

GTプロバイダ
KEYENCE 接触式変位センサ

ユーザーズ ガイド

Version 1.0.0

February 13, 2017

【備考】

【改版履歴】

日付	版数	内容
2012-11-15	1.0.0	初版.
2017-02-13		変数のデータ型訂正.

【対応機器】

機種	バージョン	注意事項
GT2		DL-RS1A 使用
GT70A		DL-RS1A 使用

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::AddExtension メソッド	7
2.2.3. CaoController::Execute メソッド	7
2.2.4. CaoExtension::AddVariable メソッド	7
2.2.5. CaoExtension::Execute メソッド	8
2.2.6. CaoExtention::get_VariableNames プロパティ	8
2.2.7. CaoVariable::put_ID プロパティ	8
2.2.8. CaoVariable::get_ID プロパティ	8
2.2.9. CaoVariable::put_Value プロパティ	8
2.2.10. CaoVariable::get_Value プロパティ	8
2.3. 変数一覧	9
2.3.1. 拡張ボードクラス	9
2.4. エラーコード	12
3. コマンドリファレンス	13
3.1. Execute コマンド	13
3.1.1. CaoController::Execute("Raw") コマンド	13
3.2. Execute コマンド	13
3.2.1. CaoExtention::Execute("Preset") コマンド	13
3.2.2. CaoExtention::Execute("PresetReset") コマンド	14
3.2.3. CaoExtention::Execute("Reset") コマンド	14
3.2.4. CaoExtention::Execute("InitReset") コマンド	14
3.2.5. CaoExtention::Execute("ErrClear") コマンド	14
4. サンプルプログラム	15
4.1. VisualBasic Ver6.0	15
4.2. PacScript	16

1. はじめに

本書は KEYENCE 製接触式変位センサ用の CAO プロバイダである，GT プロバイダのユーザーズガイドです。

GT プロバイダは，RS-232C 通信ユニット(DL-RS1A)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。

2. プロバイダの概要

2.1. 概要

GT プロバイダは、コマンドの実行方法として `CaoController::Execute`、`CaoExtension::Execute`、`CaoVariable` による 3 通りの方法を提供しています。

`CaoController::Execute` メソッドは、指定したコマンドをそのまま送信し、受信したレスポンスをそのまま取得することができます。

`CaoExtension::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。

GT 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 GT プロバイダ

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

GT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、接続パラメータ、タイムアウトの設定を指定します。

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.KEYENCE.GT”
 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)

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, 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::AddExtension メソッド

変位センサのコマンドを自動生成する CaoExtension を生成します。

書式 AddExtension(<bstrChannelNo:VT_BSTR>,[< bstrOption: VT_BSTR >])
 bstrChannelNo : [in] 変位センサ ID 番号
 bstrOption : [in] オプション文字列

表 2-3 CaoController::AddExtension のオプション文字列

オプション	意味
Type =<機種タイプ>	機種タイプを指定します. (デフォルト : 0) 0 : GT2 シリーズ, 1 : GT70A シリーズ

このとき拡張ボード名にコマンドを送信する変位センサの ID 番号を指定しなければなりません。以下に拡張ボード名の命名規則を示します。

“CH<ID 番号>”

※GT2 シリーズの場合 ID : 0~14 GT70A シリーズの場合 ID : 0~9

2.2.3. 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

2.2.4. CaoExtension::AddVariable メソッド

変位センサのデータの読書きを実行する CaoVariable オブジェクトを生成します。

変数名は変位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)

