Interfacing VB.NET with Embedded System

Table of Contents

OVERVIEW

VB.Net is a simple, modern, object-oriented computer programming language developed by Microsoft to combine the power of .NET Framework and the common language runtime with the productivity benefits that are the hallmark of Visual Basic.

Visual Basic .NET (VB.NET) is an object-oriented computer programming language implemented on the .NET Framework. Although it is an evolution of classic Visual Basic language, it is not backwards-compatible with VB6, and any code written in the old version does not compile under VB.NET.

The following reasons make VB.NET a widely used professional language:

- Modern, general-purpose programming language.
- Object oriented.
- Component oriented.
- Easy to learn.
- Structured language.
- It produces efficient programs.
- It can be compiled on a variety of computer platforms.
- Part of .Net Framework.

GETTING STARTED

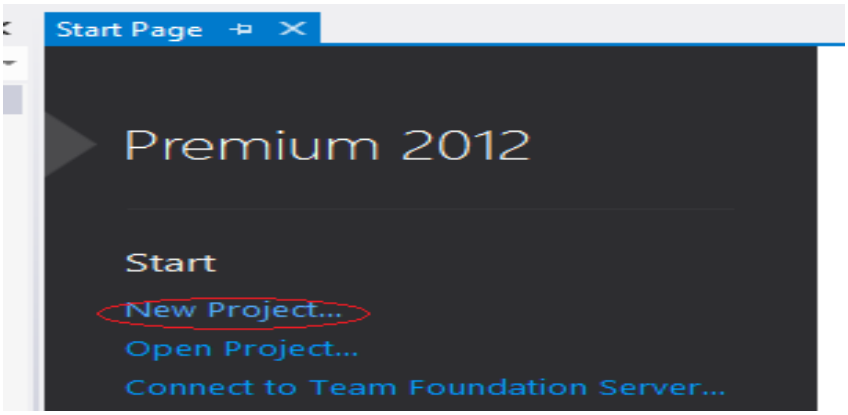
Creating a new VB.NET project:

Once Visual Studio is running the first step is to create a new project. Do this by selecting *New Project* from the *File* menu. This will cause the *New Project* window to appear containing a range of different types of project. For the purposes of this tutorial we will be developing a *Windows Forms Application* so make sure that this option is selected.

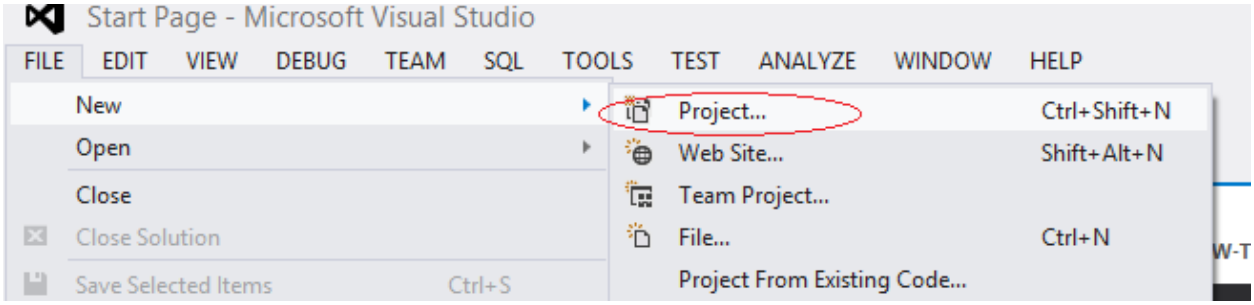
CREATING NEW PROJECT

- The first thing you do when you want to create a new application is to create a NEW PROJECT.

This can be done from start page.



New project created from the NEW PROJECT window:



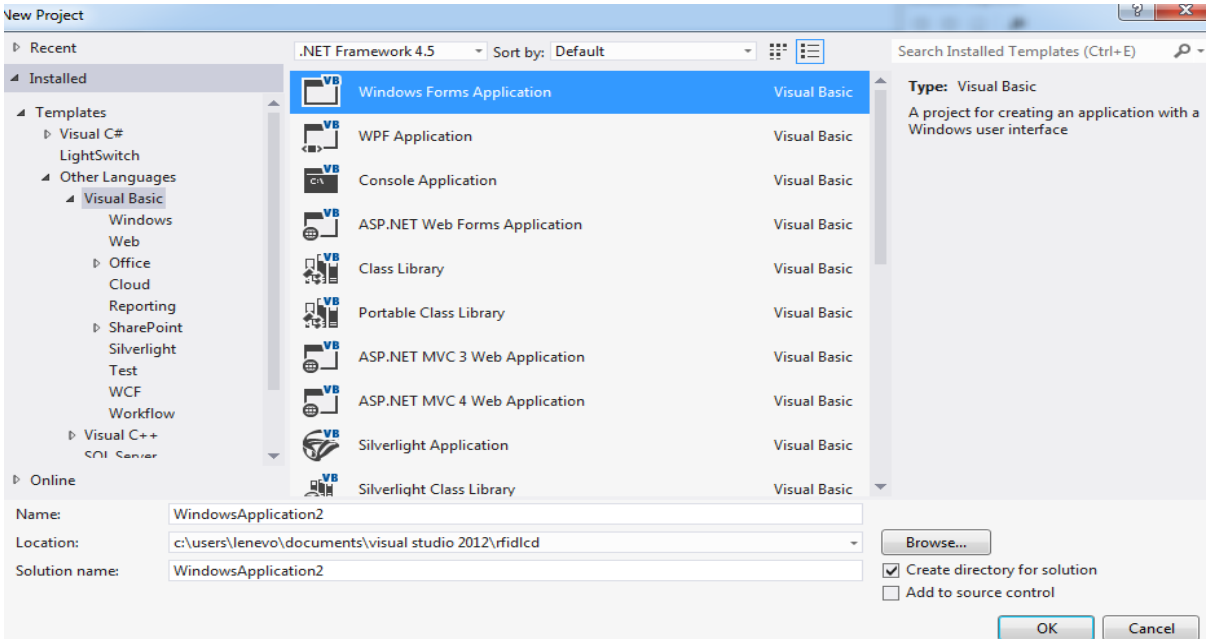
Then the "NEW PROJECT" window will appear.

In this window you will select an appropriate template based on what kind of application you want to create, and a name and location for your project and solution.

The most common applications are:

- Windows form application.
- Console application.
- WPF application
- ASP.NET web application.
- Silverlight application.

Select WINDOWS FORMS APPLICATION.



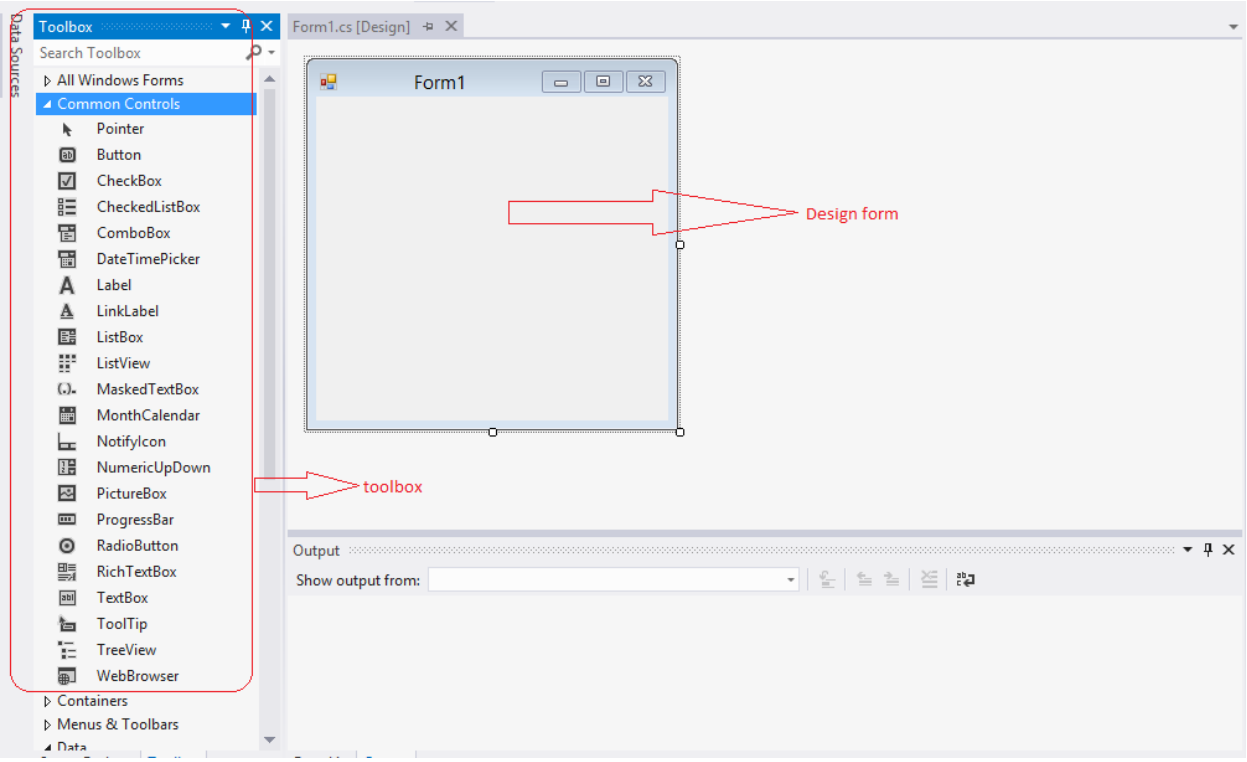
TOOLBOX

When you select WINDOWS FORM APPLICATION, you will get FORM DESIGN WINDOW,it is used to design USER interface by making use of TOOLBOX on the left side of window,

