

TCmini プロバイダ

芝浦機械 PLC

Version 1.1.1

ユーザーズ ガイド

August 30, 2021

【備考】

【改版履歴】

バージョン	日付	内容
1.0.0.0	2006-02-24	初版.
1.0.0.1	2010-02-12	エラーコード追加
1.0.0	2012-07-17	ドキュメントのバージョンルールを変更
1.1.0	2020-02-26	システム変数のバグ修正. プロバイダの概要を修正. Conn パラメータのデフォルト値を修正. 使用例(C#)を追加. 通信プロトコル対応表を追加.
1.1.1	2020-08-30	ドキュメント内の東芝機械の社名を芝浦機械に変更

【動作確認機器】

機種	バージョン	注意事項
TCmini		

目次

1. はじめに	4
2. プロバイダの概要	4
2.1. 概要	4
2.2. メソッド・プロパティ	6
2.2.1. CaoWorkspace::AddController メソッド	6
2.2.2. CaoController::get_VariableNames プロパティ	7
2.2.3. CaoController::AddVariable メソッド	8
2.2.4. CaoVariable::put_Value プロパティ	9
2.2.5. CaoVariable::get_Value プロパティ	9
2.3. エラーコード	10
3. サンプルプログラム(C#)	11
4. 通信プロトコルコマンド対応表	12

1. はじめに

本書は芝浦機械製の小型プログラマブルコントローラ TCmini 用の CAO プロバイダである, TCmini プロバイダのユーザーズガイドです.

TCmini プロバイダは, CAO のインタフェースを用いて TCmini のコマンドの送受信を行うプロバイダです.

2. プロバイダの概要

2.1. 概要

TCmini プロバイダは, 変数クラスでコマンドの生成及び送受信を行います. これにより, クライアントは TCmini のリレーへ値の設定, 取得を容易に操作することができます.

図 2-1 にプロバイダの各クラスと TCmini の対応を示します. CaoProvController は TCmini 本体に, CaoProvVariable は TCmini のリレーアドレスおよびデータレジスタアドレスに対応しています.

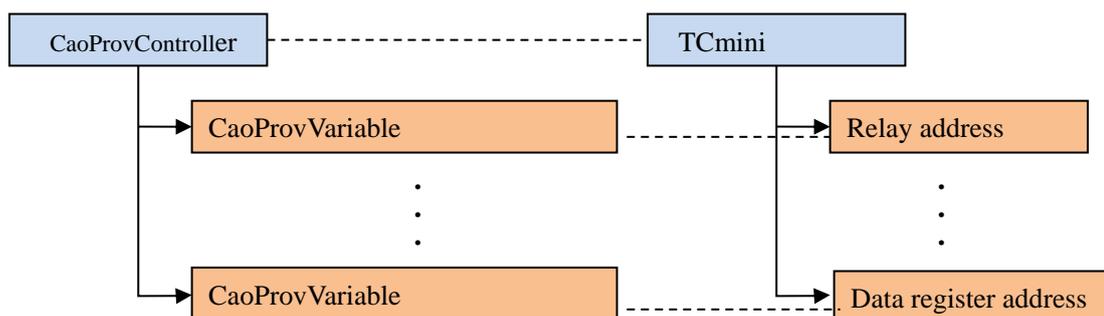


図 2-1 プロバイダの各クラスと TCmini の対応図

図 2-2 にプロバイダと TCmini との接続を示します. 接続対象とする TCmini にはシリアル接続 (RS232C)で接続してください.

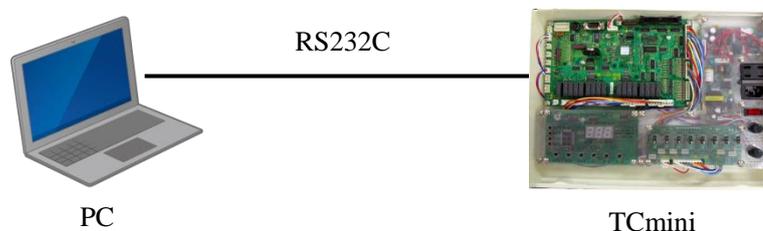


図 2-2 プロバイダと TCmini との接続図

TCmini 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 TCmini プロバイダ

ファイル名	CaoProvTCmini.dll
ProgID	CaoProv.TCmini
レジストリ登録 ¹	regsvr32 CaoProvTCmini.dll
レジストリ登録の抹消	regsvr32 /u CaoProvTCmini.dll

¹ ORiN SDK でインストールした場合は手動で登録/抹消する必要はありません。

2.2. メソッド・プロパティ

2.2.1. CaoWorkspace::AddController メソッド

TCmini プロバイダでは 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。オプション文字列で接続パラメータ、タイムアウトの設定を指定します。オプション文字列で記述するオプションを表 2-2 に示します。



```
AddController
(
    "<Controller 名>",           // コントローラ名
    "CaoProv.TCmini",          // プロバイダ名. 固定.
    "<マシン名>",              // プロバイダの実行マシン名.
    "<オプション>"             // オプション文字列
)
```

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。 詳細は 2.2.1.1 に示します。
Timeout[=<タイムアウト時間>]	送受信時のタイムアウト時間を 0~4294967295ms で指定します。 (デフォルト : 500 ms)

2.2.1.1. Conn パラメータ

以下に Parameter 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。

```
"com:<COM Port>[:<BaudRate>[:<Parity>:<DataBits>:<StopBits>[:<Flow>]]]"
<COM Port>          : COM ポート番号. '1' -COM1, '2' -COM2, ...
<BaudRate>          : 通信速度. 4800, 9600, 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, '8' -8bit.
<StopBits>          : ストップビット数. '1' -1bit, '2' -2bit.
<Flow>              : フロー制御. '0' -フロー制御なし, '1' -Xon/Xoff,
                    '2' -ハードウェア制御. (OR 演算指定可能)
```


