

株式会社 横河電機
YOKOGAWA PCLinkASCII プロバイダ

Version 1.0.2

ユーザーズ ガイド

August 08, 2022

備考：

【改版履歴】

バージョン	日付	内容
1.0.0	2016-09-28	初版.
	2016-09-12	非対応の変数を削除しました.
1.0.1	2019-08-23	内部ロジックの修正しました.
1.0.2	2022-08-08	内部ロジックの修正しました.

【動作確認機種】

機種	バージョン	注意事項
YOKOGAQA F3LE01	01	
	-	

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. CaoWork space::AddController メソッド	6
2.2.1.1. Conn オプション	6
2.2.2. CaoController::AddExtension メソッド	7
2.2.3. CaoController::AddVariable メソッド	7
2.2.4. CaoController::get_VariableNames プロパティ.....	8
2.2.5. CaoController::Execute メソッド	8
2.2.6. CaoExtension::AddVariable メソッド	8
2.2.7. CaoExtension::get_VariableNames プロパティ.....	8
2.2.8. CaoVariable::get_Value プロパティ	8
2.2.9. CaoVariable::put_Value プロパティ	8
2.3. 変数一覧.....	9
2.3.1. CaoController.....	9
2.3.2. CaoExtension.....	9
2.3.2.1. DV<??>.....	9
2.3.2.2. SP<??>.....	10
2.3.2.3. PG<??>.....	11
2.3.2.4. PB<??>.....	11
2.3.2.5. IF<??>.....	12
2.3.2.6. MN<??>.....	13
2.3.2.7. UL<??>.....	13
2.4. エラーコード.....	14
3. サンプルプログラム	15

1. はじめに

本書は、横河電機製 PLC から上位リンク機能を用いてデータを取得する ORiN プロバイダ (YOKOGAWA PCLinkASCII プロバイダ) のユーザーズガイドです。

このプロバイダを用いれば、横河製 PLC の上位リンク機能を使用することができます。なお、本プロバイダは ASCII コードによる通信のみに対応しています。

本書は、この YOKOGAWA PCLink プロバイダの機能と、実装されているメソッドについて説明します。

2. プロバイダの概要

2.1. 概要

YOKOGAWA PCLinkASCII プロバイダは、横河電機製 PLC から上位リンク機能(TCP/IP 通信)を利用してデータを取得します。本プロバイダのファイル形式は DLL (Dynamic Link Library) となっており、その詳細は表 2-1 のようになっています。

表 2-1 YOKOGAWA PCLinkASCII プロバイダ

ファイル名	CaoProvYOKOGAWAPCLinkASCII.dll
ProgID	CaoProv.YOKOGAWA.PCLinkASCII
レジストリ登録	regsvr32 CaoProvYOKOGAWAPCLinkASCII.dll
レジストリ登録の抹消	regsvr32 /u CaoProvYOKOGAWAPCLinkASCII.dll

