IOT DOMAIN ANALYST

LAB ASSESSMENT-1

Name: SREENIVASAN S

Reg No: 21BEC0256

AIM

1. Connect 5 devices to a hub and switch and note down the differences between a hub and a switch Write down the IP addresses of all the connected end devices.
2. Write an R-Pi code to detect the presence of a target object in a military application and measure the distance at which the target object is located along with a LED alert notification .
3. Build an R-Pi based prototype and develop a code to detect the presence of an obstacle in a robotic path planning project using suitable sensor and use a buzzer to indicate the presence of an obstacle.

COMPONENTS/TOOLS REQUIRED

1.CISCO PACKET TRACER SOFTWARE

2.RASPBERRY PI

3.BREADBOARD

4.BLINKING LED

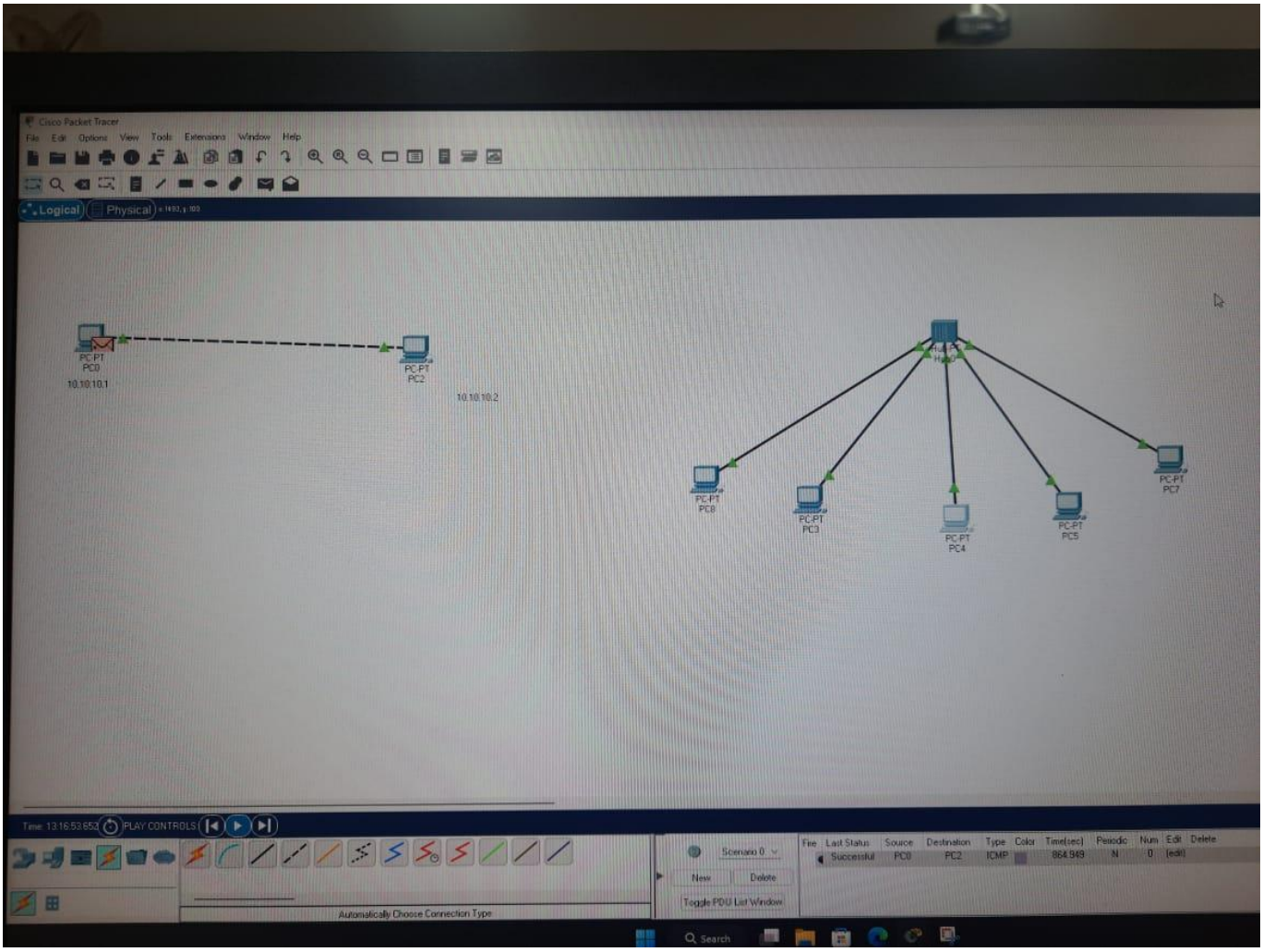
5.CONNECTING WIRES

6.IR SENSOR

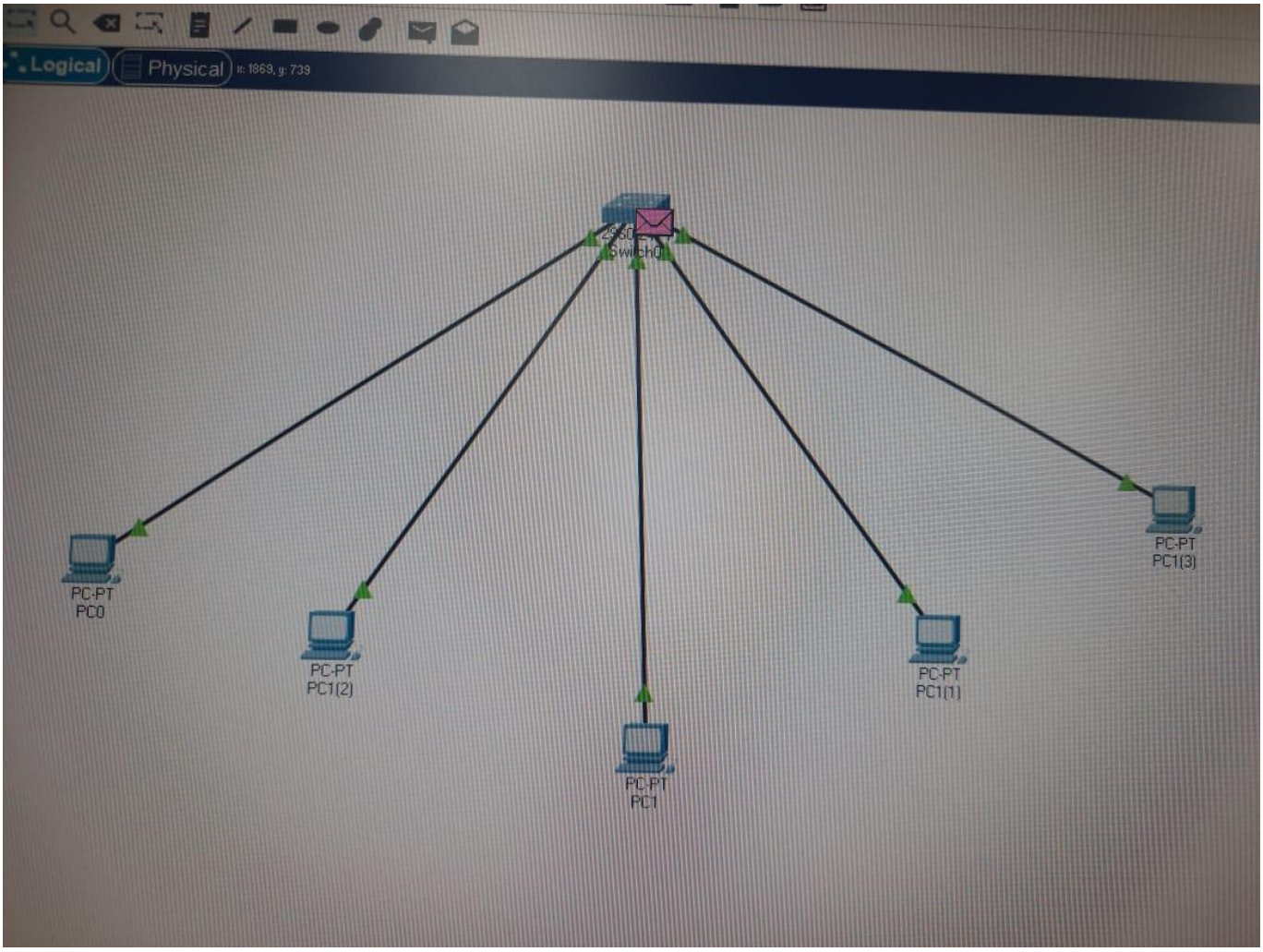
