## Assignment No :-01

**Assignment Name: –** Design and implement basics embedded circuits

### Automatic Alarm system- Alarm should get tigger by senor

1. Timer based buzzer
2. Sensor based Counting device

## Name :- Nishikant Pawar Class: MSc Cs Part 1

**Roll no: - Date: -**

#### Automatic Alarm system- Alarm should get tigger by senor: Requirements:

* + **1 Power source (DC 12.5V)**

#### 1 Buzzer

* + **1 Red LED Unit**

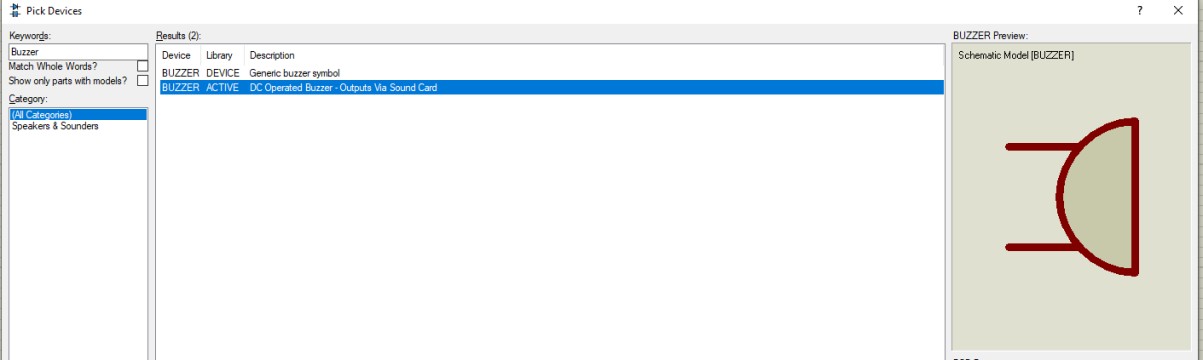
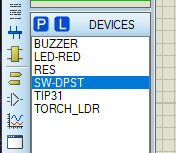
#### 1 Rain Sensor (SW-DPST)

* + **1 Transistor (TIP31)**

#### 2 Resistors – 1)100Ω 2)1K Ω

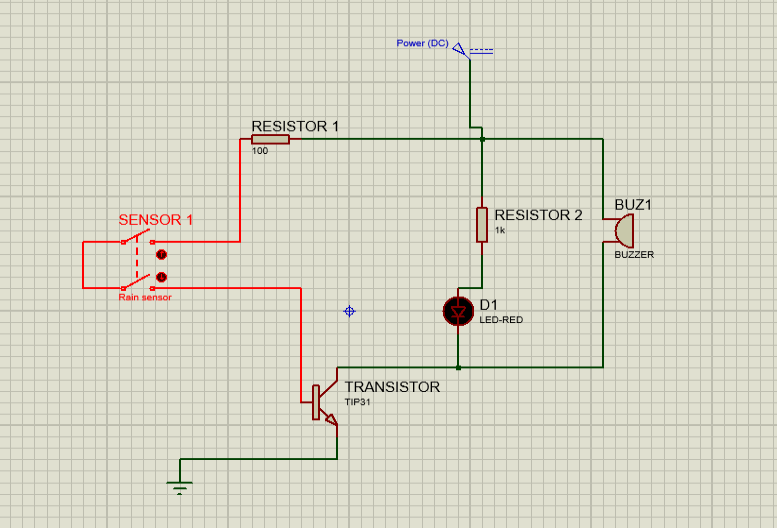
**Step 1: Open Proteus and create a new project.**

#### Step 2: Under the devices tab, select the required components.

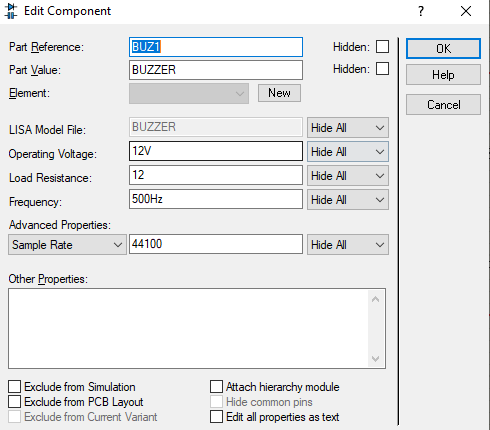
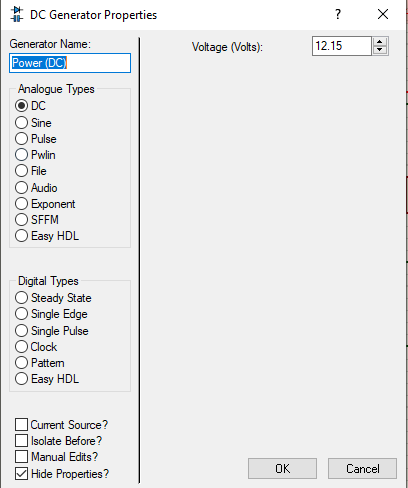


Note: Similarly search for the other devices from the given requirements.

#### Step 3: Make the following configuration.



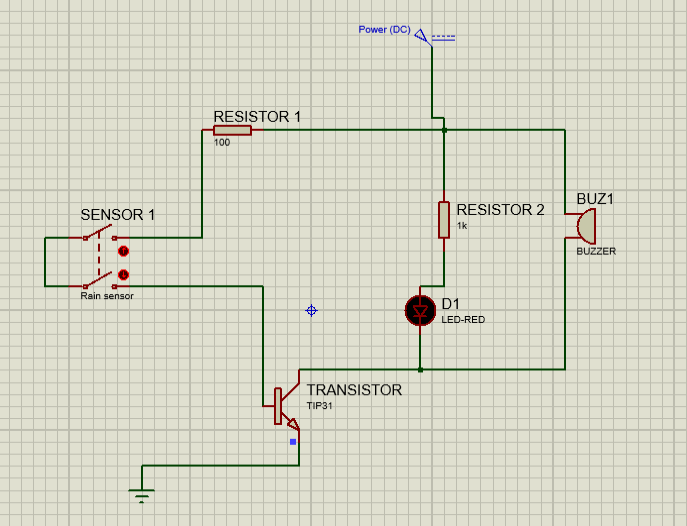
Note: You can configure each device by double clicking on it.



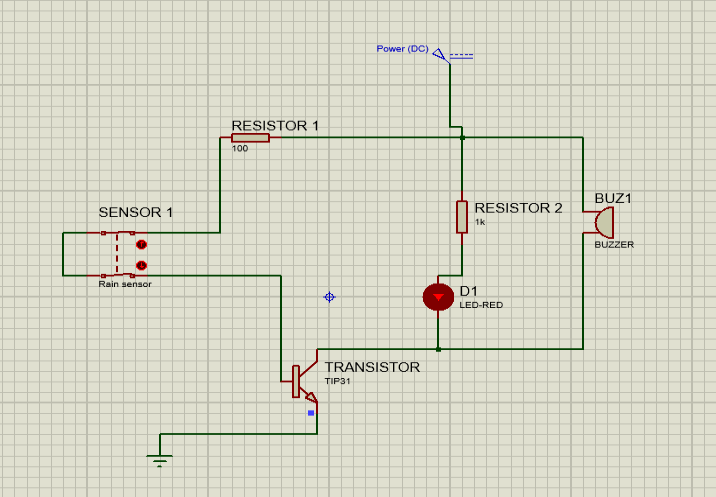
Note: You can perform the connections by connecting all the nodes using the selection tool.

#### Step 4: Start the simulation to check whether all the components are connected properly, and that the device does the intended task.

When the sensor is in the off state:



When the sensor is in the on state (The buzzer will create a sound that can be heard via your system speakers):



#### Step 5: Stop the simulation and note down your observations.

1. **Timer based buzzer:**

#### Requirements:

* + **1 Battery 12V**

#### 1 Capacitor 1µF

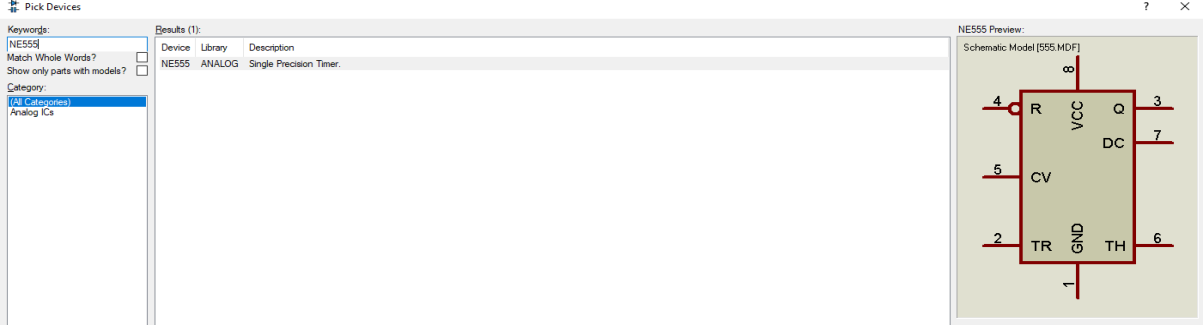
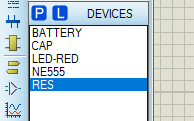
* + **1 Red LED**

#### 3 Resistors – R1 220 Ω, R2 100k Ω, R3 470k Ω

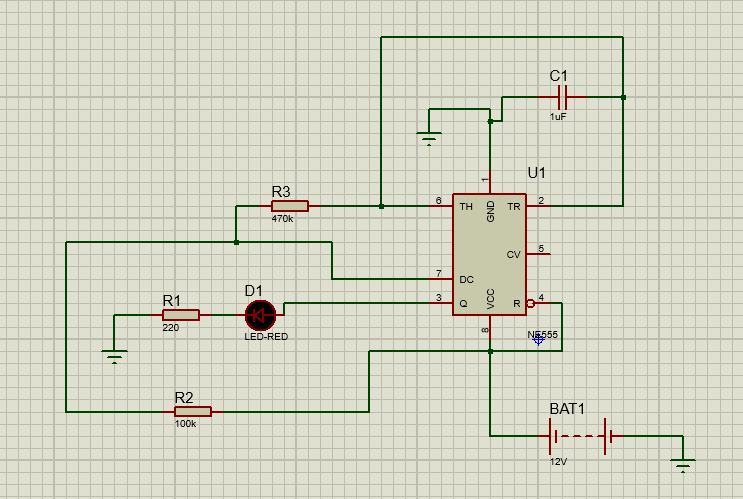
* + **1 PCB (NE555)**

#### Step 1: Open Proteus and create a new project.

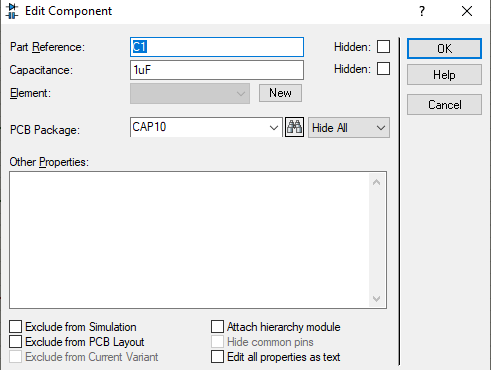
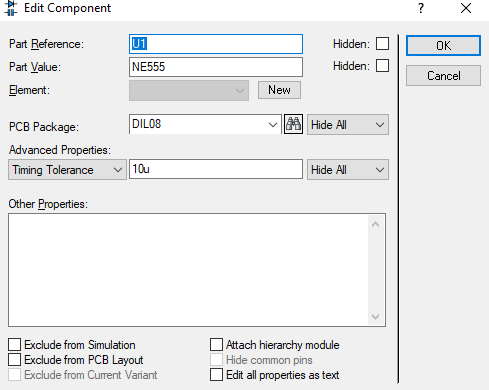
**Step 2: Under the devices tab, select the required components.**



