Experiment-III: Python Programs for Raspberry Pi

1. Run a python program on pi to calculate the area of rectangle, triangle, and circle.

```
# Run a python program on pi to calculate the area of rectangle, triangle, and circle.

shape = input("Enter the shape (rectangle, triangle, circle): ").lower()


if shape == 'rectangle':

    length = float(input("Enter the length: "))

    width = float(input("Enter the width: "))

    area = length * width

    print("Area of rectangle:", area)


elif shape == 'triangle':

    base = float(input("Enter the base: "))

    height = float(input("Enter the height: "))

    area = 0.5 * base * height

    print("Area of triangle:", area)


elif shape == 'circle':

    radius = float(input("Enter the radius: "))

    area = 3.14159 * radius * radius

    print("Area of circle:", area)


else:

    print("Invalid shape")
```

2. Run a python program on pi to demonstrate while loop.

```
# Run a python program on pi to demonstrate while loop
i = 1
while i <= 5:
    print("This is iteration", i)
    i += 1
```

3. Run a python program on pi to demonstrate for loop.

```
# Run a python program on pi to demonstrate for loop
for i in range(1, 6):
    print("This is iteration", i)
```

4. Run a python program on pi to demonstrate handle Divide by Zero Exception.

```
# Run a python program on pi to demonstrate handle Divide by Zero Exception
try:
    num = int(input("Enter a number: "))
    result = 10 / num
    print("Result:", result)
except ZeroDivisionError:
    print("Error: Division by zero is not allowed")
```

5. Run a python program on pi to demonstrate file operations.

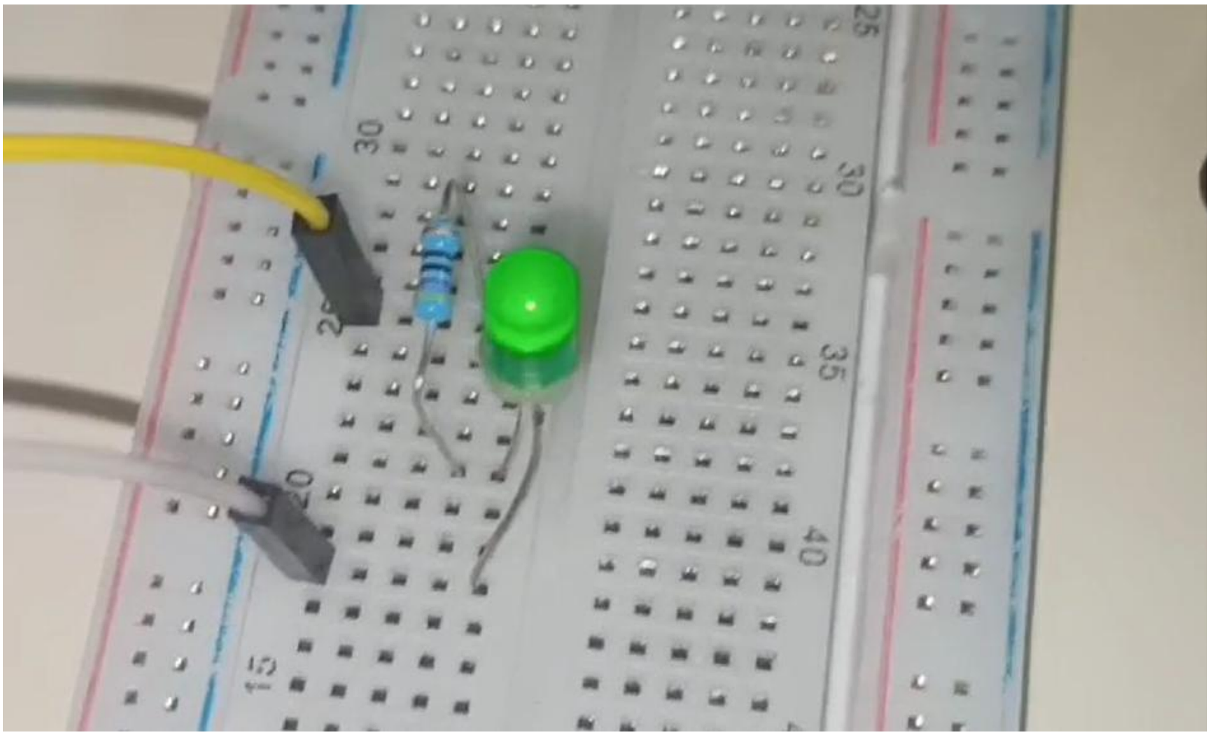
```
# Run a python program on pi to demonstrate file operations
file_name = "sample.txt"

# Write to file
```

```
with open(file_name, 'w') as file:
    file.write("Hello, this is a sample file written using Python on Raspberry Pi\n")

# Read from file
with open(file_name, 'r') as file:
    content = file.read()
    print("File content:", content)
```

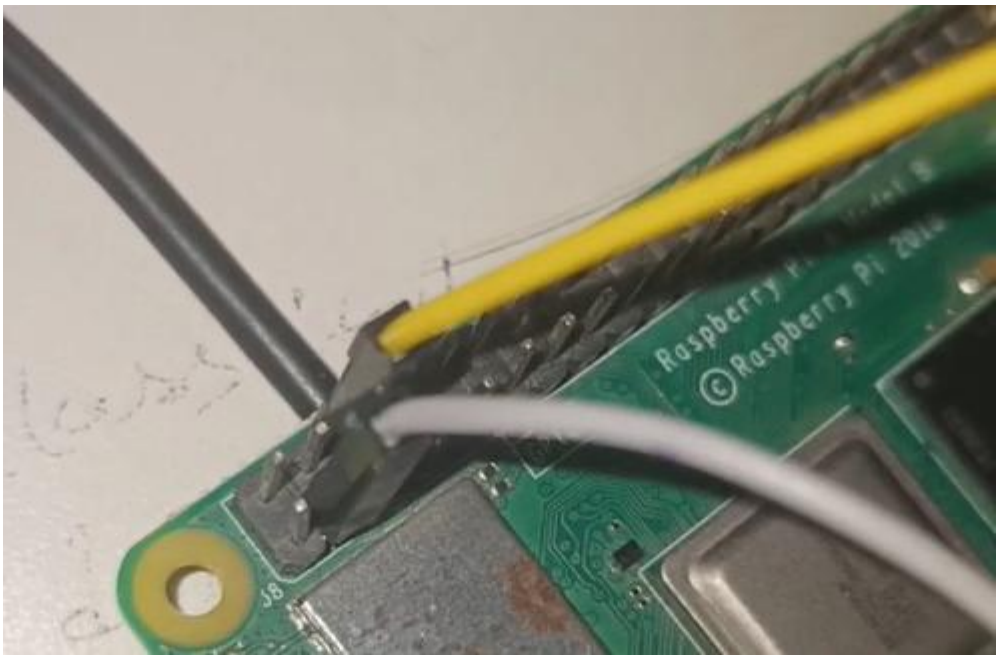
EXPERIMENT-4

1. Demonstrate Light an LED through Python program.

```
import RPi.GPIO as GPIO
import time
GPIO.setmode(GPIO.BOARD)
GPIO.setup(3,GPIO.OUT)
while True:
   GPIO.output(3,GPIO.HIGH)
   time.sleep(1)
   GPIO.output(3,GPIO.LOW)
   time.sleep(1)
```

