B-2451(0)

Network User's Manual

FC6A series MICROSmart
SmartAXIS FT2J/1J
MICRO/I HG2J/1J, HG5G/4G/3G/2G-V

IDEC CORPORATION

Revision history

July 2024 First Edition

Caution

- The contents of this manual, the WindLDR and WindO/I-NV4 applications are copyright, and all rights are reserved by IDEC Corporation. Unauthorized reproduction, reprinting, sale, transfer, or rental is prohibited.
- The contents of this manual, the WindLDR and WindO/I-NV4 applications are subject to change without notice.
- IDEC Corporation accepts no responsibility for circumstances arising from the use of this manual, the WindLDR or WindO/I-NV4 application.
- Please contact your vendor or IDEC Corporation with any problems regarding the operation of this product.
- This product cannot be directly connected to the communication lines (including public wireless LAN) of telecommunication carriers (mobile communication companies, fixed-line communication companies, Internet providers, etc.). When connecting this product to the Internet, be sure to connect via a device, such as a router.

Trademarks

MICROSmart, SmartAXIS, MICRO/I, WindLDR and WindO/I are registered trademarks of IDEC CORPORATION in JAPAN.

All other company names and product names used in this manual are trademarks of their respective owners.

Preface

This manual describes the necessary communication settings and precautions when using protocols such as BACnet/IP communication and EtherNet/IP communication.

Please read this manual and the user's manual and hardware manual of your product carefully before use and ensure that you fully understand the functions and performance of the FC6A, display unit integrated controller SmartAXIS FT2J/1J, programmable display MICRO/I HG2J/1J, HG5G/4G/3G/2G-V, WindLDR and the programmable display drawing software WindO/I-NV4.

IDEC Corporation makes the latest product manual PDFs available on our website at no additional cost. Please download the latest product manual PDFs from our website.

Read the following materials as necessary for your particular application.

References	Content
FC6A Series MICROSmart User's Manual (PDF)	Describes product specifications, installation and wiring instructions, instructions for basic programming operations and special functions, device and instruction lists, and troubleshooting procedures for the FC6A.
FC6A Series MICROSmart PID Module User's Manual (PDF)	Describes PID module specifications and functions.
WindO/I-NV4 User's Manual (PDF)	Describes the hardware specifications of the FT2J/1J, HG2J/1J, HG5G/4G/3G/2G-V.
Ladder Programming Manual (PDF)	Describes basic operations for programming with ladders, monitoring methods on the WindLDR, instruction lists, and details of each instruction.
SmartAXIS Hardware Manual (PDF)	Describes the product specifications, installation and wiring instructions of the FT2J/1J, optional items, and I/O cartridges.
MICRO/I Hardware Manual (PDF)	Describes the product specifications, installation and wiring instructions of the HG2J/1J and HG5G/4G/3G/2G-V, optional items, and expansion modules.
Network User's Manual (This manual)	Describes how to configure and use specific communication protocols (BACnet/IP communication, EtherNet/IP communication, MQTT communication).

Symbols Used in this Manual

This manual uses the following symbols to facilitate explanation.

Symbols

 ..... Information that requires special attention. Failure to operate the product in accordance with the information provided can lead to serious injury or damage.

 ..... Information relating to requests or material to reference in the use of a function

 ..... Useful information relating to a function

 ..... Indicates the chapter and page of related reference information.

OK ..... Screen buttons are indicated by **bold** text or by using the actual graphic icon.

**** ..... Controls are indicated by **bold** text.

Abbreviations, Generic Terms, and Terminology Used in this Manual

Item	Description
All-in-One CPU module	General term for FC6A-C*****E.
CAN J1939 All-in-One CPU module	General term for FC6A-C40***EJ.
Plus CPU module	General term for FC6A-D****CEE.
FC6A	Generic term for All-in-One CPU module, CAN J1939 All-in-One CPU module, Plus CPU module.
FT2J	The name is short for SmartAXIS FT2J-7U22*AF-B.
FT1J	The name is short for SmartAXIS FT1J-4F1**AG-*.
FT2J/1J	General term for FT2J and FT1J.
HG2J	The name is short for MICRO/I HG2J-7UT22TF-B.
HG1J	The name is short for MICRO/I HG1J-4FT22T*-*.
HG5G-V	The name is short for MICRO/I HG5G-VFXT22MF-B.
HG4G-V	The name is short for MICRO/I HG4G-VCXT22MF-B.
HG3G-V	The name is short for MICRO/I HG3G-V*XT22MF-*.
HG2G-V	The name is short for MICRO/I HG2G-V5FT22TF-*.
HG2J/1J	General term for HG2J and HG1J.
HG5G/4G/3G/2G-V	The format used to refer to HG5G-V, HG4G-V, HG3G-V and HG2G-V.
WindO/I-NV4	Integrated configuration software application for creating projects of the FT2J/1J, HG2J/1J, and HG5G/4G/3G/2G-V.
WindLDR	Software application for creating ladder programs. Editor for control function settings that can be launched from WindO/I-NV4.
HMI function	Functions such as screens and settings of the main unit that can be created by the WindO/I-NV4.
Control function	Functions such as ladder program and setting of SmartAXIS that can be created by the WindLDR.
Project	Data including image data required for operating the main unit, which is created with WindO/I-NV4.
Function area settings	General term for various settings of the control functions of FC6A and SmartAXIS. These are the settings to be configured in the Settings tab of WindLDR and the module configuration editor.
Ladder program	Generic term used to refer to main program, subroutine program, user-defined macros created using WindLDR.
User program	It refers to the data that combines function area settings and ladder programs configured by the ladder programming software WindLDR.
Device Address	Memory that is capable of storing values in unit of bits or words loaded on the main unit and external device.
Internal Device	The generic term for internal device addressing on the main unit such as internal relays, registers, etc.
HMI Device	Generic term for internal devices that control the HMI functions.
Control Device	Generic term for internal devices that control the control functions.
System Software	Software for basic control and management of the main unit.
External Memory Device	Generic term for an SD memory card and a USB flash drive.

Table of Contents

1: BACnet/IP Communication

FC6A Plus FC6A All-in-One FT2J/1J HG2J/1J HG5G/4G/3G/2G-V

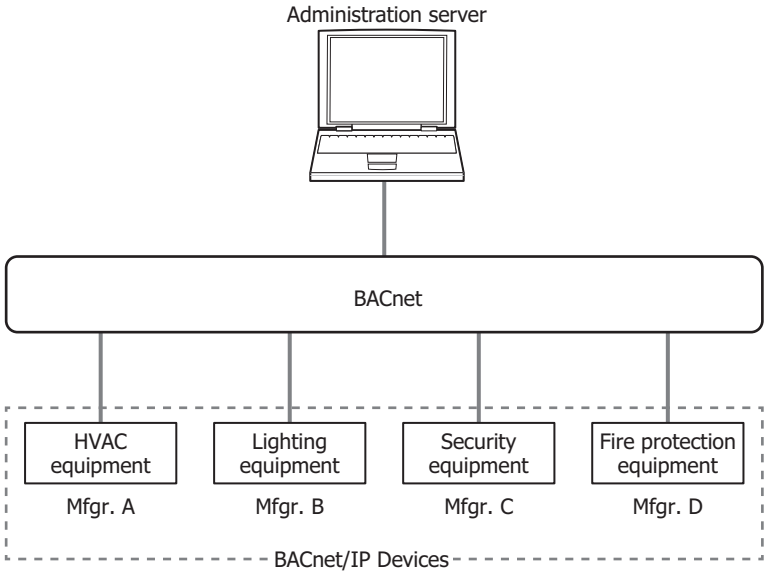
This chapter describes the BACnet/IP communication function.

Overview

The main unit supports BACnet communication using the Internet protocol (IP), which is called BACnet/IP.

BACnet stands for "Building Automation and Control Networking Protocol". In building equipment, it is an open protocol that has been standardized for interoperability between systems built by different manufacturers, and a communication standard for building networks.

Previous air conditioning, lighting, security, and fire protection systems were connected using methods unique to the building equipment and systems manufacturers. However, conforming to BACnet allows equipment and systems to be connected and monitored using a common method.



A device connected by BACnet, and a device compatible with BACnet/IP is called a BACnet/IP device.

Specifications

Supported models

Supported IDEC's PLCs and programmable displays are as follows.

Series Name	Abbreviation	Type Number
MICROSmart	FC6A	FC6A-D16**CEE, FC6A-D32**CEE
SmartAXIS	FT2J	FT2J-7U22*AF-B
	FT1J	FT1J-4F1**AG-*
MICRO/I	HG2J	HG2J-7UT22TF-B
	HG1J	HG1J-4FT22TG-*
	HG5G/4G/3G/2G-V	HG5G-VFXT22MF-B, HG4G-VCXT22MF-B, HG3G-V*XT22MF-*, HG2G-V5FT22TF-*

Basic specifications

Model (Abbreviation)		FC6A	FT2J/1J	HG2J/1J	HG5G/4G/3G/2G-V
Interface		Ethernet port 1	Ethernet Interface		
Applicable Standards		ANSI/ASHRAE135-2012			
Standard Specifications	Protocol	BACnet/IP			
	Profile	B-ASC			
	Object Type	Device Object, Analog Input Object, Analog Output Object, Analog Value Object, Binary Input Object, Binary Output Object, Binary Value Object			
	Number of Objects	256 maximum*1			
	BIBBs	DS-RP-B, DS-WP-B, DS-RPM-B, DS-WPM-B, DS-COV-B, DS-COVU-B, DM-DDB-B, DM-DOB-B, DM-DCC-B			
	BBMD	None-BBMD Device			
	Virtual Device	Not supported			
	Foreign Device	Supported			
Subscribed COV Function	Number of Requests That Can Be Accepted	256 requests maximum			
Unsubscribed COV Function	Transmission Unit	Every object			
	Transmission Cycle	1 to 65,535 msec*2			
Foreign Device Function	Registration Method	Registration as needed by registration trigger device			
	Lifetime	0 to 65,535 sec			
Device Binding Function		• Synchronization between properties and device addresses*3 • Data type conversion of Present _Value*4 • Coefficient conversion of Present _Value*4			

*1 Device Object is not included.
*2 The transmission cycle is set for all objects.
*3 The properties of objects created in internal memory are synchronized with specified device addresses.
*4 Supported objects are Analog Input Object, Analog Output Object, and Analog Value Object.

About BACnet

■Objects

Information, such as input and output values handled by BACnet/IP devices that support BACnet/IP, are managed in units called objects. Objects are separated into a number of different types, depend on the content of the object, and these are called object types. The object types supported by BACnet/IP in the main unit are as follows.

YES: Supported, NO: Not supported

Object Type				Standards
Name		Abbreviation	Identifier	ANSI/ASHRAE135-2012
Basic input and output	Analog Input Object	AI	0	YES
	Analog Output Object	AO	1	YES
	Analog Value Object	AV	2	YES
	Binary Input Object	BI	3	YES
	Binary Output Object	BO	4	YES
	Binary Value Object	BV	5	YES
BACnet device attributes	Device Object	DV	8	YES

The main unit can set objects for each object type. This information can be written to and read from BACnet/IP devices on the same BACnet/IP network. Regardless of the object type, the maximum number of objects that can be set is 256 in total.

For details on objects, refer to " Objects" on page 1-34.

■Properties

Properties are the detailed information and attributes of each object.

For details on the properties of each object type, refer to " Objects" on page 1-34.

■Services

Services are interfaces used to exchange information between BACnet/IP devices. Services are classified into the client side that initiates services and the server side that executes services.

The Services supported by BACnet/IP in the main unit are as follows.

YES: Supported, NO: Not supported

Services	Initiate*1	Execute*2
ReadProperty	NO	YES
ReadPropertyMultiple	NO	YES
WriteProperty	NO	YES
WritePropertyMultiple	NO	YES
SubscribeCOV	NO	YES
ConfirmedCOVNotification	YES	NO
UnconfirmedCOVNotification	YES	NO
Who-Is	NO	YES
I-Am	YES	NO
Who-Has	NO	YES
I-Have	YES	NO
DeviceCommunicationControl	NO	YES

*1 The main unit initiates services for other BACnet/IP devices.

*2 The main unit executes services that are initiated from other BACnet/IP devices.

For details on services, refer to the ANSI/ASHRAE 135-2012 (ISSN 1041-2336) standard or "BACnet Building Automation Data Communication Protocol", a book published by the Institute of Electrical Installation Engineers of Japan.

■**BIBB**

BACnet interoperability building blocks (BIBB) are groupings of multiple services for each function that will be implemented. BIBBs are classified into the client side that uses the function and the server side that provides the function. '-A' is appended to the end of the client-side BIBBs, and '-B' is appended to the end of the server-side BIBBs. BACnet/IP devices define the functions they support using BIBBs.

The BIBB supported by BACnet/IP in the main unit are as follows.

BIBB Category	BIBB		Services
Data Sharing	DS-RP-B	Data Sharing Read Property B	ReadProperty
	DS-WP-B	Data Sharing Write Property B	ReadPropertyMultiple
	DS-RPM-B	Data Sharing Read Property Multiple B	WriteProperty
	DS-WPM-B	Data Sharing Write Property Multiple B	WritePropertyMultiple
	DS-COV-B	Data Sharing COV B	SubscribeCOV ConfirmedCOVNotification UnconfirmedCOVNotification
	DS-COVU-B	Data Sharing COV Unsubscribed B	UnconfirmedCOVNotification
Device & Network Management	DM-DDB-B	Device Management Dynamic Device Binding B (Who-Is, I-Am)	Who-Is I-Am
	DM-DOB-B	Device Management Dynamic Object Binding B (Who-Has, I-Have)	Who-Has I-Have
	DM-DCC-B	Device Management Device Communication Control B	DeviceCommunicationControl

For details on BIBBs, refer to "BACnet Building Automation Data Communication Protocol", a book published by the Institute of Electrical Installation Engineers of Japan.

Function

The main unit provides the following functions as a single BACnet/IP device.

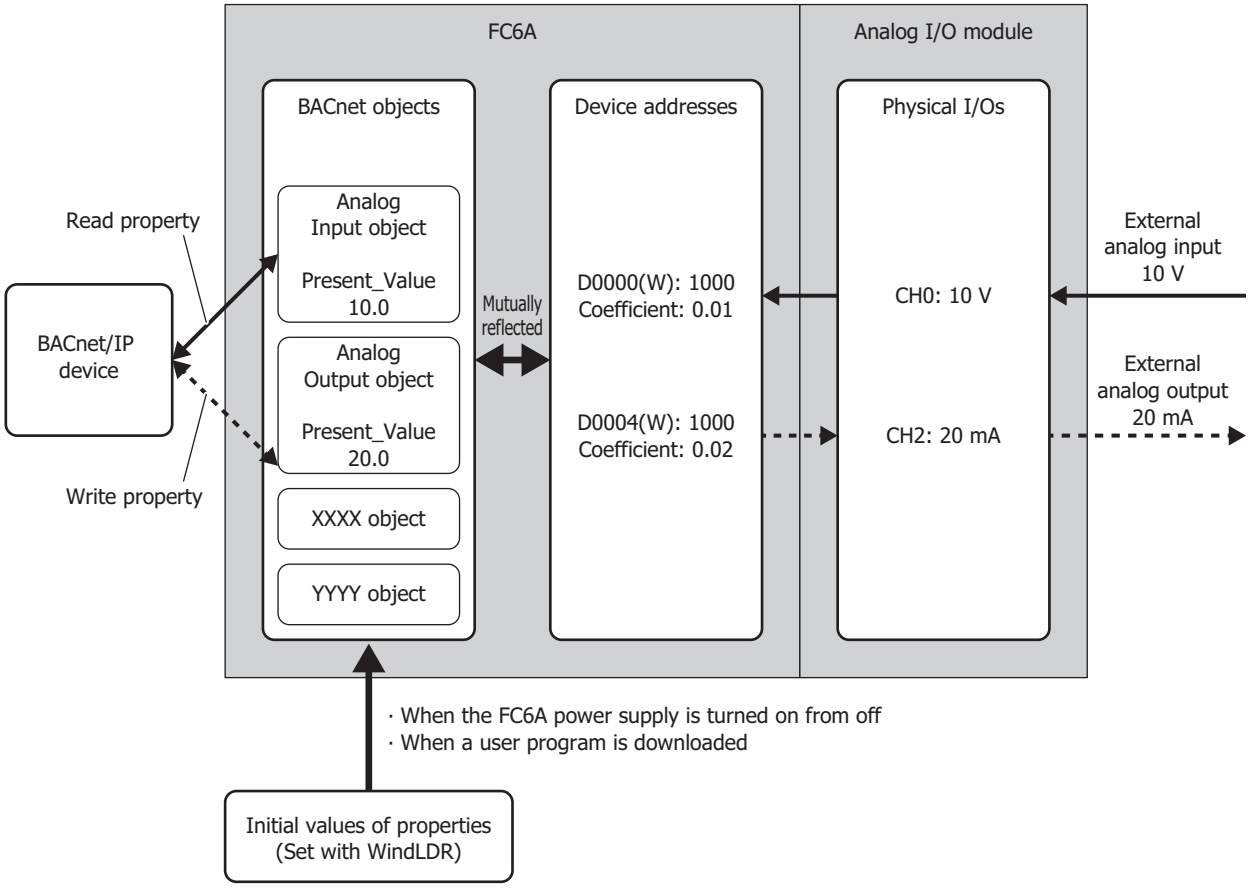
- Object and Device Binding function (page 1-5)
- Read Property function (page 1-6)
- Write Property function (page 1-6)
- Subscribed COV function (page 1-7)
- Unsubscribed COV function (page 1-9)
- Foreign Device function (page 1-10)

Object and Device Binding Function

A portion of the properties held by objects created in the main unit can be allocated to device addresses, such as data registers, and those device addresses can be written to and read.

For example, the analog input value of the FC6A can be stored in a data register (D0000) allocated as Present_Value of the Analog Input object, and that analog input value can be read by BACnet/IP devices. The analog output value of the FC6A can also be changed from BACnet/IP devices by allocating the data register (D0004) that stores the analog output value as Present_Value of the Analog Output object.

The types and numbers of objects can be freely set. The initial values of properties are also set with WindLDR or WindO/I-NV4.

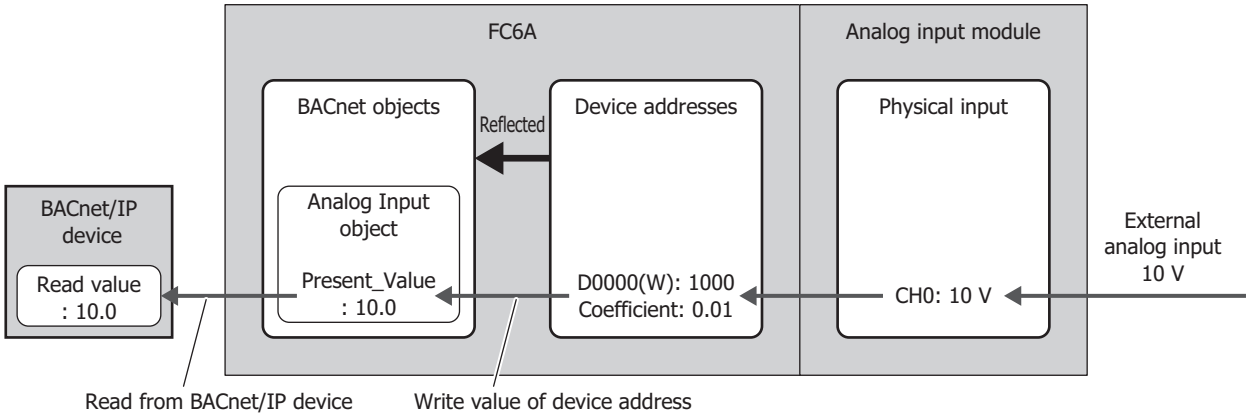


- The processing to mutually reflect properties and device addresses is performed with no relationship to the execution cycle of the ladder program or the processing for parts on the screen of the main unit. Because device addresses allocated to objects will be written and read during execution of the ladder program or the processing for parts on the screen, create them so that no problems will occur when device addresses are referenced or refreshed.
- If a device address changes at an interval which is shorter than the cycle at which changes are reflected to the property of an object, those changes may not be reflected to the property. To reflect those changes to the property, keep the values of device addresses for 1 second.
  For example, in the FC6A type, the changes may not be reflected to the property when M0000, which has been allocated to Present_Value of a Binary Input object, is turned on for only 10 msec. To reflect those changes to the property, ensure that M0000 stays 1 for 1 second.

Read Property Function

In the Read Property function, the main unit returns the value of a property when there is a read property request from a BACnet/IP device. This function is implemented using DS-RP-B (Data Sharing Read Property B) and DS-RPM-B (Data Sharing Read Property Multiple B).

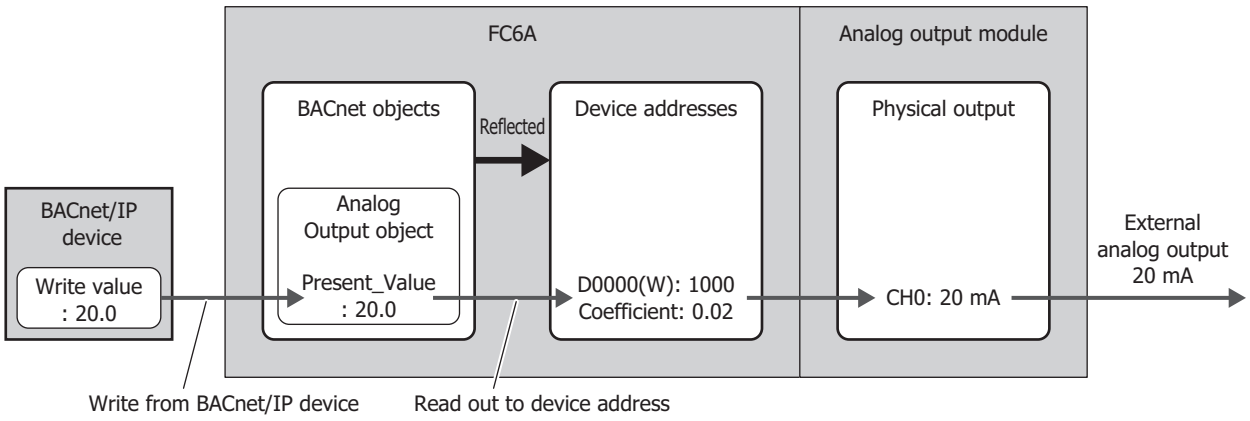
The below diagram shows a BACnet/IP device reading the analog input value of the FC6A that has been associated with the Analog Input object.



Write Property Function

In the Write Property function, the main unit writes a value to the property of an object when there is a write property request from a BACnet/IP device. This function is implemented using DS-WP-B (Data Sharing Write Property B) and DS-WPM-B (Data Sharing Read Property Multiple B).

The below diagram shows a BACnet/IP device changing the analog output value of the FC6A that has been associated with the Analog Output object.



Subscribed COV (COV) function

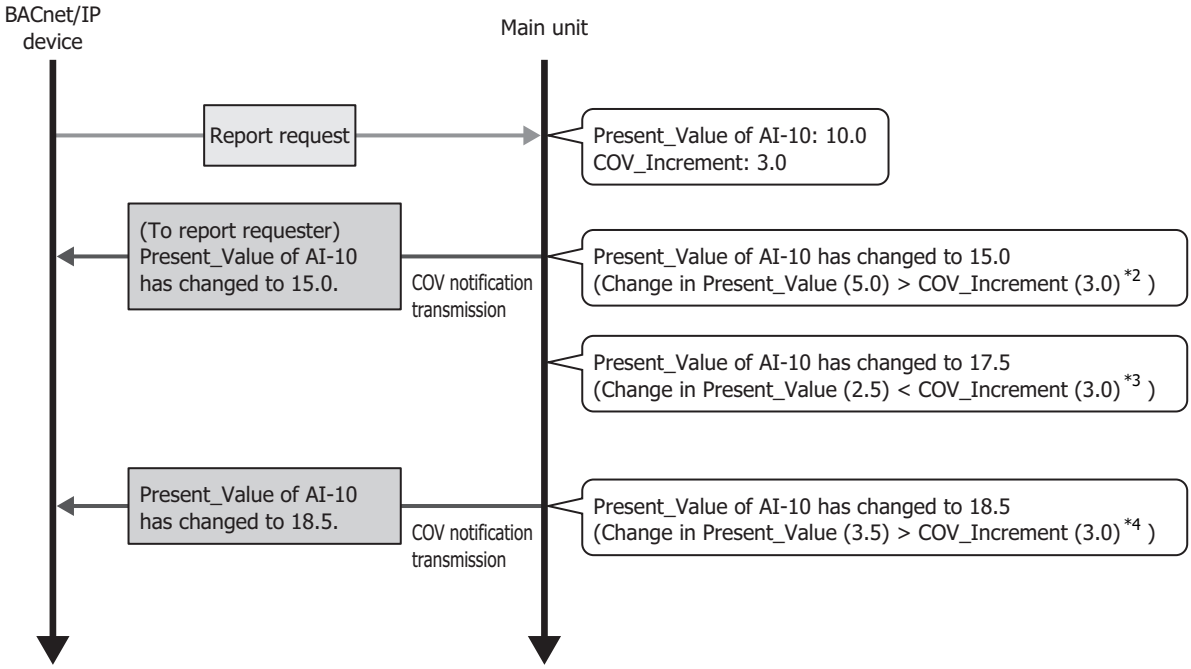
In the COV (Change Of Value) function, the main unit monitors an object for which a report request has been received from a BACnet/IP device. Then when Present_Value or Status_Flags has changed, the main unit notifies the BACnet/IP device of the change.

The properties of objects that correspond to the COV function and the timing of when to provide notification of changes are given as follows.

Object	Properties	Timing to Provide Notification of Change
Analog Input Analog Output Analog Value	Present_Value Status_Flags	Notification of change is provided when (1) or (2) as follows. (1) When Present_Value has changed to a value greater than or equal to the value set for COV_Increment (starting point is from when the previous COV notification was sent)*1 (2) When any of the bits in Status_Flags have changed
Binary Input Binary Output Binary Value	Present_Value Status_Flags	Notification of change is provided when (1) or (2) as follows. (1) When Present_Value has changed (2) When any of the bits in Status_Flags have changed

*1 For example, the following diagram shows when a report request is received from a BACnet/IP device and AI-10 (Analog Input object, instance number 10) has the status in the following table.

AI-10 Properties	Value
Present_Value	10.0
COV_Increment	3.0



*2 This change corresponds to timing to provide notification of change (1).
Change in Present_Value = Present value - Value when report request was received = 15.0 - 10.0 = 5.0
Because COV_Increment = 3.0, change in present value ≥ COV_Increment has been satisfied.

*3 Change in Present_Value = Present value - Value when previous COV notification was transmitted = 17.5 - 15.0 = 2.5
Change in Present_Value < COV_Increment, and the condition for timing to provide notification of change (1) is not satisfied.
Therefore, COV notification is not sent.

*4 This change corresponds to timing to provide notification of change (1).
Change in Present_Value = Present value - Value when previous COV notification was transmitted = 18.5 - 15.0 = 3.5
Because COV_Increment = 3.0, Change in Present Value ≥ COV_Increment has been satisfied.

- Notification may not be provided for changes in values that are faster than the synchronization cycle of objects and device addresses.
- The COV notification may not be sent when concurrent transmission requests exceed the transmission queue limit.

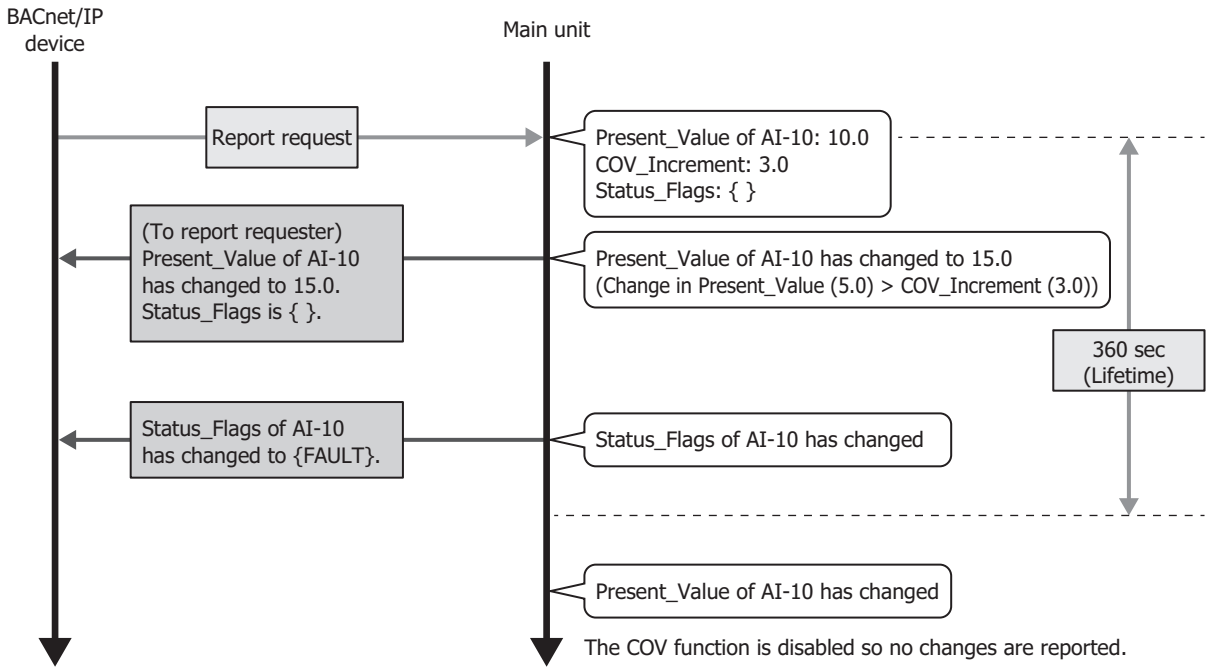
The main unit implements the COV function by sending the Confirmed COV Notification service/Unconfirmed COV Notification service to the report requesting BACnet/IP device according to the parameters contained in the report request when that report request (Subscribe COV service) was transmitted from the BACnet/IP device.

The key parameters contained in the report request are as follows.

Parameters	Description
Monitored Object Identifier	Type and ID of object for which to enable the COV function.
Issue Confirmed Notifications	Selects whether or not to confirm messages that will be sent from the main unit to the BACnet/IP device. • With confirmation (ConfirmedCOVNotification) • Without confirmation (UnconfirmedCOVNotification)
Lifetime	The time to enable the COV function in 1 sec increments. If 0 or omitted, the COV function is enabled with no time limit.

The following diagram shows when the main unit has received a report request transmitted from a BACnet/IP device like that in the following table.

Report Request Parameters	Description
Monitored Object Identifier	Analog Input object (instance number: 10)
Issue Confirmed Notifications	With confirmation (ConfirmedCOVNotification)
Lifetime	360 sec



- The maximum number of COVs that can be registered is 256.
- For details on the parameters in a response request, refer to "BACnet Building Automation Data Communication Protocol", a book published by the Institute of Electrical Installation Engineers of Japan.

Unsubscribed COV (COVU) function

In the COVU (Change Of Value Unsubscribed) function, the main unit spontaneously provides notification that Present_Value or Status_Flags of a specific object has changed to all BACnet/IP devices connected to the same network.

The properties of objects that correspond to the COVU function and the timing of when to provide notification of changes are given as follows.

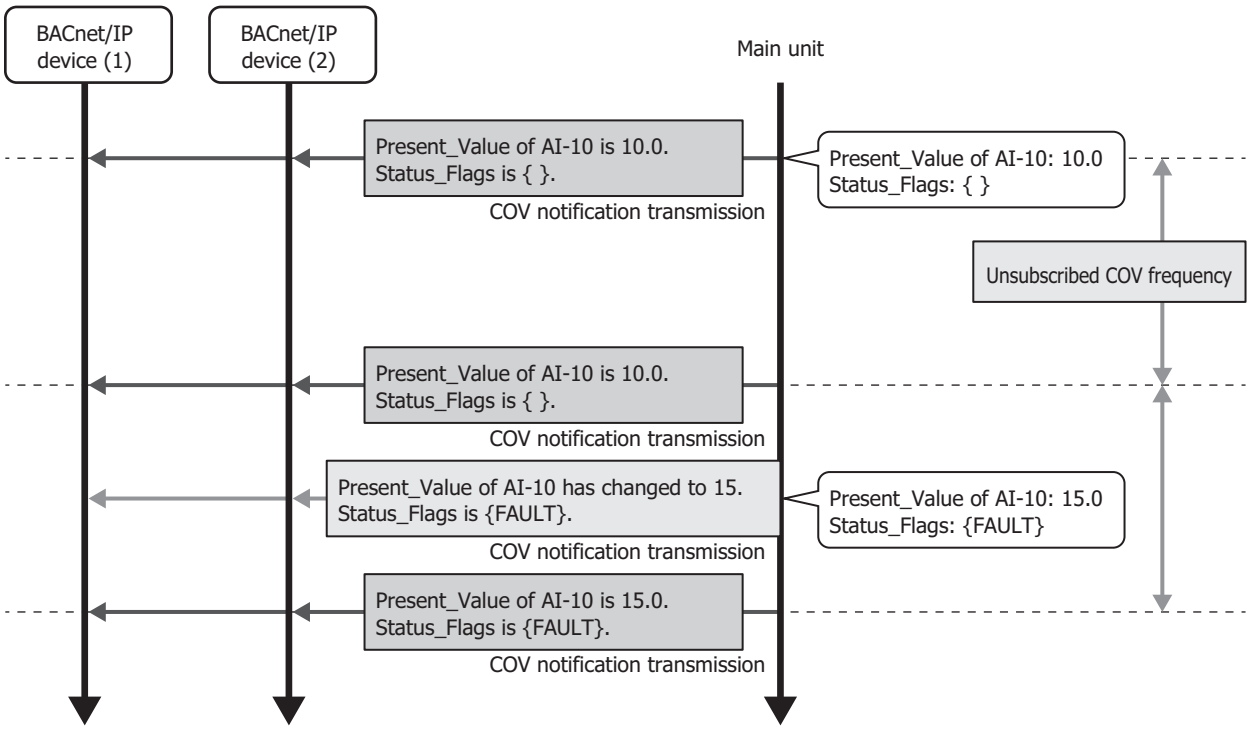
Object	Properties	Timing to Provide Notification of Change
Analog Input Analog Output Analog Value	Present_Value Status_Flags	Notification of the status is provided at the interval of the Unsubscribed COV frequency*1. Notification of change is also provided when (1) or (2) as follows. (1) When Present_Value has changed to a value greater than or equal to the value set for COV_Increment (starting point is from when the previous COV notification was sent) (2) When any of the bits in Status_Flags have changed
Binary Input Binary Output Binary Value	Present_Value Status_Flags	Notification of change is provided when (1) or (2) as follows. (1) When Present_Value has changed (2) When any of the bits in Status_Flags have changed

*1 See "BACnet/IP Settings" on page 1-13.

The main unit implements the COVU function by sending the Unconfirmed COV Notification service to BACnet/IP devices.

For example, when BACnet/IP is started with AI-10 (Analog Input object, instance number 10) having the status in the following table, notification is provided on the status of the properties at the set cycle (Unsubscribed COV frequency).

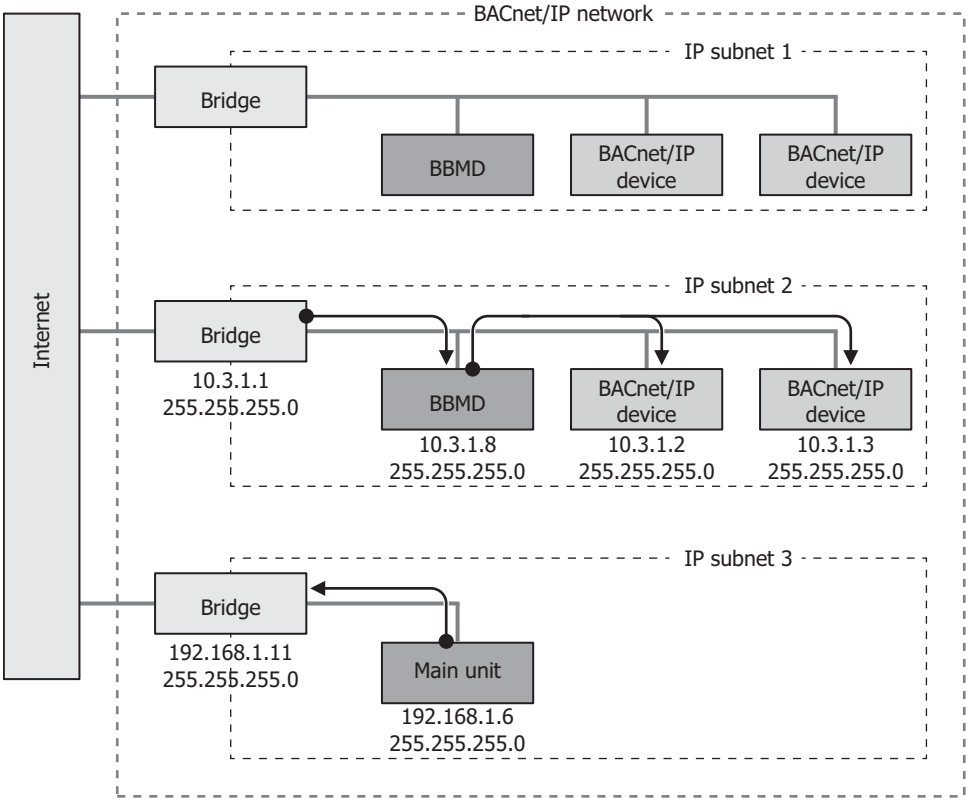
AI-10 Properties	Value
Present_Value	10.0
COV_Increment	3.0
Status_Flags	{ }



- When using the COVU function, the values of properties can be regularly sent to BACnet/IP devices, regardless of any change in the value of the properties.
- The COVU function can be set for each object.
- Only one Unsubscribed COV Frequency can be set for the main unit. Different frequencies cannot be set for each object.
- If the Unsubscribed COV Frequency is set to 0, the COV function is stopped, and notification is provided only when there is a change in the target property.
- When the Unsubscribed COV frequency is changed by a device address, the change will be applied after the service is next initiated.
- Notification may not be provided for changes in values that are faster than the synchronization cycle of objects and device addresses.
- The COV notification may not be sent when concurrent transmission requests exceed the transmission queue limit.

Foreign device function

When the BACnet/IP network is built with multiple IP subnets, install one BBMD (BACnet Broadcast Management Device) on each IP subnet. A BBMD is a device that transfers broadcast communications of BACnet/IP devices to different IP subnets. BACnet/IP devices perform broadcast communications with BACnet/IP devices on different IP subnets via BBMDs. The Foreign Device function is used for performing broadcast communications with BACnet/IP devices on different IP subnets even when there is no BBMD on the IP subnet of the main unit. Broadcast communications can be performed with a BACnet/IP device on the BACnet/IP network by registering the main unit as a foreign device for a specified BBMD.



BACnet/IP Operation

BACnet/IP is enabled and disabled according to the value of BACnet/IP Communication Bit (M8450/LSM70).

Properties with allocated device addresses work by referencing those device addresses.

When BACnet/IP is enabled and properties or the values of device addresses that are associated with properties are changed, they operate with the values after the change.

BACnet/IP	Properties with Allocated Device Addresses and the Values of Those Device Addresses
Enabled	Bound
Disabled	Not bound

Ladder Program

FC6A Plus FC6A All-in-One FT2J/1J HG2J/1J HG5G/4G/3G/2G-V

Ladder Program Operation during RUN and STOP

BACnet/IP communication is enabled and disabled according to BACnet Communication Bit (M8450/LSM70), regardless of the RUN/STOP status.

Properties with allocated devices work by referencing those devices.

When BACnet/IP communication is enabled and properties or the values of devices that are associated with properties are changed, even during STOP, they operate with the values after the change.

Output during STOP is performed according to the status of Maintain Outputs While CPU Stopped (M8025), regardless of whether or not BACnet/IP is enabled or disabled. For details on Maintain Outputs While CPU Stopped (M8025), see the following manuals.
FC6A: Chapter 6 "Special Internal Relay" in the "FC6A Series MICROSmart User's Manual".
FT2J/1J: Chapter 35 "1.2 Control Device Addresses" in the "WindO/I-NV4 User's Manual".

Operation During STOP→RUN

Properties with allocated devices work by referencing those devices.

When the power supply is turned on and off or the main unit is switched between STOP and RUN and the values of devices associated with properties are changed, such as due to changing device values with the ladder program, the devices operate with the values after the change. Use caution because the main unit may operate unexpectedly for this reason.

For example, when M0000 is allocated to Out_Of_Service for FC6A, the value of Out_Of_Service may differ by the memory backup setting of the device.

(1) When internal relays are set to "Clear All" by STOP→RUN

RUN/STOP Status	BACnet/IP Enabled/ Disabled	Out_Of_Service	M0000
STOP	Enabled	TRUE	ON
↓			
RUN	Enabled	**FALSE**	**OFF**

At the start of RUN, Out_Of_Service=FALSE because the status of internal relays is cleared.

(2) When internal relays are set to "Keep All" by STOP→RUN

RUN/STOP Status	BACnet/IP Enabled/ Disabled	Out_Of_Service	M0000
STOP	Enabled	TRUE	ON
↓			
RUN	Enabled	**TRUE**	**ON**

Out_Of_Service=TRUE because the status of internal relays is kept.

- Set devices that are allocated to properties so that their statuses are kept at STOP→RUN.
- The default settings of memory backup for internal relays and data registers are as follows.

Device	Symbol	Default Setting
Internal relay	M	Clear All
Data register	D	Keep All

For the memory backup settings of devices, see the following manuals.
FC6A: Chapter 5 "Functions and Settings" in the "FC6A Series MICROSmart User's Manual".
FT2J/1J: Chapter 28 "2.4 Memory Backup" in the "WindO/I-NV4 User's Manual".

Special Devices

The special devices used in BACnet/IP communication differ depending on the main unit type.

Internal Device Name		FC6A	FT2J/1J	HG2J/1J	HG5G/4G/3G/2G-V
Special Internal Relay		Yes	No	No	No
Special Data Register		Yes	No	No	No
HMI Device	HMI Special Internal Relay	No	Yes	Yes	Yes
	HMI Special Data Register	No	Yes	Yes	Yes

R/W is an abbreviation for read/write.
The notation for the R/W field is as follows.
R/W: The device can be both read from and written to
R: Read-only
W: Write-only

■Special Internal Relay, HMI Special Internal Relay

Special Internal Relay	HMI Special Internal Relay	Description		R/W
-	LSM69	BACnet/IP Communication Interface	Configure the interface for BACnet/IP communication. 0: LAN 1: WLAN	R/W
M8450	LSM70	BACnet/IP Communication Bit	The special internal relay permits or prohibits BACnet/IP communication. OFF: BACnet/IP communication prohibited ON: BACnet/IP communication permitted	R/W

When the BACnet Operation Status (D8782/LSD260) is stopped due to an error, follow the procedure below to resume BACnet/IP communication.
1. Turn off the BACnet/IP Communication Bit (M8450/LSM70) once.
   Wait until the BACnet/IP Operation Status (D8782/LSD260) becomes stopped.
2. Turn on the BACnet/IP Communication Bit (M8450/LSM70).

The BACnet/IP Communication Interface (LSM69) must be configured before allowing BACnet/IP communication.

■Special Data Register, HMI Special Data Register

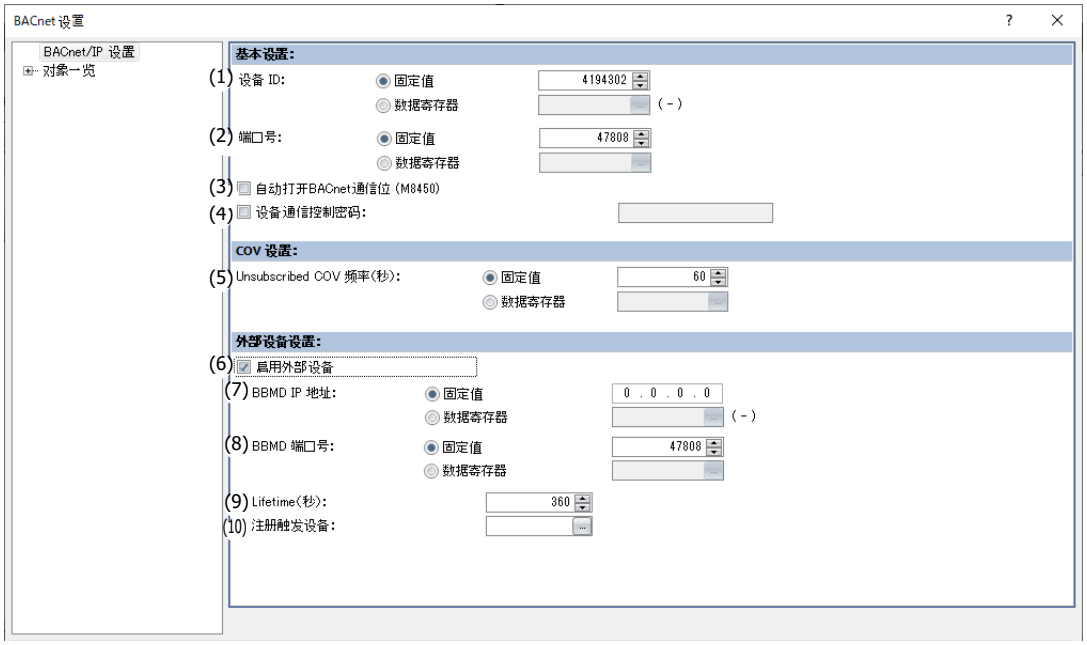
Special Data Register	HMI Special Data Register	Description		R/W
D8782	LSD260	BACnet/IP Operation Status	This special data register stores the operation status of BACnet/IP communication. 0: Stopped 1: Ready 2: Operating 3: Stopped by error	R
D8783	LSD261	BACnet/IP Error Information	This special data register stores error information that has occurred in BACnet/IP communication. Information for the error that last occurred is stored. 0: Normal 1: Invalid device ID 2: Invalid IP address 3: Invalid BBMD IP address 4: BBDM registration failure	R

BACnet Settings Dialog Box

This section describes items on the **BACnet Settings** dialog box.

BACnet/IP Settings

This section describes operation procedures and details about items for performing communication using BACnet/IP.



■Basic Settings

(1) Device ID

The assigned ID used to identify BACnet/IP devices on the BACnet/IP network. The device ID for the main unit can be configured to one of the following two types.

Setting Method	Description
Fixed Value	Specifies the Device ID as a constant. Sets a fixed value in the range from 1 to 4,194,302. The default value depends on the main unit type.*1
Data Register	Specifies the Device ID as a value of word device*2. This option uses 2 words of address numbers starting from the specified device address. Set the first device address so that the device range is not exceeded. Set the device ID in the range from 1 to 4,194,302.

*1 The default value for the device ID is as follows.

Main Unit Type	Device ID
FC6A	4194302
FT2J/1J	4194301
HG2J/1J	
HG5G/4G/3G/2G-V	

*2 The word device that can be specified depends on the main unit type.

Internal Device Name		Symbol	FC6A	FT2J/1J	HG2J/1J	HG5G/4G/3G/2G-V
Data Register		D	Yes	No	No	No
HMI Devices	HMI Data Register	LDR	No	Yes	Yes	No
	HMI Keep Register	LKR	No	Yes	Yes	No
	HMI Temporary Register	LBR	No	Yes	Yes	No
Control Devices	Data Register	D	No	Yes	No	No
		#D	No	No	No	Yes

For details, see the following manuals.
FC6A: Chapter 6 "Devices" in the "FC6A Series MICROSmart User's Manual".
FT2J/1J, HG2J/1J, HG5G/4G/3G/2G-V: Chapter 35 "2 Word Devices" in the "WindO/I-NV4 User's Manual".

(2) Port Number

Configures the port number for performing BACnet/IP Communication. Configures the port number in the range from 0 to 65,535.

Setting Method	Description
Fixed Value	Specifies the port number as a constant. (Default: 47808 (BAC0h)).
Data Register	Specifies the port number as a value of word device*1.

*1 The word device that can be set depends on the main unit type.

Internal Device Name		Symbol	FC6A	FT2J/1J	HG2J/1J	HG5G/4G/3G/2G-V
Data Register		D	Yes	No	No	No
HMI Devices	HMI Data Register	LDR	No	Yes	Yes	No
	HMI Keep Register	LKR	No	Yes	Yes	No
	HMI Temporary Register	LBR	No	Yes	Yes	No
Control Devices	Data Register	D	No	Yes	No	No
		#D	No	No	No	Yes

For details, see the following manuals.
FC6A: Chapter 6 "Devices" in the "FC6A Series MICROSmart User's Manual".
FT2J/1J, HG2J/1J, HG5G/4G/3G/2G-V: Chapter 35 "2 Word Devices" in the "WindO/I-NV4 User's Manual".

(3) Turn ON BACnet Communication bit (M8450/LSM70) automatically

Configures whether or not to turn on BACnet Communication Bit (M8450/LSM70) in the END processing of the first scan when starting operation.
The following table shows the statuses.

BACnet Communication Bit (M8450/LSM70) Status	RUN/STOP Status*1	BACnet/IP	Present_Value*2 and Device Value	Overridden Flag in Status_Flags*3
ON	STOP	Enabled	Bound	FALSE
	RUN			
OFF	STOP	Disabled	Not bound	—
	RUN			

*1 FC6A and FT2J/1J only
*2 For Present_Value, see "Objects" on page 1-34 and "Present_Value" on page 1-41.
*3 For Status_Flags, see "Objects" on page 1-34 and "Status_Flags" on page 1-54.

- During Online Edit of the FC6A, BACnet Communication Bit (M8450) is turned on in the END processing of the first scan in which downloading the user program has completed.
- When the following operations are performed, the value of the BACnet Communication Bit (M8450/LSM70) is turned off.
  - Downloads the user program or project data.
  - Turns off the main unit.
  - Switches to System Mode (LSM70 only)

(4) Device Communication Control Password

Configures the password to request when the Device Communication Control service is received from a BACnet/IP device. The maximum number is 20 characters. When the Device Communication Control service is received, the main unit does not initiate a service or provide a response for the specified period of time.

■COV Settings

(5) Unsubscribed COV Frequency (sec)

Configures the cycle to regularly provide notifications of property values with the COVU function as 0 to 65,535 (seconds). If 0 is set, the COVU function is stopped, and notification is provided only when there is a change in the target property.

Setting Method	Description
Fixed Value	Specifies the interval as a constant. (Default: 60 sec)
Data Register	Specifies the interval as a value of word device*1. The frequency can be changed during RUN.

*1 The word device that can be set depends on the main unit type.

Internal Device Name		Symbol	FC6A	FT2J/1J	HG2J/1J	HG5G/4G/3G/2G-V
Data Register		D	Yes	No	No	No
HMI Devices	HMI Data Register	LDR	No	Yes	Yes	No
	HMI Keep Register	LKR	No	Yes	Yes	No
	HMI Temporary Register	LBR	No	Yes	Yes	No
Control Devices	Data Register	D	No	Yes	No	No
		#D	No	No	No	Yes

For details, see the following manuals.
FC6A: Chapter 6 "Devices" in the "FC6A Series MICROSmart User's Manual".
FT2J/1J, HG2J/1J, HG5G/4G/3G/2G-V: Chapter 35 "2 Word Devices" in the "WindO/I-NV4 User's Manual".

- Only one Unsubscribed COV frequency can be set for the main unit, and different frequencies cannot be set per object.
- Whether or not to use the COVU function can be selected per object. The COVU function can also be enabled and disabled by specifying a data register and changing its value.
- When changing the value of the data register, the Unsubscribed COV frequency after it has been changed will be applied after the service is next initiated.

■Foreign Device Settings

(6) Enable Foreign Device

Configures whether or not to use the Foreign Device function. Using the Foreign Device function, the main unit registers itself as a Foreign Device on the configure BBMD.

(7) BBMD IP Address

To join a BACnet/IP network when there is no BBMD on the IP subnet of the main unit, configures the IP address of the BBMD on the other IP subnet.

Setting Method	Description
Fixed Value	Specifies the IP address as a constant.
Data Register	Specifies the IP address as a value of word device*1. This option uses 4 words of address numbers starting from the specified device address. Example: Configures 192.168.2.5 to the IP address of the BBMD. The address number of top device address+0 = 192 The address number of top device address+1 = 168 The address number of top device address+2 = 2 The address number of top device address+3 = 5

*1 The word device that can be set depends on the main unit type.

Internal Device Name		Symbol	FC6A	FT2J/1J	HG2J/1J	HG5G/4G/3G/2G-V
Data Register		D	Yes	No	No	No
HMI Devices	HMI Data Register	LDR	No	Yes	Yes	No
	HMI Keep Register	LKR	No	Yes	Yes	No
	HMI Temporary Register	LBR	No	Yes	Yes	No
Control Devices	Data Register	D	No	Yes	No	No
		#D	No	No	No	Yes

For details, see the following manuals.
FC6A: Chapter 6 "Devices" in the "FC6A Series MICROSmart User's Manual".
FT2J/1J, HG2J/1J, HG5G/4G/3G/2G-V: Chapter 35 "2 Word Devices" in the "WindO/I-NV4 User's Manual".

(8) BBMD Port Number

To join a BACnet/IP network when there is no BBMD on the IP subnet of the main unit, configures the port number of BBMD on other IP subnet from 0 to 65,535.

Setting Method	Description
Fixed Value	Specifies the port number as a constant. (Default: 47808)
Data Register	Specifies the port number as a value of word device*1.

*1 The word device that can be set depends on the main unit type.

Internal Device Name		Symbol	FC6A	FT2J/1J	HG2J/1J	HG5G/4G/3G/2G-V
Data Register		D	Yes	No	No	No
HMI Devices	HMI Data Register	LDR	No	Yes	Yes	No
	HMI Keep Register	LKR	No	Yes	Yes	No
	HMI Temporary Register	LBR	No	Yes	Yes	No
Control Devices	Data Register	D	No	Yes	No	No
		#D	No	No	No	Yes

For details, see the following manuals.
FC6A: Chapter 6 "Devices" in the "FC6A Series MICROSmart User's Manual".
FT2J/1J, HG2J/1J, HG5G/4G/3G/2G-V: Chapter 35 "2 Word Devices" in the "WindO/I-NV4 User's Manual".

(9) Lifetime (sec)

Configures the interval of time to register the main unit with the BBMD as a Foreign Device in the range from 0 to 65,535 (seconds). (Default: 360 sec)
When (Lifetime + 30) sec has elapsed from registration, the registered content will be deleted from the BBMD.

(10) Registration Trigger Device

The device address used to register the main unit with the BBMD configure with **BBMD IP Address** above as a Foreign Device. The bit device that can be set depends on the main unit type.

Internal Device Name		Symbol	FC6A	FT2J/1J	HG2J/1J	HG5G/4G/3G/2G-V
Internal relay		M	Yes	No	No	No
HMI Devices	HMI Internal Relay	LM	No	Yes	Yes	No
	HMI Keep Relay	LK	No	Yes	Yes	No
	HMI Temporary Relay	LBM	No	Yes	Yes	No
Control Devices	Internal relay	M	No	Yes	No	No
		#M	No	No	No	Yes

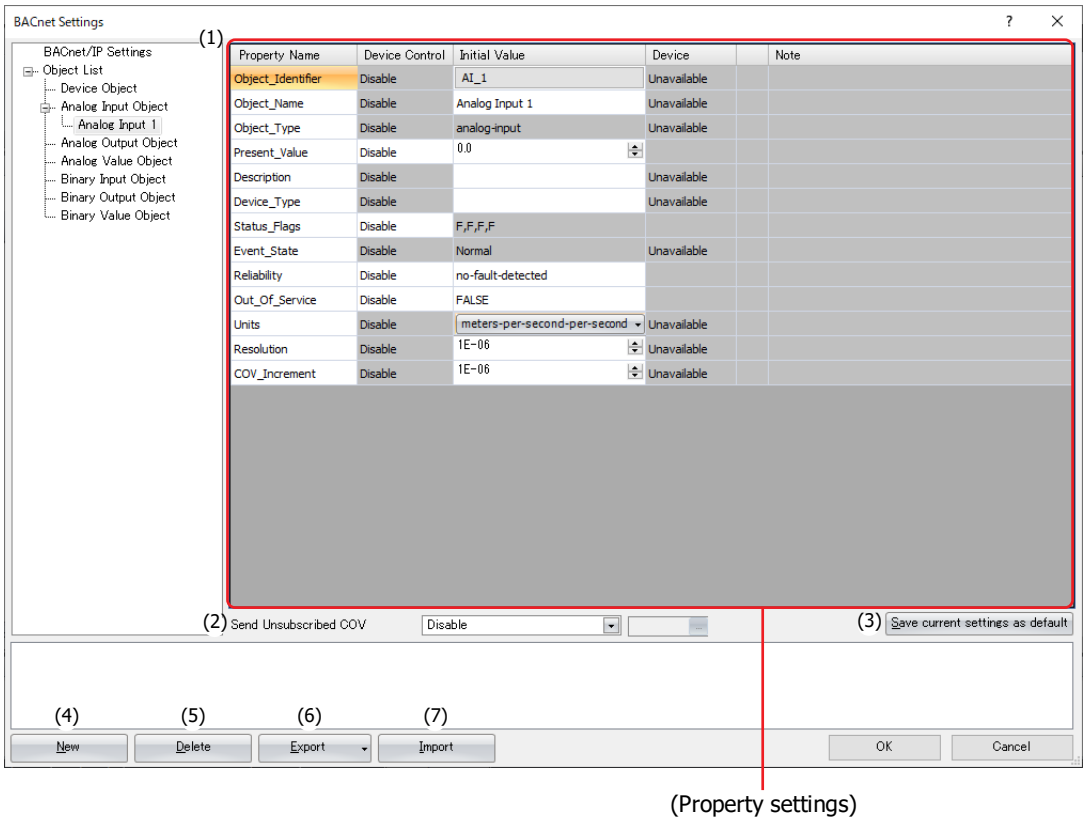
For details, see the following manuals.
FC6A: Chapter 6 "Devices" in the "FC6A Series MICROSmart User's Manual".
FT2J/1J, HG2J/1J, HG5G/4G/3G/2G-V: Chapter 35 "1 Bit Devices" in the "WindO/I-NV4 User's Manual".

When the value of the **Registration Trigger Device** is turned on from off, the main unit is registered with the BBMD. To continuously register the main unit with the BBMD, use the **Registration Trigger Device** to re-register the main unit before (Lifetime + 30) sec elapses.

Object List

Select an object node to display the list of registered objects.

For example, select Analog Input to display the list of Analog Input objects, and select the object list node to display the list of all objects. The properties can also be changed when the list is displayed.



(1) (Property settings)

This area displays the properties of the object ID selected in the object list.

A portion of the properties can be edited.

(2) Send Unsubscribed COV

Configures whether or not to send Unsubscribed COV.

Setting	Description
Enable	When the value of the BACnet Communication Bit (M8450/LSM70) is on, the COVU function is always enabled and the Unconfirmed COV Notification service is sent.
Disable	The COVU function is always disabled and the Unconfirmed COV Notification service is not sent.
Control by device	Toggles between enabled and disabled with the value of device address. Specify the bit device to use as the condition to switch between enabled and disabled. It is enabled when the value of device address turns on from off and it is disabled when the value of device address turns off from on.

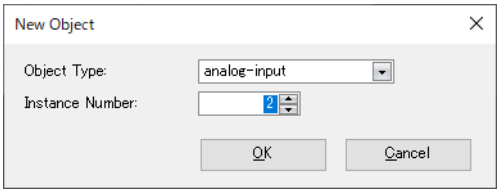
(3) Save current settings as default

Saves the settings of the properties as the default values of the object.

(4) New

Adds a new object.

Click **New** to display the **New Object** dialog box. Configures the object type and instance number (0 to 4,194,302). Ensures that objects of the same object type do not have the same instance number.



(5) Delete

Deletes the object under the node selected in the object list.

(6) Export

Exports the settings of the following objects as a CSV file.
Analog Input, Analog Output, Analog Value, Binary Input, Binary Output, Binary Value

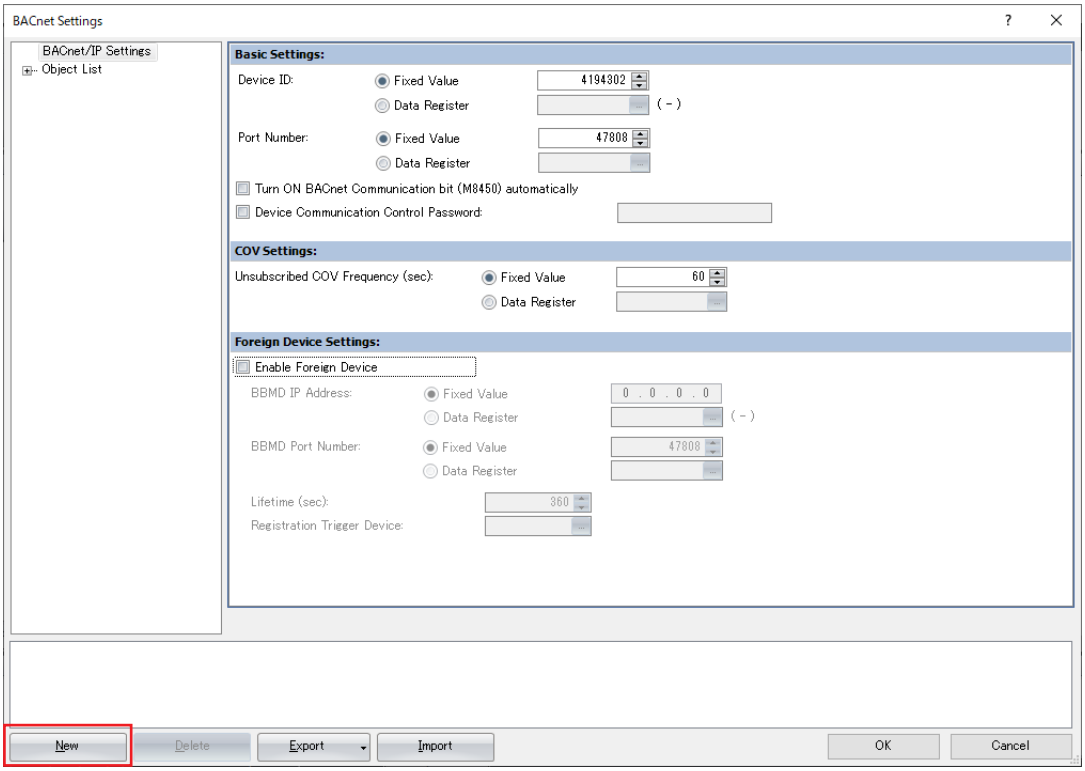
(7) Import

Imports a CSV file that was export and automatically creates objects.
Objects cannot be imported if the format of the CSV file is invalid or if the maximum number of objects has been reached.

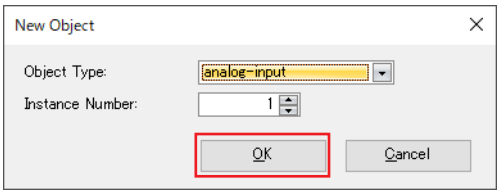
Adding a New Object

This section describes the procedure to add a new object.

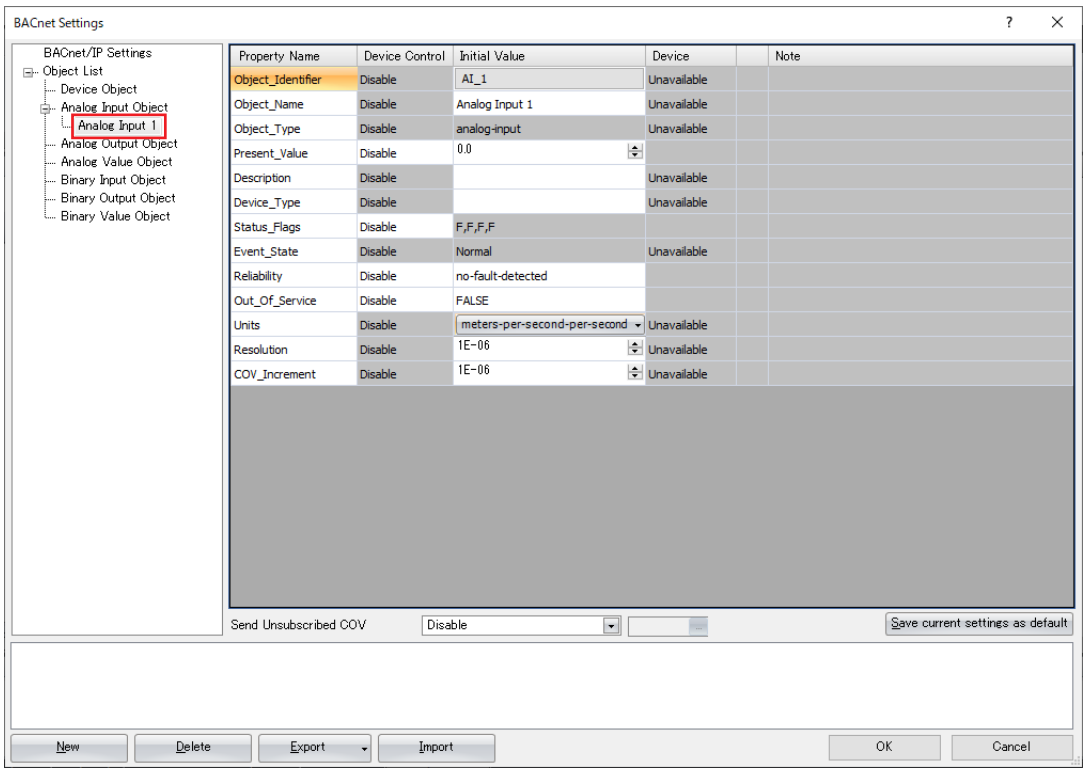
1. Click **New**.
   The **New Object** dialog box is displayed.



2. Select the object type of the object to register, and assign it an instance number.
   Click **OK**.



3. Click the registered object ID in the object list.
The properties are displayed.



4. Configure the properties and click **OK**.
This adds a new object.

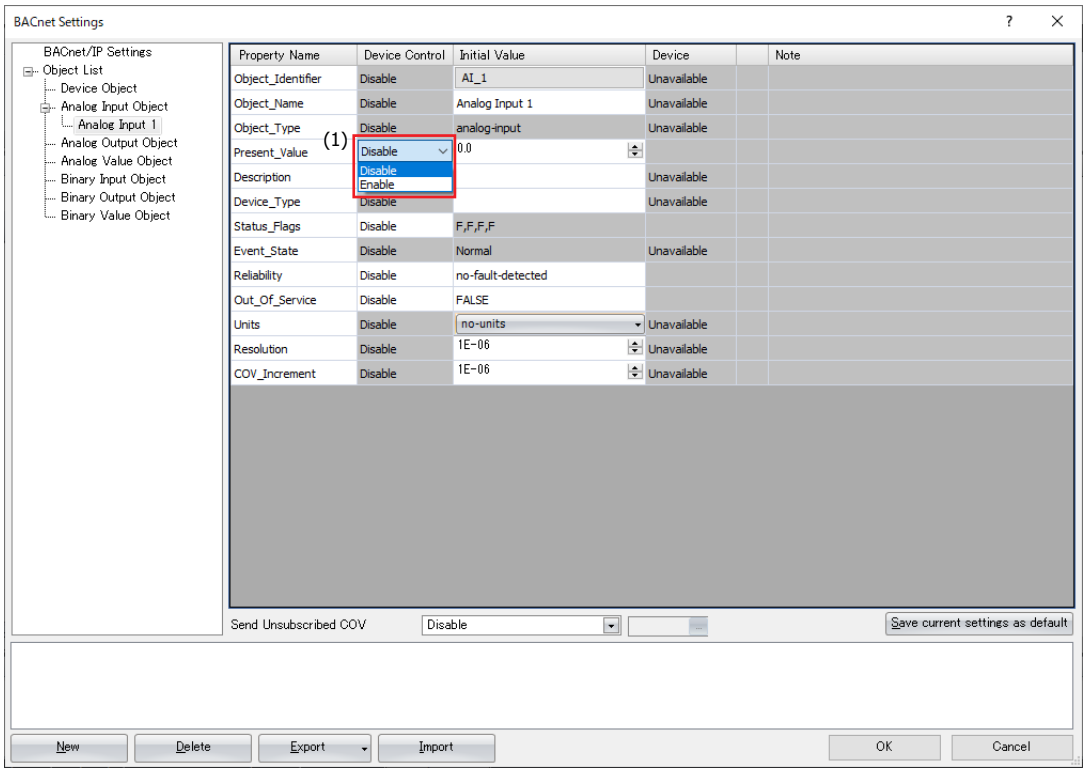
The maximum number of objects that can be registered is 256.

Present_Value Settings

This section describes how to configure Present_Value. Present_Value is a float numeric value.

Analog Input Object

Use these settings to configure fixed values and device addresses for Present_Value of the Analog Input object.



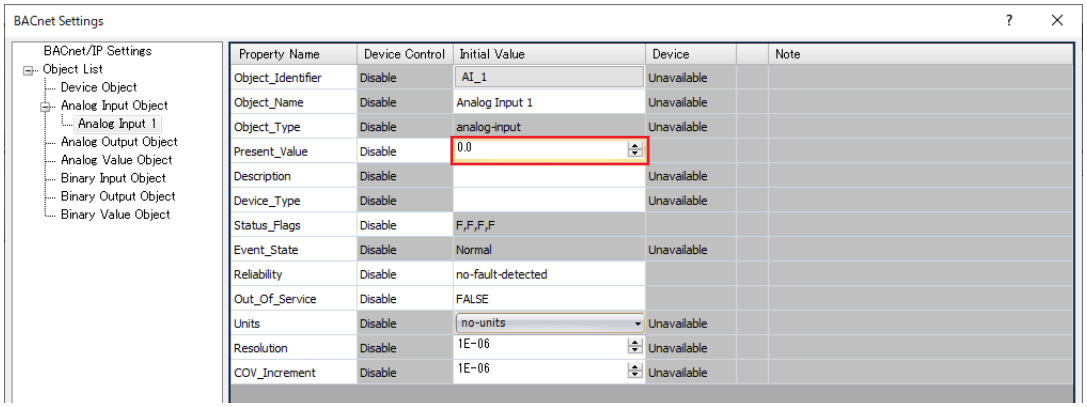
(1) Device Control

Selects whether to set a fixed value for Present_Value or to allocate device addresses.

Device Control	Description
Disable	Specifies the Present_Value as a constant.
Enable	Specifies the Present_Value as a value of word device.

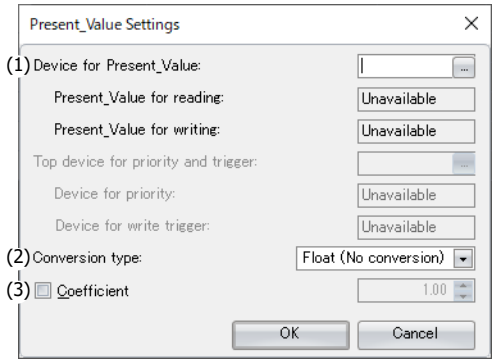
■Setting a Fixed Value for Present_Value

Select **Disable** in the **Device Control** of **Present_Value**, enter a constant value for **Initial Value**.



■**Allocating Device Addresses to Present_Value**

Select **Enable** in the **Device Control** of **Present_Value**, and then click a button displayed in **Device** to display the **Present_Value Settings** dialog box. Configure the parameters in the **Present_Value Settings** dialog box.



(1) Device for Present_Value

Configures the device address to store Present_Value.

The word device that can be set depends on the main unit type.

Internal Device Name		Symbol	FC6A	FT2J/1J	HG2J/1J	HG5G/4G/3G/2G-V
Data Register		D	Yes	No	No	No
HMI Devices	HMI Data Register	LDR	No	Yes	Yes	No
	HMI Keep Register	LKR	No	Yes	Yes	No
	HMI Temporary Register	LBR	No	Yes	Yes	No
Control Devices	Data Register	D	No	Yes	No	No
		#D	No	No	No	Yes

For details, see the following manuals.

FC6A: Chapter 6 "Devices" in the "FC6A Series MICROSmart User's Manual".

FT2J/1J, HG2J/1J, HG5G/4G/3G/2G-V: Chapter 35 "2 Word Devices" in the "WindO/I-NV4 User's Manual".

According to **Conversion type**, 1 or 2 continuous words of address number are used starting from the specified address number of Device Address.

Device for Present_Value	Conversion type		Storage Destination
	FC6A	FT2J/1J, HG2J/1J, HG5G/4G/3G/2G-V	
Present_Value for writing	Word (W) Integer (I)	UBIN16(W) BIN16(I)	The address number of Top Device Address+0
	Double (D) Long (L) Float (F)	UBIN32(D) BIN32(L) Float32(F)	The address number of Top Device Address+0, The address number of Top Device Address+1

(2) Conversion type

Configures the data type of the device address to which Present_Value (float) is allocated. For details, see Present_Value in "Analog Input Object" on page 1-45.

(3) Coefficient

Configures Present_Value as the product of the value stored in the device address multiplied by the coefficient.

Present_Value = Value stored in the device address × Coefficient

Example: FC6A

Device Address	Data Type	Value	Coefficient	Present_Value
D0000	Word (W)	1,000	0.01	10.0
D0000, D0001	Float (F)	2.5	0.5	1.25

The coefficient is multiplied as a float-type value. The conversion order is as follows.

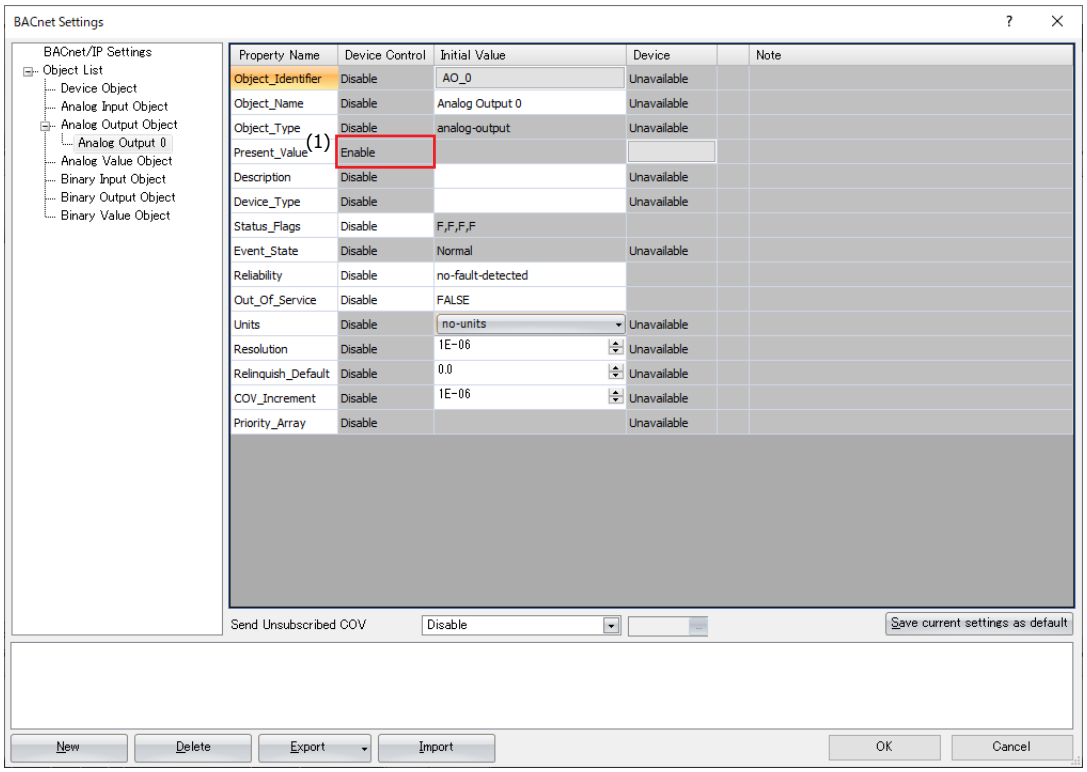
Device Addresses→Present_Value

(1) The value of the device addresses is converted to a float-type value.

(2) The coefficient is multiplied by the conversion result in (1).

Analog Output Object

Use these settings to configure devices for Present_Value of the Analog Output object.

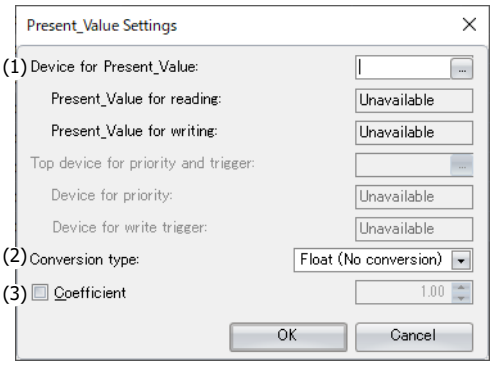


(1) Device Control

Device Control of **Present_Value** is **Enable**. A constant value cannot be configured for **Present_Value** of the Analog Output object.

