

Yokogawa Electric Corporation
YOKOGAWA PCLink Binary provider

Version 1.0.2

User's guide

August 08, 2022

[Remarks]

[Revision history]

Version	Date	Description
1.0.0	2018-09-14	First edition.
1.0.1	2019-09-03	Fixed internal processing.
1.0.2	2022-08-08	Fixed internal processing.

[Supported models]

Model	Version	Note
FA-M3 Simulation Software Virtual-M3	1.0.0.1	

Contents

1. Introduction.....	4
2. Outline of provider.....	5
2.1. Outline	5
2.2. Method and Properties	6
2.2.1. CaoWorkspace::AddController method.....	6
2.2.1.1. Conn option.....	6
2.2.2. CaoController::AddExtension method	7
2.2.3. CaoController::AddVariable method	7
2.2.4. CaoController::get_VariableNames method.....	7
2.2.5. CaoController::Execute method	7
2.2.6. CaoExtension::AddVariable method	8
2.2.7. CaoExtension::get_VariableNames method	8
2.3. List of Variables	8
2.3.1. CaoController	8
2.3.2. CaoExtension.....	8
2.4. Error code.....	12
3. Sample program.....	13

1. Introduction

This is a user's guide of ORiN provider (YOKOGAWA PCLink Binary provider) that enables to obtain data by using Host link function from PLC manufactured by Yokogawa Electric.

You can use the host link function of PLC manufactured by Yokogawa Electric with this provider.

This guide tells about functions of YOKOGAWA PCLink Binary provider and mounted methods on the controller.

2. Outline of provider

2.1. Outline

YOKOGAWA PCLink Binary provider obtains data by using Host link function (TCP/IP Communication) from PLC manufactured by Yokogawa Electric. The file format of this provider is DLL, Dynamic Link Library; for details, see the following Table 2-1.

Table 2-1 YOKOGAWA PCLinkBinary provider

File name	CaoProvYOKOGAWAPCLinkBinary.dll
ProgID	CaoProv.YOKOGAWA.PCLinkBinary
Registration	regsvr32 CaoProvYOKOGAWAPCLinkBinary.dll
Deregistration	regsvr32 /u CaoProvYOKOGAWAPCLinkBinary.dll

Figure 2-1 shows each provider class and its corresponding flows.

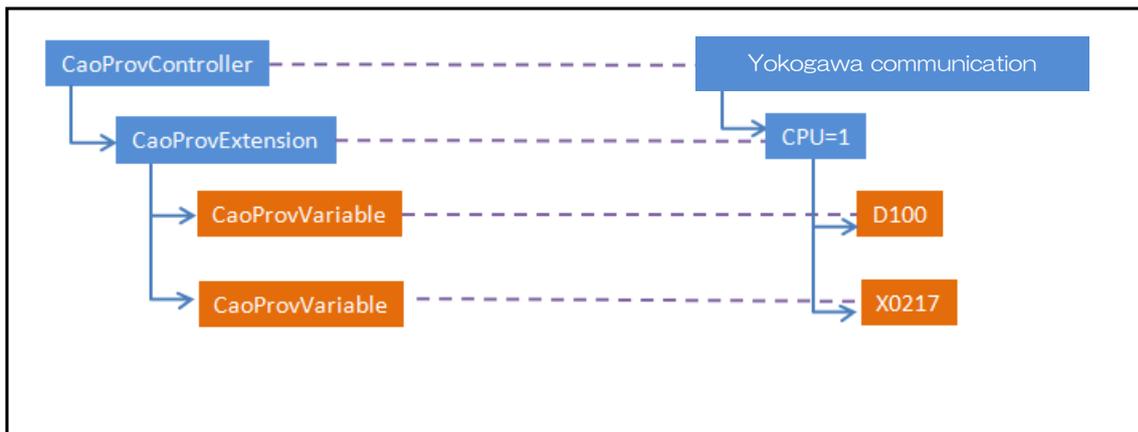


Figure 2-1 each class of providers and corresponding flows of Virtual-M3

2.2. Method and Properties

2.2.1. CaoWorkspace::AddController method

This provider establishes communication by setting the connection parameters when CaoWorkspace::AddController is executed. The specification of AddController is shown below.

