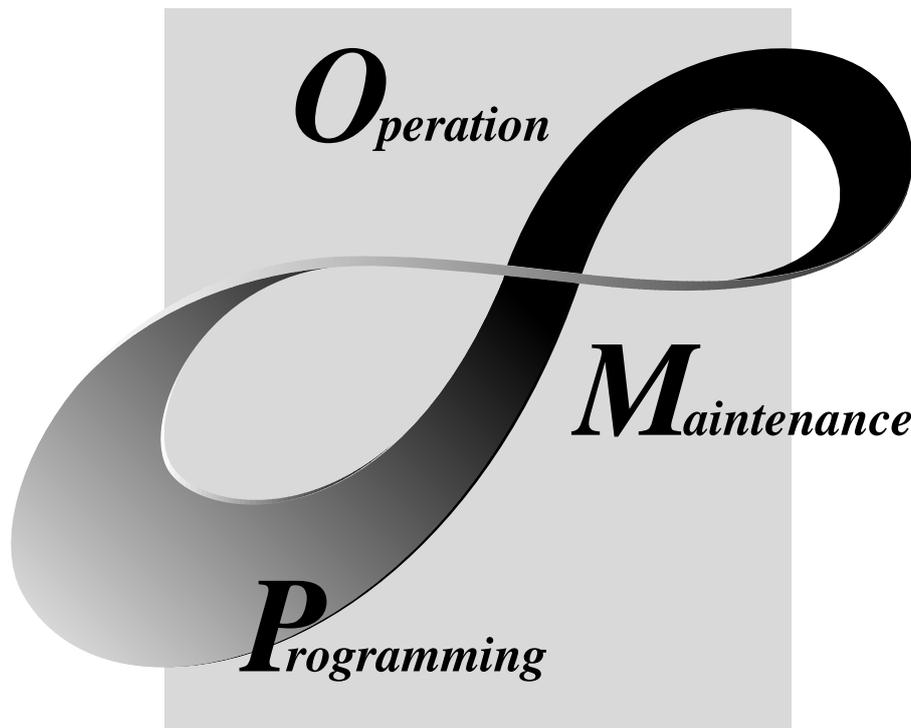


MX Component Version 2

Programming Manual

mitsubishi



MELSOFT
Integrated FA Software

SW2D5C-ACT-E

• SAFETY PRECAUTIONS •

(Always read these instructions before using this equipment.)

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

The instructions given in this manual are concerned with this product. For the safety instructions of the programmable controller system, please read the CPU module user's manual.

In this manual, the safety instructions are ranked as "DANGER" and "CAUTION".



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



Indicates that incorrect handling may cause hazardous conditions, resulting in medium or slight personal injury or physical damage.

Note that the  CAUTION level may lead to a serious consequence according to the circumstances. Always follow the instructions of both levels because they are important to personal safety.

Please save this manual to make it accessible when required and always forward it to the end user.

[Design Instructions]

DANGER

- When performing data changes or status control from the personal computer to the running PLC, configure up an interlock circuit outside the PLC system to ensure that the whole system will operate safely.

In addition, predetermine corrective actions for the system so that you can take measures against any communication error caused by a cable connection fault or the like in online operations performed from the peripheral device to the PLC.

CAUTION

- Read the manual carefully before performing the online operations (especially forced output and operating status change) which will be executed with the personal computer connected to the running CPU module.

Not doing so can damage the machine or cause an accident due to misoperation.

REVISIONS

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

| Print Date | * Manual Number | Revision |
|------------|------------------|---------------|
| Mar., 2001 | SH (NA)-080155-A | First edition |
| | | |

Japanese Manual Version SH-080152-A

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

© 2001 MITSUBISHI ELECTRIC CORPORATION

Operating Instructions

- (1) When using Microsoft® Windows NT® Workstation Operating System Version 4.0 and Microsoft® Windows® 2000 Professional Operating System
When using Windows NT® Workstation 4.0 and Windows® 2000 Professional, MX Component may be installed and used only on the administrator's authority.
- (2) About Ethernet communication, computer link communication and CPU COM communication on Microsoft® Windows® 95 Operating System
 - (a) Making Ethernet communication using TCP/IP and UDP/IP on Windows® 95 of the version older than OSR2 will cause a memory leak. When performing continuous operation on Windows® 95, use Window® 95 OSR2 or later.
 - (b) On Windows® 95, communication using the COM port, e.g. computer link communication or CPU COM communication, will cause a memory leak. Therefore, do not perform continuous operation.
- (3) About installation
 - (a) When performing overwrite installation, install the software in the folder where it had already been installed.
 - (b) If you install the MELSEC board driver or GX Developer into the personal computer where MX Component has already been installed, communication using a specific path (e.g. ASCII packet of the AJ71E71) may result in a receive, device number or other error.
If any of these phenomena has occurred, perform overwrite installation of MX Component again.
- (4) Precautions for performing installation and uninstallation on a dual boot machine where two different operating systems are installed in a single IBM-PC/AT compatible personal computer

On a dual boot machine having Windows NT® Workstation 4.0 (hereafter referred to as OS1) and Windows® 95/98 (hereafter referred to as OS2), note the following points when MX Component was installed on OS1 first and MX Component was then installed over the same folder on OS2.

 - (a) If MX Component is uninstalled first on the OS2 side, uninstallation does not delete the control DLLs and ACT folders, and they remain within the IBM-PC/AT compatible.
To delete the control DLLs and ACT folders, perform uninstallation also on the OS1 side.
 - (b) If MX Component is uninstalled first on the OS1 side, the control DLLs and ACT folders are deleted.
In this case, MX Component may not operate properly or cannot be uninstalled on the OS2 side.
Install MX Component again on the OS2 side to operate MX Component properly or uninstall it on the OS2 side.
- (5) About start menu
When you have uninstalled MX Component, the item may remain in the start menu.
In that case, restart the IBM-PC/AT compatible personal computer.

- (6) About the resume and other functions of personal computer
 A communications error may occur if communications are made with the PLC CPU after setting the resume function, suspend setting, power-saving function and/or standby mode of the personal computer.
 Therefore, do not set the above functions when making communications with the PLC CPU.
- (7) About transmission speed
 As the transmission speed of the QCPU(Q mode) and QCPU(A mode), you can set 9600bps, 19200bps, 38400bps, 57600bps or 115200bps.
 For the QnACPU of version 9707B or later, you can set the transmission speed of 9600bps, 19200bps or 38400bps.
 For the QnACPU of other versions, you can set 9600bps or 19200bps.
 The transmission speeds of the ACPUs (except A2USHCPU-S1), FXCPU and motion controller CPU are fixed to 9600bps. (The A2USHCPU-S1 may be set to 19200bps.)
- (8) About use of the Q4ARCPU
 (a) When using the UDP/IP protocol of Ethernet communication, use the Q4ARCPU whose year and month of manufacture is "0012" or later and the QE71 whose function version is B or later.
 (b) The duplexing function cannot be used.
- (9) Restrictions on use of the FXCPU
 (a) For the index registers (Z, V) of the FXCPU, data cannot be written to 2 or more consecutive points using WriteDeviceBlock(). (Data may be written to only one point.)
 (b) When the FXCPU is used, access to the TN devices (timer present values) or CN devices (counter present values) is not permitted if the device numbers specified are split across 199 or earlier and 200 or later.
- (10) About clock data of the PLC CPU
 (a) For the ACPUs (including the motion controller CPU), clock data setting may be made only when the PLC CPU is in the STOP status.
 For the QCPU (Q mode), QCPU (A mode), QnACPU and FXCPU, clock data setting may be made if the PLC CPU is in the RUN status.
 (b) For the A0J2HCPU, A2CCPU and A2CJCPU, setting cannot be made as they do not have the clock function.
 (c) For the ACPUs, setting can be made independently of whether the clock setting special relay "M9028" is ON or OFF. (Note that the special relay "M9028" turns OFF after execution.)
 For the QCPU (Q mode), QCPU (A mode) and QnACPU, setting can be made independently of whether the clock setting device "SM1028" is ON or OFF.
 (d) Among the FXCPUs, setting may be made for only the FX1N (clock built-in), FX1NC (clock built-in), FX1S (clock built-in), FX2N (clock built-in), FX2NC (clock built-in), FX2 (when RTC cassette is fitted) and FX2C (when RTC cassette is fitted).
 (e) Note that an error for transfer time will be produced in clock setting.

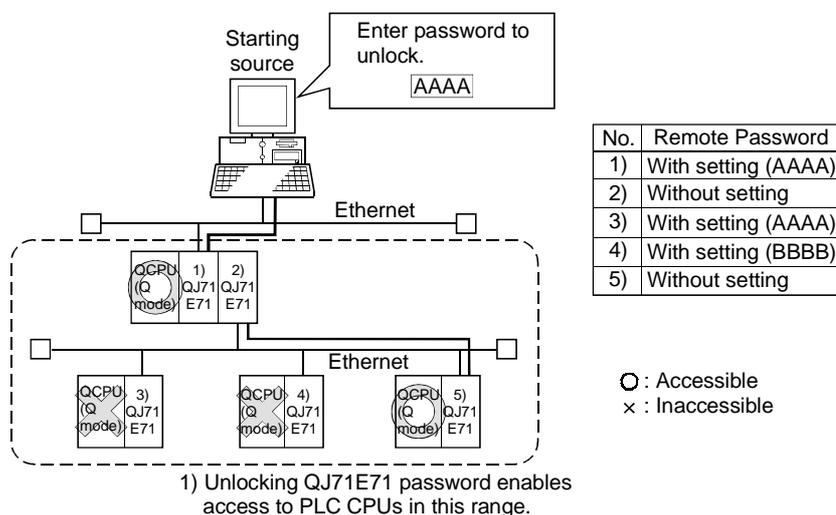
- (11) About simultaneous use of MX Component and GX Developer
When using GX Developer and MX Component together for the same E71 module to make Ethernet communication, make the following settings.
(a) Set the protocol of the communication setting wizard screen to "UDP/IP".
(b) Set "SW2" of the communications setting switches of the E71 module to OFF (binary).
- (12) Simultaneous access when using Q series-compatible Ethernet module
The following conditions should be satisfied when communication is to be made simultaneously from multiple IBM-PC/AT compatibles to the same module using the TCP/IP protocol.
- The Q series-compatible Ethernet module is of function version B or later.
 - Using GX Developer Version 6.05F or later, set "MELSOFT connection" in the Ethernet parameter "open system".
- (13) About target existence check starting interval*1 of Ethernet module
If close processing (Close) is executed from the IBM-PC/AT compatible, the Ethernet module may not perform close processing (Close).
One of its causes is the open cable.
If open processing (Open) is executed from the IBM-PC/AT compatible with the Ethernet module not performing close processing (Close), open processing (Open) from the IBM-PC/AT compatible is not terminated normally until the Ethernet module makes a target existence check and executes close processing (Close).
If you want to terminate open processing (Open) early from the IBM-PC/AT compatible, shorten the target existence check starting interval setting of the Ethernet module.
(The target existence check starting interval setting of the Ethernet module defaults to 10 minutes.)
*1: It can be set for the E71 of AJ71E71-S3 or later.
- (14) Replacement of Ethernet module
If you changed the Ethernet module during Ethernet communication due to debugging, failure or like, the other node (IBM-PC/AT compatible) must be restarted.
(Since the Ethernet addresses (MAC addresses) differ between devices)
- (15) Software version of CC-Link master/local module
As the CC-Link master/local module used in CC-Link communication or CC-Link G4 communication(only when the AJ65BT-G4 is used), use the module of software version "N" or later.
The module of software version "M" or earlier will not operate properly.
- (16) Software version of CC-Link G4 module
As the CC-Link G4 module used in CC-Link G4 communication(only when the AJ65BT-G4 is used), use the module of software version "D" or later.
The module of software version "C" or earlier will not operate properly.
- (17) About relaying from the MELSECNET/10 loaded station
When the module is loaded to the AnNCPUs or AnACPU, it is recognized as a MELSECNET(II) module.
When the connected station is the AnNCPUs or AnACPU, set the relayed network as MELSECNET(II).
In addition, set the station number to "0" when making access to the control station.

- (18) About computer link communication
- (a) If the connected station CPU is the AnUCPU and the computer link module is the UC24 for computer link connection, remote operation will result in an error when access is made to the AnNCPU, AnACPU or QnACPU via the MELSECNET/10.
 - (b) On any computer link modules other than the UC24 and C24, remote "PAUSE" operation will result in an error for all connections.
 - (c) For the QC24, note that the illegal case of specifying the first I/O number of a nonexisting module and reading/writing U**\G** will not return an error if the software version of the module is "k" or earlier.
 - (d) In any connection form (direct coupling, relaying) where the target station of the UC24 or C24 is the QnACPU, an error is returned if clock data read/write is executed.
- (19) Precautions for USB communication
- Frequently disconnecting/reconnecting the USB cable or resetting or powering ON/OFF the PLC CPU during communications with the PLC CPU may cause a communications error which cannot be recovered.
- If it is not recovered, completely disconnect the USB cable once and then reconnect it after 5 or more seconds have elapsed.
- (If this error occurs at the initial communication after the above operation, the function will be performed properly in and after the second communications.)
- (20) Precautions for GX Simulator communication
- Before executing the monitor utility, communication setting utility or user program, make sure that GX Simulator and GX Developer are operating.
- In addition, do not terminate the GX Simulator and GX Developer while the user program is running.
- If you do so, you will not be able to terminate the user program normally.
- (21) About forced termination of processes during communication
- If communication is being made 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.
- (22) About sample programs, test programs and sample sequence programs
- (a) Sample programs, test programs
 - The sample programs are attached for your reference to create user programs.
 - The test programs are attached to conduct communication tests.
 - Use these programs on your own responsibility.
 - (b) Sample sequence programs
 - The sample sequence programs attached to MX Component assume that only an IBM-PC/AT compatible personal computer and Ethernet modules exist in the network.
 - Depending on your system configuration and parameter settings, the programs must be modified. Make corrections to make the programs optimum for your system.
 - Also, use the sample sequence programs on your own responsibility.

(23) 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 set also on the lower layer, communication cannot be made with the PLC CPU on the lower layer.



(24) Resetting PLC CPU during TCP/IP connection setting

If you reset the PLC CPU during TCP/IP connection setting (during opening) using MX Component, a communication or receive error will occur at the time of communication after that. In that case, close the application that uses MX Component and then perform open processing again.

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

MX Component does not have the Internet/intranet security function. When you need the security function, make setting on the user side.

(26) Precautions for use of Microsoft® Access 2000

- (a) When you paste the ACT control to an Access 2000 form and double-click the ACT control or choose the custom control in the property, the following error message will appear but this does not affect the operation of ACT control.
(Other error message may appear.)

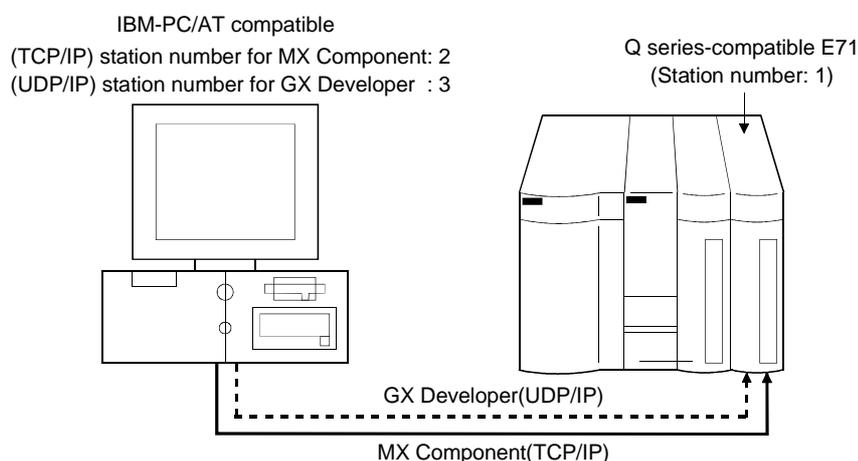


- (b) When you paste the ACT control and display the properties, the property names displayed may be broken. As this phenomenon occurs for only the property indication, there will be no problem in the property functions.

- (27) Precautions for use of Microsoft® Excel 2000
- (a) If you paste the control to Excel 2000, it may sometimes not be pasted. This phenomenon occurs if the cache file (temporary file) of Excel 2000 remains.
In such a case, perform operation in the following procedure.
 - 1) Close Excel 2000.
 - 2) Delete *.exd in the Excel 8.0 folder of the temp folders.
 - 3) Restart Excel 2000.
 - (b) The size of the ACT control can be changed but this does not affect the operation of MX Component.
To restore the size, set the Height and Width properties of ACT control to "24" again.
- (28) Precautions for use of Microsoft® Windows® Millennium Edition Operating System
- It is not recommended to use MX Component with the "system restoring function" made invalid by the operating system.
If the free space of the system drive becomes less than 200MB, the "system restoring function" is made invalid by the operating system. When using Windows® Me, reserve a 200MB or more free space for the system drive.
- (29) About error at communication start
- A communication error may occur within the preset time-out period at a communication start, e.g. when the communication diagnostic button is pressed, at a monitor start, or at the execution of any function.
These errors are assumed to be detected before a time-out error.
(Example: Connection cable not connected, at PLC power-off)
- (30) About Ethernet communication
- (a) When access is made to the QnACPU, AnUCPU, QCPU (A mode) or motion controller CPU via the E71, the device range is equivalent to that of the AnACPU.
 - (b) When making access to the PLC CPU through Ethernet communication, the functions may not be executed depending on the PLC CPU status.
 - 1) When the protocol is TCP/IP (target module: E71, QE71)
The functions can be executed only when the communication target PLC CPU is in the RUN mode.
An error is returned if the PLC CPU is in other than the RUN mode.
 - 2) When the protocol is UDP/IP (target module: E71, QE71)
The functions cannot be executed until the communication target PLC CPU is RUN once.
An error is returned if the PLC CPU has not been RUN once.
 - (c) The communication line is broken if the CPU becomes faulty or the Ethernet module is reset during Ethernet communication (when the protocol is TCP/IP).
In that case, perform line close processing (Close) and then execute reopen processing (Open).

- (d) When two different communication systems (protocols) are used to make access from one IBM-PC/AT compatible to one Q series-compatible E71, two station numbers, i.e. for TCP/IP and for UDP/IP, must be set.

(Example) When MX Component uses TCP/IP and GX Developer uses UDP/IP



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

- (31) About switch settings of E71 and QE71
If the four lower digits of the error code that occurred during Ethernet communication using the E71 or QE71 is not indicated in the E71 or QE71 manual, check the DIP switch (SW2) setting of the E71 or QE71. If the DIP switch is not set correctly, a difference has occurred in the packet format (ASCII/binary) and therefore the error code returned from the module cannot be recognized correctly.
- (32) Instructions for relaying the MELSECNET(II)
When access is made to the QnACPU, AnUCPU, QCPU (A mode) or motion controller CPU via the MELSECNET(II), the device range is equivalent to that of the AnACPU.
- (33) Restrictions on use of the FXCPU
- (a) When the FXCPU is used, access to the TN devices (timer present values) or CN devices (counter present values) is not permitted if the device numbers specified are split across 199 or earlier and 200 or later.
 - (b) As the FXCPU does not have a PAUSE switch as the PLC CPU, an error is returned if remote pause is specified in SetCpuStatus.
 - (c) Note that specifying the first I/O number of a nonexisting module and executing the WriteBuffer() method will not return an error.
 - (d) For the index registers (Z, V) of the FXCPU, data cannot be written to 2 or more consecutive points using WriteDeviceBlock(). (Data may be written to only one point.)

- (34) CheckDeviceString
Do not use the CheckDeviceString method of each ACT control.
- (35) About ActUMsg control, ActUWzd control, ActMnet2BD control and ActAFBD control
Installing MX Component registers the ActUMsg control, ActUWzd control, ActMnet2BD control and ActAFBD control, but do not use them.
- (36) Precautions for use of Act(ML)QJ71E71TCP, Act(ML)AJ71QE71TCP and Act(ML)AJ71E71TCP controls
- (a) Provide an interval longer than the sequence scan time of the Ethernet module loaded station from when the Open method is executed until the Close method is executed.
- (b) Provide an interval of at least 500ms from when the Close method is executed until the Open method is executed again.
- (37) Precautions for use of EXCEL VBA
Do not set the page feed preview function in the application that uses EXCEL VBA.
Doing so can cause a memory leak or OS basic operation (file operation, printing or other) fault.
- (38) Serial communication function of Q00J/Q00/Q01CPU
When the following conditions are all satisfied, communication between the personal computer and the Q00J/Q00/Q01CPU is made at 9600bps speed.
- 1) The connected CPU is the Q00CPU or Q01CPU
 - 2) The serial communication function of the connected CPU is valid.
 - 3) The personal computer side baud rate setting differs from the Q00J/Q00/Q01CPU side baud rate setting.
- To increase the communication speed, match the personal computer side baud rate with the Q00J/Q00/Q01CPU side baud rate.
- (39) Precautions for starting multiple Excel files on Windows® Me
Note that Windows® Me has been confirmed to stop if you run multiple Excel files which use many control objects.
- * This phenomenon is not attributable to this product.
- (a) Conditions on which this phenomenon has been confirmed to occur
- | | |
|--|--|
| Graphic driver | : Matrox make MGA Mystique display driver |
| OS | : Windows® Me (English version) |
| Number of controls pasted to Excel files | : A total of 150 or more controls used in the whole BOOK |
- <Other devices checked by Mitsubishi (reference)>
- | | |
|-----------|------------------------|
| CPU | : Pentium® 166MHz |
| Memory | : 64MB |
| Hard disk | : 8GB (free space 6GB) |

- (b) Cause
The phenomenon has been confirmed to occur when the Matrox make MGA Mystique graphic card display driver is used.
This is because Version 4.12 of the MGA Mystique graphic card display driver is not compatible with Windows® Me.
- (c) How to judge whether the phenomenon is the same or not
After changing the used graphic driver for the standard VGA driver, delete the temporary data (*.emf) left in the temporary folder.
After that, try starting multiple Excel files.
The phenomenon seems to be the same if it does not occur by changing the driver for the standard VGA driver.
- (d) Corrective action
If this phenomenon occurs, the temporary data (*.emf) will be left in the temporary folder of the system.
You have to delete the remaining temporary data (*.emf) manually.
The temporary folder of the system is normally in C:\Temp.
After that, take either of the following actions.
1) Use the graphic card and display driver which support Windows® Me.
2) Reduce the number of control objects pasted to the Excel files.
- (40) Precautions for COM communication or TCP/IP communication on ASP page and application*1
If the ASP page opens COM or TCP/IP communication earlier than the application, communication in the same path cannot be made on the application until the ASP page is closed. Therefore, note the following points.
- (a) COM or TCP/IP communication should be opened on the application earlier.
After it has been opened on the application, communication can be made on both the application and ASP page until it is closed.
- (b) When COM or TCP/IP communication has been opened on the ASP page, always close the communication.
- *1 The application indicates any of the user applications created using the MX series and MELSOFT products.
- (41) Precautions for connecting personal computer and serial communication module
- (a) When QJ71C24-R2 of function version A is used
An MX Component application can use only either of CH1 and CH2.
When the MELSOFT product, such as GX Developer or GOT, is using one channel, the application cannot use the other channel.
When the QJ71C24-R2 of function version B is used, the application can use both channels.
- (b) When AJ71QC24-R2 or A1SJ71QC4-R2 is used
The MX Component application can use only CH1.
It cannot use CH2.

INTRODUCTION

Thank you for choosing the Mitsubishi MELSOFT series comprehensive Factory Automation software. Read this manual and make sure you understand the functions and performance of MELSOFT series thoroughly in advance to ensure correct use.

CONTENTS

| | |
|---|------|
| SAFETY PRECAUTIONS..... | A- 1 |
| REVISIONS..... | A- 2 |
| Operating Instructions..... | A- 3 |
| CONTENTS..... | A-12 |
| About Manuals..... | A-15 |
| How to Use This Manual..... | A-16 |
| Abbreviations and Terms in This Manual..... | A-17 |

| | |
|-------------------|---------------------|
| 1 OVERVIEW | 1- 1 to 1- 3 |
|-------------------|---------------------|

| | |
|---|------|
| 1.1 Outline of ACT controls..... | 1- 1 |
| 1.2 ACT control and Function Lists..... | 1- 2 |
| 1.2.1 ACT control list..... | 1- 2 |
| 1.2.2 Function list..... | 1- 3 |

| | |
|---------------------------------|---------------------|
| 2 ABOUT THE ACT CONTROLS | 2- 1 to 2-16 |
|---------------------------------|---------------------|

| | |
|--|------|
| 2.1 Settings Made for Use of the ACT controls..... | 2- 1 |
| 2.1.1 When using Microsoft® Visual Basic® 6.0..... | 2- 1 |
| 2.1.2 When using Microsoft® Visual C++® 6.0..... | 2- 3 |
| 2.1.3 When using VBA..... | 2- 7 |
| 2.1.4 When using VBScript..... | 2- 9 |
| 2.2 Programming Procedures..... | 2-10 |
| 2.2.1 When using Visual Basic®..... | 2-10 |
| 2.2.2 When using Visual C++®..... | 2-11 |
| 2.2.3 When using VBA..... | 2-12 |
| 2.2.4 When using VBScript..... | 2-13 |
| 2.3 Device Types..... | 2-14 |
| 2.4 Accessible Devices and Ranges..... | 2-16 |

| | |
|--------------------------------------|---------------------|
| 3 DETAILS OF THE ACT CONTROLS | 3- 1 to 3-54 |
|--------------------------------------|---------------------|

| | |
|--|------|
| 3.1 Details of the ACT Controls..... | 3- 1 |
| 3.2 Details of the Properties..... | 3- 2 |
| 3.3 Lists of Properties Possessed by the ACT Controls..... | 3- 7 |
| 3.3.1 ActEasyIF, ActMLEasyIF control..... | 3- 8 |
| 3.3.2 ActQJ71E71TCP, ActMLQJ71E71TCP control..... | 3- 9 |
| 3.3.3 ActQJ71E71UDP, ActMLQJ71E71UDP control..... | 3-11 |
| 3.3.4 ActAJ71QE71TCP, ActMLAJ71QE71TCP control..... | 3-13 |
| 3.3.5 ActAJ71QE71UDP, ActMLAJ71QE71UDP control..... | 3-14 |
| 3.3.6 ActAJ71E71TCP, ActMLAJ71E71TCP control..... | 3-15 |

| | |
|--|------|
| 3.3.7 ActAJ71E71UDP, ActMLAJ71E71UDP control | 3-16 |
| 3.3.8 ActQCPUQ, ActMLQCPUQ control..... | 3-17 |
| 3.3.9 ActQCPUA, ActMLQCPUA control..... | 3-19 |
| 3.3.10 ActQnACPU, ActMLQnACPU control | 3-20 |
| 3.3.11 ActACPU, ActMLACPU control | 3-21 |
| 3.3.12 ActFXCPU, ActMLFXCPU control..... | 3-22 |
| 3.3.13 ActQJ71C24, ActMLQJ71C24 control..... | 3-23 |
| 3.3.14 ActAJ71QC24, ActMLAJ71QC24 control..... | 3-27 |
| 3.3.15 ActAJ71UC24, ActMLAJ71UC24 control | 3-29 |
| 3.3.16 ActAJ71C24, ActMLAJ71C24 control..... | 3-31 |
| 3.3.17 ActQCPUQUSB, ActMLQCPUQUSB control | 3-33 |
| 3.3.18 ActCCG4QnA, ActMLCCG4QnA control | 3-35 |
| 3.3.19 ActCCG4A, ActMLCCG4A control | 3-36 |
| 3.3.20 ActMnet10BD, ActMLMnet10BD control..... | 3-37 |
| 3.3.21 ActMnetHBD, ActMLMnetHBD control..... | 3-41 |
| 3.3.22 ActCCBD, ActMLCCBD control..... | 3-48 |
| 3.3.23 ActAnUBD, ActMLAnUBD control | 3-52 |
| 3.3.24 ActLLT, ActMLLLT control | 3-53 |
| 3.3.25 ActQCPUQBus, ActMLQCPUQBus control..... | 3-54 |

| | |
|-------------|--------------|
| 4 FUNCTIONS | 4- 1 to 4-38 |
|-------------|--------------|

| | |
|--|------|
| 4.1 Programming Instructions..... | 4- 1 |
| 4.2 Details of the Functions (Dispatch Interface) | 4- 3 |
| 4.2.1 Open (Communication line opening)..... | 4- 3 |
| 4.2.2 Close (Communication line closing) | 4- 4 |
| 4.2.3 ReadDeviceBlock (Device batch-read) | 4- 5 |
| 4.2.4 WriteDeviceBlock (Device batch-write) | 4- 7 |
| 4.2.5 ReadDeviceRandom (Device random-read)..... | 4- 9 |
| 4.2.6 WriteDeviceRandom (Device random-write)..... | 4-11 |
| 4.2.7 SetDevice (Device data setting) | 4-13 |
| 4.2.8 GetDevice (Device data acquisition) | 4-14 |
| 4.2.9 ReadBuffer (Buffer memory read) | 4-15 |
| 4.2.10 WriteBuffer (Buffer memory write) | 4-17 |
| 4.2.11 GetClockData (Clock data read)..... | 4-19 |
| 4.2.12 SetClockData (Clock data write)..... | 4-21 |
| 4.2.13 GetCpuType (PLC CPU type read) | 4-23 |
| 4.2.14 SetCpuStatus (Remote control)..... | 4-27 |
| 4.2.15 EntryDeviceStatus (Device status monitor registration) | 4-29 |
| 4.2.16 FreeDeviceStatus (Device status monitor deregistration) | 4-32 |
| 4.2.17 OnDeviceStatus (Announces event) | 4-33 |
| 4.3 Details of the Functions (Custom Interface)..... | 4-34 |
| 4.3.1 Open (Communication line opening)..... | 4-34 |
| 4.3.2 Close (Communication line closing) | 4-34 |
| 4.3.3 ReadDeviceBlock (Device batch-read) | 4-34 |
| 4.3.4 WriteDeviceBlock (Device batch-write) | 4-34 |
| 4.3.5 ReadDeviceRandom (Device random-read)..... | 4-35 |

| | |
|--|------|
| 4.3.6 WriteDeviceRandom (Device random-write)..... | 4-35 |
| 4.3.7 SetDevice (Device data setting) | 4-35 |
| 4.3.8 GetDevice (Device data acquisition) | 4-35 |
| 4.3.9 ReadBuffer (Buffer memory read) | 4-36 |
| 4.3.10 WriteBuffer (Buffer memory write) | 4-36 |
| 4.3.11 GetClockDSata (Clock data read) | 4-36 |
| 4.3.12 SetClockData (Clock data write)..... | 4-37 |
| 4.3.13 GetCpuType (PLC CPU type read) | 4-37 |
| 4.3.14 SetCpuStatus (Remote control)..... | 4-37 |
| 4.3.15 EntryDeviceStatus (Device status monitor registration) | 4-38 |
| 4.3.16 FreeDeviceStatus (Device status monitor deregistration) | 4-38 |
| 4.3.17 OnDeviceStatus (Announces event) | 4-38 |

| | |
|--------------------------|---------------------|
| 5 SAMPLE PROGRAMS | 5- 1 to 5-14 |
|--------------------------|---------------------|

| | |
|--|------|
| 5.1 Visual Basic® Sample Program | 5- 1 |
| 5.2 Visual C++® Sample Programs | 5- 3 |
| 5.2.1 Dispatch interface..... | 5- 3 |
| 5.2.2 Custom interface | 5- 5 |
| 5.3 VBA Sample Programs..... | 5- 6 |
| 5.3.1 Excel Sample Program | 5- 6 |
| 5.3.2 Access Sample Program | 5- 8 |
| 5.4 VBScript Sample Program..... | 5-10 |
| 5.5 ASP Sample Program..... | 5-12 |

| | |
|----------------------|---------------------|
| 6 ERROR CODES | 6- 1 to 6-11 |
|----------------------|---------------------|

| | |
|--|------|
| 6.1 Error Codes Returned by the ACT controls | 6- 1 |
| 6.2 Error Codes Returned by the CPUs, Modules and Network Boards | 6- 9 |
| 6.3 HRESULT Type Error Codes | 6-10 |

About Manuals

The following lists the manuals for this software package.
Refer to the following table when ordering manuals.

Related Manuals

| Manual Name | Manual Number (Model Code) |
|---|-------------------------------|
| MX Component Version 2 Operating Manual (Startup) Provides procedures for installing and uninstalling MX Component and for browsing the operating manual. (Sold separately) | IB-080153 (13JU10) |
| MX Component Version 2 Operating Manual Gives how to perform setting and operation of each utility on MX Component. (Sold separately) | SH-080154 (13JU11) |
| Type A70BDE-J71QLP23/A70BDE-J71QLP23GE/A70BDE-J71QBR13/A70BDE-J71QLR23 MELSECNET/10 Interface Board User's Manual(For SW3DNF-MNET10) Describes the features, specifications, part names and setting of the MELSECNET/10 board, and the installation, uninstallation and others of the driver. (Sold separately) | IB-0800035 (13JL93) |
| Type A80BDE-J61BT11 Control & Communication Link System Master/Local Interface Board User's Manual (For SW4DNF-CCLINK-B) Describes the features, specifications, part names and setting of the CC-Link master board, and the installation, uninstallation and others of the driver. (Sold separately) | IB-0800175 (13JR28) |
| Type A80BDE-J61BT13 Control & Communication Link System Local Interface Board User's Manual (For SW4DNF-CCLINK-B) Describes the features, specifications, part names and setting of the CC-Link local board, and the installation, uninstallation and others of the driver. (Sold separately) | IB-0800176 (13JR29) |
| Type A80BDE-A2USH-S1 PLC CPU Board User's Manual (For SW1DNF-ANU-B) Describes the features, specifications, part names and setting of the CPU board, and the installation, uninstallation and others of the driver. (Sold separately) | IB-0800174 (13JR27) |
| MELSECNET/H Interface Board User's Manual(For SW0DNC-MNETH-B) Describes the features, specifications, part names and setting of the MELSECNET/H board, and the installation, uninstallation and others of the driver. (Sold separately) | SH-080128 (13JR24) |

Note: The MX Component Version 2 Operating Manual (Startup) and MX Component Version 2 Operating Manual are contained in the CD-ROM together with the software package as a set.
When you want to purchase the manual alone, it is optionally available as the printed matter of the manual number (Model code) in the above table.

How to Use This Manual

"How to Use This Manual" is given purpose-by-purpose for use of MX Component. Refer to the following outlines and use this manual.

- (1) To know the feature and ACT control lists (Chapter 1)
Chapter 1 gives the ACT control outline and ACT control lists.
- (2) To use the ACT controls on Visual Basic® or Visual C++® (Section 2.1)
Section 2.1 provides how to make settings on Visual Basic® and Visual C++® to use the ACT controls.
- (3) To know the programming procedure (Section 2.2)
Section 2.2 contains programming procedures.
- (4) To know the device types to be specified in the functions (Section 2.3)
Section 2.3 lists the device types.
- (5) To know the details of the ACT controls (Chapter 3)
Chapter 3 provides the details of the ACT controls.
Read this chapter when creating a program.
- (6) To know the details of the functions (Chapter 4)
Chapter 4 gives the details of the functions.
Read this chapter when creating a program.
- (7) To know how to use the sample programs (Chapter 5)
Chapter 5 provides the sample programs and how to use them.
Use them as reference when creating a program.
- (8) To know the definitions of the error codes (Chapter 6)
Chapter 6 lists the error codes returned by the ACT controls and the error codes returned by the CPUs, modules and network boards.
- (9) To know the accessible devices and ranges
The MX Component operating manual contains the accessible devices and ranges.
Refer to the MX Component operating manual.

Abbreviations and Terms in This Manual

Unless otherwise started, this manual uses the following abbreviations and terms for the explanation of MX Component.

| Generic Term/Abbreviation | Description |
|---------------------------|--|
| MX Component | Generic product name for product types SWnD5C-ATC-E and SWnD5C-ACT-EA. (n denotes version 0 or 2) -EA denotes a multiple license product. |
| IBM-PC/AT compatible | Abbreviation of the IBM PC/AT or its compatible personal computer |
| PC CPU module | Abbreviation of the MELSEC-Q series compatible PC CPU module (CONTEC CO., LTD. make). |
| GX Developer | Abbreviation of Type SW□D5C-GPPW-E/SW□D5F-GPPW-E GPP function software package |
| GX Simulator | Abbreviation of Type SW□D5C-LLT-E/SW□D5F-LLT-E Ladder Logic Test tool function software package |
| MELSECNET/10 board | Abbreviation of Type A70BDE-J71QLP23/A70BDE-J71QLP23GE/A70BDE-J71QBR13/A70BDE-J71QLR23 MELSECNET/10 interface board |
| MELSECNET/H board | Abbreviation of Type Q80BD-J71LP21-25/Q80BD-J71LP21G/Q80BD-J71BR11 MELSECNET/H board |
| CC-Link board | Abbreviation of Type A80BDE-J61BT11 CC-Link system master/local interface board and Type A80BDE-J61BT13 CC-Link interface board |
| CPU board | Abbreviation of Type A80BDE-A2USH-S1 PLC CPU board |
| AnNCP | Generic term of the A0J2HCPU, A1SCPU, A1SCPU-S1, A1SCPUC24-R2, A1SHCPU, A1SJCPU, A1SJHCPU, A1NCP, A2CCPU, A2CCPUC24, A2CCPUC24-PRF, A2CJCPU, A2NCP, A2NCP-S1, A2SCPU, A2SCPU-S1, A2SHCPU, A2SHCPU-S1, A3NCP and A1FXCPU |
| AnACPU | Generic term of the A2ACPU, A2ACPU-S1, A2ACPUP21/R21, A2ACPUP21-S1, A3ACPU and A3ACPUP21/R21 |
| AnUCPU | Generic term of the A2UCPU, A2UCPU-S1, A2USCPU, A2USCPU-S1, A2ASCPU, A2ASCPU-S1, A2ASCPU-S30, A2USHCPU-S1, A3UCPU and A4UCPU |
| QnACPU | Generic term of the Q2ACPU, Q2ACPU-S1, Q2ASCPU, Q2ASCPU-S1, Q2ASHCPU, Q2ASHCPU-S1, Q3ACPU, Q4ACPU and Q4ARCPU |
| ACPU | Generic term of the AnNCP, AnACPU and AnUCPU |
| QCPU (A mode) | Generic term of the Q02CPU-A, Q02HCPU-A and Q06HCPU-A |
| QCPU (Q mode) | Generic term of the Q00JCPU, Q00CPU, Q01CPU, Q02CPU, Q02HCPU, Q06HCPU, Q12HCPU and Q25HCPU Note that especially when the CPU is indicated as a different model, the Q00JCPU, Q00CPU and Q01CPU are described as the Q00J/Q00/Q01CPU, and the Q02CPU, Q02HCPU, Q06HCPU, Q12HCPU and Q25HCPU as the Q02/Q02H/Q06H/Q12H/Q25HCPU. |
| FXCPU | Generic term of the FX0, FX0S, FX0N, FX1, FX1N, FX1NC, FX1S, FX2, FX2C, FX2N and FX2NC series |
| Motion controller CPU | Generic term of the A171SHCPU, A172SHCPU, A173UHCPU, A173UHCPU-S1, A273UHCPU and A273UHCPU-S3 |
| PLC CPU | Generic term of the QCPU(Q mode), QCPU(A mode), QnACPU, ACP, FXCPU and motion controller CPU |
| C24 | Generic term of the A1SCPUC24-R2, A1SJ71C24-PRF, A1SJ71C24-R2, A1SJ71C24-R4, A2CCPUC24, A2CCPUC24-PRF, AJ71C24-S6 and AJ71C24-S8 |
| UC24 | Generic term of the AJ71UC24, A1SJ71UC24-R2, A1SJ71UC24-R4 and A1SJ71UC24-PRF |
| QC24 | Generic term of the AJ71QC24, AJ71QC24-R2, AJ71QC24-R4, A1SJ71QC24-R2 and A1SJ71QC24-R2 |
| QC24N | Generic term of the AJ71QC24N, AJ71QC24N-R2, AJ71QC24N-R4, A1SJ71QC24N and A1SJ71QC24N-R2 |
| QC24(N) | Generic term of the QC24 and QC24N |
| Q series-compatible C24 | Generic term of the QJ71C24 and QJ71C24-R2 |

| Generic Term/Abbreviation | Description |
|---|---|
| Computer link module (Serial communication module) | Generic term of the C24, UC24, QC24(N) and Q series-compatible C24 Described as the serial communication module especially to indicate the QC24(N) or Q series-compatible C24. |
| E71 | Generic term of the AJ71E71, AJ71E71-S3, A1SJ71E71-B2, A1SJ71E71-B5, A1SJ71E71-B2-S3 and A1SJ71E71-B5-S3 |
| QE71 | Generic term of the AJ71QE71, AJ71QE71-B5, A1SJ71QE71-B2 and A1SJ71QE71-B5 |
| Q series-compatible E71 | Generic term of the QJ71E71 and QJ71E71-B2 |
| Ethernet module | Generic term of the E71, QE71 and Q series-compatible E71 |
| CC-Link G4 module | Generic term of the AJ65BT-G4 GPP function peripheral connection module and the AJ65BT-G4-S3 GPP function peripheral connection module |
| Computer link communication (Serial communication) | Abbreviation of communication made with the PLC CPU using the computer link module Described as serial communication especially in communication that uses the QC24(N) or Q series-compatible C24. |
| Ethernet communication | Abbreviation of communication made with the PLC CPU using the Ethernet module |
| CPU COM communication | Abbreviation of communication made by connecting the IBM-PC/AT compatible to the RS-232C or RS-422 connector of the PLC CPU |
| CPU USB communication | Abbreviation of communication made by connecting the IBM-PC/AT compatible to the USB connector of the QCPU (Q mode) |
| MELSECNET/10 communication | Abbreviation of communication made with the PLC CPU using the MELSECNET/10 board |
| MELSECNET/H communication | Abbreviation of communication made with the PLC CPU using the MELSECNET/H board |
| CC-Link communication | Abbreviation of communication made with the PLC CPU using the CC-Link board |
| CC-Link G4 communication | Abbreviation of communication made with the PLC CPU using the CC-Link G4 module |
| CPU board communication | Abbreviation of communication made with the PLC CPU using the CPU board |
| Q series bus communication | Abbreviation of communication made with the PLC CPU on the same base using the PC CPU module |
| GX Simulator communication | Abbreviation of communication made with the GX Simulator |
| Utility setting type | Abbreviation of user program creation using the communication settings utility |
| Program setting type | Abbreviation of user program creation without using the communication settings utility |
| ACT controls | Generic term of the ActiveX controls offered by MX Component |

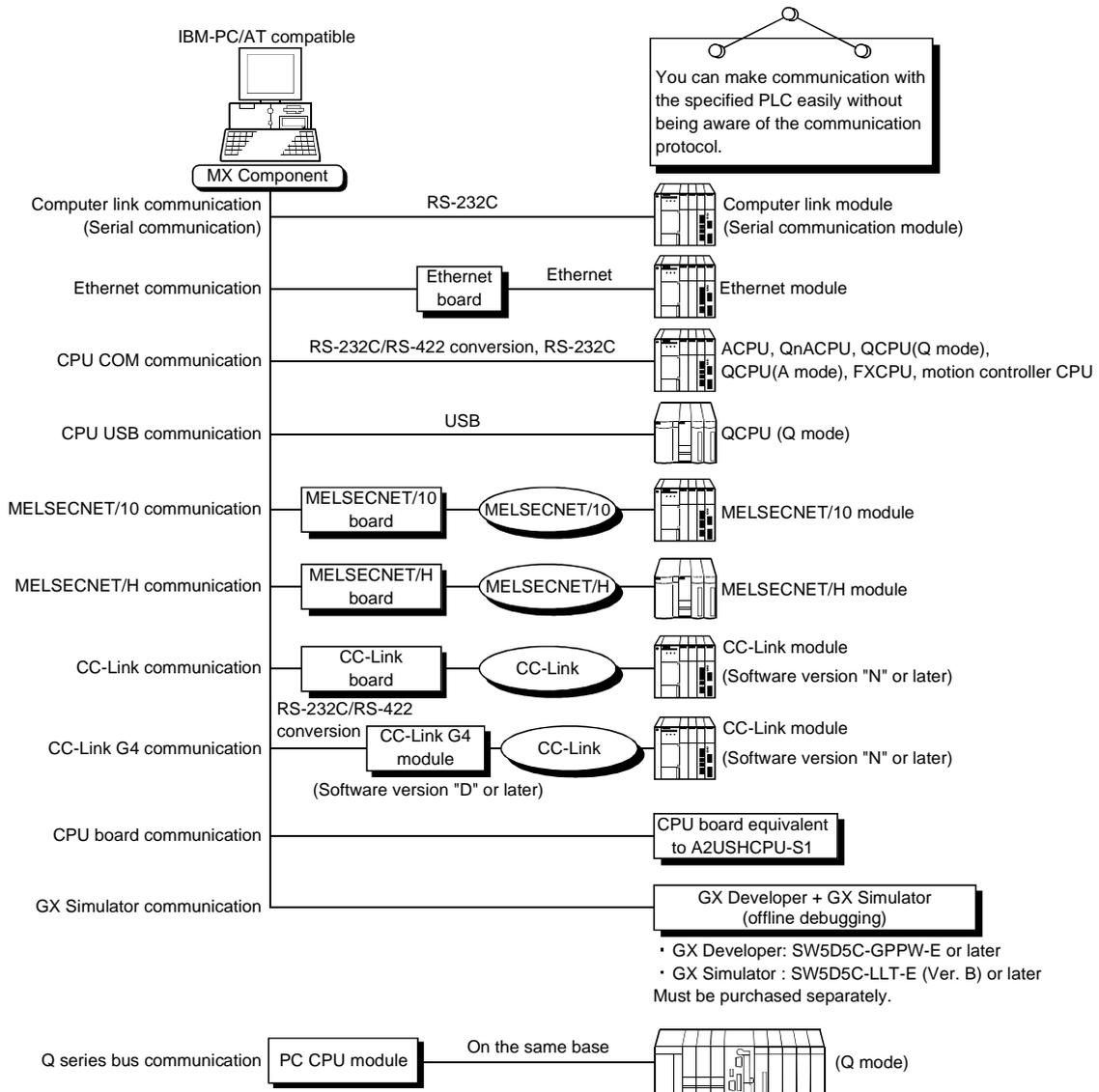
Microsoft Windows, Microsoft Windows NT, Microsoft Visual Basic and Microsoft Visual C++ are either trademarks or registered trademarks of Microsoft Corporation in the United States and/or other countries. Ethernet is the registered trademark of Xerox Corporation. Other company and product names herein may be either trademarks or registered trademarks of their respective owners.
SPREAD
Copyright(C) 1999 Far Point Technologies, Inc.

