

SLMP provider

Version 1.0.3

User's guide

October 29, 2021

【 remarks 】

- This document uses the machine translation.
- Some of the images used in this book are taken from the "SLMP Reference Manual".

【 revision history 】

Version	Date	Content
1.0.0	2018-02-01	First edition.
1.0.1	2019-09-04	Bug fixed at Param option.
1.0.2	2020-01-21	Bug fixed at AddController.
1.0.3	2020-04-07	Series option added. Processing correction at AddVariable. Exclusive control processing modification.
	2020-11-04	Added Appendix
	2021-10-29	Added a list of request destination unit I / O numbers and station numbers that can be specified. Corrected the notes of I1 and UI1 of VT item.

【 Operation check model 】

Model	Version	Notes
R02CPU		MELSEC iQ-R series
Tool	v1.1	SLMP Conformance test tool.

Contents

1. Introduction	6
2. Outline of provider.....	7
2.1. Outline	7
2.2. Directions point.....	7
2.2.1. About the connection with the equipment	7
2.3. Method property.....	8
2.3.1. CaoWorkspace::AddController method.....	8
2.3.1.1. Conn is optional.	9
2.3.1.2. Series Option	9
2.3.1.3. Sample program	10
2.3.2. CaoController::AddVariable method	11
2.3.2.1. Device is optional.	12
2.3.2.2. Address is optional.....	13
2.3.2.3. Param is optional.	16
2.3.2.4. VT is optional.	18
2.3.2.5. Elem is optional.	20
2.3.2.6. Array is optional.....	20
2.3.2.7. Random read/writing	21
2.3.2.8. Sample program	22
2.4. Error code.....	24
3. Command reference.....	25
3.1. Execute command.....	25
3.1.1. CaoController::Execute("Raw") command	25
3.1.2. Sample program	26
4. Appendix.....	29
4.1. Other station access	29
4.1.1. Setting the parent PLC	30
4.1.1.1. Setting of built-in Ethernet unit	30
4.1.1.2. Expansion Ethernet Unit 1 Setting.....	31
4.1.1.3. Expansion Ethernet Unit 2 setting	32
4.1.2. Setting of child PLC	33
4.1.2.1. Setting of built-in Ethernet unit	33

4.1.2.2. Expansion Ethernet Unit Settings	34
4.1.3. Connection parameter setting	35

1. Introduction

This book is a data communication function by other party equipment and SLMP(Seamless Message Protocol) ⁽¹⁾, and the user's guides of the CAO provider of the data of the equipment that writes reading. CAO provider (CaoProvSLMP.dll) that treats in this book is called SLMP provider.

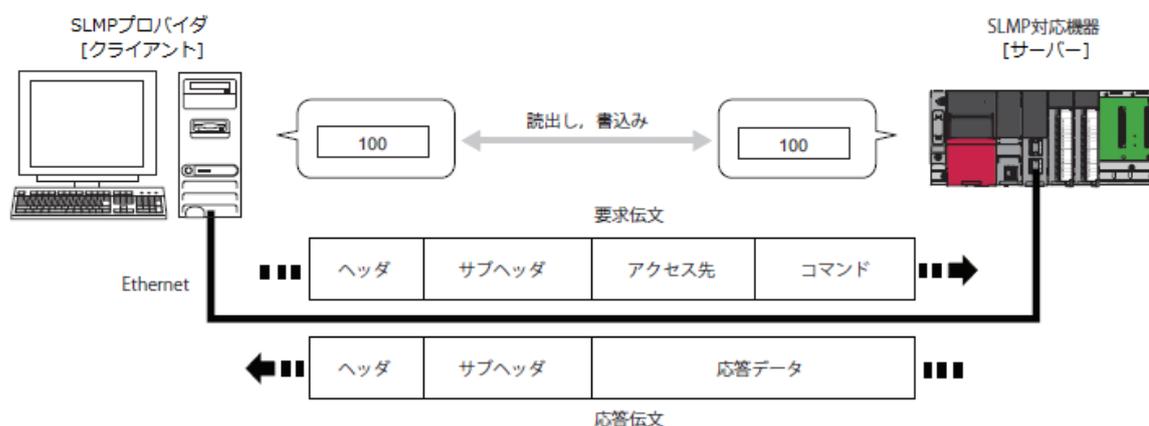
The SLMP provider uses Ethernet, and the protocol for the equipment and the external instrument (personal computer and indicator, etc.) to access the equipment for SLMP communicates.

Any device that can send and receive messages using the SLMP control procedure can communicate using SLMP.

The SLMP provider functions to the equipment (server) for SLMP as a client.

The SLMP provider uses ST type and binary communication.

For details of communication and setting method, refer to "SLMP Reference Manual" and the manual of each unit.



Please refer to chapter 2 for details of the outline and the variable of the SLMP provider.

¹ Seamless Message Protocol(SLMP) is a protocol for the seamless communication that the CC-Link society advocates.

2. Outline of provider

2.1. Outline

The SLMP provider is CAO provider that absorbs the part that depends on the unit to communicate with the equipment for SLMP and offers the function provided for by the CAO provider and the interface specification. The file format is DLL(Dynamic Link Library), and when using it from the CAO engine, it is dynamically loaded. When the SLMP provider is used, it is necessary to install ORiN2SDK or to register the registry by the hand work referring to the table below.

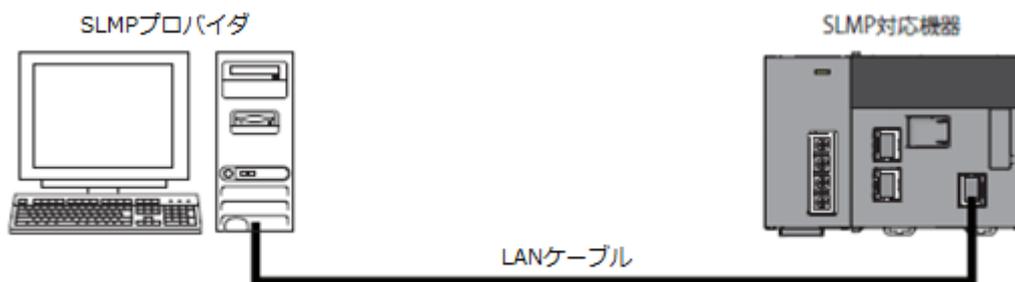
Table 2-1 SLMP provider

File name	CaoProvSLMP.dll
ProgID	CaoProv.SLMP
Registry registration	regsvr32 CaoProvSLMP.dll
Blotting out of registry registration	regsvr32 /u CaoProvSLMP.dll

2.2. Directions point

2.2.1. About the connection with the equipment

Refer to the user's manual of the unit used for the connection of PC and the equipment.



