**Practical Assignment**

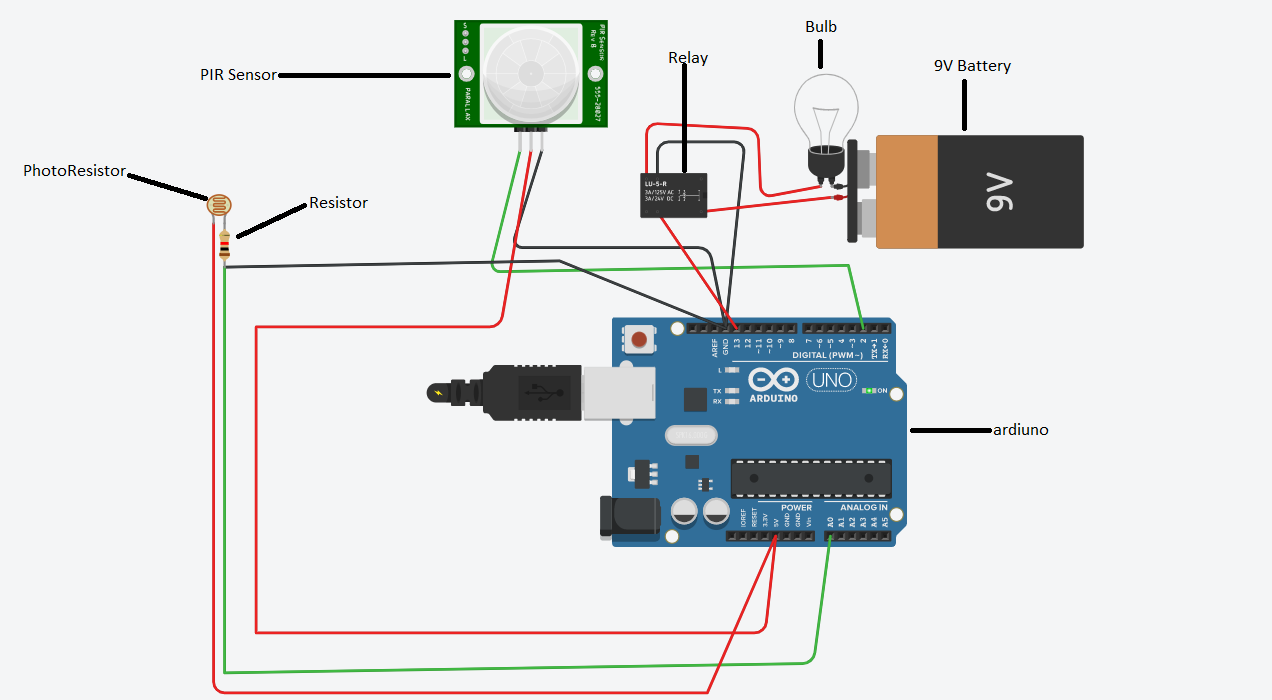
**Question 1. Create a web application based on .NET framework/Android(Based on Specialization Opted by student) App based on following Theme.**

1. Control all the electrical equipment by using Raspberry pi/Arduino.
2. Consider the following details for designing of the interface.
3. Room is having – 2 Tube Lights,2 Fans, smoke sensor with Alarm
4. Dining room is having- 1 Tv, 2 Fans,2 Tube Lights, Motor controlled windows and doors.

