DIO プロバイダ

CONTEC DIO ボード

Version 1.1.3

ユーザーズ ガイド

May 15, 2017

【備考】

ORiN2SDK2.0.14 以降，このプロバイダはCONTEC 社の API-DIO(WDM)を使用しています．以前までの API-DIO(98PC)を使用する場合は，DIO98 プロバイダを使用してください．

【改版履歴】

バージョン	日付	内容
1.0.0.0	2006-02-23	初版
1.0.1.0	2008-02-06	マニュアル修正
1.0.2.0	2008-11-10	内容を WDM 版に変更
1.0.2.1	2010-02-10	エラーコード追加
1.1.0.0	2010-09-13	USB 機器対応 Variable：IN,OUT,IO のオプションに LEN=16,32 を追加
1.1.0.1	2011-03-11	プロバイダ登録ツールに関する追記
1.1.1.0	2012-05-25	“DeviceName”オプション追加
1.1.2.0	2012-05-28	メタモード追加.
1.1.2	2012-07-17	ドキュメントのバージョンルールを変更.
1.1.3	2013-07-12	“Size”オプション追加, “Len”オプション修正
	2017-05-15	マニュアル修正

【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::AddVariable メソッド 6
2.2.3. CaoController::Execute メソッド 7
2.2.4. CaoVariable::get_Attribute プロパティ 8
2.2.5. CaoVariable::get_Value プロパティ 8
2.2.6. CaoVariable::put_Value プロパティ 8
2.3. 変数一覧 9
2.3.1. コントローラクラス 9
2.4. エラーコード 9
2.5. CAO-DIO API 対応表 10

3. サンプルプログラム 11

1. はじめに

本書は，CONTEC 製 DIO ボードにアクセスするためのプロバイダである，DIO プロバイダのユーザーズガイドです．

詳細については，CONTEC 社 API-DIO (WDM) Help を参照して下さい．

注意：DIO プロバイダを使用するには，DIO 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．対象機器が PCI ボードであれば API-PAC（W32）、USB であれば API-USBP（WDM）よりインストールして下さい．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

DIO プロバイダは，CAO API を実行するときに対応する DIO (WDM) の API を実行します． CAO API と DIO (WDM) API の対応については表 2-7 を参照してください．

プロバイダの概要について記述します．

表 2-1 DIO プロバイダ

ファイル名	CaoProvDIO.dll
ProgID	CaoProv.CONTEC.DIO
レジストリ登録 [1]	regsvr32 CaoProvDIO.dll
レジストリ登録の抹消	regsvr32 /u CaoProvDIO.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．DIO ボードのドライバをインストールしていないと，DIO 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

DIO プロバイダでは Controller オブジェクトの生成時に DIO ボードとの接続処理を行います．接続時にコントローラ名でデバイス名を指定します．

```
AddController
(
    "<コントローラ名>",            // デバイス名.
    " CaoProv.CONTEC.DIO",        // 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.
)
```

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

オプション	意味
DeviceName=[<デバイス名>]	接続先ボードのデバイス名※1 デフォルト:“”（指定なし） “”指定なし場合，コントローラ名に指定した文字列をデバイス名とします．

2.2.2. CaoController::AddVariable メソッド

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

```
AddVariable
(
    "<変数名><論理番号>",          // 変数名.
    "<オプション>"                // オプション文字列.
)
```

変数名には，2.3.1 の変数に 10 進数の論理番号を付加した名前を使用します．

2.3.1 以外の変数名を指定したとき，または論理番号を 10 進数以外で指定したときは，このメソッドはエラーを返します．

論理番号の意味は，表 2-3 の Len オプションの値によって意味が変わります．

Len=1 のとき：論理ビット

Len=8 のとき：論理ポート

入力できる値の範囲については，CONTEC 社 API-DIO Help を参照して下さい．このメソッドでは，入力の範囲外を指定してもエラーは発生しません．get_Value，put_Value を実行した時にエラーが発生します．

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 CaoController::AddVariable のオプション文字列

オプション	意味
Len [=<1｜8>]	データ長の指定(bit) (デフォルト:1)
Size[=<データサイズ>]	データサイズの指定(デフォルト:1) 生成された変数オブジェクトはこのオプションで指定したサイズを一括で読書きします. (例 1) “IO8”, Len=1, Size=16 の場合 8bit 目から 23bit 目までを一括で読書きします. (例 2) “IO2”, Len=8, Size=4 の場合 2byte 目から 6byte 目までを一括で読書きします.
Filter=<設定値>	デジタルフィルタの設定値の指定 (デフォルト:デジタルフィルタなし) 有効範囲については, CONTEC 社 API-DIO Help を参照して下さい.

2.2.3. CaoController::Execute メソッド

割り込みイベントの設定を行います.