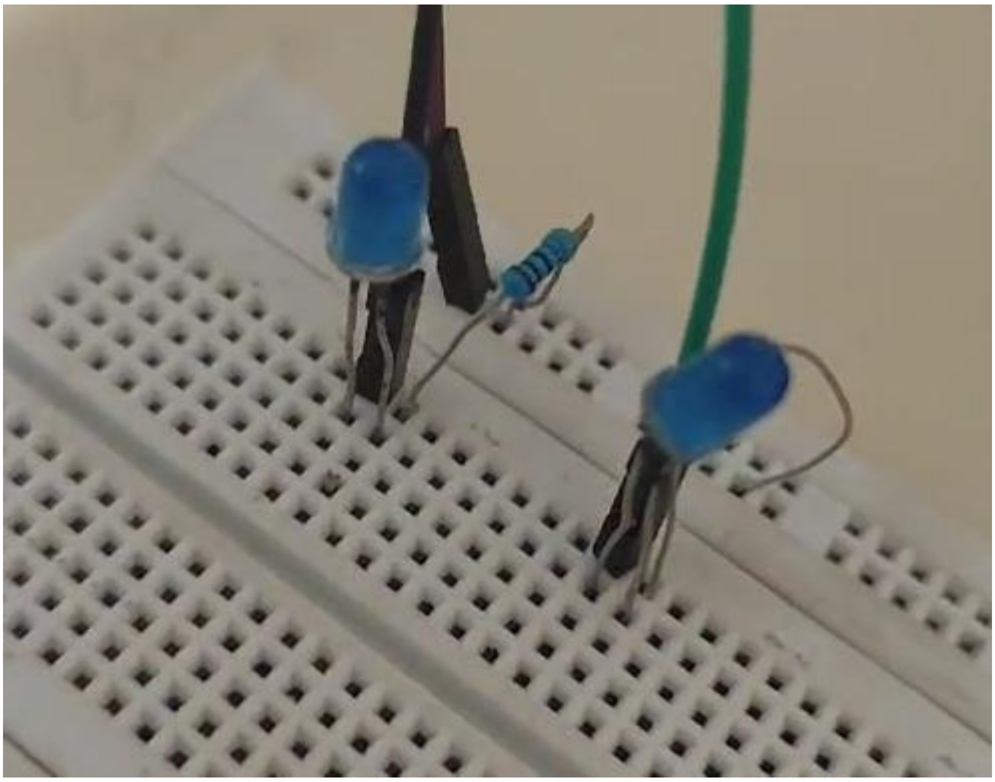


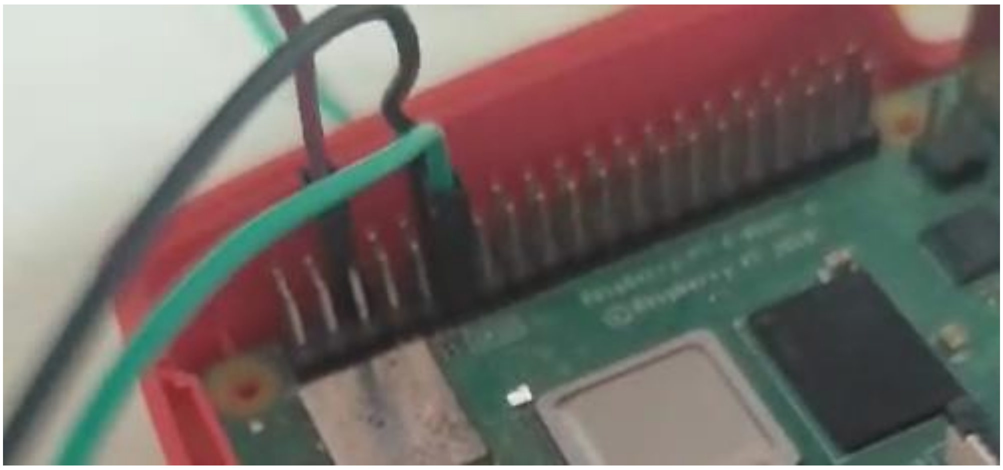
Pins – 3 , 6

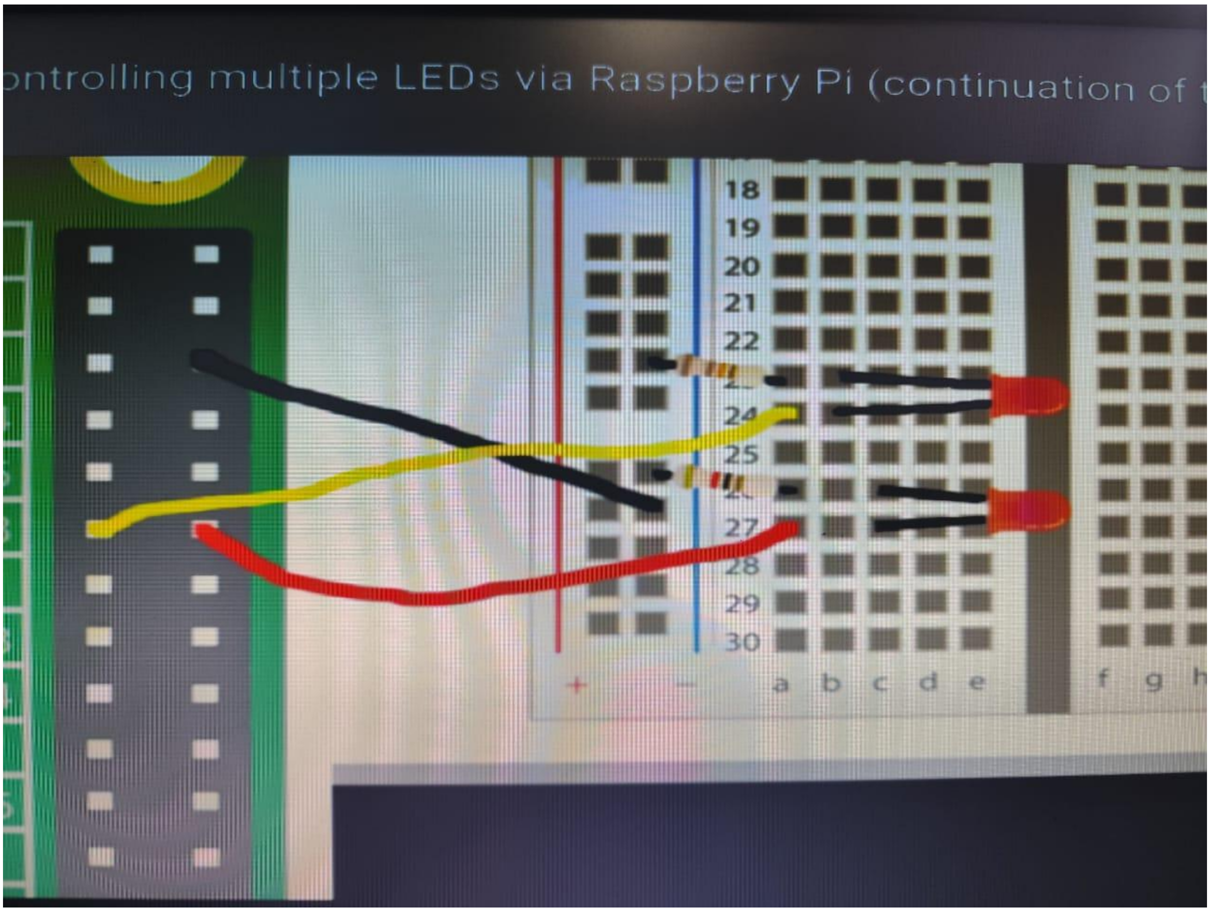


2. Write a program to demonstrate light an LED which are connected in series.

https://youtu.be/byRWKuFn9TY?si=FCEuXr0-ckWGhb5P







ontrolling multiple LEDs via Raspberry Pi (continuation of t

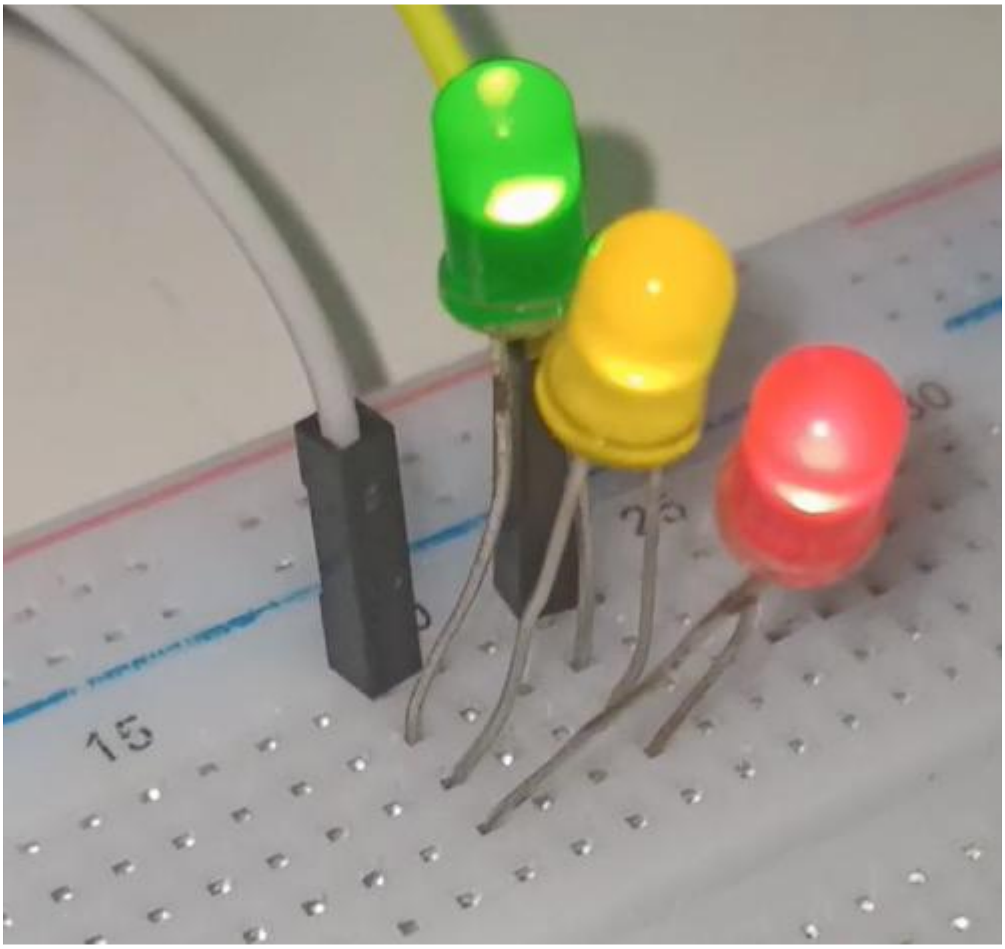
Pins – 6 , 11 , 12

```
from gpiozero import LED

import time
```

```
# Get the sleep duration from the user
K = int(input("sleep for: "))

# Define two LEDs connected to GPIO 18 and GPIO 17
l1 = LED(18)
l2 = LED(17)

# Infinite loop to control the LEDs
while True:
    l1.on()
    time.sleep(K)
    l1.off()
    time.sleep(K)

    l2.on()
    time.sleep(K)
    l2.off()
    time.sleep(K)

    l1.on()
    l2.on()
    time.sleep(K)

    l1.off()
    l2.off()
    time.sleep(K)
```

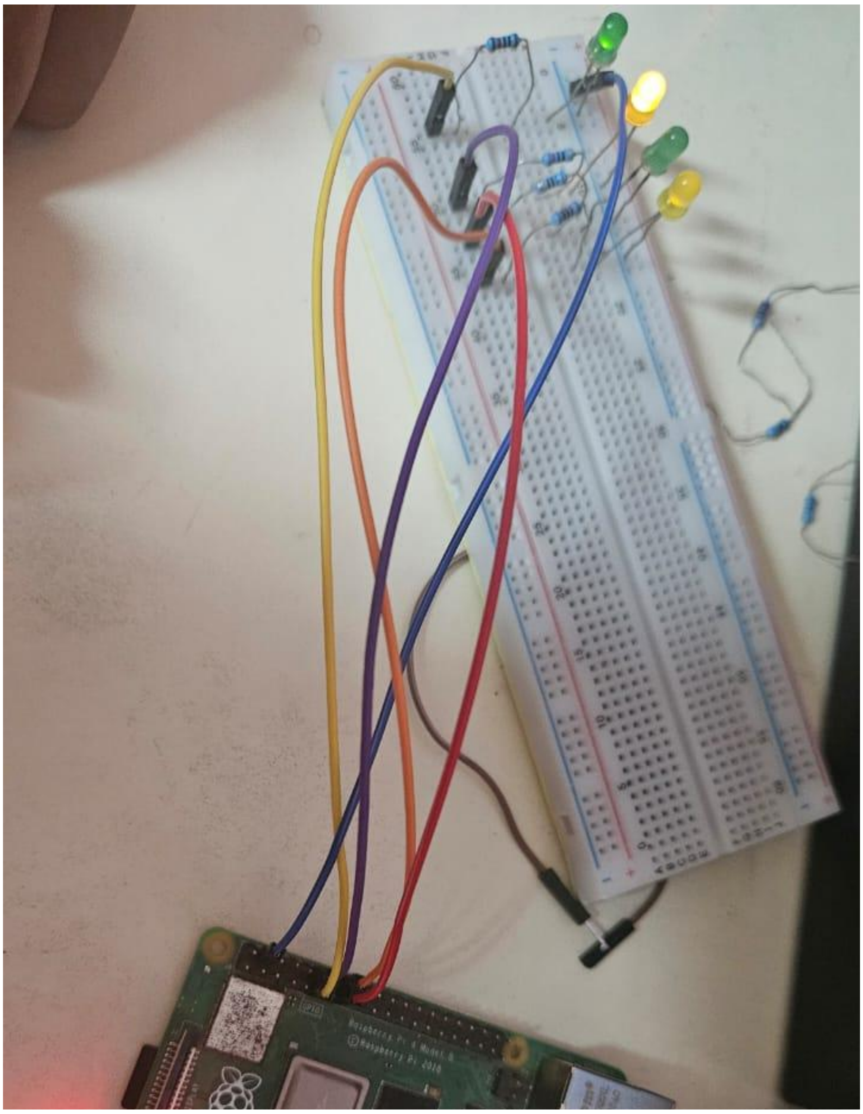
3. Write a program to demonstrate light an LED which are connected in parallel.

