MxComponent プロバイダ

三菱電機 MxComponent 対応デバイス

Version 1.0.3

ユーザーズ ガイド

July 6, 2022

【備考】

【改版履歴】

バージョン	日付	内容
1.0.0.0	2007-05-08	初版.
1.0.1.0	2007-08-28	マニュアル追記.
1.0.1.1	2010-02-12	エラーコード追加
1.0.1.2	2011-08-06	Size オプションのデフォルト値変更
1.0.1	2012-07-17	ドキュメントのバージョンルールを変更
1.0.2	2015-01-30	AddVariable()に VARTYPE, DATATYPE オプションを追加
	2022-07-06	ドキュメント内の無効なリンクを削除 誤記訂正
1.0.3	2022-08-23	@Buffer 変数が生成できない問題を修正

【対応機器】

機種	バージョン	注意事項

目次

1. はじめに

本書は，MELCO 製 MxComponent の ACT コントローラを使用して通信を行うプロバイダである，MxCompo プロバイダのユーザーズガイドです.

MxCompo プロバイダは，MX Component Version 3 の ACT コントローラ“ActEasyIF”を使用することで MxComponent が対応している全通信経路で通信することができます.

ActEasyIF の詳細については，MxComponent のマニュアルを参照して下さい.

注意:

MxCompo プロバイダを使用するには，MxComponent をインストールしなければなりません.

“ActEasyIF”が対応しているシーケンサに対してのみ，接続することができます.

2. プロバイダの概要

2.1. 概要

MxCompo プロバイダは，CAO API を実行するときに対応する ActEasyIF コントロールのメソッドを実行します．以下に CAO API と MELSEC API の対応について示します．

表 2-1 CAO API と MELSEC API の対応表

CAO API		ActEasyIF Method
CaoController::FinalConstruct		put_ActLogicalStationNumber Open
CaoController::FinalRelease		Close
CaoTask::Start		SetCpuStatus
CaoTask::Stop		SetCpuStatus
CaoVariable::put_Value	（ユーザ変数）	SetDevice（1 点書込みの場合） WriteDeviceBlock
	（@BUFFER）	WriteBuffer
CaoVariable::get_Value	（ユーザ変数）	GetDevice（1 点読込みの場合） ReadDeviceBlock
	（@BUFFER）	ReadBuffer
	（@CLOCK）	GetClockData
	（@CPUTYPE）	GetCpuType

また，CAO API を通して実行された ActEasyIF のメソッドがエラーをとなった場合は，Controller::Execute()で“GetLastError”コマンドを使用することで MxComponent のエラー番号を取得することができます．

プロバイダの概要について記述します．

表 2-2 MxCompo プロバイダ

ファイル名	CaoProvMxCompo.DLL
ProgID	CaoProv.MELCO.MxCompo
レジストリ登録	regsvr32 CaoProvMxCompo.DLL
レジストリ登録の抹消	regsvr32 /u CaoProvMxCompo.DLL

2.2. メソッド・プロパティ

2.2.1. CaoWorkspace::AddController メソッド

MxCompo プロバイダでは Controller オブジェクトの生成時に ActEasyIF::Open()を使用して通信接続を行います. このとき通信先の論理局番を指定します.

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

bstrCtrlName	:	[in]	コントローラ名
bstrProvName	:	[in]	プロバイダ名. 固定値 =" CaoProv.MELCO.MxCompo".
bstrPcName	:	[in]	プロバイダの実行マシン名
bstrOption	:	[in]	オプション文字列＝"LogicID=<論理局番>"

以下にオプション文字列に指定するリストを示します.

表 2-3 CaoWorkspace::AddController のオプション文字列

オプション	意味
LogicID=<論理局番>	通信先を指定する論理局番(必須) 指定する論理局番は, あらかじめ MxComponent の通信設定ユーティリティで設定されている必要があります. 通信設定ユーティリティの詳細については MxComponent のマニュアルを参照して下さい.

2.2.2. CaoController::AddTask メソッド

通信先の CPU 状態の実行及び停止を行う Task オブジェクトを生成します.
タスク名には, 任意の名前を使用することができます.

書式 AddTask(<bstrName:BSTR > [,<bstrOption:BSTR>])

bstrName	:	[in]	タスク名
bstrOption	:	[in]	オプション文字列(未使用)

2.2.3. CaoController::AddVariable メソッド

通信先デバイスにアクセスし, 値の取得及び設定を行う Variable オブジェクトを生成します.

