Name: Avinash Rajkumar Kauran, Roll No: CS22006
Course: Embedded Internet Of Things

PRACTICAL 1

Aim: Design and implement basics embedded circuits

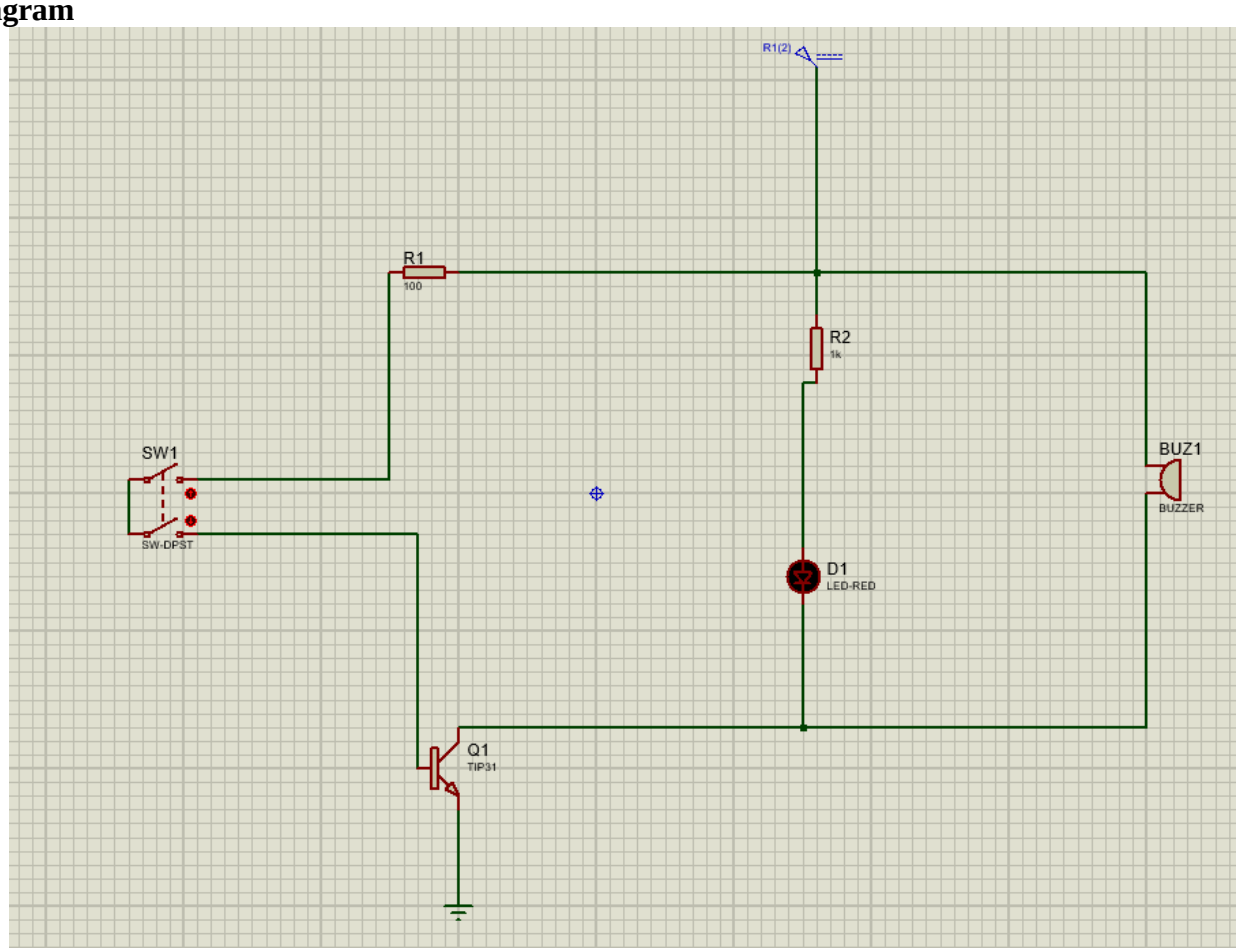
a) Automatic Alarm system – Alarm should get tigger by senor
b) Timer based buzzer
c) Sensor based counting device

a) Automatic Alarm system – Alarm should get tigger by senor

Requirements:

- 1 Led-Red
- 1 Transistor(TIP31)
- 2 Resistor
- 1 Buzzer
- 1 Power source(DC12.5v)
- Ground
- Sw-dpst

Diagram

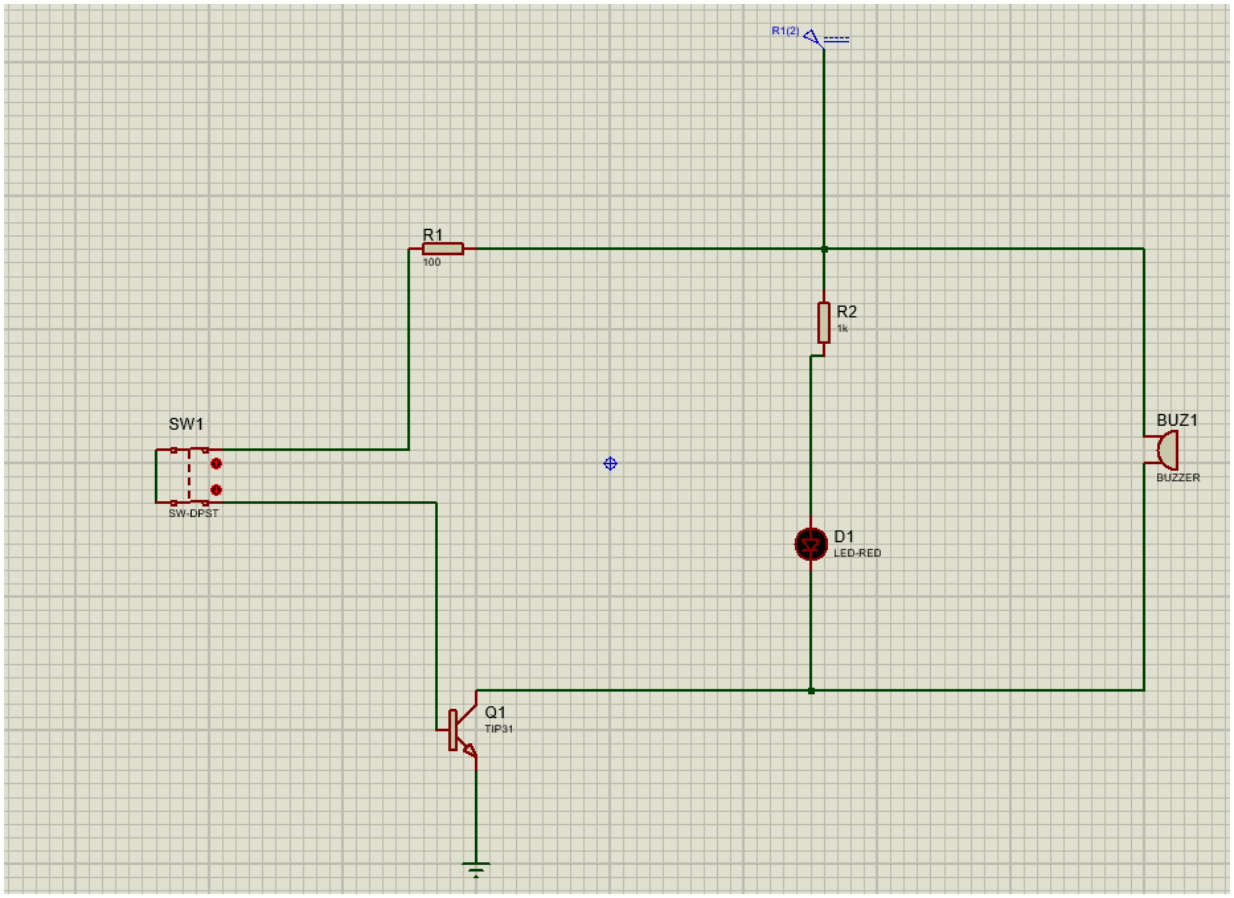


Voltage of DC power

Name: Avinash Rajkumar Kauran, Roll No: CS22006
Course: Embedded Internet Of Things

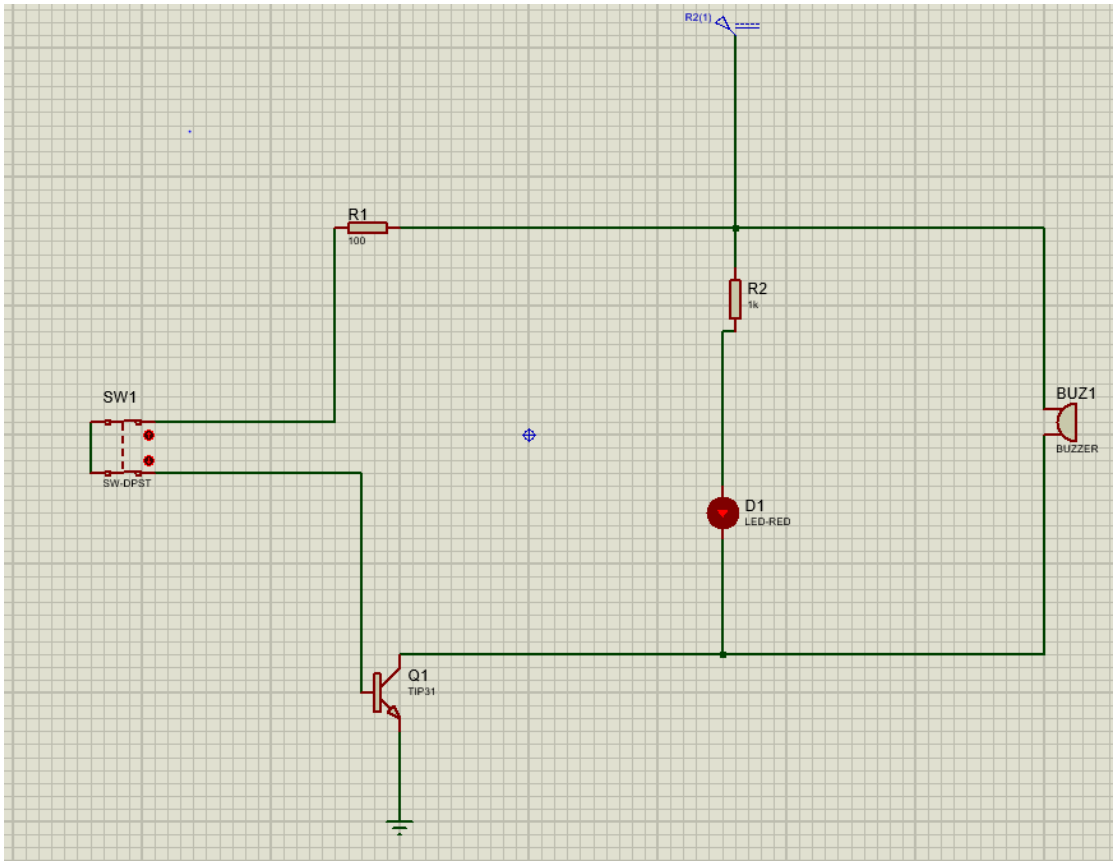
OUTPUT:

When the sensor is in the off state :



When the sensor is in the on state

Name: Avinash Rajkumar Kauran, Roll No: CS22006
Course: Embedded Internet Of Things

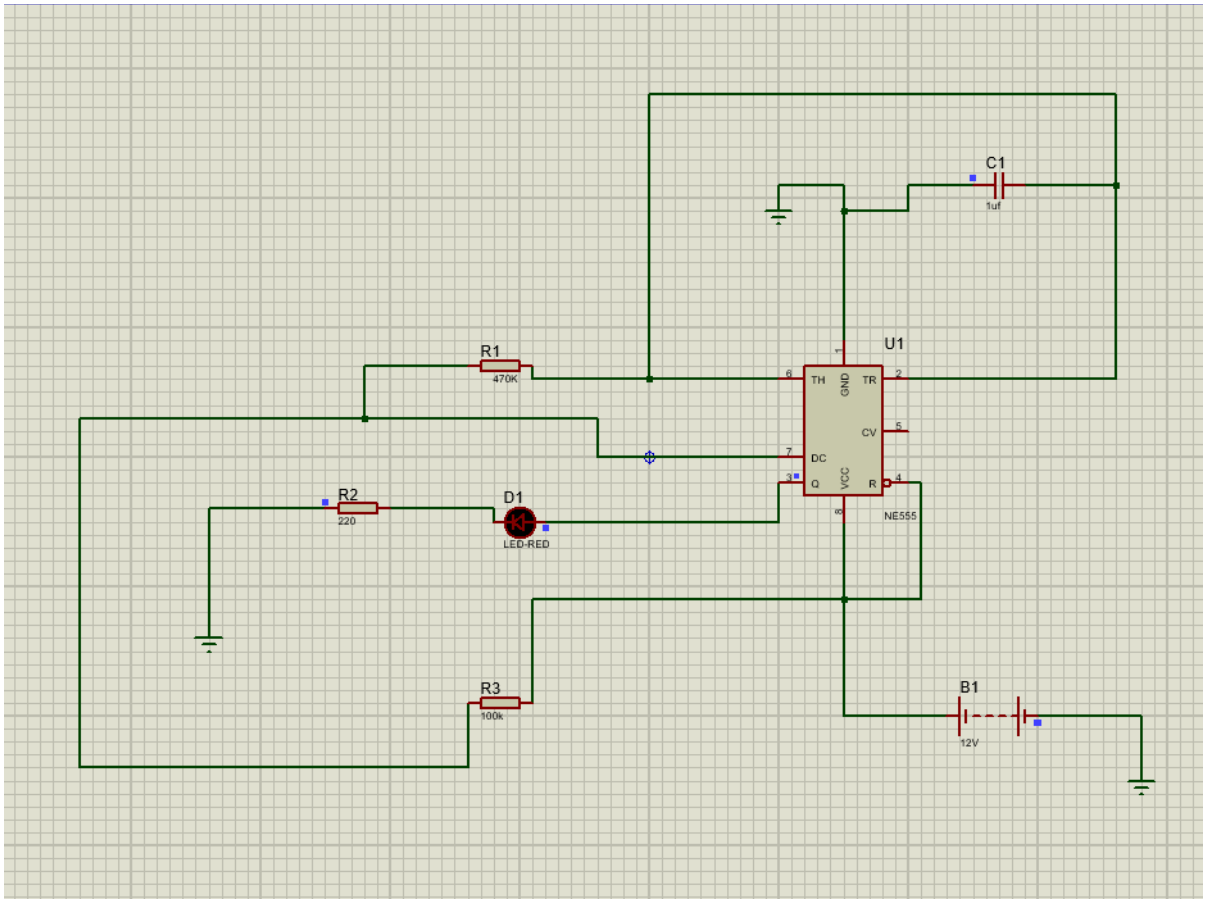


Name: Avinash Rajkumar Kauran, Roll No: CS22006
Course: Embedded Internet Of Things

b) Timer based buzzer Device used:

- 1 Battery 12v
- 1 Led-Red
- 3 Resistor
- 3 Ground
- 1 Capacitor
- 1 PCB (NE555)

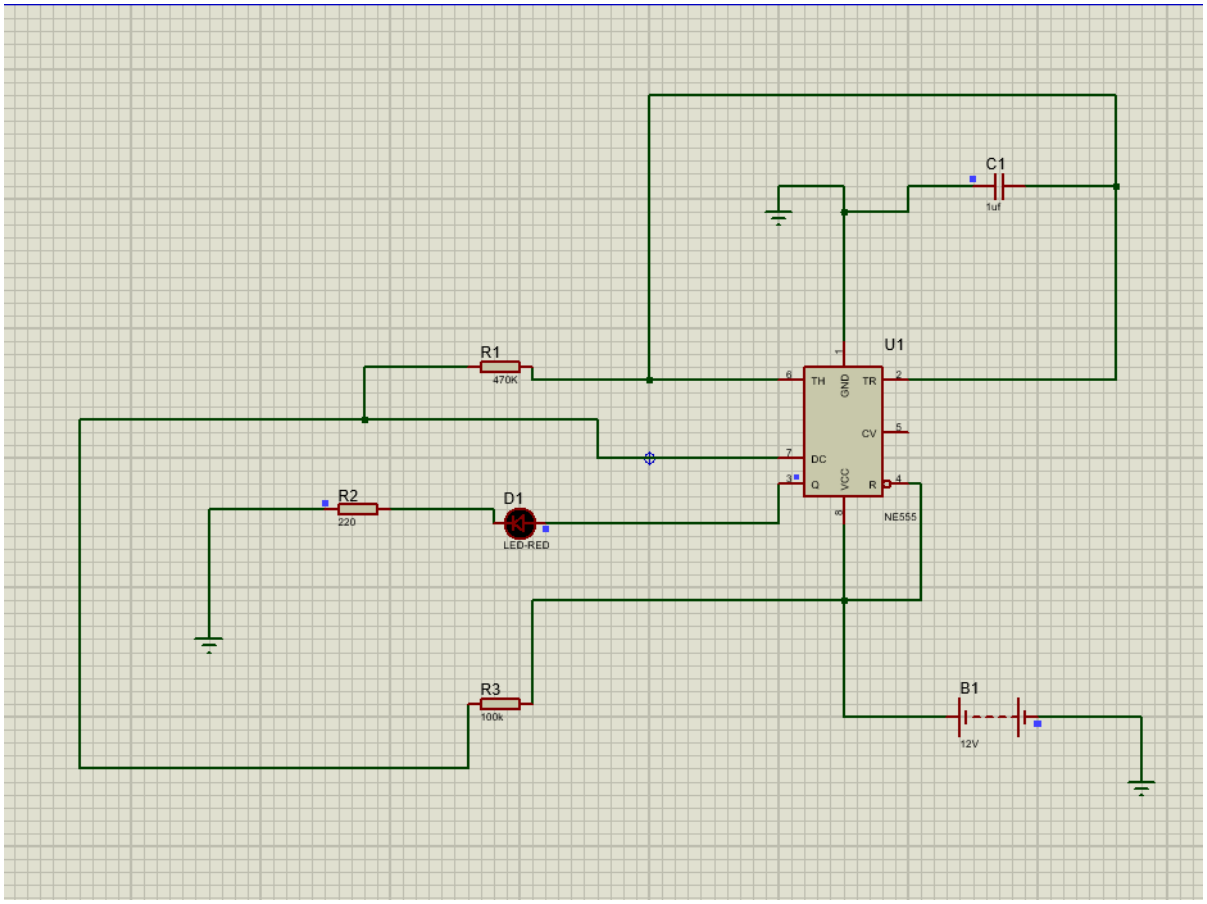
Diagram



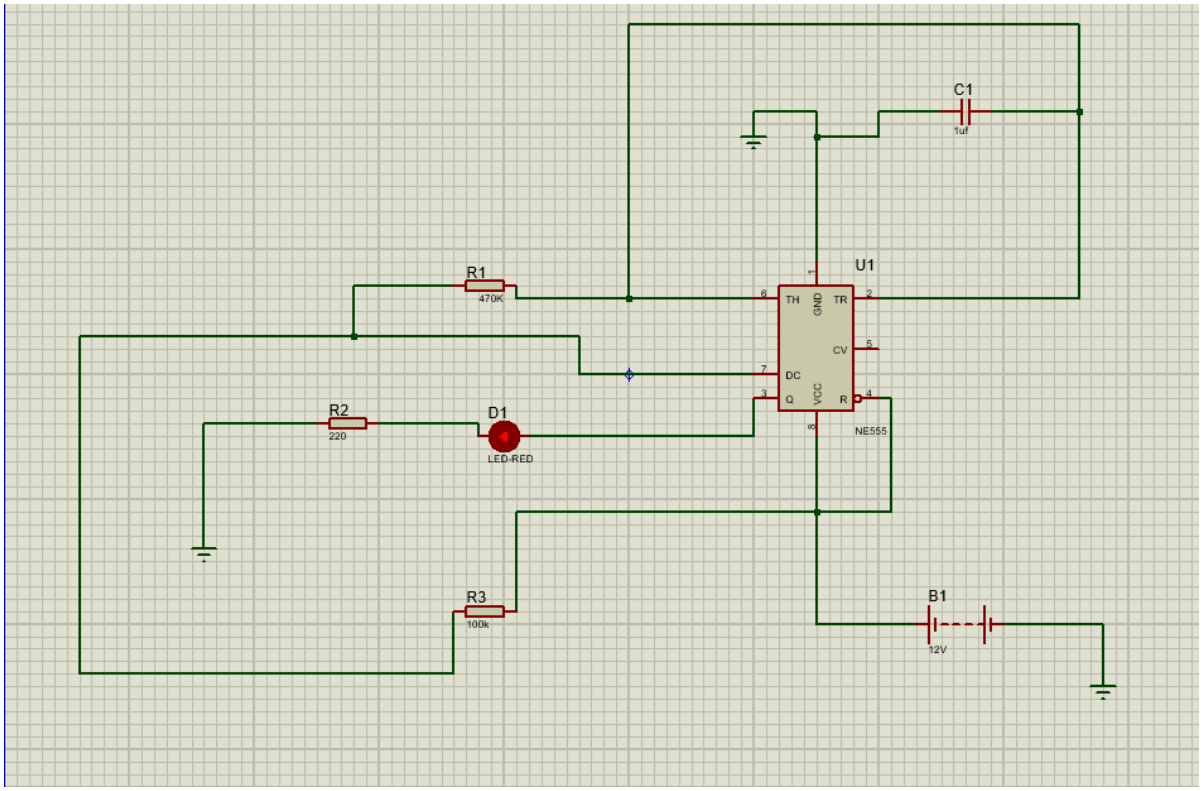
Name: Avinash Rajkumar Kauran, Roll No: CS22006
Course: Embedded Internet Of Things

OUTPUT:

The simulation starts the Red LED will keep blinking based on the configuration made to the PCB (NE555).



