properties which must be set for the corresponding Act controls, refer to the following:

☞ Page 172 Details of Control Properties

For module settings when using MX Component, refer to the following:

Item	Reference
Serial communication	Page 24 SYSTEM CONFIGURATIONS
Ethernet communication (when using Ethernet modules)	
CC-Link system RS-232 interface communication	

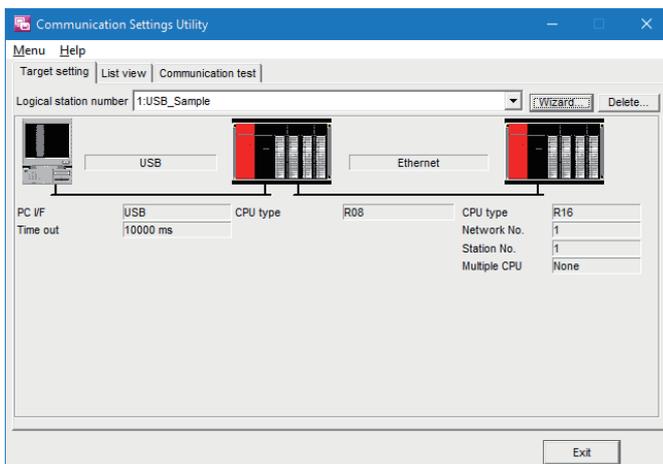
Operation procedure

1. Check the required property by using Communication Settings Utility.
2. Set the required property in a user program.

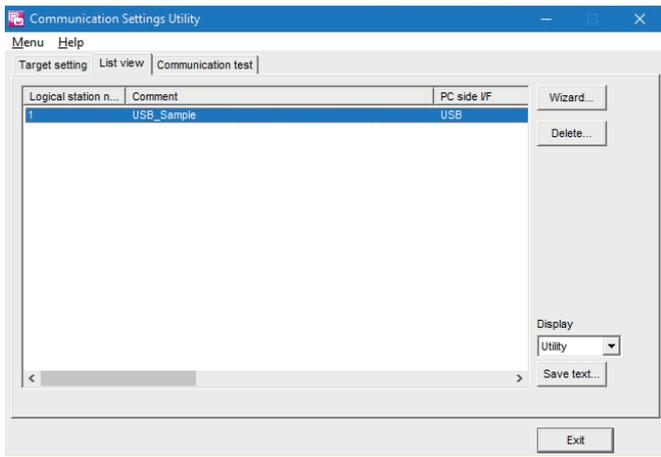
Checking a property with Communication Settings Utility

Check a property by the following procedure.

Operating procedure



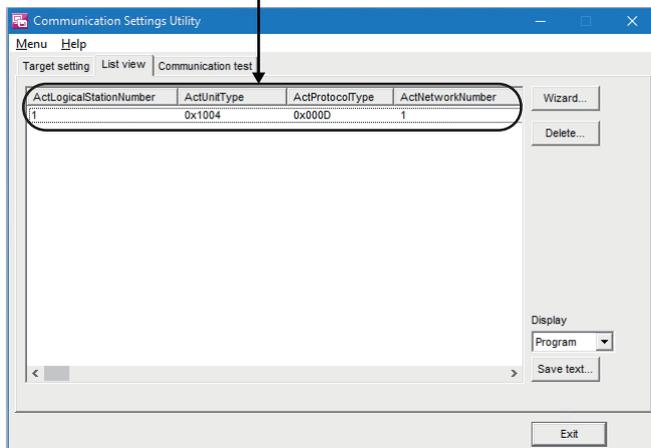
1. Specify a communication route in "Communication Setting Wizard" on Communication Settings Utility. For details on Communication Setting Wizard, refer to the following:
☞ Page 56 Operations on Communication Setting Wizard screens



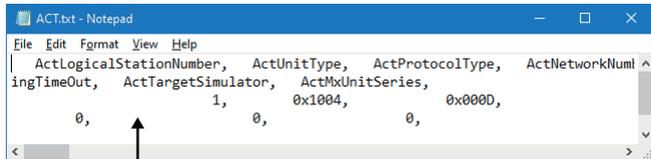
2. Click the [List view] tab.
For details on the [List view] tab, refer to the following:
📖 Page 60 [List view] tab
3. Select "Program" from "Display".



Control names to be used and properties to be set are displayed.



4. Control the scroll bar to confirm the property.
The file can be saved in the .txt format by clicking the [Save text] button.



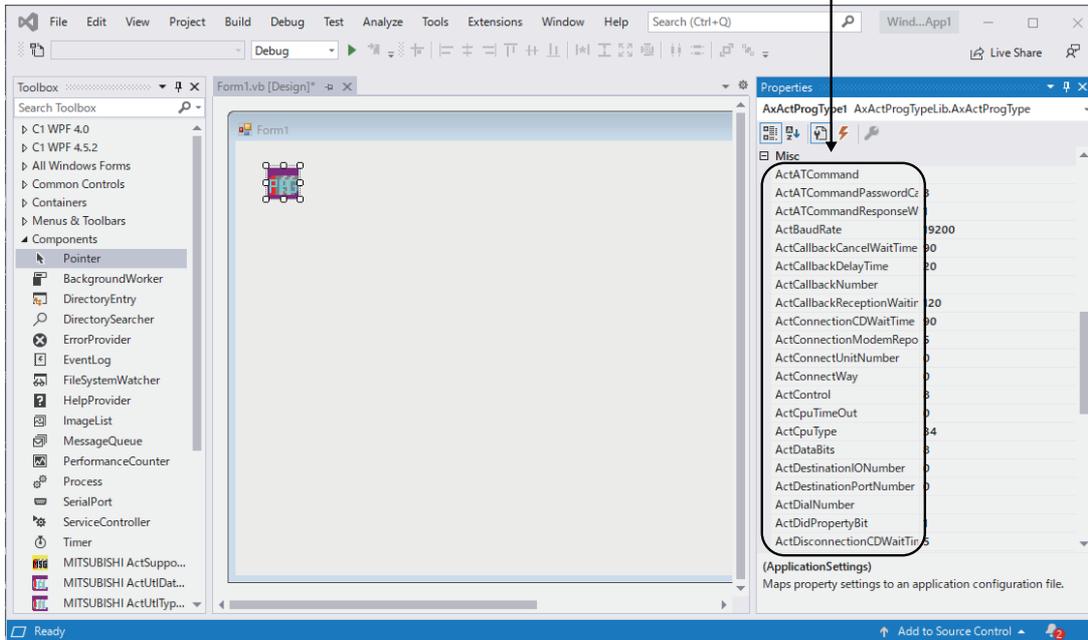
Saved into file in .txt format.

Setting a required property in a user program

Check the property values with the text file output from Communication Settings Utility, and enter the values into Properties of the property window.

Example) Visual Basic .NET

	A	B	C	D	E	F	G	H	I
1	ActLogicalStationNumber	ActUnitType	ActProtocolType	ActNetworkNumber	ActStationNumber	ActUnitNumber	ActConnectUnitNumber	ActIONumber	ActCPUType
2	1	0x1004	0x000D	1	1	0	0	0x03E0	0x1003
3	2	0x0013	0x0013	0	255	0	0	0x03FF	0x0072
4	3	0x0013	0x0004	0	255	0	0	0x03E0	0x00D3



The settings of the property can be changed with the source code.

6 COMMUNICATION SETTING EXAMPLES (UTILITY SETTING TYPE)

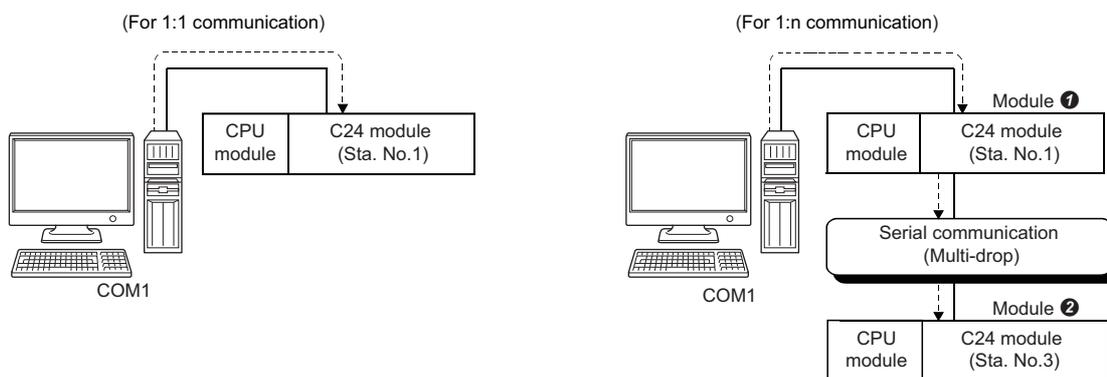
This chapter explains the setting procedure and setting example of each communication route when the utility setting type is used for programming.

Point

- The communication settings can be registered or changed easily with the utility setting type in MX Component.
- For the settings other than "As set by user," set the value shown in the table.
- When checking the communication settings in MX Component for the first time, make sure that communication of each communication route can be established properly in each engineering tool in advance so that the cause of the communication failure can be identified easily.
- For details on the settings of modules, refer to the manual of each module.

6.1 Serial Communication

This section provides the serial communication procedure and its setting example using the utility setting type.



Access procedure

The following shows the procedure before accessing a CPU module.

Operating procedure

- 1. Set a module.**
☞ Page 70 Settings of serial communication modules
- 2. Connect the module and a personal computer.**
☞ Page 24 SYSTEM CONFIGURATIONS
- 3. Set a logical station number with Communication Settings Utility.**
☞ Page 69 Logical station number setting example
- 4. Perform a communication test to check the communication.**
☞ Page 61 [Communication test] tab

Logical station number setting example

For the operating procedure of Communication Setting Wizard screens, refer to the following:

☞ Page 56 Operations on Communication Setting Wizard screens

Setting data

■For 1:n communication

Communication Setting Wizard	Item	Setting example
Introduction	Logical station number	0 to 1023
PC side	PC side I/F	Serial
	Connect port	COM1
	Time out	10000
PLC side	PLC side I/F	C24 module
	Module type	Select a C24 module to be used.
	Station No.	1
	Transmission speed	19200
	Parity	Odd
	Control	DTR or RTS Control
Network	Station type	Other station
	Network	C24
	Network type	Multidrop(combine)
Other station	CPU type	Select a CPU module to be used.
	Station No.	3
Finished	Comment	(Optional)

Settings of serial communication modules

The following explains the settings of serial communication modules for the use of MX Component.
The settings of MX Component need to be the same as the module.

MELSEC iQ-R series-compatible C24

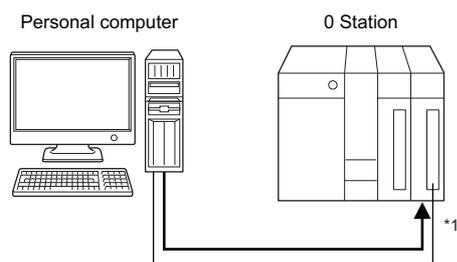
For details, refer to the following:

📖 MELSEC iQ-R Serial Communication Module User's Manual(Application)

Q series-compatible C24, L series-compatible C24

■For 1:1 communication

The following configuration is used for a setting example.



*1 Use of CH1



When connecting a personal computer with Q/L series compatible C24 on a 1:1 basis, the communication can be performed without the switch setting of Q/L series compatible C24. When the switch setting is not set, the module operates as GX Developer/MELSOFT connection as shown below.

Item	Setting		Setting value
	b15 to b8	b7 to b0	
Switch 1	CH1 communication speed	CH1 transmission setting	0000H
Switch 2	—	CH1 communications protocol	0000H
Switch 3	CH2 communication speed	CH2 transmission setting	0000H
Switch 4	—	CH2 communications protocol	0000H
Switch 5	Module station number		0000H

When the communication protocol is set to '00H' (GX Developer/MELSOFT connection), the module operates with the value shown in the following table.

For details, refer to the following:

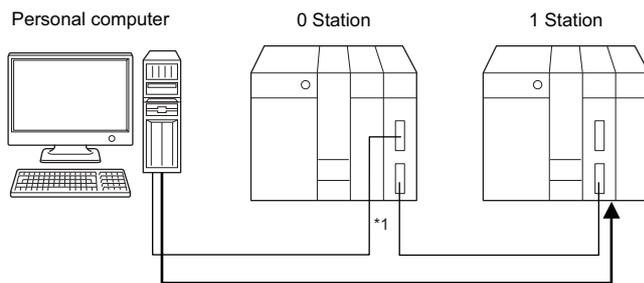
📖 Q Corresponding Serial Communication Module User's Manual (Basic)

📖 MELSEC-L Serial Communication Module User's Manual (Basic)

Transmission setting	Setting
Data bit	8
Parity bit	Yes
Odd/even number parity	Odd number
Stop bit	1
Sum check code	Yes
Online change	Enable

■ For 1:n communication

The following configuration is used for a setting example.



*1 Use of both CH1 and CH2

① 0 Station

Item	Setting		Setting value
	b15 to b8	b7 to b0	Synchronous operation
Switch 1	CH1 communication speed	CH1 transmission setting	07E6H
Switch 2	—	CH1 communications protocol	0008H
Switch 3	CH2 communication speed	CH2 transmission setting	07E7H
Switch 4	—	CH2 communications protocol	0000H
Switch 5	Module station number		0000H (As set by user)

② 1 Station

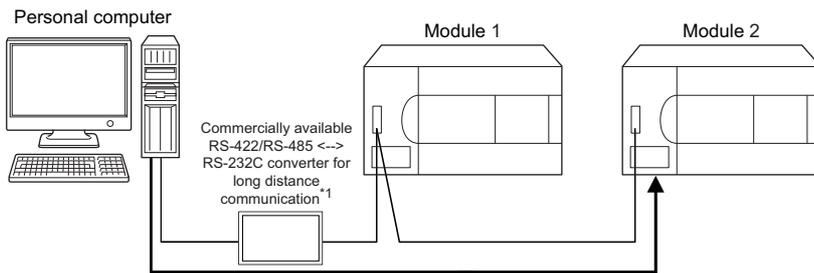
Item	Setting		Setting value
	b15 to b8	b7 to b0	Independent operation
Switch 1	CH1 communication speed	CH1 transmission setting	Set to satisfy the application of CH1.
Switch 2	—	CH1 communications protocol	
Switch 3	CH2 communication speed	CH2 transmission setting	07E6H
Switch 4	—	CH2 communications protocol	0005H
Switch 5	Module station number		0000H (As set by user)

The module operates with the value shown in the following table.

Transmission setting	Setting	
Operation setting	0 Station	Synchronous
	1 Station	Independent
Data bit	8	
Parity bit	Yes	
Odd/even number parity	Odd number	
Stop bit	1	
Sum check code	Yes	
Online change	Enable	
Transmission speed setting	19200 bps	

FX extended port

The following configuration is used for a setting example.



*1 For details, refer to the technical bulletin (HIME-T-P-0202).

Set the FXCPU parameters with GX Works2 before using FX extended port to perform communication.

Execute either of the following two setting methods by following the instruction described on the reference page.

- To use the PLC parameters (Page 72 Settings with PLC parameters)
- To write values to special data registers (D8120, D8121, D8129) in a sequence program (Page 73 Settings by writing values to the special data registers in a sequence program)

■Settings with PLC parameters

Operating procedure

1. Start GX Works2.
2. Select [Parameter] ⇒ [PLC parameter] in the project list.
3. Click the [PLC System(2)] tab in the "FX Parameter" screen.

The screenshot shows the "FX Parameter" dialog box with the "PLC System(2)" tab selected. The "Operate Communication Setting" checkbox is checked. The settings are as follows:

- CH1: CH1
- Protocol: Non-Procedural
- Data Length: 7Bit
- Parity: Odd
- Stop Bit: 1Bit
- Transmission Speed: 9600 (bps)
- Header:
- Terminator:
- Control Line:
- H/W Type: Regular/RS-232C
- Control Mode: Invalid
- Sum Check:
- Transmission Control Procedure: Form1(Without CR,LF)
- Station Number Setting: 00 H (00H-0FH)
- Time Out Judge Time: 1 X 10ms (1--255)

Buttons at the bottom: Print Window..., Print Window Preview, Default, Check, End, Cancel.

4. Set the items on the screen.

Item	Description
Operate Communication Setting	Select the checkbox.
Protocol* ¹	Select the communication protocol.
Data Length* ¹	7bit/8bit
Parity* ¹	NONE/Odd/Even
Stop Bit* ¹	1bit/2bit
Transmission Speed* ¹	300 bps/600 bps/1200 bps/2400 bps/4800 bps/9600 bps/19200 bps
H/W Type* ¹	Select the cable type used in the communication.
Sum Check* ¹	Select/clear
Transmission Control Procedure* ¹	Select the transmission control procedure.
Station Number Setting* ¹	Set the station number.
Time Out Judge Time* ¹	Set the time out period.

*1 Not applicable when "Operate Communication Setting" is OFF.

Point

- When communication setting is set, turn ON the FXCPU again after writing to programmable controller.
- When performing multi-drop connection, set the same communication settings for the devices. However, check that the station number is not overlapped.

■ Settings by writing values to the special data registers in a sequence program

① D8120 (communication format)

	b15	b14	b13	b12	b11	b10	b9	b8	b7	b6	b5	b4	b3	b2	b1	b0
D8120																

Bit	Description	Setting
b0	Data length	0: 7 bits 1: 8 bits
b1	Parity	None: b2=0, b1=0
b2		Odd number: b2=0, b1=1
		Even number: b2=1, b1=1
b3	Stop bit	0: 1 bit 1: 2 bits
b4	Transmission speed	Set the transmission speed.* ¹
b5		
b6		
b7		
b8	—	0
b9	—	0
b10	H/W type	RS-485: b11=0, b10=0
b11		
b12	—	0
b13	Sum check	0: Not available 1: Available
b14	Communication protocol	1: Serial communication
b15	Transmission control procedure	0: Form 1

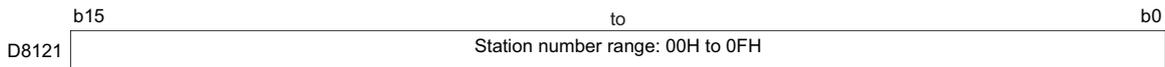
*1 The setting contents of transmission speed are shown as below:

Transmission speed	b7	b6	b5	b4
300 bps	0	0	1	1
600 bps	0	1	0	0
1200 bps	0	1	0	1
2400 bps	0	1	1	0
4800 bps	0	1	1	1
9600 bps	1	0	0	0
19200 bps	1	0	0	1

② D8121 (station number setting)

Specify the station number.

The station number can be specified in the range of 00H to 0FH.



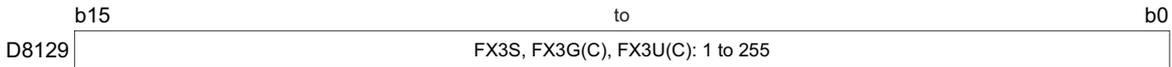
③ D8129 (time out judge time setting)

Specify the FXCPU time out judge time in 10 ms units.

The setting range is as follows:

For FX3S, FX3G, FX3GC, FX3U, FX3UC, 1 to 255 (10 to 2550 ms).

If "0" is stored, 100 ms is set.

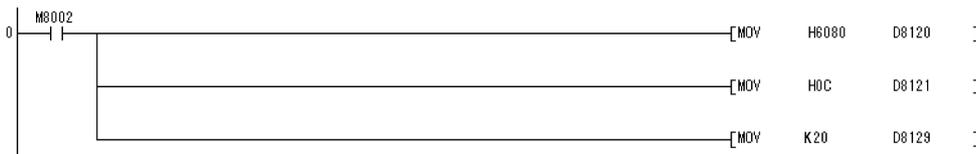


Point

- When communication setting is set, turn ON the FXCPU again after writing to programmable controller.
- When performing multi-drop connection, set the same communication settings for the devices.
However, check that the station number is not overlapped.

Point

The following shows an example of setting values to the special data registers.

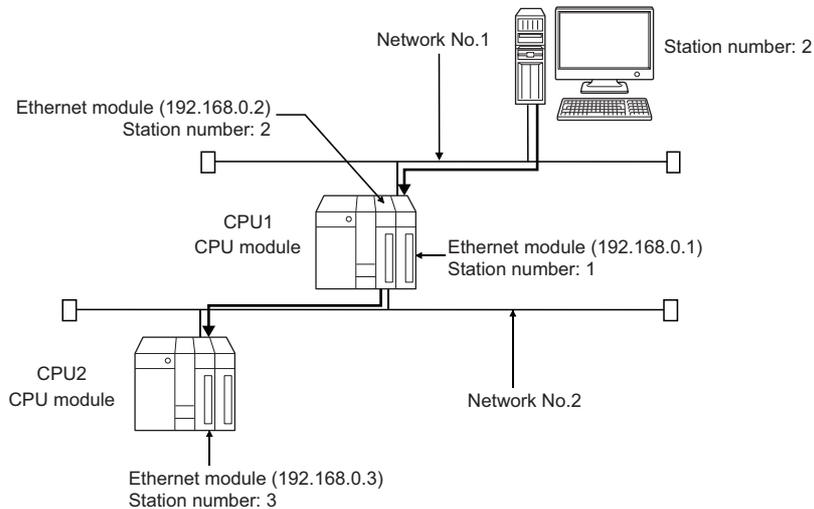


The following shows the setting details of each special data register in the above program.

- D8120
Transmission control procedure: Form 1
Communication protocol: Serial communication
Sum check: Available
H/W type: RS-485
Transmission speed: 9600 bps
Stop bit: 1 bit
Parity: NONE
Data length: 7 bit
- D8121
Station number : 12
- D8129
Time out time : 200 ms

6.2 Ethernet Communication (When Using Ethernet Modules)

This section provides the procedure for Ethernet communication with an Ethernet module and its setting example using the utility setting type.



Access procedure

The following shows the procedure before accessing a CPU module.

Operating procedure

1. Set the module parameters.

☞ Page 77 Settings of modules

2. Connect the module and a personal computer.

☞ Page 24 SYSTEM CONFIGURATIONS

3. Edit a HOSTS file.

The HOSTS file is not required to be edited when entering an IP address in a host name (IP Address) of Communication Settings Utility and the ActHostAddress property.

4. Set a logical station number with Communication Settings Utility.

☞ Page 76 Logical station number setting example

5. Perform a communication test to check the communication.

☞ Page 61 [Communication test] tab

Logical station number setting example

For the operating procedure of Communication Setting Wizard screens, refer to the following:

☞ Page 56 Operations on Communication Setting Wizard screens

Setting data

Communication Setting Wizard	Item	Setting example
Introduction	Logical station number	0 to 1023
PC side	PC side I/F	Ethernet board
	Connect module	Select an Ethernet board to be used.
	Protocol	TCP
	Network No.	1
	Station No.	2
PLC side	Time out	60000
	PLC side I/F	Ethernet module
	Module type	Select an Ethernet module to be used.
	Host (IP Address)	192.168.0.1
Network	Station No.	1
	Station type	Other station
Other station	Network	Ethernet
	CPU type	Select a CPU module to be used.
	Network No.	2
	Station No.	3
Multiple CPU	Multiple CPU	None
	Finished	Comment

Settings of modules

The following shows the settings of a module when using MX Component.
The settings of MX Component need to be the same as the module.

Settings of parameters

Set the network type, start I/O number, network number, station number, mode and operational settings on the Network Parameter - MELSECNET/CC IE/Ethernet Module Configuration screen of GX Works2.

CPU to be set	Setting screen example																																																	
CPU 1	<p>Ethernet parameters</p> <table border="1"> <thead> <tr> <th></th> <th>Module 1</th> <th>Module 2</th> </tr> </thead> <tbody> <tr><td>Network Type</td><td>Ethernet</td><td>Ethernet</td></tr> <tr><td>Start I/O No.</td><td>0000</td><td>0020</td></tr> <tr><td>Network No.</td><td>2</td><td>1</td></tr> <tr><td>Total Stations</td><td></td><td></td></tr> <tr><td>Group No.</td><td>0</td><td>0</td></tr> <tr><td>Station No.</td><td>2</td><td>1</td></tr> <tr><td>Mode</td><td>Online</td><td>Online</td></tr> <tr><td colspan="3" style="text-align: center;">Operation Setting</td></tr> <tr><td colspan="3" style="text-align: center;">Initial Setting</td></tr> <tr><td colspan="3" style="text-align: center;">Open Setting</td></tr> <tr><td colspan="3" style="text-align: center;">Router Relay Parameter</td></tr> <tr><td colspan="3" style="text-align: center;">Station No. <->IP Information</td></tr> <tr><td colspan="3" style="text-align: center;">FTP Parameters</td></tr> <tr><td colspan="3" style="text-align: center;">E-mail Setting</td></tr> <tr><td colspan="3" style="text-align: center;">Interrupt Settings</td></tr> </tbody> </table>		Module 1	Module 2	Network Type	Ethernet	Ethernet	Start I/O No.	0000	0020	Network No.	2	1	Total Stations			Group No.	0	0	Station No.	2	1	Mode	Online	Online	Operation Setting			Initial Setting			Open Setting			Router Relay Parameter			Station No. <->IP Information			FTP Parameters			E-mail Setting			Interrupt Settings			<p>Operation setting</p>
	Module 1	Module 2																																																
Network Type	Ethernet	Ethernet																																																
Start I/O No.	0000	0020																																																
Network No.	2	1																																																
Total Stations																																																		
Group No.	0	0																																																
Station No.	2	1																																																
Mode	Online	Online																																																
Operation Setting																																																		
Initial Setting																																																		
Open Setting																																																		
Router Relay Parameter																																																		
Station No. <->IP Information																																																		
FTP Parameters																																																		
E-mail Setting																																																		
Interrupt Settings																																																		
CPU 2	<p>Ethernet parameters</p> <table border="1"> <thead> <tr> <th></th> <th>Module 1</th> <th>Module 2</th> </tr> </thead> <tbody> <tr><td>Network Type</td><td>Ethernet</td><td>None</td></tr> <tr><td>Start I/O No.</td><td>0000</td><td></td></tr> <tr><td>Network No.</td><td>2</td><td></td></tr> <tr><td>Total Stations</td><td></td><td></td></tr> <tr><td>Group No.</td><td>0</td><td></td></tr> <tr><td>Station No.</td><td>3</td><td></td></tr> <tr><td>Mode</td><td>Online</td><td></td></tr> <tr><td colspan="3" style="text-align: center;">Operation Setting</td></tr> <tr><td colspan="3" style="text-align: center;">Initial Setting</td></tr> <tr><td colspan="3" style="text-align: center;">Open Setting</td></tr> <tr><td colspan="3" style="text-align: center;">Router Relay Parameter</td></tr> <tr><td colspan="3" style="text-align: center;">Station No. <->IP Information</td></tr> <tr><td colspan="3" style="text-align: center;">FTP Parameters</td></tr> <tr><td colspan="3" style="text-align: center;">E-mail Setting</td></tr> <tr><td colspan="3" style="text-align: center;">Interrupt Settings</td></tr> </tbody> </table>		Module 1	Module 2	Network Type	Ethernet	None	Start I/O No.	0000		Network No.	2		Total Stations			Group No.	0		Station No.	3		Mode	Online		Operation Setting			Initial Setting			Open Setting			Router Relay Parameter			Station No. <->IP Information			FTP Parameters			E-mail Setting			Interrupt Settings			<p>Operation setting</p>
	Module 1	Module 2																																																
Network Type	Ethernet	None																																																
Start I/O No.	0000																																																	
Network No.	2																																																	
Total Stations																																																		
Group No.	0																																																	
Station No.	3																																																	
Mode	Online																																																	
Operation Setting																																																		
Initial Setting																																																		
Open Setting																																																		
Router Relay Parameter																																																		
Station No. <->IP Information																																																		
FTP Parameters																																																		
E-mail Setting																																																		
Interrupt Settings																																																		

6



Set the parameters in GX Works3 for an RCP. For details on parameter settings, refer to the following:
[MELSEC iQ-R Ethernet User's Manual \(Application\)](#)

Settings of routing parameters

Set the following items on the Network Parameter - Setting the Ethernet/CC IE/MELSECNET Routing Information screen of GX Works2.

For the routing parameters, refer to the following:

 Page 647 Concept of Routing Parameters

CPU to be set	Setting screen example																		
CPU 1	<table border="1"><thead><tr><th></th><th>Target Network No.</th><th>Relay Network No.</th><th>Relay Station No.</th></tr></thead><tbody><tr><td>1</td><td>1</td><td>2</td><td>2</td></tr><tr><td>2</td><td>2</td><td>1</td><td>1</td></tr><tr><td>3</td><td></td><td></td><td></td></tr></tbody></table>		Target Network No.	Relay Network No.	Relay Station No.	1	1	2	2	2	2	1	1	3					
	Target Network No.	Relay Network No.	Relay Station No.																
1	1	2	2																
2	2	1	1																
3																			
CPU 2	<table border="1"><thead><tr><th></th><th>Target Network No.</th><th>Relay Network No.</th><th>Relay Station No.</th></tr></thead><tbody><tr><td>1</td><td>1</td><td>2</td><td>2</td></tr><tr><td>2</td><td></td><td></td><td></td></tr><tr><td>3</td><td></td><td></td><td></td></tr></tbody></table>		Target Network No.	Relay Network No.	Relay Station No.	1	1	2	2	2				3					
	Target Network No.	Relay Network No.	Relay Station No.																
1	1	2	2																
2																			
3																			

Point

Set routing settings in GX Works3 for an RCPU.

For details on the routing setting, refer to the following:

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

Checking the communication

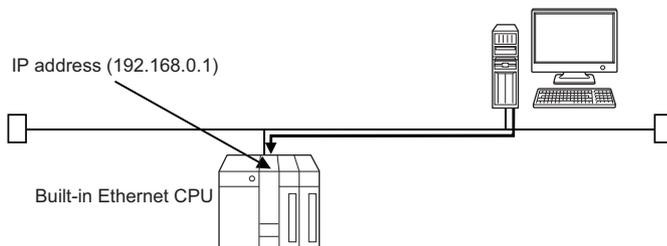
After completion of preparations for Ethernet communication, execute ping in the MS-DOS mode to check connection before starting communications on MX Component.

- When normal
C:\>ping 192.168.0.2
Reply from 192.168.0.2:bytes=32 time<10ms TTL=32
- When abnormal
C:\>ping 192.168.0.2
Request timed out.

If ping does not pass through, check a cable and module connections and Windows side IP address and other settings.

6.3 Ethernet Communication (When Using Built-in Ethernet CPUs)

This section provides the procedure for the Ethernet communication with the built-in Ethernet CPU and its setting example using the utility setting type.



Access procedure

The following shows the procedure before accessing a CPU module.

6

Operating procedure

■When directly connecting to an Ethernet port

1. Connect a module and personal computer.
☞ Page 24 SYSTEM CONFIGURATIONS
2. Set a logical station number with Communication Settings Utility.
☞ Page 80 When directly connecting to an Ethernet port
3. Perform a communication test to check the communication.
☞ Page 61 [Communication test] tab

■When specifying a host name (IP address)

1. Connect a module and personal computer.
☞ Page 24 SYSTEM CONFIGURATIONS
2. Set the module parameters.
☞ Page 81 Setting of parameters (Only when specifying a host name (IP address)), ☞ Page 81 Checking the communication

3. Edit a HOSTS file.

The HOSTS file is not required to be edited when entering an IP address in a host name (IP Address) of Communication Settings Utility and the ActHostAddress property.

4. Set a logical station number with Communication Settings Utility.
☞ Page 80 When specifying a host name (IP address)
5. Perform a communication test to check the communication.
☞ Page 61 [Communication test] tab

Logical station number setting example

For the operating procedure of Communication Setting Wizard screens, refer to the following:

☞ Page 56 Operations on Communication Setting Wizard screens

Setting data

■When directly connecting to an Ethernet port

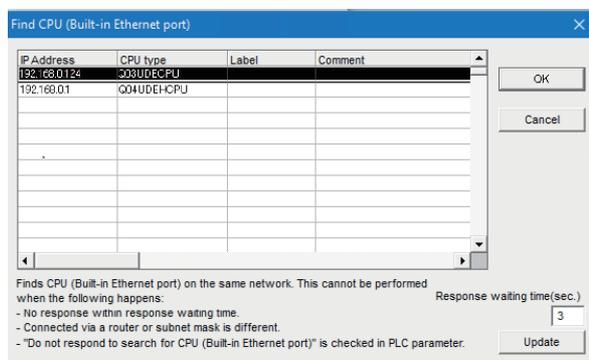
Communication Setting Wizard	Item	Setting example
Introduction	Logical station number	0 to 1023
PC side	PC side I/F	Ethernet board
	Connect module	CPU module
	Protocol	UDP
	Time out	60000
PLC side	PLC side I/F	CPU module
	Ethernet port direct connection	Select the checkbox
Network	Station type	Host station
	CPU type	Select a CPU module to be used.
	Multiple CPU	None
Finished	Comment	(Optional)

■When specifying a host name (IP address)

Communication Setting Wizard	Item	Setting example
Introduction	Logical station number	0 to 1023
PC side	PC side I/F	Ethernet board
	Connect module	CPU module
	Protocol	TCP
	Time out	60000
PLC side	PLC side I/F	CPU module
	Host (IP Address)	192.168.0.1
Network	Station type	Host station
	CPU type	Select a CPU module to be used.
	Multiple CPU	None
Finished	Comment	(Optional)



When using the built-in Ethernet CPU search, click the [Find CPU (Built-in Ethernet port) on network] button. Built-in Ethernet CPUs on the same network is the target of the search.



Choose the target CPU from the list and clicking the [OK] button.

An IP address can be applied to the "Communication Setting Wizard - PLC side" setting.

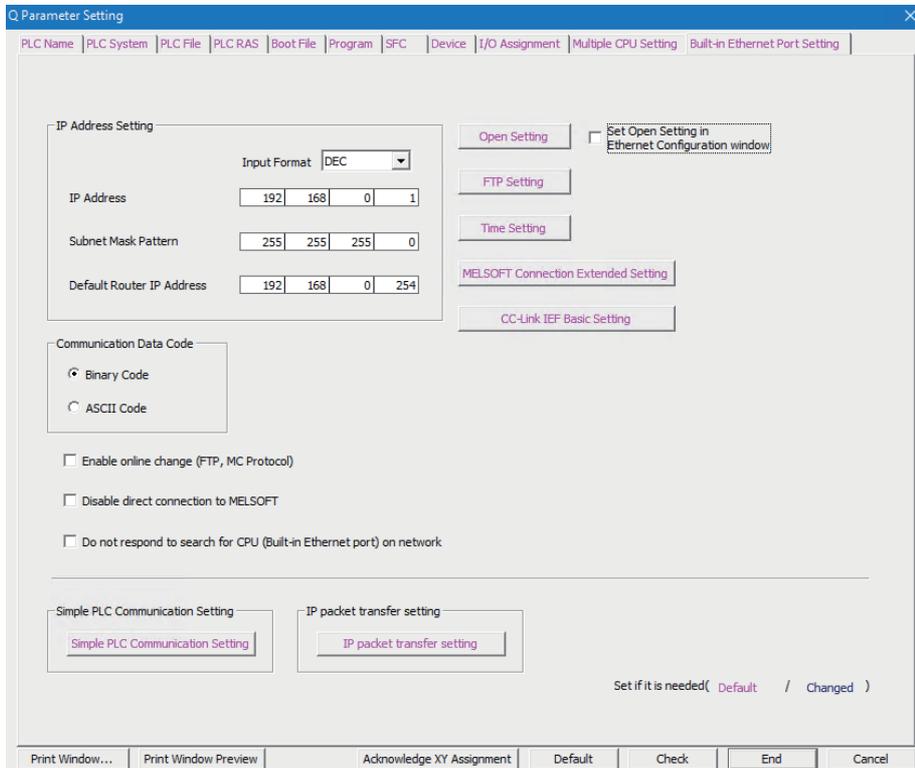
Settings of modules

The following shows the settings of a module when using MX Component.
The settings of MX Component need to be the same as the module.

Setting of parameters (Only when specifying a host name (IP address))

Parameter settings are set from the PLC parameter of GX Works2.

Select the [Built-in Ethernet Port Setting] tab and set the communication settings : IP address or the like.



Point

For RCPUs and LHCPUs, set the parameters in GX Works3.
For details on parameter settings, refer to the following:
[MELSEC iQ-R Ethernet User's Manual \(Application\)](#)

Checking the communication

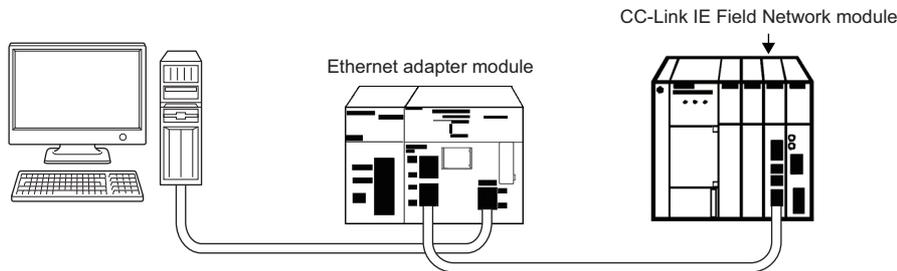
After completion of preparations for Ethernet communication, execute ping in the MS-DOS mode to check connection before starting communications on MX Component.

- When normal
C:\>ping 192.168.0.1
Reply from 192.168.0.1: bytes=32 time=1ms TTL=64
- When abnormal
C:\>ping 192.168.0.1
Request timed out.

If ping does not pass through, check a cable and module connections and Windows side IP address and other settings.

6.4 Ethernet Communication (When Using Ethernet Adapter Modules)

This section explains the procedure for the Ethernet communication using Ethernet adapter module and its setting example using the utility setting type.



Access procedure

The following shows the procedure before accessing a CPU module.

Operating procedure

1. Connect a module and personal computer.
 ☞ Page 24 SYSTEM CONFIGURATIONS
2. Connect an Ethernet adapter module to CC-Link IE Field Network.
3. Set a logical station number with Communication Settings Utility.
 ☞ Page 85 Logical station number setting example
4. Perform a communication test to check the communication.
 ☞ Page 61 [Communication test] tab

Logical station number setting example

For the operating procedure of Communication Setting Wizard screens, refer to the following:

☞ Page 56 Operations on Communication Setting Wizard screens

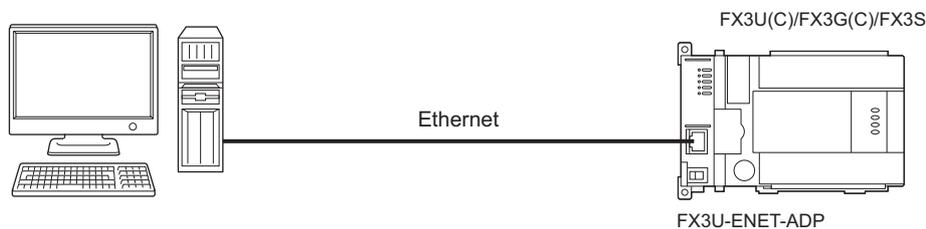
Setting data

Communication Setting Wizard	Item	Setting example
Introduction	Logical station number	0 to 1023
PC side	PC side I/F	Ethernet board
	Connect module	CC IE Field Ethernet adapter
	Protocol	TCP
	Time out	60000
PLC side	PLC side I/F	CC IE Field Ethernet adapter
	Host (IP Address)	192.168.3.30
Network	Station type	Other station (Single)
Other station	CPU type	Select a CPU module to be used.
	Network No.	1
	Station No.	0
	Multiple CPU	None
Finished	Comment	(Optional)

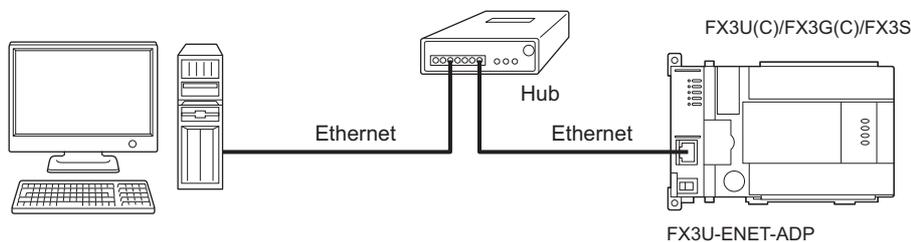
6.5 Ethernet Communication (When Using Ethernet Adapter)

This section provides the procedure for the Ethernet communication with in case of using Ethernet adapter and its setting example using the utility setting type.

<Direct connection>



<IP address specified communication>



6

Access procedure

The following shows the procedure before accessing a CPU module.

Operating procedure

■When directly connecting to an Ethernet port

1. Connect an Ethernet adapter and a personal computer.
☞ Page 24 SYSTEM CONFIGURATIONS
2. Set a logical station number with Communication Settings Utility.
☞ Page 84 Direct connection
3. Perform a communication test to check the communication.
☞ Page 61 [Communication test] tab

■When specifying a host name (IP address)

1. Connect an Ethernet adapter and a personal computer.
☞ Page 24 SYSTEM CONFIGURATIONS
2. Edit a HOSTS file.
The HOSTS file is not required to be edited when entering an IP address in a host name (IP Address) of Communication Settings Utility and the ActHostAddress property.
3. Set a logical station number with Communication Settings Utility.
☞ Page 84 IP address specified communication
4. Perform a communication test to check the communication.
☞ Page 61 [Communication test] tab

Logical station number setting example

For the operating procedure of Communication Setting Wizard screens, refer to the following:

☞ Page 56 Operations on Communication Setting Wizard screens

Setting data

■ Direct connection

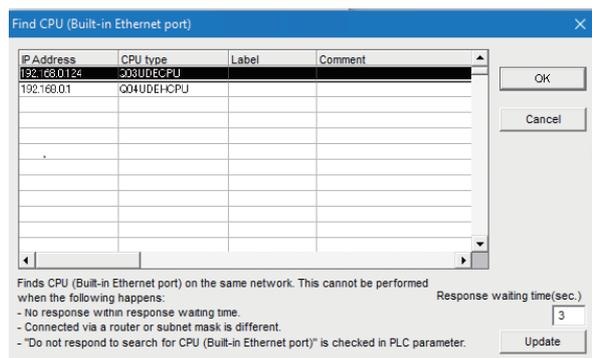
Communication Setting Wizard	Item	Setting example (direct connection)
Introduction	Logical station number	0 to 1023
PC side	PC side I/F	Ethernet board
	Connect module	FX3U-ENET-ADP
	Time out	60000
PLC side	PLC side I/F	FX3U-ENET-ADP
	Ethernet port direct connection	Select the checkbox
Network	Station type	Host station
	CPU type	Select an FXCPU to be used.
Finished	Comment	(Optional)

■ IP address specified communication

Communication Setting Wizard	Item	Setting example (IP address specified communication)
Introduction	Logical station number	0 to 1023
PC side	PC side I/F	Ethernet board
	Connect module	FX3U-ENET-ADP
	Time out	60000
PLC side	PLC side I/F	FX3U-ENET-ADP
	Host (IP Address)	192.168.0.1
Finished	Comment	(Optional)



When using the built-in Ethernet CPU search, click the [Find CPU (Built-in Ethernet port) on network] button. Built-in Ethernet CPUs on the same network is the target of the search.

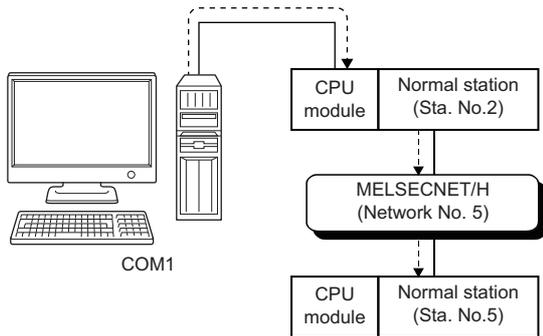


Choose the target CPU from the list and clicking the [OK] button.

An IP address can be applied to the "Communication Setting Wizard - PLC side" setting.

6.6 CPU COM Communication

This section provides the CPU COM communication procedure and its setting example using the utility setting type.



Access procedure

The following shows the procedure before accessing a CPU module.

Operating procedure

1. Connect a module and personal computer.
 ☞ Page 24 SYSTEM CONFIGURATIONS
2. Set a logical station number with Communication Settings Utility.
 ☞ Page 85 Logical station number setting example
3. Perform a communication test to check the communication.
 ☞ Page 61 [Communication test] tab

Logical station number setting example

For the operating procedure of Communication Setting Wizard screens, refer to the following:

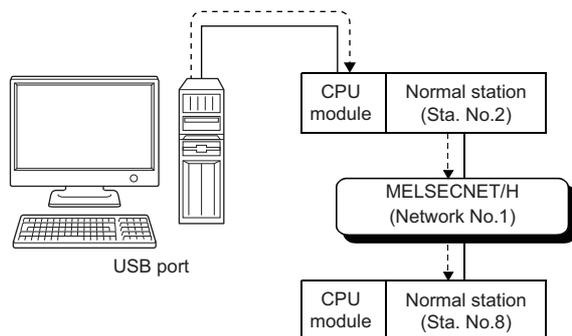
☞ Page 56 Operations on Communication Setting Wizard screens

Setting data

Communication Setting Wizard	Item	Setting example
Introduction	Logical station number	0 to 1023
PC side	PC side I/F	Serial
	Connect port	COM1
	Time out	10000
PLC side	PLC side I/F	CPU module
	CPU type	Select a CPU module to be used.
	Transmission speed	19200
	Control	DTR or RTS Control
Network	Station type	Other station
	Network	MELSECNET/10(H)
	Mode	MELSECNET/H
Other station	CPU type	Select a CPU module to be used.
	Network No.	5
	Station No.	5
	Multiple CPU	None
Finished	Comment	(Optional)

6.7 CPU USB Communication

This section provides the CPU USB communication procedure and its setting example using the utility setting type.



Access procedure

The following shows the procedure before accessing a CPU module.

Operating procedure

1. Connect a module and personal computer.
☞ Page 24 SYSTEM CONFIGURATIONS
2. Install a USB driver.
☞ Page 669 Installing a USB Driver
3. Set a logical station number with Communication Settings Utility.
☞ Page 86 Logical station number setting example
4. Perform a communication test to check the communication.
☞ Page 61 [Communication test] tab

Logical station number setting example

For the operating procedure of Communication Setting Wizard screens, refer to the following:

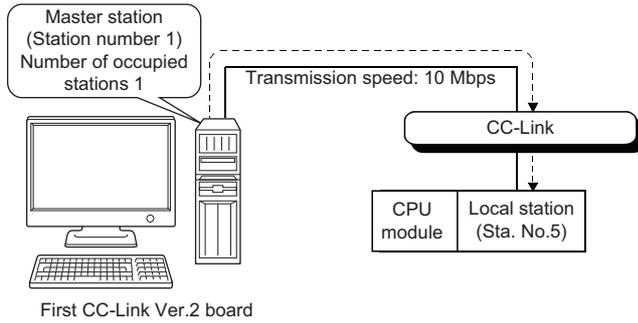
- ☞ Page 56 Operations on Communication Setting Wizard screens

Setting data

Communication Setting Wizard	Item	Setting example
Introduction	Logical station number	0 to 1023
PC side	PC side I/F	USB
	Time out	10000
PLC side	PLC side I/F	CPU module
	CPU type	Select a CPU module to be used.
Network	Station type	Other station
	Network	MELSECNET/10(H)
	Mode	MELSECNET/H
Other station	CPU type	Select a CPU module to be used.
	Network No.	1
	Station No.	8
	Multiple CPU	None
Finished	Comment	(Optional)

6.8 CC-Link Communication

This section provides the CC-Link communication procedure and its setting example using the utility setting type.



Supported CPU and module in CC-Link communication via a personal computer board to an RCPU are as follows.

CC-Link module	Compatible CPU	Station No.
RJ61BT11	RCPU	0 to 64

Supported CPUs and module in CC-Link communication to an FXCPU are as follows.

CC-Link module	Compatible CPU	Station No.
FX3U-64CCL	FX3G, FX3GC, FX3U, FX3UC	1 to 63

- Accessible to an FXCPU with CC-Link communication only via the serial/USB connection of QCPU or via Q series-compatible C24.

Access procedure

The following shows the procedure before accessing a CPU module by using CC-Link communication.

Operating procedure

1. Set a CC-Link Ver.2 board.
☞ Page 24 SYSTEM CONFIGURATIONS, ☞ Page 89 Checking the CC-Link Ver.2 board
2. Connect a personal computer to CC-Link.
☞ Page 24 SYSTEM CONFIGURATIONS, ☞ Page 89 Checking the CC-Link Ver.2 board
3. Set a logical station number with Communication Settings Utility.
☞ Page 88 Logical station number setting example
4. Perform a communication test to check the communication.
☞ Page 61 [Communication test] tab

Logical station number setting example

For the operating procedure of Communication Setting Wizard screens, refer to the following:

☞ Page 56 Operations on Communication Setting Wizard screens

Setting data

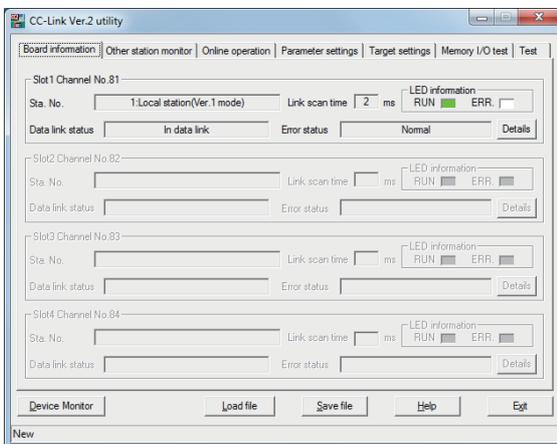
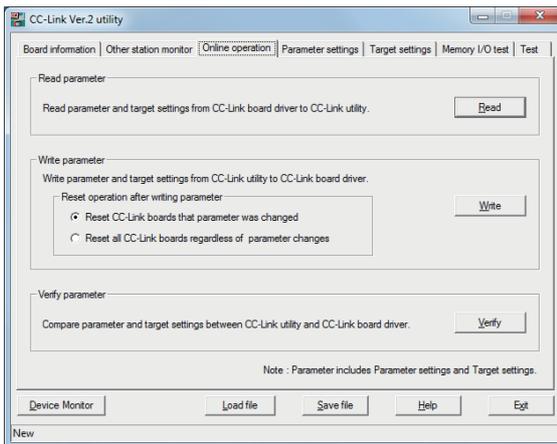
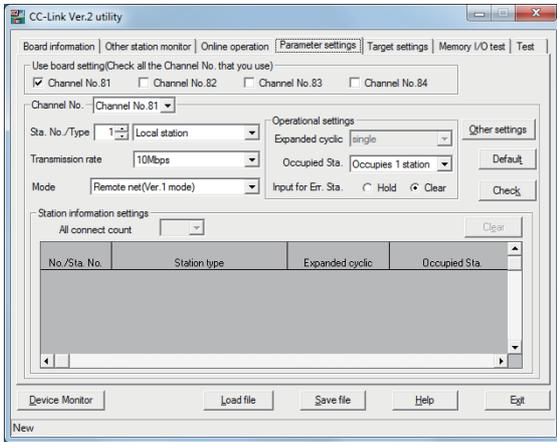
Communication Setting Wizard	Item	Setting example
Introduction	Logical station number	0 to 1023
PC side	PC side I/F	CC-Link board
	Board No.	1st module
Network	Station type	Other station (Single)
Other station	CPU type	Select a CPU module to be used.
	Station No.	5
	Multiple CPU	None
Finished	Comment	(Optional)

Settings of modules

Checking the CC-Link Ver.2 board

Check whether a personal computer is connected properly to CC-Link.

Operating procedure



1. Start MELSEC CC-Link Ver.2 Utility from "MELSEC" in Windows Start.

2. Click the [Parameter settings] tab and set the following items:

Channel No.: Channel No.81

Sta. No.: 1

Type: Local station

Transmission rate: 10Mbps

Mode: Remote net (Ver.1 mode)

Occupied Sta.: Occupies 1 station

Input for Err Sta.: Clear

3. Click the [Online operation] tab and click the [Write] button.

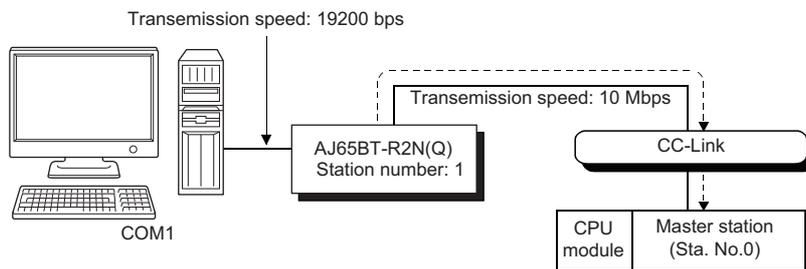
4. Click the [Board information] tab.

Check that the loop of the own station is normal.

5. Click the [Exit] button to exit from the utility.

6.9 CC-Link system RS-232 interface Communication

This section provides the CC-Link system RS-232 interface communication procedure and its setting example using the utility setting type.



Access procedure

The following shows the procedure before accessing a CPU module by using CC-Link system RS-232 interface communication.

Operating procedure

1. Set the operation setting DIP switches of an AJ65BT-R2N module.
📖 CC-Link System RS-232 Interface Module User's Manual
2. Connect the AJ65BT-R2N module to a CC-Link module.
📖 CC-Link System RS-232 Interface Module User's Manual
3. Connect a personal computer to the AJ65BT-R2N module.
📖 CC-Link System RS-232 Interface Module User's Manual
4. Set the network parameters.
📖 CC-Link System RS-232 Interface Module User's Manual
5. Set a logical station number with Communication Settings Utility.
📖 Page 91 Logical station number setting example
6. Perform a communication test to check the communication.
📖 Page 61 [Communication test] tab

Logical station number setting example

For the operating procedure of Communication Setting Wizard screens, refer to the following:

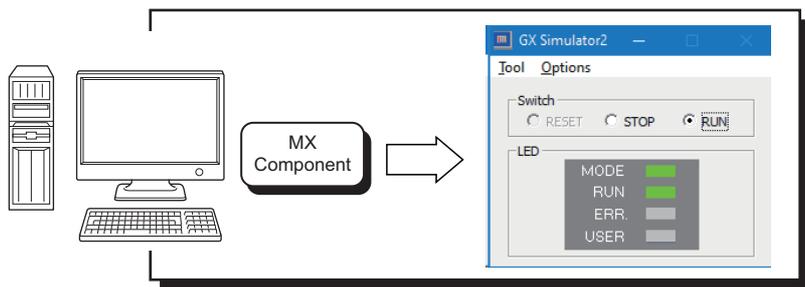
 Page 56 Operations on Communication Setting Wizard screens

Setting data

Communication Setting Wizard	Item	Setting example
Introduction	Logical station number	0 to 1023
PC side	PC side I/F	Serial
	Connect port	COM1
	Time out	10000
PLC side	PLC side I/F	AJ65BT-R2N
	Mode	Q
	Transmission speed	19200
	Control	DTR or RTS Control
Network	Station type	Host station
	CPU type	Select a CPU module to be used.
	CC-Link module No.	0
Finished	Comment	(Optional)

6.10 GX Simulator2 Communication

This section provides GX Simulator2 communication procedure and its setting example using the utility setting type.



Point

The simulation function of GX Works2 cannot be terminated even if stopped by GX Works2 while connecting to MX Component.

(The simulation function of GX Works2 cannot be terminated even if GX Works2 is terminated.)

The simulation function of GX Works2 can be terminated after disconnected to MX Component.

Access procedure

The following shows the procedure before accessing the simulation function of GX Works2 by using GX Simulator2 communication.

Operating procedure

1. Start GX Works2 and the simulation function.
 ☞ Page 24 SYSTEM CONFIGURATIONS
2. Set a logical station number with Communication Settings Utility.
 ☞ Page 92 Logical station number setting example
3. Perform a communication test to check the communication.
 ☞ Page 61 [Communication test] tab

Logical station number setting example

For the operating procedure of Communication Setting Wizard screens, refer to the following:

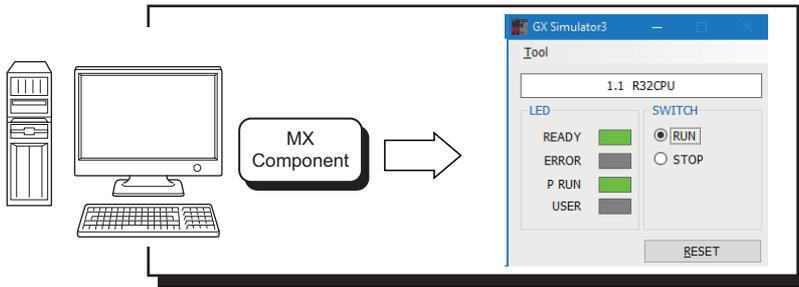
- ☞ Page 56 Operations on Communication Setting Wizard screens

Setting data

Communication Setting Wizard	Item	Setting example
Introduction	Logical station number	0 to 1023
PC side	PC side I/F	GX Simulator2
	Target Simulator	Simulator A
	CPU Series	Select a CPU series to be used.
Finished	Comment	(Optional)

6.11 GX Simulator3 Communication

This section provides GX Simulator3 communication procedure and its setting example using the utility setting type.



Point

The simulation function of GX Works3 can be stopped while connecting to MX Component by using GX Simulator3.

However, it is necessary to open after closing on the communication line of MX Component in order to restart the simulation function of GX Works3.

If the function is restarted without closing on the communication line of MX Component, an error occurs.

6

Access procedure

The following shows the procedure before accessing the simulation function of GX Works3 by using GX Simulator3 communication.

Operating procedure

1. Start GX Works3 and the simulation function.
☞ Page 24 SYSTEM CONFIGURATIONS
2. Set a logical station number with Communication Settings Utility.
☞ Page 94 Logical station number setting example
3. Perform a communication test to check the communication.
☞ Page 61 [Communication test] tab

Logical station number setting example

For the operating procedure of Communication Setting Wizard screens, refer to the following:

 Page 56 Operations on Communication Setting Wizard screens

Setting data

Communication Setting Wizard	Item	Setting example
Introduction	Logical station number	0 to 1023
PC side	PC side I/F	GX Simulator3
	CPU type	Select a CPU module to be used.
	Set Port No.*1	Unselected
	System No.	1
	PLC No.*2	1
	Time out	10000
Finished	Comment	(Optional)

*1 When specifying a port number, select the checkbox and then specify a port number.

The target port number is calculated with the following calculation formula.

PortNumber=5500+ System No. × 10+ PLC No

<Example> For System No. = 1, PLC No = 1

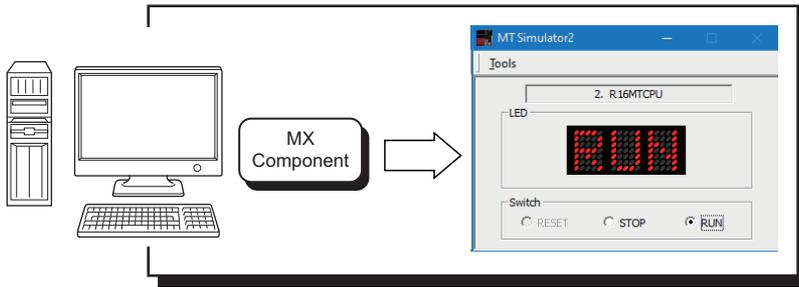
5511=5500+1×10+1

*2 For the system number and PLC number, refer to the following:

 GX Works3 Operating Manual

6.12 MT Simulator2 Communication

This section provides MT Simulator2 communication procedure and its setting example using the utility setting type.



Point

The communication will be disconnected when the simulation function is stopped by MT Developer2 while communicating to MX Component.

Access procedure

The following shows the procedure before accessing the simulation function of MT Developer2 by using MT Simulator2 communication.

Operating procedure

1. Start MT Developer2 and the simulation function.
[Page 24 SYSTEM CONFIGURATIONS](#)
2. Set a logical station number with Communication Settings Utility.
[Page 95 Logical station number setting example](#)
3. Perform a communication test to check the communication.
[Page 61 \[Communication test\] tab](#)

Logical station number setting example

For the operating procedure of Communication Setting Wizard screens, refer to the following:

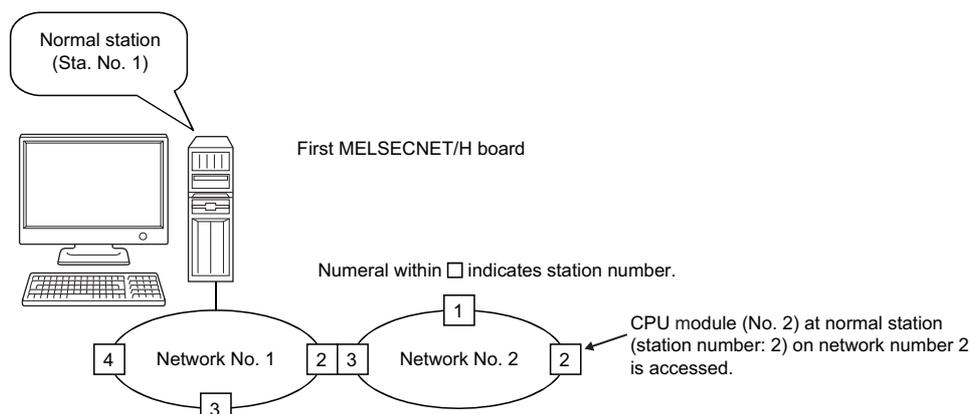
- [Page 56 Operations on Communication Setting Wizard screens](#)

Setting data

Communication Setting Wizard	Item	Setting example
Introduction	Logical station number	0 to 1023
PC side	PC side I/F	MT Simulator2
	Target Simulator	Select a target simulator to be used.
Finished	Comment	(Optional)

6.13 MELSECNET/H Communication

This section provides the MELSECNET/H communication procedure and its setting example using the utility setting type.



Access procedure

The following shows the procedure before accessing a CPU module by using MELSECNET/H communication.

Operating procedure

1. Set an MELSECNET/H board.
2. Connect a personal computer to MELSECNET/H.
 ☞ Page 24 SYSTEM CONFIGURATIONS, ☞ Page 97 Checking the MELSECNET/H board
3. Set a logical station number with Communication Settings Utility.
 ☞ Page 96 Logical station number setting example
4. Perform a communication test to check the communication.
 ☞ Page 61 [Communication test] tab

Logical station number setting example

For the operating procedure of Communication Setting Wizard screens, refer to the following:

- ☞ Page 56 Operations on Communication Setting Wizard screens

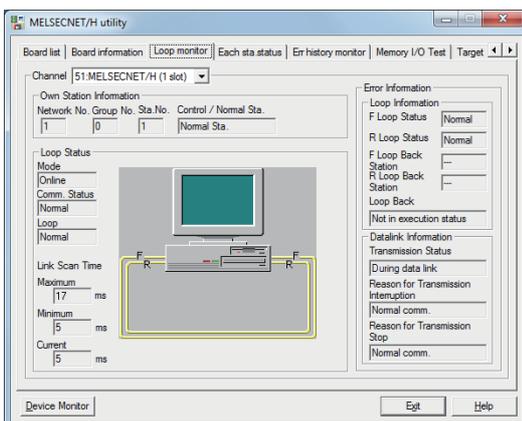
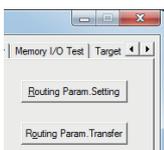
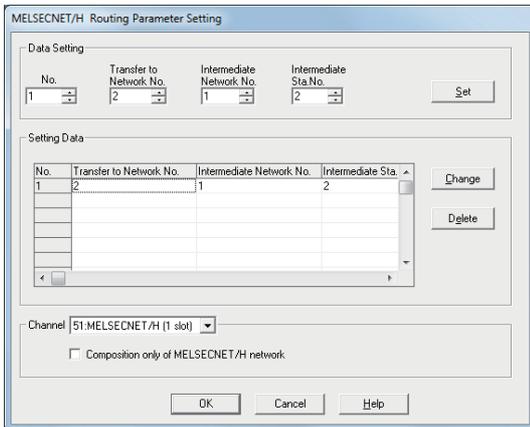
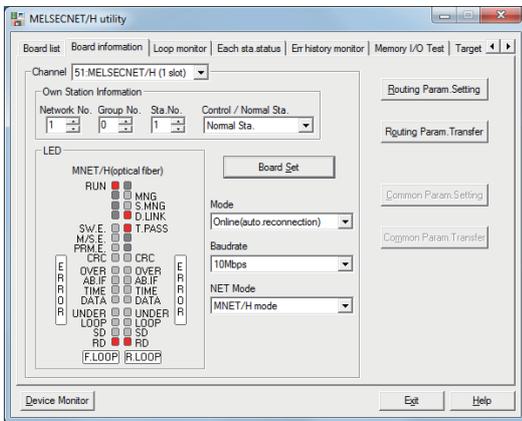
Setting data

Communication Setting Wizard	Item	Setting example
Introduction	Logical station number	0 to 1023
PC side	PC side I/F	MELSECNET/H board
	Board No.	1st module
Network	Station type	Other station (Single)
Other station	CPU type	Select a CPU module to be used.
	Network No.	2
	Station No.	2
	Multiple CPU	No. 2
Finished	Comment	(Optional)

Checking the MELSECNET/H board

Check whether a personal computer is connected properly to MELSECNET/H.

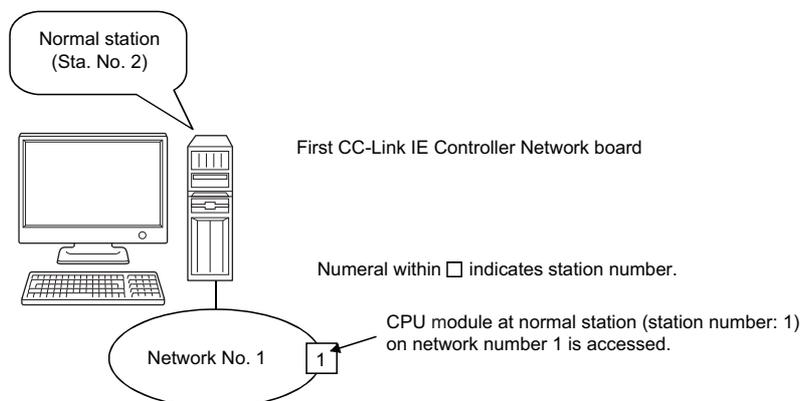
Operating procedure



1. Start MELSECNET/H Utility from "MELSEC" in Windows Start.
2. Click the [Board information] tab and set the following items, and click the [Board Set] button.
Channel: 51: MELSECNET/H (1 slot)
Mode: Online (auto.reconnection)
Baudrate: Any (10Mbps is set for this example.)
NET mode: MNET/H mode
Then, click the [Routing Param.Setting] button.
3. Set the following items and click the [Set] button.
Transfer to Network No.: 2
Intermediate Network No.: 1
Intermediate Sta.No.: 2
Then, click the [OK] button to close the dialog box.
4. Click the [Routing Param.Transfer] button to transfer the routing parameters to the MELSECNET/H board.
5. Click the [Loop monitor] tab.
Check that the loop is normal.
6. Click the [Exit] button to exit from the utility.

6.14 CC-Link IE Controller Network Communication

This section provides the CC-Link IE Controller Network communication procedure and its setting example using the utility setting type.



Supported CPU and modules in CC-Link IE Controller Network communication via a personal computer board to RCPU are as follows.

Module	Compatible CPU	Station No.
RJ71GP21-SX	RCPU	Refer to the manual of a module to be used.
RJ71EN71*1	RCPU	

*1 CCIEC setting or CCIEC + Ether setting only

Access procedure

The following shows the procedure before accessing a CPU module by using CC-Link IE Controller Network communication.

Operating procedure

1. Set a CC-Link IE Controller Network board.
2. Connect a personal computer to CC-Link IE Controller Network.
[Page 24 SYSTEM CONFIGURATIONS](#), [Page 99 Checking the CC-Link IE Controller Network board](#)
3. Set a logical station number with Communication Settings Utility.
[Page 98 Logical station number setting example](#)
4. Perform a communication test to check the communication.
[Page 61 \[Communication test\] tab](#)

Logical station number setting example

For the operating procedure of Communication Setting Wizard screens, refer to the following:

[Page 56 Operations on Communication Setting Wizard screens](#)

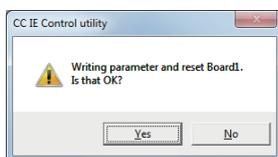
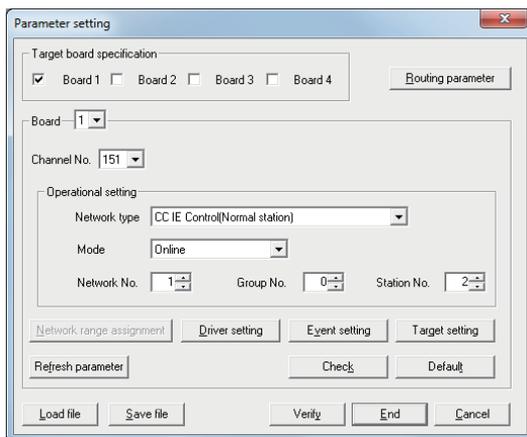
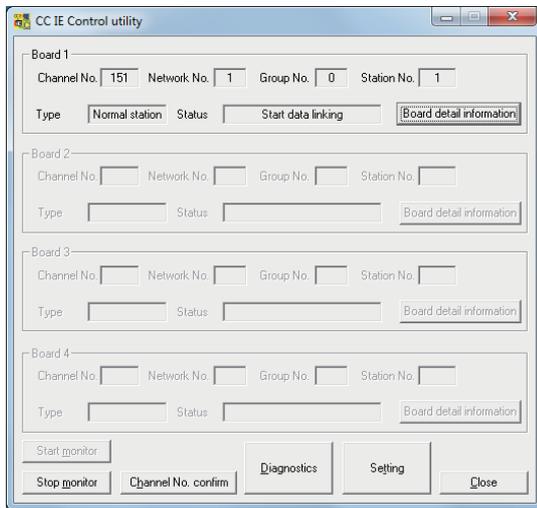
Setting data

Communication Setting Wizard	Item	Setting example
Introduction	Logical station number	0 to 1023
PC side	PC side I/F	CC IE Control board
	Board No.	1st module
Network	Station type	Host station
Finished	Comment	(Optional)

Checking the CC-Link IE Controller Network board

Check whether a personal computer is connected properly to CC-Link IE Controller Network.

Operating procedure



1. Start CC IE Control Utility from "MELSEC" in Windows Start.

2. Click the [Setting] button.

3. Set the following items and click the [End] button.

Channel No.: 151

Network type: CC IE Control (Normal station)

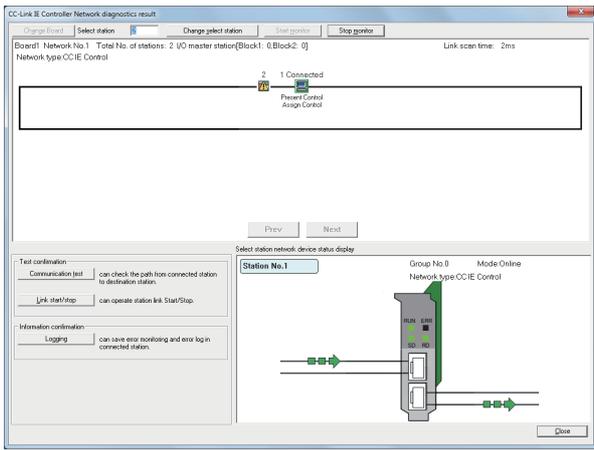
Mode: Online

Network No.: 1

Group No.: 0

Station No.: 2

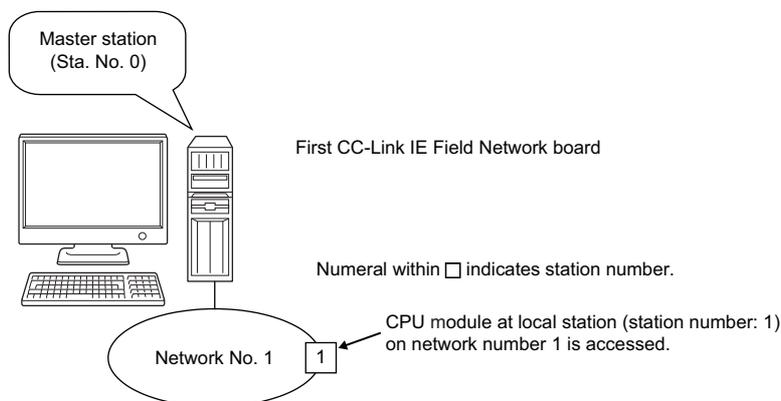
4. Click the [Yes] button and write the parameter to the CC-Link IE Controller Network board.



5. Click the [Diagnostics] button on the CC IE Control utility screen.
Check that the loop is normal.
6. Click the [Close] button to exit from the utility.

6.15 CC-Link IE Field Network Communication

This section provides the CC-Link IE Field Network communication procedure and its setting example using the utility setting type.



Supported CPU and modules in CC-Link IE Field Network communication via a personal computer board to RCPU are as follows.

Module	Compatible CPU	Station No.
RJ71GF11-T2	RCPU	Refer to the manual of a module to be used.
RJ71EN71*1	RCPU	

*1 CCIEF setting or CCIEF + Ether setting only

Access procedure

The following shows the procedure before accessing a CPU module by using CC-Link IE Field Network communication.

Operating procedure

1. Set a CC-Link IE Field Network board.
 ↳ Page 24 SYSTEM CONFIGURATIONS, ↳ Page 103 Checking the CC-Link IE Field Network board
2. Connect a personal computer to CC-Link IE Field Network.
 ↳ Page 102 Logical station number setting example
3. Set a logical station number with Communication Settings Utility.
 ↳ Page 61 [Communication test] tab
4. Perform a communication test to check the communication.

Logical station number setting example

For the operating procedure of Communication Setting Wizard screens, refer to the following:

☞ Page 56 Operations on Communication Setting Wizard screens

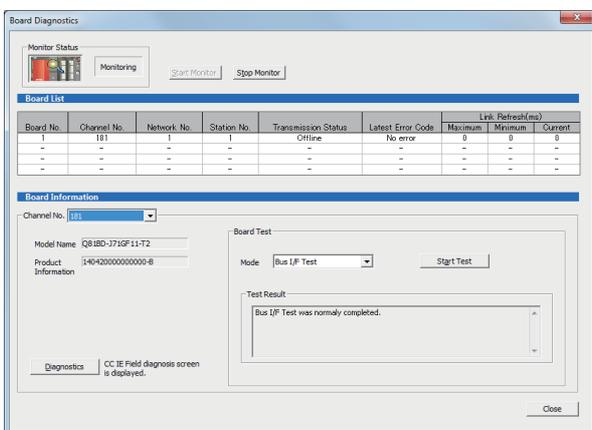
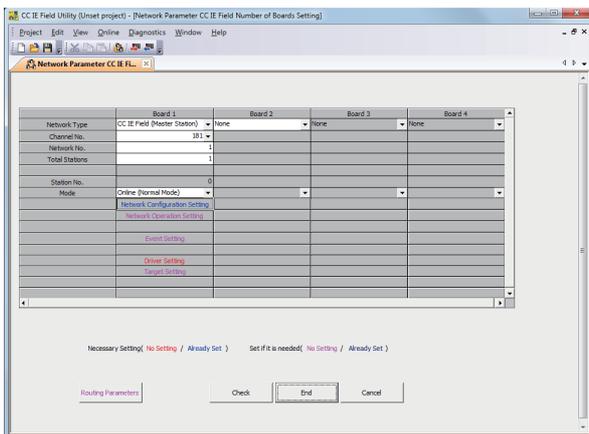
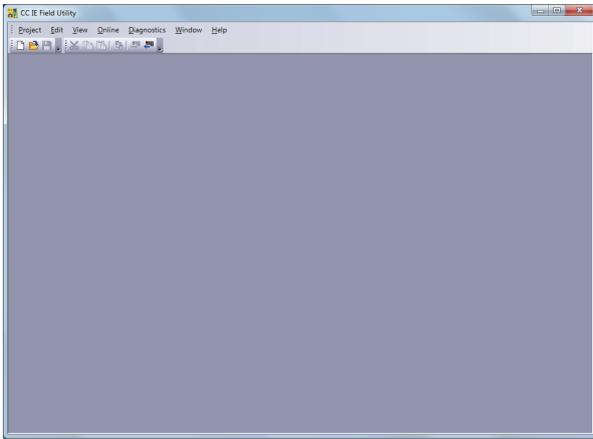
Setting data

Communication Setting Wizard	Item	Setting example
Introduction	Logical station number	0 to 1023
PC side	PC side I/F	CC IE Field board
	Board No.	1st module
Network	Station type	Other station (Single)
Other station	CPU type	Select a CPU module to be used.
	Network No.	1
	Station No.	1
	Multiple CPU	None
Finished	Comment	(Optional)

Checking the CC-Link IE Field Network board

Check whether a personal computer is connected properly to CC-Link IE Field Network.

Operating procedure



1. Start CC IE Field Utility of CC IE Field Board from "MELSEC" in Windows Start.

2. Select [Project] ⇒ [New].

3. Set the following items and click the [End] button.

Network Type: CC IE Field (Master Station)

Channel No.: 181

Network No.: 1

Total Stations: 1

Mode: Online (Normal Mode)

4. Select [Online] ⇒ [Write to Board].

Write the parameter settings of the project to the CC-Link IE Field Network board.

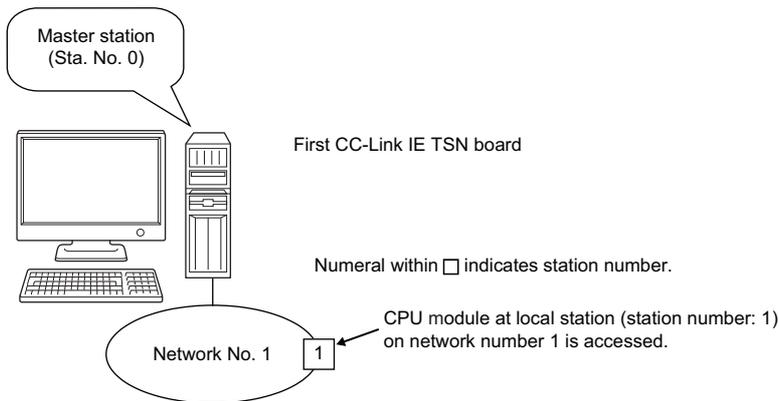
5. Select [Diagnostics] ⇒ [Board Diagnostics].

Check that the loop is normal.

6. Click the [Close] button to exit from the utility.

6.16 CC-Link IE TSN Communication

This section provides the CC-Link IE TSN communication procedure and its setting example using the utility setting type.



Supported CPU and modules in CC-Link IE TSN communication via a personal computer board to RCPU are as follows.

Module	Compatible CPU	Station No.
RJ71GN11-T2	RCPU	Refer to the manual of a module to be used.
RJ71GN11-SX	RCPU	

Access procedure

The following shows the procedure before accessing a CPU module by using CC-Link IE TSN communication.

Operating procedure

1. Set a CC-Link IE TSN board.
2. Connect a personal computer to CC-Link IE TSN.
☞ Page 24 SYSTEM CONFIGURATIONS
3. Set a logical station number with Communication Settings Utility.
☞ Page 105 Logical station number setting example
4. Perform a communication test to check the communication.
☞ Page 61 [Communication test] tab

Logical station number setting example

For the operating procedure of Communication Setting Wizard screens, refer to the following:

 Page 56 Operations on Communication Setting Wizard screens

Setting data

Communication Setting Wizard	Item	Setting example
Introduction	Logical station number	0 to 1023
PC side	PC side I/F	CC IE TSN board
	Board No.	1st module
Network	Station type	Other station (Single)
Other station	CPU type	Select a CPU module to be used.
	Network No.	1
	Station No.	1
	Multiple CPU	None
Finished	Comment	(Optional)

Checking the CC-Link IE TSN board

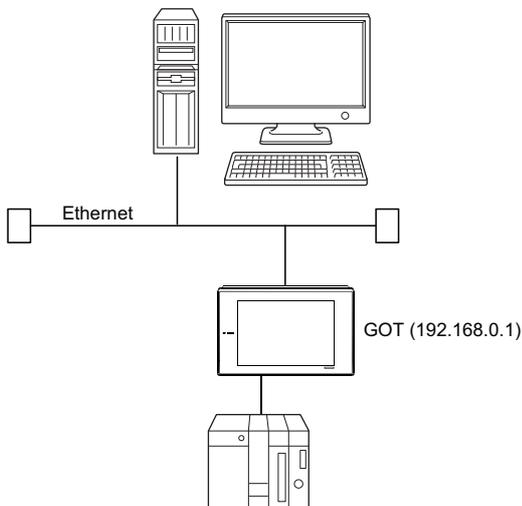
Check whether a personal computer is connected properly to CC-Link IE TSN.

For details on the checking method, refer to the following:

 CC-Link IE TSN Interface Board User's Manual

6.17 GOT Gateway Device Communication

This section explains the procedure for the GOT gateway device communication and its setting example using the utility setting type.



Access procedure

The following shows the procedure before accessing a GOT by using GOT gateway device communication.

Operating procedure

1. Set a GOT.
📖 GOT operating manual to be used
2. Connect the GOT to Ethernet.
📖 GOT operating manual to be used
3. Connect a personal computer to Ethernet.
📖 GOT operating manual to be used
4. Edit a HOSTS file.
The HOSTS file is not required to be edited when entering an IP address in a host name (IP Address) of Communication Settings Utility and the ActHostAddress property.
5. Set a logical station number with Communication Settings Utility.
📖 Page 107 Logical station number setting example
6. Perform a communication test to check the communication.
📖 Page 61 [Communication test] tab

Logical station number setting example

For the operating procedure of Communication Setting Wizard screens, refer to the following:

☞ Page 56 Operations on Communication Setting Wizard screens

Setting data

Communication Setting Wizard	Item	Setting example
Introduction	Logical station number	0 to 1023
PC side	PC side I/F	Ethernet board
	Connect module	GOT
	Port No.	5011
	Time out	60000
PLC side	PLC side I/F	Ethernet module
	Module type	GOT
	Host (IP Address)	192.168.0.1
Finished	Comment	(Optional)

Checking the communication

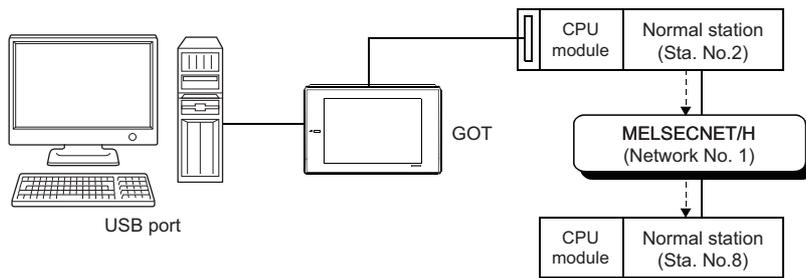
After completion of preparations for GOT gateway device communication, execute ping in the MS-DOS mode to check connection before starting communications on MX Component.

- When normal
C:\>ping 192.168.0.1
Reply from 192.168.0.1 : bytes=32 time<10ms TTL=32
- When abnormal
C:\>ping 192.168.0.1
Request timed out.

If ping does not pass through, check a cable and module connections and Windows side IP address and other settings.

6.18 GOT Transparent Communication

This section provides the GOT transparent communication procedure and its setting example using the utility setting type.



Point

For applicable system configuration, refer to the following:

📖 Connection manuals for GOT2000 series

(Mitsubishi Products), (Non-Mitsubishi Products 1), (Non-Mitsubishi Products 2), (Microcomputer, MODBUS/Fieldbus Products, Peripherals)

Access procedure

The following shows the procedure before accessing a GOT by using the GOT transparent function.

Operating procedure

1. Set a GOT.
📖 GOT operating manual to be used
2. Connect the GOT and a personal computer.
📖 GOT operating manual to be used
3. Connect the GOT and a CPU module.
📖 GOT operating manual to be used
4. Install a USB driver.
📖 Page 669 Installing a USB Driver
5. Set a logical station number with Communication Settings Utility.
📖 Page 109 Logical station number setting example
6. Perform a communication test to check the communication.
📖 Page 61 [Communication test] tab

Logical station number setting example

For the operating procedure of Communication Setting Wizard screens, refer to the following:

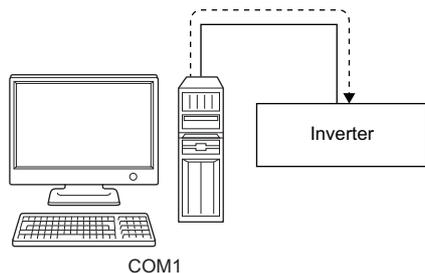
☞ Page 56 Operations on Communication Setting Wizard screens

Setting data

Communication Setting Wizard	Item	Setting example
Introduction	Logical station number	0 to 1023
PC side	PC side I/F	USB (via GOT)
	Time out	10000
GOT side	GOT - PLC I/F	Serial
	Connect module	CPU module
PLC side	PLC side I/F	CPU module
	CPU type	Select a CPU module to be used.
Network	Station type	Other station
	Network	MELSECNET/10(H)
	Mode	MELSECNET/H
Other station	CPU type	Select a CPU module to be used.
	Network No.	1
	Station No.	8
	Multiple CPU	None
Finished	Comment	(Optional)

6.19 Inverter COM Communication

This section provides the inverter COM communication procedure and its setting example using the utility setting type.



Access procedure

The following shows the procedure before accessing an inverter by using inverter COM communication.

Operating procedure

1. Connect an inverter and personal computer.
☞ Page 24 SYSTEM CONFIGURATIONS
2. Set a logical station number with Communication Settings Utility.
☞ Page 110 Logical station number setting example
3. Perform a communication test to check the communication.
☞ Page 61 [Communication test] tab

Logical station number setting example

For the operating procedure of Communication Setting Wizard screens, refer to the following:

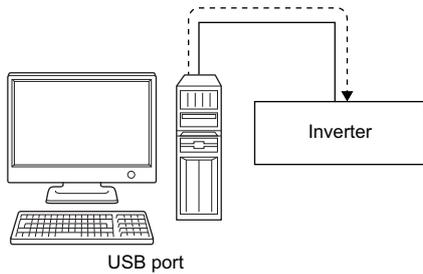
☞ Page 56 Operations on Communication Setting Wizard screens

Setting data

Communication Setting Wizard	Item	Setting example
Introduction	Logical station number	0 to 1023
PC side	PC side I/F	Serial
	Connect port	COM1
	Time out	10000
PLC side	PLC side I/F	Inverter
	Inverter	A800
	Station No.	0
	Transmission speed	19200
	Parity	Even
	Data bit	8
	Stop bit	1
	Delimiter	CR
	Transmission wait time	0
Finished	Comment	(Optional)

6.20 Inverter USB Communication

This section provides the inverter USB communication procedure and its setting example using the utility setting type.



Access procedure

The following shows the procedure before accessing an inverter by using inverter USB communication.

Operating procedure

1. Connect an inverter and personal computer.
[Page 24 SYSTEM CONFIGURATIONS](#)
2. Install a USB driver.
[Page 669 Installing a USB Driver](#)
3. Set a logical station number with Communication Settings Utility.
[Page 111 Logical station number setting example](#)
4. Perform a communication test to check the communication.
[Page 61 \[Communication test\] tab](#)

Logical station number setting example

For the operating procedure of Communication Setting Wizard screens, refer to the following:

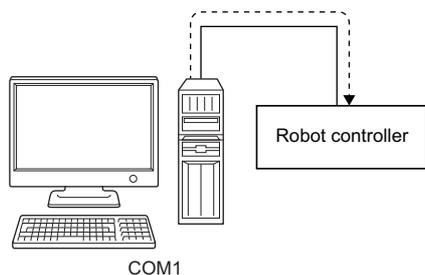
- [Page 56 Operations on Communication Setting Wizard screens](#)

Setting data

Communication Setting Wizard	Item	Setting example
Introduction	Logical station number	0 to 1023
PC side	PC side I/F	USB
	Time out	10000
PLC side	PLC side I/F	Inverter
	Inverter	A800
	Station No.	0
	Transmission wait time	0
Finished	Comment	(Optional)

6.21 Robot controller COM Communication

This section provides the robot controller COM communication procedure and its setting example using the utility setting type.



Access procedure

The following shows the procedure before accessing a robot controller by using robot controller COM communication.

Operating procedure

1. Connect a robot controller and personal computer.
 ☞ Page 24 SYSTEM CONFIGURATIONS
2. Set a logical station number with Communication Settings Utility.
 ☞ Page 112 Logical station number setting example
3. Perform a communication test to check the communication.
 ☞ Page 61 [Communication test] tab

Logical station number setting example

For the operating procedure of Communication Setting Wizard screens, refer to the following:

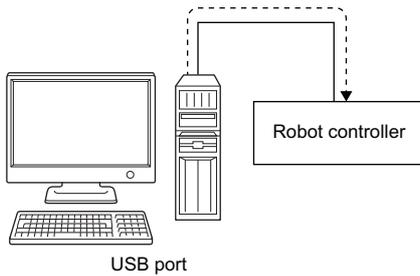
- ☞ Page 56 Operations on Communication Setting Wizard screens

Setting data

Communication Setting Wizard	Item	Setting example
Introduction	Logical station number	0 to 1023
PC side	PC side I/F	Serial
	Connect port	COM1
	Time out	10000
PLC side	PLC side I/F	Robot controller
	R/C type	CRnD-7xx/CR75x-D
	Transmission speed	19200
	Parity	Even
	Character Size	8
	Stop bit	1
	Send timeout	5000
	Receive timeout	30000
	Retries	3
	Usage protocol	Procedural
Finished	Comment	(Optional)

6.22 Robot controller USB Communication

This section provides the robot controller USB communication procedure and its setting example using the utility setting type.



Access procedure

The following shows the procedure before accessing a robot controller by using robot controller USB communication.

Operating procedure

1. Connect a robot controller and personal computer.
 ☞ Page 24 SYSTEM CONFIGURATIONS
2. Install a USB driver.
 ☞ Page 669 Installing a USB Driver
3. Set a logical station number with Communication Settings Utility.
 ☞ Page 113 Logical station number setting example
4. Perform a communication test to check the communication.
 ☞ Page 61 [Communication test] tab

Logical station number setting example

For the operating procedure of Communication Setting Wizard screens, refer to the following:

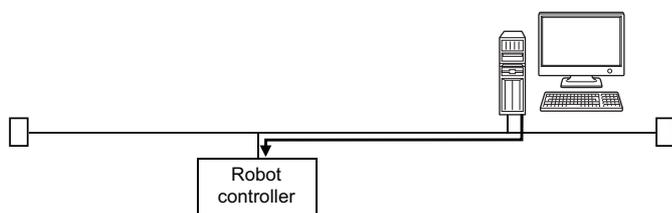
- ☞ Page 56 Operations on Communication Setting Wizard screens

Setting data

Communication Setting Wizard	Item	Setting example
Introduction	Logical station number	0 to 1023
PC side	PC side I/F	USB
	Time out	10000
PLC side	PLC side I/F	Robot controller
	R/C type	CRnD-7xx/CR75x-D
	Send timeout	5000
	Receive timeout	30000
	Retries	3
Finished	Comment	(Optional)

6.23 Robot controller Ethernet Communication

This section provides the robot controller Ethernet communication procedure and its setting example using the utility setting type.



Access procedure

The following shows the procedure before accessing a robot controller by using robot controller Ethernet communication.

Operating procedure

1. Connect a robot controller to Ethernet.
📖 Robot controller Manual to use
2. Connect a personal computer to Ethernet.
3. Set a logical station number with Communication Settings Utility.
📖 Page 114 Logical station number setting example
4. Perform a communication test to check the communication.
📖 Page 61 [Communication test] tab

Logical station number setting example

For the operating procedure of Communication Setting Wizard screens, refer to the following:

📖 Page 56 Operations on Communication Setting Wizard screens

Setting data

Communication Setting Wizard	Item	Setting example
Introduction	Logical station number	0 to 1023
PC side	PC side I/F	Ethernet board
	Connect module	Robot controller
	Port No.	10001
PLC side	PLC side I/F	Robot controller
	R/C type	CRnD-7xx/CR75x-D
	Host (IP Address)	192.168.0.1
	Send timeout	5000
	Receive timeout	30000
	Retries	3
Finished	Comment	(Optional)

7 PROGRAMMING

This chapter explains the method for creating labels used in a program, coding, debugging by utilizing PLC Monitor Utility.

Process	Overview	Reference
Creating labels	Explains Label Utility used when creating a label.	Page 115 Creating a Label With Label Utility
Coding	Explains the method for coding by using the functions and properties provided by a control.	Page 142 Coding
Debugging	Explains PLC Monitor Utility used for debugging.	Page 143 Debugging

7.1 Creating a Label With Label Utility

When using a label in a program, create a label with Label Utility firstly. This section explains the operation and setting method of Label Utility.

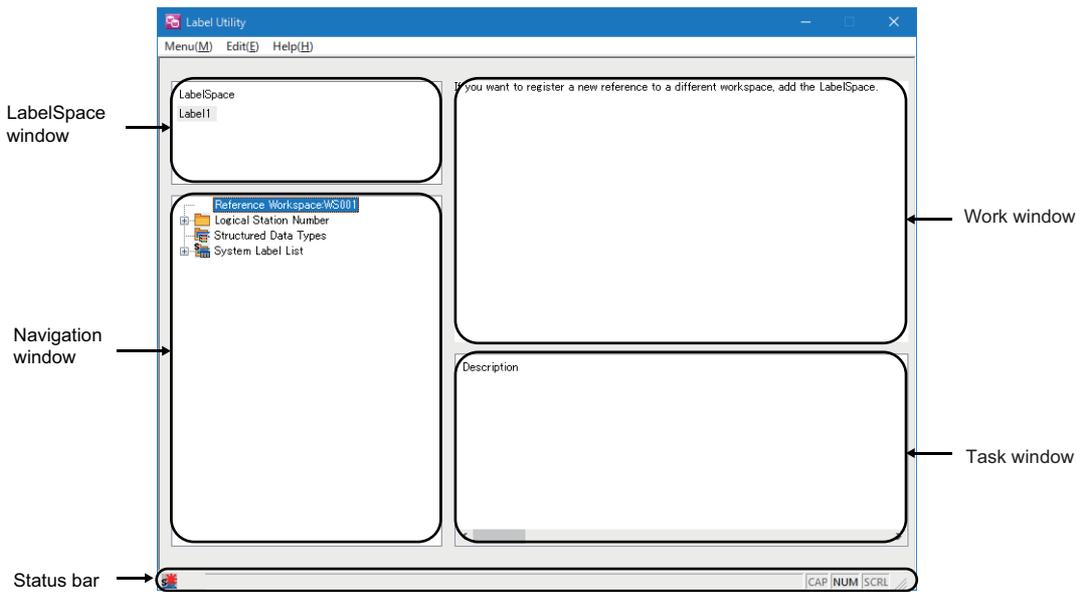
Starting Label Utility

Operating procedure

Start Label Utility from "MELSOFT" in Windows Start.

Label screen

Window



Item	Description	Reference
LabelSpace window	Displays the list of LabelSpace. The LabelSpace names are displayed on the list in ascending order based on the date of creation.	—
Navigation window	Displays the logical station number, structured data type and system label list of LabelSpace in tree format.	—
Work window	Displays the system label list and the Structure Setting screen.	Page 118 System label list Page 123 Structure setting
Task window	Displays the processing result in list format.	—
Status bar	Displays the current status.	Page 125 Change notification

Basic operations

Operating procedure

■ Adding LabelSpace

- Select [Menu] ⇒ [Add LabelSpace].
- Right-click the LabelSpace window, and select [Add LabelSpace].

For system label Ver.1, select [Add LabelSpace]. As for system label Ver.2, select [Add LabelSpace Ver.2].

Point

The maximum number of LabelSpaces is 8. Note that the sum of LabelSpaces and referred Workspaces should be 16 or less.

■ Changing LabelSpace name

- Select the LabelSpace name on the LabelSpace window, and select [Menu] ⇒ [Rename LabelSpace].
- Right-click the LabelSpace name on the LabelSpace window, and select [Rename LabelSpace].

Point

Up to 32 characters can be used in the LabelSpace name.

■ Saving LabelSpace

- Select [Menu] ⇒ [Save LabelSpace].
- Exit Label Utility.

■ Deleting LabelSpace

- Select the LabelSpace name on the LabelSpace window, and select [Menu] ⇒ [Delete LabelSpace].
- Right-click the LabelSpace name on the LabelSpace window, and select [Delete LabelSpace].

Registering/deleting a logical station number

Registration

Register a logical station number to LabelSpace.

Operating procedure

- Select a LabelSpace name on the LabelSpace window, and select [Menu] ⇒ [Logical Station Number] ⇒ [Register].
- Select a LabelSpace name on the LabelSpace window, right-click "Logical Station Number" on the Navigation window, and select [Logical Station Number] ⇒ [Register].

Window



Item	Description
Logical station number	Select the logical station number defined in Communication Settings Utility from the list box. The logical station number which has already been registered to other LabelSpace is not displayed.
[Communication setting] button	Click this to start Communication Settings Utility.
[Register] button	Click this to register the logical station number, and close the screen.
[Cancel] button	Click this to close the screen without registering the logical station number.

Deletion

Delete the logical station number registered to LabelSpace.

Operating procedure

- Select the station number to be deleted from the Navigation window, and select [Menu] ⇒ [Logical Station Number] ⇒ [Deregister].
- Right-click the station number to be deleted from the Navigation window, and select [Logical Station Number] ⇒ [Deregister].

Point

- After registering the logical station number, save the LabelSpace.
- The logical station number and LabelSpace will be registered on a 1:1 basis.
When the logical station number is registered in some other label space, delete the registered logical station number from the label space and then register a new one.

System label list

Register, edit, and browse a system label.

Window

<System label Ver.1>				<System label Ver.2>					
	Del...	System Label Name	Data Type	Device		Del...	System Label Name	Data Type	Label Name
1	<input type="checkbox"/>	LABEL.11	Bit		1	<input type="checkbox"/>	LABEL.5	Bit	SLABEL.5
2	<input type="checkbox"/>	LABEL.12	Word[Signed]	D0	2	<input type="checkbox"/>	LABEL.6	Double Word[Signed]	SLABEL.6
3	<input type="checkbox"/>	LABEL.13	Double Word[Signed]		3	<input type="checkbox"/>	LABEL.7	Long Timer	SLABEL.7
4	<input type="checkbox"/>	LABEL.14	FLOAT[Single Precision]		4	<input type="checkbox"/>	LABEL.8	String[UNICODE]	
5					5				

Item	Description	Maximum number of characters	
		Ver.1	Ver.2
Delete	Select the system label to be deleted.	—	—
System Label Name	Enter a desired system label name. For characters that cannot be used for label names, refer to ( Page 663 Character Strings that cannot be Used for Label Names).	32 characters	256 characters
Data Type	Set a data type from the "Data Type Selection" screen displayed by clicking [...]. ( Page 119 Selecting data types) It can also be entered directly.	128 characters	512 characters
Device	Set the device to be assigned to system label Ver.1. The bit-specified word device (D0.1) can be specified. When the structure is set for data type, "Details setting" is displayed. Set the items on the "Structured Data Device Setting" screen displayed by clicking "Details setting." ( Page 121 Assigning devices to structured data type labels, Page 122 Assigning devices to structure array type labels)	32 characters	—
Label Name	Enter the label name to be managed by system label Ver.2.	—	1763 characters

Basic operations

Operating procedure

■ Adding system label list

- Select a LabelSpace name on the LabelSpace window, and select [Menu] ⇒ [System Label] ⇒ [System Label List] ⇒ [New].
- Select a LabelSpace name on the LabelSpace window, right-click "System Label List" on the Navigation window, and select [New].

■ Changing system label list name

- Select the system label list name on the Navigation window, and select [Menu] ⇒ [System Label] ⇒ [System Label List] ⇒ [Rename].
- Right-click the system label list name on the Navigation window, and select [System Label List] ⇒ [Rename].

Point

Up to 32 characters can be used in the LabelSpace name.

■Deleting LabelSpace

- Select the system label list name to be deleted on the Navigation window, and select [Menu] ⇒ [System Label] ⇒ [System Label List] ⇒ [Delete].
- Right-click the system label list name to be deleted on the Navigation window, and select [System Label List] ⇒ [Delete].
- Select the system label list name to be deleted on the Navigation window, and press the  key.

Selecting data types

The applicable data types are as follows.

○: Supported, —: Not supported

Data type	MELSOFT Navigator		MX Component	
	System label Ver.2	System label Ver.1	LabelSpace Ver.2 ^{*4}	LabelSpace Ver.1
			Simple Types/Array	Simple Types/Array
Bit	○	○	○	○
Word [Signed]	○	○	○	○
Double Word [Signed]	○	○	○	○
Word [Unsigned]/Bit String [16-bit]	○	○	○	○
Double Word [Unsigned]/Bit String [32-bit]	○	○	○	○
Float (Single Precision)	○	○	○	○
Float (Double Precision)	○	○	○	○
String	○	○	○	○
String[Unicode] ^{*1}	○	—	○	—
Time	○	○	○	○
Timer ^{*1}	—	○	—	○
Timer[Unsigned] ^{*1*3}	○	—	○	—
Long timer ^{*1*3}	○	—	○	—
Counter ^{*1}	—	○	—	○
Counter[Unsigned] ^{*1*3}	○	—	○	—
Long counter ^{*1*3}	○	—	○	—
Retentive timer ^{*1}	—	○	—	○
Retentive Timer[Unsigned] ^{*1*3}	○	—	○	—
Retentive long timer ^{*1*3}	○	—	○	—
Structure ^{*2}	○	○	○	○

*1 Select "Simple Types" when arraying.

An error occurs during usage, when "Structure" is selected.

*2 For a label to which devices are not assigned in system label Ver.2, a five-level structure can be used.

For other types, only one-level structure can be used.

*3 To perform reading in system label Ver.2, a target to be read needs to be specified expressly.

Registered by connecting a target to be read (contact ".S"/Coil ".C"/ Current value ".N") to the end of a label name.

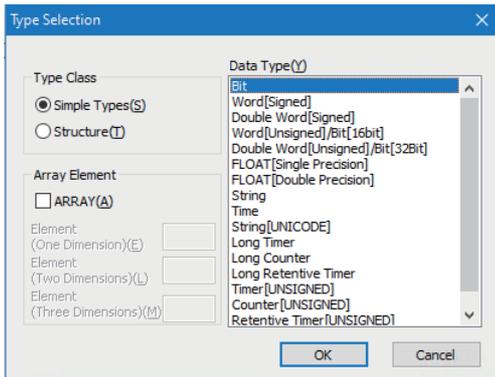
Example) When connecting current value ".N": (label name).N

*4 For a label to which devices are not assigned (labels that uses a label memory), the operation is same as when devices are assigned regardless of the data types.

For bit type, the operation same as bit-specified word device.

Window

Click [...] on the data type input field of the system label list or the structure setting.



Operating procedure

1. Select a type from "Type Class."

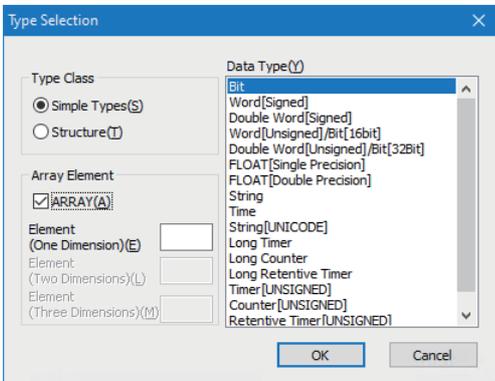
Item	Description
Simple Types	Specify this to select a data type from basic types such as bit, word.
Structure	Specify this to select a data type from the defined structures. (Not displayed for the structured data settings)

2. In the "Data Type" field, select the data type and structure.
3. Click the [OK] button.
The settings are displayed in the "Data Type" column.

■Setting arrays for data type

Define a data type as an array.

To define a data type as an array, set the items of "Array Element" on the "Type Selection" screen.



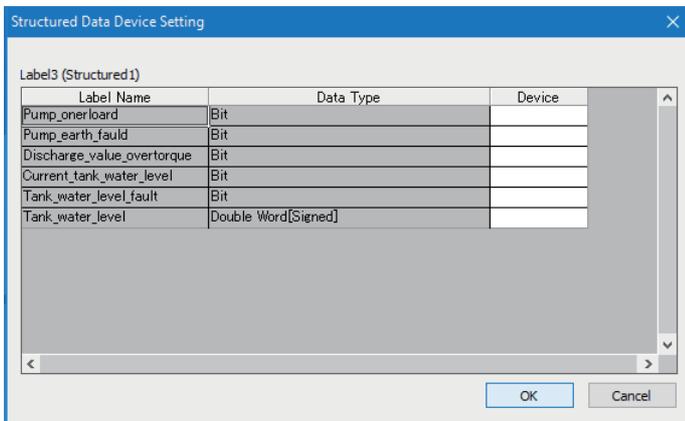
Operating procedure

1. Click [...] on the data type input field of the system label list or the structure setting.
2. Select the check box under "Array Element."
3. Set "Element (One Dimension)," and if necessary, set "Element (Two Dimension)," "Element (Three Dimension)."
4. Set the data type of the array element in the same manner as setting the normal data type.

Assigning devices to structured data type labels

Window

Click "Detail Setting" on the "Device" column of System Label List.



Operating procedure

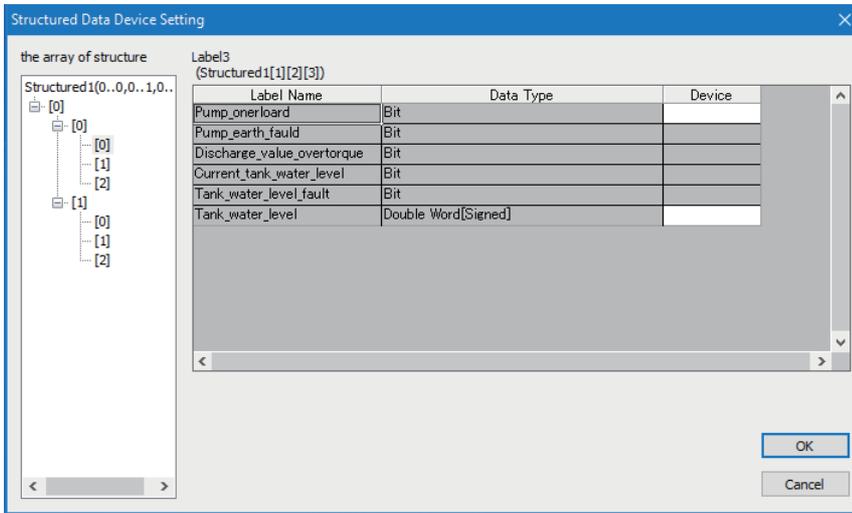
Set the items on the screen.

Item	Description
Label Name	Displays label names defined as structure.
Data Type	Displays data types set to data names.
Device	Set device names to be assigned.

Assigning devices to structure array type labels

Window

Click "Detail Setting" on the "Device" column of System Label List.



Operating procedure

Set the items on the screen.

Item	Description
the array of structure	Displays elements of the structure array in tree format. The device setting of the element selected in the tree is displayed in the right area of the screen.
Label Name	Displays label names defined as structure.
Data Type	Displays data types set to data names.
Device	Set device names to be assigned. Device names can be entered for the start array element only.

Structure setting

Register, edit, and browse each data (element) that is configured in the structure managed in the LabelSpace.

Window

	Label Name	Data Type
1	ST_data1	Bit
2	ST_data2	Bit
3	ST_data3	Bit
4	ST_data4	Bit
5	ST_data5	Bit
6	ST_data6	Bit
7	ST_data7	Bit
8	ST_data8	Bit
9	ST_data9	Bit
10	ST_data10	Bit
11		

Item	Description	Reference
Label Name	Displays a label name.	—
Data Type	Displays a data type of the label. Specify a data type from the "Type Selection" screen displayed by clicking [...].	Page 119 Selecting data types

Basic operations

Operating procedure

■ Adding system label list

- Select a LabelSpace name on the LabelSpace window, and select [Menu] ⇒ [System Label] ⇒ [Structured Data Types] ⇒ [New].
- Select a LabelSpace name on the LabelSpace window, right-click "Structured Data Types" on the Navigation window, and select [Structured Data Types] ⇒ [New].

■ Changing system label list name

- Select the structure name on the Navigation window, and select [Menu] ⇒ [System Label] ⇒ [Structured Data Types] ⇒ [Rename].
- Right-click the structure name on the Navigation window, and select [Structured Data Types] ⇒ [Rename].



Up to 32 characters can be used in the LabelSpace name.

■ Deleting LabelSpace

- Select the structure name to be deleted on the Navigation window, and select [Menu] ⇒ [System Label] ⇒ [Structured Data Types] ⇒ [Delete].
- Right-click the structure name to be deleted on the Navigation window, and select [Structured Data Types] ⇒ [Delete].
- Select the structure name to be deleted on the Navigation window, and press the key.

Referring and registering/canceling Workspaces

Referring and registering

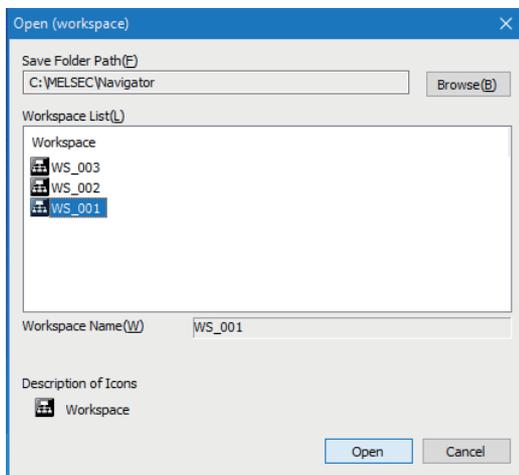
Register a Workspace to be referred when importing a system label.

- A system label can be used in MX Component independently by canceling the reference.
- When registering a canceled reference, the reference is regarded as a new reference. (The canceled references are not restored.)
- The label information within the LabelSpace cannot be edited after the reference registration. (Note that the logical station numbers can be registered.)

Operating procedure

- Select the LabelSpace name on the LabelSpace window, and select [Menu] ⇒ [Workspace] ⇒ [Reference Registration].
- Right-click "Reference Workspace" on the Navigation window, and select [Workspace] ⇒ [Reference Registration].

Window



Item	Description
Save Folder Path	Click the [Browse] button, and select a folder in the "Browse for Folder" screen.
Workspace List	Displays the list of Workspaces.

Point

The structure array labels whose devices are automatically assigned or whose devices are bit-specified cannot be referred and registered properly.
Register them with the system label list.

Canceling

Cancel the reference of registered Workspace for importing the system label.

- The label information within the LabelSpace can be edited after canceling the reference registration.

Operating procedure

- Select the LabelSpace name on the LabelSpace window, and select [Menu] ⇒ [Workspace] ⇒ [Dereference Registration].
- Right-click "Reference Workspace" on the Navigation window, and select [Workspace] ⇒ [Dereference Registration].

Change notification

Change notification

When Workspace is referred and registered, the system label notification icon is displayed by executing the change confirmation when the status-changed labels exist.

Operating procedure

Select [Menu] ⇒ [System Label] ⇒ [Check Changes of System Label Database].

(The change confirmation is also executed when Label Utility starts.)

Window



Point

- For using the change notification function, use MELSOFT Navigator Version 1.39R or later.
- Receive a change notification when executing any of the following operations on the reference Workspace.
(The change notification is not sent when creating a new system label.)
 - Change in system labels (including the change of comment and remark)
 - Deletion of system labels

Applying the changed data

Apply the status-changed label information when Workspace is referred and registered.

When more than one Workspace is referred, apply them for each LabelSpace.

Operating procedure

- Select [Menu] ⇒ [System Label] ⇒ [Change Contents of System Label Database].
- Right-click the system label notification icon (), and select [Change Contents of System Label Database].

Updating system label data

Update the system label data.

Operating procedure

Select [Menu] ⇨ [Refresh system label data for Control].

Exporting/Importing LabelSpace

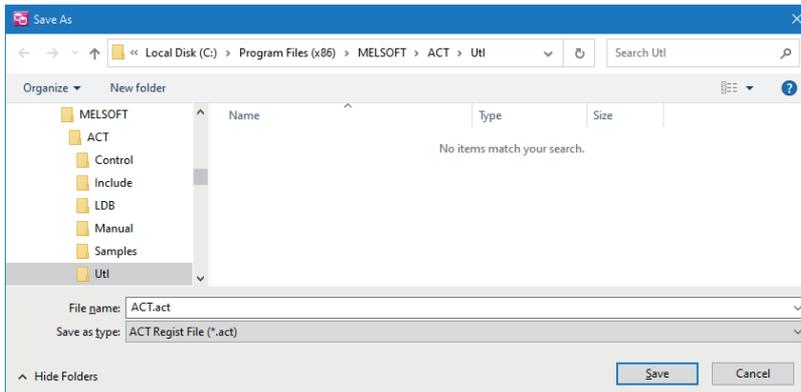
Exporting

Save the information used in Label Utility to a file.

Operating procedure

Select [Menu] ⇒ [Export].

Window



Point

The assignment information of the logical station number is not included in the exported information. When using the exported information by importing it, the assignment of logical station number is required.

Importing

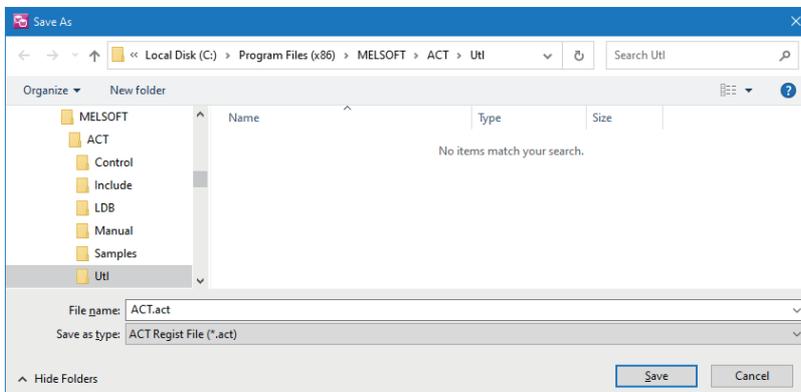
Import the saved information saved in a file by exporting LabelSpace (Page 127 Exporting/Importing LabelSpace) to Label Utility.

To import LabelSpace of MX Component Version 4 to MX Component Version 5, change the number of characters used in the LabelSpace name to less than 32 characters.

Operating procedure

Select [Menu] ⇒ [Import].

Window



System label

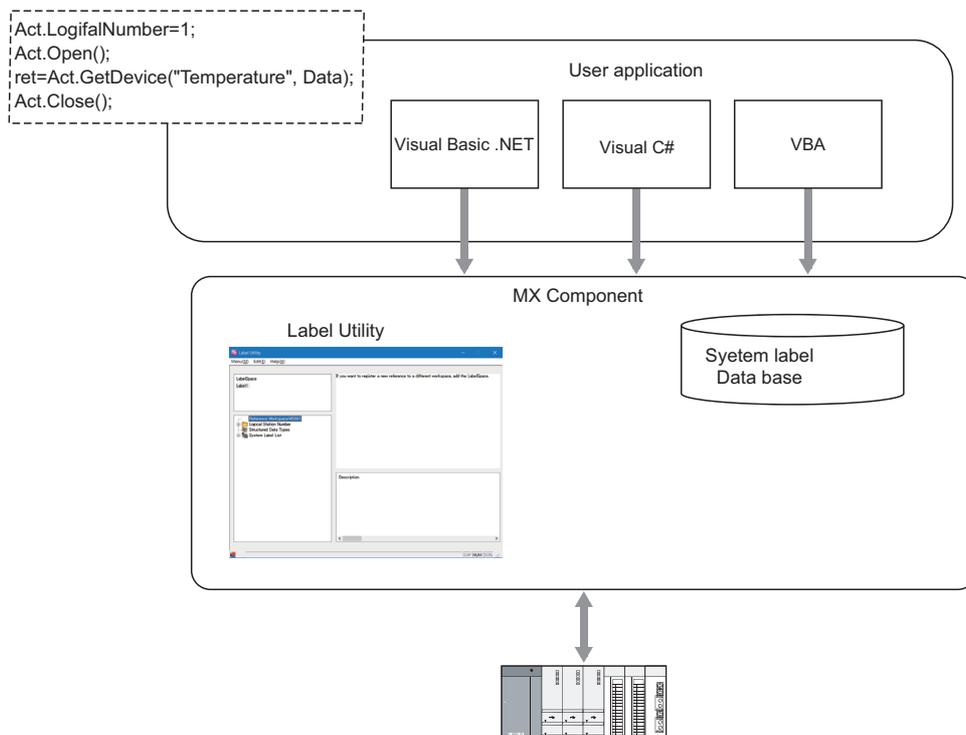
System labels are labels that can be used with sharing them with multiple projects by opening devices of programmable controller projects and motion controller projects as system labels.

Therefore, programming efficiency is improved.

As the device assignment settings are changed in bulk, device assignment changes on applications are not required.

Using system labels

Register a label using Label Utility of MX Component, and use it from controls.



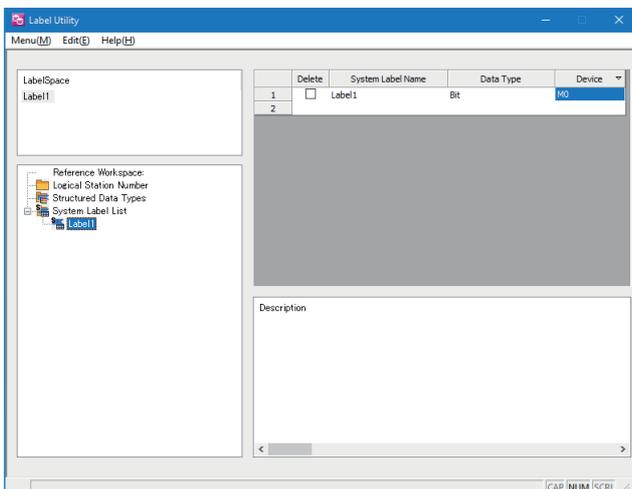
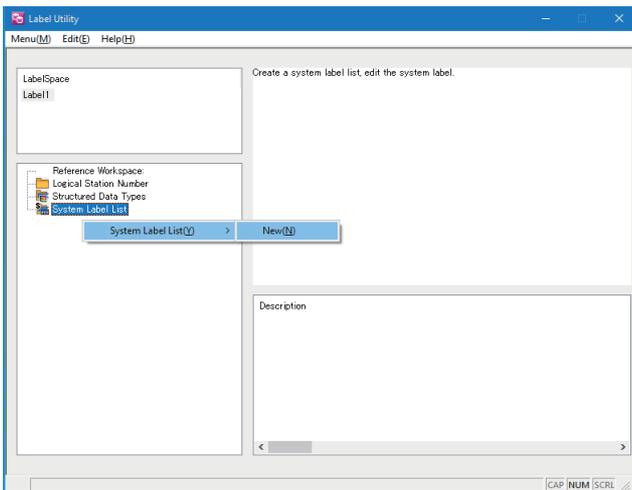
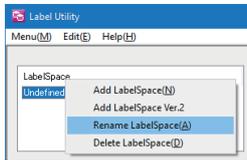
- System labels
 - Define labels on a 1:1 basis with devices, and register the labels.
 - Structure can be used. Structure consists of aggregate of various devices.
 - Set data types. Any data types can be set as array.
 - System labels used in MELSOFT Navigator can be utilized in MX Component.
- System label data base
 - A system label data base is a data base to manage system labels.
- System label Ver.2
 - The labels can be registered to system label Ver.2 using MX Component Version 4.07H or later.
 - Structure labels and array labels can also be registered.
 - System label structure of up to five-level can be used by MX Component Version 4.11M or later.

