Assignment 5

Name: Ebenezer Isaac Veeraraju

Roll No: 2020178014

Batch: MCA Regular

Teammate: Dennis Churchill J

Teammate Roll No: 2020178011

Components Used:

Raspberry Pi x1

Bread Board x1

Push Button x1

Photoresistor x1

LED Lights x3

Potentiometer x1

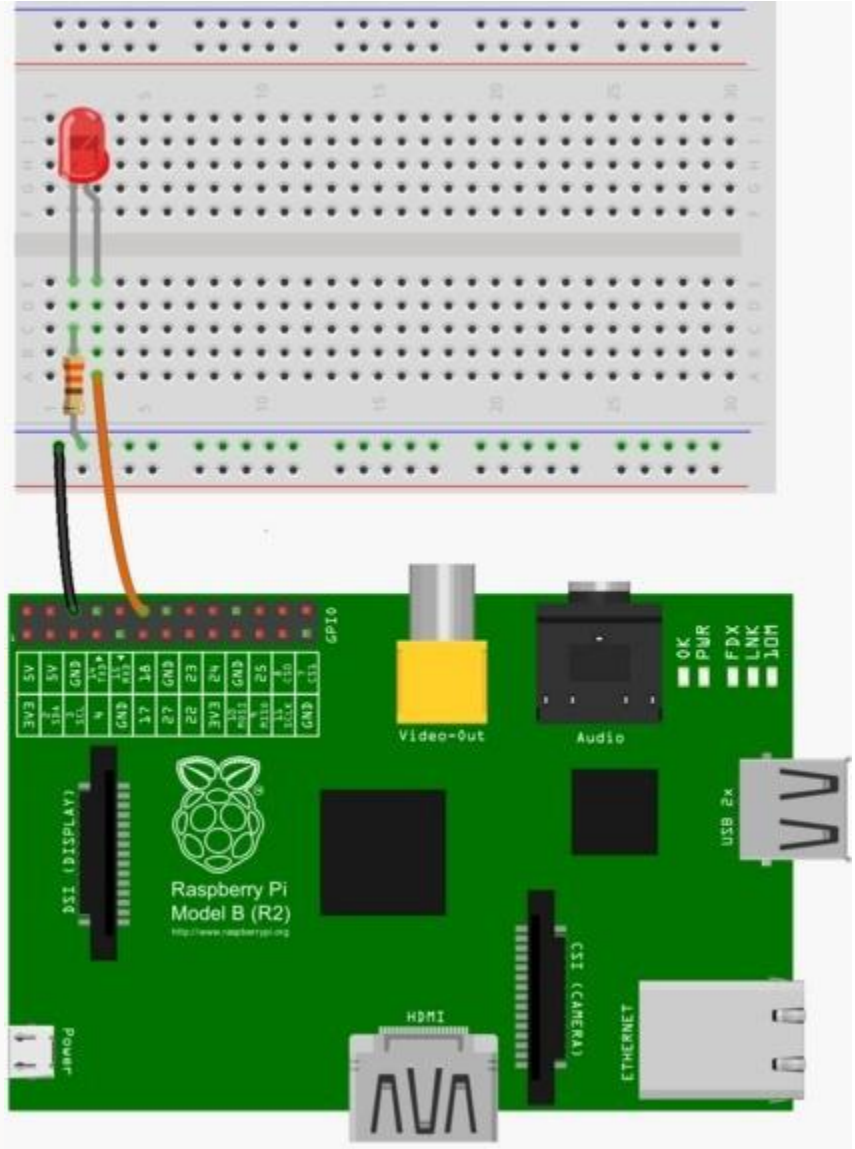
Resistors x3

Wires x4

Task 1: Build a very simple circuit consisting of an LED and resistor connected to the GPIO port on your Raspberry Pi and make it to blink.

Aim: To Build a very simple circuit consisting of an LED and resistor connected to the GPIO port on your Raspberry Pi and make it to blink.

