

CIFX プロバイダ hilscher CIFX ボード

Version 1.0.6

ユーザーズ ガイド

Jan 17, 2017

【備考】

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::AddVariable メソッド	7
2.2.4. CaoController:: get_VariableNames プロパティ	7
2.2.5. CaoController::get_CommandNames プロパティ	7
2.2.6. CaoController::get_ID プロパティ	7
2.2.7. CaoVariable::get_Value プロパティ	7
2.2.8. CaoVariable::put_Value プロパティ	7
2.2.9. CaoCommand::Execute プロパティ	8
2.3. 変数一覧	8
2.3.1. コントローラクラス	8
3. 付録	10
3.1. CIFX の動作	10

1. はじめに

本書は hilscher 製 CIFX ボードにアクセスするためのプロバイダです。CIFX ボードの対象は CIFX50-DN, CIFX50-CC, CIFX50-DP, CIFX50-RE です。

詳細については hilscher 社の [cifx device driver manual](#) を参照してください。

注意: CIFX プロバイダを使用するには、CIFX ボードのデバイス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 2-1 を参照してください。

2. プロバイダの概要

2.1. 概要

CIFX プロバイダは CIFX ドライバの各 API をラップし、ORiN に沿った方法で操作を行うための 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。CIFX プロバイダを使用するにあたっては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 CIFX プロバイダ

ファイル名	GaoProvCIFX.dll
ProgID	GaoProv.hilscher.CIFX
レジストリ登録 ¹	regsvr32 GaoProvCIFX.dll
レジストリ登録の抹消	regsvr32 /u GaoProvCIFX.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。CIFX ボードのデバイスドライバをインストールしていないと、CIFX 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

CIFX プロバイダでは Controller オブジェクトの生成時に CIFX ボードとの通信処理を行います。

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

bstrCtrlName : [in] コントローラ名
 bstrProvName : [in] プロバイダ名. 固定値 =” CaoProv.hilscher.CIFX”.
 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
Board[=<ボード番号>]	接続先の CIFX ボード番号(デフォルト:0)
Slot[=<スロット番号>]	接続先 CIFX をロータリースイッチで識別. このオプションは Board より優先される
Timeout[=<通信タイムアウト>]	通信タイムアウト(デフォルト:10)
ResetTimeout[=<リセットタイムアウト>]	リセットタイムアウト(デフォルト:10000)

2.2.2. CaoController::AddCommand メソッド

使用できるコマンド名と詳細は表 2-3 を参照してください。

指定するオプション文字列はありません。

書式 AddCommand(<bstrName:BSTR > [,<bstrOption:BSTR>])

bstrName : [in] 任意の名前
 bstrOption : [in] オプション文字列(未使用)

表 2-3 コマンドクラス 予約語一覧

コマンド予約語	パラメータデータ型	パラメータ設定
Reset	なし	なし
Init	なし	なし
HostStart	なし	なし
HostStop	なし	なし

BusOn	なし	なし
BusOff	なし	なし

2.2.3. CaoController::AddVariable メソッド

このメソッドでは、CIFX ボードにアクセスする変数オブジェクトを生成します。

変数名には、2.3.1 の変数のみ使用することができます。これら以外の変数名を指定したときは、このメソッドはエラーを返します。

書式 AddVariable(<bstrName:BSTR> [,<bstrOption:BSTR>])

bstrName : [in] 任意の名前

bstrOption : [in] オプション文字列

オプション文字列には以下のものを用いて、バッファサイズを指定することができます。

表 2-4 CaoController::AddVariable のオプション文字列

オプション	意味
Size[=<データ長>]	読書きデータ長 (byte 単位 デフォルト:1)

2.2.4. CaoController:: get_VariableNames プロパティ

2.3.1 の変数を取得します。

2.2.5. CaoController::get_CommandNames プロパティ

コマンドクラスで使用できる予約語リストを出力します。

指定するオプションはありません。

詳しくは表 2-3 を参照してください。

2.2.6. CaoController::get_ID プロパティ

接続中の CIFX ボードのボード番号を取得します。

2.2.7. CaoVariable::get_Value プロパティ

変数に対応する情報を取得します。各変数の実装状況および取得データについては、2.3.1 を参照して下さい。

2.2.8. CaoVariable::put_Value プロパティ

変数に対応する情報を設定します。各変数の実装状況および設定データについては、2.3.1 を参照して下さい。

2.2.9. GaoCommand::Execute プロパティ

対応するコマンドを実行します。対応するコマンドについては 2.2.2 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 2-5 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		オプション
			get	put	Size
SYSPACKET	VT_ARRAY VT_UI1	システムパケットの取得, 設定を行います。 送信パケットは CIFX_PACKET 構造体です。	○	○	-
CHANNELPACKET	VT_ARRAY VT_UI1	チャンネルパケットの取得, 設定を行います。 送信パケットは CIFX_PACKET 構造体です。	○	○	-
IO?	VT_ARRAY VT_UI1	I/O 現在値の設定, 取得を行います。 変数名の最後の”?”にオフセット値を指定することができます。	○	○	○
FIO?	VT_ARRAY VT_UI1	I/O 現在値の設定, および設定値の取得を行います。 変数名の最後の”?”にオフセット値を指定することができます。	○	○	○