7.ULTRASONIC SENSOR

CIRCUIT/BLOCK DIAGRAM

HUB



SWITCH



IP ADDRESS of each node

NODE	IP ADDRESS
PC0	10.10.10.1
PC1	10.10.10.2
PC2	10.10.10.3
PC3	10.10.10.4
PC4	10.10.10.5

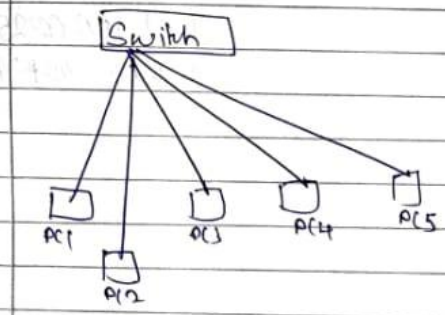
Differences between a hub and a switch

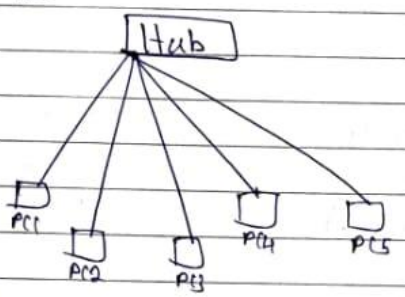
HUB	SWITCH
Operates at the physical layer of OSI model	Operates at the data link layer (Layer 2) of OSI model
Broadcasting data to all connected devices,	forwarding data only to intended recipients based on MAC.
Potentially causing network congestion.	addresses, reducing congestion and improving efficiency.
Creates a single collision domain for all	Creates separate collision domains for each connected
Connected devices, leading to collisions	device, preventing collisions and enabling simultaneous.
When multiple devices transmit simultaneously.	Data transmissions.

