DIO Provider

CONTEC DIO Board

Version 1.1.3

User's Guide

May 15, 2017

[Remarks]
This provider uses CONTEC API-DIO (WDM) for ORiN2SDK2.0.14 or later. Use the DIO98 provider for the earlier versions of API-DIO (98PC).

[Revision history]

Version	Date	Content
1.0.0.0	2006-02-23	First edition.
1.0.1.0	2008-02-06	Manual was corrected.
1.0.2.0	2008-11-10	Content was changed for WDM.
1.0.2.1	2010-02-10	Error codes were added.
1.1.0.0	2010-09-13	USB devices were supported. LEN=16,32 was added to Variable: IN, OUT, IO options.
1.1.0.1	2011-03-11	Information related to provider registration tool was added.
1.1.1.0	2012-05-25	"DeviceName" option was added.
1.1.2.0	2012-05-28	Meta mode was added.
1.1.2	2012-07-17	Version rule of the document was changed.
1.1.3	2013-07-12	"Size" option was added. "Len" option was corrected.
	2017-05-15	Manual was corrected.

[Hardware]

Model	Version	Notes

Contents

1. Introduction

This document is a user's guide of the DIO provider which is used to access CONTEC DIO board.
Refer to CONTEC API-DIO (WDM) Help for details.

NOTE: The DIO device driver of the DIO board needs to be installed to use the DIO provider. Install the driver from API-PAC (W32) for PCI board or from API-USBP (WDM) for USB. After installing it, register the provider in the registry with reference to Table 2-1.

2. Outline of Provider

2.1. Outline

The DIO provider executes DIO (WDM) API corresponding to CAO API at the time the CAO API is executed.

Refer to Table 2-7 for CAO API and corresponding DIO (WDM) API.

The following shows the outline of the provider.

Table 2-1 DIO provider

File name	CaoProvDIO.dll
ProgID	CaoProv.CONTEC.DIO
Registry registration[1]	regsvr32 CaoProvDIO.dll
Remove registry registration	regsvr32 /u CaoProvDIO.dll

[1] Provider registration can be executed by regsvr32.exe or RegCOM.exe ([Start] -> [ORiN2] -> [Tools]). The DIO board driver must be installed to register the DIO provider.

2.2. Methods and properties

2.2.1. CaoWorkspace::AddController method

The DIO provider establishes connection to the DIO board when the Controller object is created. The device name is specified with controller name when connection is established.

```
AddController
(
    "<Controller name>",                    // Device name.
    " CaoProv.CONTEC.DIO",                  // Provider name. Fixed.
    "<Machine name>",                       // Provider execution machine name.
    "<Option>"                              // Option character string.
)
```

Table 2-2 Option character string of CaoWorkspace::AddController

Option	Meaning
DeviceName=[<Device name>]	Device name of the board to be connected[*1] Default: "" (no value specified) For the case of "" (no value specified), the character string specified for controller name will be used as device name.

2.2.2. CaoController::AddVariable method

This method creates a variable object used to access the DIO board.

```
AddVariable
(
    "<Variable name><Logical number>",      // Variable name.
    "<Option>"                              // Option character string.
)
```

Attach a decimal logical number to a variable in 2.3.1, and use it as a variable name.
If a variable other than those in 2.3.1 is specified, or a logical number is not specified with a decimal, this method will return an error.
The meaning of logical number depends on the value of Len option in Table 2-3.

Len=1: Logical bit

Len=8: Logical port

Refer to CONTEC API-DIO Help for the input value range. This method does not generate an error even if a specified value is out of the input range. An error occurs when get_Value or put_Value is executed.

Following is a list of option string items.

Table 2-3 Option character string of CaoController::AddVariable

Option	Meaning
Len [=<1｜8｜16｜32>]	Specify data length (bit). (Default: 1)
Size[=<Data size>]	Specify data size（Default :1） Variable objects created will read/write the size of data specified by this option. (Example 1) If "IO8", Len=1, and Size=16 are specified, bit 8 through 23 will be read or written at one time. (Example 2) If "IO2", Len=8 and Size=4 are specified, byte 2 through byte 6 will be read or written at one time.
Filter=<Setting value>	Specify digital filter setting value. (Default: No digital filter) Refer to CONTEC API-DIO Help for the effective range.