#### Step 3: Make the following configuration.



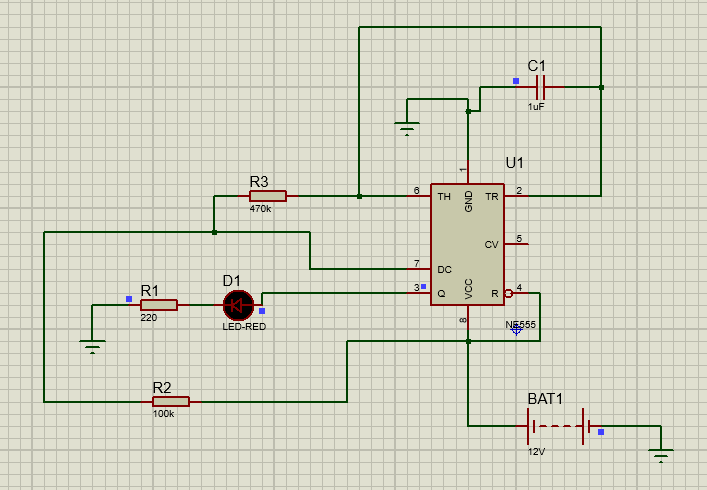
Note: You can configure each device by double clicking on it.



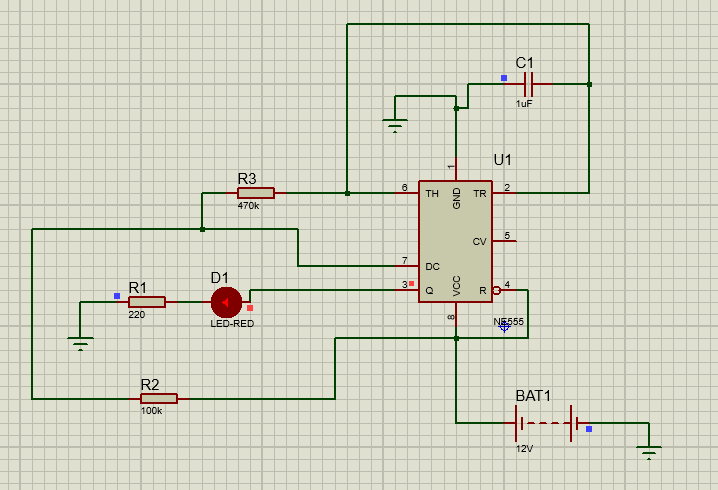
Note: You can perform the connections by connecting all the nodes using the selection tool.

#### Step 4: Start the simulation to check whether all the components are connected properly, and that the device does the intended task.

Since the device is based on a timer, once the simulation starts the Red LED will keep blinking based on the configuration made to the PCB (NE555).



#### Step 5: Stop the simulation and note down your observations.



#### 3. Sensor based Counting device:

#### Requirements:

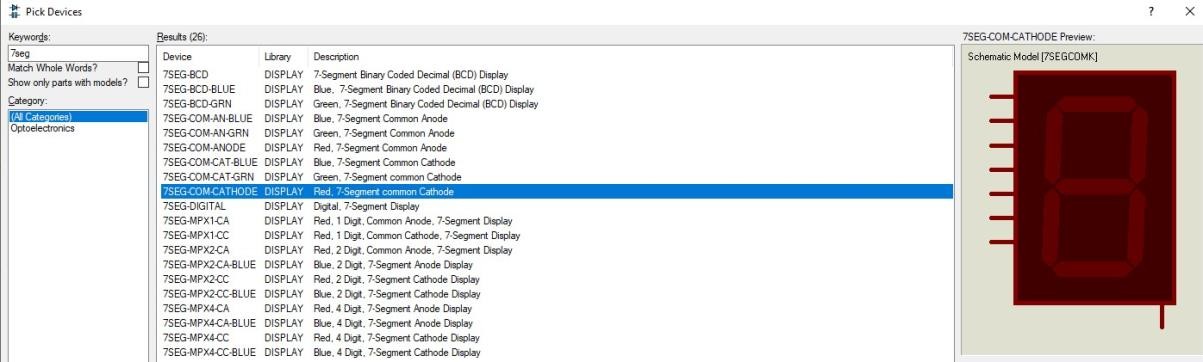
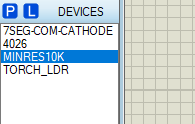
* + **1 PCB (4026)**

#### 1 Resistor (MINRES10K) R1 10kΩ

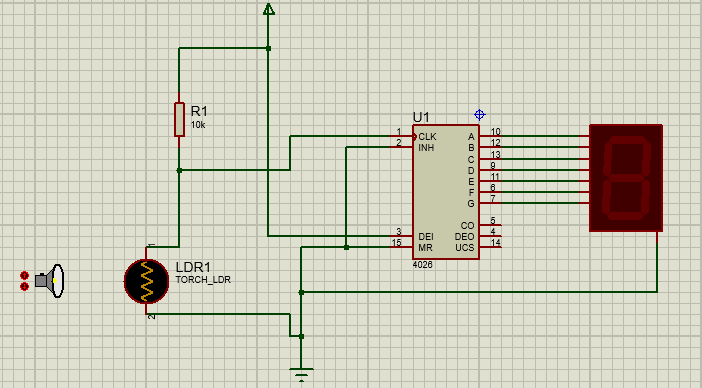
* + **1 LED Screen (7SEG-COM-CATHODE)**

#### 1 Torch and Light Dependent Resistor Step 1: Open Proteus and create a new project.

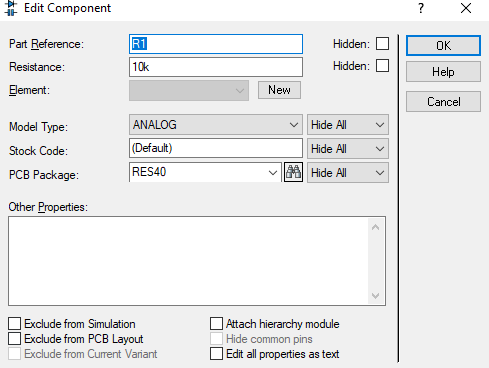
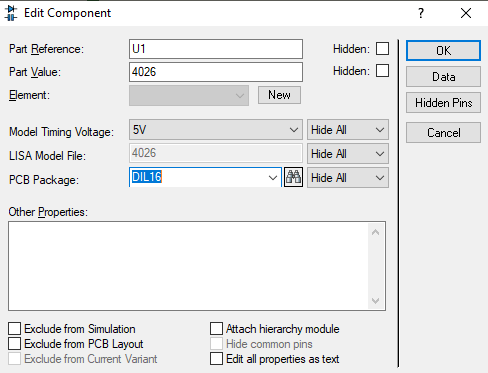
**Step 2: Under the devices tab, select the required components.**



#### Step 3: Make the following configuration.



Note: You can configure each device by double clicking on it.

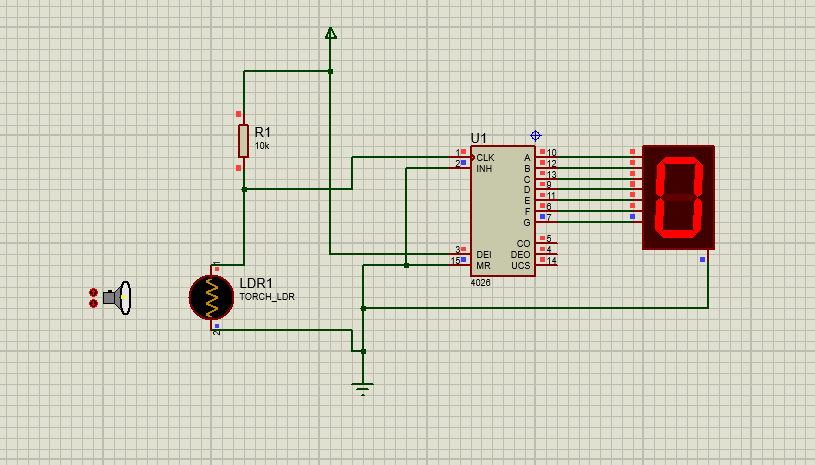


Note: You can perform the connections by connecting all the nodes using the selection tool.

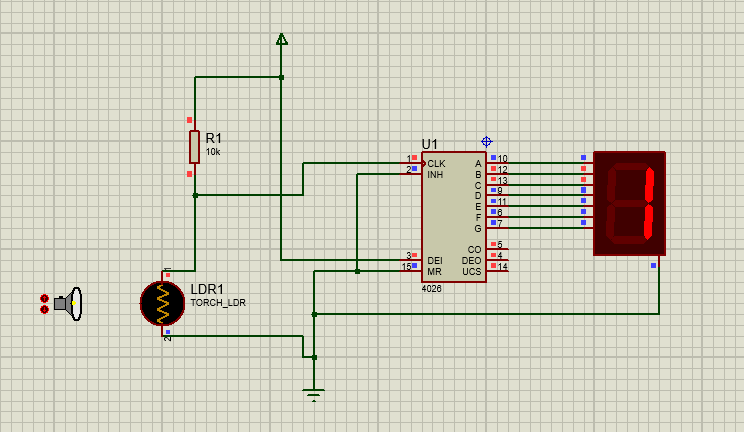
#### Step 4: Start the simulation to check whether all the components are connected properly, and that the device does the intended task.

This device works when a light source is shone upon the light dependent resistor, which will pass a signal to the PCB which will show an increment of single digits on the LED Screen.

Using Proteus start the simulation. Press the ‘+’ and ‘-’ on the TORCH\_LDR component to change the torch distance from the LDR.



As the torch is brought closer to the LDR and taken away from it, the LDR will detect the light source it will send a signal to the PCB which will send an increment signal to the LED screen which will show digits up to ‘9’.