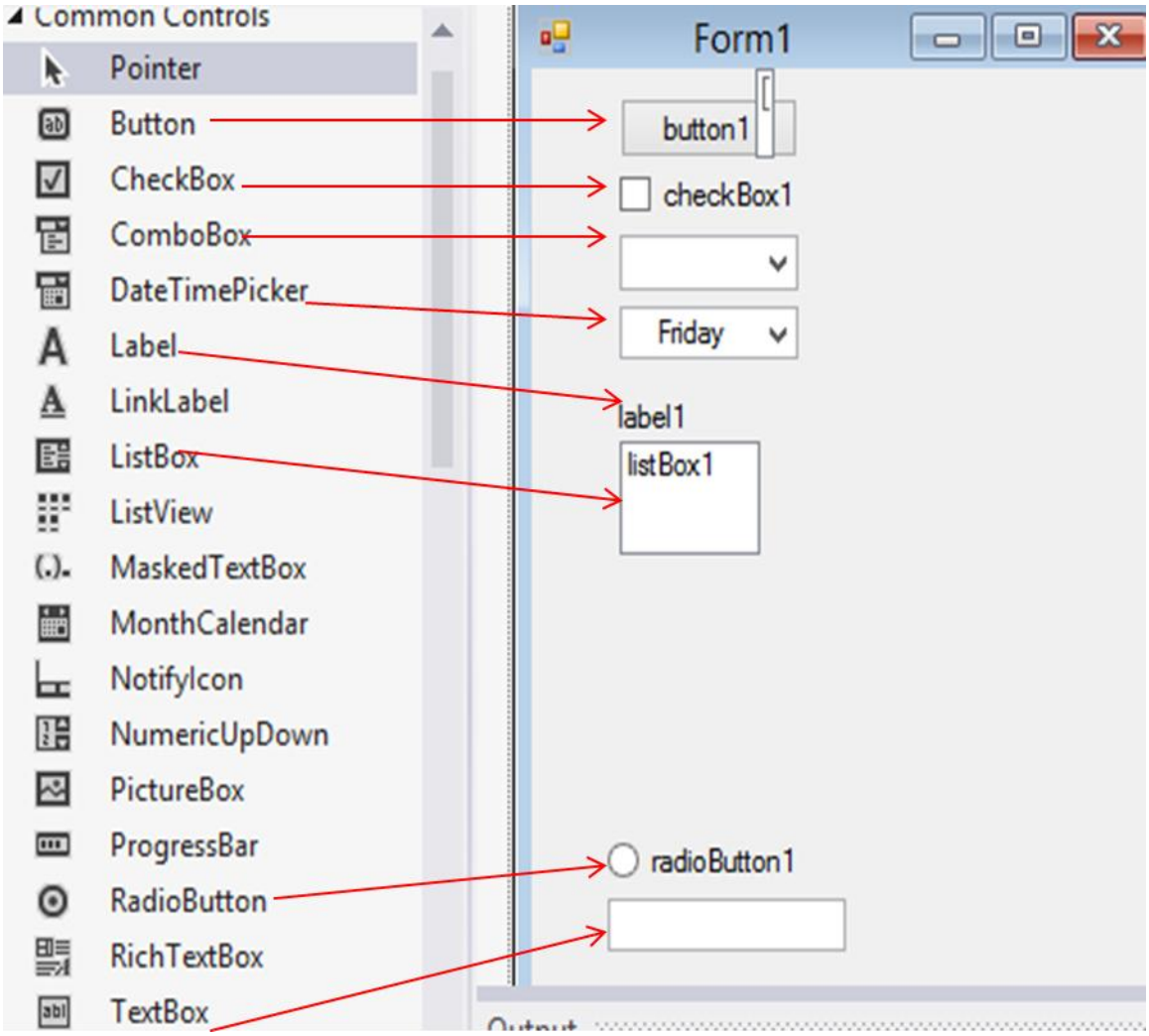
The TOOLBOX contains all the necessary controls, etc. You need to create a user interface by making use of these controls as shown in figure below.

In order to use these controls, just drag and drop it on to your Design forms, as shown in figure.

Figure shows TOOLBOX and DESIGN FORM:

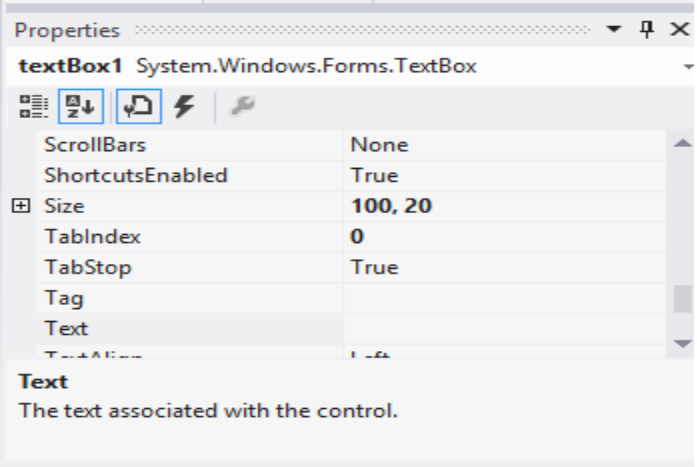


The following screenshot shows, making use of these toolbox controls for designing the user interface on DESIGN FORM.



PROPERTIES WINDOW

Each TOOLBOX we have used on our form has many properties that we can set. This is done by using Properties window. We can find the property window on the right bottom side of your project

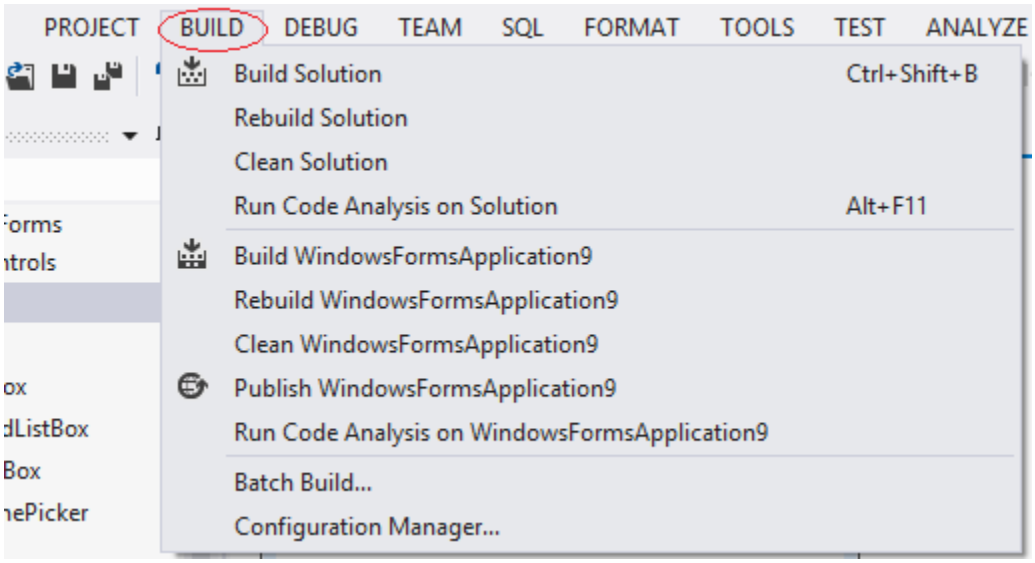


BUILD AND DEBUGGING TOOL

The visual studio has lots of Build and Debugging Tools,

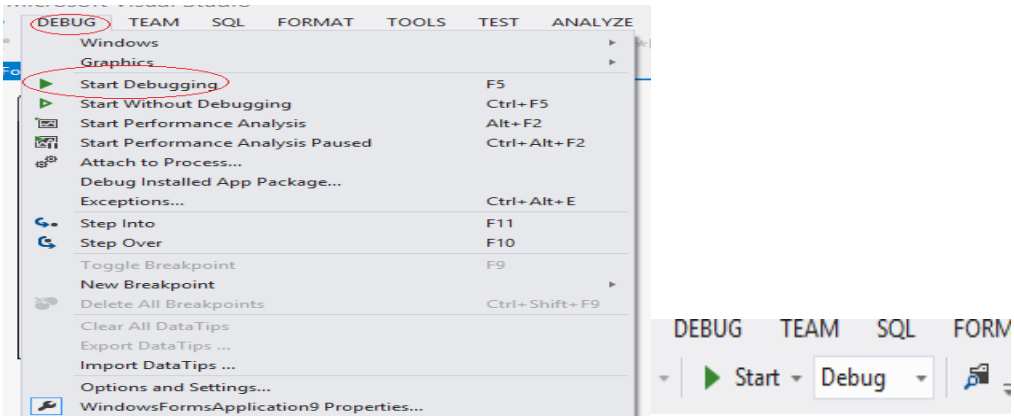
BUILD MENU:

Below we see the Build menu. The most used Build tool is **BUILD SOLUTIONS**.



