

ActiveX Control

For Visual Basic 2005.NET

[Version: 1.0]

1

Installing Windows Driver

Follow these steps:

1. Set up the Windows driver. (*For example: PIO-DIO*)

The driver is location at:

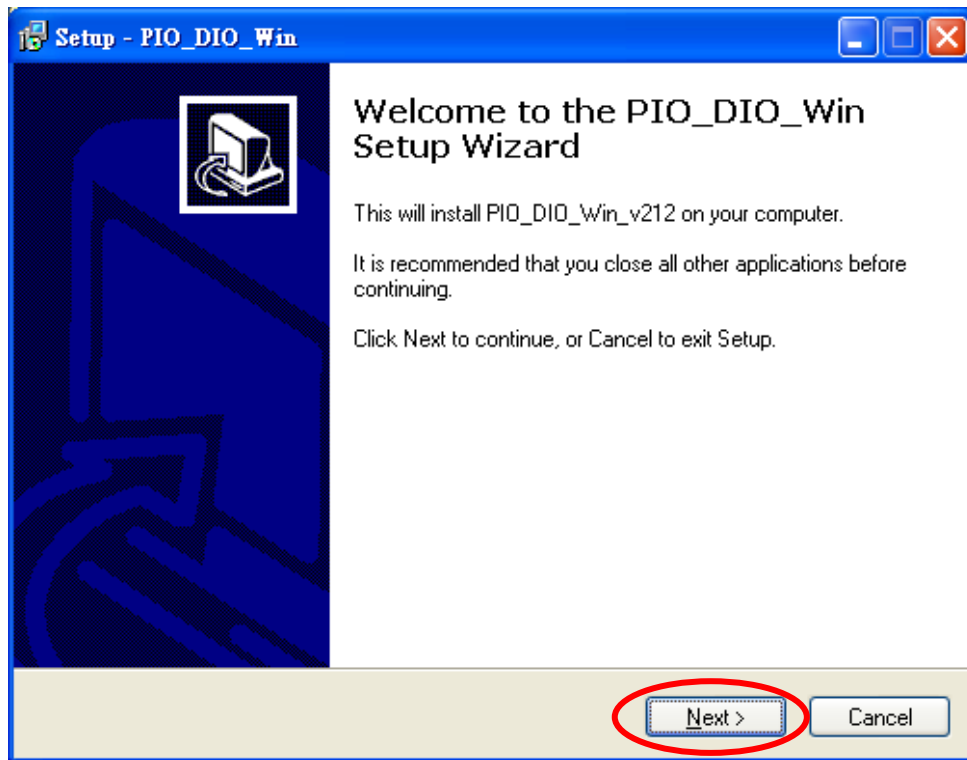
CD:\NAPDOS\PCI\PIO-DIO\dll_ocx\Driver

