Introduction to Raspberrypi

Raspberry Pi is a small **single board computer**. By connecting peripherals like Keyboard, mouse, display to the Raspberry Pi, it will act as a mini personal computer. Raspberry Pi is popularly used for real time Image/Video Processing, IoT based applications and Robotics applications.

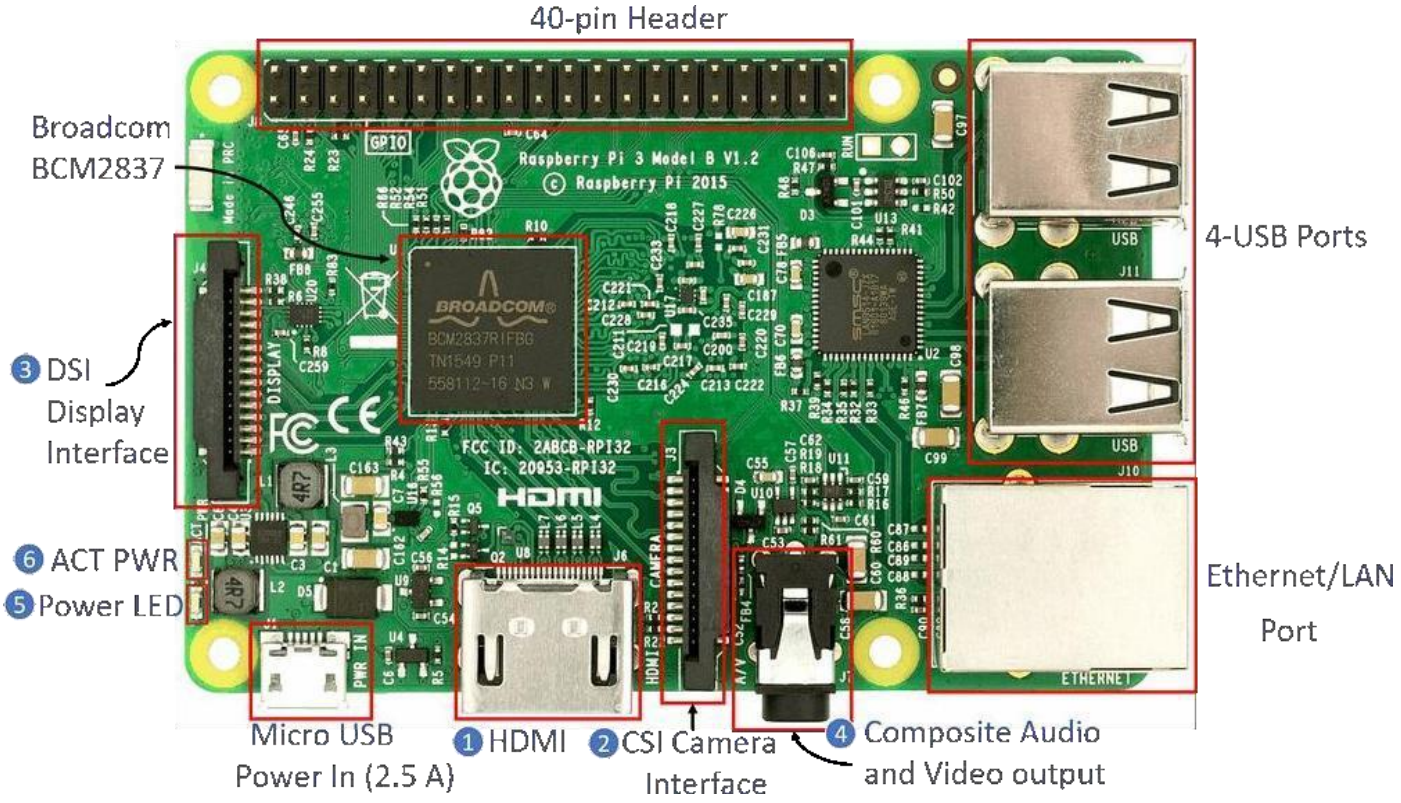


Figure 1: Raspberrypimodel3

Raspberry Pi GPIO Access

Introduction

GPIO (General Purpose Input Output) pins can be used as input or output and allows raspberry pi to connect with general purpose I/O devices.

- Raspberrypi3modelBtookout26GPIOpinsonboard.
- Raspberry pi can control many external I/O devices using these GPIO‟s.
- These pins are a physical interface between the Pi and the outside world.
- We can program these pins according to our needs to interact with external devices. For example, if we want to read the state of a physical switch, we can configure any of the available GPIO pins as input and read the switch status to make decisions. We can also configure any GPIO pin as an output to control LED ON/OFF.
- RaspberryPi can connect to the Internet using on-board Wi-Fi or Wi-Fi USB adapter. Once the Raspberry Pi is connected to the Internet then we can control devices, which are connected to the Raspberry Pi, remotely.