Name: Avinash Rajkumar Kauran, Roll No: CS22006
Course: Embedded Internet Of Things

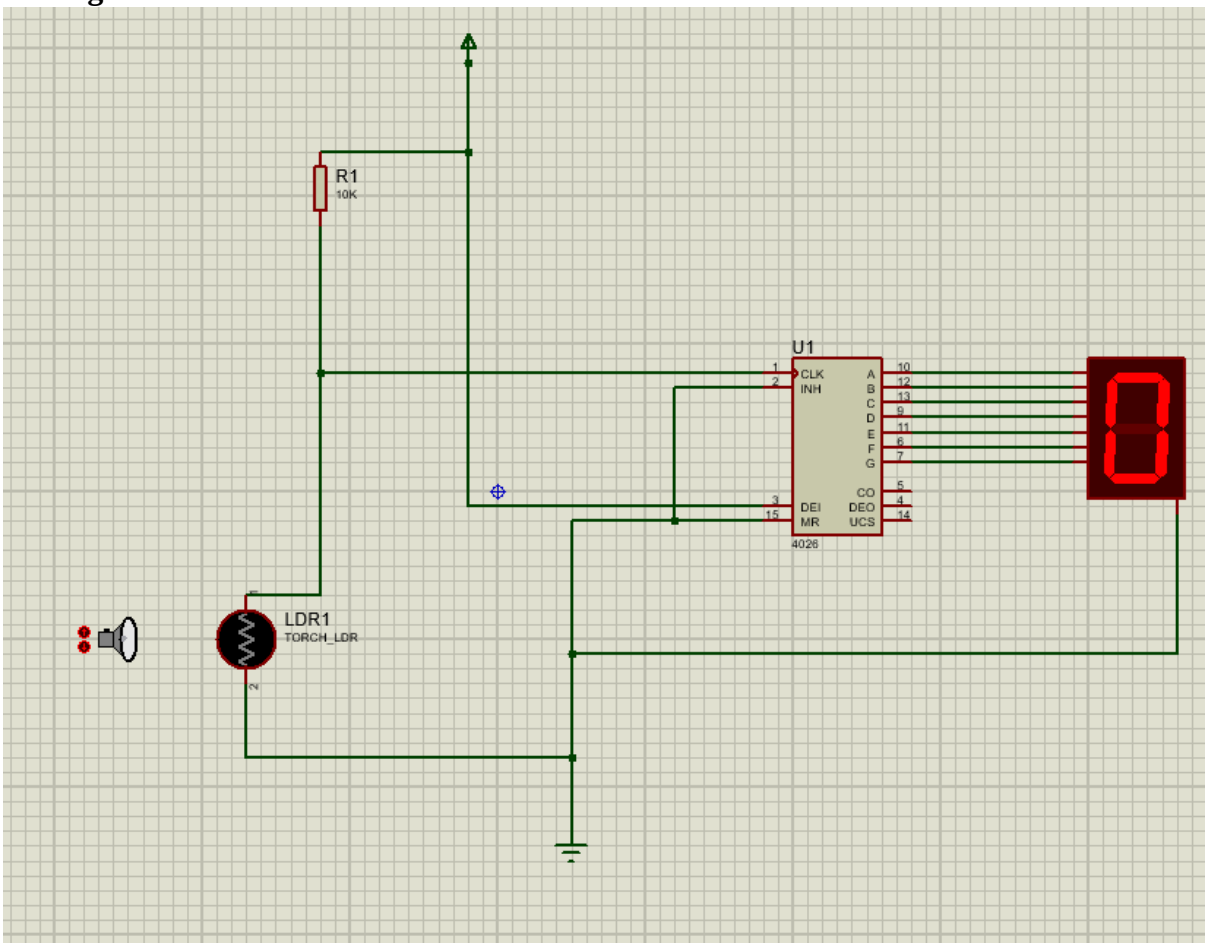


c) Sensor based counting

device Device used:

- 1 led screen (7SEG-COM-CATHODE)
- 1 PCB (4026)
- 1 Ressistor(MINRES10K)
- 1 TORCH_LDR

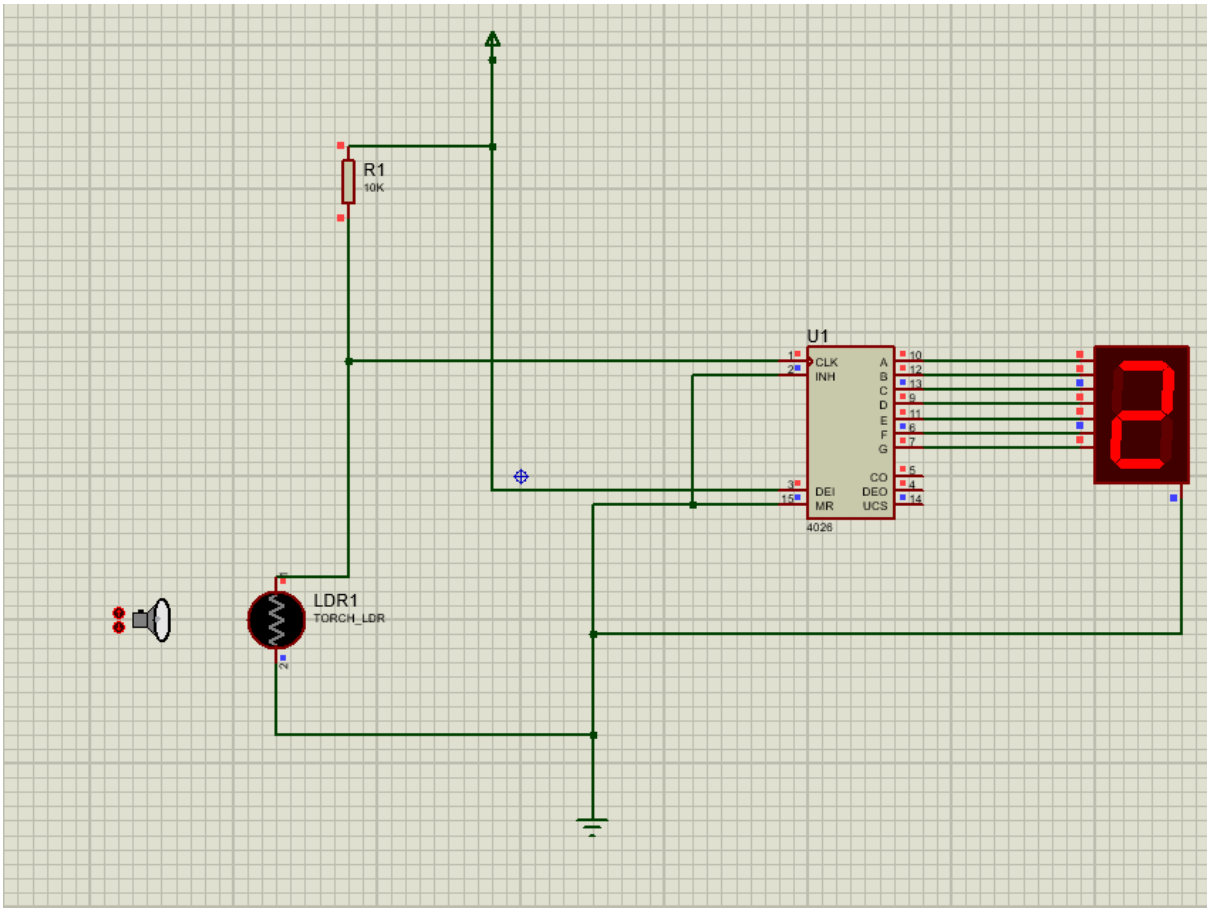
Diagram



Name: Avinash Rajkumar Kauran, Roll No: CS22006
Course: Embedded Internet Of Things

OUTPUT:

The simulation. Press the '+' and '-' on the TORCH_LDR component to change the torch distance from the LDR.



Name: Avinash Rajkumar Kauran, Roll No: CS22006
Course: Embedded Internet Of Things

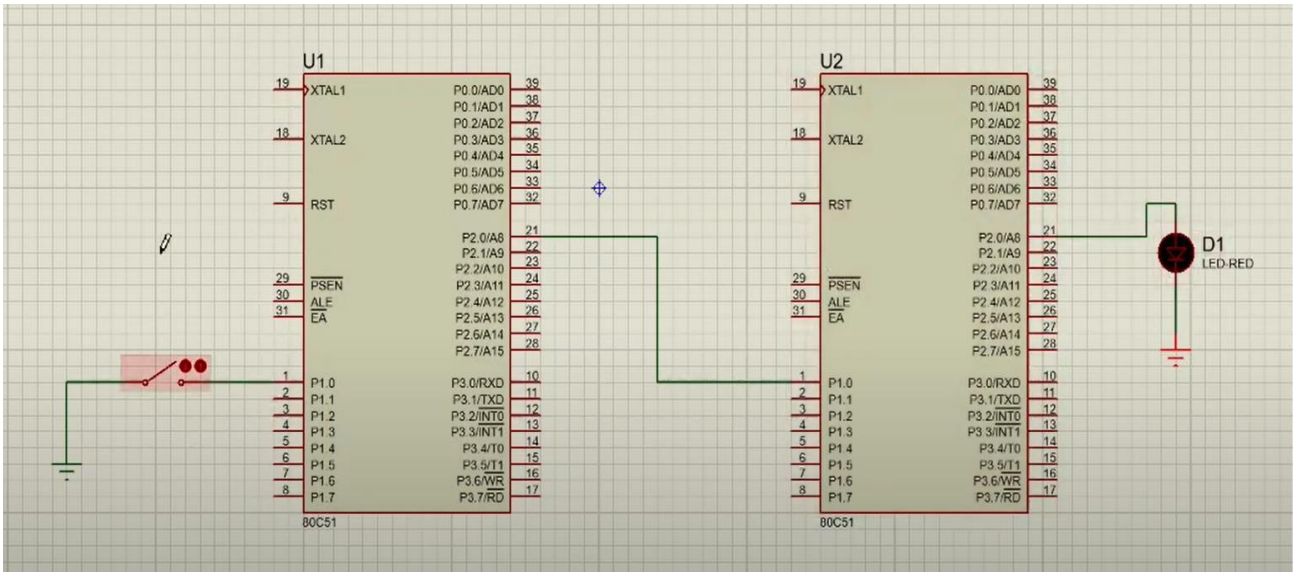
PRACTICAL 2

AIM: Demonstrate communication between two embedded devices using UART port

CONCEPT: The transmitting microcontroller (on the left) will transmit a signal to the receiving microcontroller (on the right) from P3.1/TXD to P3.0/RXD .

When signal is passed from transmitter to receiver LED will be turned on COMPONENTS USED:

DIAGRAM:



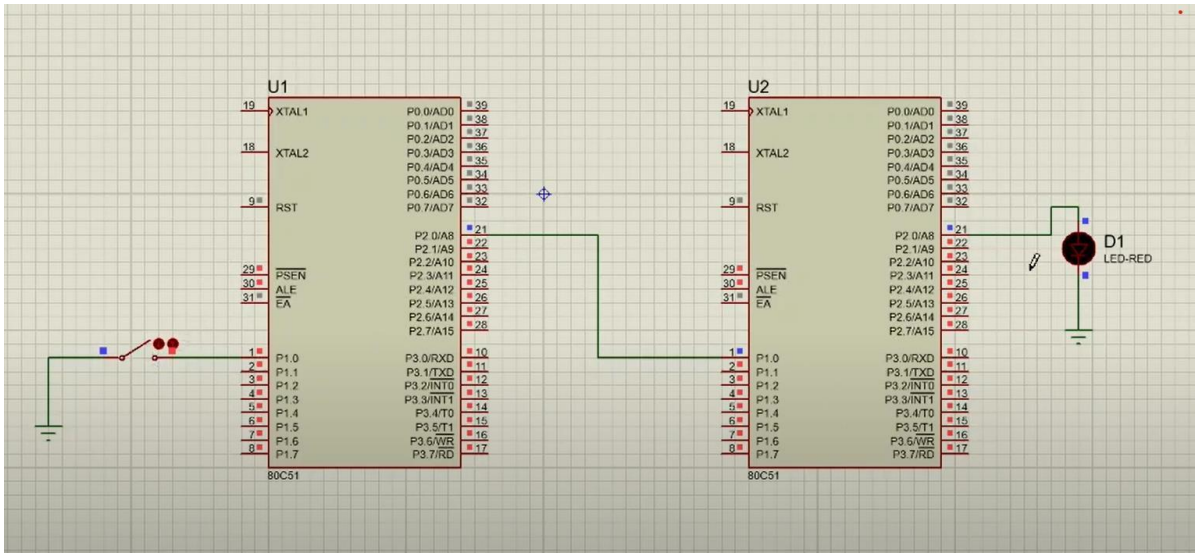

```
Code for
transmitter
#include<reg51
.h> sbit
sw=P1^0;
sbit outPin=P2^0;

void main(void)
{
while(1)
{
if(sw==0)
{

outPin=1;
}
else
{
outPin=0;
```

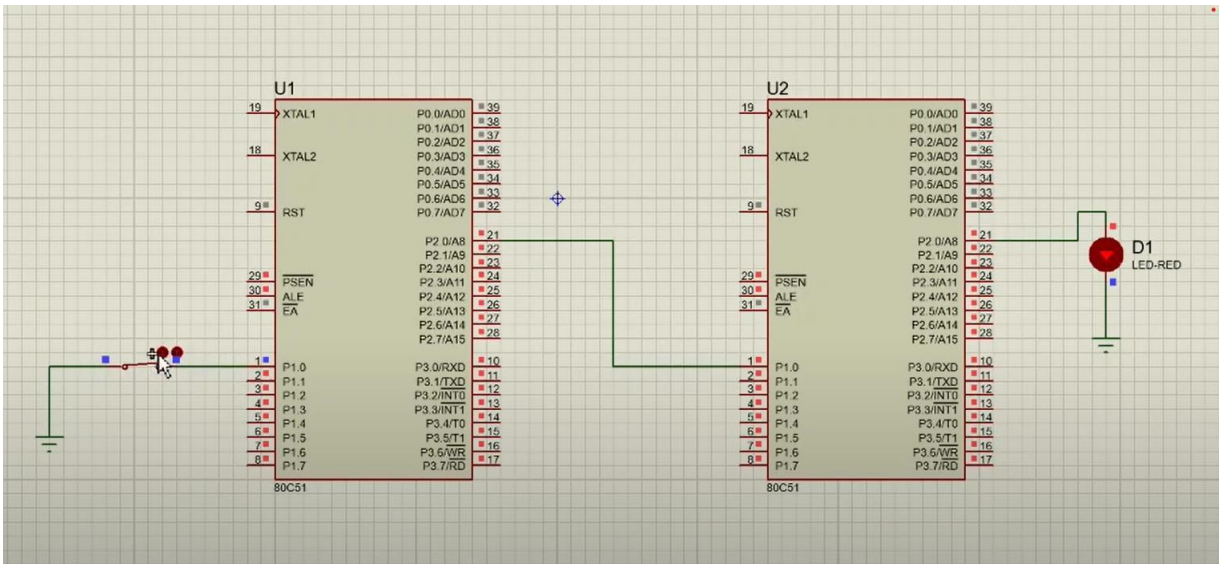
```
}



}

}


CODEFOR RECEVIRE
```

```
#include<r
eg51.h>
sbit
Inpin=P1^0
; sbit
led=P2^0;

void main(void)
{
while(1)
{
if(Inpin==1)
{

led=1;// led on
}
else
{
led=0;//led off
}




}
```

OUTUP:

Name: Avinash Rajkumar Kauran, Roll No: CS22006
Course: Embedded Internet Of Things





Components used:

1. 80c51 microcontroller
2. Led-red
3. LCD screen LM016L
4. Switch

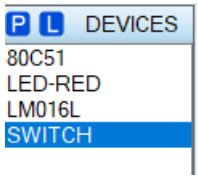
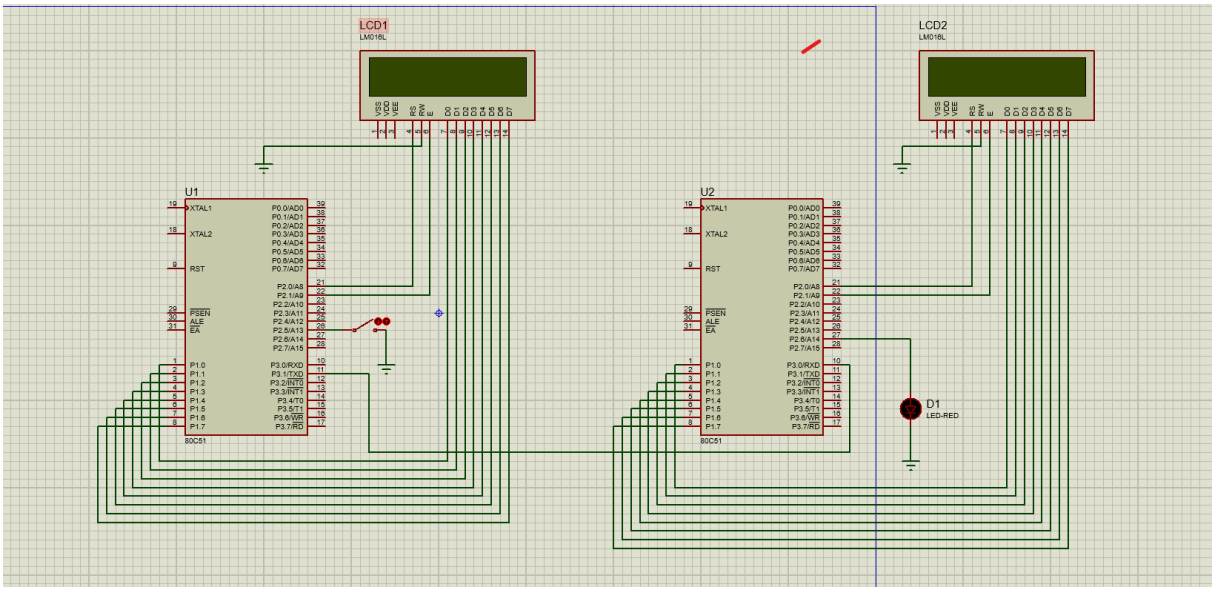


DIAGRAM:

Name: Avinash Rajkumar Kauran, Roll No: CS22006
Course: Embedded Internet Of Things



CONCEPT: The transmitting microcontroller (on the left) will transmit a signal to the receiving microcontroller (on the right) from P3.1/TXD to P3.0/RXD .

