

MEWTOCOL-COM プロバイダ

Panasonic 製 PLC FP シリーズ用

Version 1.2.3

ユーザーズ ガイド

February 7, 2023

備考:

【改版履歴】

バージョン	日付	内容
1.0.0	2018-11-13	初版
1.1.0	2020-05-22	AddController 時のオプションに FrameSize を追加.
1.2.0	2020-09-24	AddController 時のオプション StationNo を TCP/IP 通信時も指定可能に変更.
1.2.1	2021-12-21	エラー誤検知を修正しました.
1.2.2	2022-03-18	<ul style="list-style-type: none"> •インデックスレジスタにアクセスできない問題を修正しました. •Raw コマンドを追加しました. •@PLC_STATUS および@PLC_ERROR でメモリ破壊が発生するバグを修正しました. •Elem オプションに大きな値を指定するとメモリ破壊が発生するバグを修正しました.
1.2.3	2023-02-07	<ul style="list-style-type: none"> •通信異常発生時にメモリ破壊が発生するバグを修正しました. •AddController の CONN オプションで使用できるパラメータを追記しました.

【動作確認機器】

機種	注意事項
FP7	TCP/IP, シリアル接続に対応. MEWTOCOL-COM 通信コマンドのフォーマット制限により, アクセスできないデバイスやシステム変数があります. タイマ設定値(TS), タイマ経過値(TE)は下位 16 ビットの読み出し/書き込みとなります.
ABXC32T	

目次

1. はじめに.....	4
2. プロバイダの概要	5
2.1. 概要	5
2.2. メソッド・プロパティ	6
2.2.1. CaoWorkspace::AddController メソッド.....	6
2.2.1.1. Conn オプション	7
2.2.1.2. サンプルプログラム	8
2.2.2. CaoController::GetVariableNames プロパティ.....	9
2.2.3. CaoController::AddVariable メソッド.....	9
2.2.3.1. DeviceType オプション	10
2.2.3.2. Address オプション	10
2.2.3.3. ランダム読出し/書込み	10
2.2.3.4. VT オプション	11
2.2.3.5. サンプルプログラム	12
2.2.4. CaoController::Execute メソッド	13
2.2.4.1. Raw	13
2.2.5. CaoVariable::get_Value プロパティ.....	14
2.2.6. CaoVariable::put_Value プロパティ.....	14
2.3. 変数一覧.....	15
2.3.1. CaoController クラス.....	15
2.4. エラーコード.....	17

1. はじめに

本書は、Panasonic 製 PLC に対しデータの書込み/読出しを行う CAO プロバイダのユーザーズガイドです。本書で扱う CAO プロバイダ(CaoProvPanasonicMEWTOCOL-COM.dll)を MEWTOCOL-COM プロバイダと呼びます。

第 2 章に MEWTOCOL-COM プロバイダの概要、変数やコマンドの詳細を記載しています。

MEWTOCOL-COM プロバイダで実装している通信コマンドの対応状況及びデータ列については、通信先となる PLC に依存します。通信の詳細については Panasonic の PLC の取扱説明書の「MEWTOCOL 通信手順」を参照ください。

2. プロバイダの概要

2.1. 概要

MEWTOCOL-COM プロバイダは、Panasonic 製 PLC に対して TCP/IP またはシリアル接続し、コンピュータリンク機能を用いてデータの書込み/読出しを行う CAO プロバイダです。

そのファイル形式は DLL(Dynamic Link Library)であり、CAO エンジンから使用時に動的ロードされます。MEWTOCOL-COM プロバイダを使用するにあたっては ORiN2SDK をインストールするか、下表を参照して手作業でレジストリ登録を行う必要があります。

表 2-1 MEWTOCOL-COM プロバイダ

ファイル名	CaoProvPanasonicMEWTOCOL-COM.dll
ProgID	CaoProv.Panasonic.MEWTOCOL-COM
レジストリ登録	regsvr32 CaoProvPanasonicMEWTOCOL-COM.dll
レジストリ登録の抹消	regsvr32 /u CaoProvPanasonicMEWTOCOL-COM.dll

