

Subject :IOT

# Practical

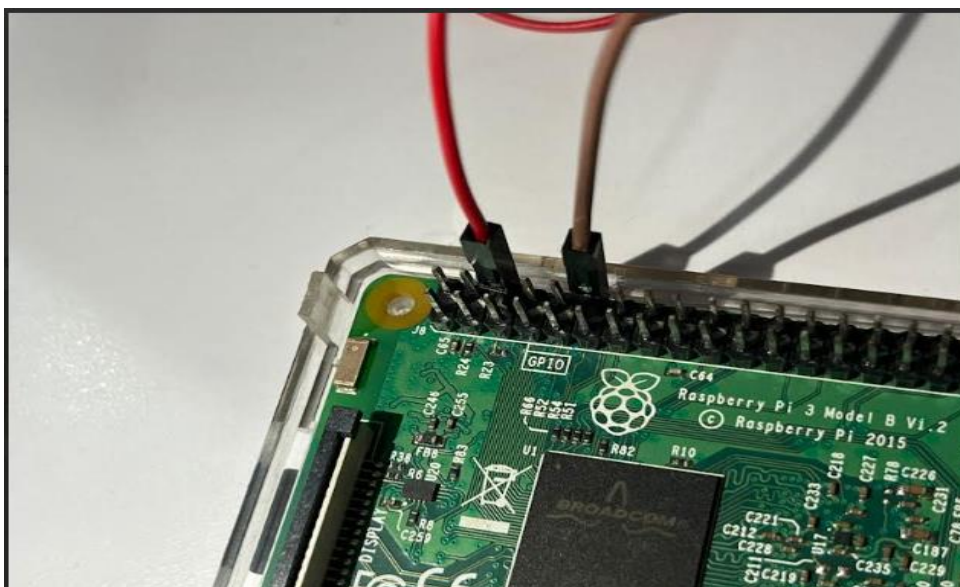
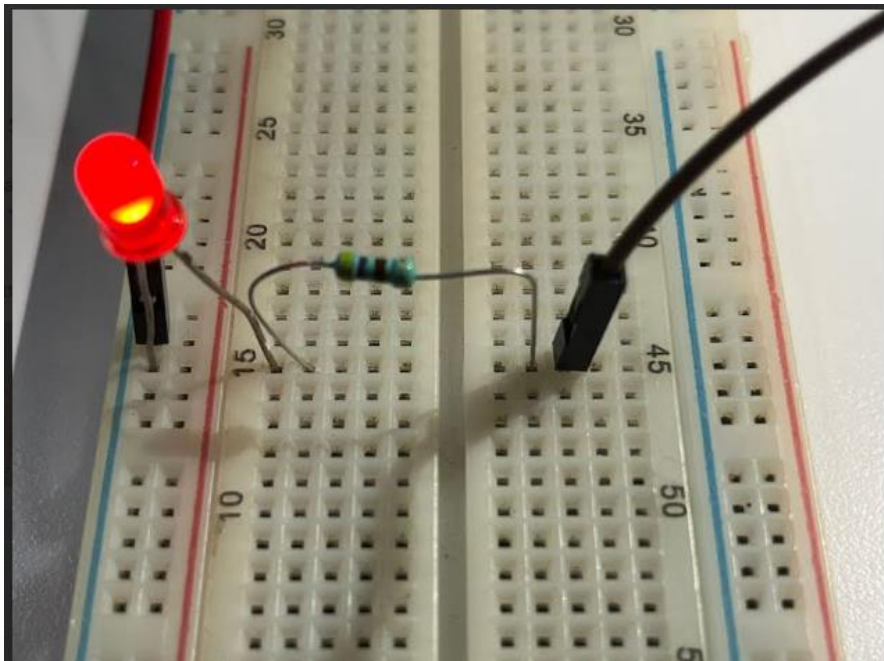
## 1.Blinking single LED

### Connection

(led:cathode(smaller side) on -ve side of the breadboard)

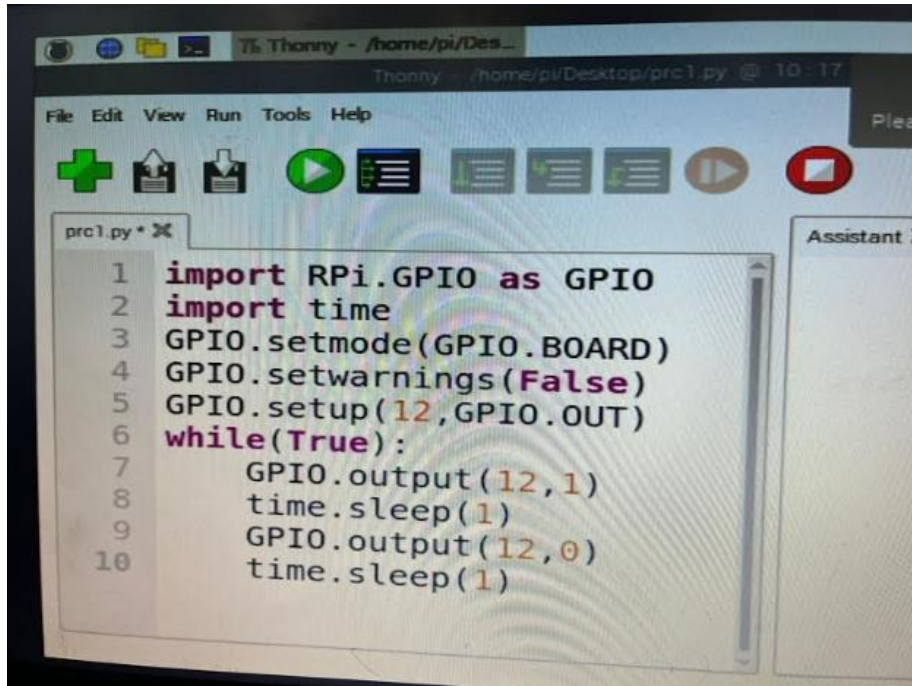
One male to female wire on Ground pin 6

Second male to female wire on GPIO pin 12



Subject :IOT

Code:



The image shows a screenshot of a Thonny IDE window. The title bar indicates the file path is `/home/pi/Desktop/prc1.py` and the time is 10:17. The menu bar includes File, Edit, View, Run, Tools, and Help. Below the menu bar is a toolbar with icons for opening files, saving, running, and other functions. The main editor area displays a Python script named `prc1.py` with the following code:

```
1 import RPi.GPIO as GPIO
2 import time
3 GPIO.setmode(GPIO.BOARD)
4 GPIO.setwarnings(False)
5 GPIO.setup(12, GPIO.OUT)
6 while(True):
7     GPIO.output(12, 1)
8     time.sleep(1)
9     GPIO.output(12, 0)
10    time.sleep(1)
```

On the right side of the window, there is a panel labeled "Assistant".

**Subject :IOT**

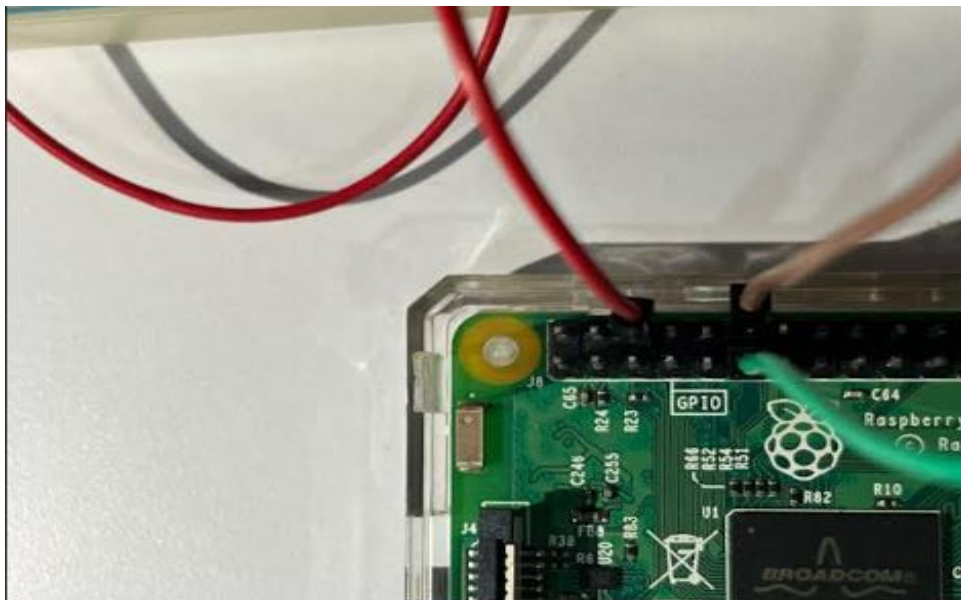
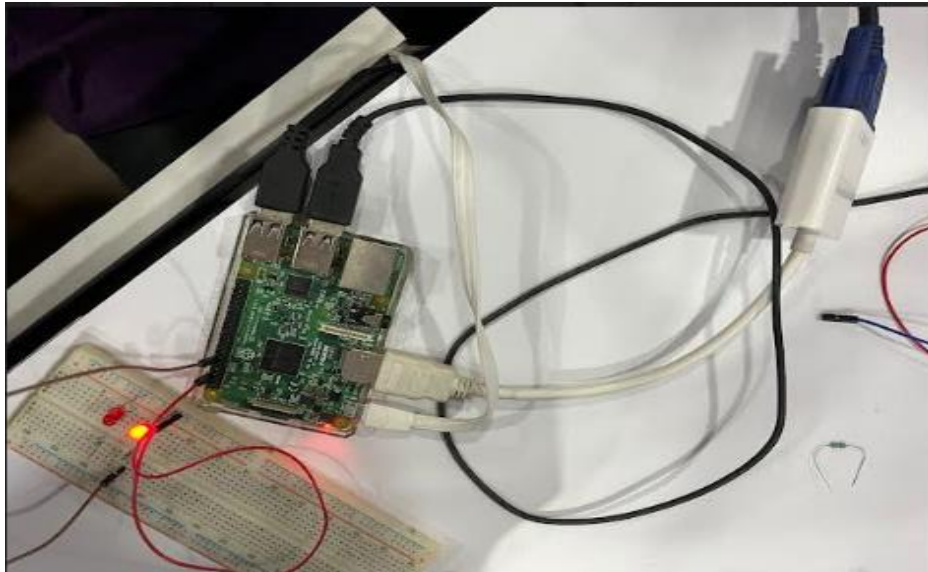
## **2.Toggle 2 LED**