http://ftp.icpdas.com/pub/cd/iocard/pci/napdos/pci/pio-dio/dll_ocx/driver/

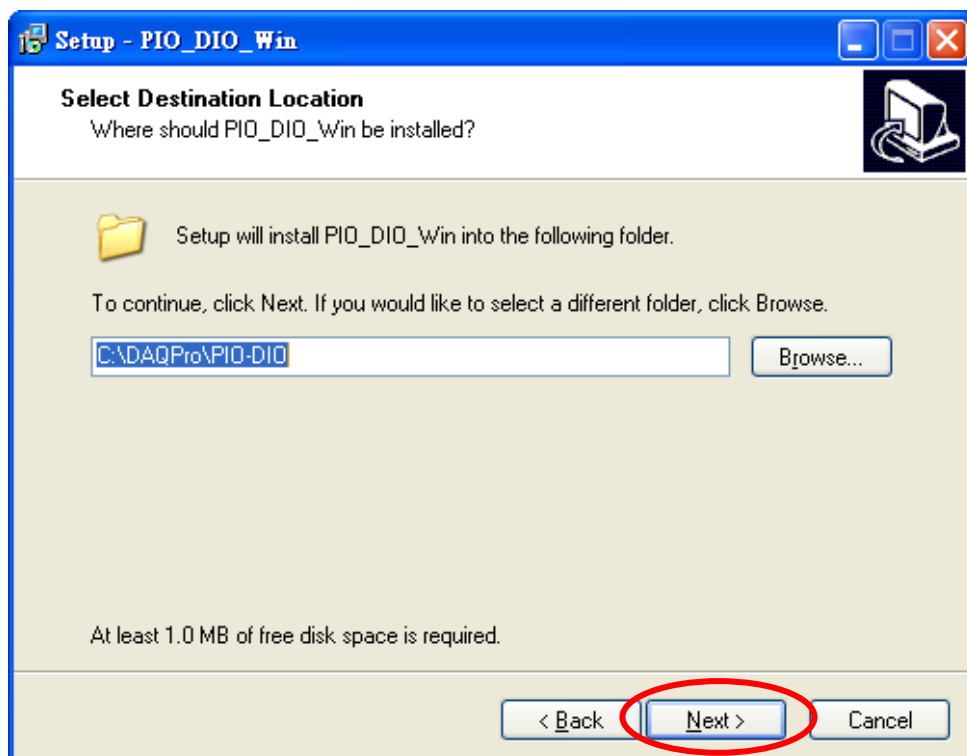


The Windows driver only supports Windows 98/NT/2000 and XP/2003/vista 32-bit versions.

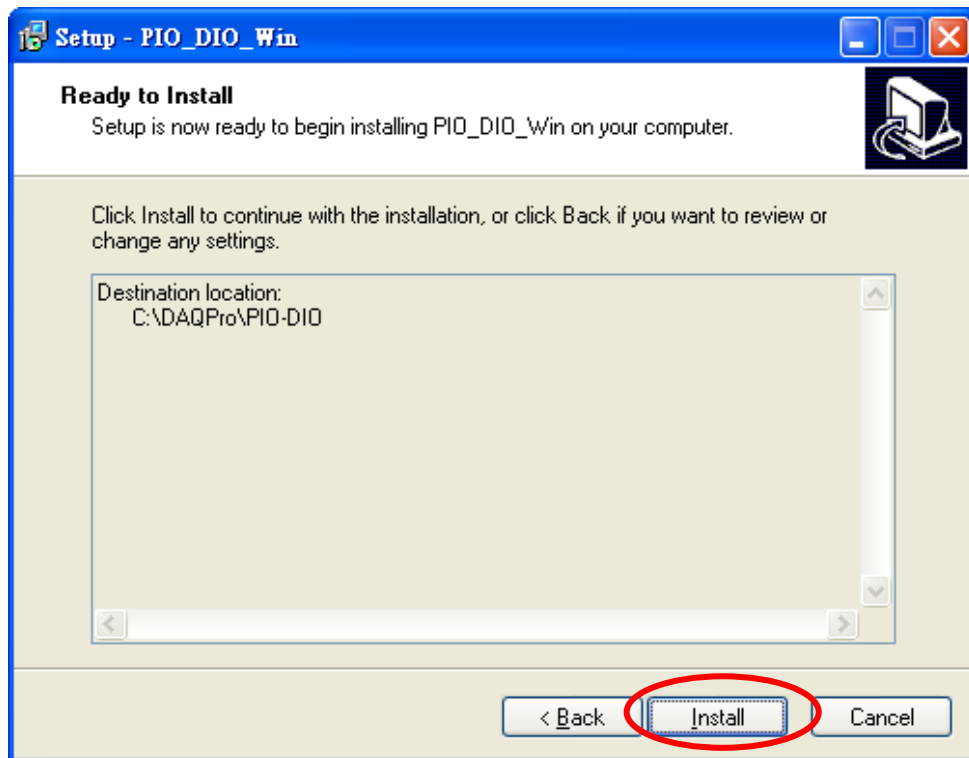
2. Click the “**Next >**” button to start the installation.



