WACOH
DynPick プロバイダ

Version 1.0.1

ユーザーズ ガイド

March 27, 2018

備考:

【改版履歴】

バージョン	日付	内容
1.0.0	2013-01-25	初版.
1.0.1	2013-04-11	エラーコード追加
1.0.1	2013-04-22	対応機器追加
1.0.1	2013-10-10	対応機器追加
1.0.1	2017-02-06	対応機器追加
1.0.1	2018-03-27	計測値取得処理の修正

【対応機器】

機種	バージョン	注意事項
WEF-3A200-1.5-U		3 軸力覚センサ USB タイプ 定格荷重:Fz 200N、Mx,My 1.5Nm
WEF-3A200-1.5-R		3 軸力覚センサ RS422 タイプ 定格荷重:Fz 200N、Mx,My 1.5Nm
WEF-6A100-2-UG5		6 軸力覚センサ USB タイプ 定格荷重:Fx,Fy,Fz 100N、Mx,My,Mz 2Nm
WEF-6A100-2-RG5		6 軸力覚センサ RS422 タイプ 定格荷重:Fx,Fy,Fz 100N、Mx,My,Mz 2Nm
WEF-6A200-4-UG5		6 軸力覚センサ USB タイプ 定格荷重:Fx,Fy,Fz 200N、Mx,My,Mz 4Nm
WEF-6A200-4-RG5		6 軸力覚センサ RS422 タイプ 定格荷重:Fx,Fy,Fz 200N、Mx,My,Mz 4Nm
WEF-6A1000-30-UG5		6 軸力覚センサ USB タイプ 定格荷重:Fx,Fy,Fz 1000N、Mx,My,Mz 30Nm
WEF-6A1000-30-RG5		6 軸力覚センサ RS422 タイプ 定格荷重:Fx,Fy,Fz 1000N、Mx,My,Mz 30Nm
WEF-6A200-4-EGP		6 軸力覚センサ Ethernet タイプ 定格荷重:Fx,Fy,Fz 200N、Mx,My,Mz 4Nm

WEF-6A200-4-RC5		6 軸力覚センサ RS422 タイプ(5V) 定格荷重:Fx,Fy,Fz 200N,Mx,My,Mz 4Nm
WEF-6A500-10-RC5		6 軸力覚センサ RS422 タイプ(5V) 定格荷重:Fx,Fy,Fz 500N,Mx,My,Mz 10Nm
WEF-6A1000-30-RC5-B		6 軸力覚センサ RS422 タイプ(5V) 定格荷重:Fx,Fy,Fz 1000N,Mx,My,Mz 30Nm
WEF-6A200-4-RC5-B		6 軸力覚センサ RS422 タイプ(TypeS series) (5V) 定格荷重:Fx,Fy,Fz 200N,Mx,My,Mz 4Nm
WEF-6A500-10-RC5-B		6 軸力覚センサ RS422 タイプ(TypeS series) (5V) 定格荷重:Fx,Fy,Fz 500N,Mx,My,Mz 10Nm
WGF-3A50-5-RG5		3 軸力覚センサ RS422 タイプ(5V) 定格荷重:Fz 50N、Mx,My 5Nm
WEF-6A200-4-RC24		6 軸力覚センサ RS422 タイプ(24V) 定格荷重:Fx,Fy,Fz 200N,Mx,My,Mz 4Nm
WEF-6A500-10-RC24		6 軸力覚センサ RS422 タイプ(24V) 定格荷重:Fx,Fy,Fz 500N,Mx,My,Mz 10Nm
WEF-6A1000-30-RC24-B		6 軸力覚センサ RS422 タイプ(24V) 定格荷重:Fx,Fy,Fz 1000N,Mx,My,Mz 30Nm
WEF-6A200-4-RC24-B		6 軸力覚センサ RS422 タイプ(TypeS series) (24V) 定格荷重:Fx,Fy,Fz 200N,Mx,My,Mz 4Nm
WEF-6A500-10-RC24-B		6 軸力覚センサ RS422 タイプ(TypeS series) (24V) 定格荷重:Fx,Fy,Fz 500N,Mx,My,Mz 10Nm
WGF-3A50-5-RG24		3 軸力覚センサ RS422 タイプ(24V) 定格荷重:Fz 50N、Mx,My 5Nm

目次

1. はじめに

本書は WACOH-TECH 力覚センサ DynPick シリーズ用の 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(CaoProvDynPick.dll)を DynPick プロバイダと呼びます.