■**Allocating Device Addresses to Present_Value**

Configure the parameters in the **Present_Value Settings** dialog box.



(1) Device for Present_Value

Configures the device address to store Present_Value.

The word device that can be set depends on the main unit type.

Internal Device Name		Symbol	FC6A	FT2J/1J	HG2J/1J	HG5G/4G/3G/2G-V
Data Register		D	Yes	No	No	No
HMI Devices	HMI Data Register	LDR	No	Yes	Yes	No
	HMI Keep Register	LKR	No	Yes	Yes	No
	HMI Temporary Register	LBR	No	Yes	Yes	No
Control Devices	Data Register	D	No	Yes	No	No
		#D	No	No	No	Yes

For details, see the following manuals.

FC6A: Chapter 6 "Devices" in the "FC6A Series MICROSmart User's Manual".

FT2J/1J, HG2J/1J, HG5G/4G/3G/2G-V: Chapter 35 "2 Word Devices" in the "WindO/I-NV4 User's Manual".

According to **Conversion type**, 1 or 2 continuous words of address number are used starting from the specified address number of Device Address.

Device for Present_Value	Conversion type		Storage Destination
	FC6A	FT2J/1J, HG2J/1J, HG5G/4G/3G/2G-V	
Present_Value for reading	Word (W) Integer (I)	UBIN16(W) BIN16(I)	The address number of Top Device Address+0
	Double (D) Long (L) Float (F)	UBIN32(D) BIN32(L) Float32(F)	The address number of Top Device Address+0, The address number of Top Device Address+1

(2) Conversion type

Configures the data type of the device addresses to which Present_Value (float) is allocated. For details, see Present_Value in "Analog Output Object" on page 1-46.

(3) Coefficient

The product of Present_Value multiplied by 1/coefficient multiplied is stored in the device addresses.

Value of device addresses = Present_Value × (1/coefficient)

Example: FC6A

Present_Value	Coefficient	Data Register	Data Type	Value
10.0	0.01	D0000	Word (W)	1,000
1.25	0.5	D0000, D0001	Float (F)	2.5

The coefficient is multiplied as a float-type value. The conversion order is as follows.

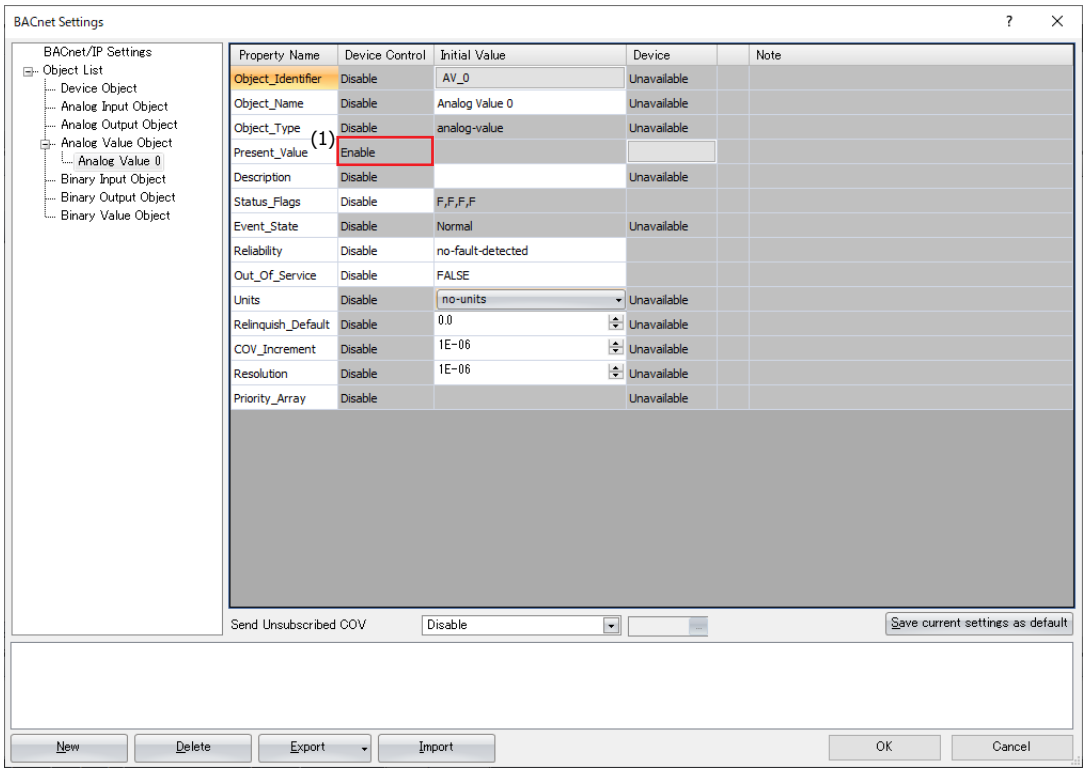
Present_Value→Device Addresses

(1) Present_Value is multiplied by (1/coefficient).

(2) The data type of the result of (1) (float-type) is converted.

Analog Value Object

Use these settings to configure devices for Present_Value of the Analog Value object.

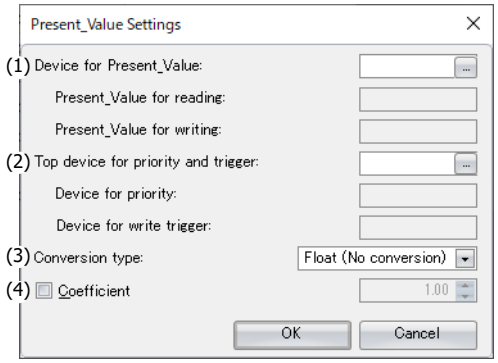


(1) Device Control

Device Control of **Present_Value** is **Enable**. A constant value cannot be configured for **Present_Value** of the Analog Value object.

■Allocating Device Addresses to Present_Value

Configure the parameters in the **Present_Value Settings** dialog box.



(1) Device for Present_Value

Configures the device address for reading Present_Value and the device address for writing Present_Value.
The word device that can be set depends on the main unit type.

Internal Device Name		Symbol	FC6A	FT2J/1J	HG2J/1J	HG5G/4G/3G/2G-V
Data Register		D	Yes	No	No	No
HMI Devices	HMI Data Register	LDR	No	Yes	Yes	No
	HMI Keep Register	LKR	No	Yes	Yes	No
	HMI Temporary Register	LBR	No	Yes	Yes	No
Control Devices	Data Register	D	No	Yes	No	No
		#D	No	No	No	Yes

For details, see the following manuals.
FC6A: Chapter 6 "Devices" in the "FC6A Series MICROSmart User's Manual".
FT2J/1J, HG2J/1J, HG5G/4G/3G/2G-V: Chapter 35 "2 Word Devices" in the "WindO/I-NV4 User's Manual".

The device address for reading Present_Value and writing Present_Value are automatically assigned according to the configured device address and **Conversion type**. Starting from the specified address number of Device Address, 2 or 4 continuous words of address number are used.

Device for Present_Value	Conversion type		Storage Destination
	FC6A	FT2J/1J, HG2J/1J, HG5G/4G/3G/2G-V	
Present_Value for reading	Word (W) Integer (I)	UBIN16(W) BIN16(I)	The address number of Top Device Address+0
	Double (D) Long (L) Float (F)	UBIN32(D) BIN32(L) Float32(F)	The address number of Top Device Address+0, The address number of Top Device Address+1
Present_Value for writing	Word (W) Integer (I)	UBIN16(W) BIN16(I)	The address number of Top Device Address+0
	Double (D) Long (L) Float (F)	UBIN32(D) BIN32(L) Float32(F)	The address number of Top Device Address+0, The address number of Top Device Address+1

(2) Top device for priority and trigger

Use this device address when writing the value of the device addresses to Present_Value. For details, refer to Present_Value in "Analog Value Object" on page 1-47.

Device for priority and **Device for write trigger** are automatically assigned when the device address is set. Starting from the specified address number of device address, 2 continuous words of address number are used.

Main unit	Storage Destination	Description
Device for priority	The address number of Top Device Address+0	Bit 15: 0: the value of the Present_Value for writing. 1: NULL(00h). Bits 14 to 5: Disabled Bits 4 to 0: Priority*1
Device for write trigger	The address number of Top Device Address+1	When the Device for write trigger turns on from off, writes a value to Priority_Array of index number indicated by the priority (bit 4 to 0 of Device for priority). The value to be written varies based on the value of bit 15 of Device for priority.

*1 Specify the priority between 1 and 16. If the priority is out of range, nothing is executed even when the Device for write trigger turns on from off.

The word device that can be set depends on the main unit type.

Internal Device Name		Symbol	FC6A	FT2J/1J	HG2J/1J	HG5G/4G/3G/2G-V
Data Register		D	Yes	No	No	No
HMI Devices	HMI Data Register	LDR	No	Yes	Yes	No
	HMI Keep Register	LKR	No	Yes	Yes	No
	HMI Temporary Register	LBR	No	Yes	Yes	No
Control Devices	Data Register	D	No	Yes	No	No
		#D	No	No	No	Yes

For details, see the following manuals.

FC6A: Chapter 6 "Devices" in the "FC6A Series MICROSmart User's Manual".

FT2J/1J, HG2J/1J, HG5G/4G/3G/2G-V: Chapter 35 "2 Word Devices" in the "WindO/I-NV4 User's Manual".

(3) Conversion type

Configures the data type of the device address to which Present_Value is allocated. For details, see Present_Value in "Analog Value Object" on page 1-47.

(4) Coefficient

The product of Present_Value multiplied by 1/coefficient is stored in the device addresses allocated as Present_Value for reading.

Present_Value for reading = Present_Value × (1/coefficient)

When the device for write trigger is turned on from off, the product of multiplying the value of the device addresses allocated as Present_Value for writing by the coefficient is set as Present_Value.

Present_Value = Present_Value for writing × Coefficient

Example: FC6A

Present_Value	Coefficient	Conversion Type	Device Addresses Allocated as Present_Value for Reading or Writing	
			Device Addresses	Value
10.0	0.01	Word (W)	D0000	1,000
1.25	0.5	Float (F)	D0002, D0003	2.5

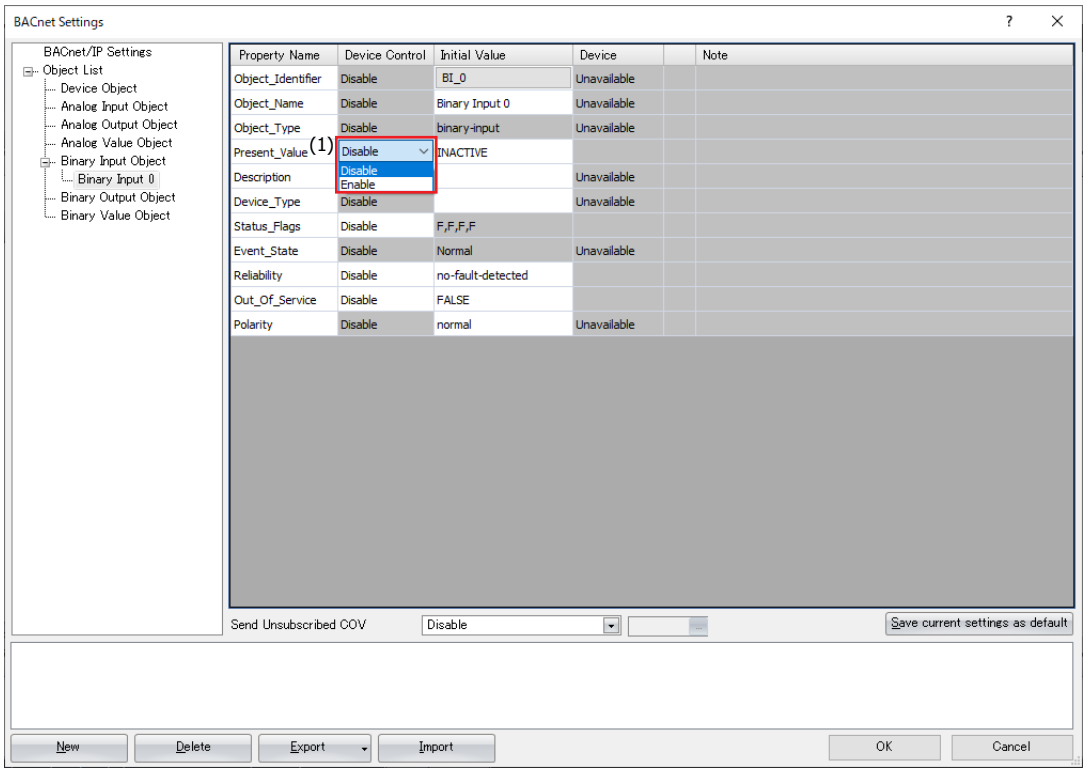
The coefficient is multiplied as a float-type value. The conversion order is as follows.

Data registers→Present_Value
(1) The value of the data registers is converted to a float-type value.
(2) The coefficient is multiplied by the conversion result in (1).

Present_Value→Data registers
(1) Present_Value is multiplied by (1/coefficient).
(2) The data type of the result of (1) (float) is converted.

Binary Input Object

Use these settings to configure fixed values and device addresses for Present_Value of the Binary Input object.



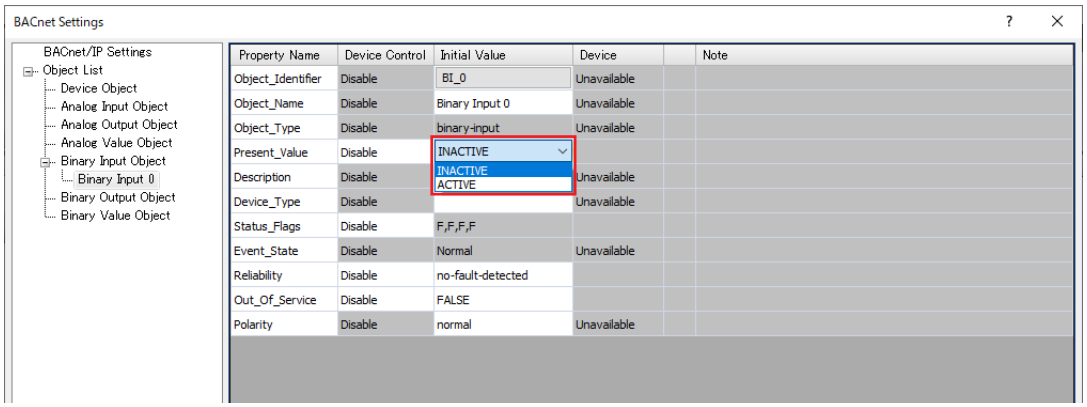
(1) Device Control

Selects whether to set a fixed value for Present_Value or to allocate device addresses.

Device Control	Description
Disable	Specifies the Present_Value as a constant.
Enable	Specifies the Present_Value as a value of word device.

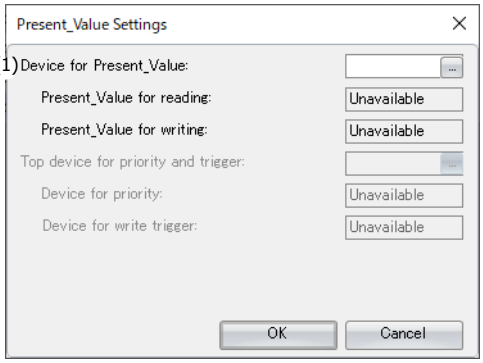
■Configures a fixed value to Present_Value

Select **Disable** in the **Device Control** of **Present_Value**, and then select **INACTIVE** or **ACTIVE** for **Initial Value**.



■**Allocating Device Addresses to Present_Value**

Select **Enable** in the **Device Control** of **Present_Value**, and then click a button displayed in **Device** to display the **Present_Value Settings** dialog box. Configure the parameters in the **Present_Value Settings** dialog box.



(1) Device for Present_Value

Configures the device address to store Present_Value.

The bit device that can be set depends on the main unit type.

Internal Device Name		Symbol	FC6A	FT2J/1J	HG2J/1J	HG5G/4G/3G/2G-V
Internal relay		M	Yes	No	No	No
HMI Devices	HMI Internal Relay	LM	No	Yes	Yes	No
	HMI Keep Relay	LK	No	Yes	Yes	No
	HMI Temporary Relay	LBM	No	Yes	Yes	No
Control Devices	Internal relay	M	No	Yes	No	No
		#M	No	No	No	Yes

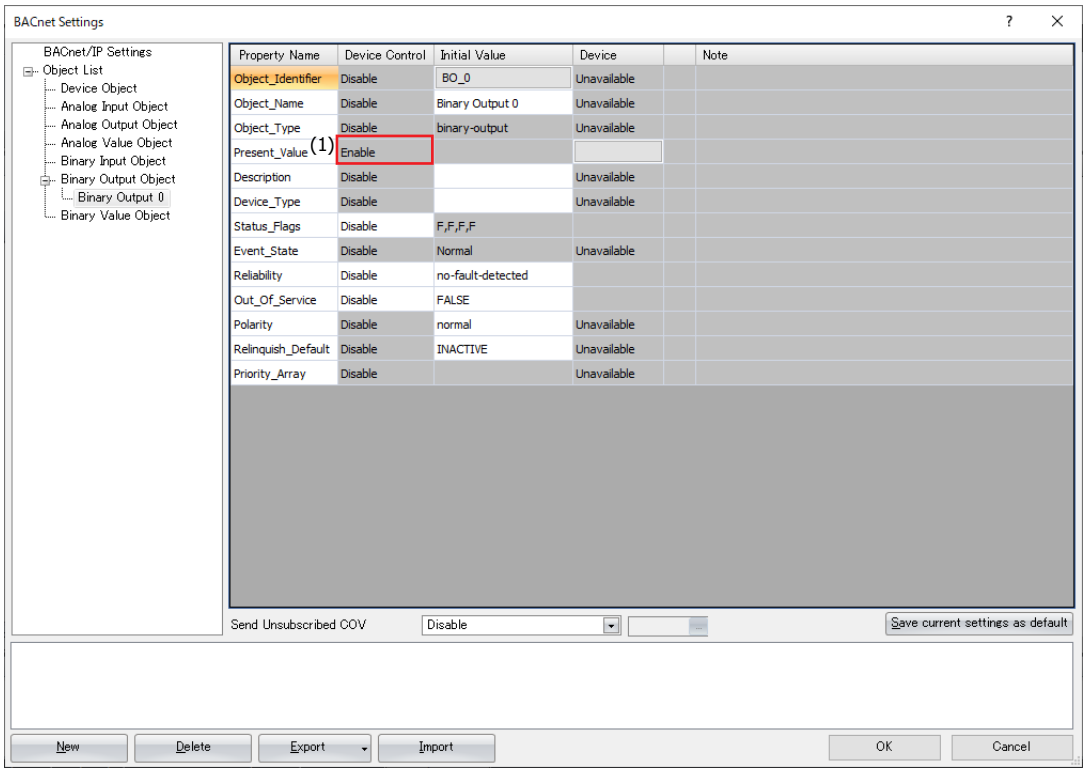
For details, see the following manuals.

FC6A: Chapter 6 "Devices" in the "FC6A Series MICROSmart User's Manual".

FT2J/1J, HG2J/1J, HG5G/4G/3G/2G-V: Chapter 35 "1 Bit Devices" in the "WindO/I-NV4 User's Manual".

Binary Output Object

Use these settings to configure device addresses for Present_Value of the Binary Output object.

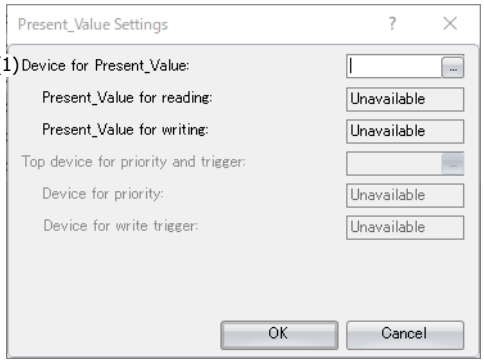


(1) Device Control

Device Control of **Present_Value** is **Enable**. A constant value cannot be configured for **Present_Value** of the Binary Output object.

■**Allocating Device Addresses to Present_Value**

Configure the parameters in the **Present_Value Settings** dialog box.



(1) Device for Present_Value

Configures the device address to store Present_Value.

The bit device that can be set depends on the main unit type.

Internal Device Name		Symbol	FC6A	FT2J/1J	HG2J/1J	HG5G/4G/3G/2G-V
Internal relay		M	Yes	No	No	No
HMI Devices	HMI Internal Relay	LM	No	Yes	Yes	No
	HMI Keep Relay	LK	No	Yes	Yes	No
	HMI Temporary Relay	LBM	No	Yes	Yes	No
Control Device	Internal relay	M	No	Yes	No	No
		#M	No	No	No	Yes

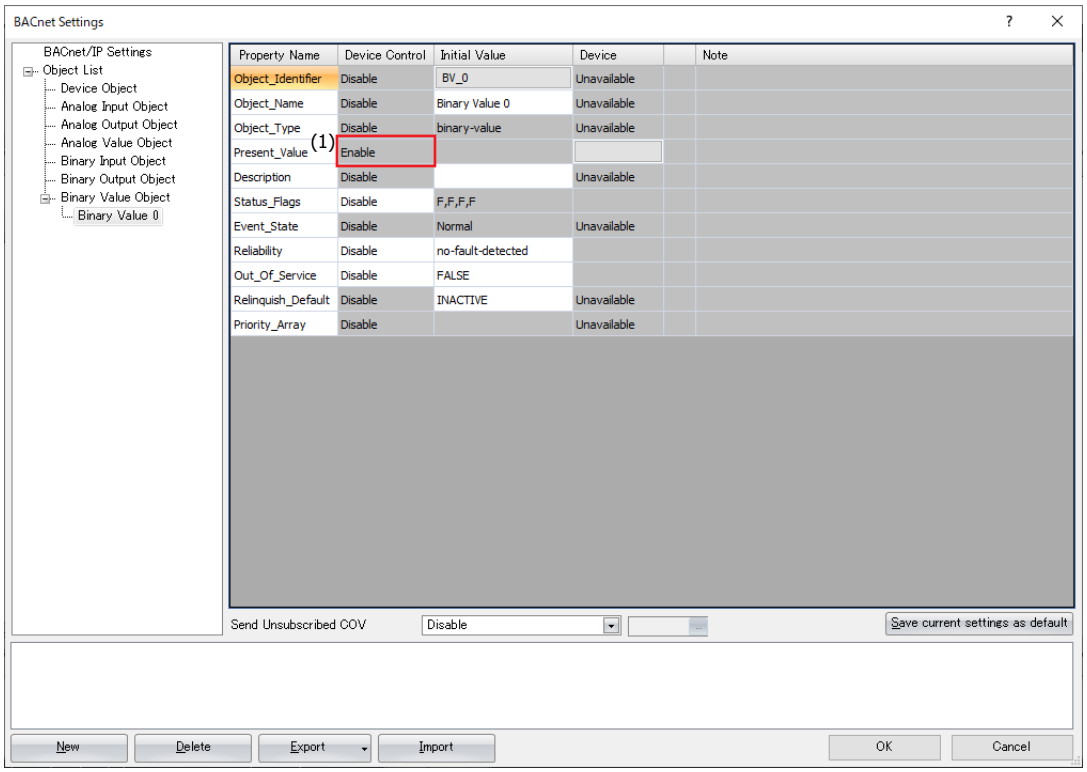
For details, see the following manuals.

FC6A: Chapter 6 "Devices" in the "FC6A Series MICROSmart User's Manual".

FT2J/1J, HG2J/1J, HG5G/4G/3G/2G-V: Chapter 35 "1 Bit Devices" in the "WindO/I-NV4 User's Manual".

Binary Value Object

Use these settings to configure device addresses for Present_Value of the Binary Value object.

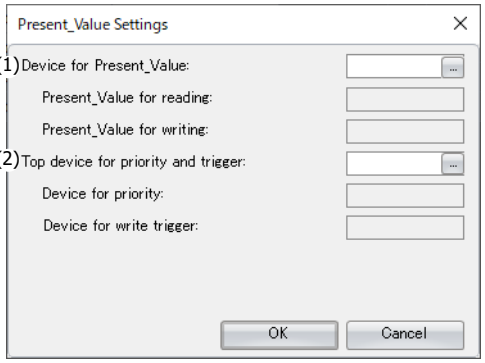


(1) Device Control

Device Control of **Present_Value** is **Enable**. A constant value cannot be configured for **Present_Value** of the Binary Value object.

■**Allocating Device Addresses to Present_Value**

Configure the parameters in the **Present_Value Settings** dialog box.



(1) Device for Present_Value

Configures the device address for reading Present_Value and the device for writing Present_Value.
The bit device that can be set depends on the main unit type.

Internal Device Name		Symbol	FC6A	FT2J/1J	HG2J/1J	HG5G/4G/3G/2G-V
Internal relay		M	Yes	No	No	No
HMI Devices	HMI Internal Relay	LM	No	Yes	Yes	No
	HMI Keep Relay	LK	No	Yes	Yes	No
	HMI Temporary Relay	LBM	No	Yes	Yes	No
Control Device	Internal relay	M	No	Yes	No	No
		#M	No	No	No	Yes

For details, see the following manuals.
FC6A: Chapter 6 "Devices" in the "FC6A Series MICROSmart User's Manual".
FT2J/1J, HG2J/1J, HG5G/4G/3G/2G-V: Chapter 35 "1 Bit Devices" in the "WindO/I-NV4 User's Manual".

The devices for reading Present_Value and writing Present_Value are automatically assigned according to the set device address. Starting from the specified address number of Device Address, 2 bits of address number are used.

Device for Present_Value	Storage Destination
Present_Value for reading	The address number of top device address+0
Present_Value for writing	The address number of top device address+1

(2) Top device for priority and trigger

Use this device address when writing the value of the device addresses to Present_Value. For details, see Present_Value in "Binary Value Object" on page 1-52.

Device for priority and **Device for write trigger** are automatically assigned when the device address is set. Starting from the specified address number of device address, 2 continuous words of address number are used.

Main unit	Storage Destination	Description
Device for priority	The address number of Top Device Address+0	Bit 15: 0: the value of the Present_Value for writing. 1: NULL(00h). Bits 14 to 5: Disabled Bits 4 to 0: Priority*1
Device for write trigger	The address number of Top Device Address+1	When the Device for write trigger turns on from off, writes a value to Priority_Array of index number indicated by the priority (bit 4 to 0 of Device for priority). The value to be written varies based on the value of bit 15 of Device for priority.

*1 Specify the priority between 1 and 16. If the priority is out of range, If the Device for write trigger turns on from off, then nothing is executed.

The word device that can be set depends on the main unit type.

Internal Device Name		Symbol	FC6A	FT2J/1J	HG2J/1J	HG5G/4G/3G/2G-V
Data Register		D	Yes	No	No	No
HMI Devices	HMI Data Register	LDR	No	Yes	Yes	No
	HMI Keep Register	LKR	No	Yes	Yes	No
	HMI Temporary Register	LBR	No	Yes	Yes	No
Control Devices	Data Register	D	No	Yes	No	No
		#D	No	No	No	Yes

For details, see the following manuals.
FC6A: Chapter 6 "Devices" in the "FC6A Series MICROSmart User's Manual".
FT2J/1J, HG2J/1J, HG5G/4G/3G/2G-V: Chapter 35 "2 Word Devices" in the "WindO/I-NV4 User's Manual".

Objects

The main unit holds objects registered with WindLDR or WindO/I-NV4 in internal memory. Part of the properties of the object can be allocated to the device address, and the value of the device address can be written and read. BACnet/IP devices can read and write the properties of objects on the main unit using services.

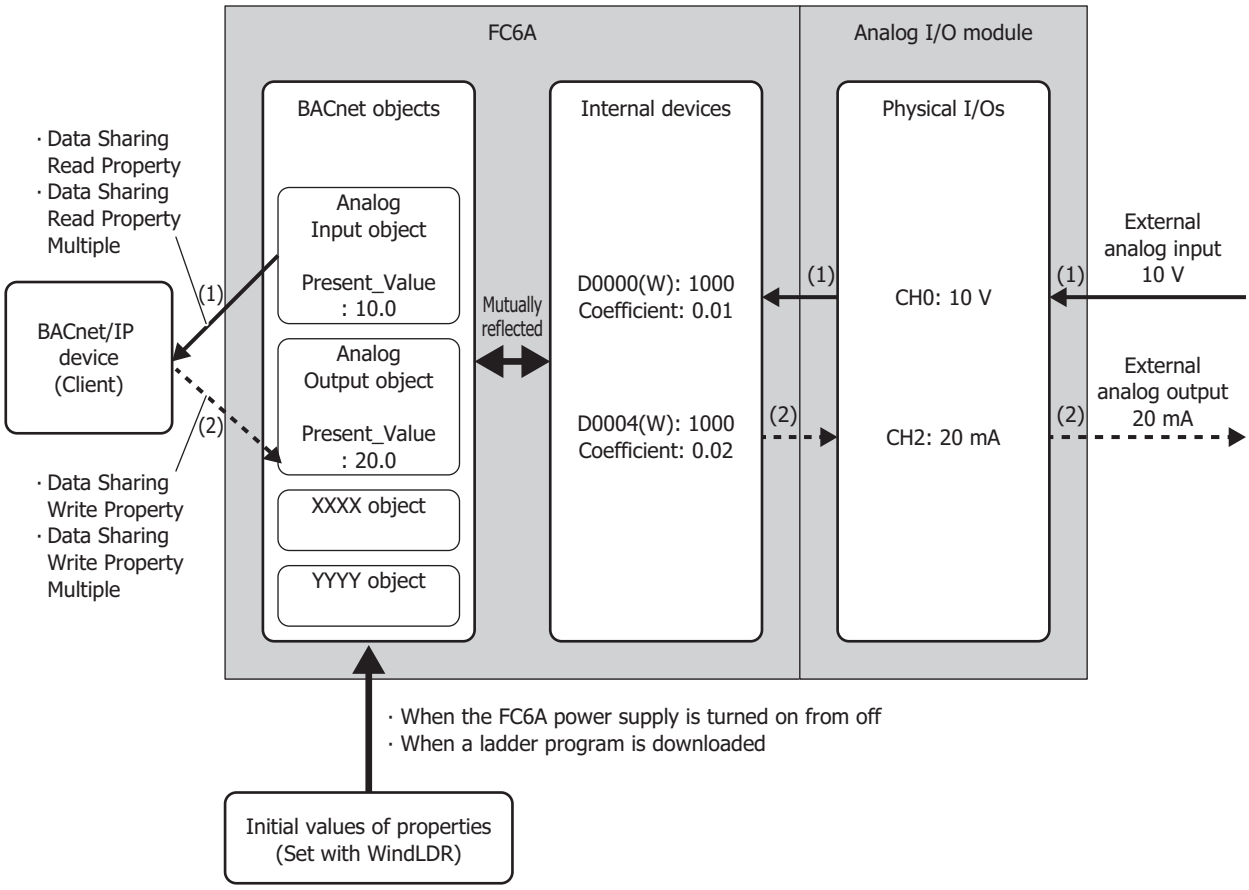
Properties of objects on the main unit and values of device addresses allocated to properties are continually synchronized.

For how to register objects using WindLDR or WindO/I-NV4, see "Adding a New Object" on page 1-18. The maximum number of objects that can be registered is 256.

The following diagram shows two concepts.

(1) The BACnet/IP device reads the analog input value of the Analog Input object.

(2) The BACnet/IP device writes the analog output value of the Analog Output object.



For details on properties held by objects, refer to "BACnet Building Automation Data Communication Protocol".

Analog Input Object

This object manages a float-type numeric value. The main unit can make arbitrary numeric values available to BACnet/IP devices. For example, use this object when handling analog values input with the analog input module, the measured room temperature, and other analog values.

■Properties List

This list shows the properties supported by the Analog Input object.

(1): Read and write from BACnet/IP devices

(2): Read and write from device addresses when properties are allocated to device addresses

R: Read-only, W: Write-only, R/W: Readable/writable, -: Not readable/writable

Property Identifier	Data Type of Property	(1)	(2)	Comments
Object_Identifier	BACnetObjectIdentifier	R	-	Set a fixed value with WindLDR or WindO/I-NV4.
Object_Name	Character string*1	R	-	Set a fixed value with WindLDR or WindO/I-NV4.
Object_Type	BACnetObjectType	R	-	Set a fixed value with WindLDR or WindO/I-NV4.
Present_Value	Real number	R	W	See "Present_Value" on page 1-41.
Description	Character string*1	R	-	Set a fixed value with WindLDR or WindO/I-NV4.
Device_Type	Character string*1	R	-	Set a fixed value with WindLDR or WindO/I-NV4.
Status_Flags	BACnetStatus_Flags	R	R	See "Status_Flags" on page 1-54.
Event_State	BACnetEventState	R	-	Normal (fixed).
Reliability	BACnetReliability	R	R/W	Indicates whether or not the value of Present_Value is reliable.
Out_Of_Service	Logical value	R/W	R/W	See "Out_Of_Service" on page 1-55.
Units	BACnetEngineeringUnits	R/W	-	Set the initial value with WindLDR or WindO/I-NV4.
Resolution	Real number	R	-	Set a fixed value with WindLDR or WindO/I-NV4.
COV_Increment	Real number	R/W	-	See "COV_Increment" on page 1-54.
Property_List	BACnetARRAY[N] of type BACnetPropertyIdentifier	R	-	Cannot be displayed on WindLDR and WindO/I-NV4.

*1 Character encoding is ISO 10646 (UTF-8), and the maximum size is 64 bytes.

Analog Output Object

This object manages a float-type numeric value. The main unit can receive arbitrary numeric values from BACnet/IP devices. For example, this object is used when receiving analog values such as those values that are output with the analog output module and temperature set points used as operating parameters from BACnet/IP devices.

■Properties List

This list shows the properties supported by the Analog Output object.

(1): Read and write from BACnet/IP devices

(2): Read and write from device addresses when properties are allocated to device addresses

R: Read-only, W: Write-only, R/W: Readable/writable, -: Not readable/writable

Property Identifier	Data Type of Property	(1)	(2)	Comments
Object_Identifier	BACnetObjectIdentifier	R	-	Set a fixed value with WindLDR or WindO/I-NV4.
Object_Name	Character string*1	R	-	Set a fixed value with WindLDR or WindO/I-NV4.
Object_Type	BACnetObjectType	R	-	Set a fixed value with WindLDR or WindO/I-NV4.
Present_Value	Real number	R	R	See "Present_Value" on page 1-41.
Description	Character string*1	R	-	Set a fixed value with WindLDR or WindO/I-NV4.
Device_Type	Character string*1	R	-	Set a fixed value with WindLDR or WindO/I-NV4.
Status_Flags	BACnetStatus_Flags	R	R	See "Status_Flags" on page 1-54.
Event_State	BACnetEventState	R	-	Normal (fixed).
Reliability	BACnetReliability	R	R/W	Indicates whether or not the value of Present_Value is reliable.
Out_Of_Service	Logical value	R/W	R/W	See "Out_Of_Service" on page 1-55.
Units	BACnetEngineeringUnits	R/W	-	Set the initial value with WindLDR or WindO/I-NV4.
Resolution	Real number	R	-	Set a fixed value with WindLDR or WindO/I-NV4.
Priority_Array	BACnetPriority_Array	R/W	-	See "Priority_Array" on page 1-54.
Relinquish_Default	Real number	R/W	-	See "Relinquish_Default" on page 1-54.
COV_Increment	Real number	R/W	-	See "COV_Increment" on page 1-54.
Property_List	BACnetARRAY[N] of type BACnetPropertyIdentifier	R	-	Cannot be displayed on WindLDR and WindO/I-NV4.

*1 Character encoding is ISO 10646 (UTF-8), and the maximum size is 64 bytes.

Analog Value Object

This object can be used in the same manner as the Analog Input object and Analog Output object.

■Properties List

This list shows the properties supported by the Analog Value object.

(1): Read and write from BACnet/IP devices

(2): Read and write from device addresses when properties are allocated to device addresses

R: Read-only, W: Write-only, R/W: Readable/writable, -: Not readable/writable

Property Identifier	Data Type of Property	(1)	(2)	Comments
Object_Identifier	BACnetObjectIdentifier	R	-	Set a fixed value with WindLDR or WindO/I-NV4.
Object_Name	Character string*1	R	-	Set a fixed value with WindLDR or WindO/I-NV4.
Object_Type	BACnetObjectType	R	-	Set a fixed value with WindLDR or WindO/I-NV4.
Present_Value	Real number	R	R	See "Present_Value" on page 1-41.
Description	Character string*1	R	-	Set a fixed value with WindLDR or WindO/I-NV4.
Status_Flags	BACnetStatus_Flags	R	R	See "Status_Flags" on page 1-54.
Event_State	BACnetEventState	R	-	Normal (fixed).
Reliability	BACnetReliability	R	R/W	Indicates whether or not Present_Value is reliable.
Out_Of_Service	Logical value	R/W	R/W	See "Out_Of_Service" on page 1-55.
Units	BACnetEngineeringUnits	R/W	-	Set the initial value with WindLDR or WindO/I-NV4.
Priority_Array	BACnetPriority_Array	R/W	R/W	See "Priority_Array" on page 1-54.
Relinquish_Default	Real number	R/W	-	See "Relinquish_Default" on page 1-54.
COV_Increment	Real number	R/W	-	See "COV_Increment" on page 1-54.
Resolution	Real number	R	-	Set a fixed value with WindLDR or WindO/I-NV4.
Property_List	BACnetARRAY[N] of type BACnetPropertyIdentifier	R	-	Cannot be displayed on WindLDR and WindO/I-NV4.

*1 Character encoding is ISO 10646 (UTF-8), and the maximum size is 64 bytes.

Binary Input Object

This object manages a binary value (on/off). Use this object when the main unit makes binary values available to BACnet/IP devices.

■Properties List

This list shows the properties supported by the Binary Input object.

(1): Read and write from BACnet/IP devices

(2): Read and write from device addresses when properties are allocated to device addresses

R: Read-only, W: Write-only, R/W: Readable/writable, -: Not readable/writable

Property Identifier	Data Type of Property	(1)	(2)	Comments
Object_Identifier	BACnetObjectIdentifier	R	-	Set a fixed value with WindLDR or WindO/I-NV4.
Object_Name	Character string*1	R	-	Set a fixed value with WindLDR or WindO/I-NV4.
Object_Type	BACnetObjectType	R	-	Set a fixed value with WindLDR or WindO/I-NV4.
Present_Value	BACnetBinaryPV	R	W	See "Present_Value" on page 1-41.
Description	Character string*1	R	-	Set a fixed value with WindLDR or WindO/I-NV4.
Device_Type	Character string*1	R	-	Set a fixed value with WindLDR or WindO/I-NV4.
Status_Flags	BACnetStatus_Flags	R	R	See "Status_Flags" on page 1-54.
Event_State	BACnetEventState	R	-	Normal (fixed).
Reliability	BACnetReliability	R	R/W	Indicates whether or not the value of Present_Value is reliable.
Out_Of_Service	Logical value	R/W	R/W	See "Out_Of_Service" on page 1-55.
Polarity	BACnetPolarity	R/W	-	See "Polarity" on page 1-55.
Property_List	BACnetARRAY[N] of type BACnetPropertyIdentifier	R	-	Cannot be displayed on WindLDR and WindO/I-NV4.

*1 Character encoding is ISO 10646 (UTF-8), and the maximum size is 64 bytes.

Binary Output Object

This object manages a binary value (on/off). Use this object when the main unit receives binary values from BACnet/IP devices.

■Properties List

This list shows the properties supported by the Binary Output object.

(1): Read and write from BACnet/IP devices

(2): Read and write from device addresses when properties are allocated to device addresses

R: Read-only, W: Write-only, R/W: Readable/writable, -: Not readable/writable

Property Identifier	Data Type of Property	(1)	(2)	Comments
Object_Identifier	BACnetObjectIdentifier	R	-	Set a fixed value with WindLDR or WindO/I-NV4.
Object_Name	Character string*1	R	-	Set a fixed value with WindLDR or WindO/I-NV4.
Object_Type	BACnetObjectType	R	-	Set a fixed value with WindLDR or WindO/I-NV4.
Present_Value	BACnetBinaryPV	R	R	See "Present_Value" on page 1-41.
Description	Character string*1	R	-	Set a fixed value with WindLDR or WindO/I-NV4.
Device_Type	Character string*1	R	-	Set a fixed value with WindLDR or WindO/I-NV4.
Status_Flags	BACnetStatus_Flags	R	R	See "Status_Flags" on page 1-54.
Event_State	BACnetEventState	R	-	Normal (fixed).
Reliability	BACnetReliability	R	R/W	Indicates whether or not the value of Present_Value is reliable.
Out_Of_Service	Logical value	R/W	R/W	See "Out_Of_Service" on page 1-55.
Polarity	BACnetPolarity	R/W	-	See "Polarity" on page 1-55.
Priority_Array	BACnetPriority_Array	R/W	-	See "Priority_Array" on page 1-54.
Relinquish_Default	BACnetBinaryPV	R/W	-	See "Relinquish_Default" on page 1-54.
Property_List	BACnetARRAY[N] of type BACnetPropertyIdentifier	R	-	Cannot be displayed on WindLDR and WindO/I-NV4.

*1 Character encoding is ISO 10646 (UTF-8), and the maximum size is 64 bytes.

Binary Value Object

This object can be used in the same manner as the Binary Input object and Binary Output object.

■Properties List

This list shows the properties supported by the Binary Value object.

(1): Read and write from BACnet/IP devices

(2): Read and write from device addresses when properties are allocated to device addresses

R: Read-only, W: Write-only, R/W: Readable/writable, -: Not readable/writable

Property Identifier	Data Type of Property	(1)	(2)	Comments
Object_Identifier	BACnetObjectIdentifier	R	-	Set a fixed value with WindLDR or WindO/I-NV4.
Object_Name	Character string*1	R	-	Set a fixed value with WindLDR or WindO/I-NV4.
Object_Type	BACnetObjectType	R	-	Set a fixed value with WindLDR or WindO/I-NV4.
Present_Value	BACnetBinaryPV	R	R	See "Present_Value" on page 1-41.
Description	Character string*1	R	-	Set a fixed value with WindLDR or WindO/I-NV4.
Status_Flags	BACnetStatus_Flags	R	R	See "Status_Flags" on page 1-54.
Event_State	BACnetEventState	R	-	Normal (fixed).
Reliability	BACnetReliability	R	R/W	Indicates whether or not the value of Present_Value is reliable.
Out_Of_Service	Logical value	R/W	R/W	See "Out_Of_Service" on page 1-55.
Priority_Array	BACnetPriority_Array	R/W	R/W	See "Priority_Array" on page 1-54.
Relinquish_Default	BACnetBinaryPV	R/W	-	See "Relinquish_Default" on page 1-54.
Property_List	BACnetARRAY[N] of type BACnetPropertyIdentifier	R	-	Cannot be displayed on WindLDR and WindO/I-NV4.

*1 Character encoding is ISO 10646 (UTF-8), and the maximum size is 64 bytes.

Device Object

Use this object when the main unit makes basic information available to BACnet/IP devices.

■Properties List

This list shows the properties supported by the Device object.

(1): Read and write from BACnet/IP devices

(2): Read and write from device addresses when properties are allocated to device addresses

R: Read-only, W: Write-only, R/W: Readable/writable, -: Not readable/writable

Property Identifier	Data Type of Property	(1)	(2)	Comments
Object_Identifier	BACnetObjectIdentifier	R	-	Cannot be edited with WindLDR and WindO/I-NV4.
Object_Name	Character string*1	R	-	Set a fixed value with WindLDR or WindO/I-NV4.
Object_Type	BACnetObjectType	R	-	Cannot be edited with WindLDR and WindO/I-NV4.
System_Status	BACnetDeviceStatus	R	-	Cannot be edited with WindLDR and WindO/I-NV4.
Vendor_Name	Character string*1	R	-	Cannot be edited with WindLDR and WindO/I-NV4.
Vendor_Identifier	16-bit unsigned integer	R	-	Cannot be edited with WindLDR and WindO/I-NV4.
Model_Name	Character string*1	R	-	Set a fixed value with WindLDR or WindO/I-NV4.
Firmware_Revision	Character string*1	R	-	Cannot be edited with WindLDR and WindO/I-NV4.
Application_Software_Version	Character string*1	R	-	Set a fixed value with WindLDR or WindO/I-NV4.
Location	Character string*1	R/W	-	Set the initial value with WindLDR or WindO/I-NV4.
Description	Character string*1	R/W	-	Set the initial value with WindLDR or WindO/I-NV4.
Protocol_Version	Unsigned integer	R	-	Cannot be edited with WindLDR and WindO/I-NV4.
Protocol_Revision	Unsigned integer	R	-	Cannot be edited with WindLDR and WindO/I-NV4.
Protocol_Services_Supported	BACnetServicesSupported	R	-	Cannot be edited with WindLDR and WindO/I-NV4.
Protocol_Object_Types_Supported	BACnetObjectType Supported	R	-	Cannot be edited with WindLDR and WindO/I-NV4.
Object_List	BACnetARRAY[N] of type BACnetObjectIdentifier	R	-	Cannot be edited with WindLDR and WindO/I-NV4.
Max_APDU_Length_Accepted	Unsigned integer	R	-	Cannot be edited with WindLDR and WindO/I-NV4.
Segmentation_Supported	BACnetSegmentation	R	-	Cannot be edited with WindLDR and WindO/I-NV4.
Local_Time	Time	R	-	Cannot be edited with WindLDR and WindO/I-NV4.
Local_Date	Date	R	-	Cannot be edited with WindLDR and WindO/I-NV4.
APDU_Timeout	Unsigned integer	R	-	Cannot be edited with WindLDR and WindO/I-NV4.
Number_of_APDU_Retries	Unsigned integer	R	-	Cannot be edited with WindLDR and WindO/I-NV4.
Device_Address_Binding	BACnetLIST of type BACnetAddressBiding	R	-	Cannot be edited with WindLDR and WindO/I-NV4.
Database_Revision	Unsigned integer	R	-	Cannot be edited with WindLDR and WindO/I-NV4.
Property_List	BACnetARRAY[N] of type BACnetPropertyIdentifier	R	-	Cannot be displayed on WindLDR and WindO/I-NV4.
Profile_Name	Character string*1	R	-	Set a fixed value with WindLDR or WindO/I-NV4.

*1 Character encoding is ISO 10646 (UTF-8), and the maximum size is 64 bytes.

Key Properties

Present_Value

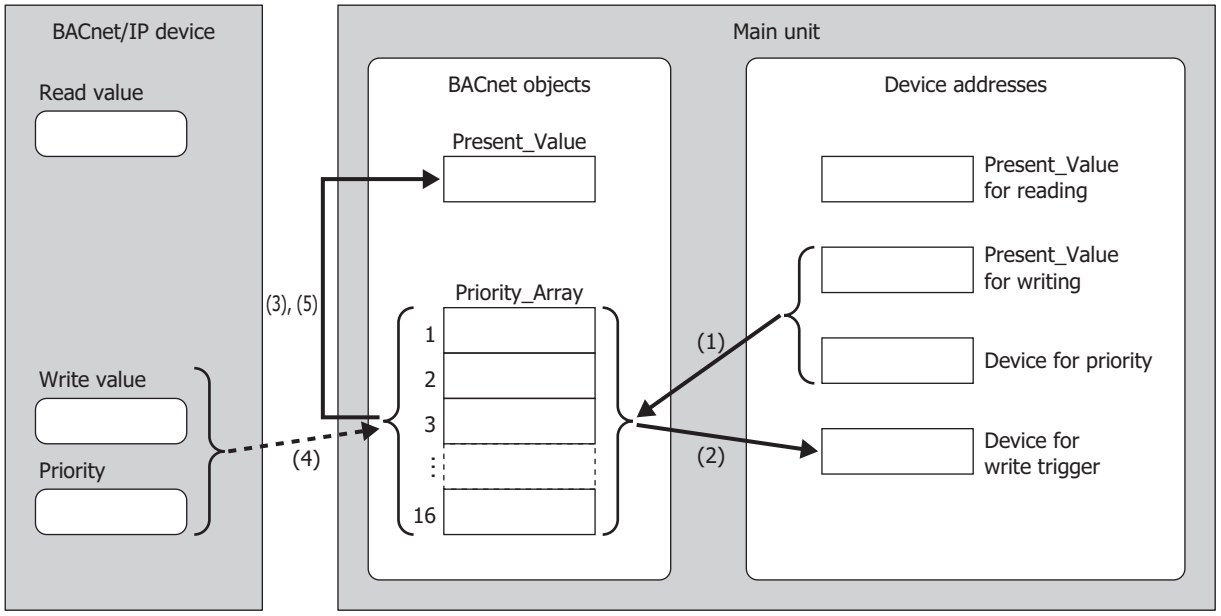
This property represents the current value. This property handles input and output values associated with objects.

Priority Mechanism

Objects*1 with Present_Value that can be written from BACnet/IP devices use a priority mechanism with Priority_Array in which write instructions are ranked and the value of Present_Value is determined. In this case, values cannot be directly written to Present_Value. To write a value to Present_Value, the value is first stored in Priority_Array ("Present_Value Settings" on page 1-20) at the index number indicated by **Priority** ((4) "Priority_Array" on page 1-54). Then the value stored in Priority_Array with the smallest index number out of all non-NULL values is used as the value of Present_Value.

*1 Analog Output/Analog Value/Binary Output/Binary Value objects

Writing the Value of a Device Address as Present_Value or Writing Present_Value from a BACnet/IP Device



The items used in the above diagram are described in the following table.

Item	Description
Device for reading	This device address stores Present_Value read from the BACnet/IP device.
Device for writing	This device address stores the value to be written to the BACnet/IP device as Present_Value.
Device for priority	This device address stores the index number in the array (Priority_Array) where the priority value is stored.
Device for write trigger	When this device address is 1, the value stored in the Present_Value for writing is stored in Priority_Array at the index number stored in device address for priority.
Present_Value	Prevent_Value held by the BACnet object in the main unit.
Priority_Array	See "Priority_Array" on page 1-54.
Read value	Current Present_Value.
Write value	Present_Value to be written from the BACnet/IP device.

Writing from a Device Address

(1) When bit 15 of **Device for priority** is 0 and the value of **Device for write trigger** changes from 0 to 1, writes the value of **Present_Value for writing** to the Priority_Array at the index number stored in the bit 4 to 0 of **Device for priority**.

(2) **Device for write trigger** is automatically reset to 0.

(3) The value stored in Priority_Array with the smallest index number out of all non-NULL(00h) values is used as the value of Present_Value. (The value of Priority_Array at the index number is used as Present_Value until it is reset to NULL(00h).)

When bit 15 of **Device for priority** is 1 and the value of **Device for write trigger** changes from 0 to 1, NULL(00h) is written to Priority_Array at the index number stored in **Device for priority**.
For details about the bit assignments of the Device for priority, see "Analog Value Object" on page 1-24 in "(2) Top device for priority and trigger" on page 1-25.

Writing from the BACnet/IP Device

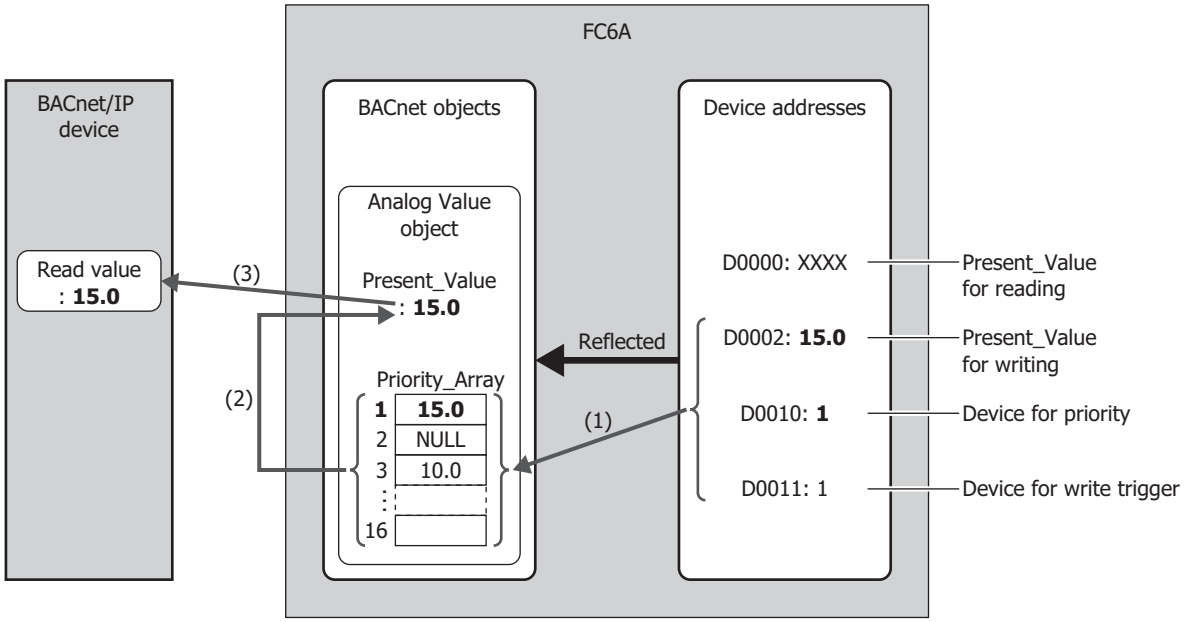
(4) The BACnet/IP device writes the value of Present_Value to Priority_Array at the index number indicated by the specified priority.

(5) The value stored in the Priority_Array property with the smallest index number out of all non-NULL(00h) values is used as the value of Present_Value. (The value of Priority_Array at the index number is used as Present_Value until it is reset to NULL(00h).)

If all values stored in Priority_Array are NULL(00h), "Relinquish_Default" on page 1-54 is used as Present_Value.

Writing the Value of a Device Address to Present_Value

Example: When D0000 is allocated to the Present_Value device, D0010 is allocated to the top device for priority and trigger, and Float(F) is allocated to the conversion type, the device addresses are allocated as follows: D0000 is the Present_Value for reading, D0002 is the Present_Value for writing, D0010 is the Device for priority, and D0011 is the Device for write trigger.



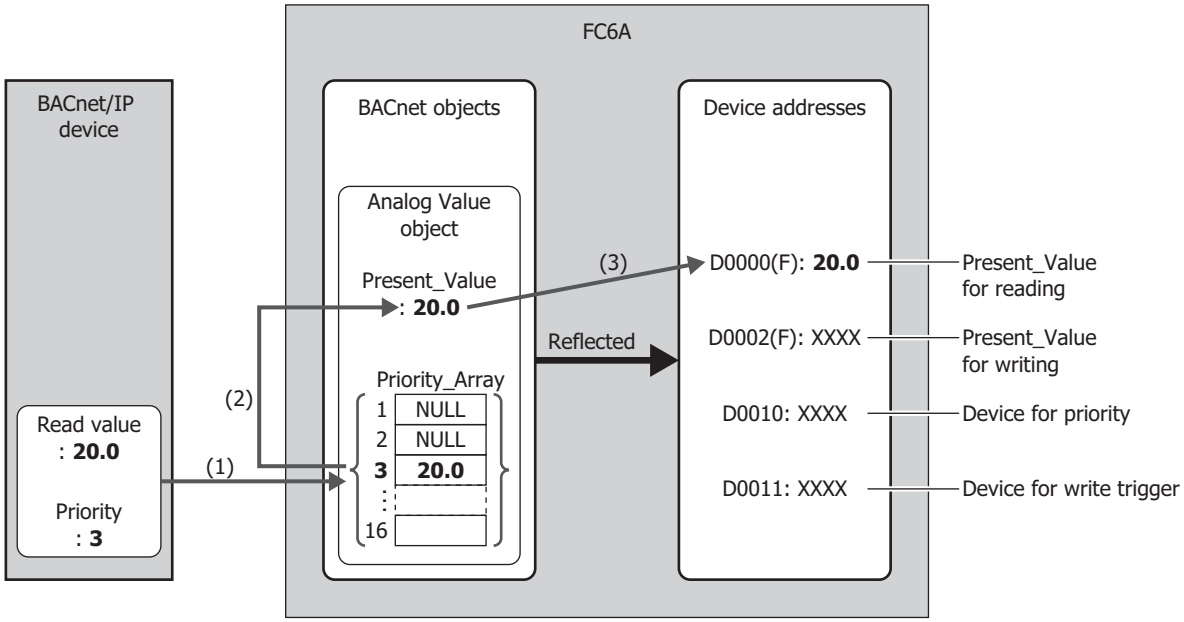
(1) If the value of the Device for priority (D0010) is 1 when the value of the Device for write trigger (D0011) is 1, writes the value (15.0) of the Present_Value for writing (D0002) to the element 1 of Priority_Array. After the write has completed, the value of the Device for write trigger (D0011) is reset to 0.

(2) Priority_Array with the smallest index number out of all non-NULL(00h) values is element 1 (15.0), so 15.0 is used as the value of Present_Value.

(3) Present_Value (15.0) is read from the BACnet/IP device.

Writing a Value to Present_Value from the BACnet/IP Device

Example: When D0000 is allocated to the Present_Value device, D0010 is allocated to the top device for priority and trigger, and Float(F) is allocated to the conversion type, the device addresses are allocated as follows: D0000 is the Present_Value for reading, D0002 is the Present_Value for writing, D0010 is the Device for priority, and D0011 is the Device for write trigger.



(1) The BACnet/IP device writes the write value (20.0) to Priority_Array at the index number indicated by the specified priority (3).

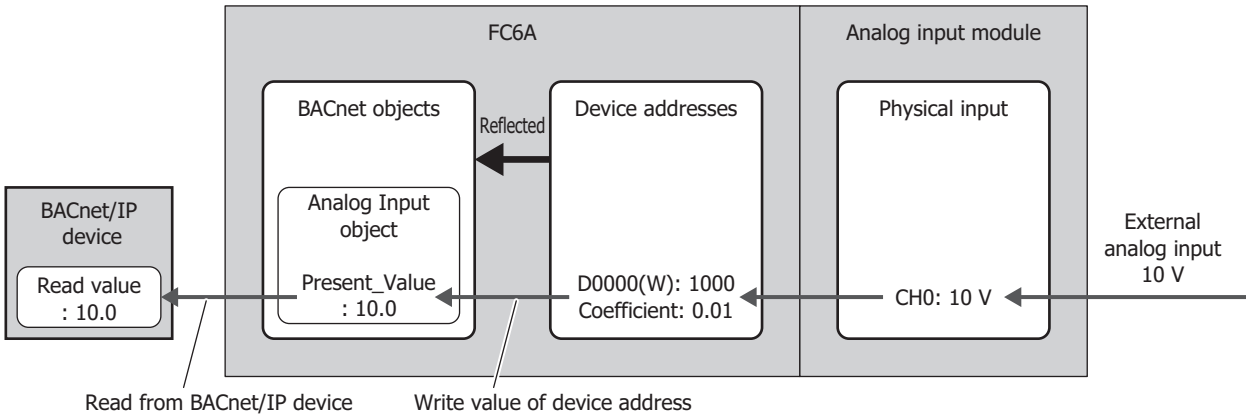
(2) Priority_Array with the smallest index number out of all non-NULL(00h) values is element 3 (20.0), so 20.0 is used as the value of Present_Value.

(3) Present_Value (20.0) is written to the Present_Value for reading (D0000).

Analog Input Object

Present_Value of the Analog Input object can be set to a fixed value or allocated device addresses and set to the value of those device address. Present_Value is a float-type numeric value. When device addresses are allocated to Present_Value, Present_Value is set to the product of those device addresses multiplied by the coefficient.

The following diagram illustrates the concept when device addresses are allocated to Present_Value of the Analog Input object and the BACnet/IP device reads out the analog input value.



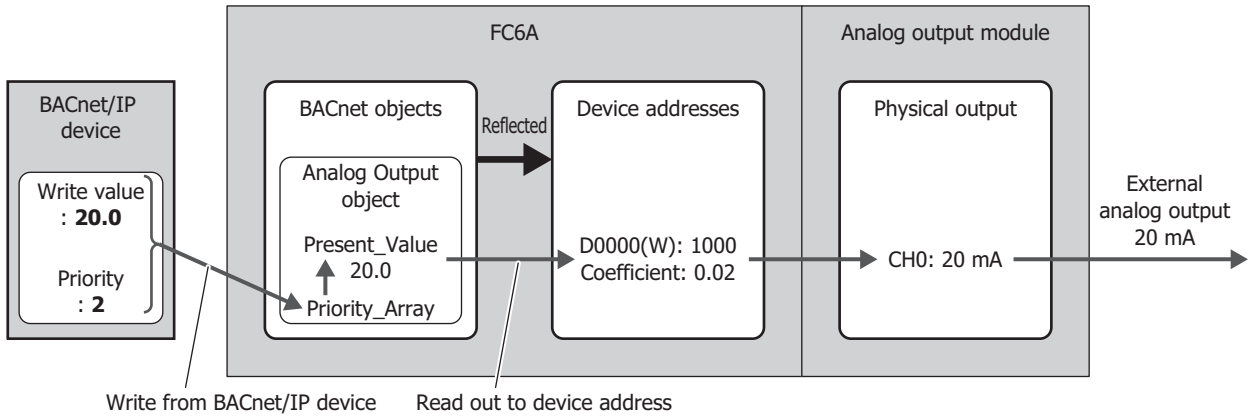
When writing the value of device addresses as Present_Value, Present_Value changes in the following manner.

Data Type		Value of Device Addresses	Present_Value of BACnet/IP Device
FC6A	FT2J/1J, HG2J/1J, HG5G/4G/3G/2G-V		
Word (W)	UBIN16(W)	Value within range of each data type	Value of Device Addresses
Integer (I)	BIN16(I)		
Double Word (D)	UBIN32(D)		
Long (L)	BIN32(L)		
Float (F)	Float32(F)	±0	±0.0
		Denormalized number	Value of Device Addresses
		Normalized number	
		±∞ (±infinity)	Present_Value does not change
		Non-number	

Analog Output Object

Present_Value of the Analog Output object can be allocated device addresses and set to the value of those device addresses. Present_Value is a float-type numeric value. When device addresses are allocated to Present_Value, the product of Present_Value multiplied by 1/coefficient is stored in the device addresses.

The following diagram illustrates the concept when device addresses are allocated to Present_Value of the Analog Output object and the BACnet/IP device writes the analog output value.



Present_Value of the Analog Output object cannot be changed from the main unit and ladder prpgram.

When reading out Present_Value to device addresses, you must be aware of the data type. Depending on the data type of the device addresses, the value is stored in the device addresses as follows. Set the data type according to the value of Present_Value. The data type of Present_Value is set in "(2) Conversion type" on page 1-23 of the **Present_Value Settings** dialog box (page 1-23).

Data Type		Present_Value of BACnet/IP Device	Value of Data Registers
FC6A	FT2J/1J, HG2J/1J, HG5G/4G/3G/2G-V		
Word (W)	UBIN16(W)	Value within range of 0 to 65,535	Present_Value
		Value outside range of 0 to 65,535	0
Integer (I)	BIN16(I)	Value within range of -32,768 to 32,767	Present_Value
		Value outside range of -32,768 to 32,767	0
Double Word (D)	UBIN32(D)	Value within range of 0 to 4,294,967,295	Present_Value
		Value outside range of 0 to 4,294,967,295	0
Long (L)	BIN32(L)	Value within range of -2,147,483,648 to 2,147,483,647	Present_Value
		Value outside range of -2,147,483,648 to 2,147,483,647	0
Float (F)	Float32(F)	—	Present_Value

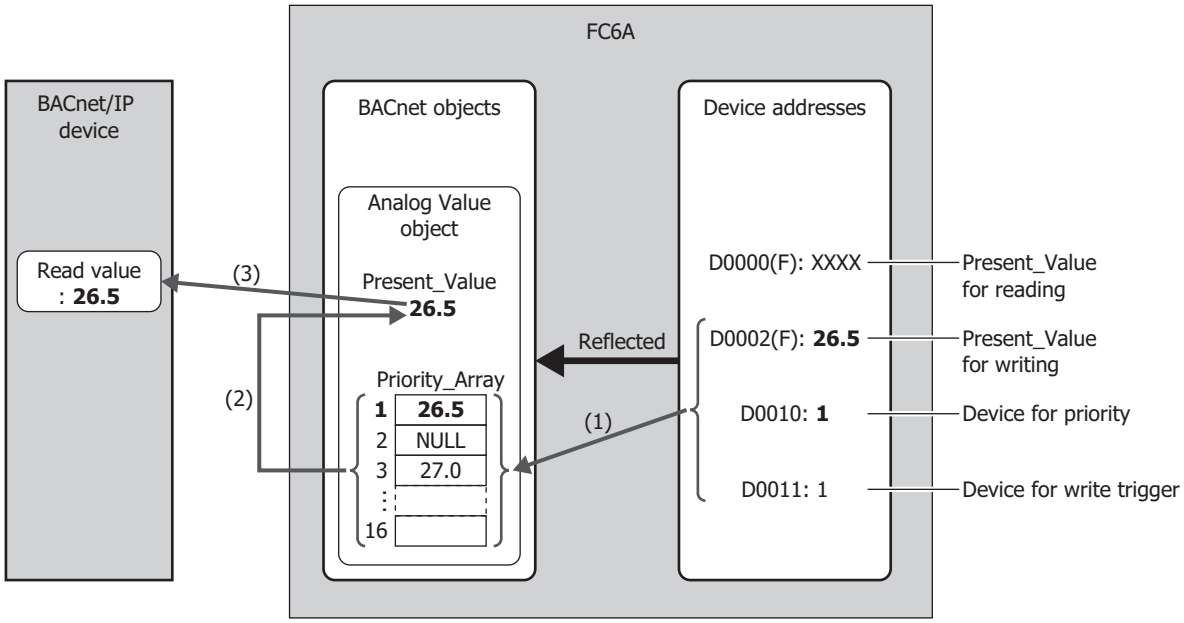
Analog Value Object

Present_Value of the Analog Value object can be used as Present_Value of both the Analog Input and Analog Output objects. Present_Value is a float-type numeric value.

The following diagram illustrates the concept when device addresses are allocated to Present_Value of the Analog Value object and the temperature of an office air conditioner is temporarily lowered from the base temperature (27.0°C) to 26.5°C.

Example: When D0000 is allocated to the Present_Value device, D0010 is allocated to the top device for priority and trigger, and Float(F) is allocated to the conversion type, the device addresses are allocated as follows: D0000 is the Present_Value for reading, D0002 is the Present_Value for writing, D0010 is the Device for priority, and D0011 is the Device for write trigger.

Writing the Value of a Device Address to Present_Value

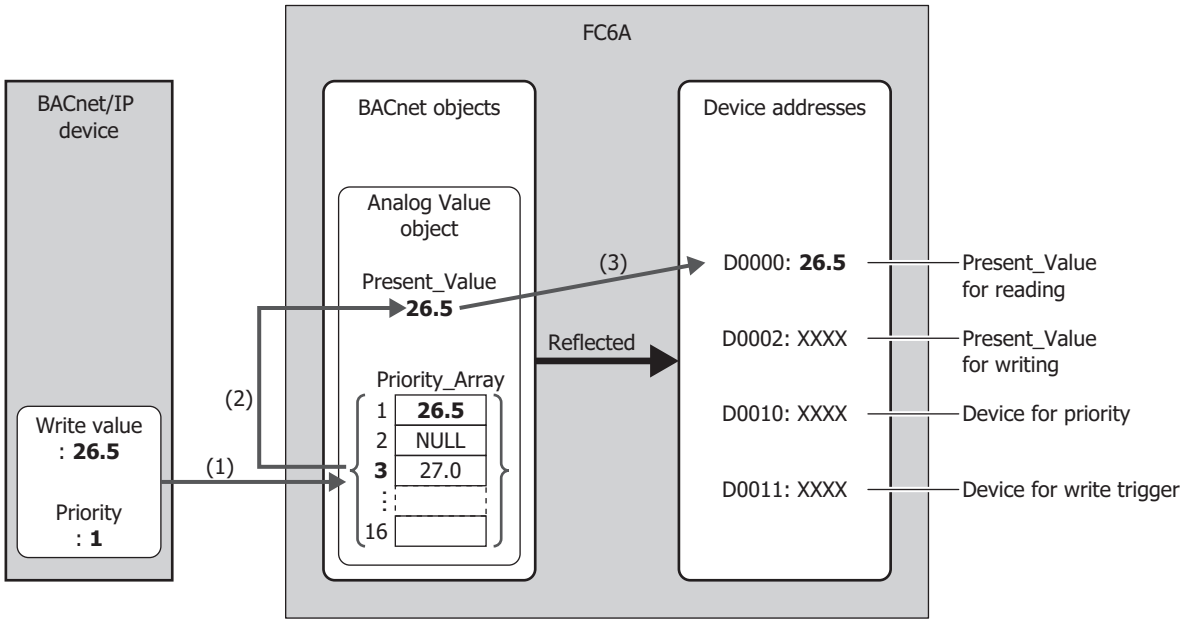


(1) When the value of the Device for write trigger (D0011) is 1, writes the value of the Present_Value for writing (D0002) to the Priority_Array at the index number stored in the Device for priority (D0010).

(2) The Device for write trigger (D0011) is automatically reset to 0.

(3) Priority_Array with the smallest index number out of all non-NULL(00h) values is element 1 (26.5), so 26.5 is used as the value of Present_Value.

Writing a Value to Present_Value from the BACnet/IP Device



(1) The BACnet/IP device writes the write value (26.5) to Priority_Array at the index number indicated by the specified priority (1).

(2) Priority_Array with the smallest index number out of all non-NULL(00h) values is element 1 (26.5), so 26.5 is used as the value of Present_Value.

(3) Present_Value (26.5) is written to the Present_Value for reading (D0000).

Element 1 (26.5) of Priority_Array is used as the value of Present_Value until it is reset to NULL(00h). When element 1 is reset to NULL(00h), element 3 (27.0) is used as the value of Present_Value.
For how to write NULL(00h), see "Analog Value Object" on page 1-24 in "(2) Top device for priority and trigger" on page 1-25.

Depending on the data type of the allocated device addresses, Present_Value is stored in the device addresses as follows. Set the data type according to the value of Present_Value. The data type of Present_Value is set in "(3) Conversion type" on page 1-25 of the "Analog Value Object".

Data Type		Present_Value of BACnet/IP Device × (1/Coefficient)	Value of Device Addresses
FC6A	FT2J/1J, HG2J/1J, HG5G/4G/3G/2G-V		
Word (W)	UBIN16(W)	Value within range of 0 to 65,535	Present_Value
		Value outside range of 0 to 65,535	0
Integer (I)	BIN16(I)	Value within range of -32,768 to 32,767	Present_Value
		Value outside range of -32768 to 32,767	0
Double Word (D)	UBIN32(D)	Value within range of 0 to 4,294,967,295	Present_Value
		Value outside range of 0 to 4,294,967,295	0
Long (L)	BIN32(L)	Value within range of -2,147,483,648 to 2,147,483,647	Present_Value
		Value outside range of -2,147,483,648 to 2,147,483,647	0
Float (F)	Float32(F)	—	Present_Value

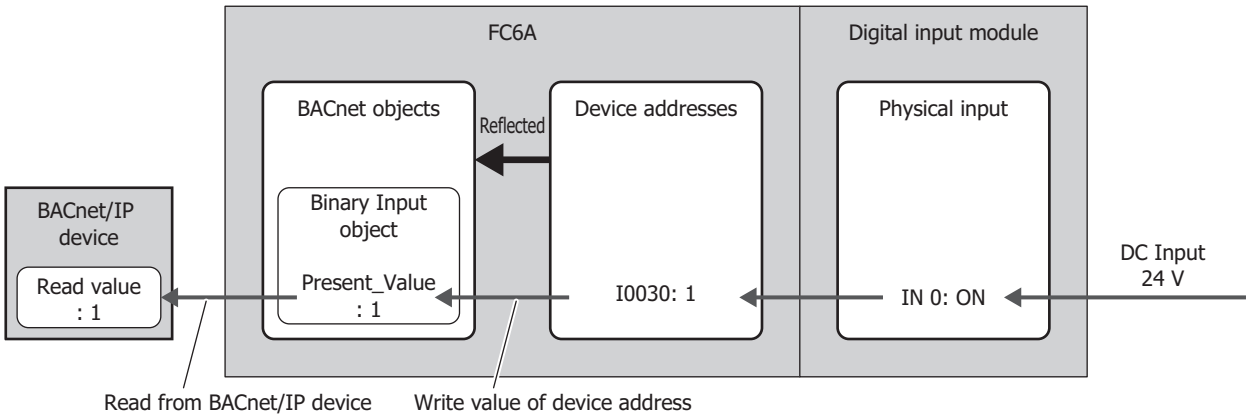
Additionally, when writing the value of device addresses to Present_Value, Present_Value changes in the following manner.

Data Type		Value of Data Registers	Present_Value of BACnet/IP Device
FC6A	FT2J/1J, HG2J/1J, HG5G/4G/3G/2G-V		
Word (W)	UBIN16(W)	Value within range of each data type	Value of data registers × Coefficient
Integer (I)	BIN16(I)		
Double Word (D)	UBIN32(D)		
Long (L)	BIN32(L)		
Float (F)	Float32(F)	±0	±0.0
		Denormalized number	Value of data registers × Coefficient
		Normalized number	
		±∞ (±infinity)	Present_Value does not change
		Non-number	

Binary Input Object

Present_Value of the Binary Input object can be set to a fixed value or allocated to a bit device address and set to the value of that bit device.

The following diagram illustrates the concept when an external input is allocated to Present_Value of the Binary Input object and the BACnet/IP device reads out the state of the external input.



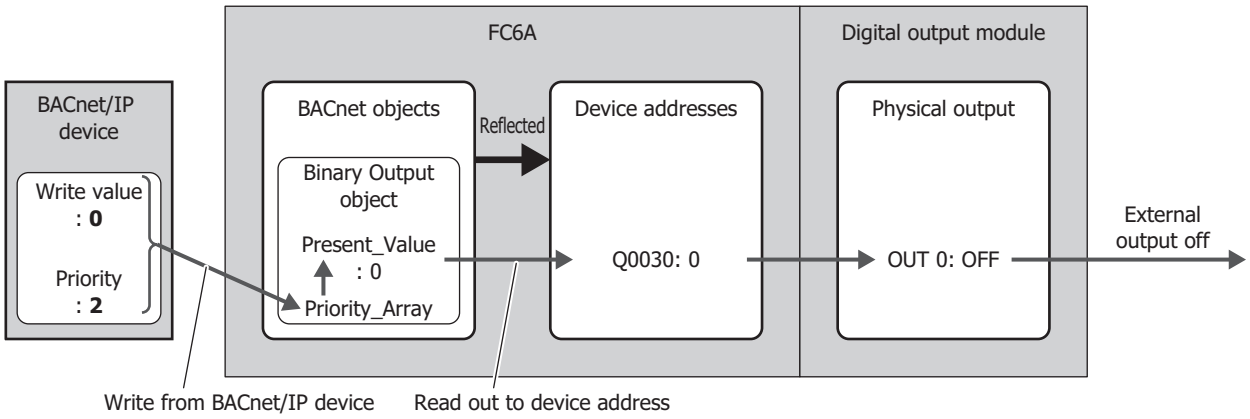
The following table shows Present_Value, Polarity, and the physical state of the input.

Present_Value	Polarity	Physical State of Input
INACTIVE	NORMAL	OFF or INACTIVE
ACTIVE	NORMAL	ON or ACTIVE
INACTIVE	REVERSE	ON or ACTIVE
ACTIVE	REVERSE	OFF or INACTIVE

Binary Output Object

Present_Value of the Binary Output object can be allocated to a bit device and set to the value of that bit device.

The following diagram illustrates the concept when a device address is allocated to Present_Value of the Binary Output object and the BACnet/IP device writes the state of the external output.



The following table shows Present_Value, Polarity, and the physical state of the output.

