# UNIVERSITY OF MUMBAI

DEPARTMENT OF COMPUTER SCIENCE

M.SC (Computer Science)

## CERTIFICATE

Certified that the work entered in this journal was done in the computer laboratory by the student

**Ms. Komal Gangurde**

**Roll No.20** of the Class – **MSc computer Science First Year Sem-II**

Semester during the year **2021-2022** in a Satisfactory Manner

# Embedded and IoT Technology

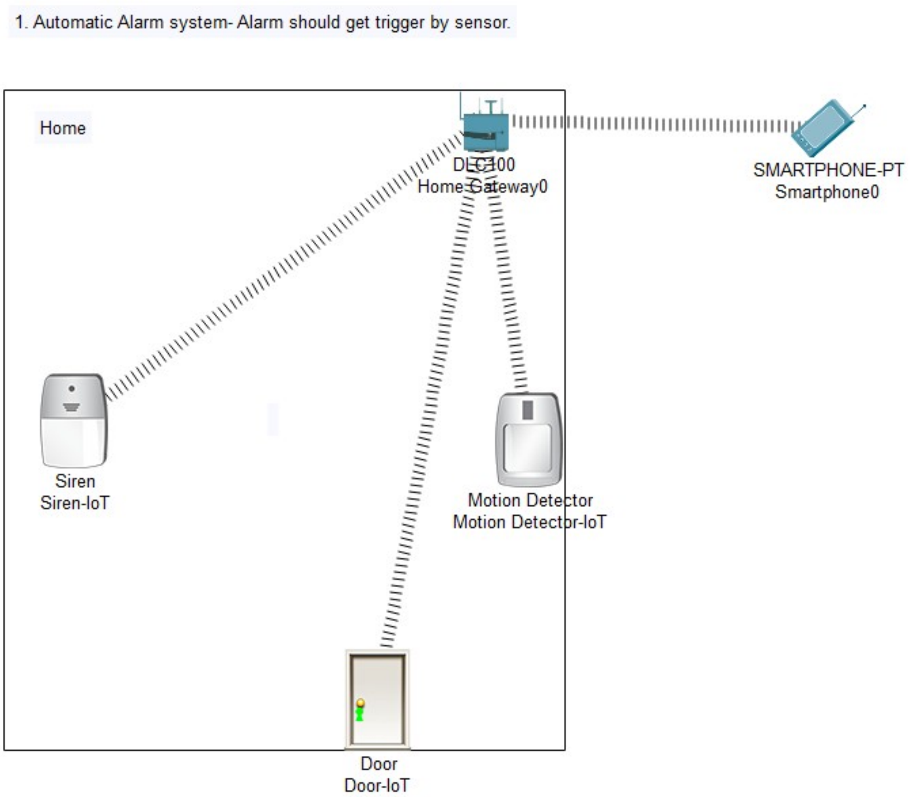
# Practical No:01

**Aim: Design and implement basics embedded circuits.**
Components:- Siren, HomeGateway, Motion Detector, Door, Smartphone.

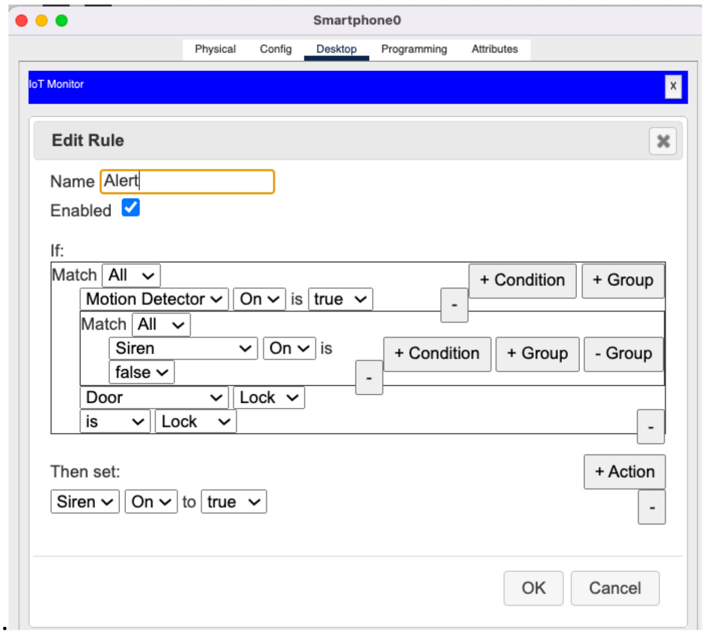
**Theory:** A system designed with the embedding of hardware and software together for a specific function with a larger area is embedded system design. In this Practical Implementing basic Embeded system to perform following task such as Alarm system and Time buzzer etc.
**1.** Automatic Alarm system-
Alarm should get trigger by sensor: in this Practical using
Home Gateway, The main functions of Home Gateway As a physical interface that connects all external access networks to the home and connects the home internal network to the outside.In Automatic Alarm system Motion detector detect motion and trigger to Siren when door is open.
2. Timer based buzzer -
Connecting MCU with Toggle Push Button and MCU with IoT Cables. When we tap on toggle its trigger the alarm and we get indication.
3. Sensor based Counting device - this device count the number of Move your mouse over trip sensor while holding alt. And the output will be displayed on the command line.

## 1.Automatic Alarm system- Alarm should get trigger by sensor.

Step 1: Add all the components in the cisco packet tracer.



Step 2: Click on Conditions and Add condition where if Motion Detector on is true and Door Lock is lock then set Siren on to true.



Edit Rule

Name Alert

Enabled ☑

If:

Match All

Motion Detector On is true

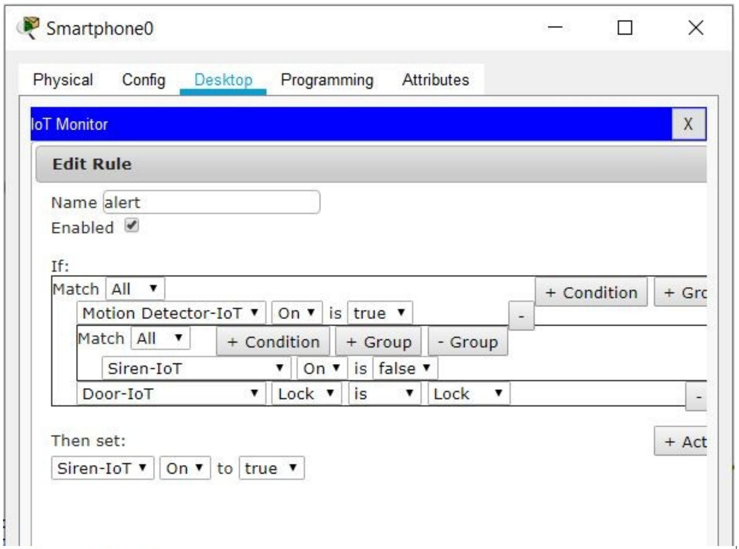
Match All

Siren On is false

Door Lock is Lock

Then set:

Siren On to true



Edit Rule

Name alert

Enabled ☑

If:

Match All

Motion Detector-IoT On is true

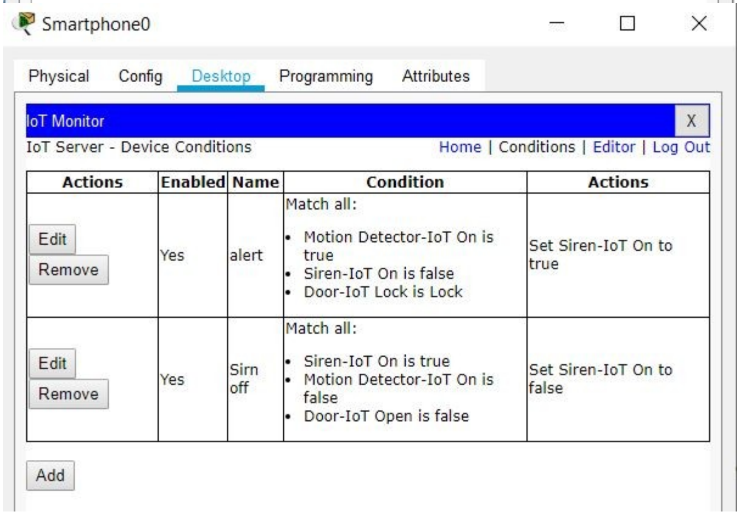
Match All

Siren-IoT On is false

Door-IoT Lock is Lock

Then set:

Siren-IoT On to true

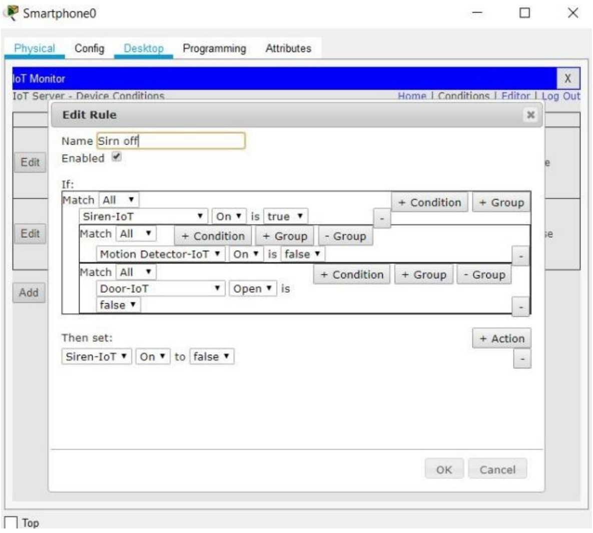


IoT Server - Device Conditions

Home | Conditions | Editor | Log Out

| Actions | Enabled | Name | Condition | Actions |
|---|---|---|---|---|
| Edit<br>Remove | Yes | alert | Match all:<br>• Motion Detector-IoT On is true<br>• Siren-IoT On is false<br>• Door-IoT Lock is Lock | Set Siren-IoT On to true |
| Edit<br>Remove | Yes | Sirn off | Match all:<br>• Siren-IoT On is true<br>• Motion Detector-IoT On is false<br>• Door-IoT Open is false | Set Siren-IoT On to false |

Add



IoT Server - Device Conditions

Edit Rule

Name Sirn off

Enabled ☑

If:

Match All ▾

Siren-IoT ▾ On ▾ is true ▾

Match All ▾

Motion Detector-IoT ▾ On ▾ is false ▾

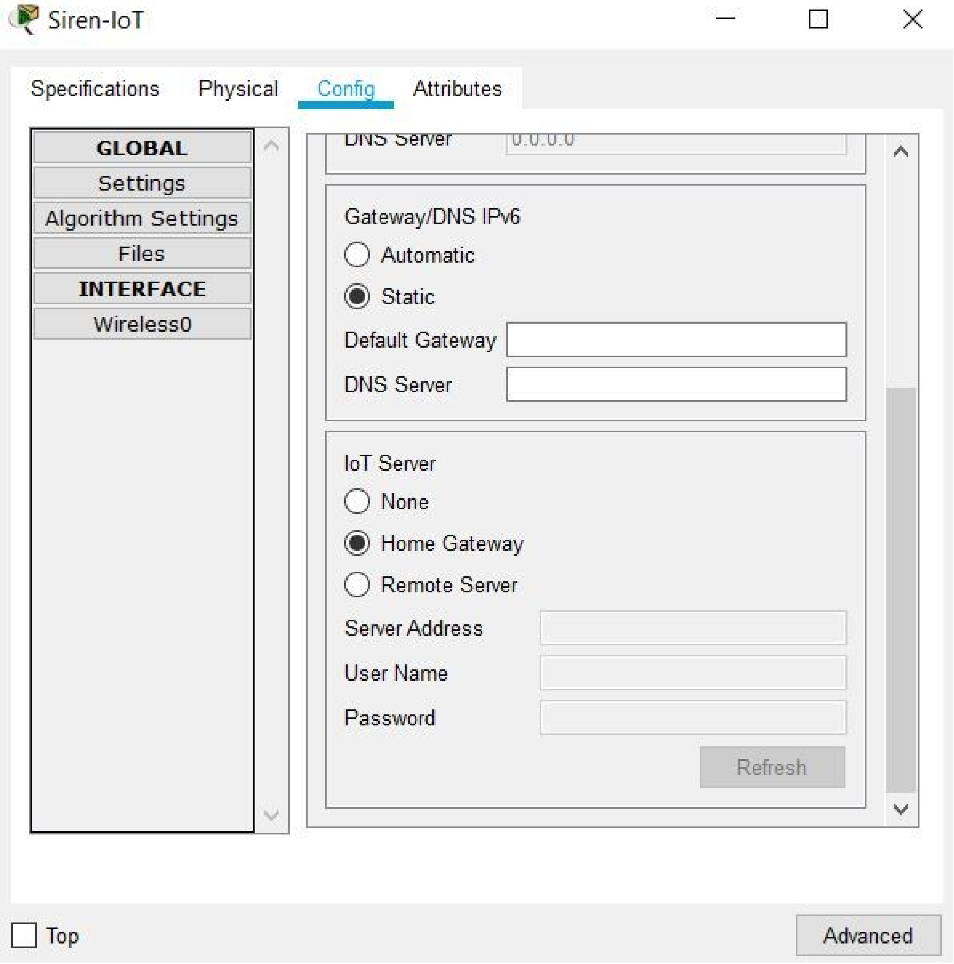
Match All ▾

Door-IoT ▾ Open ▾ is false ▾

Then set:

Siren-IoT ▾ On ▾ to false ▾

Step 3: Click on the component and set its SSID to HomeGateway and IoT Server to HomeGateway.



