
在 VB.Net 2005 中使用 OMEGA I/O Card 的 DLL 檔案

本文件說明如何在 VB.Net 程式中使用 OMEGA I/O Card 的 DLL 檔案。

[下載安裝 DLL 驅動程式與 VB 範例程式]

多年來, OMEGA 完整的提供了全系列 PCI 與 ISA BUS I/O Card 在 Windows 上的 Driver 與 DLL 函式庫, 並提供 Microsoft Visual C++, Visual Basic, Borland C++ builder 與 Delphi 的範例程式暨原始碼幫助使用者開發自己的程式。以下, 將介紹如何在 VB.NET 程式中呼叫 DLL 的函式。

本說明以 Win2000/XP 搭配 PIO-D48 為範例。在開始之前請先安裝 Win2000/XP 的 DLL/OCX 驅動程式。以下為驅動程式 pio_dio_win2k_v207.exe 的 ftp 位址

ftp://ftp.icpdas.com/pub/cd/iocard/pci/napdos/pci/pio-dio/dll_ocx/win2k_xp/

或者您可以在以下的 CD 目錄中找到:

\\NAPDOS\PC\PIO-DIO\DLL_OCX\Win2K_XP\

安裝 DLL/OCX 驅動程式後, 請從下面的 ftp 網址下載 VB demo 程式, 解壓縮後, 選擇合適的範例程式

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

這個檔案也可以在以下的 CD 目錄找到

\\NAPDOS\PC\PIO-DIO\DLL_OCX\Demo\

選好的 Visual Basic 6.0 範例程式碼可以直接複製到 VB.NET 上再加以修改。

[從 Visual Basic 6.0 範例程式修改]

您可以在下面的 ftp 網址下載 dll_vb6_xxxxxx.exe 檔案:

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

或者從下述的 CD 目錄複製檔案到您的硬碟
\\NAPDOS\PC\PIO-DIO\DLL_OCX\Demo\

執行這個自動解壓縮檔案。參考解壓縮後的 PIODIO.bas 檔案與範例程式的架構建立您的 VB.Net 專案, 並使用 DllImport 關鍵字將 PIODIO.bas 中宣告的函式轉換成 VB.Net 的格式。

舉例來說, PIODIO.bas 中 PIODIO_InputByte 函式宣告方式如下。

```
Declare Function 函式名稱 Lib "XXXX.DLL" _  
    (ByVal 參數 資料型態) As 函數型態
```

把 VB6.0 轉換成 VB.Net 的格式

```
<DllImport("XXXX.DLL")> _  
    Public Function 函式名稱(ByVal 參數 資料型態) As 函數型態  
    END Function
```

範例 1

```
Declare Function PIODIO_InputByte Lib "PIODIO.dll" _  
    (ByVal address As Long) As Integer
```

將上述的程式宣告改寫為

```
<DllImport("PIODIO.dll")>_  
    Public Function PIODIO_InputByte(ByVal address As Integer) As Short  
    END Function
```

範例 2

```
Declare Sub PIODIO_OutputWord Lib "PIODIO.dll" _  
    (ByVal address As Long, ByVal dataout As Long)
```

改寫為

```
<DllImport("PIODIO.dll")>_  
    Public Sub PIODIO_OutputWord(ByVal address As Integer, ByVal dataout As Integer)  
    END Function
```

改寫 PIODIO.bas 檔完成後，可以使用以下的方法來呼叫函式

InVal1 = PIODIO_InputByte(wBaseAddr + &HC0);

InVal2 = PIODIO_InputByte(wBaseAddr + &HC4);

InVal3 = PIODIO_InputByte(wBaseAddr + &HC8);

[參考資料：資料型態映照圖]

Bytes	VB++ 6 data type	VB.net data type
4	ByVal a As Long	ByVal a As Integer
2	ByVal a As Integer	ByVal a As short
4	ByVal a As Single	ByVal a As Single
8	ByVal a As double	ByVal a As double
2	a As Integer	ByRef a As short
4	a As Single	ByRef a As Single