1 OVERVIEW

This chapter provides the function outline of the ACT controls offered by MX Component.

1.1 Outline of ACT controls

These controls are used to create user programs for communication with a PLC CPU. This enables the user to make communication without being aware of the hardware and communication protocol on the other end.



1.2 ACT control and Function Lists

1

The following sections give the lists of ACT controls and functions.

1.2.1 ACT control list

The following table lists the ACT controls included in each DLL offered by MX Component.

| DLL Name | Included Control Name | | Application |
|--------------|-----------------------|------------------|---|
| | For VB, VC++, VBA | For VBScript | |
| ActMulti.DLL | ActEasyIF | ActMLEasyIF | Used to make communication settings easily on the communication settings utility to make communication. |
| ActPcCom.DLL | ActQCPUQ | ActMLQCPUQ | Used to make communication via the serial port of the corresponding PLC CPU. |
| | ActQCPUA | ActMLQCPUA | |
| | ActQnACPU | ActMLQnACPU | |
| | ActACPU | ActMLACPU | |
| | ActFXCPU | ActMLFXCPU | |
| ActComLk.DLL | ActQJ71C24 | ActMLQJ71C24 | Used to make communication via the computer link module (serial communication module). |
| | ActAJ71QC24 | ActMLAJ71QC24 | |
| | ActAJ71UC24 | ActMLAJ71UC24 | |
| | ActAJ71C24 | ActMLAJ71C24 | |
| ActEther.DLL | ActQJ71E71TCP | ActMLQJ71E71TCP | Used to make communication via the Ethernet module. |
| | ActQJ71E71UDP | ActMLQJ71E71UDP | |
| | ActAJ71QE71TCP | ActMLAJ71QE71TCP | |
| | ActAJ71QE71UDP | ActMLAJ71QE71UDP | |
| | ActAJ71E71TCP | ActMLAJ71E71TCP | |
| | ActAJ71E71UDP | ActMLAJ71E71UDP | |
| ActPcUsb.DLL | ActQCPUQUSB | ActMLQCPUQUSB | Used to make communication via the USB port of the PLC CPU. |
| ActCcG4.DLL | ActCCG4QnA | ActMLCCG4QnA | Used to make communication via the CC-Link G4 module. |
| | ActCCG4A | ActMLCCG4A | |
| ActBoard.DLL | ActMnet10BD | ActMLMnet10BD | Used to make communication with or via the network board. |
| | ActMnetHBD | ActMLMnetHBD | |
| | ActCCBD | ActMLCCBD | |
| | ActAnUBD | ActMLAnUBD | |
| ActLLT.DLL | ActLLT | ActMLLLT | Used to make communication with the GX Simulator. |
| ActPcQbf.DLL | ActQCPUQBus | ActMLQCPUQBus | Used to make Q series bus communication with the PC CPU module. |

1.2.2 Function list

The following table lists the features of the functions and the functions available for the ACT controls.

(1) Function list

Refer to "CHAPTER 4 FUNCTIONS" for full information on the functions.

| Function Name | Feature |
|-------------------|------------------------------------|
| Open | Opens a communication line. |
| Close | Closes a communication line. |
| ReadDeviceBlock | Batch-reads data from devices. |
| WriteDeviceBlock | Batch-writes data to devices. |
| ReadDeviceRandom | Randomly reads data from devices. |
| WriteDeviceRandom | Randomly writes data to devices. |
| SetDevice | Sets one device. |
| GetDevice | Acquires the data of one device. |
| ReadBuffer | Reads data from buffer memory. |
| WriteBuffer | Writes data to buffer memory. |
| GetClockData | Reads clock data from PLC CPU. |
| SetClockData | Writes clock data to PLC CPU. |
| GetCpuType | Reads PLC CPU type. |
| SetCpuStatus | Remote run/stop/pause of PLC CPU. |
| EntryDeviceStatus | Registers device status monitor. |
| FreeDeviceStatus | Deregisters device status monitor. |
| OnDeviceStatus | Announces event. |

(2) Functions available for the ACT controls

Refer to "CHAPTER 4 FUNCTIONS" for full information on the functions available for the ACT controls.

2 ABOUT THE ACT CONTROLS

This chapter explains the settings made for use of the ACT controls, the programming procedures, the device types and the accessible ranges.

2.1 Settings Made for Use of the ACT controls

2

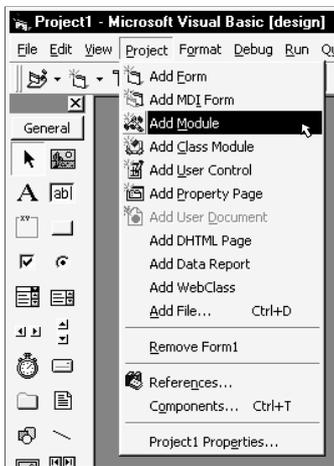
This section describes the setting operation performed for use of the ACT controls.

2.1.1 When using Microsoft® Visual Basic® 6.0

Perform the following setting operation when using Visual Basic® .

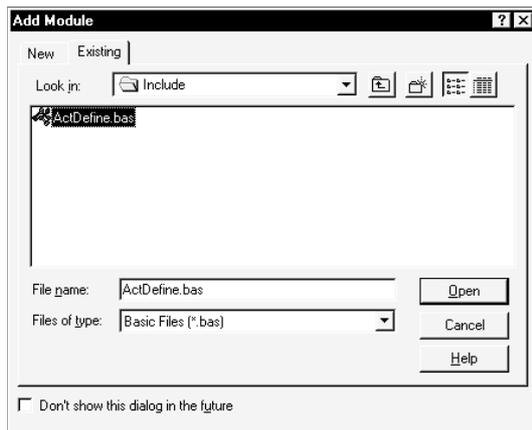
(1) Setting the include file

- 1) Start Visual Basic® and choose the [Project]-[Add Module] menu.

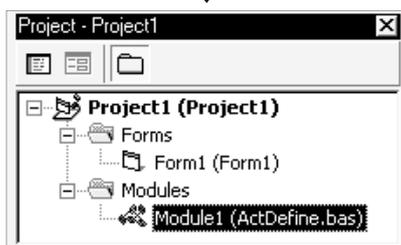


- 2) Choose the <<Existing>> tab and select "ActDefine.bas".

"ActDefine.bas" is stored in <User specified folder>-<Act>-<Include> at the time of installation.

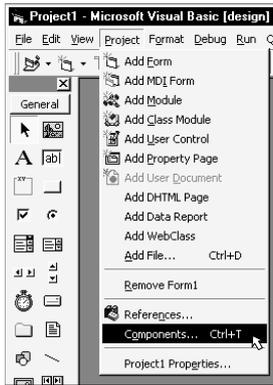


- 3) Registering "ActDefine.bas" adds it to Modules.



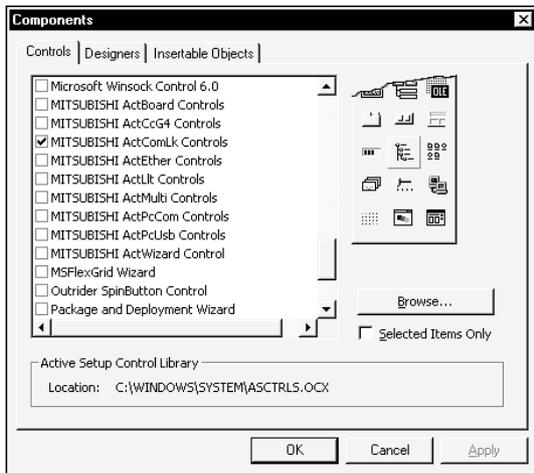
(2) Registering the ACT controls

1) Choose the [Project]-[Components] menu.



2

2) Select the <<Controls>> tab and choose the DLL which includes the ACT controls you want to use.



3) The ACT controls included in the selected DLL are added to the toolbox.

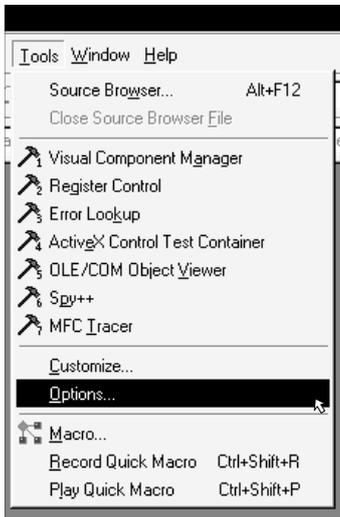


2.1.2 When using Microsoft® Visual C++® 6.0

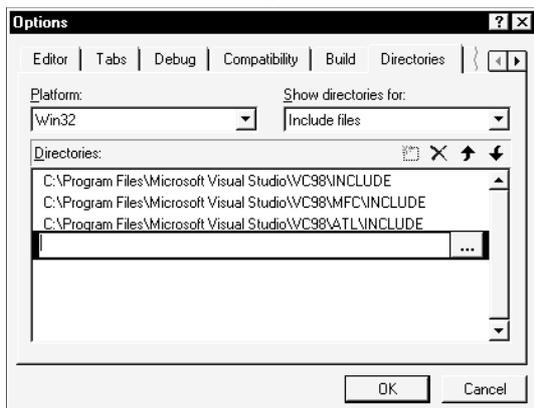
Perform the following setting operation when using Visual C++® .

(1) Setting the include file

1) Start Visual C++® and choose the [Tools]-[Options] menu.

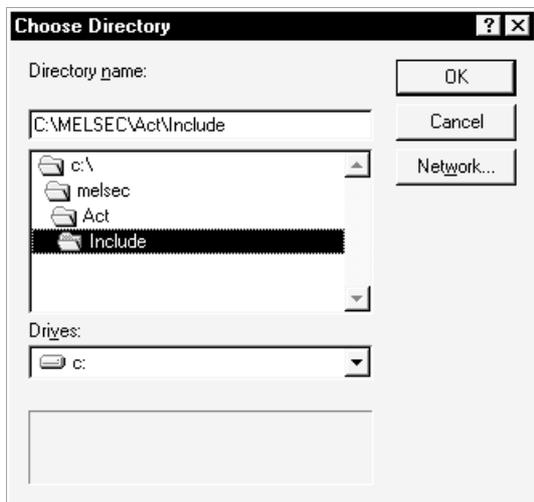


2) Choose the <<Directories>> tab and set "Include files" in "Show directories for:".

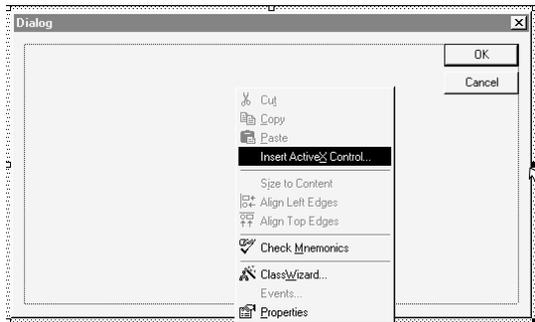


3) Double-click the item to be set, and browse the include file.

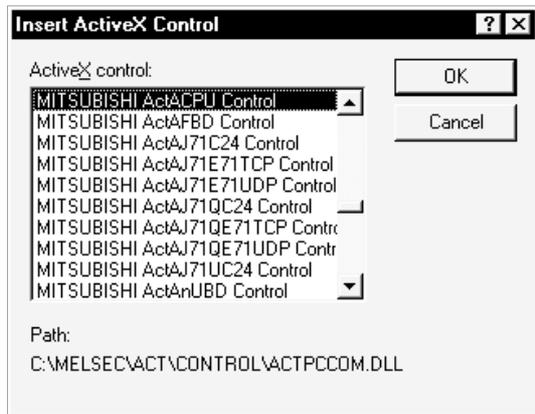
"ActDefine.H" is stored in <User specified folder>-<Act>-<Include> at the time of installation.



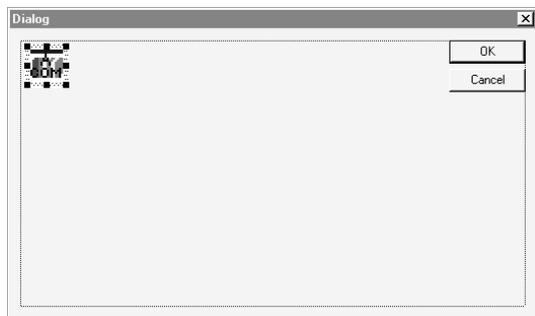
(2) Registering the ACT control



1) Right-click the form to choose "Insert ActiveX Control".

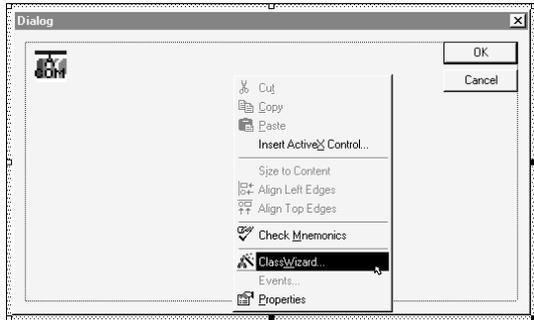


2) Select the ACT control you want to use.

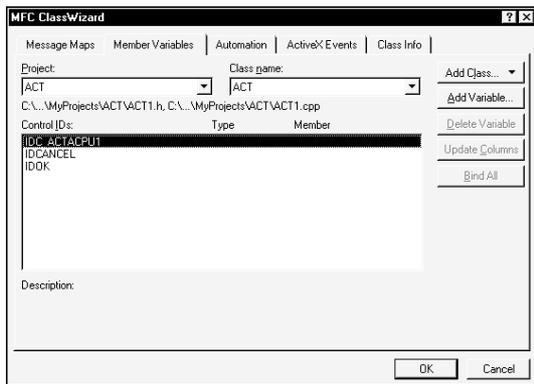


3) The selected ACT control is pasted to the form.

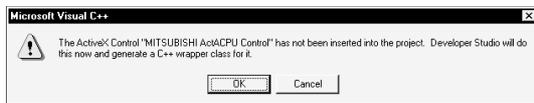
(3) Adding the member variable



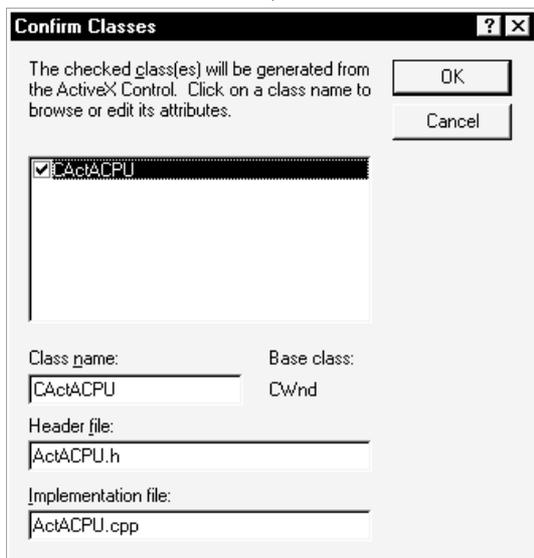
1) Click the form to choose "Class Wizard".



2) When the left dialog box appears, choose the <<Member Variables>> tab. Choose the member variable adding control ID and click the **Add Variable** button.



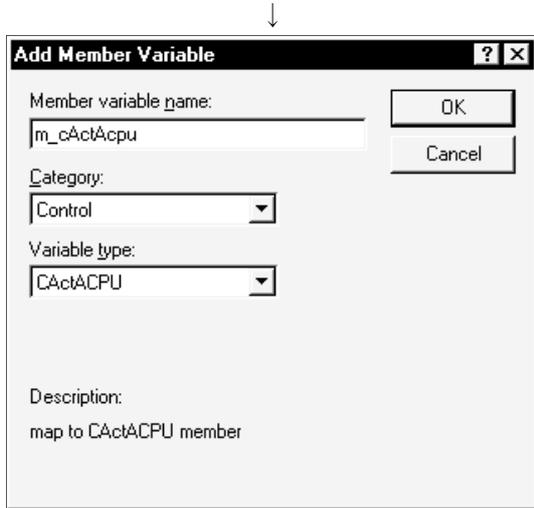
3) When the left screen appears, read the information and click the **OK** button.



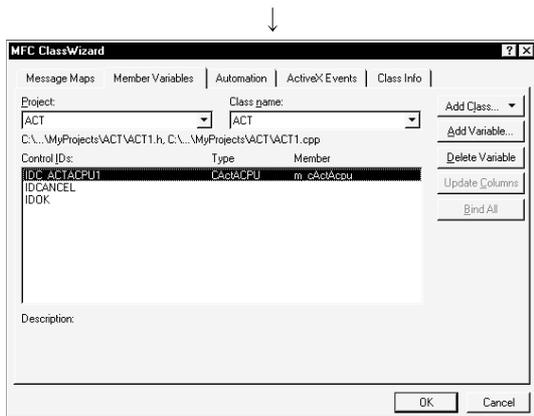
4) Check the class checkbox and click the **OK** button.

(To the next page.)

(From the previous page)



5) Enter the member variable name and click the **OK** button.

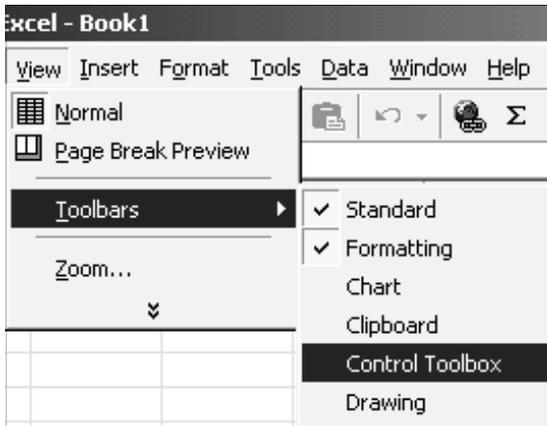


6) Make sure that the member variable has been registered.

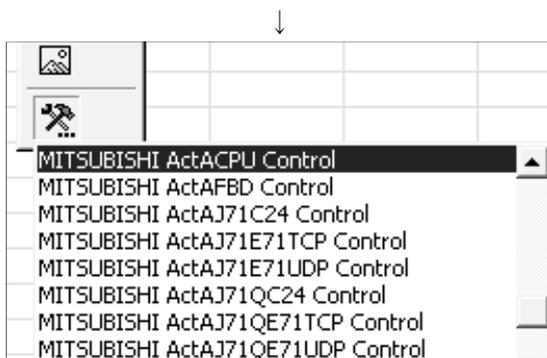
2.1.3 When using VBA

Perform the following setting operation when using VBA.

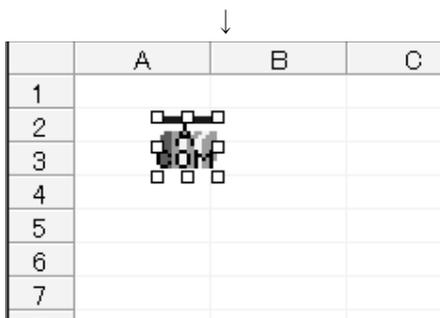
(1) When using Microsoft® Excel 2000



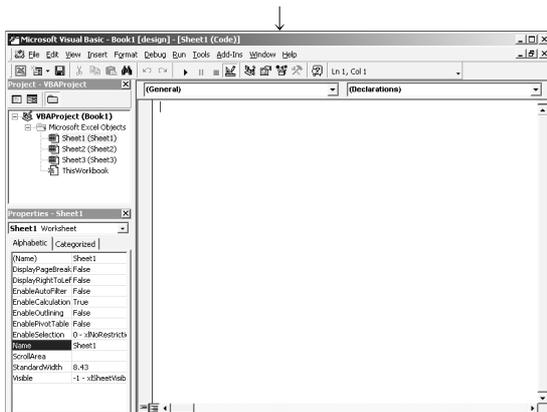
1) Boot Excel 2000 and choose the [View]-[Toolbars]-[Control Toolbox] menu.



2) Click the  button of the displayed Control Toolbox. As this displays a menu, choose the ACT control you want to use.



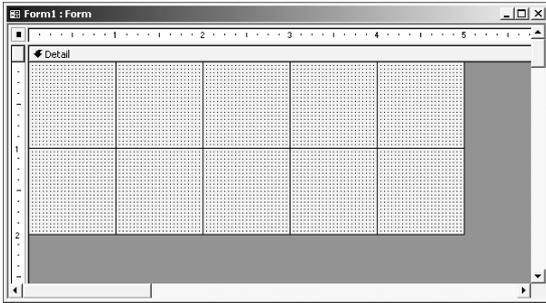
3) Paste the selected ACT control to a sheet.



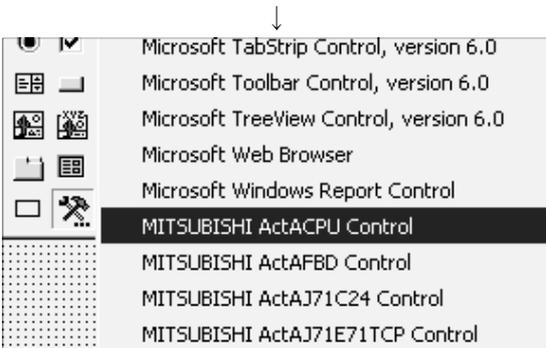
4) Choose the [Tools]-[Macro]-[Visual Basic Editor] menu to start Visual Basic Editor.

5) Perform programming on Visual Basic Editor.

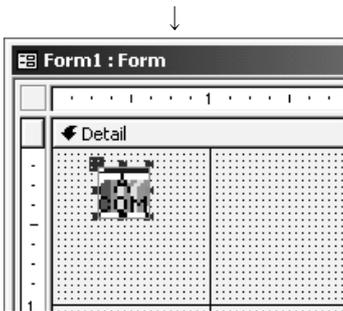
(2) When using Microsoft® Access 2000



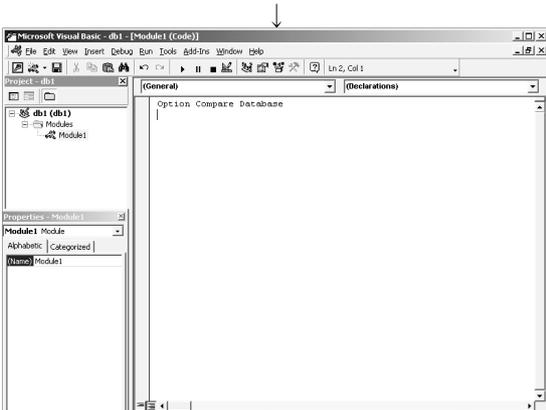
1) Boot Access 2000 and make the database form active.



2) Click the  button of the toolbox. As this displays a menu, choose the ACT control you want to use.



3) Paste the selected ACT control to a sheet.



4) Choose the [Tools]-[Macro]-[Visual Basic Editor] menu to start Visual Basic Editor.

5) Perform programming on Visual Basic Editor.

2.1.4 When using VBScript

Create HTML or ASP using the notepad, commercially available text editor, HTML creation tool or like.

Refer to the commercially available references and so on for the grammars of HTML and ASP.

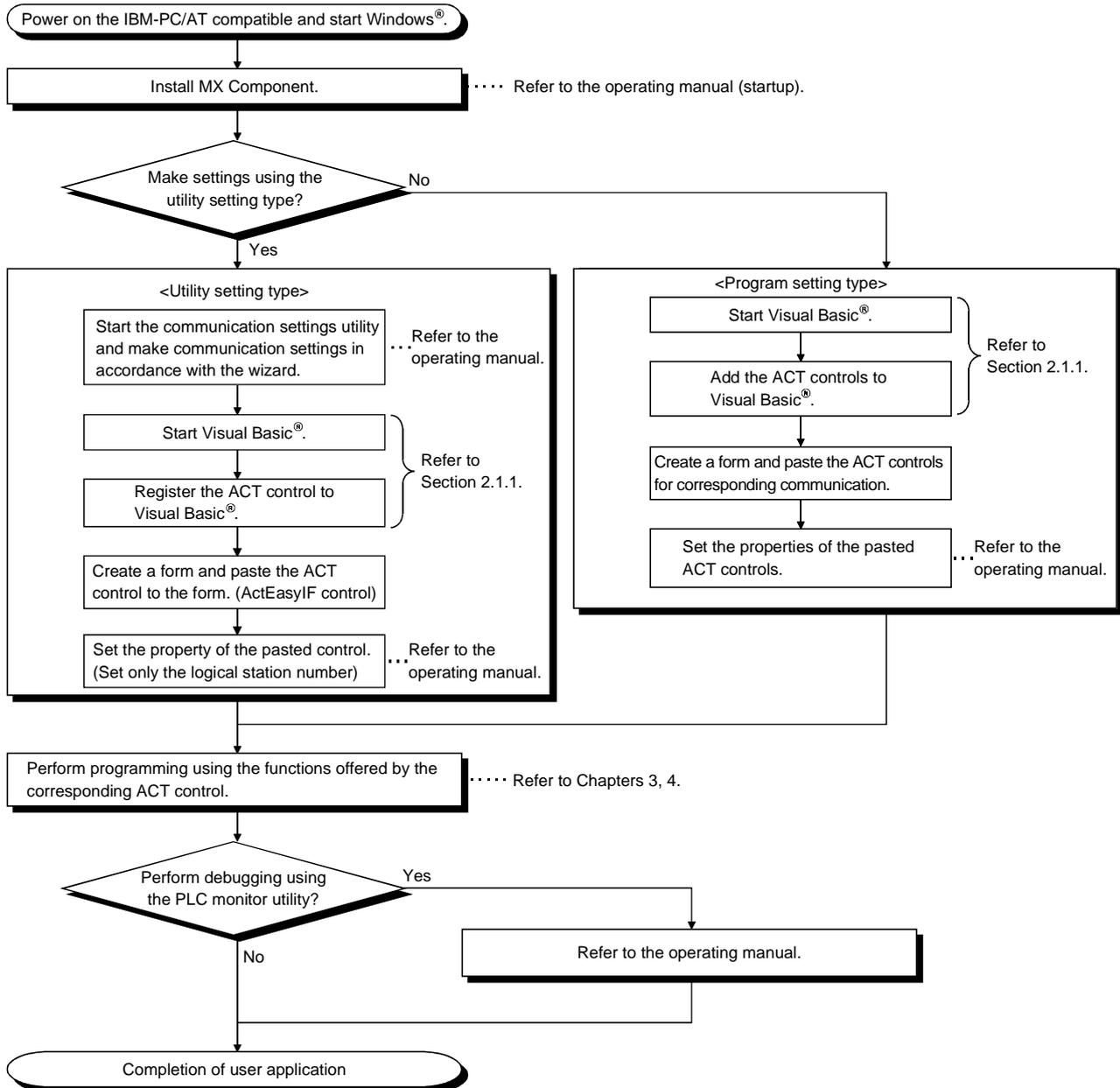
Also refer to the HTML and ASP sample programs installed in MX Component.

2.2 Programming Procedures

This section gives the procedures of creating a user application.

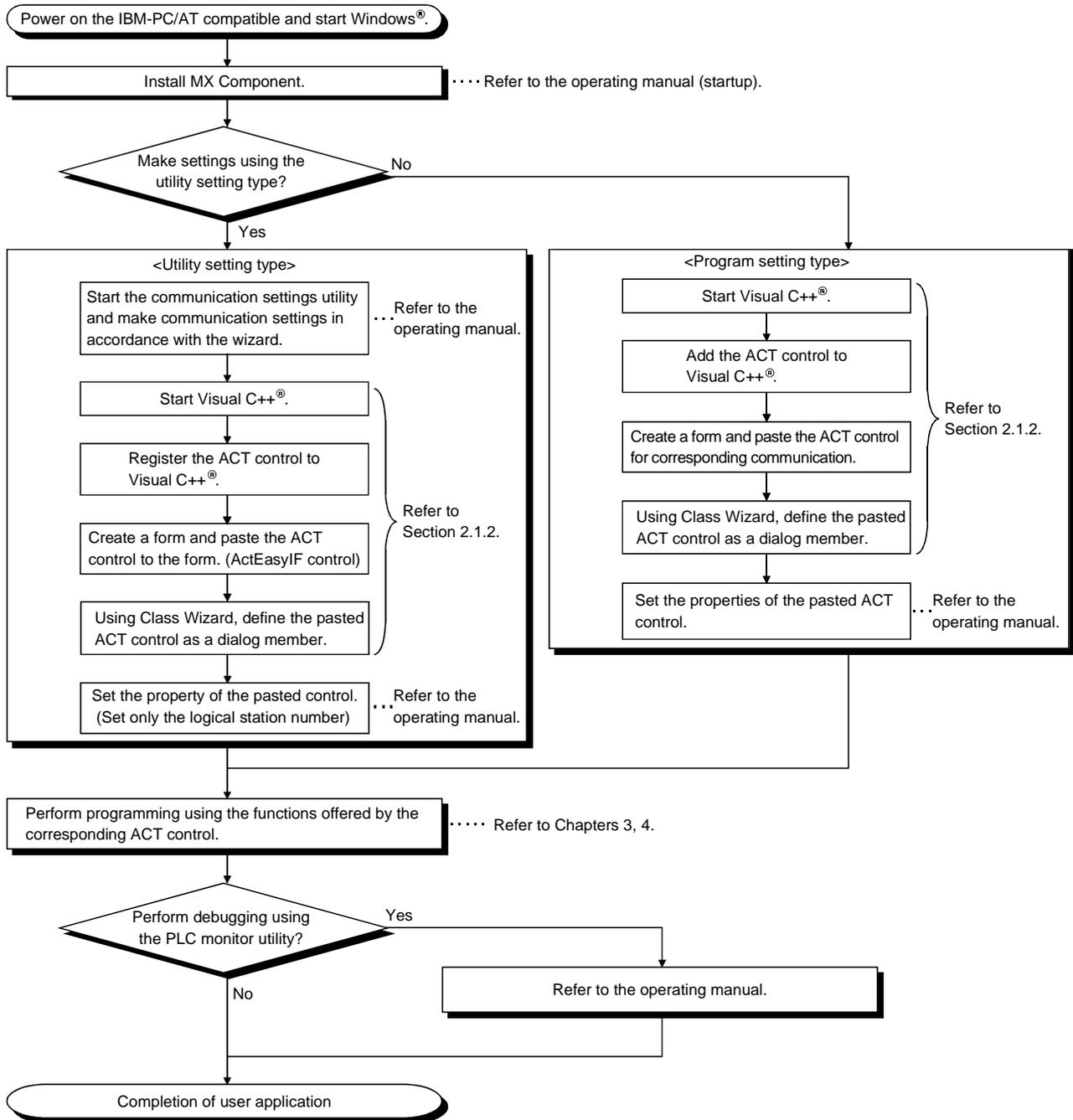
2.2.1 When using Visual Basic®

When using Visual Basic®, create a user application in the following procedure.



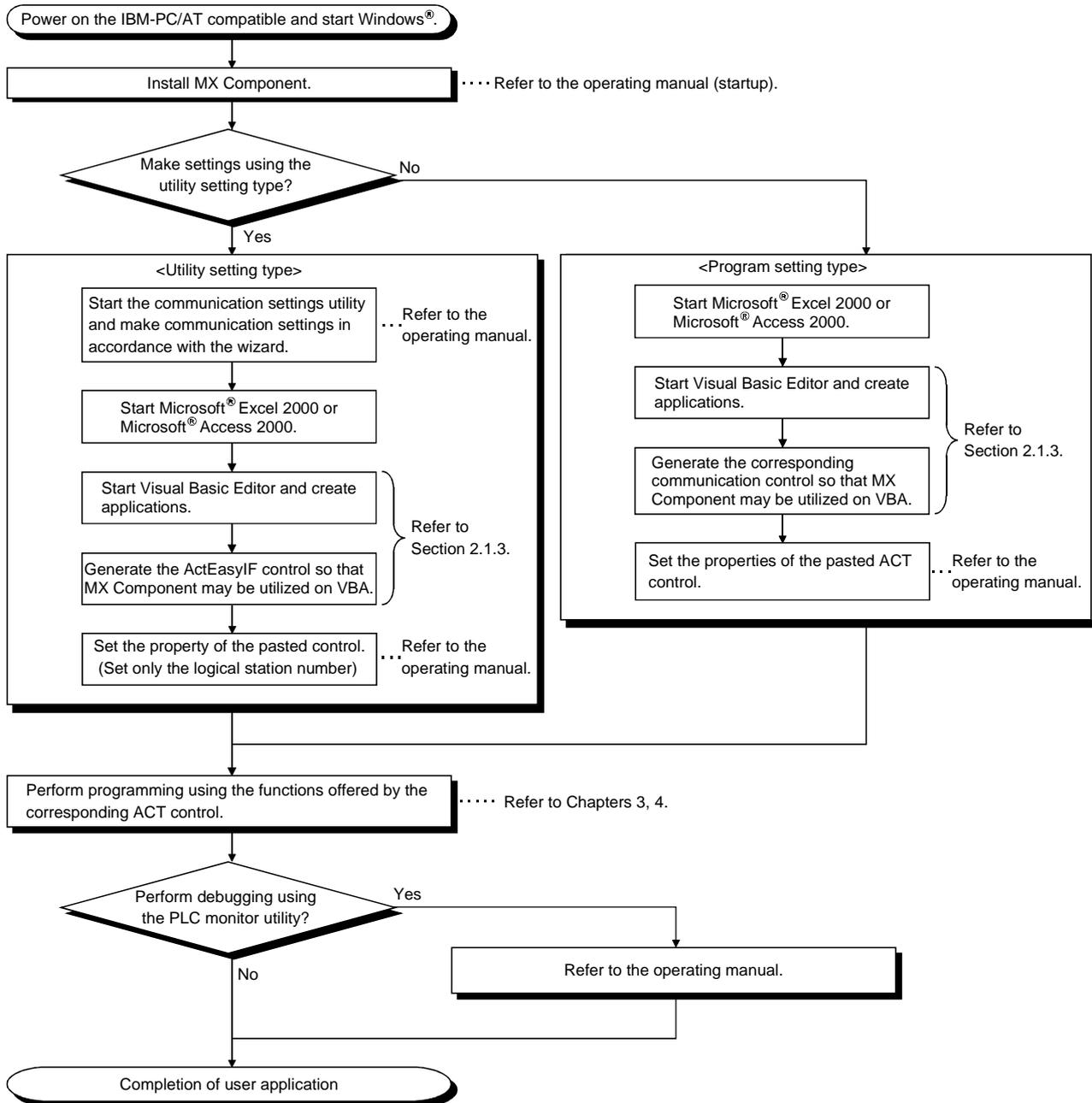
2.2.2 When using Visual C++®

When using Visual C++®, create a user application in the following procedure.



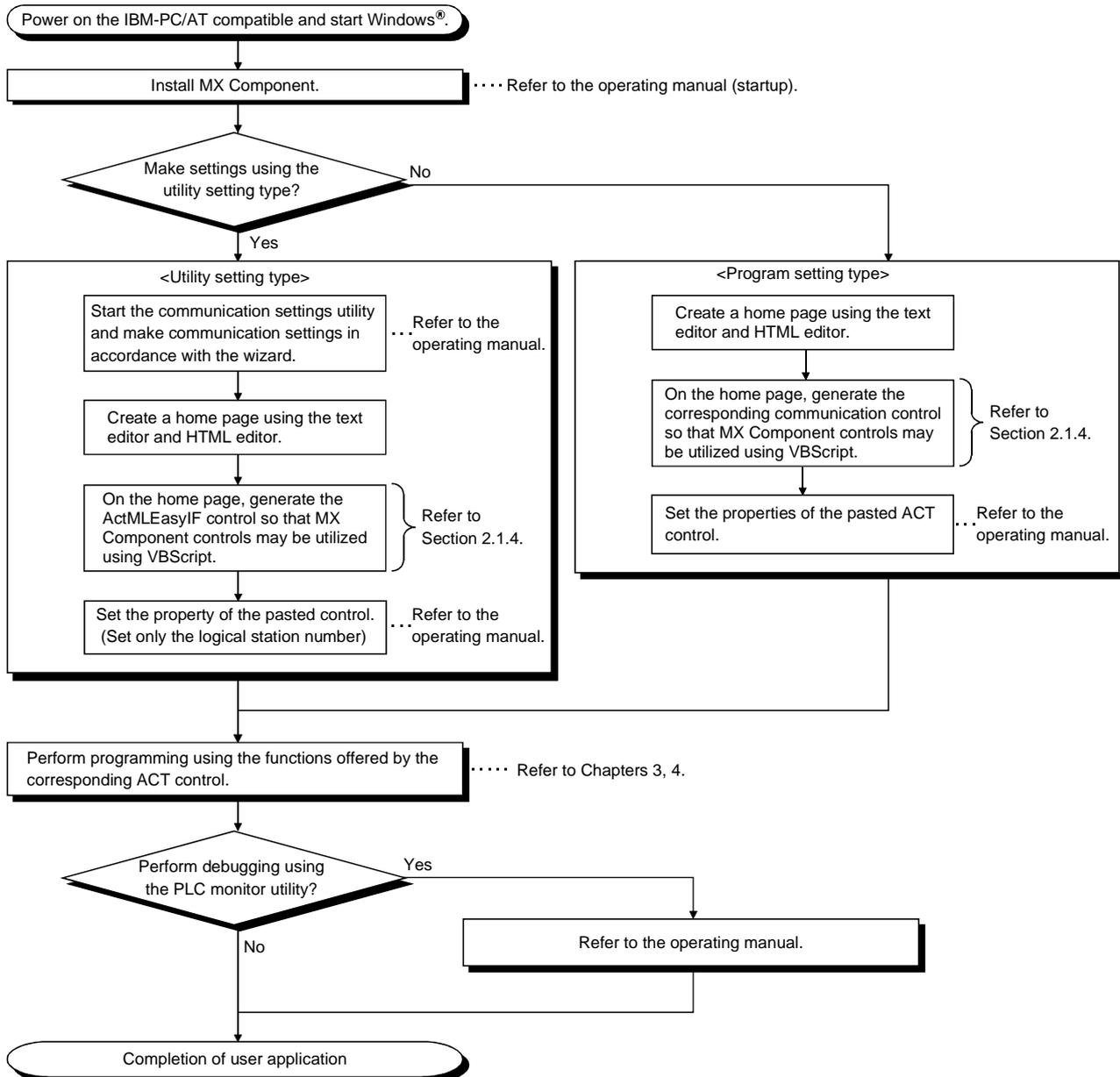
2.2.3 When using VBA

When using VBA, create a user application in the following procedure.



2.2.4 When using VBScript

When using VBScript, create a user application in the following procedure.



2.3 Device Types

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

| POINT |
|--|
| (1) For the functions (ReadDeviceBlock, WriteDeviceBlock, ReadDeviceRandom, WriteDeviceRandom, SetDevice and GetDevice), specify the devices in the form of "device name + device number". For the device numbers, note the differences between octal, decimal and hexadecimal numbers. |
| (2) When specifying bit devices for ReadDeviceBlock or WriteDeviceBlock, specify the device number as a multiple of 16. |
| (3) Only the devices indicated in this section are supported. Do not use unsupported devices. |

(1) Common

The following device types are common to all communication paths.

| Device | Device Name | Device No. Type | Representation | Remarks | |
|-----------------------|---------------|-----------------|----------------|--------------------|--|
| Function input | FX | Decimal | Bit | — | |
| Function output | FY | Decimal | Bit | — | |
| Function register | FD | Decimal | Word | 4 words/1 point *1 | |
| Special relay | SM | Decimal | Bit | — | |
| Special register | SD | Decimal | Word | — | |
| Input relay | X | Hexadecimal | Bit | Octal for FXCPU | |
| Output relay | Y | Hexadecimal | Bit | Octal for FXCPU | |
| Internal relay | M | Decimal | Bit | *2 | |
| Latch relay | L | Decimal | Bit | *2 | |
| Annunciator | F | Decimal | Bit | — | |
| Edge relay | V | Decimal | Bit | — | |
| Link relay | B | Decimal | Bit | — | |
| Data register | D | Decimal | Word | — | |
| Link register | W | Hexadecimal | Word | — | |
| Timer | Contact | TS | Decimal | Bit | — |
| | Coil | TC | Decimal | Bit | — |
| | Present value | TN | Decimal | Word | — |
| Counter | Contact | CS | Decimal | Bit | — |
| | Coil | CC | Decimal | Bit | — |
| | Present value | CN | Decimal | Word | For FXCPU, 200 or more is 32-bit data. |
| Retentive timer | Contact | SS | Decimal | Bit | For ACPU, use timer to specify. |
| | Coil | SC | Decimal | Bit | For ACPU, use timer to specify. |
| | Present value | SN | Decimal | Word | For ACPU, use timer to specify. |
| Link special relay | SB | Hexadecimal | Bit | — | |
| Link special register | SW | Hexadecimal | Word | — | |
| Step relay | S | Decimal | Bit | *2 | |

Bit: Bit device Word: Word device

*1: For batch operation, operation is performed continuously in units of one word.

For random operation, only the first one word is read.

*2: For the QCPU (A mode) and ACPU, the M, L and S devices have the same regions independently of the device setting in the parameters.

| Device | Device Name | Device No. | Type | Representation | Remarks |
|------------------------------------|-----------------------|----------------------|-------------|----------------|---------|
| Accumulator | A | Decimal | | Word | *5 |
| Index register | Z | Decimal | | Word | *5 |
| | V | Decimal | | Word | *5 |
| File register | R | Decimal | | Word | *3 |
| | ZR | Decimal | | Word | — |
| Extended file register | ER*\R | Decimal | | Word | *4 |
| Direct link *6 | Link input | J*\X | Hexadecimal | Bit | *4 |
| | Link output | J*\Y | Hexadecimal | Bit | *4 |
| | Link relay | J*\B | Hexadecimal | Bit | *4 |
| | Link special relay | J*\SB | Hexadecimal | Bit | *4 |
| | Link register | J*\W | Hexadecimal | Bit | *4 |
| | Link special register | J*\SW | Hexadecimal | Word | *4 |
| Special direct buffer memory *7 *9 | U*\G | Hexadecimal /decimal | | Word | *4, *8 |

Bit: Bit device Word: Word device

- *3: To specify the extended file register, describe "\" between the block number part and file register part.
 Specifying R** specifies R of block No. 0.
 Specifying ER\R** returns an error.
 Specifying ER*\R** does not enable extension representation (indirect specification, digit specification).
- *4: For direct specification, describe "\" between the direct specification part and device specification part.
- *5: Cannot be used when E71 is relayed.
- *6: For J*, specify the network number.
- *7: Specify the special module I/O number (hexadecimal) for U*, and the buffer memory address (decimal) for G*.
 (Example: Specify "U20\G100" when the special module I/O number is 200H and the buffer memory address is 100.)
- *8: FXCPU cannot be used.
- *9: In a multi-QCPU configuration, an error will occur if the shared memory of the host QCPU is specified.
 Also, independently of the host or other CPU, an error will occur if write to the shared memory is performed.