表 2-6 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		オプション
			get	put	Size
@FIRMWARE_NAME	VT_BSTR	ファームウェアのファイル名を取得します。	○	-	-
@FIRMWARE_VERSION	VT_BSTR	ファームウェアのバージョンを取得します。	○	-	-
@DEVICE_NO	VT_UI4	CIFX ボードの製品番号を取得します。	○	-	-
@SERIAL_NO	VT_UI4	CIFX ボードのシリアル番号を取得します。	○	-	-

@RECEIVE_PACKET_CNT	VT_UI4	CIFX ボードから受信できるパケットの数を取得します。	○	-	-
@SEND_PACKET_CNT	VT_UI4	CIFX ボードに送信できるパケットの数を取得します。	○	-	-
@CHANNEL_OPEN	VT_BOOL	Channel0 が開かれているかを取得します。 AddController 時に CIFX ボードにファームウェアがロードされていない場合にこの値が 0 となります。	○	-	-
@HOST_STATE	VT_BOOL	Channel0 の HOST の Ready 状態を取得します。 True:Ready 状態 False:Not Ready 状態	○	-	-
@BUS_STATE	VT_BOOL	Channel0 の BUS 状態を取得します。 True:BUS ON False:BUS OFF	○	-	-
@SLOT_NO	VT_UI4	CIFX ロータリースイッチの番号を取得します。	○	-	-
@HW_OPTIONS	VT_ARRAY VT_I2	CIFX のハードウェアオプション情報を取得します。	○	-	-
@MASTER_LICENSE	VT_BOOL	CIFX にマスタライセンスが入っているか判断します。 True:マスタライセンス有り False:マスタライセンス無し	○	-	-
@HW_REVISION	VT_UI4	ハードウェアリビジョンを取得します。	○	-	-

3. 付録

3.1. CIFX の動作

CIFX プロバイダを使用するには 2nd stage bootloader とフィールドネットワークのプロトコルスタック, 設定を行う WarmStartFile が必要です.

それぞれのファイルは下記に示すフォルダ構成で保存する必要があります.

```
+ <BASEDIR>
| -- (2nd stage bootloader)
| -- + <型番>_<シリアル番号> (デバイスのベースディレクトリ 例:1250100_20217)
|   | -- + Channel0 (Channel0 用ファイルのディレクトリ)
|     | -- cifXeis.nxf (プロトコルスタック. フィールドネットワークの種類毎にファイルが異なる)
|     | -- warmstart.dat (WarmStartFile)
```

<BASEDIR>はレジストリの HKLM¥SYSTEM¥CurrentControlSet¥Services¥CIFxDrv¥InstallDir に格納されています. デフォルトは C:¥Program Files¥cifX Device Driver となっています.

また, HKLM¥SYSTEM¥CurrentControlSet¥Services¥CIFxDrv¥Devices¥Board0 の DeviceNumber に型番, SerialNumber にシリアル番号が記載されています.

これらのファイルは CIFX 付属の CD に入っている”cifX setup”を使用して設定を行うと生成され, Windows 起動時に読み込まれます. warmstart ファイルのみ xChannelPutPacket で送信することで, 後から設定を変更することができます.

各ファイルの説明を表 3-1 に示します.

表 3-1 CIFX ファイル

ファイル	役割
2nd stage bootloader	<p>プロトコルスタック以前のボードの初期化に使用します。ファイルは CIFX を購入した際に添付される CD に保存されています。</p> <p>ファイル名 : NXCIF50-RTE.bin</p>
プロトコルスタック	<p>ファームウェア(拡張子 nxm または nxf)</p> <p>このファームウェアによってどのフィールドネットワークで動作するか決定します。ただし、ボードによって使用できるフィールドネットワークに制限があります。</p> <p>各 CIFX ボードとファイルの組み合わせは以下の組み合わせです。これ以外の組み合わせでファイルを保存した場合にはボードの使用はできません。</p> <p>ファイルは CIFX を購入した際に添付される CD に保存されています。</p> <p>CIFX50-DN:</p> <p> cifxdnm.nxf(DeviceNet Master), cifxdns.nxf(DeviceNet Slave)</p> <p>CIFX50-CC:</p> <p> cifxccs.nxf(CC-Link Slave)</p> <p>CIFX50-DP:</p> <p> cifxdpm.nxf(Profibus-DP Master), cifxdps.nxf(Profibus-DP Slave)</p> <p>CIFX50-RE:</p> <p> cifxecm.nxf(EtherCAT Master), cifxecs.nxf(EtherCAT Slave), cifxeim.nxf(EtherNet/IP Master), cifxeis.nxf(EtherNet/IP Slave), cifxomb.nxf(Modbus TCP/IP Server), cifxpls.nxf(PowerLink Slave), cifxpnm.nxf(Profinet Master), cifxpns.nxf(Profinet Slave), cifxs3s.nxf(SercosIII Slave)</p>
WarmStartFile	<p>各フィールドネットワークの設定を行うためのファイルです。このファイルは構造体のダンプファイルです。</p> <p>選択したフィールドネットワークによって異なる構造体を使用します。どの構造体を使用するのかは各フィールドネットワークの API マニュアルを参照してください。</p> <p>(例: EtnetNet/IP Adapter は</p> <p> EIP_APS_PACKET_WARMSTART_PRM_REQ_T)</p> <p>ファイル名 : warmstart.dat</p>