DNS Server 0.0.0.0

Gateway/DNS IPv6

○ Automatic

● Static

Default Gateway

DNS Server

IoT Server

○ None

● Home Gateway

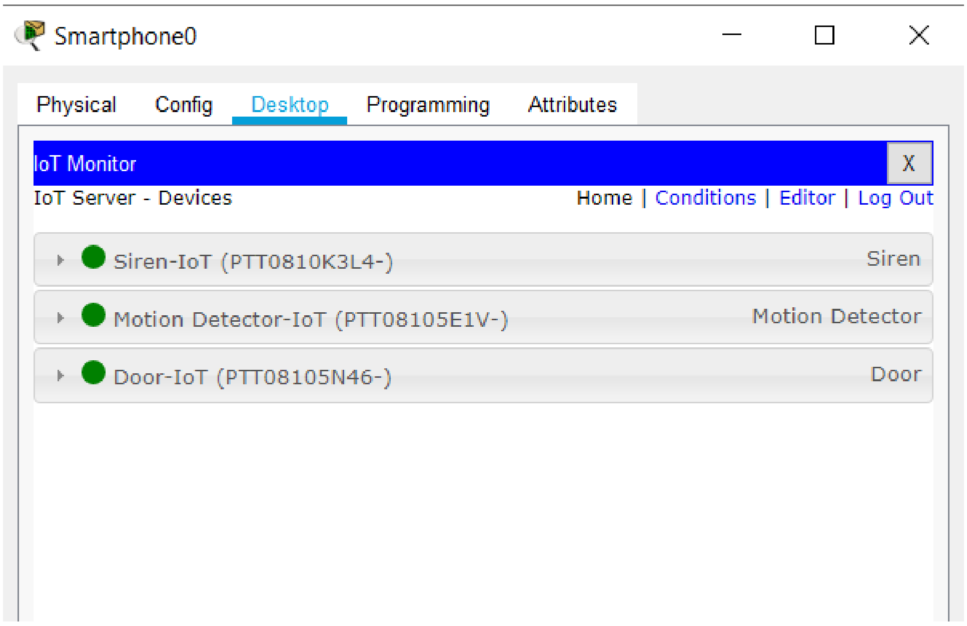
○ Remote Server

Server Address

User Name

Password

Step 4: All the devices connected to the HomeGateway will be displayed.



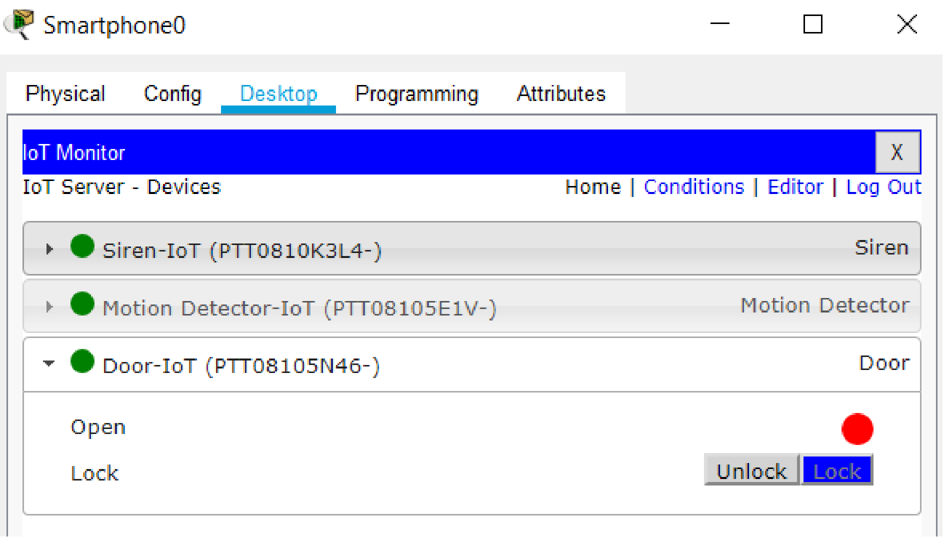
IoT Server - Devices

● Siren-IoT (PTT0810K3L4-) Siren

● Motion Detector-IoT (PTT08105E1V-) Motion Detector

● Door-IoT (PTT08105N46-) Door

Step 5: Make sure Door is locked. Check that in IoT monitor.



IoT Server - Devices

● Siren-IoT (PTT0810K3L4-) Siren

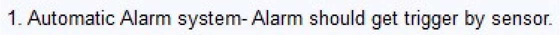
● Motion Detector-IoT (PTT08105E1V-) Motion Detector

● Door-IoT (PTT08105N46-) Door

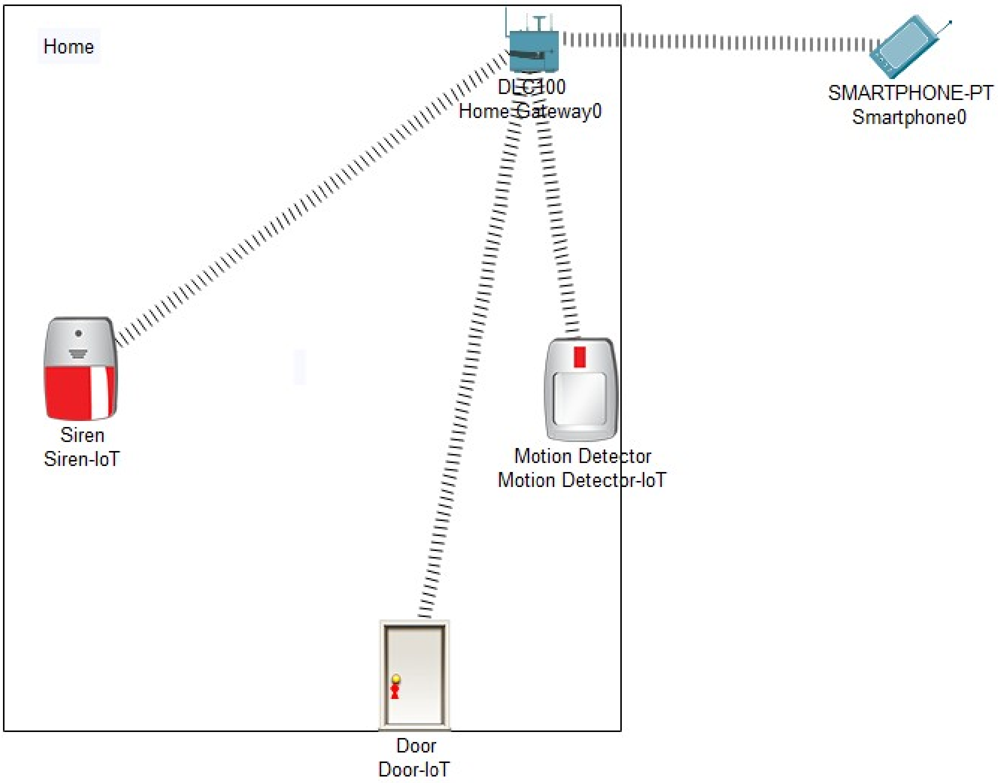
Open ●

Lock

Step 6: Test the alert by hovering the mouse over the motion detector sensor while holding alt on keyboard. The siren should go off when motion detector senses motion.



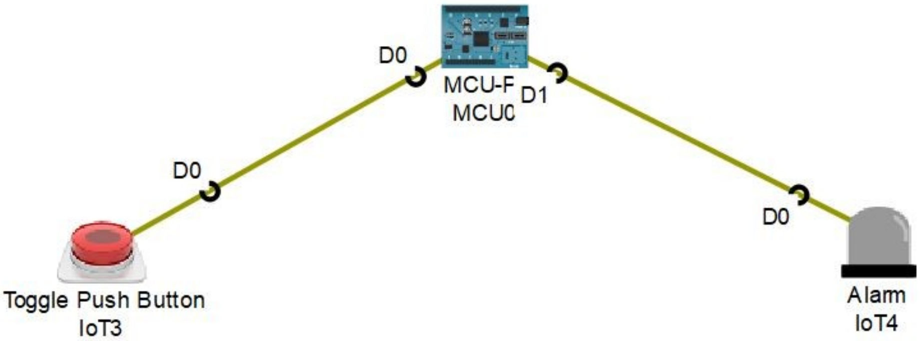
1. Automatic Alarm system- Alarm should get trigger by sensor.



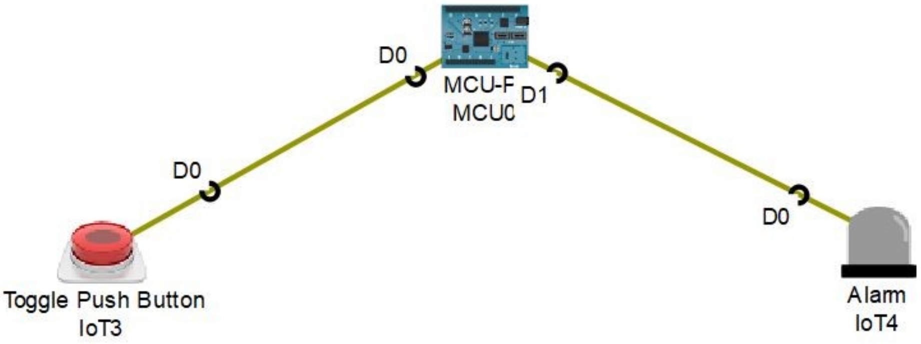
## 2. Timer based buzzer

Components:- Toggle Push Button, MCU Board, Alarm, IoT cables.
Steps:-

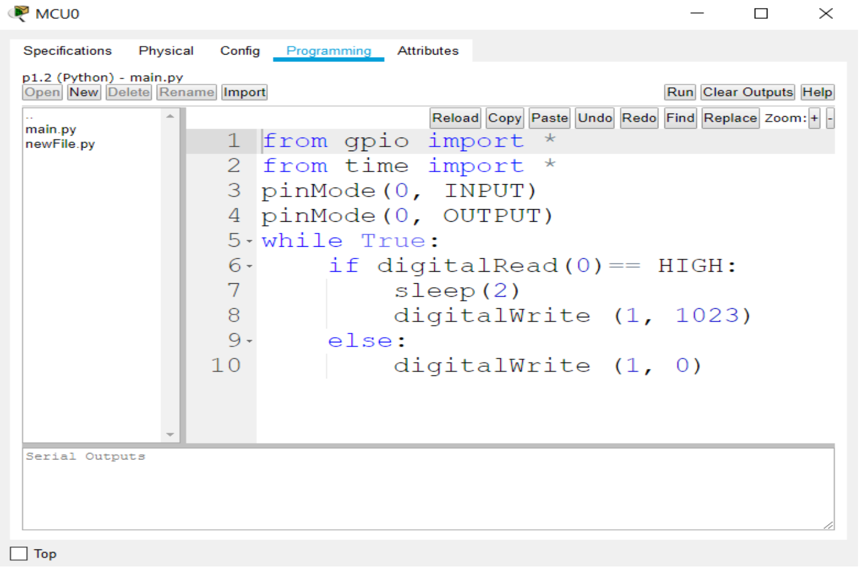
Step 1: Add all the components in the cisco packet tracer.



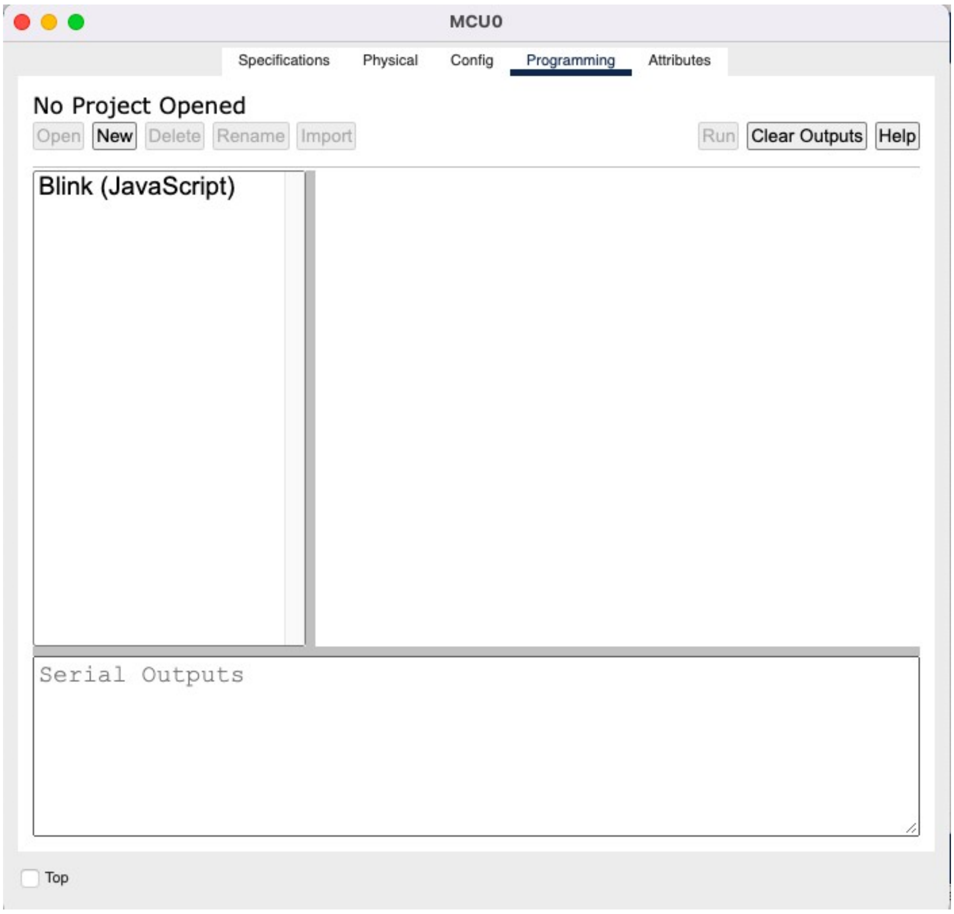
Step 2: Connect MCU with Toggle Push Button and MCU  with IoT Cables.