(2) For CC-Link communication only

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

| Device | Device Name | Device No. | Type | Representation | Remarks |
|--|-------------|------------|------|----------------|---|
| Special relay | SM | Bit | | Decimal | Special relay of own board |
| Special register | SD | Word | | Decimal | Special register of own board |
| Link special register (for CC-Link) | SB | Bit | | Hexadecimal | Link special relay of own board |
| Link special register (for CC-Link) | SW | Word | | Hexadecimal | Link special register of own board |
| Remote input | X | Bit | | Hexadecimal | RX |
| Remote output | Y | Bit | | Hexadecimal | RY |
| Link register | W | Word | | Hexadecimal | — |
| Remote register (write area for CC-Link) | WW | Word | | Hexadecimal | RWw |
| Remote register (read area for CC-Link) | WR | Word | | Hexadecimal | RWr |
| Buffer memory | ML | Word | | Hexadecimal | Buffer memory of own station CC-Link module |
| Random access buffer | MC | Word | | Hexadecimal | Random access buffer in buffer memory of own station CC-Link module |
| Automatic refresh buffer | MF | Bit | | Hexadecimal | Automatic refresh buffer of own station CC-Link module |

(3) About device extension representation

The following table indicates whether the device extension representations are usable or not for the available CPUs.

They cannot be used with ReadDeviceBlock and WriteDeviceBlock.

When the ActAJ71E71TCP, ActMLAJ71E71TCP, ActAJ71QE71TCP or ActMLAJ71QE71TCP control is used, device expansion representation is unusable.

| Device Extension Representation | Target CPU | | | | | |
|---|---------------|---------------|--------|------|-------|-----------------------|
| | QCPU (Q mode) | QCPU (A mode) | QnACPU | ACPU | FXCPU | Motion controller CPU |
| Digit specification (example: K4M0) *2 | ○ | ○ | ○ | ○ | ○ | ○ |
| Bit specification (example: D0.1) *3 | ○ | ○ | ○ | ○ | ○ | ○ |
| Index qualification (example: M100Z0) *4 | ○ | × | ○*1 | × | × | × |

○: Usable ×: Unusable

*1: Unusable when QE71 is relayed.

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

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

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

2.4 Accessible Devices and Ranges

Refer to the MX Component operating manual for the accessible devices and ranges for corresponding communication.

3 DETAILS OF THE ACT CONTROLS

This chapter describes the details of the ACT controls, the details of the properties, and the possessed property list.

3.1 Details of the ACT Controls

The following table lists the definitions and usable setting types of the ACT controls.

| Control Name | | Definition | Usable Setting Type |
|-------------------|------------------|--|---------------------|
| For VB, VC++, VBA | For VBScript | | |
| ActEasyIF | ActMLEasyIF | Can communicate with any communication path. Use the communication settings utility to set the information for communication. | U |
| ActQJ71E71TCP | ActMLQJ71E71TCP | Used for Ethernet communication where the connected module is the Q series-compatible E71 (TCP/IP communication). | P |
| ActQJ71E71UDP | ActMLQJ71E71UDP | Used for Ethernet communication where the connected module is the Q series-compatible E71 (UDP/IP communication). | P |
| ActAJ71QE71TCP | ActMLAJ71QE71TCP | Used for Ethernet communication where the connected module is the QE71 (TCP/IP communication). | P |
| ActAJ71QE71UDP | ActMLAJ71QE71UDP | Used for Ethernet communication where the connected module is the QE71 (UDP/IP communication). | P |
| ActAJ71E71TCP | ActMLAJ71E71TCP | Used for Ethernet communication where the connected module is the E71 (TCP/IP communication). | P |
| ActAJ71E71UDP | ActMLAJ71E71UDP | Used for Ethernet communication where the connected module is the E71 (UDP/IP communication). | P |
| ActQCPUQ | ActMLQCPUQ | Used for CPU COM communication where the connected PLC CPU is the QCPU (Q mode). | P |
| ActQCPUA | ActMLQCPUA | Used for CPU COM communication where the connected PLC CPU is the QCPU (A mode). | P |
| ActQnACPU | ActMLQnACPU | Used for CPU COM communication where the connected PLC CPU is the QnACPU. | P |
| ActACPU | ActMLACPU | Used for CPU COM communication where the connected PLC CPU is the ACPU (including motion controller CPU). | P |
| ActFXCPU | ActMLFXCPU | Used for CPU COM communication where the connected PLC CPU is the FXCPU. | P |
| ActQJ71C24 | ActMLQJ71C24 | Used for computer link communication where the connected module is the Q series-compatible C24. | P |
| ActAJ71QC24 | ActMLAJ71QC24 | Used for computer link communication where the connected module is the QC24(N). | P |
| ActAJ71UC24 | ActMLAJ71UC24 | Used for computer link communication where the connected module is the UC24. | P |
| ActAJ71C24 | ActMLAJ71C24 | Used for computer link communication where the connected module is the C24. | P |
| ActQCPUQUSB | ActMLQCPUQUSB | Used for USB communication where the connected PLC CPU is the QCPU (Q mode). | P |
| ActCCG4QnA | ActMLCCG4QnA | Used for CC-Link G4 communication where the connected module is the AJ65BT-G4 (QnA mode). | P |
| ActCCG4A | ActMLCCG4A | Used for CC-Link G4 communication where the connected module is the AJ65BT-G4 (A mode). | P |
| ActMnet10BD | ActMLMnet10BD | Used for MELSECNET/10 communication. | P |
| ActMnetHBD | ActMLMnetHBD | Used for MELSECNET/H communication. | P |
| ActCCBD | ActMLCCBD | Used for CC-Link communication. | P |
| ActAnUBD | ActMLAnUBD | Used for CPU board communication. | P |
| ActLLt | ActMLLLt | Used for GX Simulator communication. | P |
| ActQCPUQBus | ActMLQCPUQBus | Used for Q series bus communication. | |

U : Utility setting type
P : Program setting type

3.2 Details of the Properties

The following tables give the details of the properties which must be set to create a user application.

| |
|---|
| POINT |
| When entering a property value directly into the property window of Visual Basic® or Visual C++®, change a character string such as a hexadecimal number or CPU type into a decimal property value. |

| Property Name (Type) | Description | | | | | | |
|-----------------------------------|---|-----------------------------------|-------------|------|--------------------------|------|---|
| ActLogicalCtationNumber (LONG) | Logical station number set on the communication settings utility. | | | | | | |
| ActNetworkNumber (LONG) | Specify the network number on the MELSENET/10(H). (Specify "0x00" when specifying the own station.) Specify as follows for multidrop connection (via Q series-compatible C24, QJ61BT11). <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>ActIntelligentPreferenceBit value</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0x00</td> <td style="text-align: center;">Specify the own network.</td> </tr> <tr> <td style="text-align: center;">0x01</td> <td style="text-align: center;">Specify another network of multidrop destination.</td> </tr> </tbody> </table> | ActIntelligentPreferenceBit value | Description | 0x00 | Specify the own network. | 0x01 | Specify another network of multidrop destination. |
| ActIntelligentPreferenceBit value | Description | | | | | | |
| 0x00 | Specify the own network. | | | | | | |
| 0x01 | Specify another network of multidrop destination. | | | | | | |
| ActStationNumber (LONG) | Specify the station number for MELSENET/10(H) or CC-Link. (Specify "0x00" when specifying the own station.) Handled as the own station when access to the CPU of the CPU board is made. Specify as follows for multidrop connection (via Q series-compatible C24, QJ61BT11). <table border="1" style="margin-left: 20px;"> <thead> <tr> <th>ActIntelligentPreferenceBit value</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td style="text-align: center;">0x00</td> <td style="text-align: center;">Specify the own network.</td> </tr> <tr> <td style="text-align: center;">0x01</td> <td style="text-align: center;">Specify another network of multidrop destination.</td> </tr> </tbody> </table> | ActIntelligentPreferenceBit value | Description | 0x00 | Specify the own network. | 0x01 | Specify another network of multidrop destination. |
| ActIntelligentPreferenceBit value | Description | | | | | | |
| 0x00 | Specify the own network. | | | | | | |
| 0x01 | Specify another network of multidrop destination. | | | | | | |
| ActUnitNumber (LONG) | Specify the module number of the computer link module or the station number when the target is the Q series-compatible intelligent special function module. However, specify "00x0" when setting the QnA series own station (module loaded to the own station CPU). Invalid when the target is not the computer link communication or Q series-compatible intelligent special function module. For multidrop link, specify the module number of the target computer link module. | | | | | | |
| ActConnectUnitNumber (LONG) | Specify the module number of the computer link module, QE71 or Q series-compatible E71. For multidrop link, specify the module number of the requesting computer link module. For multidrop link via CPU COM communication, however, the module number of the requesting station is not needed (specify "00x0"). Specify "0x00" for other than multidrop link. For the QE71 and Q series-compatible E71, specify the relay target station number (fixed to "0x00" for access within the own network). For access to another network via MELSENET/10, specify the station number set in the parameter of the connected Ethernet module. | | | | | | |
| ActIONumber (LONG) | Specify the module I/O number. For multidrop link or intelligent special function module access, specify the actual I/O number (first I/O number ÷ 16) of the target computer link module or intelligent special function module (specify the I/O number of the relayed or requesting station for multidrop link). Specify "0x3FF" when making access to another station via the own station CPU or network. | | | | | | |



| Property Name(Type) | Description | | | |
|----------------------|---|--|---|--|
| ActCpuType (LONG) | Specify the target CPU to communicate with. In the parameter, specify any of the CPU types in the following table. | | | |
| | Property value (Property window input value) | Target CPU | Property value (Property window input value) | Target CPU |
| | CPU_Q00JCPU (0x30) | Q00JCPU | CPU_A2SHCPU(0x109) | A2SHCPU (–S1) |
| | CPU_Q00CPU (0x31) | Q00CPU | CPU_A3NCPUCPU (0x10A) | A3NCPUCPU |
| | CPU_Q01CPU (0x32) | Q01CPU | CPU_A2ACPU (0x10C) | A2ACPU (–S1), A2ACPUP21/R21(–S1) |
| | CPU_Q02CPU (0x22) | Q02(H)CPU | CPU_A3ACPU (0x10D) | A3ACPU, A3ACPUP21/R21 |
| | CPU_Q06CPU (0x23) | Q06HCPU | CPU_A2UCPU (0x10E) | A2UCPU (–S1), A2USCPU (–S1), A2ASCPU (–S1) |
| | CPU_Q12CPU (0x24) | Q12HCPU | CPU_A2USHS1CPU (0x10F) | A2USHCPU-S1CPU, CPU board |
| | CPU_Q25CPU (0x25) | Q25HCPU | CPU_A3UCPU (0x110) | A3UCPU, A2ASCPU-S30 |
| | CPU_Q02CPU_A (0x141) | Q02(H)CPU-A | CPU_A4UCPU (0x111) | A4UCPU |
| | CPU_Q06CPU_A (0x142) | Q06HCPU-A | CPU_FX0CPU (0x201) | FX0, FX0s |
| | CPU_Q2ACPU (0x11) | Q2ACPU, Q2ASCPU, Q2ASHCPU | CPU_FX0NCPUCPU (0x202) | FX0N |
| | CPU_Q2AS1CPU (0x12) | Q2ACPU-S1, Q2ASCPU(–S1), Q2ASHCPU(–S1) | CPU_FX1CPU (0x203) | FX1 |
| | CPU_Q3ACPU (0x13) | Q3ACPU | CPU_FX2CPU (0x204) | FX2, FX2c |
| | CPU_Q4ACPU (0x14) | Q4ACPU, Q4ARCPU | CPU_FX2NCPUCPU (0x205) | FX2N, FX2NC |
| | CPU_A0J2HCPUCPU (0x102) | A0J2HCPUCPU | CPU_FX1SCPU (0x206) | FX1S |
| | CPU_A1FXCPUCPU (0x103) | A1FXCPUCPU | CPU_FX1NCPUCPU (0x207) | FX1N, FX1NC |
| | CPU_A1SCPU (0x104) | A1SCPU(–S1), A1SCPUC24-R2, A1SJCPU | CPU_A171SHCPUCPU (0x601) | A171SHCPUCPU |
| | CPU_A1SHCPUCPU (0x105) | A1SHCPUCPU, A1SJHCPUCPU | CPU_A172SHCPUCPU (0x602) | A172SHCPUCPU |
| | CPU_A1NCPUCPU (0x106) | A1NCPUCPU | CPU_A273UHCPUCPU (0x603) | A273UHCPUCPU (–S3) |
| | CPU_A2CCCPU (0x107) | A2CCCPU, A2CCPUC24 (–PRF), A2CJCPU | CPU_A173UHCPUCPU (0x604) | A173UHCPUCPU (–S1) |
| | CPU_A2NCPUCPU (0x108) | A2NCPUCPU (–S1), A2SCPU (–S1) | CPU_BOARD (0x401) | For own board access * 1 |

*1: Except CPU board

| Property Name(Type) | Description | | | | | | | | | | | | | | | | | | | | | | | | |
|---|--|---|-------------|---|----------------------|-----------------------|----------------------|-------------------------|----------------------|-----------------------|----------------------|---------------------------|----------------------|-------------------------|----------------------|---------------------------|----------------------|-------------------------|----------------------|---------------------------|----------------------|-------------------------|-----------------------|-----------------------------|-----------|
| ActPortNumber (LONG) | <p>Specify the connection port number of the IBM-PC/AT compatible.</p> <p>When the Ethernet module is connected, set any value as the port number of the requesting source (IBM-PC/AT compatible).</p> <p>When "=0" was specified as the port number, the MELSECNET/10 routing system should be the automatic response system. (When the system selected is other than the automatic response system via QE71, you should set the fixed value "5001".)</p> <p>Also, when the control for network board is used, specify the first board as PORT_1, and the second and subsequent boards as PORT_2, PORT_3 ...</p> <table border="1" data-bbox="571 622 1187 1084"> <thead> <tr> <th data-bbox="571 622 938 689">Property value (Property window input value)</th> <th data-bbox="938 622 1187 689">Description</th> </tr> </thead> <tbody> <tr><td data-bbox="571 689 938 734">PORT_1 (0x01)</td><td data-bbox="938 689 1187 734">Communication port 1</td></tr> <tr><td data-bbox="571 734 938 779">PORT_2 (0x02)</td><td data-bbox="938 734 1187 779">Communication port 2</td></tr> <tr><td data-bbox="571 779 938 824">PORT_3 (0x03)</td><td data-bbox="938 779 1187 824">Communication port 3</td></tr> <tr><td data-bbox="571 824 938 869">PORT_4 (0x04)</td><td data-bbox="938 824 1187 869">Communication port 4</td></tr> <tr><td data-bbox="571 869 938 913">PORT_5 (0x05)</td><td data-bbox="938 869 1187 913">Communication port 5</td></tr> <tr><td data-bbox="571 913 938 958">PORT_6 (0x06)</td><td data-bbox="938 913 1187 958">Communication port 6</td></tr> <tr><td data-bbox="571 958 938 1003">PORT_7 (0x07)</td><td data-bbox="938 958 1187 1003">Communication port 7</td></tr> <tr><td data-bbox="571 1003 938 1048">PORT_8 (0x08)</td><td data-bbox="938 1003 1187 1048">Communication port 8</td></tr> <tr><td data-bbox="571 1048 938 1093">PORT_9 (0x09)</td><td data-bbox="938 1048 1187 1093">Communication port 9</td></tr> <tr><td data-bbox="571 1093 938 1084">PORT_10 (0x0A)</td><td data-bbox="938 1093 1187 1084">Communication port 10</td></tr> </tbody> </table> | Property value (Property window input value) | Description | PORT_1 (0x01) | Communication port 1 | PORT_2 (0x02) | Communication port 2 | PORT_3 (0x03) | Communication port 3 | PORT_4 (0x04) | Communication port 4 | PORT_5 (0x05) | Communication port 5 | PORT_6 (0x06) | Communication port 6 | PORT_7 (0x07) | Communication port 7 | PORT_8 (0x08) | Communication port 8 | PORT_9 (0x09) | Communication port 9 | PORT_10 (0x0A) | Communication port 10 | | |
| Property value (Property window input value) | Description | | | | | | | | | | | | | | | | | | | | | | | | |
| PORT_1 (0x01) | Communication port 1 | | | | | | | | | | | | | | | | | | | | | | | | |
| PORT_2 (0x02) | Communication port 2 | | | | | | | | | | | | | | | | | | | | | | | | |
| PORT_3 (0x03) | Communication port 3 | | | | | | | | | | | | | | | | | | | | | | | | |
| PORT_4 (0x04) | Communication port 4 | | | | | | | | | | | | | | | | | | | | | | | | |
| PORT_5 (0x05) | Communication port 5 | | | | | | | | | | | | | | | | | | | | | | | | |
| PORT_6 (0x06) | Communication port 6 | | | | | | | | | | | | | | | | | | | | | | | | |
| PORT_7 (0x07) | Communication port 7 | | | | | | | | | | | | | | | | | | | | | | | | |
| PORT_8 (0x08) | Communication port 8 | | | | | | | | | | | | | | | | | | | | | | | | |
| PORT_9 (0x09) | Communication port 9 | | | | | | | | | | | | | | | | | | | | | | | | |
| PORT_10 (0x0A) | Communication port 10 | | | | | | | | | | | | | | | | | | | | | | | | |
| ActBaudRate (LONG) | <p>Specify the baudrate for computer link communication.</p> <table border="1" data-bbox="459 1178 1410 1608"> <thead> <tr> <th data-bbox="459 1178 699 1279">Property value (Property window input value)</th> <th data-bbox="699 1178 938 1279">Description</th> <th data-bbox="938 1178 1177 1279">Property value (Property window input value)</th> <th data-bbox="1177 1178 1410 1279">Description</th> </tr> </thead> <tbody> <tr> <td data-bbox="459 1279 699 1346">BAUDRATE_300 (300)</td> <td data-bbox="699 1279 938 1346">300bps</td> <td data-bbox="938 1279 1177 1346">BAUDRATE_9600 (9600)</td> <td data-bbox="1177 1279 1410 1346">9600bps</td> </tr> <tr> <td data-bbox="459 1346 699 1413">BAUDRATE_600 (600)</td> <td data-bbox="699 1346 938 1413">600bps</td> <td data-bbox="938 1346 1177 1413">BAUDRATE_19200 (19200)</td> <td data-bbox="1177 1346 1410 1413">19200bps</td> </tr> <tr> <td data-bbox="459 1413 699 1480">BAUDRATE_1200 (1200)</td> <td data-bbox="699 1413 938 1480">1200bps</td> <td data-bbox="938 1413 1177 1480">BAUDRATE_38400 (38400)</td> <td data-bbox="1177 1413 1410 1480">38400bps</td> </tr> <tr> <td data-bbox="459 1480 699 1547">BAUDRATE_2400 (2400)</td> <td data-bbox="699 1480 938 1547">2400bps</td> <td data-bbox="938 1480 1177 1547">BAUDRATE_57600 (57600)</td> <td data-bbox="1177 1480 1410 1547">57600bps</td> </tr> <tr> <td data-bbox="459 1547 699 1608">BAUDRATE_4800 (4800)</td> <td data-bbox="699 1547 938 1608">4800bps</td> <td data-bbox="938 1547 1177 1608">BAUDRATE_115200 (115200)</td> <td data-bbox="1177 1547 1410 1608">115200bps</td> </tr> </tbody> </table> | Property value (Property window input value) | Description | Property value (Property window input value) | Description | BAUDRATE_300 (300) | 300bps | BAUDRATE_9600 (9600) | 9600bps | BAUDRATE_600 (600) | 600bps | BAUDRATE_19200 (19200) | 19200bps | BAUDRATE_1200 (1200) | 1200bps | BAUDRATE_38400 (38400) | 38400bps | BAUDRATE_2400 (2400) | 2400bps | BAUDRATE_57600 (57600) | 57600bps | BAUDRATE_4800 (4800) | 4800bps | BAUDRATE_115200 (115200) | 115200bps |
| Property value (Property window input value) | Description | Property value (Property window input value) | Description | | | | | | | | | | | | | | | | | | | | | | |
| BAUDRATE_300 (300) | 300bps | BAUDRATE_9600 (9600) | 9600bps | | | | | | | | | | | | | | | | | | | | | | |
| BAUDRATE_600 (600) | 600bps | BAUDRATE_19200 (19200) | 19200bps | | | | | | | | | | | | | | | | | | | | | | |
| BAUDRATE_1200 (1200) | 1200bps | BAUDRATE_38400 (38400) | 38400bps | | | | | | | | | | | | | | | | | | | | | | |
| BAUDRATE_2400 (2400) | 2400bps | BAUDRATE_57600 (57600) | 57600bps | | | | | | | | | | | | | | | | | | | | | | |
| BAUDRATE_4800 (4800) | 4800bps | BAUDRATE_115200 (115200) | 115200bps | | | | | | | | | | | | | | | | | | | | | | |
| ActDataBit(LONG) | Specify the number of bits (7 or 8) of the byte data sent and received for computer link communication. | | | | | | | | | | | | | | | | | | | | | | | | |
| ActParity (LONG) | <p>Specify the parity system used for computer link communication.</p> <table border="1" data-bbox="571 1760 1187 1951"> <thead> <tr> <th data-bbox="571 1760 938 1827">Property value (Property window input value)</th> <th data-bbox="938 1760 1187 1827">Description</th> </tr> </thead> <tbody> <tr><td data-bbox="571 1827 938 1872">NO_PARITY (0)</td><td data-bbox="938 1827 1187 1872">No parity</td></tr> <tr><td data-bbox="571 1872 938 1917">ODD_PARITY (1)</td><td data-bbox="938 1872 1187 1917">Odd</td></tr> <tr><td data-bbox="571 1917 938 1951">EVEN_PARITY (2)</td><td data-bbox="938 1917 1187 1951">Even</td></tr> </tbody> </table> | Property value (Property window input value) | Description | NO_PARITY (0) | No parity | ODD_PARITY (1) | Odd | EVEN_PARITY (2) | Even | | | | | | | | | | | | | | | | |
| Property value (Property window input value) | Description | | | | | | | | | | | | | | | | | | | | | | | | |
| NO_PARITY (0) | No parity | | | | | | | | | | | | | | | | | | | | | | | | |
| ODD_PARITY (1) | Odd | | | | | | | | | | | | | | | | | | | | | | | | |
| EVEN_PARITY (2) | Even | | | | | | | | | | | | | | | | | | | | | | | | |

| Property Name(Type) | Description | | | | | | | | | | |
|---|---|---|-------------|------------------|------------------|----------------------------------|-----------------|----------------------------------|-----------------------------|-----------------------|----------------------------|
| ActStopBit (LONG) | Specify the number of stop bits used for computer link communication <table border="1"> <thead> <tr> <th>Property value (Property window input value)</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>STOPBIT_ONE (0)</td> <td>1 stop bit</td> </tr> <tr> <td>STOPBITS_TWO (2)</td> <td>2 stop bits</td> </tr> </tbody> </table> | Property value (Property window input value) | Description | STOPBIT_ONE (0) | 1 stop bit | STOPBITS_TWO (2) | 2 stop bits | | | | |
| Property value (Property window input value) | Description | | | | | | | | | | |
| STOPBIT_ONE (0) | 1 stop bit | | | | | | | | | | |
| STOPBITS_TWO (2) | 2 stop bits | | | | | | | | | | |
| ActControl (LONG) | Specify the control setting of the signal line. <table border="1"> <thead> <tr> <th>Property value (Property window input value)</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>TRC_DTR (0x01)</td> <td>DTR control</td> </tr> <tr> <td>TRC_RTS (0x02)</td> <td>RTS control</td> </tr> <tr> <td>TRC_DRT_AND_RTS (0x07)</td> <td>DTR control and RTS control</td> </tr> <tr> <td>TRC_DTR_OR_RTS (0x08)</td> <td>DTR control or RTS control</td> </tr> </tbody> </table> | Property value (Property window input value) | Description | TRC_DTR (0x01) | DTR control | TRC_RTS (0x02) | RTS control | TRC_DRT_AND_RTS (0x07) | DTR control and RTS control | TRC_DTR_OR_RTS (0x08) | DTR control or RTS control |
| Property value (Property window input value) | Description | | | | | | | | | | |
| TRC_DTR (0x01) | DTR control | | | | | | | | | | |
| TRC_RTS (0x02) | RTS control | | | | | | | | | | |
| TRC_DRT_AND_RTS (0x07) | DTR control and RTS control | | | | | | | | | | |
| TRC_DTR_OR_RTS (0x08) | DTR control or RTS control | | | | | | | | | | |
| ActHostAddress(BSTR) | Pointer which indicates the connection host name (IP address) for Ethernet communication. | | | | | | | | | | |
| ActCpuTimeOut(LONG) | Specify the CPU watchdog timer for Ethernet communication. (Unit = "×250ms") | | | | | | | | | | |
| ActTimeOut(LONG) | Set the time-out value of communication between the IBM-PC/AT compatible and PLC. (Unit = "ms") For MX Component, depending on the communication path, the internal processing of time-out may be repeated 3 times. In this case, it may take 3 times the time-out value that has been set. | | | | | | | | | | |
| ActSumCheck (LONG) | Specify whether sumcheck is made or not. Valid only via computer link module. <table border="1"> <thead> <tr> <th>Property value (Property window input value)</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>NO_SUM_CHECK (0)</td> <td>Without sumcheck</td> </tr> <tr> <td>SUM_CHECK (1)</td> <td>With sumcheck</td> </tr> </tbody> </table> | Property value (Property window input value) | Description | NO_SUM_CHECK (0) | Without sumcheck | SUM_CHECK (1) | With sumcheck | | | | |
| Property value (Property window input value) | Description | | | | | | | | | | |
| NO_SUM_CHECK (0) | Without sumcheck | | | | | | | | | | |
| SUM_CHECK (1) | With sumcheck | | | | | | | | | | |
| ActSourceNetworkNumber (LONG) | Specify the requesting network number when the QE71 or Q series-compatible E71 is specified. Specify the same network number as for the connected QE71 or Q series-compatible E71 (network number specified in the network parameter). | | | | | | | | | | |
| ActSourceStationNumber (LONG) | Specify the requesting station number (IBM-PC/AT compatible side station number) when the QE71 or Q series-compatible E71 is specified. Make setting to avoid setting the same station number as that of the QE71 set within the same Ethernet loop. | | | | | | | | | | |
| ActDestinationPort Number (LONG) | Specify the port number of the target when Ethernet communication is specified. For access to another network, specify the relay destination port number. For other than the automatic response system, make setting as indicated in the following table. <table border="1"> <thead> <tr> <th>Communication</th> <th>Setting</th> </tr> </thead> <tbody> <tr> <td>QE71(UDP/IP)</td> <td>Fixed to "5001"</td> </tr> <tr> <td>Q series-compatible E71 (TCP/IP)</td> <td>Fixed to "5002"</td> </tr> <tr> <td>Q series-compatible E71 (UDP/IP)</td> <td>Fixed to "5001"</td> </tr> </tbody> </table> | Communication | Setting | QE71(UDP/IP) | Fixed to "5001" | Q series-compatible E71 (TCP/IP) | Fixed to "5002" | Q series-compatible E71 (UDP/IP) | Fixed to "5001" | | |
| Communication | Setting | | | | | | | | | | |
| QE71(UDP/IP) | Fixed to "5001" | | | | | | | | | | |
| Q series-compatible E71 (TCP/IP) | Fixed to "5002" | | | | | | | | | | |
| Q series-compatible E71 (UDP/IP) | Fixed to "5001" | | | | | | | | | | |
| ActDestinationIONumber (LONG) | For multidrop connection (via Q series-compatible C24/CC-Link), specify the actual I/O number (first I/O÷16) of the last access target station. (When the target is the intelligent special function module) When the target is the CPU, specify "0x3FF". | | | | | | | | | | |

| Property Name(Type) | Description | | | | | | |
|---|--|---|-------------|------------------|--|-------------------|---|
| ActMultiDropChannel Number (LONG) | For multidrop connection (via Q series-compatible C24/CC-Link), specify the multidrop connection channel number (Ch1/Ch2). Invalid for other connections. | | | | | | |
| ActThroughNetworkType (LONG) | You can select the MELSECNET/H or MELSECNET/10 mode to make access to the own station QCPOU (Q mode) or to the QCPU (Q mode) via the MELSECNET/H when using the ActQJ71C24, ActQJ71E71TCP, ActQJ71E71UDP, ActQCPUQ or ActQCPUQUSB control. When the control used is other than the above, the mode is fixed to the MELSECNET/10 mode. <table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th>Property value</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0x00</td> <td>MELSECNET/H mode</td> </tr> <tr> <td>0x01</td> <td>MELSECNET/10 mode</td> </tr> </tbody> </table> | Property value | Description | 0x00 | MELSECNET/H mode | 0x01 | MELSECNET/10 mode |
| Property value | Description | | | | | | |
| 0x00 | MELSECNET/H mode | | | | | | |
| 0x01 | MELSECNET/10 mode | | | | | | |
| ActIntelligent PreferenceBit (LONG) | For multidrop connection (via Q series-compatible C24/CC-Link), specify whether the network of the multidrop link destination will be relayed or not. (To differentiate the own network module.) <table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th>Property value</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0x00</td> <td>Another network of multidrop link destination is not accessed.</td> </tr> <tr> <td>0x01</td> <td>Another network of multidrop link destination is accessed.</td> </tr> </tbody> </table> | Property value | Description | 0x00 | Another network of multidrop link destination is not accessed. | 0x01 | Another network of multidrop link destination is accessed. |
| Property value | Description | | | | | | |
| 0x00 | Another network of multidrop link destination is not accessed. | | | | | | |
| 0x01 | Another network of multidrop link destination is accessed. | | | | | | |
| ActDidPropertyBit (LONG) | For access to the Q series-compatible own station intelligent special function module (intelligent special function module load on the own station CPU), making the following setting invalid makes it unnecessary to specify "ActUnitNumber". (Only "ActIONumber" is used to specify the module I/O number.) <table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th>Property value</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0x00</td> <td>Module number is made valid.</td> </tr> <tr> <td>0x01</td> <td>Module number is made invalid.</td> </tr> </tbody> </table> | Property value | Description | 0x00 | Module number is made valid. | 0x01 | Module number is made invalid. |
| Property value | Description | | | | | | |
| 0x00 | Module number is made valid. | | | | | | |
| 0x01 | Module number is made invalid. | | | | | | |
| ActDsidPropertyBit (LONG) | For multidrop connection (via Q series-compatible C24/CC-Link), making the following setting invalid makes it unnecessary to specify "ActDestinationIONumber". However, when the following setting is made invalid, "ActDidPropertyBit" must be made valid. (Use "ActUnitNumber" to specify.) <table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th>Property value</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>0x00</td> <td>I/O number of the last access target station is made valid.</td> </tr> <tr> <td>0x01</td> <td>I/O number of the last access target station is made invalid.</td> </tr> </tbody> </table> | Property value | Description | 0x00 | I/O number of the last access target station is made valid. | 0x01 | I/O number of the last access target station is made invalid. |
| Property value | Description | | | | | | |
| 0x00 | I/O number of the last access target station is made valid. | | | | | | |
| 0x01 | I/O number of the last access target station is made invalid. | | | | | | |
| ActPacketType (LONG) | Specify the packet type for communication with the A series or QnA series Ethernet module. <table border="1" style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th>Property value (Property window input value)</th> <th>Description</th> </tr> </thead> <tbody> <tr> <td>PACKET_ASCII (2)</td> <td>ASCII packet</td> </tr> <tr> <td>PACKET_BINARY (3)</td> <td>Binary packet</td> </tr> </tbody> </table> | Property value (Property window input value) | Description | PACKET_ASCII (2) | ASCII packet | PACKET_BINARY (3) | Binary packet |
| Property value (Property window input value) | Description | | | | | | |
| PACKET_ASCII (2) | ASCII packet | | | | | | |
| PACKET_BINARY (3) | Binary packet | | | | | | |
| ActPassword (BSTR) | Specify the password set to the Q series-compatible E71 to unlock that password. It is ignored if the communication target is other than the Q series-compatible E71 when the ActEasyIF or ActMLEasyIF control is used. Specifying any characters other than alphanumeric characters will result in a character code conversion error (0xF1000001) at the Open time. | | | | | | |

3.3 Lists of Properties Possessed by the ACT Controls

This section lists the properties possessed by the ACT controls and their default values. How to use the manual in Section 3.3.1 to Section 3.3.25 is provided below.

<How to use the manual in Section 3.3.1 to Section 3.3.25>

Configuration
Sketch of system configuration

3 DETAILS OF THE ACT CONTROLS MELSOFT

3.3.2 ActQJ71E71TCP, ActMLQJ71E71TCP control

The following table indicates the properties possessed by the ActQJ71E71TCP, ActMLQJ71E71TCP control and their default values.

(1) Configuration

```

    graph TD
      A[Connected station CPU] --- B[Q series compatible E71]
      A --- C[Relayed module]
      B --- D[Relayed network]
      C --- D
      D --- E[Relayed station CPU module]
      F[IBM-PC/AT compatible] --- G[Relayed station CPU module]
      G --- E
    
```

(2) Property patterns

| | Connected Station CPU QCPU (Q mode) | Relayed Station CPU | | | | |
|---------------|-------------------------------------|---------------------|---------------|---------------|---------|---------|
| | | Relayed Network | QCPU (Q mode) | QCPU (A mode) | QnA CPU | ACPU #1 |
| MELSECNET/10 | ② | × | × | × | × | × |
| MELSECNET/10 | ② | ② | ② | ② | ② | ② |
| MELSECNET/II | × | × | × | × | × | × |
| Ethernet | ② | × | ② | × | × | × |
| Computer link | ③ | × | × | × | × | × |
| CC-Link | ④ | × | × | × | × | × |

○ : Accessible (Property pattern within circle)
 × : Inaccessible
 *1 : Including motion controller CPU

(3) Property list

| Property | Default Value | Property Patterns | | | |
|-------------------------|-----------------|--|--|--|--|
| | | ① | ② | ③ | ④ |
| ActConnectUnitNumber #1 | 0 (0x00) | Fixed to 0x00 | Connected station side module station number | Fixed to 0x00 | Fixed to 0x00 |
| ActCpuType | 34 (CPU_Q02CPU) | CPU type corresponding to target station | | | |
| ActDestinationQNumber | 0 (0x00) | Fixed to 0x00 | Fixed to 0x00 | Target station side For single CPU 0x3FF fixed | Target station side For single CPU 0x3FF fixed |
| | | | | For multiple CPUs Connected CPU 0x3FF | For multiple CPUs Connected CPU 0x3FF |

Property patterns
Indicates the accessible ranges of the used control and the patterns of the properties.

Property list
(1) Property
Gives the property name.

(2) Default value
 Gives the default value of the property.
 The default values used when the properties are changed in the program are given within the "parentheses".

(3) Property pattern
 Gives the property settings necessary to make communication settings.
 Refer to the "property pattern table" for the property pattern numbers.

POINT

The default values indicated are the property values shown in the property window of Visual Basic® or Visual C++® .

The default values of the properties, whose values must be changed in other than decimal when changed in a program, are indicated in parentheses.

3.3.1 ActEasyIF, ActMLEasyIF control

The following table indicates the property possessed by the ActEasyIF, ActMLEasyIF control and its default value.

| Property | Default Value | Property Pattern |
|-------------------------|---------------|---|
| ActLogicalStationNumber | 0 | Logical station number set on the communication settings utility |
| ActPassword | Empty | Password set to the Q series-compatible E71 on the connected station side |

POINT

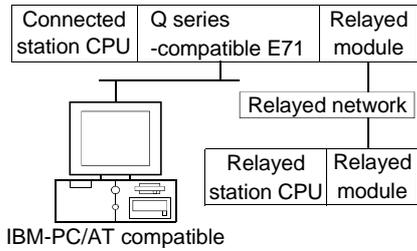
Depending on the communication path (Ethernet communication, MELSECNET/10 communication, etc.), there will be restrictions as placed on the corresponding communication path controls.

For restrictions, refer to the corresponding communication path controls (Sections 3.3.2 to 3.3.25).

3.3.2 ActQJ71E71TCP, ActMLQJ71E71TCP control

The following table indicates the properties possessed by the ActQJ71E71TCP, ActMLQJ71E71TCP control and their default values.

(1) Configuration



(2) Property patterns

| Connected Station CPU | Relayed Network | Relayed Station CPU | | | | |
|-----------------------|-----------------|---------------------|---------------|---------|---------|-------|
| | | QCPU (Q mode) | QCPU (A mode) | QnA CPU | ACPU *1 | FXCPU |
| ① | MELSECNET/H | ② | × | × | × | × |
| | MELSECNET/10 | ② | ② | ② | ② | × |
| | MELSECNET(II) | × | × | × | × | × |
| | Ethernet | ② | × | ② | × | × |
| | Computer link | ③ | × | × | × | × |
| | CC-Link | ④ | × | × | × | × |

○ : Accessible (Property pattern within circle)

× : Inaccessible

*1 : Including motion controller CPU

(3) Property list

| Property | Default Value | Property Patterns | | | |
|-------------------------|-----------------|--|---|---|---|
| | | ① | ② | ③ | ④ |
| ActConnectUnitNumber *1 | 0 (0x00) | Fixed to 0x00 | Connected station side module station number | Fixed to 0x00 | Fixed to 0x00 |
| ActCpuType | 34 (CPU_Q02CPU) | CPU type corresponding to target station | | | |
| ActDestinationIONumber | 0 (0x00) | Fixed to 0x00 | Fixed to 0x00 | Target station side For single CPU 0x3FF fixed For multiple CPUs Connected CPU: 0x3FF No. 1: 0x3E0 No. 2: 0x3E1 No. 3: 0x3E2 No. 4: 0x3E3 | Target station side For single CPU 0x3FF fixed For multiple CPUs Connected CPU: 0x3FF No. 1: 0x3E0 No. 2: 0x3E1 No. 3: 0x3E2 No. 4: 0x3E3 |
| ActDidPropertyBit | 1 (0x01) | Fixed to 0x01 | Fixed to 0x01 | Fixed to 0x00 | Fixed to 0x00 |
| ActDsidPropertyBit | 1 (0x01) | Fixed to 0x01 | Fixed to 0x01 | Fixed to 0x00 | Fixed to 0x00 |
| ActHostAddress | 1.1.1.1 | Host name or IP address of connected station side module | | | |
| ActIONumber *6 | 1023 (0x3FF) | For single CPU 0x3FF fixed For multiple CPUs Connected CPU: 0x3FF No. 1: 0x3E0 No. 2: 0x3E1 No. 3: 0x3E2 No. 4: 0x3E3 | Target station side For single CPU 0x3FF fixed For multiple CPUs Connected CPU: 0x3FF No. 1: 0x3E0 No. 2: 0x3E1 No. 3: 0x3E2 No. 4: 0x3E3 | Connected station side relayed module I/O address | Connected station side relayed module I/O address |

*1: For access to another station via MELSECNET/10 (for the property pattern of ②), specify the station number of the connected station side Q series-compatible E71 set in the Ethernet parameter of the connected station side Q series-compatible E71.

*6: As the I/O address, specify the value found by dividing the actual first I/O number by 16.

(To next page)

| Property | Default Value | Property Patterns | | | |
|---------------------------|---------------|--|--|---|---|
| | | ① | ② | ③ | ④ |
| ActMultiDropChannelNumber | 0 (0x00) | Fixed to 0x00 | Fixed to 0x00 | 0x00 or 0x02 *5 | Fixed to 0x00 |
| ActNetworkNumber *2 | 1 (0x01) | Network number of target station side module | Network number of target station side module | Connected station side Q series-compatible E71 network number | Connected station side Q series-compatible E71 network number |
| ActPassword | Empty | Password set to the Q series-compatible E71 on the connected station side | | | |
| ActSourceNetworkNumber *3 | 1 (0x01) | IBM-PC/AT compatible side network number | | | |
| ActSourceStationNumber *4 | 2 (0x02) | IBM-PC/AT compatible side station number | | | |
| ActStationNumber *2 | 1 (0x01) | Connected station side module station number | Connected 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) | QCPU (Q mode): 0x00 (MELSECNET/H only), other than QCPU (Q mode): 0x01 (including MELSECNET/10). Note that the setting must be the same as set in the network parameter of the GPP function. | | | |
| ActTimeOut | 10000 | Any value specified by user in ms units. | | | |
| ActUnitNumber | 0 (0x00) | Fixed to 0x00 | Fixed to 0x00 | Target station side module station number | Target station side module station number (valid) |

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

*3: Specify the same network number as the MELSECNET/10 network number set to the Q series-compatible E71 in the Ethernet parameter setting of the target station side Q series-compatible E71.

*4: Specify the station number on the IBM-PC/AT compatible side to avoid setting the same station number as set to the Q series-compatible E71 within the same Ethernet loop.

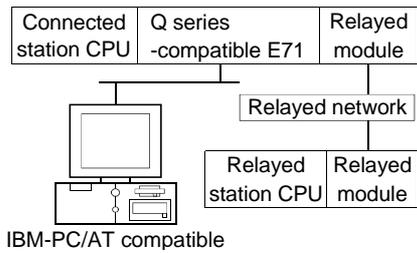
*5: If the following conditions are all satisfied, change the ActMultiDropChannelNumber value to 0x00.

- A remote password has been set to the Q series-compatible E71 connected.
- The version of the Q series-compatible E71 connected is J or earlier.

3.3.3 ActQJ71E71UDP, ActMLQJ71E71UDP control

The following table indicates the properties possessed by the ActQJ71E71UDP, ActMLQJ71E71UDP control and their default values.

(1) Configuration



(2) Property patterns

| Connected Station CPU | Relayed Network | Relayed Station CPU | | | | |
|-----------------------|-----------------|---------------------|---------------|---------|---------|-------|
| | | QCPU (Q mode) | QCPU (A mode) | QnA CPU | ACPU *1 | FXCPU |
| ① | MELSECNET/H | ② | × | × | × | × |
| | MELSECNET/10 | ② | ② | ② | ② | × |
| | MELSECNET(II) | × | × | × | × | × |
| | Ethernet | ② | × | ② | × | × |
| | Computer link | ③ | × | × | × | × |
| | CC-Link | ④ | × | × | × | × |

○ : Accessible (Property pattern within circle)
 × : Inaccessible
 *1 : Including motion controller CPU

(3) Property list

| Property | Default Value | Property Patterns | | | |
|-------------------------|-----------------|---|---|---|---|
| | | ① | ② | ③ | ④ |
| ActConnectUnitNumber *1 | 0 (0x00) | Fixed to 0x00 | Connected station side module station number | Fixed to 0x00 | Fixed to 0x00 |
| ActCpuType | 34 (CPU_Q02CPU) | CPU type corresponding to target station | | | |
| ActDestinationIONumber | 0 (0x00) | Fixed to 0x00 | Fixed to 0x00 | Target station side For single CPU 0x3FF fixed For multiple CPUs Connected CPU: 0x3FF No. 1: 0x3E0 No. 2: 0x3E1 No. 3: 0x3E2 No. 4: 0x3E3 | Target station side For single CPU 0x3FF fixed For multiple CPUs Connected CPU: 0x3FF No. 1: 0x3E0 No. 2: 0x3E1 No. 3: 0x3E2 No. 4: 0x3E3 |
| ActDidPropertyBit | 1 (0x01) | Fixed to 0x01 | Fixed to 0x01 | Fixed to 0x00 | Fixed to 0x00 |
| ActDsidPropertyBit | 1 (0x01) | Fixed to 0x01 | Fixed to 0x01 | Fixed to 0x00 | Fixed to 0x00 |
| ActHostAddress | 1.1.1.1 | Host name or IP address of connected station side module | | | |
| ActIONumber *7 | 1023 (0x3FF) | For single CPU 0x3FF fixed For multiple CPUs Connected CPU: 0x3FF No. 1: 0x3E0 No. 2: 0x3E1 No. 3: 0x3E2 No. 4: 0x3E3 | Target station side For single CPU 0x3FF fixed For multiple CPUs Connected CPU: 0x3FF No. 1: 0x3E0 No. 2: 0x3E1 No. 3: 0x3E2 No. 4: 0x3E3 | Connected station side relayed module I/O address | Connected station side relayed module I/O address |

*1: For access to another station via MELSECNET/10 (for the property pattern of ②), specify the station number of the connected station side Q series-compatible E71 set in the Ethernet parameter of the connected station side Q series-compatible E71.

