

S-Link プロバイダ SUNX SL-PCI ボード

Version 1.0.0

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.
1.0.0.1	2010-02-12	エラーコード追加
1.1.0.1	2011-03-11	プロバイダ登録ツールに関する追記
1.1.0	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	7
2.2.1. CaoWorkspace::AddController メソッド	7
2.2.2. CaoController::AddCommand メソッド	8
2.2.3. CaoController::AddVariable メソッド	8
2.2.4. CaoController::get_CommandNames プロパティ	9
2.2.5. CaoController::get_ID プロパティ	9
2.2.6. CaoController::get_VariableNames プロパティ	9
2.2.7. CaoCommand::Execute メソッド	9
2.2.8. CaoCommand::put_Parameters プロパティ	9
2.2.9. CaoCommand::get_Parameters プロパティ	10
2.2.10. CaoVariable::get_Value プロパティ	10
2.2.11. CaoVariable::put_Value プロパティ	10
2.3. 変数一覧	11
2.3.1. コントローラクラス	11
2.4. エラーコード	11
3. サンプルプログラム	13

1. はじめに

この S-Link プロバイダは SUNX 社製 SL-PCI ボードと CAO アプリケーションを接続するためのゲートウェイ型の CAO プロバイダです。このプロバイダを使うことで、SUNX 社製 SL-PCI ボードが対応している様々な PLC に接続することができます。

注意: S-Linkプロバイダを使用するには、SL-PCI ボードのドライバをインストールしなければなりません。ドライバインストール後にプロバイダをレジストリ登録する必要があります。レジストリ登録の方法は表 2-2 を参照してください。

2. プロバイダの概要

2.1. 概要

S-Link プロバイダは、CAO API を実行するときに対応する S-Link API を実行します。

以下に CAO API と S-Link API の対応表を示します。

表 2-1 CAO API と S-Link API の対応表

CAO API		S-Link API
CaoController::FinalConstruct		SLPciCreate
CaoController::FinalRelease		SLPciClose
CaoCommand::Execute	(コマンド名 : SYSTEMSET)	SLPciSystemSet
	(コマンド名 : BANKCHANGE)	SLPciBankChange
CaoVariable::put_Value	(変数名 : IO<アドレス>)	SLPciOutBit
		SLPciOutByte
	(変数名 : STATUS<アドレス>)	SLPciReadStatusArea
	(変数名 : @WAITTIME)	SLPciSetWaitTime
	(変数名 : @IOS)	SLPciSetIOS
CaoVariable::get_Value	(変数名 : IO<アドレス>)	SLPciInBit
		SLPciInByte
	(変数名 : STATUS<アドレス>)	SLPciWriteStatusArea
	(変数名 : BS1A<アドレス>)	SLPciGetBS1AStatus
	(変数名 : @VERSION)	SLPciGetVersion
	(変数名 : @IOS)	SLPciGetIOS
	(変数名 : @TOTALUNIT)	SLPciGetTotalUnitNumber
	(変数名 : @SYSADDRESS)	SLPciSystemAddressLength
	(変数名 : @ERRORFLAGS)	SLPciGetErrorFlags
	(変数名 : @COMMSTATUS)	SLPciGetCommStatus
(変数名 : @LASTERROR)	SLPciGetLastError	

S-Link 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 2-2 のようになっています.

表 2-2 S-Link プロバイダ

ファイル名	GaoProvSLink.dll
ProgID	GaoProv.SUNX.SLINK
レジストリ登録 ¹	regsvr32 GaoProvSLink.dll
レジストリ登録の抹消	regsvr32 /u GaoProvSLink.dll

¹ プロバイダの登録は regsvr32.exe または RegCOM.exe ([スタート]→[ORiN2]→[Tools])で実行できます. SL-PCI ボードのドライバをインストールしていないと, S-Link プロバイダの登録はできません.

