

KVプロバイダ

KEYENCE KVシリーズ用

Version 1.0.4

ユーザーズ ガイド

October 18, 2021

【備考】

【改版履歴】

バージョン	日付	内容
1.0.0	2016-10-31	初版.
1.0.1	2017-03-06	バグ修正:インデックスレジスタのデフォルトデータ形式が誤っていた.
1.0.2	2017-03-27	バグ修正:デバイス番号の 16 進指定が出来なかった問題に対応. Address オプションの指定方法を 10 進数数値から文字列に変更.
	2017-07-21	誤植修正 Ethernet/IP ⇒ TCP/IP
1.0.3	2017-10-06	バグ修正:大きな配列アクセス時のバグ修正.
1.0.4	2021-10-18	データ受信時の処理を修正. オプションの解析処理を修正.

【接続確認機器】

機種	バージョン	注意事項
KV-5500		内臓 TCP/IP ユニット経由で接続

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. StationNumber オプション.....	7
2.2.2. CaoController::AddVariable メソッド.....	8
2.2.2.1. DeviceType オプション.....	9
2.2.2.2. VT オプション.....	10
2.2.3. CaoVariable:put_Value プロパティ.....	11
2.2.4. CaoVariable:get_Value プロパティ.....	11
2.2.5. CaoControllere::Execute.....	11
2.3. 変数一覧.....	12
2.3.1. CaoController クラス.....	12
2.4. コマンド一覧.....	13
2.4.1. CaoController クラス.....	13
2.4.1.1. ExecuteCommand.....	13
2.5. エラーコード.....	13
付録 A. デバイス毎のデータアクセスについて.....	14

1. はじめに

本書は、KEYENCE 製 PLC に対しデータの書き込み/読み出しを行う 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(CaoProvKEYENCEKV.dll)を KV プロバイダと呼びます。

第 2 章に KV 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。

KV 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 KEYENCE の PLC の取説や各種通信インタフェースユニットの「上位リンク通信機能」を参照ください。

2. プロバイダの概要

2.1. 概要

KV プロバイダは、KEYENCE 製 PLC に対して TCP/IP 又はシリアル接続し、上位リンク通信を用いてデータの書き込み/読み出しを行う 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にロードされます。KV 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 KV プロバイダ

ファイル名	CaoProvKEYENCEKV.dll
ProgID	CaoProv.KEYENCE.KV
レジストリ登録	regsvr32 CaoProvKEYENCEKV.dll
レジストリ登録の抹消	regsvr32 /u CaoProvKEYENCEKV.dll

2.2.1.1. Conn オプション

通信形態と接続パラメータを指定します。

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。

TCP/IP デバイス

“Conn=ETH:<Dest IP Address>[:<Dest Port No>[:<Src IP Address>[:<Src Port No>]]]”

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

< Dest IP Address >	:	TCP/IP 接続先 IP アドレス. 例: “127.0.0.1”, “192.168.0.1”
<Dest Port No>	:	TCP/IP 接続ポート番号. 例: <u>8501</u> , 5006, 5007
<Src IP Address>	:	自 IP アドレス. (複数 NIC 用途) ²⁾
<Src Port No>	:	自ポート番号. (複数 NIC 用途)

シリアルデバイス

“Conn=com:<COM Port>[:<BaudRate>[:<Parity>:<DataBits>:<StopBits>]]”

<COM Port>	:	COM 番号. 例: 1, 2, 3
<BaudRate>	:	ボーレート. 例: 1200, 2400, 4800, <u>9600</u> , 19200, 38400, 57600, 115200
<Parity>	:	パリティ. ‘N’-NONE, ‘E’-EVEN, ‘O’-ODD
<DataBits>	:	データビット数. ‘7’-7bit, <u>‘8’-8bit</u> .
<StopBits>	:	ストップビット数. <u>‘1’-1bit</u> , ‘2’-2bit.

