UNIVERSITY OF MUMBAI

DEPARTMENT OF COMPUTER SCIENCE

मुंबई विद्यापीठ
University of Mumbai
Re-accredited with A++ Grade
(CGPA 3.65) by NAAC (3rd Cycle 2021)

M.Sc. Computer Science – Semester II (NEP 2020)

Embedded Internet Of Things

JOURNAL

2023-2024

Seat No. _____

मुंबई विद्यापीठ
University of Mumbai
Re-accredited with A++ Grade
(CGPA 3.65) by NAAC (3rd Cycle 2021)

UNIVERSITY OF MUMBAI

DEPARTMENT OF COMPUTER SCIENCE

CERTIFICATE

This is to certify that the work entered in this journal was done in the University Department of Computer Science laboratory by Mr./Ms._____ Seat No. _____ for the course of M.Sc. (Computer Science) - Semester II (NEP 2020) during the academic year 2023- 2024 in a satisfactory manner.

_____ _____

Subject In-charge **Head of Department**

_____

External Examiner

INDEX

Sr. no.	Name of the practical	Page No.	Date	Sign
1	**Design and Implement basics embedded circuits.** 1. **Automatic Alarm system-Alarm Should get trigger by senor.** 2. **Timer Based Buzzer** 3. **Sensor based Counting Device.**			
2	**Demonstrate Communication between two embedded devices using UART port.**			
3	**Built an IOT system to send Ticket before entering the bus.**			
4	**Develop a IOT application which will record the movement and Orientation of your Phone and give the data to the PC.**			
5	**Develop an IOT application that will raise an alarm whenever with going to rain outside based on the weather predication data.**			
6	**Develop an IoT module to which measure the intensity of light and send the same to your PC/ Phone.**			
7	**Develop and IOT application for monitoring water level in tanks and automatically start the motor to fill the tank if the level goes below the critical level.**			
8	**Develop an IOT application for motion detection.**			

PRACTICAL-01

Aim: Design and Implement basics embedded circuits.

Theory: An embedded system is a microprocessor-based computer hardware system with software that is designed to perform a dedicated function, either as an independent system or as a part of a large system. At the core is an integrated circuit designed to carry out computation for real-time operations.

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

Components:- Siren, HomeGateway, Motion Detector, Door, Smartphone.

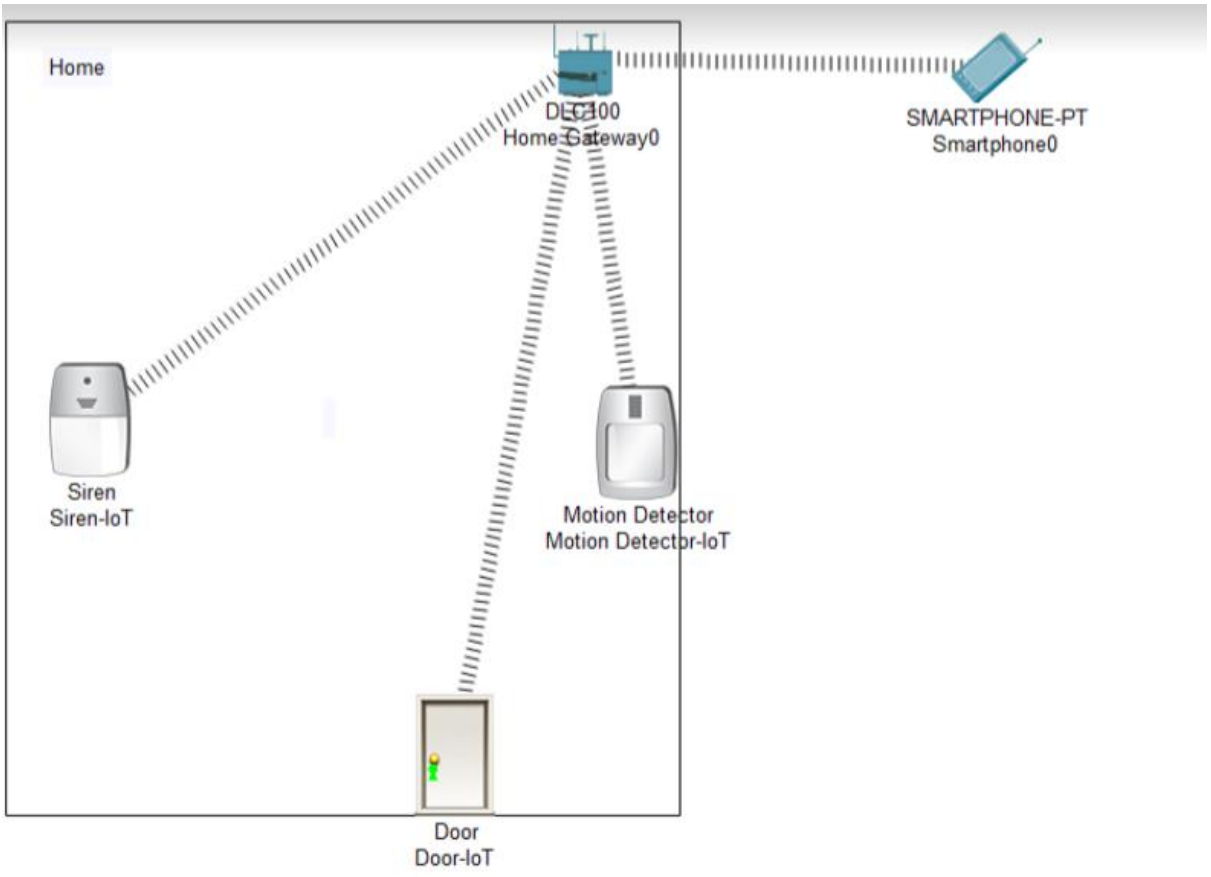
Step 1: Add all the components in the cisco packet tracer and do the following settings.

i. Click on the siren, go to config tab, then go to display name and change the display name to Siren-IoT. Scroll down and select the IOT SERVER as “HomeGateway”.

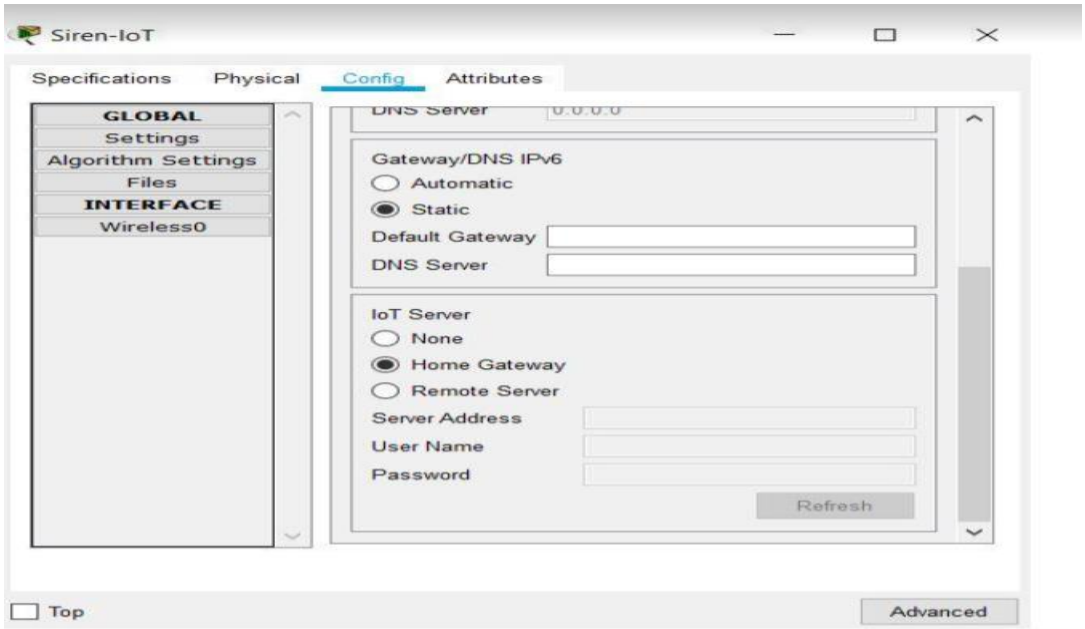
   Go to the Advanced tab, choose the I/O config tab, and select the network adapter as “PT-IOT-NM-1W”.

ii. Repeat step i for motion detector and door.

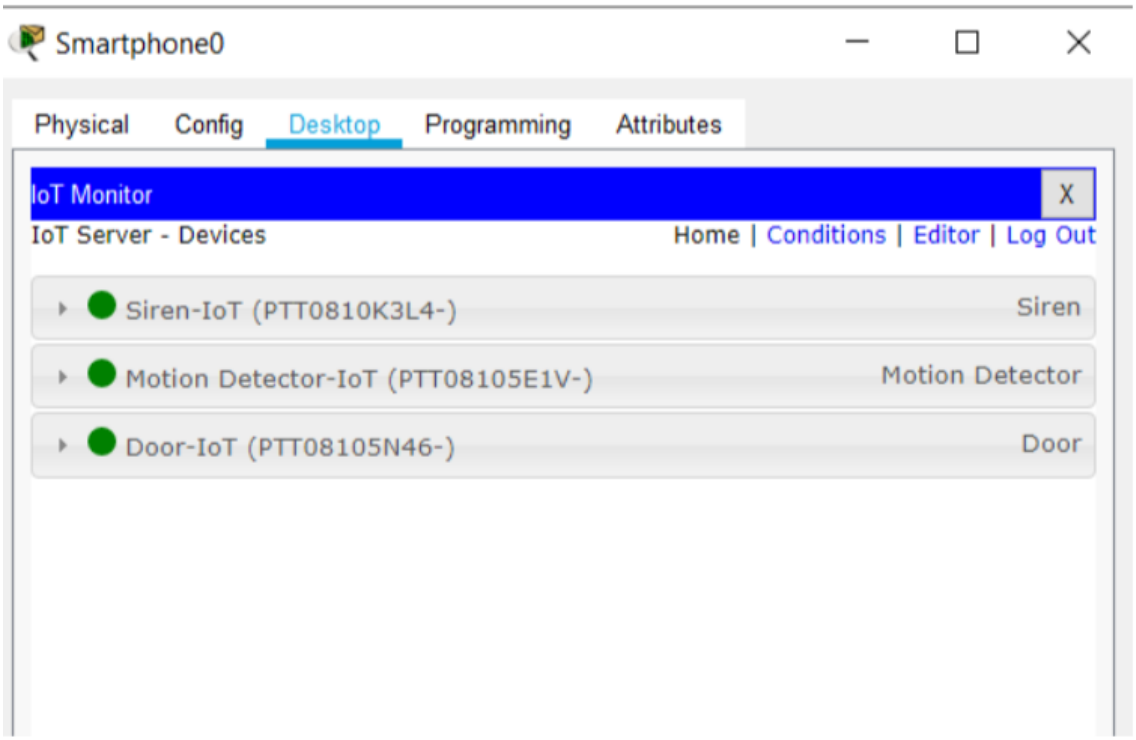
iii. Go to HomeGateway, and under the wireless tab, copy it’s SSID.