*7: As the I/O address, specify the value found by dividing the actual first I/O number by 16.

(To next page)

| Property | Default Value | Property Patterns | | | |
|----------------------------|---------------|--|--|---|---|
| | | ① | ② | ③ | ④ |
| ActMultiDropChannelNumber | 0 (0x00) | Fixed to 0x00 | Fixed to 0x00 | 0x00 or 0x02 * 6 | Fixed to 0x00 |
| ActNetworkNumber * 2 | 1 (0x01) | Network number of target station side module | Network number of target station side module | Connected station side Q series-compatible E71 network number | Connected station side Q series-compatible E71 network number |
| ActPassword | Empty | Password set to the Q series-compatible E71 on the connected station side | | | |
| ActPortNumber * 4 | 5001 | IBM-PC/AT compatible side port number | | | |
| ActSourceNetworkNumber * 3 | 1 (0x01) | IBM-PC/AT compatible side network number | | | |
| ActSourceStationNumber * 5 | 2 (0x02) | IBM-PC/AT compatible side station number | | | |
| ActStationNumber * 2 | 1 (0x01) | 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) | QCPU (Q mode): 0x00 (MELSECNET/H only), other than QCPU (Q mode): 0x01 (including MELSECNET/10). Note that the setting must be the same as set in the network parameter of the GPP function. | | | |
| ActTimeOut | 10000 | Any value specified by user in ms units. | | | |
| ActUnitNumber | 0 (0x00) | Fixed to 0x00 | Fixed to 0x00 | Target station side module station number | Target station side module station number |

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

* 3: Specify the same network number as the MELSECNET/10 network number set to the Q series-compatible E71 in the Ethernet parameter setting of the target station side Q series-compatible E71.

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

* 5: Specify the station number on the IBM-PC/AT compatible side to avoid setting the same station number as set to the Q series-compatible E71 within the same Ethernet loop.

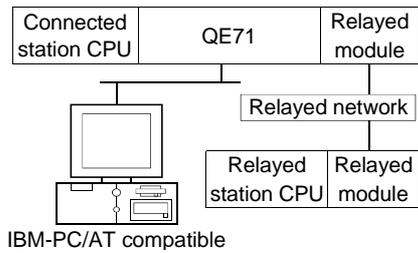
* 6: If the following conditions are all satisfied, change the ActMultiDropChannelNumber value to 0x00.

- A remote password has been set to the Q series-compatible E71 connected.
- The version of the Q series-compatible E71 connected is J or earlier.

3.3.4 ActAJ71QE71TCP, ActMLAJ71QE71TCP control

The following table indicates the properties possessed by the ActAJ71QE71TCP, ActMLAJ71QE71TCP control and their default values.

(1) Configuration



(2) Property patterns

| Connected Station CPU | Relayed Network | Relayed Station CPU | | | | |
|-----------------------|-----------------|---------------------|---------------|---------------|---------|---------|
| | | QnACPU | QCPU (Q mode) | QCPU (A mode) | QnA CPU | ACPU *1 |
| ① | MELSECNET/H | × | × | × | × | × |
| | MELSECNET/10 | × | × | ② | × | × |
| | MELSECNET(II) | × | × | × | × | × |
| | Ethernet | × | × | × | × | × |
| | Computer link | × | × | × | × | × |
| | CC-Link | × | × | × | × | × |

○ : Accessible (Property pattern within circle)

× : Inaccessible

*1 : Including motion controller CPU

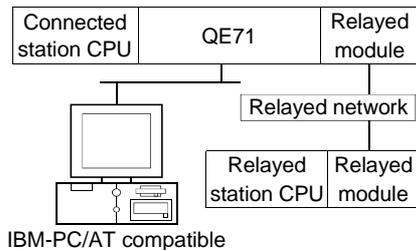
(3) Property list

| Property | Default Value | Property Patterns | |
|--------------------------|---------------------|--|---|
| | | ① | ② |
| ActCpuTimeOut | 40 | Any value specified by user in 250ms units | |
| ActCpuType | 17 (CPU_Q2ACPU) | CPU type corresponding to target station | |
| ActDestinationPortNumber | 1280 (0x500) | Port number of connected station side module | |
| ActHostAddress | 1.1.1.1 | Host name or IP address of connected station side module | |
| ActNetworkNumber | 0 (0x00) | 0x00 | Target station side module network number |
| ActPacketType | 2 (PACKET_ASCII) | PACKET_BINARY or PACKET_ASCII | |
| ActStationNumber | 255 (0xFF) | 0xFF | Target station side module station number |
| ActTimeOut | 10000 | Any value specified by user in ms units | |

3.3.5 ActAJ71QE71UDP, ActMLAJ71QE71UDP control

The following table indicates the properties possessed by the ActAJ71QE71UDP, ActMLAJ71QE71UDP control and their default values.

(1) Configuration



(2) Property patterns

| Connected Station CPU | Relayed Network | Relayed Station CPU | | | | |
|-----------------------|-----------------|---------------------|---------------|---------|---------|-------|
| | | QCPU (Q mode) | QCPU (A mode) | QnA CPU | ACPU *1 | FXCPU |
| ① | MELSECNET/H | × | × | × | × | × |
| | MELSECNET/10 | × | × | ② | × | × |
| | MELSECNET(II) | × | × | × | × | × |
| | Ethernet | × | × | ② | × | × |
| | Computer link | × | × | ③ | × | × |
| | CC-Link | × | × | × | × | × |

○ : Accessible (Property pattern within circle)

× : Inaccessible

* 1 : Including motion controller CPU

(3) Property list

| Property | Default Value | Property Patterns | | |
|----------------------------|-----------------|--|--|---|
| | | ① | ② | ③ |
| ActConnectUnitNumber * 1 | 0 (0x00) | Fixed to 0x00 | Connected station side module station number | Fixed to 0x00 |
| ActCpuType | 17 (CPU_Q2ACPU) | CPU type corresponding to target station | | |
| ActHostAddress | 1.1.1.1 | Host name or IP address of connected station side module | | |
| ActIONumber * 7 | 1023 (0x3FF) | Fixed to 0x3FF | Fixed to 0x3FF | Connected station side relayed module I/O address |
| ActNetworkNumber * 2 | 1 (0x01) | Target station side module network number | Target station side module network number | Connected station side QE71 network number |
| ActPortNumber * 3 * 6 | 5001 | IBM-PC/AT compatible side port number | | |
| ActSourceNetworkNumber * 4 | 1 (0x01) | IBM-PC/AT compatible side network number | | |
| ActSourceStationNumber * 5 | 2 (0x02) | IBM-PC/AT compatible side station number | | |
| ActStationNumber * 2 | 1 (0x01) | Target station side module station number | Target station side module station number | Connected station side QE71 station number |
| ActHostAddress | 1.1.1.1 | Host name or IP address of connected station side module | | |
| ActTimeOut | 10000 | Any value specified by user in ms units | | |
| ActUnitNumber | 0 (0x00) | Fixed to 0x00 | Fixed to 0x00 | Target station side module station number |

* 1: For access to another station via MELSECNET/10 (for the property pattern of ②), specify the station number of the connected station side QE71 set in the Ethernet parameter of the connected station side QE71.

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

* 3: Specify fixed "5001" when the Ethernet parameter setting of the connected station side QE71 is other than the "automatic response system". Specify fixed "0" when the Ethernet parameter setting of the connected station side QE71 is the "automatic response system".

* 4: Specify the same network number as the MELSECNET/10 network number set to the QE71 in the Ethernet parameter setting of the target station side QE71.

* 5: Specify the station number on the IBM-PC/AT compatible side to avoid setting the same station number as set to the QE71 within the same Ethernet loop.

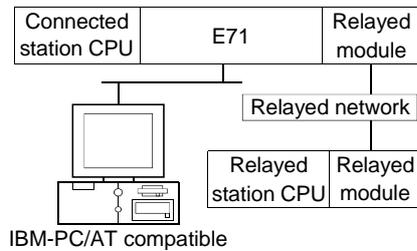
* 6: Do not use 1 to 1024 of ActPortNumber.

* 7: As the I/O address, specify the value found by dividing the actual first I/O number by 16.

3.3.6 ActAJ71E71TCP, ActMLAJ71E71TCP control

The following table indicates the properties possessed by the ActAJ71E71TCP, ActMLAJ71E71TCP control and their default values.

(1) Configuration



(2) Property patterns

| Connected Station CPU | | | Relayed Network | Relayed Station CPU | | | | |
|-----------------------|---------|---------|-----------------|---------------------|---------------|---------|---------|-------|
| QCPU (A mode) | QnA CPU | ACPU *1 | | QCPU (Q mode) | QCPU (A mode) | QnA CPU | ACPU *1 | FXCPU |
| ① | ①*2 | ① | MELSECNET/H | × | × | × | × | × |
| | | | MELSECNET/10 | × | ② | ②*2 | ② | × |
| | | | MELSECNET(II) | × | ② | ②*2 | ② | × |
| | | | Ethernet | × | × | × | × | × |
| | | | Computer link | × | × | × | × | × |
| | | | CC-Link | × | × | × | × | × |

○ : Accessible (Property pattern within circle)
 × : Inaccessible
 *1 : Including motion controller CPU
 *2 : Operates as the one equivalent to AnACPU.

(3) Property list

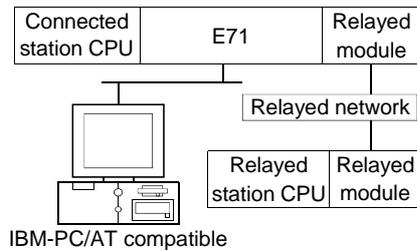
| Property | Default Value | Property Patterns | |
|--------------------------|------------------|--|---|
| | | ① | ② |
| ActCpuTimeOut | 40 | Any value specified by user in 250ms units | |
| ActCpuType | 262 (CPU_A1NCPU) | CPU type corresponding to target station | |
| ActDestinationPortNumber | 1280 (0x500) | Port number of connected station side module | |
| ActHostAddress | 1.1.1.1 | Host name or IP address of connected station side module | |
| ActPacketType | 2 (PACKET_ASCII) | PACKET_BINARY or PACKET_ASCII | |
| ActStationNumber *1 | 255 (0xFF) | Fixed to 0xFF | Target station side module station number |
| ActTimeOut | 10000 | Any value specified by user in ms units | |

*1: Note the following points depending on whether the connected station side MELSECNET/10 module is the control station or ordinary station.
 When the connected station side MELSECNET/10 module is the control station..... Specify the actual station number of the target station side MELSECNET/10 module in ActStationNumber.
 When the connected station side MELSECNET/10 module is the ordinary station.... Always set the target station side MELSECNET/10 module as the control station and specify "0x00" in ActStationNumber.

3.3.7 ActAJ71E71UDP, ActMLAJ71E71UDP control

The following table indicates the properties possessed by the ActAJ71E71UDP, ActMLAJ71E71UDP control and their default values.

(1) Configuration



(2) Property patterns

| Connected Station CPU | | | Relayed Network | Relayed Station CPU | | | | |
|-----------------------|---------|---------|-----------------|---------------------|---------------|---------|---------|-------|
| QCPU (A mode) | QnA CPU | ACPU *1 | | QCPU (Q mode) | QCPU (A mode) | QnA CPU | ACPU *1 | FXCPU |
| ① | ① *2 | ① | MELSECNET/H | × | × | × | × | × |
| | | | MELSECNET/10 | × | ② | ② *2 | ② | × |
| | | | MELSECNET(II) | × | ② | ② *2 | ② | × |
| | | | Ethernet | × | × | × | × | × |
| | | | Computer link | × | × | × | × | × |
| | | | CC-Link | × | × | × | × | × |

○ : Accessible (Property pattern within circle)
 × : Inaccessible
 *1 : Including motion controller CPU
 *2 : Operates as the one equivalent to AnACPU.

(3) Property list

| Property | Default Value | Property Patterns | |
|--------------------------|----------------------|--|---|
| | | ① | ② |
| ActCpuTimeOut | 40 | Any value specified by user in 250ms units | |
| ActCpuType | 262 (CPU_A1NCPU) | CPU type corresponding to target station | |
| ActDestinationPortNumber | 1280 (0x500) | Port number of connected station side module | |
| ActHostAddress | 1.1.1.1 | Host name or IP address of connected station side module | |
| ActPacketType | 3 (PACKET_BINARY) | PACKET_BINARY or PACKET_ASCII | |
| ActPortNumber *1 | 0 | IBM-PC/AT compatible side port number | |
| ActStationNumber *2 | 255 (0xFF) | Fixed to 0xFF | Target station side module station number |
| ActTimeOut | 10000 | Any value specified by user in ms units | |

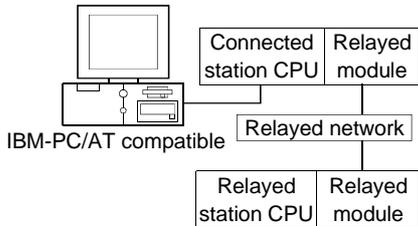
*1: 0 The free port number of the IBM-PC/AT compatible is assigned automatically.
 Other than 0 The specified port number is used to generate the UDP socket.
 Do not use 1 to 1024 of ActPortNumber.

*2: Note the following points depending on whether the connected station side MELSECNET/10 module is the control station or ordinary station.
 When the connected station side MELSECNET/10 module is the control station ... Specify the actual station number of the target station side MELSECNET/10 module in ActStationNumber.
 When the connected station side MELSECNET/10 module is the ordinary station ... Always set the target station side MELSECNET/10 module as the control station and specify "0x00" in ActStationNumber.

3.3.8 ActQCPUQ, ActMLQCPUQ control

The following table indicates the properties possessed by the ActQCPUQ, ActMLQCPUQ control and their default values.

(1) Configuration



(2) Property patterns

| Connected Station CPU | Relayed Network | Relayed Station CPU | | | | |
|-----------------------|-----------------|---------------------|---------------|---------|---------|-------|
| | | QCPU (Q mode) | QCPU (A mode) | QnA CPU | ACPU *1 | FXCPU |
| ① | MELSECNET/H | ② | × | × | × | × |
| | MELSECNET/10 | ② | ② | ② | ② | × |
| | MELSECNET(II) | × | × | × | × | × |
| | Ethernet | ② | × | ② | × | × |
| | Computer link | ③ | × | ③ | × | × |
| | CC-Link | ④ | ④*2 | ④*2 | ④*2 | × |

○ : Accessible (Property pattern within circle)
 × : Inaccessible
 *1 : Including motion controller CPU
 *2 : Use the QnA or ACPUs side CC-Link module whose ROM version is "S" or later.

(3) Property list

| Property | Default Value | Property Patterns | | | |
|-----------------------------|------------------------|--|---------------|---|---|
| | | ① | ②*2 | ③ | ④ |
| ActBaudRate | 19200 (BAUDRATE_19200) | BAUDRATE_9600, BAUDRATE_19200, BAUDRATE_38400, BAUDRATE_57600, BAUDRATE_115200 | | | |
| ActControl | 8 (TCR_DTR_OR_RTS) | Depending on used cable. | | | |
| ActCpuType | 34 (CPU_Q02CPU) | CPU type corresponding to target station | | | |
| ActDestinationIONumber | 0 (0x00) | Fixed to 0x00 | Fixed to 0x00 | Target station side For single CPU 0x3FF fixed For multiple CPUs Connected CPU: 0x3FF No. 1: 0x3E0 No. 2: 0x3E1 No. 3: 0x3E2 No. 4: 0x3E3 | Target station side For single CPU 0x3FF fixed For multiple CPUs Connected CPU: 0x3FF No. 1: 0x3E0 No. 2: 0x3E1 No. 3: 0x3E2 No. 4: 0x3E3 |
| ActDidPropertyBit | 1 (0x01) | Fixed to 0x01 | Fixed to 0x01 | Fixed to 0x00 | Fixed to 0x00 |
| ActDisdPropertyBit | 1 (0x01) | Fixed to 0x01 | Fixed to 0x01 | Fixed to 0x00 | Fixed to 0x00 |
| ActIntelligentPreferenceBit | 0 (0x00) | Fixed to 0x00 | Fixed to 0x00 | 0x01 (target station is QCPU (Q mode), 0x00 (target station is other than QCPU (Q mode)) | 0x01 (target station is QCPU (Q mode), 0x00 (target station is other than QCPU (Q mode)) |

(To next page)

| Property | Default Value | Property Patterns | | | |
|---------------------------|-----------------|--|---|---|---|
| | | ① | ② *2 | ③ | ④ |
| ActIONumber * 1 | 1023 (0x3FF) | For single CPU 0x3FF fixed For multiple CPUs Connected CPU: 0x3FF No. 1: 0x3E0 No. 2: 0x3E1 No. 3: 0x3E2 No. 4: 0x3E3 | Target station side For single CPU 0x3FF fixed For multiple CPUs Connected CPU: 0x3FF No. 1: 0x3E0 No. 2: 0x3E1 No. 3: 0x3E2 No. 4: 0x3E3 | Connected station side module I/O address | Connected station side module I/O address |
| ActMultiDropChannelNumber | 0 (0x00) | Fixed to 0x00 | Fixed to 0x00 | 0x00 or 0x02 | Fixed to 0x02 |
| ActNetworkNumber | 0 (0x00) | Fixed to 0x00 | Target station side module network number | Fixed to 0x00 | Fixed to 0x00 |
| ActPortNumber | 1 (PORT_1) | IBM-PC/AT compatible side COM port number | | | |
| ActStationNumber | 255 (0xFF) | Fixed to 0xFF | Target station side module station number | Fixed to 0xFF | Fixed to 0xFF |
| ActThroughNetworkType | 0 (0x00) | QCPU (Q mode): 0x00 (MELSECNET/H only), other than QCPU (Q mode): 0x01 (including MELSECNET/10). Note that the setting must be the same as set in the network parameter of the GPP function. | | | |
| ActTimeOut | 10000 | Any value specified by user in ms units | | | |
| ActUnitNumber | 0 (0x00) | Fixed to 0x00 | Fixed to 0x00 | Target station side module station number | Target station side module station number |

* 1: As the I/O address, specify the value found by dividing the actual first I/O number by 16.

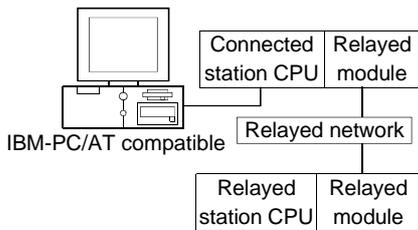
* 2: Note the following points when making access via the Ethernet module (Q series-compatible E71, QE71).

- For ActNetworkNumber and ActStationNumber, specify the value set in the parameter setting of the target station side Q series-compatible E71 or QE71.
- Set the "MNET/10 routing information" in the parameter setting of the Q series-compatible E71 or QE71. Also, when making setting, specify other than the automatic response system (any of the IP address calculation system, table conversion system and combined system) as the "MNET/10 routing system".

3.3.9 ActQCPUA, ActMLQCPUA control

The following table indicates the properties possessed by the ActQCPUA, ActMLQCPUA control and their default values.

(1) Configuration



(2) Property patterns

| Connected Station CPU | Relayed Network | Relayed Station CPU | | | | |
|-----------------------|-----------------|---------------------|---------------|---------|---------|-------|
| | | QCPU (Q mode) | QCPU (A mode) | QnA CPU | ACPU *1 | FXCPU |
| ① | MELSECNET/H | × | × | × | × | × |
| | MELSECNET/10 | × | ② | × | ② | × |
| | MELSECNET(II) | × | ③ | × | ③ | × |
| | Ethernet | × | × | × | × | × |
| | Computer link | × | × | × | × | × |
| | CC-Link | × | × | × | × | × |

○ : Accessible (Property pattern within circle)

× : Inaccessible

* 1 : Including motion controller CPU

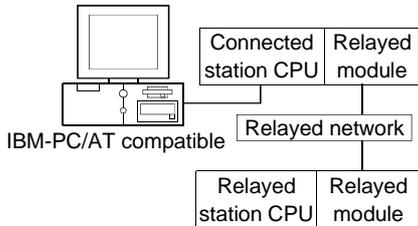
(3) Property list

| Property | Default Value | Property Patterns | | |
|------------------|-------------------------|--|---|---|
| | | ① | ② | ③ |
| ActBaudRate | 9600 (BAUDRATE_9600) | BAUDRATE_9600, BAUDRATE_19200, BAUDRATE_38400, BAUDRATE_57600, BAUDRATE_115200 | | |
| ActControl | 8 (TCR_DTR_OR_RTS) | Depending on used cable. | | |
| ActCpuType | 321 (CPU_Q02CPU_A) | CPU type corresponding to target station | | |
| ActNetworkNumber | 0 (0x00) | Fixed to 0x00 | Target station side module network number | Fixed to 0x00 |
| ActPortNumber | 1 (PORT_1) | IBM-PC/AT compatible side COM port number | | |
| ActStationNumber | 255 (0xFF) | Fixed to 0xFF | Target station side module station number | Target station side module station number |
| ActTimeOut | 10000 | Any value specified by user in ms units. | | |

3.3.10 ActQnACPU, ActMLQnACPU control

The following table indicates the properties possessed by the ActQnACPU, ActMLQnACPU control and their default values.

(1) Configuration



(2) Property patterns

| Connected Station CPU | Relayed Network | Relayed Station CPU | | | | |
|-----------------------|-----------------|---------------------|---------------|---------|---------|-------|
| | | QCPU (Q mode) | QCPU (A mode) | QnA CPU | ACPU *1 | FXCPU |
| ① | MELSECNET/H | × | × | × | × | × |
| | MELSECNET/10 | × | × | ② | × | × |
| | MELSECNET(II) | × | × | ③ | × | × |
| | Ethernet | × | × | ② | × | × |
| | Computer link | × | × | ④ | × | × |
| | CC-Link | × | × | × | × | × |

○ : Accessible (Property pattern within circle)
 × : Inaccessible
 * 1 : Including motion controller CPU

(3) Property list

| Property | Default Value | Property Patterns | | | |
|------------------|------------------------|--|---|---|---|
| | | ① | ② *2 | ③ | ④ |
| ActBaudRate | 19200 (BAUDRATE_19200) | BAUDRATE_9600, BAUDRATE_19200, BAUDRATE_38400 *3 | | | |
| ActControl | 8 (TCR_DTR_OR_RTS) | Depending on used cable. | | | |
| ActCpuType | 17 (CPU_Q2ACPU) | CPU type corresponding to target station | | | |
| ActIONumber *1 | 1023 (0x3FF) | Fixed to 0x3FF | Fixed to 0x3FF | Fixed to 0x3FF | Connected station side module I/O address |
| ActNetworkNumber | 0 (0x00) | Fixed to 0x00 | Target station side module network number | Fixed to 0x00 | Fixed to 0x00 |
| ActPortNumber | 1 (PORT_1) | IBM-PC/AT compatible side COM port number | | | |
| ActStationNumber | 255 (0xFF) | Fixed to 0xFF | Target station side module station number | Target station side module station number | Fixed to 0xFF |
| ActTimeOut | 10000 | Any value specified by user in ms units. | | | |
| ActUnitNumber | 0 (0x00) | Fixed to 0x00 | Fixed to 0x00 | Fixed to 0x00 | Target station side module station number |

* 1: As the I/O address, specify the value found by dividing the actual first I/O number by 16.

* 2: Note the following points when making access via the Ethernet module (QE71).

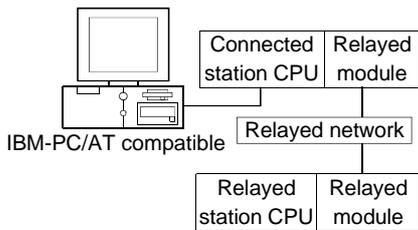
- For ActNetworkNumber and ActStationNumber, specify the value set in the parameter setting of the target station side QE71.
- Set the "MNET/10 routing information" in the parameter setting of the QE71. Also, when making setting, specify other than the automatic response system (any of the IP address calculation system, table conversion system and combined system) as the "MNET/10 routing system".

* 3: Usable for only the QnACPU version 9707B or later.

3.3.11 ActACPU, ActMLACPU control

The following table indicates the properties possessed by the ActACPU, ActMLACPU control and their default values.

(1) Configuration



(2) Property patterns

| Connected Station CPU | Relayed Network | Relayed Station CPU | | | | |
|-----------------------|-----------------|---------------------|---------------|---------|---------|-------|
| | | QCPU (Q mode) | QCPU (A mode) | QnA CPU | ACPU *1 | FXCPU |
| ① | MELSECNET/H | × | × | × | × | × |
| | MELSECNET/10 | × | ② | × | ② | × |
| | MELSECNET(II) | × | ③ | × | ③ | × |
| | Ethernet | × | × | × | × | × |
| | Computer link | × | × | × | × | × |
| | CC-Link | × | × | × | × | × |

○ : Accessible (Property pattern within circle)

× : Inaccessible

* 1 : Including motion controller CPU

(3) Property list

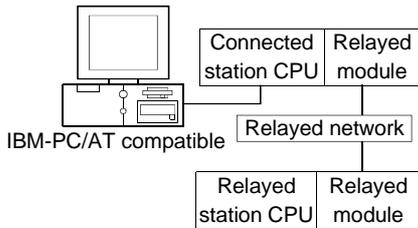
| Property | Default Value | Property Patterns | | |
|------------------|-------------------------|---|---|---|
| | | ① | ② | ③ |
| ActBaudRate | 9600 (BAUDRATE_9600) | Fixed to BAUDRATE_9600 * 1 | | |
| ActControl | 8 (TCR_DTR_OR_RTS) | Depending on used cable. | | |
| ActCpuType | 262 (CPU_A1NCPU) | CPU type corresponding to target station | | |
| ActNetworkNumber | 0 (0x00) | Fixed to 0x00 | Target station side module network number | Fixed to 0x00 |
| ActPortNumber | 1 (PORT_1) | IBM-PC/AT compatible side COM port number | | |
| ActStationNumber | 255 (0xFF) | Fixed to 0xFF | Target station side module station number | Target station side module station number |
| ActTimeOut | 10000 | Any value specified by user in ms units. | | |

* 1: BAUDRATE_9600 may be used only when the connected station CPU is the A2USHCPU-S1.

3.3.12 ActFXCPU, ActMLFXCPU control

The following table indicates the properties possessed by the ActFXCPU, ActMLFXCPU control and their default values.

(1) Configuration



(2) Property patterns

| Connected Station CPU | Relayed Network | Relayed Station CPU | | | | |
|-----------------------|-----------------|---------------------|---------------|---------|---------|-------|
| | | QCPU (Q mode) | QCPU (A mode) | QnA CPU | ACPU *1 | FXCPU |
| ① | MELSECNET/H | × | × | × | × | × |
| | MELSECNET/10 | × | × | × | × | × |
| | MELSECNET(II) | × | × | × | × | × |
| | Ethernet | × | × | × | × | × |
| | Computer link | × | × | × | × | × |
| | CC-Link | × | × | × | × | × |

○ : Accessible (Property pattern within circle)

× : Inaccessible

* 1 : Including motion controller CPU

(3) Property list

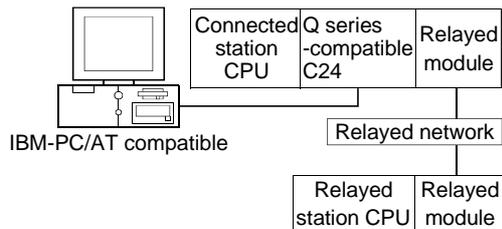
| Property | Default Value | Property Patterns |
|---------------|-----------------------|---|
| | | ① |
| ActControl | 8 (TCR_DTR_OR_RTS) | Depending on used cable. |
| ActCpuType | 513 (CPU_FX0CPU) | CPU type corresponding to target station |
| ActPortNumber | 1 (PORT_1) | IBM-PC/AT compatible side COM port number |
| ActTimeOut | 10000 | Any value specified by user in ms units. |

3.3.13 ActQJ71C24, ActMLQJ71C24 control

The following table indicates the properties possessed by the ActQJ71C24, ActMLQJ71C24 control and their default values.

(1) When there is relayed module in addition to connected station side Q series-compatible C24

(a) Configuration



(b) Property patterns

| Connected Station CPU | Relayed Network | Relayed Station CPU | | | | |
|-----------------------|-----------------|---------------------|---------------|---------|---------|-------|
| | | QCPU (Q mode) | QCPU (A mode) | QnA CPU | ACPU *1 | FXCPU |
| ① | MELSECNET/H | ② | × | × | × | × |
| | MELSECNET/10 | ② | ② | ② | ② | × |
| | MELSECNET(II) | × | × | × | × | × |
| | Ethernet | ② | × | ② | × | × |
| | Computer link | ③ | × | ③ | × | × |
| | CC-Link | ④ | ④ | ④ | ④ | × |

○ : Accessible (Property pattern within circle)
 × : Inaccessible
 *1 : Including motion controller CPU

(c) Property list

| Property | Default Value | Property Patterns | | | |
|------------------------|---------------------------|--|---------------|---|---|
| | | ① | ② *2 | 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 (TCR_DTR_OR_RTS) | Depending on used cable. | | | |
| ActCpuType | 34 (CPU_Q02CPU) | CPU type corresponding to target station | | | |
| ActDestinationIONumber | 0 (0x00) | Fixed to 0x00 | Fixed to 0x00 | Target station side For single CPU 0x3FF fixed For multiple CPUs Connected CPU: 0x3FF No. 1: 0x3E0 No. 2: 0x3E1 No. 3: 0x3E2 No. 4: 0x3E3 | Target station side For single CPU 0x3FF fixed For multiple CPUs Connected CPU: 0x3FF No. 1: 0x3E0 No. 2: 0x3E1 No. 3: 0x3E2 No. 4: 0x3E3 |
| ActDidPropertyBit | 1 (0x01) | Fixed to 0x01 | Fixed to 0x01 | Fixed to 0x00 | Fixed to 0x00 |

*2: Note the following points when making access via the Ethernet module (Q series-compatible E71, QE71).

- For ActNetworkNumber and ActStationNumber, specify the value set in the parameter setting of the target station side Q series-compatible E71 or QE71.
- Set the "MNET/10 routing information" in the parameter setting of the Q series-compatible E71 or QE71. Also, when making setting, specify other than the automatic response system (any of the IP address calculation system, table conversion system and combined system) as the "MNET/10 routing system".

(To next page)

| Property | Default Value | Property Patterns | | | |
|-----------------------------|-------------------|--|---|---|---|
| | | ① | ② *2 | ③ | ④ |
| ActDisdPropertyBit | 1 (0x01) | Fixed to 0x01 | Fixed to 0x01 | Fixed to 0x00 | Fixed to 0x00 |
| ActIntelligentPreferenceBit | 0 (0x00) | Fixed to 0x00 | Fixed to 0x00 | Fixed to 0x00 | Fixed to 0x00 |
| ActIONumber * 1 | 1023 (0x3FF) | For single CPU 0x3FF fixed For multiple CPUs Connected CPU: 0x3FF No. 1: 0x3E0 No. 2: 0x3E1 No. 3: 0x3E2 No. 4: 0x3E3 | Target station side For single CPU 0x3FF fixed For multiple CPUs Connected CPU: 0x3FF No. 1: 0x3E0 No. 2: 0x3E1 No. 3: 0x3E2 No. 4: 0x3E3 | Connected station side module I/O address | Connected station side module I/O address |
| ActMultiDropChannelNumber | 0 (0x00) | Fixed to 0x00 | Fixed to 0x00 | 0x00 or 0x02 | Fixed to 0x00 |
| ActNetworkNumber | 0 (0x00) | Fixed to 0x00 | Target station side module network number | Fixed to 0x00 | Fixed to 0x00 |
| ActParity | 1 (ODD_PARITY) | Match to the setting of Q series-compatible C24. | | | |
| ActPortNumber | 1 (PORT_1) | IBM-PC/AT compatible side COM port number | | | |
| ActStationNumber | 255 (0xFF) | Fixed to 0xFF | Target station side module station number | Fixed to 0xFF | Fixed to 0xFF |
| ActThroughNetworkType | 0 (0x00) | QCPU (Q mode): 0x00 (MELSECNET/H only), other than QCPU (Q mode): 0x01 (including MELSECNET/10). Note that the setting must be the same as set in the network parameter of the GPP function. | | | |
| ActTimeOut | 10000 | Any value specified by user in ms units | | | |
| ActUnitNumber | 0 (0x00) | Fixed to 0x00 | Fixed to 0x00 | Target station side module station number | Target station side module station number |

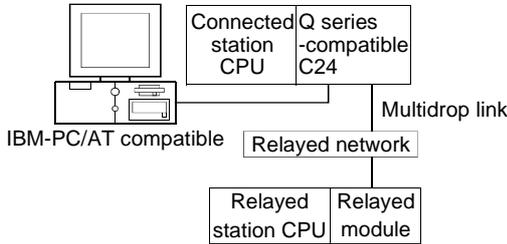
* 1: As the I/O address, specify the value found by dividing the actual first I/O number by 16.

* 2: Note the following points when making access via the Ethernet module (Q series-compatible E71, QE71).

- For ActNetworkNumber and ActStationNumber, specify the value set in the parameter setting of the target station side Q series-compatible E71 or QE71.
- Set the "MNET/10 routing information" in the parameter setting of the Q series-compatible E71 or QE71. Also, when making setting, specify other than the automatic response system (any of the IP address calculation system, table conversion system and combined system) as the "MNET/10 routing system".

(2) When connected station side Q series-compatible C24 is used for multidrop link with relayed module

(a) Configuration



(b) Property patterns

| Connected Station CPU | Relayed Network | Relayed Station CPU | | | | |
|-----------------------|-----------------|---------------------|---------------|---------|---------|-------|
| | | QCPU (Q mode) | QCPU (A mode) | QnA CPU | ACPU *1 | FXCPU |
| Independent mode *2 | Computer link | ② | × | ② | × | × |
| Synchronous mode *2 | | ③ | × | × | × | × |

○ : Accessible (Property pattern within circle)
 × : Inaccessible
 *1 : Including motion controller CPU
 *2 : Indicates the CH2 side setting (CH1 side fixed to independent mode)

(c) Property list

| 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 (TCR_DTR_OR_RTS) | Depending on used cable. | | |
| ActCpuType | 34 (CPU_Q02CPU) | CPU type corresponding to target station | | |
| ActDestinationIONumber | 0 (0x00) | Fixed to 0x00 | Target station side For single CPU 0x3FF fixed For multiple CPUs Connected CPU: 0x3FF No. 1: 0x3E0 No. 2: 0x3E1 No. 3: 0x3E2 No. 4: 0x3E3 | Fixed to 0x00 |
| ActDidPropertyBit | 1 (0x01) | Fixed to 0x01 | Fixed to 0x00 | Fixed to 0x01 |
| ActDsidPropertyBit | 1 (0x01) | Fixed to 0x01 | Fixed to 0x00 | Fixed to 0x01 |
| ActIntelligentPreferenceBit | 0 (0x00) | Fixed to 0x00 | Fixed to 0x00 | Fixed to 0x00 |

(To next page)

| Property | Default Value | Property Patterns | | |
|---------------------------|-------------------|--|--|----------------|
| | | ① | ② | ③ |
| ActIONumber * 1 | 1023 (0x3FF) | For single CPU 0x3FF fixed For multiple CPUs Connected CPU: 0x3FF No. 1: 0x3E0 No. 2: 0x3E1 No. 3: 0x3E2 No. 4: 0x3E3 | Connected station side module I/O address | Fixed to 0x3FF |
| ActMultiDropChannelNumber | 0 (0x00) | Fixed to 0x00 | 0x00 or 0x02 | Fixed to 0x00 |
| ActNetworkNumber | 0 (0x00) | Fixed to 0x00 | Fixed to 0x00 | Fixed to 0x00 |
| ActParity | 1 (ODD_PARITY) | Match to the setting of Q series-compatible C24. | | |
| ActPortNumber | 1 (PORT_1) | IBM-PC/AT compatible side COM port number | | |
| ActStationNumber | 255 (0xFF) | Fixed to 0x0FF | Fixed to 0x0FF | Fixed to 0x0FF |
| ActThroughNetworkType | 0 (0x00) | QCPU (Q mode): 0x00 (MELSECNET/H only), other than QCPU (Q mode): 0x01 (including MELSECNET/10). Note that the setting must be the same as set in the network parameter of the GPP function. | | |
| ActTimeOut | 10000 | Any value specified by user in ms units | | |
| ActUnitNumber | 0 (0x00) | Fixed to 0x00 | Target station side module station number | Fixed to 0x00 |

* 1: As the I/O address, specify the value found by dividing the actual first I/O number by 16.

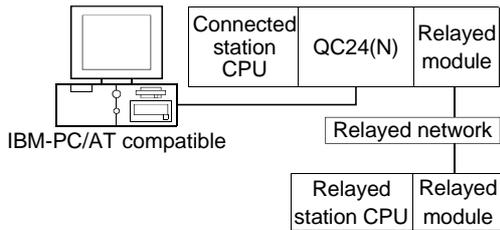
| | |
|--------------|---|
| POINT | <p>When the connected station side Q series-compatible C24 is set to the synchronous mode, always set the "sumcheck (SW06)" transmission specification software switch setting of the Q series-compatible C24 parameters to Yes (ON). If it is set to No (OFF), a communication error will occur, disabling proper communication.</p> |
|--------------|---|

3.3.14 ActAJ71QC24, ActMLAJ71QC24 control

The following table indicates the properties possessed by the ActAJ71QC24, ActMLAJ71QC24 control and their default values.

(1) When there is relayed module in addition to connected station side QC24(N)

(a) Configuration



(b) Property patterns

| Connected Station CPU | Relayed Network | Relayed Station CPU | | | | |
|-----------------------|-----------------|---------------------|---------------|---------------|---------|---------|
| | | QnACPU | QCPU (Q mode) | QCPU (A mode) | QnA CPU | ACPU *1 |
| ① | MELSECNET/H | × | × | × | × | × |
| | MELSECNET/10 | × | × | ② | × | × |
| | MELSECNET(II) | × | × | ③ | × | × |
| | Ethernet | × | × | ② | × | × |
| | Computer link | × | × | ④ | × | × |
| | CC-Link | × | × | ④ | × | × |

○ : Accessible (Property pattern within circle)
 × : Inaccessible
 * 1 : Including motion controller CPU

(c) Property list

| Property | Default Value | Property Patterns | | | |
|----------------------|------------------------|--|---|---|---|
| | | ① | ② *2 | ③ | ④ |
| ActBaudRate | 19200 (BAUDRATE_19200) | Match to the setting of QC24(N). | | | |
| ActConnectUnitNumber | 0 (0x00) | Connected station side module station number | | | |
| ActControl | 8 (TCR_DTR_OR_RTS) | Depending on used cable. | | | |
| ActCpuType | 17 (CPU_Q2ACPU) | CPU type corresponding to target station | | | |
| ActIONumber *1 | 1023 (0x3FF) | Fixed to 0x3FF | Fixed to 0x3FF | Fixed to 0x3FF | Connected station side module I/O address |
| ActNetworkNumber | 0 (0x00) | Fixed to 0x00 | Target station side module network number | Fixed to 0x00 | Fixed to 0x00 |
| ActParity | 1 (ODD_PARITY) | Match to the setting of QC24(N). | | | |
| ActPortNumber | 1 (PORT_1) | IBM-PC/AT compatible side COM port number | | | |
| ActStationNumber | 255 (0xFF) | Fixed to 0xFF | Target station side module station number | Target station side module station number | Fixed to 0xFF |
| ActTimeOut | 10000 | Any value specified by user in ms units | | | |
| ActUnitNumber | 0 (0x00) | Fixed to 0x00 | Fixed to 0x00 | Fixed to 0x00 | Target station side module station number |

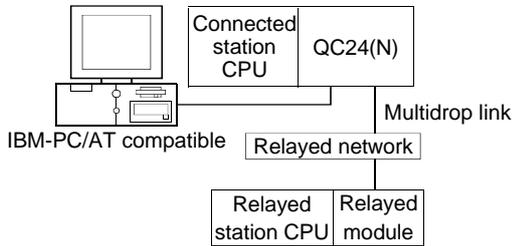
* 1: As the I/O address, specify the value found by dividing the actual first I/O number by 16.

* 2: Note the following points when making access via the Ethernet module (QE71).

- For ActNetworkNumber and ActStationNumber, specify the value set in the parameter setting of the target station side QE71.
- Set the "MNET/10 routing information" in the parameter setting of the QE71. Also, when making setting, specify other than the automatic response system (any of the IP address calculation system, table conversion system and combined system) as the "MNET/10 routing system".

(2) When connected station side QC24(N) is used for multidrop link with relayed module

(a) Configuration



(b) Property patterns

| Connected Station CPU | | Relayed Network | Relayed Station CPU | | | | |
|-----------------------|---|-----------------|---------------------|---------------|---------|---------|-------|
| QnACPU | | | QCPU (Q mode) | QCPU (A mode) | QnA CPU | ACPU *1 | FXCPU |
| Independent mode *2 | ① | Computer link | × | × | ② | × | × |
| Synchronous mode *2 | | | × | × | ③ | × | × |

○ : Accessible (Property pattern within circle)
 × : Inaccessible
 *1 : Including motion controller CPU
 *2 : Indicates the CH2 side setting (CH1 side fixed to independent mode)

(c) Property list

| Property | Default Value | Property Patterns | | |
|----------------------|------------------------|--|---|----------------|
| | | ① | ② | ③ |
| ActBaudRate | 19200 (BAUDRATE_19200) | Match to the setting of QC24(N). | | |
| ActConnectUnitNumber | 0 (0x00) | Connected station side module station number | | |
| ActControl | 8 (TCR_DTR_OR_RTS) | Depending on used cable. | | |
| ActCpuType | 17 (CPU_Q2ACPU) | CPU type corresponding to target station | | |
| ActIONumber *1 | 1023 (0x3FF) | Fixed to 0x3FF | Connected station side module I/O address | Fixed to 0x3FF |
| ActNetworkNumber | 0 (0x00) | Fixed to 0x00 | Fixed to 0x00 | Fixed to 0x00 |
| ActParity | 1 (ODD_PARITY) | Match to the setting of QC24(N). | | |
| ActPortNumber | 1 (PORT_1) | IBM-PC/AT compatible side COM port number | | |
| ActStationNumber | 255 (0xFF) | Fixed to 0x0FF | Fixed to 0x0FF | Fixed to 0x0FF |
| ActTimeOut | 10000 | Any value specified by user in ms units | | |
| ActUnitNumber | 0 (0x00) | Fixed to 0x00 | Target station side module station number | Fixed to 0x00 |

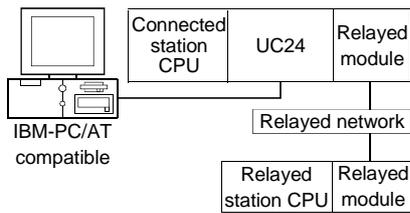
*1: As the I/O address, specify the value found by dividing the actual first I/O number by 16.

3.3.15 ActAJ71UC24, ActMLAJ71UC24 control

The following table indicates the properties possessed by the ActAJ71UC24, ActMLAJ71UC24 control and their default values.

(1) When there is relayed module in addition to connected station side UC24

(a) Configuration



(b) Property patterns

| Connected Station CPU | | | Relayed Network | Relayed Station CPU | | | | |
|-----------------------|---------|---------|-----------------|---------------------|---------------|---------|---------|-------|
| QCPU (A mode) | QnA CPU | ACPU *1 | | QCPU (Q mode) | QCPU (A mode) | QnA CPU | ACPU *1 | FXCPU |
| ① | ① *2 | ① | MELSECNET/H | × | × | × | × | × |
| | | | MELSECNET/10 | × | ② | ② *2 | ② | × |
| | | | MELSECNET(II) | × | ③ | ③ *2 | ③ | × |
| | | | Ethernet | × | × | × | × | × |
| | | | Computer link | × | × | × | × | × |
| | | | CC-Link | × | × | × | × | × |

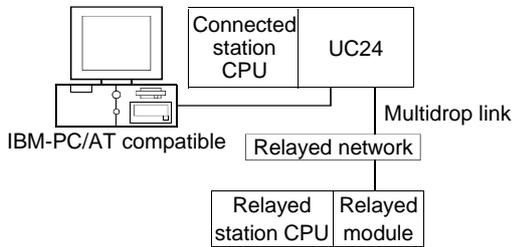
- : Accessible (Property pattern within circle)
- × : Inaccessible
- *1 : Including motion controller CPU
- *2 : Operates as the one equivalent to AnACPU.