2.2.5. CaoExtension::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

2.2.6. CaoExtention::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。

2.2.7. CaoVariable::put_ID プロパティ

CaoVariable の ID を設定します。設定値は引数で指定した値を設定します。設定できる変数は、2.3.1 を参照して下さい。

2.2.8. CaoVariable::get_ID プロパティ

CaoVariable の ID を取得します。取得できる変数は、2.3.1 を参照して下さい。

2.2.9. CaoVariable::put_Value プロパティ

変数名で指定したデータを設定します。設定値は引数で指定した値を設定します。

2.2.10. 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. 拡張ボードクラス

表 2-4 拡張ボードクラス ユーザ変数一覧

変数名	データ型	説明	属性		対応機器	
			get	put	GT2	GT70A
@CriteriaValue	VT_R8	判定値(P.V.値)	○	-	○	○
@CalcValue	VT_R8	演算表示値	○	-	○	○
@RawValue	VT_R8	生値(R.V.値)	○	-	○	○
@PeekValue	VT_R8	P-P モード時のピークホールド値	○	-	○	○
@BottomValue	VT_R8	P-P モード時のボトムホールド値	○	-	○	○
@ControlOutput	VT_I4	制御出力	○	-	○	○
@ErrStatus	VT_I4	センサアンプエラー状態	○	-	○	○
@CalcRawValue	VT_R8	演算結果の元となった R.V.値 ※機器 ID は ID プロパティにて指定 GT2 : 0~14 GT70A : 0~9	○	-	○	○
@Preset	VT_I4	プリセット実行要求	○	○	○	○
@Reset	VT_I4	リセット要求	○	○	○	○
@InitReset	VT_I4	イニシャルリセット要求	○	○	○	○
@ErrClear	VT_I4	エラークリア要求	○	○	○	○
@Bank	VT_I4	バンク選択状態	○	○	○	○
@Timing	VT_I4	タイミング状態	○	○	○	○
@KeyLock	VT_I4	キーロック	○	○	○	○
@BarDisp	VT_I4	バー表示モード	○	○	○	○
@Bank_HH	VT_R8	バンク*の HH 側設定値 ※バンク No は ID プロパティにて指定(0~3)	○	○	○	-

@Bank_HIGH	VT_R8	バンク*の High 側設定値 ※バンク No は ID プロパティにて指定(0~3)	○	○	○	○
@Bank_LOW	VT_R8	バンク*の Low 側設定値 ※バンク No は ID プロパティにて指定(0~3)	○	○	○	○
@Bank_LL	VT_R8	バンク*の LL 側設定値 ※バンク No は ID プロパティにて指定(0~3)	○	○	○	-
@Bank_Preset	VT_R8	バンク*のプリセット側設定値 ※バンク No は ID プロパティにて指定(0~3)	○	○	○	○
@CalcMode	VT_I4	演算モード・演算設定	○	○	○	○
@DetectingMode	VT_I4	検出モード	○	○	○	○
@HoldUpdate	VT_I4	ホールド更新方式	○	○	○	-
@ResponseTime	VT_I4	応答時間	○	○	○	○
@TimingCategory	VT_I4	タイミング種別	○	○	○	○
@TimingLevel	VT_R8	セルフタイミングレベル	○	○	○	○
@TimingDelay	VT_I4	セルフタイミングディレイ種別	○	○	○	○
@DelayTime	VT_I4	ユーザ指定ディレイ時間	○	○	○	○
@JadgeDelay	VT_I4	スタティックホールド ディレイ安定判断	○	○	○	○
@HoldRange	VT_R8	スタティックホールド ディレイ安定幅	○	○	○	○
@MeasureDirect	VT_I4	計測増減方向	○	○	○	○
@LeverRatio	VT_R8	レバー比	○	○	○	○
@Output	VT_I4	出力様式	○	○	○	○
@DisplayDigit	VT_I4	表示桁数	○	○	○	○
@Hysteresis	VT_R8	ヒステリシス	○	○	○	○
@AllInput	VT_I4	一斉出力設定	○	○	○	-
@SpecialOutput	VT_I4	特別出力設定	○	○	○	-
@InputSel1	VT_I4	入力(紫線)機能選択 1	○	○	-	○

@InputSel2	VT_I4	入力(桃/紫線)機能選択 2	○	○	-	○
@Limit_HH	VT_R8	リミット出力 HH 側判定位置設定	○	○	○	-
@Limit_LL	VT_R8	リミット出力 LL 側判定位置設定	○	○	○	-
@PresetData	VT_I4	プリセットデータ選択	○	○	○	-
@PresetMem	VT_I4	プリセット記憶	○	○	○	○
@PresetPoint	VT_I4	プリセットポイント	○	○	○	-
@EcoMode	VT_I4	省電力機能(ECO)	○	○	○	○
@CheckHook	VT_I4	ひっかかりチェック機能	○	○	○	○
@CheckHookPos	VT_R8	ひっかかりチェック動作位置	○	○	○	○
@BunchSetting	VT_I4	一括設定	○	○	○	-
@Filter	VT_I4	表示フィルタ機能	○	○	-	○
@AnalogSetting	VT_I4	アナログレンジ設定	○	○	○	-
@FreeRange_Hi	VT_R8	フリーレンジ設定(Hi 側)	○	○	○	-
@FreeRange_Lo	VT_R8	フリーレンジ設定(Lo 側)	○	○	○	-

