

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

PRINT

Article



Turning LED OFF And ON Through Voice Recognition using Microsoft Visual Studio

By **Kishore Chowdary** on **Sep 28 2016**

Requirements

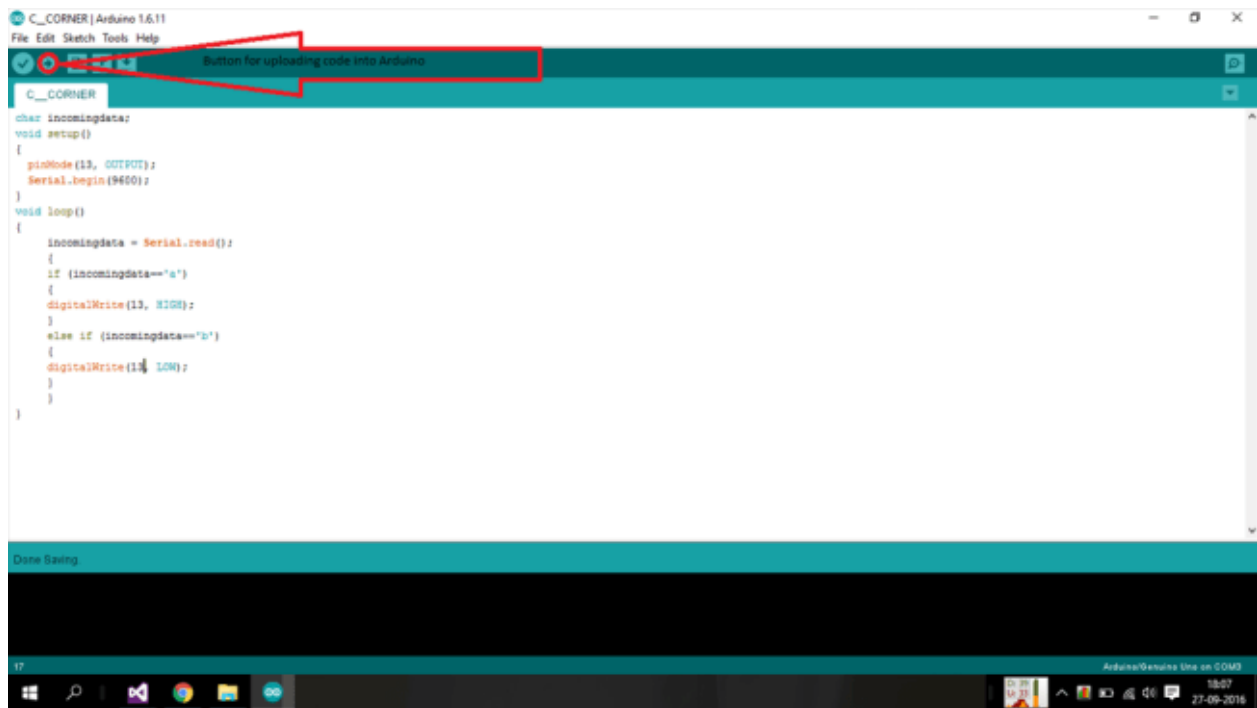
1. Arduino UNO R3 or any other versions.
2. Arduino IDE.
3. Visual Studio of any version.
4. LED light.