When no signal is passed from transmitter to receiver

switch status will be shown (SW not pressed) on the transmitter lcd screen and led status will be shown (LED off) on the receiver lcd screen

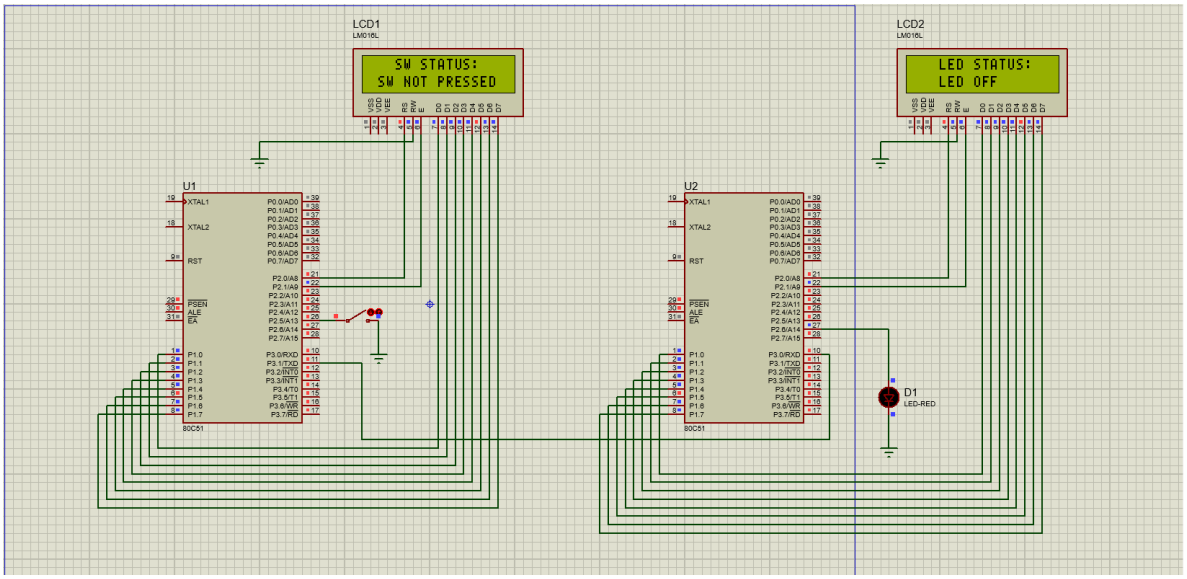
when a signal is passed from transmitter to receiver

switch status will be shown (SW pressed) on the transmitter lcd screen and led status will be shown (LED ON) on the lcd screen

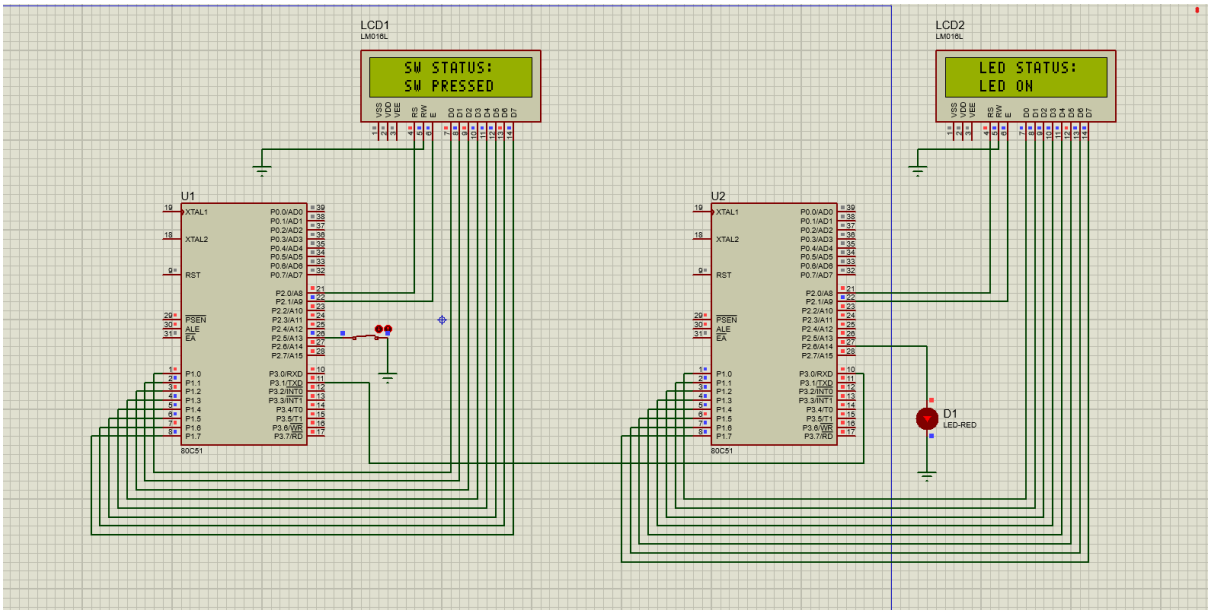
OUTPUT:

CIRCUIT (SWITCH NOT PRESSED AND LED OFF)

Name: Avinash Rajkumar Kauran, Roll No: CS22006
Course: Embedded Internet Of Things



CIRCUIT (SWITCH PRESSED AND LED ON)



CODE :

TRANSMITTE

R

```
//CODE of lcd interfacing with 8051 microcontroller
 #include<reg51.h>

void delay(unsigned int i);
void lcd_cmd(unsigned
char a); void
lcd_data(unsigned char b);
void lcd_init(void);
void lcd_str(unsigned char *str);
```

```
Void sendser_char(unsigned char b);
void sendser_str(unsigned char *str);
sbit rs=P2^0;
sbit
en=P2^1;
sbit
sw=P2^5;
//sbit led=P2^6;
sfr ldata=0x90;//port1
void main()

{
          //led=0;//led off
     TMOD=0x20;//timer1 mode2 -auto reload mode TH1=0xfd;//9600
     baud rate
     SCON=0x50;//8bit data ,1start bit,1stop bit
     TR1=1;
lcd_init();
     lcd_str("WELCOME TO      ");
     lcd_cmd(0xc0);
     lcd_str("MY PROJECT      ");
     delay(65000);
          lcd_cmd(0x01);
          lcd_cmd(0x80);
lcd_str("YT TRAINING      ");
     lcd_cmd(0xc0);
     lcd_str("INSTITUTE      ");
          delay(65000);
          lcd_cmd(0x01);
          lcd_cmd(0x80);
     lcd_str("SW STATUS:  ");
     while(1)
{
     if(sw==0)
     {
     //led=1;//led on
                    lcd_cmd(0xc0
     ); lcd_str("    SW PRESSED
     ");
          //sendser_str("  LED ON  ");
                    sendser_char('A');}
     else
     {
     //  led=0;//led off
                    lcd_cmd(0xc0);
     lcd_str(" SW NOT PRESSED  ");
     //sendser_str("  LED OFF  ");
          sendser_char('B');
     }
```

```
}

void lcd_init()
{
lcd_cmd(0x38);
     lcd_cmd(0x0c);
     lcd_cmd(0x01);
     lcd_cmd(0x80);


}
void delay(unsigned int i)
{
unsigned int j;
     for(j=0;j<i;j++);
}

void lcd_cmd(unsigned char a)
{
  rs=0;//c
     md
     ldata=
     a;
     en=1;
     delay(
     5);
     en=0;
     delay(
     5);
}
void lcd_data(unsigned char b)
{
  rs=1;//d
     ata
     ldata=
     b;
     en=1;
     delay(
     5);
     en=0;
     delay(
     5);
}
void lcd_str(unsigned char *str)
{
while(*str)
{
     lcd_data(*str++);
}}
void sendser_char(unsigned char b)
{
```

```
SBUF=b;
     while(
     TI==0)
     ;
     TI=0;
}
void sendser_str(unsigned char *str)
{
while(*str)
{
     sendser_char(*str++);
}
}
```

RECEVIER

```
//CODE of lcd interfacing with 8051 microcontroller
#include<reg51.h>
void delay(unsigned int i);
void lcd_cmd(unsigned char a);
void lcd_data(unsigned char b);
void lcd_init(void);
void lcd_str(unsigned char *str); void
sendser_char(unsigned char b); void
sendser_str(unsigned char *str); sbit
rs=P2^0;
sbit en=P2^1;
//sbit sw=P2^5;
sbit led=P2^6;
sfr ldata=0x90;//port1
unsigned char rx;
void main()

{
          led=0;//led off
     TMOD=0x20;//timer1 mode2 -auto reload mode
     TH1=0xfd;//9600 baud rate
     SCON=0x50;//8bit data ,1start bit,1stop bit TR1=1;
lcd_init();
```

```
        lcd_str("
               WELCOME TO      ");
        lcd_cmd(0xc0);
        lcd_str("              MY PROJECT       ");

        delay(65000);
             lcd_cmd(0x01);
             lcd_cmd(0x80);
    lcd_str("               YT
        TRAINING            ");
        lcd_cmd(0xc0);
        lcd_str("
               INSTITUTE      ");
             delay(65000);
             lcd_cmd(0x01);
             lcd_cmd(0x80);
        lcd_str("    LED STATUS:
        "); while(1)
    {
        while(RI
        ==0);
        rx=SBUF;
        RI=0;
        if(rx=='A')
        {
        led=1;//led on
                       lcd_cmd(0x
        c0); lcd_str("      LED
        ON  ");
           //sendser_str("    LED ON  ");
                     //sendser_char(0x0d);
        }
        else if (rx=='B')
        {
             led=0;//led off
                       lcd_cmd(0x
        c0); lcd_str("      LED
        OFF  ");
    //  sendser_str("LED OFF  ");
        //  sendser_char(0x0d);
        }
        else
        {

        }
    }
    }
    void lcd_init()
```

Name: Avinash Rajkumar Kauran, Roll No: CS22006
Course: Embedded Internet Of Things

```
{
lcd_cmd(0x38);
     lcd_cmd(0x0c);
     lcd_cmd(0x01);
     lcd_cmd(0x80);


}
void delay(unsigned int i)
{
unsigned int j; for(j=0;j<i;j++);
}

void lcd_cmd(unsigned char a)
{
  rs=0;//cmd
     ldata=a;
     en=1;
     delay(5);
     en=0;
     delay(5);
}
void lcd_data(unsigned char b)
{
  rs=1;//data
     ldata=b;
     en=1;
     delay(5);
     en=0;
     delay(5);
}
void lcd_str(unsigned char *str)
{
while(*str)
{
     lcd_data(*str++);
}

}


void sendser_char(unsigned char b)
{
SBUF=b;
     while(TI==0); TI=0;
}


void sendser_str(unsigned char *str)
```

```
{
while(*str)
{
    sendser_char(*str++);
}

}
```

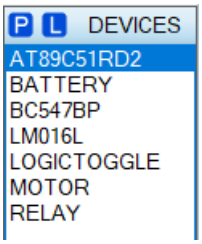
Name: Avinash Rajkumar Kauran, Roll No: CS22006
Course: Embedded Internet Of Things

PRACTICAL 3

AIM: Develop an IoT application for monitoring water levels in tanks and automatically start the motor to fill the tank if the level goes below the critical level.

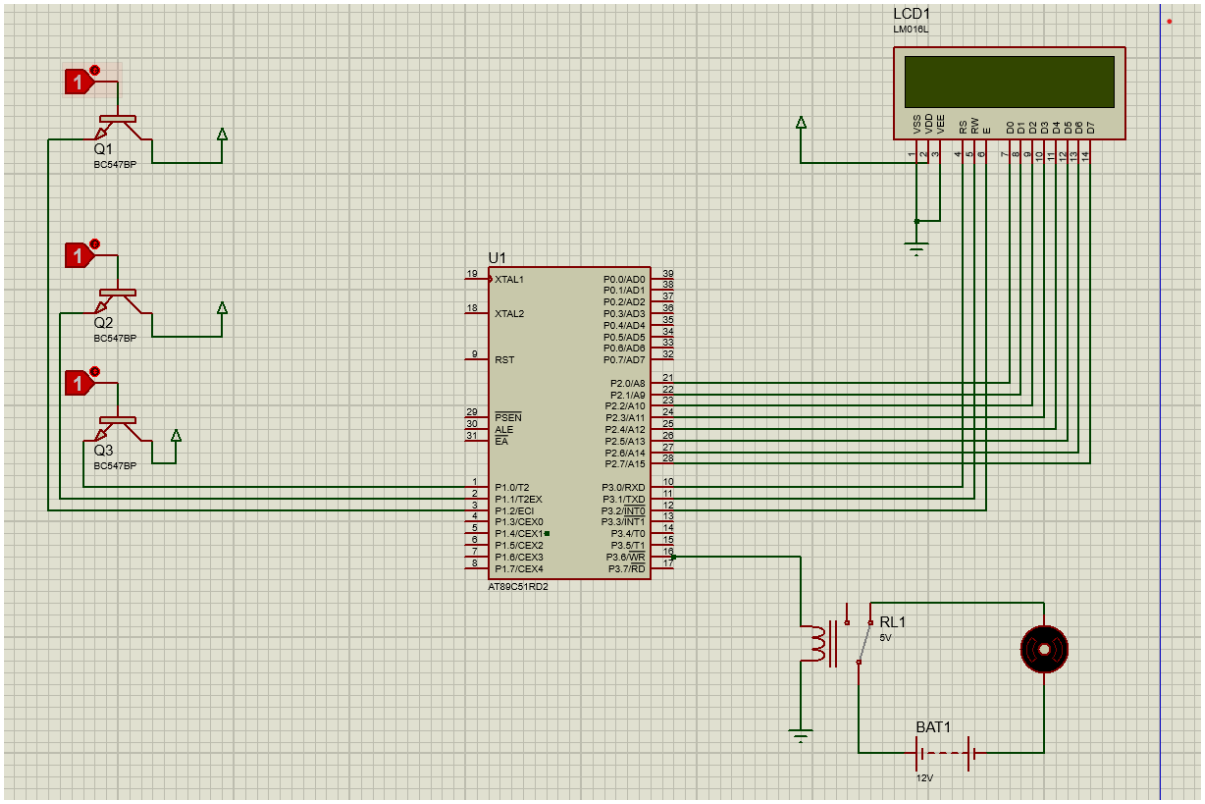
CONCEPT: When the water tank is empty or water level is low the motor will keep working and the display with show the water level as soon as the water tank is full and then the display show full and the will automatically stop working

COMPONENTS USED :

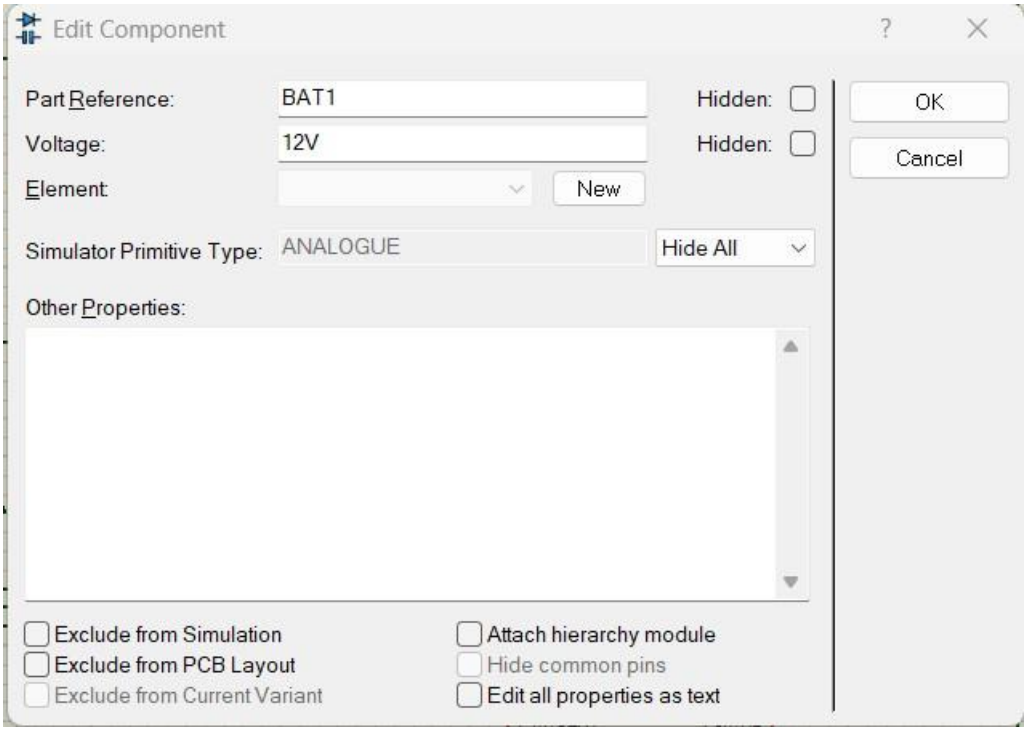


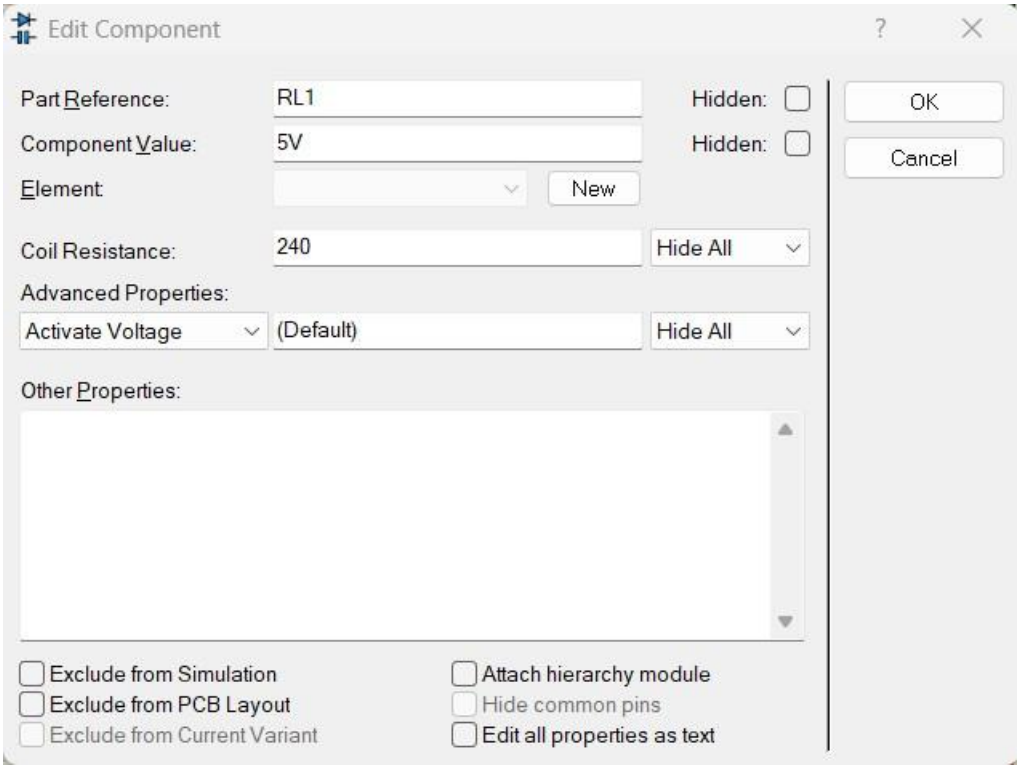
STEP 1 : Create the circuit diagram in proteus 8

DIAGRAM:



STEP 2: Set battery at 12 volts

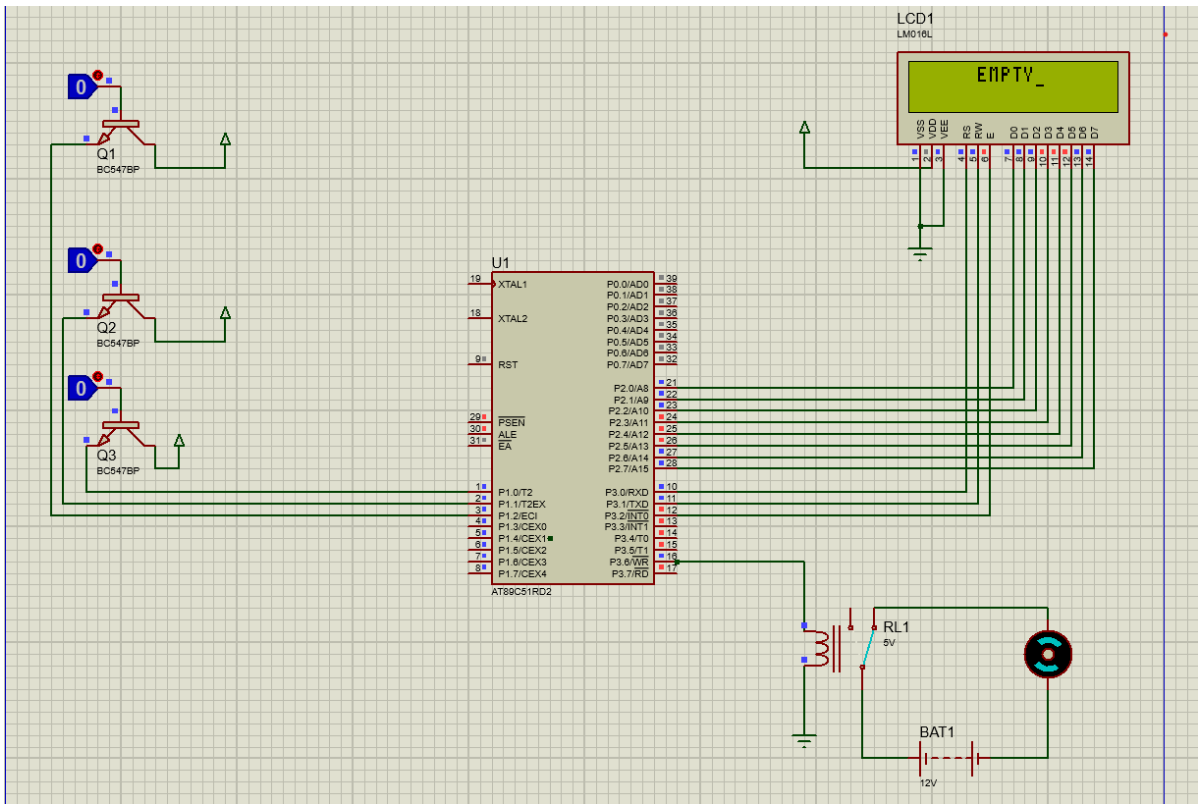




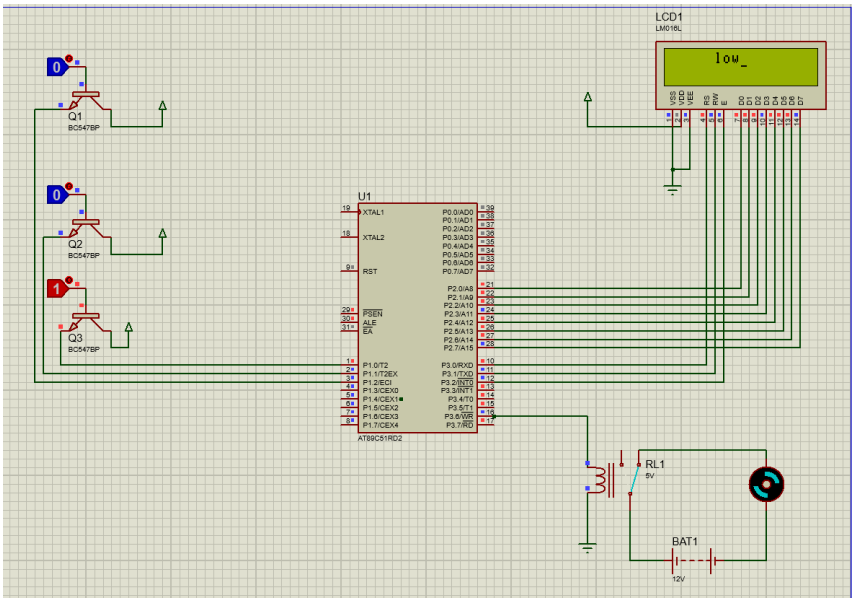
STEP 3 :Keep the relay at 5 volts

Name: Avinash Rajkumar Kauran, Roll No: CS22006
Course: Embedded Internet Of Things

