Name- Suraj muresh Vishwakarma
Class: TYBSCIT

Roll No: IT22096
Subject: Internet of things
Date: / /

Practical 1.

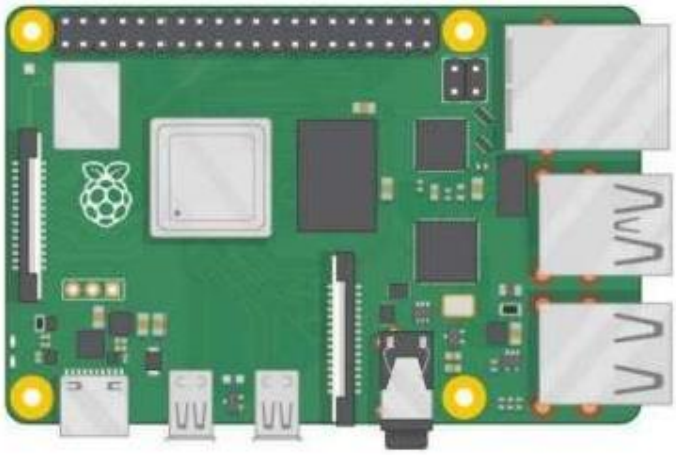
Aim : Booting Raspberry Pie

Step 1 : What will you need ?

Which Raspberry Pi ?

There are several models of Raspberry Pi (https://www.raspberrypi.org /products/), and for most people, Raspberry Pi 4 Model B is the one to choose. Raspberry Pi 4 Model B is the newest, fastest, and easiest to use.

Raspberry Pi 4 comes with 2GB, 4GB, or 8GB of RAM. For most educational purposes and hobbyist projects, and for use as a desktop computer, 2GB is enough.

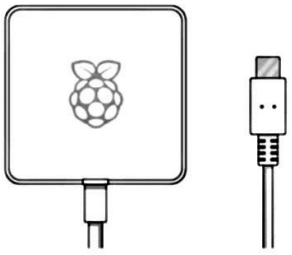


Raspberry Pi Zero, Raspberry Pi Zero W, and Raspberry Pi Zero WH are smaller and require less power, so they're useful for portable projects such as robots. It's generally easier to start a project with Raspberry Pi 4, and to move to Raspberry Pi Zero when you have a working prototype that a smaller Raspberry Pi would be useful for.

If you want to buy a Raspberry Pi, head to rpf.io/products (https://rpf.io/products).

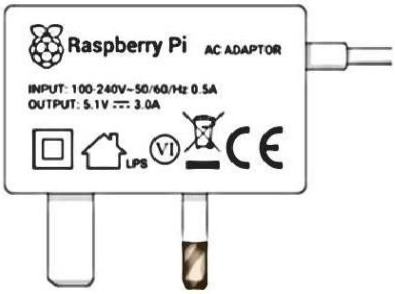
A power supply

To connect to a power socket, all Raspberry Pi models have a USB port (the same found on many mobilephones):either USB-C for Raspberry Pi 4, or micro USB for Raspberry Pi 3, 2, and 1.



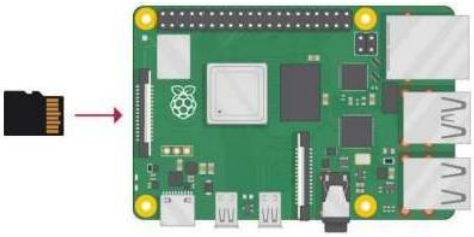
You need a power supply that provides :

- At least 3.0 amps for Raspberry Pi 4



- At least 2.5 amps for Raspberry Pi 3

A microSD Card



Your Raspberry Pi needs an SD card to store all its files and the Raspberry Pi OS operating system.
You need a microSD card with a capacity of at least 8GB. Many sellers supply SD cards for Raspberry Pi that are already set up with Raspberry Pi OS andreadytogo.

A Keyboard and a Mouse

To start using your Raspberry Pi, you need a USB keyboard and a USB mouse. Once you've set up your Raspberry Pi, you can use a Bluetooth keyboard and mouse, but you'll need a USB keyboard and mouse for the first setup.

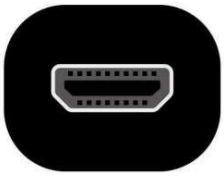
ATV or computer screen

To view the Raspberry Pi OS desktop environment, you need a screen, and a cable to link the screen and your Raspberry Pi. The screen can be a TV or a computer monitor. If the screen has built-in speakers. Raspberry Pi is able to use these to play sounds.

HDMI

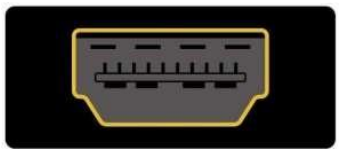
Your Raspberry Pi has an HDMI output port that is compatible with the HDMI port of most modern TVs and computer monitors. Many computer monitors may also have DVI or VGA ports.

Raspberry Pi 4 has two micro HDMI ports, allowing you to connect two separate monitors.



You need either a micro HDMI to HDMI cable, or a standard HDMI to HDMI cable plus a micro HDMI to HDMI adapter, to connect Raspberry Pi 4 to a screen.

Raspberry Pi 1, 2, and 3 have a single full-size HDMI port, so you can connect them to a screen using a standard HDMI to HDMI cable.

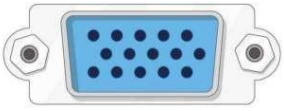


DVI

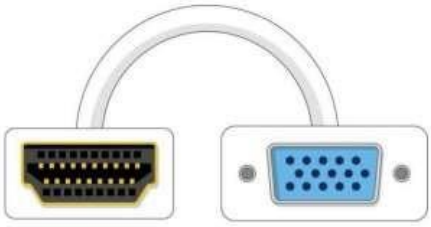
If your screen has a DVI port, you can connect your Raspberry Pi to it using an HDMI to DVI cable.

VGA

Some screens only have a VGA port.



To connect your Raspberry Pi to such a screen, you can use an HDMI to VGA adapter.



Headphones or Speakers

The large Raspberry Pi models (but not Raspberry Pi Zero or Raspberry Pi Zero W) have a standard audio port like the one on a smartphone or MP3 player. If you want to, you can connect your headphones or speakers so that your Raspberry Pi can play sound. If the screen you're connecting your Raspberry Pi to has built-in speakers, Raspberry Pi can play sound through these.

An Ethernet Cable

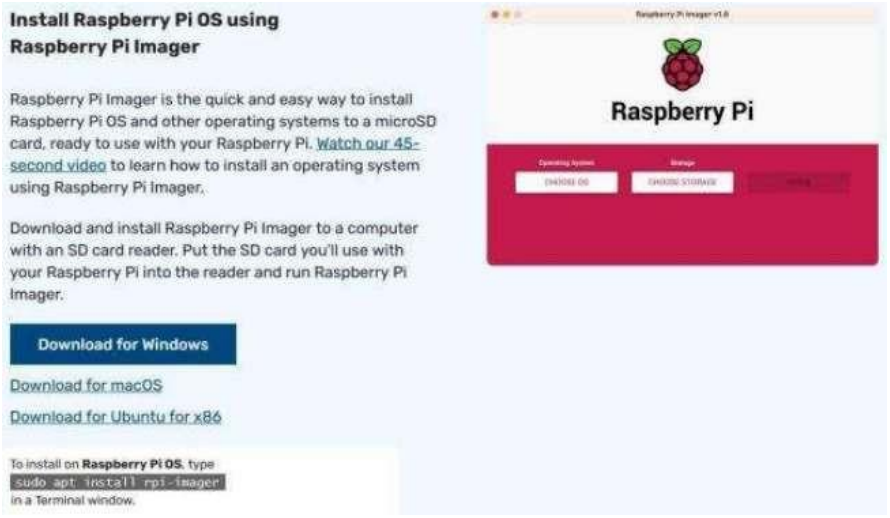
The large Raspberry Pi models (but not Raspberry Pi Zero or Raspberry Pi Zero W) have a standard Ethernet port to connect them to the internet: to connect Raspberry Pi Zero to the internet. you need a USB to Ethernet adapter. Raspberry Pi 4, Raspberry Pi 3, and Raspberry Pi Zero W can also be wirelessly connected to the internet.

Step 2 : Set up your SD card

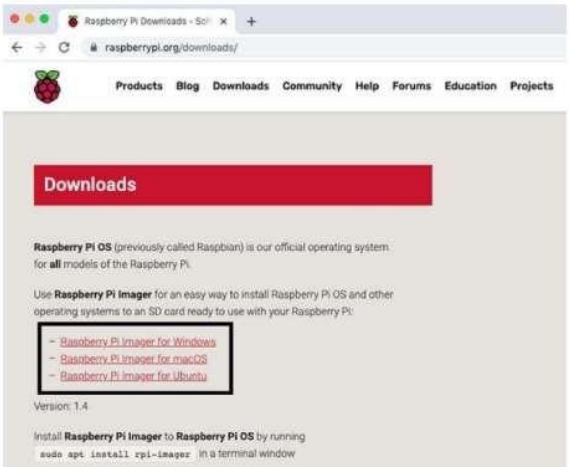
If you have an SD card that doesn't have the Raspberry Pi OS operating system on it yet, or if you want to reset your Raspberry Pi, you can easily install Raspberry Pi OS yourself. To do so. you need a computer that has an SD card port — most laptop and desktop computers have one.
The Raspberry Pi OS operating system via the Raspberry Pi Imager.
Using the Raspberry Pi Imager is the easiest way to install Raspberry Pi OS on your SD card.

Note: More advanced users looking to install a particular operating system should use this guide to installing operating system images (https://www.raspberrypi.org/documentation/installation/installing-images/ READM E.md}. Download and launch the Raspberry Pi Imager :

- Visit the Raspberry Pi downloads page (https://www.raspberrypi.org/downloads)



- Click on the link for the Raspberry Pi Imager that matches your operating system.



- When the download finishes, click it to launch the installer.

Using the Raspberry Pi Imager

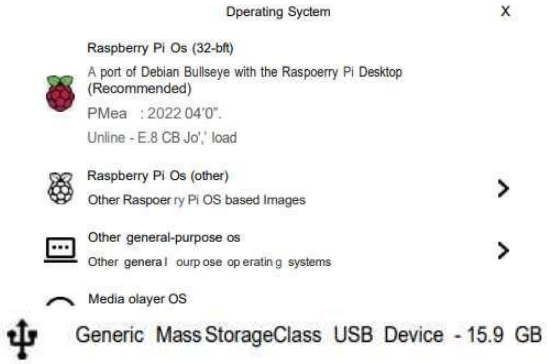
Anything that's stored on the SD card will be overwritten during formatting. If your SD card currently has any files on it, e.g. from an older version of Raspberry Pi OS. you may wish to back up these files first to prevent you from permanently losing them.

When you launch the installer. your operating system may try to block you from running it. For example, on Windows I receive the following message:

- If this pops up, click on and then.
- Follow the instructions to install and run the Raspberry Pi Imager.
- Insert your SD card into the computer or laptop SD card slot.
- In the Raspberry Pi Imager, select the OS that you want to install and the SD card you would like to install it on.