2.2. メソッド・プロパティ

2.2.1. CaoWorkspace::AddController メソッド

S-Link プロバイダでは, AddController 時に SL-PCI ボードに指定した BSN で接続を行います。AddController メソッドの引数仕様は下記に示す通りです。

```
AddController
(
    "<コントローラ名>",           // コントローラ名
    "GaoProv. SUNX. SLINK",       // プロバイダ名. 固定.
    "<マシン名>",                 // プロバイダ実行マシン名.
    "<オプション>"                // オプション文字列
)

例)
AddController
(
    "SLCtrl1",                    コントローラ名 : "SLCtrl1"
    "GaoProv. SUNX. SLINK",
    "",                            ローカルマシンで起動
    "BSN=1"                        BSN が 1 の S-Link ボードを使用
)
```

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 2-3 CaoWorkspace::AddController のオプション文字列

オプション	意味
BSN =<ボードセレクトナンバ>	使用する BSN (ボードセレクトナンバ) を指定します。SLPciCreate 関数の引数に使用されます。詳しい動作は S-Link API のドキュメントを参照して下さい。

2.2.2. CaoController::AddCommand メソッド

システムセット、バンク切り替えを行う CaoCommand オブジェクトを作成します。
指定できるオプション文字列はありません。

“BANKCHANGE” 時のみ、バンク番号を Parameter プロパティに指定します。

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

使用できるコマンドは表 2-4 を参照して下さい。

表 2-4 Command クラス使用可能予約語

コマンド名	Parameter	動作
SYSTEMSET	—	システムセットボタンを押した場合と同じ動作をします。 SLPciSystemSet を実行します。
BANKCHANGE	<バンク番号>	指定したバンク番号にバンクを切り替えます。（デフォルト：0） SLPciBankChange を実行します。

2.2.3. CaoController::AddVariable メソッド

値の取得/設定を行う CaoVariable オブジェクトを作成します。

ユーザ変数名の書式は以下の通りです。

<アドレス種別><アドレス番地>

使用できるアドレス種別は表 2-6 を参照して下さい。

アドレス番地は 10 進表記で指定して下さい。

アドレス種別が “IO” の時のみ、オプション文字列に IO 種別を指定できます。詳しくは表 2-5 を参照して下さい。

システム変数については表 2-7 を参照して下さい。

AddVariable メソッドの引数仕様は下記に示す通りです。

```
AddVariable
(
  “<コマンド名>”           // 変数名
  “<オプション文字列>”     // オプション文字列
)
```

```

)
例)
AddVariable
(
  "105" // アドレス種別：IO, アドレス番地：5 を指定
  "Len=8" // IO 種別：バイトを指定
)

