

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

PRINT

Article



Home Automation In UWP Using Raspberry Pi Through Microsoft Visual Studio

By [Kishore Chowdary](#) on Jan 27 2017

Introduction

We are going to create an automation project, using Raspberry Pi. In my previous articles, I have used the same concept of home automation and implemented it by using Arduino. You can find the link here.

- [Home Automation Using Arduino with Microsoft Visual Studio](#)

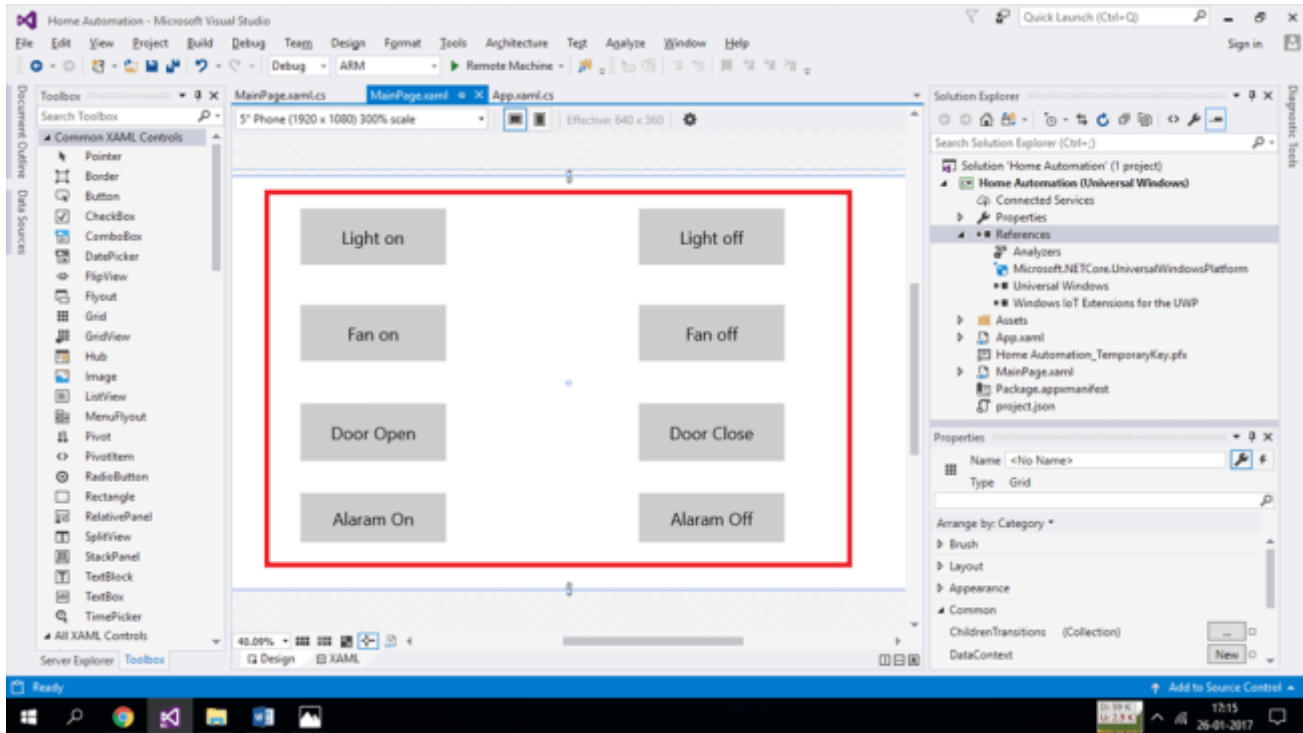
Now, we will see the same concept with the help of Raspberry Pi. Here, in this article, we will create a Universal Windows Application in which the controls are done with the help of buttons. After this, we shall try the speech controls. Now, let us go on with this one.

Requirements

1. Raspberry Pi of any version.
2. LED.
3. DC motors.
4. Male to female jumper wires.

Designing

Create new Universal Windows Application from Visual Studio. Here, in this project, we are going to connect LED, a fan, a door and control them. Hence, we will need a total of six buttons in which three will be for turning on and three will be for turning off. You may place them as you wish. Afterwards, change the content in them as light on, light off, fan on, fan off, door open and door close to recognize the use of them. The designed page will somewhat look, as shown below.