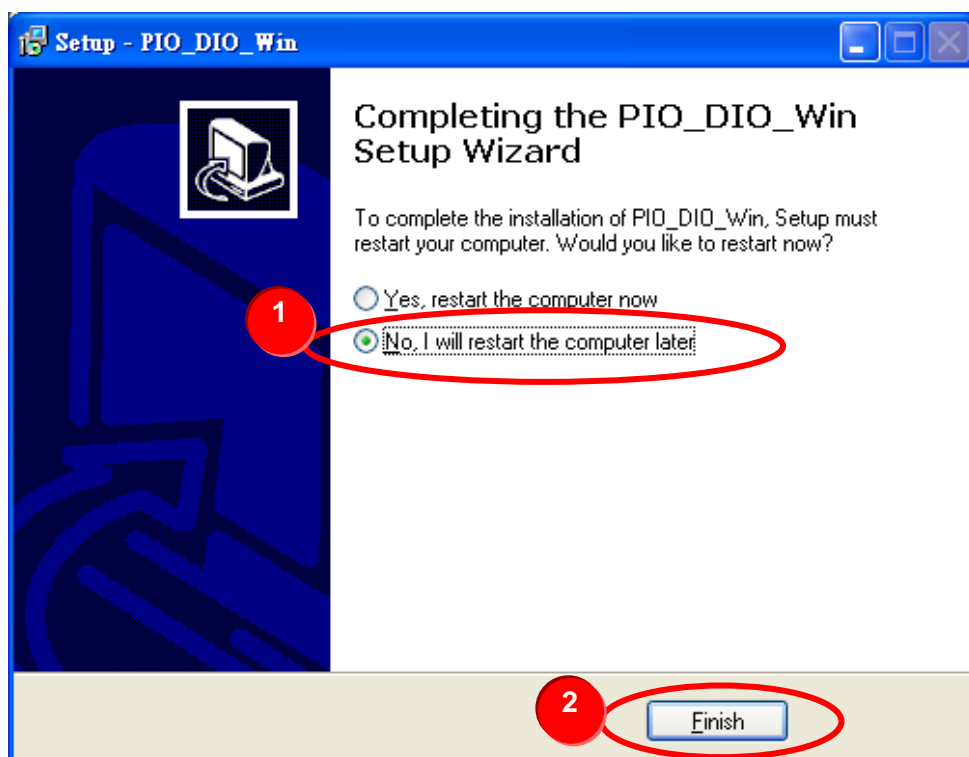
3. Click the “**Next >**” button to install the driver into the default folder.



4. Click the “**I**nstall” button to continue the installation.



5. Select the “**N**o, I will restart the computer later” and then click the “**F**inish” button.



2

Installing Hardware on PC

Follow these steps:

1. Shut down and power off your computer.
2. Remove the cover from the computer.
3. Select an unused PCI slot.
4. Carefully insert your I/O card into the PCI slot.
5. Replace the PC cover.
6. Power on the computer.

After powering-on the computer, please finished the Plug&Play steps according to the prompt message.