Present_Value	Polarity	Physical State of Output
INACTIVE	NORMAL	OFF or INACTIVE
ACTIVE	NORMAL	ON or ACTIVE
INACTIVE	REVERSE	ON or ACTIVE
ACTIVE	REVERSE	OFF or INACTIVE

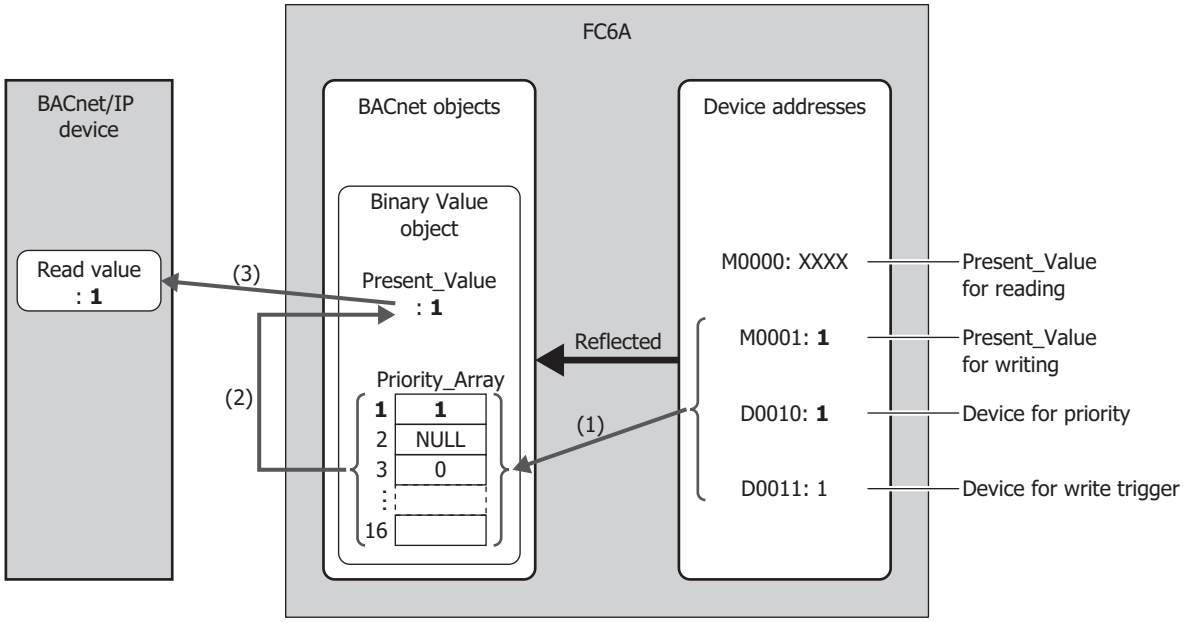
Binary Value Object

Present_Value of the Binary Value object can be used as Present_Value of both the Binary Input and Binary Output objects.

The following diagram illustrates the concept when an internal relay is allocated to Present_Value of the Binary Value object and the office lights are temporarily turned on from off.

Example: When M0000 is allocated to the Present_Value device and D0010 is allocated to the top device for priority and trigger, the device address are allocated as follows: M0000 is the Present_Value for reading, M0001 is the Present_Value for writing, D0010 is the Device for priority, and D0011 is the Device for write trigger.

Writing the Value of a Device Address to Present_Value

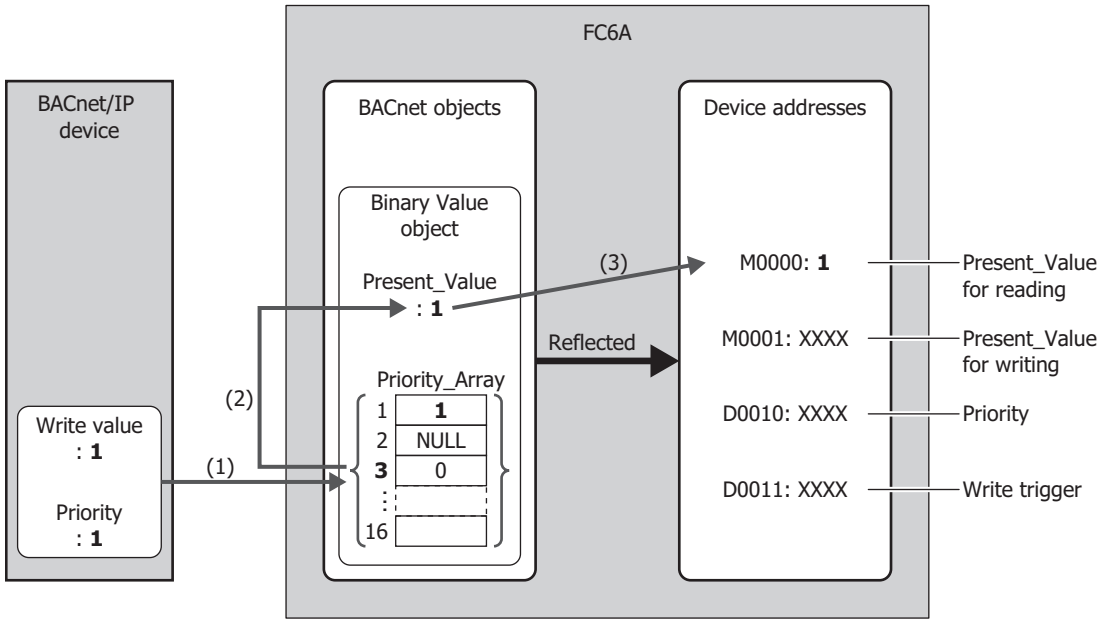


(1) When the value of the Device for write trigger (D0011) is 1, writes the value of the Present_Value for writing (M0001) to the Priority_Array at the index number stored in the Device for priority (D0010).

(2) The value of the Device for write trigger (D0011) is automatically reset to 0.

(3) Priority_Array with the smallest index number out of all non-NULL(00h) values is element 1 (1), so 1 is used as the value of Present_Value.

Writing a Value to Present_Value from the BACnet/IP Device



(1) The BACnet/IP device writes the write value (1) to Priority_Array at the index number indicated by the specified priority (1).

(2) Priority_Array with the smallest index number out of all non-NULL(00h) values is element 1 (1), so 1 is used as the value of Present_Value.

(3) Present_Value (1) is written to the Present_Value for reading (M0000).

Element 1 (1) of Priority_Array is used as the value of Present_Value until it is reset to NULL(00h). When element 1 is reset to NULL(00h), element 3 (0) is used as the value of Present_Value.
For how to write NULL(00h), see "Binary Value Object" on page 1-31 in "(2) Top device for priority and trigger" on page 1-32.

Status_Flags

This property represents the current status of the object (in alarm, fault, out of service, etc.).

Status_Flags	Value	Logical Value	Condition
IN_ALARM*1	0	FALSE	When obtaining a value where Event State is Normal.
	1	TRUE	Other than above
FAULT	0	FALSE	Other than below
	1	TRUE	When Reliability is present and the no-fault-detected value is not obtained
OVERRIDDEN*1	0	FALSE	Other than below
	1	TRUE	Present_Value and Reliability did not follow the change in physical input
OUT_OF_SERVICE	0	FALSE	When Out_Of_Service is FALSE
	1	TRUE	When Out_Of_Service is TRUE

*1 Always FALSE on the main unit.

Status_Flags can be read out to a device address. The word device that can be specified depends on the main unit type.

Internal Device Name		Symbol	FC6A	FT2J/1J	HG2J/1J	HG5G/4G/3G/2G-V
Data Register		D	Yes	No	No	No
HMI Devices	HMI Data Register	LDR	No	Yes	Yes	No
	HMI Keep Register	LKR	No	Yes	Yes	No
	HMI Temporary Register	LBR	No	Yes	Yes	No
Control Devices	Data Register	D	No	Yes	No	No
		#D	No	No	No	Yes

For details, see the following manuals.
FC6A: Chapter 6 "Devices" in the "FC6A Series MICROSmart User's Manual".
FT2J/1J, HG2J/1J, HG5G/4G/3G/2G-V: Chapter 35 "2 Word Devices" in the "WindO/I-NV4 User's Manual".

The allocation of flags in the device address is as follows.

Bit	Flag	Value
Bits 15 to 8	Reserved	Undefined
Bit 7	IN_ALARM	0 (fixed)
Bit 6	FAULT	0 / 1
Bit 5	OVERRIDDEN	0 (fixed)
Bit 4	OUT_OF_SERVICE	0 / 1
Bits 3 to 0	Reserved	Undefined

COV_Increment

This property represents the minimum amount of change in Present_Value.

COV notifications are sent when the value of Present_Value for the COV notification that was last sent changes to a value that is greater than or equal to the value set with COV_Increment. COV_Increment cannot be read out to device address. Set the initial value with WindLDR or WindO/I-NV4.

Priority_Array

Priority_Array is a read-only property representing the array that stores the priority values.

Of the 16 elements in the array (element 1 to element 16), the value stored in Priority_Array with the smallest index number out of all non-NULL(00h) values is used as the value of Present_Value. If all values stored in Priority_Array are NULL(00h), Relinquish_Default is used as Present_Value.

Relinquish_Default

Relinquish_Default is the default value used as Present_Value when all values stored in Priority_Array are NULL(00h).

Polarity

This property represents the relationship between the physical state of the input/output and the logical state indicated by Present_Value in the Binary Input and Binary Output objects.

Polarity	Physical State of Input/Output	Present_Value	Physical State of Device
NORMAL	OFF/ INACTIVE	INACTIVE	Not running
NORMAL	ON/ ACTIVE	ACTIVE	Running
REVERSE	ON/ ACTIVE	INACTIVE	Not running
REVERSE	OFF/ INACTIVE	ACTIVE	Running

Out_Of_Service

Out_Of_Service is the property that represents whether or not Present_Value and the physical input/output have been unbound.

Properties	Value	Logical Value	Condition
Out_Of_Service	0	FALSE	In service (Present_Value and the physical input/output are bound.)
	1	TRUE	Out of service (Present_Value and the physical input/output are unbound.)

Out_Of_Service can be read out to an internal relay, and the status of an internal relay can be written as Out_Of_Service. The bit device that can be set depends on the main unit type.

Internal Device Name		Symbol	FC6A	FT2J/1J	HG2J/1J	HG5G/4G/3G/2G-V
Internal relay		M	Yes	No	No	No
HMI Devices	HMI Internal Relay	LM	No	Yes	Yes	No
	HMI Keep Relay	LK	No	Yes	Yes	No
	HMI Temporary Relay	LBM	No	Yes	Yes	No
Control Devices	Internal relay	M	No	Yes	No	No
		#M	No	No	No	Yes

For details, see the following manuals.
FC6A: Chapter 6 "Devices" in the "FC6A Series MICROSmart User's Manual".
FT2J/1J, HG2J/1J, HG5G/4G/3G/2G-V: Chapter 35 "1 Bit Devices" in the "WindO/I-NV4 User's Manual".

Out_Of_Service = TRUE is used for simulations.

Reliability

This property represents the reliability of the object property.

The following table shows the definition of Reliability for each object type.

Yes: Valid, No: Invalid

Definition	Value	Analog Input	Analog Output	Analog Value	Binary Input	Binary Output	Binary Value
no-fault-detected	0	Yes	Yes	Yes	Yes	Yes	Yes
no-sensor	1	Yes	No	No	Yes	No	No
over-range	2	Yes	No	Yes	No	No	No
under-range	3	Yes	No	Yes	No	No	No
open-loop	4	Yes	Yes	No	Yes	Yes	No
shorted-loop	5	Yes	Yes	No	Yes	Yes	No
no-output	6	No	Yes	No	No	Yes	No
unreliable-other	7	Yes	Yes	Yes	Yes	Yes	Yes
process-error	8	No	No	No	No	No	No
multi-state-fault	9	No	No	No	No	No	No
configuration-error	10	No	No	No	No	No	No
-- enumeration value 11 is reserved for a future addendum	11	Yes	Yes	No	Yes	Yes	Yes
communication-failure	12	Yes	Yes	Yes	Yes	Yes	Yes
member-fault	13	No	No	No	No	No	No
monitored-object-fault	14	No	No	No	No	No	No
tripped	15	No	No	No	No	No	No

Reliability can be read out to a device address, and the value of a device address can be written as Reliability. The word device that can be specified depends on the main unit type.

Internal Device Name		Symbol	FC6A	FT2J/1J	HG2J/1J	HG5G/4G/3G/2G-V
Data Register		D	Yes	No	No	No
HMI Devices	HMI Data Register	LDR	No	Yes	Yes	No
	HMI Keep Register	LKR	No	Yes	Yes	No
	HMI Temporary Register	LBR	No	Yes	Yes	No
Control Devices	Data Register	D	No	Yes	No	No
		#D	No	No	No	Yes

For details, see the following manuals.
FC6A: Chapter 6 "Devices" in the "FC6A Series MICROSmart User's Manual".
FT2J/1J, HG2J/1J, HG5G/4G/3G/2G-V: Chapter 35 "2 Word Devices" in the "WindO/I-NV4 User's Manual".

System_Status

Indicates the physical status and logical status of the main unit.

Parameter	Value
OPERATIONAL	0
OPERATIONAL_READ_ONLY	1
DOWNLOAD_REQUIRED	2
DOWNLOAD_IN_PROGRESS	3
NON_OPERATIONAL	4
BACKUP_IN_PROGRESS	5

System_Status of the main unit is fixed as OPERATIONAL.

Firmware_Revision

This property is set with the System Software Version of the main unit.

Application_Software_Version

Set information about the application, such as the modification date of the created project data, as a fixed string with WindLDR or WindO/I-NV4.

Protocol_Services_Supported

This property represents the types of services that are supported by the main unit.

Protocol_Object_Types_Supported

This property represents the types of objects that are supported by the main unit.

Object_List

This property represents the list of created objects.

2: EtherNet/IP Communication

FC6A Plus FC6A All-in-One FT2J/1J HG2J/1J HG5G/4G/3G/2G-V

This chapter describes the EtherNet/IP communication function.

Overview

The main unit supports EtherNet/IP (scanner and adapter) communication.

EtherNet/IP stands for "Ethernet Industrial Protocol", which is an industrial multi-vendor network that uses Ethernet.

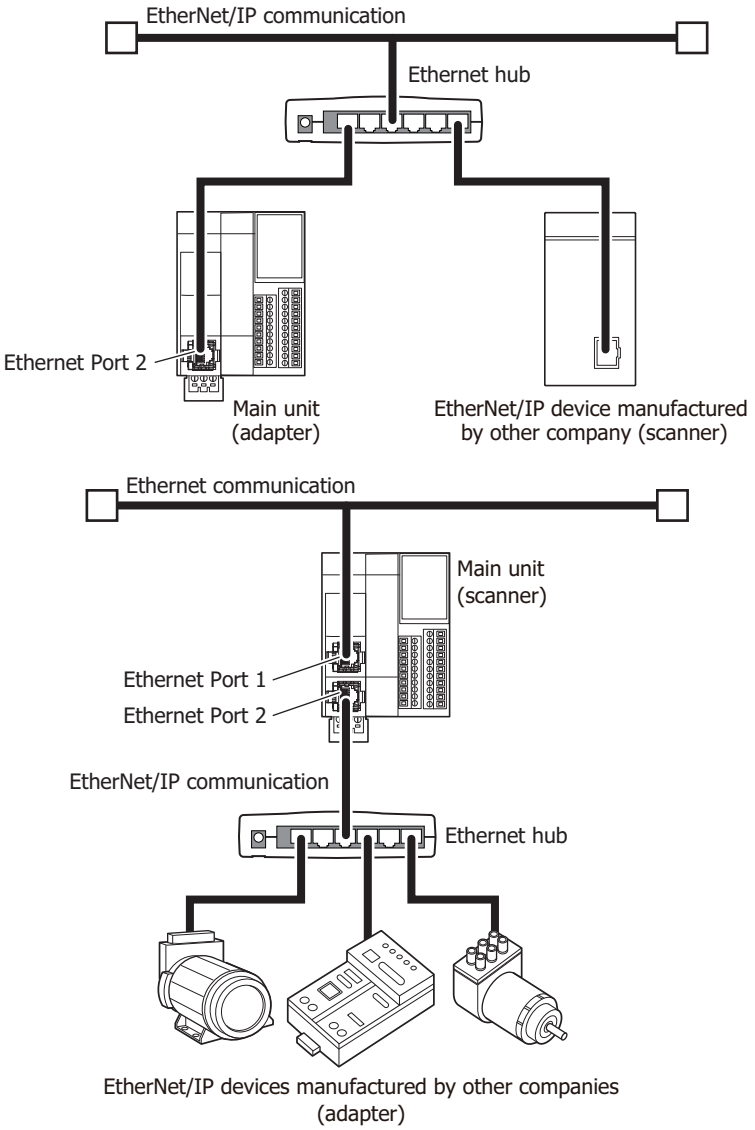
The main unit works as a scanner or adapter in EtherNet/IP communication and communication can be performed with devices that are compatible with EtherNet/IP communication.

EtherNet/IP communication also uses standard Ethernet technologies, which allows networks to be built that include various Ethernet-compatible devices.

- In EtherNet/IP communication, the master side is called the scanner and the slave side is called the adapter. In general, an EtherNet/IP device that has a function to be given a request to open a CIP connection is called an adapter, and an EtherNet/IP device that also combines a function to give a request to open a CIP connection is called a scanner.
- In this manual, a device that is compatible with EtherNet/IP communication is subsequently called an EtherNet/IP device.

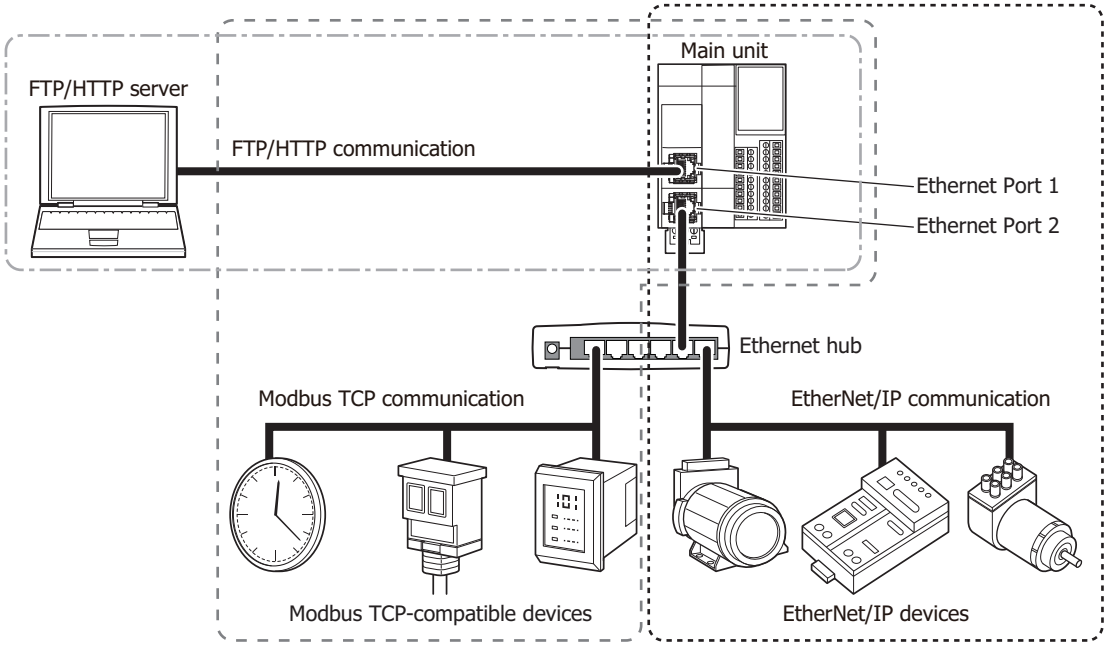
Multi-Vendor Network between EtherNet/IP Devices

You can build a multi-vendor network between EtherNet/IP devices.



Network That Mixes General-Purpose Ethernet Devices

You can build a network that mixes EtherNet/IP devices and general-purpose Ethernet devices.



When EtherNet/IP communication and other Ethernet communications are performed simultaneously on the same network, data may no longer be transmitted or received at the intended timing, depending on the load status. In this situation, build the network using an Ethernet switch that supports the QoS function and adjust the priority of data that flows on the network.

Specifications

Supported models

Supported IDEC's PLCs and programmable displays are as follows.

Series Name	Abbreviation	Type Number
MICROSmart	FC6A	FC6A-D16**CEE, FC6A-D32**CEE
SmartAXIS	FT2J	FT2J-7U22*AF-B
	FT1J	FT1J-4F1**AG-*
MICRO/I	HG2J	HG2J-7UT22TF-B
	HG1J	HG1J-4FT22TG-*

Basic specifications

Model (Abbreviation)			FC6A	FT2J/1J	HG2J/1J
Interface			Ethernet Port 2	Ethernet Interface	
I/O Message Communication Function	Number of CIP connections		32*1		
	CIP connection points	Number of settings	8		
		Predefined	Instance ID 198 (for Input Only output) Instance ID 199 (for Listen Only output)		
	RPI (communication cycle)		10 to 10,000 ms (in 1 ms increments)		
	CIP connection type		Input Only/Exclusive Owner/Listen Only		
	Trigger of send		Cyclic/COS (change of state)*2		
	Maximum data size per CIP connection		504 bytes or 1,444 bytes*3, *4		
	I/O message communication allowable bandwidth	504 bytes/packet	600pps		
		1,444 bytes/packet	200pps		
	Multicast filter function*5		Available (IGMP client function)		
Explicit Message Communication Function	Class 3 (connection type)	Server	Number of CIP connections: 32*6		
		Client	Not supported		
	UCMM	Server	Number of simultaneous executions: 32		
		Client	Not supported		
EtherNet/IP Conformance Test			CT17 Compliant	CT20 Compliant	

*1 The maximum number of connections is 32 in total with the CIP connections used in the class 3 (connection type) explicit message communication function.

*2 Communication can be performed with EtherNet/IP devices that output data with the Change Of State (transmit data when there is a change of state) method.

*3 For 1,444 bytes, the EtherNet/IP device must support Large Forward Open (CIP option specification).

*4 Make the total data size within 10,240 bytes for one CIP connection.

*5 With IGMP client function (igmpv2), you can filter an unnecessary multicast packet using an Ethernet switch corresponding to IGMP Snooping.

*6 The maximum number of connections is 32 in total with the CIP connections used in I/O message communication function.

About EtherNet/IP

■**Supported Standards**

EtherNet/IP communication in the main unit supports the following standards.

- EtherNet/IP

■**Vendor ID**

The vendor ID is 159.

■**Device Profile**

EtherNet/IP communication in the main unit supports the scanner and adapter.

■**Objects**

EtherNet/IP communication in the main unit supports the following objects.

Object Name	Class Code
Identity	01H
Message Router	02H
Assembly	04H
Connection Manager	06H
TCP/IP Interface	F5H
Ethernet Link	F6H
LLDP Management*1	109H

*1 FT2J/1J and HG2J/1J only

For details on objects, see "Objects" on page 2-45.

Function

The main unit provides the following functions as an EtherNet/IP device scanner or adapter.

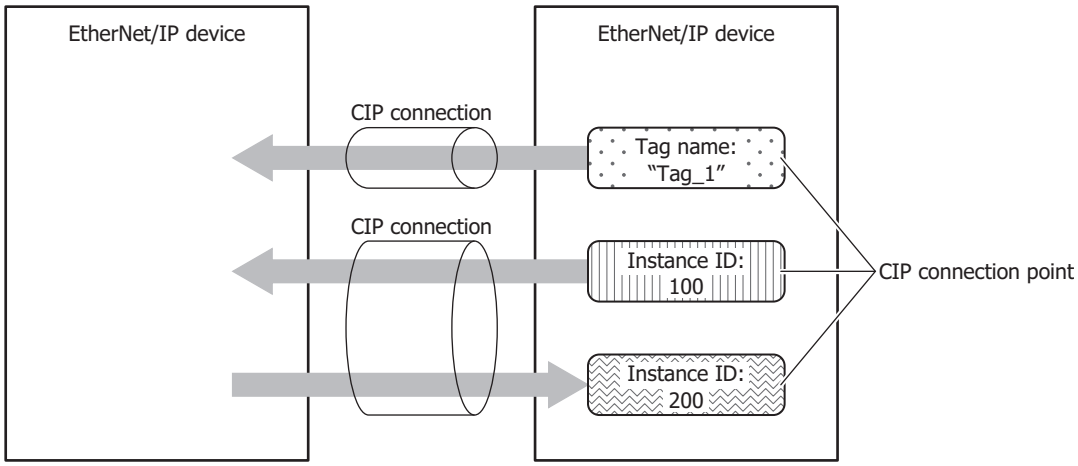
- I/O message communication (target) function (page 2-9)
- I/O message communication (originator) function (page 2-10)
- Class 3/UCMM Communication (page 2-11)
- Data and device binding function (page 2-12)

I/O Message Communication

What is I/O Message Communication?

I/O message communication is a function that allows EtherNet/IP devices to cyclically read and write data with each other. The target unit for reading/writing data is called a CIP connection point. CIP tag names and instance IDs are assigned to CIP connection points. In I/O message communication, an EtherNet/IP device reads/writes data by specifying the CIP tag name or instance ID of a CIP connection point.

Conceptual Diagram of I/O Message Communication



What is a CIP Connection?

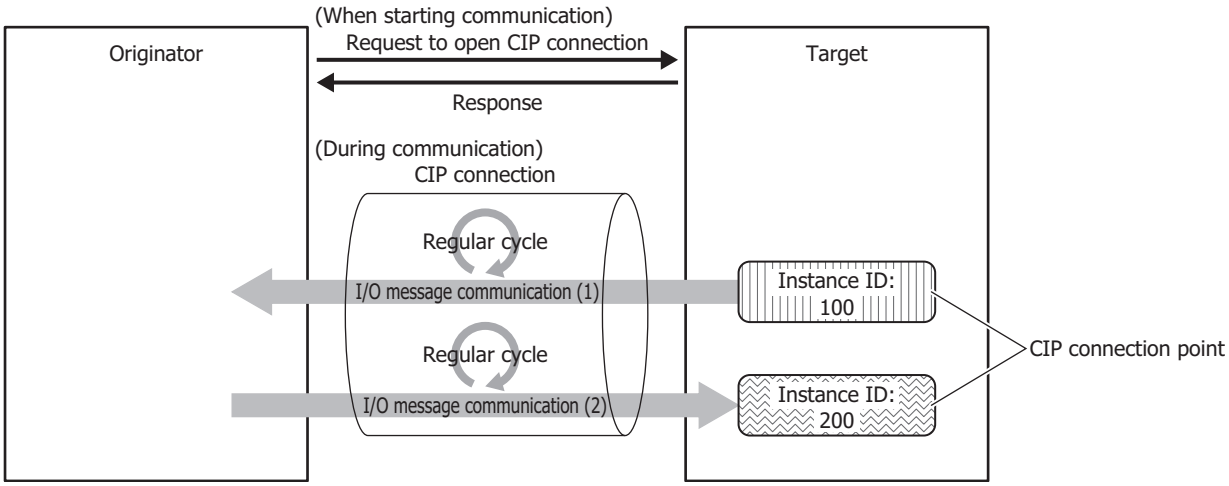
A CIP connection is a virtual communication line between EtherNet/IP devices. In I/O message communication, one device sends a request to open a CIP connection to the CIP connection point of the other device, and when successful, data is read and written. The side that sends the request to open the CIP connection is called the originator. The side that receives the request is called the target. The main unit can be the originator and the target. The main unit can open 32 CIP connections at the same time.

Types of CIP Connections

There are the following three types of CIP connections. Exclusive Owner can transmit and receive data. Input Only and Listen Only can only receive data.

(1) Exclusive Owner

This CIP connection cyclically transmits data from the target to the originator (1) and cyclically transmits data from the originator to the target (2). The CIP connection point is specified as a CIP tag name or an instance ID. Communication*1 is performed at the cycle (RPI) specified when the CIP connection is opened. Data can be transmitted and received when the originator opens the connection*2.

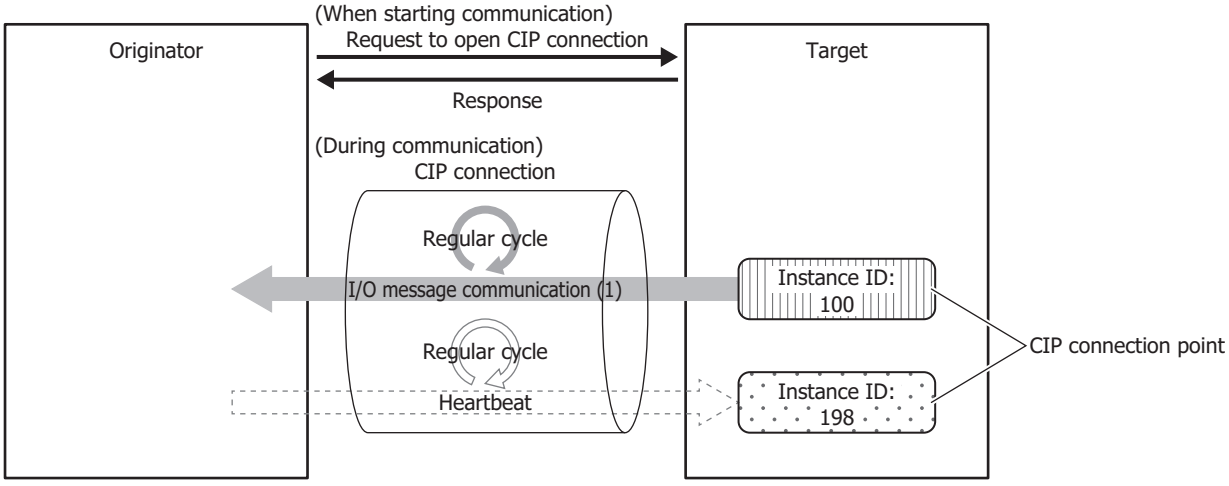


*1Class 1 communication (cyclic communication) is performed.

*2The originator requests the Forward Open service in Class 3 communication and opens a connection.

(2) Input Only

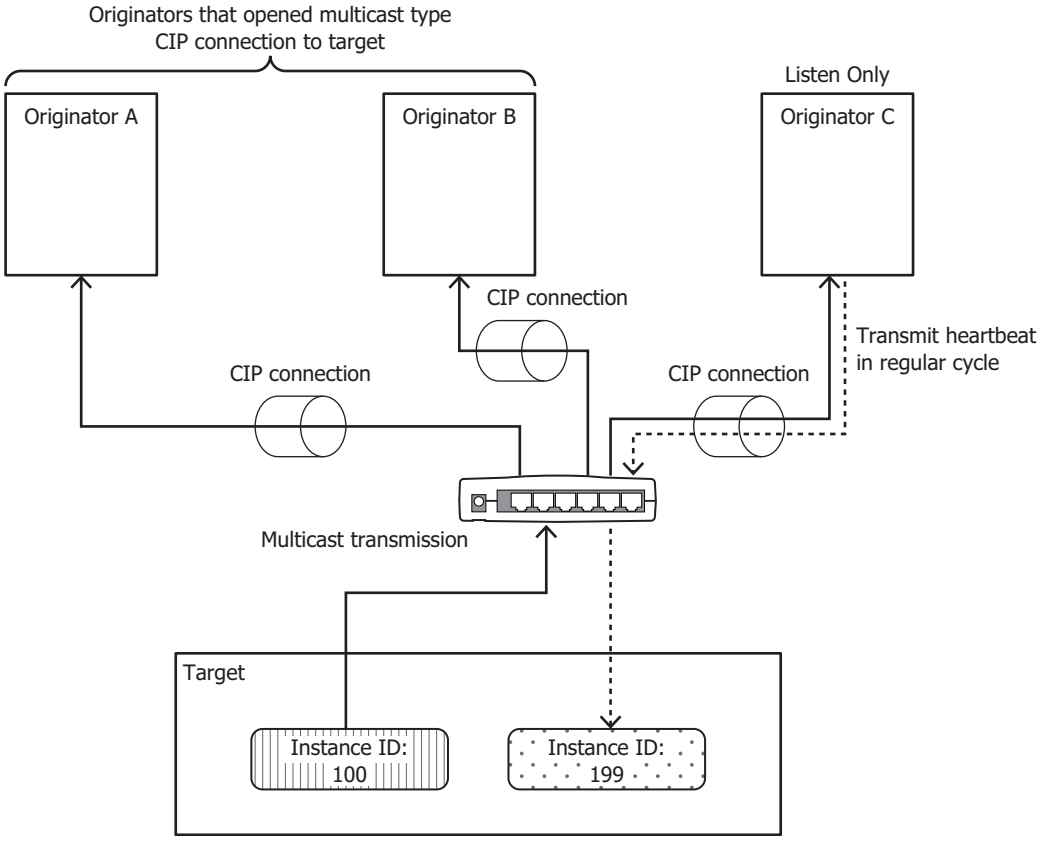
This CIP connection cyclically transmits data from the target to the originator (1). The CIP connection point is specified as a CIP tag name or an instance ID. Communication is performed at the cycle (RPI) specified when the CIP connection is opened. In order for the target to perform alive monitoring of the originator, a heartbeat that does not include data is cyclically transmitted from the originator to the target. The originator specifies the Input Only instance ID (198) on the target as the destination for the heartbeat. The heartbeat is transmitted at the specified cycle (RPI).



- For Input Only, when setting the CIP connection point of the input data to receive from the target as an instance ID, set the CIP connection point of the heartbeat to transmit to the target as an instance ID too.
- For Input Only, when setting the CIP connection point of the input data to receive from the target as a CIP tag name, a CIP tag name of the CIP connection point of the output data to transmit to the target does not exist and is not set.
- The heartbeat is always performed in Input Only.

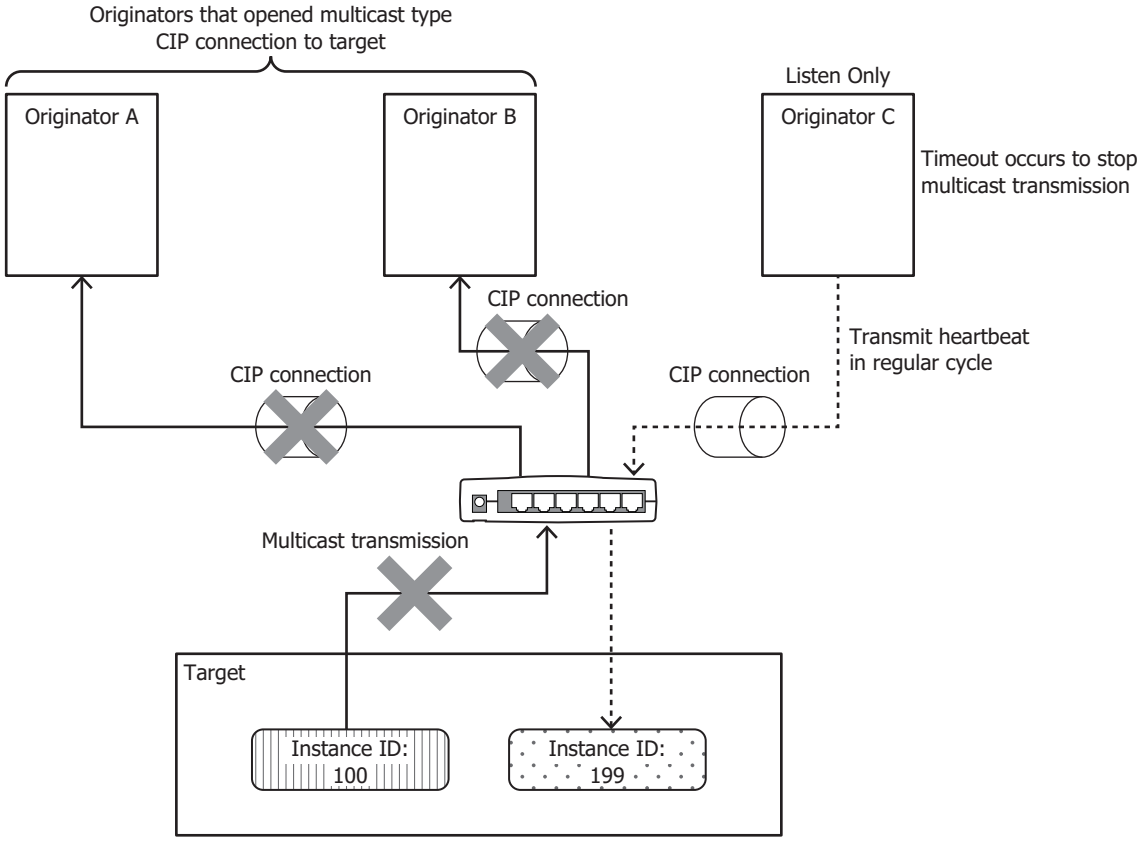
(3) Listen Only

This CIP connection cyclically receives data for originators from a target that already has a multicast type CIP connection using Exclusive Owner or Input Only opened. Set this connection type to simultaneously receive data when the target transmits that data by multicast to other originators. The CIP connection point is specified as an instance ID. Communication is performed at the cycle (RPI) specified when the CIP connection is opened. In order for the target to perform alive monitoring of the originator, a heartbeat that does not include data is cyclically transmitted from the originator to the target. The originator specifies the Listen Only instance ID (199) on the target as the destination for the heartbeat. The heartbeat is transmitted at the specified cycle (RPI).



- For Listen Only, when setting the CIP connection point of the input data to receive from the target as an instance ID, set the CIP connection point of the output data to transmit to the target as an instance ID too.
- The heartbeat is always performed in Listen Only.
- You can use an Ethernet switch that supports IGMP snooping to filter unnecessary multicast packets. In this case, please install the originator that supports the IGMP client function and IGMP Queria (routers, etc.) to periodically send the IGMP query. Without IGMP Queria, the originator will not be able to receive multicast packets after a period of time.

A Listen Only CIP connection can be opened only when a multicast type CIP connection is opened. When the CIP connection between the target and originator that opened a multicast type CIP connection using Exclusive Owner or Input Only is disconnected, the multicast transmissions from the target to originators that opened Listen Only CIP connections are stopped.



Transmit Trigger Type

There are two types of transmission triggers for main unit.

■**Cyclic**

Send data from the originator to the target or from the target to the originator at every regular cycle in the RPI.

■**Change Of State**

Send data when there is a change from originator to target or from target to originator. If there is no change in data, it will send data at every regular cycle in the RPI.

Types of connections

There are two types of main unit with the following connection types.

■**Point To Point**

Send data from the originator to the target or from the target to the originator by one-to-one.

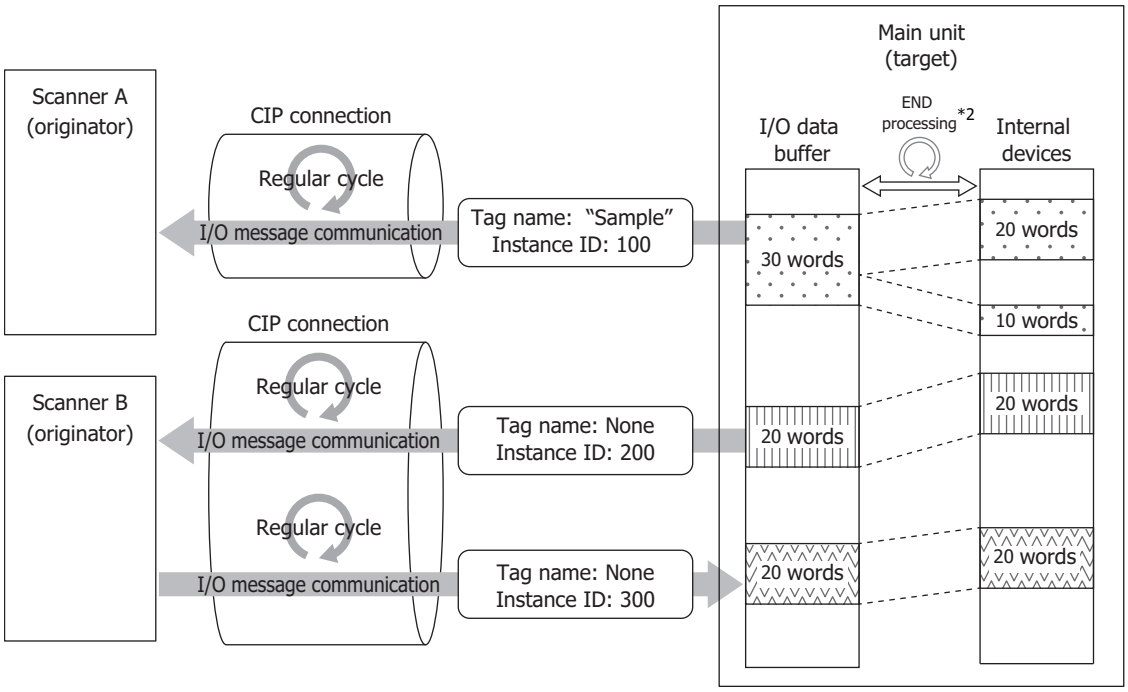
■**Multicast**

Sends data from the target to the multicast addresses of multiple originators. The direction of the target from the originator is not supported.

I/O Message Communication (Target) Function

A function where the main unit is the target with CIP connection points and originators cyclically read and write data. A maximum of 8 CIP connection points can be defined.

The originator sends a request to open a CIP connection to the CIP connection point of the main unit, and when successful, the CIP connection point data is read and written. The main unit first writes the values of internal devices to the I/O data buffer*1 and then transmits the data to the originator. Data received from the originator is also first written to the I/O data buffer and then written to internal devices. The main unit performs the process of reflecting the I/O data buffer and the internal devices mutually in the END processing*2.

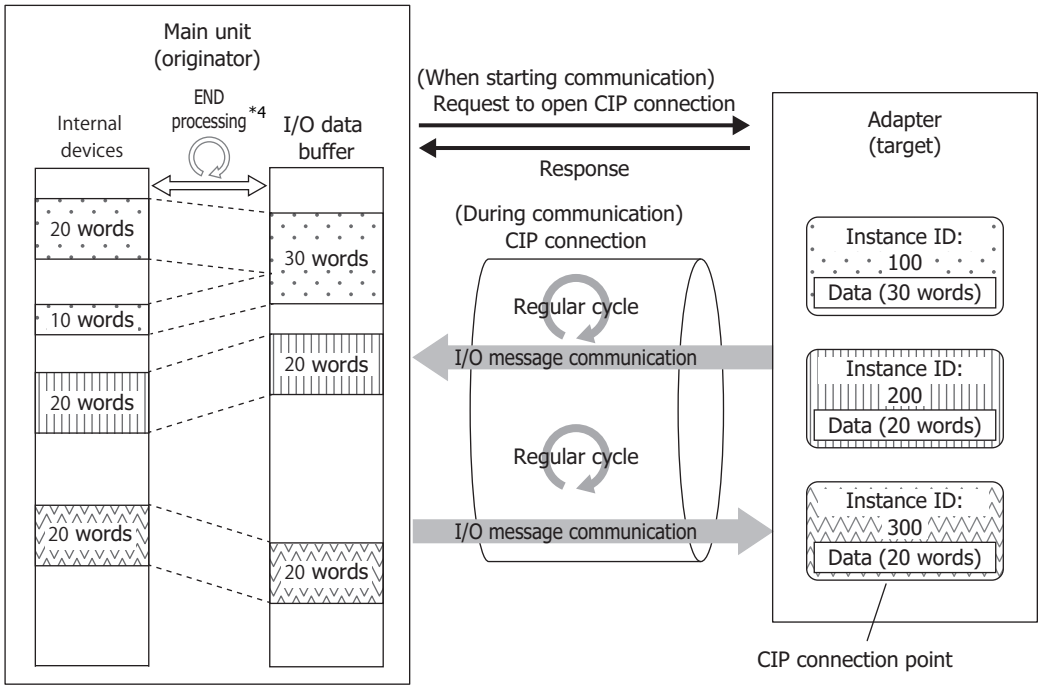


*1 The I/O data buffer is a buffer inside the main unit that stores the data to transmit and receive in I/O message communication. For details on the I/O data buffer and internal device binding, see "Data and Device Address Binding Function" on page 2-12.

*2 For the FC6A, the processing to reflect data between the I/O data buffer and internal devices is performed in the END processing of the ladder program.
For the FT2J/1J, the processing to reflect data between the I/O data buffer and HMI devices is performed in the END processing of the HMI function. For the control device, see Chapter 1 "1.1 Overview" in the "WindO/I-NV4 User's Manual".
For the HG2J/1J, the processing to reflect data between the I/O data buffer and internal devices is performed in the END processing of the HMI function.

I/O Message Communication (Originator) Function

A function where the main unit cyclically reads and writes data of the CIP connection points of targets. When the EtherNet/IP Communication Bit (M8460/LSM87) is turned on, the main unit sends a request to open the CIP connections to the CIP connection point of targets registered to the scan list*1. When successful*2, the CIP connection point data is read and written. The main unit first writes the data received from the target to the I/O data buffer*3 inside the main unit and then writes the data to internal devices. The values of internal devices are first written to the I/O data buffer and then transmitted to the target. The main unit performs the process of reflecting the I/O data buffer and the internal devices mutually in the END processing*4.



*1 The scan list is a list of CIP connections the main unit opens as the originator. The list of configured targets or CIP connections is displayed in the EtherNet/IP tree area on the **EtherNet/IP Settings** dialog box. For details, see "EtherNet/IP Tree Area" on page 2-22.

*2 If opening a CIP connection fails, opening the CIP connection continues being retried until successful. During I/O message communication, if I/O messages from the target cannot be received even after the specified timeout elapses, a new CIP connection is automatically reopened. If opening the new CIP connection fails, opening the CIP connection continues being retried until successful.

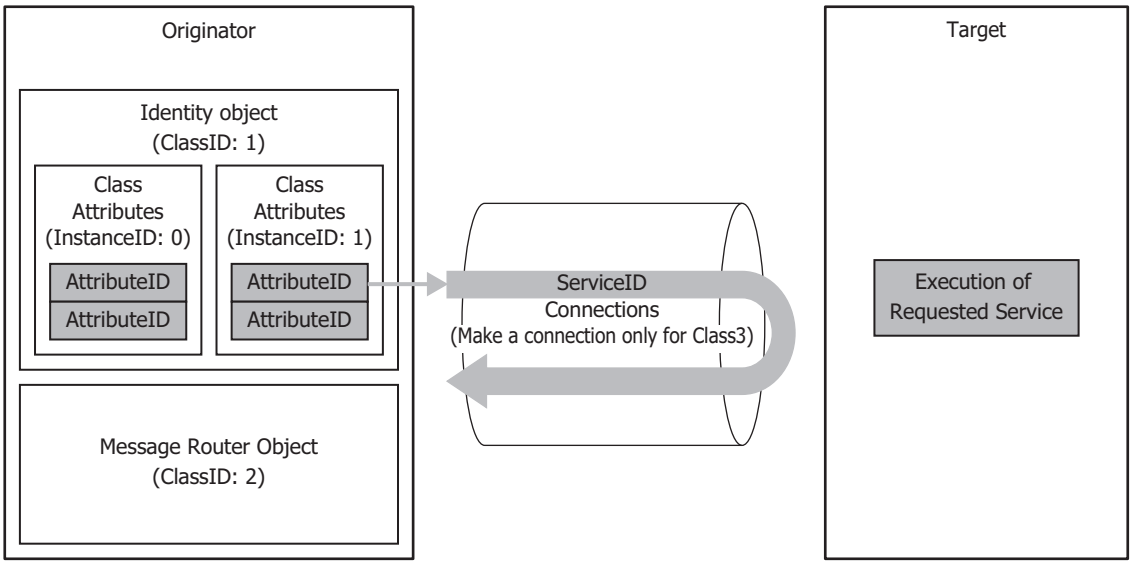
*3 The I/O data buffer is a buffer inside the main unit that stores the data to transmit and receive in I/O message communication. For details on the I/O data buffer and internal device binding, see "Data and Device Address Binding Function" on page 2-12.

*4 For the FC6A, the processing to reflect data between the I/O data buffer and internal devices is performed in the END processing of the ladder program.
For the FT2J/1J, the processing to reflect data between the I/O data buffer and HMI devices is performed in the END processing of the HMI function. For the control device, see Chapter 1 "1.1 Overview" in the "WindO/I-NV4 User's Manual".
For the HG2J/1J, the processing to reflect data between the I/O data buffer and internal devices is performed in the END processing of the HMI function.

Class 3/UCMM Communication

Communication in which the target executes and responds to the service requested by the originator.
There are Class3 communication that opens the connection and UCMM communication that does not open.
The method to perform the service is to specify the CLassID, InstanceID and AttributeID to request the assigned service to the target in the access method that corresponds to the ServiceID.

- Get_Attribute_Single(0EH): Get target data using one Attribute in Instance
- Get_Attribute_All(01H): Get target data using all Attribute in Instance
- Set_Attribute_Single(10H): Use one Attribute in Instance to populate the target



Service List

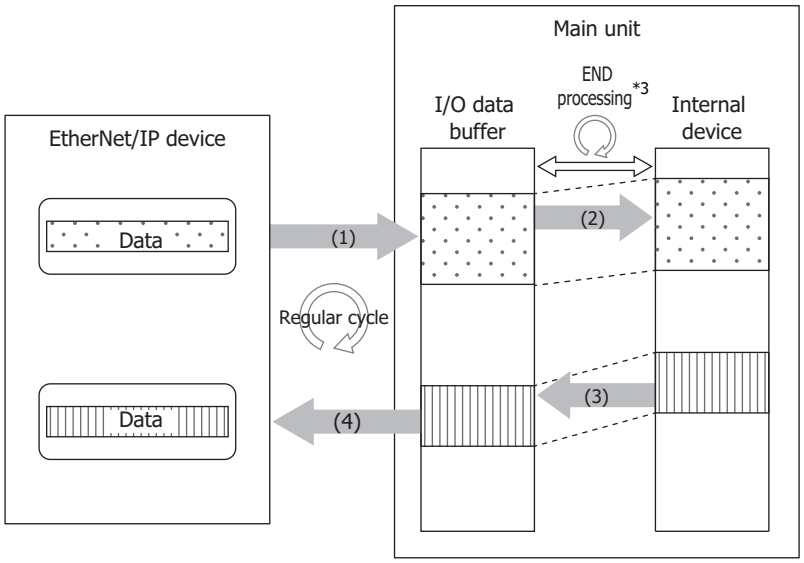
ClassID	InstanceID	AttributeID
Identity Object (01H)	Class Attributes (0)	Get_Attribute_Single (0EH)
		Get_Attribute_All (01H)
	Instance attributes (1)	Get_Attribute_Single (0EH)
		Get_Attribute_All (01H)
		Reset (05H)
Message Router Object (02H)	Class Attribute (0)	Get_Attribute_Single (0EH)
	Instance attribute (1)	Get_Attribute_Single (0EH)
Assembly Object (04H)	Class Attribute (0)	Get_Attribute_Single (0EH)
	Instance attributes (100, 200)	Get_Attribute_Single (0EH)
		Set_Attribute_Single(10H)
Connection Manager Object (06H)	Instance attributes (1)	Forward_Open(54H)
		Forward_Close(4EH)
TCP/IP Interface Object (F5H)	Class Attribute (0)	Get_Attribute_Single (0EH)
	Instance attributes (1)	Get_Attribute_Single (0EH)
		Set_Attribute_Single(10H)
Ethernet Link Object (F6H)	Class Attribute (0)	Get_Attribute_Single (0EH)
	Instance attribute (1)	Get_Attribute_Single (0EH)
LLDP Management Object (109H)*1	Class Attribute (0)	Get_Attribute_Single (0EH)
	Instance attributes (1)	Get_Attribute_Single (0EH)
		Set_Attribute_Single (10H)

*1 FT2J/1J and HG2J/1J only

Data and Device Address Binding Function

The main unit assigns IN data*1 and OUT data*2 set on the scan list and in CIP connection points to the I/O data buffer in the main unit.

The main unit first writes the data received from the other device in I/O message communications to the I/O data buffer (1) and then writes the data to internal devices (2). The values written to internal devices are also first written to the I/O data buffer (3) and then transmitted to the other device (4). The processing to reflect data between the I/O data buffer and internal devices is performed in the END processing*3.

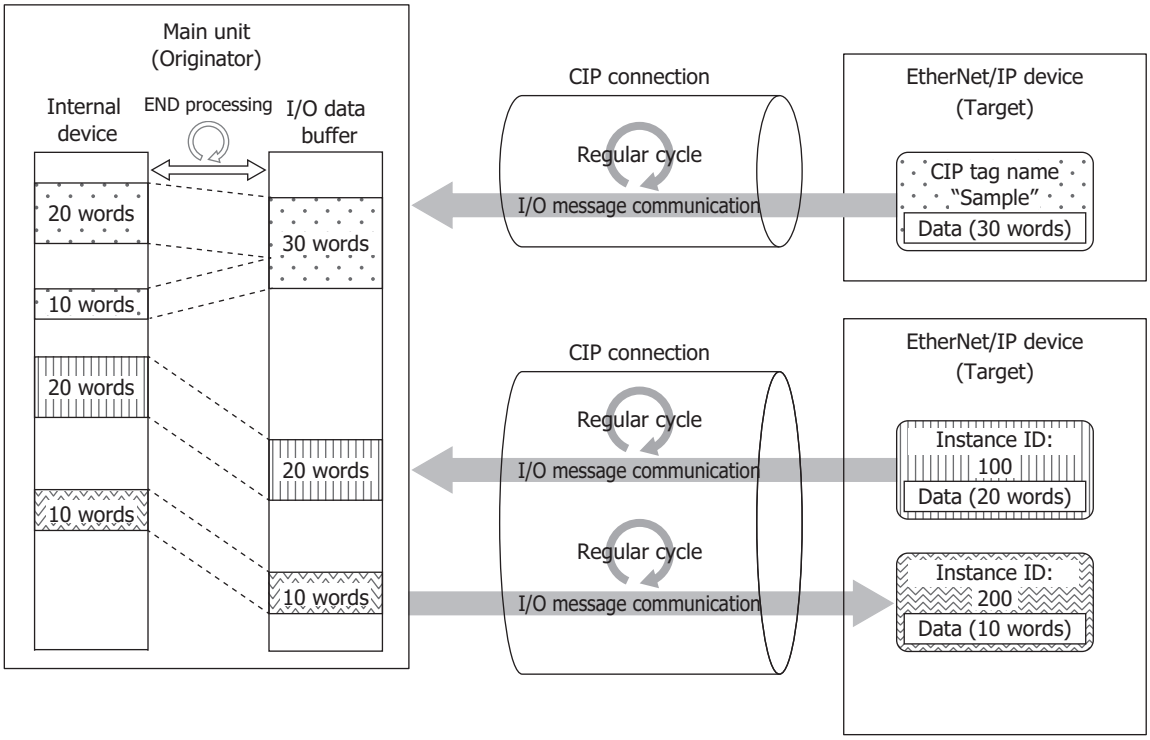


*1 IN data is received by the originator from the target in I/O message communication.

*2 OUT data is transmitted to the target from the originator in I/O message communication.

*3 For the FC6A, the processing to reflect data between the I/O data buffer and internal devices is performed in the END processing of the ladder program.
For the FT2J/1J, the processing to reflect data between the I/O data buffer and HMI devices is performed in the END processing of the HMI function. For the control device, see Chapter 1 "1.1 Overview" in the "WindO/I-NV4 User's Manual".
For the HG2J/1J, the processing to reflect data between the I/O data buffer and internal devices is performed in the END processing of the HMI function.

For example, to have the main unit (originator) cyclically read and write the CIP connection point data of the target, the data received from the target is first written to the I/O data buffer and then written to the device addresses. The values of device addresses are also first written to the I/O data buffer and then transmitted to the target.



Refresh Limit Count

FC6A Plus | FC6A All-in-One | FT2J/1J | HG2J/1J | HG5G/4G/3G/2G-V

The size of data that can be reflected between the I/O data buffer and data registers in END processing one time is a maximum of 512 words. This size can be set as the refresh limit count between 1 and 512 words. For example, if the refresh limit count is set to 256 words, a maximum of 256 words of data are reflected from the I/O data buffer to device addresses and a maximum of 256 words of data are reflected from device addresses to the I/O data buffer in END processing one time. When IN data and OUT data are set that exceeded the refresh limit count, the processing is split into multiple portions of END processing. When the refresh limit count is increased, the amount of data that is reflected in the END processing increases, so the scan time of the ladder program becomes longer. Adjust the value according to the target system.

I/O message communication is performed more frequently. If the values received from an EtherNet/IP device change in an amount of time shorter than the scan time, those changes may not be reflected in the device addresses.

EtherNet/IP Communication Operations

EtherNet/IP communication is enabled or disabled depending on the status of EtherNet/IP Communication Bit (M8460/LSM87).

Special Devices

The special devices used in EtherNet/IP communication differ depending on the main unit type.

Internal Device Name		FC6A	FT2J/1J	HG2J/1J
Special Internal Relay		Yes	No	No
Special Data Register		Yes	No	No
HMI Device	HMI Special Internal Relay	No	Yes	Yes
	HMI Special Data Register	No	Yes	Yes

R/W is an abbreviation for read/write.
The notation for the R/W field is as follows.
R/W: The device can be both read from and written to
R: Read-only
W: Write-only

■Special Internal Relay, HMI Special Internal Relay

Special Internal Relay	HMI Special Internal Relay	Description		R/W
M8460	LSM87	EtherNet/IP Communication Bit	This special internal relay permits or prohibits EtherNet/IP communication. OFF: Prohibit EtherNet/IP communication ON: Permit EtherNet/IP communication	R/W

When EtherNet/IP Communication Bit (M8460/LSM87) is turned from off to on, the main unit enables the EtherNet/IP communication function. When EtherNet/IP Communication Bit (M8460/LSM87) is turned off, the main unit disables the EtherNet/IP communication function.

To resume EtherNet/IP communication when EtherNet/IP Operation Status (D8790/LSD410) is stopped by an error, follow the procedure below.
1. Turn off EtherNet/IP Communication Bit (M8460/LSM87) once.
   Wait for EtherNet/IP Operation Status (D8790/LSD410) to become stopped.
2. Turn on EtherNet/IP Communication Bit (M8460/LSM87).

EtherNet/IP Communication Bit (LSM87) for FT2J/1J and HG2J/1J

Switch from	Turn ON EtherNet/IP Communication bit automatically check box	Operating Mode	Switch to		
			Run Mode	Monitor Mode	Offline Mode
Switch from	OFF	Run Mode	—	LSM87 maintains its previous state.	(Operating mode cannot be switched)
		Monitor Mode	LSM87 maintains its previous state.	—	LSM87 turns off automatically.
		Offline Mode	(Operating mode cannot be switched)	LSM87 turns off automatically.	—
	ON	Run Mode	—	LSM87 maintains its previous state.	(Operating mode cannot be switched)
		Monitor Mode	LSM87 maintains its previous state.	—	LSM87 automatically turns off and then on.
		Offline Mode	(Operating mode cannot be switched)	LSM87 automatically turns off and then on.	—

■Special Data Register, HMI Special Data Register

Special Data Register	HMI Special Data Register	Description		R/W
D8790	LSD410	EtherNet/IP Operation Status	This special data register stores the operating status of EtherNet/IP communication. 0000h: Stopped 0100h: In preparation 0200h: Executing 0300h: Restarting 0F00h: Stopped by error	R
D8791	LSD411	EtherNet/IP Error Information	This special data register stores error information that has occurred in EtherNet/IP communication. 0000h: Normal 0800h: Timeout waiting for link up 8000h: Timeout waiting for communication to start 8100h: Timeout waiting for communication to stop	R

When the EtherNet/IP Operation Status (D8790/LSD410) is in preparation, the EtherNet/IP Error Information (D8791/LSD411) is reset to 0.

Basic Operation

This chapter describes the basic operation of EtherNet/IP communication.

■Starting and Stopping EtherNet/IP Communication

When the EtherNet/IP Communication Bit (M8460/LSM87) is changed from off to on, the main unit establishes CIP connections to the targets or CIP connection points registered to the scan list and starts EtherNet/IP communication (starts cyclic data exchange by I/O message communication).

For details, see "Special Internal Relay, HMI Special Internal Relay" on page 2-14.

When the EtherNet/IP Communication Bit (M8460/LSM87) is turned off, EtherNet/IP communication stops (I/O message communication stops and all CIP connections are disconnected).

Operation during Ladder Program RUN and STOP

FC6A Plus FC6A All-in-One FT2J/1J HG2J/1J HG5G/4G/3G/2G-V

The following table shows the status of EtherNet/IP communication while the ladder program is running or stopped.

RUN/STOP State of Ladder Program	EtherNet/IP Communication Bit (M8460) Status	EtherNet/IP Communication Status
STOP	OFF	Stopped
	ON	
RUN	OFF	
	OFF -> ON	Start
	ON	Executing
	ON -> OFF	End

When changing from run to stop, EtherNet/IP Communication Bit (M8460) is automatically turned off.

Operation when Switching Operating Modes

FC6A Plus FC6A All-in-One FT2J/1J HG2J/1J HG5G/4G/3G/2G-V

For FT2J/1J and HG2J/1J, the EtherNet/IP Communication Bit (LSM87) is automatically turned off when the operating mode is switched. After that, EtherNet/IP Communication Bit (LSM87) operate according to the setting of **Turn ON EtherNet/IP Communication bit automatically**. The status of EtherNet/IP communication (start or stop) follows the status of EtherNet/IP Communication Bit (LSM87).

However, in the following cases, EtherNet/IP Communication Bit (LSM87) is not be automatically turned off.

Operating Mode		EtherNet/IP Communication Bit (LSM87) Status	EtherNet/IP Communication Status
Switch from	Switch to		
Run Mode	Monitor Mode	The on or off state immediately before switching is maintained.	The previous EtherNet/IP communication state (enabled or disabled) is maintained.
Monitor Mode	Run Mode		
Run Mode/ Monitor Mode/ Offline Mode	System Mode	—	EtherNet/IP communication is disabled.

■Operation during EtherNet/IP Communication

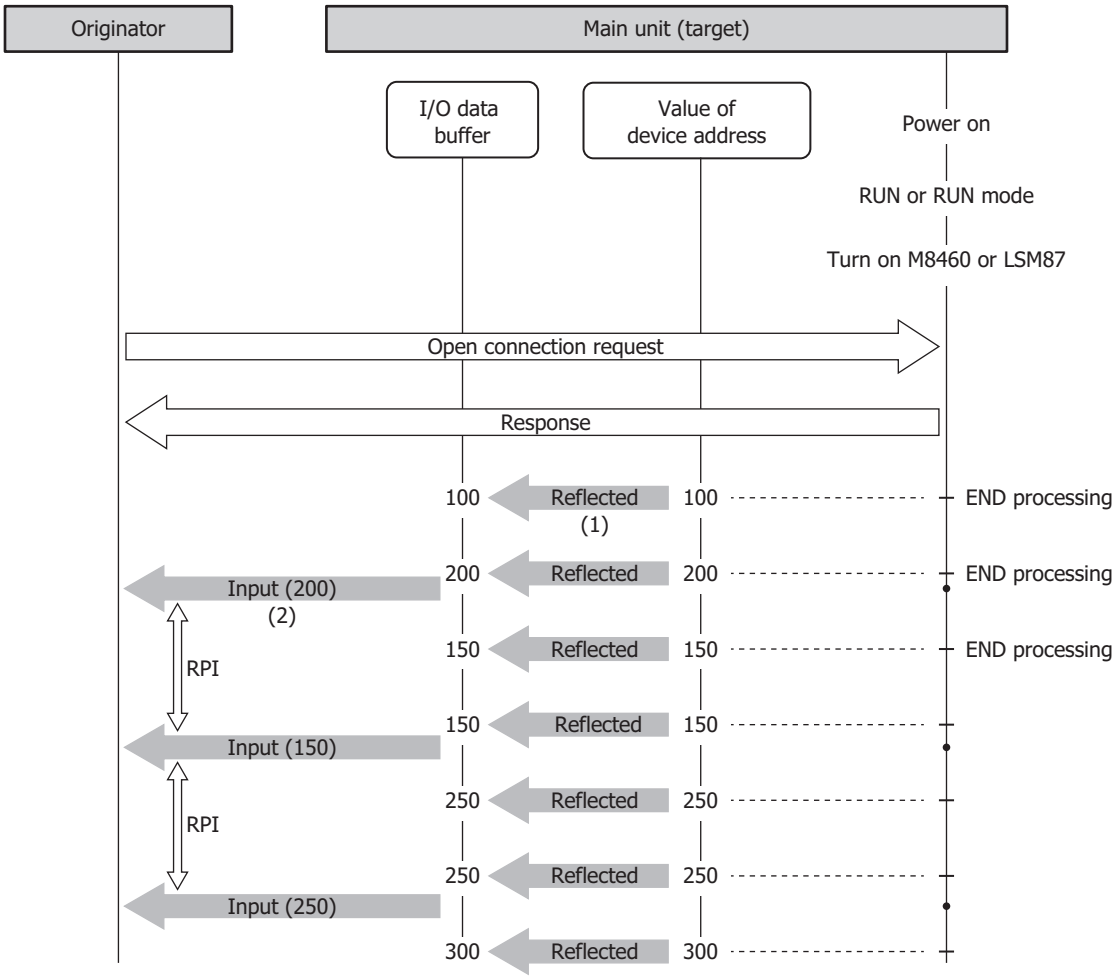
The I/O data buffer held in the main unit corresponds with device addresses and data is reflected between the two in the END processing. The processing to reflect data between the I/O data buffer and device addresses and I/O message communication between the main unit and EtherNet/IP device is not synchronized.

I/O message communication is performed more frequently. If the values received from an EtherNet/IP device change in an amount of time shorter than the scan time, those changes may not be reflected in the device addresses.

For the FC6A, the processing to reflect data between the I/O data buffer and internal devices is performed in the END processing of the ladder program.
For the FT2J/1J, the processing to reflect data between the I/O data buffer and HMI devices is performed in the END processing of the HMI function. For the control device, see Chapter 1 "1.1 Overview" in the "WindO/I-NV4 User's Manual".
For the HG2J/1J, the processing to reflect data between the I/O data buffer and internal devices is performed in the END processing of the HMI function.

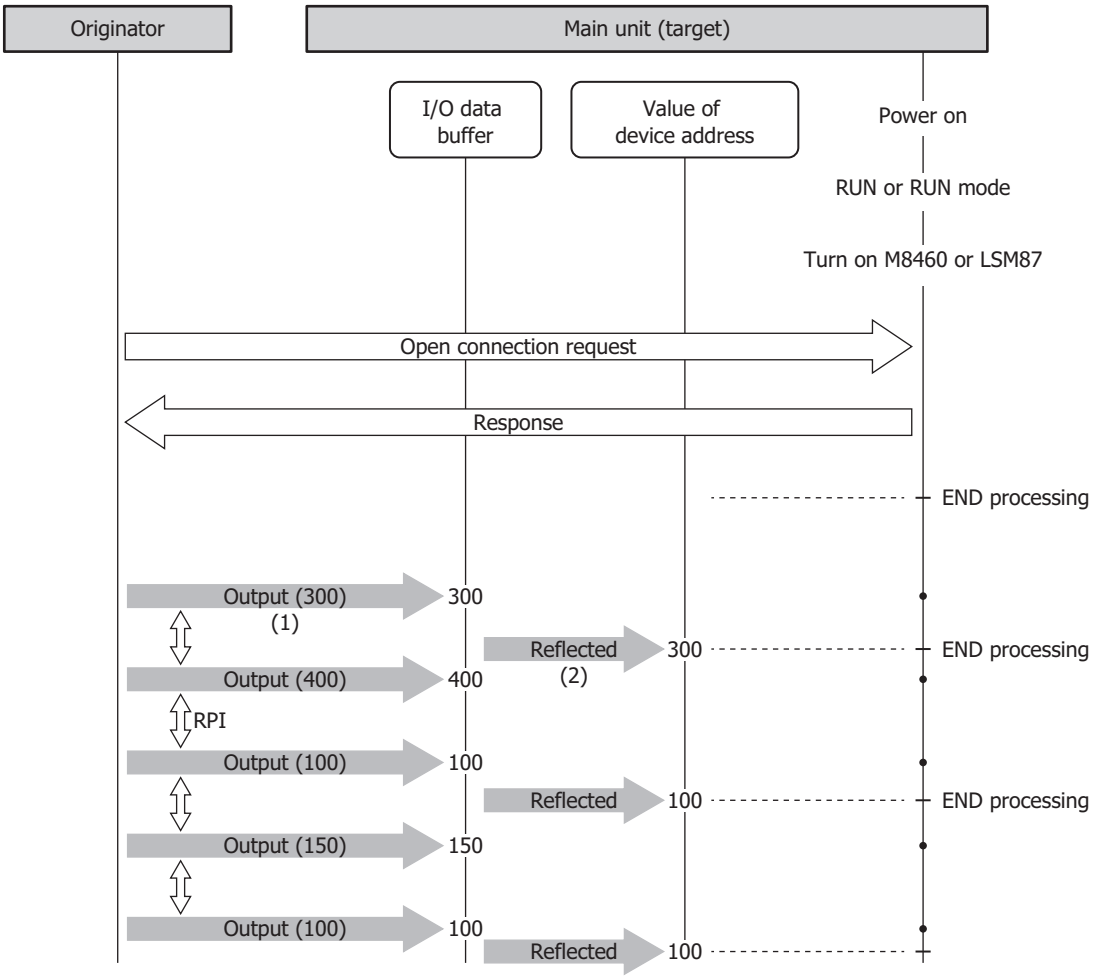
When the Main Unit (Target) Transmits Data to the Originator (Input)

When the value written to the device address is transmitted to the originator, the value written to the device address is reflected to the I/O data buffer in the END processing immediately before the data is transmitted (1). Then the data in the I/O data buffer is transmitted to the target in I/O message communication immediately afterward (2).



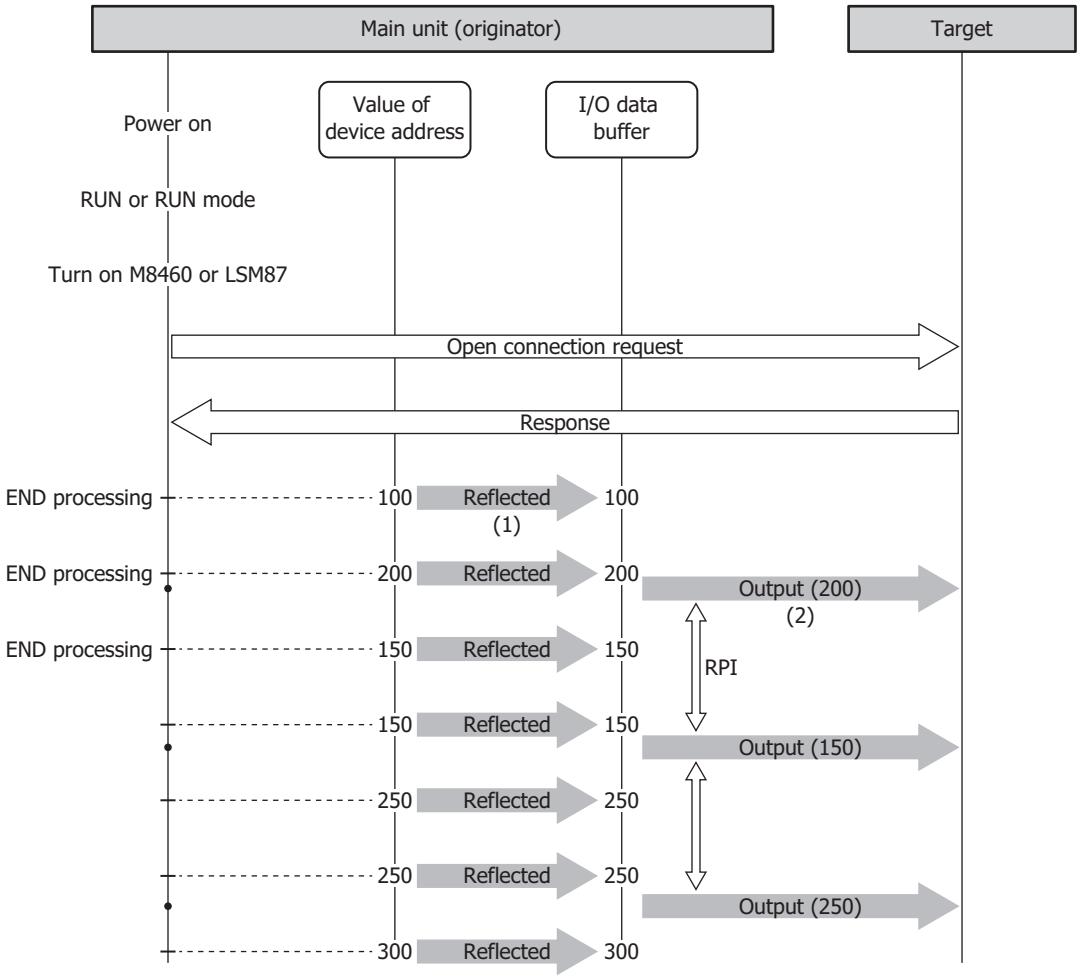
When the Main Unit (Target) Receives Data from the Originator (Output)

When the data received from the originator is written to the device address, the main unit writes the data received from the originator in I/O message communication to the I/O data buffer (1). Then the data in the I/O data buffer is reflected to the device address in the END processing immediately after the data was received (2).



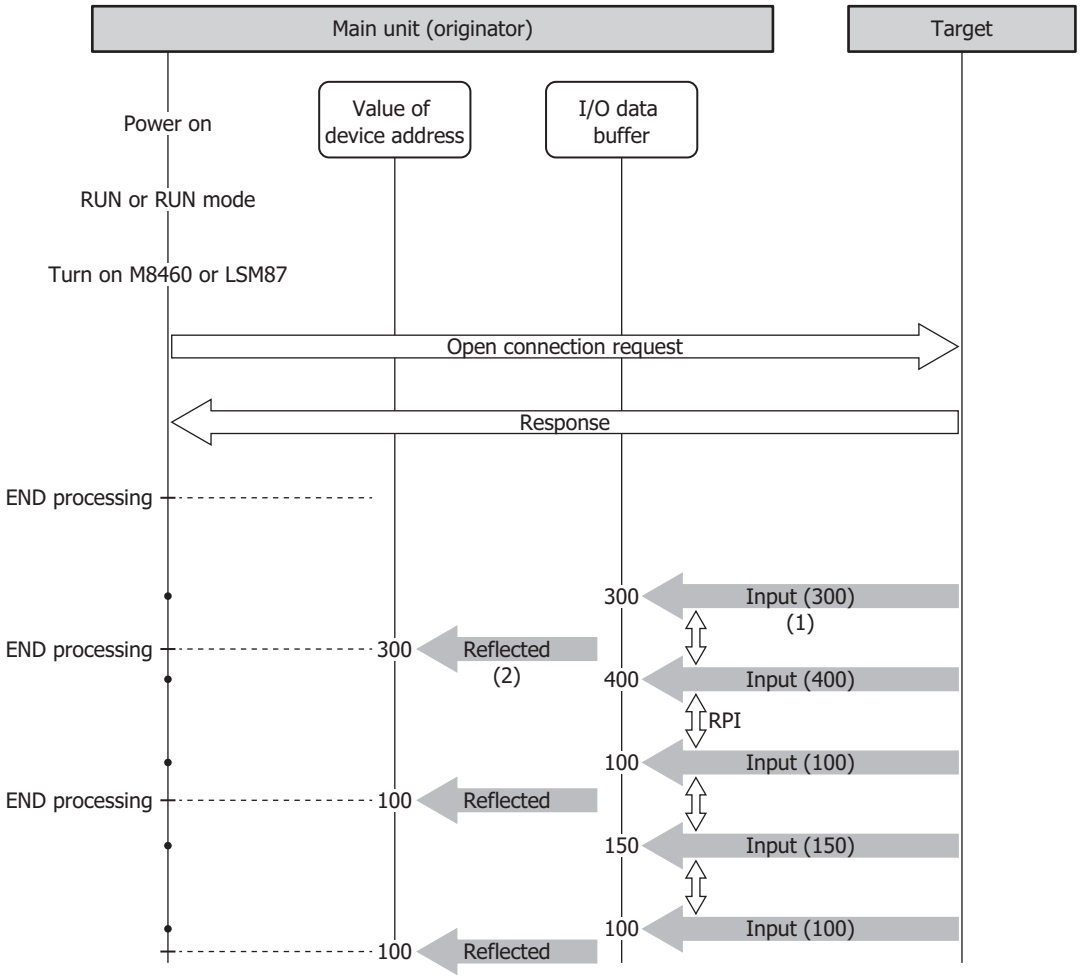
When the Main Unit (Originator) Transmits Data from the Target (Output)

When the value written to the device address is transmitted to the target, the value written to the device address is reflected to the I/O data buffer in the END processing immediately before the data is transmitted (1). Then the data in the I/O data buffer is transmitted to the target in I/O message communication immediately afterward (2).



When the Main Unit (Originator) Receives Data from the Target (Input)

When the data received from the target is written to the device address, the main unit writes the data received from the target in I/O message communication to the I/O data buffer (1). Then the data in the I/O data buffer is reflected to the device address in the END processing immediately after the data was received (2).