Note : You will need to be connected to the internet the first time for the Raspberry Pi Imager to download the OS that you choose. That OS will then be stored for future of

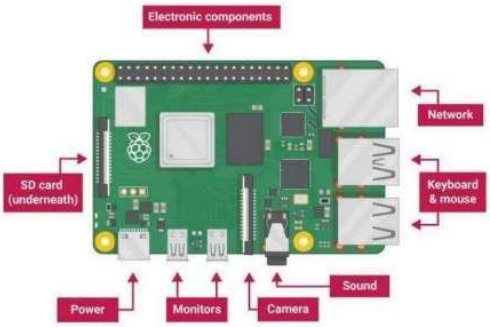
online use. Being online for later uses means that the Raspberry Pi imager will always give you the latest version.



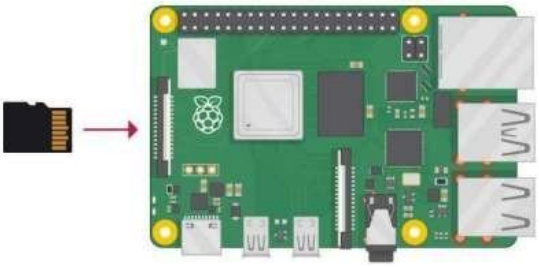
- Then simply click the WRITE button.
- Wait for the Raspberry Pi Imager to finish writing.
- Once you get the following message, you can eject your SD card :

Step 3 : Connect your Raspberry Pi

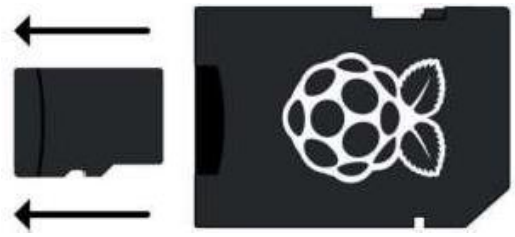
Now get everything connected to your Raspberry Pi. It's important to do this in the right order, so that all your components are safe.



- Insert the SD card you've set up with Raspberry Pi OS into the microSD card slot on the underside of your Raspberry Pi.

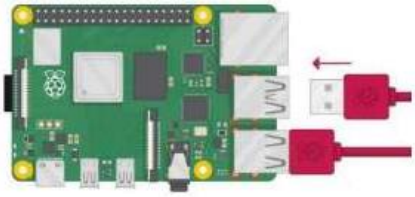


Note : Many microSD cards come inside a larger adapter - you can slide the smaller card out using the lip at the bottom.



- Find the USB connector end of your mouse's cable and connect the m use to a USB port on Raspberry Pi (it doesn't matter which port you use).

- Connect the keyboard in the same way.

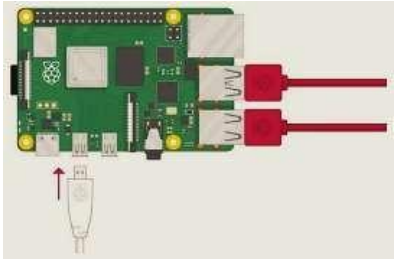


- Make sure your screen is plugged into a wall socket and switched on.
- Look at the HDMI port(s) on your Raspberry Pi - notice that they have a flat side on top.
- Use a cable to connect the screen to Raspberry Pi's HDMI port - use an adapter if necessary.

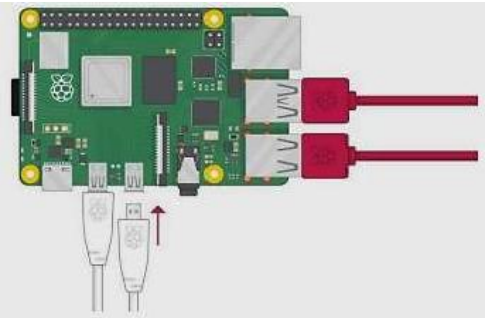
Raspberry Pi 4

Connect your screen to the first of Raspberry Pi 4's HDMI ports labelled HDMIO.

Note : Make sure you have used HDMIO (nearest the power in port) rather than HDMI1.

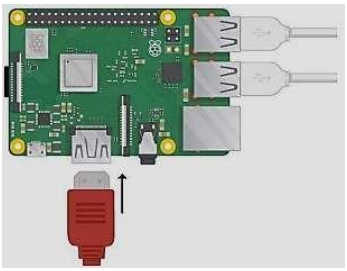


You can connect an optional second screen in the same way.



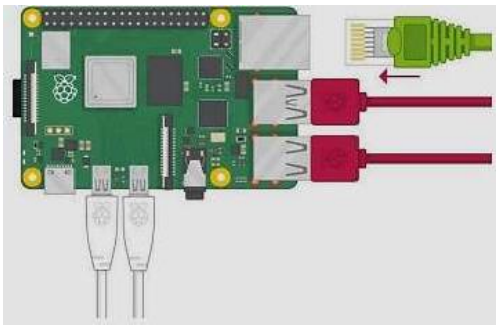
Raspberry Pi 1, 2, 3

- Connect your screen to the single HDMI port



Note : Nothing will display on the screen, because your Raspberry Pi is not running yet.

- If you want to connect your Raspberry Pi to the internet via Ethernet, use an Ethernet cable to connect the Ethernet port on Raspberry Pi to an Ethernet socket on the wall or on your internet router. You don't need to do this if you want to use wireless connectivity, or if you don't want to connect to the internet.

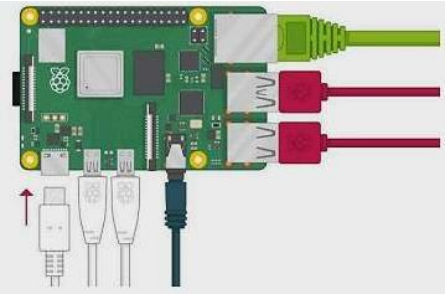


- If the screen you are using has speakers. sound will play through those. Alternatively, connect headphones or speakers to the audio port if you prefer.

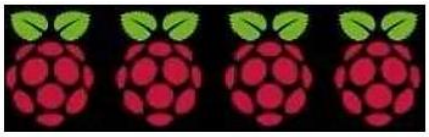
Step 4 : Start up your Raspberry Pi

Your Raspberry Pi doesn't have a power switch. As soon as you connect it to a power outlet. it will turn on.

- Plug the power supply into a socket and connect it to your Raspberry Pi's power port.



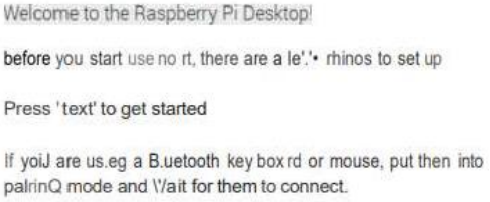
You should see a red LED light up on the Raspberry Pi, which indicates that Raspberry Pi is connected to power. As it starts up (this is also called booting), you will see raspberries appear in the top left-hand corner of your screen.



After a few seconds the Raspberry Pi OS desktop will appear.

Finishing the setup

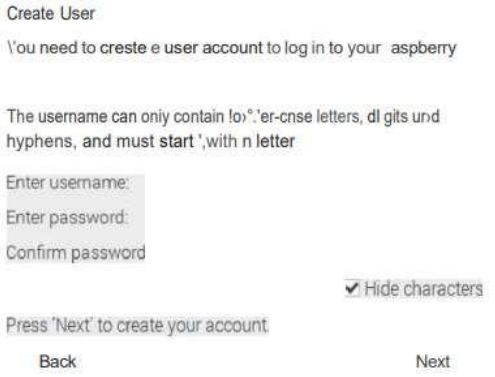
When you start your Raspberry Pi for the first time, the Welcome to Raspberry Pi application will pop up and guide you through the initial setup.



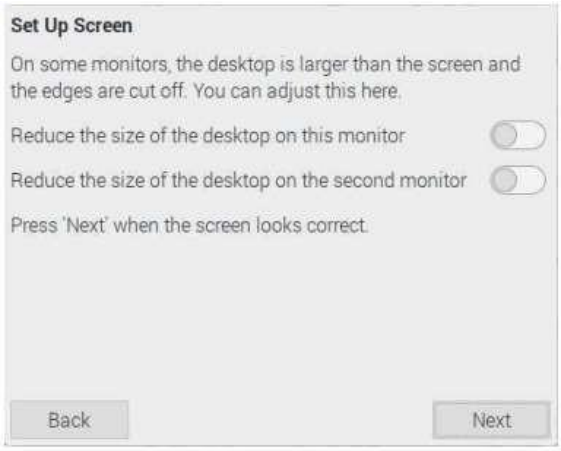
- Click on Next to start the setup.

- Set your Country, Language. and Time Zone, then click on Next again.

- Enter a new username and password for your Raspberry Pi and click on Next.



- Set up your screen so that the Desktop completely fills your monitor.



- Connect to your wireless network by selecting its name, entering the password, and clicking on Next.

Note : If your model of Raspberry Pi doesn't have wireless connectivity, you won't see this screen.

Note : Wait until the wireless connection icon appears and the correct time is shown before trying to update the software.

- Click on Next, and let the wizard check for updates to Raspberry Pi OS and install them(thismight takealittlewhile).

- Click on Restart to finish the setup.

Name- Suraj muresh Vishwakarma
Class: TYBSCIT

Roll No: IT22096
Subject: Internet of things
Date: / /

Practical 2.

Aim : Displaying Different LED Patterns with Raspberry Pi

Hardware Requirements :

1. **Breadboard** :- A breadboard is a tool used in electronics to prototype circuits without soldering. It has a grid of interconnected holes for components, power rails, and is reusable for experimenting with circuit designs.
2. **LED (Light-emitting-diode)** :- An LED is a small, energy-efficient semiconductor device that emits light when an electric current passes through it. It's used in lighting, displays, indicators, and various electronic applications.
3. **Resistor** :- A resistor is an electrical component that limits the flow of electric current in a circuit, typically used to control voltage levels, current flow, and adjust signal levels in electronics.
4. **Jumper Wire** :- A jumper wire is a short, flexible electrical wire used to establish connections between different points on a breadboard or electronic circuit, allowing for easy and temporary wiring during prototyping and testing.
5. **Raspberry Pi** :- A Raspberry Pi is a small, affordable, single-board computer that's widely used for various DIY projects, education, and prototyping. It can run a variety of operating systems and is popular for programming, learning about computing, and building innovative electronics projects.
6. **Keyboard** :- A keyboard is an input device used to type or enter text and commands into a computer or other digital devices by pressing individual keys, each representing a specific character or function.
7. **Mouse** :- A mouse is a pointing device used to control the movement of a cursor or pointer on a computer screen. It typically has buttons that allow users to interact with graphical user interfaces, select items, and perform actions by clicking or dragging the cursor.
8. **HDMI Cable** :- HDMI (High-Definition Multimedia Interface) cables are digital cables used to transmit high-quality audio and video signals between devices, such as TVs, monitors, projectors, and multimedia sources (like laptops, game consoles, or streaming devices).
9. **Ethernet Cable** :- Ethernet cable is used to connect devices in a wired network, enabling data communication, and it comes in different categories for various performance levels.
10. **Power Supply** :- A power supply converts incoming electrical energy into the right form to power electronic device.
11. **Male to Female Jumper Wire** :- A male-to-female jumper wire is a type of electrical cable with a male connector on one end and a female connector on the other, commonly used for connecting components or devices on a breadboard or in electronics projects.