Registering system labels in MX Component

Register a system label with Label Utility.

■ Registering system labels

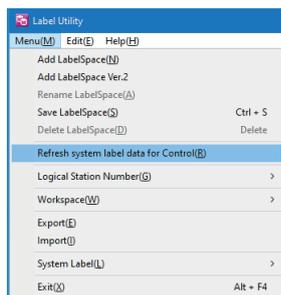
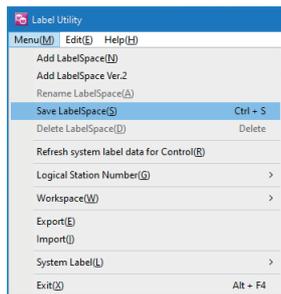
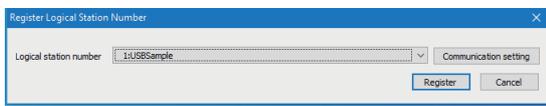
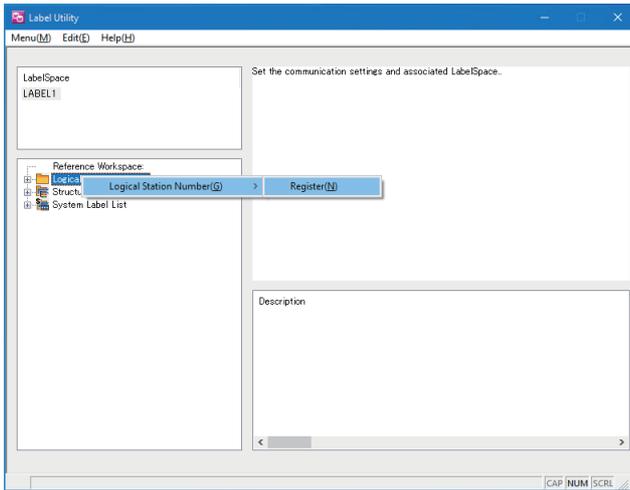
Operating procedure



1. Start Label Utility.
2. Right-click "Undefined_Name" on the LabelSpace window, and select [Rename LabelSpace]. Change the LabelSpace name.
3. Right-click "System Label List" on the Navigation window, and select [System Label List] ⇒ [New]. Set the system label list name.

Select the set system label list name to display the system label list.

4. Set system label names, data types, and devices.
([Page 118 System label list](#))



5. Right-click "Logical Station Number" on the Navigation window, and select [Logical Station Number] ⇒ [Register].

6. Select a logical station number, and click the [Register] button.

For setting the communication settings, click the [Communication setting] button, and set the settings in Communication Setting Wizard.

(☞ Page 56 Operations on Communication Setting Wizard screens)

The logical station number is displayed on the Navigation window.

7. Select [Menu] ⇒ [Save LabelSpace].

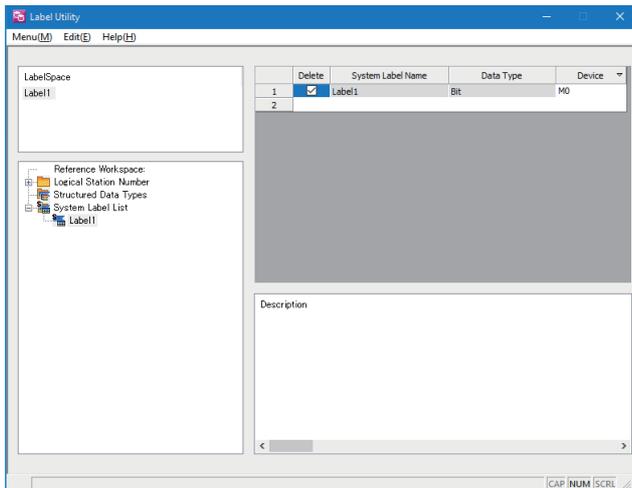
The system label is registered.

8. Select [Menu] ⇒ [Refresh system label data for Control].

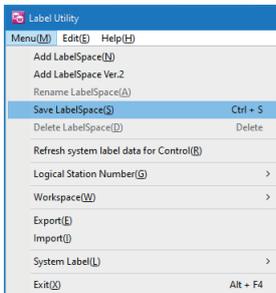
The labels referred from controls are updated.

■Deleting system labels

Operating procedure

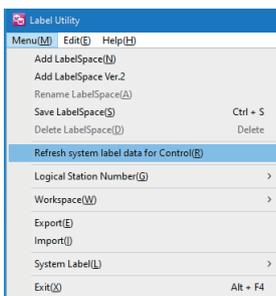


1. Select "Delete" for a system label to be deleted.



2. Select [Menu] ⇒ [Save LabelSpace].
The system label is deleted.

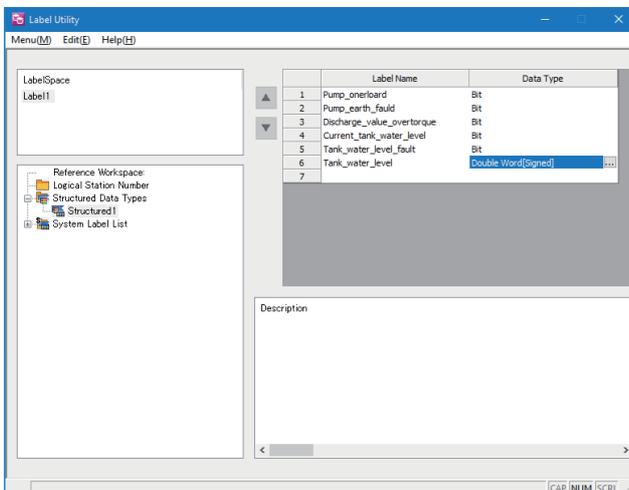
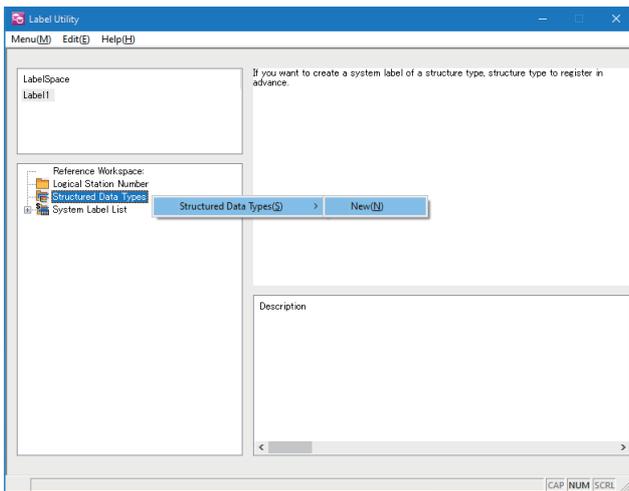
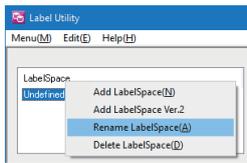
7



3. Select [Menu] ⇒ [Refresh system label data for Control].
The labels referred from controls are updated.

■ Registering system labels (structured data type)

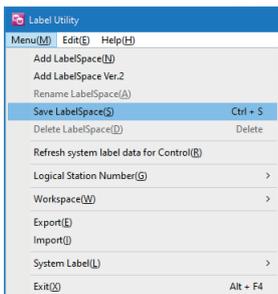
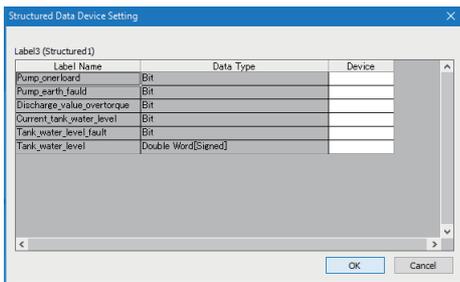
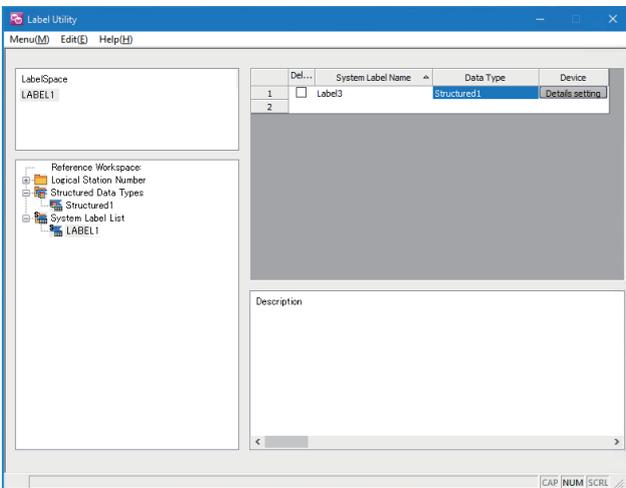
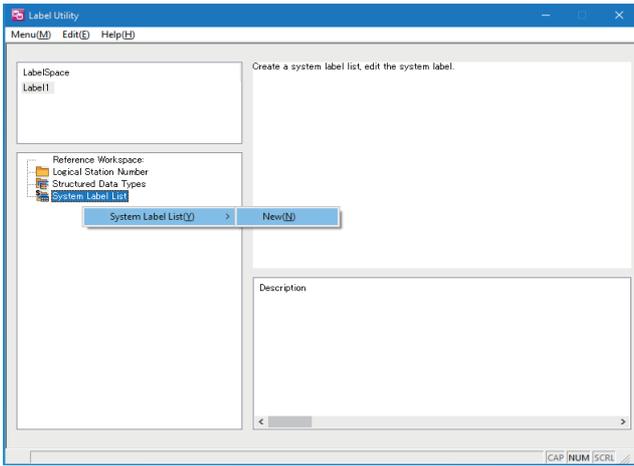
Operating procedure



1. Start Label Utility.
2. Right-click "Undefined_Name" on the LabelSpace window, and select [Rename LabelSpace]. Change the LabelSpace name.
3. Right-click "Structured Data Types" on the Navigation window, and select [Structured Data Types] ⇒ [New]. Set the structure name.

Select the set structure name to display the structure setting list.

4. Set label names and data types for structure member. (☞ Page 123 Structure setting)



5. Right-click "System Label List" on the Navigation window, and select [System Label List] ⇒ [New]. Set the system label list name.

Select the set system label list name to display the system label list.

6. Set system label names and data types.

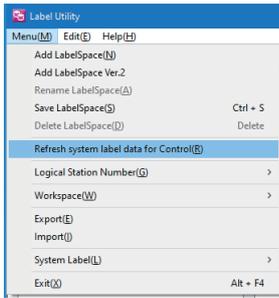
(Page 118 System label list)

Click "Detail Setting" on the "Device" column of the system label list.

7. Specify a device.

8. Select [Menu] ⇒ [Save LabelSpace].

The system label is registered.

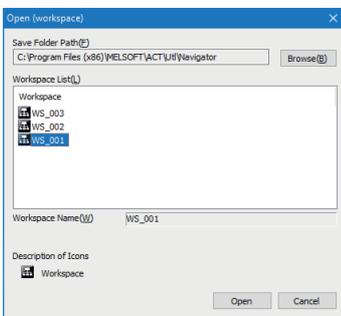
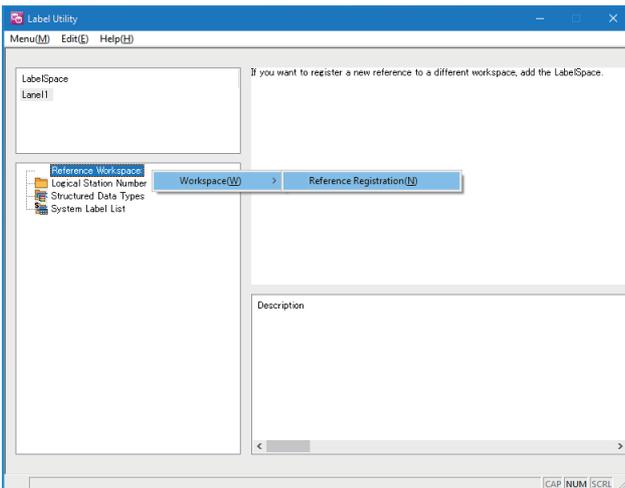
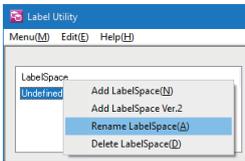


9. Select [Menu] ⇒ [Refresh system label data for Control].
The labels referred from controls are updated.

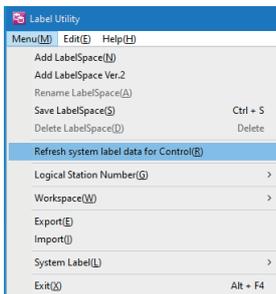
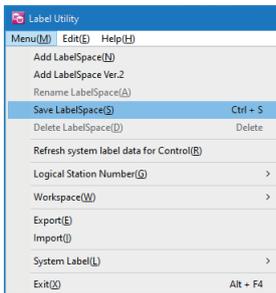
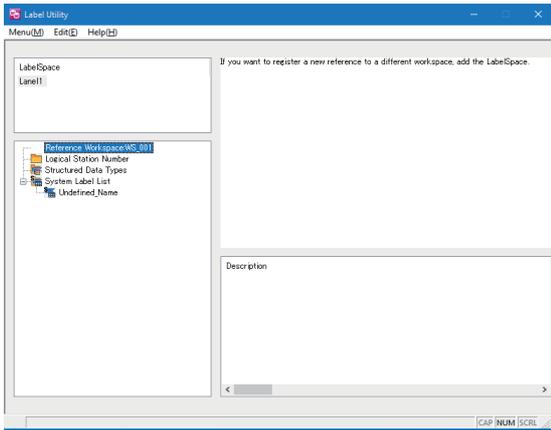
Utilizing labels used in MELSOFT Navigator

Refer a system label from existing Workspace, and register it with Label Utility.

Operating procedure



1. Start Label Utility.
2. Right-click "Undefined_Name" on the LabelSpace window, and select [Rename LabelSpace]. Change the LabelSpace name.
3. Right-click "Reference Workspace" on the Navigation window, and select [Workspace] ⇒ [Reference Registration].
4. Select a Workspace that is to be referred and registered. (☞ Page 124 Referring and registering/canceling Workspaces)



The referred and registered Workspace name is displayed on the Navigation window.

Select the system label name to display the referred system label list.

5. Select [Menu] ⇒ [Save LabelSpace].

The system label is registered.

For referring more than one Workspace, add LabelSpace first, and operate the procedure 2 to 4.

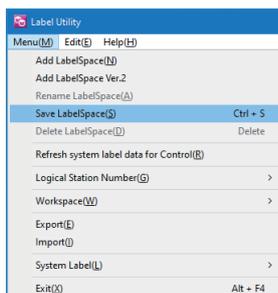
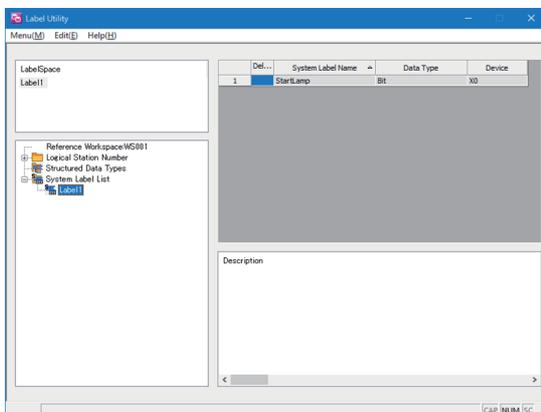
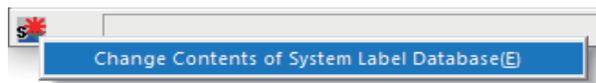
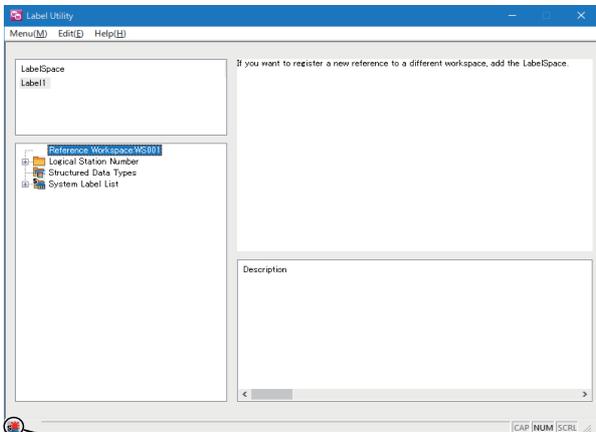
6. Select [Menu] ⇒ [Refresh system label data for Control].

The labels referred from controls are updated.

Applying device settings changed in GX Works2 to MX Component

Update the settings with Label Utility after changing the device assignment settings of referred and registered system labels with GX Works2.

Operating procedure



1. Start Label Utility.

The system label notification icon is displayed on the status bar of Label Utility.

2. Select the LabelSpace name.

3. Right-click the system label notification icon, and select [Change Contents of System Label Database].

(☞ Page 125 Change notification)

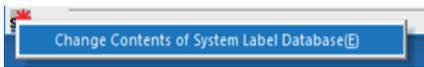
4. Click the [Yes] button in the window asking whether to reflect the changes.

Select the system label list name to display the system label list.

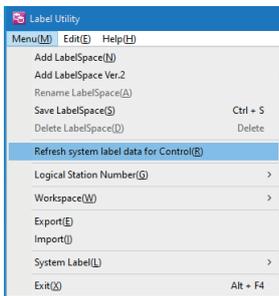
5. Select [Menu] ⇒ [Save LabelSpace].

The system label is registered.

For referring more than one Workspace, select LabelSpace from the LabelSpace window first, and operate the procedure 2 to 4.



The system label notification icon is displayed while referring more than one Workspace.



6. Select [Menu] ⇒ [Refresh system label data for Control].
The labels referred from controls are updated.



The system label notification icon is not displayed when the system labels are used without referring and registering the Workspace.

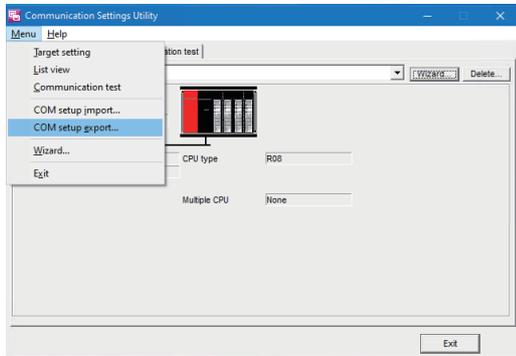
Using System Labels on another personal computer

Export a LabelSpace, and import it to another personal computer to use a system label.

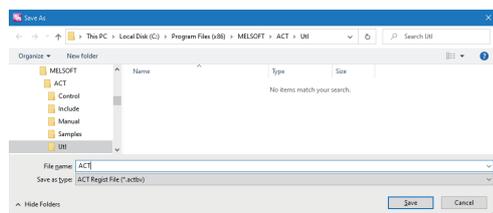
■Exporting LabelSpace

Export a LabelSpace, and create a file.

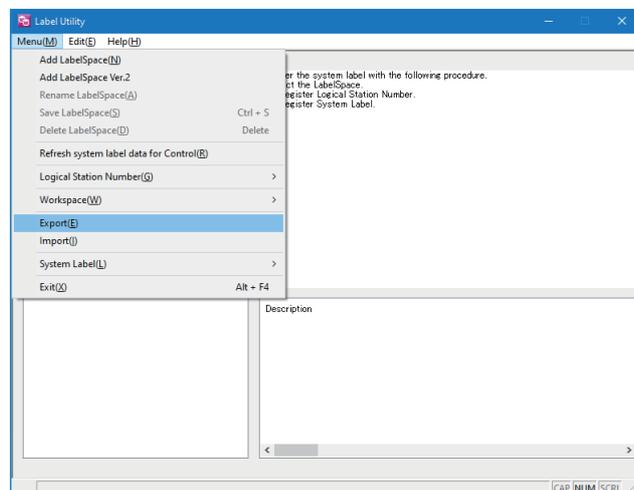
Operating procedure



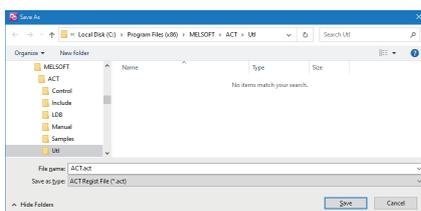
1. Start Communication Settings Utility.
2. Select [Menu] ⇒ [COM setup export].



3. Enter the file name to be saved.



4. Start Label Utility.
5. Select [Menu] ⇒ [Export].



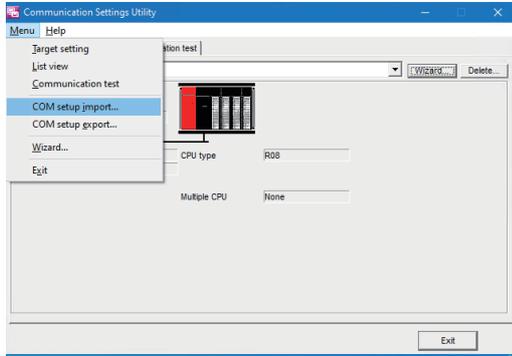
6. Enter the file name to be saved.

■Importing LabelSpace

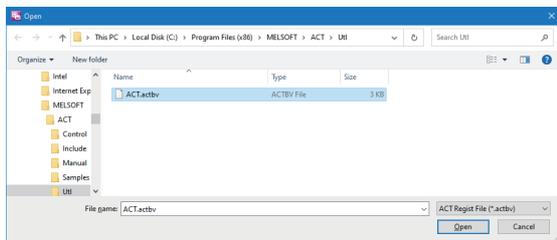
Import an exported file to LabelSpace of another personal computer.

Copy the exported files of the communication setting and the LabelSpace to the personal computer to be imported beforehand.

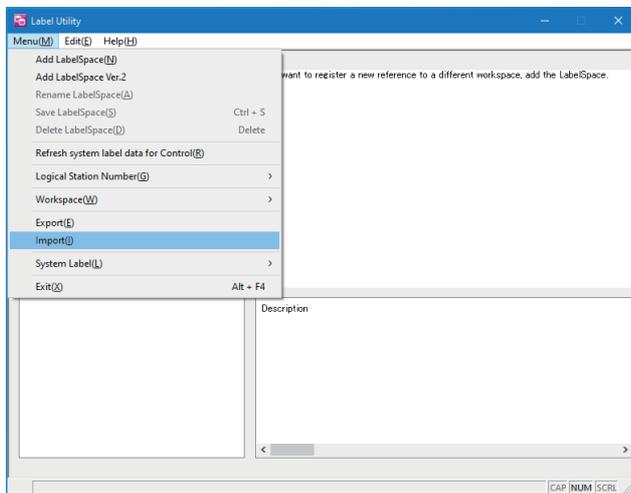
Operating procedure



1. Start Communication Settings Utility.
2. Select [Menu] ⇒ [COM setup import].

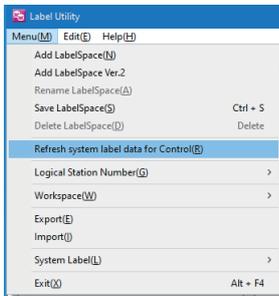
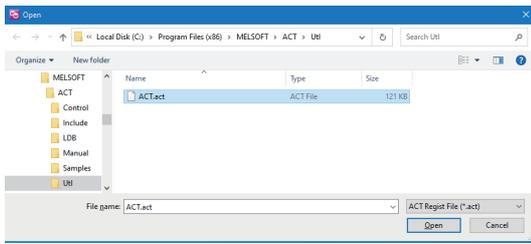


3. Specify the file name of communication setting to be imported.



4. Start Label Utility.
5. Select [Menu] ⇒ [Import].





6. Specify the file name of LabelSpace to be imported.

7. Select [Menu] ⇒ [Refresh system label data for Control].
The labels referred from controls are updated.

7.2 Coding

Create the processing for reading/writing the target devices, devices, and labels by using the properties and functions provided by a control.

For accessible devices, properties of controls, and functions, refer to the following:

☞ Page 154 Accessible Devices

☞ Page 170 PROPERTIES OF CONTROLS

☞ Page 416 FUNCTIONS

7.3 Debugging

Debugging can be performed efficiently by using PLC Monitor Utility when checking the device value. The following explains the operation and setting method of PLC Monitor Utility.

Checking a value using PLC Monitor Utility

The following explains how to operate and set PLC Monitor Utility.

Start

Operating procedure

Start PLC Monitor Utility from "MELSOFT" in Windows Start.

Transfer setting screen

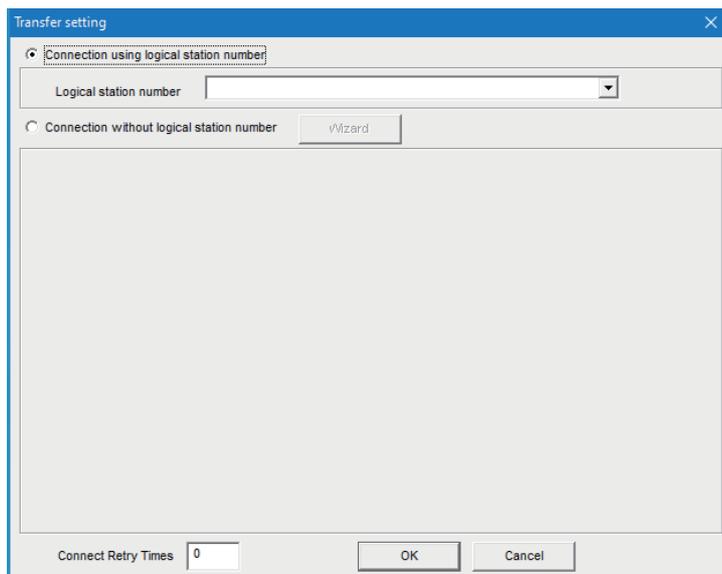
This screen is used to set connection from the personal computer to the programmable controller.

Operating procedure

Select [Online] ⇒ [Transfer setting].

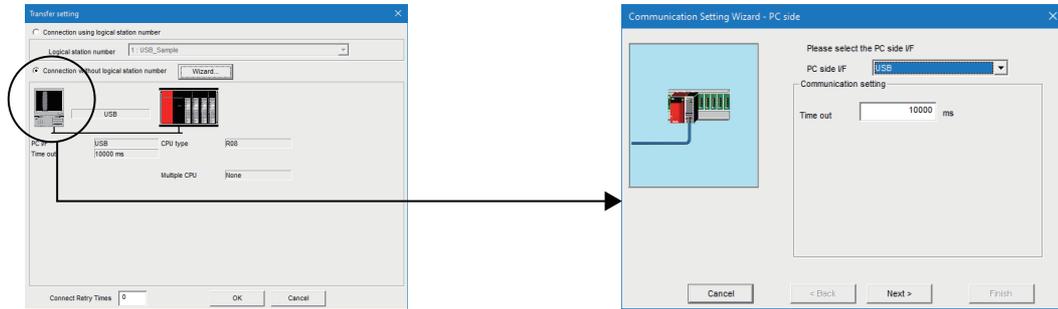
(This screen is also displayed when PLC Monitor Utility is started.)

Window



Item	Description
Connection using logical station number	Select this when using a logical station number.
Logical station number	Select the logical station number set in Communication Settings Utility.
Connection without logical station number	Select this when not using a logical station number.
[Wizard] button	Click this to start Communication Setting Wizard and set the transfer setting.
Connect Retry Times	Set the number of retries (0 to 9) to be performed when an error occurs during monitoring with PLC Monitor Utility.

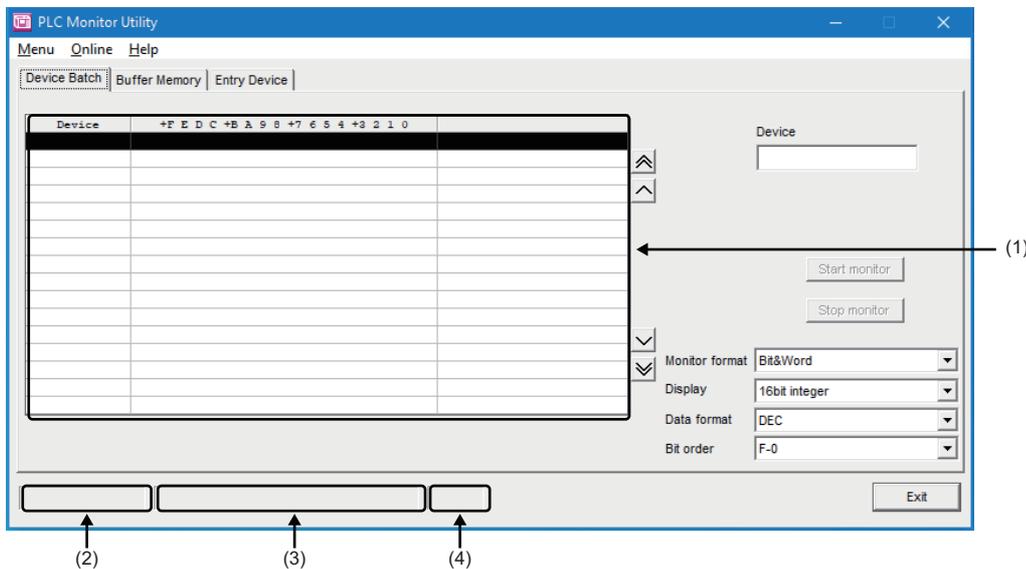
- Before specifying the logical station number, confirm that the settings of the logical station number, (the CPU type, station number, or the like) are correct in Communication Settings Utility.
- When the program setting type is selected, the details of the transfer setting can be changed by clicking the programmable controller or personal computer sketch.



Device Batch tab

This tab is used to monitor the specified device.

Window



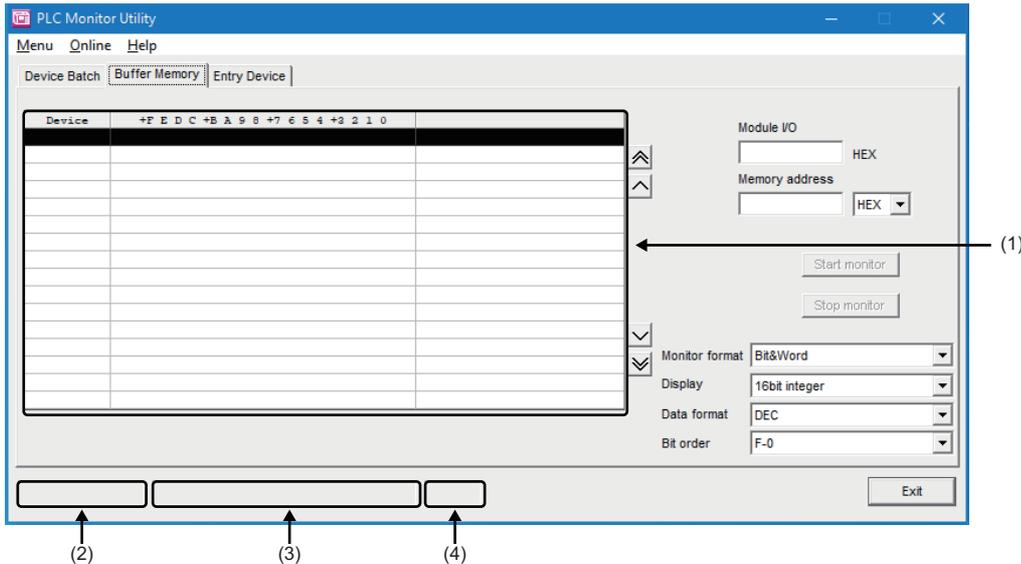
Item	Description
Device	Enter the device name to be monitored in batch. For the X and Y devices of FXCPUs and FX5CPUs, enter the device number in octal.
[Start monitor] button ([Stop monitor] button)	Click this to start (stop) monitor.
Monitor format	Set the monitor format. <ul style="list-style-type: none"> ■Item Bit&Word: Set the monitor screen to the bit and word display. Bit: Set the monitor screen to the bit display only. Word: Set the monitor screen to the word display only.
Display	Set the display format of the device values to be displayed when the monitor format is "Bit&Word" or "Word." <ul style="list-style-type: none"> ■Item 16bit integer: Set to the 16-bit integer display. 32bit integer: Set to the 32-bit integer display. Real number (single precision): Set to the real number (single precision) display. Real number (double precision): Set to the real number (double precision) display. ASCII character: Set to the ASCII character string display.
Data format	Set the radix when the display is "16 bit integer" or "32 bit integer." <ul style="list-style-type: none"> ■Item DEC: Set to the decimal display. HEX: Set to the hexadecimal display.
Bit order	Set the order of the bit devices being monitored. <ul style="list-style-type: none"> ■Item F-0: Display in order of F, E, ... 1, 0 from left to right. 0-F: Display in order of 0, 1, ... E, F from left to right.
(1) Monitor screen	Displays the device status. <ul style="list-style-type: none"> • For the bit device status, 1 indicates an ON status and 0 an OFF status. • Bit devices are monitored in units of 16 points. <ul style="list-style-type: none"> If any device outside the range supported by the CPU module is included in the 16 points, its value is displayed "0." • For the C devices of FXCPU, C0 to C199 (16 bit) and C200 and later (32 bit) are displayed separately. The "Write to Device" screen is displayed by clicking the device name. (Page 150 Write to Device screen) "***" flickers under the scroll button during monitoring.
(2) Target CPU name	Displays the communication target CPU name specified in Communication Setting Wizard.
(3) Communication route information	Displays such information as the network type, network number, start I/O address and station number.
(4) Logical station number	Displays the logical station number set for the utility setting type. This number is not displayed when the program setting type is used.

- Specifying the device memory in the Un\G□ format enables the buffer memory to be monitored.
- When monitoring the setting values of the timers and counters, indirectly specify the data registers.
- Devices cannot be monitored if the connection destination is not established.
- The transfer settings cannot be set during monitoring.
- Device data of QSCPU cannot be changed.

Buffer Memory tab

This tab is used to monitor the specified buffer memory.

Window



Item	Description
Module I/O	Enter the start address of the module to be monitored. For access to an FXCPU, enter the block number of the special expansion device into Module I/O.
Memory address	Enter the address of the buffer memory to be monitored in hexadecimal or decimal.
[Start monitor] button ([Stop monitor] button)	Click this to start (stop) monitor.
Monitor format	Set the monitor format. <ul style="list-style-type: none"> ■Item Bit&Word: Set the monitor screen to the bit and word display. Bit: Set the monitor screen to the bit display only. Word: Set the monitor screen to the word display only.
Display	Set the display format of the device values to be displayed when the monitor format is "Bit&Word" or "Word." <ul style="list-style-type: none"> ■Item 16bit integer: Set to the 16-bit integer display. 32bit integer: Set to the 32-bit integer display. Real number (single precision): Set to the real number (single precision) display. Real number (double precision): Set to the real number (double precision) display. ASCII character: Set to the ASCII character string display.
Data format	Set the radix when the display is "16 bit integer" or "32 bit integer." <ul style="list-style-type: none"> ■Item DEC: Set to the decimal display. HEX: Set to the hexadecimal display.
Bit order	Set the order of the bit devices being monitored. <ul style="list-style-type: none"> ■Item F-0: Display in order of F, E, ... 1, 0 from left to right. 0-F: Display in order of 0, 1, ... E, F from left to right.
(1) Monitor screen	Displays the buffer memory status. For the bit device status, 1 indicates an ON status and 0 an OFF status. "*" flickers under the scroll button during monitoring.
(2) Target CPU name	Displays the communication target CPU name specified in Communication Setting Wizard.

Item	Description
(3) Communication route information	Displays such information as the network type, network number, start I/O address and station number.
(4) Logical station number	Displays the logical station number set for the utility setting type. This number is not displayed when the program setting type is used.

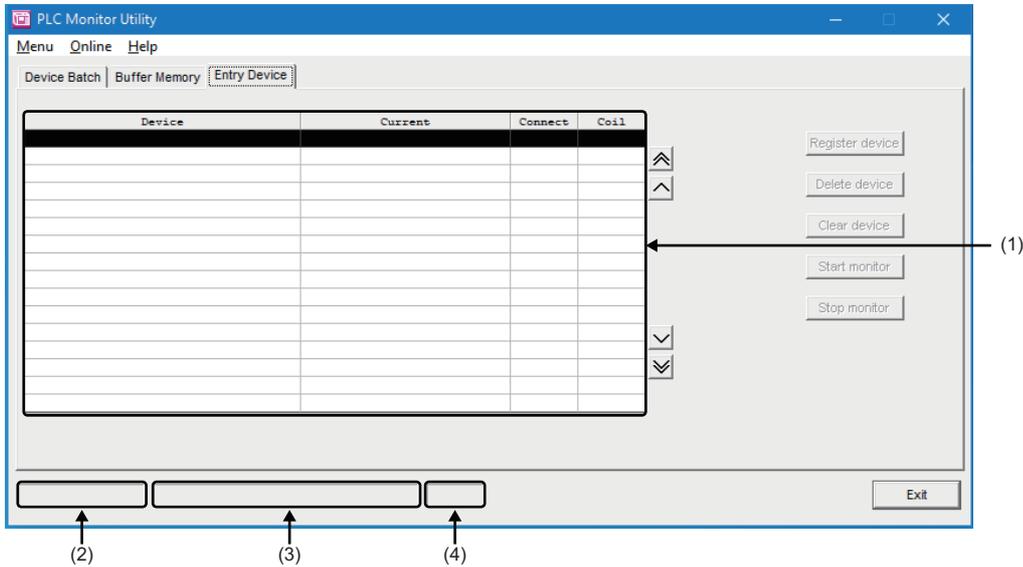
Point 

- Devices cannot be monitored if the connection destination is not established.
- The transfer settings cannot be set during monitoring.
- Monitoring for an FX5CPU are not supported.
- The devices cannot be monitored during GOT gateway device communication.

Entry Device tab

This tab is used to monitor the specified devices on a single screen at the same time.

Window



Item	Description
[Register device] button	Click this to register the device to be monitored. The following screen is displayed by clicking the [Register device] button. ☞ Page 149 The "Register device" screen
[Delete device] button	Click this to delete the device to be monitored.
[Clear device] button	Click this to delete all devices registered in device entry monitor from the monitor screen.
[Start monitor] button ([Stop monitor] button)	Click this to start (stop) monitor.
(1) Monitor screen	Displays the device status. The "Write to Device" screen is displayed by clicking the device name. (☞ Page 150 Write to Device screen) "" flickers under the scroll button during monitoring.
(2) Target CPU name	Displays the communication target CPU name specified in Communication Setting Wizard.
(3) Communication route information	Displays such information as the network type, network number, start I/O address and station number.
(4) Logical station number	Displays the logical station number set for the utility setting type. This number is not displayed when the program setting type is used.

■The "Register device" screen

The screen is displayed by clicking the [Register device] button.

Register the device to be monitored.

Window



Item	Description
Device	Enter the device to be registered.
Value	Set the value to be entered when a word device is specified. ■Item DEC: Set to decimal. HEX: Set to hexadecimal.
Display	Set the display format when a word device is specified. ■Item 16bit integer: Set to the 16-bit integer display. 32bit integer: Set to the 32-bit integer display. Real number (single precision): Set to the real number (single precision) display. Real number (double precision): Set to the real number (double precision) display. ASCII character: Set to the ASCII character string display.
[Register] button	Click this to register the device.
[Close] button	Click this to close the dialog box.

Point

- When monitoring the setting values of the timers and counters, indirectly specify the data registers.
- Devices cannot be monitored if the connection destination is not established.
- The transfer settings cannot be set during monitoring.

Write to Device screen

This screen is used to change the ON/OFF of a bit device or the present value of a word device or buffer memory.

Operating procedure

- Select [Online] ⇒ [Device write].
- Double-click the monitor screen of the corresponding tab.

Window

Item		Description
Bit device	Device	Enter the device name.
	[Force ON] button	Click this to forcibly change the specified device to the ON status.
	[Force OFF] button	Click this to forcibly change the specified device to the OFF status.
	[Toggle force] button	Click this to forcibly change the specified device from the ON to OFF status or from the OFF to ON status.
Word device/Buffer memory	Device	Select this to enter the word device to be written.
	Buffer memory	Select this to enter a module start I/O address and a buffer memory address.
	Setting value	Enter the value to be written. The following table indicates the input range. <ul style="list-style-type: none"> ■Item 16bit integer: -32768 to 32767 32bit integer: -2147483648 to 2147483647 Real number (single precision), Real number (double precision): -9999999999999999 to 9999999999999999 Number of significant figures: 13 digits
	[Set] button	Click this to write the set data.

Point

- When using an RnSF CPU, writing to the safety device cannot be performed in the safety mode. Page 158 When the access target is an RnSF CPU or RnPSF CPU (safety device)
- When using a QSCPU, [Device write] cannot be selected.
- When using a QSCPU, buffer memory cannot be written.

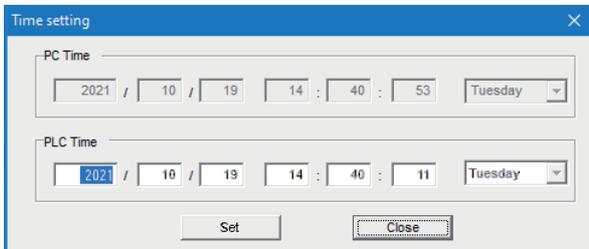
Time setting screen

This screen is used to read or change the clock data of the programmable controller.

Operating procedure

Select [Online] ⇒ [Set time].

Window



Item	Description
PC Time	Displays the time of the personal computer. (Write disabled)
PLC Time	Displays the time of the CPU module.
[Set] button	Click this to write the "PLC Time" information to the CPU module.
[Close] button	Click this to close the "Time setting" screen.

Point

Time setting is not available when either of the following communications is selected.

- GOT gateway device communication (An error occurs.)

For a QSCPU, the clock data cannot be changed.

PART 2

DETAILED SPECIFICATIONS OF PROGRAMS

This part explains the details of the properties and functions used in a program.

8 ACCESSIBLE DEVICES AND DEVICE TYPES

9 PROPERTIES OF CONTROLS

10 PROPERTY SETTINGS OF COMMUNICATION ROUTES

11 FUNCTIONS

12 SAMPLE PROGRAMS

13 ERROR CODES

8 ACCESSIBLE DEVICES AND DEVICE TYPES

This chapter explains the accessible devices in each communication route and device types that can be specified with the functions.

Point

The following devices and CPUs are not supported by MX Component. Therefore, do not specify them.

- Devices that are not in the list of accessible devices
- Connected station CPUs and relayed station CPUs that are not in the list of accessible devices (Page 181 PROPERTY SETTINGS OF COMMUNICATION ROUTES)

8.1 Accessible Devices

This section explains accessible devices for each access target.

The accessible devices do not depend on communication routes except for GX Simulator2 communication, GX Simulator3 communication, MT Simulator2 communication, and GOT gateway device communication. However, a device the communication route of which is not accessible cannot be accessed.

For the accessibility list using relayed network in each communication route, refer to the following:

Page 181 PROPERTY SETTINGS OF COMMUNICATION ROUTES

Accessible devices depending on communication routes

■For GX Simulator2 communication

The accessible devices during GX Simulator2 communication depend on the device supported by GX Simulator2.

For details, refer to the following:

GX Works2 Version 1 Operating Manual (Common)

■For GX Simulator3 communication

The accessible devices during GX Simulator3 communication depend on the device supported by GX Simulator3.

For details, refer to the following:

GX Works3 Operating Manual

■For MT Simulator2 communication

The accessible devices during MT Simulator2 communication depend on the device supported by MT Simulator2.

For the accessible device list of a Q motion CPU, refer to the following:

Page 157 When the access target is a motion CPU

■For GOT gateway device communication

Only the following device is accessible for GOT gateway device communication.

Device: Gateway device

Device name: EG

When the access target is a CPU module (other than motion CPU and MELSECWinCPU)

○: Accessible, ×: Inaccessible

Device (device name)		Access target								
		RCPU	RCCPU	LHCPU	FX5CPU	QCPU (Q mode)	QCCPU	LCPU	QSCPU ¹	FXCPU
Function input (FX)		×	×	×	×	○	×	○	×	×
Function output (FY)		×	×	×	×	○	×	○	×	×
Function register (FD)		×	×	×	×	○	×	○	×	×
Special relay (SM)		○	○	○	○	○	○	○	○	×
Special register (SD)		○	○	○	○	○	○	○	○	×
Input relay (X)		○	○	○	○	○	○	○	○	○
Output relay (Y)		○	○	○	○	○	○	○	○	○
Internal relay (M)		○	○	○	○	○	○	○	○	○
Latch relay (L)		○	×	○	○	○	×	○	×	×
Annunciator (F)		○	×	○	○	○	×	○	○	×
Edge relay (V)		○	×	○	×	○	×	○	○	×
Link relay (B)		○	○	○	○	○	○ ²	○	○	×
Data register (D)		○	○	○	○	○	○	○	○	○
Link register (W)		○	○	○	○	○	○ ²	○	○	×
Timer (T)	Contact (TS)	○	×	○	○	○	×	○	○	○
	Coil (TC)	○	×	○	○	○	×	○	○	○
	Present value (TN)	○	×	○	○	○	×	○	○	○
Counter (C)	Contact (CS)	○	×	○	○	○	×	○	○	○
	Coil (CC)	○	×	○	○	○	×	○	○	○
	Present value (CN)	○	×	○	○	○	×	○	○	○
Retentive timer (ST)	Contact (STS/SS)	○	×	○	○	○	×	○	○	×
	Coil (STC/SC)	○	×	○	○	○	×	○	○	×
	Present value (STN/SN)	○	×	○	○	○	×	○	○	×
Long timer (LT)	Contact (LTS)	○	×	○	×	×	×	×	×	×
	Coil (LTC)	○	×	○	×	×	×	×	×	×
	Present value (LTN)	○	×	○	×	×	×	×	×	×
Long counter (LC)	Contact (LCS)	○	×	○	○	×	×	×	×	×
	Coil (LCC)	○	×	○	○	×	×	×	×	×
	Present value (LCN)	○	×	○	○	×	×	×	×	×
Long retentive timer (LST)	Contact (LSTS/LSS)	○	×	○	×	×	×	×	×	×
	Coil (LSTC/LSC)	○	×	○	×	×	×	×	×	×
	Present value (LSTN/LSN)	○	×	○	×	×	×	×	×	×
Link special relay (SB)		○	×	○	○	○	×	○	○	×
Link special register (SW)		○	×	○	○	○	×	○	○	×
Step relay (S)		×	×	×	○	×	×	×	×	○
Direct input (DX)		×	×	×	×	×	×	×	×	×
Direct output (DY)		×	×	×	×	×	×	×	×	×
Accumulator (A)		×	×	×	×	×	×	×	×	×
Index register	(Z)	○	×	○	○	○	×	○	×	○ ³
	(V)	×	×	×	×	×	×	×	×	○ ³
Long index register (LZ)		○	×	○	○	×	×	×	×	×
File register	(R)	○	×	○	○	○ ⁴	×	○	×	○ ⁵
	(ZR)	○	○	○	×	○ ⁴	×	○	×	×
Refresh device for modules (RD)		○	×	○	×	×	×	×	×	×
Extended file register (ERn\R)		×	×	×	×	×	×	×	×	×

Device (device name)		Access target								
		RCPU	RCCPU	LHCPU	FX5CPU	QCPU (Q mode)	QCCPU	LCPU	QSCPU [*] ₁	FXCPU
Direct link	Link input (Jn\X)	○	○	×	×	○	○	○	×	×
	Link output (Jn\Y)	○	○	×	×	○	○	○	×	×
	Link relay (Jn\B)	○	○	×	×	○	○	○	×	×
	Link special relay (Jn\SB)	○	○	×	×	○	○	○	×	×
	Link register (Jn\W)	○	○	×	×	○	○	○	×	×
	Link special register (Jn\SW)	○	○	×	×	○	○	○	×	×
Module access device (Un\G)		○	○	○	○	○ ^{*6}	○	○	×	○

*1 Writing to device data cannot be performed.

*2 Q12DCCPU-V (Basic mode) cannot be accessed.

*3 Data cannot be written to 2 or more consecutive points using WriteDeviceBlock or WriteDeviceBlock2. (Data may be written to only one point.)

*4 Q00UJCPU cannot be accessed.

*5 When specifying a file register in FX series CPU other than FX3G(C)CPU and FX3U(C)CPU, specify the data register (D). The extended register (R) can be specified only in FX3G(C)CPU or FX3U(C)CPU.

*6 In a multi-CPU configuration, reading from the shared memory of the host CPU cannot be performed. Writing to the shared memory cannot be performed regardless of the host or other CPU.

When the access target is a motion CPU

○: Accessible, ×: Inaccessible

Device (device name)		Access target		
		R16MT/R32MT/R64MT	Q172D/Q173D	Q172DS/Q173DS
Input relay (X)		○	○	○
Output relay (Y)		○	○	○
Internal relay (M)		○	○	○
Latch relay (L)		×	×	×
Annunciator (F)		○	○	○
Link relay (B)		○	○	○
Data register (D)		○	○	○
Link register (W)		○	○	○
Special register M (SPM)		×	×	×
Special register D (SPD)		×	×	×
Motion register (#)		○	○	○
Coasting timer (FT)		×	×	×
Special register (SD)		○	○	○
Special relay (SM)		○	○	○
CPU buffer memory access device	Multiple CPU area device (U3En\G)	×	○ ^{*1}	○ ^{*1}
	CPU buffer memory (U3En\G)	○	×	×
	CPU buffer memory fixed scan communication area (U3En\HG)	○	×	×
Module access device (Un\G)		○	×	×

*1 In a multi-CPU configuration, reading from the shared memory of the host CPU cannot be performed.
Writing to the shared memory cannot be performed regardless of the host or other CPU.

When the access target is a MELSECWinCPU

○: Accessible

Device (device name)		Access target
		MELSECWinCPU
Input relay (X)		○
Output relay (Y)		○
Internal relay (M)		○
Data register (D)		○
Link relay (B)		○
Link register (W)		○
Special register (SD)		○
Special relay (SM)		○
Special register (SD)		○
Special relay (SM)		○
Direct link	Link input (Jn\X)	○
	Link output (Jn\Y)	○
	Link relay (Jn\B)	○
	Link special relay (Jn\SB)	○
	Link register (Jn\W)	○
	Link special register (Jn\SW)	○
CPU buffer memory access device	CPU buffer memory (U3En\G)	○
	CPU buffer memory fixed scan communication area (U3En\HG)	○
Module access device (Un\G)		○

When the access target is an RnSF CPU or RnPSF CPU (safety device)

The safety devices described in the following table can be used when access target is an RnSF CPU and RnPSF CPU. The safety mode is read-only. An attempt to write into the safety device will cause an error.

○: Accessible

Device (device name)	Access target
	RnSF CPU, RnPSF CPU
Safety input (SA\X)	○
Safety output (SA\Y)	○
Safety internal relay (SA\M)	○
Safety link relay (SA\B)	○
Safety timer (SA\T)	○
Safety retentive timer (SA\ST)	○
Safety counter (SA\C)	○
Safety data register (SA\D)	○
Safety link register (SA\W)	○
Safety special relay (SA\SM)	○
Safety special register (SA\SD)	○

When the access target is an own board for CC-Link communication

The following devices are usable only for own board access.

Device	Device name	Remarks
Link special relay (for CC-Link)	SB	Link special relay of own board
Link special register (for CC-Link)	SW	Link special register of own board
Remote input	X	RX
Remote output	Y	RY
Remote register (Data write area for CC-Link)	WW	RWw
Remote register (Data read area for CC-Link)	WR	RWr
Buffer memory	ML	Buffer memory of own station CC-Link module
Random access buffer	MC	Random access buffer in buffer memory of own station CC-Link module

When the access target is an own board for MELSECNET/H communication

The following devices are usable only for own board access.

Device	Device name	Remarks
Link relay	B	Link relay of own board
Link register	W	Link register of own board
Link special relay	SB	Link special relay of own board
Link special register	SW	Link special register of own board
Link input	X	LX
Link output	Y	LY

When the access target is an own board for CC-Link IE Controller Network communication

The following devices are usable only for own board access.

Device	Device name	Remarks
Link relay	B	Link relay of own board
Link register	W	Link register of own board
Link special relay	SB	Link special relay of own board
Link special register	SW	Link special register of own board
Link input	X	LX
Link output	Y	LY
Buffer memory	ML	Buffer memory of own board

When the access target is an own board for CC-Link IE Field Network communication

The following devices are usable only for own board access.

Device	Device name	Remarks
Remote register	W	W0000 to W1FFF = RWw0 to RWw1FFF Remote register of own board for sending W2000 to W3FFF = RWr0 to RWr1FFF Remote register of own board for receiving
Link special relay	SB	Link special relay of own board
Link special register	SW	Link special register of own board
Remote input	X	RX
Remote output	Y	RY
Buffer memory	ML	Buffer memory of own board

When the access target is an own board for CC-Link IE TSN communication

The following devices are usable only for own board access.

Device	Device name	Remarks
Remote input	X	RX
Remote output	Y	RY
Link relay	B	Link relay of own board
Link register	W	Link register of own board
Remote register	WW	Remote register of own board (for sending)
Remote register	WR	Remote register of own board (for receiving)
Link special relay	SB	Link special relay of own board
Link special register	SW	Link special register of own board

When the access target is an inverter

The following table shows the accessible monitor types during inverter communication.

Monitor type (Decimal)	Description
1	Output frequency/speed
2	Output current
3	Output voltage
5	Frequency setting value/speed setting
6	Running speed
7	Motor torque
8	Converter output voltage
9	Regenerative brake duty
10	Electronic thermal O/L relay load factor
11	Output current peak value
12	Converter output voltage peak value
13	Input power
14	Output power
17	Load meter
18	Motor excitation current
19	Position pulse
20	Cumulative energization time
22	Orientation status
23	Actual operation time
24	Motor load factor
25	Cumulative power
32	Torque command
33	Torque current command
34	Motor output
35	Feedback pulse
40	PLC function user monitor 1
41	PLC function user monitor 2
42	PLC function user monitor 3
50	Energy saving effect
51	Cumulative energy saving
52	PID set point
53	PID measured value
61	Motor thermal load factor
62	Inverter thermal load factor
64	PTC thermistor resistance
67	PID measured value 2
87	Remote output value 1
88	Remote output value 2
89	Remote output value 3
90	Remote output value 4
91	PID manipulated variable
92	Second PID set point
93	Second PID measured value
94	Second PID deviation
95	Second PID measured value 2
96	Second PID manipulated variable
97	Dancer main speed setting

When the access target is a robot controller

The following table indicates the accessible monitor type during robot controller communication.

Specify the monitor type with the following format: (Request ID).(Data type).(Argument).

Monitor type			Acquired value
Request ID	Data type	Argument	
231	1	(1) Input signal read start number	Input signal information (16 points)
	2	(1) Output signal read start number	Output signal information (16 points)
237	1	(1) Register number	Input register description
238	1	(1) Register number	Output register description

8.2 Device Types

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

Point

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

Common device types

The following device types are common to all communication routes except for the communication to an FX5CPU, GOT gateway device communication, inverter communication, and robot controller communication.

Device	Device name	Device type	Device number	
Function input	FX	Bit device	Decimal	
Function output	FY	Bit device	Decimal	
Function register	FD	Word device ^{*1}	Decimal	
Special relay	SM	Bit device	Decimal	
Special register	SD	Word device	Decimal	
Input relay	X	Bit device	Hexadecimal ^{*2}	
Output relay	Y	Bit device	Hexadecimal ^{*2}	
Internal relay	M	Bit device	Decimal	
Latch relay	L	Bit device	Decimal	
Annunciator	F	Bit device	Decimal	
Edge relay	V	Bit device	Decimal	
Link relay	B	Bit device	Hexadecimal	
Data register	D	Word device	Decimal	
Link register	W	Word device	Hexadecimal	
Timer ^{*11}	Contact	TS	Bit device	Decimal
	Coil	TC	Bit device	Decimal
	Present value	TN	Word device	Decimal
Counter ^{*11}	Contact	CS	Bit device	Decimal
	Coil	CC	Bit device	Decimal
	Present value ^{*3}	CN	Word device	Decimal
Retentive timer ^{*11}	Contact	STS/SS	Bit device	Decimal
	Coil	STC/SC	Bit device	Decimal
	Present value	STN/SN	Word device	Decimal
Link special relay	SB	Bit device	Hexadecimal	
Link special register	SW	Word device	Hexadecimal	
Accumulator	A ^{*4}	Word device	Decimal	
Index register	Z ^{*4}	Word device	Decimal	
	V ^{*4}	Word device	Decimal	
File register	R ^{*5}	Word device	Decimal	
	ZR	Word device	Decimal	

Device		Device name	Device type	Device number
Direct link ^{*7*12}	Link input	Jn\X ^{*6}	Bit device	Hexadecimal
	Link output	Jn\Y ^{*6}	Bit device	Hexadecimal
	Link relay	Jn\B ^{*6}	Bit device	Hexadecimal
	Link special relay	Jn\SB ^{*6}	Bit device	Hexadecimal
	Link register	Jn\W ^{*6}	Word device	Hexadecimal
	Link special register	Jn\SW ^{*6}	Word device	Hexadecimal
Special direct buffer memory ^{*8*9}		Un\G□ ^{*6*10}	Word device	Hexadecimal/Decimal

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

*2 For FXCPU, the device number is octal.

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

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

*5 When specifying a file register in FX series CPU other than FX3G(C)CPU and FX3U(C)CPU, specify the data register (D). The extended register (R) can be specified only in FX3G(C)CPU or FX3U(C)CPU.

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

*7 Jn specifies a network number.

*8 Un specifies a special module I/O number (hexadecimal), and G□ specifies a buffer memory address (decimal).

(Example: When the special module I/O number is 200H and the buffer memory address is 100, the device name will be "U20\G100.")

*9 In a QCPU multiple CPU configuration, an error occurs if the shared memory of the host QCPU is specified.

Regardless of whether the CPU is a host CPU or other CPU, an error occurs if data is written to the shared memory.

*10 For FXCPU, this device name can be used on FX3U(C)CPU only.

*11 When accessing the device using a label of system label Ver.2, the label of system label Ver.2 which is corresponding to a device for contact/coil/present value is required to be defined.

*12 The device is not available in LHCPU.

When the access target is an RCPU or LHCPU

The devices described in the following table can be used when the access target is an RCPU or LHCPU.

Device		Device name	Device type	Device number
Long timer* ¹	Contact	LTS	Bit device	Decimal
	Coil	LTC	Bit device	Decimal
	Present value	LTN	Double word device	Decimal
Long counter* ¹	Contact	LCS	Bit device	Decimal
	Coil	LCC	Bit device	Decimal
	Present value	LCN	Double word device	Decimal
Retentive long timer* ¹	Contact	LSTS/LSS	Bit device	Decimal
	Coil	LSTC/LSC	Bit device	Decimal
	Present value	LSTN/LSN	Double word device	Decimal
Long index register		LZ	Double word device	Decimal
Refresh device for modules		RD	Word device	Decimal

*1 When accessing the device using a label of system label Ver.2, the label of system label Ver.2 which is corresponding to a device for contact/coil/present value is required to be defined.

When the access target is an RnSF CPU or RnPSF CPU (safety device)

The safety devices described in the following table can be used when the access target is an RnSF CPU or RnPSF CPU.

The safety mode is read-only. An attempt to write into the safety device will cause an error.

Device		Device name	Device type	Device number
Safety input		SAIX	Bit device	Hexadecimal
Safety output		SAIY	Bit device	Hexadecimal
Safety internal relay		SAIM	Bit device	Decimal
Safety link relay		SAIB	Bit device	Hexadecimal
Safety timer		SAIT	Bit device/Word device	Decimal
Safety retentive timer		SAIST	Bit device/Word device	Decimal
Safety counter		SAIC	Bit device/Word device	Decimal
Safety data register		SAID	Word device	Decimal
Safety link register		SAIW	Word device	Hexadecimal
Safety special relay		SAISM	Bit device	Decimal
Safety special register		SAISD	Word device	Decimal

When the access target is an R motion CPU or Q motion CPU

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

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

When the access target is an FX5CPU

The devices described in the following table can be used when the access target is an FX5CPU.

Device	Device name	Device type	Device number
Special relay	SM	Bit device	Decimal
Special register	SD	Word device	Decimal
Input relay	X	Bit device	Octal
Output relay	Y	Bit device	Octal
Internal relay	M	Bit device	Decimal
Latch relay	L	Bit device	Decimal
Annunciator	F	Bit device	Decimal
Link relay	B	Bit device	Hexadecimal
Data register	D	Word device	Decimal
Link register	W	Word device	Hexadecimal
Timer	Contact	TS	Bit device
	Coil	TC	Bit device
	Present value	TN	Word device
Counter	Contact	CS	Bit device
	Coil	CC	Bit device
	Present value	CN	Word device
Retentive timer	Contact	STS/SS	Bit device
	Coil	STC/SC	Bit device
	Present value	STN/SN	Word device
Long counter	Contact	LCS	Bit device
	Coil	LCC	Bit device
	Present value	LCN	Bit device
Link special relay	SB	Bit device	Hexadecimal
Link special register	SW	Word device	Hexadecimal
Step relay	S	Bit device	Decimal
Index register	Z	Word device	Decimal
File register	R	Word device	Decimal
Long index register	LZ	Double word device	Decimal
Module access device	Un\G	Word device	Decimal

For CC-Link communication

The devices described in the following table can be used when accessing to the own board with CC-Link communication. These devices cannot be used for other communication routes.

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

For MELSECNET/H communication

The devices described in the following table can be used when accessing to the own board with MELSECNET/H communication.

These devices cannot be used for other communication routes.

Device	Device name	Device type	Device number	Remarks
Link relay	B	Bit device	Hexadecimal	Link relay of own board
Link register	W	Word device	Hexadecimal	Link register of own board
Link special relay	SB	Bit device	Hexadecimal	Link special relay of own board
Link special register	SW	Word device	Hexadecimal	Link special register of own board
Link input	X	Bit device	Hexadecimal	LX
Link output	Y	Bit device	Hexadecimal	LY

For CC-Link IE Controller Network communication

The devices described in the following table can be used when accessing to the own board with CC-Link IE Controller Network communication.

These devices cannot be used for other communication routes.

Device	Device name	Device type	Device number	Remarks
Link relay	B	Bit device	Hexadecimal	Link relay of own board
Link register	W	Word device	Hexadecimal	Link register of own board
Link special relay	SB	Bit device	Hexadecimal	Link special relay of own board
Link special register	SW	Word device	Hexadecimal	Link special register of own board
Link input	X	Bit device	Hexadecimal	LX
Link output	Y	Bit device	Hexadecimal	LY
Buffer memory	ML	Word device	Hexadecimal	Buffer memory of own board

For CC-Link IE Field Network communication

The devices described in the following table can be used when accessing to the own board with CC-Link IE Field Network communication.

These devices cannot be used for other communication routes.

Device	Device name	Device type	Device number	Remarks
Remote register	W	Word device	Hexadecimal	W0000 to W1FFF = RWw0 to RWw1FFF Remote register of own board (for sending) W2000 to W3FFF = RWr0 to RWr1FFF Remote register of own board (for receiving)
Link special relay	SB	Bit device	Hexadecimal	Link special relay of own board
Link special register	SW	Word device	Hexadecimal	Link special register of own board
Remote input	X	Bit device	Hexadecimal	RX
Remote output	Y	Bit device	Hexadecimal	RY
Buffer memory	ML	Word device	Hexadecimal	Buffer memory of own board

For CC-Link IE TSN communication

The devices described in the following table can be used when accessing to the own board with CC-Link IE TSN communication.

These devices cannot be used for other communication routes.

Device	Device name	Device type	Device number	Remarks
Remote input	X	Bit device	Hexadecimal	RX
Remote output	Y	Bit device	Hexadecimal	RY
Link relay	B	Bit device	Hexadecimal	Link relay of own board
Link register	W	Word device	Hexadecimal	Link register of own board
Remote register	WW	Word device	Hexadecimal	Remote register of own board (for sending)
Remote register	WR	Word device	Hexadecimal	Remote register of own board (for receiving)
Link special relay	SB	Bit device	Hexadecimal	Link special relay of own board
Link special register	SW	Word device	Hexadecimal	Link special register of own board
Buffer memory	ML	Word device	Hexadecimal	Buffer memory of own board

For GOT gateway device communication

The following list shows how to specify the device name used for GOT gateway device communication

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

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

For inverter communication/robot controller communication

For the monitor types used for inverter communication/robot controller communication, refer to the following:

☞ Page 161 When the access target is an inverter

☞ Page 162 When the access target is a robot controller

Device extension representations

The following table shows applicability of device extension representation.

These representations cannot be used for ReadDeviceBlock and WriteDeviceBlock.

○: Applicable ×: Not applicable

Device extension representation	Target CPU				
	RCPU	RCCPU	R motion CPU	LHCPU	FX5CPU
Digit specification (Example: K4M0)	○ ^{*1}	○ ^{*6}	×	○ ^{*1}	○
Bit specification (Example: D0.1)	○ ^{*2}	○	○	○ ^{*2}	○
Index setting (Example: M100Z0)	○ ^{*3}	×	×	○ ^{*3}	○

Device extension representation	Target CPU						GOT	Inverter	Robot controller
	QCPU (Q mode)	QCCPU	LCPU	QSCPU	FXCPU	Q motion CPU			
Digit specification (Example: K4M0) ^{*4}	○	○ ^{*6}	○	○	○	×	×	×	×
Bit specification (Example: D0.1)	○ ^{*7}	○	○ ^{*7}	○ ^{*7}	○ ^{*7}	×	○	×	×
Index setting (Example: M100Z0) ^{*5}	○	×	○	×	×	×	×	×	×

*1 FX/FY, FD/SD, V, T/C/ST, LT/LC/LST, W/SW, G, Z, R/ZR and LZ cannot be specified.

*2 Z, T/C/ST, LT/LC/LST and LZ cannot be specified.

*3 FX/F, Z and LZ cannot be specified.

*4 FX/FY, DX/DY, and T/C/ST (contact, coil) cannot be specified.

*5 FX/FY, DX/DY, T/C/ST (contact, coil), Z, and S cannot be specified.

*6 The bit devices whose device numbers are multiple of 16 can only be used for digit specification. (For link direct devices, the digit specification only for K4 and K8 is supported.)

*7 Z, V, and T/C/ST (present value) cannot be specified.

9 PROPERTIES OF CONTROLS

This chapter explains the property list of controls and details of the respective controls.

9.1 Property List

The following table shows the properties of each control.

Control name	Property name
ACT Control	ActLogicalStationNumber
	ActPassword
	ActBaudRate
	ActConnectUnitNumber
	ActControl
	ActCpuTimeOut
	ActCpuType
	ActDataBits
	ActDestinationIONumber
	ActDestinationPortNumber
	ActDidPropertyBit
	ActDsidPropertyBit
	ActHostAddress
	ActIntelligentPreferenceBit
	ActIONumber
	ActMultiDropChannelNumber
	ActMxUnitSeries ^{*3}
	ActNetworkNumber
	ActPacketType
	ActParity
	ActPassword
	ActPortNumber
	ActProtocolType
	ActSourceNetworkNumber
	ActSourceStationNumber
	ActStationNumber
	ActStopBits
	ActSumCheck
	ActTargetSimulator
	ActThroughNetworkType
	ActTimeOut
	ActUnitNumber
	ActUnitType

Control name		Property name
ACT Control	ActProgType	ActATCommand ^{*4}
	ActProgType64	ActATCommandPasswordCancelRetryTimes ^{*4}
	ActProgDataLogging ^{*1*2}	ActATCommandResponseWaitTime ^{*4}
	ActProgDataLogging64 ^{*1*2}	ActCallbackCancelWaitTime ^{*4}
		ActCallbackDelayTime ^{*4}
		ActCallbackNumber ^{*4}
		ActCallbackReceptionWaitingTimeOut ^{*4}
		ActConnectionCDWaitTime ^{*4}
		ActConnectionModemReportWaitTime ^{*4}
		ActConnectWay ^{*4}
		ActDialNumber ^{*4}
		ActDisconnectionCDWaitTime ^{*4}
		ActDisconnectionDelayTime ^{*4}
		ActLineType ^{*4}
		ActOutsideLineNumber ^{*4}
		ActPasswordCancelResponseWaitTime ^{*4}
		ActTransmissionDelayTime ^{*4}
	ActSupportMsg	—
	ActSupportMsg64	—
.NET Control	DotUtilType	ActLogicalStationNumber
	DotUtilType64	ActPassword
	DotSupportMsg	—
	DotSupportMsg64	—

*1 Only supported by RCPUs.

*2 GX Simulator3 is not supported.

*3 The property value is set automatically, and it cannot be changed.

*4 Properties for modem communication. They can be used in only MX Component Version 4, and cannot be used in MX Component Version 5 or later.

9.2 Details of Control Properties

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

Properties of utility setting type controls

The following table shows the properties of the ActUtilType control and DotUtilType control.

Property name (Type)	Description	Default value
ActLogicalStationNumber(LONG)	A logical station number set in Communication Settings Utility. (Applicable setting range: 0 to 1023)	0 (0x00)
ActPassword(BSTR)	Specify a password to disable the password set to the password protected modules.*1 This setting is ignored when a password protected module is not used.	Empty

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

If a character other than alphanumeric is specified, a character code conversion error (0xF100001) occurs at the execution of the Open function.

Properties of program setting type controls

The following table shows the properties of the ActProgType control.

Property name (Type)	Description	Default value
ActMxUnitSeries(LONG)	Specify the series of connection target module. The property value is set automatically, and it cannot be changed. ■Property value 0 (0x00): Specify programmable controller/motion controller/GOT. 1 (0x01): Specify inverter. 2 (0x02): Specify robot controller.	0 (0x00)
ActNetworkNumber(LONG)	Specify the network number for MELSECNET/H. (Specify "0" (0x00) when specifying the host station.) Specify the following value for the multi-drop connection (via Q series-compatible C24). ■Property value of ActIntelligentPreferenceBit 0 (0x00): Specify the own network. 1 (0x01): Specify another network of multi-drop destination.	0 (0x00)
ActStationNumber(LONG)	Specify the station number for MELSECNET/H or CC-Link. (Specify "255" (0xFF) when specifying the host station.) Specify the system number for GX Simulator3. Specify the following value for the multi-drop connection (via Q series-compatible C24). ■Property value of ActIntelligentPreferenceBit 0 (0x00): Specify the own network. 1 (0x01): Specify another network of multi-drop destination. For inverter communication, specify the inverter station number (0 to 31) to be connected.	255 (0xFF)
ActUnitNumber(LONG)	Specify the module number of the serial communication module or the station number when the target is the Q series-compatible intelligent function module. This setting is invalid when the target is not a serial communication or Q series-compatible intelligent function module. For multi-drop link, specify the module number of the target serial communication module.	0 (0x00)
ActConnectUnitNumber(LONG)	Specify the module number of a serial communication module or Q series-compatible E71. For multi-drop link, specify the module number of the requesting serial communication module. For multi-drop link via CPU COM communication, however, the module number of the requesting station is not required. (Specify "0" (0x00)) Specify "0" (0x00) for other than multi-drop link. For Q series-compatible E71, specify the relay target station number. (Fixed to "0" (0x00) for access within the own network) For access to another network via MELSECNET/10, specify the station number set in the parameter of the connected Ethernet module.	0 (0x00)
ActIONumber(LONG)	Specify the module I/O number. For multi-drop link or intelligent function module access, specify the actual I/O number (start I/O number divided by 16) of the target serial communication module or intelligent function module. (For multi-drop link, specify the I/O number of the relayed or requesting station) Specify "992" (0x3E0) to "1023" (0x3FF) when making access to another station via the host station CPU or network.	1023 (0x3FF)

Property name (Type)	Description	Default value
ActCpuType(LONG)	<p>When the ActMxUnitSeries property is set to '0' (programmable controller/motion controller) Specify the target CPU to communicate with. In the parameter, specify the CPU type in the following table.  Page 176 ActCpuType(LONG): When the ActMxUnitSeries property is set to '0' (programmable controller/motion controller)</p> <p>When the ActMxUnitSeries property is set to '1' (inverter) Specify the target Inverter to communicate with. In the parameter, specify the CPU type in the following table.  Page 177 ActCpuType(LONG): When the ActMxUnitSeries property is set to '1' (inverter)</p> <p>When the ActMxUnitSeries property is set to '2' (robot) Specify the target robot to communicate with. In the parameter, specify the CPU type in the following table.  Page 177 ActCpuType(LONG): When the ActMxUnitSeries property is set to '2' (robot)</p>	34 (CPU_Q02CPU)
ActPortNumber(LONG)	<p>Specify the connection port number of personal computer. When an Ethernet module is connected, set any value as a port number of the requesting source (personal computer). When "=0" was specified as a port number, the Station No. ↔ IP information system should be the automatic response system. (When the system other than the automatic response system is selected, set the fixed value "5001.") When the network board is used, specify the first board as PORT_1, and the second and subsequent boards as PORT_2, PORT_3, and so on.  Page 178 ActPortNumber(LONG)</p>	1 (PORT_1)
ActBaudRate(LONG)	<p>Specify the baud rate for serial communication. ■Property value (Property window input value) BAUDRATE_300 (300): 300 bps BAUDRATE_600 (600): 600 bps BAUDRATE_1200 (1200): 1200 bps BAUDRATE_2400 (2400): 2400 bps BAUDRATE_4800 (4800): 4800 bps BAUDRATE_9600 (9600): 9600 bps BAUDRATE_19200 (19200): 19200 bps BAUDRATE_38400 (38400): 38400 bps BAUDRATE_57600 (57600): 57600 bps BAUDRATE_115200 (115200): 115200 bps For inverter communication, specify the property value greater than BAUDRATE_4800 (4800).</p>	19200 (BAUDRATE_19200)
ActDataBits(LONG)	<p>Specify the number of bits of the byte data sent and received for serial communication. ■Property value (Property window input value) DATABIT_7 (7): 7 bits DATABIT_8 (8): 8 bits For robot controller, specify the character size.</p>	8 (DATABIT_8)
ActParity(LONG)	<p>Specify the parity system used for serial communication. ■Property value (Property window input value) NO_PARITY (0): No parity ODD_PARITY (1): Odd EVEN_PARITY (2): Even</p>	1 (ODD_PARITY)
ActStopBits(LONG)	<p>Specify the number of stop bits used for serial communication. ■Property value (Property window input value) STOPBIT_ONE (0): 1 stop bit STOPBITS_TWO (2): 2 stop bits For robot controller, specify the following setting. ■Property value (Property window input value) ONESTOPBIT (0): 1 stop bit ONE5STOPBITS (1): 1.5 stop bits TWOSTOPBITS (2): 2 stop bits</p>	0 (STOPBIT_ONE)
ActControl(LONG)	<p>Specify the control setting of the signal line.  Page 178 ActControl(LONG)</p>	8 (TRC_DTR_OR_RTS)
ActHostAddress(BSTR)	<p>Pointer which indicates the connection host name (IP address) for Ethernet communication. When setting the first character of each octet to '0,' the value is processed as octal number.</p>	1.1.1.1
ActCpuTimeOut(LONG)	<p>Specify the CPU watchdog timer for Ethernet communication. (Unit: Multiplied by 250 ms) For FXCPU/inverter, specify the transmission waiting time for serial communication of FXCPU/inverter. (Unit: Multiplied by 10 ms) For robot controller, specify the transmission timeout time. (1000 to 30000 ms)</p>	0 (0x00)

Property name (Type)	Description	Default value
ActTimeOut(LONG)	Set the time-out value of communication between the personal computer and programmable controller/inverter. (Unit: ms) For robot controller, specify the transmission timeout time. (5000 to 120000 ms) A time-out processing may be performed internally depending on the communication route in MX Component. For details, refer to the following:  Page 673 Time-Out Periods	10000
ActSumCheck(LONG)	Specify whether sum check is applied or not. This setting is valid for serial communication module only. ■Property value (Property window input value) NO_SUM_CHECK (0): Without sum check SUM_CHECK (1): With sum check	0 (NO_SUM_CHECK)
ActSourceNetworkNumber(LONG)	Specify the requesting network number when an Ethernet module is specified. Specify the same network number (which is specified in the network parameter) as that of Ethernet connected.	0 (0x00)
ActSourceStationNumber(LONG)	Specify the requesting station number (personal computer side station number) when an Ethernet module is specified. Set the setting to avoid setting the same station number as that of an Ethernet module set within the same Ethernet loop.	0 (0x00)
ActDestinationPortNumber(LONG)	Specify the port number of the target when Ethernet communication is specified. When accessing another network, specify the relay destination port number. For the system other than the automatic response system, set the following setting. ■Setting MELSEC iQ-R series-compatible E71 (TCP/IP) - MELSOFT connection: Fixed to "5002" MELSEC iQ-R series-compatible E71 (TCP/IP) - OPS connection: Depending on network parameter MELSEC iQ-R series-compatible E71 (UDP/IP): Fixed to "5001" Q series-compatible E71 (TCP/IP) - Other than redundant CPU: Fixed to "5002" Q series-compatible E71 (TCP/IP) - MELSOFT connection ^{*1} : Fixed to "5002" Q series-compatible E71 (TCP/IP) - OPS connection ^{*1} : Depending on network parameter Q series-compatible E71 (UDP/IP): Fixed to "5001"	0 (0x00)
ActDestinationIONumber(LONG)	For multi-drop connection (via MELSEC iQ-R series-compatible C24/Q series-compatible C24/L series-compatible C24/CC-Link), specify the actual I/O number (start I/O number divided by 16) of the last access target station. (When the target is an intelligent function module) Specify "992" (0x3E0) to "1023" (0x3FF) when making access to another station via the host station CPU or network.	0 (0x00)
ActMultiDropChannelNumber(LONG)	For multi-drop connection (via MELSEC iQ-R series-compatible C24/Q series-compatible C24/L series-compatible C24/CC-Link), specify the multi-drop connection channel number (CH1/CH2). For robot controller, specify the retry count at communication error. (0 to 10 times) This setting is invalid for other connections.	0 (0x00)
ActThroughNetworkType(LONG)	Specify whether MELSECNET/10 is included in the relayed network when accessing other station via network. ■Property value 0 (0x00): MELSECNET/10 is not included. 1 (0x01): MELSECNET/10 is included. For robot controller, specify the protocol to be used (communication method). (Procedural (0)/ Nonprocedural (1))	0 (0x00)
ActIntelligentPreferenceBit(LONG)	For multi-drop connection (via MELSEC iQ-R series-compatible C24/Q series-compatible C24/L series-compatible C24/CC-Link), specify whether to relay the network of the multi-drop link destination. (To differentiate the own network module.) ■Property value 0 (0x00): Another network of multi-drop link destination is not accessed. 1 (0x01): Another network of multi-drop link destination is accessed.	0 (0x00)
ActDidPropertyBit(LONG)	For accessing the Q series-compatible host station intelligent function module (intelligent function module mounted on the host station CPU), the setting of "ActUnitNumber" is not necessary by invalidating the following setting. (Specify the property with the setting of "ActIONumber" only.) ■Property value 0 (0x00): Module number is validated. 1 (0x01): Module number is invalidated.	1 (0x01)
ActDsidPropertyBit(LONG)	For multi-drop connection (via MELSEC iQ-R series-compatible C24/Q series-compatible C24/L series-compatible C24/CC-Link), the setting of "ActDestinationIONumber" is not necessary by invalidating the following setting. However, when the following setting is invalidated, validate the setting of "ActDidPropertyBit." (Specify the property with the setting of "ActUnitNumber.") ■Property value 0 (0x00): I/O number of the last access target station is validated. 1 (0x01): I/O number of the last access target station is invalidated.	1 (0x01)

Property name (Type)	Description	Default value
ActPacketType(LONG)	Specify whether CR/LF exists or not. ■Property value (Property window input value) CRLF_NONE (0): Without CR/LF CRLF_CR (1): With CR CRLF_CRLF (2): With CR/LF	1(CRLF_CR)
ActPassword(BSTR)*2	Specify a password to disable the password set to the password protected modules.*3*4*5 This setting is ignored when a password protected module is not used.	Empty
ActTargetSimulator(LONG)	Specify the connection destination GX Simulator2 in start status. When connecting to FXCPU, specify "0" (0x00). ■Property value 0 (0x00): None (When only one simulator is in start status, connects to the simulator in start status. When multiple simulators are in start status, search for the simulators in start status and connect them in alphabetical order.) 1 (0x01): Simulator A 2 (0x02): Simulator B 3 (0x03): Simulator C 4 (0x04): Simulator D Specify the PLC number of the connection destination GX Simulator3 in start status. Specify the connection destination MT Simulator2 in start status. ■Property value 2 (0x02): Simulator No.2 3 (0x03): Simulator No.3 4 (0x04): Simulator No.4	0 (0x00)
ActUnitType(LONG)	Specify the module type connected to the physical port. ☞ Page 178 ActUnitType(LONG)	0x13 (UNIT_QNCPU)
ActProtocolType(LONG)	Specify the communication protocol type of the module (board) to be connected. ☞ Page 180 ActProtocolType(LONG)	0x04 (PROTOCOL_SERIAL)
ActATCommand	They can be used in only MX Component Version 4, and cannot be used in MX Component Version 5 or later.	—
ActATCommandPasswordCancelRetryTimes		
ActATCommandResponseWaitTime		
ActCallbackCancelWaitTime		
ActCallbackDelayTime		
ActCallbackNumber		
ActCallbackReceptionWaitingTimeOut		
ActConnectionCDWaitTime		
ActConnectionModemReportWaitTime		
ActConnectWay		
ActDialNumber		
ActDisconnectionCDWaitTime		
ActDisconnectionDelayTime		
ActLineType		
ActOutsideLineNumber		
ActPasswordCancelResponseWaitTime		
ActTransmissionDelayTime		

*1 For details, refer to the following:

☞ Q Corresponding Ethernet Interface Module User's Manual (Basic)

*2 This property can be used for Ethernet communication.

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

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

*5 If characters other than alphanumeric are specified, a character code conversion error (0xF100001) occurs at the execution of the Open function.

■ActCpuType(LONG): When the ActMxUnitSeries property is set to '0' (programmable controller/motion controller)

Property value (Property window input value)			Target CPU
CPU type	Dec.	Hex.	
CPU_R00CPU	4609	0x1201	R00CPU
CPU_R01CPU	4610	0x1202	R01CPU
CPU_R02CPU	4611	0x1203	R02CPU
CPU_R04CPU	4097	0x1001	R04CPU
CPU_R04ENCPU	4104	0x1008	R04ENCPU
CPU_R08CPU	4098	0x1002	R08CPU
CPU_R08ENCPU	4105	0x1009	R08ENCPU
CPU_R08PCPU	4354	0x1102	R08PCPU
CPU_R08PSFCPU	4369	0x1111	R08PSFCPU
CPU_R08SFCPU	4386	0x1122	R08SFCPU
CPU_R16CPU	4099	0x1003	R16CPU
CPU_R16ENCPU	4106	0x100A	R16ENCPU
CPU_R16PCPU	4355	0x1103	R16PCPU
CPU_R16PSFCPU	4370	0x1112	R16PSFCPU
CPU_R16SFCPU	4387	0x1123	R16SFCPU
CPU_R32CPU	4100	0x1004	R32CPU
CPU_R32ENCPU	4107	0x100B	R32ENCPU
CPU_R32PCPU	4356	0x1104	R32PCPU
CPU_R32PSFCPU	4371	0x1113	R32PSFCPU
CPU_R32SFCPU	4388	0x1124	R32SFCPU
CPU_R120CPU	4101	0x1005	R120CPU
CPU_R120ENCPU	4108	0x100C	R120ENCPU
CPU_R120PCPU	4357	0x1105	R120PCPU
CPU_R120PSFCPU	4372	0x1114	R120PSFCPU
CPU_R120SFCPU	4389	0x1125	R120SFCPU
CPU_R16MTCPU	4113	0x1011	R16MTCPU
CPU_R32MTCPU	4114	0x1012	R32MTCPU
CPU_R64MTCPU	4115	0x1013	R64MTCPU
CPU_R12CCPU_V	4129	0x1021	R12CCPU-V
CPU_R102WCPU_W	4136	0x1028	R102WCPU_W
CPU_L04HCPU	4625	0x1211	L04HCPU
CPU_L08HCPU	4626	0x1212	L08HCPU
CPU_L16HCPU	4627	0x1213	L16HCPU
CPU_L32HCPU	4628	0x1214	L32HCPU
CPU_Q00UJCPU	128	0x80	Q00UJCPU
CPU_Q00UCPU	129	0x81	Q00UCPU
CPU_Q01UCPU	130	0x82	Q01UCPU
CPU_Q02UCPU	131	0x83	Q02UCPU
CPU_Q03UDCPU	112	0x70	Q03UDCPU
CPU_Q03UDECPU	144	0x90	Q03UDECPU
CPU_Q03UDVCPU	209	0xD1	Q03UDVCPU
CPU_Q04UDHCPU	113	0x71	Q04UDHCPU
CPU_Q04UDEHCPU	145	0x91	Q04UDEHCPU
CPU_Q04UDVCPU	210	0xD2	Q04UDVCPU
CPU_Q04UDPVCPU	71	0x47	Q04UDPVCPU
CPU_Q06UDHCPU	114	0x72	Q06UDHCPU
CPU_Q06UDEHCPU	146	0x92	Q06UDEHCPU
CPU_Q06UDVCPU	211	0xD3	Q06UDVCPU
CPU_Q06UDPVCPU	72	0x48	Q06UDPVCPU

Property value (Property window input value)			Target CPU
CPU type	Dec.	Hex.	
CPU_Q10UDHCPU	117	0x75	Q10UDHCPU
CPU_Q10UDEHCPU	149	0x95	Q10UDEHCPU
CPU_Q12PRHCPU	67	0x43	Q12PRHCPU
CPU_Q13UDHCPU	115	0x73	Q13UDHCPU
CPU_Q13UDEHCPU	147	0x93	Q13UDEHCPU
CPU_Q13UDVCPU	212	0xD4	Q13UDVCPU
CPU_Q13UDPVCPU	73	0x49	Q13UDPVCPU
CPU_Q20UDHCPU	118	0x76	Q20UDHCPU
CPU_Q20UDEHCPU	150	0x96	Q20UDEHCPU
CPU_Q25PRHCPU	68	0x44	Q25PRHCPU
CPU_Q26UDHCPU	116	0x74	Q26UDHCPU
CPU_Q26UDEHCPU	148	0x94	Q26UDEHCPU
CPU_Q26UDVCPU	213	0xD5	Q26UDVCPU
CPU_Q26UDPVCPU	74	0x4A	Q26UDPVCPU
CPU_Q50UDEHCPU	152	0x98	Q50UDEHCPU
CPU_Q100UDEHCPU	154	0x9A	Q100UDEHCPU
CPU_L02SCPU	163	0xA3	L02SCPU
CPU_L02CPU	161	0xA1	L02CPU
CPU_L06CPU	165	0xA5	L06CPU
CPU_L26CPU	164	0xA4	L26CPU
CPU_L26CPUBT	162	0xA2	L26CPU-BT
CPU_Q12DC_V	88	0x58	Q12DCCPU-V
CPU_Q24DHC_V	89	0x59	Q24DHCCPU-V
CPU_Q24DHC_LS	91	0x5B	Q24DHCCPU-LS
CPU_Q24DHC_VG	92	0x5C	Q24DHCCPU-VG
CPU_Q26DHC_LS	93	0x5D	Q26DHCCPU-LS
CPU_QS001CPU	96	0x60	QS001CPU
CPU_Q172DCPU	1573	0x625	Q172DCPU
CPU_Q173DCPU	1574	0x626	Q173DCPU
CPU_Q172DSCPU	1578	0x62A	Q172DSCPU
CPU_Q173DSCPU	1579	0x62B	Q173DSCPU
CPU_FX3SCPU	522	0x20A	FX3SCPU
CPU_FX3GCPU	521	0x209	FX3G(C)CPU
CPU_FX3UCCPU	520	0x208	FX3U(C)CPU
CPU_FX5UCPU	528	0x0210	FX5UCPU
CPU_FX5UJCPU	529	0x0211	FX5UJCPU
CPU_FX5SCPU	530	0x0212	FX5SCPU
CPU_BOARD	1025	0x401	For own board access ^{*1}

*1 Except for CPU boards and AF boards.

ActCpuType(LONG): When the ActMxUnitSeries property is set to '1' (inverter)

Property value (Property window input value)			Target inverter
CPU type	Dec.	Hex.	
INV_A800	7776	0x1E60	A800

ActCpuType(LONG): When the ActMxUnitSeries property is set to '2' (robot)

Property value (Property window input value)			Target robot
CPU type	Dec.	Hex.	
UCPU_CRD700	77825	0x13001	CRnD-7xx/CR75x-D

■ActPortNumber(LONG)

Property value (Property window input value)			Description
Port number	Dec.	Hex.	
PORT_1	1	0x01	Communication port 1
PORT_2	2	0x02	Communication port 2
PORT_3	3	0x03	Communication port 3
PORT_4	4	0x04	Communication port 4
PORT_5	5	0x05	Communication port 5
PORT_6	6	0x06	Communication port 6
PORT_7	7	0x07	Communication port 7
PORT_8	8	0x08	Communication port 8
PORT_9	9	0x09	Communication port 9
PORT_10	10	0x0A	Communication port 10

■ActControl(LONG)

Property value (Property window input value)			Description
Control setting	Dec.	Hex.	
TRC_DTR	1	0x01	DTR control
TRC_RTS	2	0x02	RTS control
TRC_DRT_AND_RTS	7	0x07	DTR control and RTS control
TRC_DTR_OR_RTS	8	0x08	DTR control or RTS control

■ActUnitType(LONG)

Module type	Connection system	Value
UNIT_RJ71C24	MELSEC iQ-R series-compatible C24 module direct connection	0x1000
	MELSEC iQ-R series-compatible C24 connection via GOT2000 series	
UNIT_QJ71C24	Q series-compatible C24 module direct connection	0x19
	Q series-compatible C24 connection via GOT2000 series	
UNIT_LJ71C24	L series-compatible C24 module direct connection	0x54
	L series-compatible C24 connection via GOT2000 series	
UNIT_FX485BD	FX extended port (RS-485) connection	0x24
UNIT_RJ71EN71	Via MELSEC iQ-R series-compatible E71 connection (IP specification)	0x1001
UNIT_RJ71EN71_DIRECT	Via MELSEC iQ-R series-compatible E71 connection (Direct connection)	0x1005
UNIT_QJ71E71	Q series-compatible E71 Ethernet port connection/direct connection	0x1A
UNIT_LJ71E71	L series-compatible E71 module connection	0x5C
UNIT_REETHER	RCPU Ethernet port connection (IP specification)	0x1002
UNIT_LHETHER	LHCPU Ethernet port connection (IP specification)	
UNIT_REETHER_DIRECT	RCPU Ethernet port direct connection	0x1003
UNIT_LHETHER_DIRECT	LHCPU Ethernet port direct connection	
UNIT_FXVETHER	FX5CPU Ethernet port connection	0x2001
UNIT_FXVETHER_DIRECT	FX5CPU Ethernet port direct connection	0x2002
UNIT_LNETHER	LCPU Ethernet port connection	0x52
UNIT_LNETHER_DIRECT	LCPU Ethernet port direct connection	0x53
UNIT_QNETHER	QCPU Ethernet port connection	0x2C
UNIT_QNETHER_DIRECT	QCPU Ethernet port direct connection	0x2D
UNIT_RJ71GN11	RJ71GN11-T2 module for RCPUs IP specified connection	0x1006
UNIT_RJ71GN11_DIRECT	RJ71GN11-T2 module for RCPUs direct connection	0x1007
UNIT_NZ2GF_ETB	Ethernet adapter module (NZ2GF-ETB) connection	0x59
UNIT_NZ2GF_ETB_DIRECT	Ethernet adapter module (NZ2GF-ETB) direct connection	0x5A
UNIT_FXETHER	FXCPU Ethernet adapter (FX3U-ENET-ADP) via hub	0x4A
UNIT_FXETHER_DIRECT	FXCPU Ethernet adapter (FX3U-ENET-ADP) direct connection	0x4B
UNIT_FXVENET	FX5CPU Ethernet module (FX5-ENET, FX5-ENET/IP) IP specified connection	0x2004

Module type	Connection system	Value
UNIT_FXVENET_DIRECT	FX5CPU Ethernet module (FX5-ENET, FX5-ENET/IP) direct connection	0x200E
UNIT_FXVCLGN	Via FX5-CCLGN-MS module connection	0x200D
UNIT_FXVCPU	FX5CPU Serial port connection	0x2000
UNIT_QNCPU	QCPU (Q mode) RS232C port direct connection	0x13
	GOT2000 series QCPU (Q mode) direct connection	
UNIT_LNCPUCPU	LCPU-RS232C port direct connection	0x50
	GOT2000 series LCPU direct connection	
UNIT_QNMOTION	Q motion CPU-RS232C port direct connection	0x1C
UNIT_FXCPU	FXCPU-RS422 port direct connection	0x0F
	FXCPU USB direct connection	
	GOT2000 series FXCPU direct connection	
UNIT_RUSB	RCPU USB port direct connection	0x1004
UNIT_LHUSB	LHCPU USB port direct connection	
UNIT_FXVUSB	FX5CPU USB port direct connection	0x200C
UNIT_QNUSB	QCPU (Q mode) USB port direct connection	0x16
UNIT_LNUSB	LCPU USB port direct connection	0x51
UNIT_QSUSB	QSCPU (Safety programmable controller) USB port connection	0x29
UNIT_QNMOTIONUSB	Q motion CPU USB port direct connection	0x1D
UNIT_MNETHBOARD	MELSECNET/H board connection	0x1E
UNIT_MNETGBOARD	CC-Link IE Controller Network board connection	0x2B
UNIT_CCIETSNBOARD	CC-Link IE TSN board connection	0x73
UNIT_CCIIEFBOARD	CC-Link IE Field Network board connection	0x2F
UNIT_CCLINKBOARD	CC-Link Ver.2 board connection	0x0C
UNIT_G4QNCPU	Q series-compatible AJ65BT-G4-S3 module direct connection	0x1B
UNIT_SIMULATOR2*1	Simulator (GX Simulator2) connection	0x30
	Simulator (MT Simulator2) connection	
UNIT_SIMULATOR3	Simulator (GX Simulator3) connection	0x31
UNIT_A900GOT	GOT2000 series connection	0x21
UNIT_GOT_RJ71EN71	MELSEC iQ-R series-compatible E71 connection via GOT2000 series	0x1051
UNIT_GOT_QJ71E71	Q series-compatible E71 connection via GOT2000 series	0x40
UNIT_GOT_LJ71E71	L series-compatible E71 connection via GOT2000 series	0x5D
UNIT_GOT_REETHER	RCPU Ethernet port connection via GOT2000 series	0x1052
UNIT_GOT_QNETHER	QCPU Ethernet port connection via GOT2000 series	0x41
UNIT_GOT_LNETHER	LCPU Ethernet port connection via GOT2000 series	0x55
UNIT_GOT_NZ2GF_ETB	Ethernet adapter module (NZ2GF-ETB) connection via GOT2000 series	0x5B
UNIT_GOT_FXETHER	FXCPU Ethernet adapter (FX3U-ENET-ADP) connection via GOT2000 series	0x61
UNIT_GOT_FXENET	FXCPU Ethernet module (FX3U-ENET) connection via GOT2000 series	0x62
UNIT_GOT_FXVENET	FX5CPU Ethernet module (FX5-ENET, FX5-ENET/IP) connection via GOT2000 series	0x2008
UNIT_GOT_FXVCPU	FX5CPU Serial port connection via GOT2000/1000 series	0x2005
UNIT_GOTETHER_FXVCPU	FX5CPU Ethernet port connection via GOT2000/1000 series	0x2006
UNIT_GOT_FXVETHER	Built-in Ethernet port FX5CPU connection via GOT2000/1000 series	0x2007
UNIT_GOTETHER_RJ71C24	RCPU Ethernet port (RS422) connection via GOT2000 series	0x1061
UNIT_GOTETHER_QNCPU	QCPU (Q mode) Ethernet port (RS422) connection via GOT2000 series	0x56
UNIT_GOTETHER_LNCPUCPU	LCPU Ethernet port (RS232C) connection via GOT2000 series	0x57
UNIT_GOTETHER_FXCPU	FXCPU Ethernet port connection via GOT2000 series	0x60
UNIT_GOTETHER_QN_ETHER	GOT Ethernet transparent (Ethernet-GOT-Ethernet-QnCPU) connection	0x6F

*1 When the CPU type is Q motion CPU, connected to MT Simulator2. Other than that, connected to GX Simulator2.

■ActProtocolType(LONG)

Communication protocol type	Connection system	Value
PROTOCOL_SERIAL	Via serial port	0x04(4)
PROTOCOL_USB	Via USB port	0x0D(13)
PROTOCOL_TCPIP	Via TCP/IP	0x05(5)
PROTOCOL_UDPIP	Via UDP/IP	0x08(8)
PROTOCOL_MNETH	Via MELSECNET/H board	0x0F(15)
PROTOCOL_MNETG	Via CC-Link IE Controller Network board	0x14(20)
PROTOCOL_CCIETSN	Via CC-Link IE TSN board	0x1C(28)
PROTOCOL_CCIEF	Via CC-Link IE Field Network board	0x15(21)
PROTOCOL_CCLINK	Via CC-Link	0x07(7)
PROTOCOL_USBGOT	Via USB port and GOT	0x13(19)
PROTOCOL_SHAREDMEMORY	Via shared memory server (Simulator)	0x06(6)
COMM_RS232C* ¹	Inverter RS-232C communication	0x00(0)
COMM_USB* ¹	Inverter USB communication	0x01(1)
RC_PROTOCOL_SERIAL	Robot controller RS-232C communication	0x01(1)
RC_PROTOCOL_USB	Robot controller USB communication	0x04(4)
RC_PROTOCOL_TCPIP	Robot controller Ethernet (TCP/IP) communication	0x02(2)

*1 The communication protocol type can be used when ActMxUnitSeries is 1 (0x01).

10 PROPERTY SETTINGS OF COMMUNICATION ROUTES

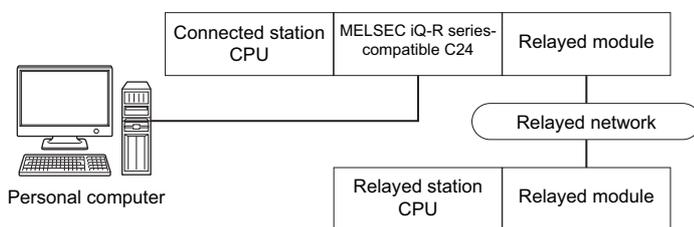
This chapter explains the details of accessible communication routes and property settings.

10.1 Serial Communication

Serial communication when the connected station is MELSEC iQ-R series-compatible C24 (1)

When a relayed module other than the connected station side MELSEC iQ-R series-compatible C24 exists

Configuration



Accessibility and property patterns

○(n): Accessible ('n' is the property pattern number), ×: Inaccessible

Connected station CPU		Relayed network	Relayed station CPU				
R CPU	R motion CPU		R CPU	RCCPU	R motion CPU	LH CPU	FX5 CPU
○(1)	○(1) ^{*1*3}	CC IE TSN	○(2)	×	×	×	×
		CC IE Control CC IE Field	○(2) ^{*2}	○(2)	×	×	×
		MELSECNET/H	×	×	×	×	×
		Ethernet	○(2)	×	○(2)	×	×
		Serial communication	○(3)	×	○(3)	×	×
		CC-Link	○(4)	○(4)	○(4)	×	×

Connected station CPU		Relayed network	Relayed station CPU					
R CPU	R motion CPU		Q CPU (Q mode)	QCCPU	L CPU	QSCPU	Q motion CPU	FX CPU
○(1)	○(1) ^{*1*3}	CC IE TSN	×	×	×	×	×	×
		CC IE Control CC IE Field	○(2)	×	○(2) ^{*2}	×	×	×
		MELSECNET/H	○(2)	×	×	×	×	×
		Ethernet	○(2)	×	○(2)	×	×	×
		Serial communication	○(3) ^{*4}	×	○(3)	×	×	×
		CC-Link	○(4)	×	○(4)	×	×	×

*1 Relayed station CPUs cannot be accessed via R motion CPU.

*2 Since CC-Link IE Controller Network is not supported, R CPU and L CPU relayed by CC-Link IE Controller Network cannot be accessed.

*3 Only CPU No.2 or later in a multiple CPU configuration can be accessed.

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

Property list

The following table shows the property settings of communication routes.

The setting is not necessary for those properties without description.

Property	Default value	Property patterns			
		○(1)	○(2) ^{*1}	○(3)	○(4)
ActBaudRate	19200 (BAUDRATE_19200)	Match to the setting of MELSEC iQ-R series-compatible C24.			
ActConnectUnitNumber	0 (0x00)	Connected station side module station number			
ActControl	8 (TRC_DTR_OR_RTSL)	Depending on the used cable.			
ActCpuType	34 (CPU_Q02CPU)	CPU type corresponding to the target station			
ActDestinationIONumber	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station side ■For single CPU • Fixed to 1023 (0x3FF) ■For multiple CPUs • Connected CPU: 1023 (0x3FF) • No. 1: 992 (0x3E0) • No. 2: 993 (0x3E1) • No. 3: 994 (0x3E2) • No. 4: 995 (0x3E3)	Target station side ■For single CPU • Fixed to 1023 (0x3FF) ■For multiple CPUs • Connected CPU: 1023 (0x3FF) • No. 1: 992 (0x3E0) • No. 2: 993 (0x3E1) • No. 3: 994 (0x3E2) • No. 4: 995 (0x3E3) ■For redundant CPU • No specification: 1023 (0x3FF)
ActDidPropertyBit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActDsidPropertyBit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActIntelligentPreferenceBit	0 (0x00)	Fixed to 0 (0x00)			
ActIONumber ^{*2}	1023 (0x3FF)	■For single CPU • Fixed to 1023 (0x3FF) ■For multiple CPUs • Connected CPU: 1023 (0x3FF) • No. 1: 992 (0x3E0) • No. 2: 993 (0x3E1) • No. 3: 994 (0x3E2) • No. 4: 995 (0x3E3) ■For redundant CPU • No specification: 1023 (0x3FF)	Target station side ■For single CPU • Fixed to 1023 (0x3FF) ■For multiple CPUs • Connected CPU: 1023 (0x3FF) • No. 1: 992 (0x3E0) • No. 2: 993 (0x3E1) • No. 3: 994 (0x3E2) • No. 4: 995 (0x3E3) ■For redundant CPU • No specification: 1023 (0x3FF)	Connected station side relayed module I/O address	Connected station side relayed module I/O address
ActMultiDropChannelNumber ^{*3}	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Multi-drop channel number	Fixed to 0 (0x00)
ActNetworkNumber	0 (0x00)	Fixed to 0 (0x00)	Target station side module network number	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActParity	1 (ODD_PARITY)	Fixed to ODD_PARITY			
ActPortNumber	1 (PORT_1)	Personal computer side COM port number			
ActProtocolType	0x04 (PROTOCOL_SERIAL)	PROTOCOL_SERIAL			
ActPacketType	0x01 (PACKET_PL1)	PACKET_PL1			
ActHostAddress	1.1.1.1	Fixed to NULL			
ActCpuTimeOut	0 (0x00)	Fixed to 0 (0x00)			
ActDataBits	8 (DATABIT_8)	Fixed to 8 (0x08)			
ActStopBits	0 (STOPBIT_ONE)	Fixed to 0 (0x00)			
ActSumCheck	0 (NO_SUM_CHECK)	Fixed to 0 (0x00)			
ActSourceNetworkNumber	0 (0x00)	Fixed to 0 (0x00)			
ActSourceStationNumber	0 (0x00)	Fixed to 0 (0x00)			

Property	Default value	Property patterns			
		○(1)	○(2) ^{*1}	○(3)	○(4)
ActDestinationPortNumber	0 (0x00)	Fixed to 0 (0x00)			
ActStationNumber	255 (0xFF)	Fixed to 255 (0xFF)	Target station side module station number	Fixed to 255 (0xFF)	Fixed to 255 (0xFF)
ActThroughNetworkType	0 (0x00)	<ul style="list-style-type: none"> • MELSECNET/10 is not included.: 0 (0x00) • MELSECNET/10 is included.: 1 (0x01) 			
ActTimeOut	10000	Any value specified by user in ms units			
ActUnitNumber	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station side module station number	Target station side module station number
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_RJ71C24			

*1 Note the following considerations when accessing via Ethernet module (MELSEC iQ-R series-compatible E71).

- For ActNetworkNumber and ActStationNumber, specify the value set in the parameter setting of the target station side MELSEC iQ-R series-compatible E71.

- Set "Station No. ⇔ IP information" in the parameter setting of MELSEC iQ-R series-compatible E71. When setting the parameter, specify the setting other than the automatic response system (any of the IP address calculation system, table conversion system, and combined system) for "Station No. ⇔ IP information system."

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

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

0: Default channel of module

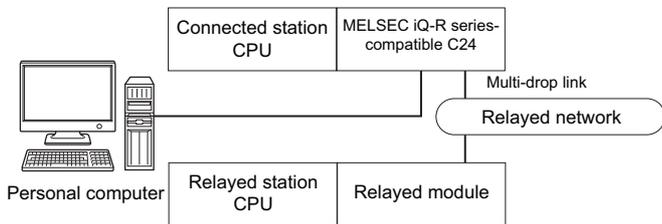
1: Channel 1

2: Channel 2

Serial communication when the connected station is MELSEC iQ-R series-compatible C24 (2)

When performing multi-drop link on the connected station side MELSEC iQ-R series-compatible C24 with the relayed module

Configuration



Accessibility and property patterns

○(n): Accessible ('n' is the property pattern number), ×: Inaccessible

Connected station CPU		Relayed network	Relayed station CPU				
RCPU			RCPU	RCCPU	R motion CPU	LHCPU	FX5CPU
Independent mode*1	○(1)	Serial communication	○(2)	×	×	×	×
Synchronous mode*2	×		○(3)	×	×	×	×

Connected station CPU		Relayed network	Relayed station CPU					
RCPU			QCPU (Q mode)	QCCPU	LCPU	QSCPU	Q motion CPU	FXCPU
Independent mode*1	○(1)	Serial communication	○(2)*3	×	○(2)	×	×	×
Synchronous mode*2	×		○(3)*3	×	○(3)	×	×	×

*1 For the Independent mode, set the following parameters.

- CH1 side: Operation setting of Transmission setting = Independent (0), Communication protocol setting = 0
- CH2 side: Operation setting of Transmission setting = Independent (0)

*2 For the Synchronous mode, set the following parameters.

- CH1 side: Operation setting of Transmission setting = Independent (0), Communication protocol setting = 8
- CH2 side: Operation setting of Transmission setting = Synchronous (1), Communication protocol setting = 0

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

Point

When the connected station side MELSEC iQ-R series-compatible C24 is set to the synchronous mode, always validate (turn ON) "sum check (SW06)" for the transmission specification software switch setting of the MELSEC iQ-R series-compatible C24 parameter.

If it is invalidated (OFF), a communication error occurs and the communication is disabled.

Property list

The following table shows the property settings of communication routes.

The setting is not necessary for those properties without description.

Property	Default value	Property patterns		
		○(1)	○(2)	○(3)
ActBaudRate	19200 (BAUDRATE_19200)	Match to the setting of MELSEC iQ-R series-compatible C24.		
ActConnectUnitNumber	0 (0x00)	Connected station side module station number		Target station side module station number
ActControl	8 (TRC_DTR_OR_RT S)	Depending on the used cable.		
ActCpuType	34 (CPU_Q02CPU)	CPU type corresponding to the target station		
ActDestinationIONumber	0 (0x00)	Fixed to 0 (0x00)	Target station side ■For single CPU • Fixed to 1023 (0x3FF) ■For multiple CPUs • Connected CPU: 1023 (0x3FF) • No. 1: 992 (0x3E0) • No. 2: 993 (0x3E1) • No. 3: 994 (0x3E2) • No. 4: 995 (0x3E3)	Fixed to 0 (0x00)
ActDidPropertyBit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 1 (0x01)
ActDsidPropertyBit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 1 (0x01)
ActIntelligentPreferenceBit	0 (0x00)	Fixed to 0 (0x00)		
ActIONumber*1	1023 (0x3FF)	■For single CPU • Fixed to 1023 (0x3FF) ■For multiple CPUs • Connected CPU: 1023 (0x3FF) • No. 1: 992 (0x3E0) • No. 2: 993 (0x3E1) • No. 3: 994 (0x3E2) • No. 4: 995 (0x3E3)	Connected station side relayed module I/O address	Target station side ■For single CPU • Fixed to 1023 (0x3FF) ■For multiple CPUs • Connected CPU: 1023 (0x3FF) • No. 1: 992 (0x3E0) • No. 2: 993 (0x3E1) • No. 3: 994 (0x3E2) • No. 4: 995 (0x3E3)
ActMultiDropChannelNumber*2	0 (0x00)	Fixed to 0 (0x00)	Multi-drop channel number	Fixed to 0 (0x00)
ActNetworkNumber	0 (0x00)	Fixed to 0 (0x00)		
ActParity	1 (ODD_PARITY)	Fixed to ODD_PARITY		
ActPortNumber	1 (PORT_1)	Personal computer side COM port number		
ActProtocolType	0x04 (PROTOCOL_SERIAL)	PROTOCOL_SERIAL		
ActPacketType	0x01 (PACKET_PLC1)	PACKET_PLC1		
ActHostAddress	1.1.1.1	Fixed to NULL		
ActCpuTimeOut	0 (0x00)	Fixed to 0 (0x00)		
ActDataBits	8 (DATABIT_8)	Fixed to 8 (0x08)		
ActStopBits	0 (STOPBIT_ONE)	Fixed to 0 (0x00)		
ActSumCheck	0 (NO_SUM_CHECK)	Fixed to 0 (0x00)		
ActSourceNetworkNumber	0 (0x00)	Fixed to 0 (0x00)		
ActSourceStationNumber	0 (0x00)	Fixed to 0 (0x00)		
ActDestinationPortNumber	0 (0x00)	Fixed to 0 (0x00)		
ActStationNumber	255 (0xFF)	Fixed to 255 (0xFF)		
ActThroughNetworkType	0 (0x00)	• MELSECNET/10 is not included.: 0 (0x00) • MELSECNET/10 is included.: 1 (0x01)		
ActTimeOut	10000	Any value specified by user in ms units		
ActUnitNumber	0 (0x00)	Fixed to 0 (0x00)	Target station side module station number	Fixed to 0 (0x00)

Property	Default value	Property patterns		
		○(1)	○(2)	○(3)
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_RJ71C24		

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

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

0: Default channel of module

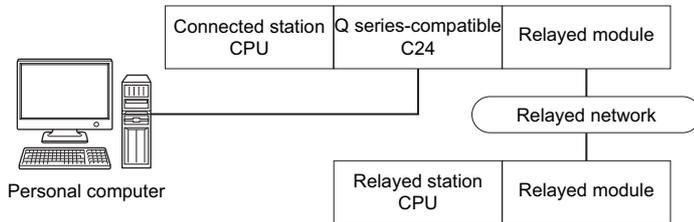
1: Channel 1

2: Channel 2

Serial communication when the connected station is Q series-compatible C24 (1)

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

Configuration



Accessibility and property patterns

○(n): Accessible ('n' is the property pattern number), ×: Inaccessible

Connected station CPU			Relayed network	Relayed station CPU				
QCPU (Q mode)	QCCP U	Q motion CPU		RCPU	RCCPU	R motion CPU	LHCPU	FX5CPU
○(1)	○(1) ^{*6}	○(1) ^{*1}	CC IE TSN	×	×	×	×	×
			CC IE Control CC IE Field	×	×	×	×	×
			MELSECNET/H	×	×	×	×	×
			Ethernet	×	×	×	×	×
			Serial communication	×	×	×	×	×
			CC-Link	×	×	×	×	×

Connected station CPU			Relayed network	Relayed station CPU					
QCPU (Q mode)	QCCP U	Q motion CPU		QCPU (Q mode)	QCCPU	LCPU	QSCPU	Q motion CPU	FXCPU
○(1)	○(1) ^{*6}	○(1) ^{*1}	CC IE TSN	×	×	×	×	×	×
			CC IE Control CC IE Field	○(2)	○(2) ^{*2}	○(2) ^{*3}	○(2) ^{*2}	○(2) ^{*2}	×
			MELSECNET/H	○(2)	○(2)	×	○(2)	○(2)	×
			Ethernet	○(2)	×	×	○(2)	○(2)	×
			Serial communication	○(3) ^{*4}	×	○(3)	×	○(3)	×
			CC-Link	○(4)	○(4)	○(4)	×	○(4)	○(4) ^{*5}

*1 Relayed station CPUs cannot be accessed via Q motion CPU.

*2 Since CC-Link IE Field Network is not supported, R motion CPU, Q12DCCPU-V (Basic mode), QSCPU, and Q motion CPU relayed by CC-Link IE Field Network cannot be accessed.

*3 Since CC-Link IE Controller Network is not supported, LCPUs relayed by CC-Link IE Controller Network cannot be accessed.

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

*5 Supported by FX3G(C)CPU and FX3U(C)CPU within the own network only.

*6 Only CPU No.2 or later in a multiple CPU configuration can be accessed.

Property list

The following table shows the property settings of communication routes.

The setting is not necessary for those properties without description.

Property	Default value	Property patterns			
		○(1)	○(2) ^{*1}	○(3)	○(4)
ActBaudRate	19200 (BAUDRATE_19200)	Match to the setting of Q series-compatible C24.			
ActConnectUnitNumber	0 (0x00)	Connected station side module station number			
ActControl	8 (TRC_DTR_OR_RTS)	Depending on the used cable.			
ActCpuType	34 (CPU_Q02CPU)	CPU type corresponding to the target station			
ActDestinationIONumber	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station side ■For single CPU • Fixed to 1023 (0x3FF) ■For multiple CPUs • Connected CPU: 1023 (0x3FF) • No. 1: 992 (0x3E0) • No. 2: 993 (0x3E1) • No. 3: 994 (0x3E2) • No. 4: 995 (0x3E3)	Target station side ■For single CPU • Fixed to 1023 (0x3FF) ■For multiple CPUs • Connected CPU: 1023 (0x3FF) • No. 1: 992 (0x3E0) • No. 2: 993 (0x3E1) • No. 3: 994 (0x3E2) • No. 4: 995 (0x3E3) ■For redundant CPU • Control system: 976 (0x3D0) • No specification: 1023 (0x3FF)
ActDidPropertyBit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActDsidPropertyBit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActHostAddress	1.1.1.1	Fixed to NULL			
ActIntelligentPreferenceBit	0 (0x00)	Fixed to 0 (0x00)			
ActIONumber ^{*2}	1023 (0x3FF)	■For single CPU • Fixed to 1023 (0x3FF) ■For multiple CPUs • Connected CPU: 1023 (0x3FF) • No. 1: 992 (0x3E0) • No. 2: 993 (0x3E1) • No. 3: 994 (0x3E2) • No. 4: 995 (0x3E3) ■For redundant CPU • Control system: 976 (0x3D0) • No specification: 1023 (0x3FF)	Target station side ■For single CPU • Fixed to 1023 (0x3FF) ■For multiple CPUs • Connected CPU: 1023 (0x3FF) • No. 1: 992 (0x3E0) • No. 2: 993 (0x3E1) • No. 3: 994 (0x3E2) • No. 4: 995 (0x3E3) ■For redundant CPU • Control system: 976 (0x3D0) • No specification: 1023 (0x3FF)	Connected station side module I/O address	Connected station side module I/O address
ActMultiDropChannelNumber ^{*3}	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Multi-drop channel number	Fixed to 0 (0x00)
ActNetworkNumber	0 (0x00)	Fixed to 0 (0x00)	Target station side module network number	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActParity	1 (ODD_PARITY)	Match to the setting of Q series-compatible C24.			
ActPortNumber	1 (PORT_1)	Personal computer side COM port number			
ActProtocolType	0x04 (PROTOCOL_SERIAL)	PROTOCOL_SERIAL (0x04)			
ActStationNumber	255 (0xFF)	Fixed to 255 (0xFF)	Target station side module station number	Fixed to 255 (0xFF)	Fixed to 255 (0xFF)
ActThroughNetworkType	0 (0x00)	• MELSECNET/10 is not included.: 0 (0x00) • MELSECNET/10 is included.: 1 (0x01)			
ActTimeOut	10000	Any value specified by user in ms units			

Property	Default value	Property patterns			
		○(1)	○(2) ^{*1}	○(3)	○(4)
ActUnitNumber	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station side module station number	Target station side module station number
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_QJ71C24 (0x19)			

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

- For ActNetworkNumber and ActStationNumber, specify the value set in the parameter setting of the target station side Q series-compatible E71.

- Set "Station No. ↔ IP information" in the parameter setting of Q series-compatible E71. When setting the parameter, specify the setting other than the automatic response system (any of the IP address calculation system, table conversion system, and combined system) for "Station No. ↔ IP information system."

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

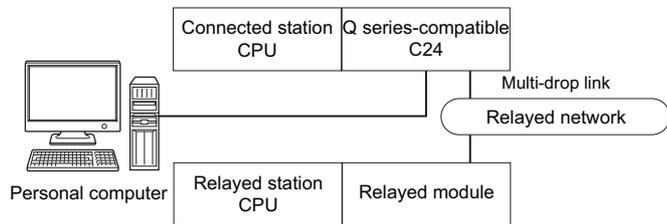
*3 Specify the following value for the channel number to be multi-drop linked.

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

Serial communication when the connected station is Q series-compatible C24 (2)

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

Configuration



Accessibility and property patterns

○(n): Accessible ('n' is the property pattern number), ×: Inaccessible

Connected station CPU		Relayed network	Relayed station CPU				
QCPU (Q mode), QCCPU ^{*3}			RCPU	RCCPU	R motion CPU	LHCPU	FX5CPU
Independent mode ^{*1}	○(1) ^{*2}	Serial communication	×	×	×	×	×
Synchronous mode ^{*1}	×		×	×	×	×	×

Connected station CPU		Relayed network	Relayed station CPU					
QCPU (Q mode), QCCPU ^{*3}			QCPU (Q mode)	QCCPU	LCPU	QSCPU	Q motion CPU	FXCPU
Independent mode ^{*1}	○(1) ^{*2}	Serial communication	○(2) ^{*2}	×	○(2)	×	×	×
Synchronous mode ^{*1}	×		○(3) ^{*2}	×	○(3)	×	×	×

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

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

*3 Only CPU No.2 or later in a multiple CPU configuration can be accessed.

Point

When the connected station side Q series-compatible C24 is set to the synchronous mode, always validate (turn ON) "sum check (SW06)" for the transmission specification software switch setting of the Q series-compatible C24 parameter.

If it is invalidated (OFF), a communication error occurs and the communication is disabled.

Property list

The following table shows the property settings of communication routes.

The setting is not necessary for those properties without description.