次章に DynPick プロバイダの概要, 3 章にコマンドリファレンスを記載しています.

2. プロバイダの概要

2.1. 概要

DynPick プロバイダは，WACOH-TECH 力覚センサに依存する部分を吸収し CAO プロバイダ・インターフェース仕様で規定された機能を提供する CAO プロバイダです.

DynPick プロバイダのファイル形式は 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. DynPick 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 DynPick プロバイダ

ファイル名	CaoProvDynPick.dll
ProgID	CaoProv.WACOH.DynPick
レジストリ登録	regsvr32 CaoProvDynPick.dll
レジストリ登録の抹消	regsvr32 /u CaoProvDynPick.dll

DynPick プロバイダは以下の 3 種類の動作モードを持ちます.

・ 通常モード

計測データを CaoVariable::get_Value()で取得します.

処理は DynPick からの応答データがあるまで待ちます.

・ 高速モード

計測データを CaoVariable::get_Value()で取得します.

DynPick からの前回の取得データを返します. このため，データの取得間隔が長い場合は，正しくデータが取れなくなります.

・ サイクルモード

計測データを OnMessage イベントで取得します.

取得間隔は CaoController::AddController()の Interval オプションで指定します.

このモードのとき，CaoVariable::get_Value()によるデータの取得は実行できません.

2.2. メソッド・プロパティ

2.2.1. CaoWorkspace::AddController メソッド

DynPick 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，タイムアウト，OnMessage イベントでの計測周期を指定します．

書式　AddController(<bstrCtrlName:BSTR>,<bstrProvName:BSTR>,
<bstrPCName:BSTR>,<bstrOption:BSTR>)

<bstrCtrlName> : [in] コントローラ名（任意）
<bstrProvName> : [in] オプション文字列
固定値 =" CaoProv.WACOH.DynPick
<bstrPcName> [in] プロバイダの実行マシン名
<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． 詳細については 2.2.1.1 を参照してください．
[Mode=<モード>]	動作モードを指定します． 0：通常モード 1：高速モード 2：サイクルモード
[Timeout=<タイムアウト時間>]	送受信時のタイムアウト時間(ミリ秒)を指定します． (デフォルト：500) モードが高速モードのとき（Mode=1），このオプションの値は無視されます．
[Interval=<周期時間>]	OnMessage イベントによるデータ取得の周期時間(ミリ秒)を設定します． (デフォルト：1000) モードがサイクルモードでないとき（Mode≠2），このオプションの値は無視されます．

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("[]")内のパラメータは省略可

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指定を省略した時のデフォルト値を示します.

・ **TCP の場合**

"Conn=TCP:<Dest IP>[:<Dest Port>[:<Src IP>[:<Src Port>]]]"

<Dest IP> : 接続先の IP アドレス. (デフォルト:127.0.0.1)

<Dest Port> : 接続先の TCP ポート番号.（デフォルト:5001）

<Src IP> : 接続元の IP アドレス. (デフォルト:255.255.255.255)

接続元の IP アドレスに"255.255.255.255"を指定した場合は, ローカルの IP アドレスを自動で設定します.

<Src Port> : 接続元のポート番号.（デフォルト:0）

接続もとのポート番号に"0"を指定した場合は, 使用可能なポート番号を自動で設定します.

・ **UDP の場合**

"Conn=UDP:<Dest IP>[:<Dest Port>[:<Src IP>[:<Src Port>]]]"

<Dest IP> : 接続先の IP アドレス. (デフォルト:127.0.0.1)

<Dest Port> : 接続先の TCP ポート番号.（デフォルト:5001）

<Src IP> : 接続元の IP アドレス. (デフォルト:255.255.255.255)

接続元の IP アドレスに"255.255.255.255"を指定した場合は, ローカルの IP アドレスを自動で設定します.

