

Dai-ichiSeiko ESTORQ プロバイダ

Version 1.0.0

ユーザーズ ガイド

September 30, 2019

【備考】

【改版履歴】

バージョン	日付	内容
1.0.0	2019-09-30	初版.

【対応機器】

機種	バージョン	注意事項
ESTORQ		
ES-Gripper		

【動作確認機器】

機種	バージョン	注意事項
ES-Gripper	4.19	

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::Execute メソッド	7
2.2.2.1. CaoController::Execute("GetTorque")コマンド	7
2.2.2.2. CaoController::Execute("GetDeviceVersion")コマンド	8
2.2.2.3. CaoController::Execute("Reset")コマンド	8
2.2.3. CaoController::get_VariableNames プロパティ	8
2.2.4. CaoController::AddVariable メソッド	9
2.2.5. CaoVariable::get_Value プロパティ	9
2.3. エラーコード	10
3. サンプルコード(C#).....	11
4. 通信プロトコルコマンド対応表	12

1. はじめに

本書は第一精工株式会社製の ESTORQ, ES-Gripper(以下, ES-Gripper は記述を省略)に対してトルク値の読み込みを行う 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(CaoProvDai-ichiSeikoESTORQ.dll)を ESTORQ プロバイダと呼びます。ESTORQ プロバイダは第一精工株式会社から提供された「静電容量型トルクセンサ ESTORQ 通信仕様書 (USB, RS422)」に沿って開発しています。本書は ESTORQ 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

図 2-1 にプロバイダの各クラスと ESTORQ との対応を示します。CaoProvController は ESTORQ 本体に、CaoProvVariable はトルク値およびバージョン情報に対応しています。

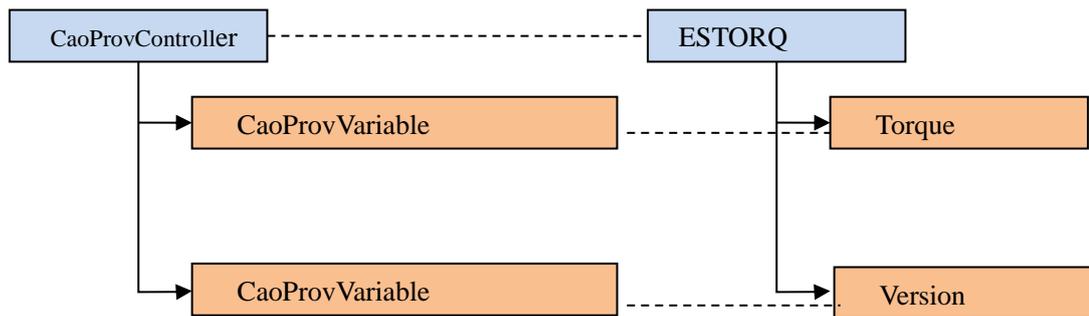


図 2-1 プロバイダの各クラスと ESTORQ の対応図

図 2-2 に ESTORQ との接続図を示します。接続対象とする ESTORQ には RS-422A で接続してください。ボーレートのデフォルト値は 307.2 kbps です。ESTORQ のボーレートを 115.2 kbps に変更する場合、第一精工株式会社様にお問い合わせください。

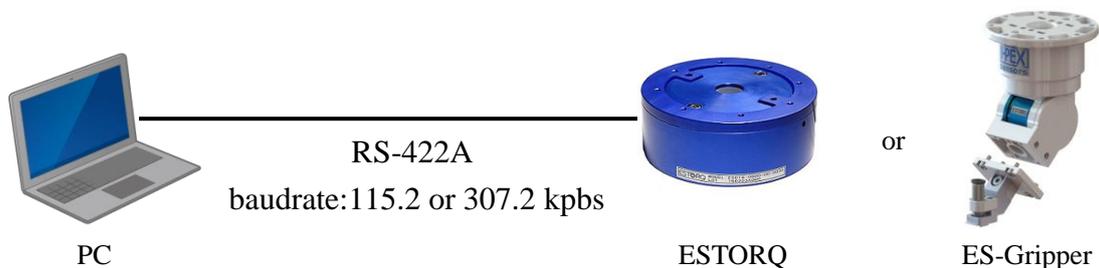


図 2-2 ESTORQ との接続図

ESTORQ 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. ESTORQ 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, 表 2-1 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 ESTORQ プロバイダのファイル形式