Property	Default value	Property patterns		
		○(1)	○(2)	○(3)
ActBaudRate	19200 (BAUDRATE_19200)	Match to the setting of Q series-compatible C24.		
ActConnectUnitNumber	0 (0x00)	Connected station side module station number		
ActControl	8 (TRC_DTR_OR_RTS)	Depending on the used cable.		
ActCpuType	34 (CPU_Q02CPU)	CPU type corresponding to the target station		
ActDestinationIONumber	0 (0x00)	Fixed to 0 (0x00)	Target station side ■For single CPU Fixed to 1023 (0x3FF) ■For multiple CPUs • Connected CPU: 1023 (0x3FF) • No. 1: 992 (0x3E0) • No. 2: 993 (0x3E1) • No. 3: 994 (0x3E2) • No. 4: 995 (0x3E3) ■For redundant CPU • Control system: 976 (0x3D0) • No specification: 1023 (0x3FF)	Fixed to 0 (0x00)
ActDidPropertyBit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 1 (0x01)
ActDsidPropertyBit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 1 (0x01)
ActIntelligentPreferenceBit	0 (0x00)	Fixed to 0 (0x00)		
ActIONumber* ¹	1023 (0x3FF)	■For single CPU • Fixed to 1023 (0x3FF) ■For multiple CPUs • Connected CPU: 1023 (0x3FF) • No. 1: 992 (0x3E0) • No. 2: 993 (0x3E1) • No. 3: 994 (0x3E2) • No. 4: 995 (0x3E3)	Connected station side relayed module I/O address	Target station side ■For single CPU • Fixed to 1023 (0x3FF) ■For multiple CPUs • Connected CPU: 1023 (0x3FF) • No. 1: 992 (0x3E0) • No. 2: 993 (0x3E1) • No. 3: 994 (0x3E2) • No. 4: 995 (0x3E3)
ActMultiDropChannelNumber* ²	0 (0x00)	Fixed to 0 (0x00)	Multi-drop channel number	Fixed to 0 (0x00)
ActNetworkNumber	0 (0x00)	Fixed to 0 (0x00)		
ActParity	1 (ODD_PARITY)	Match to the setting of Q series-compatible C24.		
ActPortNumber	1 (PORT_1)	Personal computer side COM port number		
ActProtocolType	0x04 (PROTOCOL_SERIAL)	PROTOCOL_SERIAL (0x04)		
ActStationNumber	255 (0xFF)	Fixed to 255 (0xFF)		
ActThroughNetworkType	0 (0x00)	• MELSECNET/10 is not included.: 0 (0x00) • MELSECNET/10 is included.: 1 (0x01)		
ActTimeOut	10000	Any value specified by user in ms units		
ActUnitNumber	0 (0x00)	Fixed to 0 (0x00)	Target station side module station number	Fixed to 0 (0x00)
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_QJ71C24 (0x19)		

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

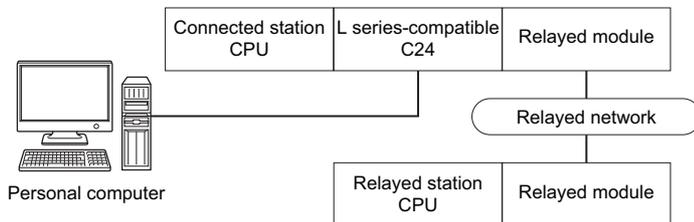
*2 Specify the following value for the channel number to be multi-drop linked.

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

Serial communication when the connected station is L series-compatible C24 (1)

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

Configuration



Accessibility and property patterns

○(n): Accessible ('n' is the property pattern number), ×: Inaccessible

Connected station CPU	Relayed network	Relayed station CPU				
		RCPU	RCCPU	R motion CPU	LHCPU	FX5CPU
○(1)	CC IE TSN	×	×	×	×	×
	CC IE Field ^{*1}	×	×	×	×	×
	MELSECNET/H	×	×	×	×	×
	Ethernet	×	×	×	×	×
	Serial communication	×	×	×	×	×
	CC-Link	×	×	×	×	×

Connected station CPU	Relayed network	Relayed station CPU					
		QCPU (Q mode)	QCCPU	LCPU	QSCPU	Q motion CPU	FXCPU
○(1)	CC IE TSN	×	×	×	×	×	×
	CC IE Field ^{*1}	○(4)	○(4) ^{*2}	○(4)	×	×	×
	MELSECNET/H	×	×	×	×	×	×
	Ethernet	○(4)	×	○(4)	×	×	×
	Serial communication	○(2) ^{*3}	×	○(2)	×	×	×
	CC-Link	○(3)	○(3)	○(3)	×	×	×

*1 Since CC-Link IE Controller Network is not supported, LCPU relayed by CC-Link IE Controller Network cannot be accessed.

*2 Since CC-Link IE Field Network is not supported, Q12DCCPU-V (Basic mode) relayed by CC-Link IE Field Network cannot be accessed.

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

Property list

The following table shows the property settings of communication routes.

The setting is not necessary for those properties without description.

Property	Default value	Property patterns			
		○(1)	○(2)	○(3)	○(4)
ActBaudRate	19200 (BAUDRATE_19200)	BAUDRATE_9600, BAUDRATE_19200, BAUDRATE_38400, BAUDRATE_57600, BAUDRATE_115200			
ActConnectUnitNumber	0 (0x00)	Connected station side module station number			
ActControl	8 (TRC_DTR_OR_RTS)	Depending on the used cable.			
ActCpuTimeOut	0 (0x00)	Fixed to 0 (0x00)			
ActCpuType	34 (CPU_Q02CPU)	CPU type corresponding to the target station			
ActDataBits	8 (DATABIT_8)	Fixed to 8 (DATABIT_8)			
ActDestinationIONumber	0 (0x00)	Fixed to 0 (0x00)	Target station side ■For single CPU • Fixed to 1023 (0x3FF) ■For multiple CPUs • Connected CPU: 1023 (0x3FF) • No. 1: 992 (0x3E0) • No. 2: 993 (0x3E1) • No. 3: 994 (0x3E2) • No. 4: 995 (0x3E3)	Target station side ■For single CPU • Fixed to 1023 (0x3FF) ■For multiple CPUs • Connected CPU: 1023 (0x3FF) • No. 1: 992 (0x3E0) • No. 2: 993 (0x3E1) • No. 3: 994 (0x3E2) • No. 4: 995 (0x3E3) ■For redundant CPU • Control system: 976 (0x3D0) • No specification: 1023 (0x3FF)	Fixed to 0 (0x00)
ActDestinationPortNumber	0 (0x00)	Fixed to 0 (0x00)			
ActDidPropertyBit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Fixed to 1 (0x01)
ActDsidPropertyBit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Fixed to 1 (0x01)
ActHostAddress	1.1.1.1	Fixed to NULL			
ActIntelligentPreferenceBit	0 (0x00)	Fixed to 0 (0x00)			
ActIONumber*1	1023 (0x3FF)	Fixed to 1023 (0x3FF)	Connected station side relayed module I/O address	Connected station side relayed module I/O address	Target station side ■For single CPU • Fixed to 1023 (0x3FF) ■For multiple CPUs • Connected CPU: 1023 (0x3FF) • No. 1: 992 (0x3E0) • No. 2: 993 (0x3E1) • No. 3: 994 (0x3E2) • No. 4: 995 (0x3E3) ■For redundant CPU • Control system: 976 (0x3D0) • No specification: 1023 (0x3FF)
ActMultiDropChannelNumber*2	0 (0x00)	Fixed to 0 (0x00)	Multi-drop channel number	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActNetworkNumber	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station side module network number
ActPacketType	0x01 (PACKET_PLC1)	Fixed to 0x01 (PACKET_PLC1)			
ActParity	1 (ODD_PARITY)	Fixed to 1 (ODD_PARITY)			
ActPortNumber	1 (PORT_1)	Personal computer side COM port number			
ActProtocolType	0x04 (PROTOCOL_SERIAL)	PROTOCOL_SERIAL (0x04)			
ActSourceNetworkNumber	0 (0x00)	Fixed to 0 (0x00)			
ActSourceStationNumber	0 (0x00)	Fixed to 0 (0x00)			

Property	Default value	Property patterns			
		○(1)	○(2)	○(3)	○(4)
ActStationNumber	255 (0xFF)	Fixed to 255 (0xFF)	Fixed to 255 (0xFF)	Fixed to 255 (0xFF)	Target station side module station number
ActStopBits	0 (STOPBIT_ONE)	Fixed to 0 (STOPBIT_ONE)			
ActSumCheck	0 (NO_SUM_CHECK)	Fixed to 0 (NO_SUM_CHECK)			
ActThroughNetworkType	0 (0x00)	<ul style="list-style-type: none"> • MELSECNET/10 is not included.: 0 (0x00) • MELSECNET/10 is included.: 1 (0x01) 			
ActTimeOut	10000	Any value specified by user in ms units			
ActUnitNumber	0 (0x00)	Fixed to 0 (0x00)	Target station side module station number	Target station side module station number	Fixed to 0 (0x00)
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_LJ71C24 (0x54)			

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

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

0: Default channel of module

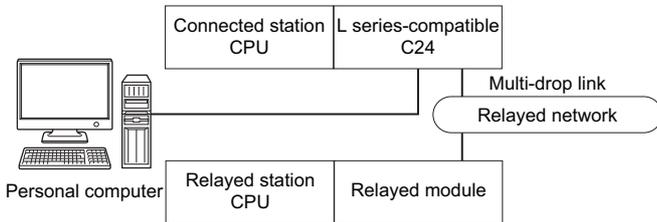
1: Channel 1

2: Channel 2

Serial communication when the connected station is L series-compatible C24 (2)

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

Configuration



Accessibility and property patterns

○(n): Accessible ('n' is the property pattern number), ×: Inaccessible

Connected station CPU		Relayed network	Relayed station CPU				
LCPU			RCPU	RCCPU	R motion CPU	LHCPU	FX5CPU
Independent mode*1	○(1)*3	Serial communication	×	×	×	×	×
Synchronous mode*2	×		×	×	×	×	×

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

*1 For the Independent mode, set the following parameters.

- CH1 side: Operation setting of Transmission setting = Independent (0), Communication protocol setting = 0
- CH2 side: Operation setting of Transmission setting = Independent (0)

*2 For the Synchronous mode, set the following parameters.

- CH1 side: Operation setting of Transmission setting = Independent (0), Communication protocol setting = 8
- CH2 side: Operation setting of Transmission setting = Synchronous (1), Communication protocol setting = 0

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

Point

When the connected station side L series-compatible C24 is set to the synchronous mode, always validate (turn ON) "sum check (SW06)" for the transmission specification software switch setting of the L series-compatible C24 parameter.

If it is invalidated (OFF), a communication error occurs and the communication is disabled.

Property list

The following table shows the property settings of communication routes.

The setting is not necessary for those properties without description.

Property	Default value	Property patterns		
		○(1)	○(2)	○(3)
ActBaudRate	19200 (BAUDRATE_19200)	Match to the setting of L series-compatible C24.		
ActConnectUnitNumber	0 (0x00)	Connected station side module station number		
ActControl	8 (TRC_DTR_OR_RTS)	Depending on the used cable.		
ActCpuType	34 (CPU_Q02CPU)	CPU type corresponding to the target station		
ActDestinationIONumber	0 (0x00)	Fixed to 0 (0x00)	Target station side ■For single CPU • Fixed to 1023 (0x3FF) ■For multiple CPUs • Connected CPU: 1023 (0x3FF) • No. 1: 992 (0x3E0) • No. 2: 993 (0x3E1) • No. 3: 994 (0x3E2) • No. 4: 995 (0x3E3)	Fixed to 0 (0x00)
ActDidPropertyBit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 1 (0x01)
ActDsidPropertyBit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 1 (0x01)
ActIntelligentPreferenceBit	0 (0x00)	Fixed to 0 (0x00)		
ActIONumber*1	1023 (0x3FF)	Fixed to 1023 (0x3FF)	Connected station side relayed module I/O address	Target station side ■For single CPU • Fixed to 1023 (0x3FF) ■For multiple CPUs • Connected CPU: 1023 (0x3FF) • No. 1: 992 (0x3E0) • No. 2: 993 (0x3E1) • No. 3: 994 (0x3E2) • No. 4: 995 (0x3E3)
ActMultiDropChannelNumber*2	0 (0x00)	Fixed to 0 (0x00)	Multi-drop channel number	Fixed to 0 (0x00)
ActNetworkNumber	0 (0x00)	Fixed to 0 (0x00)		
ActParity	1 (ODD_PARITY)	Match to the setting of L series-compatible C24.		
ActPortNumber	1 (PORT_1)	Personal computer side COM port number		
ActProtocolType	0x04 (PROTOCOL_SERIAL)	PROTOCOL_SERIAL (0x04)		
ActStationNumber	255 (0xFF)	Fixed to 255 (0xFF)		
ActThroughNetworkType	0 (0x00)	• MELSECNET/10 is not included.: 0 (0x00) • MELSECNET/10 is included.: 1 (0x01)		
ActTimeOut	10000	Any value specified by user in ms units		
ActUnitNumber	0 (0x00)	Fixed to 0 (0x00)	Target station side module station number	Fixed to 0 (0x00)
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_LJ71C24 (0x54)		

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

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

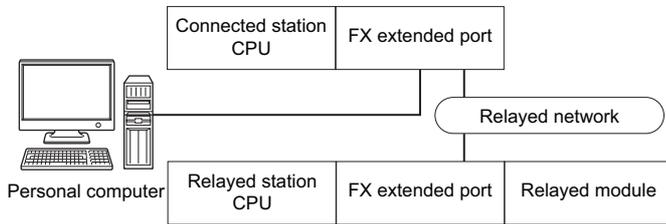
0: Default channel of module

1: Channel 1

2: Channel 2

Serial communication when the connected station is FX extended port

Configuration



Accessibility and property patterns

○(n): Accessible ('n' is the property pattern number), ×: Inaccessible

Connected station CPU FXCPU	Relayed network	Relayed station CPU				
		RCPU	RCCPU	R motion CPU	LHCPU	FX5CPU
○(1)	CC IE TSN	×	×	×	×	×
	CC IE Control CC IE Field	×	×	×	×	×
	MELSECNET/H	×	×	×	×	×
	Ethernet	×	×	×	×	×
	Serial communication	×	×	×	×	×
	CC-Link	×	×	×	×	×

Connected station CPU FXCPU	Relayed network	Relayed station CPU					
		QCPU (Q mode)	QCCPU	LCPU	QSCPU	Q motion CPU	FXCPU
○(1)	CC IE TSN	×	×	×	×	×	×
	CC IE Control CC IE Field	×	×	×	×	×	×
	MELSECNET/H	×	×	×	×	×	×
	Ethernet	×	×	×	×	×	×
	Serial communication	×	×	×	×	×	○(1) ^{*1}
	CC-Link	×	×	×	×	×	×

*1 FX0CPU, FX0SCPU, FX1CPU, FXUCPU, and FX2CCPU cannot be accessed.

Property list

The following table shows the property settings of communication routes.

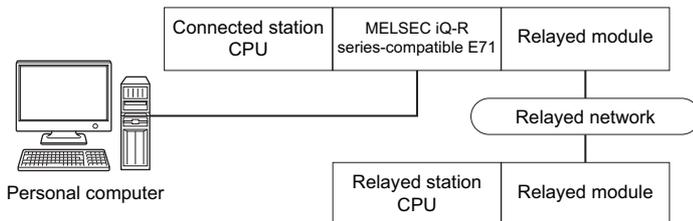
The setting is not necessary for those properties without description.

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

10.2 Ethernet Communication

Ethernet communication when the connected station is MELSEC iQ-R series-compatible E71 (TCP)

Configuration



Accessibility and property patterns

○(n): Accessible ('n' is the property pattern number), ×: Inaccessible

Connected station CPU		Relayed network	Relayed station CPU				
R CPU	R motion CPU		R CPU	RCCPU	R motion CPU	LH CPU	FX5 CPU
○(1)	○(1) ^{*1*2}	CC IE TSN	×	×	×	×	×
		CC IE Control CC IE Field	×	×	×	×	×
		MELSECNET/H	×	×	×	×	×
		Ethernet	○(2)	×	○(2)	×	×
		Serial communication	○(3)	×	○(3)	×	×
		CC-Link	○(4)	○(4)	○(4)	×	×

Connected station CPU		Relayed network	Relayed station CPU					
R CPU	R motion CPU		Q CPU (Q mode)	QCCPU	L CPU	QSCPU	Q motion CPU	FX CPU
○(1)	○(1) ^{*1*2}	CC IE TSN	×	×	×	×	×	×
		CC IE Control CC IE Field	×	×	×	×	×	×
		MELSECNET/H	×	×	×	×	×	×
		Ethernet	○(2)	×	○(2)	×	×	×
		Serial communication	○(3)	×	○(3)	×	×	×
		CC-Link	○(4)	×	○(4)	×	×	×

*1 Relayed station CPUs cannot be accessed via R motion CPU.

*2 Only CPU No.2 or later in a multiple CPU configuration can be accessed.

Property list

The following table shows the property settings of communication routes.

The setting is not necessary for those properties without description.

Property	Default value	Property patterns			
		○(1)	○(2) ^{*1}	○(3)	○(4)
ActConnectUnitNumber	0 (0x00)	Fixed to 0 (0x00)	Connected station side module station number	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActCpuType	34 (CPU_Q02CPU)	CPU type corresponding to the target station			
ActDestinationIONumber ^{*3}	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station side ■For single CPU • Fixed to 1023 (0x3FF) ■For multiple CPUs • Connected CPU: 1023 (0x3FF) • No. 1: 992 (0x3E0) • No. 2: 993 (0x3E1) • No. 3: 994 (0x3E2) • No. 4: 995 (0x3E3)	Target station side ■For single CPU • Fixed to 1023 (0x3FF) ■For multiple CPUs • Connected CPU: 1023 (0x3FF) • No. 1: 992 (0x3E0) • No. 2: 993 (0x3E1) • No. 3: 994 (0x3E2) • No. 4: 995 (0x3E3) ■For redundant CPU • No specification: 1023 (0x3FF)
ActDestinationPortNumber	0 (0x00)	5002 for MELSOFT connection Any port number for OPS connection. ^{*2}			
ActDidPropertyBit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActDsidPropertyBit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActHostAddress	1.1.1.1	Host name or IP address of the connected station side module			
ActIONumber ^{*3}	1023 (0x3FF)	■For single CPU • Fixed to 1023 (0x3FF) ■For multiple CPUs • Connected CPU: 1023 (0x3FF) • No. 1: 992 (0x3E0) • No. 2: 993 (0x3E1) • No. 3: 994 (0x3E2) • No. 4: 995 (0x3E3) ■For redundant CPU • Control system: 976 (0x3D0) • No specification: 1023 (0x3FF)	Target station side ■For single CPU • Fixed to 1023 (0x3FF) ■For multiple CPUs • Connected CPU: 1023 (0x3FF) • No. 1: 992 (0x3E0) • No. 2: 993 (0x3E1) • No. 3: 994 (0x3E2) • No. 4: 995 (0x3E3) ■For redundant CPU • No specification: 1023 (0x3FF)	Connected station side relayed module I/O address	Connected station side relayed module I/O address
ActMultiDropChannelNumber ^{*4}	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Multi-drop channel number	Fixed to 0 (0x00)
ActNetworkNumber ^{*5}	0 (0x00)	Target station side module network number	Target station side module network number	Connected station side MELSEC iQ-R series-compatible E71 network number	Connected station side MELSEC iQ-R series-compatible E71 network number
ActPassword	Empty	Password set to MELSEC iQ-R series-compatible E71 on the connected station side			
ActProtocolType	0x04 (PROTOCOL_SERIAL)	PROTOCOL_TCPIP (0x05)			
ActPacketType	0x01 (PACKET_PLC1)	PACKET_PLC1			
ActPortNumber	1 (PORT_1)	Personal computer side port number			
ActBaudRate	19200 (BAUDRATE_19200)	Fixed to 0 (0x00)			
ActCpuTimeOut	0 (0x00)	Fixed to 0 (0x00)			
ActControl	8 (TRC_DTR_OR_RTSD)	Fixed to 0 (0x00)			
ActDataBits	8 (DATABIT_8)	Fixed to 0 (0x00)			
ActParity	1 (ODD_PARITY)	Fixed to 0 (0x00)			

Property	Default value	Property patterns			
		○(1)	○(2) ^{*1}	○(3)	○(4)
ActStopBits	0 (STOPBIT_ONE)	Fixed to 0 (0x00)			
ActSumCheck	0 (NO_SUM_CHECK)	Fixed to 0 (0x00)			
ActIntelligentPreferenceBit	0 (0x00)	Fixed to 0 (0x00)			
ActSourceNetworkNumber	0 (0x00)	Personal computer side network number			
ActSourceStationNumber ^{*6}	0 (0x00)	Personal computer side station number			
ActStationNumber ^{*5}	255 (0xFF)	Target station side module station number	Target station side module station number	Connected station side MELSEC iQ-R series-compatible E71 station number	Connected station side MELSEC iQ-R series-compatible E71 station number
ActThroughNetworkType	0 (0x00)	Fixed to 0 (0x00)			
ActTimeOut	10000	Any value specified by user in ms units			
ActUnitNumber	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station side module station number	Target station side module station number
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_RJ71EN71 (0x1001)			

*1 Note the following considerations when accessing via Ethernet module (MELSEC iQ-R series-compatible E71).

· For ActNetworkNumber and ActStationNumber, specify the value set in the parameter setting of the target station side MELSEC iQ-R series-compatible E71.

· Set "Station No. ⇔ IP information" in the parameter setting of MELSEC iQ-R series-compatible E71. When setting the parameter, specify the setting other than the automatic response system (any of the IP address calculation system, table conversion system, and combined system) for "Station No. ⇔ IP information system."

*2 When using the OPS connection function for Redundant CPU, specify any port number that was set to the network parameter of the CPU. (The setting range is $1025 \leq \text{port number} \leq 4999$ or $5003 \leq \text{port number} \leq 65534$)

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

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

0: Default channel of module

1: Channel 1

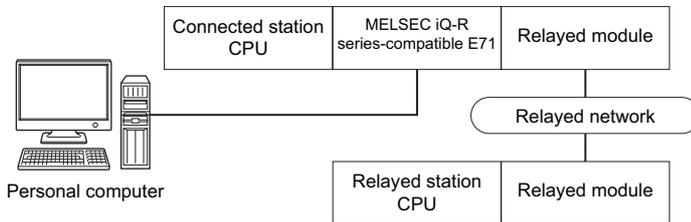
2: Channel 2

*5 When the property pattern is ○(1) or ○(2), specify the value set in the target station side parameter for ActNetworkNumber and ActStationNumber.

*6 Specify the personal computer side station number to avoid the same station number set for MELSEC iQ-R series-compatible E71 within the same Ethernet loop.

Ethernet communication when the connected station is MELSEC iQ-R series-compatible E71 (UDP)

Configuration



Accessibility and property patterns

○(n): Accessible ('n' is the property pattern number), ×: Inaccessible

Connected station CPU		Relayed network	Relayed station CPU				
RCPU	R motion CPU		RCPU	RCCPU	R motion CPU	LHCPU	FX5CPU
○(1)*1	○(1)*2*3	CC IE TSN	×	×	×	×	×
		CC IE Control CC IE Field	×	×	×	×	×
		MELSECNET/H	×	×	×	×	×
		Ethernet	○(2)	×	○(2)	×	×
		Serial communication	○(3)	×	○(3)	×	×
		CC-Link	○(4)	○(4)	○(4)	×	×

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

*1 Relayed station CPUs cannot be accessed when directly connecting to an MELSEC iQ-R series-compatible E71.

*2 Relayed station CPUs cannot be accessed via R motion CPU.

*3 Only CPU No.2 or later in a multiple CPU configuration can be accessed.

Property list

The following table shows the property settings of communication routes.

The setting is not necessary for those properties without description.

Property	Default value	Property patterns			
		○(1)	○(2) ^{*1}	○(3)	○(4)
ActConnectUnitNumber	0 (0x00)	Fixed to 0 (0x00)	Connected station side module station number	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActCpuType	34 (CPU_Q02CPU)	CPU type corresponding to the target station			
ActDestinationIONumber ^{*2}	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station side ■For single CPU • Fixed to 1023 (0x3FF) ■For multiple CPUs • Connected CPU: 1023 (0x3FF) • No. 1: 992 (0x3E0) • No. 2: 993 (0x3E1) • No. 3: 994 (0x3E2) • No. 4: 995 (0x3E3)	Target station side ■For single CPU • Fixed to 1023 (0x3FF) ■For multiple CPUs • Connected CPU: 1023 (0x3FF) • No. 1: 992 (0x3E0) • No. 2: 993 (0x3E1) • No. 3: 994 (0x3E2) • No. 4: 995 (0x3E3) ■For redundant CPU • No specification: 1023 (0x3FF)
ActDestinationPortNumber	0 (0x00)	For communication with IP address specified: 5001 For direct communication without IP address specified: 5003			
ActDidPropertyBit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActDsidPropertyBit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActHostAddress	1.1.1.1	For communication with IP address specified: Host name or IP address of the connected station side For direct communication without IP address specified: 255.255.255.255			
ActIONumber ^{*2}	1023 (0x3FF)	■For single CPU • Fixed to 1023 (0x3FF) ■For multiple CPUs • Connected CPU: 1023 (0x3FF) • No. 1: 992 (0x3E0) • No. 2: 993 (0x3E1) • No. 3: 994 (0x3E2) • No. 4: 995 (0x3E3) ■For redundant CPU • No specification: 1023 (0x3FF)	Target station side ■For single CPU • Fixed to 1023 (0x3FF) ■For multiple CPUs • Connected CPU: 1023 (0x3FF) • No. 1: 992 (0x3E0) • No. 2: 993 (0x3E1) • No. 3: 994 (0x3E2) • No. 4: 995 (0x3E3) ■For redundant CPU • No specification: 1023 (0x3FF)	Connected station side relayed module I/O address	Connected station side relayed module I/O address
ActMultiDropChannelNumber ^{*3}	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Multi-drop channel number	Fixed to 0 (0x00)
ActNetworkNumber ^{*4}	0 (0x00)	Target station side module network number	Target station side module network number	Connected station side MELSEC iQ-R series-compatible E71 network number	Connected station side MELSEC iQ-R series-compatible E71 network number
ActPassword	Empty	Password set to MELSEC iQ-R series-compatible E71 on the connected station side			
ActPortNumber	1 (PORT_1)	For communication with IP address specified: Personal computer side COM port number For direct communication without IP address specified: Unused			
ActProtocolType	0x04 (PROTOCOL_SERIAL)	PROTOCOL_UDPIP (0x08)			
ActPacketType	0x01 (PACKET_PLC1)	PACKET_PLC1			
ActBaudRate	19200 (BAUDRATE_19200)	Fixed to 0 (0x00)			
ActCpuTimeOut	0 (0x00)	Fixed to 0 (0x00)			
ActControl	8 (TRC_DTR_OR_RTSL)	Fixed to 0 (0x00)			
ActDataBits	8 (DATABIT_8)	Fixed to 0 (0x00)			

Property	Default value	Property patterns			
		○(1)	○(2) ^{*1}	○(3)	○(4)
ActParity	1 (ODD_PARITY)	Fixed to 0 (0x00)			
ActStopBits	0 (STOPBIT_ONE)	Fixed to 0 (0x00)			
ActSumCheck	0 (NO_SUM_CHECK)	Fixed to 0 (0x00)			
ActIntelligentPreferenceBit	0 (0x00)	Fixed to 0 (0x00)			
ActSourceNetworkNumber	0 (0x00)	Personal computer side network number			
ActSourceStationNumber ^{*5}	0 (0x00)	Personal computer side station number			
ActStationNumber ^{*4}	255 (0xFF)	Target station side module station number	Target station side module station number	Connected station side MELSEC iQ-R series-compatible E71 station number	Connected station side MELSEC iQ-R series-compatible E71 station number
ActThroughNetworkType	0 (0x00)	Fixed to 0 (0x00)			
ActTimeOut	10000	Any value specified by user in ms units			
ActUnitNumber	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station side module station number	Target station side module station number
ActUnitType	0x13 (UNIT_QNCPU)	<ul style="list-style-type: none"> • For communication with IP address specified: UNIT_RJ71EN71 (0x1001) • For direct communication without IP address specified: UNIT_RJ71EN71_DIRECT (0x1005) 			

*1 Note the following considerations when accessing via Ethernet module (MELSEC iQ-R series-compatible E71).

- For ActNetworkNumber and ActStationNumber, specify the value set in the parameter setting of the target station side MELSEC iQ-R series-compatible E71.

- Set "Station No. ⇔ IP information" in the parameter setting of MELSEC iQ-R series-compatible E71. When setting the parameter, specify the setting other than the automatic response system (any of the IP address calculation system, table conversion system, and combined system) for "Station No. ⇔ IP information system."

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

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

0: Default channel of module

1: Channel 1

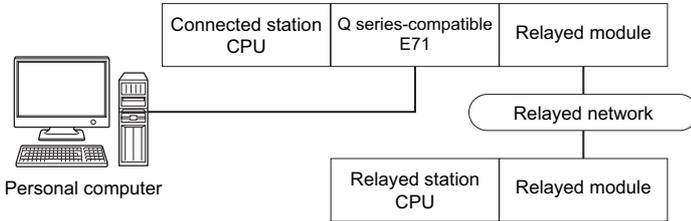
2: Channel 2

*4 When the property pattern is ○(2), specify the value set in the target station side parameter for ActNetworkNumber and ActStationNumber.

*5 Specify the personal computer side station number to avoid the same station number set for MELSEC iQ-R series-compatible E71 within the same Ethernet loop.

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

Configuration



Accessibility and property patterns

○(n): Accessible ('n' is the property pattern number), ×: Inaccessible

Connected station CPU				Relayed network	Relayed station CPU				
QCPU (Q mode)	QCCP U	QSCP U	Q motion CPU		RCPU	RCCPU	R motion CPU	LHCPU	FX5CP U
○(1)	○(1) ^{*6}	○(1) ^{*2}	○(1) ^{*1*2}	CC IE TSN	×	×	×	×	×
				CC IE Control CC IE Field	×	×	×	×	×
				MELSECNET/H	×	×	×	×	×
				Ethernet	×	×	×	×	×
				Serial communication	×	×	×	×	×
				CC-Link	×	×	×	×	×

Connected station CPU				Relayed network	Relayed station CPU					
QCPU (Q mode)	QCCP U	QSCP U	Q motion CPU		QCPU (Q mode)	QCCP U	LCP U	QSCP U	Q motion CPU	FXCP U
○(1)	○(1) ^{*6}	○(1) ^{*2}	○(1) ^{*1*2}	CC IE TSN	×	×	×	×	×	×
				CC IE Control CC IE Field	○(2)	○(2) ^{*3}	○(2) ^{*4}	○(2) ^{*2*3}	○(2) ^{*3}	×
				MELSECNET/H	○(2)	○(2)	×	○(2) ^{*2}	○(2)	×
				Ethernet	○(2)	×	×	○(2) ^{*2}	○(2)	×
				Serial communication	○(3) ^{*5}	×	○(3) ^{*5}	×	○(3)	×
				CC-Link	○(4)	○(4)	○(4)	×	○(4)	×

- *1 Only Q172D, Q173D, Q172DS and Q173DS can be accessed.
- *2 Relayed station CPUs cannot be accessed via QSCPU or Q motion CPU.
- *3 Since CC-Link IE Field Network is not supported, R motion CPU, Q12DCCPU-V (Basic mode), QSCPU, and Q motion CPU relayed by CC-Link IE Field Network cannot be accessed.
- *4 Since CC-Link IE Controller Network is not supported, LCP relayed by CC-Link IE Controller Network cannot be accessed.
- *5 For redundant CPU, serial communication modules on the main base unit cannot be accessed.
- *6 Only CPU No.2 or later in a multiple CPU configuration can be accessed.

Property list

The following table shows the property settings of communication routes.

The setting is not necessary for those properties without description.

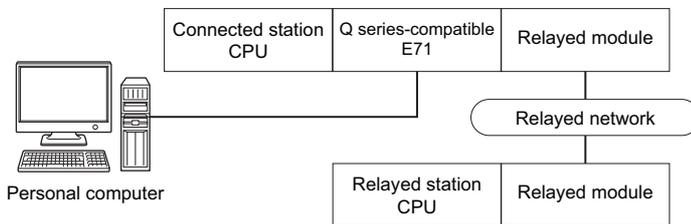
Property	Default value	Property patterns			
		○(1)	○(2)	○(3)	○(4)
ActConnectUnitNumber	0 (0x00)	Fixed to 0 (0x00)	Connected station side module station number	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActCpuType	34 (CPU_Q02CPU)	CPU type corresponding to the target station			
ActDestinationIONumber	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station side ■For single CPU • Fixed to 1023 (0x3FF) ■For multiple CPUs • Connected CPU: 1023 (0x3FF) • No. 1: 992 (0x3E0) • No. 2: 993 (0x3E1) • No. 3: 994 (0x3E2) • No. 4: 995 (0x3E3)	Target station side ■For single CPU • Fixed to 1023 (0x3FF) ■For multiple CPUs • Connected CPU: 1023 (0x3FF) • No. 1: 992 (0x3E0) • No. 2: 993 (0x3E1) • No. 3: 994 (0x3E2) • No. 4: 995 (0x3E3) ■For redundant CPU • Control system: 976 (0x3D0) • No specification: 1023 (0x3FF)
ActDestinationPortNumber	0 (0x00)	5002 for MELSOFT connection Any port number for OPS connection*1			
ActDidPropertyBit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActDsidPropertyBit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActHostAddress	1.1.1.1	Host name or IP address of the connected station side module			
ActIONumber*2	1023 (0x3FF)	■For single CPU • Fixed to 1023 (0x3FF) ■For multiple CPUs • Connected CPU: 1023 (0x3FF) • No. 1: 992 (0x3E0) • No. 2: 993 (0x3E1) • No. 3: 994 (0x3E2) • No. 4: 995 (0x3E3) ■For redundant CPU • Control system: 976 (0x3D0) • No specification: 1023 (0x3FF)	Target station side ■For single CPU • Fixed to 1023 (0x3FF) ■For multiple CPUs • Connected CPU: 1023 (0x3FF) • No. 1: 992 (0x3E0) • No. 2: 993 (0x3E1) • No. 3: 994 (0x3E2) • No. 4: 995 (0x3E3) ■For redundant CPU • Control system: 976 (0x3D0) • No specification: 1023 (0x3FF)	Connected station side relayed module I/O address	Connected station side relayed module I/O address
ActMultiDropChannelNumber*3	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Multi-drop channel number	Fixed to 0 (0x00)
ActNetworkNumber*4	0 (0x00)	Target station side module network number	Target station side module network number	Connected station side Q series-compatible E71 network number	Connected station side Q series-compatible E71 network number
ActPassword	Empty	Password set to Q series-compatible E71 on the connected station side			
ActProtocolType	0x04 (PROTOCOL_SERIAL)	PROTOCOL_TCPIP (0x05)			
ActSourceNetworkNumber	0 (0x00)	Personal computer side network number			
ActSourceStationNumber*5	0 (0x00)	Personal computer side station number			
ActStationNumber*4	255 (0xFF)	Target station side module station number	Target station side module station number	Connected station side Q series-compatible E71 station number	Connected station side Q series-compatible E71 station number
ActThroughNetworkType	0 (0x00)	• MELSECNET/10 is not included.: 0 (0x00) • MELSECNET/10 is included.: 1 (0x01)			
ActTimeOut	10000	Any value specified by user in ms units			

Property	Default value	Property patterns			
		○(1)	○(2)	○(3)	○(4)
ActUnitNumber	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station side module station number	Target station side module station number
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_QJ71E71 (0x1A)			

- *1 When using the OPS connection function for Redundant CPU, specify any port number that was set to the network parameter of the CPU. (The setting range is $1025 \leq \text{port number} \leq 4999$ or $5003 \leq \text{port number} \leq 65534$)
- *2 For the I/O address, specify the value of the actual start I/O number divided by 16.
- *3 Specify the following value for the channel number to be multi-drop linked.
0: Default channel of module
1: Channel 1
2: Channel 2
- *4 When the property pattern is ○(1) or ○(2), specify the value set in the target station side parameter for ActNetworkNumber and ActStationNumber.
- *5 Specify the personal computer side station number to avoid the same station number set for Q series-compatible E71 within the same Ethernet loop.

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

Configuration



Accessibility and property patterns

○(n): Accessible ('n' is the property pattern number), ×: Inaccessible

Connected station CPU				Relayed network	Relayed station CPU				
QCPU (Q mode)	QCCP U	QSCP U	Q motion CPU		RCPU	RCCPU	R motion CPU	LHCPU	FX5CP U
○(1)	○(1) ^{*6*7}	○(1) ^{*2}	○(1) ^{*1*2}	CC IE TSN	×	×	×	×	×
				CC IE Control CC IE Field	×	×	×	×	×
				MELSECNET/H	×	×	×	×	×
				Ethernet	×	×	×	×	×
				Serial communication	×	×	×	×	×
				CC-Link	×	×	×	×	×

Connected station CPU				Relayed network	Relayed station CPU					
QCPU (Q mode)	QCCP U	QSCP U	Q motion CPU		QCPU (Q mode)	QCCP U	LCPU	QSCP U	Q motion CPU	FXCP U
○(1)	○(1) ^{*6*} 7	○(1) ^{*2}	○(1) ^{*1*2}	CC IE TSN	×	×	×	×	×	×
				CC IE Control CC IE Field	○(2)	○(2) ^{*3}	○(2) ^{*4}	○(2) ^{*2*} 3	○(2) ^{*3}	×
				MELSECNET/H	○(2)	○(2)	×	○(2) ^{*2}	○(2)	×
				Ethernet	○(2)	×	×	○(2) ^{*2}	○(2)	×
				Serial communication	○(3) ^{*5}	×	○(3) ^{*5}	×	○(3)	×
				CC-Link	○(4)	○(4)	○(4)	×	○(4)	×

*1 Only Q172D, Q173D, Q172DS and Q173DS can be accessed.

*2 Relayed station CPUs cannot be accessed via QSCPU or Q motion CPU.

*3 Since CC-Link IE Field Network is not supported, R motion CPU, Q12DCCPU-V (Basic mode), QSCPU, and Q motion CPU relayed by CC-Link IE Field Network cannot be accessed.

*4 Since CC-Link IE Controller Network is not supported, LCPU relayed by CC-Link IE Controller Network cannot be accessed.

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

*6 Only CPU No.2 or later in a multiple CPU configuration can be accessed.

*7 Since QCCPU is not supported by MELSOFT direct connection, Ethernet port direct connection is not applicable.

Property list

The following table shows the property settings of communication routes.

The setting is not necessary for those properties without description.

Property	Default value	Property patterns			
		○(1)	○(2)	○(3)	○(4)
ActConnectUnitNumber	0 (0x00)	Fixed to 0 (0x00)	Connected station side module station number	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActCpuType	34 (CPU_Q02CPU)	CPU type corresponding to the target station			
ActDestinationIONumber	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station side ■For single CPU • Fixed to 1023 (0x3FF) ■For multiple CPUs • Connected CPU: 1023 (0x3FF) • No. 1: 992 (0x3E0) • No. 2: 993 (0x3E1) • No. 3: 994 (0x3E2) • No. 4: 995 (0x3E3)	Target station side ■For single CPU • Fixed to 1023 (0x3FF) ■For multiple CPUs • Connected CPU: 1023 (0x3FF) • No. 1: 992 (0x3E0) • No. 2: 993 (0x3E1) • No. 3: 994 (0x3E2) • No. 4: 995 (0x3E3) ■For redundant CPU • Control system: 976 (0x3D0) • No specification: 1023 (0x3FF)
ActDestinationPortNumber	0 (0x00)	5001			
ActDidPropertyBit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActDsidPropertyBit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActHostAddress	1.1.1.1	Host name or IP address of the connected station side module			
ActIONumber ^{*1}	1023 (0x3FF)	■For single CPU • Fixed to 1023 (0x3FF) ■For multiple CPUs • Connected CPU: 1023 (0x3FF) • No. 1: 992 (0x3E0) • No. 2: 993 (0x3E1) • No. 3: 994 (0x3E2) • No. 4: 995 (0x3E3) ■For redundant CPU • Control system: 976 (0x3D0) • No specification: 1023 (0x3FF)	Target station side ■For single CPU • Fixed to 1023 (0x3FF) ■For multiple CPUs • Connected CPU: 1023 (0x3FF) • No. 1: 992 (0x3E0) • No. 2: 993 (0x3E1) • No. 3: 994 (0x3E2) • No. 4: 995 (0x3E3) ■For redundant CPU • Control system: 976 (0x3D0) • No specification: 1023 (0x3FF)	Connected station side relayed module I/O address	Connected station side relayed module I/O address
ActMultiDropChannelNumber ^{*2}	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Multi-drop channel number	Fixed to 0 (0x00)
ActNetworkNumber ^{*3}	0 (0x00)	Target station side module network number	Target station side module network number	Connected station side Q series-compatible E71 network number	Connected station side Q series-compatible E71 network number
ActPassword	Empty	Password set to Q series-compatible E71 on the connected station side			
ActPortNumber ^{*4}	1 (PORT_1)	Personal computer side port number			
ActProtocolType	0x04 (PROTOCOL_SERIAL)	PROTOCOL_UDPIP (0x08)			
ActSourceNetworkNumber	0 (0x00)	Personal computer side network number			
ActSourceStationNumber ^{*5}	0 (0x00)	Personal computer side station number			
ActStationNumber ^{*3}	255 (0xFF)	Target station side module station number	Target station side module station number	Connected station side Q series-compatible E71 station number	Connected station side Q series-compatible E71 station number
ActThroughNetworkType	0 (0x00)	• MELSECNET/10 is not included.: 0 (0x00) • MELSECNET/10 is included.: 1 (0x01)			
ActTimeOut	10000	Any value specified by user in ms units			

Property	Default value	Property patterns			
		○(1)	○(2)	○(3)	○(4)
ActUnitNumber	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station side module station number	Target station side module station number
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_QJ71E71 (0x1A)			

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

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

0: Default channel of module

1: Channel 1

2: Channel 2

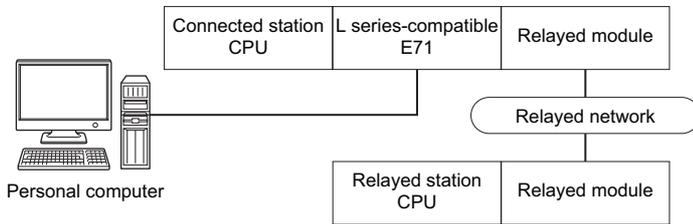
*3 When the property pattern is ○(1) or ○(2), specify the value set in the target station side parameter for ActNetworkNumber and ActStationNumber.

*4 Do not use 1 to 1024 of ActPortNumber.

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

Ethernet communication when the connected station is L series-compatible E71 (TCP)

Configuration



Accessibility and property patterns

○(n): Accessible ('n' is the property pattern number), ×: Inaccessible

Connect ed station CPU	Relayed network	Relayed station CPU				
		RCPU	RCCPU	R motion CPU	LHCPU	FX5CPU
○(1)	CC IE TSN	×	×	×	×	×
	CC IE Control CC IE Field	×	×	×	×	×
	MELSECNET/H	×	×	×	×	×
	Ethernet	×	×	×	×	×
	Serial communication	×	×	×	×	×
	CC-Link	×	×	×	×	×

Connect ed station CPU	Relayed network	Relayed station CPU					
		QCPU (Q mode)	QCCPU	LCPU	QSCPU	Q motion CPU*1	FXCPU
○(1)	CC IE TSN	×	×	×	×	×	×
	CC IE Control CC IE Field	×	×	×	×	×	×
	MELSECNET/H	×	×	×	×	×	×
	Ethernet	○(2)	×	○(2)	×	×	×
	Serial communication	○(3)*2	×	○(3)*2	×	×	×
	CC-Link	○(4)	×	○(4)	×	×	×

*1 Only Q172D, Q173D, Q172DS and Q173DS can be accessed.

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

Property list

The following table shows the property settings of communication routes.

The setting is not necessary for those properties without description.

Property	Default value	Property patterns			
		○(1)	○(2)	○(3)	○(4)
ActBaudRate	19200 (BAUDRATE_19200)	Fixed to 0 (0x00)			
ActConnectUnitNumber	0 (0x00)	Fixed to 0 (0x00)	Connected station side module station number	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActControl	8 (TRC_DTR_OR_RT S)	Fixed to 0 (0x00)			
ActCpuType	34 (CPU_Q02CPU)	CPU type corresponding to the target station			
ActDataBits	8 (DATABIT_8)	Fixed to 0 (0x00)			
ActDestinationIONumber	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station side ■For single CPU • Fixed to 1023 (0x3FF) ■For multiple CPUs • Connected CPU: 1023 (0x3FF) • No. 1: 992 (0x3E0) • No. 2: 993 (0x3E1) • No. 3: 994 (0x3E2) • No. 4: 995 (0x3E3)	Target station side ■For single CPU • Fixed to 1023 (0x3FF) ■For multiple CPUs • Connected CPU: 1023 (0x3FF) • No. 1: 992 (0x3E0) • No. 2: 993 (0x3E1) • No. 3: 994 (0x3E2) • No. 4: 995 (0x3E3) ■For redundant CPU • Control system: 976 (0x3D0) • No specification: 1023 (0x3FF)
ActDestinationPortNumber	0 (0x00)	5002 for MELSOFT connection Any port number for OPS connection*1			
ActDidPropertyBit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActDsidPropertyBit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActHostAddress	1.1.1.1	Host name or IP address of the connected station side module			
ActIONumber*2	1023 (0x3FF)	■For single CPU • Fixed to 1023 (0x3FF) ■For multiple CPUs • Connected CPU: 1023 (0x3FF) • No. 1: 992 (0x3E0) • No. 2: 993 (0x3E1) • No. 3: 994 (0x3E2) • No. 4: 995 (0x3E3) ■For redundant CPU • Control system: 976 (0x3D0) • No specification: 1023 (0x3FF)	Target station side ■For single CPU • Fixed to 1023 (0x3FF) ■For multiple CPUs • Connected CPU: 1023 (0x3FF) • No. 1: 992 (0x3E0) • No. 2: 993 (0x3E1) • No. 3: 994 (0x3E2) • No. 4: 995 (0x3E3) ■For redundant CPU • Control system: 976 (0x3D0) • No specification: 1023 (0x3FF)	Connected station side relayed module I/O address	Connected station side relayed module I/O address
ActMultiDropChannelNumber*3	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Multi-drop channel number	Fixed to 0 (0x00)
ActNetworkNumber*4	0 (0x00)	Target station side module network number	Target station side module network number	Connected station side L series-compatible E71 network number	Connected station side L series-compatible E71 network number
ActParity	1 (ODD_PARITY)	Fixed to 0 (0x00)			
ActPassword	Empty	Password set to L series-compatible E71 on the connected station side			
ActPortNumber	1 (PORT_1)	Personal computer side port number			
ActProtocolType	0x04 (PROTOCOL_SERIA L)	PROTOCOL_TCPIP (0x05)			
ActSourceNetworkNumber	0 (0x00)	Personal computer side network number			

Property	Default value	Property patterns			
		○(1)	○(2)	○(3)	○(4)
ActSourceStationNumber* ⁵	0 (0x00)	Personal computer side station number			
ActStationNumber* ⁴	255 (0xFF)	Target station side module station number	Target station side module station number	Connected station side L series-compatible E71 station number	Connected station side L series-compatible E71 station number
ActThroughNetworkType	0 (0x00)	<ul style="list-style-type: none"> • MELSECNET/10 is not included.: 0 (0x00) • MELSECNET/10 is included.: 1 (0x01) 			
ActTimeOut	10000	Any value specified by user in ms units			
ActUnitNumber	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station side module station number	Target station side module station number
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_LJ71E71 (0x5C)			

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

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

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

0: Default channel of module

1: Channel 1

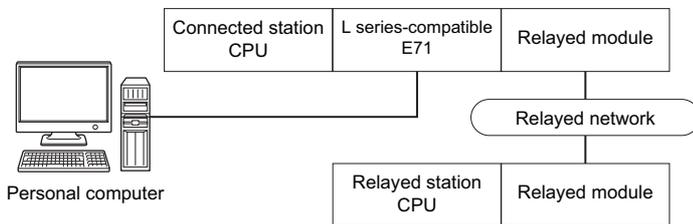
2: Channel 2

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

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

Ethernet communication when the connected station is L series-compatible E71 (UDP)

Configuration



Accessibility and property patterns

○(n): Accessible ('n' is the property pattern number), ×: Inaccessible

Connect ed station CPU	Relayed network	Relayed station CPU				
		R CPU	RCCPU	R motion CPU	LH CPU	FX5CPU
○(1)	CC IE TSN	×	×	×	×	×
	CC IE Control CC IE Field	×	×	×	×	×
	MELSECNET/H	×	×	×	×	×
	Ethernet	×	×	×	×	×
	Serial communication	×	×	×	×	×
	CC-Link	×	×	×	×	×

Connect ed station CPU	Relayed network	Relayed station CPU					
		Q CPU (Q mode)	QCCPU	L CPU	QSCPU	Q motion CPU ^{*1}	FXCPU
○(1)	CC IE TSN	×	×	×	×	×	×
	CC IE Control CC IE Field	×	×	×	×	×	×
	MELSECNET/H	×	×	×	×	×	×
	Ethernet	○(2)	×	○(2)	×	×	×
	Serial communication	○(3) ^{*2}	×	○(3) ^{*2}	×	×	×
	CC-Link	○(4)	×	○(4)	×	×	×

*1 Only Q172D, Q173D, Q172DS and Q173DS can be accessed.

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

Property list

The following table shows the property settings of communication routes.

The setting is not necessary for those properties without description.

Property	Default value	Property patterns			
		○(1)	○(2)	○(3)	○(4)
ActBaudRate	19200 (BAUDRATE_19200)	Fixed to 0 (0x00)			
ActConnectUnitNumber	0 (0x00)	Fixed to 0 (0x00)	Connected station side module station number	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActControl	8 (TRC_DTR_OR_RT S)	Fixed to 0 (0x00)			
ActCpuType	34 (CPU_Q02CPU)	CPU type corresponding to the target station			
ActDataBits	8 (DATABIT_8)	Fixed to 0 (0x00)			
ActDestinationIONumber	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station side ■For single CPU • Fixed to 1023 (0x3FF) ■For multiple CPUs • Connected CPU: 1023 (0x3FF) • No. 1: 992 (0x3E0) • No. 2: 993 (0x3E1) • No. 3: 994 (0x3E2) • No. 4: 995 (0x3E3)	Target station side ■For single CPU • Fixed to 1023 (0x3FF) ■For multiple CPUs • Connected CPU: 1023 (0x3FF) • No. 1: 992 (0x3E0) • No. 2: 993 (0x3E1) • No. 3: 994 (0x3E2) • No. 4: 995 (0x3E3) ■For redundant CPU • Control system: 976 (0x3D0) • No specification: 1023 (0x3FF)
ActDestinationPortNumber	0 (0x00)	5001			
ActDidPropertyBit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActDsidPropertyBit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActHostAddress	1.1.1.1	Host name or IP address of the connected station side module			
ActIONumber ^{*1}	1023 (0x3FF)	■For single CPU • Fixed to 1023 (0x3FF) ■For multiple CPUs • Connected CPU: 1023 (0x3FF) • No. 1: 992 (0x3E0) • No. 2: 993 (0x3E1) • No. 3: 994 (0x3E2) • No. 4: 995 (0x3E3) ■For redundant CPU • Control system: 976 (0x3D0) • No specification: 1023 (0x3FF)	Target station side ■For single CPU • Fixed to 1023 (0x3FF) ■For multiple CPUs • Connected CPU: 1023 (0x3FF) • No. 1: 992 (0x3E0) • No. 2: 993 (0x3E1) • No. 3: 994 (0x3E2) • No. 4: 995 (0x3E3) ■For redundant CPU • Control system: 976 (0x3D0) • No specification: 1023 (0x3FF)	Connected station side relayed module I/O address	Connected station side relayed module I/O address
ActMultiDropChannelNumber ^{*2}	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Multi-drop channel number	Fixed to 0 (0x00)
ActNetworkNumber ^{*3}	0 (0x00)	Target station side module network number	Target station side module network number	Connected station side L series-compatible E71 network number	Connected station side L series-compatible E71 network number
ActParity	1 (ODD_PARITY)	Fixed to 0 (0x00)			
ActPassword	Empty	Password set to L series-compatible E71 on the connected station side			
ActPortNumber ^{*4}	1 (PORT_1)	Personal computer side port number			
ActProtocolType	0x04 (PROTOCOL_SERIAL)	PROTOCOL_UDPIP (0x08)			
ActSourceNetworkNumber	0 (0x00)	Personal computer side network number			
ActSourceStationNumber ^{*5}	0 (0x00)	Personal computer side station number			

Property	Default value	Property patterns			
		○(1)	○(2)	○(3)	○(4)
ActStationNumber*3	255 (0xFF)	Target station side module station number	Target station side module station number	Connected station side L series-compatible E71 station number	Connected station side L series-compatible E71 station number
ActThroughNetworkType	0 (0x00)	<ul style="list-style-type: none"> • MELSECNET/10 is not included.: 0 (0x00) • MELSECNET/10 is included.: 1 (0x01) 			
ActTimeOut	10000	Any value specified by user in ms units			
ActUnitNumber	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station side module station number	Target station side module station number
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_LJ71E71 (0x5C)			

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

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

0: Default channel of module

1: Channel 1

2: Channel 2

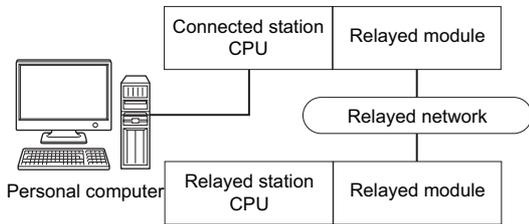
*3 When the property pattern is ○(1) or ○(2), specify the value set in the target station side parameter for ActNetworkNumber and ActStationNumber.

*4 Do not use 1 to 1024 of ActPortNumber.

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

Ethernet communication when the connected station is a MELSEC iQ-R series CPU module (TCP)

Configuration



Accessibility and property patterns

○(n): Accessible ('n' is the property pattern number), ×: Inaccessible

Connected station CPU			Relayed network	Relayed station CPU				
RCPU	RCCPU	R motion CPU		RCPU	RCCPU	R motion CPU	LHCPU	FX5CPU
○(1)	○(1) ^{*3}	○(1) ^{**1*3}	CC IE TSN	○(2)	×	×	×	×
			CC IE Control CC IE Field	○(2) ^{*2}	○(2)	×	×	○(2) ^{*5}
			MELSECNET/H	×	×	×	×	×
			Ethernet	○(2) ^{*4}	×	○(2) ^{*4}	×	×
			Serial communication	○(3) ^{*4}	×	○(3) ^{*4}	×	×
			CC-Link	○(4)	○(4)	○(4)	×	○(4)

Connected station CPU			Relayed network	Relayed station CPU					
RCPU	RCCPU	R motion CPU		QCPU (Q mode)	QCCPU	LCPU	QSCPU	Q motion CPU	FXCPU
○(1)	○(1) ^{*3}	○(1) ^{**1*3}	CC IE TSN	×	×	×	×	×	×
			CC IE Control CC IE Field	○(2) ^{*4}	×	○(2) ^{**2*4}	×	×	×
			MELSECNET/H	○(2) ^{*4}	×	×	×	×	×
			Ethernet	○(2) ^{*4}	×	○(2)	×	×	×
			Serial communication	○(3) ^{*4}	×	○(3) ^{*4}	×	×	×
			CC-Link	○(4) ^{*4}	×	○(4) ^{*4}	×	×	×

Connected station CPU		Relayed network	Relayed station CPU				
MELSECWinCPU			RCPU	RCCPU	R motion CPU	LHCPU	FX5CPU
○(1)		CC IE TSN	×	×	×	×	×
		CC IE Control CC IE Field	×	×	×	×	×
		MELSECNET/H	×	×	×	×	×
		Ethernet	×	×	×	×	×
		Serial communication	×	×	×	×	×
		CC-Link	×	×	×	×	×

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

*1 Relayed station CPUs cannot be accessed via R motion CPU.

*2 Since CC-Link IE Controller Network is not supported, RCPUs and LCPUs relayed by CC-Link IE Controller Network cannot be accessed.

*3 Only CPU No.2 or later in a multiple CPU configuration can be accessed.

*4 RCPUs cannot be accessed because the communication route is not supported.

*5 Only CC-Link IE Field Network on which connected station CPU is RCPUs can be accessed.

Connected station CPU	Relayed network	Relayed station CPU					
		QCPU (Q mode)	QCCPU	LCPU	QSCPU	Q motion CPU	FXCPU
○(1) LCPU	CC IE TSN	×	×	×	×	×	×
	CC IE Field ^{*2}	○(2)	○(2) ^{*1}	○(2)	×	×	×
	MELSECNET/H	×	×	×	×	×	×
	Ethernet	×	×	×	×	×	×
	Serial communication	○(3)	×	○(3)	×	×	×
	CC-Link	○(4)	○(4)	○(4)	×	×	×

*1 Since CC-Link IE Field Network is not supported, Q12DCCPU-V (Basic mode) relayed by CC-Link IE Field Network cannot be accessed.

*2 Since CC-Link IE Controller Network is not supported, LCPUs relayed by CC-Link IE Controller Network cannot be accessed.

- *1 For the I/O address, specify the value of the actual start I/O number divided by 16.
- *2 Specify the following value for the channel number to be multi-drop linked.
 - 0: Default channel of module
 - 1: Channel 1
 - 2: Channel 2
- *3 When the property pattern is ○(1), ○(3), or ○(4), specify the parameter value set for the CC-Link IE Field Network module on the connected station side for ActNetworkNumber and ActStationNumber.
When the property pattern is ○(2), specify the value set in the target station side parameter for ActNetworkNumber and ActStationNumber.

Connected station CPU	Relayed network	Relayed station CPU					
		QCPU (Q mode)	QCCPU	LCPU	QSCPU	Q motion CPU	FXCPU
○(1) LCPU	CC IE TSN	×	×	×	×	×	×
	CC IE Field ^{*2}	○(2)	○(2) ^{*1}	○(2)	×	×	×
	MELSECNET/H	×	×	×	×	×	×
	Ethernet	×	×	×	×	×	×
	Serial communication	○(3)	×	○(3)	×	×	×
	CC-Link	○(4)	○(4)	○(4)	×	×	×

*1 Since CC-Link IE Field Network is not supported, Q12DCCPU-V (Basic mode) relayed by CC-Link IE Field Network cannot be accessed.

*2 Since CC-Link IE Controller Network is not supported, LCPU relayed by CC-Link IE Controller Network cannot be accessed.

- *1 For the I/O address, specify the value of the actual start I/O number divided by 16.
- *2 Specify the following value for the channel number to be multi-drop linked.
 - 0: Default channel of module
 - 1: Channel 1
 - 2: Channel 2
- *3 When the property pattern is ○(1), ○(3), or ○(4), specify the parameter value set for the CC-Link IE Field Network module on the connected station side for ActNetworkNumber and ActStationNumber.
When the property pattern is ○(2), specify the value set in the target station side parameter for ActNetworkNumber and ActStationNumber.

Property	Default value	Property patterns
		○(1)
ActMultiDropChannelNumber	0 (0x00)	Fixed to 0 (0x00)
ActNetworkNumber	0 (0x00)	Fixed to 0 (0x00)
ActProtocolType	0x04 (PROTOCOL_SERIAL)	PROTOCOL_TCPIP
ActPacketType	0x01 (PACKET_PLC1)	Fixed to PACKET_PLC1
ActPortNumber	1 (PORT_1)	Fixed to 0 (0x00)
ActBaudRate	19200 (BAUDRATE_19200)	Fixed to 0 (0x00)
ActCpuTimeOut	0 (0x00)	Fixed to 0 (0x00)
ActControl	8 (TRC_DTR_OR_RTS)	Fixed to 0 (0x00)
ActDataBits	8 (DATABIT_8)	Fixed to 0 (0x00)
ActParity	1 (ODD_PARITY)	Fixed to 0 (0x00)
ActStopBits	0 (STOPBIT_ONE)	Fixed to 0 (0x00)
ActSumCheck	0 (NO_SUM_CHECK)	Fixed to 0 (0x00)
ActIntelligentPreferenceBit	0 (0x00)	Fixed to 0 (0x00)
ActSourceNetworkNumber	0 (0x00)	Fixed to 0 (0x00)
ActSourceStationNumber	0 (0x00)	Fixed to 0 (0x00)
ActStationNumber	255 (0xFF)	Fixed to 255 (0xFF)
ActThroughNetworkType	0 (0x00)	Fixed to 0 (0x00)
ActTimeOut	10000	Any value specified by user in ms units
ActUnitNumber	0 (0x00)	Fixed to 0 (0x00)
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_FXVENET

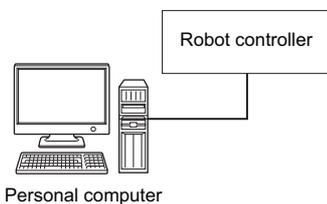
Property	Default value	Property patterns
		○(1)
ActMultiDropChannelNumber	0 (0x00)	Fixed to 0 (0x00)
ActNetworkNumber	0 (0x00)	Fixed to 0 (0x00)
ActProtocolType	0x04 (PROTOCOL_SERIAL)	PROTOCOL_UDPIP
ActPacketType	0x01 (PACKET_PLC1)	Fixed to PACKET_PLC1
ActPortNumber	1 (PORT_1)	Fixed to 0 (0x00)
ActBaudRate	19200 (BAUDRATE_19200)	Fixed to 0 (0x00)
ActCpuTimeOut	0 (0x00)	Fixed to 0 (0x00)
ActControl	8 (TRC_DTR_OR_RTS)	Fixed to 0 (0x00)
ActDataBits	8 (DATABIT_8)	Fixed to 0 (0x00)
ActParity	1 (ODD_PARITY)	Fixed to 0 (0x00)
ActStopBits	0 (STOPBIT_ONE)	Fixed to 0 (0x00)
ActSumCheck	0 (NO_SUM_CHECK)	Fixed to 0 (0x00)
ActIntelligentPreferenceBit	0 (0x00)	Fixed to 0 (0x00)
ActSourceNetworkNumber	0 (0x00)	Fixed to 0 (0x00)
ActSourceStationNumber	0 (0x00)	Fixed to 0 (0x00)
ActStationNumber	255 (0xFF)	Fixed to 255 (0xFF)
ActThroughNetworkType	0 (0x00)	Fixed to 0 (0x00)
ActTimeOut	10000	Any value specified by user in ms units
ActUnitNumber	0 (0x00)	Fixed to 0 (0x00)
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_FXVENET

Property	Default value	Property patterns
		○(1)
ActIONumber	1023 (0x3FF)	Fixed to 1023 (0x3FF)
ActMultiDropChannelNumber	0 (0x00)	Fixed to 0 (0x00)
ActNetworkNumber	0 (0x00)	Fixed to 0 (0x00)
ActProtocolType	0x04 (PROTOCOL_SERIAL)	PROTOCOL_UDPIP
ActPacketType	0x01 (PACKET_PLC1)	Fixed to PACKET_PLC1
ActPortNumber	1 (PORT_1)	Fixed to 0 (0x00)
ActBaudRate	19200 (BAUDRATE_19200)	Fixed to 0 (0x00)
ActCpuTimeOut	0 (0x00)	Fixed to 0 (0x00)
ActControl	8 (TRC_DTR_OR_RTS)	Fixed to 0 (0x00)
ActDataBits	8 (DATABIT_8)	Fixed to 0 (0x00)
ActParity	1 (ODD_PARITY)	Fixed to 0 (0x00)
ActStopBits	0 (STOPBIT_ONE)	Fixed to 0 (0x00)
ActSumCheck	0 (NO_SUM_CHECK)	Fixed to 0 (0x00)
ActIntelligentPreferenceBit	0 (0x00)	Fixed to 0 (0x00)
ActSourceNetworkNumber	0 (0x00)	Fixed to 0 (0x00)
ActSourceStationNumber	0 (0x00)	Fixed to 0 (0x00)
ActStationNumber	255 (0xFF)	Fixed to 255 (0xFF)
ActThroughNetworkType	0 (0x00)	Fixed to 0 (0x00)
ActTimeOut	10000	Any value specified by user in ms units
ActUnitNumber	0 (0x00)	Fixed to 0 (0x00)
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_FXVENET_DIRECT

Ethernet communication when the connected station is a robot controller

Configuration

10



Property list

The following table shows the property settings of communication routes.

The setting is not necessary for those properties without description.

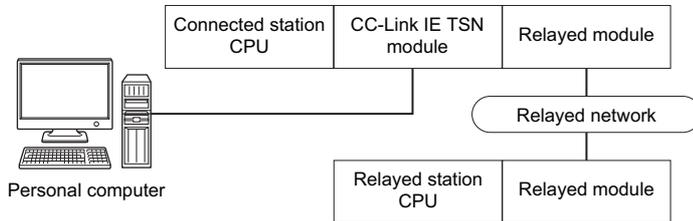
Property	Default value	Property patterns
ActProtocolType	0x04 (PROTOCOL_SERIAL)	RC_PROTOCOL_TCPIP (0x02)
ActCpuType	34 (CPU_Q02CPU)	Robot controller type (0x013001)
ActPortNumber	1 (PORT_1)	Port number to be connected (specify 10001 normally)
ActHostAddress	1.1.1.1	Host name or IP address of the connected station side robot controller
ActTimeOut	10000	Any value specified by user in ms units
ActCpuTimeOut	0 (0x00)	Any value specified by user in 10ms units
ActMultiDropChannelNumber	8 (TRC_DTR_OR_RTS)	Retry count
ActMxUnitSeries	0 (0x00)	2 (0x02)

Point

When robot controller communication, the program setting type control cannot be used.
Use the utility setting type control.

Ethernet communication when the connected station is a CC-Link IE TSN module (TCP)

Configuration



Accessibility and property patterns

○(n): Accessible ('n' is the property pattern number), ×: Inaccessible

Connected station CPU	Relayed network	Relayed station CPU				
		RCPU	RCCPU	R motion CPU	LHCPU	FX5CPU
○(1)	CC IE TSN	○(2)	×	×	×	×
	CC IE Control CC IE Field	×	×	×	×	×
	MELSECNET/H	×	×	×	×	×
	Ethernet	×	×	×	×	×
	Serial communication	○(3)	×	×	×	×
	CC-Link	○(4)	×	×	×	×

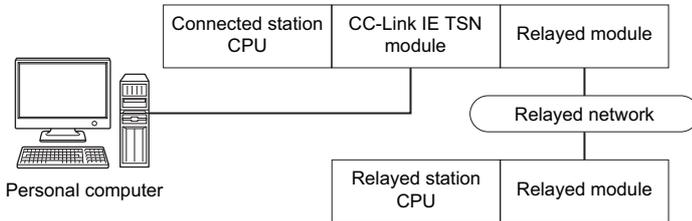
Connected station CPU	Relayed network	Relayed station CPU					
		QCPU (Q mode)	QCCPU	LCPU	QSCPU	Q motion CPU	FXCPU
○(1)	CC IE TSN	×	×	×	×	×	×
	CC IE Control CC IE Field	×	×	×	×	×	×
	MELSECNET/H	×	×	×	×	×	×
	Ethernet	×	×	×	×	×	×
	Serial communication	×	×	×	×	×	×
	CC-Link	×	×	×	×	×	×

Property	Default value	Property patterns			
		○(1)	○(2)	○(3)	○(4)
ActStationNumber ^{*16}	255 (0xFF)	Target station side module station number	Target station side module station number	Connected station side CC-Link IE TSN module station number	Connected station side CC-Link IE TSN module station number
ActThroughNetworkType	0 (0x00)	Fixed to 0 (0x00)			
ActTimeOut	10000	Any value specified by user in ms units			
ActUnitNumber	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station side module station number	Target station side module station number
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_RJ71GN11 (0x1006)			

- *1 When the property pattern is ○(1) or ○(2), specify the value set in the target station side parameter for ActNetworkNumber and ActStationNumber.
- *2 Specify the following value.
0: An empty slot number in a personal computer is automatically assigned.
- *3 Specify the personal computer side station number to avoid the same station number set for CC-Link IE TSN module within the same Ethernet loop.
- *4 Specify the following value for the channel number to be multi-drop linked.
0: Default channel of module
1: Channel 1
2: Channel 2
- *5 When the target station is a motion controller or a robot controller, set the value for multiple CPUs.
Example: When a robot controller is set as the CPU No.2, specify '0x3E1'.
- *6 Note the following considerations when accessing via CC-Link IE TSN module.
·For ActNetworkNumber and ActStationNumber, specify the value set in the parameter setting of the target station side CC-Link IE TSN module.
·Set "Station No. ↔ IP information" in the parameter setting of CC-Link IE TSN module. When setting the parameter, specify the setting other than the automatic response system (any of the IP address calculation system, table conversion system, and combined system) for "Station No. ↔ IP information system."

Ethernet communication when the connected station is a CC-Link IE TSN module (UDP)

Configuration



Accessibility and property patterns

○(n): Accessible ('n' is the property pattern number), ×: Inaccessible

Connected station CPU	Relayed network	Relayed station CPU				
		RCPU	RCCPU	R motion CPU	LHCPU	FX5CPU
○(1)*1	CC IE TSN	○(2)	×	×	×	×
	CC IE Control CC IE Field	×	×	×	×	×
	MELSECNET/H	×	×	×	×	×
	Ethernet	×	×	×	×	×
	Serial communication	○(3)	×	×	×	×
	CC-Link	○(4)	×	×	×	×

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

*1 Access to a CC-Link IE TSN module with a direct connection is not supported.

Property	Default value	Property patterns			
		○(1)	○(2)	○(3)	○(4)
ActStationNumber ^{*16}	255 (0xFF)	Target station side module station number	Target station side module station number	Connected station side CC-Link IE TSN module station number	Connected station side CC-Link IE TSN module station number
ActThroughNetworkType	0 (0x00)	Fixed to 0 (0x00)			
ActTimeOut	10000	Any value specified by user in ms units			
ActUnitNumber	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station side module station number	Target station side module station number
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_RJ71GN11 (0x1006)			

- *1 When the property pattern is ○(1) or ○(2), specify the value set in the target station side parameter for ActNetworkNumber and ActStationNumber.
- *2 When other than the 'automatic response system' is set in the Ethernet parameter of the connected station side CC-Link IE TSN module, fix the value to '5001'.
When the 'automatic response system' is set in the Ethernet parameter of the connected station side CC-Link IE TSN module, specify the following value.
0: An empty slot number in a personal computer is automatically assigned.
Other than '0': a UDP socket is generated with the specified port number.
- *3 Specify the personal computer side station number to avoid the same station number set for CC-Link IE TSN module within the same Ethernet loop.
- *4 Specify the following value for the channel number to be multi-drop linked.
0: Default channel of module
1: Channel 1
2: Channel 2
- *5 When the target station is a motion controller or a robot controller, set the value for multiple CPUs.
Example: When a robot controller is set as the CPU No.2, specify '0x3E1'.
- *6 Note the following considerations when accessing via CC-Link IE TSN module.
·For ActNetworkNumber and ActStationNumber, specify the value set in the parameter setting of the target station side CC-Link IE TSN module.
·Set "Station No. ↔ IP information" in the parameter setting of CC-Link IE TSN module. When setting the parameter, specify the setting other than the automatic response system (any of the IP address calculation system, table conversion system, and combined system) for "Station No. ↔ IP information system."

Property	Default value	Property patterns			
		○(1)	○(2) ^{*1}	○(3)	○(4)
ActTimeOut	10000	Any value specified by user in ms units			
ActUnitNumber	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station side module station number	Target station side module station number
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_QNCPU (0x13)			

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

- For ActNetworkNumber and ActStationNumber, specify the value set in the parameter setting of the target station side Q series-compatible E71.

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

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

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

0: Default channel of module

1: Channel 1

2: Channel 2

Property list

The following table shows the property settings of communication routes.

The setting is not necessary for those properties without description.

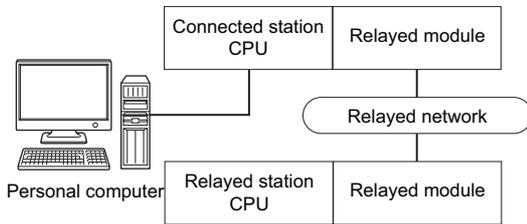
Property	Default value	Property patterns			
		○(1)	○(2)	○(3)	○(4)
ActBaudRate	19200 (BAUDRATE_19200)	BAUDRATE_9600, BAUDRATE_19200, BAUDRATE_38400, BAUDRATE_57600, BAUDRATE_115200			
ActConnectUnitNumber	0 (0x00)	Fixed to 0 (0x00)			
ActControl	8 (TRC_DTR_OR_RT S)	Depending on the used cable.			
ActCpuTimeOut	0 (0x00)	Fixed to 0 (0x00)			
ActCpuType	34 (CPU_Q02CPU)	CPU type corresponding to the target station			
ActDataBits	8 (DATABIT_8)	Fixed to 8 (DATABIT_8)			
ActDestinationIONumber	0 (0x00)	Fixed to 0 (0x00)	Target station side ■For single CPU • Fixed to 1023 (0x3FF) ■For multiple CPUs • Connected CPU: 1023 (0x3FF) • No. 1: 992 (0x3E0) • No. 2: 993 (0x3E1) • No. 3: 994 (0x3E2) • No. 4: 995 (0x3E3) ■For redundant CPU • Control system: 976 (0x3D0) • No specification: 1023 (0x3FF)	Target station side ■For single CPU • Fixed to 1023 (0x3FF) ■For multiple CPUs • Connected CPU: 1023 (0x3FF) • No. 1: 992 (0x3E0) • No. 2: 993 (0x3E1) • No. 3: 994 (0x3E2) • No. 4: 995 (0x3E3) ■For redundant CPU • Control system: 976 (0x3D0) • No specification: 1023 (0x3FF)	Fixed to 0 (0x00)
ActDestinationPortNumber	0 (0x00)	Fixed to 0 (0x00)			
ActDidPropertyBit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Fixed to 1 (0x01)
ActDsidPropertyBit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Fixed to 1 (0x01)
ActHostAddress	1.1.1.1	Fixed to NULL			
ActIntelligentPreferenceBit	0 (0x00)	Fixed to 0 (0x00)	Target station • QCPU (Q mode), LCPU: 1 (0x01) • Other than the above: 0 (0x00)	Target station • QCPU (Q mode), QCCPU, LCPU: 1 (0x01) • Other than the above: 0 (0x00)	Fixed to 0 (0x00)
ActIONumber ^{*1}	1023 (0x3FF)	Fixed to 1023 (0x3FF)	Connected station side module I/O address	Connected station side module I/O address	Target station side ■For single CPU • Fixed to 1023 (0x3FF) ■For multiple CPUs • Connected CPU: 1023 (0x3FF) • No. 1: 992 (0x3E0) • No. 2: 993 (0x3E1) • No. 3: 994 (0x3E2) • No. 4: 995 (0x3E3) ■For redundant CPU • Control system: 976 (0x3D0) • No specification: 1023 (0x3FF)
ActMultiDropChannelNumber ^{*2}	0 (0x00)	Fixed to 0 (0x00)	Multi-drop channel number	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActNetworkNumber	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station side module network number
ActPacketType	0x01 (PACKET_PLC1)	Fixed to 0x01 (PACKET_PLC1)			

Property	Default value	Property patterns			
		○(1)	○(2)	○(3)	○(4)
ActParity	1 (ODD_PARITY)	Fixed to 1 (ODD_PARITY)			
ActPortNumber	1 (PORT_1)	Personal computer side COM port number			
ActProtocolType	0x04 (PROTOCOL_SERIAL)	PROTOCOL_SERIAL (0x04)			
ActSourceNetworkNumber	0 (0x00)	Fixed to 0 (0x00)			
ActSourceStationNumber	0 (0x00)	Fixed to 0 (0x00)			
ActStationNumber	255 (0xFF)	Fixed to 255 (0xFF)	Fixed to 255 (0xFF)	Fixed to 255 (0xFF)	Target station side module station number
ActStopBits	0 (STOPBIT_ONE)	Fixed to 0 (STOPBIT_ONE)			
ActSumCheck	0 (NO_SUM_CHECK)	Fixed to 0 (NO_SUM_CHECK)			
ActThroughNetworkType	0 (0x00)	<ul style="list-style-type: none"> • MELSECNET/10 is not included.: 0 (0x00) • MELSECNET/10 is included.: 1 (0x01) 			
ActTimeOut	10000	Any value specified by user in ms units			
ActUnitNumber	0 (0x00)	Fixed to 0 (0x00)	Target station side module station number	Target station side module station number	Fixed to 0 (0x00)
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_LNCPUR (0x50)			

- *1 For the I/O address, specify the value of the actual start I/O number divided by 16.
- *2 Specify the following value for the channel number to be multi-drop linked.
 - 0: Default channel of module
 - 1: Channel 1
 - 2: Channel 2

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

Configuration



Accessibility and property patterns

○(n): Accessible ('n' is the property pattern number), ×: Inaccessible

Connected station CPU	Relayed network	Relayed station CPU				
		RCPU	RCCPU	R motion CPU	LHCPU	FX5CPU
○(1) ^{*1}	CC IE TSN	×	×	×	×	×
	CC IE Control CC IE Field	×	×	×	×	×
	MELSECNET/H	×	×	×	×	×
	Ethernet	×	×	×	×	×
	Serial communication	×	×	×	×	×
	CC-Link	×	×	×	×	×

Connected station CPU	Relayed network	Relayed station CPU					
		QCPU (Q mode)	QCCPU	LCPU	QSCPU	Q motion CPU	FXCPU
○(1) ^{*1}	CC IE TSN	×	×	×	×	×	×
	CC IE Control CC IE Field	×	×	×	×	×	×
	MELSECNET/H	×	×	×	×	×	×
	Ethernet	×	×	×	×	×	×
	Serial communication	×	×	×	×	×	×
	CC-Link	×	×	×	×	×	×

*1 Only Q172D, Q173D, Q172DS and Q173DS can be accessed.

Property list

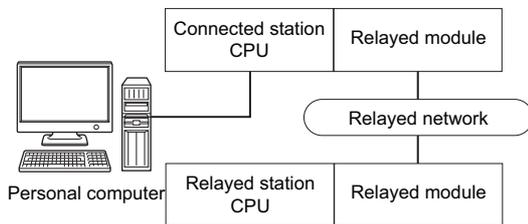
The following table shows the property settings of communication routes.

The setting is not necessary for those properties without description.

Property	Default value	Property patterns
		○(1)
ActBaudRate	19200 (BAUDRATE_19200)	BAUDRATE_9600, BAUDRATE_19200, BAUDRATE_38400, BAUDRATE_57600, BAUDRATE_115200
ActControl	8 (TRC_DTR_OR_RTS)	Depending on the used cable.
ActCpuType	34 (CPU_Q02CPU)	CPU type corresponding to the target station
ActIONumber	1023 (0x3FF)	<ul style="list-style-type: none"> ■ For multiple CPUs <ul style="list-style-type: none"> • Control CPU: 1023 (0x3FF) • No. 2: 993 (0x3E1) • No. 3: 994 (0x3E2) • No. 4: 995 (0x3E3)
ActPortNumber	1 (PORT_1)	Personal computer side COM port number
ActProtocolType	0x04 (PROTOCOL_SERIAL)	PROTOCOL_SERIAL (0x04)
ActTimeOut	10000	Any value specified by user in ms units
ActUnitType	0x13 (UNIT_QNCP)	UNIT_QNMOTION (0x1C)

CPU COM communication when the connected station is an FXCPU

Configuration



Accessibility and property patterns

○(n): Accessible ('n' is the property pattern number), ×: Inaccessible

Connected station CPU	Relayed network	Relayed station CPU				
		RCPU	RCCPU	R motion CPU	LHCPU	FX5CPU
○(1)	CC IE TSN	×	×	×	×	×
	CC IE Control CC IE Field	×	×	×	×	×
	MELSECNET/H	×	×	×	×	×
	Ethernet	×	×	×	×	×
	Serial communication	×	×	×	×	×
	CC-Link	×	×	×	×	×

Connected station CPU	Relayed network	Relayed station CPU					
		QCPU (Q mode)	QCCPU	LCPU	QSCPU	Q motion CPU	FXCPU
○(1)	CC IE TSN	×	×	×	×	×	×
	CC IE Control CC IE Field	×	×	×	×	×	×
	MELSECNET/H	×	×	×	×	×	×
	Ethernet	×	×	×	×	×	×
	Serial communication	×	×	×	×	×	×
	CC-Link	×	×	×	×	×	○(2) ^{*1}

*1 Only FX3G(C)CPU and FX3U(C)CPU can be accessed.

Property list

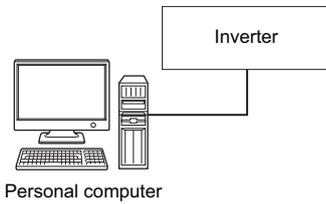
The following table shows the property settings of communication routes.

The setting is not necessary for those properties without description.

Property	Default value	Property patterns	
		○(1)	○(2)
ActBaudRate	19200 (BAUDRATE_19200)	• FX3S, FX3UC, FX3G(C), FX5U, FX5UJ: (BAUDRATE_9600, BAUDRATE_19200, BAUDRATE_38400, BAUDRATE_57600, BAUDRATE_115200)	
ActControl	8 (TRC_DTR_OR_RTS)	Depending on the used cable.	
ActCpuType	34 (CPU_Q02CPU)	CPU type corresponding to the target station	
ActDestinationIONumber	0 (0x00)	Fixed to 0 (0x00)	1023 (0x3FF)
ActDidPropertyBit	1 (0x01)	0 (0x00)	
ActDsidPropertyBit	1 (0x01)	0 (0x00)	
ActIONumber	1023 (0x3FF)	Fixed to 0 (0x00)	Module number of the connected station
ActPortNumber	1 (PORT_1)	Personal computer side COM port number	
ActProtocolType	0x04 (PROTOCOL_SERIAL)	PROTOCOL_SERIAL (0x04)	
ActTimeOut	10000	Any value specified by user in ms units	
ActUnitNumber	0 (0x00)	Fixed to 0 (0x00)	Target station side module station number
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_FXCPU (0x0F)	

CPU COM communication when the connected station is an inverter

Configuration



Property list

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

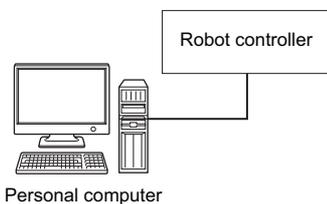
Property	Default value	Property patterns
ActMxUnitSeries	0 (0x00)	1 (0x01)
ActProtocolType	0x04 (PROTOCOL_SERIAL)	COMM_RS232C (0x00)
ActStationNumber	255 (0xFF)	Inverter station number (0 to 31)
ActCpuType	34 (CPU_Q02CPU)	INV_A800 (0x1E60)
ActPortNumber	1 (PORT_1)	Personal computer side COM port number
ActBaudRate	19200 (BAUDRATE_19200)	BAUDRATE_4800, BAUDRATE_9600, BAUDRATE_19200, BAUDRATE_38400, BAUDRATE_57600, BAUDRATE_115200
ActTimeOut	10000	Any value specified by user in ms units
ActCpuTimeOut	0 (0x00)	Any value specified by user in 10ms units
ActPacketType	0x01 (PACKET_PL1)	<ul style="list-style-type: none">• CRLF_NONE (0): Without CR/LF• CRLF_CR (1): With CR• CRLF_CRLF (2): With CR/LF
ActDataBits	8 (DATABIT_8)	<ul style="list-style-type: none">• DATABIT_7• DATABIT_8
ActParity	1 (ODD_PARITY)	<ul style="list-style-type: none">• NO_PARRITY• ODD_PARITY• EVEN_PARITY
ActStopBits	0 (STOPBIT_ONE)	<ul style="list-style-type: none">• STOPBIT_ONE• STOPBIT_TWO

Point

When performing inverter communication, the program setting type control cannot be used.
Use the utility setting type control.

CPU COM communication when the connected station is a robot controller

Configuration



Property list

The following table shows the property settings of communication routes.

The setting is not necessary for those properties without description.

Property	Default value	Property patterns
ActProtocolType	0x04 (PROTOCOL_SERIAL)	RC_PROTOCOL_SERIAL (0x01)
ActCpuType	34 (CPU_Q02CPU)	Robot controller type (0x013001)
ActPortNumber	1 (PORT_1)	Personal computer side COM port number
ActBaudRate	19200 (BAUDRATE_19200)	Transmission speed
ActTimeOut	10000	Receive timeout time (msec)
ActCpuTimeOut	0 (0x00)	Send timeout time (msec)
ActDataBits	8 (DATABIT_8)	<ul style="list-style-type: none">• DATABIT_7• DATABIT_8
ActParity	1 (ODD_PARITY)	<ul style="list-style-type: none">• NO_PARRITY• ODD_PARITY• EVEN_PARITY
ActStopBits	0 (STOPBIT_ONE)	<ul style="list-style-type: none">• ONESTOPBIT(0)• ONE5STOPBITS(1)• TWOSTOPBITS(2)
ActMultiDropChannelNumber	0 (0x00)	Retry count
ActThroughNetworkType	0 (0x00)	Procedural/Nonprocedural (0/1)
ActMxUnitSeries	0 (0x00)	2 (0x02)

Point

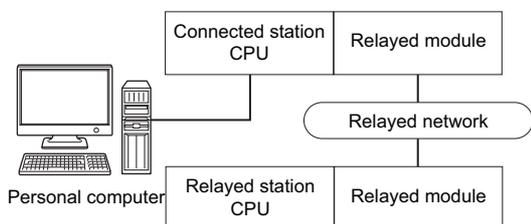
When robot controller communication, the program setting type control cannot be used.

Use the utility setting type control.

10.4 CPU USB Communication

CPU USB communication when the connected station is an RCPU

Configuration



Accessibility and property patterns

○(n): Accessible ('n' is the property pattern number), ×: Inaccessible

Connected station CPU RCPU	Relayed network	Relayed station CPU				
		RCPU	RCCPU	R motion CPU	LHCPU	FX5CPU
○(1)	CC IE TSN	○(2)	×	×	×	×
	CC IE Control CC IE Field	○(2) ^{*1}	○(2)	×	×	○(2) ^{*2}
	MELSECNET/H	×	×	×	×	×
	Ethernet	○(2)	×	○(2)	×	×
	Serial communication	○(3)	×	○(3)	×	×
	CC-Link	○(4)	○(4)	○(4)	×	○(4)

Connected station CPU RCPU	Relayed network	Relayed station CPU					
		QCPU (Q mode)	QCCPU	LCPU	QSCPU	Q motion CPU	FXCPU
○(1)	CC IE TSN	×	×	×	×	×	×
	CC IE Control CC IE Field	○(2)	×	○(2) ^{*1}	×	×	×
	MELSECNET/H	×	×	×	×	×	×
	Ethernet	○(2)	×	○(2)	×	×	×
	Serial communication	○(3)	×	○(3)	×	×	×
	CC-Link	○(4)	×	○(4)	×	×	×

*1 Since CC-Link IE Controller Network is not supported, RCPU and LCPU relayed by CC-Link IE Controller Network cannot be accessed.

*2 Only CC-Link IE Field Network can be accessed.

Property	Default value	Property patterns			
		○(1)	○(2) ^{*1}	○(3)	○(4)
ActDataBits	8 (DATABIT_8)	Fixed to 0 (0x00)			
ActParity	1 (ODD_PARITY)	Fixed to 0 (0x00)			
ActStopBits	0 (STOPBIT_ONE)	Fixed to 0 (0x00)			
ActSumCheck	0 (NO_SUM_CHECK)	Fixed to 0 (0x00)			
ActSourceNetworkNumber	0 (0x00)	Fixed to 0 (0x00)			
ActSourceStationNumber	0 (0x00)	Fixed to 0 (0x00)			
ActDestinationPortNumber	0 (0x00)	Fixed to 0 (0x00)			
ActThroughNetworkType	0 (0x00)	<ul style="list-style-type: none"> • MELSECNET/10 is not included.: 0 (0x00) • MELSECNET/10 is included.: 1 (0x01) 			MELSECNET/10 is not included.: 0 (0x00)
ActTimeOut	10000	Any value specified by user in ms units			
ActUnitNumber	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station side module station number	Target station side module station number
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_RUSB			

*1 Note the following considerations when accessing via Ethernet module (MELSEC iQ-R series-compatible E71).

- For ActNetworkNumber and ActStationNumber, specify the value set in the parameter setting of the target station side MELSEC iQ-R series-compatible E71.

- Set "Station No. ⇔ IP information" in the parameter setting of MELSEC iQ-R series-compatible E71. When setting the parameter, specify the setting other than the automatic response system (any of the IP address calculation system, table conversion system, and combined system) for "Station No. ⇔ IP information system."

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

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

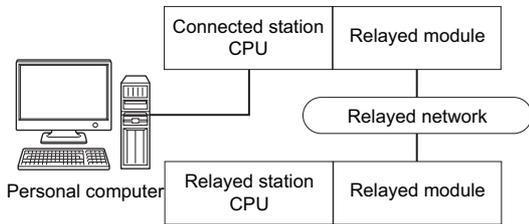
0: Default channel of module

1: Channel 1

2: Channel 2

CPU USB communication when the connected station is an R motion CPU

Configuration



Accessibility and property patterns

○(n): Accessible ('n' is the property pattern number), ×: Inaccessible

Connected station CPU	Relayed network	Relayed station CPU				
		R CPU	RCCPU	R motion CPU	LH CPU	FX5 CPU
○(1) ^{*1*2}	CC IE TSN	×	×	×	×	×
	CC IE Control CC IE Field	×	×	×	×	×
	MELSECNET/H	×	×	×	×	×
	Ethernet	×	×	×	×	×
	Serial communication	×	×	×	×	×
	CC-Link	×	×	×	×	×

Connected station CPU	Relayed network	Relayed station CPU					
		Q CPU (Q mode)	QCCPU	L CPU	QSCPU	Q motion CPU	FX CPU
○(1) ^{*1*2}	CC IE TSN	×	×	×	×	×	×
	CC IE Control CC IE Field	×	×	×	×	×	×
	MELSECNET/H	×	×	×	×	×	×
	Ethernet	×	×	×	×	×	×
	Serial communication	×	×	×	×	×	×
	CC-Link	×	×	×	×	×	×

*1 Relayed station CPUs cannot be accessed via R motion CPU.
 *2 Only CPU No.2 or later in a multiple CPU configuration can be accessed.

Property list

The following table shows the property settings of communication routes.

The setting is not necessary for those properties without description.

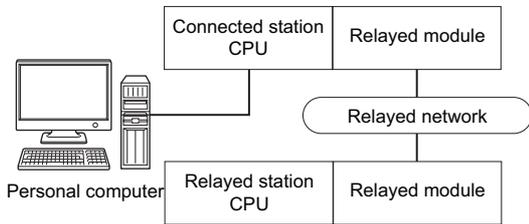
Property	Default value	Property patterns
		○(1)
ActCpuType	34 (CPU_Q02CPU)	CPU type corresponding to the target station
Act DestinationIONumber	0 (0x00)	Fixed to 0 (0x00)
ActDidPropertyBit	1 (0x01)	Fixed to 1 (0x01)
ActDsidPropertyBit	1 (0x01)	Fixed to 1 (0x01)
Act IntelligentPreference Bit	0 (0x00)	Fixed to 0 (0x00)
ActIONumber	1023 (0x3FF)	<ul style="list-style-type: none"> ■ For single CPU <ul style="list-style-type: none"> • Fixed to 1023 (0x3FF) ■ For multiple CPUs <ul style="list-style-type: none"> • Connected CPU: 1023 (0x3FF) • No. 1: 992 (0x3E0) • No. 2: 993 (0x3E1) • No. 3: 994 (0x3E2) • No. 4: 995 (0x3E3) ■ For redundant CPU <ul style="list-style-type: none"> • No specification: 1023 (0x3FF)
ActMultiDropChannelNumber*1	0 (0x00)	Fixed to 0 (0x00)
ActNetworkNumber	0 (0x00)	Fixed to 0 (0x00)
ActProtocolType	0x04 (PROTOCOL_SERIAL)	PROTOCOL_USB
ActStationNumber	255 (0xFF)	Fixed to 255 (0xFF)
ActThroughNetworkType	0 (0x00)	<ul style="list-style-type: none"> • MELSECNET/10 is not included.: 0 (0x00) • MELSECNET/10 is included.: 1 (0x01)
ActTimeOut	10000	Any value specified by user in ms units
ActUnitNumber	0 (0x00)	Fixed to 0 (0x00)
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_RUSB

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

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

CPU USB communication when the connected station is an LHCPU

Configuration



Accessibility and property patterns

○(n): Accessible ('n' is the property pattern number), ×: Inaccessible

Connected station CPU	Relayed network	Relayed station CPU				
		RCPU	RCCPU	R motion CPU	LHCPU	FX5CPU
○(1)	CC IE TSN	×	×	×	×	×
	CC IE Control CC IE Field	×	×	×	×	×
	MELSECNET/H	×	×	×	×	×
	Ethernet	×	×	×	×	×
	Serial communication	×	×	×	×	×
	CC-Link	×	×	×	×	×

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

Property list

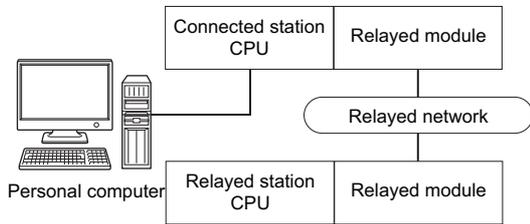
The following table shows the property settings of communication routes.

The setting is not necessary for those properties without description.

Property	Default value	Property patterns
		○(1)
ActCpuType	34 (CPU_Q02CPU)	CPU type corresponding to the target station
ActDestinationIONumber	0 (0x00)	Fixed to 0 (0x00)
ActDidPropertyBit	1 (0x01)	Fixed to 1 (0x01)
ActDsidPropertyBit	1 (0x01)	Fixed to 1 (0x01)
ActIntelligentPreferenceBit	0 (0x00)	Fixed to 0 (0x00)
ActIONumber	1023 (0x3FF)	■For single CPU Fixed to 1023 (0x3FF)
ActMultiDropChannelNumber	0 (0x00)	Fixed to 0 (0x00)
ActNetworkNumber	0 (0x00)	Fixed to 0 (0x00)
ActProtocolType	0x04 (PROTOCOL_SERIAL)	PROTOCOL_USB
ActPacketType	0x01 (PACKET_PLC1)	PACKET_PLC1
ActStationNumber	255 (0xFF)	Fixed to 255 (0xFF)
ActConnectUnitNumber	0 (0x00)	Fixed to 0 (0x00)
ActPortNumber	1 (PORT_1)	Fixed to 1 (PORT_1)
ActBaudRate	19200 (BAUDRATE_19200)	Fixed to 0 (0x00)
ActHostAddress	1.1.1.1	Fixed to NULL
ActCpuTimeOut	0 (0x00)	Fixed to 0 (0x00)
ActControl	8 (TRC_DTR_OR_RTS)	Fixed to 0 (0x00)
ActDataBits	8 (DATABIT_8)	Fixed to 0 (0x00)
ActParity	1 (ODD_PARITY)	Fixed to 0 (0x00)
ActStopBits	0 (STOPBIT_ONE)	Fixed to 0 (0x00)
ActSumCheck	0 (NO_SUM_CHECK)	Fixed to 0 (0x00)
ActSourceNetworkNumber	0 (0x00)	Fixed to 0 (0x00)
ActSourceStationNumber	0 (0x00)	Fixed to 0 (0x00)
ActDestinationPortNumber	0 (0x00)	Fixed to 0 (0x00)
ActThroughNetworkType	0 (0x00)	<ul style="list-style-type: none"> • MELSECNET/10 is not included.: 0 (0x00) • MELSECNET/10 is included.: 1 (0x01)
ActTimeOut	10000	Any value specified by user in ms units
ActUnitNumber	0 (0x00)	Fixed to 0 (0x00)
ActUnitType	0x13 (UNIT_QNCP)	UNIT_LHUSB

CPU USB communication when the connected station is an FX5CPU

Configuration



Accessibility and property patterns

○(n): Accessible ('n' is the property pattern number), ×: Inaccessible

Connected station CPU	Relayed network	Relayed station CPU				
		RCPU	RCCPU	R motion CPU	LHCPU	FX5CPU
○(1) ^{*1}	CC IE TSN	×	×	×	×	×
	CC IE Control CC IE Field	×	×	×	×	○(2) ^{*2*3}
	MELSECNET/H	×	×	×	×	×
	Ethernet	×	×	×	×	×
	Serial communication	×	×	×	×	×
	CC-Link	×	×	×	×	○(3) ^{*3}

Connected station CPU	Relayed network	Relayed station CPU					
		QCPU (Q mode)	QCCPU	LCPU	QSCPU	Q motion CPU	FXCPU
○(1) ^{*1}	CC IE TSN	×	×	×	×	×	×
	CC IE Control CC IE Field	×	×	×	×	×	×
	MELSECNET/H	×	×	×	×	×	×
	Ethernet	×	×	×	×	×	×
	Serial communication	×	×	×	×	×	×
	CC-Link	×	×	×	×	×	×

*1 Only FX5UJCPU and FX5SCPU can be accessed.

*2 Only CC-Link IE Field Network can be accessed.

*3 FX5SCPU cannot be accessed.

Property list

The following table shows the property settings of communication routes.

The setting is not necessary for those properties without description.

Property	Default value	Property patterns		
		○(1)	○(2)	○(3)
ActBaudRate	19200 (BAUDRATE_19200)	0 (0x00)		
ActConnectUnitNumber	0 (0x00)	Fixed to 0 (0x00)		
ActControl	8 (TRC_DTR_OR_RTS)	0 (0x00)		
ActCpuTimeOut	0 (0x00)	Fixed to 0 (0x00)		
ActCpuType	34 (CPU_Q02CPU)	CPU type corresponding to the target station		
ActDataBits	8 (DATABIT_8)	0 (0x00)		
ActDestinationIONumber	0 (0x00)	Fixed to 0 (0x00)		1023 (0x3FF)
ActDestinationPortNumber	0 (0x00)	Fixed to 0 (0x00)		
ActDidPropertyBit	1 (0x01)	0 (0x00)	1 (0x01)	0 (0x00)
ActDsidPropertyBit	1 (0x01)	0 (0x00)	1 (0x01)	0 (0x00)
ActHostAddress	1.1.1.1	0 (0x00)		
ActIntelligentPreferenceBit	0 (0x00)	0 (0x00)		1 (0x01)
ActIONumber	1023 (0x3FF)	1023 (0x3FF)		Module number of the connected station
ActNetworkNumber	0 (0x00)	0 (0x00)	Target station side module network number	0 (0x00)
ActPacketType	0x01 (PACKET_PL_C1)	Fixed to 0x01 (PACKET_PL_C1)		
ActPassword	Empty	NULL		
ActPortNumber	1 (PORT_1)	Fixed to 1 (PORT_1)		
ActProtocolType	0x04 (PROTOCOL_SERIAL)	PROTOCOL_USB		
ActSourceNetworkNumber	0 (0x00)	0 (0x00)		
ActSourceStationNumber	0 (0x00)	0 (0x00)		
ActStationNumber	255 (0xFF)	255 (0xFF)	Target station side module station number	255 (0xFF)
ActStopBits	0 (STOPBIT_ONE)	0 (0x00)		
ActSumCheck	0 (NO_SUM_CHECK)	0 (0x00)		
ActThroughNetworkType	0 (0x00)	Fixed to 0 (0x00)		
ActTimeOut	10000	Any value specified by user in ms units		
ActUnitNumber	0 (0x00)	0 (0x00)		Target station side module station number
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_FXVUSB		

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

*1 Since CC-Link IE Field Network is not supported, R motion CPU, Q12DCCPU-V (Basic mode), QSCPU, and Q motion CPU relayed by CC-Link IE Field Network cannot be accessed.

*2 Since CC-Link IE Controller Network is not supported, LCPU relayed by CC-Link IE Controller Network cannot be accessed.

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

*4 Supported by FX3G(C)CPU and FX3U(C)CPU within the own network only.

- *1 Note the following considerations when accessing via Ethernet module (Q series-compatible E71).
 - For ActNetworkNumber and ActStationNumber, specify the value set in the parameter setting of the target station side Q series-compatible E71.
 - Set "Station No. ⇔ IP information" in the parameter setting of Q series-compatible E71. When setting the parameter, specify the setting other than the automatic response system (any of the IP address calculation system, table conversion system, and combined system) for "Station No. ⇔ IP information system."
- *2 For the I/O address, specify the value of the actual start I/O number divided by 16.
- *3 Specify the following value for the channel number to be multi-drop linked.
 - 0: Default channel of module
 - 1: Channel 1
 - 2: Channel 2

Property list

The following table shows the property settings of communication routes.

The setting is not necessary for those properties without description.

Property	Default value	Property patterns			
		○(1)	○(2)	○(3)	○(4)
ActBaudRate	19200 (BAUDRATE_19200)	Fixed to 0 (0x00)			
ActConnectUnitNumber	0 (0x00)	Fixed to 0 (0x00)			
ActControl	8 (TRC_DTR_OR_RT S)	Fixed to 0 (0x00)			
ActCpuTimeOut	0 (0x00)	Fixed to 0 (0x00)			
ActCpuType	34 (CPU_Q02CPU)	CPU type corresponding to the target station			
ActDataBits	8 (DATABIT_8)	Fixed to 0 (0x00)			
ActDestinationIONumber	0 (0x00)	Fixed to 0 (0x00)	Target station side ■For single CPU • Fixed to 1023 (0x3FF) ■For multiple CPUs • Connected CPU: 1023 (0x3FF) • No. 1: 992 (0x3E0) • No. 2: 993 (0x3E1) • No. 3: 994 (0x3E2) • No. 4: 995 (0x3E3) ■For redundant CPU • Control system: 976 (0x3D0) • No specification: 1023 (0x3FF)	Target station side ■For single CPU • Fixed to 1023 (0x3FF) ■For multiple CPUs • Connected CPU: 1023 (0x3FF) • No. 1: 992 (0x3E0) • No. 2: 993 (0x3E1) • No. 3: 994 (0x3E2) • No. 4: 995 (0x3E3) ■For redundant CPU • Control system: 976 (0x3D0) • No specification: 1023 (0x3FF)	Fixed to 0 (0x00)
ActDestinationPortNumber	0 (0x00)	Fixed to 0 (0x00)			
ActDidPropertyBit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Fixed to 1 (0x01)
ActDsidPropertyBit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Fixed to 1 (0x01)
ActHostAddress	1.1.1.1	Fixed to NULL			
ActIntelligentPreferenceBit	0 (0x00)	Fixed to 0 (0x00)	Target station QCPU (Q mode), LCPU: 1 (0x01) Other than the above: 0 (0x00)	Target station QCPU (Q mode), QCCPU, LCPU: 1 (0x01) Other than the above: 0 (0x00)	Fixed to 0 (0x00)
ActIONumber ^{*1}	1023 (0x3FF)	Fixed to 1023 (0x3FF)	Connected station side module I/O address	Connected station side module I/O address	Target station side ■For single CPU • Fixed to 1023 (0x3FF) ■For multiple CPUs • Connected CPU: 1023 (0x3FF) • No. 1: 992 (0x3E0) • No. 2: 993 (0x3E1) • No. 3: 994 (0x3E2) • No. 4: 995 (0x3E3) ■For redundant CPU • Control system: 976 (0x3D0) • No specification: 1023 (0x3FF)
ActMultiDropChannelNumber ^{*2}	0 (0x00)	Fixed to 0 (0x00)	Multi-drop channel number	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActNetworkNumber	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station side module network number
ActPacketType	0x01 (PACKET_PLC1)	Fixed to 0x01 (PACKET_PLC1)			

Property	Default value	Property patterns			
		○(1)	○(2)	○(3)	○(4)
ActParity	1 (ODD_PARITY)	Fixed to 0 (0x00)			
ActPortNumber	1 (PORT_1)	Fixed to 1 (PORT_1)			
ActProtocolType	0x04 (PROTOCOL_SERIAL)	PROTOCOL_USB (0x0D)			
ActSourceNetworkNumber	0 (0x00)	Fixed to 0 (0x00)			
ActSourceStationNumber	0 (0x00)	Fixed to 0 (0x00)			
ActStationNumber	255 (0xFF)	Fixed to 255 (0xFF)	Fixed to 255 (0xFF)	Fixed to 255 (0xFF)	Target station side module station number
ActStopBits	0 (STOPBIT_ONE)	Fixed to 0 (STOPBIT_ONE)			
ActSumCheck	0 (NO_SUM_CHECK)	Fixed to 0 (NO_SUM_CHECK)			
ActThroughNetworkType	0 (0x00)	<ul style="list-style-type: none"> • MELSECNET/10 is not included.: 0 (0x00) • MELSECNET/10 is included.: 1 (0x01) 			
ActTimeOut	10000	Any value specified by user in ms units			
ActUnitNumber	0 (0x00)	Fixed to 0 (0x00)	Target station side module station number	Target station side module station number	Fixed to 0 (0x00)
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_LNUSB (0x51)			

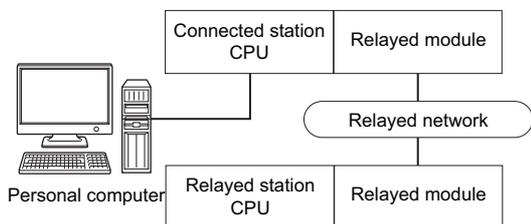
*1 For the I/O address, specify the value of the actual start I/O number divided by 16.

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

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

CPU USB communication when the connected station is a QSCPU

Configuration



Accessibility and property patterns

○(n): Accessible ('n' is the property pattern number), ×: Inaccessible

Connected station CPU	Relayed network	Relayed station CPU				
		R CPU	RCCPU	R motion CPU	LH CPU	FX5 CPU
○(1) ^{*1}	CC IE TSN	×	×	×	×	×
	CC IE Control CC IE Field	×	×	×	×	×
	MELSECNET/H	×	×	×	×	×
	Ethernet	×	×	×	×	×
	Serial communication	×	×	×	×	×
	CC-Link	×	×	×	×	×

Connected station CPU	Relayed network	Relayed station CPU					
		Q CPU (Q mode)	QCCPU	L CPU	QSCPU	Q motion CPU	FX CPU
○(1) ^{*1}	CC IE TSN	×	×	×	×	×	×
	CC IE Control CC IE Field	×	×	×	×	×	×
	MELSECNET/H	×	×	×	×	×	×
	Ethernet	×	×	×	×	×	×
	Serial communication	×	×	×	×	×	×
	CC-Link	×	×	×	×	×	×

*1 Relayed station CPUs cannot be accessed via QSCPU.

Property list

The following table shows the property settings of communication routes.

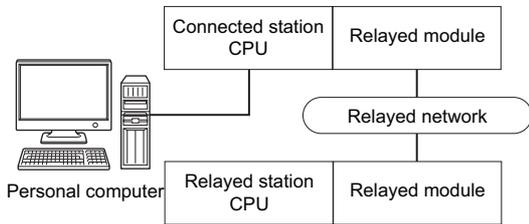
The setting is not necessary for those properties without description.

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

Property	Default value	Property patterns
		○(1)
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_QNMOTIONUSB (0x1D)

CPU USB communication when the connected station is an FXCPU

Configuration



Accessibility and property patterns

○(n): Accessible ('n' is the property pattern number), ×: Inaccessible

Connected station CPU FXCPU	Relayed network	Relayed station CPU				
		RCPU	RCCPU	R motion CPU	LHCPU	FX5CPU
○(1)	CC IE TSN	×	×	×	×	×
	CC IE Control CC IE Field	×	×	×	×	×
	MELSECNET/H	×	×	×	×	×
	Ethernet	×	×	×	×	×
	Serial communication	×	×	×	×	×
	CC-Link	×	×	×	×	×

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

*1 Only FX3G(C)CPU and FX3U(C)CPU can be accessed.

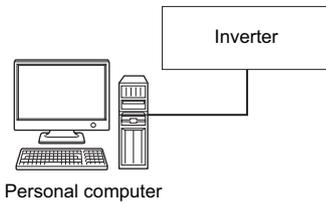
Property list

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

Property	Default value	Property patterns	
		○(1)	○(2)
ActCpuType	34 (CPU_Q02CPU)	CPU type corresponding to the target station	
ActDestinationIONumber	0 (0x00)	Fixed to 0 (0x00)	1023 (0x3FF)
ActIONumber	1023 (0x3FF)	Fixed to 0 (0x00)	Module number of the connected station
ActProtocolType	0x04 (PROTOCOL_SERIAL)	PROTOCOL_USB (0x0D)	
ActTimeOut	10000	Any value specified by user in ms units	
ActUnitNumber	0 (0x00)	Fixed to 0 (0x00)	Target station side module station number
ActUnitType	0x13 (UNIT_QNCP)	UNIT_FXCPU (0x0F)	

CPU USB communication when the connected station is an inverter

Configuration



Property list

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

Property	Default value	Property patterns
ActMxUnitSeries	0 (0x00)	1 (0x01)
ActProtocolType	0x04 (PROTOCOL_SERIAL)	COMM_USB (0x01)
ActStationNumber	255 (0xFF)	Inverter station number (0 to 31)
ActCpuType	34 (CPU_Q02CPU)	Fixed to INV_A800 (0x1E60)
ActTimeOut	10000	Any value specified by user in ms units
ActCpuTimeOut	0 (0x00)	Any value specified by user in 10ms units

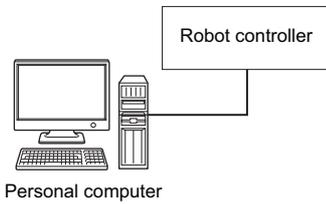
Point

When performing inverter communication, the program setting type control cannot be used.
Use the utility setting type control.

CPU USB communication when the connected station is a robot controller

Configuration

10



Property list

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

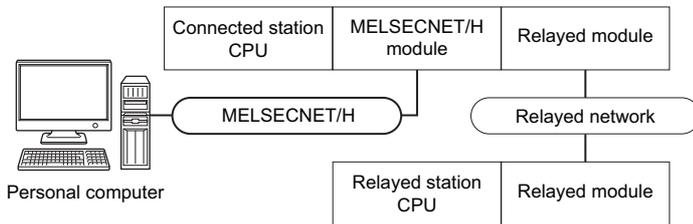
Property	Default value	Property patterns
ActProtocolType	0x04 (PROTOCOL_SERIAL)	RC_PROTOCOL_USB (0x04)
ActCpuType	34 (CPU_Q02CPU)	Robot controller type (0x013001)
ActTimeOut	10000	Receive timeout time (msec)
ActCpuTimeOut	0 (0x00)	Send timeout time (msec)
ActMultiDropChannelNumber	0 (0x00)	Retry count
ActMxUnitSeries	0 (0x00)	2 (0x02)

Point

When robot controller communication, the program setting type control cannot be used.
Use the utility setting type control.

10.5 MELSECNET/H Communication

Configuration



Accessibility and property patterns

○(n): Accessible ('n' is the property pattern number), ×: Inaccessible

Own board	Connected station CPU			Relayed network	Relayed station CPU				
	QCPU (Q mode)	QSCPU	Q motion CPU		RCPU	RCCPU	R motion CPU	LHCPU	FX5CPU
○(1)	○(2)	○(2) ^{*1*5}	○(2) ^{*1*5}	CC IE TSN	×	×	×	×	×
				CC IE Control CC IE Field	×	×	×	×	×
				MELSECNET/H	×	×	×	×	×
				Ethernet	×	×	×	×	×
				Serial communication	×	×	×	×	×
				CC-Link	×	×	×	×	×

Own board	Connected station CPU			Relayed network	Relayed station CPU					
	QCPU (Q mode)	QSCPU	Q motion CPU		QCPU (Q mode)	QCCP U	LCPU	QSCP U	Q motion CPU ^{*5}	FXCPU
○(1)	○(2)	○(2) ^{*1*5}	○(2) ^{*1*5}	CC IE TSN	×	×	×	×	×	×
				CC IE Control CC IE Field	○(2)	○(2) ^{*2}	○(2) ^{*3}	○(2) ^{*1*2}	×	×
				MELSECNET/H	○(2)	○(2)	×	○(2) ^{*1}	×	×
				Ethernet	○(2)	×	×	○(2) ^{*1}	×	×
				Serial communication	○(3) ^{*4}	×	○(3) ^{*4}	×	○(3)	×
				CC-Link	○(4)	○(4)	○(4)	×	○(4)	×

Own board	Connected station CPU			Relayed network	Relayed station CPU				
	QCCPU				RCPU	RCCPU	R motion CPU	LHCPU	FX5CPU
○(1)	○(2) ^{*6}			CC IE TSN	×	×	×	×	×
				CC IE Control CC IE Field	×	×	×	×	×
				MELSECNET/H	×	×	×	×	×
				Ethernet	×	×	×	×	×
				Serial communication	×	×	×	×	×
				CC-Link	×	×	×	×	×

Own board	Connected station CPU	Relayed network	Relayed station CPU					
	QCCPU		QCPU (Q mode)	QCCPU	LCPU	QSCPU	Q motion CPU*5	FXCPU
○(1)	○(2)*6	CC IE TSN	×	×	×	×	×	×
		CC IE Control CC IE Field	×	×	×	×	×	×
		MELSECNET/H	×	×	×	×	×	×
		Ethernet	×	×	×	×	×	×
		Serial communication	×	×	×	×	×	×
		CC-Link	○(4)	○(4)	○(4)	×	○(4)	×

- *1 Relayed station CPUs cannot be accessed via QSCPU or Q motion CPU.
- *2 Since CC-Link IE Field Network is not supported, Q12DCCPU-V (Basic mode) and QSCPU relayed by CC-Link IE Field Network cannot be accessed.
- *3 Since CC-Link IE Controller Network is not supported, LCPU relayed by CC-Link IE Controller Network cannot be accessed.
- *4 For redundant CPU, serial communication modules on the main base unit cannot be accessed.
- *5 Only CPU No.2 or later in a multiple CPU configuration can be accessed.
- *6 Q24DHCCPU-V, Q24DHCCPU-LS, Q24DHCCPU-VG and Q26DHCCPU-LS cannot be accessed because the communication route is not supported.

Property list

The following table shows the property settings of communication routes.

The setting is not necessary for those properties without description.

Property	Default value	Property patterns			
		○(1)	○(2)	○(3)	○(4)
ActCpuType	34 (CPU_Q02CPU)	CPU type corresponding to the target station			
ActDestinationIONumber	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station side ■For single CPU • Fixed to 1023 (0x3FF) ■For multiple CPUs • Connected CPU: 1023 (0x3FF) • No. 1: 992 (0x3E0) • No. 2: 993 (0x3E1) • No. 3: 994 (0x3E2) • No. 4: 995 (0x3E3) ■For redundant CPU • Control system: 976 (0x3D0) • No specification: 1023 (0x3FF)	Target station side ■For single CPU • Fixed to 1023 (0x3FF) ■For multiple CPUs • Connected CPU: 1023 (0x3FF) • No. 1: 992 (0x3E0) • No. 2: 993 (0x3E1) • No. 3: 994 (0x3E2) • No. 4: 995 (0x3E3) ■For redundant CPU • Control system: 976 (0x3D0) • No specification: 1023 (0x3FF)
ActDidPropertyBit	1 (0x01)	Fixed to 0 (0x00)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActDsidPropertyBit	1 (0x01)	Fixed to 0 (0x00)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActIONumber*1	1023 (0x3FF)	Fixed to 0 (0x00)	Target station side ■For single CPU • Fixed to 1023 (0x3FF) ■For multiple CPUs • Connected CPU: 1023 (0x3FF) • No. 1: 992 (0x3E0) • No. 2: 993 (0x3E1) • No. 3: 994 (0x3E2) • No. 4: 995 (0x3E3) ■For redundant CPU • Control system: 976 (0x3D0) • No specification: 1023 (0x3FF)	Connected station side relayed module I/O address	Connected station side relayed module I/O address
ActMultiDropChannelNumber*2	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Multi-drop channel number	Fixed to 0 (0x00)
ActNetworkNumber	0 (0x00)	Fixed to 0 (0x00)	Target station side module network number	Connected station side module network number	Connected station side module network number
ActPortNumber	1 (PORT_1)	Port number of personal computer side MELSECNET/H board, PORT 1 to PORT 4 (first to fourth boards)			
ActProtocolType	0x04 (PROTOCOL_SERIAL)	PROTOCOL_MNETH (0x0F)			
ActStationNumber	255 (0xFF)	Fixed to 255 (0xFF)	Target station side module station number	Connected station side module station number	Connected station side module station number
ActThroughNetworkType	0 (0x00)	• MELSECNET/10 is not included.: 0 (0x00) • MELSECNET/10 is included.: 1 (0x01)			
ActUnitNumber	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station side module station number	Target station side module station number
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_MNETHBOARD (0x1E)			

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

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

0: Default channel of module

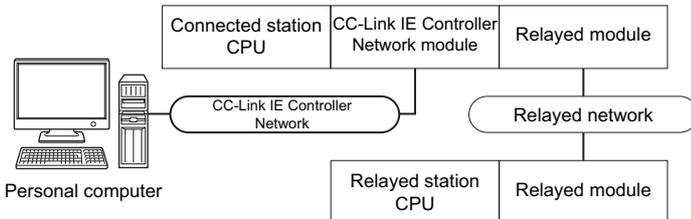
1: Channel 1

2: Channel 2

10.6 CC-Link IE Controller Network Communication

CC-Link IE Controller Network communication when the connected station is an RCPU

Configuration



Accessibility and property patterns

○(n): Accessible ('n' is the property pattern number), ×: Inaccessible

Own board	Connected station CPU	Relayed network	Relayed station CPU					
	RCPU		RCPU	RCCPU	R motion CPU	LHCPU	FX5CPU	
×	○(2)	CC IE TSN	○(2)	×	×	×	×	×
		CC IE Control CC IE Field	○(2)	×	×	×	×	×
		MELSECNET/H	×	×	×	×	×	×
		Ethernet	○(2)	×	×	×	×	×
		Serial communication	○(3) ^{*1}	×	×	×	×	×
		CC-Link	○(4)	×	×	×	×	×

Own board	Connected station CPU	Relayed network	Relayed station CPU							
	RCPU		QCPU (Q mode)	QCCPU	LCPU	QSCPU	Q motion CPU ^{*2}	FXCPU		
×	○(2)	CC IE TSN	×	×	×	×	×	×	×	
		CC IE Control CC IE Field	×	×	×	×	×	×	×	
		MELSECNET/H	×	×	×	×	×	×	×	
		Ethernet	×	×	×	×	×	×	×	
		Serial communication	○(3) ^{*1}	×	×	○(3) ^{*1}	×	×	×	×
		CC-Link	○(4)	×	×	○(4)	×	×	×	×

*1 For redundant CPU, serial communication modules on the main base unit cannot be accessed because the multi-drop connection cannot be performed.

*2 Only CPU No.2 or later in a multiple CPU configuration can be accessed.

Property list

The following table shows the property settings of communication routes.

The setting is not necessary for those properties without description.

