

YASKAWA
Extended MEMOBUS provider

Version 1.0.2

User's Guide

January 21, 2022

[Remark]

[Revision History]

Version	Date	Description
1.0.0	2020-06-30	First edition
		Added explanation of register type, details of data type, and list of VT options that can be specified for each data type.
1.0.1	2021-05-19	Fixed a bug with random access
1.0.2	2021-05-27	Fixed a malfunction due to put_Value data type
	2021-06-30	Corrected VT option description
	2022-01-21	Corrected the address value range of the Reference option

[Compatible devices]

Model	Version	Notes
MP900 Series		
MP2000 Series		
MP3000 Series		

[Operation check devices]

Model	Version	Notes
MP3300		

Table of Contents

1. Introduction	4
2. Provider Overview	4
2.1. Introduction	4
2.2. Method properties	6
2.2.1. CaoWorkspace::AddController method	6
2.2.1.1. Conn Optional	7
2.2.2. CaoController::property get_VariableNames	8
2.2.3. CaoVariable:: AddVariable method	9
2.2.3.1. Reference Optional.....	11
2.2.3.2. Elem Optional.....	15
2.2.3.3. VT option.....	15
2.2.3.4. Array Optional.....	16
2.2.3.5. Return Values Type.....	17
2.2.1. CaoVariable::get_Value Property	18
2.2.2. CaoVariable::put_Value Property	19
2.3. Error code.....	20
3. Sample program	22
4. Communication protocol command correspondence table	23
5. Appendix Access Limit	24

1. Introduction

This document is a device compatible with the Extended MEMOBUS protocol. (MP3000/2000/900 series made by Yaskawa Electric Co., Ltd., etc.) From now on, it is referred to as an Extended MEMOBUS device. This is the user's guide for CAO providers that perform batch-write/batch-read/random-write/random-read of register content.

The CAO-providers (CaoProvYaskawaExMemobus.dll) covered in this document are called "Extended MEMOBUS provider".

This provider is developed by "Machine Controller MP3000 Series Communication Function User's Manual", "Machine Controller MP2000 Series Communication Module User's Manual", "Machine Controller MP900 Series User's Manual Communication Module" (hereinafter referred to as the "Extended MEMOBUS Communication Manual"), which are provided by Yaskawa Electric Corporation. This document describes the features of Extended MEMOBUS providers and the methods they implement.

2. Provider Overview

2.1. Introduction

Fig. 2-1 shows the correspondence between the classes of providers and laser markers. CaoProvController corresponds to the Extended MEMOBUS device main unit, and CaoProvVariable corresponds to the register to be accessed.

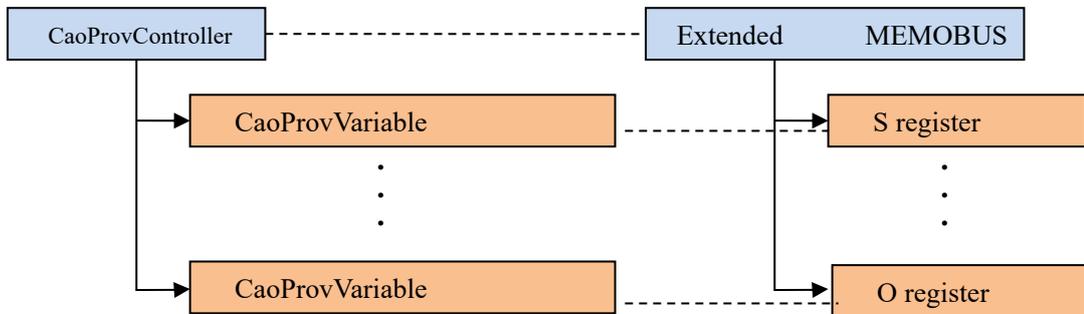


Fig. 2-1 Supported Diagrams of Providers and Extended MEMOBUS Equipment

Fig. 2-2 shows the connectivity between the provider and Extended MEMOBUS compatible devices. Connect the device by Ethernet.