## Assignment No :-02

**Assignment Name: –**Develop an IOT Application for Motion Detection.

## Name :- Nishikant Pawar Class: MSc Cs Part 1

**Roll no: - Date: -**

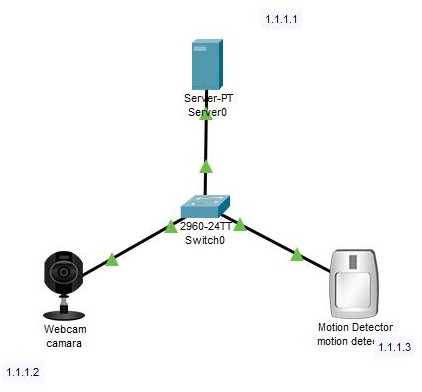
**Step 1 –** Take four devices (Switch-2960, Server, Webcam, Motion detector).

**Step 2 –** Connect those devices through copper straight cable.

1. Webcam to switch (F0 –F20)
2. Motion Detector to switch (F0 - F16)
3. Server to switch (F0 – F13)

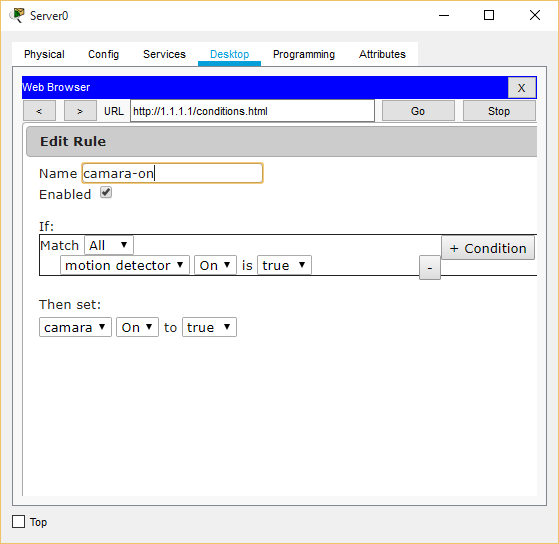
**Step 3 –** Provide IP address to all devices

1. Server – 1.1.1.1
2. Webcam – 1.1.1.2
3. Motion Detector – 1.1.1.3

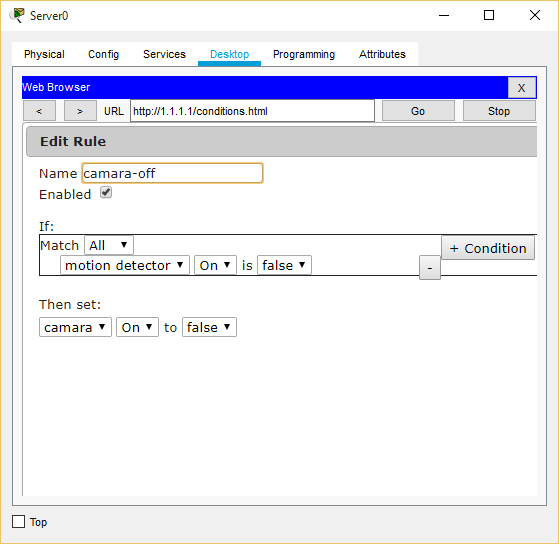


**Step 4 –** In server go to the web browser and pass the URL(1.1.1.1) and click on GO. Then register with the Username and password and sign in, and go to the condition and add the conditions for webcam and motion detector.

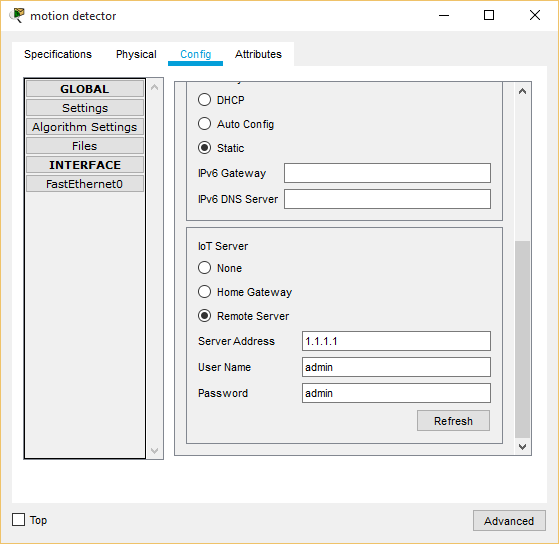
#### Condition for camera on :-



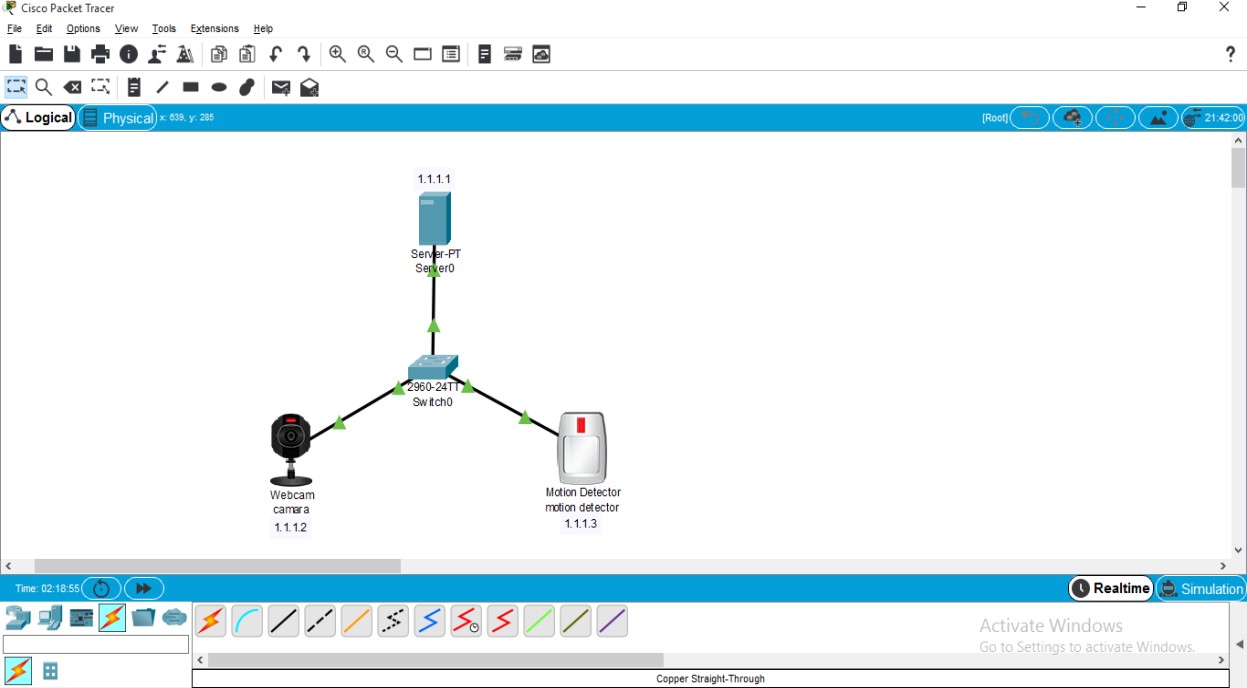
**condition for camera off :-**



**Step 5 –**Set the username and password for Motion Detector**.**



**Output -**



# Assignment No:-03

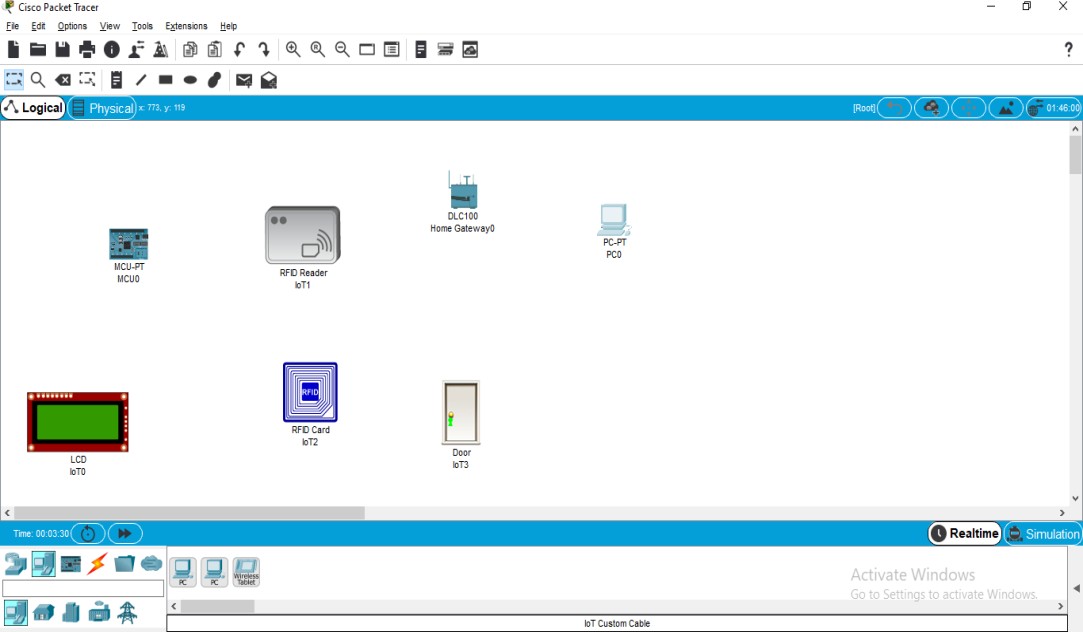
**Assignment Name:-** Built an IoT system to send ticket before entering the bus.

**Name: Nishikant Pawar** **Roll No:-**

**Class:-** Msc-1(CS) **Date:-**

## Requirements:-

### Take devices i.e. LCD , MCU-PT , RFID Reader , RFID Card , DLC100(Home Gateway ), Door , Pc :



Door Advanced  IO config  select PT-IOT-NM-1W

#### Step 1 :-

Connect LCD (D0) to MUC-PT(D0) by using IOT custom cable

#### Step 2 :-

Click on RFID reader Advanced IO config and set analog slot value as 2

#### Step 3 :-

Connect RFID reader(A0) to MUC-PT(A0) by using IOT custom cable.

#### Step 4 :-

Click on the pc and off the red button and remove module and add PT-HOST-NM-1W module and on the button .