Property	Default value	Property patterns			
		○(1)	○(2)	○(3)	○(4)
ActCpuType	34 (CPU_Q02CPU)	CPU type corresponding to the target station			
ActDestinationIONumber	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station side ■For single CPU • Fixed to 1023 (0x3FF) ■For multiple CPUs • Connected CPU: 1023 (0x3FF) • No. 1: 992 (0x3E0) • No. 2: 993 (0x3E1) • No. 3: 994 (0x3E2) • No. 4: 995 (0x3E3)	Target station side ■For single CPU • Fixed to 1023 (0x3FF) ■For multiple CPUs • Connected CPU: 1023 (0x3FF) • No. 1: 992 (0x3E0) • No. 2: 993 (0x3E1) • No. 3: 994 (0x3E2) • No. 4: 995 (0x3E3) ■For redundant CPU • No specification: 1023 (0x3FF)
ActDidPropertyBit	1 (0x01)	Fixed to 0 (0x00)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActDsidPropertyBit	1 (0x01)	Fixed to 0 (0x00)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActIONumber*1	1023 (0x3FF)	Fixed to 0 (0x00)	Target station side ■For single CPU • Fixed to 1023 (0x3FF) ■For multiple CPUs • Connected CPU: 1023 (0x3FF) • No. 1: 992 (0x3E0) • No. 2: 993 (0x3E1) • No. 3: 994 (0x3E2) • No. 4: 995 (0x3E3) ■For redundant CPU • Control system: 976 (0x3D0) • No specification: 1023 (0x3FF)	Connected station side relayed module I/O address	Connected station side relayed module I/O address
ActMultiDropChannelNumber*2	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Multi-drop channel number	Fixed to 0 (0x00)
ActNetworkNumber	0 (0x00)	Fixed to 0 (0x00)	Target station side module network number	Connected station side module network number	Connected station side module network number
ActPortNumber	1 (PORT_1)	Port number of personal computer side CC-Link IE Controller Network board, PORT 1 to PORT 4 (first to fourth boards)			
ActProtocolType	0x04 (PROTOCOL_SERIAL)	PROTOCOL_MNETG (0x14)			
ActStationNumber	255 (0xFF)	Fixed to 255 (0xFF)	Target station side module station number	Connected station side module station number	Connected station side module station number
ActThroughNetworkType	0 (0x00)	• MELSECNET/10 is not included.: 0 (0x00) • MELSECNET/10 is included.: 1 (0x01)			
ActUnitNumber	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station side module station number	Target station side module station number
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_MNETGBOARD (0x2B)			

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

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

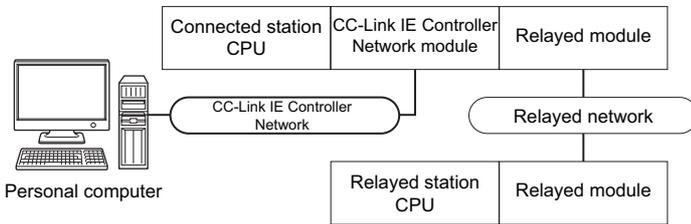
0: Default channel of module

1: Channel 1

2: Channel 2

CC-Link IE Controller Network communication when the connected station is a module supporting Q series

Configuration



Accessibility and property patterns

○(n): Accessible ('n' is the property pattern number), ×: Inaccessible

Own board	Connected station CPU			Relayed network	Relayed station CPU					
	QCPU (Q mode)	QSCPU	Q motion CPU		RCPU	RCCPU	R motion CPU	LHCPU	FX5CPU	
○(1)	○(2)	○(2) ^{*1*5}	○(2) ^{*1*5}	CC IE TSN	×	×	×	×	×	
				CC IE Control CC IE Field	×	×	×	×	×	
				MELSECNET/H	×	×	×	×	×	
				Ethernet	×	×	×	×	×	
				Serial communication	×	×	×	×	×	
				CC-Link	×	×	×	×	×	
Own board	Connected station CPU			Relayed network	Relayed station CPU					
	QCPU (Q mode)	QSCPU	Q motion CPU		QCPU (Q mode)	QCCP U	LCPU	QSCP U	Q motion CPU ^{*5}	FXCP U
○(1)	○(2)	○(2) ^{*1*5}	○(2) ^{*1*5}	CC IE TSN	×	×	×	×	×	×
				CC IE Control CC IE Field	○(2)	○(2) ^{*2}	○(2) ^{*3}	○(2) ^{*1*2}	×	×
				MELSECNET/H	○(2)	○(2)	×	○(2) ^{*1}	×	×
				Ethernet	○(2)	×	×	○(2) ^{*1}	×	×
				Serial communication	○(3) ^{*4}	×	○(3)	×	○(3)	×
				CC-Link	○(4)	○(4)	○(4)	×	○(4)	×
Own board	Connected station CPU			Relayed network	Relayed station CPU					
	QCCPU				RCPU	RCCPU	R motion CPU	LHCPU	FX5CPU	
○(1)	○(2) ^{*6}			CC IE TSN	×	×	×	×	×	
				CC IE Control CC IE Field	×	×	×	×	×	
				MELSECNET/H	×	×	×	×	×	
				Ethernet	×	×	×	×	×	
				Serial communication	×	×	×	×	×	
				CC-Link	×	×	×	×	×	

Own board	Connected station CPU	Relayed network	Relayed station CPU					
	QCCPU		QCPU (Q mode)	QCCPU	LCPU	QSCPU	Q motion CPU*5	FXCPU
○(1)	○(2)*6	CC IE TSN	×	×	×	×	×	×
		CC IE Control CC IE Field	×	×	×	×	×	×
		MELSECNET/H	×	×	×	×	×	×
		Ethernet	×	×	×	×	×	×
		Serial communication	×	×	×	×	×	×
		CC-Link	○(4)	○(4)	○(4)	×	○(4)	×

*1 Relayed station CPUs cannot be accessed via QSCPU or Q motion CPU.

*2 Since CC-Link IE Field Network is not supported, Q12DCCPU-V (Basic mode) and QSCPU relayed by CC-Link IE Field Network cannot be accessed.

*3 Since CC-Link IE Controller Network is not supported, LCPU relayed by CC-Link IE Controller Network cannot be accessed.

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

*5 Only CPU No.2 or later in a multiple CPU configuration can be accessed.

*6 Q24DHCCPU-V, Q24DHCCPU-LS, Q24DHCCPU-VG and Q26DHCCPU-LS cannot be accessed because the communication route is not supported.

Property list

The following table shows the property settings of communication routes.

The setting is not necessary for those properties without description.

Property	Default value	Property patterns			
		○(1)	○(2)	○(3)	○(4)
ActCpuType	34 (CPU_Q02CPU)	CPU type corresponding to the target station			
ActDestinationIONumber	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station side ■For single CPU • Fixed to 1023 (0x3FF) ■For multiple CPUs • Connected CPU: 1023 (0x3FF) • No. 1: 992 (0x3E0) • No. 2: 993 (0x3E1) • No. 3: 994 (0x3E2) • No. 4: 995 (0x3E3) ■For redundant CPU • Control system: 976 (0x3D0) • No specification: 1023 (0x3FF)	Target station side ■For single CPU • Fixed to 1023 (0x3FF) ■For multiple CPUs • Connected CPU: 1023 (0x3FF) • No. 1: 992 (0x3E0) • No. 2: 993 (0x3E1) • No. 3: 994 (0x3E2) • No. 4: 995 (0x3E3) ■For redundant CPU • Control system: 976 (0x3D0) • No specification: 1023 (0x3FF)
ActDidPropertyBit	1 (0x01)	Fixed to 0 (0x00)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActDsidPropertyBit	1 (0x01)	Fixed to 0 (0x00)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActIONumber* ¹	1023 (0x3FF)	Fixed to 0 (0x00)	Target station side ■For single CPU • Fixed to 1023 (0x3FF) ■For multiple CPUs • Connected CPU: 1023 (0x3FF) • No. 1: 992 (0x3E0) • No. 2: 993 (0x3E1) • No. 3: 994 (0x3E2) • No. 4: 995 (0x3E3) ■For redundant CPU • Control system: 976 (0x3D0) • No specification: 1023 (0x3FF)	Connected station side relayed module I/O address	Connected station side relayed module I/O address
ActMultiDropChannelNumber* ²	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Multi-drop channel number	Fixed to 0 (0x00)
ActNetworkNumber	0 (0x00)	Fixed to 0 (0x00)	Target station side module network number	Connected station side module network number	Connected station side module network number
ActPortNumber	1 (PORT_1)	Port number of personal computer side CC-Link IE Controller Network board, PORT 1 to PORT 4 (first to fourth boards)			
ActProtocolType	0x04 (PROTOCOL_SERIAL)	PROTOCOL_MNETG (0x14)			
ActStationNumber	255 (0xFF)	Fixed to 255 (0xFF)	Target station side module station number	Connected station side module station number	Connected station side module station number
ActThroughNetworkType	0 (0x00)	• MELSECNET/10 is not included.: 0 (0x00) • MELSECNET/10 is included.: 1 (0x01)			
ActUnitNumber	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station side module station number	Target station side module station number
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_MNETGBOARD (0x2B)			

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

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

0: Default channel of module

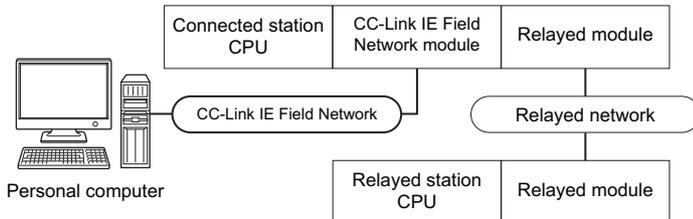
1: Channel 1

2: Channel 2

10.7 CC-Link IE Field Network Communication

CC-Link IE Field Network communication when the connected station is an RCPU

Configuration



Accessibility and property patterns

○(n): Accessible ('n' is the property pattern number), ×: Inaccessible

Own board	Connected station CPU	Relayed network	Relayed station CPU				
	RCPU		RCPU	RCCPU	R motion CPU	LHCPU	FX5CPU
×	○(1)	CC IE TSN	○(1)	×	×	×	×
		CC IE Field	○(1)	×	×	×	×
		MELSECNET/H	×	×	×	×	×
		Ethernet	○(1)	×	×	×	×
		Serial communication	○(2) ^{*1}	×	×	×	×
		CC-Link	○(3)	×	×	×	×

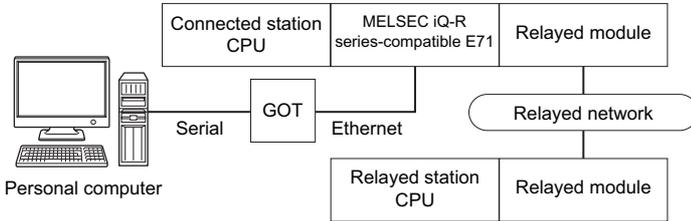
Own board	Connected station CPU	Relayed network	Relayed station CPU					
	RCPU		QCPU (Q mode)	QCCPU	LCPU	QSCPU	Q motion CPU	FXCPU
×	○(1)	CC IE TSN	×	×	×	×	×	×
		CC IE Field	×	×	×	×	×	×
		MELSECNET/H	×	×	×	×	×	×
		Ethernet	×	×	×	×	×	×
		Serial communication	○(2) ^{*1}	×	○(2) ^{*1}	×	×	×
		CC-Link	○(3)	×	○(3)	×	×	×

*1 For redundant CPU, serial communication modules on the main base unit cannot be accessed because the multi-drop connection cannot be performed.

- *1 For the I/O address, specify the value of the actual start I/O number divided by 16.
- *2 Specify the following value for the channel number to be multi-drop linked.
 - 0: Default channel of module
 - 1: Channel 1
 - 2: Channel 2

Personal computer side port: Serial, GOT2000 side port: Serial, CPU side port: MELSEC iQ-R series-compatible E71

Configuration



Accessibility and property patterns

○(n): Accessible ('n' is the property pattern number), ×: Inaccessible

Connected station CPU	Relayed network	Relayed station CPU				
		RCPU	RCCPU	R motion CPU	LHCPU	FX5CPU
○(1)*1	CC IE TSN	×	×	×	×	×
	CC IE Control CC IE Field	×	×	×	×	×
	MELSECNET/H	×	×	×	×	×
	Ethernet	○(2)	×	×	×	×
	Serial communication	○(3)	×	×	×	×
	CC-Link	○(4)	○(4)	×	×	×

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

*1 Only CPU No.2 or later in a multiple CPU configuration can be accessed.

Property list

The following table shows the property settings of communication routes.

The setting is not necessary for those properties without description.

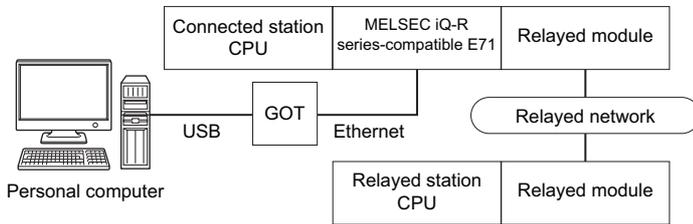
Property	Default value	Property patterns			
		○(1)	○(2)	○(3)	○(4)
ActBaudRate	19200 (BAUDRATE_19200)	BAUDRATE_9600, BAUDRATE_19200, BAUDRATE_38400, BAUDRATE_57600, BAUDRATE_115200			
ActConnectUnitNumber ^{*1}	0 (0x00)	Fixed to 0 (0x00)	Connected station side module station number	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActCpuType	34 (CPU_Q02CPU)	CPU type corresponding to the target station			
ActDestinationIONumber ^{*2}	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station side ■For single CPU • Fixed to 1023 (0x3FF) ■For multiple CPUs • Connected CPU: 1023 (0x3FF) • No. 1: 992 (0x3E0) • No. 2: 993 (0x3E1) • No. 3: 994 (0x3E2) • No. 4: 995 (0x3E3)	Target station side ■For single CPU • Fixed to 1023 (0x3FF) ■For multiple CPUs • Connected CPU: 1023 (0x3FF) • No. 1: 992 (0x3E0) • No. 2: 993 (0x3E1) • No. 3: 994 (0x3E2) • No. 4: 995 (0x3E3)
ActDestinationPortNumber	0 (0x00)	5001 Host station port number for OPS connection ^{*3}			
ActDidPropertyBit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActDsidPropertyBit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActHostAddress	1.1.1.1	Host name or IP address of the connected station side module			
ActIONumber ^{*2}	1023 (0x3FF)	■For single CPU • Fixed to 1023 (0x3FF) ■For multiple CPUs • Connected CPU: 1023 (0x3FF) • No. 1: 992 (0x3E0) • No. 2: 993 (0x3E1) • No. 3: 994 (0x3E2) • No. 4: 995 (0x3E3)	Target station side ■For single CPU • Fixed to 1023 (0x3FF) ■For multiple CPUs • Connected CPU: 1023 (0x3FF) • No. 1: 992 (0x3E0) • No. 2: 993 (0x3E1) • No. 3: 994 (0x3E2) • No. 4: 995 (0x3E3)	Connected station side module I/O address	Connected station side module I/O address
ActMultiDropChannelNumber ^{*4}	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Multi-drop channel number	Fixed to 0 (0x00)
ActNetworkNumber ^{*5}	0 (0x00)	Target station side module network number	Target station side module network number	Connected station side MELSEC iQ-R series- compatible E71 network number	Connected station side MELSEC iQ-R series- compatible E71 network number
ActPortNumber	1 (PORT_1)	Personal computer side COM port number			
ActProtocolType	0x04 (PROTOCOL_SERIAL)	PROTOCOL_SERIAL			
ActPacketType	0x01 (PACKET_PLC1)	PACKET_PLC1			
ActCpuTimeOut	0 (0x00)	Fixed to 0 (0x00)			
ActControl	8 (TRC_DTR_OR_RTS)	Depending on the used cable.			
ActDataBits	8 (DATABIT_8)	Fixed to 8 (0x08)			
ActParity	1 (ODD_PARITY)	Fixed to 1 (0x01)			
ActStopBits	0 (STOPBIT_ONE)	Fixed to 0 (0x00)			
ActSumCheck	0 (NO_SUM_CHECK)	Fixed to 0 (0x00)			
ActIntelligentPreferenceBit	0 (0x00)	Fixed to 0 (0x00)			
ActSourceNetworkNumber ^{*6}	0 (0x00)	GOT side network number			
ActSourceStationNumber ^{*7}	0 (0x00)	GOT side station number			

Property	Default value	Property patterns			
		○(1)	○(2)	○(3)	○(4)
ActStationNumber ^{*5}	255 (0xFF)	Target station side module station number	Target station side module station number	Connected station side MELSEC iQ-R series-compatible E71 station number	Connected station side MELSEC iQ-R series-compatible E71 station number
ActThroughNetworkType	0 (0x00)	<ul style="list-style-type: none"> • MELSECNET/10 is not included.: 0 (0x00) • MELSECNET/10 is included.: 1 (0x01) 			
ActTimeOut	10000	Any value specified by user in ms units			
ActUnitNumber	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station side module station number	Target station side module station number
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_GOT_RJ71EN71			

- *1 When the property pattern is ○(2), specify the connected station side MELSEC iQ-R series-compatible E71 station number set to the connected station side MELSEC iQ-R series-compatible E71 Ethernet parameter.
- *2 For the I/O address, specify the value of the actual start I/O number divided by 16.
- *3 When using OPS connection function at redundant CPU connection, set the port No. specified to "Host Station Port No." in Network parameter.
- *4 Specify the following value for the channel number to be multi-drop linked.
0: Default channel of module
1: Channel 1
2: Channel 2
- *5 When the property pattern is ○(1) or ○(2), specify the value set in the target station side parameter for ActNetworkNumber and ActStationNumber.
- *6 Specify the same network No. as MELSECNET/10 network No. set to MELSEC iQ-R series-compatible E71 in the Ethernet parameter settings of target station side MELSEC iQ-R series-compatible E71.
- *7 Specify the station number on the GOT side to avoid setting the same station number as set to the MELSEC iQ-R series-compatible E71 on the same Ethernet loop.

Personal computer side port: USB, GOT2000 side port: USB, CPU side port: MELSEC iQ-R series-compatible E71

Configuration



Accessibility and property patterns

○(n): Accessible ('n' is the property pattern number), ×: Inaccessible

Connected station CPU		Relayed network	Relayed station CPU				
RCPU	RCCPU		RCPU	RCCPU	R motion CPU	LHCPU	FX5CPU
○(1)*1	○(1)*1*2	CC IE TSN	×	×	×	×	×
		CC IE Control CC IE Field	×	×	×	×	×
		MELSECNET/H	×	×	×	×	×
		Ethernet	○(2)	×	×	×	×
		Serial communication	○(3)	×	×	×	×
		CC-Link	○(4)	○(4)	×	×	×

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

*1 Only CPU No.2 or later in a multiple CPU configuration can be accessed.

*2 A communication with a relayed station via network cannot be established.

Property list

The following table shows the property settings of communication routes.

The setting is not necessary for those properties without description.

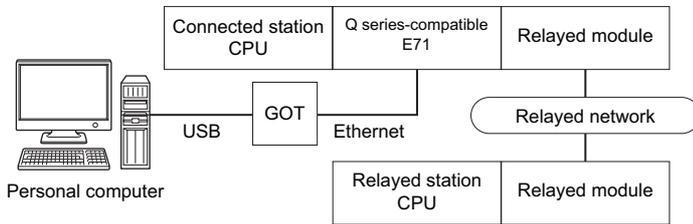
Property	Default value	Property patterns			
		○(1)	○(2)	○(3)	○(4)
ActConnectUnitNumber*1	0 (0x00)	Fixed to 0 (0x00)	Connected station side module station number	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActCpuType	34 (CPU_Q02CPU)	CPU type corresponding to the target station			
ActDestinationIONumber*2	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station side ■For single CPU • Fixed to 1023 (0x3FF) ■For multiple CPUs • Connected CPU: 1023 (0x3FF) • No. 1: 992 (0x3E0) • No. 2: 993 (0x3E1) • No. 3: 994 (0x3E2) • No. 4: 995 (0x3E3)	Target station side ■For single CPU • Fixed to 1023 (0x3FF) ■For multiple CPUs • Connected CPU: 1023 (0x3FF) • No. 1: 992 (0x3E0) • No. 2: 993 (0x3E1) • No. 3: 994 (0x3E2) • No. 4: 995 (0x3E3)
ActDestinationPortNumber	0 (0x00)	5001 Host station port number for OPS connection*3			
ActDidPropertyBit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActDsidPropertyBit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActHostAddress	1.1.1.1	Host name or IP address of the connected station side module			
ActIONumber*2	1023 (0x3FF)	■For single CPU • Fixed to 1023 (0x3FF) ■For multiple CPUs • Connected CPU: 1023 (0x3FF) • No. 1: 992 (0x3E0) • No. 2: 993 (0x3E1) • No. 3: 994 (0x3E2) • No. 4: 995 (0x3E3)	Target station side ■For single CPU • Fixed to 1023 (0x3FF) ■For multiple CPUs • Connected CPU: 1023 (0x3FF) • No. 1: 992 (0x3E0) • No. 2: 993 (0x3E1) • No. 3: 994 (0x3E2) • No. 4: 995 (0x3E3)	Connected station side module I/O address	Connected station side module I/O address
ActMultiDropChannelNumber*4	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Multi-drop channel number	Fixed to 0 (0x00)
ActNetworkNumber*5	0 (0x00)	Target station side module network number	Target station side module network number	Connected station side MELSEC iQ-R series-compatible E71 network number	Connected station side MELSEC iQ-R series-compatible E71 network number
ActProtocolType	0x04 (PROTOCOL_SERIAL)	PROTOCOL_USBGOT			
ActPacketType	0x01 (PACKET_PLC1)	PACKET_PLC1			
ActPortNumber	1 (PORT_1)	Fixed to 0 (0x00)			
ActBaudRate	19200 (BAUDRATE_19200)	Fixed to 0 (0x00)			
ActCpuTimeOut	0 (0x00)	Fixed to 0 (0x00)			
ActControl	8 (TRC_DTR_OR_RTS)	Fixed to 0 (0x00)			
ActDataBits	8 (DATABIT_8)	Fixed to 0 (0x00)			
ActParity	1 (ODD_PARITY)	Fixed to 0 (0x00)			
ActStopBits	0 (STOPBIT_ONE)	Fixed to 0 (0x00)			
ActSumCheck	0 (NO_SUM_CHECK)	Fixed to 0 (0x00)			
ActIntelligentPreferenceBit	0 (0x00)	Fixed to 0 (0x00)			
ActSourceNetworkNumber*6	0 (0x00)	GOT side network number			
ActSourceStationNumber*7	0 (0x00)	GOT side station number			

Property	Default value	Property patterns			
		○(1)	○(2)	○(3)	○(4)
ActStationNumber*5	255 (0xFF)	Target station side module station number	Target station side module station number	Connected station side MELSEC iQ-R series-compatible E71 station number	Connected station side MELSEC iQ-R series-compatible E71 station number
ActThroughNetworkType	0 (0x00)	<ul style="list-style-type: none"> • MELSECNET/10 is not included.: 0 (0x00) • MELSECNET/10 is included.: 1 (0x01) 			
ActTimeOut	10000	Any value specified by user in ms units			
ActUnitNumber	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station side module station number	Target station side module station number
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_GOT_RJ71EN71			

- *1 When the property pattern is ○(2), specify the connected station side MELSEC iQ-R series-compatible E71 station number set to the connected station side MELSEC iQ-R series-compatible E71 Ethernet parameter.
- *2 For the I/O address, specify the value of the actual start I/O number divided by 16.
- *3 When using OPS connection function at redundant CPU connection, set the port No. specified to "Host Station Port No." in Network parameter.
- *4 Specify the following value for the channel number to be multi-drop linked.
0: Default channel of module
1: Channel 1
2: Channel 2
- *5 When the property pattern is ○(1) or ○(2), specify the value set in the target station side parameter for ActNetworkNumber and ActStationNumber.
- *6 Specify the same network No. as MELSECNET/10 network No. set to MELSEC iQ-R series-compatible E71 in the Ethernet parameter settings of target station side MELSEC iQ-R series-compatible E71.
- *7 Specify the station number on the GOT side to avoid setting the same station number as set to the MELSEC iQ-R series-compatible E71 on the same Ethernet loop.

Personal computer side port: USB, GOT2000 side port: USB, CPU side port: Q series-compatible E71

Configuration



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

Accessibility and property patterns

○(n): Accessible ('n' is the property pattern number), ×: Inaccessible

Connected station CPU		Relayed network	Relayed station CPU				
QCPU (Q mode)	QCCPU		RCPU	RCCPU	R motion CPU	LHCPU	FX5CPU
○(1)*1	○(1)*4	CC IE TSN	×	×	×	×	×
		CC IE Control CC IE Field	×	×	×	×	×
		MELSECNET/H	×	×	×	×	×
		Ethernet	×	×	×	×	×
		Serial communication	×	×	×	×	×
		CC-Link	×	×	×	×	×

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

*1 Redundant CPU cannot be accessed.

*2 Since CC-Link IE Field Network is not supported, Q12DCCPU-V (Basic mode) relayed by CC-Link IE Field Network cannot be accessed.

*3 Since CC-Link IE Controller Network is not supported, LCPU relayed by CC-Link IE Controller Network cannot be accessed.

*4 Only CPU No.2 or later in a multiple CPU configuration can be accessed.

Property list

The following table shows the property settings of communication routes.

The setting is not necessary for those properties without description.

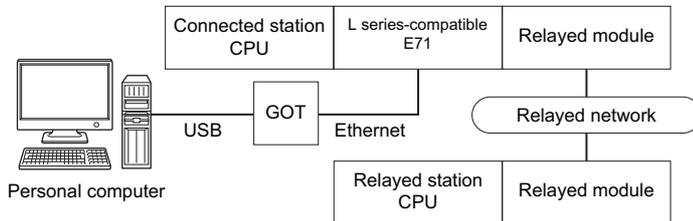
Property	Default value	Property patterns			
		○(1)	○(2)	○(3)	○(4)
ActConnectUnitNumber	0 (0x00)	Fixed to 0 (0x00)	Connected station side module station number	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActCpuType	34 (CPU_Q02CPU)	CPU type corresponding to the target station			
ActDestinationIONumber	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station side ■For single CPU • Fixed to 1023 (0x3FF) ■For multiple CPUs • Connected CPU: 1023 (0x3FF) • No. 1: 992 (0x3E0) • No. 2: 993 (0x3E1) • No. 3: 994 (0x3E2) • No. 4: 995 (0x3E3)	Target station side ■For single CPU • Fixed to 1023 (0x3FF) ■For multiple CPUs • Connected CPU: 1023 (0x3FF) • No. 1: 992 (0x3E0) • No. 2: 993 (0x3E1) • No. 3: 994 (0x3E2) • No. 4: 995 (0x3E3) ■For redundant CPU • Control system: 976 (0x3D0) • No specification: 1023 (0x3FF)
ActDestinationPortNumber	0 (0x00)	5001			
ActDidPropertyBit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActDsidPropertyBit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActHostAddress	1.1.1.1	Host name or IP address of the connected station side module			
ActIONumber ^{*1}	1023 (0x3FF)	■For single CPU • Fixed to 1023 (0x3FF) ■For multiple CPUs • Connected CPU: 1023 (0x3FF) • No. 1: 992 (0x3E0) • No. 2: 993 (0x3E1) • No. 3: 994 (0x3E2) • No. 4: 995 (0x3E3) ■For redundant CPU • Control system: 976 (0x3D0) • No specification: 1023 (0x3FF)	Target station side ■For single CPU • Fixed to 1023 (0x3FF) ■For multiple CPUs • Connected CPU: 1023 (0x3FF) • No. 1: 992 (0x3E0) • No. 2: 993 (0x3E1) • No. 3: 994 (0x3E2) • No. 4: 995 (0x3E3) ■For redundant CPU • Control system: 976 (0x3D0) • No specification: 1023 (0x3FF)	Connected station side module I/O address	Connected station side module I/O address
ActMultiDropChannelNumber ^{*2}	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Multi-drop channel number	Fixed to 0 (0x00)
ActNetworkNumber ^{*3}	0 (0x00)	Target station side module network number	Target station side module network number	Connected station side Q series-compatible E71 network number	Connected station side Q series-compatible E71 network number
ActProtocolType	0x04 (PROTOCOL_SERIAL)	PROTOCOL_USBGOT (0x13)			
ActSourceNetworkNumber	0 (0x00)	GOT side network number			
ActSourceStationNumber ^{*4}	0 (0x00)	GOT side station number			
ActStationNumber ^{*3}	255 (0xFF)	Target station side module station number	Target station side module station number	Connected station side Q series-compatible E71 station number	Connected station side Q series-compatible E71 station number
ActThroughNetworkType	0 (0x00)	• MELSECNET/10 is not included.: 0 (0x00) • MELSECNET/10 is included.: 1 (0x01)			
ActTimeOut	10000	Any value specified by user in ms units			
ActUnitNumber	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station side module station number	Target station side module station number

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

- *1 For the I/O address, specify the value of the actual start I/O number divided by 16.
- *2 Specify the following value for the channel number to be multi-drop linked.
 - 0: Default channel of module
 - 1: Channel 1
 - 2: Channel 2
- *3 When the property pattern is ○(1) or ○(2), specify the value set in the target station side parameter for ActNetworkNumber and ActStationNumber.
- *4 Specify the station number on the GOT side to avoid setting the same station number as set to the Q series-compatible E71 on the same Ethernet loop.

Personal computer side port: USB, GOT2000 side port: USB, CPU side port: L series-compatible E71

Configuration



The communication cannot be established when a remote password is set to the connected station side L series-compatible E71.

Accessibility and property patterns

○(n): Accessible ('n' is the property pattern number), ×: Inaccessible

Connected station CPU	Relayed network	Relayed station CPU				
		RCPU	RCCPU	R motion CPU	LHCPU	FX5CPU
○(1) ^{*1}	CC IE TSN	×	×	×	×	×
	CC IE Control CC IE Field	×	×	×	×	×
	MELSECNET/H	×	×	×	×	×
	Ethernet	×	×	×	×	×
	Serial communication	×	×	×	×	×
	CC-Link	×	×	×	×	×

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

*1 Redundant CPU cannot be accessed.

*2 Since CC-Link IE Field Network is not supported, Q12DCCPU-V (Basic mode) relayed by CC-Link IE Field Network cannot be accessed.

*3 Since CC-Link IE Controller Network is not supported, LCPUs relayed by CC-Link IE Controller Network cannot be accessed.

*4 For redundant CPU, serial communication modules on the main base unit cannot be accessed because the multi-drop connection cannot be performed.

Property list

The following table shows the property settings of communication routes.

The setting is not necessary for those properties without description.

Property	Default value	Property patterns			
		○(1)	○(2)	○(3)	○(4)
ActBaudRate	19200 (BAUDRATE_19200)	Fixed to 0 (0x00)			
ActConnectUnitNumber ^{*1}	0 (0x00)	Fixed to 0 (0x00)	Connected station side module station number	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActControl	8 (TRC_DTR_OR_RT S)	Fixed to 0 (0x00)			
ActCpuType	34 (CPU_Q02CPU)	CPU type corresponding to the target station			
ActDataBits	8 (DATABIT_8)	Fixed to 0 (0x00)			
ActDestinationIONumber	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station side ■For single CPU • Fixed to 1023 (0x3FF) ■For multiple CPUs • Connected CPU: 1023 (0x3FF) • No. 1: 992 (0x3E0) • No. 2: 993 (0x3E1) • No. 3: 994 (0x3E2) • No. 4: 995 (0x3E3)	Target station side ■For single CPU • Fixed to 1023 (0x3FF) ■For multiple CPUs • Connected CPU: 1023 (0x3FF) • No. 1: 992 (0x3E0) • No. 2: 993 (0x3E1) • No. 3: 994 (0x3E2) • No. 4: 995 (0x3E3) ■For redundant CPU • Control system: 976 (0x3D0) • No specification: 1023 (0x3FF)
ActDestinationPortNumber ^{*2}	0 (0x00)	5001 Host station port number for OPS connection			
ActDidPropertyBit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActDsidPropertyBit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActHostAddress	1.1.1.1	Host name or IP address of the connected station side module			
ActIONumber ^{*3}	1023 (0x3FF)	■For single CPU • Fixed to 1023 (0x3FF) ■For multiple CPUs • Connected CPU: 1023 (0x3FF) • No. 1: 992 (0x3E0) • No. 2: 993 (0x3E1) • No. 3: 994 (0x3E2) • No. 4: 995 (0x3E3) ■For redundant CPU • Control system: 976 (0x3D0) • No specification: 1023 (0x3FF)	Target station side ■For single CPU • Fixed to 1023 (0x3FF) ■For multiple CPUs • Connected CPU: 1023 (0x3FF) • No. 1: 992 (0x3E0) • No. 2: 993 (0x3E1) • No. 3: 994 (0x3E2) • No. 4: 995 (0x3E3) ■For redundant CPU • Control system: 976 (0x3D0) • No specification: 1023 (0x3FF)	Connected station side module I/O address	Connected station side module I/O address
ActMultiDropChannelNumber ^{*4}	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Multi-drop channel number	Fixed to 0 (0x00)
ActNetworkNumber ^{*5}	0 (0x00)	Target station side module network number	Target station side module network number	Connected station side L series-compatible E71 network number	Connected station side L series-compatible E71 network number
ActParity	1 (ODD_PARITY)	Fixed to 0 (0x00)			
ActPortNumber	1 (PORT_1)	Fixed to 0 (0x00)			
ActProtocolType	0x04 (PROTOCOL_SERI AL)	PROTOCOL_USBGOT (0x13)			
ActSourceNetworkNumber ^{*6}	0 (0x00)	GOT side network number			
ActSourceStationNumber ^{*7}	0 (0x00)	GOT side station number			

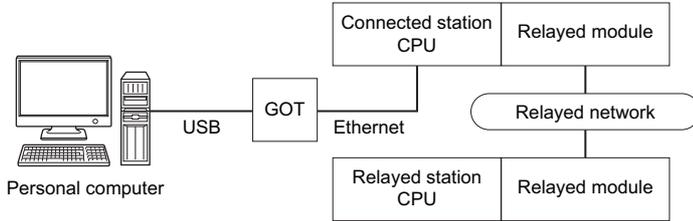
Property	Default value	Property patterns			
		○(1)	○(2)	○(3)	○(4)
ActStationNumber ^{*5}	255 (0xFF)	Target station side module station number	Target station side module station number	Connected station side L series-compatible E71 station number	Connected station side L series-compatible E71 station number
ActThroughNetworkType	0 (0x00)	<ul style="list-style-type: none"> • MELSECNET/10 is not included.: 0 (0x00) • MELSECNET/10 is included.: 1 (0x01) 			
ActTimeOut	10000	Any value specified by user in ms units			
ActUnitNumber	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station side module station number	Target station side module station number
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_GOT_LJ71E71 (0x5D)			

- *1 When the property pattern is ○(2), specify the connected station side L series-compatible E71 Ethernet parameter.
- *2 When using OPS connection function at redundant CPU connection, set the port No. specified to "Host Station Port No." in Network parameter.
- *3 For the I/O address, specify the value of the actual start I/O number divided by 16.
- *4 Specify the following value for the channel number to be multi-drop linked.
0: Default channel of module
1: Channel 1
2: Channel 2
- *5 When the property pattern is ○(1) or ○(2), specify the value set in the target station side parameter for ActNetworkNumber and ActStationNumber.
- *6 Specify the same network No. as MELSECNET/10 network No. set to L series-compatible E71 in the Ethernet parameter settings of target station side L series-compatible E71.
- *7 Specify the station number on the GOT side to avoid setting the same station number as set to the L series-compatible E71 on the same Ethernet loop.

Personal computer side port: USB, GOT2000 side port: USB, CPU side port: Ethernet port (1)

When the connected station is an RCP

Configuration



Accessibility and property patterns

○(n): Accessible ('n' is the property pattern number), ×: Inaccessible

Connected station CPU			Relayed network	Relayed station CPU				
RCPU	RCCPU	R motion CPU		RCPU	RCCPU	R motion CPU	LHCPU	FX5CPU
○(1)	○(1) ^{*3}	○(1) ^{*1*3}	CC IE TSN	×	×	×	×	×
			CC IE Control CC IE Field	○(2) ^{*2}	○(2)	×	×	○(2) ^{*4*5}
			MELSECNET/H	×	×	×	×	×
			Ethernet	○(2)	×	×	×	×
			Serial communication	○(3)	×	×	×	×
			CC-Link	○(4)	○(4)	×	×	○(4) ^{*5}

Connected station CPU			Relayed network	Relayed station CPU					
RCPU	RCCPU	R motion CPU		QCPU (Q mode)	QCCPU	LCPU	QSCPU	Q motion CPU	FXCPU
○(1)	○(1) ^{*3}	○(1) ^{*1*3}	CC IE TSN	×	×	×	×	×	×
			CC IE Control CC IE Field	×	×	×	×	×	×
			MELSECNET/H	×	×	×	×	×	×
			Ethernet	×	×	○(2)	×	×	×
			Serial communication	×	×	×	×	×	×
			CC-Link	×	×	×	×	×	×

*1 Relayed station CPUs cannot be accessed via R motion CPU.

*2 Since CC-Link IE Controller Network is not supported, RCP and LCP relayed by CC-Link IE Controller Network cannot be accessed.

*3 Only CPU No.2 or later in a multiple CPU configuration can be accessed.

*4 Only CC-Link IE Field Network on which connected station CPU is RCP can be accessed.

*5 FX5SCPU cannot be accessed.

Property list

The following table shows the property settings of communication routes.

The setting is not necessary for those properties without description.

Property	Default value	Property patterns			
		○(1)	○(2) ^{*1}	○(3)	○(4)
ActCpuType	34 (CPU_Q02CPU)	CPU type corresponding to the target station			
ActDestinationIONumber ^{*2}	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station side ■For single CPU • Fixed to 1023 (0x3FF) ■For multiple CPUs • Connected CPU: 1023 (0x3FF) • No. 1: 992 (0x3E0) • No. 2: 993 (0x3E1) • No. 3: 994 (0x3E2) • No. 4: 995 (0x3E3)	Target station side ■For single CPU • Fixed to 1023 (0x3FF) ■For multiple CPUs • Connected CPU: 1023 (0x3FF) • No. 1: 992 (0x3E0) • No. 2: 993 (0x3E1) • No. 3: 994 (0x3E2) • No. 4: 995 (0x3E3)
ActDidPropertyBit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActDsidPropertyBit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActHostAddress	1.1.1.1	Host name or IP address of the connected station side module			
ActIntelligentPreferenceBit	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station RCPU, QCPU (Q mode), LCPU: 1 (0x01) Other than the above: 0 (0x00)	Target station RCPU, QCPU (Q mode), LCPU: 1 (0x01) Other than the above: 0 (0x00)
ActIONumber ^{*2}	1023 (0x3FF)	■For single CPU • Fixed to 1023 (0x3FF) ■For multiple CPUs • Connected CPU: 1023 (0x3FF) • No. 1: 992 (0x3E0) • No. 2: 993 (0x3E1) • No. 3: 994 (0x3E2) • No. 4: 995 (0x3E3)	Target station side ■For single CPU • Fixed to 1023 (0x3FF) ■For multiple CPUs • Connected CPU: 1023 (0x3FF) • No. 1: 992 (0x3E0) • No. 2: 993 (0x3E1) • No. 3: 994 (0x3E2) • No. 4: 995 (0x3E3)	Connected station side relayed module I/O address	Connected station side relayed module I/O address
ActMultiDropChannelNumber ^{*3}	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Multi-drop channel number	Fixed to 0 (0x00)
ActNetworkNumber ^{*4}	0 (0x00)	Fixed to 0 (0x00)	Target station side module network number	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActProtocolType	0x04 (PROTOCOL_SERIAL)	PROTOCOL_USBGOT			
ActPacketType	0x01 (PACKET_PLC1)	PACKET_PLC1			
ActConnectUnitNumber	0 (0x00)	Fixed to 0 (0x00)			
ActPortNumber	1 (PORT_1)	Fixed to 0 (0x00)			
ActBaudRate	19200 (BAUDRATE_19200)	Fixed to 0 (0x00)			
ActCpuTimeOut	0 (0x00)	Fixed to 0 (0x00)			
ActControl	8 (TRC_DTR_OR_RTS)	Fixed to 0 (0x00)			
ActDataBits	8 (DATABIT_8)	Fixed to 0 (0x00)			
ActParity	1 (ODD_PARITY)	Fixed to 0 (0x00)			
ActStopBits	0 (STOPBIT_ONE)	Fixed to 0 (0x00)			
ActSumCheck	0 (NO_SUM_CHECK)	Fixed to 0 (0x00)			
ActSourceStationNumber	0 (0x00)	Fixed to 0 (0x00)			
ActDestinationPortNumber	0 (0x00)	5006			

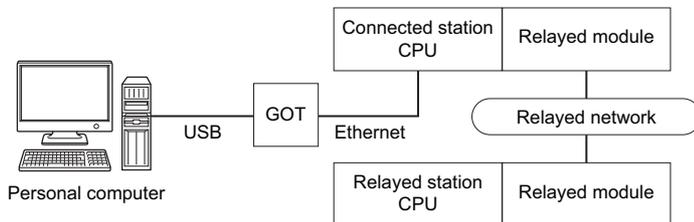
Property	Default value	Property patterns			
		○(1)	○(2) ^{*1}	○(3)	○(4)
ActStationNumber ^{*4}	255 (0xFF)	Fixed to 255 (0xFF)	Target station side module station number	Fixed to 255 (0xFF)	Fixed to 255 (0xFF)
ActThroughNetworkType	0 (0x00)	<ul style="list-style-type: none"> • MELSECNET/10 is not included.: 0 (0x00) • MELSECNET/10 is included.: 1 (0x01) 			MELSECNET/10 is not included.: 0 (0x00)
ActTimeOut	10000	Any value specified by user in ms units			
ActUnitNumber	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station side module station number	Target station side module station number
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_GOT_RETHET			

- *1 Note the following considerations when accessing via Ethernet module (MELSEC iQ-R series-compatible E71).
 - For ActNetworkNumber and ActStationNumber, specify the value set in the parameter setting of the target station side MELSEC iQ-R series-compatible E71.
 - Set "MNET/10 routing information" to MELSEC iQ-R series-compatible E71 parameter setting. When setting parameters, specify other than Automatic Response System (any of IP address calculation system, table conversion system, or combined system) for "MNET/10 routing method."
- *2 For the I/O address, specify the value of the actual start I/O number divided by 16.
- *3 Specify the following value for the channel number to be multi-drop linked.
 - 0: Default channel of module
 - 1: Channel 1
 - 2: Channel 2
- *4 When the property pattern is ○(2), specify the value set in the target station side parameter for ActNetworkNumber and ActStationNumber.

Personal computer side port: USB, GOT2000 side port: USB, CPU side port: Ethernet port (2)

When the connected station is an FX5CPU

Configuration



Accessibility and property patterns

○(n): Accessible ('n' is the property pattern number), ×: Inaccessible

Connected station CPU	Relayed network	Relayed station CPU				
		RCPU	RCCPU	R motion CPU	LHCPU	FX5CPU
○(1)	CC IE TSN	×	×	×	×	×
	CC IE Control CC IE Field	×	×	×	×	○(2) ^{*1*2}
	MELSECNET/H	×	×	×	×	×
	Ethernet	×	×	×	×	×
	Serial communication	×	×	×	×	×
	CC-Link	×	×	×	×	○(3) ^{*2}

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

*1 Only CC-Link IE Field Network can be accessed.

*2 FX5SCPU cannot be accessed.

Property list

The following table shows the property settings of communication routes.

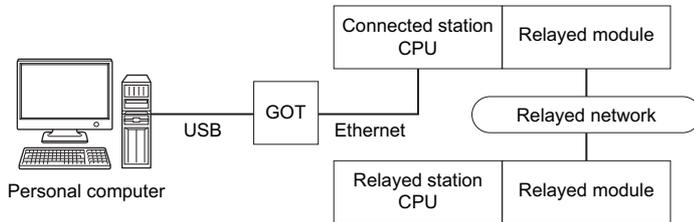
The setting is not necessary for those properties without description.

Property	Default value	Property patterns		
		○(1)	○(2)	○(3)
ActBaudRate	19200 (BAUDRATE_19200)	0 (0x00)		
ActConnectUnitNumber	0 (0x00)	0 (0x00)		
ActControl	8 (TRC_DTR_OR_RTS)	0 (0x00)		
ActCpuTimeOut	0 (0x00)	0 (0x00)		
ActCpuType	34 (CPU_Q02CPU)	CPU type corresponding to the target station		
ActDataBits	8 (DATABIT_8)	0 (0x00)		
ActDestinationIONumber	0 (0x00)	0 (0x00)	1023 (0x3FF)	
ActDestinationPortNumber	0 (0x00)	5562		
ActDidPropertyBit	1 (0x01)	1 (0x01)	0 (0x00)	
ActDsidPropertyBit	1 (0x01)	1 (0x01)	0 (0x00)	
ActHostAddress	1.1.1.1	Host name or IP address of the connected station side module		IP address of a connected station CPU
ActIntelligentPreferenceBit	0 (0x00)	0 (0x00)	1 (0x01)	
ActIONumber	1023 (0x3FF)	1023 (0x3FF)	Module number of the connected station	
ActMultiDropChannelNumber	0 (0x00)	0 (0x00)		
ActNetworkNumber	0 (0x00)	0 (0x00)	Target station side module network number	0 (0x00)
ActPacketType	0x01 (PACKET_PLC1)	PACKET_PLC1 (0x01)		
ActParity	1 (ODD_PARITY)	0 (0x00)		
ActPassword	Null	Null		
ActPortNumber	1 (PORT_1)	1 (0x01)		
ActProtocolType	0x04 (PROTOCOL_SERIAL)	PROTOCOL_USBGOT (0x13)		
ActStopBits	0 (STOPBIT_ONE)	0 (0x00)		
ActSumCheck	0 (NO_SUM_CHECK)	0 (0x00)		
ActSourceNetworkNumber	0 (0x00)	0 (0x00)		
ActSourceStationNumber	0 (0x00)	0 (0x00)		
ActStationNumber	255 (0xFF)	255 (0xFF)	Target station side module station number	255 (0xFF)
ActTargetSimulator	0 (0x00)	0 (0x00)		
ActThroughNetworkType	0 (0x00)	0 (0x00)		
ActTimeOut	10000	Any value specified by user in ms units		
ActUnitNumber	0 (0x00)	0 (0x00)	Target station side module station number	
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_GOT_FXVETHER (0x2007)		

Personal computer side port: USB, GOT2000 side port: USB, CPU side port: Ethernet port (3)

When the connected station is a QCPU (Q mode) or QCCPU

Configuration



Accessibility and property patterns

○(n): Accessible ('n' is the property pattern number), ×: Inaccessible

Connected station CPU ^{*1}	Relayed network	Relayed station CPU				
		RCPU	RCCPU	R motion CPU	LHCPU	FX5CPU
○(1)	CC IE TSN	×	×	×	×	×
	CC IE Control CC IE Field	×	×	×	×	×
	MELSECNET/H	×	×	×	×	×
	Ethernet	×	×	×	×	×
	Serial communication	×	×	×	×	×
	CC-Link	×	×	×	×	×

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

Connected station CPU ^{*1}	Relayed network	Relayed station CPU				
		RCPU	RCCPU	R motion CPU	LHCPU	FX5CPU
○(1)	CC IE TSN	×	×	×	×	×
	CC IE Control CC IE Field	×	×	×	×	×
	MELSECNET/H	×	×	×	×	×
	Ethernet	×	×	×	×	×
	Serial communication	×	×	×	×	×
	CC-Link	×	×	×	×	×

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

*1 The communication cannot be established when a remote password is set to the connected station CPU.

*2 Redundant CPU cannot be accessed.

*3 Since CC-Link IE Field Network is not supported, Q12DCCPU-V (Basic mode) relayed by CC-Link IE Field Network cannot be accessed.

*4 Since CC-Link IE Controller Network is not supported, LCPU relayed by CC-Link IE Controller Network cannot be accessed.

Property list

The following table shows the property settings of communication routes.

The setting is not necessary for those properties without description.

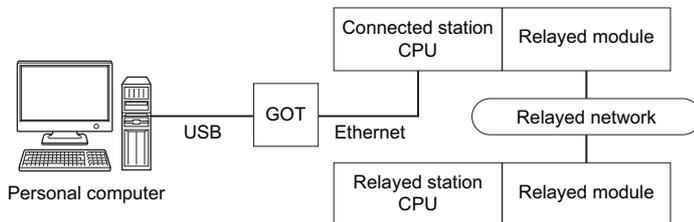
Property	Default value	Property patterns			
		○(1)	○(2) ^{*1}	○(3)	○(4)
ActCpuType	34 (CPU_Q02CPU)	CPU type corresponding to the target station			
ActDestinationIONumber	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station side ■For single CPU • Fixed to 1023 (0x3FF) ■For multiple CPUs • Connected CPU: 1023 (0x3FF) • No. 1: 992 (0x3E0) • No. 2: 993 (0x3E1) • No. 3: 994 (0x3E2) • No. 4: 995 (0x3E3)	Target station side ■For single CPU • Fixed to 1023 (0x3FF) ■For multiple CPUs • Connected CPU: 1023 (0x3FF) • No. 1: 992 (0x3E0) • No. 2: 993 (0x3E1) • No. 3: 994 (0x3E2) • No. 4: 995 (0x3E3) ■For redundant CPU • Control system: 976 (0x3D0) • No specification: 1023 (0x3FF)
ActDidPropertyBit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActDsidPropertyBit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActHostAddress	1.1.1.1	Host name or IP address of the connected station side module			
ActIntelligentPreferenceBit	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station QCPU (Q mode), LCPU: 1 (0x01) Other than the above: 0 (0x00)	Target station QCPU (Q mode), QCCPU, LCPU: 1 (0x01) Other than the above: 0 (0x00)
ActIONumber ^{*2}	1023 (0x3FF)	■For single CPU • Fixed to 1023 (0x3FF) ■For multiple CPUs • Connected CPU: 1023 (0x3FF) • No. 1: 992 (0x3E0) • No. 2: 993 (0x3E1) • No. 3: 994 (0x3E2) • No. 4: 995 (0x3E3)	Target station side ■For single CPU • Fixed to 1023 (0x3FF) ■For multiple CPUs • Connected CPU: 1023 (0x3FF) • No. 1: 992 (0x3E0) • No. 2: 993 (0x3E1) • No. 3: 994 (0x3E2) • No. 4: 995 (0x3E3) ■For redundant CPU • Control system: 976 (0x3D0) • No specification: 1023 (0x3FF)	Connected station side relayed module I/O address	Connected station side relayed module I/O address
ActMultiDropChannelNumber ^{*3}	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Multi-drop channel number	Fixed to 0 (0x00)
ActNetworkNumber ^{*4}	0 (0x00)	Fixed to 0 (0x00)	Target station side module network number	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActProtocolType	0x04 (PROTOCOL_SERIAL)	PROTOCOL_USBGOT (0x13)			
ActStationNumber ^{*4}	255 (0xFF)	Fixed to 255 (0xFF)	Target station side module station number	Fixed to 255 (0xFF)	Fixed to 255 (0xFF)
ActThroughNetworkType	0 (0x00)	• MELSECNET/10 is not included.: 0 (0x00) • MELSECNET/10 is included.: 1 (0x01)			
ActTimeOut	10000	Any value specified by user in ms units			
ActUnitNumber	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station side module station number	Target station side module station number
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_GOT_QNETHER (0x41)			

- *1 Note the following considerations when accessing via Ethernet module (Q series-compatible E71).
 - For ActNetworkNumber and ActStationNumber, specify the value set in the parameter setting of the target station side Q series-compatible E71.
 - Set "Station No. ⇔ IP information" in the parameter setting of Q series-compatible E71. When setting the parameter, specify the setting other than the automatic response system (any of the IP address calculation system, table conversion system, and combined system) for "Station No. ⇔ IP information system."
- *2 For the I/O address, specify the value of the actual start I/O number divided by 16.
- *3 Specify the following value for the channel number to be multi-drop linked.
 - 0: Default channel of module
 - 1: Channel 1
 - 2: Channel 2
- *4 When the property pattern is ○(2), specify the value set in the target station side parameter for ActNetworkNumber and ActStationNumber.

Personal computer side port: USB, GOT2000 side port: USB, CPU side port: Ethernet port (4)

When the connected station is an LCPU

Configuration



Accessibility and property patterns

○(n): Accessible ('n' is the property pattern number), ×: Inaccessible

Connected station CPU ^{*1}	Relayed network	Relayed station CPU				
		RCPU	RCCPU	R motion CPU	LHCPU	FX5CPU
○(1)	CC IE TSN	×	×	×	×	×
	CC IE Field ^{*3}	×	×	×	×	×
	MELSECNET/H	×	×	×	×	×
	Ethernet	×	×	×	×	×
	Serial communication	×	×	×	×	×
	CC-Link	×	×	×	×	×

Connected station CPU ^{*1}	Relayed network	Relayed station CPU					
		QCPU (Q mode) ^{*2}	QCCPU	LCPU	QSCPU	Q motion CPU	FXCPU
○(1)	CC IE TSN	×	×	×	×	×	×
	CC IE Field ^{*3}	○(4)	○(4) ^{*4}	○(4)	×	×	×
	MELSECNET/H	×	×	×	×	×	×
	Ethernet	○(4)	×	○(4)	×	×	×
	Serial communication	○(2)	×	○(2)	×	×	×
	CC-Link	○(3)	○(3)	○(3)	×	×	×

*1 The communication cannot be established when a remote password is set to the connected station CPU.

*2 Redundant CPU cannot be accessed.

*3 Since CC-Link IE Controller Network is not supported, LCPU relayed by CC-Link IE Controller Network cannot be accessed.

*4 Since CC-Link IE Field Network is not supported, Q12DCCPU-V (Basic mode) relayed by CC-Link IE Field Network cannot be accessed.

Property list

The following table shows the property settings of communication routes.

The setting is not necessary for those properties without description.

Property	Default value	Property patterns			
		○(1)	○(2)	○(3)	○(4)
ActBaudRate	19200 (BAUDRATE_19200)	Fixed to 0 (0x00)			
ActControl	8 (TRC_DTR_OR_RT S)	Fixed to 0 (0x00)			
ActCpuType	34 (CPU_Q02CPU)	CPU type corresponding to the target station			
ActDataBits	8 (DATABIT_8)	Fixed to 0 (0x00)			
ActDestinationIONumber	0 (0x00)	Fixed to 0 (0x00)	Target station side ■For single CPU • Fixed to 1023 (0x3FF) ■For multiple CPUs • Connected CPU: 1023 (0x3FF) • No. 1: 992 (0x3E0) • No. 2: 993 (0x3E1) • No. 3: 994 (0x3E2) • No. 4: 995 (0x3E3)	Target station side ■For single CPU • Fixed to 1023 (0x3FF) ■For multiple CPUs • Connected CPU: 1023 (0x3FF) • No. 1: 992 (0x3E0) • No. 2: 993 (0x3E1) • No. 3: 994 (0x3E2) • No. 4: 995 (0x3E3) ■For redundant CPU • Control system: 976 (0x3D0) • No specification: 1023 (0x3FF)	Fixed to 0 (0x00)
ActDestinationPortNumber	0 (0x00)	5006			
ActDidPropertyBit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Fixed to 1 (0x01)
ActDsidPropertyBit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Fixed to 1 (0x01)
ActHostAddress	1.1.1.1	Host name or IP address of the connected station side module			
ActIntelligentPreferenceBit	0 (0x00)	Fixed to 0 (0x00)	Target station QCPU (Q mode), LCPU: 1 (0x01) Other than the above: 0 (0x00)	Target station QCPU (Q mode), QCCPU, LCPU: 1 (0x01) Other than the above: 0 (0x00)	Fixed to 0 (0x00)
ActIONumber*1	1023 (0x3FF)	Fixed to 1023 (0x3FF)	Connected station side relayed module I/O address	Connected station side relayed module I/O address	Target station side ■For single CPU • Fixed to 1023 (0x3FF) ■For multiple CPUs • Connected CPU: 1023 (0x3FF) • No. 1: 992 (0x3E0) • No. 2: 993 (0x3E1) • No. 3: 994 (0x3E2) • No. 4: 995 (0x3E3) ■For redundant CPU • Control system: 976 (0x3D0) • No specification: 1023 (0x3FF)
ActMultiDropChannelNumber*2	0 (0x00)	Fixed to 0 (0x00)	Multi-drop channel number	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActNetworkNumber	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station side module network number
ActParity	1 (ODD_PARITY)	Fixed to 0 (0x00)			
ActPortNumber	1 (PORT_1)	Fixed to 0 (0x00)			
ActProtocolType	0x04 (PROTOCOL_SERI AL)	PROTOCOL_USBGOT (0x13)			

Property	Default value	Property patterns			
		○(1)	○(2)	○(3)	○(4)
ActStationNumber	255 (0xFF)	Fixed to 255 (0xFF)	Fixed to 255 (0xFF)	Fixed to 255 (0xFF)	Target station side module station number
ActThroughNetworkType	0 (0x00)	<ul style="list-style-type: none"> • MELSECNET/10 is not included.: 0 (0x00) • MELSECNET/10 is included.: 1 (0x01) 			
ActTimeOut	10000	Any value specified by user in ms units			
ActUnitNumber	0 (0x00)	Fixed to 0 (0x00)	Target station side module station number	Target station side module station number	Fixed to 0 (0x00)
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_GOT_LNETHER (0x55)			

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

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

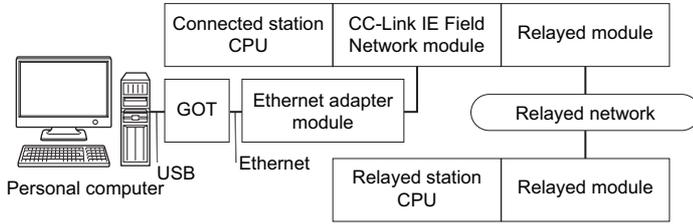
0: Default channel of module

1: Channel 1

2: Channel 2

Personal computer side port: USB, GOT2000 side port: USB, CPU side port: Ethernet adapter module

Configuration



Accessibility and property patterns

○(n): Accessible ('n' is the property pattern number), ×: Inaccessible

Connected station CPU	Relayed network	Relayed station CPU				
		RCPU	RCCPU	R motion CPU	LHCPU	FX5CPU
QnUDE(H)CPU ○(1)	CC IE TSN	×	×	×	×	×
	CC IE Control CC IE Field	×	×	×	×	×
	MELSECNET/H	×	×	×	×	×
	Ethernet	×	×	×	×	×
	Serial communication	×	×	×	×	×
	CC-Link	×	×	×	×	×

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

Connected station CPU	Relayed network	Relayed station CPU				
		RCPU	RCCPU	R motion CPU	LHCPU	FX5CPU
LCP ○(1)	CC IE TSN	×	×	×	×	×
	CC IE Field*3	×	×	×	×	×
	MELSECNET/H	×	×	×	×	×
	Ethernet	×	×	×	×	×
	Serial communication	×	×	×	×	×
	CC-Link	×	×	×	×	×

Connected station CPU	Relayed network	Relayed station CPU					
		QCPU (Q mode) ^{*1}	QCCPU	LCPU	QSCPU	Q motion CPU	FXCPU
○(1)	CC IE TSN	×	×	×	×	×	×
	CC IE Field ^{*3}	○(2)	×	○(2)	×	×	×
	MELSECNET/H	×	×	×	×	×	×
	Ethernet	×	×	×	×	×	×
	Serial communication	○(3)	×	○(3)	×	×	×
	CC-Link	○(4)	○(4)	○(4)	×	×	×

*1 Redundant CPU cannot be accessed.

*2 Since CC-Link IE Field Network is not supported, Q12DCCPU-V (Basic mode) relayed by CC-Link IE Field Network cannot be accessed.

*3 Since CC-Link IE Controller Network is not supported, LCPUs relayed by CC-Link IE Controller Network cannot be accessed.

Property list

The following table shows the property settings of communication routes.

The setting is not necessary for those properties without description.

Property	Default value	Property patterns			
		○(1)	○(2)	○(3)	○(4)
ActCpuType	34 (CPU_Q02CPU)	CPU type corresponding to the target station			
ActDestinationIONumber	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station side ■For single CPU • Fixed to 1023 (0x3FF) ■For multiple CPUs • Connected CPU: 1023 (0x3FF) • No. 1: 992 (0x3E0) • No. 2: 993 (0x3E1) • No. 3: 994 (0x3E2) • No. 4: 995 (0x3E3)	Target station side ■For single CPU • Fixed to 1023 (0x3FF) ■For multiple CPUs • Connected CPU: 1023 (0x3FF) • No. 1: 992 (0x3E0) • No. 2: 993 (0x3E1) • No. 3: 994 (0x3E2) • No. 4: 995 (0x3E3) ■For redundant CPU • Control system: 976 (0x3D0) • No specification: 1023 (0x3FF)
ActDidPropertyBit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActDsidPropertyBit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActHostAddress	1.1.1.1	Host name or IP address of the Ethernet adapter module			
ActIntelligentPreferenceBit	0 (0x00)	Fixed to 0 (0x00)			
ActIONumber ^{*1}	1023 (0x3FF)	■For single CPU • Fixed to 1023 (0x3FF) ■For multiple CPUs • Connected CPU: 1023 (0x3FF) • No. 1: 992 (0x3E0) • No. 2: 993 (0x3E1) • No. 3: 994 (0x3E2) • No. 4: 995 (0x3E3)	Target station side ■For single CPU • Fixed to 1023 (0x3FF) ■For multiple CPUs • Connected CPU: 1023 (0x3FF) • No. 1: 992 (0x3E0) • No. 2: 993 (0x3E1) • No. 3: 994 (0x3E2) • No. 4: 995 (0x3E3) ■For redundant CPU • Control system: 976 (0x3D0) • No specification: 1023 (0x3FF)	Connected station side relayed module I/O address	Connected station side relayed module I/O address
ActMultiDropChannelNumber ^{*2}	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Multi-drop channel number	Fixed to 0 (0x00)
ActNetworkNumber ^{*3}	0 (0x00)	Connected station side CC-Link IE Field Network module network number	Target station side module network number	Connected station side CC-Link IE Field Network module network number	Connected station side CC-Link IE Field Network module network number
ActProtocolType	0x04 (PROTOCOL_SERIAL)	PROTOCOL_USBGOT (0x13)			
ActStationNumber ^{*3}	255 (0xFF)	Connected station side CC-Link IE Field Network module station number	Target station side module station number	Connected station side CC-Link IE Field Network module station number	Connected station side CC-Link IE Field Network module station number
ActThroughNetworkType	0 (0x00)	• MELSECNET/10 is not included.: 0 (0x00) • MELSECNET/10 is included.: 1 (0x01)			
ActTimeOut	10000	Any value specified by user in ms units			
ActUnitNumber	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station side module station number	Target station side module station number
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_GOT_NZ2GF_ETB (0x5B)			

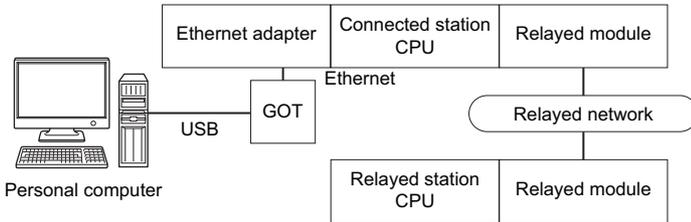
- *1 For the I/O address, specify the value of the actual start I/O number divided by 16.
- *2 Specify the following value for the channel number to be multi-drop linked.
 - 0: Default channel of module
 - 1: Channel 1
 - 2: Channel 2
- *3 For the property pattern of ○(1), ○(3), or ○(4), specify the parameter value set on the CC-Link IE Field Network module on the connected station side for ActNetworkNumber and ActStationNumber.
For the property pattern of ○(2), specify the value set in the target station side parameter for ActNetworkNumber and ActStationNumber.

Personal computer side port: USB, GOT2000 side port: USB, CPU side port: Ethernet adapter/module

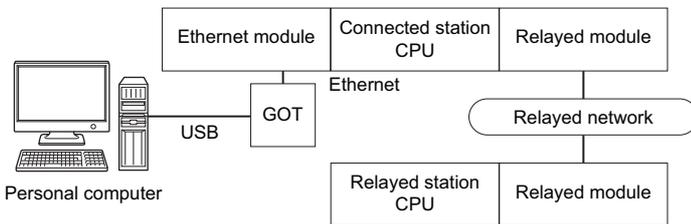
When the connected station is an FXCPU

Configuration

Ethernet adapter: FX3U-ENET-ADP



Ethernet module: FX3U-ENET



Accessibility and property patterns

○(n): Accessible ('n' is the property pattern number), ×: Inaccessible

Connected station CPU	Relayed network	Relayed station CPU				
		RCPU	RCCPU	R motion CPU	LHCPU	FX5CPU
FXCPU ○(1) ^{*1}	CC IE TSN	×	×	×	×	×
	CC IE Control CC IE Field	×	×	×	×	×
	MELSECNET/H	×	×	×	×	×
	Ethernet	×	×	×	×	×
	Serial communication	×	×	×	×	×
	CC-Link	×	×	×	×	×

Connected station CPU	Relayed network	Relayed station CPU					
		QCPU (Q mode)	QCCPU	LCPU	QSCPU	Q motion CPU	FXCPU
FXCPU ○(1) ^{*1}	CC IE TSN	×	×	×	×	×	×
	CC IE Control CC IE Field	×	×	×	×	×	×
	MELSECNET/H	×	×	×	×	×	×
	Ethernet	×	×	×	×	×	×
	Serial communication	×	×	×	×	×	×
	CC-Link	×	×	×	×	×	×

*1 Only FX3SCPU(FX3U-ENET-ADP), FX3G(C)CPU and FX3U(C)CPU can be accessed.

Property list

The following table shows the property settings of communication routes.

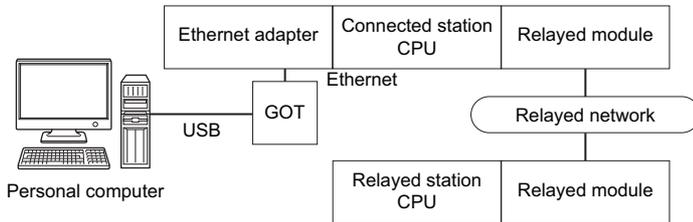
The setting is not necessary for those properties without description.

Property	Default value	Property patterns
		○(1)
ActCpuType	34 (CPU_Q02CPU)	CPU type corresponding to the target station
ActHostAddress	1.1.1.1	Host name or IP address of FX3U-ENET-ADP module
		Host name or IP address of FX3U-ENET module
ActProtocolType	0x04 (PROTOCOL_SERIAL)	PROTOCOL_USBGOT
ActTimeOut	10000	Any value specified by user in ms units
ActUnitType	0x13 (UNIT_QNCPU)	FX3U-ENET-ADP: UNIT_GOT_FXETHER
		FX3U-ENET: UNIT_GOT_FXENET

Personal computer side port: USB, GOT2000 side port: USB, CPU side port: Ethernet adapter

When a connection target CPU is an FX5CPU

Configuration



Accessibility and property patterns

○(n): Accessible ('n' is the property pattern number), ×: Inaccessible

Connected station CPU	Relayed network	Relayed station CPU				
		RCPU	RCCPU	R motion CPU	LHCPU	FX5CPU
×	CC IE TSN	×	×	×	×	×
	CC IE Control CC IE Field	×	×	×	×	×
	MELSECNET/H	×	×	×	×	×
	Ethernet	×	×	×	×	○(1)*1
	Serial communication	×	×	×	×	×
	CC-Link	×	×	×	×	×

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

*1 Only FX5UCPU and FX5UJCPU can be accessed.

Property list

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

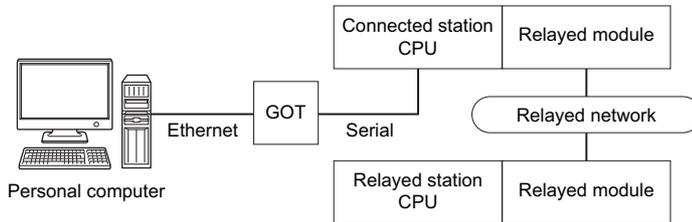
Property	Default value	Property patterns
		○(1)
ActConnectUnitNumber	0 (0x00)	Fixed to 0 (0x00)
ActCpuType	34 (CPU_Q02CPU)	CPU type corresponding to the target station
ActDestinationIONumber	0 (0x00)	Fixed to 0 (0x00)
ActDestinationPortNumber	0 (0x00)	Fixed to 5554
ActDidPropertyBit	1 (0x01)	1 (0x01) (invalid)
ActDsidPropertyBit	1 (0x01)	1 (0x01) (invalid)
ActHostAddress	1.1.1.1	Host name or IP address of a connection target CPU
ActIONumber	1023 (0x3FF)	Fixed to 1023 (0x3FF)

Property	Default value	Property patterns
		○(1)
ActMultiDropChannelNumber	0 (0x00)	Fixed to 0 (0x00)
ActNetworkNumber	0 (0x00)	Fixed to 0 (0x00)
ActProtocolType	0x04 (PROTOCOL_SERIAL)	PROTOCOL_USBGOT
ActPacketType	0x01 (PACKET_PLC1)	Fixed to PACKET_PLC1
ActPortNumber	1 (PORT_1)	PORT_1
ActBaudRate	19200 (BAUDRATE_19200)	Fixed to 0 (0x00)
ActCpuTimeOut	0 (0x00)	Fixed to 0 (0x00)
ActControl	8 (TRC_DTR_OR_RTS)	Fixed to 0 (0x00)
ActDataBits	8 (DATABIT_8)	Fixed to 0 (0x00)
ActParity	1 (ODD_PARITY)	Fixed to 0 (0x00)
ActStopBits	0 (STOPBIT_ONE)	Fixed to 0 (0x00)
ActSumCheck	0 (NO_SUM_CHECK)	Fixed to 0 (0x00)
ActIntelligentPreferenceBit	0 (0x00)	Fixed to 0 (0x00)
ActSourceNetworkNumber	0 (0x00)	Fixed to 0 (0x00)
ActSourceStationNumber	0 (0x00)	Fixed to 0 (0x00)
ActStationNumber	255 (0xFF)	Fixed to 255 (0xFF)
ActThroughNetworkType	0 (0x00)	Fixed to 0 (0x00)
ActTimeOut	10000	Any value specified by user in ms units
ActUnitNumber	0 (0x00)	Fixed to 0 (0x00)
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_GOT_FXVENET

Personal computer side port: Ethernet board, GOT2000 port: Ethernet port, CPU side port: Serial (1)

When the connected station is an FX5CPU

Configuration



Accessibility and property patterns

○(n): Accessible ('n' is the property pattern number), ×: Inaccessible

Connected station CPU	Relayed network	Relayed station CPU					
		FX5CPU	RCPU	RCCPU	R motion CPU	LHCPU	FX5CPU
○(1)	CC IE TSN	×	×	×	×	×	×
	CC IE Control CC IE Field	×	×	×	×	×	○(2) ^{*1*2}
	MELSECNET/H	×	×	×	×	×	×
	Ethernet	×	×	×	×	×	×
	Serial communication	×	×	×	×	×	×
	CC-Link	×	×	×	×	×	○(3) ^{*2}

Connected station CPU	Relayed network	Relayed station CPU						
		FX5CPU	QCPU (Q mode)	QCCPU	LCPU	QSCPU	Q motion CPU	FXCPU
○(1)	CC IE TSN	×	×	×	×	×	×	×
	CC IE Control CC IE Field	×	×	×	×	×	×	×
	MELSECNET/H	×	×	×	×	×	×	×
	Ethernet	×	×	×	×	×	×	×
	Serial communication	×	×	×	×	×	×	×
	CC-Link	×	×	×	×	×	×	×

*1 Only CC-Link IE Field Network can be accessed.

*2 FX5SCPU cannot be accessed.

Property list

The following table shows the property settings of communication routes.

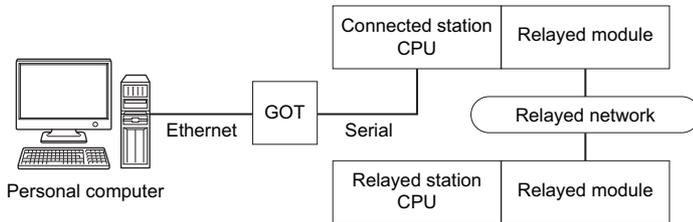
The setting is not necessary for those properties without description.

Property	Default value	Property patterns		
		○(1)	○(2)	○(3)
ActBaudRate	19200 (BAUDRATE_19200)	0 (0x00)		
ActConnectUnitNumber	0 (0x00)	0 (0x00)		
ActControl	8 (TRC_DTR_OR_RTS)	0 (0x00)		
ActCpuTimeOut	0 (0x00)	0 (0x00)		
ActCpuType	34 (CPU_Q02CPU)	CPU type corresponding to the target station		
ActDataBits	8 (DATABIT_8)	0 (0x00)		
ActDestinationIONumber	0 (0x00)	0 (0x00)		1023 (0x3FF)
ActDestinationPortNumber	0 (0x00)	GOT port number		
ActDidPropertyBit	1 (0x01)	1 (0x01)		0 (0x00)
ActDsidPropertyBit	1 (0x01)	1 (0x01)		0 (0x00)
ActHostAddress	1.1.1.1	Host name or IP address of GOT		
ActIntelligentPreferenceBit	0 (0x00)	0 (0x00)		1 (0x01)
ActIONumber	1023 (0x3FF)	1023 (0x3FF)		Module number of the connected station
ActMultiDropChannelNumber	0 (0x00)	0 (0x00)		
ActNetworkNumber	0 (0x00)	0 (0x00)	Target station side module network number	0 (0x00)
ActPacketType	0x01 (PACKET_PLC1)	PACKET_PLC1 (0x01)		
ActParity	1 (ODD_PARITY)	1 (0x01)		
ActPassword	Null	Null		
ActPortNumber	1 (PORT_1)	0 (0x00)		
ActProtocolType	0x04 (PROTOCOL_SERIAL)	PROTOCOL_TCPIP (0x05)		
ActStopBits	0 (STOPBIT_ONE)	0 (0x00)		
ActSumCheck	0 (NO_SUM_CHECK)	0 (0x00)		
ActSourceNetworkNumber	0 (0x00)	0 (0x00)		
ActSourceStationNumber	0 (0x00)	0 (0x00)		
ActStationNumber	255 (0xFF)	255 (0xFF)	Target station side module station number	255 (0xFF)
ActTargetSimulator	0 (0x00)	0 (0x00)		
ActThroughNetworkType	0 (0x00)	0 (0x00)		
ActTimeOut	10000	Any value specified by user in ms units		
ActUnitNumber	0 (0x00)	0 (0x00)		Target station side module station number
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_GOTETHER_FXVCPU (0x2006)		

Personal computer side port: Ethernet board, GOT2000 port: Ethernet port, CPU side port: Serial (2)

When the connected station is a QCPU (Q mode) or QCCPU

Configuration



Accessibility and property patterns

○(n): Accessible ('n' is the property pattern number), ×: Inaccessible

Connected station CPU		Relayed network	Relayed station CPU				
QCPU (Q mode)	QCCPU		RCPU	RCCPU	R motion CPU	LHCPU	FX5CPU
○(1) ^{*1}	○(1) ^{*4}	CC IE TSN	×	×	×	×	×
		CC IE Control CC IE Field	×	×	×	×	×
		MELSECNET/H	×	×	×	×	×
		Ethernet	×	×	×	×	×
		Serial communication	×	×	×	×	×
		CC-Link	×	×	×	×	×

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

*1 Redundant CPU cannot be accessed.

*2 Since CC-Link IE Field Network is not supported, Q12DCCPU-V (Basic mode) relayed by CC-Link IE Field Network cannot be accessed.

*3 Since CC-Link IE Controller Network is not supported, LCPU relayed by CC-Link IE Controller Network cannot be accessed.

*4 For Q24DHCCPU-V, Q24DHCCPU-LS, Q24DHCCPU-VG and Q26DHCCPU-LS, only CPU No.2 or later in a multiple CPU configuration can be accessed.

Property list

The following table shows the property settings of communication routes.

The setting is not necessary for those properties without description.

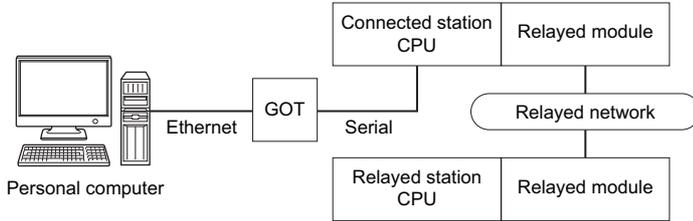
Property	Default value	Property patterns			
		○(1)	○(2)	○(3)	○(4)
ActCpuType	34 (CPU_Q02CPU)	CPU type corresponding to the target station			
ActDestinationIONumber	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station side ■For single CPU • Fixed to 1023 (0x3FF) ■For multiple CPUs • Connected CPU: 1023 (0x3FF) • No. 1: 992 (0x3E0) • No. 2: 993 (0x3E1) • No. 3: 994 (0x3E2) • No. 4: 995 (0x3E3)	Target station side ■For single CPU • Fixed to 1023 (0x3FF) ■For multiple CPUs • Connected CPU: 1023 (0x3FF) • No. 1: 992 (0x3E0) • No. 2: 993 (0x3E1) • No. 3: 994 (0x3E2) • No. 4: 995 (0x3E3) ■For redundant CPU • Control system: 976 (0x3D0) • No specification: 1023 (0x3FF)
ActDestinationPortNumber	0 (0x00)	GOT port number			
ActDidPropertyBit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActDsidPropertyBit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActHostAddress	1.1.1.1	Host name or IP address of GOT			
ActIntelligentPreferenceBit	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station QCPU (Q mode), LCPU: 1 (0x01) Other than the above: 0 (0x00)	Target station QCPU (Q mode), QCCPU, LCPU: 1 (0x01) Other than the above: 0 (0x00)
ActIONumber ^{*1}	1023 (0x3FF)	■For single CPU • Fixed to 1023 (0x3FF) ■For multiple CPUs • Connected CPU: 1023 (0x3FF) • No. 1: 992 (0x3E0) • No. 2: 993 (0x3E1) • No. 3: 994 (0x3E2) • No. 4: 995 (0x3E3)	Target station side ■For single CPU • Fixed to 1023 (0x3FF) ■For multiple CPUs • Connected CPU: 1023 (0x3FF) • No. 1: 992 (0x3E0) • No. 2: 993 (0x3E1) • No. 3: 994 (0x3E2) • No. 4: 995 (0x3E3) ■For redundant CPU • Control system: 976 (0x3D0) • No specification: 1023 (0x3FF)	Connected station side module I/O address	Connected station side module I/O address
ActMultiDropChannelNumber ^{*2}	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Multi-drop channel number	Fixed to 0 (0x00)
ActNetworkNumber	0 (0x00)	Fixed to 0 (0x00)	Target station side module network number	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActProtocolType	0x04 (PROTOCOL_SERIAL)	PROTOCOL_TCPIP (0x05)			
ActStationNumber	255 (0xFF)	Fixed to 255 (0xFF)	Target station side module station number	Fixed to 255 (0xFF)	Fixed to 255 (0xFF)
ActThroughNetworkType	0 (0x00)	• MELSECNET/10 is not included.: 0 (0x00) • MELSECNET/10 is included.: 1 (0x01)			
ActTimeOut	10000	Any value specified by user in ms units			
ActUnitNumber	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station side module station number	Target station side module station number
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_GOTETHER_QNCPU (0x56)			

- *1 For the I/O address, specify the value of the actual start I/O number divided by 16.
- *2 Specify the following value for the channel number to be multi-drop linked.
 - 0: Default channel of module
 - 1: Channel 1
 - 2: Channel 2

Personal computer side port: Ethernet board, GOT2000 port: Ethernet port, CPU side port: Serial (3)

When the connected station is an LCPU

Configuration



Accessibility and property patterns

○(n): Accessible ('n' is the property pattern number), ×: Inaccessible

Connected station CPU	Relayed network	Relayed station CPU				
		RCPU	RCCPU	R motion CPU	LHCPU	FX5CPU
○(1)	CC IE TSN	×	×	×	×	×
	CC IE Field ^{*2}	×	×	×	×	×
	MELSECNET/H	×	×	×	×	×
	Ethernet	×	×	×	×	×
	Serial communication	×	×	×	×	×
	CC-Link	×	×	×	×	×

Connected station CPU	Relayed network	Relayed station CPU					
		QCPU (Q mode) ^{*1}	QCCPU	LCPU	QSCPU	Q motion CPU	FXCPU
○(1)	CC IE TSN	×	×	×	×	×	×
	CC IE Field ^{*2}	○(4)	○(4) ^{*3}	○(4)	×	×	×
	MELSECNET/H	×	×	×	×	×	×
	Ethernet	×	×	×	×	×	×
	Serial communication	○(2)	×	○(2)	×	×	×
	CC-Link	○(3)	○(3)	○(3)	×	×	×

*1 Redundant CPU cannot be accessed.

*2 Since CC-Link IE Controller Network is not supported, LCPU relayed by CC-Link IE Controller Network cannot be accessed.

*3 Since CC-Link IE Field Network is not supported, Q12DCCPU-V (Basic mode) relayed by CC-Link IE Field Network cannot be accessed.

Property list

The following table shows the property settings of communication routes.

The setting is not necessary for those properties without description.

Property	Default value	Property patterns			
		○(1)	○(2)	○(3)	○(4)
ActCpuType	34 (CPU_Q02CPU)	CPU type corresponding to the target station			
ActDestinationIONumber	0 (0x00)	Fixed to 0 (0x00)	Target station side ■For single CPU • Fixed to 1023 (0x3FF) ■For multiple CPUs • Connected CPU: 1023 (0x3FF) • No. 1: 992 (0x3E0) • No. 2: 993 (0x3E1) • No. 3: 994 (0x3E2) • No. 4: 995 (0x3E3)	Target station side ■For single CPU • Fixed to 1023 (0x3FF) ■For multiple CPUs • Connected CPU: 1023 (0x3FF) • No. 1: 992 (0x3E0) • No. 2: 993 (0x3E1) • No. 3: 994 (0x3E2) • No. 4: 995 (0x3E3) ■For redundant CPU • Control system: 976 (0x3D0) • No specification: 1023 (0x3FF)	Fixed to 0 (0x00)
ActDestinationPortNumber	0 (0x00)	GOT port number			
ActDidPropertyBit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Fixed to 1 (0x01)
ActDsidPropertyBit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Fixed to 1 (0x01)
ActHostAddress	1.1.1.1	Host name or IP address of GOT			
ActIntelligentPreferenceBit	0 (0x00)	Fixed to 0 (0x00)	Target station QCPU (Q mode), LCPU: 1 (0x01) Other than the above: 0 (0x00)	Target station QCPU (Q mode), QCCPU, LCPU: 1 (0x01) Other than the above: 0 (0x00)	Fixed to 0 (0x00)
ActIONumber ^{*1}	1023 (0x3FF)	Fixed to 1023 (0x3FF)	Connected station side relayed module I/O address	Connected station side relayed module I/O address	Target station side ■For single CPU • Fixed to 1023 (0x3FF) ■For multiple CPUs • Connected CPU: 1023 (0x3FF) • No. 1: 992 (0x3E0) • No. 2: 993 (0x3E1) • No. 3: 994 (0x3E2) • No. 4: 995 (0x3E3) ■For redundant CPU • Control system: 976 (0x3D0) • No specification: 1023 (0x3FF)
ActMultiDropChannelNumber ^{*2}	0 (0x00)	Fixed to 0 (0x00)	Multi-drop channel number	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActNetworkNumber	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station side module network number
ActProtocolType	0x04 (PROTOCOL_SERI AL)	PROTOCOL_TCIP (0x05)			
ActStationNumber	255 (0xFF)	Fixed to 255 (0xFF)	Fixed to 255 (0xFF)	Fixed to 255 (0xFF)	Target station side module station number
ActThroughNetworkType	0 (0x00)	• MELSECNET/10 is not included.: 0 (0x00) • MELSECNET/10 is included.: 1 (0x01)			
ActTimeOut	10000	Any value specified by user in ms units			
ActUnitNumber	0 (0x00)	Fixed to 0 (0x00)	Target station side module station number	Target station side module station number	Fixed to 0 (0x00)

Property	Default value	Property patterns			
		○(1)	○(2)	○(3)	○(4)
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_GOTETHER_LNCPU (0x57)			

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

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

0: Default channel of module

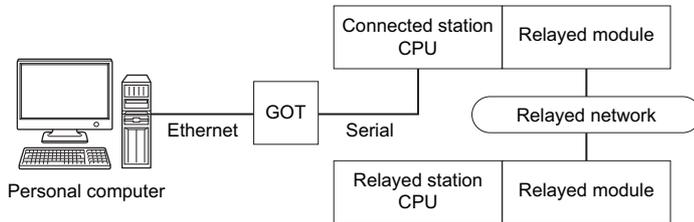
1: Channel 1

2: Channel 2

Personal computer side port: Ethernet board, GOT2000 port: Ethernet port, CPU side port: Serial (4)

When the connected station is an FXCPU

Configuration



Accessibility and property patterns

○(n): Accessible ('n' is the property pattern number), ×: Inaccessible

Connected station CPU	Relayed network	Relayed station CPU					
		FXCPU	RCPU	RCCPU	R motion CPU	LHCPU	FX5CPU
○(1)	CC IE TSN	×	×	×	×	×	×
	CC IE Control CC IE Field	×	×	×	×	×	×
	MELSECNET/H	×	×	×	×	×	×
	Ethernet	×	×	×	×	×	×
	Serial communication	×	×	×	×	×	×
	CC-Link	×	×	×	×	×	×

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

Property list

The following table shows the property settings of communication routes.

The setting is not necessary for those properties without description.

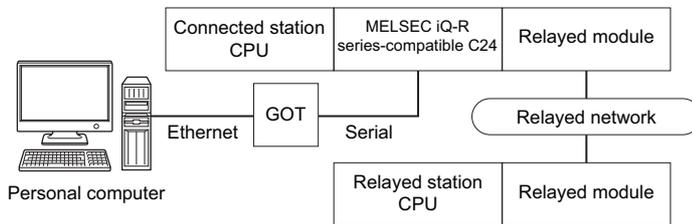
Property	Default value	Property patterns
		○(1)
ActCpuType	34 (CPU_Q02CPU)	CPU type corresponding to the target station
ActDestinationPortNumber*1	0 (0x00)	GOT port number
ActHostAddress	1.1.1.1	Host name or IP address of GOT
ActProtocolType	0x04 (PROTOCOL_SERIAL)	PROTOCOL_TCPIP
ActTimeOut	10000	Any value specified by user in ms units
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_GOTETHER_FXCPU

*1 For ActPortNumber, specify the value set to GOT side.

Personal computer side port: Ethernet board, GOT2000 side port: Ethernet port, CPU side port: Serial communication module (1)

When the connected station is MELSEC iQ-R series-compatible C24

Configuration



Accessibility and property patterns

○(n): Accessible ('n' is the property pattern number), ×: Inaccessible

Connected station CPU		Relayed network	Relayed station CPU				
RCP U	R motion CPU		RCPU	RCCPU	R motion CPU	LHCPU	FX5CPU
○(1)	○(1) ^{*1*3}	CC IE TSN	×	×	×	×	×
		CC IE Control CC IE Field	○(4) ^{*2}	○(4)	×	×	×
		MELSECNET/H	×	×	×	×	×
		Ethernet	×	×	×	×	×
		Serial communication	○(2)	×	×	×	×
		CC-Link	○(3)	○(3)	×	×	×

Connected station CPU		Relayed network	Relayed station CPU					
RCP U	R motion CPU		QCPU (Q mode)	QCCPU	LCPU	QSCPU	Q motion CPU	FXCPU
○(1)	○(1) ^{*1*3}	CC IE TSN	×	×	×	×	×	×
		CC IE Control CC IE Field	○(4)	×	○(4) ^{*2}	×	×	×
		MELSECNET/H	○(4)	×	×	×	×	×
		Ethernet	×	×	×	×	×	×
		Serial communication	○(2)	×	○(2)	×	×	×
		CC-Link	○(3)	×	○(3)	×	×	×

*1 Relayed station CPUs cannot be accessed via R motion CPU.

*2 Since CC-Link IE Controller Network is not supported, RCPU and LCPU relayed by CC-Link IE Controller Network cannot be accessed.

*3 Only CPU No.2 or later in a multiple CPU configuration can be accessed.

Property list

The following table shows the property settings of communication routes.

The setting is not necessary for those properties without description.

Property	Default value	Property patterns			
		○(1)	○(2)	○(3)	○(4)
ActCpuType	34 (CPU_Q02CPU)	CPU type corresponding to the target station			
ActDestinationIONumber ^{*1}	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station side ■For single CPU • Fixed to 1023 (0x3FF) ■For multiple CPUs • Connected CPU: 1023 (0x3FF) • No. 1: 992 (0x3E0) • No. 2: 993 (0x3E1) • No. 3: 994 (0x3E2) • No. 4: 995 (0x3E3)	Target station side ■For single CPU • Fixed to 1023 (0x3FF) ■For multiple CPUs • Connected CPU: 1023 (0x3FF) • No. 1: 992 (0x3E0) • No. 2: 993 (0x3E1) • No. 3: 994 (0x3E2) • No. 4: 995 (0x3E3)
ActDestinationPortNumber ^{*2}	0 (0x00)	GOT port number			
ActDidPropertyBit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActDsidPropertyBit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActHostAddress	1.1.1.1	Host name or IP address of GOT			
ActIntelligentPreferenceBit	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station RCP, QCPU (Q mode), LCP: 1 (0x01) Other than the above: 0 (0x00)	Target station RCP, QCPU (Q mode), QCCP, LCP: 1 (0x01) Other than the above: 0 (0x00)
ActIONumber ^{*1}	1023 (0x3FF)	■For single CPU • Fixed to 1023 (0x3FF) ■For multiple CPUs • Connected CPU: 1023 (0x3FF) • No. 1: 992 (0x3E0) • No. 2: 993 (0x3E1) • No. 3: 994 (0x3E2) • No. 4: 995 (0x3E3)	Target station side ■For single CPU • Fixed to 1023 (0x3FF) ■For multiple CPUs • Connected CPU: 1023 (0x3FF) • No. 1: 992 (0x3E0) • No. 2: 993 (0x3E1) • No. 3: 994 (0x3E2) • No. 4: 995 (0x3E3)	Connected station side module I/O address	Connected station side module I/O address
ActMultiDropChannelNumber ^{*3}	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Multi-drop channel number	Fixed to 0 (0x00)
ActNetworkNumber ^{*4}	0 (0x00)	Fixed to 0 (0x00)	Target station side module network number	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActProtocolType	0x04 (PROTOCOL_SERIAL)	PROTOCOL_TCPIP			
ActPacketType	0x01 (PACKET_PLC1)	PACKET_PLC1			
ActConnectUnitNumber	0 (0x00)	Fixed to 0 (0x00)			
ActPortNumber	1 (PORT_1)	Fixed to 0 (0x00)			
ActBaudRate	19200 (BAUDRATE_19200)	Fixed to 0 (0x00)			
ActCpuTimeOut	0 (0x00)	Fixed to 0 (0x00)			
ActControl	8 (TRC_DTR_OR_RTS)	Fixed to 0 (0x00)			
ActDataBits	8 (DATABIT_8)	Fixed to 0 (0x00)			
ActParity	1 (ODD_PARITY)	Fixed to 0 (0x00)			
ActStopBits	0 (STOPBIT_ONE)	Fixed to 0 (0x00)			
ActSumCheck	0 (NO_SUM_CHECK)	Fixed to 0 (0x00)			
ActSourceNetworkNumber	0 (0x00)	Fixed to 0 (0x00)			

Property	Default value	Property patterns			
		○(1)	○(2)	○(3)	○(4)
ActSourceStationNumber	0 (0x00)	Fixed to 0 (0x00)			
ActStationNumber ^{*4}	255 (0xFF)	Fixed to 255 (0xFF)	Target station side module station number	Fixed to 255 (0xFF)	Fixed to 255 (0xFF)
ActThroughNetworkType	0 (0x00)	<ul style="list-style-type: none"> • MELSECNET/10 is not included.: 0 (0x00) • MELSECNET/10 is included.: 1 (0x01) 			
ActTimeOut	10000	Any value specified by user in ms units			
ActUnitNumber	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station side module station number	Target station side module station number
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_GOTETHER_RJ71C24			

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

*2 Specify the value set to GOT side for ActDestinationPortNumber.

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

0: Default channel of module

1: Channel 1

2: Channel 2

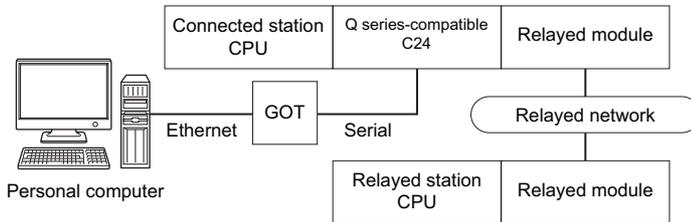
*4 When the property pattern is ○(2), specify the value set in the target station side parameter for ActNetworkNumber and ActStationNumber.

Personal computer side port: Ethernet board, GOT2000 side port: Ethernet port, CPU side port: Serial communication module (2)

When the connected station is Q series-compatible C24

Configuration

10



Accessibility and property patterns

○(n): Accessible ('n' is the property pattern number), ×: Inaccessible

Connected station CPU		Relayed network	Relayed station CPU				
QCPU (Q mode)	QCCPU		RCPU	RCCPU	R motion CPU	LHCPU	FX5CPU
○(1)*1	○(1)	CC IE TSN	×	×	×	×	×
		CC IE Control CC IE Field	×	×	×	×	×
		MELSECNET/H	×	×	×	×	×
		Ethernet	×	×	×	×	×
		Serial communication	×	×	×	×	×
		CC-Link	×	×	×	×	×

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

*1 Redundant CPU cannot be accessed.

*2 Since CC-Link IE Field Network is not supported, Q12DCCPU-V (Basic mode) relayed by CC-Link IE Field Network cannot be accessed.

*3 Since CC-Link IE Controller Network is not supported, LCPU relayed by CC-Link IE Controller Network cannot be accessed.

Property list

The following table shows the property settings of communication routes.

The setting is not necessary for those properties without description.

Property	Default value	Property patterns			
		○(1)	○(2)	○(3)	○(4)
ActCpuType	34 (CPU_Q02CPU)	CPU type corresponding to the target station			
ActDestinationIONumber	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station side ■For single CPU • Fixed to 1023 (0x3FF) ■For multiple CPUs • Connected CPU: 1023 (0x3FF) • No. 1: 992 (0x3E0) • No. 2: 993 (0x3E1) • No. 3: 994 (0x3E2) • No. 4: 995 (0x3E3)	Target station side ■For single CPU • Fixed to 1023 (0x3FF) ■For multiple CPUs • Connected CPU: 1023 (0x3FF) • No. 1: 992 (0x3E0) • No. 2: 993 (0x3E1) • No. 3: 994 (0x3E2) • No. 4: 995 (0x3E3) ■For redundant CPU • Control system: 976 (0x3D0) • No specification: 1023 (0x3FF)
ActDestinationPortNumber	0 (0x00)	GOT port number			
ActDidPropertyBit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActDsidPropertyBit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActHostAddress	1.1.1.1	Host name or IP address of GOT			
ActIntelligentPreferenceBit	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station QCPU (Q mode), LCPU: 1 (0x01) Other than the above: 0 (0x00)	Target station QCPU (Q mode), QCCPU, LCPU: 1 (0x01) Other than the above: 0 (0x00)
ActIONumber ^{*1}	1023 (0x3FF)	■For single CPU • Fixed to 1023 (0x3FF) ■For multiple CPUs • Connected CPU: 1023 (0x3FF) • No. 1: 992 (0x3E0) • No. 2: 993 (0x3E1) • No. 3: 994 (0x3E2) • No. 4: 995 (0x3E3)	Target station side ■For single CPU • Fixed to 1023 (0x3FF) ■For multiple CPUs • Connected CPU: 1023 (0x3FF) • No. 1: 992 (0x3E0) • No. 2: 993 (0x3E1) • No. 3: 994 (0x3E2) • No. 4: 995 (0x3E3) ■For redundant CPU • Control system: 976 (0x3D0) • No specification: 1023 (0x3FF)	Connected station side module I/O address	Connected station side module I/O address
ActMultiDropChannelNumber ^{*2}	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Multi-drop channel number	Fixed to 0 (0x00)
ActNetworkNumber	0 (0x00)	Fixed to 0 (0x00)	Target station side module network number	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActProtocolType	0x04 (PROTOCOL_SERIAL)	PROTOCOL_TCPIP (0x05)			
ActStationNumber	255 (0xFF)	Fixed to 255 (0xFF)	Target station side module station number	Fixed to 255 (0xFF)	Fixed to 255 (0xFF)
ActThroughNetworkType	0 (0x00)	• MELSECNET/10 is not included.: 0 (0x00) • MELSECNET/10 is included.: 1 (0x01)			
ActTimeOut	10000	Any value specified by user in ms units			
ActUnitNumber	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station side module station number	Target station side module station number

Property	Default value	Property patterns			
		<input type="radio"/> (1)	<input type="radio"/> (2)	<input type="radio"/> (3)	<input type="radio"/> (4)
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_GOTETHER_QNCPU (0x56)			

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

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

0: Default channel of module

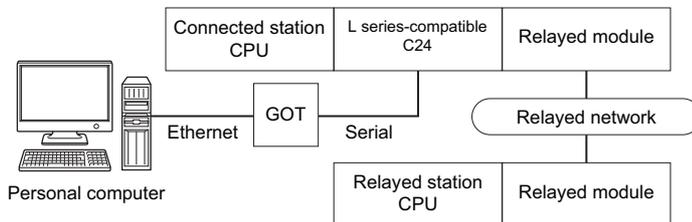
1: Channel 1

2: Channel 2

Personal computer side port: Ethernet board, GOT2000 side port: Ethernet port, CPU side port: Serial communication module (3)

When the connected station is L series-compatible C24

Configuration



Accessibility and property patterns

○(n): Accessible ('n' is the property pattern number), ×: Inaccessible

Connected station CPU LCPU	Relayed network	Relayed station CPU				
		RCPU	RCCPU	R motion CPU	LHCPU	FX5CPU
○(1)	CC IE TSN	×	×	×	×	×
	CC IE Field* ²	×	×	×	×	×
	MELSECNET/H	×	×	×	×	×
	Ethernet	×	×	×	×	×
	Serial communication	×	×	×	×	×
	CC-Link	×	×	×	×	×

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

*1 Redundant CPU cannot be accessed.

*2 Since CC-Link IE Controller Network is not supported, LCPU relayed by CC-Link IE Controller Network cannot be accessed.

*3 Since CC-Link IE Field Network is not supported, Q12DCCPU-V (Basic mode) relayed by CC-Link IE Field Network cannot be accessed.

Property list

The following table shows the property settings of communication routes.

The setting is not necessary for those properties without description.

Property	Default value	Property patterns			
		○(1)	○(2)	○(3)	○(4)
ActCpuType	34 (CPU_Q02CPU)	CPU type corresponding to the target station			
ActDestinationIONumber	0 (0x00)	Fixed to 0 (0x00)	Target station side ■For single CPU • Fixed to 1023 (0x3FF) ■For multiple CPUs • Connected CPU: 1023 (0x3FF) • No. 1: 992 (0x3E0) • No. 2: 993 (0x3E1) • No. 3: 994 (0x3E2) • No. 4: 995 (0x3E3)	Target station side ■For single CPU • Fixed to 1023 (0x3FF) ■For multiple CPUs • Connected CPU: 1023 (0x3FF) • No. 1: 992 (0x3E0) • No. 2: 993 (0x3E1) • No. 3: 994 (0x3E2) • No. 4: 995 (0x3E3) ■For redundant CPU • Control system: 976 (0x3D0) • No specification: 1023 (0x3FF)	Fixed to 0 (0x00)
ActDestinationPortNumber	0 (0x00)	GOT port number			
ActDidPropertyBit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Fixed to 1 (0x01)
ActDsidPropertyBit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Fixed to 1 (0x01)
ActHostAddress	1.1.1.1	Host name or IP address of GOT			
ActIntelligentPreferenceBit	0 (0x00)	Fixed to 0 (0x00)	Target station QCPU (Q mode), LCPU: 1 (0x01) Other than the above: 0 (0x00)	Target station QCPU (Q mode), QCCPU, LCPU: 1 (0x01) Other than the above: 0 (0x00)	Fixed to 0 (0x00)
ActIONumber ^{*1}	1023 (0x3FF)	Fixed to 1023 (0x3FF)	Connected station side relayed module I/O address	Connected station side relayed module I/O address	Target station side ■For single CPU • Fixed to 1023 (0x3FF) ■For multiple CPUs • Connected CPU: 1023 (0x3FF) • No. 1: 992 (0x3E0) • No. 2: 993 (0x3E1) • No. 3: 994 (0x3E2) • No. 4: 995 (0x3E3) ■For redundant CPU • Control system: 976 (0x3D0) • No specification: 1023 (0x3FF)
ActMultiDropChannelNumber ^{*2}	0 (0x00)	Fixed to 0 (0x00)	Multi-drop channel number	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActNetworkNumber	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station side module network number
ActProtocolType	0x04 (PROTOCOL_SERIAL)	PROTOCOL_TCPIP (0x05)			
ActStationNumber	255 (0xFF)	Fixed to 255 (0xFF)	Fixed to 255 (0xFF)	Fixed to 255 (0xFF)	Target station side module station number
ActThroughNetworkType	0 (0x00)	• MELSECNET/10 is not included.: 0 (0x00) • MELSECNET/10 is included.: 1 (0x01)			
ActTimeOut	10000	Any value specified by user in ms units			
ActUnitNumber	0 (0x00)	Fixed to 0 (0x00)	Target station side module station number	Target station side module station number	Fixed to 0 (0x00)

Property	Default value	Property patterns			
		<input type="radio"/> (1)	<input type="radio"/> (2)	<input type="radio"/> (3)	<input type="radio"/> (4)
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_GOTETHER_LNCPU (0x57)			

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

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

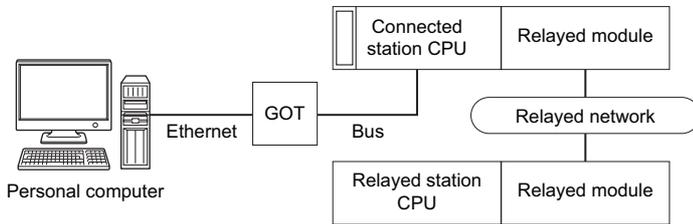
0: Default channel of module

1: Channel 1

2: Channel 2

Personal computer side port: Ethernet board, GOT2000 side port: Ethernet port, CPU side port: Bus connection

Configuration



Accessibility and property patterns

○(n): Accessible ('n' is the property pattern number), ×: Inaccessible

Connected station CPU QCPU (Q mode)	Relayed network	Relayed station CPU				
		RCPU	RCCPU	R motion CPU	LHCPU	FX5CPU
○(1) ^{*1}	CC IE TSN	×	×	×	×	×
	CC IE Control CC IE Field	×	×	×	×	×
	MELSECNET/H	×	×	×	×	×
	Ethernet	×	×	×	×	×
	Serial communication	×	×	×	×	×
	CC-Link	×	×	×	×	×

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

Connected station CPU QCCPU	Relayed network	Relayed station CPU				
		RCPU	RCCPU	R motion CPU	LHCPU	FX5CPU
○(1)	CC IE TSN	×	×	×	×	×
	CC IE Control CC IE Field	×	×	×	×	×
	MELSECNET/H	×	×	×	×	×
	Ethernet	×	×	×	×	×
	Serial communication	×	×	×	×	×
	CC-Link	×	×	×	×	×

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

*1 Redundant CPU cannot be accessed.

*2 Since CC-Link IE Field Network is not supported, Q12DCCPU-V (Basic mode) relayed by CC-Link IE Field Network cannot be accessed.

*3 Since CC-Link IE Controller Network is not supported, LCPU relayed by CC-Link IE Controller Network cannot be accessed.

Property list

The following table shows the property settings of communication routes.

The setting is not necessary for those properties without description.

Property	Default value	Property patterns			
		○(1)	○(2)	○(3)	○(4)
ActCpuType	34 (CPU_Q02CPU)	CPU type corresponding to the target station			
ActDestinationIONumber	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station side ■For single CPU Fixed to 1023 (0x3FF) ■For multiple CPUs • Connected CPU: 1023 (0x3FF) • No. 1: 992 (0x3E0) • No. 2: 993 (0x3E1) • No. 3: 994 (0x3E2) • No. 4: 995 (0x3E3)	Target station side ■For single CPU • Fixed to 1023 (0x3FF) ■For multiple CPUs • Connected CPU: 1023 (0x3FF) • No. 1: 992 (0x3E0) • No. 2: 993 (0x3E1) • No. 3: 994 (0x3E2) • No. 4: 995 (0x3E3)
ActDestinationPortNumber	0 (0x00)	GOT port number			
ActDidPropertyBit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActDsidPropertyBit	1 (0x01)	Fixed to 1 (0x01)	Fixed to 1 (0x01)	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActHostAddress	1.1.1.1	Host name or IP address of GOT			
ActIntelligentPreferenceBit	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station QCPU (Q mode), LCPU: 1 (0x01) Other than the above: 0 (0x00)	Target station QCPU (Q mode), QCCPU, LCPU: 1 (0x01) Other than the above: 0 (0x00)
ActIONumber* ¹	1023 (0x3FF)	■For single CPU • Fixed to 1023 (0x3FF) ■For multiple CPUs • Connected CPU: 1023 (0x3FF) • No. 1: 992 (0x3E0) • No. 2: 993 (0x3E1) • No. 3: 994 (0x3E2) • No. 4: 995 (0x3E3)	Target station side ■For single CPU • Fixed to 1023 (0x3FF) ■For multiple CPUs • Connected CPU: 1023 (0x3FF) • No. 1: 992 (0x3E0) • No. 2: 993 (0x3E1) • No. 3: 994 (0x3E2) • No. 4: 995 (0x3E3)	Connected station side module I/O address	Connected station side module I/O address
ActMultiDropChannelNumber* ²	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Multi-drop channel number	Fixed to 0 (0x00)
ActNetworkNumber	0 (0x00)	Fixed to 0 (0x00)	Target station side module network number	Fixed to 0 (0x00)	Fixed to 0 (0x00)
ActProtocolType	0x04 (PROTOCOL_SERIAL)	PROTOCOL_TCPIP (0x05)			
ActStationNumber	255 (0xFF)	Fixed to 255 (0xFF)	Target station side module station number	Fixed to 255 (0xFF)	Fixed to 255 (0xFF)
ActThroughNetworkType	0 (0x00)	• MELSECNET/10 is not included.: 0 (0x00) • MELSECNET/10 is included.: 1 (0x01)			
ActTimeOut	10000	Any value specified by user in ms units			
ActUnitNumber	0 (0x00)	Fixed to 0 (0x00)	Fixed to 0 (0x00)	Target station side module station number	Target station side module station number
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_GOTETHER_QBUS (0x58)			

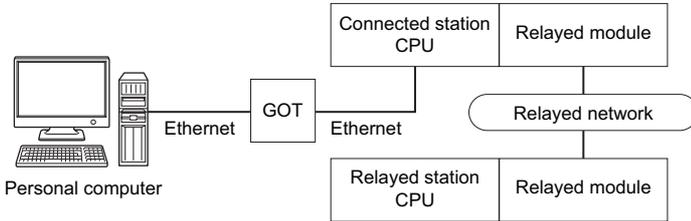
*1 For the I/O address, specify the value of the actual start I/O number divided by 16.

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

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

Personal computer side port: Ethernet board, GOT2000 port: Ethernet port, CPU side port: Ethernet port

Configuration



Accessibility and property patterns

○(n): Accessible ('n' is the property pattern number), ×: Inaccessible

Connected station CPU	Relayed network	Relayed station CPU				
QCPU (Q mode)		RCPU	RCCPU	R motion CPU	LHCPU	FX5CPU
○(1) ^{*1}	CC IE TSN	×	×	×	×	×
	CC IE Control CC IE Field	×	×	×	×	×
	MELSECNET/H	×	×	×	×	×
	Ethernet	×	×	×	×	×
	Serial communication	×	×	×	×	×
	CC-Link	×	×	×	×	×

Connected station CPU	Relayed network	Relayed station CPU					
QCPU (Q mode)		QCPU (Q mode)	QCCPU	LCPU	QSCPU	Q motion CPU	FXCPU
○(1) ^{*1}	CC IE TSN	×	×	×	×	×	×
	CC IE Control CC IE Field	×	×	×	×	×	×
	MELSECNET/H	×	×	×	×	×	×
	Ethernet	×	×	×	×	×	×
	Serial communication	×	×	×	×	×	×
	CC-Link	×	×	×	×	×	×

*1 Only operates for Qn(U)(J)(P)(D)(E)(V)(H). QnPRHCPU cannot be accessed because it is not supported.

Property list

The following table shows the property settings of communication routes.

The setting is not necessary for those properties without description.

Property	Default value	Property patterns
		○(1)
ActBaudRate	19200 (BAUDRATE_19200)	Fixed to 0 (0x00)
ActConnectUnitNumber	0 (0x00)	Fixed to 0 (0x00)
ActControl	8 (TRC_DTR_OR_RTS)	Fixed to 0 (0x00)
ActCpuTimeOut	0 (0x00)	Fixed to 0 (0x00)
ActCpuType	34 (CPU_Q02CPU)	CPU type corresponding to the target station
ActDataBits	8 (DATABIT_8)	Fixed to 0 (0x00)
ActDestinationIONumber	0 (0x00)	Fixed to 0 (0x00)
ActDestinationPortNumber*1	0 (0x00)	GOT port number
ActDidPropertyBit	1 (0x01)	Fixed to 1 (0x01)
ActDsidPropertyBit	1 (0x01)	Fixed to 1 (0x01)
ActHostAddress	1.1.1.1	Host name or IP address of GOT
ActIntelligentPreferenceBit	0 (0x00)	Fixed to 0 (0x00)
ActIONumber*2	1023 (0x3FF)	<ul style="list-style-type: none"> ■ For single CPU <ul style="list-style-type: none"> • Fixed to 1023 (0x3FF) ■ For multiple CPUs <ul style="list-style-type: none"> • Connected CPU: 1023 (0x3FF) • No. 1: 992 (0x3E0) • No. 2: 993 (0x3E1) • No. 3: 994 (0x3E2) • No. 4: 995 (0x3E3)
ActMultiDropChannelNumber*3	0 (0x00)	Fixed to 0 (0x00)
ActNetworkNumber*4	0 (0x00)	Fixed to 0 (0x00)
ActPacketType	0x01 (PACKET_PL1)	PACKET_PL1
ActParity	1 (ODD_PARITY)	Fixed to 0 (0x00)
ActPortNumber	1 (PORT_1)	Fixed to 0 (0x00)
ActProtocolType	0x04 (PROTOCOL_SERIAL)	PROTOCOL_TCPIP (0x05)/PROTOCOL_UDPIP (0x08)
ActStopBits	0 (STOPBIT_ONE)	Fixed to 0 (0x00)
ActSumCheck	0 (NO_SUM_CHECK)	Fixed to 0 (0x00)
ActSourceNetworkNumber*5	0 (0x00)	Fixed to 0 (0x00)
ActSourceStationNumber*6	0 (0x00)	Fixed to 0 (0x00)
ActStationNumber*4	255 (0xFF)	Fixed to 255 (0xFF)
ActThroughNetworkType	0 (0x00)	Fixed to 1 (0x01)
ActTimeOut	10000	Any value specified by user in ms units
ActUnitNumber	0 (0x00)	Fixed to 0 (0x00)
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_GOTETHER_QN_ETHER (0x6F)

*1 Specify the value set on the GOT side.

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

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

0: Default channel of module

1: Channel 1

2: Channel 2

*4 Specify the value set in the parameter setting on the target station side.

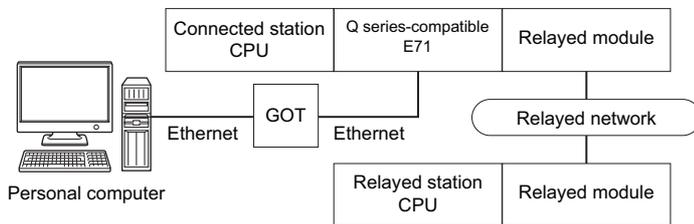
*5 When the connected station includes an Ethernet port and the extension setting (network No. station number setting), specify the same network No. as that set for the Ethernet port.

When the connected station does not include an Ethernet port and extension setting (network No. station number setting), specify the same network No. as that of the connected station set in the Ethernet setting of a GOT. For the setting addressing to the own station (1NetworkNumber: 0x00, 1StationNumber: 0xFF), set '0x00' same as '1NetworkNumber'.

*6 Specify the personal computer side station number to avoid the same station number set for QJ71E71 in the same Ethernet network.

Personal computer side port: Ethernet board, GOT2000 port: Ethernet port, CPU side port: Q series-compatible E71

Configuration



Accessibility and property patterns

○(n): Accessible ('n' is the property pattern number), ×: Inaccessible

Connected station CPU	Relayed network	Relayed station CPU				
QCPU (Q mode)		RCPU	RCCPU	R motion CPU	LHCPU	FX5CPU
○(1) ^{*1}	CC IE TSN	×	×	×	×	×
	CC IE Control CC IE Field	×	×	×	×	×
	MELSECNET/H	×	×	×	×	×
	Ethernet	×	×	×	×	×
	Serial communication	×	×	×	×	×
	CC-Link	×	×	×	×	×

Connected station CPU	Relayed network	Relayed station CPU					
QCPU (Q mode)		QCPU (Q mode)	QCCPU	LCPU	QSCPU	Q motion CPU	FXCPU
○(1) ^{*1}	CC IE TSN	×	×	×	×	×	×
	CC IE Control CC IE Field	×	×	×	×	×	×
	MELSECNET/H	×	×	×	×	×	×
	Ethernet	×	×	×	×	×	×
	Serial communication	×	×	×	×	×	×
	CC-Link	×	×	×	×	×	×

*1 Only operates for Qn(U)(J)(P)(D)(E)(V)(H). QnPRHCPU cannot be accessed because it is not supported.

Property list

The following table shows the property settings of communication routes.

The setting is not necessary for those properties without description.

Property	Default value	Property patterns
		○(1)
ActBaudRate	19200 (BAUDRATE_19200)	Fixed to 0 (0x00)
ActConnectUnitNumber	0 (0x00)	Fixed to 0 (0x00)
ActControl	8 (TRC_DTR_OR_RTS)	Fixed to 0 (0x00)
ActCpuTimeOut	0 (0x00)	Fixed to 0 (0x00)
ActCpuType	34 (CPU_Q02CPU)	CPU type corresponding to the target station
ActDataBits	8 (DATABIT_8)	Fixed to 0 (0x00)
ActDestinationIONumber	0 (0x00)	Fixed to 0 (0x00)
ActDestinationPortNumber*1	0 (0x00)	GOT port number
ActDidPropertyBit	1 (0x01)	Fixed to 1 (0x01)
ActDsidPropertyBit	1 (0x01)	Fixed to 1 (0x01)
ActHostAddress	1.1.1.1	Host name or IP address of GOT
ActIntelligentPreferenceBit	0 (0x00)	Fixed to 0 (0x00)
ActIONumber*2	1023 (0x3FF)	<ul style="list-style-type: none"> ■ For single CPU <ul style="list-style-type: none"> • Fixed to 1023 (0x3FF) ■ For multiple CPUs <ul style="list-style-type: none"> • Connected CPU: 1023 (0x3FF) • No. 1: 992 (0x3E0) • No. 2: 993 (0x3E1) • No. 3: 994 (0x3E2) • No. 4: 995 (0x3E3)
ActMultiDropChannelNumber*3	0 (0x00)	Fixed to 0 (0x00)
ActNetworkNumber*4	0 (0x00)	Target station side module network number
ActPacketType	0x01 (PACKET_PLC1)	PACKET_PLC1
ActParity	1 (ODD_PARITY)	Fixed to 0 (0x00)
ActPortNumber	1 (PORT_1)	Fixed to 0 (0x00)
ActProtocolType	0x04 (PROTOCOL_SERIAL)	PROTOCOL_TCPIP (0x05)/PROTOCOL_UDPIP (0x08)
ActStopBits	0 (STOPBIT_ONE)	Fixed to 0 (0x00)
ActSumCheck	0 (NO_SUM_CHECK)	Fixed to 0 (0x00)
ActSourceNetworkNumber*5	0 (0x00)	GOT side network number
ActSourceStationNumber*6	0 (0x00)	GOT side station number
ActStationNumber*4	255 (0xFF)	Target station side module station number
ActThroughNetworkType	0 (0x00)	Fixed to 1 (0x01)
ActTimeOut	10000	Any value specified by user in ms units
ActUnitNumber	0 (0x00)	Fixed to 0 (0x00)
ActUnitType	0x13 (UNIT_QNCPU)	UNIT_GOTETHER_QN_ETHER (0x6F)

*1 Specify the value set on the GOT side.

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

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

0: Default channel of module

1: Channel 1

2: Channel 2

*4 Specify the value set in the parameter setting on the target station side.

*5 Specify the same network No. as the one set for QJ71E71 in the parameter setting of the QJ71E71.

*6 Specify the personal computer side station number to avoid the same station number set for QJ71E71 in the same Ethernet network.

11 FUNCTIONS

This chapter explains the programming considerations, function list, and details of functions.

11.1 Programming Considerations

This section explains the considerations for creating a program using the function.

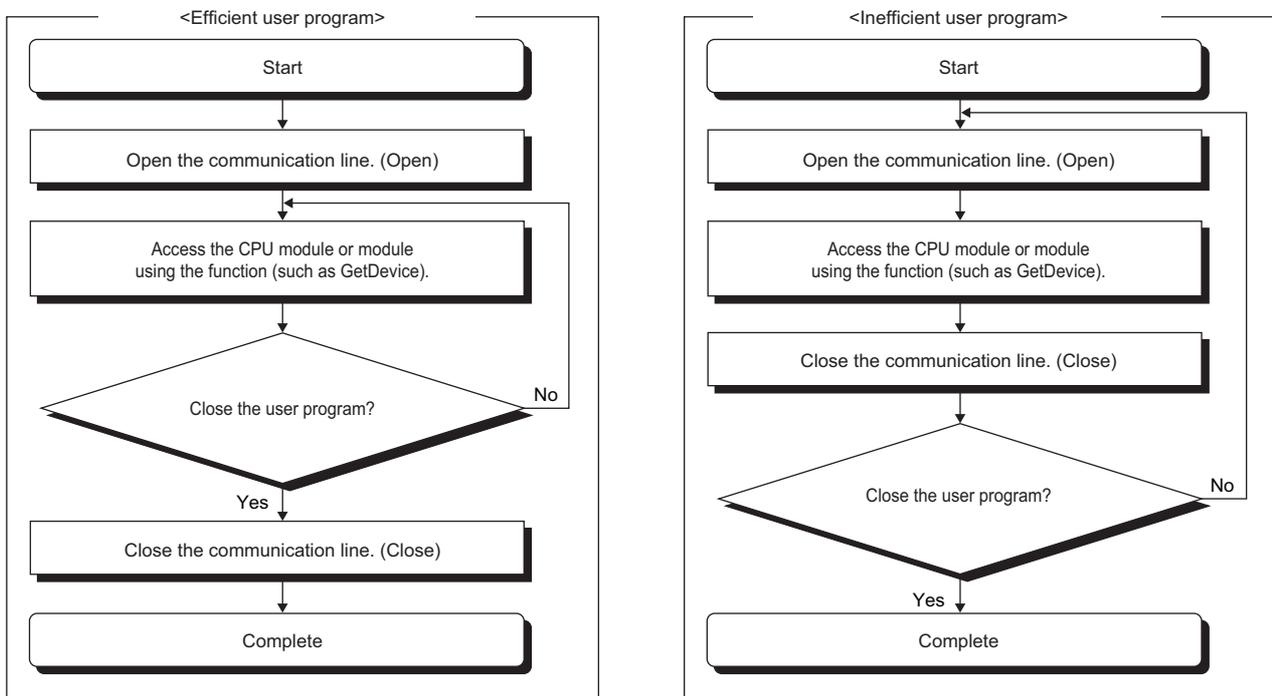
Read these precautions before starting creating a program.