Name- Suraj muresh Vishwakarma
Class: TYBSCIT

Roll No: IT22096
Subject: Internet of things
Date: / /

Steps :

1. Connect the LED to the breadboard.
2. LED had 2 terminals (One is the longer terminal that is positive, second is the shorter terminal that is negative).
3. Connect one side for the jumper wire below the LED positive side on the breadboard connect the jumper wire below the LED longer terminal.
4. Connect one end of the resistor on the breadboard right below the LED negative side.
5. The other end of the resistor connects it anywhere on the breadboard. Now connect another jumper wire right above the second end of the resistor.
6. Connect the positive side of the jumper wire on PIN 7, 29, 31, 33 and the negative side on PIN 9 of the raspberry pi.
   i. Longer terminal = Positive (Raspberry Pi Pin7)
   ii. Shorter terminal = Negative (Raspberry Pi Pin 9)

Code :

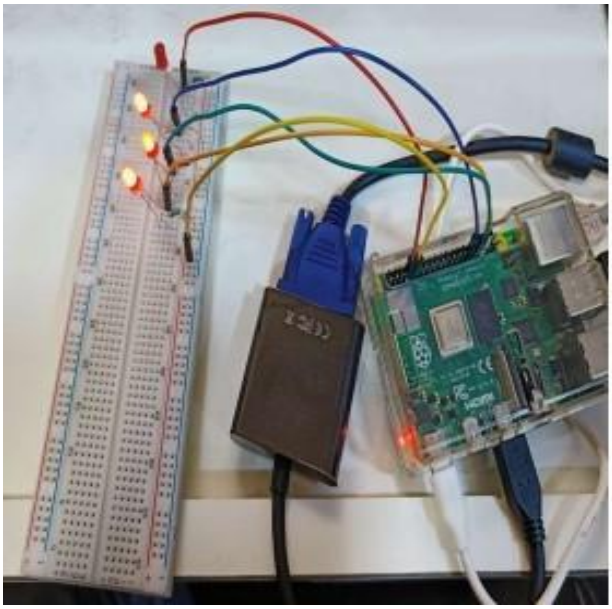
```
import RPi.GPIO as GPIO
import time
GPIO.setmode(GPIO.BOARD)
GPIO.setwarnings(False)
GPIO.setup(7,GPIO.OUT)
GPIO.setup(29,GPIO.OUT)
GPIO.setup(31,GPIO.OUT)
GPIO.setup(33,GPIO.OUT)
while(1):
  GPIO.output(7,False)
  print("LED 1 IS OFF")
  time.sleep(1)
  GPIO.output(29,False)
  print("LED 2 IS OFF")
  time.sleep(1)
  GPIO.output(31,False)
  print("LED 3 IS OFF")
  time.sleep(1)
  GPIO.output(33,False)
  print("LED 4 IS OFF")
  time.sleep(1)

  GPIO.output(7,True)
  print("LED 1 IS FINALLY ON")
  time.sleep(1)
  GPIO.output(29,True)
```

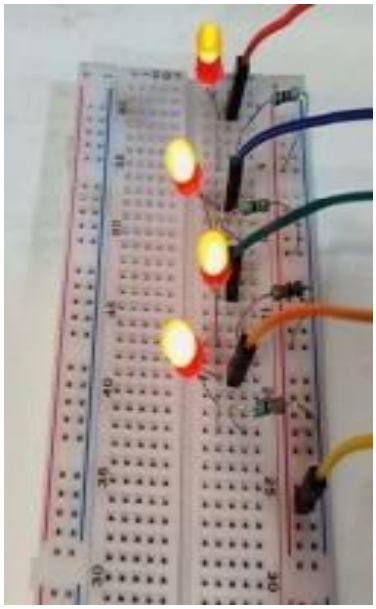
Name- Suraj muresh Vishwakarma
Class: TYBSCIT

Roll No: IT22096
Subject: Internet of things
Date: / /

```
  print("LED 2 IS FINALLY ON")
  time.sleep(1.5)
  GPIO.output(31,True)
  print("LED 3 IS FINALLY ON")
  time.sleep(2)
  GPIO.output(33,True)
  print("LED 4 IS FINALLY ON")
  time.sleep(2.5)

  GPIO.output(7,False)
  print("LED 1 IS OFF")
  time.sleep(1)
  GPIO.output(29,False)
  print("LED 2 IS OFF")
  time.sleep(1)
  GPIO.output(31,False)
  print("LED 3 IS OFF")
  time.sleep(1)
  GPIO.output(33,False)
  print("LED 4 IS OFF")
  time.sleep(1)
  print("PROGRAM COMPLETE!")
GPIO.cleanup()
```

Output :





Name- Suraj muresh Vishwakarma
Class: TYBSCIT

Roll No: IT22096
Subject: Internet of things
Date: / /

Practical 3.

Aim : Displaying Time over 4-Digit 7-Segment Display using Raspberry Pi.

Hardware Requirements :

1. Digit Display : In IoT, a digit display is a visual interface that shows numerical data from connected devices or sensors, aiding users in monitoring and interacting with real-time information.

2. Raspberry Pi : A Raspberry Pi is a small, affordable, singleboard computer that's widely used for various DIY projects, education, and prototyping. It can run a variety of operating systems and is popular for programming, learning about computing, and building innovative electronics projects.

3. Keyboard : A keyboard is an input device used to type or enter text and commands into a computer or other digital devices by pressing individual keys, each representing a specific character or function.

4. Mouse : A mouse is a pointing device used to control the movement of a cursor or pointer on a computer screen. It typically has buttons that allow users to interact with graphical user interfaces, select items, and perform actions by clicking or dragging the cursor.

5. HDMI Cable : HDMI (High-Definition Multimedia Interface) cables are digital cables used to transmit high-quality audio and video signals between devices, such as TVs, monitors, projectors, and multimedia sources (like laptops, game consoles, or streaming devices).

6. Ethernet Cable : Ethernet cable is used to connect devices in a wired network, enabling data communication, and it comes in different categories for various performance levels.

7. Power Supply : A power supply converts incoming electrical energy into the right form to power electronic device.

8. Female to Female Jumper Wire: A female-to-female jumper wire in IoT is a connector cable with female connectors on both ends, used to link components or sensors with female pins or headers.

Steps :

1. Open the web browser and go to the link: https://github.com/timwaizenegger/raspberrypiexamples/tree/master/actor-led-7segment-4numbers.

2. Click on the actor–>led-7segment-4numbers.zip folder and download the zip file.

3. Go to the File Manager→ Downloads→ unzip the actor led-7 segment- 4 numbers.zip file → documents folder →python projects.

Name- Suraj muresh Vishwakarma
Class: TYBSCIT

Roll No: IT22096
Subject: Internet of things
Date: / /

4. Make the connections as follows: a. Connect Pin2 (5V) of RPI to VCC PIN of 7 segment module. b. Connect Pin6 (Ground) of RPI to Ground pin of 7 Segment Module. c. Connect Pin 38 of RPI to DIO pin of 7 Segment Module. d. Connect Pin 40 of RPI to CLK of the 7 Segment Module.

5. Go to location where you have downloaded seven segment file, copy the location of that file.

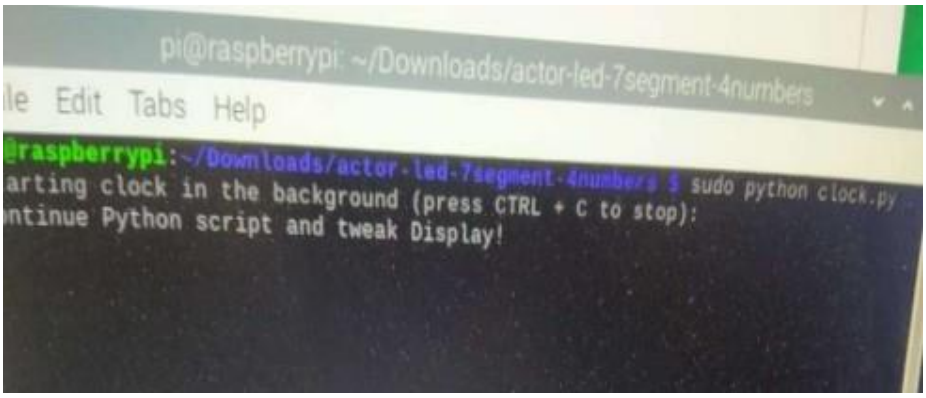
6. Open the terminal and paste your location as “cd location”.

7. After entering location we have to give the command to run seven segment as “sudo python clock.py”.

Terminal Commands :

cd (file location) sudo python clock.py

Output :



Name- Suraj muresh Vishwakarma
Class: TYBSCIT

Roll No: IT22096
Subject: Internet of things
Date: / /

Practical 4.

Aim : Connecting 2x16 LCD Display to Raspberry Pi

Hardware Requirements :

1. **Raspberry Pi Model 3 B/B+**
2. **2x16 LCD Display**
3. **Jumper wires (Female to Male, Male to Male)**
4. **Breadboard**
5. **Potentiometer (to adjust Contrast)**

Software Requirements :

1. **Raspbian Stretch OS**
2. **Adafruit CharLCD Library**

Connect your 2x16 LCD display with Raspberry Pi's GPIO Pins :

All the lcds can be used in 8-bit as well as 4-bit mode.

The major difference in 4 bit and 8 bit mode lies in data pins used and LCD initializing commands.

In 4 bit mode only four data pins D4-D5-D6-D7 are used. Character 8-bit ASCII value is divided in to two 4-bit nibbles. High nibble is send first following by the lower nibble.

In 8 bit mode only one pulse signal is required to display a character on LCD. Thus it is faster than 4-bit mode. It uses all eight data pins D0-D7.

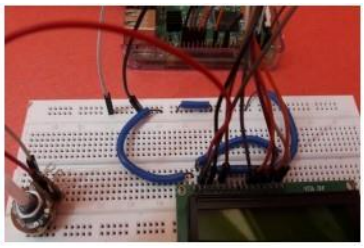
So, 4 bit mode saves 4 GPIO pins. Which can be utilized elsewhere.