GPIO Pins of Raspberry Pi 3 are shown in below figure 3

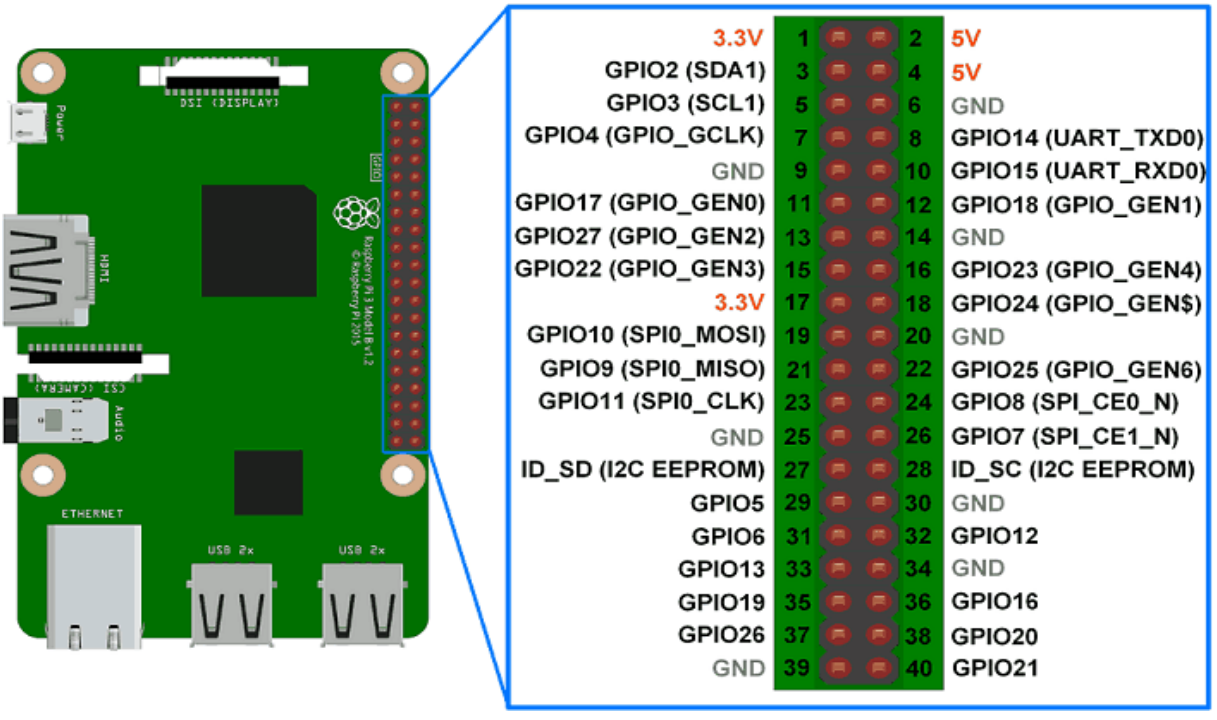


Figure 2: GPIO Pins for Raspberry Pi 3 Model B

Some of the GPIO pins are multiplexed with alternate functions like I2C, SPI, UART etc.

We can use any of the GPIO pins for our application.

- **Pin Numbering**

We should define GPIO pin which we want to use as an output or input. But Raspberry Pi has two ways of defining pin number which are as follows:

GPIO Numbering

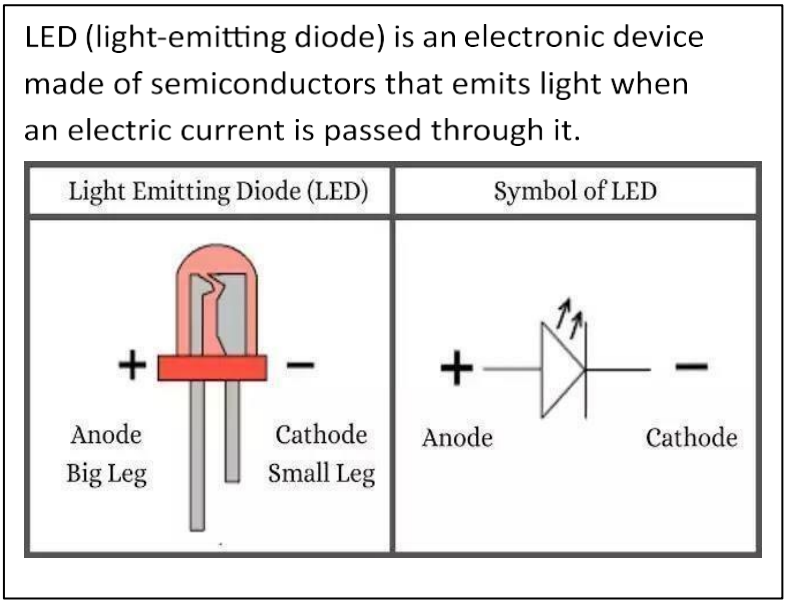
- **Physical Numbering**

In **GPIO Numbering**, pin number refers to number on Broadcom SoC (SystemonChip).So we should always consider the pin mapping focusing GPIO pin.

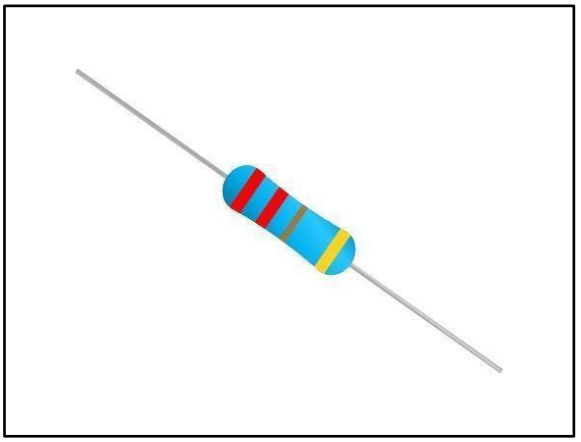
While in **Physical Numbering** pin number refers to the pin of 40-pinP1 header on Raspberry Pi Board. The above physical numbering is simple as we can count pin number on P1 header and assign it as GPIO.

