Practical No :- 01

Title :- starting Rasphian OS, Familiarising with Respberry pi, components and interface, connecting to enternet, Monitor, USB.

Requirements :-

1) Respberry pi
2) power supply
3) USB keyword
4) USB mouse
5) Micro SD card
6) Micro SD USB card reader
7) A monitor or TV that support HDMI or composite video
8) An HDMI cable or composite video cable
9) An ethernet cable cor wifi dongle

procedure:

Downloading NOOBS.

Using NOOBS is the easiest way to intall Respbian on your SD Card. To get hold of a copy of NOOBS.

you should see a box a link to the NOOBS Files. Click on the link.

Rasphian is the Foundations. Official supported operating system. Download it here, or use NOOBS or easy installer for Rasphian and more.

sit back and relax. It will take a while, but Raspian will Install.

When Raspbian has been installed, click ok and your Raspberry Pi will restart and Resphian will then boot up.

FOR EDUCATIONAL USE

Goto your Downloads Folder and find the zip file you downloaded.

Extract the files and keep the resulting Explore/Finder window open.

Copying the files.

Now open another Explore/Finder window and navigate to the SD Card. It's best to position the two windows side by side.

Select all the files from the NOOBS folder and drag them anto the SD card.

Eject the SD card.

Booting from NOOBS

Once the files have been copied over, insert the micro SD card into your Respberry pi, and plug the pi into powser source.

you will be offered a choice when the installer has loaded. you should check the box Raspbian, and then click install.

click yes at the warning dialog, and then

FOR EDUCATIONAL USE

The simplest option is to download the zip archive of the files.

NOOBS Lite contains the same operating system installer without Raspbian pre-loaded. It provides the same operating system selection menu allowing Raspbian and other images to downloaded and installed.

Formatting the SD card.

Visit the SD Association's website and download SD Formatter 4.0 for windows or Mac.

Follow the instructions to install the software.

Insert your SD card into computer or laptop's SD card reader and make a note of the drive letter allocated to it. eg F:/

In SD Formatter, select the drive letter for your SD card, and format it.

Extracting xNOOBS from zip archive

Next, you will need to extract the files from the NOOBS zip achive you downloaded from the Resphemy pi website.

FOR EDUCATIONAL USE

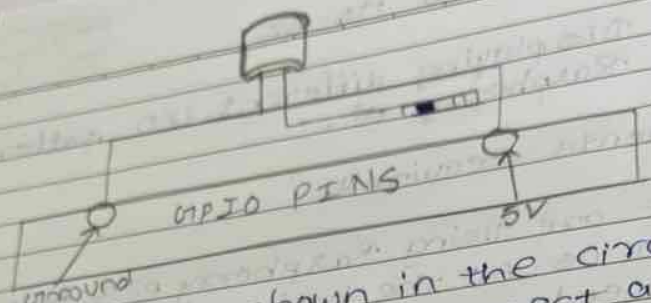
4. Save this Files as blinky on the desktop.

5. After that write the program for blinky as given below and execute the program by clicking on 'RUN' an 'DEBUG' option

If the program has no errors in it, you will see a ">>" which means the program is executed successfully.

We will see the python program code for LED Blinking.

Code :-
```
import RPi, GPIO as IO
import time
IO. Set mode (IO, BOARD)
PIN40
IO. setup (40, IO, OUT)
IO. Output (40, 1)
time. sleep (1)
IO. Cleanup ()
time. sleep (1)
```



procedure: As shown in the circuit diagram we are going to connect an LED between PIN40 (GPIO21) and PIN39 (Ground) As said earlier, we cannot draw more that 15mA from any one of these pins, so to limit the current we are connecting a 220 Ω or 1K Ω register in series with the LED.

Working Explanation.

Since we have everything ready, turn ON your pi and go to the desktop.

1. On the desktop, go to start Menu and choose For the python 3

2. After that python will run and you will see a window.

3. After that click on New File in File Menu, you will see a window open.

Practical No - 02

Title :- Displaying different LED patterns with Raspberry pi.

Components Required.

Here we are Using Raspberry pi 3 Model B+ with Raspbian Os. All the basic Hardware and software requirements are previously discussed. you can look it up in the Raspberry pi introduction. Other than that we need.

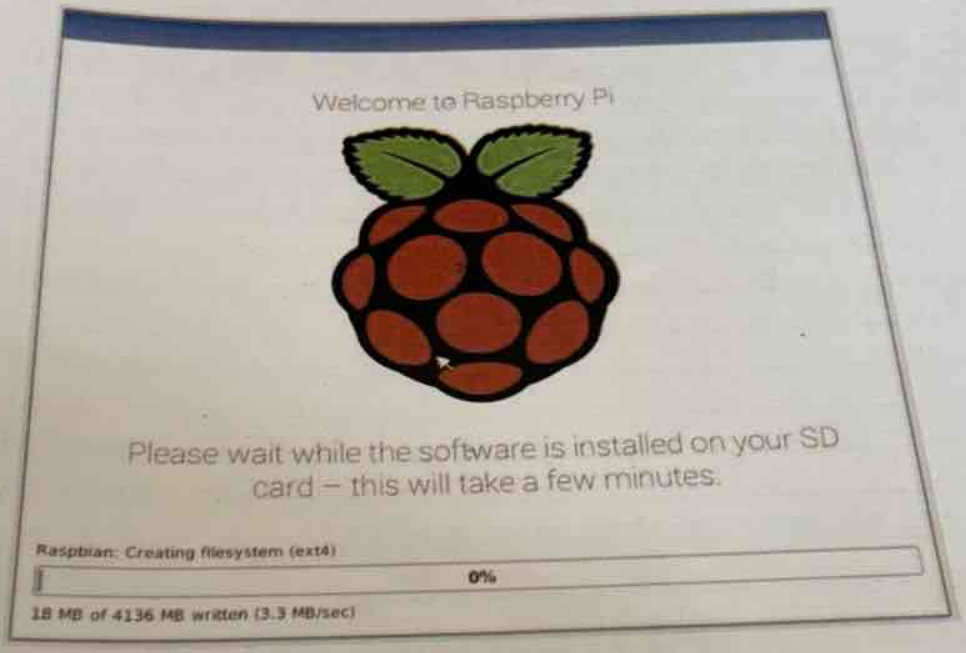
Connecting pins

220 Ω or 1k registor
LED
Bread Board
Circuit diagram

Respberry Pi A+/B+ and Raspberry Pi 2 physical pin numbers.

→ GPI
→ Ground
→ 3.3V
→ 5V
→ ID, EEPROM. Advanced Use Only!

OUTPUT:-


Welcome to Raspberry Pi
Please wait while the software is installed on your SD card – this will take a few minutes.
Raspbian: Creating filesystem (ext4)
0%
18 MB of 4136 MB written (3.3 MB/sec)

c. Tab to Finish

7. Intall the software:

$ cd /home/pi
$ sudo apt update
$ sudo apt install git
$ git clone http://github.com/timewarizenger/rasberry pi - examples /tree/master/actor-led-7 segment 4 numbers.

8. power down your pi for setting up the hardware.
$ shutdown now.
After the LED goes off unplug the power

PIN	DESC
CLK	clock
DIO	Data In
GND	Ground
5V	+5 volts

. Once you have a module temporarily wired up and checked your witing

c. Tab to Finish

7. Intall the software:

$ cd /home/pi
$ sudo apt update
$ sudo apt install git
$ git clone http://github.com/timewaizengg/rasberry pi-examples/tree/master/actor-led-7 segment 4 numbers.