(c) Property list

| Property | Default Value | Property Patterns | | |
|------------------|------------------------|---|--|--|
| | | ① | ② | ③ |
| ActBaudRate | 19200 (BAUDRATE_19200) | Match to the setting of UC24. | | |
| ActControl | 8 (TCR_DTR_OR_RTS) | Depending on used cable. | | |
| ActCpuType | 262 (CPU_A1NCPU) | CPU type corresponding to target station | | |
| ActDataBits | 8 (DATABIT_8) | Match to the setting of UC24. | | |
| ActNetworkNumber | 0 (0x00) | Fixed to 0x00 | Target station side module network number | Fixed to 0x00 |
| ActParity | 1 (ODD_PARITY) | Match to the setting of UC24. | | |
| ActPortNumber | 1 (PORT_1) | IBM-PC/AT compatible side COM port number | | |
| ActStationNumber | 255 (0xFF) | Fixed to 0xFF | Target station side module station number | Target station side module station number |
| ActStopBits | 0 (STOPBIT_ONE) | Match to the setting of UC24. | | |
| ActSumCheck | 1 (SUM_CHECK) | Match to the setting of UC24. | | |
| ActTimeOut | 10000 | Any value specified by user in ms units | | |
| ActUnitNumber | 0 (0x00) | Target station side module station number | Connected station side module station number | Connected station side module station number |

(2) When connected station side UC24 is used for multidrop link with relayed module

(a) Configuration



(b) Property patterns

| Connected Station CPU | Relayed Network | Relayed Station CPU | | | | |
|---|-----------------|---------------------|---------------|---------|---------|-------|
| | | QCPU (Q mode) | QCPU (A mode) | QnA CPU | ACPU *1 | FXCPU |
| QCPU (A mode), QnACPU *3, ACPU *1 | | | | | | |
| Independent mode *2 | ① | × | ① | ① *3 | ① | × |

- : Accessible (Property pattern within circle)
- × : Inaccessible
- *1 : Including motion controller CPU
- *2 : Use the mode setting switch and main channel setting to make setting.
- *3 : Operates as the one equivalent to AnACPU.

(c) Property list

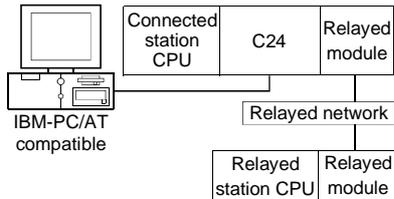
| Property | Default Value | Property Patterns |
|------------------|---------------------------|---|
| | | ① |
| ActBaudRate | 19200 (BAUDRATE_19200) | Match to the setting of UC24. |
| ActControl | 8 (TCR_DTR_OR_RTS) | Depending on used cable. |
| ActCpuType | 262 (CPU_A1NCPU) | CPU type corresponding to target station |
| ActDataBits | 8 (DATABIT_8) | Match to the setting of UC24. |
| ActNetworkNumber | 0 (0x00) | Fixed to 0x00 |
| ActParity | 1 (ODD_PARITY) | Match to the setting of UC24. |
| ActPortNumber | 1 (PORT_1) | IBM-PC/AT compatible side COM port number |
| ActStationNumber | 255 (0xFF) | Fixed to 0x0FF |
| ActStopBits | 0 (STOPBIT_ONE) | Match to the setting of UC24. |
| ActSumCheck | 1 (SUM_CHECK) | Match to the setting of UC24. |
| ActTimeOut | 10000 | Any value specified by user in ms units |
| ActUnitNumber | 0 (0x00) | Target station side module station number |

3.3.16 ActAJ71C24, ActMLAJ71C24 control

The following table indicates the properties possessed by the ActAJ71C24, ActMLAJ71C24 control and their default values.

(1) When there is relayed module in addition to connected station side C24

(a) Configuration



(b) Property patterns

| Connected Station CPU | | | Relayed Network | Relayed Station CPU | | | | |
|-----------------------|---------|---------|-----------------|---------------------|---------------|---------|---------|-------|
| QCPU (A mode) | QnA CPU | ACPU *1 | | QCPU (Q mode) | QCPU (A mode) | QnA CPU | ACPU *1 | FXCPU |
| ① | ① *2 | ① | MELSECNET/H | × | × | × | × | × |
| | | | MELSECNET/10 | × | ② | ② *2 | ② | × |
| | | | MELSECNET(II) | × | ② | ② *2 | ② | × |
| | | | Ethernet | × | × | × | × | × |
| | | | Computer link | × | × | × | × | × |
| | | | CC-Link | × | × | × | × | × |

○ : Accessible (Property pattern within circle)
 × : Inaccessible
 *1 : Including motion controller CPU
 *2 : Operates as the one equivalent to AnACPU.

(c) Property list

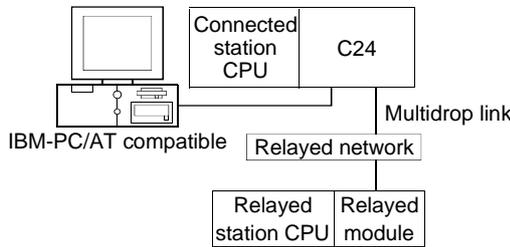
| Property | Default Value | Property Patterns | |
|---------------------|------------------------|---|--|
| | | ① | ② *2 |
| ActBaudRate | 19200 (BAUDRATE_19200) | Match to the setting of C24. | |
| ActControl | 8 (TCR_DTR_OR_RTS) | Depending on used cable. | |
| ActCpuType | 262 (CPU_A1NCPU) | CPU type corresponding to target station | |
| ActDataBits | 8 (DATABIT_8) | Match to the setting of C24. | |
| ActParity | 1 (ODD_PARITY) | Match to the setting of C24. | |
| ActPortNumber | 1 (PORT_1) | IBM-PC/AT compatible side COM port number | |
| ActStationNumber *1 | 255 (0xFF) | Fixed to 0xFF | Target station side module station number |
| ActStopBits | 0 (STOPBIT_ONE) | Match to the setting of C24. | |
| ActSumCheck | 1 (SUM_CHECK) | Match to the setting of C24. | |
| ActTimeOut | 10000 | Any value specified by user in ms units | |
| ActUnitNumebr | 0 (0x00) | Target station side module station number | Connected station side module station number |

*1: Note the following points depending on whether the connected station side MELSECNET/10 module is the control station or ordinary station.
 When the connected station side MELSECNET/10 module is the control station... Specify the actual station number of the target station side MELSECNET/10 module in ActStationNumber.
 When the connected station side MELSECNET/10 module is the ordinary station... Always set the target station side MELSECNET/10 module as the control station and specify "0x00" in ActStationNumber.

*2: Access via network is enabled only to the network on the side specified in "valid module for another station access" in the connected station side network parameters.

(2) When connected station side C24 is used for multidrop link with relayed module

(a) Configuration



(b) Property patterns

| Connected Station CPU | Relayed Network | Relayed Station CPU | | | | |
|-------------------------------------|-----------------|---------------------|---------------|---------|----------|-------|
| | | QCPU (Q mode) | QCPU (A mode) | QnA CPU | ACPU * 1 | FXCPU |
| QCPU (A mode), QnACPU * 3, ACPU * 1 | | | | | | |
| Independent mode * 2 | ① | × | ① | ① * 3 | ① | × |

- : Accessible (Property pattern within circle)
- × : Inaccessible
- * 1 : Including motion controller CPU
- * 2 : Use the mode setting switch and main channel setting to make setting.
- * 3 : Operates as the one equivalent to AnACPU.

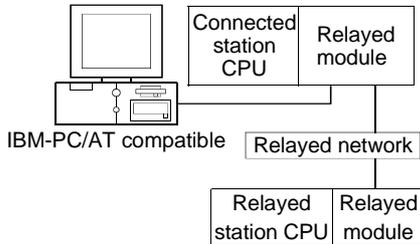
(c) Property list

| Property | Default Value | Property Patterns |
|------------------|------------------------|---|
| | | ① |
| ActBaudRate | 19200 (BAUDRATE_19200) | Match to the setting of C24. |
| ActControl | 8 (TCR_DTR_OR_RTS) | Depending on used cable. |
| ActCpuType | 262 (CPU_A1NCPU) | CPU type corresponding to target station |
| ActDataBits | 8 (DATABIT_8) | Match to the setting of C24. |
| ActParity | 1 (ODD_PARITY) | Match to the setting of C24. |
| ActPortNumber | 1 (PORT_1) | IBM-PC/AT compatible side COM port number |
| ActStationNumber | 255 (0xFF) | Fixed to 0xFF |
| ActStopBits | 0 (STOPBIT_ONE) | Match to the setting of C24. |
| ActSumCheck | 1 (SUM_CHECK) | Match to the setting of C24. |
| ActTimeOut | 10000 | Any value specified by user in ms units |
| ActUnitNumbr | 0 (0x00) | Target station side module station number |

3.3.17 ActQCPUQUSB, ActMLQCPUQUSB control

The following table indicates the properties possessed by the ActQCPUQUSB, ActMLQCPUQUSB control and their default values.

(1) Configuration



(2) Property patterns

| Connected Station CPU | Relayed Network | Relayed Station CPU | | | | |
|-----------------------|-----------------|---------------------|---------------|---------|---------|-------|
| | | QCPU (Q mode) | QCPU (A mode) | QnA CPU | ACPU *1 | FXCPU |
| ① | MELSECNET/H | ② | × | × | × | × |
| | MELSECNET/10 | ② | ② | ② | ② | × |
| | MELSECNET(II) | × | × | × | × | × |
| | Ethernet | ② | × | ② | × | × |
| | Computer link | ③ | × | ③ | × | × |
| | CC-Link | ④ | ④*2 | ④*2 | ④*2 | × |

○ : Accessible (Property pattern within circle)
 × : Inaccessible
 * 1 : Including motion controller CPU
 * 2 : *2: Use the QnA or ACPUs side CC-Link module whose ROM version is "S" or later.

(3) Property list

| Property | Default Value | Property Patterns | | | |
|-----------------------------|-----------------|--|---------------|---|---|
| | | ① | ②*2 | ③ | ④ |
| ActCpuType | 34 (CPU_Q02CPU) | CPU type corresponding to target station | | | |
| ActDestinationIONumber | 0 (0x00) | Fixed to 0x00 | Fixed to 0x00 | Target station side For single CPU 0x3FF fixed For multiple CPUs Connected CPU: 0x3FF No. 1: 0x3E0 No. 2: 0x3E1 No. 3: 0x3E2 No. 4: 0x3E3 | Target station side For single CPU 0x3FF fixed For multiple CPUs Connected CPU: 0x3FF No. 1: 0x3E0 No. 2: 0x3E1 No. 3: 0x3E2 No. 4: 0x3E3 |
| ActDidPropertyBit | 1 (0x01) | Fixed to 0x01 | Fixed to 0x01 | Fixed to 0x00 | Fixed to 0x00 |
| ActDisdPropertyBit | 1 (0x01) | Fixed to 0x01 | Fixed to 0x01 | Fixed to 0x00 | Fixed to 0x00 |
| ActIntelligentPreferenceBit | 0 (0x00) | Fixed to 0x00 | Fixed to 0x00 | 0x01 (target station is QCPU (Q mode), 0x00 (target station is other than QCPU (Q mode)) | 0x01 (target station is QCPU (Q mode), 0x00 (target station is other than QCPU (Q mode)) |

- * 2: Note the following points when making access via the Ethernet module (Q series-compatible E71, QE71).
- For ActNetworkNumber and ActStationNumber, specify the value set in the parameter setting of the target station side Q series-compatible E71 or QE71.
 - Set the "MNET/10 routing information" in the parameter setting of the Q series-compatible E71 or QE71. Also, when making setting, specify other than the automatic response system (any of the IP address calculation system, table conversion system and combined system) as the "MNET/10 routing system".

(To next page)

| Property | Default Value | Property Patterns | | | |
|---------------------------|-----------------|--|---|---|---|
| | | ① | ② *2 | ③ | ④ |
| ActIONumber * 1 | 1023 (0x3FF) | For single CPU 0x3FF fixed For multiple CPUs Connected CPU: 0x3FF No. 1: 0x3E0 No. 2: 0x3E1 No. 3: 0x3E2 No. 4: 0x3E3 | Target station side For single CPU 0x3FF fixed For multiple CPUs Connected CPU: 0x3FF No. 1: 0x3E0 No. 2: 0x3E1 No. 3: 0x3E2 No. 4: 0x3E3 | Connected station side module I/O address | Connected station side module I/O address |
| ActMultiDropChannelNumber | 0 (0x00) | Fixed to 0x00 | Fixed to 0x00 | 0x00 or 0x02 | Fixed to 0x00 |
| ActNetworkNumber | 0 (0x00) | Fixed to 0x00 | Target station side module network number | Fixed to 0x00 | Fixed to 0x00 |
| ActStationNumber | 255 (0xFF) | Fixed to 0xFF | Target station side module station number | Fixed to 0xFF | Fixed to 0xFF |
| ActThroughNetworkType | 0 (0x00) | QCPU (Q mode): 0x00 (MELSECNET/H only), other than QCPU (Q mode): 0x01 (including MELSECNET/10). Note that the setting must be the same as set in the network parameter of the GPP function. | | | |
| ActTimeOut | 10000 | Any value specified by user in ms units | | | |
| ActUnitNumber | 0 (0x00) | Fixed to 0x00 | Fixed to 0x00 | Target station side module station number | Target station side module station number |

* 1: As the I/O address, specify the value found by dividing the actual first I/O number by 16.

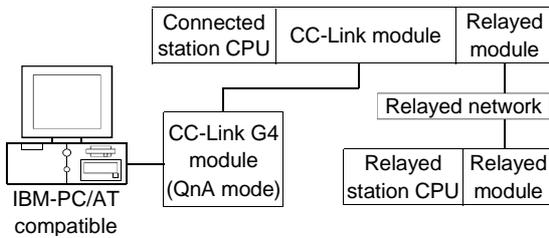
* 2: Note the following points when making access via the Ethernet module (Q series-compatible E71, QE71).

- For ActNetworkNumber and ActStationNumber, specify the value set in the parameter setting of the target station side Q series-compatible E71 or QE71.
- Set the "MNET/10 routing information" in the parameter setting of the Q series-compatible E71 or QE71. Also, when making setting, specify other than the automatic response system (any of the IP address calculation system, table conversion system and combined system) as the "MNET/10 routing system".

3.3.18 ActCCG4QnA, ActMLCCG4QnA control

The following table indicates the properties possessed by the ActCCG4QnA, ActMLCCG4QnA control and their default values.

(1) Configuration



(2) Property patterns

| Connected Station CPU | Relayed Network | Relayed Station CPU | | | | |
|-----------------------|-----------------|---------------------|---------------|---------|---------|-------|
| | | QCPU (Q mode) | QCPU (A mode) | QnA CPU | ACPU *1 | FXCPU |
| ① | MELSECNET/H | × | × | × | × | × |
| | MELSECNET/10 | × | × | ② | × | × |
| | MELSECNET(II) | × | × | ③ | × | × |
| | Ethernet | × | × | ② | × | × |
| | Computer link | × | × | ④ | × | × |
| | CC-Link | × | × | × | × | × |

○ : Accessible (Property pattern within circle)

× : Inaccessible

*1 : Including motion controller CPU

(3) Property list

| Property | Default Value | Property Patterns | | | |
|----------------------|------------------------|--|---|---|---|
| | | ① | ② *2 | ③ | ④ |
| ActBaudRate | 19200 (BAUDRATE_19200) | Match to the setting of CC-Link G4 module. | | | |
| ActConnectUnitNumber | 0 (0x00) | Connected station side CC-Link module station number | | | |
| ActControl | 8 (TCR_DTR_OR_RTS) | Depending on used cable. | | | |
| ActCpuType | 17 (CPU_Q2ACPU) | CPU type corresponding to target station | | | |
| ActIONumber *1 | 1023 (0x3FF) | Fixed to 0x3FF | Fixed to 0x3FF | Fixed to 0x3FF | Connected station side relayed module I/O address |
| ActNetworkNumber | 0 (0x00) | Fixed to 0x00 | Target station side module network number | Fixed to 0x00 | Fixed to 0x00 |
| ActPortNumber | 1 (PORT_1) | IBM-PC/AT compatible side COM port number | | | |
| ActStationNumber | 255 (0xFF) | Fixed to 0xFF | Target station side module station number | Target station side module station number | Fixed to 0xFF |
| ActTimeOut | 10000 | Any value specified by user in ms units | | | |
| ActUnitNumber | 0 (0x00) | Fixed to 0x00 | Fixed to 0x00 | Fixed to 0x00 | Target station side module station number |

*1: As the I/O address, specify the value found by dividing the actual first I/O number by 16.

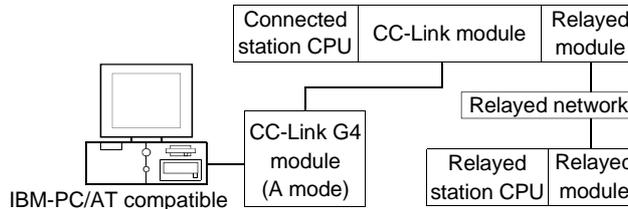
*2: Note the following points when making access via the Ethernet module (QE71).

- For ActNetworkNumber and ActStationNumber, specify the value set in the parameter setting of the target station side QE71.
- Set the "MNET/10 routing information" in the parameter setting of the QE71. Also, when making setting, specify other than the automatic response system (any of the IP address calculation system, table conversion system and combined system) as the "MNET/10 routing system".

3.3.19 ActCCG4A, ActMLCCG4A control

The following table indicates the properties possessed by the ActCCG4A, ActMLCCG4A control and their default values.

(1) Configuration



(2) Property patterns

| Connected Station CPU | | | Relayed Network | Relayed Station CPU | | | | |
|-----------------------|---------|---------|-----------------|---------------------|---------------|---------|---------|-------|
| QCPU (A mode) | QnA CPU | ACPU *1 | | QCPU (Q mode) | QCPU (A mode) | QnA CPU | ACPU *1 | FXCPU |
| ① | × | ① | MELSECNET/H | × | × | × | × | × |
| | | | MELSECNET/10 | × | × | × | × | × |
| | | | MELSECNET(II) | × | × | × | × | × |
| | | | Ethernet | × | × | × | × | × |
| | | | Computer link | × | × | × | × | × |
| | | | CC-Link | × | × | × | × | × |

○ : Accessible (Property pattern within circle)
 × : Inaccessible
 *1 : Including motion controller CPU

(3) Property list

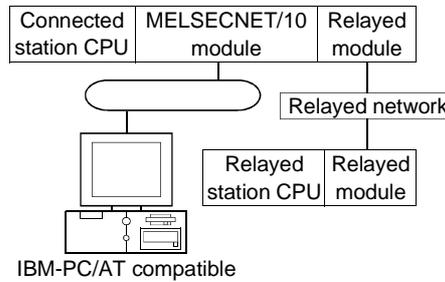
| Property | Default Value | Property Patterns |
|------------------|-----------------------|---|
| | | ① |
| ActControl | 8 (TCR_DTR_OR_RTS) | Depending on used cable. |
| ActCpuType | 262 (CPU_A1NCPU) | CPU type corresponding to target station |
| ActPortNumber | 1 (PORT_1) | IBM-PC/AT compatible side COM port number |
| ActStationNumber | 0 (0x00) | Target station side module station number |
| ActTimeOut | 10000 | Any value specified by user in ms units |

3.3.20 ActMnet10BD, ActMLMnet10BD control

The following table indicates the properties possessed by the ActMnet10BD, ActMLMnet10BD control and their default values.

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

(a) Configuration



(b) Property patterns

| Own Board | Connected Station CPU | Relayed Network | Relayed Station CPU | | | | |
|-----------|-----------------------|-----------------|---------------------|---------------|---------|---------|-------|
| | | | QCPU (Q mode) | QCPU (A mode) | QnA CPU | ACPU *1 | FXCPU |
| ① | ② | MELSECNET/H | × | × | × | × | × |
| | | MELSECNET/10 | ② | ② | ② | ② | × |
| | | MELSECNET(II) | × | × | × | × | × |
| | | Ethernet | ② | × | × | × | × |
| | | Computer link | ③ | × | × | × | × |
| | | CC-Link | ④ | × | × | × | × |

○ : Accessible (Property pattern within circle)
 × : Inaccessible
 *1 : Including motion controller CPU

(c) Property list

| Property | Default Value | Property Patterns | | | |
|------------------------|------------------|--|---------------|---|---|
| | | ① | ② | ③ | ④ |
| ActCpuType | 1025 (CPU_BOARD) | CPU type corresponding to target station | | | |
| ActDestinationIONumber | 0 (0x00) | Fixed to 0x00 | Fixed to 0x00 | Target station side For single CPU 0x3FF fixed For multiple CPUs Connected CPU: 0x3FF No. 1: 0x3E0 No. 2: 0x3E1 No. 3: 0x3E2 No. 4: 0x3E3 | Target station side For single CPU 0x3FF fixed For multiple CPUs Connected CPU: 0x3FF No. 1: 0x3E0 No. 2: 0x3E1 No. 3: 0x3E2 No. 4: 0x3E3 |
| ActDidPropertyBit | 0 (0x00) | Fixed to 0x00 | Fixed to 0x01 | Fixed to 0x00 | Fixed to 0x00 |
| ActDsidPropertyBit | 0 (0x00) | Fixed to 0x00 | Fixed to 0x01 | Fixed to 0x00 | Fixed to 0x00 |

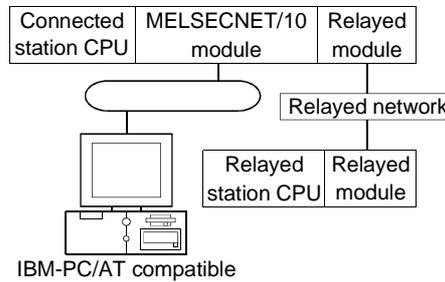
(To next page)

| Property | Default Value | Property Patterns | | | |
|---------------------------|---------------|---|--|---|---|
| | | ① | ② | ③ | ④ |
| ActIONumber * 1 | 0 (0x00) | Fixed to 0x00 | For single CPU 0x3FF fixed For multiple CPUs Connected CPU: 0x3FF No. 1: 0x3E0 No. 2: 0x3E1 No. 3: 0x3E2 No. 4: 0x3E3 | Connected station side relayed module I/O address | Connected station side relayed module I/O address |
| ActMultiDropChannelNumber | 0 (0x00) | Fixed to 0x00 | Fixed to 0x00 | 0x00 or 0x02 | Fixed to 0x00 |
| ActNetworkNumber | 0 (0x00) | Fixed to 0x00 | Target station side module network number | Connected station side module network number | Connected station side module network number |
| ActPortNumber | 1 (PORT_1) | Board No. of IBM-PC/AT compatible side MELSECNET/10 board, PORT 1 to PORT 4 (first to fourth boards) | | | |
| ActStationNumber | 255 (0xFF) | Fixed to 0xFF | Target station side module station number | Connected station side module station number | Connected station side module station number |
| ActUnitNumber | 0 (0x00) | Fixed to 0x00 | Fixed to 0x00 | Target station side module station number | Target station side module station number |

* 1: As the I/O address, specify the value found by dividing the actual first I/O number by 16.

(2) When connected station CPU is QCPU (A mode) or ACPU

(a) Configuration



(b) Property patterns

| Own Board | Connected Station CPU | | Relayed Network | Relayed Station CPU | | | | |
|-----------|-----------------------|----------|-----------------|---------------------|---------------|---------|----------|-------|
| | QCPU (A mode) | ACPU * 1 | | QCPU (Q mode) | QCPU (A mode) | QnA CPU | ACPU * 1 | FXCPU |
| ① | ② | ② | MELSECNET/H | × | × | × | × | × |
| | | | MELSECNET/10 | ③ | ② | ④ | ② | × |
| | | | MELSECNET(II) | × | × | × | × | × |
| | | | Ethernet | × | × | × | × | × |
| | | | Computer link | × | × | × | × | × |
| | | | CC-Link | × | × | × | × | × |

○ : Accessible (Property pattern within circle)

× : Inaccessible

* 1 : Including motion controller CPU

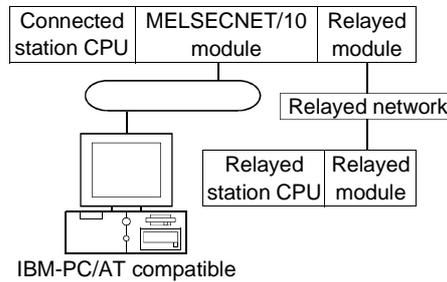
(c) Property list

| Property | Default Value | Property Patterns | | | |
|---------------------------|------------------|--|---|---|----------------|
| | | ① | ② | ③ | ④ |
| ActCpuType | 1025 (CPU_BOARD) | CPU type corresponding to target station | | | |
| ActDestinationIONumber | 0 (0x00) | Fixed to 0x00 | | | |
| ActDidPropertyBit | 0 (0x00) | Fixed to 0x00 | | Fixed to 0x01 | Fixed to 0x00 |
| ActDsidPropertyBit | 0 (0x00) | Fixed to 0x00 | | Fixed to 0x01 | Fixed to 0x00 |
| ActIONumber * 1 | 0 (0x00) | Fixed to 0x00 | | Target station side For single CPU 0x3FF fixed For multiple CPUs Connected CPU: 0x3FF No. 1: 0x3E0 No. 2: 0x3E1 No. 3: 0x3E2 No. 4: 0x3E3 | Fixed to 0x3FF |
| ActMultiDropChannelNumber | 0 (0x00) | Fixed to 0x00 | | | |
| ActNetworkNumber | 0 (0x00) | Fixed to 0x00 | Target station side module network number | | |
| ActPortNumber | 1 (PORT_1) | Board No. of IBM-PC/AT compatible side MELSECNET/10 board, PORT 1 to PORT 4 (first to fourth boards) | | | |
| ActStationNumber | 255 (0xFF) | Fixed to 0xFF | Target station side module station number | | |
| ActUnitNumber | 0 (0x00) | Fixed to 0x00 | | | |

* 1: As the I/O address, specify the value found by dividing the actual first I/O number by 16.

(3) When connected station CPU is QnACPU

(a) Configuration



(b) Property patterns

| Own Board | Connected Station CPU | Relayed Network | Relayed Station CPU | | | | |
|-----------|-----------------------|-----------------|---------------------|---------------|---------|---------|-------|
| | QnACPU | | QCPU (Q mode) | QCPU (A mode) | QnA CPU | ACPU *1 | FXCPU |
| ① | ② | MELSECNET/H | × | × | × | × | × |
| | | MELSECNET/10 | ③ | ② | ② | ② | × |
| | | MELSECNET(II) | × | × | × | × | × |
| | | Ethernet | × | × | ② | × | × |
| | | Computer link | × | × | ④ | × | × |
| | | CC-Link | × | × | × | × | × |

○ : Accessible (Property pattern within circle)
 × : Inaccessible
 * 1 : Including motion controller CPU

(c) Property list

| Property | Default Value | Property Patterns | | | |
|---------------------------|------------------|--|---|--|---|
| | | ① | ② | ③ | ④ |
| ActCpuType | 1025 (CPU_BOARD) | CPU type corresponding to target station | | | |
| ActDestinationIONumber | 0 (0x00) | Fixed to 0x00 | | | |
| ActDidPropertyBit | 0 (0x00) | Fixed to 0x00 | | Fixed to 0x01 | Fixed to 0x00 |
| ActDsidPropertyBit | 0 (0x00) | Fixed to 0x00 | | Fixed to 0x01 | Fixed to 0x00 |
| ActIONumber * 1 | 0 (0x00) | Fixed to 0x00 | Fixed to 0x3FF | Target station side For single CPU 0x3FF fixed For multiple CPUs Connected CPU: 0x3FF No. 1: 0x3E0 No. 2: 0x3E1 No. 3: 0x3E2 No. 4: 0x3E3 | Connected station side relayed module I/O address |
| ActMultiDropChannelNumber | 0 (0x00) | Fixed to 0x00 | | | |
| ActNetworkNumber | 0 (0x00) | Fixed to 0x00 | Target station side module network number | Target station side module network number | Connected station side module network number |
| ActPortNumber | 1 (PORT_1) | Board No. of IBM-PC/AT compatible side MELSECNET/10 board, PORT 1 to PORT 4 (first to fourth boards) | | | |
| ActStationNumber | 255 (0xFF) | Fixed to 0xFF | Target station side module station number | Target station side module station number | Connected station side module station number |
| ActUnitNumber | 0 (0x00) | Fixed to 0x00 | Fixed to 0x00 | Fixed to 0x00 | Target station side module station number |

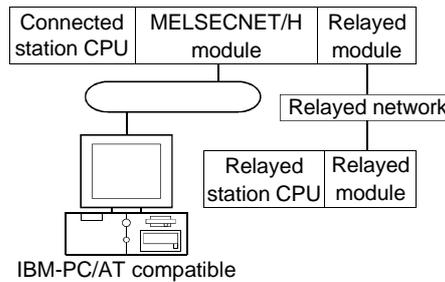
* 1: As the I/O address, specify the value found by dividing the actual first I/O number by 16.

3.3.21 ActMnetHBD, ActMLMnetHBD control

The following table indicates the properties possessed by the ActMnetHBD, ActMLMnetHBD control and their default values.

(1) When access is made via MELSECNET/H mode using QCPU (Q mode) as connected station CPU

(a) Configuration



(b) Property patterns

| Own Board | Connected Station CPU | Relayed Network | Relayed Station CPU *2 | | | | |
|-----------|-----------------------|-----------------|------------------------|---------------|---------|---------|-------|
| | QCPU (Q mode) | | QCPU (Q mode) | QCPU (A mode) | QnA CPU | ACPU *1 | FXCPU |
| ① | ② | MELSECNET/H | ② | × | × | × | × |
| | | MELSECNET/10 | × | ② | ② | ② | × |
| | | MELSECNET(II) | × | × | × | × | × |
| | | Ethernet | ② | × | × | × | × |
| | | Computer link | ③ | × | × | × | × |
| | | CC-Link | ④ | × | × | × | × |

○ : Accessible (Property pattern within circle)

× : Inaccessible

*1 : Including motion controller CPU

*2 : Operates only when QCPU (Q mode) is used.

(c) Property list

| Property | Default Value | Property Patterns | | | |
|------------------------|------------------|--|---------------|---|---|
| | | ① | ② | ③ | ④ |
| ActCpuType | 1025 (CPU_BOARD) | CPU type corresponding to target station | | | |
| ActDestinationIONumber | 0 (0x00) | Fixed to 0x00 | Fixed to 0x00 | Target station side For single CPU 0x3FF fixed For multiple CPUs Connected CPU: 0x3FF No. 1: 0x3E0 No. 2: 0x3E1 No. 3: 0x3E2 No. 4: 0x3E3 | Target station side For single CPU 0x3FF fixed For multiple CPUs Connected CPU: 0x3FF No. 1: 0x3E0 No. 2: 0x3E1 No. 3: 0x3E2 No. 4: 0x3E3 |
| ActDidPropertyBit | 0 (0x00) | Fixed to 0x00 | Fixed to 0x01 | Fixed to 0x00 | Fixed to 0x00 |
| ActDsidPropertyBit | 0 (0x00) | Fixed to 0x00 | Fixed to 0x01 | Fixed to 0x00 | Fixed to 0x00 |

(To next page)

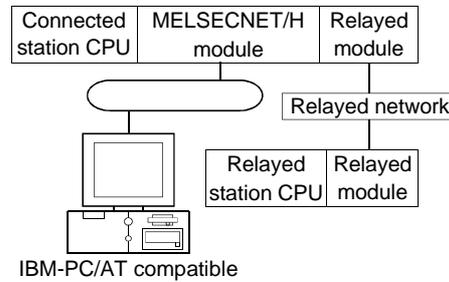
| Property | Default Value | Property Patterns | | | |
|---------------------------|---------------|--|--|---|---|
| | | ① | ② | ③ | ④ |
| ActIONumber * 1 | 0 (0x00) | Fixed to 0x00 | For single CPU 0x3FF fixed For multiple CPUs Connected CPU: 0x3FF No. 1: 0x3E0 No. 2: 0x3E1 No. 3: 0x3E2 No. 4: 0x3E3 | Connected station side relayed module I/O address | Connected station side relayed module I/O address |
| ActMultiDropChannelNumber | 0 (0x00) | Fixed to 0x00 | Fixed to 0x00 | Fixed to 0x02 | Fixed to 0x00 |
| ActNetworkNumber | 0 (0x00) | Fixed to 0x00 | Target station side module network number | Connected station side module network number | Connected station side module network number |
| ActPortNumber | 1 (PORT_1) | Board No. of IBM-PC/AT compatible side MELSECNET/H board, PORT 1 to PORT 4 (first to fourth boards) | | | |
| ActStationNumber | 255 (0xFF) | Fixed to 0xFF | Target station side module station number | Connected station side module station number | Connected station side module station number |
| ActUnitNumber | 0 (0x00) | Fixed to 0x00 | Fixed to 0x00 | Target station side module station number | Target station side module station number |

* 1: As the I/O address, specify the value found by dividing the actual first I/O number by 16.

(2) When access is made via MELSECNET/10 mode using QCPU (Q mode) as connected station CPU

Operates when both the MELSECNET/H board and relayed module are in the MELSECNET/10 mode.

(a) Configuration



(b) Property patterns

| Own Board | Connected Station CPU | Relayed Network | Relayed Station CPU | | | | |
|-----------|-----------------------|-----------------|---------------------|---------------|---------|---------|-------|
| | | | QCPU (Q mode) | QCPU (A mode) | QnA CPU | ACPU *1 | FXCPU |
| ① | ② | MELSECNET/H | × | × | × | × | × |
| | | MELSECNET/10 | ② | ② | ② | ② | × |
| | | MELSECNET(II) | × | × | × | × | × |
| | | Ethernet | ② | × | × | × | × |
| | | Computer link | ③ | × | × | × | × |
| | | CC-Link | ④ | × | × | × | × |

○ : Accessible (Property pattern within circle)
 × : Inaccessible
 *1 : Including motion controller CPU

(c) Property list

| Property | Default Value | Property Patterns | | | |
|------------------------|------------------|--|---------------|---|---|
| | | ① | ② | ③ | ④ |
| ActCpuType | 1025 (CPU_BOARD) | CPU type corresponding to target station | | | |
| ActDestinationIONumber | 0 (0x00) | Fixed to 0x00 | Fixed to 0x00 | Target station side For single CPU 0x3FF fixed For multiple CPUs Connected CPU: 0x3FF No. 1: 0x3E0 No. 2: 0x3E1 No. 3: 0x3E2 No. 4: 0x3E3 | Target station side For single CPU 0x3FF fixed For multiple CPUs Connected CPU: 0x3FF No. 1: 0x3E0 No. 2: 0x3E1 No. 3: 0x3E2 No. 4: 0x3E3 |
| ActDidPropertyBit | 0 (0x00) | Fixed to 0x00 | Fixed to 0x01 | Fixed to 0x00 | Fixed to 0x00 |
| ActDsidPropertyBit | 0 (0x00) | Fixed to 0x00 | Fixed to 0x01 | Fixed to 0x00 | Fixed to 0x00 |

(To next page)

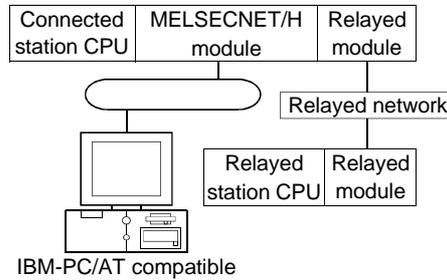
| Property | Default Value | Property Patterns | | | |
|---------------------------|---------------|--|--|---|---|
| | | ① | ② | ③ | ④ |
| ActIONumber * 1 | 0 (0x00) | Fixed to 0x00 | For single CPU 0x3FF fixed For multiple CPUs Connected CPU: 0x3FF No. 1: 0x3E0 No. 2: 0x3E1 No. 3: 0x3E2 No. 4: 0x3E3 | Connected station side relayed module I/O address | Connected station side relayed module I/O address |
| ActMultiDropChannelNumber | 0 (0x00) | Fixed to 0x00 | Fixed to 0x00 | 0x00 or 0x02 | Fixed to 0x00 |
| ActNetworkNumber | 0 (0x00) | Fixed to 0x00 | Target station side module network number | Connected station side module network number | Connected station side module network number |
| ActPortNumber | 1 (PORT_1) | Board No. of IBM-PC/AT compatible side MELSECNET/H board, PORT 1 to PORT 4 (first to fourth boards) | | | |
| ActStationNumber | 255 (0xFF) | Fixed to 0xFF | Target station side module station number | Connected station side module station number | Connected station side module station number |
| ActUnitNumber | 0 (0x00) | Fixed to 0x00 | Fixed to 0x00 | Target station side module station number | Target station side module station number |

* 1: As the I/O address, specify the value found by dividing the actual first I/O number by 16.

(3) When access is made via MELSECNET/10 mode using QCPU (A mode) or ACPU as connected station CPU

Operates when the MELSECNET/H board is in the MELSECNET/10 mode.

(a) Configuration



(b) Property patterns

| Own Board | Connected Station CPU | | Relayed Network | Relayed Station CPU | | | | |
|-----------|-----------------------|----------|-----------------|---------------------|---------------|---------|----------|-------|
| | QCPU (A mode) | ACPU * 1 | | QCPU (Q mode) | QCPU (A mode) | QnA CPU | ACPU * 1 | FXCPU |
| ① | ② | ② | MELSECNET/H | × | × | × | × | × |
| | | | MELSECNET/10 | ③ | ② | ④ | ② | × |
| | | | MELSECNET(II) | × | × | × | × | × |
| | | | Ethernet | × | × | × | × | × |
| | | | Computer link | × | × | × | × | × |
| | | | CC-Link | × | × | × | × | × |

○ : Accessible (Property pattern within circle)

× : Inaccessible

* 1 : Including motion controller CPU

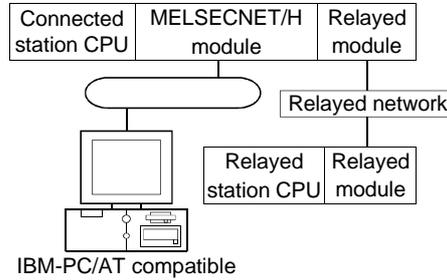
(c) Property list

| Property | Default Value | Property Patterns | | | |
|---------------------------|------------------|---|---|---|----------------|
| | | ① | ② | ③ | ④ |
| ActCpuType | 1025 (CPU_BOARD) | CPU type corresponding to target station | | | |
| ActDestinationIONumber | 0 (0x00) | Fixed to 0x00 | | | |
| ActDidPropertyBit | 0 (0x00) | Fixed to 0x00 | | Fixed to 0x01 | Fixed to 0x00 |
| ActDsidPropertyBit | 0 (0x00) | Fixed to 0x00 | | Fixed to 0x01 | Fixed to 0x00 |
| ActIONumber | 0 (0x00) | Fixed to 0x00 | | Target station side For single CPU 0x3FF fixed For multiple CPUs Connected CPU: 0x3FF No. 1: 0x3E0 No. 2: 0x3E1 No. 3: 0x3E2 No. 4: 0x3E3 | Fixed to 0x3FF |
| ActMultiDropChannelNumber | 0 (0x00) | Fixed to 0x00 | | | |
| ActNetworkNumber | 0 (0x00) | Fixed to 0x00 | Target station side module network number | | |
| ActPortNumber | 1 (PORT_1) | Board No. of IBM-PC/AT compatible side MELSECNET/H board, PORT 1 to PORT 4 (first to fourth boards) | | | |
| ActStationNumber | 255 (0xFF) | Fixed to 0xFF | Target station side module station number | | |
| ActUnitNumber | 0 (0x00) | Fixed to 0x00 | | | |

(4) When access is made via MELSECNET/10 mode using QnACPU as connected station CPU

Operates when the MELSECNET/H board is in the MELSECNET/10 mode.

(a) Configuration



(b) Property patterns

| Own Board | Connected Station CPU | Relayed Network | Relayed Station CPU | | | | |
|-----------|-----------------------|-----------------|---------------------|---------------|---------|---------|-------|
| | QnACPU | | QCPU (Q mode) | QCPU (A mode) | QnA CPU | ACPU *1 | FXCPU |
| ① | ② | MELSECNET/H | × | × | × | × | × |
| | | MELSECNET/10 | ③ | ② | ② | ② | × |
| | | MELSECNET(II) | × | × | × | × | × |
| | | Ethernet | × | × | ② | × | × |
| | | Computer link | × | × | ④ | × | × |
| | | CC-Link | × | × | × | × | × |

○ : Accessible (Property pattern within circle)

× : Inaccessible

* 1 : Including motion controller CPU

(c) Property list

| Property | Default Value | Property Patterns | | | |
|------------------------|---------------------|--|----------------|---|---|
| | | ① | ② * 1 | ③ | ④ |
| ActCpuType | 1025 (CPU_BOARD) | CPU type corresponding to target station | | | |
| ActDestinationIONumber | 0 (0x00) | Fixed to 0x00 | | | |
| ActDidPropertyBit | 0 (0x00) | Fixed to 0x00 | | Fixed to 0x01 | Fixed to 0x00 |
| ActDsidPropertyBit | 0 (0x00) | Fixed to 0x00 | | Fixed to 0x01 | Fixed to 0x00 |
| ActIONumber * 2 | 0 (0x00) | Fixed to 0x00 | Fixed to 0x3FF | Target station side For single CPU 0x3FF fixed For multiple CPUs Connected CPU: 0x3FF No. 1: 0x3E0 No. 2: 0x3E1 No. 3: 0x3E2 No. 4: 0x3E3 | Connected station side relayed module I/O address |

* 1: Note the following for access to be made via the Ethernet module (Q series-compatible E71, QE71).

- Specify ActNetworkNumber and ActStationNumber using the values set in the target station side Q series-compatible E71 or QE71.
- Set "MNET/10 routing information" in the Q series-compatible E71 or QE71 parameter setting. When making setting, specify other than the automatic response system (any of the IP address calculation system, table conversion system and combined system) as the "MNET/10 routing system".

* 2: As the I/O address, specify the value found by dividing the actual first I/O number by 16.

(To next page)

| Property | Default Value | Property Patterns | | | |
|---------------------------|---------------|--|---|---|--|
| | | ① | ② * 1 | ③ | ④ |
| ActMultiDropChannelNumber | 0 (0x00) | Fixed to 0x00 | | | |
| ActNetworkNumber | 0 (0x00) | Fixed to 0x00 | Target station side module network number | Target station side module network number | Connected station side module network number |
| ActPortNumber | 1 (PORT_1) | Board No. of IBM-PC/AT compatible side MELSECNET/H board, PORT 1 to PORT 4 (first to fourth boards) | | | |
| ActStationNumber | 255 (0xFF) | Fixed to 0xFF | Target station side module station number | Target station side module station number | Connected station side module station number |
| ActUnitNumber | 0 (0x00) | Fixed to 0x00 | Fixed to 0x00 | Fixed to 0x00 | Target station side module station number |

* 1: Note the following for access to be made via the Ethernet module (Q series-compatible E71, QE71).

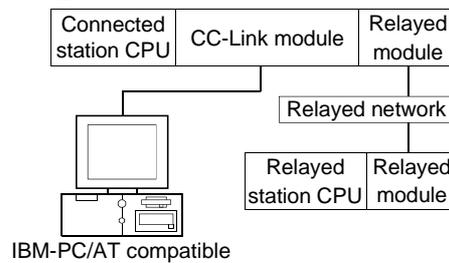
- Specify ActNetworkNumber and ActStationNumber using the values set in the target station side Q series-compatible E71 or QE71.
- Set "MNET/10 routing information" in the Q series-compatible E71 or QE71 parameter setting. When making setting, specify other than the automatic response system (any of the IP address calculation system, table conversion system and combined system) as the "MNET/10 routing system".

3.3.22 ActCCBD, ActMLCCBD control

The following table indicates the properties possessed by the ActCCBD, ActMLCCBD control and their default values.