2.3. Method property

2.3.1. CaoWorkspace::AddController method

Process initial of Ethernet (TCP, UDP) communication in the SLMP provider when you generate the Controller object. Specify the device by an optional character string when you connect it.

Format AddController (< bstrCtrlName:BSTR > and < bstrProvName:BSTR >
<bstrPcName:BSTR >,<bstrOption:BSTR>)

bstrCtrlName : In controller name
 bstrProvName : In provider name. fixed value ="CaoProv.SLMP"
 bstrPcName : Execution machine name of in provider
 bstrOption : In optional character string

The list specified for an optional character string is shown as follows.

Table 2-2 Optional character string of CaoWorkspace::AddController

Option	Explanation
Conn=< connected parameter >	Indispensability. Set a communication form and the connected parameter. (See 2.3.1.1)
Series[=<communication type>]	Specify the type of communication. (See 2.3.1.2)
Timeout=< timeout period >	Set the timeout period when it sends and receives it by the millisecond. (default: 3000)
Retry=< Number of retries >	Set the number of communication retries for sending and receiving. (default: 0)

2.3.1.1. Conn is optional.

Connected parameter character string of optional Conn is shown as follows.

"Conn=ETH:<Dest IP Address>:<Dest Port No>"

"Conn=TCP:<Dest IP Address>:<Dest Port No>"

"Conn=UDP:<Dest IP Address>:<Dest Port No>"

< Dest IP Address > : Internet Protocol address of connection destination.

< Dest Port No > : Port number of connection destination.

2.3.1.2. Series Option

Specify the type of communication.

See Table 2-3 for the settings.If the Series option is omitted, the default is 0.

Table 2-3 Series option setting list

value	detail	remarks
0	16-bit long address (compatible with Q/L series)	<ul style="list-style-type: none"> • Access with a 16-bit length address. If you are using MELSEC Q/L series, please specify this. • You can access up to 6 hexadecimal digits.
1	32-bit length address	<ul style="list-style-type: none"> • If you are using a MELSEC R series device that supports 32-bit address, you can access up to 8 hexadecimal digits by specifying this address.

(Example) "Conn=ETH:192.168.0.1:1025, Series=0" MELSEC Q/L compatible mode (16-bit length address)

(Example) "Conn=ETH:192.168.0.1:1025, Series=1" MELSEC R series (32-bit length address)

2.3.1.3. Sample program

The sample program of AddController is shown below.

```
HRESULT hr = S_OK;
ICaoEngine* pEng = NULL;
ICaoWorkspaces *pWss = NULL;
ICaoWorkspace *pWs = NULL;
ICaoController *pCtrl = NULL;

// Generation of CaoEngine
hr = CoCreateInstance (CLSID_CaoEngine,
NULL,
CLSCTX_LOCAL_SERVER,
IID_ICaoEngine,
(void **)&pEng);
if (FAILED(hr)) {
goto EndProc;
}
// Acquisition of CaoWorkspace collection
hr = pEng->get_Workspaces (&pWss);
if (FAILED(hr)) {
goto EndProc;
}
// Acquisition of CaoWorkspace
hr = pWss->Item(CComVariant(0L), &pWs);
if (FAILED(hr)) {
goto EndProc;
}

// Generation of CaoController
hr = pWs->AddController(CComBSTR(L"SLMP_TEST"),
CComBSTR(L"CaoProv.SLMP"),
CComBSTR(L""),
CComBSTR(L"Conn=ETH:169.254.225.206:61442"),
&pCtrl);
if (FAILED(hr)) {
goto EndProc;
}

// Put processing necessary for here.
// The value is set, and acquired.

EndProc:
if (pCtrl) pCtrl->Release();
if (pWs) pWs->Release();
if (pWss) pWss->Release();
if (pEng) pEng->Release();
```

2.3.2. CaoController::AddVariable method

Make the variable object that acquisition/sets equipment information.

Format AddVariable(<bstrName:BSTR >,<bstrOption:BSTR>)

bstrName : In variable identifier. Arbitrary, unique character string to manage.

bstrOption : In optional character string

Table 2-4 Optional character string of CaoController::AddVariable

Option	Meaning
Device=< Device code >	Indispensability. Specify the device code of the device at the access destination. (See 2.3.2.1)
Address=< initial address >	Indispensability. Specify the initial address of the device at the access destination. Specify the address by the decimal number and the hexadecimal number value. (Please refer to table 2-4 for details.) Two or more addresses can be specified. (Please refer to chapter 2.3.2.7 for details.)
Param=< parameter of Variable>	Set the parameter of the variable. (See 2.3.2.3)
VT=< Variable type >	Specify the type of the data used when inputting and outputting to the device memory. (See 2.3.2.4)
Elem=< Number of elements >	Specify the number of variables. Input it by the following formats when specifying it by the hexadecimal number. 0x[0-9,A-F]+, &H[0-9,A-F]+, [0-9,A-F]+H (See 2.3.2.5)
Array=< True or False >	Specify whether to acquire the value in the form of the array at the time of reading one element. (See 2.3.2.6)

2.3.2.1. Device is optional.

It can access the equipment by specifying the device code for optional Device.

Bit device : SM,X,Y,M,L,F,V,B,TS,TC,LTS,LTC,SS,SC,LSTS,LSTC,
CS,CC,LCS,LCC,SB,DX,DY

Word device : SD,D,W,TN,SN,CN,SW,Z,R,ZR,RD

Double word device: LTN,LSTN,LCN,LZ

2.3.2.2. Address is optional.

Specify the initial address of the device that is read or written.

Specify the address by the decimal number or the hexadecimal number according to the kind of the device.

The maximum value of the address that can be specified depends on the setting value of the Series option.

Table 2-5 Device list

Device		Device code	Type	Method of addressing
Special relay		SM	Bit	Decimal number
Special register		SD	Word	Decimal number
Input		X	Bit	Hexadecimal number
Output		Y		Hexadecimal number
Internal relay		M		Decimal number
Latch relay		L		Decimal number
Annunciator		F		Decimal number
Edge relay		V		Decimal number
Link relay		B		Hexadecimal number
Data register		D		Word
Link register		W	Hexadecimal number	
Timer	Contact	TS	Bit	Decimal number
	Coil	TC		
	Current value	TN	Word	
Long timer	Contact	LTS	Bit	Decimal number
	Coil	LTC		
	Current value	LTN	Double word	
Multiplication timer	Contact	SS	Bit	Decimal number
	Coil	SC		
	Current value	SN	Word	
Long multiplication	Contact	LSTS	Bit	Decimal number
	Coil	LSTC		

timer	Current value	LSTN	Double word	
Counter	Contact	CS	Bit	Decimal number
	Coil	CC		
	Current value	CN	Word	
Long counter	Contact	LCS	Bit	Decimal number
	Coil	LCC		
	Current value	LCN	Double word	
Linking special relay		SB	Bit	Hexadecimal number
Special register of link		SW	Word	Hexadecimal number
Direct access input		DX	Bit	Hexadecimal number
Direct access output				Hexadecimal number
Index register		Z	Word	Decimal number
Long index register		LZ	Double word	Decimal number
File register		R	Word	Decimal number
		ZR		Decimal number
Register for unit refreshing		RD	Word	Decimal number

(example)

X0 : "Device=X, Address=0"
 Y1F : "Device=Y, Address=1F"
 D50 : "Device=D, Address=50"
 M350 : "Device=M, Address=350"

The data structure of the long timer and long integration timer of the programmable controller CPU manufactured by Mitsubishi Electric Corp. is different from that of ordinary devices, and is as shown in Figure 1.