8. power down your pi for setting up the hardware.
$ shutdown now.
After the LED goes off unplug the power

PIN	DESC
CLK	clock
DIO	Data In
GND	Ground
5V	+5 volts

1. Once you have a module temporarily wired up and checked your wiring

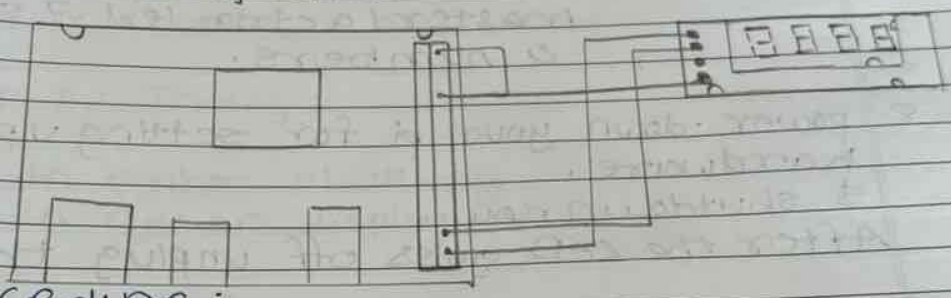
Practical No - 03

Title : Displaying Time over 4- Digit 7- Segment Display using Raspberry pi.

Requirement :

1) Raspberry pi
2) 4- Digit 7- Segment Display

Circuit diagrams.

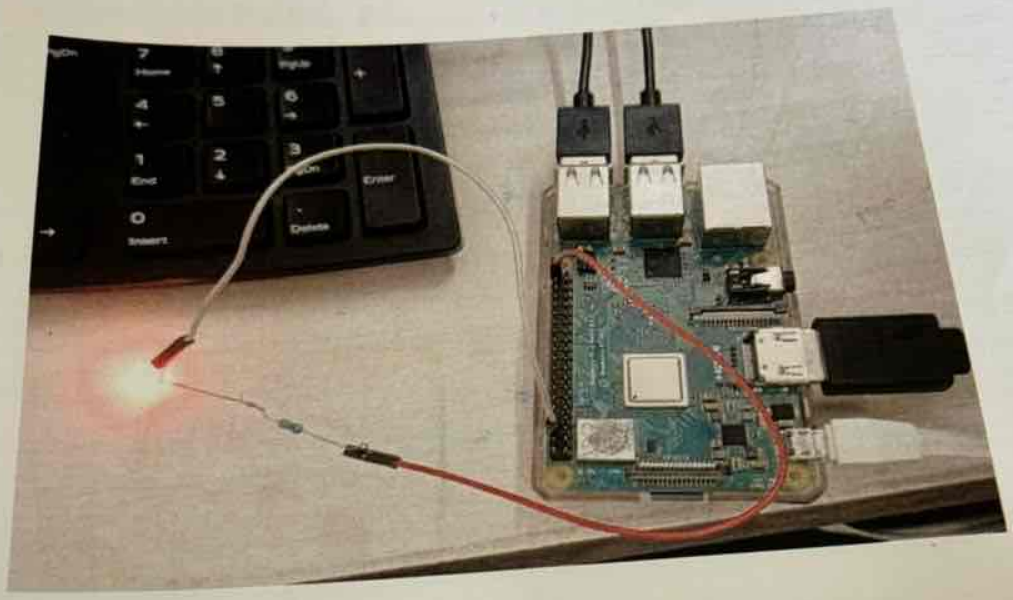


procedure :

Update your Raspbian
$ sudo apt update
$ sudo apt upgrade

6. Configure the raspberry:
$ sudo raspi - config
a. change User password
b. Localization ~~oper~~ option → change Timezon
Select your Local Time Zone.

OUTPUT:-



Practical No - 05

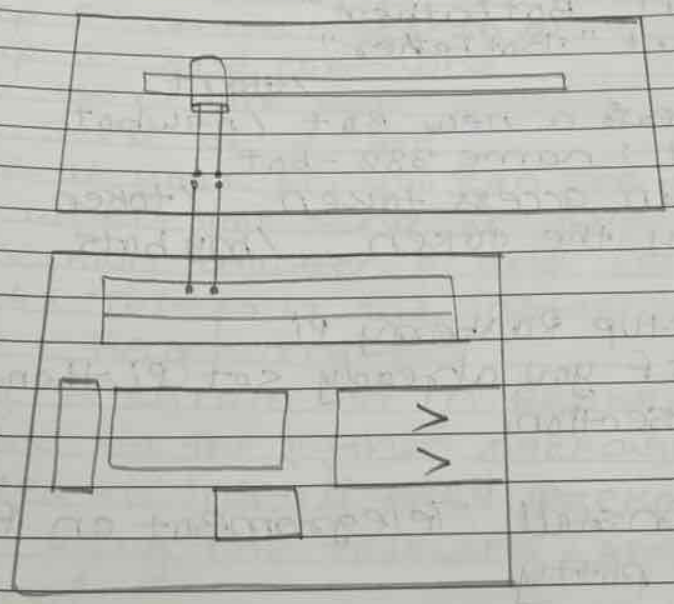
Title : Controlling Respberry Pi with WhatApp.

Requirements :

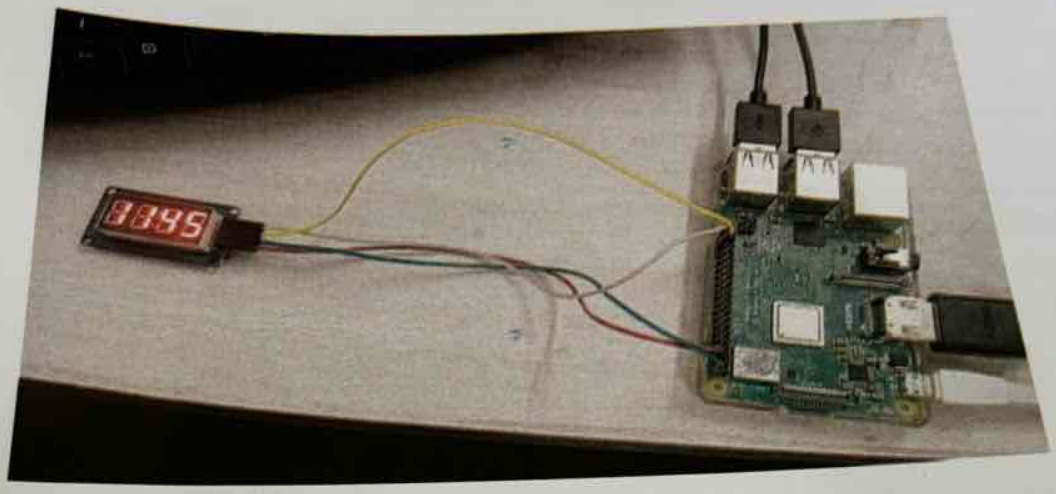
1) Respberry Pi 3 Model B x 1
2) LED (generic) x 1
3) Jumper wires (generic) x 1
4) Breadboard (generic) x 1

Circuit diagram :

All set, now time to connect the Pi and LED



OUTPUT:-



```
number= IP.split(*.*)
loop=3
 while.loops>0 :
    for number in numbers:
        Display.showInt(number)
    loops-=1
    Display.clear()
    time.sleep(2)

Print("done")
Display.cleanup()
```

2. power up the Raspberry pi
The red LED on the module should light but there will be NO Display yet.