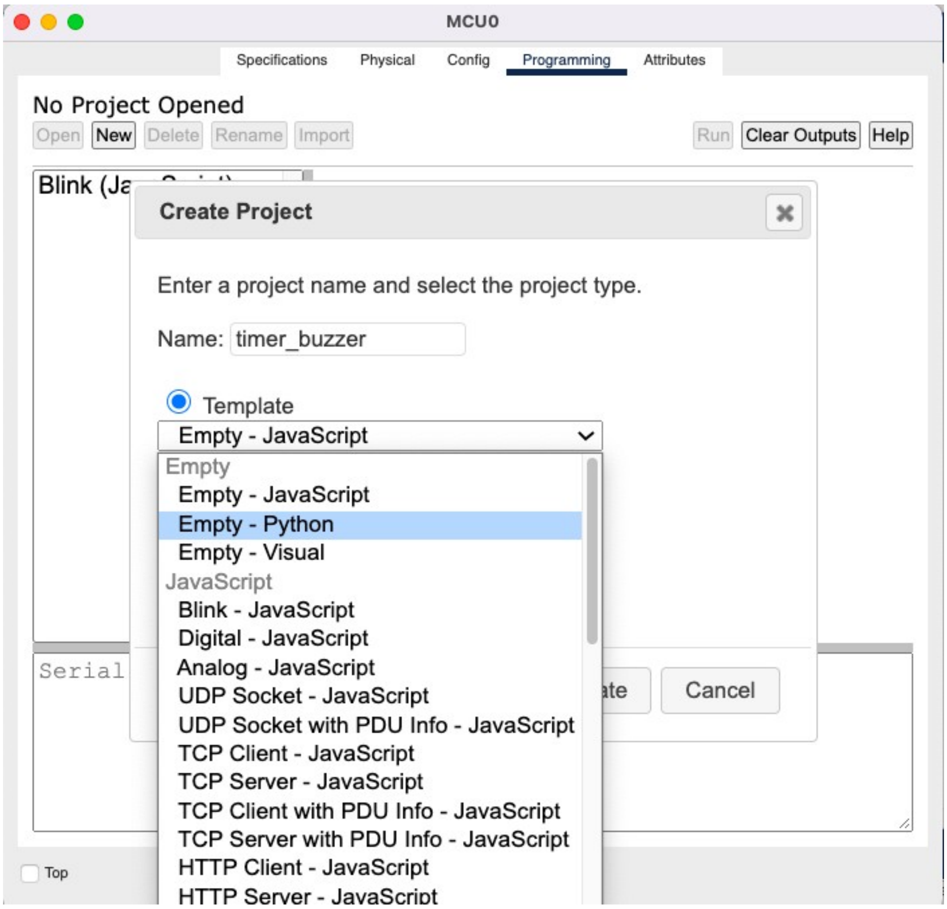
Step 3: Click on MCU and then select Programming.



```
from gpio import *
from time import *
pinMode(0, INPUT)
pinMode(0, OUTPUT)
while True:
    if digitalRead(0)== HIGH:
        sleep(2)
        digitalWrite (1, 1023)
    else:
        digitalWrite (1, 0)
```



No Project Opened



No Project Opened

Create Project

Enter a project name and select the project type.

Name: timer_buzzer

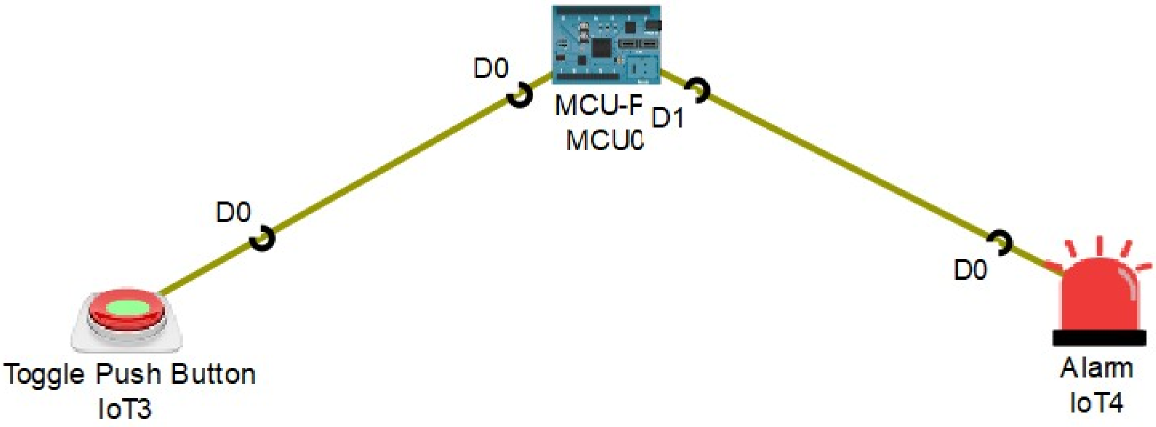
● Template

Empty - JavaScript

Code:

```
from gpio import *
from time import *
pinMode(0, INPUT)
pinMode(0, OUTPUT)
while True:
        if digitalRead(0)== HIGH:
                sleep(2)
                digitalWrite (1, 1023)
        else:
                digitalWrite (1, 0)
```

Step 7: Click Run and Push Toggle Button by hovering mouse over it and clicking it with holding alt key on keyboard.

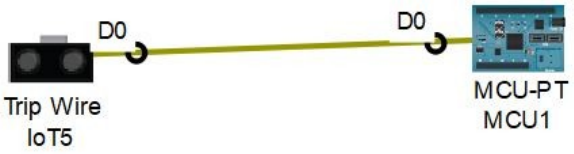


## 3. Sensor based Counting device

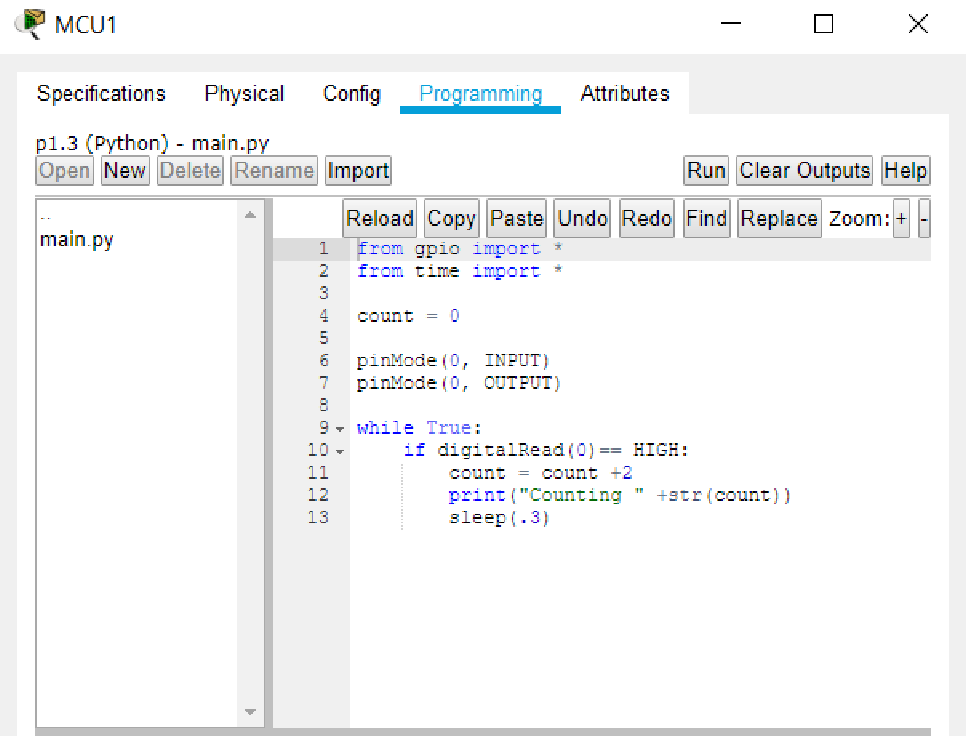
Components:- Trip Wire, MCU Board, IoT cables.

Steps:-

Step 1: Click MCU board and go to Programming.



Step 2: Click on New and write following code and run.

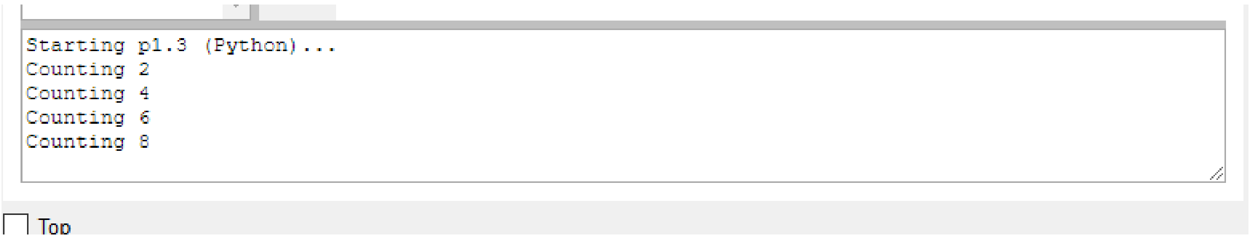


```
from gpio import *
from time import *

count = 0

pinMode(0, INPUT)
pinMode(0, OUTPUT)

while True:
    if digitalRead(0)== HIGH:
        count = count +2
        print("Counting " +str(count))
        sleep(.3)
```

Code:

```
from gpio import *
from time import *

count = 0

pinMode(0, INPUT)
pinMode(0, OUTPUT)

while True:
        if digitalRead(0)== HIGH:
                count = count +2
                print("Counting " +str(count))
                sleep(.3)
```

Step 4: Move your mouse over trip sensor while holding alt. And the output will be displayed on the command line.



```
Starting p1.3 (Python)...
Counting 2
Counting 4
Counting 6
Counting 8
```

# Practical No:02

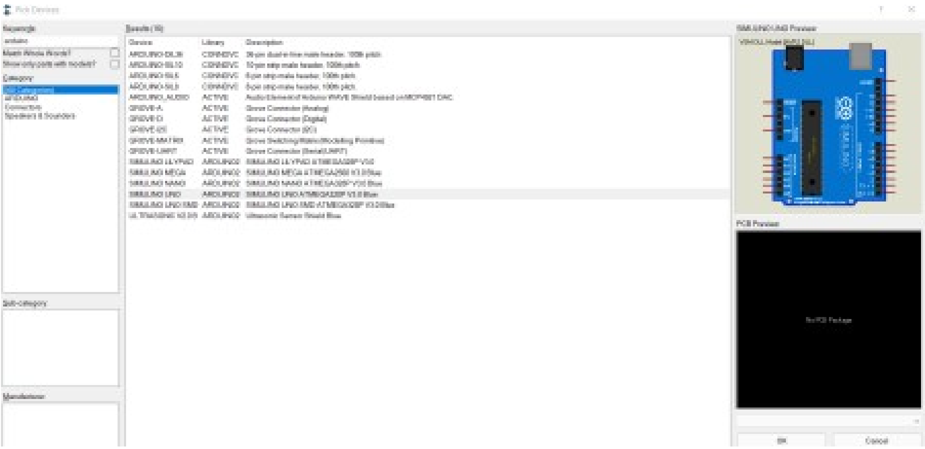
**AIM**: Demonstrate communication between two embedded devices using UART port.

## Theory: -

UART (stands for **Universal Asynchronous Reception and Transmission**). It is a simple serial communication protocol that allows the host to communicate with the other devices. It has two data lines, one to transmit (TX) and another to receive (RX). For different embedded devices, you will need to check on the datasheets to understand which pins can be used for UART.

**STEPS TO PERFORM THIS PRACTICAL:**

1. Click on pick device icon and select the following



Pick Devices

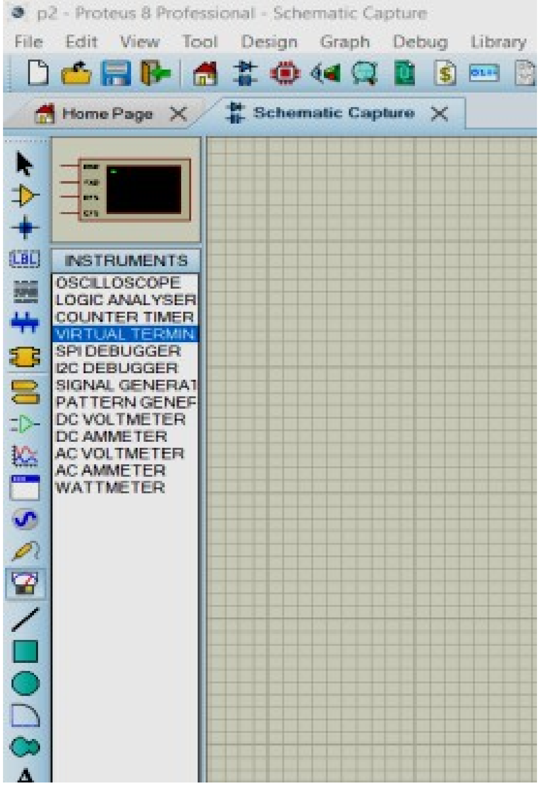
Keywords: arduino

Category: (All Categories), ARDUINO, Connectors, Speakers & Sounders

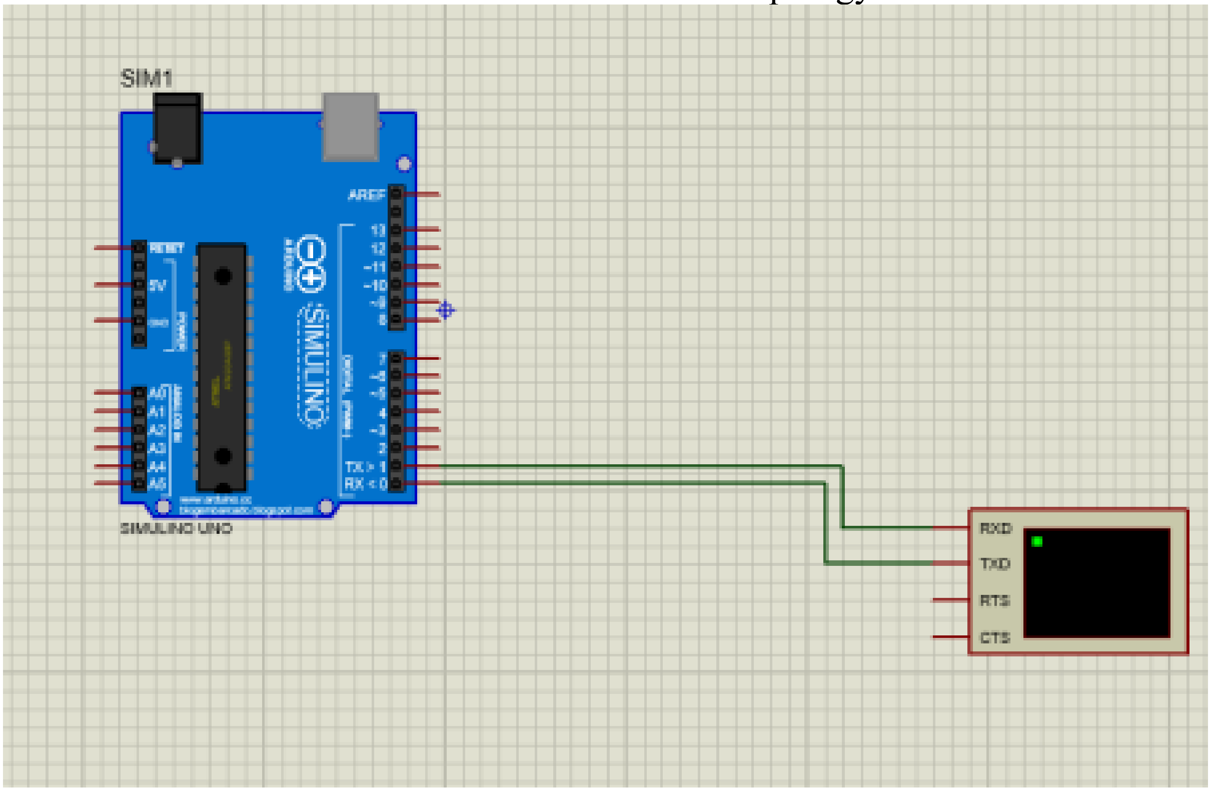
Results (16)