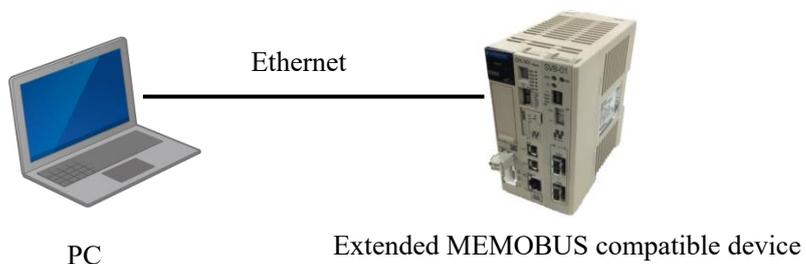


Fig. 2-2 Diagram of connecting providers and Extended MEMOBUS devices

The file format for Extended MEMOBUS providers is DLLs (Dynamic Link Library), which are dynamically loaded when used by the CAO-engine. To use Extended MEMOBUS providers, you must install ORiN2SDK or manually register the registry by referring to Table 2-1.

Table 2-1 Extended MEMOBUS providers file format

File name	CaoProvYaskawaExMemobus.dll
ProgID	CaoProv.YASKAWA.ExMEMOBUS
Register registry ¹	regsvr32 CaoProvYaskawaExMemobus.dll
Delete registry	regsvr32 /u CaoProvYaskawaExMemobus.dll

¹ If it is installed by ORiN SDK, it does not need to be registered/deleted manually.

2.2. Method properties

2.2.1. CaoWorkspace::AddController method

In this provider, the connection parameters are set in CaoWorkspace::AddController and communication(TCP) is connected. The specifications of AddController are shown below.

In addition, the number of simultaneous connections is different depending on the connected device.

Format

```
AddController(
    BSTR bstrCtrlName,           // [in] Controller name (enter any controller name)
    BSTR bstrProvName,          // [in] Provider name = "CaoProv.YASKAWA.ExMEMOBUS"(fixed)
    BSTR bstrPcName,            // [in] Provider's running machine name
    BSTR bstrOption              // [in] Option character string
);
```

Table 2-2 CaoWorkspace::AddController Option character string

Setting items	Option String	Required	Description	Setting range	Default Value
Host	Conn	✓	Specify the IP address and port number of the connection destination. For details about how to write, see 2.2.1. 1Conn Options.2.2.1.1Conn Optional	2.2.1.1Conn Optional	2.2.1.1Conn Optional
Timeout	Timeout	-	Specify the timeout with the connection destination in ms.	1~4294967295	3000
Register address capacity	Bit	✓	Specify the address size of the connected device in 16 or 32. For details, refer to the manual of each device. (e.g.) MP3000 Series: 32, MP2000/900 Series: 16	16,32	-
Host CPU number	CPUNo	-	Specify the CPU number of the connected device. Equipment released before April 2020 is fixed at 1. For other equipment, refer to the manual of each equipment.	1~15	1

2.2.1.1. Conn Optional

The following is a Conn optional connection parameter string:

"tcp:< A >:[< B >]"²

Parameter	Meaning	Required	Setting range	Default Value
A	Destination IP address	✓	0.0.0.0~255.255.255.255	-
B	Port Number	-	1~65535	10001

e.g. 1) Connecting to MP3000 Series (IP address 192.168.1.1, port number 10001)

"Conn = tcp:192.168.1.1, Bit = 32", "Conn = tcp:192.168.1.1:10001, Bit = 32"

e.g. 2) Connecting to MP2000/900 Series (IP address 192.168.1.1, port number 10001), timeout is 1000 ms

"Conn = tcp:192.168.1.1, Timeout = 1000, Bit = 16"

"Conn = tcp:192.168.1.1:10001, Timeout = 1000, Bit = 16"

Usage (CaoScript)

' Connecting to MP3000 Series

Dim ctrl

```
Set ctrl = cao.AddController("ExMemobus", "CaoProv.YASKAWA.ExMEMOBUS", "", "Conn = tcp:192.168.1.1:10001, Bit=32")
```

' Connecting to MP2000/900 Series

Dim ctrl

```
Set ctrl = cao.AddController("ExMemobus", "CaoProv.YASKAWA.ExMEMOBUS", "", "Conn = tcp:192.168.1.1:10001, Bit=16")
```

²Square brackets ("[]") are optional parameters.