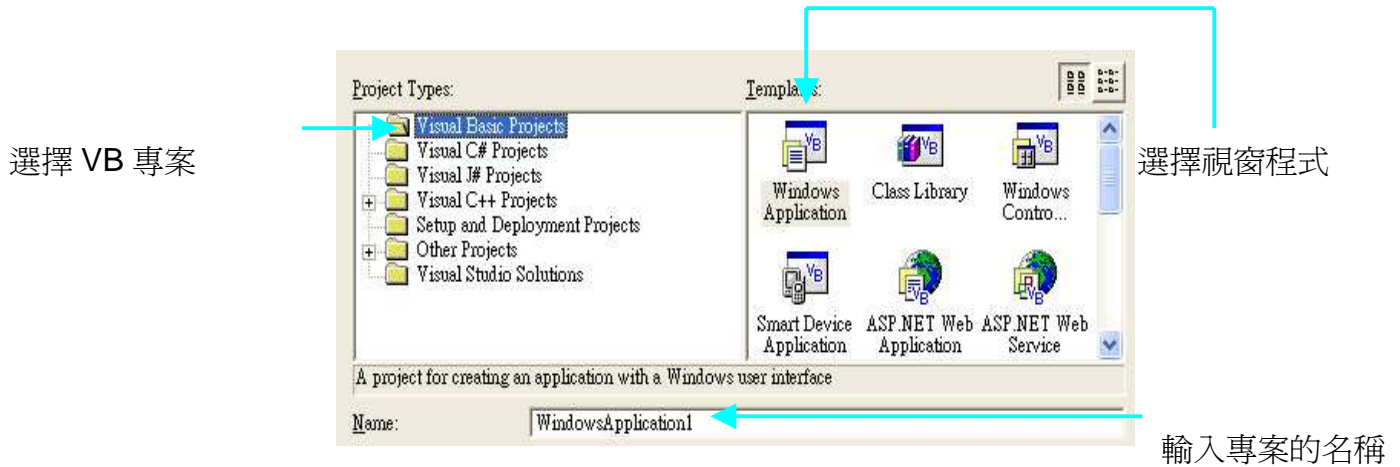
參考文件：PIODIO 軟體手冊，介紹 PIODIO.dll 所有函式功能與參數

ftp://ftp.icpdas.com/pub/cd/iocard/pci/napdos/pci/pio-dio/manual/pio-dio_dll_software_manual.pdf

以下為更詳細的步驟:

步驟 1.

