

Source: C# Corner (www.c-sharpcorner.com)

PRINT

Article



Turning LED On And Off Using Buttons In Microsoft Visual Studio

By **Kishore Chowdary** on **Oct 02 2016**

Requirements

1. Visual Studio.
2. Arduino IDE.
3. LED Light (1 number).
4. Aurduino Board.

Step 1- Programming Arduino

- Open Arduino IDE, if already installed or download and install it from this [link](#).
- Now, open the IDE and enter the code, given below in it.

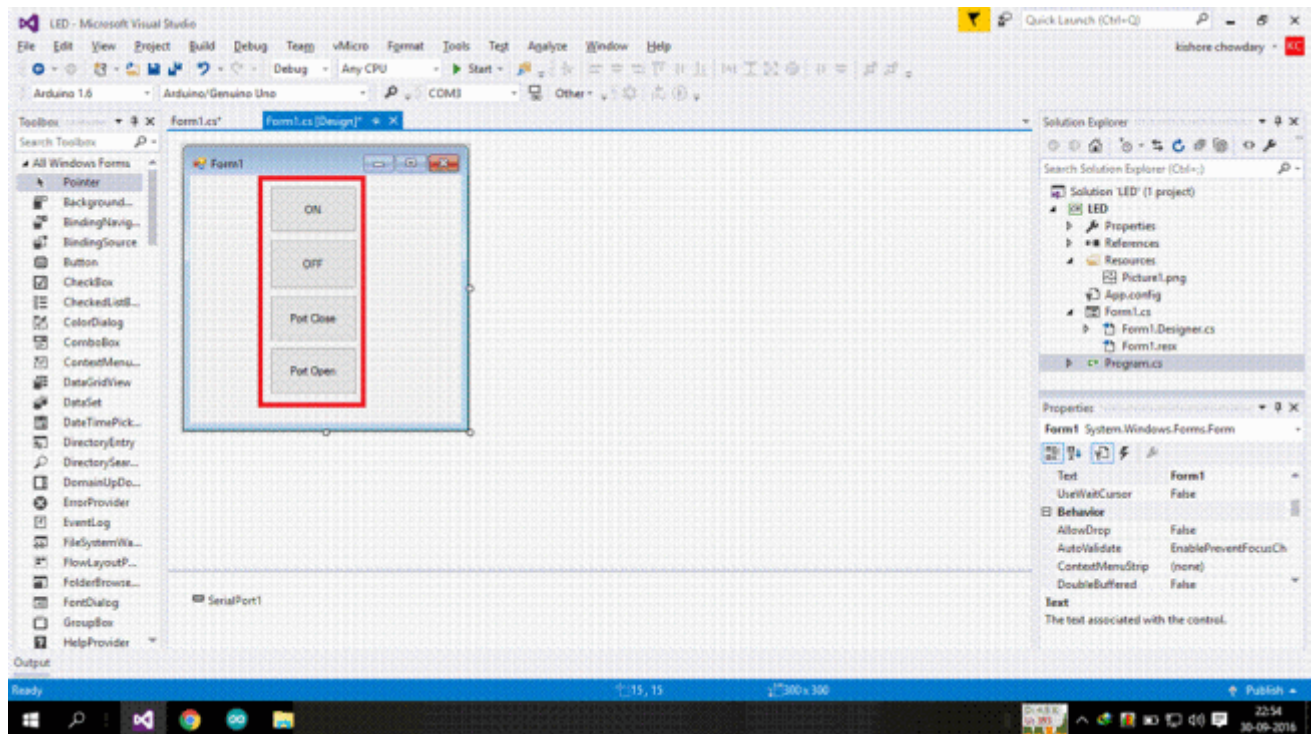
```
1. char incomingOption;  
2. void setup()  
3. {  
4. pinMode(13, OUTPUT);  
5. Serial.begin(9600);  
6. }  
7. void loop()  
8. {  
9. incomingOption = Serial.read();  
10. if (incomingOption=='1')  
11. {  
12. digitalWrite(13, HIGH);  
13. }  
14. else if (incomingOption=='0')  
15. {  
16. digitalWrite(13, LOW);  
17. }  
18. }
```

- Save this sketch in any location and then verify the code.

