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

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

[DLL driver and demo file related information]

In the past, ICPDAS has provided the relevant DLL files for our various I/O cards for use in applications that were developed by our customers using Microsoft Visual C++, Visual Basic, Borland C++ builder and Delphi. By following the instructions in this document, it will be possible to use our DLL 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_v205.exe file from the ftp site:

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

or from the attached CD path:

\NAPDOS\PCI\PIO-DIO\DLL_OCX\Win2K_XP\

After installing the DLL/OCX driver, select the suitable demo and download the existing VC 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 then be copied, pasted and modified to VB.NET code.

[To modify from Visual Basic 6.0]

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

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. Refer to the PIODIO.bas and the program structure of the old demo program to create your VB.Net project. Into your class, insert the name of the functions you will use in your application with “DllImport”. For example, imagine an application has a class named PIODIO that will use a function called “PIODIO_InputByte” which is declared in the PIODIO.bas file.

To import

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

into the class, insert the following two lines into your code:

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

    End Function
```

Refer to

ftp://ftp.icpdas.com/pub/cd/iocard/pci/napdos/pci/piso-dio/manual/PISO-DIO_Win32_SDK_Manual.pdf

for more information about using ICPDAS functions.

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

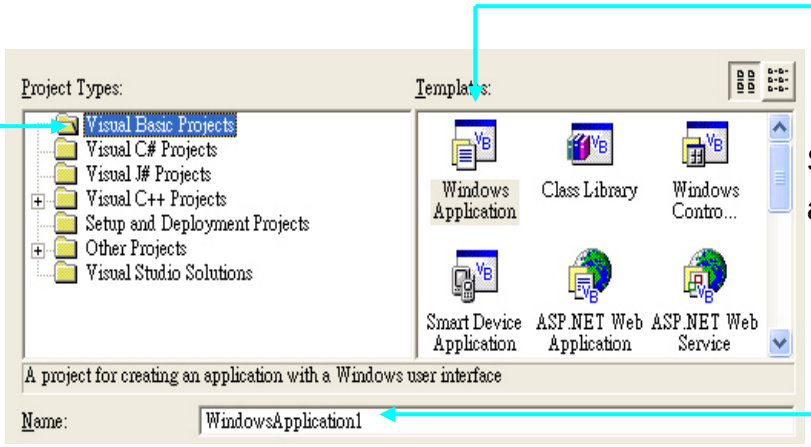
Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

InVal1 = cpiodio.PIODIO_InputByte((ushort)(wBaseAddr + 0xC0));
InVal2 = cpiodio.PIODIO_InputByte((ushort)(wBaseAddr + 0xC4));
InVal3 = cpiodio.PIODIO_InputByte((ushort)(wBaseAddr + 0xC8));

A 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 the following lines at the start of the 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 3.
Import the function declaration from the PIODIO.bas into your code.

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

The function declarations in the original PIODIO.bas file:

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

' define the interrupt signal source
Global Const PIOD144_P2C0 = 0 'pin29 of CN1(37 pin D-type, pin1 to pin37)
Global Const PIOD144_P2C1 = 1 'pin28 of CN1(37 pin D-type, pin1 to pin37)
Global Const PIOD144_P2C2 = 2 'pin27 of CN1(37 pin D-type, pin1 to pin37)
Global Const PIOD144_P2C3 = 3 'pin26 of CN1(37 pin D-type, pin1 to pin37)

' Interrupt Channel for PIO-D48
Global Const PIOD48_INTCH0 = 1   ' INT_CHAN_0
Global Const PIOD48_INTCH1 = 2   ' INT_CHAN_1
Global Const PIOD48_INTCH2 = 4   ' INT_CHAN_2
Global Const PIOD48_INTCH3 = 8   ' INT_CHAN_3
' Interrupt ActiveMode for PIOD48_XXX functions
Global Const PIOD48_ActiveLow = 1     ' Active When Low
Global Const PIOD48_ActiveHigh = 2    ' Active When High

' to trigger a interrupt when high -> low
Global Const PIODIO_ActiveLow = 0
' to trigger a interrupt when low -> high
Global Const PIODIO_ActiveHigh = 1
```