開啓 Visual Studio.Net 然後點選 File->New ->Project。如下圖所示去建立一個新專案

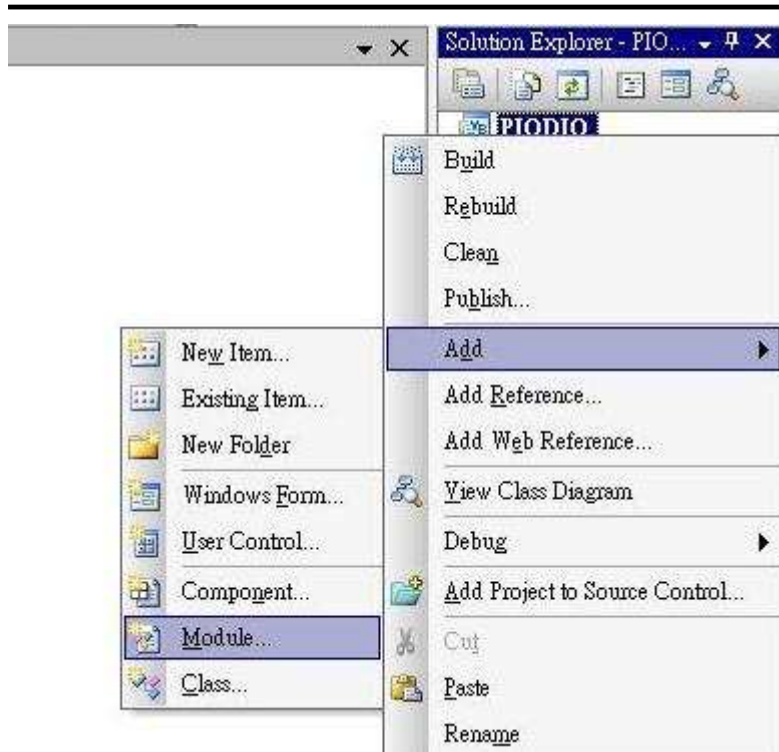


步驟 2.

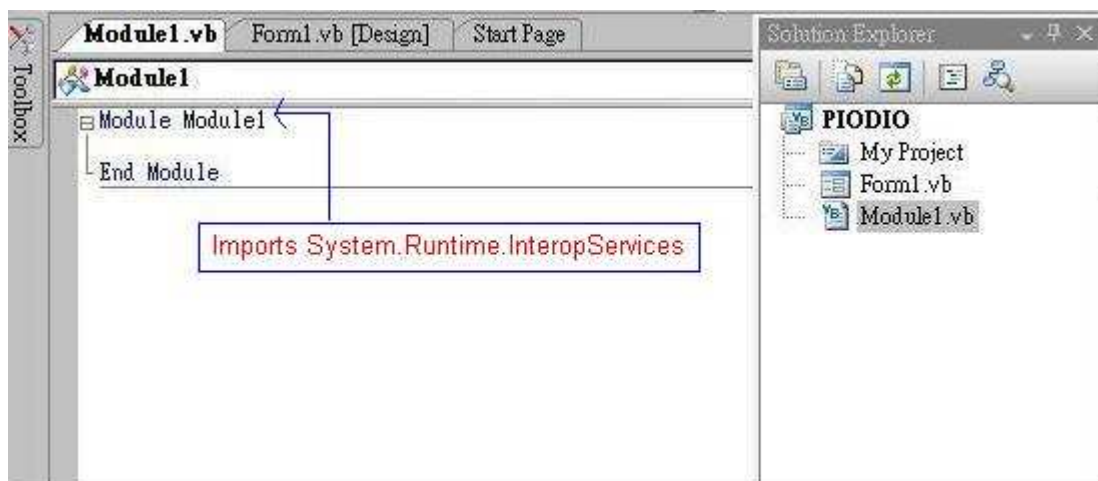
增加一個已經寫好的 VB.Net 宣告檔或者新增一個宣告然後貼上 PIODIO.bas 的內容, 再改寫成 VB.Net 的格式。

用滑鼠右鍵點擊專案的名稱並選擇 Add>>Module

選擇一個已經寫好的 VB.Net 宣告檔或者新增一個宣告檔



在此模組檔的開頭加入以下的紅字



從 PIODIO.bas 修改後的內容如下:

```
Imports System.Runtime.InteropServices
```

```
Module PIODIO
```

'Return Code

```
Public Const PIODIO_NoError = 0
Public Const PIODIO_DriverOpenError = 1
Public Const PIODIO_DriverNoOpen = 2
Public Const PIODIO_GetDriverVersionError = 3
Public Const PIODIO_InstallIrqError = 4
Public Const PIODIO_ClearIntCountError = 5
Public Const PIODIO_GetIntCountError = 6
Public Const PIODIO_RegisterApcError = 7
Public Const PIODIO_RemoveIrqError = 8
Public Const PIODIO_FindBoardError = 9
Public Const PIODIO_ExceedBoardNumber = 10
Public Const PIODIO_ResetError = 11
Public Const PIODIO_IrqMaskError = 12
Public Const PIODIO_ActiveModeError = 13
Public Const PIODIO_GetActiveFlagError = 14
Public Const PIODIO_ActiveFlagEndOfQueue = 15
```