2.2. メソッド・プロパティ

2.2.1. CaoWorkspace::AddController メソッド

MEWTOCOL-COM プロバイダは AddController 時に通信用の接続パラメータを参照し、通信の接続を行います。



AddController (<bstrCtrlName:BSTRT>,<bstrProvName:BSTRT>,
<bstrPcName:BSTRT>,<bstrOption:BSTRT>)

<bstrCtrlName> : [in] コントローラ名
 <bstrProvName> : [in] プロバイダ名. 固定値
 ="CaoProv.Panasonic.MEWTOCOL-COM"
 <bstrPcName> : [in] プロバイダの実行マシン名 (未使用)
 <bstrOption> : [in] オプション文字列

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

オプション	意味
Conn=<接続パラメータ>	必須. 通信形態とその接続パラメータを設定します. (参照 2.2.1.1)
BlockCheck[=<TRUE/FALSE>]	水平パリティチェックの有効/無効を指定します. (デフォルト:TRUE)
StationNo[=<局番>]	局番指定. 任意の 2 文字を指定します ¹ . (デフォルト:01)
FrameSize[=<フレームサイズ>]	1 フレームで送信できる最大文字数を指定します ² . 0:118 文字, 1:2048 文字 (デフォルト:1)
ConnTimeout[=<タイムアウト時間>]	接続時のタイムアウト時間(ミリ秒)を指定します. (デフォルト:3000)
Timeout[=<タイムアウト時間>]	TCP 通信におけるタイムアウト時間(ミリ秒)を指定します. (デフォルト:3000)

¹ 局番として指定できる文字は接続する機器の仕様により異なります(FP7 の場合は 01~99 の 10 進数文字). 詳しくは接続する機器の取り扱い説明書をご確認ください.

² 接続する機器によって拡張サイズ(2048 文字)を使用できない場合があります.

2.2.1.1. Conn オプション

以下に Conn オプションの接続パラメータ文字列を示します。ここで角括弧(“[]”)内は省略可能を示します。また、各パラメータの解説中の下線部はオプションを指定しなかったときのデフォルト値を示します。

【シリアルデバイス】

“Conn=COM:<COM Port>[[[:<BaudRate>]:<Parity>:<DataBit>:<StopBit>]:<FlowControl>]”

- <COM Port> : COM ポート番号. ‘1’-COM1, ‘2’-COM2, ...
- <BaudRate> : 通信速度.
300, 600, 1200, 2400, 4800, 9600,
19200, 38400, 57600, 115200, 230400.
- <Parity> : パリティ.
‘N’-パリティなし, ‘O’-奇数パリティ, ‘E’-偶数パリティ
- <DataBit> : データビット.
7, 8
- <StopBit> : ストップビット.
1, 2
- <FlowControl> : フロー制御.
‘0’-フロー制御なし, ‘2’-ハードウェア制御

【Ethernet デバイス】

TCP/IP

“Conn=TCP:<Dest IP Address>:<Dest Port No>”

- <Dest IP Address> : 接続先の IP アドレス.
- <Dest Port No> : 接続先のポート番号.

2.2.1.2. サンプルプログラム

AddController のサンプルプログラムを以下に示します。

List 2-1

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

// CaoEngine の生成
hr = CoCreateInstance(CLSID_CaoEngine,
                    NULL,
                    CLSCTX_LOCAL_SERVER,
                    IID_ICaoEngine,
                    (void **)&Eng);
if (FAILED(hr)) {
    goto EndProc;
}

// CaoWorkspace コレクションの取得
hr = pEng->get_Workspaces(&pWss);
if (FAILED(hr)) {
    goto EndProc;
}

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

// CaoController の生成
hr = pWs->AddController(CComBSTR(L"MEWTOCOL_SAMPLE"),
                      CComBSTR(L"GaoProv. Panasonic. MEWTOCOL-COM"),
                      CComBSTR(L""),
                      CComBSTR(L"Conn=TCP:192.168.1.5:60001"),
                      &pCtrl);
if (FAILED(hr)) {
    goto EndProc;
}