Step 2- Designing in Visual Studio

- Open Visual Studio and create new Windows Form Application with any name, you want.
- Once you have created a new form Application, open the designer Window of Form1.
- Now, from the tools box, drag and drop the SerialPort tool into Form 1.
- This will help you to communicate your program with the Arduino board.
- Now, add four buttons to Form 1.
- Name the buttons as ON, OFF, Port Open and Port Close.

- It will appear as the screenshot, given below-



Step 3 - Writing C# code in Visual Studio

You have four buttons. Hence, you need to write a separate code for each button.

- Double click on "ON button". You will get a coding page. Paste the code, given below, there-

```

1. private void button1_Click(object sender, EventArgs e)
2. {
3.     try
4.     {
5.         SerialPort1.Write("1"); //send 1 to Arduino
6.     }
7.     catch (Exception ex)
8.     {
9.         MessageBox.Show(ex.Message);
10.    }
11. }

```

- Double click on "OFF button" and paste the code, given below, in it-

```

1. private void button2_Click(object sender, EventArgs e)
2. {
3.     try
4.     {
5.         SerialPort1.Write("0"); //send 0 to Arduino
6.     }
7.     catch (Exception ex)
8.     {
9.         MessageBox.Show(ex.Message);
10.    }
11. }

```

- Now, double click Form1.

- You will get a coding page for it.
- Paste the code, given below, in it-

```

1. private void Form1_Load(object sender, EventArgs e)
2. {
3.     SerialPort1.Open(); //open serialPort
4. }

```

- Now, double click on the Port Close button and paste the code, given below, in it.

```

1. private void button3_Click(object sender, EventArgs e)
2. {
3.     SerialPort1.Close(); //close serialPort
4. }

```

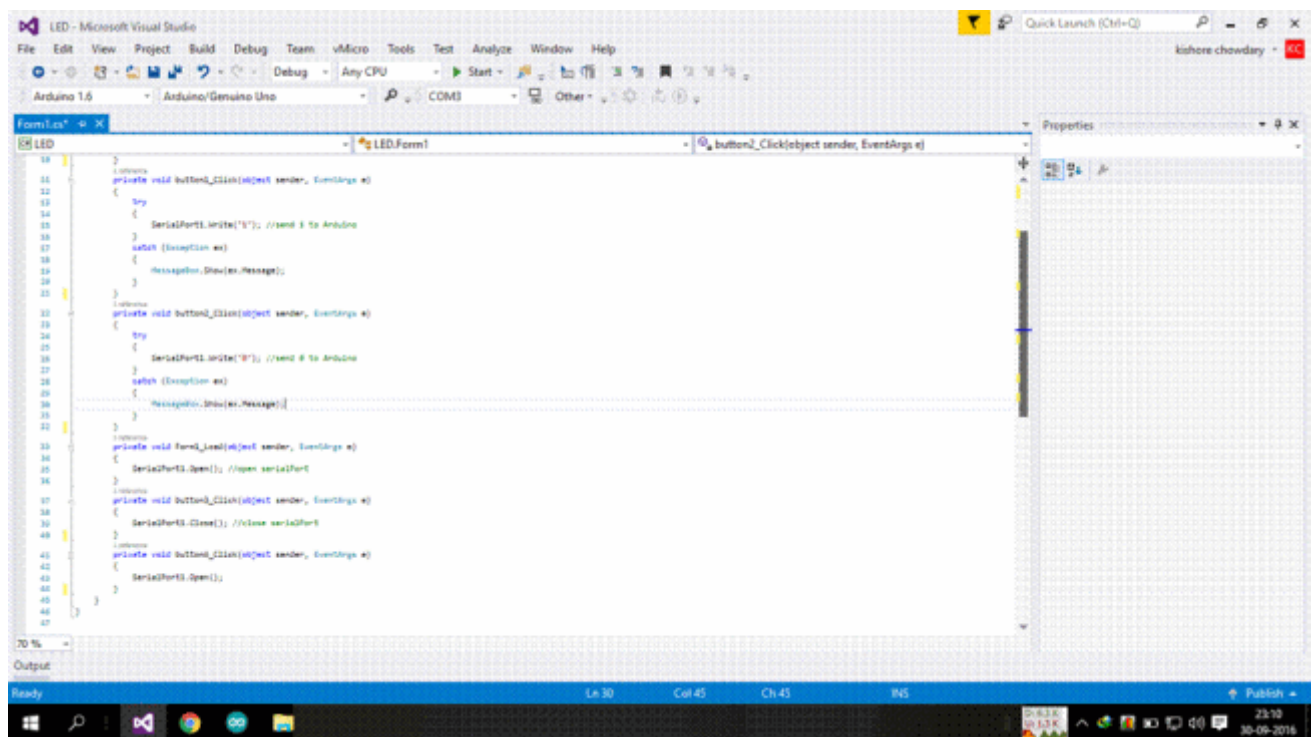
- Now, double click on the Port Open button and paste the code, given below, in it-

```

1. private void button4_Click(object sender, EventArgs e)
2. {
3.     SerialPort1.Open();
4. }