Circuit:



```
import RPi.GPIO as GPIO
import time
GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)
GPIO.setup(18,GPIO.OUT)
while True:
        print("LED On")
        GPIO.output(18,GPIO.HIGH)
        time.sleep(1)
        print("LED Off")
        GPIO.output(18,GPIO.LOW)
        time.sleep(1)
```

Result:

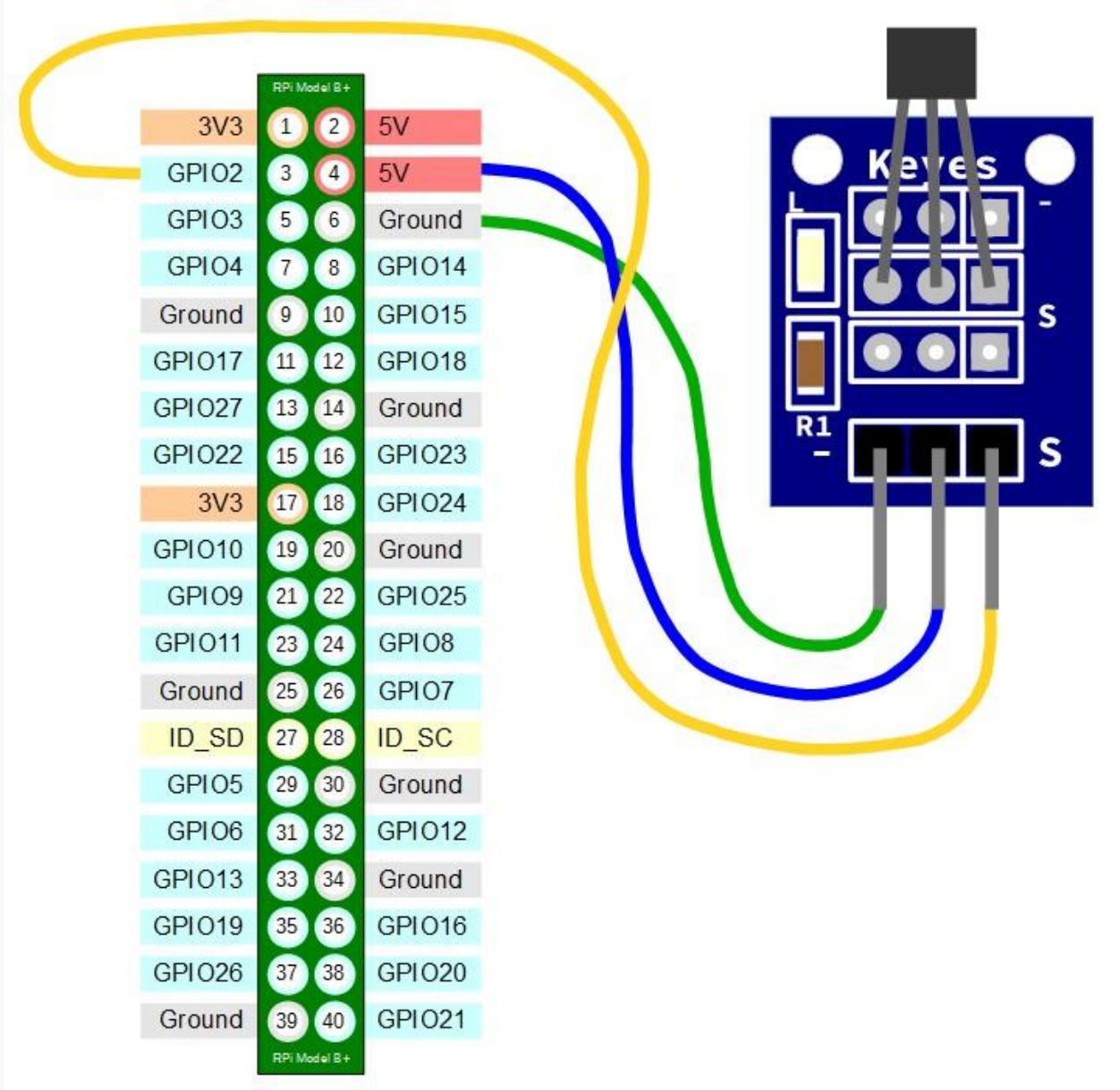
The Output For Given Tasks Were Obtained Successfully.