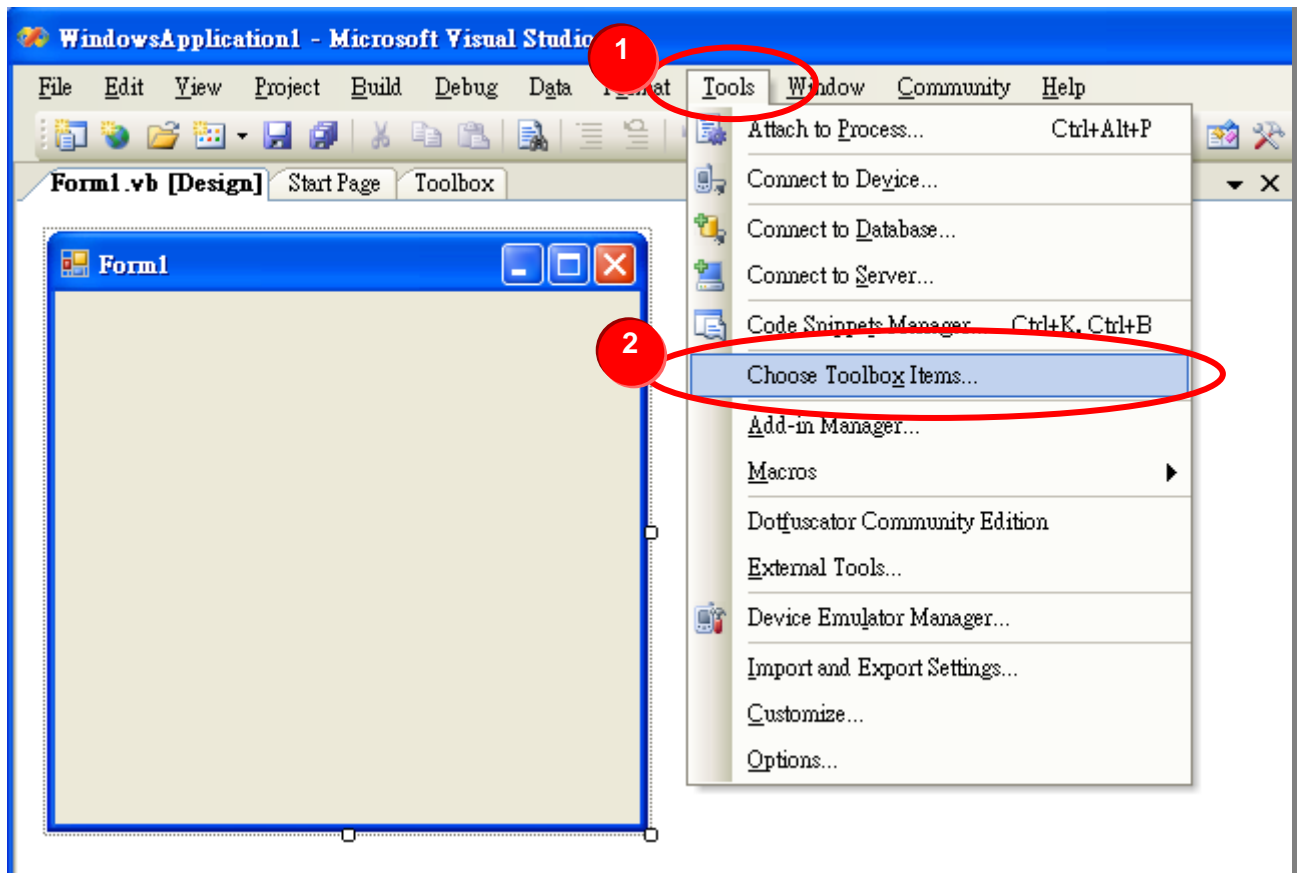
3

Using ActiveX Control (OCX)