```

表 2-5 IO 指定時に使用可能なオプション文字列

オプション	意味
Len =<1 8>	IO の種類を指定します。(デフォルト：1) 1：ビット 8：バイト

2.2.4. CaoController::get_CommandNames プロパティ

CaoCommand で使用できるコマンド予約語一覧を返します。表示されるコマンド予約語は表 2-4 を参照して下さい。

指定できるオプションはありません。

2.2.5. CaoController::get_ID プロパティ

BSN (ボードセレクトナンバ) を返します。

指定できるオプションはありません。

2.2.6. CaoController::get_VariableNames プロパティ

CaoCommand で使用できるシステム/ユーザ変数名一覧を返します。表示されるシステム/ユーザ変数名一覧は 2.3.1 を参照して下さい。

指定できるオプションはありません。

2.2.7. CaoCommand::Execute メソッド

各コマンド名に対応したコマンドを実行します。

動作の詳細については、表 2-4 を参照して下さい。

2.2.8. CaoCommand::put_Parameters プロパティ

コマンド名が“BANKCHANGE”のときのみ、バンク番号を設定します。

動作の詳細については、表 2-4 を参照して下さい。

2.2.9. CaoCommand::get_Parameters プロパティ

2.2.8 で設定したパラメータを取得します。パラメータが設定されていないときは VT_EMPTY を返します。

2.2.10. CaoVariable::get_Value プロパティ

各システム/ユーザ変数に対応した処理を行います。
詳しくは 2.3.1 を参照して下さい。

2.2.11. CaoVariable::put_Value プロパティ

各システム/ユーザ変数に対応した処理を行います。
詳しくは 2.3.1 を参照して下さい。

2.3. 変数一覧

2.3.1. コントローラクラス

表 2-6 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
IO<アドレス>	VT_I2	<p>特定 S-LINK アドレスの ON/OFF 制御/確認を行います。 AddVariable の Len オプションでビット/バイトを切替えます。</p> <p>Len=1 の場合 :</p> <p>get_Value で SLPciInBit 関数を実行します。 put_Value で SLPciOutBit 関数を実行します。</p> <p>Len=8 の場合 :</p> <p>get_Value で SLPciInByte 関数を実行します。 put_Value で SLPciOutByte 関数を実行します。</p>	○	○
STATUS<アドレス>	VT_I2	<p>S-LINK ステータスエリアへの書込み/読み出しを行います。</p> <p>get_Value で SLPciReadStatusArea 関数を実行します。 put_Value で SLPciWriteStatusArea 関数を実行します。</p>	○	○
BS1A<アドレス>	VT_I2	<p>BS1A の接続/動作状態を取得します。</p> <p>get_Value で SLPciGetBS1AStatus 関数を実行します。</p>	○	-

2.4. エラーコード

S-Link 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。

表 2-7 コントローラクラス システム変数一覧

変数名	データ型	説明	属性	
			get	put
@VERSION	VT_ARRAY VT_I4	DLL のバージョンを取得します。 要素 0 に Dll のバージョン, 要素 1 にドライバのバージョンを格納します。 get_Value で SLPciGetVersion 関数を実行します。	○	-
@WAITTIME	VT_I4	出力待ち時間の設定を行います。 put_Value で, SLPciSetWaitTime 関数を実行します。	-	○
@IOS	VT_I4	入出力設定の取得/設定を行います。 get_Value で SLPciGetIOs 関数を実行します。 put_Value で SLPciSetIOs 関数を実行します。	○	○
@TOTALUNIT	VT_ARRAY VT_VARIANT	要素 0 に S-LINK 入出力ユニット接続台数(VT_I4)を, 要素 1 にシステムセット完了状態(VT_BOOL)を取得します。 get_Value で SLPciGetTotalUnitNumber 関数を実行します。	○	-
@SYSADDRESS	VT_I4	システムアドレス長を取得します。 get_Value で SLPciGetSystemAddressLength 関数を実行します。	○	-
@ERRORFLAGS	VT_ARRAY VT_I4	エラーフラグの状態を取得します。 要素 0 にエラーフラグの状態を, 要素 1 に ERR4 発生台数を格納します。 get_Value で SLPciGetErrorFlags 関数を実行します。	○	-
@COMMSTATUS	VT_BOOL	S-LINK の通信ステータスを取得します。 get_Value で SLPciGetCommStatus 関数を実行します。	○	-
@LASTERROR	VT_I4	最も最後に起きたエラーの番号を返します。 get_Value で SLPciGetLastError 関数を実行します。	○	-

3. サンプルプログラム

SL-PCI ボードに接続し、バンク切り替えと特定 S-Link アドレスの on/off の確認と制御を行うコードを示します。

List 3-1 **Sample.frm**

```
Dim eng As CaoEngine
Dim ctrl As CaoController
Dim cmd As CaoCommand
Dim val As CaoVariable

Private Sub Form_Load()

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

    ' SLPci ボードへ接続
    Set ctrl = eng.Workspaces(0).AddController("Sample", "CaoProv.SUNX.SLINK", "", "BSN=2")

    ' コマンドオブジェクトの生成
    Set cmd = ctrl.AddCommand("BANKCHANGE")

    ' 変数オブジェクトの生成
    Set val = Ext.AddVariable("I05", "Len=8")

End Sub

Private Sub Command1_Click()

    ' コマンドの実行
    cmd.Parameter = 1
    cmd.Execute 0

End Sub

Private Sub Command2_Click()

    ' 値の設定
    val = Text1.Text

End Sub

Private Sub Command3_Click()

    ' 値の取得
    Text2.Text = val

End Sub
```