(例 1)	“com:1”	通信ポート COM1 (, 9600bps, None, 8bits, 1bit)
(例 2)	“com:2:19200”	通信ポート COM2, 19200bps (, None, 8bits, 1bit)
(例 3)	“com:3:38400:N:8:2”	通信ポート COM3, 38400bps, None, 8bits, 2bit

2.2.1.2. StationNumber オプション

StationNumber オプションを指定することでシリアル接続時に局番(0~9)を指定することが可能です。未指定時は指定なしとなります。(TCP/IP 接続時には使用しません)

指定された局番はシリアル接続確立後に送出する通信開始コマンド(CR)に付与されます。詳細は KEYENCE の “シリアルコミュニケーションユニット ユーザーズマニュアル”を参照してください。

²⁾ Conn オプションと MyIP オプションの両方で自 IP アドレスを指定するとエラーになります。利用する場合は必ずどちらか片方で指定するようにしてください。

2.2.2. CaoController::AddVariable メソッド

CaoController クラスの AddVariable メソッドは、PLC 内の各デバイスに対しデータの書き込み/読み出しを行うための変数オブジェクトを作成するためのメソッドです。



AddVariable(<bstrVariableName:VT_BSTR>[,<bstrOption:VT_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⁾	説明
DeviceType=<デバイス種別>	必須。 デバイス種別をコードで指定。 (参照 2.2.2.1)
Address[=<デバイス番号>]	デバイス番号を 10 進数文字列又は 16 進文字列で指定。 リンクリレー、ワークリレー、リンクレジスタにアクセスする場合は 16 進数文字列で、その他のデバイスへアクセスする場合は 10 進数文字列でデバイス番号を指定する必要があります。 不正な文字列を指定した場合は変数の Put/Get 時にエラー応答が返ります。 (デフォルト:0000)
VT[=<データ型>]	Put/Get するデータ型を指定。 (参照 2.2.2.2)
Elem[=<要素数>]	Put/Get するデータの要素数を 10 進数数値で指定。 (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。 (デフォルト:FALSE)

³⁾ 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になります。

2.2.2.1. DeviceType オプション

アクセス対象のデバイス種別のコードを文字列で指定します。
指定可能なデバイス毎のコードを以下に示します。⁽⁴⁾

表 2-4 デバイス毎のコード一覧

デバイス種別	コード	デフォルト データ形式
リレー	R	ビット
リンクリレー	B	ビット
内部補助リレー	MR	ビット
ラッチリレー	LR	ビット
コントロールリレー	CR	ビット
ワークリレー	VB	ビット
データメモリ	DM	2 バイト
拡張データメモリ	EM	2 バイト
ファイルレジスタ	FM	2 バイト
	ZF	2 バイト
リンクレジスタ	W	2 バイト
テンポラリデータメモリ	TM	2 バイト
インデックスレジスタ	Z	2 バイト
タイマ(現在値)	TC	4 バイト
タイマ(設定値)	TS	4 バイト
カウンタ(現在値)	CC	4 バイト
カウンタ(設定値)	CS	4 バイト
デジタルトリマ	AT	4 バイト
コントロールメモリ	CM	2 バイト
ワークメモリ	VM	2 バイト

