UNIVERSAL

VIKAS VIDYA EDUCATION TRUST'S

Lords Universal College

Department of Information Technology

CERTIFICATE

Class: __________ **Year: _________**

This is to certify that the work entered in this Journal is the work of

Shree / Kumari:____

Of________________________________ Roll no:_________________________

University Exam No: _____________ has satisfactorily completed the required number of practical and worked for the 5th term the term of the year 20__20__ in the college laboratory as laid down by the university.

______________________ ________________ ________________

Head of the Department **External Examiner** **Internal Examiner**
Subject teacher

Date: / / 20 **Department of Bsc.IT**

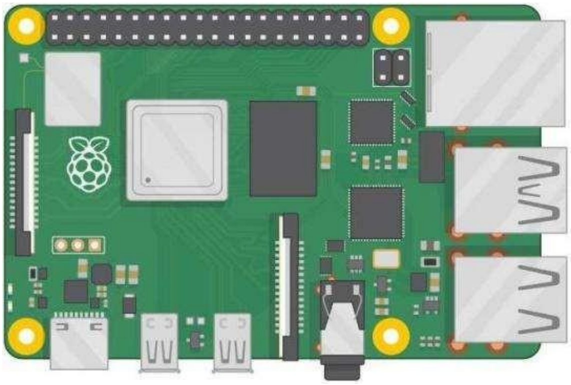
INDEX

Practical 0

Aim: Starting Raspbian OS, familiarizing with raspberry pi components and Interface, connecting to ethernet, monitor, USB.

Step 1 - What you will need

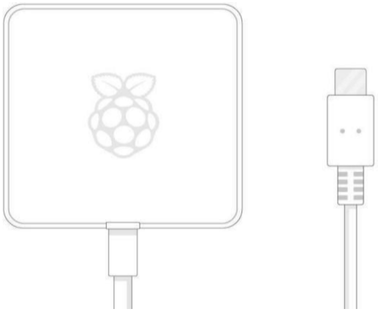
- Which Raspberry Pi?
- There are several models of Raspberry Pi (https://www.raspberrypi.org /products/), and for most people Raspberry Pi 4 Model B is the one to choose. Raspberry Pi 4 Model B is the newest, fastest, and easiest to use.
- Raspberry Pi 4 comes with 2GB, 4GB, or 8GB of RAM. For most 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).

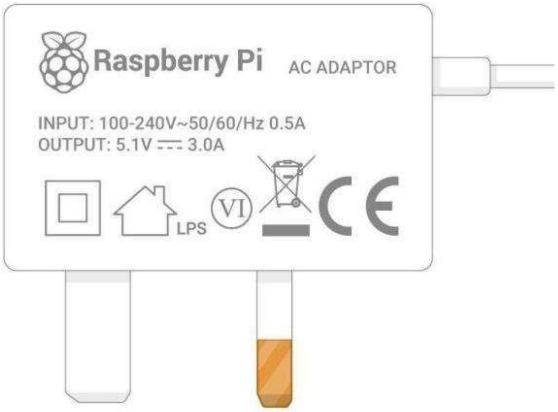
A power supply

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



You need a power supply that provides:

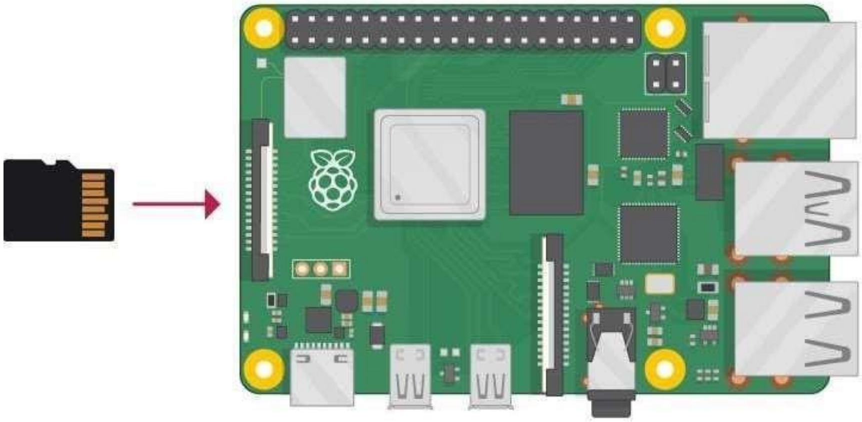
- At least 3.0 amps for Raspberry Pi 4



- At least 2.5 amps for Raspberry Pi 3

INPUT: 100-240V ~ 50/60Hz 0.3A
OUTPUT: +5V ⎓ 2.5A
FC
CE

- A micro SD card
- Your Raspberry Pi needs an SD card to store all its files and the Raspberry Pi OS operating system.



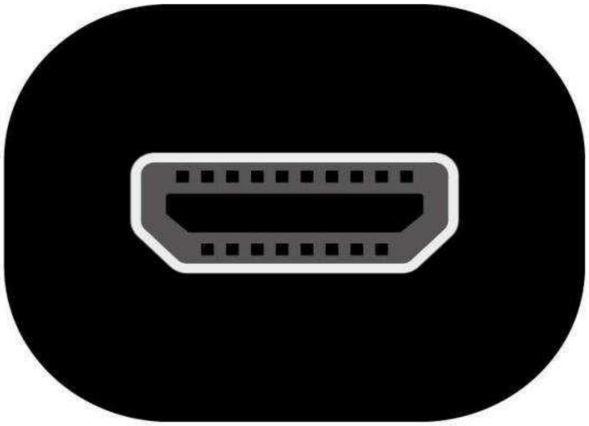
- You need a micro SD card with a capacity of at least 8GB.
- Many sellers supply SD cards for Raspberry Pi that are already set up with Raspberry Pi OS and ready to go.
- A keyboard and a mouse
- To start using your Raspberry Pi, you need a USB keyboard and a USB mouse.
- Once you’ve set up your Raspberry 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 sound.

HDMI

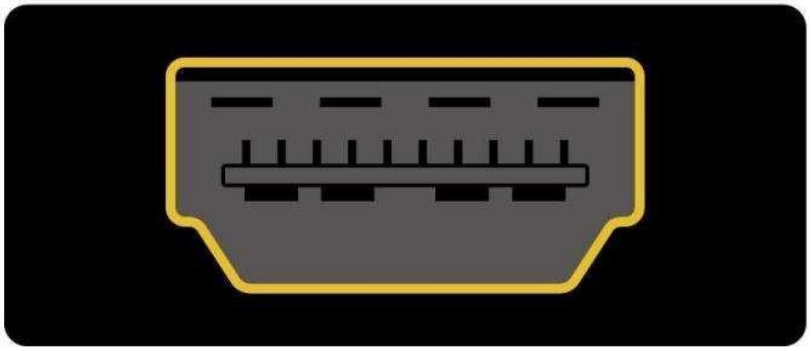
Your Raspberry Pi has an HDMI output port that is compatible with the HDMI port of most modern TVs and computer monitors. Many computer monitors may also have DVI or VGA ports.

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



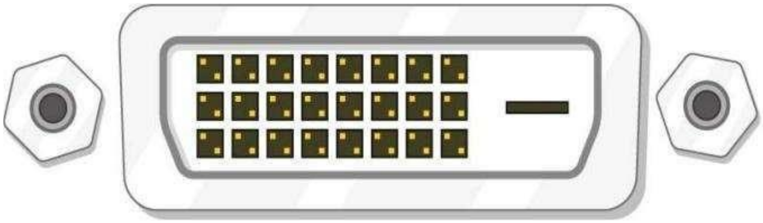
You need either a micro HDMI to HDMI cable. or a standard HDMI to HDMI cable plus a micro HDMI to HDMI adapter, to connect Raspberry Pi 4 to a screen.

Raspberry Pi1, 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.



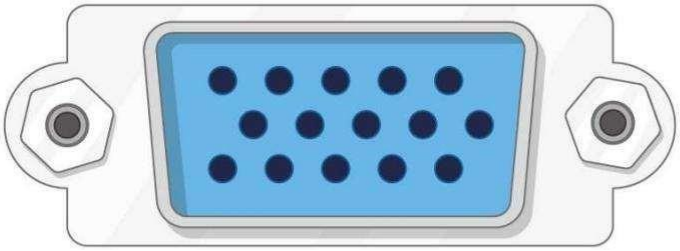
DVI

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

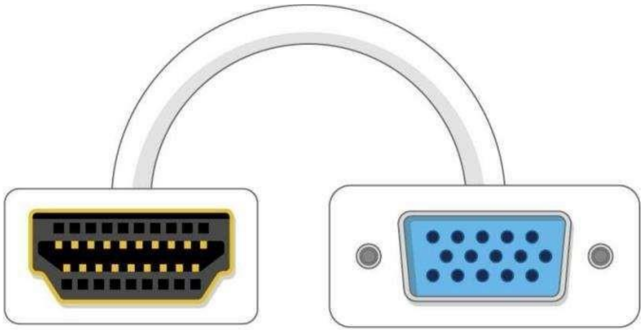


VGA

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 you're 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.