DEBUG MENU:

In order to RUN or DEBUG your windows form we make use of DEBUG TOOLs. The most used debug tool is START DEBUGGING. it can be find the shortcut for this on the top of your visual studio windows.



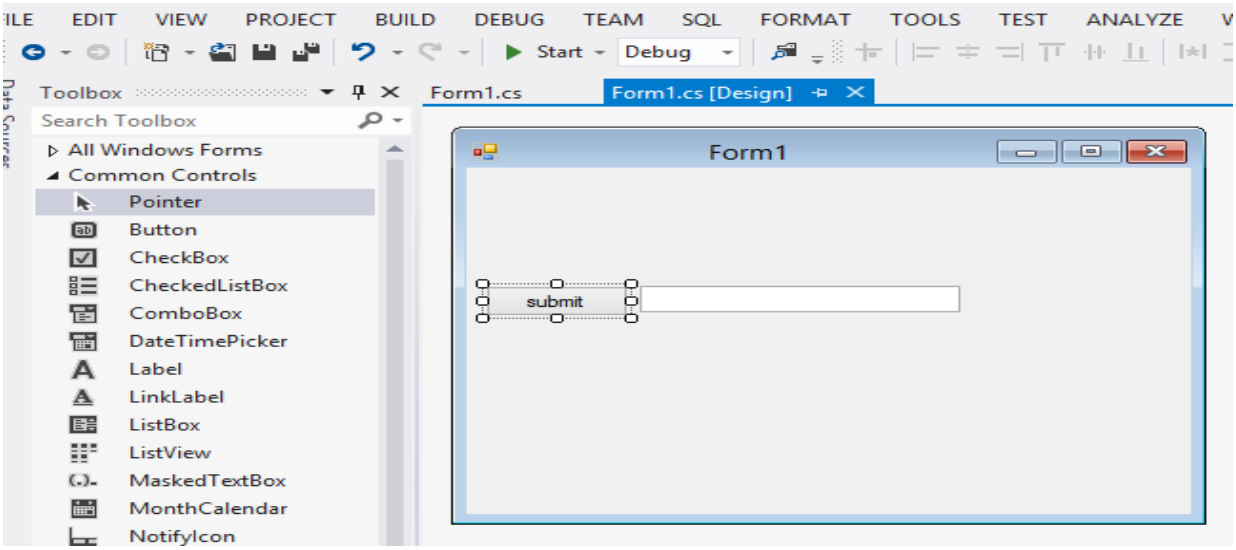
WINDOWS PROGRAMMING

When creating ordinary windows form application, we can select between the following:

- Windows form Application
- WPF application

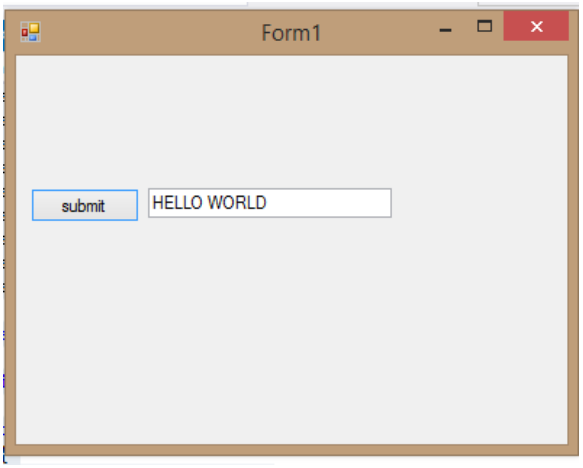
HELLO WORLD

We start by creating traditional “HELLO WORLD” application using Windows Form Application is shown below. The visual studio UI shown below.



In this application we make use of simple textbox and Button(Button name is changed to Submit in the properties) when we click on submit the “HELLO WORLD ”massage will be displayed in the Textbox.

The OUTPUT of this form as shown below:



The code is as follow:

```
Imports System.Text

Imports System
```

```
Public Class Form1

    Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
        TextBox1.Text = "HELLO WORLD"
    End Sub

End Class
```

DATA TYPES AND VERIABLES

"Variables" are simply storage locations for data. You can place data into them and retrieve their contents as part of a VB.NET expression. The interpretation of the data in a variable is controlled through "Types".

The VB.NET simple types consist of:

- Boolean type
- Numeric types: Integrals, Decimal
- String type

BOOLEAN TYPES

Boolean types are declared using the keyword "**Boolean**". They have two values: "true" or "false". In other languages, such as C and C++, boolean conditions can be satisfied where 0 means false and anything else means true. However, in VB.NET the only values that satisfy a boolean condition is true and false, which are official keywords.

Example:

Dim Content as Boolean

Content=True

Numeric types: Integrals, Floating Point, Decimal

Example:

Dim I As Integer
I=35
Dim x As Decimal
X=10.5

Dim d As

Double

d=10000

String type

Example:

Dim abc As String
abc=”Hai..”

Special characters that may be used in strings:

Escape Sequence	Meaning
\'	Single Quote
\"	Double Quote
\\	Backslash
\0	Null, not the same as the C# *null* value
\a	Bell
\b	Backspace
\f	form Feed
\n	Newline
\r	Carriage Return
\t	Horizontal Tab
\v	Vertical Tab

Arrays

Example:

```
Dim A(100) As Integer
```

A(0)=40
A(1)=10
A(2)=20
A(3)=30

CONTROL FLOW

To be able to control the flow in your program is important in every programming language.

The two most important techniques are:

- The **if** Statement
- The **switch** Statement

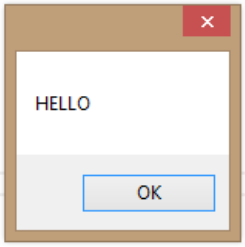
THE IF STATEMENT

The if statement is probably the most used mechanism to control the flow in your application. An if statement allows you to take different paths of logic, depending on a given condition. When the condition evaluates to a Boolean true, a block of code for that true condition will execute. You have the option of a single if statement, multiple else if statements, and an optional else statement.

Example:

```
myTest=false
if myTest=false Then
MsgBox ("Hello")
End If
```

output:



For more complex logic we use the **if … else** statement.

Example:

```
Dim myTest As Boolean
myTest=true

if myTest=false Then
      MsgBox ("Hello1")
else
      MsgBox ("Hello2")
End If
```

Or you can use **nested if… else** if sentences.

Example:

```
Dim myTest As Integer
myTest=2
if myTest =1 Then
      MsgBox ("Hello1")
elseif myTest = 2 Then
      MsgBox ("Hello2")
else
```

```
        MsgBox ("Hello3")
End If
```

THE SELECT CASE STATEMENT

Another form of selection statement is the **select case** statement, which executes a set of logic depending on the value of a given parameter. The types of the values a select statement operates on can be booleans, enums, integral types, and strings.

Example:

```
Dim number As Integer = 8
Select Case number
    Case 1 To 5
        MsgBox("Between 1 and 5, inclusive")
        ' The following is the only Case clause that evaluates to True.
    Case 6, 7, 8
        MsgBox ("Between 6 and 8, inclusive")
    Case 9 To 10
        MsgBox ("Equal to 9 or 10")
    Case Else
        MsgBox ("Not between 1 and 10, inclusive")
End Select
```

LOOPS

In VB.NET we have different kind of loops:

- The while loop
- The do loop
- The for loop
- The foreach loop

The while Loop

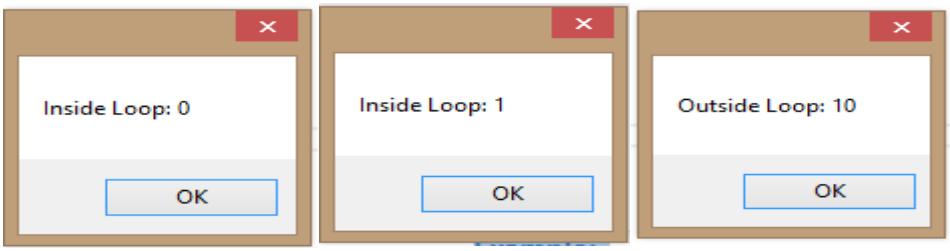
A while loop will check a condition and then continues to execute a block of code as long as

the condition evaluates to a boolean value of true.