このメソッドでは第 1 引数に“SetInterrupt”, 第 2 引数にパラメータを配列で指定します. 詳細については以下を参照してください.

表 2-4 ControllerExecute のパラメータ

要素番号	データ型	説明
1	VT_I2	入力ビット番号. このビット番号は, ハードウェア上で割り込みとして使用可能なビットでなければなりません.
2	VT_I2	割り込み論理 0:割り込みをマスクする 1:入力値 0→1 2:入力値 1→0

```
Execute
(
    "SetInterrupt"          // コマンド名. 固定.
    "<パラメータ>"          // パラメータ(配列  [1]: 入力論理ビット番号, [2]: 割り込み論理)
)
```

割り込みメソッドは USB デバイスによっては対応していない機器があります. 詳細については API-DIO(WDM)のヘルプを参照し、割り込み機能関数:DioNotifyInterrupt が使用可能か確認して下さい.

2.2.4. CaoVariable::get_Attribute プロパティ

変数の Read/Write の属性情報を取得します.

表 2-5 属性とその値の対応表

属性	値
Read	1
Write	2

2.2.5. 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.

2.2.6. 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.

2.3. 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
IN?	VT_UI1[2] VT_UI1|VT_ARRAY[3]	データの取得を行います. 変数名の後ろに論理ポート番号をします. 例)“IN20”	○	-
OUT?	VT_UI1[2] VT_UI1|VT_ARRAY[3]	データの設定を行います. 変数名の後ろに論理ポート番号をします. 例)“OUT20” CaoVariable::get_Value 時にはリードバックを行います.	○	○
IO?	VT_UI1[2] VT_UI1|VT_ARRAY[3]	データの取得, 設定を行います. 変数名の後ろに論理ポート番号をします. 例)“IO20” CaoVariable::get_Value プロパティ時には変数名“IN”, CaoVariable::put_Value プロパティ時には変数名“OUT” と同様の動作をします.	○	○

2.4. エラーコード

DIO プロバイダでは, 固有のエラーコードとして DIO API で規定されたエラー番号を“0x8010000”でマスクした値を返します.

例) DIO API のエラー : 0x0003　→　CAO API のエラー : 0x80100003

DIO API の詳細については, CONTEC 社 API-DIO (WDM) Help を参照してください.

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.

[2] Len オプションで 1 または 8 を指定した場合
[3] Size オプションで 1 以外を指定した場合

2.5. CAO-DIO API 対応表

表 2-7 CAO-DIO API 対応表

CAO API	DIO API	備考
CaoWorkspace::AddController()	DioInit()	
CaoWorkspaces::Remove()	DioExit()	
CaoVariable::get_Value()	DioInpBit()	変数名“IN”又は“IO”の場合 Len=1, Size=1 を指定した場合
	DioInpByte()	変数名“IN”又は“IO”の場合 Len=8, Size=1 を指定した場合
	DioInpMultiBit()	変数名“IN”又は“IO”の場合 Len=1, Size>1 を指定した場合
	DioInpMultiByte()	変数名“IN”又は“IO”の場合 Len=8, Size>1 を指定した場合
	DioEchoBackBit()	変数名“OUT”の場合 Len=1, Size=1 を指定した場合
	DioEchoBackByte()	変数名“OUT”の場合 Len=8, Size=1 を指定した場合
	DioEchoBackMultiBit()	変数名“OUT”の場合 Len=1, Size>1 を指定した場合
	DioEchoBackMultiByte()	変数名“OUT”の場合 Len=8, Size>1 を指定した場合
CaoVariable::put_Value()	DioOutBit()	変数名 OUT”又は“IO”の場合 Len=1, Size=1 を指定した場合
	DioOutByte()	変数名 OUT”又は“IO”の場合 Len=8, Size=1 を指定した場合
	DioOutMultiByte()	変数名“OUT”又は“IO”の場合 Len=1, Size>1 を指定した場合
	DioOutMultiByte()	変数名“OUT”又は“IO”の場合 Len=8, Size>1 を指定した場合

DIO API の詳細については, CONTEC 社 API-DIO Help を参照して下さい.

3. サンプルプログラム

以下に変数“DPM”でデュアルポートメモリにアクセスするサンプルを示します.

List 3-1	Sample.frm

```
Private caoEng As CaoEngine
Private caoCtrl As CaoController
Private caoVar As CaoVariable

Private Sub Form_Load()

    Set caoEng = New CaoEngine
    Set caoCtrl = caoEng.Workspaces(0).AddController("DIOTest", " CaoProv.CONTEC.DIO ", "","")
    Set caoVar = caoCtrl.AddVariable("IO20", "")
End Sub

Private Sub CmdPut_Click()

    Dim data As Byte
    data = 1

    caoVar.Value = data

End Sub

Private Sub cmdGet_Click()

    Dim Ret As Variant

    Text1.Text = caoVar.Value

End Sub
```