'Define the Interrupt Signal Source

```
Public Const PIOD144_P2C0 = 0 'pin29 of CN1(37 pin D-type, pin1 to pin37)
Public Const PIOD144_P2C1 = 1 'pin28 of CN1(37 pin D-type, pin1 to pin37)
Public Const PIOD144_P2C2 = 2 'pin27 of CN1(37 pin D-type, pin1 to pin37)
Public Const PIOD144_P2C3 = 3 'pin26 of CN1(37 pin D-type, pin1 to pin37)
```

' Interrupt Channel for PIO-D48

```
Public Const PIOD48_INTCH0 = 1 ' INT_CHAN_0
Public Const PIOD48_INTCH1 = 2 ' INT_CHAN_1
Public Const PIOD48_INTCH2 = 4 ' INT_CHAN_2
Public Const PIOD48_INTCH3 = 8 ' INT_CHAN_3
```

'Interrupt ActiveMode for PIOD48_XXX functions

```
Public Const PIOD48_ActiveLow = 1 ' Active When Low
Public Const PIOD48_ActiveHigh = 2 ' Active When High
'to trigger a interrupt when high -> low
Public Const ActiveLow = 0
```

```
'to trigger a interrupt when low -> high
```

```

Public Const ActiveHigh = 1

'*****'
'Card ID          '
'*****'

Public Const PIO_D24 = &H800140
Public Const PIO_D48 = &H800130
Public Const PIO_D56 = &H800140
Public Const PIO_D64 = &H800120
Public Const PIO_D96 = &H800110
Public Const PIO_D144 = &H800100
Public Const PIO_D168 = &H98800150
Public Const PIO_D168A = &H800150

'*****'
'Test Function    '
'*****'

<DllImport("Piodio.dll")> _
    Public Function PIODIO_FloatSub(ByVal fA As Single, ByVal fB As Single) As Single
    End Function

<DllImport("Piodio.dll")> _
    Public Function PIODIO_ShortSub(ByVal nA As Short, ByVal nB As Short) As Short
    End Function

<DllImport("Piodio.dll")> _
    Public Function PIODIO_GetDllVersion() As Short
    End Function

'*****'
'Driver Function  '
'*****'

<DllImport("Piodio.dll")> _
    Public Function PIODIO_DriverInit() As Short
    End Function

<DllImport("Piodio.dll")> _
    Public Sub PIODIO_DriverClose()

```

```

End Sub

<DllImport("Piodio.dll")> _
    Public Function PIODIO_SearchCard(ByRef wBoards As Short, ByVal dwPIOCardID As Integer)_
As Short
    End Function

<DllImport("Piodio.dll")> _
    Public Function PIODIO_GetDriverVersion(ByRef wDriverVersion As Short) As Short
    End Function

<DllImport("Piodio.dll")> _
    Public Function PIODIO_GetConfigAddressSpace(ByVal wBoards As Short, ByRef wAddrBase As_
Integer, ByRef wIrqNo As Short, ByRef wSubVendor As Short, ByRef wSubDevice As Short, ByRef_
wSubAux As Short, ByRef wSlotBus As Short, ByRef wSlotDevice As Short) As Short
    End Function

<DllImport("Piodio.dll")> _
    Public Function PIODIO_ActiveBoard(ByVal wBoardNo As Short) As Short
    End Function

<DllImport("Piodio.dll")> _
    Public Function PIODIO_WhichBoardActive() As Short
    End Function

'*****'
'DIO Function
'*****'

<DllImport("Piodio.dll")> _
    Public Sub PIODIO_OutputByte(ByVal wBaseAddr As Integer, ByVal bOutputValue As Short)
    End Sub

<DllImport("Piodio.dll")> _
    Public Function PIODIO_InputByte(ByVal wBaseAddr As Integer) As Short
    End Function

'*****'