ファイル名	CaoProvDai-ichiSeikoESTORQ.dll
ProgID	CaoProv.Dai-ichiSeiko.ESTORQ
レジストリ登録 ¹	regsvr32 CaoProvDai-ichiSeikoESTORQ.dll
レジストリ登録の抹消	regsvr32 /u CaoProvDai-ichiSeikoESTORQ.dll

¹ ORiN2 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

本プロバイダでは、CaoWorkspace::AddController で接続パラメータを設定して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(
    BSTR bstrCtrlName,           // [in] コントローラ名(任意のコントローラ名を入力して下さい)
    BSTR bstrProvName,          // [in] プロバイダ名. 固定値 = "CaoProv.Dai-ichiSeiko.ESTORQ"
    BSTR bstrPcName,            // [in] プロバイダの実行マシン名
    BSTR bstrOption              // [in] オプション文字列
);
```

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

設定項目	オプション文字列	必須	説明
接続先	Conn	○	com:<ポート番号>[:<ボーレート>] ² の形式で記述してください。 ポート番号:1~256 ボーレート ³ [bps]: <u>115200/307200</u>
定格トルク	RatedTorque	○	接続先の ESTORQ の定格トルク Nm を指定してください。 0.0 < 定格トルク ≤ 100000.0 の範囲で指定可能です。
タイムアウト	Timeout	-	タイムアウトを 1~4294967295 ms で指定してください。 デフォルト値: 500 ms

例 1) 接続先 COM1, ボーレート 307200 bps, 定格トルク 10 Nm で接続する場合

"Conn = com:1 , RatedTorque = 10"

例 2) 接続先 COM256, ボーレート 115200 bps, 定格トルク 0.5 Nm, タイムアウト 1000 ms で接続する場合

"Conn = com:256:115200 , RatedTorque = 0.5 , Timeout=1000"

使用例(C#)

```
using ORiN2.ManagedCAO;
CCaoEngine eng = new CCaoEngine();
CCaoWorkspace ws = eng.AddWorkspace("SampleWorkspace", "");
CCaoController ctrl = ws.AddController("controller1", "CaoProv.Dai-ichiSeiko.ESTORQ",
    string.Empty, "Conn=com:1,RatedTorque=10,Timeout=1000");
```

² 角括弧("[]")内は省略可能なパラメータです。

³ 下線の値はデフォルト値です。

2.2.2. CaoController::Execute メソッド

CaoController クラスに属するプロバイダ固有の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 型、パラメータを VARIANT 型で指定します。パラメータは省略可能です。

書式

```
Execute (
    BSTR bstrCmd           // [in] コマンド名
    [,VARIANT vntParam]   // [in] パラメータ
);
```

引数	説明
bstrCmd	GetTorque, GetDeviceVersion, Reset コマンドの中から Execute メソッド名を指定してください。
vntParam	コマンド名で使用するパラメータを指定します。

2.2.2.1. CaoController::Execute("GetTorque")コマンド

現在のトルク値を取得します。

引数の型	説明
なし	-

戻り値の型	説明
VT_R8	現在のトルク値[Nm]です。AddController 時に指定した定格トルクより、 -(定格トルク × 1.1) [Nm] ~ +(定格トルク × 1.1)[Nm]の範囲で測定可能です。

使用例(C#)

```
double torque = (double)ctrl.Execute("GetTorque", "");
Debug.WriteLine(torque + "[Nm]");           // 0.00134293737028446[Nm]
```

2.2.2.2. CaoController::Execute("GetDeviceVersion")コマンド

ESTORQ のデバイスバージョンを取得します。

引数の型	説明
なし	-

戻り値の型	説明
VT_BSTR	ESTORQ のデバイスバージョンです。

使用例(C#)

```
string deviceVersion = ctrl.Execute("GetDeviceVersion", "").ToString();
Debug.WriteLine(deviceVersion); // TRQ Ver 4.19 2018/02/06
```

2.2.2.3. CaoController::Execute("Reset")コマンド

本コマンドを実行した時点のトルク値を 0 Nm に設定します。設定は電源再投入後も保持されます。

引数の型	説明
なし	-

戻り値の型	説明
なし	-

使用例(C#)