IoT component

LED



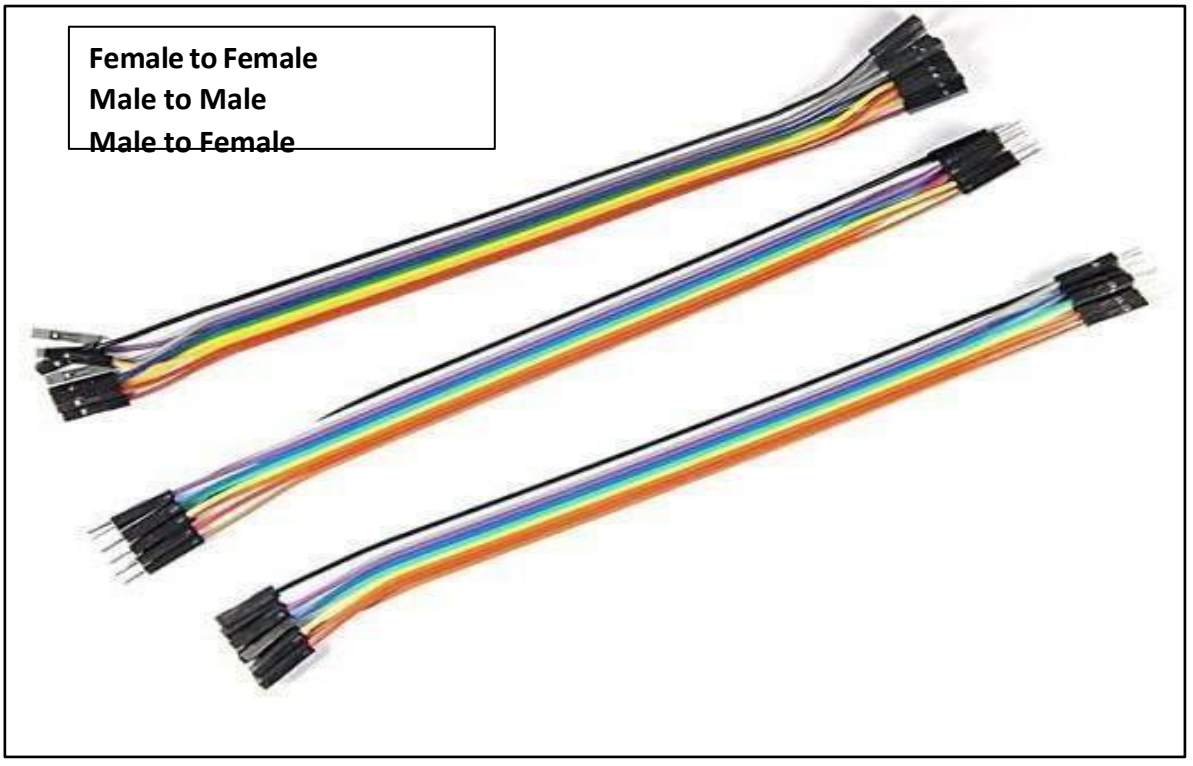
Resistor



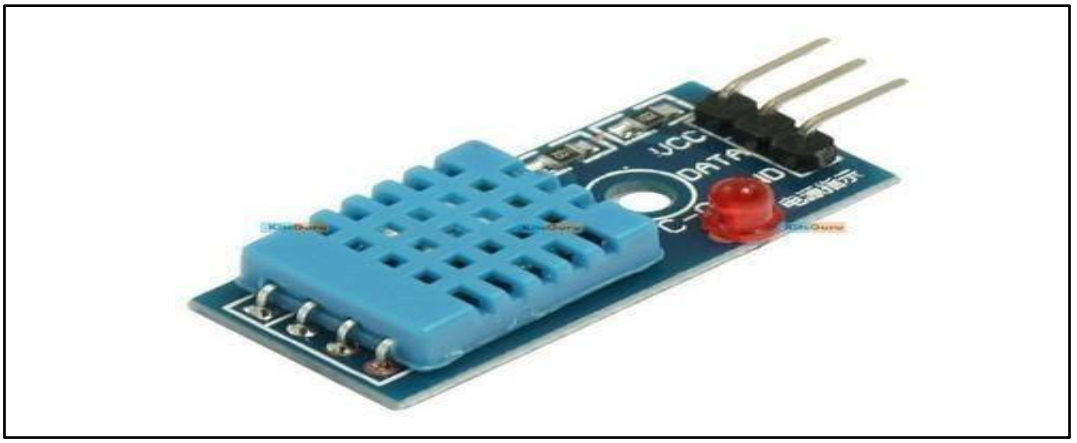
A passive electrical component with two terminals that are used for either limiting or regulating the flow of electric current in electrical circuits.

Jumper Wires

Jumper wires are used on breadboards to „jump" from one connection to another. The ones you will be using in this circuit have different connectors on each end. The end with the „pin" will go into the Breadboard. The end with the piece of plastic with a hole in it will go onto the Raspberry Pi"s GPIO pins.



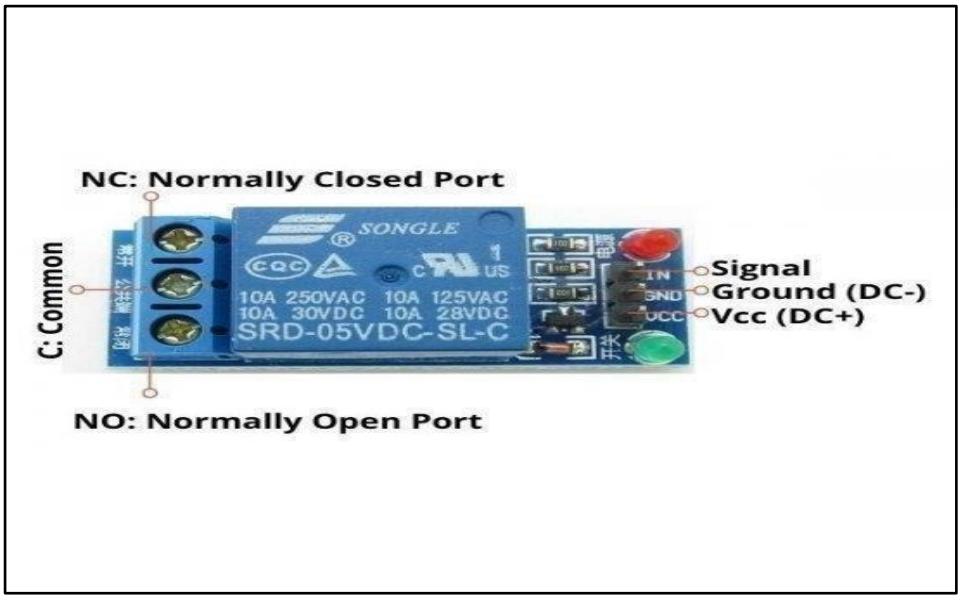
DHT11(Temperature and Humiditysensor)



DHT11 is a low-cost digital sensor for sensing temperature and humidity. This sensor can be easily interfaced with any micro-controller such as Arduino, Raspberry Pi etc to measure humidity and temperature instantaneously.

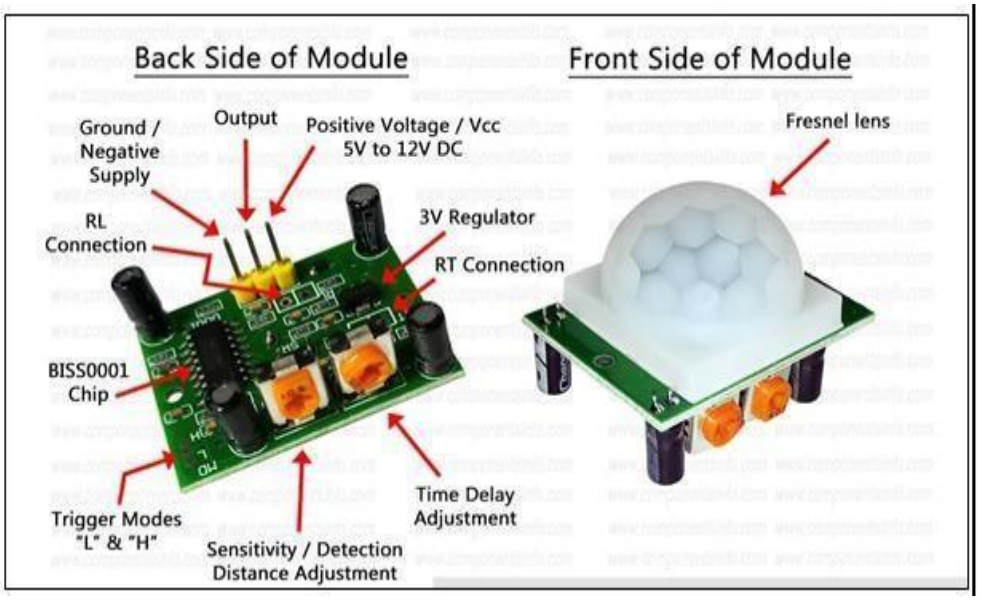
Relay

A relay is a switch. It is an electromagnetic switch where a small control signal at the input of the Relay will control a high voltage supply (usually AC mains).Relays are switches that open and close circuits electromechanically or electronically. Relays control one electrical circuit by opening and closing contacts in another circuit. When a relay contact is Normally Closed (NC), there is a closed contact when the relay is not energized.