【指定例】

リレー : DeviceType=R
データメモリ : DeviceType=DM

⁴ アクセス可能なデバイスは PLC の機種毎に異なります。対応デバイスの詳細に関しては各機種のユーザーズマニュアルを参照してください。

2.2.2.2. VT オプション

Put/Get するデータ型を指定します。

指定は VT 文字列又は対応する VARTYPE の値(10 進数数値)のいずれかでを行います。

未指定時はデバイス毎のデフォルトデータ形式に合わせた型になります。

指定可能なデータ型の一覧を以下に示します。

表 2-5 データ型一覧

VT	データ型	説明
BIT ⁵⁾	VT_UI1	データを 0/1 の 2 値に変換して書き込み/読み出し
BOOL	VT_BOOL	データを 0/1 の 2 値に変換して書き込み/読み出し
BSTR	VT_BSTR	Elem バイトの ASCII として書き込み/読み出し
I1	VT_I1	1 バイトデータ(符号あり)として書き込み/読み出し
I2	VT_I2	2 バイトデータ(符号あり)として書き込み/読み出し
I4	VT_I4	4 バイトデータ(符号あり)として書き込み/読み出し
I8	VT_I8	8 バイトデータ(符号あり)として書き込み/読み出し
UI1	VT_UI1	1 バイトデータ(符号なし)として書き込み/読み出し
UI2	VT_UI2	2 バイトデータ(符号なし)として書き込み/読み出し
UI4	VT_UI4	4 バイトデータ(符号なし)として書き込み/読み出し
UI8	VT_UI8	8 バイトデータ(符号なし)として書き込み/読み出し
R4	VT_R4	4 バイトデータ(浮動小数)として書き込み/読み出し
R8	VT_R8	8 バイトデータ(倍精度浮動小数)として書き込み/読み出し

表 2-6 未指定時のデータ型

デフォルトデータ形式	VT	データ型
ビット	BIT	VT_UI1
2 バイト	UI2	VT_UI2
4 バイト	UI4	VT_UI4

⁵ VT=BIT はリレー系デバイス(DeviceType=R, B, MR, LR, CR, VB)に対してのみ指定できます。

2.2.3. CaoVariable:put_Value プロパティ

引数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き込みを行うコマンド(連続データ書き込み:WRS)を送出します。

2.2.4. 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み出しを行うコマンド(連続データ読み出し:RDS)を送出し、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。

2.2.5. CaoControllere::Execute

CaoController クラスの Execute メソッドは、コマンドを実行するためのメソッドです。



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

< bstrCmd> : [in] コマンド名

< vntParam> : [in] パラメータ

2.3. 変数一覧

2.3.1. CaoController クラス

表 2-7 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="KEYENCE"を返す.	○	—
@VERSION	VT_BSTR	プロバイダバージョン情報を返す.	○	—
@PLC_MODEL	VT_I4	「機種の間い合わせ」コマンド(?K)を送出しその結果を返す.	○	—
@PLC_MODE	get: VT_I4 put: VT_BOOL	get: 「動作モード確認」コマンド(?M)を送出しその結果を返す. put: 「モード変更」コマンド(Mn)を送出する. TRUE=RUN モード FALSE= PROGRAM モード	○	○
@PLC_ERROR	get: VT_I4 put: -	get: 「エラー番号確認」コマンド(?E)を送出しその結果を返す. put: 「エラークリア」コマンド(ER)を送出する.	○	○

表 2-8 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.	○	○

2.4. コマンド一覧

2.4.1. CaoController クラス

表 2-9 CaoController クラス コマンド一覧

コマンド名	データ型	説明
ExecuteCommand	VT_BSTR	汎用コマンド送出.

2.4.1.1. ExecuteCommand

パラメータで渡された文字列の末尾に(CR)を付与したものをコマンドとして送出し、その結果を文字列(末尾の(CR)(LF)を取り除いたもの)として返します。

応答結果がエラー(E0 や E1)であった場合でも Execute メソッドの実行は正常として返り、"E0"や"E1"といった文字列の形式で結果が得られます。

2.5. エラーコード

KV プロバイダでは、以下の固有エラーコードが定義されています。

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。

表 2-10 固有エラーコード

エラー名	エラー番号	説明
受信データフォーマット異常	0x80100000	受信データに異常があり、正しく解析できなかった場合に返されます。
受信データ欠落	0x80100001	受信データに欠落(最小データサイズ未満等)が見られた場合に返されます。
エラー応答	0x801001XX	コマンドの応答結果としてエラーが返された場合に返されます。 エラーコード ⁶⁾ の番号が XX に挿入されます。 例) E0 → 0x80100100 E1 → 0x80100101