Since the long timer and long integration timer have a data structure of 4 words per device, specify the number of devices in units of 4 points.

■ロングタイマ、ロング積算タイマデバイス読み出し時の注意事項

ロングタイマ、ロング積算タイマの現在値を4ワード単位で指定することで、接点およびコイルも同時に読み出されます。データ構成は下記となります。

応答データ	内容
1ワード目	現在値が格納されます。
2ワード目	
3ワード目	b0: コイルの値が格納されます。 b1: 接点の値が格納されます。 b2~b15: システムで使用
4ワード目	システムで使用

ロングタイマ、ロング積算タイマは、上記のようにデバイス1点あたり4ワードのデータ構成となるため、要求データのデバイス点数は、4点単位で指定してください。

例

ロングタイマ2点(LT0, LT1)を読み出す場合、先頭デバイスはLTN0、デバイス点数は8点を指定します。

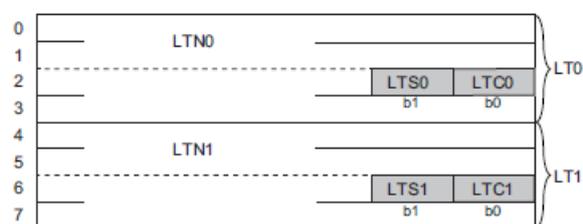


Figure 1 Data structure of long timer and long integration timer

- (例) LTS : “Device=LTN, Address=0, Elem=4”
The value of contact (LTS0) is stored in b1 of the 3rd word.
- LTC : “Device=LTN, Address=0, Elem=4”
The value of the coil (LTC0) is stored in b2 of the 3rd word.
- LTN : “Device=LTN Address=0, Elem=4”
The current value (LTN0) is stored in the 1st and 2nd words.
- LTN : “Device=LTN Address=0, Elem=8”
The current value (LTN0) is stored in the 1st and 2nd words, and the current value (LTN1) is stored in the 4th and 5th words.

2.3.2.3. Param is optional.

Connected parameter character string of optional Param is shown as follows. The parameter in the square bracket) shows a possible omission here.

For each element, it is necessary to set the set value by the specified number of digits in hexadecimal. Fill in the missing digits with zeros.

Please refer to the SLMP reference manual for the contents of each item.

"[Param=

[< Destination network No >:< Destination station No >:< Destination module U/O No >:< Destination multidrop station NO>[:< Monitoring timer >]]

- | | | |
|-------------------------------------|---|----------------------------------------------------------------------------|
| < Destination network No > | : | Network number of access destination(2digit).
Default is 00H. |
| < Destination station No > | : | Station number of access destination (4digit) .
Default is FFH. |
| < Destination module I/O No > | : | Module I/O number of access destination(4digit).
Default is 03FFH. |
| < Destination multidrop station NO> | : | Multidrop station number of access destination(2digit).
Default is 00H. |
| < Monitoring timer > | : | Waiting timer(4digit).
Default is 0000H(unlimited wait). |

(example) "Device=X, Address=0, Param=00:FF:03FF:00:0000" Acquire the X0 value.

Table 2-6 List of request destinations module I/O No that can be specified

	Access station of the other device	Request destination module I/O No specified by the other device
1	Control CPU	03D0 _H
2	Standby CPU	03D1 _H
3	A system CPU	03D2 _H
4	B system CPU	03D3 _H
5	Multi-CPU system No. 1	03E0 _H
6	Multi-CPU system No. 2	03E1 _H
7	Multi-CPU system No. 3	03E2 _H
8	Multi-CPU system No. 4	03E3 _H
9	Management CPU	03FF _H
10	Station C24 management CPU on multi-drop connection (Specify 1 to 9 above when accessing the CC-Link IE controller network, CC-Link IE field network, MELSECNET / H, MELSECNET / 10 connection station sequencer that goes through last)	0000 _H ~01FF _H

Table 2-7 List of request destination unit station numbers that can be specified

	Access station of the other device	Request destination unit station number specified by the other device
1	Stations other than the following	00 _H (0)
2	Stations on multi-drop connections (When accessing the CC-Link IE controller network, CC-Link IE field network, MELSECNET / H, and Q / L / QnA CPU of the MELSECNET / 10 connection station via the last, refer to 1 above)	00 _H ~1F _H (0~31)

2.3.2.4. VT is optional.

Specify the point a read and written data type and element (One point = 1 bit).

When optional VT is omitted, the value of default is set with the device specified for optional Device. When "I2" specifies the double word device when "BIT" specifies the word device when the bit device is specified for optional Device, "I4" reaches the value of default.

Table 2-8 Optional VT that can be specified list

VT Option	Data type	Point/number of elements	Meaning ⁽²⁾
BIT	VT_I2	One point	It reads and writes it by the unit of the bit (Every one point). Note) It is possible to specify it only for the bit device (X, Y, and M etc.).
BOOL	VT_BOOL	One point	It reads and writes it by the unit of the bit (Every one point). Note) It is possible to specify it only for the bit device (X, Y, and M etc.).
I1	VT_I1	Eight points	It reads and writes it in eight points. Note) When writing is performed by specifying an odd number of elements with the Elements option or Elem option, it is treated as an even number of elements, and the added 8 points are filled with 0 and written. example) Variable name: X0 Option: VT = I1, Elem = 1 When assigning a value, all X8 to XF will be 0.
UI1	VT_UI1	Eight points	It reads and writes it in eight points. Note) When writing is performed by specifying an odd number of elements with the Elements option or Elem option, it is treated as an even number of elements, and the added 8 points are filled with 0 and written. example) Variable name: X0 Option: VT = UI1, Elem = 1 When assigning a value, all X8 to XF will be 0.
I2	VT_I2	16 points	It reads and writes it by the unit of the word (Every 16 points).

² Read and write in word units except VT=BIT, VT=BOOL, and VT=BSTR.