| Device | Library | Description |
|---|---|---|
| ARDUINO-DIL36 | CONNDVC | 36-pin dual-in-line male header, 100th pitch |
| ARDUINO-SIL10 | CONNDVC | 10-pin strip male header, 100th pitch |
| ARDUINO-SIL8 | CONNDVC | 8-pin strip male header, 100th pitch |
| ARDUINO-SIL6 | CONNDVC | 6-pin strip male header, 100th pitch |
| ARDUINO_AUDIO | ACTIVE | [illegible] |
| GROVE-A | ACTIVE | Grove Connector (Analog) |
| GROVE-D | ACTIVE | Grove Connector (Digital) |
| GROVE-I2C | ACTIVE | Grove Connector (I2C) |
| GROVE-MATRIX | ACTIVE | [illegible] |
| GROVE-UART | ACTIVE | Grove Connector (Serial/UART) |
| SIMULINO LLYPAD | ARDUINO2 | [illegible] |
| SIMULINO MEGA | ARDUINO2 | SIMULINO MEGA ATMEGA2560 [illegible] |
| SIMULINO NANO | ARDUINO2 | SIMULINO NANO ATMEGA328P [illegible] |
| SIMULINO UNO | ARDUINO2 | SIMULINO UNO ATMEGA328P V3.0 Blue |
| SIMULINO UNO SMD | ARDUINO2 | SIMULINO UNO SMD ATMEGA328P [illegible] |
| ULTRASONIC V2.0 | ARDUINO2 | Ultrasonic Sensor Shield Blue |

2. Now click on instruments and select virtual terminal.

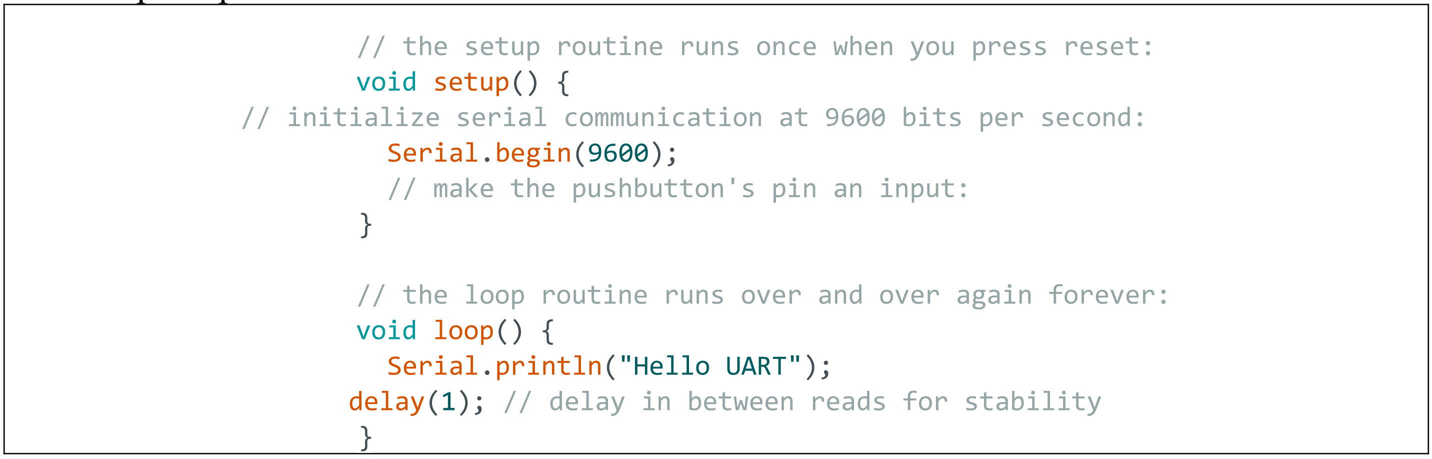


3. Now connect Arduino and virtual terminal the topology will look like this.



4. Now open Arduino idle and do the following:
Click on file menu examples Basic DigitalReadSerial

It will open up the code

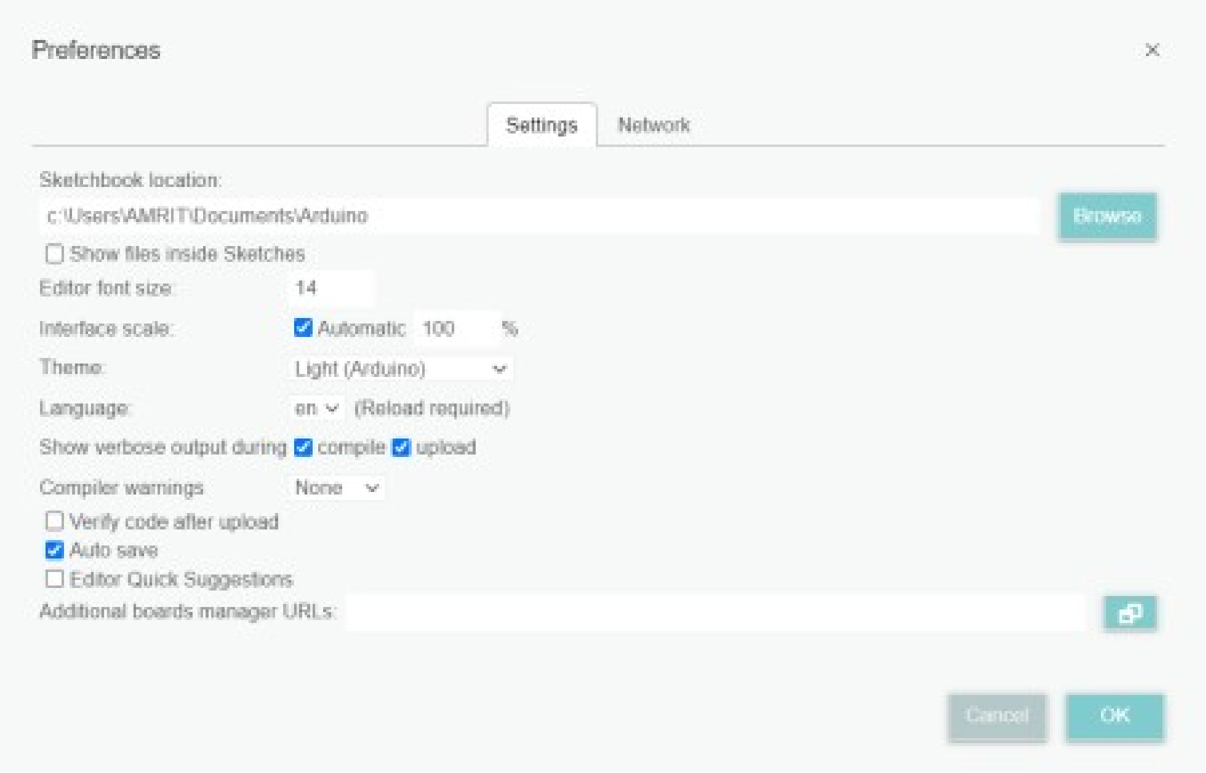


```
            // the setup routine runs once when you press reset:
            void setup() {
        // initialize serial communication at 9600 bits per second:
              Serial.begin(9600);
              // make the pushbutton's pin an input:
            }

            // the loop routine runs over and over again forever:
            void loop() {
              Serial.println("Hello UART");
            delay(1); // delay in between reads for stability
            }
```

Now run this program

And in output it will give .hex file location copy that path

Note: if .hex path is not shown then click on file preferences and select the compile and upload checkboxes and click on ok and re run the code.



Sketchbook location:
c:\Users\AMRIT\Documents\Arduino

☐ Show files inside Sketches

Editor font size: 14

Interface scale: ☑ Automatic 100 %

Theme: Light (Arduino)

Language: en (Reload required)

Show verbose output during ☑ compile ☑ upload

Compiler warnings None

☐ Verify code after upload

☑ Auto save

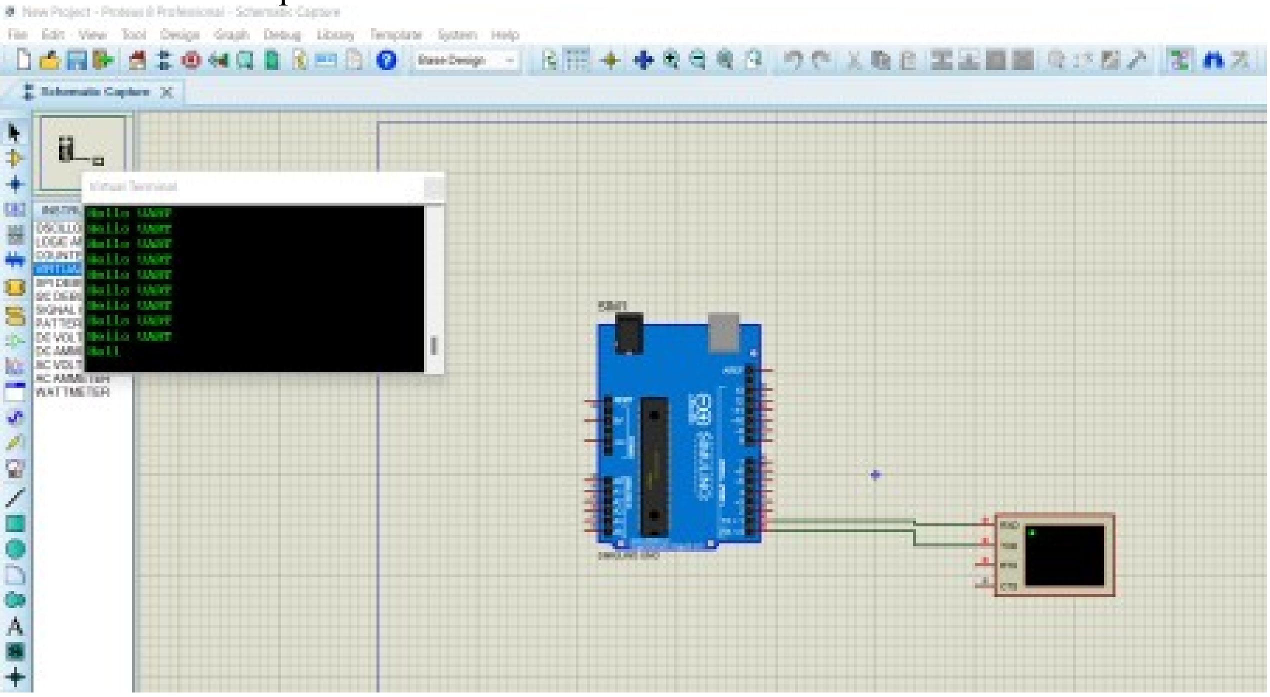
☐ Editor Quick Suggestions

Additional boards manager URLs:

5. NowinProteusclickonArduinoandinprogramfilespastethe .hex filepath

6. OUTPUT:

Now click on play button this will give u an output ie we are connecting two devices on UART port.



```
Hello UART
Hello UART
Hello UART
Hello UART
Hello UART
Hello UART
Hello UART
Hello UART
Hello UART
Hell
```

# Practical No:03

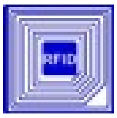
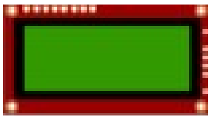
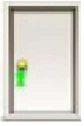
**AIM**: Built an IoT system to send ticket before entering the bus.

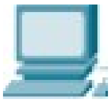
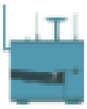
**Theory:**

IoT system to send a ticket before entering the bus is mad on cisco packet tracer and this is inspired from railway ticketing system. Basically, when we travel from metro, we can just take a ticket from Automatic ticket machine using Railway smart card and travel.

In this System user have the card and before entering in the bus ti selects the destination and place the card on sensors present on door and the door opens up and u can travel in this bus.