3. SSH into your pi again like previously
$ cd actor-led-7-segment-4number
$ sudo python tm1637.py
$ python -V (capital "V")
python 2.7x
$ sudo shutdown now

Step 6: Testing:

$ sudo python clock.py
$ sudo python display ip.py

Code:
```
import time
import tm1637
import socket
Display = tm1637.TM1637(CLK=21, DIO=20,
  brightness=0.3)
s = socket.socket(socket.AF_INET, socket.
  SOCK_DGRAM)
IP = s.getsocktname()[0]
s.close()
print("IP of Raspi is")
print(IP)
number(IP)
```

```
bot = telepot.Bot('5652606111:AAGTn,NJOWPUEOCK
    SmomGzMhsesIDOEqKAI')
bot.message_loop(handle)
print ('I am listening...')
while 1:
    try:
        time.sleep (10)
    except keyboard Interup:
        print ('In program interupted')
        GPIO.clean ()
        exit ()

except:
print ('other error or exception accured!')
GPIO.cleanup ()
```

Sudo apt-get install python-pip
Sudo pip3 install telepot
Once Telepot is imported into Resberry we can use this package in out python program to communicate with out Telegram Bot.

Code: TeleLed.py

```
import sys
import time
import telepot
import RPI.GPIO as GPIO
def on(pin):
    GPIO.output(pin, GPIO.HIGH)
    return
def off(pin):
    GPIO.output(pin, GPIO.Low)
    return
GPIO.setmode(GPIO.BOARD)
GPIO.setmode(11, GPIO.OUT)
def handle(msg):
    chat_id = msg['chat']['id']
    command = msg['text']
    print('got command: %s' % command)
    if comman == 'on':
        bot.sendMessage(chat_id, on())
    elif command == 'off':
        bot.sendMessage(chat_id, off())
```

we are in
3.3 Install "python package Index"
Sudo apt-get install python-pip
Note: Make sure pi has internet access
3.4 Install "telpot"
Sudo pip install telepot
4.1 Clone the git
get clone --recursive https://gethub.com/vy
shing/tg.git && cd tg
4.2 Paste your Bot Token here
bot = telepot.Bot ('Bot Token')
python Teleled.py
Step 5: Connect LED to pi
Step 6: Send command
6.1. Start our Bot
6.2. Send 'on' & 'off'
Look at your pi, you can see the LED on and OFF when you send "on" and "off" to our bot.
Connection between pi and LED
Step 4: Teleport for installing Telegram on Raspberry Pi.
Using Telegram Bot in Resberry Pi is made possible by the python package called Telepot. We need to install this package on Raspberry pi by using the following commands on LX terminal.

procedure : story
Learn how to use the Telegram Bot, host a Telegram Bot on your Raspberry Pi, and use the messaging app to interact with your device.

Step 1 : Open Telegram app in your system or mobile.

1.1 open Telegram app in your system or mobile Download from here : Telegram

1.2 start "Bot Father"
1.3 open "Bot Father"
1.4 start "Bot Father"
/start

1.5 create a new Bot /newbot
Bot : name 332 - bot
1.6 obtain access token /token
Copy the token /my bots

Step2: Setup Rasberry Pi
Note : If you already set Pi then skip the section.

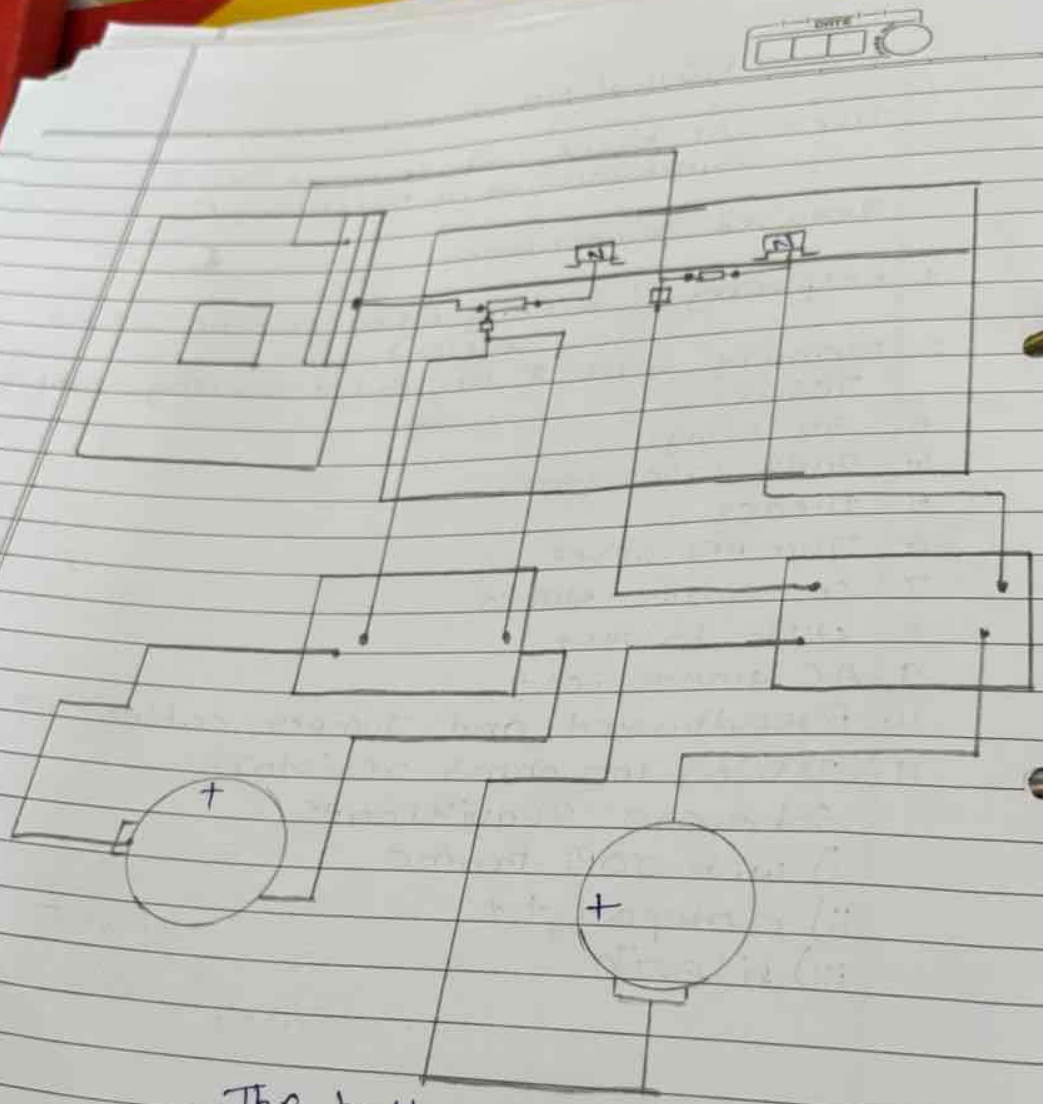
Step 3 : Install TelegramBot on Raspberry Pi
3.1 Open putty
3.2. Connect pi via SSH / VHC
enter pi ip address
pi username.