2.2.2. CaoController::property get_VariableNames

Gets the list of variables available for this provider. You can obtain the list in Table 2-3.

Return Value Type	Description
VT_ARRAY VT_BSTR	You can get a list of variables that can be used for AddVariable variable names in CaoController.

Usage (CaoScript)

Dim vntNamesList

VntNamesList = dat.ToVar(ctrl.VariableNames)

Dbg.output vntNamesList(0) ‘ @MAKER_NAME

Dbg.output vntNamesList(1) ‘ @VERSION

Dbg.output vntNamesList(2) ‘ @LAST_PACKET_SIZE

Table 2-3 CaoController Class System Variable List

Variable name	Data Type	Description	Attribute	
			Get	Put
@MAKER_NAME	VT_BSTR	Returns the manufacturer named "YASKAWA Electric Corporation".	✓	-
@VERSION	VT_BSTR	Returns the version of the provider.	✓	-
@LAST_PACKET_SIZE	VT_UI4	Returns the size of the last packet sent.	✓	-

2.2.3. CaoVariable::AddVariable method

Creates a CaoVariable from a CaoController. To use a system variable, enter a character string corresponding to each variable name in Table 2-3, and for a user variable, enter an arbitrary character string as the variable name. For option character strings, enter blanks when using system variables and strings that correspond to option character strings in Table 2-4 when using user variables.

Format

```
AddVariable (  
    BSTR bstrVarName, // [in] Variable name.  
    BSTR bstrOption   // [in] Option character string  
);
```

Usage (CaoScript)

```
Dim varMW0  
Set varMW0 = ctrl.AddVariable("MW0", "Reference=MW0, Elem=30, Array=TRUE")  
Dim varRandom  
Set varRandom = ctrl.AddVariable("Random", "Reference=MW10:SL30:GF78:ODFF ")
```

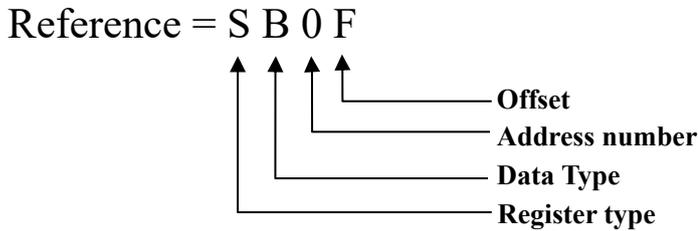
Table 2-4 CaoController::AddVariable Option character string

Setting items	Option character string	Required	Description	Setting range	Default Value
See Also Register	Reference	✓	Specify the register type, data type, and address number to be read/written. For details about how to write, see 2.2.3.1 Reference Optional	See 2.2.3.1 Reference Optional	-
Number of elements	Elem	-	Specify the number of elements to be read/written. For the description method, see 2.2.3.2 See Elem optional.	Depends on reference register	1
Variable Types	VT	-	Specify when you want to handle the data as a character string. Refer to 2.2.3.3 VT Option for the description.	See 2.2.3.3 VT Optional	Depends on reference register
Array type	Array	-	Specifies whether to obtain a value with an array type when reading one element (Elem=1). For details about how to write, see 0 Array Optional	FALSE: Non-Array TRUE: Array	FALSE
Code page	CodePage	-	Specifies the code page to be used when reading/writing. A typical code page is described below. 0: Current WindowsANSI codepage 932:Shift-JIS 1200:UTF-16 12000:UTF-32 65000:UTF-7 65001:UTF-8	0 ~ 65535	0

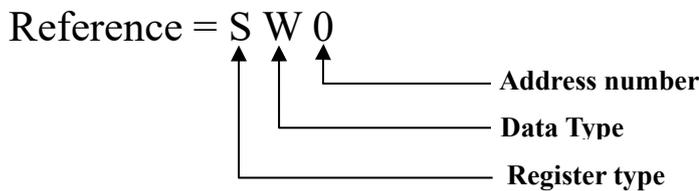
2.2.3.1. Reference Optional

Specify the register type, data type, address number, and offset to be read/written in the following format.