REQUIREMENTS:

| Component | Name Use |
|---|---|
|  | This card is used as a key to<br>RFID Card<br>enter in the bus.<br>(ACTUATOR). |
| | This component is connected<br>RFID card<br>on the door which reads RFID<br>Reader<br>card.<br>(SENSOR). |
| <br> | This component is used for<br>MCU Board<br>connecting Sensors to<br>Actuators.<br>This are used to interact with<br>humans i.e it displays results<br>LCD Display<br>of actuators. |
|  | The door Opens up when<br>Door<br>RFID card is scanned<br>successfully. |

| | |
|---|---|
|  | This is used to control PC homegateway. |
|  | This is used to connect all the smart devices to the controller HomeGateway device wirelessly. |

1- We have to create front door, which is used whenever the passenger traveling in bus is boarding the bus to reach the destination place. Components used for this part is

- MCU Board
- Door
- RFID Reader
- LCD Display
- Home gateway
- PC
- RFID card

If the person is using the front door that means the person have to pay for the ticket, without which he/she couldn't have entered inside the bus hence coding for this part is simple.

Connect LCD display to MCU board using digital port "D0". Connect RFID Reader to MCU board using Analog port "A1". Connections to MCU board should be done using IOT CUSTOM CABLE ONLY
Connect RFID Reader Wirelessly to HomeGateway.
Connect Door to HomeGateway Wirelessly.
Connect PC also to HomeGateway.

Now on MCU board type the following code:

```
from gpio import *
from time import *
from ioeclient import IoEClient
def main():
pinMode(0,OUT)
pinMode(1,IN)
while True:
customWrite(0,"Waiting");
rfid=analogRead(A1)
if(rfid==0):

customWrite(0,"success");

delay(3000)
if name ==" main
": main()
```

run the code.
Now whenever the RFID card is scanned on RFID reader the signal will be given to HomeGateway by RFID reader that card is Scanned and it check the conditions mentioned on server of HomeGateway for what has to be performed i.e it will open the front door and display Success message on LCD Display.
Open the RFID reader programming part and make the following changes in loop function of main.py

```
if not found:
cardID = lastCardID = 0
setState(1) #the state is changed from 2 to 1. There are 3 stats 0-high-success,1-low -waiting,2-waiting-we are
not using this state.
else:
if lastCardID != cardID:
lastCardID = cardID
sendReport()
if cardID=='1001':

setState(0)# we want to display success message and rfid signal to green
else:
setState(1)# we want to display waiting message and rfid signal to red

delay(DELAY_TIME)
```

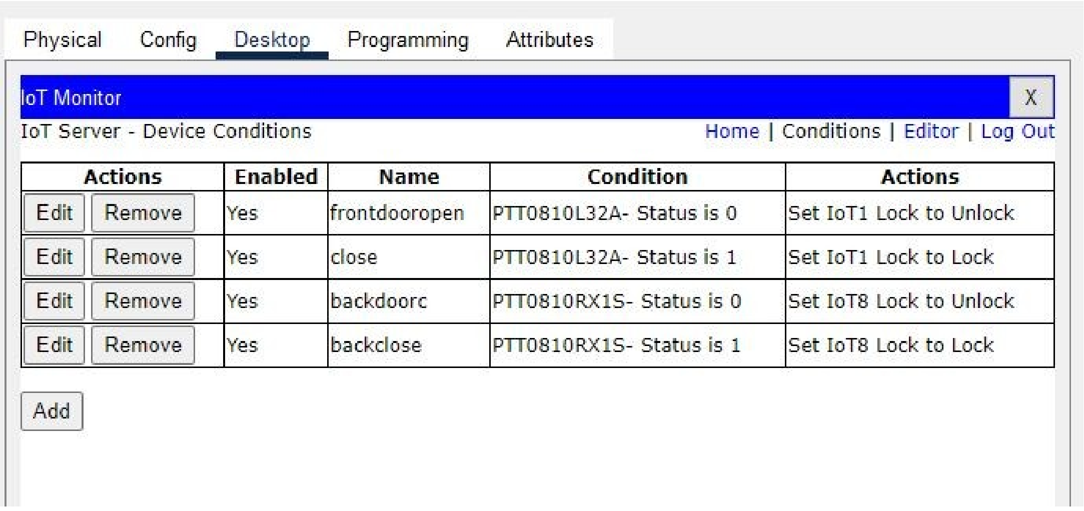
run the code.

Now in RFIDcard do the following changes in programming main.py

```
CARD_ID = restoreProperty('CardID', CARD_ID)
setDeviceProperty(getName(), 'CardID', CARD_ID)
setDeviceProperty(getName(), 'amount', 5000) # a new line is added
```

run the code

now create conditions in PC for opening the door, once you pay the ticket via RFID.



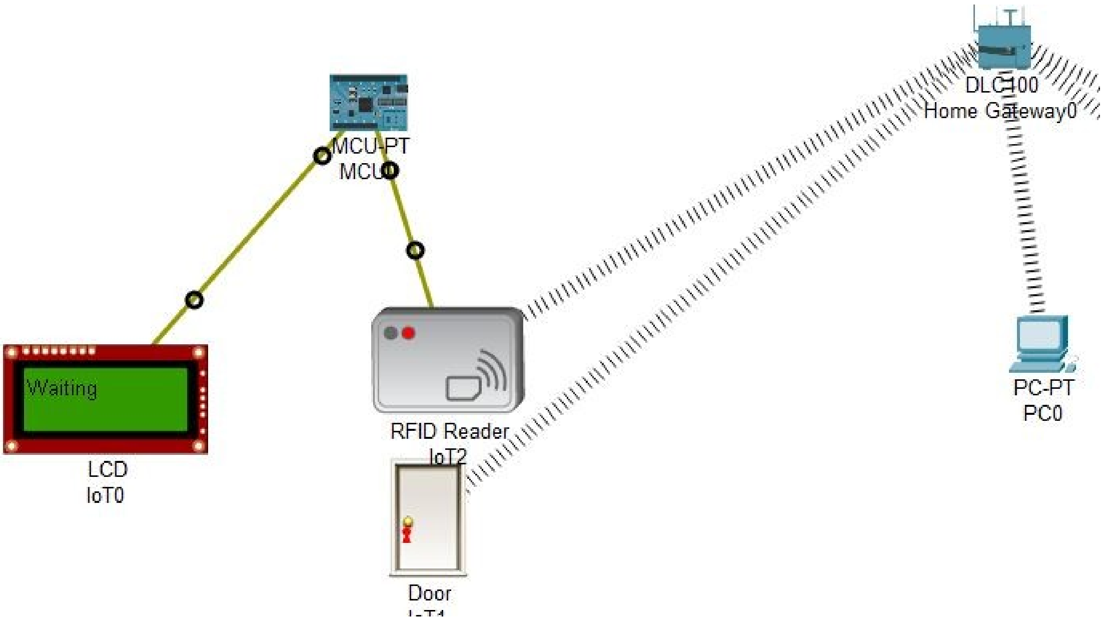
IoT Server - Device Conditions

Home | Conditions | Editor | Log Out

| Actions | Enabled | Name | Condition | Actions |
|---|---|---|---|---|
| Edit Remove | Yes | frontdooropen | PTT0810L32A- Status is 0 | Set IoT1 Lock to Unlock |
| Edit Remove | Yes | close | PTT0810L32A- Status is 1 | Set IoT1 Lock to Lock |
| Edit Remove | Yes | backdoorc | PTT0810RX1S- Status is 0 | Set IoT8 Lock to Unlock |
| Edit Remove | Yes | backclose | PTT0810RX1S- Status is 1 | Set IoT8 Lock to Lock |

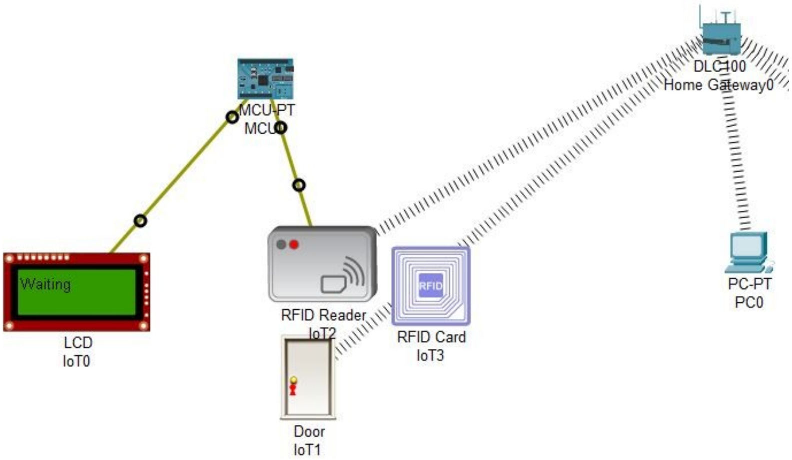
Add

Connections are done as follows:

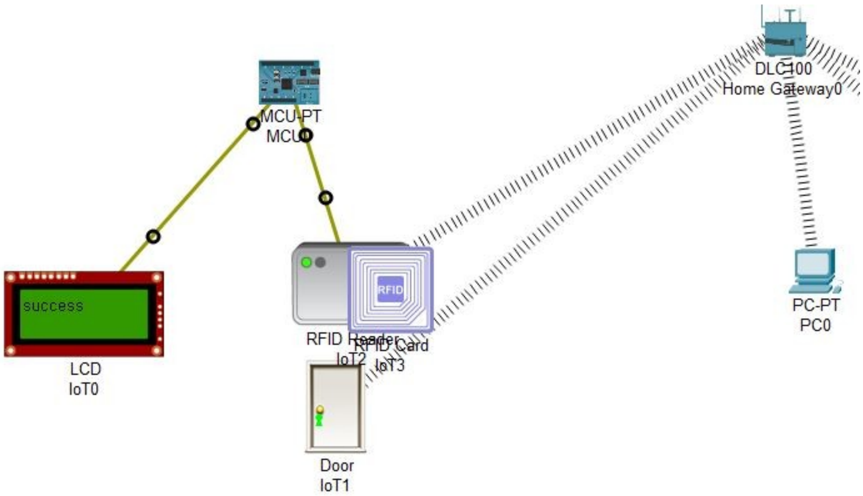


## OUTPUT FOR FRONT DOOR:

## Closed Door:



**Open Door:**



# Practical No:04

**Aim :** -Develop an IoT application that will raise an alarm whenever with going to rain outside based on the weather prediction data.

**Theory: -**

Humidity is the concentration of water vapour present in the air. Water vapor, the gaseous state of water, is generally invisible to the human eye. Humidity indicates the likelihood for precipitation, dew, or fog to be present. Humidity depends on the temperature and pressure of the system of interest.

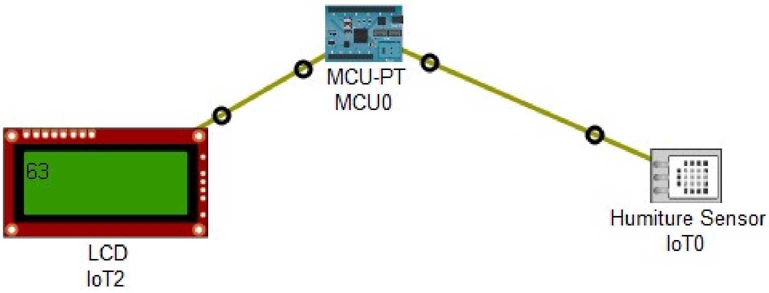
Humidity value : - when the relative humidity is greater than about 70%. Rain. Often, rain will be falling from clouds where the humidity is 100% into air with a lower humidity. For example, it can rain at a ground humidity of 60%, but over time, the humidity will increase.

**Components required:-**
LCD, MCU board, humiture sensor

**Steps:-**
Take the components and arrange as shown in diagram

**Code:**

```
function getSensorData(){
return Math.floor(map(analogRead(A0), 0, 1023, 0, 100) + 0.5);
}

function setup() {
attachInterrupt(A0, function() {
processData(getSensorData());
});
}


function processData(data) {
customWrite(0, data);
}
```



# Practical No:05

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

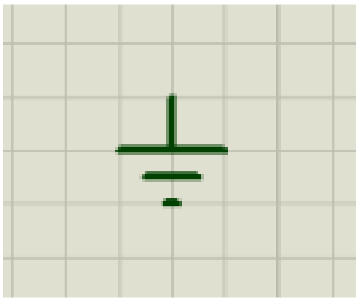
## Theory:

Vibration can be defined as the mechanical oscillation about an equilibrium position of a machine or component or simply the back and forth motion of a machine or component.

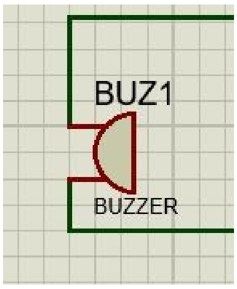
## Apparatus:

## Ground:

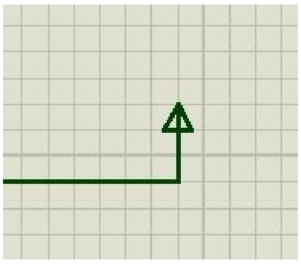
Grounding something simply means connecting it to ground. And in electronics, ground is just a name we give to a certain point in the circuit. For example, in a circuit with one battery (with a positive and a negative terminal), we usually refer to the negative terminal as ground.



