Using DLL of ICPDAS I/O card in VB.Net 2005

This document describes how to use the ICPDAS I/O card DLL file in a VB.Net application.

[DLL driver and demo files related information]

In the past, ICPDAS has provided the relevant DLL files for various I/O cards for customers to drive I/O cards in Microsoft Visual C++, Visual Basic, Borland C++ builder and Delphi. By following the instructions below, it will be possible to use the DLL files in a VB.NET application.

The following instructions will use the PIO-D48 add-on card in Win2000/XP as a demo. Before this issue, please install the DLL/OCX driver for Win2000/XP first. Download the pio_dio_win2k_v207.exe file from the ftp site:

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

or from the attached CD path:

\NAPDOS\PCI\PIO-DIO\DLL_OCX\Driver\

After installing the DLL/OCX driver, download the existing VB sample program from the ftp site:

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

or from the attached CD path:

\NAPDOS\PCI\PIO-DIO\DLL_OCX\Demo\

The source code of Visual Basic 6.0 sample programs can be copied, pasted and modified to VB.NET code.

[To modify from Visual Basic 6.0]

Download dll_vb6_xxxxxx.exe file from the ftp site:

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

or from the attached CD path:
\NAPDOS\PCI\PIO-DIO\DLL_OCX\Demo\

Extract the file to a local directory and select a suitable demo program. Refer to the PIODIO.bas and the demo program structure to create your VB.Net project. In VB.Net project, you will use “DllImport” to decorate the function declarations in PIODIO.bas.

For example, function “PIODIO_InputByte” is declared in the PIODIO.bas as:

```
Declare Function FunctionName Lib "XXXX.dll" _
        (ByVal arg1 As DataType) As FunctionType
```

To import the function into VB.Net project, you need to delete the red text in above declaration and use “DllImport” decoration as adding the following two lines in blue as below:

```
<DllImport("XXXX.dll")> _
    Public Function FunctionName (ByVal arg As _
    DataType) As FunctionType
    END Function
```

Refer to example

Example 1:

```
Declare Function PIODIO_InputByte Lib "PIODIO.dll" _
        (ByVal address As Long) As Integer
Convert to VB.NET
<DllImport("PIODIO.dll ")> _
    Public Function PIODIO_InputByte(ByVal address As
Integer) As Short
    END Function
```

Example2:

```
Declare Sub PIODIO_OutputWord Lib "PIODIO.dll" _
          (ByVal address As Long, ByVal dataout As Long)
Convert to VB.NET
<DllImport("PIODIO.dll ")> _
        Public Sub PIODIO_OutputWord(ByVal address As Integer_,
ByVal dataout As Integer)
        END Function
```

[Data type mapping table]

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

You can refer to a software manual

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

for more information about the functions in PIODIO.dll.

After adding the above two lines, the function can be called in an application in the following manner:

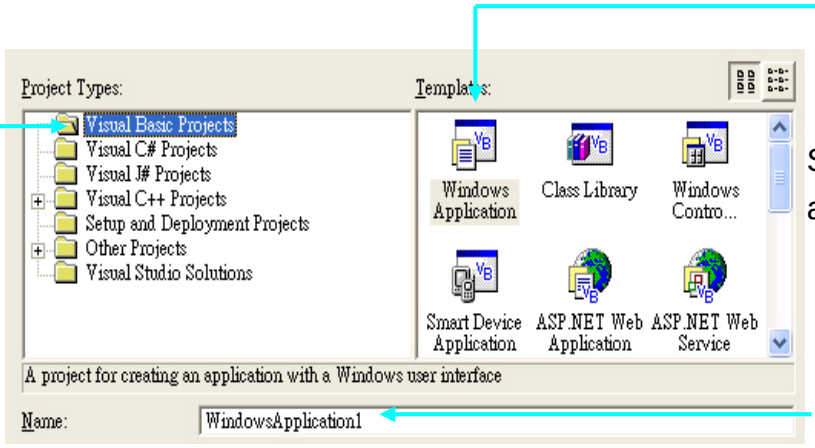
```
InVal1 = PIODIO_InputByte(wBaseAddr + &HC0);
InVal2 = PIODIO_InputByte(wBaseAddr + &HC4);
InVal3 = PIODIO_InputByte(wBaseAddr + &HC8);
```

The details description of the procedure is as follows:

Step 1.

Start Visual Studio .Net and go to File->New ->Project. Refer to the following figure to create a new project.

Select the VB Project.



Select the Windows application.

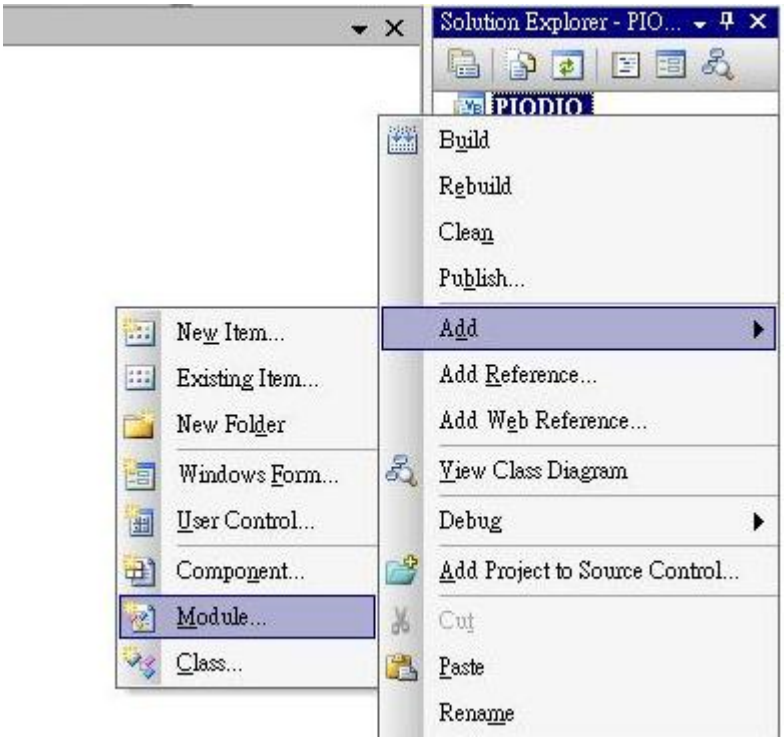
Enter the project name here.