The Javascript code:

The first piece of code we will write in that of Javascript, It a simple script to communicate with the webIOPI service.

```
webiopi().ready(function(){
  webbiopi().setfunction(17,"out");
  webiopi().setfunction(18,"out");
  webiopi().setfunction(22,"out");
  webiopi().setfunction(23,"out");
  var content, button;
  content = $("#content");
  button = webiopi().createGPIOButton
  (17," Relay1");
  content.append(button);
  button = webiopi().createGPIOButton(18,
  " Relay2");
  content.append(button);
  button = webiopi().createGPIOButton(22,
  "Relay 3");
  content.append(button);
  button = webiopi().createGPIOButton(23,"
  Relay 4");
  content.append(button);
  });
```

FOR EDUCATIONAL USE



The battery repss the main supply while x marks the ~~application~~ appliance to be controlled with orange color.

FOR EDUCATIONAL USE

Practical No : 07

Title : IOT based web controlled Home Automation using Raspberry Pi.

Required components :

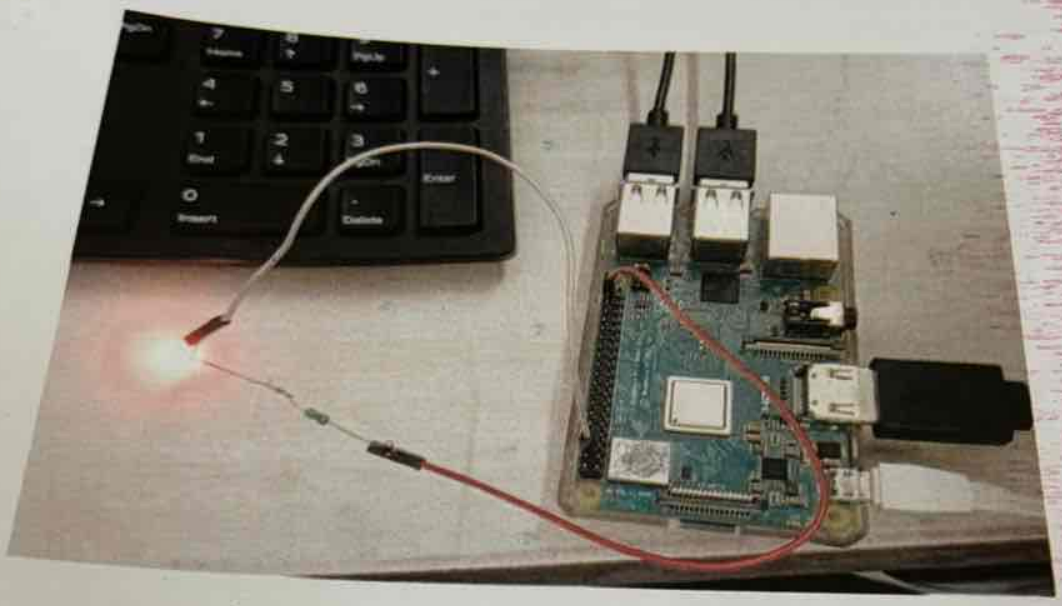
1. Raspberry Pi 3 (Any other version will be benice)
2. Memory card 8 or 16 GB running Raspbian Jessie.
3. 5V Relays
4. 2n22 transistors
5. Diodes
6. Jumper wires
7. connection Blocks
8. LEDs to test
9. AC lamp to Test
10. Breedboard and Jumper cables
11. 220 or 100 ohms resistor

Software Requirements

i) web IOPi frame
ii) notepad ++
iii) filezila

FOR EDUCATIONAL USE

OUTPUT:-



```
<!DOCTYPE html PUBLIC "-//W3C//DTD
HTML 4.01 Transitional //EN"
"http://www.w3.org/TR/html4/loose.
dtd">
<html>
<head>
<meta http-equiv="content-type"
content="text/html"; charset=UTF-8">
<meta name="mobile-web-app-
capable" content="yes">
<meta name="viewport" content="
height=device-height, width=device-
width, user-scable="no"/>
<title> smart Home</title>
<script type="text/Javascript",
src="/webiopi.js"></script>
<script type="text/Javascript",
scr="<script/smarthome.JS"></script>
<link rel="stylesheet" type="text/CSS"
href="/styles/smarthome.CSS">
<link rel="shortcut icon" sizes=
"196x196" href="/img/smart.png"/>
</head>
<body>
  <div id="content" align="center"
  </div>
  <p align="center">Push button;
  receive bacon</p>
</body></html>
```

FOR EDUCATIONAL USE

procedure:

-To update the raspberry pi below commands and then reboot the RPi;

sudo apt-get update
sudo apt-get upgrade
sudo reboot.

with this done, install the webIO pi framework. Make sure you are in home directory using:

cd ~

wget to get the file from their source forge page:

wget http://sourceforge.net/projects/webiopi/files/webIOPI-0.7.1.tar.gz

when download webIOPI-0.7.1.tar.gz
cd webIOPI.0-7.1/

Then we can run the setup installation for the webIOPI using:
sudo ./setup.sh

FOR EDUCATIONAL USE

Reboot your pi
Sudo reboot

Test webIOPi Installation:

Run the command:
Sudo webiopi -d -c /etc/webiopi/config

The web browser of you computer connected to the raspberry pi to http://raspberrypi.mshome.net:8000. The system will prompt ~~mshome.net~~ your for username and password.

Building the web Application for Raspberry Pi Home Authomation.

Make sure you are in home directory using then create the folder, go into the created folder and create html folder in the directory.

```
cd~
mKdir webapp
cd webapp
mKdir html
```

Create a folder for script, CSS and images inside the html folder.

FOR EDUCATIONAL USE

The CSS code:

```
body:
    background-color: #ffff;
    background-image ("/img/smart-.png");
    background-repeat: no-repeat;
    background-position: center;
    background-size: cover;
}

button:
    display: block;
    position: relative;
    margin: 10px;
    padding: 0 10px;
    text-align: center;
    width: 130px;
    height: 40px;
    border-radius: 30px;
}
```

HTML code:

The html code pulls everything together, javascript and the style sheet.

Practical No. 08

Title: Pi Camera Module Interface with Raspberry Pi using Python.

Requirements:
1) Pi Camera Module (V1.3)
2) Raspberry Pi

circuit diagram.

procedure:
Connect Pi Camera to CSI interface of Rasberry Pi board show above:

Now, we can use Pi camera for capturing images and videos using Raspberry Pi

Before using Pi camera, we need to enable camera for its working.

For enabling camera in Raspberry Pi Configuration using following Command.

sudo raspi - config

then select interfacing options in which select Camera option to enable its functionality reboot Raspberry pi.
No we can access Camera on Raspberrypi.
Now we can capture images video using pi camera on Raspberry Pi.

Code:
```
import picamera
from time import sleep
Camera = picamera.PiCamera()
camera.resolution = (1024, 768)
Camera.~~resolution~~ brightness = 60
Camera.start-preview()
camera.annotate-text = "Hi pi user".
sleep(5)
camera.capture("image1.Jpeg")
Camera.stop-preview()
```

webIOpi server edits for Home Automation

To edit the configuration run the following with root permission:
sudo nano/etc/webiopi/config

Then if your folder is setup like mine, point your doc-root to the path your project file:

doc-root: /home/pi/webapp/html

Lastly run the webIOpi service by issuing below command:

sudo/etc/init.d/webapp start

The server status can be checked using
sudo/etc/init-d/webapp. status

If, setup webIOpi to run at boot, use;
sudo update-rc.d webiopi defaults

If you want to reverse and stop it from running a boot, use;

sudo update-rc.d webiopi remove

FOR EDUCATIONAL USE

Thonny - /home/pi/Des...