PIR (MotionSensor)

PIR sensors allow you to sense motion, almost always used to detect whether a human has moved in or out of the sensors range. They are small, inexpensive, low- power, easy to use and don't wear out. For that reason they are commonly found in appliances and gadgets used in homes or businesses. They are often referred to as PIR, "Passive Infrared", "Pyroelectric", or "IR motion"sensors.

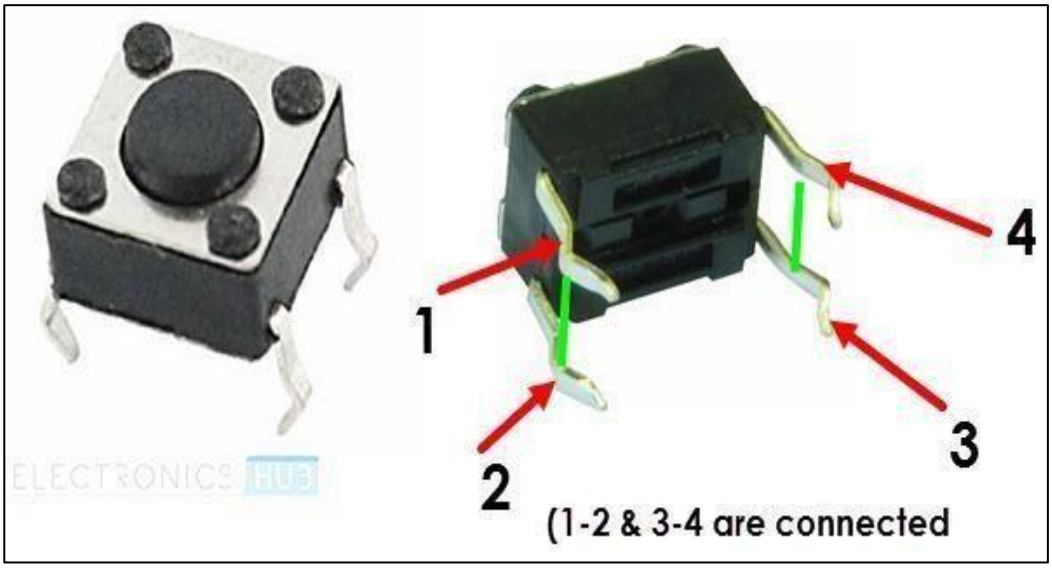


Gas Sensor(MQ-2)

MQ2 gas sensor is an electronic sensor used for sensing the concentration of gases in the air such as LPG, propane, methane, hydrogen, alcohol, smoke and carbon monoxide. MQ2 gas sensor is also known as chemi resistor. It contains a sensing material whose resistance changes when it comes in contact with the gas.

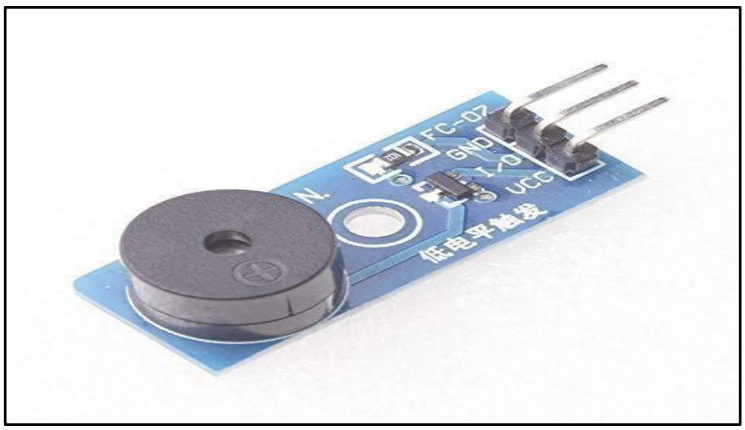
Button Switches

Button switches are an input. The button sends a signal which is received by the Raspberry Pi to be processed. The smallest buttons have four legs that are connected in signal-in and signal-out pairs (polarity does not matter for button switches).



Buzzer

A buzzer or beeper is an audio signaling device, which may be mechanical, electromechanical, or piezoelectric (piezo for short). Typical uses of buzzers and beepers include alarm devices, timers, and confirmation of user input such as a mouse click or keystroke.



Run some python programs on Pi like:

a. Read your name and print Hello message with name

Program

```
name = input("Enter your name: ")
print("Hello ",name)
```

Output

```
Enter your name:  Yogeesh S
Hello  Yogeesh S
```

b. Read two numbers and print their sum, difference, product and division.

Program

```
num1 = int(input("Enter a value for num1: "))
num2 = int(input("Enter a value for num2: "))
print("Sum: ",num1+num2)
print("Difference: ",num1-num2)
print("Product: ",num1*num2)
print("Division: ",num1/num2)
```

Output

```
Enter a value for num1:  6
Enter a value for num2:  3
Sum:  9
Difference:  3
Product:  18
Division:  2.0
```

c. Word and character count of a given string

Program

```
string = input("Enter a string: ")
print("Word count = ",len(string.split(" ")))
print("Character count = ",len(string))
```

Output

```
Enter a string:  Hello i am yogeesh
Word count = 4
Character count = 18
```

d. Area of a given shape (rectangle, triangle and circle) reading shape and appropriate values from standard input

Program

```
print("1. Rectangle\n2. Triangle\n3. Circle\nPress any
other key to exit ")
while True:
      choice = input("Enter a choice: ")
      if(choice == '1'):
            w = int(input("Enter the width: "))
            h = int(input("Enter the height: "))
            print("The area of rectangle is ", w*h)
      elif(choice == '2'):
            b = int(input("Enter breadth: "))
            h = int(input("Enter height: "))
            print("The area of triangle: ", 0.5*b*h)
      elif(choice == '3'):
            radius = int(input("Enter radius: "))
            print("The area of circle: ",
3.14*radius*radius)
            continue
      else:
            break
```

Output

```
1. Rectangle
2. Triangle
3. Circle
Press any other key to exit
Enter a choice:  1
Enter the width:  5
Enter the height:  6
The area of rectangle is  30
Enter a choice:  2
Enter breadth:  6
Enter height:  7
The area of triangle:  21.0
Enter a choice:  3
Enter radius:  5
The area of circle:  78.5
Enter a choice:  4
```

e. Print a name 'n' time, where name and n are read from standard input, using for and while loops.

