

NS300 プロバイダ

YASKAWA フィールドバス I/F ユニット

Version 1.0.1

ユーザーズ ガイド

July 17, 2012

【備考】

【改版履歴】

バージョン	日付	内容
1.0.0.0	2006-02-24	初版.
1.0.1.0	2007-06-29	サンプルプログラムの誤記訂正
1.0.1.1	2010-02-12	エラーコード追加
1.0.1	2012-07-17	ドキュメントのバージョンルールを変更

【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
2.2.2. CaoController::AddCommand メソッド	6
2.2.3. CaoController::AddRobot メソッド	7
2.2.4. CaoController::AddVariable メソッド	7
2.2.5. CaoController::get_CommandNames プロパティ	7
2.2.6. CaoController::get_VariableNames プロパティ	7
2.2.7. CaoController::Execute メソッド	7
2.2.8. CaoCommand::Execute メソッド	8
2.2.9. CaoCommand::put_Parameters プロパティ	8
2.2.10. CaoCommand::get_Parameters プロパティ	8
2.2.11. CaoCommand::get_Result プロパティ	8
2.2.12. CaoRobot::Move メソッド	8
2.2.13. CaoRobot::Halt メソッド	9
2.2.14. CaoRobot::Hold メソッド	9
2.2.15. CaoRobot::Unhold メソッド	9
2.2.16. CaoRobot::GoHome メソッド	9
2.2.17. CaoVariable::put_Value メソッド	9
2.2.18. CaoVariable::get_Value メソッド	9
2.3. 変数一覧	10
2.3.1. コントローラクラス	10
2.4. エラーコード	11
3. サンプルプログラム	12

1. はじめに

本書は、YASKAWA 製 Ns300 シリーズと DeviceNet 通信を行うプロバイダである、Ns300 プロバイダのユーザーズガイドです。

通信プロトコルの詳細については、Ns300 のマニュアルを参照して下さい。

2. プロバイダの概要

2.1. 概要

Ns300 プロバイダは、Ns300 シリーズの DeviceNet 通信コマンドをコマンド、ロボット、変数クラスで実装しています。

通信には DeviceNet プロバイダを使用して通信を行っています。このため、このプロバイダを使用するためには必ず、DeviceNet プロバイダをレジストリに登録する必要があります。

プロバイダの概要について記述します。

表 1 Ns300 プロバイダ

ファイル名	GaoProvNs300.dll
ProgID	GaoProv.YASKAWA.Ns300
レジストリ登録 ¹	regsvr32 GaoProvNs300.dll
レジストリ登録の抹消	regsvr32 /u GaoProvNs300.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

Ns300 プロバイダでは、Controller オブジェクトの生成時に DeviceNet 通信による接続処理を行います。このとき、オプション文字列を使用して、接続に必要な DeviceNet ボード ID、ボーレート、マスターノード、スレーブノードの指定を行います。また、接続時に IN/OUT のサイズは共に 8 バイトで設定されます。

```
AddController
(
    "<コントローラ名>",           // コントローラ名.
    "CaoProv. YASKAWA. Ns300",   // プロバイダ名. 固定.
    "<マシン名>",                 // プロバイダの実行マシン名.
    "<オプション>"                // 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 CaoWorkspace::AddController のオプション文字列

オプション	意味
BoardId=<BoardId>	接続先 DeviceNet ボード番号の指定(必須)
BaudRate=<BaudRate>	通信速度の指定(必須) 0: 125kbps 1: 250kbps 2: 500kbps
MasterNo =<MacId>	マスターノードアドレス(必須)
SlaveNo =<MacId>	スレーブノードアドレス(必須)

2.2.2. CaoController::AddCommand メソッド

Ns300 のコマンドを実行するコマンドオブジェクトを生成します。

以下に使用できるコマンド名の一覧を示します。