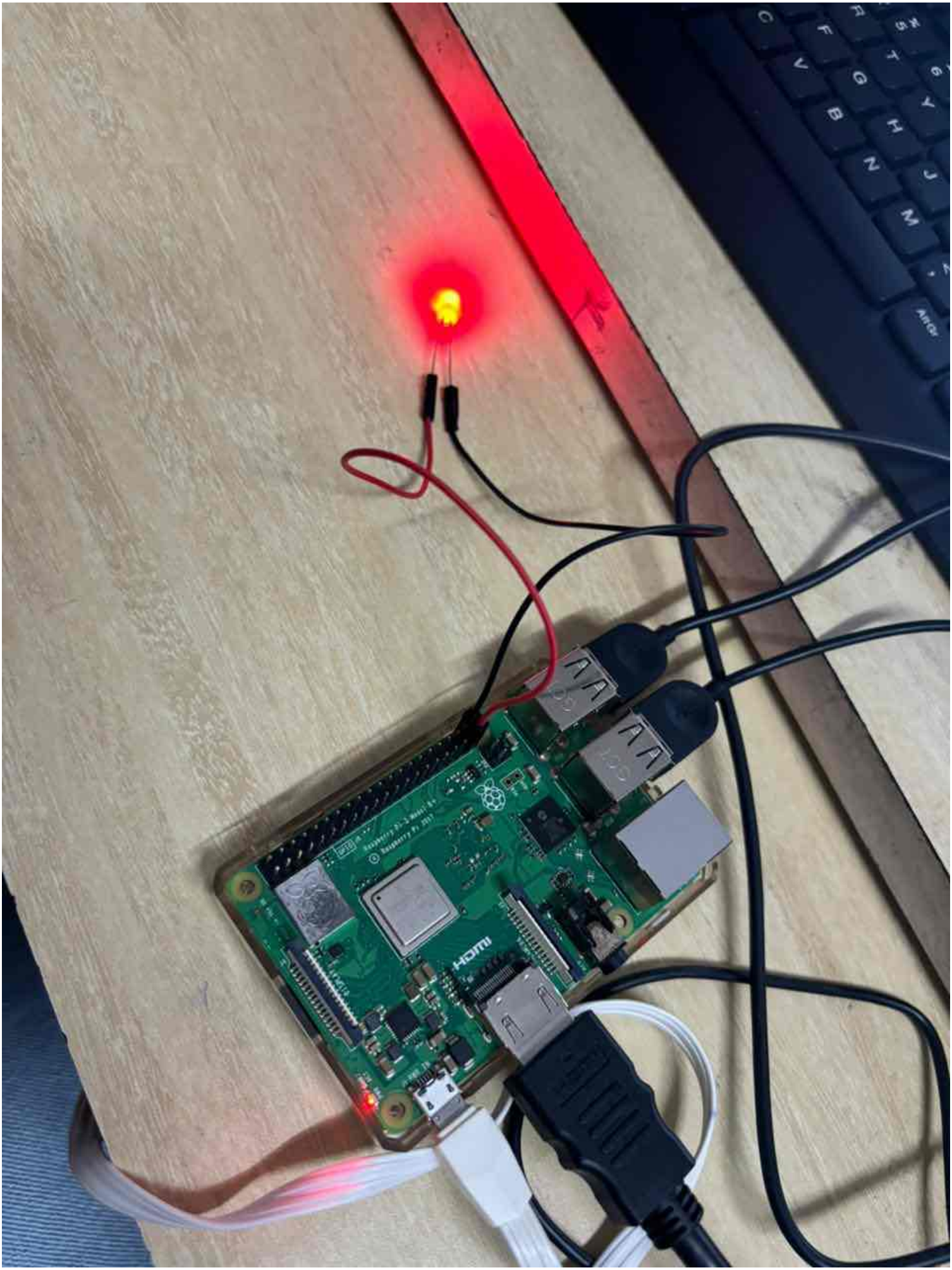
File Edit View Run Tools Help

clock.py

```python
#!/usr/bin/env python
# -*- coding: utf-8 -*-

from time import sleep
import tm1637

try:
    import thread
except ImportError:
    import _thread as thread

# Initialize the clock (GND, VCC=3.3V, Example Pins are DIO-20 and CLK21)
Display = tm1637.TM1637(CLK=21, DIO=20, brightness=1.0)

try:
    print ("Starting clock in the background (press CTRL + C to stop):");
    Display.StartClock(military_time=False)
    print ("Continue Python script and tweak Display!");
    sleep(5)
    Display.ShowDoublepoint(False)
    sleep(5)
    loops = 3
    while loops > 0:
        for i in range(0, 10):
            Display.SetBrightness(i / 10.0)
            sleep(0.5)
        loops -= 1
    Display.StopClock()
    thread.interrupt_main()
except KeyboardInterrupt:
    print ("Properly closing the clock and open GPIO pins");
    Display.cleanup()
```

Shell

```
Attempting to stop live clock
Properly closing the clock and open GPIO pins
Attempting to stop live clock
>>>
```


HDMI

Thonny - /home/pi/Des...

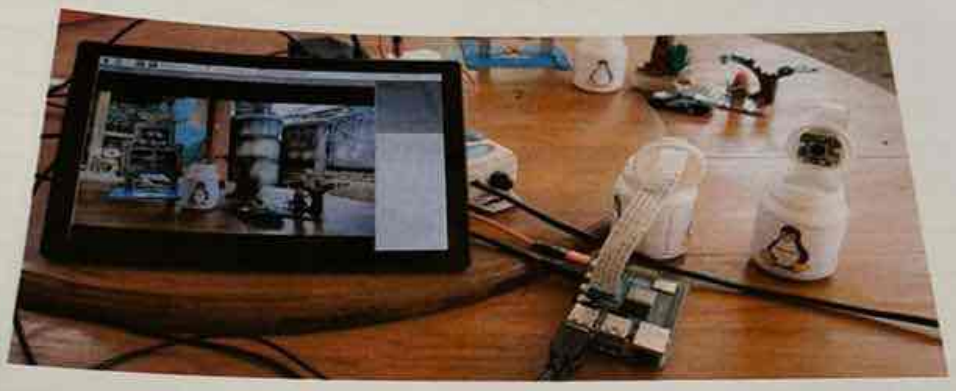
File Edit View Run Tools Help

Run current script

led40.py | ledFor40.py

```
import RPi.GPIO as IO          # calling header file for GPIO's of PI
import time                          # calling for time to provide delays i

IO.setwarnings(False)
IO.setmode (IO.BOARD)        # programming the GPIO by BOARD pin numbers, GPIO21
IO.setup(40,IO.OUT)            # initialize digital pin40 as an output.
IO.output(40,1)                  # turn the LED on (making the voltage leve
time.sleep(1)                     # sleep for a second
IO.output(40,0)                  # turn the LED off (making all the output
time.sleep(1)                    #sleep for a second

#loop is executed second time
IO.setmode (IO.BOARD)
IO.setup(40,IO.OUT)
IO.output(40,1)
time.sleep(1)
IO.cleanup()
time.sleep(1)

#loop is executed third time
IO.setmode (IO.BOARD)
IO.setup(40,IO.OUT)
IO.output(40,1)
time.sleep(1)
IO.cleanup()
time.sleep(1)
```