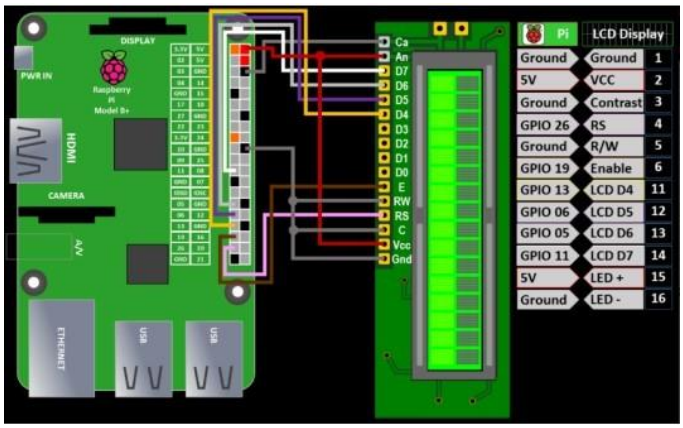
Here, I am using 4 bit mode to connect 2x16 LCD Display to Raspberry Pi

Category	Sr No	LCD Display Board Pin	RPI Physical Pin	Raspberry Function
Power Pins	1	Ground	20	GND
	2	VCC or VDD	2	5v Power
Contrast Pin	3	V0 or VEE	20	GND
Control Pins	4	RS	37	GPIO26
	5	RW	20	GND
	6	E	35	GPIO19
Data Pins	7	D0	Not used in 4 Bit Mode	
	8	D1		
	9	D2		
	10	D3		
	11	D4	33	GPIO13
	12	D5	31	GPIO6
	13	D6	29	GPIO5
	14	D7	23	GPIO11
Backlight Pins	15	LED+ or A	2	5v Power
	16	LED- or K	6	GND

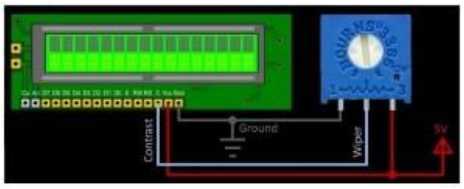
Name- Suraj muresh Vishwakarma
Class: TYBSCIT

Roll No: IT22096
Subject: Internet of things
Date: / /





A potentiometer can be added to the LCD display C pin to control the contrast. The pin is connected to the wiper of the pot which is usually the middle pin. One of the outside pins is connected to ground and the other to VCC. It doesn't matter which outside pin goes to VCC or ground because the pot is acting as a voltage divider either way. As the dial is turned the wiper's pin voltage will vary from 0V to 5V and this will cause the contrast to change. 10K Ω is a good value for the pot.



In my case, I connected middle pin of potentiometer to Contrast (Pin no 3) of 2x16 LCD display. And third pin of potentiometer is connected to GND.

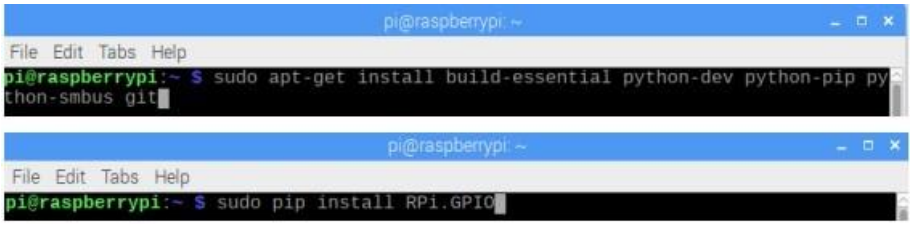
Name- Suraj muresh Vishwakarma
Class: TYBSCIT

Roll No: IT22096
Subject: Internet of things
Date: / /

Steps :

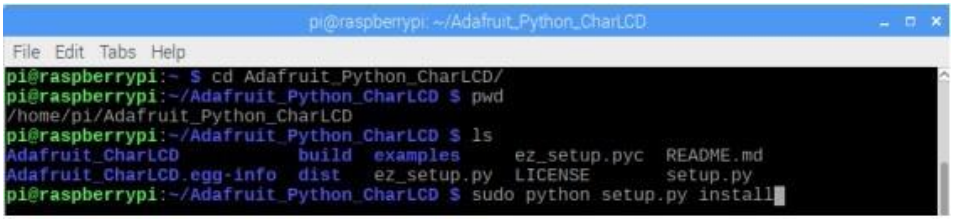
1: Update Raspberry Pi

2. Before installing the character LCD library install few dependencies by executing following the steps :

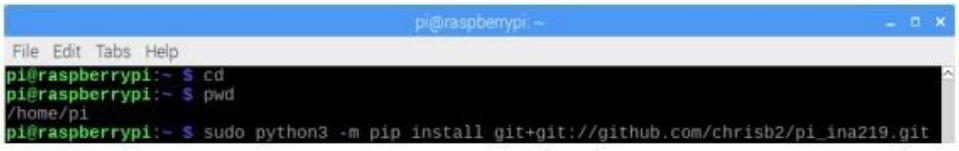


3. Download Adafruit CharLCD library

4. Run setup.py file to install Adafruit Library



5. Install dependencies for Python 3 on Raspberry Pi, following command need to be executed.



Note: This command helps to fix the ImportError occurred, when you execute script from Python IDLE.

6. Create Python script which imports Adafruit_CharLCD library and initialize required GPIO BCM Pin no. (Used in your LCD Connection)

Name- Suraj muresh Vishwakarma
Class: TYBSCIT

Roll No: IT22096
Subject: Internet of things
Date: / /

```
pi@raspberrypi:~ $ cd Adafruit_Python_CharLCD/Adafruit_CharLCD
pi@raspberrypi:~/Adafruit_Python_CharLCD/Adafruit_CharLCD $ pwd
/home/pi/Adafruit_Python_CharLCD/Adafruit_CharLCD
pi@raspberrypi:~/Adafruit_Python_CharLCD/Adafruit_CharLCD $ ls
Adafruit_CharLCD.py   display IP and date time on LCD.py   __pycache__
Adafruit_CharLCD.pyc  __init__.py
pi@raspberrypi:~/Adafruit_Python_CharLCD/Adafruit_CharLCD $ nano __init__.py
```

```
import time
from Adafruit_CharLCD import Adafruit_CharLCD

# instantiate lcd and specify pins
lcd = Adafruit_CharLCD(rs=26, en=19,
                       d4=13, d5=6, d6=5, d7=11,
                       cols=16, lines=2)
lcd.clear()

# display text on LCD display \n = new line
lcd.message('2x16 CharLCD\n  Raspberry Pi')
time.sleep(2)
lcd.clear()
lcd.message('MSD Gurukul\nLCD DEMO')

time.sleep(3)
# scroll text off display
for x in range(0, 16):
    lcd.move_right()
    time.sleep(.1)

time.sleep(3)

# scroll text on display
for x in range(0, 16):
    lcd.move_left()
    time.sleep(.1)
```

Execute above script using :

python __init__.py

7. Create Python script which displays current time and IP address on LCD Display

```
pi@raspberrypi: ~/Adafruit_Python_CharLCD/Adafruit_CharLCD
File  Edit  Tabs  Help
pi@raspberrypi:~/Adafruit_Python_CharLCD/Adafruit_CharLCD $ pwd
/home/pi/Adafruit_Python_CharLCD/Adafruit_CharLCD
pi@raspberrypi:~/Adafruit_Python_CharLCD/Adafruit_CharLCD $ ls
Adafruit_CharLCD.py   display IP and date time on LCD.py   __pycache__
Adafruit_CharLCD.pyc  __init__.py
pi@raspberrypi:~/Adafruit_Python_CharLCD/Adafruit_CharLCD $ nano display\ IP\ and\ date\ t
ime\ on\ LCD.py
```

```
#!/usr/bin/python
from Adafruit_CharLCD import Adafruit_CharLCD
from time import sleep, strftime
from datetime import datetime
import socket

#  Initialize LCD (must specify pinout and dimensions)
lcd = Adafruit_CharLCD(rs=26, en=19,
                       d4=13, d5=6, d6=5, d7=11,
                       cols=16, lines=2)

def get_ip_address():
    return [
            (s.connect(('8.8.8.8', 53)),
             s.getsockname()[0],
             s.close()) for s in
                 [socket.socket(socket.AF_INET, socket.SOCK_DGRAM)]
           ][0][1]

try:
    while 1:
        lcd.clear()
        ip = get_ip_address()
        lcd.message(datetime.now().strftime('%b %d  %H:%M:%S\n'))
        lcd.message('IP {}'.format(ip))
        sleep(2)
```

```
except KeyboardInterrupt:
    print('CTRL-C pressed.  Program exiting...')

finally:
    lcd.clear()
```

Execute above script using :

python filename.py

8. You can even run Adafruit library from Python IDLE. (Interactive mode)

```
>>> import os
>>> os.chdir('/home/pi/Adafruit_Python_CharLCD/Adafruit_CharLCD')
>>> from Adafruit_CharLCD import Adafruit_CharLCD
>>> lcd = Adafruit_CharLCD(rs=26, en=19,
                           d4=13, d5=6, d6=5, d7=11,
                           cols=16, lines=2)
>>> lcd.message('MSD Gurukul\nLCD DEMO')
>>> lcd.clear()          #clears LCD Display
>>> lcd.show_cursor(True)       #displays cursor
>>> lcd.blink(True)             #blinks cursor
>>> lcd.move_left()             #moves string on left side
>>> lcd.move_right()            #moves string on right side
>>> lcd.show_cursor(False)      #hides cursor
>>> lcd.message("Thank You")
>>>
```

Name- Suraj muresh Vishwakarma
Class: TYBSCIT

Roll No: IT22096
Subject: Internet of things
Date: / /

Practical 5.

Aim : Raspberry Pi based Oscilloscope

Hardware Requirements :

1. Raspberry Pi Model A/B/B+

2. ADS1115 ADC

3. Breadboard

4. Jumper Wires

ADS1115 ADC chip is used to convert the analog input signals to digital signals which can be visualized with the Raspberry Pi. This chip is important because the Raspberry Pi does not have an onboard analog to digital converter (ADC).

Software Requirements :

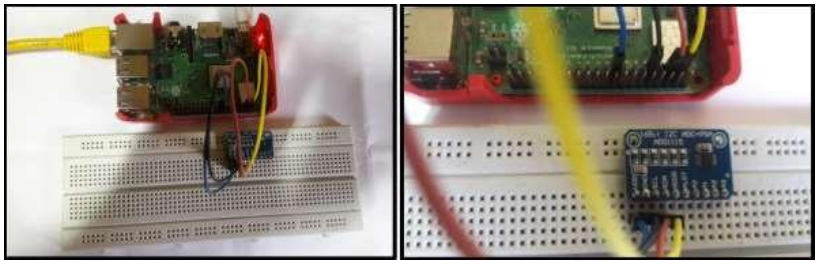
1. Raspbian Stretch OS

2. Adafruit module for interfacing with the ADS1115 ADC chip

3. Python Module matplotlib used for data visualization

Connect your ADC with Raspberry Pi's GPIO Pins

ADS1115 ADC	Pin Number	GPIO Number
VDD	Pin 17	3.3v
GND	Pin 9	GND
SCL	Pin 5	GPIO 3
SDA	Pin 3	GPIO 2



Steps :

Name- Suraj muresh Vishwakarma
Class: TYBSCIT

Roll No: IT22096
Subject: Internet of things
Date: / /

1. Enable Raspberry Pi I2C interface

Go to Interfacing Options I2C Enable (Yes)

2. Update the Raspberry pi

3. Install the Adafruit ADS1115 library for ADC

To install the dependencies starting with the Adafruit python module for the ADS115 chip, Ensure you are in the Raspberry Pi home directory ($ cd ~)

Next, clone the Adafruit git folder for the library by running.

Change into the cloned file's directory and run the setup file.

4. Test the library and 12C communication.

Now,it is important to test the library and ensure the ADC can communicate with the raspberry pi over I2C. To do this use an example script that comes with the library.
$ cd examples
$ python simpletest.py

If the I2C module is enabled and connections good, it should display the data as below.
If an error occurs, check to ensure the ADC is well connected to the PI and I2C communication is enabled on the Pi.