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

```
' ID
Global Const PIO_D168 = &H98800150    ' 168 * D/I/O
Global Const PIO_D168A = &H800150     ' 168A * D/I/O
Global Const PIO_D144 = &H800100      ' 144 * D/I/O
Global Const PIO_D96 = &H800110       ' 96 * D/I/O
Global Const PIO_D64 = &H800120       ' 64 * D/I/O
Global Const PIO_D56 = &H800140       ' D24 + 16I + 16O
Global Const PIO_D48 = &H800130       ' 48 * D/I/O
Global Const PIO_D24 = &H800140       ' 24 * D/I/O


' The Test functions
Declare Function PIODIO_ShortSub Lib "PIODIO.dll" _
        (ByVal a As Integer, ByVal b As Integer) As Integer
Declare Function PIODIO_FloatSub Lib "PIODIO.dll" (ByVal a As Single, ByVal b As Single) As Single
Declare Function PIODIO_GetDllVersion Lib "PIODIO.dll" () As Integer

' The Driver functions
Declare Function PIODIO_DriverInit Lib "PIODIO.dll" () As Integer
Declare Sub PIODIO_DriverClose Lib "PIODIO.dll" ()
Declare Function PIODIO_SearchCard Lib "PIODIO.dll" _
        (wBoards As Integer, ByVal dwPIOPISOCardID As Long) As Integer
Declare Function PIODIO_GetDriverVersion Lib "PIODIO.dll" (wDriverVersion As Integer) As Integer
Declare Function PIODIO_GetConfigAddressSpace Lib "PIODIO.dll" _
        (ByVal wBoardNo As Integer, wAddrBase As Long, wIrqNo As Integer, _
         wSubVendor As Integer, wSubDevice As Integer, wSubAux As Integer, _
         wSlotBus As Integer, wSlotDevice As Integer) As Integer

Declare Function PIODIO_ActiveBoard Lib "PIODIO.dll" (ByVal wBoardNo As Integer) As Integer
Declare Function PIODIO_WhichBoardActive Lib "PIODIO.dll" () As Integer

' DIO functions
Declare Sub PIODIO_OutputByte Lib "PIODIO.dll" (ByVal address As Long, ByVal dataout As Integer)
Declare Sub PIODIO_OutputWord Lib "PIODIO.dll" (ByVal address As Long, ByVal dataout As Long)
Declare Function PIODIO_InputByte Lib "PIODIO.dll" (ByVal address As Long) As Integer
Declare Function PIODIO_InputWord Lib "PIODIO.dll" (ByVal address As Long) As Long

' Interrupt functions
```

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

```
Declare Function PIODIO_IntInstall Lib "PIODIO.dll" _
        (ByVal wBoard As Integer, hEvent As Long, _
         ByVal wInterruptSource As Integer, _
         ByVal wActiveMode As Integer) As Integer
Declare Function PIODIO_IntRemove Lib "PIODIO.dll" () As Integer
Declare Function PIODIO_IntGetCount Lib "PIODIO.dll" (dwIntCount As Long) As Integer
Declare Function PIODIO_IntResetCount Lib "PIODIO.dll" () As Integer

' PIOD48 Counter functions
Declare Sub PIOD48_SetCounter Lib "PIODIO.dll" _
        (ByVal dwBase As Long, ByVal wCounterNo As Integer, _
         ByVal bCounterMode As Integer, ByVal wCounterValue As Long)
Declare Function PIOD48_ReadCounter Lib "PIODIO.dll" _
        (ByVal dwBase As Long, ByVal wCounterNo As Integer, _
         ByVal bCounterMode As Integer) As Long
Declare Sub PIOD48_SetCounterA Lib "PIODIO.dll" _
        (ByVal wCounterNo As Integer, ByVal bCounterMode As Integer, _
         ByVal wCounterValue As Long)
Declare Function PIOD48_ReadCounterA Lib "PIODIO.dll" _
        (ByVal wCounterNo As Integer, ByVal bCounterMode As Integer) As Long

' PIOD48 Interrupt functions
Declare Function PIOD48_IntInstall Lib "PIODIO.dll" _
        (ByVal wBoardNo As Integer, hEvent As Long, _
         ByVal wIrqMask As Integer, ByVal wActiveMode As Integer) As Integer
Declare Function PIOD48_IntRemove Lib "PIODIO.dll" () As Integer
Declare Function PIOD48_IntGetActiveFlag Lib "PIODIO.dll" _
        (bActiveHighFlag As Integer, bActiveLowFlag As Integer) As Integer
Declare Function PIOD48_IntGetCount Lib "PIODIO.dll" (dwIntCount As Long) As Integer

' PIOD64 Counter functions
Declare Sub PIOD64_SetCounter Lib "PIODIO.dll" _
        (ByVal dwBase As Long, ByVal wCounterNo As Integer, _
         ByVal bCounterMode As Integer, ByVal wCounterValue As Long)
Declare Function PIOD64_ReadCounter Lib "PIODIO.dll" _
        (ByVal dwBase As Long, ByVal wCounterNo As Integer, _
         ByVal bCounterMode As Integer) As Long
Declare Sub PIOD64_SetCounterA Lib "PIODIO.dll" _
```