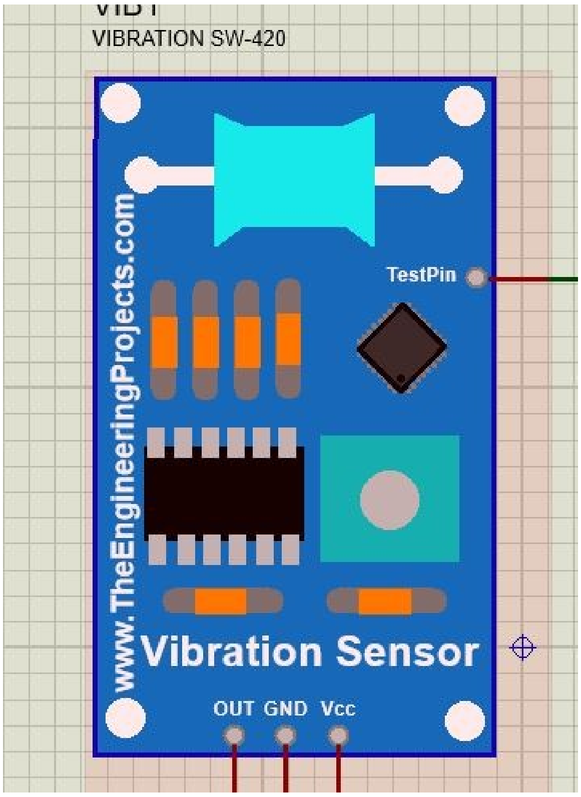
## Buzz:



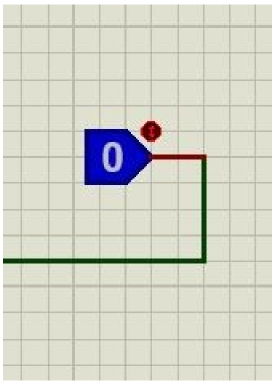
## VCC:



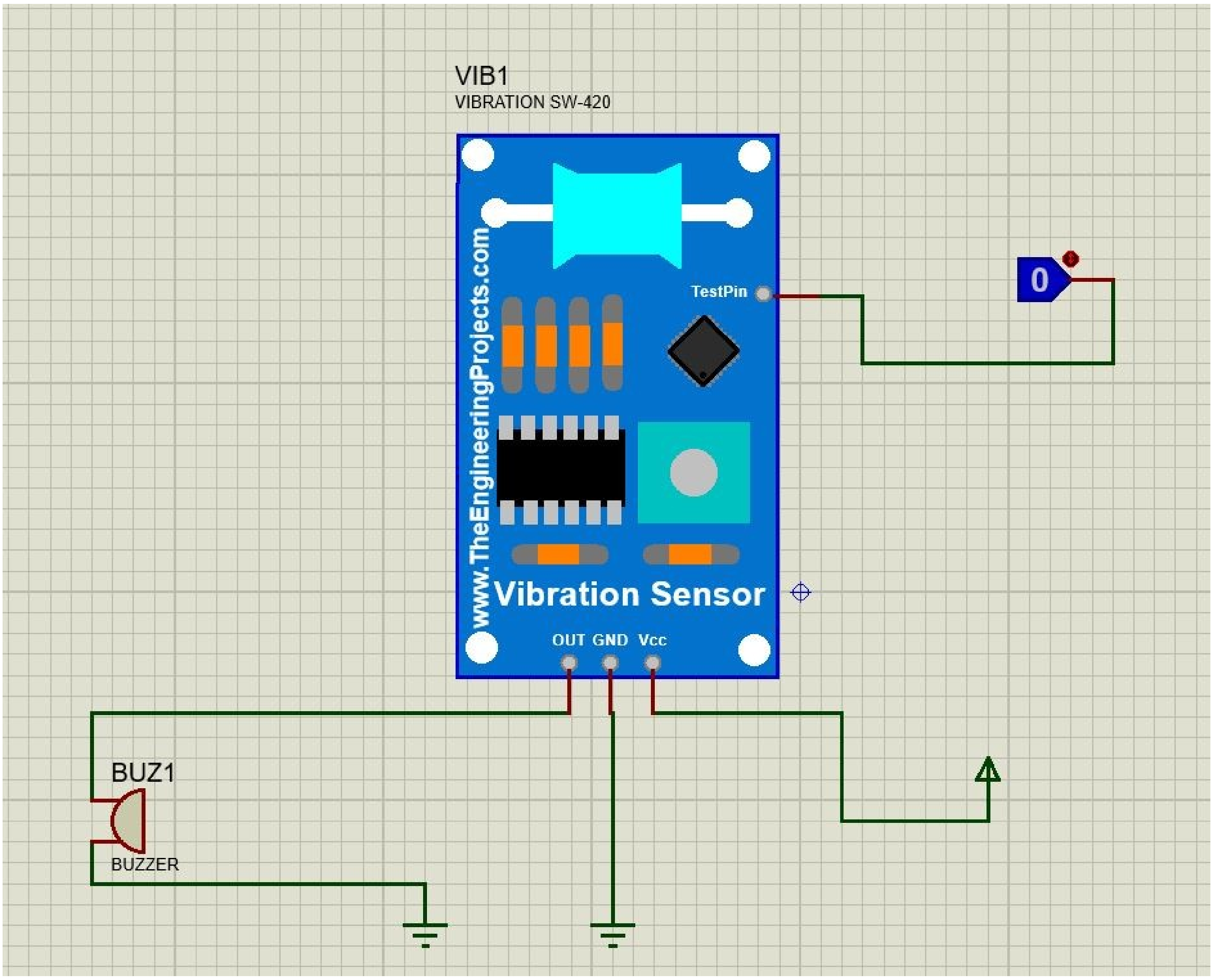
## Vibration Sensor:



## Component:



**OUTPUT:**

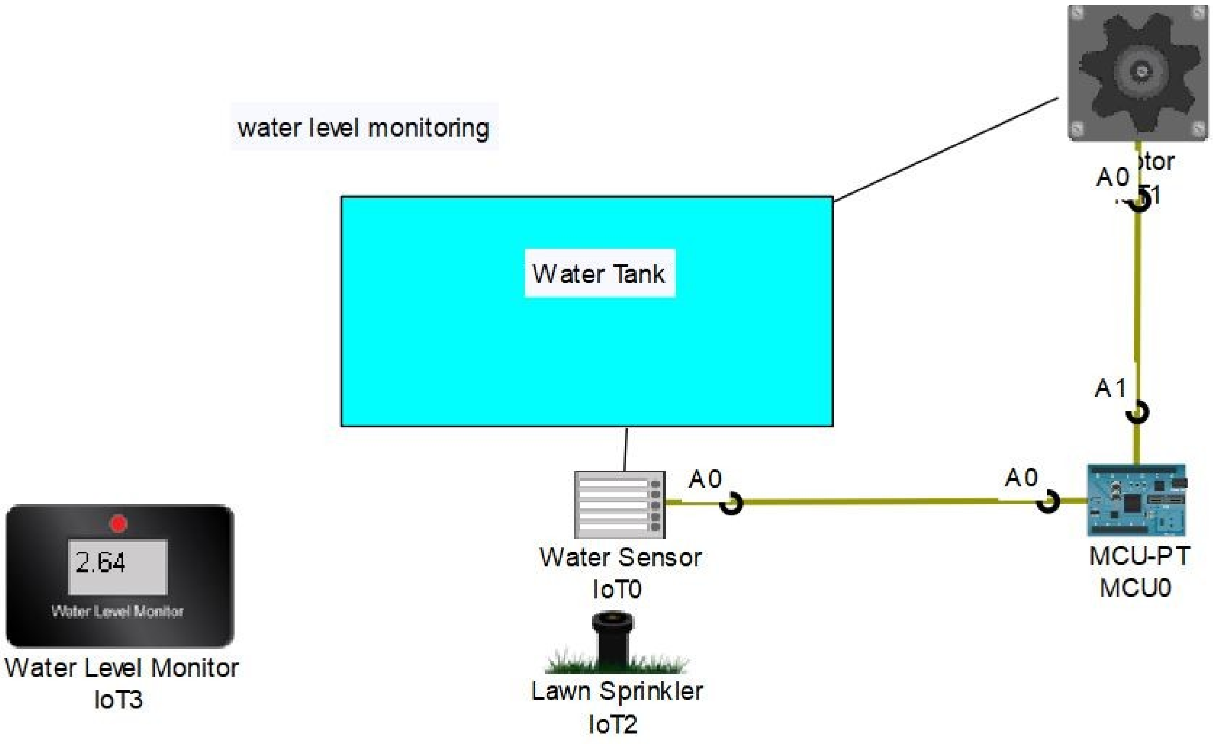


which will alert you by beeping or vibrating your phone whenever you get someone call your name.

# Practical No:06

**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.

**Theory:**General we see a lot of water overflowing from the water tanks, this results in a large scale of water wastage in our household.so to avoid this condition We can use Internet of Things to solve this problem.according to the water level in the tank, if there is a minimum water level then message will be received by the user saying that the tank is empty please switch on the motor, If the water level hey is at maximum level the  IOT module request the to send the message to the user saying that the tank is full please switch off the motor.



STEPS -

1)CREATE THE TOPOLOGY.

2PROGRAM THE MCU.

3) CHECK THE OUTPUT.

CREATE THE TOPOLOGY -

1.

Place a Water Sensor, an MCU board, and a Motor as shown in the figure.

2.
3.

Place a Water Level Monitor and a Lawn sprinkler in order to make the understanding and functionality easier.

4.
5.

Create a rectangle that represents the Water Tank.

6.

PROGRAM THE MCU -

1.

Click on the MCU and go to Programming > New and create a new Python Script called ‘ Water Monitor ‘.

2.
3.

Write the following code in the main.py file -

4.

```
"""WATER LEVEL MONITORING SYSTEM"""
from gpio import *
#provides the analogRead and digitalWrite method from time import *
#provides the sleep method



while True:
        water_level=analogRead(A0)
#reads the analog input from the water sensor at pin A0
        if water_level<400:
                digitalWrite(A1,HIGH)
#starts the motor at pin A1 else:
digitalWrite(A1,LOW)
#stops the motor at pin A1
sleep(0.3)
```

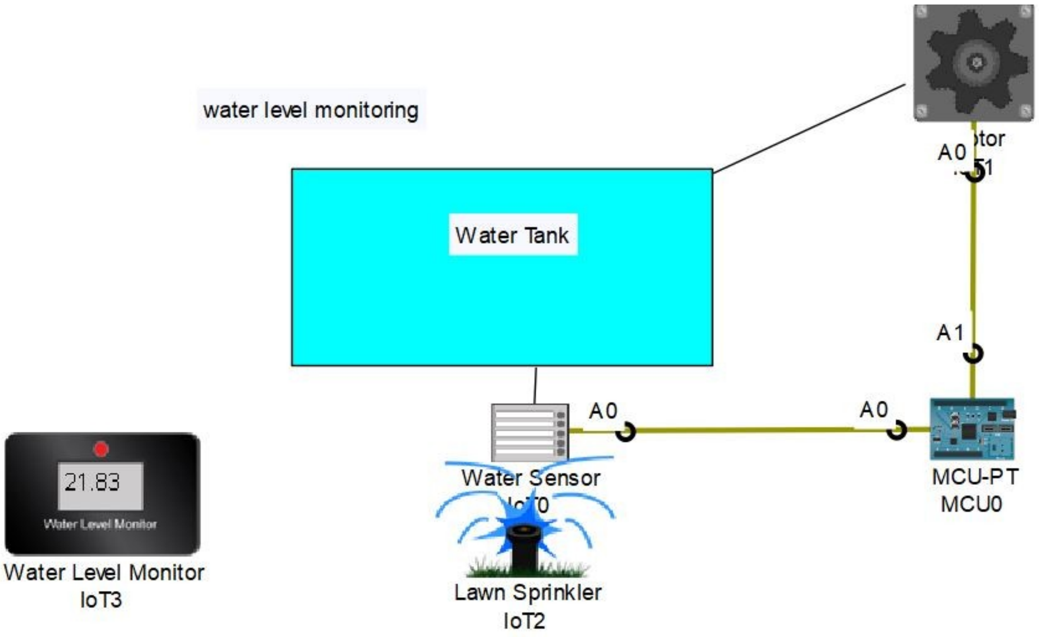
```
"""WATER LEVEL MONITORING SYSTEM"""
from gpio import *
#provides the analogRead and digitalWrite method
from time import *
#provides the sleep method

while True:
   water_level=analogRead(A0)#reads the analog input from the water sensor at pin A0
   if water_level<400:
      digitalWrite(A1,HIGH)#starts the motor at pin A1
   else:
      digitalWrite(A1,LOW)#stops the motor at pin A1
   sleep(0.3)
```

CHECK THE OUTPUT -

1.

Run the main.py file and observe the motor. It moves when the water level is low to allow the tank to fill up.

2.

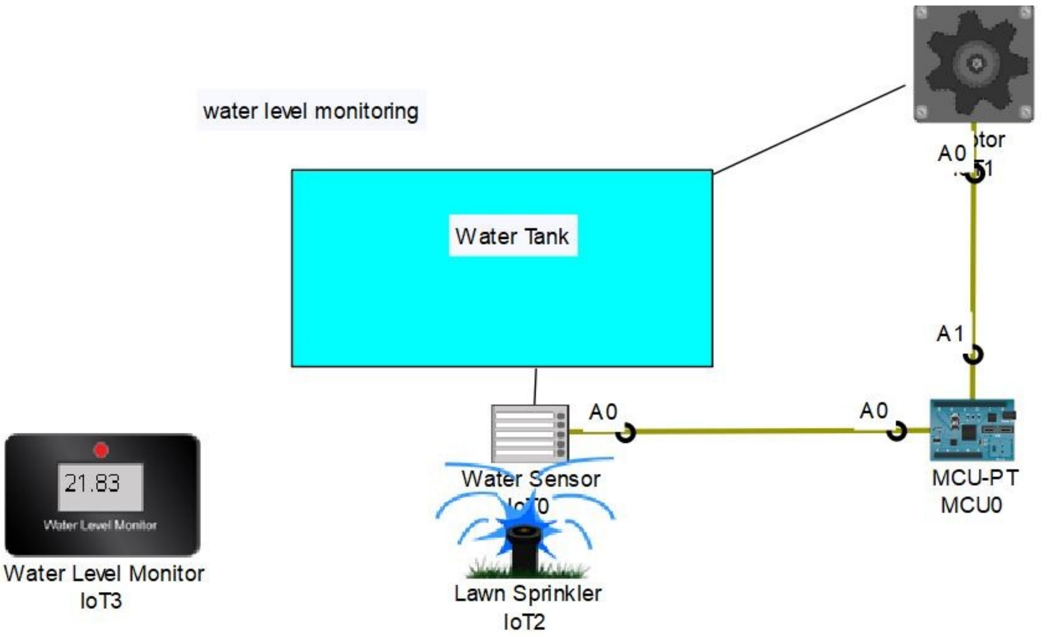


1.