5. Install Matplotlib

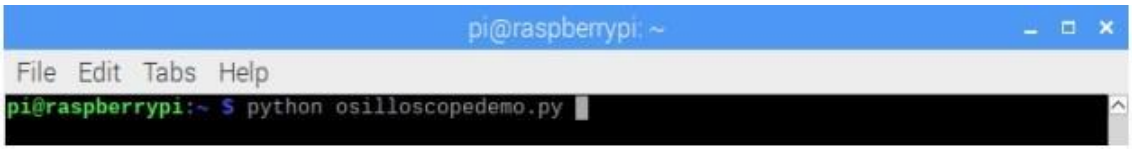
With all the dependencies installed, we are now ready to write the code.
At this stage it is important to switch to a monitor or use the VNC viewer (or Remote Desktop Connection), anything through which you can see your Raspberry Pi's desktop, as the graph being plotted won't show on the terminal.

6. Python Code for Raspberry Pi Oscilloscope :

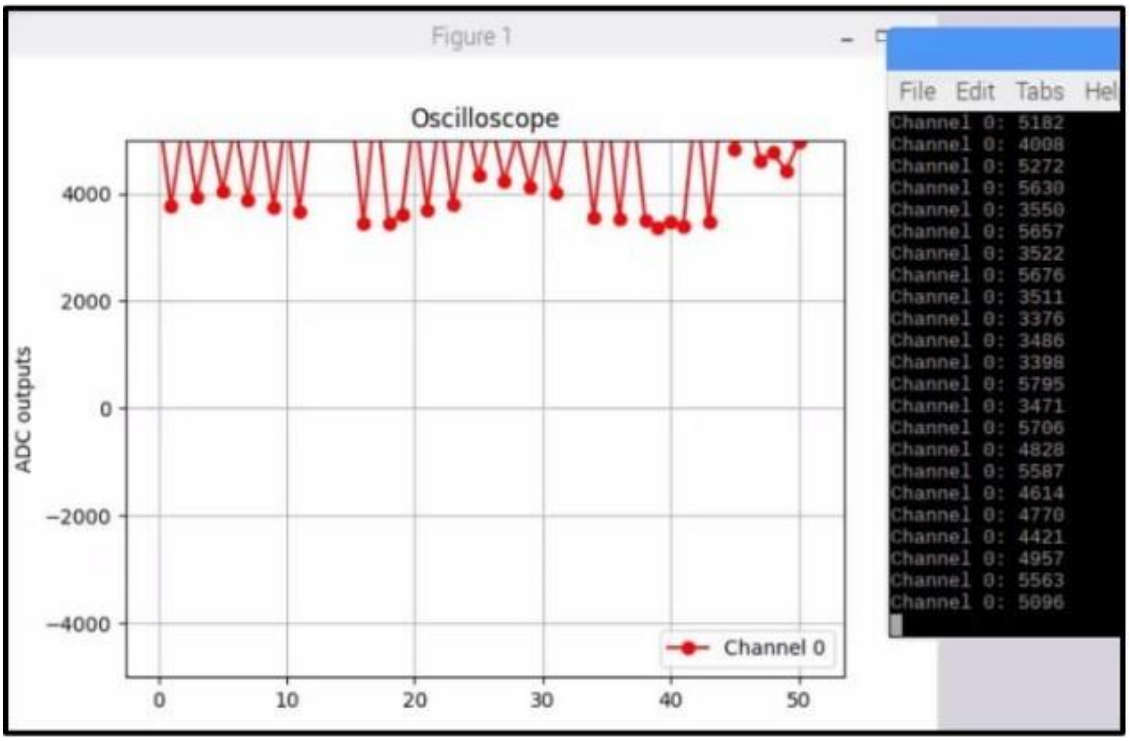
```
import matplotlib.pyplot as plt
from matplotlib.animation import FuncAnimation
import Adafruit_ADS1x15

# Create an ADS1115 ADC (16-bit) instance.
adc = Adafruit_ADS1x15.ADS1115()

GAIN = 1
val = [ ]


# Start continuous ADC conversions on channel 0 using the previous gain value.
adc.start_adc(0, gain=GAIN)
print('Reading ADS1x15 channel 0')

fig, ax = plt.subplots()
ax.set_ylim(-5000,5000)
ax.set_title('Oscilloscope')
ax.grid(True)
ax.set_ylabel('ADC outputs')

line, = ax.plot([], 'ro-', label='Channel 0')
ax.legend(loc='lower right')

def update(cnt):
    # Read the last ADC conversion value and print it out.
    value = adc.get_last_result()
    print('Channel 0: {0}'.format(value))
```

```
    # Set new data to line
    line.set_data(list(range(len(val))), val)
    ax.relim()
    ax.autoscale_view()
    #Store values for later
    val.append(int(value))
    if(cnt>50):
        val.pop(0)

ani = FuncAnimation(fig, update, interval=500)
plt.show()
```

Save the code and run using :



ADC data being printed on the terminal,and related Plot is also visible.



Name- Suraj muresh Vishwakarma
Class: TYBSCIT

Roll No: IT22096
Subject: Internet of things
Date: / /

Practical 6.

Aim : Interfacing Telegram with Raspberry Pi

Hardware Requirements :

1. **Breadboard** :- A breadboard is a tool used in electronics to prototype circuits without soldering. It has a grid of interconnected holes for components, power rails, and is reusable for experimenting with circuit designs.
2. **LED (Light-emitting-diode)** :- An LED is a small, energy-efficient semiconductor device that emits light when an electric current passes through it. It's used in lighting, displays, indicators, and various electronic applications.
3. **Resistor** :- A resistor is an electrical component that limits the flow of electric current in a circuit, typically used to control voltage levels, current flow, and adjust signal levels in electronics.
4. **Jumper Wire** :- A jumper wire is a short, flexible electrical wire used to establish connections between different points on a breadboard or electronic circuit, allowing for easy and temporary wiring during prototyping and testing.
5. **Raspberry Pi** :- A Raspberry Pi is a small, affordable, single-board computer that's widely used for various DIY projects, education, and prototyping. It can run a variety of operating systems and is popular for programming, learning about computing, and building innovative electronics projects.
6. **Keyboard** :- A keyboard is an input device used to type or enter text and commands into a computer or other digital devices by pressing individual keys, each representing a specific character or function.
7. **Mouse** :- A mouse is a pointing device used to control the movement of a cursor or pointer on a computer screen. It typically has buttons that allow users to interact with graphical user interfaces, select items, and perform actions by clicking or dragging the cursor.
8. **HDMI Cable** :- HDMI (High-Definition Multimedia Interface) cables are digital cables used to transmit high-quality audio and video signals between devices, such as TVs, monitors, projectors, and multimedia sources (like laptops, game consoles, or streaming devices).
9. **Ethernet Cable** :- Ethernet cable is used to connect devices in a wired network, enabling data communication, and it comes in different categories for various performance levels.
10. **Power Supply** :- A power supply converts incoming electrical energy into the right form to power electronic device.
11. **Mobile Phones**:- A mobile phone is a wireless handheld device that allows users to make and receive calls. While the earliest generation of mobile phones could only make and receive calls, today's mobile phones do a lot more, accommodating web browsers, games, cameras, video players and navigational systems.

Name- Suraj muresh Vishwakarma
Class: TYBSCIT
Roll No: IT22096
Subject: Internet of things
Date: / /

12. **Male to Female Jumper Wire** :- A male-to-female jumper wire is a type of electrical cable with a male connector on one end and a female connector on the other, commonly used for connecting components or devices on a breadboard or in electronics projects.

Steps :

1. First to start the practical you need 2 LEDs, 2 resistors, and 3 jumper wires.
2. Now connect two LEDs to the breadboard
3. Next connect the 2 resistors, one end to the negative end of the LEDs on the breadboard and the second end to the endpoint of the breadboard.
4. Connect 2 Jumper Wires to the Positive end of the LEDs.
5. For the Ground Connection add the third Jumper Wire to the bottom of the breadboard to connect all the resistors.
6. Turn on your mobile phone and install telegram.
7. Create a bot with the help of botfather commands to create a bot :
   a. Go to telegram and search and open botfather :
      Type the commands:
      i. /Start
      ii. /newbot
8. Once the bot father is created a bot it will generate a unique id the unique id needs to be added into the code.
9. Open the bot with the name which you have created Give the start command to turn on the bot. Now you can give commands to blink the leds or to create a pattern.

Terminal Command :

```
sudo pip3 install teleport
```

Code :

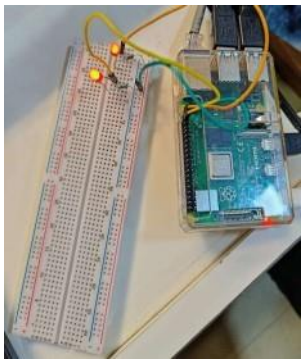
```
import datetime
import telepot
from telepot.loop import MessageLoop
import RPi.GPIO as GPIO
from time import sleep
red_led_pin = 21
green_led_pin = 20
GPIO.setmode(GPIO.BCM)
GPIO.setup(red_led_pin, GPIO.OUT)
GPIO.setup(green_led_pin, GPIO.OUT)
now = datetime.datetime.now()
def handle(msg):
  chat_id = msg['chat']['id']
  command = msg['text']
  print("Received:")
  print(command)
if command == '/hi':
```

Name- Suraj muresh Vishwakarma
Class: TYBSCIT

Roll No: IT22096
Subject: Internet of things
Date: / /

```
  bot.sendMessage(chat_id, str("Hi! "))
elif command == '/r1':
  bot.sendMessage(chat_id, str("Red led is ON"))
  GPIO.output(red_led_pin, True)
elif command == '/r0':
  bot.sendMessage(chat_id, str("Red led is OFF"))
  GPIO.output(red_led_pin, False)
elif command == '/g1':
  bot.sendMessage(chat_id, str("Green led is ON"))
  GPIO.output(green_led_pin, True)
elif command == '/g0':
  bot.sendMessage(chat_id, str("Green led is OFF"))
  GPIO.output(green_led_pin, False)
bot = telepot.Bot('6558187738:AAHvdvVes5nfG2RyLWgm7zZrzTU87DzqYiY')
print(bot.getMe())
MessageLoop(bot,handle).run_as_thread()
print("Listening...")
while (1):
  sleep(10)
```

Output :



Name- Suraj muresh Vishwakarma
Class: TYBSCIT

Roll No: IT22096
Subject: Internet of things
Date: / /

Practical 7.

Aim : Controlling Raspberry Pi with WhatsApp

Steps :

1. Installation

Update the packages with :

```
sudo apt-get update
sudo apt-get upgrade
```

Update firmware :

```
sudorpi-update
```

Prepare the system with the necessary components to Yowsup :

```
sudo apt-get install python-dateutil
sudo apt-get install python-setuptools
sudo apt-get install python-dev
sudo apt-get install libevent-dev
sudo apt-get install ncurses-dev
```

Download the library with the command :

```
git clone git://github.com/tgalal/yowsup.git
```

Navigate to the folder :

```
cd yowsup
```

And install the library with the command :

```
sudo python setup.py install
```

2. Registration

After installing the library we have to register the device to use WhatsApp. Yowsup comes with a cross platform command-line frontend called yowsup-cli. It provides you with the options of registration, and provides a few demos such as a command line client.

WhatsApp registration involves 2 steps. First you need to request a registration code. And then you resume the registration with code you got.