書式 AddVariable(<bstrName:BSTR > [,<bstrOption:BSTR>])

bstrName	:	[in]	変数名
bstrOption	:	[in]	オプション文字列

このメソッドでは，オプション文字列に以下のものを指定することができます．

表 2-4 CaoController::AddVariable のオプション文字列

オプション	意味
Size=<ワードサイズ>	ユーザ変数及び“@BUFFER”で読書きするサイズをワード単位で設定します．[1]（デフォルト:0） ただし，Size=0を指定した場合のみ，デバイス1点の読書きを行います．
VARTYPE=<データ型> （VT=<データ型>）	PLC 側のデータ型（デフォルト：VT_I4）
DATATYPE=<データ型> （DT=<データ型>）	CAO クライアントに返すデータ型 Size オプションと同時に指定した場合はこちらが優先

表 2-5 VARTYPE，DATATYPE オプションで指定できるデータ型

データ型	表記
ビットデータ	BOOL BIT
1バイト整数 （符号あり）	I1 CHAR
1バイト整数 （符号なし）	UI1 BYTE
2バイト整数 （符号あり）	I2 SHORT
2バイト整数 （符号なし）	UI2 WORD
4バイト整数 （符号あり）	I4 LONG
4バイト整数 （符号なし）	UI4 DWORD
単精度浮動小数点	R4 SINGLE

VARTYPE オプションを指定することで，PLC 側のデータ構造を判断し，適切なサイズを切り出します．

[1] ユーザ変数がビットデバイスを指定した場合，1 ワード=16 点を 1 セットとして読書きを行います．

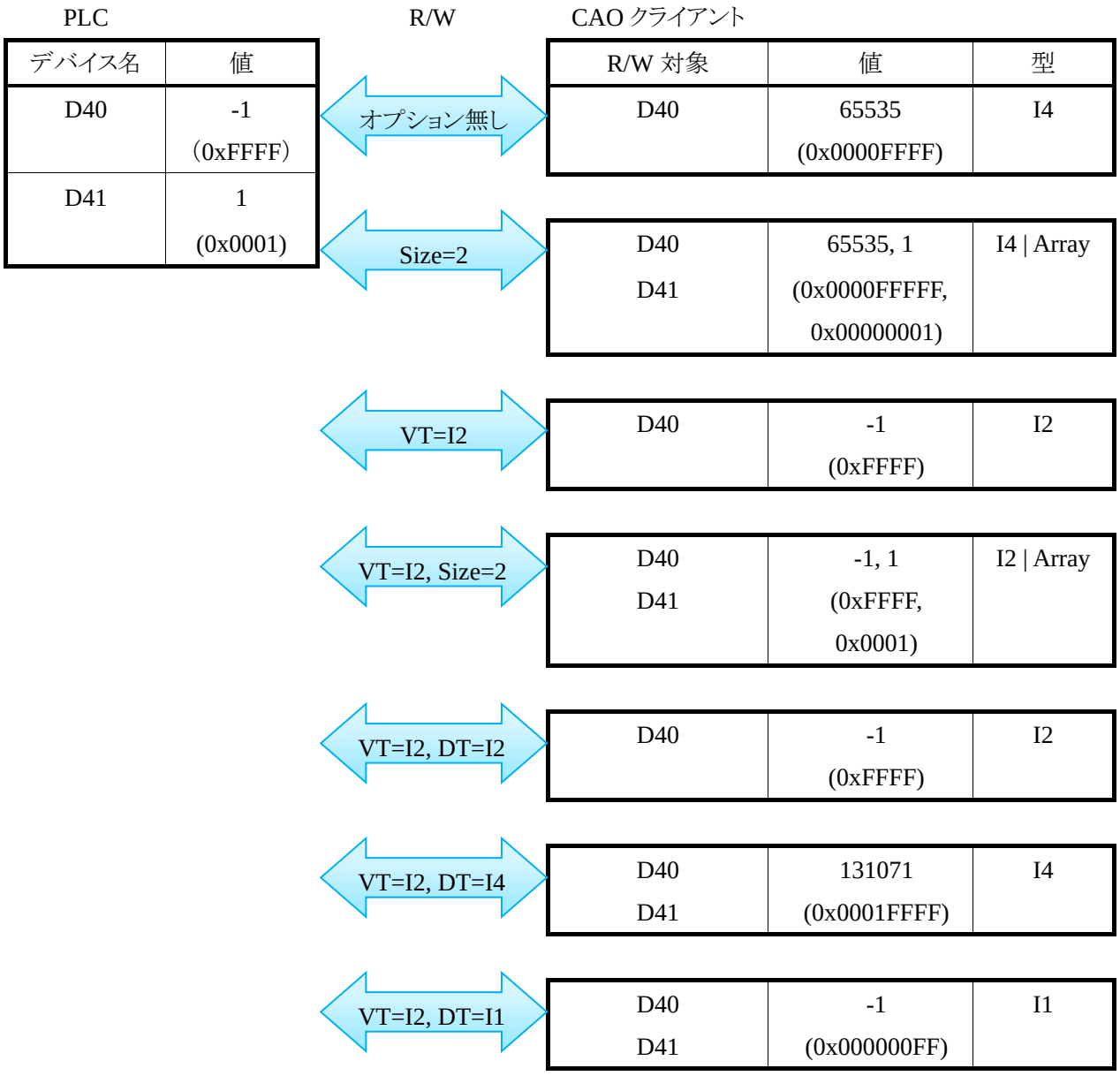
VARTYPE オプションで PLC 側のデータより小さいサイズの型を指定した場合，下位バイトの情報のみを取得設定します．また，大きいサイズの型を指定した場合は上位バイトは 0 で埋められます．