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

```
        (ByVal wCounterNo As Integer, ByVal bCounterMode As Integer, _
         ByVal wCounterValue As Long)
Declare Function PIOD64_ReadCounterA Lib "PIODIO.dll" _
        (ByVal wCounterNo As Integer, ByVal bCounterMode As Integer) As Long

' PIOD48 Frequency Measurement Functons
Declare Function PIOD48_Freq Lib "PIODIO.dll" (ByVal dwBase As Long) As Long
Declare Function PIOD48_FreqA Lib "PIODIO.dll" () As Long
```

Declare a class and Import the function to be used in the application:

```
Public Class PIODIO
    Private DriverOpened As Integer = 0

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

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

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

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

    ' ****************************************
    <DllImport("Piodio.dll")> _
    Public Shared Sub PIODIO_OutputByte(ByVal wBaseAddr As Integer, ByVal bOutputValue As
Integer)
```

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

```
    End Sub

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

    ' ****************************************
    Public Sub OutputByte(ByVal wBaseAddr As Integer, ByVal bValue As Integer)
        PIODIO_OutputByte(wBaseAddr, bValue)
    End Sub

    Public Function InputByte(ByVal wBaseAddr As Integer) As Integer
        InputByte = PIODIO_InputByte(wBaseAddr)
    End Function

    ' ****************************************
    Public Sub New() 'constroctor
        DriverOpened = 0
    End Sub

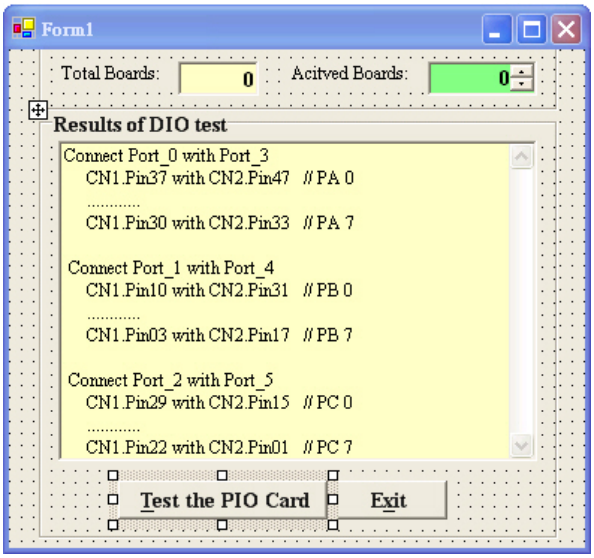
    Protected Overrides Sub Finalize()
        If DriverOpened <> 0 Then
            DriverOpened = 0
            PIODIO_DriverClose()
        End If
        MyBase.Finalize()
    End Sub
End Class
```

Step 4.

Designing your application and Using the DLL function in your application.

Designing the GUI:

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000



Designing the GUI

Using the function:

```
Namespace PIODIO_D48
    Public Class Form1
        Inherits System.Windows.Forms.Form

#Region " Windows Form Designer generated code "
        Public Sub New()
            MyBase.New()
            'This call is required by the Windows Form Designer.
            InitializeComponent()
            'Add any initialization after the InitializeComponent() call
        End Sub

        'Form overrides dispose to clean up the component list.
        Protected Overloads Overrides Sub Dispose(ByVal disposing As Boolean)
            If disposing Then
                If Not (components Is Nothing) Then
                    components.Dispose()
                End If
            End If
            MyBase.Dispose(disposing)
        End Sub
        'Required by the Windows Form Designer
```