Syntax

```
AddController
(
    <bstrCtrlName:BSTR>,          // [in] Controller name (a unique name)
    <bstrProvName:BSTR>,          // [in] Provider name
                                   (Fixed to = "CaoProv.YOKOGAWA.PCLinkBinary")
    <bstrPcName:BSTR>             // [in] Provider execution machine name (unused)
    [,bstrOption:BSTR>]         // [in] Option character strings
)
```

Table 2-2 Option character strings of CaoWorkspace::AddController

Option	Required	Description	Range of value	Default value
CONN=<Eth Connection Option>	✓	Specify IP address of the device to connect. For details, refer to 2.2.1.1	----	Transmission destination IP: -- Transmission destination port :12291 My IP: -- My port: --
TIMEOUT=<Timeout period>	--	Specify the timeout period (ms)	1 to 65535	500

2.2.1.1. Conn option

The following shows the connection parameter strings of Conn option. Parameters surrounded by the square brackets (“[]”) can be omitted. Underlined part shows the default value when the option is not specified.

This provider executes TCP connection when specifying ETH

Eth

“Conn=Eth: <Transmission destination IP>[:< Transmission destination port>[:< Local IP>[:<Local port>]]]”

<Transmission destination IP> : Transmission destination IP Address. “192.168.0.1” etc.

< Transmission destination port> : Transmission destination port No. 12291

<Local IP> : Local IP Address

<Local port> : Local port No.

2.2.2. CaoController::AddExtension method

This method creates a CaoExtension object through CaoController. The specification of AddExtension is as follows:

Syntax

```
AddExtension
(
    <bstrExtensionName:BSTR>           // [in] Variable name (arbitrary)
    [, <bstrOption:BSTR>])           // [in] Option character string
)
```

The following shows the option character string to specify.

Option	Required	Description	Range of value	Default value
CPU=<CPU No. >	✓	Specify the CPU No., which is corresponding to CaoExtension to be added. If the CPU No. which is out of that range is specified, an error may occur as a result.	1 to 4	--

2.2.3. CaoController::AddVariable method

This method creates a CaoVariable object through CaoController. Variable names listed exclusively in Table 2-3 can be used.

The specification of AddExtension is as follows:

Syntax

```
AddVariable
(
    <bstrVariableName:BSTR>           // [in] Variable name
    [, <bstrOption:BSTR>])           // [in] Option character string
)
```

2.2.4. CaoController::get_VariableNames method

This method obtains the list of variable names in Table 2-3.

2.2.5. CaoController::Execute method

In this provider, only “SendCommand” can be used as command. SendCommand can send the character string specified by vntParama (the 2nd argument) as it is to the device, and then return the received character string to pVal (the 3rd argument).

Syntax

```
Execute
(
    <bstrCommand:BSTR>               // [in] Command name (fixed = "SendCommand")
)
```

```
[,<vntParam:VARIANT>] // [in] Parameter (Sending character strings to the device)
[,<pVal:VARIANT>]]) // [in] Execution results (Receiving character strings from the device)
)
```

Example

```
Dim sendByteArray() as Byte={17,1,0,8,0,4,0,0,1,0,1}
Dim retByteArray() = m_caoController.Execute("SendCommand",sendByteArray)
```

2.2.6. CaoExtension::AddVariable method

This method creates a CaoVariable object through CaoController. Variable names listed exclusively in Table 2-4 can be used. To use, see the [CaoController::AddVariable method](#), which is the same method as this one.

2.2.7. CaoExtension::get_VariableNames method

This method obtains the list of variable names in Table 2-4.

2.3. List of Variables

2.3.1. CaoController

The list of variables which can be used for CaoController Class is as shown in Table 2-3. Only get-operation is possible with all the variables.