```

```

'Interrupt Function      '
'*****'

<DllImport("piodio.dll")> _
    Public Function PIODIO_IntInstall(ByVal wboards As Short, ByRef hEvent As Integer, ByVal _
wInterruptSource As Short, ByVal wActiveMode As Short) As Short
    End Function

<DllImport("piodio.dll")> _
    Public Function PIODIO_IntRemove() As Short
    End Function

<DllImport("piodio.dll")> _
    Public Function PIODIO_IntGetCount(ByRef intIntCount As Integer) As Short
    End Function

<DllImport("Piodio.dll")> _
    Public Function PIODIO_IntResetCount() As Short
    End Function

'*****'
'PIOD48 Counter Function      '
'*****'

<DllImport("Piodio.dll")> _
    Public Sub PIOD48_SetCounter(ByVal dwBase As Integer, ByVal wCounterNo As Short, ByVal _
bCounterMode As Short, ByVal wCounterValue As Integer)
    End Sub

<DllImport("Piodio.dll")> _
    Public Function PIOD48_ReadCounter(ByVal dwBase As Integer, ByVal wCounterNo As Short, _
ByVal bCounterMode As Short) As Integer
    End Function

<DllImport("Piodio.dll")> _
    Public Sub PIOD48_SetCounterA(ByVal wCounterNo As Short, ByVal bCounterMode As Short, _
ByVal wCounterValue As Integer)
    End Sub

```

```

<DllImport("Piodio.dll")> _
    Public Function PIOD48_ReadCounterA(ByVal wCounterNo As Short, ByVal_
bCounterMode As Short) As Integer
    End Function

'*****'
'PIOD48 Interrupt Function          '
'*****'

<DllImport("Piodio.dll")> _
    Public Function PIOD48_IntInstall(ByVal wBoardNo As Short, ByRef hEvent As Integer, ByVal_
wIrqMask As Short, ByVal wActiveMode As Short) As Short
    End Function

<DllImport("Piodio.dll")> _
    Public Function PIOD48_IntRemove() As Short
    End Function

<DllImport("Piodio.dll")> _
    Public Function PIOD48_IntGetActiveFlag(ByRef bActiveHighFlag As Short, ByRef_
bActiveLowFlag As Short) As Short
    End Function

<DllImport("Piodio.dll")> _
    Public Function PIOD48_IntGetCount(ByRef dwIntCount As Integer) As Short_
    End Function

'*****'
'PIOD64 Counter Function          '
'*****'

<DllImport("Piodio.dll")> _
    Public Sub PIOD64_SetCounter(ByVal dwBase As Integer, ByVal wCounterNo As Short, ByVal_
bCounterMode As Short, ByVal wCounterValue As Integer)
    End Sub

<DllImport("Piodio.dll")> _
    Public Function PIOD64_ReadCounter(ByVal dwBase As Integer, ByVal wCounterNo As Short,_
ByVal bCounterMode As Short) As Integer
    End Function

```

```

<DllImport("Piodio.dll")> _
    Public Sub PIOD64_SetCounterA(ByVal wCounterNo As Short, ByVal bCounterMode As Short,
ByVal wCounterValue As Integer)
    End Sub

<DllImport("Piodio.dll")> _
    Public Function PIOD64_ReadCounterA(ByVal wCounterNo As Short, ByVal_
bCounterMode As Short) As Integer
    End Function

'*****'
'PIOD48 frequence MeasurementFunction  '
'*****'

<DllImport("Piodio.dll")> _
    Public Function PIOD48_Freq(ByVal wAddrBase As Integer) As Integer
    End Function
End Module

```

步驟 3.