UI2	VT_UI2	16 points	It reads and writes it in 16 points.
I4	VT_I4	32 points	It reads and writes it in 32 points.
UI4	VT_UI4	32 points	It reads and writes it in 32 points.
R4	VT_R4	32 points	It reads and writes it in 32 points.
R8	VT_R8	64 points	It reads and writes it in 64 points.
BSTR	VT_BSTR	Eight points	The character string of ASCII (One character: 8 bit) is read and written. Note) When a character string that is shorter than the number of elements specified ..optional Elem it.. is written, 0 buries the point of the remainder.

< VT optional character string VT = "> "

(example) "Device=X, Address=0, VT=I2" The value of X15 is read and written from X0 as word value (2Byte).

2.3.2.5. Elem is optional.

Specify the number of elements by the decimal number or the hexadecimal number. Specify the numerical value as it is when specifying it by the decimal number. Specify &H 0-9 and A-F + or 0-9 and A-F in the form of + H when specifying it by the hexadecimal number 0x 0-9 and A-F +. The value of default when optional Elem is omitted becomes one ³.

< number of elements Elem = "> "

- (example) "Device=X, Address=0, Elem=5" Acquire the value of X4 from X0 as a bit value.
- (example) "Device=D, Address=10, Elem=0x10" Acquire the value of D25 from D10 as a word value.
- (example) "Device=D, Address=10, Elem=&H10" Acquire the value of D25 from D10 as a word value.
- (example) "Device=D, Address=10, Elem=10H" Acquire the value of D25 from D10 as a word value.
- (example) "Device=M, Address=100, VT=I2, Elem=2" Acquire the value of M131 from M100 wording.

※ Refer to the manual of the equipment at the access destination for the range where the number of elements can be specified.

2.3.2.6. Array is optional.

The value read excluding BSTR when specifying it is specified and whether it acquires it is specified by one specification ..optional Elem it.. in the form of the array optional VT moreover it. When True is specified, it becomes the form of the data type specified when False is specified in the form of the array. If the Array option is omitted, the default value is False.

"[Array=[< True or False >]]"

- (example) "Device=X, Address=0, VT=BOOL, Elem=1, Array=True" Acquire the value of X0 as an array of the BOOL type.
- (example) "Device=X, Address=0, VT=BOOL, Elem=1, Array=False" Acquire the value of X0 as BOOL type.

³ Refer to the manual of the equipment at the access destination for the range where the number of elements can be specified.

2.3.2.7. Random read/writing

It can access an address not consecutive by specifying two or more addresses⁴ for optional Address by colon (:). At this time, two or more only one kind of optional Device cannot be specified. Other option (VT, Elem, Array) is disregarded.

The unit of one word and the double word device must access the word device the bit device by every two words every 16 points.

Table 2-9 Data type of random access

Device type	Reading/writing data type	Remarks
Bit	VT_I2 VT_ARRAY	<ul style="list-style-type: none"> • The data type is an initial value of the device. • Specified a few minutes of the address are read by the array and/is written optional Address it.
Word	VT_I2 VT_ARRAY	
Double word	VT_I4 VT_ARRAY	

(example) "Device=M, Address=100:200"

Access M100-M115 and M200-M215.

(example) "Device=D, Address=0:2:4"

Access D0, D2, and D4.

(example) "Device=LTN, Address=1:3:5:7:9"

Access LTN1, LTN3, LTN5, LTN7, and LTN9.

⁴ Refer to the manual of the equipment at the access destination for the range of the address number of each device and the number of addresses (access point) that can be specified.

2.3.2.8. Sample program

The sample program of AddVariable is shown below.

The example) Set the value to bit device M array (number 10 of elements).

```
HRESULT hr = S_OK;
ICaoEngine* pEng = NULL;
ICaoWorkspaces *pWss = NULL;
ICaoWorkspace *pWs = NULL;
ICaoController *pCtrl = NULL;
ICaoVariable *pVar = NULL;
CComVariant vntGet;

// Generation of CaoEngine
hr = CoCreateInstance(CLSID_CaoEngine,
NULL,
CLSCTX_LOCAL_SERVER,
IID_ICaoEngine,
(void **)&pEng);
if (FAILED(hr)) {
goto EndProc;
}

// Acquisition of CaoWorkspace collection
hr = pEng->get_Workspaces(&pWss);
if (FAILED(hr)) {
goto EndProc;
}

// Acquisition of CaoWorkspace
hr = pWss->Item(CComVariant(0L), &pWs);
if (FAILED(hr)) {
goto EndProc;
}

// Generation of CaoController
hr = pWs->AddController(CComBSTR(L"SLMP_TEST"),
CComBSTR(L"CaoProv.SLMP"),
CComBSTR(L""),
CComBSTR(L"Conn=ETH:169.254.225.206:61442"),
&pCtrl);
if (FAILED(hr)) {
goto EndProc;
}

// Generation of a variable
hr = pCtrl->AddVariable(CComBSTR(L"aaa"), CComBSTR(L"Device=M, Address=16, Elem=10"), &pVar);
if (FAILED(hr)) {
goto EndProc;
}
```

```
// Setting and acquisition of value
CComVariant vntPut
vntPut.vt = (VT_I2 | VT_ARRAY);
vntPut.parray = SafeArrayCreateVector(VT_I2, 0, 10);
pVar->put_Value(vntPut);
pVar->get_Value(&vntGet);

EndProc:
if (pVar) pVar->Release();
if (pCtrl) pCtrl->Release();
if (pWs) pWs->Release();
if (pWss) pWss->Release();
if (pEng) pEng->Release();
```

2.4. Error code

In the SLMP provider, the following and peculiar the error code is defined. Moreover, "of the ORiN2 commonness error. ORiN2 Programming guide Refer to the chapter of the error code of".

Table 2-10 Peculiar error code

Error name	Error number	Explanation
Unit error	0x8010xxxx	Put the error code of the equipment in the part of xxxx and return it when the error occurs by the equipment. Refer to the reference of the equipment for the content of the error code.
Abnormal response (Format error)	0x80110000	Returns when the response packet has an unexpected abnormal format. This error is also returned when the communication data code is different.
Invalid device name	0x80110001	It occurs when a device name that does not exist is specified.
Address format error	0x80110002	Returned if the address is not specified correctly. It occurs when hexadecimal is specified for the address of the decimal device.
Out of address range	0x80110003	It occurs when an address that cannot specify the start address is specified. In some cases, it can be avoided by using the Series option (2.3.1.2).
Bit-wise access unauthorized	0x80110004	It occurs when bit unit access is specified in the word device or double word device. Check the device and VT option.

3. Command reference

This chapter explains each command of the CaoController::Execute method.