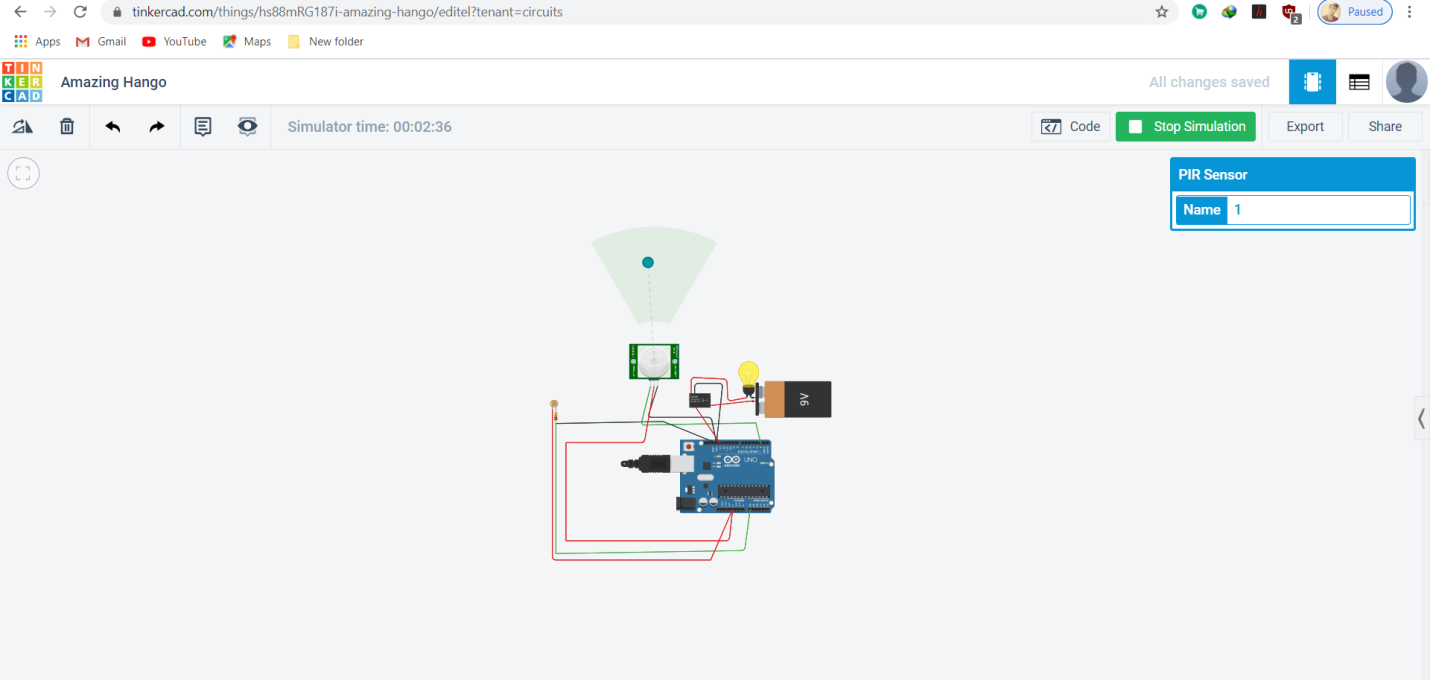
(Note: Motor controlled doors and windows are like automatic when we go closer to door it will be opened automatically either by using PIR sensor,IR sensor,Ultrasonic Sensor.)

1. Kitchen is having- 1 Tube light, 1 Fan, 1 Induction Stove,Gas Sensor.
2. Washroom is having – 1 Light ,1 Water geyser
3. Main Gate is having- 2 Cameras(Pi cameras) which will be controlled from APP.
4. Get the live Level of Water in Water Tank with Alarm at full state.
5. Get the temperature of Room and Take a decision to turn on the AC after getting presence of person inside the room.
6. Auto off features for Lights and Fans on getting No movement in Room for 3 minutes.

**Circuit Diagram for for Turn a bulb when any movement in the room and Turn off a bulb when when there is no movement in the room:-**



**Working of Bulb when there is movement in the room:-**



**Code for turn on the light using PR sensor and relay:-**

int releNO = 13;

int inputPir = 2;

int val = 0;

int resuldoSensorLDR;

int sensorLDR = A0;

void setup()

{

pinMode(releNO, OUTPUT);

pinMode(inputPir, INPUT);

pinMode(sensorLDR, INPUT);

Serial.begin(9600);

}

void loop()

{

val = digitalRead(inputPir);

resuldoSensorLDR = analogRead(sensorLDR);

if(resuldoSensorLDR<600)

{

if(val == HIGH)

{

digitalWrite(releNO, HIGH);

delay(5000);

}

else{

digitalWrite(releNO, LOW);

delay(300);

}

}

else{ digitalWrite (releNO, LOW);

Serial.println(resuldoSensorLDR);

delay(500);

}

}

The main aim of this practical is to control our home appliances from all over the world at any time.

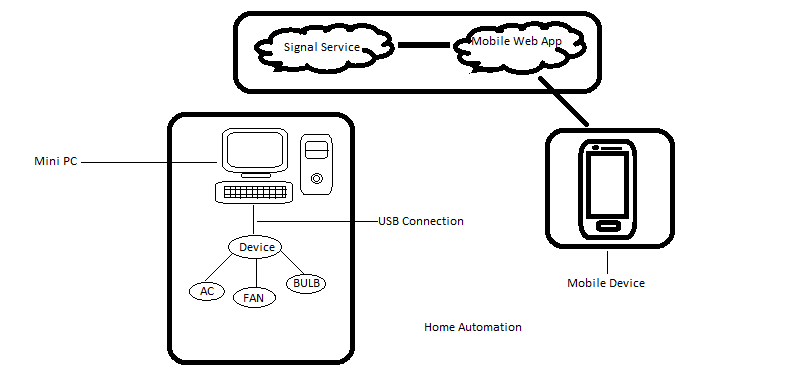
Requirement:-

1. Home automation devices.
2. Visual Studio 2015
3. Cloud Services
4. Internet Connection

In this practical we will control our home appliances from anywhere with the help of internet connection. In this we will control:-