在原始碼的開頭增加下面這幾行

```

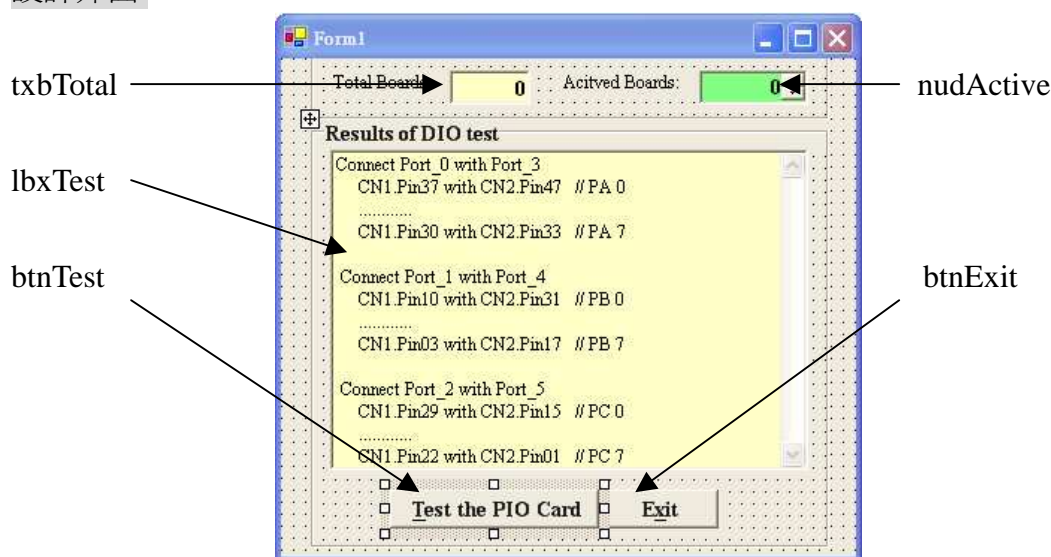
Imports System
Imports System.Drawing
Imports System.Collections
Imports System.ComponentModel
Imports System.Windows.Forms
Imports System.Data
Imports System.Runtime.InteropServices
Imports System.Threading

```

步驟 4.

設計程式，在程式中呼叫 DLL 函式

設計介面:



設計介面

函式使用:

```
Imports System.ComponentModel
```

```
Imports System.Windows.Forms
```

```
Imports System.Data
```

```
Imports System.Runtime.InteropServices
```

```
Imports System.Threading
```

```
Public Class Form1
```

```
    Inherits System.Windows.Forms.Form
```

```
    Dim wAddrBase As Long
```

```
    Dim wIrqNo As Integer
```

```
    Dim wSubVendor As Integer
```

```
    Dim wSubDevice As Integer
```

```
    Dim wSubAux As Integer
```

```
    Dim wSlotBus As Integer
```

```
    Dim wSlotDevice As Integer
```

```
    Dim wInitialCode As Integer
```

```
    Dim wBoards As Integer
```

```
    Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs)_
```

```
        Handles MyBase.Load
```

```

wInitialCode = PIODIO_DriverInit()
nudActive.Value = 0

If wInitialCode <> 0 Then
    MsgBox("Driver initialize error!!!", , "PIODIO Card Error!!!")
    btnTest.Enabled = False
    Exit Sub
End If

If PIODIO_SearchCard(wBoards, PIO_D48) <> 0 Then

    MsgBox("Search Card ERROR!!!")
    btnTest.Enabled = False
    Exit Sub
End If

btnTest.Enabled = True
tbxTotal.Text = wBoards
nudActive.Minimum = 0
nudActive.Maximum = wBoards - 1
End Sub

Private Sub btnExit_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)_
Handles btnExit.Click
    PIODIO_DriverClose()
    Me.Close()
End Sub

Private Sub btnTest_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)_
Handles btnTest.Click

    Dim wRetVal As Integer 'GetconfigAddressSpace Value
    Dim InVal0 As Integer
    Dim InVal1 As Integer
    Dim InVal2 As Integer
    Dim j As Integer

    lbxTest.Items.Clear()

