

Source: C# Corner ([www.c-sharpcorner.com](http://www.c-sharpcorner.com))

PRINT

---

## Article

---



# Home Automation In Windows Forms Using Visual Studio

By **Kishore Chowdary** on **Jan 04 2017**

### Requirements

1. Visual Studio.
2. Arduino IDE.
3. Connecting wires (10 numbers or more).
4. Breadboard.
5. DC motors which can run by just 5 volts power supply (2 numbers).
6. LED light (1 number).

### Introduction

We expect advancement in our day to day life. That is why I have thought of implementing a voice control technology using Arduino to take control over the electrical appliances at my home. Hence, in this article, I will explain how to create an application which helps you control the devices in your home, with your voice commands. So, to make it simple, I will explain how to connect a door which can be opened and closed and a fan along with a light which can be turned on and off, using the voice commands. Click on the following links to know more about home automation using voice controls.

- [Home Automation Using Arduino with Microsoft Visual Studio](#)
- [Turning LED On And Off Using Buttons In Microsoft Visual Studio](#)

### Note

If you feel that you are unable to understand this program, just click on the following link ([Turning LED On and Off using Voice commands](#)) to learn the basic programming of Arduino. Then, it will become very easy for you to implement this technique.

### 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(2, OUTPUT);
  4.   pinMode(4, OUTPUT);
  5.   pinMode(12, OUTPUT);
  6.   pinMode(13, OUTPUT);
  7.   Serial.begin(9600);
  8. }
  9. void loop() {
  10.   incomingdata = Serial.read(); {
  11.     if (incomingdata == 'a') {
  12.       digitalWrite(2, HIGH);

```

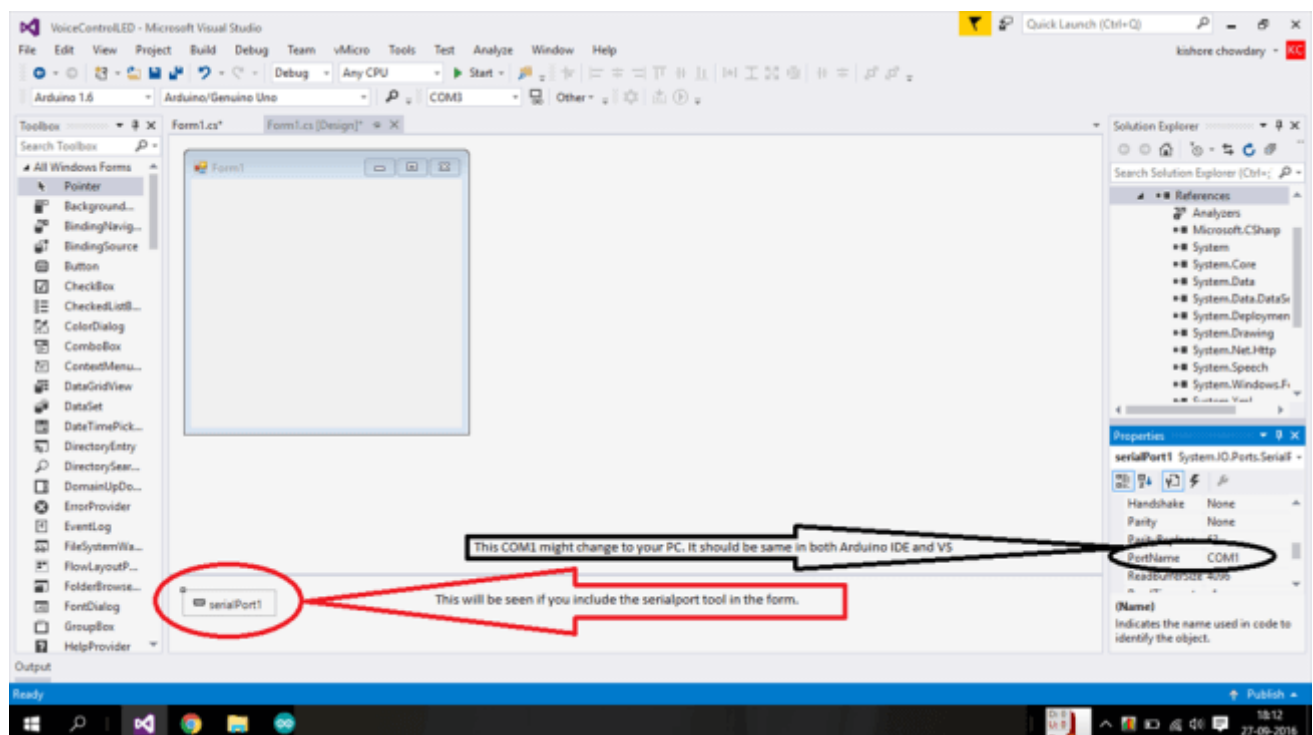
13.    } else if (incomingdata == 'b') {
14.        digitalWrite(2, LOW);
15.    } else if (incomingdata == 'c') {
16.        digitalWrite(4, HIGH);
17.    } else if (incomingdata == 'd') {
18.        digitalWrite(4, LOW);
19.    } else if (incomingdata == 'e') {
20.        digitalWrite(12, HIGH);
21.        digitalWrite(13, LOW);
22.        delay(3500);
23.        digitalWrite(12, LOW);
24.    } else if (incomingdata == 'f') {
25.        digitalWrite(12, LOW);
26.        digitalWrite(13, HIGH);
27.        delay(3500);
28.        digitalWrite(13, LOW);
29.    }
30. }
31. }