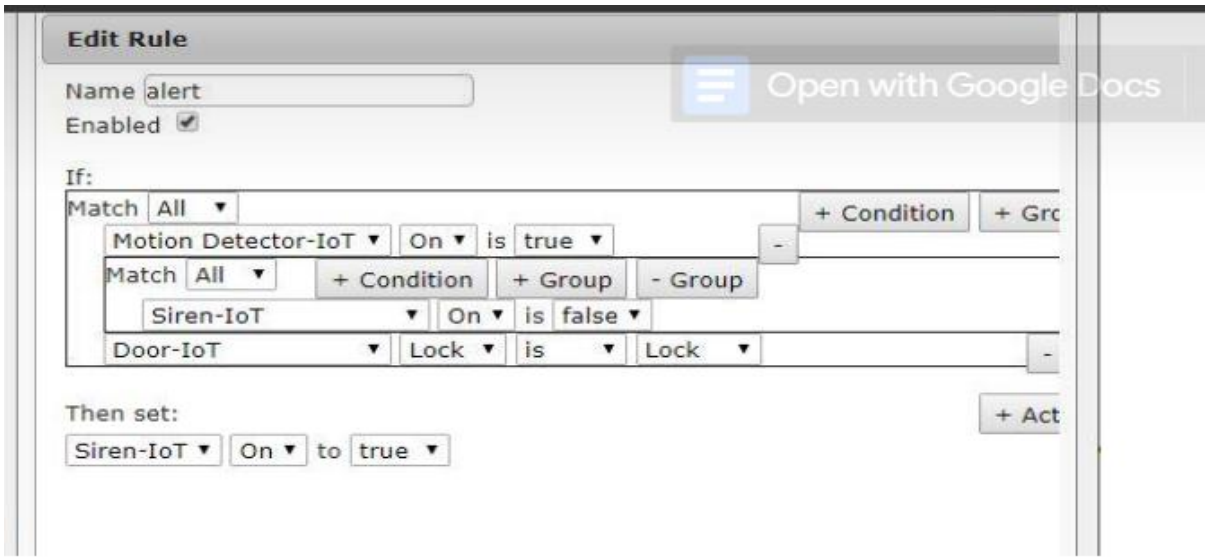
Step 2: Click on the smartphone and set its SSID to HomeGateway (copied) and IoT Server to HomeGateway.

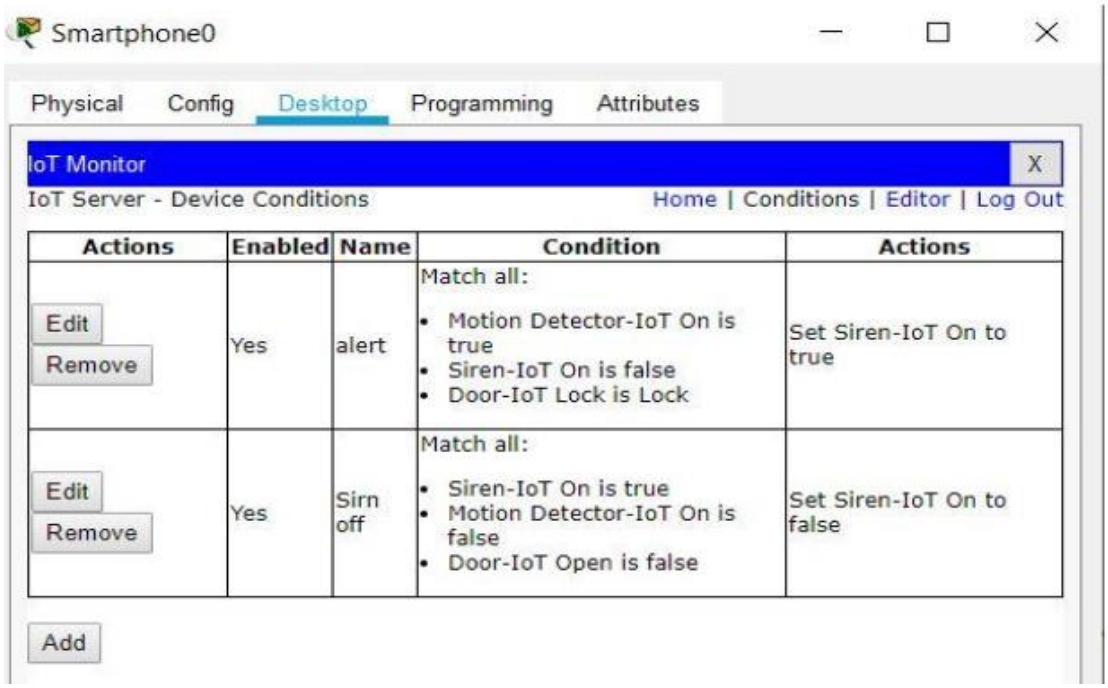


Step 3: In Smartphone, go to Desktop IoT Monitor all the devices connected to the HomeGateway will be displayed.

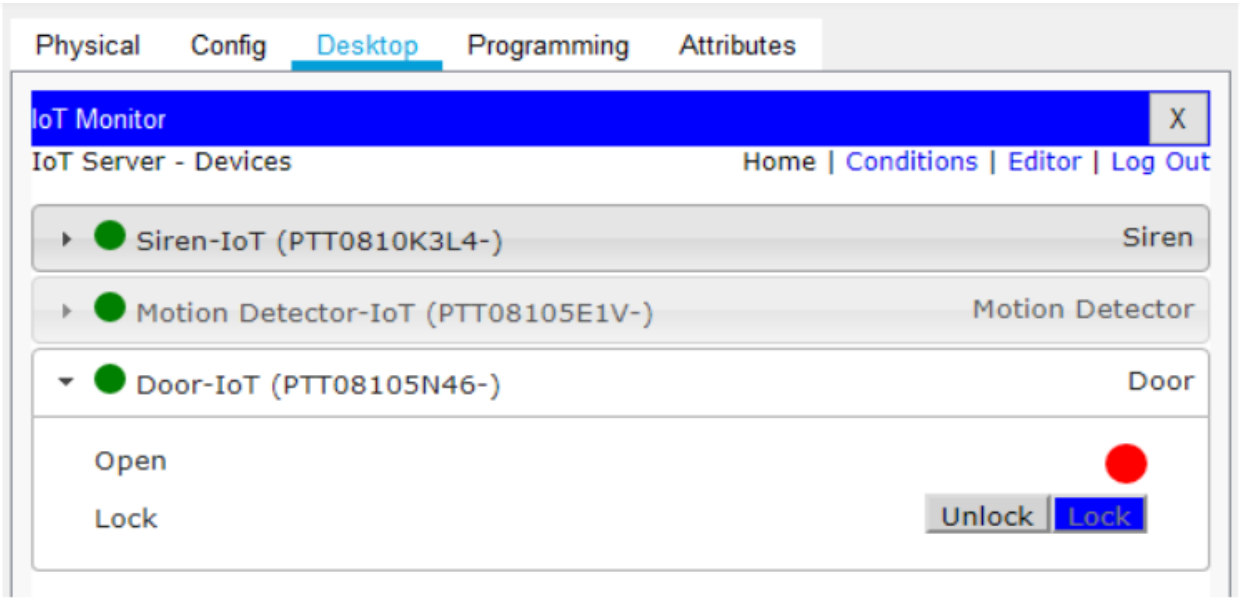


Step 4: Click on Conditions and Add conditions where if Motion Detector on is true and Door Lock is lock then set Siren on to true and also when motion detector is false and the door is lock then the siren is off.

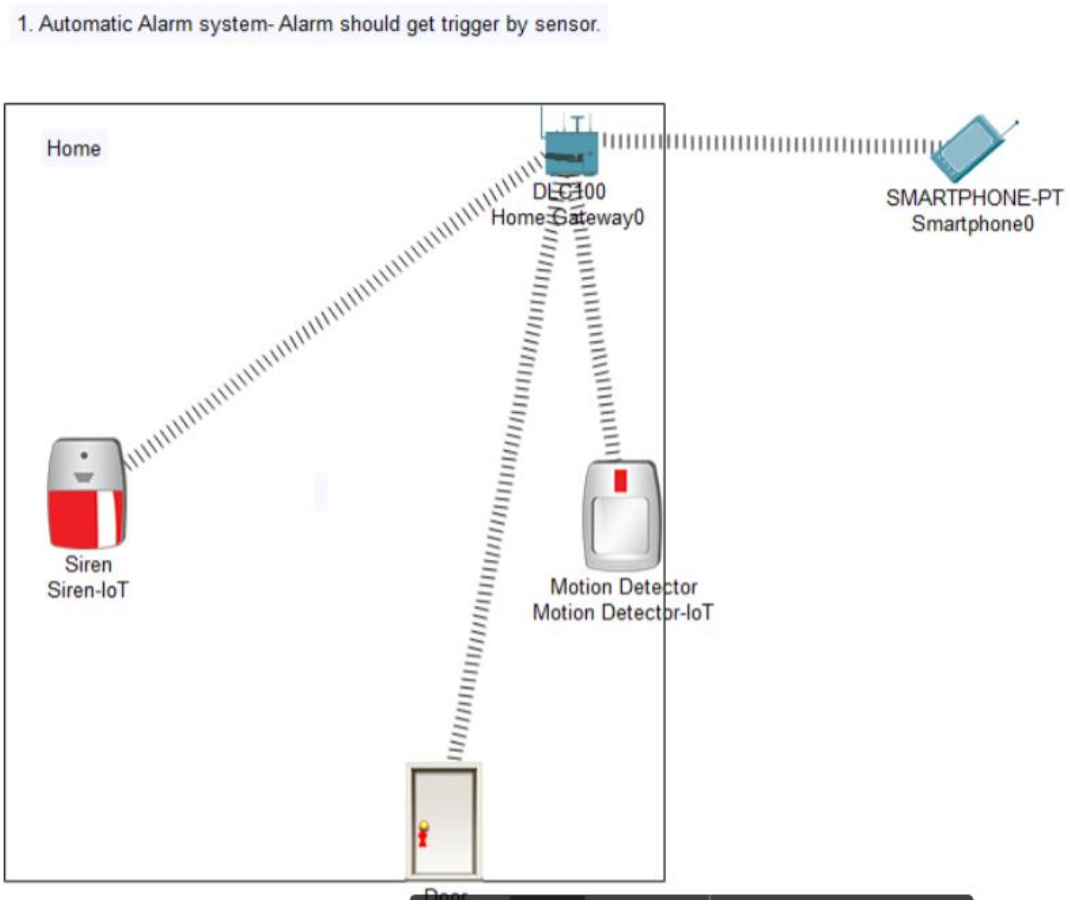




Step 5: Make sure Door is locked. Check that in IoT monitor.



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.

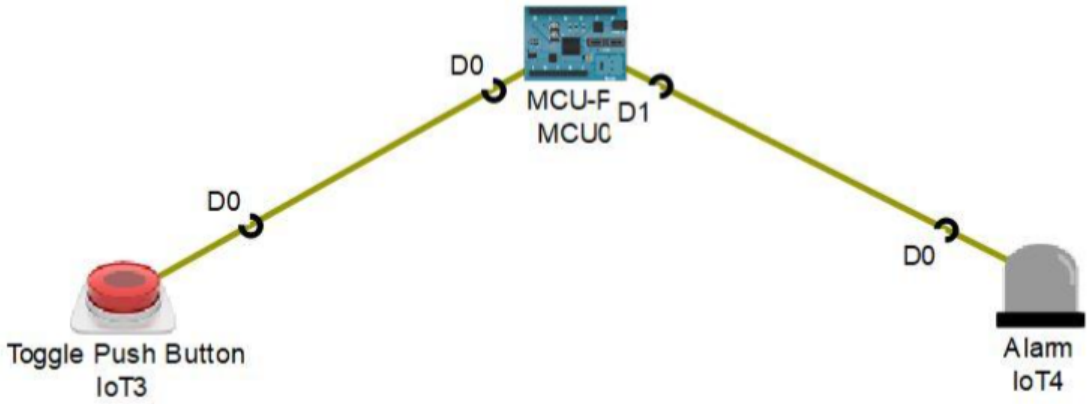


2. Timer based buzzer

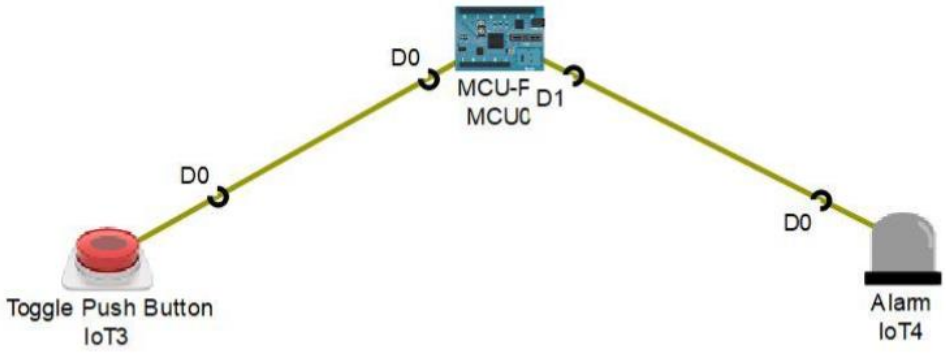
Components:- Toggle Push Button, MCU Board, Alarm, IoT cables.