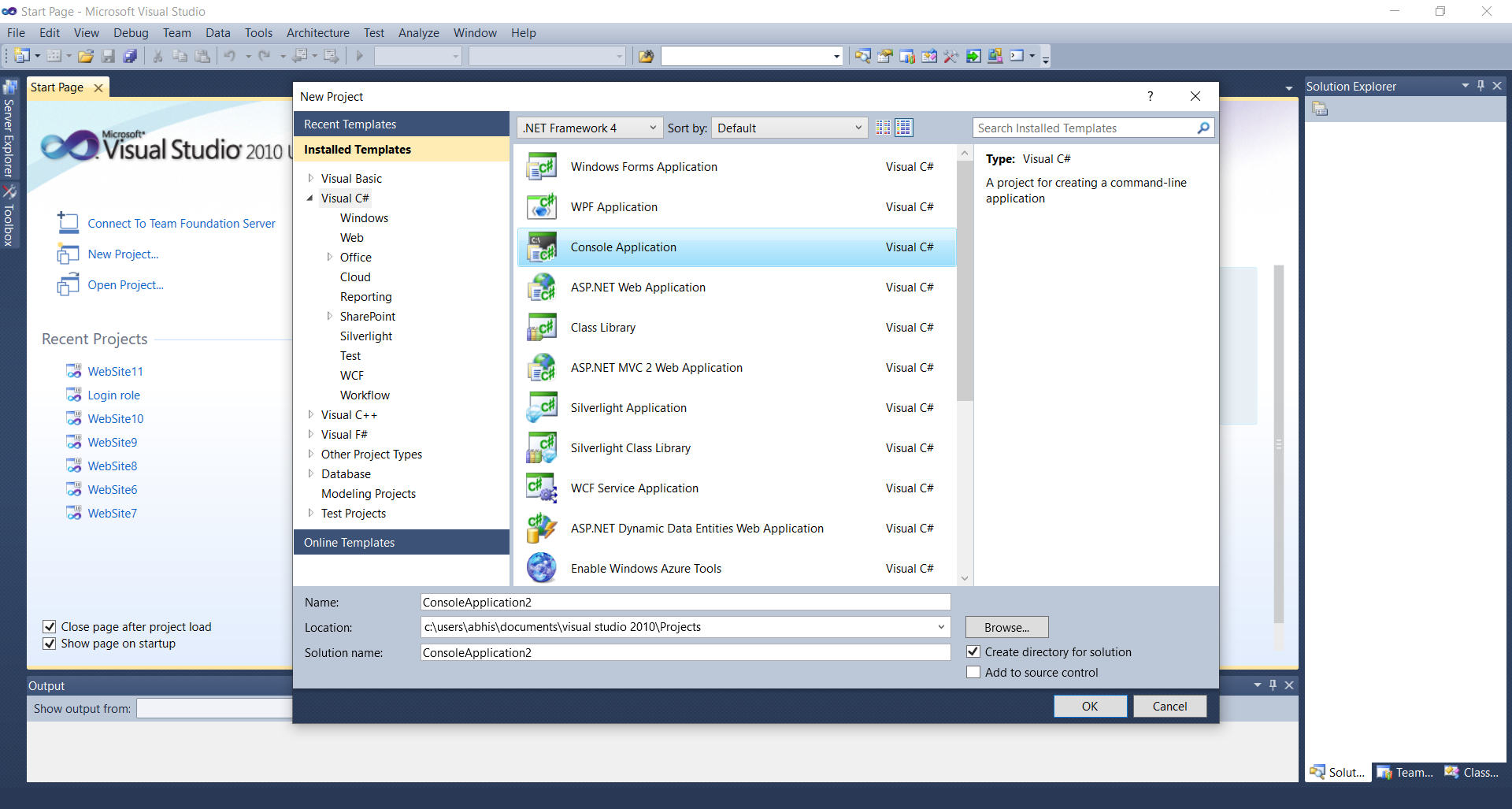
1. Fan
2. Lights
3. Air Condtioners
4. Cameras
5. TV
6. Tubelights