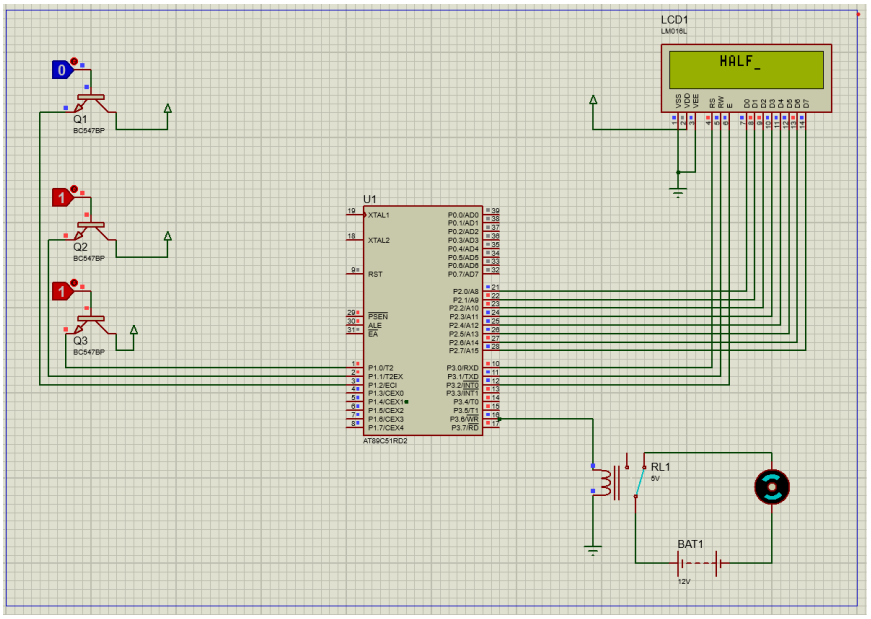
OUTPUT: (When the water tank is empty the motor keeps running)



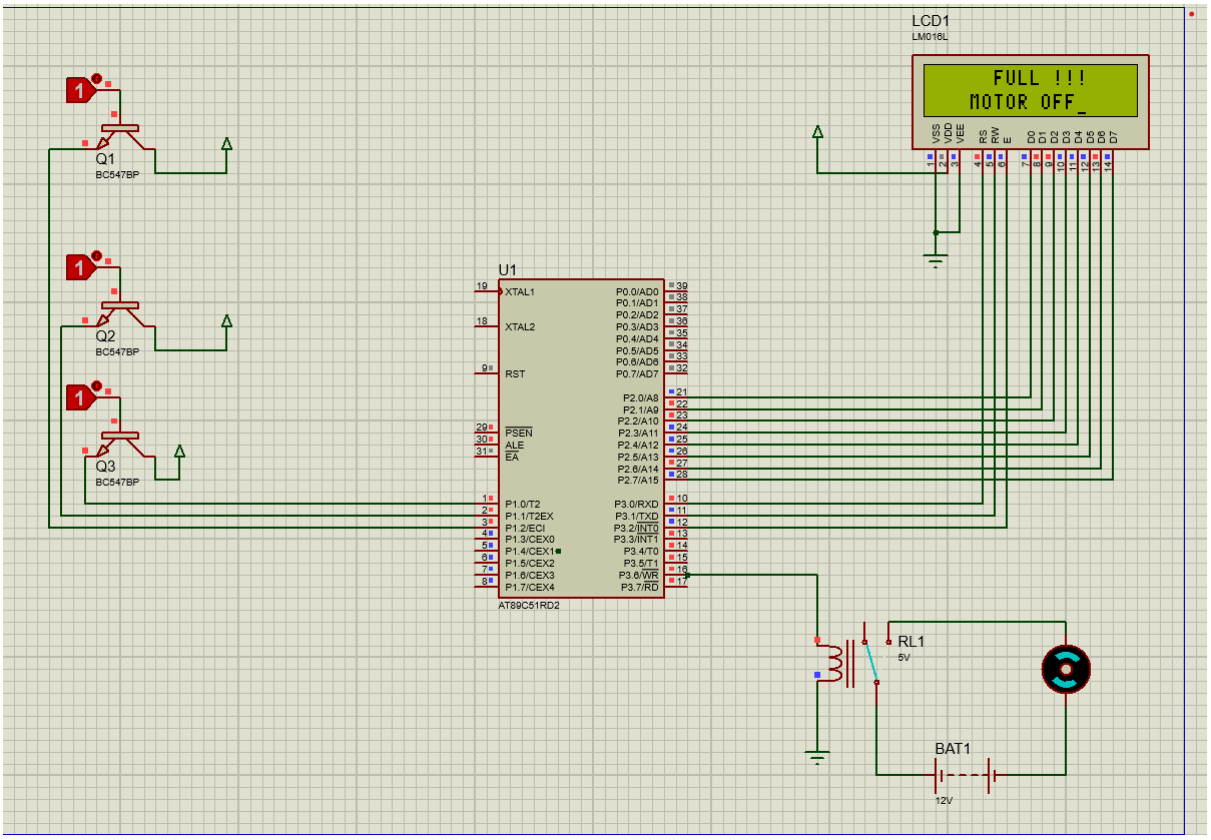
(When the water level is low , the motor keeps running)



(When the water level is half , the motor keeps running)



(When the water level is full , the motor is switched off)



CODE:

```
#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 buzzer=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; buzzer=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)

{
buzzer=1; lcdcmd(0x85); lcdstring("FULL !!!"); lcdcmd(0xc3);
lcdstring("MOTOR OFF"); buzzer=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;
}
```

Name: Avinash Rajkumar Kauran, Roll No: CS22006
Course: Embedded Internet Of Things

PRACTICAL 4

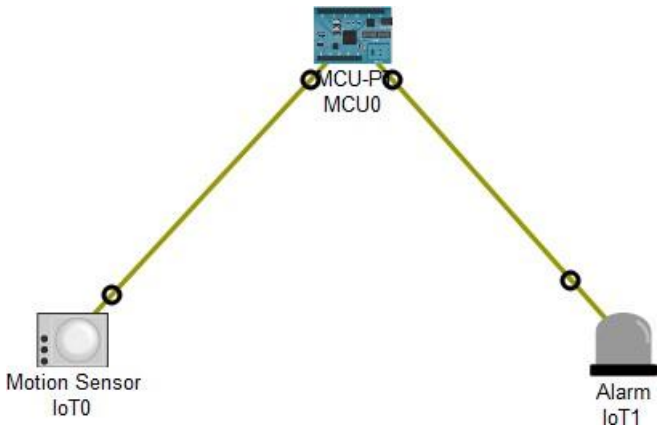
AIM: Develop an IoT application for Motion detection.

Concept: The microcontroller is connected to a motion sensor and an alarm

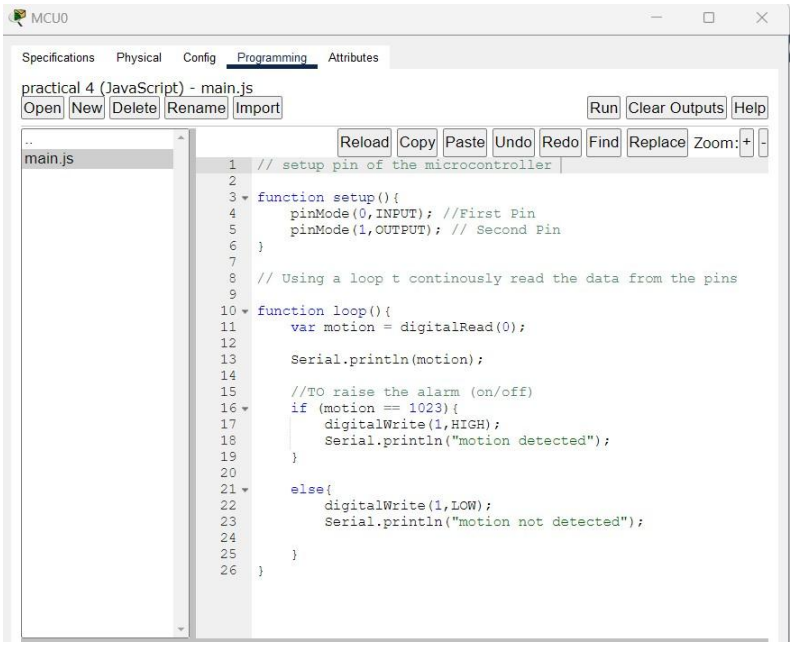
When the sensor will detect a motion around it the microcontroller will switch on the Alarm

Step 1 : set a diagram as given

below Diagram:



Step 2 : Double click the microcontroller create a new file and write the following code

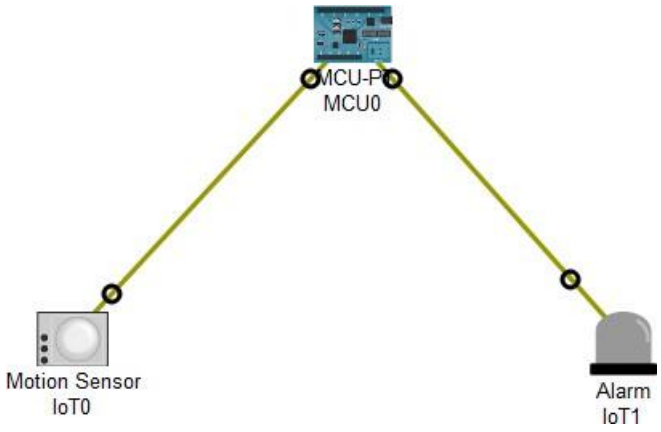


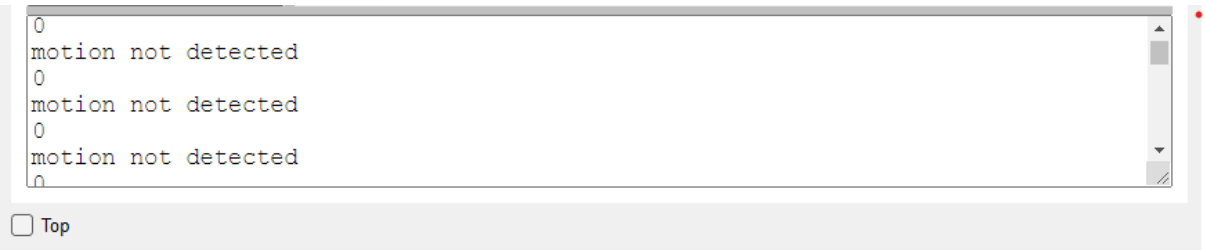
SOURCE CODE :

```
// setup pin of the

microcontroller function setup(){
        pinMode(0,INPUT); //First Pin
        pinMode(1,OUTPUT); // Second Pin
}

// Using a loop t continously read the data from the pins

function loop(){
        var motion =

        digitalRead(0);

        Serial.println(motion);

        //TO raise the alarm
        (on/off) if (motion ==
        1023){
                digitalWrite(1,HIGH);
                Serial.println("motion
                detected");
        }

        else{
           digitalWrite(1,LOW);
           Serial.println("motion not
           detected");

        }
}
```

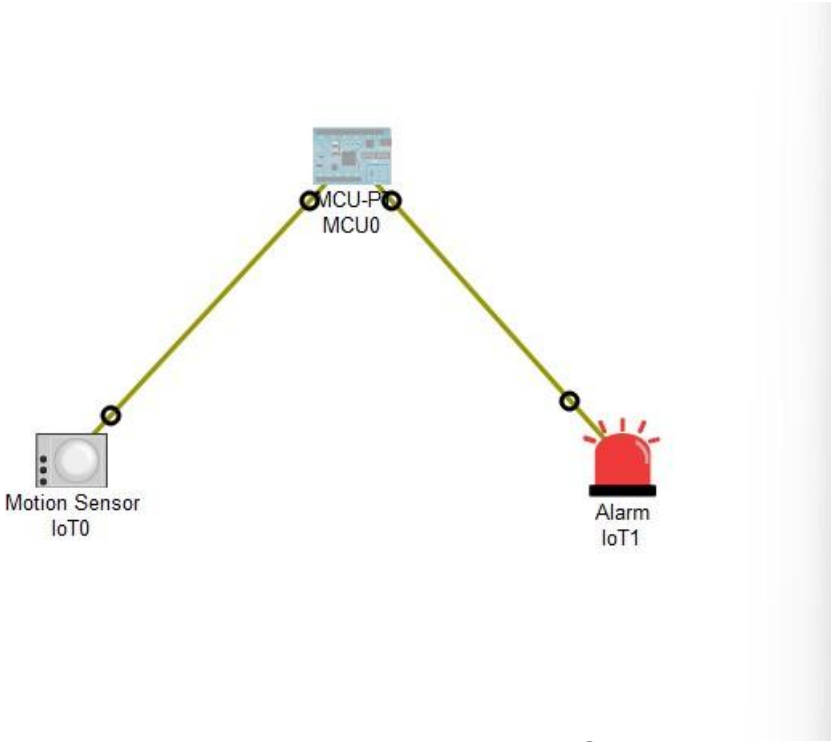
Name: Avinash Rajkumar Kauran, Roll No: CS22006
Course: Embedded Internet Of Things

OUTPUT : when the senor detects nothing





When the motion sensor detects a motion




motion detected
1023
motion detected
1023
motion detected
1023
Top

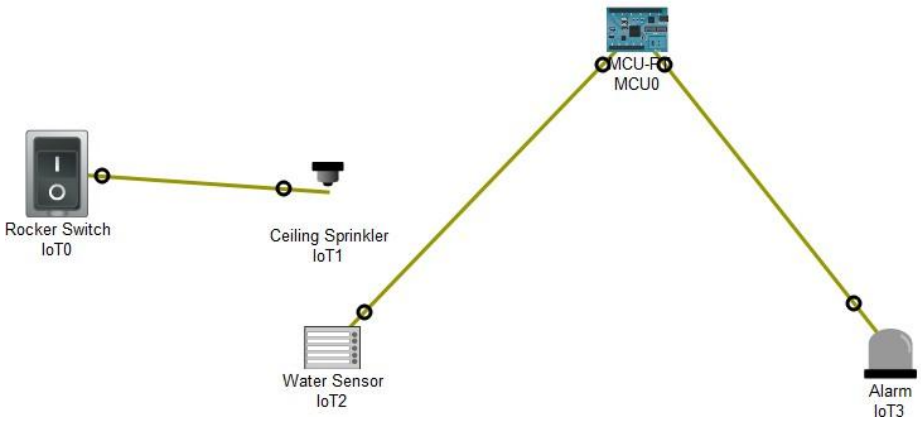
Name: Avinash Rajkumar Kauran, Roll No: CS22006
Course: Embedded Internet Of Things

PRACTICAL 5

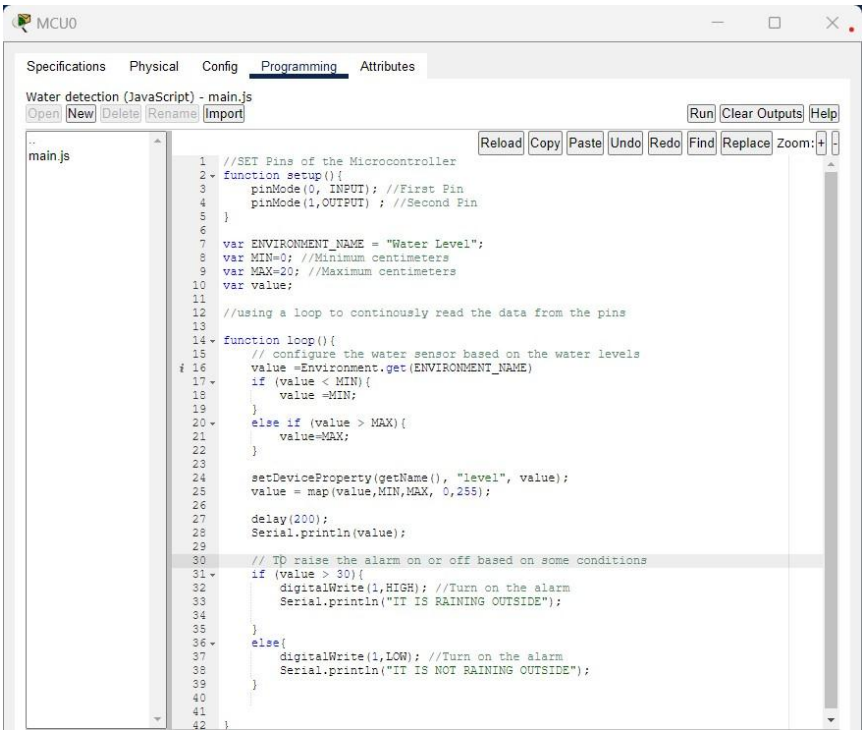
AIM: Develop an IoT application that will raise an alarm whenever with going to rain

CONCEPT :When ceiling sprinkler is switched on (mimics rain) water senor will detect it and the alarm will be switched on

DIAGRAM : Step 1 connect the components as shown below



Step 2 : double click on the microcontroller to program and write the code below



SOURCE CODE :

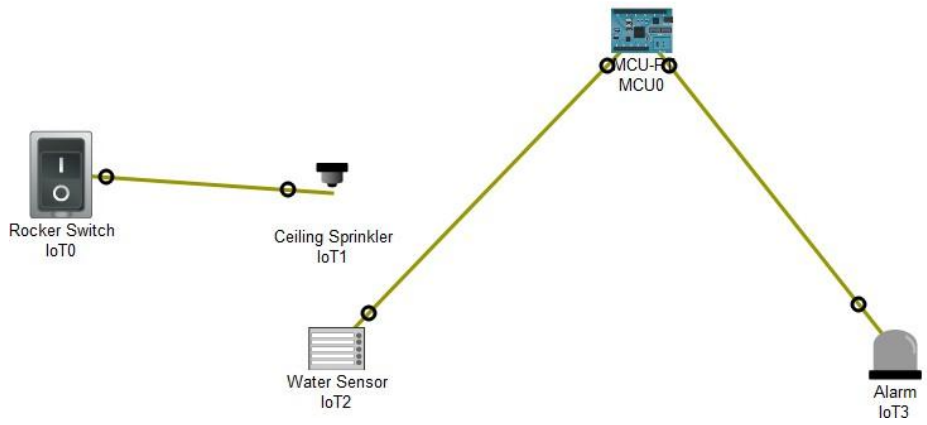
```
//SET Pins of the Microcontroller
function setup(){
        pinMode(0, INPUT); //First Pin
        pinMode(1,OUTPUT) ; //Second Pin
}

var ENVIRONMENT_NAME = "Water
Level"; var MIN=0; //Minimum
centimeters
var MAX=20; //Maximum
centimeters var value;


//using a loop to continously read the data from the pins

function loop(){
        // configure the water sensor based on the water
        levels value
        =Environment.get(ENVIRONMENT_NAME)
        if (value < MIN){
                value =MIN;
        }
        else if (value > MAX){
                value=MAX;
        }

        setDeviceProperty(getName(), "level",
        value); value = map(value,MIN,MAX, 0,255);

        delay(200);
        Serial.println(value)
        ;
```

Name: Avinash Rajkumar Kauran, Roll No: CS22006
Course: Embedded Internet Of Things

```
        // TO raise the alarm on or off based on some
        conditions if (value > 30){
                digitalWrite(1,HIGH); //Turn on the alarm
                Serial.println("IT IS RAINING OUTSIDE");


        }       digitalWrite(1,LOW); //Turn on the alarm
        e       Serial.println("IT IS NOT RAINING OUTSIDE");
        l
        s
        e
        {
}
```

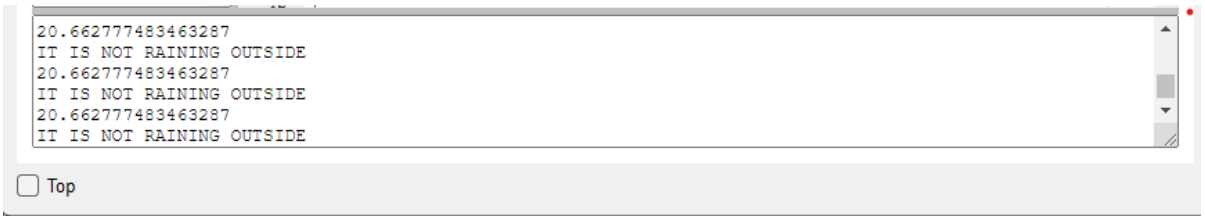
OUTPUT:

When there is no rain outside

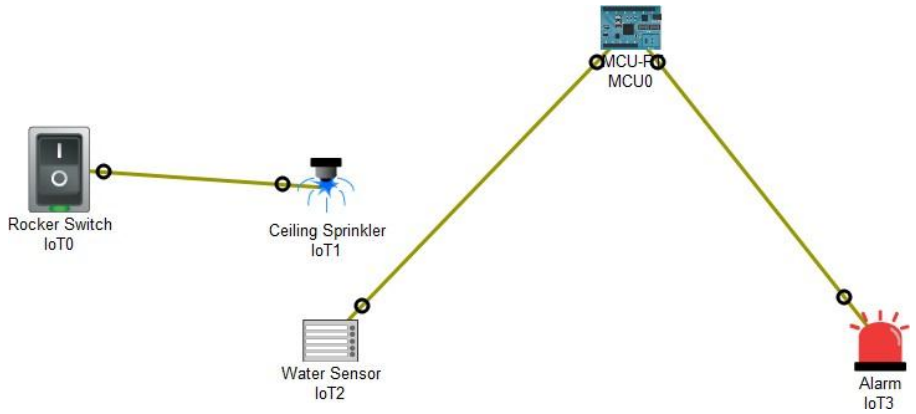


The alarm is off

The consoles shows (IT IS NOT RAINING OUTSIDE)

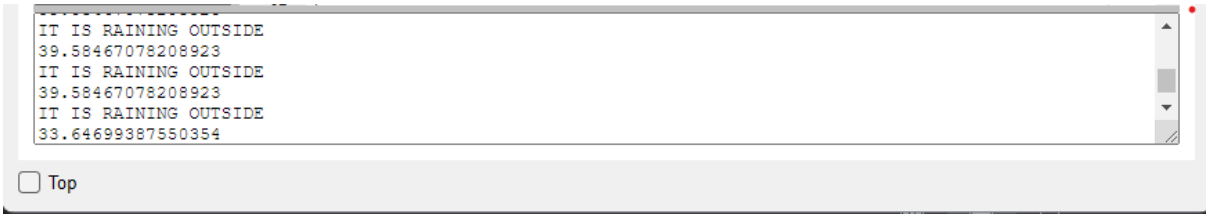


When the ceiling sprinkler is on (mimics rain)



The alarm is turned on

The consoles shows (IT IS NOT RAINING OUTSIDE)



Name: Avinash Rajkumar Kauran, Roll No: CS22006
Course: Embedded Internet Of Things

PRACTICAL 6

AIM: Develop an IoT module to which measure the intensity of light and display it on a lcd screen

CONCEPT : when the light source is near the LDR (light detecting resistor) sensor . The light intensity will be high , and when the light source is far from the sensor the light intensity will be low

COMPONENTS USED :

1. ARDUINO UNO
2. LDR SENSOR WITH TORCH
3. 10K RESISTOR
4. 16X2 LCD DISPLAY
5. GROUND

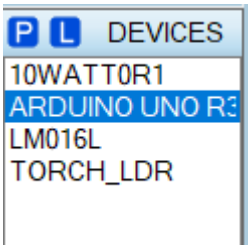
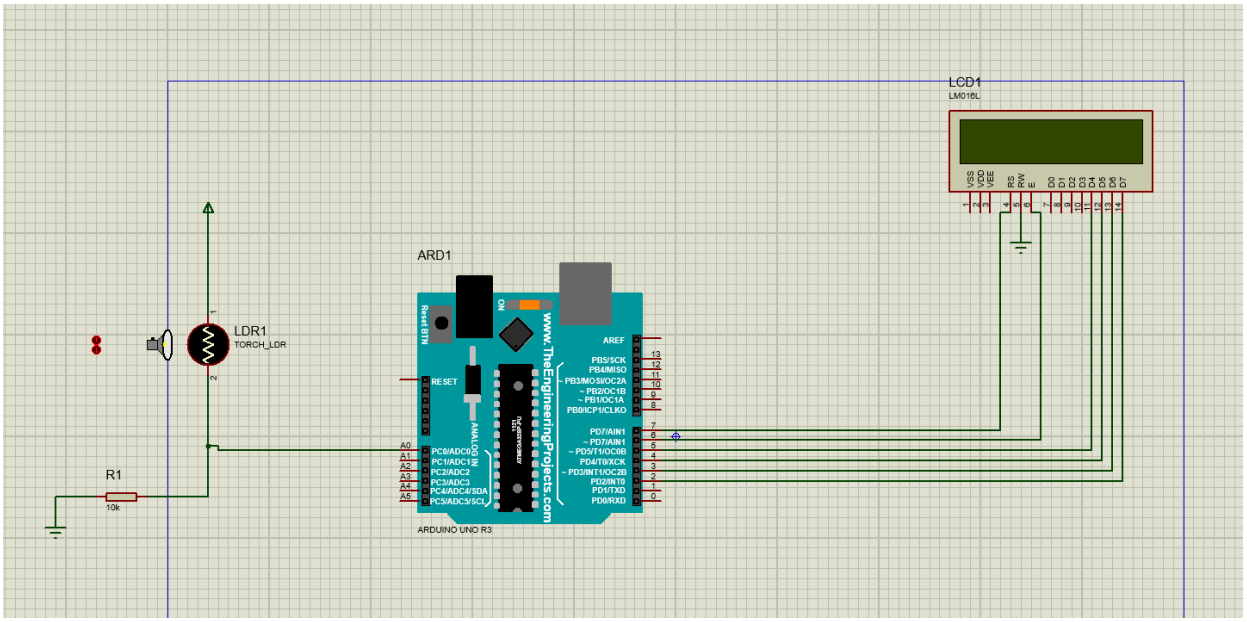
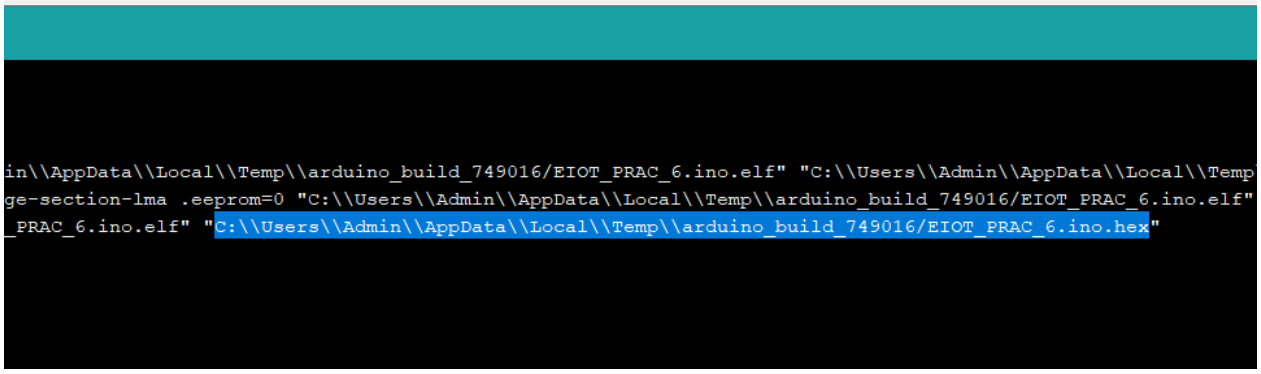


DIAGRAM:

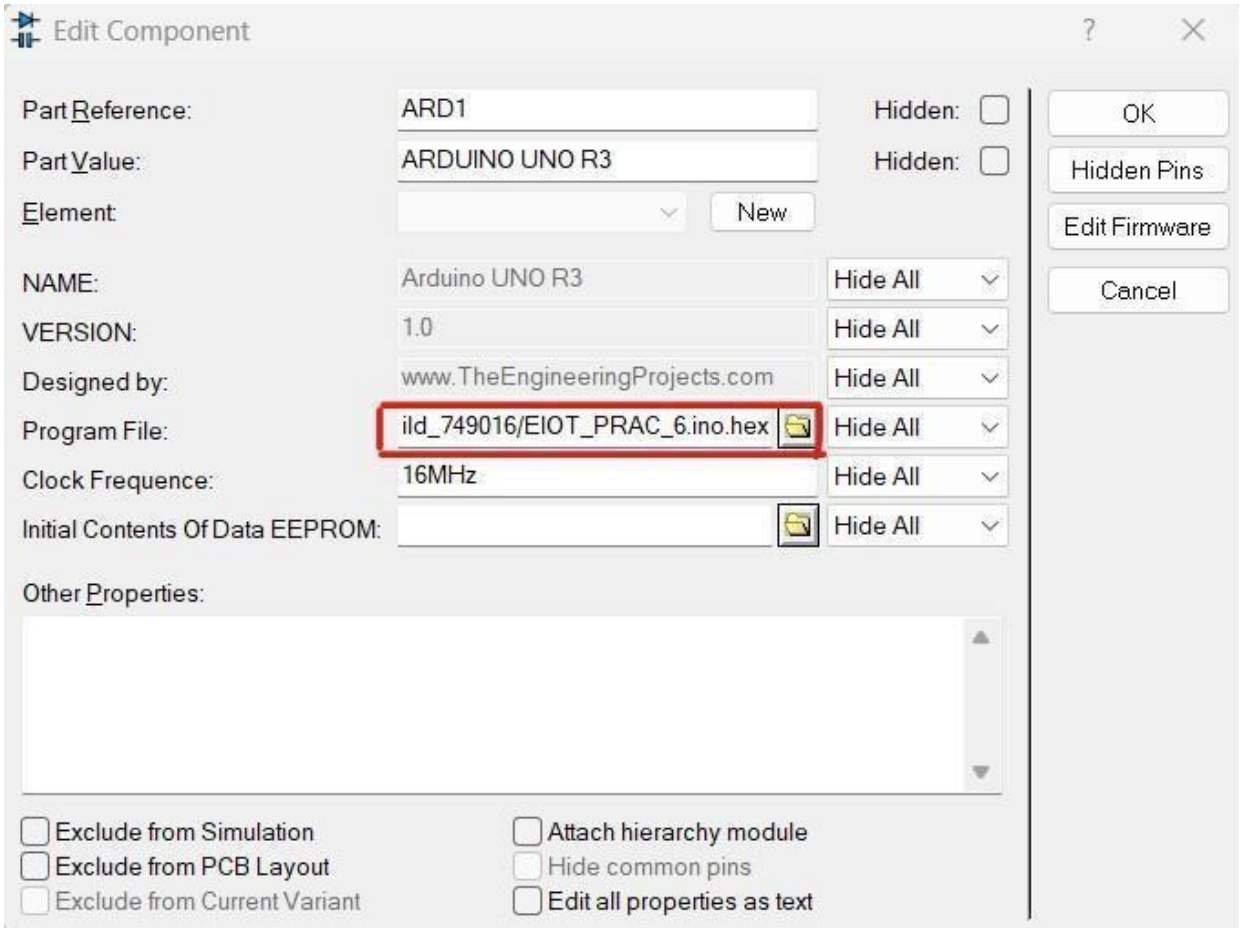


Step 1 : build the circuit in proteus as shown in the digram

Step 2 : write and run the arduino code in arduino IDE Step 3 :
Copy the hex file from the terminal of arduino IDE



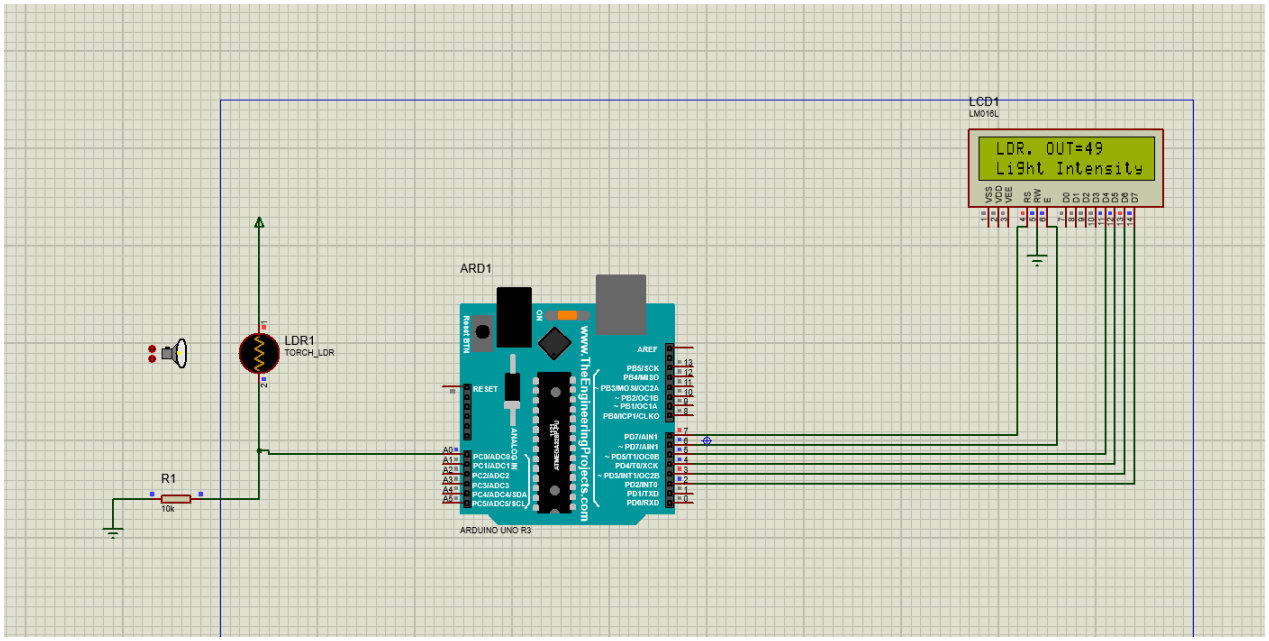
Step 4: Paste the hex file in the arduino component in proteus



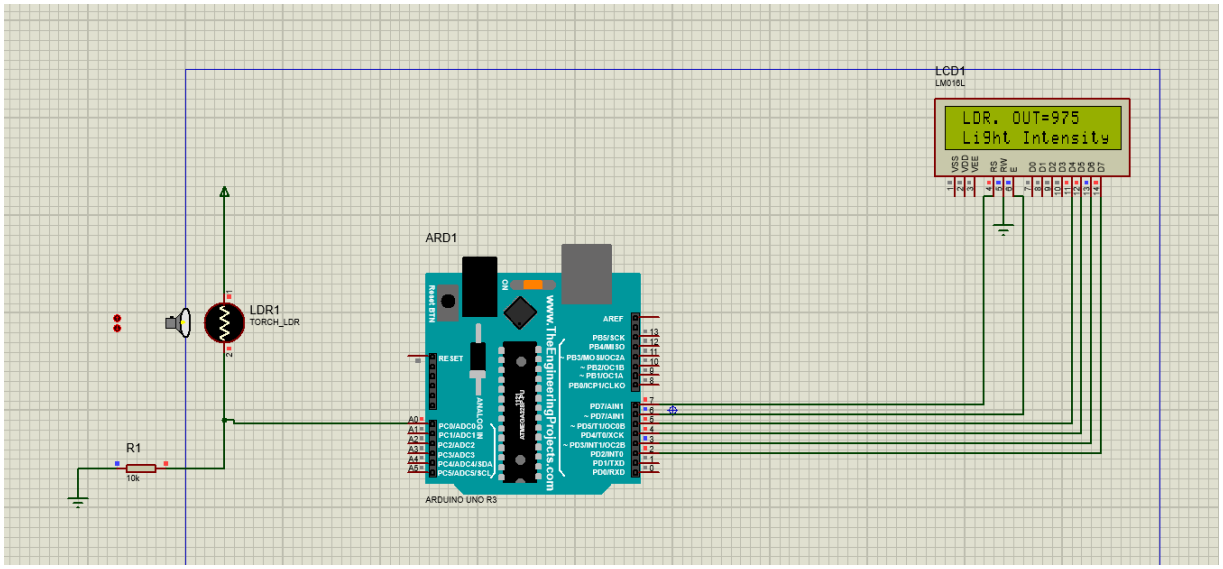
Name: Avinash Rajkumar Kauran, Roll No: CS22006
Course: Embedded Internet Of Things

OUTPUT :

When the light intensity is low



When the light intensity is low



Code :

```
#include <LiquidCrystal.h>
LiquidCrystal lcd(7,6,5,4,3,2);
void setup() {
  // put your setup code here, to run once: lcd.begin(16,2);
 // print
 lcd.setCursor(1,0);
 lcd.print("LDR.
 OUT=");
 lcd.setCursor(1,1);
 lcd.print("Light
 Intensity");
}

void loop() {
  // put your main code here, to run repeatedly: int ldr =
 analogRead(A0);
 lcd.setCursor(10,0); lcd.print(ldr);
}
```

Name: Avinash Rajkumar Kauran, Roll No: CS22006
Course: Embedded Internet Of Things

PRACTICAL 7

Aim: Built an IoT system to send a ticket before entering the bus.

Theory

RFID: Radio-frequency identification (RFID) uses electromagnetic fields to identify and track tags attached to objects automatically. An RFID system consists of a tiny radio transponder, a radio receiver and a transmitter. When triggered by an electromagnetic interrogation pulse from a nearby RFID reader device, the tag transmits digital data back to the reader, usually an identifying inventory number. This number can be used to track inventory goods.

Material

- RFID card, RFID Reader, MCU Board, LCD Display
- Door, Home Gateway, PC

Method

1. We simulate the back door of the bus where the passengers get onboard for this practical which will be conducted in Cisco Packet Tracer.

2. Drag and drop all components from apparatus section to the simulating area.

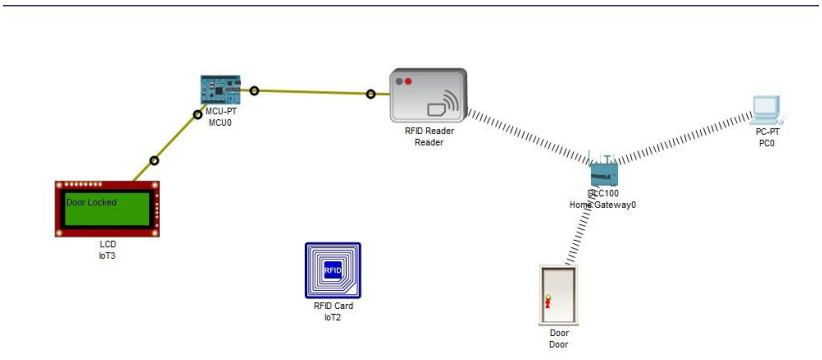
3. Connect LCD Display to MCU Board with IoT cables on the „D0" port for both sides.

4. Connect RFID Reader to MCU Board with IoT cables on the „A1" port for both sides. If A1 port is not available open RFID Reader, go to I/O config and set Analog ports to 2.

5. Now connect RFID Reader to Home Gateway by changing the network adapter in the I/O config of RFID Reader to PT-IOT-1W-AC and it should connect automatically.

6. Connect the PC to Home Gateway wirelessly by replacing its network component with a wireless one which is any of the ones shown below in the physical section of the PC.

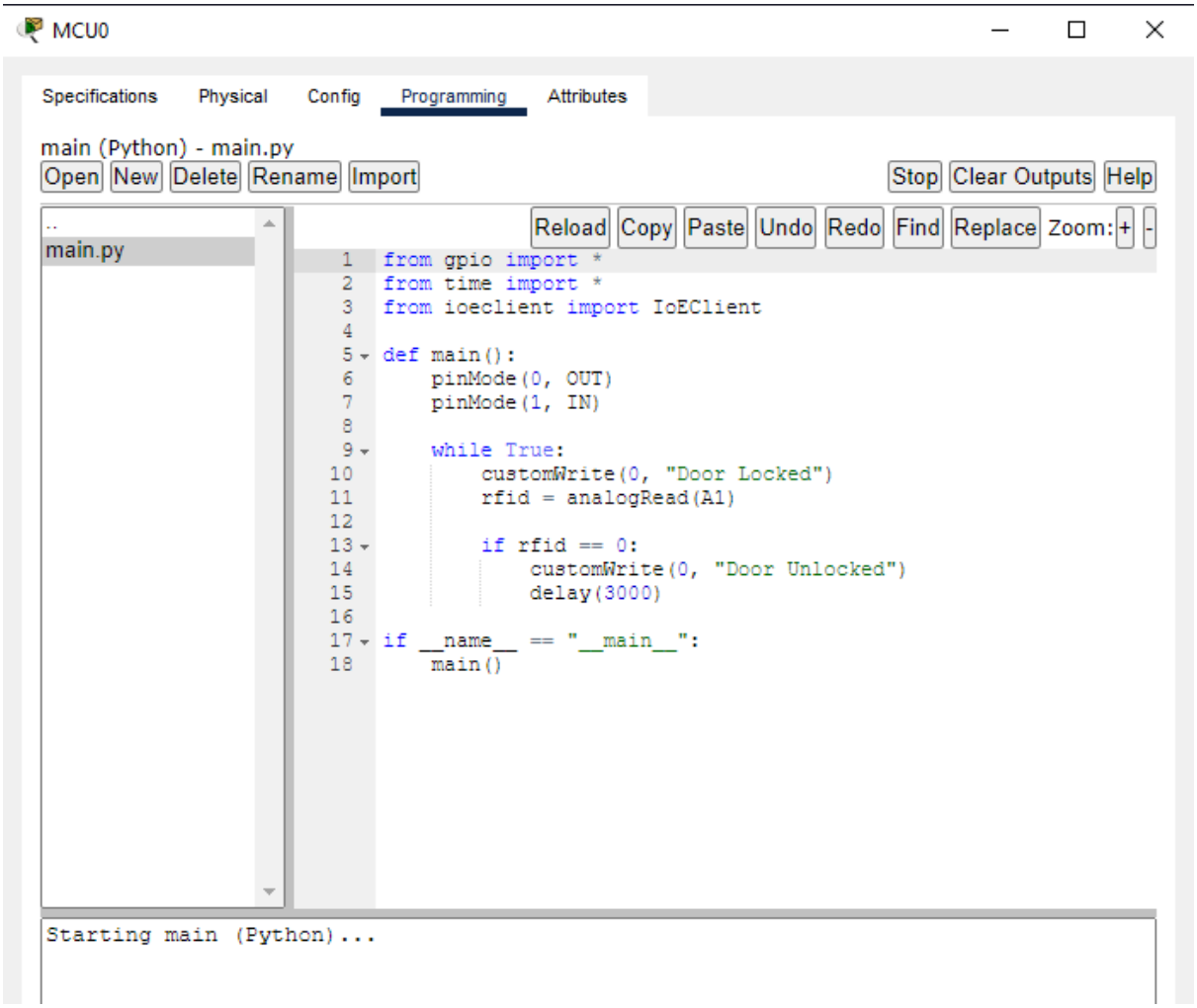
7. Your things network should be similar to this.