図 2-1 にプロバイダの各クラスと F3LE01 の対応図を示します。

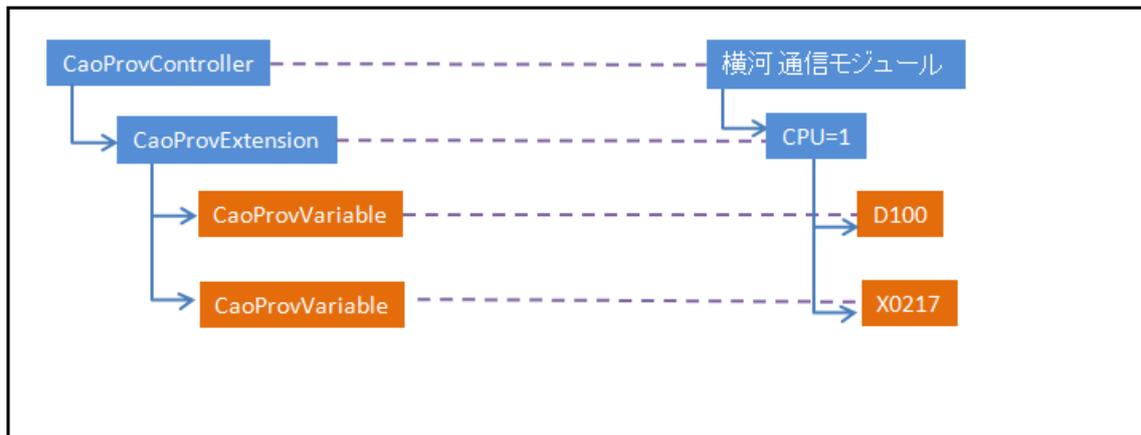


図 2-1 プロバイダの各クラスと F3LE01 の対応図

2.2. メソッド・プロパティ

2.2.1. CaoWorkspace::AddController メソッド

本プロバイダでは、CaoWorkspace::AddController で接続パラメータを設定し、通信の接続を行います。以下に、AddController の仕様を示します。

書式

```
AddController
(
    <bstrCtrlName:BSTRT>,          // [in] コントローラ名(一意な名前)
    <bstrProvName:BSTRT>,         // [in] プロバイダ名(固定 = "CaoProv. YOKOGAWA. PCLink")
    <bstrPcName:BSTRT>            // [in] プロバイダ実行マシン名(未使用)
    [, bstrOption:BSTRT]         // [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文字列を示します。

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

オプション	必須	説明	値範囲	デフォルト値
Conn=<接続オプション>	○	接続するデバイスの IP アドレスを指定します。指定方法については、2.2.1.1 を参照してください。	----	通信先 IP: -- 通信先ポート: 12289 自 IP: -- 自ポート: --
Timeout=<応答待機時間>	--	応答待機時間を指定します (ms)。	1 - 65535	500
Delay=<遅延時間>	--	コマンド間のディレイ時間を指定します (ms)。	0 - 2147483647	10

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。ここで中括弧("[]")内は省略可能なことを、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時のデフォルト値をそれぞれ示します。

本プロバイダでは ETH と指定すると TCP 接続を行います。

Eth

"Conn=Eth: <接続先 IP>[:<接続先ポート>[:<ローカル IP>[:<ローカルポート>]]]"

<接続先 IP > : 接続先 IP アドレス。 "192.168.0.1" etc..

<接続先ポート > : 接続先ポート番号。 12289

<ローカル IP > : ローカル IP アドレス。

<ローカルポート > : ローカルポート番号。

2.2.2. CaoController::AddExtension メソッド

CaoController から CaoExtension オブジェクトを生成します。以下に AddExtension の仕様を示します。

書式

```
AddExtension
(
    <bstrExtensionName:BSTR>           // [in] 変数名(任意)
    [, <bstrOption:BSTR>])           // [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文字列を示します。

オプション	必須	説明	値範囲	デフォルト値
CPU=<CPU 番号>	○	追加する CaoExtension に対応する、CPU 番号を指定します。範囲外の値を指定するとエラーとなります。	1 - 4	--

2.2.3. CaoController::AddVariable メソッド

CaoController から CaoVariable オブジェクトを生成します。変数名には表 2-3 に列挙された変数のみ使用できます。

以下に、AddVariable の仕様を示します。

書式

```
AddVariable
(
    <bstrVariableName:BSTR>           // [in] 変数名
    [, <bstrOption:BSTR>])           // [in] オプション文字列
)
```

2.2.4. CaoController::get_VariableNames プロパティ

表 2-3 の変数名リストを取得します。

2.2.5. CaoController::Execute メソッド

本プロバイダでは、“SendCommand”のみコマンドとして使用できます。SendCommand は、vntParam(第二引数)に指定された文字列をそのままデバイスに送信し、受信した文字列を pVal(第三引数)に戻します。

書式

Execute

(

```
<bstrCommand:BSTR>           // [in] コマンド名(固定 = “SendCommand”)  
[, <vntParam:VARIANT>]       // [in] パラメータ(デバイスへの送信文字列)  
[, <pVal:VARIANT>]]          // [in] 実行結果(デバイスからの受信文字列)
```

)

2.2.6. CaoExtension::AddVariable メソッド

CaoExtension から CaoVariable オブジェクトを生成します。変数名には表 2-4 に列挙された変数のみ使用できます。使用方法は、[CaoController::AddVariable メソッド](#)と同じです。