Request a code with command :

```
python yowsup-cli registration --requestcodesms --phone 39xxxxxxxxx --cc 39 --mcc 2 22 --mnc 10
```

Name- Suraj muresh Vishwakarma
Class: TYBSCIT

Roll No: IT22096
Subject: Internet of things
Date: / /

Replace with your data,
cc is your country code in this example 39 is for Italy,
mcc is Mobile Country Code
mnc is Mobile Network Code
You should receive on your phone a sms message with a code like xxx-xxx

Send a message to request registration with this command, (replace xxx-xxx with code you received)
python yowsup-cli registration --register xxx-xxx --phone 39xxxxxxxxxx --cc 39

If all goes well, we should get a message like this :
status: ok
kind: free
pw: xxxxxxxxxxxxxxxxxx=
price: € 0,89
price_expiration: 1416553637
currency: EUR
cost: 0.89
expiration: 1445241022
login: 39xxxxxxxxxx
type: existing

Warning :
WhatsApp requires the registration of a number, and with that number you can use WhatsApp on only one device at a time, so it is preferable to use a new number.
WhatsApp can be used on one device at a time and if you will make many attempts to register the number, it could be banned. We recommend you using Telegram.

Name- Suraj muresh Vishwakarma
Class: TYBSCIT

Roll No: IT22096
Subject: Internet of things
Date: / /

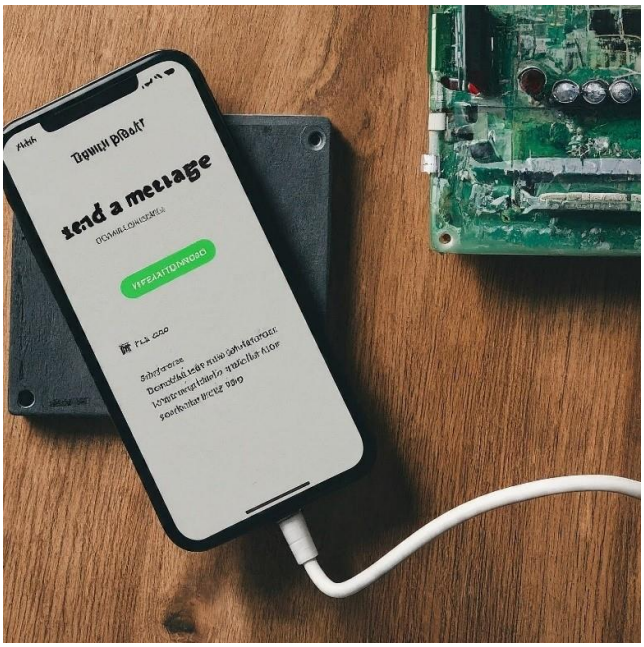
3. Utilization Create a file to save your credentials
sudonano /home/pi/yowsup/config

with this content :
Actual config starts below
cc=39 #if not specified it will be autodetected
phone=39xxxxxxxxxx
password=xxxxxxxxxxxxxxx=

Ok, we're ready for the test, Yowsup has a demo application in /home/pi/yowsup/yowsup/demos
Navigate to yowsup folder
cd /home/pi/yowsup

Start yowsup-cli demos with the command :
yowsup-cli demos --yowsup --configconfig

You can see Yowsupprompt :
If type "/help" you can see all available commands.
First use the '/L' command for login; to send a message type :
/message send 39xxxxxxxxxx "This is a message sent from Raspberry Pi"
Replace xxx with the recipient number.
If you respond with a message it will be displayed on Raspberry.



Name- Suraj muresh
Vishwakarma
Class: TYBSCIT

Roll No: IT22096
Subject: Internet of things
Date: / /

Practical 8.

Aim : Raspberry Pi GPS Module Interfacing

Hardware Requirements:

1. GPS : GPS (Global Positioning System) is a satellite-based navigation system that provides accurate location and time information to users anywhere on Earth.
2. Raspberry Pi : A Raspberry Pi is a small, affordable, singleboard computer that's widely used for various DIY projects, education, and prototyping. It can run a variety of operating systems and is popular for programming, learning about computing, and building innovative electronics projects.
3. Keyboard : A keyboard is an input device used to type or enter text and commands into a computer or other digital devices by pressing individual keys, each representing a specific character or function.
4. Mouse : A mouse is a pointing device used to control the movement of a cursor or pointer on a computer screen. It typically has buttons that allow users to interact with graphical user interfaces, select items, and perform actions by clicking or dragging the cursor.
5. HDMI Cable : HDMI (High-Definition Multimedia Interface) cables are digital cables used to transmit high-quality audio and video signals between devices, such as TVs, monitors, projectors, and multimedia sources (like laptops, game consoles, or streaming devices).
6. Ethernet Cable : Ethernet cable is used to connect devices in a wired network, enabling data communication, and it comes in different categories for various performance levels.
7. Power Supply : A power supply converts incoming electrical energy into the right form to power electronic device.
8. Female to Female Jumper Wire : A female-to-female jumper wire in IoT is a connector cable with female connectors on both ends, used to link components or sensors with female pins or headers.

Steps :

1. Connect the Pins of the GPS to raspberry Pi by using female to female jumper wire.
Follow the ports :

- VCC - Pin 4
- GND - Pin 6
- RX - Pin 8
- TX - Pin 1

Terminal Commands :

sudo raspi-config

- Then go to interface option and enable serial port and finish.
- Install gpsd and the gpsd-client :

sudo apt-get install gpsd gpsd-clients

- Once the installation is done, verify that you can receive data from the GPS module. To do that, output the data that it sends over the serial port:

cat /dev/serial0

- If it goes in loop so do Ctrl+C :
- Note that you should be able to run this command without being a superuser. If you can't, add the pi-user to the dialout group:

sudo systemctl stop gpsd.socket

- Note that you'll have to type this command every time you boot up the system. Alternatively, you can also disable it entirely:

sudo systemctl disable gpsd.socket

- Start a new gpsd instance that redirects the data of the correct serial port to a socket:

sudo gpsd /dev/serial0 -F /var/run/gpsd.sock

- And then you can run either of the following two commands to display the GPS data:

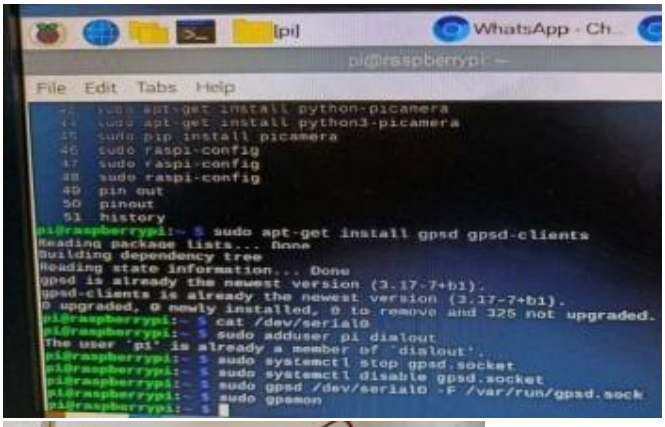
sudo gpsmon
or
sudo cgps –s

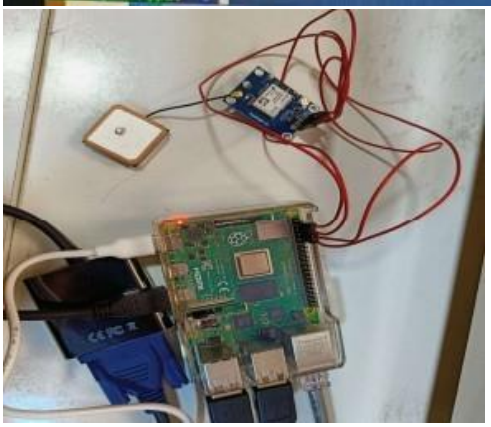
2. Now it will show your latitude and longitude of your current locations on the screen.

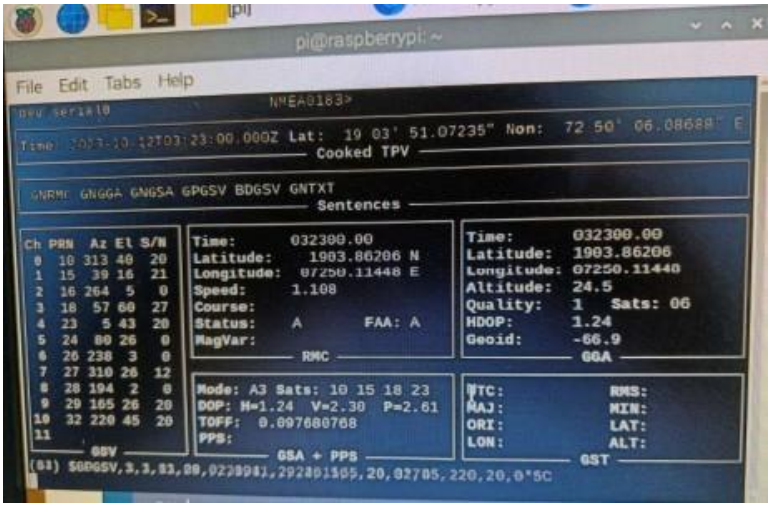
Output :

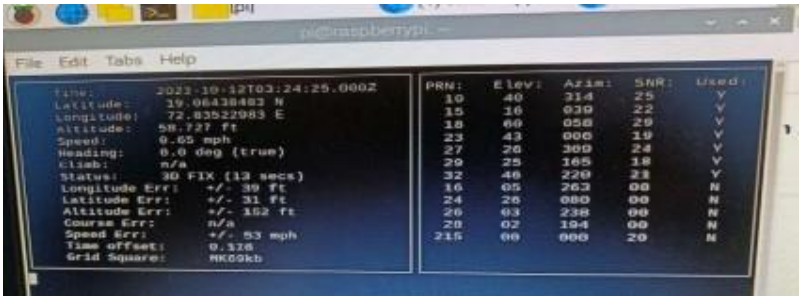
Name- Suraj muresh Vishwakarma
Class: TYBSCIT

Roll No: IT22096
Subject: Internet of things
Date: / /









Name- Suraj muresh Vishwakarma
Class: TYBSCIT

Roll No: IT22096
Subject: Internet of things
Date: / /

Practical 9.

Aim : IoT based Web Controlled Home Automation Using Raspberry Pi

Hardware Requirements :

1. Raspberry Pi Model B/B+ 5v Relays
2. LEDs to test
3. Breadboard
4. AC lamp to Test
5. Jumper wires

Software Requirements :

1. Raspbian Stretch OS
2. WebIOPi frame work

Connect your 5v Relay with Raspberry Pi's GPIO Pins using Jumper wires (Female Female)

Relay Board Pin	Function	RPI Physical Pin	Raspberry Function
+5v	+ 5V Power	4	5V
I/P	Data In	7	GPIO 4
GND	Ground	6	GND

LED to Test : Connect LED to Raspberry Pi using breadboard

LED Terminal	RPI Physical Pin	LED Terminal	RPI Physical Pin
Positive	37	Negative	39

Steps :

1. Download the WebIOPi Framework file

Use wget command to get the installer file of WebIOPi framework from sourceforge page. Make sure you are in home directory.

Name- Suraj muresh Vishwakarma
Class: TYBSCIT

Roll No: IT22096
Subject: Internet of things
Date: / /

Extract the file using tar command.

