

EWHA プロバイダ KOGANEI サーボハンド

Version 1.0.1

ユーザーズ ガイド

July 17, 2012

【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	2010-02-11	エラーコード追加
1.0.1	2012-07-17	ドキュメントのバージョンルールを変更

【対応機器】

機種	バージョン	注意事項

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
2.2.2. CaoController::AddVariableメソッド	7
2.2.3. CaoController::AddCommandメソッド	7
2.2.4. CaoController::Executeメソッド	7
2.2.5. CaoController::Cancelメソッド	8
2.2.6. CaoController::get_CommandNamesプロパティ	8
2.2.7. CaoController::get_VariableNamesプロパティ	8
2.2.8. CaoCommand::Executeメソッド	8
2.2.9. CaoCommand::put_Parameterプロパティ	8
2.2.10. CaoCommand::get_Parameterプロパティ	8
2.2.11. CaoCommand::get_Resultプロパティ	8
2.2.12. CaoVariable::put_IDプロパティ	8
2.2.13. CaoVariable::get_IDプロパティ	9
2.2.14. CaoVariable::put_Valueプロパティ	9
2.2.15. CaoVariable::get_Valueプロパティ	9
2.3. 変数一覧	10
2.3.1. コントローラクラス	10
2.4. エラーコード	10
3. サンプルプログラム	11

1. はじめに

本書は KOGANEI 製のサーボハンド(EWHA)用の CAO プロバイダである, EWHA プロバイダのユーザーズガイドです.

EWHA プロバイダは, サーボハンドに対して RS-232C によるコマンドの送受信を行います.

2. プロバイダの概要

2.1. 概要

EWHA プロバイダは、CaoController::Execute、CaoCommand::Execute、CaoVariable::putValue、CaoVariable::getValue で各コマンドの実行を行います。

CaoController::Execute ではコマンド文字列をそのままコマンドとして送信します。

CaoCommand::Execute ではコマンド名に対応したコマンドを実行します。

CaoVariable::putValue、CaoVariable::getValue では変数名に対応したコマンドで値を設定/取得します。

EWHA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 EWHA プロバイダ

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

EWHA 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。
このときオプションで通信形態、ターミネータの設定を指定します。

```
AddController
(
    "<コントローラ名>",           // コントローラ名
    "GaoProv. KOGANEI. EWHA",     // 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
)
```

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 1 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に示す.
Term[=<ターミネータ>]	通信時のターミネータを設定します. 1:CR (デフォルト) 2:LF 3:CR+LF
TimeOut[=<タイムアウト時間>]	通信時のタイムアウト時間を設定します. (デフォルト:500ms)

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>]]”

<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.

(例 1) “com:1” 通信ポート COM1 (, 38400bps, None, 8bits, 2bit)

(例 2) “com:2:9600” 通信ポート COM2, 9600bps (, None, 8bits, 2bit)

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

2.2.2. CaoController::AddVariableメソッド

EWHA の情報などを取得/設定する変数オブジェクトを作成します。

変数名には、表 2-3、表 2-4の変数のみ使用することができます。

2.2.3. CaoController::AddCommandメソッド

EWHA にコマンドを送信する CaoCommand を生成します。

以下にコマンド名の一覧表を示します。

```
AddCommand
(
    "<コマンド名>",    // コマンド名
    "<オプション>",    // オプション文字列
)
```

表 2-2 コマンド名一覧表²

コマンド名	EWHA のコマンド	意味	パラメータ	結果
ORG	ORG	原点復帰	-	-
MOVP	MOVP	ポイント指定移動	VT_I4:ポイント番号	-
MOVD	MOVD	座標指定移動実行	VT_R4:座標位置 VT_I4:速度	-
MOVH	MOVH	座標指定把持移動実行	VT_R4:座標位置 VT_I4:速度 VT_I4:力	-
XP	X+	+指定量移動	-	-
XM	X-	-指定量移動	-	-
XINC	XINC	+定速移動	-	-
XDEC	XDEC	-定速移動	-	-
SRVO	SRVO	モータ励磁	VT_I4: 励磁(1)/ 無励磁(0)	-

2.2.4. CaoController::Executeメソッド

パラメータに指定した文字列をそのまま送信します。実行結果が OK/READY の場合、OK/READY が来るまでに受信した文字列を文字列配列として返します。実行結果が NG/STOP の場合、NG/STOP の次に受信した文字列を文字列配列として返します。