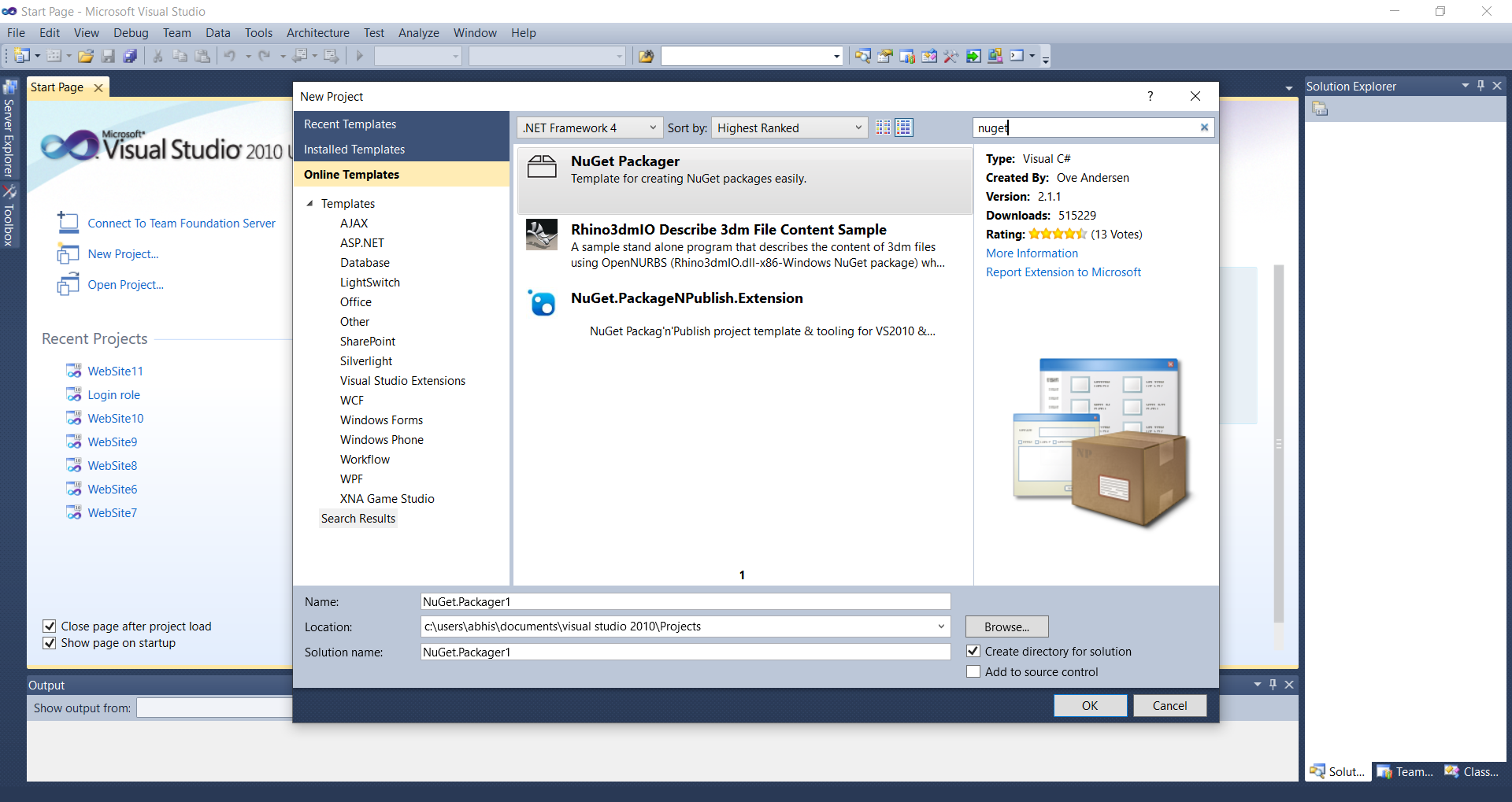
Architecture:-



**Steps to create application in Visual Studio 2010:-**

**Step 1:-** Open Visual Studio 2010 and create a console application & add reference of TopShelf to it using NuGet package manager(With topshelf we can build easily debuggable windows service).





**Step 2.** Add reference to the FluenDwelling API dll file, Programmatical interaction with Insteon hardware devices is easy because some great souls on the Internet have managed to create the API structure.

**Step 3.** We will host the signalR client in the windows service,  
In the NuGet package manager console type following command:-

Install-Package Microsoft.AspNet.SignalR.Client -Version 1.2.2

**Step 4.** Add new class called as "InsteonService.cs" to the windows service project.

using System;

using System.Collections.Generic;

using System.Configuration;

using System.Linq;

using System.Text;

using Microsoft.AspNet.SignalR.Client;

using SoapBox.FluentDwelling;

using Topshelf;

namespace IoT.WindowsService

{

public class InsteonService : ServiceControl

{

private static string ServiceUri;

private static Connection connection;

//Insteon related

private static Plm plm = new Plm("COM4");

public bool Start(HostControl hostControl)

{

ServiceUri = ConfigurationManager.AppSettings["ServiceUri"];

connection = new Connection(ServiceUri, "name=Client");

connection.Received += connection\_Received;

connection.StateChanged += connection\_StateChanged;

Console.WriteLine("Connecting...");

connection.Start().Wait();

string inputMsg;

while (!string.IsNullOrEmpty(inputMsg = Console.ReadLine()))

{

connection.Send(inputMsg).Wait();

}

connection.Stop();

return true;

}

static void connection\_StateChanged(StateChange state)

{

if (state.NewState == ConnectionState.Connected)

{

Console.WriteLine("Connected.");

}

}

static void connection\_Received(string data)

{

//Parse the commands

//Format:

//Name,Command,DeviceName(optional),Port(optional)

string[] command = data.Split(new char[] { ',' }, StringSplitOptions.RemoveEmptyEntries);

if (command.Length > 0)

if (command[0].Equals("Server", StringComparison.InvariantCultureIgnoreCase))

{

string commandName = command[1].ToLower();

if (command.Length > 2)

{

string deviceid = command[2];

switch (commandName)

{

case "turnoff":

{

plm.Network.X10

.House("A")

.Unit(2)

.Command(X10Command.Off);