Table 2-3 CaoController class variable list

Variable Name	Data type of acquisition value	Description	Attribute	
			get	put
@MAKERNAME	VT_BSTR	Return the maker name.	✓	-
@VERSION	VT_BSTR	Return the provider version.	✓	-

2.3.2. CaoExtension

The list of variables which can be used for CaoExtension Class is as shown in Table 2-4. Only get-operation is possible with all the variables.

Table 2-4 CaoExtension class variable list

Variable Name	Description	Attribute		Details
		get	put	
DV<??>	Obtain the data from the sequencer device.	✓	✓	See DV<??>
SP<??>	Obtain the data from the special module.	✓	✓	See SP<??>
IF<??>	Obtain variable information.	✓	-	See IF<??>
MN<??>	Obtain module name.	✓	-	See MN<??>

DV<??>

DV is a variable that obtains the data from the sequencer device. Create the variable with specified arbitrary character strings which come after DV. The following Table 2-5 shows the options which can be specified.

Table 2-5 DV<>option

Option	Required	Description	Range of value	Default value
Device=<Device name>	✓	Specify the device name to read out. Readable device can be referred to Yokogawa Manual. Any value which is out of range will result in an error.	--	--
UserVarWidth=<bit length>	--	Specify the length of reading out for one item. Its bit length to be specified varies as per the device to read out. If that specified bit length is out of the value, Default value will be executed instead.	Link: 1/16/32 Register: 16/32 Timer: 16	Link device: 1 Register device: 16 Timer: 16
Elem=<Number of items to read out>	--	Specify the number of items to read out. If that specified item is out of the value, Default value will be executed instead.	1 to 65535	1
Array=<True/False>	--	Whether it acquires it in the array when one is specified for optional Elem is specified.	True/False	False

Data type of acquisition value:

VT-VARIANT which has the number of elements specified in Elem option will be returned. The number of elements varies as per the UserVarWidth Option. Also, the value which has been obtained from the address specified in Device option will be stored to the head of the array of the element.

1. If you specify 1 bit;
VT_UI1 | VT_VARIANT, which is 1: ON, 0: OFF, will be returned.
2. If you specify 16 bits;
VT_UI2 | VT_VARIANT will be returned.
3. If you specify 32 bits;
VT_UI4 | VT_VARIANT will be returned.

SP<??>

SP is a variable that obtains the data from the special module. Create the variable with specified arbitrary character strings which come after SP. The following Table 2-6 shows the options which can be specified.

Table 2-6 SP<> option

Option	Required	Description	Range of value	Default value
Unit=<Unit No.>	✓	Specify the unit No. of special module. Any value which is out of range will result in an error.	0 to 7	--
Slot=<Slot No.>	✓	Specify the slot No. which is designated in the Unit option. Any value which is out of range will result in an error.	1 to 16	--
Pos=<start position to read out>	✓	Specify the position to read out. Any value which is out of range will result in an error.	0 to 9999	--
UserVarWidth=<bit length>	--	Specify the length to read out per item. If that specified bit length is out of the value, Default value will be executed instead.	16/32	16
Elem=<Number of items to read out>	--	Specify the number of items to read out. If that specified item is out of the value, Default value will be executed instead.	1 to 65535	1
Array=<True/False>	--	Whether it acquires it in the array when one is specified for optional Elem is specified.	True/False	False

Data type of acquisition value:

Although it varies as per the UserVarWidth Option, VT-VARIANT which has the number of elements specified in Elem option will be returned. Also, the value which has been obtained from the address specified in Device option will be stored to the head of the array of the element.

1. If you specify 16 bits;
VT_UI2 | VT_VARIANT will be returned.
2. If you specify 32 bits;
VT_UI4 | VT_VARIANT will be returned.

IF<??>

IF is a variable that obtains information of every kind. Create the variable with specified arbitrary character strings which come after IF. The options which can be specified are as shown in Table 2-7; Data types of acquisition values are shown in Table 2-8.