Table 3-1 CaoController::Execute command list

Command	Function	Content
Raw	Raw data sending and receiving	Send and receive raw data (binary code).

3.1. Execute command

3.1.1. CaoController::Execute("Raw") command

Send and receive raw data.

Format Raw (<vntParam>, <pVal>)

<vntParam> : [in] Request message (VT_UI1 | VT_ARRAY)
 <pVal> : [Out] response message (VT_UI1 | VT_ARRAY)
 Return value : HRESULT

S_OK	Normal termination
Excluding S_OK	Abnormal termination

Remarks : The request message (binary code) specified by the argument is sent to the device. Returns the response message in binary code.
 At this time, the contents of the request message / response message are not processed at all during transmission and reception.

3.1.2. Sample program

The sample program of Execute command (Raw) is shown below.

Read the example)① Y100.

②Write value (ON) in X110.

```
HRESULT func1()
{
    HRESULT hr = S_OK;
    ICaoEngine* pEng = NULL;
    ICaoWorkspaces *pWss = NULL;
    ICaoWorkspace *pWs = NULL;
    ICaoController *pCtrl = NULL;
    ICaoVariable *pVar = NULL;
    CComVariant vntParam;
    CComVariant vntRet;
    SAFEARRAYBOUND boundR = { 21, 0 };
    BYTE bSendRData[21] = {0x50, 0x00, 0x00, 0xFF, 0xFF, 0x03, 0x00, 0x0C, 0x00, 0x00,
    0x00, 0x01, 0x04, 0x01, 0x00, 0x00, 0x01, 0x00, 0x9D, 0x01,
    0x00};
    SAFEARRAYBOUND boundW = { 22, 0 };
    BYTE bSendWData[22] = {0x50, 0x00, 0x00, 0xFF, 0xFF, 0x03, 0x00, 0x0D, 0x00, 0x00,
    0x00, 0x01, 0x14, 0x01, 0x00, 0x10, 0x01, 0x00, 0x9C, 0x01,
    0x00, 0x10};

    // Generation of CaoEngine
    hr = CoCreateInstance(CLSID_CaoEngine,
    NULL,
    CLSCTX_LOCAL_SERVER,
    IID_ICaoEngine,
    (void **)&pEng);
    if (FAILED(hr)) {
        goto EndProc;
    }

    // Acquisition of CaoWorkspace collection
    hr = pEng->get_Workspaces(&pWss);
    if (FAILED(hr)) {
        goto EndProc;
    }

    // Acquisition of CaoWorkspace
    hr = pWss->Item(CComVariant(0L), &pWs);
    if (FAILED(hr)) {
        goto EndProc;
    }
}
```

```
// Generation of CaoController
hr = pWs->AddController(CComBSTR(L"SLMP_TEST"),
CComBSTR(L"CaoProv.SLMP"),
CComBSTR(L""),
CComBSTR(L"Conn=ETH:169.254.225.206:61442"),
&pCtrl);
if (FAILED(hr)) {
goto EndProc;
}

////////////////////////////////////
// Read ① Y100.
////////////////////////////////////
SAFEARRAY* pSa = SafeArrayCreate(VT_UI1, 1, &boundR);
// Access SAFEARRAY.
BYTE* pbyRead;

SafeArrayAccessData(pSa, (void**)&pbyRead);
for(int i = 0; i < 21; i++){
pbyRead[i] = bSendRData[i];
}

SafeArrayUnaccessData(pSa);

vntParam.Clear();
vntParam.parray = pSa;
vntParam.vt = VT_ARRAY | VT_UI1;

vntRet.Clear();
hr = pCtrl->Execute(CComBSTR("Raw"), vntParam, &vntRet);
if (FAILED(hr)) {
goto EndProc;
}

////////////////////////////////////
// Write value (ON) in ② X110.
////////////////////////////////////
// Make it because the substance of SAFEARRAY doesn't exist.
pSa = SafeArrayCreate(VT_UI1, 1, &boundW);
// Access SAFEARRAY.
BYTE* pbyWrite;

SafeArrayAccessData(pSa, (void**)&pbyWrite);
for(int i = 0; i < 22; i++){
pbyWrite[i] = bSendWData[i];
}
SafeArrayUnaccessData(pSa);
```

```
vntParam.Clear();
vntParam.parray = pSa;
vntParam.vt = VT_ARRAY | VT_UI1;

vntRet.Clear();
hr = pCtrl->Execute(CComBSTR("Raw"), vntParam, &vntRet);
if (FAILED(hr)) {
    goto EndProc;
}
std::cout << "Raw(Write) = 0x" << std::hex << std::setw(8) << std::setfill('0') << hr << "¥n";

EndProc:

// Liberating of object
if (pVar) pVar->Release();
if (pCtrl) pCtrl->Release();
if (pWs) pWs->Release();
if (pWss) pWss->Release();
if (pEng) pEng->Release();
return hr;
}
```

4. Appendix

4.1. Other station access

The Mitsubishi Electric Q series programmable controller can access other stations (acquire/write slave PLC values via the master PLC). This section explains the configuration of Figure 4-1 as an example of the setting procedure for accessing other stations. It is assumed that the parent/child PLC-PLCs are connected by a general-purpose Ethernet. Be sure to use Ethernet unit when connecting. Cannot be accessed through the built-in Ethernet unit. Use a hub or crossover cable to connect Ethernet units.