2.2.3. CaoController::Execute method

Configures interrupt event setting.

This method specifies "SetInterrupt" for the first argument, and parameters in an array for the second argument. See below for details.

Table 2-4 Parameters for ControllerExecute

Element number	Data type	Explanation
1	VT_I2	Input bit number This bit number must be an available bit for interrupt on the hardware.
2	VT_I2	Interrupt logic 0: Mask interrupt 1: Input value 0 -> 1 2: Input value 1 -> 0

```
Execute
(
    "SetInterrupt"          // Command name. Fixed.
    "<Parameter>"           // Parameter (array [1]: Input logical bit number, [2]: Interrupt logic)
)
```

Some USB devices do not support the interrupt method. Refer to API-DIO (WDM) Help for details and check

if the interrupt function: DioNotifyInterrupt is available.

2.2.4. CaoVariable::get_Attribute property

Acquires Read/Write attribute information for variables.

Table 2-5 Attribute and corresponding value

Attribute	Value
Read	1
Write	2

2.2.5. CaoVariable::get_Value property

Acquires information corresponding to a variable. For the implementation status and acquired data of each variable, refer to 2.3.1.

2.2.6. CaoVariable::put_Value property

Configures information corresponding to a variable. For the implementation status and setting data of each variable, refer to 2.3.1.

2.3. Variable list

2.3.1. Controller class

Table 2-6 Controller class user variable list

Variable name	Data type	Explanation	Attribute	
			get	put
IN?	VT_UI1[*1] VT_UI1|VT_ARRAY[*2]	Acquire data. Specify the logical port number after the variable name. Example: "IN20"	√	-
OUT?	VT_UI1[*1] VT_UI1|VT_ARRAY[*2]	Set data. Specify the logical port number after the variable name. Example: "OUT20" Read-back is performed for CaoVariable::get_Value.	√	√
IO?	VT_UI1[*1] VT_UI1|VT_ARRAY[*2]	Acquire or set data. Specify the logical port number after the variable name. Example: "IO20" This performs the "IN" variable operation for CaoVariable::get_Value property, and the "OUT" variable operation for CaoVariable::put_Value property.	√	√

* 1: For the case 1 or 8 is specified for the Len option.

* 2: This applies when the Size option specifies other than 1.

2.4. Error code

The DIO provider returns an error number specified by DIO API after masking it with "0x8010000" as a unique error code.

Example: DIO API error 0x0003 -> CAO API error 0x80100003

Refer to CONTEC API-DIO (WDM) Help for details of DIO API.

For common errors of ORiN2, refer to the error code section in "ORiN2 Programming Guide".

2.5. CAO-DIO API reference table

Table 2-7 CAO-DIO API reference table

CAO API	DIO API	Remarks
CaoWorkspace::AddController()	DioInit()	
CaoWorkspaces::Remove()	DioExit()	
CaoVariable::get_Value()	DioInpBit()	For "IN" or "IO" variable Len = 1, Size = 1
	DioInpByte()	For "IN" or "IO" variable Len = 8, Size = 1
	DioInpMultiBit()	For "IN" or "IO" variable Len = 1, Size > 1
	DioInpMultiByte()	For "IN" or "IO" variable Len = 8, Size > 1
	DioEchoBackBit()	For "OUT" variable Len = 1, Size = 1
	DioEchoBackByte()	For "OUT" variable Len = 8, Size = 1
	DioEchoBackMultiBit()	For "OUT" variable Len = 1, Size > 1
	DioEchoBackMultiByte()	For "OUT" variable Len = 8, Size > 1
CaoVariable::put_Value()	DioOutBit()	For "OUT" or "IO" variable Len = 1, Size = 1
	DioOutByte()	For "OUT" or "IO" variable Len = 8, Size = 1
	DioOutMultiBit()	For "OUT" or "IO" variable Len = 1, Size > 1
	DioOutMultiByte()	For "OUT" or "IO" variable Len = 8, Size > 1

Refer to CONTEC API-DIO Help for details of DIO API.

3. Sample Program

The following sample program shows the code to access the dual port memory with the variable "DPM".

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
```