```
ctrl.Execute("Reset", "");
```

2.2.3. CaoController::get_VariableNames プロパティ

ESTORQ プロバイダで使用可能な変数リストを取得します。表 2-3 のリストが取得できます。

戻り値の型	説明
VT_ARRAY VT_BSTR	CaoController の AddVariable の変数名に使用できる変数一覧が取得できます。

使用例(C#)

```
string[] variableNmaes = ctrl.GetVariableNames(string.Empty);
Debug.WriteLine(variableNmaes[0]); // @MAKER_NAME
Debug.WriteLine(variableNmaes[1]); // @VERSION
Debug.WriteLine(variableNmaes[2]); // @TORQUE
Debug.WriteLine(variableNmaes[3]); // @DEVICE_VERSION
```

表 2-3 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「Dai-ichiSeiko」を返します.	○	-
@VERSION	VT_BSTR	プロバイダのバージョンを返します.	○	-
@TORQUE	VT_R8	現在のトルク値を返します.	○	-
@DEVICE_VERSION	VT_BSTR	ESTORQ のデバイスバージョンを返します.	○	-

2.2.4. CaoController::AddVariable メソッド

CaoController から CaoVariable オブジェクトを生成します. 表 2-3 のシステム変数一覧から変数名を入力してください. オプション文字列は入力不要です.

書式

```
AddVariable (
    BSTR bstrVarName, // [in] 変数名(システム変数名を指定)
    BSTR bstrOption   // [in] オプション文字列
);
```

使用例(C#)

```
CCaoVariable varTorque = ctrl.AddVariable("@TORQUE", "");
CCaoVariable varDeviceVersion = ctrl.AddVariable("@DEVICE_VERSION", "");
```

2.2.5. CaoVariable::get_Value プロパティ

AddVariable メソッドで作成した変数の状態を取得します. 戻り値の型は AddVariable で指定した変数名によって異なります. 各変数の戻り値は表 2-3 のシステム変数を参照ください.

使用例(C#)

```
Debug.WriteLine(varTorque.Value + "[Nm]");           // 0.00134293737028446[Nm]
Debug.WriteLine(varDeviceVersion.Value);             // TRQ Ver 4.19 2018/02/06
```

2.3. エラーコード

本プロバイダでは固有エラーコードを定義しています。固有エラーコードを表 2-4 に示します。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 固有エラーコード

エラー番号	説明
0x80110000	AddController 時に定格トルク(RatedTorque)のパラメータが不正です。 0.0 < 定格トルク ≤ 100000.0 の範囲で指定してください。
0x80110001	応答パケットの長さが不正です。ノイズが発生している可能性があるため、伝送距離を短くしてください。
0x80110002	チェックサムが不正です。ノイズが発生している可能性があるため、伝送距離を短くしてください。

3. サンプルコード(C#)

```
using ORiN2.ManagedCAO;

CCaoEngine eng = new CCaoEngine();
CCaoWorkspace ws = eng.AddWorkspace("sample", "");
CCaoController ctrl = ws.AddController("ctrl1", "CaoProv.Dai-ichiSeiko.ESTORQ", string.Empty,
                                       "Conn=com:1:115200,RatedTorque=10,Timeout=1000");

double torque = (double)ctrl.Execute("GetTorque", "");
Debug.WriteLine(torque + "[Nm]"); // 0.00134293737028446[Nm]

string deviceVersion = ctrl.Execute("GetDeviceVersion", "").ToString();
Debug.WriteLine(deviceVersion); // TRQ Ver 4.19 2018/02/06

string[] variableNmaes = ctrl.GetVariableNames(string.Empty);

CCaoVariable varTorque = ctrl.AddVariable(variableNmaes[2].ToString(), "");
CCaoVariable varDeviceVersion = ctrl.AddVariable(variableNmaes[3].ToString(), "");

Debug.WriteLine(varTorque.Value + "[Nm]"); // 0.00134293737028446[Nm]
Debug.WriteLine(varDeviceVersion.Value); // TRQ Ver 4.19 2018/02/06

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 Execute メソッド, Variable 変数と「静電容量型トルクセンサ ESTORQ 通信仕様書(USB, RS422)」の通信コマンドとの対応表を表 4-1 に示します.

表 4-1 通信コマンド対応表

Execute メソッド	変数名	get/put	通信コマンド
GetTorque	@TORQUE	get	R
GetDeviceVersion	@DEVICE_VERSION	get	V
Reset	-	put	O