2.2.7. CaoExtension::get_VariableNames プロパティ

表 2-4 の変数名リストを取得します。

2.2.8. CaoVariable::get_Value プロパティ

オブジェクトに対応している変数の値を取得します。

変数の実装状況およびデータ型は「変数一覧」を参照して下さい。

2.2.9. CaoVariable::put_Value プロパティ

オブジェクトに対応している変数に値を設定します。

変数の実装状況およびデータ型は「変数一覧」を参照して下さい。

2.3. 変数一覧

2.3.1. CaoController

表 2-3 に CaoController クラスで使用できる変数一覧を記述します。

表 2-3 CaoController クラス 変数一覧

変数名	取得値のデータ型	説明	属性	
			get	put
@MAKERNAME	VT_BSTR	製造元名を返します。	○	-
@VERSION	VT_BSTR	プロバイダのバージョンを返します。	○	-

2.3.2. CaoExtension

表 2-4 に CaoExtension クラスで使用できる変数一覧を記述します。全ての変数で get 操作のみが可能です。

表 2-4 CaoExtension クラス 変数一覧

変数名	説明	属性		詳細
		get	put	
DV<??>	シーケンスデバイスからデータを取得/設定します。	○	○	DV<??>参照
SP<??>	特殊モジュールからデータを取得/設定します。	○	○	SP<??>参照
PG<??>	プログラム情報を取得します。	○	-	PG<??>参照
PB<??>	プログラムブロック情報を取得します。	○	-	PB<??>参照
IF<??>	各種情報を取得します。	○	-	IF<??>参照
MN<??>	モジュール名を取得します。	○	-	MN<??>参照
UL<??>	ユーザーログを取得します。	○	-	UL<??>参照

2.3.2.1. DV<??>

シーケンスデバイスからデータを取得/設定する変数です。DV 以降に任意の文字列を指定して変数を作成してください。表 2-5 に指定できるオプションを列挙します。

表 2-5 DV<>オプション

オプション	必須	説明	値範囲	デフォルト値
Device=<デバイス名>	○	読みだすデバイス名を指定します。 読み出し可能デバイスに関しては、横河の取扱説明書をご覧ください。	--	--
UserVarWidth=<ビット長>	--	1 点分の読み出し長を指定します。指定できるビット長は読みだすデバイスにより異なります。 指定されたビット長が値範囲外の場合はデフォルト値で動作します。	リンク： 1/16/32 レジスタ： 16/32 タイマ： 16	リンクデバイス： 1 レジスタデバイス： 16 タイマ： 16
Elem=<読み出し/書き込み点数>	--	読みだす点数を指定します。指定された点数が値範囲外の場合、デフォルト値で動作します。	1 - 65535	1
Array=<配列有無>	--	Elem オプションに 1 を指定した際に、配列で取得するかどうかを指定します。	True/False	False

取得/設定のデータ型:

UserVarWidth オプションにより異なりますが, Elem オプションで指定された要素数を持つ VT_VARIANT となります。

1. 1 ビット指定の場合
ON を 1, OFF を 0 とした VT_UI1 | VT_VARIANT を返します。
2. 16 ビット指定の場合
VT_UI2 | VT_VARIANT を返します。
3. 32 ビット指定の場合
VT_UI4 | VT_VARIANT を返します。

2.3.2.2. SP<??>

特殊モジュールからデータを取得/設定する変数です。SP 以降に任意の文字列を指定して変数を作成してください。表 2-6 に指定できるオプションを記述します。

表 2-6 SP<> オプション

オプション	必須	説明	値範囲	デフォルト値
Unit=<ユニット番号>	○	特殊モジュールのユニット番号を指定します。範囲外の値を指定するとエラーとなります。	0 - 7	--
Slot=<スロット番号>	○	Unit オプションで指定したユニットのスロット番号を指定します。範囲外の値を指定するとエラーとなります。	1 - 16	--
Pos=<読出し/書込み開始位置>	○	読出しデータ or 書込みデータの位置を指定します。範囲外の値を指定するとエラーとなります。	0 - 9999	--
UserVarWidth=<データ長>	--	1点分の読出し or 書込み長をビット単位で指定します。指定されたビット長が値範囲外の場合はデフォルト値で動作します。	16/32	16
Elem=<読出し/書込み点数>	--	読出し or 書込み点数を指定します。指定された点数が値範囲外の場合、デフォルト値で動作します。	1 - 65535	1
Array=<配列有無>	--	Elem オプションに 1 を指定した際に、配列で取得するかどうかを指定します。	True/False	False