**One male to female wire on Ground pin 6**

**Second male to female wire on GPIO pin 12**

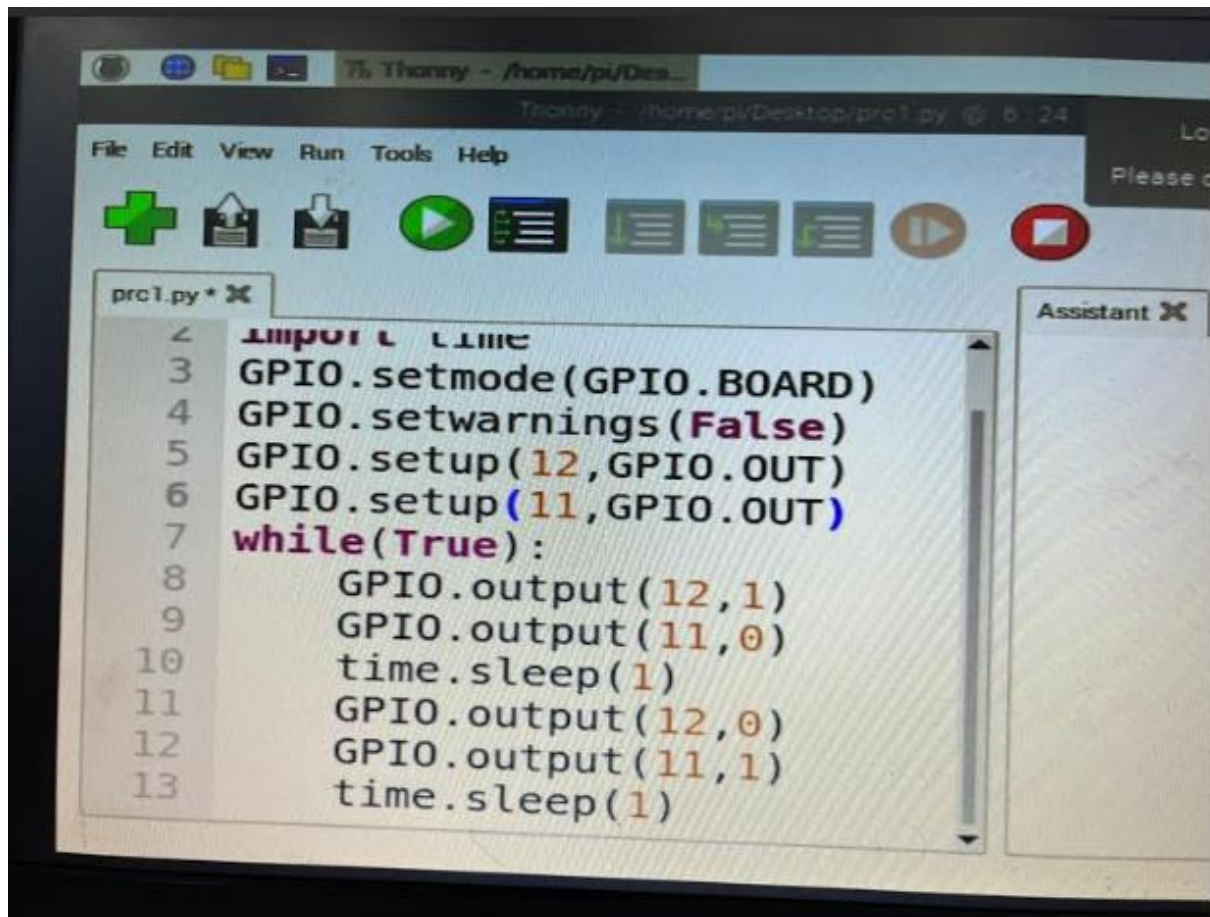
**THIRD male to female wire on GPIO pin 11**

**Connection**



**Code:**

## Subject :IOT





**Subject :IOT**

### **3.Traffic signal**

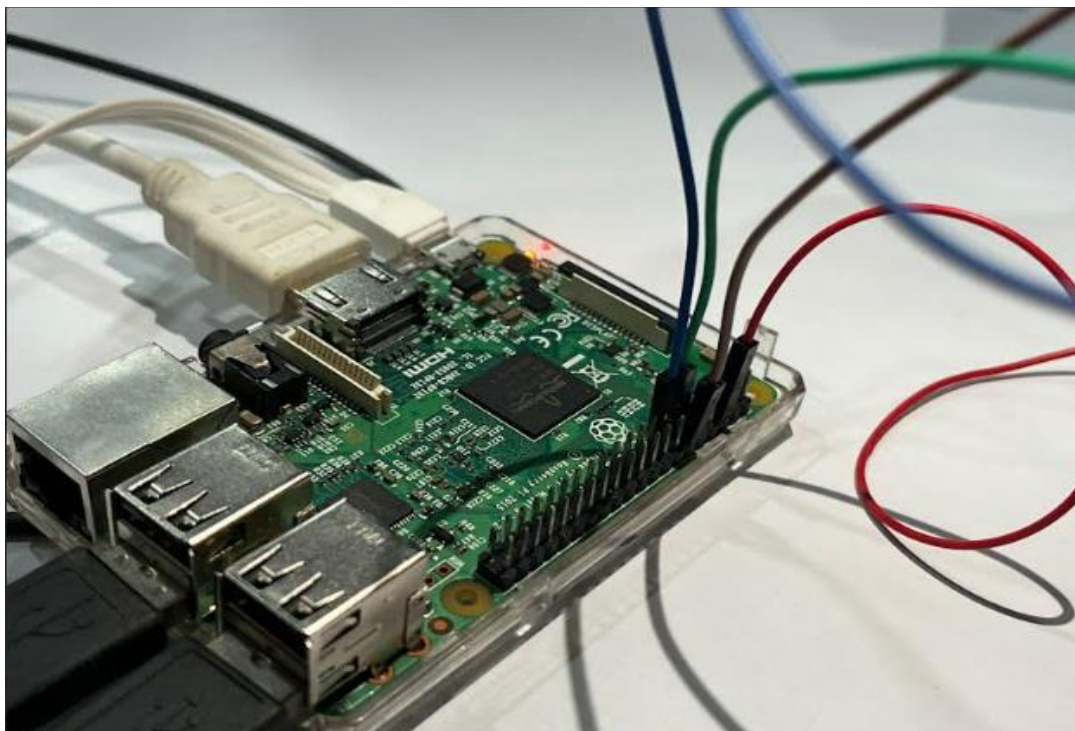
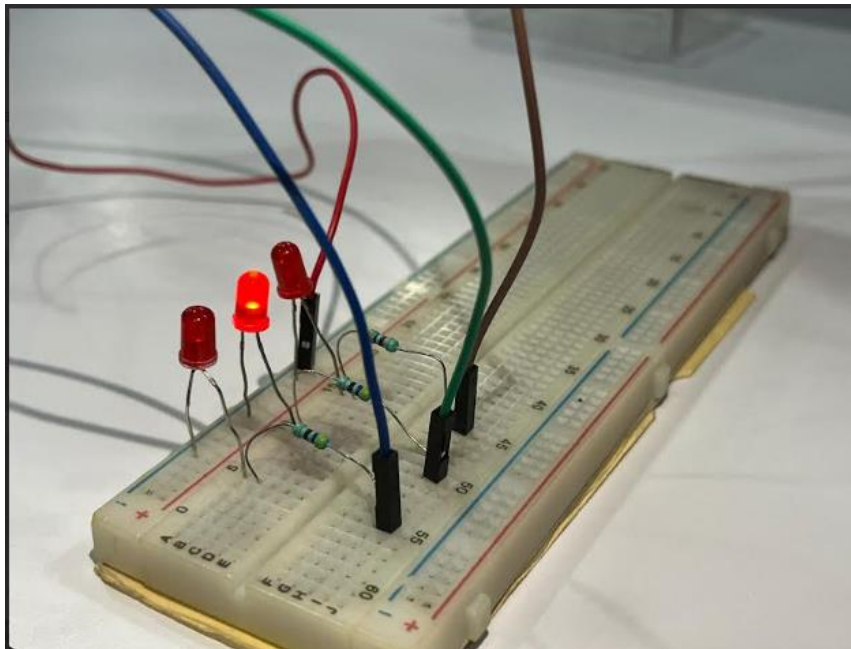
**One male to female wire on Ground pin is 6**