⁶⁾ エラーコードの詳細に関しては各機種ของผู้用者手册を参照してください。

付録A. デバイス毎のデータアクセスについて

変数オプションで指定するデータ型(VT オプション)やデータの要素数(Elem オプション)の指定によってデバイス毎に以下の様なデータアクセスになります。

■ リレー系のビットデバイス(R, B, MR, LR, CR, VB)

- BIT, BOOL, I1, UI1

「ビット」としてアクセス。(0/1)

アクセスデータ数=Elem オプション.

- I2, UI2

「10 進数 16 ビット符号なし(.U)」としてアクセス.

アクセスデータ数=Elem オプション.

- I4, UI4, R4

「10 進数 32 ビット符号なし(.D)」としてアクセス.

アクセスデータ数=Elem オプション.

- I8, UI8, R8

「10 進数 32 ビット符号なし(.D)」としてアクセス.

アクセスデータ数=Elem オプション×2.

- BSTR

「10 進数 16 ビット符号なし(.U)」としてアクセス.

文字列(1 バイト毎)の ASCII コードを数値として扱う.

Elem オプションバイトの文字として扱い、書き込む際に奇数バイトとなる場合は 0(NULL 文字)でパディングする.

アクセスデータ数=Elem オプション÷2.

- データメモリレジスタ系の 2 バイトデバイス(DM, EM, FM, ZF, W, TM, Z, CM, VM)
 - **BOOL, I1, UI1**

「10 進数 16 ビット符号なし(.U)」としてアクセス。
デバイスの下位 8 ビットに対してアクセスする。
書き込みの場合上位 8 ビットは 0 パディングされる。
アクセスデータ数=Elem オプション。
 - **I2, UI2**

「10 進数 16 ビット符号なし(.U)」としてアクセス。
アクセスデータ数=Elem オプション。
 - **I4, UI4, R4**

「10 進数 32 ビット符号なし(.D)」としてアクセス。
アクセスデータ数=Elem オプション。
 - **I8, UI8, R8**

「10 進数 32 ビット符号なし(.D)」としてアクセス。
アクセスデータ数=Elem オプション×2。
 - **BSTR**

「10 進数 16 ビット符号なし(.U)」としてアクセス。
文字列(1 バイト毎)の ASCII コードを数値として扱う。
Elem オプションバイトの文字として扱い、書き込む際に奇数バイトとなる場合は 0(NULL 文字)でパディングする。
アクセスデータ数=Elem オプション÷2。

■ タイマ/カウンタ系の 4 バイトデバイス(TC, TS, CC, CS, AT)

● BOOL, I1, UI1

「10 進数 16 ビット符号なし(.U)」としてアクセス。
デバイスの下位 8 ビットに対してアクセスする。
書き込みの場合上位 24 ビットは 0 パディングされる。
アクセスデータ数=Elem オプション。

● I2, UI2

「10 進数 16 ビット符号なし(.U)」としてアクセス。
デバイスの下位 16 ビットに対してアクセスする。
書き込みの場合上位 16 ビットは 0 パディングされる。
アクセスデータ数=Elem オプション。

● I4, UI4, R4

「10 進数 32 ビット符号なし(.D)」としてアクセス。
アクセスデータ数=Elem オプション。

● I8, UI8, R8

「10 進数 32 ビット符号なし(.D)」としてアクセス。
アクセスデータ数=Elem オプション×2。

● BSTR

「10 進数 32 ビット符号なし(.D)」としてアクセス。
文字列(1 バイト毎)の ASCII コードを数値として扱う。
Elem オプションバイトの文字として扱い、書き込む際に 4 バイトの倍数に満たない場合は 0(NULL 文字)でパディングする。
アクセスデータ数=Elem オプション÷4。