取得/設定のデータ型:

UserVarWidth オプションにより異なりますが, Elem オプションで指定された要素数を持つ VT_VARIANT となります。

1. 16 ビット指定の場合
VT_UI2 | VT_VARIANT
2. 32 ビット指定の場合
VT_UI4 | VT_VARIANT

2.3.2.3. PG<??>

登録されているプログラム情報を取得する変数です。PG 以降に任意の文字列を指定して変数を作成してください。表 2-7 に取得値のデータ型を記述します。

表 2-7 PG<> 取得値のデータ型

データ型	説明	値範囲
VT_ARRAY VT_VARIANT	--	--
0 VT_BSTR	プログラム名を格納します。	--
1 VT_UI4	プログラムのステップ数を格納します。	0 - 999999
2 VT_DATE	プログラムの作成日時を格納します。	--

2.3.2.4. PB<??>

あるブロックのプログラム情報を取得する変数です。PB 以降に任意の文字列を指定して変数を作成してください。表 2-8 に指定できるオプションを、表 2-9、表 2-10 にデータ型を記述します。

表 2-8 PB<>オプション

オプション	必須	説明	値範囲	デフォルト値
Block=<ブロック番号>	○	読み出し開始プログラムブロック番号を指定します。値範囲外の場合はエラーとなります。	1 - 999	--
Elem=<読み出し点数>	--	読み出しブロック番号数を指定します。値範囲外の場合、デフォルト値で動作します。	1 - 65535	1
Array=<配列有無>	--	Elem オプションに 1 を指定した際に、配列で取得するかどうかを指定します。	True/False	False

表 2-9 PB<> Elem が複数もしくは Array=True の時のデータ型

データ型	説明	値範囲
VT_ARRAY VT_VARIANT	--	--
0 VT_UI2	読み出しブロック数を格納します。 Elem オプションで指定されたブロック数と実際に取得できる要素数は異なる場合があるため、この項目を見て配列読み出したブロック数を判断してください。	0 - Elem
1 VT_ARRAY VT_VARIANT	--	--
2i VT_BSTR	ブロック名を格納します。	--
2i + 1 VT_UI4	ブロック内ステップ数を格納します。	0 - 999999

*i は、指定されたブロック番号を 0 とする値です。

表 2-10 PB<> Elem が 1 かつ Array=False の時のデータ型

データ型	説明	値範囲
0 VT_ARRAY VT_VARIANT	--	--
0 VT_BSTR	ブロック名を格納します。	--
1 VT_UI4	ブロック内ステップ数を格納します。	0 - 999999

2.3.2.5. IF<??>

各種情報を取得する変数です。IF 以降に任意の文字列を指定して変数を作成してください。表 2-11 に指定できるオプションを、表 2-12 に取得値のデータ型を記述します。

表 2-11 IF<> オプション

オプション	必須	説明	値範囲	デフォルト値
No=<情報種類 >	--	読み出す情報種類を指定します。ここで指定した値に応じて取得するデータ型が変化します。また、値範囲外の場合はデフォルト値で動作します。	1: CPU 情報読み出し 2: システム ID, CPU タイプ, エリアサイズ情報読み出し	1

表 2-12 IF<> 取得値のデータ型

No	データ型	説明	値範囲
1	VT_ARRAY VT_VARIANT	No=1 の時のデータ型	--
	0 VT_UI1	CPU ステータスを格納します。	0: Ready 1: Not Ready
1	VT_UI1	プログラムステータスを格納します。	1: 運転 2: 停止 3: デバッグ 4: ROM ライタ
2	VT_ARRAY VT_VARIANT	No=2 の時のデータ型	--
	0 VT_BSTR	システム ID を格納します。	--
	1 VT_BSTR	リビジョンを格納します。	--
	2 VT_UI1	CPU タイプを格納します。	1: シーケンス 2: BASIC
3	VT_UI2	プログラムエリアサイズを格納します。 シーケンスの場合: ステップ数 BASIC の場合: バイト数 をそれぞれ意味します。	--