Press the Alt key on your keyboard and click on the Lawn Sprinkler to start it. This will gradually increase the water level and as soon as the water level crosses the given threshold the Motor stops rotating.

2.

Note - We use the Lawn Sprinkler to regulate the water level correctly. This can be done in other applications without a sprinkler. We use the Water Level Monitor to ensure that the level is not 0.



# Practical No:07

**Aim:** Develop an IoT module to which measure the intensity of light and send the same to your PC/ Phone.

**Theory:-** A photoresistor (also known as a Photocell, or light-dependent resistor, LDR, or photo-conductive cell) is a passive component that decreases resistance with respect to receiving luminosity (light) on the component's sensitive surface. The resistance of a photoresistor decreases with increase in incident l intensity; in other words, it exhibits,photoconductivity.

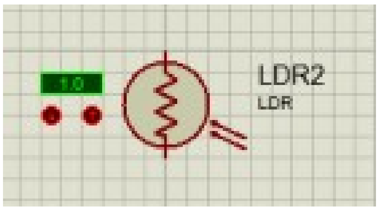
**Apparatus:**

**Arduino UNO R3:**

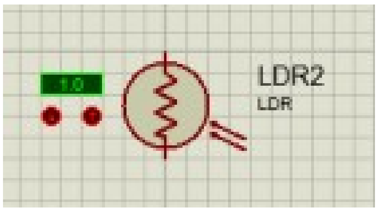
Arduino UNO is a low-cost, flexible, and easy-to-use programmable open-source microcontroller board that can be integrated into a variety of electronic projects. Arduino UNO features AVR microcontroller Atmega328, 6 analogue input pins, and 14 digital I/O pins out of which 6 are used as PWM output.

**Photoresistor/LDR:**

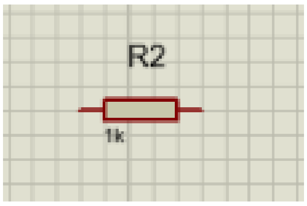
Photoresistors are also sometimes referred as LDR (Light Dependent Resistor), semiconductor photoresistor, photoconductor, or photocell. Photoresistor changes its resistance only when it is exposed to light. In other words, the flow of electric current through the photoresistor increases when the intensity of light increases.



**Resistor(MINRES1K):**

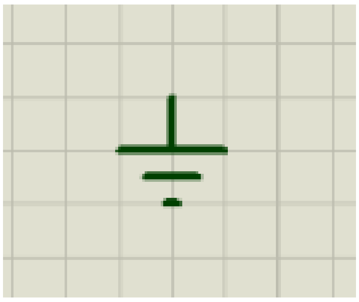


The main purpose of resistor is to reduce the current flow and to lower the voltage in any particular portion of the circuit.



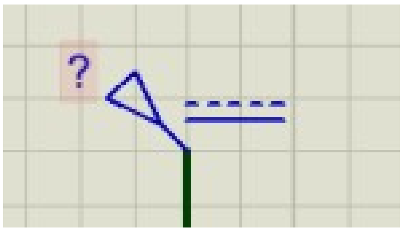
**Ground:**

Grounding something simply means connecting it to ground. And in electronics, ground is just a name we give to a certain point in the circuit. For example, in a circuit with one battery (with a positive and a negative terminal), we usually refer to the negative terminal as ground.



**DC Power:**

**Direct current** (**DC**) is one-directional flow of electric charge. An electrochemical cell is a prime example of DC power. Direct current may flow through a conductor such as a wire but can also flow through semiconductors and insulators.



**Steps to perform the practical in Proteus Software:**

**Prerequisites:**

1. Proteus software should be downloaded and installed.
2. Arduino IDE should be downloaded and installed
3. Permissions should be provided to the user for accessing the libraries. 4. Two libraries to add in Library folder for Arduino UNO R3

Library link:
https://drive.google.com/drive/folders/1JwX3KjqlxjKIjtHkyVLBfoNxQY1i0pYJ?usp=sharing Path:
C:\Program Files (x86)\Labcenter Electronics\Proteus 8 Professional\LIBRARY

**Follow the steps to setup the circuit:**

1. Open the proteus software and create a new project with default settings. 2. Once the project is created click on component mode in the left pane. 3. Click on p(pick device) and search for the components

4. Search for the UNO R3 and double click on that it will be added to components list. 5. Now search for LDR and double click on that to add

6. Now search for Resistor MINRES1K and add that to component list. 7. Click on ARDUINO UNO R3 and place it into the simulating area

8. Similarly add LDR and MINRES1K resistor

9. Now click on Generators in the left pane and select DC add it to simulating area 10. Similarly add Ground from the left pane in terminals

11. Connect the DC to first terminal of LDR

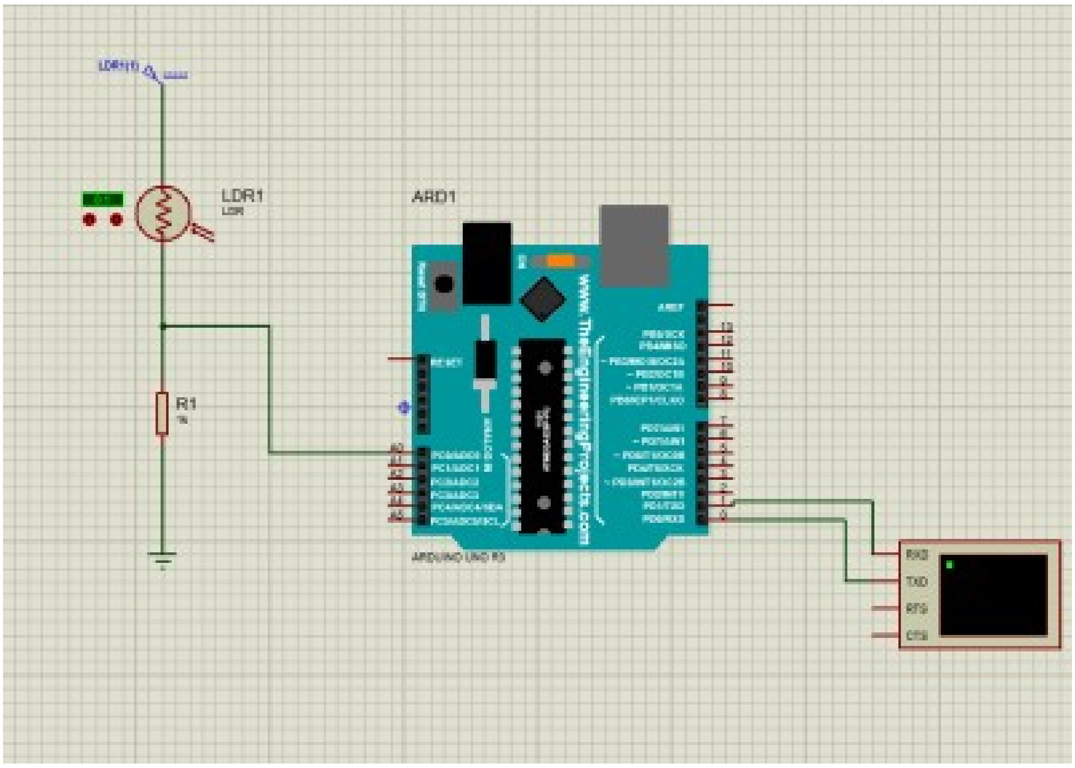
12. The second terminal of LDR will go to Analog A0 and first terminal of resistor. 13. Lastly second terminal of resistor will connect to the ground.

14. To pass the current through the circuit we have to provide some power so we will add the voltage from DC.

15. Double click on DC and enter voltage of your preference (example: 5) 16. To display the intensity, we need a monitor and for that we will add Virtual terminal 17. From the left pane click on instrument and select virtual terminal and add to the simulating area.

18. Now connect the RXD of terminal to PD1/TXD 1 in the Arduino and TXD t0 PD0/RXD 0 Your circuit should look like this:

19. Now open the Arduino IDE and write the following code

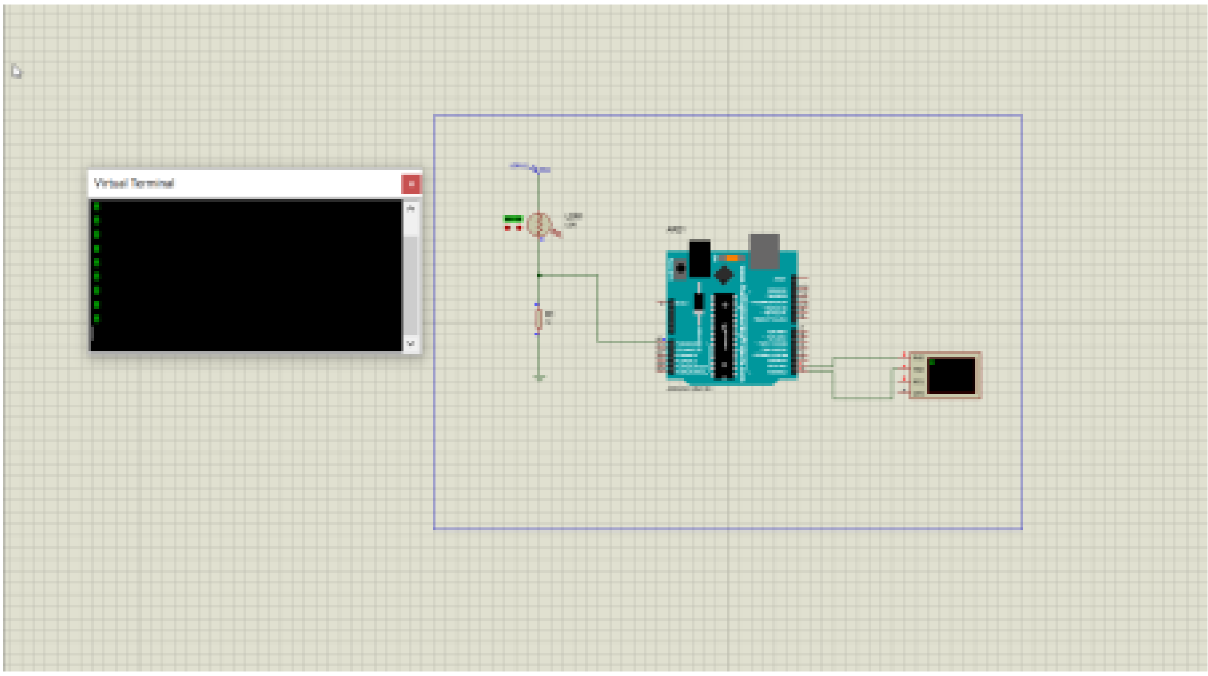


```
void setup()
{
   Serial.begin(9600);
}
```

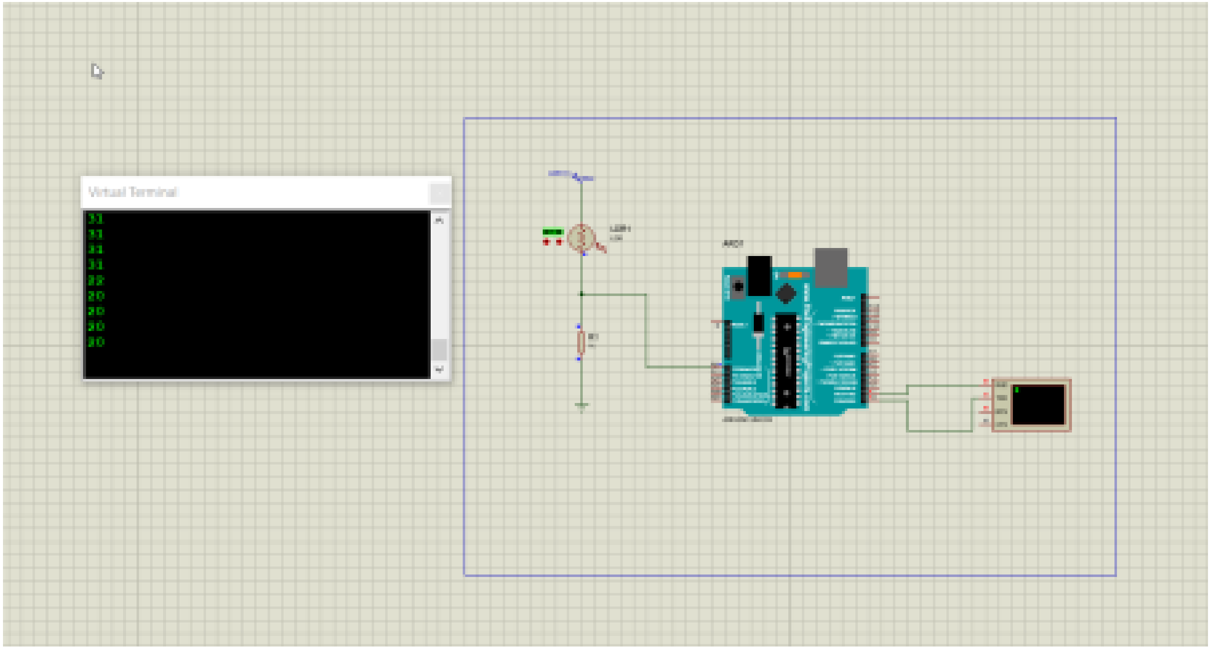
```
   void loop()
   {
    intensity = analogRead(A0);
    Serial.println(intensity);
}
```

20. To generate the binary file first we will compile the code.
21. Click on Sketch at the top menu bar and click on Verify Compile
22. If it asks for folder path provide a suitable path which you will remember 23. Then
again click on Sketch and click on Export Compiled Binary
24. The .hex extension file will be generated at specified path.
25. Now in Proteus software double click on Arduino, edit component window will be displayed.
26. Click on browse for Program files and select the file with .hex extension which we just generated and
click on open.
27. After clicking on Ok our entire setup and coding part is completed 28. Now run
the simulator by clicking on the play button.
29. To control the intensity of light we can use the + - buttons besides the LDR 30. In the
Virtual Monitor output as Intensity will be displayed.
30. In the Virtual Monitor output as Intensity will be displayed.