Steps:-

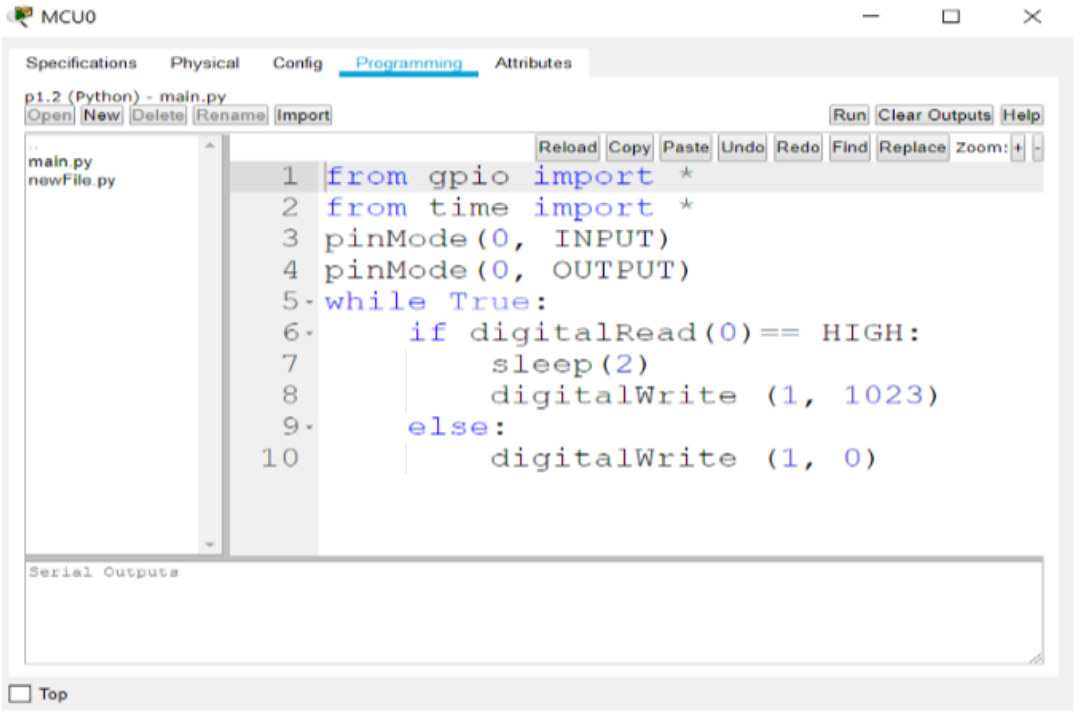
Step 1: Add all the components in the cisco packet tracer.

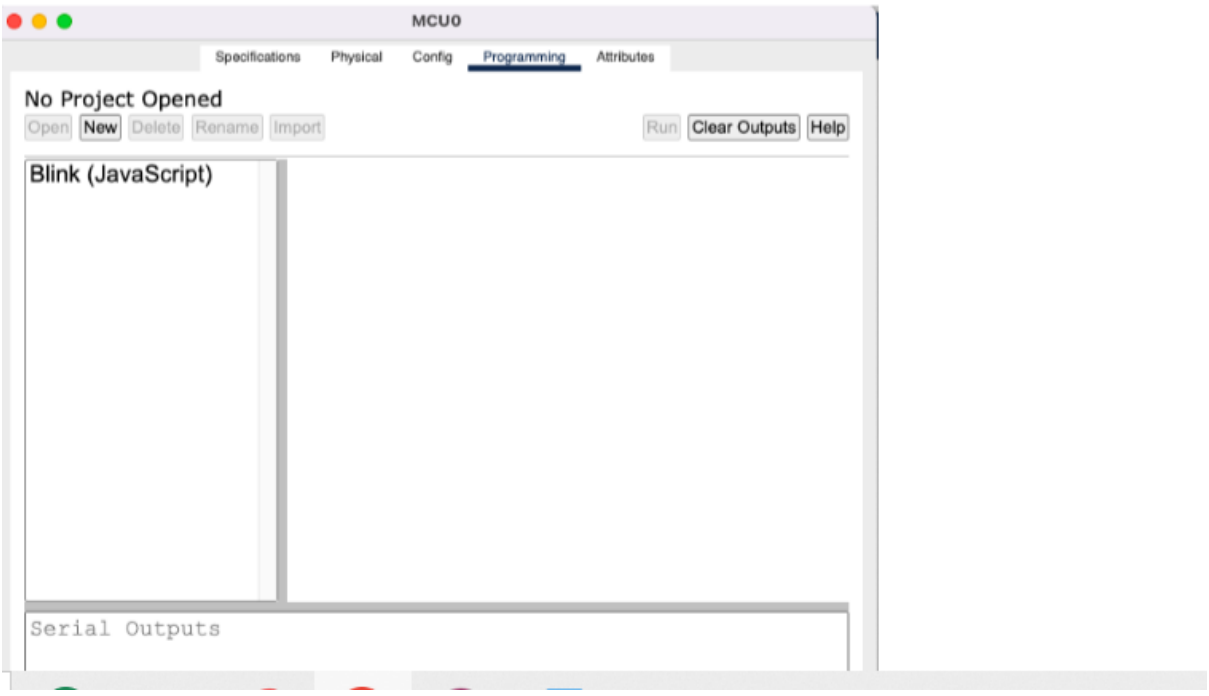


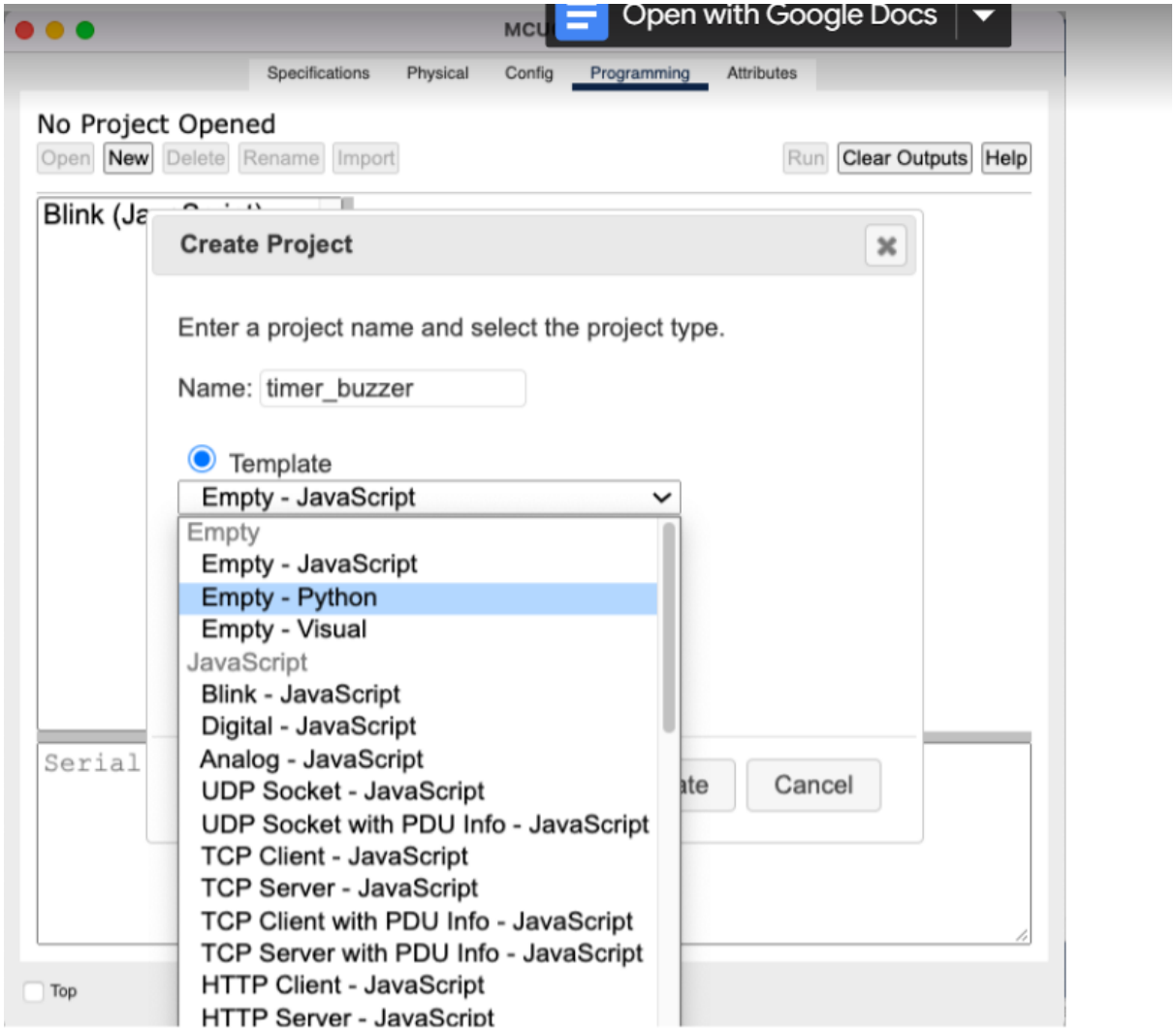
Step 2: Connect MCU with Toggle Push Button and MCU with IoT Cables.



Step 3: Click on MCU and then select Programming.


MCU0
Specifications
Physical
Config
Programming
Attributes
p1.2 (Python) - main.py
Open
New
Delete
Rename
Import
Run
Clear Outputs
Help
Reload
Copy
Paste
Undo
Redo
Find
Replace
Zoom:
main.py
newFile.py
1 from gpio import *
2 from time import *
3 pinMode(0, INPUT)
4 pinMode(0, OUTPUT)
5 while True:
6 if digitalRead(0)== HIGH:
7 sleep(2)
8 digitalWrite (1, 1023)
9 else:
10 digitalWrite (1, 0)
Serial Outputs
Top

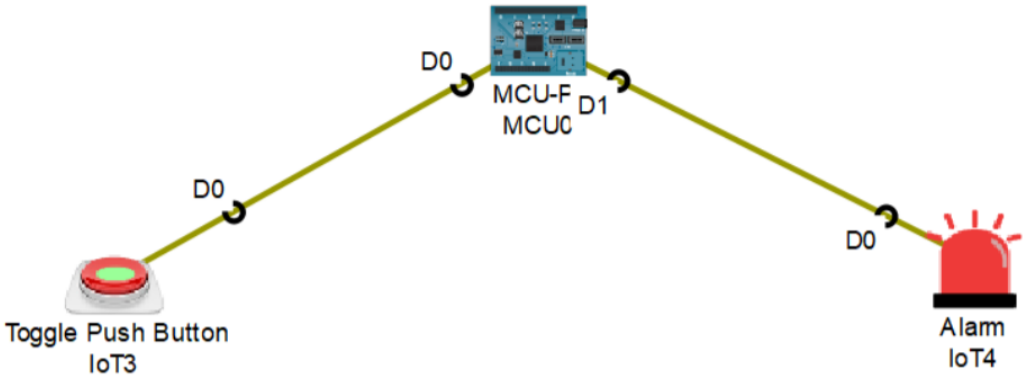

MCU0
Specifications
Physical
Config
Programming
Attributes
No Project Opened
Open
New
Delete
Rename
Import
Run
Clear Outputs
Help
Blink (JavaScript)
Serial Outputs



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.