EtherNet/IP Settings Dialog Box

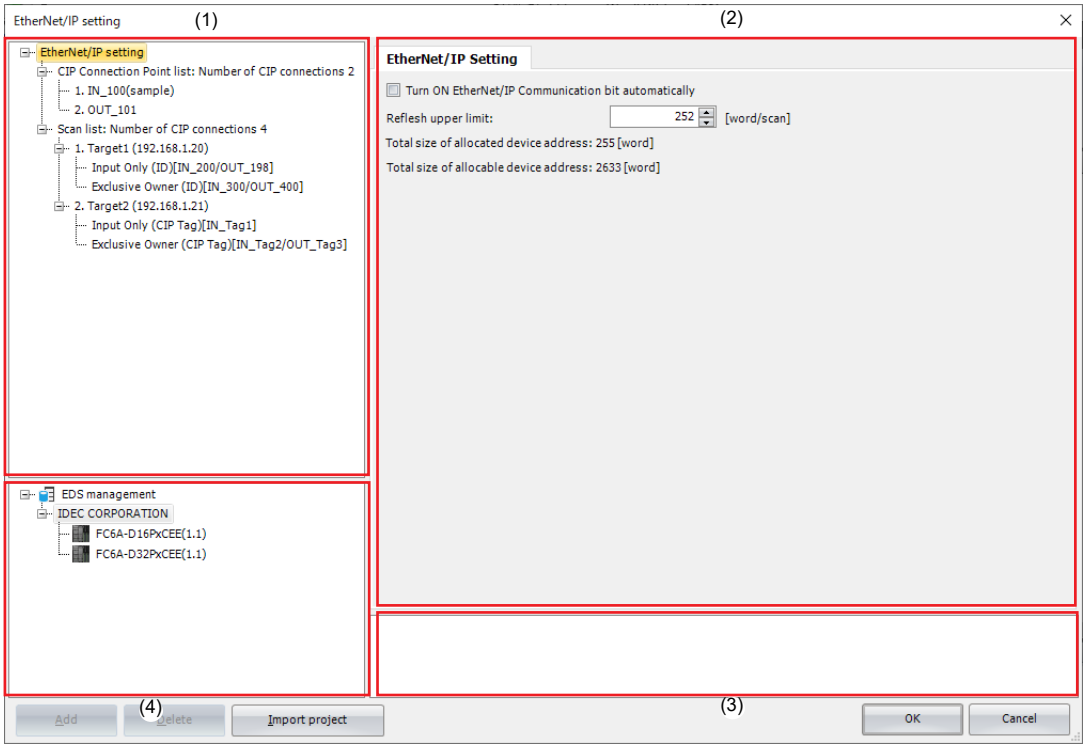
The **EtherNet/IP Settings** dialog box is composed of the following three areas.

(1) EtherNet/IP Tree Area (page 2-22)
This area displays the list of configured CIP connection points, targets, and CIP connections.

(2) Parameter Setting Area (page 2-24)
This area displays details on the node selected in the EtherNet/IP tree area.

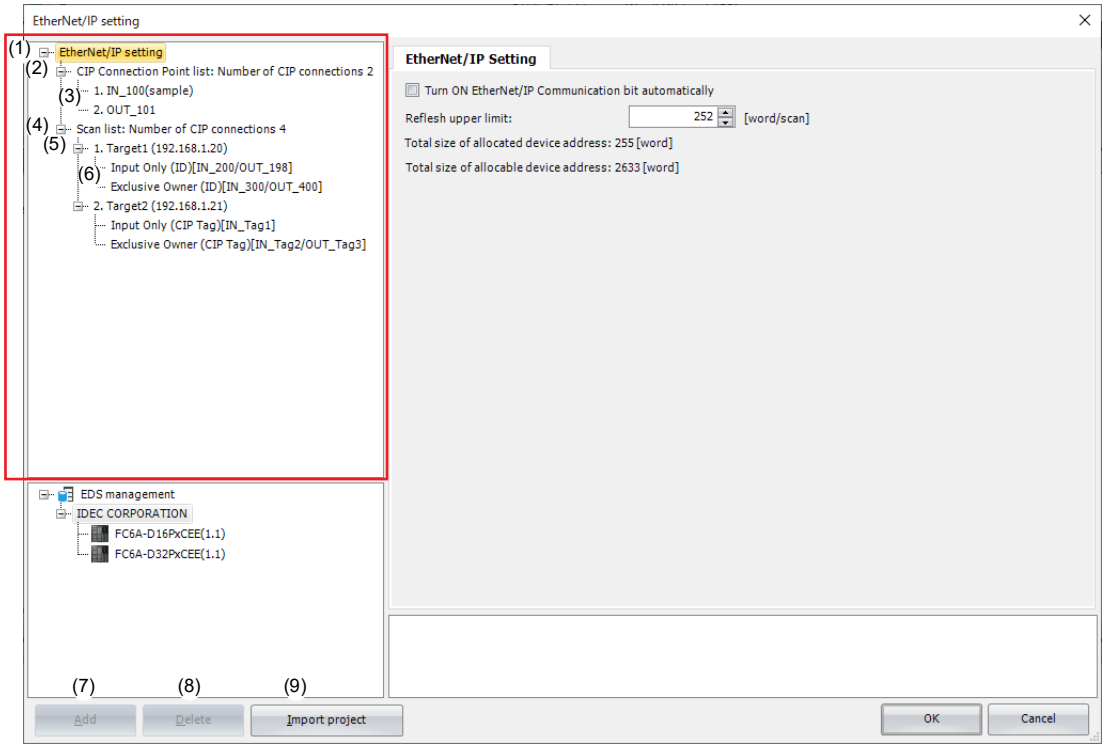
(3) Information View Area (page 2-34)
This area displays the details of errors if there are errors in content set on the **EtherNet/IP Settings** dialog box.

(4) EDS Management Area (page 2-35)
This area displays the list of EDS files that imported to your PC.



EtherNet/IP Tree Area

This area displays the list of configured CIP connection points, targets, and CIP connections.



(1) Ethernet Net/IP Settings node

When the **EtherNet/IP Settings** node is selected and expanded, the list of configured CIP Connection Point and Scan list is displayed.

(2) CIP Connection Point List node

Displays the number of CIP connection points. When the **CIP Connection Point List** node is expanded, the list of configured CIP connection points is displayed.

(3) CIP Connection Point node

Displays the CIP tag name and instance ID of a configured CIP Connection Point.

(4) Scan List node

Displays the number of CIP Connections. When the **Scan List** node is expanded, the list of configured targets is displayed.

(5) Target node

Displays the node name and IP address of a configured target. When a **Target** node is selected and expanded, the list of configured CIP connections is displayed.

(6) CIP Connection node

Displays the CIP connection name and connection point of a configured CIP connection.

(7) Add button

Adds a new node according to the node selected in the EtherNet/IP tree area. You can also add a node from the menu displayed when right-clicking the node.

(8) Delete button

Deletes the node selected in the EtherNet/IP tree area. You can also delete a node from the menu displayed when right-clicking the node.

(9) Import project button

Imports only the EtherNet/IP settings from a project file.

This section describes the operation of the context menu displayed by right-clicking each node in the EtherNet/IP tree area.

Node	Context menu	Operation
EtherNet/IP Settings	—	—
CIP connection point list	Add CIP Connection Point Setting	Add one CIP connection point.
	Delete All CIP Connection Point Setting	Delete all CIP connection points.
CIP connection point	Add CIP Connection Point Setting	Add one CIP connection point.
	Delete Selected CIP Connection Point Setting	Deletes the selected CIP connection point.
Scan list	Add Target	Add one target.
	Delete All Target	Delete all targets.
Target	Add CIP Connection	Add one CIP connection.
	Delete Selected Target	Deletes the selected target.
	Import the EDS file into 'EDS management'	Import the EDS file of the selected target to the EDS Management area. Once you import the EDS file, you can use it in other projects.
CIP connection	Add CIP Connection	Add one CIP connection.
	Delete Selected CIP Connection	Deletes the selected CIP connection.

Parameter Setting Area

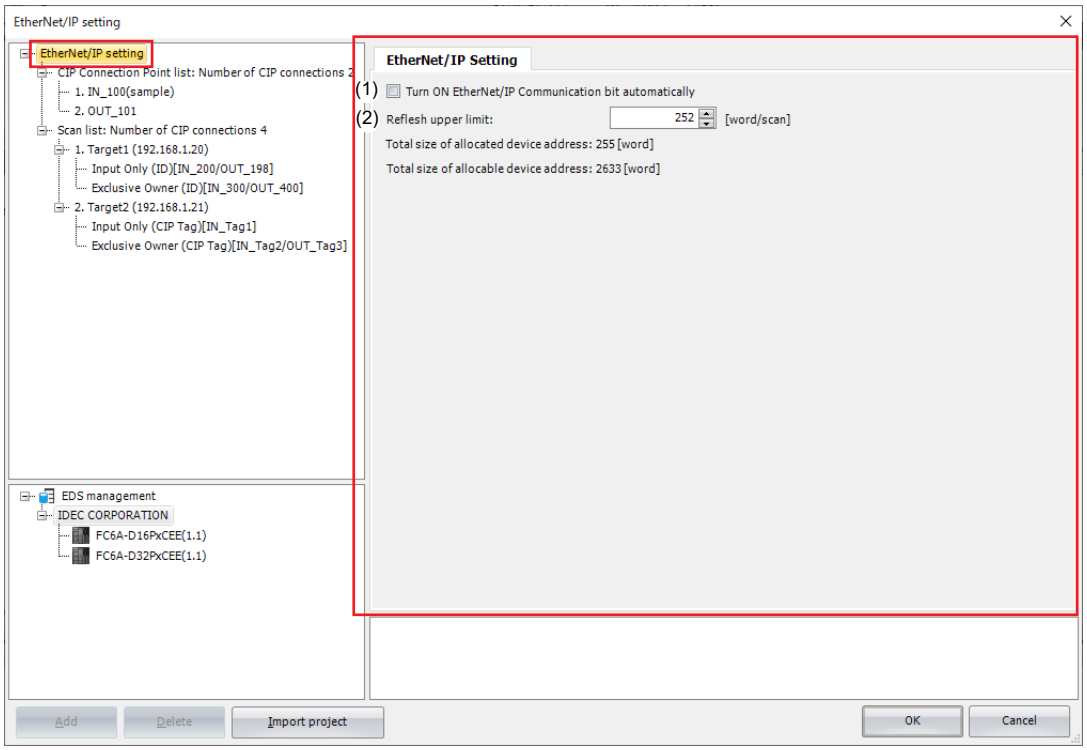
This area displays details on the node selected in the EtherNet/IP tree area.

When the EtherNet/IP Settings node is selected

Select the **EtherNet/IP Settings** node in the EtherNet/IP tree area. The **EtherNet/IP Setting** tab is displayed in the parameter settings area.

■EtherNet/IP Setting tab

Configure the basic settings for performing EtherNet/IP communication.



(1) Turn ON EtherNet/IP Communication bit automatically

Sets whether to automatically turn on the EtherNet/IP Communication Bit (M8460/LSM87). When the check box is checked, the EtherNet/IP Communication Bit (M8460/LSM87) turns on automatically.

Setting Method	Description
Selected	The EtherNet/IP communication Bit (M8460/LSM87) turns on automatically at the following times. FC6A: When the ladder program changes from stop to run. FT2J/1J, HG2J/1J: When the operating mode is changed.*1
Cleared	The EtherNet/IP Communication Bit (M8460/LSM87) does not turn on automatically.

*1 When switching between RUN mode and Monitor mode, the previous EtherNet/IP Communication Bit (LSM87) status is maintained.

(2) Refresh upper limit

FC6A Plus FC6A All-in-One FT2J/1J HG2J/1J HG5G/4G/3G/2G-V

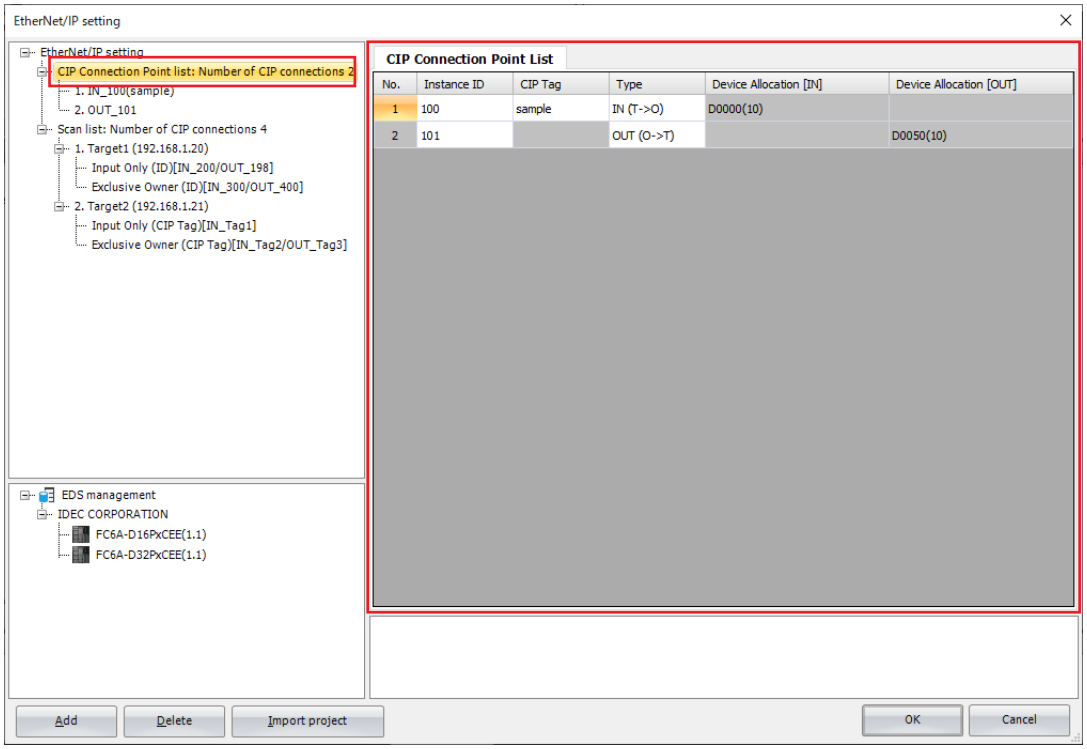
Sets the maximum size of data that can be reflected between the I/O data buffer and device addresses in END processing one time. The refresh upper limit is set between 1 and 512 words per scan. The default value is 252 words per scan. The maximum size of data to write from the I/O data buffer to the device addresses and the maximum size of data to write from the device addresses to the I/O data buffer in END processing one time are half of the refresh upper limit.

When the CIP Connection Point List node is selected

Select the **CIP Connection Point List** node in the EtherNet/IP tree area. The **CIP Connection Point List** tab is displayed in the parameter settings area.

■CIP Connection Point List tab

This tab displays the list of configured CIP connection points. The CIP tag, instance ID, and type can be changed. Double-click a device allocation cell. The **CIP Connection Point List** tab is displayed and you can move to the relevant cell for device allocation.

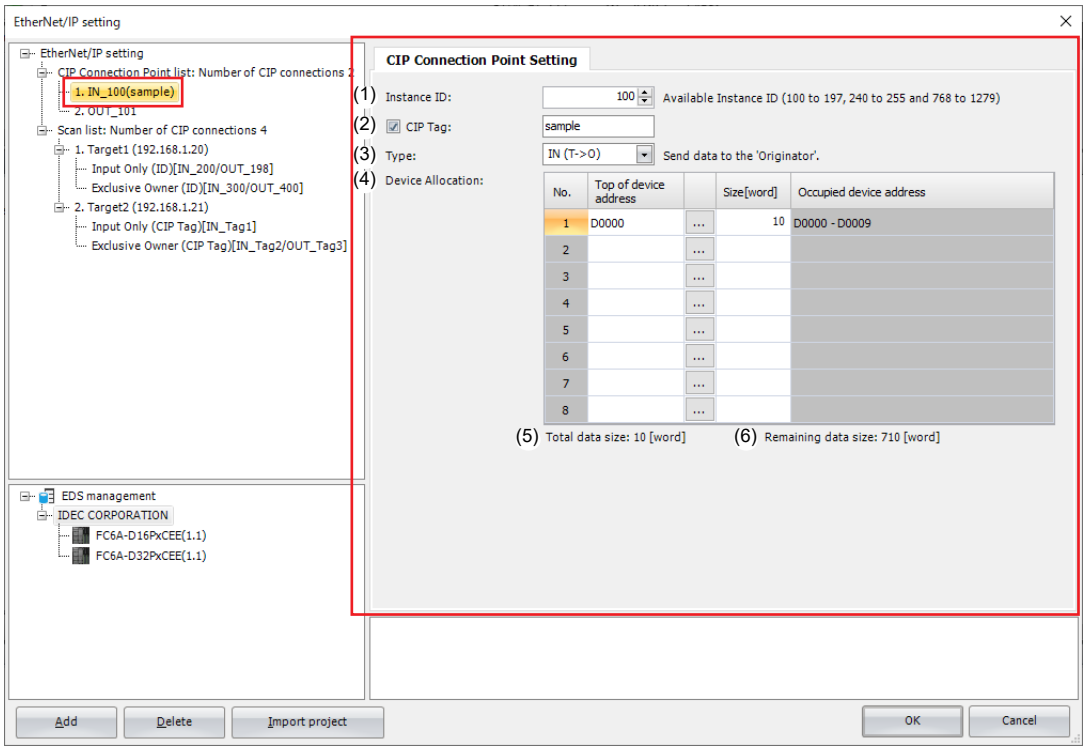


When the CIP Connection Point node is selected

Select a **CIP Connection Point** node in the EtherNet/IP tree area. The **CIP Connection Point Setting** tab is displayed in the parameter settings area.

■CIP Connection Point Setting tab

Configure the CIP connection point on this tab.



(1) Instance ID

Sets the instance ID. When the main unit (target) performs I/O message communication with another scanner device (originator) that cannot handle CIP tag names, the originator sets the CIP connection for this instance ID. Set the instance ID between 100 and 1,279. The default value is 100. Set this value so that it is not duplicated by other CIP connections.

Certain instance IDs cannot be used between 100 and 1,279. Set the value to one of the available values displayed in WindLDR or WindO/I-NV4.

(2) CIP Tag

Sets the CIP tag name. Up to 64 bytes can be set. The other scanner device (originator) sets the CIP connection for this tag name and performs I/O message communication with the main unit (target). Select this check box and a CIP tag can be set.

(3) Type

Sets the direction of the data to handle in I/O message communication. This item can be set from the following two types. The default value is "IN (T->O)".

Setting	Description
IN (T->O)	The target (main unit) transmits data to the originator.
OUT (O->T)	The target (main unit) receives data from the originator.

(4) Device Allocation

Sets the device addresses that store the data to receive or transmit in I/O message communication.
The word device that can be specified depends on the main unit type.

Internal Device Name		Symbol	FC6A	FT2J/1J	HG2J/1J
Data Register		D	Yes	No	No
HMI Devices	HMI Data Register	LDR	No	Yes	Yes
	HMI Keep Register	LKR	No	Yes	Yes
	HMI Temporary Register	LBR	No	Yes	Yes
Control Devices	Data Register	D	No	Yes	No

For details, see the following manuals.
FC6A: Chapter 6 "Devices" in the "FC6A Series MICROSmart User's Manual".
FT2J/1J, HG2J/1J: Chapter 35 "2 Word Devices" in the "WindO/I-NV4 User's Manual".

Starting from the set device address, device addresses are used in only the amount set by the data size. Set the data size between 1 and 720. The default value is 1. Set the first device address so that the device range is not exceeded.

(5) Total data size

Displays the total data size of device addresses used in I/O message communication.

(6) Remaining data size

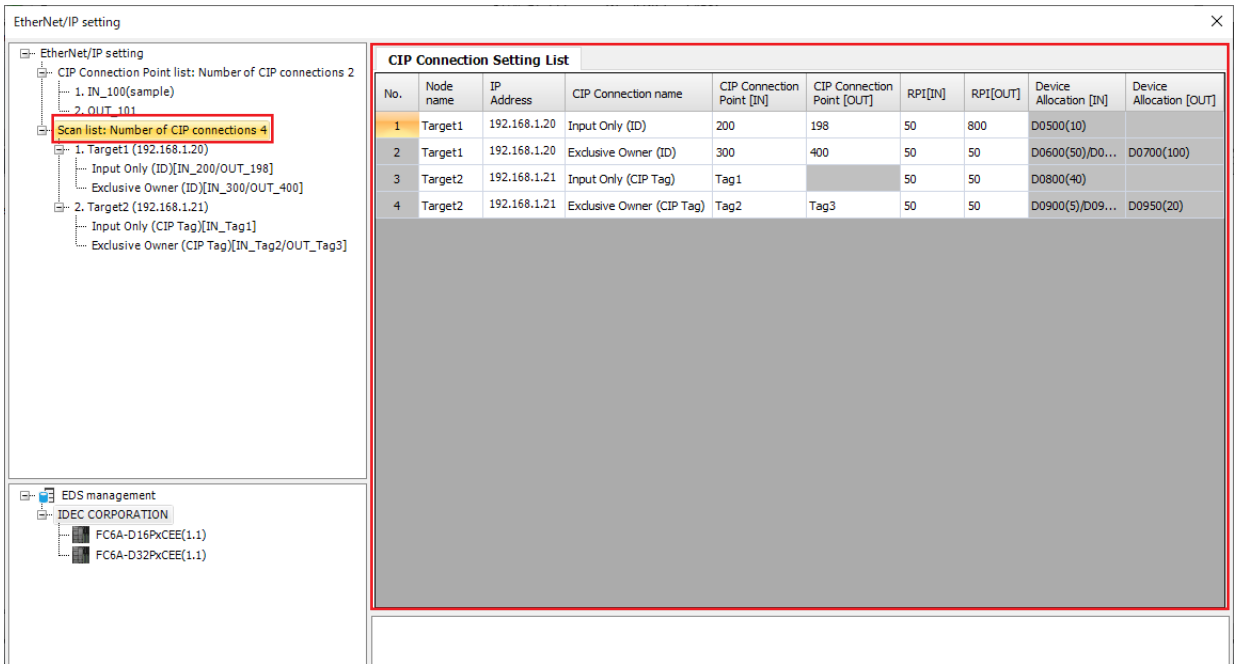
Displays the remaining data size of data that can be used (720 bytes minus the total value of device addresses that are used).

When the Scan List node is selected

Select the **Scan List** node in the EtherNet/IP tree area. The **CIP Connection Setting List** tab is displayed in the parameter settings area.

■CIP Connection Setting List tab

This tab displays the list of configured CIP connections. The node name, IP address, CIP connection name, CIP connection point, and RPI can be changed. Double-click a device allocation cell. The **CIP Connection Setting List** tab is displayed and you can move to the relevant cell for device allocation.

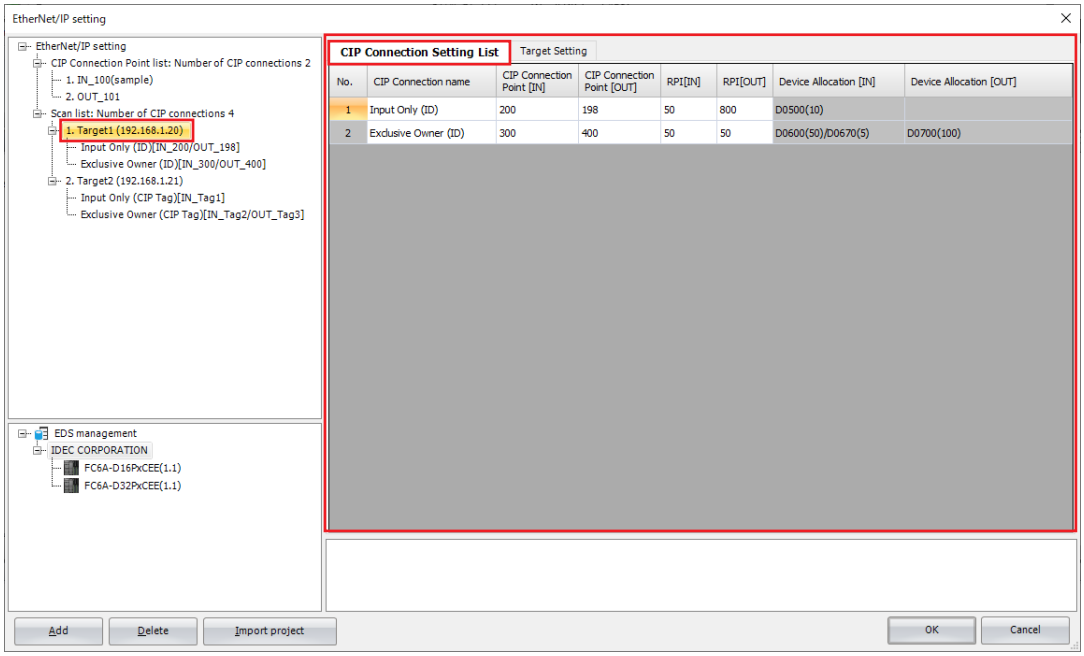


When the Target node is selected

Select a **Target** node in the EtherNet/IP tree area. The **CIP Connection Setting List** and **Target Setting** tabs are displayed in the parameter settings area.

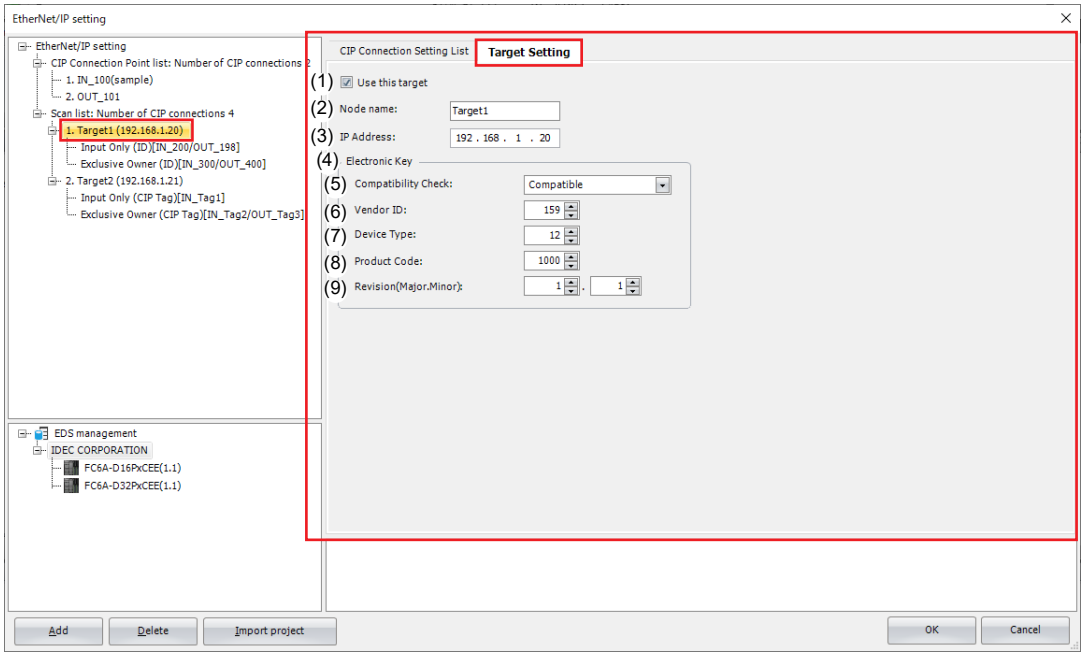
■CIP Connection Setting List tab

This tab displays the list of CIP connections of the configured target. The CIP connection name, CIP connection point, and RPI (communication cycle) can be changed on this tab.



■Target Setting tab

Configure the target device settings on this tab.



(1) Use this target

Enables or disables the CIP connections in the scan list by target. Check this box to enable the CIP connection. If the CIP connection is enabled, the main unit performs I/O message communication with the target when the EtherNet/IP Communication Bit (M8460/LSM87) is turned on. If the CIP connection is disabled, the main unit does not perform I/O message communication.

(2) Node name

Sets the node name of the target. The set node name is displayed on the scan list. Up to 30 bytes can be set.

(3) IP Address

Sets the IP address of the target.

(4) Electronic Key

Checks if the target registered to the scan list matches the EtherNet/IP device that is actually connected according to the compatibility check settings when the CIP connection is opened. I/O message communication is allowed with only a verified and matched EtherNet/IP device. If a device does not match, the main unit does not perform I/O message communication with the EtherNet/IP device.

(5) Compatibility Check

Sets the method for the compatibility check from the following three types.

Compatibility Check	Description
Disabled	Electronic key verification is not performed.
Compatibility	I/O message communication is allowed with only a target that satisfies all of the following conditions. • The settings and the actual vendor ID, device type, and product code match • The actual major revision and minor revision are larger than the settings
Complete Match	I/O message communication is performed with only a target that matches all of the set electronic key.

(6) Vendor ID

Sets the vendor ID of the EtherNet/IP device. Set this item between 0 and 65,535.

(7) Device Type

Sets the device type of the EtherNet/IP device. Set this item between 0 and 65,535.

(8) Product Code

Sets the product code of the EtherNet/IP device. Set this item between 0 and 65,535.

(9) Revision (Major, Minor)

Sets the major revision and minor revision of the EtherNet/IP device.

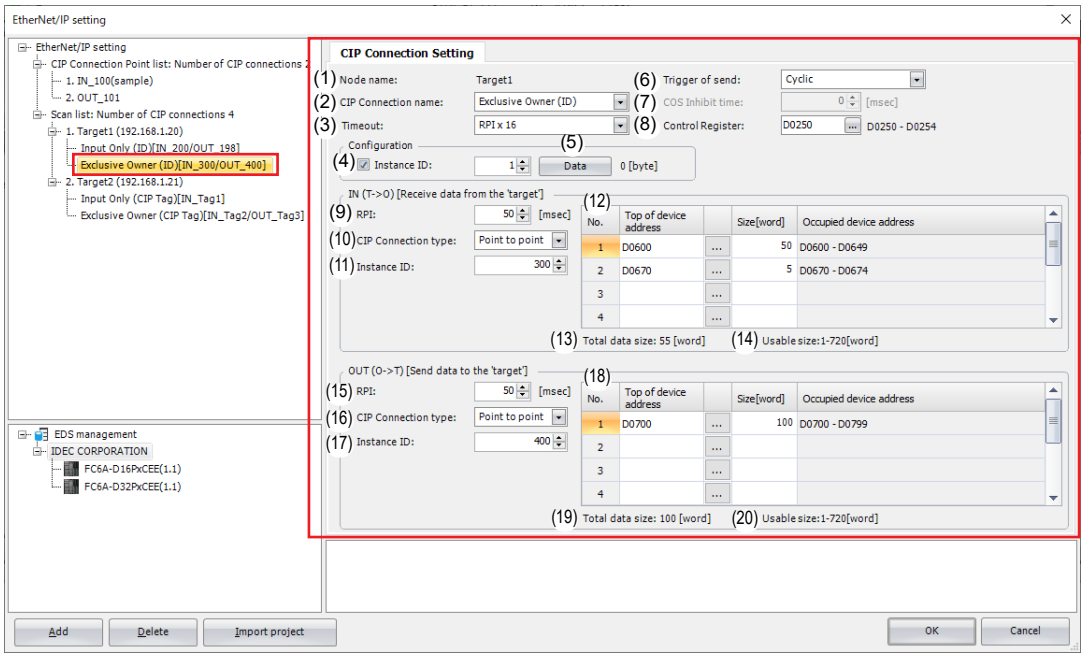
Revision	Range
Major revision	0 to 127
Minor revision	0 to 255

When the CIP Connection node is selected

Select a **CIP Connection** node in the EtherNet/IP tree area. The **CIP Connection Setting** tab is displayed in the parameter settings area.

■CIP Connection Setting tab

Configure the CIP connection of the target on this tab.



(1) Node name

Displays the node name that was set on the **Target Setting** tab. Configure the CIP connection of the target with this node name.

(2) CIP Connection name

Sets the CIP connection name of the target with the node name (1). This item can be set from the following five types.

Setting	Description
Exclusive Owner (ID)	Set this name when the main unit transmits and receives data with the target. For Exclusive Owner (ID), set the CIP connection point of the target as an instance ID. For Exclusive Owner (CIP Tag), set the CIP connection point of the target as a CIP tag name. The main unit specifies the CIP connection point of the target as a CIP tag name or an instance ID and performs I/O message communication with the target.
Exclusive Owner (CIP Tag)	
Input Only (ID)	Set this name when the main unit receives data from the target only. For Input Only (ID), set the CIP connection point of the target as an instance ID. For Input Only (CIP Tag), set the CIP connection point of the target as a CIP tag name. The main unit specifies the CIP connection point of the target as a CIP tag name or an instance ID and performs I/O message communication with the target.
Input Only (CIP Tag)	
Listen Only (ID)	When other originators and a target have opened a CIP connection as Exclusive Owner or Input Only and that target is transmitting data by multicast, set this name to have the main unit receive that data. For Listen Only (ID), the CIP connection point of the target is specified as an instance ID.

The main unit cannot specify a Listen Only CIP connection as a CIP tag name.

(3) Timeout

Sets the timeout for I/O message communication. The main unit and target both monitor the time interval to receive data. The setting range is an integer multiple (x4, x8, x16, x32, x64, x128, x256, or x512) of the RPI ((9), (15)). The default value is RPI x 16.

(4) Instance ID

Sets the instance ID of the target Configuration. Select this check box and the instance ID can be set. Set the instance ID between 1 and 65,535. The default value is 1.

(5) Data

Sets the data for the target Configuration. When the main unit opens a CIP connection, it also transmits the Configuration instance ID to the target. A 0 to 400 byte parameter can be set in hexadecimal depending on the specification of the target.

If the EDS file of the EtherNet/IP device added to the scan list defines the parameter information related to opening a CIP connection, click the **Data** button to display the list of the parameters. This parameter can be changed.

(6) Trigger of send

Sets the method for the main unit to transmit the data to the target. This item can be set from the following two types. The default value is "Cyclic".

Setting	Description
Cyclic	The main unit and target transmit data at the set RPI (communication cycle) (9).
Change Of State (COS)	The main unit and target transmit data at the set RPI (communication cycle) (9) or when a value changes.

(7) COS Inhibit time

Sets the time from when a value changes to when the transmission occurs when Change Of State (COS) is selected for the trigger of send (6). The main unit does not transmit the data until the set time has elapsed from when the change in value was detected.

A change in a value means a change in the data in the I/O data buffer held in the main unit. Data is reflected from the device addresses to the I/O data buffer in the END processing.

(8) Control Register

Sets the data registers that will be used by the CIP connection.
The word device that can be specified depends on the main unit type.

Internal Device Name		Symbol	FC6A	FT2J/1J	HG2J/1J
Data Register		D	Yes	No	No
HMI Devices	HMI Data Register	LDR	No	Yes	Yes
	HMI Keep Register	LKR	No	Yes	Yes
	HMI Temporary Register	LBR	No	Yes	Yes
Control Devices	Data Register	D	No	Yes	No

For details, see the following manuals.
FC6A: Chapter 6 "Devices" in the "FC6A Series MICROSmart User's Manual".
FT2J/1J, HG2J/1J: Chapter 35 "2 Word Devices" in the "WindO/I-NV4 User's Manual".

This option uses 5 words of address numbers starting from the specified device address.

Device Address	Item		Description
Starting number+0	Bit 0	Connection Status	ON: Connected and communicating status OFF: Not connected status (during retries, etc.)
	Bits 1 to 15	Reserved	
Starting number+1	Reserveds		
Starting number+2			
Starting number+3			
Starting number+4			

Configure the parameters related to the input direction (target -> main unit (originator)) of I/O message communication in (9) to (14).

(9) RPI

Sets the communication cycle of the input direction (target -> originator) when the main unit (originator) is performing I/O message communication with the target. The range that can be set is 10 to 10,000 ms. The default value is 50 ms.

(10) CIP Connection type

Sets the method for the main unit (originator) to receive data transmitted from the target. This item can be set from the following two types. The default value is "Point to point".

Setting	Description
Point to point	Set this type when the main unit receives data from the target on a one-to-one basis.
Multicast	Set this type when multiple originators including the main unit receive data from one target.In this case, please set the timeout (3) and RPI (9) so that the IN (T->O) timeout period exceeds 250 ms.

(11) Instance ID/CIP Tag

Sets the CIP connection point of the target. Set the instance ID or CIP tag name of the target according to the CIP connection name. This item can be set from the following five types.

CIP Connection Name	CIP Connection Point	Range
Exclusive Owner (ID)	Instance ID	1 to 65,535
Exclusive Owner (CIP Tag)	CIP tag name	UTF-8, 64 bytes maximum (including string terminator)
Input Only (ID)	Instance ID	1 to 65,535
Input Only (CIP Tag)	CIP tag name	UTF-8, 64 bytes maximum (including string terminator)
Listen Only (ID)	Instance ID	1 to 65,535

(12) Device Allocation (IN (T->O))

Sets the device addresses to which the main unit (originator) writes data that is received from the target.
The word device that can be specified depends on the main unit type.

Internal Device Name		Symbol	FC6A	FT2J/1J	HG2J/1J
Data Register		D	Yes	No	No
HMI Devices	HMI Data Register	LDR	No	Yes	Yes
	HMI Keep Register	LKR	No	Yes	Yes
	HMI Temporary Register	LBR	No	Yes	Yes
Control Devices	Data Register	D	No	Yes	No

For details, see the following manuals.
FC6A: Chapter 6 "Devices" in the "FC6A Series MICROSmart User's Manual".
FT2J/1J, HG2J/1J: Chapter 35 "2 Word Devices" in the "WindO/I-NV4 User's Manual".

The data can be allocated to multiple device addresses in word units. A maximum of 8 areas can be set.
The total size of the areas to allocate is 720 words maximum. Starting from the set device address, device addresses are used in only the amount set by the data size. Set the data size between 1 and 720. The default value is 1. Set the first device address so that the device range is not exceeded.

(13) Total data size (IN (T->O))

Displays the total data size of device addresses used in the device allocations (IN (T->O)) (12).

(14) Remaining data size (IN (T->O))

Displays the remaining data size of data that can be used (720 bytes minus the total value of device addresses that are used).

Configure the parameters related to the output direction (main unit (originator) -> target) of I/O message communication in (15) to (20).

(15) RPI

Sets the communication cycle of the output direction (originator -> target) when the main unit (originator) is performing I/O message communication with the target. The range that can be set is 10 to 10,000 ms. The default value is 50 ms.
This RPI is the heartbeat cycle when Input Only or Listen Only is set for the CIP connection name. The value of the RPI for IN (T->O) data multiplied by 16 is automatically set, but it can be changed.

(16) CIP Connection type

Sets the method for the main unit (originator) to transmit data to the target. Only "Point to point" can be set.

Setting	Description
Point to point	Set this type when the main unit transmits data from the target on a one-to-one basis.

(17) Instance ID/CIP Tag

Sets the CIP connection point of the target. Set the instance ID or CIP tag name of the target according to the CIP connection name. This item can be set from the following five types.

CIP Connection Name	CIP Connection Point	Range
Exclusive Owner (ID)	Instance ID	1 to 65,535
Exclusive Owner (CIP Tag)	CIP tag name	UTF-8, 64 bytes maximum (including string terminator)
Input Only (ID)	Instance ID	1 to 65,535 *1
Input Only (CIP Tag)	CIP tag name	Setting not required
Listen Only (ID)	Instance ID	1 to 65,535 *2

*1 Set the instance ID for an Input Only target.

*2 Set the instance ID for a Listen Only target.

(18) Device Allocation (OUT (O->T))

Sets the device addresses in which the main unit (originator) stores data to transmit to the target.
The word device that can be specified depends on the main unit type.

Internal Device Name		Symbol	FC6A	FT2J/1J	HG2J/1J
Data Register		D	Yes	No	No
HMI Devices	HMI Data Register	LDR	No	Yes	Yes
	HMI Keep Register	LKR	No	Yes	Yes
	HMI Temporary Register	LBR	No	Yes	Yes
Control Devices	Data Register	D	No	Yes	No

For details, see the following manuals.
FC6A: Chapter 6 "Devices" in the "FC6A Series MICROSmart User's Manual".
FT2J/1J, HG2J/1J: Chapter 35 "2 Word Devices" in the "WindO/I-NV4 User's Manual".

The data can be allocated to multiple device addresses in word units. A maximum of 8 areas can be set.
The total size of the areas to allocate is 720 words maximum. Starting from the set device address, device addresses are used in only the amount set by the data size. Set the data size between 1 and 720. The default value is 1. Set the first device address so that the device range is not exceeded.

(19) Total data size (OUT (O->T))

Displays the total data size of device addresses used in the device allocations (OUT (O->T)) (18).

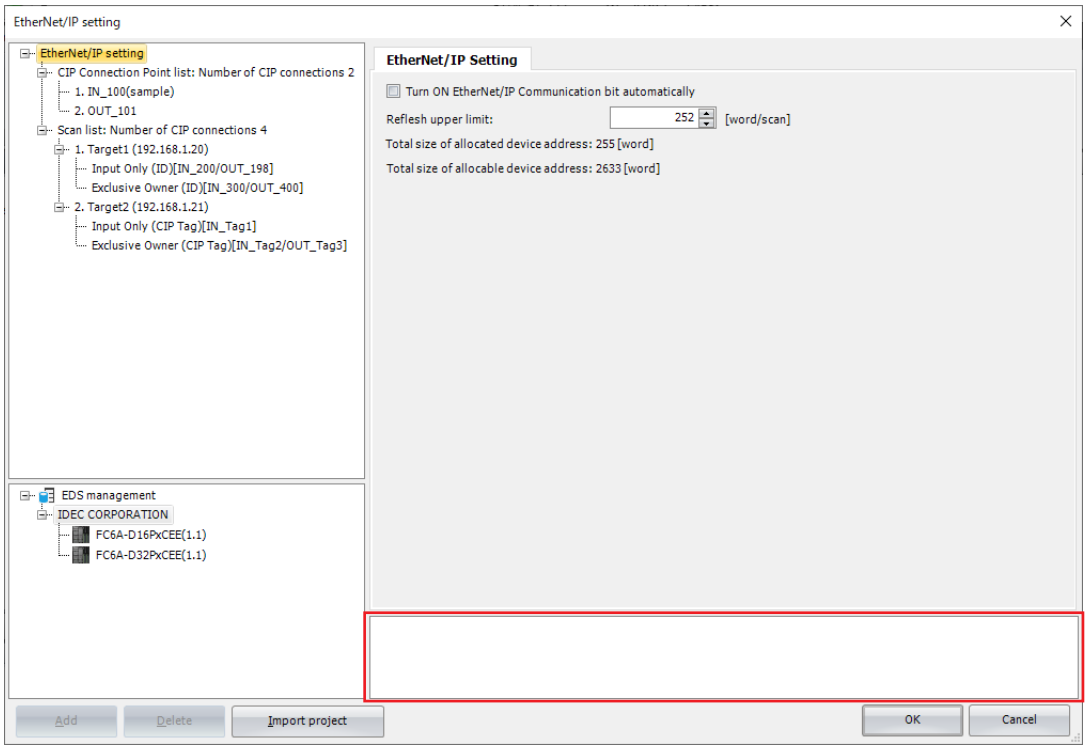
(20) Remaining data size (OUT (O->T))

Displays the remaining data size of data that can be used (720 bytes minus the total value of device addresses that are used).

Information View Area

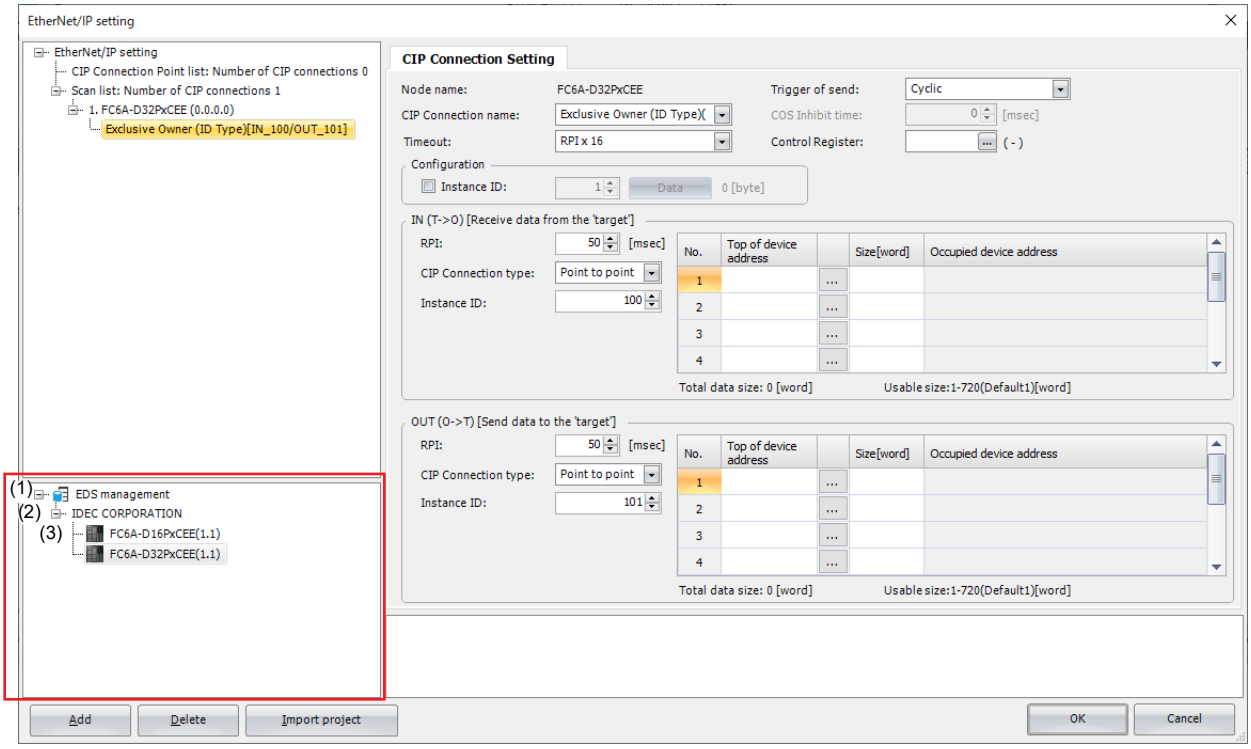
This area displays the details of errors if there are errors in content set on the **EtherNet/IP Settings** dialog box.

Click on the details of an error to display the location with the error.



EDS Management Area

Display a list of EtherNet/IP devices set in the imported EDS files.



(1) EDS Management node

Select the **EDS Management** node and expand it to display a list of EtherNet/IP device information set in the imported EDS file.

(2) Vendor node

Display the vendor name of the EtherNet/IP device set in the imported EDS file.

(3) Device node

Display the name of the EtherNet/IP device set in the imported EDS file and its revision.

This table describes the operation of the context menu displayed by right-clicking each node in the **EDS Management** area.

Node	Context menu	Operation
EDS Management	Import	Import the EDS files.
Vendor	Delete the EDS files of selected vendor	Delete the EDS file of selected vendor.
Device	Export	Outputs the EDS file of the selected EtherNet/IP device with text format.
	Open	Open the EDS file of the selected EtherNet/IP device.
	Delete	Deletes the EDS file of the selected EtherNet/IP device.
	Add to scan list	Adds the selected EtherNet/IP device to the scan list.

- You can add the selected EtherNet/IP device to the scan list by dragging and dropping it onto the scan list in the EtherNet/IP tree area with the **Device** node selected.
- The imported EDS file is saved in the PC, so the next time WindLDR is started, the information of EtherNet/IP devices configured in the EDS file will be listed in the **EDS Management** area.
- EDS files for EtherNet/IP devices of our products are imported from the beginning in WindLDR (version 8.18.0 or later) and WindO/I-NV4.
- If you use the device with an imported EDS file, the CIP connection name will be displayed in the EDS file.

Flow of EtherNet/IP Communication Settings

Basic Settings for EtherNet/IP Communication

Select the **EtherNet/IP Setting** node[*1] in the EtherNet/IP tree area. The **EtherNet/IP Setting** tab is displayed in the parameter settings area. Configure the basic settings for performing EtherNet/IP communication on this tab. For details, see "When the EtherNet/IP Settings node is selected" on page 2-24.

*1 For details on the **EtherNet/IP Setting** node, see "EtherNet/IP Tree Area" on page 2-22.

Using the Main Unit as a Target

This section describes the settings required to use the main unit as a target.

A CIP connection point must be configured when the main unit (target) transmits and receives data with another scanner device (originator).

■CIP Connection Point Settings

In the settings for the main unit to transmit and receive data with the originator, configure the device addresses for storing communication data, data size, input and output direction of data, and other settings.

Select a **CIP Connection Point** node[*1] in the EtherNet/IP tree area. The **CIP Connection Point Setting** tab is displayed in the parameter settings area. Configure the CIP connection point on this tab. For details, see "When the CIP Connection Point node is selected" on page 2-26.

*1 For details on the **CIP Connection Point** node, see "EtherNet/IP Tree Area" on page 2-22.

Using the Main Unit as an Originator

This section describes the settings required to use the main unit as an originator.

You must configure the settings for the target device that performs I/O message communication with the main unit and the CIP connection settings.These settings can be easily set using the EDS file generally provided by each maker. See "Creating Scanlists Using EDS Files" on page 2-36.

■Target Settings

Configure the target device settings, such as the IP address of the target that performs I/O message communication with the main unit.

Select a **Target** node[*1] in the EtherNet/IP tree area. The **Target Setting** tab is displayed in the parameter settings area. Configure the target device settings on this tab. For details, see "When the Target node is selected" on page 2-28.

*1 For details on the **Target** node, see "EtherNet/IP Tree Area" on page 2-22.

■CIP Connection Settings

In the settings for the main unit to transmit and receive data with the target, configure the CIP connection name, CIP connection type, device addresses for storing communication data, data size, RPI (communication cycle) for I/O message communication, and the other settings for each target.

Select a **CIP Connection** node[*1] in the EtherNet/IP tree area. The **CIP Connection Setting** tab is displayed in the parameter settings area. Configure the CIP connection settings on this tab. For details, see "When the CIP Connection node is selected" on page 2-30.

*1 For details on the **CIP Connection** node, see "EtherNet/IP Tree Area" on page 2-22.

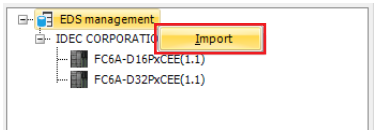
■Creating Scanlists Using EDS Files

The main unit can easily set the target device settings and CIP connection settings for sending and receiving targets and data using the EDS (Electric Data Sheets) file. The EDS files define information specific to EtherNet/IP devices such as the vendor name, data sending and receiving settings, and parameter specifications.The EDS files are created and provided by the vendor of each EtherNet/IP device.You can obtain the EDS file from the vendor's website etc.

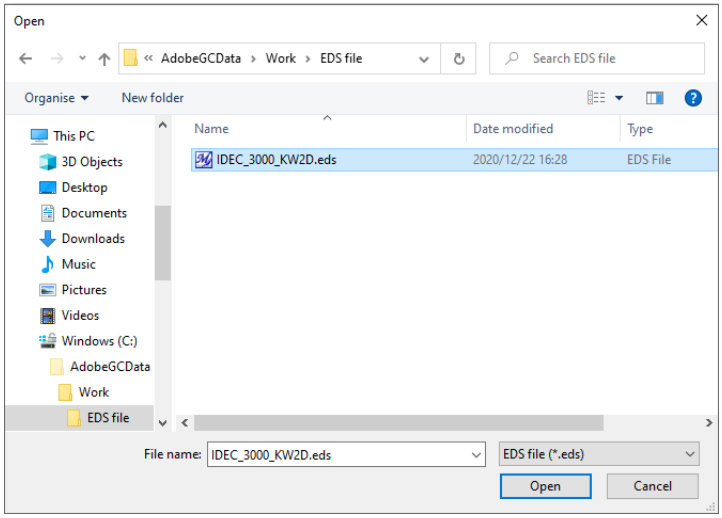
This section describes how to import an EDS file and create a scan list.

- **Operation procedure**

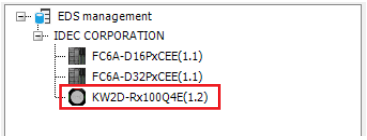
1. In the **EDS Management** area, right-click the **EDS Management** node and click **Import**.



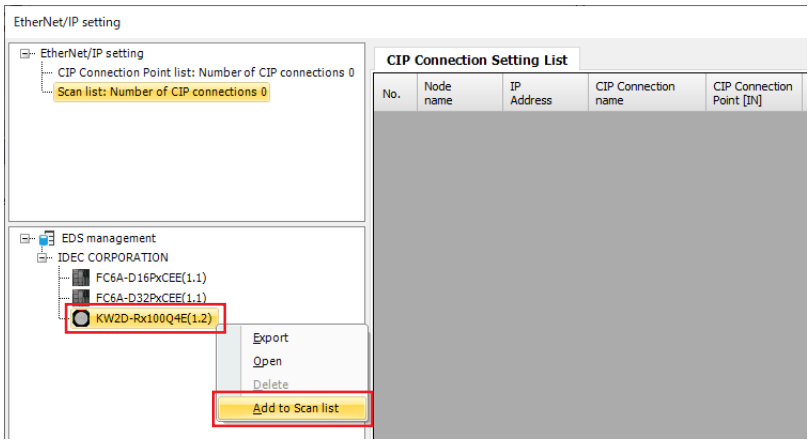
The **Open** dialog box is displayed.



2. Select the EDS file and click the **Open** button.
The selected EDS file is imported into the **EDS Management** area.

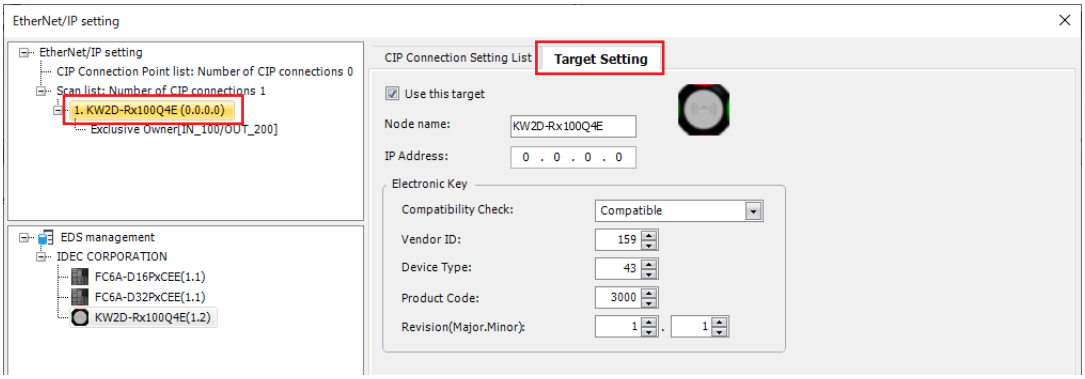


3. In the **EDS Management** area, right-click the **EDS Management** node and click **Add to Scan List**.

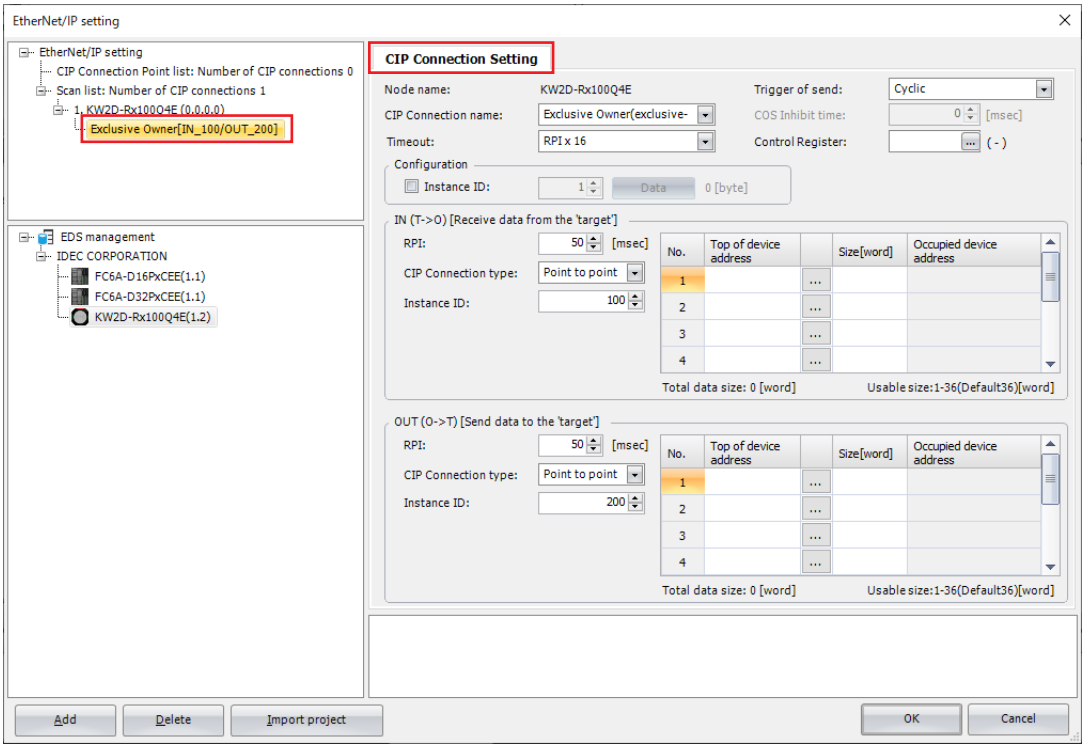


The settings of the selected EDS file are added to the scan list.

4. Select the **Target** node in the EtherNet/IP tree area and set the IP address on the **Target Settings** tab. Also, check the settings on the **Target Settings** tab and change the settings if necessary.



5. Select the **CIP Connections** node in the EtherNet/IP tree area and set the timeouts, control registers and device assignments on the **CIP Connection Settings** tab. Also, check the settings on the **CIP Connection Settings** tab and change the settings if necessary.



6. Click the **OK** button.

This concludes configuring the scan list creation.

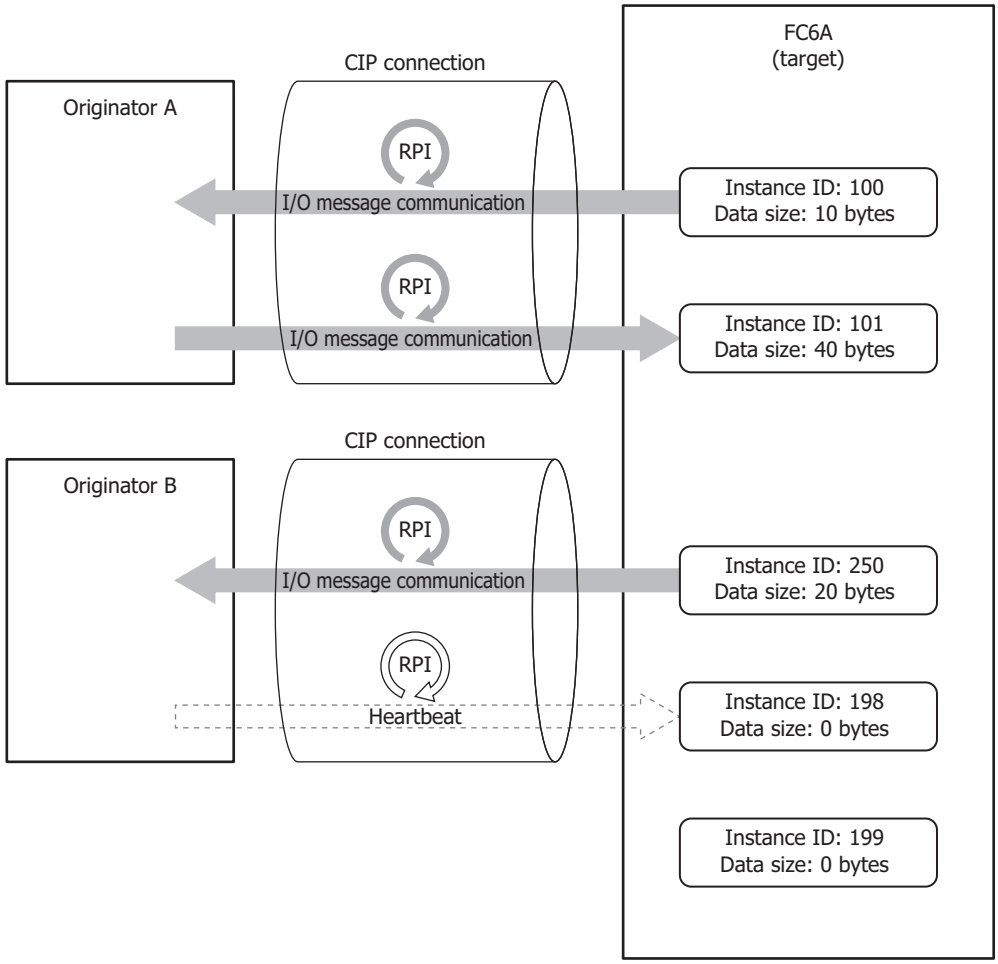
EDS files are prepared by each EtherNet/IP device manufacturer. Please contact each EtherNet/IP device manufacturer for how to obtain the EDS file.

Examples of EtherNet/IP Settings

Using the Main Unit as a Target

This section describes an example of the settings for connecting to an originator with the FC6A as the target.

■System configuration diagram



■Specifications to set for the main unit

Configure a CIP connection point to read and write from Originator A and B using the main unit as a target.

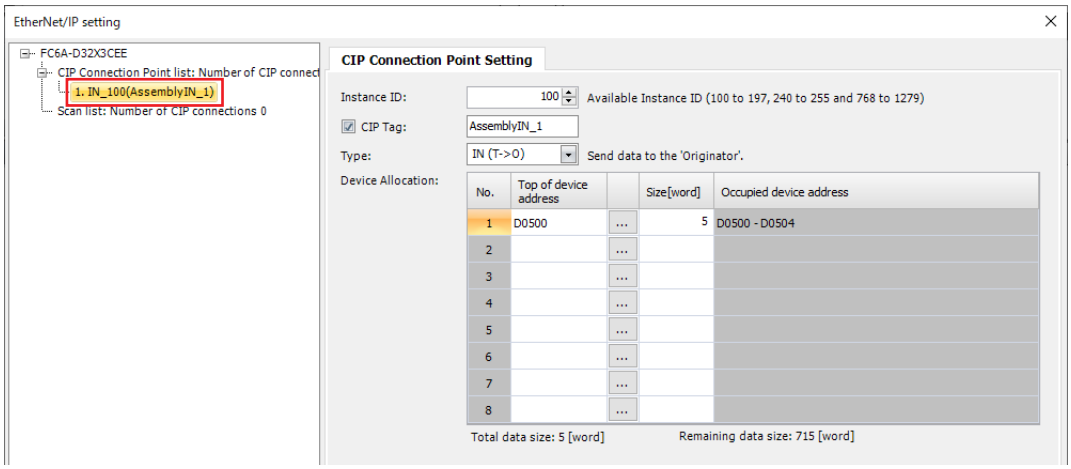
CIP Connection Point			Data Register for Storing Data
Input & Output Type	**Instance ID**	**Data Size**	
Input	100	10 bytes (5 words)	D0500
Output	101	40 bytes (20 words)	D0600
Input	250	20 bytes (10 words)	D0700
Output (for Input Only)	198 (fixed by system)	0 bytes	—
Output (for Listen Only)	199 (fixed by system)	0 bytes	—

■**Settings**

Configure the CIP connection point settings for the main unit.

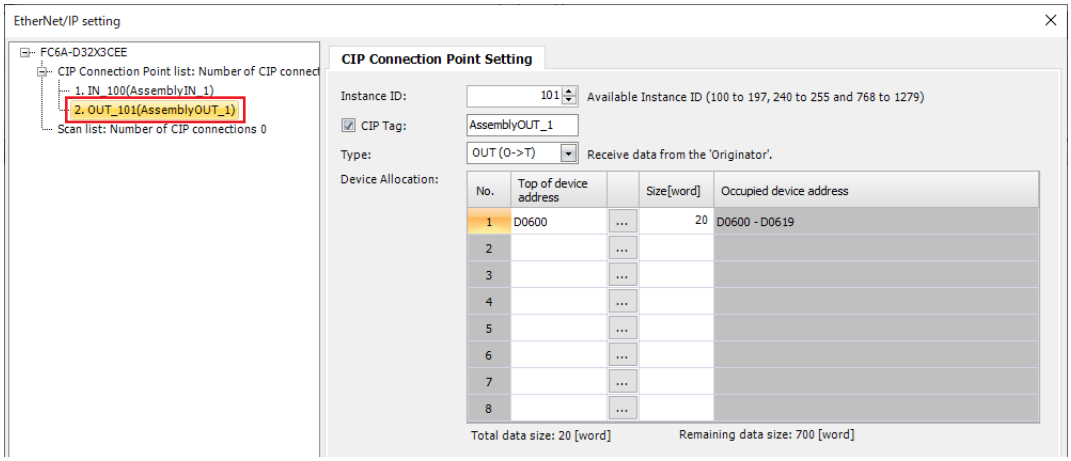
Main Unit CIP Connection Point Settings

The main unit (target) transmits the CIP connection point (instance ID: 100) data stored in device addresses to the originator.



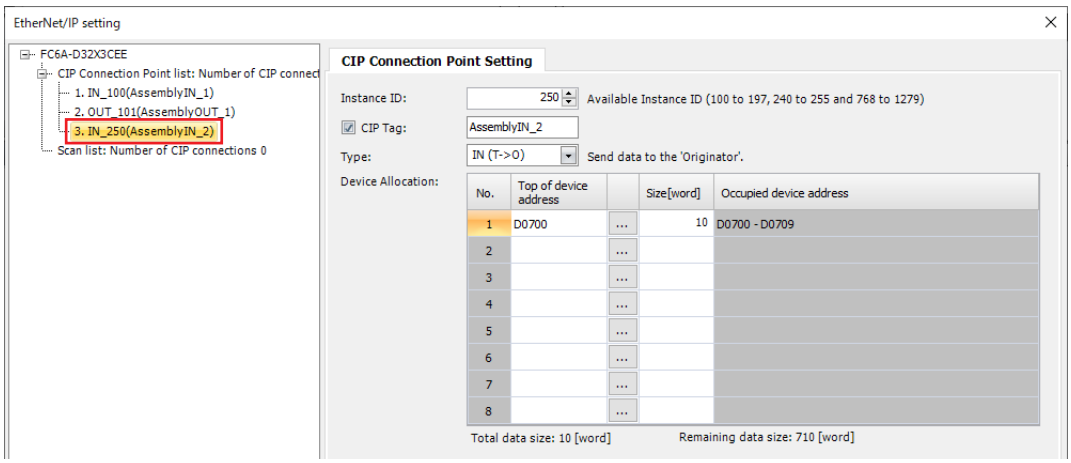
Settings		Description
1.	CIP tag	"AssemblyIN_1"
	Instance ID	100
	Type	IN (T->O)
	Device allocations	D0500 to D0504 (5 words)

The main unit (target) receives the CIP connection point (instance ID: 101) data from the originator and stores it in device addresses.



Settings		Description
2.	CIP tag	"AssemblyOUT_1"
	Instance ID	101
	Type	OUT (O->T)
	Device allocations	D0600 to D0619 (20 words)

The main unit (target) transmits the CIP connection point (instance ID: 200) data stored in device addresses to the originator.



Settings		Description
3.	CIP tag	"AssemblyIN_2"
	Instance ID	250
	Type	IN (T->O)
	Device allocations	D0700 to D0709 (10 words)

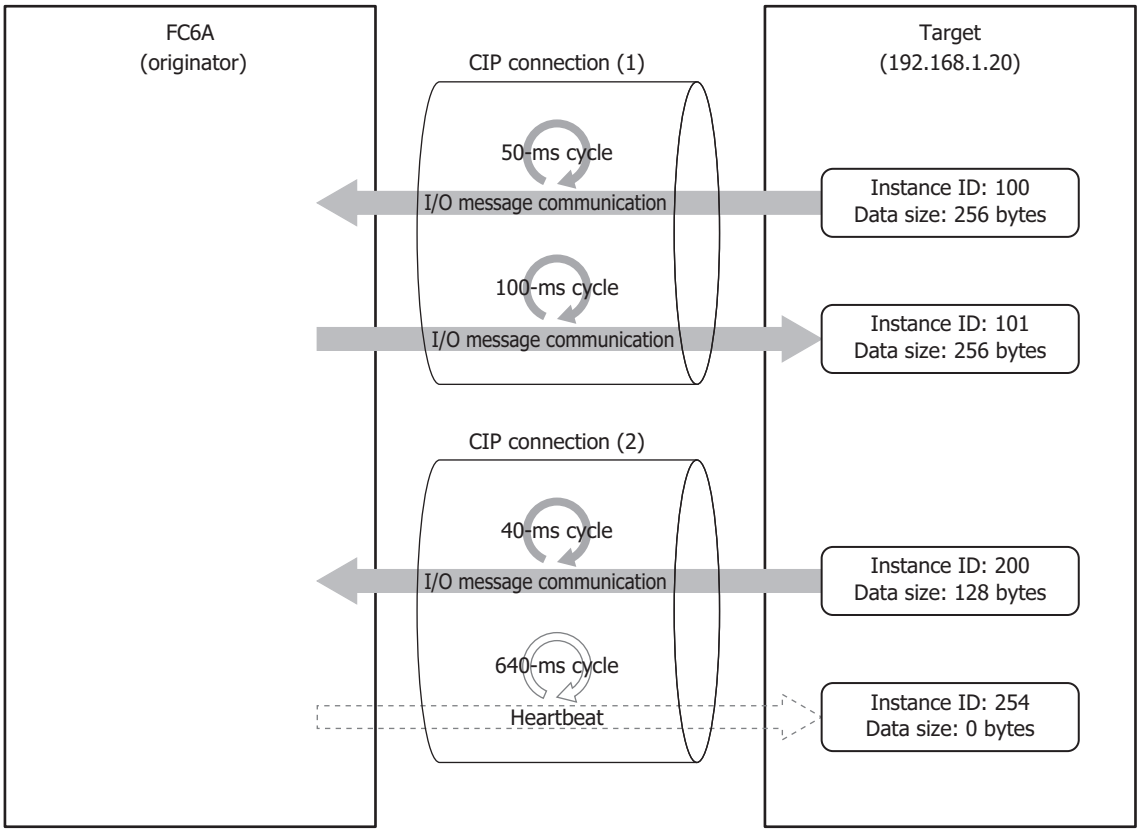
The RPI and CIP connection type, such as Exclusive Owner, are specified by the originator when it opens the CIP connection.

This concludes configuring the settings.

Using the Main Unit as an Originator

This section describes an example of the settings for connecting to a target with the FC6A as the originator.

■System configuration diagram



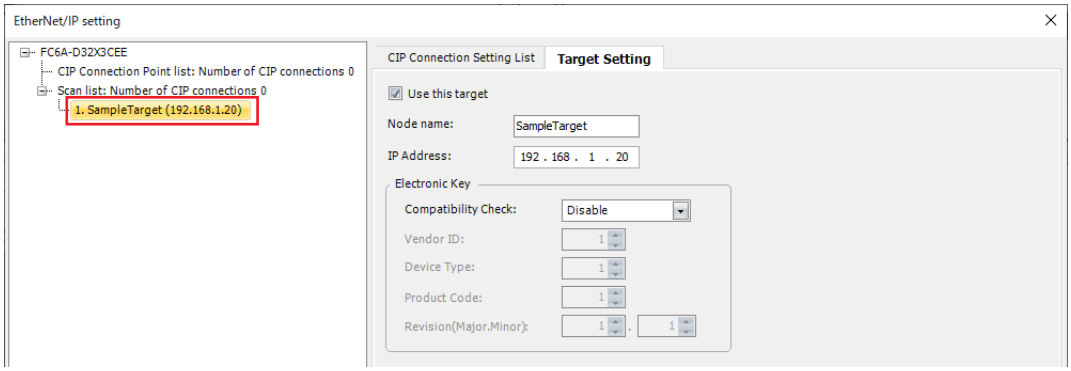
■Specifications of destination target

IP Address	192.168.1.20			
CIP Connection Point	The following table shows the CIP connection point of the target that main unit connects to as the originator.			
	CIP connection	**Type**	**Instance ID**	**Data Size**
	CIP connection (1)	Configuration	1	0 bytes
		Input	100	256 bytes (128 words)
		Output	101	256 bytes (128 words)
	CIP connection (2)	Configuration	1	0 bytes
		Input	200	128 bytes (64 words)
		Output (Heartbeat)	254	0 bytes

■**Settings**

Configure the target device and CIP connection settings.

Main unit Scan List Settings (Target Device Settings)

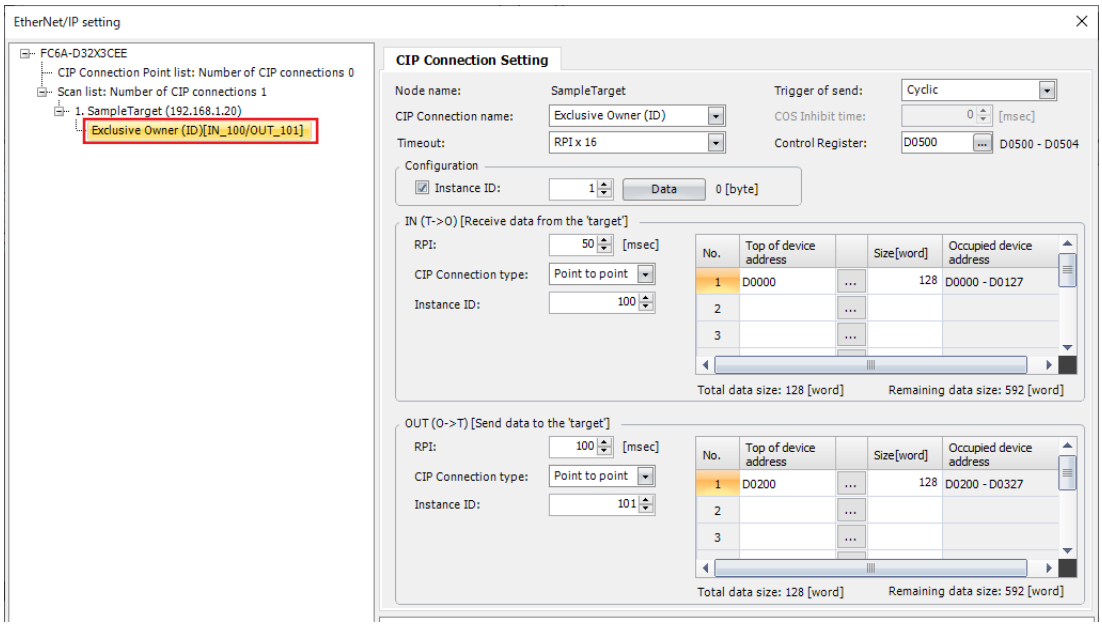


Settings		Description
Use this target		Enable
Node name		SampleTarget
IP Address		192.168.1.20
Electronic Key	Compatibility Check	Disable

Main unit Scan List Settings (CIP Connection Settings)

CIP connection (1)

The main unit (originator) sends a request to open the CIP connection to the CIP connection point (instance ID: 100 and 101), and when successful, receives data from the target every 50 ms and transmits data to the target every 100 ms.

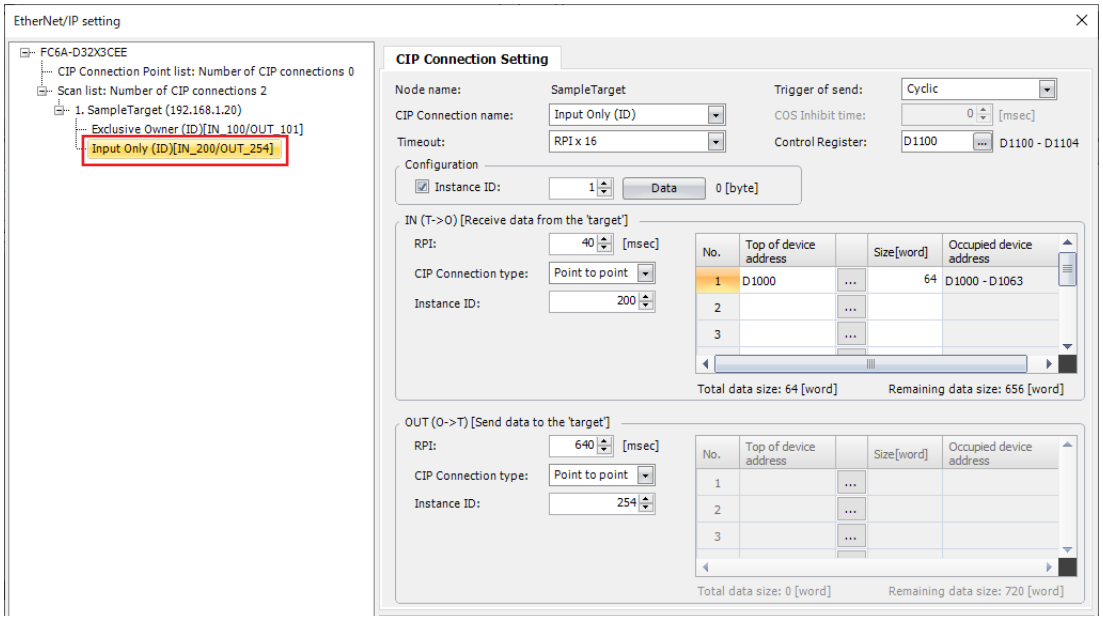


Settings		Description
CIP Connection Name		Exclusive Owner (ID)
Trigger of send		Cyclic
Timeout		RPI x 16
Control registers		D0500
Configuration	Instance ID	1
	Data	0 bytes

Settings		Description
IN (T->O)	RPI	50 ms
	CIP connection type	Point to point
	Instance ID	100
	Device allocations	D0000 to D0127 (128 words)
OUT (O->T)	RPI	100 ms
	CIP connection type	Point to point
	Instance ID	101
	Device allocations	D0200 to D0327 (128 words)

CIP connection (2)

Set Input Only for the CIP connection name when data is not transmitted to the target. The main unit (originator) sends a request to open the CIP connection to the CIP connection point (instance ID: 200 and 254), and when successful, receives data from the target every 40 ms. The heartbeat, not data, is sent to the target every 640 ms.