__/__/____

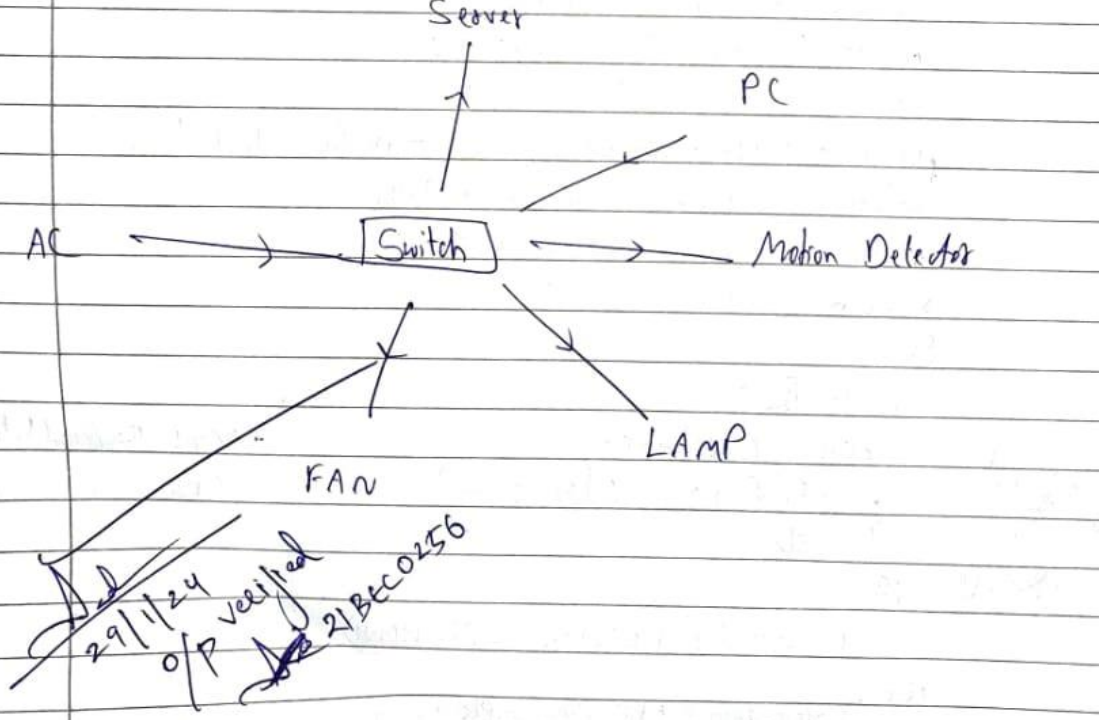
8/1/23

Name: SREENIVASAN.S
Reg: 21BEC0256

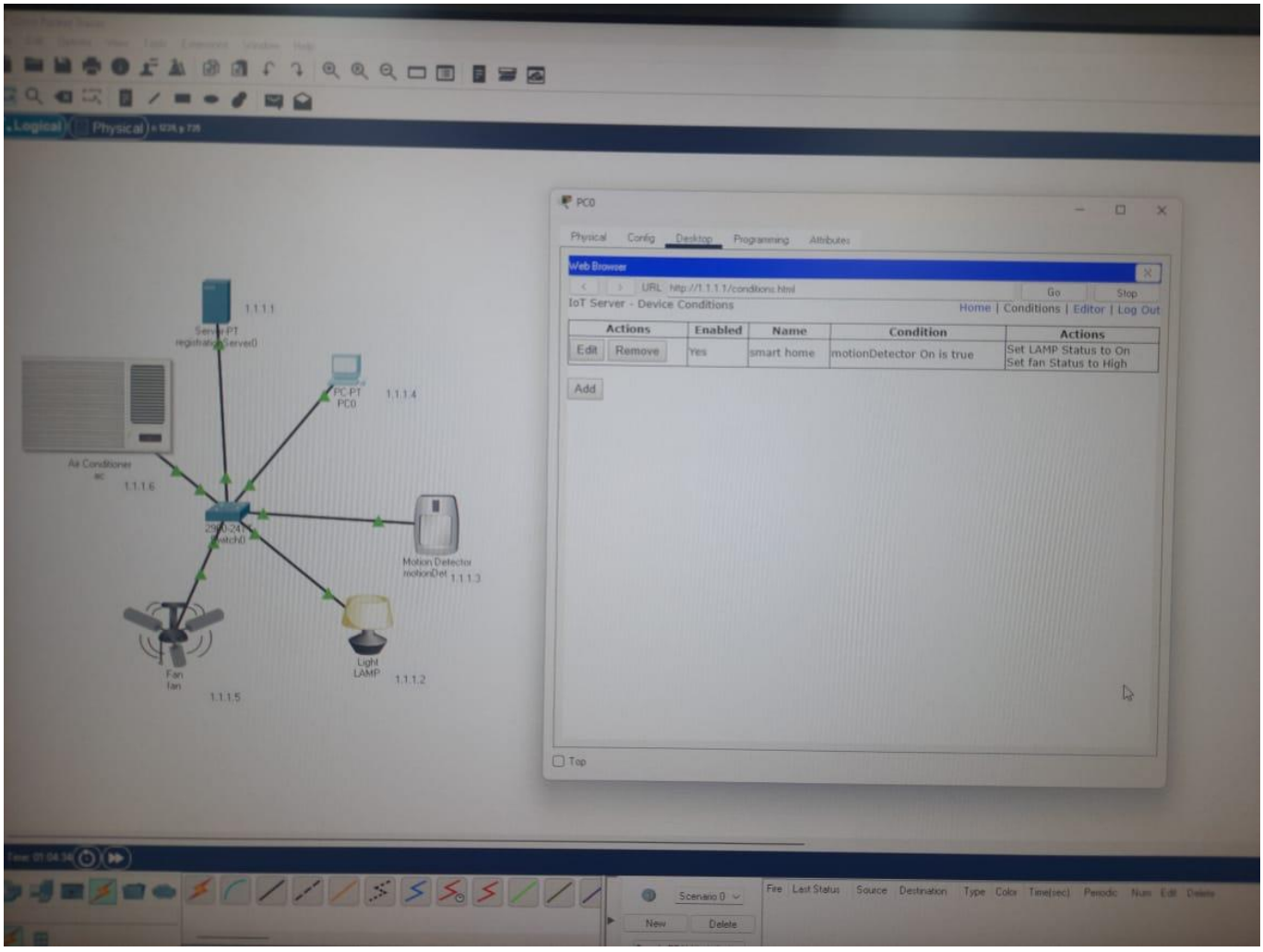


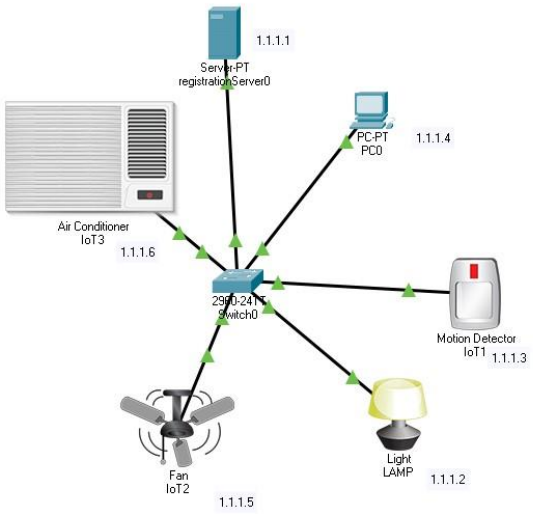


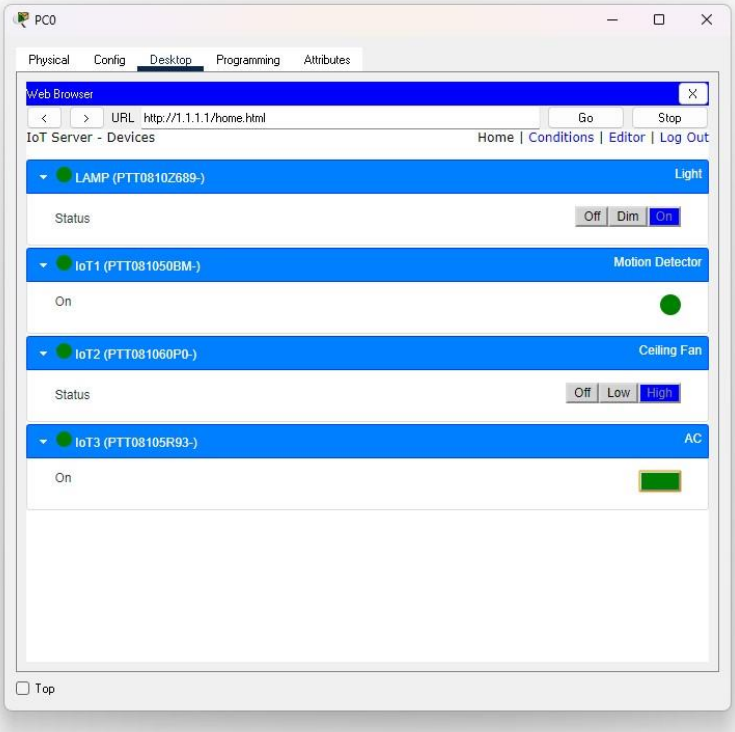
Verified
Date 8/1/24



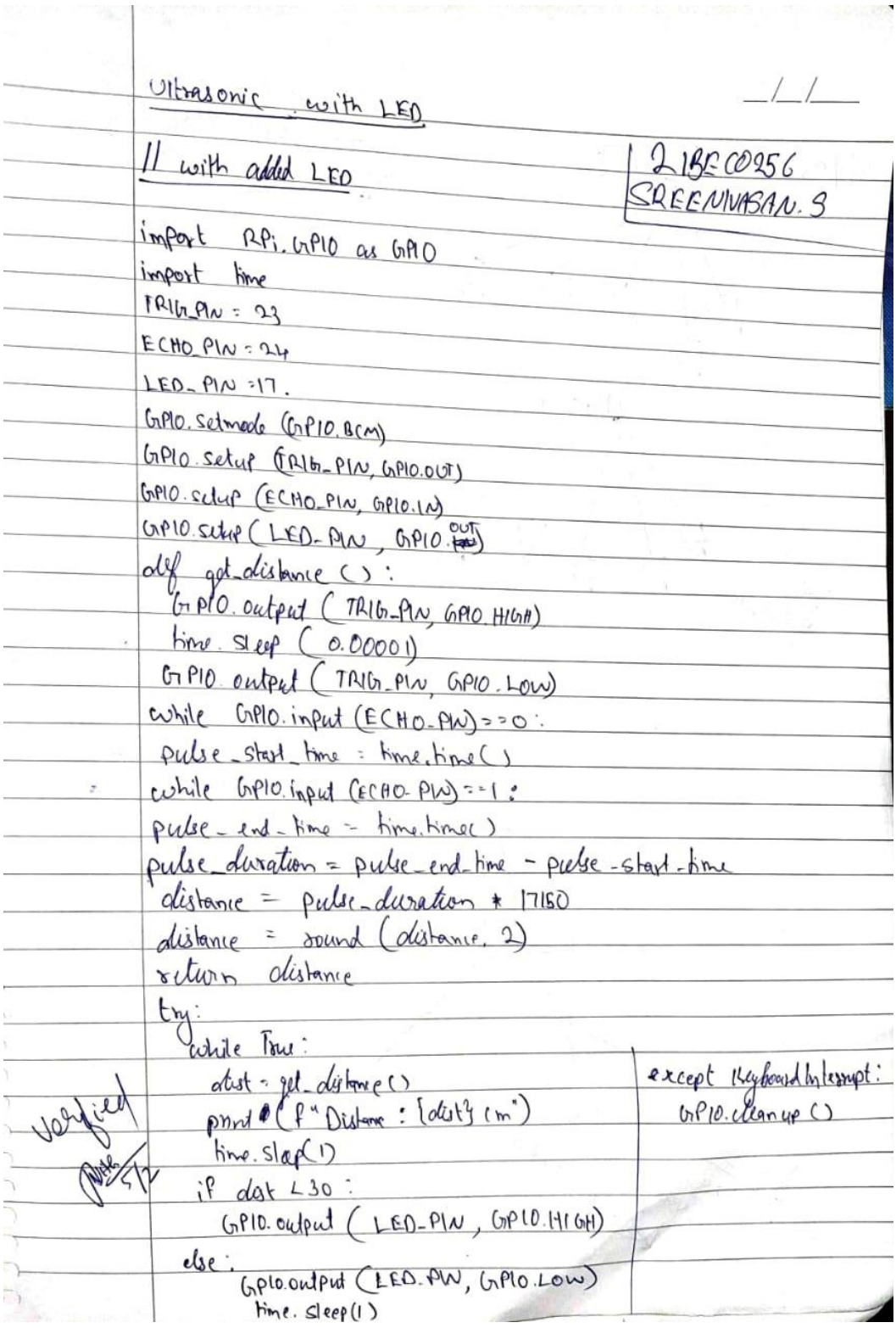
29/1/24
O/P verified
21BEC0256


Logical
Physical
PC0
Physical
Config
Desktop
Programming
Attributes
Web Browser
URL http://1.1.1.1/conditions.html
Go
Stop
IoT Server - Device Conditions
Home | Conditions | Editor | Log Out
Actions