Program

```
name = input("Enter your name: ")
n = int(input("Enter how many times to print a name: "))
while n>0:
    print(name)
    n = n-1
```

Output

```
Enter your name:  Yogeesh S
Enter how many times to print a name:  3
Yogeesh S
Yogeesh S
Yogeesh S
```

f. Handle Divided by Zero Exception.

Program

```
x = int(input("Enter a value for x: "))
y = int(input("Enter a value for y: "))
try:
    print("x / y = ",x/y)
except ZeroDivisionError:
    print("The value of y not to be zero")
```

```
Enter a value for x:  5
Enter a value for y:  0
The value of y not to be zero
```

g. Print current time for 10 times with an interval of 10 seconds.

Program

```
import time
n = 10
while n>0:
    print(time.ctime(time.time()))
    time.sleep(10)
    n = n-1
```

Output

```
Mon Jan 29 21:34:03 2024
Mon Jan 29 21:34:13 2024
Mon Jan 29 21:34:23 2024
Mon Jan 29 21:34:33 2024
Mon Jan 29 21:34:43 2024
Mon Jan 29 21:34:53 2024
Mon Jan 29 21:35:03 2024
Mon Jan 29 21:35:13 2024
Mon Jan 29 21:35:23 2024
Mon Jan 29 21:35:33 2024
```

h. Read a file line by line and print the word count of each line

Program

```
lines = open("Program1.txt",'r').readlines()
for i in range(len(lines)):
    print (f"Line {i+1} Words : {len(lines[i].split(" "))}")
```

Output

```
Line 1 Words : 5
Line 2 Words : 4
Line 3 Words : 3
```

2. Input from two switches and switch on corresponding LEDs

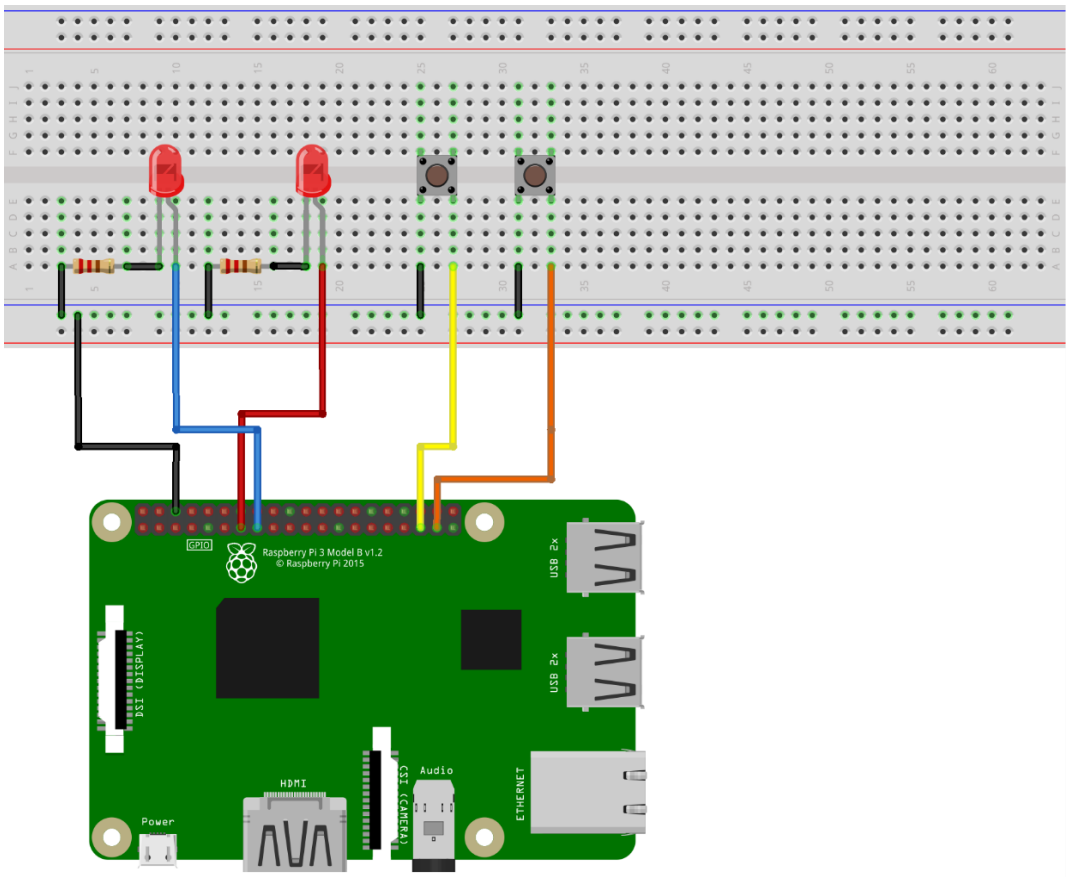
Program

```
import time
import RPi.GPIO as gpio

gpio.setwarnings(False)
gpio.setmode(gpio.BOARD)

led1 = 15
led2 = 13
switch1 = 37
switch2 = 35

gpio.setup(led1,gpio.OUT,initial=0)
gpio.setup(led2,gpio.OUT,initial=0)
gpio.setup(switch1,gpio.IN)
gpio.setup(switch2,gpio.IN)

def glow_led(event):
    if event == switch1:
        gpio.output(led1, True)
        time.sleep(3)
        gpio.output(led1, False)

    elif event == switch2:
        gpio.output(led2, True)
        time.sleep(3)
        gpio.output(led2, False)

gpio.add_event_detect(switch1, gpio.RISING , callback =
glow_led, bouncetime = 1)
gpio.add_event_detect(switch2, gpio.RISING , callback =
glow_led, bouncetime = 1)

try:
    while(True):
       time.sleep(1)
```