}

break;

case "turnon":

{

plm.Network.X10

.House("A")

.Unit(2)

.Command(X10Command.On);

}

break;

}

}

else

{

switch (commandName)

{

case "getlist"://not tested

{

var database = plm.GetAllLinkDatabase();

string devices = "";

if (!plm.Error)

{

foreach (var record in database.Records)

{

devices += record.DeviceId.ToString();

}

}

else

{

devices += "Error";

}

connection.Send(devices).Wait();

}

break;

}

}

}

}

public bool Stop(HostControl hostControl)

{

if (connection.State == ConnectionState.Connected)

connection.Stop();

return true;

}

}

}

Step 5. In the “Program.cs” write the following code.

using System;

using System.Collections.Generic;

using System.Linq;

using System.Text;

using Topshelf;

using Topshelf.HostConfigurators;

using Topshelf.ServiceConfigurators;

namespace IoT.WindowsService

{

class Program

{

static void Main(string[] args)

{

var host = HostFactory.New(ConfigureHost);

host.Run();

}

private static void ConfigureHost(HostConfigurator x)

{

x.Service<insteonservice>(ConfigureService);

x.SetDisplayName("Insteon Service");

x.SetDescription("Insteon Controller Service");

x.SetInstanceName("Insteon");

x.SetServiceName("Insteon");

x.StartAutomatically();

x.RunAsPrompt();

}

private static void ConfigureService(ServiceConfigurator<insteonservice> sc)

{

sc.ConstructUsing(() => new InsteonService());

// the start and stop methods for the service

sc.WhenStarted((s, hostControl) => s.Start(hostControl));

sc.WhenStopped((s, hostControl) => s.Stop(hostControl));

}

}

}

</insteonservice></insteonservice>

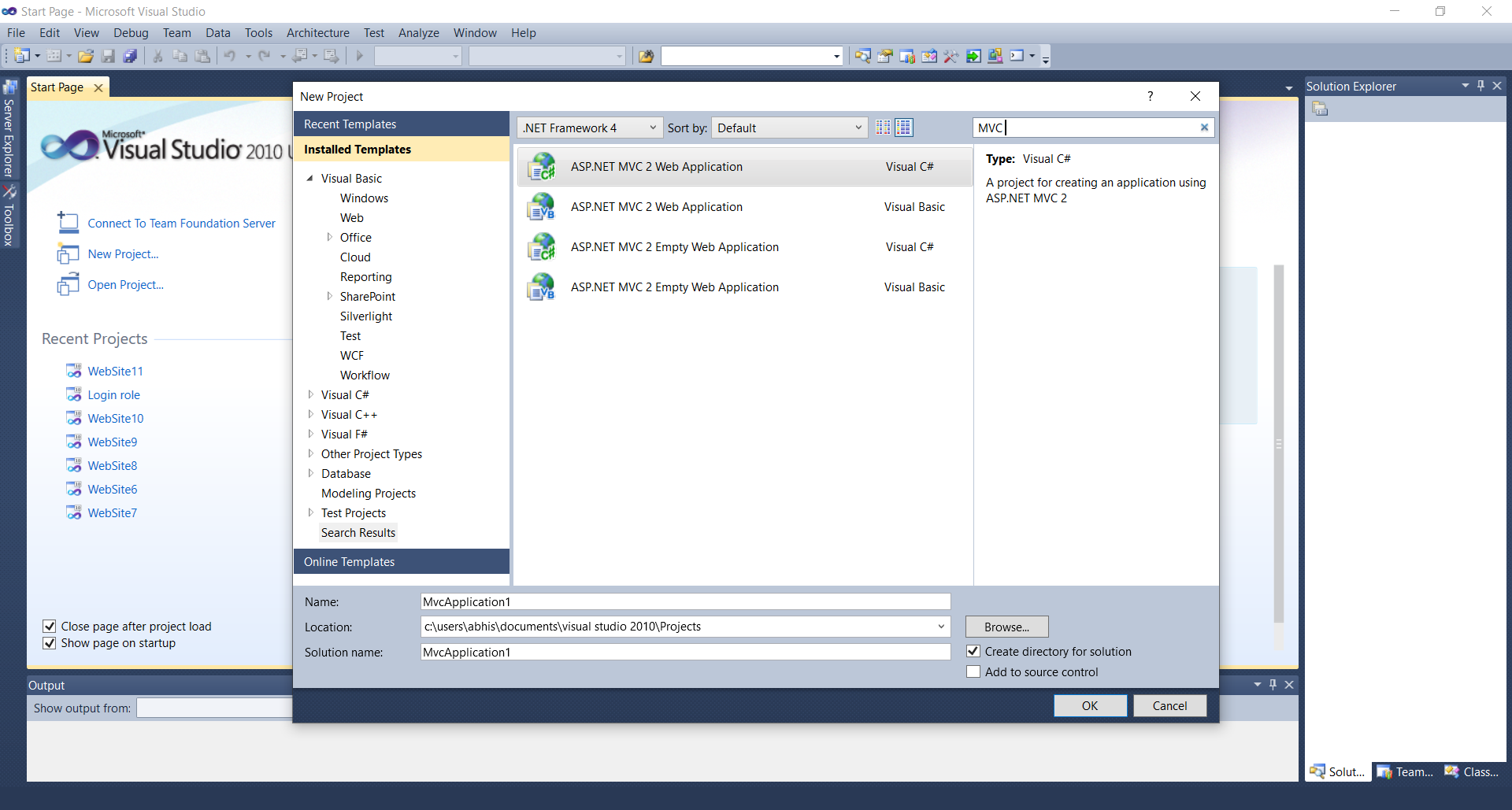
**Step 6.** Finally in the "App.config" file in the "AppSettings" section we need to add following lines of code:-