Open function (opening communication line)

The processing may take long time due to the Open function processes: establishment of communication routes, obtaining programmable controller internal information, or the like.

Efficient programs need to be created in order to improve the speed performance of user programs.

The following figures are flows for creating efficient and inefficient user programs.



Differences between LONG type functions and SHORT type/INT type functions

LONG type functions and SHORT type/INT type functions are the functions of MX Component to 'read devices in batch,' 'write devices in batch,' 'read devices randomly,' 'write devices randomly,' 'set device data,' and 'acquire device data.'

The following describes the differences between the LONG type functions and SHORT type/INT type functions.

■LONG type function

When writing/reading a negative device value using the LONG type function, the LONG type data needs to be converted to the SHORT type/INT type data in the user program.

When writing/reading negative device values, use the SHORT type/INT type functions.

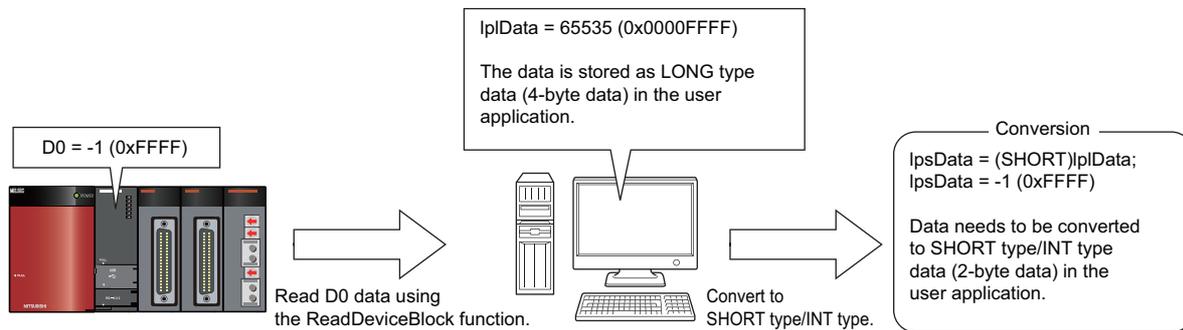
☞ Page 417 SHORT type/INT type function

Target function	Reference
ReadDeviceBlock	Page 425 ReadDeviceBlock (Reading devices in batch)
WriteDeviceBlock	Page 428 WriteDeviceBlock (Writing devices in batch)
ReadDeviceRandom	Page 432 ReadDeviceRandom (Reading devices randomly)
WriteDeviceRandom	Page 432 ReadDeviceRandom (Reading devices randomly)
SetDevice	Page 439 SetDevice (Setting device data)

Target function	Reference
GetDevice	Page 441 GetDevice (Acquiring device data)

Ex.

When reading device data with a negative value using the ReadDeviceBlock function



SHORT type/INT type function

When reading/writing a negative device value using the SHORT type/INT type function, the device value can be used as it is without converting it.

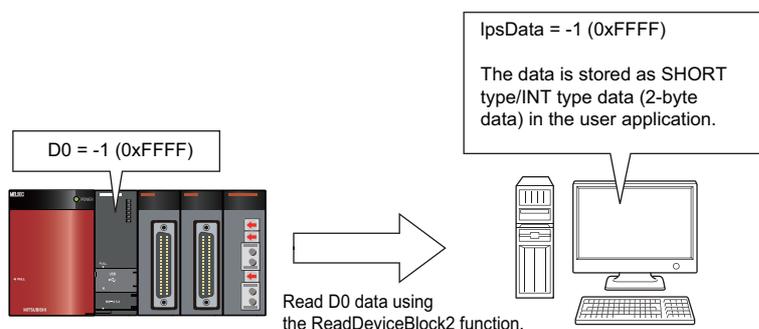
When reading/writing double word devices, use any of the LONG type functions.

☞ Page 416 LONG type function

Target function	Reference
ReadDeviceBlock2	Page 474 ReadDeviceBlock2 (Reading devices in batch)
WriteDeviceBlock2	Page 477 WriteDeviceBlock2 (Writing devices in batch)
ReadDeviceRandom2	Page 480 ReadDeviceRandom2 (Reading devices randomly)
WriteDeviceRandom2	Page 483 WriteDeviceRandom2 (Writing devices randomly)
SetDevice2	Page 487 SetDevice2 (Setting device data)
GetDevice2	Page 490 GetDevice2 (Acquiring device data)

Ex.

When reading device data with a negative value using the ReadDeviceBlock2 function



Considerations of multi-thread programming

When performing multi-thread programming, follow the rules of COM and ActiveX controls.

For details, refer to the rules and reference books of COM and ActiveX controls.

Point

- The ActiveX controls used on MX Component are those of the STA model.
- When passing the interface pointer to another apartment, the interface pointer needs to be marshaled. Synchronize programs using CoMarshalInterThreadInterfaceInStream or CoGetInterfaceAndReleaseStream of the COM function.

Considerations for using RnSF CPU

In case of safety mode, the following functions cannot be used for the RnSF CPU safety device.

Function name	Feature
WriteDeviceBlock	To write devices in batch.
WriteDeviceRandom	To write devices randomly.
SetDevice	To set device data.
WriteDeviceBlock2	To write devices in batch.
WriteDeviceRandom2	To write devices randomly.
SetDevice2	To set device data.
WriteBuffer	To write data to buffer memory.

Point

If any of these functions is used, the error code "0x010A42A5" (Operation that can not be carried out in safety mode) will be returned.

Considerations for using QSCPU

The following functions cannot be used for QSCPU.

Function name	Feature
WriteDeviceBlock	To write devices in batch.
WriteDeviceRandom	To write devices randomly.
SetDevice	To set device data.
WriteBuffer	To write data to buffer memory.
SetClockData	To write clock data.
SetCpuStatus	To perform a remote operation.
WriteDeviceBlock2	To write devices in batch.
WriteDeviceRandom2	To write devices randomly.
SetDevice2	To set device data.

Point

If any of these functions is used, the error code "0x010A42A0" (mismatched CPU access password) will be returned.

Consideration for using inverter/robot controller

The functions of .Net control and controls set in a program ( Page 40 Grasping a Control to be Used) cannot be used during performing inverter communication/robot controller communication.

Point

If any of these functions is used, the error code "0xF0000007" (function non-support error) will be returned.

Forced termination of processes during communication

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

Error at communication start

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

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

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

Considerations for using Ethernet modules

- Provide an interval longer than the sequence scan time of the Ethernet module mounted station for a period from when the Open method is executed until the Close method is executed.
- Provide an interval of at least 500 ms for a period from when the Close method is executed until the Open method is executed again.

Control

■CheckDeviceString

Do not use the CheckDeviceString method of Act control.

■ActUMsg control and ActUWzd control

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

11.2 Function List

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

Function name	Feature	Refer ^{*1}
Open	To open a communication line.	Act: Page 422 Open (Opening communication line) DataLogging: Page 494 Open (Opening communication line) Dot: Page 504 Open (Opening communication line)
Close	To close a communication line.	Act: Page 424 Close (Closing communication line) DataLogging: Page 496 Close (Closing communication line) Dot: Page 505 Close (Closing communication line)
ReadDeviceBlock	To read devices in batch (in 4-byte data unit).	Act: Page 425 ReadDeviceBlock (Reading devices in batch) Dot: Page 506 ReadDeviceBlock (Reading devices in batch)
WriteDeviceBlock	To write devices in batch (in 4-byte data unit).	Act: Page 428 WriteDeviceBlock (Writing devices in batch) Dot: Page 512 WriteDeviceBlock (Writing devices in batch)
ReadDeviceRandom	To read devices randomly (in 4-byte data unit).	Act: Page 432 ReadDeviceRandom (Reading devices randomly) Dot: Page 518 ReadDeviceRandom (Reading devices randomly)
WriteDeviceRandom	To write devices randomly (in 4-byte data unit).	Act: Page 436 WriteDeviceRandom (Writing devices randomly) Dot: Page 529 WriteDeviceRandom (Writing devices randomly)
SetDevice	To set one point of device (in 4-byte data unit).	Act: Page 439 SetDevice (Setting device data) Dot: Page 540 SetDevice (Setting device data)
GetDevice	To acquire data of one point of device (in 4-byte data unit).	Act: Page 441 GetDevice (Acquiring device data) Dot: Page 544 GetDevice (Acquiring device data)
ReadBuffer	To read data from buffer memory.	Act: Page 444 ReadBuffer (Reading data from buffer memory) Dot: Page 548 ReadBuffer (Reading data from buffer memory)
WriteBuffer	To write data to buffer memory.	Act: Page 448 WriteBuffer (Writing data to buffer memory) Dot: Page 550 WriteBuffer (Writing data to buffer memory)
GetClockData	To read clock data of a CPU module.	Act: Page 452 GetClockData (Reading clock data) Dot: Page 552 GetClockData (Reading clock data)
SetClockData	To write clock data of a CPU module.	Act: Page 456 SetClockData (Writing clock data) Dot: Page 554 SetClockData (Writing clock data)
GetCpuType	To read CPU module model.	Act: Page 460 GetCpuType (Reading CPU module model) Dot: Page 556 GetCpuType (Reading CPU module model)
SetCpuStatus	To perform remote RUN/STOP/PAUSE/RESET of CPU module.	Act: Page 462 SetCpuStatus (Remote control) Dot: Page 557 SetCpuStatus (Remote control)
EntryDeviceStatus ^{*2}	To register devices for status monitoring.	Act: Page 465 EntryDeviceStatus (Registering devices for status monitoring) Dot: Page 558 EntryDeviceStatus (Registering devices for status monitoring)
FreeDeviceStatus ^{*2}	To deregister devices for status monitoring.	Act: Page 470 FreeDeviceStatus (Deregistering devices for status monitoring) Dot: Page 562 FreeDeviceStatus (Deregistering devices for status monitoring)
OnDeviceStatus ^{*2}	To notify event.	Act: Page 471 OnDeviceStatus (Event notification) Dot: Page 563 OnDeviceStatus (Event notification)
ReadDeviceBlock2	To read devices in batch (in 2-byte data unit).	Act: Page 474 ReadDeviceBlock2 (Reading devices in batch) Dot: Page 565 ReadDeviceBlock2 (Reading devices in batch)
WriteDeviceBlock2	To write devices in batch (in 2-byte data unit).	Act: Page 477 WriteDeviceBlock2 (Writing devices in batch) Dot: Page 570 WriteDeviceBlock2 (Writing devices in batch)
ReadDeviceRandom2	To read devices randomly (in 2-byte data unit).	Act: Page 480 ReadDeviceRandom2 (Reading devices randomly) Dot: Page 575 ReadDeviceRandom2 (Reading devices randomly)
WriteDeviceRandom2	To write devices randomly (in 2-byte data unit).	Act: Page 483 WriteDeviceRandom2 (Writing devices randomly) Dot: Page 584 WriteDeviceRandom2 (Writing devices randomly)
SetDevice2	To set one point of device (in 2-byte data unit).	Act: Page 487 SetDevice2 (Setting device data) Dot: Page 595 SetDevice2 (Setting device data)
GetDevice2	To acquire data of one point of device (in 2-byte data unit).	Act: Page 490 GetDevice2 (Acquiring device data) Dot: Page 599 GetDevice2 (Acquiring device data)
GetErrorMessage	To acquire error description and corrective action.	Msg: Page 492 GetErrorMessage (Acquiring error description and corrective action)
ReadFirstFile	To search for a directly name/file name.	DataLogging: Page 497 ReadFirstFile (Searching for a file/directory)
ReadNextFile	To search for a directly name/file name.	DataLogging: Page 499 ReadNextFile (Searching for a file/directory)
ReadClose	To end the search.	DataLogging: Page 501 ReadClose (Ending the search)
GetFile	To transfer a logging file.	DataLogging: Page 502 GetFile (Transferring logging files)
Dispose	To release memory.	Dot: Page 602 Dispose (release memory)

- *1 Act: ActUtilType, ActUtilType64, ActProgType, ActProgType64,
DataLogging: ActDataLogging, ActDataLogging64, ActProgDataLogging, ActProgDataLogging64,
Dot: DotUtilType, DotUtilType64,
Msg: ActSupportMsg, ActSupportMsg64
- *2 Can be used only by pasting a control to a form for VC++ and VBA. Cannot be used by creating an object.
Controls for 64-bit cannot be used in any development languages.
For the differences of each control, refer to the following:
 Page 42 ADDING CONTROLS TO USER PROGRAM

Point 

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

11.3 Details of Functions (For Act Control)

Open (Opening communication line)

Applicable controls

This function can be used for the following controls.

- 32-bit: ActProgType, ActUtilType
- 64-bit: ActProgType64, ActUtilType64

Feature

To open a communication line.

Format

■VBA

IRet = object.Open()

Data type	Argument	Description	Input/Output
Long	IRet	Returned value	Output

■Visual C++

IRet = object.Open()

Data type	Argument	Description	Input/Output
Long	IRet	Returned value	Output

■Visual Basic .NET

IRet = object.Open()

Data type	Argument	Description	Input/Output
Integer	IRet	Returned value	Output

■Visual C#

iRet = object.Open()

Data type	Argument	Description	Input/Output
int	iRet	Returned value	Output

Format (when creating an object)

■Visual C++

hResult = object.Open(*lpRetCode)

Data type	Argument	Description	Input/Output
HRESULT	hResult	Returned value of COM	Output
LONG	*lpRetCode	Returned value of communication function	Output

Returned value

Normal termination: '0' is returned.

Abnormal termination: Any value other than '0' is returned. ( Page 616 ERROR CODES)

Description

The communication line is opened.

Lines are connected according to the set value of the Open function property.

Point

- If the Open function property is changed after the completion of the Open function, the setting of communication target is not changed.
To change the communication setting, close the communication line, set the setting of communication target, and open the communication line again.
- Even when a CPU type which is different from the CPU used for the communication is set to the ActCpuType property, the Open function may be completed normally.
In such a case, the connection range, usable method, or device range may be narrowed.
When executing the Open function, set the correct CPU type to the ActCpuType property.

Close (Closing communication line)

Applicable controls

This function can be used for the following controls.

- 32-bit: ActProgType, ActUtilType
- 64-bit: ActProgType64, ActUtilType64

Feature

To close a communication line.

Format

■VBA

IRet = object.Close()

Data type	Argument	Description	Input/Output
Long	IRet	Returned value	Output

■Visual C++

IRet = object.Close()

Data type	Argument	Description	Input/Output
Long	IRet	Returned value	Output

■Visual Basic .NET

IRet = object.Close()

Data type	Argument	Description	Input/Output
Integer	IRet	Returned value	Output

■Visual C#

iRet = object.Close()

Data type	Argument	Description	Input/Output
int	iRet	Returned value	Output

Format (when creating an object)

■Visual C++

hResult = object.Close(*lpIRetCode)

Data type	Argument	Description	Input/Output
HRESULT	hResult	Returned value of COM	Output
LONG	*lpIRetCode	Returned value of communication function	Output

Returned value

Normal termination: '0' is returned.

Abnormal termination: Any value other than '0' is returned. ( Page 616 ERROR CODES)

Description

The line connected using the Open function is closed.

ReadDeviceBlock (Reading devices in batch)

Applicable controls

This function can be used for the following controls.

- 32-bit: ActProgType, ActUtilType
- 64-bit: ActProgType64, ActUtilType64

Feature

To read devices in batch.

Format

■VBA

IRet = object.ReadDeviceBlock(szDevice, ISize, IData(0))

Data type	Argument	Description	Input/Output
Long	IRet	Returned value	Output
String	szDevice	Device name	Input
Long	ISize	Number of read points	Input
Long	IData(n)	Read device value	Output

■Visual C++

IRet = object.ReadDeviceBlock(szDevice, ISize, *lpIData)

Data type	Argument	Description	Input/Output
Long	IRet	Returned value	Output
CString	szDevice	Device name	Input
Long	ISize	Number of read points	Input
Long	*lpIData	Read device value	Output

■Visual Basic .NET

IRet = object.ReadDeviceBlock(szDevice, iSize, IData(0))

Data type	Argument	Description	Input/Output
Integer	IRet	Returned value	Output
String	szDevice	Device name	Input
Integer	iSize	Number of read points	Input
Integer	IData(n)	Read device value	Output

■Visual C#

iRet = object.ReadDeviceBlock(szDevice, iSize, out IData[0])

Data type	Argument	Description	Input/Output
int	iRet	Returned value	Output
String	szDevice	Device name	Input
int	iSize	Number of read points	Input
int[n]	IData	Read device value	Output

Format (when creating an object)

■ Visual C++

hResult = object.ReadDeviceBlock(szDevice, ISize, *IpData, *IplRetCode)

Data type	Argument	Description	Input/Output
HRESULT	hResult	Returned value of COM	Output
BSTR	szDevice	Device name	Input
LONG	ISize	Number of read points	Input
LONG	*IpData	Read device value	Output
LONG	*IplRetCode	Returned value of communication function	Output

Returned value

Normal termination: '0' is returned.

Abnormal termination: Any value other than '0' is returned. (🔗 Page 616 ERROR CODES)

Description

- The device values of ISize (iSize) are read in batch starting from the device specified for szDevice.
- The read device values are stored in IData (IpData or iData).
- For IData (IpData or iData), reserve arrays for more than the amount specified for ISize (iSize).

Point

- The maximum number of read points for ISize (iSize) should be the value which satisfies the following condition:
Start read device number + Number of read points ≤ Last device number
- When specifying bit devices, a multiple of 16 can be specified as a device number.
- Not applicable to inverter communication/robot controller communication.
- For IData (IpData or iData), reserve a memory area for the number of points specified for ISize (iSize).
If the memory area is not reserved, a critical error (an application error or the like) may occur.
- Devices of long timer contact (LTS), long timer coil (LTC), long retentive timer contact (LSTS), long retentive timer coil (LSTC) cannot be read. Use the ReadDeviceRandom or GetDevice function.

How to specify devices

The following describes how to specify devices.

■ When specifying bit devices

Example: Read 3 points (3 words) of data in 16-point unit from the devices starting from M0.

Upper 2-byte	Lower 2-byte
Not used ('0' is stored.)	M0 to M15 ^{*1}
	M16 to M31 ^{*1}
	M32 to M47 ^{*1}

*1 Devices are stored from the lower bit in the order of device number.

■ When specifying word devices

Example: Read 3 points of data from the devices starting from D0.

Upper 2-byte	Lower 2-byte
Not used ('0' is stored.)	D0
	D1
	D2

■When specifying CN200 and later devices of FXCPU

Example: Read 6 points of data from the devices starting from CN200.

For CN200 and later devices of FXCPU, 2 words are read for each 2 points. Reading only 1 point of data will result in an error.

Upper 2-byte	Lower 2-byte
Not used ('0' is stored.)	L of CN200
	H of CN200
	L of CN201
	H of CN201
	L of CN202
	H of CN202

■When specifying FD devices (4-word devices)

Example: Read 6 points of data from the devices starting from FD0.

Upper 2-byte	Lower 2-byte
Not used ('0' is stored.)	LL of FD0
	LH of FD0
	HL of FD0
	HH of FD0
	LL of FD1
	LH of FD1

■8-bit devices assigned to gateway devices

Example: Read 4 points of data from the devices starting from EG0.

(When 8-bit devices (from E0000 to E0007 of SHARP programmable controller) are assigned from EG0 to EG3.)

Upper 2-byte	Lower 2-byte
Not used ('0' is stored.)	EG0
	(E0001) (E0000)
	EG1
	(E0003) (E0002)
	EG2
	(E0005) (E0004)
	EG3
	(E0007) (E0006)

■When specifying double word devices

Example: Read 3 points of data from the devices starting from LTN0.

Upper 2-byte	Lower 2-byte
H of LTN0	L of LTN0
H of LTN1	L of LTN1
H of LTN2	L of LTN2

WriteDeviceBlock (Writing devices in batch)

Applicable controls

This function can be used for the following controls.

- 32-bit: ActProgType, ActUtilType
- 64-bit: ActProgType64, ActUtilType64

Feature

To write devices in batch.

Format

■VBA

IRet = object.WriteDeviceBlock(szDevice, ISize, IData(0))

Data type	Argument	Description	Input/Output
Long	IRet	Returned value	Output
String	szDevice	Device name	Input
Long	ISize	Number of write points	Input
Long	IData(n)	Device value to be written	Input

■Visual C++

IRet = object.WriteDeviceBlock(szDevice, ISize, *lpData)

Data type	Argument	Description	Input/Output
Long	IRet	Returned value	Output
CString	szDevice	Device name	Input
Long	ISize	Number of write points	Input
Long	*lpData	Device value to be written	Input

■Visual Basic .NET

IRet = object.WriteDeviceBlock(szDevice, iSize, IData(0))

Data type	Argument	Description	Input/Output
Integer	IRet	Returned value	Output
String	szDevice	Device name	Input
Integer	iSize	Number of write points	Input
Integer	IData(n)	Device value to be written	Input

■Visual C#

iRet = object.WriteDeviceBlock(szDevice, iSize, ref IData[0])

Data type	Argument	Description	Input/Output
int	iRet	Returned value	Output
String	szDevice	Device name	Input
int	iSize	Number of write points	Input
int[n]	IData	Device value to be written	Input

Format (when creating an object)

■ Visual C++

hResult = object.WriteDeviceBlock(szDevice, ISize, *IpIData, *IpIRetCode)

Data type	Argument	Description	Input/Output
HRESULT	hResult	Returned value of COM	Output
BSTR	szDevice	Device name	Input
LONG	ISize	Number of write points	Input
LONG	*IpIData	Device value to be written	Input
LONG	*IpIRetCode	Returned value of communication function	Output

Returned value

Normal termination: '0' is returned.

Abnormal termination: Any value other than '0' is returned. (🔗 Page 616 ERROR CODES)

Description

- The device values of ISize (iSize) are written in batch starting from the device specified for szDevice.
- The device values to be written are stored in IData (IpIData or iData).
- For IData (IpIData or iData), reserve arrays for more than the amount specified for ISize (iSize).

Point

- The maximum number of write points for ISize (iSize) should be the value which satisfies the following condition:
Start write device number + Number of write points ≤ Last device number
- When specifying bit devices, a multiple of 16 can be specified as a device number.
- For IData (IpIData or iData), reserve a memory area for the number of points specified for ISize (iSize).
If the memory area is not reserved, a critical error (an application error or the like) may occur.
- Devices of long timer contact (LTS), long timer coil (LTC), long retentive timer contact (LSTS), long retentive timer coil (LSTC) cannot be written. Use the WriteDeviceRandom or SetDevice function.
- If the function is run against the safety device in the RnSFPCPU safety mode, an error code 0x010A42A5" (an operation that can not be carried out in safety mode was performed) is returned.
- Not applicable to inverter communication/robot controller communication.

How to specify devices

The following describes how to specify devices.

■When specifying bit devices

Example: Write 3 points (3 words) of data in 16-point unit to the devices starting from M0.

Upper 2-byte	Lower 2-byte
Not used ('0' is stored.)	M0 to M15 ^{*1}
	M16 to M31 ^{*1}
	M32 to M47 ^{*1}

*1 Devices are stored from the lower bit in the order of device number.

■When specifying word devices

Example: Write 3 points of data to the devices starting from D0.

Upper 2-byte	Lower 2-byte
Not used ('0' is stored.)	D0
	D1
	D2

■When specifying CN200 and later devices of FXCPU

Example: Write 6 points of data to the devices starting from CN200.

For CN200 and later devices of FXCPU, 2 words are written for each 2 points. Writing only 1 point of data will result in an error.

Upper 2-byte	Lower 2-byte
Not used ('0' is stored.)	L of CN200
	H of CN200
	L of CN201
	H of CN201
	L of CN202
	H of CN202

■When specifying FD devices (4-word devices)

Example: Write 6 points of data to the devices starting from FD0.

Upper 2-byte	Lower 2-byte
Not used ('0' is stored.)	LL of FD0
	LH of FD0
	HL of FD0
	HH of FD0
	LL of FD1
	LH of FD1

■8-bit devices assigned to gateway devices

Example: Write 4 points of data to the devices starting from EG0.

(When 8-bit devices (from E0000 to E0007 of SHARP programmable controller) are assigned from EG0 to EG3.)

Upper 2-byte	Lower 2-byte
Not used ('0' is stored.)	EG0
	(E0001) (E0000)
	EG1
	(E0003) (E0002)
	EG2
	(E0005) (E0004)
	EG3
	(E0007) (E0006)

■When specifying double word devices

Example: Write 3 points of data to the devices starting from LTN0.

Upper 2-byte	Lower 2-byte
H of LTN0	L of LTN0
H of LTN1	L of LTN1
H of LTN2	L of LTN2

ReadDeviceRandom (Reading devices randomly)

Applicable controls

This function can be used for the following controls.

- 32-bit: ActProgType, ActUtilType
- 64-bit: ActProgType64, ActUtilType64

Feature

To read devices randomly.

Read monitor types of inverters/robot controllers randomly.

Format

■VBA

IRet = object.ReadDeviceRandom(szDeviceList, ISize, IData(0))

Data type	Argument	Description	Input/Output
Long	IRet	Returned value	Output
String	szDeviceList	Device name/monitor type	Input
Long	ISize	Number of read points	Input
Long	IData(n)	Read device value	Output

■Visual C++

IRet = object.ReadDeviceRandom(szDeviceList, ISize, *lpIData)

Data type	Argument	Description	Input/Output
Long	IRet	Returned value	Output
CString	szDeviceList	Device name/monitor type	Input
Long	ISize	Number of read points	Input
Long	*lpIData	Read device value	Output

■Visual Basic .NET

IRet = object.ReadDeviceRandom(szDeviceList, iSize, IData(0))

Data type	Argument	Description	Input/Output
Integer	IRet	Returned value	Output
String	szDeviceList	Device name/monitor type	Input
Integer	iSize	Number of read points	Input
Integer	IData(n)	Read device value	Output

■Visual C#

iRet = object.ReadDeviceRandom(szDevice, iSize, out IData[0])

Data type	Argument	Description	Input/Output
int	iRet	Returned value	Output
String	szDevice	Device name/monitor type	Input
int	iSize	Number of read points	Input
int[n]	IData	Read device value	Output

Format (when creating an object)

■ Visual C++

hResult = object.ReadDeviceRandom(szDevice, ISize, *lpIData, *lpRetCode)

Data type	Argument	Description	Input/Output
HRESULT	hResult	Returned value of COM	Output
BSTR	szDevice	Device name/monitor type	Input
LONG	ISize	Number of read points	Input
LONG	*lpIData	Read device value	Output
LONG	*lpRetCode	Returned value of communication function	Output

Returned value

Normal termination: '0' is returned.

Abnormal termination: Any value other than '0' is returned. (👉 Page 616 ERROR CODES)

Description

- Data of a device group/monitor type group specified for szDeviceList (szDevice) is written for the size of ISize (iSize).
- The read device values are stored in IData (lpIData or iData).
- Use the line feed symbol to separate the character string specified for the device list. The line feed symbol is not necessary to be suffixed to the last device.

Example:

When using Visual Basic .NET, VBA: "D0" & vbLf & "D1" & vbLf & "D2"

When using Visual C++: D0\nD1\nD2

When using Visual C#: D0\nD1\nD2

- For IData (lpIData or iData), reserve arrays for more than the amount specified for ISize (iSize).
- For the items that can be monitored (monitor types), refer to the following:

👉 Page 154 ACCESSIBLE DEVICES AND DEVICE TYPES

Point

- The maximum number of read points for ISize (iSize) is 0x7FFFFFFF.
- For IData (lpIData or iData), reserve a memory area for the number of points specified for ISize (iSize). If the memory area is not reserved, a critical error (an application error or the like) may occur.

How to specify devices

The following describes how to specify devices.

■When devices are specified as follows (number of points: 3 points)

When using Visual Basic .NET, VBA: "M0" & vbLf & "D0" & vbLf & "K8M0"

When using Visual C++: M0\nD0\nK8M0

When using Visual C#: M0\nD0\nK8M0

Upper 2-byte	Lower 2-byte
Not used ('0' is stored.)	M0
	D0
M16 to M31 ^{*1}	M0 to M15 ^{*1}

*1 Devices are stored from the lower bit in the order of device number.

■When devices including CN200 and later of FXCPU are specified (total number of points: 3 points)

For CN200 and later devices of FXCPU, 2 words are read for each point when reading devices randomly.

When using Visual Basic .NET, VBA: "D0" & vbLf & "CN200" & vbLf & "D1"

When using Visual C++: D0\nCN200\nD1

When using Visual C#: D0\nCN200\nD1

Upper 2-byte	Lower 2-byte
Not used ('0' is stored.)	D0
H of CN200	L of CN200
Not used ('0' is stored.)	D1

■When devices including FD are specified (total number of points: 3 points)

When using Visual Basic .NET, VBA: "D0" & vbLf & "FD0" & vbLf & "D1"

When using Visual C++: D0\nFD0\nD1

When using Visual C#: D0\nFD0\nD1

Upper 2-byte	Lower 2-byte
Not used ('0' is stored.)	D0
	LL of FD0
	D1

■When devices including EG are specified (total number of points: 3 points)

The following is the example when 8-bit devices (E0000 and E0001 of SHARP programmable controller) are assigned to EG0.

When using Visual Basic .NET, VBA: "D0" & vbLf & "EG0" & vbLf & "D1"
 When using Visual C++: D0\nEG0\nD1
 When using Visual C#: D0\nEG0\nD1

Upper 2-byte	Lower 2-byte
Not used ('0' is stored.)	D0
	EG0
	(E0001)
	D1

■When double word devices are specified

When using Visual Basic .NET, VBA: "LTN0" & vbLf & "LTN1" & vbLf & "LTN2"
 When using Visual C++: LTN0\nLTN1\nLTN2
 When using Visual C#: LTN0\nLTN1\nLTN2

Upper 2-byte	Lower 2-byte
H of LTN0	L of LTN0
H of LTN1	L of LTN1
H of LTN2	L of LTN2

■When monitor types (1, 2, and 5) of inverter are specified (total number of points: 3 points)

When using Visual Basic .NET, VBA: "1" & vbLf & "2" & vbLf & "5"
 When using Visual C++: 1\n2\n5
 When using Visual C#: 1\n2\n5

Upper 2-byte	Lower 2-byte
H of 1	L of 1
H of 2	L of 2
H of 5	L of 5

■When monitor types (223.102.A, and 223.103.B) of robot controller are specified (total number of points: 2 points)

Format: (Request ID).(Data type).(Argument)

When using Visual Basic .NET, VBA: "223.102.A" & vbLf & "223.103.B"
 When using Visual C++: 223.102.A\n223.103.B
 When using Visual C#: 223.102.A\n223.103.B

Upper 2-byte	Lower 2-byte
H of 223.102.A	L of 223.102.A
H of 223.103.B	L of 223.103.B

WriteDeviceRandom (Writing devices randomly)

Applicable controls

This function can be used for the following controls.

- 32-bit: ActProgType, ActUtilType
- 64-bit: ActProgType64, ActUtilType64

Feature

To write devices randomly.

Format

■VBA

IRet = object.WriteDeviceRandom(szDeviceList, ISize, IData(0))

Data type	Argument	Description	Input/Output
Long	IRet	Returned value	Output
String	szDeviceList	Device name	Input
Long	ISize	Number of write points	Input
Long	IData(n)	Device value to be written	Input

■Visual C++

IRet = object.WriteDeviceRandom(szDeviceList, ISize, *lpIData)

Data type	Argument	Description	Input/Output
Long	IRet	Returned value	Output
CString	szDeviceList	Device name	Input
Long	ISize	Number of write points	Input
Long	*lpIData	Device value to be written	Input

■Visual Basic .NET

IRet = object.WriteDeviceRandom(szDeviceList, iSize, IData(0))

Data type	Argument	Description	Input/Output
Integer	IRet	Returned value	Output
String	szDeviceList	Device name	Input
Integer	iSize	Number of write points	Input
Integer	iData(n)	Device value to be written	Input

■Visual C#

iRet = object.WriteDeviceRandom(szDevice, iSize, ref IData[0])

Data type	Argument	Description	Input/Output
int	iRet	Returned value	Output
String	szDevice	Device name	Input
int	iSize	Number of write points	Input
int[n]	iData	Device value to be written	Input

Format (when creating an object)

■ Visual C++

hResult = object.WriteDeviceRandom(szDeviceList, ISize, *IpIData, *IpIRetCode)

Data type	Argument	Description	Input/Output
HRESULT	hResult	Returned value of COM	Output
BSTR	szDeviceList	Device name	Input
LONG	ISize	Number of write points	Input
LONG	*IpIData	Device value to be written	Input
LONG	*IpIRetCode	Returned value of communication function	Output

Returned value

Normal termination: '0' is returned.

Abnormal termination: Any value other than '0' is returned. (🔗 Page 616 ERROR CODES)

Description

- Data of a device group specified for szDeviceList (szDevice) is written for the size of ISize (iSize).
- The device values to be written are stored in IData (IpIData or iData).
- Use the line feed symbol to separate the character string specified for the device list. The line feed symbol is not necessary to be suffixed to the last device.

Example:

When using Visual Basic .NET, VBA: "D0" & vbLf & "D1" & vbLf & "D2"

When using Visual C++: D0\nD1\nD2

When using Visual C#: D0\nD1\nD2

- For IData (IpIData or iData), reserve arrays for more than the amount specified for ISize (iSize).

Point

- The maximum number of write points for ISize (iSize) is 0x7FFFFFFF.
- For IData (IpIData or iData), reserve a memory area for the number of points specified for ISize (iSize). If the memory area is not reserved, a critical error (an application error or the like) may occur.
- An error is returned when a Q motion CPU is accessed.
- If the function is run against the safety device in the RnSF CPU safety mode, an error code 0x010A42A5¹ (an operation that can not be carried out in safety mode was performed) is returned.
- Not applicable to inverter communication/robot controller communication.

How to specify devices

The following describes how to specify devices.

■ When devices are specified as follows (number of points: 3 points)

When using Visual Basic .NET, VBA: "M0" & vbLf & "D0" & vbLf & "K8M0"

When using Visual C++: M0\nD0\nK8M0

When using Visual C#: M0\nD0\nK8M0

Upper 2-byte	Lower 2-byte
Not used ('0' is stored.)	M0
	D0
M16 to M31 ^{*1}	M0 to M15 ^{*1}

*1 Devices are stored from the lower bit in the order of device number.

■When devices including CN200 and later of FXCPU are specified (total number of points: 3 points)

For CN200 or later of FXCPU, 2 words are written for each point when writing devices randomly.

When using Visual Basic .NET, VBA: "D0" & vbLf & "CN200" & vbLf & "D1"
 When using Visual C++: D0\nCN200\nD1
 When using Visual C#: D0\nCN200\nD1

Upper 2-byte	Lower 2-byte
Not used ('0' is stored.)	D0
H of CN200	L of CN200
Not used ('0' is stored.)	D1

■When devices including FD are specified (total number of points: 3 points)

Upper 2-byte	Lower 2-byte
Not used ('0' is stored.)	D0
	LL of FD0
	D1

■When devices including EG are specified (total number of points: 3 points)

The following is the example when 8-bit devices (E0000 and E0001 of SHARP programmable controller) are assigned to EG0.

When using Visual Basic .NET, VBA: "D0" & vbLf & "EG0" & vbLf & "D1"
 When using Visual C++: D0\nEG0\nD1
 When using Visual C#: D0\nEG0\nD1

Upper 2-byte	Lower 2-byte
Not used ('0' is stored.)	D0
	EG0
	(E0001) (E0000)
	D1

■When double word devices are specified

When using Visual Basic .NET, VBA: "LTN0" & vbLf & "LTN1" & vbLf & "LTN2"
 When using Visual C++: LTN0\nLTN1\nLTN2
 When using Visual C#: LTN0\nLTN1\nLTN2

Upper 2-byte	Lower 2-byte
H of LTN0	L of LTN0
H of LTN1	L of LTN1
H of LTN2	L of LTN2

SetDevice (Setting device data)

Applicable controls

This function can be used for the following controls.

- 32-bit: ActProgType, ActUtilType
- 64-bit: ActProgType64, ActUtilType64

Feature

To set one point of device.

Format

■VBA

IRet = object.SetDevice(szDevice, IData)

Data type	Argument	Description	Input/Output
Long	IRet	Returned value	Output
String	szDevice	Device name	Input
Long	IData	Set data	Input

■Visual C++

IRet = object.SetDevice(szDevice, IData)

Data type	Argument	Description	Input/Output
Long	IRet	Returned value	Output
CString	szDevice	Device name	Input
Long	IData	Set data	Input

■Visual Basic .NET

IRet = object.SetDevice(szDevice, IData)

Data type	Argument	Description	Input/Output
Integer	IRet	Returned value	Output
String	szDevice	Device name	Input
Integer	IData	Set data	Input

■Visual C#

iRet = object.SetDevice(szDevice, IData)

Data type	Argument	Description	Input/Output
int	iRet	Returned value	Output
String	szDevice	Device name	Input
int	IData	Set data	Input

Format (when creating an object)

■Visual C++

hResult = object.SetDevice(szDevice, IData, *IpIRetCode)

Data type	Argument	Description	Input/Output
HRESULT	hResult	Returned value of COM	Output
BSTR	szDevice	Device name	Input
LONG	IData	Set data	Input
LONG	*IpIRetCode	Returned value of communication function	Output

Returned value

Normal termination: '0' is returned.

Abnormal termination: Any value other than '0' is returned. ( Page 616 ERROR CODES)

Description

- One point of device specified for szDevice is specified using IData (iData).
- When specifying bit devices, the least significant bit of the IData value (varData value or iData value) is valid.

Point

- If the function is run against the safety device in the RnSF CPU safety mode, an error code 0x010A42A5" (an operation that can not be carried out in safety mode was performed) is returned.
- Not applicable to inverter communication/robot controller communication.

How to specify devices

The following describes how to specify devices.

■When specifying bit devices

Example: M0

Upper 2-byte	Lower 2-byte
Not used ('0' is stored.)	M0

■When specifying word devices

Example: D0

Upper 2-byte	Lower 2-byte
Not used ('0' is stored.)	D0

■When specifying double word devices

Example: LTN0

Upper 2-byte	Lower 2-byte
H of LTN0	L of LTN0

■When specifying double word devices (digit specification)

Example: K8M0

Upper 2-byte	Lower 2-byte
M16 to M31*1	M0 to M15*1

*1 Devices are stored from the lower bit in the order of device number.

■When specifying CN200 and later devices of FXCPU

Example: CN200

Upper 2-byte	Lower 2-byte
H of CN200	L of CN200

■When specifying gateway devices

Example: When 8-bit devices (E0000 and E0001 of SHARP programmable controller) are assigned to EG0

Upper 2-byte	Lower 2-byte
Not used ('0' is stored.)	EG0
	(E0001) (E0000)

GetDevice (Acquiring device data)

Applicable controls

This function can be used for the following controls.

- 32-bit: ActProgType, ActUtilType
- 64-bit: ActProgType64, ActUtilType64

Feature

To acquire one point of device.

Acquire one point monitor type data of inverter/robot controller.

Format

■VBA

IRet = object.GetDevice(szDevice, IData)

Data type	Argument	Description	Input/Output
Long	IRet	Returned value	Output
String	szDevice	Device name/monitor type	Input
Long	IData	Acquired data	Output

■Visual C++

IRet = object.GetDevice(szDevice, *lpIData)

Data type	Argument	Description	Input/Output
Long	IRet	Returned value	Output
CString	szDevice	Device name/monitor type	Input
Long	*lpIData	Acquired data	Output

■Visual Basic .NET

IRet = object.GetDevice(szDevice, IData)

Data type	Argument	Description	Input/Output
Integer	IRet	Returned value	Output
String	szDevice	Device name/monitor type	Input
Integer	IData	Acquired data	Output

■Visual C#

iRet = object.GetDevice(szDevice, out IData)

Data type	Argument	Description	Input/Output
int	iRet	Returned value	Output
String	szDevice	Device name/monitor type	Input
int	IData	Acquired data	Output

Format (when creating an object)

■ Visual C++

hResult = object.GetDevice(szDevice, *lpData, *lpRetCode)

Data type	Argument	Description	Input/Output
HRESULT	hResult	Returned value of COM	Output
BSTR	szDevice	Device name/monitor type	Input
LONG	*lpData	Acquired data	Output
LONG	*lpRetCode	Returned value of communication function	Output

Returned value

Normal termination: '0' is returned.

Abnormal termination: Any value other than '0' is returned. ( Page 616 ERROR CODES)

Description

One point of device data specified for szDevice is stored in IData (lpData or iData).

For the items that can be monitored (monitor types), refer to the following:

 Page 154 ACCESSIBLE DEVICES AND DEVICE TYPES

How to specify devices

The following describes how to specify devices/monitor type.

■ When specifying bit devices

Example: M0

Upper 2-byte	Lower 2-byte
Not used ('0' is stored.)	M0

■ When specifying word devices

Example: D0

Upper 2-byte	Lower 2-byte
Not used ('0' is stored.)	D0

■ When specifying double word devices

Example: LTNO

Upper 2-byte	Lower 2-byte
H of LTNO	L of LTNO

■ When specifying double word devices (digit specification)

Example: K8M0

Upper 2-byte	Lower 2-byte
M16 to M31*1	M0 to M15*1

*1 Devices are stored from the lower bit in the order of device number.

■When specifying CN200 and later devices of FXCPU

Example: CN200

Upper 2-byte	Lower 2-byte
H of CN200	L of CN200

■When specifying gateway devices

Example: When 8-bit devices (E0000 and E0001 of SHARP programmable controller) are assigned to EG0

Upper 2-byte	Lower 2-byte	
Not used ('0' is stored.)	EG0	
	(E0001)	(E0000)

■When specifying monitor types of inverter

Example: 2

Upper 2-byte	Lower 2-byte
H of 1	L of 1

■When specifying monitor types of robot controller

Format: (Request ID).(Data type).(Argument)

Example: 223.102.A

Upper 2-byte	Lower 2-byte
H of 223.102.A	L of 223.102.A

ReadBuffer (Reading data from buffer memory)

Applicable controls

This function can be used for the following controls.

- 32-bit: ActProgType, ActUtilType
- 64-bit: ActProgType64, ActUtilType64

Feature

To read data from the buffer memory of an intelligent function module.

Format

■VBA

IRet = object.ReadBuffer(IStartIO,IAddress,IReadSize, iData(0))

Data type	Argument	Description	Input/Output
Long	IRet	Returned value	Output
Long	IStartIO	Start I/O number of module from which values are read	Input
Long	IAddress	Buffer memory address	Input
Long	IReadSize	Read size	Input
Integer	iData(n)	Values read from buffer memory	Output

■Visual C++

IRet = object.ReadBuffer(IStartIO, IAddress, IReadSize, *lpsData)

Data type	Argument	Description	Input/Output
Long	IRet	Returned value	Output
Long	IStartIO	Start I/O number of module from which values are read	Input
Long	IAddress	Buffer memory address	Input
Long	IReadSize	Read size	Input
Short	*lpsData	Values read from buffer memory	Output

■Visual Basic .NET

IRet = object.ReadBuffer(iStartIO, iAddress, iReadSize, iData(0))

Data type	Argument	Description	Input/Output
Integer	IRet	Returned value	Output
Integer	iStartIO	Start I/O number of module from which values are read	Input
Integer	iAddress	Buffer memory address	Input
Integer	iReadSize	Read size	Input
short	iData(n)	Values read from buffer memory	Output

■Visual C#

iRet = object.ReadBuffer(iStartIO, iAddress, iReadSize, out sData[0])

Data type	Argument	Description	Input/Output
int	iRet	Returned value	Output
int	iStartIO	Start I/O number of module from which values are read	Input
int	iAddress	Buffer memory address	Input
int	iReadSize	Read size	Input
short[n]	sData	Values read from buffer memory	Output

Format (when creating an object)

■ Visual C++

hResult = object.ReadBuffer(IStartIO, IAddress, IReadSize, *IpsData, *IplRetCode)

Data type	Argument	Description	Input/Output
HRESULT	hResult	Returned value of COM	Output
LONG	IStartIO	Start I/O number of module from which values are read	Input
LONG	IAddress	Buffer memory address	Input
LONG	IReadSize	Read size	Input
SHORT	*IpsData	Values read from buffer memory	Output
LONG	*IplRetCode	Returned value of communication function	Output

Returned value

Normal termination: '0' is returned.

Abnormal termination: Any value other than '0' is returned. (🔍 Page 616 ERROR CODES)

Description

- For the start I/O number of the module specified for IStartIO (iStartIO), specify the value of the actual start I/O number divided by 16.
For an FX5CPU, specify the module number on the target station side.
- Buffer values of buffer memory address specified for IAddress (iAddress) of an intelligent function module of the start I/O number specified for IStartIO (iStartIO) are read for the size of IReadSize (iReadSize).
- When performing CPU COM communication with FXCPU as a connected station or GX Simulator2 communication, specify the block number (0 to 7) of a special expansion device for the start I/O number of a module and 0 to 32767 for the buffer memory address.
- For iData (IpsData or sData), reserve arrays for more than the amount specified for IReadSize (iReadSize).

Point

- An error is returned when an R motion CPU or a Q motion CPU is accessed.
- For iData (IpsData or sData), reserve a memory area for the number of points specified for IReadSize (iReadSize).
If the memory area is not reserved, a critical error (an application error or the like) may occur.
- When reading data from buffer memory (ReadBuffer) on QCPU (Q mode), the read operation can only be performed on the Q series-dedicated modules.
Furthermore, data cannot be read from the shared memory of QCPU (Q mode).
- Not applicable to inverter communication/robot controller communication.

Applicable communication routes

The following table shows the applicable communication routes.

○: Applicable, ×: Not applicable

Communication route		Property setting		Applicability	
		ActUnitType	ActProtocolType		
Serial communication	Connected module: MELSEC iQ-R series-compatible C24	UNIT_RJ71C24	PROTOCOL_SERIAL	○	
	Connected module: Q series-compatible C24	UNIT_QJ71C24		○	
	Connected module: L series-compatible C24	UNIT_LJ71C24		○	
	Connected module: FX extended port	UNIT_FX485BD		×	
Ethernet communication		UNIT_RJ71EN71	PROTOCOL_TCPIP, PROTOCOL_UDPIP	○	
		UNIT_RJ71EN71_DIRECT	PROTOCOL_UDPIP		
		UNIT_QJ71E71	PROTOCOL_TCPIP, PROTOCOL_UDPIP		
		UNIT_LJ71E71	PROTOCOL_TCPIP, PROTOCOL_UDPIP		
		UNIT_RETHETHER	PROTOCOL_TCPIP, PROTOCOL_UDPIP		
		UNIT_RETHETHER_DIRECT	PROTOCOL_UDPIP		
		UNIT_FXVETHER	PROTOCOL_TCPIP		○
		UNIT_FXVETHER_DIRECT	PROTOCOL_UDPIP		
		UNIT_LNETHER	PROTOCOL_TCPIP		○
		UNIT_LNETHER, UNIT_LNETHER_DIRECT	PROTOCOL_UDPIP		
		UNIT_QNETHER	PROTOCOL_TCPIP		
		UNIT_QNETHER, UNIT_QNETHER_DIRECT	PROTOCOL_UDPIP		
		UNIT_RJ71GN11	PROTOCOL_TCPIP, PROTOCOL_UDPIP		
		UNIT_NZ2GF_ETB	PROTOCOL_TCPIP		
		UNIT_NZ2GF_ETB, UNIT_NZ2GF_ETB_DIRECT	PROTOCOL_UDPIP		
	UNIT_FXETHER	PROTOCOL_TCPIP			
	UNIT_FXETHER_DIRECT	PROTOCOL_UDPIP			
CPU COM communication	Connection target CPU: FX5CPU	UNIT_FXVCPU	PROTOCOL_SERIAL	○	
	Connection target CPU: QCPU (Q mode)	UNIT_QNCPU		○	
	Connection target CPU: LCPU	UNIT_LNCPUCPU		○	
	Connection target CPU: Q motion CPU	UNIT_QNMOTION		×	
	Connection target CPU: FXCPU	UNIT_FXCPU		○ ^{*1}	
USB communication	Connection target CPU: RCPUCPU	UNIT_RUSB	PROTOCOL_USB	○	
	Connection target CPU: LHCPUCPU	UNIT_LHUSB		○	
	Connection target CPU: FX5CPUCPU	UNIT_FXVUSB		○	
	Connection target CPU: QCPU (Q mode)	UNIT_QNUSB		○	
	Connection target CPU: LCPUCPU	UNIT_LNUSB		○	
	Connection target CPU: QSCPCPU	UNIT_QSUSB		○	
	Connection target CPU: Q motion CPUCPU	UNIT_QNMOTIONUSB		×	
	Connection target CPU: FXCPUCPU	UNIT_FXCPU		○	
MELSECNET/H communication	UNIT_MNETHBOARD	PROTOCOL_MNETH	○ ^{*2}		
CC-Link IE Controller Network communication	UNIT_MNETGBOARD	PROTOCOL_MNETG			
CC-Link IE Field Network communication	UNIT_CCIEFBOARD	PROTOCOL_CCIEF			
CC-Link IE TSN communication	UNIT_CCIETSNBOARD	PROTOCOL_CCIETSN			
CC-Link communication	UNIT_CCLINKBOARD	PROTOCOL_CCLINK			

Communication route	Property setting		Applicability
	ActUnitType	ActProtocolType	
CC-Link system RS-232 interface communication	UNIT_G4QCPU	PROTOCOL_SERIAL	○
GX Simulator2 communication	UNIT_SIMULATOR2	—	○
GX Simulator3 communication (RCPU)	UNIT_SIMULATOR3		×
GX Simulator3 communication (FX5CPU)			
MT Simulator2 communication	UNIT_SIMULATOR2	—	×
GOT gateway device communication	UNIT_A900GOT	PROTOCOL_UDPIP	×
GOT transparent function communication	UNIT_RJ71C24, UNIT_FXVCPU, UNIT_QNCP, UNIT_LNCP	PROTOCOL_SERIAL, PROTOCOL_USBGOT	○
	UNIT_FXCPU		○*1
	UNIT_QJ71C24, UNIT_LJ71C24, UNIT_GOT_RJ71EN71, UNIT_GOT_QJ71E71, UNIT_GOT_LJ71E71, UNIT_GOT_REETHER, UNIT_GOT_QNETHER, UNIT_GOT_LNETHER, UNIT_A900GOT, UNIT_GOT_NZ2GF_ETB, UNIT_GOT_FXETHER, UNIT_GOT_FXENET		○
	UNIT_GOTETHER_RJ71C24, UNIT_GOTETHER_FXVCPU, UNIT_GOTETHER_QNCP, UNIT_GOTETHER_LNCP		○
	UNIT_GOTETHER_FXCPU		○*1
	UNIT_GOTETHER_QBUS		○
Inverter communication	UNIT_GOTETHER_QN_ETHER	PROTOCOL_TCPIP	○
	—		○
	—		○
Inverter communication	—	COMM_RS232C, COMM_USB	×

*1 An error is returned when the CPU is other than FX3U, FX3UC, FX3G and FX3GC.

*2 An error is returned when the own board is accessed.

WriteBuffer (Writing data to buffer memory)

Applicable controls

This function can be used for the following controls.

- 32-bit: ActProgType, ActUtilType
- 64-bit: ActProgType64, ActUtilType64

Feature

To write data to the buffer memory of an intelligent function module.

Format

■VBA

IRet = object.WriteBuffer(IStartIO, IAddress, IWriteSize, iData(0))

Data type	Argument	Description	Input/Output
Long	IRet	Returned value	Output
Long	IStartIO	Start I/O number of module to which values are written	Input
Long	IAddress	Buffer memory address	Input
Long	IWriteSize	Write size	Input
Integer	iData(n)	Values written from buffer memory	Input

■Visual C++

IRet = object.WriteBuffer(IStartIO, IAddress, IWriteSize, *IpsData)

Data type	Argument	Description	Input/Output
Long	IRet	Returned value	Output
Long	IStartIO	Start I/O number of module to which values are written	Input
Long	IAddress	Buffer memory address	Input
Long	IWriteSize	Write size	Input
Short	*IpsData	Values written from buffer memory	Input

■Visual Basic .NET

IRet = object.WriteBuffer(iStartIO, iAddress, iWriteSize, iData(0))

Data type	Argument	Description	Input/Output
Integer	IRet	Returned value	Output
Integer	iStartIO	Start I/O number of module to which values are written	Input
Integer	iAddress	Buffer memory address	Input
Integer	iWriteSize	Write size	Input
short	iData(n)	Values written from buffer memory	Input

■Visual C#

iRet = object. WriteBuffer (iStartIO, iAddress, iWriteSize, ref sData[0])

Data type	Argument	Description	Input/Output
int	iRet	Returned value	Output
int	iStartIO	Start I/O number of module to which values are written	Input
int	iAddress	Buffer memory address	Input
int	iWriteSize	Write size	Input
short[n]	sData	Values written from buffer memory	Input

Format (when creating an object)

■ Visual C++

hResult = object.WriteBuffer(IStartIO, IAddress, IWriteSize, *IpsData, *IplRetCode)

Data type	Argument	Description	Input/Output
HRESULT	hResult	Returned value of COM	Output
LONG	IStartIO	Start I/O number of module to which values are written	Input
LONG	IAddress	Buffer memory address	Input
LONG	IWriteSize	Write size	Input
SHORT	*IpsData	Values written from buffer memory	Input
LONG	*IplRetCode	Returned value of communication function	Output

Returned value

Normal termination: '0' is returned.

Abnormal termination: Any value other than '0' is returned. (🔗 Page 616 ERROR CODES)

Description

- For the start I/O number of the module specified for IStartIO (iStartIO), specify the value of the actual start I/O number divided by 16.
For an FX5CPU, specify the module number on the target station side.
- Buffer values of buffer memory address specified for IAddress (iAddress) of an intelligent function module of the start I/O number specified for IStartIO (iStartIO) are written for the size of IWriteSize (iWriteSize).
- When performing CPU COM communication with FXCPU as a connected station or GX Simulator2 communication, specify the block number (0 to 7) of a special expansion device for the start I/O number of a module and 0 to 32767 for the buffer memory address.
- For iData (IpsData or sData), reserve arrays for more than the amount specified for IWriteSize (iWriteSize).

Point

- An error is returned when an R motion CPU or a Q motion CPU is accessed.
- For iData (IpsData, varData, or sData), reserve a memory area for the number of points specified for IWriteSize (varWriteSize or iWriteSize).
If the memory area is not reserved, a critical error (an application error or the like) may occur.
- When buffer memory write (WriteBuffer) is performed for QCPU (Q mode), write operation may be performed for only the Q series-dedicated module.
Furthermore, data cannot be written to the shared memory of QCPU (Q mode).
- Not applicable to inverter communication/robot controller communication.

Applicable communication routes

The following table shows the applicable communication routes.

○: Applicable, ×: Not applicable

Communication route		Property setting		Applicability	
		ActUnitType	ActProtocolType		
Serial communication	Connected module: MELSEC iQ-R series-compatible C24	UNIT_RJ71C24	PROTOCOL_SERIAL	○	
	Connected module: Q series-compatible C24	UNIT_QJ71C24		○	
	Connected module: L series-compatible C24	UNIT_LJ71C24		○	
	Connected module: FX extended port	UNIT_FX485BD		×	
Ethernet communication		UNIT_RJ71EN71	PROTOCOL_TCPIP, PROTOCOL_UDPIP	○	
		UNIT_RJ71EN71_DIRECT	PROTOCOL_UDPIP		
		UNIT_QJ71E71	PROTOCOL_TCPIP, PROTOCOL_UDPIP		
		UNIT_LJ71E71	PROTOCOL_TCPIP, PROTOCOL_UDPIP		
		UNIT_REETHER	PROTOCOL_TCPIP, PROTOCOL_UDPIP		
		UNIT_REETHER_DIRECT	PROTOCOL_UDPIP		
		UNIT_FXVETHER	PROTOCOL_TCPIP		○
		UNIT_FXVETHER_DIRECT	PROTOCOL_UDPIP		
		UNIT_LNETHER	PROTOCOL_TCPIP		○
		UNIT_LNETHER, UNIT_LNETHER_DIRECT	PROTOCOL_UDPIP		
		UNIT_QNETHER	PROTOCOL_TCPIP		
		UNIT_QNETHER, UNIT_QNETHER_DIRECT	PROTOCOL_UDPIP		
		UNIT_RJ71GN11	PROTOCOL_TCPIP, PROTOCOL_UDPIP		
		UNIT_NZ2GF_ETB	PROTOCOL_TCPIP		
		UNIT_NZ2GF_ETB, UNIT_NZ2GF_ETB_DIRECT	PROTOCOL_UDPIP		
UNIT_FXETHER	PROTOCOL_TCPIP				
UNIT_FXETHER_DIRECT	PROTOCOL_UDPIP				
CPU COM communication	Connection target CPU: FX5CPU	UNIT_FXVCPU	PROTOCOL_SERIAL	○	
	Connection target CPU: QCPU (Q mode)	UNIT_QNCPU		○	
	Connection target CPU: LCPU	UNIT_LNCPUP		○	
	Connection target CPU: Q motion CPU	UNIT_QNMOTION		×	
	Connection target CPU: FXCPU	UNIT_FXCPU		○ ^{*1}	
USB communication	Connection target CPU: RCPUP	UNIT_RUSB	PROTOCOL_USB	○	
	Connection target CPU: LHCPUP	UNIT_LHUSB		○	
	Connection target CPU: FX5CPUP	UNIT_FXVUSB		○	
	Connection target CPU: QCPU (Q mode)	UNIT_QNUSB		○	
	Connection target CPU: LCPUP	UNIT_LNUSB		○	
	Connection target CPU: QSCPUP	UNIT_QSUSB		× ^{*2}	
	Connection target CPU: Q motion CPUP	UNIT_QNMOTIONUSB		×	
	Connection target CPU: FXCPUP	UNIT_FXCPU		○	
MELSECNET/H communication	UNIT_MNETHBOARD	PROTOCOL_MNETH	○ ^{*3}		
CC-Link IE Controller Network communication	UNIT_MNETGBOARD	PROTOCOL_MNETG			
CC-Link IE Field Network communication	UNIT_CCIEFBOARD	PROTOCOL_CCIEF			
CC-Link IE TSN communication	UNIT_CCIETSNBOARD	PROTOCOL_CCIETSN			
CC-Link communication	UNIT_CCLINKBOARD	PROTOCOL_CCLINK			

Communication route	Property setting		Applicability
	ActUnitType	ActProtocolType	
CC-Link system RS-232 interface communication	UNIT_G4QCPU	PROTOCOL_SERIAL	○
GX Simulator2 communication	UNIT_SIMULATOR2	—	○
GX Simulator3 communication (RCPU)	UNIT_SIMULATOR3		×
GX Simulator3 communication (FX5CPU)			
MT Simulator2 communication	UNIT_SIMULATOR2	—	×
GOT gateway device communication	UNIT_A900GOT	PROTOCOL_UDPIP	×
GOT transparent function communication	UNIT_RJ71C24, UNIT_FXVCPU, UNIT_QNCP, UNIT_LNCP	PROTOCOL_SERIAL, PROTOCOL_USBGOT	○
	UNIT_FXCPU		○*1
	UNIT_QJ71C24, UNIT_LJ71C24, UNIT_GOT_RJ71EN71, UNIT_GOT_QJ71E71, UNIT_GOT_LJ71E71, UNIT_GOT_REETHER, UNIT_GOT_QNETHER, UNIT_GOT_LNETHER, UNIT_A900GOT, UNIT_GOT_NZ2GF_ETB, UNIT_GOT_FXETHER, UNIT_GOT_FXENET		○
	UNIT_GOTETHER_RJ71C24, UNIT_GOTETHER_FXVCPU, UNIT_GOTETHER_QNCP, UNIT_GOTETHER_LNCP	PROTOCOL_TCPIP	○
	UNIT_GOTETHER_FXCPU		○*1
	UNIT_GOTETHER_QBUS		○
UNIT_GOTETHER_QN_ETHER	○		
Inverter communication	—	COMM_RS232C, COMM_USB	×

*1 An error is returned when the CPU is other than FX3U, FX3UC, FX3G and FX3GC.

*2 The error code "0x010A42A0" (the access password mismatch) is returned when accessing QSCPU.

*3 An error is returned when the own board is accessed.

GetClockData (Reading clock data)

Applicable controls

This function can be used for the following controls.

- 32-bit: ActProgType, ActUtilType
- 64-bit: ActProgType64, ActUtilType64

Feature

To read time from the clock data of a CPU module.

Format

■VBA

IRet = object.GetClockData(iYear, iMonth, iDay, iDayOfWeek, iHour, iMinute, iSecond)

Data type	Argument	Description	Input/Output
Long	IRet	Returned value	Output
Integer	iYear	Read year value	Output
Integer	iMonth	Read month value	Output
Integer	iDay	Read day value	Output
Integer	iDayOfWeek	Read day-of-week value	Output
Integer	iHour	Read hour value	Output
Integer	iMinute	Read minute value	Output
Integer	iSecond	Read second value	Output

■Visual C++

IRet = object.GetClockData(*IpsYear, *IpsMonth, *IpsDay, *IpsDayOfWeek, *IpsHour, *IpsMinute, *IpsSecond)

Data type	Argument	Description	Input/Output
Long	IRet	Returned value	Output
Short	*IpsYear	Read year value	Output
Short	*IpsMonth	Read month value	Output
Short	*IpsDay	Read day value	Output
Short	*IpsDayOfWeek	Read day-of-week value	Output
Short	*IpsHour	Read hour value	Output
Short	*IpsMinute	Read minute value	Output
Short	*IpsSecond	Read second value	Output

■Visual Basic .NET

IRet = object.GetClockData(iYear, iMonth, iDay, iDayOfWeek, iHour, iMinute, iSecond)

Data type	Argument	Description	Input/Output
Integer	IRet	Returned value	Output
short	iYear	Read year value	Output
short	iMonth	Read month value	Output
short	iDay	Read day value	Output
short	iDayOfWeek	Read day-of-week value	Output
short	iHour	Read hour value	Output
short	iMinute	Read minute value	Output
short	iSecond	Read second value	Output

■Visual C#

hResult = object.GetClockData(out sYear, out sMonth, out sDay, out sDayOfWeek, out sHour, out sMinute, out sSecond)

Data type	Argument	Description	Input/Output
int	iRet	Returned value	Output
short	sYear	Read year value	Output

Data type	Argument	Description	Input/Output
short	sMonth	Read month value	Output
short	sDay	Read day value	Output
short	sDayOfWeek	Read day-of-week value	Output
short	sHour	Read hour value	Output
short	sMinute	Read minute value	Output
short	sSecond	Read second value	Output

Format (when creating an object)

■ Visual C++

hResult = object.GetClockData(*lpsYear, *lpsMonth, *lpsDay, *lpsDayOfWeek, *lpsHour, *lpsMinute, *lpsSecond, *lplRetCode)

Data type	Argument	Description	Input/Output
HRESULT	hResult	Returned value of COM	Output
SHORT	*lpsYear	Read year value	Output
SHORT	*lpsMonth	Read month value	Output
SHORT	*lpsDay	Read day value	Output
SHORT	*lpsDayOfWeek	Read day-of-week value	Output
SHORT	*lpsHour	Read hour value	Output
SHORT	*lpsMinute	Read minute value	Output
SHORT	*lpsSecond	Read second value	Output
LONG	*lplRetCode	Returned value of communication function	Output

Returned value

Normal termination: '0' is returned.

Abnormal termination: Any value other than '0' is returned. (📖 Page 616 ERROR CODES)

Description

- An error is returned when the correct clock data is not set to the CPU module.
- The values stored in iYear (lpsYear or sYear) are: the four digits of year for RCPUs and QCPUs (Q mode) and the last two digits of year for any other CPUs.

Note that the applicable years for RCPUs and QCPUs (Q mode) are from 1980 to 2079.

- The values to be stored in iDayOfWeek (lpsDayOfWeek or sDayOfWeek) are as follows:

Value	Day of Week
0	Sunday
1	Monday
2	Tuesday
3	Wednesday
4	Thursday
5	Friday
6	Saturday

Point

- An error is returned when an R motion CPU or a Q motion CPU is accessed.
- The clock data can be read on FXCPUs with the built-in clock.
For FXCPUs without the built-in clock, an error is returned.
- Note that an error of transfer time may occur in clock setting.
- Not applicable to inverter communication/robot controller communication.

Applicable communication routes

The following table shows the applicable communication routes.

○: Applicable, ×: Not applicable

Communication route		Property setting		Applicability
		ActUnitType	ActProtocolType	
Serial communication	Connected module: MELSEC iQ-R series-compatible C24	UNIT_RJ71C24	PROTOCOL_SERIAL	○
	Connected module: Q series-compatible C24	UNIT_QJ71C24		○
	Connected module: L series-compatible C24	UNIT_LJ71C24		○
	Connected module: FX extended port	UNIT_FX485BD		○
Ethernet communication		UNIT_RJ71EN71	PROTOCOL_TCPIP, PROTOCOL_UDPIP	○
		UNIT_RJ71EN71_DIRECT	PROTOCOL_UDPIP	
		UNIT_QJ71E71	PROTOCOL_TCPIP, PROTOCOL_UDPIP	
		UNIT_LJ71E71	PROTOCOL_TCPIP, PROTOCOL_UDPIP	
		UNIT_REETHER	PROTOCOL_TCPIP, PROTOCOL_UDPIP	
		UNIT_REETHER_DIRECT	PROTOCOL_UDPIP	
		UNIT_FXVETHER	PROTOCOL_TCPIP	
		UNIT_FXVETHER_DIRECT	PROTOCOL_UDPIP	
		UNIT_LNETHER	PROTOCOL_TCPIP	
		UNIT_LNETHER, UNIT_LNETHER_DIRECT	PROTOCOL_UDPIP	
		UNIT_QNETHER	PROTOCOL_TCPIP	
		UNIT_QNETHER, UNIT_QNETHER_DIRECT	PROTOCOL_UDPIP	
		UNIT_RJ71GN11	PROTOCOL_TCPIP, PROTOCOL_UDPIP	
		UNIT_NZ2GF_ETB	PROTOCOL_TCPIP	
		UNIT_NZ2GF_ETB, UNIT_NZ2GF_ETB_DIRECT	PROTOCOL_UDPIP	
	UNIT_FXETHER	PROTOCOL_TCPIP		
	UNIT_FXETHER_DIRECT	PROTOCOL_UDPIP		
CPU COM communication	Connection target CPU: FX5CPU	UNIT_FXVCPU	PROTOCOL_SERIAL	○
	Connection target CPU: QCPU (Q mode)	UNIT_QNCPU		○
	Connection target CPU: LCPU	UNIT_LNCPU		○
	Connection target CPU: Q motion CPU	UNIT_QNMOTION		×
	Connection target CPU: FXCPU	UNIT_FXCPU		○
USB communication	Connection target CPU: RCPU	UNIT_RUSB	PROTOCOL_USB	○
	Connection target CPU: LHCPU	UNIT_LHUSB		○
	Connection target CPU: FX5CPU	UNIT_FXVUSB		○
	Connection target CPU: QCPU (Q mode)	UNIT_QNUSB		○
	Connection target CPU: LCPU	UNIT_LNUSB		○
	Connection target CPU: QSCPU	UNIT_QSUSB		○
	Connection target CPU: Q motion CPU	UNIT_QNMOTIONUSB		×
	Connection target CPU: FXCPU	UNIT_FXCPU		○
MELSECNET/H communication	UNIT_MNETHBOARD	PROTOCOL_MNETH	○ ^{*1}	
CC-Link IE Controller Network communication	UNIT_MNETGBOARD	PROTOCOL_MNETG		
CC-Link IE Field Network communication	UNIT_CCIEFBOARD	PROTOCOL_CCIEF		
CC-Link IE TSN communication	UNIT_CCIETSNBOARD	PROTOCOL_CCIETSN		
CC-Link communication	UNIT_CCLINKBOARD	PROTOCOL_CCLINK		

Communication route	Property setting		Applicability
	ActUnitType	ActProtocolType	
CC-Link system RS-232 interface communication	UNIT_G4QCPU	PROTOCOL_SERIAL	○
GX Simulator2 communication	UNIT_SIMULATOR2	—	○
GX Simulator3 communication (RCPU)	UNIT_SIMULATOR3		
GX Simulator3 communication (FX5CPU)			
MT Simulator2 communication	UNIT_SIMULATOR2	—	×
GOT gateway device communication	UNIT_A900GOT	PROTOCOL_UDPIP	×
GOT transparent function communication	UNIT_RJ71C24, UNIT_FXVCPU, UNIT_QNCP, UNIT_LNCP, UNIT_FXCPU, UNIT_QJ71C24, UNIT_LJ71C24, UNIT_GOT_RJ71EN71, UNIT_GOT_QJ71E71, UNIT_GOT_LJ71E71, UNIT_GOT_REETHER, UNIT_GOT_QNETHER, UNIT_GOT_LNETHER, UNIT_A900GOT, UNIT_GOT_NZ2GF_ETB, UNIT_GOT_FXETHER, UNIT_GOT_FXENET	PROTOCOL_SERIAL, PROTOCOL_USBGOT	○
	UNIT_GOTETHER_RJ71C24, UNIT_GOTETHER_FXVCPU, UNIT_GOTETHER_QNCP, UNIT_GOTETHER_LNCP, UNIT_GOTETHER_FXCPU, UNIT_GOTETHER_QBUS, UNIT_GOTETHER_QN_ETHER	PROTOCOL_TCPIP	
Inverter communication	—	COMM_RS232C, COMM_USB	×

*1 An error is returned when the own board is accessed.

SetClockData (Writing clock data)

Applicable controls

This function can be used for the following controls.

- 32-bit: ActProgType, ActUtilType
- 64-bit: ActProgType64, ActUtilType64

Feature

To write time to the clock data of a CPU module.

Format

■VBA

object.SetClockData(iYear, iMonth, iDay, iDayOfWeek, iHour, iMinute, iSecond)

Data type	Argument	Description	Input/Output
Long	IRet	Returned value	Output
Integer	iYear	Year value to be written	Input
Integer	iMonth	Month value to be written	Input
Integer	iDay	Day value to be written	Input
Integer	iDayOfWeek	Day-of-week value to be written	Input
Integer	iHour	Hour value to be written	Input
Integer	iMinute	Minute value to be written	Input
Integer	iSecond	Second value to be written	Input

■Visual C++

IRet = object.SetClockData(sYear, sMonth, sDay, sDayOfWeek, sHour, sMinute, sSecond)

Data type	Argument	Description	Input/Output
Long	IRet	Returned value	Output
Short	sYear	Year value to be written	Input
Short	sMonth	Month value to be written	Input
Short	sDay	Day value to be written	Input
Short	sDayOfWeek	Day-of-week value to be written	Input
Short	sHour	Hour value to be written	Input
Short	sMinute	Minute value to be written	Input
Short	sSecond	Second value to be written	Input

■Visual Basic .NET

IRet = object.SetClockData(iYear, iMonth, iDay, iDayOfWeek, iHour, iMinute, iSecond)

Data type	Argument	Description	Input/Output
Integer	IRet	Returned value	Output
short	iYear	Year value to be written	Input
short	iMonth	Month value to be written	Input
short	iDay	Day value to be written	Input
short	iDayOfWeek	Day-of-week value to be written	Input
short	iHour	Hour value to be written	Input
short	iMinute	Minute value to be written	Input
short	iSecond	Second value to be written	Input

■Visual C#

IRet = object.SetClockData(sYear, sMonth, sDay, sDayOfWeek, sHour, sMinute, sSecond)

Data type	Argument	Description	Input/Output
int	iRet	Returned value	Output
short	sYear	Year value to be written	Input

Data type	Argument	Description	Input/Output
short	sMonth	Month value to be written	Input
short	sDay	Day value to be written	Input
short	sDayOfWeek	Day-of-week value to be written	Input
short	sHour	Hour value to be written	Input
short	sMinute	Minute value to be written	Input
short	sSecond	Second value to be written	Input

Format (when creating an object)

■ Visual C++

hResult = object.SetClockData(sYear,sMonth, sDay, sDayOfWeek, sHour, sMinute, sSecond, *lpRetCode)

Data type	Argument	Description	Input/Output
HRESULT	hResult	Returned value of COM	Output
SHORT	sYear	Year value to be written	Input
SHORT	sMonth	Month value to be written	Input
SHORT	sDay	Day value to be written	Input
SHORT	sDayOfWeek	Day-of-week value to be written	Input
SHORT	sHour	Hour value to be written	Input
SHORT	sMinute	Minute value to be written	Input
SHORT	sSecond	Second value to be written	Input
LONG	*lpRetCode	Returned value of communication function	Output

Returned value

Normal termination: '0' is returned.

Abnormal termination: Any value other than '0' is returned. (🔍 Page 616 ERROR CODES)

Description

- An error is returned when the clock data to be set is not the correct value.
- The applicable values to be specified for iYear (sYear) are: the four digits of year for RCPU and QCPU (Q mode) and the last two digits of year for any other CPUs.
Note that the applicable years for RCPU and QCPU (Q mode) are from 1980 to 2079.
An error occurs when four digits are set to a CPU other than RCPU and QCPU (Q mode).
- The values to be specified in iDayOfWeek (sDayOfWeek) are as follows:

Value	Day of Week
0	Sunday
1	Monday
2	Tuesday
3	Wednesday
4	Thursday
5	Friday
6	Saturday

Point

- An error is returned when an R motion CPU or a Q motion CPU is accessed.
- The clock data can be read on FXCPUs with the built-in clock.
For FXCPUs without the built-in clock, an error is returned.
- Note that an error of transfer time may occur in clock setting.
- Not applicable to inverter communication/robot controller communication.

Applicable communication routes

The following table shows the applicable communication routes.

○: Applicable, ×: Not applicable

Communication route		Property setting		Applicability
		ActUnitType	ActProtocolType	
Serial communication	Connected module: MELSEC iQ-R series-compatible C24	UNIT_RJ71C24	PROTOCOL_SERIAL	○
	Connected module: Q series-compatible C24	UNIT_QJ71C24		○
	Connected module: L series-compatible C24	UNIT_LJ71C24		○
	Connected module: FX extended port	UNIT_FX485BD		○
Ethernet communication		UNIT_RJ71EN71	PROTOCOL_TCPIP, PROTOCOL_UDPIP	○
		UNIT_RJ71EN71_DIRECT	PROTOCOL_UDPIP	
		UNIT_QJ71E71	PROTOCOL_TCPIP, PROTOCOL_UDPIP	
		UNIT_LJ71E71	PROTOCOL_TCPIP, PROTOCOL_UDPIP	
		UNIT_REETHER	PROTOCOL_TCPIP, PROTOCOL_UDPIP	
		UNIT_REETHER_DIRECT	PROTOCOL_UDPIP	
		UNIT_FXVETHER	PROTOCOL_TCPIP	
		UNIT_FXVETHER_DIRECT	PROTOCOL_UDPIP	
		UNIT_LNETHER	PROTOCOL_TCPIP	
		UNIT_LNETHER, UNIT_LNETHER_DIRECT	PROTOCOL_UDPIP	
		UNIT_QNETHER	PROTOCOL_TCPIP	
		UNIT_QNETHER, UNIT_QNETHER_DIRECT	PROTOCOL_UDPIP	
		UNIT_RJ71GN11	PROTOCOL_TCPIP, PROTOCOL_UDPIP	
		UNIT_NZ2GF_ETB	PROTOCOL_TCPIP	
		UNIT_NZ2GF_ETB, UNIT_NZ2GF_ETB_DIRECT	PROTOCOL_UDPIP	
	UNIT_FXETHER	PROTOCOL_TCPIP		
	UNIT_FXETHER_DIRECT	PROTOCOL_UDPIP		
CPU COM communication	Connection target CPU: FX5CPU	UNIT_FXVCPU	PROTOCOL_SERIAL	○
	Connection target CPU: QCPU (Q mode)	UNIT_QNCPU		○
	Connection target CPU: LCPU	UNIT_LNCPUP		○
	Connection target CPU: Q motion CPU	UNIT_QNMOTION		×
	Connection target CPU: FXCPU	UNIT_FXCPU		○
USB communication	Connection target CPU: RCPUP	UNIT_RUSB	PROTOCOL_USB	○
	Connection target CPU: LHCPUP	UNIT_LHUSB		○
	Connection target CPU: FX5CPUP	UNIT_FXVUSB		○
	Connection target CPU: QCPUP (Q mode)	UNIT_QNUSB		○
	Connection target CPU: LCPUP	UNIT_LNUSB		○
	Connection target CPU: QSCPUP	UNIT_QSUSB		× ^{*1}
	Connection target CPU: Q motion CPUP	UNIT_QNMOTIONUSB		×
	Connection target CPU: FXCPUP	UNIT_FXCPU		○
MELSECNET/H communication	UNIT_MNETHBOARD	PROTOCOL_MNETH	○ ^{*2}	
CC-Link IE Controller Network communication	UNIT_MNETGBOARD	PROTOCOL_MNETG		
CC-Link IE Field Network communication	UNIT_CCIEFBOARD	PROTOCOL_CCIEF		
CC-Link IE TSN communication	UNIT_CCIETSNBOARD	PROTOCOL_CCIETSN		
CC-Link communication	UNIT_CCLINKBOARD	PROTOCOL_CCLINK		

Communication route	Property setting		Applicability
	ActUnitType	ActProtocolType	
CC-Link system RS-232 interface communication	UNIT_G4QCPU	PROTOCOL_SERIAL	○
GX Simulator2 communication	UNIT_SIMULATOR2	—	×
GX Simulator3 communication (RCPU)	UNIT_SIMULATOR3		
GX Simulator3 communication (FX5CPU)			
MT Simulator2 communication	UNIT_SIMULATOR2	—	×
GOT gateway device communication	UNIT_A900GOT	PROTOCOL_UDPIP	×
GOT transparent function communication	UNIT_RJ71C24, UNIT_FXVCPU, UNIT_QNCP, UNIT_LNCP, UNIT_FXCPU, UNIT_QJ71C24, UNIT_LJ71C24, UNIT_GOT_RJ71EN71, UNIT_GOT_QJ71E71, UNIT_GOT_LJ71E71, UNIT_GOT_REETHER, UNIT_GOT_QNETHER, UNIT_GOT_LNETHER, UNIT_A900GOT, UNIT_GOT_NZ2GF_ETB, UNIT_GOT_FXETHER, UNIT_GOT_FXENET	PROTOCOL_SERIAL, PROTOCOL_USBGOT	○
	UNIT_GOTETHER_RJ71C24, UNIT_GOTETHER_FXVCPU, UNIT_GOTETHER_QNCP, UNIT_GOTETHER_LNCP, UNIT_GOTETHER_FXCPU, UNIT_GOTETHER_QBUS, UNIT_GOTETHER_QN_ETHER	PROTOCOL_TCPIP	
Inverter communication	—	COMM_RS232C, COMM_USB	×

*1 The error code "0x010A42A0" (the access password mismatch) is returned when accessing QSCPU.

*2 An error is returned when the own board is accessed.

GetCpuType (Reading CPU module model)

Applicable controls

This function can be used for the following controls.

- 32-bit: ActProgType, ActUtilType
- 64-bit: ActProgType64, ActUtilType64

Feature

To read the model character string and the model code (hexadecimal) of CPU module, network board, and GOT.

Format

■VBA

IRet = object.GetCpuType(szCpuName, ICpuType)

Data type	Argument	Description	Input/Output
Long	IRet	Returned value	Output
String	szCpuName	Model character string	Output
Long	ICpuType	Model code	Output

■Visual C++

IRet = object.GetCpuType(*szCpuName, *IpICpuType)

Data type	Argument	Description	Input/Output
Long	IRet	Returned value	Output
BSTR	*szCpuName	Model character string	Output
Long	*IpICpuType	Model code	Output

■Visual Basic .NET

IRet = object.GetCpuType(szCpuName, ICpuType)

Data type	Argument	Description	Input/Output
Integer	IRet	Returned value	Output
String	szCpuName	Model character string	Output
Integer	ICpuType	Model code	Output

■Visual C#

iRet = object.GetCpuType(out szCpuName, out iCpuType)

Data type	Argument	Description	Input/Output
int	iRet	Returned value	Output
String	szCpuName	Model character string	Output
int	iCpuType	Model code	Output

Format (when creating an object)

■Visual C++

hResult = object.GetCpuType(*szCpuName, *IpICpuType, *IpIRetCode)

Data type	Argument	Description	Input/Output
HRESULT	hResult	Returned value of COM	Output
BSTR	*szCpuName	Model character string	Output
LONG	*IpICpuType	Model code	Output
LONG	*IpIRetCode	Returned value of communication function	Output

Returned value

Normal termination: '0' is returned.

Abnormal termination: Any value other than '0' is returned. ( Page 616 ERROR CODES)

Description

- The model character and the model code (hexadecimal) of the communication target CPU module are stored in szCpuName and lCpuType (lplCpuType or iCpuType) respectively.
- The model character string is returned in UNICODE.
- The model code is the value of a module (hexadecimal).

SetCpuStatus (Remote control)

Applicable controls

This function can be used for the following controls.

- 32-bit: ActProgType, ActUtilType
- 64-bit: ActProgType64, ActUtilType64

Feature

To perform a remote operation of CPU module.

Format

■VBA

IRet = object.SetCpuStatus(IOperation)

Data type	Argument	Description	Input/Output
Long	IRet	Returned value	Output
Long	IOperation	Remote RUN/STOP/PAUSE/RESET	Input

■Visual C++

IRet = object.SetCpuStatus(IOperation)

Data type	Argument	Description	Input/Output
Long	IRet	Returned value	Output
Long	IOperation	Remote RUN/STOP/PAUSE/RESET	Input

■Visual Basic .NET

IRet = object.SetCpuStatus(IOperation)

Data type	Argument	Description	Input/Output
Integer	IRet	Returned value	Output
Integer	IOperation	Remote RUN/STOP/PAUSE/RESET	Input

■Visual C#

iRet = object.SetCpuStatus (iOperation)

Data type	Argument	Description	Input/Output
int	iRet	Returned value	Output
int	iOperation	Remote RUN/STOP/PAUSE/RESET	Input

Format (when creating an object)

■Visual C++

hResult = object.SetCpuStatus(IOperation, *lpIRetCode)

Data type	Argument	Description	Input/Output
HRESULT	hResult	Returned value of COM	Output
LONG	IOperation	Remote RUN/STOP/PAUSE/RESET	Input
LONG	*lpIRetCode	Returned value of communication function	Output

Returned value

Normal termination: '0' is returned.

Abnormal termination: Any value other than '0' is returned. ( Page 616 ERROR CODES)

Description

The operation specified for IOperation (iOperation) is performed.

An error occurs when a value other than the following values is specified.

Value	Operation
0	Remote RUN
1	Remote STOP
2	Remote PAUSE
3	Remote RESET

Point

- Since FXCPUs do not feature the PAUSE switch as CPU modules, an error is returned when remote pause is specified in SetCpuStatus.
- An error is returned when a Q motion CPU is accessed and PAUSE is specified.
- An error is returned when an R motion CPU is accessed.
- Not applicable to inverter communication/robot controller communication.
- Remote RESET can be used only for supported models of QCPU, LCPU, RCP, LHCPU, and FX5CPU.

Applicable communication routes

The following table shows the applicable communication routes.

○: Applicable, ×: Not applicable

Communication route		Property setting		Applicability
		ActUnitType	ActProtocolType	
Serial communication	Connected module: MELSEC iQ-R series-compatible C24	UNIT_RJ71C24	PROTOCOL_SERIAL	○
	Connected module: Q series-compatible C24	UNIT_QJ71C24		○
	Connected module: L series-compatible C24	UNIT_LJ71C24		○
	Connected module: FX extended port	UNIT_FX485BD		○**2
Ethernet communication		UNIT_RJ71EN71	PROTOCOL_TCPIP, PROTOCOL_UDPIP	○
		UNIT_RJ71EN71_DIRECT	PROTOCOL_UDPIP	
		UNIT_QJ71E71	PROTOCOL_TCPIP, PROTOCOL_UDPIP	○
		UNIT_LJ71E71	PROTOCOL_TCPIP, PROTOCOL_UDPIP	
		UNIT_REETHER	PROTOCOL_TCPIP, PROTOCOL_UDPIP	○*1
		UNIT_REETHER_DIRECT	PROTOCOL_UDPIP	
		UNIT_FXVETHER	PROTOCOL_TCPIP	○*1
		UNIT_FXVETHER_DIRECT	PROTOCOL_UDPIP	
		UNIT_LNETHER	PROTOCOL_TCPIP	○
		UNIT_LNETHER, UNIT_LNETHER_DIRECT	PROTOCOL_UDPIP	○
		UNIT_QNETHER	PROTOCOL_TCPIP	
		UNIT_QNETHER, UNIT_QNETHER_DIRECT	PROTOCOL_UDPIP	
		UNIT_RJ71GN11	PROTOCOL_TCPIP, PROTOCOL_UDPIP	○
		UNIT_NZ2GF_ETB	PROTOCOL_TCPIP	
		UNIT_NZ2GF_ETB, UNIT_NZ2GF_ETB_DIRECT	PROTOCOL_UDPIP	○*1
		UNIT_FXETHER	PROTOCOL_TCPIP	
	UNIT_FXETHER_DIRECT	PROTOCOL_UDPIP		

Communication route		Property setting		Appli cabil ity
		ActUnitType	ActProtocolType	
CPU COM communication	Connection target CPU: FX5CPU	UNIT_FXVCPU	PROTOCOL_SERIAL	○*1
	Connection target CPU: QCPU (Q mode)	UNIT_QNCPU		○
	Connection target CPU: LCPU	UNIT_LNCPUCPU		○
	Connection target CPU: Q motion CPU	UNIT_QNMOTION		○
	Connection target CPU: FXCPU	UNIT_FXCPU		○*1
USB communication	Connection target CPU: RCPU	UNIT_RUSB	PROTOCOL_USB	○
	Connection target CPU: LHCPU	UNIT_LHUSB		○
	Connection target CPU: FX5CPU	UNIT_FXVUSB		○
	Connection target CPU: QCPU (Q mode)	UNIT_QNUSB		○
	Connection target CPU: LCPU	UNIT_LNUSB		○
	Connection target CPU: QSCPU	UNIT_QSUSB		×*3
	Connection target CPU: Q motion CPU	UNIT_QNMOTIONUSB		○
	Connection target CPU: FXCPU	UNIT_FXCPU		○*1
MELSECNET/H communication	UNIT_MNETHBOARD	PROTOCOL_MNETH	○*4	
CC-Link IE Controller Network communication	UNIT_MNETGBOARD	PROTOCOL_MNETG		
CC-Link IE Field Network communication	UNIT_CCIEFBOARD	PROTOCOL_CCIEF		
CC-Link IE TSN communication	UNIT_CCIETSNBOARD	PROTOCOL_CCIETSN		
CC-Link communication	UNIT_CCLINKBOARD	PROTOCOL_CCLINK		
CC-Link system RS-232 interface communication	UNIT_G4QCPU	PROTOCOL_SERIAL	○	
GX Simulator2 communication	UNIT_SIMULATOR2	—	○	
GX Simulator3 communication (RCPU)	UNIT_SIMULATOR3			
GX Simulator3 communication (FX5CPU)			○*1	
MT Simulator2 communication	UNIT_SIMULATOR2	—	○	
GOT gateway device communication	UNIT_A900GOT	PROTOCOL_UDPIP	×	
GOT transparent function communication	UNIT_RJ71C24, UNIT_QNCPU, UNIT_LNCPUCPU	PROTOCOL_SERIAL, PROTOCOL_USBGOT	○	
	UNIT_FXVCPU, UNIT_FXCPU		○*1	
	UNIT_QJ71C24, UNIT_LJ71C24, UNIT_GOT_RJ71E71, UNIT_GOT_QJ71E71, UNIT_GOT_LJ71E71, UNIT_GOT_RETHER, UNIT_GOT_QNETHER, UNIT_GOT_LNETHER, UNIT_A900GOT, UNIT_GOT_NZ2GF_ETB		○	
	UNIT_GOT_FXETHER, UNIT_GOT_FXENET		○*1	
	UNIT_GOTETHER_RJ71C24, UNIT_GOTETHER_QNCPU, UNIT_GOTETHER_LNCPUCPU	PROTOCOL_TCPIP	○	
	UNIT_GOTETHER_FXVCPU, UNIT_GOTETHER_FXCPU		○*1	
	UNIT_GOTETHER_QBUS		○	
	UNIT_GOTETHER_QN_ETHER		○	
Inverter communication	—	COMM_RS232C, COMM_USB	×	

*1 An error is returned when PAUSE is specified.

*2 An error is returned when the CPU is other than FX3S, FX3U, FX3UC, FX3G and FX3GC.

*3 The error code "0x010A42A0" (the access password mismatch) is returned when accessing QSCPU.

*4 An error is returned when the own board is accessed.

EntryDeviceStatus (Registering devices for status monitoring)

Applicable controls

This function can be used for the following controls.

- 32-bit: ActProgType, ActUtilType

Feature

To register devices whose status to be monitored.

Format

■VBA

IRet = object.EntryDeviceStatus(szDeviceList, ISize, IMonitorCycle, IData(0))

Data type	Argument	Description	Input/Output
Long	IRet	Returned value	Output
String	szDeviceList	Registered device name list	Input
Long	ISize	Number of registered device points	Input
Long	IMonitorCycle	Status monitoring time interval	Input
Long	IData(n)	Registered device value list	Input

■Visual C++

IRet = object.EntryDeviceStatus (szDeviceList, ISize, IMonitorCycle, *lpIData)

Data type	Argument	Description	Input/Output
Long	IRet	Returned value	Output
CString	szDeviceList	Registered device name list	Input
Long	ISize	Number of registered device points	Input
Long	IMonitorCycle	Status monitoring time interval	Input
Long	*lpIData	Registered device value list	Input

■Visual Basic .NET

IRet = object.EntryDeviceStatus(szDeviceList, ISize,IMonitorCycle, IData(0))

Data type	Argument	Description	Input/Output
Integer	IRet	Returned value	Output
String	szDeviceList	Registered device name list	Input
Integer	ISize	Number of registered device points	Input
Integer	IMonitorCycle	Status monitoring time interval	Input
Integer	IData(n)	Registered device value list	Input

■Visual C#

iRet = object. EntryDeviceStatus(szDeviceList, iSize, iMonitorCycle, ref iData[0])

Data type	Argument	Description	Input/Output
int	iRet	Returned value	Output
String	szDeviceList	Registered device name list	Input
int	iSize	Number of registered device points	Input
int	iMonitorCycle	Status monitoring time interval	Input
int[n]	iData	Registered device value list	Input

Returned value

Normal termination: '0' is returned.

Abnormal termination: Any value other than '0' is returned. ( Page 616 ERROR CODES)

Description

- A device group for the size of ISize (iSize) specified for szDeviceList is checked whether it is in the status specified for IData (IpIData or iData).

Specify the check time for IMonitorCycle (iMonitorCycle).

When the status is established, the OnDeviceStatus function of the user program is executed.

- Use the line feed symbol to separate the character string specified for the device list. The line feed symbol is not necessary to be suffixed to the last device.

Example:

When using Visual Basic .NET, VBA: "D0" & vbLf & "D1" & vbLf & "D2"

When using Visual C++: D0\nD1\nD2

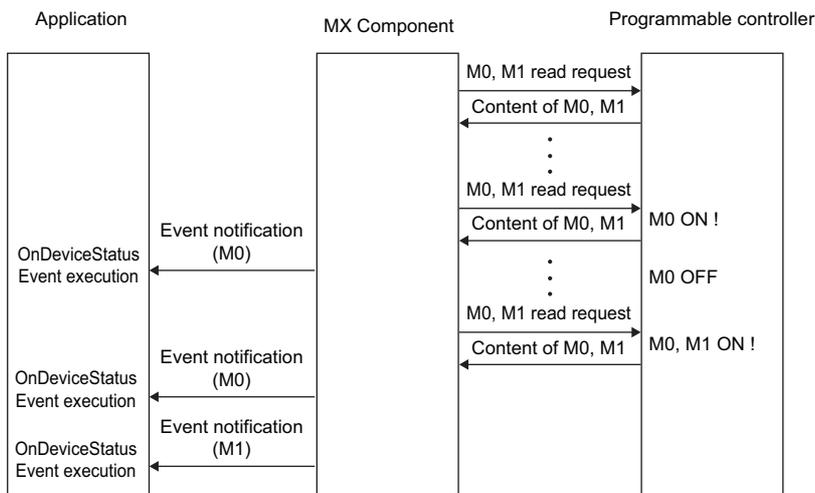
When using Visual C#: D0\nD1\nD2

- The maximum number of device points for ISize (iSize) is 20.
- Specify a value within the range from 1 second to 1 hour (set between 1 and 3600 in seconds) for IMonitorCycle (iMonitorCycle).

An error occurs when any other value outside the above range is specified.

- The registered device value list is stored in IData (IpIData or iData).

- Device status monitoring may not be performed at the specified status monitoring time intervals depending on the conditions: personal computer performance, currently executed application load, time required for communication with the programmable controller, or the like.
Simultaneous use of any other control functions would also be the cause of disabling device status monitoring at the specified status monitoring time intervals.
- For IData (IplData or iData), reserve a memory area for the number of points specified for ISize (iSize). If the memory area is not reserved, a critical error (an application error or the like) may occur.
- An error occurs when the EntryDeviceStatus function is executed during a status monitoring.
When changing any status monitor condition, execute the FreeDeviceStatus function and then execute the EntryDeviceStatus function.
- When the status of multiple devices changes at the same time, the OnDeviceStatus event is executed every time the status changes.
(Example: When M0 is monitored)



- This function is a function to check the status establishment under the constant execution of random device read by the control.
This function is not a function for a CPU module to notify the device status establishment to MX Component. Therefore, the control may not be able to check the device status establishment of CPU module depending on the specified status monitoring time interval.
- Not applicable to inverter communication/robot controller communication.

Considerations for checking word device status

When checking the word device status for negative values of -1 to -32768 (FFFFH to 8000H), set the monitor device value of the EntryDeviceStatus function to any of 65535 to 32768 (0000FFFFH to 00008000H) where '0' is stored in the upper 2-byte.

Example: When checking the D0 status for "-10"

Set the value "65526 (0000FFF6H)" where '0' is stored in the upper 2-byte of "-10 (FFFFFFF6H)" for the monitor device value. While the type of word devices of the CPU module is WORD type, the type of monitor device value of the EntryDeviceStatus function is LONG type.

Therefore, when current values of CPU module are compared with monitor device values of the EntryDeviceStatus function, the values do not match and the above setting is required. (When bit devices or double word devices are used, this consideration does not apply.)

For the programming examples regarding this consideration, refer to the following:

 Page 670 Programing Examples for Monitoring Word Device Status

How to specify devices

The following describes how to specify devices.

■When devices are specified as follows (number of points: 3 points)

When using Visual Basic .NET, VBA: "M0" & vbCrLf & "D0" & vbCrLf & "K8M0"
 When using Visual C++: M0\nD0\nK8M0
 When using Visual C#: M0\nD0\nK8M0

Upper 2-byte	Lower 2-byte
Not used ('0' is stored.)	M0 D0
M16 to M31*1	M0 to M15*1

*1 Devices are stored from the lower bit in the order of device number.

■When devices including CN200 and later of FXCPU are specified (total number of points: 3 points)

For CN200 and later devices of FXCPU, 2 words are read for each point when reading devices randomly.

When using Visual Basic .NET, VBA: "D0" & vbCrLf & "CN200" & vbCrLf & "D1"
 When using Visual C++: D0\nCN200\nD1
 When using Visual C#: D0\nCN200\nD1

Upper 2-byte	Lower 2-byte
Not used ('0' is stored.)	D0
H of CN200	L of CN200
Not used ('0' is stored.)	D1

■When devices including FD are specified (total number of points: 3 points)

When using Visual Basic .NET, VBA: "D0" & vbCrLf & "FD0" & vbCrLf & "D1"
 When using Visual C++: D0\nFD0\nD1
 When using Visual C#: D0\nFD0\nD1

Upper 2-byte	Lower 2-byte
Not used ('0' is stored.)	D0 LL of FD0 D1

■When devices including EG are specified (total number of points: 3 points)

The following is the example when 8-bit devices (E0000 and E0001 of SHARP programmable controller) are assigned to EG0.

When using Visual Basic .NET, VBA: "D0" & vbLf & "EG0" & vbLf & "D1"
 When using Visual C++: D0\nEG0\nD1
 When using Visual C#: D0\nEG0\nD1

Upper 2-byte	Lower 2-byte	
Not used ('0' is stored.)	D0	
	EG0	
	(E0001)	(E0000)
	D1	

■When double word devices are specified

When using Visual Basic .NET, VBA: "LTN0" & vbLf & "LTN1" & vbLf & "LTN2"
 When using Visual C++: LTN0\nLTN1\nLTN2
 When using Visual C#: LTN0\nLTN1\nLTN2

Upper 2-byte	Lower 2-byte
H of LTN0	L of LTN0
H of LTN1	L of LTN1
H of LTN2	L of LTN2

FreeDeviceStatus (Deregistering devices for status monitoring)

Applicable controls

This function can be used for the following controls.

- 32-bit: ActProgType, ActUtilType

Feature

To deregister devices registered by using the EntryDeviceStatus function to monitor their status.

Format

■VBA

IRet = object.FreeDeviceStatus()

Data type	Argument	Description	Input/Output
Long	IRet	Returned value	Output

■Visual C++

iRet = object.FreeDeviceStatus()

Data type	Argument	Description	Input/Output
int	iRet	Returned value	Output

■Visual Basic .NET

IRet = object.FreeDeviceStatus()

Data type	Argument	Description	Input/Output
Integer	IRet	Returned value	Output

■Visual C#

iRet = object.FreeDeviceStatus()

Data type	Argument	Description	Input/Output
int	iRet	Returned value	Output

Returned value

Normal termination: '0' is returned.

Abnormal termination: Any value other than '0' is returned. ([↩ Page 616 ERROR CODES](#))

Description

The devices that are set using the EntryDeviceStatus function to monitor their status are deregistered.

Point

Not applicable to inverter communication/robot controller communication.

OnDeviceStatus (Event notification)

Applicable controls

This function can be used for the following controls.

- 32-bit: ActProgType, ActUtilType

Feature

To execute the event notification when the device condition registered by using the EntryDeviceStatus function is satisfied.

Format (when creating an object)

■VBA

- Adding an event handler

```
Private WithEvents object As objectLib.object
```

- Event handler

```
Private Sub object_OnDeviceStatus(szDevice, IData, IReturnCode)
```

Data type	Argument	Description	Input/Output
String	szDevice	Name of device whose condition is satisfied	Input
Long	IData	Value of device whose condition is satisfied	Input
Long	IReturnCode	Returned value of condition check processing	Input

■Visual Basic .NET

- Adding an event handler

```
AddHandler object.OnDeviceStatus, AddressOf object_OnDeviceStatus
```

- Event handler

```
Private Sub object_OnDeviceStatus(szDevice, IData, IReturnCode)
```

Data type	Argument	Description	Input/Output
String	szDevice	Name of device whose condition is satisfied	Input
Integer	IData	Value of device whose condition is satisfied	Input
Integer	IReturnCode	Returned value of condition check processing	Input

- Removing an event handler

```
RemoveHandler object.OnDeviceStatus, AddressOf object_OnDeviceStatus
```

■Visual C#

- Adding an event handler

```
object.OnDeviceStatus += new objectLib._IobjectEvents_OnDeviceStatusEventHandler(object_OnDeviceStatus);
```

- Event handler

```
private void object_OnDeviceStatus(szDevice, IData, IReturnCode)
```

Data type	Argument	Description	Input/Output
String	szDevice	Name of device whose condition is satisfied	Input
int	IData	Value of device whose condition is satisfied	Input
int	IReturnCode	Returned value of condition check processing	Input

- Removing an event handler

```
object.OnDeviceStatus -= new objectLib._IobjectEvents_OnDeviceStatusEventHandler(object_OnDeviceStatus);
```

Format (when pasting a control to a form)

■VBA

Private Sub object.OnDeviceStatus(szDevice, IData, IReturnCode)

Data type	Argument	Description	Input/Output
String	szDevice	Name of device whose condition is satisfied	Input
Long	IData	Value of device whose condition is satisfied	Input
Long	IReturnCode	Returned value of condition check processing	Input

■Visual C++

- Adding an event handler

Add "OnDeviceStatus" event from "Properties" of a control pasted to a form ⇨ [Control Events] tab.

OnDeviceStatus event name: object_OnDeviceStatus

- Event handler

void object.OnDeviceStatus(szDevice, IData, IReturnCode)

Data type	Argument	Description	Input/Output
LPCTSTR	szDevice	Name of device whose condition is satisfied	Input
Long	IData	Value of device whose condition is satisfied	Input
Long	IReturnCode	Returned value of condition check processing	Input

■Visual Basic .NET

Private Sub object_OnDeviceStatus(ByVal sender As System.Object,
ByVal e As ObjectLib. _objectEvents_OnDeviceStatusEvent)

Argument	Description	Input/Output
sender	Event occurrence source	Output
e	Event data	Output

The members of e are as follows:

Member	Description
e.szDevice	Name of device whose condition is satisfied
e.IReturnCode	Value of device whose condition is satisfied
e.IData	Returned value of condition check processing

■Visual C#

private void object_OnDeviceStatus
(object sender,object._IActUtilTypeEvents_OnDeviceStatusEvent e)

Argument	Description	Input/Output
sender	Event occurrence source	Output
e	Event data	Output

The members of e are as follows:

Member	Description
e->szDevice	Name of device whose condition is satisfied
e->IReturnCode	Returned value of condition check processing
e->IData	Value of device whose condition is satisfied

Returned value

None

Description

- The event is notified to the application when the device condition registered using the EntryDeviceStatus function is satisfied.
Programming this function in the user program allows the application to receive the event when the registered device condition is satisfied.
- Device values registered by using the EntryDeviceStatus function are input to IData.
Example: When the word device is monitored for the value of "-1"
Set 65535 (0000FFFFH) as a registered device value by using the EntryDeviceStatus function.
When the value of the target word device of the CPU module becomes "-1" (FFFFH), the OnDeviceStatus function is executed and (0000FFFFH) is input to IData.

Point

- When any of the following settings is set in the user program, the event of the OnDeviceStatus function does not occur even if the condition of the device registered to the EntryDeviceStatus function is satisfied.
Note that when the event occurrence becomes in wait status, the control function is not returned to the control and device management processing stops until the following setting is terminated:
 - (1) For user programs created by using Visual Basic .NET or VBA (Excel):
 - The message box is being displayed in the user program.
 - The InputBox/OutputBox is being displayed in the user program.
 - (2) For user programs created by using Visual Basic .NET, Visual C++, Visual C#, or VBA (Excel, Access):
 - The Sleep processing, WaitForSingleObject function, or similar standby function is being used in the user program.
- For installing "References," refer to the sample programs.
- The method to paste a control to a form cannot be used for 64-bit.
- Not applicable to inverter communication/robot controller communication.

ReadDeviceBlock2 (Reading devices in batch)

Applicable controls

This function can be used for the following controls.

- 32-bit: ActProgType, ActUtilType
- 64-bit: ActProgType64, ActUtilType64

Feature

To read devices in 2-byte data unit in batch.

Format

■VBA

IRet = object.ReadDeviceBlock2(szDevice, ISize, iData(0))

Data type	Argument	Description	Input/Output
Long	IRet	Returned value	Output
String	szDevice	Device name	Input
Long	ISize	Number of read points	Input
Integer	iData(n)	Read device value	Output

■Visual C++

IRet = object.ReadDeviceBlock2(szDevice, ISize, *lpsData)

Data type	Argument	Description	Input/Output
Long	IRet	Returned value	Output
CString	szDevice	Device name	Input
Long	ISize	Number of read points	Input
Short	*lpsData	Read device value	Output

■Visual Basic .NET

IRet = object.ReadDeviceBlock2(szDevice, ISize, sData(0))

Data type	Argument	Description	Input/Output
Integer	IRet	Returned value	Output
String	szDevice	Device name	Input
Integer	ISize	Number of read points	Input
short	sData(n)	Read device value	Output

■Visual C#

iRet = object.ReadDeviceBlock2(szDevice, iSize, out sData[0])

Data type	Argument	Description	Input/Output
int	iRet	Returned value	Output
String	szDevice	Device name	Input
int	iSize	Number of read points	Input
short[n]	sData	Read device value	Output

Format (when creating an object)

■ Visual C++

hResult = object.ReadDeviceBlock2(szDevice, ISize, *IpsData, *IplRetCode)

Data type	Argument	Description	Input/Output
HRESULT	hResult	Returned value of COM	Output
BSTR	szDevice	Device name	Input
LONG	ISize	Number of read points	Input
SHORT	*IpsData	Read device value	Output
LONG	*IplRetCode	Returned value of communication function	Output

Returned value

Normal termination: '0' is returned.

Abnormal termination: Any value other than '0' is returned. (🔗 Page 616 ERROR CODES)

Description

- The device values of ISize (iSize) are read in batch starting from the device specified for szDevice as SHORT type data.
- The read device values are stored in iData (IpsData or sData).
- For iData (IpsData or sData), reserve arrays for more than the amount specified for ISize (iSize).

Point

- The maximum number of read points for ISize should be the value which satisfies the following condition:
Start read device number + Number of read points \leq Last device number
- When specifying bit devices, a multiple of 16 can be specified as a device number.
- For iData (IpsData or sData), reserve a memory area for the number of points specified for ISize (iSize).
If the memory area is not reserved, a critical error (an application error or the like) may occur.
- Devices of long timer contact (LTS), long timer coil (LTC), long retentive timer contact (LSTS), long retentive timer coil (LSTC) cannot be read. Use the ReadDeviceRandom2 or GetDevice2 function.
- Not applicable to inverter communication/robot controller communication.

How to specify devices

The following describes how to specify devices.

■ When specifying bit devices

Example: Read 3 points (3 words) of data in 16-point unit from the devices starting from M0.

2-byte

M0 to M15^{*1}

M16 to M31^{*1}

M32 to M47^{*1}

*1 Devices are stored from the lower bit in the order of device number.

■ When specifying word devices

Example: Read 3 points of data from the devices starting from D0.

2-byte

D0

D1

D2

■When specifying CN200 and later devices of FXCPU

Example: Read 6 points of data from the devices starting from CN200.

For CN200 and later devices of FXCPU, 2 words are read for each 2 points. Reading only 1 point of data will result in an error.

2-byte	
L of CN200	
H of CN200	
L of CN201	
H of CN201	
L of CN202	
H of CN202	

■When specifying FD devices (4-word devices)

Example: Read 6 points of data from the devices starting from FD0.

2-byte	
LL of FD0	
LH of FD0	
HL of FD0	
HH of FD0	
LL of FD1	
LH of FD1	

■8-bit devices assigned to gateway devices

Example: Read 4 points of data from the devices starting from EG0.

(When 8-bit devices (from E0000 to E0007 of SHARP programmable controller) are assigned from EG0 to EG3.)

2-byte	
EG0	
(E0001)	(E0000)
EG1	
(E0003)	(E0002)
EG2	
(E0005)	(E0004)
EG3	
(E0007)	(E0006)

■When specifying double word devices

Example: Read 3 points of data from the devices starting from LTN0.

2-byte	
L of LTN0	
L of LTN1	
L of LTN2	

WriteDeviceBlock2 (Writing devices in batch)

Applicable controls

This function can be used for the following controls.

- 32-bit: ActProgType, ActUtilType
- 64-bit: ActProgType64, ActUtilType64

Feature

To write devices in 2-byte data unit in batch.

Format

■VBA

IRet = object.WriteDeviceBlock2(szDevice, ISize, iData(0))

Data type	Argument	Description	Input/Output
Long	IRet	Returned value	Output
String	szDevice	Device name	Input
Long	ISize	Number of write points	Input
Integer	iData(n)	Device value to be written	Input

■Visual C++

IRet = object.WriteDeviceBlock2(szDevice, ISize, *lpsData)

Data type	Argument	Description	Input/Output
Long	IRet	Returned value	Output
CString	szDevice	Device name	Input
Long	ISize	Number of write points	Input
Short	*lpsData	Device value to be written	Input

■Visual Basic .NET

IRet = object.WriteDeviceBlock2(szDevice, ISize, sData(0))

Data type	Argument	Description	Input/Output
Integer	IRet	Returned value	Output
String	szDevice	Device name	Input
Integer	ISize	Number of write points	Input
short	sData(n)	Device value to be written	Input

■Visual C#

iRet = object.WriteDeviceBlock2(szDevice, iSize, ref sData[0])

Data type	Argument	Description	Input/Output
int	iRet	Returned value	Output
String	szDevice	Device name	Input
int	iSize	Number of write points	Input
short[n]	sData	Device value to be written	Input

Format (when creating an object)

■ Visual C++

hResult = object.WriteDeviceBlock2(szDevice, ISize, *IpsData, *IplRetCode)

Data type	Argument	Description	Input/Output
HRESULT	hResult	Returned value of COM	Output
BSTR	szDevice	Device name	Input
LONG	ISize	Number of write points	Input
SHORT	*IpsData	Device value to be written	Input
LONG	*IplRetCode	Returned value of communication function	Output

Returned value

Normal termination: '0' is returned.

Abnormal termination: Any value other than '0' is returned. (🔗 Page 616 ERROR CODES)

Description

- The device values of ISize (iSize) are written in batch starting from the device specified for szDevice.
- The device values to be written are stored in iData (IpsData or sData).
- For iData (IpsData or sData), reserve arrays for more than the amount specified for ISize (iSize).

Point

- The maximum number of write points for ISize (iSize) should be the value which satisfies the following condition:
Start write device number + Number of write points ≤ Last device number
- When specifying bit devices, a multiple of 16 can be specified as a device number.
- For iData (IpsData or sData), reserve a memory area for the number of points specified for ISize (iSize).
If the memory area is not reserved, a critical error (an application error or the like) may occur.
- Devices of long timer contact (LTS), long timer coil (LTC), long retentive timer contact (LSTS), long retentive timer coil (LSTC) cannot be written. Use the WriteDeviceRandom2 or SetDevice2 function.
- If the function is run against the safety device in the RnSFPCPU safety mode, an error code 0x010A42A5" (an operation that can not be carried out in safety mode was performed) is returned.
- Not applicable to inverter communication/robot controller communication.

How to specify devices

The following describes how to specify devices.

■ When specifying bit devices

Example: Write 3 points (3 words) of data in 16-point unit to the devices starting from M0.

2-byte

M0 to M15^{*1}

M16 to M31^{*1}

M32 to M47^{*1}

*1 Devices are stored from the lower bit in the order of device number.

■ When specifying word devices

Example: Write 3 points of data to the devices starting from D0.

2-byte

D0

D1

D2

■When specifying CN200 and later devices of FXCPU

Example: Write 6 points of data to the devices starting from CN200.

For CN200 and later devices of FXCPU, 2 words are written for each 2 points. Writing only 1 point of data will result in an error.

2-byte	
L of CN200	
H of CN200	
L of CN201	
H of CN201	
L of CN202	
H of CN202	

■When specifying FD devices (4-word devices)

Example: Write 6 points of data to the devices starting from FD0.

2-byte	
LL of FD0	
LH of FD0	
HL of FD0	
HH of FD0	
LL of FD1	
LH of FD1	

■8-bit devices assigned to gateway devices

Example: Write 4 points of data to the devices starting from EG0.

(When 8-bit devices (from E0000 to E0007 of SHARP programmable controller) are assigned from EG0 to EG3.)

2-byte	
EG0	
(E0001)	(E0000)
EG1	
(E0003)	(E0002)
EG2	
(E0005)	(E0004)
EG3	
(E0007)	(E0006)

■When specifying double word devices

Example: Write 3 points of data to the devices starting from LTN0.

2-byte	
L of LTN0	
L of LTN1	
L of LTN2	

ReadDeviceRandom2 (Reading devices randomly)

Applicable controls

This function can be used for the following controls.

- 32-bit: ActProgType, ActUtilType
- 64-bit: ActProgType64, ActUtilType64

Feature

To read devices in 2-byte data unit randomly.

Format

■VBA

IRet = object.ReadDeviceRandom2(szDeviceList, ISize, iData(0))

Data type	Argument	Description	Input/Output
Long	IRet	Returned value	Output
String	szDeviceList	Device name	Input
Long	ISize	Number of read points	Input
Integer	iData(n)	Read device value	Output

■Visual C++

IRet = object.ReadDeviceRandom2(szDeviceList, ISize, *lpsData)

Data type	Argument	Description	Input/Output
Long	IRet	Returned value	Output
CString	szDeviceList	Device name	Input
Long	ISize	Number of read points	Input
Short	*lpsData	Read device value	Output

■Visual Basic .NET

IRet = object.ReadDeviceRandom2(szDeviceList, ISize, sData(0))

Data type	Argument	Description	Input/Output
Integer	IRet	Returned value	Output
String	szDeviceList	Device name	Input
Integer	ISize	Number of read points	Input
short	sData(n)	Read device value	Output

■Visual C#

iRet = object.ReadDeviceRandom2(szDevice, iSize, out sData[0])

Data type	Argument	Description	Input/Output
int	iRet	Returned value	Output
String	szDevice	Device name	Input
int	iSize	Number of read points	Input
short[n]	sData	Read device value	Output

Format (when creating an object)

■ Visual C++

hResult = object.ReadDeviceRandom2(szDeviceList, ISize, *IpsData, *IplRetCode)

Data type	Argument	Description	Input/Output
HRESULT	hResult	Returned value of COM	Output
BSTR	szDeviceList	Device name	Input
LONG	ISize	Number of read points	Input
SHORT	*IpsData	Read device value	Output
LONG	*IplRetCode	Returned value of communication function	Output

Returned value

Normal termination: '0' is returned.

Abnormal termination: Any value other than '0' is returned. (🔗 Page 616 ERROR CODES)

Description

- Data of a device group specified for szDeviceList (szDevice) is read for the size of ISize (iSize).
- The read device values are stored in iData (IpsData or sData).
- Use the line feed symbol to separate the character string specified for the device list. The line feed symbol is not necessary to be suffixed to the last device.

Example:

When using Visual Basic .NET, VBA: "D0" & vbLf & "D1" & vbLf & "D2"
 When using Visual C++: D0\nD1\nD2
 When using Visual C#: D0\nD1\nD2

- For iData (IpsData or sData), reserve arrays for more than the amount specified for ISize (iSize).

Point

- The maximum number of read points for ISize (iSize) is 0x7FFFFFFF.
- For iData (IpsData or sData), reserve a memory area for the number of points specified for ISize (iSize). If the memory area is not reserved, a critical error (an application error or the like) may occur.
- When specifying a double word device, the ReadDeviceRandom2 function stores only the data of lower 1 word (2-byte). (An error does not occur.)
 Use the ReadDeviceRandom or GetDevice function to read data from double word devices.
- Not applicable to inverter communication/robot controller communication.

How to specify devices

The following describes how to specify devices.

■ When devices are specified as follows (number of points: 3 points)

When using Visual Basic .NET, VBA: "M0" & vbLf & "D0" & vbLf & "K8M0"
 When using Visual C++: M0\nD0\nK8M0
 When using Visual C#: M0\nD0\nK8M0

2-byte

M0^{*1}

D0

M0 to M15^{*2}

*1 The target device from which data is read is only one point of "M0," and '0' or '1' is stored as the device value.

*2 Devices are stored from the lower bit in the order of device number.

Data is not read from the upper 2-byte of M16 to M31.

■When devices including CN200 and later of FXCPU are specified (total number of points: 3 points)

When using Visual Basic .NET, VBA: "D0" & vbLf & "CN200" & vbLf & "D1"
 When using Visual C++: D0\nCN200\nD1
 When using Visual C#: D0\nCN200\nD1

2-byte	
D0	
L of CN200*1	
D1	

*1 For CN200 and later devices of FXCPU, the L (lower 2-byte) of specified devices is read for each point when the ReadDeviceRandom2 function is executed.
 The H (upper 2-byte) of specified devices is not read.

■When devices including FD are specified (total number of points: 3 points)

When using Visual Basic .NET, VBA: "D0" & vbLf & "FD0" & vbLf & "D1"
 When using Visual C++: D0\nFD0\nD1
 When using Visual C#: D0\nFD0\nD1

2-byte	
D0	
LL of FD0	
D1	

■When devices including EG are specified (total number of points: 3 points)

The following is the example when 8-bit devices (E0000 and E0001 of SHARP programmable controller) are assigned to EG0.

When using Visual Basic .NET, VBA: "D0" & vbLf & "EG0" & vbLf & "D1"
 When using Visual C++: D0\nEG0\nD1
 When using Visual C#: D0\nEG0\nD1

2-byte	
D0	
EG0	
(E0001)	(E0000)
D1	

■When double word devices are specified

When using Visual Basic .NET, VBA: "LTN0" & vbLf & "LTN1" & vbLf & "LTN2"
 When using Visual C++: LTN0\nLTN1\nLTN2
 When using Visual C#: LTN0\nLTN1\nLTN2

2-byte	
L of LTN0	
L of LTN1	
L of LTN2	

WriteDeviceRandom2 (Writing devices randomly)

Applicable controls

This function can be used for the following controls.

- 32-bit: ActProgType, ActUtilType
- 64-bit: ActProgType64, ActUtilType64

Feature

To write devices in 2-byte data unit randomly.

Format

■VBA

IRet = object.WriteDeviceRandom2(szDeviceList, ISize,iData(0))

Data type	Argument	Description	Input/Output
Long	IRet	Returned value	Output
String	szDeviceList	Device name	Input
Long	ISize	Number of write points	Input
Integer	iData(n)	Device value to be written	Input

■Visual C++

IRet = object.WriteDeviceRandom2(szDeviceList, ISize,*IpsData)

Data type	Argument	Description	Input/Output
Long	IRet	Returned value	Output
CString	szDeviceList	Device name	Input
Long	ISize	Number of write points	Input
Short	*IpsData	Device value to be written	Input

■Visual Basic .NET

IRet = object.WriteDeviceRandom2(szDeviceList, ISize, sData(0))

Data type	Argument	Description	Input/Output
Integer	IRet	Returned value	Output
String	szDeviceList	Device name	Input
Integer	ISize	Number of write points	Input
short	sData(n)	Device value to be written	Input

■Visual C#

iRet = object.WriteDeviceRandom2(szDevice, iSize, ref sData[0])

Data type	Argument	Description	Input/Output
int	iRet	Returned value	Output
String	szDevice	Device name	Input
int	iSize	Number of write points	Input
short[n]	sData	Device value to be written	Input

Format (when creating an object)

■ Visual C++

hResult = object.WriteDeviceRandom2(szDeviceList, ISize, *lpsData, *lplRetCode)

Data type	Argument	Description	Input/Output
HRESULT	hResult	Returned value of COM	Output
BSTR	szDeviceList	Device name	Input
LONG	ISize	Number of write points	Input
SHORT	*lpsData	Device value to be written	Input
LONG	*lplRetCode	Returned value of communication function	Output

Returned value

Normal termination: '0' is returned.