8. 8. Go to config of PC and in wireless set the SSID to HomeGateway“.

9. Now on MCU Board create an empty Python project and type the following code and run it.

10.Now in RFID Reader, go to programming and



make the following changes in the loop function of

main.py

Change setState() parameter from 2 to 1

Add the following line of code above

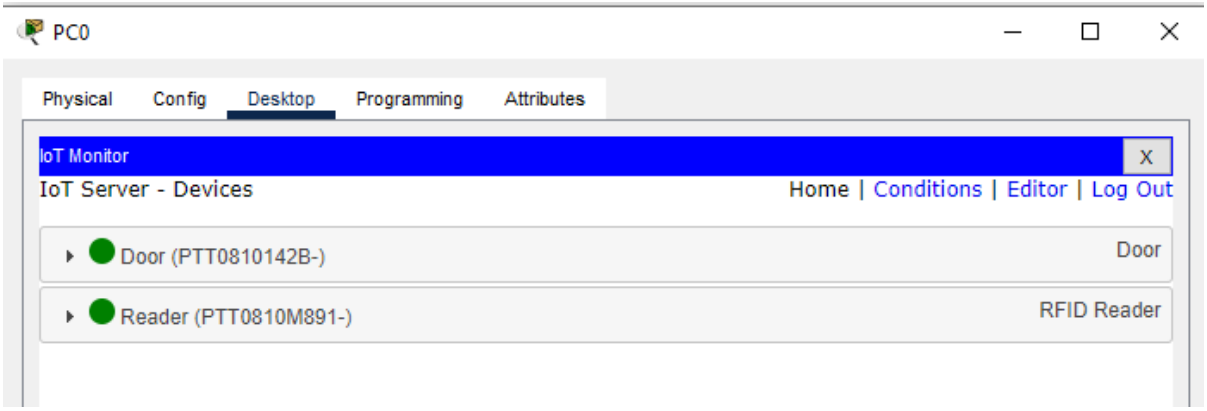
delay(DELAY_TIME) ifcardID == ‘1001’:
    setState(0)
else:
    setState(1)

the loop function should look like this:

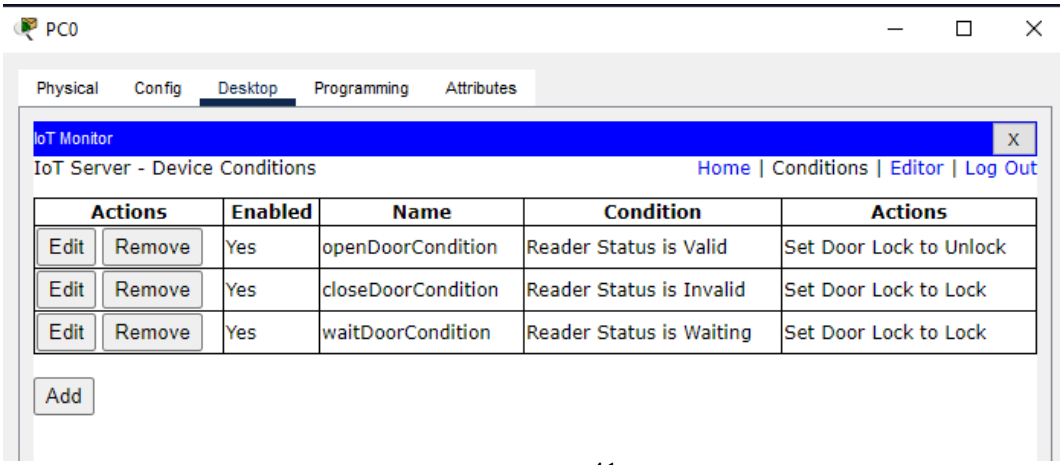
Reload Copy Paste Undo Redo Find Replace Zoom: +

```
48      devices = devicesAt(getCenterX(), getCenterY(), X_READ_DI
49      found = False        # var found
50      for i in xrange(0, len(devices)) :
51          if devices[i] is getName():
52              continue
53
54          cardID = getDeviceProperty(devices[i], 'CardID')
55          found = True
56          break
57
58
59      if not found:
60          cardID = lastCardID = 0
61          setState(2)
62      else:
63          if lastCardID != cardID:
64              lastCardID = cardID
65              sendReport()
66
67      if cardID == '1001':
68          setState(0)
69      else:
70          setState(1)
71
72
73      delay(DELAY_TIME)
```

11.Now set the IoT Server of all components to Home Gateway and in PC IoT Monitor make sure these components are visible



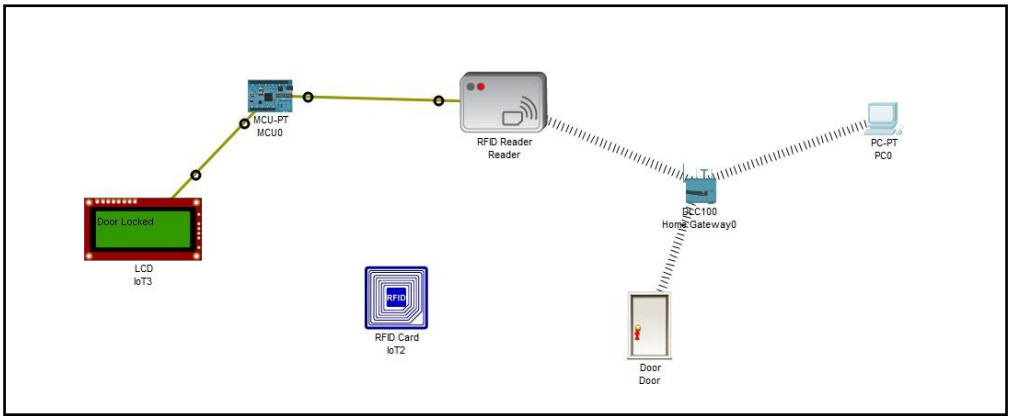
12.Now create the following functions in PC



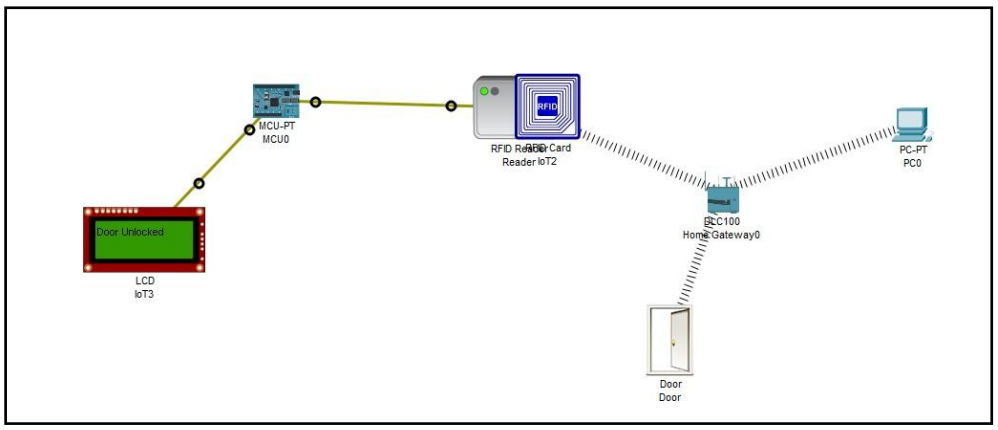
Now Drag and hover the RFID card over the reader while pressing the ALT key to get the output.

Output:

Before RFID ID Card hovered over RFID Reader(Door is Closed)



After RFID ID Card hovered over RFID Reader(Door is Open)



Name: Avinash Rajkumar Kauran, Roll No: CS22006
Course: Embedded Internet Of Things

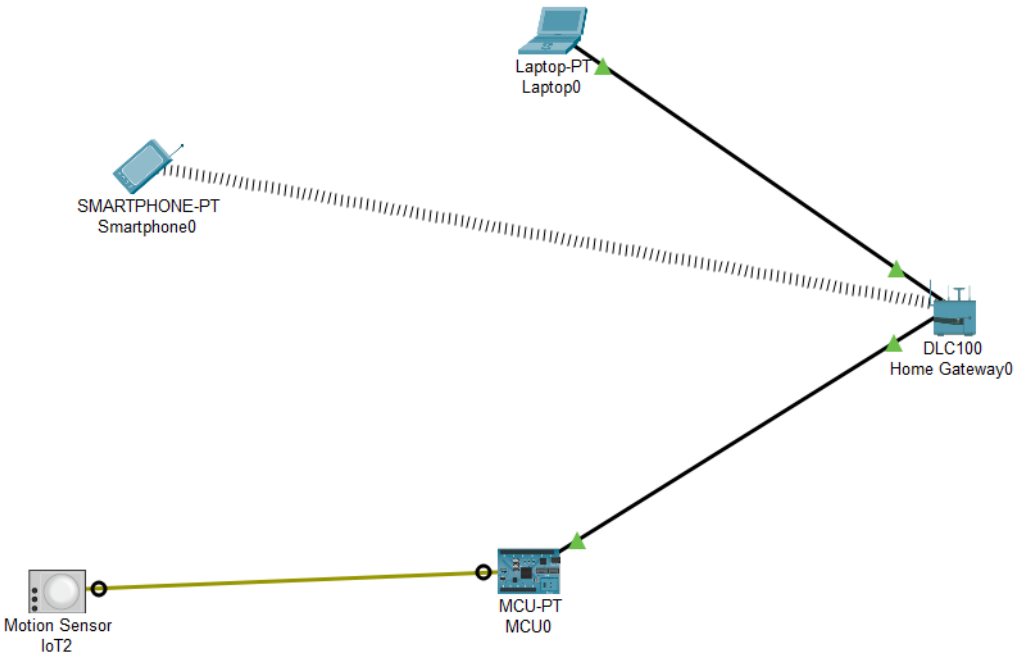
PRACTICAL 8

Aim: Develop a IoT application which will record the movement and orientation of your phone and give the data back to the PC.

Requirements:

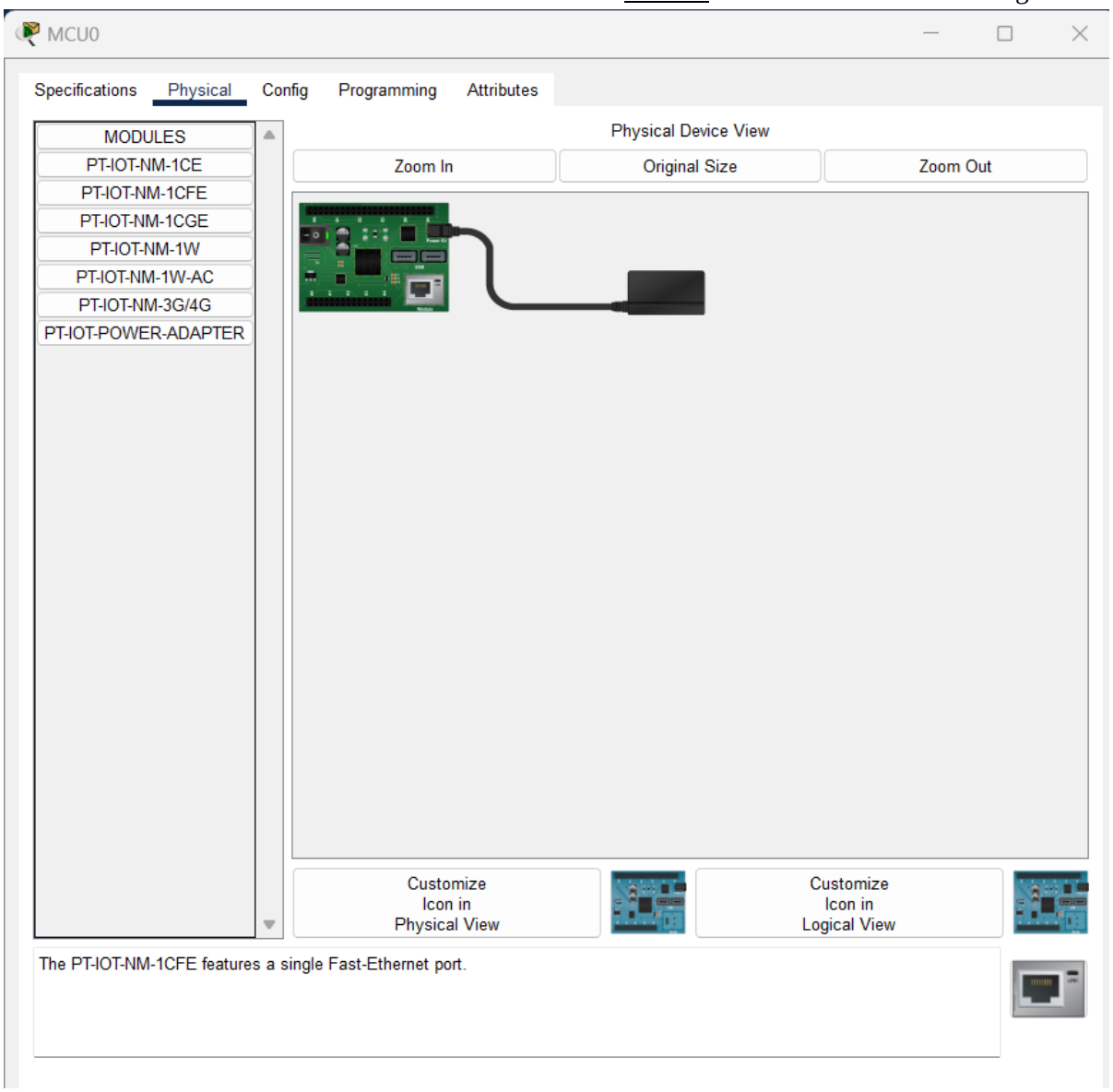
- **1 Motion Sensor (IoT1)**
- **1 Microcontroller Unit (MCU-PT)**
- **1 Home Gateway (DLC100)**
- **1 Laptop (Laptop-PT)**
- **1 Smartphone (SMARTPHONE-PT)**

Step 1: Select the requirements and make their configurations.

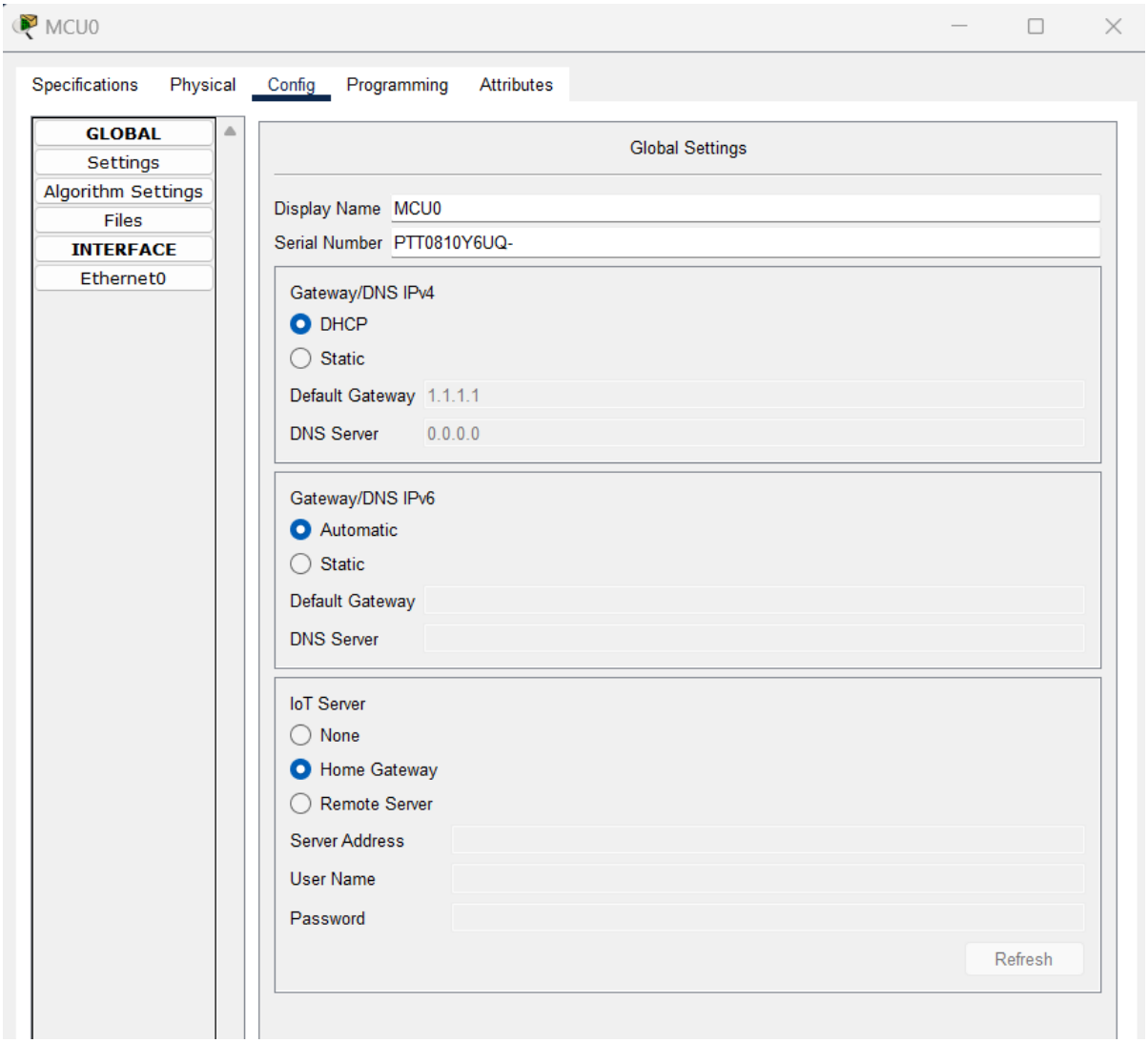


- **Motion Sensor:**
  Connect the Motion Sensor to the MCU-PT via the IOT Custom Cable.
  To use the motion sensor press ‘alt’ followed by the hovering of your cursor over the sensor.