Enabled
Name
Condition
Actions
Edit
Remove
Yes
smart home
motionDetector On is true
Set LAMP Status to On
Set fan Status to High
Add
Top
1.1.1.1
1.1.1.4
1.1.1.6
1.1.1.3
1.1.1.2
1.1.1.5
Air Conditioner
ac
Motion Detector
motionDet
Light
LAMP
Fan
fan
Scenario 0
New
Delete
Fire
Last Status
Source
Destination
Type
Color
Time(sec)
Periodic
Num
Edit
Delete


1.1.1.1
Server-PT
registrationServer0
PC-PT
PC0
1.1.1.4
Air Conditioner
IoT3
1.1.1.6
2960-24TT
Switch0
Motion Detector
IoT1
1.1.1.3
Fan
IoT2
1.1.1.5
Light
LAMP
1.1.1.2


PC0
Physical
Config
Desktop
Programming
Attributes
Web Browser
URL http://1.1.1.1/home.html
Go
Stop
IoT Server - Devices
Home | Conditions | Editor | Log Out
LAMP (PTT0810Z689-)
Light
Status
Off
Dim
On
IoT1 (PTT081050BM-)
Motion Detector
On
IoT2 (PTT081060P0-)
Ceiling Fan
Status
Off
Low
High
IoT3 (PTT08105R93-)
AC
On
Top

2. Write an R-Pi code to detect the presence of a target object in a military application and measure = the distance at which the target object is located along with a LED alert notification.