Then go to configwireless0 and set SSID as HomeGateway

**else:**

#### setState(1)

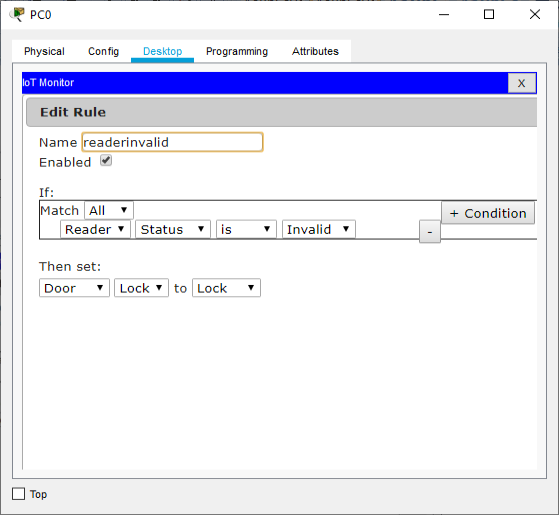
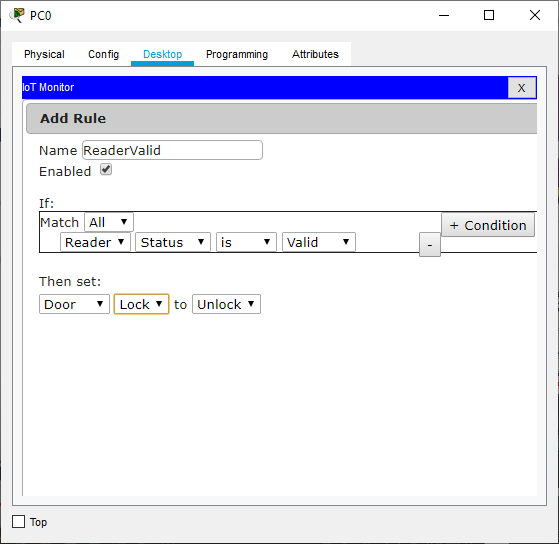
delay(DELAY\_TIME)

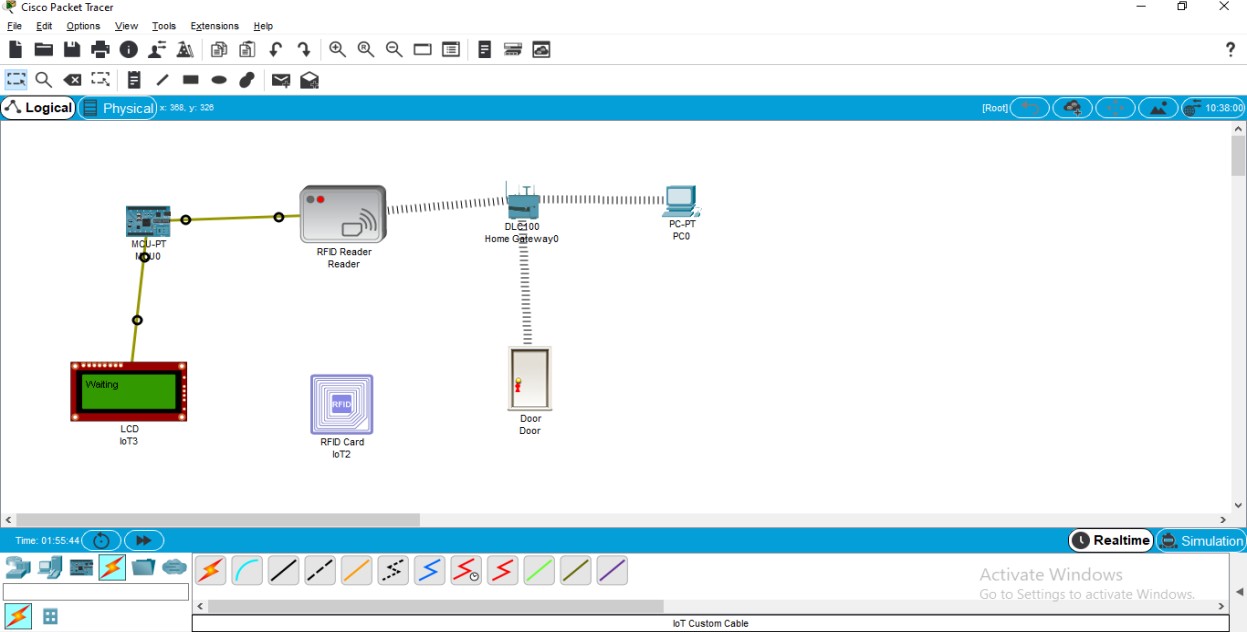
and run the code

#### Step 10 :-

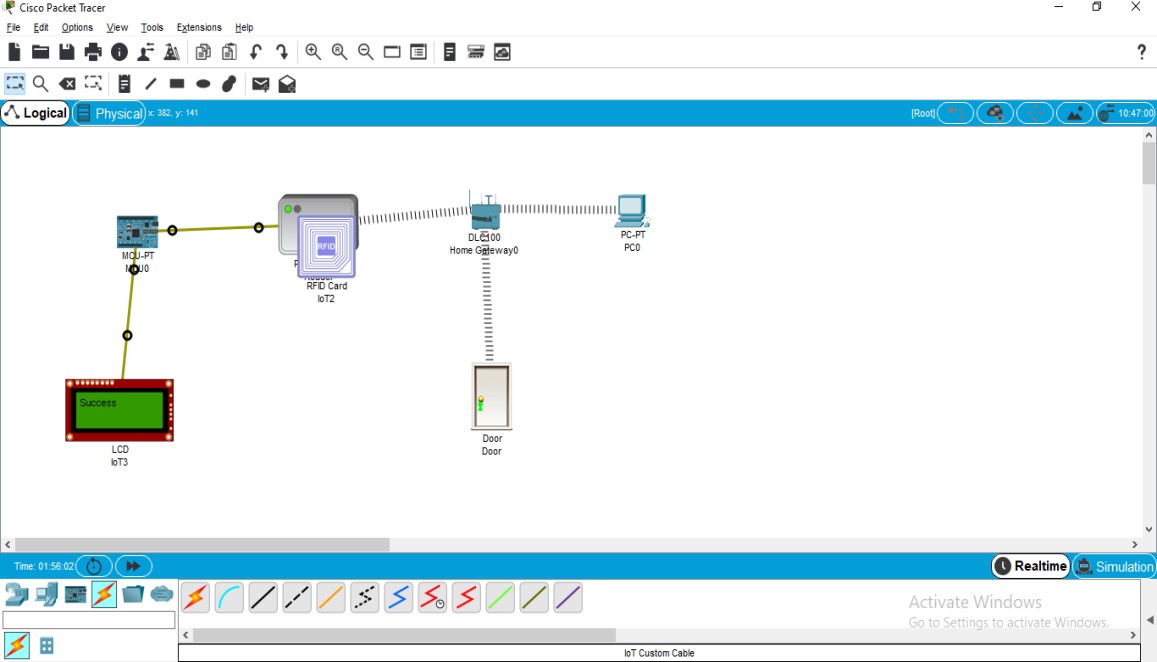
Click on pc  Desktop IOT monitor  login

Click on **conditions** **Add** and add the below two conditions and click on ok :-





**Step 11:-**



# Assignment No:-04

### **Assignment Name:-** Develop an IoT application for monitoring water levels in tanks andautomatically start the motor to fill the tank if the level goes below

the critical level.

**Name: Nishikant Pawar** **Roll No:-**

**Class:-** Msc-1(CS) **Date:-**

#### Requirements:

* **1 LCD Display (LM016L)**

#### 1 MCU (AT89C51RD2)

* **1 Motor (12V)**

#### 3 Transistors (BC547BP)

* **1 Battery (12V)**

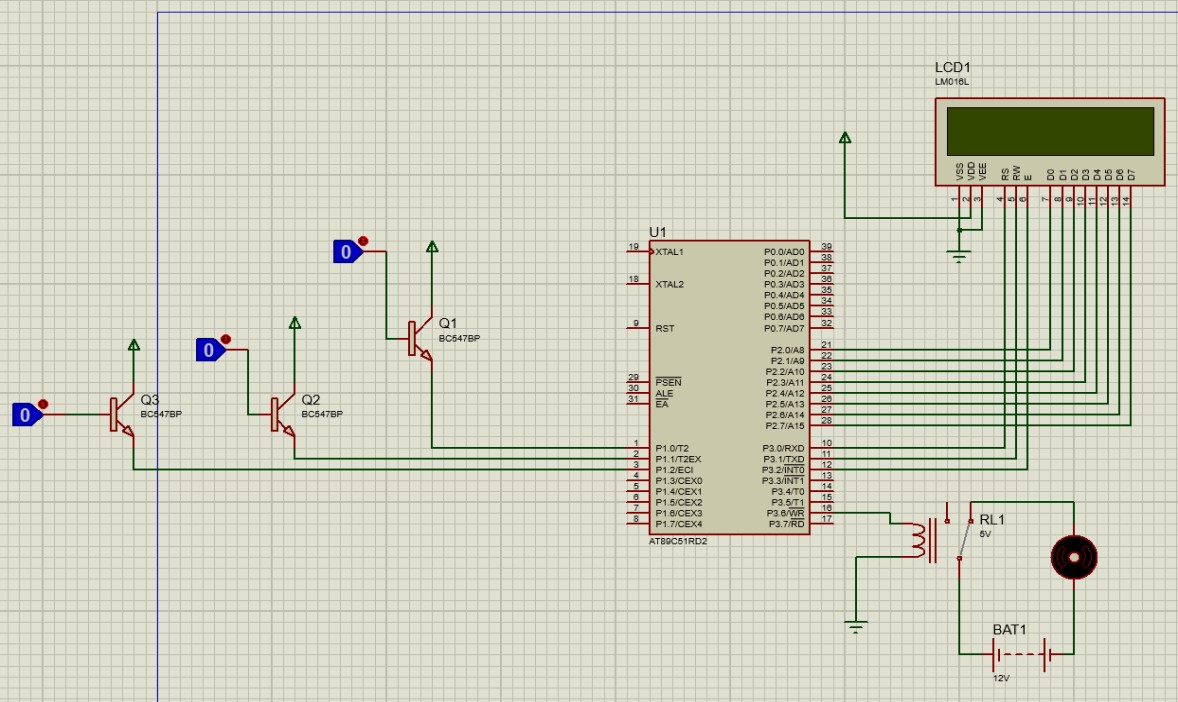
#### 3 Logic Gates (LOGICTOGGLE)

