

RLW プロバイダ
タカス技研 小型トルクセンサ

ユーザーズ ガイド

Version 1.1.0

April 2, 2015

【備考】

【改版履歴】

日付	版数	内容
2012-11-06	1.0.0	初版.
2015-04-02	1.1.0	トルクセンサ通信仕様 Ver8 に対応

【対応機器】

機種	バージョン	注意事項

【ご注意】

本プロバイダを使用する場合は別途“タカス技研製小型トルクセンサ”ライセンスが必要です。

目次

1. はじめに	5
1.1. ライセンスの追加	5
2. プロバイダの概要	6
2.1. 概要	6
2.2. メソッド・プロパティ	7
2.2.1. CaoWorkspace::AddController メソッド	7
2.2.2. CaoController::Execute メソッド	8
2.2.3. CaoController::AddVariable メソッド	8
2.2.4. CaoController::get_VariableNames プロパティ	8
2.2.5. CaoVariable::get_Value プロパティ	9
2.3. 変数一覧	9
2.3.1. コントローラクラス	9
2.4. エラーコード	9
3. コマンドリファレンス	10
3.1. Execute コマンド	10
3.1.1. CaoController::Execute("Raw") コマンド	10
3.1.2. CaoController::Execute("Reset") コマンド	10
3.1.3. CaoController::Execute("PeekReset") コマンド	11
3.1.4. CaoController::Execute("GetSensorVersion") コマンド	11
3.1.5. CaoController::Execute("GetDecimalPoint") コマンド	11
3.1.6. CaoController::Execute("Display") コマンド	11
4. サンプルプログラム	12
4.1. VisualBasic Ver6.0	12
4.2. PacScript	13

1. はじめに

本書はタカス技研製小型トルクセンサ用の CAO プロバイダである、RLW(DSP)プロバイダのユーザーズガイドです。

RLW プロバイダは、インターフェースユニットに対してコマンドの送信とレスポンスの受信を行います。

1.1. ライセンスの追加

本プロバイダを使用可能にするにはORiN2 SDKをインストール後、別途「**タカス技研製 小型トルクセンサ**」ライセンスを入力する必要があります。

下記に「**タカス技研製 小型トルクセンサ**」ライセンスの追加手順を示します。

1. CaoConfigを起動し、[Cao Provider]タブを選択する
2. Provider Listから[TAKASU RLW CAO Provider]項目を選択する
3. License項目の[...]ボタンをクリックする
4. ORiN2 License Managerで[Add]ボタンをクリックする
5. 入手したライセンスキーを入力後、[OK]ボタンをクリックする
6. [Close] ボタンをクリックし、CaoConfigを終了する