**Second male to female wire on GPIO pin 12**

**THIRD male to female wire on GPIO pin 11**

**Fourth male to femal wire on GPIO pin 13**

#### **Connection**



Subject :IOT

Code:

```
GPIO.setup(12,GPIO.OUT)
GPIO.setup(11,GPIO.OUT)
GPIO.setup(13,GPIO.OUT)
while(True):
    GPIO.output(12,1)
    GPIO.output(11,0)
    GPIO.output(13,0)
    time.sleep(1)
    GPIO.output(12,0)
    GPIO.output(11,1)
    GPIO.output(13,0)
    time.sleep(1)
```

```
16 time.sleep(1)
17 GPIO.output(12,0)
18 GPIO.output(11,0)
19 GPIO.output(13,1)
20 time.sleep(1)
```

**Subject :IOT**

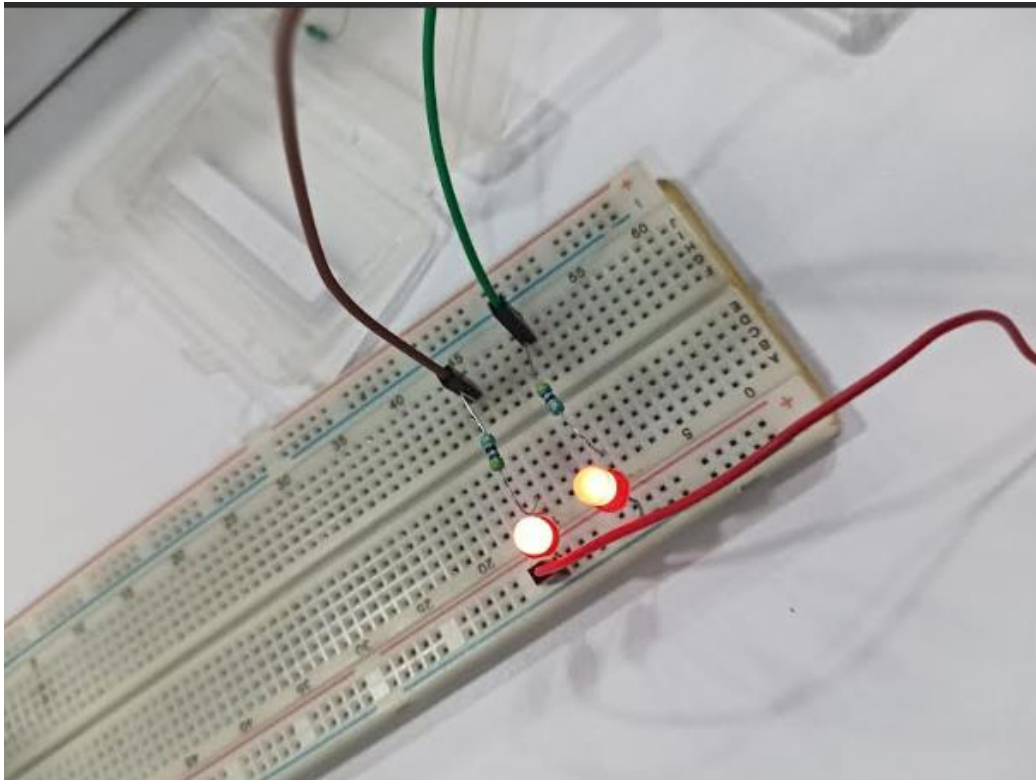
**P4.PWM with 2 LED(or with one led can be asked in practical)**

**One male to female wire on Ground pin 6**

**Second male to female wire on GPIO pin 12**

**THIRD male to female wire on GPIO pin 11**

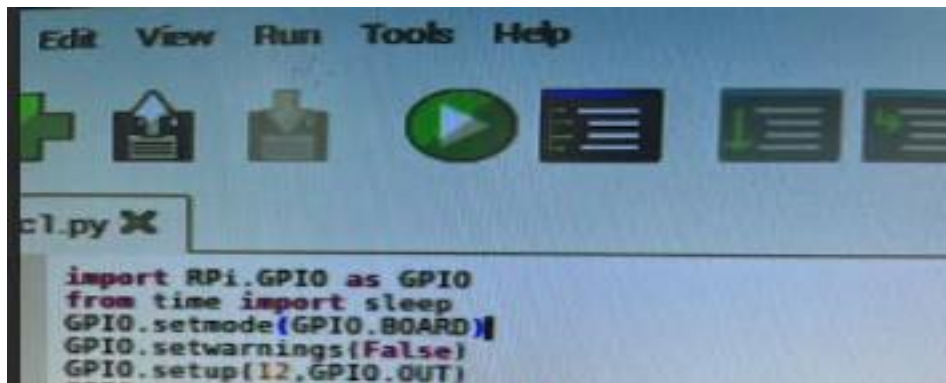
**Connection**



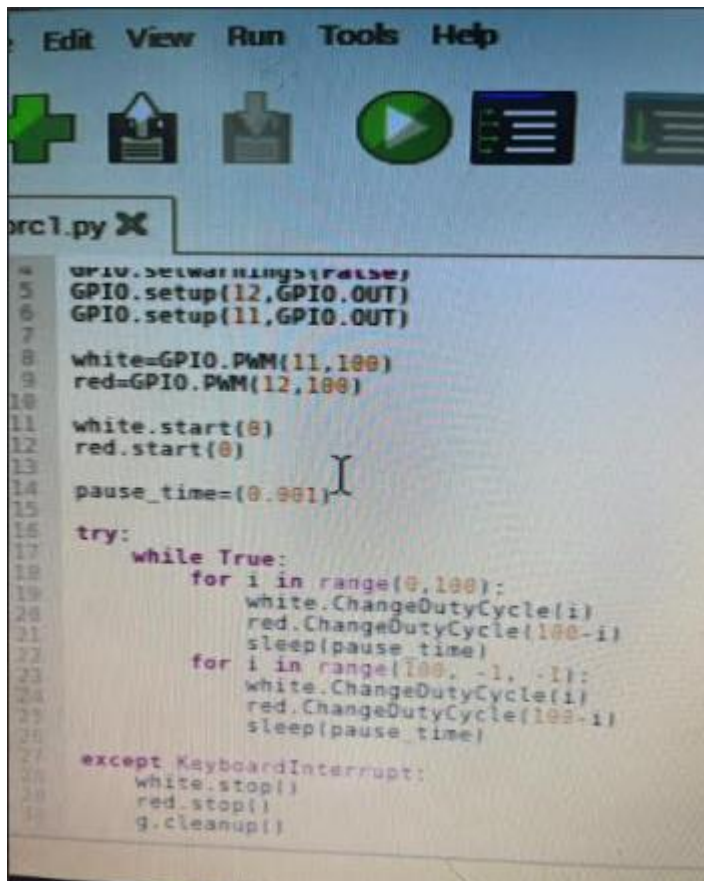
**Code:**



## Subject :IOT



```
1 import RPi.GPIO as GPIO
2 from time import sleep
3 GPIO.setmode(GPIO.BOARD)
4 GPIO.setwarnings(False)
5 GPIO.setup(12,GPIO.OUT)
```



```
1 GPIO.setwarnings(False)
2 GPIO.setup(12,GPIO.OUT)
3 GPIO.setup(11,GPIO.OUT)
4
5 white=GPIO.PWM(11,100)
6 red=GPIO.PWM(12,100)
7
8 white.start(0)
9 red.start(0)
10
11 pause_time=(0.001)
12
13 try:
14     while True:
15         for i in range(0,100):
16             white.ChangeDutyCycle(i)
17             red.ChangeDutyCycle(100-i)
18             sleep(pause_time)
19         for i in range(100,-1,-1):
20             white.ChangeDutyCycle(i)
21             red.ChangeDutyCycle(100-i)
22             sleep(pause_time)
23 except KeyboardInterrupt:
24     white.stop()
25     red.stop()
26     g.cleanup()
```



**Subject :IOT**

## **P5.on/off the LED by telegram**

**Connection**

**Bread Board**

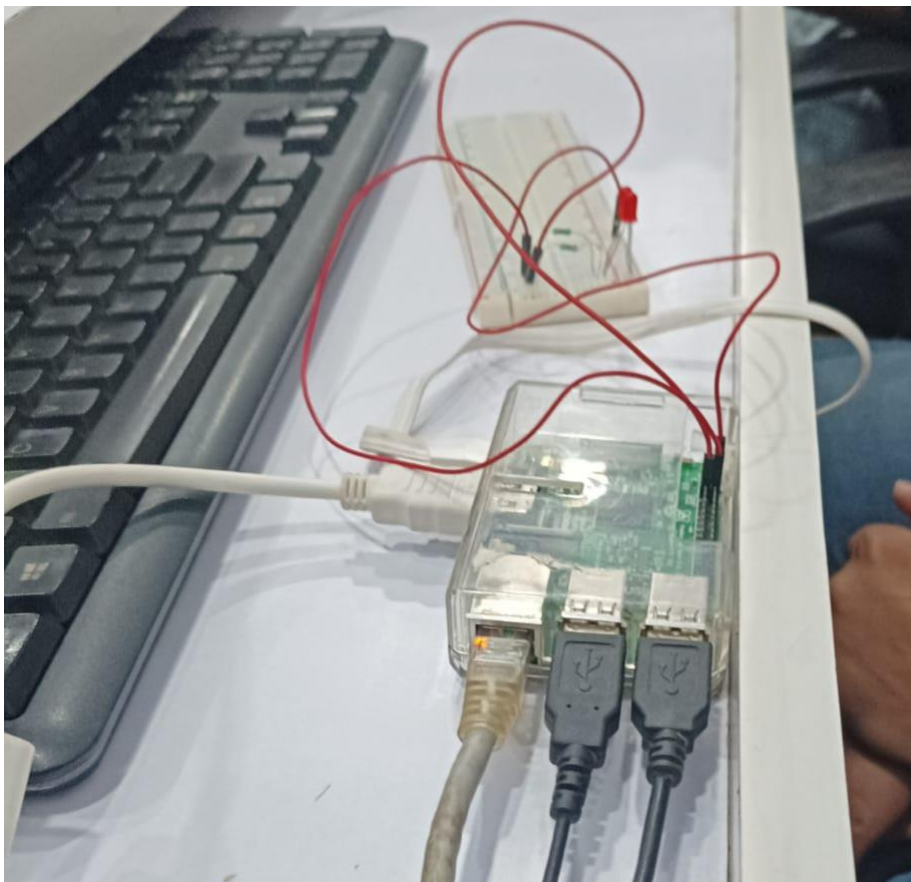
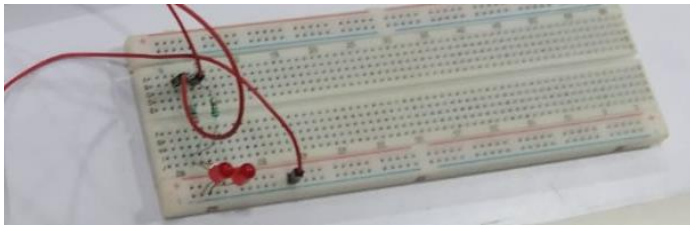
**LED,**