Settings		Description
CIP Connection Name		Input Only (ID)
Trigger of send		Cyclic
Timeout		RPI x 16
Control registers		D1100
Configuration	Instance ID	1
	Data	0 bytes
IN (T->O)	RPI	40 ms
	CIP connection type	Point to point
	Instance ID	200
	Device allocations	D1000 to D1063 (64 words)
OUT (O->T)	RPI	640msec
	CIP connection type	Point to point
	Instance ID	254

This concludes configuring the settings.

Objects

Identity Object (ClassID: 01H)

An object that provides device identification, general information and reset service, etc. The reset service performs by soft reset.

Class

■Class Services

ID	Services
0EH	Get_Attribute_Single
01H	Get_Attributes_All

■Class Attributes (Instance ID: 0)

ID	Attributes	Name	Data Type	Description	Value
1	R	Revision	UNIT	Object revisions	1
2	R	Max Instance	UNIT	Maximum instance number	1
3	R	Number of Instances	UNIT	Number of generated object instances	1
6	R	Maximum ID Number Class Attributes	UNIT	Maximum attribute ID number for class attributes	7
7	R	Maximum ID Number Instance Attributes	UNIT	Maximum attribute ID number of the instance attribute	7

Instance

■Instance Services

ID	Services
0EH	Get_Attribute_Single
01H	Get_Attributes_All
05H	Reset

■Instance Attributes (Instance ID: 1)

ID	Attributes	Name	Data Type	Description	Value
1	R	Vendor ID	UNIT	Vendor Identification Number	159
2	R	Device Type	UNIT	Common device types	*1
3	R	Product Code	UNIT	Product Identification Code	*2
4	R	Revision	STRUCT of:	Identity object revision	—
		Major Revision	UNIT	Major revision	—
		Minor Revision	UNIT	Minor revision	—
5	R	Status	UNIT	Current status of the device bit4 to 7 0010: Connection Timeout 0011: Connection disconnection 0110: One or more connections are in the RUN status 0111: All IDLE status during connection	Current Status
6	R	Serial Number	UNIT	Serial Number	—
7	R	Product Name	SHORT-STRING	Product Name	*2

*1 FC6A: 14 (Programmable Logic Controller)
FT2J/1J, HG2J/1J: 24 (Human-Machine Interface (HMI))

*2 FC6A:

Name	FC6A-D16R*CEE	FC6A-D16P*CEE	FC6A-D16K*CEE	FC6A-D32P*CEE	FC6A-D32K*CEE
Product Code	2000	2010	2020	2030	2040
Product Name	"FC6A-D16RxCEE"	"FC6A-D16PxCEE"	"FC6A-D16KxCEE"	"FC6A-D32PxCEE"	"FC6A-D32KxCEE"

FT2J/1J:

Name	FT1J-4F12RAG-*	FT1J-4F14SAG-*	FT1J-4F14KAG-*	FT2J-7U22RAF-*	FT2J-7U22SAF-*	FT2J-7U22KAF-*
Product Code	4110	4112	4111	4120	4122	4121
Product Name	"FT1J-4F12RAG-*"	"FT1J-4F14SAG-*"	"FT1J-4F14KAG-*"	"FT2J-7U22RAF-*"	"FT2J-7U22SAF-*"	"FT2J-7U22KAF-*"

HG2J/1J:

Name	HG1J-4FT22TG-*	HG2J-7UT22TF-*
Product Code	4010	4020
Product Name	"HG1J-4FT22TG-*"	"HG2J-7UT22TF-*"

Message Router Object (ClassID: 0x02)

An object to manage incoming messages.

Class

■**Class Services**

ID	Services
0EH	Get_Attribute_Single

■**Class Attributes (Instance ID: 0)**

ID	Attributes	Name	Data Type	Description	Value
1	R	Revision	UNIT	Object revisions	1
2	R	Max Instance	UNIT	Maximum instance number	1
3	R	Number of Instances	UNIT	Number of generated object instances	1
6	R	Maximum ID Number Class Attributes	UNIT	Maximum attribute ID number for class attributes	7
7	R	Maximum ID Number Instance Attributes	UNIT	Maximum attribute ID number of the instance attribute	2

Instance

■**Instance Services**

ID	Services
0EH	Get_Attribute_Single

■**Instance Attributes (Instance ID: 1)**

ID	Attributes	Name	Data Type	Description	Value
1	R	Object_list	STRUCT of:	List of objects	—
		Number	UINT	The number of classes supported in the class array	7
		Classes	UINT array	List of class codes	01H 02H 04H 06H F5H F6H 109H*1
2	R	Number Available	UNIT	Maximum number of connections	32

*1 FT2J/1J and HG2J/1J only

Assembly Object (ClassID: 04H)

Cyclic communication data objects.

Class

■**Class Services**

ID	Services
0EH	Get_Attribute_Single

■**Class Attributes (Instance ID: 0)**

ID	Attributes	Name	Data Type	Description	Value
1	R	Revision	UNIT	Object revisions	2

Instance

■**Instance Services**

ID	Services
0EH	Get_Attribute_Single
10H	Set_Attribute_Single

■**Instance Attributes (Instance ID: 100, 200)**

(1) Originator -> Target

ID	Attributes	Name	Data Type	Description	Value
3	R/W	Data	BYTE Array	Cyclic communication data assigned to InstanceID	Cyclic communication data
4	R	Size	UNIT	Cyclic communication data size allocated to InstanceID	Cyclic communication data size

(2) Target -> Originator

ID	Attributes	Name	Data Type	Description	Value
3	R	Data	BYTE Array	Cyclic communication data assigned to InstanceID	Cyclic communication data
4	R	Size	UNIT	Cyclic communication data size allocated to InstanceID	Cyclic communication data size

Connection Manager Object (ClassID: 06H)

An object used for connection-type communication. Used to open a connection to the device.

Class

■**Class Services**

There are no class services.

■**Class Attributes**

There are no class attributes.

Instance

■**Instance Services**

ID	Services
54H	Forward_Open
5BH	Large_Forward_Open
4EH	Forward_Close

■**Instance Attributes (Instance ID: 1)**

There are no instance attributes.

TCP/IP Interface Object (ClassID: F5H)

An object that provides a mechanism to set up a TCP/IP network interface.

Class

■Class Services

ID	Services
0EH	Get_Attribute_Single

■Class Attributes (Instance ID: 0)

ID	Attributes	Name	Data Type	Description	Value
1	R	Revision	UNIT	Object revisions	4
2	R	Max Instance	UNIT	Maximum instance number	1
3	R	Number of Instances	UNIT	Number of generated object instances	1
6	R	Maximum ID Number Class Attributes	UNIT	Maximum attribute ID number for class attributes	7
7	R	Maximum ID Number Instance Attributes	UNIT	Maximum attribute ID number of the instance attribute	13

Instance

■Instance Services

ID	Services
0EH	Get_Attribute_Single
10H	Set_Attribute_Single

■Instance Attributes (Instance ID: 1)

ID	Attributes	Name	Data Type	Description	Value
1	R	Status	DWORD	Interface Status 0: Communication information not set 1: Communication information has been set (nonvolatile memory) 2: Communication information has been set (hardware setting)	1
2	R	Configuration Capability	DWORD	Interface function flags bit2: With or without DHCP function (No: 0) bit4: Setting value can be changed or not (enabled: 0) bit5: Hardware settings such as IP address (enabled: 1) bit6:Reset after change or not (No: 0)	36(0024H)
3	R	Configuration Control	DWORD	Interface control flag 0x0000: use the previous IP address value	0
4	R	Physical Link Object	STRUCT of:	Path to physical layer link object 20F6=Ethernet Link object 2401=Instance 1	—
		Path size	UINT	Path size (WORD)	2
		Path	Padded EPATH	Segments to identify physical layer link objects	20F62401H
5	R	Interface Configuration	STRUCT of:	TCP/IP Network Interface Settings	—
		IP Address	UDINT	IP address of the device	Current IP address
		Network Mask	UDINT	Device Network Mask	Current Sub Net Mask
		Gateway Address	UDINT	Default Gateway Address	Current Gateway
		Name Server	UDINT	The primary name server	0 (not set)
		Name Server 2	UDINT	Secondary name servers	0 (not set)
		Domain Name	STRING	Domain name ASCII string: 48 characters max	0 (not set)
6	R	Host Name	STRING	Host Name	0 (not set)
13	R/W	Encapsulation Inactivity Timeout	UINT	encapsulation session timeout time (1 to 3600 seconds, default: 120 seconds)	Timeout value

Ethernet Link Object (ClassID: F6H)

An object that provides Ethernet status information.

Class

■Class Services

ID	Services
0EH	Get_Attribute_Single

■Class Attributes (Instance ID: 0)

ID	Attributes	Name	Data Type	Description	Value
1	R	Revision	UNIT	Object revisions	4
2	R	Max Instance	UNIT	Maximum instance number	1
3	R	Number of Instances	UNIT	Number of generated object instances	1

Instance

■Instance Services

ID	Services
0EH	Get_Attribute_Single
01H	Get_Attributes_All

■Instance Attributes (Instance ID: 1)

ID	Attributes	Name	Data Type	Description	Value
1	R	Interface Speed	UDINT	Interface communication speed (Mbps)	100
2	R	Interface Flags	DWORD	Interface status flag Full-duplex: 15 Half-duplex:13 Acquisition failure: 5	15
3	R	Physical Address	ARRAY of 6 USINTs	MAC layer address	MAC address value
10	R	Interface Label	SHORT-STRING	IP address of the device	Current IP address
11	R	Interface Capability	STRUCT of:	Interface function	—
		Capability Bits	DWORD	bit0: Reset after change or not (No: 0) bit1: Auto-negotiate function (Yes: 1) bit2: Auto-MDIX function (Yes: 1) bit3: Set value can be changed or not (No: 0)	6
		Speed/Duplex Options	STRUCT of:		—
		Speed/Duplex Array Count	USINT	Number of Speed/Duplex Array arrays	1
		Speed/Duplex Array	ARRAY of STRUCT of:		—
		Interface Speed	UINT	Interface communication speed (Mbps)	100
		Interface Duplex Mode	USINT	0: Half-duplex 1: Full-duplex	1

LLDP Management Object (ClassID: 109H)

An object that manages LLDP protocol.

Class

■**Class Services**

ID	Services
0EH	Get_Attribute_Single

■**Class Attributes (Instance ID: 0)**

ID	Attributes	Name	Data Type	Description	Value
1	R	Revision	UINT	Object revisions	1
2	R	Max Instance	UINT	Maximum instance number	1
3	R	Number of Instances	UINT	Number of generated object instances	1
6	R	Maximum ID Number Class Attributes	UINT	Maximum attribute ID number for class attributes	7
7	R	Maximum ID Number Instance Attributes	UINT	Maximum attribute ID number of the instance attribute	3

Instance

■**Instance Services**

ID	Services
0EH	Get_Attribute_Single
10H	Set_Attribute_Single

■ **Instance Attributes (Instance ID: 1)**

ID	Attributes	Name	Data Type	Description	Value
1	R/W	LLDP Enable	Struct of:	LLDP communication Enable/Disable	—
		LLDP Enable Array Length	UINT	Number of elements in the LLDP Enable Array	2
		LLDP Enable Array	ARRAY of: BYTE	bit0: Global setting bit1-N: Port transmission settings 0 = LLDP deactivated, 1 = LLDP activated	3
2	R/W	msgTxInterval	UINT	Interval in seconds after which the LLDP frames were sent from the device	1 to 3600 (Default: 30)
3	R/W	msgTxHold	USINT	A factor of msgTxInterval to determine the value of TTL TLV that was sent by the neighboring device	1 to 100 (Default: 4)

3: MQTT Communication

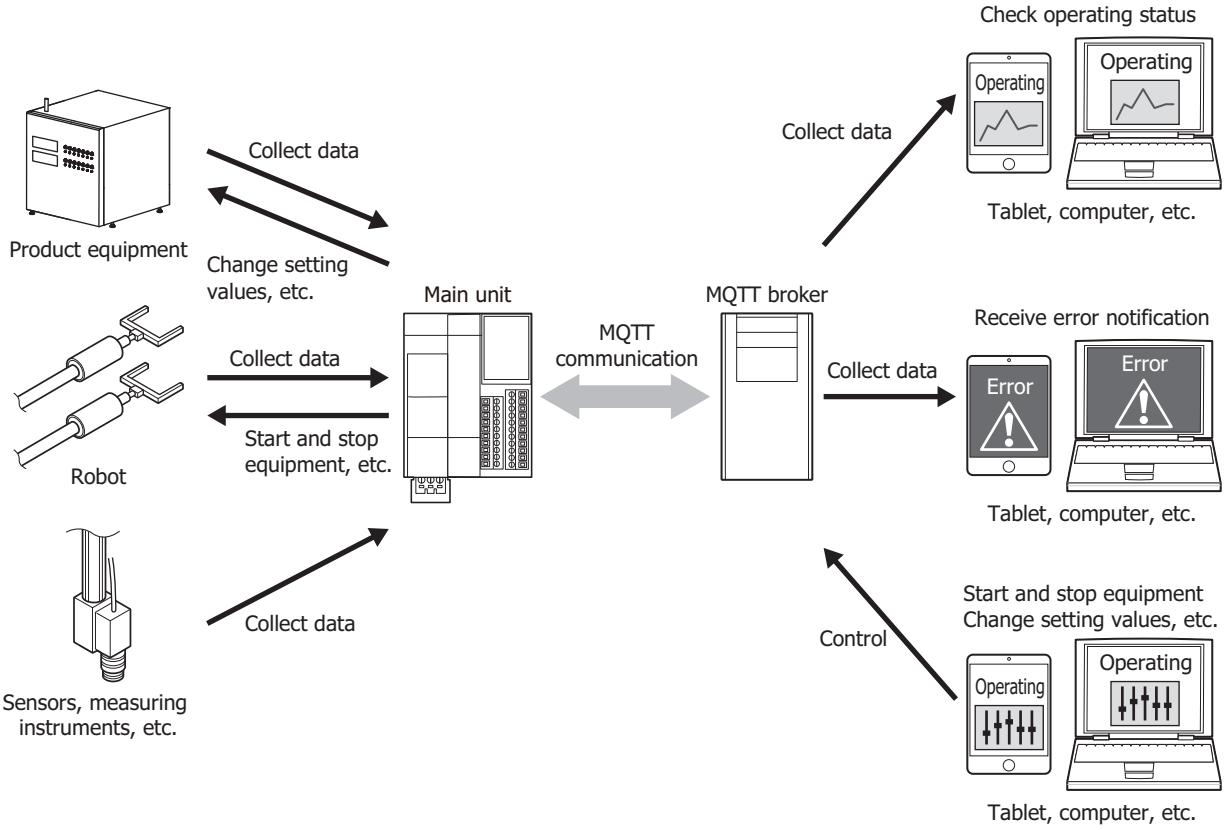
FC6A Plus FC6A All-in-One FT2J/1J HG2J/1J HG5G/4G/3G/2G-V

This chapter describes MQTT communication.

Overview

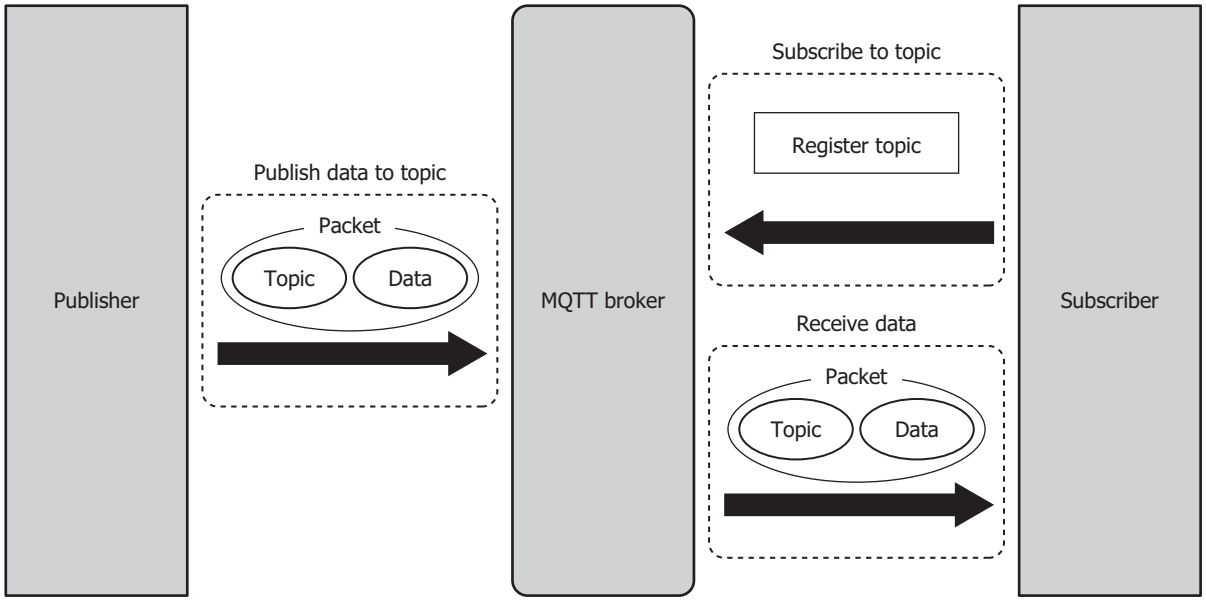
MQTT is a lightweight and easy-to-scale protocol used to send and receive data. You can accomplish the following by using MQTT communication.

- Collect data from remote equipment to check its operating status
- Receive notification when remote equipment has errors
- Start and stop remote equipment, and change the setting values



The main unit can communicate with a broker as a client (publisher and subscriber) in MQTT communication. For details on MQTT communication and descriptions of general terms used in MQTT communication, refer to the MQTT specifications published by OASIS (Organization for the Advancement of Structured Information Standards).

An MQTT network is composed of brokers, publishers, and subscribers. A subscriber connects to a broker and registers the topic to subscribe to. A publisher connects to a broker and transmits packets that include a topic and data to the broker. A broker refers to the topic inside a packet and transmits that packet to the subscribers that are subscribed to that topic, and those subscribers receive the data in that packet.



- A subscriber registering the topic to subscribe to with the broker is called "subscribing to a topic".
- A publisher transmitting packets that include a topic and data to a broker is called "publishing to a topic".
- The packets that are exchanged in MQTT communication are called MQTT packets.
- Topics are information that differentiate types of data. A publisher transmits data by adding the topic that corresponds to the type of data to transmit. A subscriber can receive only the data it needs by subscribing to the topic that corresponds to the type of data to receive.

Specifications

Supported models

Supported IDEC's PLCs and programmable displays are as follows.

Series Name	Abbreviation	Type Number
MICROSmart	FC6A	FC6A-D16**CEE, FC6A-D32**CEE
SmartAXIS	FT2J	FT2J-7U22*AF-B
	FT1J	FT1J-4F1**AG-*
MICRO/I	HG2J	HG2J-7UT22TF-B
	HG1J	HG1J-4FT22TG-*

Basic specifications

The standard specifications are as follows.

Model (Abbreviation)		FC6A		FT2J/1J	HG2J/1J
Communication Port		Ethernet port 1		Ethernet Interface	
Applicable Standards		MQTT Version 3.1.1	SparkPlug Version 3.0.0	MQTT Version 3.1.1	
Standard Specifications	Host Name	128 bytes maximum			
	Authentication	Supported			
	Account Name	128 bytes maximum			
	Password	496 bytes maximum			
	SSL/TLS	Supported			
	Client ID	128 bytes maximum			
	Keep Alive	5 to 65,535 s			
	QoS	0, 1, and 2			
	Topic	256 bytes maximum	239 bytes maximum	256 bytes maximum	
	Data Format	JSON format, 32 KB maximum (publisher), 8 KB maximum (subscriber)	Protocol Buffers format	JSON format, 32 KB maximum (publisher), 8 KB maximum (subscriber)	
Publish Specifications	Retain	Supported			
	Will	Not supported			
Subscribe Specifications	Wildcards (# and +)	Supported			
	Persistent Session	Not supported			

Supported MQTT Broker and Cloud Services

Basic operations (connecting, publishing, and subscribing) are confirmed using the following MQTT broker and cloud services. (As of July 2024)

- Mosquitto
- AWS IoT Core (Amazon.com) (port number 8883)[*1]
- Azure IoT Hub (azure.microsoft.com) (port number 8883)[*1]

You can use services supported by both the main unit and the above MQTT broker or cloud services. However, AWS IoT Core and Azure IoT Hub may not be able to connect due to changes in cloud service specifications.

The main unit also supports the following connection methods with MQTT broker and cloud services.

■Connect to a general purpose MQTT broker

Connect the main unit to a general purpose MQTT broker.

■Connect to AWS IoT Core

Connect the main unit that have been registered to AWS IoT Core in advance to it.

■Connect to Azure IoT Hub using SAS

Connect the main unit that have been registered to Azure IoT Hub in advance to it using Shared Access Signature (SAS). SAS token is used to connect the main unit to the Azure IoT Hub.

■Connect to Azure IoT Hub using X.509 certificate

Connect the main unit that have been registered to Azure IoT Hub in advance to it using X.509 certificate.

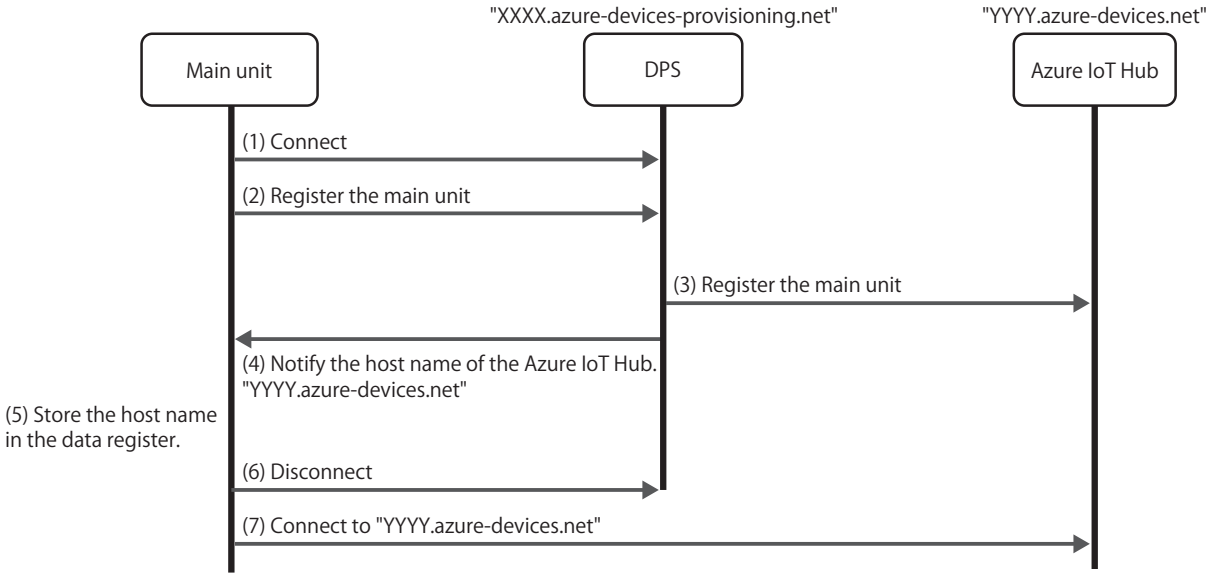
X.509 certificate is specified as the device certificate (client certificate).

■Connect to Azure IoT Hub via DPS

You can register main unit to Azure IoT Hub via Device Provisioning Service (DPS) and connect to Azure IoT Hub. SAS token is used to connect the main unit to the DPS and Azure IoT Hub.

*1 A connection cannot be made to port number 443 in AWS IoT Core (Amazon.com) / Azure IoT Hub (azure.microsoft.com).

The flow from connecting to DPS to connecting to Azure IoT Hub is shown in the figure below.



(1) The main unit connects to DPS.

(2) The main unit is registered to DPS.

(3) DPS registers the main unit to Azure IoT Hub.

(4) DPS notifies the main unit of the host name of the Azure IoT Hub to which the main unit connects.

(5) The main unit stores the host name in the data register.

(6) The main unit disconnects from DPS.

(7) The main unit connects to the Azure IoT Hub with the host name stored in the data registers in (5).

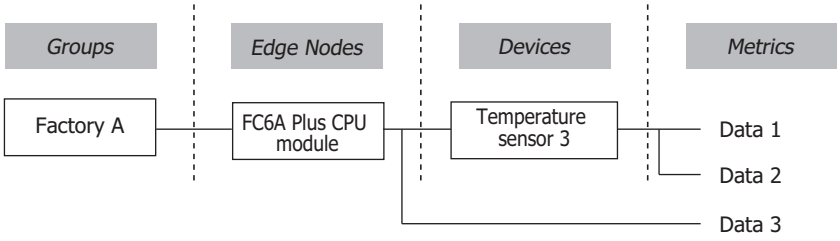
If the main unit connects again to DPS to which the main unit is already registered, the main unit may be registered to another Azure IoT Hub depending on the re-provision settings of the enrollment group of DPS. For more information, see the Azure IoT Hub Device Provisioning Service documentation.

■Connect to SparkPlug B broker

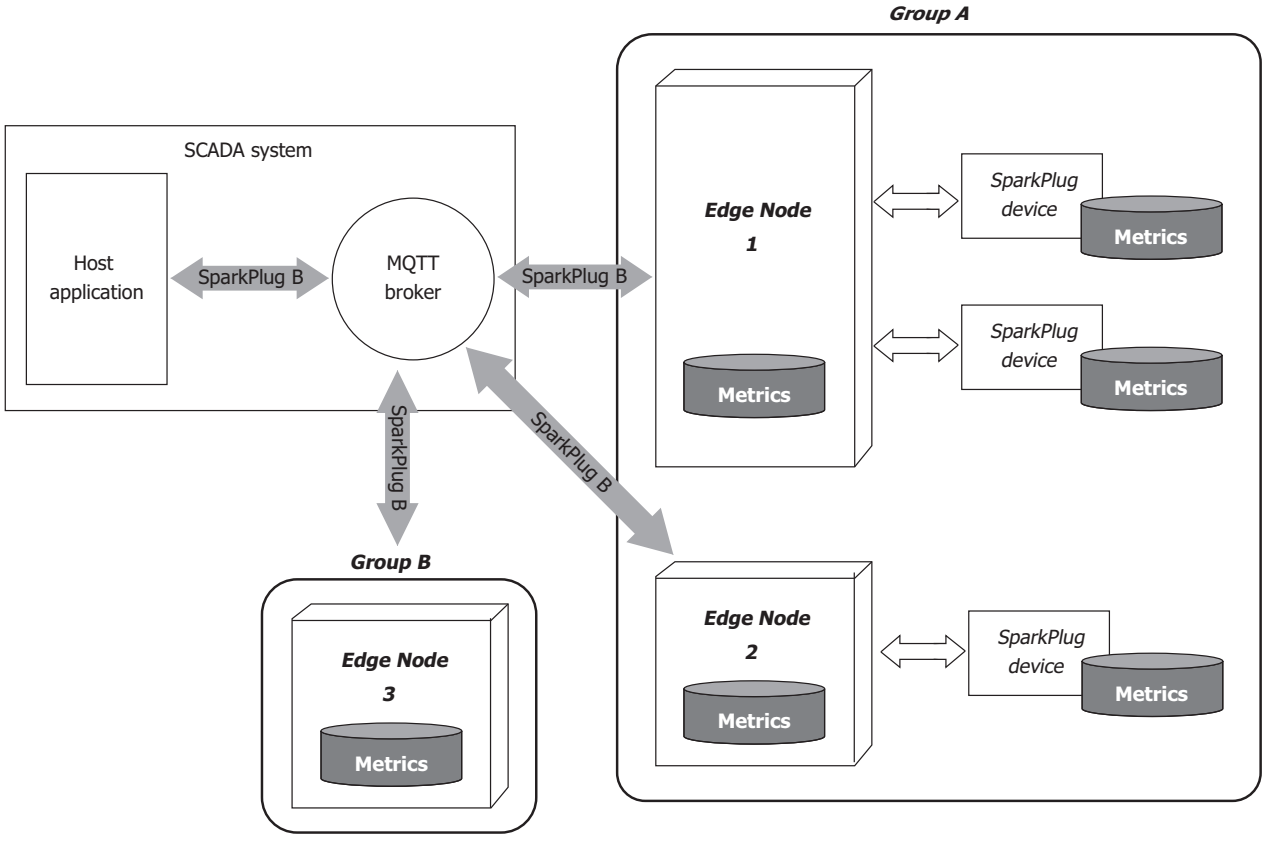
The main unit can be connected to an MQTT broker and communicate with host applications that support SparkPlug B.

Only FC6A Plus CPU module supports SparkPlug B.

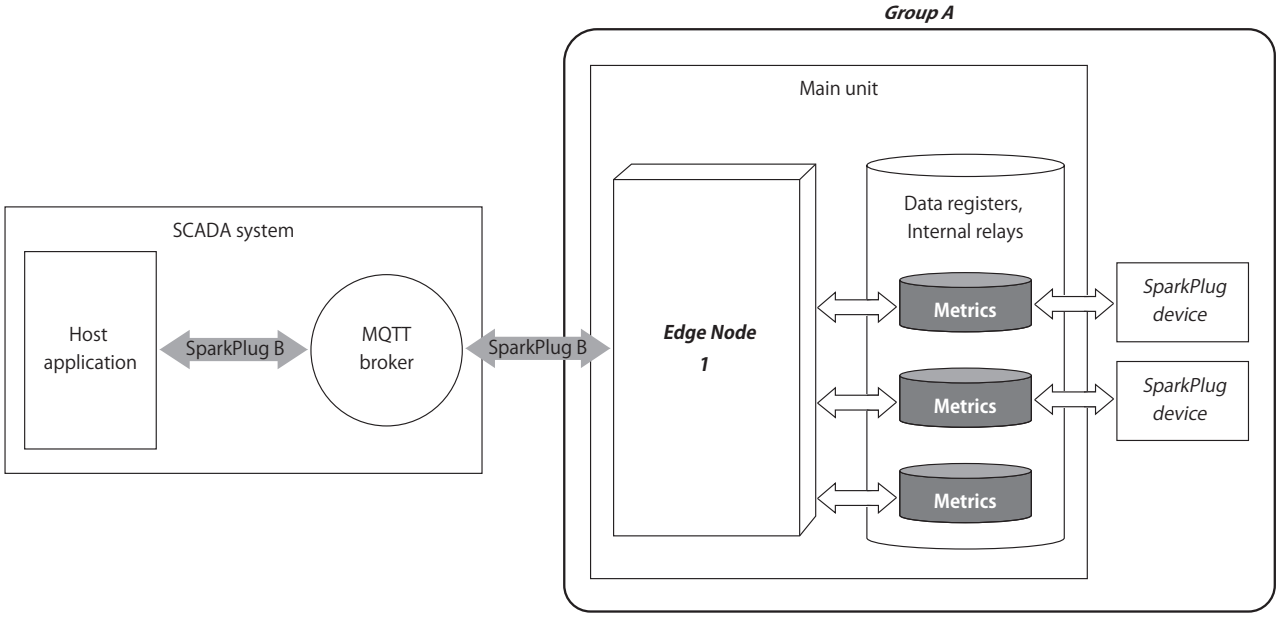
SparkPlug B is an industrial IoT protocol that uses MQTT. It specifies communication between applications such as SCADA and industrial devices and sensors. SparkPlug B defines four levels. From the top down, they are "Groups", "Edge Nodes", "Devices" and "Metrics".



The SparkPlug B system mainly consists of a host application, an MQTT broker, edge nodes, and SparkPlug devices. Edge nodes and devices have data called metrics. Edge nodes and host applications send and receive metrics over SparkPlug B via an MQTT broker. The physical SparkPlug devices are connected to the edge nodes via Modbus or some other means. Multiple edge nodes are managed together as a group.



By connecting to an MQTT broker, Main unit as an edge node can send and receive metrics to and from SparkPlug B host application. Own metrics and SparkPlug device metrics are managed through data registers and internal relays.



MQTT Settings Dialog Box

Set the MQTT broker or cloud service to which main unit connects and how the main unit connects.

(1) Enable MQTT Client function

Specifies whether to enable MQTT client function.

(2) Connect to

Sets the connection destination.

Connection to	Description
General purpose MQTT Broker	Connect the main unit to a general purpose MQTT broker. For details, see "Connect to a general purpose MQTT broker" on page 3-8.
AWS IoT Core	Connect the main unit to AWS IoT Core. For details, see "Connect to AWS IoT Core" on page 3-15.
Azure IoT Hub	Connect the main unit to Azure IoT Hub. For details, see "MQTT Connection Method" on page 3-7.
SparkPlug B	Use the payload specification of SparkPlug B to connect to an MQTT broker that supports SparkPlug B. Only FC6A Plus CPU module supports SparkPlug B. For details, see "SparkPlug B Settings" on page 3-51.

MQTT Settings

FC6A Plus FC6A All-in-One FT2J/1J HG2J/1J HG5G/4G/3G/2G-V

When you select **Enable MQTT Client function** and set **Connect to** to "General purpose", "AWS IoT Core", or "Azure IoT Hub", the **MQTT Settings** tab is displayed. Configure the basic settings for performing MQTT communication on this tab.

- When you set "SparkPlug B" in **Connect to**, the **SparkPlug B Settings** tab is displayed. Configure the basic settings for performing MQTT communication on this tab. For details of the **SparkPlug B Settings** tab, see "SparkPlug B Settings" on page 3-51.
- The basic settings for MQTT communication can be written to the main unit using an external memory drive. The external memory drive that can be used depends on the main unit type.

External Memory Drive	FC6A	FT2J/1J	HG2J/1J
SD memory card	Yes	No	No
USB flash drive*1	No	Yes	Yes

*1 For FT2J/1J and HG2J/1J, insert the USB flash drive into USB1.

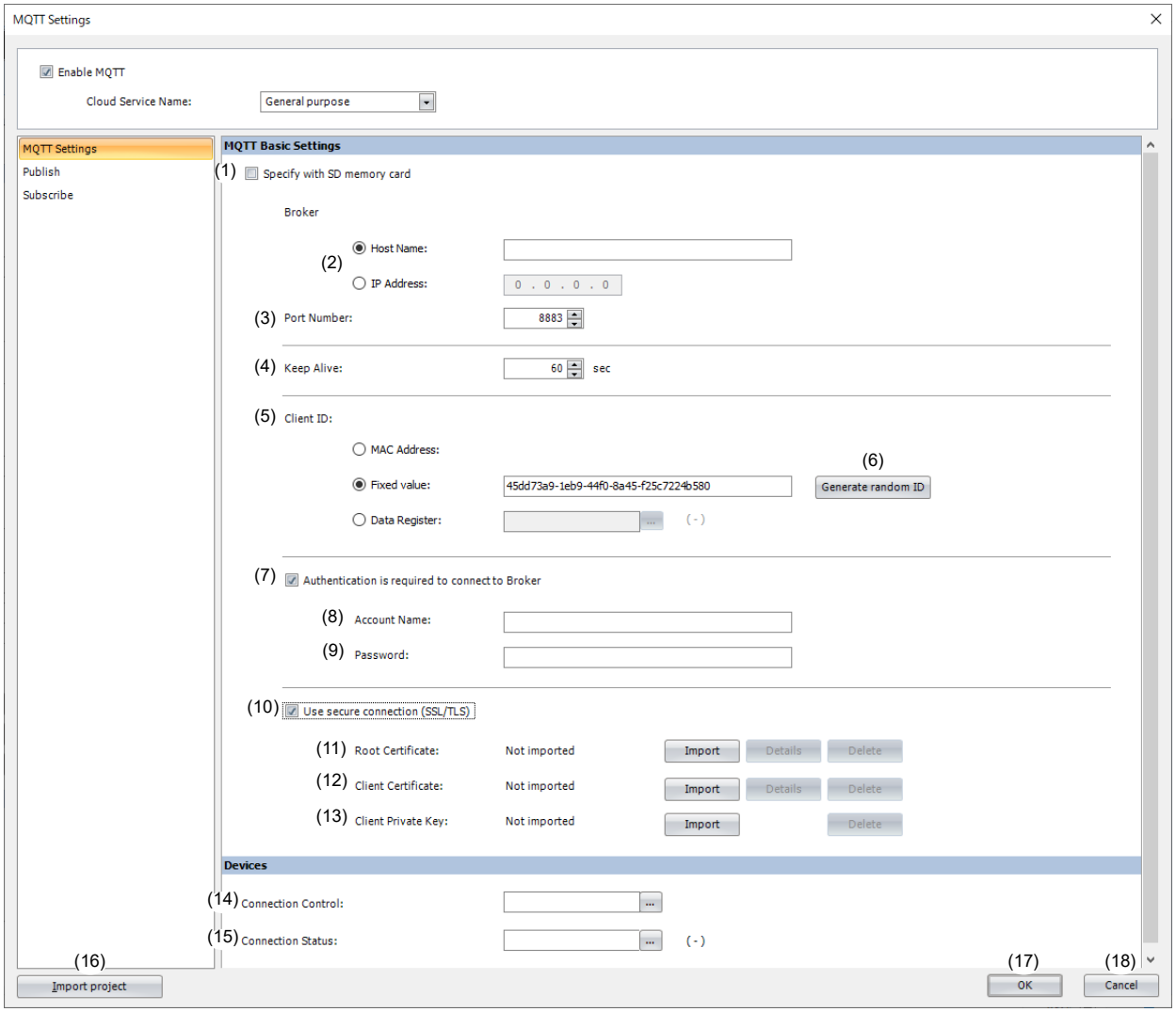
■MQTT Connection Method

Only when **Azure IoT Hub** is set as the **Connect to**, **MQTT Connection Method** group appears so that you can set the connection method to Azure IoT Hub. The connection method can be set from the following three types.

MQTT Connection Method		Description
Connect directly to Azure IoT Hub	Use Shared Access Signatures (SAS)	Connect to Azure IoT Hub using SAS. For details, see "Connect to Azure IoT Hub using SAS" on page 3-21.
	Use X.509 certificates	Connect to Azure IoT Hub using X.509 certificates. For details, see "Connect to Azure IoT Hub using X.509 certificates" on page 3-27.
Connect to Azure IoT Hub via DPS		Connect to Azure IoT Hub via DPS. For details, see "Connect to Azure IoT Hub via DPS" on page 3-32.

Connect to a general purpose MQTT broker

■MQTT Basic Settings



(1) Specify with SD memory card/Specify with USB1

Specifies whether to write the basic settings ((2) to (13)) for MQTT communication to the main unit by using an external memory drive. When you select this check box, you can write the basic settings for MQTT communication to the main unit using an external memory drive.

- **Configure SD Memory Card** button
  Configure USB1 button : Clicking this button opens the external memory drive dialog box.

  In the external memory drive dialog box, configure the basic settings for MQTT communication.
- For examples of how to write parameters when using an external memory drive, see "Restrictions" on page 3-90.

(2) Host Name/IP Address

Specifies the host name or IP address of the broker. The maximum length of the host name is 128 characters. Only alphanumeric characters and the symbol (-) can be used.
When you use an external memory drive, specify the host name or IP address in "hostname" key in the [MQTT_BROKER] section of the mqtt_basic_settings.ini file.

(3) Port Number

Specifies the port number of the broker. 1883 is normally used for MQTT and 8883 is normally used for MQTT over TLS. Set the port number between 0 and 65,535. The default value is 1883. If 0 is specified for the port number, port number 1883 is used if the **Use secure connection (SSL/TLS)** (10) check box is cleared, and port number 8883 is used if that check box is selected.

When you use an external memory drive, specify the port number in "port_number" key in the [MQTT_BROKER] section of the mqtt_basic_settings.ini file.

(4) Keep Alive

Specifies the time interval to execute a connection check for the broker when a connection is made to the broker. If the set time interval or longer continues without the main unit exchanging a packet, a connection check is performed with the broker. The range that can be set is 5 to 65,535 s. The default value is 60 s.

When you use an external memory drive, specify the setting in "keep_alive" key in the [MQTT_KEEP_ALIVE] section of the mqtt_basic_settings.ini file.

(5) Client ID

Specifies the client ID. The client ID can be set from the following three types.

Client ID	Description
MAC address	Sets the MAC address of the main unit interface.*1
Fixed value	Sets any character string. The maximum length of the client ID is 128 characters. Only alphanumeric characters and symbols (ASCII data character code 0x20 to 0x7E) can be used.
Data Resister	Sets the device addresses*2 for storing the client ID. The value is read in order from the upper byte of the specified device address, treated as character data, and used as the client ID.*3 Starting from the specified device address, 64 continuous words of device addresses are used.*4 Specify the first device address so that the device range is not exceeded. Only alphanumeric characters and symbols (ASCII data character code 0x20 to 0x7E) can be used.

*1 For example, when the MAC address of Ethernet port 1 of the Plus CPU module is 12-34-56-78-9A-BC (D8324=0012h, D8325=0034h, D8326=0056h, D8327=0078h, D8328=009Ah, D8329=00BCh), the client ID is "123456789ABC".

*2 The word device that can be specified depends on the main unit type.

Internal Device Name		Symbol	FC6A	FT2J/1J	HG2J/1J
Data Register		D	Yes	No	No
HMI Devices	HMI Data Register	LDR	No	Yes	Yes
	HMI Keep Register	LKR	No	Yes	Yes
	HMI Temporary Register	LBR	No	Yes	Yes
Control Devices	Data Register	D	No	Yes	No

For details, see the following manuals.
FC6A: Chapter 6 "Devices" in the "FC6A Series MICROSmart User's Manual".
FT2J/1J, HG2J/1J: Chapter 35 "2 Word Devices" in the "WindO/I-NV4 User's Manual".

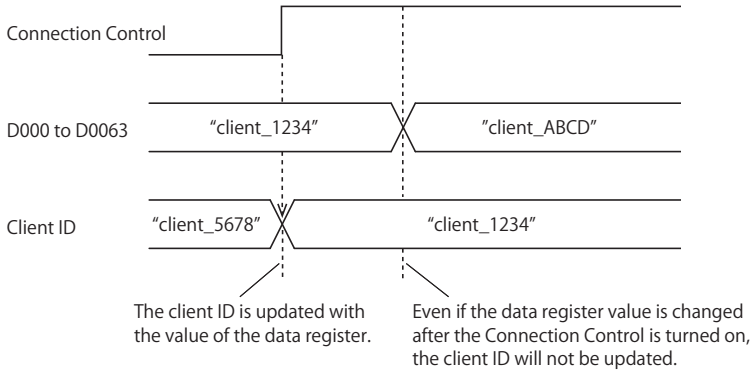
*3 For example, when D0000 is specified and the following values are stored in each data register, the client ID is "client_1234".

Data resister	Stored Value	
	Upper Byte	Lower Byte
D0000	"c"=63h	"l"=6Ch
D0001	"i"=69h	"e"=65h
D0002	"n"=6Eh	"t"=74h
D0003	"_"=5Fh	"1"=31h
D0004	"2"=32h	"3"=33h
D0005	"4"=34h	00h

*4 When the character string to be set is shorter than 128 bytes (64 words), add the terminating character NULL (00h) as the end of the character string. The data from the upper byte of the specified device address up to the data right before the terminating character NULL (00h) is treated as character data and used as the client ID.

When you use an external memory drive, specify the client ID in the [MQTT_CLIENT_ID] section of the mqtt_basic_settings.ini file.

- When setting the client ID with the data register, the MOVC instruction can be used to store a character string in the data registers. For details on MOVC instruction, see Chapter 5 "MOVC (Move Characters)" in the "Ladder Programming Manuall".
- When the Connection Control (14) is changed from off to on, the client ID stored inside the main unit is updated. For example, when the client ID is set in the data register (D0000), the client ID is updated at the following timing.



(6) Generate random ID

When the **Client ID** (5) is set as a fixed value, click this button to generate a random identifier as client ID.

(7) Authentication is required to connect to Broker

Specifies whether to perform authentication using an account name and password when connecting to the broker. If the check box is selected, authentication is performed using an account name and password when connecting to the broker. The check box is cleared by default.

When you use an external memory drive, specify the setting in "authentication" key in the [MQTT_AUTH] section of the mqtt_basic_settings.ini file.

(8) Account Name / (9) Password

Sets the account name and password when the **Authentication is required to connect to Broker** check box is selected. The maximum length of the account name is 128 characters, and the maximum length of the password is 496 characters. Only alphanumeric characters and symbols can be used.

When you use an external memory drive, specify the account name and password in "accountname" and "password" keys respectively in the [MQTT_AUTH] section of the mqtt_basic_settings.ini file.

(10) Use secure connection (SSL/TLS)

Select this check box when SSL/TLS communication is required with the destination broker. The check box is cleared by default.

When you use an external memory drive, specify the setting in "use_secure_connection" key in the [MQTT_TLS] section of the mqtt_basic_settings.ini file.

(11) Root Certificate

When the **Use secure connection (SSL/TLS)** check box is selected, you can click **Import** button and import a root certificate (pem or crt file with a size less than 2 KB (after conversion to der file)) of the server certificate to use when performing SSL/TLS communication with the broker. WindLDR and WindO/I-NV4 convert pem and crt files to der files. When a pem file contains multiple root certificates, WindLDR and WindO/I-NV4 convert each of the first two root certificates to a der file.

Click **Details** button to display the **Certificate Information** dialog box that shows the root certificate of the imported server certificate. Click **Delete** button to delete the imported certificate.

When you use an external memory drive, specify the setting in "root_certificate" key in the [MQTT_TLS] section of the mqtt_basic_settings.ini file. If a root certificate was imported, the root_certificate key is true and an mqtt_root_certificate.der file is created. If the imported pem file contains multiple root certificates, the second root certificate from the top is be generated as the mqtt_root_certificate_2.der file. If a root certificate was not imported, the root_certificate key is false and an mqtt_root_certificate.der file is not created.

(12) Client Certificate

When the **Use secure connection (SSL/TLS)** check box is selected, you can click **Import** button and import a client certificate (pem or crt file with a size less than 2 KB (after conversion to der file)) to use when performing SSL/TLS communication with the broker. WindLDR and WindO/I-NV4 convert pem and crt files to der files.

Click **Details** button to display the **Certificate Information** dialog box that shows the imported client certificate. Click **Delete** button to delete the imported certificate.

When you use an external memory drive, specify the setting in "client_certificate" key in the [MQTT_TLS] section of the mqtt_basic_settings.ini file. If a client certificate was imported, the client_certificate key is true and an mqtt_client_certificate.der file is created. If a client certificate was not imported, the client_certificate key is false and an mqtt_client_certificate.der file is not created.

(13) Client Private Key

When the **Use secure connection (SSL/TLS)** check box is selected, you can click **Import** button and import a client private key (RSA encryption, key length 3,072 bits max.) to use when performing SSL/TLS communication with the broker. WindLDR and WindO/I-NV4 support the following file formats.

- PKCS#1 format pem file or der file
- PKCS#8 format (unencrypted) pem file or der file

Click **Delete** button to delete the imported private key.

When you use an external memory drive, specify the setting in "client_private_key"key in the [MQTT_TLS] section of the mqtt_basic_settings.ini file. If a client private key was imported, the client_private_key key is true and an mqtt_private_key.der file is created. If a client private key was not imported, the client_private_key key is false and an mqtt_private_key.der file is not created.

■**Devices**

(14) Connection Control

Specifies the device address used to connect to and disconnect from the broker. The bit device that can be set depends on the main unit type.

Internal Device Name		Symbol	FC6A	FT2J/1J	HG2J/1J
Internal relay		M	Yes	No	No
HMI Devices	HMI Internal Relay	LM	No	Yes	Yes
	HMI Keep Relay	LK	No	Yes	Yes
	HMI Temporary Relay	LBM	No	Yes	Yes
Control Devices	Internal relay	M	No	Yes	No

For details, see the following manuals.
FC6A: Chapter 6 "Devices" in the "FC6A Series MICROSmart User's Manual".
FT2J/1J, HG2J/1J: Chapter 35 "1 Bit Devices" in the "WindO/I-NV4 User's Manual".

When the set device address is turned on, a connection is made to the broker. When the set device address is turned off, the connection with the broker is disconnected.

When using the data registers to set the client ID (5), the client ID stored inside the main unit is updated when the connection control is changed from off to on.

(15) Connection Status

Specifies the device addresses that store the status when connecting to the broker and the error information. The word device that can be specified depends on the main unit type.

Internal Device Name		Symbol	FC6A	FT2J/1J	HG2J/1J
Data Register		D	Yes	No	No
HMI Devices	HMI Data Register	LDR	No	Yes	Yes
	HMI Keep Register	LKR	No	Yes	Yes
	HMI Temporary Register	LBR	No	Yes	Yes
Control Devices	Data Register	D	No	Yes	No

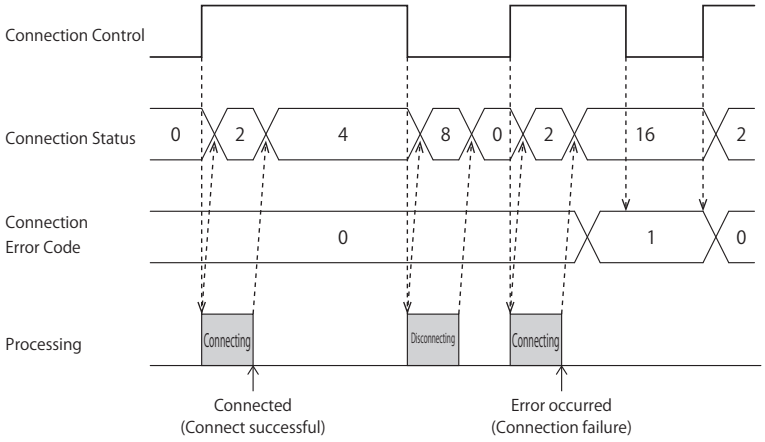
For details, see the following manuals.
FC6A: Chapter 6 "Devices" in the "FC6A Series MICROSmart User's Manual".
FT2J/1J, HG2J/1J: Chapter 35 "2 Word Devices" in the "WindO/I-NV4 User's Manual".

Starting from the specified device address, 2 words of device addresses are used. Specify the first device address so that the device range is not exceeded.

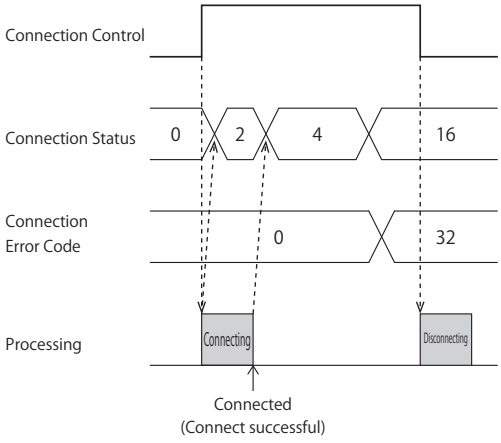
<table>
<tr><th>Storage Destination</th><th>Item</th><th>Description</th></tr>
<tr><td>Starting number+0</td><td>Connection Status</td><td>Stores the status when connecting to the broker.
<table>
<tr><th>Status Code</th><th>Status</th></tr>
<tr><td>0 (0000h)</td><td>Initial status (disconnected)</td></tr>
<tr><td>2 (0002h)</td><td>Connecting</td></tr>
<tr><td>4 (0004h)</td><td>Connected</td></tr>
<tr><td>8 (0008h)</td><td>Disconnecting</td></tr>
<tr><td>16 (0010h)</td><td>Connection processing error</td></tr>
<tr><td>32 (0020h)</td><td>Disconnection processing error</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Storage Destination</th><th>Item</th><th>Description</th></tr>
<tr><td>Starting number+1</td><td>Connection Error Code</td><td>Stores information about the error that occurred when connecting to the broker.
<table>
<tr><th>Error Code</th><th>Error Details</th></tr>
<tr><td>1 (0001h)</td><td>The Ethernet cable is disconnected or broken and the main unit cannot connect to the network properly</td></tr>
<tr><td>2 (0002h)</td><td>Authentication information was not downloaded from the external memory drive or reading the downloaded authentication information failed</td></tr>
<tr><td>4 (0004h)</td><td>Invalid client ID format</td></tr>
<tr><td>16 (0010h)</td><td>An unknown packet was received</td></tr>
<tr><td>32 (0020h)</td><td>An invalid MQTT packet was received</td></tr>
<tr><td>64 (0040h)</td><td>Keep alive timeout error</td></tr>
<tr><td>80 (0050h)</td><td>Packet could not arrive at destination host</td></tr>
<tr><td>96 (0060h)</td><td>MQTT packet receive timeout error</td></tr>
<tr><td>112 (0070h)</td><td>TLS error</td></tr>
<tr><td>256 (0100h)</td><td>Broker connection refused (unacceptable MQTT protocol version)</td></tr>
<tr><td>512 (0200h)</td><td>Broker connection refused (invalid client ID)</td></tr>
<tr><td>768 (0300h)</td><td>Broker connection refused (broker unavailable)</td></tr>
<tr><td>1024 (0400h)</td><td>Broker connection refused (invalid account name or password)</td></tr>
<tr><td>1280 (0500h)</td><td>Broker connection refused (not authorized)</td></tr>
<tr><td>32768 (8000h)</td><td>Broker response error</td></tr>
</table></td></tr>
</table>

- The following timing chart shows the operations when the main unit connects to a broker.



- Connection Error Code 16 (0010h) to 112 (0070h) may occur even when the main unit is connected to the broker (Connection Status 4 (0004h)).



- MQTT communication is not performed while the ladder program is stopped. When switching from run to stop when connected to a broker, the Connection Control relay is turned off and disconnected 0 (0000h) is stored in the Connection Status.

(16) Import from project button

Imports only the MQTT settings from a project file.

(17) OK button

Click **OK** button to save the settings and close the **MQTT Settings** dialog box.

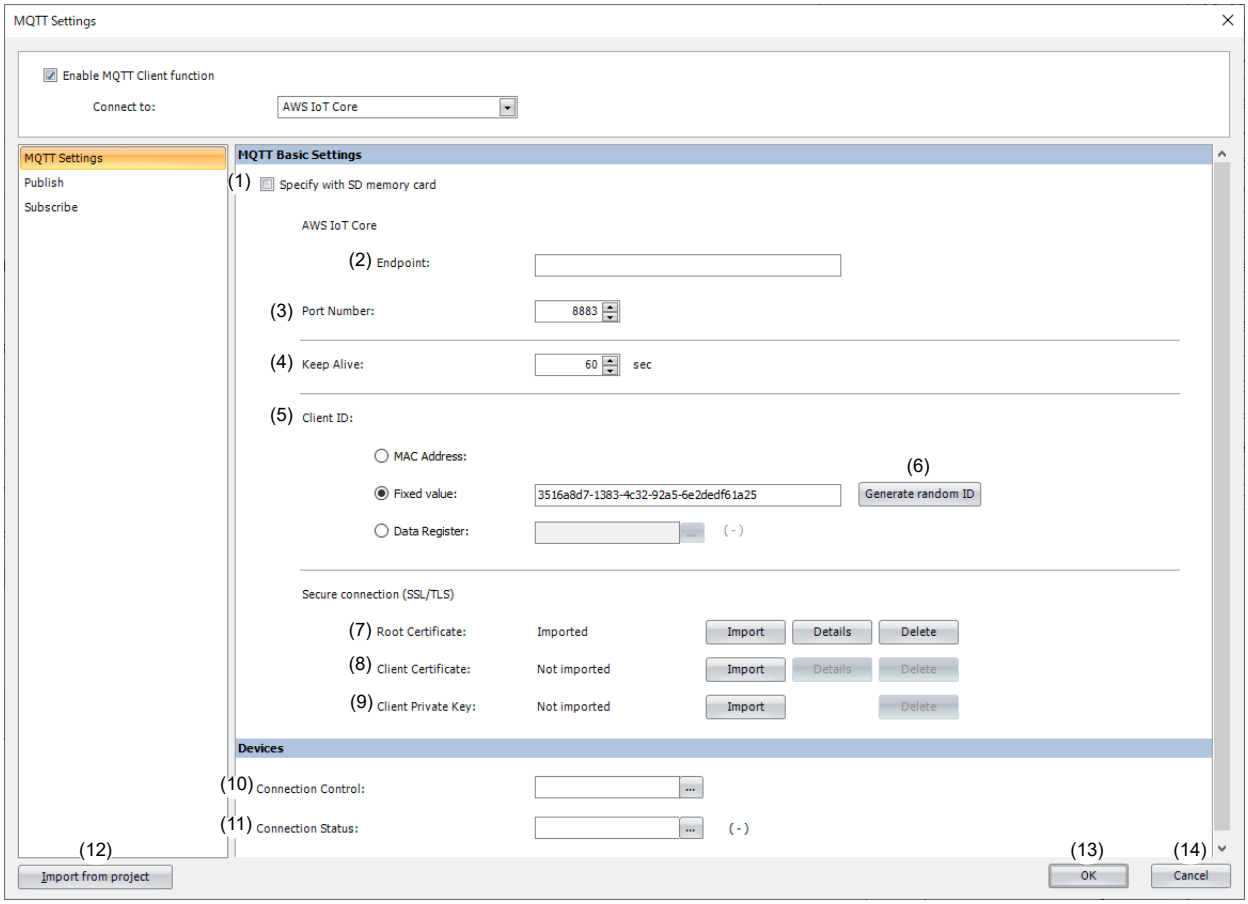
When using the data registers to set the client ID (5), pressing the **OK** button calculates the size of the topic considering the client ID as 1 byte. If the set topic exceeds 256 bytes, an error will occur.

(18) Cancel button

Click **Cancel** button to close the **MQTT Settings** dialog box without saving the settings.

Connect to AWS IoT Core

■MQTT Basic Settings



(1) Specify with SD memory card/Specify with USB1

Specifies whether to write the basic settings ((2) to (9)) for MQTT communication to the main unit by using an external memory drive. When you select this check box, you can write the basic settings for MQTT communication to the main unit using an external memory drive.

- **Configure SD Memory Card** button
  Configure USB1 button
  : Clicking this button opens the external memory drive dialog box.

  In the external memory drive dialog box, configure the basic settings for MQTT communication.
- For examples of how to write parameters when using an external memory drive, see "Restrictions" on page 3-90.

(2) Endpoint

Specifies the endpoint of the AWS IoT Core. The maximum length of the endpoint is 128 characters. Only alphanumeric characters and the symbol (-) can be used.
When you use an external memory drive, specify the endpoint in "hostname" key in the [MQTT_BROKER] section of the mqtt_basic_settings.ini file.

(3) Port Number

Sets the port number of the AWS IoT Core. Set the port number between 0 and 65,535. The default value is 8883 used for AWS IoT Core. If 0 is specified for the port number, port number 8883 is used.
When you use an external memory drive, specify the port number in "port_number" key in the [MQTT_BROKER] section of the mqtt_basic_settings.ini file.

(4) Keep Alive

Specifies the time interval to execute a connection check for the AWS IoT Core when a connection is made to the AWS IoT Core. If the set time interval or longer continues without the main unit exchanging a packet, a connection check is performed with the AWS IoT Core. The range that can be set is 5 to 65,535 s. The default value is 60 s.
When you use an external memory drive, specify the setting in "keep_alive" key in the [MQTT_KEEP_ALIVE] section of the mqtt_basic_settings.ini file.

(5) Client ID

Specifies the client ID. The client ID can be set from the following three types.

Client ID	Description
MAC address	Sets the MAC address of the main unit interface.*1
Fixed value	Sets any character string. The maximum length of the client ID is 128 characters. Only alphanumeric characters and symbols (ASCII data character code 0x20 to 0x7E) can be used.
Data Resister	Sets the device addresses*2 for storing the client ID. The value is read in order from the upper byte of the specified device address, treated as character data, and used as the client ID.*3 Starting from the specified device address, 64 continuous words of device addresses are used.*4 Specify the first device address so that the device range is not exceeded. Only alphanumeric characters and symbols (ASCII data character code 0x20 to 0x7E) can be used.

*1 For example, when the MAC address of Ethernet port 1 of the Plus CPU module is 12-34-56-78-9A-BC (D8324=0012h, D8325=0034h, D8326=0056h, D8327=0078h, D8328=009Ah, D8329=00BCh), the client ID is "123456789ABC".

*2 The word device that can be specified depends on the main unit type.

Internal Device Name		Symbol	FC6A	FT2J/1J	HG2J/1J
Data Register		D	Yes	No	No
HMI Devices	HMI Data Register	LDR	No	Yes	Yes
	HMI Keep Register	LKR	No	Yes	Yes
	HMI Temporary Register	LBR	No	Yes	Yes
Control Devices	Data Register	D	No	Yes	No

For details, see the following manuals.
FC6A: Chapter 6 "Devices" in the "FC6A Series MICROSmart User's Manual".
FT2J/1J, HG2J/1J: Chapter 35 "2 Word Devices" in the "WindO/I-NV4 User's Manual".

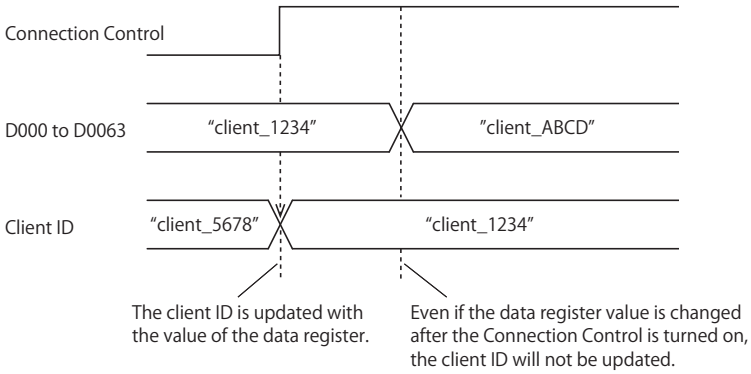
*3 For example, when D0000 is specified and the following values are stored in each data register, the client ID is "client_1234".

Data resister	Stored Value	
	Upper Byte	Lower Byte
D0000	"c"=63h	"l"=6Ch
D0001	"i"=69h	"e"=65h
D0002	"n"=6Eh	"t"=74h
D0003	"_"=5Fh	"1"=31h
D0004	"2"=32h	"3"=33h
D0005	"4"=34h	00h

*4 When the character string to be set is shorter than 128 bytes (64 words), add the terminating character NULL (00h) as the end of the character string. The data from the upper byte of the specified device address up to the data right before the terminating character NULL (00h) is treated as character data and used as the client ID.

Specify the client ID in the [MQTT_CLIENT_ID] section of the mqtt_basic_settings.ini file.

- When setting the client ID with the data register, the MOVC instruction can be used to store a character string in the data registers. For details on MOVC instruction, see Chapter 5 "MOVC (Move Characters)" in the "Ladder Programming Manuall".
- When the Connection Control (10) is changed from off to on, the client ID stored inside the main unit is updated. For example, when the client ID is set in the data register (D0000), the client ID is updated at the following timing.



(6) Generate random ID

When the **Client ID** (5) is set as a fixed value, click this button to generate a random identifier as client ID.

(7) Root Certificate

Click **Import** button to import a root certificate to use when performing SSL/TLS communication with the AWS IoT Core. Files that can be imported are pem or crt files. WindLDR and WindO/I-NV4 convert pem and crt files to der files. der files that are 2KB or bigger cannot be imported. When a pem file contains multiple root certificates, WindLDR and WindO/I-NV4 convert each of the first two root certificates to a der file.

Click **Details** button to display the **Certificate Information** dialog box that shows the imported root certificate. Click **Delete** button to delete the imported certificate.

When you use an external memory drive, specify the setting in "root_certificate" key in the [MQTT_TLS] section of the mqtt_basic_settings.ini file. If a root certificate was imported, the root_certificate key is true and an mqtt_root_certificate.der file is created. If the imported pem file contains multiple root certificates, the second root certificate from the top is be generated as the mqtt_root_certificate_2.der file. If a root certificate was not imported, the root_certificate key is false and an mqtt_root_certificate.der file is not created.

(8) Client Certificate

Click **Import** button to import a client certificate to use when performing SSL/TLS communication with the AWS IoT Core. Files that can be imported are pem or crt files. WindLDR and WindO/I-NV4 convert pem and crt files into der files. der files that are 2KB or bigger cannot be imported.

Click **Details** button to display the **Certificate Information** dialog box that shows the imported client certificate. Click **Delete** button to delete the imported certificate.

When you use an external memory drive, specify the setting in "client_certificate" key in the [MQTT_TLS] section of the mqtt_basic_settings.ini file. If a client certificate was imported, the "client_certificate" key is true and an mqtt_client_certificate.der file is created. If a client certificate was not imported, the "client_certificate" key is false and an mqtt_client_certificate.der file is not created.

(9) Client Private Key

Click **Import** button to import a client private key (RSA encryption, key length 3,072 bits max.) to use when performing SSL/TLS communication with the AWS IoT Core. WindLDR and WindO/I-NV4 support the following file formats.

- PKCS#1 format pem file or der file
- PKCS#8 format (unencrypted) pem file or der file

Click **Delete** button to delete the imported private key.

When you use an external memory drive, specify the setting in "client_private_key"key in the [MQTT_TLS] section of the mqtt_basic_settings.ini file. If a client private key was imported, the "client_private_key" key is true and an mqtt_private_key.der file is created. If a client private key was not imported, the "client_private_key" key is false and an mqtt_private_key.der file is not created.

■**Devices**

(10) Connection Control

Specifies the device address used to connect to and disconnect from the AWS IoT Core. The bit device that can be set depends on the main unit type.

Internal Device Name		Symbol	FC6A	FT2J/1J	HG2J/1J
Internal relay		M	Yes	No	No
HMI Devices	HMI Internal Relay	LM	No	Yes	Yes
	HMI Keep Relay	LK	No	Yes	Yes
	HMI Temporary Relay	LBM	No	Yes	Yes
Control Devices	Internal relay	M	No	Yes	No

For details, see the following manuals.
FC6A: Chapter 6 "Devices" in the "FC6A Series MICROSmart User's Manual".
FT2J/1J, HG2J/1J: Chapter 35 "1 Bit Devices" in the "WindO/I-NV4 User's Manual".

When the set device address is turned on, a connection is made to the AWS IoT Core. When the set device address is turned off, the connection with the AWS IoT Core is disconnected.

When using the data registers to set the client ID (5), the client ID stored inside the main unit is updated when the connection control is changed from off to on.

(11) Connection Status

Specifies the device addresses that store the status when connecting to the AWS IoT Core and the error information. The word device that can be specified depends on the main unit type.

Internal Device Name		Symbol	FC6A	FT2J/1J	HG2J/1J
Data Register		D	Yes	No	No
HMI Devices	HMI Data Register	LDR	No	Yes	Yes
	HMI Keep Register	LKR	No	Yes	Yes
	HMI Temporary Register	LBR	No	Yes	Yes
Control Devices	Data Register	D	No	Yes	No

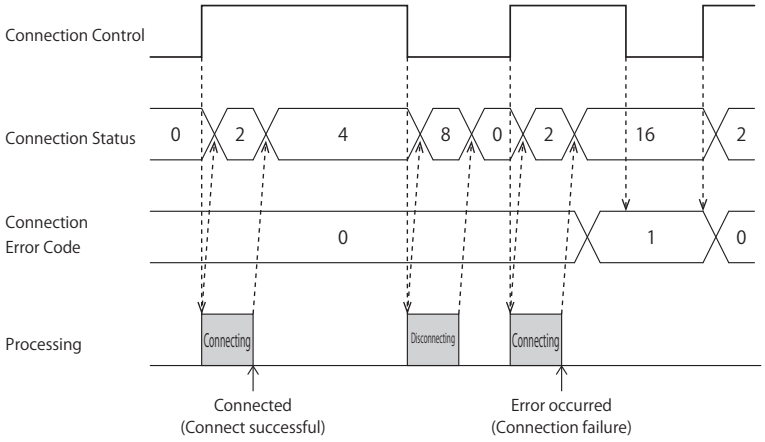
For details, see the following manuals.
FC6A: Chapter 6 "Devices" in the "FC6A Series MICROSmart User's Manual".
FT2J/1J, HG2J/1J: Chapter 35 "2 Word Devices" in the "WindO/I-NV4 User's Manual".

Starting from the specified device address, 2 words of device addresses are used. Specify the first device address so that the device range is not exceeded.

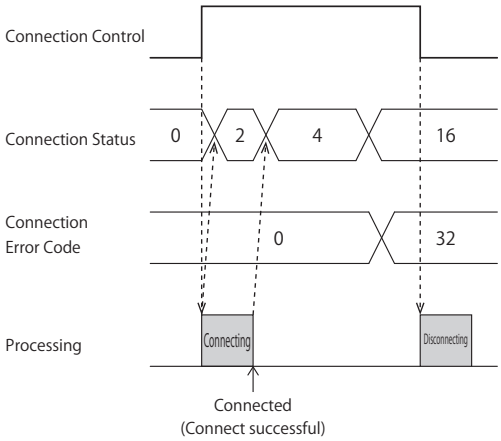
<table>
<tr><th>Storage Destination</th><th>Item</th><th>Description</th></tr>
<tr><td>Starting number+0</td><td>Connection Status</td><td>Stores the status when connecting to the AWS IoT Core.
<table>
<tr><th>Status Code</th><th>Status</th></tr>
<tr><td>0 (0000h)</td><td>Initial status (disconnected)</td></tr>
<tr><td>2 (0002h)</td><td>Connecting</td></tr>
<tr><td>4 (0004h)</td><td>Connected</td></tr>
<tr><td>8 (0008h)</td><td>Disconnecting</td></tr>
<tr><td>16 (0010h)</td><td>Connection processing error</td></tr>
<tr><td>32 (0020h)</td><td>Disconnection processing error</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Storage Destination</th><th>Item</th><th>Description</th></tr>
<tr><td>Starting number+1</td><td>Connection Error Code</td><td>Stores information about the error that occurred when connecting to the AWS IoT Core.
<table>
<tr><th>Error Code</th><th>Error Details</th></tr>
<tr><td>1 (0001h)</td><td>The Ethernet cable is disconnected or broken and the main unit cannot connect to the network properly</td></tr>
<tr><td>2 (0002h)</td><td>Authentication information was not downloaded from the external memory or reading the downloaded authentication information failed</td></tr>
<tr><td>4 (0004h)</td><td>Invalid client ID format</td></tr>
<tr><td>16 (0010h)</td><td>An unknown packet was received</td></tr>
<tr><td>32 (0020h)</td><td>An invalid MQTT packet was received</td></tr>
<tr><td>64 (0040h)</td><td>Keep alive timeout error</td></tr>
<tr><td>80 (0050h)</td><td>Packet could not arrive at destination host</td></tr>
<tr><td>96 (0060h)</td><td>MQTT packet receive timeout error</td></tr>
<tr><td>112 (0070h)</td><td>TLS error</td></tr>
<tr><td>256 (0100h)</td><td>AWS IoT Core connection refused (unacceptable MQTT protocol version)</td></tr>
<tr><td>512 (0200h)</td><td>AWS IoT Core connection refused (invalid client ID)</td></tr>
<tr><td>768 (0300h)</td><td>AWS IoT Core connection refused (AWS IoT Core unavailable)</td></tr>
<tr><td>1024 (0400h)</td><td>AWS IoT Core connection refused (invalid account name or password)</td></tr>
<tr><td>1280 (0500h)</td><td>AWS IoT Core connection refused (not authorized)</td></tr>
<tr><td>32768 (8000h)</td><td>AWS IoT Core response error</td></tr>
</table></td></tr>
</table>

- The following timing chart shows the operations when the main unit connects to AWS IoT Core.



- Connection Error Code 16 (0010h) to 112 (0070h) may occur even when the main unit is connected to the AWS IoT Core (Connection Status 4 (0004h)).



- MQTT communication is not performed while the ladder program is stopped. When switching from run to stop when connected to AWS IoT Core, the Connection Control relay is turned off and disconnected 0 (0000h) is stored in the Connection Status.

(12) Import from project button

Imports only the MQTT settings from a project file.

(13) OK button

Click **OK** button to save the settings and close the **MQTT Settings** dialog box.

When using the data registers to set the client ID (5), pressing the **OK** button calculates the size of the topic considering the client ID as 1 byte. If the set topic exceeds 256 bytes, an error will occur.

(14) Cancel button

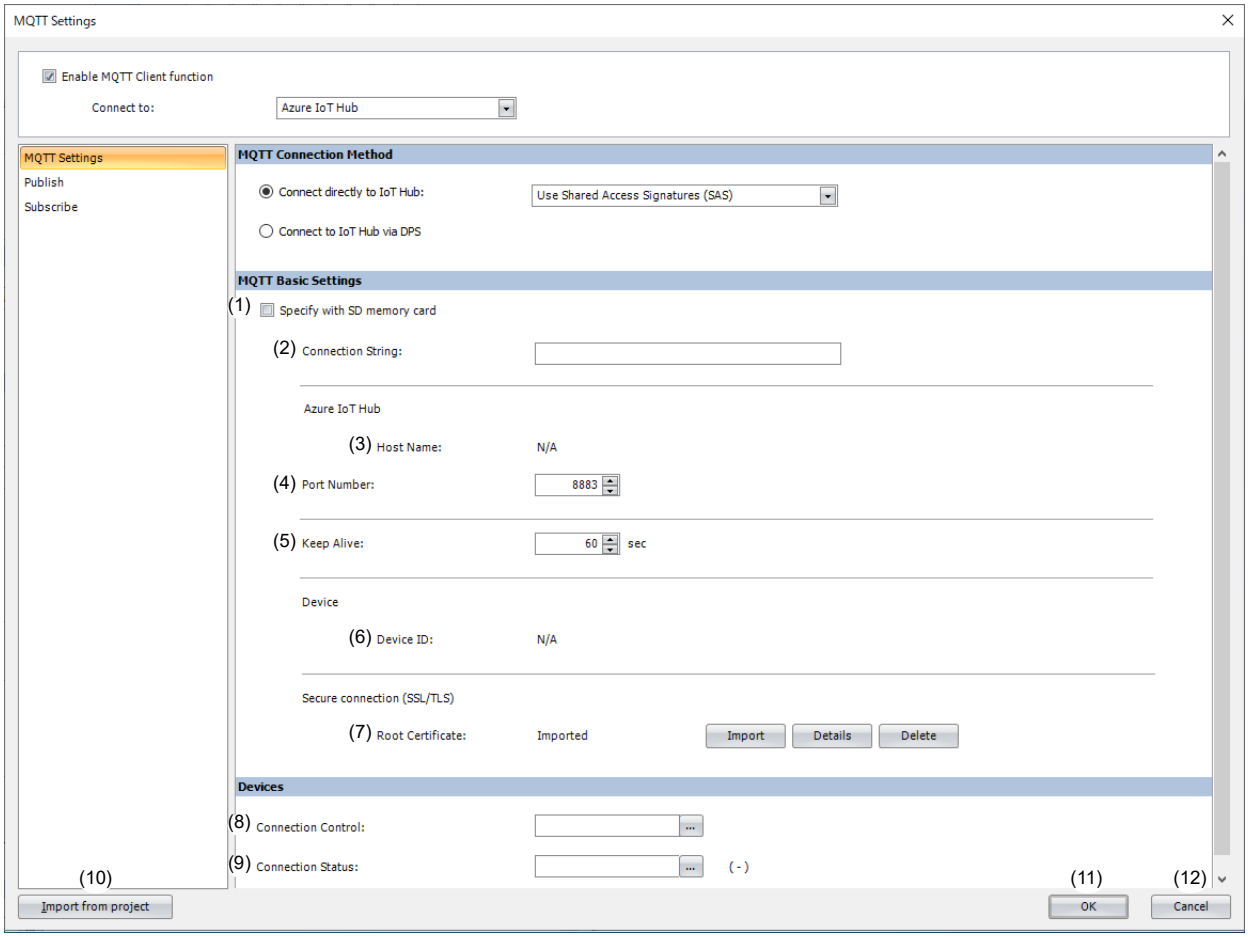
Click **Cancel** button to close the **MQTT Settings** dialog box without saving the settings.

Connect to Azure IoT Hub using SAS

■MQTT Connection Method

In the **MQTT Connection Method** group, select **Connect directly to Azure IoT Hub** radio button and set **Use Shared Access Signature (SAS)**.

■MQTT Basic Settings



(1) Specify with SD memory card/Specify with USB1

Specifies whether to write the basic settings ((2) to (7)) for MQTT communication to the main unit by using external memory. When you select this check box, you can write the basic settings for MQTT communication to the main unit using external memory.

- **Configure SD Memory Card** button
  Configure USB1 button
  : Clicking this button opens the external memory dialog box.

  In the external memory dialog box, configure the basic settings for MQTT communication.
- For examples of how to write parameters when using external memory, see "Restrictions" on page 3-90.

(2) Connection String

Specifies the connection string (primary or secondary connection string) assigned to the main unit in Azure IoT Hub. The connection string includes the Azure IoT Hub host name, device ID and SharedAccessKey. The maximum length of the connection string is 300 characters. Only alphanumeric characters and symbols can be used.
When you use an external memory drive, specify the connection string in "connection_string" key in the [MQTT_AZURE] section of the mqtt_basic_settings.ini file.

(3) Host Name

The value of HostName in the connection string is displayed. For example, when "HostName=abcd.azure-devices.net", then "abcd.azure-devices.net" is displayed.
When you use an external memory drive, the host name does not need to be in the mqtt_basic_settings.ini file.