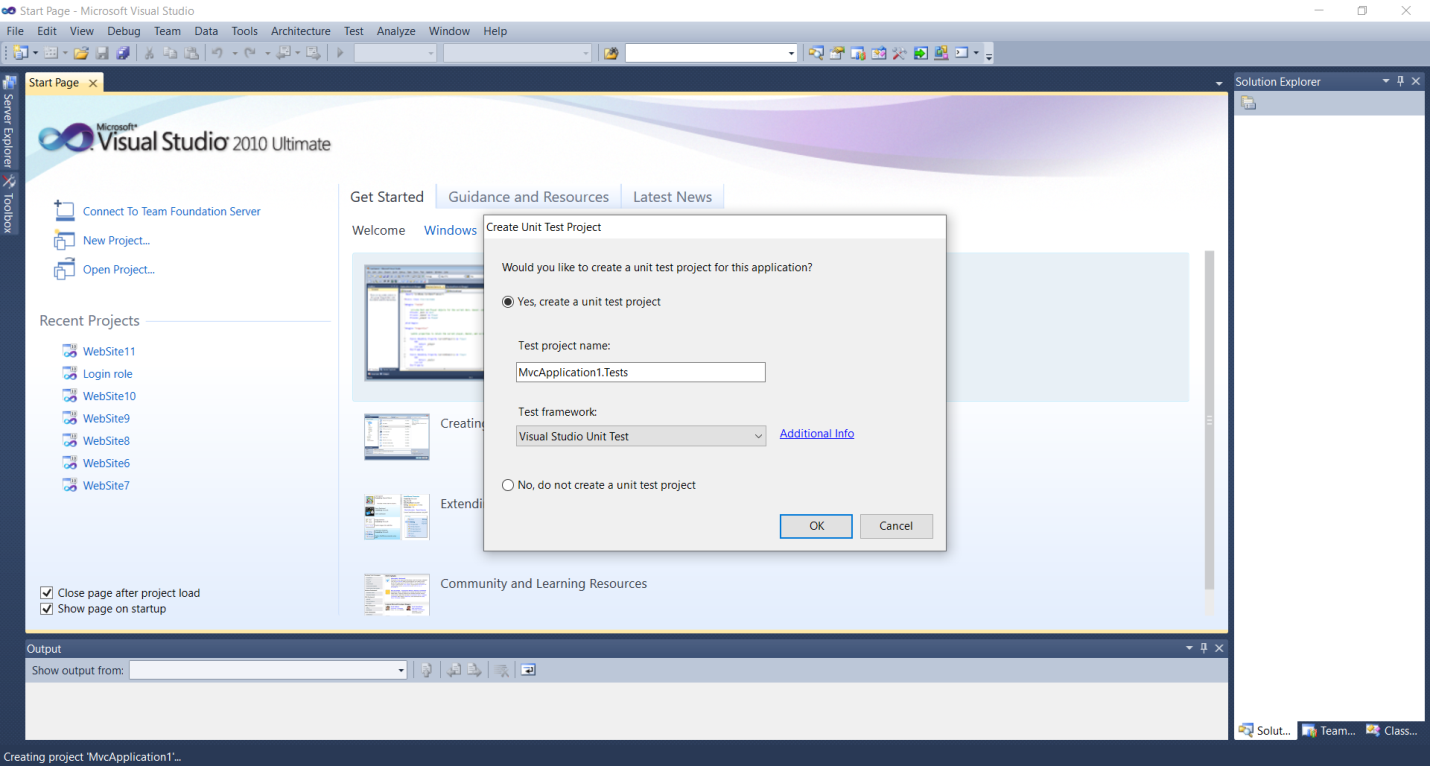
<add key="ServiceUri" value="http://insteoniot.azurewebsites.net/echo"/>

Now we have to create remote control service from ASP.NET MVC and Signal R.