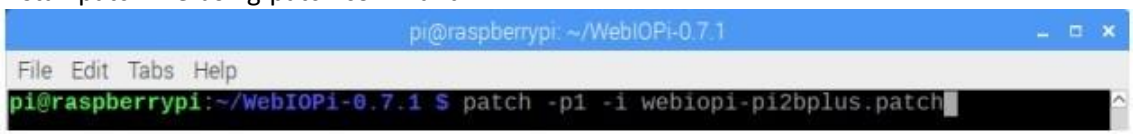
Now, Go to the WebIOPi Directory.

2. Install patch file

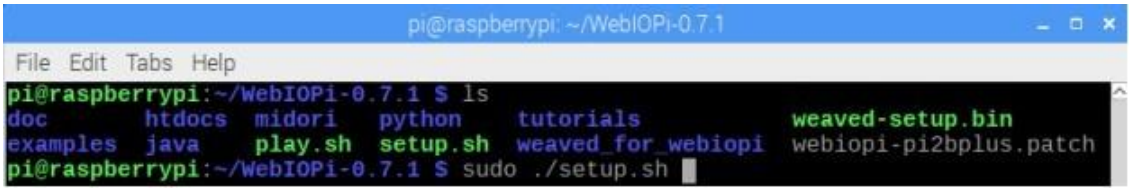
At this point before running the setup, we need to install a patch as this version of the WebIOPi does not work with the raspberry pi 3.

Download patch file using wget command Install patch file using patch command.

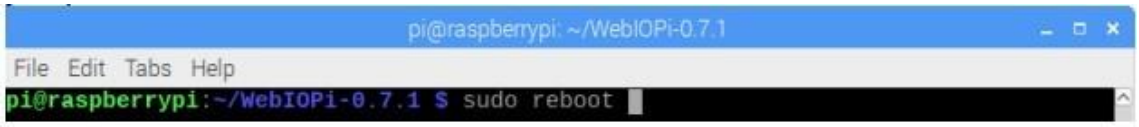
Install patch file using patch command.



3. Install setup of WebIOPi framework, run setup file.



Keep saying yes if asked to install any dependencies during setup installation. When done, reboot your pi.



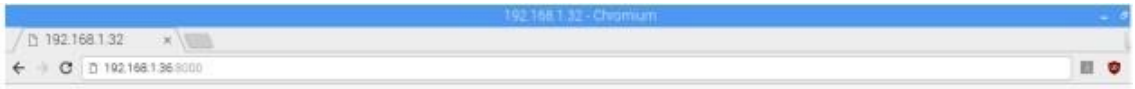
Name- Suraj muresh Vishwakarma
Class: TYBSCIT

Roll No: IT22096
Subject: Internet of things
Date: / /

4: Test WebIOPi Installation

We will need to test our WebIOPi installation to be sure everything works fine as desired. Run following command on terminal.

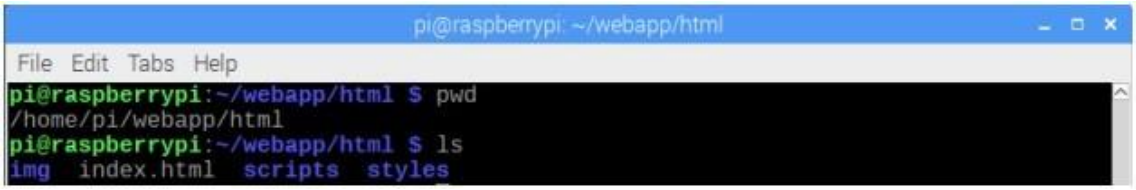
Now, open web browser and connect to http://PI's IP address:8000



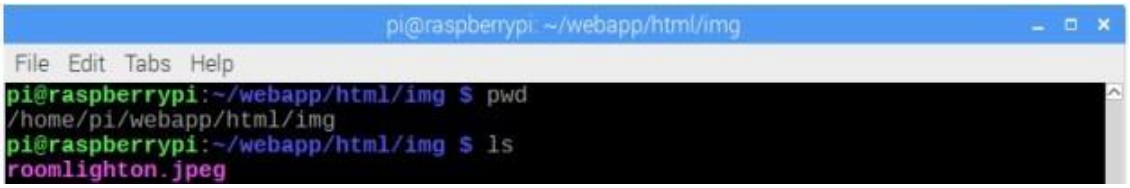
The system will prompt you for username and password.
Username: webiopi
Password: raspberry

After the login, click on the GPIO header link. Test your LED which is connected to raspberry Pi's GPIO Pins. In my case, I have used Physical Pin no 37 of Pi. So set it as output. Click the pin 37 button to turn on or off the LED. This way we can control the Raspberry Pi GPIO using WebIOPi.
After the test, if everything worked as described, then we can go back to the terminal and stop the program with CTRL + C.

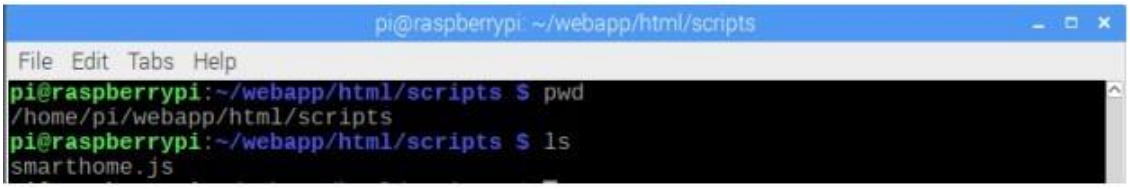
Now, building the Web Application for Raspberry Pi Home Automation Create below directory structure.



Place image inside img folder.



Create smarthome.js file inside scripts folder.



Smarthome.js file

```
webiopi().ready(function() {
                webiopi().setFunction(4,"out");
                var content, button;
                content = $("#content");

                button = webiopi().createGPIOButton(4,"Room 1");
                content.append(button);
});
```

Create smarthome.css file inside styles folder.

```
body {
                        background-color:#fff;
                        background-repeat:no-repeat;
                        background-position:center;
                        background-size:cover;
                        font: bold 18px/25px Arial, sans-serif;

        }

h1 {
font: bold 40px Arial, sans-serif;
background-color:#000;
color:white;
}

button {
                        display: block;
                        position: absolute;
                        margin: 40px 610px;
                        padding: 0 10px;
                        text-align: center;
                        text-decoration: none;
                        width: 130px;
                        height: 40px;
                        font: bold 18px/25px Arial, sans-serif;
                        color: black;

                        text-shadow: 1px 1px 1px rgba(255,255,255, .22);
                        -webkit-border-radius: 30px;
                        -moz-border-radius: 30px;
                        border-radius: 30px;

-webkit-box-shadow: 1px 1px 1px rgba(0,0,0, .29), inset 1px 1px 1px rgba(255,255,255,
.44);
 -moz-box-shadow: 1px 1px 1px rgba(0,0,0, .29), inset 1px 1px 1px rgba(255,255,255, .44);
   box-shadow: 1px 1px 1px rgba(0,0,0, .29), inset 1px 1px 1px rgba(255,255,255, .44);

                        -webkit-transition: all 0.15s ease;
                        -moz-transition: all 0.15s ease;
                        -o-transition: all 0.15s ease;
                        -ms-transition: all 0.15s ease;
                        transition: all 0.15s ease;
}

input[type="range"] {
                                display: block;
                                width: 160px;
                                height: 45px;
}

#gpio4.LOW {
                                background-color: White;
                                color: Black;
        }
#gpio4.HIGH {
                                background-color: Black;
                                color: LightGray;
        }
```

Name- Suraj muresh Vishwakarma
Class: TYBSCIT
Roll No: IT22096
Subject: Internet of things
Date: / /

Create index.html file inside html folder.

```
<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-scalable = no" />
    <title>Smart Home</title>
    <script type="text/javascript" src="/webiopi.js"></script>
    <script type="text/javascript" src="/scripts/smarthome.js"></script>
    <link rel="stylesheet" type="text/css" href="/styles/smarthome.css">
    <link rel="shortcut icon" sizes="196x196" href="/img/roomlighton.jpeg" />
</head>
<body>
<h1 align=center>Web Based Controlled Home Automation using Raspberry Pi </h1>

<div id="content" align="center"></div>
        <center><img src="/img/roomlighton.jpeg" height="500px" width="800px"
id='pic1' /></center>

</body>
</html>
```

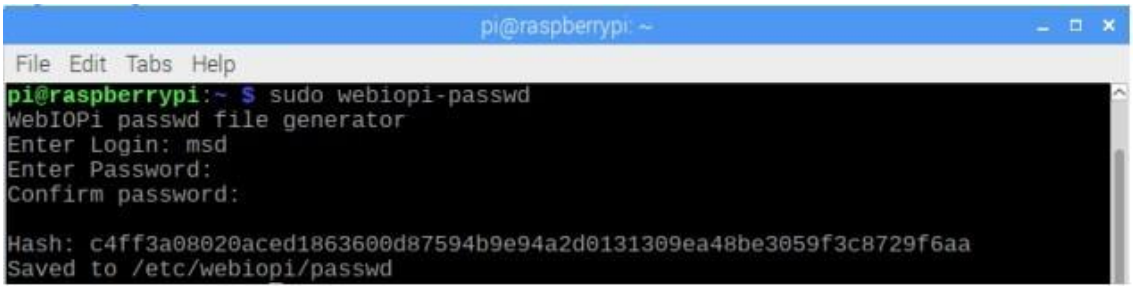
5. WebIOPi Server Edits for Home Automation.

We need to edit the config file of the webiopi service so it's pointed to use data from our html folder instead of the config files that came with it.

Under http section of the config file, comment out doc-root line and change the path to your project file.

6. Change the username & password of the WebIOPi service.

Note that you can change the password of the WebIOPi service using the command if you want. Or skip this step.



7. Run the WebIOPi service.

Start the WebIOPi service.

