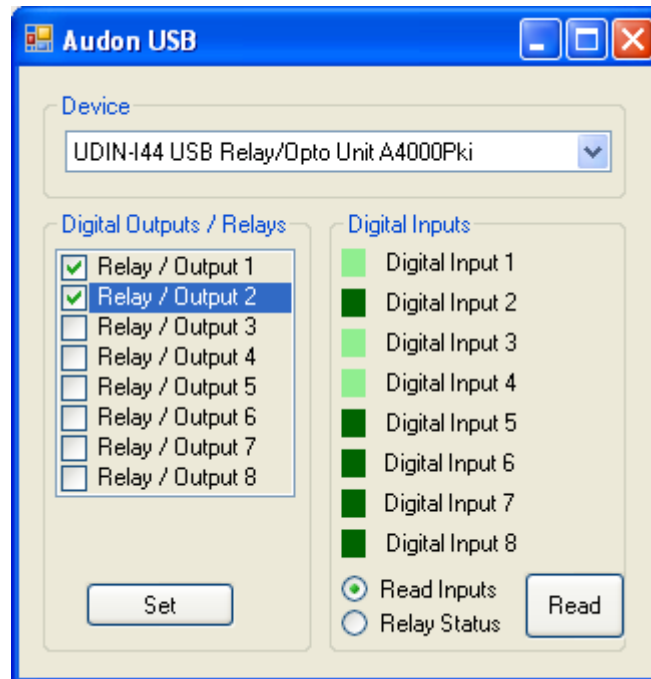


## UDIN - USB Relay / Digital Input, Output Example



To set a Relay / Digital Output, check the appropriate box and click the Set button.

To read a Digital Input, choose Read Inputs and click the Read button.

To read a Relay Status, choose Relay Status and click the Read button.

### **Visual Basic Example – How to program the device**

The following Visual Basic uses the FTDI DLL to directly interface the hardware.

```
Public Class AudonUSB

    Private Sub SetUSB_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles SetUSB.Click
        Dim BytesWritten As Integer
        Dim TempStringData As String
        Dim BytesRead As Integer

        'Open device by serial number
        FT_Status = FT_OpenBySerialNumber(FT_Serial_Number, 1,
FT_Handle)
        If FT_Status <> FT_OK Then
            MsgBox("Failed to open device.", , )
            Exit Sub
        End If

        ' Reset device
        FT_Status = FT_ResetDevice(FT_Handle)
        If FT_Status <> FT_OK Then
            Exit Sub
        End If
    End Sub
End Class
```

```

End If

' Purge buffers
FT_Status = FT_Purge(FT_Handle, FT_PURGE_RX Or FT_PURGE_TX)
If FT_Status <> FT_OK Then
    Exit Sub
End If

' Set Baud Rate
FT_Status = FT_SetBaudRate(FT_Handle, 9600)
If FT_Status <> FT_OK Then
    Exit Sub
End If

' Set parameters
FT_Status = FT_SetDataCharacteristics(FT_Handle,
FT_DATA_BITS_8, FT_STOP_BITS_1, FT_PARITY_NONE)
If FT_Status <> FT_OK Then
    Exit Sub
End If

' Set Flow Control
FT_Status = FT_SetFlowControl(FT_Handle, FT_FLOW_NONE, 0, 0)
If FT_Status <> FT_OK Then
    Exit Sub
End If

' Display in a message box all the items that are checked.
Dim indexChecked As Integer
Dim iRelays As Integer = 0
' First show the index and check state of all selected items.
For Each indexChecked In SetBits.CheckedIndices
    iRelays = iRelays + 2 ^ Val(indexChecked.ToString())
Next

' Write string data to device
Dim sOutput As String
'If iRelays < 16 Then
'sOutput = "r0" + Hex(iRelays) + Chr(13)
'Else
'sOutput = "r" + Hex(iRelays) + Chr(13)
'End If
'sOutput = Me.TextBox1.Text + Chr(13)
sOutput = "r" + Trim(Str(iRelays)) + Chr(13)
FT_Status = FT_Write_String(FT_Handle, sOutput, Len(sOutput),
BytesWritten)
If FT_Status <> FT_OK Then
    Exit Sub
End If

' Wait
Sleep(100)

' Get number of bytes waiting to be read
FT_Status = FT_GetQueueStatus(FT_Handle, FT_RxQ_Bytes)
If FT_Status <> FT_OK Then
    Exit Sub
End If

' Read number of bytes waiting
' Allocate string to receive data