1. We will host the signalR service in asp.net mvc4 web application.

2. Create a new asp.net MVC2 project in visual studio:-





**Step 7.** Install SignalR server in the project using package manager console

Install-Package Microsoft.AspNet.SignalR -Version 1.2.2

**Step 8.** Create new folder "SignalR" in the project and add a class called as "MyConnection.cs" to the project:-

using System;

using System.Collections.Generic;

using System.Linq;

using System.Threading.Tasks;

using System.Web;

using Microsoft.AspNet.SignalR;

namespace IoT.Mobile.SignalR

{

public class MyConnection : PersistentConnection

{

protected override Task OnConnected(IRequest request, string connectionId)

{

return Connection.Broadcast(string.Empty);

}

protected override Task OnReceived(IRequest request, string connectionId, string data)

{

// Broadcast data to all clients

string msg = string.Format(

"{0},{1}", request.QueryString["name"], data);

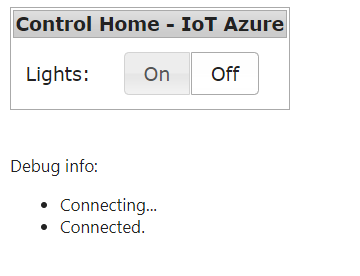
return Connection.Broadcast(msg);

}

}

}

**Step 9.** In previously created MVC4 web application we will create a controller called as "HomeController" and then add a view "Index.cshtml",



the view "Index.cshtml" contains radio buttons for turning the lights on and off and the code for calling signalR service.

@{

ViewBag.Title = "Control Home - IoT Azure";

}

<table class="ui-widget ui-widget-content">

<tr>

<td colspan="2" class="ui-widget-header">@ViewBag.Title</td>

<tr>

<td style="padding:10px">

Lights:

</td>

<td style="padding:10px">

<div id="radio">

<input type="radio" class="radio" id="radio1" name="radio"><label for="radio1">On</label>

<input type="radio" class="radio" id="radio2" name="radio" checked="checked"><label for="radio2">Off</label>

</div>

</td>

</tr>

</table>

<br/>

<br/>

Debug info:

<ul id="messages"></ul>

@section scripts

{

<script type="text/javascript">

function htmlEncode(value) {

return $("<div/>").text(value).html();

}

function addMsg(msg) {

if (msg != "") {

$("#messages").append("<li>" + htmlEncode(msg) + "</li>");

}

}

$(function() {

var connection = $.connection("/echo", "name=Server", true);;

connection.received(function(data) {

addMsg(data);

});

connection.error(function(err) {

addMsg("Error: " + err);

});

addMsg("Connecting...");

connection.start(function() {

addMsg("Connected.");

});

//UI

$("#radio").buttonset();

$('#radio').change(function () {

var $this = $('.radio:checked');

if ($this.attr("id")=="radio1") {

connection.send("turnon,com4");

} else {

connection.send("turnoff,com4");

}

});

});

</script>

}

<add key="ServiceUri" value="http://insteoniot.azurewebsites.net/echo"></add>

**HARDWARE/SOFTWARE SETUP**

1. Connect the insteon 2413U with Minix neo z64 windows mini PC using USB cable

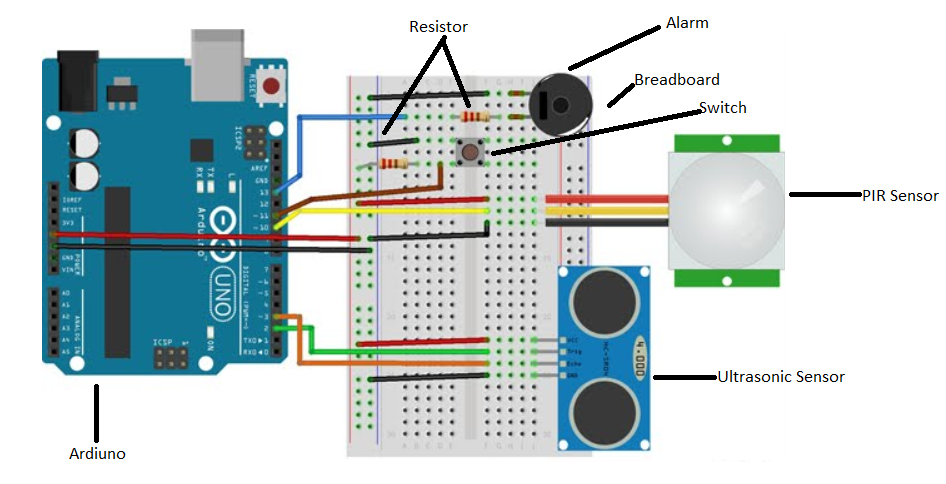
2. Install windows service on the Minix Neo z64 windows mini pc using command line