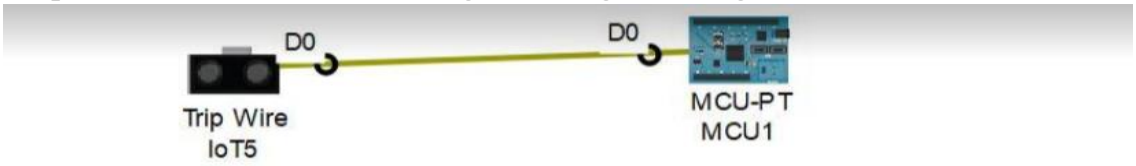


3. Sensor based Counting device

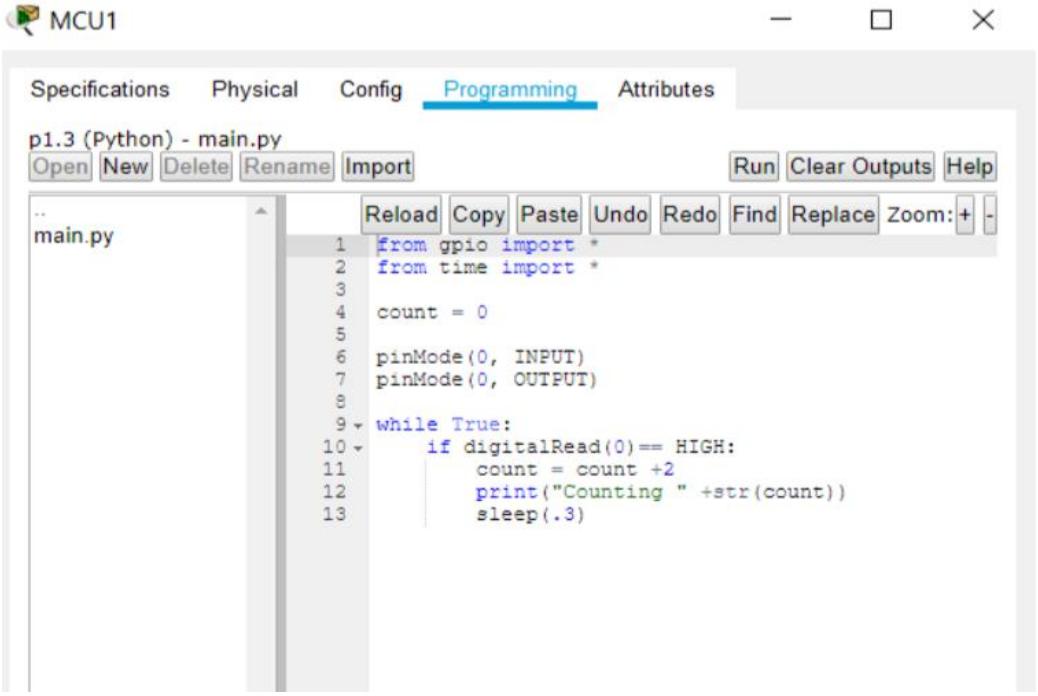
Components:- Trip Wire, MCU Board, IoT cables.

Steps:-

Step 1: Click MCU board and go to Programming.



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



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 2

Aim: Demonstrate communication between two embedded devices using UART port.

Communication between I/O ports: Requirements:

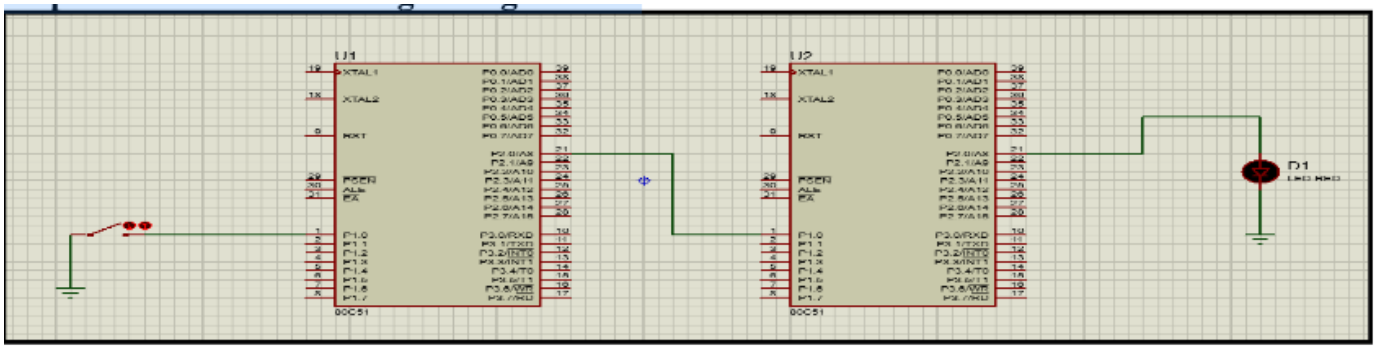
2 MCU (80C51)

1 SWITCH

2 GROUND Units

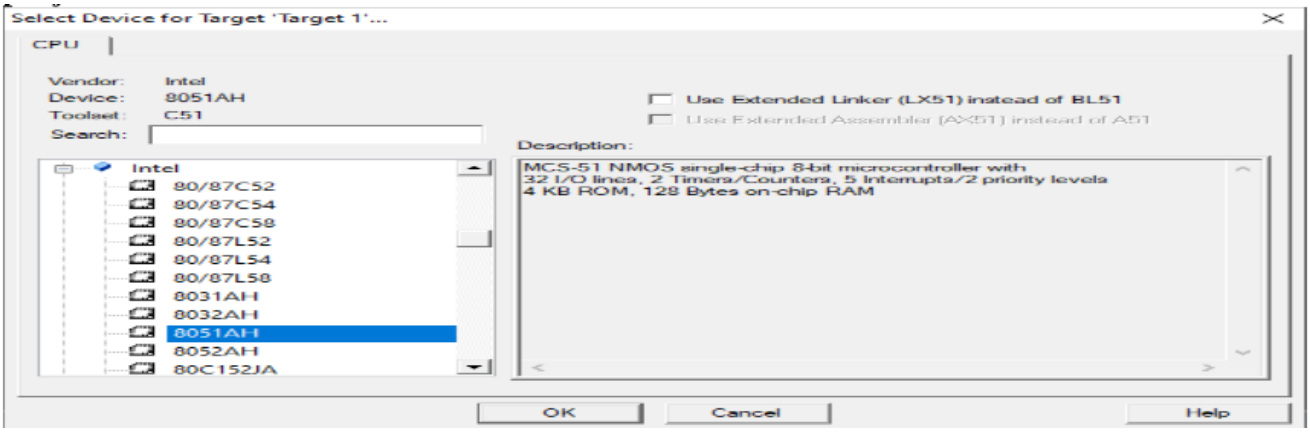
1 LED (RED)

Step 1: Make the following configuration:



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 Intel (8051AH) board and create the project.



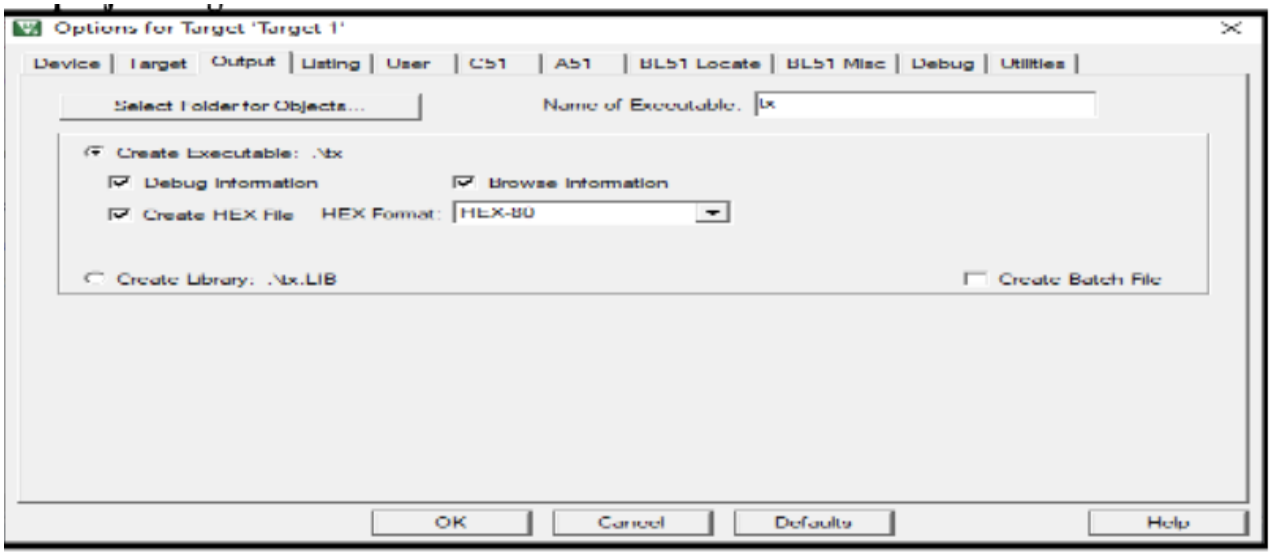
Step 3: Create a new file under ‘Source Group 1’ named ‘tx.c’. Add the following code to it.

The code for the TX microcontroller:

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

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

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

Step 4: Compile the code and generate the Hex file with the same name as the program file.

Make sure that the code is compiled with zero warnings and errors. Once that is done build the project to generate the Hex file.



Step 5: Delete the ‘tx.c’ file and make a new file named ‘rx.c’. Add the following code to it.

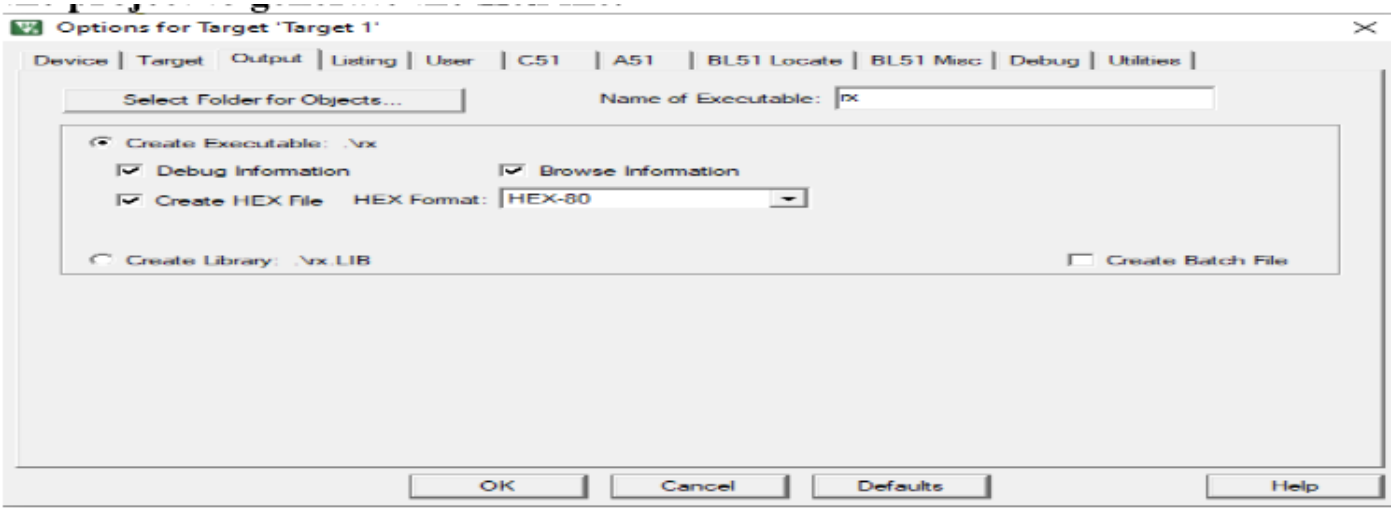
The code for the RX microcontroller:

```
#include<reg51.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
}
}
}
```