// ここで変数の生成, 値の設定/取得などを行う

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

2.2.2. CaoController::GetVariableNames プロパティ

表 2-8 に示しているシステム変数名の一覧を取得します。

2.2.3. CaoController::AddVariable メソッド

PLC 内の各デバイスに対しデータの書込み/読出しを行うための変数オブジェクトを作成します。



AddVariable (<bstrVariableName:BSTR>, [<bstrOption:BSTR>])

<bstrVariableName> : [in] 変数名

<bstrOption> : [in] オプション文字列

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 CaoController::AddVariable のオプション文字列

オプション ³⁾	意味
DeviceType=<デバイス種別>	必須. アクセス先のデバイス種別をコードで指定します. (参照 2.2.3.1)
Address=<デバイス番号>	必須. アクセス先のデバイスの先頭アドレスを指定します. (参照 2.2.3.2) ビットデバイスでは複数アドレスの指定が可能です. (参照 2.2.3.3)
[VT=<データ型>]	Put/Get するデータ型を指定します. (参照 2.2.3.4)
[Elem=<要素数>]	Put/Get するデータの要素数を 10 進数で指定します. ビット単位のアクセスの場合, 上限は 8 となります. (デフォルト:1)
[Array=<TRUE/FALSE>]	1 要素の読み込み時に配列型にするかどうかを指定します. (デフォルト:FALSE)

³⁾ 角括弧("[]")内は省略可能を示します。

2.2.3.1. DeviceType オプション

アクセス対象のデバイスをデバイスコードで指定します。指定可能なデバイスコードは表 2-4 の通りです。外部入力、タイマ、カウンタは読出しのみ、書込みを行うことはできません。

表 2-4 デバイス一覧

デバイス名	デバイスコード	アドレス種別	アドレス指定方法
外部入力	X	1 ビット	ビット単位(1 点)でのアクセス時: 上位 3 桁が 10 進数, 最下位が 16 進数
外部出力	Y		
内部リレー	R		
リンクリレー	L		ワード単位(16 点)でのアクセス時: 10 進数
タイマ	T		
カウンタ ⁴	C		
データレジスタ	DT	16 ビット	10 進数
リンクレジスタ	LD		
システムデータレジスタ	SD		
インデックスレジスタ	I	16 ビット	16 進数
タイマ設定値レジスタ ⁵	TS		10 進数
タイマ経過値レジスタ ⁵	TE		

2.2.3.2. Address オプション

アドレスはデバイス種別や読書きする点数により指定方法が異なります。指定方法は表 2-4 の「アドレス指定方法」を参照してください。

(例) X1F(ビット単位でのアクセス) DeviceType=X,Address=1F
 Y100(ワード単位でのアクセス) DeviceType=Y,Address=100,VT=I2
 DT250 DeviceType=DT,Address=250

2.2.3.3. ランダム読出し/書込み

ビットデバイス(X,Y,R,L,T,C)は Address オプションにコロン(:)区切りで複数のアドレスを指定することで、連続していないアドレスにアクセスすることが可能です。DeviceType は 1 種類のみ指定でき、その他の VT, Elem, Array オプションは無視されます。

Address オプションで指定したアドレス数分の配列(VT_UII|VT_ARRAY)で読出し/書込みします。指定できるアドレスは 1~8 個です。

(例) DeviceType=X,Address=0:2:7 X0, X2, X7 にアクセスします。

⁴ カウンタは読出し専用です。

⁵ FP7 ではタイマ設定値レジスタ、タイマ経過値レジスタは下位 16 ビットを読出し/書込み可能です。上位 16 ビットは"H0000"が書き込まれません。

2.2.3.4. VT オプション

読み書きするデータ型と一要素あたりの点数を指定します (1 点 = 1bit) . 指定は VT 文字列または対応する VARTYPE の値(10 進数数値)のいずれかで行います.

I4,UI4,R4 を指定した場合は連続した 32 ビット(ビットアドレスでは 32 要素, 16 ビットアドレスでは 2 要素)を 1 要素として扱います.

また, R8 を指定した場合は連続した 64 ビット(ビットアドレスでは 64 要素, 16 ビットアドレスでは 4 要素)を 1 要素として扱います.