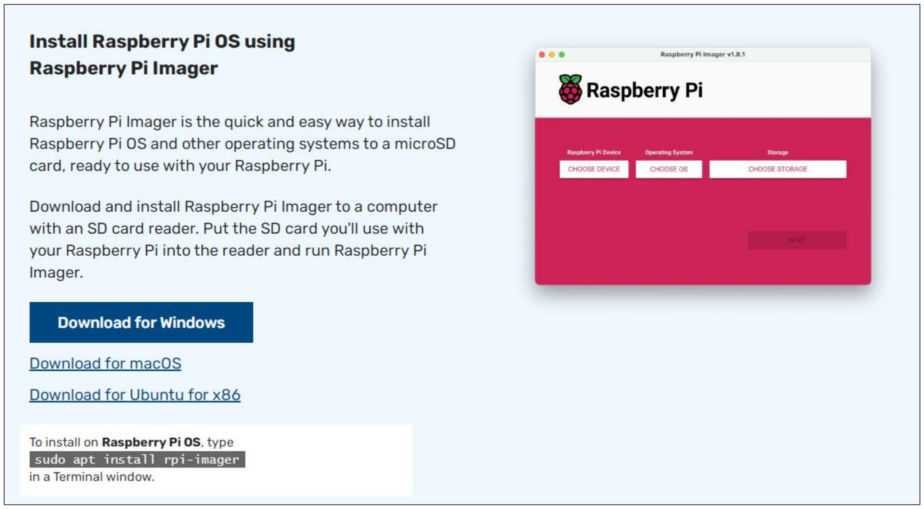
The Raspberry Pi OS operating system via the Raspberry Pi Imager

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.com/documentation/computers/getting-started.html)

Download and launch the Raspberry Pi Imager

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



Install Raspberry Pi OS using Raspberry Pi Imager

Raspberry Pi Imager is the quick and easy way to install Raspberry Pi OS and other operating systems to a microSD card, ready to use with your Raspberry Pi.

Download and install Raspberry Pi Imager to a computer with an SD card reader. Put the SD card you'll use with your Raspberry Pi into the reader and run Raspberry Pi Imager.

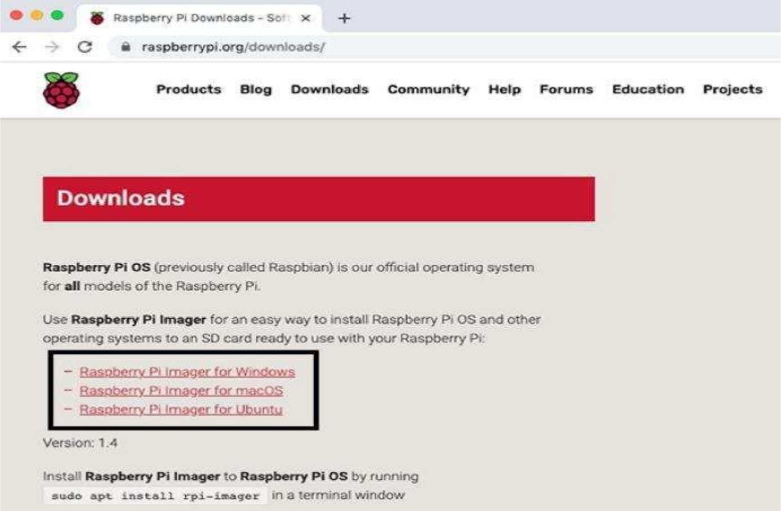
Download for Windows

Download for macOS

Download for Ubuntu for x86

To install on **Raspberry Pi OS**, type `sudo apt install rpi-imager` in a Terminal window.

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



Downloads

Raspberry Pi OS (previously called Raspbian) is our official operating system for **all** models of the Raspberry Pi.

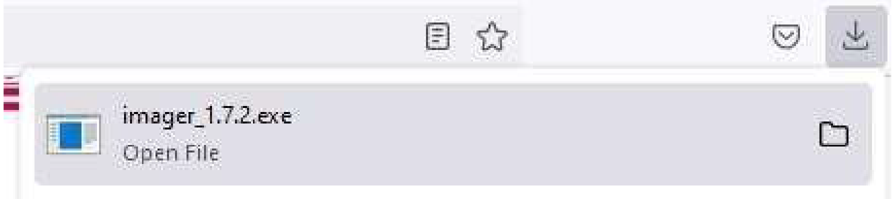
Use **Raspberry Pi Imager** for an easy way to install Raspberry Pi OS and other operating systems to an SD card ready to use with your Raspberry Pi:

- Raspberry Pi Imager for Windows
- Raspberry Pi Imager for macOS
- Raspberry Pi Imager for Ubuntu

Version: 1.4

Install **Raspberry Pi Imager** to **Raspberry Pi OS** by running `sudo apt install rpi-imager` in a terminal window

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



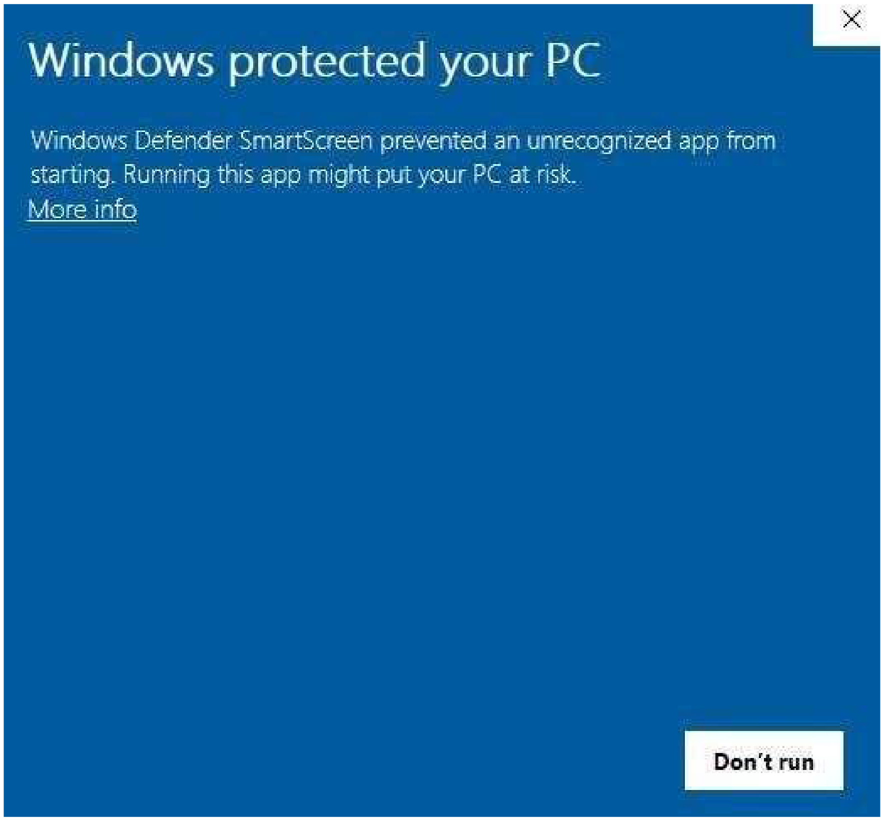
imager_1.7.2.exe

Open File

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:



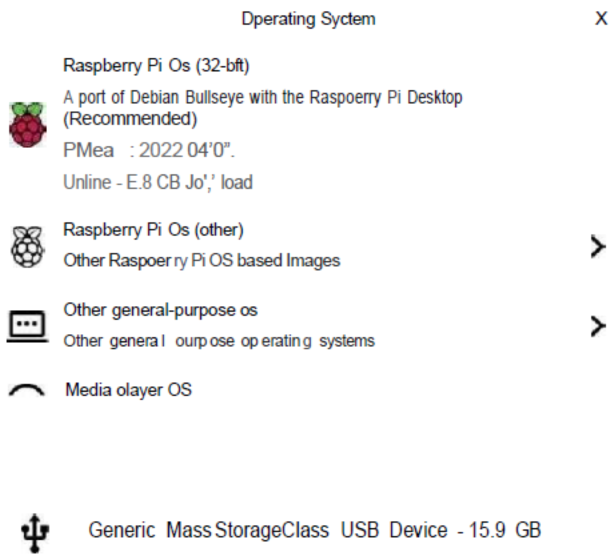
Windows protected your PC

Windows Defender SmartScreen prevented an unrecognized app from starting. Running this app might put your PC at risk.

More info

- If this pops up click on and then Run away
- 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 the Raspberry Pi Imager to download the OS that you choose. That OS will then be stored for future of in use. Being online for later uses means that the Raspberry Pi imager will always give you the latest version.



Dperating Syctem X

Raspberry Pi Os (32-bft)

A port of Debian Bullseye with the Raspoerry Pi Desktop (Recommended)

PMea : 2022 04'0".