```

- In this program, I have used a delay of 3.5 seconds in the last two conditions. This is because of the length of the door which I have used in my prototype. You can change the delay length according to your prototype.
- Save this sketch at any location and verify the code.
- Now, connect your Arduino board and upload this sketch into the board.

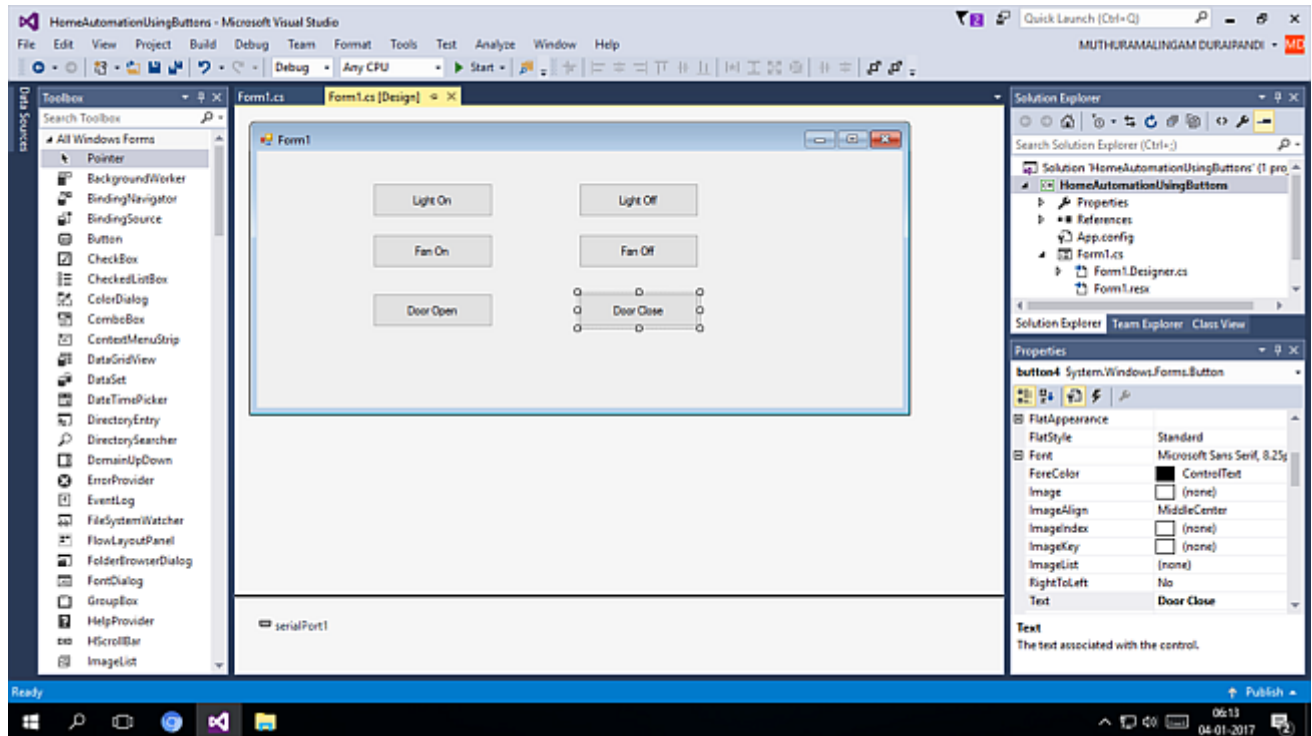
## Step 2 Programming in Visual Studio

- Open Visual Studio and create a 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 tool box, drag and drop the SerialPort tool into the form1.
- This will help your program communicate with the Arduino board.
- Now, drag and drop six buttons in the form to make use of them for controlling the lights and fan.



## Step 3 Creating buttons

- Drag and drop six buttons in the form.
- Name them as LIGHT On, LIGHT Off, FAN On, FAN Off, DOOR Open, and DOOR Close.
- Once you finish these all, the form looks like below.



#### Step 4 Writing code for controlling and using buttons

- The controls will work as they are if you build your prototype by connecting a DC motor with a small fan wing to work like a fan and another DC motor with a door which can be moved up and down like a shutter door and also an LED which can work like a light.
- Remember again that you are building a prototype for an automated house and not a real house.
- Double click on the "Light On" button. You will get to a coding page. There, you must add a small piece of code for the button click event.
- Add the button click event code for each and every separate button.
- Have a look over the code.
  1. using System;
  2. using System.Windows.Forms;
  3. namespace HomeAutomationUsingButtons {
  4. public partial class Form1: Form {
  5. public Form1() {
  6. InitializeComponent();
  7. }
  8. private void button1\_Click(object sender, EventArgs e) {
  9. serialPort1.Open();
  10. serialPort1.Write("a");
  11. serialPort1.Close();
  12. }
  13. private void button6\_Click(object sender, EventArgs e) {
  14. serialPort1.Open();
  15. serialPort1.Write("b");
  16. serialPort1.Close();
  17. }
  18. private void button2\_Click(object sender, EventArgs e) {
  19. serialPort1.Open();
  20. serialPort1.Write("c");
  21. serialPort1.Close();