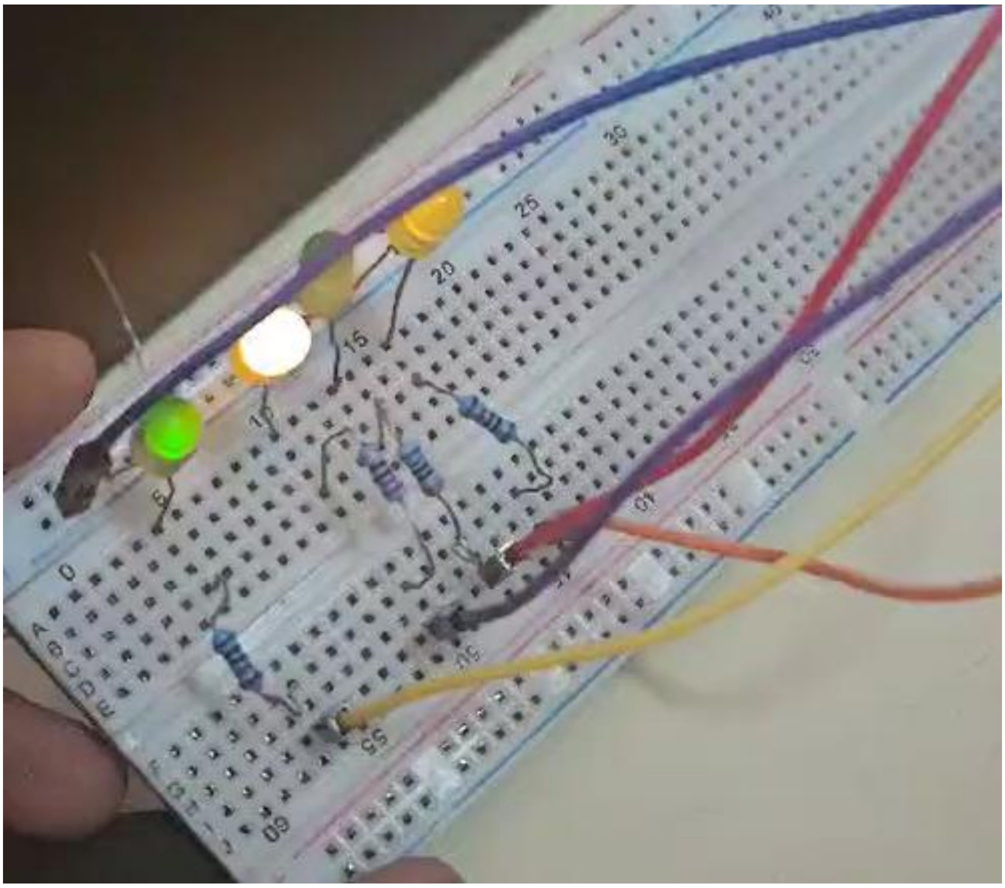
```
import RPi.GPIO as GPIO
import time
GPIO.setmode(GPIO.BOARD)
GPIO.setup(3,GPIO.OUT)
while True:
   GPIO.output(3,GPIO.HIGH)
   time.sleep(1)
   GPIO.output(3,GPIO.LOW)
   time.sleep(1)
```

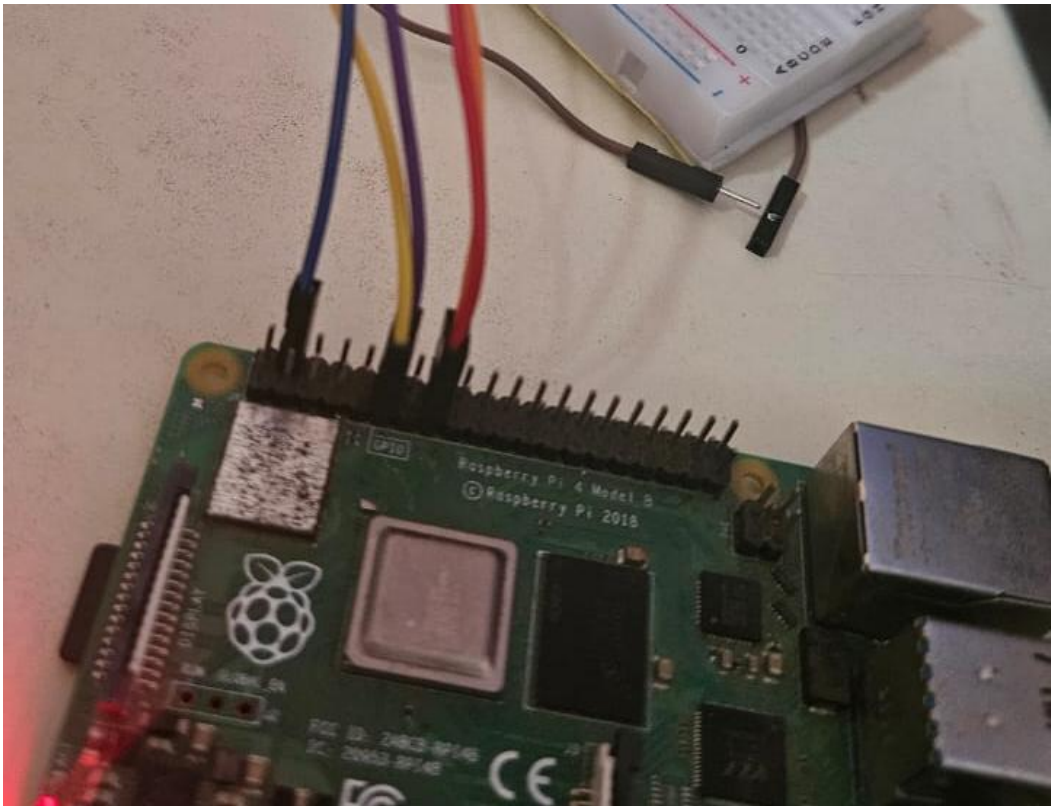


Pins – 3 , 6

4. Design a program to infinitely blink a sequence of 4 LEDs connected to Pi, one after the other with the delay of 500 ms.

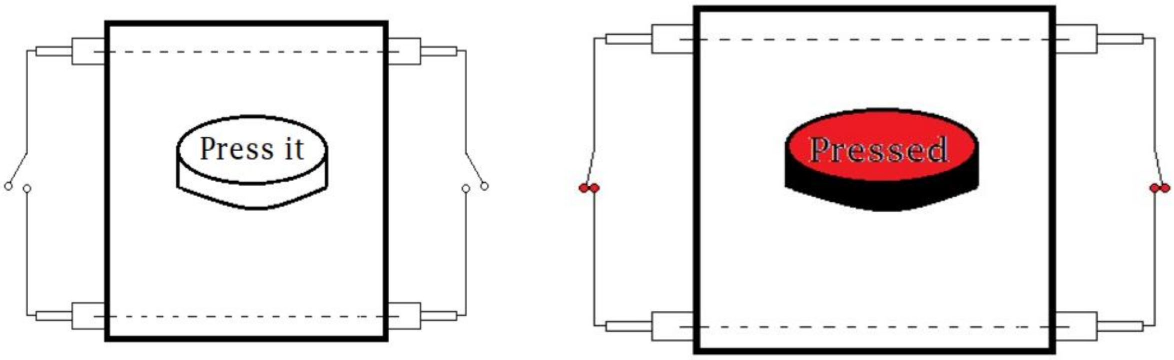



5
10
15
20
25
30
35
40
45
50
55
60


GPIO
Raspberry Pi 4 Model B
©Raspberry Pi 2018
DISPLAY
CE

```
import RPi.GPIO as GPIO
import time
while True:
    GPIO.setmode(GPIO.BCM)
    GPIO.setup(17,GPIO.OUT)
    GPIO.setup(18,GPIO.OUT)
    GPIO.setup(22,GPIO.OUT)
    GPIO.setup(23,GPIO.OUT)
    GPIO.output(17,True)
    time.sleep(0.5)
    GPIO.output(17,False)
    time.sleep(0.5)
    GPIO.output(17,True)
    time.sleep(0.5)
    GPIO.output(17,False)
    time.sleep(0.5)
    GPIO.output(22,True)
    time.sleep(0.5)
    GPIO.output(22,False)
    time.sleep(0.5)
    GPIO.output(23,True)
    time.sleep(0.5)
    GPIO.output(23,False)
```