Example:

```
Dim myInt As Integer = 0
While myInt < 10

      MsgBox("Inside Loop: " & myInt.ToString())
      myInt += 1
End While
MsgBox("Outside Loop: " & myInt.ToString())
```

OUTPUT:



The do Loop

A do loop is similar to the while loop, except that it checks its condition at the end of the loop. This means that the do loop is guaranteed to execute at least one time. On the other hand, a while loop evaluates its boolean expression at the beginning and there is generally no guarantee that the statements inside the loop will be executed, unless you program the code to explicitly do so.

Example:

```
Dim myInt As Integer = 0
Do
MsgBox("Inside Loop: " & myInt.ToString())
myInt += 1
Loop While myInt < 10
MsgBox("Outside Loop: " & myInt.ToString())
```

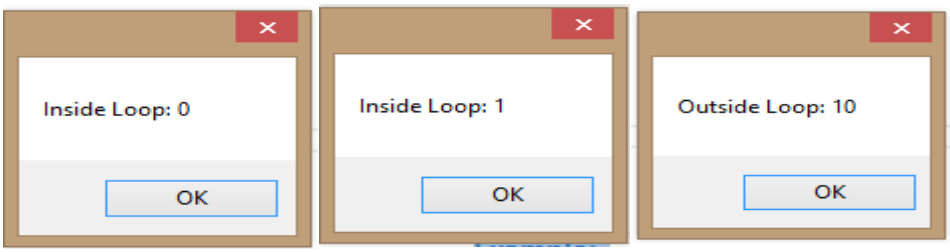
The for Loop

A for loop works like a while loop, except that the syntax of the for loop includes initialization and condition modification. for loops are appropriate when you know exactly how many times you want to perform the statements within the loop.

Example:

```
For i As Integer = 0 To 9
MsgBox("Inside Loop: " & myInt.ToString())
myInt += 1
Next
MsgBox("Outside Loop: " & myInt.ToString())
```

OUTPUT:

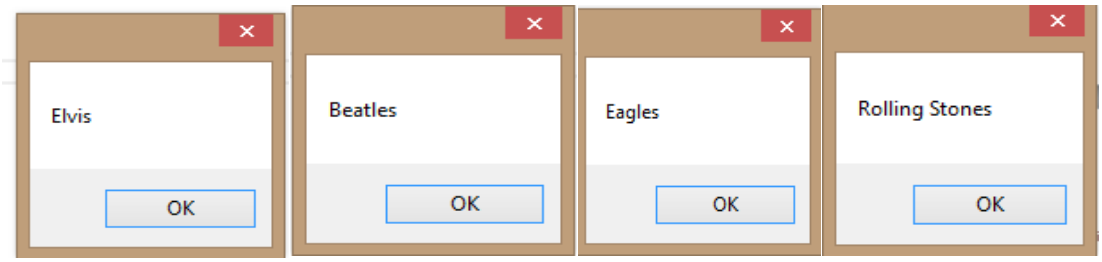


The for each Loop

A foreach loop is used to iterate through the items in a list. It operates on arrays or collections.

Example:

```
Dim names As String() = {"Elvis", "Beatles", "Eagles", "Rolling
Stones"}
For Each person As String In names
     MsgBox(person)
Next
```

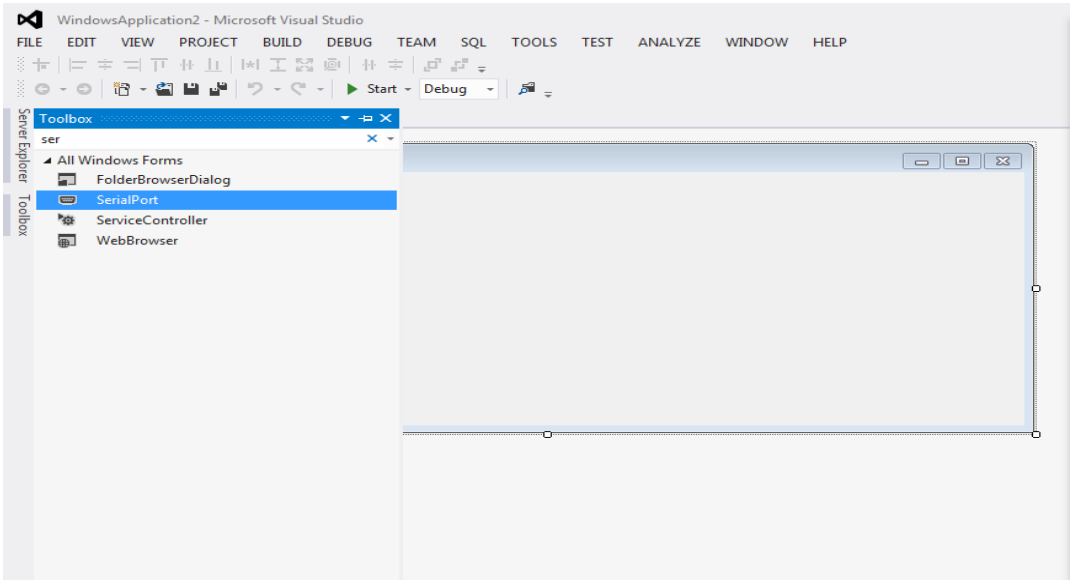


SERIAL COMMUNICATION

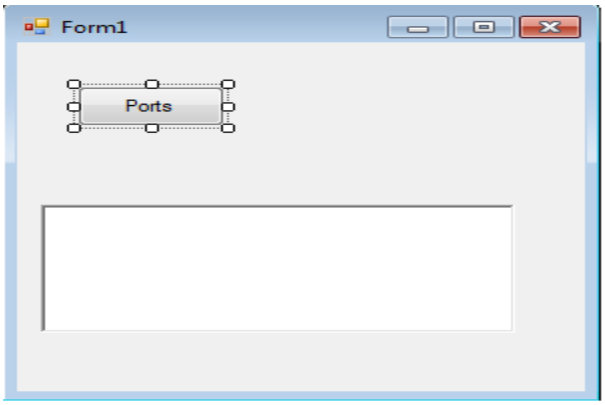
In telecommunication and computer science, serial communication is the process of sending data one bit at a time, sequentially, over a communication channel or computer bus. This is in contrast to parallel communication, where several bits are sent as a whole, on a link with several parallel channels.

Setting Up

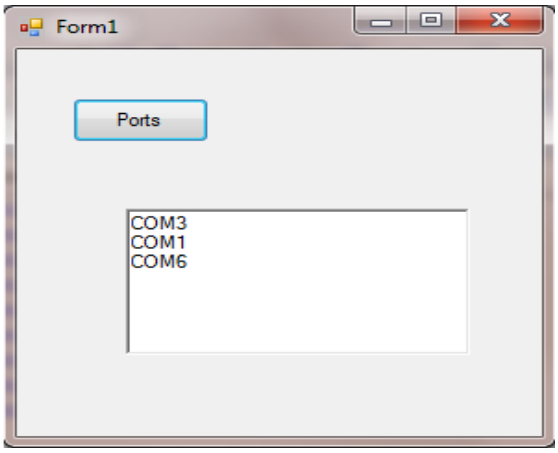
```
Imports System.Data.SqlClient
Imports System.Net.Sockets

Imports System.Text
```



Shown in the above form before communicating with the particular hardware device we should add SerialPort tool from the Toolbox.



This is standard Windows Forms Application via File menu. To this add the button (name Ports) and a Rich Text Box.The button is called btnGetSerialPorts and the Rich Text called as rtbIncomingData (the name will become apparent later).The rich text box is used as it is more flexible than the ordinary text box. Its uses for sorting and aligning text are considerably more than the straight textbox.



This shows all the devices that appear as com ports, a mistake to make is thinking that a device if plugged into the USB will appear as a COM Port.

The baud rate is the amount of possible events that can happen in a second. It is displays usually as a number of bit per second, the possible number that can be used are 300, 600, 1200, 2400, 9600, 14400, 19200, 38400, 57600, and 115200 (these come from the UAR 8250 chip is used, if a 16650 the additional rates of 230400, 460800 and 921600) .

The next box is the number of Data bits, these represent the total number of transitions of the data transmission (or Tx line) 8 is the standard (8 is useful for reading certain embedded application as it gives two nibbles (4 bit sequences).

