

HOKUYO URG04LX プロバイダ

Version 1.0.0

ユーザーズ ガイド

September 30, 2013

備考:

【改版履歴】

バージョン	日付	内容
1.0.0	2013-9-30	初版.

【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
2.2.1.1. Conn オプション	6
2.2.2. CaoController::Execute メソッド	7
3. コマンドリファレンス	8
3.1.1.1. CaoController::Execute("GetVersionInfo") コマンド	8
3.1.1.2. CaoController::Execute("SetSensorState") コマンド	8
3.1.1.3. CaoController::Execute("SetConnSettings") コマンド	9
3.1.1.4. CaoController::Execute("GetSensorValue") コマンド	9

1. はじめに

本書は HOKUYO URG-04LX 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 (CaoProvURG04LX.dll) を URG04LX プロバイダと呼びます。

次章に URG04LX 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

URG04LXプロバイダは、HOKUYO URG-04LX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。そのファイル形式は 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。URG04LX 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 URG04LX プロバイダ

ファイル名	CaoProvURG04LX.dll
ProgID	CaoProv.HOKUYO.URG04LX
レジストリ登録	regsvr32 CaoProvURG04LX.dll
レジストリ登録の抹消	regsvr32 /u CaoProvURG04LX.dll

2.2. メソッド・プロパティ

2.2.1. CaoWorkspace::AddController メソッド

URG04LX 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接続パラメータ、タイムアウトの設定を指定します。オプションとオプションの区切りは“,”で行います。

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

bstrCtrlName : [in] コントローラ名
 bstrProvName : [in] プロバイダ名. 固定値 =” CaoProv.HOKUYO.URG04LX”.
 bstrPcName : [in] プロバイダの実行マシン名
 bstrOption : [in] オプション文字列=“<オプション 1>,<オプション 2>,...”

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	説明
Conn=<接続パラメータ>	必須. 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. (デフォルト:500 msec)

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>[:<Flow>]]]”

<COM Port> : COM ポート番号. ‘1’-COM1, ‘2’-COM2, ...
 <BaudRate> : 通信速度.
19200, 38400, 57600, 115200, 250000, 500000, 750000
 <Parity> : パリティ. ‘N’-NONE
 <DataBits> : データビット数. ‘8’-8bit
 <StopBits> : ストップビット数. ‘1’-1bit
 <Flow> : フロー制御. ‘0’-フロー制御なし

2.2.2. GaoController::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。

書式 [`<vntRet:VARIANT> =] Execute(<bstrCmd:BSTR > [,<vntParam:VARIANT>])`

<code>bstrCmd</code>	:	[in]	コマンド
<code>vntParam</code>	:	[in]	パラメータ
<code>vntRet</code>		[out]	返回值

コマンドの実行に必要なパラメータおよび取得する結果は 3 章を参照して下さい。

3. コマンドリファレンス

表 3-1 CaoController::Execute メソッドのコマンド一覧

カテゴリ	コマンド名	機能	
SCIP1.1			
	GetVersionInfo	バージョン情報の取得	p. 8
	SetSensorState	レーザー発光の制御	p. 8
SCIP2.0			
その他			
	SetConnSettings	通信設定の切り替え	p. 9
	GetSensorValue	距離データの取得	p. 9

3.1.1.1. CaoController::Execute("GetVersionInfo") コマンド

バージョン情報を取得します。



GetVersionInfo ()

引数 : なし
 戻り値 : バージョン情報 (VT_BSTR|VT_ARRAY:6 要素)
 [0]:ベンダ情報
 [1]:製品情報
 [2]:ファームウェアバージョン
 [3]:プロトコルバージョン
 [4]:センサ固有シリアル番号
 [5]:ステータス



```
caoCtrl.Execute "GetVersionInfo" 'バージョン情報を取得'
```

3.1.1.2. CaoController::Execute("SetSensorState") コマンド

レーザーの点灯/消灯を設定します。



SetSensorState (<IState>)

IState : [in]制御コード (VT_I4)

	0:点灯
	1:消灯
戻り値	: ステータス(VT_I4)
	0:異常なし
	1:『制御コード』に設定可能値以外の数値
	2:故障中のためレーザー制御は不可

使用例

```
caoCtrl.Execute "SetSensorState", 0      ' レーザーを点灯
```

3.1.1.3. CaoController::Execute("SetConnSettings") コマンド

通信設定を変更します。

書式 SetConnSettings (<IBaudRate>)

IBaudRate	: [in]通信速度(VT_I4)
	19200, 38400, 57600, 115200, 250000, 500000, 750000
戻り値	: ステータス(VT_I4)
	0:異常なし
	1:『ボーレート』に数字以外の文字が混在
	2:『ボーレート』に設定可能値以外の数値

使用例

```
caoCtrl.Execute "SetConnSettings", 38400 ' ボーレートを 38.4kbps に設定
```

3.1.1.4. CaoController::Execute("GetSensorValue") コマンド

距離データを取得します。

書式 GetSensorValue (<IStart>, <IEnd>[, <IStep>])

IStart	: [in]開始方向(VT_I4)
IEnd	: [in]終了方向(VT_I4)
IStep	: [in]まとめるステップ数(VT_I4)
	省略時は 0
戻り値	: ステータス(VT_I4)または距離データ(VT_I4 VT_ARRAY)

ステータス(VT_I4)

- 1:『開始方向』に数字以外の文字が混在
- 2:『終了方向』に数字以外の文字が混在
- 3:『まとめるステップ数』に数字以外の文字が混在
- 4:『開始方向』の指定数が最大設定範囲以上
- 5:『終了方向』の指定数が『開始方向』の指定数以下
- 6:レーザー消灯コマンドによりレーザー消灯中
- 7:レーザー異常によりレーザー消灯中1
- 8:レーザー異常によりレーザー消灯中2
- 9:モータ異常などのハードウェアの故障

距離データ(VT_I4|VT_ARRAY)

使用例

```
caoCtrl.Execute "GetSensorValue", ( 0, 768 )
```

```
' 距離データを取得
```