<Src Port> : 接続元のポート番号.（デフォルト:0）

接続もとのポート番号に"0"を指定した場合は, 使用可能なポート番号を自動で設定します.

・ **RS422 の場合**

"Conn=COM:[<ComPort>"[:<BaudRate>[:<Parity>:<DataBits>:<StopBits>]]]"

<ComPort> : COM ポート番号. '1'-COM1, '2'-COM2,...(デフォルト:1)

<BaudRate> : 通信速度. (デフォルト:115200)

<ByteSize> : パリティ. 'N'-NONE, 'E'-EVEN, 'O'-ODD (デフォルト:N)

<DataBits> : データビット数. '7'-7bit, '8' – 8bit (デフォルト:8)

<StopBits> : ストップビット数. '1'-1bit, '2'-2bit (デフォルト:0)

2.2.2. CaoController::Execute メソッド

コマンドを実行します.

Execute メソッドの引数は, コマンドを BSTR,パラメータを VARIANT 配列で指定します.

各コマンドの詳細は 3.1 を参照してください.

書式 [<vntRet:VT_VARIANT>=]Execute(<bstrCmd:VT_BSTR>[,<vntParam:VT_VARIANT>])

< vntRet > : [out] コマンドの返り値

< bstrCmd > : [in] コマンド

< vntParam > : [in] パラメータ

2.2.3. CaoController::AddVariable メソッド

変数オブジェクトを生成します.

実装されているシステム変数は 2.3.1 を参照してください.

書式 AddVariable(<bstrVariableName:VT_BSTR>[,<vntOption:VT_BSTR>])

< bstrVariableName > : [in] 変数名

<bstrOption> : [in] オプション文字列

2.2.4. CaoVariable::get_Value プロパティ

変数の値を取得します.

取得する値の詳細については 2.3 を参照してください.

このプロパティは, 通常モードまたは高速モードのときのみ使用することができます.

2.2.5. CaoMessage::get_Value メソッド

メッセージに格納されている計測値を取得します.

計測値は以下の順番で配列に格納されます.

<レコード番号>, <Fx>, <Fy>, <Fz>, <Mx>, <My>, <Mz>

データ型は, long 型配列(VT_I4 | VT_ARRAY)で格納されています.

<Fx>～<Mz>にはデジタル出力値を格納します.

2.3. 変数一覧

2.3.1. コントローラクラス

表 2-3 コントローラクラス システム変数一覧

変数名	データ型	説明	属性	
			get	put
@Data	VT_I4 | VT_ARRAY	計測値 以下の順番で配列に格納されます. <レコード番号>, <Fx>, <Fy>, <Fz>, <Mx>, <My>, <Mz> <Fx>～<Mz>にはデジタル出力値を格納します.	○	-

2.4. エラーコード

DynPick 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 2-4 独自エラーコード一覧

エラー名	エラー番号	説明
E_RECV_DATA_BROKEN	0x80100001	受信データが破損していました.

3. コマンドリファレンス

本章では CaoController::Execute メソッドの各コマンドについて解説します.

3.1. コントローラクラス

表 3-1 CaoController::Execute コマンド一覧

コマンド	機能	
OffsetReset	オフセットリセット	P. 11
BufferClear	受信バッファクリア	P. 11

3.1.1. CaoController::Execute(“OffsetReset”) コマンド

DynPick シリーズのオフセットリセットを実行します. 詳細については「DYNPICK 取扱説明書」を参照してください.

書式　OffsetReset()

戻り値　：　なし

使用例

```
caoCtrl.Execute(“OffsetReset”)　‘ オフセットリセット
```

3.1.2. CaoController::Execute(“BufferClear”) コマンド

受信バッファのクリアとエラークリア処理を実行します.

書式　BufferClear()

戻り値　：　なし

使用例

```
caoCtrl.Execute(“BufferClear”)　‘ 受信バッファクリア
```