* **1 Relay (RELAY)**

#### 4 Power Supplies

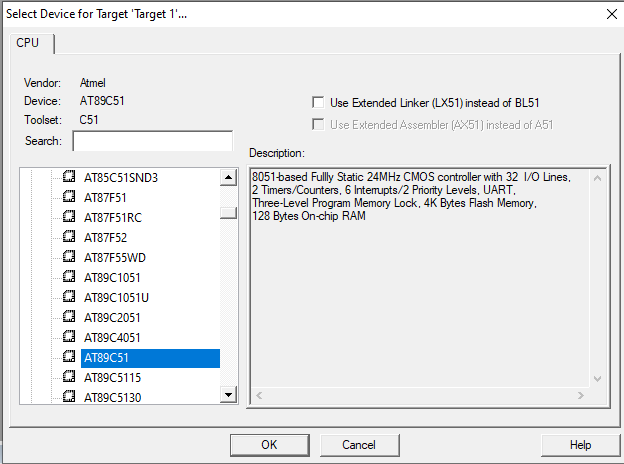
* **2 Ground Units.**

#### Step 1: Configure your system based on the following circuit diagram.

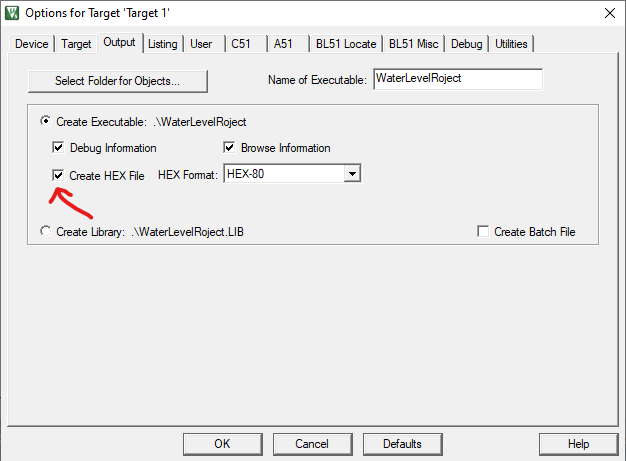
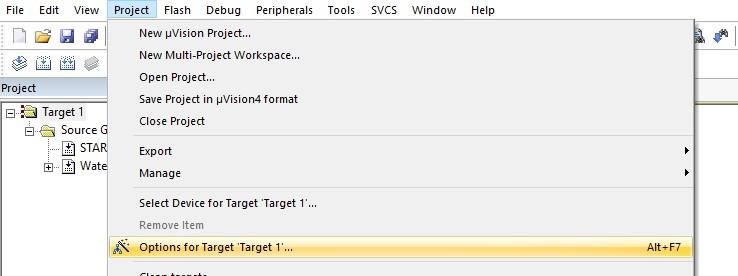


Note: Make sure that you use the correct devices as mentioned in the requirements.

#### Step 2: Create a new project in Kiel uVision5. Select the Amtel (AT89C51) board and create the project.



**Step 3: Under Project>Options for Target>Output> make sure that you select createHex file, otherwise the required hex file will not be generated.**



#### Step 4: Under Files, create a new file for the target, this will contain the code for your board. Write the code in that file and save it as a C file (WaterLevelProject.c). Once that is done click on the build option under the Projects tab. Your output should have 0errors and warnings.

**The code for the microcontroller:**

#include<reg51.h> #define lcdport P2

sbit rs=P3^0; sbit rw=P3^1; sbit en=P3^2;

sbit half=P1^0; sbit full=P1^2; sbit mid=P1^1; sbit motor=P3^6;

void lcdcmd(char); void lcdint();

void lcddata(char); void lcdstring(char \*);

void delay(unsigned int);

void main()

{

P1=0x00;

half=full=mid=0; motor=0;

while(1)

{

lcdint();

if(half==0 && full==0 && mid==0)

{

lcdcmd(0x85); lcdstring("EMPTY");

}

if(half==1 && full==0 && mid==0)

{

lcdcmd(0x85); lcdstring("LOW");

}

if(half==1 && full==0 && mid==1)

{

lcdcmd(0x85); lcdstring("HALF");

}

if(half==1 && full==1 && mid==1)

{

motor=1; lcdcmd(0x85); lcdstring("FULL!!"); lcdcmd(0xc3);

lcdstring("MOTOR OFF"); motor=0;

}

}

}

void delay(unsigned int x)

{

unsigned int i;

for(i=0;i<x;i++);

}

void lcdint()

{

lcdcmd(0x38); delay(500); lcdcmd(0x01); delay(500); lcdcmd(0x0c); delay(500); lcdcmd(0x80); delay(500); lcdcmd(0x0e); delay(500);

}

void lcdcmd(char value)

{

lcdport = value; rw=0;

rs=0; en=1; delay(500); en=0;

}

void lcdstring(char \*p)

{

while(\*p!='\0')

{

lcddata(\*p); delay(10000); p++;

}

}

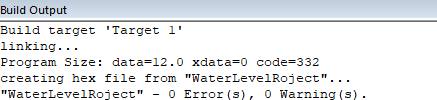
void lcddata(char value)

{

lcdport = value; rs=1;

rw=0; en=1; delay(500); en=0;

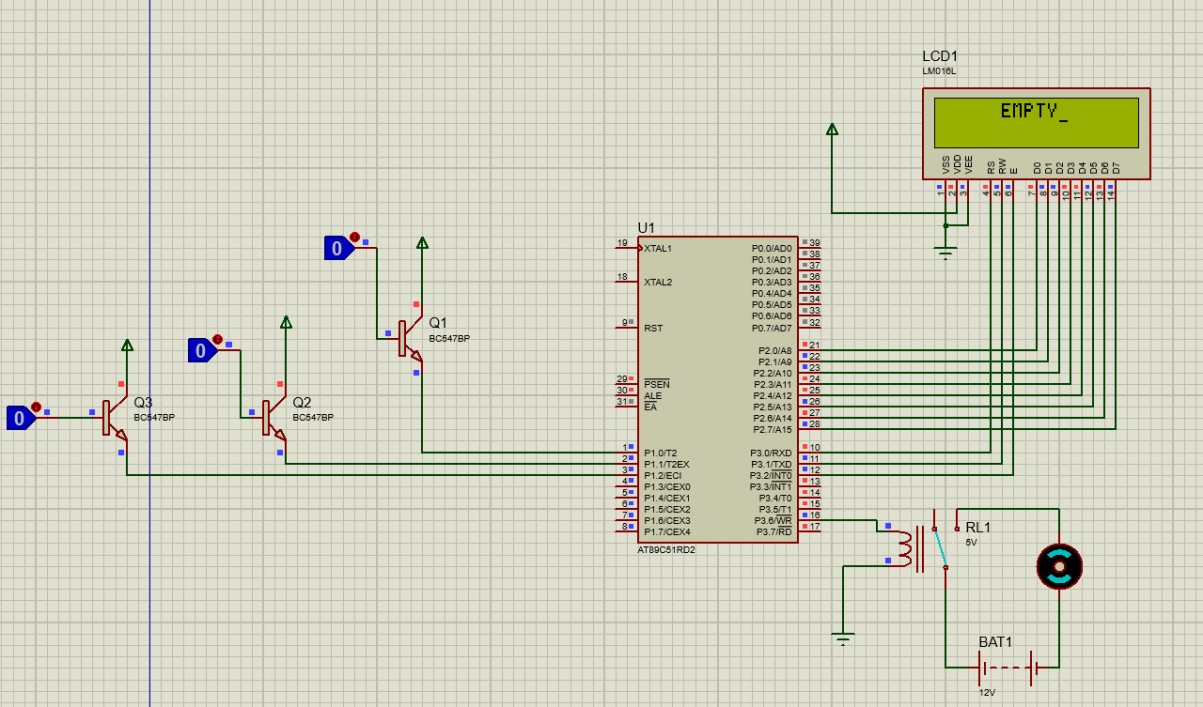
}



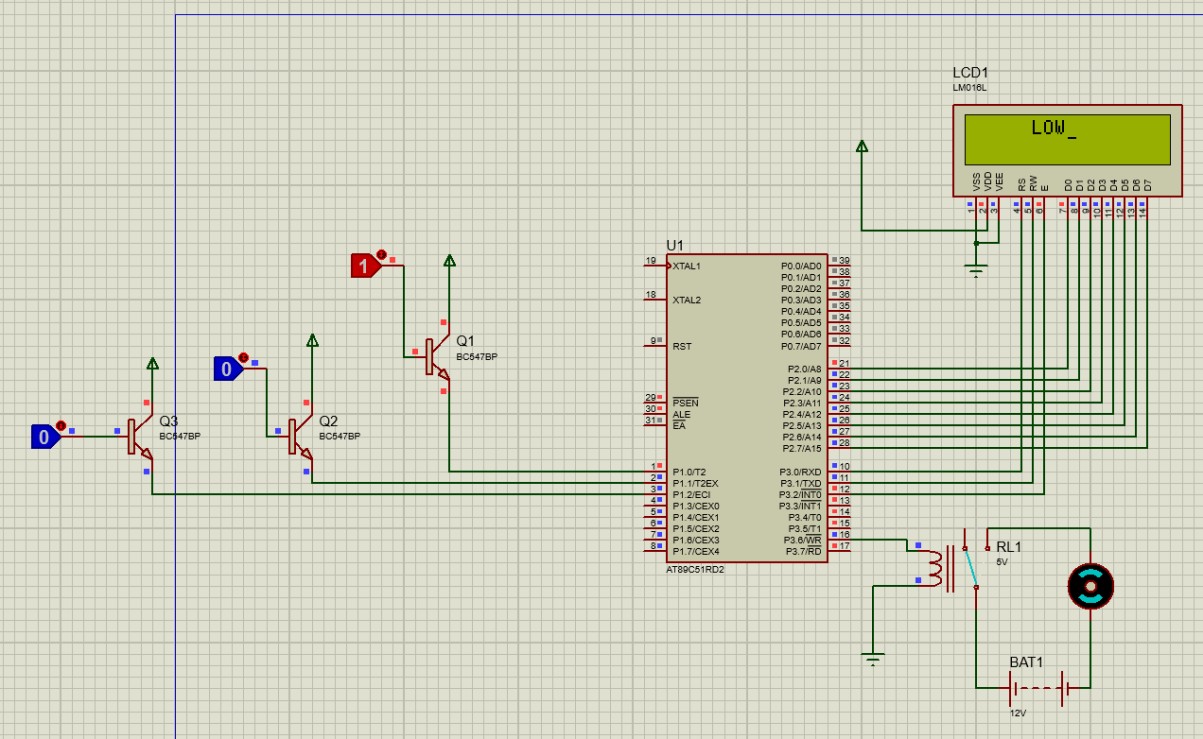
#### Step 5: Switch back to Proteus and double click on the MCU. Over here select theprogram file, i.e. the hex file, and click on ‘OK’.