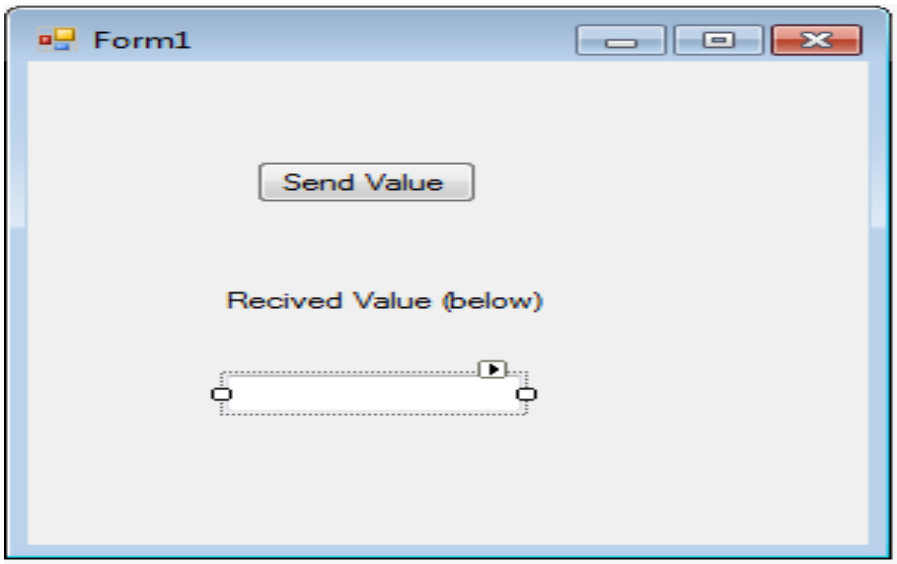
The Handshaking property is used when a full set of connections are used (such as the grey 9 way D-types that litter my desk). It was used originally to ensure both ends lined up with each other and the data was sent and received properly. A common handshake was required between both sender and receiver. Below is the code for the combo box:

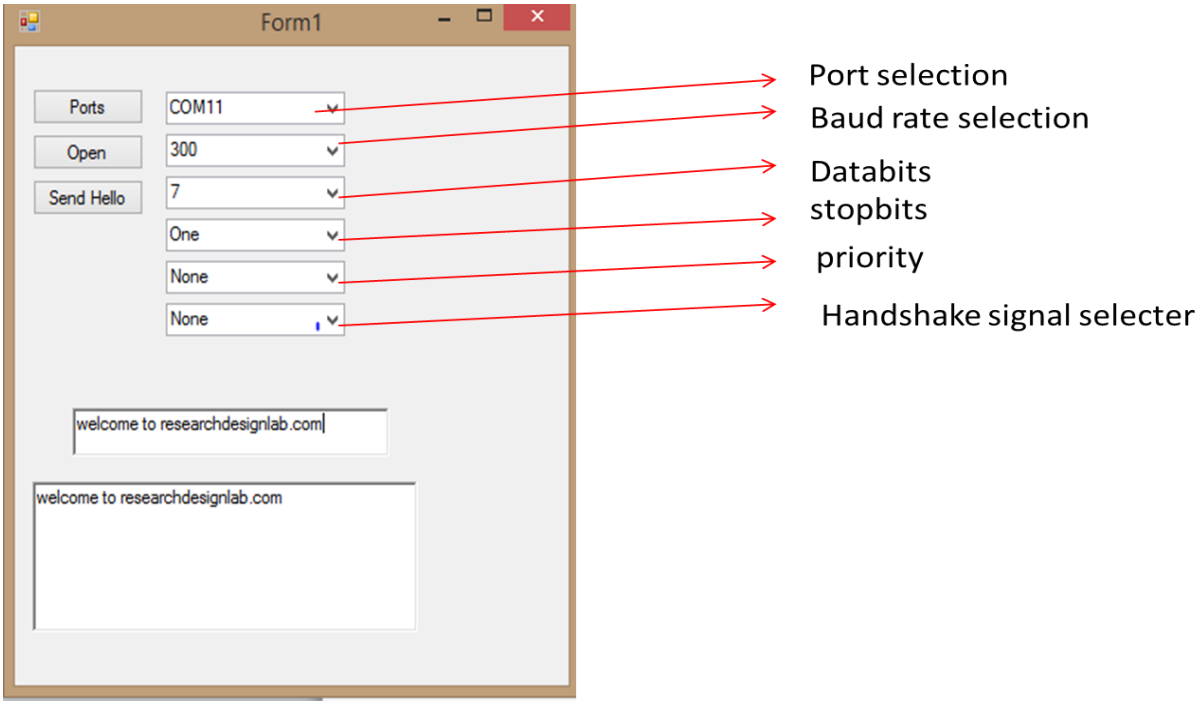
Here is the complete code for serial communication between transmitter and receiver.

```
Imports System.Data.SqlClient
Imports System.Net.Sockets
Imports System.Text
```

```
Public  class Form1
    {

Public Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
MyBase.Load
        Label11.Text = "Hi" + " " + Form2.TextBox1.Text + " , " + "Welcome you to PSA"
        user = Form2.TextBox1.Text


        total = 0
        k = 0
        If SerialPort1.IsOpen Then
            SerialPort1.Close()
        End If
        Try
            With SerialPort1
                .PortName = "COM5" ‘ Initilizing Components…Using code
                .BaudRate = 9600
                .Parity = IO.Ports.Parity.None
                .DataBits = 8
                .StopBits = IO.Ports.StopBits.One
                .Handshake = IO.Ports.Handshake.None
            End With
            SerialPort1.Open()

        Catch ex As Exception
            MsgBox(ex.ToString)
        End Try


    End Sub

Private Sub DataReceived(ByVal sender As Object, ByVal e As
System.IO.Ports.SerialDataReceivedEventArgs) Handles SerialPort1.DataReceived

InBuff = SerialPort1.ReadExisting() ‘ Receiving Value From the Serial Port
TextBox1.Text=InBuff

End Sub


Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
Seialport.WriteLine("Hello World!")
     End Sub

End Class
```

OUTPUT:


Form1
Send Value
Recived Value (below)


Form1
Ports
Open
Send Hello
COM11
300
7
One
None
None
welcome to researchdesignlab.com
welcome to researchdesignlab.com
Port selection
Baud rate selection
Databits
stopbits
priority
Handshake signal selecter

USB RFID INTERFACE WITH VB.Net

Visit http://researchdesignlab.com/rfid-reader-usb.html to buy this product.

Radio-frequency identification (**RFID**) is the wireless use of electromagnetic fields to transfer data, for the purposes of automatically identifying and tracking tags attached to objects. The tags contain electronically stored information. Some tags are powered byelectromagnetic induction from magnetic fields produced near the reader. Some types collect energy from the interrogating radio waves and act as a passive transponder. Other types have a local power source such as a battery and may operate at hundreds of meters from the reader. Unlike a barcode, the tag does not necessarily need to be within line of sight of the reader, and may be embedded in the tracked object. Radio frequency identification (RFID) is one method for Automatic Identification and Data Capture (AIDC).

Reading USB RFID data from serial port

We can use Serial port for Reading the Tag values from the RF-ID Reader. For this we need to connect the RF-ID Reader using the Serial cable to the port and use the relevant COM Port No# to the Serial port.

Normally the System.Net Contains the Serial Port Class and also available in the Toolbox as Serial port component for your Win Forms App