表 2-5 データ型一覧

VT	データ型	点数/要素数	説明
BIT	VT_UI1	1 点	データを 0/1 の 2 値に変換して読み書きします。 ビットデバイス(X, Y, R, L, T, C)にのみ指定できます。
BOOL	VT_BOOL	1 点	データを 0/1 の 2 値に変換して読み書きします。 ビットデバイス(X, Y, R, L, T, C)にのみ指定できます。
BSTR	VT_BSTR	8 点	Elem バイトの ASCII として読み書きします。 Elem オプションで指定された要素数より短い文字列を書込み値に指定した場合, 不足分を 0 埋めします。
I1	VT_I1	8 点	1 バイトデータ(符号あり)として読み書きします。 Elem オプションで奇数個の要素数を指定した場合, 偶数個の要素として扱い, 追加した 8 点分は 0 埋めして書込みます。
I2	VT_I2	16 点	2 バイトデータ(符号あり)として読み書きします。
I4	VT_I4	32 点	4 バイトデータ(符号あり)として読み書きします。
UI1	VT_UI1	8 点	1 バイトデータ(符号なし)として読み書きします。 Elem オプションで奇数個の要素数を指定した場合, 偶数個の要素として扱い, 追加した 8 点分は 0 埋めして書込みます。
UI2	VT_UI2	16 点	2 バイトデータ(符号なし)として読み書きします。
UI4	VT_UI4	32 点	4 バイトデータ(符号なし)として読み書きします。
R4	VT_R4	32 点	4 バイトデータ(浮動小数)として読み書きします。
R8	VT_R8	64 点	8 バイトデータ(倍精度浮動小数)として読み書きします。

VT オプション省略時は, デバイスのアドレス種別に合わせたデータ型を使用します.

表 2-6 未指定時のデータ型

アドレス種別	VT	データ型
ビット	BIT	VT_UI1
16 ビット	UI2	VT_UI2

2.2.3.5. サンプルプログラム

AddVariable のサンプルプログラムを以下に示します。

例) アドレス"DT0100"に配列(要素数 10)で値を設定する

List 2-2

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

// CaoEngine の生成
hr = CoCreateInstance(CLSID_CaoEngine,
                    NULL,
                    CLSCTX_LOCAL_SERVER,
                    IID_ICaoEngine,
                    (void **)&Eng);
if (FAILED(hr)) {
    goto EndProc;
}

// CaoWorkspace コレクションの取得
hr = pEng->get_Workspaces(&pWss);
if (FAILED(hr)) {
    goto EndProc;
}

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

// CaoController の生成
hr = pWs->AddController(CComBSTR(L"MEWTOCOL_SAMPLE"),
                      CComBSTR(L"CaoProv. Panasonic. MEWTOCOL-COM"),
                      CComBSTR(L""),
                      CComBSTR(L"Conn=TCP:192.168.1.5:60001"),
                      &pCtrl);
if (FAILED(hr)) {
    goto EndProc;
}

// 変数の生成
hr = pCtrl->AddVariable(CComBSTR(L"A"),
                      CComBSTR(L"DeviceType=DT, Address=100, Elem=10"),
                      &pVar);
if (FAILED(hr)) {
    goto EndProc;
}
```

```
// 値の設定, 取得
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();
```

2.2.4. CaoController::Execute メソッド

Controller にて拡張メソッドを実行します。

引数には、メソッドに対して必要な引数を指定します。

以下に Execute の仕様を示します。

書式

Execute

```
(
    "<メソッド名>",      'メソッド名
    "<引数>"           '引数
)
```

以下に、Execute で指定できるメソッド一覧を示します。使用例は各メソッドの詳細で記述しています。

表 2-7 CaoController::Execute メソッド一覧

メソッド名	説明
Raw	MEWTOCOL-COM プロトコルの伝文を直接送信して返信を受信します。

2.2.4.1. Raw

MEWTOCOL-COM プロトコルの伝文を直接送信して返信を受信します。

引数には MEWTOCOL-COM プロトコルのヘッダ ("%もしくは"<") から、BCC の直前までの伝文を VT_BSTR で指定してください。

レスポンスが正常だった場合は戻り値としてレスポンスの正常レスポンス("\$")の直後から BCC の直前までが VT_BSTR として戻ります。

レスポンスが複数フレームに分割されていた場合、分割されたフレームを連結したものが戻ります。
通信に失敗した場合もしくはレスポンスが異常だった場合は 2.4 に説明されるエラーが戻ります。

使用例