```

```

        TempStringData = Space(FT_RxQ_Bytes + 1)
        FT_Status = FT_Read_String(FT_Handle, TempStringData,
FT_RxQ_Bytes, BytesRead)
        If FT_Status <> FT_OK Then
            Exit Sub
        End If

        ' Close device
        FT_Status = FT_Close(FT_Handle)
        If FT_Status <> FT_OK Then
            Exit Sub
        End If

    End Sub

    Private Sub AudonUSB_Load(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles MyBase.Load
        Dim DeviceCount As Integer
        Dim DeviceIndex As Integer
        Dim TempDevString As String

        ' Get the number of device attached
        FT_Status = FT_GetNumberOfDevices(DeviceCount, vbNullChar,
FT_LIST_NUMBER_ONLY)
        If FT_Status <> FT_OK Then
            Exit Sub
        End If
        ' Exit if no device connected
        If DeviceCount = 0 Then
            MsgBox("No Device Connected", MsgBoxStyle.Critical)
            Exit Sub
        End If
        ' Clear device list
        DeviceList.Items.Clear()
        ' List devices in dropdown
        For DeviceIndex = 0 To DeviceCount - 1
            ' Get serial number of device with index 0
            ' Allocate space for string variable
            TempDevString = Space(16)
            FT_Status = FT_GetDeviceString(DeviceIndex,
TempDevString, FT_LIST_BY_INDEX Or FT_OPEN_BY_SERIAL_NUMBER)
            If FT_Status <> FT_OK Then
                Exit Sub
            End If
            FT_Serial_Number =
Microsoft.VisualBasic.Left(TempDevString, InStr(1, TempDevString,
vbNullChar) - 1)

            ' Get description of device with index 0
            ' Allocate space for string variable
            TempDevString = Space(64)
            FT_Status = FT_GetDeviceString(DeviceIndex,
TempDevString, FT_LIST_BY_INDEX Or FT_OPEN_BY_DESCRIPTION)
            If FT_Status <> FT_OK Then
                Exit Sub
            End If
            FT_Description =
Microsoft.VisualBasic.Left(TempDevString, InStr(1, TempDevString,
vbNullChar) - 1)
            ' Add to dropdown

```

```

        DeviceList.Items.Add(FT_Description + " " +
FT_Serial_Number)
    Next
    ' Set first device
    DeviceList.SelectedIndex = 0
End Sub

Private Sub DeviceList_SelectedIndexChanged(ByVal sender As
System.Object, ByVal e As System.EventArgs) Handles
DeviceList.SelectedIndexChanged
    Dim sDevice As String
    Dim iSpace As Integer

    sDevice =
DeviceList.Items.Item(DeviceList.SelectedIndex).ToString
    iSpace = InStrRev(sDevice, " ")
    If iSpace > 0 Then
        FT_Description = Trim(Microsoft.VisualBasic.Left(sDevice,
iSpace - 1))
        FT_Serial_Number =
Trim(Microsoft.VisualBasic.Mid(sDevice, iSpace + 1))
    End If
End Sub

Private Sub ReadUSB_Click(ByVal sender As System.Object, ByVal e
As System.EventArgs) Handles ReadUSB.Click
    Dim BytesWritten As Integer
    Dim TempStringData As String
    Dim BytesRead As Integer

    'Open device by serial number
    FT_Status = FT_OpenBySerialNumber(FT_Serial_Number, 1,
FT_Handle)
    If FT_Status <> FT_OK Then
        MsgBox("Failed to open device.", , )
        Exit Sub
    End If

    ' Reset device
    FT_Status = FT_ResetDevice(FT_Handle)
    If FT_Status <> FT_OK Then
        Exit Sub
    End If

    ' Purge buffers
    FT_Status = FT_Purge(FT_Handle, FT_PURGE_RX Or FT_PURGE_TX)
    If FT_Status <> FT_OK Then
        Exit Sub
    End If

    ' Set Baud Rate
    FT_Status = FT_SetBaudRate(FT_Handle, 9600)
    If FT_Status <> FT_OK Then
        Exit Sub
    End If

    ' Set parameters
    FT_Status = FT_SetDataCharacteristics(FT_Handle,
FT_DATA_BITS_8, FT_STOP_BITS_1, FT_PARITY_NONE)
    If FT_Status <> FT_OK Then
        Exit Sub
    End If