2.4. エラーコード

GT 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 2-5 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
E_SENserAMP_ERROR	0x80100011	センサーアンプがエラー状態です
E_UPPERLIMIT_OVER	0x80100012	検出範囲の上限を超えた値です
E_LOWERLIMIT_OVER	0x80100013	検出範囲の下限を超えた値です
E_AMP_CONDITION_ERROR	0x80100014	<ul style="list-style-type: none"> •ホールドモード使用時, アンプの P.V.値表示が「----」になっています •GT2-100 シリーズ使用時, 有効 ID 設定の範囲外 ID データを読み出している
E_DEVICE_ERROR	0x80100100～	デバイスからのエラーレスポンス

※E_DEVICE_ERROR について

デバイスからのエラーレスポンスは、"0x80100100"でマスクした値が出力されます。

例：セルフタイミングレベル(@TimingLevel)設定時にエラーが発生した場合。

パラメータエラー：22 → CAO API のエラー：0x80100116

エラー内容については KEYENCE 社の DL-RS1A ユーザーズマニュアルを参照してください。

3. コマンドリファレンス

本章では CaoController::Execute, 及び CaoExtention::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

コマンド	機能
Raw	生データ送受信.

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変位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.

表 3-2 CaoExtention::Execute コマンド一覧

コマンド	機能
Preset	プリセット実行
PresetReset	プリセットリセット実行
Reset	リセット実行
InitReset	イニシャルリセット実行
ErrClear	エラークリア実行

3.2. Execute コマンド

3.2.1. CaoExtention::Execute("Preset") コマンド

プリセット実行要求を行います.

書式 Preset ()

引数 : 無し

戻り値 : 無し

3.2.2. CaoExtention::Execute(“PresetReset”) コマンド

プリセットリセット実行要求を行います。

書式 PresetReset ()

引数 : 無し

戻り値 : 無し

3.2.3. CaoExtention::Execute(“Reset”) コマンド

リセット実行要求を行います。

書式 Reset ()

引数 : 無し

戻り値 : 無し

3.2.4. CaoExtention::Execute(“InitReset”) コマンド

イニシャルリセット実行要求を行います。

書式 InitReset ()

引数 : 無し

戻り値 : 無し

3.2.5. CaoExtention::Execute(“ErrClear”) コマンド

エラークリア実行要求を行います。

書式 ErrClear ()

引数 : 無し

戻り値 : 無し

4. サンプルプログラム

変位センサの ID : 0 をリセット, バンク 2 の High 側を設定, 判定値(P.V.値)を取得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 Ext As CaoExtension
Dim wVal As CaoVariable
Dim rVal As CaoVariable

Private Sub Form_Load()

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

    ' 変位センサへの接続
    Set ctrl = eng.Workspaces(0).AddController("Sample", "CaoProv. KEYENCE. GT", "", "Conn=com:1")

    Set Ext = ctrl.AddExtension("CH0")

    ' 変数オブジェクトの生成
    Set wVal = Ext.AddVariable("@Bank_HIGH ")
    Set rVal = Ext.AddVariable("@CriteriaValue ")

End Sub

Private Sub Command1_Click()

    ' リセット要求
    Ext.Execute "Reset"

End Sub

Private Sub Command2_Click()

    ' バンク No の設定
    wVal.ID = 2
    ' 値の設定
    wVal = cSng(Text1.Text)
End Sub

Private Sub Command3_Click()

    ' 値の取得
    Text2.Text = cStr(rVal)
End Sub
```

4.2. PacScript

List 4-2**Sample.pcs**

```
Dim ctrl As Object
Dim Ext As Object
Dim wVal As Object
Dim rVal As Object

Sub Main()

    Dim cnt As Integer

    ' 変位センサへの接続
    ctrl = cao.AddController("Sample", "CaoProv.KEYENCE.GT", "", "Conn=com:2")

    Ext = ctrl.AddExtension("CH0")

    ' 変数オブジェクトの生成
    wVal = Ext.AddVariable("@Bank_HIGH ")
    rVal = Ext.AddVariable("@CriteriaValue ")

    Ext.Execute "Reset"          ' リセット要求

    wVal.ID = 2                  ' バンク No の設定
    wVal = F[0]                  ' 値の設定

    Do
        F[1] = cStr(rVal)        ' 値の取得
    Loop Until F[1] < F[0]      ' 判定値が設定値になるまでループ

End Sub
```