NOTE: The below line in program it's a single line statement

```
gpio.add_event_detect(switch1, gpio.RISING , callback =
glow_led, bouncetime = 1)
```

Circuit



3. Flash an LED at a given on time and off time cycle, where the two times are taken from a file

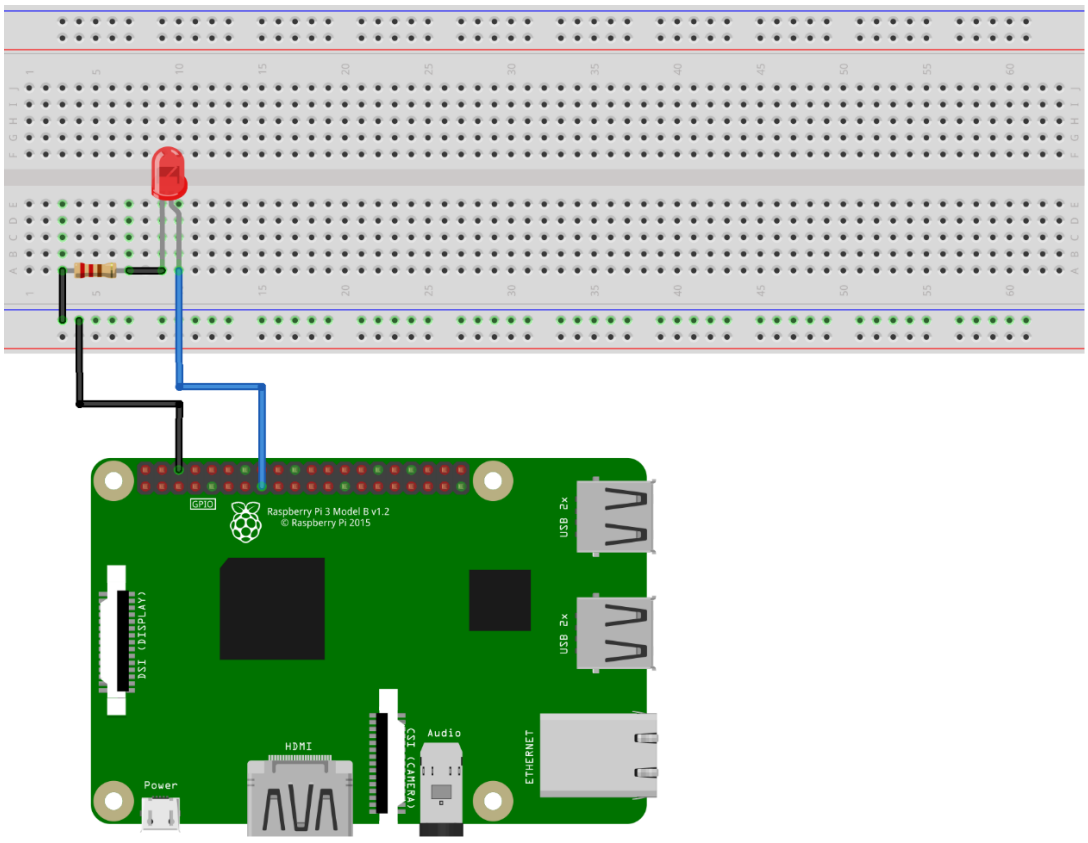
Program

```
import time
import RPi.GPIO as gpio

gpio.setwarnings(False)
gpio.setmode(gpio.BOARD)

led1 =  15
gpio.setup(led1,gpio.OUT,initial=0)

file1 = open('program3.txt', 'r')
Lines = file1.readlines()

ON_TIME = int(Lines[0].split("=")[1])
OFF_TIME = int(Lines[1].split("=")[1])

try:
    while(True):
        gpio.output(led1,True)
        time.sleep(ON_TIME)
        gpio.output(led1,False)
        time.sleep(OFF_TIME)
except KeyboardInterrupt:
    gpio.cleanup()
```

Program3.txt

```
ontime = 5
offtime = 7
```

Circuit


GPIO
Raspberry Pi 3 Model B v1.2
© Raspberry Pi 2015
USB 2x
USB 2x
DSI (DISPLAY)
HDMI
CSI (CAMERA)
Audio
ETHERNET
Power

4. Switch on a relay at a given time using cron, where the relay's contact terminals are connected to a load.

Program

```
import RPi.GPIO as gpio
import time

gpio.setwarnings(False)
gpio.setmode(gpio.BOARD)

relay = 38
gpio.setup(relay, gpio.OUT)

try:
    gpio.output(relay, False)
    time.sleep(10)
    gpio.output(relay, True)

except KeyboardInterrupt:
    gpio.cleanup()
```