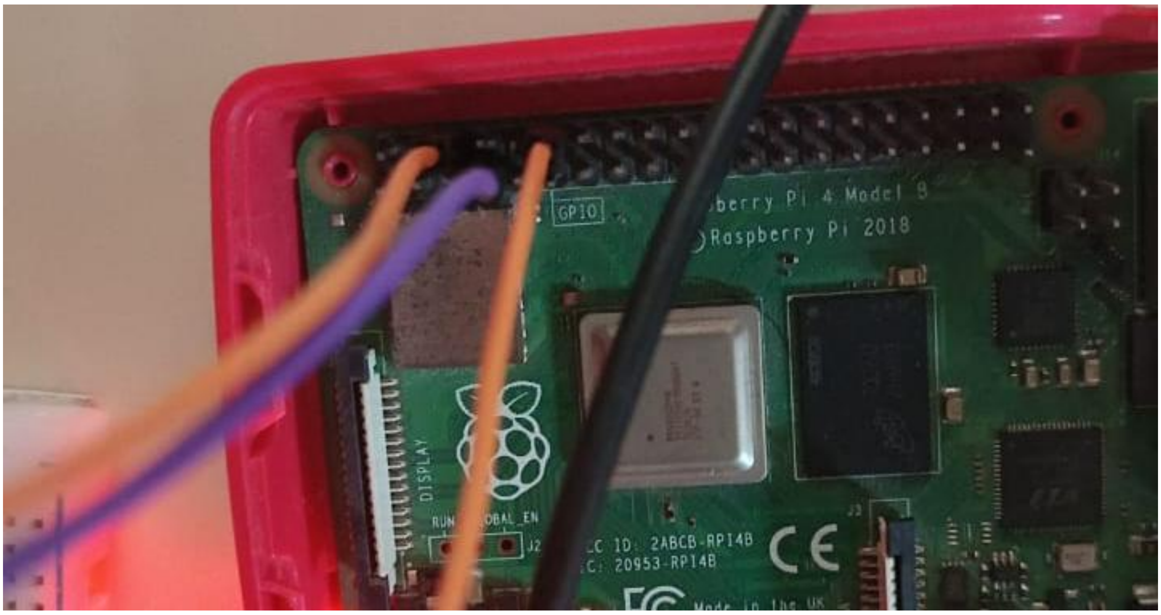
EXPERIMENT-5

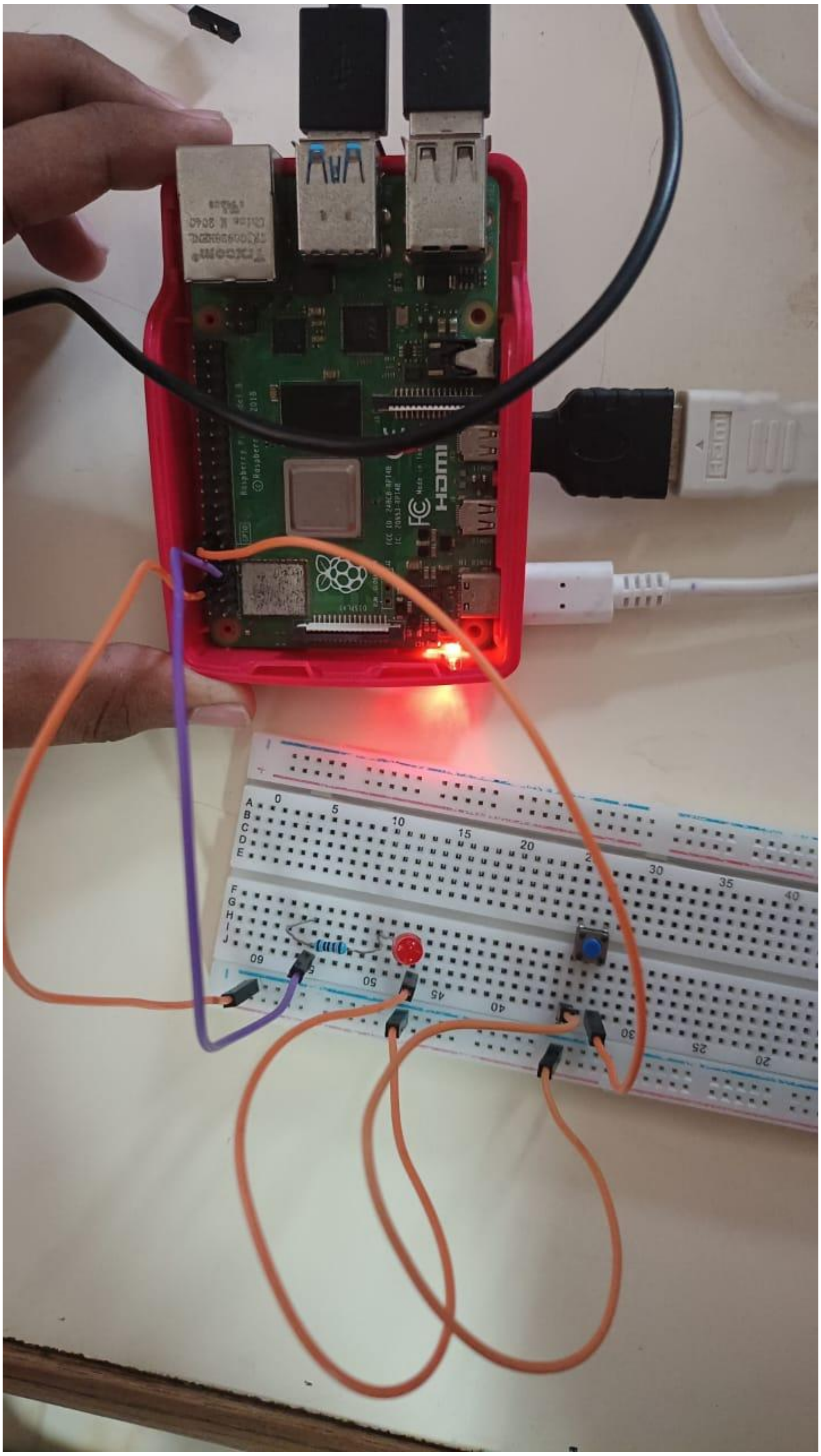


1. Write a program to demonstrate LED with button.

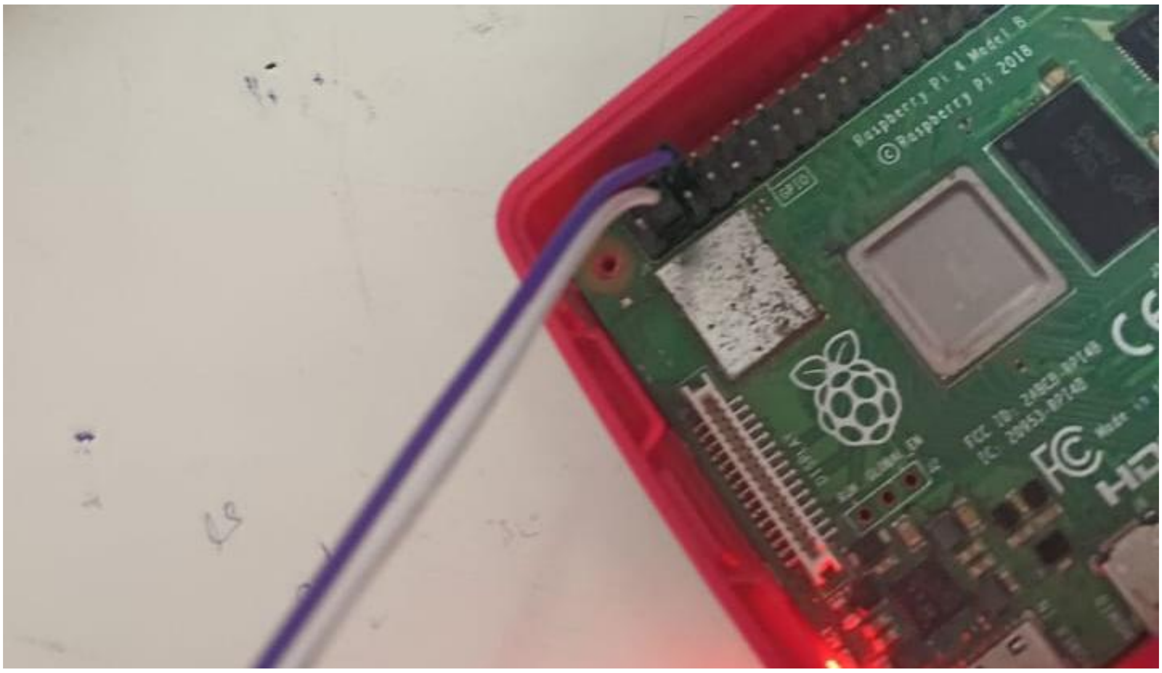
   Pins-6,7,12

```
import RPi.GPIO as GPIO
from time import sleep

GPIO.setmode(GPIO.BCM)

sleepTime = 0.1

lightPin = 4
buttonPin = 17

GPIO.setup(lightPin, GPIO.OUT)
GPIO.setup(buttonPin, GPIO.IN, pull_up_down=GPIO.PUD_UP)

while True:
   GPIO.output(lightPin, GPIO.input(buttonPin))
   sleep(0.1)
```

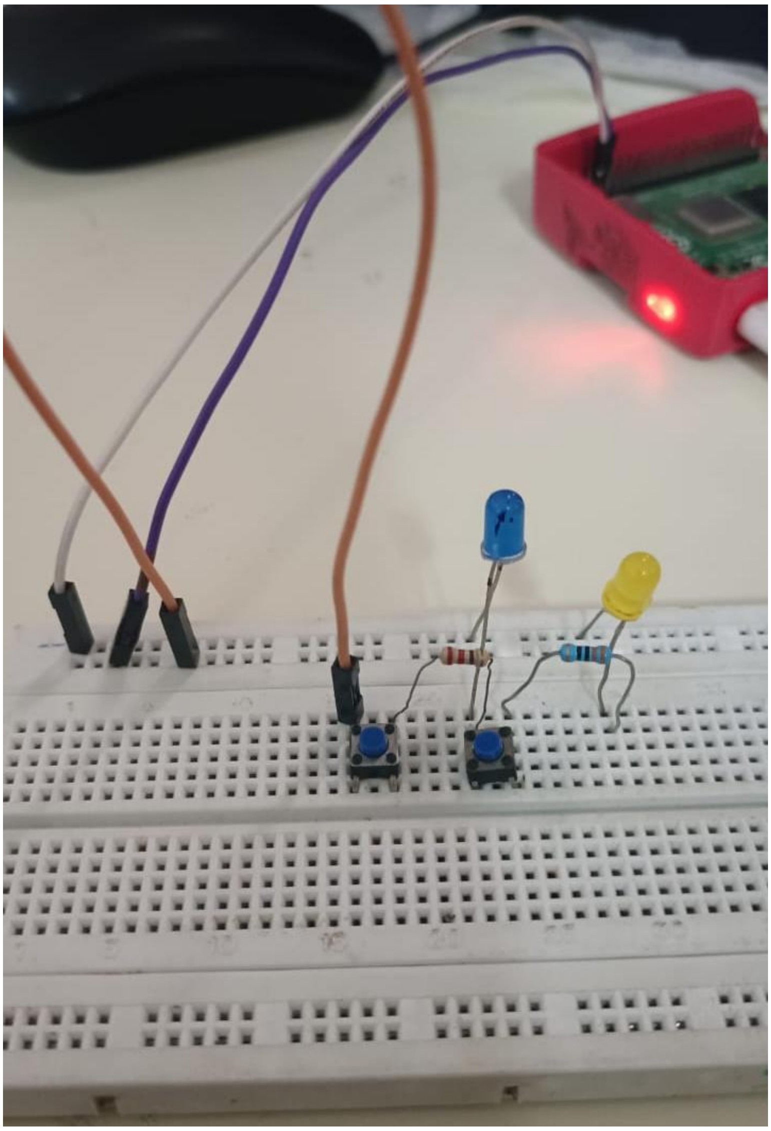




2. Write a program to demonstrate two LEDs with two buttons.



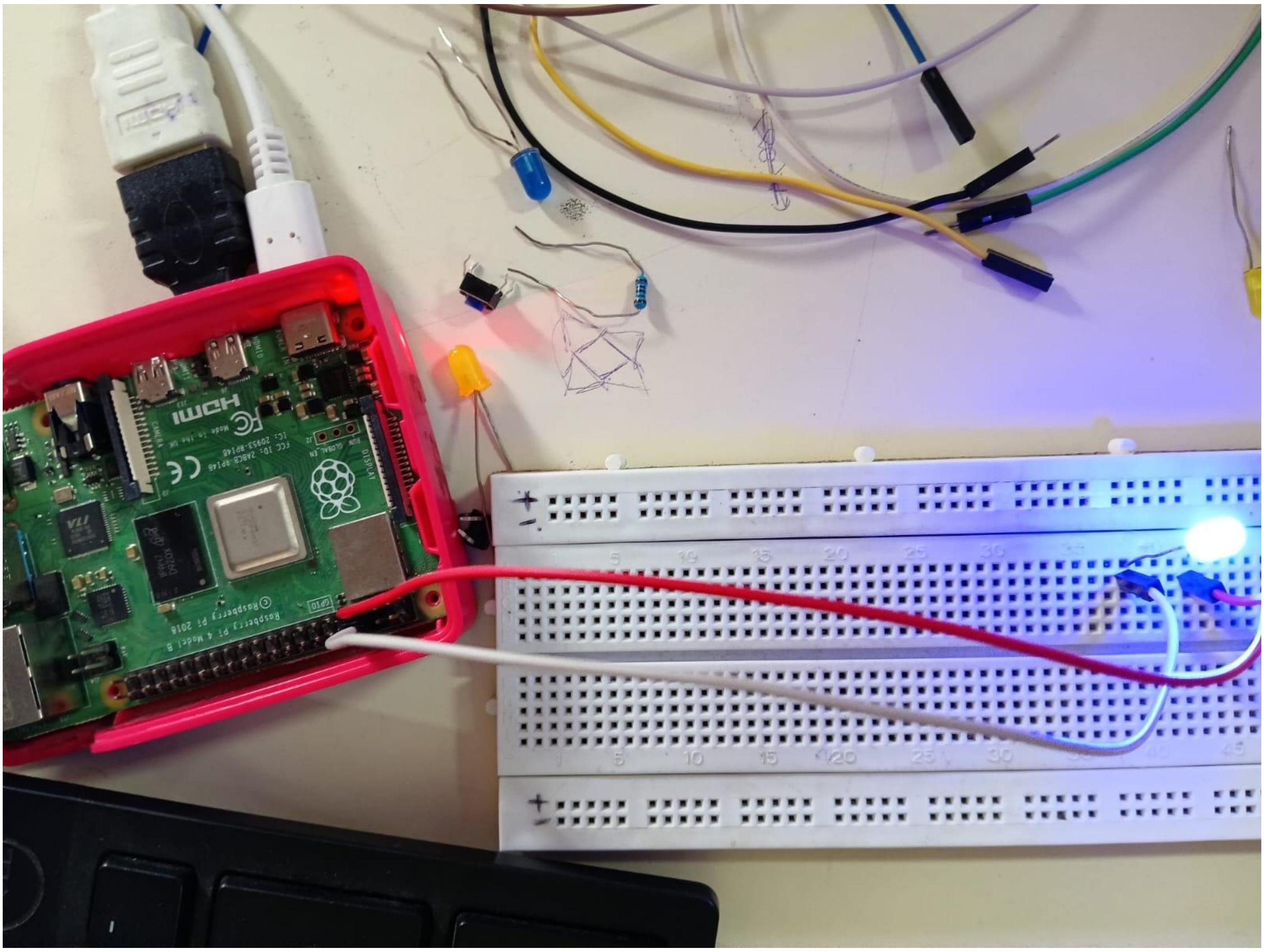
Pins- 3 , 6



```
from time import sleep
import RPi.GPIO as GPIO
GPIO.setmode(GPIO.BOARD)
button1=16
button2=12
LED1=22
LED2=18
GPIO.setup(button1,GPIO.IN,pull_up_down=GPIO.PUD_UP)
GPIO.setup(button2,GPIO.IN,pull_up_down=GPIO.PUD_UP)
GPIO.setup(LED1,GPIO.OUT,)
GPIO.setup(LED2,GPIO.OUT)
BS1=False
BS2=False
while(1):
    if GPIO.input(button1)==0:
        print "Button 1 Was Pressed:"
        if BS1==False:
            GPIO.output(LED1,True)
            BS1=True
            sleep(.5)
        else:
            GPIO.output(LED1,False)
            BS1=False
            sleep(.5)
    if GPIO.input(button2)==0:
        print "Button 2 Was Pressed:"
        if BS2==False:
            GPIO.output(LED2,True)
            BS2=True
            sleep(.5)
        else:
            GPIO.output(LED2,False)
            BS2=False
            sleep(.5)
```