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

(a) Configuration



(b) Property patterns

| Own Board | Connected Station CPU | Relayed Network | Relayed Station CPU | | | | |
|-----------|-----------------------|-----------------|---------------------|---------------|---------|---------|-------|
| | | | QCPU (Q mode) | QCPU (A mode) | QnA CPU | ACPU *1 | FXCPU |
| ① | ② | MELSECNET/H | ③ | × | × | × | × |
| | | MELSECNET/10 | ③ | × | × | × | × |
| | | MELSECNET(II) | × | × | × | × | × |
| | | Ethernet | ③ | × | × | × | × |
| | | Computer link | × | × | × | × | × |
| | | CC-Link | × | × | × | × | × |

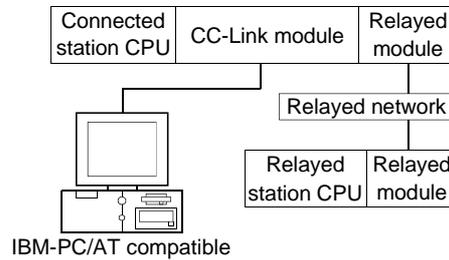
○ : Accessible (Property pattern within circle)
 × : Inaccessible
 *1 : Including motion controller CPU

(c) Property list

| Property | Default Value | Property Patterns | | |
|------------------------|------------------|---|---|---|
| | | ① | ② | ③ |
| ActCpuType | 1025 (CPU_BOARD) | CPU type corresponding to target station | | |
| ActDestinationIONumber | 0 (0x00) | Fixed to 0x00 | Target station side For single CPU 0x3FF fixed For multiple CPUs Connected CPU: 0x3FF No. 1: 0x3E0 No. 2: 0x3E1 No. 3: 0x3E2 No. 4: 0x3E3 | Target station side For single CPU 0x3FF fixed For multiple CPUs Connected CPU: 0x3FF No. 1: 0x3E0 No. 2: 0x3E1 No. 3: 0x3E2 No. 4: 0x3E3 |
| ActIONumber | 0 (0x00) | Fixed to 0x00 | Fixed to 0x3FF | Fixed to 0x3FF |
| ActNetworkNumber | 0 (0x00) | Fixed to 0x00 | Fixed to 0x00 | Target station side module network number |
| ActPortNumber | 1 (PORT_1) | Board No. of IBM-PC/AT compatible side CC-Link board, PORT 1 to PORT 4 (first to fourth boards) | | |
| ActStationNumber | 255 (0xFF) | Fixed to 0xFF | Target station side CC-Link module station number | Target station side module station number |
| ActUnitNumber | 0 (0x00) | Fixed to 0x00 | Fixed to 0x00 | Connected station side module station number |

(2) When connected station CPU is QCPU (A mode)

(a) Configuration



(b) Property patterns

| Own Board | Connected Station CPU | Relayed Network | Relayed Station CPU | | | | |
|-----------|-----------------------|-----------------|---------------------|---------------|---------|---------|-------|
| | QCPU (A mode) | | QCPU (Q mode) | QCPU (A mode) | QnA CPU | ACPU *1 | FXCPU |
| ① | ② | MELSECNET/H | × | × | × | × | × |
| | | MELSECNET/10 | × | × | × | × | × |
| | | MELSECNET(II) | × | × | × | × | × |
| | | Ethernet | × | × | × | × | × |
| | | Computer link | × | × | × | × | × |
| | | CC-Link | × | × | × | × | × |

○ : Accessible (Property pattern within circle)

× : Inaccessible

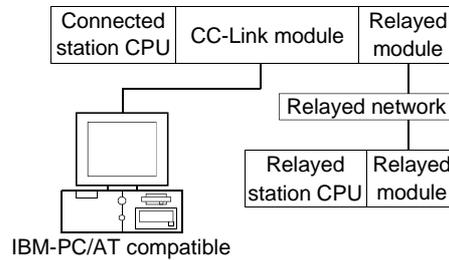
*1 : Including motion controller CPU

(c) Property list

| Property | Default Value | Property Patterns | |
|------------------------|---------------------|--|--|
| | | ① | ② |
| ActCpuType | 1025 (CPU_BOARD) | CPU type corresponding to target station | |
| ActDestinationIONumber | 0 (0x00) | Fixed to 0x00 | |
| ActIONumber | 0 (0x00) | Fixed to 0x00 | |
| ActNetworkNumber | 0 (0x00) | Fixed to 0x00 | |
| ActPortNumber | 1 (PORT_1) | Board No. of IBM-PC/AT compatible side CC-Link board, PORT 1 to PORT 4 (first to fourth boards) | |
| ActStationNumber | 255 (0xFF) | Fixed to 0xFF | Target station side CC-Link module station number |
| ActUnitNumber | 0 (0x00) | Fixed to 0x00 | |

(3) When connected station CPU is QnACPU

(a) Configuration



(b) Property patterns

| Own Board | Connected Station CPU | Relayed Network | Relayed Station CPU | | | | |
|-----------|-----------------------|-----------------|---------------------|---------------|---------|---------|-------|
| | QnACPU | | QCPU (Q mode) | QCPU (A mode) | QnA CPU | ACPU *1 | FXCPU |
| ① | ② | MELSECNET/H | × | × | × | × | × |
| | | MELSECNET/10 | × | × | ⓐ | × | × |
| | | MELSECNET(II) | × | × | × | × | × |
| | | Ethernet | × | × | ⓐ | × | × |
| | | Computer link | × | × | × | × | × |
| | | CC-Link | × | × | × | × | × |

ⓐ : Accessible (Property pattern within circle)

× : Inaccessible

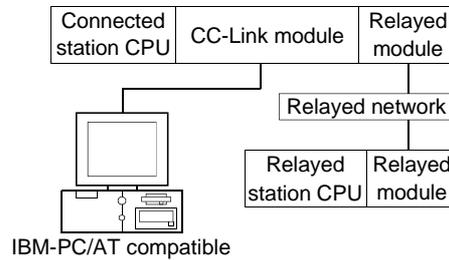
*1 : Including motion controller CPU

(c) Property list

| Property | Default Value | Property Patterns | | |
|------------------------|------------------|---|---|---|
| | | ① | ② | ③ |
| ActCpuType | 1025 (CPU_BOARD) | CPU type corresponding to target station | | |
| ActDestinationIONumber | 0 (0x00) | Fixed to 0x00 | | |
| ActIONumber | 0 (0x00) | Fixed to 0x3FF | | |
| ActNetworkNumber | 0 (0x00) | Fixed to 0x00 | Fixed to 0x00 | Target station side module network number |
| ActPortNumber | 1 (PORT_1) | Board No. of IBM-PC/AT compatible side CC-Link board, PORT 1 to PORT 4 (first to fourth boards) | | |
| ActStationNumber | 255 (0xFF) | Fixed to 0xFF | Target station side CC-Link module station number | Target station side module station number |
| ActUnitNumber | 0 (0x00) | Fixed to 0x00 | Fixed to 0x00 | Target station side CC-Link module station number |

(4) When connected station CPU is ACPU

(a) Configuration



(b) Property patterns

| Own Board | Connected Station CPU | Relayed Network | Relayed Station CPU | | | | |
|-----------|-----------------------|-----------------|---------------------|---------------|---------|----------|-------|
| | ACPU * 1 | | QCPU (Q mode) | QCPU (A mode) | QnA CPU | ACPU * 1 | FXCPU |
| ① | ② | MELSECNET/H | × | × | × | × | × |
| | | MELSECNET/10 | × | × | × | × | × |
| | | MELSECNET(II) | × | × | × | × | × |
| | | Ethernet | × | × | × | × | × |
| | | Computer link | × | × | × | × | × |
| | | CC-Link | × | × | × | × | × |

○ : Accessible (Property pattern within circle)

× : Inaccessible

* 1 : Including motion controller CPU

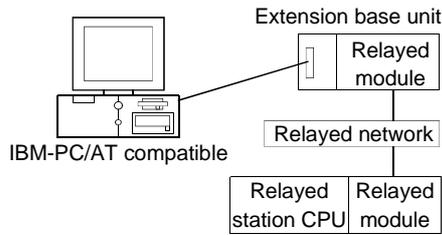
(c) Property list

| Property | Default Value | Property Patterns | |
|------------------------|---------------------|--|---|
| | | ① | ② |
| ActCpuType | 1025 (CPU_BOARD) | CPU type corresponding to target station | |
| ActDestinationIONumber | 0 (0x00) | Fixed to 0x00 | |
| ActIONumber | 0 (0x00) | Fixed to 0x00 | |
| ActNetworkNumber | 0 (0x00) | Fixed to 0x00 | |
| ActPortNumber | 1 (PORT_1) | Board No. of IBM-PC/AT compatible side CC-Link board, PORT 1 to PORT 4 (first to fourth boards) | |
| ActStationNumber | 255 (0xFF) | Fixed to 0xFF | Target station side module station number |
| ActUnitNumber | 0 (0x00) | Fixed to 0x00 | |

3.3.23 ActAnUBD, ActMLAnUBD control

The following table indicates the properties possessed by the ActAnUBD, ActMLAnUBD control and their default values.

(1) Configuration



(2) Property patterns

| Connected Station CPU | Relayed Network | Relayed Station CPU | | | | |
|-----------------------|-----------------|---------------------|---------------|---------|---------|-------|
| | | QCPU (Q mode) | QCPU (A mode) | QnA CPU | ACPU *1 | FXCPU |
| ① | MELSECNET/H | × | × | × | × | × |
| | MELSECNET/10 | × | ② | ② *2 | ② | × |
| | MELSECNET(II) | × | ③ | ③ *2 | ③ | × |
| | Ethernet | × | × | × | × | × |
| | Computer link | × | × | × | × | × |
| | CC-Link | × | × | × | × | × |

○ : Accessible (Property pattern within circle)
 × : Inaccessible
 *1 : Including motion controller CPU
 *2 : Operates as the one equivalent to AnACPU.

(3) Property list

| Property | Default Value | Property Patterns | | |
|------------------|-----------------------------|--|---|---|
| | | ① | ② | ③ |
| ActCpuType | 271 (CPU _A2USHS1CPU) | CPU type corresponding to target station | | |
| ActNetworkNumber | 0 (0x00) | Fixed to 0x00 | Target station side module network number | Fixed to 0x00 |
| ActStationNumber | 255 (0xFF) | Fixed to 0xFF | Target station side module station number | Target station side module station number |

3.3.24 ActLLT, ActMLLLT control

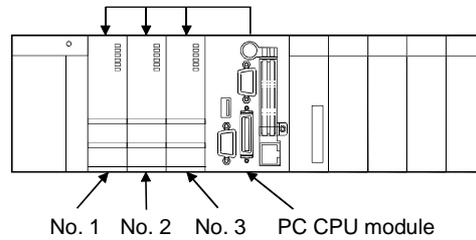
The following table indicates the properties possessed by the ActLLT, ActMLLLT control and their default values.

| Property | Default Value | Property Pattern |
|------------|--------------------|--|
| ActCpuType | 34 (CPU_Q02CPU) | CPU type corresponding to target station |
| ActTimeOut | 10000 | Any value specified by user in ms units |

3.3.25 ActQCPUQBus, ActMLQCPUQBus control

The following table indicates the properties possessed by the ActQCPUQBus, ActMLQCPUQBus control and their default values.

(1) Configuration



(2) Property

| Property | Default Value | Property Pattern |
|-------------|--------------------|--|
| ActCpuType | 34 (CPU_Q02CPU) | CPU type corresponding to target station |
| ActIONumber | 992 (0x3E0) | Target station No. 1: 0x3E0 No. 2: 0x3E1 No. 3: 0x3E2 |

POINT

Use any of the following controls for access to the PLC CPU via the corresponding module controlled by the PC CPU module.

| Relayed Module | Controls Used | |
|--------------------|-------------------|--------------|
| | For VB, VC++, VBA | For VBScript |
| MELSECNET/H module | ActMnetHBD | ActMLMnetHBD |
| CC-Link module | ActCCBD | ActMLCCBD |

4 FUNCTIONS

This chapter provides the programming instructions and function details (dispatch interface, custom interface).

| |
|--------------|
| POINT |
|--------------|

- | |
|--|
| <p>(1) For interface selection, we recommend you to choose the dispatch interface which is simpler.</p> <p>(2) For programming, refer to "Section 4.1 Programming Instructions".</p> |
|--|

4.1 Programming Instructions

This section gives the instructions for programming.

(1) Instructions common to Microsoft® Visual Basic® and Microsoft® Visual C++®

(a) Multithread

When performing multithread programming, follow the rules of COM and ActiveX controls.

For details, refer to the rules and reference books of COM and ActiveX controls.

| |
|--------------|
| POINT |
|--------------|

- | |
|--|
| <p>(1) The ActiveX controls used on MX Component are those of the STA model.</p> <p>(2) When passed to another apartment, the interface pointer must be marshaled. Provide synchronization using the CoMarshalerThreadInterfaceInStream or CoGetInterfaceAndReleadseStream COM function.</p> |
|--|

(2) Instructions for use of Visual Basic®

Only the dispatch interface is usable.

(3) Instructions for use of Visual C++® (dispatch interface, custom interface)

(a) Both the dispatch interface and custom interface are usable.

(b) BSTR* type

In the functions which acquire the methods and properties using the BSTR pointer type, memory must be secured inside the ActiveX controls and released in user programs. (This is based on the rules of COM and ActiveX controls.)

(Example)

```
BSTR  szCpuName;
```

```
LONG  ICpuCode;
```

```
Obj.GetCpuType(&szCpuName, &ICpuCode );
```

```
MessgBox( "CpuName = %s, CpuCode = %d", szCpuName, ICpuCode );
```

```
SysFreeString( szCpuName );
```

(4) Instructions for use of Visual C++® (custom interface)

(a) HRESULT type

Use the SUCCEEDED or FAILED macro to check whether the HRESULT type, i.e. returned value of COM, resulted in normal or abnormal termination.

(Example)

```
HRESULT    hResult;
LONG       IRet;
hResult = Obj.Open( &IRet );
if( SUCCEEDED( hResult ) ) {
    if( IRet = SUCCESS ) {

        } else {
            MessgeBox( "Communication Error = %x", IRet );
        }
} else {
    MessgeBox( "COM ERROR Occurd" );
}
```

4.2 Details of the Functions (Dispatch Interface)

This section explains the details of the functions.
 The details of the functions in this section assume that the dispatch interface is used.
 For the custom interface, refer to "Section 4.3 Details of the Functions (Custom Interface)".

4.2.1 Open (Communication line opening)

(1) Applicable ACT controls

This function is available for all ACT controls.

(2) Feature

Opens the communication line.

(3) Format

| | | | |
|-------------------------------------|------------------------|--------|--|
| Visual Basic® , Visual C++® , VBA : | IRet = object.Open() | | |
| | Long | IRet | Returned value Output |
| VBScript : | varRet = object.Open() | | |
| | VARIANT | varRet | Returned value (LONG type) Output |

(4) Explanation

The line is connected on the basis of the value set to the property for Open method.

(5) Returned value

Normal termination : 0 is returned.
 Abnormal termination : A value other than 0 is returned.
 (Refer to Chapter 6 ERROR CODES.)

| POINT |
|--|
| <p>(1) If the property for Open method is changed after completion of Open, the other end of communication is not changed. To change the communication settings, close the communication line once, then set the other end of communication, and open the communication line again.</p> <p>(2) Open may terminate normally if the CPU type entered into the ActCpuType property is different from the CPU used for communication. In such a case, the connection range, usable methods and device range may be reduced, for example. When performing Open, set the correct CPU type to the ActCpuType property.</p> |

4.2.2 Close (Communication line closing)

(1) Applicable ACT controls

This function is available for all ACT controls.

(2) Feature

Closes the communication line.

(3) Format

Visual Basic[®] , Visual C++[®] , VBA : IRet = object.Close()

| | | | |
|------|------|----------------|--------|
| Long | IRet | Returned value | Output |
|------|------|----------------|--------|

VBScript : varRet = object.Close()

| | | | |
|---------|--------|----------------------------|--------|
| VARIANT | varRet | Returned value (LONG type) | Output |
|---------|--------|----------------------------|--------|

(4) Explanation

The line connected using the Open function is closed.

(5) Returned value

Normal termination : 0 is returned.

Abnormal termination : A value other than 0 is returned.

(Refer to Chapter 6 ERROR CODES.)

4.2.3 ReadDeviceBlock (Device batch-read)

(1) Applicable ACT controls

This function is available for all ACT controls.

(2) Feature

Batch-reads data from devices.

(3) Format

Visual Basic[®], VBA : IRet = object.ReadDeviceBlock(szDevice, ISize, IData(0))

| | | | |
|--------|----------|-----------------------|--------|
| Long | IRet | Returned value | Output |
| String | szDevice | Device name | Input |
| Long | ISize | Number of read points | Input |
| Long | IData(n) | Read device values | Output |

Visual C++[®] : IRet = object.ReadDeviceBlock(szDevice, ISize, *lpIData)

| | | | |
|---------|----------|-----------------------|--------|
| Long | IRet | Returned value | Output |
| CString | szDevice | Device name | Input |
| Long | ISize | Number of read points | Input |
| Long | *lpIData | Read device values | Output |

VBScript : varRet = object.ReadDeviceBlock(varDevice, varSize, lpvarData)

| | | | |
|---------|-----------|-------------------------------------|--------|
| VARIANT | varRet | Returned value (LONG type) | Output |
| VARIANT | varDevice | Device name(character string type) | Input |
| VARIANT | varSize | Number of read points(LONG type) | Input |
| VARIANT | lpvarData | Read device values(LONG array type) | Output |

(4) Explanation

- The device values for ISize(varSize) are batch-read from the devices, beginning with the device specified in szDevice(varDevice).
- The read device values are stored in IData (lpIData or lpvarData).
- Reserve an array of ISize (varSize) or more for IData (lpIData or lpvarData).

(5) Device specifying methods

Specify the devices in the following methods.

<When bit device is specified>

(Example) 3 points from M0

| 2 Upper Bytes | 2 Lower Bytes |
|---------------|---------------|
| *1 | M0 to M15*2 |
| *1 | M16 to M31*2 |
| *1 | M32 to M47*2 |

<When word device is specified>

(Example) 3 points from D0

| 2 Upper Bytes | 2 Lower Bytes |
|---------------|---------------|
| *1 | D0 |
| *1 | D1 |
| *1 | D3 |

<When CN200 and later of FXCPU are specified>

(Example) 6 points from CN200 *3:

| 2 Upper Bytes | 2 Lower Bytes |
|---------------|---------------|
| *1 | L of CN200 |
| *1 | H of CN200 |
| *1 | L of CN201 |
| *1 | H of CN201 |
| *1 | L of CN202 |
| *1 | H of CN202 |

<When FD device is specified (4-word device)>

(Example) 6 points from FD0

| 2 Upper Bytes | 2 Lower Bytes |
|---------------|---------------|
| *1 | LL of FD0 |
| *1 | LH of FD0 |
| *1 | HL of FD0 |
| *1 | HH of FD0 |
| *1 | LL of FD1 |
| *1 | LH of FD1 |

*1: Not used. (0 is stored.)

*2: Lower bits are stored in device number order.

*3: For CN200 or later of FXCPU, 2 words are read from 2 points. Read from 1 point will result in an error.

(6) Returned value

Normal termination : 0 is returned.

Abnormal termination : Any value other than 0 is returned.

(Refer to Chapter 6 ERROR CODES.)

| POINT |
|--|
| (1) The maximum number of read points that may be specified in ISize(varSize) should satisfy the following range. Read starting device number + number of read points ≤ last device number |
| (2) When the bit device is specified, a multiple of 16 may be specified as the device number. |
| (3) For IData (IplData or IpvData), prepare a memory area having the number of points specified in ISize (varSize). If there is no memory area, a critical phenomenon such as an application error may occur. |

4.2.4 WriteDeviceBlock (Device batch-write)

(1) Applicable ACT controls

This function is available for all ACT controls.

(2) Feature

Batch-writes data to devices.

(3) Format

Visual Basic[®], VBA : IRet = object.WriteDeviceBlock(szDevice, ISize, IData(0))

| | | | |
|--------|----------|-----------------------------|--------|
| Long | IRet | Returned value | Output |
| String | szDevice | Device name | Input |
| Long | ISize | Number of write points | Input |
| Long | IData(n) | Device values to be written | Input |

Visual C++[®] : IRet = object.WriteDeviceBlock(szDevice, ISize, *lpIData)

| | | | |
|---------|----------|-----------------------------|--------|
| Long | IRet | Returned value | Output |
| CString | szDevice | Device name | Input |
| Long | ISize | Number of write points | Input |
| Long | *lpIData | Device values to be written | Input |

VBScript : varRet = object.WriteDeviceBlock(varDevice, varSize, varData)

| | | | |
|---------|-----------|--|--------|
| VARIANT | varRet | Returned value(LONG type) | Output |
| VARIANT | varDevice | Device name(character string type) | Input |
| VARIANT | varSize | Number of write points(LONG type) | Input |
| VARIANT | varData | Device values to be written (LONG array type) | Input |

(4) Explanation

- The device values for ISize(varSize) are batch-written to the devices, beginning with the device specified in szDevice(varDevice).
- Store the device values to be written in IData (lpIData or varData).
- Reserve an array of ISize (varSize) or more for IData (lpIData or varData).

(5) Device specifying methods

Specify the devices in the following methods.

<When bit device is specified>

(Example) 3 points from M0

| 2 Upper Bytes | 2 Lower Bytes |
|---------------|---------------|
| *1 | M0 to M15 *2 |
| *1 | M16 to M31 *2 |
| *1 | M32 to M47 *2 |

<When word device is specified>

(Example) 3 points from D0

| 2 Upper Bytes | 2 Lower Bytes |
|---------------|---------------|
| *1 | D0 |
| *1 | D1 |
| *1 | D2 |

<When CN200 and later of FXCPU are specified>

(Example) 6 points from CN200 *3:

| 2 Upper Bytes | 2 Lower Bytes |
|---------------|---------------|
| *1 | L of CN200 |
| *1 | H of CN200 |
| *1 | L of CN201 |
| *1 | H of CN201 |
| *1 | L of CN202 |
| *1 | H of CN202 |

<When FD device is specified (4-word device)>

(Example) 6 points from FD0

| 2 Upper Bytes | 2 Lower Bytes |
|---------------|---------------|
| *1 | LL of FD0 |
| *1 | LH of FD0 |
| *1 | HL of FD0 |
| *1 | HH of FD0 |
| *1 | LL of FD1 |
| *1 | LH of FD1 |

*1: Not used. (0 is stored.)

*2: Lower bits are stored in device number order.

*3: For CN200 or later of FXCPU, 2 words are written from 2 points. Write from 1 point will result in an error.

(6) Returned value

Normal termination : 0 is returned.

Abnormal termination : Any value other than 0 is returned.

(Refer to Chapter 6 ERROR CODES.)

| POINT |
|--|
| (1) The maximum number of write points that may be specified in ISize(varSize) should satisfy the following range. Write starting device number + number of write points ≤ last device number |
| (2) When the bit device is specified, a multiple of 16 may be specified as the device number. |
| (3) For IData (IplData or varData), prepare a memory area having the number of points specified in ISize (varSize). If there is no memory area, a critical phenomenon such as an application error may occur. |

4.2.5 ReadDeviceRandom (Device random-read)

(1) Applicable ACT controls

This function is available for all ACT controls.

(2) Feature

Reads data randomly from devices.

(3) Format

Visual Basic[®], VBA : IRet = object.ReadDeviceRandom(szDeviceList, ISize, IData(0))

| | | | |
|--------|--------------|-----------------------|--------|
| Long | IRet | Returned value | Output |
| String | szDeviceList | Device name | Input |
| Long | ISize | Number of read points | Input |
| Long | IData(n) | Read device values | Output |

Visual C++[®] : IRet = object.ReadDeviceRandom(szDeviceList, ISize, *lpIData)

| | | | |
|---------|--------------|-----------------------|--------|
| Long | IRet | Returned value | Output |
| CString | szDeviceList | Device name | Input |
| Long | ISize | Number of read points | Input |
| Long | *lpIData | Read device values | Output |

VBScript : varRet = object.ReadDeviceRandom(varDeviceList, varSize, IpvarData)

| | | | |
|---------|--------------|---|--------|
| VARIANT | IRet | Returned value(LONG type) | Output |
| VARIANT | szDeviceList | Device name(character string type) | Input |
| VARIANT | ISize | Number of read points(LONG type) | Input |
| VARIANT | IData(n) | Read device values (LONG array type) | Output |

(4) Explanation

- The device values for ISize(varSize) are read from the device group specified in szDeviceList(varDeviceList).
- The read device values are stored in IData (lpIData or IpvarData).
- Using the line feed symbol, separate the devices in the character string specified in the device list.

The last device need not be followed by the line feed symbol.

(Example)

Visual Basic[®], VBA, VBScript : "D0" & vbLf & "D1" & vbLf & "D2"

Visual C++[®] : D0\nD1\nD2

- Reserve an array of ISize (varSize) or more for IData (lpIData or IpvarData).

(5) Device specifying methods

Specify the devices in the following methods.

(Example 1) When devices are specified as follows (3 points)

When using Visual Basic®, VBA, VBScript : M0 & vbLf & D0 & vbLf & K8M0

When using Visual C++® : M0\nD0\nK8M0

| 2 Upper Bytes | 2 Lower Bytes |
|---------------|---------------|
| *1 | M0 |
| *1 | D0 |
| M16 to M31 *2 | M0 to M15 *2 |

(Example 2) When devices including CN200 and later of FXCPU are specified

(3 points in all) *3

When using Visual Basic®, VBA, VBScript : D0 & vbLf & CN200 & vbLf & D1

When using Visual C++® : D0\nCN200\nD1

| 2 Upper Bytes | 2 Lower Bytes |
|---------------|---------------|
| *1 | D0 |
| H of CN200 | L of CN200 |
| *1 | D1 |

(Example 3) When devices including FD are specified (3 points in all)

When using Visual Basic®, VBA, VBScript : D0 & vbLf & FD0 & vbLf & D1

When using Visual C++® : D0\nFD0\nD1

| 2 Upper Bytes | 2 Lower Bytes |
|---------------|---------------|
| *1 | D0 |
| *1 | LL of FD |
| *1 | D1 |

*1: Not used. (0 is stored.)

*2: Lower bits are stored in device number order.

*3: For CN200 or later of FXCPU, 2 words are read from 1 point by random read.

(6) Returned value

Normal termination : 0 is returned.

Abnormal termination : Any value other than 0 is returned.

(Refer to Chapter 6 ERROR CODES.)

| POINT |
|--|
| (1) The maximum number of read points that may be specified in ISize(varSize) is up to 0x7FFFFFFF points. |
| (2) For IData(lpData or lparData), prepare a memory area having the number of points specified in ISize(varSize). If there is no memory area, a critical phenomenon such as an application error may occur. |

4.2.6 WriteDeviceRandom (Device random-write)

(1) Applicable ACT controls

This function is available for all ACT controls.

(2) Feature

Writes data randomly to devices.

(3) Format

Visual Basic[®], VBA : IRet = object.WriteDeviceRandom(szDeviceList, ISize, IData(0))

| | | | |
|--------|--------------|-----------------------------|--------|
| Long | IRet | Returned value | Output |
| String | szDeviceList | Device name | Input |
| Long | ISize | Number of write points | Input |
| Long | IData(n) | Device values to be written | Input |

Visual C++[®] : IRet = object.WriteDeviceRandom(szDeviceList, ISize, *lpIData)

| | | | |
|---------|--------------|-----------------------------|--------|
| Long | IRet | Returned value | Output |
| CString | szDeviceList | Device name | Input |
| Long | ISize | Number of write points | Input |
| Long | *lpIData | Device values to be written | Input |

VBScript : varRet = object.WriteDeviceRandom(varDeviceList, varSize, varData)

| | | | |
|---------|---------------|---|--------|
| VARIANT | varRet | Returned value (LONG type) | Output |
| VARIANT | varDeviceList | Device name (character string type) | Input |
| VARIANT | varSize | Number of write points (LONG type) | Input |
| VARIANT | varData | Device values to be written (LONG array type) | Input |

(4) Explanation

- The device values for ISize are written to the devices specified in szDeviceList.
- The read device values are stored in IData (lpIData or varData).
- Using the line feed symbol, separate the devices in the character string specified in the device list.

The last device need not be followed by the line feed symbol.

(Example)

Visual Basic[®], VBA, VBScript : "D0" & vbLf & "D1" & vbLf & "D2"

Visual C++[®] : D0\nD1\nD2

- Reserve an array of ISize (varSize) or more for IData (lpIData or varData).

(5) Device specifying methods

Specify the devices in the following methods.

(Example 1) When devices are specified as follows (3 points)

When using Visual Basic[®], VBA, VBScript : M0 & vbLf & D0 & vbLf & K8M0

When using Visual C++[®] : M0\nD0\nK8M0

| 2 Upper Bytes | 2 Lower Bytes |
|---------------|---------------|
| *1 | M0 |
| *1 | D0 |
| M16 to M31 *2 | M0 to M15 *2 |

(Example 2) When devices including CN200 and later of FXCPU are specified

(3 points in all) *3

When using Visual Basic[®], VBA, VBScript : D0 & vbLf & CN200 & vbLf & D1

When using Visual C++[®] : D0\nCN200\nD1

| 2 Upper Bytes | 2 Lower Bytes |
|---------------|---------------|
| *1 | D0 |
| H of CN200 | L of CN200 |
| *1 | D1 |

(Example 3) When devices including FD are specified (3 points in all)

When using Visual Basic[®], VBA, VBScript : D0 & vbLf & FD0 & vbLf & D1

When using Visual C++[®] : D0\nFD0\nD1

| 2 Upper Bytes | 2 Lower Bytes |
|---------------|---------------|
| *1 | D0 |
| *1 | LL of FD |
| *1 | D1 |

*1: Not used. (0 is stored.)

*2: Lower bits are stored in device number order.

*3: For CN200 or later of FXCPU, 2 words are read from 1 point by random read.

(6) Returned value

Normal termination : 0 is returned.

Abnormal termination : Any value other than 0 is returned.

(Refer to Chapter 6 ERROR CODES.)

| POINT |
|--|
| (1) The maximum number of write points that may be specified in ISize(varSize) is up to 0x7FFFFFFF points. |
| (2) For IData(lplData or varData), prepare a memory area having the number of points specified in ISize(varSize). If there is no memory area, a critical phenomenon such as an application error may occur. |

4.2.7 SetDevice (Device data setting)

(1) Applicable ACT controls

This function is available for all ACT controls.

(2) Feature

Sets one point of device.

(3) Format

Visual Basic® , VBA : IRet = object.SetDevice(szDevice, IData)

| | | | |
|--------|----------|----------------|--------|
| Long | IRet | Returned value | Output |
| String | szDevice | Device name | Input |
| Long | IData | Set data | Input |

Visual C++® : IRet = object.SetDevice(szDevice, *lpIData)

| | | | |
|---------|----------|----------------|--------|
| Long | IRet | Returned value | Output |
| CString | szDevice | Device name | Input |
| Long | *lpIData | Set data | Input |

VBScript : varRet = object.SetDevice(varDevice, lpvarData)

| | | | |
|---------|-----------|--|--------|
| VARIANT | varRet | Returned value (LONG type) | Output |
| VARIANT | varDevice | Device name (character string type) | Input |
| VARIANT | varData | Set data (LONG type) | Input |

(4) Explanation

- The operation specified in IData(lpIData or varData) is performed for one point of device specified in szDevice(varDevice).
- When the bit device is specified, the least significant bit of the IData value (lpIData value or varData value) becomes valid.

(5) Device specifying methods

Specify the devices in the following methods.

<When bit device is specified>

(Example) M0

| | |
|---------------|---------------|
| 2 Upper Bytes | 2 Lower Bytes |
| *1 | M0 |

<When word device is specified>

(Example) D0

| | |
|---------------|---------------|
| 2 Upper Bytes | 2 Lower Bytes |
| *1 | D0 |

<When double-word device is specified>

(Example) K8M0

| | |
|---------------|---------------|
| 2 Upper Bytes | 2 Lower Bytes |
| M16 to M31 *2 | M0 to M15 *2 |

<When CN200 or later of FXCPU is specified>

(Example) CN200

| | |
|---------------|---------------|
| 2 Upper Bytes | 2 Lower Bytes |
| H of CN200 | L of CN200 |

*1: Not used. (0 is stored.)

*2: Lower bits are stored in device number order.

(6) Returned value

Normal termination : 0 is returned.

Abnormal termination : Any value other than 0 is returned.

(Refer to Chapter 6 ERROR CODES.)

4.2.8 GetDevice (Device data acquisition)

(1) Applicable ACT controls

This function is available for all ACT controls.

(2) Feature

Acquires data from one point of device.

(3) Format

Visual Basic[®], VBA : IRet = object.GetDevice(szDevice, IData)

| | | | |
|--------|----------|----------------|--------|
| Long | IRet | Returned value | Output |
| String | szDevice | Device name | Input |
| Long | IData | Acquired data | Output |

Visual C++[®] : IRet = object.GetDevice(szDevice, *lpIData)

| | | | |
|---------|----------|----------------|--------|
| Long | IRet | Returned value | Output |
| CString | szDevice | Device name | Input |
| Long | *lpIData | Acquired data | Output |

VBScript : varRet = object.GetDevice(varDevice, lpvarData)

| | | | |
|---------|-----------|--|--------|
| VARIANT | varRet | Returned value (LONG type) | Output |
| VARIANT | varDevice | Device name (character string type) | Input |
| VARIANT | lpvarData | Acquired data (LONG type) | Output |

(4) Explanation

The data of one point of device specified in szDevice(varDevice) is stored into IData(lpIData or lpvarData).

(5) Device specifying methods

Specify the devices in the following methods.

<When bit device is specified>

(Example) M0

| | |
|---------------|---------------|
| 2 Upper Bytes | 2 Lower Bytes |
| *1 | M0 |

<When word device is specified>

(Example) D0

| | |
|---------------|---------------|
| 2 Upper Bytes | 2 Lower Bytes |
| *1 | D0 |

<When double-word device is specified>

(Example) K8M0

| | |
|---------------|---------------|
| 2 Upper Bytes | 2 Lower Bytes |
| M16 to M31 *2 | M0 to M15 *2 |

<When CN200 or later of FXCPU is specified>

(Example) CN200

| | |
|---------------|---------------|
| 2 Upper Bytes | 2 Lower Bytes |
| H of CN200 | L of CN200 |

*1: Not used. (0 is stored.)

*2: Lower bits are stored in device number order.

(6) Returned value

Normal termination : 0 is returned.

Abnormal termination : Any value other than 0 is returned.

(Refer to Chapter 6 ERROR CODES.)

4.2.9 ReadBuffer (Buffer memory read)

(1) Applicable ACT controls

The applicable ACT controls are indicated below.

| Control Name | Usability | Control Name | Usability |
|----------------------------------|-----------|----------------------------------|-----------|
| ActEasyIF, ActMLEasyIF | ○*7 | ActAJ71QE71UDP, ActMLAJ71QE71UDP | ○*1, *2 |
| ActQCPUQ, ActMLQCPUQ | ○ | ActAJ71E71TCP, ActMLAJ71E71TCP | ○*1 |
| ActQCPUA, ActMLQCPUA | ○ | ActAJ71E71UDP, ActMLAJ71E71UDP | ○*1 |
| ActQnACPU, ActMLQnACPU | ○ | ActQCPUQUSB, ActMLQCPUQUSB | ○ |
| ActACPU, ActMLACPU | ○ | ActCCG4QnA, ActMLCCG4QnA | ○ |
| ActFXCPU, ActMLFXCPU | ○*4 | ActCCG4A, ActMLCCG4A | ○ |
| ActQJ71C24, ActMLQJ71C24 | ○ | ActMnet10BD, ActMLMnet10BD | ○*3 |
| ActAJ71QC24, ActMLAJ71QC24 | ○ | ActMnetHBD, ActMLMnetHBD | ○*3 |
| ActAJ71UC24, ActMLAJ71UC24 | × | ActCCBD, ActMLCCBD | ○*3 |
| ActAJ71C24, ActMLAJ71C24 | × | ActAnUBD, ActMLAnUBD | ○*6 |
| ActQJ71E71TCP, ActMLQJ71E71TCP | ○ | ActLLT, ActMLLLT | ○*5 |
| ActQJ71E71UDP, ActMLQJ71E71UDP | ○ | ActQCPUQBus, ActMLQCPUQBus | ○ |
| ActAJ71QE71TCP, ActMLAJ71QE71TCP | × | | |

○: Usable ×: Unusable

- *1: An error is returned if access to the AnUCPU, QCPU (A mode), A173UHCPU(-S1) or A273UH.S3) is made.
- *2: An error is returned if access to the QnACPU is made.
- *3: An error is returned if own board access is made.
- *4: An error is returned if the CPU is other than FX2N and FX2NC.
- *5: An error is returned if the CPU is other than FX0N, FX2, FX2C, FX2N and FX2NC.
- *6: An error is returned if access to the QnACPU is made via the MELSECNET/10 or MELSECNET(II).
- *7: Depending on the communication path (Ethernet communication, MELSECNET/10 communication, etc.), there will be restrictions as placed on the corresponding communication path controls.

(2) Feature

Reads the buffer memory values of the special function module.

(3) Format

Visual Basic® , VBA :IRet = object.ReadBuffer(IStartIO, IAddress, IReadSize, iData(0))

| | | | |
|---------|-----------|---|--------|
| Long | IRet | Returned value | Output |
| Long | IStartIO | First I/O number of module from where values will be 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 *IpsData)

| | | | |
|-------|-----------|---|--------|
| Long | IRet | Returned value | Output |
| Long | IStartIO | First I/O number of module from where values will be read | Input |
| Long | IAddress | Buffer memory address | Input |
| Long | IReadSize | Read size | Input |
| Short | *IpsData | Values read from buffer memory | Output |

VBScript : varRet = object.ReadBuffer(varStartIO, varAddress, varReadSize,
lpvarData)

| | | | |
|---------|-------------|---|--------|
| VARIANT | varRet | Returned value(LONG type) | Output |
| VARIANT | varStartIO | First I/O number of module from where values will be read (LONG type) | Input |
| VARIANT | varAddress | Buffer memory address(LONG type) | Input |
| VARIANT | varReadSize | Read size(LONG type) | Input |
| VARIANT | lpvarData | Values read from buffer memory (SHORT array type) | Output |

(4) Explanation

- As the module I/O number specified in IStartIO(varStartIO), specify a value found by dividing the actual I/O number by 16.
- The buffer values for IReadSize(varReadSize) at the buffer memory address specified in IAddress(varAddress) in the special function module located at the first I/O number specified in IStartIO(varStartIO) are read.
- When using the Act(ML)FXCPU control or Act(ML)LLT control, specify the block number (0 to 7) of the special expansion equipment as the module's first I/O number and any of 0 to 32767 as the buffer memory address.
- Reserve an array of IReadSize (varReadSize) or more for iData (lpIData or lpvarData).

(5) Returned value

Normal termination : 0 is returned.

Abnormal termination : Any value other than 0 is returned.

(Refer to Chapter 6 ERROR CODES.)

POINT

- (1) An error is returned if access to the motion controller CPU is made.
- (2) For iData (lpData or lpvarData), prepare a memory area having the number of points specified in IReadSize(varReadSize).
If there is no memory area, a critical phenomenon such as an application error may occur.
- (3) When buffer memory read (ReadBuffer) is performed for the QCPU (Q mode), read operation may be performed for only the Q series-dedicated module.
Read from the shared memory of the QCPU (Q mode) cannot be performed, either.

4.2.10 WriteBuffer (Buffer memory write)

(1) Applicable ACT controls

The applicable ACT controls are indicated below.

| Control Name | Usability |
|----------------------------------|-----------|
| ActEasyIF, ActMLEasyIF | ○*7 |
| ActQCPUQ, ActMLQCPUQ | ○ |
| ActQCPUA, ActMLQCPUA | ○ |
| ActQnACPU, ActMLQnACPU | ○ |
| ActACPU, ActMLACPU | ○ |
| ActFXCPU, ActMLFXCPU | ○*4 |
| ActQJ71C24, ActMLQJ71C24 | ○ |
| ActAJ71QC24, ActMLAJ71QC24 | ○ |
| ActAJ71UC24, ActMLAJ71UC24 | × |
| ActAJ71C24, ActMLAJ71C24 | × |
| ActQJ71E71TCP, ActMLQJ71E71TCP | ○ |
| ActQJ71E71UDP, ActMLQJ71E71UDP | ○ |
| ActAJ71QE71TCP, ActMLAJ71QE71TCP | × |

| Control Name | Usability |
|----------------------------------|-----------|
| ActAJ71QE71UDP, ActMLAJ71QE71UDP | ○*1, *2 |
| ActAJ71E71TCP, ActMLAJ71E71TCP | ○*1 |
| ActAJ71E71UDP, ActMLAJ71E71UDP | ○*1 |
| ActQCPUQUSB, ActMLQCPUQUSB | ○ |
| ActCCG4QnA, ActMLCCG4QnA | ○ |
| ActCCG4A, ActMLCCG4A | ○ |
| ActMnet10BD, ActMLMnet10BD | ○*3 |
| ActMnetHBD, ActMLMnetHBD | ○*3 |
| ActCCBD, ActMLCCBD | ○*3 |
| ActAnUBD, ActMLAnUBD | ○*6 |
| ActLLT, ActMLLLT | ○*5 |
| ActQCPUQBus, ActMLQCPUQBus | ○ |

○: Usable ×: Unusable

- *1: An error is returned if access to the AnUCPU, QCPU (A mode), A173UHCPU(-S1) or A273UH(-S3) is made.
- *2: An error is returned if access to the QnACPU is made.
- *3: An error is returned if own board access is made.
- *4: An error is returned if the CPU is other than FX2N and FX2NC.
- *5: An error is returned if the CPU is other than FX0N, FX2, FX2C, FX2N and FX2NC.
- *6: An error is returned if access to the QnACPU is made via the MELSECNET/10 or MELSECNET(II).
- *7: Depending on the communication path (Ethernet communication, MELSECNET/10 communication, etc.), there will be restrictions as placed on the corresponding communication path controls.

(2) Feature

Writes values to the buffer memory of the special function module.

(3) Format

Visual Basic® , VBA : IRet = object.WriteBuffer(IStartIO, IAddress, IWriteSize, iData(0))

| | | | |
|---------|------------|--|--------|
| Long | IRet | Returned value | Output |
| Long | IStartIO | First I/O number of module to where values will be written | Input |
| Long | IAddress | Buffer memory address | Input |
| Long | IWriteSize | Write size | Input |
| Integer | iData(n) | Values written to buffer memory | Input |

Visual C++® : IRet = object. WriteBuffer(IStartIO, IAddress, IWriteSize *IpsData)

| | | | |
|-------|------------|--|--------|
| Long | IRet | Returned value | Output |
| Long | IStartIO | First I/O number of module to where values will be written | Input |
| Long | IAddress | Buffer memory address | Input |
| Long | IWriteSize | Write size | Input |
| Short | *IpsData | Values written to buffer memory | Input |

VBScript : varRet = object.WriteBuffer(varStartIO, varAddress, varWriteSize, varData)

| | | | |
|---------|--------------|--|--------|
| VARIANT | varRet | Returned value (LONG type) | Output |
| VARIANT | varStartIO | First I/O number of module to where values will be written (LONG type) | Input |
| VARIANT | varAddress | Buffer memory address (LONG type) | Input |
| VARIANT | varWriteSize | Write size (LONG type) | Input |
| VARIANT | varData | Values written to buffer memory (SHORT array type) | Input |

(4) Explanation

- As the module I/O number specified in IStartIO(varStartIO), specify a value found by dividing the actual I/O number by 16.
- The buffer values for IWriteSize(varWriteSize) at the buffer memory address specified in IAddress(varAddress) in the special function module located at the first I/O number specified in IStartIO(varStartIO) are written.
- When using the Act(ML)FXCPU control or Act(ML)LLT control, specify the block number (0 to 7) of the special expansion equipment as the module's first I/O number and any of 0 to 32767 as the buffer memory address.
- Reserve an array of IWriteSize (varWriteSize) or more for iData (IpsData or varData).

(5) Returned value

Normal termination : 0 is returned.

Abnormal termination : Any value other than 0 is returned.

(Refer to Chapter 6 ERROR CODES.)

POINT

- (1) An error is returned if access to the motion controller CPU is made.
- (2) For iData(IpsData,varData), prepare a memory area having the number of points specified in IWriteSize(varWriteSize).
If there is no memory area, a critical phenomenon such as an application error may occur.
- (3) When buffer memory write (WriteBuffer) is performed for the QCPU (Q mode), write operation may be performed for only the Q series-dedicated module. Write to the shared memory of the QCPU (Q mode) cannot be performed, either.