Crontab Setting

1. GOTO Terminal

2. TYPE select-editor

1. /bin/nano <---- easiest

2. /usr/bin/vim.tiny

3. /bin/ed

3. ENTER 2

4. TYPE crontab -e

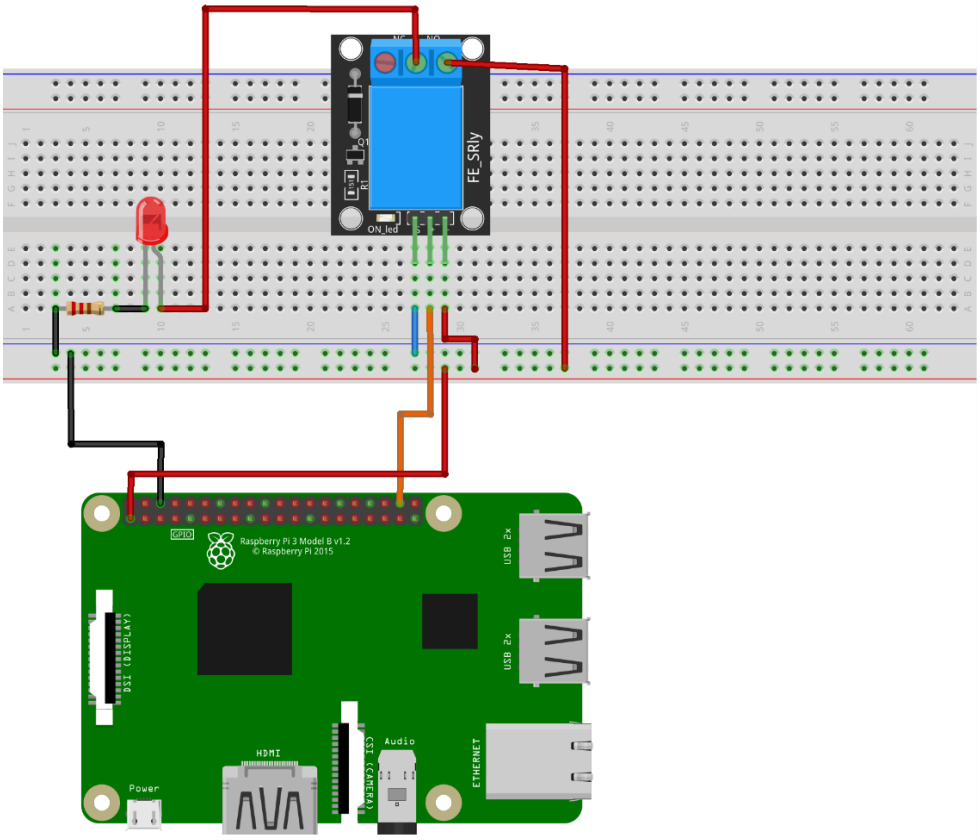
go end and press i

02 12 * * * python3 /home/yogeesh/Programs/program4.py

5. Press ESCAPE button

6. ENTER :wq

Circuit

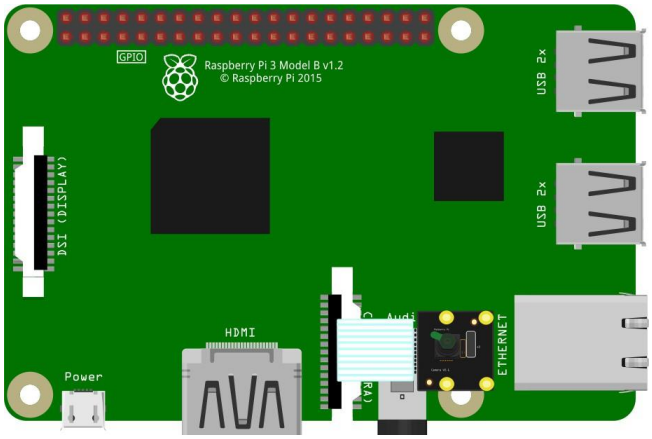

FE_SRly
ON_led
GPIO
Raspberry Pi 3 Model B v1.2
© Raspberry Pi 2015
USB 2x
USB 2x
DSI (DISPLAY)
HDMI
CSI (CAMERA)
Audio
ETHERNET
Power

5. Access an image through a Pi web cam

Program for Rasberry PI 3

```
from picamera import PiCamera
from time import sleep
camera=PiCamera()
camera.start_preview()
sleep(10)
camera.capture(['/home/pi/image02.jpg'])
camera.stop_preview()
```

Circuit



6. Control a light source using web page.

Python Program

```
import RPi.GPIO as GPIO
from flask import Flask, render_template

led_pin = 15
app = Flask(__name__)

GPIO.setmode(GPIO.BOARD)
GPIO.setwarnings(False)
GPIO.setup(led_pin, GPIO.OUT, initial=0)


@app.route('/')
def index():
    return render_template('index.html')

@app.route('/ledon')
def ledon():
    GPIO.output(led_pin, GPIO.HIGH)
    return render_template('index.html')

@app.route('/ledoff')
def ledoff():
    GPIO.output(led_pin, GPIO.LOW)
    return render_template('index.html')

if __name__ == "__main__":
        app.run(debug=False, port=4000, host='0.0.0.0')
```

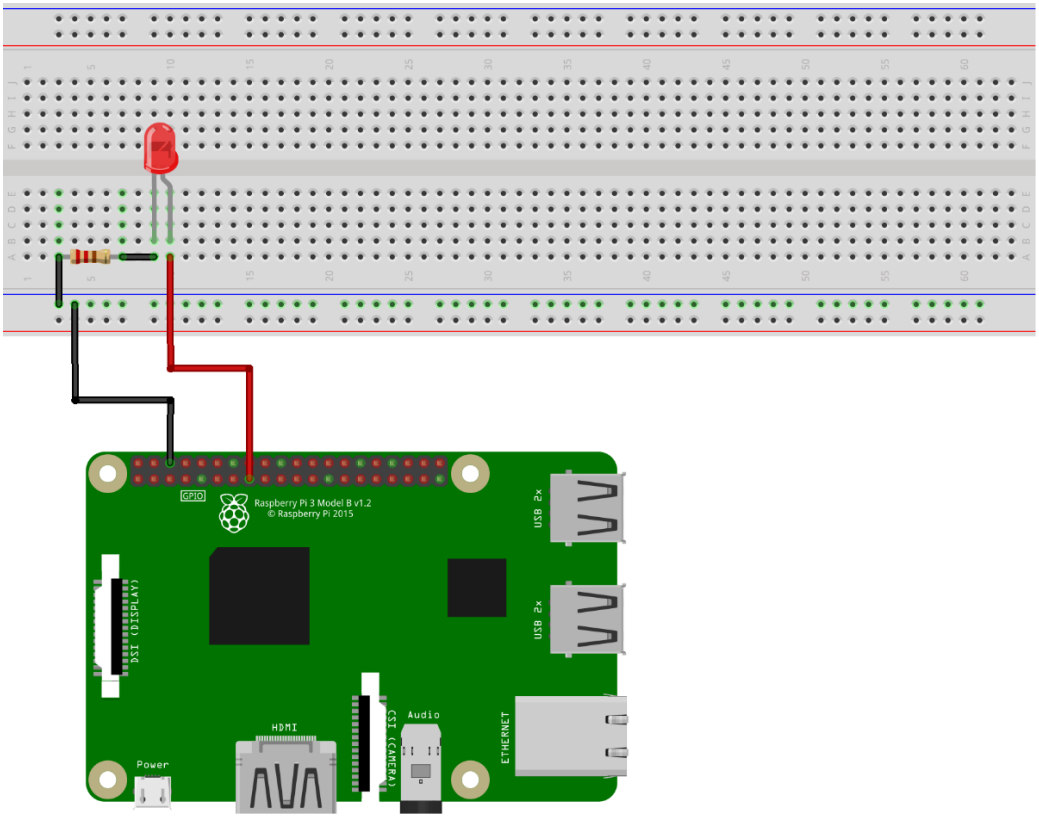
NOTE: File Structure of This Program given below

Your Programs Directory/
|
|--Program6.py
|--templates/
|--index.html

HTML Code (index.html)

```
<!DOCTYPE html>
<html>
<head>
    <title>LED Control</title>
</head>
<body>
    <h1>Control LED</h1>
    <form method="POST">
        <button name="on" type="submit">Turn ON</button>
        <button name="off" type="submit">Turn OFF</button>
    </form>
</body>
</html>
```