Unline - E.8 CB Jo',' load

Raspberry Pi Os (other)

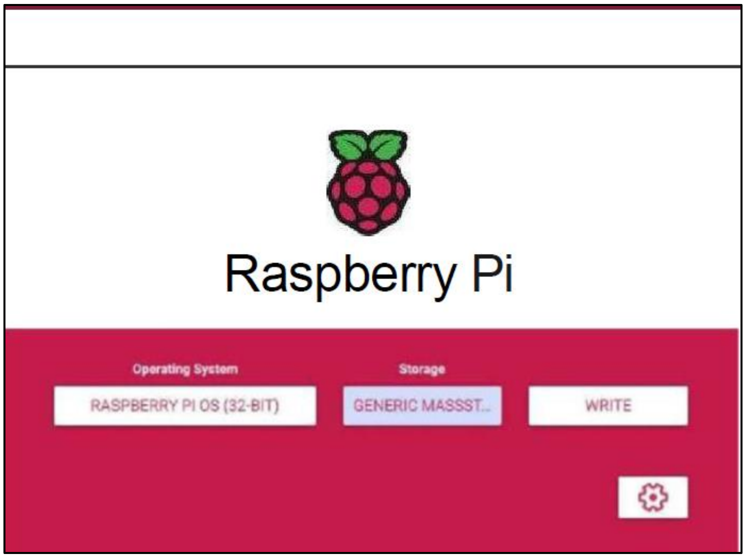
Other Raspoer ry Pi OS based Images

Other general-purpose os

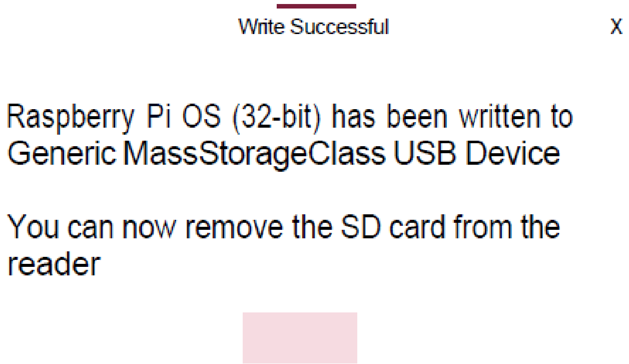
Other genera l ourp ose op erating systems

Media olayer OS

Generic Mass StorageClass USB Device - 15.9 GB



- 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.



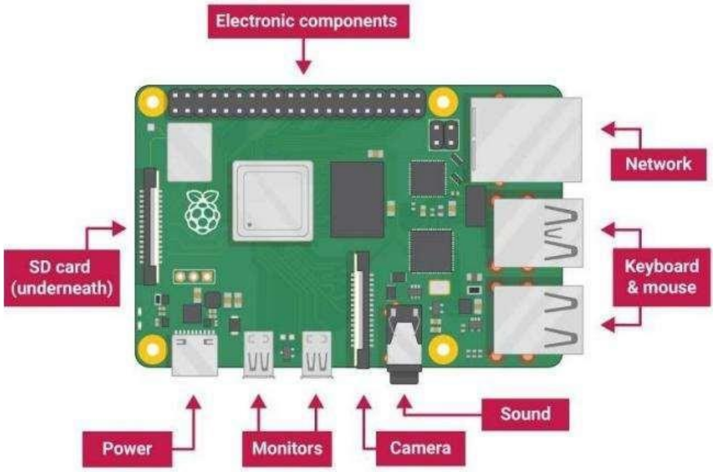
Write Successful X

Raspberry Pi OS (32-bit) has been written to Generic MassStorageClass USB Device

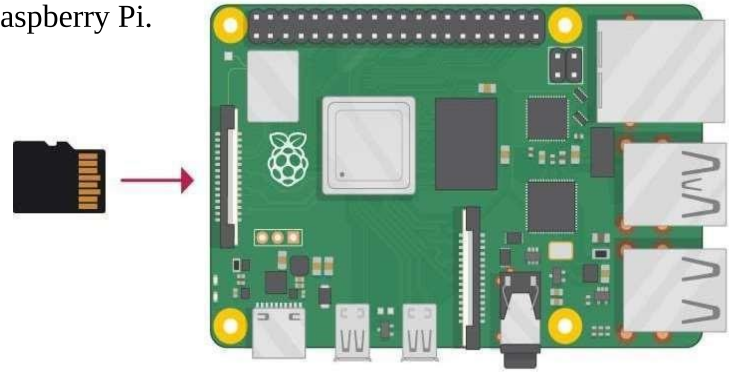
You can now remove the SD card from the reader

Step 3 Connect your Raspberry Pi

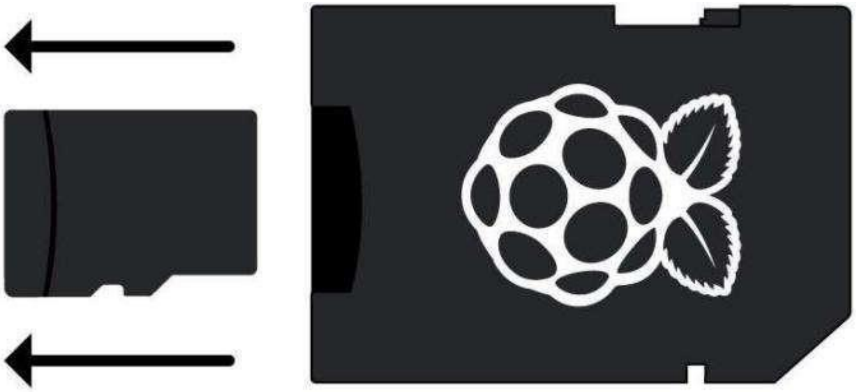
Now get everything connected to your Raspberry Pi. It’s important to do this in the right order, so that all your components are safe.



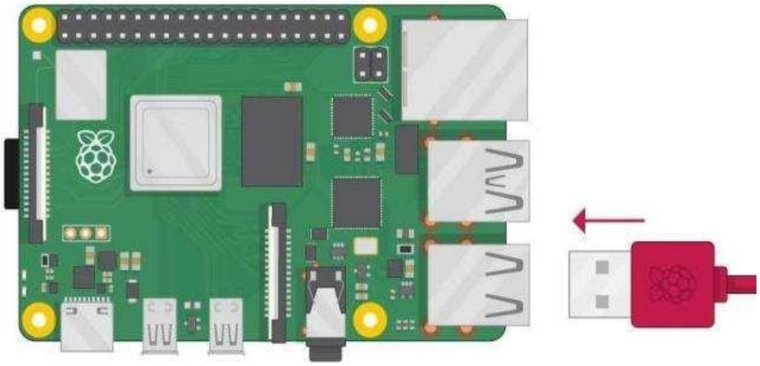
- Insert the SD card you’ve set up with Raspberry Pi OS into the micro SD card slot on the underside of your Raspberry Pi.



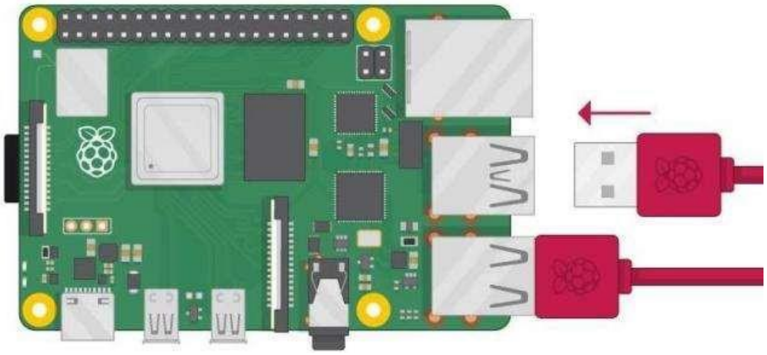
Note: Many micro SD cards come inside a larger adapter — you can slide the smaller card out using the lip at the bottom.



- Find the USB connector end of your mouse‘s cable. and connect the mouse to a USB port on Raspberry Pi (it doesn‘t matter which port you use).



- Connect the keyboard in the same way.

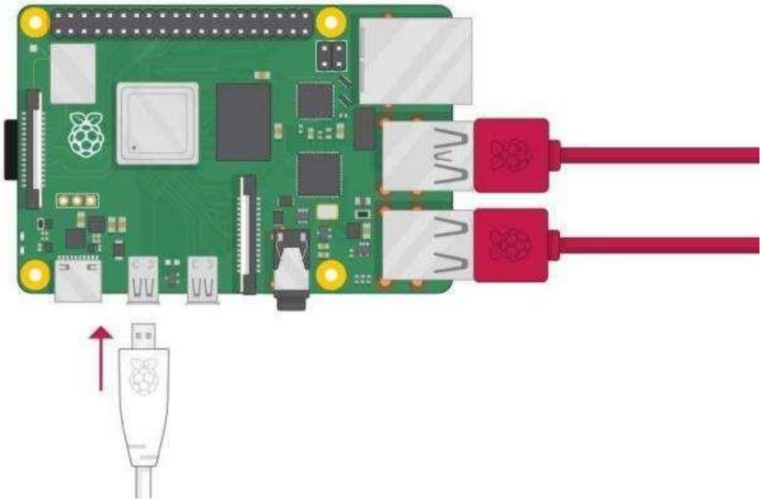


- Make sure your screen is plugged into a wall socket and switched on.
- Look at the HDMI port(s) on your Raspberry Pi - notice that they have a flat side on top.
- Use a cable to connect the screen to Raspberry Pi’s HDMI port - use an adapter if necessary.