**Step 6: If you have followed the steps correctly and configured your circuit based on the diagram, then you can run the simulation.**

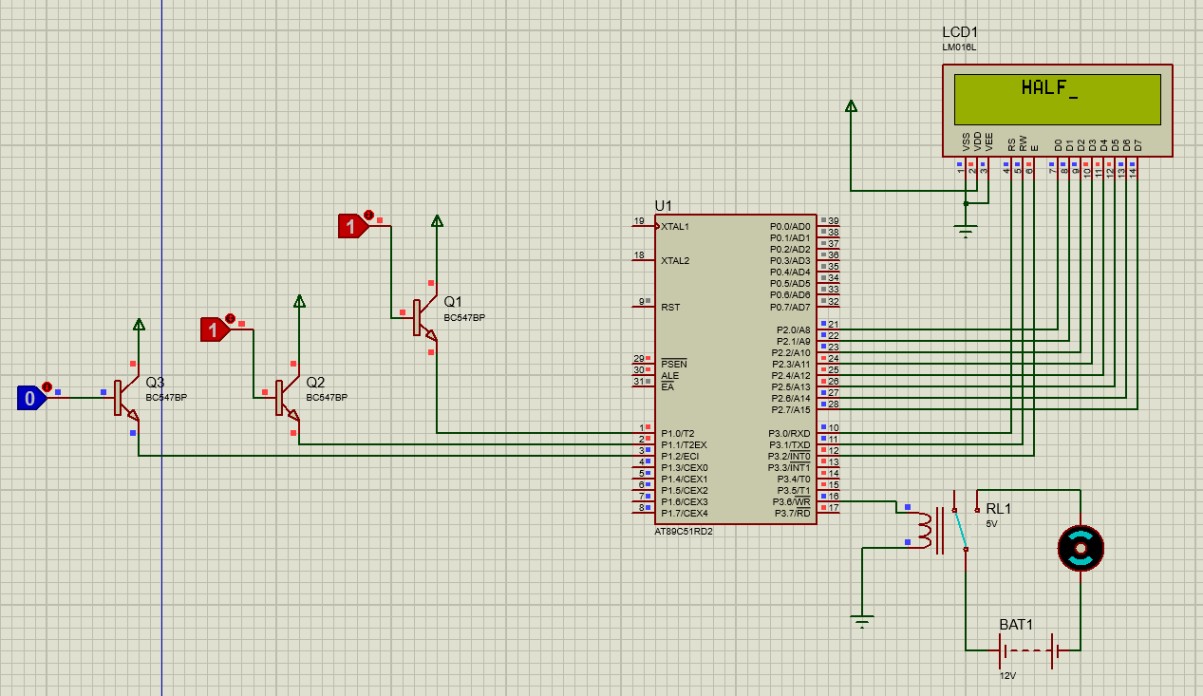
#### Step 7: When you run the simulation, you will notice that the LCD screen says that thewater tank is empty.



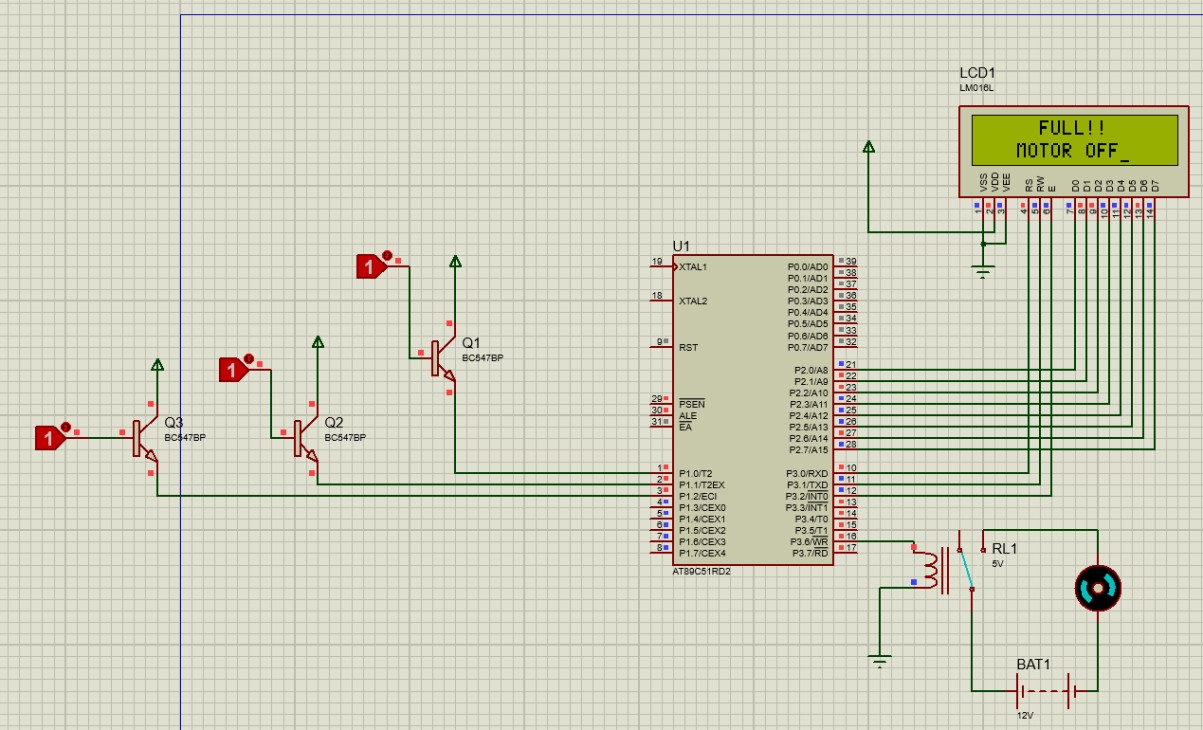
**Step 8: Use the toggle gate to simulate the filling of the tank. The first toggle gate will fill the tank to the “LOW” level.**



#### The second toggle gate will set the tank to the ‘HALF’ level.



**The third toggle gate will set the tank to the ‘FULL’ level. Here the motor will be turnedto the off state.**



Note: If the third toggle gate is toggled, the motor will be turned on again. The same applies for the first and the second toggle gate.

# Assignment No:-05

### **Assignment Name:-** Develop an IoT application that will raise an alarm whenever with going to rainoutside based on the weather prediction data.

**Name: Nishikant Pawar** **Roll No:-**

**Class:-** Msc-1(CS) **Date:-**

#### Requirements:

* **1 Rocker Switch**

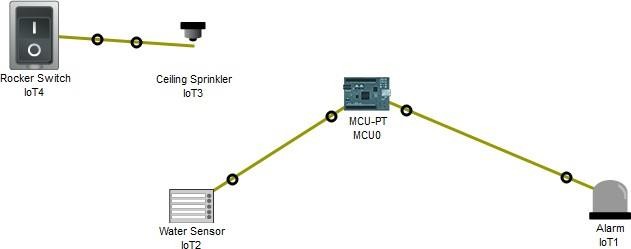
#### 1 Ceiling Sprinkler

* **1 Water Sensor**

#### 1 MCU Board

* **1 Alarm**

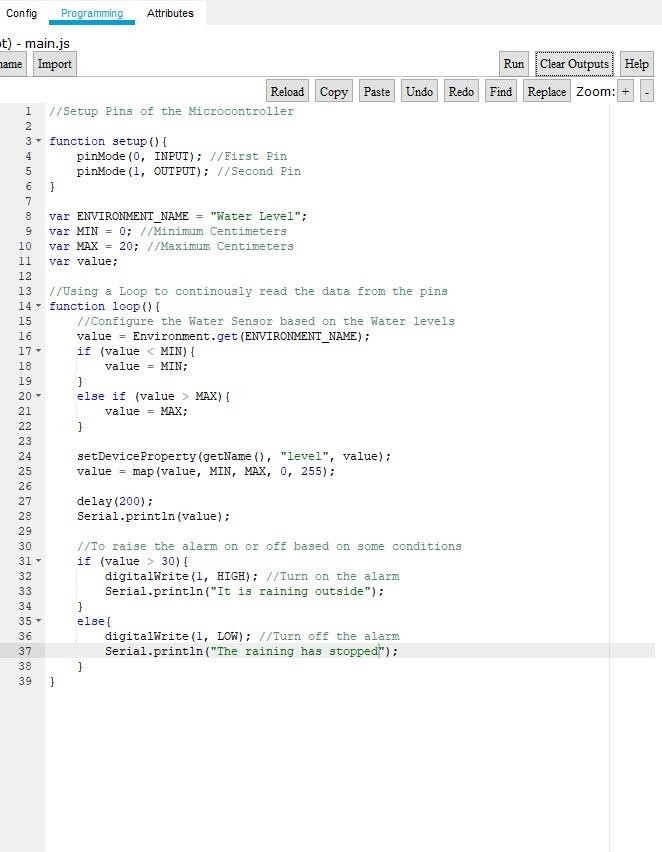
#### Step 1: This practical will take place in Cisco Packet Tracer. Now make the following configuration.



Note: All the components are connected via the ‘IoT Custom Cable

#### Step 2: Configure the MCU.

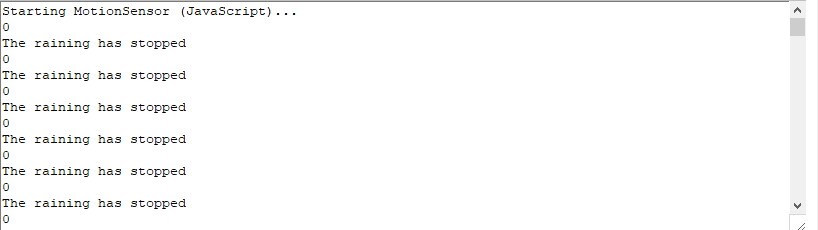
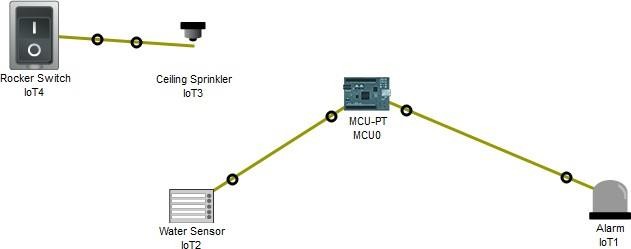
Under programming create a new project and write the following code.



#### Step 3: Run the code and note down your observations.

Press left alt on your keyboard followed by a mouse click on the rocker switch to turn on the sprinkler.

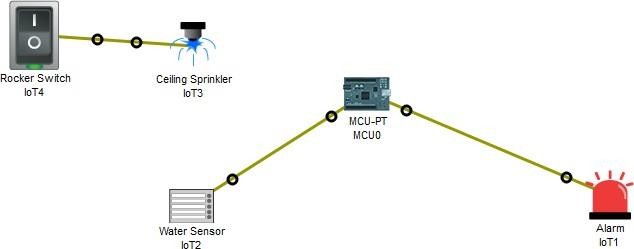
When the sprinkler is in the ‘off’ state, the water sensor does not detect any rain and continuously provides the output ‘0, The raining has stopped’ in the terminal and the alarm is in its off state.



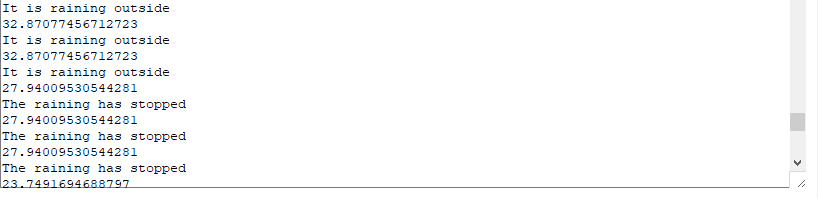
When the sprinkler has been turned to the ‘on’ state, the water sensor starts to detect the rain, but it will only provide the output ‘It is raining outside’, when the water level is above 30.