**register,**

**jumper wires (male to female)**

**connect the ground pin on Raspberry Pi at GPIO Pin 6**

**connect the one end of male-female wire on breadboard (near register) and second end on Pi at GPIO pin 11**



**Code:**

## Subject :IOT

**S1:**In cmd terminal install telepot

>>pip install telepot

**S2:** Go to telegram (by phone)

Search for BotFather>>create new bot name & username

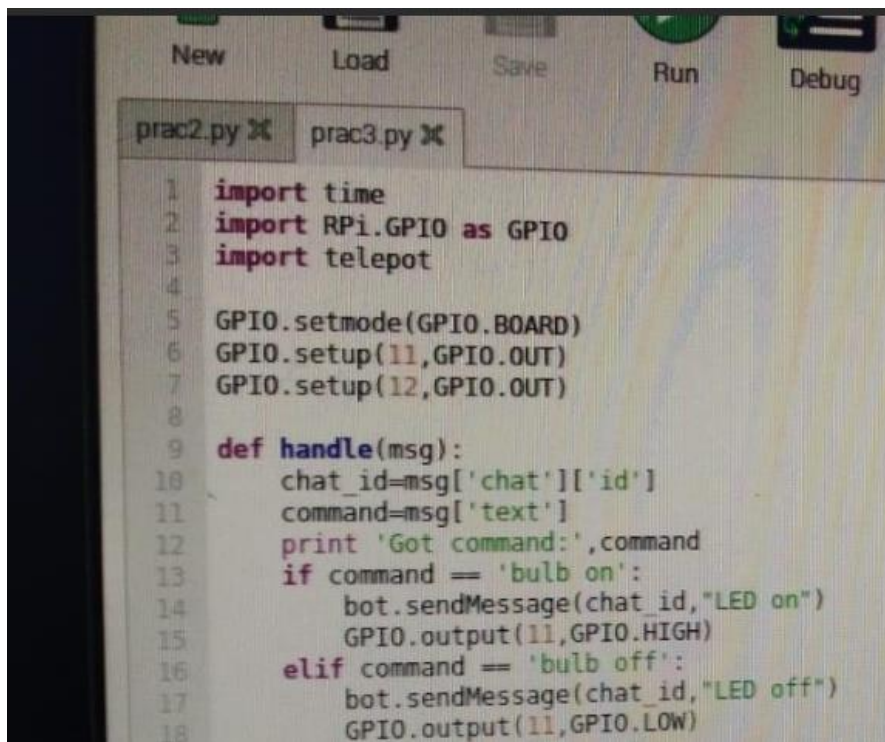
**S3:**You will get one bot token number(on BotFather telegram chat)

**S4:**Copy paste the whole number

```
17 try:
18     bot=telepot.Bot("7474473956:AAG3c7xLb8B-50Fe2IvAN_9unnowxgUANZY")
19     bot.message_loop(handle)
```

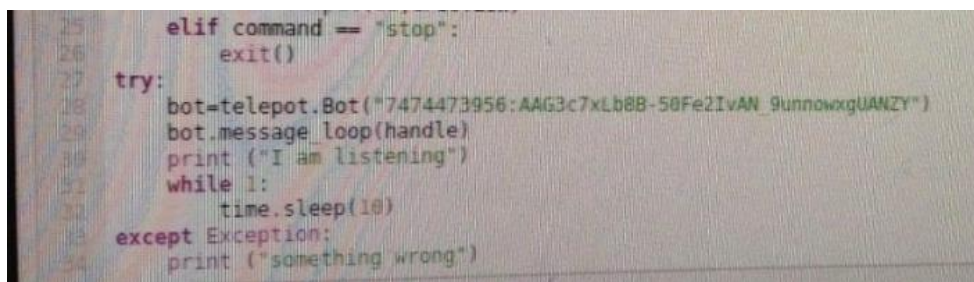
**S5:**Run the program in cmd terminal

**S6:**Search the chat of your bot username in telegram and in that give the on-off command



The screenshot shows a code editor with a menu bar (New, Load, Save, Run, Debug) and two tabs (prac2.py, prac3.py). The code in prac3.py is as follows:

```
1 import time
2 import RPi.GPIO as GPIO
3 import telepot
4
5 GPIO.setmode(GPIO.BOARD)
6 GPIO.setup(11,GPIO.OUT)
7 GPIO.setup(12,GPIO.OUT)
8
9 def handle(msg):
10     chat_id=msg['chat']['id']
11     command=msg['text']
12     print 'Got command:',command
13     if command == 'bulb on':
14         bot.sendMessage(chat_id,"LED on")
15         GPIO.output(11,GPIO.HIGH)
16     elif command == 'bulb off':
17         bot.sendMessage(chat_id,"LED off")
18         GPIO.output(11,GPIO.LOW)
```



The screenshot shows the continuation of the Python script in the code editor. The code is as follows:

```
25 elif command == "stop":
26     exit()
27 try:
28     bot=telepot.Bot("7474473956:AAG3c7xLb8B-50Fe2IvAN_9unnowxgUANZY")
29     bot.message_loop(handle)
30     print ("I am listening")
31     while 1:
32         time.sleep(10)
33 except Exception:
34     print ("something wrong")
```

## Subject :IOT

### P6. Home Automation using Telegram

Connection:

Bulb , Fan

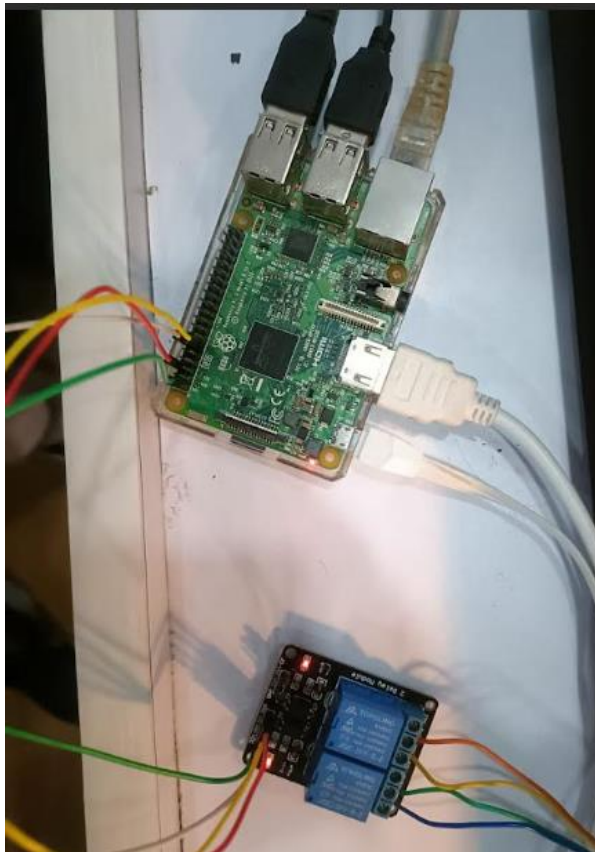
1 Relay

4 jumper wires (female to female)

Connect female to female wire at Vcc on relay to GPIO pin 2

At Ground on relay to GPIO pin 6

At Input 1 & 2 on relay to GPIO pins 11&12



Code:

**S1:**In cmd terminal install telepot

>>pip install telepot

**S2:** Go to telegram (by phone)

Search for BotFather>>create new bot name & username

## Subject :IOT

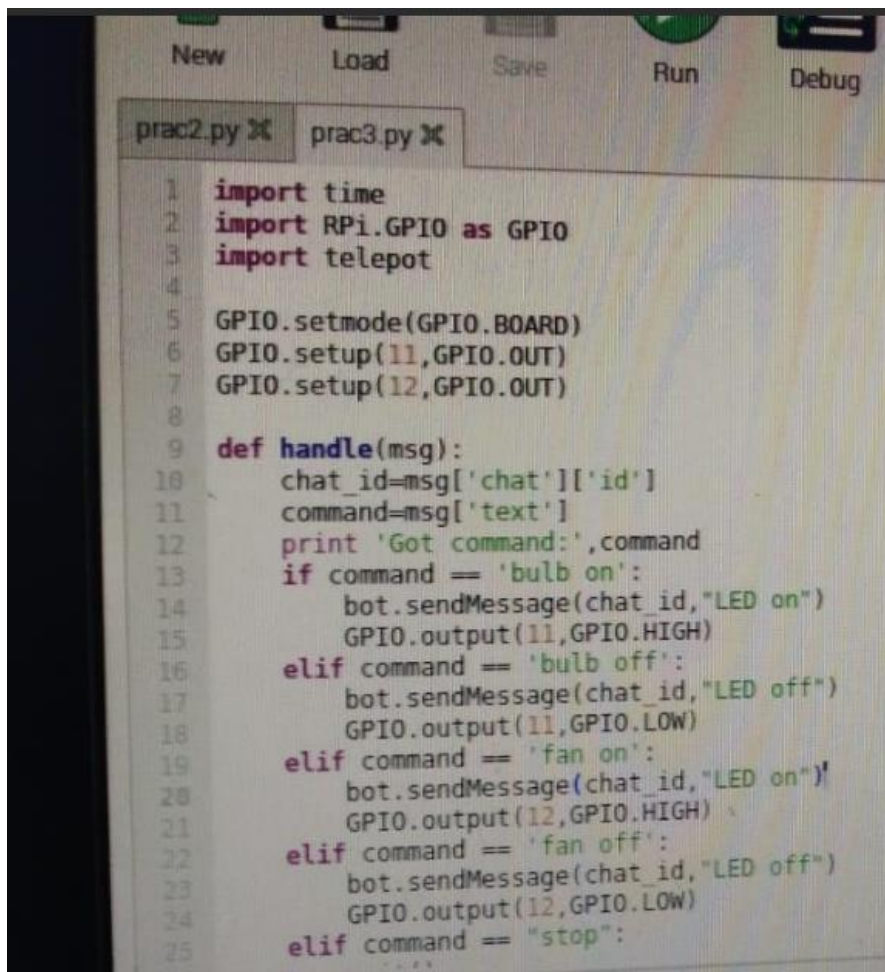
**S3:**You will get one bot token number(on BotFather telegram chat)

**S4:**Copy paste the whole number

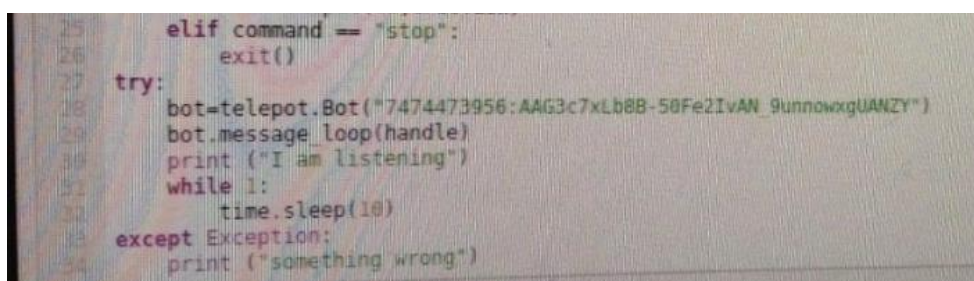
```
27 try:
28     bot=telepot.Bot("7474473956:AAG3c7xLb8B-50Fe2IvAN_9unnowxgUANZY")
29     bot.message_loop(handle)
```

**S5:**Run the program in cmd terminal

**S6:**Search the chat of your bot username in telegram and in that give the on-off command



```
New Load Save Run Debug
prac2.py X prac3.py X
1 import time
2 import RPi.GPIO as GPIO
3 import telepot
4
5 GPIO.setmode(GPIO.BOARD)
6 GPIO.setup(11,GPIO.OUT)
7 GPIO.setup(12,GPIO.OUT)
8
9 def handle(msg):
10     chat_id=msg['chat']['id']
11     command=msg['text']
12     print 'Got command:',command
13     if command == 'bulb on':
14         bot.sendMessage(chat_id,"LED on")
15         GPIO.output(11,GPIO.HIGH)
16     elif command == 'bulb off':
17         bot.sendMessage(chat_id,"LED off")
18         GPIO.output(11,GPIO.LOW)
19     elif command == 'fan on':
20         bot.sendMessage(chat_id,"LED on")
21         GPIO.output(12,GPIO.HIGH)
22     elif command == 'fan off':
23         bot.sendMessage(chat_id,"LED off")
24         GPIO.output(12,GPIO.LOW)
25     elif command == "stop":
```



```
25     elif command == "stop":
26         exit()
27 try:
28     bot=telepot.Bot("7474473956:AAG3c7xLb8B-50Fe2IvAN_9unnowxgUANZY")
29     bot.message_loop(handle)
30     print ("I am listening")
31     while 1:
32         time.sleep(10)
33 except Exception:
34     print ("something wrong")
```

**Run using cmd terminal**



## Subject :IOT

```
pi@raspberrypi:~$ cd Desktop
pi@raspberrypi:~/Desktop$ ls
prac1.py prac2.py prac3.py SVIT.py
pi@raspberrypi:~/Desktop$ python prac3.py
prac3.py:6: RuntimeWarning: This channel is already in use, continuing anyway.
Use GPIO.setwarnings(False) to disable warnings.
  GPIO.setup(11,GPIO.OUT)
I am listening
Got command: ON
Got command: on
```

Output:



**Subject :IOT**

## **P7: 4-Digit Clock**

**Connection**

**4-Digit Clock**

**4 jumper wires (female to female)**

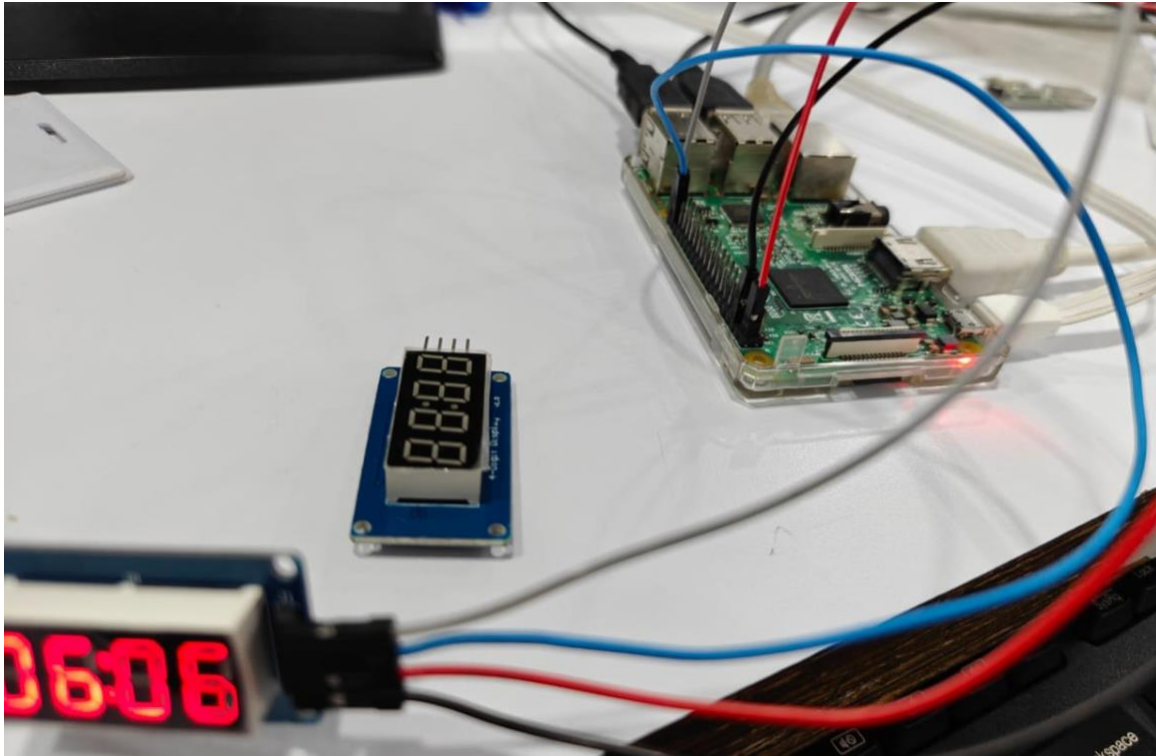
**(save tm1637 module and your code in same folder)**

**Pin connections**

<b>TM1637 Pin</b>	<b>Name (ON CLOCK)</b>	<b>Remarks</b>	<b>RPi Pin (on pi)</b>	<b>RPi Function BCM</b>
<b>1</b>	<b>GND</b>	<b>Ground</b>	<b>6</b>	<b>GND</b>
<b>2</b>	<b>VCC</b>	<b>+5V Power</b>	<b>2</b>	<b>5V</b>
<b>3</b>	<b>DIN</b>	<b>Data In</b>	<b>38</b>	<b>GPIO 20</b>
<b>4</b>	<b>CLK</b>	<b>Clock</b>	<b>40</b>	<b>GP10 21</b>



## Subject :IOT



Code:

```
from time import sleep
```

```
import tm1637
```

```
Display = tm1637.TM1637(CLK=21, DIO=20, brightness=1.0)
```