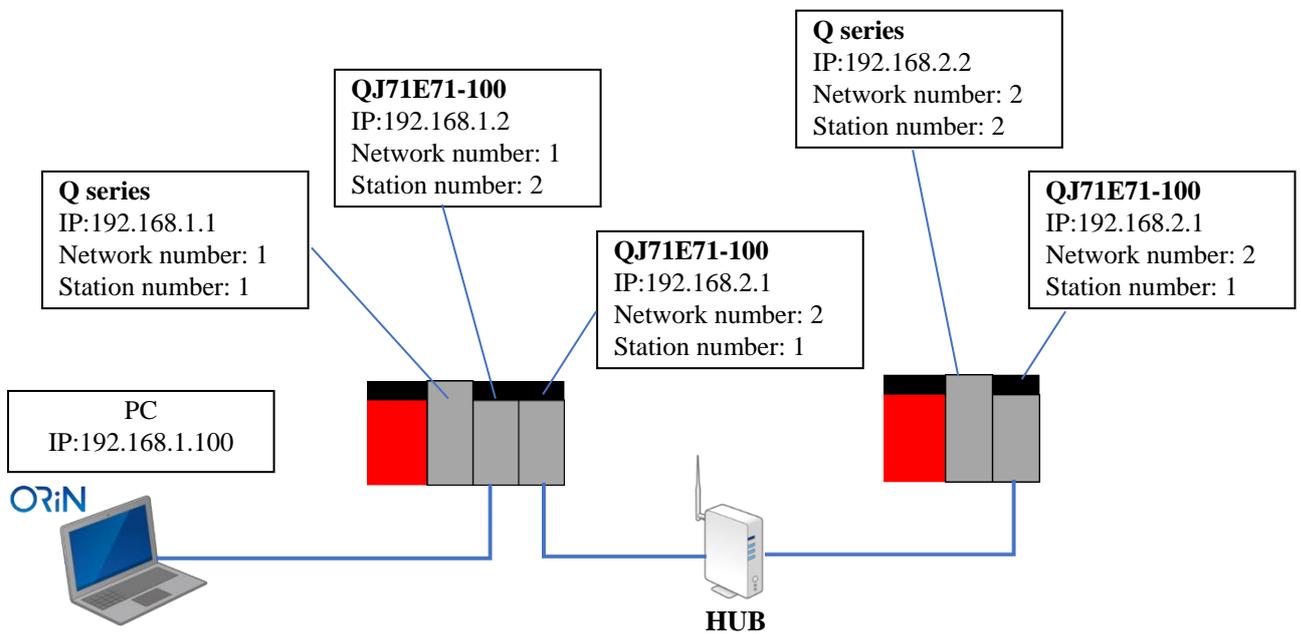


Figure 4-1 Configuration Diagram

4.1.1. Setting the parent PLC

4.1.1.1. Setting of built-in Ethernet unit

① to ④ in Figure 4-2 show the setting steps of the built-in Ethernet module.

- ① After connecting the PLC, click "PLC Parameter" on the navigation tree menu.
- ② Enter the IP address and subnet mask on the Q Parameter Setting screen.
- ③ Open MELSOFT Connection Extended Setting.
- ④ Select "MELSOFT Connection Extended Setting: Use" and enter the network number and station number.

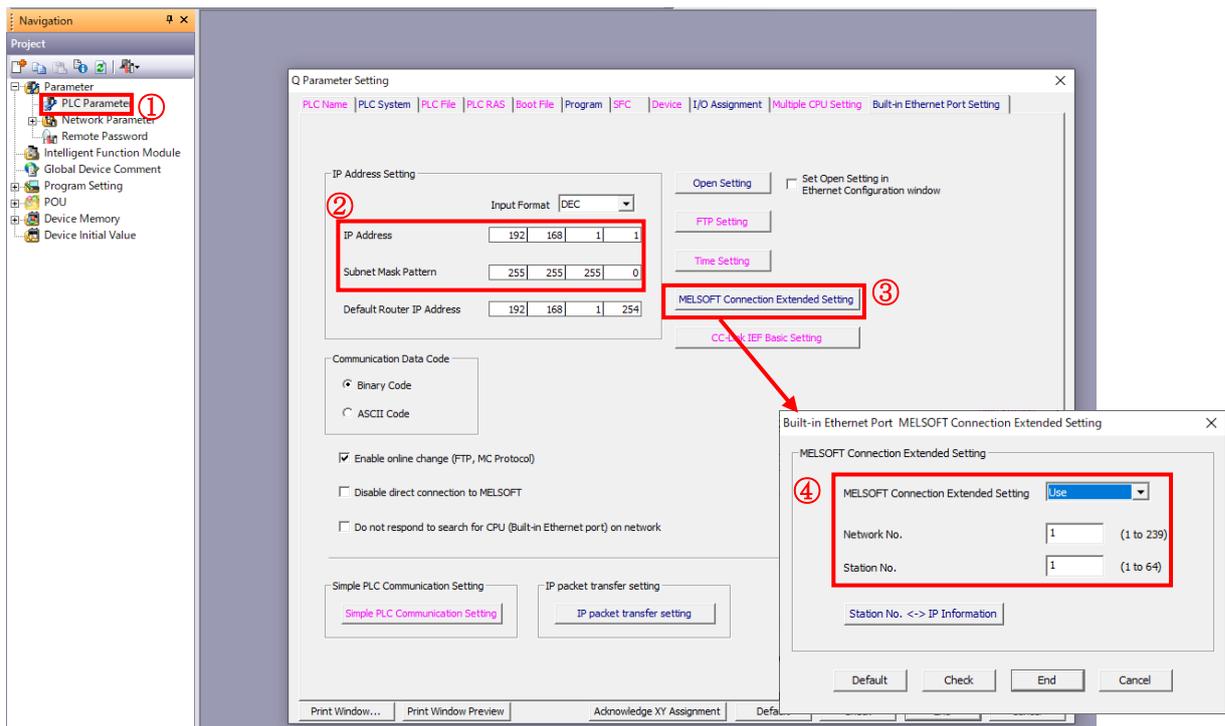


Figure 4-2 Built-in Ethernet module parameter setting (main PLCs)

4.1.1.2. Expansion Ethernet Unit 1 Setting

① to ④ in Figure 4-3 show the setting steps of Expansion Ethernet Unit 1.

- ① From the navigation tree menu, click Ethernet/CC IE/MELSECNET.
- ② Select "Network type: Ethernet" in Unit 1 settings, and enter the network number and station number.
- ③ Opens the Operation Setting.
- ④ Enter the IP address.