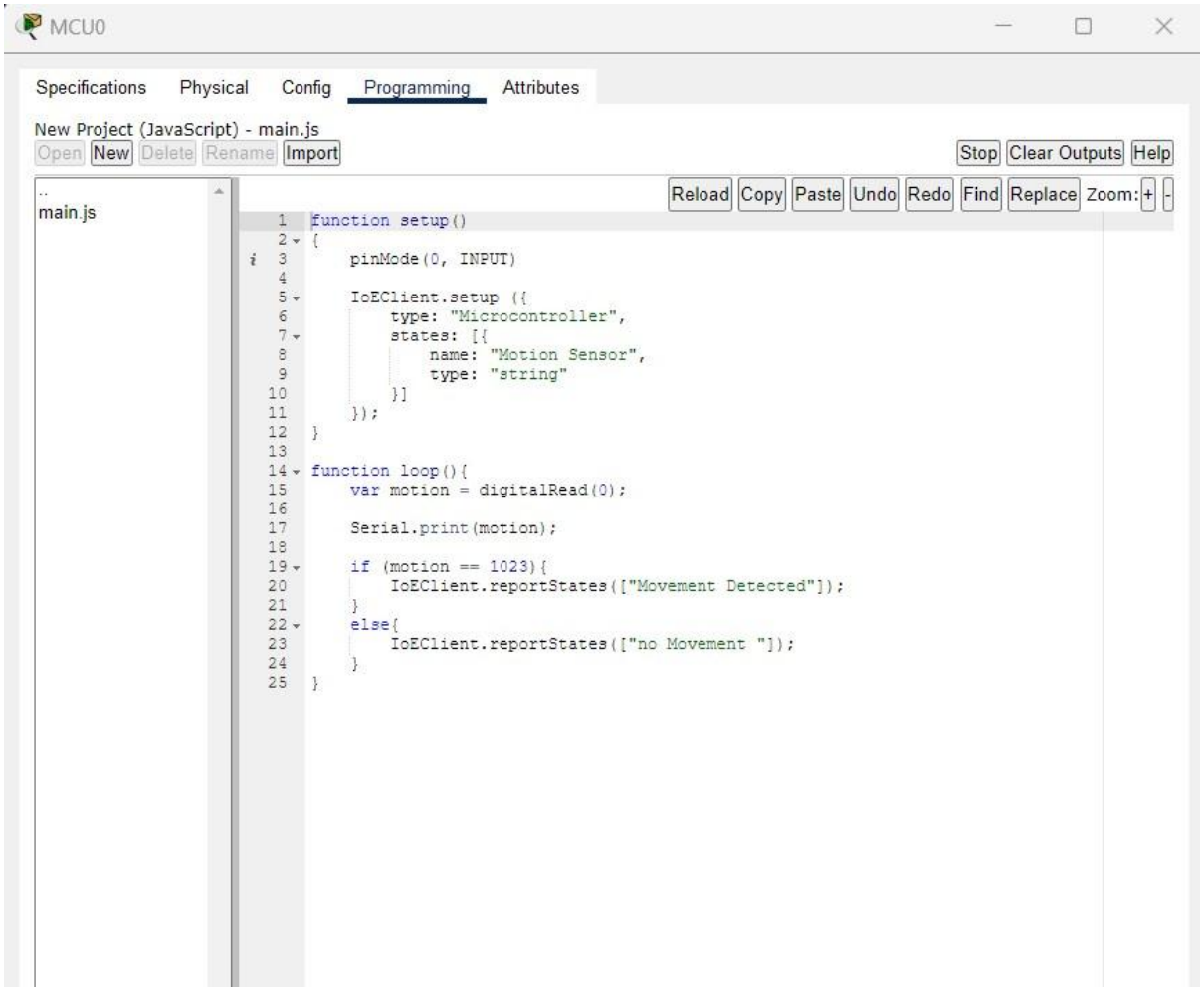
- **MCU-PT:**
  Add the FastEthernet Port to the microcontroller.



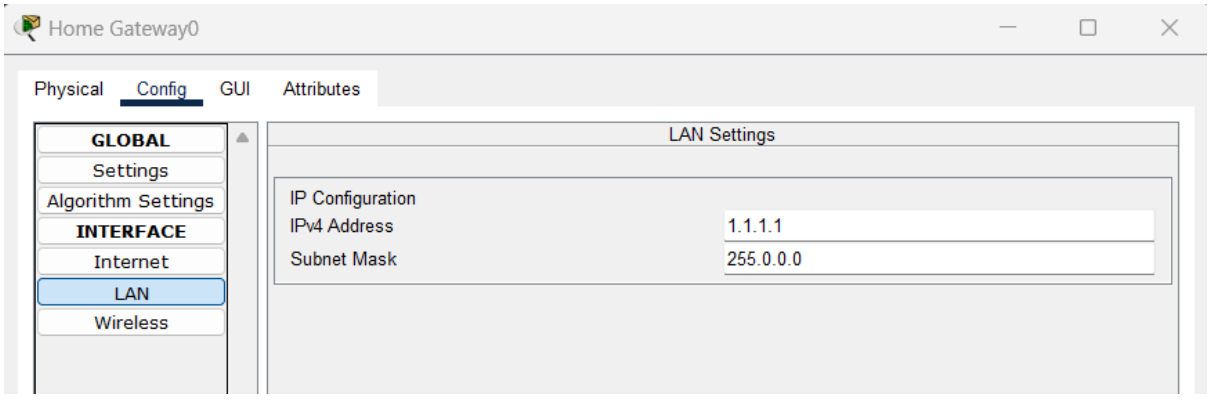
Connect the MCU-PT to the Home Gateway using a copper wire and configure its connection settings.

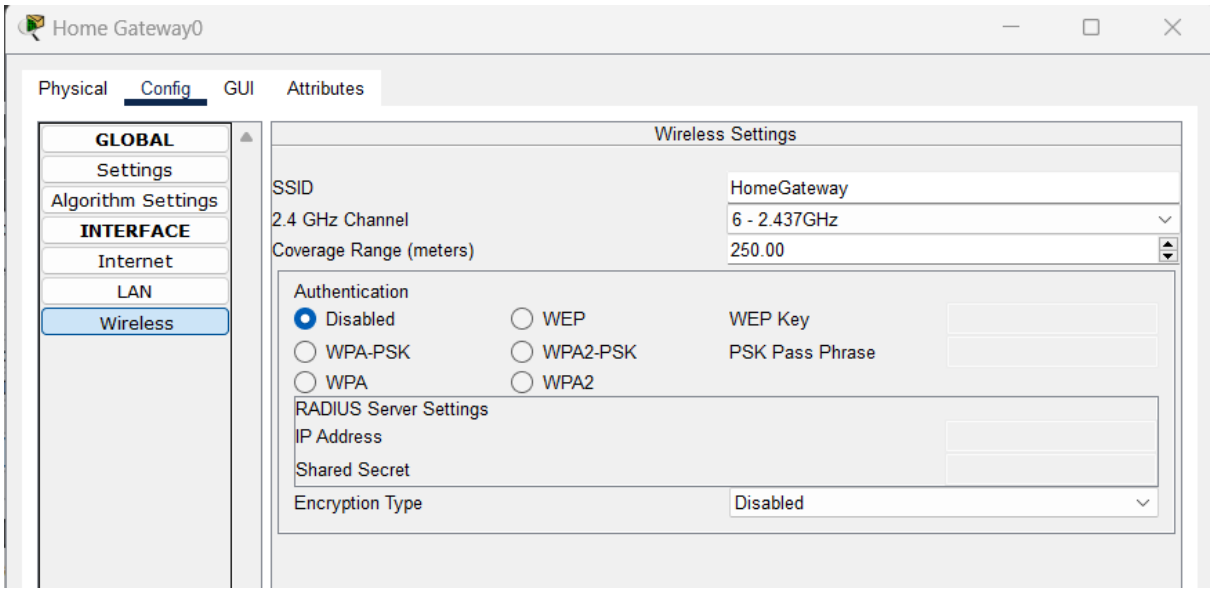


Add the code required for it to process the analog signals of the motion sensor so that it can send it to the Home Gateway.

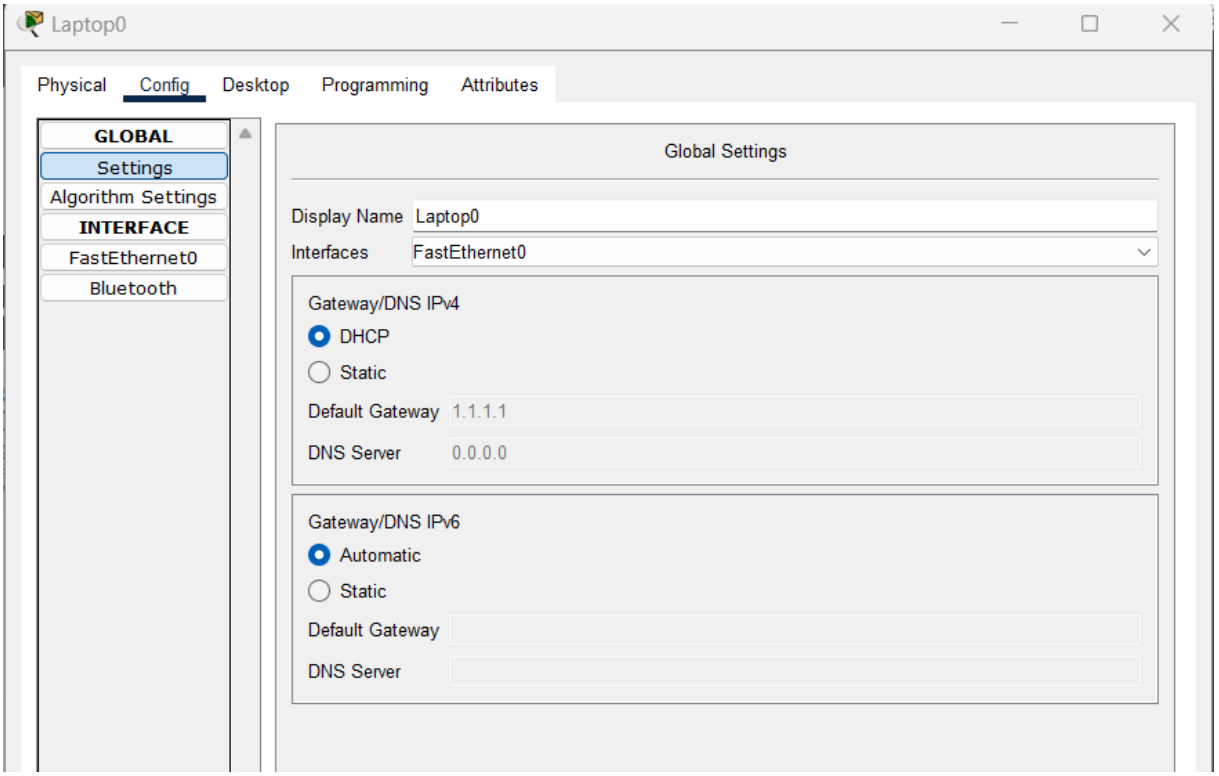


- **Home Gateway:**
  Make the following configurations to the Home Gateway and connect it to the laptop via a copper cable.

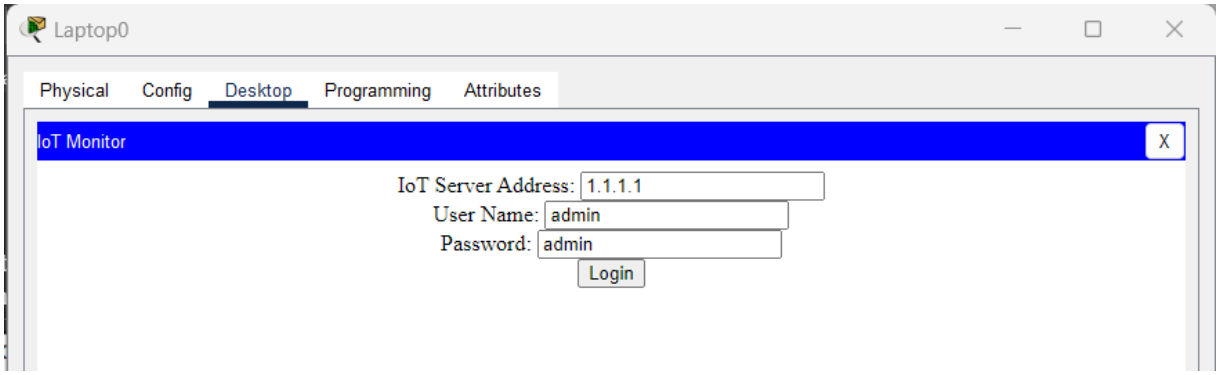


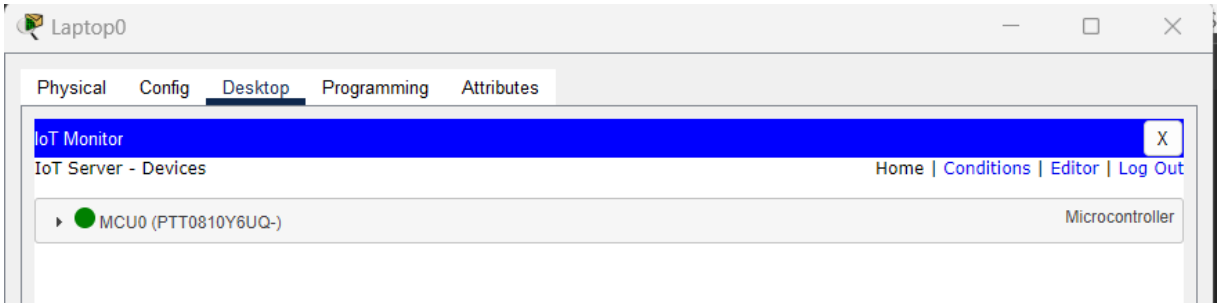


- **Laptop:**
  Make the following configuration to the laptop.

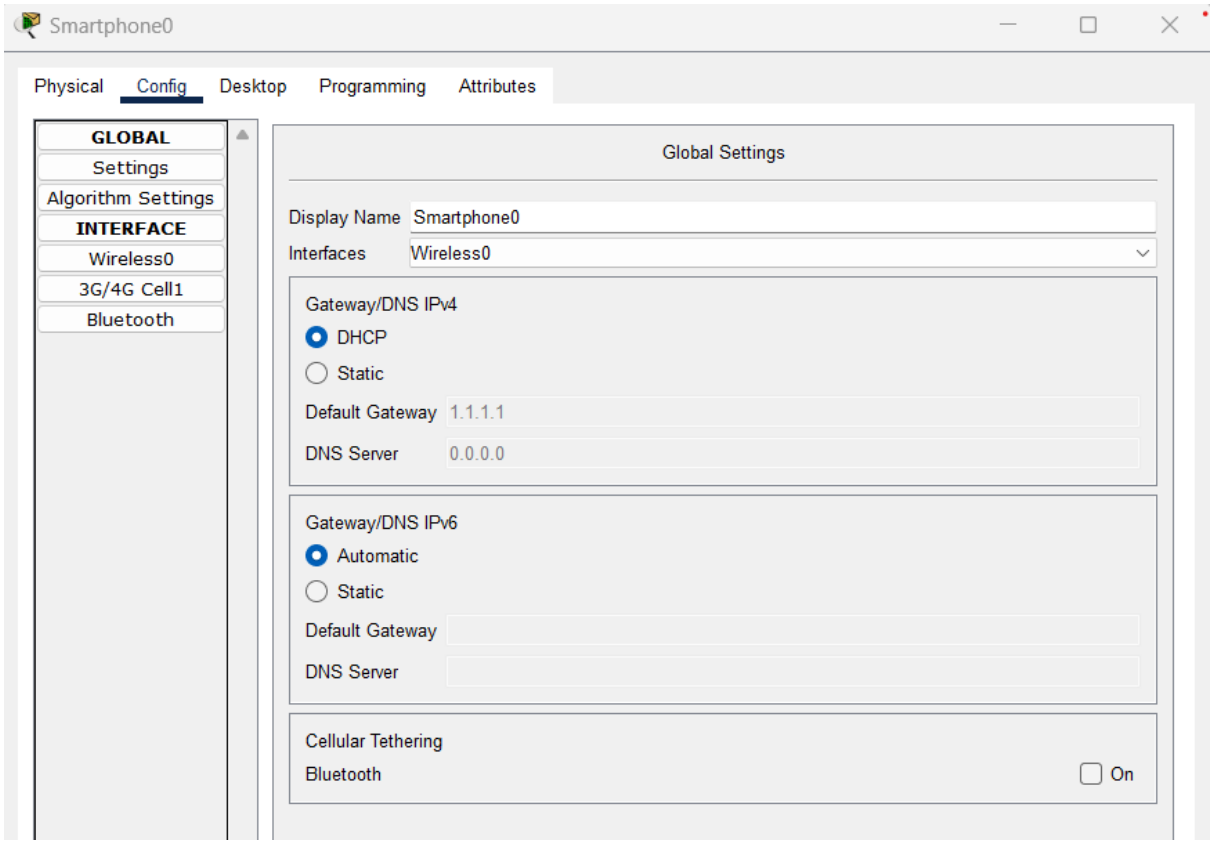


Via the Desktop Interface on the Laptop, open the IoT monitor and login via the Home Gateway Address.

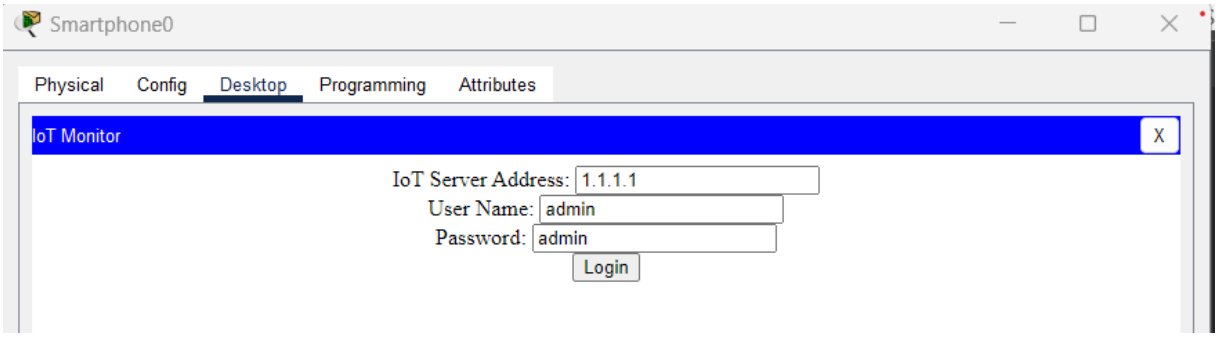


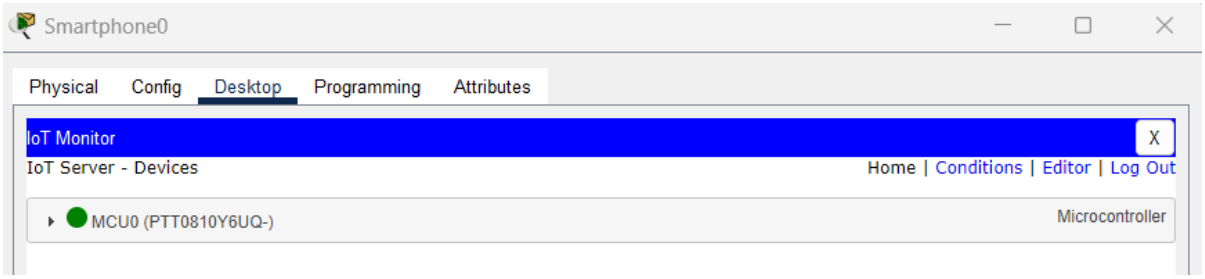


- **Smartphone:**
  Make the following configuration to the smartphone.



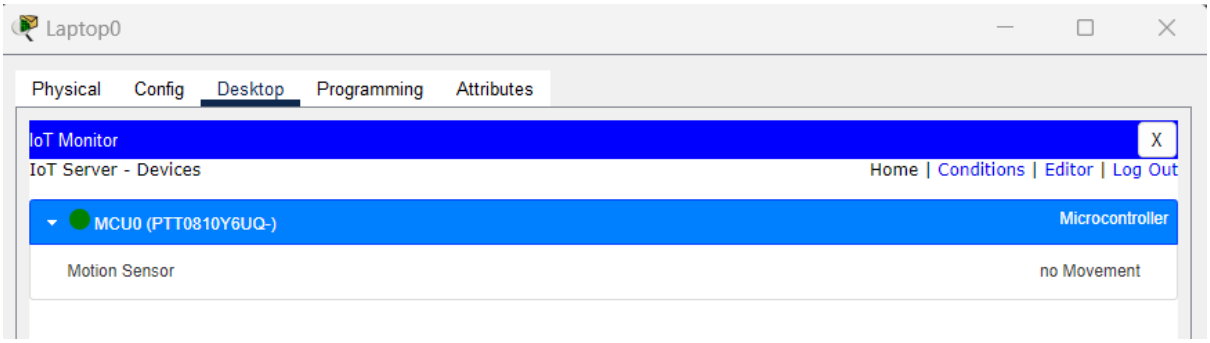
Via the Desktop Interface on the Laptop, open the IoT monitor and login via the Home Gateway Address.



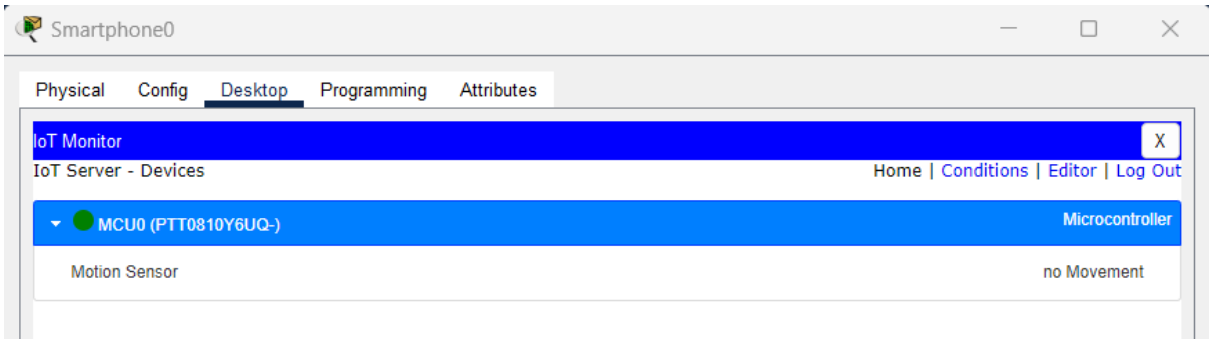


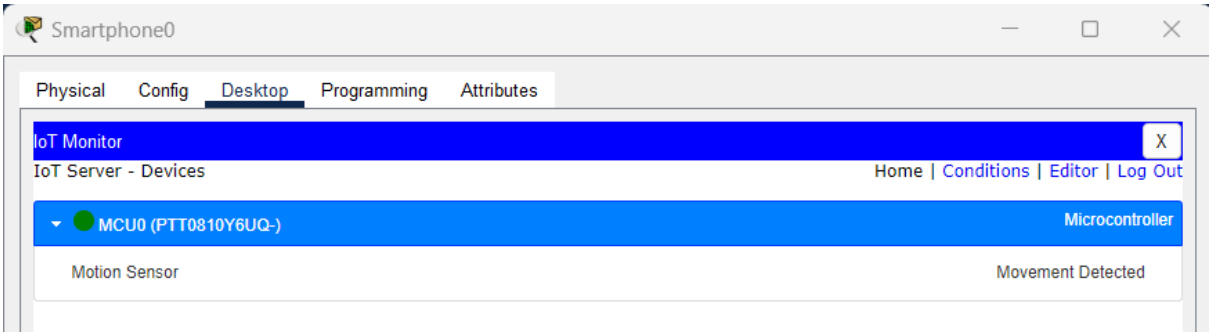
Step 2: Check whether the motion sensor works by observing the IoT monitor on the Laptop and the Smartphone.