The following code is for reading RFID data from serial port.

```
Imports System.Data.SqlClient
Imports System.Net.Sockets
Imports System.Text
Public Class Form4
    Public k, pname, desc, pid, recom
    Public Shared temp1, InBuff, smt
```

```
    Dim price As Integer
    Dim total As Integer
    Dim clientSocket As New System.Net.Sockets.TcpClient()
    Dim serverStream As NetworkStream
    Dim readData As String
    Dim infiniteCounter As Integer
    Dim clientip, hostname, ip
    Dim rng As Long
    Dim cntrl As Integer
    Dim n1, n2 As String
    Dim len1, len2, len, res As String
    Dim abc
    Public user As String
    Private Declare Sub Sleep Lib "kernel32" (ByVal dwMilliseconds As Integer)
    Private Sub PictureBox1_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs)
        Form2.Show()
        Me.Hide()
    End Sub

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button1.Click

        Me.Hide()
        Recipe.Show()

    End Sub


    Public Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles MyBase.Load
        Label11.Text = "Hi" + " " + Form2.TextBox1.Text + " , " + "Welcome you to PSA"
        user = Form2.TextBox1.Text


        total = 0
        k = 0
        If SerialPort1.IsOpen Then
            SerialPort1.Close()
        End If
        Try
            With SerialPort1
                .PortName = "COM5"
                .BaudRate = 9600
                .Parity = IO.Ports.Parity.None
                .DataBits = 8
                .StopBits = IO.Ports.StopBits.One
                .Handshake = IO.Ports.Handshake.None
            End With
            SerialPort1.Open()

        Catch ex As Exception
            MsgBox(ex.ToString)
        End Try
```

```
    End Sub


    Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Timer1.Tick

        Dim xi As Integer
        xi = Label2.Location.X
        If xi > 3 Then
            xi = xi - 10
        Else
            xi = 613
        End If
        Label2.Location = New Point(xi, 0)

        Dim b1x As Integer
        Dim b2x As Integer
        Dim b3x As Integer
        Dim b4x As Integer
        b1x = Button1.Location.X
        b2x = Button2.Location.X
        b3x = Button3.Location.X
        '      b4x = Button4.Location.X

        If b1x < 320 Then
            b1x = 1062
        Else
            b1x = b1x - 10

        End If
        Button1.Location = New Point(b1x, 608)
        If b2x < 320 Then
            b2x = 1062
        Else
            b2x = b2x - 10

        End If
        Button2.Location = New Point(b2x, 608)
        If b3x < 320 Then
            b3x = 1062
        Else
            b3x = b3x - 10

        End If
        Button3.Location = New Point(b3x, 608)
        If b4x < 320 Then
            b4x = 1062
        Else
            b4x = b4x - 10

        End If
        'Button4.Location = New Point(b4x, 608)
```

```
    End Sub

    Public Sub selectdetail()
        conn.Close()
        conn.Open()
        'lbldesc.Text = ""
        'lblid.Text = ""
        'lblprodname.Text = ""
        pid = ""
        pname = ""
        desc = ""
        price = 0
        temp1 = Mid(temp1, 1, 10)


        Dim cmd As New SqlCommand("select * from pinfo1 where pid='" & temp1 & "' ",
conn)
        Dim dr As SqlDataReader = cmd.ExecuteReader
        If dr.Read = True Then
            pid = dr(0).ToString()
            pname = dr(1).ToString()
            desc = dr(2).ToString()
            recom = dr(4).ToString()

            price = dr(3)
            ' PictureBox3.ImageLocation = dr(4).ToString()



            lblid.BeginInvoke(New myDelegate(AddressOf updateTextBox), New Object() {})
            lblprodname.BeginInvoke(New myDelegate(AddressOf updateTextBox1), New
Object() {})
            lbldesc.BeginInvoke(New myDelegate(AddressOf updateTextBox2), New Object()
{})
            lblprice.BeginInvoke(New myDelegate(AddressOf updateTextBox3), New Object()
{})
            Label16.BeginInvoke((New myDelegate(AddressOf updateTextBox12)), New Object()
{})
            ListBox2.BeginInvoke(New myDelegate(AddressOf updateTextBox5), New Object()
{})
            '  ListBox2.BeginInvoke(New myDelegate(AddressOf updateTextBox6), New
Object() {})
            lbltotal.BeginInvoke(New myDelegate(AddressOf updateTextBox4), New Object()
{})
        End If
        'ListBox1.DisplayMember = pname
        'ListBox1.ValueMember = lblid.Text
        'ListBox2.DisplayMember = price

        conn.Close()
    End Sub


    Private Sub DataReceived(ByVal sender As Object, ByVal e As
System.IO.Ports.SerialDataReceivedEventArgs) Handles SerialPort1.DataReceived
```

```
        ' MessageBox.Show("new data is recived")

        'Timer1.Stop()
        'Form4.TextBox1.Text = filename
        Sleep(300)
        'temp = ""
        price = 0
        ' total =
        pname = ""
        ' Static InBuff As String

        InBuff = SerialPort1.ReadExisting()
        'InBuff = SerialPort1.ReadLine()



        temp1 = Mid(InBuff, 1, 10)

        conn.Close()
        conn.Open()
        If temp1 = "" Then
            MsgBox(".....................")
        Else


            Dim cmd1 As New SqlCommand("select * from cart2 where pid='" & temp1 & "' ",
conn)

            Dim dr As SqlDataReader
            dr = cmd1.ExecuteReader()
            If dr.Read = True Then
                pname = dr(1).ToString
                price = dr.GetValue(2)


                lbltotal.BeginInvoke(New myDelegate(AddressOf updateTextBox10), New
Object() {})
                ListBox2.BeginInvoke(New myDelegate(AddressOf updateTextBox7), New
Object() {})
                ListBox2.BeginInvoke(New myDelegate(AddressOf updateTextBox8), New
Object() {})
                conn.Close()
                conn.Open()
                Sleep(100)
                cmd1 = New SqlCommand("delete from cart2 where pid='" & temp1 & "'",
conn)
                cmd1.ExecuteNonQuery()



            Else
                'lblid.BeginInvoke(New myDelegate(AddressOf updateTextBox), New Object()
{})
                selectdetail()
                cart()
```

```
                'abc = ListBox1.SelectedItem
            End If
        End If
    End Sub

    Public Sub cart()
        conn.Close()
        conn.Open()
        Dim cmd As New SqlCommand("insert into cart2 values('" & pid & "','" & pname &
"','" & price & "')", conn)
        cmd.ExecuteNonQuery()
        'MsgBox(pname & " " & "is added to the cart", MsgBoxStyle.Information, "Result")
        conn.Close()
        updateTextBox9()

    End Sub
    Public Sub updateTextBox9()
        'Dim total As Integer
        Sleep(100)
        total = Val(lbltotal.Text) + price
        ' smt = total


    End Sub

    Public Sub updateTextBox10()
        Sleep(100)


        With lbltotal

            total = Val(lbltotal.Text) - price
            .Text = total

            'End If
            '.ScrollToCaret()
        End With
    End Sub
    Public Sub updateTextBox8()
        Sleep(100)
        'With TextBox2



        With ListBox2
            .Items.Remove(price.ToString())
            '.ValueMember = lblid.Text
            '.ScrollToCaret()
        End With
    End Sub
    Public Sub updateTextBox7()
        Sleep(100)
        With ListBox2
            .Items.Remove(pname)
            '.ValueMember = lblid.Text
```

```
            '.ScrollToCaret()
        End With
    End Sub
    Public Sub updateTextBox5()
        Sleep(100)



        With ListBox2
            .Items.Add(pname)
            '.ValueMember = lblid.Text
            '.ScrollToCaret()
        End With
        With ListBox2
            .Items.Add(price.ToString())

            '.ValueMember = price
            '.ScrollToCaret()
        End With
    End Sub
    Public Sub updateTextBox6()
        Sleep(100)
        With ListBox2
            .Items.Add(price.ToString())

            '.ValueMember = price
            '.ScrollToCaret()
        End With
    End Sub
    Public Delegate Sub myDelegate()
    Public Sub updateTextBox4()
        Sleep(100)
        With lbltotal
            ' .Text = ""
            .Text = Val(lbltotal.Text) + price
            '.Text = total + price
            'id = .Text
            '.ScrollToCaret()
        End With
    End Sub
    Public Sub updateTextBox()
        Sleep(100)
        With lblid
            .Text = ""

            .Text = pid.ToString()



        End With
    End Sub
    Public Sub updateTextBox1()
        Sleep(100)
```

```
        With lblprodname
            .Text = ""

            .Text = pname
            'id = .Text
            'ListBox1.Items.Add(.Text.ToString)
            '.ScrollToCaret()
        End With
    End Sub
    Public Sub updateTextBox2()
        ' Sleep(100)
        With lbldesc
            .Text = ""

            .Text = desc
            'id = .Text
            '.ScrollToCaret()
        End With
    End Sub

    Public Sub updateTextBox12()
        ' Sleep(100)
        With Label16
            .Text = ""

            .Text = recom
            'id = .Text
            '.ScrollToCaret()
        End With
    End Sub

    Public Sub updateTextBox3()
        ' Sleep(100)
        With lblprice
            .Text = ""

            .Text = price.ToString()
            'id = .Text
            '.ScrollToCaret()
        End With
    End Sub
    Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button2.Click
        Me.Hide()
        bookmark.Show()

    End Sub

    Private Sub Button5_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button5.Click
        If ListBox2.Items.Count = 0 Then
            bookmark.ListBox2.Items.Clear()
            Me.Hide()
            Form1.Show()
            Form2.TextBox1.Text = ""
```