(4) Port Number

Specifies the port number of the Azure IoT Hub. Set the port number between 0 and 65,535. The default value is 8883 used for Azure IoT Hub. If 0 is specified for the port number, port number 8883 is used.

When you use an external memory drive, specify the port number in "port_number" key in the [MQTT_BROKER] section of the mqtt_basic_settings.ini file.

(5) Keep Alive

Sets the time interval to execute a connection check for the Azure IoT Hub when a connection is made to the Azure IoT Hub. If the set time interval or longer continues without the main unit sending and receiving, a connection check is performed with the Azure IoT Hub. The range that can be set is 5 to 65,535 s. The default value is 60 s.

When you use an external memory drive, specify the setting in "keep_alive" key in the [MQTT_KEEP_ALIVE] section of the mqtt_basic_settings.ini file.

(6) Device ID

The value of DeviceId in the connection string is displayed. For example, when "DeviceId=1234", then "1234" is displayed. The device ID is used as the client ID for MQTT communication.

When you use an external memory drive, the device ID does not need to be in the mqtt_basic_settings.ini file.

(7) Root Certificate

Click **Import** button to import a root certificate to use when performing SSL/TLS communication with the Azure IoT Hub. Files that can be imported are pem or crt files. WindLDR and converts pem and crt files to der files. der files that are 2KB or bigger cannot be imported. When a pem file contains multiple root certificates, WindLDR and WindO/I-NV4 convert each of the first two root certificates to a der file.

Click **Details** button to display the **Certificate Information** dialog box that shows the root certificate of the imported server certificate. Click **Delete** button to delete the imported certificate.

When you use an external memory drive, specify the setting in "root_certificate" key in the [MQTT_TLS] section of the mqtt_basic_settings.ini file. If a root certificate was imported, the "root_certificate" key is true and an mqtt_root_certificate.der file is created. If the imported pem file contains multiple root certificates, the second root certificate from the top is be generated as the mqtt_root_certificate_2.der file. If a root certificate was not imported, the "root_certificate" key is false and an mqtt_root_certificate.der file is not created.

■**Devices**

(8) Connection Control

Sets the device address used for connecting to and disconnecting from the Azure IoT Hub. The bit device that can be set depends on the main unit type.

Internal Device Name		Symbol	FC6A	FT2J/1J	HG2J/1J
Internal relay		M	Yes	No	No
HMI Devices	HMI Internal Relay	LM	No	Yes	Yes
	HMI Keep Relay	LK	No	Yes	Yes
	HMI Temporary Relay	LBM	No	Yes	Yes
Control Devices	Internal relay	M	No	Yes	No

For details, see the following manuals.
FC6A: Chapter 6 "Devices" in the "FC6A Series MICROSmart User's Manual".
FT2J/1J, HG2J/1J: Chapter 35 "1 Bit Devices" in the "WindO/I-NV4 User's Manual".

Starting from the specified device address, 5 bits of device addresses are used. Specify the first device address so that the device range is not exceeded.

Storage Destination	Item	Description
Starting number+0	Connection Control*1*2	When the set device address is turned on, a connection is made to the Azure IoT Hub. When the set device address is turned off, the connection with the Azure IoT Hub is disconnected.
Starting number+1	Reserved	
Starting number+2	Reserved	
Starting number+3	Reserved	
Starting number+4	Reserved	

*1 When generating the SAS token to connect to the Azure IoT Hub, the current time (UTC) calculated from the clock data of the main unit is used. If the clock of the main unit is far off from the actual current time, the main unit may not be able to connect to the Azure IoT Hub. Configure the clock and clock-related functions of the main unit.

When acquiring the current time (UTC) from the SNTP server

Refer to the following configuration items to configure the clock and clock-related functions of the main unit.

FC6A:

Function Name	Overview	Reference
SNTP Settings	Acquires the current time (UTC) from the SNTP server to adjust the clock of the main unit.	Chapter 3 "SNTP Settings" in the FC6A Series MICROSmart Communication Manual
Time Zone Settings	Configures the difference for each time zone in the region relative to standard time.	Chapter 5 "Time Zone" in the FC6A Series MICROSmart User's Manual
Daylight Savings Time Settings	Configures the daylight saving time period.	Chapter 5 "Daylight Savings Time" in the FC6A Series MICROSmart User's Manual

FT2J/1J, HG2J/1J:

Function Name	Overview	Reference
Internal Clock	Configures the SNTP server from which to obtain date and time data, the time zone, and daylight saving time.	Chapter 4 "3.19 Internal Clock Tab" in the WindO/I-NV4 User's Manual

The clock of the main unit loses time as time goes on according to accuracy. If more than 6 months have passed since the last time the current time (UTC) was acquired from the SNTP server, the main unit may not be able to connect to the Azure IoT Hub. Acquire the current time (UTC) from the SNTP server at the appropriate frequency.

When configuring the clock of the main unit directly

Refer to the following configuration items to configure the clock and clock-related functions of the main unit.

FC6A:

Function Name	Overview	Reference
Clock Function	Configures the clock of the main unit using WindLDR or special data registers and special internal relays.	Chapter 5 "Clock Function" in the FC6A Series MICROSmart User's Manual
Time Zone Settings	Configures the difference for each time zone in the region relative to standard time.	Chapter 5 "Time Zone" in the FC6A Series MICROSmart User's Manual
Daylight Savings Time Settings	Configures the daylight saving time period.	Chapter 5 "Daylight Savings Time" in the FC6A Series MICROSmart User's Manual

FT2J/1J, HG2J/1J:

Function Name	Overview	Reference
Clock Setting	Press **Clock Setting (Clock Setting)** in the Main Menu screen to display the clock settings screen. Use this screen to set the internal clock of the main unit.	Chapter 36 "3.2 Clock Setting" in the WindO/I-NV4 User's Manual
HMI Special Data Register	Use the HMI special data registers (LSD20 to 26) to set the internal clock of the main unit.	Chapter 35 "2.1 HMI Device Addresses" in the WindO/I-NV4 User's Manual

The clock of the main unit loses time as time goes on according to accuracy. If more than 6 months have passed since the last time the current time was configured, the main unit may not be able to connect to the Azure IoT Hub. Configure the current time at the appropriate frequency.

*2 The SAS token generated by the main unit expires 5 minutes after it is generated. If the expiration date expires while connected to Azure IoT Hub, an Azure IoT Hub response error will occur (see Connection Error Code in "(9) Connection Status" on page 3-25), and the connection to Azure IoT Hub will be disconnected. If you want to continue connecting, reconnect to Azure IoT Hub.

(9) Connection Status

Sets the data registers that store the status when connecting to the Azure IoT Hub and the error information. The word device that can be specified depends on the main unit type.

Internal Device Name		Symbol	FC6A	FT2J/1J	HG2J/1J
Data Register		D	Yes	No	No
HMI Devices	HMI Data Register	LDR	No	Yes	Yes
	HMI Keep Register	LKR	No	Yes	Yes
	HMI Temporary Register	LBR	No	Yes	Yes
Control Devices	Data Register	D	No	Yes	No

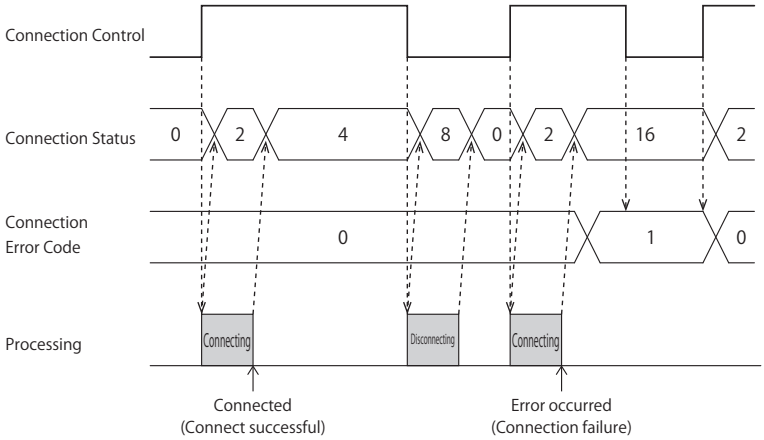
For details, see the following manuals.
FC6A: Chapter 6 "Devices" in the "FC6A Series MICROSmart User's Manual".
FT2J/1J, HG2J/1J: Chapter 35 "2 Word Devices" in the "WindO/I-NV4 User's Manual".

Starting from the specified device address, 5 words of device addresses are used. Specify the first device address so that the device range is not exceeded.

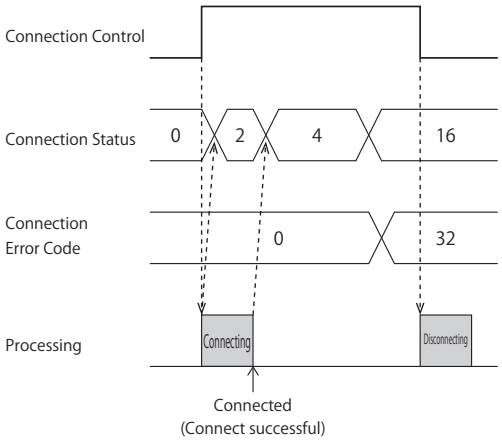
<table>
<tr><th>Storage Destination</th><th>Item</th><th>Description</th></tr>
<tr><td>Starting number+0</td><td>Connection Status</td><td>Stores the status when connecting to the Azure IoT Hub.
<table>
<tr><th>Status Code</th><th>Status</th></tr>
<tr><td>0 (0000h)</td><td>Initial status (disconnected)</td></tr>
<tr><td>2 (0002h)</td><td>Connecting</td></tr>
<tr><td>4 (0004h)</td><td>Connected</td></tr>
<tr><td>8 (0008h)</td><td>Disconnecting</td></tr>
<tr><td>16 (0010h)</td><td>Connection processing error</td></tr>
<tr><td>32 (0020h)</td><td>Disconnection processing error</td></tr>
</table></td></tr>
<tr><td>Starting number+1</td><td>Connection Error Code</td><td>Stores information about the error that occurred when connecting to the Azure IoT Hub.
<table>
<tr><th>Error Code</th><th>Error Details</th></tr>
<tr><td>1 (0001h)</td><td>The Ethernet cable is disconnected or broken and the main unit cannot connect to the network properly</td></tr>
<tr><td>2 (0002h)</td><td>Authentication information was not downloaded from the external memory or reading the downloaded authentication information failed</td></tr>
<tr><td>16 (0010h)</td><td>An unknown packet was received</td></tr>
<tr><td>32 (0020h)</td><td>An invalid MQTT packet was received</td></tr>
<tr><td>64 (0040h)</td><td>Keep alive timeout error</td></tr>
<tr><td>80 (0050h)</td><td>Packet could not arrive at destination host</td></tr>
<tr><td>96 (0060h)</td><td>MQTT packet receive timeout error</td></tr>
<tr><td>112 (0070h)</td><td>TLS error</td></tr>
<tr><td>256 (0100h)</td><td>Azure IoT Hub connection refused (unacceptable MQTT protocol version)</td></tr>
<tr><td>512 (0200h)</td><td>Azure IoT Hub connection refused (invalid device ID)*1</td></tr>
<tr><td>768 (0300h)</td><td>Azure IoT Hub connection refused (Azure IoT Hub unavailable)</td></tr>
<tr><td>1024 (0400h)</td><td>Azure IoT Hub connection refused (invalid account name or password)*1*2</td></tr>
<tr><td>1280 (0500h)</td><td>Azure IoT Hub connection refused (not authorized)</td></tr>
<tr><td>32768 (8000h)</td><td>Azure IoT Hub response error</td></tr>
</table></td></tr>
<tr><td>Starting number+2</td><td colspan="2">Reserved</td></tr>
<tr><td>Starting number+3</td><td colspan="2">Reserved</td></tr>
<tr><td>Starting number+4</td><td colspan="2">Reserved</td></tr>
</table>

*1 Check the value of the DeviceId in the connection string.
*2 Configure the clock and clock-related functions of the main unit.

- The following timing chart shows the operations when the main unit connects to Azure IoT Hub.



- Connection Error Code 16 (0010h) to 112 (0070h) may occur even when the main unit is connected to the Azure IoT Hub (Connection Status 4 (0004h)).



- MQTT communication is not performed while the ladder program is stopped. When switching from run to stop when connected to Azure IoT Hub, the Connection Control relay is turned off and disconnected 0 (0000h) is stored in the Connection Status.

(10) Import from project button

Imports only the MQTT settings from a project file.

(11) OK button

Click **OK** button to save the settings and close the **MQTT Settings** dialog box.

(12) Cancel button

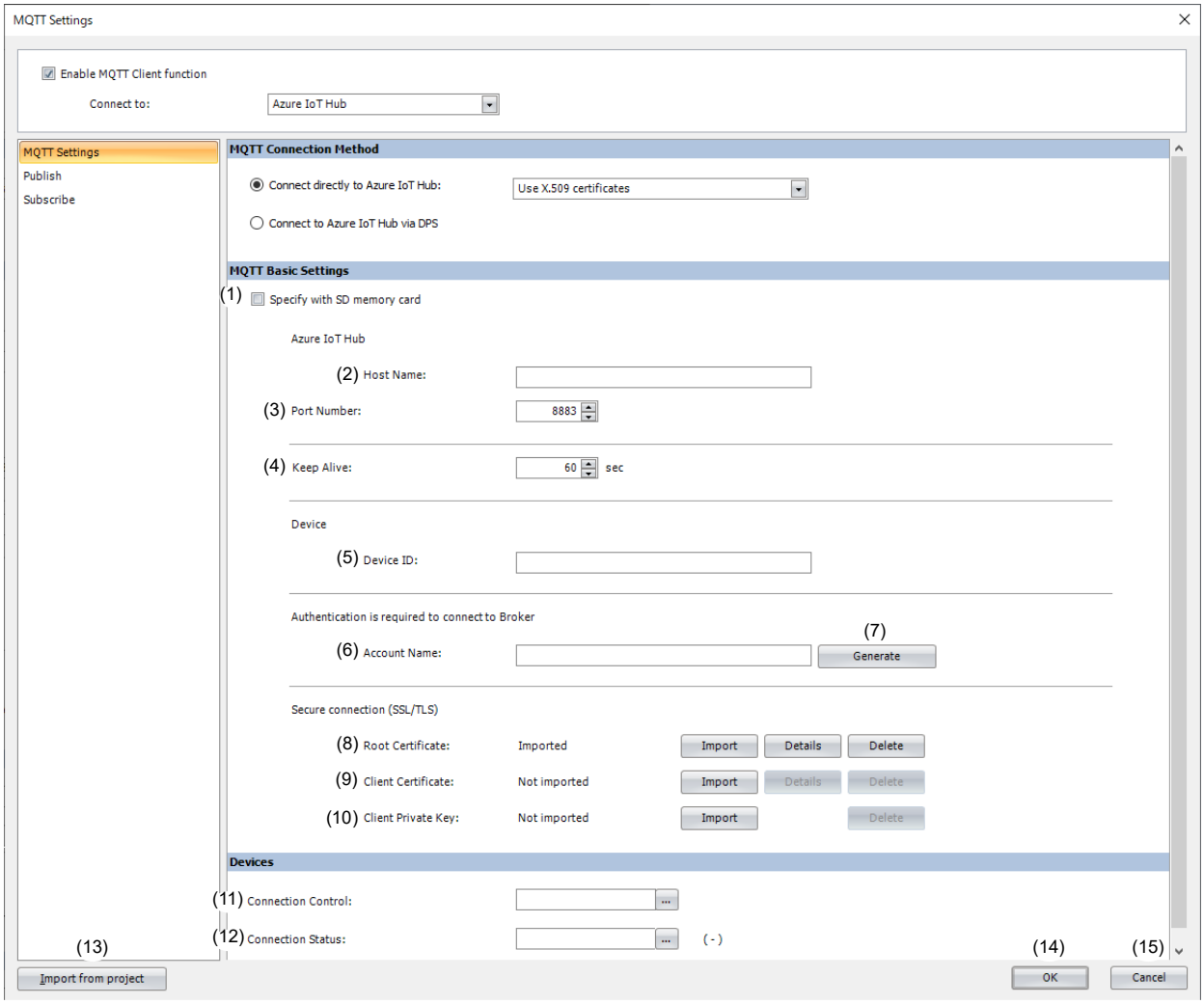
Click **Cancel** button to close the **MQTT Settings** dialog box without saving the settings.

Connect to Azure IoT Hub using X.509 certificates

■MQTT Connection Method

In the **MQTT Connection Method** group, select **Connect directly to Azure IoT Hub** radio button and set **Use X.509 certificates**.

■MQTT Basic Settings



(1) Specify with SD memory card/Specify with USB1

Specifies whether to write the basic settings ((2) to (10)) for MQTT communication to the main unit by using external memory. When you select this check box, you can write the basic settings for MQTT communication to the main unit using external memory.

- **Configure SD Memory Card** button
  Configure USB1 button : Clicking this button opens the external memory dialog box.

  In the external memory dialog box, configure the basic settings for MQTT communication.
- For examples of how to write parameters when using external memory, see "Restrictions" on page 3-90.

(2) Host Name

Sets the host name of the Azure IoT Hub. The maximum length of the host name is 68 characters. Only alphanumeric characters and the symbol (-) can be used.
When you use an external memory drive, specify the host name in "hostname" key in the [MQTT_BROKER] section of the mqtt_basic_settings.ini file.

(3) Port Number

Sets the port number of the Azure IoT Hub. Set the port number between 0 and 65,535. The default value is 8883 used for Azure IoT Hub. If 0 is specified for the port number, port number 8883 is used.
When you use an external memory drive, specify the port number in "port_number" key in the [MQTT_BROKER] section of the mqtt_basic_settings.ini file.

(4) Keep Alive

Sets the time interval to execute a connection check for the Azure IoT Hub when a connection is made to the Azure IoT Hub. If the set time interval or longer continues without the main unit sending and receiving, a connection check is performed with the Azure IoT Hub. The range that can be set is 5 to 65,535 s. The default value is 60 s.
When you use an external memory drive, specify the setting in "keep_alive" key in the [MQTT_KEEP_ALIVE] section of the mqtt_basic_settings.ini file.

(5) Device ID

Sets the device ID of main unit registered to Azure IoT Hub. The maximum length of the device ID is 128 characters. Only alphanumeric characters and symbols can be used. The device ID is used as the client ID for MQTT communication.
When you use an external memory drive, specify the client ID in "client_id_type" and "client_id" keys in the [MQTT_CLIENT_ID] section of the mqtt_basic_settings.ini file. Set "string" for the "client_id_type" key.

(6) Account Name

Sets the account name used to connect to Azure IoT Hub. The maximum length of the account name is 128 characters. Only alphanumeric characters and symbols can be used.
When you use an external memory drive, specify the account name in "accountname" key in the [MQTT_AUTH] section of the mqtt_basic_settings.ini file.

(7) Generate

Click this button to generate an account name used to connect to the Azure IoT Hub based on the host name (2) and device ID (5).

(8) Root Certificate

Click **Import** button to import a root certificate to use when performing SSL/TLS communication with the Azure IoT Hub. Files that can be imported are pem or crt files. WindLDR and WindO/I-NV4 convert pem and crt files to der files. der files that are 2KB or bigger cannot be imported. When a pem file contains multiple root certificates, WindLDR and WindO/I-NV4 convert each of the first two root certificates to a der file. Click **Details** button to display the **Certificate Information** dialog box that shows the root certificate of the imported server certificate. Click **Delete** button to delete the imported certificate.
When you use an external memory drive, specify the setting in "root_certificate" key in the [MQTT_TLS] section of the mqtt_basic_settings.ini file. If a root certificate was imported, the root_certificate key is true and an mqtt_root_certificate.der file is created. If the imported pem file contains multiple root certificates, the second root certificate from the top is be generated as the mqtt_root_certificate_2.der file. If a root certificate was not imported, the root_certificate key is false and an mqtt_root_certificate.der file is not created.

(9) Client Certificate

Click **Import** button to import a client certificate to use when performing SSL/TLS communication with the Azure IoT Hub. Files that can be imported are pem or crt files. WindLDR and WindO/I-NV4 convert pem and crt files into der files. der files that are 2KB or bigger cannot be imported. Click **Details** button to display the **Certificate Information** dialog box that shows the client certificate of the imported server certificate. Click **Delete** button to delete the imported certificate.
When you use an external memory drive, specify the setting in "client_certificate" key in the [MQTT_TLS] section of the mqtt_basic_settings.ini file. If a client certificate was imported, the client_certificate key is true and an mqtt_client_certificate.der file is created. If a client certificate was not imported, the client_certificate key is false and an mqtt_client_certificate.der file is not created.

(10) Client Private Key

Click **Import** button to import a client private key (RSA encryption, key length 3,072 bits max.) to use when performing SSL/TLS communication with the Azure IoT Hub. WindLDR and WindO/I-NV4 support the following file formats.

- PKCS#1 format pem file or der file
- PKCS#8 format (unencrypted) pem file or der file

Click **Delete** button to delete the imported client private key.
When you use an external memory drive, specify the setting in "client_private_key"key in the [MQTT_TLS] section of the mqtt_basic_settings.ini file. If a client private key was imported, the client_private_key key is true and an mqtt_private_key.der file is created. If a client private key was not imported, the client_private_key key is false and an mqtt_private_key.der file is not created.

■**Devices**

(11) Connection Control

Sets the internal relay used for connecting to and disconnecting from the Azure IoT Hub. The bit device that can be set depends on the main unit type.

Internal Device Name		Symbol	FC6A	FT2J/1J	HG2J/1J
Internal relay		M	Yes	No	No
HMI Devices	HMI Internal Relay	LM	No	Yes	Yes
	HMI Keep Relay	LK	No	Yes	Yes
	HMI Temporary Relay	LBM	No	Yes	Yes
Control Devices	Internal relay	M	No	Yes	No

For details, see the following manuals.
FC6A: Chapter 6 "Devices" in the "FC6A Series MICROSmart User's Manual".
FT2J/1J, HG2J/1J: Chapter 35 "1 Bit Devices" in the "WindO/I-NV4 User's Manual".

Starting from the specified device address, 5 bits of device addresses are used. Specify the first device address so that the device range is not exceeded.

Storage Destination	Item	Description
Starting number+0	Connection Control	Connects to and disconnects from the Azure IoT Hub. When the set device address is turned on, a connection is made to the Azure IoT Hub. When the set device address is turned off, the connection with the Azure IoT Hub is disconnected.
Starting number+1	Reserved	
Starting number+2	Reserved	
Starting number+3	Reserved	
Starting number+4	Reserved	

(12) Connection Status

Sets the data registers that store the status when connecting to the Azure IoT Hub and the error information. The word device that can be specified depends on the main unit type.

Internal Device Name		Symbol	FC6A	FT2J/1J	HG2J/1J
Data Register		D	Yes	No	No
HMI Devices	HMI Data Register	LDR	No	Yes	Yes
	HMI Keep Register	LKR	No	Yes	Yes
	HMI Temporary Register	LBR	No	Yes	Yes
Control Devices	Data Register	D	No	Yes	No

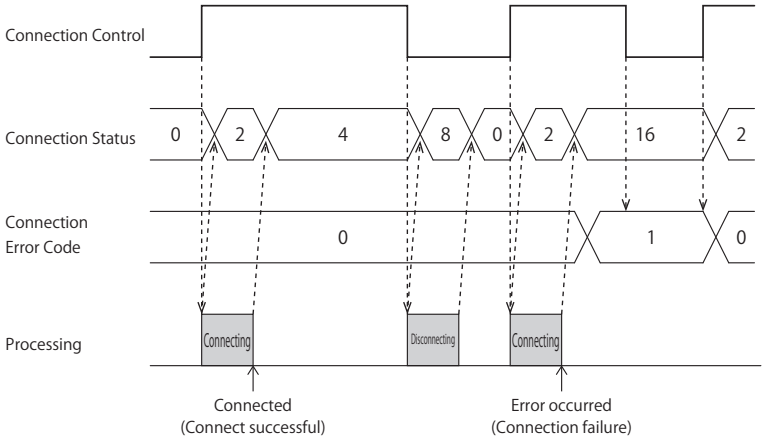
For details, see the following manuals.
FC6A: Chapter 6 "Devices" in the "FC6A Series MICROSmart User's Manual".
FT2J/1J, HG2J/1J: Chapter 35 "2 Word Devices" in the "WindO/I-NV4 User's Manual".

Starting from the specified device address, 5 words of device addresses are used. Specify the first device address so that the device range is not exceeded.

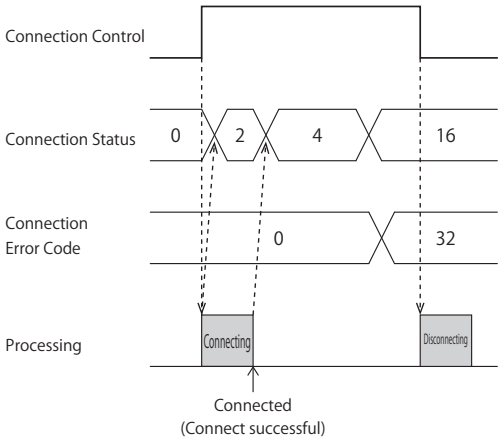
<table>
<tr><th>Storage Destination</th><th>Item</th><th>Description</th></tr>
<tr><td>Starting number+0</td><td>Connection Status</td><td>Stores the status when connecting to the Azure IoT Hub.
<table>
<tr><th>Status Code</th><th>Status</th></tr>
<tr><td>0 (0000h)</td><td>Initial status (disconnected)</td></tr>
<tr><td>2 (0002h)</td><td>Connecting</td></tr>
<tr><td>4 (0004h)</td><td>Connected</td></tr>
<tr><td>8 (0008h)</td><td>Disconnecting</td></tr>
<tr><td>16 (0010h)</td><td>Connection processing error</td></tr>
<tr><td>32 (0020h)</td><td>Disconnection processing error</td></tr>
</table></td></tr>
</table>

<table>
<tr><th>Storage Destination</th><th>Item</th><th>Description</th></tr>
<tr><td>Starting number+1</td><td>Connection Error Code</td><td>Stores information about the error that occurred when connecting to the Azure IoT Hub.
<table>
<tr><th>Error Code</th><th>Error Details</th></tr>
<tr><td>1 (0001h)</td><td>The Ethernet cable is disconnected or broken and the main unit cannot connect to the network properly</td></tr>
<tr><td>2 (0002h)</td><td>Authentication information was not downloaded from the external memory or reading the downloaded authentication information failed</td></tr>
<tr><td>4 (0004h)</td><td>Invalid device ID format</td></tr>
<tr><td>16 (0010h)</td><td>An unknown packet was received</td></tr>
<tr><td>32 (0020h)</td><td>An invalid MQTT packet was received</td></tr>
<tr><td>64 (0040h)</td><td>Keep alive timeout error</td></tr>
<tr><td>80 (0050h)</td><td>Packet could not arrive at destination host</td></tr>
<tr><td>96 (0060h)</td><td>MQTT packet receive timeout error</td></tr>
<tr><td>112 (0070h)</td><td>TLS error</td></tr>
<tr><td>256 (0100h)</td><td>Azure IoT Hub connection refused (unacceptable MQTT protocol version)</td></tr>
<tr><td>512 (0200h)</td><td>Azure IoT Hub connection refused (invalid device ID)</td></tr>
<tr><td>768 (0300h)</td><td>Azure IoT Hub connection refused (Azure IoT Hub unavailable)</td></tr>
<tr><td>1024 (0400h)</td><td>Azure IoT Hub connection refused (invalid account name or password)</td></tr>
<tr><td>1280 (0500h)</td><td>Azure IoT Hub connection refused (not authorized)</td></tr>
<tr><td>32768 (8000h)</td><td>Azure IoT Hub response error</td></tr>
</table></td></tr>
<tr><td>Starting number+2</td><td colspan="2">Reserved</td></tr>
<tr><td>Starting number+3</td><td colspan="2">Reserved</td></tr>
<tr><td>Starting number+4</td><td colspan="2">Reserved</td></tr>
</table>

- The following timing chart shows the operations when the main unit connects to Azure IoT Hub.



- Connection Error Code 16 (0010h) to 112 (0070h) may occur even when the main unit is connected to the Azure IoT Hub (Connection Status 4 (0004h))



- MQTT communication is not performed while the ladder program is stopped. When switching from run to stop when connected to Azure IoT Hub, the Connection Control relay is turned off and disconnected 0 (0000h) is stored in the Connection Status.

(13) Import from project button

Imports only the MQTT settings from a project file.

(14) OK button

Click **OK** button to save the settings and close the **MQTT Settings** dialog box.

(15) Cancel button

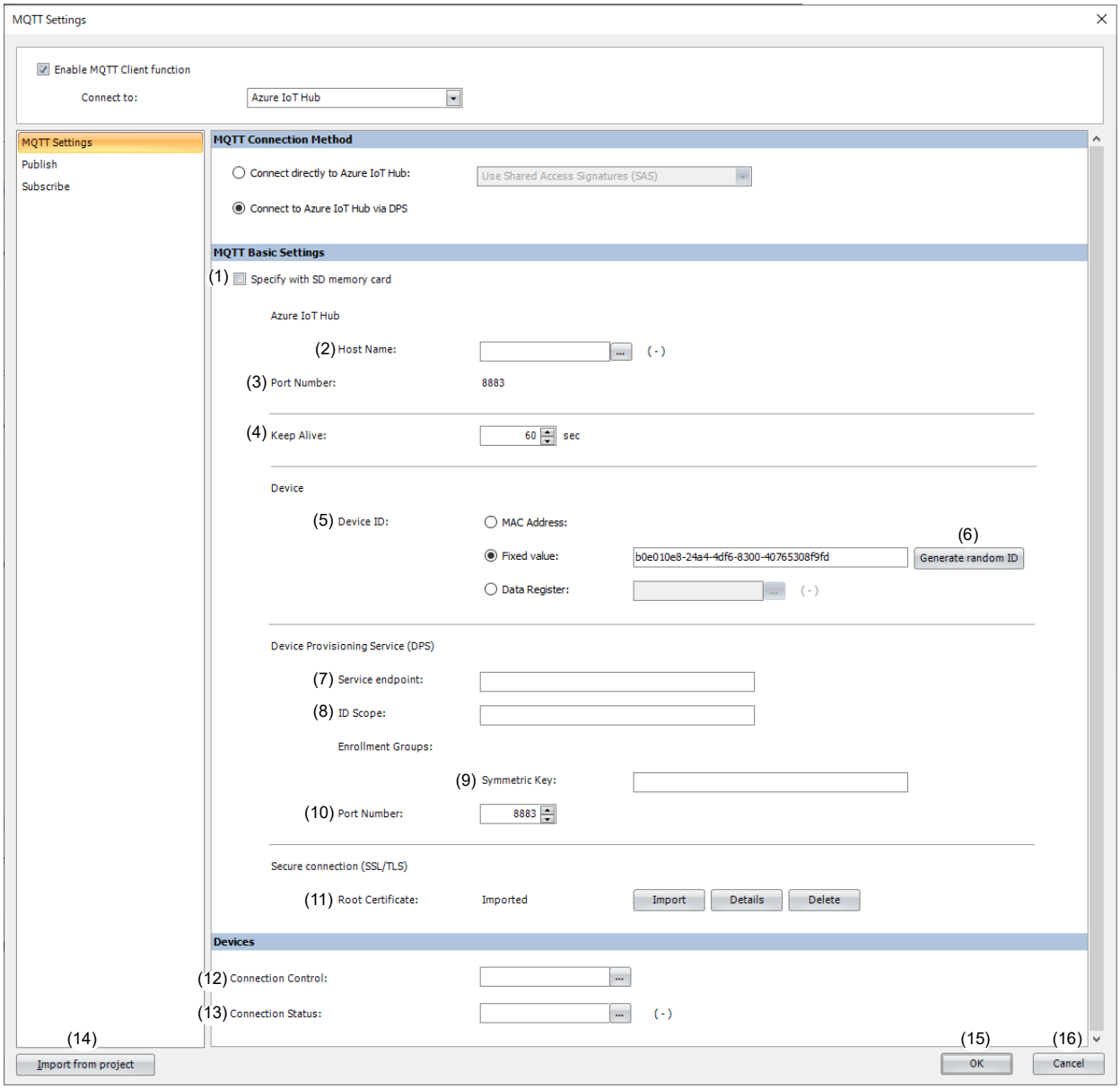
Click **Cancel** button to close the **MQTT Settings** dialog box without saving the settings.

Connect to Azure IoT Hub via DPS

■MQTT Connection Method

In the **MQTT Connection Method** group, select **Connect to Azure IoT Hub via DPS**.

■MQTT Basic Settings



(1) Specify with SD memory card/Specify with USB1

Specifies whether to write the basic settings ((2) to (11)) for MQTT communication to the main unit by using external memory. When you select this check box, you can write the basic settings for MQTT communication to the main unit using external memory.

- **Configure SD Memory Card** button
  Configure USB1 button
  : Clicking this button opens the external memory dialog box.

  In the external memory dialog box, configure the basic settings for MQTT communication.
- For examples of how to write parameters when using external memory, see "Restrictions" on page 3-90.

(2) Host Name

Sets the device address to store the host name (ASCII code) of the Azure IoT Hub got from the DPS. The word device that can be specified depends on the main unit type.

Internal Device Name		Symbol	FC6A	FT2J/1J	HG2J/1J
Data Register		D	Yes	No	No
HMI Devices	HMI Data Register	LDR	No	Yes	Yes
	HMI Keep Register	LKR	No	Yes	Yes
	HMI Temporary Register	LBR	No	Yes	Yes
Control Devices	Data Register	D	No	Yes	No

For details, see the following manuals.
FC6A: Chapter 6 "Devices" in the "FC6A Series MICROSmart User's Manual".
FT2J/1J, HG2J/1J: Chapter 35 "2 Word Devices" in the "WindO/I-NV4 User's Manual".

Starting from the specified device address, 64 words of device addresses are used. Specify the first device address so that the device range is not exceeded.
When you use an external memory drive, specify the host name in "hostname" key in the [MQTT_BROKER] section of the mqtt_basic_settings.ini file.

(3) Port Number

Sets the port number of the Azure IoT Hub. The same port number as the Device Provisioning Service (DPS) port number (10) is used.
When you use an external memory drive, the port number does not need to be in the mqtt_basic_settings.ini file.

(4) Keep Alive

Sets the time interval to execute a connection check for the Azure IoT Hub when a connection is made to the Azure IoT Hub. If the set time interval or longer continues without the main unit sending and receiving, a connection check is performed with the Azure IoT Hub. The range that can be set is 5 to 65,535 s. The default value is 60 s.
When you use an external memory drive, specify the setting in "keep_alive" key in the [MQTT_KEEP_ALIVE] section of the mqtt_basic_settings.ini file.

(5) Device ID

Sets the device ID to register to Azure IoT Hub via DPS. The device ID can be set from the following three types and is also used as a client ID for MQTT communication.

Device ID	Description
MAC address	Sets the MAC address of the main unit interface.*1
Fixed value	Sets any character string. The maximum length of the device ID is 128 characters. Only alphanumeric characters and symbols (ASCII data character code 0x20 to 0x7E) can be used.
Data Resister	Sets the device addresses*2 for storing the device ID. The value is read in order from the upper byte of the specified device address, treated as character data, and used as the device ID.*3 Starting from the specified device address, 64 continuous words of device addresses are used.*4 Specify the first device address so that the device range is not exceeded. Only alphanumeric characters and symbols (ASCII data character code 0x20 to 0x7E) can be used.

*1 For example, when the MAC address of Ethernet port 1 of the Plus CPU module is 12-34-56-78-9A-BC (D8324=0012h, D8325=0034h, D8326=0056h, D8327=0078h, D8328=009Ah, D8329=00BCh), the device ID is "123456789ABC".

*2 The word device that can be specified depends on the main unit type.

Internal Device Name		Symbol	FC6A	FT2J/1J	HG2J/1J
Data Register		D	Yes	No	No
HMI Devices	HMI Data Register	LDR	No	Yes	Yes
	HMI Keep Register	LKR	No	Yes	Yes
	HMI Temporary Register	LBR	No	Yes	Yes
Control Devices	Data Register	D	No	Yes	No

For details, see the following manuals.
FC6A: Chapter 6 "Devices" in the "FC6A Series MICROSmart User's Manual".
FT2J/1J, HG2J/1J: Chapter 35 "2 Word Devices" in the "WindO/I-NV4 User's Manual".

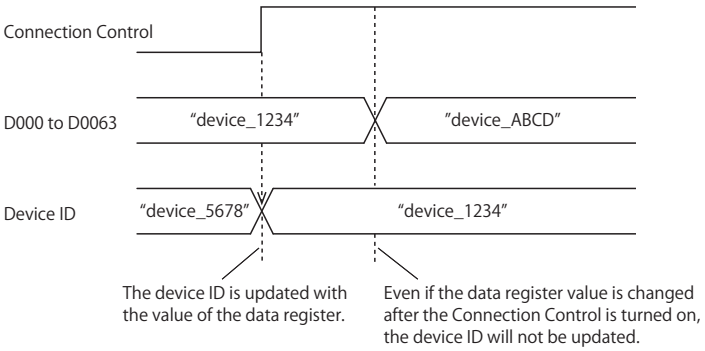
*3 For example, when D0000 is specified and the following values are stored in each data register, the device ID is "device_1234".

Data resister	Stored Value	
	Upper Byte	Lower Byte
D0000	"d"=64h	"e"=65h
D0001	"v"=76h	"i"=69h
D0002	"c"=63h	"e"=65h
D0003	"_"=5Fh	"1"=31h
D0004	"2"=32h	"3"=33h
D0005	"4"=34h	00h

*4 When the character string to be set is shorter than 128 bytes (64 words), add the terminating character NULL (00h) as the end of the character string. The data from the upper byte of the specified device address up to the data right before the terminating character NULL (00h) is treated as character data and used as the device ID.

When you use an external memory drive, specify the client ID in the [MQTT_CLIENT_ID] section of the mqtt_basic_settings.ini file.

- When the device ID is changed, connect to the DPS again and register the main unit.
- When setting the device ID with the data register, the MOVC instruction can be used to store a character string in the data registers. For details on MOVC instruction, see Chapter 5 "MOVC (Move Characters)" in the Ladder Programming Manual.
- When the Connection Control (12) is changed from off to on, the device ID stored inside the main unit is updated. For example, when the device ID is set in the data register (D0000), the device ID is updated at the following timing.



(6) Generate random ID

When the **Device ID** (5) is set as a fixed value, click this button to generate a random identifier for device ID.

(7) Service endpoint

Sets the service endpoint for the DPS. The maximum length of the device ID is 81 characters. Only alphanumeric characters and symbols can be used.

When you use an external memory drive, specify the service endpoint in "dps_endpoint" key in the [MQTT_AZURE] section of the mqtt_basic_settings.ini file.

(8) ID Scope

Sets the ID scope of the DPS. The maximum length of the device ID is 11 characters. Only alphanumeric characters can be used.

When you use an external memory drive, specify the ID scope in "dps_id_scope" key in the [MQTT_AZURE] section of the mqtt_basic_settings.ini file.

(9) Symmetric key

Sets the symmetric key (primary or secondary key) of the registration group of the DPS to which the main unit is registered. The maximum length of the symmetric key is 88 characters. Only alphanumeric characters and symbols can be used.

When you use an external memory drive, specify the symmetric key in "dps_enrollment_group_symmetric_key" key in the [MQTT_AZURE] section of the mqtt_basic_settings.ini file.

(10) Port Number

Sets the port number for the Device Provisioning Service (DPS). Set the port number between 0 and 65,535. The default value is 8883 used for DPS. If 0 is specified for the port number, port number 8883 is used.

When you use an external memory drive, specify the port number in "dps_port_number" key in the [MQTT_AZURE] section of the mqtt_basic_settings.ini file.

(11) Root Certificate

Click **Import** button to import a root certificate to use when performing SSL/TLS communication with the DPS and the Azure IoT Hub. Files that can be imported are pem or crt files. WindLDR and WindO/I-NV4 convert pem and crt files to der files. der files that are 2KB or bigger cannot be imported. When a pem file contains multiple root certificates, WindLDR and WindO/I-NV4 convert each of the first two root certificates to a der file. Click **Details** button to display the **Certificate Information** dialog box that shows the root certificate of the imported server certificate. Click **Delete** button to delete the imported certificate.

When you use an external memory drive, specify the setting in "root_certificate" key in the [MQTT_TLS] section of the mqtt_basic_settings.ini file. If a root certificate was imported, the root_certificate key is true and an mqtt_root_certificate.der file is created. If the imported pem file contains multiple root certificates, the second root certificate from the top is be generated as the mqtt_root_certificate_2.der file. If a root certificate was not imported, the root_certificate key is false and an mqtt_root_certificate.der file is not created.

■**Devices**

(12) Connection Control

Sets the device address used for connecting to and disconnecting from the DPS and Azure IoT Hub. The bit device that can be set depends on the main unit type.

Internal Device Name		Symbol	FC6A	FT2J/1J	HG2J/1J
Internal relay		M	Yes	No	No
HMI Devices	HMI Internal Relay	LM	No	Yes	Yes
	HMI Keep Relay	LK	No	Yes	Yes
	HMI Temporary Relay	LBM	No	Yes	Yes
Control Devices	Internal relay	M	No	Yes	No

For details, see the following manuals.
FC6A: Chapter 6 "Devices" in the "FC6A Series MICROSmart User's Manual".
FT2J/1J, HG2J/1J: Chapter 35 "1 Bit Devices" in the "WindO/I-NV4 User's Manual".

Starting from the specified device address, 5 bits of device addresses are used. Specify the first device address so that the device range is not exceeded.

Storage Destination	Item	Description
Starting number+0	Connection Control *1*2	Connects to and disconnects from the DPS and Azure IoT Hub. When Connection Control is turned on, the main unit connects to the DPS and Azure IoT Hub depending on the status of the DPS connection bit (Starting number+1). When Connection Control is turned off, the connection with the DPS and Azure IoT Hub is disconnected.
Starting number+1	DPS connection bit	Allows connection to the DPS. While DPS connection bit is on, when Connection Control (Starting number+0) is turned on, the main unit connects to the Azure IoT Hub via DPS.*3 While DPS connection bit is off, when Connection Control (Starting number+0) is turned on, the main unit connects directly to the Azure IoT Hub with the host name stored in the **Host Name** (2) without going through DPS.*4
Starting number+2	Reserved	
Starting number+3	Reserved	
Starting number+4	Reserved	

*1 When generating the SAS token to connect to the Azure IoT Hub, the current time (UTC) calculated from the clock data of the main unit is used. If the clock of the main unit is far off from the actual current time, the main unit may not be able to connect to the Azure IoT Hub. Configure the clock and clock-related functions of the main unit.

When acquiring the current time (UTC) from the SNTP server

Refer to the following configuration items to configure the clock and clock-related functions of the main unit.
FC6A:

Function Name	Overview	Reference
SNTP Settings	Acquires the current time (UTC) from the SNTP server to adjust the clock of the main unit.	Chapter 3 "SNTP Settings" in the FC6A Series MICROSmart Communication Manual
Time Zone Settings	Configures the difference for each time zone in the region relative to standard time.	Chapter 5 "Time Zone" in the FC6A Series MICROSmart User's Manual
Daylight Savings Time Settings	Configures the daylight saving time period.	Chapter 5 "Daylight Savings Time" in the FC6A Series MICROSmart User's Manual

FT2J/1J, HG2J/1J:

Function Name	Overview	Reference
Internal Clock	Configures the SNTP server from which to obtain date and time data, the time zone, and daylight saving time.	Chapter 4 "3.19 Internal Clock Tab" in the WindO/I-NV4 User's Manual

The clock of the main unit loses time as time goes on according to accuracy. If more than 6 months have passed since the last time the current time (UTC) was acquired from the SNTP server, the main unit may not be able to connect to the DPS and the Azure IoT Hub. Acquire the current time (UTC) from the SNTP server at the appropriate frequency.

When configuring the clock of the main unit directly

Refer to the following configuration items to configure the clock and clock-related functions of the main unit.

FC6A:

Function Name	Overview	Reference
Clock Function	Configures the clock of the main unit using WindLDR or special data registers and special internal relays.	Chapter 5 "Clock Function" in the FC6A Series MICROSmart User's Manual
Time Zone Settings	Configures the difference for each time zone in the region relative to standard time.	Chapter 5 "Time Zone" in the FC6A Series MICROSmart User's Manual
Daylight Savings Time Settings	Configures the daylight saving time period.	Chapter 5 "Daylight Savings Time" in the FC6A Series MICROSmart User's Manual

FT2J/1J, HG2J/1J:

Function Name	Overview	Reference
Clock Setting	Press **Clock Setting (Clock Setting)** in the Main Menu screen to display the clock settings screen. Use this screen to set the internal clock of the main unit.	Chapter 36 "3.2 Clock Setting" in the WindO/I-NV4 User's Manual
HMI Special Data Register	Use the HMI special data registers (LSD20 to 26) to set the internal clock of the main unit.	Chapter 35 "2.1 HMI Device Addresses" in the WindO/I-NV4 User's Manual

The clock of the main unit loses time as time goes on according to accuracy. If more than 6 months have passed since the last time the current time was configured, the main unit may not be able to connect to the DPS and Azure IoT Hub. Configure the current time at the appropriate frequency.

*2 The SAS token generated by the main unit expires 5 minutes after it is generated. If the expiration date expires while connected to Azure IoT Hub, an Azure IoT Hub response error will occur (see Connection Error Code in "(9) Connection Status" on page 3-25), and the connection to Azure IoT Hub will be disconnected. If you want to continue connecting, reconnect to Azure IoT Hub.

*3 At this time, the host name of the Azure IoT Hub got from the DPS is stored in the device address set in the **Host Name** (3).

*4 When the device ID is changed, connect to the DPS again to register the main unit.

(13) Connection Status

Sets the data registers that store the status when connecting to the DPS and Azure IoT Hub and the error information. The word device that can be specified depends on the main unit type.

Internal Device Name		Symbol	FC6A	FT2J/1J	HG2J/1J
Data Register		D	Yes	No	No
HMI Devices	HMI Data Register	LDR	No	Yes	Yes
	HMI Keep Register	LKR	No	Yes	Yes
	HMI Temporary Register	LBR	No	Yes	Yes
Control Devices	Data Register	D	No	Yes	No

For details, see the following manuals.

FC6A: Chapter 6 "Devices" in the "FC6A Series MICROSmart User's Manual".

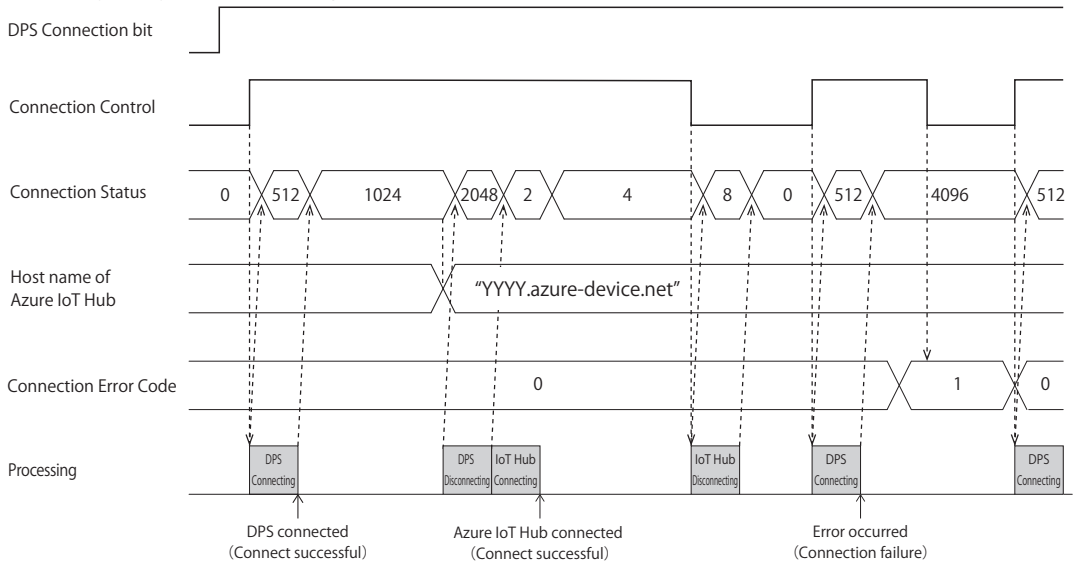
FT2J/1J, HG2J/1J: Chapter 35 "2 Word Devices" in the "WindO/I-NV4 User's Manual".

Starting from the specified device address, 5 words of device addresses are used. Specify the first device address so that the device range is not exceeded.

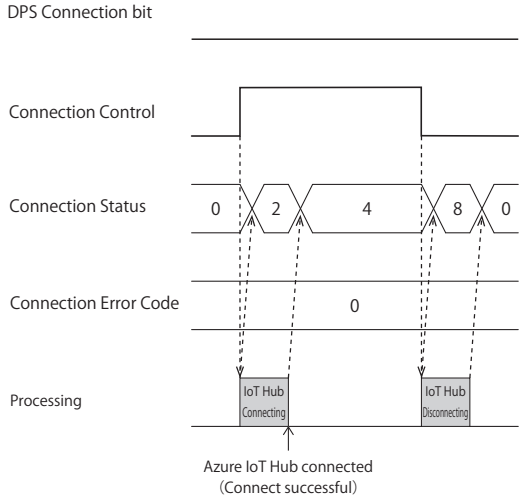
<table>
<tr><th>Storage Destination</th><th>Item</th><th>Description</th></tr>
<tr><td>Starting number+0</td><td>Connection Status</td><td>Stores the status when connecting to the DPS and Azure IoT Hub.
<table>
<tr><th>Status Code</th><th>Status</th></tr>
<tr><td>0 (0000h)</td><td>Initial status (disconnected)</td></tr>
<tr><td>2 (0002h)</td><td>Azure IoT Hub connecting</td></tr>
<tr><td>4 (0004h)</td><td>Azure IoT Hub connected</td></tr>
<tr><td>8 (0008h)</td><td>Azure IoT Hub disconnecting</td></tr>
<tr><td>16 (0010h)</td><td>Azure IoT Hub connection processing error</td></tr>
<tr><td>32 (0020h)</td><td>Azure IoT Hub disconnection processing error</td></tr>
<tr><td>512 (0200h)</td><td>DPS connecting</td></tr>
<tr><td>1024 (0400h)</td><td>DPS connected</td></tr>
<tr><td>2048 (0800h)</td><td>DPS disconnecting</td></tr>
<tr><td>4096 (1000h)</td><td>DPS connection processing error</td></tr>
<tr><td>8192 (2000h)</td><td>DPS disconnection processing error</td></tr>
</table></td></tr>
<tr><td>Starting number+1</td><td>Connection Error Code</td><td>Stores information about the error that occurred when connecting to the DPS and Azure IoT Hub.
<table>
<tr><th>Error Code</th><th>Error Details</th></tr>
<tr><td>1 (0001h)</td><td>The Ethernet cable is disconnected or broken and the main unit cannot connect to the network properly</td></tr>
<tr><td>2 (0002h)</td><td>Authentication information was not downloaded from the external memory or reading the downloaded authentication information failed</td></tr>
<tr><td>4 (0004h)</td><td>Invalid device ID format</td></tr>
<tr><td>8 (0008h)</td><td>The host name of the Azure IoT Hub stored in the device address is incorrect.</td></tr>
<tr><td>16 (0010h)</td><td>An unknown packet was received</td></tr>
<tr><td>32 (0020h)</td><td>An invalid MQTT packet was received</td></tr>
<tr><td>64 (0040h)</td><td>Keep alive timeout error</td></tr>
<tr><td>80 (0050h)</td><td>Packet could not arrive at destination host*1</td></tr>
<tr><td>96 (0060h)</td><td>MQTT packet receive timeout error</td></tr>
<tr><td>112 (0070h)</td><td>TLS error</td></tr>
<tr><td>256 (0100h)</td><td>Azure IoT Hub connection refused (unacceptable MQTT protocol version)</td></tr>
<tr><td>512 (0200h)</td><td>Azure IoT Hub connection refused (invalid device ID)</td></tr>
<tr><td>768 (0300h)</td><td>Azure IoT Hub connection refused (broker unavailable)</td></tr>
<tr><td>1024 (0400h)</td><td>Azure IoT Hub connection refused (invalid account name or password)</td></tr>
<tr><td>1280 (0500h)</td><td>Azure IoT Hub connection refused (not authorized)*2*3*4</td></tr>
<tr><td>4352 (1100h)</td><td>DPS connection refused (unacceptable MQTT protocol version)</td></tr>
<tr><td>4608 (1200h)</td><td>DPS connection refused (invalid device ID)</td></tr>
<tr><td>4864 (1300h)</td><td>DPS connection refused (DPS unavailable)</td></tr>
<tr><td>5120 (1400h)</td><td>DPS connection refused (invalid account name or password)</td></tr>
<tr><td>5376 (1500h)</td><td>DPS connection refused (not authorized)*2*3*4</td></tr>
<tr><td>32768 (8000h)</td><td>Azure IoT Hub / DPS response error</td></tr>
</table></td></tr>
<tr><td>Starting number+2</td><td colspan="2">Reserved</td></tr>
<tr><td>Starting number+3</td><td colspan="2">Reserved</td></tr>
<tr><td>Starting number+4</td><td colspan="2">Reserved</td></tr>
</table>

*1 Check the host name of the DPS service endpoint or Azure IoT Hub.
*2 Check the **Service endpoint** (7), **ID Scope** (8) and **Symmetric Key** (9).
*3 Configure the clock and clock-related functions of the main unit.
*4 Check if the main unit is registered in the DPS and Azure IoT Hub. Reconnect to the DPS if necessary.

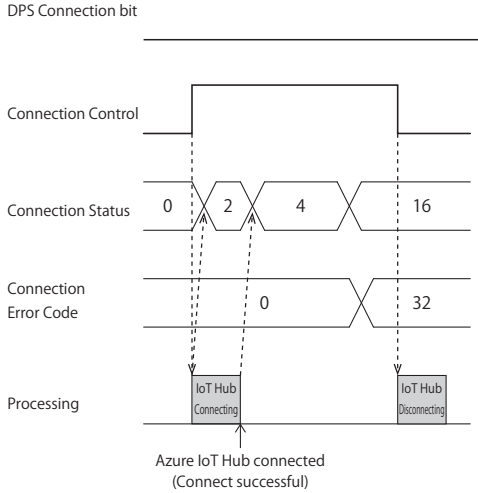
- The following timing chart shows the operations when the main unit connects to DPS and Azure IoT Hub.



- The following timing chart shows the operations when the main unit connects directly to the Azure IoT Hub when the DPS Connection bit is off.



- Connection Error Code 16 (0010h) to 112 (0070h) may occur even when the main unit is connected to the Azure IoT Hub (Connection Status 4 (0004h)).



- MQTT communication is not performed while stopped. When switching from run to stop when connected to Azure IoT Hub, the Connection Control relay is turned off and disconnected 0 (0000h) is stored in the Connection Status.

(14) Import from project button

Imports only the MQTT settings from a project file.

(15) OK button

Click **OK** button to save the settings and close the **MQTT Settings** dialog box.

When using the data registers to set the device ID (5), pressing the **OK** button calculates the size of the topic considering the device ID as 1 byte. If the set topic exceeds 256 bytes, an error will occur.

(16) Cancel button

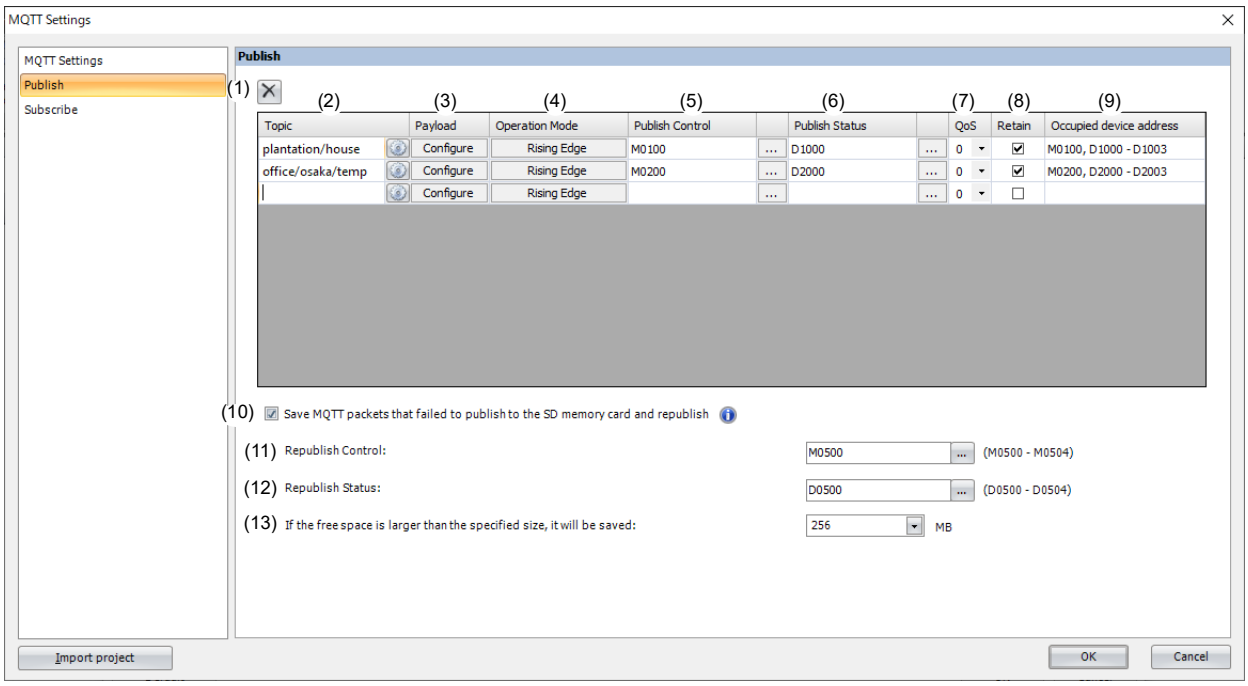
Click **Cancel** button to close the **MQTT Settings** dialog box without saving the settings.

Publish

FC6A Plus FC6A All-in-One FT2J/1J HG2J/1J HG5G/4G/3G/2G-V

When you select the **Enable MQTT Client function** check box and set **Connect to** to "General purpose MQTT Broker", "AWS IoT Core", or "Azure IoT Hub", the **Publish** tab is displayed. Configure the settings for publishing on this tab.

When you set **Connect to** to "SparkPlug B", the **Device & Tag Settings** tab is displayed. Configure the topics and metrics for edge nodes and devices on this tab. For details about the **Device & Tag Settings** tab, see "Device & Tag Settings" on page 3-59.



(1) button

Deletes the settings for the selected row.

(2) Topic

Sets the topic. The topic can be set as a UTF-8 string up to 256 bytes.
Click button to display the **Topic** dialog box. For details, see "Topic Dialog Box" on page 3-62.

- A maximum of 10 topics can be set.
- When the **Publish Control** (5) is changed from off to on, the topic set in the **Topic** dialog box is generated. When the generated topic is invalid, you cannot publish to the topic. For details on the error, see the **Publish Status** (6).
- The client ID containing special characters (/, +, or #) cannot be used.

(3) Payload

Sets the data in the payload. Click **Configure** to display the **Payload** dialog box. For details, see "Payload Dialog Box" on page 3-64.

(4) Operation Mode

Sets the operation mode. Click **Operation Mode** to display the **Operation Mode Settings** dialog box. For details, see "Operation Mode Settings Dialog Box" on page 3-47.

(5) Publish Control

Sets the device address to publish data for the topic. The bit device that can be set depends on the main unit type.

Internal Device Name		Symbol	FC6A	FT2J/1J	HG2J/1J
Internal relay		M	Yes	No	No
HMI Devices	HMI Internal Relay	LM	No	Yes	Yes
	HMI Keep Relay	LK	No	Yes	Yes
	HMI Temporary Relay	LBM	No	Yes	Yes
Control Devices	Internal relay	M	No	Yes	No

For details, see the following manuals.
FC6A: Chapter 6 "Devices" in the "FC6A Series MICROSmart User's Manual".
FT2J/1J, HG2J/1J: Chapter 35 "1 Bit Devices" in the "WindO/I-NV4 User's Manual".

When the set device address is turned on, the data is published for the topic according to **Operation Mode** (4).

(6) Publish Status

Sets the device addresses that store the status when publishing. The word device that can be specified depends on the main unit type.

Internal Device Name		Symbol	FC6A	FT2J/1J	HG2J/1J
Data Register		D	Yes	No	No
HMI Devices	HMI Data Register	LDR	No	Yes	Yes
	HMI Keep Register	LKR	No	Yes	Yes
	HMI Temporary Register	LBR	No	Yes	Yes
Control Devices	Data Register	D	No	Yes	No

For details, see the following manuals.
FC6A: Chapter 6 "Devices" in the "FC6A Series MICROSmart User's Manual".
FT2J/1J, HG2J/1J: Chapter 35 "2 Word Devices" in the "WindO/I-NV4 User's Manual".

Starting from the specified device address, 4 words of device addresses are used. Specify the first device address so that the device range is not exceeded.

<table>
<tr><th>Storage Destination</th><th>Item</th><th>Description</th></tr>
<tr><td>Starting number+0</td><td>Publish Status</td><td>Stores the status when publishing.<table><tr><th>Status Code</th><th>Status</th></tr><tr><td>0 (0000h)</td><td>Initial status</td></tr><tr><td>2 (0002h)</td><td>Publishing</td></tr><tr><td>4 (0004h)</td><td>Publish completed</td></tr><tr><td>16 (0010h)</td><td>Publish error</td></tr></table></td></tr>
<tr><td>Starting number+1</td><td>Publish Error Code</td><td>Stores information about the error that occurred when publishing.<table><tr><th>Error Code*1</th><th>Error Details</th></tr><tr><td>1 (0001h)</td><td>The Ethernet cable is disconnected or broken and the main unit cannot connect to the network properly</td></tr><tr><td>2 (0002h)</td><td>Published when the Connection Status was 0 (0000h) (disconnected)</td></tr><tr><td>4 (0004h)</td><td>Topic format is invalid or the main unit tries to publish a topic whose size is greater than 256 bytes</td></tr><tr><td>5 (0005h)</td><td>Attempted to publish JSON data larger than 32,768 bytes</td></tr><tr><td>7 (0007h)</td><td>±∞ (±infinity) or a non-numeric value was published when the data type was float, or a string with a character encoding other than UTF-8 was published when the data type was string</td></tr><tr><td>96 (0060h)</td><td>MQTT packet receive timeout error</td></tr><tr><td>32768 (8000h)</td><td>Broker response error</td></tr></table></td></tr>
<tr><td>Starting number+2</td><td>Transmitted data byte count</td><td>Stores the size of the transmitted data in bytes when Publish Status was 4 (0004h).</td></tr>
<tr><td>Starting number+3</td><td>Publish Error ID</td><td>Stores the ID*2 of the error that occurred when the Publish Error Code is 7 (0007h).</td></tr>
</table>

*1 When the Publish Error Code is a value other than 4 (0004h) to 7 (0007h), the Connection Control relay is turned off. When the Publish Error Code is 4 (0004h) to 7 (0007h), the states of the Publish Control and Connection Control relays are kept.

*2 The ID is assigned on the **Payload** dialog box to identify each element. For details, see "Payload Dialog Box" on page 3-64.

(7) QoS

Sets the QoS between 0 and 2. The default value is 0.

(8) Retain

Enables or disables Retain. When the check box is selected, Retain is enabled. The check box is cleared by default.

(9) Occupied device address

The occupied device addresses are displayed when the Publish Control and Publish Status devices are set.

(10) Save MQTT packets that failed to publish to the SD memory card and republish/ Save MQTT packets that failed to publish to the USB flash drive and republish

Sets whether MQTT packets that failed to be published are saved to the external memory drive and published again. When this check box is selected, the main unit publishes again according to the status of the **Republish Control** (11).

- MQTT packets that failed to publish are saved to the external memory as data to be published again. Publishing the republish data is called republishing.
- When republishing, set **QoS** (7) to 1 or 2. When **QoS** (7) is set to 0, the MQTT packet is not saved to the external memory as republish data because an error does not occur even if there is no response from the MQTT broker.
- The publishing process and subscribing process "Subscribe" on page 3-48 takes priority over the republishing process.
- If the client ID is changed, the republish data saved with the old client ID is deleted.

(11) Republish Control

Sets the internal relay used for republishing. The bit device that can be set depends on the main unit type.

Internal Device Name		Symbol	FC6A	FT2J/1J	HG2J/1J
Internal relay		M	Yes	No	No
HMI Devices	HMI Internal Relay	LM	No	Yes	Yes
	HMI Keep Relay	LK	No	Yes	Yes
	HMI Temporary Relay	LBM	No	Yes	Yes
Control Devices	Internal relay	M	No	Yes	No

For details, see the following manuals.
FC6A: Chapter 6 "Devices" in the "FC6A Series MICROSmart User's Manual".
FT2J/1J, HG2J/1J: Chapter 35 "1 Bit Devices" in the "WindO/I-NV4 User's Manual".

Starting from the specified device address, 5 bits of device addresses are used. Specify the first device address so that the device range is not exceeded.

Storage Destination	Item	Description
Starting number+0	Republish permitted	This bit permits republishing. When this bit is on and the republish data is saved in the external memory drive, it is republished.*1*2*3 When this bit is off, even if the republish data is saved in the external memory drive, it is not republished.
Starting number+1	Republish data save permitted	This bit permits saving republish data to the external memory drive. When this bit is on, MQTT packets that failed to be published*4 are saved to the external memory drive as republish data.*5 When this bit is off, MQTT packets are not saved to the external memory drive if publishing fails.
Starting number+2	Reserved	
Starting number+3	Reserved	
Starting number+4	Reserved	

*1 Republish data stored in the external memory is republished in no particular order. Set a timestamp in the payload if necessary.

*2 After republishing succeeded, until the republish data stored in the external memory is deleted, if the external memory is removed or the main unit is turned off, the republish data may not be deleted from the external memory. In this case, the next time the main unit connects with the MQTT broker, it republishes the republish data that could not be deleted.

*3 The next republishing process is executed at least 200 ms after the republishing process has completed. The time required for the republishing process and the time from completion of republishing process to next republishing process are longer in the following situations.

- When using other functions (Recipe function, FTP Server/Client function, etc.) and instructions (DLOG or TRACE instruction, etc.) that access the external memory
- When executing publishing or subscribing process

*4 When the Publish Error Code (6) is 4 (0004h) to 7 (0007h), MQTT packets cannot be generated because the topic or payload is invalid. In this case, the republish data is not saved to the external memory.

*5 After publishing failed, until the republish data is saved to the external memory, if the external memory is removed or the main unit is turned off, the republish data may not be saved to the external memory.

When inserting or removing the external memory, turn off the Republish permitted and Republish data save permitted bits.

(12) Republish Status

Sets the device addresses that stores the status when saving republish data or republishing. The word device that can be specified depends on the main unit type.

Internal Device Name		Symbol	FC6A	FT2J/1J	HG2J/1J
Data Register		D	Yes	No	No
HMI Devices	HMI Data Register	LDR	No	Yes	Yes
	HMI Keep Register	LKR	No	Yes	Yes
	HMI Temporary Register	LBR	No	Yes	Yes
Control Devices	Data Register	D	No	Yes	No

For details, see the following manuals.
FC6A: Chapter 6 "Devices" in the "FC6A Series MICROSmart User's Manual".
FT2J/1J, HG2J/1J: Chapter 35 "2 Word Devices" in the "WindO/I-NV4 User's Manual".

Starting from the specified device address, 5 words of device addresses are used. Specify the first device address so that the device range is not exceeded.

<table>
<tr><th>Storage Destination</th><th>Item</th><th>Description</th></tr>
<tr><td>Starting number+0</td><td>Republish Status</td><td>Stores the status when saving republish data or republishing.
<table>
<tr><th>Status Code</th><th>Status</th></tr>
<tr><td>0 (0000h)</td><td>Initial status</td></tr>
<tr><td>1 (0001h)</td><td>No republish data</td></tr>
<tr><td>2 (0002h)</td><td>Republishing</td></tr>
<tr><td>4 (0004h)</td><td>Republish completed</td></tr>
<tr><td>16 (0010h)</td><td>Republish error*1</td></tr>
<tr><td>512 (0200h)</td><td>Republish data being saved</td></tr>
<tr><td>1024 (0400h)</td><td>Republish data saving completed</td></tr>
<tr><td>2048 (0800h)</td><td>Republish data saving error*1</td></tr>
</table></td></tr>
<tr><td>Starting number+1</td><td>Republish Error Code</td><td>Stores information about the error that occurred when saving republish data or republishing.*2
<table>
<tr><th>Error Code</th><th>Error Details</th></tr>
<tr><td>3 (0003h)</td><td>Republish data cannot be stored because the client ID is not stored inside the main unit*3*6</td></tr>
<tr><td>8 (0008h)</td><td>Republish data cannot be generated because the topic or payload is invalid*3</td></tr>
<tr><td>16 (0010h)</td><td>External memory has not been inserted*3</td></tr>
<tr><td>17 (0011h)</td><td>Free space on the external memory is smaller than the specified size*3</td></tr>
<tr><td>18 (0012h)</td><td>Republish data cannot be saved due to read/write error on the external memory*3</td></tr>
<tr><td>19 (0013h)</td><td>Cannot republish due to read/write error on the external memory*4</td></tr>
<tr><td>96 (0060h)</td><td>MQTT packet receive timeout error*4*5</td></tr>
<tr><td>32768 (8000h)</td><td>Broker response error*4*5</td></tr>
</table></td></tr>
<tr><td>Starting number+2</td><td>Reserved</td><td></td></tr>
<tr><td>Starting number+3</td><td>Reserved</td><td></td></tr>
<tr><td>Starting number+4</td><td>Reserved</td><td></td></tr>
</table>

*1 The next time republishing or when the republish data is saved, the republish status is updated.
*2 The next time republishing fails or when saving republish data fails, the republish error code is updated.
*3 This error occurs when saving republish data to the external memory. In this case, the republish status is 2048 (0800h).
*4 This error occurs when republishing. In this case, the republish status is 16 (0010h).
*5 When the republish error code is 96 (0060h) to 32768 (8000h), the connection control is turned off.
*6 Change the Connection Control (Starting number+0) from off to on to store the client ID.

Republish data that fails to republish is not deleted from the external memory.

(13) If the free space is larger than the specified size, it will be saved

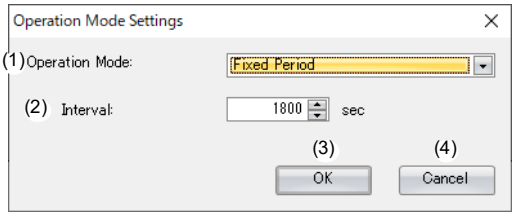
Sets the amount of free space (64MB /128MB /256MB /512MB /1024MB) on the external memory required when saving republish data. The default value is 256 MB.

When the free space on the external memory is larger than the specified size, MQTT packets that failed to be published are saved as republish data. When the free space on the SD memory card is less than the specified size, MQTT packets are not saved as republish data even if the publish fails.

Operation Mode Settings Dialog Box

Set the operation mode when publishing on the **Operation Mode Settings** dialog box.

When you set "General purpose MQTT Broker", "AWS IoT Core", or "Azure IoT Hub" in **Connect to**, the **Publish** tab is displayed. On the **Publish** tab, click the **Trigger Execution** button to open the **Operation Mode Settings** dialog box.



(1) Operation Mode

Sets the operation mode when publishing.

Operation Mode	Description
Trigger Execution	When the Publish Control relay is changed from off to on, the data is published for the specified topic. When publishing is complete, the Publish Control relay is automatically turned off.
Fixed Period	When the Publish Control relay is on, the data is published to the specified topic at the cycle set for **Interval** (2). When the Publish Control relay is turned off, publishing is ended.

(2) Interval

Sets the time interval from immediately after publish processing starts to the start of the next publish processing between 1 and 3,600 s.

The default value is 60 s. If the previous publish processing has not completed when the next publish processing is to start, the start of the next publish processing is delayed.

(3) OK button

Click **OK** button to set the operation mode and interval.

(4) Cancel button

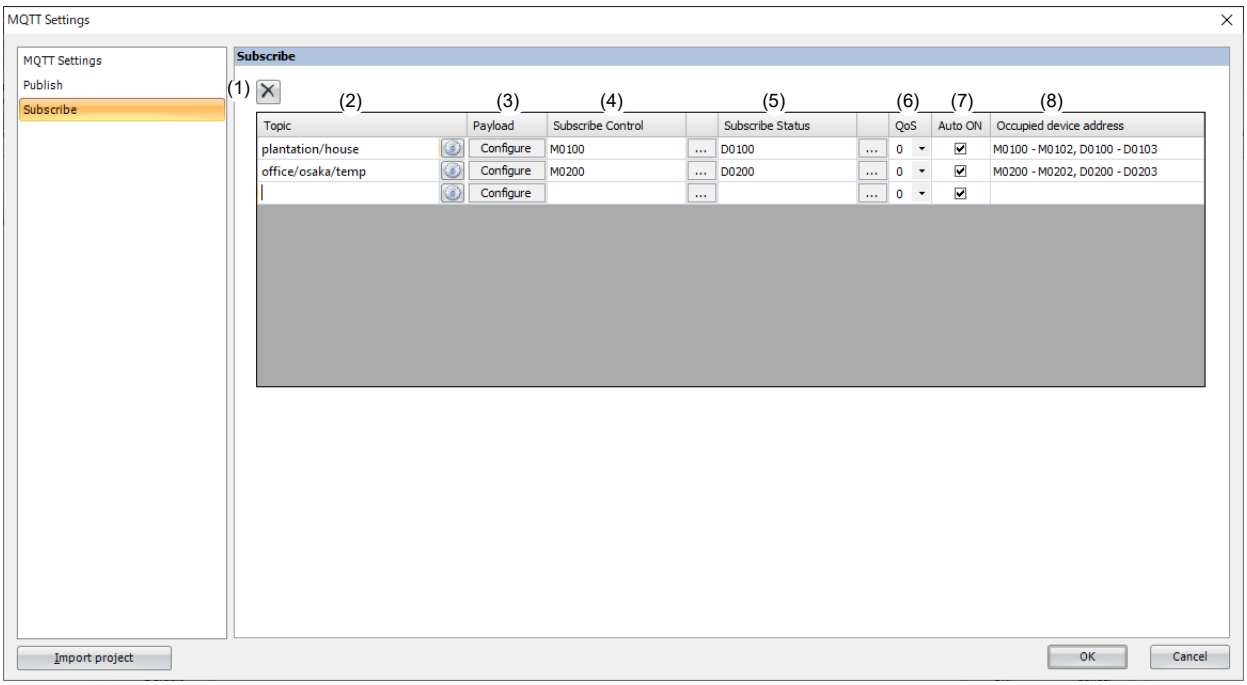
Click **Cancel** button and the operation mode and interval are not saved.

Subscribe

FC6A Plus FC6A All-in-One FT2J/1J HG2J/1J HG5G/4G/3G/2G-V

When you select the **Enable MQTT Client function** check box and set **Connect to** to "General purpose MQTT Broker", "AWS IoT Core", or "Azure IoT Hub", the **Subscribe** tab is displayed. Configure the settings for subscribing on this tab.

When you set **Connect to** to "SparkPlug B", the **Device & Tag Settings** tab is displayed. Configure the topics and metrics for edge nodes and devices on this tab. For details about the **Device & Tag Settings** tab, see "Device & Tag Settings" on page 3-59.



(1) button

Deletes the settings for the selected row.

(2) Topic

Sets the topic. The topic can be set as a UTF-8 string up to 256 bytes. Wildcards (# and +) can also be set.
Click button to display the **Topic** dialog box. For details, see "Topic Dialog Box" on page 3-62.

- A maximum of 10 topics can be set.
- When the **Subscribe Control** (4) is changed from off to on, the topic set in the **Topic** dialog box is generated. When the generated topic is invalid, you cannot subscribe to the topic. For details on the error, see the **Subscribe Status** (5).
- The client ID containing special characters (/, +, or #) cannot be used.

(3) Payload

Sets the data in the payload. Click **Configure** to display the **Payload** dialog box. For details, see "Payload Dialog Box" on page 3-64.

(4) Subscribe Control

Sets the internal relay used to subscribe to a topic. The bit device that can be set depends on the main unit type.

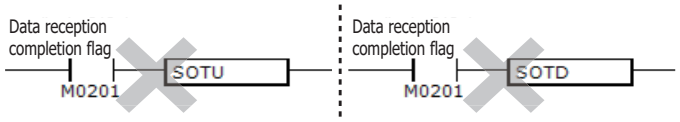
Internal Device Name		Symbol	FC6A	FT2J/1J	HG2J/1J
Internal relay		M	Yes	No	No
HMI Devices	HMI Internal Relay	LM	No	Yes	Yes
	HMI Keep Relay	LK	No	Yes	Yes
	HMI Temporary Relay	LBM	No	Yes	Yes
Control Devices	Internal relay	M	No	Yes	No

For details, see the following manuals.
FC6A: Chapter 6 "Devices" in the "FC6A Series MICROSmart User's Manual".
FT2J/1J, HG2J/1J: Chapter 35 "1 Bit Devices" in the "WindO/I-NV4 User's Manual".