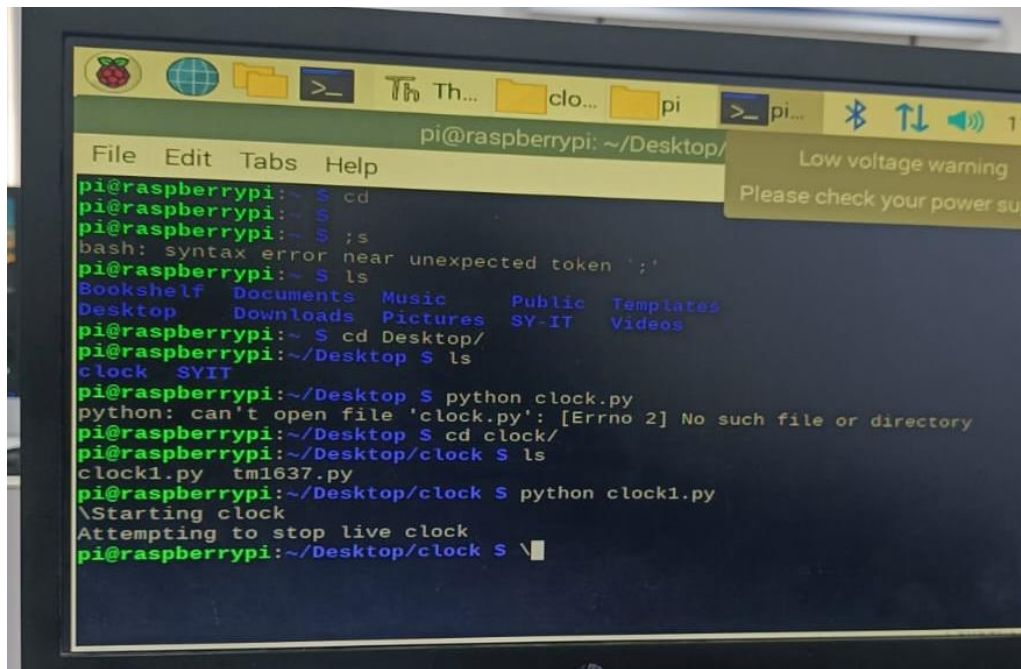
```
print ("Starting clock in the background")// (press CTRL + C to stop)
```

```
Display.StartClock()
```

```
sleep(5)
```

```
Display.StopClock()
```

## Subject :IOT



```
pi@raspberrypi: ~/Desktop/
File Edit Tabs Help
pi@raspberrypi:~$ cd
pi@raspberrypi:~$ $
pi@raspberrypi:~$ ;s
bash: syntax error near unexpected token ';'
pi@raspberrypi:~$ ls
Bookshelf  Documents  Music      Public     Templates
Desktop    Downloads  Pictures   SY-IT      Videos
pi@raspberrypi:~$ cd Desktop/
pi@raspberrypi:~/Desktop$ ls
clock  SYIT
pi@raspberrypi:~/Desktop$ python clock.py
python: can't open file 'clock.py': [Errno 2] No such file or directory
pi@raspberrypi:~/Desktop$ cd clock/
pi@raspberrypi:~/Desktop/clock$ ls
clock1.py  tm1637.py
pi@raspberrypi:~/Desktop/clock$ python clock1.py
\Starting clock
Attempting to stop live clock
pi@raspberrypi:~/Desktop/clock$ \
```



# Subject :IOT

## P8:RFID

### Connection

(Requirements required)

USB to TTL converter

RFID cards and Reader module

4 jumper wires (female to female)

Connect the 1 wire to GND on reader module and GND on TTL

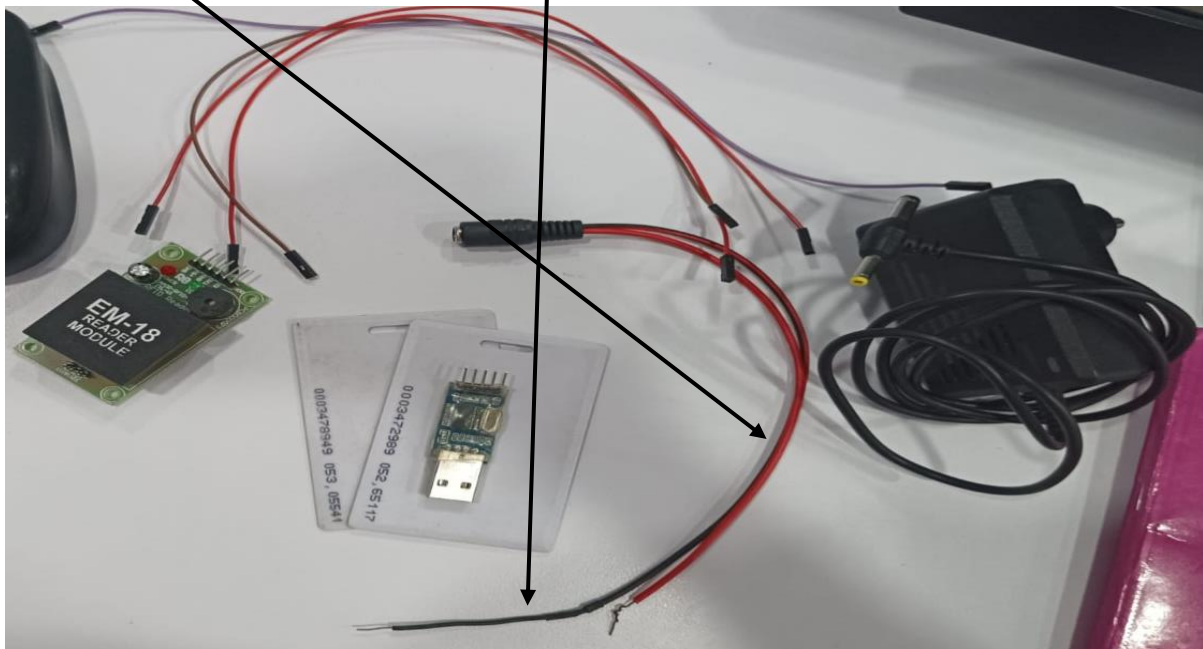
Connect the 2 wire to TX on reader module and RX on TTL

Connect the 3 wire to +ve(upper side ,from where E start on reader module) of Reader module to red wire of connection

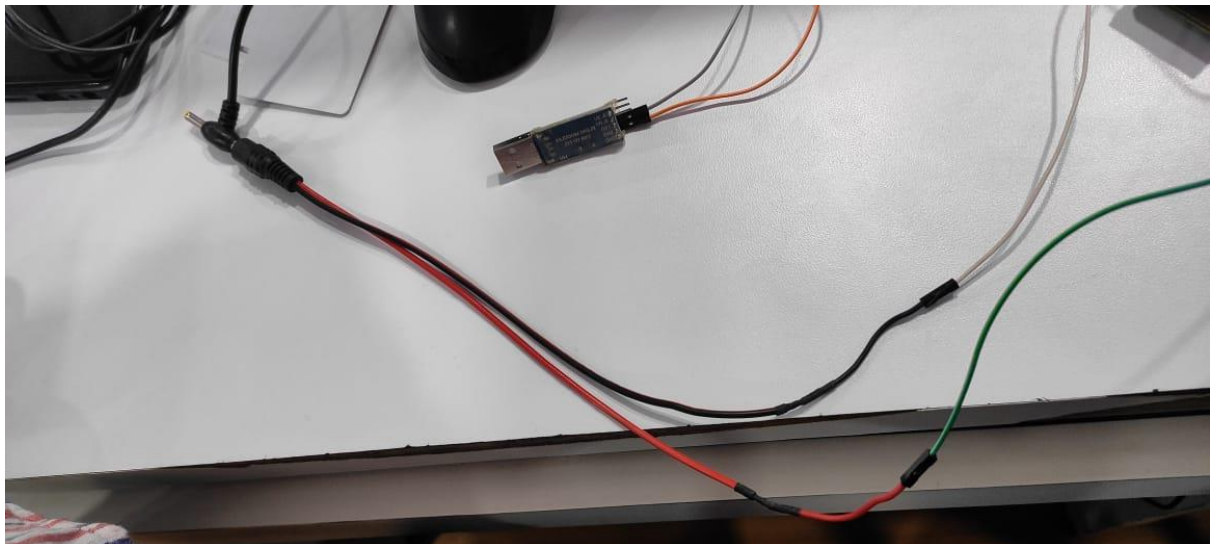
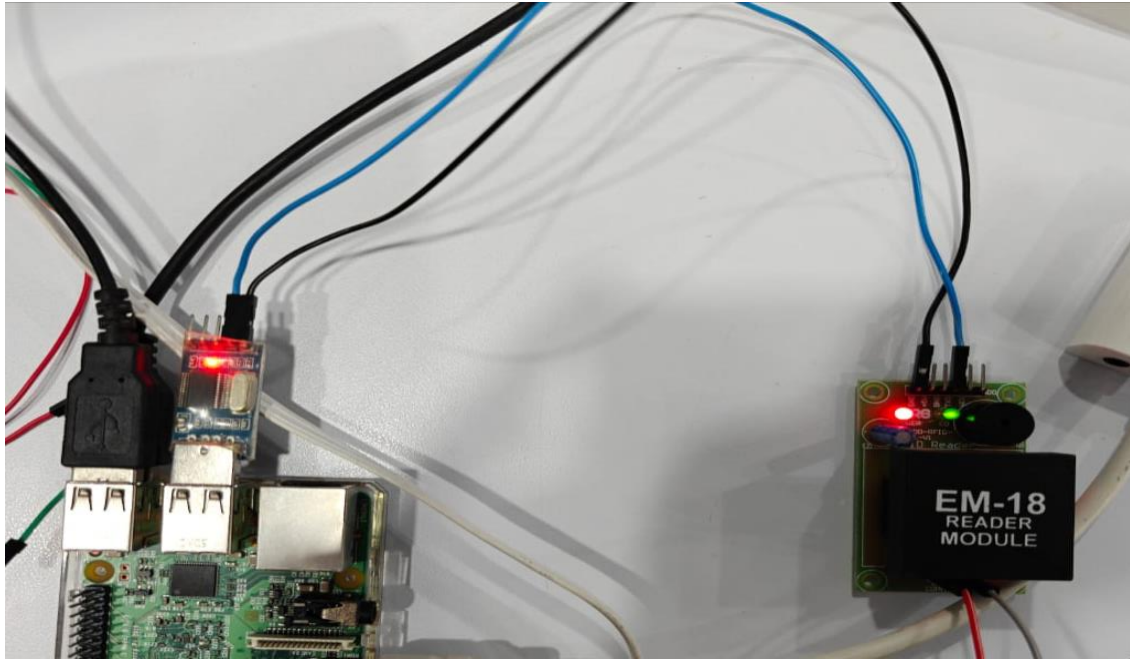
Connect the 4 wire to -ve(down side) of Reader module to Black wire of connection