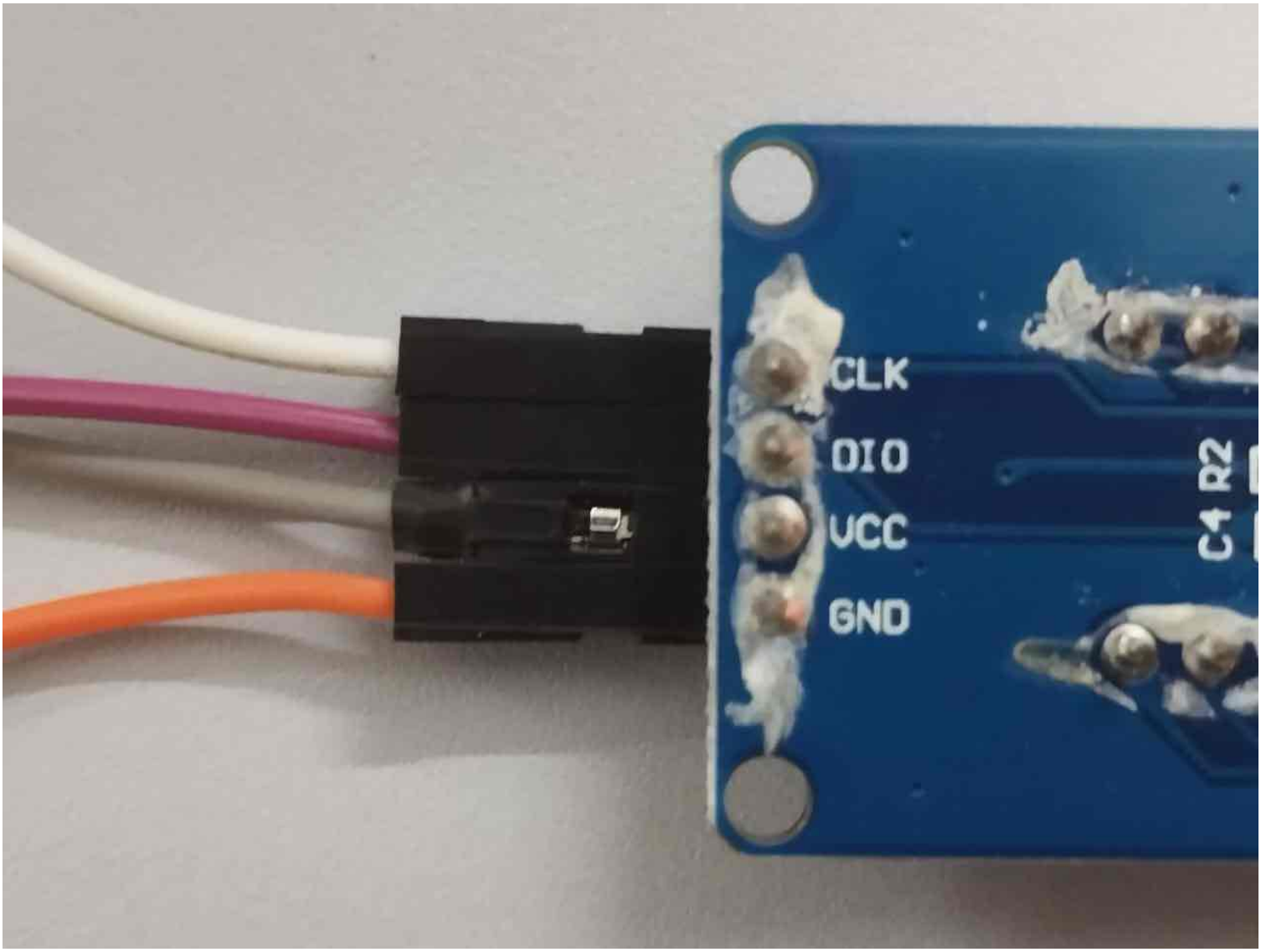
OUTPUT:-



5V	2
GND	6
CLK	40
DIO	38

See the GPIO Diagrams farther down to find the pin layouts.

The second photo shows two displays temporarily wired to a


CLK
DIO
VCC
GND
C4 R2

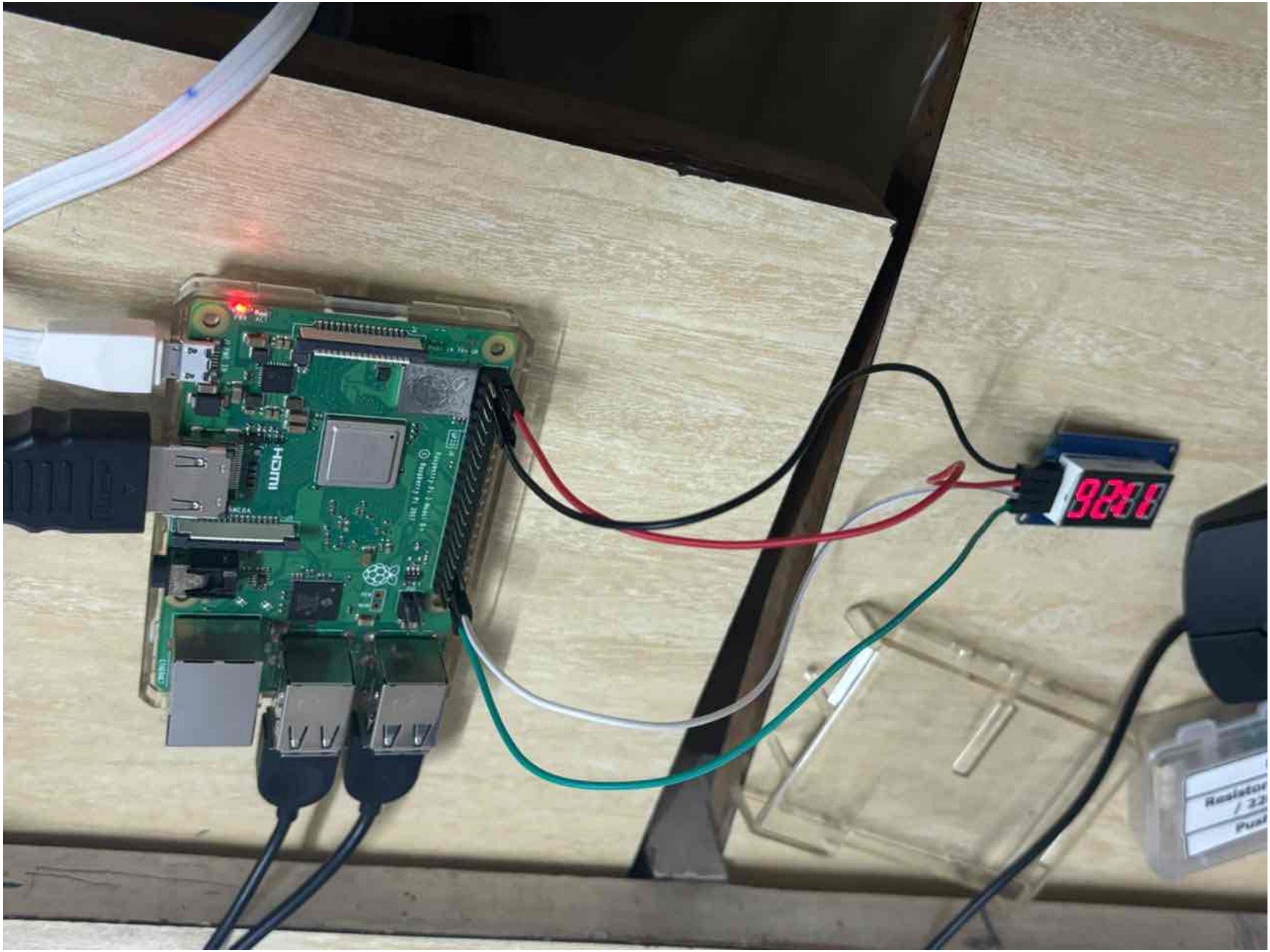
File Edit View Run Tools Help

displayIP.py

```python
#!/usr/bin/env python
# -*- coding: utf-8 -*-

import time
import tm1637
import socket

# Initialize the display (GND, VCC=3.3V, Example Pins are DIO=20 and CLK=21)
Display = tm1637.TM1637(CLK=21, DIO=20, brightness=0.3)

s = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
s.connect(("gmail.com", 80))
IP = s.getsockname()[0]

s.close()

print ("IP of RasPi is")
print (IP)

# Separate on comma,
numbers = IP.split(".")

# Loop and print each city name,
loops = 3
while loops > 0:
    for number in numbers:
        Display.ShowInt(number)
        time.sleep(2)
    loops -= 1
    Display.Clear()
    time.sleep(2)

print('done.')
Display.cleanup()
```