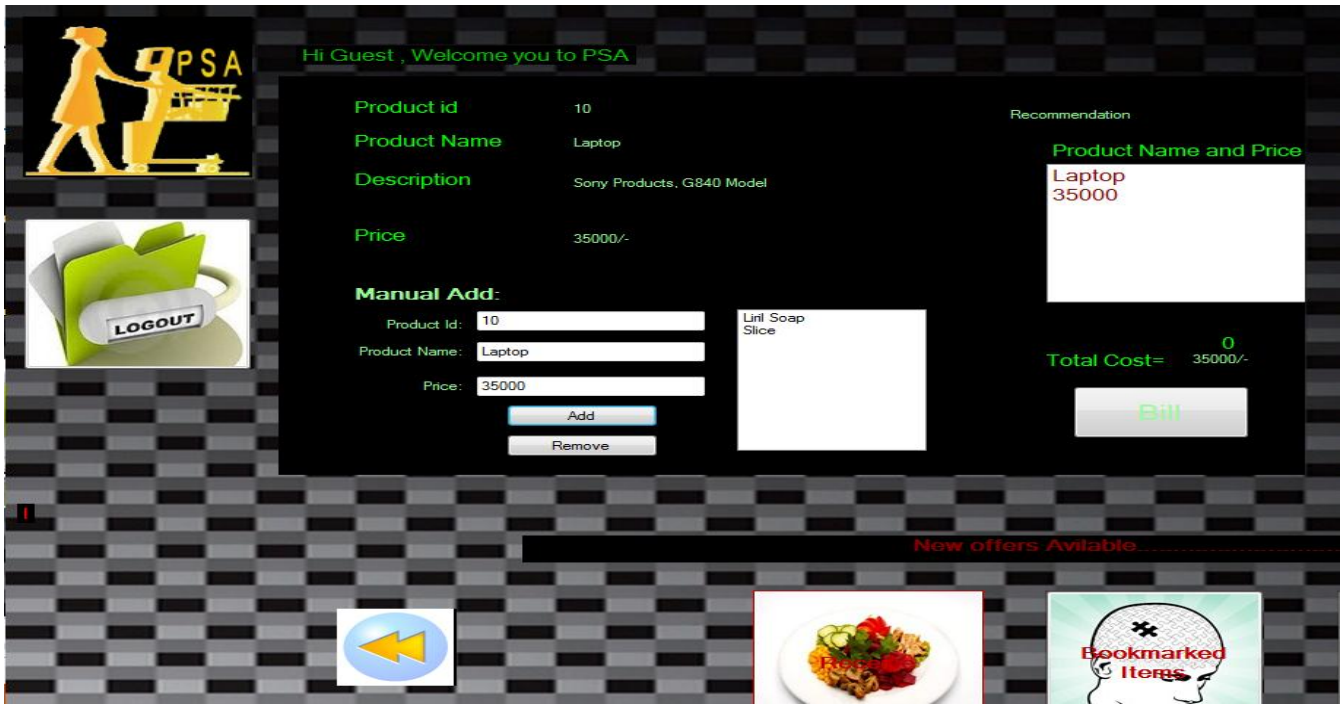
```
            Form2.TextBox2.Text = ""
            Label9.Text = ""
            Label11.Text = ""

        Else
            Label9.Text = "pls remove the items in the cart before logging out"

        End If
    End Sub

    Private Sub Button4_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
        'Me.Hide()
        'offer.Show()

    End Sub



    Private Sub Button3_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button3.Click
        Me.Hide()
        search.Show()

    End Sub



    Private Sub Button6_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button6.Click




        readData = "Conected to Server ..."
        'msg()
        clientSocket.Connect("127.0.0.1", 8888)
        ' Label1.Text = "Client Socket Program - Server Connected ..."
        serverStream = clientSocket.GetStream()


        For i As Integer = 0 To ListBox2.Items.Count - 1
            'ListBox1.SetSelected(i, True)
            ListBox2.SetSelected(i, True)


            'lst1(i)

            lst2(i)
            msg()
```

```
        Next i


        ftotal()

        TextBox1.Text = ""
        TextBox2.Text = ""
        TextBox3.Text = ""


        MsgBox("Bill Has Been Done. Thank You.")
        conn.Close()

        conn.Open()

        Dim cmd3 As New SqlCommand



        cmd3 = New SqlCommand("delete from cart2 ", conn)
        cmd3.ExecuteNonQuery()
        conn.Close()

        Form1.Show()
        Me.Hide()


    End Sub

    Sub lst2(ByVal x)

        ListBox2.SetSelected(x, True)


        Dim outStream As Byte() = _
        System.Text.Encoding.ASCII.GetBytes(ListBox2.SelectedItem + "$")

            serverStream.Write(outStream, 0, outStream.Length)
            serverStream.Flush()
        Dim ctThread As Threading.Thread = New Threading.Thread(AddressOf getMessage)
        ctThread.Start()




            'System.Text.Encoding.ASCII.GetBytes(ListBox2.SelectedItem + "$")
            'serverStream.Write(outStream, 0, outStream.Length)
            'serverStream.Flush()
            'Dim ctThread1 As Threading.Thread = New Threading.Thread(AddressOf
getMessage)
            'ctThread1.Start()


    End Sub
```

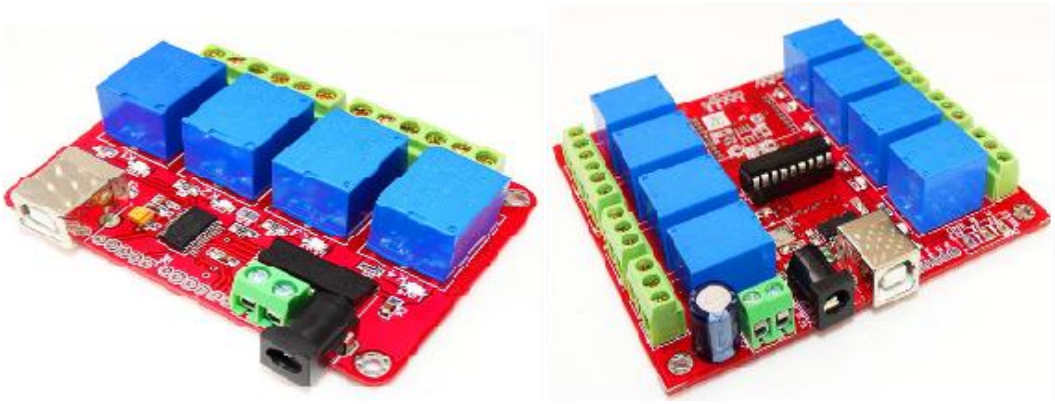
```
    Private Sub msg()
        If Me.InvokeRequired Then
            Me.Invoke(New MethodInvoker(AddressOf msg))
        Else

            TextBox1.Text = TextBox1.Text + Environment.NewLine + " >> " + readData
        End If
    End Sub


    Private Sub getMessage()
        For Me.infiniteCounter = 1 To 3
            infiniteCounter = 1
            serverStream = clientSocket.GetStream()
            Dim buffSize As Integer
            Dim inStream(10024) As Byte
            buffSize = clientSocket.ReceiveBufferSize

            serverStream.Read(inStream, 0, buffSize)
            Dim returndata As String = _
            System.Text.Encoding.ASCII.GetString(inStream)
            'readData = " ." + returndata
            'msg()
        Next

    End Sub
    Sub ftotal()
        ' lbltotal.Text = smt



        Dim outStream As Byte() = _
        System.Text.Encoding.ASCII.GetBytes("Total: " + lbltotal.Text + "$")

        serverStream.Write(outStream, 0, outStream.Length)
        serverStream.Flush()
        Dim ctThread As Threading.Thread = New Threading.Thread(AddressOf getMessage)
        ctThread.Start()
    End Sub



    Private Sub Button7_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button7.Click


        Dim b As New SqlCommand


        conn.Close()
        conn.Open()
```

```
        lbltotal.BeginInvoke(New myDelegate(AddressOf updateTextBox4), New Object() {})
        Dim name1 = TextBox1.Text
        Dim price1 = TextBox2.Text
        b = New SqlCommand("insert into cart2(pid,pname,price) values ('" & TextBox3.Text
& "', '" & TextBox1.Text & "', '" & TextBox2.Text & "')", conn)
        b.ExecuteNonQuery()


        MsgBox(" New Item Added")

        ListBox2.Items.Add(TextBox1.Text)
        '.ValueMember = lblid.Text
        '.ScrollToCaret()

        ListBox2.Items.Add(TextBox2.Text)

    End Sub

    Private Sub TextBox3_TextChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles TextBox3.TextChanged
        conn.Close()
        conn.Open()
        Dim cmd As New SqlCommand("select pname from pinfo1 where pid like '%" &
TextBox3.Text & "%' ", conn)
        Dim rd As SqlDataReader
        rd = cmd.ExecuteReader
        ListBox1.Items.Clear()



        While rd.Read = True
            ListBox1.Items.Add(rd.GetValue(0).ToString)
            'MessageBox.Show("1 item is added")

        End While
        ListBox1.Sorted = True

    End Sub

    Private Sub ListBox1_SelectedIndexChanged_1(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles ListBox1.SelectedIndexChanged
        TextBox1.Text = ListBox1.SelectedItem
        conn.Close()
        conn.Open()
        Dim cmd As New SqlCommand("select * from pinfo1 where pname='" & TextBox1.Text &
"'", conn)
        Dim dr As SqlDataReader = cmd.ExecuteReader
        If dr.Read = True Then
            pid = dr(0).ToString()
            pname = dr(1).ToString()
            desc = dr(2).ToString()

            price = dr(3)
            PictureBox3.ImageLocation = dr(4).ToString()
            TextBox3.Text = pid
```

```
            TextBox1.Text = pname
            TextBox2.Text = price
        End If

    End Sub

    Private Sub Button8_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button8.Click
        lbltotal.BeginInvoke(New myDelegate(AddressOf updateTextBox10), New Object() {})
        ListBox2.BeginInvoke(New myDelegate(AddressOf updateTextBox7), New Object() {})
        ListBox2.BeginInvoke(New myDelegate(AddressOf updateTextBox8), New Object() {})
    End Sub

    Private Sub Button9_Click(sender As Object, e As EventArgs)
        Form1.Show()

    End Sub

    Private Function this() As Object
        Throw New NotImplementedException
    End Function



End Class
```