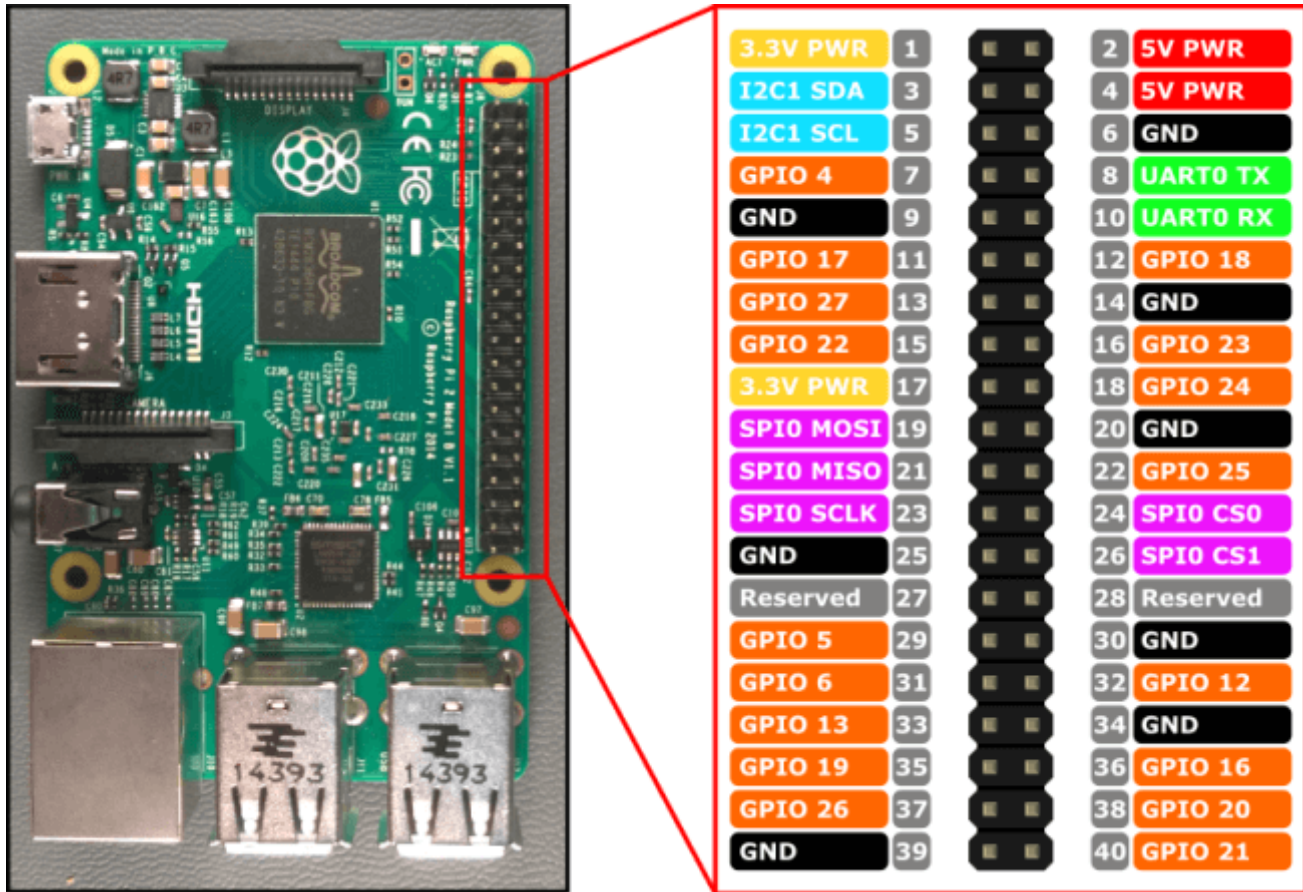


Adding reference file

I think, you all are familiar with the reference files. Here, in the Universal Windows platform, you need to add the reference file Windows IoT extensions for UWP. This extension will help you to run your program in your Raspberry Pi.