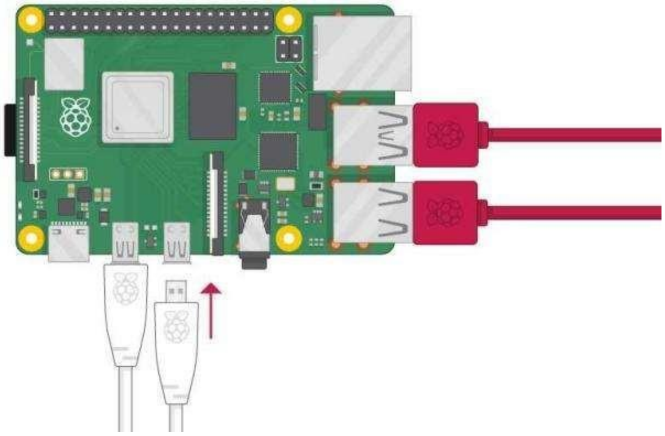
Raspberry Pi 4

Connect your screen to the first of Raspberry Pi 4’s HDMI ports. labeled HDMIO.

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

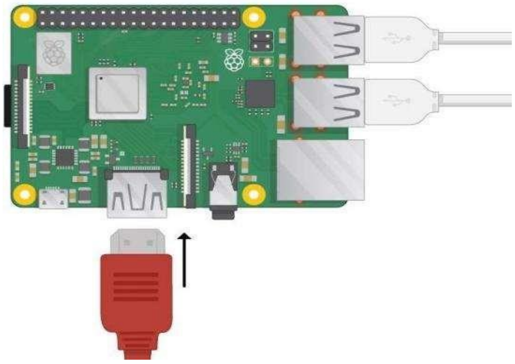


You can connect an optional second screen in the same way.



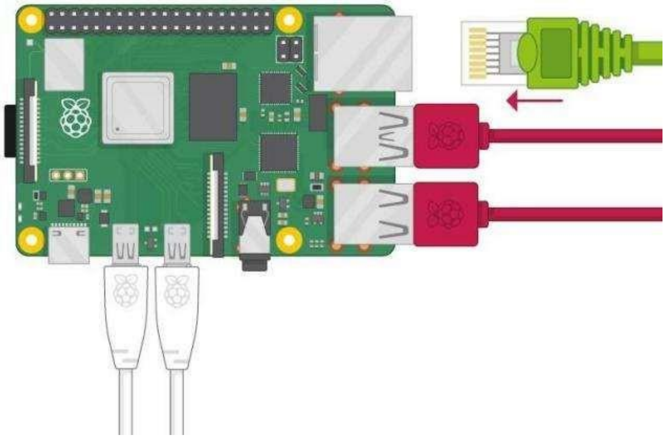
Raspberry Pi 1, 2, 3

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.

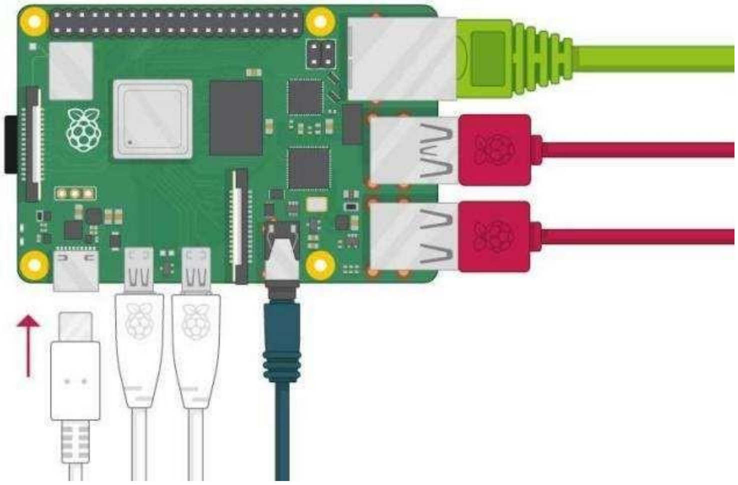


- If the screen you are using has speakers. sound will play through those. Alternatively. connect headphones or speakers to the audio port if you prefer

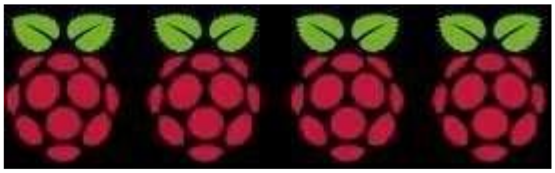
Step 4 Start up your Raspberry Pi

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

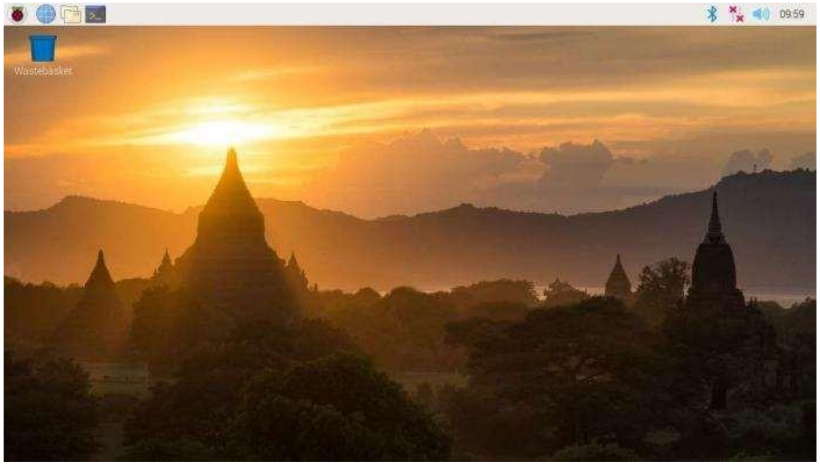
- Plug the power supply into a socket and connect it to your Raspberry Pi‘s power port.



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.

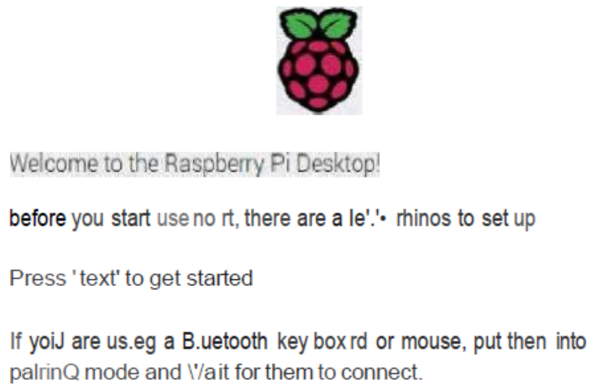


After a few seconds the Raspberry Pi OS desktop will appear.



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.



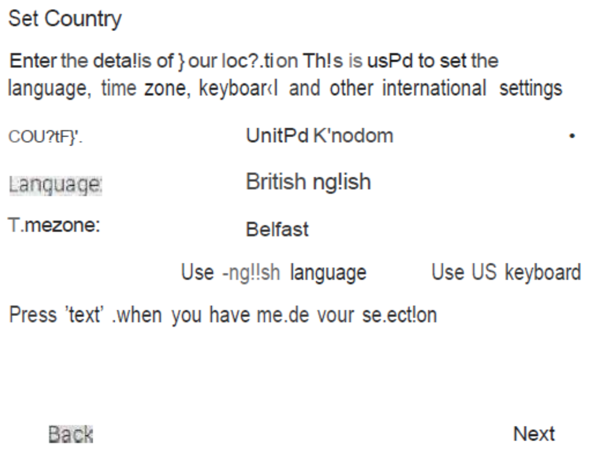
Welcome to the Raspberry Pi Desktop!

before you start use no rt, there are a le'.'• rhinos to set up

Press ' text' to get started

If yoiJ are us.eg a B.uetooth key box rd or mouse, put then into palrinQ mode and \'/ait for them to connect.

- Click on Next to start the setup.
- Set your Country, Language. and Timezone, then click on Next again.