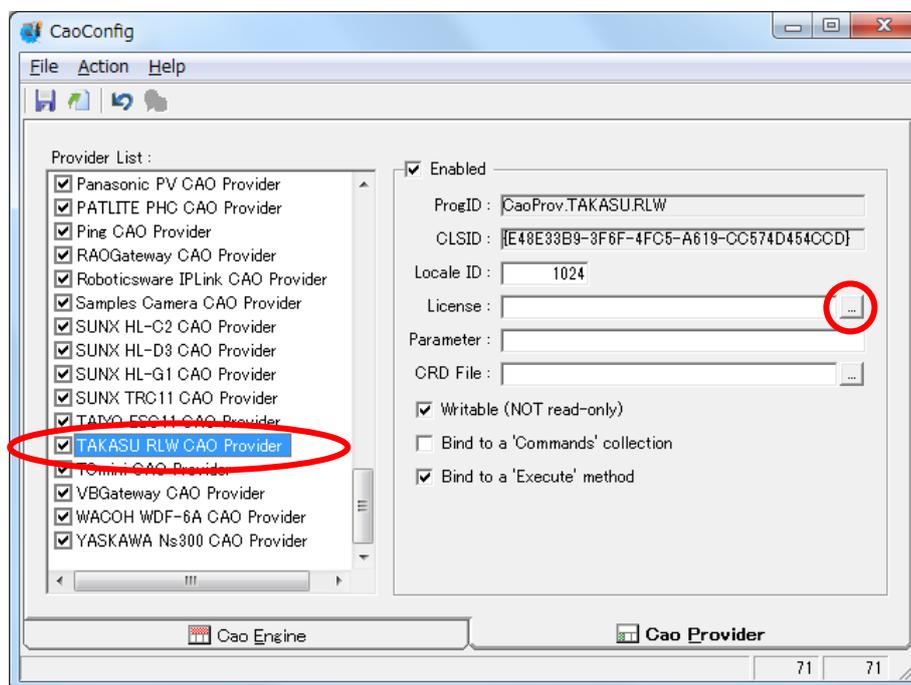


図 1-1 「タカス技研製 小型トルクセンサ」ライセンス追加

2. プロバイダの概要

2.1. 概要

RLW プロバイダは、コマンドの実行方法として `CaoController::Execute`、`CaoVariable` による 2 通りの方法を提供しています。

`CaoController::Execute` メソッド、及び `CaoVariable` は、実行時にコマンドを自動生成、送信を行います。またレスポンスの解析も行い、データを取得します。

RLW 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 RLW プロバイダ

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

RLW 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、換算係数、デバイスバージョンを指定します。

書式 AddController(<bstrCtrlName:VT_BSTR>,<bstrProvName:VT_BSTR>,
<bstrPcName:VT_BSTR > [,<bstrOption:VT_BSTR>])

bstrCtrlName : [in] コントローラ名 任意
 bstrProvName : [in] プロバイダ名 固定値 =” CaoProv.TAKASU.RLW”
 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します。

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指定します。 (デフォルト：500)
Convert[=<換算係数>]	トルクデータの換算係数を指定します。 Ver1.0.0 0：1/100 (デフォルト) 1：1/10 Ver1.1.0 以降 0：1/100 1：1/10 2：1/1000 3：1/1 ※接続時に係数を自動取得し設定されます 取得出来ない場合のみ設定値が使用されます
DeviceVer[=<デバイスバージョン>]	トルクセンサアンプのソフトバージョンを指定します。 Ver1.0.0 0：Ver5(1317)未満 1：Ver5(1317)以上 (デフォルト) Ver1.1.0 以降 0：Ver5(1317)未満 1：Ver5(1317)以上 2：Ver8(4431)以上 ※接続時にバージョン確認し自動設定されます 取得出来ない場合のみ設定値が使用されます

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. 9600, 38400, 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 (, 115200bps, None, 8bits, 1bit)
 (例 2) “com:2:9600” 通信ポート COM2, 9600bps (, None, 8bits, 1bit)
 (例 3) “com:3:38400:N:8:2” 通信ポート COM3, 38400bps, None, 8bits, 2bit

2.2.2. CaoController::Execute メソッド

コマンドの送受信を行います。第 1 引数にコマンド名、第 2 引数にコマンドのパラメータを指定します。各コマンドの詳細は 3 章コマンドリファレンスを参照してください。

書式 Execute (<bstrCommandName:VT_BSTR>,[<vntParam : VT_VARIANT>])

bstrCommandName: [in] コマンド名

vntParam : [in] パラメータ

2.2.3. CaoController::AddVariable メソッド

トルクセンサからのデータ読み込みを実行する CaoVariable オブジェクトを生成します。

変数名はトルクセンサのデータと対応します。指定できる変数名は 2.3.1 を参照して下さい。

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

bstrVariableName : [in] 変数名

bstrOption : [in] オプション文字列 (未使用)

2.2.4. 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 2.3.1 を参照して下さい。

2.2.5. CaoVariable::get_Value プロパティ

変数名で指定したデータを取得します。

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 変数一覧

変数名	データ型	説明	属性	
			get	put
@Torque	VT_R4	トルクデータ値	○	-
@MaxTorque	VT_R4	CW 方向ピークトルク値	○	-
@ MinTorque	VT_R4	CCW 方向ピークトルク値	○	-

2.4. エラーコード

RLW 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_RESPONSE_FAILED	0x80100001	無効なレスポンスを受信しました

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。

表 3-1 CaoController::Execute コマンド一覧

トルクセンサ コマンド	コマンド	機能
-	Raw	生データ送受信.
Y	Reset	トルクセンサをゼロリセットします.
W	PeekReset	トルクセンサのピーク値をリセットします.
UV	GetSensorVersion	トルクセンサアンプソフトのバージョンを取得します.
UD	GetDecimalPoint	トルクセンサの小数点位置情報を取得します
V,Q	Display	LED 表示を切替えます.