Ultrasonic with LED

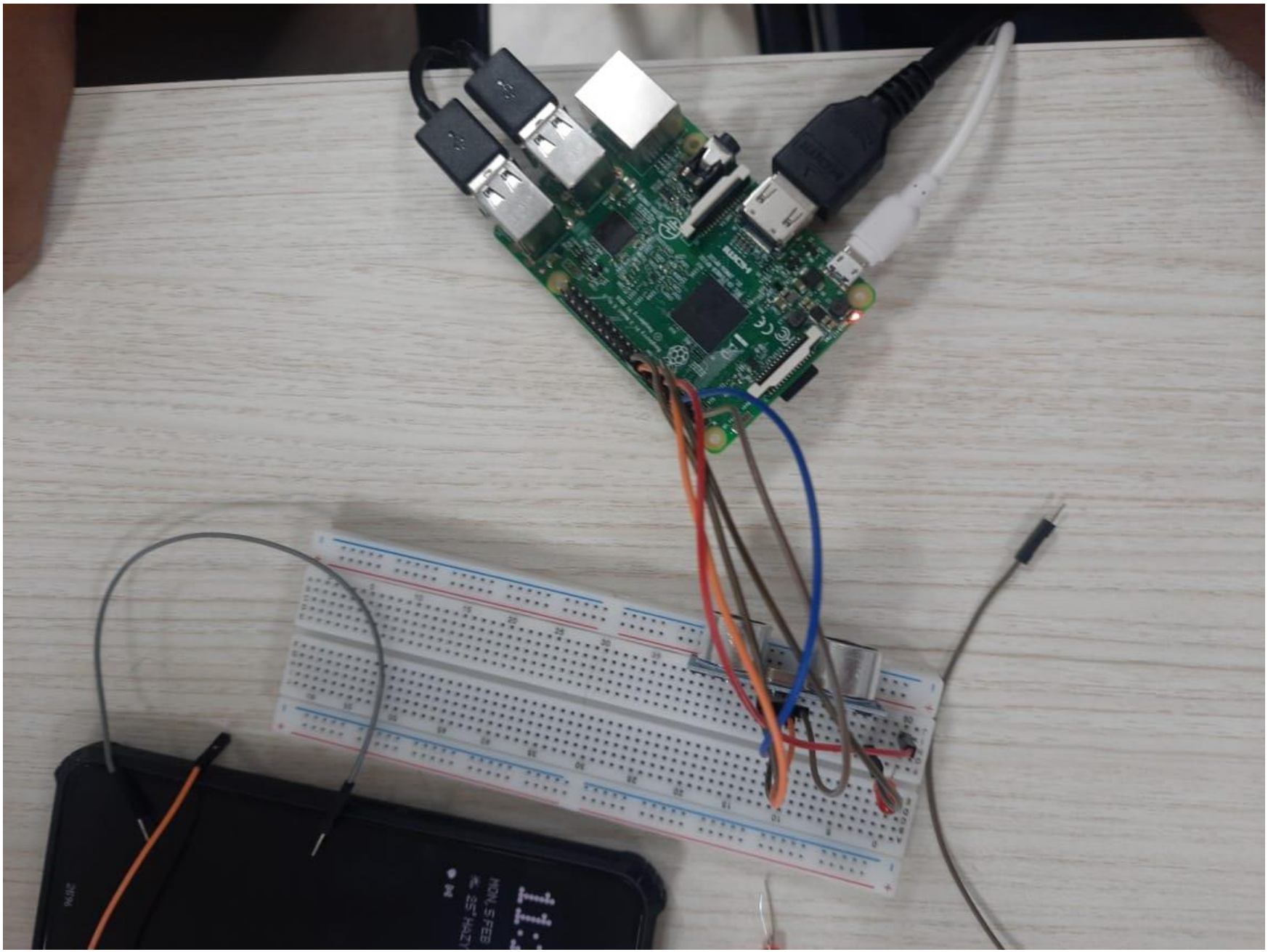
__/__/__

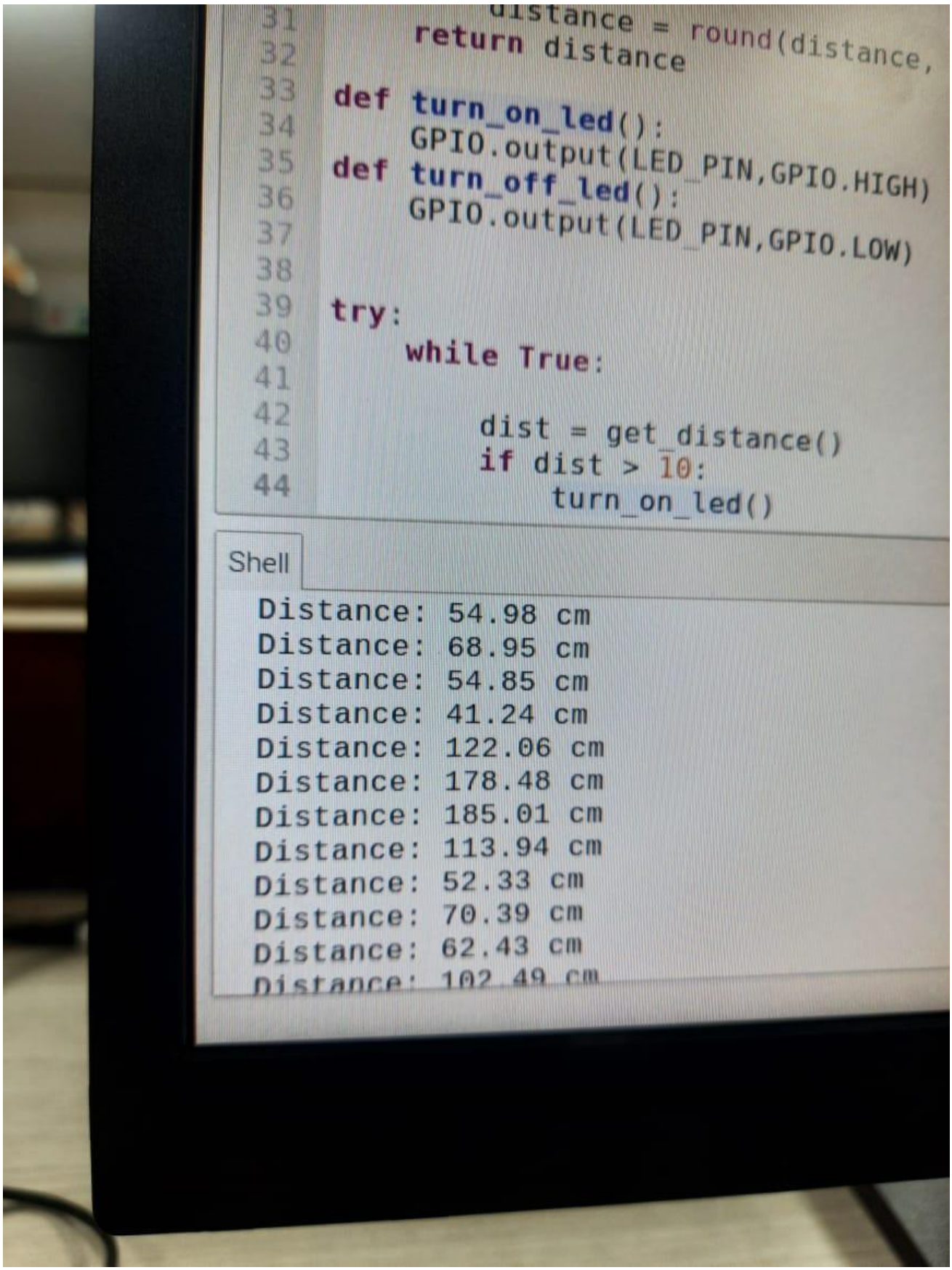
2 1BE CO256
SREENIVASAN. S

// with added LED

```
import RPi.GPIO as GPIO
import time
TRIG_PIN = 23
ECHO_PIN = 24
LED_PIN = 17.
GPIO.setmode (GPIO.BCM)
GPIO.setup (TRIG_PIN, GPIO.OUT)
GPIO.setup (ECHO_PIN, GPIO.IN)
GPIO.setup (LED_PIN, GPIO.OUT)
def get_distance ():
    GPIO.output (TRIG_PIN, GPIO.HIGH)
    time.sleep (0.00001)
    GPIO.output (TRIG_PIN, GPIO.LOW)
while GPIO.input (ECHO_PIN) == 0:
    pulse_start_time = time.time()
while GPIO.input (ECHO_PIN) == 1:
    pulse_end_time = time.time()
pulse_duration = pulse_end_time - pulse_start_time
distance = pulse_duration * 17150
distance = round (distance, 2)
return distance
try:
    while True:
        dist = get_distance()
        print (f"Distance: {dist} cm")
        time.sleep(1)
        if dist < 30:
            GPIO.output (LED_PIN, GPIO.HIGH)