Starting from the specified device address, 3 bits of device addresses are used. Specify the first device address so that the device range is not exceeded.

Storage Destination	Item	Description
Starting number+0	Subscribe Control	When on, a subscription is made to the specified topic. When off, the subscription is canceled.
Starting number+1	Data reception completion flag	This relay turns on when data was received normally (Subscribe Status remains 4 (0004)). To detect when the data is next received, turn this relay off in the ladder program.
Starting number+2	All data storage flag	This relay turns on when data was received normally (Subscribe Status remains 4 (0004)) and values were stored in all devices set on the **Payload** dialog box.

Do not use the SOTU (single output up) instruction or the SOTD (single output down) instruction in combination with the Data reception completion flag input. It may not be possible to recognize when data is received if these instructions are used in combination with the Data reception completion flag input.



(5) Subscribe Status

Sets the device addresses that store the status when subscribing to a topic. The word device that can be specified depends on the main unit type.

Internal Device Name		Symbol	FC6A	FT2J/1J	HG2J/1J
Data Register		D	Yes	No	No
HMI Devices	HMI Data Register	LDR	No	Yes	Yes
	HMI Keep Register	LKR	No	Yes	Yes
	HMI Temporary Register	LBR	No	Yes	Yes
Control Devices	Data Register	D	No	Yes	No

For details, see the following manuals.
FC6A: Chapter 6 "Devices" in the "FC6A Series MICROSmart User's Manual".
FT2J/1J, HG2J/1J: Chapter 35 "2 Word Devices" in the "WindO/I-NV4 User's Manual".

Starting from the specified device address, 4 words of device addresses are used. Specify the first device address so that the device range is not exceeded.

<table>
<tr><th>Storage Destination</th><th>Item</th><th>Description</th></tr>
<tr><td>Starting number+0</td><td>Subscribe Status</td><td>Stores the Subscribe Status.<table><tr><th>Status Code</th><th>Status</th></tr><tr><td>0 (0000h)</td><td>Initial status</td></tr><tr><td>2 (0002h)</td><td>Subscribing</td></tr><tr><td>4 (0004h)</td><td>Subscribed
(If the Subscribe Error Code is 5 (0005h) to 8 (0008h), the status is 20 (0014h).)</td></tr><tr><td>8 (0008h)</td><td>Unsubscribing</td></tr><tr><td>16 (0010h)</td><td>Subscribe error</td></tr><tr><td>32 (0020h)</td><td>Unsubscribe error</td></tr></table></td></tr>
<tr><td>Starting number+1</td><td>Subscribe Error Code</td><td>Stores information about the error that occurred in subscribing.<table><tr><th>Error Code*1</th><th>Error Details</th></tr><tr><td>1 (0001h)</td><td>The Ethernet cable is disconnected or broken and the main unit cannot connect to the network properly</td></tr><tr><td>2 (0002h)</td><td>Subscribed or unsubscribed when the Connection Status was 0 (0000h) (disconnected)</td></tr><tr><td>4 (0004h)</td><td>Topic format is invalid or the main unit tries to subscribe to a topic whose size is greater than 256 bytes</td></tr><tr><td>5 (0005h)</td><td>The size of the received data or the number of elements is out of range</td></tr><tr><td>6 (0006h)</td><td>The received data is not in the JSON format</td></tr><tr><td>7 (0007h)</td><td>A value was received that is out of range for the data type</td></tr><tr><td>8 (0008h)</td><td>Values were not stored to all devices set on the Payload dialog box</td></tr><tr><td>96 (0060h)</td><td>MQTT packet receive timeout error</td></tr><tr><td>32768 (8000h)</td><td>Broker response error</td></tr></table></td></tr>
<tr><td>Starting number+2</td><td>Received data byte count</td><td>Stores the size of the received data in bytes.</td></tr>
<tr><td>Starting number+3</td><td>Subscribe Error ID</td><td>Stores the ID*2 of the error that occurred first when one of the following errors occurs.
An element does not exist in the received JSON data that matches the depth level, name, and format of the ID set on the Payload dialog box
A value was received that is out of range for the data type</td></tr>
</table>

*1 When the Subscribe Error Code is a value other than 4 (0004h) to 8 (0008h), the Connection Control relay is turned off. When the Subscribe Error Code is 4 (0004h) to 8 (0008h), the states of the Subscribe Control and Connection Control relays are kept.

*2 The ID is assigned on the **Payload** dialog box to identify each element. For details, see "Payload Dialog Box" on page 3-64.

(6) QoS

Sets the QoS between 0 and 2. The default value is 0.

(7) Auto ON

Sets whether to automatically turn on the Subscribe Control relay immediately after the main unit connects to the broker. If the check box is selected, the Subscribe Control relay is turned on immediately after connecting to the broker. The check box is selected by default.

(8) Occupied device address

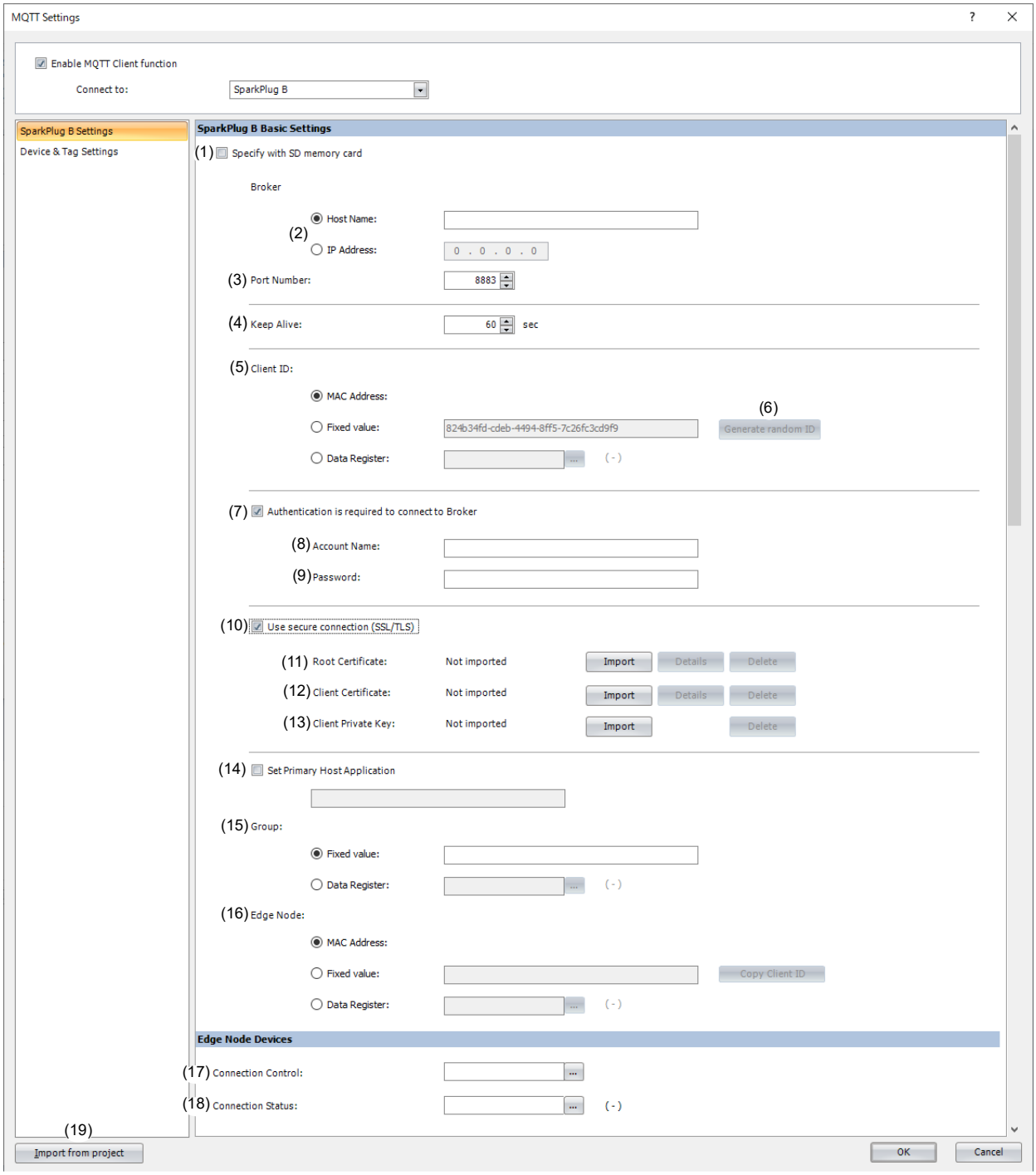
The occupied device addresses are displayed when the Subscribe Control and Subscribe Status devices are set.

SparkPlug B Settings

FC6A Plus FC6A All-in-One FT2J/1J HG2J/1J HG5G/4G/3G/2G-V

When you select the **Enable MQTT Client function** check box and set **Connect to** to "SparkPlug B", the **SparkPlug B Settings** tab is displayed. Configure the basic settings for MQTT communication on this tab.

■SparkPlug B Basic Settings



(1) Specify with SD memory card

Specifies whether to write the basic settings ((2) to (16)) for MQTT communication to the main unit by using external memory. When you select this check box, you can write the basic settings for MQTT communication to the main unit using external memory.

- **Configure SD Memory Card** button : Clicking this button opens the external memory dialog box.
  In the external memory dialog box, configure the basic settings for MQTT communication.
- For examples of how to write parameters when using external memory, see "Restrictions" on page 3-90.

(2) Host Name/IP Address

Specifies the host name or IP address of the broker. The maximum length of the host name is 128 characters. Only alphanumeric characters and the symbol (-) can be used.

When you use an external memory drive, specify the host name or IP address in "hostname" key in the [MQTT_BROKER] section of the mqtt_basic_settings.ini file.

(3) Port Number

Specifies the port number of the broker. 1883 is normally used for MQTT and 8883 is normally used for MQTT over TLS. Set the port number between 0 and 65,535. The default value is 1883. If 0 is specified for the port number, port number 1883 is used if the **Use secure connection (SSL/TLS)** (10) check box is cleared, and port number 8883 is used if that check box is selected.

When you use an external memory drive, specify the port number in "port_number" key in the [MQTT_BROKER] section of the mqtt_basic_settings.ini file.

(4) Keep Alive

Specifies the time interval to execute a connection check for the broker when a connection is made to the broker. If the set time interval or longer continues without the main unit exchanging a packet, a connection check is performed with the broker. The range that can be set is 5 to 65,535 s. The default value is 60 s.

When you use an external memory drive, specify the setting in "keep_alive" key in the [MQTT_KEEP_ALIVE] section of the mqtt_basic_settings.ini file.

(5) Client ID

Specifies the client ID. The client ID can be set from the following three types.

Client ID	Description
MAC address	Sets the MAC address of Ethernet port 1 of the Plus CPU module.*1
Fixed value	Sets any character string. The maximum length of the client ID is 128 characters. Only alphanumeric characters and symbols (ASCII data character code 0x20 to 0x7E) can be used.
Data Resister	Sets the device addresses*2 for storing the client ID. The value is read in order from the upper byte of the specified device address, treated as character data, and used as the client ID.*3 Starting from the specified device address, 64 continuous words of device addresses are used.*4 Specify the first device address so that the device range is not exceeded. Only alphanumeric characters and symbols (ASCII data character code 0x20 to 0x7E) can be used.

*1 For example, when the MAC address of Ethernet port 1 of the Plus CPU module is 12-34-56-78-9A-BC (D8324=0012h, D8325=0034h, D8326=0056h, D8327=0078h, D8328=009Ah, D8329=00BCh), the client ID is "123456789ABC".

*2 The word device that can be specified depends on the main unit type.

Internal Device Name		Symbol	FC6A
Data Register		D	Yes
HMI Devices	HMI Data Register	LDR	No
	HMI Keep Register	LKR	No
	HMI Temporary Register	LBR	No
Control Devices	Data Register	D	No

For details, see Chapter 6 "Devices" in the "FC6A Series MICROSmart User's Manual".

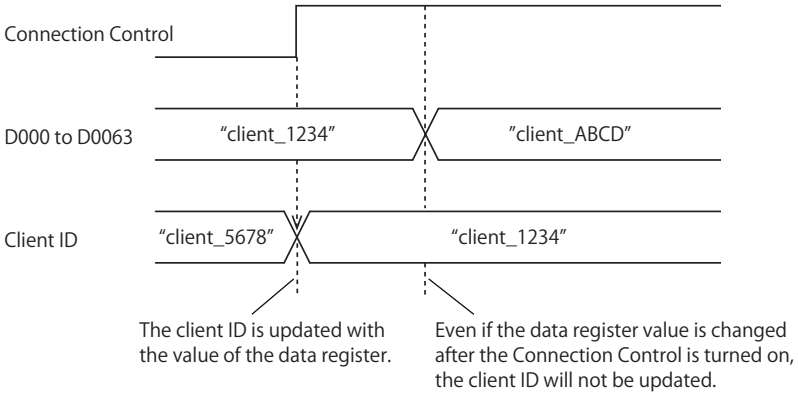
*3 For example, when D0000 is specified and the following values are stored in each data register, the client ID is "client_1234".

Data resister	Stored Value	
	Upper Byte	Lower Byte
D0000	"c"=63h	"l"=6Ch
D0001	"i"=69h	"e"=65h
D0002	"n"=6Eh	"t"=74h
D0003	"_"=5Fh	"1"=31h
D0004	"2"=32h	"3"=33h
D0005	"4"=34h	00h

*4 When the character string to be set is shorter than 128 bytes (64 words), add the terminating character NULL (00h) as the end of the character string. The data from the upper byte of the specified device address up to the data right before the terminating character NULL (00h) is treated as character data and used as the client ID.

When you use an external memory drive, specify the client ID in the [MQTT_CLIENT_ID] section of the mqtt_basic_settings.ini file.

- When setting the client ID with the data register, the MOVC instruction can be used to store a character string in the data registers. For details on MOVC instruction, see Chapter 5 "MOVC (Move Characters)" in the "Ladder Programming Manuall".
- When the Connection Control (17) is changed from off to on, the client ID stored inside the main unit is updated. For example, when the client ID is set in the data register (D0000), the client ID is updated at the following timing.



(6) Generate random ID

When the **Client ID** (5) is set as a fixed value, click this button to generate a random identifier as client ID.

(7) Authentication is required to connect to Broker

Specifies whether to perform authentication using an account name and password when connecting to the broker. If the check box is selected, authentication is performed using an account name and password when connecting to the broker. The check box is cleared by default.

When you use an external memory drive, specify the setting in "authentication" key in the [MQTT_AUTH] section of the mqtt_basic_settings.ini file.

(8) Account Name / (9) Password

Sets the account name and password when the **Authentication is required to connect to Broker** check box is selected. The maximum length of the account name is 128 characters, and the maximum length of the password is 496 characters. Only alphanumeric characters and symbols can be used.

When you use an external memory drive, specify the account name and password in "accountname" and "password" keys respectively in the [MQTT_AUTH] section of the mqtt_basic_settings.ini file.

(10) Use secure connection (SSL/TLS)

Select this check box when SSL/TLS communication is required with the destination broker. The check box is cleared by default.

When you use an external memory drive, specify the setting in "use_secure_connection" key in the [MQTT_TLS] section of the mqtt_basic_settings.ini file.

(11) Root Certificate

When the **Use secure connection (SSL/TLS)** check box is selected, you can click **Import** button and import a root certificate (pem or crt file with a size less than 2 KB (after conversion to der file)) of the server certificate to use when performing SSL/TLS communication with the broker. WindLDR converts pem and crt files to der files. When a pem file contains multiple root certificates, WindLDR converts each of the first two root certificates to a der file. Click **Details** button to display the **Certificate Information** dialog box that shows the root certificate of the imported server certificate. Click **Delete** button to delete the imported certificate.

When you use an external memory drive, specify the setting in "root_certificate" key in the [MQTT_TLS] section of the mqtt_basic_settings.ini file. If a root certificate was imported, the root_certificate key is true and an mqtt_root_certificate.der file is created. If the imported pem file contains multiple root certificates, the second root certificate from the top is be generated as the mqtt_root_certificate_2.der file. If a root certificate was not imported, the root_certificate key is false and an mqtt_root_certificate.der file is not created.

(12) Client Certificate

When the **Use secure connection (SSL/TLS)** check box is selected, you can click **Import** button and import a client certificate (pem or crt file with a size less than 2 KB (after conversion to der file)) to use when performing SSL/TLS communication with the broker. WindLDR converts pem and crt files to der files. Click **Details** button to display the **Certificate Information** dialog box that shows the imported client certificate. Click **Delete** button to delete the imported certificate.

When you use an external memory drive, specify the setting in "client_certificate" key in the [MQTT_TLS] section of the mqtt_basic_settings.ini file. If a client certificate was imported, the client_certificate key is true and an mqtt_client_certificate.der file is created. If a client certificate was not imported, the client_certificate key is false and an mqtt_client_certificate.der file is not created.

(13) Client Private Key

When the **Use secure connection (SSL/TLS)** check box is selected, you can click **Import** button and import a client private key (RSA encryption, key length 3,072 bits max.) to use when performing SSL/TLS communication with the broker. WindLDR supports the following file formats.

- PKCS#1 format pem file or der file
- PKCS#8 format (unencrypted) pem file or der file

Click **Delete** button to delete the imported private key.

When you use an external memory drive, specify the setting in "client_private_key"key in the [MQTT_TLS] section of the mqtt_basic_settings.ini file. If a client private key was imported, the client_private_key key is true and an mqtt_private_key.der file is created. If a client private key was not imported, the client_private_key key is false and an mqtt_private_key.der file is not created.

(14) Set Primary Host Application

Set whether to notify the publisher (PLC) of the status (online/offline) of the subscriber (SCADA system). When the check box is selected, the FC6A waits until it receives status notification from a subscriber that matches the configured primary host. The default is off. The primary host name can be set to a maximum of 128 bytes in UTF-8.

When you use an external memory drive, specify the setting in the [MQTT_SPARKPLUGB] section of the mqtt_basic_settings.ini file.

(15) Group

Sets the topic group ID. The group can be set from the following two types.

Group	Description
Fixed value	Sets any character string. It can be set to a maximum of 64 bytes in UTF-8. However, "+", "/" and "#" cannot be used.
Data Resister	Sets the data resisters for storing the group.*1 *2 Starting from the specified data register, 32 continuous words of data registers are used. It can be set to a maximum of 64 bytes in UTF-8. However, "+", "/" and "#" cannot be used.

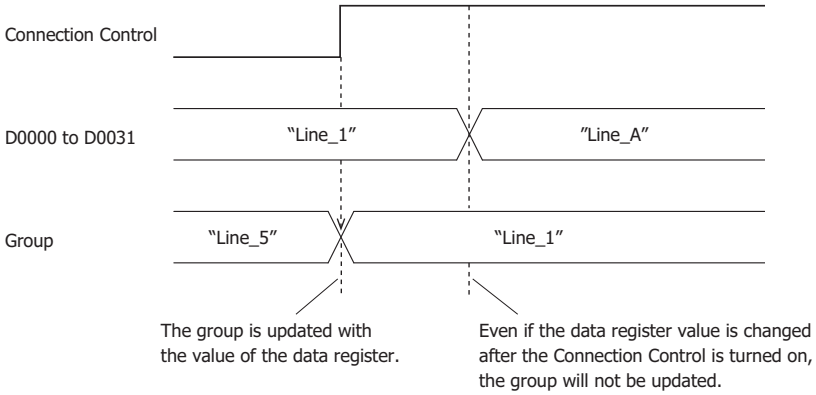
*1 For example, when D0000 is specified and the following values are stored in each data register, the group is "Line_1".

Data resister	Stored Value	
	Upper Byte	Lower Byte
D0000	"L"=4Ch	"i"=69h
D0001	"n"=6Eh	"e"=65h
D0002	"_"=5Fh	"1"=31h
D0003	00h	

*2 When the character string to be set is shorter than 64 bytes (32 words), add the terminating character NULL (00h) as the end of the character string. The data from the upper byte of the specified device address up to the data right before the terminating character NULL (00h) is treated as character data and used as the group.

When you use an external memory drive, specify the setting in the [MQTT_SPARKPLUGB] section of the mqtt_basic_settings.ini file.

- When setting the group with the data register, the MOVC instruction can be used to store a character string in the data registers. For details on MOVC instruction, see Chapter 5 "MOVC (Move Characters)" in the "Ladder Programming Manuall".
- When the Connection Control (17) is changed from off to on, the group stored inside the main unit is updated. For example, when the group is set in the data register (D0000), the group is updated at the following timing.



(16) Edge Node

Specifies the edge node. The edge node can be set from the following three types.

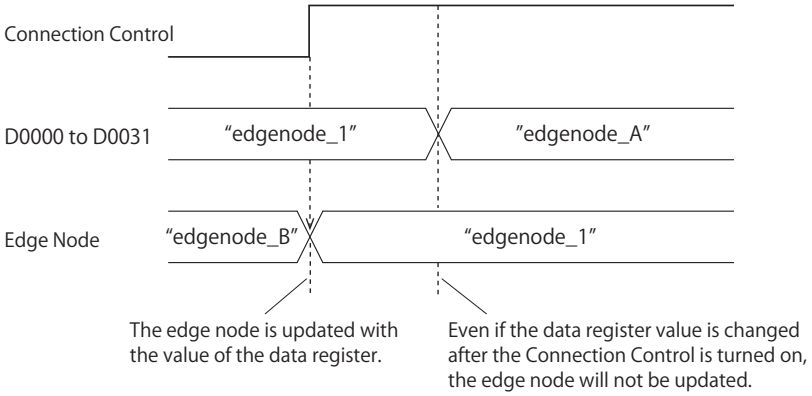
Edge Node	Description
MAC address	Sets the MAC address of Ethernet port 1 of the Plus CPU module.*1
Fixed value	Sets any character string. The edge node can be set to a maximum of 64 bytes in UTF-8. However, "+", "/" and "#" cannot be used.
Data Resister	Sets the data resisters for storing the edge node. Starting from the specified data resister, 32 continuous words of data resisters are used.*2 Specify the first data resister so that the device range is not exceeded. The edge node can be set to a maximum of 64 bytes in UTF-8. However, "+", "/" and "#" cannot be used.

*1 For example, when the MAC address of Ethernet port 1 of the Plus CPU module is 12-34-56-78-9A-BC (D8324=0012h, D8325=0034h, D8326=0056h, D8327=0078h, D8328=009Ah, D8329=00BCh), the edge node is "123456789ABC".

*2 When the character string to be set is shorter than 64 bytes (32 words), add the terminating character NULL (00h) as the end of the character string. The data from the upper byte of the specified data resister up to the data right before the terminating character NULL (00h) is treated as character data and used as the edge node.

When you use an external memory drive, specify the setting in the [MQTT_SPARKPLUGB] section of the mqtt_basic_settings.ini file.

- When setting the edge node with the data register, the MOVC instruction can be used to store a character string in the data registers. For details on MOVC instruction, see Chapter 5 "MOVC (Move Characters)" in the "Ladder Programming Manuall".
- When the Connection Control (17) is changed from off to on, the edge node stored inside the main unit is updated. For example, when the edge node is set in the data register (D0000), the edge node is updated at the following timing.



■Edge Node Devices

(17) Connection Control

Specifies the internal relay used to connect to and disconnect from the broker. When the set internal relay is turned on, a connection is made to the broker*1. When the set internal relay is turned off, the connection with the broker is disconnected.

When using the data registers to set the client ID (5), group (15) or edge node (16), the parameters stored inside the main unit is updated when the connection control is changed from off to on.

*1 If the clock of the main unit is far off from the actual current time, the main unit may not be able to connect to the SparkPlug B. Configure the clock and clock-related functions of the main unit.

When acquiring the current time (UTC) from the SNTP server

Refer to the following configuration items to configure the clock and clock-related functions of the main unit.

Function Name	Overview	Reference
SNTP Settings	Acquires the current time (UTC) from the SNTP server to adjust the clock of the main unit.	Chapter 3 "SNTP Settings" in the FC6A Series MICROSmart Communication Manual
Time Zone Settings	Configures the difference for each time zone in the region relative to standard time.	Chapter 5 "Time Zone" in the FC6A Series MICROSmart User's Manual
Daylight Savings Time Settings	Configures the daylight saving time period.	Chapter 5 "Daylight Savings Time" in the FC6A Series MICROSmart User's Manual

The clock of the main unit loses time as time goes on according to accuracy. Acquire the current time (UTC) from the SNTP server at the appropriate frequency.

When configuring the clock of the main unit directly

Refer to the following configuration items to configure the clock and clock-related functions of the main unit.

Function Name	Overview	Reference
Clock Function	Configures the clock of the main unit using WindLDR or special data registers and special internal relays.	Chapter 5 "Clock Function" in the FC6A Series MICROSmart User's Manual
Time Zone Settings	Configures the difference for each time zone in the region relative to standard time.	Chapter 5 "Time Zone" in the FC6A Series MICROSmart User's Manual
Daylight Savings Time Settings	Configures the daylight saving time period.	Chapter 5 "Daylight Savings Time" in the FC6A Series MICROSmart User's Manual

The clock of the main unit loses time as time goes on according to accuracy. Configure the current time at the appropriate frequency.

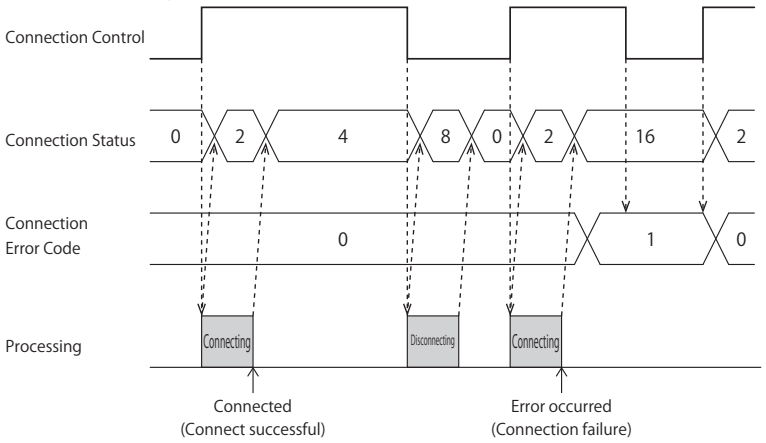
(18) Connection Status

Specifies the data resisters that store the status when connecting to the broker and the error information. Starting from the specified data resister, 3 words of data resisters are used. Specify the first data resister so that the device range is not exceeded.

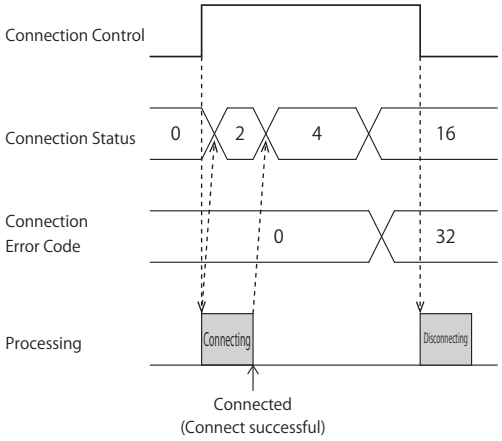
<table>
<tr><th>Storage Destination</th><th>Item</th><th>Description</th></tr>
<tr><td>Starting number+0</td><td>Connection Status</td><td>Stores the status when connecting to the broker.
<table>
<tr><th>Status Code</th><th>Status</th></tr>
<tr><td>0 (0000h)</td><td>Initial status (disconnected)</td></tr>
<tr><td>2 (0002h)</td><td>Connecting</td></tr>
<tr><td>4 (0004h)</td><td>Connected</td></tr>
<tr><td>8 (0008h)</td><td>Disconnecting</td></tr>
<tr><td>16 (0010h)</td><td>Connection processing error</td></tr>
<tr><td>32 (0020h)</td><td>Disconnection processing error</td></tr>
</table></td></tr>
<tr><td>Starting number+1</td><td>Connection Error Code</td><td>Stores information about the error that occurred when connecting to the broker.
<table>
<tr><th>Error Code</th><th>Error Details</th></tr>
<tr><td>1 (0001h)</td><td>The Ethernet cable is disconnected or broken and the main unit cannot connect to the network properly</td></tr>
<tr><td>2 (0002h)</td><td>Authentication information was not downloaded from the external memory or reading the downloaded authentication information failed</td></tr>
<tr><td>4 (0004h)</td><td>Invalid client ID format</td></tr>
<tr><td>5 (0005h)</td><td>The primary host string exceeds the maximum length, or the primary host string contains a character code other than UTF-8.</td></tr>
<tr><td>6 (0006h)</td><td>The group string exceeds the maximum length, the group string contains a character code other than UTF-8, or the group string contains "+", "/", or "#".</td></tr>
<tr><td>7 (0007h)</td><td>The edge node string exceeds the maximum length, the edge node string contains a character code other than UTF-8, or the edge node string contains "+", "/", or "#".</td></tr>
<tr><td>16 (0010h)</td><td>An unknown packet was received</td></tr>
<tr><td>32 (0020h)</td><td>An invalid MQTT packet was received</td></tr>
<tr><td>64 (0040h)</td><td>Keep alive timeout error</td></tr>
<tr><td>80 (0050h)</td><td>Packet could not arrive at destination host</td></tr>
<tr><td>96 (0060h)</td><td>MQTT packet receive timeout error</td></tr>
<tr><td>112 (0070h)</td><td>TLS error</td></tr>
<tr><td>256 (0100h)</td><td>Broker connection refused (unacceptable MQTT protocol version)</td></tr>
<tr><td>512 (0200h)</td><td>Broker connection refused (invalid client ID)</td></tr>
<tr><td>768 (0300h)</td><td>Broker connection refused (broker unavailable)</td></tr>
<tr><td>1024 (0400h)</td><td>Broker connection refused (invalid account name or password)</td></tr>
<tr><td>1280 (0500h)</td><td>Broker connection refused (not authorized)</td></tr>
<tr><td>32768 (8000h)</td><td>Broker response error</td></tr>
</table></td></tr>
<tr><td>Starting number+2</td><td>SparkPlug B Status</td><td>Stores the status of SparkPlug B.*1
<table>
<tr><th>Status Code</th><th>Status</th></tr>
<tr><td>0 (0000h)</td><td>The primary host is online.</td></tr>
<tr><td>1 (0001h)</td><td>The primary host is offline.</td></tr>
</table></td></tr>
</table>

*1 When the **Set Primary Host Application** check box is selected, the status is stored.

- The following timing chart shows the operations when the main unit connects to a broker.



- Connection Error Code 16 (0010h) to 112 (0070h) may occur even when the main unit is connected to the broker (Connection Status 4 (0004h)).



- MQTT communication is not performed while the ladder program is stopped. When switching from run to stop when connected to a broker, the Connection Control relay is turned off and disconnected 0 (0000h) is stored in the Connection Status.

(19) Import from project button

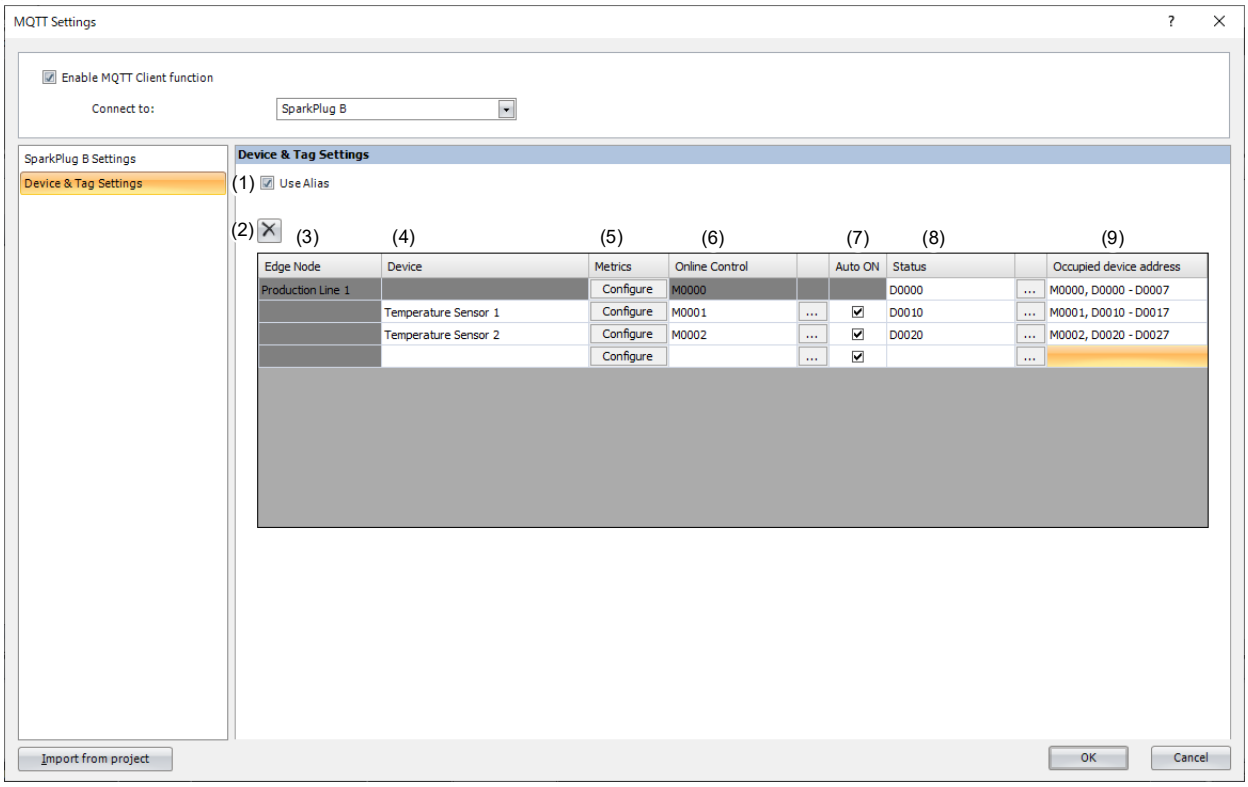
Imports only the MQTT settings from a project file.

Device & Tag Settings

FC6A Plus FC6A All-in-One FT2J/1J HG2J/1J HG5G/4G/3G/2G-V

When you select the **Enable MQTT Client function** check box and set **Connect to** to "SparkPlug B", the **Device & Tag Settings** tab is displayed. Configure topics for edge nodes or devices and metrics on this tab.

■Device & Tag Settings



(1) Use Alias

Sets whether to use metric aliases (numeric values). Used to reduce the payload size of messages that are sent or received repeatedly. When you select this check box, you can use metric aliases. The default is on.

(2) ✕ button

Deletes the setting contents of the selected line.

(3) Edge Node

The edge node you set in **Edge Node** (16) on the **SparkPlug B Settings** tab is displayed.

(4) Device

Sets the device ID for the topic. The device can set to a maximum of 111 bytes in UTF-8. However, "+", "/" and "#" cannot be used.

(5) Metrics

Sets the data in the metric. Clicking the **Configure** button opens the **Metrics** dialog box. For details, see "Metrics Dialog Box" on page 3-71.

(6) Online Control

Sets an internal relay to connect and disconnect from the broker. The online control function depends on the communication target.

Communication Target	Online Control
Edge Node	Controls the state (online/offline) of edge nodes. The internal relay set in the connection control (17) is automatically assigned. When this internal relay is turned on, it connects the edge node to the broker. When this internal relay is turned off, the connection with the broker is disconnected.
Device	Controls the state (online/offline) of the SparkPlug B device. When the set internal relay is turned on, the SparkPlug B device connects to the broker via the edge node. When the set internal relay is turned off, the connection with the broker is disconnected via the edge node.

(7) Auto ON

Sets whether to automatically turn on online control for SparkPlug B devices immediately after connecting the edge node to the broker. When this check box is selected, the online control of the SparkPlug B device is automatically turned on immediately after the edge node is connected to the broker. The default is on.

This cannot be set for an edge node. Turn on the connection control (17) to connect an edge node to a broker.

(8) Status

Specifies the data resisters that store the status of the edge node and SparkPlug B device. Starting from the specified data resister, 8 words of data resisters are used. Specify the first data resister so that the device range is not exceeded.

Edge node

<table>
<tr><th>Storage Destination</th><th>Item</th><th>Description</th></tr>
<tr><td>Starting number+0</td><td>Edge Node Status</td><td>Stores the status when connecting to the broker.
<table>
<tr><th>Status Code</th><th>Status</th></tr>
<tr><td>0 (0000h)</td><td>Edge node is not connected</td></tr>
<tr><td>2 (0002h)</td><td>Edge node is connecting</td></tr>
<tr><td>4 (0004h)</td><td>Edge node is connected (Edge node registration complete state)</td></tr>
<tr><td>16 (0010h)</td><td>Edge node connection processing error</td></tr>
</table></td></tr>
<tr><td>Starting number+1</td><td>Edge Node Error Code</td><td>Stores information about the error that occurred when sending or receiving.
<table>
<tr><th>Error Code*1</th><th>Error Details</th></tr>
<tr><td>1 (0001h)</td><td>The edge node string exceeds the maximum length, the edge node string contains a character code other than UTF-8, or the edge node string contains "+", "/", or "#".</td></tr>
<tr><td>2 (0002h)</td><td>The edge node could not subscribe to commands (NCMD) to receive control information about the edge node, or could not register the edge node with the broker.</td></tr>
<tr><td>4 (0004h)</td><td>±∞ (±infinity) or a non-numeric value was published when the data type was float, or a string with a character encoding other than UTF-8 was published when the data type was string.</td></tr>
<tr><td>8 (0008h)</td><td>A metric with an unspecified name or alias was received, or ±∞ (±infinity) or a non-number was received when the data type of the received metric was float, or a character code other than UTF-8 was received when the data type was string.</td></tr>
<tr><td>32768 (8000h)</td><td>Broker response error</td></tr>
</table></td></tr>
<tr><td>Starting number+2</td><td>Send/receive Data Error ID</td><td>Stores the ID*2 of the first error that occurred when the Edge Node Error Code is 8 (0008h) or 32768 (8000h).</td></tr>
<tr><td>Starting number+3</td><td colspan="2">Reserved</td></tr>
<tr><td>Starting number+4</td><td colspan="2">Reserved</td></tr>
<tr><td>Starting number+5</td><td colspan="2">Reserved</td></tr>
<tr><td>Starting number+6</td><td colspan="2">Reserved</td></tr>
<tr><td>Starting number+7</td><td colspan="2">Reserved</td></tr>
</table>

*1 When the Edge Node Error Code is other than 4 (0004h) or 8 (0008h), the connection control is also turned off. When the Edge Node Error Code is 4 (0004h) or 8 (0008h), the connection control state is preserved.

*2 The ID is a number assigned to each element in the **Metrics** dialog box to identify it. For details, see "Metrics Dialog Box" on page 3-71.

SparkPlug B device

<table>
<tr><th>Storage Destination</th><th>Item</th><th>Description</th></tr>
<tr><td>Starting number+0</td><td>Device Status</td><td>Stores the status when connecting to the broker via an edge node.
<table>
<tr><th>Status Code</th><th>Status</th></tr>
<tr><td>0 (0000h)</td><td>SparkPlug B device is not connected</td></tr>
<tr><td>2 (0002h)</td><td>SparkPlug B device is connecting</td></tr>
<tr><td>4 (0004h)</td><td>SparkPlug B device is connected (SparkPlug B device registration complete)</td></tr>
<tr><td>16 (0010h)</td><td>SparkPlug B device connection processing error</td></tr>
</table></td></tr>
<tr><td>Starting number+1</td><td>Device Error Code</td><td>Stores information about the error that occurred when sending or receiving.
<table>
<tr><th>Error Code*1</th><th>Error Details</th></tr>
<tr><td>1 (0001h)</td><td>The device string exceeds the maximum length, the device string contains a character code other than UTF-8, or the device string contains "+", "/", or "#".</td></tr>
<tr><td>4 (0004h)</td><td>±∞ (±infinity) or a non-numeric value was published when the data type was float, or a string with a character encoding other than UTF-8 was published when the data type was string.</td></tr>
<tr><td>8 (0008h)</td><td>A metric with an unspecified name or alias was received, or a value outside the range of the data type was received.</td></tr>
<tr><td>32768 (8000h)</td><td>Broker response error</td></tr>
</table></td></tr>
<tr><td>Starting number+2</td><td>Send/receive Data Error ID</td><td>Stores the ID*2 of the first error that occurred when the Device Error Code is 4 (0004h) or 8 (0008h).</td></tr>
<tr><td>Starting number+3</td><td colspan="2">Reserved</td></tr>
<tr><td>Starting number+4</td><td colspan="2">Reserved</td></tr>
<tr><td>Starting number+5</td><td colspan="2">Reserved</td></tr>
<tr><td>Starting number+6</td><td colspan="2">Reserved</td></tr>
<tr><td>Starting number+7</td><td colspan="2">Reserved</td></tr>
</table>

*1 When the Device Error Code is other than 4 (0004h) or 8 (0008h), the connection control is also turned off. When the Device Error Code is 4 (0004h) or 8 (0008h), the connection control state is preserved.

*2 The ID is a number assigned to each element in the **Metrics** dialog box to identify it. For details, see "Metrics Dialog Box" on page 3-71.

(9) Occupied device address

The device address occupied by setting the **Online Control** and **Status** is displayed.

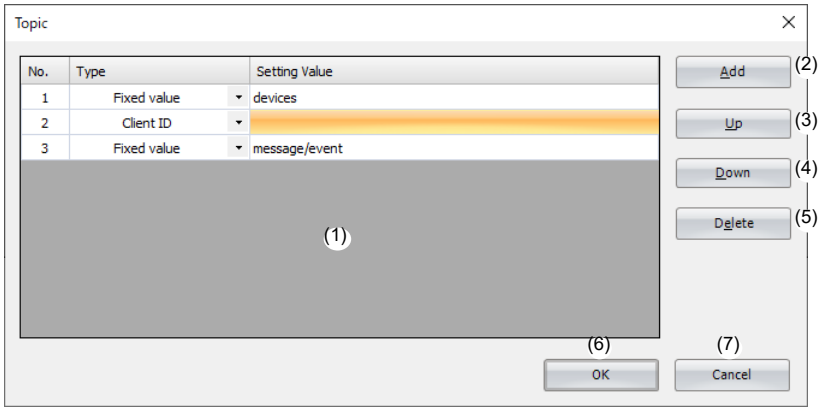
Topic Dialog Box

Use this dialog box to configure the topic.

If when the **Topic** dialog box was opened from the **Publish** tab, configure the topic to publish.

If when the **Topic** dialog box was opened from the **Subscribe** tab, configure the topic to subscribe.

Configure the string that composes the topic on each line. If you use multiple lines to configure a topic, the topic is a string consisting of the setting values of each line concatenated with "/" in number order.



(1) List of Settings

Displays the settings in this list.

No. : Displays the number (1 to 10) that identifies each row.

Type : Displays the type of string that composes the topic.

Type	Description
Fixed value	Sets any character string.
Client ID/ Device ID	The client ID or device ID set in the **MQTT Basic Settings** group of the **MQTT Settings** tab is set.

Setting Value : Displays the setting value according to the type.

(2) Add button

Adds one line to the bottom.

(3) Up button

Swaps the selected row and the row above it in List of Settings.

(4) Down button

Swaps the selected row and the row below it in List of Settings.

(5) Delete button

Deletes the selected row in List of Settings.

(6) OK button

Click **OK** button to save the settings and close the **Topic** dialog box.

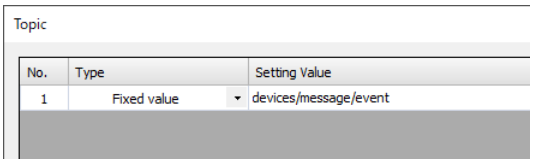
(7) Cancel button

Click **Cancel** button to close the Topic dialog box without saving the settings.

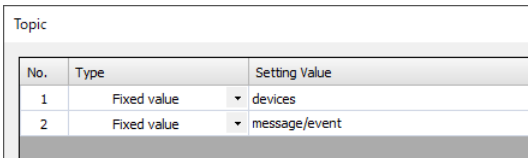
- One topic can be set with up to ten character strings (No.1 to No.10).
- Depending on the system software version of the Plus CPU module, how to configure a topic is different.

System Software	How to Configure a Topic
Less than version 2.10	Configure a topic with only one fixed value
Version 2.10 or later	Other than the above

Example of configuring a topic with only one fixed value:



Example of configuring a topic with multiple fixed values:

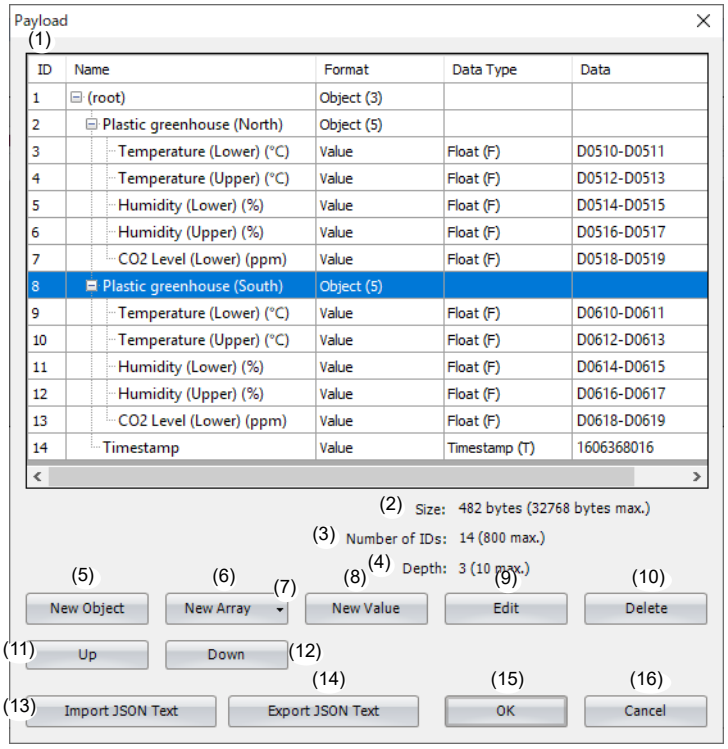


- For devices other than the Plus CPU module, you can set multiple topics to fixed values regardless of the system software version.

Payload Dialog Box

Use this dialog box to configure the data inside the payload in JavaScript Object Notation (JSON) format.

If the **Payload** dialog box was opened from the **Publish** tab, configure the content of the JSON data to transmit. If the **Payload** dialog box was opened from the **Subscribe** tab, configure what values are stored in what devices for the JSON data that was received.



(1) List of Settings

Displays the settings in this list.

ID	:	Displays the number that identifies each row.
Name	:	Displays the name of each ID.
Format	:	Displays the format of each ID (object, array, or value). Also displays the number of child elements for an object and array.
Data Type	:	Displays the data type of each value. If the data type is string (S), the size is also displayed.
Data	:	Displays a fixed value or device address. For a device address, the range of device addresses that is occupied is also displayed.

The name of ID 1 is "(root)". The names of child elements in an array are a sequential number starting from 0.

(2) Size

Displays the maximum size of the JSON data set in the list of settings. The maximum size is 32,768 bytes when the **Payload** dialog box is opened from the **Publish** tab and 8,192 bytes when the **Payload** dialog box is opened from the **Subscribe** tab.

(3) Number of IDs

Displays the number of IDs. The maximum number is 800 when the **Payload** dialog box is opened from the **Publish** tab and 200 when the **Payload** dialog box is opened from the **Subscribe** tab.

(4) Depth

Displays the current nesting depth of the JSON data set in the list of settings. The maximum value is 10.

Depth level 1 is called the root.

(5) New Object button

Adds an object to the end of the object or array selected in the list of settings. If the list of settings is empty, the object is added to the root.

(6) New Array button

Adds an array to the end of the object or array selected in the list of settings. If the list of settings is empty, the array is added to the root.

(7) Batch menu

Adds an array with continuous devices as child elements. Click this button to display the **Batch** dialog box. For details, see "Batch Dialog Box" on page 3-69.

(8) New Value button

Adds a value to the end of the object or array selected in the list of settings. If the list of settings is empty, the value is added to the root. Click this button to display the **New Value** dialog box. For details, see "New Value or Edit Dialog Box" on page 3-66.

(9) Edit button

Click this button to display the **Edit** dialog box. For details, see "New Value or Edit Dialog Box" on page 3-66.

(10) Delete button

Deletes the ID selected in the list of settings according to the format as follows.

Selected ID Format	Operation
Object	Deletes the selected object and its child elements.
Array	Deletes the selected array and its child elements.
Value	Deletes the selected value.

(11) Up button

Swaps the ID selected in the list of settings with the ID above it at the same depth.

(12) Down button

Swaps the ID selected in the list of settings with the ID below it at the same depth.

(13) Import JSON Text button

Click this button to display the **Import JSON Text** dialog box. For details, see "Import JSON Text Dialog Box" on page 3-78.

(14) Export JSON Text button

Click this button to display the **Export JSON Text** dialog box. For details, see "Export JSON Text Dialog Box" on page 3-80.

(15) OK button

Saves the settings and closes the **Payload** dialog box.

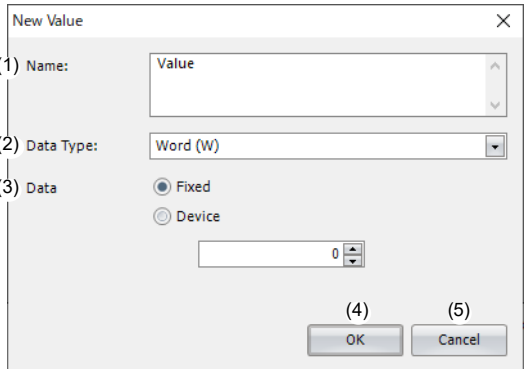
(16) Cancel button

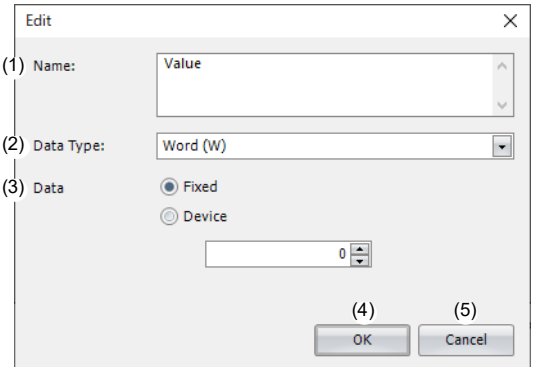
Click **Cancel** button and the settings are not saved.

New Value or Edit Dialog Box

The **New Value** dialog box is used to add object and array child elements.

The **Edit** dialog box is used to edit the names or values of objects and arrays.





(1) Name

Displays or sets the name of the value. The name can be set as a UTF-8 string up to 255 bytes.

(2) Data Type

Sets the data type of the value. This item is displayed only when adding or editing a value. The ranges of data that can be handled as each data type are as follows.

Data Type	Range
Word (W)	0 to 65,535
Integer (I)	-32,768 to 32,767
Double word (D)	0 to 4,294,967,295
Long (L)	-2,147,483,648 to 2,147,483,647
Float (F)	Range that can be handled as a single-precision 32-bit floating point number*1 However, ±∞ (±infinity) and non-numeric values are excluded.
Boolean (B)	true (1) or false (0)
String (S)	Character encoding: UTF-8 Maximum size: 1,023 bytes
Timestamp (T)*2	Time Zone: The time zone that can be set depends on the main unit type.*3 Format*6: Local Time, UTC, UNIX time

*1 For details of single-precision 32-bit floating point numbers, see the following manuals.
FT2J/1J and HG2J/1J: Chapter 2 "1.1 Available Data" in the WindO/I-NV4 User's Manual
FC6A: Chapter 3 "Data Types for Advanced Instructions" in the Ladder Programming Manual

*2 Displayed only if the **Payload** dialog box was opened from the **Publish** tab.

*3 You can set the time zone in the following table.

Main Unit	Time Zone
FC6A	Use the time zone set in the **Calendar & Clock** tab.*4
	UTC+14 to UTC-12
FT2J/1J	Use the time zone set in the internal clock.*5
HG2J/1J	

*4 Use the time zone set in the **Calendar & Clock** tab of the **Function Area Settings** dialog box. For details, see Chapter 5 "Time Zone" in the FC6A Series MICROSmart User's Manual.

*5 For details of the time zone of the internal clock, see Chapter 4 "3.19 Internal Clock Tab" in the WindO/I-NV4 User's Manual.

*6 For an explanation of the formats, see the following table.

Format		Description
Local Time (s)	YYYYMMDDThhmmss+/-hhmm	Local time in ISO 8601 basic format / extended format
	YYYY-MM-DDThh:mm:ss+/-hh:mm	
Local Time (ms)*7	YYYYMMDDThhmmssSSS+/-hhmm	
	YYYY-MM-DDThh:mm:ss.SSS+/-hh:mm	
UTC (s)	YYYYMMDDThhmmssZ	UTC in ISO 8601 basic format / extended format
	YYYY-MM-DDThh:mm:ssZ	
UTC (ms)*7	YYYYMMDDThhmmssSSSZ	
	YYYY-MM-DDThh:mm:ss.SSSZ	
UNIX Time (s)		The elapsed time in seconds (excluding leap seconds) from January 1, 1970 (UTC+0)
UNIX Time (ms)*7		The elapsed time in milliseconds (excluding leap seconds) from January 1, 1970 (UTC+0)

*7 When Local Time (ms), UTC (ms) or UNIX Time (ms) is specified, the ms digit is always 000.
For example, when the "YYYYY-MM-DDThh:mm:ss.SSS+/-hh:mm" of "Local time (ms)" is selected in the format, 2022-05-11T08:30:05.000+09:00 is displayed.

FC6A can adjust the time zone in 15-minute increments using special data register D8413.
For example, when UTC+09:00 is selected, storing +1 in D8413 advances the time by 15 minutes to make the time zone "UTC+09:15". Storing -2 in D8413 sets the time back by -30 minutes to make the time zone "UTC+08:30".

(3) Data

Sets the data of the value. This item is displayed only when adding or editing a value.

Data Type	Setting Method	Description
Word (W) / Integer (I) / Double word (D) / Long (L) / Float (F)	Fixed	Set a numeric value. The range of the numeric value that can be set depends on the data type.
	Device	The device address that can be specified depends on the main unit type.*1
Boolean (B)	Fixed	Set true or false.
	Device	The device address that can be specified depends on the main unit type.*2
String (S)	Fixed	Set a string.
	Device	Set a data register and size (in bytes) in a range between 1 and 1,023.*3
Timestamp (T)*4	—	Set the time zone and format.

*1 The word devices that can be specified are shown in the following table.

Internal Device Name		Symbol	FC6A	FT2J/1J	HG2J/1J
Data Register		D	Yes	No	No
HMI Devices	HMI Data Register	LDR	No	Yes	Yes
	HMI Keep Register	LKR	No	Yes	Yes
	HMI Temporary Register	LBR	No	Yes	Yes
Control Devices	Data Register	D	No	Yes	No

For details, see the following manuals.
FC6A: Chapter 6 "Devices" in the "FC6A Series MICROSmart User's Manual".
FT2J/1J, HG2J/1J: Chapter 35 "2 Word Devices" in the "WindO/I-NV4 User's Manual".

*2 The bit devices that can be specified are shown in the following table.

Internal Device Name		Symbol	FC6A	FT2J/1J	HG2J/1J
Internal relay		M	Yes	No	No
HMI Devices	HMI Internal Relay	LM	No	Yes	Yes
	HMI Keep Relay	LK	No	Yes	Yes
	HMI Temporary Relay	LBM	No	Yes	Yes
Control Devices	Internal relay	M	No	Yes	No

For details, see the following manuals.
FC6A: Chapter 6 "Devices" in the "FC6A Series MICROSmart User's Manual".
FT2J/1J, HG2J/1J: Chapter 35 "1 Bit Devices" in the "WindO/I-NV4 User's Manual".

*3 An address of the set size is used with the set data register as the starting address.

*4 Displayed only if the **Payload** dialog box was opened from the **Publish** tab.

(4) OK button

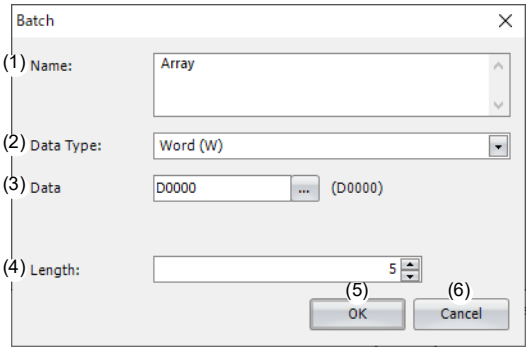
Click **OK** button to add the value to the list of settings or save the edited content.

(5) Cancel button

Click **Cancel** button and the value is not added or the edits are not saved.

Batch Dialog Box

Use this dialog box to add an array and its child elements (values).



(1) Name

Displays or sets the name of the array to add. The name can be set as a UTF-8 string up to 255 bytes.

(2) Data Type

Sets the data type of the child elements (values) of the array.

Data Type	Range
Word (W)	0 to 65,535
Integer (I)	-32,768 to 32,767
Double word (D)	0 to 4,294,967,295
Long (L)	-2,147,483,648 to 2,147,483,647
Float (F)	Range that can be handled as a single-precision 32-bit floating point number*1 However, ±∞ (±infinity) and non-numeric values are excluded.
Boolean (B)	true (1) or false (0)
String (S)	Character encoding: UTF-8 Maximum size: 1,023 bytes

*1 For details of single-precision 32-bit floating point numbers, see the following manuals.
FC6A: Chapter 3 "Data Types for Advanced Instructions" in the Ladder Programming Manual
FT2J/1J and HG2J/1J: Chapter 2 "1.1 Available Data" in the WindO/I-NV4 User's Manual

(3) Data

Sets the data of the child elements (values) of the array.

Data Type	Setting Method	Description
Word (W) / Integer (I) / Double word (D) / Long (L) / Float (F)	Device	The device address that can be specified depends on the main unit type.*1*2
Boolean (B)		The device address that can be specified depends on the main unit type.*3
String (S)		Set a device address and size (in bytes) in a range between 1 and 1,023.*4

*1 If the data type is D (double word), L (long), or F (float), 2 words of data registers are used starting from the specified device address.
*2 The word devices that can be specified are shown in the following table.

Internal Device Name		Symbol	FC6A	FT2J/1J	HG2J/1J
Data Register		D	Yes	No	No
HMI Devices	HMI Data Register	LDR	No	Yes	Yes
	HMI Keep Register	LKR	No	Yes	Yes
	HMI Temporary Register	LBR	No	Yes	Yes
Control Devices	Data Register	D	No	Yes	No

For details, see the following manuals.
FC6A: Chapter 6 "Devices" in the "FC6A Series MICROSmart User's Manual".
FT2J/1J, HG2J/1J: Chapter 35 "2 Word Devices" in the "WindO/I-NV4 User's Manual".

*3 The bit devices that can be specified are shown in the following table.

Internal Device Name		Symbol	FC6A	FT2J/1J	HG2J/1J
Internal relay		M	Yes	No	No
HMI Devices	HMI Internal Relay	LM	No	Yes	Yes
	HMI Keep Relay	LK	No	Yes	Yes
	HMI Temporary Relay	LBM	No	Yes	Yes
Control Devices	Internal relay	M	No	Yes	No

For details, see the following manuals.
FC6A: Chapter 6 "Devices" in the "FC6A Series MICROSmart User's Manual".
FT2J/1J, HG2J/1J: Chapter 35 "1 Bit Devices" in the "WindO/I-NV4 User's Manual".

*4 An address of the set size is used with the set device address as the starting address.

(4) Length

Sets the number of child elements (values) in the array. The length can be set up to the number of remaining available IDs. The default value is 1.

(5) OK button

Click **OK** button to add the array.

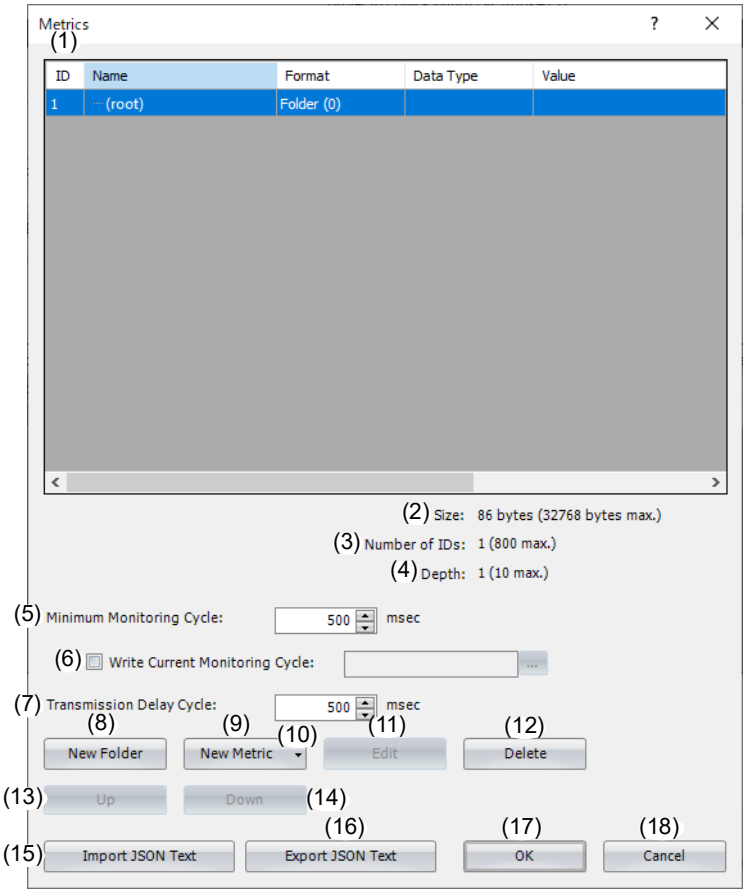
(6) Cancel button

Click **Cancel** button and the array is not added.

Metrics Dialog Box

Set the data in the metrics for SparkPlug B. Sets the data to be sent and received before encoding with Google Protocol Buffers.

The **Metrics** dialog box opens by clicking the **Configure** button on the **Device & Tag Settings** tab.



(1) List of Settings

Displays the settings in this list.

ID	:	Displays the number that identifies each row.
Name	:	Displays the name of each ID.
Format	:	Displays the format of each ID (folder or metric). Also displays the number of child elements for a folder.
Data Type	:	Displays the data type of each value. If the data type is string (S), the size is also displayed.
Data	:	Displays a fixed value or device address. For a device address, the range of device addresses that is occupied is also displayed.

- The name of ID 1 is "(root)".
- A metric is an element of metrics.

(2) Size

Displays the maximum size of the JSON data set in the list of settings. The maximum size is 32,768 bytes.

(3) Number of IDs

Displays the number of IDs. The maximum number is 800.

(4) Depth

Displays the current nesting depth of the JSON data set in the list of settings. The maximum value is 10.

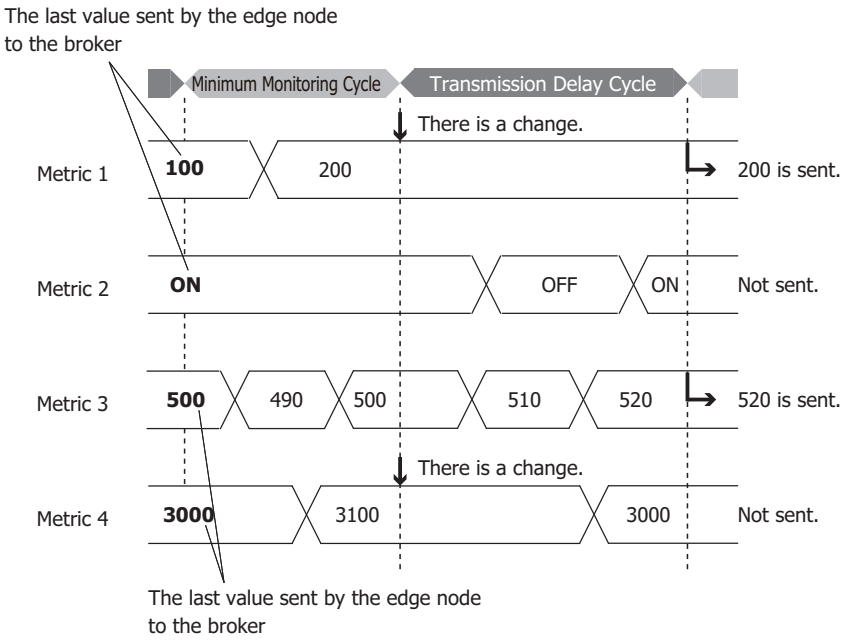
Depth level 1 is called the root.

(5) Minimum Monitoring Cycle

After an edge node connects to a broker, it monitors whether there are any changes in the device values at regular intervals. This minimum cycle is called the minimum monitoring cycle.

Compares the last value sent by the edge node to the broker to the value currently obtained from the device to monitor for changes. If there is a change within the minimum monitoring cycle, after waiting the time set in the **Transmission Delay Cycle** (7), and then again the last value sent by the edge node to the broker and the value currently obtained from the device are compared. Only if there is a change, the value currently obtained from the device will be sent to the broker.

The minimum monitoring cycle can be set from 200 to 10000 msec (100 msec units). The default is 500 msec.



(6) Write Current Monitoring Cycle

Sets the data register to which the current monitoring cycle is written. The current monitoring cycle is written to the data register in 1 msec units.

(7) Transmission Delay Cycle

If there is a change within the minimum monitoring cycle (5), after waiting the time set in the **Transmission Delay Cycle** (7), and then again the last value sent by the edge node to the broker and the value currently obtained from the device are compared. Only if there is a change, the value currently obtained from the device will be sent to the broker.

The transmission delay cycle can be set from 0 to 10000 msec (100 msec units). The default is 500 msec.

(8) New Folder button

Adds a folder to the end of the folder selected in the list of settings. If the list of settings is empty, the folder is added to the root.

(9) New Metric button

Adds a metric to the end of the folder selected in the list of settings. If the list of settings is empty, the metric is added to the root.

(10) Batch Menu

Adds a consecutive device address as a child element. Click this button to display the **Batch** dialog box. For details, see "Batch Dialog Box" on page 3-69.

(11) Edit button

Click this button to display the **Edit** dialog box. For details, see "New Value or Edit Dialog Box" on page 3-66.

(12) Delete button

Deletes the ID selected in the list of settings according to the format as follows.

Selected ID Format	Operation
Folder	Deletes the selected folder and its child elements.
Metric	Deletes the selected value.

(13) Up button

Swaps the ID selected in the list of settings with the ID above it at the same depth.

(14) Down button

Swaps the ID selected in the list of settings with the ID below it at the same depth.

(15) Import JSON Text button

Click this button to display the **Import JSON Text** dialog box. For details, see "Import JSON Text Dialog Box" on page 3-78.

(16) Export JSON Text button

Click this button to display the **Export JSON Text** dialog box. For details, see "Export JSON Text Dialog Box" on page 3-80.

(17) OK button

Saves the settings and closes the **Metrics** dialog box.

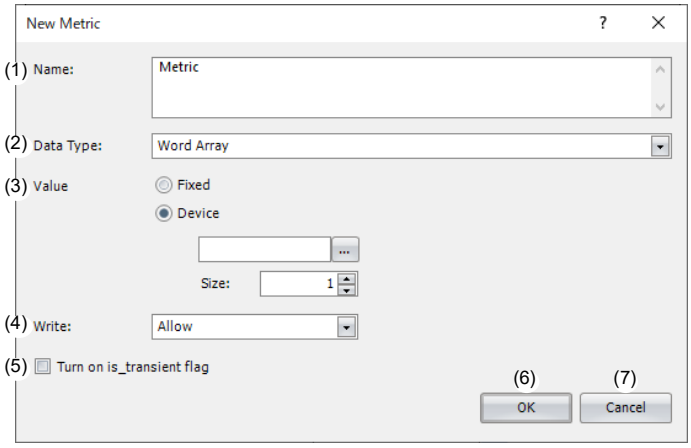
(18) Cancel button

Click **Cancel** button and the settings are not saved.

New Metric or Edit Dialog Box

The **New Metric** dialog box is used to add folder child elements.

The **Edit** dialog box is used to edit the names or values of folders.



(1) Name

Displays or sets the name of the metric. The name can be set as a UTF-8 string up to 255 bytes.

(2) Data Type

Sets the data type of the metric. The ranges of data that can be handled as each data type are as follows.

Data Type	Range
Word (W)	0 to 65,535
Integer (I)	-32,768 to 32,767
Double (D)	0 to 4,294,967,295
Long (L)	-2,147,483,648 to 2,147,483,647
Float (F)	Range that can be handled as a single-precision 32-bit floating point number*1 However, ±∞ (±infinity) and non-numeric values are excluded.
Boolean (B)	true (1) or false (0)
String (S)	Character encoding: UTF-8 Maximum size: 1,023 bytes
Timestamp (T)	Time Zone: UTC+0 Format: UNIX Time (ms)*2
Word Array	Each element of the array*3: 0 to 65,535
Integer Array	Each element of the array*3: -32,768 to 32,767
Double Array	Each element of the array*4: 0 to 4,294,967,295
Long Array	Each element of the array*4: -2,147,483,648 to 2,147,483,647
Float Array	Each element of the array*4: Range that can be handled as a single-precision 32-bit floating point number*1 However, ±∞ (±infinity) and non-numeric values are excluded.
Boolean Array	Each element of the array*3: true (1) or false (0)

*1 For details of single-precision 32-bit floating point numbers, see Chapter 3 "Data Types for Advanced Instructions" in the Ladder Programming Manual.

*2 For an explanation of the formats, see the following table.

Format	Description
UNIX Time (ms)	The elapsed time in milliseconds (excluding leap seconds) from January 1, 1970 (UTC+0)

*3 The maximum number of elements is 512.

*4 The maximum number of elements is 256.

FC6A can adjust the time zone in 15-minute increments using special data register D8413.
For example, when UTC+09:00 is selected, storing +1 in D8413 advances the time by 15 minutes to make the time zone "UTC+09:15". Storing -2 in D8413 sets the time back by -30 minutes to make the time zone "UTC+08:30".

(3) Value

Sets the value.

Data Type	Setting Method	Description
Word (W) / Integer (I) / Double word (D) / Long (L) / Float (F)	Fixed	Set a numeric value. The range of the numeric value that can be set depends on the data type.
	Device	Set a data resister.*1
Boolean (B)	Fixed	Set true or false.
	Device	Set an internal relay.
String (S)	Fixed	Set a string.
	Device	Set a data register and size (in bytes) in a range between 1 and 1,023.
Timestamp (T)	—	—
Word Array/ Integer Array/ Double Array/ Long Array/ Float Array/	Fixed	Sets a numeric value of each element in an array. The range of the numeric value that can be set depends on the data type.
	Device	Set the data register to which each element of the array is written. Set the first data register to write element [0] and the size (number of elements). The number of elements depends on the data type.
Boolean Array	Fixed	Set each element of the array. Set true or false.
	Device	Set an internal relay to which each element of the array is written. Set the first internal relay to write element [0] and the size (number of elements). The maximum number of elements is 512.

*1 If the data type is D (double word), L (long), or F (float), 2 words of data registers are used starting from the specified device address.

(4) Write

This parameter is displayed only when "Device" is selected as the setting method for **Value**. Set whether to allow writing to the set device address from the primary host.

(5) Turn on is_transient flag

Sets whether to record the last value sent by the edge node to the broker. When you select this check box, the previous value is not be recorded.

(6) OK button

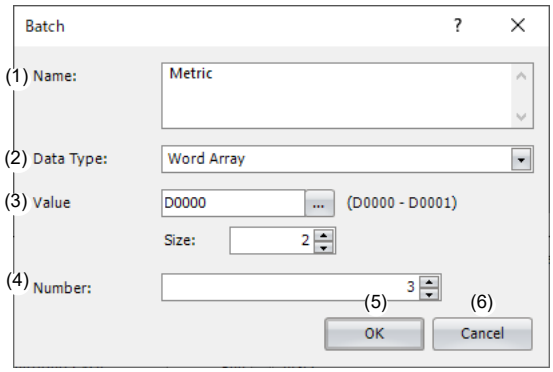
Click **OK** button to add the metric to the list of settings or save the edited content.

(7) Cancel button

Click **Cancel** button and the metric is not added or the edits are not saved.

Batch Dialog Box

Use this dialog box to add an array and its child elements (values).



(1) Name

Displays or sets the name of the array to add. The name can be set as a UTF-8 string up to 255 bytes.

(2) Data Type

Sets the data type of the metric. The ranges of data that can be handled as each data type are as follows.

Data Type	Range
Word (W)	0 to 65,535
Integer (I)	-32,768 to 32,767
Double (D)	0 to 4,294,967,295
Long (L)	-2,147,483,648 to 2,147,483,647
Float (F)	Range that can be handled as a single-precision 32-bit floating point number*1 However, ±∞ (±infinity) and non-numeric values are excluded.
Boolean (B)	true (1) or false (0)
String (S)	Character encoding: UTF-8 Maximum size: 1,023 bytes
Word Array	Each element of the array*2: 0 to 65,535
Integer Array	Each element of the array*2: -32,768 to 32,767
Double Array	Each element of the array*3: 0 to 4,294,967,295
Long Array	Each element of the array*3: -2,147,483,648 to 2,147,483,647
Float Array	Each element of the array*3: Range that can be handled as a single-precision 32-bit floating point number*1 However, ±∞ (±infinity) and non-numeric values are excluded.
Boolean Array	Each element of the array*2: true (1) or false (0)

*1 For details of single-precision 32-bit floating point numbers, see Chapter 3 "Data Types for Advanced Instructions" in the Ladder Programming Manual.

*2 The maximum number of elements is 512.

*3 The maximum number of elements is 256.

(3) Value

Sets the value.

Data Type	Description
Word (W) / Integer (I) / Double word (D) / Long (L) / Float (F)	Set the first data register to write the value.*1
Boolean (B)	Set the first internal relay to write the value.
String (S)	Set a device address and size (in bytes) in a range between 1 and 1,023.*2
Word Array/ Integer Array/ Double Array/ Long Array/ Float Array/	Set the data register to write each element of the array. Set the first data register to write element [0] and the size (number of elements). The number of elements depends on the data type.
Boolean Array	Set an internal relay to write each element of the array. Set the first internal relay to write element [0] and the size (number of elements). The maximum number of elements is 512.

*1 If the data type is D (double word), L (long), or F (float), 2 words of data registers are used starting from the specified device address.

*2 The specified data register is used as the starting address, and addresses for the specified size are used.

(4) Number

Set the number of items set by the Data Type (2) and Value (3).

(5) OK button

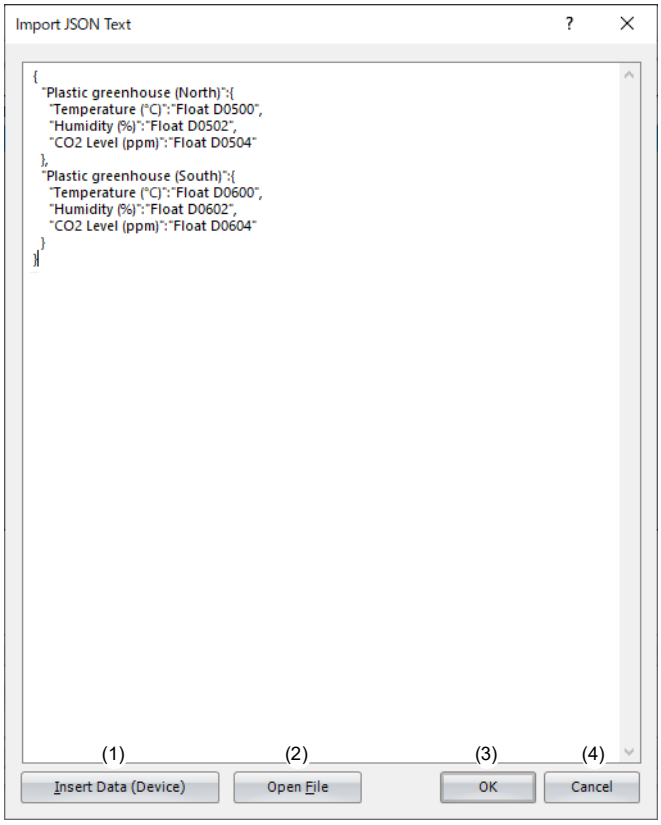
Click **OK** button to add the settings to the list of settings in the **Metrics** dialog box and close the **Batch** dialog box.

(6) Cancel button

Click **Cancel** button to return to the **Metrics** dialog box without saving any changes.

Import JSON Text Dialog Box

Use this dialog box to set the JSON text to import.



(1) Insert Data (Device)

Opens the **Insert Data (Device)** dialog box.

(2) Open File

Opens the **Open** dialog box.

(3) OK button

Applies the contents of the text to the list of settings on the **Payload** dialog box, when the dialog box is opened from the **Publish** or **Subscribe** tab.

Applies the contents of the text to the list of settings on the **Metrics** dialog box, when the dialog box is opened from the **Device & Tag Settings** tab.