```
AddCommand
(
    "<コマンド名>",           // コマンド名.
    "<オプション>"            // オプション文字列. (未使用)
)
```

表 3 コマンド名一覧

コマンド名	Ns300 の指令	パラメータ	結果
ResetAlarm	アラームリセット指令	-	-
ResetModule	モジュールリセット指令	-	-
SetCurrentPos	現在位置設定指令	VT_I4: 現在位置データ	-
SetOriginalPos	原点位置設定指令	-	-
GetAlarmHistory	アラーム読み出し指令	-	VT_ARRAY VT_UI1 : アラームコード

各コマンドの動作に関する詳細は Ns300 のマニュアルを参照して下さい。

2.2.3. CaoController::AddRobot メソッド

Ns300 の動作指令コマンドを行うことができるロボットオブジェクトを生成します。ロボット名には任意の名前を指定することができます。

```
AddRobot
(
    "<ロボット名>",           // ロボット名.
    "<オプション>"           // オプション文字列. (未使用)
)
```

2.2.4. CaoController::AddVariable メソッド

ステータス情報、各パラメータに対応した変数オブジェクトを生成します。指定できる変数名については、2.3.1 を参照してください。

```
AddVariable
(
    "<変数名>"               // 変数名.
    "<オプション>"           // オプション文字列. (未使用)
)
```

2.2.5. CaoController::get_CommandNames プロパティ

コマンド名の一覧を表示します。取得する一覧については表 3 を参照してください。

2.2.6. CaoController::get_VariableNames プロパティ

変数名の一覧を指定します。取得する一覧については表 5 を参照してください。

2.2.7. CaoController::Execute メソッド

DeviceNet の IN, OUT に直接読み書きを行います。コマンド名には以下のものを指定することができます。

コマンド名	説明	パラメータ	結果
-------	----	-------	----

GetIN	IN のデータを読み込みます。	-	VT_ARRAY VTUI1
GetOUT	OUT のデータを読み込みます。	-	VT_ARRAY VTUI1
SetOUT	OUT にデータを設定します。	VT_ARRAY VTUI1	-

このときパラメータおよび結果は 8 バイトの配列で取得, 設定します。

2.2.8. CaoCommand::Execute メソッド

コマンド名に対応する Ns300 の指令を実行します。パラメータが必要なコマンドのときは, 2.2.9 で設定したパラメータを使用します。パラメータは不正なときは, このメソッド実行時にエラーを発生させます。発生したエラー情報は 2.2.11 で取得します。

2.2.9. CaoCommand::put_Parameters プロパティ

コマンドに必要なパラメータを設定します。パラメータが不正なときでも, エラーは発生しません。

2.2.10. CaoCommand::get_Parameters プロパティ

2.2.9 で設定したパラメータを取得します。何も設定されていないときは VT_EMPTY を取得します。

2.2.11. CaoCommand::get_Result プロパティ

2.2.8 の実行結果を取得します。

2.2.12. CaoRobot::Move メソッド

PTP 動作又はポジション No. 指定実行を行います。