- **Laptop:**

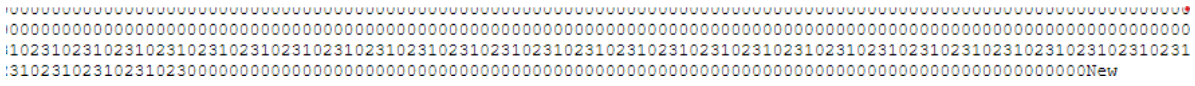


- **Smartphone:**





On the console of the Microcontroller, when there is no movement the output shown is ‘0’ and when there is movement the output shown is ‘1023’.



Name: Avinash Rajkumar Kauran, Roll No: CS22006
Course: Embedded Internet Of Things

PRACTICAL 9

Aim: Deploy an IoT application that will alert you by beeping or vibrating your phone whenever you get someone to call your name.

Theory:

In this practical, we will use a vibration sensor and virtual terminal which are connected to Arduino Uno, here we will send the input through the logic toggle and will get the output through the virtual terminal.

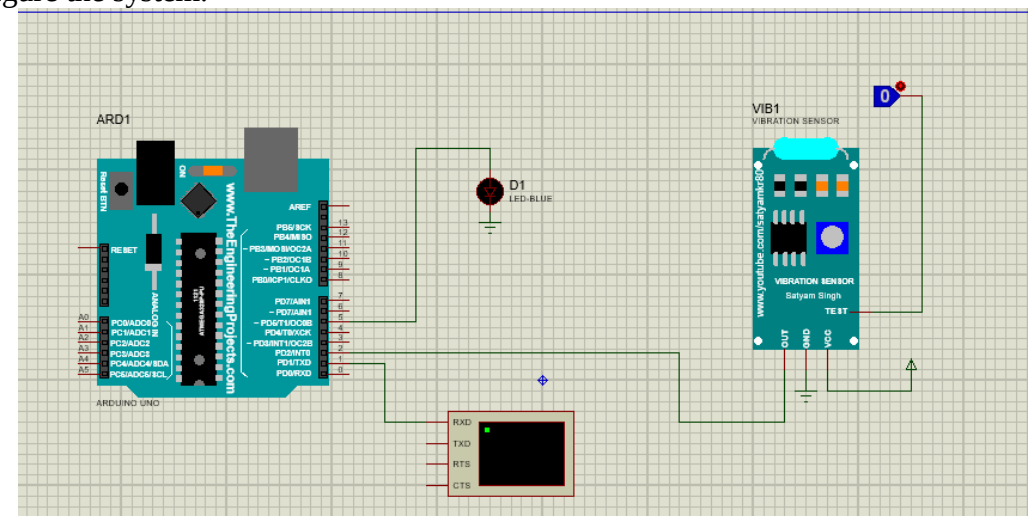
The virtual terminal in Proteus 8 provides a text-based interface where you can send commands or data to the microcontroller and receive its response or output. This feature is handy when working with microcontrollers that communicate through serial protocols like UART (Universal Asynchronous Receiver-Transmitter) or when you want to test and debug your code by observing the data being sent or received.

Requirements:

- Arduino Uno
- Logic Toggle
- Vibration Sensor(V1B1)
- Virtual Terminal
- Led-Blue
- Ground
- Power

Steps :

Step 1: Download all the Vibration Sensor Libraries for Proteus to use and then paste the libraries into the Library Folder of Proteus8.
Step 2: 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.

vibration | Arduino 1.8.18

File Edit Sketch Tools Help

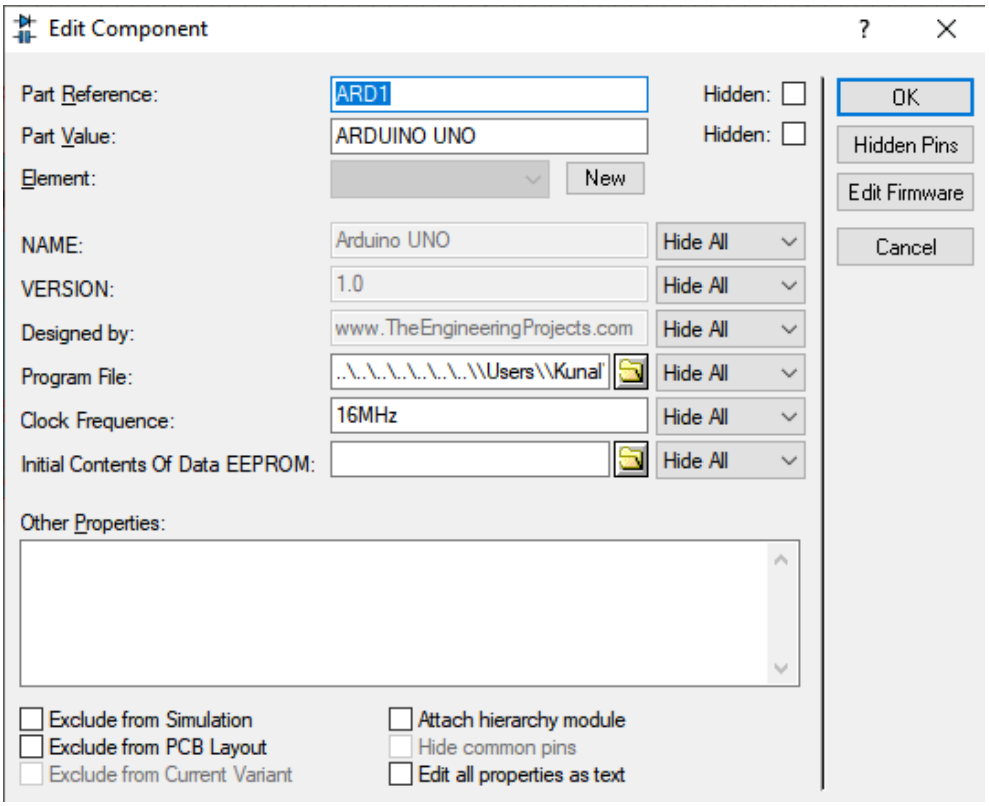
vibration

```
int bl =2;
int dl =5;

int cnt=0,cnt2;
int timer=0;

int pos = 0;
void setup() {
  Serial.begin(9600);
  pinMode(bl, INPUT_PULLUP);
  pinMode(dl, OUTPUT);
  digitalWrite(dl, HIGH);
  digitalWrite(dl, LOW);
  delay(300);
  cnt=0;
}

void loop() {
  if(digitalRead(bl)==HIGH){
    Serial.println("Someone is Calling! VIBRATION ALERT");

    digitalWrite(dl, HIGH);
    delay(300);
    digitalWrite(dl, LOW);
    delay(300);

        digitalWrite(dl, HIGH);
        delay(300);
        digitalWrite(dl, LOW);
        delay(300);

        digitalWrite(dl, HIGH);
        delay(300);
        digitalWrite(dl, LOW);
        delay(300);

        }

  }
```

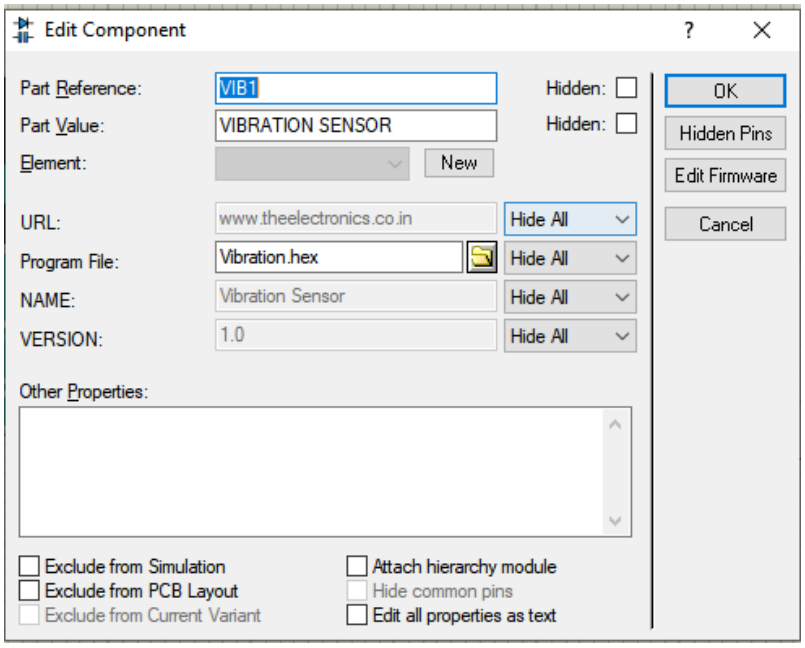
Note: Make sure that you compile the code. The generated hex file for proteus will be generated in the location.

(..\..\..\..\..\..\..\\Users\\Kunal\\AppData\\Local\\Temp\\arduino_build_385761/vibration.ino.hex)

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

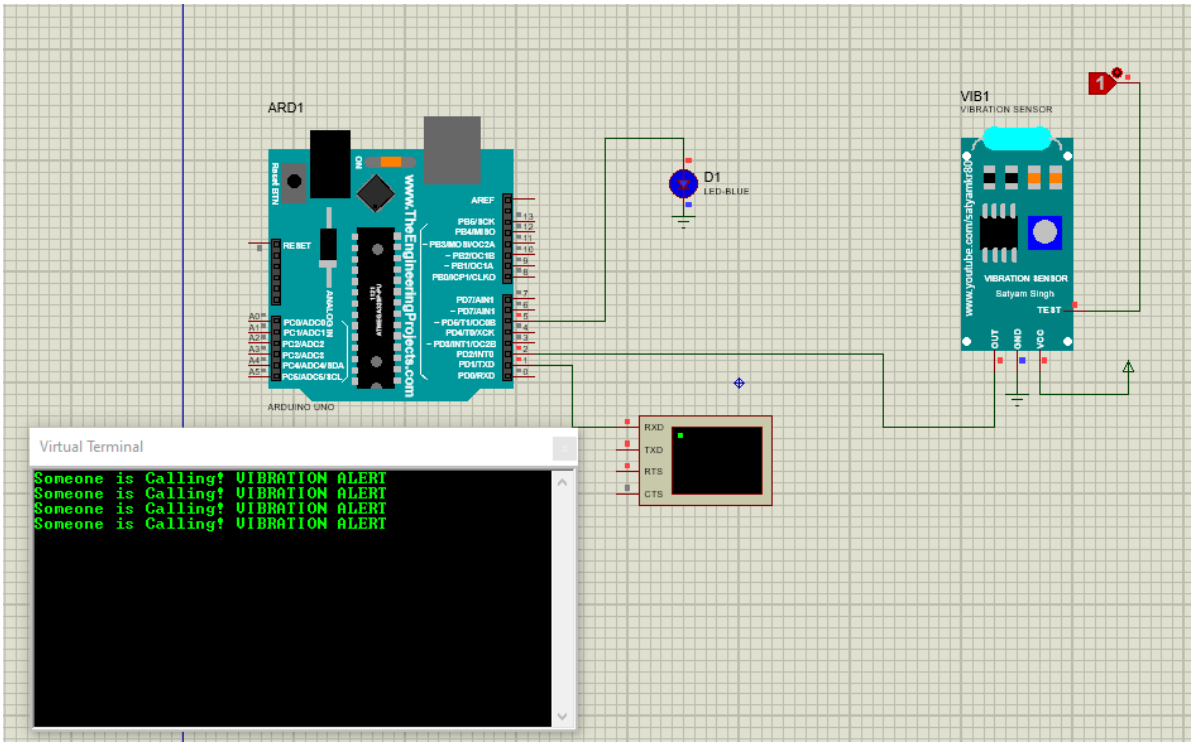


Step 5: Add the Provided hex file of the Vibration Sensor to the Vibration Sensor.



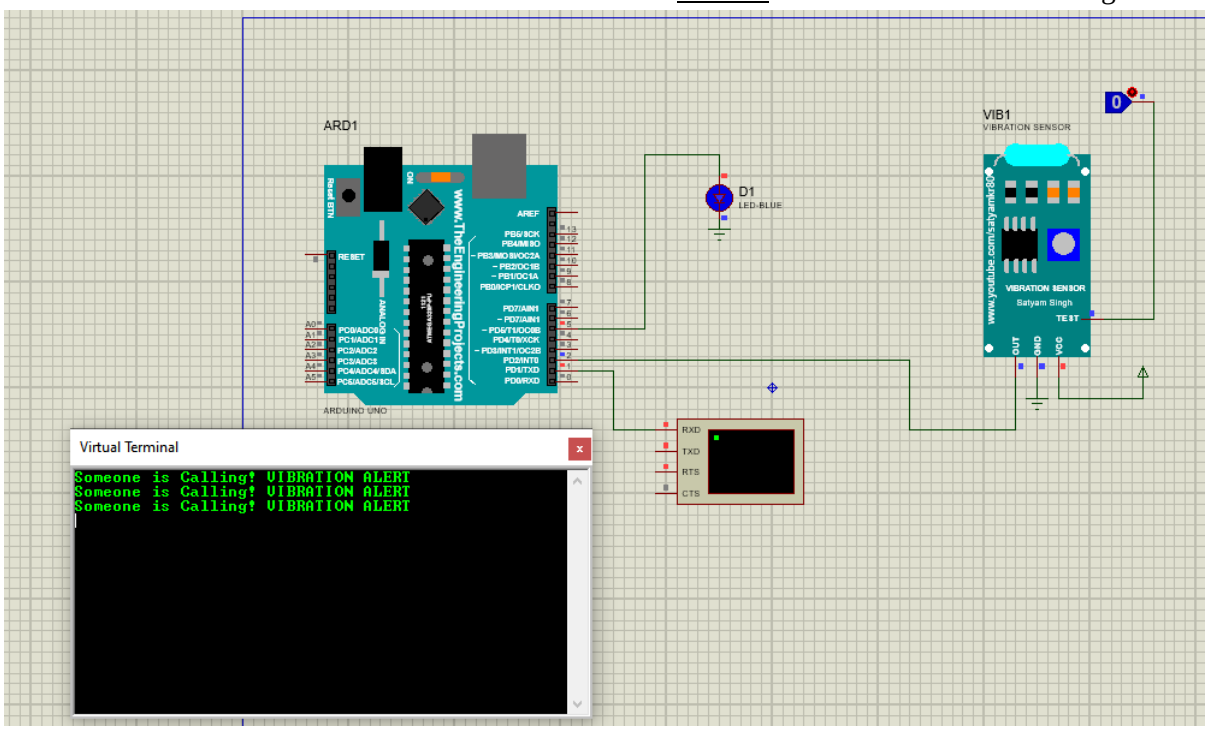
Step 6: Run the simulation and make your observations.

When we run the simulation, we press the logic toggle for the input and once the input is passed you can see the blue led light is blinking and the text is displayed on the virtual Terminal.



And as stop the input of the logic toggle the light stops blinking and the virtual terminal stops giving us alert

Name: Avinash Rajkumar Kauran, Roll No: CS22006
Course: Embedded Internet Of Things



Thus we have an IOT Module which gives us signal when someone is calling us.

Name: Avinash Rajkumar Kauran, Roll No: CS22006
Course: Embedded Internet Of Things

PRACTICAL 10

Aim: Develop an IOT application in to demonstrate TIC TAC TOE game

Game Rules

1. Traditionally the first player plays with "X". So you can decide who wants to go with "X" and who wants to go with "O".
2. Only one player can play at a time.
3. If any of the players have filled a square then the other player and the same player cannot override that square.
4. There are only two conditions that may match will be draw or may win.
5. The player that succeeds in placing three respective marks (X or O) in a horizontal, vertical, or diagonal row wins the game.

Winning condition

Whoever places three respective marks (X or O) horizontally, vertically, or diagonally will be the winner.

```
import os
import time

board = [' ',' ',' ',' ',' ',' ',' ',' ',' ',' ']
player = 1

########win Flags##########
Win = 1
Draw = -1
Running = 0
Stop = 1
############################
Game = Running
Mark = 'X'

#This Function Draws Game Board
def DrawBoard():
    print(" %c | %c | %c " % (board[1],board[2],board[3]))
    print("___|___|___")
    print(" %c | %c | %c " % (board[4],board[5],board[6]))
    print("___|___|___")
    print(" %c | %c | %c " % (board[7],board[8],board[9]))
    print("   |   |   ")

#This Function Checks position is empty or not
def CheckPosition(x):
    if(board[x] == ' '):
        return True
    else:
        return False

```

```
32. #This Function Checks player has won or not
```

```
33. def CheckWin():
34.     global Game
35.     #Horizontal winning condition
36.     if(board[1] == board[2] and board[2] == board[3] and board[1] != ' ')
    :
37.         Game = Win
38.     elif(board[4] == board[5] and board[5] == board[6] and board[4] != '
    '):
39.         Game = Win
40.     elif(board[7] == board[8] and board[8] == board[9] and board[7] != '
    '):
41.         Game = Win
42.     #Vertical Winning Condition
43.     elif(board[1] == board[4] and board[4] == board[7] and board[1] != '
    '):
44.         Game = Win
45.     elif(board[2] == board[5] and board[5] == board[8] and board[2] != '
    '):
46.         Game = Win
47.     elif(board[3] == board[6] and board[6] == board[9] and board[3] != '
    '):
48.         Game=Win
49.     #Diagonal Winning Condition
50.     elif(board[1] == board[5] and board[5] == board[9] and board[5] != '
    '):
51.         Game = Win
52.     elif(board[3] == board[5] and board[5] == board[7] and board[5] != '
    '):
53.         Game=Win
54.     #Match Tie or Draw Condition
55.     elif(board[1]!=' ' and board[2]!=' ' and board[3]!=' ' and board[4]!=
    ' ' and board[5]!=' ' and board[6]!=' ' and board[7]!=' ' and board[8]!=' '
     and board[9]!=' '):
56.         Game=Draw
57.     else:
58.         Game=Running
59.
60. print("Tic-Tac-Toe Game Designed ANISH MALIK SHARMA")
61. print("Player 1 [X] --- Player 2 [O]\n")
62. print()
63. print()
64. print("Please Wait...")
65. time.sleep(3)
66. while(Game == Running):
67.     os.system('cls')
68.     DrawBoard()
69.     if(player % 2 != 0):
70.         print("Player 1's chance")
71.         Mark = 'X'
```

```
72.     else:
```

```
        print("Player 2's chance")
        Mark = 'O'
    choice = int(input("Enter the position between [1-9] where you want to mark : "))
    if(CheckPosition(choice)):
        board[choice] = Mark
        player+=1
        CheckWin()

os.system('cls')
DrawBoard()
if(Game==Draw):
    print("Game Draw")
elif(Game==Win):
    player-=1
    if(player%2!=0):
        print("Player 1 Won")
    else:
        print("Player 2 Won")
```

OUTPUT:

Here the player have used the numbers to place x or 0 in the 3x3 matrix

```
Player 1 [X] --- Player 2 [O]



Please Wait...
   |   |
___|___|___
   |   |
___|___|___
   |   |
   |   |
Player 1's chance
Enter the position between [1-9] where you want to mark :  3
   |   | X
___|___|___
   |   |
___|___|___
   |   |
   |   |
Player 2's chance
Enter the position between [1-9] where you want to mark :  5
   |   | X
___|___|___
   | O |
___|___|___
   |   |
   |   |
Player 1's chance
Enter the position between [1-9] where you want to mark :  9
```

```
   |   | X
___|___|___
   | O |
___|___|___
   |   | X
   |   |
Player 2's chance
Enter the position between [1-9] where you want to mark :  7
   |   | X
___|___|___
   | O |
___|___|___
 O |   | X
   |   |
Player 1's chance
Enter the position between [1-9] where you want to mark :  1
 X |   | X
___|___|___
   | O |
___|___|___
 O |   | X
   |   |
Player 2's chance
Enter the position between [1-9] where you want to mark :  2
 X | O | X
___|___|___
   | O |
___|___|___
 O |   | X
   |   |
Player 1's chance
Enter the position between [1-9] where you want to mark :  6
Enter the position between [1-9] where you want to mark :  6
 X | O | X
___|___|___
   | O | X
___|___|___
 O |   | X
   |   |
Player 1 Won
```