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

```
    Private components As System.ComponentModel.IContainer

    'NOTE: The following procedure is required by the Windows Form Designer
    'It can be modified using the Windows Form Designer.
    'Do not modify it using the code editor.
    Friend WithEvents groupBox2 As System.Windows.Forms.GroupBox
    Friend WithEvents btnExit As System.Windows.Forms.Button
    Friend WithEvents btnTestDIO As System.Windows.Forms.Button
    Friend WithEvents lbxRst As System.Windows.Forms.ListBox
    Friend WithEvents groupBox1 As System.Windows.Forms.GroupBox
    Friend WithEvents nudActiveBoards As System.Windows.Forms.NumericUpDown
    Friend WithEvents label2 As System.Windows.Forms.Label
    Friend WithEvents label1 As System.Windows.Forms.Label
    Friend WithEvents tbxBoards As System.Windows.Forms.TextBox
    <System.Diagnostics.DebuggerStepThrough()> Private Sub InitializeComponent()
        Me.groupBox2 = New System.Windows.Forms.GroupBox
        Me.btnExit = New System.Windows.Forms.Button
        Me.btnTestDIO = New System.Windows.Forms.Button
        Me.lbxRst = New System.Windows.Forms.ListBox
        Me.groupBox1 = New System.Windows.Forms.GroupBox
        Me.nudActiveBoards = New System.Windows.Forms.NumericUpDown
        Me.label2 = New System.Windows.Forms.Label
        Me.label1 = New System.Windows.Forms.Label
        Me.tbxBoards = New System.Windows.Forms.TextBox
        Me.groupBox2.SuspendLayout()
        Me.groupBox1.SuspendLayout()
        CType(Me.nudActiveBoards, System.ComponentModel.ISupportInitialize).BeginInit()
        Me.SuspendLayout()
        '
        'groupBox2
        '
        Me.groupBox2.Controls.Add(Me.btnExit)
        Me.groupBox2.Controls.Add(Me.btnTestDIO)
        Me.groupBox2.Controls.Add(Me.lbxRst)
        Me.groupBox2.Font = New System.Drawing.Font("Times New Roman", 10.2!,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, CType(0, Byte))
        Me.groupBox2.Location = New System.Drawing.Point(24, 49)
        Me.groupBox2.Name = "groupBox2"
```

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

```
            Me.groupBox2.Size = New System.Drawing.Size(416, 343)
            Me.groupBox2.TabIndex = 3
            Me.groupBox2.TabStop = False
            Me.groupBox2.Text = "Results of DIO test"
            '
            'btnExit
            '
            Me.btnExit.Location = New System.Drawing.Point(232, 296)
            Me.btnExit.Name = "btnExit"
            Me.btnExit.Size = New System.Drawing.Size(96, 32)
            Me.btnExit.TabIndex = 2
            Me.btnExit.Text = "E&xit"
            '
            'btnTestDIO
            '
            Me.btnTestDIO.Enabled = False
            Me.btnTestDIO.Location = New System.Drawing.Point(64, 296)
            Me.btnTestDIO.Name = "btnTestDIO"
            Me.btnTestDIO.Size = New System.Drawing.Size(168, 32)
            Me.btnTestDIO.TabIndex = 1
            Me.btnTestDIO.Text = "&Test the PIO Card"
            '
            'lbxRst
            '
            Me.lbxRst.BackColor = System.Drawing.Color.FromArgb(CType(255, Byte), CType(255,
Byte), CType(192, Byte))
            Me.lbxRst.Font = New System.Drawing.Font("Times New Roman", 9.0!,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, CType(0, Byte))
            Me.lbxRst.ItemHeight = 17
            Me.lbxRst.Items.AddRange(New Object() {"Connect Port_0 with Port_3", "
CN1.Pin37 with CN2.Pin47    // PA 0", "      .....", "      CN1.Pin30 with CN2.Pin33    //
PA 7", "", " Connect Port_1 with Port_4", "      CN1.Pin10 with CN2.Pin31    // PB 0",
"      .....", "      CN1.Pin03 with CN2.Pin17    // PB 7", "", " Connect Port_2 with Port_5",
"      CN1.Pin29 with CN2.Pin15    // PC 0", "      .....", "      CN1.Pin22 with CN2.Pin01
// PC 7"})
            Me.lbxRst.Location = New System.Drawing.Point(16, 24)
            Me.lbxRst.Name = "lbxRst"
            Me.lbxRst.ScrollAlwaysVisible = True
```