Set Country

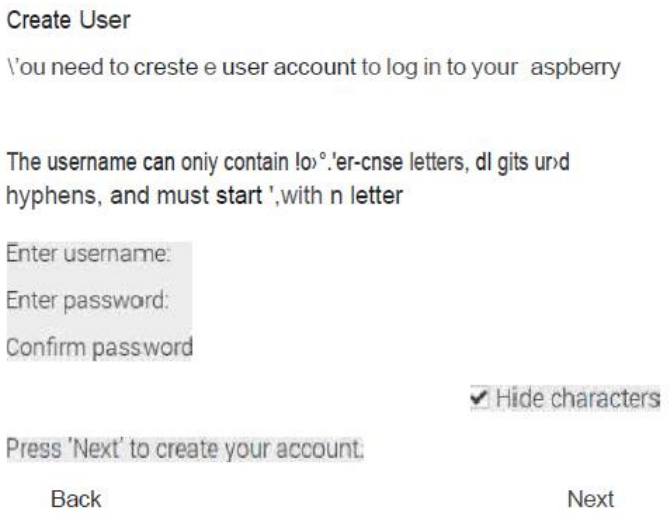
Enter the deta!is of } our loc?.ti on Th!s is usPd to set the language, time zone, keyboar‹l and other international settings

COU?tF}'.	UnitPd K'nodom •
Lanquage:	British ng!ish
T.mezone:	Belfast

Use -ng!!sh language Use US keyboard

Press 'text' .when you have me.de vour se.ect!on

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



Create User

\'ou need to creste e user account to log in to your aspberry

The username can oniy contain !o›°.'er-cnse letters, dl gits ur›d hyphens, and must start ',with n letter

Enter username:

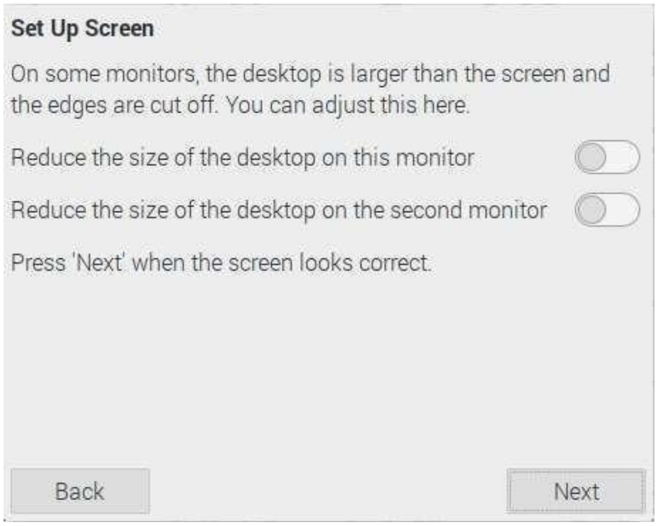
Enter password:

Confirm password

✔ Hide characters

Press 'Next' to create your account.

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



Set Up Screen

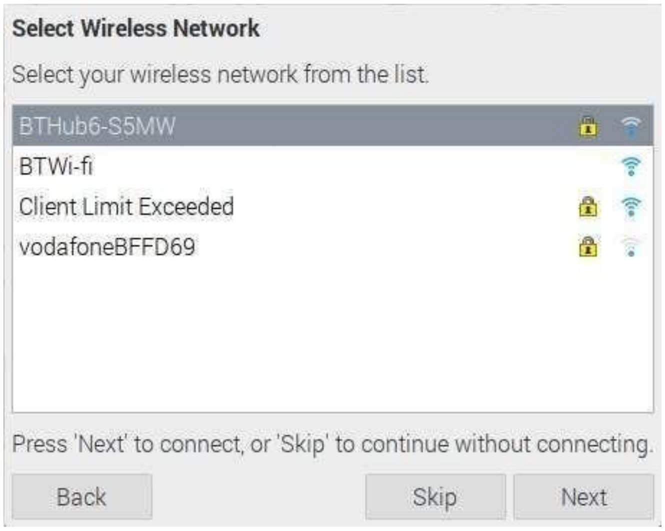
On some monitors, the desktop is larger than the screen and the edges are cut off. You can adjust this here.

Reduce the size of the desktop on this monitor ○

Reduce the size of the desktop on the second monitor ○

Press 'Next' when the screen looks correct.

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



Select Wireless Network

Select your wireless network from the list.

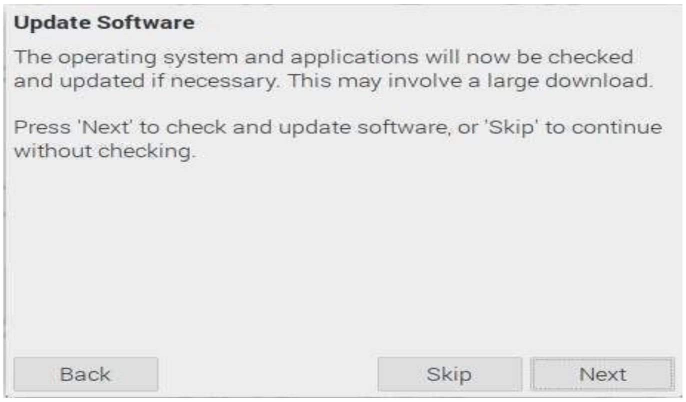
BTHub6-S5MW
BTWi-fi
Client Limit Exceeded
vodafoneBFFD69

Press 'Next' to connect, or 'Skip' to continue without connecting.

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 (this might take a little while).



Update Software

The operating system and applications will now be checked and updated if necessary. This may involve a large download.

Press 'Next' to check and update software, or 'Skip' to continue without checking.

- Click on Restart to finish the setup.

Practical - 1

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 come 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.

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)

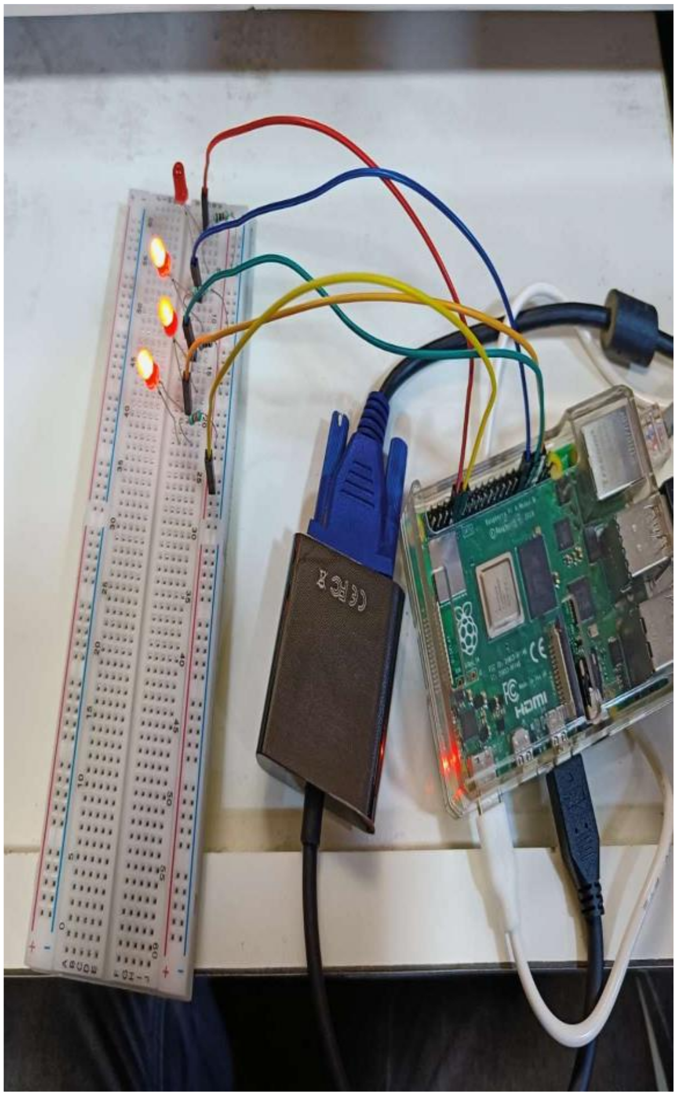
Code:

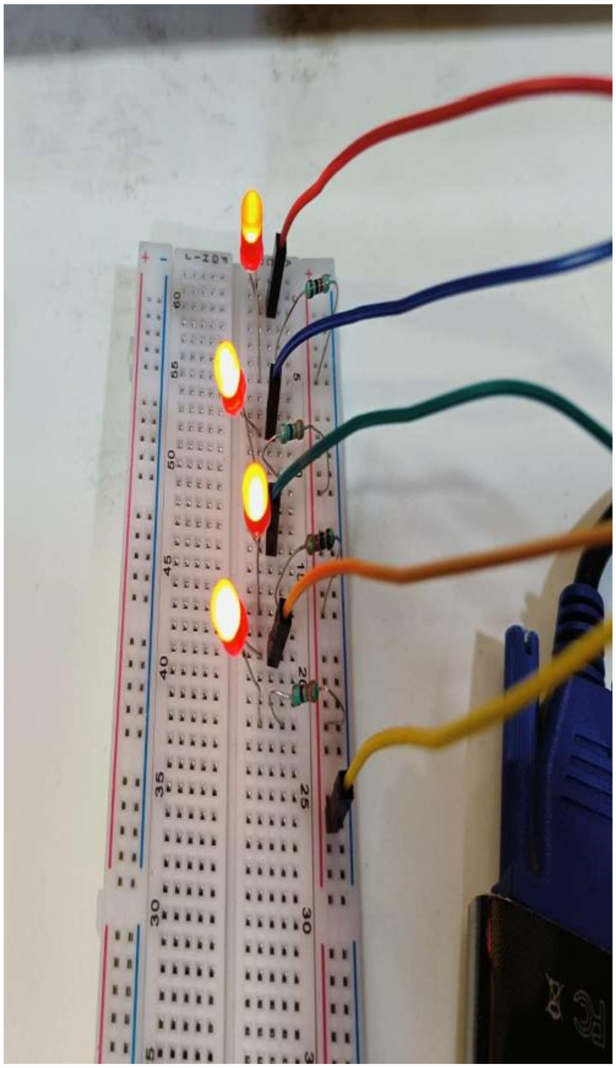
```
import RPi.GPIO as GPIO
import time
GPIO.setmode(GPIO.BOARD)
GPIO.setwarnings(False)
GPIO.setup(7, GPIO.OUT)
GPIO.setup(29, GPIO.OUT)
GPIO.setup(31, GPIO.OUT)
GPIO.setup(33, GPIO.OUT)\
try:
    while True:
```

```
GPIO.output(7, False)
print("LED 1 IS OFF")
time.sleep(1)
GPIO.output(29, False)
print("LED 2 IS OFF")
time.sleep(1.5)\
GPIO.output(31, False)
print("LED 3 IS OFF")
time.sleep(2)
GPIO.output(33, False)
print("LED 4 IS OFF")
time.sleep(2.5)
GPIO.output(7, True)
print("LED 1 IS FINALLY ON")
time.sleep(3.5)
GPIO.output(29, True)
print("LED 2 IS FINALLY ON")
time.sleep(4)
GPIO.output(31, True)
print("LED 3 IS FINALLY ON")
time.sleep(4.5)
GPIO.output(33, True)
print("LED 4 IS FINALLY ON")
time.sleep(5)
GPIO.output(7, False)
print("LED 1 IS OFF")
time.sleep(1)
GPIO.output(29, False)
print("LED 2 IS OFF")
time.sleep(1.5)
```

```
GPIO.output(31, False)
print("LED 3 IS OFF")
time.sleep(2)
GPIO.output(33, False)
print("LED 4 IS OFF")
time.sleep(2.5)
print("PROGRAM COMPLETE!")
GPIO.cleanup()
```