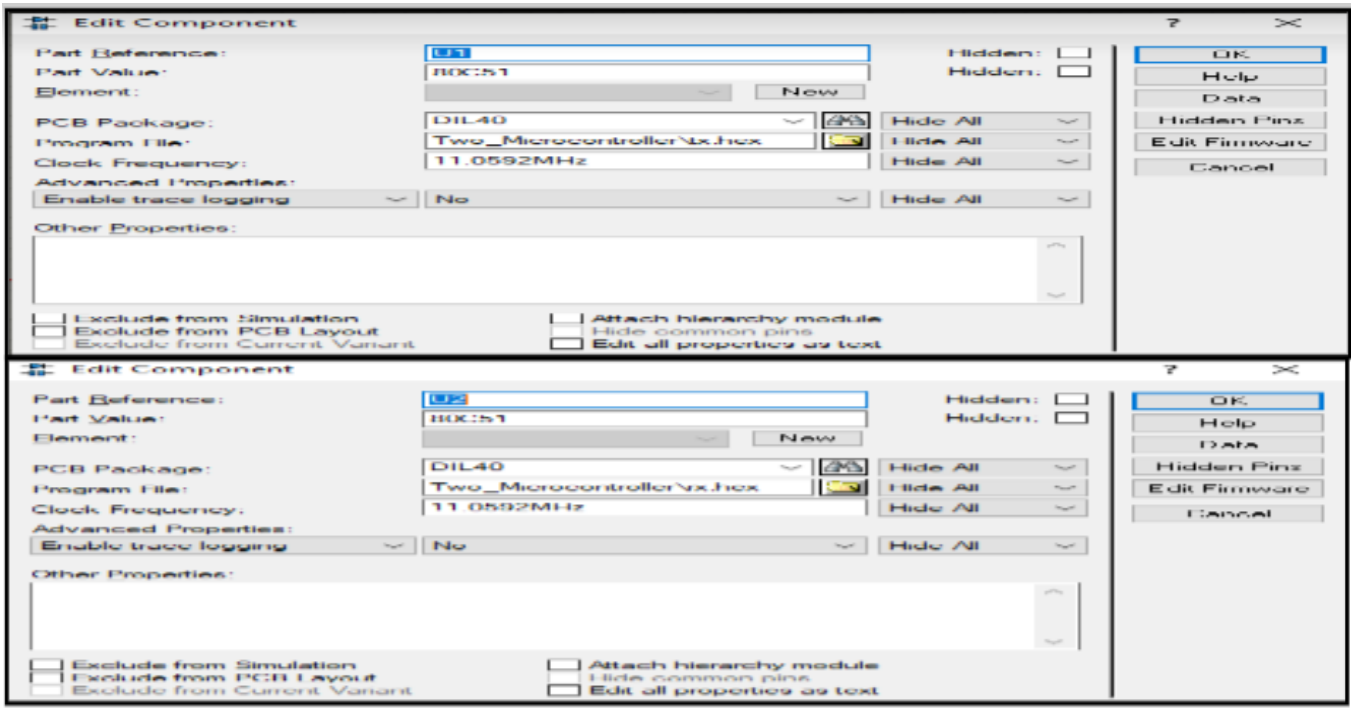
Step 6: Compile the code and generate the Hex file with the same name as the program file.

Make sure that the code is compiled with zero warnings and errors. Once that is done build

the project to generate the Hex file.



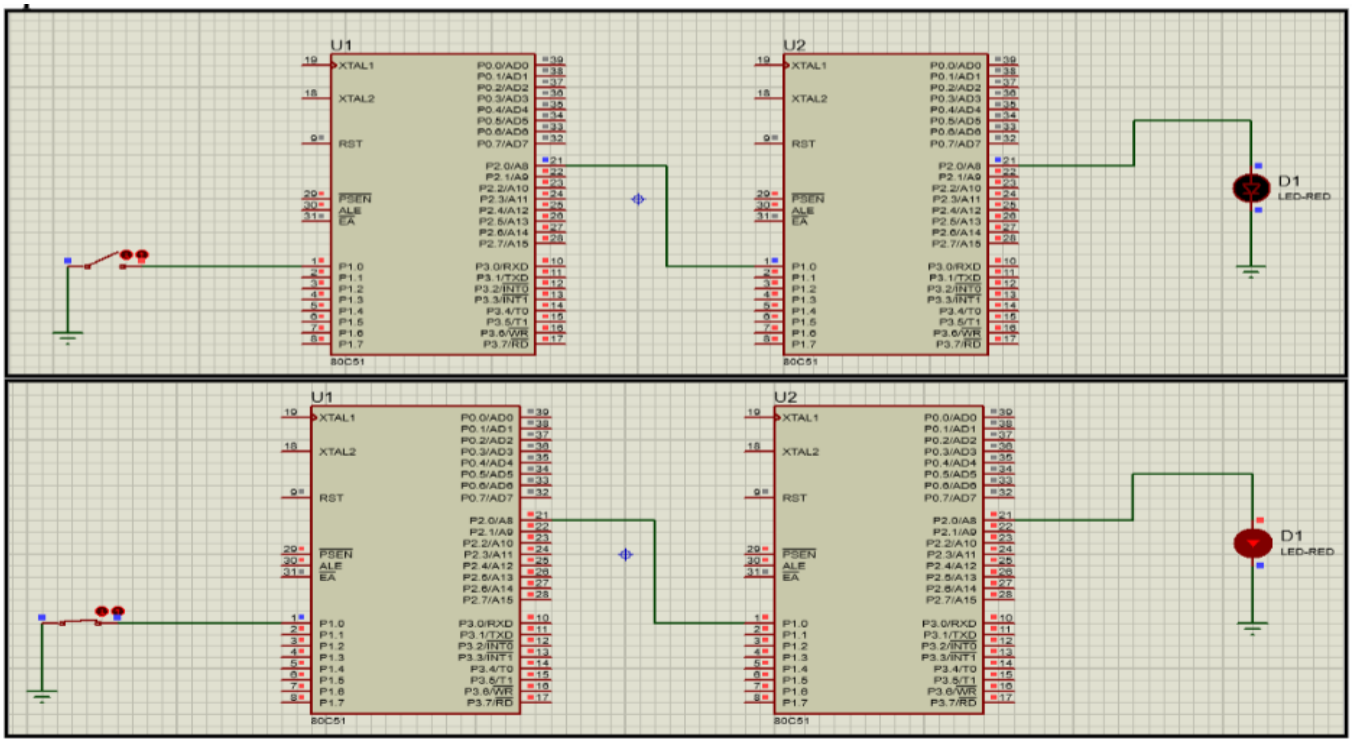
Step 7: Now flash the two Hex files in their respective microcontrollers.



Step 8: Make sure that the device is configured correctly and run the simulation. When the

switch is not connected a ‘low’ signal is sent between the two microcontrollers. But as soon as

you turn on the switch, a ‘high’ signal is sent between the two microcontrollers which lights up the LED.



Practical 3

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

Theory: IoT system to send a ticket before entering the bus is made 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 it selects the destination and place the card on sensors present on door and the door opens up and you can travel in this bus.

Components used for this part is

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

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.

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".

Click on RFID Reader, under Config tab, scroll down and select the IoT Sever to “HomeGateway”. Now, go to Advanced Tab I/O Config and set the Analog Ports to 2. Select the Network Adapter as “PT-IOT-NM-1W”.

16

Connect RFID Reader to MCU board using Analog port “A1”.
Connections to MCU board should be done using IOT CUSTOM CABLE ONLY.
Connection between RFID Reader Wirelessly to HomeGateway is made.
Click on Door, under Config Tab, select the IoT Sever to “HomeGateway” to connect to HomeGateway Wirelessly.
Click on PC, in the Physical tab, shut down the component. Scroll down and drag and drop the module already present in the component to the modules list and drag “PT-IOT-NM-1W” module from the list and drop it in the vacant space.
In the Config tab Wireless0 set SSID to “HomeGateway”. PC also gets connected to HomeGateway.
Click on RFID card, go to config, select IoT Server as “HomeGateway” and from advanced tab select the network adapter in Config as “PT-IOT-NM-1W”.

REQUIREMENTS:

Component	Name Use
	This card is used as a key to enter in the bus. (ACTUATOR). RFID Card
	This component is connected on the door which reads RFID card RFID Reader card. (SENSOR).
 	This component is used for connecting Sensors to MCU Board Actuators. This are used to interact with humans i.e it displays results
	The door Opens up when RFID card is scanned successfully. Door
	homegateway.
	This is used to connect all the smart devices to the controller device wirelessly. 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()
```

Open the RFID reader Programmin select RFID Reader Python main.py

In the def loop(), under sendReport() part, make the following changes :monitor.

```
sendReport()
#indent properly, write under else from above
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)
```

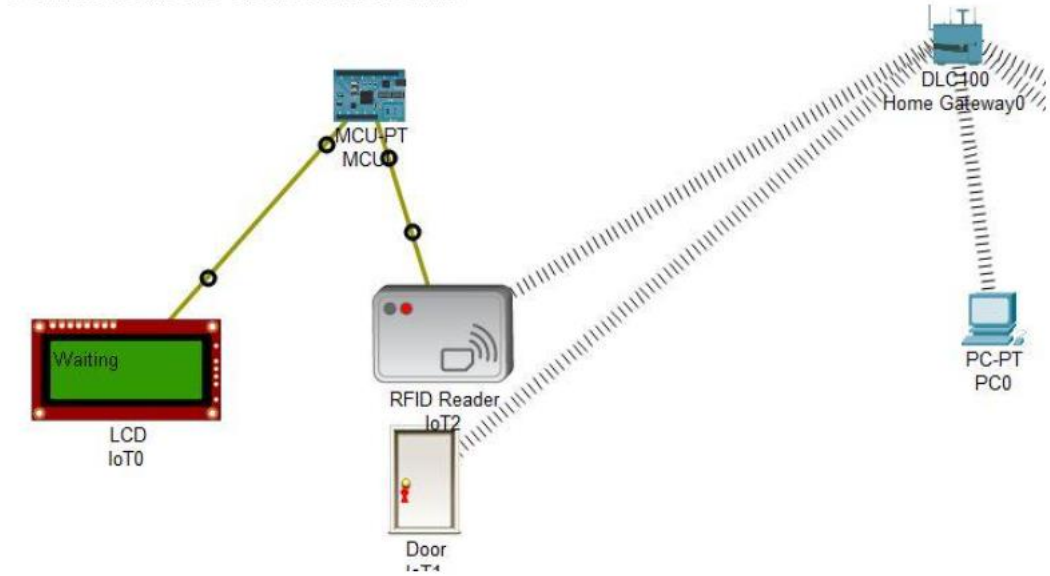
run the code.

```
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.

Connections are done as follows:

Connections are done as follows:



run the code.

Practical 04

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)

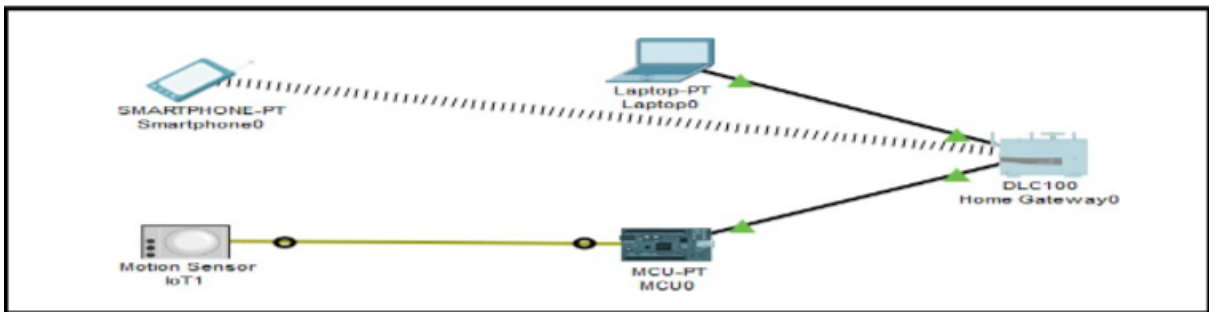
1 Microcontroller Unit (MCU-PT)

1 Home Gateway (DLC100)

1 Laptop (Laptop-PT)

1 Smartphone (SMARTPHONE-PT)

Step 1: Select the requirements and make their configurations.