Name- Suraj muresh Vishwakarma
Class: TYBSCIT

Roll No: IT22096
Subject: Internet of things
Date: / /

```
pi@raspberrypi: ~
File Edit Tabs Help
pi@raspberrypi:~ $ sudo /etc/init.d/webiopi start
```

Restart the WebIOPi service.

```
pi@raspberrypi: ~
File Edit Tabs Help
pi@raspberrypi:~ $ sudo /etc/init.d/webiopi restart
```

Status the WebIOPi service.

```
pi@raspberrypi: ~
File Edit Tabs Help
pi@raspberrypi:~ $ sudo /etc/init.d/webiopi status
```

Stop the WebIOPi service.

```
pi@raspberrypi: ~
File Edit Tabs Help
pi@raspberrypi:~ $ sudo /etc/init.d/webiopi stop
```

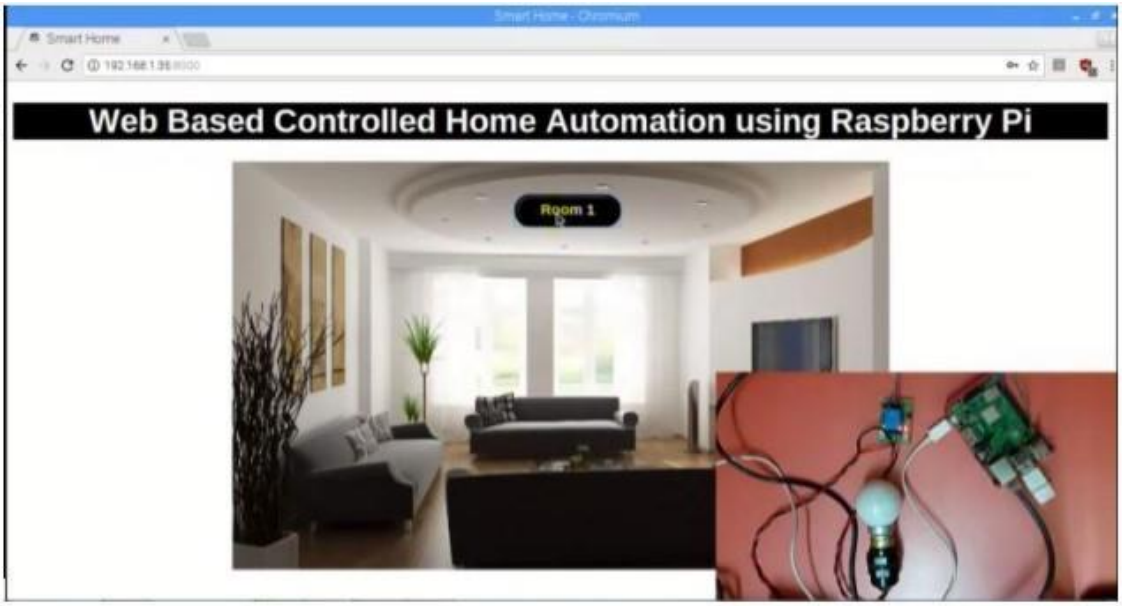
To setup WebIOPi to run at boot, use:

```
pi@raspberrypi: ~
File Edit Tabs Help
pi@raspberrypi:~ $ sudo update-rc.d webiopi defaults
```

To reverse and stop it from running at boot, use :

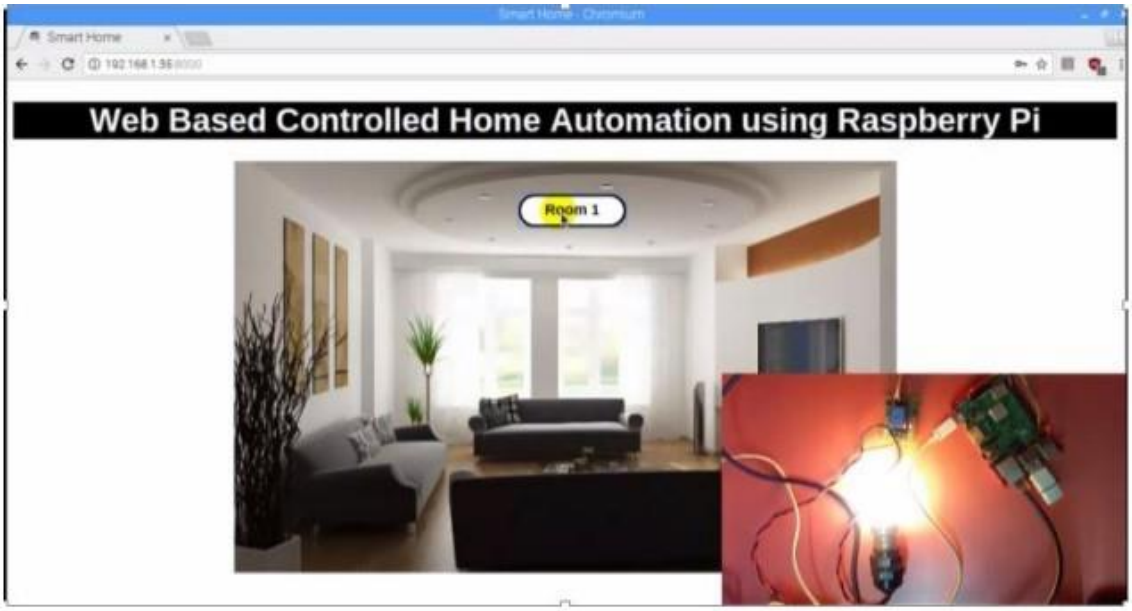
```
pi@raspberrypi: ~
File Edit Tabs Help
pi@raspberrypi:~ $ sudo update-rc.d webiopi remove
```

When you Click on button Room1,it switch on AC Lamp, again clicking on same button,it will switch off AC Lamp.



Name- Suraj muresh Vishwakarma
Class: TYBSCIT

Roll No: IT22096
Subject: Internet of things
Date: / /



Note : It is possible to open same URL http://PI's IP address:8000 on smart phone , tablet over local network.

Name- Suraj muresh
Vishwakarma
Class: TYBSCIT

Roll No: IT22096
Subject: Internet of things
Date: / /

Practical 10.

Aim : Click image and video using Raspberry Pi

Hardware Requirements:

1. **Camera** : It will automatically record, monitor and alert the user.
2. **Raspberry Pi** : A Raspberry Pi is a small, affordable, singleboard computer that's widely used for various DIY projects, education, and prototyping. It can run a variety of operating systems and is popular for programming, learning about computing, and building innovative electronics projects.
3. **Keyboard** : A keyboard is an input device used to type or enter text and commands into a computer or other digital devices by pressing individual keys, each representing a specific character or function.
4. **Mouse** : A mouse is a pointing device used to control the movement of a cursor or pointer on a computer screen. It typically has buttons that allow users to interact with graphical user interfaces, select items, and perform actions by clicking or dragging the cursor.
5. **HDMI Cable** : HDMI (High-Definition Multimedia Interface) cables are digital cables used to transmit high-quality audio and video signals between devices, such as TVs, monitors, projectors, and multimedia sources (like laptops, game consoles, or streaming devices).
6. **Ethernet Cable** : Ethernet cable is used to connect devices in a wired network, enabling data communication, and it comes in different categories for various performance levels.
7. **Power Supply**: A power supply converts incoming electrical energy into the right form to power electronic device.

Steps :

1. Connect Pi Camera to CSI interface of Raspberry Pi board as shown below :

Name- Suraj muresh Vishwakarma
Class: TYBSCIT

Roll No: IT22096
Subject: Internet of things
Date: / /

2. Now, we can use Pi Camera for capturing images and videos using Raspberry Pi.
3. Now turn on your Raspberry pi.
4. Before using Pi Camera, we need to enable camera for its working.
5. For enabling camera in Raspberry Pi, open raspberry pi configuration using following command :

Terminal Command :

sudo raspi-config

6. Then go to interface option, click on camera and enable it.

Code :

img.py

```
import time
from picamera import PiCamera
camera = PiCamera()
camera.resolution = (1280,720)
camera.start_preview()
time.sleep(5)
camera.capture('home/pi/Desktop/ty2.jpeg')
camera.stop_preview()
```

vid.py

```
import time
from picamera import PiCamera
camera = PiCamera()
camera.start_preview()
camera.start_recording("home/pi/Desktop/video1.h264")
camera.wait_recording(5)
camera.stop_recording()
print("finished  Recording")
```

Name- Suraj muresh Vishwakarma
Class: TYBSCIT

Roll No: IT22096
Subject: Internet of things
Date: / /

Note : You can also directly run the camera by using this command -

Output :

Name- Suraj muresh Vishwakarma
Class: TYBSCIT

Roll No: IT22096
Subject: Internet of things
Date: / /

Practical 11

Aim : Interfacing Raspberry Pi with RFID

Hardware Requirements :

1. RFID Tag : RFID tags are a type of tracking system that uses radio frequency to search, identify, track, and communicate with items and people. Essentially, RFID tags are smart labels that can store a range of information from serial numbers, to a short description, and even pages of data.
2. Raspberry Pi : A Raspberry Pi is a small, affordable, singleboard computer that's widely used for various DIY projects, education, and prototyping. It can run a variety of operating systems and is popular for programming, learning about computing, and building innovative electronics projects.
3. Keyboard : A keyboard is an input device used to type or enter text and commands into a computer or other digital devices by pressing individual keys, each representing a specific character or function.
4. Mouse : A mouse is a pointing device used to control the movement of a cursor or pointer on a computer screen. It typically has buttons that allow users to interact with graphical user interfaces, select items, and perform actions byclicking or dragging the cursor.
5. HDMI Cable : HDMI (High-Definition Multimedia Interface) cables are digital cables used to transmit high-quality audio and video signals between devices, such as TVs, monitors, projectors, and multimedia sources (like laptops, game consoles, or streaming devices).
6. Ethernet Cable : Ethernet cable is used to connect devices in a wired network, enabling data communication, and it comes in different categories for various performance levels.
7. Power Supply: A power supply converts incoming electrical energy into the right form to power electronic device.
8. Female to Female Jumper Wire : A female-to-female jumper wire in IoT is a connector cable with female connectors on both ends,used to link components or sensors with female pins or headers.

Steps :

read.py

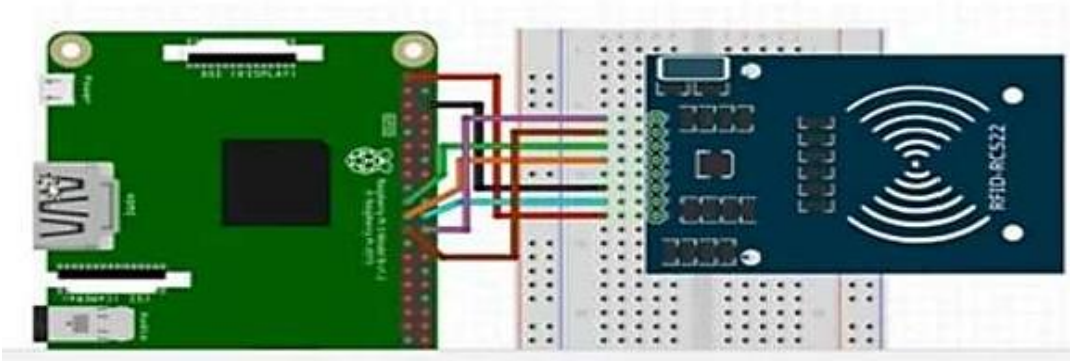
```
import RPi.GPIO as GPIO
from mfrc522 import SimpleMFRC522
reader = SimpleMFRC522()
try:
   print("Place your card:")
   id, text = reader.read()
   print(id)
   print(text)
```

```
finally:
   GPIO.cleanup()
```

write.py

```
import RPi.GPIO as GPIO
from mfrc522 import SimpleMFRC522
reader = SimpleMFRC522()
try:
   text = input('New data:')
   print("Now place your tag to write.")
   reader.write(text)
   print("written")
finally:
   GPIO.cleanup()
```

- **SDA** connects to **Pin 24**.
- **SCK** connects to **Pin 23**.
- **MOSI** connects to **Pin 19**.
- **MISO** connects to **Pin 21**.
- **IRQ : Not required**
- **GND** connects to **Pin 6**.
- **RST** connects to **Pin 22**.
- **3.3v** connects to **Pin 1**.



Terminal Commands :

sudo nano writetest.py

- Copy and paste the write.py code
- Ctrl+O + Enter -> to save the code
- Ctrl + x -> to exit

sudo pip3 install mfrc522

sudo raspi-config

- After that go to interfacing option and enable the SPI and finish.

sudo nano read.py

- Copy and paste the read.py
- Ctrl+O + Enter to save the code
- Ctrl + x -> to exit

sudo python3 writetest.py
new dataMegaBytes

Place your tag and card to read.

Now place your card or tag on the sensor for output.

Output :