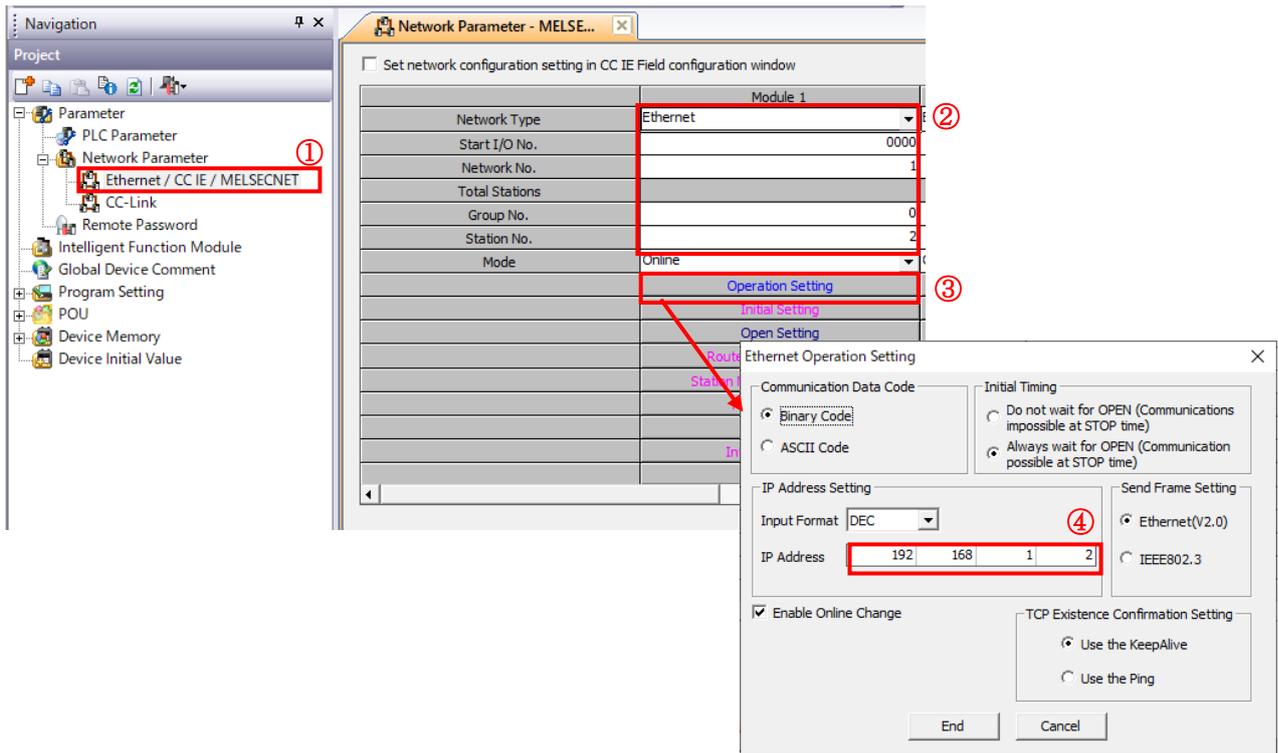


Figure 4-3 Extended Ethernet unit 1 parameter setting

4.1.1.3. Expansion Ethernet Unit 2 setting

① to ⑤ in Figure 4-4 show the setting steps of Expansion Ethernet Unit 2. The procedure for displaying the setting screen is the same as step ① of 4.1.1.2.4.1.1.2

- ① Select "Network type: Ethernet" in Unit 2 settings, and enter the network number and station number.
- ② Opens the Operation Setting.
- ③ Enter the IP address.
- ④ Open the Station No. <->IP Information.
- ⑤ Select " Station No. <->IP Information System: Table Change System" and enter the Network No. and Station No. and IP address of the extended Ethernet unit of the subsidiary PLC.

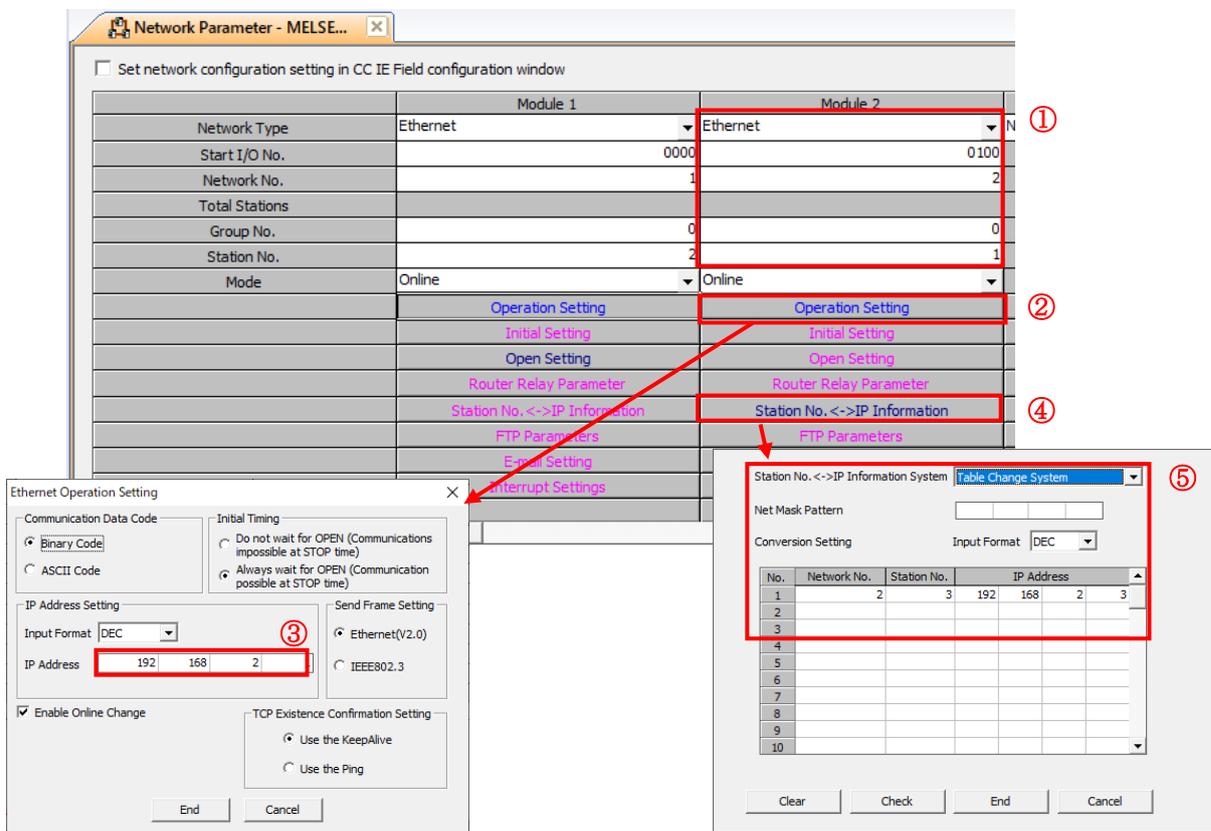


Figure 4-4 Extended Ethernet unit 2 parameter setting

4.1.2. Setting of child PLC

4.1.2.1. Setting of built-in Ethernet unit

① to ④ in Figure 4-5 show the setting steps of the built-in Ethernet module.

- ① After connecting the PLC, click "PLC Parameter" on the navigation tree menu.
- ② Enter the IP address and subnet mask on the Q Parameter Setting screen.
- ③ Open MELSOFT Connection Extended Setting.
- ④ Select "MELSOFT Connection Extended Setting: Use" and enter the network number and station number.