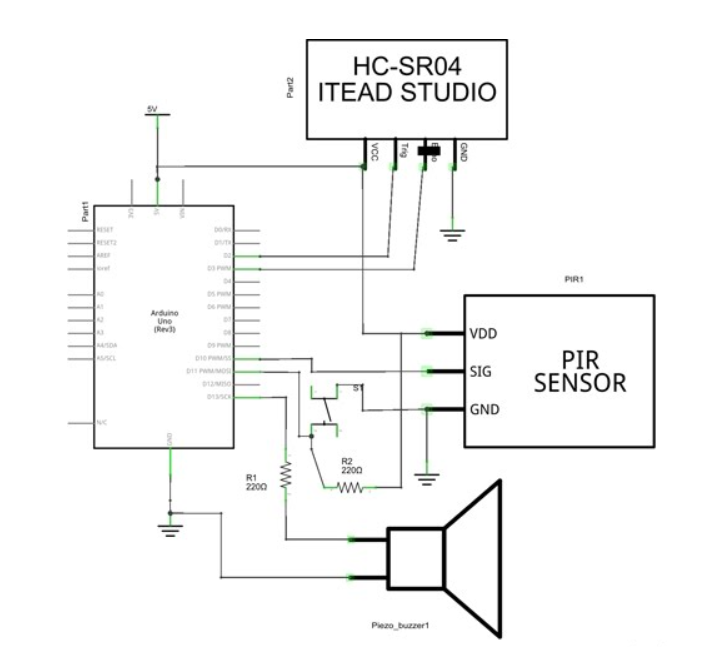
IoT.WindowsService.exe install

3. Host the mobile app in cloud using Windows Azure Cloud Hosting.

**Implementation of Ultrasonic sensor with PIR:-**

**Circuit Diagram:-**

****

****

int switch2 = 11;

int echo = 2;

int trigger = 3;

int buzzer = 13;

int pir = 10;

int distance = 0;

boolean alarm = LOW;

long unsigned int lowIn;

long unsigned int pause = 5000;

boolean lockLow = true;

boolean takeLowTime;

int calibrationTime = 30;

float sinVal;

int toneVal;

Bounce switch2Bounce = Bounce();

NewPing sonar(trigger, echo, MAX\_DISTANCE);

void setup() {

Serial.begin(9600);

pinMode(pir, INPUT);

pinMode(buzzer, OUTPUT);

pinMode(switch2,INPUT);

switch2Bounce.attach(switch2);

switch2Bounce.interval(100);

digitalWrite(pir, HIGH);

Serial.print("calibrating sensor ");

for (int i = 0; i < calibrationTime; i++) {

Serial.print(".");

delay(1000);

}

Serial.println(" done");

Serial.println("SENSOR ACTIVE");

delay(50);

}

void loop() {

readPIR();

alarmState();

}

void readPIR() {

if (digitalRead(pir) == HIGH) {

soundAlarm();

pingDist();

if (lockLow) {

lockLow = false;

Serial.println("---");

Serial.print("motion detected at ");

Serial.print(millis() / 1000);

Serial.println(" sec");

delay(50);

}

takeLowTime = true;

}

if (digitalRead(pir) == LOW) {

if (takeLowTime) {

lowIn = millis();

takeLowTime = false;

}

if (!lockLow && millis() - lowIn > pause) {

lockLow = true;

Serial.print("motion ended at "); //output

Serial.print((millis() - pause) / 1000);

Serial.println(" sec");

delay(50);

}

}

}

// Ultrasonic sensor distance reading

void pingDist () {

delay(50); // Wait 50ms between pings (about 20 pings/sec). 29ms should be the shortest delay between pings.

distance = sonar.convert\_cm(sonar.ping\_median(10));

Serial.print("Distance");

Serial.println(distance); // Convert ping time to distance in cm and print result (0 = outside set distance range)

}

////////////////////

// Alarm Sound a sinusiod is generated and outputed on the buzzer pin

void soundAlarm() {

if (alarm==LOW){

for (int x = 0; x < 180; x++) {

// convert degrees to radians then obtain value

sinVal = (sin(x \* (3.1412 / 180)));

// generate a frequency from the sin value

toneVal = 2000 + (int(sinVal \* 1000));

tone(buzzer, toneVal);

delay(1);

}

noTone(buzzer);

}

}

/////////////////

//Check the status of the switch and detect the falling edge.

void alarmState() {

if (switch2Bounce.update()) {

if (switch2Bounce.fell()) {

alarm = !alarm;

Serial.print("Alarm=");

Serial.println(alarm);

}

}

}

So, in the above code when the pir sensor detect the motion and the ultrasonic sensor measure distance the alarm get triggered.

**To access camera over internet in iot we use the following code:-**

import urllib

import cv2,time

import numpy as np

first\_frame = None

url= 'http://192.168.2.52:8080/shot.jpg’

imgResp = urllib.urlopen(url)

imgNp = np.array(bytearray(imgResp.read()),dtype=np.uint8)

img = cv2.imdecode(imgNp,-1)

cv2.namedWindow(‘image’,cv2.WINDOW\_NORMAL)

cv2.resizeWindow(‘image’,600,600)

i=0

while True:

imgResp=urllib.urlopen(url)

imgNp=np.array(bytearray(imgResp.read()),dtype=n.uint8)

img=cv2.imdecode(imgNp,-1)

cv2.imshow(‘image’,img)

cv2.imwrite(‘test.jpg’’,img)

key=cv2.waitKey(1)

if key==ord(‘q’):

break