Connect the connection wire to adapter

Red wire is +ve and black wire is -ve



## Subject :IOT



### Code:

```
import time
import serial
data=serial.Serial(port='/dev/ttyUSB0',baudrate=9600)
try:
while 1:
print ("Place the card")
x=data.read(12)
print(x)
```

## Subject :IOT

```
if x==(“1C00377ACC9D”)
```

```
print (“Access Granted”)
```

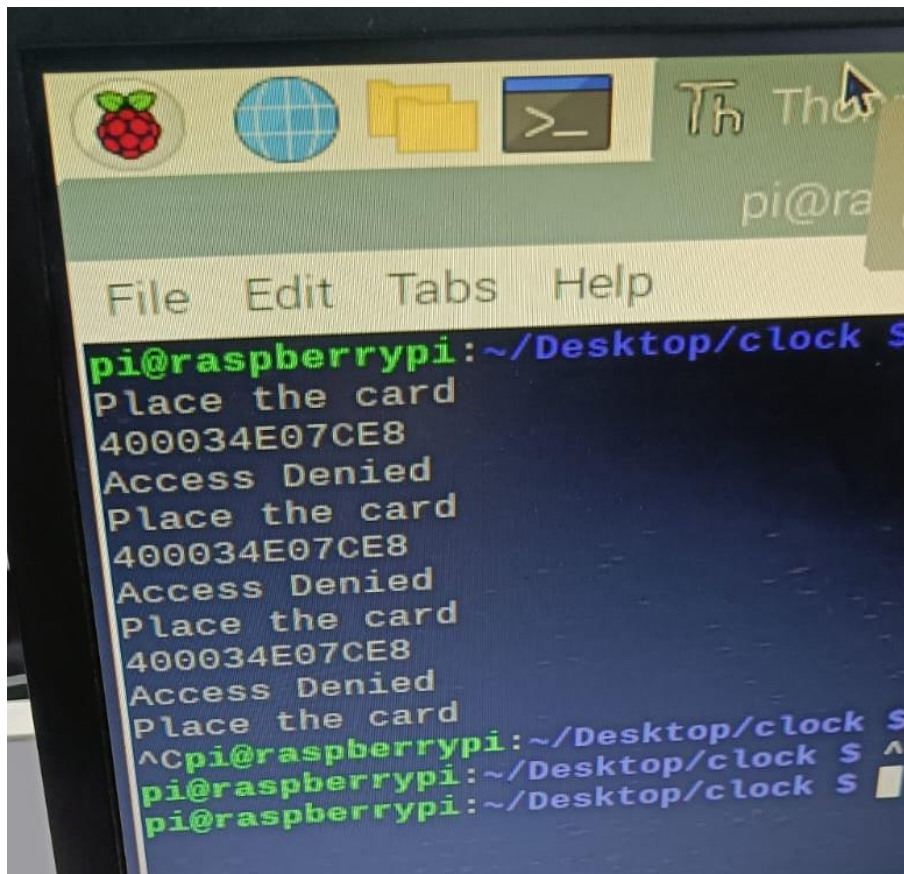
```
else:
```

```
print (“Access Denied”)
```

```
except KeyboardInterrupt:
```

```
data.close()
```

[Note: now place the RFID cards one by one over RFID module to check the output]



## Subject :IOT

### P9:Led matrix

#### Connection

5 jumper wires

#### Led Matrix

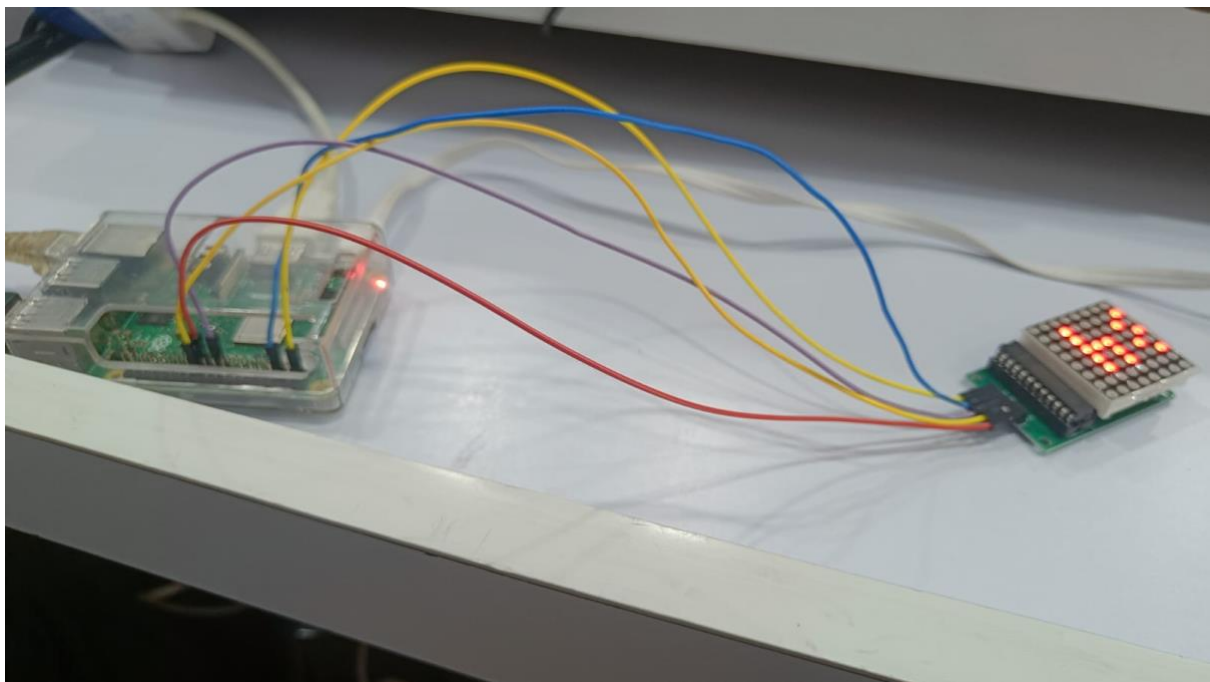
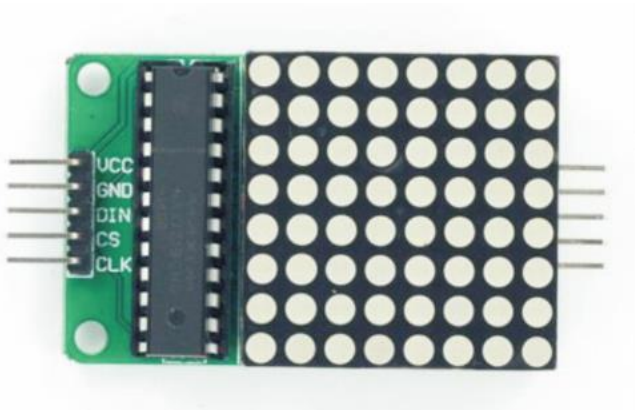
Connect the VCC pin to GPIO Pin 2 on raspberry pi.

Connect the Gnd pin to GPIO Pin6 on raspberry pi.

Connect the DIN pin to GPIO Pin 19 on raspberry pi.

Connect the CS pin to GPIO Pin 24 on raspberry pi.

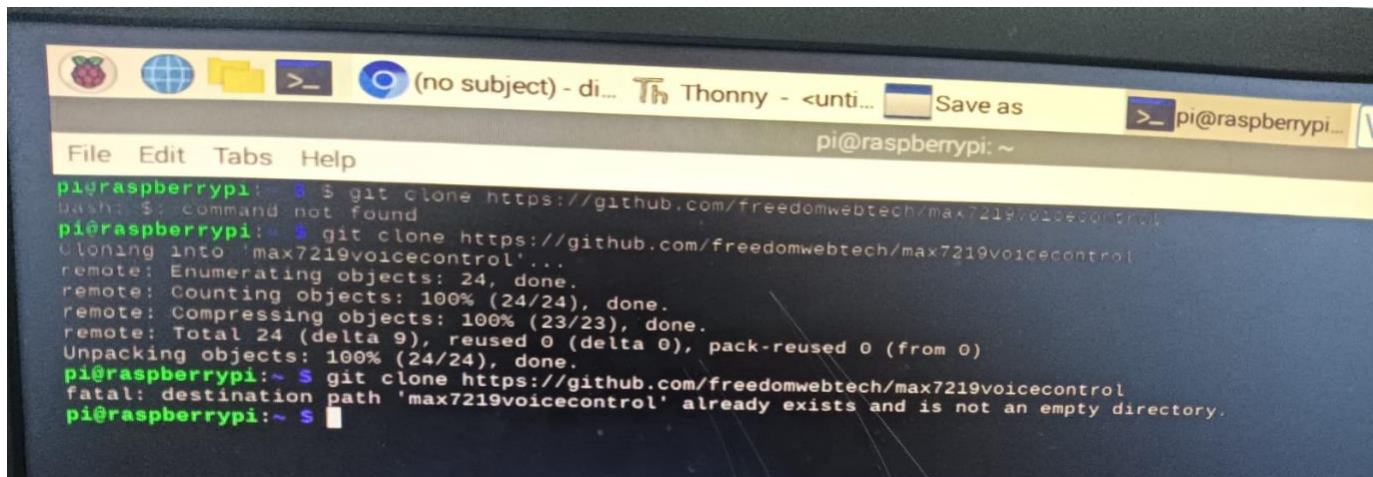
Connect the CLK Pin to GPIO Pin 23 of raspberry pi.