Initial Intensity :



After increasing/decreasing the intensity:



```
31
31
31
31
22
20
20
20
20
```

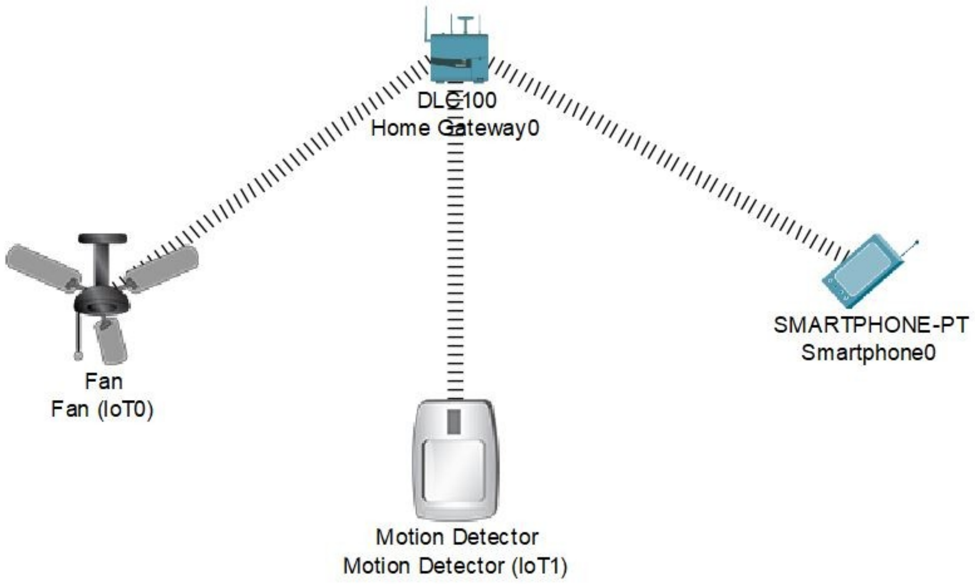
# Practical No:08

**Aim: Develop an IoT application for Motion detection**

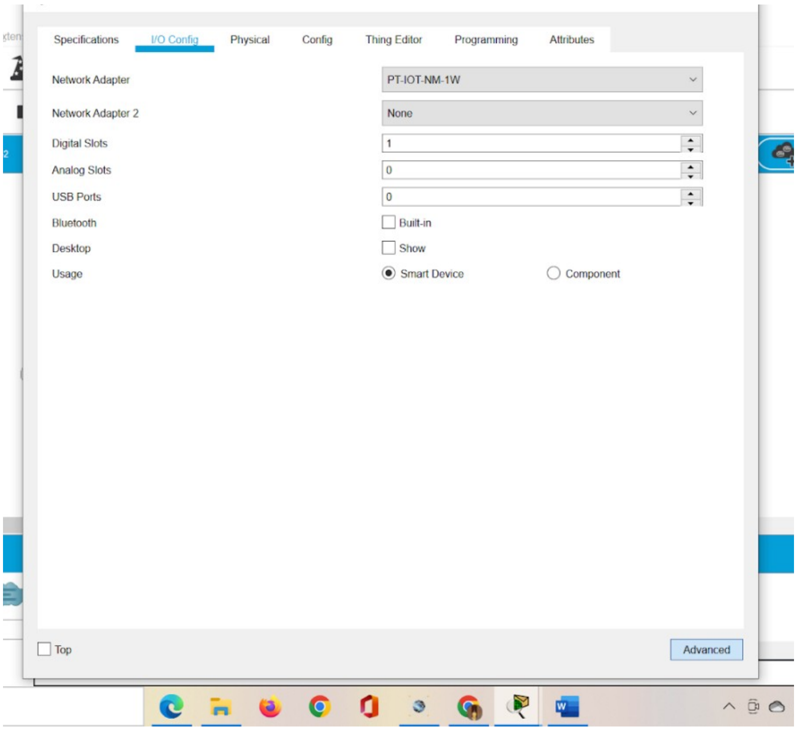
Components:- Home Gateway, Motion Detector, Fan, Smartphone.

Step 1: Add all the components in the cisco packet tracer, give proper names to each component and arrange as shown in below figure.

**Theory:**The concept as shown below is to allow monitoring detect motion and by detecting motion remotely  we can handel home appliance (here we apply on fan as example) with the additional capability to detect motion if someone is at home, and also allow switch system via smartphone.



Step 2: to connect the fan with home gateway, click on the fan go to config tab, click on advanced option and select PT-IOT-NM-1W in network adapter in I/O config tab as shown below.



Network Adapter PT-IOT-NM-1W

Network Adapter 2 None

Digital Slots 1

Analog Slots 0

USB Ports 0

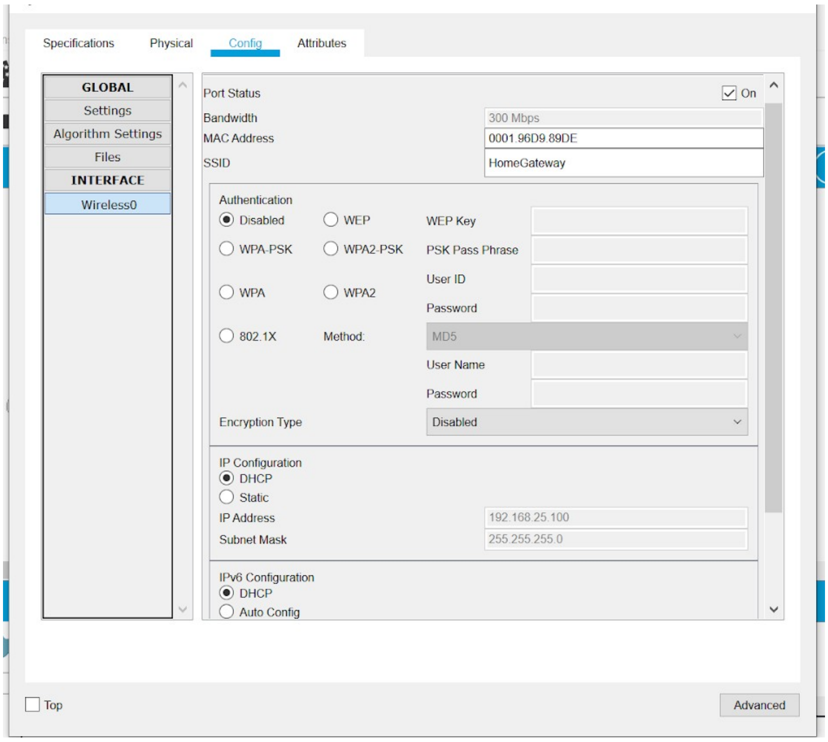
Bluetooth □ Built-in

Desktop □ Show

Usage ● Smart Device ○ Component

□ Top

Step 3: Select SSID of fan to be HomeGateway. The fan will be connected to home gateway.



Port Status ☑ On

Bandwidth 300 Mbps

MAC Address 0001.96D9.89DE

SSID HomeGateway

Authentication

● Disabled ○ WEP WEP Key

○ WPA-PSK ○ WPA2-PSK PSK Pass Phrase

User ID

○ WPA ○ WPA2 Password

○ 802.1X Method: MD5

User Name

Password

Encryption Type Disabled

IP Configuration

● DHCP

○ Static

IP Address 192.168.25.100

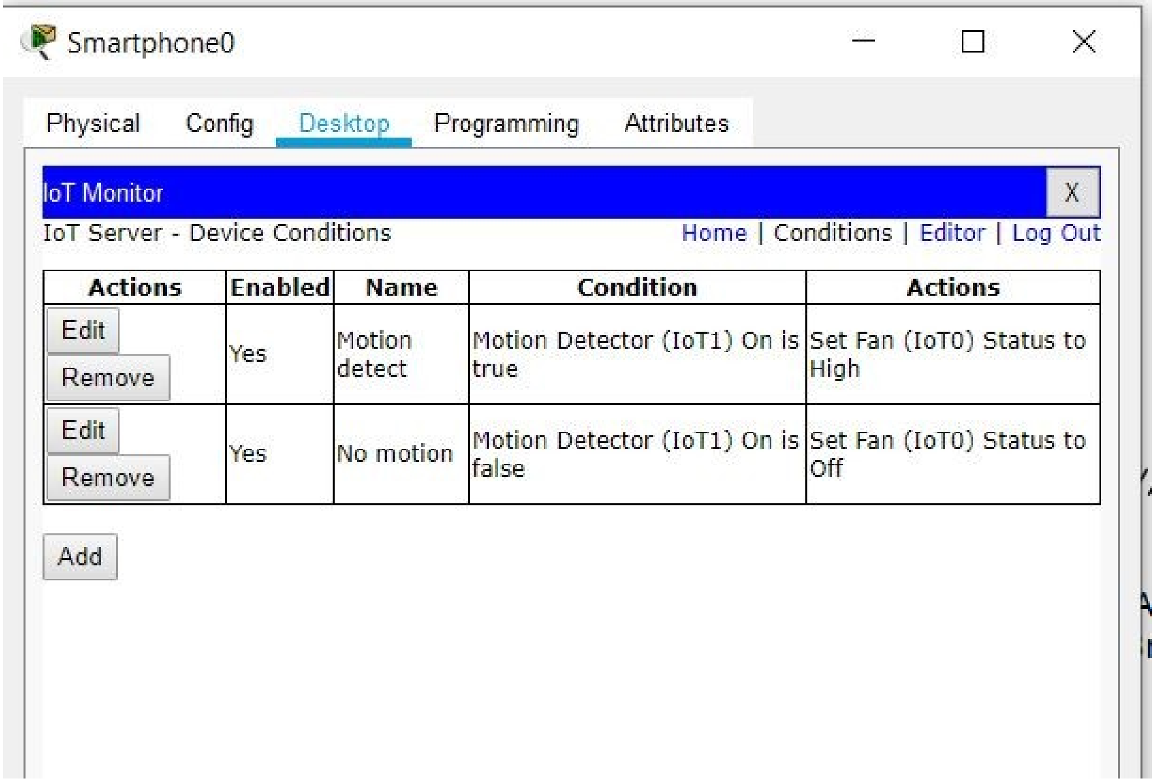
Subnet Mask 255.255.255.0

IPv6 Configuration

● DHCP

○ Auto Config

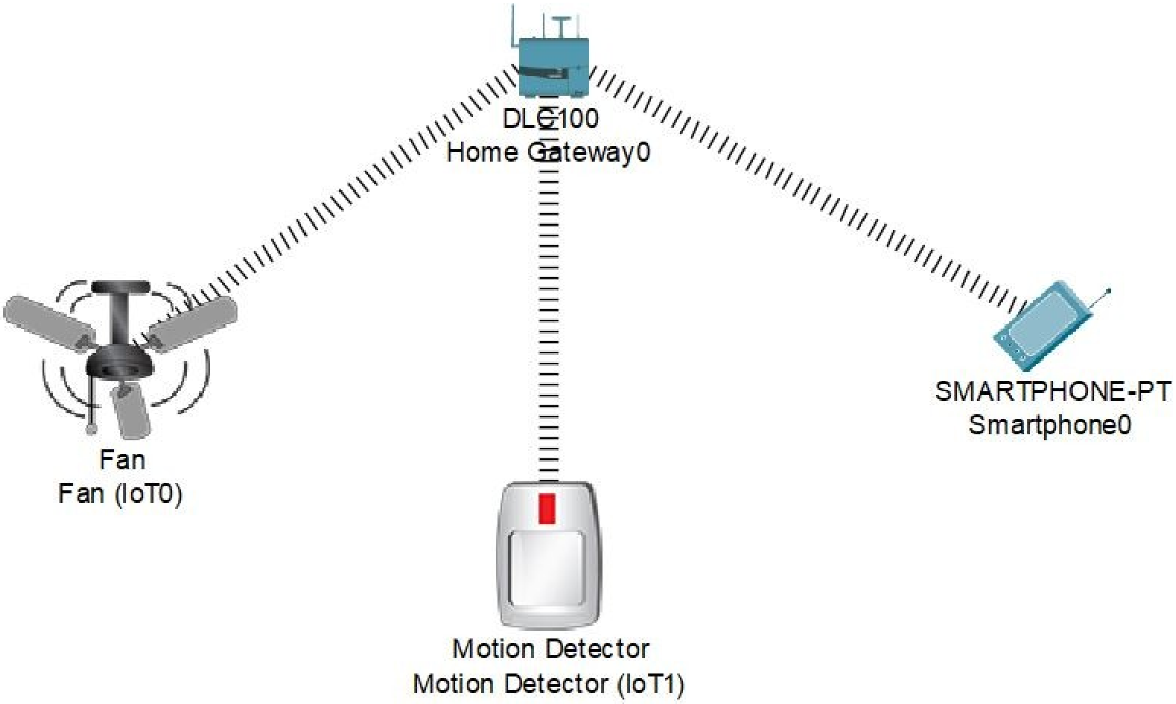
Step 4: do the same settings for motion detector as well. It will also get connected to home gateway.
Step 5: In smart phone, wireless tab put SSID as HomeGateway. It will get connected to home gateway.
Step 6: In smart phone, go to IoT monitor and set following conditions.



IoT Server - Device Conditions

| Actions | Enabled | Name | Condition | Actions |
|---|---|---|---|---|
| Edit<br>Remove | Yes | Motion detect | Motion Detector (IoT1) On is true | Set Fan (IoT0) Status to High |
| Edit<br>Remove | Yes | No motion | Motion Detector (IoT1) On is false | Set Fan (IoT0) Status to Off |

Step 7 : To execute the practical, click Alt+mouse over on motion detector, fan will switch on with high speed.
When we remove alt+ mouse over from motion detector, it will switch off the fan.


DLC100
Home Gateway0
Fan
Fan (IoT0)
Motion Detector
Motion Detector (IoT1)
SMARTPHONE-PT
Smartphone0