Table 2-7 IF<> option

Option	Required	Description	Range of value	Default value
No=<some kind of information >	--	Specify the kind of information to be read out. Data type to obtain will get changed depending on the specified values here. If that value is out of the range, Default value will be executed instead.	1: Read CPU information 2: Read system ID, CPU type, area size	1

Table 2-8 IF<> Data type of acquisition value

No	Data type	Description	Range of value
1	VT_ARRAY VT_VARIANT	Data type at the time of No=1	--
	0 VT_UI1	Store CPU status.	0: Ready 1: Not Ready
	1 VT_UI1	Store the program status.	1: Operate 2: Stop 3: Debug 4: ROM Writer
2	VT_ARRAY VT_VARIANT	Data type at the time of No=2	--
	0 VT_BSTR	Store the system ID.	--
	1 VT_BSTR	Store its revision.	--
	2 VT_UI1	Store CPU type.	1: Sequence 2: BASIC
	3 VT_UI2	Store the area size of program. The meaning of the cases is shown respectively as follows: In case of Sequence: Step No. In case of BASIC: Byte No.	--

MN<??>

MN is a variable that obtains Module name. Create the variable with specified arbitrary character strings which come after MN. The following Table 2-9 shows the data types of acquisition values.

Table 2-9 MN<> option

Option	Required	Description	Range of value	Default value
Unit=<Unit No.>	✓	Specify Unit No. to read Module name. Any value which is out of range will result in an error.	0 to 7	--

Table 2-10 MN<> Data type of acquisition value

Data type	Description	Range of value
VT_ARRAY VT_VARIANT	Store Module name information of slot 1 to 16.	--
0 VT_BSTR	Store Module name of slot 1.	--
1 VT_UI1	Store the state of in/output of the slot 1.	0: no in/output relay 1: input relay only 2: output relay only 3: in/output relay
2 VT_UI2	Store the number of in/output of the slot 1.	
:	:	:
:	:	:
44 VT_BSTR	Store Module name of slot 16.	
45 VT_UI1	Store the state of in/output of the slot 16.	
46 VT_UI2	Store the number of in/output of the slot 16.	

2.4. Error code

This provider has the following unique error codes. (Refer to Table 2-11 List of Unique error code)

For ORiN2 common errors, refer to the chapter of Error code in “ORiN2 programming guide”.

Regarding an error code which was received from the device, the code where the value masked by 0x80100000 will be provider's error code. That meaning of the error code which was received from the device can be referred to the specification of host link command of Yokogawa Electric.

Table 2-11 List of Unique error code

Error number	Description
0x80110001	Communication kinds other than TCP / ETH were specified.
0x80110002	CPU is not specified.
0x80110003	The value specified with CPU is out of range.
0x80110004	Device is not specified.
0x80110005	Character string specified with Device is incorrect.
0x80110006	Unit is not specified.
0x80110007	The value specified with Unit is out of range.
0x80110008	Slot is not specified.
0x80110009	The value specified with Slot is out of range.
0x8011000A	Pos is not specified.
0x8011000B	The value specified with Pos is out of range.
0x8011000E	Received a send command and non-compatible command.

3. Sample program

The following sample shows that data which can be obtained for from D200 to 500 words.

Precondition :

- IP address of Device shall be 192.168.0.2.

List 3-1

Sample.frm

```

Dim eng As CaoEngine
Dim ctrl As CaoController
Dim ext As CaoExtension
Dim var As CaoVariable
Private Sub Form_Load()
    Set eng = New CaoEngine
    'Create Controller object
    Set ctrl = eng.Workspaces(0).AddController(
        "YOKOGAWA", "CaoProv.YOKOGAWA.PCLinkBinary", "",
        "CONN=ETH:192.168.0.2")
    'Create Extension object
    Set ext = ctrl.AddExtension(
        "Ext00", "CPU=1")
    'Create Variable object
    Set var = ext.AddVariable(
        "DV00", "Device=D200, Elem=500")
End Sub

'Obtain Value
Private Sub Command1_Click()
    'Obtain Variable value
    Dim d200 As Variant
    Set d200 = var.Value
End Sub

```