Step 1- Programming the Arduino

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

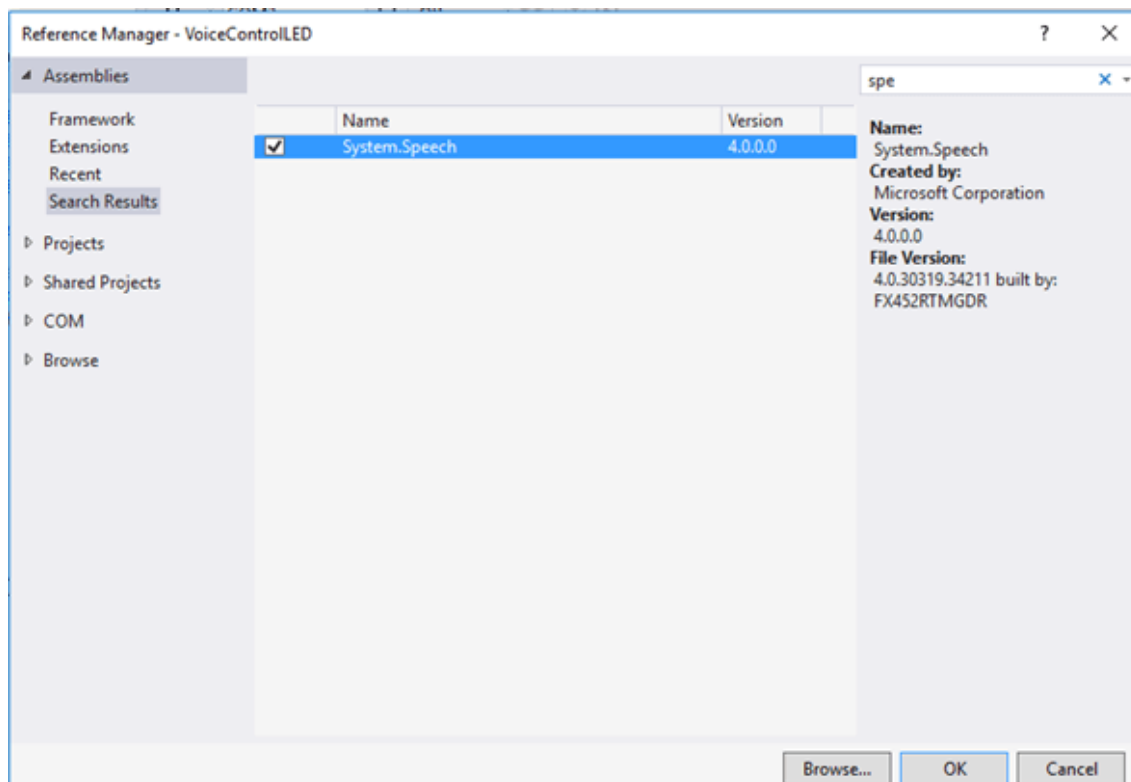
```
1. char incomingdata;  
2. void setup() {  
3.   pinMode(13, OUTPUT);  
4.   Serial.begin(9600);  
5. }  
6. void loop() {  
7.   incomingdata = Serial.read(); {  
8.     if (incomingdata == 'a') {  
9.       digitalWrite(13, HIGH);  
10.    } else if (incomingdata == 'b') {  
11.      digitalWrite(13,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

- Save this sketch in any location and then verify the code.
- Now connect your Arduino board and upload this sketch into the board.



Step 2- Programming in Visual Studio

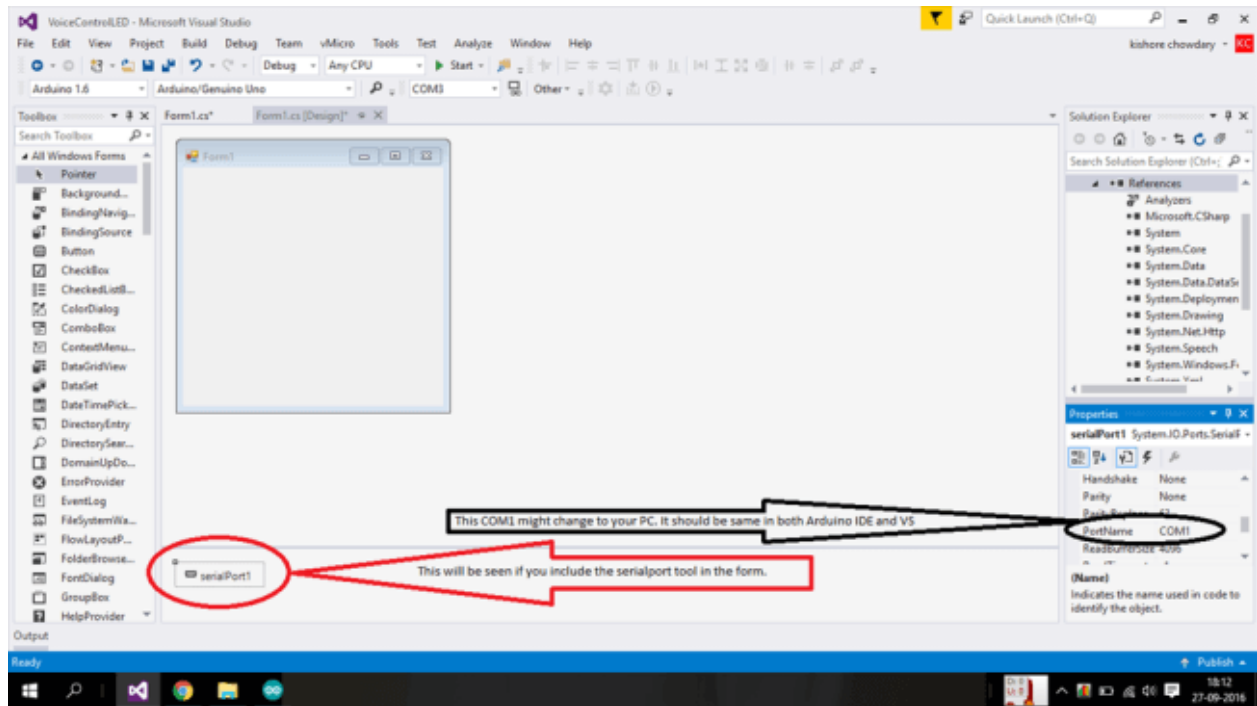
- Open Visual Studio and create new Windows Form application with any name you want.
- Once you have created a new form application, open the designer window of Form1.
- Now, from the tools box, drag and drop the SerialPort tool into the form 1.
- This will help you communicate your program with the Arduino board.
- Now, you have to make your PC recognize your voice so that it can act as a command for the system,
- For this, we need to include a reference file called System.Speech
- This will now help you to give speech as an input to the computer.



Step 3 - Writing code for speech recognition

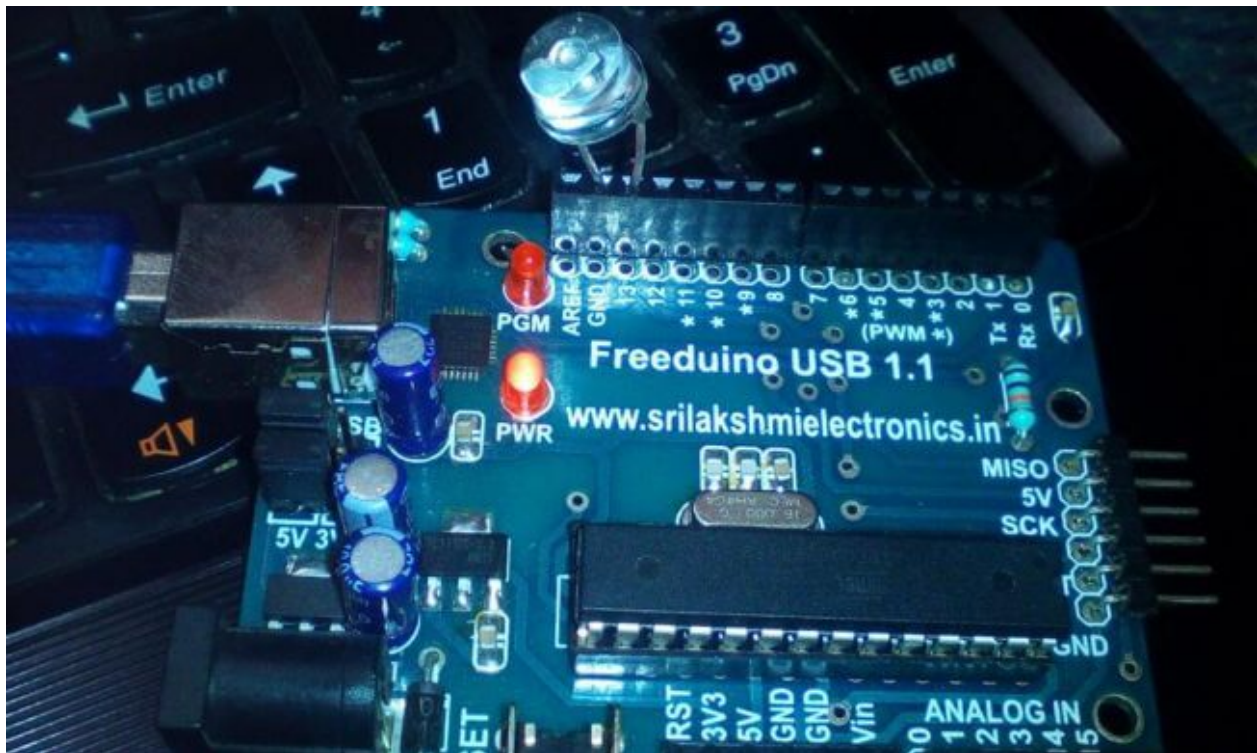
- We need to write an instance of code which will help the computer to recognize our voice input.