3. Write a program to demonstrate light an LED through web.

Pins – 3 (+ terminal of LED) , 6 (- terminal of LED)



Create a file named app.py and add the following code:

```
from flask import Flask, render_template

import RPi.GPIO as GPIO

app = Flask(_name_)


# GPIO setup

GPIO.setmode(GPIO.BCM)

led_pin = 3

GPIO.setup(led_pin, GPIO.OUT)


@app.route('/')

def index():

    return render_template('index.html')
```

```
@app.route('/led/on')
def led_on():
    GPIO.output(led_pin, GPIO.HIGH)
    return "LED is ON"

@app.route('/led/off')
def led_off():
    GPIO.output(led_pin, GPIO.LOW)
    return "LED is OFF"

if _name_ == '_main_':
    app.run(debug=True, port=3030)
```

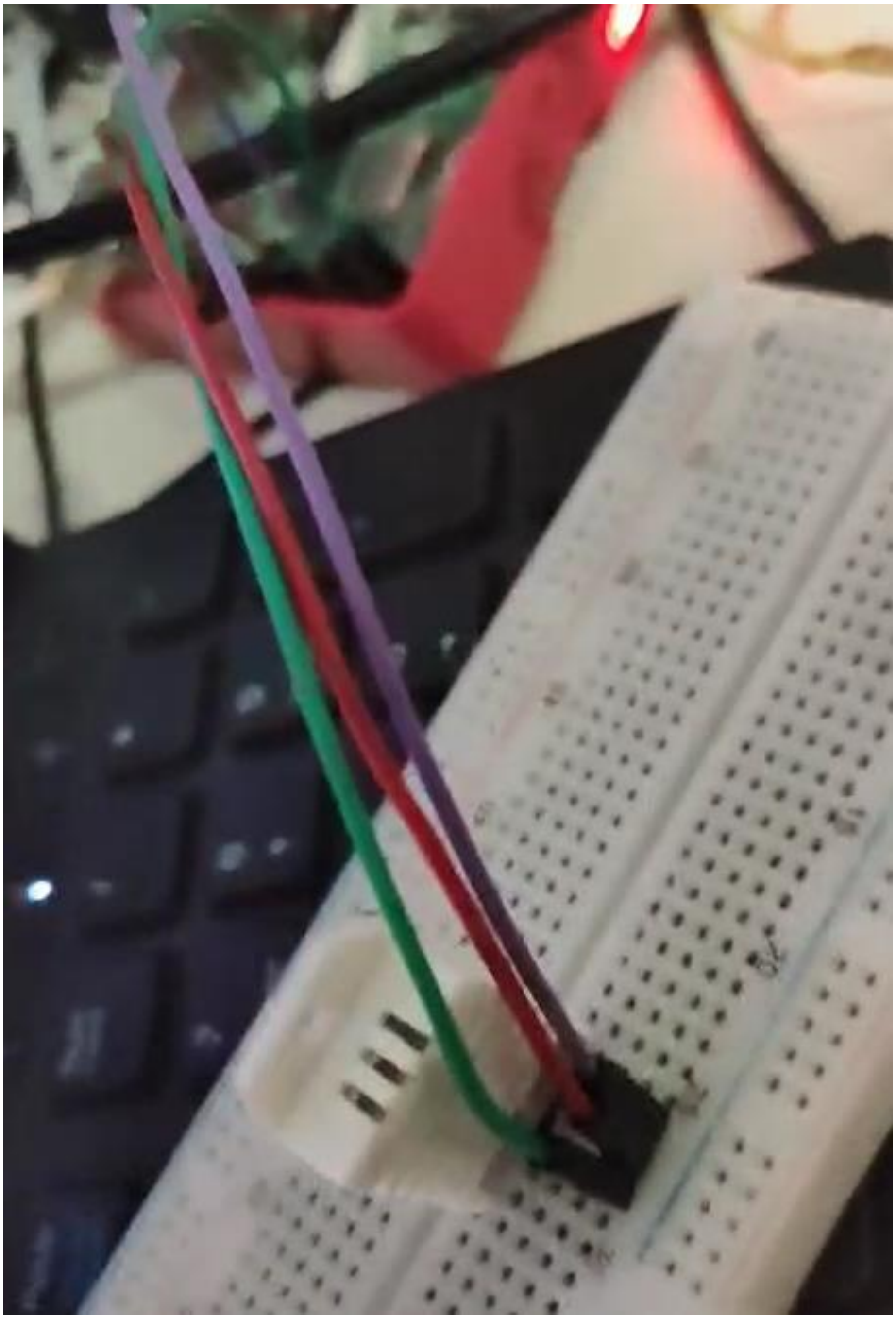
Create a folder named templates in the same directory as app.py.

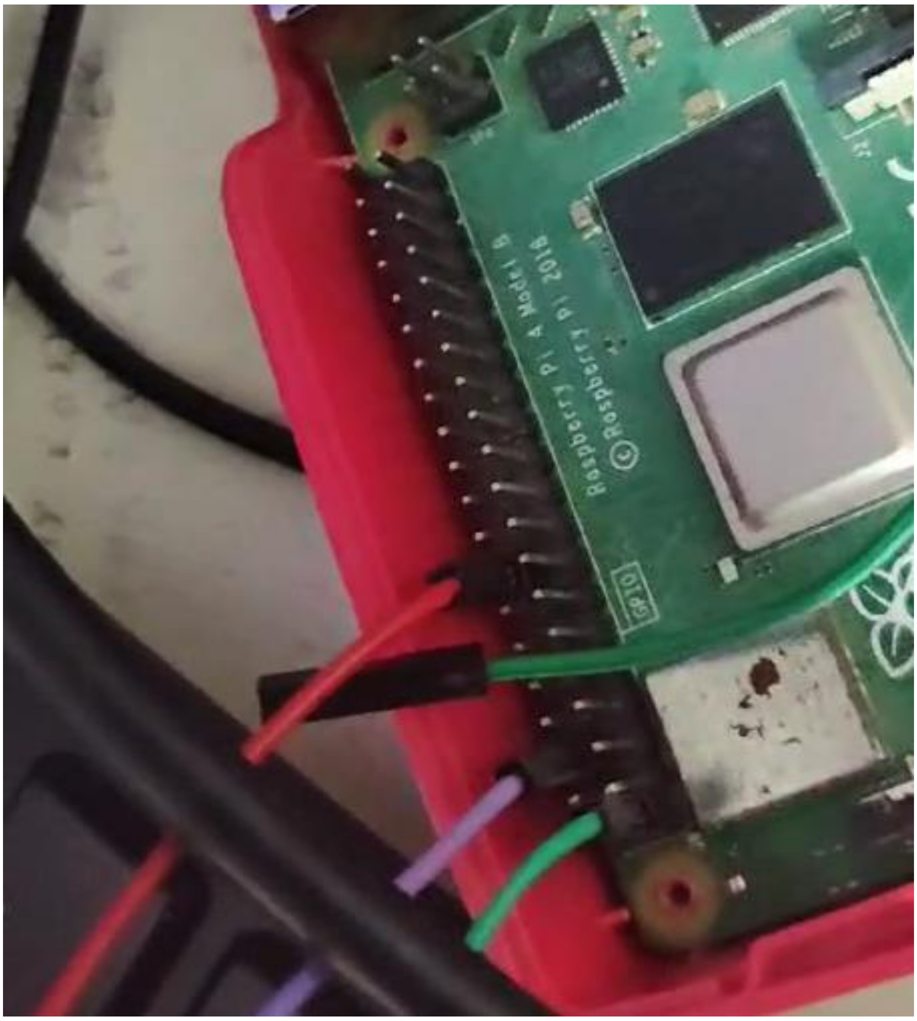
Inside templates, create a file named index.html with the following content:

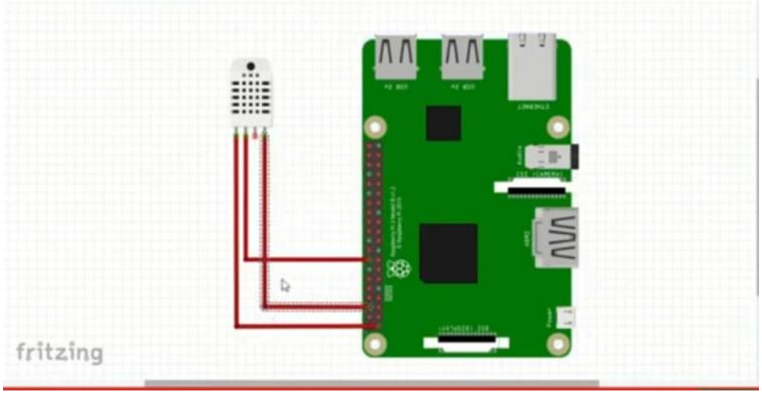
```
<!DOCTYPE html>
<html>
<head>
    <title>Raspberry Pi LED Control</title>
</head>
<body>
    <h1>Raspberry Pi LED Control</h1>
    <p><a href="/led/on">Turn LED ON</a></p>
    <p><a href="/led/off">Turn LED OFF</a></p>
</body>
</html>
```