[Data type: B]



[Data type: other than B]



The register type and data type that can be specified by the Reference option differ depending on the register capacity specified by CaoWorkspace::AddController. Refer to Table 2-5 when the register capacity is specified to 16 bits and Table 2-6 when specified to 32 bits. See Table 2-7 for a description of each register type and Table 2-8 for details on the data types.

If random availability in the table is "√", you can specify multiple Reference options separated by [:]. (hereinafter referred to as random access)

The following are usage of Reference option.

"Reference = <access destination 1>[:<access destination 2>]...[:<access destination n>]"³

- e.g.1) "Reference = MW10" Refers to the value of MW10 as an integer value.
- e.g. 2) "Reference = MB10A" Refers to the value of MB10A as a bit-value.
- e.g.3) "Reference = SL10" Refers to the value of a SL10 as a long integer value.
- e.g.4) "Reference = GQ10" Refers to GQ10 as a four-fold long integer value.
- e.g.5) "Reference = IFAB" Refers to IFAB as a real number.
- e.g.6) "Reference = OD20" Refers to the value of OD20 as a double precision real number.
- e.g.7) "Reference = MW10:SL3" References the value of a MW10 as an integer value and the value of a SL3 as a multiple-length integer value at the same time. (random access)

³ Square brackets ("[]") are optional parameters.

Table 2-5 Specifying 16-bit address Reference options

1st character	2nd character	3rd to last character		Random Availability	Data Type	Attribute		
		Register type	Data Type			Address number ⁴	Offset (Hex) ⁵	Get
M	B		0 to 4095 (decimal)	0~F	-	VT_I2	✓	✓
	W		0 to 65535 (decimal)	-	✓	VT_I2	✓	✓
	L	VT_I4						
	Q	VT_I8						
	F	VT_R4						
	D	VT_R8						
I	B		0 to FFF (Hexadecimal)	0~F	-	VT_I2	✓	-
	W		0 to FFFF (Hex)	-	-	VT_I2	✓	-
	L	VT_I4						
	Q	VT_I8						
	F	VT_R4						
	D	VT_R8						

⁴ The upper limit of the address number is as an extended MEMOBUS provider. For the actual upper limit width, refer to the manual of each device.

⁵ The offset indicates the number of bits from the specified address. The size of one address is one word.

Table 2-6 Specifying 32-bit address Reference options

1st character	2nd character	3rd to last character		Random Availability	Data Type	Attribute		
		Register type	Data Type			Address number ⁴	Offset (Hex) ⁵	Get
S	B		0 to 268435455 (decimal)	0~F	-	VT_I2	✓	✓
	W		0 to 4294967295	-	✓	VT_I2		
	L		(decimal)			VT_I4		
	Q					VT_I8		
	F					VT_R4		
	D					VT_R8		
M	B		0 to 268435455 (decimal)	0~F	-	VT_I2	✓	✓
	W		0 to 4294967295	-	✓	VT_I2		
	L		(decimal)			VT_I4		
	Q					VT_I8		
	F					VT_R4		
	D					VT_R8		
G	B		0 to 268435455 (decimal)	0~F	-	VT_I2	✓	✓
	W		0 to 4294967295	-	✓	VT_I2		
	L		(decimal)			VT_I4		
	Q					VT_I8		
	F					VT_R4		
	D					VT_R8		
I	B		0 to FFFFFFFF (Hex)	0~F	-	VT_I2	✓	-
	W		0 to FFFFFFFF	-	✓	VT_I2		
	L		(Hex)			VT_I4		
	Q					VT_I8		
	F					VT_R4		
	D					VT_R8		
O	B		0 to FFFFFFFF (Hex)	0~F	-	VT_I2	✓	✓
	W		0 to FFFFFFFF	-	✓	VT_I2		

	L	(Hex)				VT_I4		
	Q					VT_I8		
	F					VT_R4		
	D					VT_R8		

Table 2-7 Description of register type

Register type	Register type name	Description
S	System register	Registers prepared by the system. The machine controller is reported. When the system starts, SW00000 to SW00049 are cleared to 0. Battery backup is possible.
M	Data register	Registers used for I / F between programs. Battery backup is not possible.
G	Global register	Registers used for I / F between programs. Battery backup is not possible.
I	Input register	This is a register used for input data.
O	Output register	This is a register used for output data.