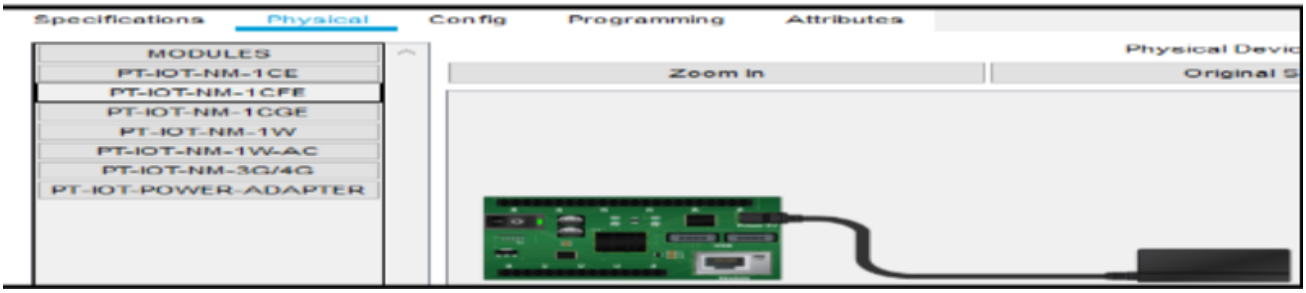
Motion Sensor:

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.

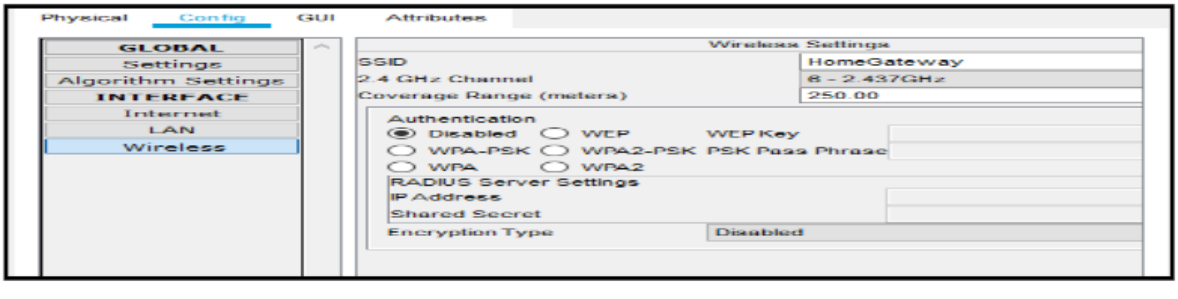


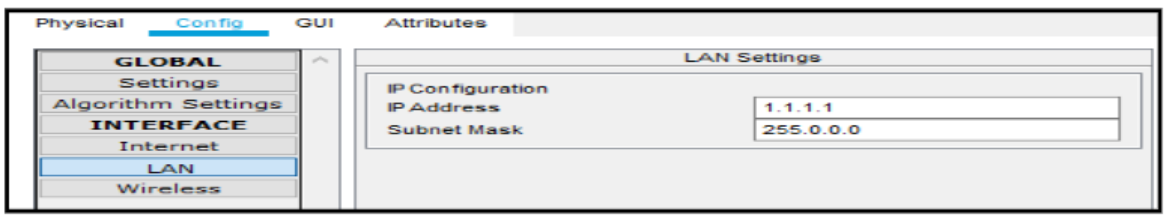
Connect the MCU-PT to the Home Gateway using a copper wire and configure its connection settings.

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.

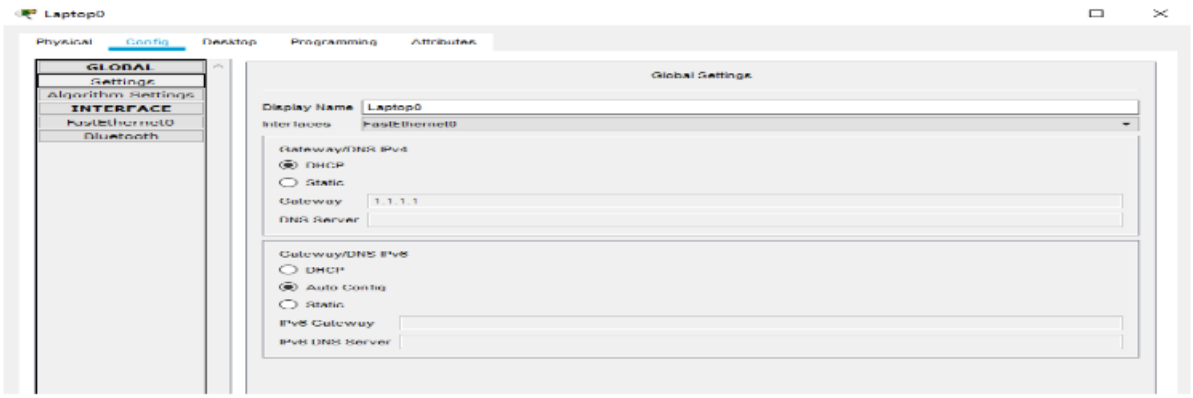
Home Gateway: Make the following configurations to the Home Gateway and connect it to the laptop via a copper cable.





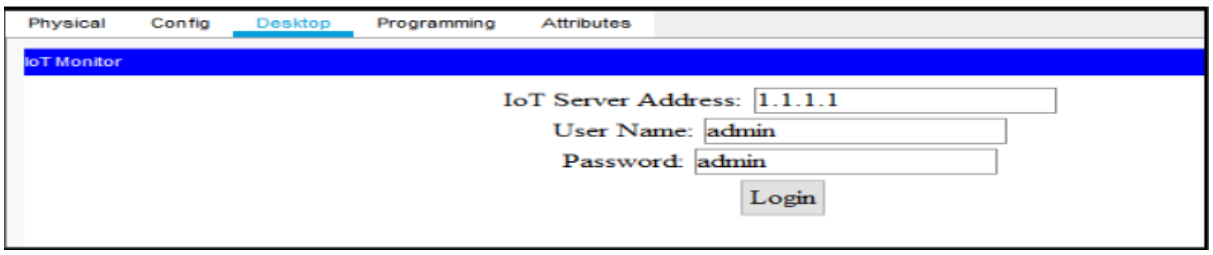
Laptop:

Make the following configuration to the laptop.



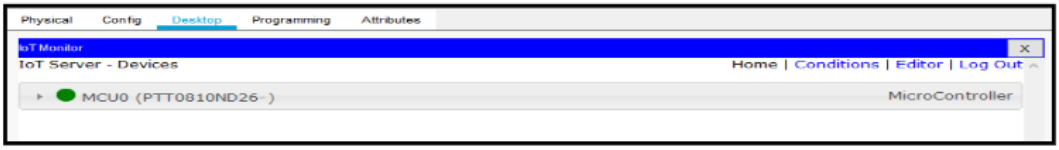
Laptop:

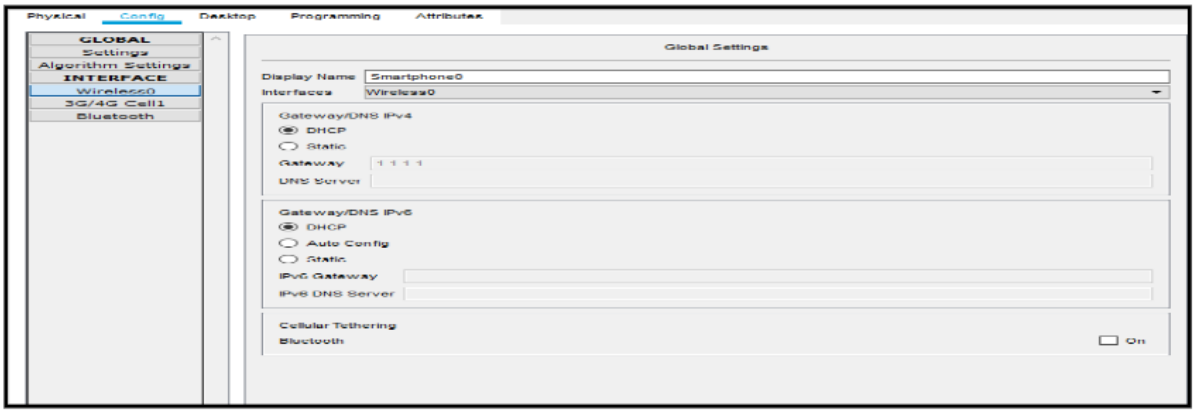
Make the following configuration to the laptop.



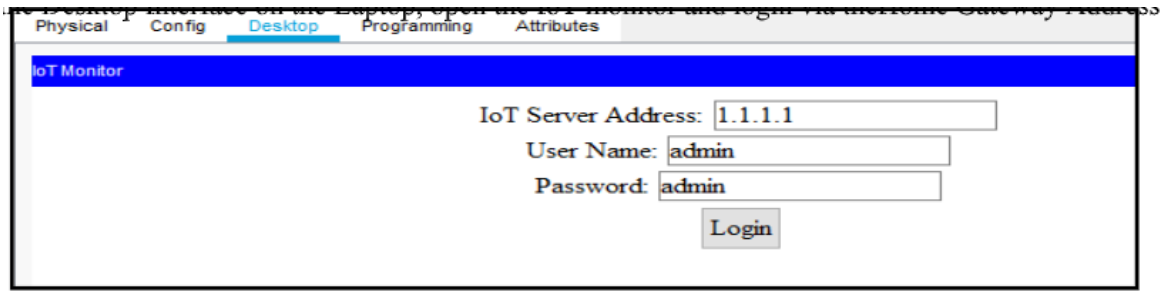
Smartphone:

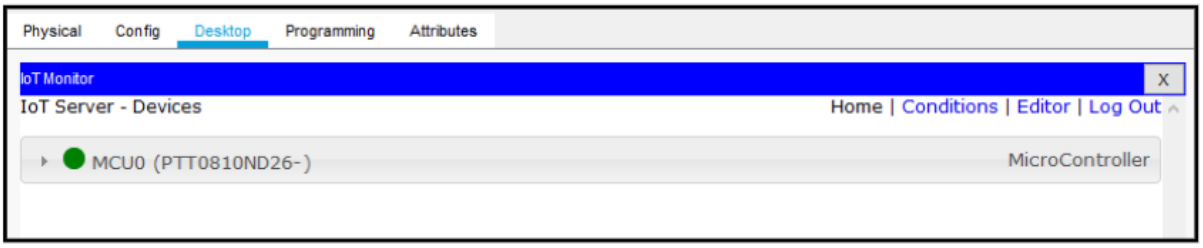
Make the following configuration to the laptop..