Abnormal termination: Any value other than '0' is returned. (🔗 Page 616 ERROR CODES)

Description

- Data of a device group specified for szDeviceList (szDevice) is written for the size of ISize (iSize).
- The device values to be written are stored in iData (lpsData or sData).
- Use the line feed symbol to separate the character string specified for the device list. The line feed symbol is not necessary to be suffixed to the last device.

Example:

When using Visual Basic .NET, VBA: "D0" & vbLf & "D1" & vbLf & "D2"

When using Visual C++: D0\nD1\nD2

When using Visual C#: D0\nD1\nD2

- For iData (lpsData or sData), reserve arrays for more than the amount specified for ISize (iSize).

Point

- The maximum number of write points for ISize (iSize) is 0x7FFFFFFF.
- For iData (lpsData or sData), reserve a memory area for the number of points specified for ISize (iSize). If the memory area is not reserved, a critical error (an application error or the like) may occur.
- When specifying a double word device, the WriteDeviceRandom2 function writes to the data area of lower 1 word (2-byte), and '0' is written to the data area of upper 1 word (2-byte).
When writing a double word device, use the WriteDeviceRandom function or the SetDevice function.
- An error is returned when a Q motion CPU is accessed.
- If the function is run against the safety device in the RnSF CPU safety mode, an error code 0x010A42A5" (an operation that can not be carried out in safety mode was performed) is returned.
- Not applicable to inverter communication/robot controller communication.

How to specify devices

The following describes how to specify devices.

■When devices are specified as follows (number of points: 3 points)

When using Visual Basic .NET, VBA:	"M0" & vbCrLf & "D0" & vbCrLf & "K8M0"
When using Visual C++:	M0\nD0\nK8M0
When using Visual C#:	M0\nD0\nK8M0

2-byte

M0*1

D0

M0 to M15*2

- *1 The target device to which data is written is only one point of "M0," and the least significant bit of the set 2-byte data is written as the device value.
- *2 Devices are stored from the lower bit in the order of device number.
'0' is written to the upper 2-byte of M16 to M31.

■When devices including CN200 and later of FXCPU are specified (total number of points: 3 points)

When using Visual Basic .NET, VBA:	"D0" & vbCrLf & "CN200" & vbCrLf & "D1"
When using Visual C++:	D0\nCN200\nD1
When using Visual C#:	D0\nCN200\nD1

2-byte

D0

L of CN200*1

D1

- *1 For CN200 or later device of FXCPU, data is written to the L (lower 2-byte) of specified devices for each point when the WriteDeviceRandom2 function is executed.
'0' is written to the H (upper 2-byte) of specified devices.

■When devices including FD are specified (total number of points: 3 points)

When using Visual Basic .NET, VBA:	"D0" & vbCrLf & "FD0" & vbCrLf & "D1"
When using Visual C++:	D0\nFD0\nD1
When using Visual C#:	D0\nFD0\nD1

2-byte

D0

LL of FD0

D1

■When devices including EG are specified (total number of points: 3 points)

The following is the example when 8-bit devices (E0000 and E0001 of SHARP programmable controller) are assigned to EG0.

When using Visual Basic .NET, VBA:	"D0" & vbCrLf & "EG0" & vbCrLf & "D1"
When using Visual C++:	D0\nEG0\nD1
When using Visual C#:	D0\nEG0\nD1

2-byte

D0

EG0

(E0001)

(E0000)

D1

■ When double word devices are specified

When using Visual Basic .NET, VBA: "LTN0" & vbLf & "LTN1" & vbLf & "LTN2"

When using Visual C++: LTN0\nLTN1\nLTN2

When using Visual C#: LTN0\nLTN1\nLTN2

2-byte

L of LTN0

L of LTN1

L of LTN2

SetDevice2 (Setting device data)

Applicable controls

This function can be used for the following controls.

- 32-bit: ActProgType, ActUtilType
- 64-bit: ActProgType64, ActUtilType64

Feature

To set one point of device in 2-byte data unit.

Format

■VBA

IRet = object.SetDevice2(szDevice, iData)

Data type	Argument	Description	Input/Output
Long	IRet	Returned value	Output
String	szDevice	Device name	Input
Integer	iData	Set data	Input

■Visual C++

IRet = object.SetDevice2(szDevice, sData)

Data type	Argument	Description	Input/Output
Long	IRet	Returned value	Output
CString	szDevice	Device name	Input
Short	sData	Set data	Input

■Visual Basic .NET

IRet = object.SetDevice2(szDevice, sData)

Data type	Argument	Description	Input/Output
Integer	IRet	Returned value	Output
String	szDevice	Device name	Input
Short	sData	Set data	Input

■Visual C#

iRet = object.SetDevice2(szDevice, sData)

Data type	Argument	Description	Input/Output
int	iRet	Returned value	Output
String	szDevice	Device name	Input
short	sData	Set data	Input

Format (when creating an object)

■Visual C++

hResult = object.SetDevice2(szDevice, sData, *lpIRetCode)

Data type	Argument	Description	Input/Output
HRESULT	hResult	Returned value of COM	Output
BSTR	szDevice	Device name	Input
LONG	iData	Set data	Input
LONG	*lpIRetCode	Returned value of communication function	Output

Returned value

Normal termination: '0' is returned.

Abnormal termination: Any value other than '0' is returned. ( Page 616 ERROR CODES)

Description

- One point of device specified for szDevice is specified using iData (sData).
- When specifying bit devices, the least significant bit of the iData value (sData value) is valid.

Point

- When specifying a double word device, the SetDevice2data function writes to the data area of lower 1 word (2-byte), and '0' is written to the data area of upper 1 word (2-byte).
Use the WriteDeviceRandom or SetDevice function to write a double word device.
- If the function is run against the safety device in the RnSF CPU safety mode, an error code 0x010A42A5" (an operation that can not be carried out in safety mode was performed) is returned.
- Not applicable to inverter communication/robot controller communication.

How to specify devices

The following describes how to specify devices.

■When specifying bit devices

Example: M0

2-byte

M0^{*1}

*1 The target device to which data is written is only one point of "M0," and the least significant bit of the set 2-byte data is written as the device value.

■When specifying word devices

Example: D0

2-byte

D0

■When specifying double word devices

Example: LTN0

2-byte

L of LTN0

■When specifying double word devices (digit specification)

Example: K8M0

2-byte

M0 to M15^{*1}

*1 Devices are stored from the lower bit in the order of device number.
'0' is written to the upper 2-byte of M16 to M31.

■When specifying CN200 and later devices of FXCPU

Example: CN200

2-byte

L of CN200*1

*1 For CN200 and later devices of FXCPU, data is written to the L (lower 2-byte) of specified devices for each point when the SetDevice2 function is executed.

'0' is written to the H (upper 2-byte) of specified devices.

■When specifying gateway devices

Example: When 8-bit devices (E0000 and E0001 of SHARP programmable controller) are assigned to EG0

2-byte

EG0

(E0001)

(E0000)

GetDevice2 (Acquiring device data)

Applicable controls

This function can be used for the following controls.

- 32-bit: ActProgType, ActUtilType
- 64-bit: ActProgType64, ActUtilType64

Feature

To acquire one point of device in 2-byte data unit.

Format

■VBA

IRet = object.GetDevice2(szDevice, iData)

Data type	Argument	Description	Input/Output
Long	IRet	Returned value	Output
String	szDevice	Device name	Input
Integer	iData	Acquired data	Output

■Visual C++

IRet = object.GetDevice2(szDevice, *lpsData)

Data type	Argument	Description	Input/Output
Long	IRet	Returned value	Output
CString	szDevice	Device name	Input
Short	*lpsData	Acquired data	Output

■Visual Basic .NET

IRet = object.GetDevice2(szDevice, sData)

Data type	Argument	Description	Input/Output
Integer	IRet	Returned value	Output
String	szDevice	Device name	Input
short	sData	Acquired data	Output

■Visual C#

iRet = object.GetDevice2(szDevice, out sData)

Data type	Argument	Description	Input/Output
int	iRet	Returned value	Output
String	szDevice	Device name	Input
short	sData	Acquired data	Output

Format (when creating an object)

■Visual C++

hResult = object.GetDevice2(szDevice, *lpsData, *lplRetCode)

Data type	Argument	Description	Input/Output
HRESULT	hResult	Returned value of COM	Output
BSTR	szDevice	Device name	Input
SHORT	*lpsData	Acquired data	Output
LONG	*lplRetCode	Returned value of communication function	Output

Returned value

Normal termination: '0' is returned.

Abnormal termination: Any value other than '0' is returned. (☞ Page 616 ERROR CODES)

Description

One point of device data specified for szDevice is stored in iData (lpsData or sData).

Point

- When a double word device is specified, only the data of lower 1 word (2-byte) is stored using the GetDevice2 function. (An error does not occur.)
- When reading data from a double word device, use the ReadDeviceRandom or GetDevice function.
- Not applicable to inverter communication/robot controller communication.

How to specify devices

The following describes how to specify devices.

■When specifying bit devices

Example: M0

2-byte

M0*1

*1 The target device from which data is read is only one point of "M0," and '0' or '1' is stored as the device value.

■When specifying word devices

Example: D0

2-byte

D0

■When specifying double word devices

Example: LTN0

2-byte

L of LTN0

■When specifying double word devices (digit specification)

Example: K8M0

2-byte

M0 to M15*1

*1 Devices are stored from the lower bit in the order of device number.
Data is not read from the upper 2-byte of M16 to M31.

■When specifying CN200 and later devices of FXCPU

Example: CN200

2-byte

L of CN200*1

*1 For CN200 and later devices of FXCPU, the L (lower 2-byte) of specified devices is read for each point when the GetDevice2 function is executed.
The H (upper 2-byte) of specified devices is not read.

■When specifying gateway devices

Example: When 8-bit devices (E0000 and E0001 of SHARP programmable controller) are assigned to EG0

2-byte

EG0

(E0001)

(E0000)

GetErrorMessage(Acquiring error description and corrective action)

Applicable controls

This function can be used for the following controls.

- 32-bit: ActSupportMsg
- 64-bit: ActSupportMsg64

Feature

To acquire the error description and its corrective action corresponding to the error code.

Format

■VBA

IRet = object.GetErrorMessage(IErrCode, szErrorMessage)

Data type	Argument	Description	Input/Output
Long	IRet	Returned value	Output
Long	IErrCode	Error code	Input
String	szErrorMessage	Error message	Output

■Visual C++

IRet = object.GetErrorMessage(IErrCode, *lpszErrorMessage)

Data type	Argument	Description	Input/Output
Long	IRet	Returned value	Output
Long	IErrCode	Error code	Input
BSTR	*lpszErrorMessage	Error message	Output

■Visual Basic .NET

IRet = object.GetErrorMessage(IErrCode, szErrorMessage)

Data type	Argument	Description	Input/Output
Integer	IRet	Returned value	Output
Integer	IErrCode	Error code	Input
String	szErrorMessage	Error message	Output

■Visual C#

iRet = object.GetErrorMessage(iErrCode, out szErrorMessage)

Data type	Argument	Description	Input/Output
int	iRet	Returned value	Output
int	iErrCode	Error code	Input
String	szErrorMessage	Error message	Output

Format (when creating an object)

■Visual C++

hResult = object.GetErrorMessage(IErrCode, *lpszErrorMessage, *lplRetCode)

Data type	Argument	Description	Input/Output
HRESULT	hResult	Returned value of COM	Output
LONG	IErrCode	Error code	Input
BSTR	*lpszErrorMessage	Error message	Output
LONG	*lplRetCode	Returned value of communication function	Output

Returned value

Normal termination: '0' is returned.

Abnormal termination: Any value other than '0' is returned. (🔗 Page 616 ERROR CODES)

Description

- The error description and its corrective action of the error code specified for IErrorCode (iErrorCode) are read.
- The read error description and corrective action are stored in szErrorMessage (lpzErrorMessage or lpvarErrorMessage).

Point

Not applicable to inverter communication.

11.4 Details of Functions (For Act Control (Logging File Transfer))

Open (Opening communication line)

Applicable controls

This function can be used for the following controls.

- 32-bit: ActProgDataLogging, ActUtilDataLogging
- 64-bit: ActProgDataLogging64, ActUtilDataLogging64

Feature

To open a communication line.

Format

■Visual C++

IRet = object.Open()

Data type	Argument	Description	Input/Output
Long	IRet	Returned value	Output

■Visual Basic .NET

IRet = object.Open()

Data type	Argument	Description	Input/Output
Integer	IRet	Returned value	Output

■Visual C#

iRet = object.Open()

Data type	Argument	Description	Input/Output
int	iRet	Returned value	Output

Format (when creating an object)

■Visual C++

hResult = object.Open(*lplReturnCode)

Data type	Argument	Description	Input/Output
HRESULT	hResult	Returned value of COM	Output
LONG	*lplReturnCode	Returned value of communication function	Output

Returned value

Normal termination: '0' is returned.

Abnormal termination: Any value other than '0' is returned. ( Page 616 ERROR CODES)

Description

Lines are connected according to the set value of the Open function property.

Point

- If the Open function property is changed after the completion of the Open function, the setting of communication target is not changed.
To change the communication setting, close the communication line, set the setting of communication target, and open the communication line again.
- Even when a CPU type which is different from the one used for the communication is set to the ActCpuType property, the Open function may be completed normally.
When executing the Open function, set the correct CPU type to the ActCpuType property.

Close (Closing communication line)

Applicable controls

This function can be used for the following controls.

- 32-bit: ActProgDataLogging, ActUtilDataLogging
- 64-bit: ActProgDataLogging64, ActUtilDataLogging64

Feature

To close a communication line.

Format

■Visual C++

IRet = object.Close()

Data type	Argument	Description	Input/Output
Long	IRet	Returned value	Output

■Visual Basic .NET

IRet = object.Close()

Data type	Argument	Description	Input/Output
Integer	IRet	Returned value	Output

■Visual C#

iRet = object.Close()

Data type	Argument	Description	Input/Output
int	iRet	Returned value	Output

Format (when creating an object)

■Visual C++

hResult = object.Close(*lpIReturnCode)

Data type	Argument	Description	Input/Output
HRESULT	hResult	Returned value of COM	Output
LONG	*lpIReturnCode	Returned value of communication function	Output

Returned value

Normal termination: '0' is returned.

Abnormal termination: Any value other than '0' is returned. ( Page 616 ERROR CODES)

Description

The line connected using the Open function is closed.



- Call this function when an error occurred after calling the Open function or when exit the program.

ReadFirstFile (Searching for a file/directory)

Applicable controls

This function can be used for the following controls.

- 32-bit: ActProgDataLogging, ActUtilDataLogging
- 64-bit: ActProgDataLogging64, ActUtilDataLogging64

Feature

To search a file or directory in the specified directory.

Format

■ Visual C++

Ret = object.ReadFirstFile(srcDir, *srcFilename, *srcType)

Data type	Argument	Description	Input/Output
Long	Ret	Returned value	Output
CString	srcDir	Directory name	Input
BSTR	*srcFileName	File name or directory name	Output
Long	*srcType	Type	Output

■ Visual Basic .NET

Ret = object.ReadFirstFile(srcDir, srcFilename, srcType)

Data type	Argument	Description	Input/Output
Integer	Ret	Returned value	Output
String	srcDir	Directory name	Input
String	srcFileName	File name or directory name	Output
Integer	srcType	Type	Output

■ Visual C#

Ret = object.ReadFirstFile(srcDir, out srcFilename, out srcType)

Data type	Argument	Description	Input/Output
int	Ret	Returned value	Output
String	srcDir	Directory name	Input
String	srcFileName	File name or directory name	Output
int	srcType	Type	Output

Format (when creating an object)

■ Visual C++

hResult = object.ReadFirstFile(srcDir, *srcFilename, *srcType, *lpIReturnCode)

Data type	Argument	Description	Input/Output
HRESULT	hResult	Returned value of COM	Output
BSTR	srcDir	Directory name	Input
BSTR	*srcFileName	File name or directory name	Output
LONG	*srcType	Type	Output
LONG	*lpIReturnCode	Returned value of communication function	Output

Returned value

Normal termination: '0' is returned.

Abnormal termination: Any value other than '0' is returned. (📖 Page 616 ERROR CODES)

Description

Search a file or directory in the specified directory.

- A read file name or directory name is stored in srcFileName.
- A value that identifies if the name set for srcFileName indicates a directory or file is stored in srcType.

For a directory, '1' is stored in the 4th bit counted from the lowest-order position.

As for a file, '0' is stored.

Example: When srcType is 24 (0x18), it indicates that the target is a directory.

Upper 3 byte	Lower 1 byte							
0	0	0	0	1	1	0	0	0

Example: When srcType is 16 (0x10), it indicates that the target is a file.

Upper 3 byte	Lower 1 byte							
0	0	0	0	1	0	0	0	0

Do not use the bits which are not described in the figure above.

- Files or directories under a sub directory are not searched.

Point

- The control does not return until the processing is completed.
- If two or more controls are called at the same time, the search result will be overwritten.
To search multiple directories, generate controls for each directory.
- This function is available while logging is stopped.
- For srcDir, specify the length of a character string within 188 bytes.
- The search result is displayed randomly.

ReadNextFile (Searching for a file/directory)

Applicable controls

This function can be used for the following controls.

- 32-bit: ActProgDataLogging, ActUtilDataLogging
- 64-bit: ActProgDataLogging64, ActUtilDataLogging64

Feature

To continue searching other files from the last search result of ReadFirstFile.

Format

■ Visual C++

Ret = object.ReadNextFile(*srcFilename, *srcType)

Data type	Argument	Description	Input/Output
Long	Ret	Returned value	Output
BSTR	*srcFileName	File name or directory name	Output
Long	*srcType	Type	Output

■ Visual Basic .NET

Ret = object.ReadNextFile(srcFilename, srcType)

Data type	Argument	Description	Input/Output
Integer	Ret	Returned value	Output
String	srcFileName	File name or directory name	Output
Integer	srcType	Type	Output

■ Visual C#

Ret = object.ReadNextFile(out srcFilename, out srcType)

Data type	Argument	Description	Input/Output
int	Ret	Returned value	Output
String	srcFileName	File name or directory name	Output
int	srcType	Type	Output

Format (when creating an object)

■ Visual C++

hResult = object.ReadNextFile(*srcFilename, *srcType, *lpReturnCode)

Data type	Argument	Description	Input/Output
HRESULT	hResult	Returned value of COM	Output
BSTR	*srcFileName	File name or directory name	Output
LONG	*srcType	Type	Output
LONG	*lpReturnCode	Returned value of communication function	Output

Returned value

Normal termination: '0' is returned.

Abnormal termination: Any value other than '0' is returned. (📖 Page 616 ERROR CODES)

Description

After searching for a file or directory using ReadFirstFile, search for the next file or directory in the specified directory.

- A read file name or directory name is stored in srcFileName.
- A value that identifies if the name set for srcFileName indicates a directory or file is stored in srcType.

For a directory, '1' is stored in the 4th bit counted from the lowest-order position.

As for a file, '0' is stored.

Do not use the bits which are not described in the figure above.

Example: When srcType is 24 (0x18), it indicates that the target is a directory.

Upper 3 byte	Lower 1 byte							
0	0	0	0	1	1	0	0	0

Example: When srcType is 16 (0x10), it indicates that the target is a file.

Upper 3 byte	Lower 1 byte							
0	0	0	0	1	0	0	0	0

Point

- The control does not return until the processing is completed.
- This function is available while logging is stopped.
- The search result is displayed randomly.

ReadClose (Ending the search)

Applicable controls

This function can be used for the following controls.

- 32-bit: ActProgDataLogging, ActUtilDataLogging
- 64-bit: ActProgDataLogging64, ActUtilDataLogging64

Feature

To end the search.

Format

■ Visual C++

Ret = object.ReadClose()

Data type	Argument	Description	Input/Output
Long	Ret	Returned value	Output

■ Visual Basic .NET

Ret = object.ReadClose()

Data type	Argument	Description	Input/Output
Integer	Ret	Returned value	Output

■ Visual C#

Ret = object.ReadClose()

Data type	Argument	Description	Input/Output
int	Ret	Returned value	Output

Format (when creating an object)

■ Visual C++

hResult = object.ReadClose(*lpReturnCode)

Data type	Argument	Description	Input/Output
HRESULT	hResult	Returned value of COM	Output
LONG	*lpReturnCode	Returned value of communication function	Output

Returned value

Normal termination: '0' is returned.

Abnormal termination: Any value other than '0' is returned. (🔗 Page 616 ERROR CODES)

Description

End the search.

- The result for searching for a directory or file name is deleted.

Point

- The control does not return until the processing is completed.
- This function is available while logging is stopped.

GetFile (Transferring logging files)

Applicable controls

This function can be used for the following controls.

- 32-bit: ActProgDataLogging, ActUtilDataLogging
- 64-bit: ActProgDataLogging64, ActUtilDataLogging64

Feature

To transfer a logging file.

Format

■Visual C++

Ret = object.GetFile(srcFileName , dstFileName)

Data type	Argument	Description	Input/Output
Long	Ret	Returned value	Output
CString	srcFileName	Transfer source file name	Input
CString	dstFileName	Transfer destination file name	Input

■Visual Basic .NET

Ret = object.GetFile(srcFileName , dstFileName)

Data type	Argument	Description	Input/Output
Integer	Ret	Returned value	Output
String	srcFileName	Transfer source file name	Input
String	dstFileName	Transfer destination file name	Input

■Visual C#

Ret = object.GetFile(srcFileName , dstFileName)

Data type	Argument	Description	Input/Output
int	Ret	Returned value	Output
String	srcFileName	Transfer source file name	Input
String	dstFileName	Transfer destination file name	Input

Format (when creating an object)

■Visual C++

hResult = object.GetFile(srcFileName , dstFileName, *lpReturnCode)

Data type	Argument	Description	Input/Output
HRESULT	hResult	Returned value of COM	Output
BSTR	srcFileName	Transfer source file name	Input
BSTR	dstFileName	Transfer destination file name	Input
LONG	*lpReturnCode	Returned value of communication function	Output

Returned value

Normal termination: '0' is returned.

Abnormal termination: Any value other than '0' is returned. ( Page 616 ERROR CODES)

Description

Transfer a logging file.

- The file specified for srcFileName is transferred to the file specified for dstFileName.

Point

- The control does not return until the processing is completed.
- This function is available while logging is stopped.
- Specify a character string for srcFileName within 318 bytes.
- Specify a character string for dstFileName within 259 bytes.

11.5 Details of Functions (For .Net Control)

Open (Opening communication line)

Applicable controls

This function can be used for the following controls.

- 32-bit: DotUtilType
- 64-bit: DotUtilType64

Feature

To open a communication line.

Format

■Visual Basic .NET

IRet = object.Open()

Data type	Argument	Description	Input/Output
Integer	IRet	Returned value	Output

■Visual C#

iRet = object.Open()

Data type	Argument	Description	Input/Output
int	iRet	Returned value	Output

Returned value

Normal termination: '0' is returned.

Abnormal termination: Any value other than '0' is returned. ( Page 616 ERROR CODES)

Description

Lines are connected according to the set value of the Open function property.

Point

- If the Open function property is changed after the completion of the Open function, the setting of communication target is not changed.
To change the communication setting, close the communication line, set the setting of communication target, and open the communication line again.
- Even when a CPU type which is different from the CPU used for the communication is set to the ActCpuType property, the Open function may be completed normally.
In such a case, the connection range, usable method, or device range may be narrowed.
When executing the Open function, set the correct CPU type to the ActCpuType property.

Close (Closing communication line)

Applicable controls

This function can be used for the following controls.

- 32-bit: DotUtilType
- 64-bit: DotUtilType64

Feature

To close a communication line.

Format

■Visual Basic .NET

IRet = object.Close()

Data type	Argument	Description	Input/Output
Integer	IRet	Returned value	Output

■Visual C#

iRet = object.Close()

Data type	Argument	Description	Input/Output
int	iRet	Returned value	Output

Returned value

Normal termination: '0' is returned.

Abnormal termination: Any value other than '0' is returned. ( Page 616 ERROR CODES)

Description

The line connected using the Open function is closed.

ReadDeviceBlock (Reading devices in batch)

Applicable controls

This function can be used for the following controls.

- 32-bit: DotUtilType
- 64-bit: DotUtilType64

Feature

To read array labels in batch.

Format

■ Visual Basic .NET

`iRet = object.ReadDeviceBlock(szLabel, iSize, iData)`

Data type	Argument	Description	Input/Output
Integer	iRet	Returned value	Output
String	szLabel	Label name	Input
Integer	iSize	Number of read points	Input
Integer	iData(n)	Read device value	Output

■ Visual C#

`iRet = object.ReadDeviceBlock(ref szLabel, iSize, ref iData)`

Data type	Argument	Description	Input/Output
Int	iRet	Returned value	Output
String	szLabel	Label name	Input
int	iSize	Number of read points	Input
int[n]	iData	Read device value	Output

Returned value

Normal termination: '0' is returned.

Abnormal termination: Any value other than '0' is returned. ([↩ Page 616 ERROR CODES](#))

Description

- Data of devices specified for iSize (number of read points) is read in batch starting from the device specified for szLabel (label name).
- The read device values are stored in iData.
- For iData, reserve arrays for more than the amount specified for iSize.

- The maximum number of read points is 0x7FFFFFFF, which satisfies the following condition:
Start read device number + Number of read points \leq Last device number
- When specifying bit type array labels, 0 or a multiple of 16 can be specified as a device number.
- For the number of read points, specify the number of words which applies to the data type specified for the label name.

For the read device values, reserve a memory area for the number of points specified for the number of read points.

If the memory area is not reserved, a critical error (an application error or the like) may occur.

- When a device which corresponds to the label name does not exist, an error occurs and data cannot be read.

When any one of devices which correspond to multiple label names does not exist, an error occurs and data cannot be read.

- Digit specified bit device and index setting cannot be used.
- Devices of long timer contact (LTS), long timer coil (LTC), long retentive timer contact (LSTS), long retentive timer coil (LSTC) cannot be read. Use the ReadDeviceRandom or GetDevice function.
- When using system label Ver.2, the data type defined in the Label Utility of MX Component and the data type managed by MELSOFT Navigator is required to match.

If the data type does not match, the read data length may wrong, or when multiple labels are specified, the correspondence of the array of the read device value and label name may not match.

How to specify labels

The following describes how to specify labels.

For the system label name, the following type class, array element, and label data type can be specified.

When specifying a system label name in a program, use the following formats.

○: Available, ×: Not available, —: Not applicable

Label definition in Label Utility		Label specification method in a program		
Type class	Array element	Label name format	Availability	Data type
Simple Types	None	System label name	×	—
	Element (One Dimension)	System label name ^{*1}	○	Integer/int (32-bit value)
		System label name [n1]	×	—
	Element (Two Dimensions)	System label name ^{*1}	○	Integer/int (32-bit value)
		System label name [n1] [n2]	×	—
	Element (Three Dimensions)	System label name ^{*1}	○	Integer/int (32-bit value)
System label name [n1] [n2] [n3]		×	—	
Structure	None	System label name	×	—
		System label name.structure member name	×	—
	Element (One Dimension)	System label name	×	—
		System label name [n1]	×	—
		System label name [n1].structure member name	×	—
	Element (Two Dimensions)	System label name	×	—
		System label name [n1] [n2]	×	—
		System label name [n1] [n2].structure member name	×	—
	Element (Three Dimensions)	System label name	×	—
		System label name [n1] [n2] [n3]	×	—
		System label name [n1] [n2] [n3].structure member name	×	—

*1 Specified without specifying an array element.

■Data type

The following data type can be specified for a label.

Bit^{*3}, Word (Signed), Double Word (Signed), Word (Unsigned), Double Word (Unsigned), Float (Single Precision), Float (Double Precision), String^{*1}, String (Unicode)^{*2,3}, Time, Timer, Timer (Unsigned), Long timer^{*3}, Counter, Counter (Unsigned), Long counter^{*3}, Retentive Timer, Retentive Timer (Unsigned), Long retentive timer^{*3}

*1 Can be specified up to 32 single-byte characters + NULL.

*2 Can be specified up to 32 characters + NULL.

*3 Cannot be used for system label Ver.1.

■Number of read points

For the number of read points, specify the following values according to the label data type.

Label (array) data type	Number of applicable words	Number of read points to be specified
Bit	1	Number of label array elements ÷ 16 (rounded up)
Word (Signed)	1	Number of label array elements
Double Word (Signed)	2 ^{*1}	Number of label array elements × 2 ^{*1}
Word (Unsigned)	1	Number of label array elements
Double Word (Unsigned)	2 ^{*1}	Number of label array elements × 2 ^{*1}
Float (Single Precision)	2 ^{*1}	Number of label array elements × 2 ^{*1}
Float (Double Precision)	4	Number of label array elements × 4
String	17	Number of label array elements × 17
String (Unicode)	33	Number of label array elements × 33
Time	2 ^{*1}	Number of label array elements × 2 ^{*1}
Timer, Timer (Unsigned), Long timer, Counter, Counter (Unsigned), Long counter, Retentive Timer, Retentive Timer (Unsigned), Long retentive timer	1	Number of label array elements

*1 When specifying a double word device for a label, the number of devices (elements) is 1.

How to specify devices

The read device values are stored as follows:

■When specifying bit devices

Example: Read 3 points (3 words = 48 bits) of data from the devices starting from M0.

· Label setting (Data type: Bit, Number of array elements: 48)

szLabel	Label name	Data type	Device
[0] to [2]	LABEL	Bit (0..47)	M0 to M47

· Number of read points: 3

· Read device values

Upper 2-byte	Lower 2-byte	Applicable label
Not used ('0' is stored.)	M0 to M15 ^{*1}	LABEL[0] to LABEL[15]
	M16 to M31 ^{*1}	LABEL[16] to LABEL[31]
	M32 to M47 ^{*1}	LABEL[30] to LABEL[47]

*1 Devices are stored from the lower bit in the order of device number.

■When specifying word devices

Example: Read 3 points (3 words) of data from the devices starting from D0.

· Label setting (Data type: Word, Number of array elements: 3)

szLabel	Label name	Data type	Device
[0] to [2]	LABEL	Word (0..2)	D0 to D2

· Number of read points: 3

· Read device values

Upper 2-byte	Lower 2-byte	Applicable label
Not used ('0' is stored.)	D0	LABEL[0]
	D1	LABEL[1]
	D2	LABEL[2]

■When specifying double word devices (CN200 and later devices of FXCPU)

Example: Read 6 points (6 words = 3 double words) of data from the devices starting from CN200.

For CN200 and later devices of FXCPU, 2 points of data are read from each 1 point of device (upper (H) and lower (L) data).

Reading only 1 point of data will result in an error.

· Label setting

szLabel	Label name	Data type	Device
[0] to [5]	LABEL	Double Word (0..2)	CN200 to CN202

· Number of read points: 6

· Read device values

Upper 2-byte	Lower 2-byte	Applicable label
Not used ('0' is stored.)	L of CN200	LABEL[0]
	H of CN200	
	L of CN201	LABEL[1]
	H of CN201	
	L of CN202	LABEL[2]
	H of CN202	

■When specifying FD devices (4-word devices)

Example: Read 8 points (8 words) of data from the devices starting from FD0.

· Label setting (Data type: Double Word, Number of array elements: 2)

szLabel	Label name	Data type	Device
[0] to [7]	LABEL	Double Word (0..1)	FD0, FD1

· Number of read points: 8

· Read device values

Upper 2-byte	Lower 2-byte	Applicable label
Not used ('0' is stored.)	LL of FD0	LABEL[0]
	LH of FD0	
	HL of FD0	
	HH of FD0	
	LL of FD1	LABEL[1]
	LH of FD1	
	HL of FD1	
	HH of FD1	

■When specifying word devices for double word array type labels

Example: Read 6 points (6 words = 3 double words) of data from the devices starting from D100.

· Label setting (Data type: Double Word, Number of array elements: 3)

szLabel	Label name	Data type	Device
[0] to [5]	LABEL	Double Word (0..2)	D100 to D105

· Number of read points: 6

· Read device values

Upper 2-byte	Lower 2-byte	Applicable label
Not used ('0' is stored.)	D100	LABEL[0]
	D101	
	D102	LABEL[1]
	D103	
	D104	LABEL[2]
	D105	

■When specifying double word devices to be acquired for 1 point in Double Word type

Example: Read 2 points (2 words) of data from the devices starting from LZ0.

· Label setting (Data type: Double Word, Number of array elements: 2)

szLabel	Label name	Data type	Device
[0] to [1]	LABEL	Double Word (0..1)	LZ0 to LZ1

· Number of read points: 2

· Read device values

Upper 2-byte	Lower 2-byte	Applicable label
H of LZ0	L of LZ0	LABEL[0]
H of LZ1	L of LZ1	LABEL[1]

WriteDeviceBlock (Writing devices in batch)

Applicable controls

This function can be used for the following controls.

- 32-bit: DotUtilType
- 64-bit: DotUtilType64

Feature

To write array labels in batch.

Format

■ Visual Basic .NET

`IRet = object.WriteDeviceBlock(szLabel, iSize, iData)`

Data type	Argument	Description	Input/Output
Integer	IRet	Returned value	Output
String	szLabel	Label name	Input
Integer	iSize	Number of write points	Input
Integer	iData(n)	Device value to be written	Input

■ Visual C#

`iRet = object.WriteDeviceBlock(ref szLabel, iSize, iData)`

Data type	Argument	Description	Input/Output
int	iRet	Returned value	Output
String	szLabel	Label name	Input
int	iSize	Number of write points	Input
int[n]	iData	Device value to be written	Input

Returned value

Normal termination: '0' is returned.

Abnormal termination: Any value other than '0' is returned. ([Page 616 ERROR CODES](#))

Description

- Data of devices specified for iSize (number of write points) is written in batch starting from the device specified for szLabel (label name).
- The device values to be written are stored in iData.
- For iData, reserve arrays for more than the amount specified for iSize.

- The maximum number of write points should be the value which satisfies the following condition:
Start write device number + Number of write points ≤ Last device number
- When specifying bit type array labels, 0 or a multiple of 16 can be specified as a device number.
- For the number of write points, specify the number of words which applies to the data type specified for the label name.
For the device values to be written, reserve a memory area for the number of points specified for the number of write points.
If the memory area is not reserved, a critical error (an application error or the like) may occur.
- When a device which corresponds to the label name does not exist, an error occurs and data cannot be written.
When any one of devices which correspond to multiple label names does not exist, an error occurs and data cannot be written.
- Digit specified bit device and index setting cannot be used.
- Devices of long timer contact (LTS), long timer coil (LTC), long retentive timer contact (LSTS), long retentive timer coil (LSTC) cannot be written. Use the WriteDeviceRandom or SetDevice function.
- When using system label Ver.2, the data type defined in the Label Utility of MX Component and the data type managed by MELSOFT Navigator is required to match.
If the data type does not match, the read data length may wrong, or when multiple labels are specified, the correspondence of the array of the read device value and label name may not match.
- If the function is run against the safety device in the RnSFPCPU safety mode, an error code 0x010A42A5" (an operation that can not be carried out in safety mode was performed) is returned.

How to specify labels

The following describes how to specify labels.

For the system label name, the following type class, array element, and label data type can be specified.

When specifying a system label name in a program, use the following formats.

○: Available, ×: Not available, —: Not applicable

Label definition in Label Utility		Label specification method in a program		
Type class	Array element	Label name format	Availability	Data type
Simple Types	None	System label name	×	—
	Element (One Dimension)	System label name ^{*1}	○	Integer/int (32-bit value)
		System label name [n1]	×	—
	Element (Two Dimensions)	System label name ^{*1}	○	Integer/int (32-bit value)
		System label name [n1] [n2]	×	—
	Element (Three Dimensions)	System label name ^{*1}	○	Integer/int (32-bit value)
System label name [n1] [n2] [n3]		×	—	
Structure	None	System label name	×	—
		System label name.structure member name	×	—
	Element (One Dimension)	System label name	×	—
		System label name [n1]	×	—
		System label name [n1].structure member name	×	—
	Element (Two Dimensions)	System label name	×	—
		System label name [n1] [n2]	×	—
		System label name [n1] [n2].structure member name	×	—
	Element (Three Dimensions)	System label name	×	—
		System label name [n1] [n2] [n3]	×	—
		System label name [n1] [n2] [n3].structure member name	×	—

*1 Specified without specifying an array element.

■Data type

The following data type can be specified for a label.

Bit^{*3}, Word (Signed), Double Word (Signed), Word (Unsigned), Double Word (Unsigned), Float (Single Precision), Float (Double Precision), String^{*1}, String (Unicode)^{*2*}, Time, Timer, Timer (Unsigned), Long timer^{*3}, Counter, Counter (Unsigned), Long counter^{*3}, Retentive Timer, Retentive Timer (Unsigned), Long retentive timer^{*3}

*1 Can be specified up to 32 single-byte characters + NULL.

*2 Can be specified up to 32 characters + NULL.

*3 Cannot be used for system label Ver.1.

■Number of write points

For the number of write points, specify the following values according to the label data type.

Label (array) data type	Number of applicable words	Number of write points to be specified
Bit	1	Number of label array elements ÷ 16 (rounded up)
Word (Signed)	1	Number of label array elements
Double Word (Signed)	2 ^{*1}	Number of label array elements × 2 ^{*1}
Word (Unsigned)	1	Number of label array elements
Double Word (Unsigned)	2 ^{*1}	Number of label array elements × 2 ^{*1}
Float (Single Precision)	2 ^{*1}	Number of label array elements × 2 ^{*1}
Float (Double Precision)	4	Number of label array elements × 4
String	17	Number of label array elements × 17
String (Unicode)	33	Number of label array elements × 33
Time	2 ^{*1}	Number of label array elements × 2 ^{*1}
Timer, Timer (Unsigned), Long timer, Counter, Counter (Unsigned), Long counter, Retentive Timer, Retentive Timer (Unsigned), Long retentive timer	1	Number of label array elements

*1 When specifying a double word device for a label, the number of devices (elements) is 1.

How to specify devices

Specify the device values to be written as follows:

■When specifying bit devices

Example: Write 3 points (3 words = 48 bits) of data to the devices starting from M0.

· Label setting (Data type: Bit, Number of array elements: 48)

szLabel	Label name	Data type	Device
[0] to [2]	LABEL	Bit (0..47)	M0 to M47

· Number of write points: 3
 · Device values to be written

Upper 2-byte	Lower 2-byte	Applicable label
Not used ('0' is stored.)	M0 to M15 ^{*1}	LABEL[0] to LABEL[15]
	M16 to M31 ^{*1}	LABEL[16] to LABEL[31]
	M32 to M47 ^{*1}	LABEL[32] to LABEL[47]

*1 Devices are stored from the lower bit in the order of device number.

■When specifying word devices

Example: Write 3 points (3 words) of data to the devices starting from D0.

· Label setting (Data type: Word, Number of array elements: 3)

szLabel	Label name	Data type	Device
[0] to [2]	LABEL	Word (0..2)	D0 to D2

· Number of write points: 3
 · Device values to be written

Upper 2-byte	Lower 2-byte	Applicable label
Not used ('0' is stored.)	D0	LABEL[0]
	D1	LABEL[1]
	D2	LABEL[2]

■When specifying double word devices (CN200 and later devices of FXCPU)

Example: Write 6 points (6 words = 3 double words) of data to the devices starting from CN200.

For CN200 and later devices of FXCPU, 2 points of data are written to each 1 point of device (upper (H) and lower (L) data).

Writing only 1 point of data will result in an error.

· Label setting (Data type: Double Word, Number of array elements: 3)

szLabel	Label name	Data type	Device
[0] to [5]	LABEL	Double Word (0..2)	CN200 to CN202

· Number of write points: 3

· Device values to be written

Upper 2-byte	Lower 2-byte	Applicable label
Not used ('0' is stored.)	L of CN200	LABEL[0]
	H of CN200	
	L of CN201	LABEL[1]
	H of CN201	
	L of CN202	LABEL[2]
	H of CN202	

■When specifying FD devices (4-word devices)

Example: Write 8 points (8 words) of data to the devices starting from FD0.

· Label setting (Data type: Double Word, Number of array elements: 2)

szLabel	Label name	Data type	Device
[0] to [7]	LABEL	Double Word (0..1)	FD0, FD1

· Number of write points: 8

· Device values to be written

Upper 2-byte	Lower 2-byte	Applicable label
Not used ('0' is stored.)	LL of FD0	LABEL[0]
	LH of FD0	
	HL of FD0	
	HH of FD0	
	LL of FD1	LABEL[1]
	LH of FD1	
	HL of FD1	
	HH of FD1	

■When specifying word devices for double word array type labels

Example: Write 6 points (6 words = 3 double words) of data to the devices starting from D100.

· Label setting (Data type: Double Word, Number of array elements: 3)

szLabel	Label name	Data type	Device
[0] to [5]	LABEL	Double Word (0..2)	D100 to D105

· Number of write points: 6

· Device values to be written

Upper 2-byte	Lower 2-byte	Applicable label
Not used ('0' is stored.)	D100	LABEL[0]
	D101	
	D102	LABEL[1]
	D103	
	D104	LABEL[2]
	D105	

■When specifying double word devices to be acquired for 1 point in Double Word type

Example: Write 2 points (2 words) of data to the devices starting from LZ0.

· Label setting (Data type: Double Word, Number of array elements: 2)

szLabel	Label name	Data type	Device
[0] to [1]	LABEL	Double Word (0..1)	LZ0 to LZ1

· Number of write points: 2

· Device values to be written

Upper 2-byte	Lower 2-byte	Applicable label
H of LZ0	L of LZ0	LABEL[0]
H of LZ1	L of LZ1	LABEL[1]

ReadDeviceRandom (Reading devices randomly)

Applicable controls

This function can be used for the following controls.

- 32-bit: DotUtilType
- 64-bit: DotUtilType64

Feature

To read devices randomly.

Format

■Visual Basic .NET

`IRet = object.ReadDeviceRandom(szLabel, iSize, iData)`

Data type	Argument	Description	Input/Output
Integer	IRet	Returned value	Output
String	szLabel	Label name	Input
Integer	iSize	Number of read points	Input
Integer	iData(n)	Read device value	Output

`IRet = object.ReadDeviceRandom(szLabelList, iSize, iData)`

Data type	Argument	Description	Input/Output
Integer	IRet	Returned value	Output
String	szLabelList(n)	Label list	Input
Integer	iSize	Number of read points	Input
Integer	iData(n)	Read device value	Output

■Visual C#

`iRet = object.ReadDeviceRandom(ref szLabel, iSize, ref iData)`

Data type	Argument	Description	Input/Output
int	iRet	Returned value	Output
String	szLabel	Label name	Input
int	iSize	Number of read points	Input
int[n]	iData	Read device value	Output

`iRet = object.ReadDeviceRandom(ref szLabelList, iSize, ref iData)`

Data type	Argument	Description	Input/Output
int	iRet	Returned value	Output
System.String[n]	szLabelList	Label list	Input
int	iSize	Number of read points	Input
int[n]	iData	Read device value	Output

Returned value

Normal termination: '0' is returned.

Abnormal termination: Any value other than '0' is returned. (📖 Page 616 ERROR CODES)

Description

- Data of a device group specified for the label name szLabel (szLabelList) is read for the size of iSize.
- The read device values are stored in iData.
- For iData, reserve arrays for more than the amount specified for iSize.

Point

- The maximum number of read points is 0x7FFFFFFF.
- For the number of read points, specify the number of words which applies to the data type specified for the label name.
For the read device values, reserve a memory area for the number of points specified for the number of read points.
If the memory area is not reserved, a critical error (an application error or the like) may occur.
- When a device which corresponds to the label name does not exist, an error occurs and data cannot be read.
When any one of devices which correspond to multiple label names does not exist, an error occurs and data cannot be read.
- Digit specified bit device and index setting cannot be used.
- Up to five levels of a structure can be defined. Note that only one-level structure can be used for a static system label.
- When using system label Ver.2, the data type defined in the Label Utility of MX Component and the data type managed by MELSOFT Navigator is required to match.
If the data type does not match, the read data length may wrong, or when multiple labels are specified, the correspondence of the array of the read device value and label name may not match.

How to specify labels

The following describes how to specify labels.

For the system label name, the following type class, array element, and label data type can be specified.

When specifying a system label name in a program, use the following formats.

○: Available, ×: Not available, —: Not applicable

Label definition in Label Utility		Label specification method in a program			
Type class	Array element	Label name format	Availability	Data type	
Simple Types	None	System label name	○	Integer/int (32-bit value)	
		System label name	○*1	Integer/int (32-bit value)	
	Element (One Dimension)	System label name [n1]	○	Integer/int (32-bit value)	
		System label name	○*1	Integer/int (32-bit value)	
	Element (Two Dimensions)	System label name [n1] [n2]	○	Integer/int (32-bit value)	
		System label name	○*1	Integer/int (32-bit value)	
	Element (Three Dimensions)	System label name [n1] [n2] [n3]	○	Integer/int (32-bit value)	
		System label name	○*1	Integer/int (32-bit value)	
	Structure	None	System label name	○*1	Integer/int (32-bit value)
			System label name.structure member name	○	Integer/int (32-bit value)
		Element (One Dimension)	System label name	○*1	Integer/int (32-bit value)
			System label name [n1]	○*1	Integer/int (32-bit value)
System label name [n1].structure member name			○	Integer/int (32-bit value)	
Element (Two Dimensions)		System label name	○*1	Integer/int (32-bit value)	
		System label name [n1] [n2]	○*1	Integer/int (32-bit value)	
		System label name [n1] [n2].structure member name	○	Integer/int (32-bit value)	
Element (Three Dimensions)		System label name	○*1	Integer/int (32-bit value)	
		System label name [n1] [n2] [n3]	○*1	Integer/int (32-bit value)	
		System label name [n1] [n2] [n3].structure member name	○	Integer/int (32-bit value)	

*1 The entire data of arrays and structures is read.

■Data type

The following data type can be specified for a label.

Bit, Word (Signed), Double Word (Signed), Word (Unsigned), Double Word (Unsigned), Float (Single Precision), Float (Double Precision), String^{*1}, String (Unicode)^{*2,4}, Time, Timer^{*3}, Timer (Unsigned)^{*3}, Long timer^{*3,4}, Counter^{*3}, Counter (Unsigned)^{*3}, Long counter^{*3,4}, Retentive Timer^{*3}, Retentive Timer (Unsigned)^{*3}, Long retentive timer^{*3,4}

*1 Can be specified up to 32 single-byte characters + NULL.

*2 Can be specified up to 32 characters + NULL.

*3 When accessing the device using a label of system label Ver.2, the label of system label Ver.2 which is corresponding to a device for contact/coil/present value is required to be defined.

*4 Cannot be used for system label Ver.1.

■Number of read points

For the number of read points, specify the sum of the following values correspond to the elements according to the label data type.

Label data type	Number of read points to be specified ^{*1}
Bit	1
Word (Signed)	1
Double Word (Signed)	2 ^{*2}
Word (Unsigned)	1
Double Word (Unsigned)	2 ^{*2}
Float (Single Precision)	2 ^{*2}
Float (Double Precision)	4
String	17
String (Unicode)	33
Time	2 ^{*2}
Timer, Timer (Unsigned), Long timer, Counter, Counter (Unsigned), Long counter, Retentive Timer, Retentive Timer (Unsigned), Long retentive timer	1
Structure	Total value of the number of applicable words for the structure member data type

*1 When accessing data by specifying the label name only (without element specification) for the array label, specify the number of read points in this table multiplied by the number of array elements.

*2 When reading a single label of "Simple Types" to which a double word device is assigned, the number of read points is 1. It is 2 when reading a label of "Structure" to which a double word device is assigned and when reading multiple labels to which double word devices are assigned at the same time.

How to specify devices

The read device values are stored as follows:

■When specifying bit devices, word devices, and double word devices acquired for one point

Example: Read a total of 6 points of data from M0, D0, D4, CN200, and D10.

· Label setting

Label name	Data type	Device
LABEL1	Bit	M0
LABEL2	Word	D0
LABEL3	Word	D4
LABEL4	Double Word	CN200
LABEL5	Word	D10

· Number of read points: 6

· Reading method

```
int iData[6]
String stLabelList[5] = {"LABEL1", "LABEL2", "LABEL3", "LABEL4", "LABEL5"}
ReadDeviceRandom(stLabelList,6,iData)
```

· Read device values

Upper 2-byte	Lower 2-byte
Not used ('0' is stored.)	M0
	D0
	D4
H of CN200*1	L of CN200*1
H of CN201*1	L of CN201*1
Not used ('0' is stored.)	D10

*1 2 words are read for 1 point. For the Random function, 2 device data is read when CN200 is specified to Double Word type.

■When specifying double word devices to be acquired for 1 point in Double Word type

Example: Read 2 points of data from the devices starting from LZ0.

· Label setting (Data type: Double Word, Number of array elements: 2)

Label name	Data type	Device
LABEL	Double Word (0..1)	LZ0

· Number of read points: 2

· Reading method

```
int iData[2]
ReadDeviceRandom("LABEL",2,iData)
```

· Read device values

Upper 2-byte	Lower 2-byte
H of LZ0	L of LZ0
H of LZ1	L of LZ1

■When specifying character strings

Example: Read 17 points of data from the devices starting from D0.

· Label setting

Label name	Data type	Device
LABEL	String	D0

· Number of read points: 17

· Reading method

```
int iData[17]
ReadDeviceRandom("LABEL",17,iData)
```

Data is not converted to character strings.

· Read device values

Upper 2-byte	Lower 2-byte
Not used ('0' is stored.)	D0
	D1
	⋮
	D15
	D16

■When specifying data types equivalent to 2 words or more for labels

Example: Read a total of 6 points of data by specifying labels of Double Word, Float (Single Precision), and Time types.

· Label setting

Label name	Data type	Device
LABEL1	Double Word	D0
LABEL2	Float (Single Precision)	D100
LABEL3	Time	D200

· Number of read points: 6

· Reading method

```
String stLabelList[3] = {"LABEL1", "LABEL2", "LABEL3" }
int iData[6]
ReadDeviceRandom(stLabelList,6,iData)
```

· Read device values

Upper 2-byte	Lower 2-byte	Applicable label
Not used ('0' is stored.)	D0	LABEL1
	D1	
	D100	LABEL2
	D101	
	D200	LABEL3
	D201	

■When specifying Float (Double Precision) and word devices

Example: Read a total of 5 points of data by specifying labels of Float (Double Precision) and Word types.

· Label setting

Label name	Data type	Device
LABEL1	Float (Double Precision)	D0
LABEL3	Word	D200

· Number of read points: 5

· Reading method

```
String stLabelList[2] = {"LABEL1", "LABEL2"}
int iData[5]
ReadDeviceRandom(stLabelList,5,iData)
```

· Read device values

Upper 2-byte	Lower 2-byte	Applicable label
Not used ('0' is stored.)	D0	LABEL1
	D1	
	D2	
	D3	
	D200	LABEL3

■When specifying array type labels

Example 1: Read 3 points of data from the bit devices (M0 to M2) by specifying array type labels.

· Label setting

Label name	Data type	Device
LABEL	Bit (0..2)	M0

· Number of read points: 3

· Reading method

```
int iData[3]
ReadDeviceRandom("LABEL",3,iData)
```

· Read device values

Upper 2-byte	Lower 2-byte
Not used ('0' is stored.)	M0
	M1
	M2

Example 2: Read 3 points of data from the word devices (D0 to D2) by specifying array type labels.

· Label setting

Label name	Data type	Device
LABEL	Word (0..2)	D0

· Number of read points: 3

· Reading method

```
int iData[3]
ReadDeviceRandom("LABEL",3,iData)
```

· Read device values

Upper 2-byte	Lower 2-byte
Not used ('0' is stored.)	D0
	D1
	D2

Example 3: Read 3 points of data from the double word devices (CN200 to CN202) by specifying array type labels.

· Label setting

Label name	Data type	Device
LABEL	Word (0..2)	CN200

· Number of read points: 3

· Reading method

```
int iData[3]
ReadDeviceRandom("LABEL",3,iData)
```

· Read device values

Upper 2-byte	Lower 2-byte
H of CN200	L of CN200
H of CN201	L of CN201
H of CN202	L of CN202

Example 4: Read 6 points (6 words = 3 double words) of data by specifying array type labels.

· Label setting

Label name	Data type	Device
LABEL	Double Word (0..2)	D0

· Number of read points: 6

· Reading method

```
int iData[6]
ReadDeviceRandom("LABEL",6,iData)
```

· Read device values

Upper 2-byte	Lower 2-byte	Applicable label
Not used ('0' is stored.)	D0	LABEL[0]
	D1	
	D2	LABEL[1]
	D3	
	D4	LABEL[2]
	D5	

Example 5: Read 51 points (D0 to D50) of string type data by specifying array type labels.

· Label setting

Label name	Data type	Device
LABEL	String (0..2)	D0

· Number of read points: 51

· Reading method

```
int iData[51]
ReadDeviceRandom("LABEL",51,iData)
```

The characters need to be converted in a user program because the characters of String type are not converted.

· Read device values

index	Upper 2-byte	Lower 2-byte	Applicable label
[0]	Not used ('0' is stored.)	D0	LABEL[0]
[1]		D1	
⋮		⋮	
[16]		D16	LABEL[1]
[17]		D17	
[18]		D18	LABEL[2]
⋮		⋮	
[33]		D33	
[34]		D34	LABEL[2]
[35]		D35	
⋮		⋮	
[50]		D50	

■When specifying structure type labels

Example 1: Read data by specifying structure type labels.

· Structure setting

Structure name	Member name	Data type
STRUCT	LABEL1	Bit
	LABEL2	Word
	LABEL3	Word

· Label setting

Label name	Device
STRUCT.LABEL1	M0
STRUCT.LABEL2	D0
STRUCT.LABEL3	W0

When accessing by specifying the label name only

· Number of read points: 3 (because the total number of read points for each structure member is 3)

· Reading method

```
int iData[3]
ReadDeviceRandom("STRUCT",3,iData)
```

· Read device values

Upper 2-byte	Lower 2-byte
Not used ('0' is stored.)	M0
	D0
	W0

When accessing by specifying the label name + member name

· Number of read points: 2 (because 2 members of structure, for which each number of read points is 1, are specified)

· Reading method

```
String stLabelList[3] = {"STRUCT.LABEL1","STRUCT.LABEL2"}
int iData[2]
ReadDeviceRandom(stLabelList,2,iData)
```

· Read device values

Upper 2-byte	Lower 2-byte
Not used ('0' is stored.)	M0
	D0

Example 2: Read data by specifying structure type labels.

· Structure setting

Structure name	Member name	Data type
STRUCT	LABEL1	Bit
	LABEL2	Double Word
	LABEL3	Float (Double Precision)
	LABEL4	Word

· Label setting

Label name	Device
STRUCT.LABEL1	D0.0
STRUCT.LABEL2	D1
STRUCT.LABEL3	D3
STRUCT.LABEL4	D7

· Number of read points: 8 (because the total number of read points for each structure member is 8)

· Reading method

```
int iData[8]
ReadDeviceRandom("STRUCT",8,iData)
```

· Read device values

Upper 2-byte	Lower 2-byte	Applicable label
Not used ('0' is stored.)	D0.0	STRUCT.LABEL1
	D1	STRUCT.LABEL2
	D2	
	D3	STRUCT.LABEL3
	D4	
	D5	
	D6	
	D7	STRUCT.LABEL4

■When specifying labels in combination with structure and array

Example: Read data by specifying structure type array and structure array type labels.

· Structure setting

Structure name	Member name	Data type
STRUCT	LABEL1	Bit
	LABEL2	Word (0..1)
	LABEL3	Double Word
	LABEL4	String
	LABEL5	Word

· Label setting

Label name	Device
STRUCT[0].LABEL1	M0
STRUCT[0].LABEL2[0]	D0
STRUCT[0].LABEL2[1]	D1
STRUCT[0].LABEL3	D2
STRUCT[0].LABEL4	D4
STRUCT[0].LABEL5	D21
STRUCT[1].LABEL1	M1
STRUCT[1].LABEL2[0]	D22
STRUCT[1].LABEL2[1]	D23
STRUCT[1].LABEL3	D24
STRUCT[1].LABEL4	D26
STRUCT[1].LABEL5	D43

· Number of read points: 46 (because the total number of read points for each structure is 23, and the number of array elements is 2)

· Reading method

```
int iData[46]
```

```
ReadDeviceRandom("STRUCT",46,iData)
```

or

```
String stLabelList[2] = {"STRUCT[0]","STRUCT[1]"};
```

```
ReadDeviceRandom(stLabelList,46 ,iData)
```

· Read device values

index	Upper 2-byte	Lower 2-byte	Applicable label
[0]	Not used ('0' is stored.)	M0	STRUCT[0].LABEL1
[1]		D0	STRUCT[0].LABEL2[0]
[2]		D1	STRUCT[0].LABEL2[1]
[3]		D2	STRUCT[0].LABEL3
[4]		D3	
[5]		D4	STRUCT[0].LABEL4
⋮		⋮	
[21]		D20	
[22]		D21	
[23]		M1	
[24]		D22	STRUCT[1].LABEL2[0]
[25]		D23	STRUCT[1].LABEL2[1]
[26]		D24	STRUCT[1].LABEL3
[27]		D25	
[28]		D26	STRUCT[1].LABEL4
⋮		⋮	
[44]		D42	
[45]		D43	STRUCT[1].LABEL5

WriteDeviceRandom (Writing devices randomly)

Applicable controls

This function can be used for the following controls.

- 32-bit: DotUtilType
- 64-bit: DotUtilType64

Feature

To write devices randomly.

Format

■ Visual Basic .NET

`IRet = object.WriteDeviceRandom(szLabel, iSize, iData)`

Data type	Argument	Description	Input/Output
Integer	IRet	Returned value	Output
String	szLabel	Label name	Input
Integer	iSize	Number of write points	Input
Integer	iData(n)	Device value to be written	Input

`IRet = object.WriteDeviceRandom(szLabelList, iSize, iData)`

Data type	Argument	Description	Input/Output
Integer	IRet	Returned value	Output
String	szLabelList(n)	Label list	Input
Integer	iSize	Number of write points	Input
Integer	iData(n)	Device value to be written	Input

■ Visual C#

`iRet = object.WriteDeviceRandom(ref szLabel, iSize, ref iData)`

Data type	Argument	Description	Input/Output
int	iRet	Returned value	Output
String	szLabel	Label name	Input
int	iSize	Number of write points	Input
int[n]	iData	Device value to be written	Input

`iRet = object. WriteDeviceRandom(ref szLabelList, iSize, ref iData)`

Data type	Argument	Description	Input/Output
int	iRet	Returned value	Output
System.String[n]	szLabelList	Label list	Input
int	iSize	Number of write points	Input
int[n]	iData	Device value to be written	Input

Returned value

Normal termination: '0' is returned.

Abnormal termination: Any value other than '0' is returned. (📖 Page 616 ERROR CODES)

Description

- Data of a device group specified for the label name szLabel (szLabelList) is written for the size of iSize.
- The device values to be written are stored in iData.
- For iData, reserve arrays for more than the amount specified for iSize.

Point

- The maximum number of write points is 0x7FFFFFFF.
- For the number of write points, specify the number of words which applies to the data type specified for the label name.
For the device values to be written, reserve a memory area for the number of points specified for the number of write points.
If the memory area is not reserved, a critical error (an application error or the like) may occur.
- When a device which corresponds to the label name does not exist, an error occurs and data cannot be written.
When any one of devices which correspond to multiple label names does not exist, an error occurs and data cannot be written.
- Digit specified bit device and index setting cannot be used.
- An error is returned when a Q motion CPU is accessed.
- When using system label Ver.2, the data type defined in the Label Utility of MX Component and the data type managed by MELSOFT Navigator is required to match.
If the data type does not match, the read data length may wrong, or when multiple labels are specified, the correspondence of the array of the read device value and label name may not match.
- Up to five levels of a structure can be defined. Note that only one-level structure can be used for a static system label.
- If the function is run against the safety device in the RnSF CPU safety mode, an error code 0x010A42A5" (an operation that can not be carried out in safety mode was performed) is returned.

How to specify labels

The following describes how to specify labels.

For the system label name, the following type class, array element, and label data type can be specified.

When specifying a system label name in a program, use the following formats.

○: Available, ×: Not available, —: Not applicable

Label definition in Label Utility		Label specification method in a program			
Type class	Array element	Label name format	Availability	Data type	
Simple Types	None	System label name	○	Integer/int (32-bit value)	
		System label name	○*1	Integer/int (32-bit value)	
	Element (One Dimension)	System label name [n1]	○	Integer/int (32-bit value)	
		System label name	○*1	Integer/int (32-bit value)	
	Element (Two Dimensions)	System label name [n1] [n2]	○	Integer/int (32-bit value)	
		System label name	○*1	Integer/int (32-bit value)	
	Element (Three Dimensions)	System label name [n1] [n2] [n3]	○	Integer/int (32-bit value)	
		System label name	○*1	Integer/int (32-bit value)	
	Structure	None	System label name	○*1	Integer/int (32-bit value)
			System label name.structure member name	○	Integer/int (32-bit value)
		Element (One Dimension)	System label name	○*1	Integer/int (32-bit value)
			System label name [n1]	○*1	Integer/int (32-bit value)
System label name [n1].structure member name			○	Integer/int (32-bit value)	
Element (Two Dimensions)		System label name	○*1	Integer/int (32-bit value)	
		System label name [n1] [n2]	○*1	Integer/int (32-bit value)	
		System label name [n1] [n2].structure member name	○	Integer/int (32-bit value)	
Element (Three Dimensions)		System label name	○*1	Integer/int (32-bit value)	
		System label name [n1] [n2] [n3]	○*1	Integer/int (32-bit value)	
		System label name [n1] [n2] [n3].structure member name	○	Integer/int (32-bit value)	

*1 The entire data of arrays and structures is written.

■Data type

The following data type can be specified for a label.

Bit, Word (Signed), Double Word (Signed), Word (Unsigned), Double Word (Unsigned), Float (Single Precision), Float (Double Precision), String^{*1}, String (Unicode)^{*2,4}, Time, Timer^{*3}, Timer (Unsigned)^{*3}, Long timer^{*3,4}, Counter^{*3}, Counter (Unsigned)^{*3}, Long counter^{*3,4}, Retentive Timer^{*3}, Retentive Timer (Unsigned)^{*3}, Long retentive timer^{*3,4}

*1 Can be specified up to 32 single-byte characters + NULL.

*2 Can be specified up to 32 characters + NULL.

*3 When accessing the device using a label of system label Ver.2, the label of system label Ver.2 which is corresponding to a device for contact/coil/present value is required to be defined.

*4 Cannot be used for system label Ver.1.

■Number of write points

For the number of write points, specify the sum of the following values correspond to the elements according to the label data type.

Label data type	Number of write points to be specified ^{*1}
Bit	1
Word (Signed)	1
Double Word (Signed)	2 ^{*2}
Word (Unsigned)	1
Double Word (Unsigned)	2 ^{*2}
Float (Single Precision)	2 ^{*2}
Float (Double Precision)	4
String	17
String (Unicode)	33
Time	2 ^{*2}
Timer, Timer (Unsigned), Long timer, Counter, Counter (Unsigned), Long counter, Retentive Timer, Retentive Timer (Unsigned), Long retentive timer	1
Structure	Total value of the number of applicable words for the structure member data type

*1 When accessing data by specifying the label name only (without element specification) for the array label, specify the number of write points in this table multiplied by the number of array elements.

*2 When writing a single label of "Simple Types" to which a double word device is assigned, the number of write points is 1. It is 2 when writing a label of "Structure" to which a double word device is assigned and when writing multiple labels to which double word devices are assigned at the same time.

How to specify devices

Specify the device values to be written as follows:

■When specifying bit devices, word devices, and double word devices acquired for one point

Example: Write 1 point of data to M0, D0, D4, CN200 and D10.

· Label setting

Label name	Data type	Device
LABEL1	Bit	M0
LABEL2	Word	D0
LABEL3	Word	D4
LABEL4	Double Word	CN200
LABEL5	Word	D10

· Number of write points: 6

· Writing method

```
stLabelList = {"LABEL1", "LABEL2", "LABEL3", "LABEL4", "LABEL5"}
int iData[6]
WriteDeviceRandom(stLabelList,6,iData)
```

· Device values to be written

Upper 2-byte	Lower 2-byte
Not used ('0' is stored.)	M0
	D0
	D4
H of CN200 ^{*1}	L of CN200 ^{*1}
H of CN201 ^{*1}	L of CN201 ^{*1}
Not used ('0' is stored.)	D10

*1 2 words are written for 1 point. For the Random function, 2 device data is written when CN200 is specified to Double Word type.

■When specifying double word devices to be acquired for 1 point in Double Word type

Example: Write 2 points of data to the devices starting from LZ0.

· Label setting (Data type: Double Word, Number of array elements: 2)

Label name	Data type	Device
LABEL	Double Word (0..1)	LZ0

· Number of write points: 2

· Writing method

```
int iData[2]
WriteDeviceRandom("LABEL",2,iData)
```

· Device values to be written

Upper 2-byte	Lower 2-byte
H of LZ0	L of LZ0
H of LZ1	L of LZ1

■When specifying character strings

Example: Write 17 points of data to the devices starting from D0.

· Label setting

Label name	Data type	Device
LABEL	String	D0

· Number of write points: 17

· Writing method

```
int iData[17]
WriteDeviceRandom("LABEL",17,iData)
```

Data is not converted to character strings.

· Device values to be written

Upper 2-byte	Lower 2-byte
Not used ('0' is stored.)	D0
	D1
	D2
	⋮
	D14
	D15
	D16

■When specifying data types equivalent to 2 words or more for labels

Example: Write data by specifying labels of Double Word, Float (Single Precision), and Time types.

· Label setting

Label name	Data type	Device
LABEL1	Double Word	D0
LABEL2	Float (Single Precision)	D100
LABEL3	Time	D200

· Number of write points: 6

· Writing method

```
String stLabelList[3] = {"LABEL1", "LABEL2", "LABEL3"}
int iData[6]
WriteDeviceRandom(stLabelList,6,iData)
```

· Device values to be written

Upper 2-byte	Lower 2-byte	Applicable label
Not used ('0' is stored.)	D0	LABEL1
	D1	
	D100	LABEL2
	D101	
	D200	LABEL3
	D201	

■When specifying Float (Double Precision) and word devices

Example: Write a total of 5 points of data by specifying Float (Double Precision) and a word device.

· Label setting

Label name	Data type	Device
LABEL1	Float (Double Precision)	D0
LABEL3	Word	D200

· Number of write points: 5

· Writing method

```
String stLabelList[2] = {"LABEL1","LABEL3"}
int iData[5]
WriteDeviceRandom(stLabelList,5,iData)
```

· Device values to be written

Upper 2-byte	Lower 2-byte	Applicable label
Not used ('0' is stored.)	D0	LABEL1
	D1	
	D2	
	D3	
	D200	LABEL3

■When specifying array type labels

Example 1: Write 3 points of data to the bit devices (M0 to M2) by specifying array type labels.

· Label setting

Label name	Data type	Device
LABEL	Bit (0..2)	M0

· Number of write points: 3

· Writing method

```
int iData[3]
WriteDeviceRandom("LABEL",3,iData)
```

· Device values to be written

Upper 2-byte	Lower 2-byte
Not used ('0' is stored.)	M0
	M1
	M2

Example 2: Write 3 points of data to the word devices (D0 to D2) by specifying array type labels.

· Label setting

Label name	Data type	Device
LABEL	Word (0..2)	D0

· Number of write points: 3

· Writing method

```
int iData[3]
WriteDeviceRandom("LABEL",3,iData)
```

· Device values to be written

Upper 2-byte	Lower 2-byte
Not used ('0' is stored.)	D0
	D1
	D2

Example 3: Write 3 points of data to the double word devices (CN200 to CN202) by specifying array type labels.

· Label setting

Label name	Data type	Device
LABEL	Word (0..2)	CN200

· Number of write points: 3

· Writing method

```
int iData[3]
WriteDeviceRandom("LABEL",3,iData)
```

The characters need to be converted in a user program because the characters of String type are not converted.

· Device values to be written

Upper 2-byte	Lower 2-byte
H of CN200	L of CN200
H of CN201	L of CN201
H of CN202	L of CN202

Example 4: Write 6 points (6 words = 3 double words) of data by specifying array type labels.

· Label setting

Label name	Data type	Device
LABEL	Double Word (0..2)	D0

· Number of write points: 6

· Writing method

```
int iData[6]
WriteDeviceRandom("LABEL",6,iData)
```

· Device values to be written

Upper 2-byte	Lower 2-byte	Applicable label
Not used ('0' is stored.)	D0	LABEL[0]
	D1	
	D2	LABEL[1]
	D3	
	D4	LABEL[2]
	D5	

Example 5: Read 51 points (D0 to D50) of string type data by specifying array type labels.

· Label setting

Label name	Data type	Device
LABEL	String (0..2)	D0

· Number of write points: 51

· Writing method

```
int iData[51]
WriteDeviceRandom("LABEL",51,iData)
```

The characters need to be converted in a user program because the characters of String type are not converted.

· Device values to be written

index	Upper 2-byte	Lower 2-byte	Applicable label
[0]	Not used ('0' is stored.)	D0	LABEL[0]
[1]		D0	
⋮		⋮	
[16]		D16	LABEL[1]
[17]		D17	
[18]		D18	LABEL[2]
⋮		⋮	
[33]		D33	
[34]		D34	LABEL[2]
[35]		D35	
⋮		⋮	
[50]		D50	

■ When specifying structure type labels

Example 1: Write data by specifying structure type labels.

· Structure setting

Structure name	Member name	Data type
STRUCT	LABEL1	Bit
	LABEL2	Word
	LABEL3	Word

· Label setting

Label name	Device
STRUCT.LABEL1	M0
STRUCT.LABEL2	D0
STRUCT.LABEL3	W0

When accessing by specifying the label name only

· Number of write points: 3 (because the total number of write points for each structure member is 3)

· Writing method

```
int iData[3]
WriteDeviceRandom("STRUCT",3,iData)
```

· Device values to be written

Upper 2-byte	Lower 2-byte
Not used ('0' is stored.)	M0
	D0
	W0

When accessing by specifying the label name + member name

· Number of write points: 2 (because 2 members of structure, for which each number of write points is 1, are specified)

· Writing method

```
String stLabelList[2] = {"STRUCT.LABEL1","STRUCT.LABEL2"}
int iData[2]
WriteDeviceRandom(stLabelList,2,iData)
```

· Device values to be written

Upper 2-byte	Lower 2-byte
Not used ('0' is stored.)	M0
	D0

Example 2: Write data by specifying structure type labels.

· Structure setting

Structure name	Member name	Data type
STRUCT	LABEL1	Bit
	LABEL2	Double Word
	LABEL3	Float (Double Precision)
	LABEL4	Word

· Label setting

Label name	Device
STRUCT.LABEL1	D0.0
STRUCT.LABEL2	D1
STRUCT.LABEL3	D3
STRUCT.LABEL4	D7

· Number of write points: 8 (because the total number of write points for each structure member is 8)

· Writing method

```
int iData[8]
WriteDeviceRandom("STRUCT",8,iData)
```

· Device values to be written

Upper 2-byte	Lower 2-byte	Applicable label
Not used ('0' is stored.)	D0.0	STRUCT.LABEL1
	D1	STRUCT.LABEL2
	D2	
	D3	STRUCT.LABEL3
	D4	
	D5	
	D6	
	D7	STRUCT.LABEL4

■When specifying labels in combination with structure and array

Example: Write data by specifying structure type array and structure array type labels.

· Structure setting

Structure name	Member name	Data type
STRUCT	LABEL1	Bit
	LABEL2	Word (0..1)
	LABEL3	Double Word
	LABEL4	String
	LABEL5	Word

· Label setting

Label name	Device
STRUCT[0].LABEL1	M0
STRUCT[0].LABEL2[0]	D0
STRUCT[0].LABEL2[1]	D1
STRUCT[0].LABEL3	D2
STRUCT[0].LABEL4	D4
STRUCT[0].LABEL5	D21
STRUCT[1].LABEL1	M1
STRUCT[1].LABEL2[0]	D22
STRUCT[1].LABEL2[1]	D23
STRUCT[1].LABEL3	D24
STRUCT[1].LABEL4	D26
STRUCT[1].LABEL5	D43

· Number of write points: 46 (because the total number of write points for each structure is 23, and the number of array elements is 2)

· Writing method

```
int iData[46]
```

```
WriteDeviceRandom("STRUCT",46,iData)
```

or

```
String stLabelList[2] = {"STRUCT[0]","STRUCT[1]"};
```

```
WriteDeviceRandom(stLabelList,46 ,iData)
```

· Device values to be written

index	Upper 2-byte	Lower 2-byte	Applicable label	
[0]	Not used ('0' is stored.)	M0	STRUCT[0].LABEL1	
[1]		D0	STRUCT[0].LABEL2[0]	
[2]		D1	STRUCT[0].LABEL2[1]	
[3]		D2	STRUCT[0].LABEL3	
[4]		D3		
[5]	⋮	D4	STRUCT[0].LABEL4	
[21]		D20		
[22]		D21		STRUCT[0].LABEL5
[23]		M1		STRUCT[1].LABEL1
[24]		D22		STRUCT[1].LABEL2[0]
[25]	D23	STRUCT[1].LABEL2[1]		
[26]	D24	STRUCT[1].LABEL3		
[27]	D25			
[28]	⋮	D26	STRUCT[1].LABEL4	
[44]		D42		
[45]		D43		STRUCT[1].LABEL5

SetDevice (Setting device data)

Applicable controls

This function can be used for the following controls.

- 32-bit: DotUtilType
- 64-bit: DotUtilType64

Feature

To set one point of device.

Format

■Visual Basic .NET

iRet = object.SetDevice(szLabel, iData)

Data type	Argument	Description	Input/Output
Integer	iRet	Returned value	Output
String	szLabel	Label name	Input
Integer	iData	Device value to be written	Input

■Visual C#

iRet = object.SetDevice(ref szLabel, iData)

Data type	Argument	Description	Input/Output
int	iRet	Returned value	Output
String	szLabel	Label name	Input
int	iData	Device value to be written	Input

Returned value

Normal termination: '0' is returned.

Abnormal termination: Any value other than '0' is returned. (📖 Page 616 ERROR CODES)

Description

- The values of iData (device values to be written) are written to 1 point of device specified for szLabel (label name).
- When specifying bit devices, the least significant bit of the iData (device values to be written) is valid.

Point

- When a device which corresponds to the label name does not exist, an error occurs and data cannot be written.
- When specifying a double word device, 2-byte is written in Double Word type.
When specifying other than a double word device for a data type equivalent to 2 words or more, this function writes to the data area of lower 1 word (2-byte). (The values of upper 2-byte is indefinite.)
Use the WriteDeviceRandom function to write devices equivalent to 2 words or more.
- When using a label memory without specifying a device in system label Ver.2, the data is handled as 2 points of word.
- Digit specified bit device and index setting cannot be used.
- If the function is run against the safety device in the RnSF CPU safety mode, an error code 0x010A42A5" (an operation that can not be carried out in safety mode was performed) is returned.

How to specify labels

The following describes how to specify labels.

For the system label name, the following type class, array element, and label data type can be specified.

When specifying a system label name in a program, use the following formats.

○: Available, ×: Not available, —: Not applicable

Label definition in Label Utility		Label specification method in a program			
Type class	Array element	Label name format	Availability	Data type	
Simple Types	None	System label name	○	Integer/int (32-bit value)	
		System label name [n1]	○	Integer/int (32-bit value)	
	Element (One Dimension)	System label name	×	—	
		System label name [n1]	○	Integer/int (32-bit value)	
	Element (Two Dimensions)	System label name	×	—	
		System label name [n1] [n2]	○	Integer/int (32-bit value)	
	Element (Three Dimensions)	System label name	×	—	
		System label name [n1] [n2] [n3]	○	Integer/int (32-bit value)	
	Structure	None	System label name	×	—
			System label name.structure member name	○	Integer/int (32-bit value)
		Element (One Dimension)	System label name	×	—
			System label name [n1]	×	—
System label name [n1].structure member name			○	Integer/int (32-bit value)	
Element (Two Dimensions)		System label name	×	—	
		System label name [n1] [n2]	×	—	
		System label name [n1] [n2].structure member name	○	Integer/int (32-bit value)	
Element (Three Dimensions)		System label name	×	—	
		System label name [n1] [n2] [n3]	×	—	
		System label name [n1] [n2] [n3].structure member name	○	Integer/int (32-bit value)	

■Data type

The following data type can be specified for a label.

Bit, Word (Signed), Double Word (Signed)^{*1}, Word (Unsigned), Double Word (Unsigned)^{*1}, Float (Single Precision)^{*1}, Float (Double Precision)^{*1}, String^{*1}, String (Unicode)^{*1,2}, Time^{*1}, Timer, Timer (Unsigned), Long timer^{*2}, Counter, Counter (Unsigned), Long counter^{*2}, Retentive Timer, Retentive Timer (Unsigned), Long retentive timer^{*2}

*1 Only the lower 2-byte of start device can be written. Use the WriteDeviceRandom function to write devices equivalent to 2 words or more.

*2 Cannot be used for system label Ver.1.

How to specify devices

Specify the device values to be written as follows:

■When specifying bit devices

Example: Write data to M0.

· Label setting

szLabel	Label name	Data type	Device
[0]	LABEL	Bit	M0

· Number of write points: 1

· Device values to be written

Upper 2-byte	Lower 2-byte	Applicable label
Not used ('0' is stored.)	M0	LABEL

■When specifying word devices

Example: Write data to D0.

· Label setting

szLabel	Label name	Data type	Device
[0]	LABEL	Word	D0

· Number of write points: 1

· Device values to be written

Upper 2-byte	Lower 2-byte	Applicable label
Not used ('0' is stored.)	D0	LABEL

■When specifying double word devices (ones to which data of 2 words can be written for 1 point)

Example: Write data to CN200.

For CN200 and later devices of FXCPU, 4-byte is written.

· Label setting

szLabel	Label name	Data type	Device
[0]	LABEL	Double Word	CN200

· Number of write points: 1

· Device values to be written

Upper 2-byte	Lower 2-byte	Applicable label
H of CN200	L of CN200	LABEL

■When specifying double word devices (ones to which data of 2 words cannot be written for 1 point)

Example: Write data by specifying D0.

· Label setting

szLabel	Label name	Data type	Device
[0]	LABELW	Double Word	D0

· Number of write points: 1

· Device values to be written

Upper 2-byte	Lower 2-byte	Applicable label
Not used.	D0*1	LABELW

*1 Data is written in order from the lower 2-byte. For system label Ver.2, an indefinite value is written to the upper 2-byte.

■When specifying data types equivalent to 2 words or more for labels

Example: Write data by specifying a string type label.

· Label setting

szLabel	Label name	Data type	Device
[0]	LABEL	String	D0

· Number of write points: 1

· Device values to be written

Upper 2-byte	Lower 2-byte	Applicable label
Not used.	D0*1	LABEL

*1 Data is written in order from the lower 2-byte. For system label Ver.2, an indefinite value is written to the upper 2-byte.

GetDevice (Acquiring device data)

Applicable controls

This function can be used for the following controls.

- 32-bit: DotUtilType
- 64-bit: DotUtilType64

Feature

To acquire one point of device.

Format

■Visual Basic .NET

`IRet = object.GetDevice(szLabel, IData)`

Data type	Argument	Description	Input/Output
Integer	IRet	Returned value	Output
String	szLabel	Label name	Input
Integer	iData	Read device value	Output

■Visual C#

`iRet = object.GetDevice(ref szLabel, ref iData)`

Data type	Argument	Description	Input/Output
int	iRet	Returned value	Output
String	szLabel	Label name	Input
int	iData	Read device value	Output

Returned value

Normal termination: '0' is returned.

Abnormal termination: Any value other than '0' is returned. (🔗 Page 616 ERROR CODES)

Description

- One point of device data specified for szLabel (label name) is stored in iData (read device values).

Point

- When a device which corresponds to the label name does not exist, an error occurs and data cannot be read.
- For the read device values, reserve 4 bytes of memory area.
If the memory area is not reserved, a critical error (an application error or the like) may occur.
- Digit specified bit device and index setting cannot be used.

How to specify labels

The following describes how to specify labels.

For the system label name, the following type class, array element, and label data type can be specified.

When specifying a system label name in a program, use the following formats.

○: Available, ×: Not available, —: Not applicable

Label definition in Label Utility		Label specification method in a program			
Type class	Array element	Label name format	Availability	Data type	
Simple Types	None	System label name	○	Integer/int (32-bit value)	
		System label name [n1]	○	Integer/int (32-bit value)	
	Element (One Dimension)	System label name	×	—	
		System label name [n1]	○	Integer/int (32-bit value)	
	Element (Two Dimensions)	System label name	×	—	
		System label name [n1] [n2]	○	Integer/int (32-bit value)	
	Element (Three Dimensions)	System label name	×	—	
		System label name [n1] [n2] [n3]	○	Integer/int (32-bit value)	
	Structure	None	System label name	×	—
			System label name.structure member name	○	Integer/int (32-bit value)
		Element (One Dimension)	System label name	×	—
			System label name [n1]	×	—
System label name [n1].structure member name			○	Integer/int (32-bit value)	
Element (Two Dimensions)		System label name	×	—	
		System label name [n1] [n2]	×	—	
		System label name [n1] [n2].structure member name	○	Integer/int (32-bit value)	
Element (Three Dimensions)		System label name	×	—	
		System label name [n1] [n2] [n3]	×	—	
		System label name [n1] [n2] [n3].structure member name	○	Integer/int (32-bit value)	

■Data type

The following data type can be specified for a label.

Bit, Word (Signed), Double Word (Signed)^{*1}, Word (Unsigned), Double Word (Unsigned)^{*1}, Float (Single Precision)^{*1}, Float (Double Precision)^{*1}, String^{*1}, String (Unicode)^{*1,2}, Time^{*1}, Timer, Timer (Unsigned), Long timer^{*2}, Counter, Counter (Unsigned), Long counter^{*2}, Retentive Timer, Retentive Timer (Unsigned), Long retentive timer^{*2}

*1 Only the lower 2-byte of start device can be read.

Use the ReadDeviceRandom function to read 2 words or more of devices.

However, the upper 2-byte is also read in Double Word type when specifying a double word device that can be acquired for 1 point.

*2 Cannot be used for system label Ver.1.

How to specify devices

The read device values are stored as follows:

■When specifying bit devices

Example: Read data from M0.

· Label setting

szLabel	Label name	Data type	Device
[0]	LABEL	Bit	M0

· Number of read points: 1

· Read device values

Upper 2-byte	Lower 2-byte	Applicable label
Not used ('0' is stored.)	M0*1	LABEL

*1 The target device from which data is read is only one point of "M0," and '0' or '1' is stored as the device value.

■When specifying word devices

Example: Read data from D0.

· Label setting

szLabel	Label name	Data type	Device
[0]	LABEL	Word	D0

· Number of read points: 1

· Read device values

Upper 2-byte	Lower 2-byte	Applicable label
Not used ('0' is stored.)	D0	LABEL

■When specifying double word devices (ones from which data of 2 words can be read for 1 point)

Example: Read data from CN200.

· Label setting

szLabel	Label name	Data type	Device
[0]	LABEL	Double Word	CN200

· Number of read points: 1

· Read device values

Upper 2-byte	Lower 2-byte	Applicable label
H of CN200	L of CN200*1	LABEL

*1 For CN200 and later devices of FXCPU, 4-byte is read.

■When specifying double word devices (ones from which data of 2 words cannot be read for 1 point)

Example: Read data by specifying D0.

· Label setting

szLabel	Label name	Data type	Device
[0]	LABELW	Double Word	D0

· Number of read points: 1

· Read device values

Upper 2-byte	Lower 2-byte	Applicable label
Not used ('0' is stored.)	D0*1	LABELW

*1 Data is read in order from the lower 2 bytes.

■When specifying data types equivalent to 2 words or more for labels

Example: Read data by specifying a string type label.

· Label setting

szLabel	Label name	Data type	Device
[0]	LABEL	String	D0

· Number of read points: 1

· Read device values

Upper 2-byte	Lower 2-byte	Applicable label
Not used ('0' is stored.)	D0 ^{*1}	LABEL

*1 Data is read in order from the lower 2 bytes.

ReadBuffer (Reading data from buffer memory)

Applicable controls

This function can be used for the following controls.

- 32-bit: DotUtilType
- 64-bit: DotUtilType64

Feature

To read data from the buffer memory of an intelligent function module.

Format

■Visual Basic .NET

`IRet = object.ReadBuffer(iStartIO, iAddress, iReadSize, sData)`

Data type	Argument	Description	Input/Output
Integer	IRet	Returned value	Output
Integer	iStartIO	Start I/O number of module from which values are read	Input
Integer	iAddress	Buffer memory address	Input
Integer	iReadSize	Read size	Input
Short	sData(n)	Values read from buffer memory	Output

■Visual C#

`iRet = object.ReadBuffer(iStartIO, iAddress, iReadSize, sData)`

Data type	Argument	Description	Input/Output
int	iRet	Returned value	Output
int	iStartIO	Start I/O number of module from which values are read	Input
int	iAddress	Buffer memory address	Input
int	iReadSize	Read size	Input
short[n]	sData	Values read from buffer memory	Output

Returned value

Normal termination: '0' is returned.

Abnormal termination: Any value other than '0' is returned. ([Page 616 ERROR CODES](#))

Description

- For the start I/O number of the module specified for iStartIO, specify the value of the actual start I/O number divided by 16. For an FX5CPU, specify the module number on the target station side.
- Buffer values of buffer memory address specified for iAddress of an intelligent function module of the start I/O number specified for iStartIO are read for the size of iReadSize.
- When performing CPU COM communication with FXCPU as a connected station or GX Simulator2 communication, specify the block number (0 to 7) of a special expansion device for the start I/O number of a module and 0 to 32767 for the buffer memory address.
- For sData, reserve arrays for more than the amount specified for iReadSize.

Point

- An error is returned when an R motion CPU or a Q motion CPU is accessed.
- For iData, reserve a memory area for the number of points specified for iReadSize. If the memory area is not reserved, a critical error (an application error or the like) may occur.
- When reading data from buffer memory (ReadBuffer) on QCPU (Q mode), the read operation can only be performed on the Q series-dedicated modules.
Furthermore, data cannot be read from the shared memory of QCPU (Q mode).

Applicable communication routes

Refer to the following:

 Page 446 Applicable communication routes

WriteBuffer (Writing data to buffer memory)

Applicable controls

This function can be used for the following controls.

- 32-bit: DotUtilType
- 64-bit: DotUtilType64

Feature

To write data to the buffer memory of an intelligent function module.

Format

■Visual Basic .NET

`IRet = object.WriteBuffer(iStartIO, iAddress, iWriteSize, sData)`

Data type	Argument	Description	Input/Output
Integer	IRet	Returned value	Output
Integer	iStartIO	Start I/O number of module to which values are written	Input
Integer	iAddress	Buffer memory address	Input
Integer	iWriteSize	Write size	Input
Short	sData(n)	Values written from buffer memory	Input

■Visual C#

`iRet = object. WriteBuffer (iStartIO, iAddress, iWriteSize, sData)`

Data type	Argument	Description	Input/Output
int	iRet	Returned value	Output
int	iStartIO	Start I/O number of module to which values are written	Input
int	iAddress	Buffer memory address	Input
int	iWriteSize	Write size	Input
short[n]	sData	Values written from buffer memory	Input

Returned value

Normal termination: '0' is returned.

Abnormal termination: Any value other than '0' is returned. ([Page 616 ERROR CODES](#))

Description

- For the start I/O number of the module specified for iStartIO, specify the value of the actual start I/O number divided by 16. For an FX5CPU, specify the module number on the target station side.
- Buffer values of buffer memory address specified for iAddress of an intelligent function module of the start I/O number specified for iStartIO are written for the size of iWriteSize.
- When performing CPU COM communication with FXCPU as a connected station or GX Simulator2 communication, specify the block number (0 to 7) of a special expansion device for the start I/O number of a module and 0 to 32767 for the buffer memory address.
- For sData, reserve arrays for more than the amount specified for iWriteSize.

Point

- An error is returned when an R motion CPU or a Q motion CPU is accessed.
- For sData, reserve a memory area for the number of points specified for iReadSize. If the memory area is not reserved, a critical error (an application error or the like) may occur.
- When reading data from buffer memory (ReadBuffer) on QCPU (Q mode), the read operation can only be performed on the Q series-dedicated modules.
Furthermore, data cannot be read from the shared memory of QCPU (Q mode).

Applicable communication routes

Refer to the following:

 Page 450 Applicable communication routes

GetClockData (Reading clock data)

Applicable controls

This function can be used for the following controls.

- 32-bit: DotUtilType
- 64-bit: DotUtilType64

Feature

To read time from the clock data of a CPU module.

Format

■Visual Basic .NET

`IRet = object.GetClockData(sYear, sMonth, sDay, sDayOfWeek, sHour, sMinute, sSecond)`

Data type	Argument	Description	Input/Output
Integer	IRet	Returned value	Output
Short	sYear	Read year value	Output
Short	sMonth	Read month value	Output
Short	sDay	Read day value	Output
Short	sDayOfWeek	Read day-of-week value	Output
Short	sHour	Read hour value	Output
Short	sMinute	Read minute value	Output
Short	sSecond	Read second value	Output

■Visual C#

`hResult = object.GetClockData`

(`ref sYear, ref sMonth, ref sDay, ref sDayOfWeek, ref sHour, ref sMinute, ref sSecond`)

Data type	Argument	Description	Input/Output
int	iRet	Returned value	Output
short	sYear	Read year value	Output
short	sMonth	Read month value	Output
short	sDay	Read day value	Output
short	sDayOfWeek	Read day-of-week value	Output
short	sHour	Read hour value	Output
short	sMinute	Read minute value	Output
short	sSecond	Read second value	Output

Returned value

Normal termination: '0' is returned.

Abnormal termination: Any value other than '0' is returned. ( Page 616 ERROR CODES)

Description

- An error is returned when the correct clock data is not set to the CPU module.
- The values stored in sYear are: the four digits of year for RCPU and QCPU (Q mode) and the last two digits of year for any other CPUs.

Note that the applicable years for RCPU and QCPU (Q mode) are from 1980 to 2079.

- The values to be stored in sDayOfWeek are as follows:

Value	Day of Week
0	Sunday
1	Monday
2	Tuesday
3	Wednesday
4	Thursday
5	Friday
6	Saturday

Point

- An error is returned when an R motion CPU or a Q motion CPU is accessed.
- The clock data can be read on FXCPUs with the built-in clock.
For FXCPUs without the built-in clock, an error is returned.
- Note that an error of transfer time may occur in clock setting.
- For the availability of communication routes, refer to the table in  Page 452 GetClockData (Reading clock data).

Applicable communication routes

Refer to the following:

 Page 454 Applicable communication routes

SetClockData (Writing clock data)

Applicable controls

This function can be used for the following controls.

- 32-bit: DotUtilType
- 64-bit: DotUtilType64

Feature

To write time to the clock data of a CPU module.

Format

■Visual Basic .NET

`IRet = object.SetClockData(sYear, sMonth, sDay, sDayOfWeek, sHour, sMinute, sSecond)`

Data type	Argument	Description	Input/Output
Integer	IRet	Returned value	Output
Short	sYear	Year value to be written	Input
Short	sMonth	Month value to be written	Input
Short	sDay	Day value to be written	Input
Short	sDayOfWeek	Day-of-week value to be written	Input
Short	sHour	Hour value to be written	Input
Short	sMinute	Minute value to be written	Input
Short	sSecond	Second value to be written	Input

■Visual C#

`IRet = object.SetClockData(sYear, sMonth, sDay, sDayOfWeek, sHour, sMinute, sSecond)`

Data type	Argument	Description	Input/Output
int	iRet	Returned value	Output
short	sYear	Year value to be written	Input
short	sMonth	Month value to be written	Input
short	sDay	Day value to be written	Input
short	sDayOfWeek	Day-of-week value to be written	Input
short	sHour	Hour value to be written	Input
short	sMinute	Minute value to be written	Input
short	sSecond	Second value to be written	Input

Returned value

Normal termination: '0' is returned.

Abnormal termination: Any value other than '0' is returned. ([📖 Page 616 ERROR CODES](#))

Description

- An error is returned when the clock data to be set is not the correct value.
- The applicable values to be specified for sYear are: the four digits of year for RCPU and QCPU (Q mode) and the last two digits of year for any other CPUs.

Note that the applicable years for RCPU and QCPU (Q mode) are from 1980 to 2079.

An error occurs when four digits are set to a CPU other than RCPU and QCPU (Q mode).

- The values to be specified for sDayOfWeek are as follows:

Value	Day of Week
0	Sunday
1	Monday
2	Tuesday
3	Wednesday
4	Thursday
5	Friday
6	Saturday

Point

- An error is returned when an R motion CPU or a Q motion CPU is accessed.
- The clock data can be read on FXCPUs with the built-in clock.
For FXCPUs without the built-in clock, an error is returned.
- Note that an error of transfer time may occur in clock setting.
- For the availability of communication routes, refer to the table in  Page 456 SetClockData (Writing clock data).
- An error is returned when the own board is accessed.

Applicable communication routes

Refer to the following:

 Page 458 Applicable communication routes

GetCpuType (Reading CPU module model)

Applicable controls

This function can be used for the following controls.

- 32-bit: DotUtilType
- 64-bit: DotUtilType64

Feature

To read the model character string and the model code (hexadecimal) of CPU module, network board, and GOT.

Format

■ Visual Basic .NET

`iRet = object.GetCpuType(szCpuName, iCpuType)`

Data type	Argument	Description	Input/Output
Integer	iRet	Returned value	Output
String	szCpuName	Model character string	Output
Integer	iCpuType	Model code	Output

■ Visual C#

`iRet = object.GetCpuType(ref szCpuName, ref iCpuType)`

Data type	Argument	Description	Input/Output
int	iRet	Returned value	Output
String	szCpuName	Model character string	Output
int	iCpuType	Model code	Output

Returned value

Normal termination: '0' is returned.

Abnormal termination: Any value other than '0' is returned. ( Page 616 ERROR CODES)

Description

- The model character and the model code (hexadecimal) of the communication target CPU module are stored in `szCpuName` and `iCpuType` respectively.
- The model character string is returned in UNICODE.
- The model code is the value of a module (hexadecimal).

SetCpuStatus (Remote control)

Applicable controls

This function can be used for the following controls.

- 32-bit: DotUtilType
- 64-bit: DotUtilType64

Feature

To perform a remote operation of CPU module.

Format

■ Visual Basic .NET

IRet = object.SetCpuStatus(IOperation)

Data type	Argument	Description	Input/Output
Integer	IRet	Returned value	Output
Integer	IOperation	Remote RUN/STOP/PAUSE/RESET	Input

■ Visual C#

iRet = object.SetCpuStatus (iOperation)

Data type	Argument	Description	Input/Output
int	iRet	Returned value	Output
int	iOperation	Remote RUN/STOP/PAUSE/RESET	Input

Returned value

Normal termination: '0' is returned.

Abnormal termination: Any value other than '0' is returned. ( Page 616 ERROR CODES)

Description

The operation specified for iOperation is performed.

An error occurs when a value other than the following values is specified.

Value	Operation
0	Remote RUN
1	Remote STOP
2	Remote PAUSE
3	Remote RESET

Point

- Since FXCPUs do not feature the PAUSE switch as CPU modules, an error is returned when remote pause is specified in SetCpuStatus.
- An error is returned when a Q motion CPU is accessed and PAUSE is specified.
- An error is returned when an R motion CPU is accessed.
- For the availability of communication routes, refer to the table in  Page 462 SetCpuStatus (Remote control).
- Remote RESET can be used only for supported models of QCPU, LCPU, RCPU, LHCPU, and FX5CPU.

Applicable communication routes

Refer to the following:

 Page 463 Applicable communication routes

EntryDeviceStatus (Registering devices for status monitoring)

Applicable controls

This function can be used for the following controls.

- 32-bit: DotUtilType

Feature

To register devices whose status to be monitored.

Format

■ Visual Basic .NET

`IRet = object.EntryDeviceStatus(szLabelList, iSize, iMonitorCycle, iData)`

Data type	Argument	Description	Input/Output
Integer	IRet	Returned value	Output
String	szLabelList(n)	Registered label name list	Input
Integer	iSize	Number of registered device points	Input
Integer	iMonitorCycle	Status monitoring time interval	Input
Integer	iData(n)	Registered device value list	Input

■ Visual C#

`iRet = object. EntryDeviceStatus(szLabelList, iSize, iMonitorCycle, iData)`

Data type	Argument	Description	Input/Output
int	iRet	Returned value	Output
System.String[n]	szLabelList	Registered label name list	Input
int	iSize	Number of registered device points	Input
int	iMonitorCycle	Status monitoring time interval	Input
int[n]	iData	Registered device value list	Input

Returned value

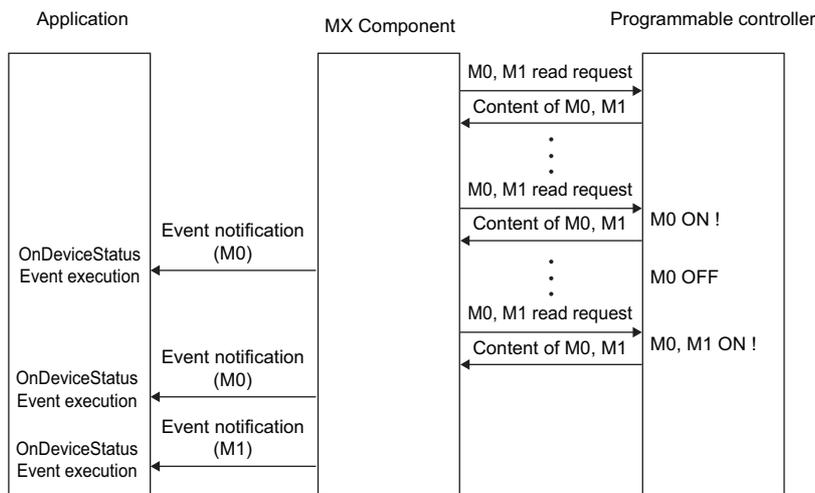
Normal termination: '0' is returned.

Abnormal termination: Any value other than '0' is returned. ( Page 616 ERROR CODES)

Description

- A device group for the size of iSize specified for szLabelList is checked whether it is in the status specified for iData. Specify the check time for iMonitorCycle. When the status is established, the OnDeviceStatus function of the user program is executed.
- The maximum number of device points for iSize is 20.
- Specify a value within the range from 1 second to 1 hour (set between 1 and 3600 in seconds) for iMonitorCycle. An error occurs when any other value outside the above range is specified.
- The registered device value list is stored in iData.

- Device status monitoring may not be performed at the specified status monitoring time intervals depending on the conditions: personal computer performance, currently executed application load, time required for communication with the programmable controller, or the like.
Simultaneous use of any other control functions would also be the cause of disabling device status monitoring at the specified status monitoring time intervals.
- For iData, reserve a memory area for the number of points specified for iSize.
If the memory area is not reserved, a critical error (an application error or the like) may occur.
- An error occurs when the EntryDeviceStatus function is executed during a status monitoring.
When changing any status monitor condition, execute the FreeDeviceStatus function and then execute the EntryDeviceStatus function.
- When the status of multiple devices changes at the same time, the OnDeviceStatus event is executed every time the status changes.
(Example: When M0 is monitored)



- This function is a function to check the status establishment under the constant execution of random device read by the control.
This function is not a function for a CPU module to notify the device status establishment to MX Component. Therefore, the control may not be able to check the device status establishment of CPU module depending on the specified status monitoring time interval.
- Digit specified bit device and index setting cannot be used.

Considerations for checking word device status

When checking the word device status for negative values of -1 to -32768 (FFFFH to 8000H), set the monitor device value of the EntryDeviceStatus function to any of 65535 to 32768 (0000FFFFH to 00008000H) where '0' is stored in the upper 2-byte. (Example) When checking the D0 status for "-10"

Set the value "65526 (0000FFF6H)" where '0' is stored in the upper 2-byte of "-10 (FFFFFFF6H)" for the monitor device value. While the type of word devices of the CPU module is WORD type, the type of monitor device value of the EntryDeviceStatus function is LONG type. Therefore, when current values of CPU module are compared with monitor device values of the EntryDeviceStatus function, the values do not match and the above setting is required. (When bit devices or Double Word devices are used, this consideration does not apply.)

For the programming examples regarding this consideration, refer to the following:

📖 Page 670 Programing Examples for Monitoring Word Device Status

How to specify labels

The following describes how to specify labels.

For the system label name, the following type class, array element, and label data type can be specified.

When specifying a system label name in a program, use the following formats.

○: Available, ×: Not available, —: Not applicable

Label definition in Label Utility		Label specification method in a program			
Type class	Array element	Label name format	Availability	Data type	
Simple Types	None	System label name	○	Integer/int (32-bit value)	
		System label name [n1]	○	Integer/int (32-bit value)	
	Element (One Dimension)	System label name	×	—	
		System label name [n1]	○	Integer/int (32-bit value)	
	Element (Two Dimensions)	System label name	×	—	
		System label name [n1] [n2]	○	Integer/int (32-bit value)	
	Element (Three Dimensions)	System label name	×	—	
		System label name [n1] [n2] [n3]	○	Integer/int (32-bit value)	
	Structure	None	System label name	×	—
			System label name.structure member name	○	Integer/int (32-bit value)
		Element (One Dimension)	System label name	×	—
			System label name [n1]	×	—
System label name [n1].structure member name			○	Integer/int (32-bit value)	
Element (Two Dimensions)		System label name	×	—	
		System label name [n1] [n2]	×	—	
		System label name [n1] [n2].structure member name	○	Integer/int (32-bit value)	
Element (Three Dimensions)		System label name	×	—	
		System label name [n1] [n2] [n3]	×	—	
		System label name [n1] [n2] [n3].structure member name	○	Integer/int (32-bit value)	

■Data type

The following data type can be specified for a label.

Bit, Word (Signed), Word (Unsigned)

How to specify devices

Specify the device values to be registered as follows:

■When specifying bit devices and word devices

Example: Register 1 point of M0 and D0.

· Label setting

szLabelList	Label name	Data type	Device
[0]	LABEL1	Bit	M0
[1]	LABEL2	Word	D0

· umber of registered device points: 2

· Registered device values

Upper 2-byte	Lower 2-byte	Applicable label
Not used ('0' is stored.)	M0	LABEL1
	D0	LABEL2

■When specifying CN200 and later devices of FXCPU

Example: Register 3 points of devices including CN200.

· Label setting

szLabelList	Label name	Data type	Device
[0]	LABEL1	Word	D0
[1]	LABEL2	Word	CN200
[2]	LABEL3	Word	D1

· Number of registered device points: 3

· Registered device values

Upper 2-byte	Lower 2-byte	Applicable label
Not used ('0' is stored.)	D0	LABEL1
H of CN200	L of CN200*1	LABEL2
Not used ('0' is stored.)	D1	LABEL3

*1 For CN200 and later devices of FXCPU, 4-byte can be registered.

■When specifying FD devices (4-word devices)

Example: Register 3 points of data from FD0.

· Label setting

szLabelList	Label name	Data type	Device
[0]	LABEL1	Word	D0
[1]	LABEL2	Word	FD0
[2]	LABEL3	Word	D1

· Number of registered device points: 3

· Registered device values

Upper 2-byte	Lower 2-byte	Applicable label
Not used ('0' is stored.)	D0	LABEL1
	LL of FD0*1	LABEL2
	D1	LABEL3

*1 Only the lower 2-byte can be registered. The HH, HL, and LH (upper 6-byte) of specified devices cannot be registered.

FreeDeviceStatus (Deregistering devices for status monitoring)

Applicable controls

This function can be used for the following controls.

- 32-bit: DotUtilType

Feature

To deregister devices registered by using the EntryDeviceStatus function to monitor their status.

Format

■ Visual Basic .NET

IRet = object.FreeDeviceStatus()

Data type	Argument	Description	Input/Output
Integer	IRet	Returned value	Output

■ Visual C#

iRet = object.FreeDeviceStatus()

Data type	Argument	Description	Input/Output
int	iRet	Returned value	Output

Returned value

Normal termination: '0' is returned.

Abnormal termination: Any value other than '0' is returned. ( Page 616 ERROR CODES)

Description

The devices that are set using the EntryDeviceStatus function to monitor their status are deregistered.

OnDeviceStatus (Event notification)

Applicable controls

This function can be used for the following controls.

- 32-bit: DotUtilType

Feature

To execute the event notification when the device condition registered by using the EntryDeviceStatus function is satisfied.

Format

■Visual Basic .NET

- Adding an event handler

AddHandler object.OnDeviceStatus, AddressOf object_OnDeviceStatus

- Event handler

```
Private Sub object_OnDeviceStatus(ByVal sender As System.Object,  
ByVal e As MITSUBISHI.Component.DeviceStatusEventArgs)
```

Argument	Description	Input/Output
sender	Event occurrence source	Output
e	Event data	Output

The members of e are as follows:

Member	Description
e.szDevice	Name of label whose condition is satisfied
e.IData	Value of device whose condition is satisfied
e.IReturnCode	Returned value of condition check processing

- Removing an event handler

RemoveHandler object.OnDeviceStatus, AddressOf object_OnDeviceStatus

■Visual C#

- Adding an event handler

```
object.OnDeviceStatus += new MITSUBISHI.Component.DotUtilType.DeviceStatusEventHandler (object_OnDeviceStatus);
```

- Event handler

```
private void object_OnDeviceStatus(object sender, MITSUBISHI.Component.DeviceStatusEventArgs e)
```

Argument	Description	Input/Output
sender	Event occurrence source	Output
e	Event data	Output

The members of e are as follows:

Member	Description
e.szDevice	Name of label whose condition is satisfied
e.IData	Value of device whose condition is satisfied
e.IReturnCode	Returned value of condition check processing

- Removing an event handler

```
object.OnDeviceStatus -= new MITSUBISHI.Component.DotUtilType.DeviceStatusEventHandler(object_OnDeviceStatus);
```

Returned value

None

Description

- The event is notified to the application when the device condition registered using the `EntryDeviceStatus` function is satisfied.
Programming this function in the user program allows the application to receive the event when the registered device condition is satisfied.
- Device values registered by using the `EntryDeviceStatus` function are input to `IData`.
Example: When the word device is monitored for the value of "-1"
Set 65535 (0000FFFFH) as a registered device value by using the `EntryDeviceStatus` function.
When the value of the target word device of the CPU module becomes "-1" (FFFFH), the `OnDeviceStatus` function is executed and (0000FFFFH) is input to `IData`.

Point

When any of the following settings is set in the user program, the event of the `OnDeviceStatus` function does not occur even if the condition of the device registered to the `EntryDeviceStatus` function is satisfied.

Note that when the event occurrence becomes in wait status, the control function is not returned to the `.Net` control and device management processing stops until the following setting is terminated.

- For user applications created by using Visual Basic .NET:
The message box is being displayed in the user program.
The `InputBox/OutputBox` is being displayed in the user program.
- For user programs created by using Visual Basic .NET or Visual C#:
The Sleep processing, `WaitForSingleObject` function, or similar standby function is being used in the user program.

ReadDeviceBlock2 (Reading devices in batch)

Applicable controls

This function can be used for the following controls.

- 32-bit: DotUtilType
- 64-bit: DotUtilType64

Feature

To read array labels in 2-byte data unit in batch.

Format

■ Visual Basic .NET

`IRet = object.ReadDeviceBlock2(szLabel, iSize, sData)`

Data type	Argument	Description	Input/Output
Integer	IRet	Returned value	Output
String	szLabel	Label name	Input
Integer	iSize	Number of read points	Input
Short	sData(n)	Read device value	Output

■ Visual C#

`IRet = object.ReadDeviceBlock2(ref szLabel, iSize, ref sData)`

Data type	Argument	Description	Input/Output
int	iRet	Returned value	Output
String	szLabel	Label name	Input
int	iSize	Number of read points	Input
short[n]	sData	Read device value	Output

Returned value

Normal termination: '0' is returned.

Abnormal termination: Any value other than '0' is returned. ([Page 616 ERROR CODES](#))

Description

- Data of devices specified for iSize (number of read points) is read in batch starting from the device specified for szLabel (label name).
- The read device values are stored in sData.
- For sData, reserve arrays for more than the amount specified for iSize.

-
- The maximum number of read points should be the value which satisfies the following condition:
Start read device number + Number of read points \leq Last device number
 - When specifying bit type array labels, 0 or a multiple of 16 can be specified as a device number.
 - For the number of read points, specify the number of words which applies to the data type specified for the label name.
For the read device values, reserve a memory area for the number of points specified for the number of read points.
If the memory area is not reserved, a critical error (an application error or the like) may occur.
 - When a device which corresponds to the label name does not exist, an error occurs and data cannot be read.
When any one of devices which correspond to multiple label names does not exist, an error occurs and data cannot be read.
 - Digit specified bit device and index setting cannot be used.
 - Devices of long timer contact (LTS), long timer coil (LTC), long retentive timer contact (LSTS), long retentive timer coil (LSTC) cannot be read. Use the ReadDeviceRandom2 or GetDevice2 function.
-

How to specify labels

The following describes how to specify labels.

For the system label name, the following type class, array element, and label data type can be specified.

When specifying a system label name in a program, use the following formats.

○: Available, ×: Not available, —: Not applicable

Label definition in Label Utility		Label specification method in a program		
Type class	Array element	Label name format	Availability	Data type
Simple Types	None	System label name	×	—
	Element (One Dimension)	System label name ^{*1}	○	Short/short (16-bit value)
		System label name [n1]	×	—
	Element (Two Dimensions)	System label name ^{*1}	○	Short/short (16-bit value)
		System label name [n1] [n2]	×	—
	Element (Three Dimensions)	System label name ^{*1}	○	Short/short (16-bit value)
System label name [n1] [n2] [n3]		×	—	
Structure	None	System label name	×	—
		System label name.structure member name	×	—
	Element (One Dimension)	System label name	×	—
		System label name [n1]	×	—
		System label name [n1].structure member name	×	—
	Element (Two Dimensions)	System label name	×	—
		System label name [n1] [n2]	×	—
		System label name [n1] [n2].structure member name	×	—
	Element (Three Dimensions)	System label name	×	—
		System label name [n1] [n2] [n3]	×	—
		System label name [n1] [n2] [n3].structure member name	×	—

*1 Specified without specifying an array element.

■Data type

The following data type can be specified for a label.

Bit^{*3}, Word (Signed), Double Word (Signed), Word (Unsigned), Double Word (Unsigned), Float (Single Precision), Float (Double Precision), String^{*1}, String (Unicode)^{*2*}, Time, Timer, Timer (Unsigned), Long timer^{*3}, Counter, Counter (Unsigned), Long counter^{*3}, Retentive Timer, Retentive Timer (Unsigned), Long retentive timer^{*3}

*1 Can be specified up to 32 single-byte characters + NULL.

*2 Can be specified up to 32 characters + NULL.

*3 Cannot be used for system label Ver.1.

■Number of read points

For the number of read points, specify the following values according to the label data type.

Label (array) data type	Number of applicable words	Number of read points to be specified
Bit	1	Number of label array elements ÷ 16 (rounded up)
Word (Signed)	1	Number of label array elements
Double Word (Signed)	2 ^{*1}	Number of label array elements × 2 ^{*1}
Word (Unsigned)	1	Number of label array elements
Double Word (Unsigned)	2 ^{*1}	Number of label array elements × 2 ^{*1}
Float (Single Precision)	2 ^{*1}	Number of label array elements × 2 ^{*1}
Float (Double Precision)	4	Number of label array elements × 4
String	17	Number of label array elements × 17
String (Unicode)	33	Number of label array elements × 33
Time	2 ^{*1}	Number of label array elements × 2 ^{*1}
Timer, Timer (Unsigned), Long timer, Counter, Counter (Unsigned), Long counter, Retentive Timer, Retentive Timer (Unsigned), Long retentive timer	1	Number of label array elements

*1 When specifying a double word device for a label, the number of devices (elements) is 1.

How to specify devices

The read device values are stored as follows:

■When specifying bit devices

Example: Read 3 points of data (3 words = 48 bits) from the devices starting from M0.

· Label setting (Data type: Bit, Number of array elements: 48)

szLabel	Label name	Data type	Device
[0] to [2]	LABEL	Bit (0..47)	M0 to M47

· Number of read points: 3

· Read device values

2-byte	Applicable label
M0 to M15 ^{*1}	LABEL[0] to LABEL[15]
M16 to M31 ^{*1}	LABEL[16] to LABEL[31]
M32 to M47 ^{*1}	LABEL[30] to LABEL[47]

*1 Devices are stored from the lower bit in the order of device number.

■When specifying word devices

Example: Read 3 points of data from the devices starting from D0.

· Label setting (Data type: Word, Number of array elements: 3)

szLabel	Label name	Data type	Device
[0] to [2]	LABEL	Word (0..2)	D0 to D2

· Number of read points: 3

· Read device values

2-byte	Applicable label
D0	LABEL[0]
D1	LABEL[1]
D2	LABEL[2]

■When specifying CN200 and later devices of FXCPU

Example: Read 6 points (6 words = 3 double words) of data from the devices starting from CN200.

For CN200 and later devices of FXCPU, 2 points of data are read from each 1 point of device (upper (H) and lower (L) data).

Reading only 1 point of data will result in an error.

· Label setting (Data type: Double Word, Number of array elements: 3)

szLabel	Label name	Data type	Device
[0] to [5]	LABEL	Double Word (0..2)	CN200 to CN202

· Number of read points: 6

· Read device values

2-byte	Applicable label
L of CN200	LABEL[0]
H of CN200	
L of CN201	LABEL[1]
H of CN201	
L of CN202	LABEL[2]
H of CN202	

■When specifying FD devices (4-word devices)

Example: Read 8 points of data from the devices starting from FD0.

· Label setting (Data type: Double Word, Number of array elements: 2)

szLabel	Label name	Data type	Device
[0] to [7]	LABEL	Double Word (0..1)	FD0, FD1

· Number of read points: 8

· Read device values

2-byte	Applicable label
LL of FD0	LABEL[0]
LH of FD0	
HL of FD0	
HH of FD0	
LL of FD1	LABEL[1]
LH of FD1	
HL of FD1	
HH of FD1	

■When specifying double word devices in Double Word type

Example: Read 2 points of data from the devices starting from LZ0.

· Label setting (Data type: Double Word, Number of array elements: 2)

szLabel	Label name	Data type	Device
[0] to [1]	LABEL	Double Word (0..1)	LZ0 to LZ1

· Number of read points: 6

· Read device values

2-byte	Applicable label
L of LZ0	LABEL[0]
L of LZ1	LABEL[1]

WriteDeviceBlock2 (Writing devices in batch)

Applicable controls

This function can be used for the following controls.

- 32-bit: DotUtilType
- 64-bit: DotUtilType64

Feature

To write array labels in 2-byte data unit in batch.

Format

■ Visual Basic .NET

IRet = object.WriteDeviceBlock2(szLabel, iSize, sData)

Data type	Argument	Description	Input/Output
Integer	IRet	Returned value	Output
String	szLabel	Label name	Input
Integer	iSize	Number of write points	Input
Short	sData(n)	Device value to be written	Input

■ Visual C#

iRet = object.WriteDeviceBlock2(ref szLabel, iSize, sData)

Data type	Argument	Description	Input/Output
int	iRet	Returned value	Output
String	szLabel	Label name	Input
int	iSize	Number of write points	Input
short[n]	sData	Device value to be written	Input

Returned value

Normal termination: '0' is returned.

Abnormal termination: Any value other than '0' is returned. ([Page 616 ERROR CODES](#))

Description

- Data of devices specified for iSize (number of write points) is written in batch starting from the device specified for szLabel (label name).
- Store the device values to be written in sData.
- For sData, reserve arrays for more than the amount specified for iSize.

- The maximum number of write points should be the value which satisfies the following condition:
Start write device number + Number of write points \leq Last device number
- When specifying bit type array labels, 0 or a multiple of 16 can be specified as a device number.
- For the number of write points, specify the number of words which applies to the data type specified for the label name.
For the device values to be written, reserve a memory area for the number of points specified for the number of write points.
If the memory area is not reserved, a critical error (an application error or the like) may occur.
- When a device which corresponds to the label name does not exist, an error occurs and data cannot be written.
When any one of devices which correspond to multiple label names does not exist, an error occurs and data cannot be written.
- Digit specified bit device and index setting cannot be used.
- Devices of long timer contact (LTS), long timer coil (LTC), long retentive timer contact (LSTS), long retentive timer coil (LSTC) cannot be read. Use the WriteDeviceRandom2 or SetDevice2 function.
- If the function is run against the safety device in the RnSFCEPU safety mode, an error code 0x010A42A5" (an operation that can not be carried out in safety mode was performed) is returned.

How to specify labels

The following describes how to specify labels.

For the system label name, the following type class, array element, and label data type can be specified.

When specifying a system label name in a program, use the following formats.

○: Available, ×: Not available, —: Not applicable

Label definition in Label Utility		Label specification method in a program		
Type class	Array element	Label name format	Availability	Data type
Simple Types	None	System label name	×	—
	Element (One Dimension)	System label name ^{*1}	○	Short/short (16-bit value)
		System label name [n1]	×	—
	Element (Two Dimensions)	System label name ^{*1}	○	Short/short (16-bit value)
		System label name [n1] [n2]	×	—
	Element (Three Dimensions)	System label name ^{*1}	○	Short/short (16-bit value)
System label name [n1] [n2] [n3]		×	—	
Structure	None	System label name	×	—
		System label name.structure member name	×	—
	Element (One Dimension)	System label name	×	—
		System label name [n1]	×	—
		System label name [n1].structure member name	×	—
	Element (Two Dimensions)	System label name	×	—
		System label name [n1] [n2]	×	—
		System label name [n1] [n2].structure member name	×	—
	Element (Three Dimensions)	System label name	×	—
		System label name [n1] [n2] [n3]	×	—
		System label name [n1] [n2] [n3].structure member name	×	—

*1 Specified without specifying an array element.

■Data type

The following data type can be specified for a label.

Bit^{*3}, Word (Signed), Double Word (Signed), Word (Unsigned), Double Word (Unsigned), Float (Single Precision), Float (Double Precision), String^{*1}, String (Unicode)^{*2*3}, Time, Timer, Timer (Unsigned), Long timer^{*3}, Counter, Counter (Unsigned), Long counter^{*3}, Retentive Timer, Retentive Timer (Unsigned), Long retentive timer^{*3}

*1 Can be specified up to 32 single-byte characters + NULL.

*2 Can be specified up to 32 characters + NULL.

*3 Cannot be used for system label Ver.1.

■Number of write points

For the number of write points, specify the following values according to the label data type.

Label (array) data type	Number of applicable words	Number of write points to be specified
Bit	1	Number of label array elements ÷ 16 (rounded up)
Word (Signed)	1	Number of label array elements
Double Word (Signed)	2 ^{*1}	Number of label array elements × 2 ^{*1}
Word (Unsigned)	1	Number of label array elements
Double Word (Unsigned)	2 ^{*1}	Number of label array elements × 2 ^{*1}
Float (Single Precision)	2 ^{*1}	Number of label array elements × 2 ^{*1}
Float (Double Precision)	4	Number of label array elements × 4
String	17	Number of label array elements × 17
String (Unicode)	33	Number of label array elements × 33
Time	2 ^{*1}	Number of label array elements × 2 ^{*1}
Timer, Timer (Unsigned), Long timer, Counter, Counter (Unsigned), Long counter, Retentive Timer, Retentive Timer (Unsigned), Long retentive timer	1	Number of label array elements

*1 When specifying a double word device for a label, the number of devices (elements) is 1.