```
string response = ctrl.Execute("Raw", "<01#RDD00010002");
```

2.2.5. CaoVariable::get_Value プロパティ

アクセス対象のデバイスからオプションで指定されたサイズになるように読出しを行うコマンドを送出します。
読み出した結果をオプションで指定されたデータ型に変換して返します。

2.2.6. CaoVariable::put_Value プロパティ

引数で渡された値をオプション指定に従い変換した後、アクセス対象のデバイスに対し書込みを行うコマンドを送出します。

2.3. 変数一覧

2.3.1. CaoController クラス

表 2-8 CaoController クラス システム変数一覧

変数名	データ型	説明	属性	
			get	put
@MAKER_NAME	VT_BSTR	メーカー名="Panasonic"を返す.	○	-
@VERSION	VT_BSTR	プロバイダバージョンを返す.	○	-
@PLC_MODEL	VT_BSTR	PC ステータスリードコマンドを送出し, その結果のうち機種コードとバージョンを返す.	○	-
@PLC_MODE	VT_I2	get : PC ステータスリードコマンドを送出し, PROGRAM モードか RUN モードかを返す. put: リモートコントロールコマンドを送出し, PLCの動作モードを切り替える. 0: PROGRAM モード, 1: RUN モード	○	○
@PLC_STATUS	VT_ARRAY VT_I2	PC ステータスリードコマンドを送出し, その結果のうち動作モードに関して 0, 1 の配列で返す. 配列の要素番号と各モードの対応は以下の通り. 0 : 動作モード 0: PROGRAM モード, 1: RUN モード 1 : テストモード 0: 通常運転, 1: テストモード 2 : BRK 命令・1 ステップ実行 0: 通常運転, 1: BRK/1 ステップ実行時 3 : BRK 命令実行許可 0: BRK 命令無効, 1: BRK 命令実行許可 4 : 外部出力許可 0: 外部出力しない, 1: 外部出力する 5 : 1 ステップ実行許可 0: 通常運転, 1: 1 ステップ実行可能	○	-

		6 : MSG 命令実行 1:メッセージ表示時 7 : 動作モード 1:REMOTE		
@PLC_ERROR	VT_ARRAY VT_I2	PCステータスリードコマンドを送出し, その結果のうちエラーフラグに関して正常/異常(0:正常, 1:異常)の配列で返す. 配列の要素番号と各エラーの対応は以下の通り. 0 : 自己診断エラー 1 : 瞬時停電検出 2 : ヒューズ断検知 3 : 高機能ユニット異常 4 : I/O 照合異常 5 : 電池異常 6 : 電池異常保持 7 : 演算エラーフラグ	○	-

表 2-9 CaoController クラス ユーザ変数一覧

変数名	データ型	説明	属性	
			get	put
任意	変数型依存	PLC 内のデバイスへアクセスする.	○	○

※外部入力, タイマ, カウンタデバイスは読出しのみ

2.4. エラーコード

MEWTOCOL-COM プロバイダでは、以下の固有のエラーコードを返します。

表 2-10 固有エラーコード

エラー名	エラー番号	説明
応答なし	0x80100000	PLC からデータが受信できない場合に返します。 PLC の通信設定と接続オプションが一致しているか確認してください。
受信データ異常	0x80100001	データが欠損しているなど、想定外のデータを受信した場合に返します。
ブロックチェックエラー	0x80100002	受信データのブロックチェックコードが一致しない場合に返します。
書込みデータ異常	0x80100003	ランダムアクセスにおいてアクセス数とデータ数が一致しない場合に返します。
インデックスレジスタ型異常	0x80100004	インデックスレジスタで VT オプションに I2, UI2, I4, UI4 が指定されました。
エラー応答	0x801001XX	コマンドの応答結果としてエラーを受信した場合に返します。 PLC から受信した 16 進数のエラーコード ⁶⁾ が XX に挿入されます。 例) 22 → 0x80100122

⁶⁾ エラーコードの詳細に関しては MEWTOCOL 通信仕様説明書を参照してください。