As the water sensor detects that the water level is above 30, it also provides a signal to the alarm which is switched to its ‘on’ state.



The water sensor will keep sending the output ‘It is raining outside’ till the water level drops below 30. Then it will turn off the alarm and will produce the output ‘The raining has stopped’.



# Assignment No:-06

### **Assignment Name:-** Develop a IoT application which will record the movement and orientation of your phone and give the data back to the PC.

**Name: Nishikant Pawar** **Roll No:-**

**Class:-** Msc-1(CS) **Date:-**

#### Requirements:

* **1 Motion Sensor (IoT1)**

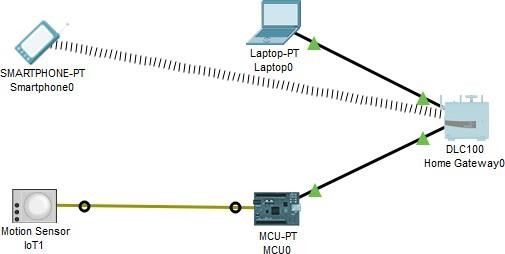
#### 1 Microcontroller Unit (MCU-PT)

* **1 Home Gateway (DLC100)**

#### 1 Laptop (Laptop-PT)

* **1 Smartphone (SMARTPHONE-PT)**

#### Step 1: Select the requirements and make their configurations.



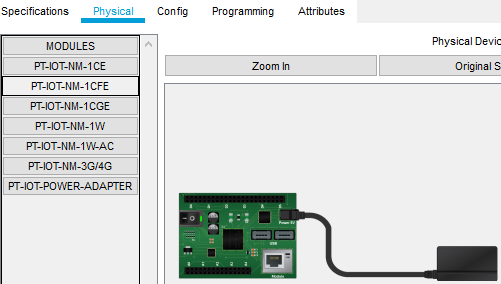
* **Motion Sensor:**

Connect the Motion Sensor to the MCU-PT via the IOT Custom Cable.

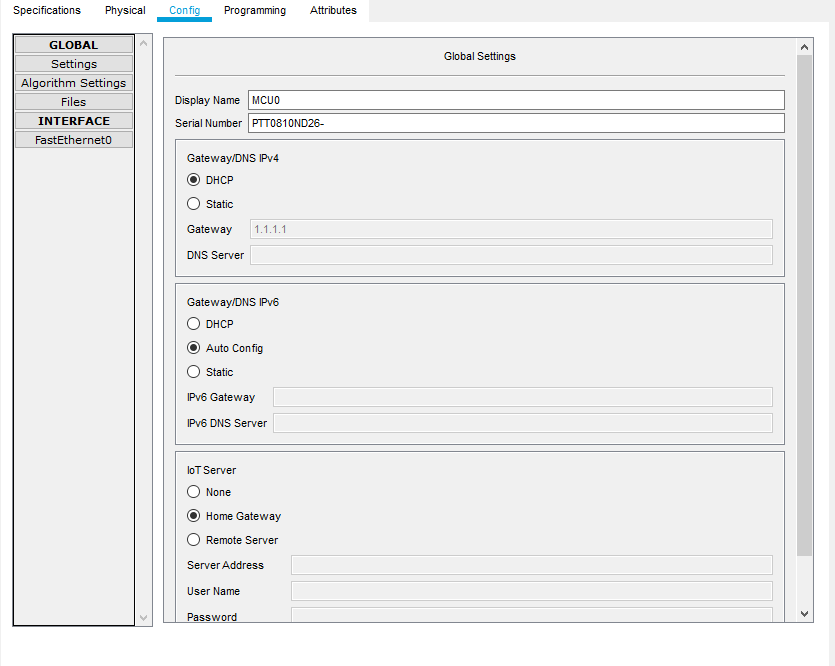
To use the motion sensor press ‘alt’ followed by the hovering of your cursor over the sensor.

#### MCU-PT:

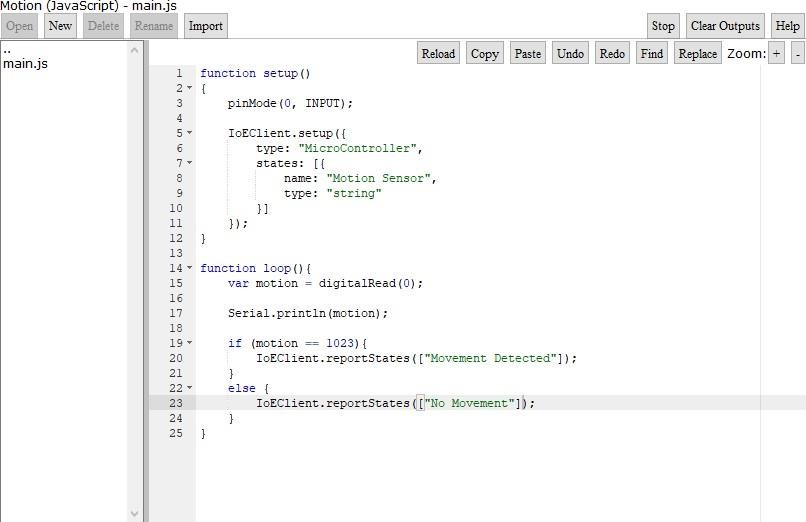
Add the FastEthernet Port to the microcontroller.



Connect the MCU-PT to the Home Gateway using a copper wire and configure its connection settings.

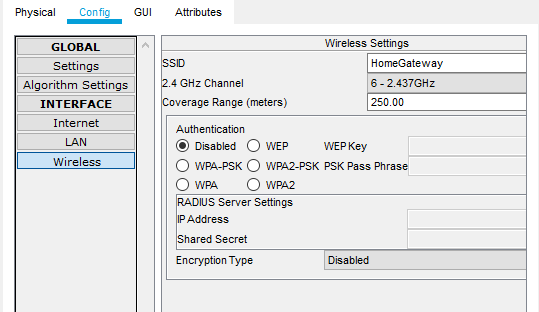
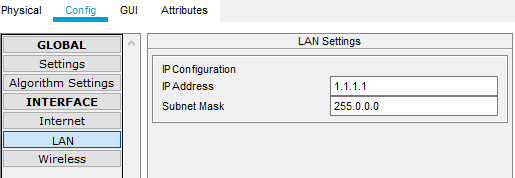


Add the code required for it to process the analog signals of the motion sensor so that it can send it to the Home Gateway.



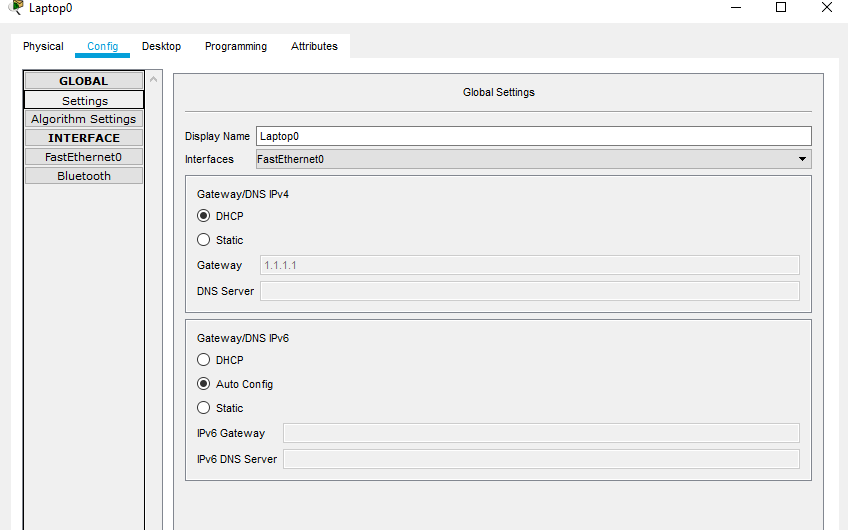
#### Home Gateway:

Make the following configurations to the Home Gateway and connect it to the laptop via a copper cable.

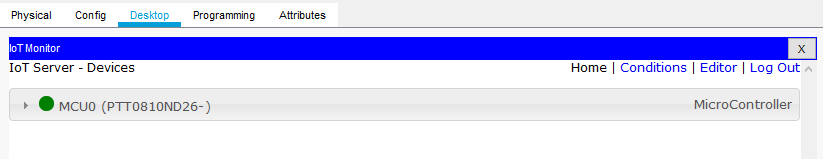
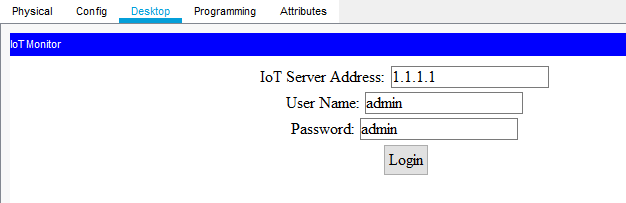


#### Laptop:

Make the following configuration to the laptop.

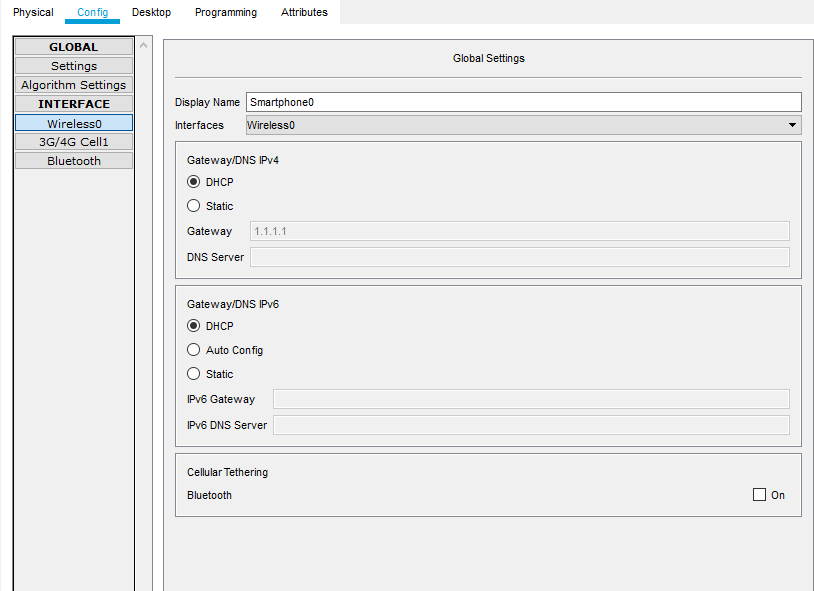


Via the Desktop Interface on the Laptop, open the IoT monitor and login via the Home Gateway Address.