Table 2-8 Data type details

Data type	Type	Size	Description
B	Bit	1 bit	It is used to judge the relay circuit and on / off conditions.
W	Integer	1 Word	It is used for numerical calculation.
L	Long integer	2 Word	It is used for numerical calculation.
Q	4 long integer	4 Word	It is used for numerical calculation.
F	Real	2 Word	It is used for advanced numerical operations.
D	Double	4 Word	It is used for advanced numerical operations.

2.2.3.2. Elem Optional

Specify the number of elements to be read/written in decimal. When specifying in hexadecimal, specify in the type "0x", "&H~" or "~H". When Elem option is omitted, the default value is 1. When Reference option is specified more than once (random access), Elem option is ignored.

The following are Elem optional parameter strings:

"[Elem = [<number of elements>]]"⁶

<When specifying 20 elements>

- e.g.1) "Reference=MW10, Elem=20" Refers to the value of MW29 from MW10 as an integer value.
- e.g.2) "Reference=SL20, Elem=0x14" Refers to the value of SL29 from SL20 as a long integer value.
- e.g.3) "Reference=GF1A, Elem=&H14" Refers to GF2D from GF1A as a real number.
- e.g.4) "Reference=OB 9001, Elem=14 H" Refers to the value of OB9014 from OB9001 as a bit-value.

2.2.3.3. VT option

When Get_Value is executed, the return type is the data type specified in Table 2-5 or Table 2-6 in Reference option. You can get the data as a character string by specifying "BSTR" in the VT option. The code page of the character string is specified by the "CodePage" option. The method of specifying the VT option differs depending on whether it is random access or non-random access.

- For non-random access
You can omit the VT option or specify the BSTR. However, it cannot be specified when the data type specified in the Reference option is "B".

"[VT = [<VT character string>]]"⁷

- (e.g.) "Reference=MW10" Return Values Type: VT_I2
- (e.g.) "Reference=MW10, VT=BSTR" Return Values Type: BSTR

Table 2-9 List of VT options that can be specified during non-random access

Reference Optional data type	Specifiable VT option	Data type
B	-	-
W	BSTR	VT_BSTR
L	BSTR	VT_BSTR
Q	BSTR	VT_BSTR
F	BSTR	VT_BSTR

⁶ Square brackets ("[]") are optional parameters.
⁷ Square brackets ("[]") are optional parameters.

D	BSTR	VT_BSTR
---	------	---------

- For random access

You can omit the VT option or specify as many VT options as reference options. Please note that only a part cannot be specified or omitted. Table 2-10 shows the VT options that can be specified.

"[VT = [<VT character string>][:<VT character string>]...[:<VT character string>]]"⁸

- (e.g.) "Reference=MW10:IL45, VT=I2:BSTR" Return Values Type: VT_VARIANT|VT_ARRAY
- (e.g.) "Reference=MW10:IL45" Return Values Type: VT_VARIANT|VT_ARRAY
- (e.g.) "Reference=MW10:IL45, VT=BSTR:BSTR" Return Values Type: VT_VARIANT|VT_ARRAY

Table 2-10 List of VT options that can be specified during random access

Reference Optional data type	Specifiable VT option	Data type
W	I2	VT_I2
	BSTR	VT_BSTR
L	I4	VT_I4
	BSTR	VT_BSTR
Q	I8	VT_I8
	BSTR	VT_BSTR
F	R4	VT_R4
	BSTR	VT_BSTR
D	R8	VT_R8
	BSTR	VT_BSTR

2.2.3.4. Array Optional

Specifies whether to obtain the read value in array format when Elem option is 1. If Array option is omitted, the default value is False. If you specify BSTR for random access or VT options, Array option is ignored.