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

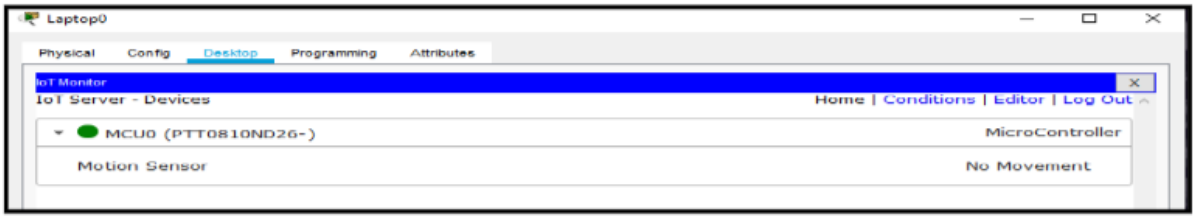


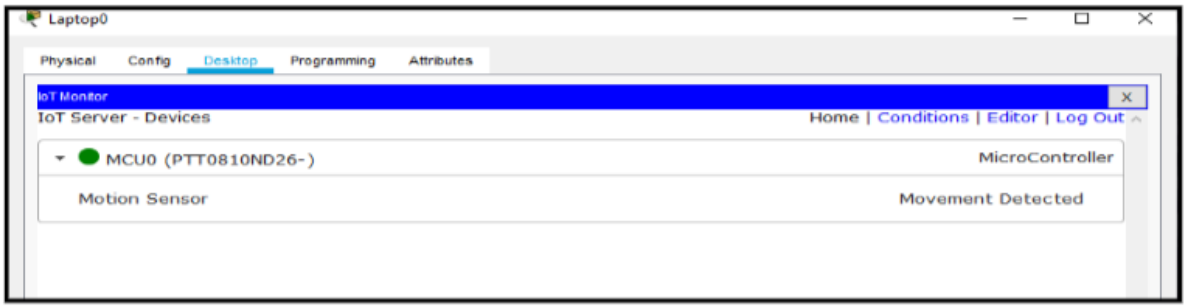


Step 2: Check whether the motion sensor works by observing the IoT monitor on the Laptop

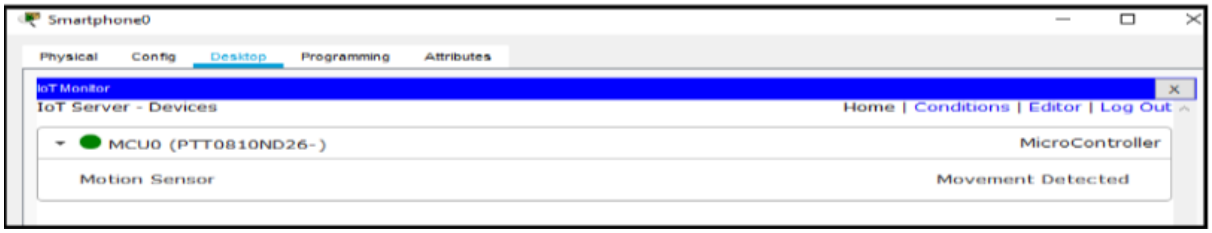
and the Smartphone.

Laptop:

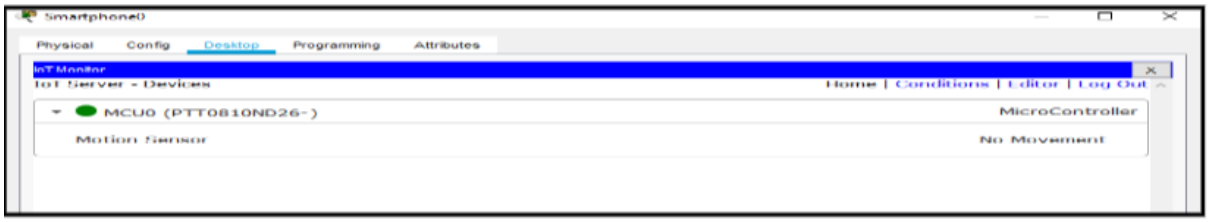




Smartphone:



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



Practical 5

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

Theory: -

Humidity is the concentration of water vapor 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

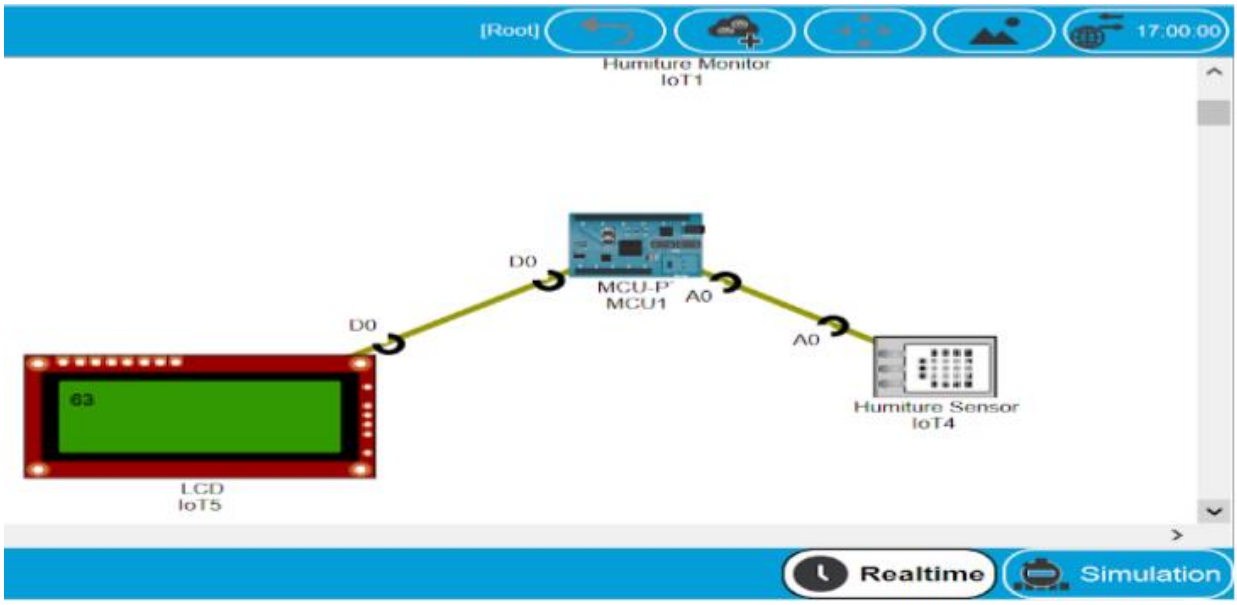
example, it can rain at a ground humidity of 60%, but over time, the humidity will increase.

Components required:-

LCD, MCU board, humiture sensor

Steps:-

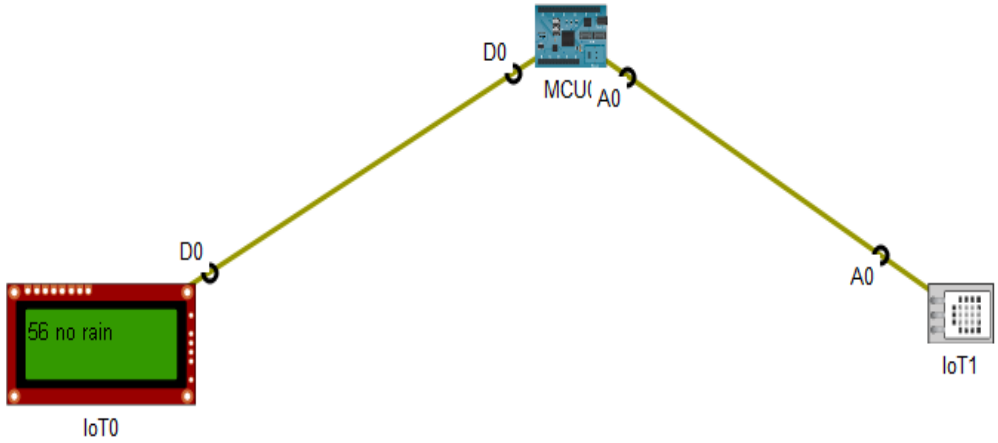
Take the components and arrange as shown in diagram



Now code a java script to display the value of humiture sensor on LCD monitor.

```
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); }
```

Run the code to see the value of sensed data on LCD monitor.



Practical 6

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

Requirements:

1 Power Supply

2 Grounds

1 Arduino UNO

1 LCD Screen (LM016L)

1 Torch and Light Dependent Resistor (LDR)

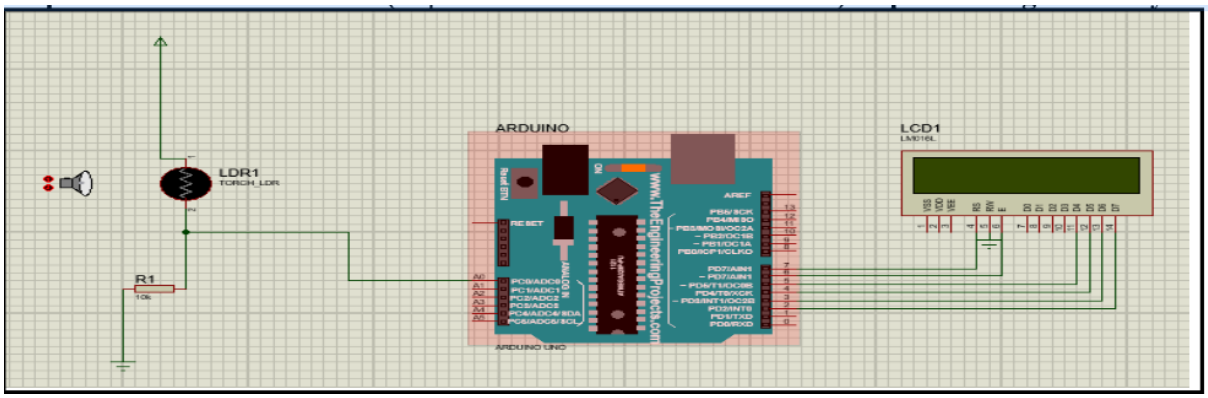
1 Resistor 1kΩ

Step1 : Install Arduino Libraries for Proteus to use

(https://www.theengineeringprojects.com/2015/12/arduino-library-proteus-simulation.html)

Extract the files in to the Library Folder of Proteus.

Step 2: Install Arduino IDE. (https://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.

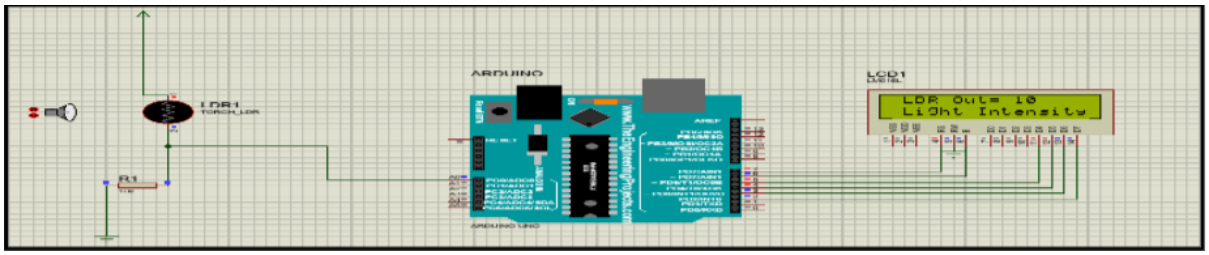
Code:

```
#include <LiquidCrystal.h>

//Initialize libraries with the pins interface
LiquidCrystal lcd(7, 6, 5, 4, 3, 2);

void setup() {
  // Set up the LCD number of columns and rows
  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);

}
```

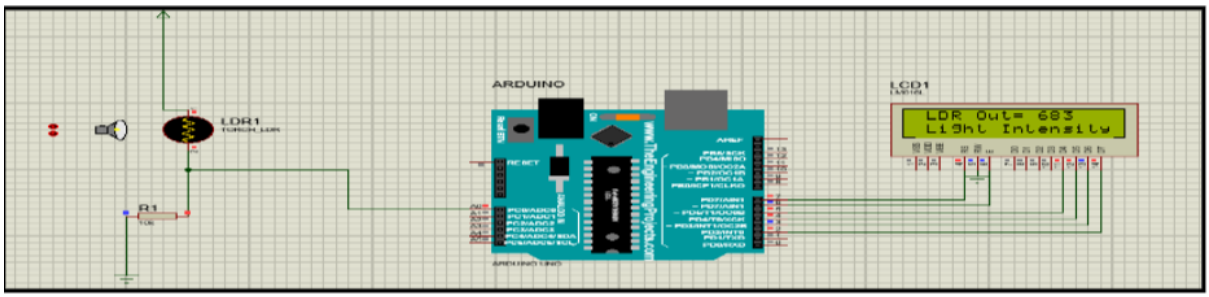
Note: Make sure that you compile the code. The generated hex file for proteus will be generated in the location (C:\Users\rudra\AppData\Local\Temp\arduino_build_333966\ldr.ino.with_bootloader.hex). Upload this code to the Arduino UNO in Protes by double clicking on the MCU and adding the path of the .hex file to the program file.

Step 5: Run the simulation and make your observations.