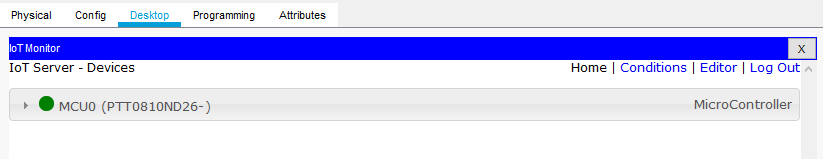
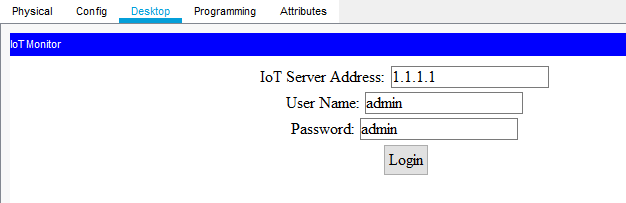


#### Smartphone:

Make the following configuration to the smartphone.

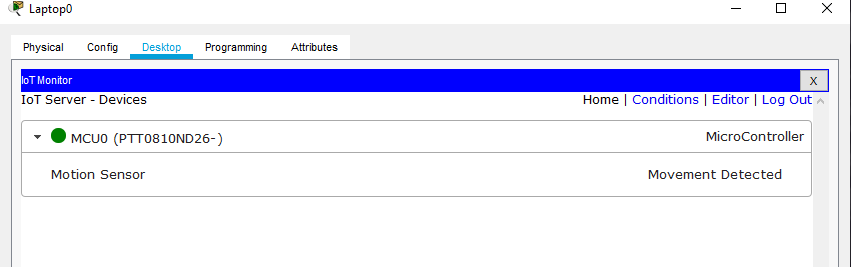
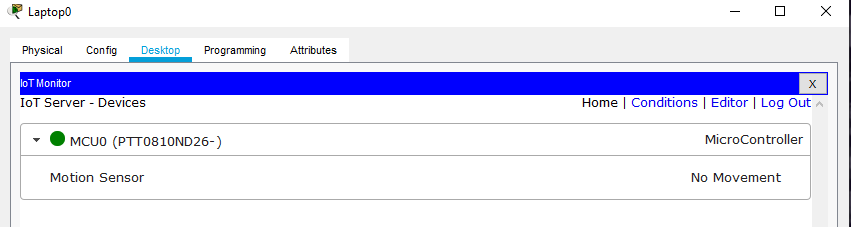


Via the Desktop Interface on the Laptop, open the IoT monitor and login via the Home Gateway Address.

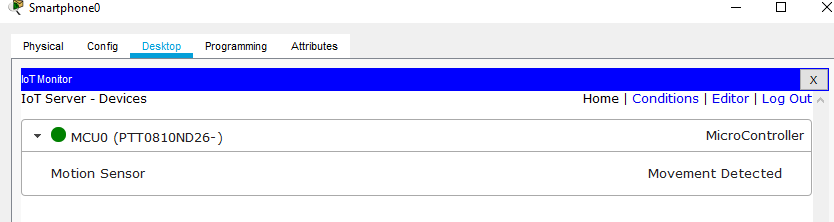
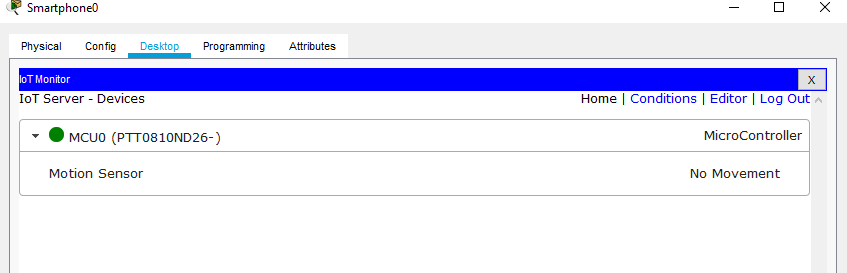


#### Step 2: Check whether the motion sensor works by observing the IoT monitor on the Laptop and the Smartphone.

* **Laptop:**



#### Smartphone:



Note: On the console of the Microcontroller, when there is no movement the outputshown is ‘0’ and when there is movement the output shown is ‘1

# Assignment No:-07

**Assignment Name:-** Develop an IoT module to which measure the intensity of light.

**Name :**  **Nishikant Pawar** **Roll No:-**

**Class:-** Msc-1(CS) **Date:-**

#### Requirements:

* **1 Power Supply**

#### 2 Grounds

* **1 Arduino UNO**

#### 1 LCD Screen (LM016L)

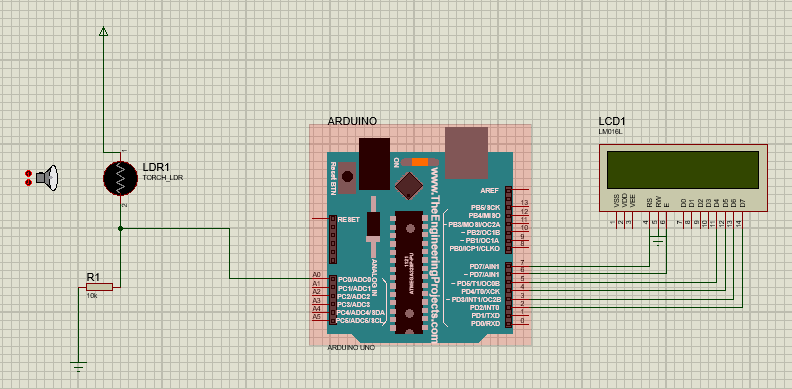
* **1 Torch and Light Dependent Resistor (LDR)**

#### 1 Resistor 1kΩ

**Step1 : Install Arduino Libraries for Proteus to use (**[https://www.theengineeringprojects.com/2015/12/arduino-library-proteus-](https://www.theengineeringprojects.com/2015/12/arduino-library-proteus-simulation.html) [simulation.html](https://www.theengineeringprojects.com/2015/12/arduino-library-proteus-simulation.html)**)**

Extract the files in to the Library Folder of Proteus.

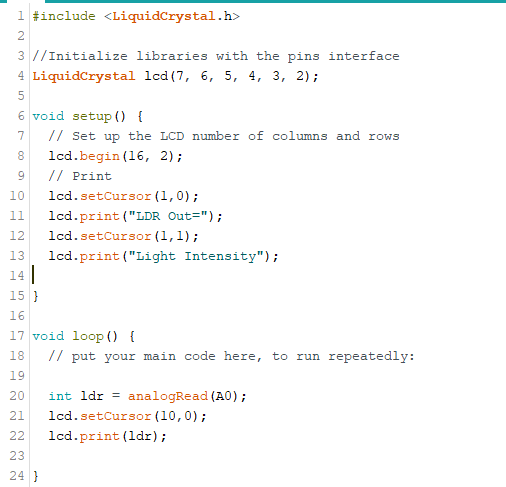
**Step 2: Install Arduino IDE. (**https:/[/www.arduino.c](http://www.arduino.cc/en/software))c[/en/software**)**](http://www.arduino.cc/en/software)) **Step 3: Configure the system.**



Add the devices that are required and make the connections depicted above.

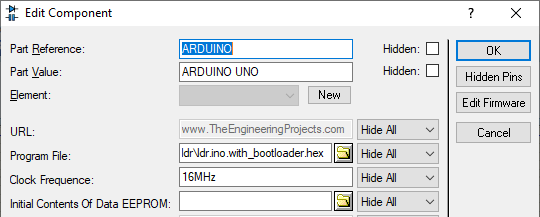
#### Step 4: Write the code for the Arduino Microcontroller using the Arduino IDE.

Code:



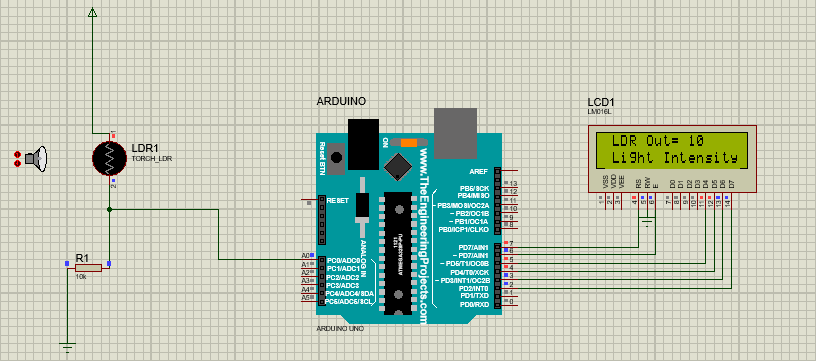
Note: Make sure that you compile the code. The generated hex file for proteus will be generated in the location (C:\Users\rudra\AppData\Local\Temp\arduino\_build\_333966\ldr.ino.with\_bootloader.hex).

Upload this code to the Arduino UNO in Protes by double clicking on the MCU and adding the path of the hex file to the program file.

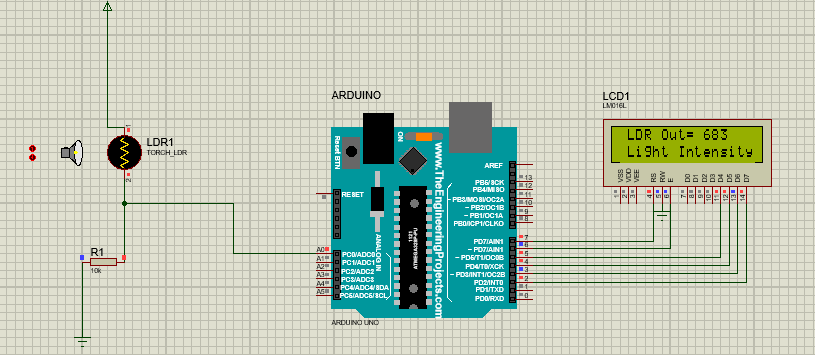


#### Step 5: Run the simulation and make your observations.

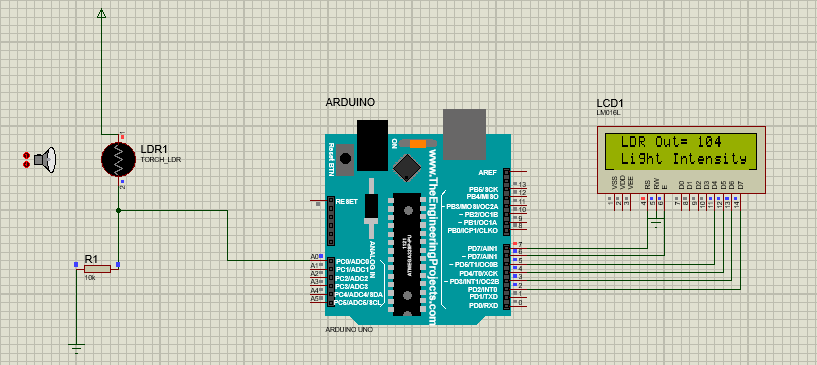
When we run the simulation there is a fixed intensity of light measure on the LCD screen.



When we bring the torch closer to the LDR, the intensity increases.



And as we take the torch away from the LDR the intensity decreases.



Thus we have measured the intensity of light by developing an IoT module.