Circuit

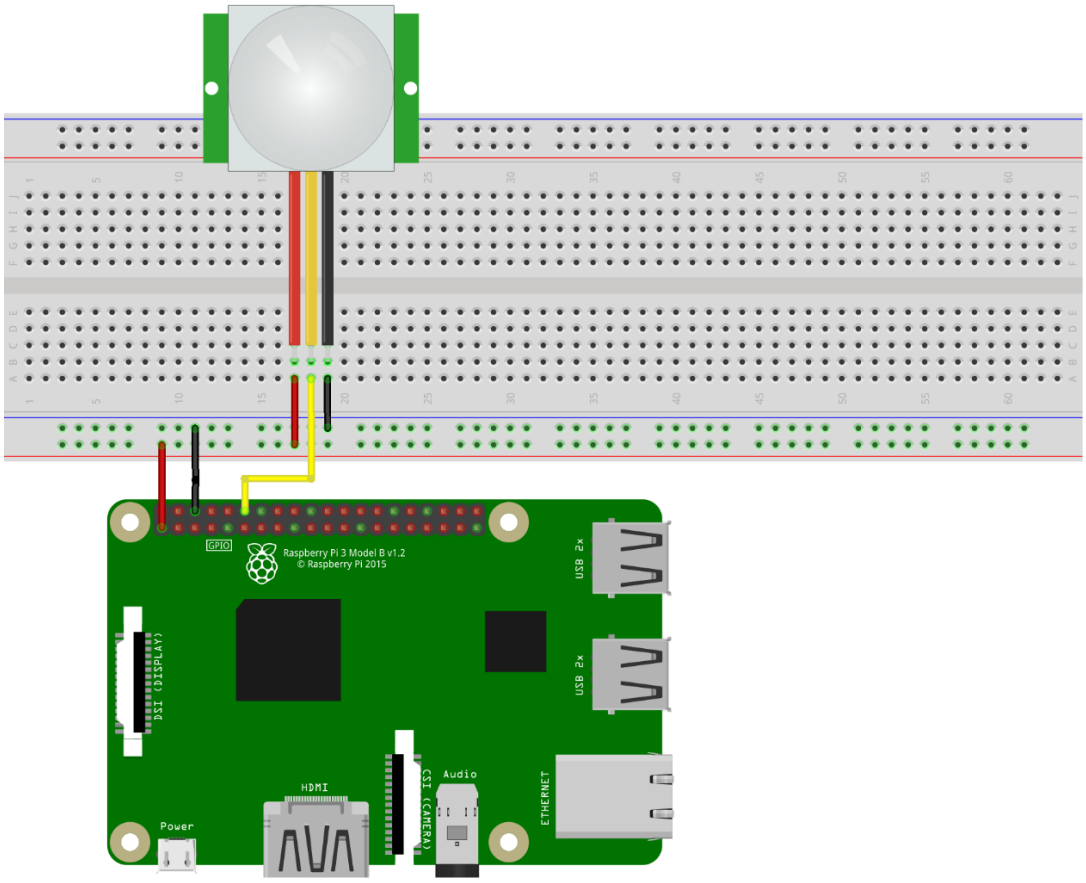


7. Implement an intruder system that sends an alert to the given email

Program

```
import RPi.GPIO as GPIO
import time
import smtplib
from email.mime.text import MIMEText

# Set your Gmail details here
EMAIL_ADDRESS = "iotxmca@gmail.com"
EMAIL_PASSWORD = "hezr rgch bffu ioyl"
TO_EMAIL = "amanxamn@gmail.com"

# Setup
GPIO.setmode(GPIO.BCM)
PIR_PIN = 17
GPIO.setup(PIR_PIN, GPIO.IN)

def send_email():
    msg = MIMEText("Intruder detected!")
    msg["Subject"] = "□ Intruder Alert!"
    msg["From"] = EMAIL_ADDRESS
    msg["To"] = TO_EMAIL

    try:
        with smtplib.SMTP_SSL("smtp.gmail.com", 465) as smtp:
            smtp.login(EMAIL_ADDRESS, EMAIL_PASSWORD)
            smtp.send_message(msg)
        print("Email sent!")
    except Exception as e:
        print("Failed to send email:", e)

print("Monitoring for intruders...")

try:
    while True:
        if GPIO.input(PIR_PIN):
            print("Intruder detected!")
            send_email()
            time.sleep(10)  # wait 10 seconds before next detection
        time.sleep(1)
except KeyboardInterrupt:
    print("Program stopped")
finally:
    GPIO.cleanup()
```

Circuit


GPIO
Raspberry Pi 3 Model B v1.2
© Raspberry PI 2015
USB 2x
USB 2x
DSI (DISPLAY)
HDMI
CSI (CAMERA)
Audio
ETHERNET
Power

8. Get the status of a bulb at a remote place (on the LAN) through web.

Program

Circuit

Python Program

```
import RPi.GPIO as GPIO
from flask import Flask, render_template

led_pin = 15
app = Flask(__name__)

GPIO.setmode(GPIO.BOARD)
GPIO.setwarnings(False)
GPIO.setup(led_pin, GPIO.OUT, initial=0)

def get_status():
    return {'status': 'ON' if GPIO.input(led) == GPIO.HIGH
else 'OFF'}

@app.route('/')
def index():
    return render_template('index.html', **get_status())

@app.route('/ledon')
def ledon():
    GPIO.output(led_pin, GPIO.HIGH)
    return render_template('index.html', **get_status())

@app.route('/ledoff')
def ledoff():
    GPIO.output(led_pin, GPIO.LOW)
    return render_template('index.html', **get_status())

if __name__ == "__main__":
        app.run(debug=False, port=4000, host='0.0.0.0')
```

NOTE: File Structure of This Program given below

Your Programs Directory/
|
|--Program6.py
|--templates/
|--index.html

HTML Code (index.html)

```
<html>
<head>
   <meta name="viewport" content="width=device-width, initial-
scale=1.0">
    <title>Raspberry PI Remote Control</title>
</head>
<body>
    <h1>Raspberry PI Remote Control</h1>
    <h2>Light Status: {{ status }}</h2>
     <form action="ledon">
          <input type="submit" value="LED On">
     </form>
     <form action="ledoff">
          <input type="submit" value="LED Off">
     </form>
</body>
</html>
```

Circuit