The following are Array optional parameter strings:

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

- (e.g.) "Reference=MW10" Return Values Type: VT_I2

⁸ Square brackets ("[]") are optional parameters.

(e.g.) "Reference=MW10, Array=FALSE" Return Values Type: VT_I2
 (e.g.) "Reference=MW10, Array=TRUE" Return Values of type: VT_I2 | VT_ARRAY

2.2.3.5. Return Values Type

When Get_Value is executed, the return type is the data type specified in Table 2-5 or Table 2-6 in Reference option. When the VT option is specified, the VT option type takes precedence.

Whether the return value is in array format depends on the combination of random access, Elem options, and Array options.

Table 2-11 shows whether an array format exists.

Table 2-11 Availability of array formats

Random access ⁹	Elem	Array	Array type availability
N	1	FALSE	-
		TRUE	✓
	more than 2	-	✓
Y	-	-	✓

(e.g.) "Reference=MW10" Return Values Type: VT_I2
 (e.g.) "Reference=MW10, Array=FALSE" Return Values Type: VT_I2
 (e.g.) "Reference=MW10, Array=TRUE" Return Values of type: VT_I2 | VT_ARRAY
 (e.g.) "Reference=MW10, Elem=3, Array=FALSE" Return Values of type: VT_I2 | VT_ARRAY
 (e.g.) "Reference=MW10:MW20, Array=FALSE" Return Type: VT_VARIANT| VT_ARRAY

⁹ Whether or not more than one Reference is specified

2.2.1. CaoVariable::get_Value Property

Gets the status of the variables created by AddVariable method. The return type depends on the option data specified in AddVariable. For more information, see section 2.2.3, CaoWorkspace::AddController Method.

Usage (CaoScript)

```
Dim varMW0
```

```
Set varMW0 = ctrl.AddVariable("MW0", "Reference=MW0")
```

```
dbg.Output varMW0.value           'e.g. 100
```

```
Dim varMultiMW
```

```
Dim vntMultiMW
```

```
Set varMultiMW = ctrl.AddVariable("MultiMW", "Reference=MW0, Elem=3")
```

```
vntMultiMW = DAT.ToVar (varMultiMW.Value)
```

```
dbg.Output vntMultiMW(0)         'e.g. 100
```

```
dbg.Output vntMultiMW(1)         'e.g. 50
```

```
dbg.Output vntMultiMW(2)         'e.g. 200
```

```
Dim varRandom
```

```
Dim vntRandom
```

```
Set varRandom = ctrl.AddVariable("Random", "Reference=MW10:SL30:GF78:ODFF")
```

```
vntRandom = DAT.ToVar (varRandom.Value)
```

```
dbg.Output vntRandom(0)         'e.g. 1000
```

```
dbg.Output vntRandom(1)         'e.g. 100000
```

```
dbg.Output vntRandom(2)         'e.g. -4.316021E+008
```

```
dbg.Output vntRandom(3)         'e.g. -3.386099378622854E+216
```

2.2.2. CaoVariable::put_Value Property

Set the value of the variable created by AddVariable method. If you specify an option with "✓" for put in Table 2-5 and Table 2-6 in AddVariable, you can use the put_Value property. The data size that can be set depends on the data type of Reference option. Table 2-12 shows the data sizes that can be set for each data type. If the number of data items to be set is more than the number of elements of Elem option or the number of random accesses, the excess data items are ignored.

Table 2-12 Data sizes that can be set

Data Type	MIN	MAX
B	0 (False)	Non-zero (True)
W	-32788	32767
L	-2147483648	2147483647
Q	-9223372036854775808	9223372036854775807
F	-3.402823E+038	3.402823E+038
D	-1.797693E+308	1.797693E+308

Usage (CaoScript)

```
Dim varMW0
```

```
Set varMW0 = ctrl.AddVariable("MW0", "Reference=MW0")
```

```
VarMW0.value = 100
```

```
Dim varMultiMW
```

```
Dim arrMultiMW
```

```
Set varMultiMW = ctrl.AddVariable("MultiMW", "Reference=MW0, Elem=3")
```

```
ArrMultiMW = Array(100, 50, 200)
```

```
VarMultiMW.Value = DAT.ChangeType(arrMultiMW, 2)
```

```
Dim varRandom
```

```
Dim arrRandom
```

```
Set varRandom = ctrl.AddVariable("Random", "Reference=MW10:SL30:GF78:ODFF")
```

```
ArrRandom = Array(1, 2, 3, 4)
```

```
VarRandom.Value = DAT.ChangeType(arrRandom, 2)
```