How to specify devices

Specify the device values to be written as follows:

■When specifying bit devices

Example: Write 3 points (3 words = 48 bits) of data to the devices starting from M0.

· Label setting (Data type: Bit, Number of array elements: 48)

szLabel	Label name	Data type	Device
[0] to [2]	LABEL	Bit (0..47)	M0 to M47

· Number of write points: 3

· Device values to be written

2-byte	Applicable label
M0 to M15 ^{*1}	LABEL[0]
M16 to M31 ^{*1}	LABEL[1]
M32 to M47 ^{*1}	LABEL[2]

*1 Devices are stored from the lower bit in the order of device number.

■When specifying word devices

Example: Write 3 points of data to the devices starting from D0.

· Label setting (Data type: Word, Number of array elements: 3)

szLabel	Label name	Data type	Device
[0] to [2]	LABEL	Word (0..2)	D0 to D2

· Number of write points: 3

· Device values to be written

2-byte	Applicable label
D0	LABEL[0]
D1	LABEL[1]
D2	LABEL[2]

■When specifying CN200 and later devices of FXCPU

Example: Write 6 points (6 words = 3 double words) of data to the devices starting from CN200.

For CN200 and later devices of FXCPU, 2 points of data are written to each 1 point of device (upper (H) and lower (L) data).

Writing only 1 point of data will result in an error.

· Label setting (Data type: Double Word, Number of array elements: 3)

szLabel	Label name	Data type	Device
[0] to [5]	LABEL	Double Word (0..2)	CN200 to CN202

· Number of write points: 6

· Device values to be written

2-byte	Applicable label
L of CN200	LABEL[0]
H of CN200	
L of CN201	LABEL[1]
H of CN201	
L of CN202	LABEL[2]
H of CN202	

■When specifying FD devices (4-word devices)

Example: Read 8 points (8 words) of data from the devices starting from FD0.

· Label setting (Data type: Double Word, Number of array elements: 2)

szLabel	Label name	Data type	Device
[0] to [7]	LABEL	Double Word (0..1)	FD0, FD1

· Number of write points: 8

· Device values to be written

2-byte	Applicable label
LL of FD0	LABEL[0]
LH of FD0	
HL of FD0	
HH of FD0	
LL of FD1	LABEL[1]
LH of FD1	
HL of FD1	
HH of FD1	

■When specifying word devices for double word array type labels

Example: Write 6 points (6 words = 3 double words) of data to the devices starting from D0.

· Label setting (Data type: Double Word, Number of array elements: 3)

szLabel	Label name	Data type	Device
[0] to [5]	LABEL	Double Word (0..2)	D0 to D5

· Number of write points: 6

· Device values to be written

2-byte	Applicable label
D0	LABEL[0]
D1	
D2	LABEL[1]
D3	
D4	LABEL[2]
D5	

ReadDeviceRandom2 (Reading devices randomly)

Applicable controls

This function can be used for the following controls.

- 32-bit: DotUtilType
- 64-bit: DotUtilType64

Feature

To read devices in 2-byte data unit randomly.

Format

■Visual Basic .NET

`IRet = object.ReadDeviceRandom2(szLabel, iSize, sData)`

Data type	Argument	Description	Input/Output
Integer	IRet	Returned value	Output
String	szLabel	Label name	Input
Integer	iSize	Number of read points	Input
Short	sData(n)	Read device value	Output

`IRet = object.ReadDeviceRandom2(szLabelList, iSize, sData)`

Data type	Argument	Description	Input/Output
Integer	IRet	Returned value	Output
String	szLabelList(n)	Label list	Input
Integer	iSize	Number of read points	Input
Short	sData(n)	Read device value	Output

■Visual C#

`iRet = object.ReadDeviceRandom2(ref szLabel, iSize, ref sData)`

Data type	Argument	Description	Input/Output
Int	iRet	Returned value	Output
String	szLabel	Label name	Input
int	iSize	Number of read points	Input
short[n]	sData	Read device value	Output

`iRet = object.ReadDeviceRandom2(ref szLabelList, iSize, ref sData)`

Data type	Argument	Description	Input/Output
Int	iRet	Returned value	Output
System.String[n]	szLabelList	Label list	Input
int	iSize	Number of read points	Input
short[n]	sData	Read device value	Output

Returned value

Normal termination: '0' is returned.

Abnormal termination: Any value other than '0' is returned. (📖 Page 616 ERROR CODES)

Description

- Data of a device group specified for the label name szLabel (szLabelList) is read for the size of iSize.
- The read device values are stored in sData.
- For sData, reserve arrays for more than the amount specified for iSize.

Point

- The maximum number of read points is 0x7FFFFFFF.
- For the number of read points, specify the number of words which applies to the data type specified for the label name.
For the read device values, reserve a memory area for the number of points specified for the number of read points.
If the memory area is not reserved, a critical error (an application error or the like) may occur.
- When a device which corresponds to the label name does not exist, an error occurs and data cannot be read.
When any one of devices which correspond to multiple label names does not exist, an error occurs and data cannot be read.
- When specifying a double word device, this function stores only the data of lower 1 word (2-byte). (An error does not occur.)
Use the ReadDeviceRandom or ReadDeviceBlock function to read data from double word devices.
- Digit specified bit device and index setting cannot be used.
- Up to five levels of a structure can be defined. Note that only one-level structure can be used for a static system label.

How to specify labels

The following describes how to specify labels.

For the system label name, the following type class, array element, and label data type can be specified.

When specifying a system label name in a program, use the following formats.

○: Available, ×: Not available, —: Not applicable

Label definition in Label Utility		Label specification method in a program			
Type class	Array element	Label name format	Availability	Data type	
Simple Types	None	System label name	○	Short/short (16-bit value)	
		System label name	○ ^{*1}	Short/short (16-bit value)	
	Element (One Dimension)	System label name [n1]	○	Short/short (16-bit value)	
		System label name	○ ^{*1}	Short/short (16-bit value)	
	Element (Two Dimensions)	System label name [n1] [n2]	○	Short/short (16-bit value)	
		System label name	○ ^{*1}	Short/short (16-bit value)	
	Element (Three Dimensions)	System label name [n1] [n2] [n3]	○	Short/short (16-bit value)	
		System label name	○ ^{*1}	Short/short (16-bit value)	
	Structure	None	System label name	○ ^{*1}	Short/short (16-bit value)
			System label name.structure member name	○	Short/short (16-bit value)
		Element (One Dimension)	System label name	○ ^{*1}	Short/short (16-bit value)
			System label name [n1]	○ ^{*1}	Short/short (16-bit value)
System label name [n1].structure member name			○	Short/short (16-bit value)	
Element (Two Dimensions)		System label name	○ ^{*1}	Short/short (16-bit value)	
		System label name [n1] [n2]	○ ^{*1}	Short/short (16-bit value)	
		System label name [n1] [n2].structure member name	○	Short/short (16-bit value)	
Element (Three Dimensions)		System label name	○ ^{*1}	Short/short (16-bit value)	
		System label name [n1] [n2] [n3]	○ ^{*1}	Short/short (16-bit value)	
		System label name [n1] [n2] [n3].structure member name	○	Short/short (16-bit value)	

*1 The entire data of arrays and structures is read.

■Data type

The following data type can be specified for a label.

Bit, Word (Signed), Double Word (Signed), Word (Unsigned), Double Word (Unsigned), Float (Single Precision), Float (Double Precision), String^{*1}, String (Unicode)^{*2*4}, Time, Timer^{*3}, Timer (Unsigned)^{*3}, Long timer^{*3*4}, Counter^{*3}, Counter (Unsigned)^{*3}, Long counter^{*3*4}, Retentive Timer^{*3}, Retentive Timer (Unsigned)^{*3}, Long retentive timer^{*3*4}

*1 Can be specified up to 32 single-byte characters + NULL.

*2 Can be specified up to 32 characters + NULL.

*3 When accessing the device using a label of system label Ver.2, the label of system label Ver.2 which is corresponding to a device for contact/coil/present value is required to be defined.

*4 Cannot be used for system label Ver.1.

■Number of read points

For the number of read points, specify the sum of the following values correspond to the elements according to the label data type.

Label data type	Number of read points to be specified ^{*1}
Bit	1
Word (Signed)	1
Double Word (Signed)	2 ^{*2}
Word (Unsigned)	1
Double Word (Unsigned)	2 ^{*2}
Float (Single Precision)	2 ^{*2}
Float (Double Precision)	4
String	17
String (Unicode)	33
Time	2 ^{*2}
Timer, Timer (Unsigned), Long timer, Counter, Counter (Unsigned), Long counter, Retentive Timer, Retentive Timer (Unsigned), Long retentive timer	1
Structure	Total value of the number of applicable words for the structure member data type

*1 When accessing data by specifying the label name only (without element specification) for the array label, specify the number of read points in this table multiplied by the number of array elements.

*2 When reading a single label of "Simple Types" to which a double word device is assigned, the number of read points is 1. It is 2 when reading a label of "Structure" to which a double word device is assigned and when reading multiple labels to which double word devices are assigned at the same time.

How to specify devices

The read device values are stored as follows:

■When specifying CN200 and later devices of FXCPU or double word devices

Example: Read 4 points of data from the devices including CN200.

· Label setting

Label name	Data type	Device
LABEL1	Word	D0
LABEL2	Double Word	CN200
LABEL3	Word	D1

· Number of read points: 4

· Reading method

```
short iData[4]
String stLabelList[3] = {"LABEL1", "LABEL2", "LABEL3"}
ReadDeviceRandom2(stLabelList,4,iData)
```

· Read device values

2-byte
D0
L of CN200* ¹
L of CN201* ¹
D1

*1 2 words are read for 1 point.

■When specifying array type labels

Example 1: Read 3 points of data from the bit devices (M0 to M2) by specifying array type labels.

· Label setting

Label name	Data type	Device
LABEL	Bit (0..2)	M0

· Number of read points: 3

· Reading method

```
short sData[3]
ReadDeviceRandom2("LABEL",3,sData)
```

· Read device values

2-byte
M0
M1
M2

Example 2: Read 3 points of data from the word devices (D0 to D2) by specifying array type labels.

· Label setting

Label name	Data type	Device
LABEL	Word (0..2)	D0

· Number of read points: 3

· Reading method

```
short sData[3]
ReadDeviceRandom2("LABEL",3,sData)
```

· Read device values

2-byte
D0
D1
D2

Example 3: Read 3 points of data from the double word devices (CN200 to CN202) by specifying array type labels.

· Label setting

Label name	Data type	Device
LABEL	Word (0..2)	CN200

· Number of read points: 3

· Reading method

```
short sData[3]
ReadDeviceRandom2("LABEL",3,sData)
```

· Read device values

2-byte
L of CN200
L of CN201
L of CN202

Example 4: Read 6 points (6 words = 3 double words) of data by specifying array type labels.

· Label setting

Label name	Data type	Device
LABEL	Double Word (0..2)	D0

· Number of read points: 6

· Reading method

```
short sData[6]
ReadDeviceRandom2("LABEL",6,sData)
```

· Read device values

2-byte	Applicable label
D0	LABEL[0]
D1	
D2	LABEL[1]
D3	
D4	LABEL[2]
D5	

Example 5: Read 51 points (D0 to D50) of string type data by specifying array type labels.

· Label setting

Label name	Data type	Device
LABEL	String (0..2)	D0

· Number of read points: 51

· Reading method

```
short sData[51]
ReadDeviceRandom2("LABEL",51,sData)
```

The characters need to be converted in a user program because the characters of String type are not converted.

· Read device values

index	2-byte	Applicable label
[0]	D0	LABEL[0]
[1]	D1	
⋮	⋮	
[16]	D16	LABEL[1]
[17]	D17	
⋮	⋮	
[18]	D18	LABEL[2]
[33]	D33	
⋮	⋮	
[34]	D34	LABEL[2]
[35]	D35	
[50]	D50	

■ When specifying structure type labels

Example 1: Read data by specifying structure type labels.

· Structure setting

Structure name	Member name	Data type
STRUCT	LABEL1	Bit
	LABEL2	Word
	LABEL3	Word

· Label setting

Label name	Device
STRUCT.LABEL1	M0
STRUCT.LABEL2	D0
STRUCT.LABEL3	W0

When accessing by specifying the label name only

· Number of read points: 3 (because the total number of read points for each structure member is 3)

· Reading method

```
short sData[3]
ReadDeviceRandom2("STRUCT",3,sData)
```

· Read device values

2-byte
M0
D0
W0

When accessing by specifying the label name + member name

· Number of read points: 2 (because 2 members of structure, for which each number of read points is 1, are specified)

· Reading method

```
String stLabelList[2] = {"STRUCT.LABEL1","STRUCT.LABEL2"}
short sData[2]
ReadDeviceRandom2(stLabelList,2,sData)
```

· Read device values

2-byte
M0
D0

Example 2: Read data by specifying structure type labels.

· Structure setting

Structure name	Member name	Data type
STRUCT	LABEL1	Bit
	LABEL2	Double Word
	LABEL3	Float (Double Precision)
	LABEL4	Word

· Label setting

Label name	Device
STRUCT.LABEL1	D0.0
STRUCT.LABEL2	D1
STRUCT.LABEL3	D3
STRUCT.LABEL4	D7

· Number of read points: 8 (because the total number of read points for each structure member is 8)

· Reading method

```
short sData[8]
ReadDeviceRandom2("STRUCT",8,sData)
```

· Read device values

2-byte	Applicable label
D0.0	STRUCT.LABEL1
D1	STRUCT.LABEL2
D2	
D3	
D4	STRUCT.LABEL3
D5	
D6	
D7	STRUCT.LABEL4

■When specifying labels in combination with structure and array

Example: Read data by specifying structure type array and structure array type labels.

· Structure setting

Structure name	Member name	Data type
STRUCT	LABEL1	Bit
	LABEL2	Word (0..1)
	LABEL3	Double Word
	LABEL4	String
	LABEL5	Word

· Label setting

Label name	Device
STRUCT[0].LABEL1	M0
STRUCT[0].LABEL2[0]	D0
STRUCT[0].LABEL2[1]	D1
STRUCT[0].LABEL3	D2
STRUCT[0].LABEL4	D4
STRUCT[0].LABEL5	D21
STRUCT[1].LABEL1	M1
STRUCT[1].LABEL2[0]	D22
STRUCT[1].LABEL2[1]	D23
STRUCT[1].LABEL3	D24
STRUCT[1].LABEL4	D26
STRUCT[1].LABEL5	D43

· Number of read points: 46 (because the total number of read points for each structure is 23, and the number of array elements is 2)

· Reading method

```
short iData[46]
```

```
ReadDeviceRandom2("STRUCT",46,iData)
```

or

```
String stLabelList[2] = {"STRUCT[0]","STRUCT[1]"};
```

```
ReadDeviceRandom2(stLabelList,46 ,iData)
```

· Read device values

index	2-byte	Applicable label
[0]	M0	STRUCT[0].LABEL1
[1]	D0	STRUCT[0].LABEL2[0]
[2]	D1	STRUCT[0].LABEL2[1]
[3]	D2	STRUCT[0].LABEL3
[4]	D3	
[5]	D4	STRUCT[0].LABEL4
⋮	⋮	
[21]	D20	
[22]	D21	STRUCT[0].LABEL5
[23]	M1	STRUCT[1].LABEL1
[24]	D22	STRUCT[1].LABEL2[0]
[25]	D23	STRUCT[1].LABEL2[0]
[26]	D24	STRUCT[1].LABEL3
[27]	D25	
[28]	D26	STRUCT[1].LABEL4
⋮	⋮	
[44]	D42	
[45]	D43	STRUCT[1].LABEL5

WriteDeviceRandom2 (Writing devices randomly)

Applicable controls

This function can be used for the following controls.

- 32-bit: DotUtilType
- 64-bit: DotUtilType64

Feature

To write devices in 2-byte data unit randomly.

Format

■Visual Basic .NET

Ret = object.WriteDeviceRandom2(szLabel, iSize, sData)

Data type	Argument	Description	Input/Output
Integer	IRet	Returned value	Output
String	szLabel	Label name	Input
Integer	iSize	Number of write points	Input
Short	sData(n)	Device value to be written	Input

Ret = object.WriteDeviceRandom2(szLabelList, iSize, sData)

Data type	Argument	Description	Input/Output
Integer	IRet	Returned value	Output
String	szLabelList(n)	Label list	Input
Integer	iSize	Number of write points	Input
Short	sData(n)	Device value to be written	Input

■Visual C#

iRet = object.WriteDeviceRandom2(szLabel, iSize, sData)

Data type	Argument	Description	Input/Output
int	iRet	Returned value	Output
String	szLabel	Label name	Input
int	iSize	Number of write points	Input
short[n]	sData	Device value to be written	Input

iRet = object. WriteDeviceRandom2(szLabelList, iSize, sData)

Data type	Argument	Description	Input/Output
int	iRet	Returned value	Output
System.String[n]	szLabelList	Label list	Input
int	iSize	Number of write points	Input
short[n]	sData	Device value to be written	Input

Returned value

Normal termination: '0' is returned.

Abnormal termination: Any value other than '0' is returned. (📖 Page 616 ERROR CODES)

Description

- Data of a device group specified for the label name `szLabel` is written for the size of `iSize`.
- The device values to be written are stored in `sData`.
- For `sData`, reserve arrays for more than the amount specified for `iSize`.

Point

- The maximum number of write points is `0x7FFFFFFF`.
- For the number of write points, specify the number of words which applies to the data type specified for the label name.
For the device values to be written, reserve a memory area for the number of points specified for the number of write points.
If the memory area is not reserved, a critical error (an application error or the like) may occur.
- When a device which corresponds to the label name does not exist, an error occurs and data cannot be written.
When any one of devices which correspond to multiple label names does not exist, an error occurs and data cannot be written.
- When specifying a double word device, this function writes to the data area of lower 1 word (2-byte), and '0' is written to the data area of upper 1 word (2-byte).
Use the `WriteDeviceRandom` or `WriteDeviceBlock` function to write double word devices.
- Digit specified bit device and index setting cannot be used.
- An error is returned when a Q motion CPU is accessed.
- Up to five levels of a structure can be defined. Note that only one-level structure can be used for a static system label.
- If the function is run against the safety device in the `RnSF` CPU safety mode, an error code `0x010A42A5` (an operation that can not be carried out in safety mode was performed) is returned.

How to specify labels

The following describes how to specify labels.

For the system label name, the following type class, array element, and label data type can be specified.

When specifying a system label name in a program, use the following formats.

○: Available, ×: Not available, —: Not applicable

Label definition in Label Utility		Label specification method in a program			
Type class	Array element	Label name format	Availability	Data type	
Simple Types	None	System label name	○	Short/short (16-bit value)	
		System label name	○ ^{*1}	Short/short (16-bit value)	
	Element (One Dimension)	System label name [n1]	○	Short/short (16-bit value)	
		System label name	○ ^{*1}	Short/short (16-bit value)	
	Element (Two Dimensions)	System label name [n1] [n2]	○	Short/short (16-bit value)	
		System label name	○ ^{*1}	Short/short (16-bit value)	
	Element (Three Dimensions)	System label name [n1] [n2] [n3]	○	Short/short (16-bit value)	
		System label name	○ ^{*1}	Short/short (16-bit value)	
	Structure	None	System label name	○ ^{*1}	Short/short (16-bit value)
			System label name.structure member name	○	Short/short (16-bit value)
Element (One Dimension)		System label name	○ ^{*1}	Short/short (16-bit value)	
		System label name [n1]	○ ^{*1}	Short/short (16-bit value)	
		System label name [n1].structure member name	○	Short/short (16-bit value)	
Element (Two Dimensions)		System label name	○ ^{*1}	Short/short (16-bit value)	
		System label name [n1] [n2]	○ ^{*1}	Short/short (16-bit value)	
		System label name [n1] [n2].structure member name	○	Short/short (16-bit value)	
Element (Three Dimensions)		System label name	○ ^{*1}	Short/short (16-bit value)	
		System label name [n1] [n2] [n3]	○ ^{*1}	Short/short (16-bit value)	
		System label name [n1] [n2] [n3].structure member name	○	Short/short (16-bit value)	

*1 The entire data of arrays and structures is written.

■Data type

The following data type can be specified for a label.

Bit, Word (Signed), Double Word (Signed), Word (Unsigned), Double Word (Unsigned), Float (Single Precision), Float (Double Precision), String^{*1}, String (Unicode)^{*2,4}, Time, Timer^{*3}, Timer (Unsigned)^{*3}, Long timer^{*3,4}, Counter^{*3}, Counter (Unsigned)^{*3}, Long counter^{*3,4}, Retentive Timer^{*3}, Retentive Timer (Unsigned)^{*3}, Long retentive timer^{*3,4}

*1 Can be specified up to 32 single-byte characters + NULL.

*2 Can be specified up to 32 characters + NULL.

*3 When accessing the device using a label of system label Ver.2, the label of system label Ver.2 which is corresponding to a device for contact/coil/present value is required to be defined.

*4 Cannot be used for system label Ver.1.

■Number of write points

For the number of write points, specify the sum of the following values correspond to the elements according to the label data type.

Label data type	Number of write points to be specified ^{*1}
Bit	1
Word (Signed)	1
Double Word (Signed)	2 ^{*2}
Word (Unsigned)	1
Double Word (Unsigned)	2 ^{*2}
Float (Single Precision)	2 ^{*2}
Float (Double Precision)	4
String	17
String (Unicode)	33
Time	2 ^{*2}
Timer, Timer (Unsigned), Long timer, Counter, Counter (Unsigned), Long counter, Retentive Timer, Retentive Timer (Unsigned), Long retentive timer	1
Structure	Total value of the number of applicable words for the structure member data type

*1 When accessing data by specifying the label name only (without element specification) for the array label, specify the number of write points in this table multiplied by the number of array elements.

*2 When writing a single label of "Simple Types" to which a double word device is assigned, the number of write points is 1. It is 2 when writing a label of "Structure" to which a double word device is assigned and when writing multiple labels to which double word devices are assigned at the same time.

How to specify devices

Specify the device values to be written as follows:

■When specifying CN200 and later devices of FXCPU or double word devices

Example: Write 4 points of data to the devices including CN200.

· Label setting

Label name	Data type	Device
LABEL1	Word	D0
LABEL2	Double Word	CN200
LABEL3	Word	D1

· Number of write points: 4

· Writing method

```
short sData[4]
String stLabelList[3] = {"LABEL1", "LABEL2", "LABEL3"}
WriteDeviceRandom2(stLabelList,4, sData)
```

· Device values to be written

2-byte
D0
L of CN200 ^{*1} *2
L of CN201 ^{*1} *2
D1

*1 For CN200 and later devices of FXCPU and double word devices, only the lower 2-byte of a specified device is written for 1 point when the SetDevice2 function is executed.

*2 Only the lower 2-byte is written. Data is not written to the H (upper 2-byte) of specified devices.

■When specifying character strings

Example: Write 17 points of data to the devices starting from D0.

· Label setting

Label name	Data type	Device
LABEL	String	D0

· Number of write points: 17

· Writing method

```
short sData[17]
WriteDeviceRandom2("LABEL",17, sData)
```

Data is not converted to character strings.

· Device values to be written

2-byte
D0
D1
D2
⋮
D14
D15
D16

■When specifying data types equivalent to 2 words or more for labels

Example: Write a total of 6 points of data by specifying labels of Double Word, Float (Single Precision), and Time types.

· Label setting

Label name	Data type	Device
LABEL1	Double Word	D0
LABEL2	Float (Single Precision)	D100
LABEL3	Time	D200

· Number of write points: 6

· Writing method

```
String stLabelList[3] = {"LABEL1", "LABEL2", "LABEL3"}
short sData[6]
WriteDeviceRandom2(stLabelList, 6, sData)
```

· Device values to be written

2-byte	Applicable label
D0	LABEL1
D1	
D100	LABEL2
D101	
D200	LABEL3
D201	

■When specifying Float (Double Precision) and word devices

Example: Write a total of 5 points of data by specifying Float (Double Precision) and a word device.

· Label setting

Label name	Data type	Device
LABEL1	Float (Double Precision)	D0
LABEL3	Word	D200

· Number of write points: 5

· Writing method

```
String stLabelList[2] = {"LABEL1", "LABEL3"}
short sData[5]
WriteDeviceRandom2(stLabelList, 5, sData)
```

· Device values to be written

2-byte	Applicable label
D0	LABEL1
D1	
D2	
D3	
D200	LABEL3

■ When specifying array type labels

Example 1: Write 3 points of data to the bit devices (M0 to M2) by specifying array type labels.

· Label setting

Label name	Data type	Device
LABEL	Bit (0..2)	M0

· Number of write points: 3

· Writing method

```
short sData[3]
WriteDeviceRandom2("LABEL",3, sData)
```

· Device values to be written

2-byte
M0
M1
M2

Example 2: Write 3 points of data to the word devices (D0 to D2) by specifying array type labels.

· Label setting

Label name	Data type	Device
LABEL	Word (0..2)	D0

· Number of write points: 3

· Writing method

```
short sData[3]
WriteDeviceRandom2("LABEL",3, sData)
```

· Device values to be written

2-byte
D0
D1
D2

Example 3: Write 6 points (6 words = 3 double words) of data by specifying array type labels.

· Label setting

Label name	Data type	Device
LABEL	Double Word (0..2)	D0

· Number of write points: 6

· Writing method

```
short sData[6]
WriteDeviceRandom2("LABEL",6, sData)
```

· Device values to be written

2-byte	Applicable label
D0	LABEL[0]
D1	
D2	LABEL[1]
D3	
D4	LABEL[2]
D5	

Example 4: Write 34 points (D0 to D33) of string type data by specifying array type labels.

· Label setting

Label name	Data type	Device
LABEL	String (0..1)	D0

· Number of write points: 34

· Writing method

```
short sData[34]
WriteDeviceRandom2("LABEL",34, sData)
```

The characters need to be converted in a user program because the characters of String type are not converted.

· Device values to be written

index	2-byte	Applicable label
[0]	D0	LABEL[0]
[1]	D1	
⋮	⋮	
[16]	D16	LABEL[1]
[17]	D17	
⋮	⋮	
[18]	D18	LABEL[1]
[33]	D33	
⋮	⋮	

■ When specifying structure type labels

Example 1: Write data by specifying structure type labels.

· Structure setting

Structure name	Member name	Data type
STRUCT	LABEL1	Bit
	LABEL2	Word
	LABEL3	Word

· Label setting

Label name	Device
STRUCT.LABEL1	M0
STRUCT.LABEL2	D0
STRUCT.LABEL3	W0

When accessing by specifying the label name only

· Number of write points: 3 (because the total number of write points for each structure member is 3)

· Writing method

```
short sData[3]  
WriteDeviceRandom2("STRUCT",3,sData)
```

· Device values to be written

2-byte
M0
D0
W0

When accessing by specifying the label name + member name

· Number of write points: 2 (because 2 members of structure, for which each number of write points is 1, are specified)

· Writing method

```
String stLabelList[2] = {"STRUCT.LABEL1","STRUCT.LABEL2"}  
short sData[2]  
WriteDeviceRandom2(stLabelList,2,sData)
```

· Device values to be written

2-byte
M0
D0

Example 2: Write data by specifying structure type labels.

· Structure setting

Structure name	Member name	Data type
STRUCT	LABEL1	Bit
	LABEL2	Double Word
	LABEL3	Float (Double Precision)
	LABEL4	Word

· Label setting

Label name	Device
STRUCT.LABEL1	D0.0
STRUCT.LABEL2	D1
STRUCT.LABEL3	D3
STRUCT.LABEL4	D7

· Number of write points: 8 (because the total number of write points for each structure member is 8)

· Writing method

```
short sData[8]
WriteDeviceRandom2("STRUCT",8,sData)
```

· Device values to be written

2-byte	Applicable label
D0.0	STRUCT.LABEL1
D1	STRUCT.LABEL2
D2	
D3	STRUCT.LABEL3
D4	
D5	
D6	
D7	STRUCT.LABEL4

■When specifying labels in combination with structure and array

Example: Write data by specifying structure type array and structure array type labels.

· Structure setting

Structure name	Member name	Data type
STRUCT	LABEL1	Bit
	LABEL2	Word (0..1)
	LABEL3	Double Word
	LABEL4	String
	LABEL5	Word

· Label setting

Label name	Device
STRUCT[0].LABEL1	M0
STRUCT[0].LABEL2[0]	D0
STRUCT[0].LABEL2[1]	D1
STRUCT[0].LABEL3	D2
STRUCT[0].LABEL4	D4
STRUCT[0].LABEL5	D21
STRUCT[1].LABEL1	M1
STRUCT[1].LABEL2[0]	D22
STRUCT[1].LABEL2[1]	D23
STRUCT[1].LABEL3	D24
STRUCT[1].LABEL4	D26
STRUCT[1].LABEL5	D43

· Number of write points: 46 (because the total number of write points for each structure is 23, and the number of array elements is 2)

· Writing method

```
short sData[46]
WriteDeviceRandom2("STRUCT",46,sData)
```

or

```
String stLabelList[2] = {"STRUCT[0]","STRUCT[1]"}
WriteDeviceRandom2(stLabelList,46 ,sData)
```

· Device values to be written

index	Upper 2-byte	Lower 2-byte	Applicable label	
[0]	Not used ('0' is stored.)	M0	STRUCT[0].LABEL1	
[1]		D0	STRUCT[0].LABEL2[0]	
[2]		D1	STRUCT[0].LABEL2[1]	
[3]		D2	STRUCT[0].LABEL3	
[4]		D3		
[5]	⋮	D4	STRUCT[0].LABEL4	
[21]		D20		
[22]		D21		
[23]		M1		STRUCT[1].LABEL1
[24]		D22		STRUCT[1].LABEL2[0]
[25]	D23	STRUCT[1].LABEL2[1]		
[26]	D24	STRUCT[1].LABEL3		
[27]	D25			
[28]	⋮	D26	STRUCT[1].LABEL4	
[44]		D42		
[45]		D43		STRUCT[1].LABEL5

SetDevice2 (Setting device data)

Applicable controls

This function can be used for the following controls.

- 32-bit: DotUtilType
- 64-bit: DotUtilType64

Feature

To set one point of device in 2-byte data unit.

Format

■ Visual Basic .NET

`IRet = object.SetDevice2(szLabel, sData)`

Data type	Argument	Description	Input/Output
Integer	IRet	Returned value	Output
String	szLabel	Label name	Input
Short	sData	Device value to be written	Input

■ Visual C#

`iRet = object.SetDevice2(ref szLabel, sData)`

Data type	Argument	Description	Input/Output
int	iRet	Returned value	Output
String	szLabel	Label name	Input
short	sData	Device value to be written	Input

Returned value

Normal termination: '0' is returned.

Abnormal termination: Any value other than '0' is returned. (🔗 Page 616 ERROR CODES)

Description

- The values of sData (device values to be written) are written to 1 point of device specified for szLabel (label name).
- When specifying bit devices, the least significant bit of the sData (device values to be written) is valid.

Point

- When a device which corresponds to the label name does not exist, an error occurs and data cannot be written.
- When specifying a double word device at the time other than specifying a double word device which can be written for 1 point (such as CN200 and later devices of FXCPU), this function writes to the data area of lower 1 word (2-byte). Use the WriteDeviceRandom or WriteDeviceBlock function to write double word devices.
- Digit specified bit device and index setting cannot be used.
- If the function is run against the safety device in the RnSF CPU safety mode, an error code 0x010A42A5" (an operation that can not be carried out in safety mode was performed) is returned.

How to specify labels

The following describes how to specify labels.

For the system label name, the following type class, array element, and label data type can be specified.

When specifying a system label name in a program, use the following formats.

○: Available, ×: Not available, —: Not applicable

Label definition in Label Utility		Label specification method in a program			
Type class	Array element	Label name format	Availability	Data type	
Simple Types	None	System label name	○	Short/short (16-bit value)	
		System label name [n1]	○	Short/short (16-bit value)	
	Element (One Dimension)	System label name	×	—	
		System label name [n1] [n2]	○	Short/short (16-bit value)	
	Element (Two Dimensions)	System label name	×	—	
		System label name [n1] [n2]	○	Short/short (16-bit value)	
	Element (Three Dimensions)	System label name	×	—	
		System label name [n1] [n2] [n3]	○	Short/short (16-bit value)	
	Structure	None	System label name	×	—
			System label name.structure member name	○	Short/short (16-bit value)
		Element (One Dimension)	System label name	×	—
			System label name [n1]	×	—
System label name [n1].structure member name			○	Short/short (16-bit value)	
Element (Two Dimensions)		System label name	×	—	
		System label name [n1] [n2]	×	—	
		System label name [n1] [n2].structure member name	○	Short/short (16-bit value)	
Element (Three Dimensions)		System label name	×	—	
		System label name [n1] [n2] [n3]	×	—	
		System label name [n1] [n2] [n3].structure member name	○	Short/short (16-bit value)	

■Data type

The following data type can be specified for a label.

Bit, Word (Signed), Double Word (Signed)^{*1}, Word (Unsigned), Double Word (Unsigned)^{*1}, Float (Single Precision)^{*1}, Float (Double Precision)^{*1}, String, String (Unicode)^{*1*2}, Time^{*1}, Timer, Timer (Unsigned), Long timer^{*2}, Counter, Counter (Unsigned), Long counter^{*2}, Retentive Timer, Retentive Timer (Unsigned), Long retentive timer^{*2}

*1 Only the lower 2-byte of start device can be written.

Use the WriteDeviceRandom function to write 2 words or more of devices.

*2 Cannot be used for system label Ver.1.

How to specify devices

Specify the device values to be written as follows:

■When specifying bit devices

Example: Write data to M0.

· Label setting

szLabel	Label name	Data type	Device
[0]	LABEL	Bit	M0

· Device values to be written

2-byte	Applicable label
M0*1	LABEL

*1 The target device to which data is written is only one point of "M0," and the least significant bit of the set 2-byte data is written as the device value.

■When specifying word devices

Example: Write data to D0.

· Label setting

szLabel	Label name	Data type	Device
[0]	LABEL	Word	D0

· Device values to be written

2-byte	Applicable label
D0	LABEL

■When specifying CN200 and later devices of FXCPU (double word devices to which data of 2 words can be written for 1 point)

Example: Write data to CN200.

For CN200 and later devices of FXCPU, 4-byte is written.

Only the value of lower 2-byte can be set.

'0' is written to the H (upper 2-byte) of specified devices.

· Label setting

szLabel	Label name	Data type	Device
[0]	LABEL	Double Word	CN200

· Device values to be written

2-byte	Applicable label
L of CN200	LABEL

■When specifying double word devices (ones to which data of 2 words cannot be written for 1 point)

Example: Write data by specifying D0.

· Label setting

szLabel	Label name	Data type	Device
[0]	LABELW	Double Word	D0

· Number of write points: 1

· Device values to be written

2-byte	Applicable label
D0*1	LABELW

*1 Data is written in order from the lower 2 bytes.

■When specifying data types equivalent to 2 words or more for labels

Example: Write data by specifying a string type label.

· Label setting

szLabel	Label name	Data type	Device
[0]	LABELW	String	D0

· Device values to be written

2-byte	Applicable label
D0*1	LABELW

*1 Data is written in order from the lower 2 bytes.

GetDevice2 (Acquiring device data)

Applicable controls

This function can be used for the following controls.

- 32-bit: DotUtilType
- 64-bit: DotUtilType64

Feature

To acquire one point of device in 2-byte data unit.

Format

■Visual Basic .NET

`IRet = object.GetDevice2(szLabel, sData)`

Data type	Argument	Description	Input/Output
Integer	IRet	Returned value	Output
String	szLabel	Label name	Input
Short	sData	Read device value	Output

■Visual C#

`iRet = object.GetDevice2(ref szLabel, ref sData)`

Data type	Argument	Description	Input/Output
int	iRet	Returned value	Output
String	szLabel	Label name	Input
short	sData	Read device value	Output

Returned value

Normal termination: '0' is returned.

Abnormal termination: Any value other than '0' is returned. (🔗 Page 616 ERROR CODES)

Description

- One point of device data specified for szLabel (label name) is stored in sData (read device values).

Point

- When a device which corresponds to the label name does not exist, an error occurs and data cannot be read.
- For the read device values, reserve 2 bytes of memory area.
If the memory area is not reserved, a critical error (an application error or the like) may occur.
- When specifying a label equivalent to 2 words or more, this function stores only the data of lower 1 word (2-byte). (An error does not occur.) Use the ReadDeviceRandom or ReadDeviceBlock function to read labels equivalent to 2 words or more.
- Digit specified bit device and index setting cannot be used.

How to specify labels

The following describes how to specify labels.

For the system label name, the following type class, array element, and label data type can be specified.

When specifying a system label name in a program, use the following formats.

○: Available, ×: Not available, —: Not applicable

Label definition in Label Utility		Label specification method in a program			
Type class	Array element	Label name format	Availability	Data type	
Simple Types	None	System label name	○	Short/short (16-bit value)	
		System label name [n1]	○	Short/short (16-bit value)	
	Element (One Dimension)	System label name	×	—	
		System label name [n1]	○	Short/short (16-bit value)	
	Element (Two Dimensions)	System label name	×	—	
		System label name [n1] [n2]	○	Short/short (16-bit value)	
	Element (Three Dimensions)	System label name	×	—	
		System label name [n1] [n2] [n3]	○	Short/short (16-bit value)	
	Structure	None	System label name	×	—
			System label name.structure member name	○	Short/short (16-bit value)
		Element (One Dimension)	System label name	×	—
			System label name [n1]	×	—
System label name [n1].structure member name			○	Short/short (16-bit value)	
Element (Two Dimensions)		System label name	×	—	
		System label name [n1] [n2]	×	—	
		System label name [n1] [n2].structure member name	○	Short/short (16-bit value)	
Element (Three Dimensions)		System label name	×	—	
		System label name [n1] [n2] [n3]	×	—	
		System label name [n1] [n2] [n3].structure member name	○	Short/short (16-bit value)	

■Data type

The following data type can be specified for a label.

Bit, Word (Signed), Double Word (Signed)^{*1}, Word (Unsigned), Double Word (Unsigned)^{*1}, Float (Single Precision)^{*1}, Float (Double Precision)^{*1}, String^{*1}, String (Unicode)^{*1,2}, Time^{*1}, Timer, Timer (Unsigned), Long timer^{*2}, Counter, Counter (Unsigned), Long counter^{*2}, Retentive Timer, Retentive Timer (Unsigned), Long retentive timer^{*2}

*1 Only the lower 2-byte of start device can be read.

Use the ReadDeviceRandom function to read 2 words or more of devices.

*2 Cannot be used for system label Ver.1.

How to specify devices

The read device values are stored as follows:

■When specifying bit devices

Example: Read data from M0.

· Label setting

szLabel	Label name	Data type	Device
[0]	LABEL	Bit	M0

· Read device values

2-byte	Applicable label
M0*1	LABEL

*1 The target device from which data is read is only one point of "M0," and '0' or '1' is stored as the device value.

■When specifying word devices

Example: Read data from D0.

· Label setting

szLabel	Label name	Data type	Device
[0]	LABEL	Word	D0

· Read device values

2-byte	Applicable label
D0	LABEL

■When specifying data types equivalent to 2 words or more for labels

Example: Read data by specifying a string type label.

· Label setting

szLabel	Label name	Data type	Device
[0]	LABEL	String	D0

· Read device values

2-byte	Applicable label
D0*1	LABEL

*1 Only the lower 2-byte is read to the start device.

Dispose (release memory)

Applicable controls

This function can be used for the following controls.

- 64-bit: DotUtilType64

Feature

To end the control for 64-bit (DotUtilType64_Server.exe).

Format

■ Visual Basic .NET

object.Dispose()

■ Visual C#

object.Dispose()

Returned value

None

Description

- The control used for 64-bit access (DotUtilType64_Server.exe) ends.

Point

This function must be executed before discarding DotUtilType instances. Otherwise, DotUtilType64_Server.exe cannot be ended.

12 SAMPLE PROGRAMS

This chapter explains the sample programs registered at the installation of MX Component.

Precautions

- The sample programs are stored in (installation folder)\Samples at the installation. When executing the sample files, copy them to any folder. When using the sample programs of VC++, also copy the include file.
- Sample programs are provided as a reference for creating user application programs. Use the programs with your responsibility.

■Error at building sample programs

- When the message "System.Runtime.InteropServices.COMException was not handled." appears, specify "x86" (32-bit) for the target CPU when creating a program using MX Component.
- When using MX Component in a 64-bit program, use a control for 64-bit.

Sample program list

The following table shows the list of sample programs registered under (installation folder)\Samples when installing MX Component.

■Supported language: Visual Basic.NET

Folder name	Sample program name	Target Control	Reference
Vb.Net\Act	Visual Studio 2019: Act_2019.sln Visual Studio 2022: Act_2022.sln	ActUtilType ActProgType ActSupportMsg ActUtilType64 ActProgType64 ActSupportMsg64	Page 605 Act_2019.sln, Dot_2019.sln
Vb.Net\ActDatalogging	Visual Studio 2019: ActDatalogging_2019.sln Visual Studio 2022: ActDatalogging_2022.sln	ActUtilDataLogging ActProgDataLogging ActUtilDataLogging64 ActProgDataLogging64	Page 608 ActDatalogging_2019.sln
Vb.Net\Dot	Visual Studio 2019: Dot_2019.sln Visual Studio 2022: Dot_2022.sln	DotUtilType DotUtilType64	Page 605 Act_2019.sln, Dot_2019.sln

■Supported language: Visual C#

Folder name	Sample program name	Target Control	Reference
Vcs.NET\Act	Visual Studio 2019: Act_2019.sln Visual Studio 2022: Act_2022.sln	ActUtilType ActProgType ActSupportMsg ActUtilType64 ActProgType64 ActSupportMsg64	Page 605 Act_2019.sln, Dot_2019.sln
Vcs.NET\ActDatalogging	Visual Studio 2019: ActDatalogging_2019.sln Visual Studio 2022: ActDatalogging_2022.sln	ActUtilDataLogging ActProgDataLogging ActUtilDataLogging64 ActProgDataLogging64	Page 608 ActDatalogging_2019.sln
Vcs.NET\Dot	Visual Studio 2019: Dot_2019.sln Visual Studio 2022: Dot_2022.sln	DotUtilType DotUtilType64	Page 605 Act_2019.sln, Dot_2019.sln

■Supported language: Visual C++

Folder name	Sample program name	Target Control	Reference
Vc\Act	Visual Studio 2019: Act_2019.sln Visual Studio 2022: Act_2022.sln	ActUtilType ActProgType ActSupportMsg ActUtilType64 ActProgType64 ActSupportMsg64	Page 605 Act_2019.sln, Dot_2019.sln
Vc\ActDataLogging	Visual Studio 2019: ActDataLogging_2019.sln Visual Studio 2022: ActDataLogging_2022.sln	ActUtilDataLogging ActProgDataLogging ActUtilDataLogging64 ActProgDataLogging64	Page 608 ActDataLogging_2019.sln

■Supported language: VBA (Access)

Folder name	Sample program name	Target Control	Reference
VBA\Access	For EarlyBind: ActTest_EarlyBind.accdb ^{*1} For LateBind: ActTest_LateBind.accdb ^{*2}	ActUtilType ActProgType ActSupportMsg	Page 609 For Access (ActTest_EarlyBind.accdb, ActTest_LateBind.accdb, ActTest64.accdb)
	For LateBind: ActTest64.accdb ^{*2}	ActUtilType64 ActProgType64 ActSupportMsg64	Page 609 For Access (ActTest_EarlyBind.accdb, ActTest_LateBind.accdb, ActTest64.accdb)

*1 A sample to use by pasting it to a form

*2 A sample to use by creating an object

■Supported language: VBA (Excel)

Folder name	Sample program name	Target Control	Reference
VBA\Excel	For EarlyBind and LateBind: ActUtilType.xlsm ^{*1*2}	ActUtilType ActSupportMsg	Page 613 For Excel (ActUtilType.xlsm, ActUtilType64.xlsm, ActProgType.xlsm, and ActProgType64.xlsm)
	For EarlyBind and LateBind: ActProgType.xlsm ^{*1*2}	ActProgType	
	For LateBind: ActUtilType64.xlsm ^{*2}	ActUtilType64 ActSupportMsg64	Page 613 For Excel (ActUtilType.xlsm, ActUtilType64.xlsm, ActProgType.xlsm, and ActProgType64.xlsm)
	For LateBind: ActProgType64.xlsm ^{*2}	ActProgType64	

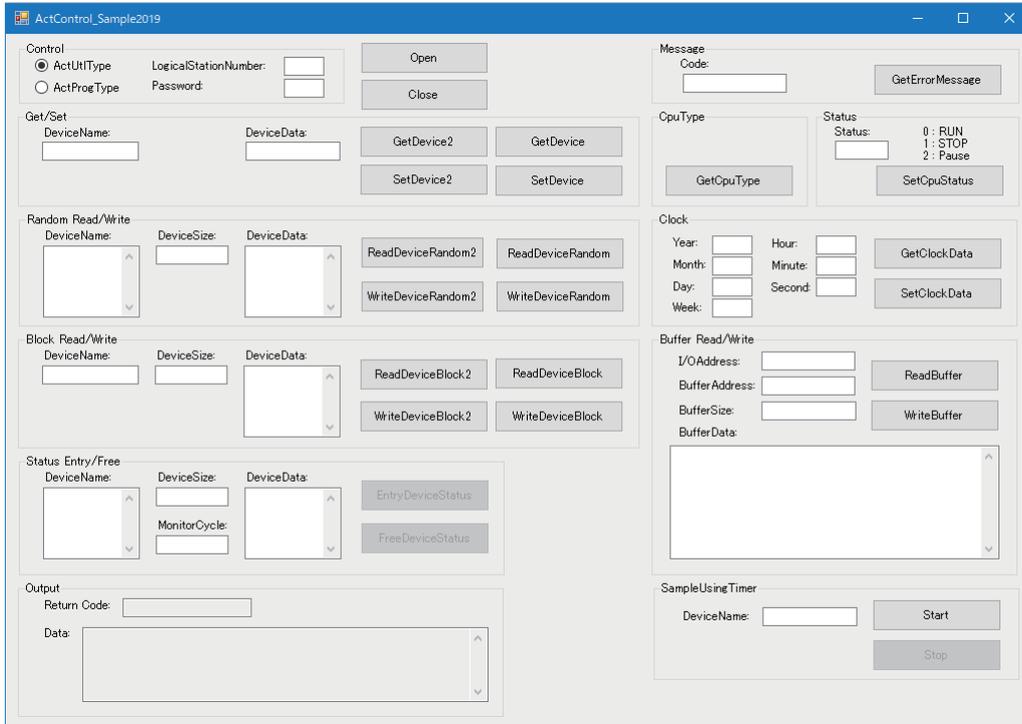
*1 A sample to use by pasting it to a form

*2 A sample to use by creating an object

12.1 Visual Basic.NET, Visual C#, Visual C++

Act_2019.sln, Dot_2019.sln

Window



Displayed items

Frame	Select/Input item	Description
Control	ActUtilType, ActProgType ^{*1}	Select the control to be used.
	LogicalStationNumber	Enter the logical station number which was specified for the communication setting in Communication Settings Utility.
	Password	Enter a password when using a CPU which requires a password.
	[Open] button	Click this to open a communication line.
	[Close] button	Click this to close a communication line.
Get/Set	DeviceName/LabelName ^{*2}	Enter a device/label name for acquiring/setting devices or labels.
	DeviceData	Enter a device value for setting devices.
	[GetDevice2] button	Click this to acquire device data from a CPU module by executing the GetDevice2 with the data entered in the "DeviceName"("LabelName") column in the same frame. ^{*3}
	[SetDevice2] button	Click this to set device data to a CPU module by executing the SetDevice2 with the data entered in the "DeviceName"("LabelName") column and "DeviceData" column in the same frame.
	[GetDevice] button	Click this to acquire device data from a CPU module by executing the GetDevice with the data entered in the "DeviceName"("LabelName") column in the same frame. ^{*3}
	[SetDevice] button	Click this to set device data to a CPU module by executing SetDevice with the data the entered in the "DeviceName"("LabelName") column and "DeviceData" column in the same frame.

Frame	Select/Input item	Description
Random Read/Write	DeviceName/LabelName ^{*2}	Enter a device/label name for reading/writing devices or labels randomly.
	DeviceSize	Enter the number of device points for reading/writing devices randomly.
	DeviceData	Enter the device value for writing devices randomly.
	[ReadDeviceRandom2] button	Click this to acquire device data from a CPU module by executing the ReadDeviceRandom2 with the data entered in the "DeviceName"("LabelName") column and "DeviceSize" column in the same frame. ^{*3}
	[WriteDeviceRandom2] button	Click this to set device data to a CPU module by executing the WriteDeviceRandom2 with the data entered in the "DeviceName"("LabelName") column, "DeviceSize" column, and "DeviceData" column in the same frame.
	[ReadDeviceRandom] button	Click this to acquire device data from a CPU module by executing the ReadDeviceRandom with the data entered in the "DeviceName"("LabelName") column and "DeviceSize" column in the same frame. ^{*3}
	[WriteDeviceRandom] button	Click this to set device data to a CPU module by executing WriteDeviceRandom with the data entered in the "DeviceName"("LabelName") column and "DeviceSize" column, and "DeviceData" column in the same frame.
Block Read/Write	DeviceName/LabelName ^{*2}	Enter a device/label name for reading/writing devices or labels in batch.
	DeviceSize	Enter the number of device points for reading/writing devices in batch.
	DeviceData	Enter a device value for writing devices in batch.
	[ReadDeviceBlock2] button	Click this to acquire device data from a CPU module by executing the ReadDeviceBlock2 with the data entered in the "DeviceName"("LabelName") column and "DeviceSize" column in the same frame. ^{*3}
	[WriteDeviceBlock2] button	Click this to set device data to a CPU module by executing the WriteDeviceBlock2 with the data entered in the "DeviceName"("LabelName") column, "DeviceSize" column, and "DeviceData" column in the same frame.
	[ReadDeviceBlock] button	Click this to acquire device data from a CPU module by executing the ReadDeviceBlock with the data entered in the "DeviceName"("LabelName") column and "DeviceSize" column in the same frame. ^{*3}
	[WriteDeviceBlock] button	Click this to set device data to a CPU module by executing the WriteDeviceBlock with the data entered in the "DeviceName"("LabelName") column and "DeviceSize" column, and "DeviceData" column in the same frame.
Status Entry/Free ^{*4}	DeviceName/LabelName ^{*2}	Enter a device/label name for monitoring device status.
	DeviceSize	Enter the number of devices for monitoring device status.
	MonitorCycle	Enter the device status monitoring interval.
	DeviceData	Enter a registered device value for monitoring devices.
	[EntryDeviceStatus] button	Click this to register device monitoring by executing the EntryDeviceStatus with the data entered in the "DeviceName"("LabelName") column and "DeviceSize" column, "MonitorCycle" column and "DeviceData" column in the same frame.
	[FreeDeviceStatus] button	Click this to deregister a device that is registered using the EntryDeviceStatus function to monitor its status.
Output	ReturnCode	Displays the execution result of each method (function).
	Data	Display the read device values.
Message	Code	Enter a message code.
	[GetErrorMessage] button	Click this to acquire a message by executing the GetErrorMessage with the data entered in "the Code" column in the same frame. ^{*3}
CpuType	[GetCpuType] button	Click this to acquire the model character and the model code (hexadecimal) from a CPU module by executing the GetCpuType. ^{*3}
Status	Status	Enter a remote control to be set.
	[SetCpuStatus] button	Click this to set a remote control to a CPU module by executing the SetCpuStatus with the data entered in the "Status" column in the same frame.

Frame	Select/Input item	Description
Clock	Year	Enter a year to be set for a CPU module.
	Month	Enter a month to be set for a CPU module.
	Day	Enter a day to be set for a CPU module.
	Week	Enter a day of the week to be set for a CPU module.
	Hour	Enter an hour to be set for a CPU module.
	Minute	Enter a minute to be set for a CPU module.
	Second	Enter a second to be set for a CPU module.
	[GetClockData] button	Click this to acquire the clock data from a CPU module. ^{*3}
	[SetClockData] button	Click this to set the clock data to a CPU module with the data entered in the "Year" column, "Month" column, "Day" column, "Week" column, "Hour" column, "Minute" column, and "Second" column in the same frame.
Buffer Read/Write	I/OAddress	Enter an I/O address for reading/writing buffer memories.
	BufferAddress	Enter an address for reading/writing buffer memories.
	BufferSize	Enter a size for reading/writing buffer memories.
	BufferData	Enter a data for reading/writing buffer memories.
	[ReadBuffer] button	Click this to acquire device data from a CPU module by executing the ReadBuffer with the data entered in the "I/OAddress" column, "BufferAddress" column, and "BufferSize" column in the same frame. ^{*3}
	[WriteBuffer] button	Click this to set device data to a CPU module by executing the WriteBuffer with the data entered in "I/OAddress" column, "BufferAddress" column, "BufferSize" column, and "BufferData" column in the same frame.
Sample Using Timer	DeviceName/LabelName ^{*2}	Enter a device/label name for reading devices or labels.
	[Start] button	Click this to acquire device data from a CPU module by executing the ReadDeviceRandom with the data entered in the "DeviceName"("LabelName") column in the same frame. ^{*3} ReadDeviceRandom is executed every five seconds until the [Stop] button is clicked.
	[Stop] button	Click this to stop the processing started by clicking the [Start] button in the same frame.

*1 Appears in Act_2019.sln only. Not appears in Dot_2019.sln.

*2 For Act_2019.sln, DeviceName is displayed, and for Dot_2019.sln, LabelName is displayed.

*3 The acquisition result is displayed in the "Data" column in the "Output" frame.

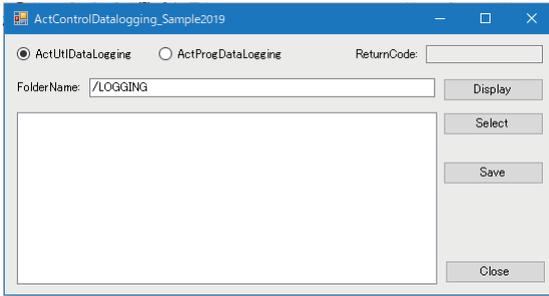
*4 Not be displayed in the sample programs of Visual C++.

Operating procedure

1. Select a control to be used (ActUtilType or ActProgType) in "Control."
2. Enter a logical number in the "LogicalStationNumber" column.
3. Click the [Open] button to open a communication line.
4. Click the button of a method (function) to execute the function after entering the data which is an argument of method (function) in each item.
The read device value is displayed in the "Data" on "Output," and the execution result of each method (function) is displayed in the "ReturnCode" column.
5. Click the [Close] button on "Control" to close the communication line.

ActDatalogging_2019.sln

Window



Displayed items

Item	Description
ActUtilDataLogging /ActProgDataLogging	Select the control to be used.
ReturnCode	Displays the execution result of each method.
FolderName	Enter a name of a folder to be accessed.
[Display] button	Click this to update the content displayed in "File list" column.
[Select] button	Click this to apply a folder name selected in the "File list" column to "FolderName." When a file name has already selected, a logging file is saved in a personal computer.
[Save] button	Click this to display "Save As" screen. Only a text file (*.txt) can be selected in "Save as type."
[Close] button	Click this to exit the sample program.
File list	Displays files or folders in the path specified for "Folder" in a list. For a folder name, a slash '/' is prefixed.

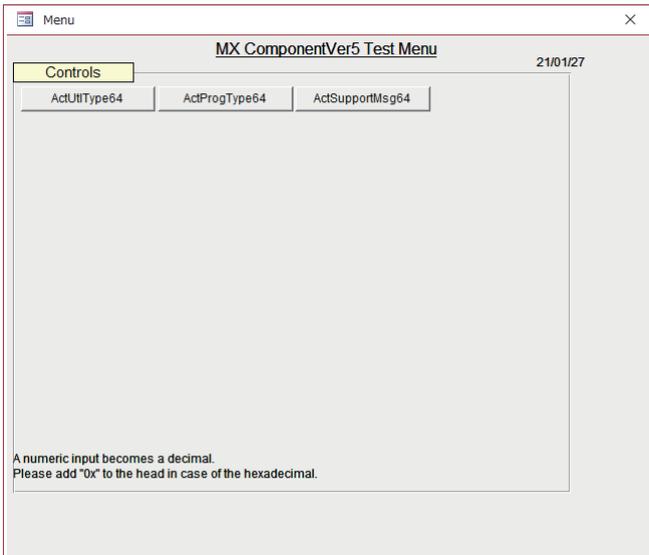
12.2 VBA

This section explains the VBA sample programs for Access and Excel.

For Access (ActTest_EarlyBind.accdb, ActTest_LateBind.accdb, ActTest64.accdb)

Menu window

Window



Displayed items

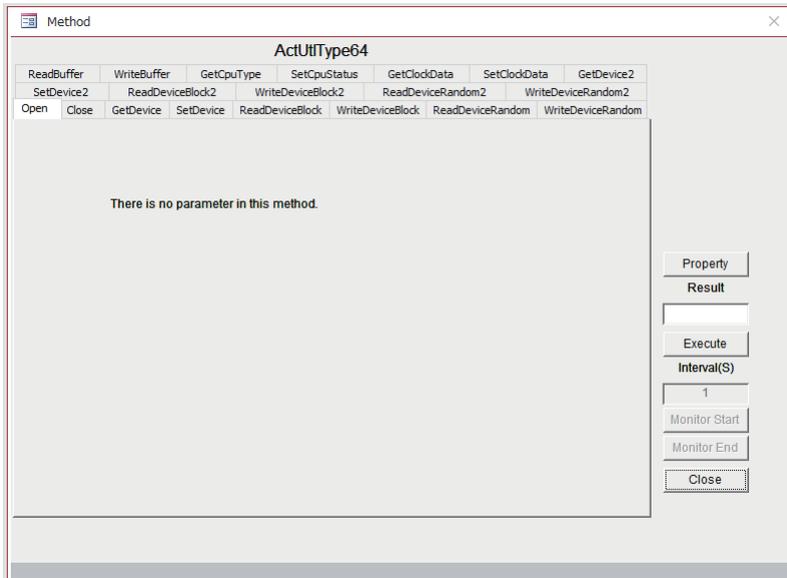
Item name	Description
[ActUtilType] button*1	Click this to open the Method screen for the ActUtilType control (for ActUtilType64 control).
[ActUtilType64] button*2	
[ActProgType] button*1	Click this to open the Method screen for the ActProgType control (for (ActProgType64 control).
[ActProgType64] button*2	
[ActSupportMsg] button*1	Click this to open the Method screen for the ActSupportMsg control (for ActSupportMsg64control).
[ActSupportMsg64] button*2	

*1 When using ActTest_EarlyBind.accdb, ActTest_LateBind.accdb

*2 When using ActTest64.accdb

Method window

Window



Displayed items

Tab name	Item	Description
—	[Property] button	Click this to set the properties.
—	Result	Displays the execution result of each method (function).
—	[Execute] button	Click this to execute a method (function) in the selected tab.
—	Interval(S)	Displays a time interval for monitoring.
—	[Monitor Start] button	Click this to start the monitoring of the ReadDeviceBlock/ ReadDeviceBlock2. A method (function) is executed in the time interval displayed in the "Interval(S)," and data is acquired.
—	[Monitor End] button	Click this to stop the monitoring of the ReadDeviceBlock/ ReadDeviceBlock2.
—	[Close] button	Click this to return to the menu window.
Open	—	—
Close	—	—
GetDevice, GetDevice2	Device Name	Enter a device name for acquiring devices.
	Device Data	Displays an acquired device value.
SetDevice, SetDevice2	Device Name	Enter a device name for setting devices.
	Device Data	Enter a device value for setting devices.
ReadDeviceRandom, ReadDeviceRandom2	Device Name	Enter a device name for reading devices randomly.
	Device Points	Enter the number of device points for reading devices randomly.
	Data file name	Specify a save destination binary file name when saving the read data.
	Dump	When this checkbox is selected, displays the read data.
WriteDeviceRandom, WriteDeviceRandom2	Device Name	Enter a device name for writing devices randomly.
	Device Points	Enter the number of device points for writing devices randomly.
	Written device value	Enter the device value for writing devices randomly. Data can be entered automatically by selecting a data not specified.
ReadDeviceBlock, ReadDeviceBlock2	Device Name	Enter a device name for reading devices in batch.
	Device Points	Enter the number of device points for reading devices in batch.
	Data file name	Specify a save destination binary file name when saving the read data.
	Dump	When this checkbox is selected, displays the read data.

Tab name	Item	Description
WriteDeviceBlock, WriteDeviceBlock2	Device Name	Enter a device name for writing devices in batch.
	Device Points	Enter the number of device points for writing devices in batch.
	Written device value	Enter the device value for writing devices randomly. Data can be entered automatically by selecting a data not specified.
EntryDeviceStatus ^{*1}	Device Name	Enter a device name for monitoring device status.
	Points	Enter the number of devices for monitoring device status.
	Interval	Enter the device status monitoring interval.
	Device value	Enter a registered device value for monitoring devices.
FreeDeviceStatus ^{*1}	—	—
GetCpuType	PLC CPU type	Displays an acquired CPU type name.
	PLC CPU type code	Displays an acquired CPU type code.
SetCpuStatus	Operation	Enter a remote control to be set.
GetClockData, SetClockData	Year	Enter/displays a year to be set for a CPU module.
	Month	Enter/displays a month to be set for a CPU module.
	Day	Enter/displays a day to be set for a CPU module.
	WeekDay	Enter/displays a day of the week to be set for a CPU module.
	Hour	Enter/displays an hour to be set for a CPU module.
	Minute	Enter/displays a minute to be set for a CPU module.
	Second	Enter/displays a second to be set for a CPU module.
ReadBuffer	First I/O address	Enter an I/O address for reading buffer memories.
	Buffer memory address	Enter an address for reading buffer memories.
	Read size	Enter a size for reading buffer memories.
	Data file name	Specify a save destination binary file name when saving the read data.
	Dump	When this checkbox is selected, displays the read data.
WriteBuffer	First I/O address	Enter an I/O address for writing buffer memories.
	Buffer memory address	Enter an address for writing buffer memories.
	Write size	Enter a size for writing buffer memories.
	Values written to buffer memory	Enter a device value for writing buffer memories. Data can be entered automatically by selecting a data not specified.

*1 Displayed only when using ActTest_EarlyBind.accdb.

Operating procedure

1. Click the [Property] button to set the properties.
2. Select [Open] tab, and click the [Execute] button to open a communication line.
3. Select the tab of a method (function) to be executed, and click the [Execute] button after entering the data which is an argument.
 When the checkbox of "Dump" is selected, the read device value is displayed at the bottom in the screen.
 The execution result of each method (function) is displayed in the "Result" column.
4. Select [Close] tab, and click the [Execute] button to close the communication line.

GetErrorMessage screen

Window

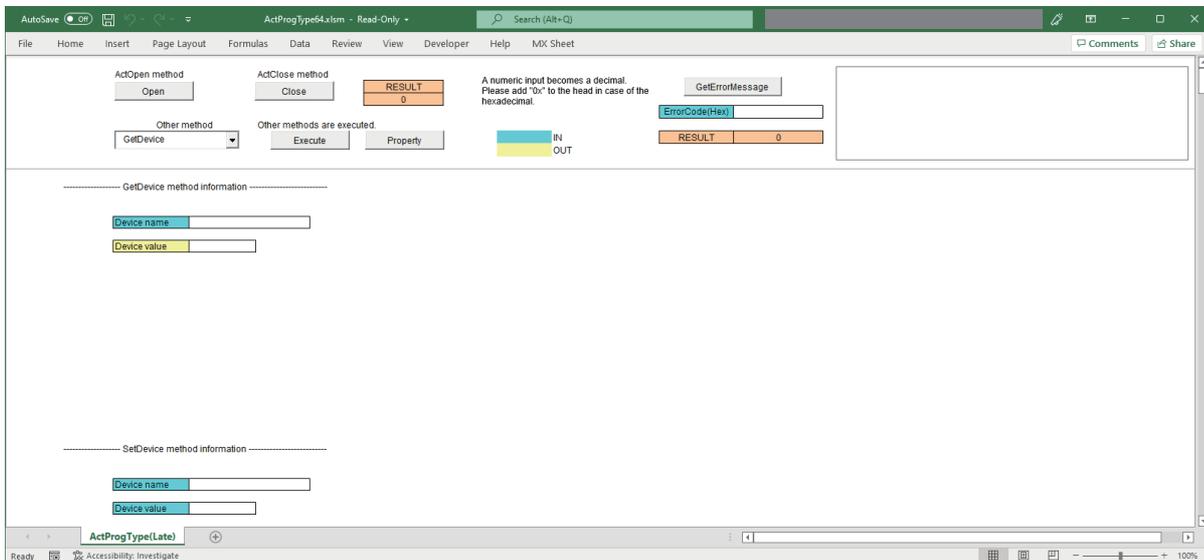


Displayed items

Item name	Description
Error code	Enter a message code.
Trouble shooting	Displays an acquired message.
Result	Displays an execution result of the GetErrorMessage.
[Execute] button	Click this to execute the GetErrorMessage with the data entered in "Error code" column.
[Close] button	Click this to return to the menu window.

For Excel (ActUtilType.xlsm, ActUtilType64.xlsm, ActProgType.xlsm, and ActProgType64.xlsm)

Window



Displayed items

Method (function)	Item	Description
Open	[Open] button	Click this to open a communication line.
Close	[Close] button	Click this to close a communication line.
—	RESULT	Displays the execution result of each method (function).
—	Other method	Select a method (function) to which the cell moves from the combo box.
—	[Execute] button	Click this to execute a method (function) selected in the combo box based on the input data.
—	[Property] button	Click this to set a property value of the ActUtilType/ActProgType.
—	Result	Displays the execution result of each method (function).
GetDevice, GetDevice2	Device name	Enter a device name for acquiring devices.
	Device value	Displays an acquired device value.
SetDevice, SetDevice2	Device name	Enter a device name for setting devices.
	Device value	Enter a device value for setting devices.
ReadDeviceRandom, ReadDeviceRandom2	Device name	Enter a device name for reading devices randomly.
	Points	Enter the number of device points for reading devices randomly.
	Data file name	Specify a save destination binary file name when saving the read data.
	Dump	When this checkbox is selected, displays the read data.
WriteDeviceRandom, WriteDeviceRandom2	Device name	Enter a device name for writing devices randomly.
	Points	Enter the number of device points for writing devices randomly.
	Device value	Enter the device value for writing devices randomly. Data can be entered automatically by selecting a data not specified.
ReadDeviceBlock, ReadDeviceBlock2	Device name	Enter a device name for reading devices in batch.
	Points	Enter the number of device points for reading devices in batch.
	Data file name	Specify a save destination binary file name when saving the read data.
	Dump	When this checkbox is selected, displays the read data.
	Interval(sec)	Enter a time interval for monitoring (second).
	Start monitor	Click this to start monitoring of a method (function). The method (function) is executed in the time interval displayed in "Interval(sec)," and the data is acquired.

Method (function)	Item	Description
WriteDeviceBlock, WriteDeviceBlock2	Device name	Enter a device name for writing devices in batch.
	Points	Enter the number of device points for writing devices in batch.
	Device value	Enter the device value for writing devices randomly. Data can be entered automatically by selecting a data not specified.
EntryDeviceStatus* ¹	Device name	Enter a device name for monitoring device status.
	Points	Enter the number of devices for monitoring device status.
	Interval(sec)	Enter the device status monitoring interval (second).
	Device value	Enter a registered device value for monitoring devices.
OnDeviceStatus* ¹	Device and value to which condition consists.	Displays a result of the OnDeviceStatus event.
FreeDeviceStatus* ¹	—	—
GetCpuType	CPU type	Displays an acquired CPU type name.
	CPU type code	Displays an acquired CPU type code.
SetCpuStatus	Operation	Enter a remote control to be set.
GetClockData, SetClockData	Year	Enter/displays a year to be set for a CPU module.
	Month	Enter/displays a month to be set for a CPU module.
	Day	Enter/displays a day to be set for a CPU module.
	Day-of-week	Enter/displays a day of the week to be set for a CPU module.
	Hour	Enter/displays an hour to be set for a CPU module.
	Minute	Enter/displays a minute to be set for a CPU module.
	Second	Enter/displays a second to be set for a CPU module.
ReadBuffer	First I/O address	Enter an I/O address for reading buffer memories.
	Buffer memory address	Enter an address for reading buffer memories.
	Read size	Enter a size for reading buffer memories.
	Data file name	Specify a save destination binary file name when saving the read data.
	Dump	When this checkbox is selected, displays the read data.
WriteBuffer	First I/O address	Enter an I/O address for writing buffer memories.
	Buffer memory address	Enter an address for writing buffer memories.
	Write size	Enter a size for writing buffer memories.
	Device value	Enter a device value for writing buffer memories. Data can be entered automatically by selecting a data not specified.
GetErrorMessage	[GetErrorMessage] button	Click this to execute the GetErrorMessage based on the data input in "ErrorCode(Hex)." An acquired message is output to the text right to the button.
	ErrorCode(Hex)	Enter a message code.
	RESULT	Displays an execution result of the GetErrorMessage.

*1 Displayed only when using a control by pasting it to a form (EarlyBind)

Point

When using a control by pasting it to a form (EarlyBind) and creating an object (LateBind), switch between [Early binding] and [Late binding] of [Select the binding] in the [Add-Ins] tab of Excel.

Operating procedure

1. Click the [Property] button to set the properties.
2. Click the [Open] button to open a communication line.
3. Select a method (function) to be executed in "Other method" and click the [Execute] button after entering the data which is an argument of method (function).

When the checkbox of "Dump" is selected, the read device value is displayed.

The execution result of each method (function) is displayed in the "RESULT" column.

13 ERROR CODES

This chapter explains the error codes returned by controls and the error codes returned by CPUs, modules, and network boards.

13.1 Error Code List

The following table shows the error codes.

Error code	Error description	Corrective action
0x00000000	Normal end	-
0x01010002	Time-out error	<ul style="list-style-type: none"> • Check the property timeout value. • Check the settings in the communication settings utility. • Check the CPU module, unit settings, state of the cable, etc. • Close and Open again. • Exit the program and restart the personal computer.
0x01010005	Message error	<ul style="list-style-type: none"> • Check the system noise. • Check the property timeout value. • Check the settings in the communication settings utility. • Check the CPU module, unit settings, state of the cable, etc. • Close and Open again. • Exit the program and restart the personal computer.
0x01010010	Programmable controller No. error Communication could not be made with the specified station number.	<ul style="list-style-type: none"> • Check the station number set on the communication settings utility. • Check the station number set to ActStationNumber.
0x01010011	Mode error Command not supported.	<ul style="list-style-type: none"> • Check if the correct CPU type setting is done. • Check the CPU module, unit settings, status of the cable, etc. • Exit the program and restart the personal computer. • Reinstall MX Component. • Please contact your local Mitsubishi Electric sales office or representative.
0x01010012	Special Unit Specification error	<ul style="list-style-type: none"> • Check the specified address of the special unit.
0x01010013	Other data error Communication cannot be made for some cause.	<ul style="list-style-type: none"> • Check that the system configuration is not an unsupported configuration. • Check if the correct CPU type setting is done. • Exit the program and restart the personal computer. • Please contact your local Mitsubishi Electric sales office or representative.
0x01010018	Remote request error Remote operation is being performed in a route different from the communicating route.	<ul style="list-style-type: none"> • Cancel the remote operation being performed in another route.
0x01010020	Link error Link communications could not be made.	<ul style="list-style-type: none"> • Check that reset operation is not performed for the other end of communication, the control station (master station) or the station passed through by routing. • Check that the network parameter setting is correct.
0x01010021	Special Unit Bus error There is no response from the special unit under consideration.	<ul style="list-style-type: none"> • Repair or exchange the special unit under consideration. • Exit the program and restart the personal computer. • Reinstall MX Component. • Please contact your local Mitsubishi Electric sales office or representative.
0x01800001	No command error The method does not support.	<ul style="list-style-type: none"> • The corresponding method does not support.
0x01800002	Memory lock error	<ul style="list-style-type: none"> • Exit the program and restart the personal computer. • When using an interface board for personal computer, increase the minimum working set size of the personal computer.*1
0x01800003	Memory securing error	<ul style="list-style-type: none"> • Exit the program and restart the personal computer. • Exit other programs and secure free memory area. • When using an interface board for personal computer, increase the minimum working set size of the personal computer.*1
0x01800004	DLL load error	<ul style="list-style-type: none"> • Exit the program and restart the personal computer. • Exit other programs and secure free memory area. • Reinstall MX Component.
0x01800005	Resource securing error	<ul style="list-style-type: none"> • Exit the program and restart the personal computer. • Exit other programs and secure free memory area.

Error code	Error description	Corrective action
0x01801001	Resource Timeout error The resource could not be retrieved within the specified time.	<ul style="list-style-type: none"> Execute again after the other object completes the communication. Execute again after increasing the timeout value. Exit the program and restart the personal computer. Reinstall MX Component.
0x01801002	Multi-line open error	<ul style="list-style-type: none"> Exit the program and restart the personal computer.
0x01801003	Open not yet executed	
0x01801004	Open Type error	<ul style="list-style-type: none"> Exit the program and restart the personal computer. Reinstall MX Component. Please contact your local Mitsubishi Electric sales office or representative.
0x01801005	Specified port error	<ul style="list-style-type: none"> Exit the program and restart the personal computer. Reinstall MX Component.
0x01801006	Specified module error	<ul style="list-style-type: none"> Check that the actual system configuration matches to the settings in the communication settings utility or the values of the properties. Exit the program and restart the personal computer. Reinstall MX Component. Please contact your local Mitsubishi Electric sales office or representative.
0x01801007	Specified CPU error	<ul style="list-style-type: none"> Check the CPU type set to ActCpuType. Check that the system configuration is not an unsupported configuration. Exit the program and restart the personal computer. Reinstall MX Component. Check the packet type set to ActPacketType.
0x01801008	Target station access error	<ul style="list-style-type: none"> Review the target station.
0x01801009	Registry open failure Failed while opening data key of the registry.	<ul style="list-style-type: none"> Exit the program and restart the personal computer. Reinstall MX Component. Please contact your local Mitsubishi Electric sales office or representative.
0x0180100A	Packet Type error The packet type specified is incorrect.	<ul style="list-style-type: none"> Recheck the ActPacketType. Exit the program and restart the personal computer. Reinstall MX Component.
0x0180100B	Protocol Type error The protocol specified is incorrect.	<ul style="list-style-type: none"> Exit the program and restart the personal computer. Reinstall MX Component. Please contact your local Mitsubishi Electric sales office or representative.
0x0180100C	Registry search failure	<ul style="list-style-type: none"> Exit the program and restart the personal computer. Reinstall MX Component.
0x0180100D	GetProcAddress failure	
0x0180100E	DLL non-load error	
0x0180100F	Another Object in execution Method cannot be executed because of exclusive control in progress.	<ul style="list-style-type: none"> Execute again after some time
0x01802001	Device error The device character string specified in the method is an unauthorized device character string.	<ul style="list-style-type: none"> Review the device name.
0x01802002	Device number error The device character string number specified in the method is an unauthorized device number.	<ul style="list-style-type: none"> Review the device number.
0x01802003	Program Type error	<ul style="list-style-type: none"> Exit the program and restart the personal computer. Reinstall MX Component. Please contact your local Mitsubishi Electric sales office or representative.
0x01802004	Sumcheck error The sumcheck value of the received data is abnormal.	<ul style="list-style-type: none"> Check the module side sumcheck setting. Check the sumcheck property of the control. Check the cable. Exit the program and restart the personal computer. Reinstall MX Component.
0x01802005	Size error The number of points specified in the method is unauthorized.	<ul style="list-style-type: none"> Check the number of points specified in the method. Review the system, e.g. CPU module, module setting and cable status. Exit the program and restart the personal computer. Reinstall MX Component.
0x01802006	Block number error The block specifying number in the device character string specified in the method is unauthorized.	<ul style="list-style-type: none"> Review the block specifying number in the device character string specified in the method.
0x01802007	Receive data error The data received is abnormal.	<ul style="list-style-type: none"> Review the system, e.g. CPU module, module setting and cable status. Check the cable. Exit the program and restart the personal computer.

Error code	Error description	Corrective action
0x01802008	Write Protect error	<ul style="list-style-type: none"> • Exit the program and restart the personal computer. • Reinstall MX Component. • Please contact your local Mitsubishi Electric sales office or representative.
0x01802009	Reading Parameters error	
0x0180200A	Writing Parameters error	
0x0180200B	Programmable controller type mismatch The CPU type set to the property and the CPU type set on the communication settings utility do not match the CPU type on the other end of communication.	
0x0180200C	Request Cancel error The request was cancelled while being processed.	
0x0180200D	Drive Name error The specified drive name is incorrect.	
0x0180200E	Beginning Step error The beginning step specified is incorrect.	
0x0180200F	Parameter Type error The parameter type is incorrect.	
0x01802010	File Name error The file name is incorrect.	
0x01802011	Status error The status of Registration/Cancellation/Setting is incorrect.	
0x01802012	Detailed Condition Field error	<ul style="list-style-type: none"> • Check the station number. • Check if the method being executed is supported or not. • Check the system configuration such as programmable controller, unit, etc.
0x01802013	Step Condition error	
0x01802014	Bit Device Condition error	
0x01802015	Parameter Settings error	
0x01802016	Error in specifying station number Method does not support the operations corresponding to the specified station number.	
0x01802017	Keyword error	
0x01802018	Read/Write Flag error	
0x01802019	Refresh Method error	
0x0180201A	Buffer Access Method error	
0x0180201B	Start Mode/Stop Mode error	
0x0180201C	Written clock data error Clock data specified for write cannot be written properly since that data is in error.	<ul style="list-style-type: none"> • Review the clock data to be written.
0x0180201D	Online clock data write error Write of clock data failed. Clock data cannot be written since the CPU module is during RUN.	<ul style="list-style-type: none"> • Place the CPU module in the STOP status
0x0180201E	ROM drive error	<ul style="list-style-type: none"> • Exit the program and restart the personal computer. • Reinstall MX Component. • Please contact your local Mitsubishi Electric sales office or representative.
0x0180201F	While Tracing error Invalid operation was carried out during trace.	
0x01802020	First I/O number error The first I/O number specified in the method is an unauthorized value.	
0x01802021	First address error The buffer address specified in the method is an unauthorized value.	
0x01802022	Pattern error	
0x01802023	SFC Block No. error	
0x01802024	SFC Step No. error	
0x01802025	Step No. error	
0x01802026	Data error	
0x01802027	System Data error	
0x01802028	Error in number of TC settings Value	<ul style="list-style-type: none"> • Exit the program and restart the personal computer. • Reinstall MX Component. • Please contact your local Mitsubishi Electric sales office or representative.
0x01802029	Clear Mode error	
0x0180202A	Signal Flow error	

Error code	Error description	Corrective action	
0x0180202B	Version Control error	<ul style="list-style-type: none"> • Exit the program and restart the personal computer. • Reinstall MX Component. • Please contact your local Mitsubishi Electric sales office or representative. 	
0x0180202C	Monitor Not Registered error		
0x0180202D	PI Type error		
0x0180202E	PI No error		
0x0180202F	Error in Number of PIs		
0x01802030	Shift error		
0x01802031	File Type error		
0x01802032	Specified module error		
0x01802033	Error check flag error		
0x01802034	Step RUN operation error		
0x01802035	Step RUN data error		
0x01802036	During Step RUN error		
0x01802037	Write error while running program corresponding to E2ROM		
0x01802038	Clock data read/write error The clock data read/write method was executed for the CPU module which does not have the clock devices.		<ul style="list-style-type: none"> • Do not execute clock data read/write.
0x01802039	Trace not completed error		<ul style="list-style-type: none"> • Exit the program and restart the personal computer. • Reinstall MX Component. • Please contact your local Mitsubishi Electric sales office or representative.
0x0180203A	Registration Clear Flag error		
0x0180203B	Operation error		
0x0180203C	Error in the number of exchanges		
0x0180203D	Error in number of loops specified		
0x0180203E	Retrieve data selection		
0x0180203F	Error in number of SFC cycles		
0x01802040	Motion PLC Error		
0x01802041	Motion PLC Communication error		
0x01802042	Fixed execution time setting error		
0x01802043	Error in number of functions		
0x01802044	System information specification error		
0x01802045	Registration Condition Not Formed error		
0x01802046	Function No. error		
0x01802047	RAM drive error		
0x01802048	ROM drive error at the booting side		
0x01802049	Transfer mode specification error at the booting side		
0x0180204A	Insufficient memory error	<ul style="list-style-type: none"> • Exit the program and restart the personal computer. • Reinstall MX Component. • Please contact your local Mitsubishi Electric sales office or representative. 	
0x0180204B	Back up drive ROM error		
0x0180204C	Block size error		
0x0180204D	Detached during RUN state error		
0x0180204E	Unit Already Registered error		
0x0180204F	Password Registration Data Full error		
0x01802050	Password Not Registered error		
0x01802051	Remote Password error		
0x01802052	IP Address error		
0x01802053	Timeout value out of range error		
0x01802054	Command not detected error		
0x01802055	Trace execution type error		
0x01802056	Version error		
0x01802057	Tracking cable error The tracking cable is faulty. The CPU module status is error.		<ul style="list-style-type: none"> • Reexamine the system such as the CPU module, module setting and cable status.
0x0180205C	Keyword protection error CPU module is protected by the key word.		<ul style="list-style-type: none"> • Disable the keyword and execute again.
0x0180205D	Keyword disable error The inputted keyword is wrong.		<ul style="list-style-type: none"> • Input a correct keyword.
0x0180205E	Keyword protecting error CPU module did not accept the protecting command.		<ul style="list-style-type: none"> • Execute again or re-switch the power of the CPU module.

Error code	Error description	Corrective action
0x0180205F	Keyword entry error An illegal character is included in the inputted keyword.	<ul style="list-style-type: none"> Input a correct keyword.
0x01802060	Keyword deletion error The inputted keyword is wrong.	
0x01802062	Received packet CRC check error An error occurred in CRC check for receive packet data.	<ul style="list-style-type: none"> Execute the communication process again.
0x01802063	Received packet CRC check error An error occurred in CRC check for whole data file of receive packet.	
0x01802064	FX Series programmable controller connection error	<ul style="list-style-type: none"> Execute the communication process again.
0x01802070	Online change program error No target program for online change exists in the CPU module.	<ul style="list-style-type: none"> Execute the online change after turning the CPU module to STOP.
0x01802071	Ether direct communication multiple response receive error Multiple responses were received during Ether direct communication.	<ul style="list-style-type: none"> Check that the personal computer and the CPU module are in a one-to-one connection.
0x01802072	Ether direct communication error Cannot communicate because the CPU module is being accessed by another personal computer during Ether direct communication.	
0x01802073	CPU module search response error The number of responses in the CPU module search exceeded the maximum number to be searched.	<ul style="list-style-type: none"> Reduce the number of CPU modules on the network to 1024 or less.
0x01802074	Redundant system other system connection diagnostics error	<ul style="list-style-type: none"> Disconnect the cable and connect it to the currently disconnected CPU module. Or, change the redundant CPU specification to the self system.
0x01808001	Multiple Open error Open method was executed while it was open.	<ul style="list-style-type: none"> Exit the program and restart the personal computer. Execute any method other than Open.
0x01808002	Channel number specifying error The port number set to the property and the port number set on the communication settings utility are unauthorized values.	<ul style="list-style-type: none"> Set the correct value to the port number of the property. Make communication settings again on the communication settings utility.
0x01808003	Driver not yet started The network board driver is not started.	<ul style="list-style-type: none"> Start the driver.
0x01808004	Error in overlap event generation	<ul style="list-style-type: none"> Exit the program and restart the personal computer. Reinstall MX Component. Please contact your local Mitsubishi Electric sales office or representative.
0x01808005	MUTEX generation error Creation of MUTEX to exercise exclusive control failed.	<ul style="list-style-type: none"> Exit the program and restart the personal computer. Reinstall MX Component.
0x01808006	Error in socket object generation Socket object could not be created	<ul style="list-style-type: none"> Exit the program and restart the personal computer. Reinstall MX Component. Please contact your local Mitsubishi Electric sales office or representative.
0x01808007	Socket object generation error Creation of the Socket object failed.	<ul style="list-style-type: none"> Check for a running application which uses the same port number. Retry after changing the port number value of the property. Retry after changing the port number value on the communication settings utility. Check whether the first character of IP address is not zero. Make Ethernet board and protocol settings on the control panel of the OS. Right-click and select "Run as administrator" when starting the application. Exit the program and restart the personal computer. Please contact your local Mitsubishi Electric sales office or representative.
0x01808008	Port connection error Establishment of connection failed. The other end does not respond.	<ul style="list-style-type: none"> Review the IP address and port number values of the properties. Review the port number value on the communication settings utility. Review the system, e.g. CPU module, module setting and cable status Exit the program and restart the personal computer.
0x01808009	COM port handle error The handle of the COM port cannot be acquired. The COM port object cannot be copied. The SOCKET object cannot be copied.	<ul style="list-style-type: none"> Check for an application which uses the COM port. Exit the program and restart the personal computer.

Error code	Error description	Corrective action
0x0180800A	Buffer size setting error Setting of the COM port buffer size failed.	<ul style="list-style-type: none"> • Check for an application which uses the COM port. • Make COM port setting on the control panel of the OS. • Exit the program and restart the personal computer.
0x0180800B	DCB value acquisition error Acquisition of the COM port DCB value failed.	
0x0180800C	DCB setting error Setting of the COM port DCB value failed.	
0x0180800D	Time-out value setting error Setting of the COM port time-out value failed.	<ul style="list-style-type: none"> • Check the property timeout value. • Review the time-out value on the communication settings utility. • Check for an application which uses the COM port. • Make COM port setting on the control panel of the OS. • Exit the program and restart the personal computer.
0x0180800E	Shared memory open error Open processing of shared memory failed.	<ul style="list-style-type: none"> • Check whether the GX Simulator has started. • Exit the program and restart the personal computer.
0x01808101	Duplex close error	<ul style="list-style-type: none"> • Exit the program and restart the personal computer.
0x01808102	Handle close error Closing of the COM port handle failed.	
0x01808103	Driver close error Closing of the driver handle failed.	
0x01808104	Overlap Event Close error	
0x01808105	Mutex Handle Close error	<ul style="list-style-type: none"> • Exit the program and restart the personal computer. • Reinstall MX Component. • Please contact your local Mitsubishi Electric sales office or representative.
0x01808106	COM Port Handle Close error	
0x01808201	Send error Data send failed.	<ul style="list-style-type: none"> • Review the system, e.g. CPU module, module setting and cable status. • Make COM port setting on the control panel of the OS. • Make Ethernet board and protocol settings on the control panel. • Retry the method. • Exit the program and restart the personal computer.
0x01808202	Send data size error Data send failed.	<ul style="list-style-type: none"> • Exit the program and restart the personal computer.
0x01808203	Queue clear error Clearing of the COM port queue failed.	<ul style="list-style-type: none"> • Exit the program and restart the personal computer. • Close and Open again.
0x01808301	Receive error Data receive failed.	<ul style="list-style-type: none"> • Review the system, e.g. CPU module, module setting and cable status. • Check the property timeout value. • Review the time-out value on the communication settings utility. • Retry the method. • Exit the program and restart the personal computer.
0x01808302	Not Sent error	<ul style="list-style-type: none"> • Exit the program and restart the personal computer. • Reinstall MX Component. • Please contact your local Mitsubishi Electric sales office or representative.
0x01808303	Error in retrieving Overlap Event	<ul style="list-style-type: none"> • Exit the program and restart the personal computer. • Reinstall MX Component. • Please contact your local Mitsubishi Electric sales office or representative.
0x01808304	Receive buffer size shortage Receive data was larger than the receive buffer size prepared for the system.	<ul style="list-style-type: none"> • Exit the program and restart the personal computer.
0x01808401	Control error Changing of the COM port communication control failed.	
0x01808402	Signal Line Control error	<ul style="list-style-type: none"> • Exit the program and restart the personal computer. • Reinstall MX Component. • Please contact your local Mitsubishi Electric sales office or representative.
0x01808403	Signal line specifying error Changing of the COM port communication control failed.	<ul style="list-style-type: none"> • Exit the program and restart the personal computer.
0x01808404	Open not yet executed	<ul style="list-style-type: none"> • Execute Open. • Exit the program and restart the personal computer.
0x01808405	Communication parameter error The data bit and stop bit combination of the properties is unauthorized.	<ul style="list-style-type: none"> • Review the data bit and stop bit values of the properties. • Set it again on the communication settings utility.
0x01808406	Transmission speed value specifying error The transmission speed of the property is unauthorized.	<ul style="list-style-type: none"> • Review the transmission speed value of the property. • Set it again on the communication settings utility.
0x01808407	Data length error The data bit value of the property is unauthorized.	<ul style="list-style-type: none"> • Review the data bit value of the property. • Set it again on the communication settings utility.
0x01808408	Parity specifying error The parity value of the property is unauthorized.	<ul style="list-style-type: none"> • Review the parity value of the property. • Set it again on the communication settings utility.

Error code	Error description	Corrective action
0x01808409	Stop bit specifying error The stop bit value of the property is unauthorized.	<ul style="list-style-type: none"> Review the stop bit value of the property. Set it again on the communication settings utility.
0x0180840A	Communication control setting error The control value of the property is unauthorized.	<ul style="list-style-type: none"> Review the control value of the property. Set it again on the communication settings utility.
0x0180840B	Time-out error Though the time-out period had elapsed, data could not be received.	<ul style="list-style-type: none"> Check the property timeout value. Set it again on the communication settings utility. Review the system, e.g. CPU module, module setting and cable status. Check if communication can be established with the Ping command. Retry the method. Close and Open again. Exit the program and restart the personal computer.
0x0180840C	Connect error	<ul style="list-style-type: none"> Exit the program and restart the personal computer.
0x0180840D	Duplex connect error	
0x0180840E	Attach failure Attaching of the socket object failed.	
0x0180840F	Signal line status acquisition failure Acquisition of the COM port signal line status failed.	
0x01808410	CD signal line OFF The CD signal on the other end of communication is in the OFF status.	<ul style="list-style-type: none"> Review the system, e.g. CPU module, module setting and cable status. Exit the program and restart the personal computer.
0x01808411	Password mismatch error	<ul style="list-style-type: none"> Check the remote password of the property.
0x01808412	TEL Communication error	<ul style="list-style-type: none"> Exit the program and restart the personal computer. Reinstall MX Component. Please contact your local Mitsubishi Electric sales office or representative.
0x01808501	USB driver load error Loading of the USB driver failed.	<ul style="list-style-type: none"> Exit the program and restart the personal computer. Reinstall MX Component. Check USB driver installation.
0x01808502	USB driver connect error Connection of the USB driver failed.	
0x01808503	USB driver send error Data send failed.	<ul style="list-style-type: none"> Review the system, e.g. CPU module, module setting and cable status. Make USB setting on the control panel (device manger) of the OS. Retry the method. Exit the program and restart the personal computer.
0x01808504	USB driver receive error Data receive failed.	
0x01808505	USB Driver Timeout error	
0x01808506	USB driver initialization error Initialization of the USB driver failed.	<ul style="list-style-type: none"> Make USB setting on the control panel (device manger) of the OS. Exit the program and restart the personal computer.
0x01808507	Other USB error Error related to data send/receive occurred.	<ul style="list-style-type: none"> Disconnect the cable once, then reconnect. Exit the program and restart the personal computer. Reinstall MX Component.
0x01809000	GX Works2 uninstallation error The error occurred by retrieving the installation passing of GX Simulator2.	<ul style="list-style-type: none"> Install GX Works2.
0x01809001	GX Simulator2 unstart error GX Simulator2 did not start.	<ul style="list-style-type: none"> Start GX Simulator2.
0x01809002	GX Simulator2 start error	<ul style="list-style-type: none"> Exit the program and restart the personal computer. Reinstall MX Component. Please contact your local Mitsubishi Electric sales office or representative.
0x01809003	GX Simulator2 start time-out error	
0x01809004	GX Simulator2 stop error	
0x01809005	GX Simulator2 start error	
0x01809007	GX Simulator2 stop error	
0x01809008	GX Simulator2 start error Because it had reached upper bounds of the number of simulations that was able to be started at the same time, it was not possible to start.	
0x01809009	GX Simulator2 start error The simulation of only one project that can be started has started.	

Error code	Error description	Corrective action
0x01809010	GX Simulator2 start information illegal error The error occurred because it was not able to secure the memory area to allocate GX Simulator2 start information.	<ul style="list-style-type: none"> Exit the program and restart the personal computer. Reinstall MX Component. Please contact your local Mitsubishi Electric sales office or representative.
0x01809021	GX Simulator2 start error Because it had reached upper bounds of the number of simulations that was able to be started at the same time, it was not possible to start.	
0x01809022	GX Simulator2 start error The simulation of other CPU was not able to begin because the simulation of the project of FXCPU had already been begun.	<ul style="list-style-type: none"> Exit the program and restart the personal computer. Reinstall MX Component. Please contact your local Mitsubishi Electric sales office or representative.
0x02000001	Points Exceeded error The number of points registered in the monitoring server is very high.	<ul style="list-style-type: none"> Reduce the no. of points registered by the monitor. Exit the program and restart the personal computer. Reinstall MX Component. Please contact your local Mitsubishi Electric sales office or representative.
0x02000002	Shared memory creation error Failed in creating shared memory.	<ul style="list-style-type: none"> Exit the program and restart the personal computer. Reinstall MX Component. Please contact your local Mitsubishi Electric sales office or representative.
0x02000003	Shared memory access error	
0x02000004	Memory securing error Failed in securing memory for the monitoring server.	<ul style="list-style-type: none"> Close the other applications. Increase the system memory. Exit the program and restart the personal computer. Reinstall MX Component. Please contact your local Mitsubishi Electric sales office or representative.
0x02000005	Device Not Registered error Monitor has not been registered.	<ul style="list-style-type: none"> Register the monitor in the monitoring server. Exit the program and restart the personal computer. Reinstall MX Component. Please contact your local Mitsubishi Electric sales office or representative.
0x02000006	Monitoring Server Startup error Monitoring Server is not started.	<ul style="list-style-type: none"> Start the Monitoring Server. Exit the program and restart the personal computer. Reinstall MX Component. Please contact your local Mitsubishi Electric sales office or representative.
0x02000010	Yet to retrieve Device Value error Monitoring is not yet completed.	<ul style="list-style-type: none"> Try to retrieve the value again after waiting for a fixed amount of time. Exit the program and restart the personal computer. Reinstall MX Component. Please contact your local Mitsubishi Electric sales office or representative.
0x03000001	Command not Supported. Command is not supported.	<ul style="list-style-type: none"> Exit the program and restart the personal computer. Reinstall MX Component. Please contact your local Mitsubishi Electric sales office or representative.
0x03000002	Memory lock error Failed while locking memory.	
0x03000003	Memory securing error Failed in securing the memory.	
0x03000004	DLL read error Failed in reading DLL.	
0x03000005	Resource securing error Failed in securing the resources.	
0x03010001	File Creation error Failed in creating the file.	
0x03010002	File Open error Failed to open the file.	
0x03010003	Buffer Size error The buffer size specified is either incorrect or not enough.	<ul style="list-style-type: none"> Please contact your local Mitsubishi Electric sales office or representative.
0x03010004	SIL Sentence formation error SIL sentence formation is incorrect.	
0x03010005	File Name error The specified filename is too long.	<ul style="list-style-type: none"> Specify a shorter filename. Exit the program and restart the personal computer. Reinstall MX Component. Please contact your local Mitsubishi Electric sales office or representative.
0x03010006	File does not exist error The specified file does not exist.	<ul style="list-style-type: none"> Check the file name. Check if the file exists or not. Exit the program and restart the personal computer. Reinstall MX Component. Please contact your local Mitsubishi Electric sales office or representative.

Error code	Error description	Corrective action
0x03010007	File Structure error The data structure in the specified file is incorrect.	<ul style="list-style-type: none"> • Exit the program and restart the personal computer. • Reinstall MX Component. • Please contact your local Mitsubishi Electric sales office or representative.
0x03010008	File already exists error The specified file already exists.	<ul style="list-style-type: none"> • Check the file name. • Exit the program and restart the personal computer.
0x03010009	File does not exist error The specified file does not exist.	<ul style="list-style-type: none"> • Reinstall MX Component. • Please contact your local Mitsubishi Electric sales office or representative.
0x0301000A	File Deletion error The specified file could not be deleted.	<ul style="list-style-type: none"> • Exit the program and restart the personal computer. • Reinstall MX Component.
0x0301000B	Multiple Open error The specified project has been opened twice.	<ul style="list-style-type: none"> • Please contact your local Mitsubishi Electric sales office or representative.
0x0301000C	File Name error The specified filename is incorrect.	<ul style="list-style-type: none"> • Check the file name. • Exit the program and restart the personal computer. • Reinstall MX Component. • Please contact your local Mitsubishi Electric sales office or representative.
0x0301000D	File Read error Failed in reading the file.	<ul style="list-style-type: none"> • Exit the program and restart the personal computer. • Reinstall MX Component.
0x0301000E	File Write error Failed in writing the file.	<ul style="list-style-type: none"> • Please contact your local Mitsubishi Electric sales office or representative.
0x0301000F	File Seek error File seek failed.	
0x03010010	File Close error Failed while closing the file.	
0x03010011	Folder Creation error Failed while creating the folder.	
0x03010012	File Copy error Failed while copying the file.	<ul style="list-style-type: none"> • Exit the program and restart the personal computer. • Reinstall MX Component.
0x03010013	Project Path error The length of the project path is incorrect.	<ul style="list-style-type: none"> • Please contact your local Mitsubishi Electric sales office or representative.
0x03010014	Project Type error The project type is incorrect.	
0x03010015	File Type error The file type is incorrect.	
0x03010016	Sub-File Type error The sub-file type is incorrect.	
0x03010017	Insufficient Disk space error The disk space is insufficient.	
0x03020002	Multiple Open error Tried to open DBProduct more than once.	
0x03020003	Not yet open error DBProduct is not opened.	
0x03020004	Extract error DBProduct is not extracted.	

Error code	Error description	Corrective action	
0x03020010	Parameter error The parameters of DBProduct are incorrect.	<ul style="list-style-type: none"> • Exit the program and restart the personal computer. • Reinstall MX Component. • Please contact your local Mitsubishi Electric sales office or representative. 	
0x03020011	Language error The language parameter is incorrect.		
0x03020012	Error in specifying Maker The maker parameter is incorrect.		
0x03020013	Error in specifying Unit The unit parameter is incorrect.		
0x03020014	SQL Parameter error SIL, SQL Parameter of DBProduct is incorrect.		
0x03020015	SIL Sentence formation error SIL sentence formation is incorrect.		
0x03020016	Field Key Input Error The entered field key is incorrect.		
0x03020050	Record Data Construction error. Failed in reconstructing the record data of DBProduct.		
0x03020060	Error Retrieving Record Data Failed while retrieving DBProduct record data.		
0x03020061	Last Record error Cannot retrieve the next record since the current record is the last record.		
0x03FF0000	Initialization error		<ul style="list-style-type: none"> • Exit the program and restart the personal computer. • Reinstall MX Component. • Please contact your local Mitsubishi Electric sales office or representative.
0x03FF0001	Not Initialized error		
0x03FF0002	Multiple Initialization error		
0x03FF0003	Workspace Initialization error		
0x03FF0004	Database Initialization error		
0x03FF0005	Recordset Initialization error		
0x03FF0006	Error Closing Database		
0x03FF0007	Error Closing Recordset		
0x03FF0008	Database Not Opened error Database is not opened.		
0x03FF0009	Recordset Not Opened error Recordset is not opened.		
0x03FF000A	Table Initialization error Failed in initializing TtableInformation table.	<ul style="list-style-type: none"> • Exit the program and restart the personal computer. • Reinstall MX Component. • Please contact your local Mitsubishi Electric sales office or representative. 	
0x03FF000B	Table Initialization error Failed in initializing TfieldInformation table.		
0x03FF000C	Table Initialization error Failed in initializing TrelationInformation table.		
0x03FF000D	Table Initialization error Failed in initializing Tlanguage table.		
0x03FF000E	Table Initialization error Failed in initializing Tmaker table.		
0x03FF000F	Table Initialization error Failed in initializing TOpenDatabase table.		
0x03FF0010	Field Value error		
0x03FF0011	Field Value error		
0x03FF0012	Exit error Failed to exit the database.	<ul style="list-style-type: none"> • Exit the program and restart the personal computer. • Reinstall MX Component. • Please contact your local Mitsubishi Electric sales office or representative. 	
0x03FF0100	Moving Record error Failed while moving the record.		
0x03FF0101	Retrieving Record Count error Failed to retrieve the record count.		
0x03FF0110	Retrieving Field Value error Failed in retrieving the field value.		
0x03FF0111	Setting Field Value error Failed in setting the field value.		
0x03FFFFFF	Other errors		

Error code	Error description	Corrective action
0x04000001	No command error The specified CPU type cannot be used to perform processing.	<ul style="list-style-type: none"> • Check the CPU type set to ActCpuType. • Check that the system configuration is not an unsupported configuration. • Exit the program and restart the personal computer. • Reinstall MX Component.
0x04000002	Memory lock error Failed while locking memory.	<ul style="list-style-type: none"> • Exit the program and restart the personal computer. • Reinstall MX Component.
0x04000003	Memory securing error Failed in securing the memory.	<ul style="list-style-type: none"> • Please contact your local Mitsubishi Electric sales office or representative.
0x04000004	Internal server DLL load error Start of the internal server failed.	<ul style="list-style-type: none"> • Check for the deleted or moved installation file of MX Component. • Exit the program and restart the personal computer. • Reinstall MX Component.
0x04000005	Resource securing error Failed in securing the resources.	<ul style="list-style-type: none"> • Exit the program and restart the personal computer. • Reinstall MX Component.
0x04000006	Error Loading Main Object Failed in reading the file.	<ul style="list-style-type: none"> • Please contact your local Mitsubishi Electric sales office or representative.
0x04000007	Error Loading Conversion Table Failed in reading table data.	
0x04000100	Incorrect Intermediate Code Size error	
0x04010001	Intermediate Code Not Converted error The converted machine code for one command is more than 256 bytes.	
0x04010002	Intermediate Code Completion error Intermediate code area of the code to be converted ended abruptly.	
0x04010003	Insufficient Intermediate Code error The intermediate code of the code to be converted was insufficient.	
0x04010004	Intermediate Code Data error The intermediate code to be converted is incorrect.	

Error code	Error description	Corrective action
0x04010005	Intermediate Code Structure error The number of steps in the intermediate code is incorrect.	<ul style="list-style-type: none"> • Exit the program and restart the personal computer. • Reinstall MX Component. • Please contact your local Mitsubishi Electric sales office or representative.
0x04010006	Error in Number of Steps The number of steps in comment intermediate code is incorrect.	
0x04010007	Insufficient Storage Space for Machine Code error The storage space for machine code is insufficient.	
0x04010008	Other errors (Other errors generated during the conversion of Intermediate code to machine code.)	
0x04011001	Machine Code Not Converted error The converted intermediate code for one command is more than 256 bytes.	
0x04011002	Machine Code Completion error The machine code area to be converted ended abruptly.	
0x04011003	Could not convert since the machine code to be converted was abnormal.	
0x04011004	Insufficient Storage Space for Intermediate Code error The storage area for intermediate code is insufficient.	
0x04011005	Other errors (Other errors generated while converting machine code to Intermediate code.)	
0x04020001	Text Code Not Converted error The converted intermediate code for one command is more than 256 bytes.	
0x04020002	No Input error The input list code is insufficient.	
0x04020003	Command error The command name of list code to be converted is incorrect.	
0x04020004	Device error The device name of list code to be converted is incorrect.	
0x04020005	Device No. error The device number of the list code to be converted is out of range.	
0x04020006	Conversion error The list code to be converted conversion could not be identified.	
0x04020007	Text Data error The list code to be converted is incorrect.	<ul style="list-style-type: none"> • Exit the program and restart the personal computer. • Reinstall MX Component. • Please contact your local Mitsubishi Electric sales office or representative.
0x04020008	Error in SFC Operation Output The output command of SFC operation is incorrect.	
0x04020009	SFC Shift Condition error SFC shift condition command is incorrect.	
0x0402000A	Error in Statements between lines The statements entered between lines are incorrect.	
0x0402000B	P.I Statement error The P.I statement entered is incorrect.	
0x0402000C	Note error The Note entered is incorrect.	
0x0402000D	Comment error The comment entered is incorrect.	
0x0402000E	Other errors (Other errors generated during the conversion of list to Intermediate code)	
0x04021001	Intermediate Code Not Converted error The converted list code for one command has exceeded 256 bytes.	
0x04021002	Intermediate Code Area Full error Intermediate code area to be converted is full.	
0x04021003	Command error The command specified by the intermediate code to be converted is incorrect.	

Error code	Error description	Corrective action
0x04021004	Device error The device specified in the intermediate code to be converted is incorrect.	<ul style="list-style-type: none"> • Exit the program and restart the personal computer. • Reinstall MX Component. • Please contact your local Mitsubishi Electric sales office or representative.
0x04021005	Intermediate Code error The structure of intermediate code to be converted is incorrect.	
0x04021006	Insufficient List Storage Space error The space for storing the converted list code is insufficient.	
0x04021007	Other errors (Other errors generated during the conversion of intermediate code to list)	
0x04030001	Not Converted error The storage space for converted intermediate code is insufficient.	
0x04030002	Bad Circuit Creation error The character memory circuit is not completed in a sequence.	
0x04030003	Specified Circuit Size Exceeded Specified circuit size is too big.	
0x04030004	Incorrect Return Circuit error There is no consistency before and after the return circuit. The setting for the return circuit is too high.	
0x04030005	Other errors (Other errors generated while converting from Character Memory to Intermediate Code)	
0x04031001	Not Converted error The size (vertical/horizontal) of the character memory specified is incorrect.	
0x04031002	Abnormal Command Code error The command intermediate code to be converted is incorrect.	<ul style="list-style-type: none"> • Exit the program and restart the personal computer. • Reinstall MX Component. • Please contact your local Mitsubishi Electric sales office or representative.
0x04031003	Bad Circuit Creation error Could not be converted to Sequence Circuit. There is no END command.	
0x04031004	Specified Circuit Size exceeded error Specified circuit size is too big.	
0x04031005	Fatal error Fatal error has occurred.	
0x04031006	Insufficient number of storage blocks error The space to store the converted character memory circuit blocks is not sufficient.	
0x04031007	Circuit Block Search error Data is broken off in the circuit block.	
0x04031008	Other errors (Other errors generated during the conversion of intermediate code to character memory)	
0x04040001	CAD Data error The CAD data format is incorrect.	
0x04040002	Output Data error The input CAD data type and the output CAD data type are not matching.	
0x04040003	Library Load error Failed to load the library.	
0x04040004	Storage Space Secure error The space secured to store the converted data is not sufficient.	
0x04040005	No END Command error There is no END command in the data to be converted.	
0x04040006	Abnormal Command Code There is abnormal command code in the CAD data to be converted.	
0x04040007	Device No. error The device number is out of range.	

Error code	Error description	Corrective action
0x04040008	Step No. error The step number is out of range.	<ul style="list-style-type: none"> • Exit the program and restart the personal computer. • Reinstall MX Component. • Please contact your local Mitsubishi Electric sales office or representative.
0x04040009	Specified Circuit Size exceeded error 1 circuit block is too big.	
0x0404000A	Return Circuit Error The return circuit is incorrect.	
0x0404000B	Bad Circuit Creation error The circuit data is incorrect.	
0x0404000C	SFC Data error The SFC data to be converted is incorrect.	
0x0404000D	List Data error The list data to be converted is incorrect.	
0x0404000E	Comment Data error The comment data to be converted is incorrect.	
0x0404000F	Statement error The statement data to be converted is incorrect.	
0x04040010	Other errors (Other errors generated during the conversion of CAD code to Intermediate code.)	
0x04041001	Intermediate Code Data error There is no intermediate code to be converted. The format of the intermediate code is incorrect.	
0x04041002	CAD Data Type error The input CAD data type and the output CAD data type are not matching.	<ul style="list-style-type: none"> • Exit the program and restart the personal computer. • Reinstall MX Component. • Please contact your local Mitsubishi Electric sales office or representative.
0x04041003	Library error Failed to load the library.	
0x04041004	Insufficient Input Data error Data to be converted is insufficient.	
0x04041005	Insufficient Storage Space error There is not enough space to store the CAD data to be converted.	
0x04041006	No END Command error There is no END command in the data to be converted.	
0x04041007	Abnormal Command Code error There is abnormal command code in the CAD data to be converted.	
0x04041008	Device No. error The device number is out of range.	
0x04041009	Step No. error The step number is out of range.	
0x0404100A	Specified Circuit Size exceeded error 1 circuit block is too big.	
0x0404100B	Return Circuit Error The return circuit is incorrect.	
0x0404100C	Bad Circuit Creation error The circuit data is incorrect.	
0x0404100D	SFC Data error The SFC data to be converted is incorrect.	
0x0404100E	List Data error The list data to be converted is incorrect.	
0x0404100F	Comment Data error The comment data to be converted is incorrect.	

Error code	Error description	Corrective action	
0x04041010	Statement error The statement data to be converted is incorrect.	<ul style="list-style-type: none"> • Exit the program and restart the personal computer. • Reinstall MX Component. • Please contact your local Mitsubishi Electric sales office or representative. 	
0x04041011	Other errors (Other errors generated during the conversion of Intermediate code to CAD code.)		
0x04050001	Abnormal Character String Specified error Device character string specified is incorrect.		
0x04050002	Device Points error Device points are out of range.		
0x04050003	Other errors (The errors generated during the conversion of the Device Character String to Device Intermediate Code)		
0x04051001	Device Name error The classification specified for the device intermediate code is incorrect.		
0x04051002	Device Name error The classification specified for the extended specification device intermediate code is incorrect.		
0x04051003	Other errors (The errors generated during the conversion of the Device Intermediate Code to Device Character String)		
0x04052001	Abnormal Character String Specified error Device character string specified is incorrect.		
0x04052002	Device Points error Device points are out of range.		
0x04052003	Other errors (The errors generated during the conversion of the Device Character String to Device Representation Code)		
0x04053001	Device Representation error The classification specified for the device intermediate code is incorrect.		
0x04053002	Device Representation error The classification specified for the extended specification device intermediate code is incorrect.		<ul style="list-style-type: none"> • Exit the program and restart the personal computer. • Reinstall MX Component. • Please contact your local Mitsubishi Electric sales office or representative.
0x04053003	Device Representation error The rectification part specified for the device is incorrect.		
0x04053004	Device Representation error The rectification part specified for the extended device is incorrect.		
0x04053005	Other errors (The errors generated during the conversion of the Device Representation Code to Device Character String)		
0x04064001	Abnormal Device Intermediate Code error The intermediate code for the device is incorrect.		
0x04064002	Other errors (Other errors generated during the conversion of the Intermediate code for the Device to Device Name)		
0x04065001	Abnormal Device Name error The classification specified for the device intermediate code is incorrect.		
0x04065002	Abnormal Device Name error The classification for the intermediate code of the extended specification device is incorrect.		
0x04065003	Other errors (Other errors generated during the conversion of the device name to Intermediate code)		
0x04066001	Device Intermediate Code error The intermediate code for the device is incorrect.		
0x04066002	Other errors (Other errors generated during the conversion of the device intermediate code to device representation code.)		

Error code	Error description	Corrective action
0x04067001	Device Representation error The classification specified for the device intermediate code is incorrect.	<ul style="list-style-type: none"> • Exit the program and restart the personal computer. • Reinstall MX Component. • Please contact your local Mitsubishi Electric sales office or representative.
0x04067002	Device Representation error The classification specified for the extended specification device intermediate code is incorrect.	
0x04067003	Device Representation error The rectification part specified for the device is incorrect.	
0x04067004	Device Representation error The rectification part specified for the extended device is incorrect.	
0x04067005	Other errors (Other errors generated during the conversion of device representation code to the device intermediate code)	
0x04070001	Common Data Conversion error The input data of the device comment conversion is incorrect.	
0x04070002	Insufficient Common Data The data to be converted is insufficient.	
0x04070003	Insufficient Storage Area The area where the conversion data is stored is insufficient.	
0x04071001	Error in CPU Data Conversion The input data of the device comment conversion is incorrect.	
0x04071002	Insufficient CPU Data error The data to be converted is insufficient.	
0x04071003	Insufficient Storage Area The area where the conversion data is stored is insufficient.	<ul style="list-style-type: none"> • Exit the program and restart the personal computer. • Reinstall MX Component. • Please contact your local Mitsubishi Electric sales office or representative.
0x04072001	Open error Failed in creating conversion object.	
0x04072002	CPU Type error The specified CPU type does not exist.	
0x04072003	Not Converted error Converted object does not exist.	
0x04072004	Input Data error The input data is incorrect.	
0x04073001	Program Common Data Conversion error	
0x04073002	Program Common Data Conversion error	
0x04073101	Program CPU Data Conversion error	
0x04074001	Common Data Parameter error	
0x04074002	Network Parameter Common Data error The parameter block exists, but the data inside is not set.	
0x04074101	Parameter CPU Data error	
0x04074102	Network Parameter CPU Data error The parameter block exists, but the data inside is not set.	
0x04074103	Offset error	

Error code	Error description	Corrective action
0x04074201	Error in Specifying Network Type The CPU specified does not support the network type.	<ul style="list-style-type: none"> • Exit the program and restart the personal computer. • Reinstall MX Component. • Please contact your local Mitsubishi Electric sales office or representative.
0x04074202	Parameter Block Number error The Block corresponding to the parameter block number specified does not exist.	
0x04074203	Parameter Block Content error It is different from the content supported by the specified.	
0x04074204	Parameter Block Information error The specified block number does not exist.	
0x04074205	Default Parameter Block is Abnormal The specified block number does not exist.	
0x04074301	Error in Conversion of the Common Parameter Block	
0x04074302	Error in Common Parameter Block No. 1001 The value of the RUN-PAUSE settings existence flag is incorrect.	
0x04074303	Error in Common Parameter Block No. 1003	
0x04074304	Error in Common Parameter Block No. 1008	
0x04074305	Error in Common Parameter Block No. 1100	
0x04074306	Error in Common Parameter Block No. 2001 The device intermediate code specified does not exist.	
0x04074307	Error in Common Parameter Block No. 3000	
0x04074308	Error in Common Parameter Block No. 3002	
0x04074309	Error in Common Parameter Block No. 3004 The settings for the annunciator display mode is incorrect.	
0x0407430A	Error in Common Parameter Block No. 4000 I/O Allotment Data is not created.	
0x0407430B	Error in Common Parameter Block No. 5000 The specified network is not supported.	
0x0407430C	Error in Common Parameter Block No. 5001 Valid unit No is not set while accessing other exchange.	
0x0407430D	Error in Common Parameter Block No. 5002	
0x0407430E	Error in Common Parameter Block No. 5003	
0x0407430F	Error in Common Parameter Block No. 5NM0	
0x04074310	Error in Common Parameter Block No. 5NM1	<ul style="list-style-type: none"> • Exit the program and restart the personal computer. • Reinstall MX Component. • Please contact your local Mitsubishi Electric sales office or representative.
0x04074311	Error in Common Parameter Block No. 5NM2	
0x04074312	Error in Common Parameter Block No. 5NM3	
0x04074313	Error in Common Parameter Block No. 6000	
0x04074314	Error in Common Parameter Block No. FF18 Link parameter Capacity is not set.	
0x04074315	Error in Common Parameter Block No. FF25 Calculation circuit check is not set.	
0x04074316	Error in Common Parameter Block No. FF30 Sampling Trace Data is not created.	
0x04074317	Error in Common Parameter Block No. FF31 Status latch data is not created.	
0x04074318	Error in Common Parameter Block No. F42 Timer processing points are not set.	
0x04074319	Error in Common Parameter Block No. FF30 Setting value device for specified extended timer does not exist.	
0x0407431A	Error in Common Parameter Block No. FF44	
0x0407431B	Error in Common Parameter Block No. FF45	
0x0407431C	Error in Common Parameter Block No. FF60 Terminal Settings are not set.	
0x0407431D	Error in Common Parameter Block No. FF70 User Release area is not set.	

Error code	Error description	Corrective action	
0x04074401	Error in Conversion of CPU Parameter Block	<ul style="list-style-type: none"> • Exit the program and restart the personal computer. • Reinstall MX Component. • Please contact your local Mitsubishi Electric sales office or representative. 	
0x04074402	Error in CPU Parameter Block No.1001		
0x04074403	Error in CPU Parameter Block No.1003		
0x04074404	Error in CPU Parameter Block No.1008		
0x04074405	Error in CPU Parameter Block No.1100		
0x04074406	Error in CPU Parameter Block No. 2001		
0x04074407	Error in CPU Parameter Block No. 3000		
0x04074408	Error in CPU Parameter Block No. 3002		
0x04074409	Error in CPU Parameter Block No. 3004		
0x0407440A	Error in CPU Parameter Block No. 4000		
0x0407440B	Error in CPU Parameter Block No. 5000 The specified network type is not supported.		
0x0407440C	Error in CPU Parameter Block No. 5001		
0x0407440D	Error in CPU Parameter Block No. 5002		
0x0407440E	Error in CPU Parameter Block No. 5003		
0x0407440F	Error in CPU Parameter Block No. 5NM0 The specified network type is not supported.		
0x04074410	Error in CPU Parameter Block No. 5NM1		<ul style="list-style-type: none"> • Exit the program and restart the personal computer. • Reinstall MX Component. • Please contact your local Mitsubishi Electric sales office or representative.
0x04074411	Error in CPU Parameter Block No. 5NM2 The specified network type is not supported.		
0x04074412	Error in CPU Parameter Block No. 5NM3		
0x04074413	Error in CPU Parameter Block No. 6000		
0x04074414	Error in CPU Parameter Block No. FF18		
0x04074415	Error in CPU Parameter Block No. FF25		
0x04074416	Error in CPU Parameter Block No. FF30		
0x04074417	Error in CPU Parameter Block No. FF31		
0x04074418	Error in CPU Parameter Block No. FF42		
0x04074419	Error in CPU Parameter Block No. FF43		
0x0407441A	Error in CPU Parameter Block No. FF44		
0x0407441B	Error in CPU Parameter Block No. FF45		
0x0407441C	Error in CPU Parameter Block No. FF60		
0x0407441D	Error in CPU Parameter Block No. FF70		
0x04075001	Common Data Conversion error Failed while converting the device memory settings portion.	<ul style="list-style-type: none"> • Exit the program and restart the personal computer. • Reinstall MX Component. • Please contact your local Mitsubishi Electric sales office or representative. 	
0x04075002	Common Data Conversion error Failed while converting the device memory settings portion.		
0x04075003	Common Data Conversion error Device memory data portion did not exist.		
0x04075101	Error in CPU Data Conversion Failed while converting the device memory settings portion.		
0x04075102	Error in CPU Data Conversion Failed while converting the device memory settings portion.		
0x04076001	Common Data Conversion error Failed while converting the settings portion of the device comments.		
0x04076002	Common Data Conversion error Failed while converting the data portion of the device comments.		
0x04076101	Error in CPU Data Conversion Failed while converting the settings portion of the device comments.		
0x04076102	Error in CPU Data Conversion Failed while converting the data portion of the device comments.		