```

```

End If

' Set Flow Control
FT_Status = FT_SetFlowControl(FT_Handle, FT_FLOW_NONE, 0, 0)
If FT_Status <> FT_OK Then
    Exit Sub
End If

' Write string data to device
Dim sOutput As String
If ReadInputs.Checked = True Then
    ' Read inputs
    sOutput = "i0" + Chr(13)
Else
    ' Relay Status
    sOutput = "s0" + Chr(13)
End If
FT_Status = FT_Write_String(FT_Handle, sOutput, Len(sOutput),
BytesWritten)
If FT_Status <> FT_OK Then
    Exit Sub
End If

' Wait
Sleep(100)

' Get number of bytes waiting to be read
FT_Status = FT_GetQueueStatus(FT_Handle, FT_RxQ_Bytes)
If FT_Status <> FT_OK Then
    Exit Sub
End If

' Read number of bytes waiting
' Allocate string to recieve data
TempStringData = Space(FT_RxQ_Bytes + 1)
FT_Status = FT_Read_String(FT_Handle, TempStringData,
FT_RxQ_Bytes, BytesRead)
If FT_Status <> FT_OK Then
    Exit Sub
End If
' Change LED color depending on input
Dim iInput As Integer
TempStringData = Replace(TempStringData, "i0", " ")
TempStringData = Replace(TempStringData, "s0", " ")
TempStringData = Replace(TempStringData, Chr(13), " ")
TempStringData = Trim(Replace(TempStringData, Chr(10), " "))
iInput = Val(TempStringData)

If (iInput And 1) Then
    LED0.BackColor = Color.LightGreen
Else
    LED0.BackColor = Color.DarkGreen
End If
If (iInput And 2) Then
    LED1.BackColor = Color.LightGreen
Else
    LED1.BackColor = Color.DarkGreen
End If
If (iInput And 4) Then
    LED2.BackColor = Color.LightGreen
Else

```

```
        LED2.BackColor = Color.DarkGreen
    End If
    If (iInput And 8) Then
        LED3.BackColor = Color.LightGreen
    Else
        LED3.BackColor = Color.DarkGreen
    End If
    If (iInput And 16) Then
        LED4.BackColor = Color.LightGreen
    Else
        LED4.BackColor = Color.DarkGreen
    End If
    If (iInput And 32) Then
        LED5.BackColor = Color.LightGreen
    Else
        LED5.BackColor = Color.DarkGreen
    End If
    If (iInput And 64) Then
        LED6.BackColor = Color.LightGreen
    Else
        LED6.BackColor = Color.DarkGreen
    End If
    If (iInput And 128) Then
        LED7.BackColor = Color.LightGreen
    Else
        LED7.BackColor = Color.DarkGreen
    End If

    ' Close device
    FT_Status = FT_Close(FT_Handle)
    If FT_Status <> FT_OK Then
        Exit Sub
    End If
End Sub
End Class
```