2.3. Error code

This provider defines a unique error code. Unique error codes are shown in Table 2-13. For ORiN2 common error, refer to the Error Codes section of ORiN2 Programming Guide.¹⁰

Table 2-13 Unique error codes

Error Number	Description
0x80100000	The option character string Conn specified in AddController method is invalid. Specify in the type Conn = tcp:<IP address>[:<port number>]. Specify an IP address in the range of 0.0.0.0 to 255.255.255.255 and a port number in the range of 1 to 65535.
0x80100001	The option character string Timeout specified in AddController method is invalid. Specify a value in the range of 1 to 4294967295 ms.
0x80100002	The option character string Bit specified in AddController method is invalid. Specify 16 or 32.
0x80100003	The option character string CPUNo specified in AddController method is invalid. Specify a value in the range from 0 to 15.
0x80100004	The option character string Reference specified in AddVariable method is invalid. Specify in the following format. Reference = <access destination 1>[:<access destination 2>]...[:<access destination n>] Follow Table 2-5 or Table 2-6. In random access, 'B' cannot be specified for the data type. When 'B' is specified as the data type, specify the offset.
0x80100005	The option character string VT specified in AddVariable method is invalid. Specify in the format VT=[<VT option character string>][:<VT option character string>]...[:<VT option character string>]. Follow Table 2-10 Ensure that the number is equal to the number of Reference option-specified.
0x80100006	The option character string Elem specified in AddVariable method is invalid. Specify a value in the range from 1 to 4294967295. For hexadecimal notation, use "0x~", "&H~", and "~H".
0x80100007	The option character string Array specified in AddVariable method is invalid. Specify "TRUE" or "FALSE".
0x80100008	The random access count for the option character string Reference specified in

¹⁰ If a timeout error (0x80000900) occurs while communication is established, obtain the packet size sent to the device from the system variable [:@LAST_PACKET_SIZE] and check whether the packet size exceeds the maximum packet size of the connected device.

	AddVariable method exceeded the upper limit. Review the number specified by referring to the appendix.
0x80100009	The specified number of option character string Elem specified in AddVariable method exceeded the upper limit. Review the number specified by referring to the appendix.
0x8010000A	Put_Value cannot be executed if the register type specified in Reference option is 'I'.
0x8010000B	The data type specified during Put_Value execution is invalid. For random access, specify the array type.
0x8010000C	The number of data items specified during execution of Put_Value is invalid. Specify the same number as the number of elements to be accessed.
0x8010000D	An out-of-range value was detected during type conversion of read data. If it occurs during Get_Value execution, the data type specified in the VT option may be smaller than the number of access points of the data type specified in Reference option. If it occurs during Put_Value execution, the specified data may be larger than the number of access points of the data type specified by Reference option. Refer to Table 5-1 for the number of access points for each data type. Follow Table 5-1.
0x8010000E	Error code that is output when an illegal packet is received. Use a WireShark, etc. to capture the response packet and investigate the cause of the error from the user's manual that is used.
0x8010000F	The option character string CodePage specified in AddVariable method is invalid. Specify a value in the range from 1 to 65535.
0x801100xx	An error code was received from an Extended MEMOBUS device. The last two digits "xx" are the error codes received. Check the cause of the error from the user's manual that you are using.

3. Sample program

An example AddVariable programme is shown below.