```

- Once you finish all the code, the total view will be like the screen shot, shown below-



- The complete set of code is given below. It will show some errors, if you copy and paste the code without double clicking the buttons and the Form. It is good to follow the steps correctly.

```

1. using System;
2. using System.Windows.Forms;
3. namespace LED
4. {
5.     public partial class Form1 : Form
6.     {
7.         public Form1()
8.         {

```

```
9.      InitializeComponent();
10.    }
11.    private void button1_Click(object sender, EventArgs e)
12.    {
13.        try
14.        {
15.            SerialPort1.Write("1"); //send 1 to Arduino
16.        }
17.        catch (Exception ex)
18.        {
19.            MessageBox.Show(ex.Message);
20.        }
21.    }
22.    private void button2_Click(object sender, EventArgs e)
23.    {
24.        try
25.        {
26.            SerialPort1.Write("0"); //send 0 to Arduino
27.        }
28.        catch (Exception ex)
29.        {
30.            MessageBox.Show(ex.Message);
31.        }
32.    }
33.    private void Form1_Load(object sender, EventArgs e)
34.    {
35.        SerialPort1.Open(); //open serialPort
36.    }
37.    private void button3_Click(object sender, EventArgs e)
38.    {
39.        SerialPort1.Close(); //close serialPort
40.    }
41.    private void button4_Click(object sender, EventArgs e)
42.    {
43.        SerialPort1.Open();
44.    }
45. }
46. }
```

Step 4 - Connecting the LED to the Arduino and uploading program

- Take the positive side of the LED, connect it to the pin 13 and negative side to the GND.
- Now, simply verify whether the COM ports are same in the Arduino IDE and the properties of the Form in Visual Studio.
- After you verify it, upload the sketch into Arduino board. The code, which you wrote in IDE in the beginning is called as sketch.
- Now, you are completely ready with the whole setup.

Final step - Working of the project

- Deploy the whole setup.
- Now, click on "ON button". The LED will glow.
- Now, click on the "OFF button". The LED will turn off.
- Now; click on the Port Close button and try to turn on the LED. It will show a message, which port is closed.
- Now, click on the Port Open button and try to turn on the LED. It will now glow.
- Thus you have made your own LED lighting program using the C# language in visual studio.

Note: For more projects, based on the IOT, go through the links, given below-

1. [Lighting LED through voice](#)
2. [Home Automation Using IOT](#)

Thank you for using C# Corner