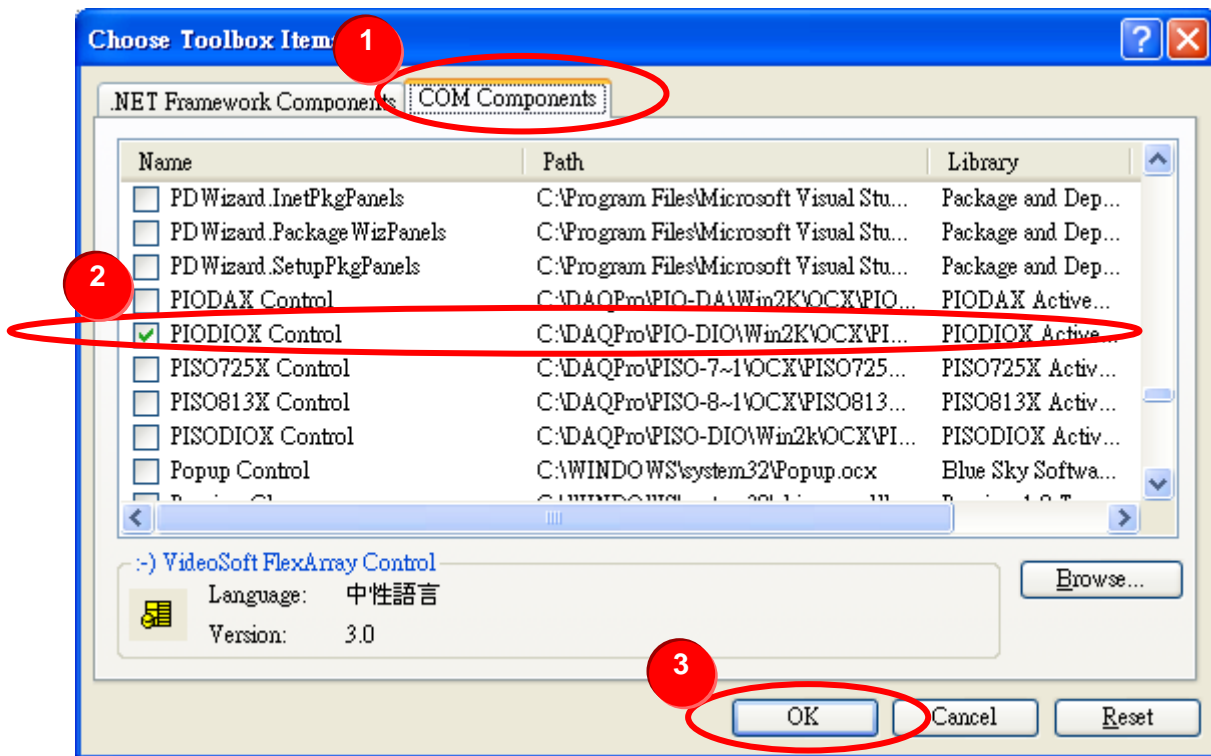
The ActiveX Control applied in VB.NET 2005,
Follow these steps:

1. Adding the ActiveX Control (OCX) into the VB.NET 2005 toolbox.

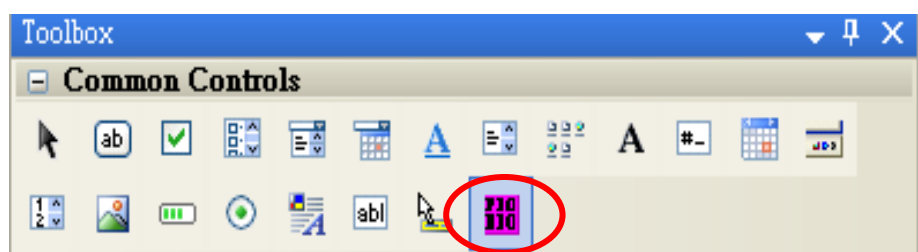
Click the “Tools/ Choose Toolbox Items...” in VB.NET 2005 menu.



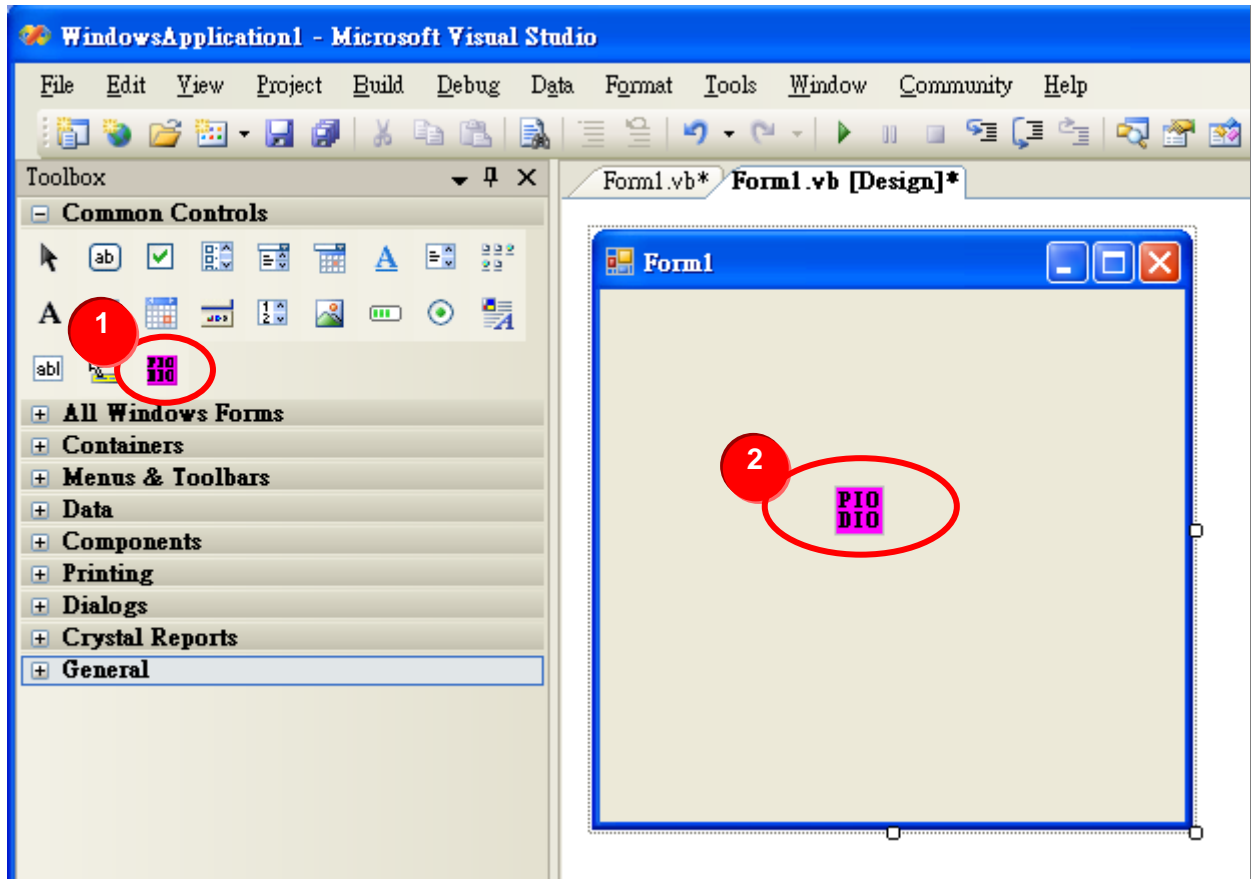
2. Check the item “**PIODIOX Control**” in the “**COM Components**” controls dialog box and then click “**OK**” button.



3. The icon of **PIODIO** (PCI Card OCX) will display in the Toolbox.



4. Create a new standard project. And then add a **PIODIO** (PCI Card OCX) component into this project.



NOTES:

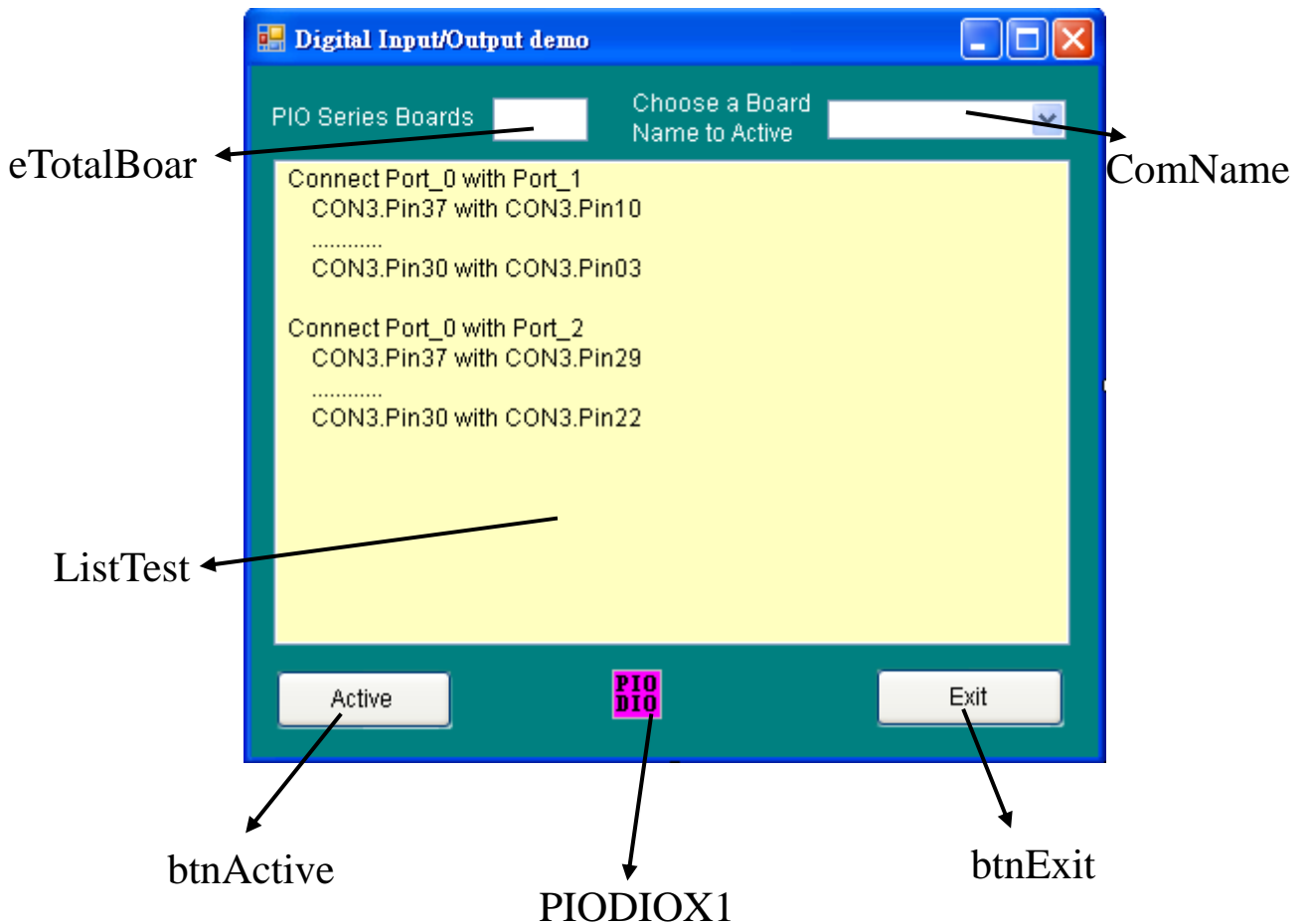
Please note, all OCX controls cannot correctly handle array type parameters on .NET environment since they are in different sub-systems. For using array type parameters, you have to call the Win32 DLL functions directly instead of using OCX controls on .NET.

4

Writing DIO sample program

Design your application and use the OCX functions.
Follow these steps:

1. Designing the GUI:



2. The following tables are the function lists for the specific PIODIOX.

property	BSTR ErrorString;
property	short ErrorCode;
property	short ActiveBoard;
method	short DriverInit();
method	long GetConfigAddressSpace(short nAddrNum);
method	short GetDIIVersion();
method	short GetDriverVersion();
method	short InputByte(long IBaseAddr);
method	long InputWord(long IBaseAddr);
method	void OutputByte(long IBaseAddr, short nOutputData);
method	void OutputWord(long IBaseAddr, long IOutputData);
method	short DigitalIn(short nPort);
method	void DigitalOut(short nPort, short nOutputValue);
method	void SetCounter(short nCounterNo, short nCounterMOde, long nCounterVal);
method	long ReadCounter(short nCounterNo, short nCounterMOde);
method	long ResetIrqCount();
method	void InstallIrq(long* hEvent, short nIrqSource, short nActiveMode);
method	void RemoveIrq();
method	void GetIrqCount();
method	long D48Freq();
method	void D48InstallIrq(long* IHandle, short nIrqMask, short nActiveMode);
method	void D48RemoveIrq();
method	long D48GetIrqcount();
method	void SaveIrqActiveFlag();
method	short GetIrqActiveFlag(short FlagNum);
method	void DriverClose();

The detail “**OCX Functions**” information. Please refer to:
ActiveX Control for PCI series boards User's Manual

http://ftp.icpdas.com/pub/cd/iocard/pci/napdos/pci/manual/ocx%28activex%29_manual_for_pci_pio_piso_cards.pdf