Experiment-VI

Get input from DHT sensor







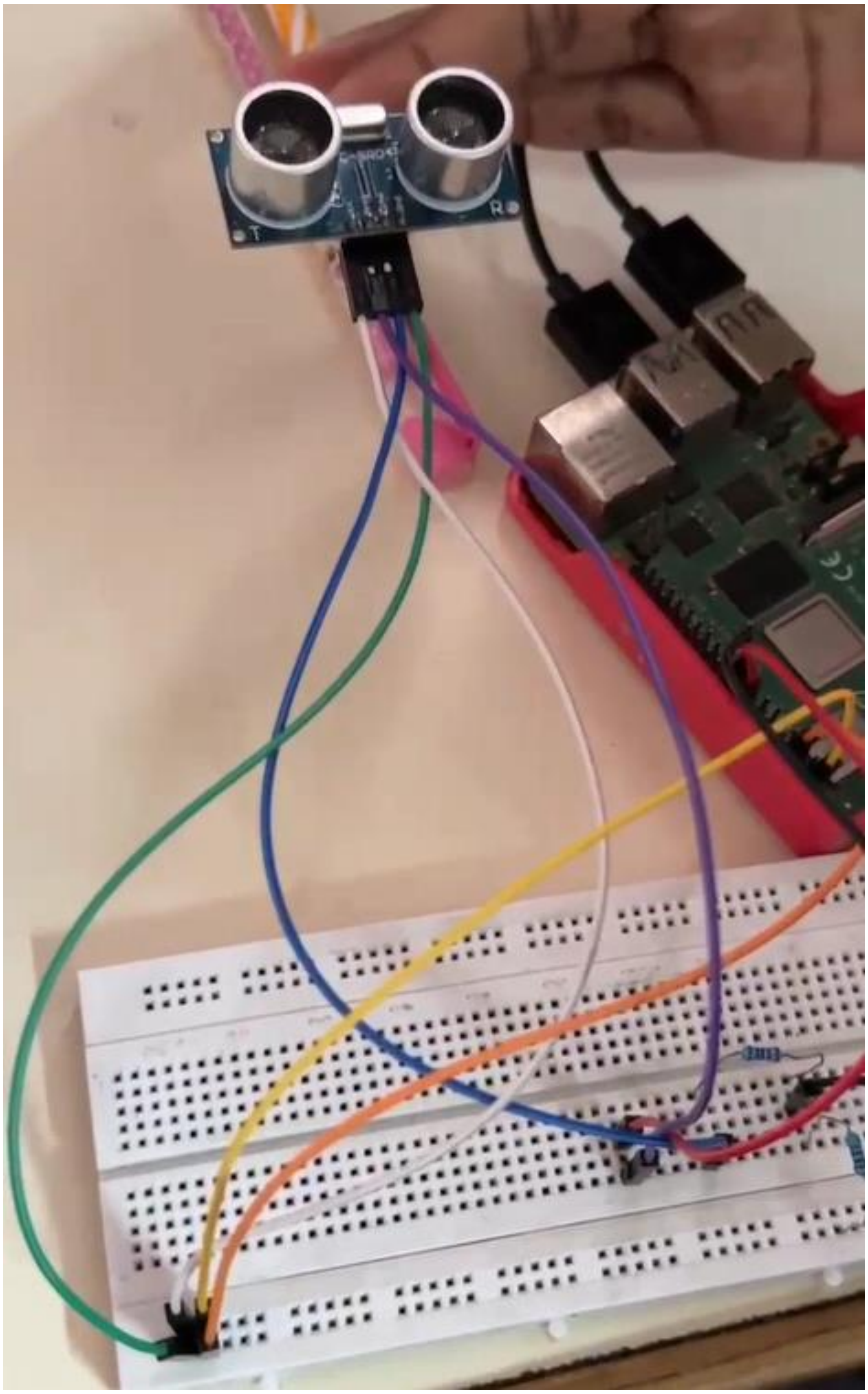
DHT22 pins	RaspberryPi pins
+	1
out	16
-	6

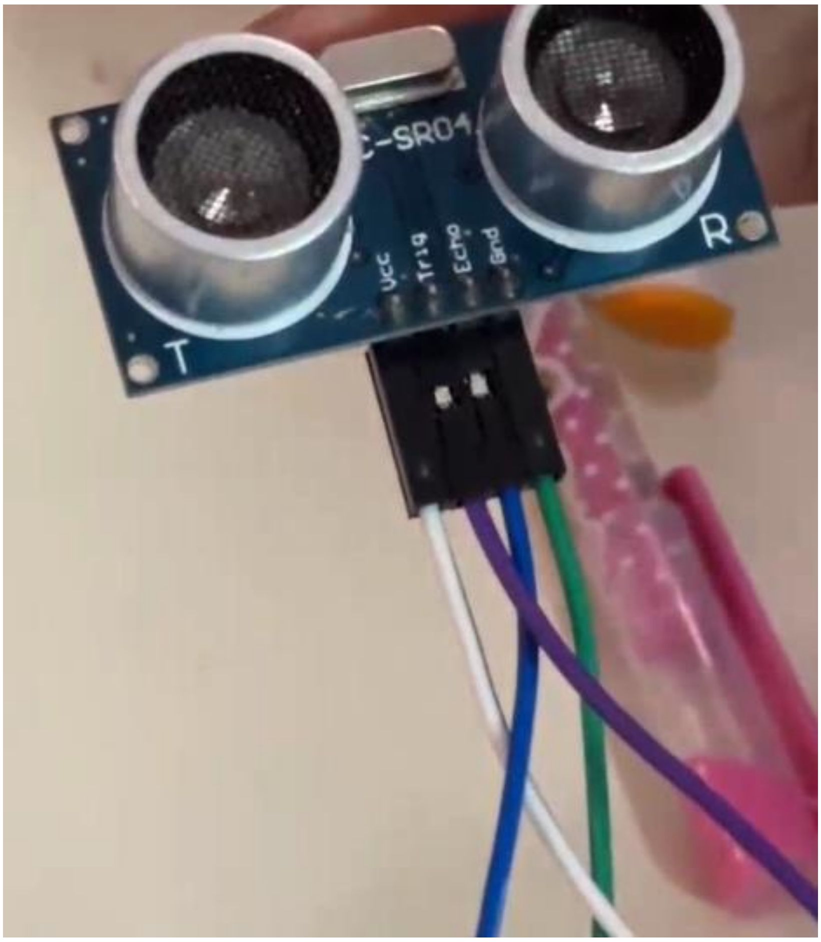
```
import Adafruit_DHT
import time
import RPi.GPIO as GPIO
GPIO.setmode(GPIO.BCM)
sensor=Adafruit_DHT.DHT22
pin=23
while True:
    humidity, temp = Adafruit_DHT.read_retry(sensor, pin)
    if humidity is None and temp is None:
        print("Failed to get reading. Try again")
    else:
        print("Temp={0:0.1f}*C
Humidity={1:0.2f}%".format(temp,humidity))
    time.sleep(1)
```

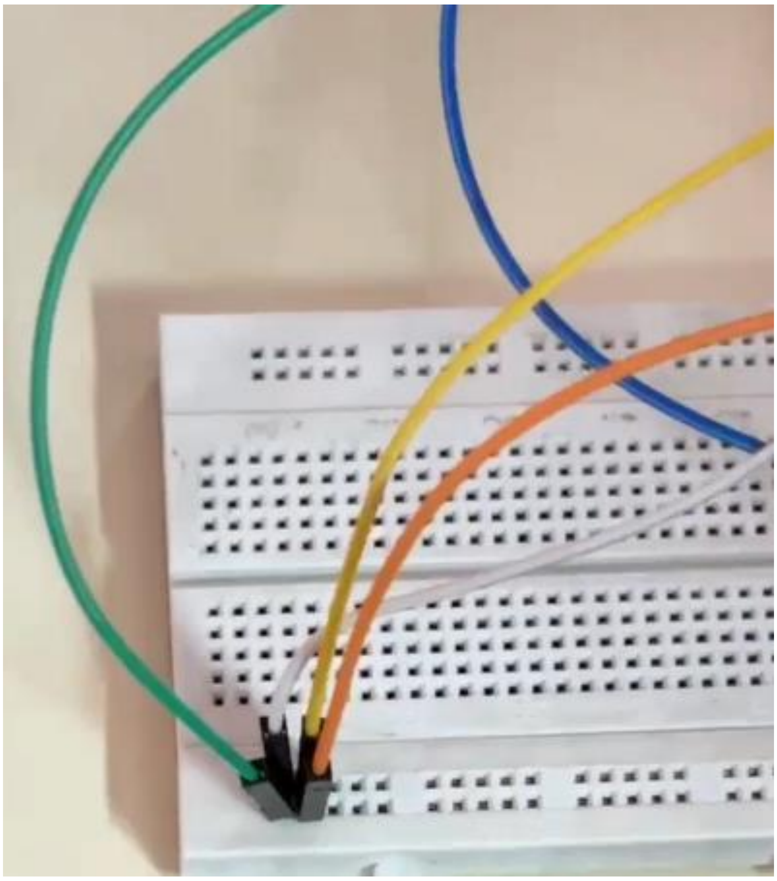
EXPERIMENT-VII

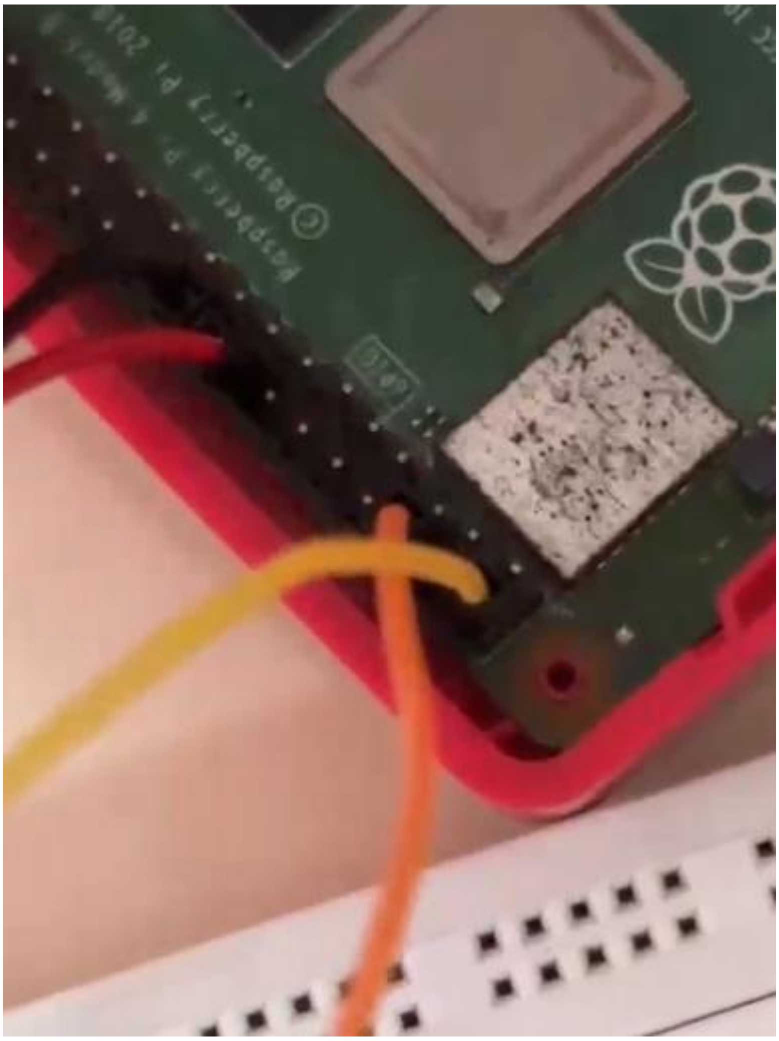
Get input from ultrasonic sensor

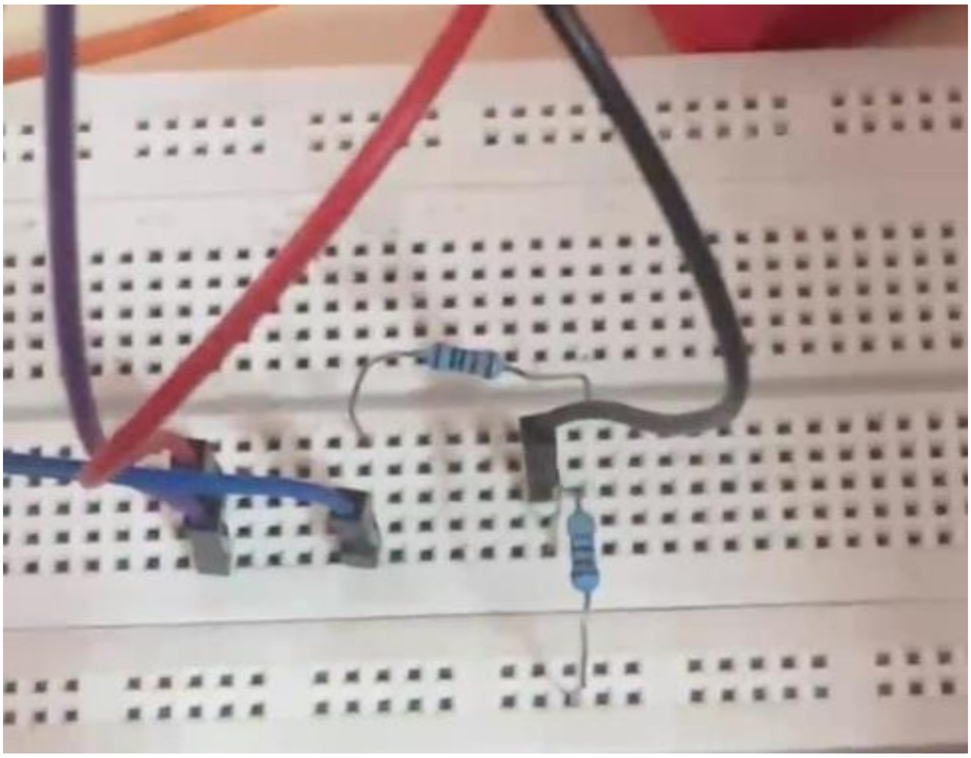
https://youtu.be/_7drIUmC8Zo?si=yn6JjJYJnzeclqUh