Task 2: Using the same format, try to create a new project to make the photoresistor to trigger the LED, like a sustainable energy saving device.

Aim:

To create a new project to make the photoresistor to trigger the LED, like a sustainable energy saving device.

Circuit:



```
import RPi.GPIO as GPIO
from time import sleep
GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)
GPIO.setup(2,GPIO.OUT)
GPIO.output(2,False)
GPIO.setup(2,GPIO.IN)
GPIO.setup(18,GPIO.OUT)
while True:
        ldr = GPIO.input(2)
        if ldr==1:
                GPIO.output(18,GPIO.HIGH)
        else:
                GPIO.output(18,GPIO.LOW)
        sleep(0.1)
```

Result:

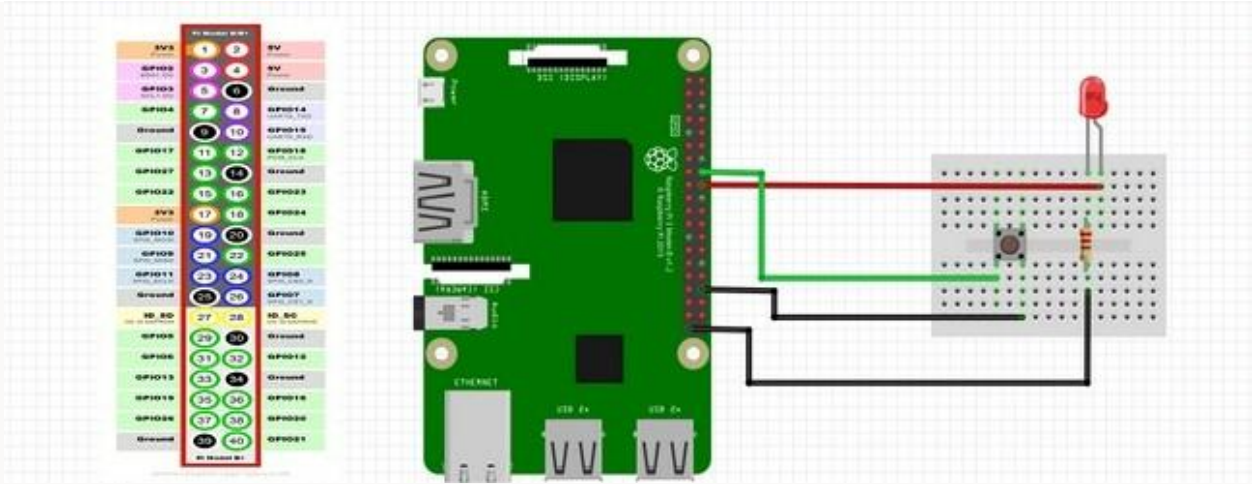
The Output For Given Task Was Obtained Successfully.

Task 3: Using the same format, try to create a new project to trigger the LED using the pushbutton to control it.

Aim:

To create a new project to trigger the LED using the pushbutton to control it.

Circuit:



```
import RPi.GPIO as GPIO
from time import sleep
GPIO.setwarnings(False)
GPIO.setmode(GPIO.BCM)
button = 23
led = 24
GPIO.setup(button,GPIO.IN,pull_up_down=GPIO.PUD_UP)
GPIO.setup(led,GPIO.OUT)
while True:
        button_state = GPIO.input(button)
        if button_state == 0:
                GPIO.output(led,GPIO.HIGH)
        else:
                GPIO.output(led,GPIO.LOW)
        sleep(1)
```

Result:

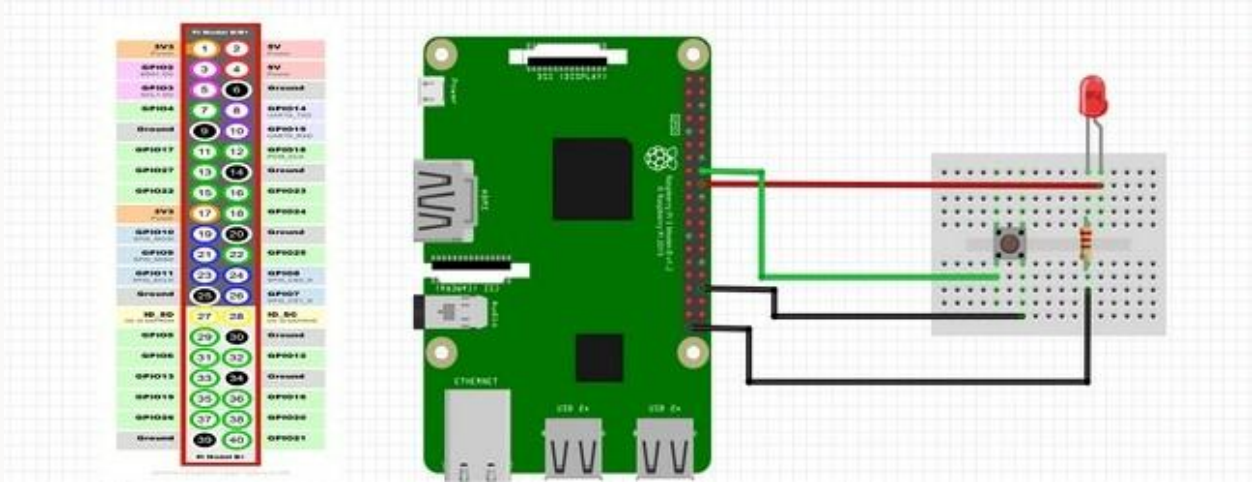
The Output For Given Task Was Obtained Successfully.

Task 4: Create a new project to make a LED fading using the potentiometer.

Aim:

To Create a new project to make a LED fading using the potentiometer.

Circuit:



Task 4's circuit is similar to task 3's with the button being replaced by the potentiometer. Here, the Ground goes to first pin from the left, pin 24 goes to second and 3.3V goes to third

```
import RPi.GPIO as GPIO
from time import sleep
GPIO.setwarnings(False)
GPIO.setmode(GPIO.BCM)
button = 23
led = 24
GPIO.setup(button,GPIO.IN,pull_up_down=GPIO.PUD_UP)
GPIO.setup(led,GPIO.OUT)
while True:
        button_state = GPIO.input(button)
        if button_state == 0:
                GPIO.output(led,GPIO.HIGH)
        else:
                GPIO.output(led,GPIO.LOW)
        sleep(1)
```

Result:

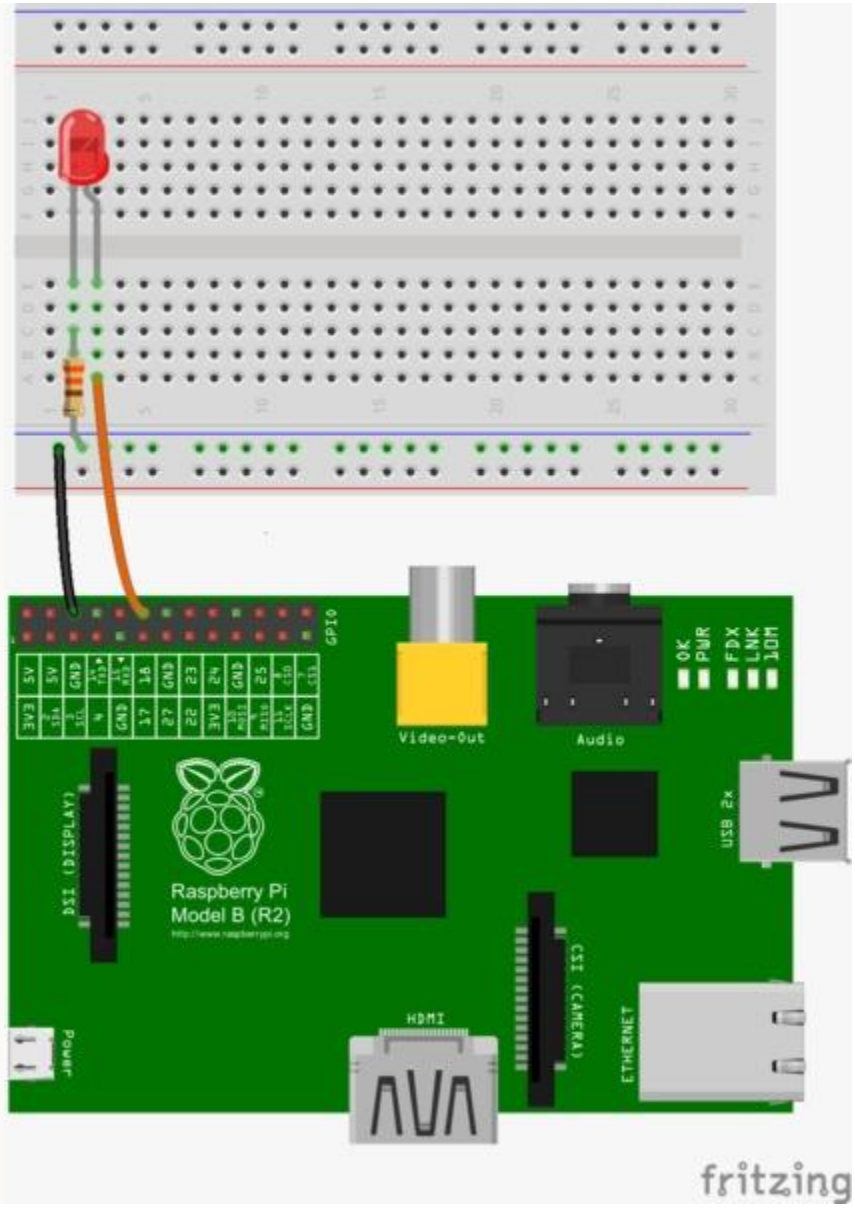
The Output For Given Task Was Obtained Successfully.

Task 5: Try adding other LEDs and develop your own morse code flashing

Aim:

To develop a morse code flashing

Circuit:



.json file:


```
{
  "0": "-----",
  "1": ".----",
  "2": "..---",
  "3": "...--",
  "4": "....-",
  "5": ".....",
  "6": "-....",
  "7": "--...",
  "8": "---..",
  "9": "----.",
  "a": ".-",
  "b": "-...",
  "c": "-.-.",
  "d": "-..",
  "e": ".",
  "f": "..-.",
  "g": "--.",
  "h": "....",
  "i": "..",
  "j": ".---",
  "k": "-.-",
  "l": ".-..",
```

```
  "m": "--",
  "n": "-.",
  "o": "---",
  "p": ".--.",
  "q": "--.-",
  "r": ".-.",
  "s": "...",
  "t": "-",
  "u": "..-",
  "v": "...-",
  "w": ".--",
  "x": "-..-",
  "y": "-.--",
  "z": "--..",
  ".": ".-.-.-",
  ",": "--..--",
  "?": "..--..",
  "!": "-.-.--",
  "-": "-....-",
  "/": "-..-.",
  "@": ".--.-.",
  "(": "-.--.",
  ")": "-.--.-"
}
```

Python Code:

```
import RPi.GPIO as GPIO
import json, time
GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)
GPIO.setup(18,GPIO.OUT)
GPIO.output(18,GPIO.LOW)
morse = {}
unit = 0.250
with open("morse-code.json") as json_file:
        morse = json.load(json_file)
text = "Ebenezer"
text = text.lower()
print("Morse Code for :",text)
for letter in text:
        if letter==' ':
                sleep(unit*7)
        code = morse[letter]
        print(letter,":",code)
        for part in code:
                if part=='.':
                        GPIO.output(18,GPIO.HIGH)
                        time.sleep(unit)
                elif part=='-':
```

```
                    GPIO.output(18,GPIO.HIGH)
                    time.sleep(unit*3)
                GPIO.output(18,GPIO.LOW)
                time.sleep(unit)
            GPIO.output(18,GPIO.LOW)
            time.sleep(unit*3)
```

Result:

The Output For Given Task Was Obtained Successfully.