

S-LinkV プロバイダ SUNX SL-VPCI 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 |
| 1.0.0.2 | 2011-03-11 | プロバイダ登録ツールに関する追記 |
| 1.0.0 | 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::AddVariable メソッド | 8 |
| 2.2.3. CaoController::get_ID プロパティ | 8 |
| 2.2.4. CaoController::get_VariableNames プロパティ | 8 |
| 2.2.5. CaoVariable::put_ID プロパティ | 9 |
| 2.2.6. CaoVariable::get_ID プロパティ | 9 |
| 2.2.7. CaoVariable::get_Value プロパティ | 9 |
| 2.2.8. CaoVariable::put_Value プロパティ | 9 |
| 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-LinkV プロバイダは SUNX 社製 SLV-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 SLV-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-LinkV プロバイダを使用するには、SLV-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-LinkV 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-LinkV API の対応表

| CAO API | | S-LinkV API |
|-------------------------------|-----------------------|--------------------|
| CaoController::FinalConstruct | | SLVCreate |
| CaoController::FinalRelease | | SLVClose |
| CaoVariable::put_Value | (変数名 : IO<アドレス>) | SLVOutBit |
| | | SLVOutByte |
| | (変数名 : STATUS<アドレス>) | SLVReadStatusArea |
| CaoVariable::get_Value | (変数名 : IO<アドレス>) | SLVInBit |
| | | SLVInByte |
| | (変数名 : STATUS<アドレス>) | SLVWriteStatusArea |
| | (変数名 : @VERSION) | SLVGetVersion |
| | (変数名 : @LASTERROR) | SLVGetLastError |
| | (変数名 : @IOPORT_COUNT) | SLVGetResource |
| | (変数名 : @IOPORT_BASE) | |
| | (変数名 : @IOPORT_LEN) | |
| | (変数名 : @IRQ_COUNT) | |
| | (変数名 : @IRQ_REGISTER) | |
| (変数名 : @IRQ_ATTRIBUTE) | | |

S-LinkV 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-LinkV プロバイダ

| | |
|----------------------|-------------------------------|
| ファイル名 | GaoProvSLinkV.dll |
| ProgID | GaoProv.SUNX.SLINKV |
| レジストリ登録 ¹ | regsvr32 GaoProvSLinkV.dll |
| レジストリ登録の抹消 | regsvr32 /u GaoProvSLinkV.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. SLV-PCI ボードのドライバをインストールしていないと, S-LinkV 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-LinkV 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. SLINKV",     // 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. SLINKV",
    "",                           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::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 2-4 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
(
  "I05"                     // 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 2-4 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 |

2.2.3. 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。

2.2.4. 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。

2.2.5. CaoVariable::put_ID プロパティ

ID を設定します。ID の意味は変数ごとに異なり、それぞれ以下のようになっています。

表 2-5 変数名ごとの ID プロパティの意味

| 変数名 | ID プロパティの意味 |
|----------------|-------------|
| IO | - |
| STATUS | ビット位置 |
| @VERSION | - |
| @LASTERROR | - |
| @IOPORT_COUNT | - |
| @IOPORT_BASE | IO ポート番号 |
| @IOPORT_LEN | IO ポート番号 |
| @IRQ_COUNT | - |
| @IRQ_REGISTER | IRQ 番号 |
| @IRQ_ATTRIBUTE | IRQ 番号 |

上記表で ID プロパティの意味が“-”となっている変数は ID プロパティを使用できません。

2.2.6. CaoVariable::get_ID プロパティ

2.2.5 で設定した ID を取得します。パラメータが設定されていないときは 0 を返します。

2.2.7. 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。

2.2.8. 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.</p> <p>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 SLVInBit 関数を実行します.</p> <p>put_Value で SLVOutBit 関数を実行します.</p> <p>Len=8 の場合 :</p> <p>get_Value で SLVInByte 関数を実行します.</p> <p>put_Value で SLVOutByte 関数を実行します.</p> | ○ | ○ |
| STATUS<アドレス> | VT_BOOL | <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>ID プロパティで読み出しのビット位置を指定します.</p> <p>get_Value で SLVStatusInBit 関数を実行します.</p> <p>put_Value で SLVStatusOutBit 関数を実行します.</p> | ○ | ○ |

2.4. エラーコード

S-LinkV 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 SLVGetVersion 関数を実行します。 | ○ | - |
| @LASTERROR | VT_I4 | 最後に起きたエラーの番号を返します。 get_Value で SLVGetLastError 関数を実行します。 | ○ | - |
| @IOPORT_COUNT | VT_I4 | IO ポートの総数を取得します。 get_Value で SLVGetResource 関数を実行します。 | ○ | - |
| @IOPORT_BASE | VT_I4 | IO ポートのベースアドレスを取得します。 ID プロパティで IO ポート番号を指定します。 get_Value で SLVGetResource 関数を実行します | ○ | - |
| @IOPORT_LEN | VT_I4 | IO ポート長を取得します。 ID プロパティで IO ポート番号を指定します。 get_Value で SLVGetResource 関数を実行します | ○ | - |
| @IRQ_COUNT | VT_I4 | IRQ の総数を取得します。 get_Value で SLVGetResource 関数を実行します | ○ | - |
| @IRQ_REGISTER | VT_I4 | IRQ レジスタを取得します。 ID プロパティで IRQ 番号を指定します。 get_Value で SLVGetResource 関数を実行します | ○ | - |
| @IRQ_ATTRIBUTES | VT_I4 | IRQ の属性を取得します。 ID プロパティで IRQ 番号を指定します。 get_Value で SLVGetResource 関数を実行します | ○ | - |

3. サンプルプログラム

SLV-PCI ボードに接続し、特定S-LinkV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.SLINKV", "", "BSN=2")

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

End Sub

Private Sub Command1_Click()

    ' 値の設定
    val = Text1.Text

End Sub

Private Sub Command2_Click()

    ' 値の取得
    Text2.Text = val

End Sub
```