- So, first we need to set a predefined command like LED ON for turning LED on and LED OFF for turning the LED off.
- Double click on the Form1 and paste the following code in there.
 1. using System;
 2. using System.Speech.Recognition;
 3. using System.Speech.Synthesis;
 4. using System.Windows.Forms;
 5. namespace VoiceControlLED {
 6. public partial class Form1: Form {
 7. public Form1() {
 8. InitializeComponent();
 9. }
 10. SpeechSynthesizer speechSynthesizerObj;
 11. SpeechRecognitionEngine recognizer = new SpeechRecognitionEngine();
 12. private void Form1_Load(object sender, EventArgs e) {
 13. SpeechRecognitionEngine sRecognize = new SpeechRecognitionEngine();
 14. Choices sList = new Choices();
 15. sList.Add(new string[] {
 16. "light on",
 17. "light off"
 18. });
 19. Grammar gr = new Grammar(new GrammarBuilder(sList));
 20. try {
 21. sRecognize.RequestRecognizerUpdate();
 22. sRecognize.LoadGrammar(gr);
 23. sRecognize.SpeechRecognized += sRecognize_SpeechRecognized;
 24. sRecognize.SetInputToDefaultAudioDevice();
 25. sRecognize.RecognizeAsync(RecognizeMode.Multiple);
 26. sRecognize.Recognize();
 27. } catch {
 28. return;
 29. }
 30. }
 31. private void sRecognize_SpeechRecognized(object sender, SpeechRecognizedEventArgs e) {
 32. {
 33. if (e.Result.Text == "light on") {
 34. serialPort1.Open();
 35. serialPort1.Write("a"); //send a to Arduino
 36. serialPort1.Close();
 37. } else if (e.Result.Text == "light off") {
 38. serialPort1.Open();
 39. serialPort1.Write("b"); //send b to Arduino
 40. serialPort1.Close();
 41. }
 42. }
 43. }
 44. }
 45. }

- The above code will quote an error in the serialPort1 if you don't include the serialport tool into the form1.
- Have a look over step 2 if you face this issue.



Step 4 - Connecting the LED in the Arduino board

- You must be cautious while you give connections to the board.
- A wrong connection might end up in wrecking your board.
- Here, in the program for Arduino, we have used Pin 13.
- So, let us fix the LED in the 13th pin.
- Make sure to fix the Positive side of the LED in the 13th pin and the negative side of the LED in the GND pin.
- Have a look over the image.



Final Step - Executing and working with the LED

- Before you deploy and run your code, make sure that you made proper connection and coding.

- Now, simply run your application.
- When you say the command "Light On", the LED will glow.
- When you say the command "Light Off", the LED will turn off.
- That's it. You have made a lighting system which works with the voice commands.

Note

Make sure that the COM port is same in the properties of the Serial port tool which you have included in the form and in the Arduino IDE. To check the port, go to Tools->Port in the IDE.

Thank you for using C# Corner