3.1. Execute コマンド

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

生データ送受信を行います。

書式 Raw (<bstrSendData>)

<bstrSendData> : [in] 送信文字列 (VT_BSTR)

戻り値 : [out] 受信文字列 (VT_BSTR)

引数で指定した文字列をトルクセンサに送信します。受信したレスポンスを文字列で返します。このとき送受信でコマンド及びレスポンスの内容は一切加工しません。

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

トルクセンサをゼロリセットします。

書式 Reset ()

引数 : なし

戻り値 : なし

※ AddController 時の接続オプションに、DeviceVer=0 を指定した場合は、トルクセンサからの応答を取得しません。

3.1.3. CaoController::Execute(“PeekReset”) コマンド

トルクセンサのピーク値をリセットします。

※要トルクセンサアンプバージョン Ver8(4431)以上

書式 PeekReset ()

引数 : なし

戻り値 : なし

3.1.4. CaoController::Execute(“GetSensorVersion”) コマンド

トルクセンサアンプのバージョン情報を取得します。

※要トルクセンサアンプバージョン Ver8(4431)以上

書式 GetSensorVersion ()

引数 : なし

戻り値 : [out] バージョン情報 (VT_I4)

Ver8(4431)の場合、4431 が取得されます

3.1.5. CaoController::Execute(“GetDecimalPoint”) コマンド

トルクセンサアンプのバージョン情報を取得します。

※要トルクセンサアンプバージョン Ver8(4431)以上

書式 GetDecimalPoint ()

引数 : なし

戻り値 : [out] 小数点位置情報 (VT_I4)

0	0 桁目(xxxx.)
1	1 桁目(xxx.x)
2	2 桁目(xx.xx)
3	3 桁目(x.xxx)
4	4 桁目(.xxxx)
9	小数点なし

3.1.6. CaoController::Execute(“Display”) コマンド

トルクセンサアンプの LED 表示を切替えます。

書式 Display (<iSwitch>)

<iSwitch> : [in] 切替えパラメータ (VT_I4)

0	LED 表示を OFF にします
1	LED 表示を ON にします。

戻り値 : なし

4. サンプルプログラム

トルクセンサのデータ取得，ピーク値リセットをするコードを示します。

4.1. VisualBasic Ver6.0

List 4-1 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

    ' トルクセンサへの接続
    Set ctrl = eng.Workspaces(0).AddController("Sample", "CaoProv.TAKASU.RLW", "",
"Conn=com:1")

    ' 変数オブジェクトの生成
    Set val = ctrl.AddVariable("@Torque")

End Sub

Private Sub Command1_Click()

    ' トルク値の取得
    Text1.Text = val.Value

End Sub

Private Sub Command2_Click()

    ' ピーク値リセット
    Ctrl.Execute("PeekReset")

End Sub
```

4.2. PacScript

List 4-2**Sample.pcs**

```
Dim ctrl As Object
Dim val As Object

Sub Main()

    Dim cnt As Integer

    ' トルクセンサへの接続
    ctrl = cao.AddController("Sample", "CaoProv. TAKASU. RLW", "", "Conn=com:2")

    ' 変数オブジェクトの生成
    val = ctrl.AddVariable("@Torque")

    ' ピーク値リセット
    ctrl.PeekReset

    Do
        ' トルク値の取得
        F[0] = val.Value
    Loop Until F[0] > F[1] ' トルク値が設定値を超えるまでループ

    val = Nothing
    ctrl.Variables.Clear
    ctrl = Nothing

End Sub
```