```

Move
(
    "<補間指定>",           // 入力内容は無視されます。
    "<ポーズ列>",           // 指定方法により動作を変えます。
    "<オプション>"         // 入力内容は無視されます。
)

```

第 1 引数の内容は全て無視されます。

第 2 引数のポーズ列の指定方法により以下のように動作を変えます。

第 3 引数の内容は全て無視されます。

表 4 ポーズ列の指定と動作の対応表

ポーズ列	動作
P<数値>	ポジション No. 指定実行(Q3 コマンド)
<数値>	絶対位置指令 PTP 動作(c コマンド)

* + (<数値>)	相対位置指令 PTP 動作(e コマンド)
------------	-----------------------

- 例1) ポジション No. 10 の場合 : "P10"
例2) 絶対位置 100 の場合 : "100"
例3) 相対位置 200 の場合 : "* + (200)"

ポジションデータの目標位置及び移動速度はコントローラクラスの“Pn”変数を介してアクセスすることが出来ます。必要に応じてこの変数を値を設定してください。

2.2.13. GaoRobot::Halt メソッド

Ns300 のキャンセル指令を実行します。
詳細は Ns300 のマニュアルを参照して下さい。

2.2.14. GaoRobot::Hold メソッド

Ns300 の一時停止指令を実行します。
詳細は Ns300 のマニュアルを参照して下さい。

2.2.15. GaoRobot::Unhold メソッド

Ns300 のキャンセル指令のフラグを解除します。
詳細は Ns300 のマニュアルを参照して下さい。

2.2.16. GaoRobot::GoHome メソッド

Ns300 の原点復帰指令を実行します。
詳細は Ns300 のマニュアルを参照して下さい。

2.2.17. GaoVariable::put_Value メソッド

変数名に対応した情報を書込みます。各変数の動作については、2.3.1 を参照してください。

2.2.18. GaoVariable::get_Value メソッド

変数名に対応した情報を取得します。各変数の動作については、2.3.1 を参照してください。

2.3. 変数一覧

2.3.1. コントローラクラス

表 5 コントローラクラス ユーザ変数一覧

変数名	データ型	説明	属性	
			get	put
Pn <ユーザ定数番号 ² >	VT_I4	ユーザ定数値の取得, 設定を行います. ユーザ定数番号には 16 進数で値を指定します. Ex. ポイント No.1: "Pn900" = 目標位置, "Pn940" = 速度 ポイント No.2: "Pn901" = 目標位置, "Pn941" = 速度 : ポイント No.50: "Pn931" = 目標位置, "Pn971" = 速度 値保存は ResetModule コマンドを実行で行います.	○	○
Servo	VT_BOOL	サーボの状態の取得, 設定を行います. True : サーボ ON False : サーボ OFF	○	○
Alarm	VT_BOOL	アラームの状態を取得します. True : アラーム発生中 False : アラームなし	○	-
Warning	VT_BOOL	ワーニングの状態を取得します. True : ワーニング発生中 False : ワーニングなし	○	-
Ready	VT_BOOL	コマンドレディの状態を取得します. True : コマンドレディ False : コマンドビジジー	○	-
Power	VT_BOOL	主電源の状態を取得します. True : 主電源 ON False : 主電源 OFF	○	-

²詳細は Ns300 のマニュアルを参照して下さい.

Running	VT_BOOL	コマンドの実行状態を取得します。直前に CaoRobot::Move メソッドを実行したときのみ取得することができます。それ以外ではエラーを返します。 True : コマンド実行中 False : コマンド実行完了	○	-
Hold	VT_BOOL	ホールド状態を取得します。直前に CaoRobot::Move メソッドを実行したときのみ取得することができます。それ以外ではエラーを返します。 True : ホールド False : ホールド解除	○	-
OperationPosition	VT_I4	指令位置を取得します。(指令単位)	○	-
CurrentPosition	VT_I4	現在位置を取得します。(指令単位)	○	-
Deviation	VT_I4	位置偏差を取得します。(指令単位)	○	-
OperationSpeed	VT_I4	指令速度を取得します。(1000 指令単位/分)	○	-
CurrentSpeed	VT_I4	現在速度を取得します。(1000 指令単位/分)	○	-
Torque	VT_I4	トルクを取得します。(%)	○	-
StationNo	VT_I4	ステーション番号を取得します。	○	-
PositionNo	VT_I4	ポイントテーブル番号を取得します。	○	-

2.4. エラーコード

Ns300 プロバイダでは、固有のエラーコードはありません。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。

3. サンプルプログラム

以下に Text1 の位置へ移動するサンプルを示します。

List 3-1**Sample.frm**

```
Private eng As CaoEngine
Private ctrl As CaoController
Private rob As CaoRobot

Private Sub Form_Load()
    Dim ws As CaoWorkspace

    Set eng = New CaoEngine
    Set ws = eng.Workspaces(0)

    Set ctrl = ws.AddController("Sample", _
                                "CaoProv. YASKAWA. Ns300", _
                                ""', _
                                "BoardId=0, BaudRate=2, MasterNo=63, SlaveNo=32")

    Set rob = ctrl.AddRobot("0")

    Dim var As CaoVariable
    With rob
        .AddRobot("0") ' YAS 走行軸オブジェクトを用意
        Set var = .AddVariable("Servo")
        caoVar.Value = True ' サーボ ON
    End With

End Sub

Private Sub Command1_Click()
    rob.Move 1, Text1.Text
    ' Ex.
    ' Move 1, "150000"           ' 絶対移動
    ' Move 1, "P1"              ' P1 へ移動
    ' Move 1, "*+(10000)"      ' 10000 だけ移動
End Sub
```