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

```
            Me.lbxRst.Size = New System.Drawing.Size(384, 259)
            Me.lbxRst.TabIndex = 0
            '
            'groupBox1
            '
            Me.groupBox1.Controls.Add(Me.nudActiveBoards)
            Me.groupBox1.Controls.Add(Me.label2)
            Me.groupBox1.Controls.Add(Me.label1)
            Me.groupBox1.Controls.Add(Me.tbxBoards)
            Me.groupBox1.Font = New System.Drawing.Font("Times New Roman", 10.2!,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, CType(0, Byte))
            Me.groupBox1.Location = New System.Drawing.Point(24, -23)
            Me.groupBox1.Name = "groupBox1"
            Me.groupBox1.Size = New System.Drawing.Size(416, 72)
            Me.groupBox1.TabIndex = 2
            Me.groupBox1.TabStop = False
            Me.groupBox1.Text = "Boards' Information"
            '
            'nudActiveBoards
            '
            Me.nudActiveBoards.BackColor = System.Drawing.Color.FromArgb(CType(128, Byte),
CType(255, Byte), CType(128, Byte))
            Me.nudActiveBoards.Location = New System.Drawing.Point(312, 32)
            Me.nudActiveBoards.Name = "nudActiveBoards"
            Me.nudActiveBoards.Size = New System.Drawing.Size(88, 27)
            Me.nudActiveBoards.TabIndex = 3
            Me.nudActiveBoards.TextAlign = System.Windows.Forms.HorizontalAlignment.Right
            '
            'label2
            '
            Me.label2.Font = New System.Drawing.Font("Times New Roman", 9.0!,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, CType(0, Byte))
            Me.label2.Location = New System.Drawing.Point(200, 32)
            Me.label2.Name = "label2"
            Me.label2.Size = New System.Drawing.Size(112, 24)
            Me.label2.TabIndex = 2
            Me.label2.Text = "Acitved Boards:"
            '
```

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

```
            'label1
            '
            Me.label1.Font = New System.Drawing.Font("Times New Roman", 9.0!,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, CType(0, Byte))
            Me.label1.Location = New System.Drawing.Point(16, 32)
            Me.label1.Name = "label1"
            Me.label1.Size = New System.Drawing.Size(96, 23)
            Me.label1.TabIndex = 1
            Me.label1.Text = "Total Boards:"
            '
            'tbxBoards
            '
            Me.tbxBoards.BackColor = System.Drawing.Color.FromArgb(CType(255, Byte), CType(255,
Byte), CType(192, Byte))
            Me.tbxBoards.Font = New System.Drawing.Font("Times New Roman", 10.2!,
System.Drawing.FontStyle.Bold, System.Drawing.GraphicsUnit.Point, CType(0, Byte))
            Me.tbxBoards.Location = New System.Drawing.Point(112, 32)
            Me.tbxBoards.Name = "tbxBoards"
            Me.tbxBoards.Size = New System.Drawing.Size(64, 27)
            Me.tbxBoards.TabIndex = 0
            Me.tbxBoards.Text = "0"
            Me.tbxBoards.TextAlign = System.Windows.Forms.HorizontalAlignment.Right
            '
            'Form1
            '
            Me.AutoScaleBaseSize = New System.Drawing.Size(6, 18)
            Me.ClientSize = New System.Drawing.Size(463, 400)
            Me.Controls.Add(Me.groupBox2)
            Me.Controls.Add(Me.groupBox1)
            Me.Name = "Form1"
            Me.Text = "Form1"
            Me.groupBox2.ResumeLayout(False)
            Me.groupBox1.ResumeLayout(False)
            CType(Me.nudActiveBoards, System.ComponentModel.ISupportInitialize).EndInit()
            Me.ResumeLayout(False)

        End Sub
```

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

```
#End Region

        Dim cpiodio As PIODIO
        Dim wBaseAddr As Long
        Dim wIrq As Integer
        Dim wSubVendor As Integer, wSubDevice As Integer, wSubAux As Integer
        Dim wSlotBus As Integer, wSlotDevice As Integer
        Dim wInitialCode As Integer
        Dim wTotalBoards As Integer

        Private Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles MyBase.Load
            Dim Rtn
            cpiodio = New PIODIO
            Const PIO_D48 As Integer = &H800130

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

            If PIODIO.PIODIO_SearchCard(wTotalBoards, PIO_D48) <> 0 Then
                Rtn = MsgBox("Search Card Error!!", , "PIODIO Card Error")
                btnTestDIO.Enabled = False
                Exit Sub
            End If

            btnTestDIO.Enabled = True
            tbxBoards.Text = wTotalBoards
            nudActiveBoards.Minimum = 0
            nudActiveBoards.Maximum = wTotalBoards - 1
        End Sub

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

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