Output:





Practical - 2

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

Hardware Requirements:

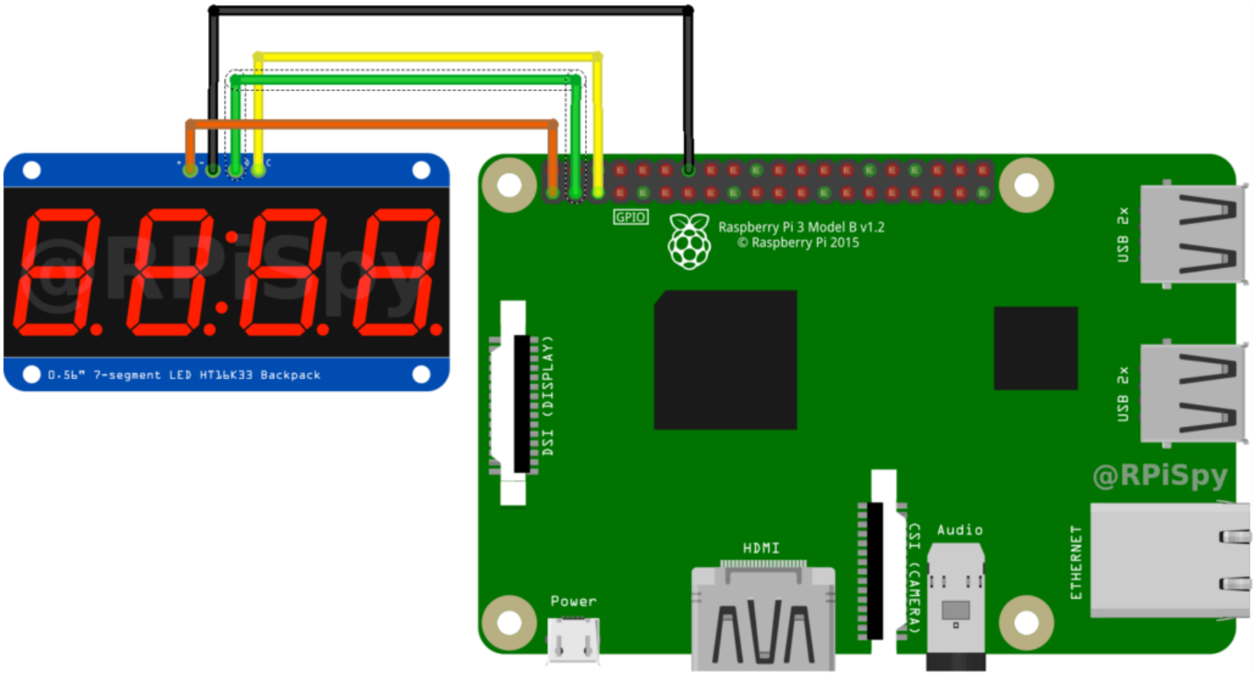
1. **Digit Display:** In IOT, a digit display is a visual interface that shows numerical data from connected devices or sensors, aiding users in monitoring and interacting with real-time information.
2. **Raspberry Pi:** A Raspberry Pi is a small, affordable, single- board computer that's widely used 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.
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.

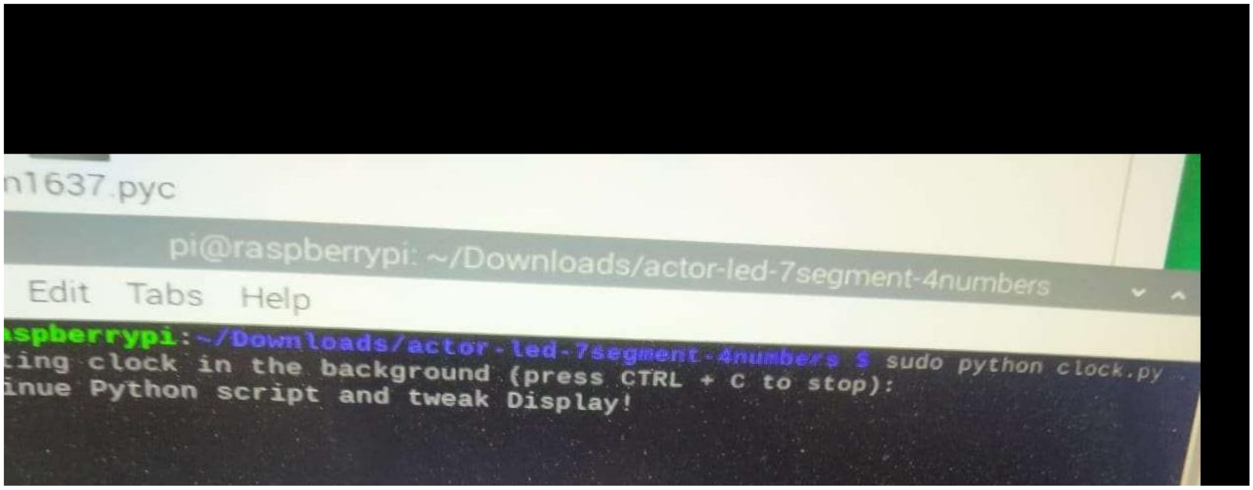
Steps:-

1. Open the web browser and go to the link: https://github.com/timwaizenegger/raspberrypi-%20examples/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.
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
```