        else:
            GPIO.output (LED_PIN, GPIO.LOW)
            time.sleep(1)
except KeyboardInterrupt:
    GPIO.cleanup()
```

verified



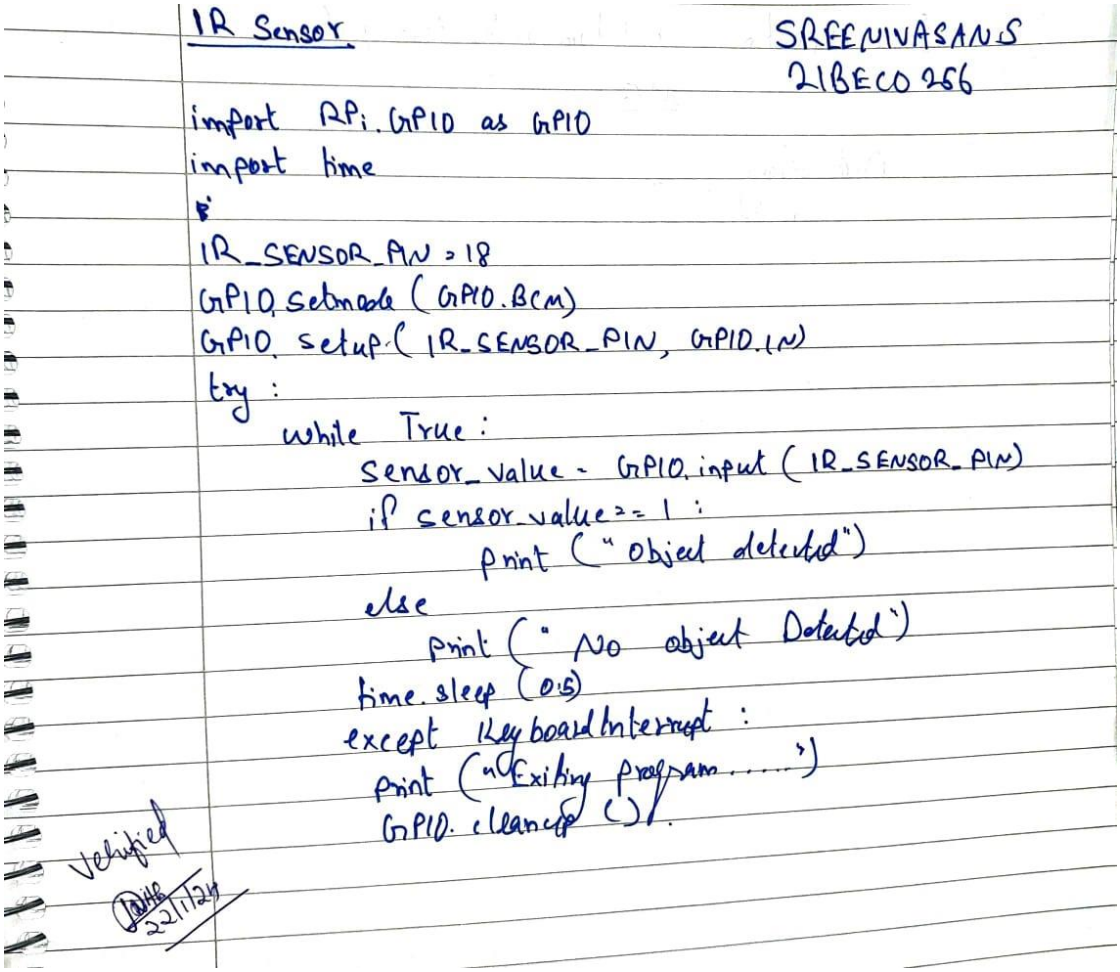


```
        distance = round(distance,
    return distance

def turn_on_led():
    GPIO.output(LED_PIN,GPIO.HIGH)
def turn_off_led():
    GPIO.output(LED_PIN,GPIO.LOW)


try:
    while True:

        dist = get_distance()
        if dist > 10:
            turn_on_led()
```

Shell

```
Distance: 54.98 cm
Distance: 68.95 cm
Distance: 54.85 cm
Distance: 41.24 cm
Distance: 122.06 cm
Distance: 178.48 cm
Distance: 185.01 cm
Distance: 113.94 cm
Distance: 52.33 cm
Distance: 70.39 cm
Distance: 62.43 cm
Distance: 102.49 cm
```

3. Build an R-Pi based prototype and develop a code to detect the presence of an obstacle in a robotic path planning project using suitable sensor and use a buzzer to indicate the presence of an obstacle.