Install all library given in ss



## Subject :IOT

A screenshot of a Raspberry Pi terminal window. The window has a title bar with icons for Raspberry Pi, a globe, a folder, a terminal, and a web browser. The title text is "(no subject) - di... Thonny - <unti... Save as". The terminal shows the following commands and output:

```
pi@raspberrypi:~$ git clone https://github.com/freedomwebtech/max7219voicecontrol
bash: $: command not found
pi@raspberrypi:~$ git clone https://github.com/freedomwebtech/max7219voicecontrol
Cloning into 'max7219voicecontrol'...
remote: Enumerating objects: 24, done.
remote: Counting objects: 100% (24/24), done.
remote: Compressing objects: 100% (23/23), done.
remote: Total 24 (delta 9), reused 0 (delta 0), pack-reused 0 (from 0)
Unpacking objects: 100% (24/24), done.
pi@raspberrypi:~$ git clone https://github.com/freedomwebtech/max7219voicecontrol
fatal: destination path 'max7219voicecontrol' already exists and is not an empty directory.
pi@raspberrypi:~$
```

### Code:

```
from luma.led_matrix.device import max7219
from luma.core.interface.serial import spi, noop
from luma.core.render import canvas
from luma.core.virtual import viewport
from luma.core.legacy import text, show_message
from luma.core.legacy.font import proportional, CP437_FONT, TINY_FONT,
SINCLAIR_FONT, LCD_FONT
from datetime import datetime
import time

serial = spi(port=0, device=0, gpio=noop())

device = max7219(serial, cascaded=1, block_orientation=-90,
blocks_arranged_in_reverse_order=True)

device.contrast(16)

def test():
    now = datetime.now()
    # dt1_string = now.strftime("%H:%M:%S")
    dt1_string = now.strftime("%I:%M:%S")
    with canvas(device) as draw:
        text(draw, (3, 1), dt1_string, fill="white", font=proportional(TINY_FONT))
```

## **Subject :IOT**

```
# show_message(device, "Hello EDKITS",  
fill="red",font=(CP437_FONT),scroll_delay=0.08)  
  
while True:  
  
test()
```

## **Subject :IOT**

### **P10: Motor**

#### Connection

#### Stepper Motor

#### Stepper motor driver

#### Jumper wires

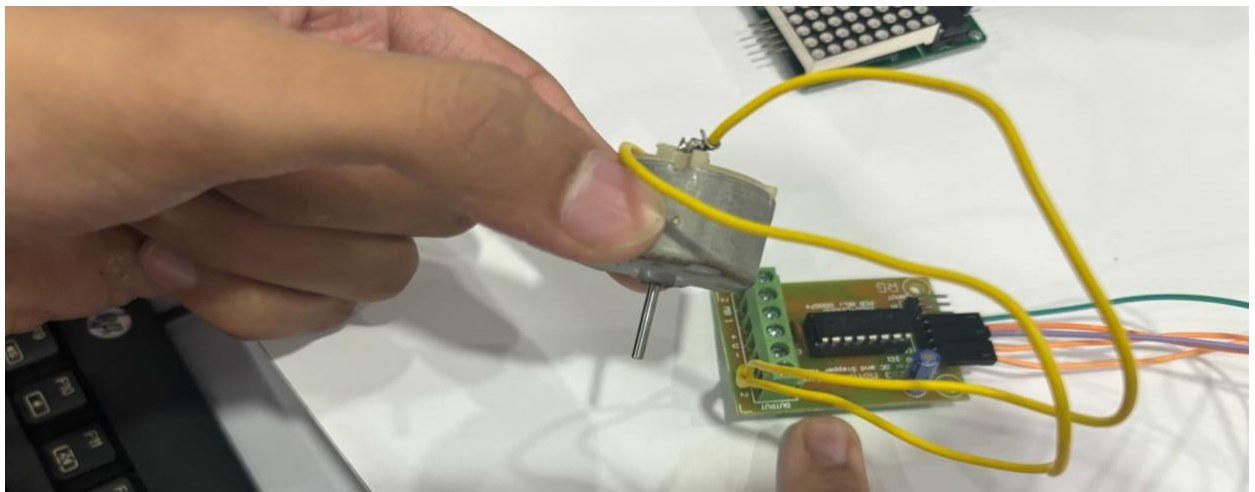
Connect Vcc of Stepper Motor driver -GPIO Pin 4 on raspberry pi

Connect Gnd of Stepper Motor driver -GPIO Pin 6 on raspberry pi

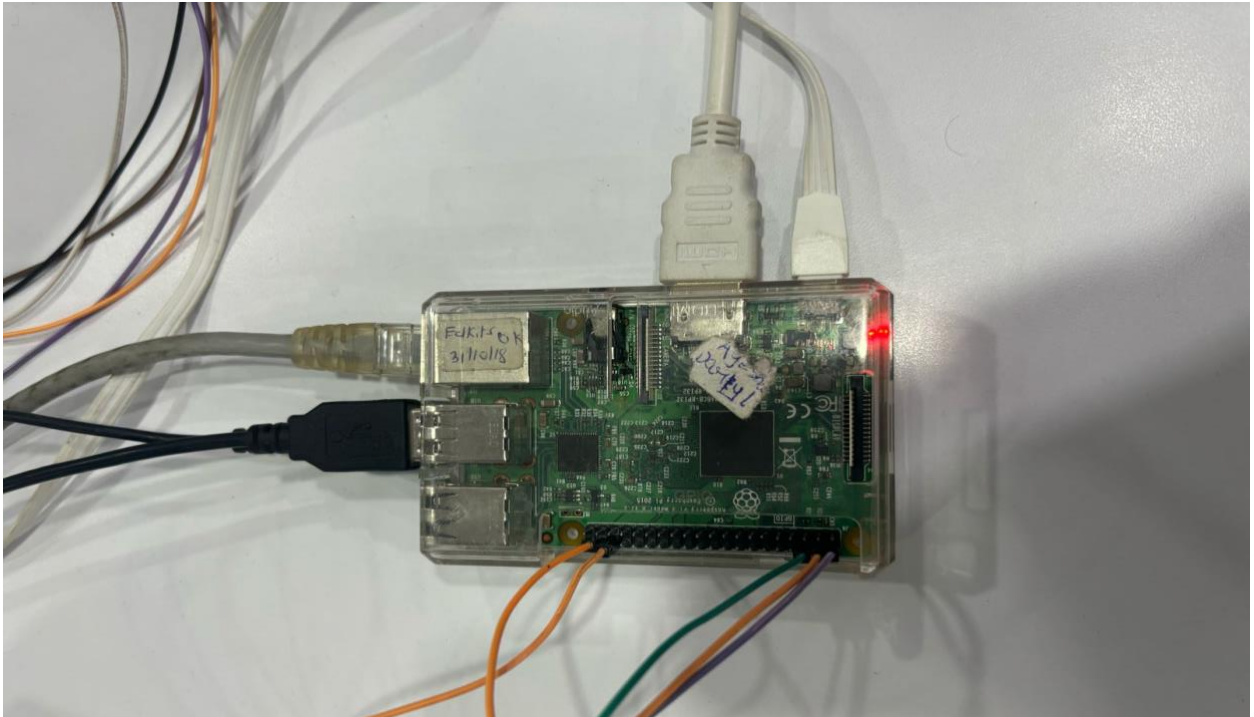
Connect A1 of Stepper Motor driver -GPIO Pin 36 on raspberry pi

Connect A2 of Stepper Motor driver -GPIO Pin 38 on raspberry pi

Connect EN-A of Stepper Motor driver -GPIO Pin 2 on raspberry pi



## Subject :IOT



### Code:

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

# Pins for Motor Driver Inputs
Motor1A = 36
Motor1B = 38

GPIO.setmode(GPIO.BOARD)    # BOARD Numbering
GPIO.setup(Motor1A,GPIO.OUT) # All pins as Outputs
GPIO.setup(Motor1B,GPIO.OUT)
#GPIO.setup(Motor1E,GPIO.OUT)

try:
    while True:
        # Going forwards
        GPIO.output(Motor1A,GPIO.HIGH)
        GPIO.output(Motor1B,GPIO.LOW)
        sleep(5)
```



## **Subject :IOT**

```
# Going backwards
```

```
GPIO.output(Motor1A,GPIO.LOW)
```

```
GPIO.output(Motor1B,GPIO.HIGH)
```

```
sleep(5)
```

finally:

```
#reset the GPIO Pins
```

```
GPIO.cleanup()
```