4.2.11 GetClockData (Clock data read)

(1) Applicable ACT controls

The applicable ACT controls are indicated below.

| Control Name | Usability |
|----------------------------------|-----------|
| ActEasyIF, ActMLEasyIF | ○*4 |
| ActQCPUQ, ActMLQCPUQ | ○ |
| ActQCPUA, ActMLQCPUA | ○ |
| ActQnACPU, ActMLQnACPU | ○ |
| ActACPU, ActMLACPU | ○ |
| ActFXCPU, ActMLFXCPU | ○ |
| ActQJ71C24, ActMLQJ71C24 | ○ |
| ActAJ71QC24, ActMLAJ71QC24 | ○*2 |
| ActAJ71UC24, ActMLAJ71UC24 | ○*2 |
| ActAJ71C24, ActMLAJ71C24 | ○ |
| ActQJ71E71TCP, ActMLQJ71E71TCP | ○ |
| ActQJ71E71UDP, ActMLQJ71E71UDP | ○ |
| ActAJ71QE71TCP, ActMLAJ71QE71TCP | × |

| Control Name | Usability |
|----------------------------------|-----------|
| ActAJ71QE71UDP, ActMLAJ71QE71UDP | ○ |
| ActAJ71E71TCP, ActMLAJ71E71TCP | ○*2 |
| ActAJ71E71UDP, ActMLAJ71E71UDP | ○*2 |
| ActQCPUQUSB, ActMLQCPUQUSB | ○ |
| ActCCG4QnA, ActMLCCG4QnA | ○ |
| ActCCG4A, ActMLCCG4A | ○ |
| ActMnet10BD, ActMLMnet10BD | ○*1 |
| ActMnetHBD, ActMLMnetHBD | ○*1 |
| ActCCBD, ActMLCCBD | ○*1 |
| ActAnUBD, ActMLAnUBD | ○*3 |
| ActLLT, ActMLLLT | × |
| ActQCPUQBus, ActMLQCPUQBus | ○ |

○: Usable ×: Unusable

- *1: An error is returned if own board access is made.
- *2: An error is returned if access to the QnACPU is made.
- *3: An error is returned if access to the QnACPU is made via the MELSECNET/10 or MELSECNET(II).
- *4: Depending on the communication path (Ethernet communication, MELSECNET/10 communication, etc.), there will be restrictions as placed on the corresponding communication path controls.

(2) Feature

Reads time from the clock data of the PLC CPU.

(3) Format

Visual Basic® , VBA : IRet = object.GetClockData(iYear, iMonth, iDay, iDayOfWeek, iHour, iMinute, iSecond)

| | | | |
|---------|------------|------------------------|--------|
| Long | IRet | Returned value | Output |
| Integer | iYear | Read year value | Output |
| Integer | iManth | 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)

| | | | |
|-------|---------------|------------------------|--------|
| 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 |

VBScript : varRet = object.GetClockData(lpvarYear, lpvarMonth, lpvarDay, lpvarDayOfWeek, lpvarHour, *lpvarMinute, lpSecond)

| | | | |
|---------|----------------|------------------------------------|--------|
| VARIANT | varRet | Returned value(LONG type) | Output |
| VARIANT | lpvarYear | Read year value(SHORT type) | Output |
| VARIANT | lpvarManth | Read month value(SHORT type) | Output |
| VARIANT | lpvarDay | Read day value(SHORT type) | Output |
| VARIANT | lpvarDayOfWeek | Read day-of-week value(SHORT type) | Output |
| VARIANT | lpvarHour | Read hour value(SHORT type) | Output |
| VARIANT | lpvarMinute | Read minute value(SHORT type) | Output |
| VARIANT | lpvarSecond | Read second value(SHORT type) | Output |

(4) Explanation

- An error is returned if correct clock data is not set to the PLC CPU.
- As the value stored into iYear (lpsYear or lpvarYear), a four-digit year is returned for the QCPU (Q mode) or a two-digit year for any other CPU. Note that the year for the QCPU (Q mode) is between 1980 and 2079.
- The value stored into iDayOfWeek (lpsDayOfWeek or lpvarDayOfWeek) is as follows.

| Value | Day of Week |
|-------|-------------|
| 0 | Sunday |
| 1 | Monday |
| 2 | Tuesday |
| 3 | Wednesday |
| 4 | Thursday |
| 5 | Friday |
| 6 | Saturday |

(5) Returned value

Normal termination : 0 is returned.

Abnormal termination : Any value other than 0 is returned.

(Refer to Chapter 6 ERROR CODES.)

| POINT |
|--|
| (1) Clock data cannot be read from the A0J2HCPU, A2CCPU and A2CJCPU as they do not have clock data. |
| (2) The QCPU (A mode) and ACPUCPU can get clock data only when the target station is in the STOP status. |
| (3) For the FXCPU, clock data can be read from the FX1N, FX1NC, FX1S, FX2N or FX2NC when it has a built-in clock, or from the FX2 or FX2C when it is fitted with the RTC cassette. An error is returned if the FXCPU is other than the FX1N, FX1NC,FX1S, FX2, FX2C, FX2N and FX2NC. |
| (4) Note that an error of transfer time is produced in clock setting. |

4.2.12 SetClockData (Clock data write)

(1) Applicable ACT controls

The applicable ACT controls are indicated below.

| Control Name | Usability | Control Name | Usability |
|----------------------------------|-----------|----------------------------------|-----------|
| ActEasyIF, ActMLEasyIF | ○*4 | ActAJ71QE71UDP, ActMLAJ71QE71UDP | ○ |
| ActQCPUQ, ActMLQCPUQ | ○ | ActAJ71E71TCP, ActMLAJ71E71TCP | ○*2 |
| ActQCPUA, ActMLQCPUA | ○ | ActAJ71E71UDP, ActMLAJ71E71UDP | ○*2 |
| ActQnACPU, ActMLQnACPU | ○ | ActQCPUQUSB, ActMLQCPUQUSB | ○ |
| ActACPU, ActMLACPU | ○ | ActCCG4QnA, ActMLCCG4QnA | ○ |
| ActFXCPU, ActMLFXCPU | ○ | ActCCG4A, ActMLCCG4A | ○ |
| ActQJ71C24, ActMLQJ71C24 | ○ | ActMnet10BD, ActMLMnet10BD | ○*1 |
| ActAJ71QC24, ActMLAJ71QC24 | ○*2 | ActMnetHBD, ActMLMnetHBD | ○*1 |
| ActAJ71UC24, ActMLAJ71UC24 | ○*2 | ActCCBD, ActMLCCBD | ○*1 |
| ActAJ71C24, ActMLAJ71C24 | ○ | ActAnUBD, ActMLAnUBD | ○*3 |
| ActQJ71E71TCP, ActMLQJ71E71TCP | ○ | ActLLT, ActMLLLT | × |
| ActQJ71E71UDP, ActMLQJ71E71UDP | ○ | ActQCPUQBus, ActMLQCPUQBus | ○ |
| ActAJ71QE71TCP, ActMLAJ71QE71TCP | × | | |

○: Usable ×: Unusable

- *1: An error is returned if own board access is made.
- *2: An error is returned if access to the QnACPU is made.
- *3: An error is returned if access to the QnACPU is made via the MELSECNET/10 or MELSECNET(II).
- *4: Depending on the communication path (Ethernet communication, MELSECNET/10 communication, etc.), there will be restrictions as placed on the corresponding communication path controls.

(2) Feature

Writes time to the clock data of the PLC CPU.

(3) Format

Visual Basic® , VBA : IRet = object.SetClockData(iYear, iMonth, iDay, iDayOfWeek, iHour, iMinute, iSecond)

| | Long | IRet | Returned value | Output |
|---------|------------|------|---------------------------------|--------|
| Integer | iYear | | Year value to be written | Input |
| Integer | iManth | | 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)

| | 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 |

VBScript : varRet = object.SetClockData(varYear, varMonth, varDay, varDayOfWeek, varHour, varMinute, varSecond)

| | | | |
|---------|--------------|--|--------|
| VARIANT | varRet | Returned value(LONG type) | Output |
| VARIANT | varYear | Year value to be written(SHORT type) | Input |
| VARIANT | varManth | Month value to be written(SHORT type) | Input |
| VARIANT | varDay | Day value to be written(SHORT type) | Input |
| VARIANT | varDayOfWeek | Day-of-week value to be written (SHORT type) | Input |
| VARIANT | varHour | Hour value to be written(SHORT type) | Input |
| VARIANT | varMinute | Minute value to be written(SHORT type) | Input |
| VARIANT | varSecond | Second value to be written(SHORT type) | Input |

(4) Explanation

- An error is returned if the clock data to be set are not correct values.
- As to the value specified in iYear (sYear or varYear), a four-digit year is valid for the QCPU (Q mode) or a two-digit year for any other CPU.
Note that the year valid for the QCPU (Q mode) is between 1980 and 2079. An error will occur if a four-digit year is set to any CPU other than the QCPU (Q mode).
- The value to be specified in iDayOfWeek (sDayOfWeek or varDayOfWeek) is as follows.

| Value | Day of Week |
|-------|-------------|
| 0 | Sunday |
| 1 | Monday |
| 2 | Tuesday |
| 3 | Wednesday |
| 4 | Thursday |
| 5 | Friday |
| 6 | Saturday |

(5) Returned value

Normal termination : 0 is returned.

Abnormal termination : Any value other than 0 is returned.

(Refer to Chapter 6 ERROR CODES.)

| POINT |
|--|
| (1) Clock data cannot be read from the A0J2HCPU, A2CCPU and A2CJCPU as they do not have clock data. |
| (2) For the QCPU (A mode) and ACPU, clock data can be set only when the target station is in the STOP status. |
| (3) For the QCPU (A mode) and ACPU, the clock setting special relay "M9028" changes to OFF after clock data setting. |
| (4) For the FXCPU, clock setting can be made to the FX1N, FX1NC, FX1S, FX2N or FX2NC when it has a built-in clock, or to the FX2 or FX2C when it is fitted with the RTC cassette. An error is returned if the FXCPU is other than the FX1N, FX1NC, FX1S, FX2, FX2C, FX2N and FX2NC. |
| (5) Note that an error of transfer time is produced in clock setting. |

4.2.13 GetCpuType (PLC CPU type read)

(1) Applicable ACT controls

This function is available for all ACT controls*1.

*1: MELSECNET/10 board will result in an error if own board access is made.

(2) Feature

Returns the type character string and type code of the PLC CPU.

(3) Format

Visual Basic® , VBA : IRet = object.GetCpuType(szCpuName, ICpuType)

| | | | |
|--------|-----------|-------------------------------|--------|
| Long | IRet | Returned value | Output |
| String | szCpuName | PLC CPU type character string | Output |
| Long | ICpuType | PLC CPU type code | Output |

Visual C++® : IRet = object.GetCpuType(*szCpuType, *IpICpuType)

| | | | |
|------|-------------|-------------------------------|--------|
| Long | IRet | Returned value | Output |
| BSTR | *szCpuName | PLC CPU type character string | Output |
| Long | *IpICpuType | PLC CPU type code | Output |

VBScript : varRet = object.GetCpuType(varCpuName, IpvarCpuCode)

| | | | |
|---------|--------------|--|--------|
| VARIANT | varRet | Returned value(LONG type) | Output |
| VARIANT | IpvarCpuName | PLC CPU type character string (character string type) | Output |
| VARIANT | IpvarCpuCode | PLC CPU type code(LONG type) | Output |

(4) Explanation

- The type of the PLC which is making communication is stored into szCpuName (IpvarCpuName) and its type code into ICpuType (IpICpuType or IpvarCpuCode).
- The PLC CPU type character string is returned in UNICODE.

(5) CPU type character string and type code

The following table lists the CPU type character strings and type codes read using GetCpuType.

(a) Type character string list

| CPU/Network Board Type | Type Character String | | CPU/Network Board Type | Type Character String | |
|------------------------|------------------------|--------------------------------|------------------------|-------------------------------------|-------------------------------------|
| | CPU/Network Board Type | When GX Simulator is connected | | CPU/Network Board Type | When GX Simulator is connected |
| Q00JCPU | Q00JCPU | Q00JCPU | A2ACPUP21/R21-S1 | A2AS1 | A2AS1 |
| Q00CPU | Q00CPU | Q00CPU | A2UCPU | A2U | A2U |
| Q01CPU | Q01CPU | Q01CPU | A2UCPU-S1 | A2US1 | A2U |
| Q02CPU | Q02CPU | Q02CPU | A2USCPU | A2U | A2U |
| Q02HCPU | Q02HCPU | Q02CPU | A2USCPU-S1 | A2US1 | A2U |
| Q06HCPU | Q06HCPU | Q06HCPU | A2ASCPU | A2U | A2U |
| Q12HCPU | Q12HCPU | Q12HCPU | A2ASCPU-S1 | A2US1 | A2U |
| Q25HCPU | Q25HCPU | Q25HCPU | A2ASCPU-S30 | A3U | A3U |
| Q02CPU-A | Q02CPU | Q02CPU-A | A2USHCPU-S1 | A2USH | A2USH |
| Q02HCPU-A | Q02HCPU | Q02CPU-A | A3NCPUCPU | A3N | A3N |
| Q06HCPU-A | Q06HCPU | Q06HCPU-A | A3ACPU | A3A | A3A |
| Q2ACPU | Q2ACPU | Q2ACPU | A3ACPUP21/R21 | A3A | A3A |
| Q2ACPU-S1 | Q2ACPU-S1 | Q2ACPU-S1 | A3UCPU | A3U | A3U |
| Q2ASCPU | Q2ACPU | Q2ACPU | A4UCPU | A4U | A4U |
| Q2ASCPU-S1 | Q2ACPU-S1 | Q2ACPU-S1 | A1FXCPU | A1FX | A1FX |
| Q2ASHCPU | Q2ACPU | Q2ACPU | FX ₀ | FX ₀ /FX _{0S} | FX ₀ /FX _{0S} |
| Q2ASHCPU-S1 | Q2ACPU-S1 | Q2ACPU-S1 | FX _{0S} | FX ₀ /FX _{0S} | FX ₀ /FX _{0S} |
| Q3ACPU | Q3ACPU | Q3ACPU | FX _{0N} | FX _{0N} | FX _{0N} |
| Q4ACPU | Q4ACPU | Q4ACPU | FX ₁ | FX ₁ | FX ₁ |
| Q4ARCPU | Q4ACPU | Q4ACPU | FX _{1S} | FX _{1S} | FX _{1S} |
| A0J2HCPU | A0J2H | A0J2H | FX _{1N} | FX _{1N} | FX _{1N} |
| A1SCPU | A1S | A1S | FX _{1NC} | FX _{1N} | FX _{1N} |
| A1SCPU-S1 | A1S | A1S | FX ₂ | FX ₂ /FX _{2C} | FX ₂ /FX _{2C} |
| A1SCPUC24-R2 | A1S | A1S | FX _{2C} | FX ₂ /FX _{2C} | FX ₂ /FX _{2C} |
| A1SHCPU | A1SH | A1SH | FX _{2N} | FX _{2N} /FX _{2NC} | FX _{2N} /FX _{2NC} |
| A1SJCPU | A1S | A1S | FX _{2NC} | FX _{2N} /FX _{2NC} | FX _{2N} /FX _{2NC} |
| A1SJHCPU | A1SH | A1SH | A171SHCPU | A171SH | A171SH |
| A1NCPUCPU | A1N | A1N | A172SHCPU | A172SH | A172SH |
| A2CCPU | A2C | A2C | A173UHCPU | A173UHCPU | A173UH |
| A2CCPUC24 | A2C | A2C | A173UHCPU-S1 | A173UHCPU-S1 | A173UH |
| A2CCPUC24-PRF | A2C | A2C | A273UHCPU | A273UH | A273UH |
| A2CJCPU | A2C | A2C | A273UHCPU-S3 | A273UH | A273UH |
| A2NCPUCPU | A2N | A2N | A70BDE-J71QLP23(GE) | A70BDE-J71QLP23 | — |
| A2NCPUCPU-S1 | A2N | A2N | A70BDE-J71QBR13 | A70BDE-J71QBR13 | — |
| A2SCPU | A2S | A2N | A70BDE-J71QLR23 | A70BDE-J71QLR23 | — |
| A2SCPU-S1 | A2S | A2N | A80BDE-J61BT11 | A80BDE-J61BT11 | — |
| A2SHCPU | A2SH | A2SH | A80BDE-J61BT13 | A80BDE-J61BT13 | — |
| A2SHCPU-S1 | A2SH | A2SH | A80BDE-A2USH-S1 | A2USH-S1 | — |
| A2ACPU | A2A | A2AS1 | Q80BD-J71LP21-25 | Q80BD-J71LP21-25 | — |
| A2ACPU-S1 | A2AS1 | A2AS1 | Q80BD-J71LP21G | Q80BD-J71LP21G | — |
| A2ACPUP21/R21 | A2AS1 | A2AS1 | Q80BD-J71BR11 | Q80BD-J71BR11 | — |

(b) Type code list

| CPU/Network Board Type | Type Code | | CPU/Network Board Type | Type Code | |
|------------------------|---------------------------------|--------------------------------|------------------------|---------------------------------|--------------------------------|
| | When CPU/own board is connected | When GX Simulator is connected | | When CPU/own board is connected | When GX Simulator is connected |
| Q00JCPU | 250H | 250H | A2ACPUP21/R21-S1 | 93H | 93H |
| Q00CPU | 251H | 251H | A2UCPU | 82H | 82H |
| Q01CPU | 252H | 252H | A2UCPU-S1 | 83H | 83H |
| Q02CPU | 41H | 41H | A2USCPU | 82H | 82H |
| Q02HCPU | 41H | 41H | A2USCPU-S1 | 83H | 82H |
| Q06HCPU | 42H | 42H | A2ASCPU | 82H | 82H |
| Q12HCPU | 43H | 43H | A2ASCPU-S1 | 82H | 82H |
| Q25HCPU | 44H | 44H | A2ASCPU-S30 | 94H | 84H |
| Q02CPU-A | 141H | 141H | A2USHCPU-S1 | 84H | 84H |
| Q02HCPU-A | 141H | 141H | A3NCPU | A3H | A3H |
| Q06HCPU-A | 142H | 142H | A3ACPU | 94H | 94H |
| Q2ACPU | 21H | 21H | A3ACPUP21/R21 | 94H | 94H |
| Q2ACPU-S1 | 22H | 22H | A3UCPU | 84H | 84H |
| Q2ASCPU | 21H | 21H | A4UCPU | 85H | 85H |
| Q2ASCPU-S1 | 22H | 22H | A1FXCPU | A2H | A2H |
| Q2ASHCPU | 21H | 21H | FX ₀ | F0H | F0H |
| Q2ASHCPU-S1 | 22H | 22H | FX _{0S} | F0H | F0H |
| Q3ACPU | 23H | 23H | FX _{0N} | 8EH | 8EH |
| Q4ACPU | 24H | 24H | FX ₁ | F1H | F1H |
| Q4ARCPU | 24H | 24H | FX _{1S} | F2H | F2H |
| A0J2HCPU | 98H | 98H | FX _{1N} | 9EH | 9EH |
| A1SCPU | 98H | 98H | FX _{1NC} | 9EH | 9EH |
| A1SCPU-S1 | 98H | 98H | FX ₂ | 8DH | 8DH |
| A1SCPUC24-R2 | 98H | 98H | FX _{2C} | 8DH | 8DH |
| A1SHCPU | A3H | A3H | FX _{2N} | 9DH | 9DH |
| A1SJCPU | 98H | 98H | FX _{2NC} | 9DH | 9DH |
| A1SJHCPU | A3H | A3H | A171SHCPU | A3H | A3H |
| A1NCPU | A1H | A1H | A172SHCPU | A3H | A3H |
| A2CCPU | 9AH | 9AH | A173UHCPU | 84H | 84H |
| A2CCPUC24 | 9AH | 9AH | A173UHCPU-S1 | 84H | 84H |
| A2CCPUC24-PRF | 9AH | 9AH | A273UHCPU | 84H | 84H |
| A2CJCPU | 9AH | 9AH | A273UHCPU-S3 | 84H | 84H |
| A2NCPU | A2H | A2H | A70BDE-J71QLP23(GE) | 90H | — |
| A2NCPU-S1 | A2H | A2H | A70BDE-J71QBR13 | 90H | — |
| A2SCPU | A2H | A2H | A70BDE-J71QLR23 | 90H | — |
| A2SCPU-S1 | A2H | A2H | A80BDE-J61BT11 | 90H | — |
| A2SHCPU | A3H | A3H | A80BDE-J61BT13 | 90H | — |
| A2SHCPU-S1 | A3H | A3H | A80BDE-A2USH-S1 | 84H | — |
| A2ACPU | 92H | 92H | Q80BD-J71LP21-25 | 90H | — |
| A2ACPU-S1 | 93H | 93H | Q80BD-J71LP21G | 90H | — |
| A2ACPUP21/R21 | 92H | 93H | Q80BD-J71BR11 | 90H | — |

- 1) When using the TCP/IP of the E71 or QE71, refer to the manual of the corresponding module.
- 2) When access to the AnUCPU, QnACPU, QCPU (A mode) or A273UHCPU(-S3) is made from the C24 or E71, the type code equivalent to that of the AnACPU is returned. (92H, 93H, 94H)

- 3) When access to the AnUCPU, QnACPU, QCPU (A mode) or A273UHCPU(-S3) is made from the C24, E71 or UC24 via the network, the type code equivalent to that of the AnACPU is returned. (92H, 93H, 94H)
 - 4) When access to the AnUCPU, QCPU (A mode) or A273UHCPU(-S3) is made from the AnNCPU or AnACPU via the network by CPU COM communication, the type code equivalent to that of the AnACPU is returned. (92H, 93H, 94H)
 - 5) When access to the QnACPU or QCPU (A mode) is made from the CPU board, the type code equivalent to that of the AnACPU (92H, 93H, 94H) is returned for the QnACPU or the type code equivalent to that of the A4UCPU (85H) is returned for the QCPU (A mode).
 - 6) When access to the QCPU (A mode) is made from the UC24, the type code equivalent to that of the A4UCPU (85H) is returned.
 - 7) When access to the QCPU (A mode) is made from the CC-Link G4 module, the type code equivalent to that of the A4UCPU (85H) is returned.
- (5) Returned value
- Normal termination : 0 is returned.
 - Normal termination : Abnormal termination: A value other than 0 is returned.
(Refer to Chapter 6 ERROR CODES.)

4.2.14 SetCpuStatus (Remote control)

(1) Applicable ACT controls

The applicable ACT controls are indicated below.

| Control Name | Usability | Control Name | Usability |
|----------------------------------|-----------|----------------------------------|-----------|
| ActEasyIF, ActMLEasyIF | ○*5 | ActAJ71QE71UDP, ActMLAJ71QE71UDP | ○ |
| ActQCPUQ, ActMLQCPUQ | ○ | ActAJ71E71TCP, ActMLAJ71E71TCP | ○*1 |
| ActQCPUA, ActMLQCPUA | ○ | ActAJ71E71UDP, ActMLAJ71E71UDP | ○ |
| ActQnACPU, ActMLQnACPU | ○ | ActQCPUQUSB, ActMLQCPUQUSB | ○ |
| ActACPU, ActMLACPU | ○ | ActCCG4QnA, ActMLCCG4QnA | ○ |
| ActFXCPU, ActMLFXCPU | ○ | ActCCG4A, ActMLCCG4A | ○ |
| ActQJ71C24, ActMLQJ71C24 | ○ | ActMnet10BD, ActMLMnet10BD | ○*2 |
| ActAJ71QC24, ActMLAJ71QC24 | ○ | ActMnetHBD, ActMLMnetHBD | ○*2 |
| ActAJ71UC24, ActMLAJ71UC24 | ○*4 | ActCCBD, ActMLCCBD | ○*2 |
| ActAJ71C24, ActMLAJ71C24 | ○*4 | ActAnUBD, ActMLAnUBD | ○*3 |
| ActQJ71E71TCP, ActMLQJ71E71TCP | ○ | ActLLT, ActMLLLT | ○ |
| ActQJ71E71UDP, ActMLQJ71E71UDP | ○ | ActQCPUQBus, ActMLQCPUQBus | ○ |
| ActAJ71QE71TCP, ActMLAJ71QE71TCP | ○*1 | | |

○: Usable ×: Unusable

- *1: An error is returned when remote operation is performed for the own station.
- *2: An error is returned when own board access is made.
- *3: When access to the QnACPU is made via the MELSECNET/10 or MELSECNET(II), making PAUSE specification for the QnACPU results in an error.
- *4: An error is returned if PAUSE specification is made.
- *5: Depending on the communication path (Ethernet communication, MELSECNET/10 communication, etc.), there will be restrictions as placed on the corresponding communication path controls.

(2) Feature

Performs remote operation of the PLC CPU.

(3) Format

Visual Basic® , VBA : IRet = object.SetCpuStatus(IOperation)

| | | | |
|------|------------|-----------------------|--------|
| Long | IRet | Returned value | Output |
| Long | IOperation | Remote run/stop/pause | Input |

Visual C++® : IRet = object.SetCpuStatus(IOperation)

| | | | |
|------|------------|-----------------------|--------|
| Long | IRet | Returned value | Output |
| Long | IOperation | Remote run/stop/pause | Input |

VBScript : varRet = object.SetCpuStatus(varOperation)

| | | | |
|---------|--------------|----------------------------------|--------|
| VARIANT | varRet | Returned value(LONG type) | Output |
| VARIANT | varOperation | Remote run/stop/pause(LONG type) | Input |

(4) Explanation

- The operation specified in IOperation (varOperation) is performed. Specifying any value other than the following will result in an error.

| Value | Operation |
|-------|--------------|
| 0 | Remote run |
| 1 | Remote stop |
| 2 | Remote pause |

(5) Returned value

Normal termination : 0 is returned.

Abnormal termination : A value other than 0 is returned.

(Refer to Chapter 6 ERROR CODES.)

| |
|-------|
| POINT |
|-------|

| |
|---|
| Since the FXCPU does not have the PAUSE switch as the PLC CPU, an error is returned if remote pause is specified in SetCpuStatus. |
|---|

4.2.15 EntryDeviceStatus (Device status monitor registration)

(1) Applicable ACT controls

This function is available for all ACT controls.

(2) Feature

Registers devices whose statuses will be monitored.

(3) Format

Visual Basic[®], VBA : IRet = object.EntryDeviceStatus(szDeviceList, ISize, IMonitorCycle, IData(0))

| Long | IRet | Returned value | Output |
|------|---------------|------------------------------------|--------|
| BSTR | szDeviceList | Registered device name list | Input |
| Long | ISize | Number of registered device points | Input |
| Long | IMonitorCycle | Status monitor time interval | Input |
| Long | IData(n) | Registered device value list | Input |

Visual C++[®] : IRet = object.EntryDeviceStatus(szDeviceList, ISize, IMonitorCycle, *lpIData)

| Long | IRet | Returned value | Output |
|------|---------------|------------------------------------|--------|
| BSTR | szDeviceList | Registered device name list | Input |
| Long | ISize | Number of registered device points | Input |
| Long | IMonitorCycle | Status monitor time interval | Input |
| Long | *lpIData | Registered device value list | Input |

VBScript: varRet = object.EntryDeviceStatus(varDeviceList, varSize, varMonitorCycle, varData)

| VARIANT | varRet | Returned value(LONG type) | Output |
|---------|-----------------|--|--------|
| VARIANT | varDeviceList | Registered device name list (BSTR type) | Input |
| VARIANT | varSize | Number of registered device points (LONG type) | Input |
| VARIANT | varMonitorCycle | Status monitor time interval (LONG type) | Input |
| VARIANT | varData | Registered device value list (LONG type) | Input |

(4) Explanation

- Check whether ISize (varSize) of the device group specified in szDeviceList (varDeviceList) is in the status specified in IData (lpIData or varData). Specify the check time in IMonitorCycle (varMonitorCycle).

When the status is established, the OnDeviceStatus function of the user application is executed.

- Using the line feed symbol, separate the devices in the character string specified in the device list.

The last device need not be followed by the line feed symbol.

(Example)

Visual Basic[®], VBA, VBScript : "D0" & vbLf & "D1" & vbLf & "D2"

Visual C++[®] : D0\nD1\nD2

- The maximum number of device points that may be specified in ISize (varSize) is 20 points.

- In IMonitorCycle (varMonitorCycle), specify a value within the range 1 second to 1 hour (set between 1 to 3600 in seconds).
Specifying any other value outside the above range will result in an error.
- Store the registered device value list in IData (IplData or varData).

(5) Device specifying methods

Specify the devices in the following methods.

(Example 1) When devices are specified as follows (3 points)

When using Visual Basic[®], VBA, VBScript : M0 & vbLf & D0 & vbLf & K8M0

When using Visual C++[®] : M0\nD0\nK8M0

| 2 Upper Bytes | 2 Lower Bytes |
|---------------|---------------|
| *1 | M0 |
| *1 | D0 |
| M16 to M31 *2 | M0 to M15 *2 |

(Example 2) When devices including CN200 and later of FXCPU are specified
(3 points in all) *3

When using Visual Basic[®], VBA, VBScript : D0 & vbLf & CN200 & vbLf & D1

When using Visual C++[®] : D0\nCN200\nD1

| 2 Upper Bytes | 2 Lower Bytes |
|---------------|---------------|
| *1 | D0 |
| H of CN200 | L of CN200 |
| *1 | D1 |

(Example 3) When devices including FD are specified (3 points in all)

When using Visual Basic[®], VBA, VBScript : D0 & vbLf & FD0 & vbLf & D1

When using Visual C++[®] : D0\nFD0\nD1

| 2 Upper Bytes | 2 Lower Bytes |
|---------------|---------------|
| *1 | D0 |
| *1 | LL of FD |
| *1 | D1 |

*1: Not used. (0 is stored.)

*2: Lower bits are stored in device number order.

*3: For CN200 or later of FXCPU, 2 words are read from 1 point by random read.

(6) Returned value

Normal termination : 0 is returned.

Abnormal termination : Any value other than 0 is returned.

(Refer to Chapter 6 ERROR CODES.)

POINT

(1) Device status monitoring may not be performed at the specified status monitor time intervals depending on such conditions as the personal computer performance, currently executed application load, and time required for communication with the PLC.

Simultaneous use of any other ACT control function would also be the cause of disabling device status monitoring at the specified status monitor time intervals.

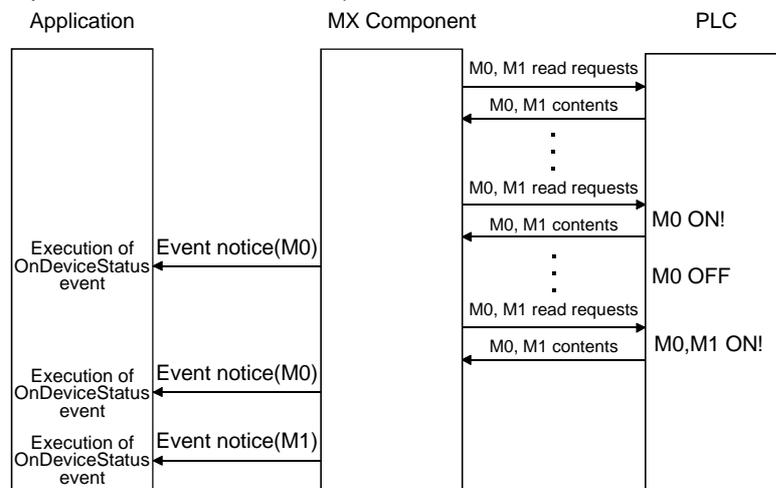
(2) For IData (IpIData or IpvarData), prepare a memory area having the number of points specified in ISize (varSize).

If no memory area is available, a serious phenomenon such as an application error may occur.

(3) Execution of EntryDeviceStatus during status monitoring will result in an error. When changing any status monitor condition, execute FreeDeviceStatus and then execute EntryDeviceStatus.

(4) If the statuses of multiple devices change at the same time, the OnDeviceStatus event is executed every time the status changes.

(Example: When M0 is monitored)



(5) With this function, the ACT control performs device random read periodically and confirms that the status has been established.

Therefore, this function is not designed for the PLC CPU to notify MX Component that the device status has been established.

Hence, depending on the specified status monitor time interval, the ACT control may not be able to confirm that the PLC CPU device status has been established.

4.2.16 FreeDeviceStatus (Device status monitor deregistration)

(1) Applicable ACT controls

This function is available for all ACT controls.

(2) Feature

Deregisters the devices that have been registered in EntryDeviceStatus to monitor statuses.

(3) Format

Visual Basic[®] , Visual C++[®] , VBA : IRet = object.FreeDeviceStatus()

| | | | |
|------|------|----------------|--------|
| Long | IRet | Returned value | Output |
|------|------|----------------|--------|

VBScript : varRet = object.FreeDeviceStatus()

| | | | |
|---------|--------|----------------------------|--------|
| VARIANT | varRet | Returned value (LONG type) | Output |
|---------|--------|----------------------------|--------|

(4) Explanation

The devices that have been set by the EntryDeviceStatus function to monitor statuses are deregistered.

(5) Returned value

Normal termination : 0 is returned.

Abnormal termination : Any value other than 0 is returned.

(Refer to Chapter 6 ERROR CODES.)

4.2.17 OnDeviceStatus (Announces event)

(1) Applicable ACT controls

This function is available for all ACT controls.

(2) Feature

Gives event notice when the device condition registered with the EntryDeviceStatus function holds.

(3) Format

Visual Basic®, Visual C++®, VBA : IRet = object.OnDeviceStatus(szDevice, IData, IReturnCode)

| | | | |
|------|-------------|--|-------|
| BSTR | szDevice | Name of device whose condition has held | Input |
| Long | IData | Value of device whose condition has held | Input |
| Long | IReturnCode | Returned value of condition check processing | Input |

VBScript : varRet = object.OnDeviceStatus(varDevice, varData, varReturnCode)

| | | | |
|---------|---------------|---|-------|
| VARIANT | varDevice | Name of device whose condition has held(BSTR type) | Input |
| VARIANT | varData | Value of device whose condition has held(LONG type) | Input |
| VARIANT | varReturnCode | Returned value of condition check processing(LONG type) | Input |

(4) Explanation

The application is notified of an event when the device condition registered with the EntryDeviceStatus function holds.

Placing this function in the user application allows the application to receive the event when the registered device condition holds.

(5) Returned value

None

| | |
|--------------|--|
| POINT | <p>Displaying the message box or modal dialog, for example, within the OnDeviceStatus function using the user application will cause the user application to wait for data entry.</p> <p>In this status, note that control will not return to the ATC control and device control processing will stop until the message box or modal dialog is closed.</p> |
|--------------|--|

4.3 Details of the Functions (Custom Interface)

This section explains the details of the functions.

The details of the functions in this section assume that the custom interface is used.

The custom interface may be used on only Visual C++®.

For the dispatch interface, refer to "Section 4.2 Details of the Functions (Dispatch Interface)".

This section describes only the formats of the functions.

For details of other than the formats, refer to "Section 4.2 Details of the Functions (Dispatch Interface)".

4.3.1 Open (Communication line opening)

```
hResult = object.Open(*lpRetCode )
```

| | | | |
|---------|------------|--|--------|
| HRESULT | hResult | Returned value of COM | Output |
| LONG | *lpRetCode | Returned value of communication function | Output |

4.3.2 Close (Communication line closing)

```
hResult = object.Close(*lpRetCode )
```

| | | | |
|---------|------------|--|--------|
| HRESULT | hResult | Returned value of COM | Output |
| LONG | *lpRetCode | Returned value of communication function | Output |

4.3.3 ReadDeviceBlock (Device batch-read)

```
hResult = object.ReadDeviceBlock( szDevice, lSize, *lpData, *lpRetCode )
```

| | | | |
|---------|------------|--|--------|
| HRESULT | hResult | Returned value of COM | Output |
| BSTR | szDevice | Device name | Input |
| LONG | lSize | Number of read points | Input |
| LONG | *lpData | Read device values | Output |
| LONG | *lpRetCode | Returned value of communication function | Output |

4.3.4 WriteDeviceBlock (Device batch-write)

```
hResult = object.WriteDeviceBlock( szDevice, lSize, *lpData, *lpRetCode )
```

| | | | |
|---------|------------|--|--------|
| HRESULT | hResult | Returned value of COM | Output |
| BSTR | szDevice | Device name | Input |
| LONG | lSize | Number of write points | Input |
| LONG | *lpData | Written device values | Input |
| LONG | *lpRetCode | Returned value of communication function | Output |

4.3.5 ReadDeviceRandom (Device random-read)

```
hResult = object.ReadDeviceBlock( szDevice, ISize, *lpData, *lpRetCode )
```

| | | | |
|---------|------------|--|--------|
| HRESULT | hResult | Returned value of COM | Output |
| BSTR | szDevice | Device name | Input |
| LONG | ISize | Number of read points | Input |
| LONG | *lpData | Read device values | Output |
| LONG | *lpRetCode | Returned value of communication function | Output |

4.3.6 WriteDeviceRandom (Device random-write)

```
hResult = object.WriteDeviceRandom( szDeviceList, ISize, *lpData,
                                     *lpRetCode )
```

| | | | |
|---------|------------|--|--------|
| HRESULT | hResult | Returned value of COM | Output |
| BSTR | szDevice | Device name | Input |
| LONG | ISize | Number of write points | Input |
| LONG | *lpData | Written device values | Input |
| LONG | *lpRetCode | Returned value of communication function | Output |

4.3.7 SetDevice (Device data setting)

```
hResult = object.SetDevice( szDeviceList, *lpData, *lpRetCode )
```

| | | | |
|---------|--------------|--|--------|
| HRESULT | hResult | Returned value of COM | Output |
| BSTR | szDeviceList | Device name | Input |
| LONG | *lpData | Set data | Input |
| LONG | *lpRetCode | Returned value of communication function | Output |

4.3.8 GetDevice (Device data acquisition)

```
hResult = object.GetDevice( szDeviceList, *lpData, *lpRetCode )
```

| | | | |
|---------|--------------|--|--------|
| HRESULT | hResult | Returned value of COM | Output |
| BSTR | szDeviceList | Device name | Input |
| LONG | *lpData | Set data | Output |
| LONG | *lpRetCode | Returned value of communication function | Output |

4.3.9 ReadBuffer (Buffer memory read)

```
hResult = object.ReadBuffer( IStartIO, IAddress, IReadSize,
                             *lpsData, *lplRetCode )
```

| | | | |
|---------|-------------|---|--------|
| HRESULT | hResult | Returned value of COM | Output |
| LONG | IStartIO | First I/O number of module from where values will be read | Input |
| LONG | IAddress | Buffer memory address | Input |
| LONG | IReadSize | Read size | Input |
| SHORT | *lpsData | Values read from buffer memory | Output |
| LONG | *lplRetCode | Returned value of communication function | Output |

4.3.10 WriteBuffer (Buffer memory write)

```
hResult = object.WriteBuffer( IStartIO, IAddress, IWriteSize,
                               *lpsData, *lplRetCode )
```

| | | | |
|---------|-------------|--|--------|
| HRESULT | hResult | Returned value of COM | Output |
| LONG | IStartIO | First I/O number of module to where values will be written | Input |
| LONG | IAddress | Buffer memory address | Input |
| LONG | IWriteSize | Write size | Input |
| SHORT | *lpsData | Values written to buffer memory | Input |
| LONG | *lplRetCode | Returned value of communication function | Output |

4.3.11 GetClockDSata (Clock data read)

```
hResult = object.GetClockData(*lpsYear, *lpsMonth, *lpsDay,
                               *lpsDayOfWeek, *lpsHour, *lpsMinute, *lpsSecond, *lplRetCode )
```

| | | | |
|---------|---------------|--|--------|
| HRESULT | hResult | Returned value of COM | Output |
| SHORT | *lpsYear | Read year value | Output |
| SHORT | *lpsMonth | Read month value | Output |
| SHORT | *lpsDay | Read day value | Output |
| SHORT | *lpsDayOfWeek | Read day-of-week value | Output |
| SHORT | *lpsHour | Read hour value | Output |
| SHORT | *lpsMinute | Read minute value | Output |
| SHORT | *lpsSecond | Read second value | Output |
| LONG | *lplRetCode | Returned value of communication function | Output |

4.3.12 SetClockData (Clock data write)

```
hResult = object.SetClockData( sYear, sMonth, sDay, sDayOfWeek,
                               sHour, sMinute, sSecond, *lplRetCode )
```

| | | | |
|---------|-------------|--|--------|
| HRESULT | hResult | Returned value of COM | Output |
| SHORT | sYear | Year value to be written | Input |
| SHORT | sMonth | Month value to be written | Input |
| SHORT | sDay | Day value to be written | Input |
| SHORT | sDayOfWeek | Day-of-week value to be written | Input |
| SHORT | sHour | Hour value to be written | Input |
| SHORT | sMinute | Minute value to be written | Input |
| SHORT | sSecond | Second value to be written | Input |
| LONG | *lplRetCode | Returned value of communication function | Output |

4.3.13 GetCpuType (PLC CPU type read)

```
hResult = object.GetDevice( *szDeviceList, *lplData, *lplRetCode )
```

| | | | |
|---------|-------------|--|--------|
| HRESULT | hResult | Returned value of COM | Output |
| BSTR | *szCpuName | PLC CPU type character string | Output |
| LONG | *lplCpuType | PLC CPU type code | Output |
| LONG | *lplRetCode | Returned value of communication function | Output |

4.3.14 SetCpuStatus (Remote control)

```
hResult = object.SetCpuStatus( IOperation, *lplRetCode )
```

| | | | |
|---------|-------------|--|--------|
| HRESULT | hResult | Returned value of COM | Output |
| LONG | IOperation | Remote run/stop/pause | Input |
| LONG | *lplRetCode | Returned value of communication function | Output |

4.3.15 EntryDeviceStatus (Device status monitor registration)

hResult = object.EntryDeviceStatus(szDeviceList, ISize, IMonitorCycle, *lpIData, *lpIRetCode)

| | | | |
|---------|---------------|--|--------|
| HRESULT | hResult | Returned value of COM | Output |
| BSTR | szDeviceList | Registered device name list | Input |
| LONG | ISize | Number of registered device points | Input |
| LONG | IMonitorCycle | Status monitor time interval | Input |
| LONG | *lpIData | Registered device value list | Input |
| LONG | *lpIRetCode | Returned value of communication function | Output |

4.3.16 FreeDeviceStatus (Device status monitor deregistration)

hResult = object.FreeDeviceStatus(*lpIRetCode)

| | | | |
|---------|-------------|--|--------|
| HRESULT | hResult | Returned value of COM | Output |
| LONG | *lpIRetCode | Returned value of communication function | Output |

4.3.17 OnDeviceStatus (Announces event)

hResult = object.OnDeviceStatus(szDevice, IData, IReturnCode, *lpIRetCode)

| | | | |
|------|-------------|--|--------|
| BSTR | szDevice | Name of device whose condition has held | Input |
| LONG | IData | Value of device whose condition has held | Input |
| LONG | IReturnCode | Returned value of condition check processing | Input |
| LONG | *lpIRetCode | Returned value of communication function | Output |

5 SAMPLE PROGRAMS

The sample programs given in this section are those created using Microsoft® Visual Basic® 6.0, Microsoft® Visual C++® 6.0, VBA and VBScript.

The sample programs are attached to create user programs.

Please use them on your responsibility.

5.1 Visual Basic® Sample Program

This sample program is designed to read PLC CPU data and read/write device values using the ActEasyIF control and ActACPU control.

This sample program was created on Visual Basic® 6.0.

(1) Using method

Load the form and choose the control to be used.

Click the button to open the communication line.

When you selected the ActEasyIF control, type the logical number, which was specified in communication settings on the communication setting utility, in the "LogicalStationNumber" text box before clicking the button.

Click the button to close the communication line.

When you click the button, the type and type code of the PLC CPU to which the line is currently connected appear in the "Data" list box.

Typing the device names from where you want to read data and the number of points in the "DeviceName" and "DeviceSize" text boxes, respectively, and clicking the button displays the device data in the "Data" list box.

Typing the device names to where you want to write data and the number of points in the "DeviceName" and "DeviceSize" text boxes, respectively, and the device values to be written in the "DeviceData" text box and clicking the button writes the device values to the PLC CPU.

If an error occurs during execution, the error code appears in the "ReturnCode" text box.

If an error has occurred, refer to "6 ERROR CODES" and remove the error cause.

(2) Precautions for use of the sample program

(a) When using the ActEasyIF control, make communication settings on the communication setting utility before running this sample program.

(b) When the ActACPU control is used, this sample program is designed to use the PLC CPU as "A1N" and the COM port as "COM1" exclusively.

(c) When changing the control to be used, click the button to close the communication line once, then change the control, and click the button to reopen the communication line.