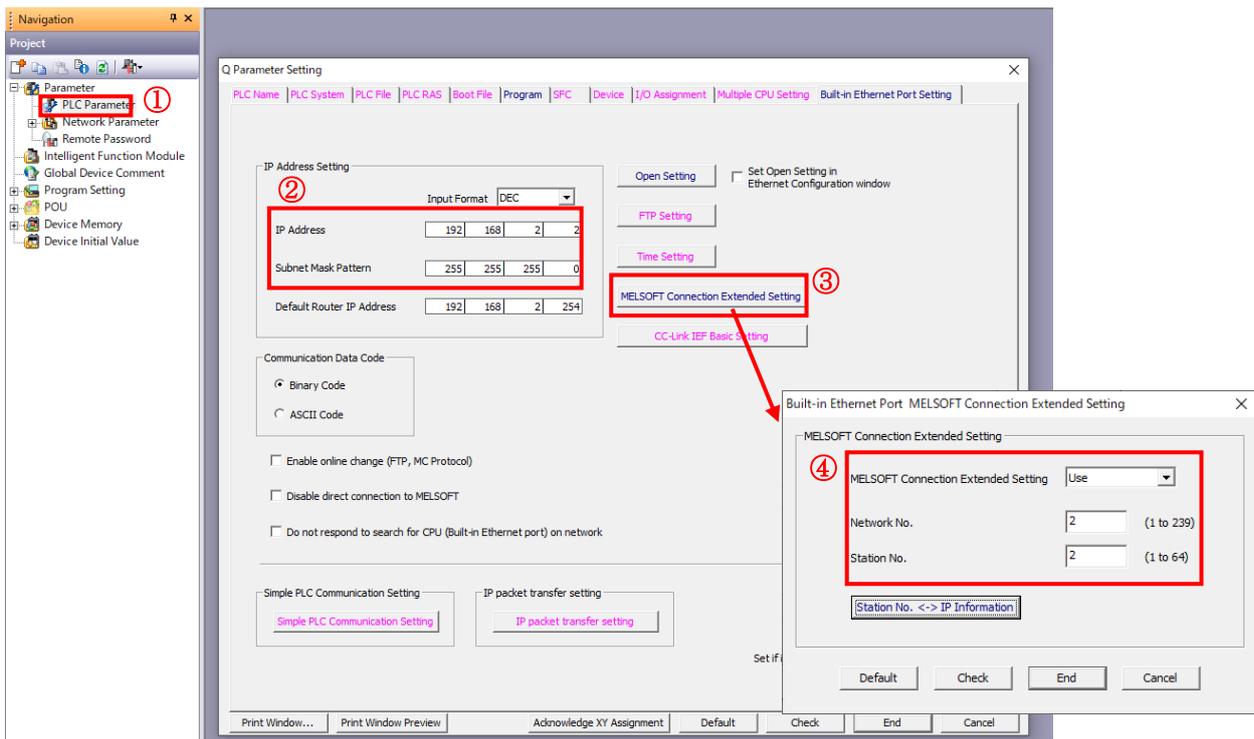


Figure 4-5 Built-in Ethernet module parameter setting (Child PLCs)

4.1.2.2. Expansion Ethernet Unit Settings

① to ④ in Figure 4-6 show the setting steps of Expansion Ethernet Unit 1.

- ① From the navigation tree menu, click Ethernet/CC IE/MELSECNET.
- ② Select "Network type: Ethernet" in Unit 1 settings, and enter the network number and station number.
- ③ Opens the Operation Setting.
- ④ Enter the IP address.

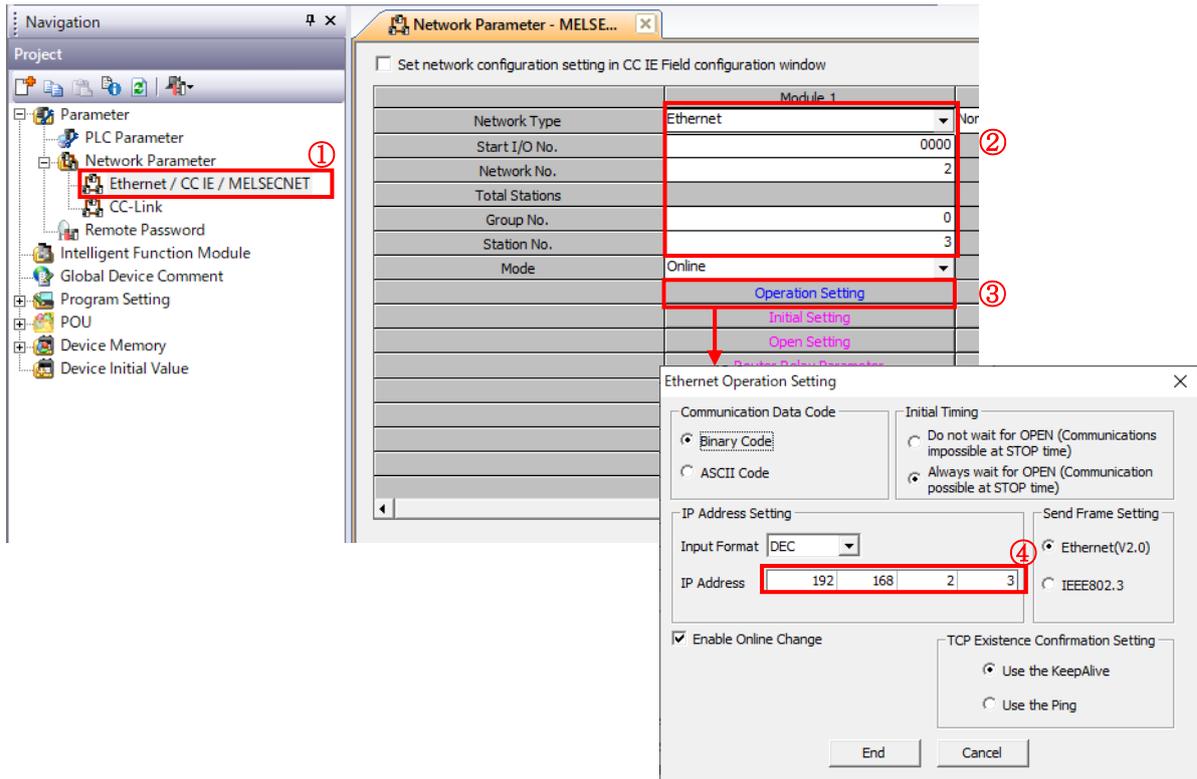


Figure 4-6 Extension Ethernet unit settings (Child PLCs)

4.1.3. Connection parameter setting

To access other stations, set the following parameters when creating variable objects.

"Param=

<request destination network number>:<request destination station number>:<request destination unit I/O number>:<request destination multi-drop station number>[:<monitoring timer>]"

<Request destination network number> : The network number of the child PLC (in hexadecimal).

<Request destination station number> : station number (in 16th decile) of the child PLC.

<Request destination unit I/O number> : 03FF (Management CPU)

<Request destination multi-drop station number> : 00 (Non-multidrop connection).

<Monitoring timer> : Optional.

(ex) "Device=X, Address=0, Param=00:FF:03FF:00:0000"

Acquires the X0 value of PLC with network No.2 and station No.3.