3. Using the functions:

Form1_Load

```
Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles MyBase.Load

    Dim i As Integer, wTotalBoards As Integer, iBoards As Integer
    Dim wSubAux As Long

    ' Initialize the driver and get the total boards
    wTotalBoards = PIODIOX1.DriverInit
    btnActive.Text = "Active"

    eTotalBoards.Text = wTotalBoards
    iBoards = 0
    For i = 0 To (wTotalBoards - 1)
        PIODIOX1.ActiveBoard = i ' Set the active board
        wSubAux = PIODIOX1.GetConfigAddressSpace(4) ' Get the AddrBase
        If 64 = wSubAux Then
            ComName.Items.Add(Str(i) & ":PIO-D24/56")
            iBoards = iBoards + 1
        End If
    Next i

    If iBoards = 0 Then
        ComName.Items.Add("No Device")
        btnActive.Enabled = False
    Else
        btnActive.Enabled = True
    End If
    ComName.SelectedIndex = 0

End Sub
```

btnEXit

```
Private Sub btnExit_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles btnExit.Click

    PIODIOX1.DriverClose() ' Release the device resource
    Me.Close()

End Sub
```

btnActive

```
Private Sub btnActive_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)  
Handles btnActive.Click
```

```
Dim InVal1, InVal2, InVal3 As Integer  
Dim i As Long
```

```
PIODIOX1.ActiveBoard = 0  
ListTest.Items.Clear()
```

```
ListTest.Items.Add("Setting Port 0 to Output-Mode and Port 1,2 to Input-mode")  
ListTest.Items.Add(" ")  
Thread.Sleep(10)
```

```
i = 1
```

```
While i <= &H80
```

```
PIODIOX1.DigitalOut(0, i) '// Port 0 digital output value
```

```
ListTest.Items.Add("Output Port 0 (Hex)= " + Hex(i))
```

```
Thread.Sleep(100)
```

```
InVal2 = PIODIOX1.DigitalIn(1) '// Port 1 digital input value
```

```
InVal3 = PIODIOX1.DigitalIn(2) '// Port 2 digital input value
```

```
ListTest.Items.Add(" Input Port 2, 1 (Hex)= " _  
+ Hex(InVal3) + " " _  
+ Hex(InVal2))
```

```
ListTest.Items.Add(" ")
```

```
Application.DoEvents()
```

```
Thread.Sleep(100)
```

```
i = i * 2
```

```
End While
```

```
PIODIOX1.DigitalIn(0) ' Port 0, Back to input mode
```

```
...  
...  
...  
...  
...  
...
```

```
ListTest.Items.Add(" The End ")
```

```
End Sub
```

AxPIODIOX1_OnError

```
Private Sub AxPIODIOX1_OnError(ByVal sender As System.Object, ByVal e As  
AxPIODIOXLib._DPIODIOXEvents_OnErrorEvent) Handles PIODIOX1.OnError
```

```
    'Get the error message
```

```
    MsgBox("Error Code:" + Str(PIODIOX1.ErrorCode) + Chr(13) + "Error Message:" +  
PIODIOX1.ErrorString)
```

```
End Sub
```

Form1_FormClosed

```
Private Sub Form1_FormClosed(ByVal sender As Object, ByVal e As  
System.Windows.Forms.FormClosedEventArgs) Handles Me.FormClosed
```

```
    PIODIOX1.DriverClose() ' Release the device resource
```

```
End Sub
```

5

Additional Information

- ✓ ActiveX Control for PCI series boards User's Manual
http://ftp.icpdas.com/pub/cd/iocard/pci/napdos/pci/manual/ocx%28activex%29_manual_for_pci_pio_piso_cards.pdf
- ✓ PCI1800X, PCI1602X, PCI1202X ActiveX Control User's Manual
http://ftp.icpdas.com/pub/cd/iocard/pci/napdos/pci/manual/ocx%28activex%29_manual_for_pci_1002_1202_1602_1800_1802.pdf
- ✓ ActiveX Control (OCX) Installation in VB, Delphi and BCB Manual
http://ftp.icpdas.com/pub/cd/iocard/pci/napdos/pci/manual/ocx%28activex%29_installation_in_vb_delphi_bcb.pdf