Usage (CaoScript)

e.g.: Rewrite the value of the Extended MEMOBUS device MW0 of the 32-bit address.

```
' Create CaoController
Dim ctrl
Set ctrl = cao.AddController("ExMemobus", "CaoProv.YASKAWA.ExMEMOBUS", "",
                                                                    "Conn = tcp:192.168.1.1:10001, bit=32")

' Create CaoVariable
Dim varMW0
Set varMW0 = ctrl.AddVariable("MW0", "Reference=MW0")

' Read MW0
dbg.Output varMW0.value           'e.g. 50

' Write MW0
varMW0.value = 100
dbg.Output varMW0.value           'e.g. 100

' Delete CaoVariable
If Not varMW0 Is Nothing Then
    ctrl.Variables.remove varMW0.Index
    Set varMW0 = Nothing
End If

' Delete CaoController
If Not ctrl Is Nothing Then
    cao.Controllers.remove ctrl.Index
    Set ctrl = Nothing
End If
```

4. Communication protocol command correspondence table

Table 4-1 shows the correspondence between Variable variables implemented by this provider and communication commands in the Extended MEMOBUS communication manual.

Table 4-1 Correspondence between communication commands

Register Capacity	Reference Optional			Get/put	Communication Commands			
	Random Access	Register Type	Data Type		MFC ¹¹	SFC ¹²	Function Description	
16	✓	-	-	Get	20	0D	Discontinuous read of holding register (Extended)	
				Put	20	0E	Discontinuous Write Holding Register (Extended)	
	-	M	B	Get	20	01	Read coil status	
				Put	20	0F	Changing the status of multiple coils	
			WLQFD	Get	20	09	Read Holding Register Contents (Extended)	
				Put	20	10	Writing to Multiple Holding Registers	
		I	B	Get	20	02	Read input relay status	
				Put	-	-	-	
			WLQFD	Get	20	0A	Read the contents of the input register (Extended)	
				Put	-	-	-	
	32	✓	-	-	Get	43	4D	Discontinuous register read
					Put	43	0E	Discontinuous register write
-		-	B	Get	43	41	Read bit data status	
				Put	20	4F	Changing the state of multi-bit data	
			WLQFD	Get	20	49	Read register contents	
				Put	20	4B	Writing to Multiple Registers	

¹¹ Major Function Codes

¹² Sub-function code

5. Appendix Access Limit

Based on the number of random accesses, the number of points per data type element, and the number of elements specified in Reference and Elem options, the number of access points is determined as follows: Table 5-1 shows the number of points per element for each data type. To the upper limit of the number of access points of the Extended MEMOBUS protocol. Read each manual for the access limit for each device.

[Except for random access]

Number of access points = (number of data type points) - (number of elements)

[For random access]

Number of access points = total number of points of each data type

e.g.1) "Reference=MW10"	Number of access points: 2 (= 2 - 1)
e.g.2) "Reference=MF10, Elem=10"	Number of access points: 40 (= 4 - 10)
e.g.3) "Reference=MD10, Elem=100"	Number of access points: 800 (= 8 - 100)
e.g.4) "Reference=MW10:ML20:MQ30"	Number of access points: 14 (= 2 + 4 + 8)

Table 5-1 Number of points per element for each data type

Data Type	Points
B	1
W	2
L	4
Q	8
F	4
D	8

Table 5-2 Access upper limit (Extended MEMOBUS protocol)

Reference			Access limit				
Random access	1st char.	2nd char.	16-bit address		32-bit address		
	Register type	Data Type	Get	Put	Get	Put	
✓	-	-	32757	16378	10919	8189	
-	S	B	-		524113	524049	
		W			32756	32754	
		L			16378	16377	
		Q			8189	8188	
		F			16378	16377	
		D			8189	8188	
	M	B	524144	524105	524113	524049	
		W	32757	32756	32756	32754	
		L	16378	16378	16378	16377	
		Q	8189	8189	8189	8188	
		F	16378	16378	16378	16377	
		D	8189	8189	8189	8188	
	G	B	-		524113	524049	
		W			32756	32754	
		L			16378	16377	
		Q			8189	8188	
		F			16378	16377	
		D			8189	8188	
	I	B	524144	-		524113	-
		W	32757			32756	
		L	16278			16378	
		Q	8189			8189	
		F	16278			16378	
		D	8189			8189	
O	B	-		524113	524049		
	W			32756	32754		
	L			16378	16377		
	Q			8189	8188		
	F			16378	16377		
	D			8189	8188		