If a data type and device are specified in the value portion of the key and value pairs in the JSON text, the text is converted to a device when imported.

Example: When you open the **Payload** dialog box from the **Publish** tab and import the following text

{ "key" : "Word D100" }

The data type of ID 2 is Word (W) and the Device is D0100.

{ "key" : "String D200 10" }

The data type of ID 2 is String (S) and the Device is D0200. The Size is 10 bytes.

{ "Timestamp" : "Timestamp UTC+0 UnixTime" }

The data type of ID 2 is Timestamp (T), the Time Zone is UTC+0, and the Format is UNIX Time.

Example: When you open the **Metrics** dialog box from the **Device & Tag Settings** tab and import the following text

{ "key" : "Word Array D0300 5"}

The data type of ID 2 is Word Array and the Device is D0300. The Size is 5 words.

{ "key": [0, 1, 2] }

The format of ID 2 is Metric and the data type is Integer Array.

{ "key": [0, true, 1] }

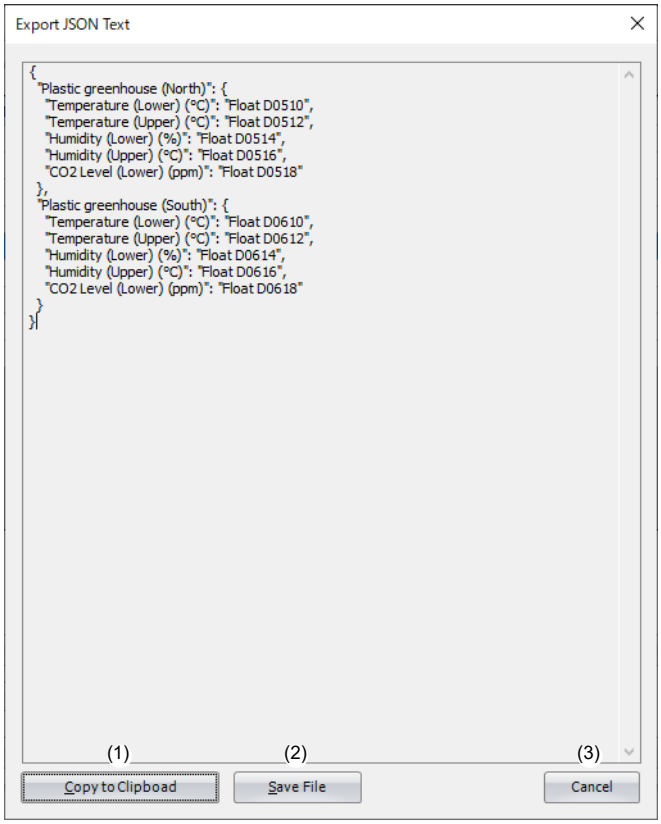
The format of ID 2 is Folder and the formats of ID3 to ID5 are Metric. Additionally, ID3 to ID5 are set to one level below ID2, and their data types are Word (W), Boolean (B), and Word (W), respectively.

(4) Cancel button

Returns to the **Payload** dialog box without applying the contents of the text to the list of settings on the **Payload** dialog box.

Export JSON Text Dialog Box

Outputs the payload configured in the **Payload** dialog box and the **Metrics** dialog box in JSON format text.



(1) Copy to Clipboad

Copies the text to the clipboard.

(2) Save File

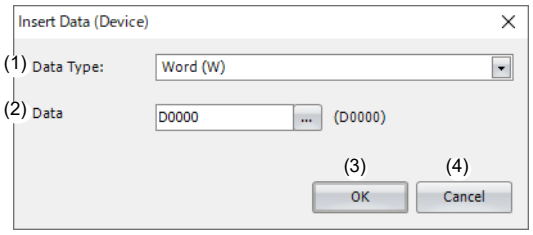
Saves the text in text format (.txt).

(3) Cancel button

Closes the **Export JSON Text** dialog box.

Insert Data (Device) Dialog Box

Use this dialog box to create a string that specifies a device and data type.



(1) Data Type

Sets the data type of the data (device) to insert. The ranges of data that can be handled as each data type are as follows.

Data Type	Range
Word (W)	0 to 65,535
Integer (I)	-32,768 to 32,767
Double (D)	0 to 4,294,967,295
Long (L)	-2,147,483,648 to 2,147,483,647
Float (F)	Range that can be handled as a single-precision 32-bit floating point number*1 However, ±∞ (±infinity) and non-numeric values are excluded.
Boolean (B)	true (1) or false (0)
String (S)	Character encoding: UTF-8 Maximum size: 1,023 bytes
Timestamp (T)*2	Time Zone: The time zone that can be set depends on the main unit type.*3 Format*6: Local Time, UTC, UNIX time
Word Array*8	Each element of the array: 0 to 65,535
Integer Array*8	Each element of the array: -32,768 to 32,767
Double Array*8	Each element of the array: 0 to 4,294,967,295
Long Array*8	Each element of the array: -2,147,483,648 to 2,147,483,647
Float Array*8	Each element of the array: Range that can be handled as a single-precision 32-bit floating point number*1 However, ±∞ (±infinity) and non-numeric values are excluded.
Boolean Array*8	Each element of the array: true (1) or false (0)

*1 For details of single-precision 32-bit floating point numbers, see the following manuals.
FT2J/1J and HG2J/1J: Chapter 2 "1.1 Available Data" in the WindO/I-NV4 User's Manual
FC6A: Chapter 3 "Data Types for Advanced Instructions" in the Ladder Programming Manual

*2 Timestamp is displayed when you open the **Payload** dialog box from the **Publish** tab, and when you open the **Metrics** dialog box from the **Device & Tag Settings** tab.

*3 You can set the time zone in the following table.

Main Unit	Time Zone
FC6A	Use the time zone set in the **Calendar & Clock** tab.*4
	UTC+14 to UTC-12
FT2J/1J	Use the time zone set in the internal clock.*5
HG2J/1J	

*4 Use the time zone set in the **Calendar & Clock** tab of the **Function Area Settings** dialog box. For details, see Chapter 5 "Time Zone" in the FC6A Series MICROSmart User's Manual.

*5 For details of the time zone of the internal clock, see Chapter 4 "3.19 Internal Clock Tab" in the WindO/I-NV4 User's Manual.

FC6A can adjust the time zone in 15-minute increments using special data register D8413.
For example, when UTC+09:00 is selected, storing +1 in D8413 advances the time by 15 minutes to make the time zone "UTC+09:15". Storing -2 in D8413 sets the time back by -30 minutes to make the time zone "UTC+08:30".

*6 For an explanation of the formats, see the following table.

Format		Description
Local Time (s)	YYYYMMDDThhmmss+/-hhmm	Local time in ISO 8601 basic format / extended format
	YYYY-MM-DDThh:mm:ss+/-hh:mm	
Local Time (ms)*7	YYYYMMDDThhmmssSSS+/-hhmm	
	YYYY-MM-DDThh:mm:ss.SSS+/-hh:mm	
UTC (s)	YYYYMMDDThhmmssZ	UTC in ISO 8601 basic format / extended format
	YYYY-MM-DDThh:mm:ssZ	
UTC (ms)*7	YYYYMMDDThhmmssSSSZ	
	YYYY-MM-DDThh:mm:ss.SSSZ	
UNIX Time (s)		The elapsed time in seconds (excluding leap seconds) from January 1, 1970 (UTC+0)
UNIX Time (ms)*7		The elapsed time in milliseconds (excluding leap seconds) from January 1, 1970 (UTC+0)

*7 When Local Time (ms), UTC (ms) or UNIX Time (ms) is specified, the ms digit is always 000.
For example, when the "YYYYY-MM-DDThh:mm:ss.SSS+/-hh:mm" of "Local time (ms)" is selected in the format, 2022-05-11T08:30:05.000+09:00 is displayed.

When "SparkPlug B" is selected in **Connect to**, the timestamp format is fixed to "UNIX time (ms)".

*8 They are displayed when you open the **Metrics** dialog box from the **Device & Tag Settings** tab.

(2) Data

Sets the data of the data (device) to insert.

Data Type	Description
Word (W)	Set a device address.*1
Integer (I)	
Double (D)	
Long (L)	
Float (F)	
Boolean (B)	Set a device address.*2
String (S)	Set a device address*1 and size (in bytes) in a range between 1 and 1,023.*3
Timestamp (T)*4	Set the time zone and format.*5
Word Array*6	Set a device address*1 and size (number of elements).
Integer Array*6	
Double Array*6	
Long Array*6	
Float Array*6	
Boolean Array*6	

*1 The word devices that can be specified are shown in the following table.

Internal Device Name		Symbol	FC6A	FT2J/1J	HG2J/1J
Data Register		D	Yes	No	No
HMI Devices	HMI Data Register	LDR	No	Yes	Yes
	HMI Keep Register	LKR	No	Yes	Yes
	HMI Temporary Register	LBR	No	Yes	Yes
Control Devices	Data Register	D	No	Yes	No

For details, see the following manuals.
FC6A: Chapter 6 "Devices" in the "FC6A Series MICROSmart User's Manual".
FT2J/1J, HG2J/1J: Chapter 35 "2 Word Devices" in the "WindO/I-NV4 User's Manual".

If the data type is D (double word), L (long), or F (float), 2 words of data registers are used starting from the specified data register.

*2 The bit devices that can be specified are shown in the following table.

Internal Device Name		Symbol	FC6A	FT2J/1J	HG2J/1J
Internal relay		M	Yes	No	No
HMI Devices	HMI Internal Relay	LM	No	Yes	Yes
	HMI Keep Relay	LK	No	Yes	Yes
	HMI Temporary Relay	LBM	No	Yes	Yes
Control Devices	Internal relay	M	No	Yes	No

For details, see the following manuals.
FC6A: Chapter 6 "Devices" in the "FC6A Series MICROSmart User's Manual".
FT2J/1J, HG2J/1J: Chapter 35 "1 Bit Devices" in the "WindO/I-NV4 User's Manual".

*3 An address of the set size is used with the set device address as the starting address.

*4 Timestamp is displayed when you open the **Payload** dialog box from the **Publish** tab, and when you open the **Metrics** dialog box from the **Device & Tag Settings** tab.

*5 Set the time zone and format only when you open the **Payload** dialog box from the **Publish** tab.

*6 They are displayed when you open the **Metrics** dialog box from the **Device & Tag Settings** tab.

(3) OK button

Creates a string from the set data type and data and inserts that at the cursor position in the **Import JSON Text** dialog box.

(4) Cancel button

Cancels inserting the data (device) and returns to the **Import JSON Text** dialog box.

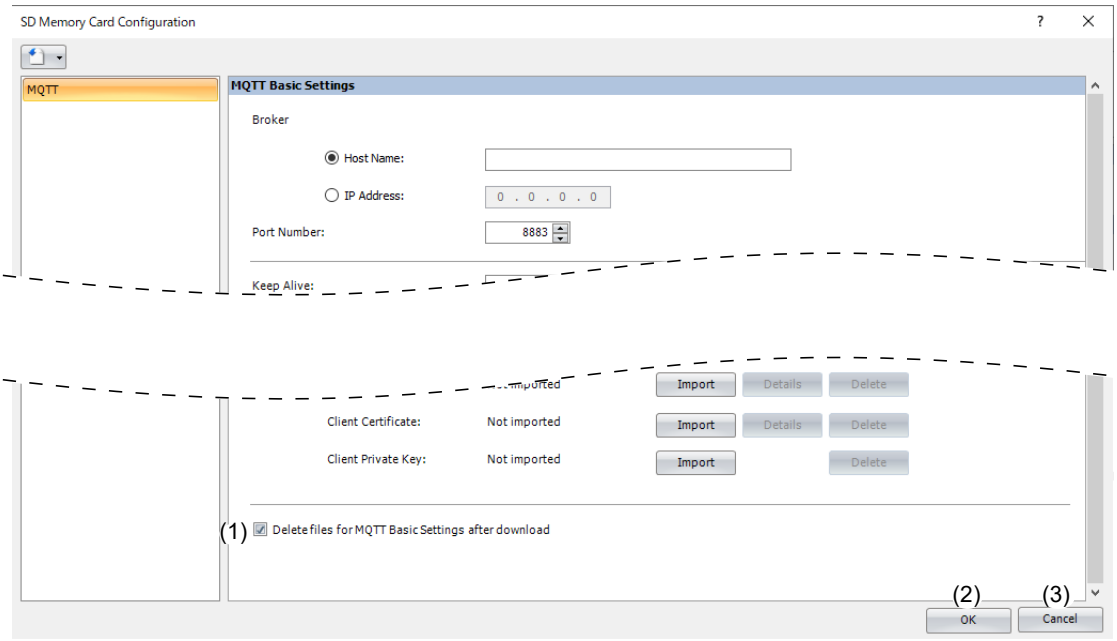
SD Memory Card Configuration, USB Flash Drive Dialog Box

If you want to write the basic settings for MQTT communication, the root certificate for the server certificate, the client certificate, and the client private key to the main unit using external memory, you must save files for MQTT basic settings to the external memory. Files for MQTT basic settings can be created on the **MQTT** tab of the external memory dialog box and saved to the external memory.

- The following five files are collectively called files for MQTT basic settings.
  mqtt_basic_settings.ini, mqtt_root_certificate.der, mqtt_root_certificate_2.der, mqtt_client_certificate.der, mqtt_client_privatekey.der
  For details about files for MQTT basic settings, see "Files for MQTT Basic Settings" on page 3-85.
- The external memory drive dialog box can be opened in the following ways.
  In the **MQTT Settings** dialog box, select **Specify with SD memory card** or **Specify with USB1** check box, and then click **Configure SD memory card** or **Configure USB1** button.

MQTT Settings

■MQTT Basic Settings



For details about each setting item on the **MQTT** tab, see the following page.

Connect to	Connection Method	Reference
General purpose MQTT Broker	—	"Connect to a general purpose MQTT broker" on page 3-8
AWS IoT Core	—	"Connect to AWS IoT Core" on page 3-15
Azure IoT Hub	Connect to Azure IoT Hub using SAS	"Connect to Azure IoT Hub using SAS" on page 3-21
	Connect to Azure IoT Hub using X.509 certificates	"Connect to Azure IoT Hub using X.509 certificates" on page 3-27
	Connect to Azure IoT Hub via DPS	"Connect to Azure IoT Hub via DPS" on page 3-32
SparkPlug B	—	"SparkPlug B Settings" on page 3-51

(1) Delete files for MQTT Basic Settings after download

Specify whether to delete the files for MQTT Basic Settings after the contents are downloaded. If this check box is selected, the files for MQTT Basic Settings are deleted after the contents are downloaded. The check box is cleared by default. Specify the setting in "delete_files" key in the [MQTT_DELETE_FILES] section of the mqtt_basic_settings.ini file.

If **Delete files for MQTT Basic Settings after download** is selected, the mqtt_basic_settings.ini file saved in the external memory folder and the mqtt_root_certificate.der, mqtt_root_certificate_2.der, mqtt_client_certificate.der, and mqtt_client_privatekey.der files saved in the \TLS folder are all deleted. Make a backup of these files in advance when using this function.

(2) OK button

Displays **Browse For Folder** dialog box.

In the **Browse For Folder** dialog box, select the drive where the external memory drive is inserted in the **Browse For Folder** dialog box and save files for MQTT basic settings.

(3) Cancel button

Closes the external memory dialog box and the settings are not saved.

Files for MQTT Basic Settings

■**mqtt_basic_settings.ini**

The mqtt_basic_settings.ini file is the configuration file that defines basic settings for performing MQTT communication.

The basic settings for performing MQTT communication, root certificate of the server certificate, client certificate, and client private key can be downloaded to the main unit by using an mqtt_basic_settings.ini file when the main unit power is turned on and by special device address operations.

The mqtt_basic_settings.ini file is created using WindLDR or WindO/I-NV4, but it can also be edited using any text editor. For the mqtt_basic_settings.ini file format, see "mqtt_basic_settings. ini File Format" on page 3-86.

When downloading or uploading data using an external memory drive, the folder that is the target of the operation is called the external memory drive folder. The FC6A is created on the SD memory card and the external memory drive folder name is "FCDATA01". The FT2J/1J and HG2J/1J models are created on a USB memory stick, and the default external memory drive folder name is "HGDATA01".

■**mqtt_root_certificate.der, mqtt_root_certificate_2.der**

This is the root certificate of the server certificate in the der format used when performing TLS communication with the destination broker in MQTT communication.

When you import the root certificate in the external memory drive dialog box, the mqtt_root_certificate.der file is generated. When the imported pem format file contains multiple root certificates, the second root certificate from the top is be generated as the mqtt_root_certificate_2.der file.

■**mqtt_client_certificate.der**

This is the client certificate in the der format used when performing TLS communication with the destination broker in MQTT communication.

When you import the root certificate in the external memory drive dialog box, the mqtt_client_certificate.der file is generated.

■**mqtt_private_key.der**

This is the client private key in the der format used when performing TLS communication with the destination broker in MQTT communication.

When you import the client private key in the external memory drive dialog box, the mqtt_private_key.der file is generated.

mqtt_basic_settings. ini File Format

The mqtt_basic_settings.ini file is made up of the following sections.

Section	Description
[COMMON_SETTING]	The section that describes the mqtt_basic_settings.ini file action.
[MQTT_BROKER]	The section that describes settings for the broker.
[MQTT_AZURE]	The section that describes settings for connecting to Azure IoT Hub.
[MQTT_KEEP_ALIVE]	The section that describes the keep alive.
[MQTT_CLIENT_ID]	The section that describes the client ID.
[MQTT_AUTH]	The section that describes settings for authentication when connecting to the broker.
[MQTT_TLS]	The section that describes settings for performing TLS communication with the broker.
[MQTT_DELETE_FILES]	The section that describes whether to delete the files for MQTT Basic Settings after the download.
[MQTT_SPARKPLUGB]	The section that describes settings for SparkPlug B.

■[COMMON_SETTING] section keys and parameters

Key	Description	Parameter	Default Value
format_version	Specifies the format version of the mqtt_basic_setting.ini file.	1	3
		2	
		3	

- The FC6A Plus CPU module has different versions of the mqtt_basic_setting.ini file format depending on the system software version. The settings supported by each format version are as follows.

Format Version	Settings	System Software
1	First format version	version 1.80 or later
2	Setting the client ID using MAC address or data register is supported.	version 2.10 or later
3	Setting based on broker type and two root certificates are supported.	version 2.20 or later

- When the "format_version" key is not specified, the format version will be processed as format version 1.
- The format version of the FT2J/1J and HG2J/1J is 3.

■[MQTT_BROKER] section keys and parameters

Key	Description	Parameter		Default Value
broker_type	Specifies the broker to connect to and the connection method.	0	Specify 0 when connecting to a general purpose MQTT broker.	0
		5	Specify 5 when connecting to an MQTT broker using SparkPlug B.	
		10	Specify 10 when connecting to AWS IoT Core.	
		20	Specify 20 when connecting to Azure IoT Hub using X.509 certificate.	
		21	Specify 21 when connecting to Azure IoT Hub using SAS.	
		22	Specify 22 when connecting to Azure IoT Hub via DPS.	
hostname	Specifies the host name or IP address of the broker, and the device address to store the host name.	Host Name	• The host name is enclosed in quotation marks (" "). • The maximum length of the host name is 128 characters.	"www.example.com"
		IP Address	Specify the IP address without enclosing it in quotation marks (" ").	
		Device Address	Specify the device address to store the host name of the Azure IoT Hub when connecting to the Azure IoT Hub via DPS.*1*2	
port_number	Specifies the port number.	Specify the port number between 0 and 65535 without enclosing it in quotation marks (" ").		• When the "broker_type" key is 0, 1883 • When the "broker_type" key is not 0, 8883

*1 The word device that can be specified depends on the main unit type.

Internal Device Name		Symbol	FC6A	FT2J/1J	HG2J/1J
Data Register		D	Yes	No	No
HMI Devices	HMI Data Register	LDR	No	Yes	Yes
	HMI Keep Register	LKR	No	Yes	Yes
	HMI Temporary Register	LBR	No	Yes	Yes
Control Devices	Data Register	D	No	Yes	No

For details, see the following manuals.
FC6A: Chapter 6 "Devices" in the "FC6A Series MICROSmart User's Manual".
FT2J/1J, HG2J/1J: Chapter 35 "2 Word Devices" in the "WindO/I-NV4 User's Manual".

*2 The setting method depends on the device type.
When setting a data register, set only the address number.
Example: When setting D0100, hostname=100
When setting a device other than a data register, set the device symbol and address number as a string.
Example: hostname="LDR100"

When setting the broker hostname using the device address, starting from the specified device address, 64 continuous words of device addresses are used. Specify the first device address so that the device range is not exceeded.

■[MQTT_AZURE] section keys and parameters

Key	Description	Parameter	Default Value
connection_string	Specifies the connection string (primary or secondary connection string) assigned to the main unit in Azure IoT Hub.	• The connection string is enclosed in quotation marks (" "). • The maximum length of the connection string is 300 characters.	Blank
dps_endpoint	Specifies the service endpoint for the DPS.	• The service endpoint is enclosed in quotation marks (" "). • The maximum length of the service endpoint is 128 characters.	Blank
dps_port_number	Specifies the port number for the DPS.	Specify the port number between 0 and 65535 without enclosing it in quotation marks (" ").	Blank
dps_id_scope	Specifies the ID scope of the DPS.	• The ID scope is enclosed in quotation marks (" "). • The maximum length of the ID scope is 11 characters.	Blank
dps_enrollment_group_symmetric_key	Specifies the symmetric key of the registration group of the DPS.	• The symmetric key is enclosed in quotation marks (" "). • The maximum length of the symmetric key is 88 characters.	Blank

■[MQTT_KEEP_ALIVE] section keys and parameters

Key	Description	Parameter	Default Value
keep_alive	Specifies the keep alive.	Specify the keep alive between 5 and 65535 s without enclosing it in quotation marks (" ").	60

■[MQTT_CLIENT_ID] section keys and parameters

Key	Description	Parameter		Default Value
client_id_type	Specifies the type of client ID.	mac	Sets the MAC address of the main unit interface to the client ID.	mac
		string	Sets the character string set in the "client_id" key to the client ID.	
		dr	Sets the client ID using the data register set in the "client_id_dr" key.	
client_id	Specifies any character string.	The client ID is enclosed in quotation marks (" "). The maximum length of the client ID is 128 characters. Only alphanumeric characters and symbols can be used.		Random string
client_id_dr	Specifies the device address.*1*2	Data register: Set only the address number of the data register. Device other than a data register: Set the device address as a string.		Blank

*1 The word device that can be specified depends on the main unit type.

Internal Device Name		Symbol	FC6A	FT2J/1J	HG2J/1J
Data Register		D	Yes	No	No
HMI Devices	HMI Data Register	LDR	No	Yes	Yes
	HMI Keep Register	LKR	No	Yes	Yes
	HMI Temporary Register	LBR	No	Yes	Yes
Control Devices	Data Register	D	No	Yes	No

For details, see the following manuals.
FC6A: Chapter 6 "Devices" in the "FC6A Series MICROSmart User's Manual".
FT2J/1J, HG2J/1J: Chapter 35 "2 Word Devices" in the "WindO/I-NV4 User's Manual".

*2 The setting method depends on the device type.
When setting a data register, set only the address number.
Example: When setting D0100, client_id_dr=100
When setting a device other than a data register, set the device symbol and address number as a string.
Example: client_id_dr="LDR100"

When setting the client ID using the device address, starting from the specified device address, 64 continuous words of device addresses are used. Specify the first device address so that the device range is not exceeded.

■[MQTT_AUTH] section keys and parameters

Key	Description	Parameter		Default Value
authentication	Specifies the setting for authentication when connecting to the broker.	true	Performs authentication using an account name and password when connecting to the broker.	false
		false	Does not perform authentication using an account name and password when connecting to the broker.	
accountname	Specifies the account name.	• The account name is enclosed in quotation marks (" "). • The maximum length of the account name is 128 characters.		Blank
password	Specifies the password.	• The password is enclosed in quotation marks (" "). • The maximum length of the password is 496 characters.		Blank

■ [MQTT_TLS] section keys and parameters

Key	Description	Parameter		Default Value
use_secure_connection	Specifies whether to perform TLS communication with the broker.	true	Performs TLS communication with the broker.	false
		false	Does not perform TLS communication with the broker.	
root_certificate	Specifies whether there is a root certificate of the server certificate.	true	There is a root certificate of the server certificate.	false
		false	There is no root certificate of the server certificate.	
client_certificate	Specifies whether there is a client certificate.	true	There is a client certificate.	false
		false	There is no client certificate.	
client_private_key	Specifies whether there is a client private key.	true	There is a client private key.	false
		false	There is no client private key.	

■[MQTT_DELETE_FILES] section keys and parameters

Key	Description	Parameter		Default Value
delete_files	Specifies whether to delete the files for MQTT Basic Settings after the download.	true	Deletes the files for MQTT Basic Settings.	false
		false	Does not delete the files for MQTT Basic Settings.	

■[MQTT_SPARKPLUGB] section keys and parameters

Key	Description	Parameter		Default Value
use_primary_host	Specifies whether to set the primary host.	true	Set the primary host.	false
		false	Does not set the primary host.	
primary_host_name	Specifies the primary host name.	• The primary host name is enclosed in quotation marks (" "). • The maximum length of the primary host name is 128 bytes in UTF-8.*1		Blank
group_id_type	Specifies the type of group ID for the topic.	string	Sets the character string set in the "group_id" key to the group ID.	string
		dr	Sets the group ID using the device address set in the "group_id_dr" key.	
group_id	Specifies any character string.	• The character string is enclosed in quotation marks (" "). • The maximum length of the group ID is 64 bytes in UTF-8.*1		Random string
group_id_dr	Specifies the data resister.	Sets only the address number of the data register.		Blank
edge_node_id_type	Specifies the edge node.	mac	Sets the MAC address of the main unit interface to the edge node.	mac
		string	Sets the character string set in the "edge_node_id" key to the edge node.	
		dr	Sets the edge node using the data register set in the "edge_node_dr" key.	
edge_node_id	Specifies any character string.	• The character string is enclosed in quotation marks (" "). • The maximum length of the edge node is 64 bytes in UTF-8.*1		Random string
edge_node_id_dr	Specifies the data resister.	Sets only the address number of the data register.		Blank

*1 "+", "/" and "#" cannot be set.

Restrictions

- The maximum number of characters per line is 512 single-byte characters including line feed codes. All the text on the line will be ignored if there are more than 512 single-byte characters on the line.
- Each item must be described as a single line. If a line feed occurs before the end of the line, all characters after the line feed are ignored.
- Only line feed codes of the format generally supported by Windows (CR+LF) are supported. If any other line feed format is used, the mqtt_basic_settings.ini file will fail to run properly.
- Data from a semicolon (;) to the line feed code is ignored. The space after the semicolon can be freely used for comments. However, if the mqtt_basic_settings.ini file is opened in WindLDR and then saved again, comments written after the semicolon will be deleted and the file will only contain standard comments.
- Only strings with single-byte characters can be used.
- If the same key appears multiple times, only the bottom key is valid.

■mqtt_basic_settings.ini file example

- Example of setting an arbitrary character string as the client ID

```
[COMMON_SETTING]
format_version=2
[MQTT_BROKER]
hostname="www.example.com"
port_number=8883
[MQTT_KEEP_ALIVE]
keep_alive=60
[MQTT_CLIENT_ID]
client_id_type=string
client_id="device0"
[MQTT_AUTH]
authentication=true
accountname="idec"
password="password"
[MQTT_TLS]
use_secure_connection=true
root_certificate=true
client_certificate=true
client_private_key=true
[MQTT_DELETE_FILES]
delete_files=true
```

- Example of setting the client ID using the data register

```
[COMMON_SETTING]
format_version=2
[MQTT_BROKER]
hostname="www.example.com"
port_number=8883
[MQTT_KEEP_ALIVE]
keep_alive=60
[MQTT_CLIENT_ID]
client_id_type=dr
client_id_dr=100
[MQTT_AUTH]
authentication=true
accountname="idec"
password="password"
[MQTT_TLS]
use_secure_connection=true
root_certificate=true
client_certificate=true
client_private_key=true
[MQTT_DELETE_FILES]
delete_files=true
```

- Example of connecting to Azure IoT Hub via DPS

```
[COMMON_SETTING]
format_version=3
[MQTT_BROKER]
broker_type=23
hostname=100
[MQTT_AZURE]
dps_endpoint=”example.azure-devices-provisioning.net”
dps_port_numer=8883
dps_id_scope=”ABCDEFGH000”
dps_enrollment_group_symmetric_key=”dMksEu5HCU/fOQdmJepJCuX6c6eslQDMXzXnGdaClqrjeej8bLTtu290/SG4oNqV3KV62jZA==”
[MQTT_KEEP_ALIVE]
keep_alive=60
[MQTT_CLIENT_ID]
client_id_type=mac
[MQTT_TLS]
use_secure_connection=true
root_certificate=true
[MQTT_DELETE_FILES]
delete_files=true
```

Download the MQTT Basic Settings

The files for MQTT basic settings files saved in the external memory drive can be downloaded to the main unit.

The files for MQTT Basic Settings can be downloaded by performing the following operations.

Operation	Action
Turn on the main unit power when an external memory drive to which a mqtt_basic_settings.ini file was saved in the external memory drive folder is inserted in the interface.	The setting contents specified by the mqtt_basic_settings.ini file are downloaded to the main unit. When downloading certificates and other files, the certificates and private key in the der format saved in the \TLS\ folder are downloaded.
Turn on the special internal relay when an external memory drive to which a mqtt_basic_settings.ini file was saved in the external memory drive folder is inserted in the interface.	

Simply inserting an external memory drive with the files for mqtt_basic_settings.ini saved in the external memory drive folder into the main unit interface is not download the files for MQTT basic settings.

- The files for MQTT Basic Settings can be downloaded even if the project in the main unit is protected.
- If you select **Specify with SD memory card**/ **Specify with USB1** check box in **MQTT Basic Settings** on the **MQTT Settings** dialog box, and download that user program, make sure to download the files for MQTT Basic Settings from the external memory drive.
- If you clear **Specify with SD memory card/ Specify with USB1** check box in **MQTT Basic Settings** on the **MQTT Settings** dialog box, and download that project, the MQTT basic settings set in the project are used, even if files for MQTT Basic Settings are downloaded from an external memory drive.
- The root certificate, client certificate, and client private key downloaded from the external memory drive card cannot be uploaded.
- If an mqtt_basic_settings.ini file exists in the external memory drive folder on the external memory drive, the files for MQTT basic settings will be downloaded to the main unit each time the main unit power is turned on. To avoid this from happening, use **Delete files for MQTT Basic Settings after download**. For details, see "(1) Delete files for MQTT Basic Settings after download" on page 3-84.
- MQTT communication does not work, even if the root certificate, client certificate, and client private key in pem format are saved to the external memory drive and downloaded to the main unit. Convert them to der format and save them to the \TLS\ folder on the SD memory card. Save the private key in PKCS#1 format to the \TLS\ folder on the external memory drive. The conversion from pem format to der format can be done in the **SD Memory Card Configuration/ USB Flash Drive** dialog box or in the Data File Manager. For details on the Data File Manager, see Chapter 4 "Command Line" in the Data File Manager User's Manual.

Special Devices

The special devices used to download the files for MQTT basic settings to the main unit depend on the main unit type.

Internal Device Name		FC6A	FT2J/1J	HG2J/1J
Special Internal Relay		Yes	No	No
Special Data Resister		Yes	No	No
HMI Device	HMI Special Internal Relay	No	Yes	Yes
	HMI Special Data Resister	No	Yes	Yes

R/W is an abbreviation for read/write.
The notation for the R/W field is as follows.
R/W: The device can be both read from and written to
R: Read-only
W: Write-only

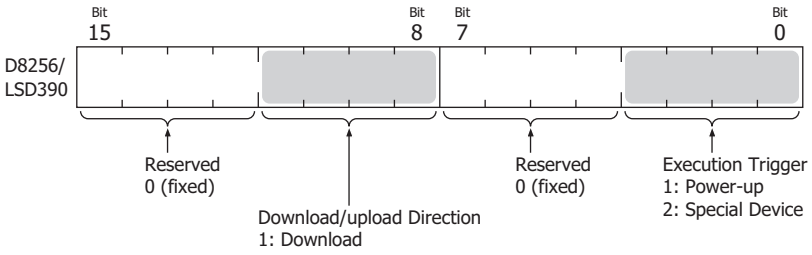
■Special Internal Relay, HMI special Internal Relay

Special Internal Relay	HMI Special Internal Relay	Description		R/W
M8271	LSM83	Download MQTT Basic Settings from SD Memory Card Execution Flag	When this internal relay is turned on, the files for MQTT Basic Settings are downloaded.	R/W
M8272	LSM84	Download MQTT Basic Settings from SD Memory Card Execution Completed Output	This internal relay is turned off when downloading the files for MQTT basic settings is started, and it is turned on once the download is finished.	R
M8273	LSM85	Download MQTT Basic Settings from SD Memory Card Error Output	This internal relay is turned off when downloading the files for MQTT basic settings is started. It is turned on if an error occurs during the process.	R

■**Special Data Resister, HMI Special Data Resister**

Special Data Resister	HMI Special Data Resister	Description		R/W
D8256	LSD390	Download MQTT Basic Settings from External Memory Drive Execution Information	Information in executing the download of the files for MQTT Basic Settings. For details, see "Download MQTT Basic Settings from External Memory Drive Execution Information" on page 3-92.	R
D8257	LSD391	Download MQTT Basic Settings from External Memory Drive Execution Error Information	Error information in executing the download of the files for MQTT Basic Settings. For details, see "Download MQTT Basic Settings from External Memory Drive Execution Error Information" on page 3-93.	R

Download MQTT Basic Settings from External Memory Drive Execution Information
The allocation of bits in the device (bit assignment) is as follows.



Download MQTT Basic Settings from External Memory Drive Execution Error Information

Error Code	Description	LSM85/ M8273	ROM Status
0	Normal termination	OFF	Refreshed
1	M8271/LSM83 (Download MQTT Basic Settings from External Memory Drive Execution Flag) was turned on when an external memory drive was not inserted.	ON	Not refreshed
2	In the mqtt_basic_settings.ini file, delete_files=true was set, but the SD memory card is write protected.	ON	Not refreshed
3	Invalid external memory drive format.	ON	Not refreshed
4	M8271/LSM83 (Download MQTT Basic Settings from External Memory Drive Execution Flag) was turned on when there is no mqtt_basic_settings.ini file on the external memory drive.	ON	Not refreshed
5	In the mqtt_basic_settings.ini file, root_certificate=true was set, but there is no mqtt_root_certificate.der file on the external memory drive or reading the file failed.*1	ON	Not refreshed
6	In the mqtt_basic_settings.ini file, client_certificate=true was set, but there is no mqtt_client_certificate.der file on the external memory drive or reading the file failed.*1	ON	Not refreshed
7	In the mqtt_basic_settings.ini file, client_private_key=true was set, but there is no mqtt_private_key.der file on the external memory drive or reading the file failed.*1	ON	Not refreshed
8	Attempted to save mqtt_root_certificate.der file to the external memory drive, but the file size is invalid.	ON	Not refreshed
9	Attempted to save mqtt_client_certificate.der file to the external memory drive, but the file size is invalid.	ON	Not refreshed
10	Attempted to save mqtt_private_key.der file to the external memory drive, but the file size is invalid.	ON	Not refreshed
11	Downloading the files for MQTT Basic Settings failed.	ON	Undefined
12	Reading the mqtt_basic_settings.ini file failed.*1	ON	Not refreshed
13	In the mqtt_basic_settings.ini file, the settings in the [MQTT_SPARKPLUGB] section are invalid.	ON	Not refreshed
14	In the mqtt_basic_settings.ini file, the primary_host_name key in the [MQTT_SPARKPLUGB] section is invalid.	ON	Not refreshed
15	In the mqtt_basic_settings.ini file, the group_id key in the [MQTT_SPARKPLUGB] section is invalid.	ON	Not refreshed
16	In the mqtt_basic_settings.ini file, the edge_node_id key in the [MQTT_SPARKPLUGB] section is invalid.	ON	Not refreshed
17	The number of device addresses used for the FT2J/1J has exceeded the upper limit*2, or the device type for the FT2J/1J or HG2J/1J is invalid.	*3	Not refreshed

*1 Status when the data in the files for MQTT Basic Settings cannot be read properly from the external memory drive for a reason such as an error in the data in the files for MQTT Basic Settings or a specific area on the external memory drive has malfunctioned.

*2 The total number of control devices, including publish and subscribe settings, has exceeded the maximum number.

*3 LSM85 turns on.

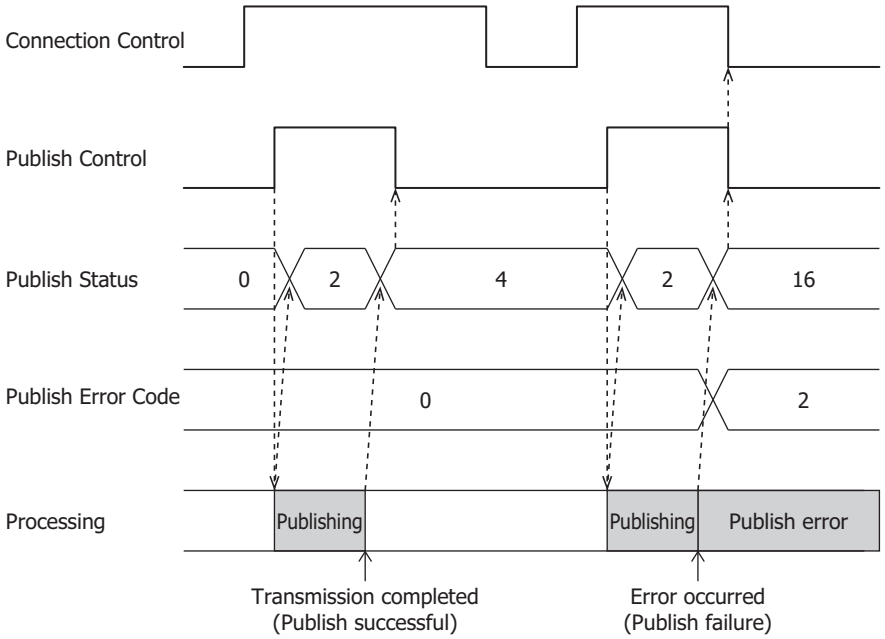
Publishing to a Topic

The main unit publishes data according to the **Operation Mode** setting on the **Operation Mode Settings** dialog box. For details, see "Operation Mode Settings Dialog Box" on page 3-47.

When the Operation Mode Is Set to Trigger Execution

After the main unit connects to the broker and when the Publish Control relay is changed from off to on, the data is published to the specified topic and the status is stored in Publish Status.

If publishing was successful, status 4 (0004h) is stored in Publish Status and the Publish Control relay is turned off. If publishing fails, status 16 (0010h) is stored in Publish Status, the error code is stored in Publish Error Code, and the Publish Control relay is turned off. When there is an error other than error code 7 (0007h), the Connection Control relay is also turned off. When the error is error code 7 (0007h), the states of the Publish Control and Connection Control relays are kept.



Do not keep the Publish Control relay always on. If publishing was successful, status 4 (0004h) is stored in Publish Status and the Publish Control relay is turned off. If the Publish Control relay is always on, the data will be published repeatedly. Be careful as this may result in extremely high packet communication fees.

When turning off the Publish Control relay, publishing may not execute if the Publish Control relay is turned off when the status is a value other than status 2 (0002h).

When Publish Status is status 2 (0002h), do not change the values of devices set on the **Payload** dialog box. Change the values when Publish Status is a value other than status 2 (0002h).

Operation Example

Operation

- Temperature, humidity, and CO_2 concentration are measured at two locations (north and south) in a greenhouse for strawberry cultivation.
- If the measured temperature, humidity, and/or CO_2 concentration are outside the following ranges, the values are published to the topic "plantation/house".

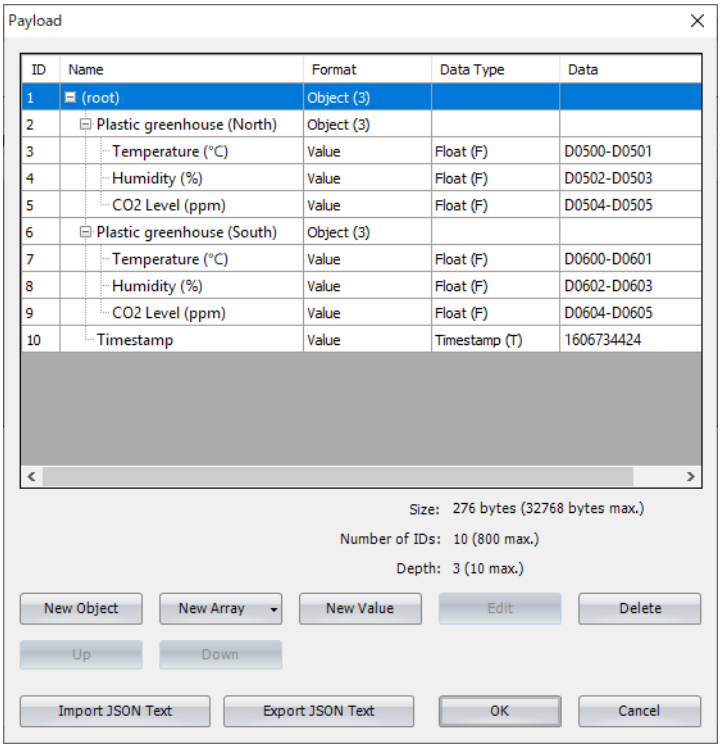
Item	Range
Temperature	15.0 to 25.0°C
Humidity	60.0 to 80.0%
CO_2 concentration	400.0 ppm or higher

Settings

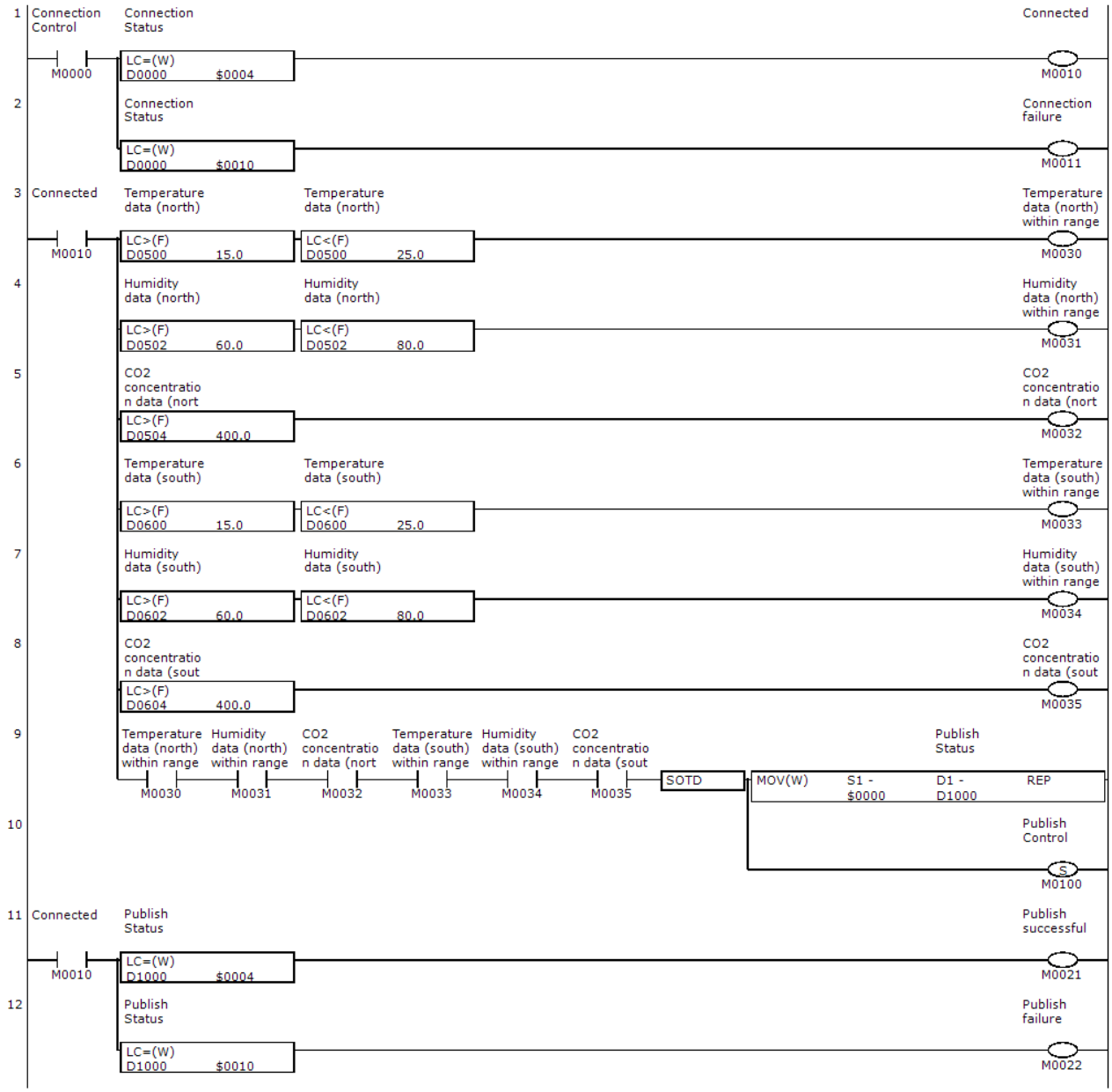
Configure the basic settings on the **MQTT Settings** tab in the **MQTT Settings** dialog box, and then configure the following items on the other tabs.

Tabs	Item	Description
MQTT Settings tab	Connection Control	M0000
	Connection Status	D0000
Publish tab	Topic	"plantation/house"
	Payload dialog box settings	Refer to the following screenshot
	Operation Mode	Trigger Execution
	Publish Control	M0100
	Publish Status	D1000
	QoS	0
	Retain	Disabled

Payload dialog box settings



Ladder Program



Device Address	Comment
M0000	Connection Control
M0010	Connected
M0011	Connection failure
M0021	Publish successful
M0022	Publish failure
M0030	Temperature data (north) within range
M0031	Humidity data (north) within range
M0032	CO2 concentration data (north) within range
M0033	Temperature data (south) within range
M0034	Humidity data (south) within range
M0035	CO2 concentration data (south) within range
M0100	Publish Control

Device Address	Comment
D0000	Connection Status
D0500	Temperature data (north)
D0502	Humidity data (north)
D0504	CO2 concentration data (north)
D0600	Temperature data (south)
D0602	Humidity data (south)
D0604	CO2 concentration data (south)
D1000	Publish Status

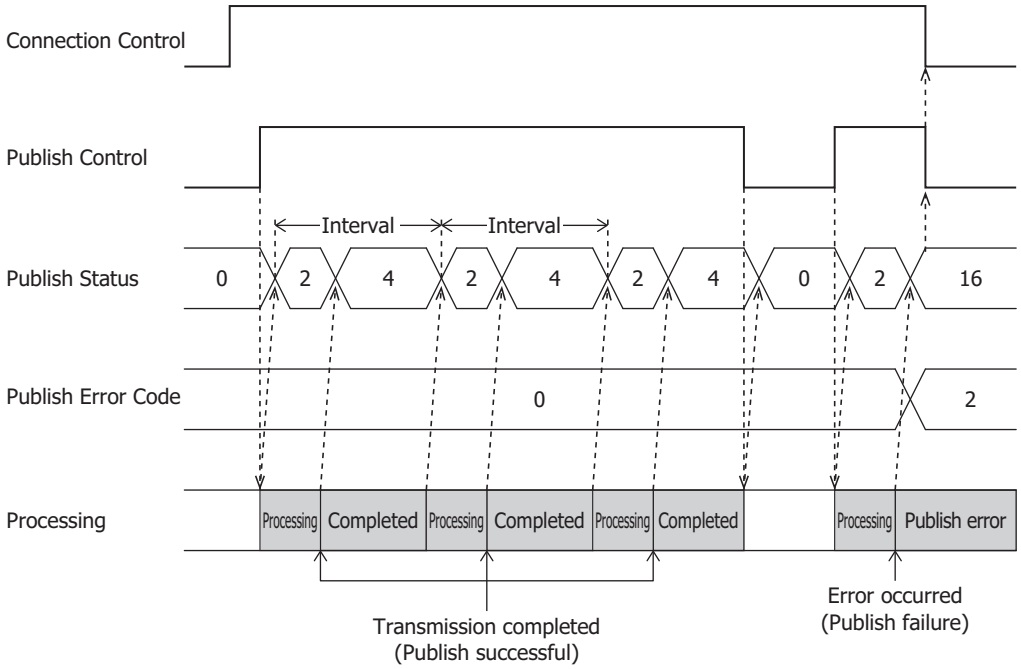
<table>
<tr><th>Ladder Line</th><th>Description</th></tr>
<tr><td>1</td><td>When M0000 is turned on, connecting to the broker is started. When M0010 is on, a connection was made to the broker.</td></tr>
<tr><td>2</td><td>When a connection cannot be made to the broker, M0011 is turned on.</td></tr>
<tr><td>3 to 8</td><td>When M0010 is on (a connection was made to the broker), if the temperature, humidity, and CO_2 concentration measured at two locations (north and south) in the greenhouse are within the setting range, M0030 to M0035 are turned on.</td></tr>
<tr><td>9 and 10</td><td>When M0010 is on (a connection was made to the broker), if either temperature, humidity, or CO_2 concentration is outside the setting range, the relevant internal relay (M0030 to M0035) changes from on to off. At this time, 0 (0000h) is stored in D1000 and Publish Status is cleared. Then M0100 is turned on and the data is published for the topic. (When publishing is complete, M0100 is automatically turned off.)</td></tr>
<tr><td>11</td><td>When publishing was successful, M0021 is turned on.
When the conditions are the same as those in the following table, the device values are: (D0500, D0501) = 18.5, (D0502, D0503) = 55.5, (D0504, D0505) = 410.1, (D0600, D0601) = 26.1, (D0602, D0603) = 64.5, and (D0604, D0605) = 420.2.
<table>
<tr><th>Item</th><th>Inside Greenhouse (North)</th><th>Inside Greenhouse (South)</th></tr>
<tr><td>Temperature</td><td>18.5°C</td><td>26.1°C</td></tr>
<tr><td>Humidity</td><td>55.5%</td><td>64.5%</td></tr>
<tr><td>CO_2 concentration</td><td>410.1 ppm</td><td>420.2 ppm</td></tr>
</table>
If the current time data for the internal clock (D8008 to D8014) is 9:05:46 on December 9, 2020 and the value of D8413 (Time Zone Offset) is 0, the JSON data to transmit is as follows.

{
"Plastic greenhouse (North)": {
"Temperature (°C)": 18.5,
"Humidity (%)": 55.5,
"CO2 Level (ppm)": 410.1
},
"Plastic greenhouse (South)": {
"Temperature (°C)": 26.1,
"Humidity (%)": 64.5,
"CO2 Level (ppm)": 420.2
},
"Timestamp": 1607504746
}
At this time, 224 is stored in D1002 (Transmitted data byte count).</td></tr>
<tr><td>12</td><td>When publishing fails, M0022 is turned on.</td></tr>
</table>

When the Operation Mode Is Set to Fixed Period

After the main unit connects to the broker and when the Publish Control relay is on, the data is published for the specified topic at the cycle set by **Interval** (2) and the status is stored in Publish Status. When the Publish Control relay is turned off, publishing is ended. At this time, status 0 (0000h) is stored in Publish Status.

If publishing fails, status 16 (0010h) is stored in Publish Status, the error code is stored in Publish Error Code, and the Publish Control relay is turned off. When there is an error other than error code 7 (0007h), the Connection Control relay is also turned off. When the error is error code 7 (0007h), the states of the Publish Control and Connection Control relays are kept.



When Publish Status is status 2 (0002h), do not change the values of devices set on the **Payload** dialog box. Change the values when Publish Status is a value other than status 2 (0002h).

Operation Example

Operation

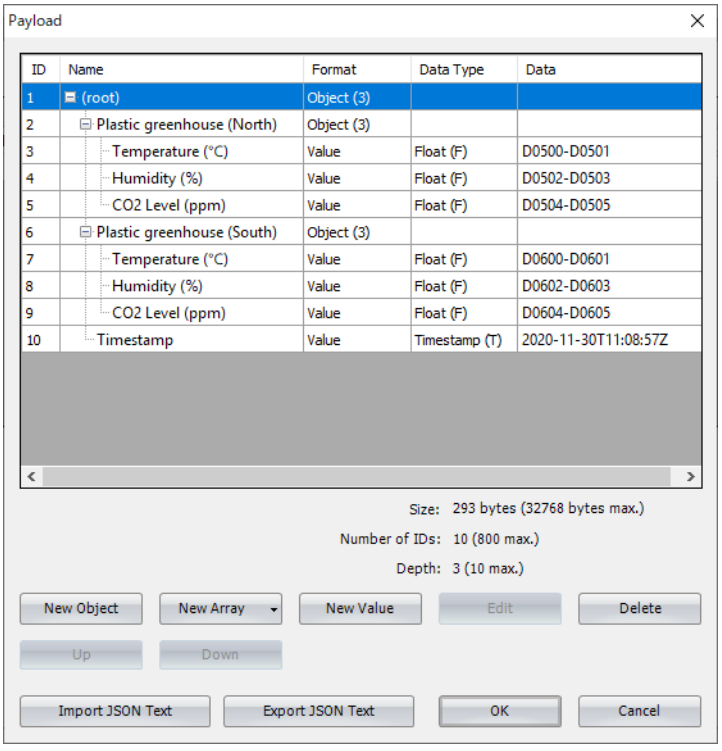
- Temperature, humidity, and CO_2 concentration are measured every 30 minutes at two locations in a greenhouse for strawberry cultivation and published to the topic "plantation/house".
- The status inside the greenhouse is checked from a remote location.

Settings

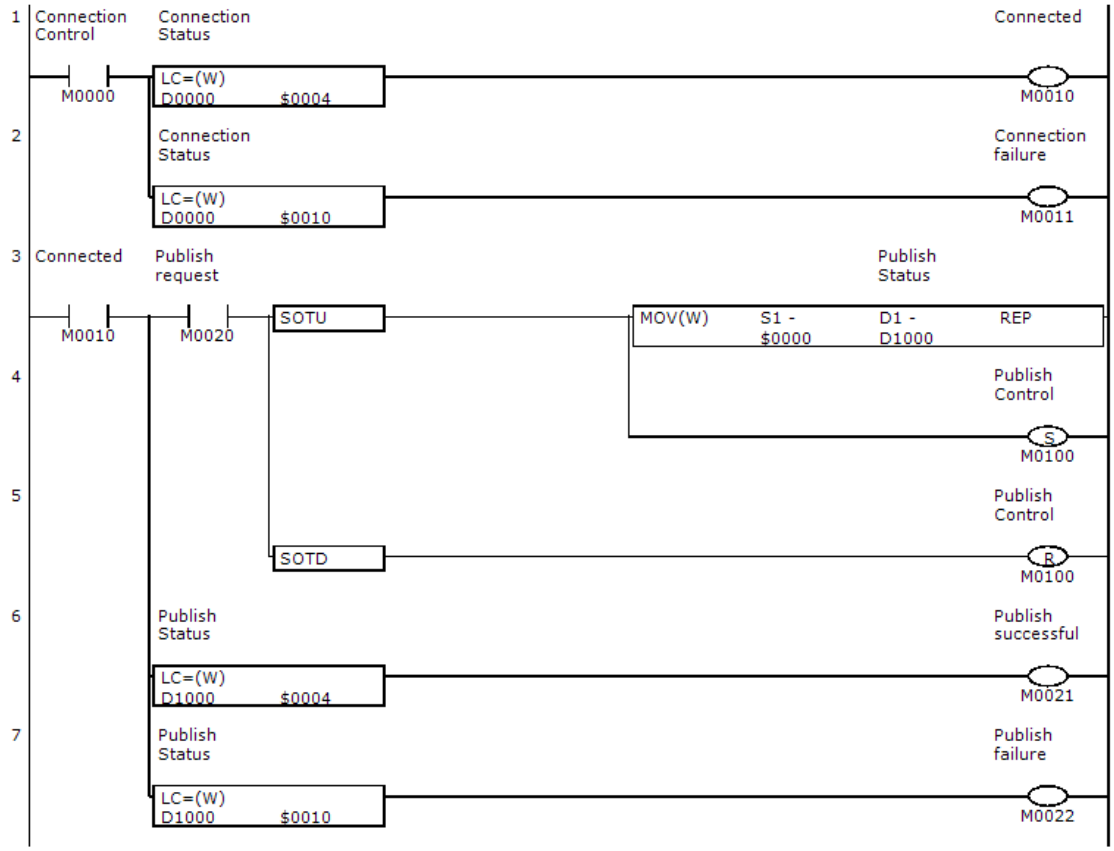
Configure the basic settings on the **MQTT Settings** tab in the **MQTT Settings** dialog box, and then configure the following items on the other tabs.

Tabs	Item	Description
MQTT Settings tab	Connection Control	M0000
	Connection Status	D0000
Publish tab	Topic	"plantation/house"
	Payload dialog box settings	Refer to the following screenshot
	Operation Mode	Operation mode: Fixed Period
		Interval: 1,800 s
	Publish Control	M0100
	Publish Status	D1000
	QoS	1
	Retain	Disabled

Payload dialog box settings



Ladder Program



Device Address	Comment
M0000	Connection Control
M0010	Connected
M0011	Connection failure
M0020	Publish request
M0021	Publish successful
M0022	Publish failure
M0100	Publish Control

Device Address	Comment
D0000	Connection Status
D0500	Temperature data (north)
D0502	Humidity data (north)
D0504	CO2 concentration data (north)
D0600	Temperature data (south)
D0602	Humidity data (north)
D0604	CO2 concentration data (north)
D1000	Publish Status

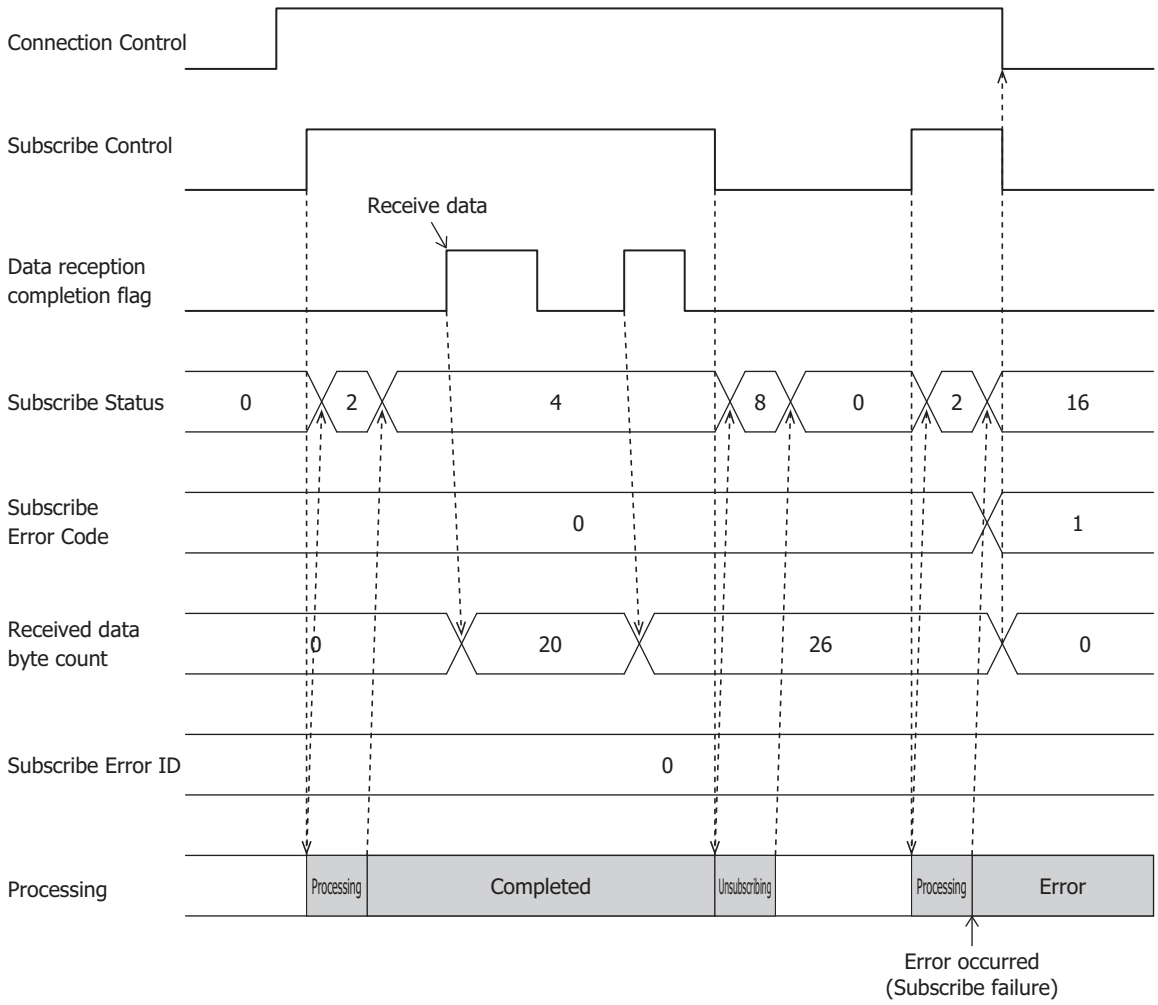
<table>
<tr><th>Ladder Line</th><th>Description</th></tr>
<tr><td>1</td><td>When M0000 is turned on, connecting to the broker is started. When M0010 is on, a connection was made to the broker.</td></tr>
<tr><td>2</td><td>When a connection cannot be made to the broker, M0011 is turned on.</td></tr>
<tr><td>3</td><td>When M0010 is on (a connection was made to the broker) and M0020 is changed from off to on, 0 (0000h) is stored in D1000 and Publish Status is cleared.</td></tr>
<tr><td>4</td><td>M0100 is turned on and the data is published for the topic every 1,800 s (30 m).</td></tr>
<tr><td>5</td><td>When M0020 is turned off, publishing is ended.</td></tr>
<tr><td>5</td><td>When publishing was successful, M0021 is turned on.
When the conditions are the same as those in the following table, the device values are: (D0500, D0501) = 20.5, (D0502, D0503) = 60.5, (D0504, D0505) = 410.1, (D0600, D0601) = 26.1, (D0602, D0603) = 64.5, and (D0604, D0605) = 420.2.
<table>
<tr><th>Item</th><th>Inside Greenhouse (North)</th><th>Inside Greenhouse (South)</th></tr>
<tr><td>Temperature</td><td>20.5°C</td><td>26.1°C</td></tr>
<tr><td>Humidity</td><td>60.5%</td><td>64.5%</td></tr>
<tr><td>CO_2 concentration</td><td>410.1 ppm</td><td>420.2 ppm</td></tr>
</table>
If the current time data for the internal clock (D8008 to D8014) is 10:42:08 on November 9, 2020 and the value of D8413 (time zone offset) is 0, the JSON data to transmit is as follows.
<pre>{
  "Plastic greenhouse (North)": {
    "Temperature (°C)": 18.5,
    "Humidity (%)": 55.5,
    "CO2 Level (ppm)": 410.1
  },
  "Plastic greenhouse (South)": {
    "Temperature (°C)": 26.1,
    "Humidity (%)": 64.5,
    "CO2 Level (ppm)": 420.2
  },
  "Timestamp": "2020-11-09T10:42:08Z"
}</pre>
At this time, 236 is stored in D1002 (Transmitted data byte count).</td></tr>
<tr><td>12</td><td>When publishing fails, M0022 is turned on.</td></tr>
</table>

Subscribing to a Topic

Subscribing and Unsubscribing

After the main unit connects to the broker and when the Subscribe Control relay is changed from off to on, the specified topic is subscribed to and the status is stored in Subscribe Status.

When the subscription is successful, status 4 (0004h) is stored in Subscribe Status. When data is published from the broker for the specified topic in this state, the data reception completion flag is turned on and the length of received data is stored in Received data byte count in bytes. For how the received data is processed, see "Storing the Contents of Received JSON Data in Devices" on page 3-105.



Operation Example

Operation

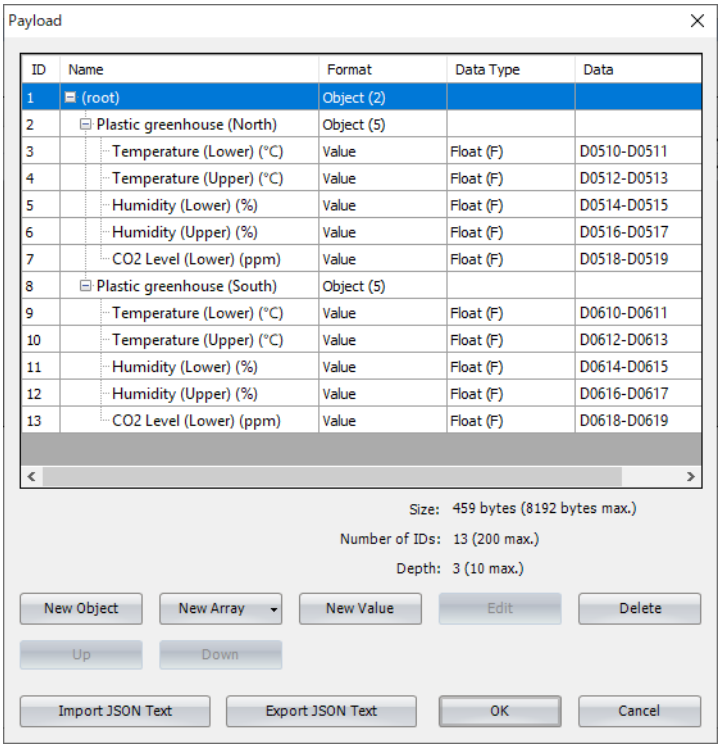
- Temperature, humidity, and CO_2 concentration in a greenhouse for strawberry cultivation are managed.
- An administrator publishes the upper/lower limit values of temperature and humidity and the lower limit value of CO_2 concentration in a greenhouse to the topic "plantation/house" from a remote location.
- The main unit subscribes to the topic "plantation/house" and operates with settings published by the administrator.

Settings

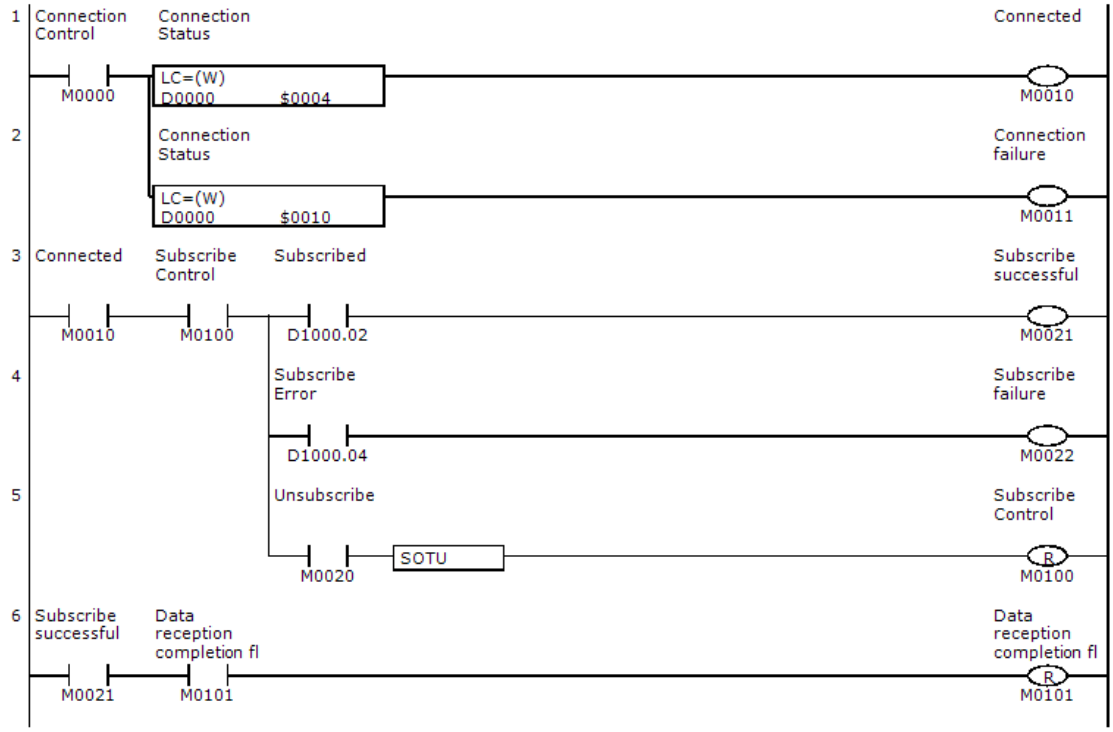
Configure the basic settings on the **MQTT Settings** tab in the **MQTT Settings** dialog box, and then configure the following items on the other tabs.

Tabs	Item	Description
MQTT Settings tab	Connection Control	M0000
	Connection Status	D0000
Subscribe tab	Topic	"plantation/house"
	Payload dialog box settings	Refer to the following screenshot
	Subscribe Control	M0100
	Subscribe Status	D1000
	QoS	1
	Auto ON	Enabled

Payload dialog box settings



Ladder Program



Device Address	Comment
M0000	Connection Control
M0010	Connected
M0011	Connection failure
M0020	Unsubscribe
M0021	Subscribe successful
M0022	Subscribe failure
M0100	Subscribe Control
M0101	Data reception completion flag

Device Address	Comment
D0000	Connection Status
D0510	Lower limit value of temperature data (north)
D0512	Upper limit value of temperature data (north)
D0514	Lower limit value of humidity data (north)
D0516	Upper limit value of humidity data (north)
D0518	Lower limit value of CO2 concentration data (north)
D0610	Lower limit value of temperature data (south)
D0612	Upper limit value of temperature data (south)
D0614	Lower limit value of humidity data (south)
D0616	Upper limit value of humidity data (south)
D0618	Lower limit value of CO2 concentration data (south)
D1000	Subscribe Status

Ladder Line	Description
1	When M0000 is turned on, connecting to the broker is started. When M0010 is on, a connection was made to the broker.
2	When a connection cannot be made to the broker, M0011 is turned on.
3	When M0010 is turned on (when connecting to the broker), M0100 is automatically turned on, and the topic is subscribed to. When subscribing was successful, M0021 is turned on.
4	When subscribing fails, M0022 is turned on.
5	When M0020 is turned on, the subscription is canceled.
6	When data is received, M0101 is turned off.

Storing the Contents of Received JSON Data in Devices

Received JSON data and IDs set on the **Payload** dialog box are searched in order from the beginning of the list.

If an element exists in the received JSON data that matches the depth level, name, and format of an ID set on the **Payload** dialog box and the value of that element is within the range of the data type for that ID, then that value is stored in the device.

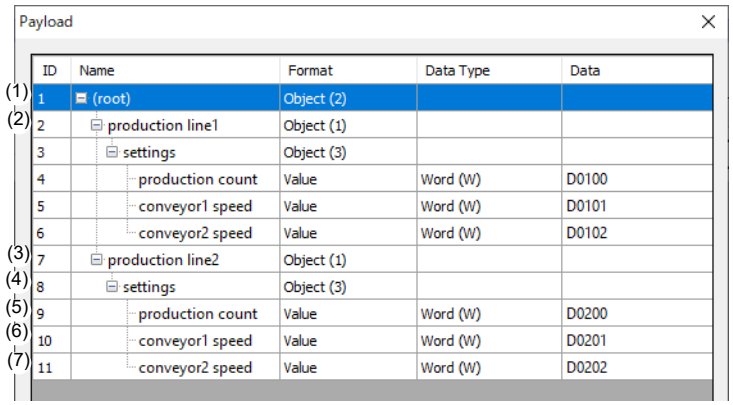
- If an element does not exist in the received JSON data that matches the depth level, name, and format of an ID set on the **Payload** dialog box, then the corresponding ID is stored in Subscribe Error ID.
- If an element exists in the received JSON data that matches the depth level, name, and format of an ID set on the **Payload** dialog box and the value of that element is outside the range of the data type for that ID, then error code 7 (0007h) is stored in Subscribe Error Code and the ID with the error is stored in Subscribe Error ID.

Example

- The main unit controls two production lines in a factory.
- When the main unit receives setting values, it performs processing according those setting values. There are multiple setting values, but only setting values with changes are actually received.

This example explains the processing when the **Payload** dialog box is set as shown in the following screenshot.

Payload Dialog Box



ID	Name	Format	Data Type	Data
1	(root)	Object (2)		
2	production line1	Object (1)		
3	settings	Object (3)		
4	production count	Value	Word (W)	D0100
5	conveyor1 speed	Value	Word (W)	D0101
6	conveyor2 speed	Value	Word (W)	D0102
7	production line2	Object (1)		
8	settings	Object (3)		
9	production count	Value	Word (W)	D0200
10	conveyor1 speed	Value	Word (W)	D0201
11	conveyor2 speed	Value	Word (W)	D0202

Received JSON Data

Line Number	Description
1	{
2	"production line2" :{
3	"settings":{
4	"conveyor1 speed":50
5	}
6	}
7	}

Processing Flow

(1) The first to seven lines of the received JSON data are enclosed in curly brackets. This matches the settings for ID 1 on the **Payload** dialog box, so ID 2 is searched for next.

(2) ID 2 does not exist in the received JSON data, so the search for ID 2 and its child elements is skipped and 2 (0002h) is stored in Subscribe Error ID.

(3) The second to sixth lines of the received JSON data are enclosed in curly brackets and set with the name "production line2". This matches the settings for ID 7, so ID 8 is searched for next.

(4) The third to fifth lines of the received JSON data are enclosed in curly brackets and set with the name "settings". This matches the settings for ID 8, so ID 9 is searched for next.

(5) ID 9 does not exist in the received JSON data, so it is skipped. ID 10 is searched for next.

(6) The name "conveyor1 speed" on the fourth line of the received JSON data matches the name of ID 10 and the value of this element (50) is within the range of word (W), so 50 is stored in D0201.

(7) ID 11 does not exist in the received JSON data, so it is skipped and 0 (0000h) is stored in Subscribe Error Code.

ABOUT THE WARRANTY OF THE PRODUCTS

1. Warranty Period

The Products are warranted for 3 years from the date of purchase, or from the date of delivery completion.

* Consumable/maintenance parts such as batteries and relays if the operation exceeds 100,000 times are excluded from the 3-year warranty.

2. Extent of Warranty

IDEC CORPORATION is responsible for failures or defects of the Products during the above warranty period, either a replacement part will be provided or the defective parts of the Products will be repaired free of charge. If such failure or defects should occur, please offer them to the distributor, dealer or IDEC CORPORATION with the materials in which the date of purchase is specified.

* The expenses for installation and construction at the time of repair will not be borne.

3. Indemnification

IDEC CORPORATION will not be liable under this Warranty and be indemnified and held harmless from any and all demands, suits, expenses, claims, damages and liabilities in the following event that:

1) The Products are used or operated beyond the conditions or environment range as described in catalog, specifications or instruction; or
2) The failure or defects of the Products arise from the cause other than the Products; or
3) The Products are improved, modified or altered by the party other than IDEC; or
4) The failure or defects and damages of the Products arise from the usage of the Product in the way that is not intended; or
5) The failure or defects and damages of the Products arise from the cause beyond IDEC's control including, but not limited to, fire, earthquake, flood, lightning, other natural disasters, and acts of God; or
6) The failure or defects and damages of the Products arise from the relocation, transportation or drop after you purchase the Products; or
7) The failure or defects and damages of the Products arise from improper installation; or
8) Maintenance and inspection are not carried out in accordance with instruction.

* Customers assume their own risk in programming products, Company will not be held liable for damages as a result of improper programming.

IDEC CORPORATION DISCLAIMS ALL IMPLIED WARRANTIES OF MERCHANTABILITY AND/OR FITNESS FOR A PARTICULAR USE OR PURPOSE, AS WELL AS LIABILITY FOR INCIDENTAL, SPECIAL, INDIRECT, CONSEQUENTIAL OR OTHER DAMAGES RELATING TO THE PRODUCTS

4. Extent of Service

The price of the Products will not include the fee for any service such as sending technicians and engineers, IDEC CORPORATION will charge you the fee for the following:

1) Instruction for installment and visiting for test operation, including, but not limited to creating application software and operation tests; and
2) Maintenance and inspection, arrangement and repair; and
3) Technical assistance and technical education; and
4) Product test and inspection based on you request.

IDEC CORPORATION

Head Office 6-64, Nishi-Miyahara-2-Chome, Yodogawa-ku, Osaka 532-0004, Japan

www.idec.com

USA	IDEC Corporation	**Singapore**	IDEC Izumi Asia Pte. Ltd.	**China**	IDEC (Shanghai) Corporation	**Japan**	IDEC Corporation
EMEA	APEM SAS	**Thailand**	IDEC Asia (Thailand) Co., Ltd.		IDEC Izumi (H.K.) Co., Ltd.		
		India	IDEC Controls India Private Ltd.	**Taiwan**	IDEC Taiwan Corporation		

B-2451(0)