C-SR04
Vcc
Trig
Echo
Gnd
T
R



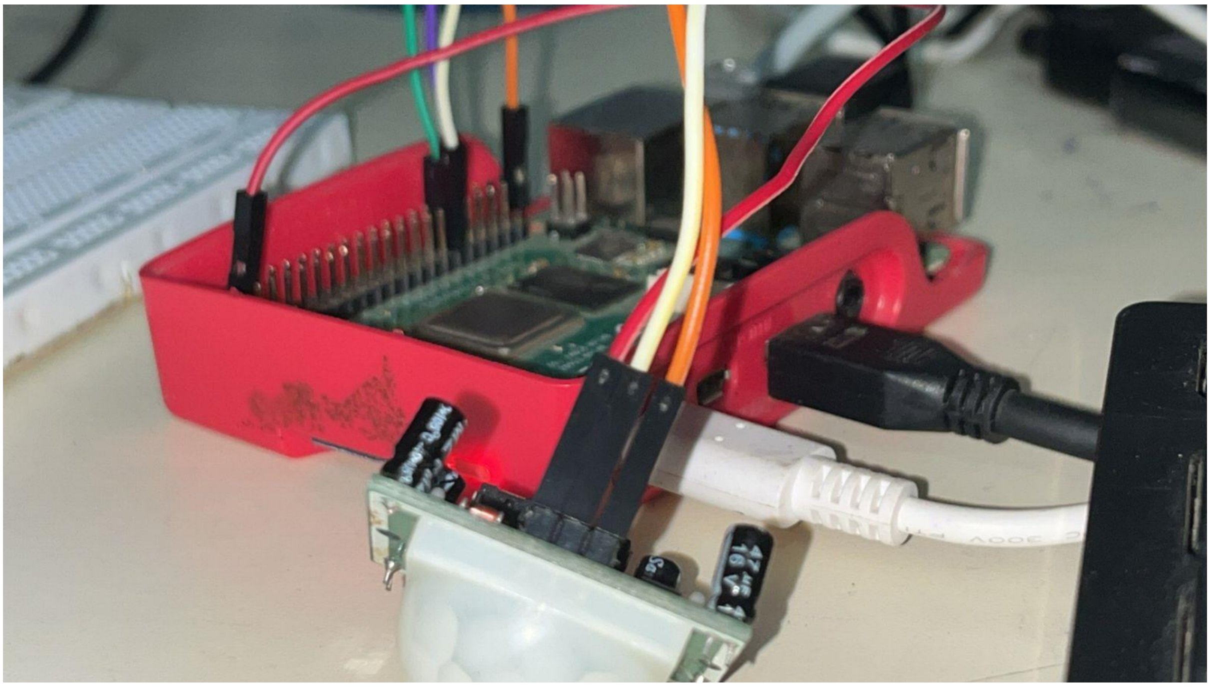

© Raspberry Pi 4 Model B
Raspberry Pi 2018
GPIO

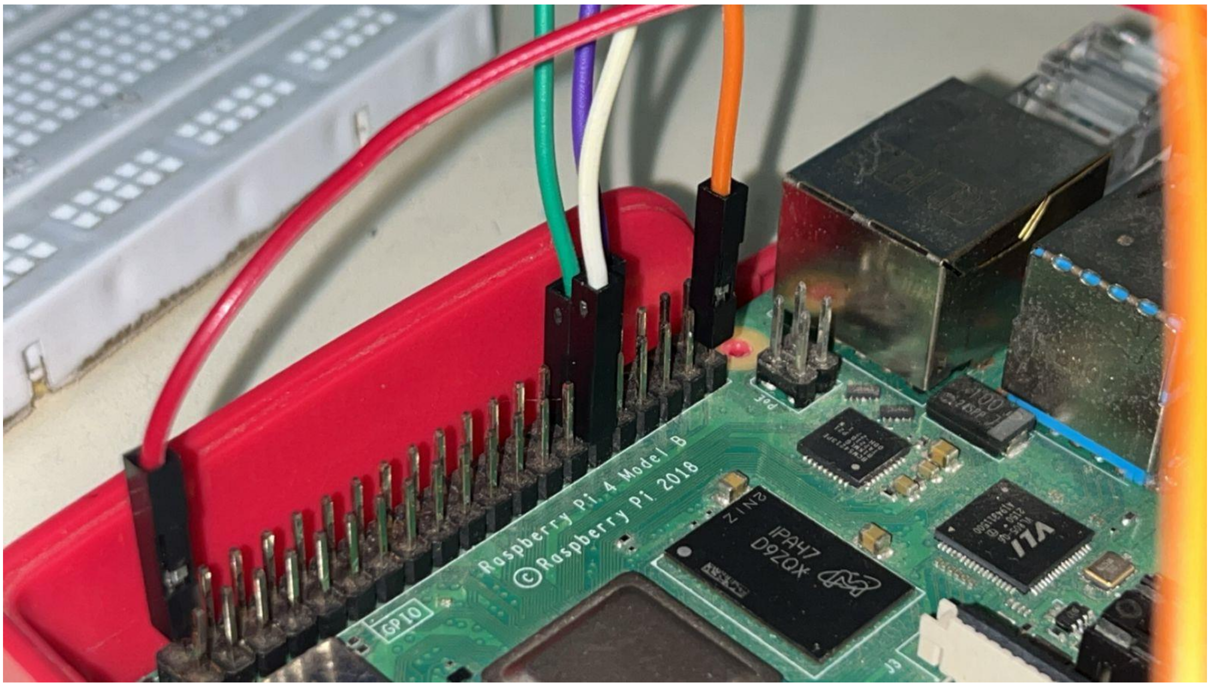


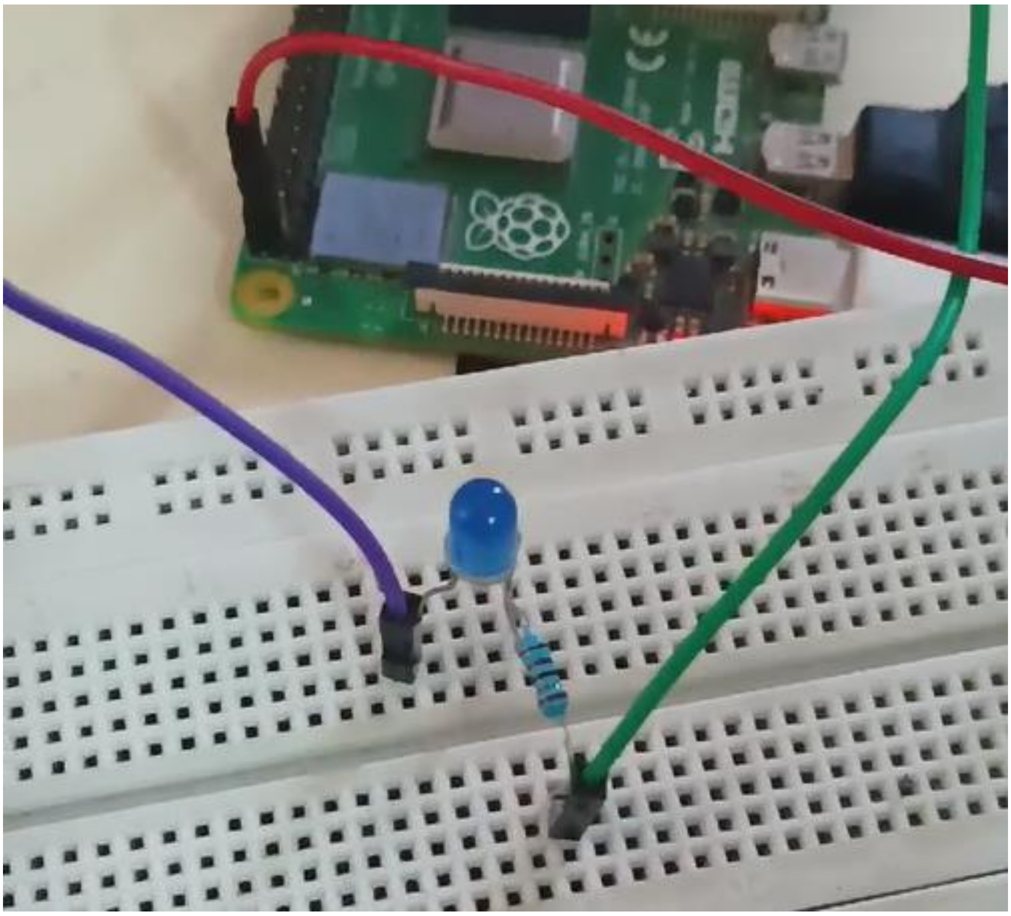
```
import RPi.GPIO as GPIO
import time
GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)
TRIG=23
ECHO=24
print("DIstance Measurment in Progress")
GPIO.setup(TRIG,GPIO.OUT)
GPIO.setup(ECHO,GPIO.IN)
GPIO.output(TRIG,False)
print("Waiting for Sensor To Settle")
while True:
    time.sleep(2)
    GPIO.output(TRIG,True)
    time.sleep(0.00001)
    GPIO.output(TRIG,False)
    while GPIO.input(ECHO)==0:
        pulse_start=time.time()
    while GPIO.input(ECHO)==1:
        pulse_end=time.time()
    pulse_duration=pulse_end-pulse_start
    distance=pulse_duration*17150
    distance=round(distance,2)
    print("Distance:",distance,"cm")
```

EXPERIMENT-VIII

Working with LED, pirsensor.