² 各コマンドの詳細については EWHA のドキュメントを参照してください。

このメソッドでは、第 1 引数のコマンド名に“Raw”，第 2 引数のパラメータに EWHA のコマンドを指定します。このときパラメータは必ず文字列型で指定してください。

```
Execute
(
    "Raw"                // コマンド名. 固定.
    "<パラメータ>"      // EWHA コマンド文字列
)
```

2.2.5. CaoController::Cancelメソッド

キャンセルコマンド(^C)を送信します。

2.2.6. CaoController::get_CommandNamesプロパティ

コマンド名リストを取得します。取得するコマンド名は表 2-2を参照して下さい。

2.2.7. CaoController::get_VariableNamesプロパティ

変数名リストを取得します。取得する変数名は表 2-5を参照して下さい。

2.2.8. CaoCommand::Executeメソッド

コマンド名、2.2.9で設定した文字列からEWHAへのコマンドを生成し、送信します。

例えば、コマンド名“ORG”の Execute メソッドを実行した場合、原点復帰動作を行います。

2.2.9. CaoCommand::put_Parameterプロパティ

コマンド送信時のパラメータを設定します。

各コマンドのパラメータは表 2-2を参照して下さい。パラメータが複数の場合はVARIANT配列に各要素を格納するか、カンマ区切りの文字列を指定して下さい。

不正なパラメータのチェックはこのプロパティでは行いません。パラメータのチェックはコマンドの実行結果で判定されます。

2.2.10. CaoCommand::get_Parameterプロパティ

2.2.9で設定したパラメータを取得します。パラメータが設定されていないときはVT_EMPTYを返します。

2.2.11. CaoCommand::get_Resultプロパティ

最後に実行した2.2.8の実行結果を取得します。取得する内容はEWHAからのレスポンスのデータ部を文字列で返します。

2.2.12. CaoVariable::put_IDプロパティ

ポイント番号を指定します。変数名が“P”の時にのみ、使用されます。このプロパティを設定することで、指定したポイント番号の情報の取得が可能です。

2.2.13. CaoVariable::get_IDプロパティ

2.2.12で設定した値を取得します。パラメータが設定されていないときはVT_EMPTYが返ります。

2.2.14. CaoVariable::put_Valueプロパティ

EHWA の Write 系コマンドを実行します。

使用可能な変数名は表 2-5を参照して下さい。

以下に変数名とEWHA のコマンドの対応表を示します。

表 2-3 put_Value 時の変数名と EWHA コマンド名の対応表

変数名	EWHA のコマンド	備考
SRVO	SRVO	-
PNO	MOVP ORG	VT_EMPTYを指定した時はORGコマンドによる原点復帰を行います。

2.2.15. CaoVariable::get_Valueプロパティ

EHWA の Read 系コマンドを実行します。

使用可能な変数名は表 2-5を参照して下さい。

表 2-4 get_Value 時の変数名と EWHA コマンド名の対応表

変数名	EWHA のコマンド	備考
SRVO	?SRVO	-
POS	?POS	-
PNO	?PNO	-
P	?P	取得するポイント番号を ID プロパティで指定
ORG	?ORG	-
VER	?VER	-
DIO	READ DIO	-

2.3. 変数一覧

2.3.1. コントローラクラス

表 2-5 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
SRVO	VT_BSTR	モータ励起状態	○	○
POS	VT_BSTR	現在位置取得	○	-
PNO	VT_BSTR	get: ポイント番号 put: ポイント番号移動 ³	○	○
P	VT_BSTR	ポイント情報	○	-
ORG	VT_BSTR	原点復帰状態	○	-
VER	VT_BSTR	バージョン情報	○	-
DIO	VT_BSTR	I/O 状態	○	-

2.4. エラーコード

EWHA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 値が VT_EMPTY の場合、原点復帰動作を行います。

3. サンプルプログラム

EWHA の原点復帰状態の取得と原点復帰動作をさせるコードを示します。

List 3-1 Sample.frm

```
Dim eng As CaoEngine
Dim ctrl As CaoController
Dim val As CaoVariable

Private Sub Form_Load()

    ' CAO エンジンの生成
    Set eng = New CaoEngine

    ' EWHA への接続
    Set ctrl = eng.Workspaces(0).AddController("Sample", "CaoProv.KOGANEI.EWHA", "",
"Conn=com:1")

    ' 原点復帰の変数生成
    Set val = ctrl.AddVariable("ORG", "")

End Sub

Private Sub Command1_Click()

    ' 値の設定
    val = Text1.Text

End Sub

Private Sub Command2_Click()

    ' 値の取得
    Text2.Text = val

End Sub
```