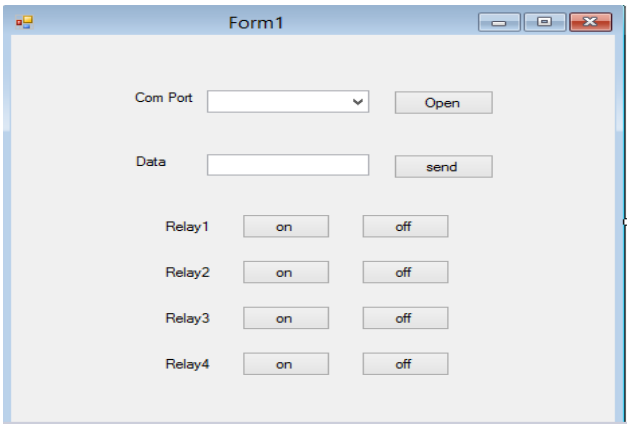
FT245 RELAY CONTROLLER



Visit http://researchdesignlab.com/usb-4-channel-relay-board.html to buy 4-channel relay and visit http://researchdesignlab.com/usb-8-channel-relay-board.htmlto buy 8-channel relay.

A **relay** is an electrically operated switch. Many relays use an electromagnet to mechanically operate a switch, but other operating principles are also used, such as solid-state relays. Relays are used where it is necessary to control a circuit by a low-power signal (with complete electrical isolation between control and controlled circuits), or where several circuits must be controlled by one signal. The first relays were used in long distance telegraph circuits as amplifiers: they repeated the signal coming in from one circuit and re-transmitted it on another circuit. Relays were used extensively in telephone exchanges and early computers to perform logical operations.

Here we are making use of 4 channel relay to controlling it, The following picture show the design part of it, in this we have used one combo box for reading com port and open button to open the selected port, and DATA text box, this is for entering manually which relay should turn on suppose if you enter 'ff' it will turn on relay1.



The complete source code for controlling 4/8 channel relay.

```
Imports System.Data.SqlClient
Imports System.Net.Sockets
Imports System.Text

Public Class Form1

Public Sub Form1_Load(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles
MyBase.Load
        Label11.Text = "Hi" + " " + Form2.TextBox1.Text + " , " + "Welcome you to PSA"
        user = Form2.TextBox1.Text


        total = 0
        k = 0
        If SerialPort1.IsOpen Then
            SerialPort1.Close()
        End If
        Try
            With SerialPort1
                .PortName = "COM5"
                .BaudRate = 9600
                .Parity = IO.Ports.Parity.None
                .DataBits = 8
                .StopBits = IO.Ports.StopBits.One
                .Handshake = IO.Ports.Handshake.None
            End With
            SerialPort1.Open()

        Catch ex As Exception
            MsgBox(ex.ToString)
        End Try


    End Sub
```

```
        Private Sub Button1_Click(sender As Object, e As EventArgs) Handles Button1.Click
                    If SerialPort1.IsOpen Then
            SerialPort1.Close()
         End If
            SerialPort1.Open()


              Button2.Enabled = true
              Button3.Enabled = true
              Button4.Enabled = true
              Button5.Enabled = true
              Button6.Enabled = true
              Button7.Enabled = true
              Button8.Enabled = true
              Button9.Enabled = true
              Button10.Enabled = true
        End Sub


Private Sub Button2_Click(sender As Object, e As EventArgs) Handles Button1.Click

SerialPort1.WriteLine(TextBox1.Text);

End Sub
Private Sub Button3_Click(sender As Object, e As EventArgs) Handles Button1.Click
            SerialPort1.WriteLine ("02");

End Sub

Private Sub Button4_Click(sender As Object, e As EventArgs) Handles Button1.Click
           SerialPort1.WriteLine ("00");
         End Sub


Private Sub Button6_Click(sender As Object, e As EventArgs) Handles Button1.Click
          SerialPort1.WriteLine ("08");
         End Sub


Private Sub Button5_Click(sender As Object, e As EventArgs) Handles Button1.Click
            SerialPort1.WriteLine ("00");
         End Sub


Private Sub Button10_Click(sender As Object, e As EventArgs) Handles Button1.Click
            SerialPort1.WriteLine ("20");
         End Sub


Private Sub Button9_Click(sender As Object, e As EventArgs) Handles Button1.Click
         SerialPort1.WriteLine ("00");
         End Sub
```

```
Private Sub Button8_Click(sender As Object, e As EventArgs) Handles Button1.Click

SerialPort1.WriteLine ("80");
        End Sub


Private Sub Button7_Click(sender As Object, e As EventArgs) Handles Button1.Click
SerialPort1.WriteLine ("00");
        End Sub

    End Class
```

GSM INERFACE

Visit http://researchdesignlab.com/gsm-sim-900.html to buy this product.

There are many different kinds of applications SMS applications in the market today, and many others are being developed. Applications in which SMS messaging can be utilized are virtually unlimited. Some common examples of these are given below:

- Person-to-person text messaging is the most commonly used SMS application, and it is what the SMS technology was originally designed for.
- Many content providers make use of SMS text messages to send information such as news, weather report, and financial data to their subscribers.
- SMS messages can carry binary data, and so SMS can be used as the transport medium of wireless downloads. Objects such as ringtones, wallpapers, pictures, and operator logos can be encoded in SMS messages.

- SMS is a very suitable technology for delivering alerts and notifications of important events.
- SMS messaging can be used as a marketing tool.

In general, there are two ways to send SMS messages from a computer / PC to a mobile phone:

1. Connect a mobile phone or GSM/GPRS modem to a computer / PC. Then use the computer / PC and AT commands to instruct the mobile phone or GSM/GPRS modem to send SMS messages.
2. Connect the computer / PC to the SMS center (SMSC) or SMS gateway of a wireless carrier or SMS service provider. Then send SMS messages using a protocol / interface supported by the SMSC or SMS gateway

AT Commands

AT commands are instructions used to control a modem. AT is the abbreviation of ATtention. Every command line starts with "AT" or "at". That's why modem commands are called AT commands. There are two types of AT commands:

1. Basic commands are AT commands that do not start with a "+". For example, D (Dial), A (Answer), H (Hook control), and O (Return to online data state) are the basic commands.
2. Extended commands are AT commands that start with a "+". All GSM AT commands are extended commands. For example, +CMGS (Send SMS message), +CMGL (List SMS messages), and +CMGR (Read SMS messages) are extended commands.

The FORM DESIGN as show below, Here we using combo box for port selection and textbox for entering mobile number to send sms,and message field to type message and send button.

The complete code as given below, Here we have to create two class 1)sms ,2)program
The class sms will set all pre-requirements in order to send sms,and port values and program class will load the forms and this will initiate the application.

```
Imports System.Data.SqlClient
Imports System.Net.Sockets
Imports System.Text
 Public Class Form1

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

        If SerialPort1.IsOpen Then
            SerialPort1.Close()
        End If

        Try
            With SerialPort1
                .PortName = "COM11"
                .BaudRate = 9600
                .Parity = IO.Ports.Parity.None
                .DataBits = 8
                .StopBits = IO.Ports.StopBits.One
                .Handshake = IO.Ports.Handshake.None
                .NewLine = vbCrLf
                .RtsEnable = True
                .RtsEnable = True
            End With
```

```
            SerialPort1.Open()
            SerialPort1.WriteLine("at+cmgf=1" & vbCrLf)
        Catch ex As Exception
            MsgBox(ex.ToString)
        End Try



    End Sub


Private Sub Button2_Click(sender As Object, e As EventArgs) Handles Button2.Click
Dim Num as Integer
Dim Msg as String
Num=TexBox1.text
Msg=TexBox2.text
Send_sms(Num,Msg)

End Sub

Sub send_sms(ByVal Cont_no As String, ByVal acc_details As String)
        Dim Mobile = Cont_no
        Dim message = acc_details
        SerialPort1.WriteLine("at+cmgd=1" & vbCrLf)
        Sleep(1000)
        Dim a As String
        a = "at+cmgs=""+91" & Mobile & """"
        SerialPort1.WriteLine(a & vbCrLf)
        Sleep(1000)
        SerialPort1.WriteLine(message & Chr(26))
    End Sub

End Class
```