Writing the code

Since Raspberry Pi is a microprocessor, the pins in it area little more complicated than the pins of Arduino. It has two types of pins namely GPIO and SPI. Here, GPIO are general purpose input output, whereas SPI are special purpose input output. Now, for our activity, we are going to use GPIO pins. To know more about this pin configuration, have a look at the image given below.



Now, for implementing the concept, I am going to use GPIO pins 6,13,19,26 and a ground pin. In MainPage.Xaml, write the code snippet , which I have given below. Here, you might have to change your Namespace according to the name, which you have given to your Application.

```

1. using Windows.UI.Xaml;
2. using Windows.UI.Xaml.Controls;
3. using Windows.Devices.Gpio;
4. using System.Threading.Tasks;
5. // The Blank Page item template is documented at https://go.microsoft.com/fwlink/?
   LinkId=402352&clcid=0x409
6. namespace Home_Automation {
7.     public sealed partial class MainPage: Page {
8.         GpioPin light, fan, door, door1, alarm;
9.         private void Button_Click_2(object sender, RoutedEventArgs e) {
10.             fan.Write(GpioPinValue.High);
11.         }
12.         private void Button_Click_3(object sender, RoutedEventArgs e) {
13.             fan.Write(GpioPinValue.Low);
14.         }
15.         private void Button_Click_4(object sender, RoutedEventArgs e) {
16.             door.Write(GpioPinValue.Low);
17.             door1.Write(GpioPinValue.High);
18.             Task.Delay(3500).Wait();
19.             door1.Write(GpioPinValue.Low);
20.         }
21.         private void Button_Click_6(object sender, RoutedEventArgs e) {
22.             alarm.Write(GpioPinValue.High);
23.             door.Write(GpioPinValue.High);
24.             door1.Write(GpioPinValue.Low);