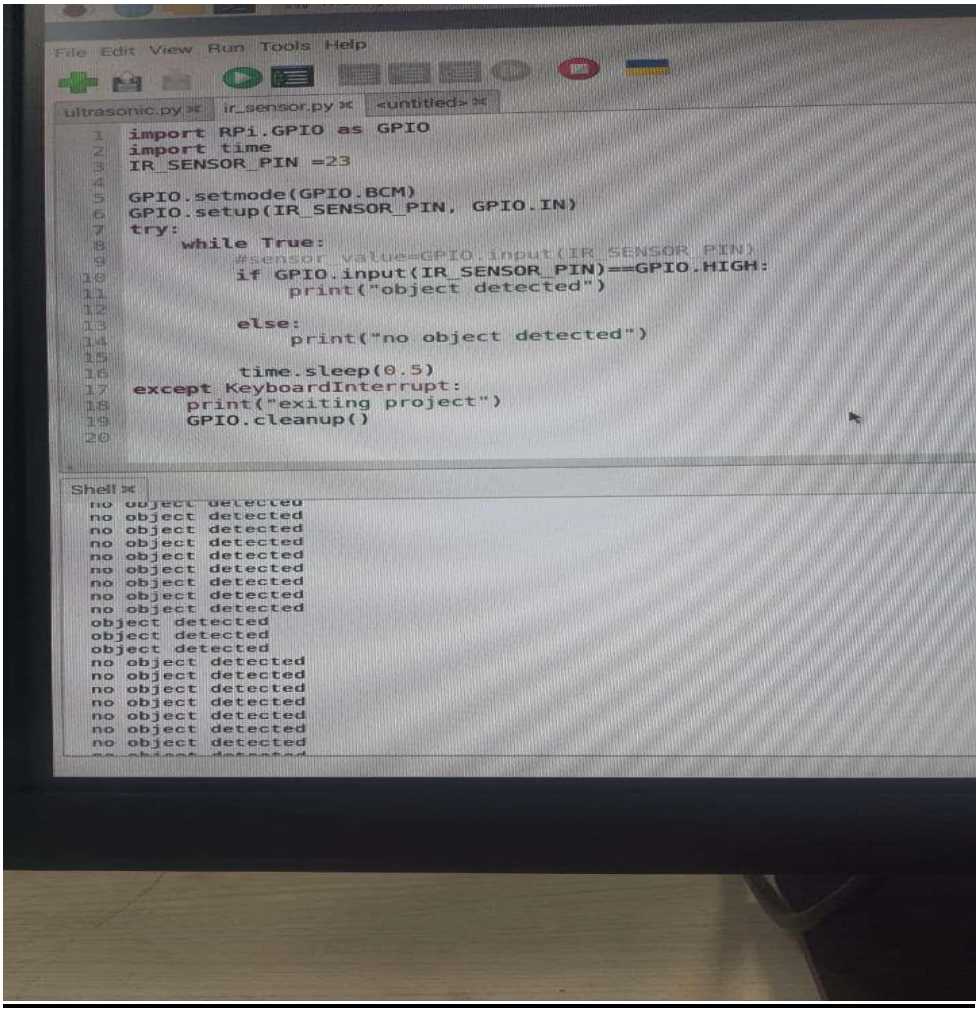
IR Sensor

SREENIVASAN S
21BECO256

```
import RPi.GPIO as GPIO
import time

IR_SENSOR_PIN = 18
GPIO.setmode(GPIO.BCM)
GPIO.setup(IR_SENSOR_PIN, GPIO.IN)
try:
    while True:
        sensor_value = GPIO.input(IR_SENSOR_PIN)
        if sensor_value == 1:
            print("object detected")
        else
            print("No object Detected")
        time.sleep(0.5)
except KeyboardInterrupt:
    print("Exiting program.....")
    GPIO.cleanup()
```

Verified
Date 22/1/24


VIT
Vellore Institute of Technology
VELLORE CAMPUS
Sreenivasan S
21BEC0256
HOSTELLER



PROCEDURE

QUESTION 1

Open Cisco Packet Tracer and then select the appropriate LANs and do the needful connections. Use the appropriate commands to run the simulation.

QUESTION 2

Use a bread board and R-Pi board and make the connections with Ultrasonic sensor. Now connect the buzzer or LED to R-Pi sensor. Connect the R-Pi to the computer and connect the display and mouse to R-Pi. Now write the code in Thonny and execute the program to get the results.

QUESTION 3

Use a Breadboard and R-Pi board and make the connections with IR Sensor and the LED. Connect the R-Pi to the computer. Connect the display and mouse to R-Pi and write the code in Thonny and execute the program to get the results.

RESULTS

1) The smart home appliance experiment using switches and hub has been successfully implemented using cisco packet tracer. A switch operates at the data link layer of the OSI model and forwards data packets to specific devices based on their MAC addresses, enabling simultaneous communication between' multiple devices on a network without broadcasting to all ports. In contrast, a hub operates at the physical layer and simply broadcasts data to all connected devices, resulting in less efficient use of network bandwidth and increased chances of collisions.
2) The experiment has been successfully executed using ultrasonic sensor and results were obtained.
3) The experiment has been successfully executed using infrared sensor and results were obtained as object was detected.