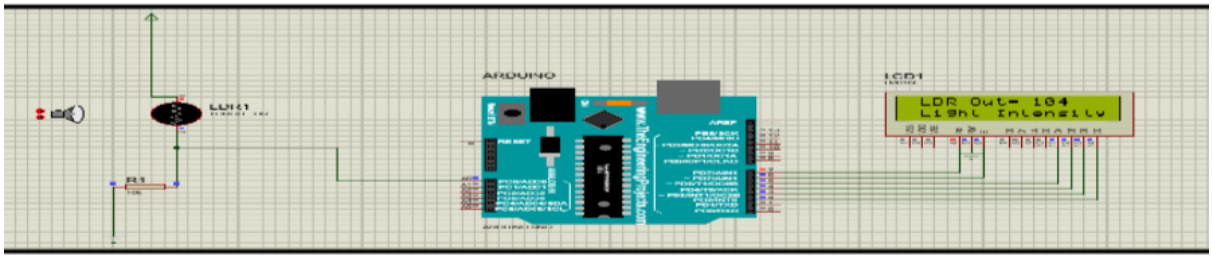
When we run the simulation there is a fixed intensity of light measure on the LCD screen.



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.

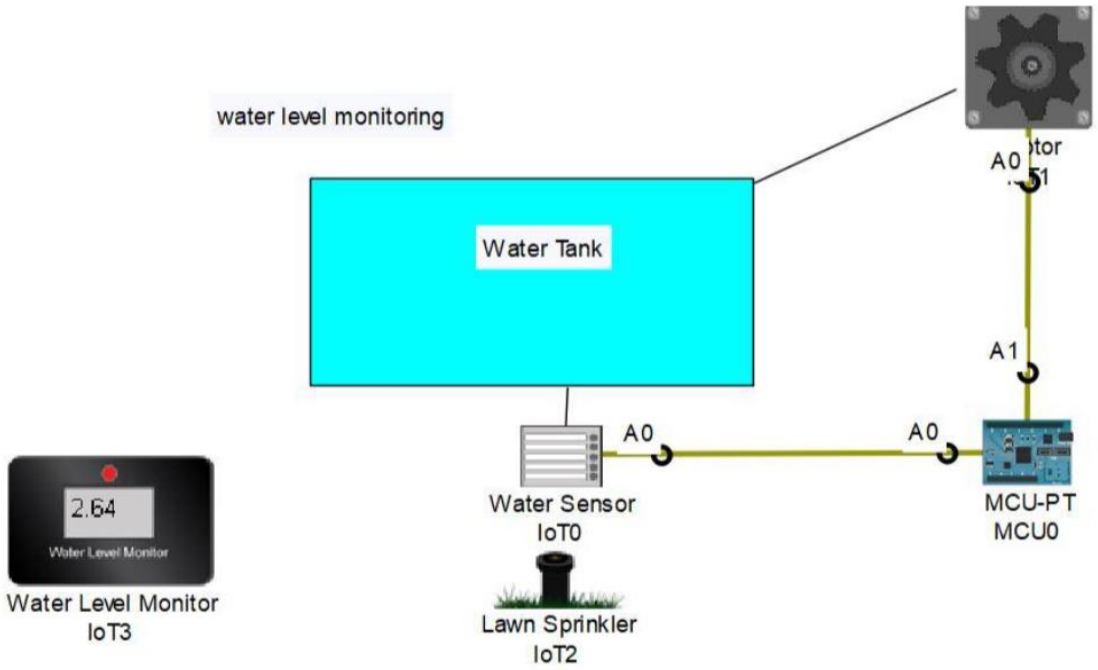


Practical 7

Aim: Develop and IOT application for monitoring water level in tanks and automatically start the motor to fill the tank if the level goes below the critical level.

Theory:

Long-term, systematic measurements of water levels provide essential data needed to evaluate changes in the resource over time, to develop ground-water models and forecast trends, and to design, implement, and monitor the effectiveness of water management and protection programs. When the tank in a home is managed for sprinkling water in the lawn, sensors are given for various works like sensing the texture of ground, heat, etc. The sprinkler heads work on a pressure system that when the water goes through the pipe, it pushes them up above the ground. When the water pressure stops, they retract back down to ground level.



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

understanding and functionality easier.

3. Create a rectangle that represents the Water Tank.

PROGRAM THE MCU -

1. Click on the MCU and go to Programming > New and create a new Python

Script called ‘ Water Monitor ‘.

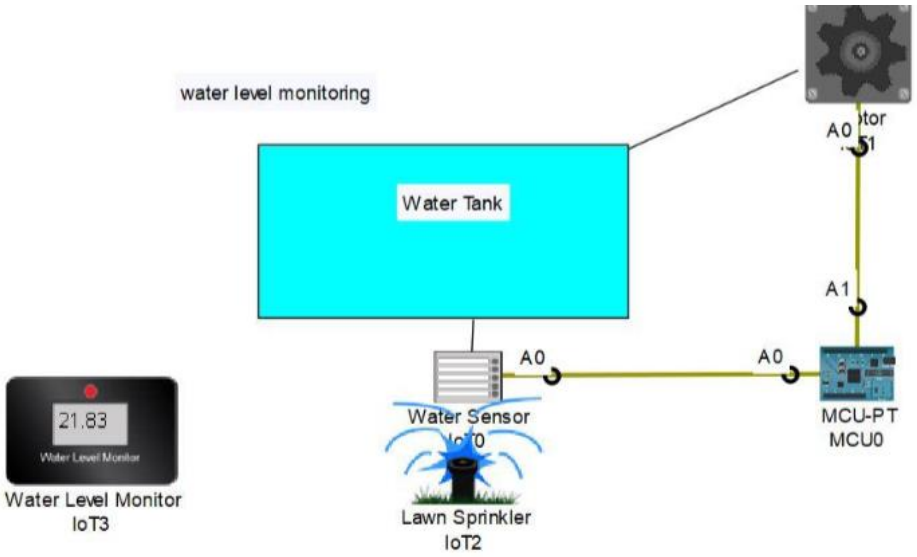
2. Write the following code in the main.py file –

```
"""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 -

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

allow the tank to fill up.



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.

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 8

Aim: Develop an IOT application for motion detection.

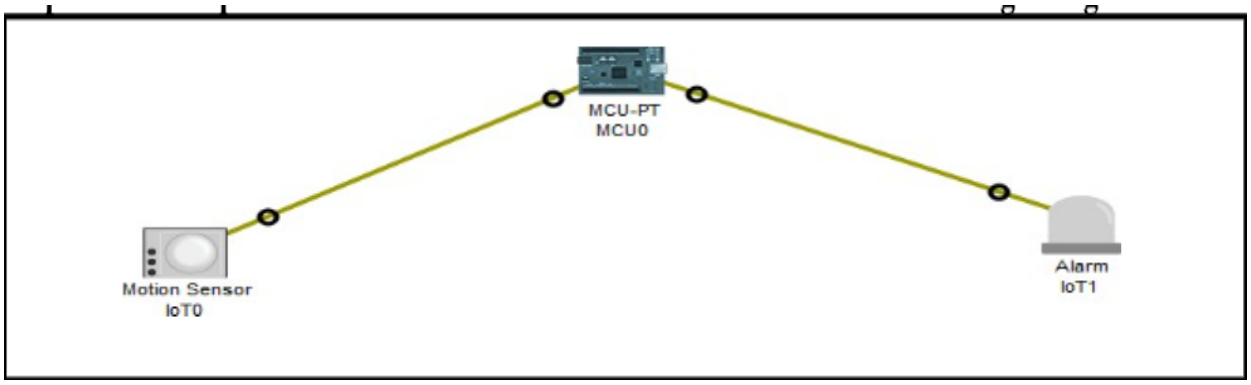
Requirements:

1 Motion Sensor

1 MCU Board

1 Alarm

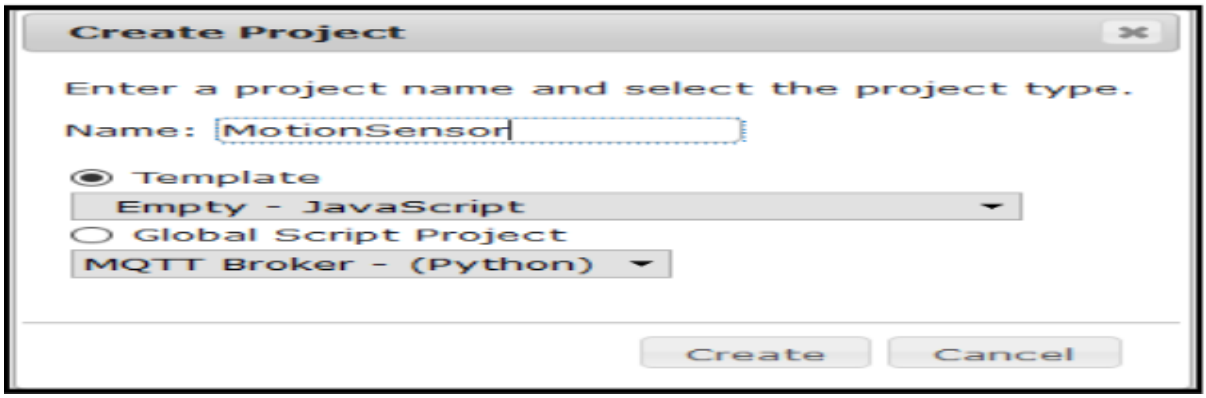
Step 1: For this practical use Cisco Packet Tracer. Now make the following configuration.



Note: All the components are connected via the 'IoT Custom Cable'.

Step 2: Configure the MCU Board:

Create a new project under programming.



Write the following code within it.

CODE

//Setup Pins of the Microcontroller

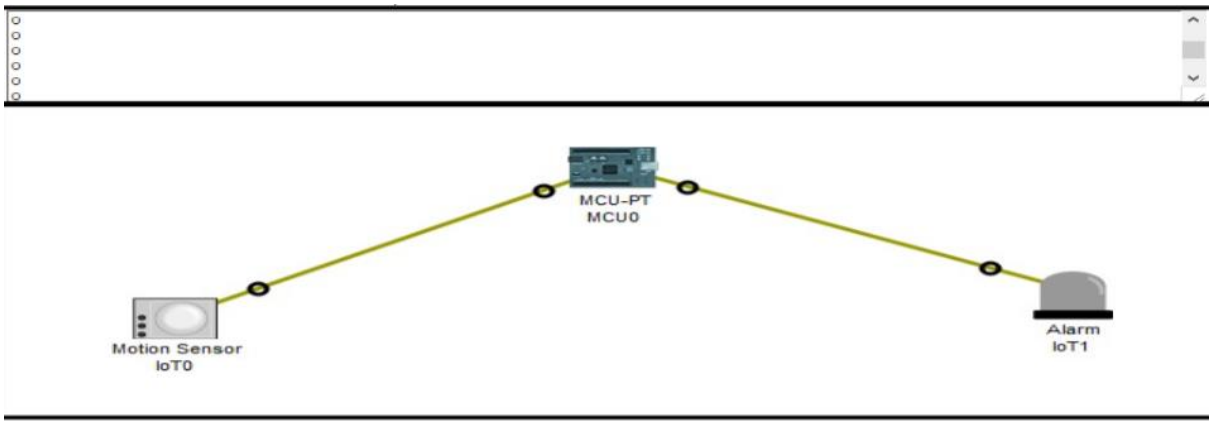
```
function setup () {
pinMode (0, INPUT); //First Pin
pinMode (1, OUTPUT) ; //Second Pin
}
//Using a Loop to continously read the data from the pins
function loop (){
var motion = digitalRead (0);
Serial.println(motion);
//To raise the alarm (On/Off)
if (motion == 1023) {
digitalWrite (1, HIGH);
}
else {
digitalWrite (1, LOW);
}
}
```

Step 3: Run the code.

To use the motion sensor press the left alt on your keyboard followed by the hovering of the mouse over the

motion sensor.

When there is no motion detected, the out received in the terminal will be ‘0’ and the alarm will not turn on.



When the motion sensor detects any motion the output received on the terminal will be ‘1023’ and the alarm will be turned on.