```

```

If Val(nudActive.Value) > Val(tbxTotal.Text) - 1 Or Val(nudActive.Value) < 0 Then
    lbxTest.Items.Add("Invalid board number,Please Retry!!!")
    Exit Sub
End If

wRetVal = PIODIO_GetConfigAddressSpace(Val(nudActive.Value), wAddrBase, wIrqNo,_
wSubVendor, wSubDevice, wSubAux, wSlotBus, wSlotDevice)
If wRetVal <> 0 Then
    lbxTest.Items.Add("Get Config-Address-Space Error !!!")
    Exit Sub
End If

'*****'
'Enable All DI/DO port          '
'*****'

btnTest.Enabled = False
lbxTest.Items.Add("Enable All DI/DO")
lbxTest.Items.Add(" ")
PIODIO_OutputByte(wAddrBase, 1)      'enable DI/DO
lbxTest.Items.Add("")

lbxTest.Items.Add("Setting Port 0, 1, 2 to Output-Mode")
PIODIO_OutputByte((wAddrBase + &HCC), &H80)

lbxTest.Items.Add("Setting Port 3, 4, 5 to Input-Mode")
PIODIO_OutputByte((wAddrBase + &HDC), &H9B)
lbxTest.Items.Add(" ")
lbxTest.Items.Add("Outut Port 0, 1, 2 Input Port 3, 4, 5 ")
j = 1
While j <= &HFF
    PIODIO_OutputByte(wAddrBase + &HC0, j)  'Port 0
    PIODIO_OutputByte(wAddrBase + &HC4, j)  'Port 1
    PIODIO_OutputByte(wAddrBase + &HC8, j)  'Port 2

    InVal0 = PIODIO_InputByte(wAddrBase + &HD0) 'Port 3
    InVal1 = PIODIO_InputByte(wAddrBase + &HD4) 'Port 4
    InVal2 = PIODIO_InputByte(wAddrBase + &HD8) 'Port 5
    lbxTest.Items.Add("Output Port 0, 1, 2 (Hex)= " + Hex(j) + " " + Hex(j) + " " +
Hex(j))

```

```

        lbxTest.Items.Add(" Input Port 3, 4, 5 (Hex)= " + Hex(InVal0) + " " + Hex(InVal1)_
+ " " + Hex(InVal2))
        j = j * 2
        Thread.Sleep(100)
        Application.DoEvents()

End While

lbxTest.Items.Add("Setting Port 3, 4, 5 to Output-Mode")
PIODIO_OutputByte((wAddrBase + &HDC), &H80)
lbxTest.Items.Add("Setting Port 0, 1, 2 to Input-Mode")
PIODIO_OutputByte((wAddrBase + &HCC), &H9B)
lbxTest.Items.Add("")
lbxTest.Items.Add("Output Port 3, 4, 5 Input Port 1, 2, 3 ")

j = 1
While j <= &HFF
    PIODIO_OutputByte(wAddrBase + &HD0, j) 'Port 3
    PIODIO_OutputByte(wAddrBase + &HD4, j) 'Port 4
    PIODIO_OutputByte(wAddrBase + &HD8, j) 'Port 5

    InVal0 = PIODIO_InputByte(wAddrBase + &HC0) 'Port 1
    InVal1 = PIODIO_InputByte(wAddrBase + &HC4) 'Port 2
    InVal2 = PIODIO_InputByte(wAddrBase + &HC8) 'Port 3

    lbxTest.Items.Add("Output Port 3, 4, 5 (Hex)= " + Hex(j) + " " + Hex(j) + " " _
+ Hex(j))
    lbxTest.Items.Add(" Input Port 0, 1, 2 (Hex)= " + Hex(InVal0) + " " + Hex(InVal1)_
+ " " + Hex(InVal2))
    j = j * 2
    Thread.Sleep(100)
    Application.DoEvents()

End While

lbxTest.Items.Add(" ")
lbxTest.Items.Add(" Test End ")
btnTest.Enabled = True

End Sub

End Class

```