Visual Studio C# connection to Arduino

In this lab you will connect a Windows Form via the COM port to an Arduino platform. When you open the Arduino IDE, you are required to select the COM port which connects the PC to the Arduino Platform. This is the same port that will be used by the C# code to read and write data.

It is import that all these practical labs are completed because the code used will be directly utilised in the final project.

Lab: WinForm Control Red/ Blue LED and Buzzer

In this lab you will use the following code to send data to the Arduino Platform to control the red and blue LEDs and the Buzzer sensor.

```
int redLED = 12; // Red LED
int blueLED = 13; // Blue LED
int buzzerPin = 5; // Buzzer

void setup() {
  Serial.begin(9600);
  pinMode(redLED, OUTPUT);
  pinMode(blueLED, OUTPUT);
  pinMode(buzzerPin, OUTPUT);
}
void loop() {
  // LED and buzzer
  int val = Serial.parseInt();
  switch (val) {
    case 100: digitalWrite(redLED, HIGH); break;
    case 101: digitalWrite(redLED, LOW); break;
    case 200: digitalWrite(blueLED, HIGH); break;
    case 201: digitalWrite(blueLED, LOW); break;
    case 300: {
        tone(buzzerPin, 1000, 50);
        delay(500);
        tone(buzzerPin, 3000, 100); break;
      }
  } // end of switch
}
```

The Arduino sketch uses a Serial.parseInt method to read the string data from C# and convert into an Integer.

Create a new Visual Studio Windows Forms application, then add the following code, ensure your Form has the same naming conventions;

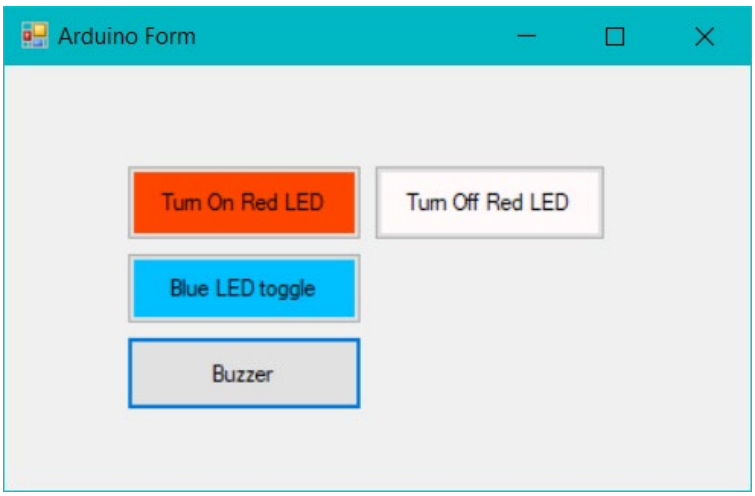
Project Name	ArduinoForm
Form	
(Name)	ArduForm
Text	Arduino Form
Button	
(Name)	BtnRedOn
Text	Turn On Red LED
Button	
(Name)	BtnRedOff
Text	Turn Off Red LED
Button	
(Name)	BtnBlueLEDBtn
Text	Blue LED toggle
Button	

(Name)	BtnBuzzer
Text	Buzzer

C Sharp Code

```
using System;
using System.IO.Ports;
using System.Windows.Forms;

namespace ArduinoForm
{
    public partial class ArduForm : Form
    {
        public ArduForm()
        {
            InitializeComponent();
            Init(); // connect to COM serial port
        }

        private SerialPort myPort;
        bool blueLEDStatus = true;

        // initialise the com port for serial data
        private void Init()
        {
            try
            {
                myPort = new SerialPort();
                myPort.BaudRate = 9600;
                myPort.PortName = "COM5"; // modify to your local COM
                myPort.Open();
            }
            catch (Exception)
            {
                MessageBox.Show("Not connected");
            }
        }

        // code to control red LED
        // send text value 100 - arduino will parse to integer
        private void BtnRedOn_Click(object sender, EventArgs e)
        {
            myPort.WriteLine("100");
        }
        // send text value 101 - arduino will parse to integer
        private void BtnRedOff_Click(object sender, EventArgs e)
        {
            myPort.WriteLine("101");
        }
        // code to control blue LED - uses bool status
        private void BtnBlueLED_Click(object sender, EventArgs e)
        {
            if (blueLEDStatus)
            {
                myPort.WriteLine("200");
                blueLEDStatus = false;
            }
            else
            {
                myPort.WriteLine("201");
                blueLEDStatus = true;
            }
        }
        // send text value 300 - activate buzzer
        private void BtnBuzzer_Click(object sender, EventArgs e)
        {
            myPort.WriteLine("300");
        }
    }
}
```

Form Layout



Notes

There are several points to understand, this code is event driven and requires the user to click the button before any data is sent. The code sends a text values of 100, 101, 200, 201 or 300 to the Arduino which is read and converted into and integer for the Arduino Switch/Case statement.

The COM port listed in the initialization method will need to be changed to suit your Arduino IDE connection. In the previous code the COM port was COM5.

Once your code is error free you can start the program and click the buttons which will turn on the associated LED or sound the buzzer.