Shell

```
  IN,setup(self, channel, GPIO.OUT)
/home/pi/Desktop/Rpi_oct_2021/4x7_led/tm1637.py:115: RuntimeWarning: This channel is already in u
  IO.setup(self._DataPin, IO.OUT)
IP of RasPi is
192.168.0.205
done.
Attempting to stop live clock
No clock to close

>>>
```


HDMI

File Edit View Run Tools Help

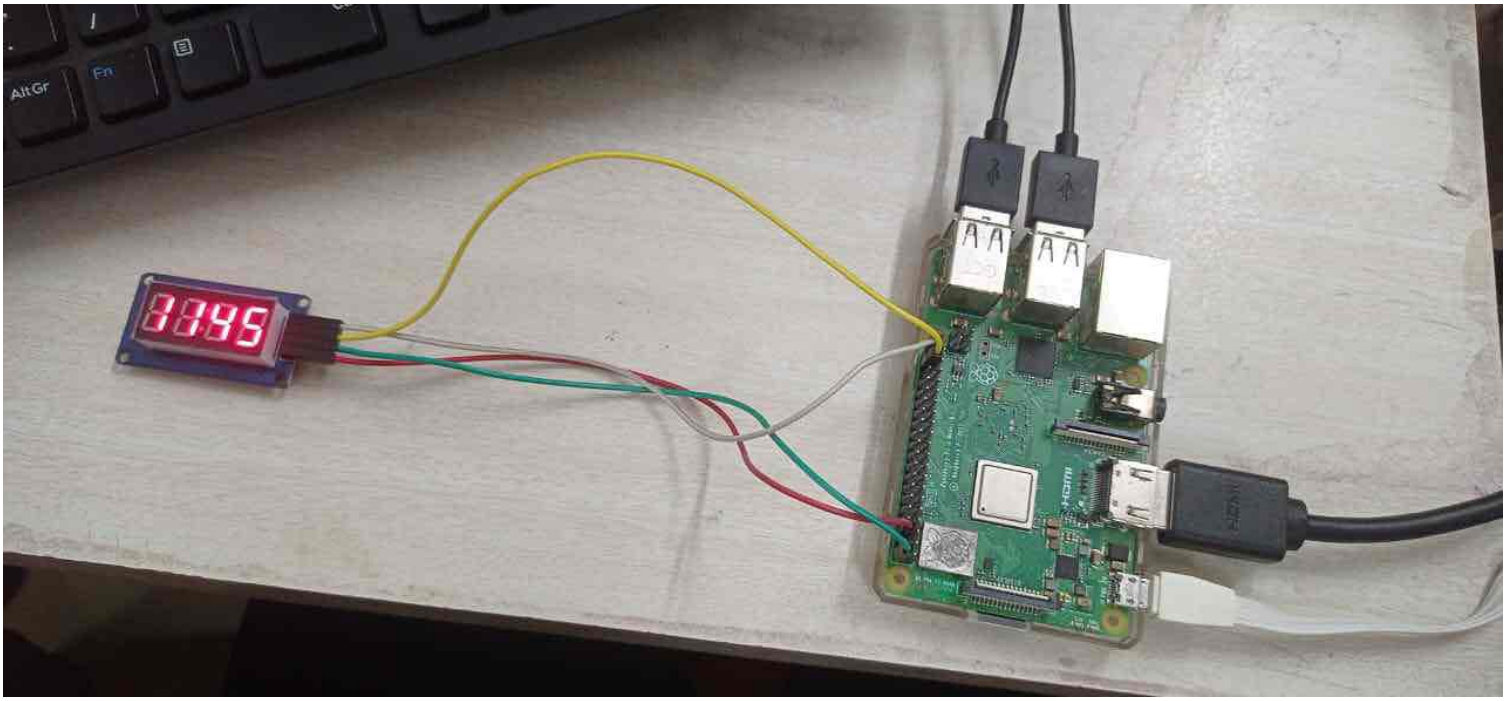
Run current script

LED.py | LED2.py | clock.py | <untitled> | display.py

```
#!/usr/bin/env python
# -*- coding: utf-8 -*-

from time import sleep
import tm1637

try:
    import thread
except ImportError:
    import _thread as thread

# Initialize the clock (GND, VCC=3.3V, Example Pins are DIO=20 and CLK=21)
Display = tm1637.TM1637(CLK=21, DIO=20, brightness=1.0)

try:
    print ("Starting clock in the background (press CTRL + C to stop):");
    Display.StartClock(military_time=False)
    print ("Continue Python script and tweak Display!");
    sleep(5)
    Display.ShowDoublepoint(False)
    sleep(5)
    loops = 3
    while loops > 0:
        for i in range(0, 10):
            Display.SetBrightness(i / 10.0)
            sleep(0.5)
        loops -= 1
    Display.StopClock()
    thread.interrupt_main()
except KeyboardInterrupt:
    print ("Properly closing the clock and open GPIO pins");
    Display.cleanup()
```

Shell

```
Starting clock in the background (press CTRL + C to stop):
Continue Python script and tweak Display!
Attempting to stop live clock
Properly closing the clock and open GPIO pins
Attempting to stop live clock

pwn@Pwn-Clock:~$

Starting clock in the background (press CTRL + C to stop):
Continue Python script and tweak Display!
```