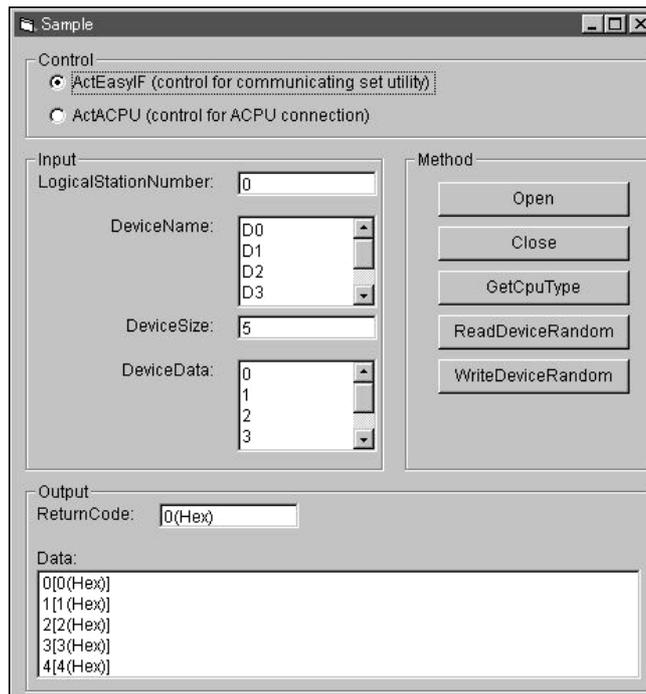
(3) Sample file list

By default, the sample program is installed in the following folders.

- C:\MELSEC\Act\Sample\Vb\Sample.exe Execution file
- C:\MELSEC\Act\Sample\Vb\Sample.vbp Project file
- C:\MELSEC\Act\Sample\Vb\SampleForm.frm Visual Basic Form file
- C:\MELSEC\Act\Sample\Vb\SampleForm.frx Visual Basic Form Binary file
- C:\MELSEC\Act\Sample\Vb\ActDefine.bas Header file

(4) Screen

The sample program screen will be explained.



| Item | Description |
|----------------------|---|
| Control | Choose the control to be used. |
| LogicalStationNumber | Enter the logical station number that was specified in communication settings on the communication setting utility. |
| DeviceName | Enter the device names from/to where values will be read/written. |
| DeviceSize | Enter the number of points of the devices from/to where values will be read/written. |
| DiviceData | Enter the device values to be written. |
| [Open] | Used to open the communication line. |
| [Close] | Used to close the communication line. |
| [GetCpuType] | Used to read the PLC CPU type. |
| [ReadDeviceRandom] | Used to read the data of the devices entered in the "DeviceName" text box. |
| [WriteDeviceRandom] | Used to write the data of the devices entered in the "DeviceName" text box. |
| ReturnCode | Shows the result of executing the corresponding method. |
| Data | Shows the CPU type, CPU type code and read device values. |

(5) For use in another communication path

Run the program after changing the logical station number (only when the utility setting type is used) or the ACT control properties and functions.

5.2 Visual C++® Sample Programs

This section explains the sample programs for Visual C++® which were created using the dispatch interface and custom interface.

These sample programs were created on Visual C++® 6.0.

5.2.1 Dispatch interface

This sample program is designed to read the type of the connection destination CPU and read/write device values using the ActAJ71QE71UDP control or ActEasyIF control on the dispatch interface.

(1) Using method

Load the form and choose the control to be used.

Clicking the button opens the communication line through Ethernet communication.

By clicking the button, the type code of the PLC CPU which is currently connecting the line appears in the "Output Data" text box (top) and the CPU type in the "Output Data" text box (bottom).

Entering the device from where you want to read a value into the "Device Name" text box and clicking the button shows the device data in the "Output Data" text box (top).

To write a device value, enter the device where you want to write a value into the "Device Name" text box and the device value to be written into the "Device Value" text box and click the button.

Clicking the button closes the communication line.

If an error occurs at the execution of any function, an error code appears in the "Return Value" text box.

If an error has occurred, refer to "CHAPTER 6 ERROR CODES" and eliminate the error cause.

(2) Precautions for use of the sample program

(a) When using the ActEasyIF control, set the Ethernet communication information to the logical station number "2" on the communication settings utility before starting the sample program running.

(b) When changing the control used, click the button to close the communication line once, then change the control, and open the line again.

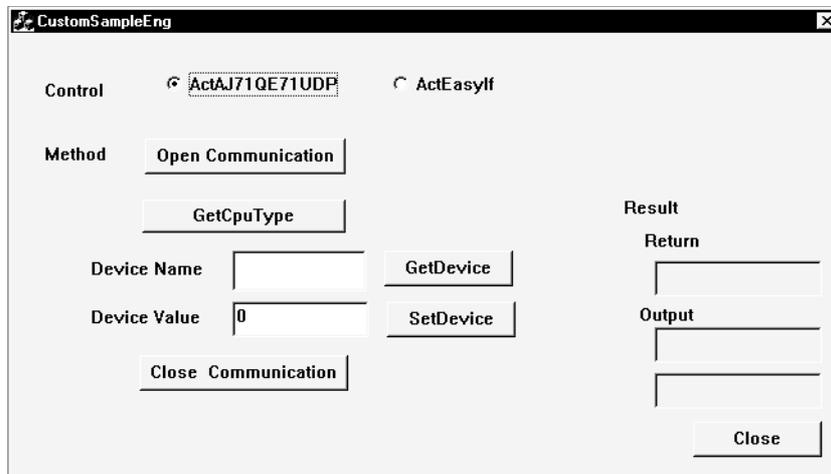
(3) Sample file list

The sample files are installed into the following folders at default installation.

| | |
|--|------------------------------------|
| C:\MELSEC\Act\Sample\Vc\SampleEng\sampleEng.rc | Resource file |
| C:\MELSEC\Act\Sample\Vc\SampleEng\sampleEng.dsw | Project work space |
| C:\MELSEC\Act\Sample\Vc\SampleEng\sampleEng.dsp | Project file |
| C:\MELSEC\Act\Sample\Vc\SampleEng\sampleEng.cpp | Class define file |
| C:\MELSEC\Act\Sample\Vc\SampleEng\sampleEngDlg.cpp | Dialog Implementation source file |
| C:\MELSEC\Act\Sample\Vc\SampleEng\ActDefine.h | ACT Header file |
| C:\MELSEC\Act\Sample\Vc\SampleEng\actaj71qe71udp.cpp | ActAJ71QE71UDP Control source file |
| C:\MELSEC\Act\Sample\Vc\SampleEng\actaj71qe71udp.h | ActAJ71QE71UDP Control header file |
| C:\MELSEC\Act\Sample\Vc\SampleEng\acteasyif.cpp | ActEasyIF Control source file |
| C:\MELSEC\Act\Sample\Vc\SampleEng\acteasyif.h | ActEasyIF Control header file |

(4) Screen

The sample program screen will be explained.



| Item | Description | |
|---------------------|---|--|
| Control | Used to choose the control to be used. | |
| Open Communication | Used to open the communication line. | |
| GetCpuType | Used to read the PLC CPU type. | |
| Device Name | Enter the device from/to where a value will be read/written. | |
| Device Value | Enter the device value to be written. | |
| Close Communication | Used to close the communication line. | |
| GetDevice | Used to read the data of the device entered into the "Device Name" text box. | |
| SetDevice | Used to write the data of the device entered into the "Device Name" text box. | |
| Return | Shows the result of executing the function. | |
| Output | Top | Shows the CPU type code and read device value. |
| | Bottom | Shows the CPU type. |

(5) For use in another communication path

Run the program after changing the logical station number (only when the utility setting type is used) or the ACT control properties and functions.

5.2.2 Custom interface

This sample program is designed to read the type of the connection destination CPU and read/write device values using the ActAJ71QE71UDP control or ActEasyIF control on the custom interface.

(1) Using method

The using method is the same as that of the sample program for dispatch interface.

Refer to "Section 5.2.1 Dispatch interface, (1) Using method".

(2) Precautions for use of the sample program

The precautions are the same as those of the sample program for dispatch interface.

Refer to "Section 5.2.1 Dispatch interface, (2) Precautions for use of the sample program".

(3) Sample file list

The sample files are installed into the following folders at default installation.

| | |
|---|--------------------|
| C:\MELSEC\ACT\SAMPLE\VC\CUSTOMSAMPLE\CustomSample.rc | Resource file |
| C:\MELSEC\ACT\SAMPLE\VC\CUSTOMSAMPLE\CustomSample.dsw | Project work space |
| C:\MELSEC\ACT\SAMPLE\VC\CUSTOMSAMPLE\CustomSample.dsp | Project file |

(4) Screen

The screen is the same as that of the the sample program for dispatch interface.

Refer to "Section 5.2.1 Dispatch interface, (4) Screen".

(5) For use in another communication path

Run the program after changing the logical station number (only when the utility setting type is used) or the ACT control properties and functions.

5.3 VBA Sample Programs

This section explains the sample programs for Excel and Access.

5.3.1 Excel Sample Program

This sample program is designed to log and graph the device values of the PLC CPU using the ActEasyIF control.

This sample program was created on Excel 2000.

(1) Using method

Open the sample sheet.

Type the logical station number, which was specified in communication settings on the communication setting utility, in the "LogicalStationNumber" input cell.

Type the first device of the devices whose data you want to read in the "DeviceName" input cell.

Type the logging interval in the "LoggingTiming" input cell.

Clicking the button starts logging.

When logging starts, the past 10 device values and line graphs appear on the graph.

Clicking the button stops logging.

Note that the logging data on the screen are not cleared.

If an error occurs during execution, the error message appears in the "Message" output cell and the error code appears in the "Return Code" output cell.

If an error has occurred, refer to "6 ERROR CODES" and remove the error cause.

(2) Precautions for use of the sample program

(a) Before running this sample program, make communication settings on the communication setting utility.

(b) When changing the input value, click the button to stop logging once, then change the input value, and click the button to start logging.

(c) In this sample program, the number of device points to be logged is preset to 10 and the number of logging times to 10.

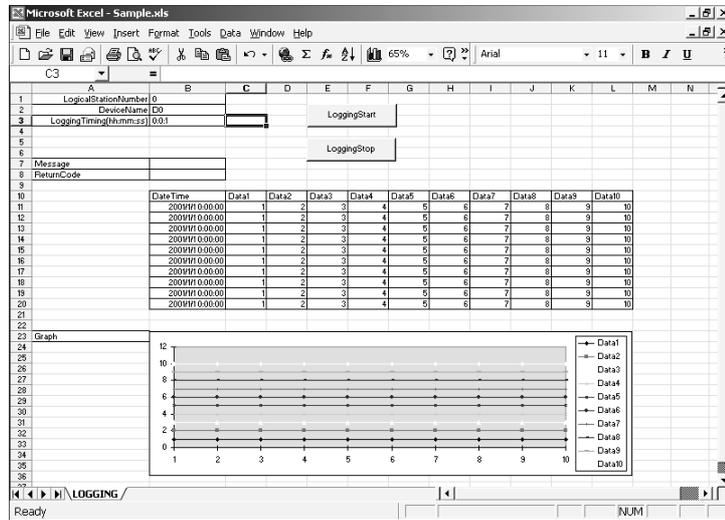
(3) Sample file

By default, the sample program is installed into the following folder.

C:\MELSEC\Act\Sample\Excel\VBA\Sample\Sample.xls Excel file

(4) Screen

The sample program screen will be explained.



| Item | Description |
|---------------------------|--|
| LogicalStationNumber | Enter the logical station number that was specified in communication settings on the communication setting utility. |
| DeviceName | Enter the first device of the devices whose data will be read. |
| LoggingTimeing (hh:mm:ss) | Enter the logging interval. Example: Logging at intervals of 1 second 0:0:1 Logging at intervals of 1 hour 30 minutes 1:30:0 |
| Message | Shows the function execution result. (Character string) |
| ReturnCode | Shows the function execution result. (Hexadecimal value) |
| Time | Shows the logging system time. |
| Data01 to 10 | Show the logged device values. |
| Graph | Shows the past 10 logged values of 10 devices in the form of line graphs. |
| LoggingStart | Used to start logging. |
| LoggingStop | Used to stop logging. |

(5) For use in another communication path

Run the program after changing the logical station number (only when the utility setting type is used) or the ACT control properties and functions.

5.3.2 Access Sample Program

This sample program is designed to log and monitor the device values of the PLC CPU using the ActEasyIF control.

This sample program was created on Access 2000.

(1) Using method

Open the database.

Type the logical station number, which was specified in communication settings on the communication setting utility, in the "LogicalStationNumber" text box.

Type the logging interval in the "LoggingTiming" text box.

Clicking the button starts logging.

Clicking the button stops logging.

Note that the logging data on the screen are not cleared.

If an error occurs during execution, the error message and error code appear in the message box.

If an error has occurred, refer to "6 ERROR CODES" and remove the error cause.

(2) Precautions for use of the sample program

(a) Before running this sample program, make communication settings on the communication setting utility.

(b) When changing the input value, click the button to stop logging once, then change the input value, and click the button to start logging.

(c) In this sample program, devices "D0" to "D4" are preset as the devices to be monitored, and "D10" to "17" as the devices to be logged.
Also, monitoring is designed to occur at 1-second intervals.

(d) This sample program logs device values up to 100 times.
If the number of logging times exceeds 100, the oldest logging data is deleted and the newest logging data is registered.

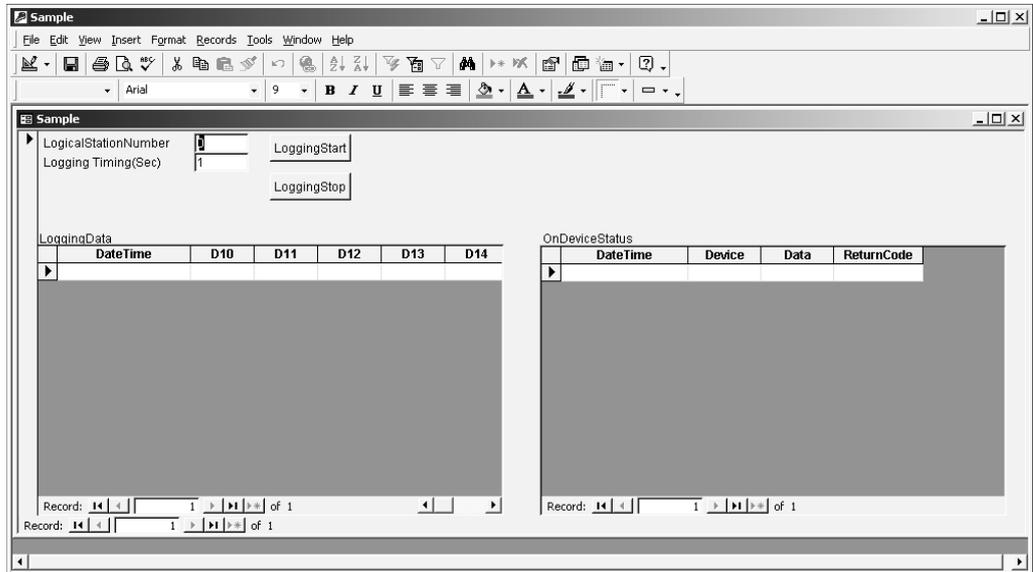
(3) Sample file

By default, the sample program is installed into the following folder.

C:\MELSEC\Act\Sample\Access\VBA\Sample\Sample.mdb Access file

(4) Screen

The sample program screen will be explained.



| Item | Description |
|----------------------|---|
| LogicalStationNumber | Enter the logical station number that was specified in communication settings on the communication setting utility. |
| LoggingTimeing | Enter the logging interval. (Unit: Seconds) |
| LoggingData | Shows the logged data. |
| OnDeviceStatus | Shows the devices whose conditions held among the devices being monitored. |
| LoggingStart | Used to start logging. |
| LoggingStop | Used to stop logging. |

(5) For use in another communication path

Run the program after changing the logical station number (only when the utility setting type is used) or the ACT control properties and functions.

5.4 VBScript Sample Program

This section describes the VBScript sample program.

This sample program is designed to define the device values of the PLC CPU as the capacity and status of a tank and monitor their values using the ActMLEasyIF control. This sample program was created on Microsoft® FrontPage® 2000.

(1) Using method

Opening the sample file opens the communication line to the PLC CPU used.

After that, the device values of the PLC CPU are acquired at intervals of 1 second and are used to display the capacity and status of the tank.

If an error occurs during execution, the error message and error code appear in the message box.

If an error has occurred, refer to "6 ERROR CODES" and remove the error cause.

(2) Precautions for use of the sample program

(a) Before running this sample program, make communication settings on the communication setting utility to set the logical station number to "0".

(b) In this sample program, device "D100" is used to denote the tank capacity, and device "D101" to denote the tank status.

(3) Sample file

By default, the sample program is installed into the following folders.

| | |
|---|------------|
| C:\MELSEC\Act\Sample\VBScript\SampleHTML\Sample.html | HTML file |
| C:\MELSEC\Act\Sample\VBScript\SampleHTML\Pics\Fill.gif | Image file |
| C:\MELSEC\Act\Sample\VBScript\SampleHTML\Pics\Frame.gif | Image file |
| C:\MELSEC\Act\Sample\VBScript\SampleHTML\Pics\LampERR.gif | Image file |
| C:\MELSEC\Act\Sample\VBScript\SampleHTML\Pics\LampOFF.gif | Image file |
| C:\MELSEC\Act\Sample\VBScript\SampleHTML\Pics\tank.gif | Image file |

(4) Screen

The sample program screens will be explained.

(a) Tank capacity : Empty
 Tank status : Normal



(b) Tank capacity : Full
 Tank status : Error



| Item | Description | remark |
|---------------|---|---------------------------|
| Tank capacity | Indicates the tank capacity. (Within the range $0 \leq \text{device value} \leq 200$) | Linked with device "D100" |
| Tank status | Indicates the tank status. Normal (device value = 0) : Blue lamp lit Error (device value <> 0) : Red lamp lit | Linked with device "D101" |

5.5 ASP Sample Program

This sample program is designed to read PLC CPU data and read device values using the ActMLEasyIF control.

(1) File structure

(a) Sample.asp (data input screen)

Initial display screen used to set data for monitoring.

(b) SampleControl.asp (data getting screen)

Used to get the input data of Sample.asp, store them into the global variables, and check them for errors.

(c) SampleMon.asp (data display screen)

Displays data in accordance with the input data of Sample.asp. Displays a message if an error occurs.

(2) Using method

Store Sample.asp, SampleControl.asp and SampleMon.asp in the same folder on the WWW server.

Browse the URL of Sample.asp on the Microsoft® Internet Explorer and display the initial screen.

Entering "MonitorTiming", "LogicalStationNumber", "DeviceName" and

"DeviceSize" on the initial screen and pressing the button start

Open processing, ReadDeviceBlock processing, GetCpuType processing and Close processing.

The input data appear in "MonitorTiming:", "LogicalStationNumber:",

"DeviceName:" and "DeviceSize:" on the data display screen.

"Message:" displays error occurrence processing when an error occurs in the corresponding processing.

"Return Code:" shows the result of the corresponding processing.

"CpuType:" shows the type of the PLC CPU connected.

"ReadData(Hex)" displays the device data of the PLC CPU connected.

If an error has occurred, refer to "6 ERROR CODES" and remove the error cause.

The data display screen repeats updating at the intervals of "MonitorTiming:".

Pressing the button returns to the initial screen.

(3) Precautions for use of the sample program

Before running this sample program, make communication settings on the communication setting utility.

(4) Sample file list

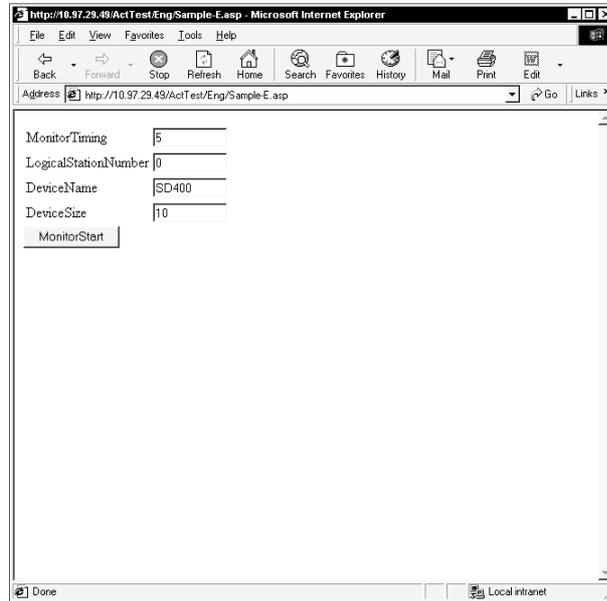
By default, the sample program is installed in the following folders.

| | |
|---|----------|
| C:\MELSEC\Act\Sample\VBScript\SampleASP\Sample.asp | ASP file |
| C:\MELSEC\Act\Sample\VBScript\SampleASP\SampleControl.asp | ASP file |
| C:\MELSEC\Act\Sample\VBScript\SampleASP\SampleMon.asp | ASP file |

(5) Screen

The sample program screens will be explained.

(a) Sample.asp (data input screen)



| Item | Description |
|----------------------|---|
| MonitorTiming | Enter the monitoring intervals. |
| LogicalStationNumber | Enter the logical station number. |
| DeviceName | Enter the device name from where data will be read. |
| DeviceSize | Enter the number of points of the devices from where data will be read. |
| MonitorStart | Used to start monitor processing. |

(b) Data getting screen (SampleControl.asp)

The data getting screen is a page used to store the data entered on the data input screen into the global variables and check them for errors. This screen is not displayed on the Internet Explorer screen.

(c) Data display screen (SampleMon.asp)



| Item | Description |
|------------------------|---|
| MonitorTiming : | Shows the monitoring intervals. |
| LogicalStationNumber : | Shows the logical station number. |
| DeviceName : | Shows the device name from where data will be read. |
| DeviceSize : | Shows the number of points of the devices from where data will be read. |
| Message : | Shows a message at error occurrence. |
| Return Code : | Shows the execution result of the corresponding processing. |
| CpuType | Shows the CPU type. |
| ReadData(Hex) | Shows the read device values. |
| Back | Used to return to the initial screen. |

6 ERROR CODES

This chapter describes the error codes returned by the ACT controls and the error codes returned by the CPUs, modules and network boards.

6.1 Error Codes Returned by the ACT controls

The following table gives the error codes returned by the ACT controls.

| Error Code | Error Definition | Corrective action |
|------------|--|--|
| 0x00000000 | Normal end | ————— |
| 0x01010002 | RUN-time disable error Operation that was performed must not be done during RUN. | Execute after setting to the STOP status. *1 |
| 0x01010005 | Sumcheck error Packet sumcheck was abnormal. | Check for system noise. |
| 0x01010010 | PLC No. error Communication could not be made with the specified station number. | Check the station number set on the communication setup utility. Check the station number set to ActStationNumber. |
| 0x01010013 | Other data error Communication cannot be made for some cause. | Check that the system configuration is not an unsupported configuration. Check that the CPU type setting is correct. Exit the program and restart the IBM-PC/AT compatible. Contact our telephone center. |
| 0x01010018 | Remote request error Remote operation is being performed in the path different from the communicating path. | Cancel the remote operation being performed in the other path. |
| 0x01010020 | Link error Link communications could not be made. | Check that reset operation is not performed for the other end of communication, the control station (master station) or the station passed through by routing. Check that the network parameter setting is correct. |
| 0x01800001 | No command error | The corresponding method does not support. |
| 0x01800002 | Memory lock error | Exit the program and restart the IBM-PC/AT compatible. |
| 0x01800003 | Memory securing error | Exit the program and restart the IBM-PC/AT compatible. Exit other programs and secure free memory area. |
| 0x01800004 | DLL load error | Exit the program and restart the IBM-PC/AT compatible. Exit other programs and secure free memory area. Reinstall MX Component. |
| 0x01800005 | Resource securing error | Exit the program and restart the IBM-PC/AT compatible. Exit other programs and secure free memory area. |
| 0x01801002 | Multi-line open error | Exit the program and restart the IBM-PC/AT compatible. |
| 0x01801003 | Open not yet executed | Exit the program and restart the IBM-PC/AT compatible. |
| 0x01801005 | Specified port error | Exit the program and restart the IBM-PC/AT compatible. Reinstall MX Component. |
| 0x01801006 | Specified module error | Exit the program and restart the IBM-PC/AT compatible. Reinstall MX Component. |

*1: When the network board is relayed, a time-out error may occur. Check the cable state.

| Error Code | Error Definition | Corrective action |
|------------|--|---|
| 0x01801007 | Specified CPU error | Check the CPU type set to ActCpuType. Check that the system configuration is not an unsupported configuration. Exit the program and restart the IBM-PC/AT compatible. Reinstall MX Component. Check the packet type set to ActPacketType. |
| 0x01801008 | Target station access error | Review the target station. |
| 0x0180100C | Registry search failure | Exit the program and restart the IBM-PC/AT compatible. Reinstall MX Component. |
| 0x0180100D | GetProcAddress failure | Exit the program and restart the IBM-PC/AT compatible. Reinstall MX Component. |
| 0x0180100E | DLL non-load error | Exit the program and restart the IBM-PC/AT compatible. Reinstall MX Component. |
| 0x0180100F | Another Object in execution Method cannot be executed because of exclusive control in progress | Execute again after some time. |
| 0x01802001 | Device error The device character string specified in the method is an unauthorised device character string. | Review the device name. |
| 0x01802002 | Device number error The device character string number specified in the method is an unauthorised device number. | Review the device number. |
| 0x01802004 | Sumcheck error The sumcheck value of the received data is abnormal. | Check the module side sumcheck setting. Check the sumcheck property of the control. Check the cable. Exit the program and restart the IBM-PC/AT compatible. Reinstall MX Component. |
| 0x01802005 | Size error The number of points specified in the method is unauthorised. | Check the number of points specified in the method. Review the system, e.g. PLC CPU, module setting and cable status. Exit the program and restart the IBM-PC/AT compatible. Reinstall MX Component. |
| 0x01802006 | Block number error The block specifying number in the device character string specified in the method is unauthorised. | Review the block specifying number in the device character string specified in the method. |
| 0x01802007 | Receive data error The data received is abnormal. | Review the system, e.g. PLC CPU, module setting and cable status. Check the cable. Exit the program and restart the IBM-PC/AT compatible. |
| 0x0180200B | PLC type mismatch The CPU type set to the property and the CPU type set on the communication settings utility do not match the CPU type on the other end of communication. | Set the correct CPU type as the CPU type of the property. Set the correct CPU type on the communication settings utility. Review the system, e.g. PLC CPU, module setting and cable status. |

| Error Code | Error Definition | Corrective action |
|------------|---|---|
| 0x01802016 | Station number specifying error The method does not support the operation performed for the specified station number. | Review the station number. |
| 0x0180201C | Written clock data error Clock data specified for write cannot be written properly since that data is in error. | Review the clock data to be written. |
| 0x0180201D | Online clock data write error Write of clock data failed. Clock data cannot be written since the PLC CPU is during RUN. | Place the PLC CPU in the STOP status. |
| 0x01802020 | First I/O number error The first I/O number specified in the method is an unauthorised value. | Check the value of the first I/O number specified in the method. Using the GPP function, check the PLC CPU parameters (I/O assignment). Exit the program and restart the IBM-PC/AT compatible. |
| 0x01802021 | First address error The buffer address specified in the method is an unauthorised value. | Check the value of the buffer address specified in the method. Exit the program and restart the IBM-PC/AT compatible. |
| 0x01802038 | Clock data read/write error The clock data read/write method was executed for the PLC CPU which does not have the clock devices. | Do not execute clock data read/write. |
| 0x01808001 | Duplex open error | Exit the program and restart the IBM-PC/AT compatible. |
| 0x01808002 | Channel number specifying error The port number set to the property and the port number set on the communication settings utility are unauthorised values. | Set the correct value to the port number of the property. Make communication settings again on the communication settings utility. |
| 0x01808003 | Driver not yet started The network board driver is not started. | Start the driver. |
| 0x01808005 | MUTEX generation error Creation of MUTEX to exercise exclusive control failed. | Exit the program and restart the IBM-PC/AT compatible. Reinstall MX Component. |
| 0x01808007 | Socket object generation error Creation of the Socket object failed. | Check for a running application which uses the same port number. Retry after changing the port number value of the property. Retry after changing the port number value on the communication settings utility. Make Ethernet board and protocol settings on the control panel of the OS. Exit the program and restart the IBM-PC/AT compatible. |
| 0x01808008 | Port connection error Establishment of connection failed. The other end does not respond. | Review the IP address and port number values of the properties. Review the port number value on the communication settings utility. Review the system, e.g. PLC CPU, module setting and cable status. Exit the program and restart the IBM-PC/AT compatible. |

| Error Code | Error Definition | Corrective action |
|------------|--|--|
| 0x01808009 | COM port handle error The handle of the COM port cannot be acquired. The COM port objet cannot be copied. The SOCKET object cannot be copied. | Check for an application which uses the COM port. Exit the program and restart the IBM-PC/AT compatible. |
| 0x0180800A | Buffer size setting error Setting of the COM port buffer size failed. | Check for an application which uses the COM port. Make COM port setting on the control panel of the OS. Exit the program and restart the IBM-PC/AT compatible. |
| 0x0180800B | DCB value acquisition error Acquisition of the COM port DCB value failed. | Check for an application which uses the COM port. Make COM port setting on the control panel of the OS. Exit the program and restart the IBM-PC/AT compatible. |
| 0x0180800C | DCB setting error Setting of the COM port DCB value failed. | Check for an application which uses the COM port. Make COM port setting on the control panel of the OS. Exit the program and restart the IBM-PC/AT compatible. |
| 0x0180800D | Time-out value setting error Setting of the COM port time-out value failed. | Review the time-out value of the property. Review the time-out value on the communication settings utility. Check for an application which uses the COM port. Make COM port setting on the control panel of the OS. Exit the program and restart the IBM-PC/AT compatible. |
| 0x0180800E | Shared memory open error Open processing of shared memory failed. | Check whether the GX Simulator has started. Exit the program and restart the IBM-PC/AT compatible. |
| 0x01808101 | Duplex close error | Exit the program and restart the IBM-PC/AT compatible. |
| 0x01808102 | Handle close error Closing of the COM port handle failed. | Exit the program and restart the IBM-PC/AT compatible. |
| 0x01808103 | Driver close error Closing of the driver handle failed. | Exit the program and restart the IBM-PC/AT compatible. |
| 0x01808201 | Send error Data send failed. | Review the system, e.g. PLC CPU, module setting and cable status. Make COM port setting on the control panel of the OS. Make Ethernet board and protocol settings on the control panel. Exit the program and restart the IBM-PC/AT compatible. |
| 0x01808202 | Send data size error Data send failed. | Exit the program and restart the IBM-PC/AT compatible. |
| 0x01808203 | Queue clear error Clearing of the COM port queue failed. | Exit the program and restart the IBM-PC/AT compatible. Perform Close once and execute Open again. |
| 0x01808301 | Receive error Data receive failed. | Review the system, e.g. PLC CPU, module setting and cable status. Review the time-out value of the property. Review the time-out value on the communication settings utility. Exit the program and restart the IBM-PC/AT compatible. |
| 0x01808304 | Receive buffer size shortage Receive data was larger than the receive buffer size prepared for the system. | Exit the program and restart the IBM-PC/AT compatible. |
| 0x01808401 | Control error Changing of the COM port communication control failed. | Exit the program and restart the IBM-PC/AT compatible. |

| Error Code | Error Definition | Corrective action |
|------------|---|--|
| 0x01808403 | Signal line specifying error Changing of the COM port communication control failed. | Exit the program and restart the IBM-PC/AT compatible. |
| 0x01808404 | Open not yet executed | Execute Open. Exit the program and restart the IBM-PC/AT compatible. |
| 0x01808405 | Communication parameter error The data bit and stop bit combination of the properties is unauthorised. | Review the data bit and stop bit values of the properties. Set them again on the communication settings utility. |
| 0x01808406 | Baudrate value specifying error The baudrate of the property is unauthorised. | Review the baudrate value of the property. Set it again on the communication settings utility. |
| 0x01808407 | Data length error The data bit value of the property is unauthorised. | 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 unauthorised. | Review the parity value of the property. Set it again on the communication settings utility. |
| 0x01808409 | Stop bit specifying error The stop bit value of the property is unauthorised. | 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 unauthorised. | Review the control value of the property. Set it again on the communication settings utility. |
| 0x0180840B | Time-out error Though the time-out period had elapsed, data could not be received. | Review the time-out value of the property. Set it again on the communication settings utility. Review the system, e.g. PLC CPU, module setting and cable status. Perform Close once and execute Open again. Exit the program and restart the IBM-PC/AT compatible. |
| 0x0180840C | Connect error | Exit the program and restart the IBM-PC/AT compatible. |
| 0x0180840D | Duplex connect error | Exit the program and restart the IBM-PC/AT compatible. |
| 0x0180840E | Attach failure Attaching of the socket object failed. | Exit the program and restart the IBM-PC/AT compatible. |
| 0x0180840F | Signal line status acquisition failure Acquisition of the COM port signal line status failed. | Exit the program and restart the IBM-PC/AT compatible. |
| 0x01808410 | CD signal line OFF The CD signal on the other end of communication is in the OFF status. | Review the system, e.g. PLC CPU, module setting and cable status. Exit the program and restart the IBM-PC/AT compatible. |
| 0x01808411 | Password mismatch error | Check the remote password of the property. |
| 0x01808501 | USB driver load error Loading of the USB driver failed. | Exit the program and restart the IBM-PC/AT compatible. Reinstall MX Component. |
| 0x01808502 | USB driver connect error Connection of the USB driver failed. | Exit the program and restart the IBM-PC/AT compatible. Reinstall MX Component. |
| 0x01808503 | USB driver send error Data send failed. | Review the system, e.g. PLC CPU, module setting and cable status. Make USB setting on the control panel (device manger) of the OS. Exit the program and restart the IBM-PC/AT compatible. |

| Error Code | Error Definition | Corrective action |
|------------|--|---|
| 0x01808504 | USB driver receive error Data receive failed. | Review the system, e.g. PLC CPU, module setting and cable status. Make USB setting on the control panel (device manger) of the OS. Exit the program and restart the IBM-PC/AT compatible. |
| 0x01808506 | USB driver initialisation error Initialisation of the USB driver failed. | Make USB setting on the control panel (device manger) of the OS. Exit the program and restart the IBM-PC/AT compatible. |
| 0x01808507 | Other USB error Error related to data send/receive occurred. | Disconnect the cable once, then reconnect. Exit the program and restart the IBM-PC/AT compatible. Reinstall MX Component. |
| 0x04000001 | No command error The specified CPU type cannot be used to perform processing. | Check the CPU type set to ActCpuType. Check whether the system configuration is supported or not. Exit the program and restart the IBM-PC/AT compatible. Reinstall MX Component. |
| 0x04000004 | Internal server DLL load error Start of the internal server failed. | Check for the deleted or moved installation file of MX Component. Exit the program and restart the IBM-PC/AT compatible. Reinstall MX Component. |
| 0x10000002 | Start of communication DLL of MX Component failed. | Exit the program and restart the IBM-PC/AT compatible. Reinstall MX Component. |
| 0x10000003 | Open failed. (DiskDrive) | Exit the program and restart the IBM-PC/AT compatible. Reinstall MX Component. |
| 0x10000004 | Duplex open error | Exit the program and restart the IBM-PC/AT compatible. |
| 0x1000000C | Execution failed since another application or thread is making a request. | Execute again after some time. Perform programming according to the multithread rules of COM and ActiveX. Exit the program and restart the IBM-PC/AT compatible. |
| 0x10000011 | Memory securing error | Exit the program and restart the IBM-PC/AT compatible. Reinstall MX Component. |
| 0x10000012 | Open not yet executed | Exit the program and restart the IBM-PC/AT compatible. |
| 0x10000017 | The specified size (number of devices) is unauthorised. | Check the number of points specified in the method. Exit the program and restart the IBM-PC/AT compatible. |
| 0x10000018 | There is no registered device. | Exit the program and restart the IBM-PC/AT compatible. |
| 0x1000001E | Registry search failed. | Exit the program and restart the IBM-PC/AT compatible. Exit other programs and secure free memory area. Reinstall MX Component. |
| 0x10000032 | Specified device error | Review the specified device data. Exit the program and restart the IBM-PC/AT compatible. Exit other programs and secure free memory area. |
| 0x10000033 | Specified device range error | Review the specified device data. Exit the program and restart the IBM-PC/AT compatible. Exit other programs and secure free memory area. |
| 0x10000040 | Server start failed. | Exit the program and restart the IBM-PC/AT compatible. |
| 0xF0000001 | No-license error The license is not given to the IBM-PC/AT compatible. | Using the license FD, give the license to the IBM-PC/AT compatible. |
| 0xF0000002 | Set data read error Reading of the set data of the logical station number failed. | Specify the correct logical station number. Set the logical station number on the communication settings utility. |

| Error Code | Error Definition | Corrective action |
|------------|--|---|
| 0xF0000003 | Already open error The Open method was executed in the open status. | When changing the communication target CPU, execute the Open method after performing Close. |
| 0xF0000004 | Not yet open error The Open method is not yet executed. | After executing the Open method, execute the corresponding method. |
| 0xF0000005 | Initialisation error Initialisation of the object possessed internally in MX Component failed. | Exit the program and restart the IBM-PC/AT compatible. Reinstall MX Component. |
| 0xF0000006 | Memory securing error Securing of MX Component internal memory failed. | Exit the program and restart the IBM-PC/AT compatible. Exit other programs and secure free memory area. |
| 0xF0000007 | Function non-support error The method does not support. | The corresponding method does not support. |
| 0xF1000001 | Character code conversion error Character code conversion (UNICODE→ASCII code or ASCII code→UNICODE) failed. | Check the character string specified in the method. The ASCII character string acquired from the PLC CPU is abnormal. Review the system, e.g. PLC CPU, module setting and cable status. Exit the program and restart the IBM-PC/AT compatible. Retry the GetCpuType method. |
| 0xF1000002 | First I/O number error The first I/O number specified is an unauthorised value. A matching first I/O number does not exist. | Check the value of the first I/O number specified in the method. Using the GPP function, check the PLC CPU parameters (I/O assignment). |
| 0xF1000003 | Buffer address error The buffer address specified is an unauthorised value. The buffer address is outside the range. | 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. | Perform reopen processing. Review the system, e.g. PLC CPU, module setting and cable status. Retry. Exit the program. |
| 0xF1000005 | Size error The size specified in the read/write method is abnormal. The read/write first number plus size exceeds the device or buffer area. | Check the size specified in the method. |
| 0xF1000006 | Operation error The operation specified for remote operation is an abnormal value. | Check the operation specifying value specified in the method. |
| 0xF1000007 | Clock data error The clock data is abnormal. | Check the clock data specified in the method. Set the correct clock data to the clock data of the PLC CPU. |

| Error Code | Error Definition | Corrective action |
|------------|---|--|
| 0xF1000008 | Monitored device registration count excess The number of device points registered in the EntryDeviceStatus method was 0 or less. The number of device points registered in the EntryDeviceStatus method was more than 20. | Register the device points between 1 and 20 in the EntryDeviceStaus method. |
| 0xF1000009 | Monitored device data registration error | After making deregistration in the FreeDeviceStatus method, execute the EntryDeviceStatus method again. |
| 0xF1000010 | Device status monitor processing failed to start. Device status monitor processing failed to end. | Start/end the device status monitor processing again in the EntryDeviceStatus() method. |
| 0xF1000011 | The VARIANT argument data type is wrong. | Reexamine the data type specified for the VARIANT argument. <ul style="list-style-type: none"> • Check whether the array variable size is large enough. • Check whether the data type specified in the corresponding method has been set. |
| 0xF1000012 | The device status monitoring time interval is a value outside the range 1 second to 1 hour (1 to 3600). | Specify the device status monitoring time between 1 and 3600. |

6.2 Error Codes Returned by the CPUs, Modules and Network Boards

This section explains the error codes returned by the CPUs, modules and network boards.

| POINT |
|---|
| <p>Error codes may not be returned as "(1) Error code list" describes. At the beginning, confirm "(2) Precautions for confirming error code", and then refer to "(1) Error code list".</p> |

(1) Error code list

If the CPU, module or network board has detected an error, any of the error codes indicated in the following table is returned.

The two upper bytes denote Error detection module, and the two lower bytes denote the error code returned by Error detection module.

For detail of the error, refer to the manual of CPU, module or network board corresponding to the error code.

| Error Code | Error detection module |
|-----------------------------|--|
| 0x01010000 to 0x0101FFFF *1 | QCPU (A mode), ACPU, motion controller CPU |
| 0x01020000 to 0x0102FFFF *1 | QnACPU |
| 0x01030000 to 0x0103FFFF *1 | C24 |
| 0x01040000 to 0x0104FFFF *1 | QC24(N) |
| 0x01050000 to 0x0105FFFF *1 | E71 |
| 0x01060000 to 0x0106FFFF *1 | QE71 |
| 0x01070000 to 0x0107FFFF *1 | MELSECNET/H board, MELSECNET/10 board, MELSECNET(II) board, CC-Link board, CPU board, AF board |
| 0x01090000 to 0x0109FFFF *1 | FXCPU |
| 0x010A0000 to 0x010AFFFF *1 | QCPU (Q mode) |
| 0x010B0000 to 0x010BFFFF *1 | Q series-compatible C24 |
| 0x010C0000 to 0x010CFFFF *1 | Q series-compatible E71 |
| 0x010D0000 to 0x010DFFFF *1 | PC CPU module |

*1 : Refer to "Point"

(2) Precautions for confirming error code

Precautions for confirming the error codes returned by CPU, module and network board are described below.

(a) Property setting error

If the used system configuration does not match the preset property values, the two upper bytes do not indicate the correct error detection module.

For example, when the property values of the ACPU have been set to ActCpuType as those of the QCPU (Q mode), the two upper bytes may indicate that Error detection module is the ACPU.

In such a case, make communication again after checking the system configuration and all preset property values.

When the ActMulti or ActMLMulti control is used, check the settings of the communication setting utility.

(b) When AJ71E71 or AJ71QE71 is used

If the two lower-byte error code that occurred during E71 or QE71 communication is not given in the E71 or QE71 manual, check whether the DIP switch (SW2) at the front of the E71 or QE71 module matches the value set to the ActPacketType property.

If the DIP switch setting is incorrect, the packet format (ASCII/binary) is different, disabling correct recognition of the error code returned from the module.

(c) At access to another station

At access to another station, the error code of the relayed module used (MELSECNET/H, MELSECNET/10, MELSECNET(II), CC-Link, computer link, Ethernet module) may enter the two lower bytes.

In such a case, the two upper bytes that indicate Error detection module may not always match the faulty module. Therefore, after checking the system configuration, also check the manuals of the used CPU, relayed network module and network board.

6.3 HRESULT Type Error Codes

Normally, the ActiveX control returns the HRESULT type returned value. So does the ACT control.

When the custom interface is used, the returned value is equivalent to the returned value of method API.

When the dispatch interface is used, the HRESULT type returned value can be acquired by performing exception processing.

The following table indicates the HRESULT type returned values of the ACT controls.

| Returned Value | Termination Status | Description |
|----------------|----------------------|---|
| S_OK | Normal termination | Function processing terminated normally. |
| S_FALSE | Normal termination | Function processing (as ActiveX control) terminated normally, but operation (access to PLC) failed. |
| E_POINTER | Abnormal termination | The pointer passed to the function is abnormal. |
| E_OUTOFMEMORY | Abnormal termination | Memory securing or object creation failed. |
| E_FAIL | Abnormal termination | An indefinite error occurred. |

POINT

If exception processing for acquiring the HRESULT type returned value has not been performed, the dispatch interface shows the error dialog box on the OS level when E_POINTER (E_XXXXX defined returned value) or the like is returned from the ACT control.

MX Component Version 2

Programming Manual

| | |
|--------------------------|----------------|
| MODEL | MELS2-ACTE-P-E |
| MODEL CODE | 13JF65 |
| SH(NA)-080155-A(0104)MEE | |



HEAD OFFICE : MITSUBISHI DENKI BLDG MARUNOUCHI TOKYO 100-8310 TELEX : J24532 CABLE MELCO TOKYO
NAGOYA WORKS : 1-14 , YADA-MINAMI 5 , HIGASHI-KU, NAGOYA , JAPAN

When exported from Japan, this manual does not require application to the Ministry of Economy, Trade and Industry for service transaction permission.

Specifications subject to change without notice.