Error code	Error description	Corrective action
0x04077001	Common Data Conversion error Failed during the conversion of sampling trace settings portion.	<ul style="list-style-type: none"> • Exit the program and restart the personal computer. • Reinstall MX Component. • Please contact your local Mitsubishi Electric sales office or representative.
0x04077002	Common Data Conversion error Failed during the conversion of sampling trace data portion.	
0x04077101	Error in CPU Data Conversion Failed during the conversion of sampling trace settings portion.	
0x04077102	Error in CPU Data Conversion Failed during the conversion of sampling trace data portion.	
0x04078001	Common Data Conversion error Failed in the conversion of the status latch settings portion.	
0x04078002	Common Data Conversion error Failed in the conversion of the status latch data portion.	
0x04078101	Error in CPU Data Conversion Failed in the conversion of the status latch settings portion.	
0x04078102	Error in CPU Data Conversion Failed in the conversion of the status latch data portion.	
0x04079101	PLC Data Conversion Error Failed in the conversion of the Error History.	
0x0407A101	PLC Data Conversion Error Failed in the conversion of the File List.	
0x0407B101	PLC Data Conversion Error Failed in the conversion of the Error Information.	
0x0407C001	Error in Conversion of Indirect Address to Device Name The device name storage area is not secured.	
0x0407C002	Error in Conversion of Device Name to Indirect Address Indirect Address storage area is not secured.	
0x0407C003	Error in Conversion of Indirect Address to Device Representation The device representation storage area is not secured.	
0x0407C004	Error in Conversion of Device Representation to Indirect Address Indirect Address storage area is not secured.	
0x0407C005	Error in Conversion of Indirect Address to Device Character String Device Character String storage area is not secured.	
0x0407C006	Error in Conversion of Device Character String to Indirect Address Indirect Address storage area is not secured.	<ul style="list-style-type: none"> • Exit the program and restart the personal computer. • Reinstall MX Component. • Please contact your local Mitsubishi Electric sales office or representative.
0x0407C007	Error in Conversion of Intermediate Code to Device Name The device name storage area is not secured.	
0x0407C008	Error in Conversion of Device Name to Intermediate Code Intermediate Code storage area is not secured.	
0x0407C009	Error in Conversion of Intermediate Code to Device representation Device Representation storage area is not secured.	
0x0407C00A	Error in Conversion of Device Representation to Intermediate Code Intermediate Code storage area is not secured.	
0x0407C00B	Error in Conversion of Intermediate Code to Indirect Address Indirect Address storage area is not secured.	
0x0407C00C	Error in Conversion of Indirect Address to Intermediate Code Intermediate Code storage area is not secured.	
0x0407C00D	CPU Type error The specified CPU type is not supported.	

Error code	Error description	Corrective action	
0x0407C00E	Device Character String error The specified device is not supported.	<ul style="list-style-type: none"> • Exit the program and restart the personal computer. • Reinstall MX Component. • Please contact your local Mitsubishi Electric sales office or representative. 	
0x0407C00F	Device Character String error The specified device character string, type is incorrect.		
0x0407C010	Device error The specified device is not supported by the specified CPU.		
0x0407C011	CPU Type error The specified CPU is not supported.		
0x0407C012	Device out of Range error		
0x0407D001	Common Data Conversion error Error in Conversion of SFC trace condition settings portion.		
0x0407D002	Common Data Conversion error Error in Conversion of SFC trace condition data portion.		
0x0407D101	Error in CPU Data Conversion Error in Conversion of SFC trace condition settings portion.		
0x0407D102	Error in CPU Data Conversion Error in Conversion of SFC trace condition data portion.		
0x04080001	Intermediate Code classification out of range error The intermediate code classification specified is out of range.		<ul style="list-style-type: none"> • Exit the program and restart the personal computer. • Reinstall MX Component. • Please contact your local Mitsubishi Electric sales office or representative.
0x04080002	Extended specification Intermediate Code classification out of range error The extended specification intermediate code specified is out of range.		
0x04080003	Device Points check absent error The device does not check the device points.		
0x04090001	GPP Project error The specified CPU type and GPP project type are not matching.		
0x04090002	File Type error The specified GPP project type and file type are not matching.		
0x04090010	Insufficient GPP Data to be converted There is no data to be converted. The data size specified is incorrect.		
0x04090011	Insufficient Storage Space for Converted Data The space for storing converted data is insufficient.		
0x04090012	Error in GPP Data to be converted The GPP data to be converted is incorrect.		
0x04090110	Insufficient Data to be converted error There is no data to be converted. The data size specified is insufficient.	<ul style="list-style-type: none"> • Exit the program and restart the personal computer. • Reinstall MX Component. • Please contact your local Mitsubishi Electric sales office or representative. 	
0x04090111	Insufficient Storage Space for Converted Data error. The storage space for converted data is insufficient.		
0x04090112	Error in data to be converted The data to be converted is incorrect.		
0x040A0001	Insufficient Intermediate Code Storage Space The space secured to store the converted data is not sufficient.		
0x040A0002	The space to store addition SFC information is not sufficient.		
0x040A0003	Conversion error		
0x040A0004	Non-SFC Program error		
0x040A1001	Step Not Used / No Output error	<ul style="list-style-type: none"> • Exit the program and restart the personal computer. • Reinstall MX Component. • Please contact your local Mitsubishi Electric sales office or representative. 	
0x040A1002	Step No. out of range error		
0x040A1003	Step Not Used / No Output error		
0x040A1004	Transition No. out of range		
0x040A1005	Maximum Number Exceeded error		
0x040A1006	Microcontroller Program space error		
0x040A1007	Non-SFC Program error		

Error code	Error description	Corrective action
0x040B0001	Insufficient Intermediate Code Storage Space The space secured to store the converted data is not sufficient.	<ul style="list-style-type: none"> • Exit the program and restart the personal computer. • Reinstall MX Component. • Please contact your local Mitsubishi Electric sales office or representative.
0x040B0002	Conversion error	
0x040B1001	Failed in creating Step Start position table	
0x040B1002	Error Reading Step Information	
0x040B1003	Step No. Error	
0x040B1004	Failed in reading the output of operation/Transition condition intermediate code error	
0x040B1005	Securing Internal Work Area Failed error	
0x040B1006	Error in setting the maximum value of X direction for character memory	
0x040B1007	Insufficient Internal Work Area error	
0x040B1008	Stack Overflow, Abnormal Character Memory	
0x040B1009	Insufficient number of storage blocks error	
0x040B100A	Non-SFC Program error	
0x04FFFFFF	Other errors	
0x10000001	No command error	
0x10000002	Start of communication DLL of MX Component failed.	<ul style="list-style-type: none"> • Exit the program and restart the personal computer. • Reinstall MX Component.
0x10000003	Open failed. (DiskDrive)	
0x10000004	Duplex open error	<ul style="list-style-type: none"> • Exit the program and restart the personal computer.
0x10000005	File Access error	<ul style="list-style-type: none"> • Exit the program and restart the personal computer. • Reinstall MX Component. • Please contact your local Mitsubishi Electric sales office or representative.
0x10000006	Incorrect Folder Name error	
0x10000007	File Access Denied error	
0x10000008	Disk Full error	
0x10000009	File Deletion error	
0x1000000A	Incorrect File Name error	
0x1000000C	Execution failed since another application or thread is making a request.	
0x1000000D	Folder Creation error	<ul style="list-style-type: none"> • Exit the program and restart the personal computer. • Reinstall MX Component. • Please contact your local Mitsubishi Electric sales office or representative.
0x1000000E	Folder/ File Type error	
0x1000000F	Offset Address error	
0x10000010	Request Cancel Cancel Process has occurred.	<ul style="list-style-type: none"> • Exit the program and restart the personal computer. • Reinstall MX Component. • Please contact your local Mitsubishi Electric sales office or representative.
0x10000011	Memory securing error	
0x10000012	Open not yet executed	
0x10000013	Attach Not Executed error	
0x10000014	Object Invalid error	
0x10000015	Request Cancel Failed error	
0x10000016	Failed in Reading Status error	
0x10000017	The specified size (number of devices) is unauthorized.	
0x10000018	There is no registered device.	
0x10000019	Data set Not Executed	
0x1000001A	Read Not Executed error	<ul style="list-style-type: none"> • Exit the program and restart the personal computer. • Reinstall MX Component. • Please contact your local Mitsubishi Electric sales office or representative.
0x1000001B	Incorrect Create Flag error	
0x1000001C	Operation Over Access	
0x1000001D	Redundant Device error	
0x1000001E	Registry search failed.	<ul style="list-style-type: none"> • Exit the program and restart the personal computer. • Exit other programs and secure free memory area. • Reinstall MX Component.

Error code	Error description	Corrective action	
0x1000001F	File Type error	<ul style="list-style-type: none"> Exit the program and restart the personal computer. Reinstall MX Component. Please contact your local Mitsubishi Electric sales office or representative. 	
0x10000020	Device Memory Type error		
0x10000021	Program Range error		
0x10000022	TEL Type error		
0x10000023	TEL Access error		
0x10000024	Cancel Flag Type error	<ul style="list-style-type: none"> Exit the program and restart the personal computer. Reinstall MX Component. Please contact your local Mitsubishi Electric sales office or representative. 	
0x10000030	Multiple Device Registration error		
0x10000031	Device Not Registered error		
0x10000032	Specified device error		
0x10000033	Specified device range error		
0x10000034	File Write error	<ul style="list-style-type: none"> Exit the program and restart the personal computer. Reinstall MX Component. Please contact your local Mitsubishi Electric sales office or representative. 	
0x10000040	Server start failed.		
0x10000041	Server Stop error Failed while stopping the server.		
0x10000042	Server Started Twice error		
0x10000043	Server Not Started error		
0x10000044	Resource Timeout error		
0x10000045	Server Type error		
0x10000046	Failed to Access Server error		
0x10000047	Server Already Accessed error		
0x10000048	Failed in Simulator Startup		
0x10000049	Failed in exiting Simulator		
0x1000004A	Simulator Not Started error		
0x1000004B	Simulator Type error		
0x1000004C	Simulator Not Supported error		
0x1000004D	Simulator Started Twice error		
0x1000004E	Shared Memory Not Started error		
0x10000055	GX Simulator3 did not start error		<ul style="list-style-type: none"> Start GX Simulator3.
0x10000059	Time-out error		<ul style="list-style-type: none"> Check the property timeout value. Check the settings in the communication settings utility. Check the CPU module, unit settings, state of the cable, etc. Close and Open again. Exit the program and restart the personal computer.
0x1000005A	System number error		<ul style="list-style-type: none"> Check the system number set on the communication settings utility.
0x1000007E	Failed to Access MELSOFT Mirror error		<ul style="list-style-type: none"> Close and Open again. Check the personal computer on which MELSOFT Mirror is running, state of the Network.
0x8001000C	Data out of the allowable setting range is specified.	<ul style="list-style-type: none"> Review the value of the property. Set it again on the communication settings utility. 	
0x80010016			
0x80010101	Time-out error	<ul style="list-style-type: none"> Check the property timeout value. Check the settings in the communication settings utility. Check the unit settings, state of the cable, etc. Close and Open again. Exit the program and restart the personal computer. 	
0x80020001	Type of data is invalid.	<ul style="list-style-type: none"> Review the value of the property. Set it again on the communication settings utility. 	
0x80020002	Range of the data is incorrect.		
0x80030001	The specification of the communication port is incorrect.	<ul style="list-style-type: none"> Review the communication port value of the property. Set it again on the communication settings utility. 	
0x80030002	The specification of the Transmission speed is incorrect.	<ul style="list-style-type: none"> Review the transmission speed value of the property. Set it again on the communication settings utility. 	
0x80030003	The specification of the Data bit length is incorrect.	<ul style="list-style-type: none"> Review the data bit value of the property. Set it again on the communication settings utility. 	
0x80030004	The specification of the parity is incorrect.	<ul style="list-style-type: none"> Review the parity value of the property. Set it again on the communication settings utility. 	
0x80030005	The specification of the stop bit length is incorrect.	<ul style="list-style-type: none"> Review the stop bit value of the property. Set it again on the communication settings utility. 	

Error code	Error description	Corrective action
0x80030006	The specification of the wait time is incorrect.	<ul style="list-style-type: none"> Review the wait time of the property. Set it again on the communication settings utility.
0x80030007	The specification of the CR/LF is incorrect.	<ul style="list-style-type: none"> Review the CR/LF of the property. Set it again on the communication settings utility.
0x80030008	The specification of the timeout is incorrect.	<ul style="list-style-type: none"> Check the property timeout value. Set it again on the communication settings utility.
0x80030009	The specification of the station number is incorrect.	<ul style="list-style-type: none"> Review the station number of the property. Set it again on the communication settings utility.
0x8003000A	USB communication setting is invalid.	<ul style="list-style-type: none"> Review the value of the property. Set it again on the communication settings utility.
0x8003000B	USB model code is invalid.	
0x8003000C	Duplication was detected in station number of the USB communication.	<ul style="list-style-type: none"> Review the station number of the property. Set it again on the communication settings utility.
0x80200107	Communication error	<ul style="list-style-type: none"> Try the same method again. Exit the program and restart the personal computer.
0x80200203	Memory securing error	<ul style="list-style-type: none"> Exit the program and restart the personal computer. Exit other programs and secure free memory area.
0x80201001	Specified CPU error	<ul style="list-style-type: none"> Check the CPU type set to ActCpuType. Check that the system configuration is not an unsupported configuration.
0x80201101	Already open error The Open method was executed in the open status.	<ul style="list-style-type: none"> When changing the communication target CPU, execute the Open method after performing Close.
0x80201104	DLL load error	<ul style="list-style-type: none"> Exit the program and restart the personal computer. Exit other programs and secure free memory area. Reinstall MX Component.
0x80201106	Error in Communication object generation	<ul style="list-style-type: none"> Exit the program and restart the personal computer. Reinstall MX Component.
0x80201201	Not yet open error The Open method is not yet executed.	<ul style="list-style-type: none"> After executing the Open method, execute the corresponding method.
0x80201203	Memory securing error	<ul style="list-style-type: none"> Exit the program and restart the personal computer. Exit other programs and secure free memory area.
0x80204203	Memory securing error	
0x80205203	Memory securing error	
0x80205001	Specified CPU error	<ul style="list-style-type: none"> Check the CPU type set to ActCpuType. Check that the system configuration is not an unsupported configuration.
0x80209501	MT Simulator2 start error	<ul style="list-style-type: none"> Exit the program and restart the personal computer.
0x80209502	MT Simulator2 start error	<ul style="list-style-type: none"> Exit the program and restart the personal computer.
0x80209503	MT Simulator2 communication error	<ul style="list-style-type: none"> Exit the program and restart the personal computer.
0x80209504	MT Simulator2 start error Because it had reached upper bounds of the number of simulations that was able to be started at the same time, it was not possible to start.	<ul style="list-style-type: none"> Retry after exiting MT Simulator2.
0x80209505	MT Simulator2 stop error	<ul style="list-style-type: none"> Exit the program and restart the personal computer.
0x80209506	MT Simulator2 stop error	<ul style="list-style-type: none"> Exit the program and restart the personal computer.
0x8020950F	MT Simulator2 stop error MT Simulator2 is not started.	<ul style="list-style-type: none"> Retry after starting MT Simulator2.
0x80209510	MT Simulator2 start error MT Simulator2 is not started.	<ul style="list-style-type: none"> Retry after starting MT Simulator2.
0x80209516	MT Simulator2 start error MT Works2 uninstallation error.	<ul style="list-style-type: none"> Install MT Works2.
0x80209518	MT Simulator2 start error The specified number is already used.	<ul style="list-style-type: none"> Execute after exiting the specified No. of MT Simulator2.
0x80209519	Send data illegal error	<ul style="list-style-type: none"> Review the argument.
0x8020951C	MT Simulator2 start error MT Simulator2 is ending.	<ul style="list-style-type: none"> Retry after exiting MT Simulator2.
0x8020951D	MT Simulator2 start error Already executed Open.	<ul style="list-style-type: none"> Retry after performing Close.
0x8020951E	MT Simulator2, Not yet open error	<ul style="list-style-type: none"> Please contact your local Mitsubishi Electric sales office or representative.
0x8020951F	MT Simulator2 start error Already executed Open.	<ul style="list-style-type: none"> Retry after performing Close.
0x80206004	Remote request error	<ul style="list-style-type: none"> Change the switch on the module to RUN, exit the test mode, and try again.

Error code	Error description	Corrective action
0x8020A104	DLL load error	<ul style="list-style-type: none"> Exit the program and restart the personal computer. Exit other programs and secure free memory area. Reinstall MX Component.
0x8020A203	Memory securing error	<ul style="list-style-type: none"> Exit the program and restart the personal computer. Exit other programs and secure free memory area.
0x80A00101	Not yet open error The Open method is not yet executed.	<ul style="list-style-type: none"> After executing the Open method, execute the corresponding method.
0x80A00104	Already open error The Open method was executed in the open status.	<ul style="list-style-type: none"> When changing the communication target CPU, execute the Open method after performing Close.
0x80A00105	Incorrect data type of the argument	<ul style="list-style-type: none"> Check the data type being used for the argument.
0x80A00106	Data range of the argument is invalid	<ul style="list-style-type: none"> Check the value being used for the argument.
0x80A00107	No command error	<ul style="list-style-type: none"> Not supported by the method.
0x80A00109	Data source cannot be opened.	<ul style="list-style-type: none"> Check the cable connection.
0x80A0010C	The response format from the robot controller was incorrect.	<ul style="list-style-type: none"> Take measures against noise
0xF0000001	No-license error The license is not given to the personal computer.	<ul style="list-style-type: none"> Using the license FD, give the license to the personal computer.
0xF0000002	Set data read error Reading of the set data of the logical station number failed.	<ul style="list-style-type: none"> Specify the correct logical station number. Set the logical station number on the communication settings utility.
0xF0000003	Already open error The Open method was executed in the open status.	<ul style="list-style-type: none"> When changing the communication target CPU, execute the Open method after performing Close.
0xF0000004	Not yet open error The Open method is not yet executed.	<ul style="list-style-type: none"> After executing the Open method, execute the corresponding method.
0xF0000005	Initialization error Initialization of the object possessed internally in MX Component failed.	<ul style="list-style-type: none"> Exit the program and restart the personal computer. Reinstall MX Component.
0xF0000006	Memory securing error Securing of MX Component internal memory failed.	<ul style="list-style-type: none"> Exit the program and restart the personal computer. Exit other programs and secure free memory area.
0xF0000007	Function non-support error The method does not support.	<ul style="list-style-type: none"> Can not use because the corresponding method is not supported.
0xF1000001	Character code conversion error Character code conversion (UNICODE to ASCII code or ASCII code to UNICODE) failed.	<ul style="list-style-type: none"> Check the character string specified in the method. The ASCII character string acquired from the CPU module is abnormal. Review the system, e.g. CPU module, module setting and cable status. Exit the program and restart the personal computer. Retry the GetCpuType method.
0xF1000002	First I/O number error The first I/O number specified is an unauthorized value. A matching first I/O number does not exist.	<ul style="list-style-type: none"> Check the value of the first I/O number specified in the method. Using the GPP function, check the CPU module parameters (I/O assignment).
0xF1000003	Buffer address error The buffer address specified is an unauthorized value. The buffer address is outside the range.	<ul style="list-style-type: none"> Check the value of the buffer address specified in the method.
0xF1000004	Buffer read size error As a result of buffer read, the specified size could not be acquired.	<ul style="list-style-type: none"> Perform reopen processing. Review the system, e.g. CPU module, module setting and cable status. Retry. Exit the program.
0xF1000005	Size error The size specified in the read/write method is abnormal. The read/write first number plus size exceeds the device or buffer area.	<ul style="list-style-type: none"> Check the size specified in the method.
0xF1000006	Operation error The operation specified for remote operation is an abnormal value.	<ul style="list-style-type: none"> Check the operation specifying value specified in the method.
0xF1000007	Clock data error The clock data is abnormal.	<ul style="list-style-type: none"> Check the clock data specified in the method. Set the correct clock data to the clock data of the CPU module.
0xF1000008	Monitored device registration count excess The number of device points registered in the EntryDeviceStatus method was 0 or less. The number of device points registered in the EntryDeviceStatus method was more than 20.	<ul style="list-style-type: none"> Register the device points between 1 and 20 in the EntryDeviceStatus method.
0xF1000009	Monitored device data registration error	<ul style="list-style-type: none"> After making deregistration in the FreeDeviceStatus method, execute the EntryDeviceStatus method again.

Error code	Error description	Corrective action
0xF100010	Device status monitor processing failed to start. Device status monitor processing failed to end.	<ul style="list-style-type: none"> Start/end the device status monitor processing again in the EntryDeviceStatus method.
0xF100011	The VARIANT argument data type is wrong.	<ul style="list-style-type: none"> Reexamine the data type specified for the VARIANT argument. Check whether the array variable size is large enough. Check whether the data type specified in the corresponding method has been set.
0xF100012	The device status monitoring time interval is a value outside the range 1 second to 1 hour (1 to 3600).	<ul style="list-style-type: none"> Specify the device status monitoring time between 1 and 3600.
0xF100013	Already Connected error. Connect was executed again after it was executed for the same object.	<ul style="list-style-type: none"> Execute the Connect method after executing the Disconnect method.
0xF100014	Invalid Telephone Number error. Characters other than "0123456789-*#" that are allowed for telephone numbers are included.	<ul style="list-style-type: none"> Rectify the Telephone number and try to Connect again.
0xF100015	Exclusive Control Failure error. There was failure in the exclusive control process while executing the Connect and Disconnect method.	<ul style="list-style-type: none"> In case if Connect/Disconnect method is being executed for any other object, execute the failed method (Connect/ Disconnect) again after the completion of the Connect/ Disconnect method of that object. If the Connect/Disconnect process is in progress only for the self object, perform the following. <ul style="list-style-type: none"> Exit the program. Restart the personal computer. Reinstall MX Component.
0xF100016	While connecting to the telephone line error. The telephone line is connected to some other application, other than the one using MX Component.	<ul style="list-style-type: none"> Try Connecting again after disconnecting the application that is using the telephone line.
0xF100017	Telephone line not connected error. Telephone line is not connected. Connect was executed and the telephone line was connected, but it got disconnected due to some reason.	<ul style="list-style-type: none"> (When Connect method has failed) Execute Connect again after executing Disconnect method. (When method other than Connect has failed) Execute Disconnect method, Execute Connect and connect to the telephone line. After connecting, execute the method that failed once again.
0xF100018	No Telephone number error. The telephone No. is not set. The telephone No. or call back No. is not set, if the connection method is Automatic (when specifying the call back No.), call back connection (when specifying the number), or call back Request(when specifying the number).	<ul style="list-style-type: none"> In case of program settings type, set the telephone No. to the property ActDialNumber. (Set the telephone No. to the properties ActDialNumber and ActCallbackNumber, if the connection method is automatic (when specifying the call back No.), call back connection (when specifying the telephone No.), or call back request (when specifying the number).) In case of utility settings type, set the telephone No. using the wizard. (Set the telephone No. and call back No. , if the connection method is automatic (when specifying the call back No.), call back connection (when specifying the telephone No.), or call back request (when specifying the number).)
0xF100019	Not Closed error. Disconnect was executed while in Open state.	<ul style="list-style-type: none"> Try Disconnect again after executing Close.
0xF10001A	Target telephone line connection mismatch error. Connect was tried for a different telephone number using the port which is already connected to a telephone line. (When the method of connection is a callback reception, it is considered that the telephone number is different from methods of connection in other than the callback reception.)	<ul style="list-style-type: none"> If you want to connect to a different telephone number, Execute Disconnect with respect to the telephone line that is already connected and executes Connect after it gets disconnected. In case of connecting the telephone line with callback reception, use the Connect of the connection method that is executed at the earliest in the same port as callback reception.
0xF10001B	Control Type Mismatch error. An object, whose control type is different from that of the object already connected to the telephone line, tried to Connect.	<ul style="list-style-type: none"> Execute Disconnect for the object currently connected to the telephone line and execute Connect once again after the telephone line gets disconnected.
0xF10001C	Not Disconnected error. When Disconnect method is executed for the object connected to the telephone line, it is found that other objects are in connected state.	<ul style="list-style-type: none"> Execute Disconnect for all the Connected objects. Try Disconnect again for the object that actually performed the telephone line connection.
0xF10001D	Not Connected error. Open was executed before Connect. Or, Disconnect was executed.	<ul style="list-style-type: none"> Execute Open again after executing Connect. Or execute Disconnect again after executing Connect.
0xF10001E	Fatal error	<ul style="list-style-type: none"> Exit the program. Restart the personal computer. Reinstall MX Component. Please contact your local Mitsubishi Electric sales office or representative.

Error code	Error description	Corrective action
0xF100001F	Open time setting error There are some differences in telephone number and the port number settings used during Connect and Open. There are some errors in Connect way.	<ul style="list-style-type: none"> • Check the telephone number and the port number. • Check the Connect way.
0xF1000020	GX Simulator3 did not start error	<ul style="list-style-type: none"> • Start GX Simulator3.
0xF1000021	No data error The size of the specified file is '0'.	<ul style="list-style-type: none"> • Check the file.
0xF1000022	File size error The specified file size is too large.	<ul style="list-style-type: none"> • Check the file.
0xF1000023	File input/output error An error occurred while saving a file.	<ul style="list-style-type: none"> • Check the file name.
0xF1000024	No file found error No more files not found.	<ul style="list-style-type: none"> • End the search.
0xF1000025	File name error The file name is too long. Set a transfer source directory name within 188 bytes. Set a transfer source file name within 318 bytes. Set a transfer destination file name within 259 bytes.	<ul style="list-style-type: none"> • Check the file name.
0xF2000002	There is an error response from the target telephone. Causes can be the following. Communication error has occurred.	<ul style="list-style-type: none"> • Check the value of the properties set in case of program settings type. • Check the contents of the detailed settings that were set using the wizard in case of utility settings type.
0xF2000003	Invalid data was received. Causes can be the following. Incorrect data packet is received due to noise.	<ul style="list-style-type: none"> • Retry. • Check the communication device used at the other end.
0xF2000004	There is no response from the modem. Causes can be the following. Abnormality in the modem Telephone number setting mistake	<ul style="list-style-type: none"> • Check the status of the modem. • Check the telephone number. • If the problem persists even after checking the above points, change the value of the properties set (Properties such as ActConnectionCDWaitTime etc. , which set the timing) in case of program settings type and change the contents of the detailed settings that were set using the wizard in case of utility settings type.
0xF2000005	There are chances that the line is not disconnected.	<ul style="list-style-type: none"> • Check the telephone line.
0xF2000006	The PC modem did not receive the AT command. Causes can be the following. Invalid AT command was specified. Abnormality in the modem.	<ul style="list-style-type: none"> • Check the contents of the AT command. • Check the status of the modem.
0xF2000007	Modem did not respond properly to the standard escape command.	<ul style="list-style-type: none"> • Check the modem. • Confirm whether the value of the time-out is too small. (5000ms or more is recommended.)
0xF2000009	Modem does not respond properly to the line Disconnect command.	<ul style="list-style-type: none"> • Check the modem.
0xF200000A	Target did not receive the signal. The Receive settings of the modem at the other end may be incorrect. The other end may be busy. The telephone number may be incorrect.	<ul style="list-style-type: none"> • Check the Receive settings of the modem at the other end. • Check if the other end is busy. • Check the telephone number.
0xF200000B	Timeout reached for the call back receive waiting time.	<ul style="list-style-type: none"> • Increase the call back receive waiting time ActCallbackReceptionWaitingTimeOut and execute connect again.
0xF200000C	Password of QJ71C24 units could not be resolved.	<ul style="list-style-type: none"> • Set the password to ActPassword property and execute the failed method again.
0xF2010001	The callback line disconnect wait time is other than 0 -180 Seconds. The callback execution delay time is other than 0 -1800 Seconds. The telephone number is more than 62 characters.	<ul style="list-style-type: none"> • Check whether the callback line disconnect wait time is with in 0 - 180 Seconds. • Check whether the callback execution delay time is within 0 - 1800 Seconds. • Check whether the telephone number is less than or equal to 62 characters. • Exit the program and restart the personal computer. • Reinstall MX Component. • Please contact your local Mitsubishi Electric sales office or representative.
0xF2010002	QJ71C24 did not receive the specified connection method. Causes can be the following. Incorrect Connection method Incorrect telephone number for Call back	<ul style="list-style-type: none"> • Check whether the settings of QJ71C24 and the MX Component are matching.

Error code	Error description	Corrective action
0xF2010003	QJ71C24 does not permit the automatic connection (during fixed Call back or when the number is specified.)	<ul style="list-style-type: none"> • Check the settings of QJ71C24.
0xF2100005	There are chances that the line is not disconnected.	<ul style="list-style-type: none"> • If there is no problem with the modem or the telephone line, change the value of the properties set (Properties like ActConnectionCDWaitTime etc. , which set the timing) in case of program settings type and change the contents of the detailed settings that were set using the wizard in case of utility settings type.
0xF2100006	Modem did not receive the startup command AT.	<ul style="list-style-type: none"> • Change the settings of the property ActATCommand in case of program settings type. • Change the command AT that were set using the wizard in case of utility settings type.
0xF2100007	The PC modem does not respond to the Escape command.	<ul style="list-style-type: none"> • If there is no problem with the modem, change the value of the properties set (Properties like ActConnectionCDWaitTime etc. , which set the timing) in case of program settings type and change the contents of the detailed settings that were set using the wizard in case of utility settings type.
0xF2100008	There was no response from the modem for the data sent from the PC.	<ul style="list-style-type: none"> • Change the value of the properties set (Properties such as ActConnectionCDWaitTime etc. , which set the timing) in case of program settings type and change the contents of the detailed settings that were set using the wizard in case of utility settings type.
0xF21000**	There is no response from the modem. Causes can be the following. Abnormality in the modem Telephone number setting mistake	<ul style="list-style-type: none"> • Check the status of the modem. • Check the telephone number. • If the problem persists even after checking the above points, change the value of the properties set (Properties such as ActConnectionCDWaitTime etc. , which set the timing) in case of program settings type and change the contents of the detailed settings that were set using the wizard in case of utility settings type.
0xF202****	There was a communication failure. Following causes can be considered depending on the status. Communication time over (Break in cable, the specified port not supported, mistake in specifying the COM port) Modem's power is switched OFF.	<ul style="list-style-type: none"> • Check whether the cable is broken. • Check whether the specified port is not supported. • Check whether correct COM port is set. • Check if the modem power is switched OFF. • For detailed troubleshooting, please refer to the details of the error code got after replacing the first four digits with "0x0180". e.g. In case of "0xF202480B", please refer to the code "0x0180480B".
0xF3000001	Label error The label character string specified in the method is an unauthorized label character string.	<ul style="list-style-type: none"> • Review the label name. • Start Label Utility, please check System label information. • Refer to the label name format of each function, and check if the correct format is used.
0xF3000002	Label Service not started	<ul style="list-style-type: none"> • Exit the program and restart the personal computer. • Reinstall MX Component. • Start Label Utility, please check System label information.
0xF3000003	MMS Service not started	<ul style="list-style-type: none"> • Exit the program and restart the personal computer. • Reinstall MX Component.
0xF3000004	The specified LabelSpace is currently being used and cannot be opened.	<ul style="list-style-type: none"> • Retry.
0xF3000005	Label data access timeout error	<ul style="list-style-type: none"> • Retry. • Reduce system label definition.
0xF3000006	MUTEX acquisition error Acquisition of MUTEX to exercise exclusive control failed.	<ul style="list-style-type: none"> • Exit the program and restart the personal computer.
0xF4000001 to 0xF400000F	64-bit control error	<ul style="list-style-type: none"> • Exit the program and restart the personal computer. • Reinstall MX Component. • Please contact your local Mitsubishi Electric sales office or representative.
0xF4000010 to 0xF4000012	64-bit control access error	<ul style="list-style-type: none"> • Execute the Open method again after executing the Close method. • Check if 10 minutes or longer have passed since the DotUtilType64 control was last accessed. • Reduce the amount of data handled in a single process. • Exit the program and restart the personal computer. • Reinstall MX Component. • Please contact your local Mitsubishi Electric sales office or representative.
0xF4001001	Not yet open error The Open method is not yet executed.	<ul style="list-style-type: none"> • After executing the Open method, execute the corresponding method.
0xF4001002	Already open error The Open method was executed in the open status.	<ul style="list-style-type: none"> • When changing the communication target CPU, execute the Open method after performing Close.
0xF4001003	Registry search failed.	<ul style="list-style-type: none"> • Exit the program and restart the personal computer. • Exit other programs and secure free memory area. • Reinstall MX Component.

Error code	Error description	Corrective action
0xFF000001	MX Component trial version error Expired MX Component trial version	<ul style="list-style-type: none"> Uninstall the trial version and install the commercial version, or reinstall the trial version.

*1 For the method for increasing the minimum working set size of the personal computer, refer to the following:

 MELSEC Data Link Library Reference Manual

- Error code for the robot controller

Error code	Error description	Corrective action
100	Number of ENQ retries is exceeded.	<ul style="list-style-type: none"> Take measures against noise
101		
102	Number of NACK retries is exceeded.	<ul style="list-style-type: none"> Check the cable connection.
103	The message is too long.	
104	Reception time-out	<ul style="list-style-type: none"> Check the cable connection. Open again.
105	DSR signal was not detected.	
106	The line was disconnected.	<ul style="list-style-type: none"> Check the cable connection.
107	Transmission time-out	
108	Sequence number is incorrect.	<ul style="list-style-type: none"> Take measures against noise
200	Included DLL was not found.	<ul style="list-style-type: none"> Reinstall MX Component.

13.2 Error Codes Returned by CPUs, Modules, and Network Boards

This section explains the error codes returned by CPUs, modules, and network boards.



Error codes may not be returned as described in [Page 644 Error code list](#).

Check [Page 644 Considerations for confirming error codes firstly](#).

Error code list

If a CPU, module, or network board detected an error, any of the error codes indicated in the following table is returned. The two upper bytes denote the error detection module, and the two lower bytes denote the error code returned by the error detection module.

For error details, refer to the manual of CPU, module, or network board corresponds to the error code.

Error code	Error detection module
0x01010000 to 0x0101FFFF *1*2	Motion controller CPU
0x01070000 to 0x0107FFFF *1	CC-Link IE Controller Network board, MELSECNET/H board, CC-Link Ver.2 board
0x01090000 to 0x0109FFFF *1	FXCPU
0x010A0000 to 0x010AFFFF *1	QCPU (Q mode), QSCPU, RCPU, FX5CPU
0x010B0000 to 0x010BFFFF *1	Q series-compatible C24
0x010C0000 to 0x010CFFFF *1	Q series-compatible E71
0x010D0000 to 0x010DFFFF *1	PC CPU module
0x010F0000 to 0x010FFFFF *1	GOT

*1 Refer to "Point."

*2 The error codes are also described in the following section:

[Page 616 Error Code List](#).

Considerations for confirming error codes

The following are the considerations for confirming the error codes returned by CPU, module, and network board.

■Property setting error

If the used system configuration does not match the preset property values, the two upper bytes do not indicate the correct error detection module.

For example, when the property values of FXCPU are set to ActCpuType for QCPU (Q mode), the two upper bytes may indicate that the error detection module is FXCPU.

In such a case, perform the communication again after checking the system configuration and all preset property values.

When the ActUtilType control is used, check the settings of Communication Settings Utility.

■When accessing another station

When accessing another station, the error code of the used relayed module (CC-Link IE Controller Network, MELSECNET/H, CC-Link, Serial communication, Ethernet module) may be entered to the two lower bytes.

In such a case, the two upper bytes that indicate the error detection module may not always match the faulty module. Check the system configuration, and check the manuals of the used CPU, relayed network module, and network board.

Label specification error

The following error code will be returned if an error related to labels in the CPU occurred such as; system label Ver.2 does not exist, the information of system label Ver.2 is modified while reading/writing devices that are corresponding to the system label Ver.2.

Error code	Error description	Corrective action
0x01802001	Label information error	<ul style="list-style-type: none"> When a system label is specified as a label, review the global label name assigned to the system label name in Label Utility. Check if the global labels exist in the CPU.
0x010A4000 to 0x010A4FFF	For the errors and their corrective actions, refer to the following manual.  MELSEC iQ-R CPU Module User's Manual (Application)	

13.3 HRESULT Type Error Codes

Normally, the ActiveX control and the Act control return HRESULT type returned values.

When the custom interface is used, the returned value is equivalent to the returned value of method API.

When the dispatch interface is used, HRESULT type returned values can be acquired by performing the exception processing.

The following table shows HRESULT type returned values of Act controls.

Returned value	Termination status	Description
S_OK	Normal end	The function processing is normally terminated.
S_FALSE	Normal end	The function processing (as ActiveX control) is normally terminated, but the operation (access to programmable controller) failed.
E_POINTER	Abnormal termination	The pointer passed to the function is abnormal.
E_OUTOFMEMORY	Abnormal termination	Memory reservation or object creation failed.
E_FAIL	Abnormal termination	An indefinite error occurred.



If the exception processing to acquire the HRESULT type returned value is not performed, the dispatch interface displays the error dialog box on the operating system level when E_POINTER (E_XXXXX defined returned value) or the like is returned from the Act control.

13.4 Error Codes Displayed on Event Viewer

This section explains the error codes displayed on Windows Event Viewer by MX Component.

MX Component uses MMS (system label database) Service and Label Service to manage label information.

When an error occurs in Label Service, an error description is displayed on the system log in Event Viewer.

For errors regarding Label Service of MX Component, "MXLabelService" is displayed on the source field in Event Viewer.

The following table shows the errors and their corrective actions.

Event ID	Error message	Corrective action
3	MMS Service not started.	<ul style="list-style-type: none"> Wait for the completion of MMS (system label database) Service startup. If MMS Service startup does not complete, exit the program and restart the personal computer. Reinstall MX Component. Please contact your local Mitsubishi Electric sales office or representative.
5	MXLabelService error.(%1,%2) please perform one of the followings. -Re-start Personal computer. -Re-install the application and try again.	<ul style="list-style-type: none"> Exit the program and restart the personal computer. Reinstall MX Component. Please contact your local Mitsubishi Electric sales office or representative.

APPENDIX

Appendix 1 Changes from Previous Version

This section shows the additions and changes with upgrade.

Version	Item	Description	Reference
Version 5.000A	Control	<ul style="list-style-type: none"> Controls for 64-bit are supported. The DotSupportMsg control is integrated into the ActSupportMsg control. 	—
	Function	Functions for 64-bit are supported.	Page 416 FUNCTIONS
	Operating environment	Only Windows 10 is supported.	<ul style="list-style-type: none"> MX Component Version 5 Installation Instructions Page 24 Operating Environment
	Development software	Only the following software is supported: <ul style="list-style-type: none"> Visual Studio 2019 Microsoft Excel 2019 Microsoft Access 2019 	
	Product to be discontinued	Products to be discontinued are no longer supported.	—
	Communication type	The following Communication types are no longer supported. <ul style="list-style-type: none"> Q series bus communication GX Simulator communication Modem communication 	Page 26 Supported Communication Routes
	Installation folder	The default installation folder is changed from "C:\MELSEC" to "C:\Program Files\MELSOFT."	—
	Manual	Available for e-Manual Viewer	—
Version 5.002C	Module	FX5-ENET and FX5-ENET/IP are supported.	Page 31 Module list
	Communication route	The following communication routes are supported: <ul style="list-style-type: none"> Ethernet communication when a connection target CPU is an FX5CPU (using an Ethernet adapter) GOT Transparent Communication when a connection target CPU is an FX5CPU (using an Ethernet adapter) 	<ul style="list-style-type: none"> Page 199 Ethernet Communication Page 333 GOT Transparent Communication
	Development software	The following software is supported: <ul style="list-style-type: none"> Microsoft Excel for Microsoft 365 Microsoft Access for Microsoft 365 	<ul style="list-style-type: none"> MX Component Version 5 Installation Instructions Page 24 Operating Environment
Version 5.003D	Module	FX5S and FX5-CCLGN-MS are supported.	Page 31 Module list
	Operating environment	Windows11 is supported.	<ul style="list-style-type: none"> MX Component Version 5 Installation Instructions Page 24 Operating Environment
	Development software	The following software is supported: <ul style="list-style-type: none"> Visual Studio 2022 Microsoft Excel 2021 Microsoft Access 2021 	
Version 5.004E	Module	R64MTCPU, MELSECWinCPU, and L32HCPU are supported.	Page 31 Module list
	Operating environment	Windows 10 IoT Enterprise 2019 LTSC is supported.	<ul style="list-style-type: none"> MX Component Version 5 Installation Instructions Page 24 Operating Environment
	Network	For CC-Link IE Controller Network communication and CC-Link IE Field Network communication the connected stations of which are RCPUs, the access via MELSEC iQ-R series CC-Link IE TSN, CC-Link IE Controller Network, CC-Link IE Field Network, and Ethernet is supported.	<ul style="list-style-type: none"> Page 309 CC-Link IE Controller Network Communication Page 314 CC-Link IE Field Network Communication
Version 5.006G	Module	RJ71GN11-SX, NZ81GN11-T2, and NZ81GN11-SX are supported.	Page 31 Module list
	Network	The communication using a CC-Link IE TSN board (NZ81GN11-T2, NZ81GN11-SX) is supported.	Page 319 CC-Link IE TSN Communication
Version 5.007H	Network	For a MELSEC iQ-R series redundant CPU, the access to the control system is supported.	<ul style="list-style-type: none"> Page 199 Ethernet communication when the connected station is MELSEC iQ-R series-compatible E71 (TCP) Page 309 CC-Link IE Controller Network communication when the connected station is an RCPU Page 652 For RnPCPU (redundant mode) and RnPSFCPU

Appendix 2 Concept of Routing Parameters

The routing function is used by a station of a CPU module in a multi-level system to perform the transient transmission to a station of another network number.

To perform the routing function, the "Routing parameters" must be set to associate the network numbers and stations acting as bridges.

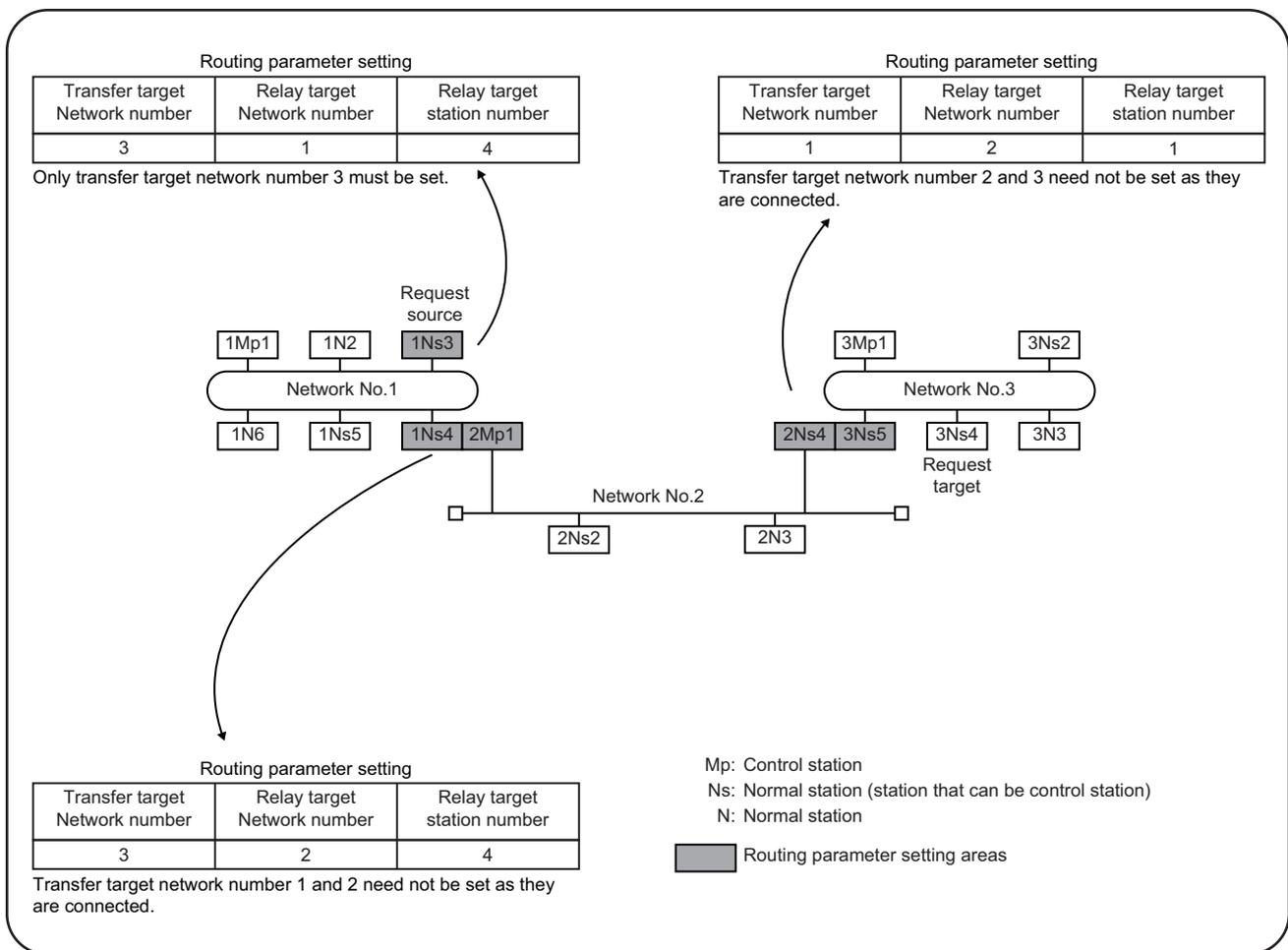
Routing parameter settings

■ Source and relayed station of a CPU module settings

The routing parameters must be set to the request source and relayed station of a CPU module.

- The request source must be set to access the request target.
- The relayed station must be set to access from the request source to the request target and vice versa.
- The request target is not required to be set.

For example, to perform the transient transmission from 1Ns3 to 3Ns4 in the following diagram, the routing parameters must be set to the CPU module 1Ns3 which performs transient transmission, to the CPU modules 1Ns4 and 2Mp1 which serve as bridges, and to the CPU modules 2Ns4 and 3Ns5.



■ "Transfer target Network number" settings

Up to 16 "Transfer target Network number" can be set to a CPU module.

16 different network numbers allow a host station to be a request source or other stations to be accessed via the host station.

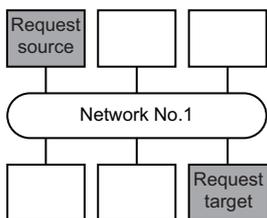


Routing parameter setting areas and data

For transient transmission, the routing parameter setting areas differ according to the system.

■Two-level system

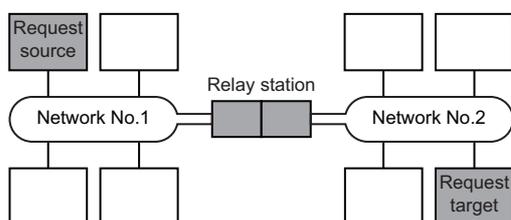
The routing parameters are not required to be set because transient transmission is performed to within the same network.



■Multi-level 1 (two networks)

Set the routing parameters only to the station of the request source. *1

To the request source, set the data to access the request target (network No.2).



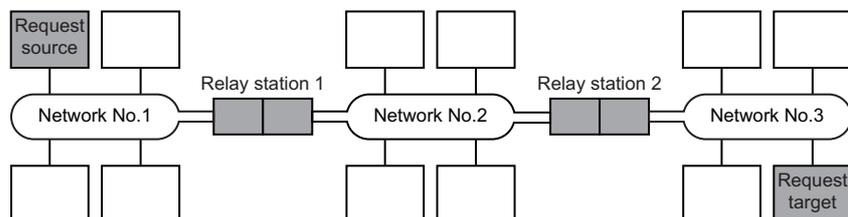
■Multi-level 2 (three networks)

Set the routing parameters to the request source and relayed stations. *1

To the request source, set the data to access the request target (network No.3).

To the relayed station 1, set the data to access the request target (network No.3).

To the relayed station 2, set the data to access the request source (network No.1).



■Multi-level 3 (four or more networks)

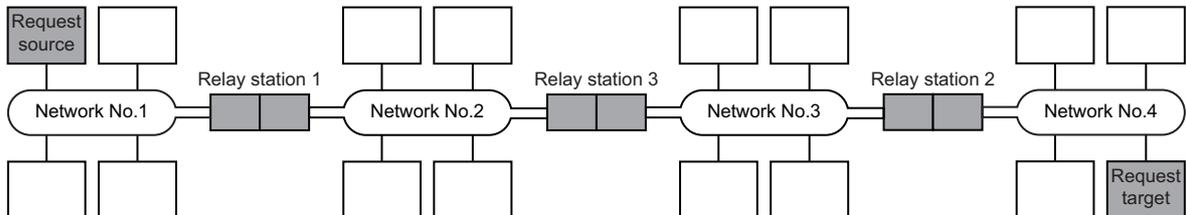
Set the routing parameters to the request source and relayed stations. *1

To the request source, set the data to access the request target (network No.4).

To the relayed station 1 (the nearest relayed station to the request source), set the data to access the request target (network No.4).

To the relayed station 2 (the nearest relayed station to the request target), set the data to access the request source (network No.1).

To the relayed station 3 (relayed station other than 1 and 2), set the data to access the request target (network number No.4) and request source (network No.1).



*1 The following explains the case when the request source is a personal computer connected to Ethernet.

The routing parameter settings are not necessary for the request source.

The routing parameter settings (routing settings for MELSEC iQ-R series) are necessary for relay stations so that they can access the request source.

Set the routing parameter by referring the following manual:

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

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

Appendix 3 Multi-CPU System

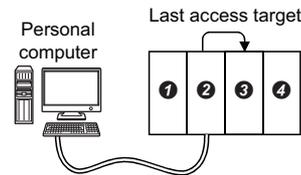
The valid CPU number specified for a multi-CPU system is that of the last accessed station only.

When accessing the non-controlled CPU of the relayed module on the accessed station, use the modules of function version B as the relayed modules and QCPUs (Q mode) on the host station, all relayed stations and accessed station.

CPU COM communication

Ex.

When the CPU module No. 3 (0x3E2) is specified for access, the CPU ③ can be accessed.



- ①: QCPU (Q mode) No. 1
- ②: QCPU (Q mode) No. 2
- ③: QCPU (Q mode) No. 3
- ④: QCPU (Q mode) No. 4

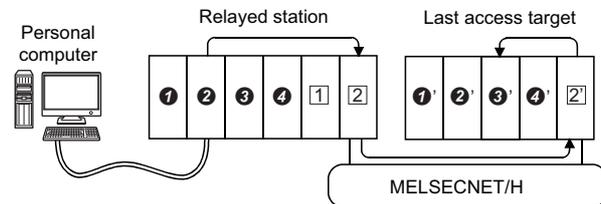
CPU COM communication (via MELSECNET/H)

Ex.

When the CPU module No. 3 (0x3E2), network number 2 and station number 2 are specified for access, the CPU ③' can be accessed.

The CPU number cannot be specified for the relayed station.

Therefore, if the network number 1 is accessed in the following case, an error will occur because the network number controlled by the CPU ② is only "2."



- ①: QCPU (Q mode) No. 1
- ②: QCPU (Q mode) No. 2
- ③: QCPU (Q mode) No. 3
- ④: QCPU (Q mode) No. 4

- ①: MELSECNET/H module controlled by ① (Network number: 1, station number: 1)
- ②: MELSECNET/H module controlled by ② (Network number: 2, station number: 1)
- ②': MELSECNET/H module controlled by ②' (Network number: 2, station number: 2)

Appendix 4 The Number of Mountable Network Modules for Q00UJCPU, Q00UCPU, and Q01UCPU

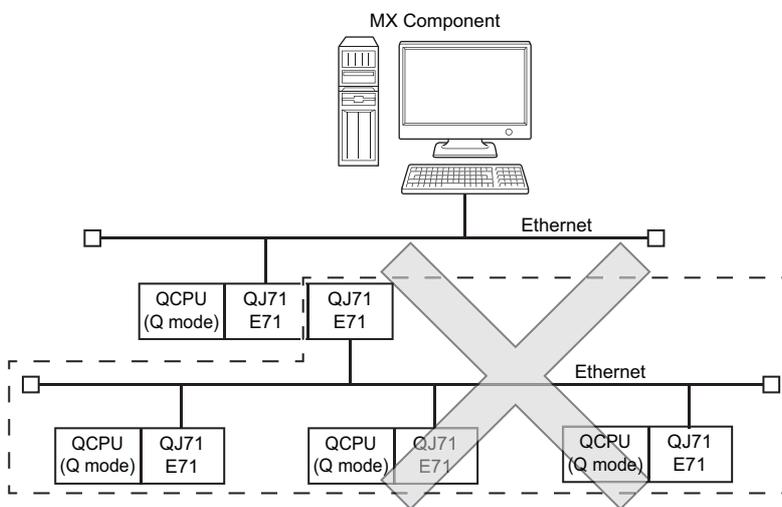
The following shows the number of mountable network modules when using Q00UJCPU, Q00UCPU, and Q01UCPU.

Network Module	Number of Mountable Modules
MELSECNET/H module	1 module
Ethernet module	1 module
CC-Link module (Function version B or later)	2 modules
CC-Link IE Controller Network module	1 module

Therefore, the following systems cannot be configured.

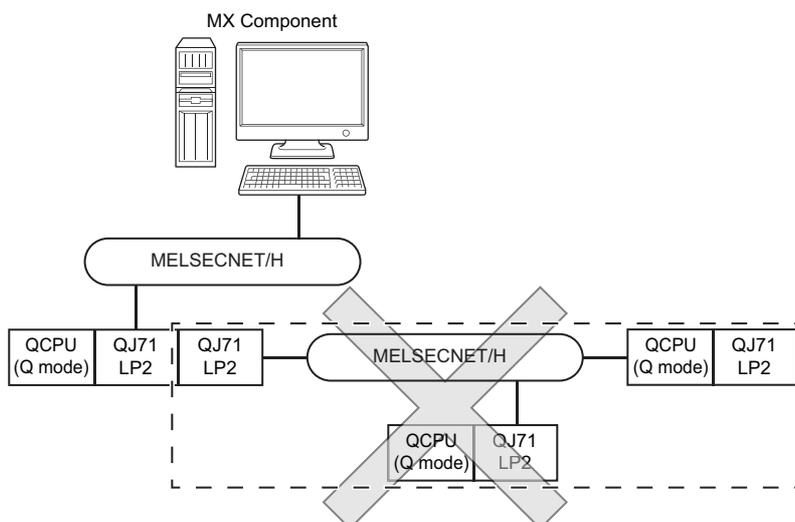
Ex.

Since the number of applicable Ethernet modules is 1, the part of the system indicated by the dotted line cannot be configured.



Ex.

Since the number of applicable MELSECNET/H modules is one, the part of the system indicated by the dotted line cannot be configured.



A

Appendix 5 Compatibility with Redundant CPU

This section explains the compatibility of MX Component with a redundant CPU.

For RnPCPU (redundant mode) and RnPSFCPU

Redundant CPU specification

In redundant CPU specification in MELSEC iQ-R series, select either "Control system" or "Not specified" to access the compatible redundant CPU.

When selecting "Control system," MX Component connects to the control system and continues to access the control system in response to system switching.

For conditions of automatic communication route switching, refer to the following:

 Page 654 Route switch conditions

When selecting "Not specified," MX Component connects to the connection target CPU module as before.

Redundant CPU specification setting can either be set on the utility setting type Communication Setting Wizard screen or on the program setting type control property.

To judge which system in the redundant CPU system is being accessed by MX Component, monitor the following special relays.

■When checking which system is being accessed, System A or System B

Indicate system A/system B of a redundant system.

Remain ON/OFF even if the tracking cable is disconnected while the redundant system is running.

Special relay	System A	System B	System not determined.
SM1632 (System A identification flag)	ON	OFF	OFF
SM1633 (System B identification flag)	OFF	ON	OFF

■When checking the control/standby system status

Indicate the CPU module control/standby status.

Remain ON/OFF even if the tracking cable is disconnected while the redundant system is running.

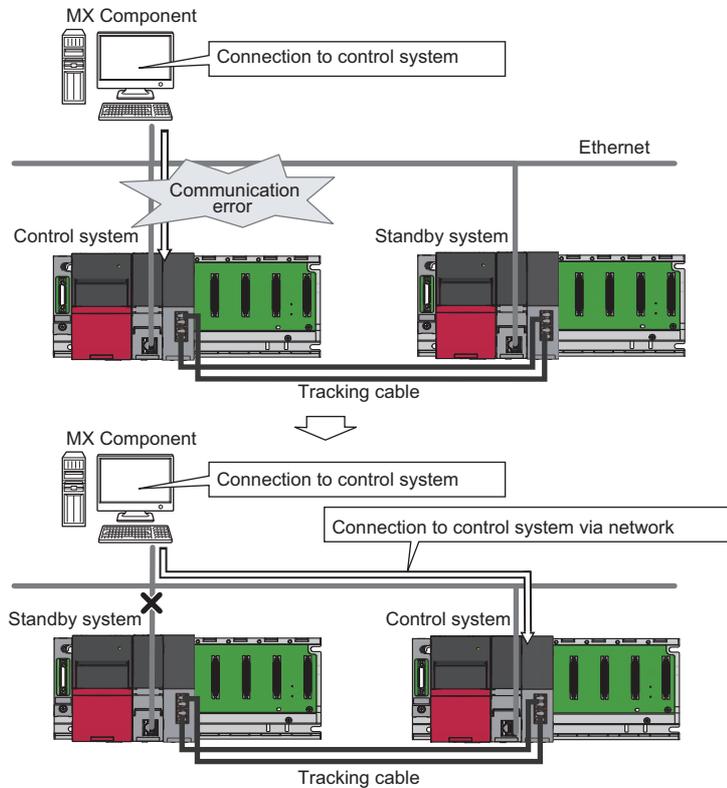
Special relay	Control system	Standby system	System not determined.
SM1634 (Control system status)	ON	OFF	OFF
SM1635 (Standby system status)	OFF	ON	OFF

Operation at occurrence of system switching

Access to the control system after system switching is continued even if a communication error occurs. The following shows an example of Ethernet connection.

Ex.

When communication error occurs



A

Point

It may take time to connect to the control system and start communication after a communication error occurs.

Automatic switching of communication route

If a communication error occurs during access to the redundant CPU connected to Ethernet and CC-Link IE Controller Network, the communication route is automatically switched to continue access to the control system.

Hereinafter, this automatic switching of the communication route is referred to as route switch.

For details on special registers related to route switch, refer to the following:

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

■Route switch conditions

When a CPU is accessed under the following conditions, access to the redundant CPU is continued by route switch even if a communication error occurs.

- For Ethernet connection

Supported module: RJ71EN71

Item		Condition for continued access
Operation mode of programmable controller		Backup mode
MX Component communication settings	IP address	Control system IP address*1
	Communication protocol	TCP
	Redundant CPU specification	Control system

*1 For details on control system IP address, refer to the following:

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

- For CC-Link IE Controller Network connection

Supported module: RJ71GP21-SX

Item		Condition for continued access
Operation mode of programmable controller		Backup mode
MX Component communication settings	Station number	System A/B station number
	Redundant CPU specification	Control system

For QnPRH

Redundant CPU specification

In redundant CPU specification in Q series, select either "Control system" or "Not specified" to access the compatible redundant CPU.

When selecting "Control system," MX Component connects to the control system and continues to access the control system in response to system switching.

When selecting "Not specified," MX Component connects to the connection target CPU module as before.

Redundant CPU specification setting can either be set on the utility setting type Communication Setting Wizard screen or on the program setting type control property.

To judge which system in the redundant CPU system is being accessed by MX Component, monitor the following special relays.

■When checking which system is being accessed, System A or System B

Indicate system A/system B of a redundant system.

Remain ON/OFF even if the tracking cable is disconnected while the redundant system is running.

Special relay	System A	System B	System not determined.(At the time of TRK.CABLE ERR.(Error code: 6120) occurrence)
SM1511 (System A identification flag)	ON	OFF	OFF
SM1512 (System B identification flag)	OFF	ON	OFF

■When checking the control/standby system status

Indicate the CPU module control/standby status.

Remain ON/OFF even if the tracking cable is disconnected while the redundant system is running.

Special relay	Control system	Standby system	System not determined.(At the time of TRK.CABLE ERR.(Error code: 6120) occurrence)
SM1515 (Control system status)	ON	OFF	OFF
SM1516 (Standby system status)	OFF	ON	OFF



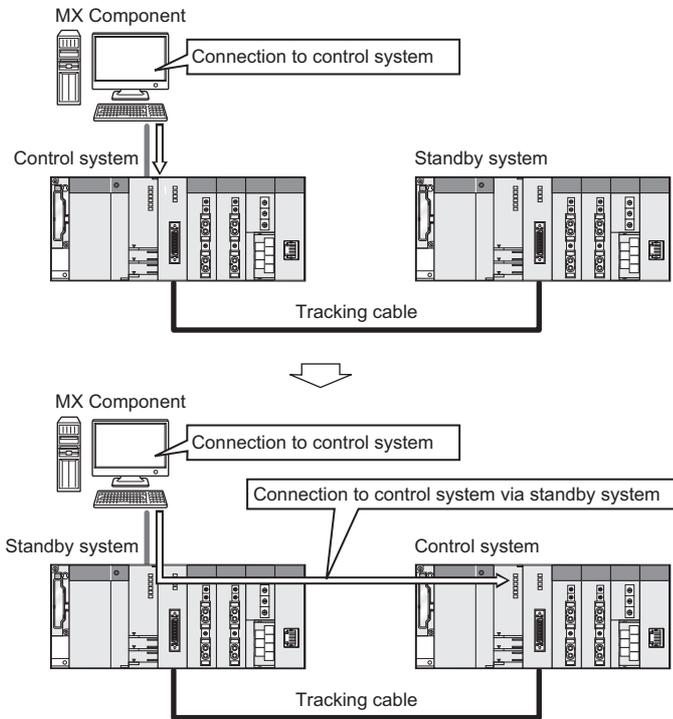
Operation at occurrence of system switching

When system switching occurs during access to the redundant CPU after selection of "Control system," access is continued as described below.

■ Connection via other than MELSECNET/H, Ethernet or CC-Link IE Controller Network

Access to the control system after system switching is continued.

The following shows an example of CPU direct connection.



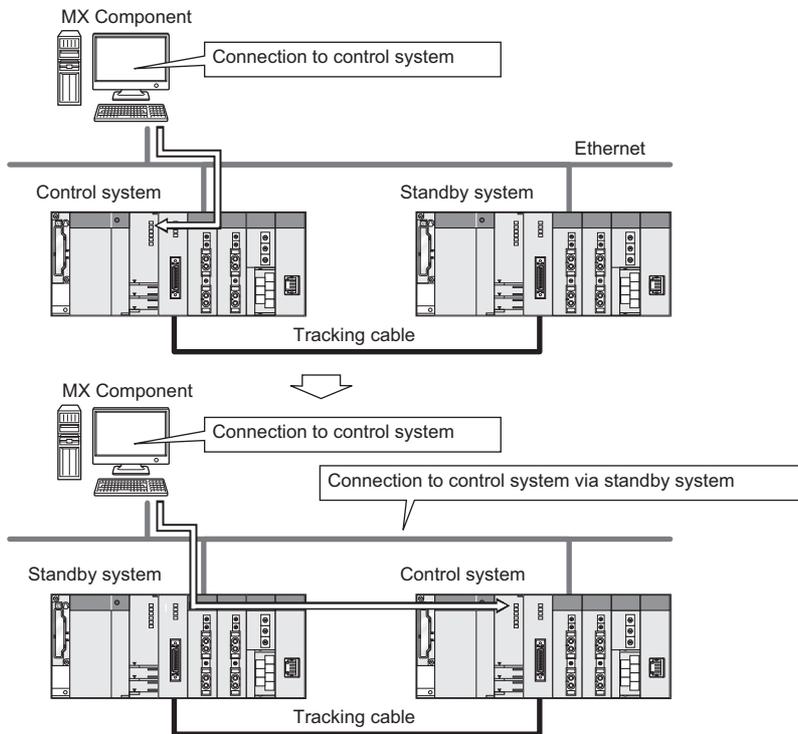
■Connection to MELSECNET/H, Ethernet or CC-Link IE Controller Network

Access to the control system after system switching is continued even if a communication error occurs.

The following shows an example of Ethernet connection.

Ex.

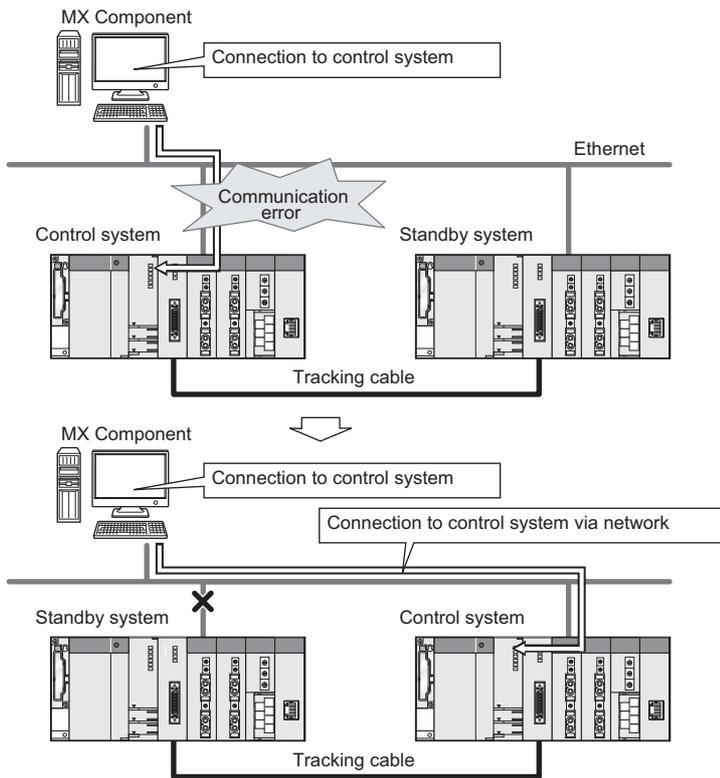
When communication is normal



A

Ex.

When communication error occurs



Point

In the case of Ethernet connection, it may take time to connect to the control system and start communication after a communication error occurs.

Automatic switching of communication route

If a communication error occurs during access to the redundant CPU connected to MELSECNET/H, Ethernet or CC-Link IE Controller Network in Control system specification, the communication route is automatically switched to continue access to the control system.

Hereinafter, this automatic switching of the communication route is referred to as route switch.

The following describes the route switch conditions, how to check for route switch occurrence, and examples of access by route switch.

Route switch conditions

When a CPU is accessed under the following conditions, access to the redundant CPU is continued by route switch even if a communication error occurs.

	Condition for continued access
Operation mode	Backup mode, separate mode ^{*1}
Target system	Control system

^{*1} In the separate mode, a system switching by the system does not occur at power OFF/RESET of the control system.

However, if a tracking error occurred at a start of communication, access to the control system is not continued by route switch even if tracking is recovered after that.

Includes the status in which either redundant CPU is shut OFF or reset.

How to check whether route switch occurred or not

When communication is performed with the redundant system specified, whether communication is continued by route switch due to communication error can be estimated.

Special relay and special registers to be monitored as well as the estimated possibility of route switch are shown below.

SM1600 ^{*1}	SD1590 ^{*2}	SD1690 ^{*2}	Possibility of route switch	Reference
OFF	Either one is other than 0		Since a system switching request from the network module was detected, route switch may be executed.	Page 660 Examples of access by route switch When system switching occurs at communication error
ON	0	0	Since another system error occurred, route switch may be executed.	Page 660 Examples of access by route switch When standby system error occurs
ON	Either one is other than 0		Since another system error occurred or a system switching request from the network module was detected, route switch may be executed.	Page 660 Examples of access by route switch When system switching occurs at communication error, When standby system error occurs

^{*1} Even if SM1600 is ON, route switch does not occur when the CPU is not accessed via the tracking cable.

^{*2} When using SM1600, SD1590 and SD1690 to estimate whether route switch has occurred or not for the redundant CPU connected to Ethernet, select the following items in the redundant setting of the network parameter of GX Works2.

Issue a system switching request at disconnection detection.

Issue a system switching request at communication error.

Check the following based on the status of the above special relay and special registers, and remove the error cause.

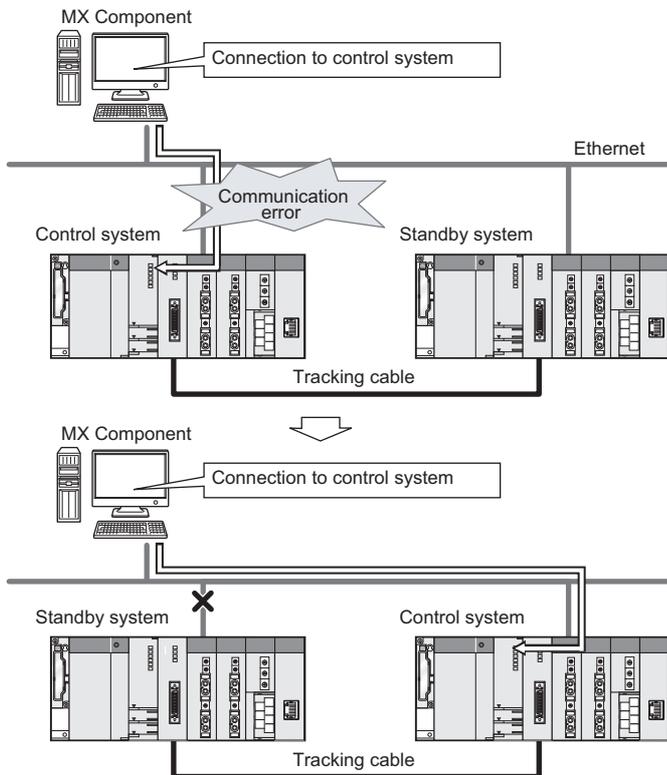
- Check the redundant CPU for an error.
- Check the tracking cable status and whether the tracking cable is correctly connected.
- Check the relevant network module for an error and the network where the relevant network module is connected for an error.

■ Examples of access by route switch

The following shows examples of route switch during access to the control system by Ethernet connection.

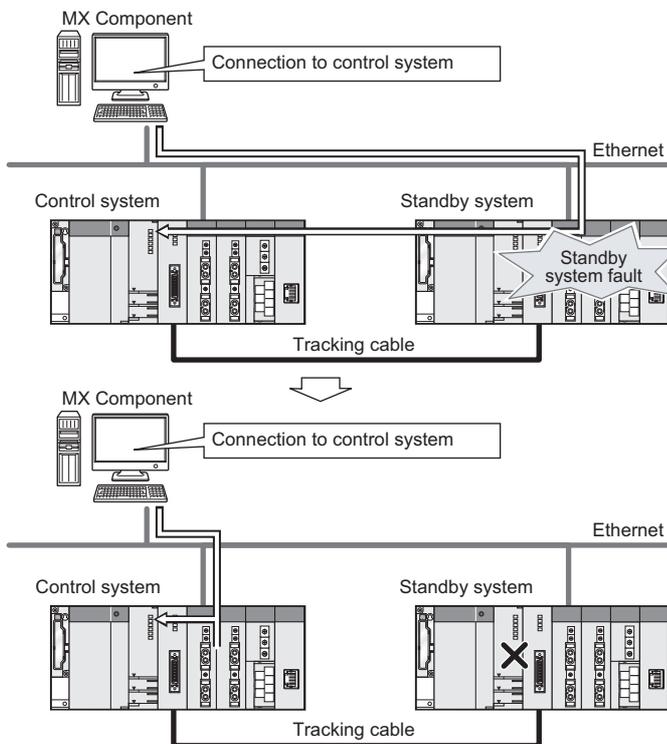
Ex.

When system switching occurs at communication error



Ex.

When standby system error occurs



- Route switch is not executed if a communication error occurs at a communication start for the redundant CPU specified as the target. (A communication error occurs.)
- In the case of Ethernet connection, it may take time to connect to the control system and start communication after a communication error occurs.
- If a communication error occurs, refer to the following and remove the communication disturbance.

☞ Page 659 How to check whether route switch occurred or not

☞ Page 660 Examples of access by route switch

Special relays and special registers for estimating route switch occurrence

The following indicates details of the special relay and special registers to be monitored when estimating whether route switch occurred or not.

Number	Name	Description	Explanation
SM1600	Other system error flag	OFF : No error ON : Error	<ul style="list-style-type: none"> • Turn ON when an error occurs during redundant system error check. (Turn on when either of bits for SD1600 is ON.) • Turn OFF after recovery from error.
SD1590	Module number for network module requesting path switch in host system	Module number for network module requesting path switch in host system	<ul style="list-style-type: none"> • Any of the following bits turns ON corresponding to module number for network module requesting path switch in host system. • Turn OFF by the system after recovery from error of the relevant module by user. <div style="text-align: center;"> <p>SD1590 bit diagram: b15 to b11 to b1 b0. Bit status: 0: OFF, 1: ON. Legend: Module No. 0: Invalid, as CPU module uses 2 slots; Module No. 1: Indicates the module to the right of CPU module; Module No. 11: Indicates the module at the right end of a 12-I/O slot base (Q312B).</p> </div> <ul style="list-style-type: none"> • Refer to SD1690 for module number for network module requesting path switch in other system.
SD1690	Module number for network module requesting path switch in other system	Module number for network module requesting path switch in other system	<ul style="list-style-type: none"> • Any of the following bits turns ON corresponding to module number for network module requesting path switch in other systems. • Turn OFF by the system after recovery from error of the relevant module by user. <div style="text-align: center;"> <p>SD1690 bit diagram: b15 to b11 to b1 b0. Bit status: 0: OFF, 1: ON. Legend: Module No. 0: Invalid, as CPU module uses 2 slots; Module No. 1: Indicates the module to the right of CPU module; Module No. 11: Indicates the module at the right end of a 12-I/O slot base (Q312B).</p> </div> <ul style="list-style-type: none"> • Refer to SD1590 for module number for network module requesting path switch in host system.

Combination table

Communications via Q series redundant type extension base unit (Q65WRB) are supported.

The following table shows the supported/unsupported combinations.

○: Applicable, ×: Not applicable

Product name	Module model	Function version	Combination	
			When mounted to the main base	When mounted to the extension base
MELSECNET/H module	QJ71LP21-25	D or later	○	×
	QJ71LP21S-25	D or later	○	×
	QJ71LP21G	D or later	○	×
	QJ71BR11	D or later	○	×
	QJ72LP25-25	—	×	×
	QJ72LP25G	—	×	×
	QJ72BR15	—	×	×
	QJ71LP21	—	×	×
Ethernet module	QJ71E71	—	×	×
	QJ71E71(N1)-B2	D or later	○	○
	QJ71E71(N1)-B5	D or later	○	○
	QJ71E71-100	D or later	○	○
Serial communication module	QJ71C24N	—	×	○
	QJ71C24N-R2	—	×	○
	QJ71C24N-R4	—	×	○
CC-Link module	QJ61BT11N	—	○ ^{*1}	○
CC-Link IE Controller Network module	QJ71GP21-SX	D or later	○	×
	QJ71GP21S-SX	D or later	○	×

*1 Cannot be used when the first five digits of the serial number is 06051 or lower.

Appendix 6 Character Strings that cannot be Used for Label Names

Reserved word

The reserved words cannot be used for label names.

The following tables shows character strings that cannot be used for label names.

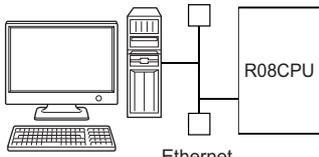
Category	Character string
Class identifier	VAR, VAR_RETAIN, VAR_ACCESS, VAR_CONSTANT, VAR_CONSTANT_RETAIN, VAR_INPUT, VAR_INPUT_RETAIN, VAR_OUTPUT, VAR_OUTPUT_RETAIN, VAR_IN_OUT, VAR_IN_EXT, VAR_EXTERNAL, VAR_EXTERNAL_CONSTANT, VAR_EXTERNAL_CONSTANT_RETAIN, VAR_EXTERNAL_RETAIN, VAR_GLOBAL, VAR_GLOBAL_CONSTANT, VAR_GLOBAL_CONSTANT_RETAIN, VAR_GLOBAL_RETAIN
Data Type	BOOL, BYTE, INT, SINT, DINT, LINT, UINT, USINT, UDINT, ULINT, WORD, DWORD, LWORD, ARRAY, REAL, LREAL, TIME, STRING, TIMER, COUNTER, RETENTIVETIMER, POINTER, Bit, Word [Unsigned]/Bit String [16-bit], Double Word [Unsigned]/Bit String [32-bit], Word [Signed], Double Word [Signed], FLOAT (Single Precision), FLOAT (Double Precision), String, Time, Timer, Counter, Retentive Timer, Pointer
Data type hierarchy	ANY, ANY_NUM, ANY_BIT, ANY_REAL, ANY_INT, ANY_DATE ANY_SIMPLE, ANY16, ANY32
Device name	X, Y, D, M, T, B, C, F, L, P, V, Z, W, I, N, U, J, K, H, E, A, SD, SM, SW, SB, FX, FY, DX, DY, FD, TR, BL, SG, VD, ZR, ZZ
Character string recognized as device (Device name + numeral)	X0 or the like
ST operator	NOT, MOD (,), -
IL operator	LD, LDN, ST, STN, S, S1, R, R1, AND, ANDN, OR, ORN, XOR, XORN, ADD, SUB, MUL, DIV, GT, GE, EQ, NE, LE, LT, JMP, JMPC, JMPCN, CAL, CALC, CALCN, RET, RETC, RETCN LDI, LDP, LDF, ANI, ANDP, ANDF, ANB, ORI, ORP, ORF, ORB, MPS, MRD, MPP, INV, MEP, MEF, EGP, EGF, OUT(H), SET, RST, PLS, PLF, FF, DELTA(P), SFT(P), MC, MCR, STOP, PAGE, NOP, NOPLF
SFC instruction	SFCP, SFCPEND, BLOCK, BEND, TRANL, TRANO, TRANA, TRANC, TRANCA, TRANOA, SEND, TRANOC, TRANOCA, TRANCO, TRANCOC, STEPN, STEPDP, STEPSC, STEPSE, STEPST, STEPR, STEPC, STEPG, STEPI, STEPID, STEPISC, STEPISE, STEPIST, STEPIR, TRANJ, TRANOJ, TRANOCJ, TRANCJ, TRANCOJ, TRANCOCJ
ST code body	RETURN, IF, THEN, ELSE, ELSIF, END_IF, CASE, OF, END_CASE, FOR, TO, BY, DO, END_FOR, WHILE, END_WHILE, REPEAT, UNTIL, END_REPEAT, EXIT, TYPE, END_TYPE, STRUCT, END_STRUCT, RETAIN, VAR_ACCESS, END_VAR, FUNCTION, END_FUNCTION, FUNCTION_BLOCK, END_FUNCTION_BLOCK, STEP, INITIAL_STEP, END_STEP, TRANSITION, END_TRANSITION, FROM, UNTILWHILE
Function name in application function	Function names in application functions AND_E, NOT_E or the like
Function block name in application function	Function block names in application functions CTD, CTU or the like
Symbol	/, \, *, ?, <, >, , ", :, [,], ,, =, +, %, ', ~, @, {, }, &, ^, ., ., tab character ; !, #, \$, `
Date and time literal	DATE, DATE_AND_TIME, DT, TIME, TIME_OF_DAY, TOD
Others	ACTION, END_ACTION, CONFIGURATION, END_CONFIGURATION, CONSTANT, F_EDGE, R_EDGE, AT, PROGRAM, WITH, END_PROGRAM, TRUE, FALSE, READ_ONLY, READ_WRITE, RESOURCE, END_RESOURCE, ON, TASK, EN, ENO, BODY_CCE, BODY_FBD, BODY_IL, BODY_LD, BODY_SFC, BODY_ST, END_BODY, END_PARAMETER_SECTION, PARAM_FILE_PATH, PARAMETER_SECTION, SINGLE, RETAIN, INTERVAL
String that starts with K1 to K8	K1AAA or the like
Statement in ladder language	;FB BLK START, ;FB START, ;FB END, ;FB BLK END, ;FB IN, ;FB OUT, ;FB_NAME, ;INSTANCE_NAME, ;FB, ;INSTANCE
Common instruction	MOV or the like
Windows reserved word	COM1, COM2, COM3, COM4, COM5, COM6, COM7, COM8, COM9, LPT1, LPT2, LPT3, LPT4, LPT5, LPT6, LPT7, LPT8, LPT9, AUX, CON, PRN, NUL
Surrogate pair	0xD800 to 0xDBFF, 0xDC00 to 0xDFFF
Control code	U+0000 to U+001F, U+0080 to U+009F, U+00A0 to U+00BF, U+FFFE, U+FFFF
Environment dependent character	

Considerations on using labels

- More than 32 characters cannot be used for system label Ver.1.
- More than 256 characters cannot be used for system label Ver.2.
(More than 32 characters cannot be used for a system label list name, and more than 246 characters cannot be used for a structure name.)
- A space cannot be used.
- A numeral cannot be used at the beginning of label name.
- A label name is not case-sensitive.
- An underscore (`_`) cannot be used at the beginning or end of label name.
Consecutive underscores (`_`) cannot be used for a data name or a label name.
- The digit-specified bit devices cannot be used.
- The indexing cannot be used.
- The buffer memory cannot be specified.
- The device check cannot be executed when registering labels.

Appendix 7 Performance

This section explains the performances in the following system configuration.

Image diagram	Item	Description	
 <p>Ethernet</p>	Personal computer	OS	Windows 10 Enterprise (64-bit version)
		CPU	Intel Core i5-4590 (3.30GHz)
		Memory	16 GB
	Target CPU	R08CPU (Connecting personal computer and target CPU on 1:1 basis)	
	Network	TCP/IP	
	Number of system label registration	1000	

Performance of MX Component (Control)

Time performance

This explains read/write performance of the device and label.

Processing/Number of label registration		Processing time (ms)				
		Device	system label Ver.1		system label Ver.2	
			first time	second time	first time	second time
Batch read	1 word (1 label)	5.9	230.5	10.7	174.4	12.5
	960 words (1 label)	12.3	227.3	17.3	182.0	19.8
	12288 words (1 label)	157.9	353.5	164.5	330.3	166.9
Batch write	1 word (1 label)	5.6	218.1	11.0	177.5	12.0
	960 words (1 label)	17.6	232.0	22.6	189.9	24.6
	12288 words (1 label)	161.4	367.7	170.1	327.3	171.6
Random read	1 word (1 label)	5.6	31.0	4.5	40.8	5.0
	128 words (128 labels)	24.1	1802.7	38.2	2381.9	45.3
	256 words (256 labels)	42.3	3550.2	136.3	4866.6	169.8
Random write	1 word (1 label)	5.0	26.2	4.3	38.5	5.0
	128 words (128 labels)	30.5	1817.0	41.8	2334.5	47.0
	256 words (256 labels)	52.6	3550.1	140.2	4799.8	166.1

Memory performance

The following shows the amount of cache memory used for DotUtilType control.

Processing		Memory usage amount [KB]	
		system label Ver.1	system label Ver.2
Batch read	1000 labels	(160) ^{*1}	(160) ^{*1}
Batch write			
Random read			
Random write			

*1 Value in the parentheses is a theoretical value.

Cache size (byte) = 160 × Number of system labels used

160 (byte) = Cache data size per one system label

Number of system labels = Specify a total number of 1000 different system labels

Performance of the Label Service

Time performance

This explains time performance of the Label Service.

Processing	Type of system label	Processing time (ms)
Memory load processing at startup of personal computer by Label service	System label of structure type	4101
	System label other than structure type	6161
Update of system label data	System label of structure type	5064
	System label other than structure type	7081

Memory performance

The following shows the amount of memory usage (working set) of the Label Service.

- Measurement condition: A number of registered system labels: 1000

Type of system label	Memory usage amount [KB]	
	system label Ver.1	system label Ver.2
System label of structure type	30256	30268
System label other than structure type	31604	32156

Point

A number of registered system labels written above as measurement condition differ depending on whether the data type is structure type or other than structure type.

- Other than structure type: Number of registered system labels = Number of system labels registered in Label Utility
- Structure type: Number of registered system labels = (1+ Number of structure elements) × Number of system labels registered in Label Utility

As for "A number of registered system labels: 1000" in the table above, 100 system labels of which data types are structure, and which have nine structure elements are defined. The formula is as follows: $(1 + 9) \times 100 = 1000$.

Performance of the Label Utility

Time performance

This explains time performance of the Label Utility.

Processing	Type of system label	Processing time (s)	
		system label Ver.1	system label Ver.2
Change of label space	System label of structure type	7.2	6.2
	System label other than structure type	7.7	8.0
Save of label space	System label of structure type	8.7	6.8
	System label other than structure type	9.2	9.1
Import	System label of structure type	0.9	0.6
	System label other than structure type	0.8	0.9
Export	System label of structure type	0.8	0.9
	System label other than structure type	0.8	0.9

The processing time of "Save of label space" in the table above will be increased as the number of registered system labels is increased because the check processing before the save takes longer time for its processing.

Therefore, the processing time is increased more than the increase rate of a number of registered system labels.

Memory performance

The following shows the amount of memory usage (working set) of the Label Utility.

- Measurement condition: A number of registered system labels: 1000

Type of system label	Memory usage amount [KB]	
	system label Ver.1	system label Ver.2
System label undefined	4108	4316
System label of structure type	8608	7500
System label other than structure type	12872	12256

DISK occupied size

The following table shows the DISK occupied size of the system label database.

Type of system label	DISK occupied size (KB)	
	system label Ver.1	system label Ver.2
System label undefined (A number of registered system labels: 0)	1756	24
System label of structure type (A number of registered system labels: 1000)	2310	89
System label other than structure type (A number of registered system labels: 1000)	2370	246

Point

A number of registered system labels written above as measurement condition differ depending on whether the data type is structure type or other than structure type.

- Other than structure type: Number of registered system labels = Number of system labels registered in Label Utility
- Structure type: Number of registered system labels = (1+ Number of structure elements) × Number of system labels registered in Label Utility

As for "A number of registered system labels: 1000" in the table above, 100 system labels of which data types are structure, and which have nine structure elements are defined. The formula is as follows: $(1 + 9) \times 100 = 1000$.

Appendix 8 Installing a USB Driver

To communicate with a CPU module via USB, installing a USB driver is required.

Operating procedure

1. Connect a personal computer and a CPU module with a USB cable, and turn the power of the CPU module ON.
2. Right-click "Unknown Device" and click "Update driver" in Windows Device Manager.
3. Select "Browse my computer for drivers" on the "Update Drivers" screen, and specify 'Easysocket\USBDrivers' in the folder where MX Component Version 5 is installed on the next screen.

When an installation folder is not changed at the installation, the installation folder is as follows:

- 64-bit version operating system: C:\Program Files (x86)\MELSOFT\Easysocket\USBDrivers
- 32-bit version operating system: C:\Program Files\MELSOFT\Easysocket\USBDrivers

If multiple MELSOFT products are already installed, specify 'Easysocket\USBDrivers' which is in the folder of the first product installed.

If 'Easysocket\USBDrivers' does not exist, search for 'USBDrivers' in the Windows explorer.

The installation succeeded if "MITSUBISHI Easysocket Driver" is displayed on "Universal Serial Bus controllers."

Appendix 9 Programming Examples for Monitoring Word Device Status

This section explains the programming examples to monitor word devices for negative values using the `EntryDeviceStatus` function.

When using Visual Basic .NET

The following is a programming example to monitor D0 for -10, D1 for 0, and D2 for 10 using Visual Basic .NET.

```
Dim szDevice As String           'Checked device list
Dim IInputData(2) As Long        'Set value
Dim IEntryData(2) As Long        'Value set to argument of EntryDeviceStatus
Dim IReturnCode As Long          'Returned value to EntryDeviceStatus
Dim ICount As Long               'Loop counter
'Sets D0, D1 and D2 to the checked device list.
szDevice = "D0" + vbLf + "D1" + vbLf + "D2"
'Sets the checked device value "-10" for D0.
IInputData(0) = -10
'Sets the checked device value "0" for D1.
IInputData(1) = 0
'Sets the checked device value "10" for D2.
IInputData(2) = 10
'If the set value is negative, stores "0"s into the upper 2 bytes
'for conversion into the value to be set to EntryDeviceStatus.
'Loops through the number of device points.
For ICount = 0 To 2
    'If the set value is negative
    If IInputData(ICount) < 0 Then
        'Masks with 65535 (0000FFFF[hex]) to store "0"s into upper 2 bytes.
        IEntryData(ICount) = IInputData(ICount) And 65535
    Else
        'If the set value is positive, assigns the value as-is to IEntryData.
        IEntryData(ICount) = IInputData(ICount)
    End If
Next
'Executes EntryDeviceStatus.
IReturnCode = AxActUtilType1.EntryDeviceStatus(szDevice, 3, 5, IEntryData(0))
```

■When IData = -1

```
Private Sub AxActUtilType1_OnDeviceStatus(ByVal szDevice As String, ByVal IData As Long, ByVal IReturnCode As Long)
Dim ICheckData As Long           'Value set to EntryDeviceStatus (value before "0"s are stored into the upper 2 bytes)
'If the device value whose condition was established is a WORD type negative value (greater than 32767 (7FFF[Hex])
If IData > 32767 Then
    'Since "0"s are stored in the upper 2 bytes,
    'calculates OR with FFFF0000[Hex] to convert the device value into a LONG type negative value.
    ICheckData = IData Or &HFFF0000
Else
    'If the device value whose condition was established is positive, assigns the value as-is to ICheckData.
    ICheckData = IData
End If
End Sub
```

When using Visual C++

The following is a programming example to monitor D0 for -10, D1 for 0, and D2 for 10 using Visual C++.

```
CString szDevice;           // Checked device list
LONG IInputData[3];        // Set value
LONG IEntryData[3];        // Value set to argument of EntryDeviceStatus
LONG IReturnCode;          // Returned value to EntryDeviceStatus
LONG ICount;               // Loop counter
// Sets D0, D1 and D2 to the checked device list.
szDevice = "D0\nD1\nD2";
// Sets the checked device value "-10" for D0.
IInputData[0] = -10;
// Sets the checked device value "0" for D1.
IInputData[1] = 0;
// Sets the checked device value "10" for D2.
IInputData[2] = 10;

// If the set value is negative, stores "0"s into the upper 2 bytes
// for conversion into the value to be set to EntryDeviceStatus.
// Loops through the number of device points.
for(ICount = 0; ICount <= 2; ICount++) {
    // If the set value is negative
    if (IInputData[ICount] < 0) {
        // Masks with 0x0000FFFF to store "0"s into upper 2 bytes.
        IEntryData[ICount] = IInputData[ICount] & 0x0000FFFF;
    }
    // If the set value is positive
    }else{
        // Assigns the value as-is to IEntryData.
        IEntryData[ICount] = IInputData[ICount];
    }
}

// Executes EntryDeviceStatus.
IReturnCode = m_Actutltype.EntryDeviceStatus(szDevice,3,5,IEntryData);
```

■ When IData = -1

```
void CSampleDlg::OnDeviceStatusActutltype1(LPCTSTR szDevice, long IData, long IReturnCode)
{
    LONG ICheckData;        // Value set to EntryDeviceStatus (value before "0"s are stored into the upper 2 bytes)
    // If the device value whose condition was established is a WORD type negative value
    // (greater than 7FFF[Hex])
    if(IData > 0x7FFF){
        // Since "0"s are stored in the upper 2 bytes,
        // calculates OR with FFFF0000[Hex] to convert the device value into a LONG type negative value.
        ICheckData = IData | 0xFFFF0000;
    }else{
        // If the device value whose condition was established is positive,
        // assigns the value as-is to ICheckData.
        ICheckData = IData;
    }
}
```

When using Visual C#

The following is a programming example to monitor D0 for -10, D1 for 0, and D2 for 10 using Visual C#.

```
String szDevice;           // Checked device list
int[] iInputData = new int[3]; // Set value
int[] iEntryData = new int[3]; // Value set to argument of EntryDeviceStatus
int iReturnCode;         // Returned value to EntryDeviceStatus
int iCount;              // Loop counter
// Sets D0, D1 and D2 to the checked device list.
szDevice = "D0\nD1\nD2";
// Sets the checked device value "-10" for D0.
iInputData[0] = -10;
// Sets the checked device value "0" for D1.
iInputData[1] = 0;
// Sets the checked device value "10" for D2.
iInputData[2] = 10;

// If the set value is negative, stores "0"s into the upper 2 bytes
// for conversion into the value to be set to EntryDeviceStatus.
// Loops through the number of device points.
for(iCount = 0; iCount <= 2; iCount++) {
    // If the set value is negative
    if (iInputData[iCount] < 0) {
        // Masks with 0x0000FFFF to store "0"s into upper 2 bytes.
        iEntryData[iCount] = iInputData[iCount] & 0x0000FFFF;
    }
    // If the set value is positive
    }else{
        // Assigns the value as-is to iEntryData.
        iEntryData[iCount] = iInputData[iCount];
    }
}
axActUtilType1.ActLogicalStationNumber = 0;
iReturnCode = axActUtilType1.Open();
// Executes EntryDeviceStatus.
iReturnCode = axActUtilType1.EntryDeviceStatus(szDevice, 3, 5, ref iEntryData[0]);
```

Appendix 10 Time-Out Periods

In MX Component, a time-out may occur at the period different from the value set to the ActTimeOut property in the Act control.

This section explains the time-out periods in various status.

Communication retries at time-out error occurrence

If a time-out error occurs during communication, the time-out processing may be repeated for a maximum of three times in the Act control.

At that time, the process takes a maximum of three times longer period than the set time-out value which is a period until the time-out occurrence.

Communication retries at receive data error occurrence

If a receive data error occurs during communication, send/receive retry processing may be repeated for a maximum of three times in the Act control.

At that time, the process takes a maximum of three times longer period than the set time-out value which is a period until the normal or abnormal termination of the function.

Time-out errors at fixed time in the Act control

MX Component performs communication to check whether a personal computer and a programmable controller system are connected normally before executing the Open function.

When performing the above communication, the fixed time-out period (1000 ms to 4500 ms) in the Act control is used.

Note that if an error occurs during the above communication, an error other than the time-out error may occur.

Appendix 11 Troubleshooting

This section explains the errors which may occur when using MX Component and the troubleshooting.

When self-registration is not performed at the time of executing an installer

Some policies of [User Account Control] in the local security policy may not be installed properly.

Check the following policy settings.

■Detect application installations and prompt for elevation

Select [Enabled].

If selecting [Disabled], a self-registration error may occur during installation.

■Only elevate executables that are signed and validated

Select [Disabled].

If selecting [Enabled], an unsigned execution file cannot be executed and an installer cannot start properly.

■Behavior of the elevation prompt for standard users

Select [Prompt for credentials on the secure desktop] or [Prompt for credentials].

If selecting [Automatically deny elevation requests], elevation fails and an installer cannot start properly.

For CPU modules

■When performing USB communication

A communication error may occur and it may not be recovered if connecting and disconnecting a USB cable, resetting a CPU module, or turning the power ON or OFF is performed frequently during communication with the CPU module.

If it is not recovered, completely disconnect the USB cable and then reconnect it after 5 or more seconds.

(If this error occurs at the initial communication after the above operation, the function will be performed properly in and after the second communication.)

■Clock data of a CPU module

- For QCPUs (Q mode), LCPUs, and FXCPUs, the clock data setting can be set even if the CPU module is in the RUN status.
- For QCPU (Q mode) and LCPU, the setting can be set regardless of the ON/OFF status of the time setting device "SM1028."
- The clock data can be set on FXCPUs with the built-in clock.
- Note that an error for transfer time occurs in the time setting.

For when using Ethernet modules

■Resetting a CPU module during TCP/IP connection establishment

When resetting a CPU module during TCP/IP connection establishment (during opening) by using MX Component, a communication error or receive error occurs at subsequent communication.

In this case, perform the close processing in the application that uses MX Component, and perform the open processing again.

■Target existence check starting interval of an Ethernet module

If the close processing (Close) is executed from a personal computer, an Ethernet module may not perform the close processing (Close). One of its causes is the cable disconnection.

If the open processing (Open) is executed from a personal computer with the Ethernet module not executing the close processing (Close), the open processing (Open) from the personal computer is not terminated normally until the Ethernet module performs a target existence check and executes the close processing (Close).

When terminating the open processing (Open) early from the personal computer, shorten the target existence check starting interval setting of the Ethernet module.

(The default setting of target existence check starting interval of the Ethernet module is 10 minutes.)

■Replacement of Ethernet modules

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

(Because the Ethernet addresses (MAC addresses) differ between devices.)

■Simultaneous access when using a Q series-compatible Ethernet module

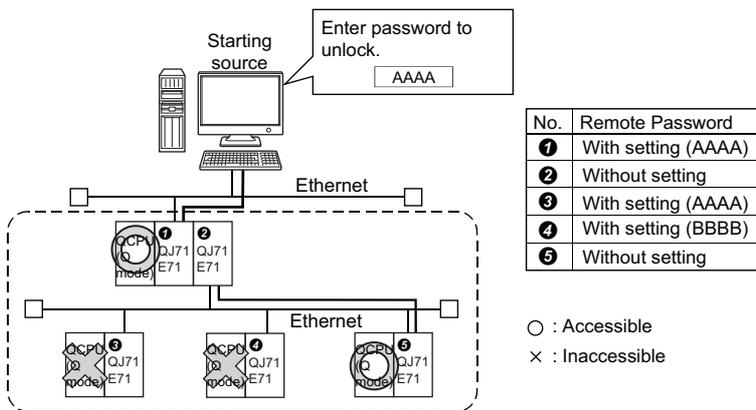
The following conditions should be satisfied when communication is performed simultaneously from multiple personal computers to the same module using the TCP/IP protocol.

- Using Q series-compatible E71 module (except QJ71E71-100) whose first five digits of the serial number are "02122" or higher and whose function version is B or later.
- Using GX Works2, set "MELSOFT connection" in an Ethernet parameter [open system].

■Unlocking password when using QJ71E71

The range where the password can be unlocked by remote operation is up to the connection target station.

If the password is also set on the lower layer, communication cannot be performed with a CPU module on the lower layer.



Unlocking ①QJ71E71 password enables access to CPU modules in this range.

■ Ethernet communication

- The communication line is disconnected if a CPU becomes faulty or an Ethernet module is reset during Ethernet communication (when the protocol is TCP/IP).

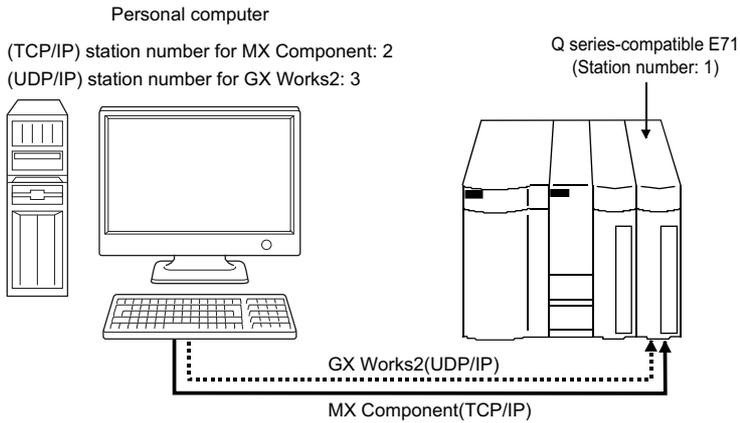
In this case, perform the line close processing (Close) and then perform the reopen processing (Open).

- When two different communication systems (protocols) are used to access from one personal computer to one Q series-compatible E71, two station numbers TCP/IP and for UDP/IP must be set.

However, it is not required to set different station numbers for TCP/IP and for UDP/IP when using MX Component Version 3 or later and Q series-compatible E71 with serial number "05051" or higher.

Ex.

When MX Component uses TCP/IP and GX Works2 uses UDP/IP



Set different station numbers as the (TCP/IP) station number for MX Component and (UDP/IP) station number for GX Works2. If they are set to the same station number, an error will occur on the Ethernet module side.

For communication

■When the socket object generation error (0x01808007) occurs during communication

No.	Condition	Corrective action
1	When using GT SoftGOT	Right-click and select [Run as administrator] when starting an application.
2	When the error cannot be solved by the above No.1	<ul style="list-style-type: none"> • Set a port number different from one used for another application to a personal computer side. • Check whether the first character of the specified IP address is not '0.' When setting the first character to '0,' the value of the target octet is processed as octal number.
3	When the error cannot be solved by the above No.1 and No. 2	Check the following items, and contact your local Mitsubishi Electric representative. <ul style="list-style-type: none"> • System configuration (model name of the target programmable controller, model name of the module, and used network) • Name of personal computer (Manufacturer), CPU, memory size • OS, Edition, 32-bit/64-bit • MX Component version • A function and argument of MX Component in which the symptom occurs • Settings within "Communication Settings Utility" • Occurrence frequency and procedure for a symptom

For saving the setting of Communication Settings Utility

■Errors which occur when saving the setting with Communication Settings Utility and corrective actions

Error message	Corrective action	Considerations
Updating the system information data failed. <ul style="list-style-type: none"> • There is a possibility without the operation authority. Please execute it by the administrator authority user. • There is a possibility that the operating environment of the application is corrupt. Re-install the application and try again. 	Select "Communication Settings Utility" from "MELSOFT" in Windows Start, right-click, and select [Run as administrator] to start up the application.	When Label Utility is executed, the status is switched to the administrator authority automatically. The performance differs according to the user account control (UAC) settings.

When a header file is not created properly at the time of pasting a control to a form with Visual C++

A

A header file may not be created properly when pasting a control to a form with Visual C++ depending on the version of Visual Studio 2019.

In this case, overwrite the created file with the following header files stored in "[User-specified folder]\Act\Include."

Control to be used	Header file
ActUtilType	actutiltype.h
ActUtilDataLogging	actutildatalogging.h
ActSupportMsg	actsupportmsg.h
ActProgType	actprogtype.h
ActProgDataLogging	actprogdatalogging.h

When an error occurs in the setting for using controls in Visual Studio .NET

An error may occur in the setting for using controls.

For the corrective actions, refer to the following 'Point'.

Point

When creating an application using .NET Framework 4, .NET Framework 4.5, or .NET Framework 4.6 (The following is an example for .NET Framework 4.) Add an application configuration file with the following elements (app.config) to the folder which contains the .exe file of the application.

For details, refer to Visual Studio Help.

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

When pasting a control to a form

Point

- When using Visual C++ .NET/Visual Basic .NET/Visual C# projects

The following library is added to "References":

For ActProgType: ActProgTypeLib

For ActUtilType: ActUtilTypeLib

For ActSupportMsg: ActSupportMsgLib

For DotUtilType: MITSUBISHI.Component.DotUtilType

For DotSupportMsg: MITSUBISHI.Component.DotSupportMsg

Set "False" for "Embed Interop Types" in the property of each library.

When a link error occurs in creating a user program in Visual C++

A link error occurs if both '.NET control' (DotUtilType) and 'Act control' (ActUtilType) are used at the same time by pasting them in a single program.

The corrective actions are shown below.

Procedure for adding an application configuration file

1. Select [Project] ⇒ [Add New Item].
2. Select [Installed] ⇒ [Visual C++] ⇒ [Utility].
3. Select "Configuration file (app.config)" and click the [Add] button.
4. Write the following code to the application configuration file (app.config).

```
<?xml version="1.0" encoding="utf-8" ?>
<configuration>
  <startup useLegacyV2RuntimeActivationPolicy="true">
    <supportedRuntime version="v4.0"/>
  </startup>
</configuration>
```

5. Select [Project] ⇒ [Properties].
6. Select [Configuration Properties] ⇒ [Build Events] ⇒ [Post-Build Event].
7. Set "copy app.config"\$(TargetPath).config"" for "Command Line."

Procedure for pasting a control

For the corrective actions, refer to the following 'Point'.

Point

When using both .NET control (DotUtilType) and Act control (ActUtilType) by pasting them into a program, perform the following operations by selecting [Project] ⇒ [Properties] from Solution Explorer.

- ① Select [Common Properties] ⇒ [Framework and references].
- ② Select "ActUtilTypeLib" from "Name" in "References," and change the setting of "Reference Assembly Output" under "Build Properties" to "False."
- ③ Select "Interop.ActUtilTypeLib.1.0" from "Name" in "References," and change the setting of "Reference Assembly Output" under "Build Properties" to "False."
- ④ Click the [OK] button to reflect the change of the setting.

A

When an unstart error occurs during communication with GX Simulator3

If GX Simulator3 communication is performed by importing the setting of MX Component Version 4.15R or earlier, GX Simulator3 unstart error (0xF1000020) occurs.

The corrective actions are shown below.

When using a utility setting type control

Calculate the port number of the simulator using the average value '5500', and set the value using Communication Settings Utility again.

- $\text{PortNumber} = 5500 + \text{System No.} \times 10 + \text{Unit No.}$ (For System No.=1 and Unit No.=1, $5511 = 5500 + 1 \times 10 + 1$)

When using a program setting type control

Calculate the port number of the simulator using the average value '5500', and change the setting to pass the value to the property of the control.

- $\text{PortNumber} = 5500 + \text{System No.} \times 10 + \text{Unit No.}$ (For System No.=1 and Unit No.=1, $5511 = 5500 + 1 \times 10 + 1$)

Appendix 12 Differences with previous version of MX Component

This section explains the differences between MX Component Version 4 and Version 5.

For the functions added or changed in version 5.000A and later, refer to the following:

☞ Page 646 Changes from Previous Version

Comparison with MX Component Version 4

Development environment

The following table shows the differences of the development environment between MX Component Version 4 and Version 5.

Item		MX Component Version 4	MX Component Version 5	
OS	Windows	Microsoft Windows XP Microsoft Windows Vista Microsoft Windows 7 Microsoft Windows 8 Microsoft Windows 8.1 Microsoft Windows 10	Microsoft Windows 10 Microsoft Windows 10 IoT Enterprise 2016 LTSC Microsoft Windows 10 IoT Enterprise 2019 LTSC Microsoft Windows 11	
Programming language	Visual Basic.Net	Microsoft Visual Studio 2005 Visual Basic Microsoft Visual Studio 2008 Visual Basic Microsoft Visual Studio 2010 Visual Basic Microsoft Visual Studio 2012 Visual Basic Microsoft Visual Studio 2013 Visual Basic Microsoft Visual Studio 2015 Visual Basic Microsoft Visual Studio 2017 Visual Basic	Microsoft Visual Studio 2019 Visual Basic Microsoft Visual Studio 2022 Visual Basic	
	Visual C++	Microsoft Visual Studio 2005 Visual C++ Microsoft Visual Studio 2008 Visual C++ Microsoft Visual Studio 2010 Visual C++ Microsoft Visual Studio 2012 Visual C++ Microsoft Visual Studio 2013 Visual C++ Microsoft Visual Studio 2015 Visual C++ Microsoft Visual Studio 2017 Visual C++	Microsoft Visual Studio 2019 Visual C++ Microsoft Visual Studio 2022 Visual C++	
	Visual C#	Microsoft Visual Studio 2005 Visual C# Microsoft Visual Studio 2008 Visual C# Microsoft Visual Studio 2010 Visual C# Microsoft Visual Studio 2012 Visual C# Microsoft Visual Studio 2013 Visual C# Microsoft Visual Studio 2015 Visual C# Microsoft Visual Studio 2017 Visual C#	Microsoft Visual Studio 2019 Visual C# Microsoft Visual Studio 2022 Visual C#	
	VBA		Microsoft Excel 2003 Microsoft Excel 2007 Microsoft Excel 2010 (32-bit version) Microsoft Excel 2013 (32-bit version) Microsoft Excel 2016 (32-bit version) Microsoft Excel 2019 (32-bit version)	Microsoft Excel 2019 (32-bit version/64-bit version) Microsoft Excel 2021 (32-bit version/64-bit version) Microsoft Excel for Microsoft 365 (64-bit version)
			Microsoft Access 2003 Microsoft Access 2007 Microsoft Access 2010 (32-bit version) Microsoft Access 2013 (32-bit version) Microsoft Access 2016 (32-bit version) Microsoft Access 2019 (32-bit version)	Microsoft Access 2019 (32-bit version/64-bit version) Microsoft Access 2021 (32-bit version/64-bit version) Microsoft Access for Microsoft 365 (64-bit version)
	VBScript	Text editor and commercially available HTML tool	Not supported	

A

Communication route

The following table shows the differences of the communication routes between MX Component Version 4 and Version 5.

Point

Whether the communication can be established or not differs depending on the operating system to be used. In addition, restrictions on versions may apply even for the applicable boards and drivers.

For details, refer to the following:

-  Page 26 Supported Communication Routes
-  Page 31 Module list

○: Available, ×: Not available

Communication route	MX Component Version 4 ^{*1}	MX Component Version 5
Serial communication	○	○
Ethernet communication	○	○
CPU COM communication	○	○
CPU USB communication	○	○
MELSECNET/H communication	○	○ ^{*2*3}
CC-Link IE Controller Network communication	○	○ ^{*2*3}
CC-Link IE Field Network communication	○	○ ^{*2*3}
CC-Link IE TSN communication	×	○ ^{*2*3}
CC-Link communication	○	○ ^{*2*3}
CC-Link system RS-232 interface communication	○	○
Q series bus communication (only when PC CPU module is used)	○	×
GX Simulator communication	○	×
GX Simulator2 communication	○	○
GX Simulator3 communication	○	○
MT Simulator2 communication	○	○
Modem communication	○	×
GOT gateway device communication	○	○
GOT transparent communication	○	○
Inverter COM communication	○	○
Inverter USB communication	○	○
Robot controller COM communication	○	○ ^{*3}
Robot controller USB communication	○	○ ^{*3}
Robot controller Ethernet communication	○	○ ^{*3}

*1 Whether the communication can be established or not differs depending on the operating system to be used. ( MX Component Version 4 Operating Manual)

*2 Not supported when using Windows 10 Education.

*3 64-bit version is not supported when using Windows 10 IoT Enterprise 2016 LTSC.

Applicable models

The following table shows the differences of the applicable models between MX Component Version 4 and Version 5.

○: Applicable, ×: Not applicable

Category	Series	MX Component Version 4	MX Component Version 5
CPU	RCPU	○	○
	RCCPU	○	○
	R motion CPU	○	○
	LHCPU	○	○
	FX5CPU	○	○
	QCPU (Q mode)	○	○ ^{*1}
	LCPU	○	○
	QCCPU	○	○
	QSCPU	○	○
	Q motion CPU	○	○ ^{*2}
	FXCPU (FX3CPU) ^{*3}	○	○
	FXCPU (FX0CPU to FXUCPU) ^{*4}	○	×
GOT	GOT1000	○	×
	GOT2000	○	○
Inverter	A800	○	○
Robot controller	CR750	○	○
	CRnD-700	○	○
Simulator	GX Simulator	○	×
	GX Simulator2	○	○
	GX Simulator3	○	○
	MT Simulator2	○	○

*1 Basic model QCPUs, high performance model QCPUs, and process CPUs are not supported.

*2 Q172CPU, Q173CPU, Q172HCPU, and Q173HCPU are not supported.

*3 FX3SCPU, FX3GCPU, FX3GCCPU, FX3UCPU, FX3UCCPU

*4 FX0CPU, FX0SCPU, FX0NCP, FX1CPU, FX1NCP, FX1NCCPU, FX1SCPU, FXUCPU, FX2CCPU, FX2NCP, FX2NCCPU

Controls

The following table shows differences of controls.

MX Component Version 4	MX Component Version 5	
	32-bit	64-bit ^{*2}
ActUtilType	ActUtilType	ActUtilType64
ActProgType	ActProgType	ActProgType64
ActSupportMsg	ActSupportMsg	ActSupportMsg64
ActUtilDataLogging	ActUtilDataLogging	ActUtilDataLogging64
ActProgDataLogging	ActProgDataLogging	ActProgDataLogging64
DotUtilType	DotUtilType	DotUtilType64
DotSupportMsg ^{*1}	ActSupportMsg	ActSupportMsg64
ActMLUtilType	Not supported because VBScript is not supported.	
ActMLProgType		
ActMLSupportMsg		

*1 Controls of the DotSupportMsg are integrated into the ActSupportMsg in Version 5.000A.

*2 The method to paste this control to a form cannot be used.

A

Compatibility

The programs and communication settings of MX Component Version 4 can be utilized in MX Component Version 5.

Point

For the programs and communication settings which are not supported in MX Component Version 5, use MX Component Version 4.

Note that MX Component Version 4 and MX Component Version 5 cannot be installed on the same personal computer.

Install MX Component Version 4 after uninstalling MX Component Version 5.

Utilizing programs

Controls of MX Component Version 4 are included in MX Component Version 5.

Therefore, programs created with the controls of MX Component Version 4 can be used in MX Component Version 5.

Utilizing communication settings

Import the setting file (.ACT file) of MX Component Version 4 using Communication Settings Utility of MX Component Version 5. (☞ Page 62 Importing)

The settings can be used for controls of both MX Component Version 4 and MX Component Version 5.

■Exporting/importing

The communication settings and system labels of MX Component Version 4 can be utilized in MX Component Version 5 by using exporting/importing of communication settings.

The following is a correspondence table for the communication setting and system label.

Exporting	Importing	
	MX Component Version 4	MX Component Version 5
MX Component Version 4	<input type="radio"/>	<input type="radio"/>
MX Component Version 5	<input type="checkbox"/>	<input type="radio"/>

Using a control

■When using control in 32-bit

When using the DotSupportMsg control of MX Component Version 4 in MX Component Version 5, replace it with the ActSupportMsg control.

Controls of MX Component Version 4 other than the above can be used as it is in MX Component Version 5.

■When using control in 64-bit

- When using a control of MX Component Version 4 by creating an object

A control can be used in MX Component Version 5 by replacing it with a control for 64-bit version.

Ex.

DotSupportMsg → ActSupportMsg64

- When using a control of MX Component Version 4 by pasting it to a form

When using a control in VBA, the control can be used in MX Component Version 5 by replacing it with a control for 64-bit version.

When using the control in a language other than VBA, the control cannot be used by pasting to a form in MX Component Version 5.

Use the control by creating an object. (☞ Page 46 When using a control by creating an object, Page 49 When using a control by creating an object)

MEMO

A

FUNCTION INDEX

C

Close(Closing communication line) . . . 424,496,505

D

Dispose(release memory) 602

E

EntryDeviceStatus(Registering devices for status monitoring) 465,558

F

FreeDeviceStatus(Deregistering devices for status monitoring) 470,562

G

GetClockData(Reading clock data) 452,552
GetCpuType(Reading CPU module model) . 460,556
GetDevice(Acquiring device data) 441,544
GetDevice2(Acquiring device data) 490,599
GetErrorMessage(Acquiring error description and corrective action) 492
GetFile(Transferring logging files) 502

O

OnDeviceStatus(Event notification) 471,563
Open(Opening communication line) . . 422,494,504

R

ReadBuffer(Reading data from buffer memory) 444,548
ReadClose(Ending the search) 501
ReadDeviceBlock(Reading devices in batch) 425,506
ReadDeviceBlock2(Reading devices in batch) 474,565
ReadDeviceRandom(Reading devices randomly) . . . 432,518
ReadDeviceRandom2(Reading devices randomly) . . 480,575
ReadFirstFile(Searching for a file/directory) 497
ReadNextFile(Searching for a file/directory) 499

S

SetClockData(Writing clock data) 456,554
SetCpuStatus(Remote control) 462,557
SetDevice(Set device data) 439,540
SetDevice2(Set device data) 487,595

W

WriteBuffer(Writing data to buffer memory) . 448,550
WriteDeviceBlock(Writing devices in batch) . 428,512
WriteDeviceBlock2(Writing devices in batch) 477,570
WriteDeviceRandom(Writing devices randomly) . . . 436,529

WriteDeviceRandom2(Writing devices randomly) . . . 483,584

MEMO

REVISIONS

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

Revision date	*Manual number	Description
April 2021	SH(NA)-082395ENG-A	First edition
November 2021	SH(NA)-082395ENG-B	■Added or modified parts GENERIC TERMS AND ABBREVIATIONS, Section 2.1, Section 2.4, Section 5.1, Section 9.1, Section 9.2, Section 10.2, Section 10.14, Section 11.2, Chapter 12, Section 13.1, Appendix 1, Appendix 12, Appendix 13, TRADEMARKS
March 2022	SH(NA)-082395ENG-C	■Added or modified parts Section 2.1, Appendix 1, Appendix 13
October 2022	SH(NA)-082395ENG-D	■Added or modified parts TERMS, GENERIC TERMS AND ABBREVIATIONS, Section 2.1, Section 2.4, Section 9.1, Section 9.2, Section 10.2, Section 10.3, Section 10.4, Section 10.14, Section 11.2, Section 11.5, Chapter 12, Chapter 13, Appendix 1, Appendix 13
May 2023	SH(NA)-082395ENG-E	■Added or modified parts Section 2.1, Section 2.4, Section 6.1, Section 8.1, Section 9.2, Appendix 1, Appendix 13
October 2023	SH(NA)-082395ENG-F	■Added or modified parts GENERIC TERMS AND ABBREVIATIONS, Section 2.2, Section 2.3, Section 2.4, Section 2.5, Chapter 4, Section 5.1, Section 6.16, Section 8.1, Section 8.2, Section 9.2, Section 10.8, Section 11.2, Section 11.3, Section 11.5, Section 13.1, Appendix 1, Appendix 12
April 2024	SH(NA)-082395ENG-G	■Added or modified parts Section 10.2, Section 10.6, Section 13.1, Section 13.4, Appendix 1, Appendix 5

Japanese manual number: SH-082394-G

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.

© 2021 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 to:

- (1) Damages caused by any cause found not to be the responsibility of Mitsubishi.
- (2) Loss in opportunity, lost profits incurred to the user by Failures of Mitsubishi products.
- (3) Special damages and secondary damages whether foreseeable or not, compensation for accidents, and compensation for damages to products other than Mitsubishi products.
- (4) Replacement by the user, maintenance of on-site equipment, start-up test run and other tasks.

5. Changes in product specifications

The specifications given in the catalogs, manuals or technical documents are subject to change without prior notice.

INFORMATION AND SERVICES

For further information and services, please contact your local Mitsubishi Electric sales office or representative.
Visit our website to find our locations worldwide.

MITSUBISHI ELECTRIC Factory Automation Global Website
Locations Worldwide
www.MitsubishiElectric.com/fa/about-us/overseas/

TRADEMARKS

Intel is either a registered trademark or a trademark of Intel Corporation in the United States and/or other countries.

Java is a registered trademark of Oracle and/or its affiliates.

Microsoft, Microsoft Access, Excel, Visual Basic .NET, Visual C++, Visual C#, Visual Studio, Windows, Windows Vista, and Windows XP are trademarks of the Microsoft group of companies.

The company names, system names and product names mentioned in this manual are either registered trademarks or trademarks of their respective companies.

In some cases, trademark symbols such as [™] or [®] are not specified in this manual.

COPYRIGHTS

SPREAD

Copyright © 2004 FarPoint Technologies, Inc.

VSFlexGrid8 Pro

Copyright © 2008 ComponentOne LLC.

SH(NA)-082395ENG-G(2404)

MODEL: MXCV5-R-E

MITSUBISHI ELECTRIC CORPORATION

HEAD OFFICE: TOKYO BLDG., 2-7-3, MARUNOUCHI, CHIYODA-KU, TOKYO 100-8310, JAPAN
NAGOYA WORKS: 1-14, YADA-MINAMI 5-CHOME, HIGASHI-KU, NAGOYA 461-8670, JAPAN

When exported from Japan, this manual does not require application to the
Ministry of Economy, Trade and Industry for service transaction permission.

Specifications subject to change without notice.