Step 2.

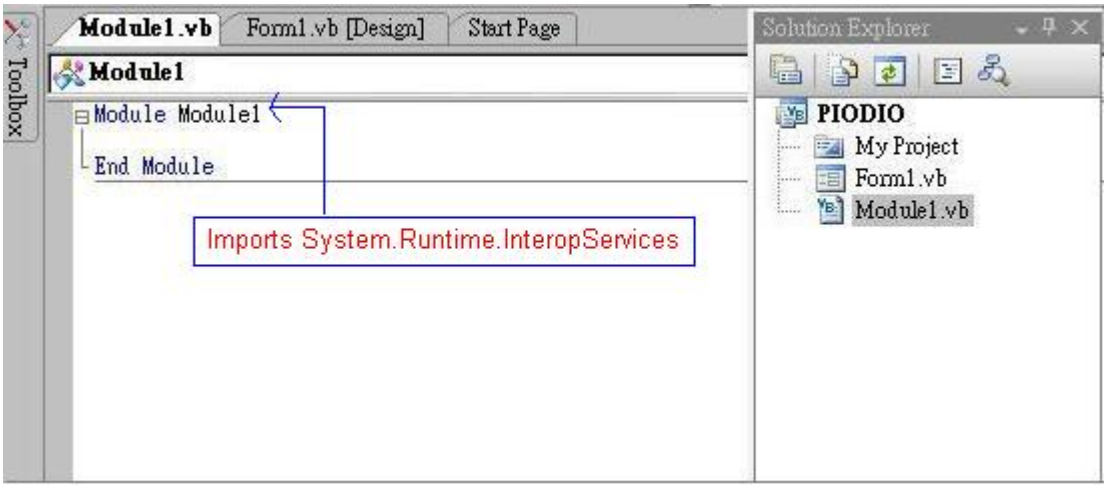
Add an existing function declaration file or a new module file and edit it from the declaration file in VB6.0 demo.

Right-click the project name and select Add >>Module

Select an existing declaration file in VB.Net format or a new file and modify it from the PIODIO.bas.



Add the description at the start of the module file



The module file modified form PIODIO.bas is as:

```
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
```

Step 3.

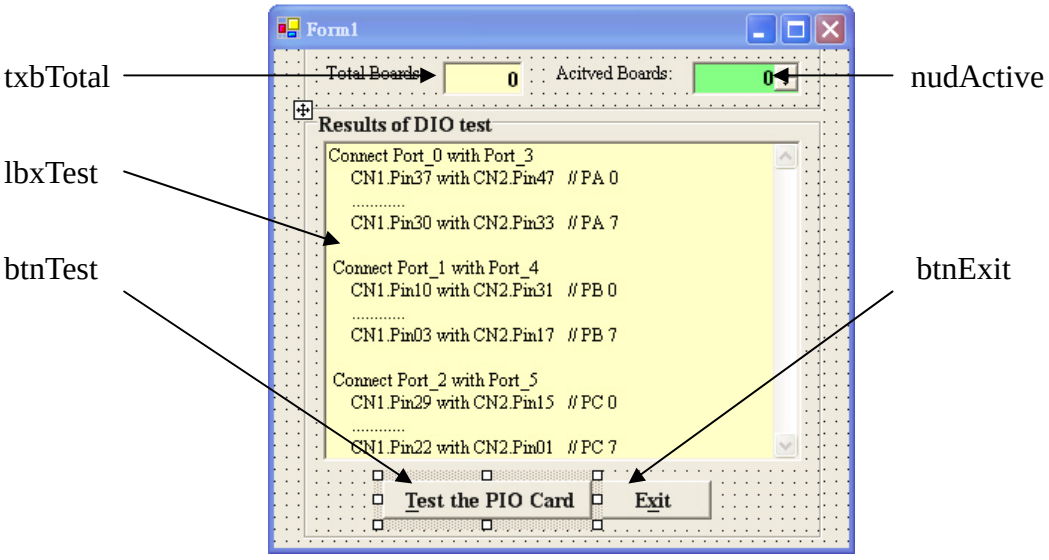
Add the following lines at the start of the source code.

```
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
```

Step 4.

Design your application and use the DLL functions.

Designing the GUI:



Designing the GUI

Using the functions:

```
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)         'enble 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 Iutput-Mode")
        PIODIO_OutputByte((wAddrBase + &HCC), &H9B)
        lbxTest.Items.Add("")
        lbxTest.Items.Add("Outut 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
```

Writer: Dan Huang (2006/07)