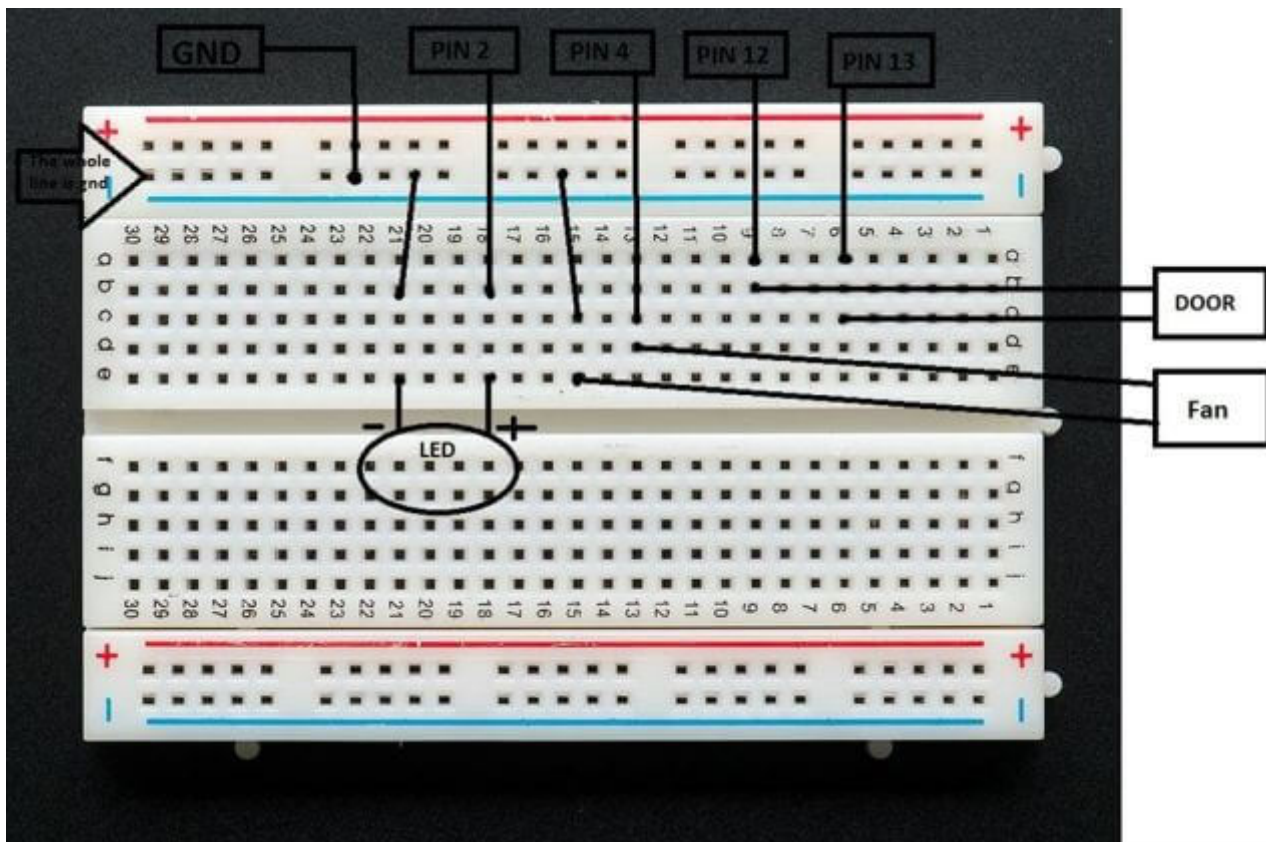
```
22.     }
23.     private void button5_Click(object sender, EventArgs e) {
24.         serialPort1.Open();
25.         serialPort1.Write("d");
26.         serialPort1.Close();
27.     }
28.     private void button3_Click(object sender, EventArgs e) {
29.         serialPort1.Open();
30.         serialPort1.Write("e");
31.         serialPort1.Close();
32.     }
33.     private void button4_Click(object sender, EventArgs e) {
34.         serialPort1.Open();
35.         serialPort1.Write("f");
36.         serialPort1.Close();
37.     }
38. }
39. }
```

### Step 5 Giving the connections in the breadboard

- According to the program, I have used pin 2 of the Arduino for light, pin 4 for the fan, and pin 12, 13 for the door.
- Since the door needs to move in both directions, we need to make the dc motor of the door run in both directions.
- Hence, the door is given with two pin connections.

### Connecting the setup

1. Take a ground connection from the Arduino and connect it to the breadboard.
2. Take pin 2 and connect it to the LED along with the ground connection.
3. Take pin 4 and connect it to the DC motor along with the ground connection which will act as a fan.
4. Take pins 12, 13 and connect each of them to one end of the motor so that it can run in both directions.
5. Take a look over the image for clarification.



### Working of the setup

1. When you click on Light On button, lights will get turned on and when you click on Light Off button, lights will get turned off.
2. When you click on Fan On button, fan will turn on and when you click on Fan Off button, fan will turn off.
3. When you click on Door Open button, the motor of the door will rotate in a direction and then turn off after a few seconds.
4. When you click on Door Close button, the motor will rotate in another direction and then turn off after few seconds.

### Final step

- If you are unable to build a simulated model, you can just give connections for the LEDs and the DC motors and check the output.
- Here, for the door, the motor will just rotate in bi-directions. Hence, you need to create the door such that it can move up and down.
- Once again, check all your code and the connections.
- Now, click on the buttons and check the output.

### Note

I have a few other concepts also for this automation technique. They include the password, security alarm, scheduled powering on and off of the gadgets, and so on. I hope you will understand and like this concept. Thank you.