DATATYPE オプションを指定すると，データを指定した型に変換します．

DATATYPE オプションの方が VARTYPE オプションより大きい型を指定した場合，下位バイトを指定したデバイス，上位バイトに次のアドレスのでデバイスの値が格納されます．

DATATYPE オプションの方が VARTYPE オプションより小さい型を指定した場合，不足している箇所を0で埋めたデータが使用されます．

図 2-1 AddVariable のオプションによるデータ変換



2.2.4. CaoController::Execute メソッド

コマンド名に“GetLastError”を指定することで，ActEasyIF コントロールで最後に発生したエラー番号を取得します．

書式 [<vntRet:VARIANT> =] Execute(<bstrCmd:BSTR > [,<vntParam:VARIANT>])

bstrCmd	:	[in]	コマンド＝“GetLastError”
vntParam	:	[in]	パラメータ（未使用）
vntRet	:	[out]	返り値

2.2.5. CaoTask::Start メソッド

接続先の CPU を実行状態にします．

書式 Start(<lMode:LONG > [,<bstrOption:BSTR>])

lMode	:	[in]	開始モード（未使用）
bstrOption	:	[in]	オプション（未使用）

第 1, 2 引数はともに無視されます．しかし，第 1 引数の開始モードは必須入力なのため，常に 0 を設定してください．

2.2.6. CaoTask::Start メソッド

接続先の CPU を停止状態にします．

書式 Stop(<lMode:LONG > [,<bstrOption:BSTR>])

lMode	:	[in]	停止モード
bstrOption	:	[in]	オプション（未使用）

停止モードに“2”（ステップ停止）を指定した場合は CPU 状態を PAUSE に，それ以外の値を指定したときは STOP に設定します．

2.2.7. CaoVariable::put_Value プロパティ

変数名に対応する情報を書き込みます．詳細については 2.3.1 を参照してください．

2.2.8. CaoVariable::get_Value プロパティ

変数名に対応する情報を読み込みます．詳細については 2.3.1 を参照してください．

2.3. 変数一覧

2.3.1. コントローラクラス

表 2-6 コントローラクラス ユーザ変数一覧

変数名	データ型	説明	属性	
			get	put
<デバイス名>	VARTYPE オプションで指定した型または DATATYPE オプションで指定した型	指定したデバイスに対して読書きを行います. デバイス名の指定書式については, MxComponent のマニュアルを参照してください. AddVariable の Size オプションで指定したワードサイズ分を一括読み書きします. Size オプションで0を指定したときは, デバイス名で指定したデバイス1点の読み書きを行います. 読み書き時のデータの格納方法については, 対応する ActEasyIF のメソッドに従います. 対応する関数については表 2-1 を参照してください.	○	○

表 2-7 コントローラクラス システム変数一覧

変数名	データ型	説明	属性	
			get	put
@BUFFER: <IO 番号>: <先頭アドレス>	VT_I2| VT_ARRAY	接続先のバッファメモリに対して読み書きを行います. AddVariable の Size オプションで指定したサイズ分を一括読み書きします.	○	○
@CLOCK	VT_DATE	接続先 CPU の時間を取得します.	○	-
@CPUTYPE	VT_VARIANT｜ VTARRAY	接続先 CPU 情報を取得します. 1 番目の要素に CPU 型名文字列, 2 番目の要素に CPU 型名番号を格納します.	○	-

2.4. エラーコード

MxCompo プロバイダでは, 固有のエラーコードはありません. ORiN2 共通エラーについては,「ORiN2 プログラミングガイド」のエラーコードの章を参照してください.