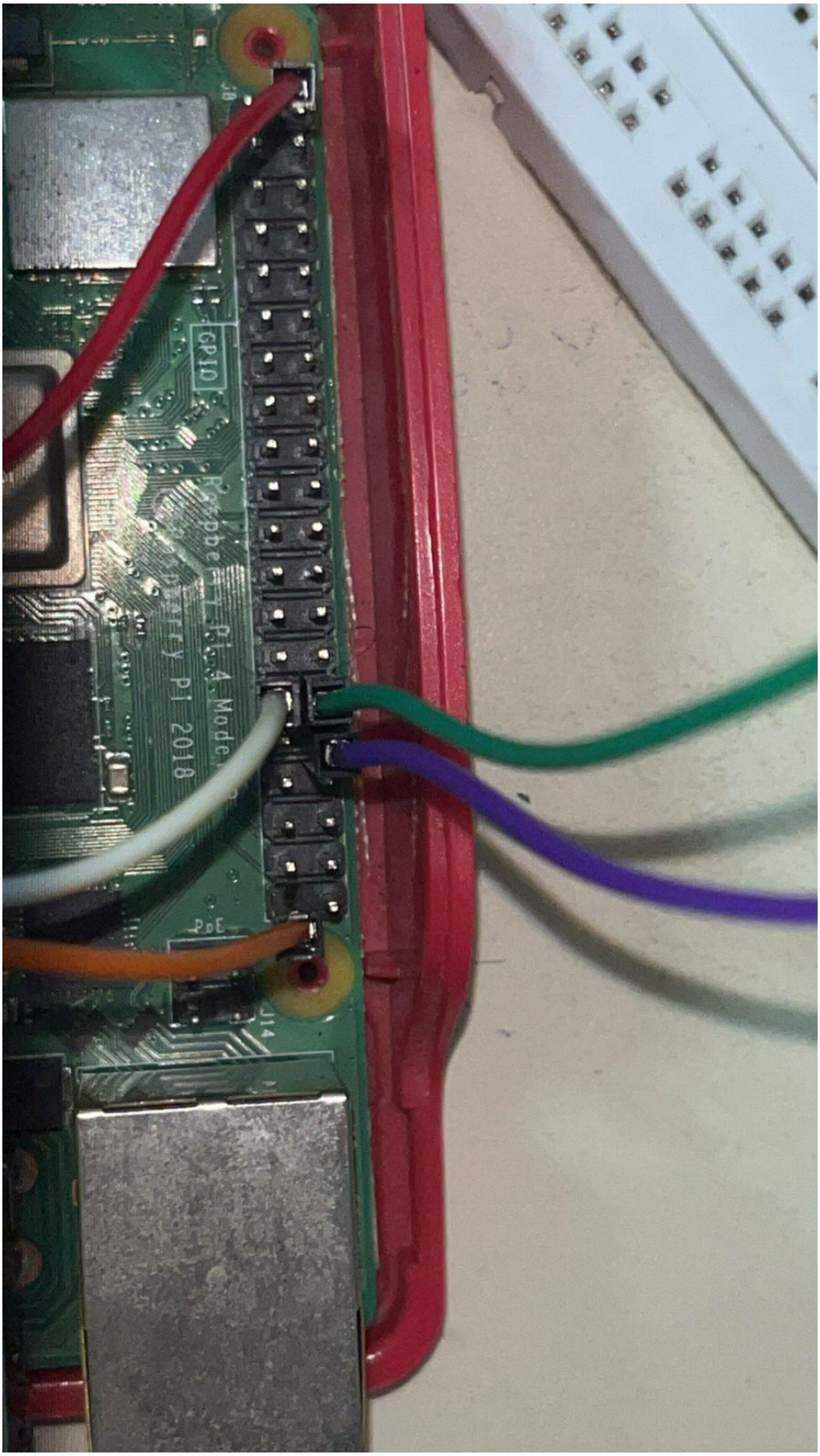
Red- 2nd pin

White - 29th pin

Green - 30th pin

Violet - 32nd pin

Orange- 39th


J8
GPIO
Raspberry Pi 4 Model B
© Raspberry Pi 2018
PoE
J14

```
import RPi.GPIO as GPIO
import time
PIR_PIN=29
LED_PIN=32
GPIO.setwarnings(False)
GPIO.setmode(GPIO.BOARD)
GPIO.setup(PIR_PIN,GPIO.IN)
GPIO.setup(LED_PIN,GPIO.OUT)
GPIO.output(LED_PIN,GPIO.LOW)
try:
    while True:
        if(GPIO.input(PIR_PIN)):
            print("Movement Detected");
            GPIO.output(LED_PIN,GPIO.HIGH)
        else:
            print("Not Detected")
            GPIO.output(LED_PIN,GPIO.LOW)
        time.sleep(1)
except KeyboardInterrupt:
    GPIO.cleanup()
```

EXPERIMENT-9

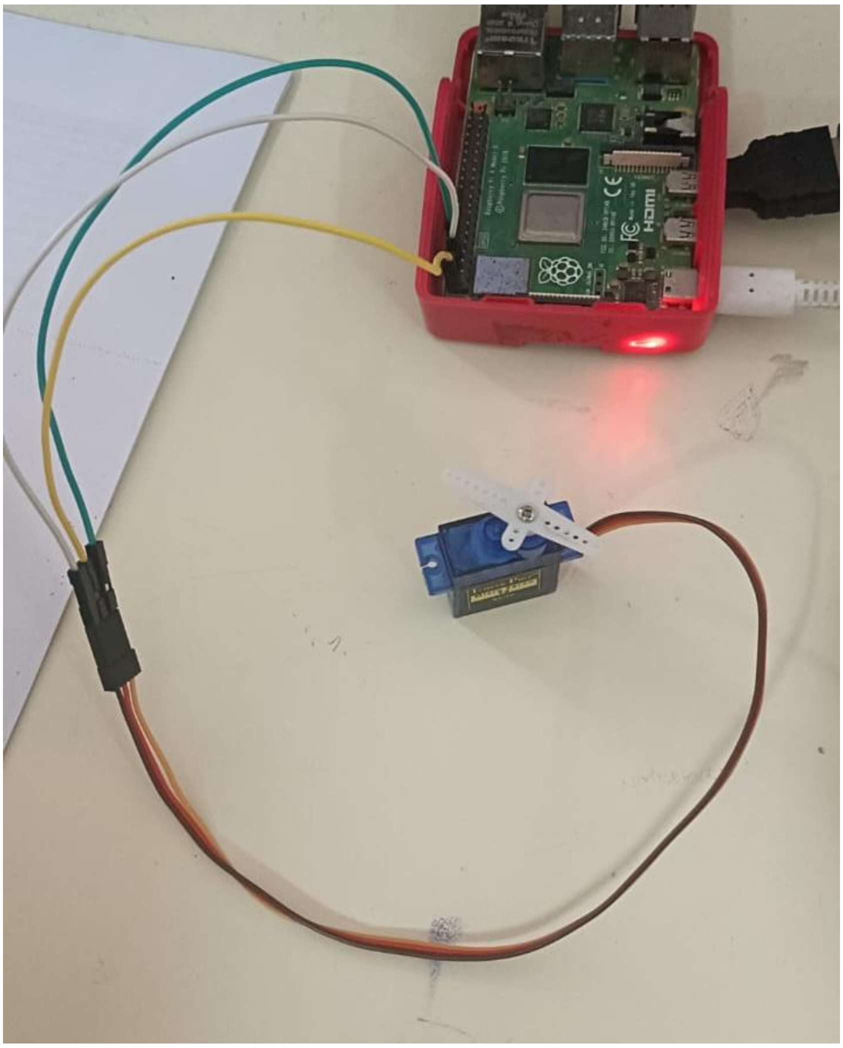
Servo motor

Pins:

Red wire of motor - 2

Brown wire of motor - 6

Orange wire of motor - 8

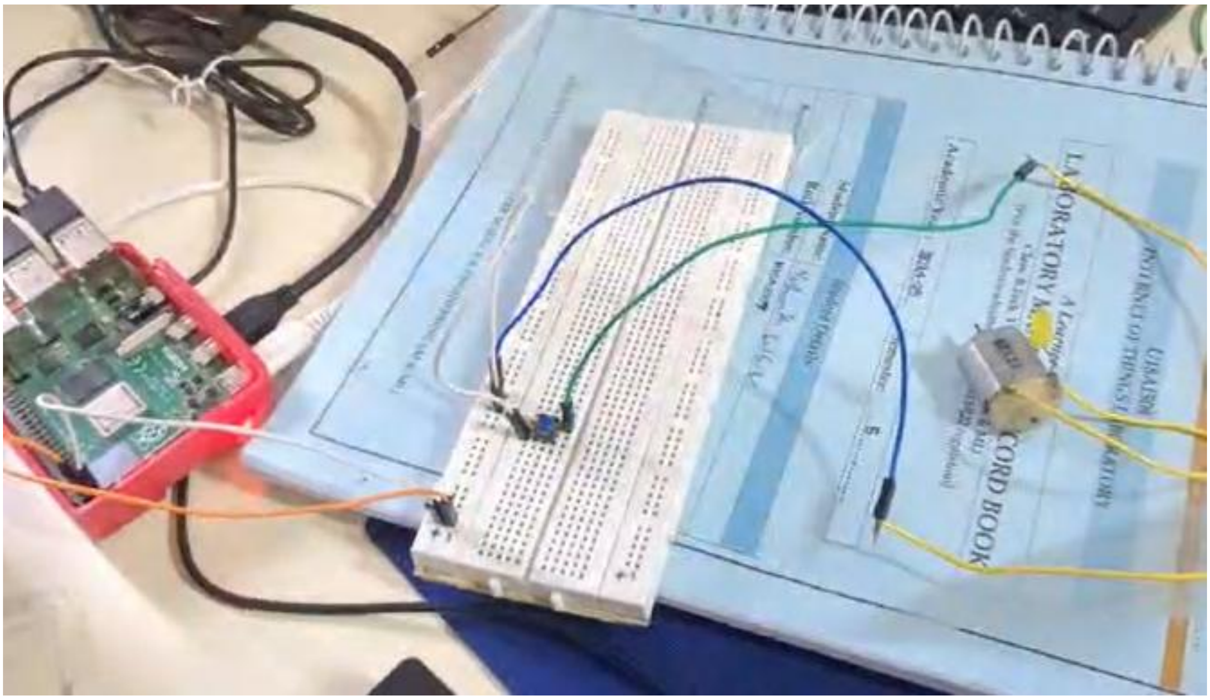


```
import RPi.GPIO as GPIO

import time


# Define the GPIO pin for the servo
```

```
SERVO_PIN = 18  # Change this to your GPIO pin

# Setup GPIO
GPIO.setmode(GPIO.BCM)

GPIO.setup(SERVO_PIN, GPIO.OUT)

# Create a PWM instance with a frequency of 50Hz
pwm = GPIO.PWM(SERVO_PIN,50)
pwm.start(0)  # Start with a duty cycle of 0
GPIO.setwarnings(False)
def set_angle(angle):
    # Convert angle to duty cycle
    duty_cycle = (angle / 18) + 2
    pwm.ChangeDutyCycle(duty_cycle)
    time.sleep(1)  # Allow time for the servo to reach the position
    pwm.ChangeDutyCycle(0)  # Stop sending signal to prevent jitter

try:
    while True:
        set_angle(90)     # Move to 0 degrees
        time.sleep(0.5)
        set_angle(135)    # Move to 135 degrees
        time.sleep(0.5)
        set_angle(180)    # Move to 270 degrees
        time.sleep(0.5)


except KeyboardInterrupt:
    pass  # Exit on Ctrl+C

finally:
    pwm.stop()  # Stop the PWM
    GPIO.cleanup()  # Clean up GPIO pins
```

EXPERIMENT-X

1. Design a program to demonstrate DC motor.



Pins – 3(+ve) , 6(-ve)