Alt Gr
Fn

HDMI

IoT Journal Thonny - /home/pi/... 05:45

Thonny - /home/pi/Desktop/Rol Oct 2023/Led/ledFor40.py @ 10 : 28

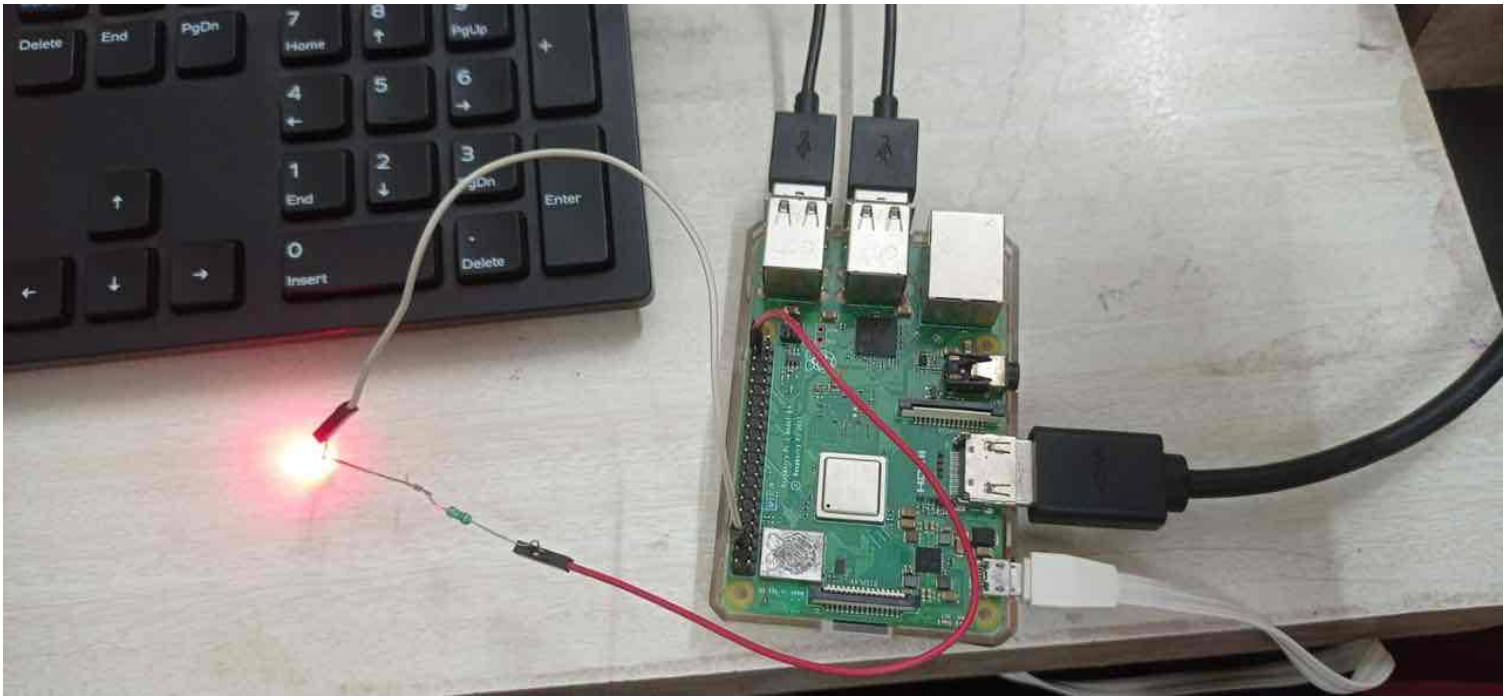
File Edit View Run Tools Help

ledFor40.py

```
import RPi.GPIO as IO              # calling header file for GPIO's of PI
import time                                # calling for time to provide delays in program

for i in range(0,5):
    IO.setwarnings(False)
    IO.setmode (IO.BOARD)          # programming the GPIO by BOARD pin numbers, GPIO21 is called as PIN40
    IO.setup(40,IO.OUT)                # initialize digital pin40 as an output.
    IO.output(40,1)                        # turn the LED on (making the voltage level HIGH)
    time.sleep(1)                           # sleep for a second
    IO.cleanup()                           # turn the LED off (making all the output pins LOW)
    time.sleep(1)                          #sleep for a second
```

Shell

```
Python 3.5.3 (/usr/bin/python3)
>>>
```

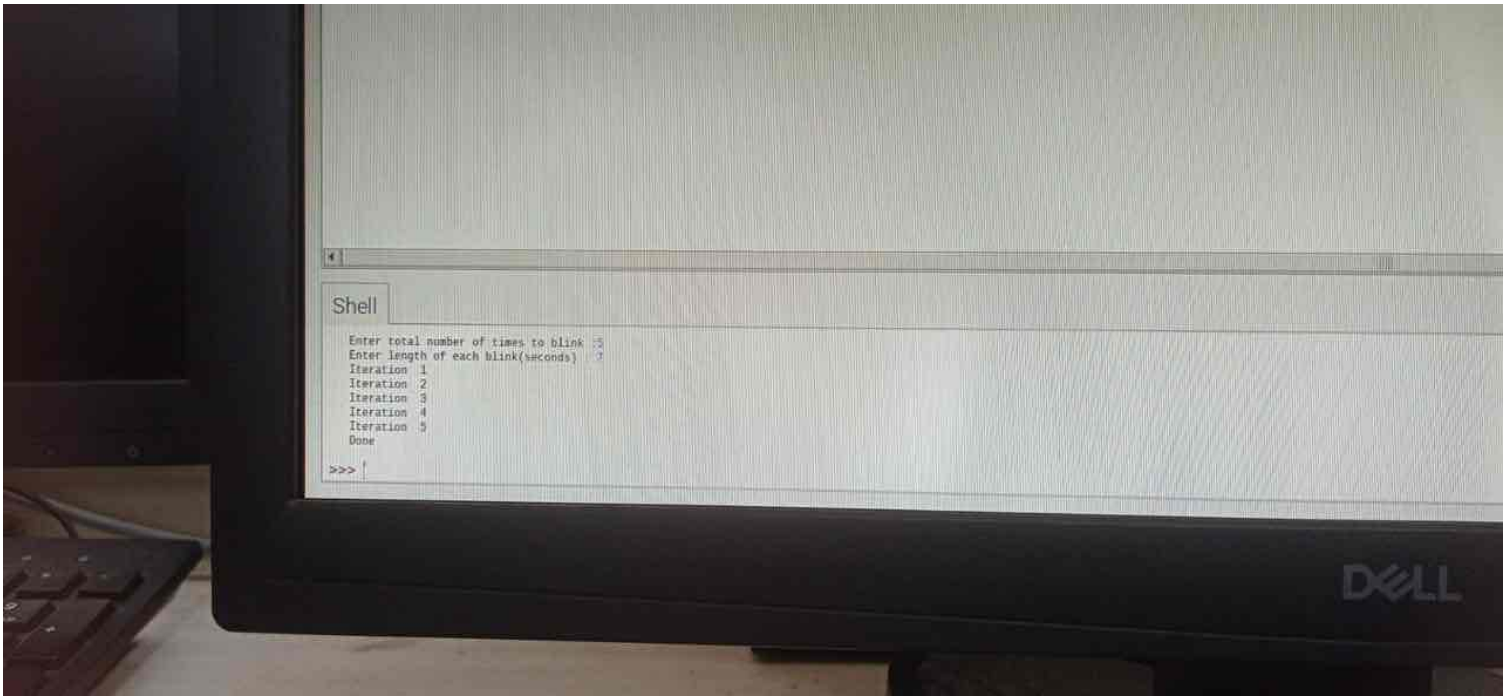

Delete
End
PgDn
7
Home
8
PgUp
4
5
6
1
End
2
3
PgDn
Enter
0
Insert
Delete

Thonny - /home/pi/Desktop/Rpi Oct 2023/Led/blink

File Edit View Run Tools Help

blinkSeries.py *

```
import RPi.GPIO as GPIO
import time
x=1
numTimes= int(input("Enter total number of times to blink : "))
speed = float(input("Enter length of each blink(seconds) : "))

GPIO.setwarnings(False)
GPIO.setmode(GPIO.BOARD)
GPIO.setup(40,GPIO.OUT)

def Blink(numTimes,speed):
    for i in range(0,numTimes):
        GPIO.output(40, True)
        print ("Iteration ",(i+1))
        time.sleep(speed)
        GPIO.output(40, False)
        time.sleep(speed)

Blink(numTimes, speed)
print("Done")
```


Shell
Enter total number of times to blink :5
Enter length of each blink(seconds) :2
Iteration 1
Iteration 2
Iteration 3
Iteration 4
Iteration 5
Done
>>>
DELL