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 のシステム変数およびユーザー変数の指定が可能です。オプション文字列は入力不要です。

書式

```
AddVariable
(
    "<変数名>"           // 変数名
    "<オプション>"      // オプション文字列(未使用)
)
```

ユーザー変数は以下のように指定できます。

書式

区分コード(1文字) + アドレス(3桁の16進数)

項目	リレー/レジスタ	指定可能な値
区分コード	リレー(1bit)	X,Y,Z,R,E,L,T,C,A
	レジスタ(16bit)	D,P,V
アドレス	リレー(1bit)	アドレスを3桁の16進数で示します。3桁目を'W','L','H'にすることで16bitまたは8bitになります。 W…16bitを意味します。 L…下位8bitを意味します。 H…上位8bitを意味します。
	レジスタ(16bit)	アドレスを3桁の16進数で示します。

- (例 1) “X100”, “Y020”, “D150”, “T12A”
- (例 2) “X10W” … X100 から X10F までの範囲
- (例 3) “x10L” … X100 から X107 までの範囲²
- (例 4) “X10H” … X108 から X10F までの範囲

また、任意の通信コマンドを変数名として指定することも可能です。get_Value 実行時に TCmini のプロトコルに従い、<ESC>+<変数名>+<CR>の電文を送信します。put_Value には対応していません。

- (例 1) “2,X100” … X100 の I/O 読出し
- (例 2) “9,X100,X200” … X100, X200 の I/O 強制リセット

² ユーザー変数名には大文字、小文字の区別はありません。すべて大文字として処理されます。

ユーザー変数のデータ型および get/put 属性の一覧を表 2-4 に示します。

表 2-4 Variable クラス ユーザー変数一覧

変数名	データ型	属性	
		get	put
データメモリ ex) “X100”, “D000”	VT_I2 (存在しないアドレス(“X999”など)を指定して get を実行すると, VT_BSTR で”パラメーター”の文字列が返ります)	○	○
通信コマンド ex) “2,X100”	VT_BSTR (不正なコマンドを実行すると, ”パラメーター”の文字列が返ります)	○	-

使用例(C#)

```
CCaoVariable varCurrentTime = ctrl.AddVariable("@CURRENT_TIME", "");
CCaoVariable varX100 = ctrl.AddVariable("X100", "");
CCaoVariable varX10W = ctrl.AddVariable("X10W", "");
CCaoVariable varD150 = ctrl.AddVariable("D150", "");
CCaoVariable varX100X200 = ctrl.AddVariable("2,X100,X200", "");
```

2.2.4. CaoVariable::put_Value プロパティ

AddVariable メソッドで作成した変数に値を設定します。

使用例(C#)

```
varX100.Value = 1;           // X100をON
varX10W.Value = -1;        // X100からX10Fまで全てON
varD150.Value = 32767;     // D150に32767をセット
```

2.2.5. CaoVariable::get_Value プロパティ

AddVariable メソッドで作成した変数から値を取得します。

使用例(C#)

```
Debug.WriteLine(varCurrentTime.Value); // 2020/01/01 12:00:00
Debug.WriteLine(varX100.Value);        // X100がOFFの場合 : 0
                                         // X100がONの場合 : 1
Debug.WriteLine(varX10W.Value);        // X100~X10Fが全てOFFの場合 : -1
```

```
Debug.WriteLine(varD150.Value); // X100~X10Fが全てONの場合 : 0
Debug.WriteLine(varX100X200.Value); // -32768~32767
// X100,X200 が OFF の場合 : 0,0
```

2.3. エラーコード

TCmini 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。

3. サンプルプログラム(C#)

TCmini の現在時刻を取得, アドレス”D000”の値を設定・取得するサンプルを示します.

```
using ORiN2.ManagedCAO;

// Create CaoEngine, CaoWorkspace, CaoController
CCaoEngine eng = new CCaoEngine();
CCaoWorkspace ws = eng.AddWorkspace("sample", "");
CCaoController ctrl = ws.AddController("ctrl1", "CaoProv.TCmini", string.Empty,
                                       "Conn=com:1,Timeout=1000");

// Create system variable(@CURRENT_TIME)
string[] VariableNames = ctrl.GetVariableNames(string.Empty);
CCaoVariable varCurrentTime = ctrl.AddVariable(VariableNames[0].ToString(), "");
Debug.WriteLine(varCurrentTime.Value);           // 2020/01/01 12:00:00

// Create user variable(D000)
CCaoVariable varD000 = ctrl.AddVariable("D000", "");
varD000.Value = 1;                               // Write 1 to D000
Debug.WriteLine(varD000.Value);                 // 1

// Remove object
if(eng != null)
{
    eng.Dispose();
}
eng = null;
ws = null;
ctrl = null;
```

4. 通信プロトコルコマンド対応表

本プロバイダで実装している Variable 変数と TCmini との通信コマンドの対応表を表 4-1 に示します。

表 4-1 通信コマンド対応表

変数名	get/put	通信コマンド
@CURRENT_TIME	get	5,D126,D125,D124,D123,D122,D121,D120
@ERROR_DESCRIPTION	get	1
○*** (○…X/Y/Z/R/E/L/T/C/A) (***…3桁の16進数)	get	2, (X/Y/Z/R/E/L/T/C/A) ***
	put	ON 状態に変更 : 8, ○*** OFF 状態に変更 : 9, ○***
○**△ (○…X/Y/Z/R/E/L/T/C/A) (**…2桁の16進数) (△…W/L/H)	get	5, ○**△
	put	10, ○**△
○*** (○…D/P/V) (***…3桁の16進数)	get	5, ○***
	put	10, ○***