2.3.2.6. MN<??>

モジュール名を取得する変数です。MN 以降に任意の文字列を指定して変数を作成してください。表 2-13 に指定できるオプションを、に取得値のデータ型を記述します。

表 2-13 MN<> オプション

オプション	必須	説明	値範囲	デフォルト値
Unit=<ユニット番号>	○	モジュール名を读出したいユニット番号を指定します。範囲外の値を指定するとエラーとなります。	0 - 7	--

表 2-14 MN<> 取得値のデータ型

データ型	説明	値範囲
VT_ARRAY VT_VARIANT	スロット 1 ~ スロット 16 までのモジュール名情報を格納します。	--
0 VT_BSTR	スロット 1 のモジュール名を格納します。	--
1 VT_UI1	スロット 1 の入出力種別を格納します。	0: 入出力リレーなし 1: 入力リレーのみ 2: 出力リレーのみ 3: 入出力あり
2 VT_UI2	スロット 1 の入出力点数を格納します。	
: :	: :	: :
44 VT_BSTR	スロット 16 のモジュール名を格納します。	
45 VT_UI1	スロット 16 の入出力種別を格納します。	
46 VT_UI2	スロット 16 の入出力点数を格納します。	

2.3.2.7. UL<??>

ユーザログを取得する変数です。UL 以降に任意の文字列を指定して変数を作成してください。表 2-15 に指定できるオプションを記述します。

表 2-15 UL<> オプション

オプション	必須	説明	値範囲	デフォルト値
Pos=<読出し位置>	○	ユーザログ読出し位置を指定します。範囲外の値を指定するとエラーとなります。	0: 最新のデータ 1 - 999: 最新から N 番目のデータ	--

UL<??>で取得されるデータ型は、VT_BSTR となります。

2.4. エラーコード

本プロバイダには、以下の独自エラーコードが存在します。（表 2-16 独自エラーコード表参照）

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。

また、デバイスから受信したエラーコードについては、そのコードを 0x80100000 でマスクした値をプロバイダのエラーコードとします。デバイスから受信したエラーコードの意味については、横河の上位リンクコマンドの仕様書を参照してください。

表 2-16 独自エラーコード表

エラー番号	説明
0x80110001	TCP / ETH以外の通信種類を指定された。
0x80110002	CPUが指定されていない。
0x80110003	CPUで指定された値が範囲外。
0x80110004	Deviceが指定されていない。
0x80110005	Deviceで指定された文字列が不正。
0x80110006	Unitが指定されていない。
0x80110007	Unitで指定された値が範囲外。
0x80110008	Slotが指定されていない。
0x80110009	Slotで指定された値が範囲外。
0x8011000A	Posが指定されていない。
0x8011000B	Posで指定された値が範囲外。
0x8011000C	Blockが指定されていない。
0x8011000D	Blockで指定された値が範囲外

3. サンプルプログラム

以下に D200 から 500 ワード分のデータを取得する簡単なサンプルを示します。

前提条件：

- ・デバイスの IP アドレスは 192.168.0.2 とする。

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
    ' コントローラーオブジェクト作成
    Set ctrl = eng.Workspaces(0).AddController(
        "YOKOGAWA", "CaoProv.YOKOGAWA.PCLinkASCII", "",
        "CONN=ETH:192.168.0.2")
    ' エクステンションオブジェクト作成
    Set ext = ctrl.AddExtension(
        "Ext00", "CPU=1")
    ' 変数オブジェクト作成
    Set var = ext.AddVariable(
        "DV00", "Device=D200, Elem=500")
End Sub

' 値の取得
Private Sub Command1_Click()
    ' 変数値の取得
    Dim d200 As Variant
    Set d200 = var.Value
End Sub
```