```

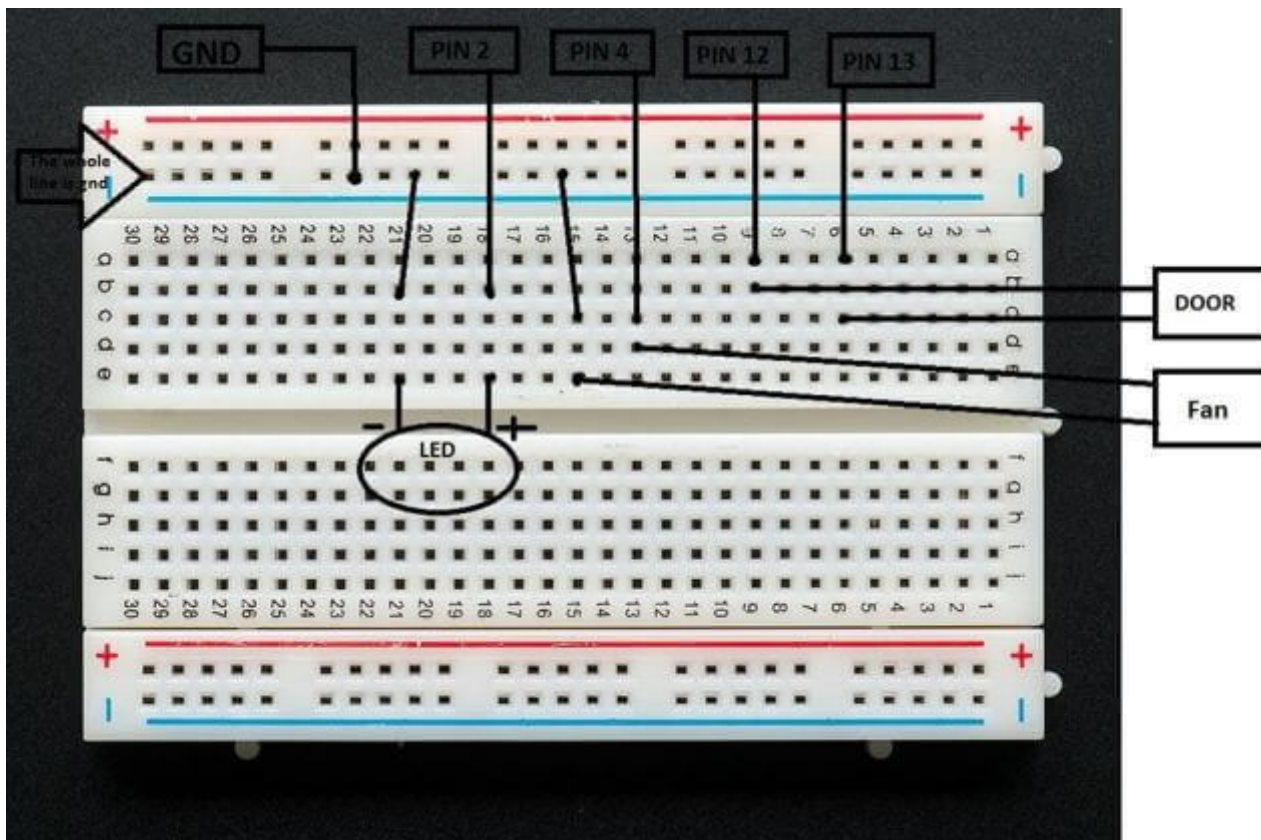
```

25.     Task.Delay(3500).Wait();
26.     light.Write(GpioPinValue.Low);
27.     fan.Write(GpioPinValue.Low);
28.     door.Write(GpioPinValue.Low);
29. }
30. private void Button_Click_7(object sender, RoutedEventArgs e) {
31.     alaram.Write(GpioPinValue.Low);
32. }
33. private void Button_Click_5(object sender, RoutedEventArgs e) {
34.     door.Write(GpioPinValue.High);
35.     door1.Write(GpioPinValue.Low);
36.     Task.Delay(3500).Wait();
37.     door.Write(GpioPinValue.Low);
38. }
39. private void Button_Click_1(object sender, RoutedEventArgs e) {
40.     light.Write(GpioPinValue.Low);
41. }
42. private void Button_Click(object sender, RoutedEventArgs e) {
43.     light.Write(GpioPinValue.High);
44. }
45. public MainPage() {
46.     this.InitializeComponent();
47.     Loaded += MainPage_Load;
48. }
49. private void MainPage_Load(object sender, RoutedEventArgs e) {
50.     var controller = GpioController.Default();
51.     door = controller.OpenPin(19);
52.     door1 = controller.OpenPin(26);
53.     alaram = controller.OpenPin(5);
54.     fan = controller.OpenPin(13);
55.     light = controller.OpenPin(6);
56.     door.SetDriveMode(GpioPinDriveMode.Output);
57.     door1.SetDriveMode(GpioPinDriveMode.Output);
58.     alaram.SetDriveMode(GpioPinDriveMode.Output);
59.     fan.SetDriveMode(GpioPinDriveMode.Output);
60.     light.SetDriveMode(GpioPinDriveMode.Output);
61. }
62. }
63. }

```

Connecting the pins

Connect the ground pin to the breadboard, using the jumper wires. Subsequently connect pin 6 for using LED, pin 13 for using the fan and pin 19 and 26 both for using the door, since DC motor of the door must rotate in the dual directions. Connect your devices according to their pins and ground but for the door, connect both the positive and negative wires to GPIO pins. Only then will the door functionality work properly. Remember that you are creating only a prototype and not a real model. If you want to create a real model, you might have to use Relay, which can help you to control the real time appliances of your home. Have a look over the image to understand the connections.



Deploying the program

In Raspberry Pi, you will be deploying the program in the device itself. Hence, you should configure your device to Visual Studio and connect it with Raspberry. To do it, choose the remote machine option to run your solution and also use ARM option for the deployment process. I have added some extra functionality called alarm in the code snippet. You can either connect it or leave it as it is. The code will work normally.

Finally, when you make proper connections, you can see the output which is running in your Raspberry Pi. To see the output, you must connect your Pi to a monitor or some other device. You can even use Windows IOT remote client app for this, which I will explain in the next article. Hence, when you click on the buttons, the corresponding action will take place properly. Thank you.

Thank you for using C# Corner