```
        Me.Close()
    End Sub

    Private Sub btnTestDIO_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles btnTestDIO.Click
        Dim wRetVal As Integer
        Dim InVal1 As Integer
        Dim InVal2 As Integer
        Dim InVal3 As Integer
        Dim i As Long

        lbxRst.Items.Clear()
        If Val(nudActiveBoards.Value) > Val(tbxBoards.Text - 1) _
        Or Val(nudActiveBoards.Value) < 0 Then
            lbxRst.Items.Add("Invalid board number, Pls retry!!")
        End If
        wRetVal = PIODIO.PIODIO_GetConfigAddressSpace(Val(nudActiveBoards.Value), _
            wBaseAddr, wIrq, wSubVendor, wSubDevice, wSubAux, wSlotBus, wSlotDevice)

        If wRetVal <> 0 Then
            lbxRst.Items.Add("Get Config-Address-Space Error !!")
            Exit Sub
        End If

        btnTestDIO.Enabled = False
        '//***********************************************************
        '// step 1: enable all DI/DO port
        '//***
        lbxRst.Items.Add("Enable All DI/DO ")
        lbxRst.Items.Add(" ")
        cpiodio.PIODIO_OutputByte(wBaseAddr, 1)          '// enable DI/DO

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

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

```
i = 1
While i <= &H80

    cpiodio.PIODIO_OutputByte(wBaseAddr + &HC0, i) '// Port 0
    cpiodio.PIODIO_OutputByte(wBaseAddr + &HC4, i) '// Port 1
    cpiodio.PIODIO_OutputByte(wBaseAddr + &HC8, i) '// Port 2
    lbxRst.Items.Add("Output Port 2, 1, 0 (Hex)=  " _
                + Hex(i) + " " _
                + Hex(i) + " " _
                + Hex(i))

    InVal1 = cpiodio.PIODIO_InputByte(wBaseAddr + &HD0) '// Port 3
    InVal2 = cpiodio.PIODIO_InputByte(wBaseAddr + &HD4) '// Port 4
    InVal3 = cpiodio.PIODIO_InputByte(wBaseAddr + &HD8) '// Port 5
    lbxRst.Items.Add("  Input Port 5, 4, 3 (Hex)=  " _
                      + Hex(InVal3) + " " _
                      + Hex(InVal2) + " " _
                      + Hex(InVal1))

    lbxRst.Items.Add("Delay 100 mSec.")
    lbxRst.Items.Add("  ")
    Thread.Sleep(100)
    i = i * 2
End While


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

i = 1
While i <= &H80

    cpiodio.PIODIO_OutputByte(wBaseAddr + &HD0, i)  '// Port 3
```

LRI Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000

```
                cpiodio.PIODIO_OutputByte(wBaseAddr + &HD4, i) '// Port 4
                cpiodio.PIODIO_OutputByte(wBaseAddr + &HD8, i) '// Port 5
                lbxRst.Items.Add("Output Port 5, 4, 3 (Hex)=  " _
                            + Hex(i) + " " _
                            + Hex(i) + " " _
                            + Hex(i))

                InVal1 = cpiodio.PIODIO_InputByte(wBaseAddr + &HC0) '// Port 0
                InVal2 = cpiodio.PIODIO_InputByte(wBaseAddr + &HC4) '// Port 1
                InVal3 = cpiodio.PIODIO_InputByte(wBaseAddr + &HC8) '// Port 2
                lbxRst.Items.Add("  Input Port 2, 1, 0 (Hex)=  " _
                                 + Hex(InVal3) + " " _
                                 + Hex(InVal2) + " " _
                                 + Hex(InVal1))

                lbxRst.Items.Add("Delay 100 mSec.")
                lbxRst.Items.Add("  ")
                Thread.Sleep(100)
                i = i * 2
            End While


            lbxRst.Items.Add("   The End  ")

            btnTestDIO.Enabled = True
        End Sub
    End Class
End Namespace
```

Distribuída no Brasil pela LRI | vendas@lri.com.br | 51 2129-3000