```
aspberrypi:~/Downloads/actor-led-7segment-4numbers $ sudo python clock.py
ting clock in the background (press CTRL + C to stop):
inue Python script and tweak Display!
```

Practical - 3

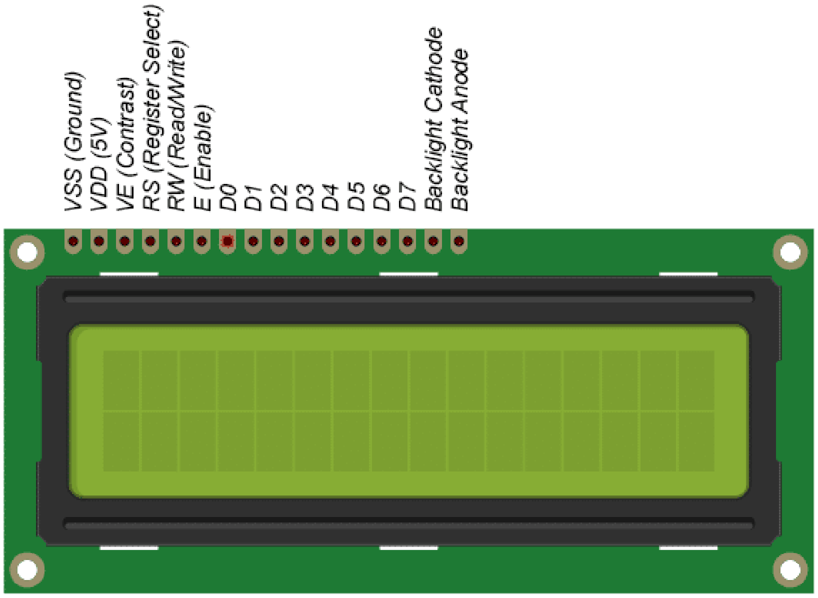
Aim: Interfacing 16X2 LCD with Raspberry Pi to display different messages.

Hardware Requirements:

1. **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.
2. **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.
3. **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.
4. **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).
5. **Power Supply:** A power supply converts incoming electrical energy into the right form to power electronic device.
6. **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.
7. **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.
8. **Resistor 330 Ω:** 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.
9. **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.

10. **LCD 16X2:** A 16×2 LCD display is a liquid crystal display that can show 16 characters in each of its two rows, providing a total of 32 characters of information. It's commonly used to display alphanumeric information in various electronic devices.

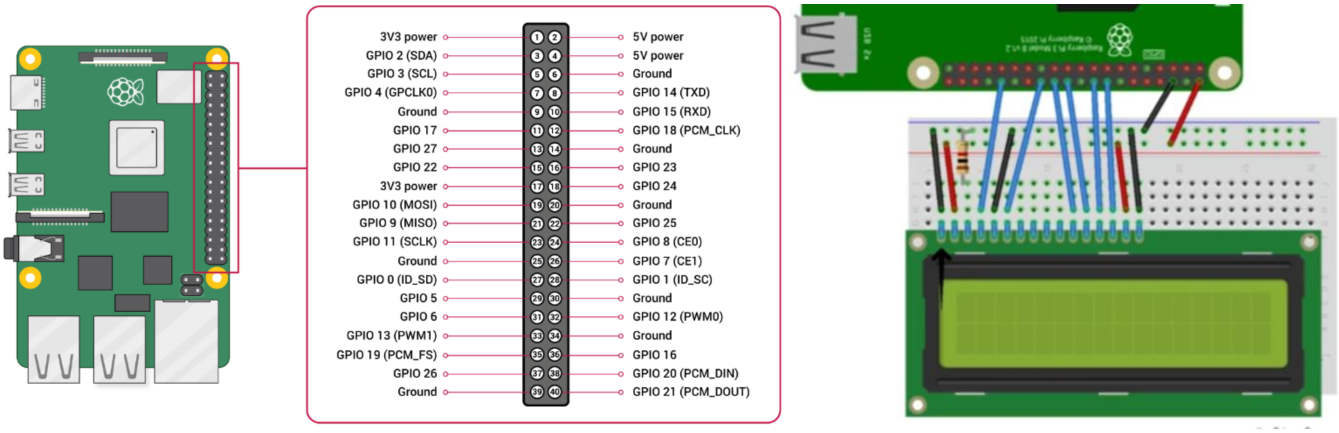
Pin out of Liquid Crystal Display



- **VSS:** This is the ground pin.
- **VDD:** This is the 5V pin.
- **V0:** This pin controls the contrast of the LCD.
- **RS (Register Select Pin):** This pin control where you are writing data in the LCD's memory. There are two types of registers; Data register which holds what goes on the screen and the instruction register where the LCD looks for the next instruction.
- **R/W (Read/Write Pin):** This pin selects the mode; Reading mode or Writing mode. Connecting it to ground will put the LCD in the read mode.
- **E (Enable Pin):** This pin enables the writing to the registers.

- **Data Pins:** There are 8 data pins (D0-D7). The high or low state of these pins represents the bits that you are writing to register in the write mode or the values you are reading in the read mode.
- The last two pins are for the LCD back light. Some LCD’s have 16 pins and some have 14 pins. If you have a 14 pin LCD then it means that there is no back light.
- **A (LED+):** This pin is the positive connection of the back light.
-
- **K (LED-):** This pin is the negative connection of the back light.

Circuit Diagram and Connections



Source:- Raspberry Pi

The LCD can be connected with the Raspberry pi in both 4 bit as well as 8 bit mode. The 4 bit mode required 6 GPIO pins while the 8 bit mode requires 10 GPIO pins. Because the 4-bit mode requires less GPIO pins so we are going to use the 4-bit mode. Both modes have same speed

16X2 LCD	Raspberry Pi
VSS	GND
VDD	5V
V0	MIDDLE PIN OF 4.7K POT
RS	GPIO26
R/W	GND
E	GPIO19
D4	GPIO13

D5	GPIO6
D6	GPIO5
D7	GPIO21
A	TO 5V THROUGH 330 OHMS RESISTOR
K	GND

Installing the Library

First of all, you will need to install the Adafruit library for LCD, so type the below command in the Raspberry pi terminal.

```
git clone https://github.com/adafruit/Adafruit_Python_CharLCD.git
```

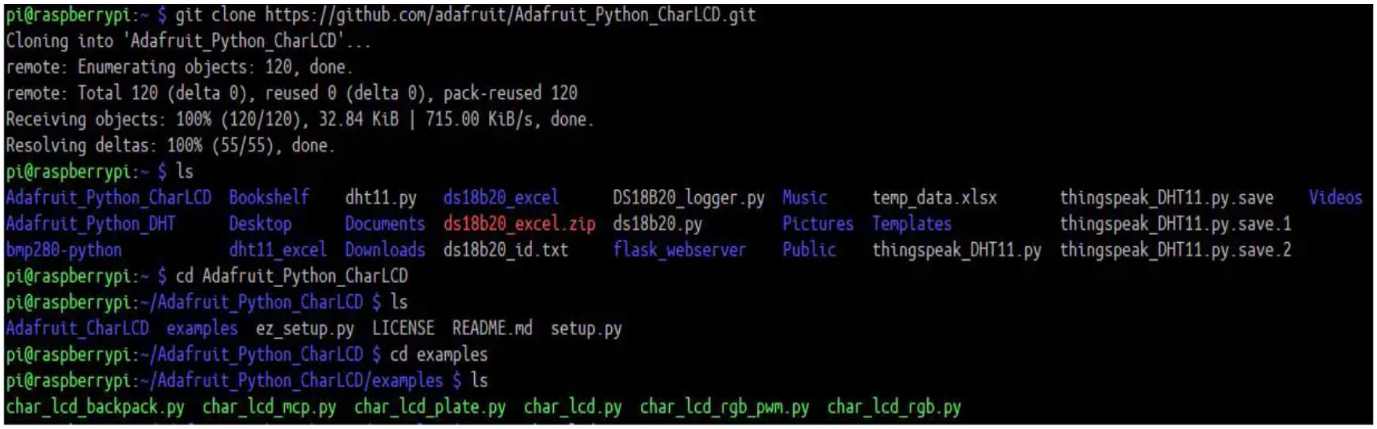
Then enter into the directory we just created by typing the following command.

```
cd ./Adafruit_Python_CharLCD
```

Now run the setup by typing the below command.

```
sudo python setup.py install
sudo python3 setup.py install
```

Then go the examples folder and open file ‘**char_lcd.py**’ file. Change the pin numbers in that example to the ones shown below.



```
pi@raspberrypi:~ $ git clone https://github.com/adafruit/Adafruit_Python_CharLCD.git
Cloning into 'Adafruit_Python_CharLCD'...
remote: Enumerating objects: 120, done.
remote: Total 120 (delta 0), reused 0 (delta 0), pack-reused 120
Receiving objects: 100% (120/120), 32.84 KiB | 715.00 KiB/s, done.
Resolving deltas: 100% (55/55), done.
pi@raspberrypi:~ $ ls
Adafruit_Python_CharLCD  Bookshelf    dht11.py   ds18b20_excel      DS18B20_logger.py  Music     temp_data.xlsx       thingspeak_DHT11.py.save    Videos
Adafruit_Python_DHT      Desktop      Documents  ds18b20_excel.zip  ds18b20.py         Pictures  Templates            thingspeak_DHT11.py.save.1
bmp280-python            dht11_excel  Downloads  ds18b20_id.txt     flask_webserver    Public    thingspeak_DHT11.py  thingspeak_DHT11.py.save.2
pi@raspberrypi:~ $ cd Adafruit_Python_CharLCD
pi@raspberrypi:~/Adafruit_Python_CharLCD $ ls
Adafruit_CharLCD  examples  ez_setup.py  LICENSE  README.md  setup.py
pi@raspberrypi:~/Adafruit_Python_CharLCD $ cd examples
pi@raspberrypi:~/Adafruit_Python_CharLCD/examples $ ls
char_lcd_backpack.py  char_lcd_mcp.py  char_lcd_plate.py  char_lcd.py  char_lcd_rgb_pwm.py  char_lcd_rgb.py
```

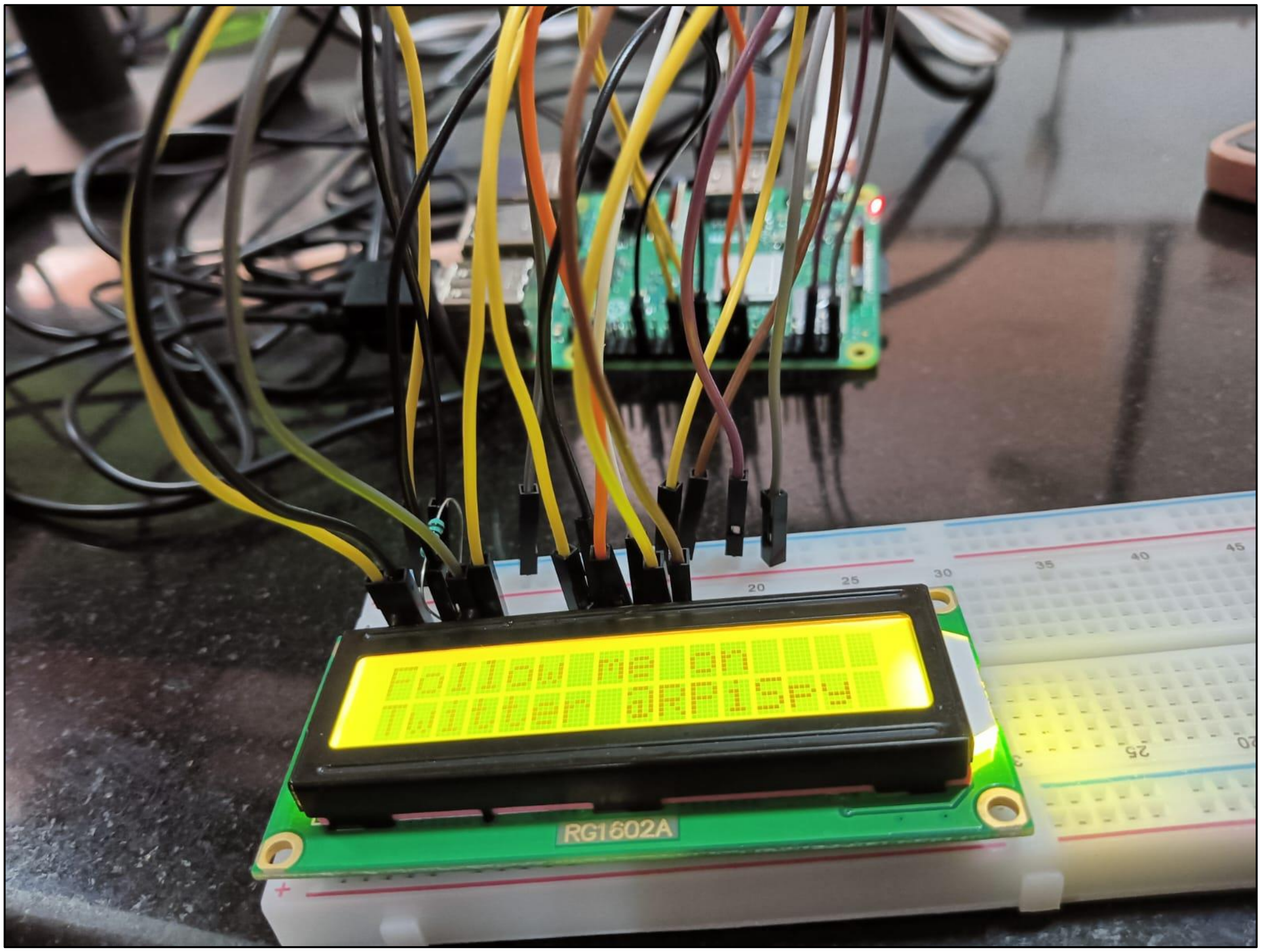
Python Code to Print and Scroll Text on LCD

The below code will simply print and scroll the text on the LCD.

Code:

```
from Adafruit_CharLCD import Adafruit_CharLCD  # Importing Adafruit library for LCD
from time import sleep  # Importing sleep from time library to add delay in program
# Initiate lcd and specify pins
lcd = Adafruit_CharLCD(rs=26, en=19, d4=13, d5=6, d6=5, d7=21, cols=16, lines=2)
lcd.clear()
# Display text on LCD, \n = new line
lcd.message('WELCOME TO \nIoT STARTERS')
sleep(2)
# Scroll text on display
try:
    while True:
        # Scroll Left
        for x in range(0, 16):
            lcd.move_left()
            sleep(0.1)
        sleep(1)
        # Scroll Right
        for x in range(0, 16):
            lcd.move_right()
            sleep(0.1)
        sleep(3)
# If Keyboard Interrupt command is pressed
except KeyboardInterrupt:
    pass
# Clear the screen
lcd.clear()
```

Output:



```
Follow me on
Twitter @RPispy
```

Practical 4

Aim: Raspberry Pi based Oscilloscope Installation Manual

One of the most important tools in Electrical/Electronic engineering is The Oscilloscope.

An oscilloscope is a laboratory instrument commonly used to display and analyze the waveform of electronic signals. In effect, the device draws a graph of the instantaneous signal voltage as a function of time.

In this project we will seek to replicate the signal visualization capabilities of the oscilloscope using the Raspberry Pi and an analog to digital converter module.

Replicating the signal visualization of the oscilloscope using the Raspberry Pi will require the following steps:

1. Perform Digital to analog conversion of the Input signal.
2. Prepare the resulting data for representation.
3. Plot the data on a live time graph

Analog Signals → **Signal Processing** → **Visualize using the Graph**

Hardware Requirements:

1. Raspberry Pi Model A/B/B+
2. ADS1115 ADC.
3. Breadboard.
4. Jumper Wires

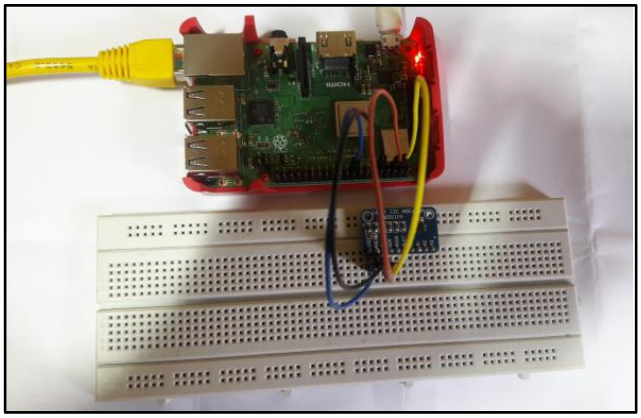
ADS1115 ADC chip is used to convert the analog input signals to digital signals which can be visualized with the Raspberry Pi. This chip is important because the Raspberry Pi does not have an on board analog to digital converter (ADC).

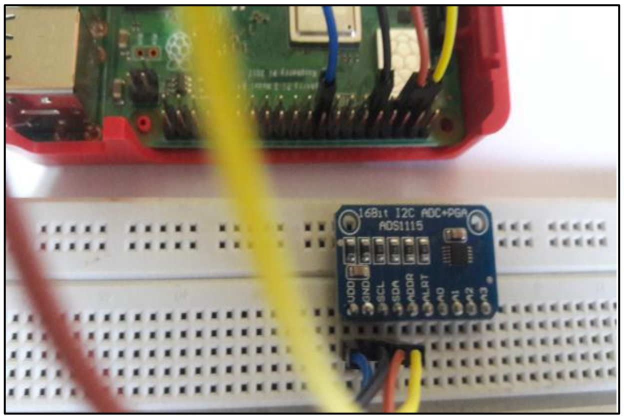
Software Requirements:

1. Raspbian Stretch OS
2. Adafruit module for interfacing with the ADS1115 ADC chip
3. Python Module matplotlib used for data visualization

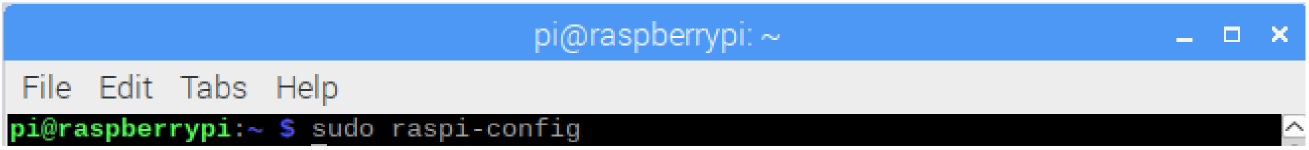
Connect your ADC with Raspberry Pi's GPIO Pins.

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





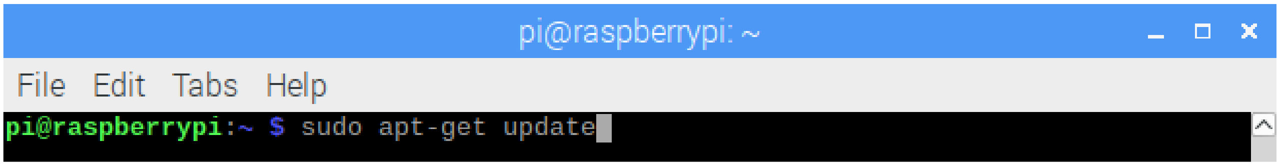
Step 1: Enable Raspberry Pi I2C interface



```
pi@raspberrypi:~ $ sudo raspi-config
```

Go to Interfacing Options → I2C → Enable (Yes)

Step 2: Update the Raspberry pi



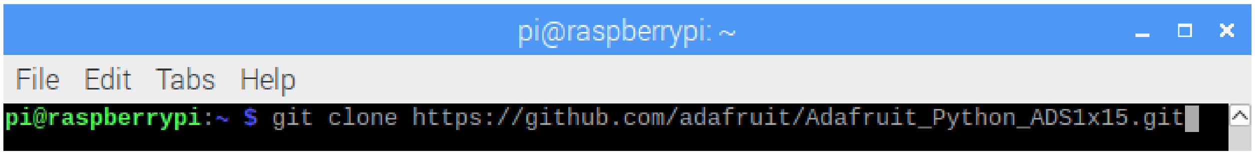
```
pi@raspberrypi:~ $ sudo apt-get update
```

Step 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 ~)

```
  $ sudo apt-get install build-essential python3-dev python3-smbus git.
```

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



```
pi@raspberrypi:~ $ git clone https://github.com/adafruit/Adafruit_Python_ADS1x15.git
```

Change into the cloned file’s directory and run the setup file

```
    $ sudo python3 setup.py install
```

Step 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
```

```
$ python3 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.

Step 5: Install *Matplotlib.*

```
   $ sudo apt-get install python3-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.

Step 6: Python Code for Raspberry Pi Oscilloscope:

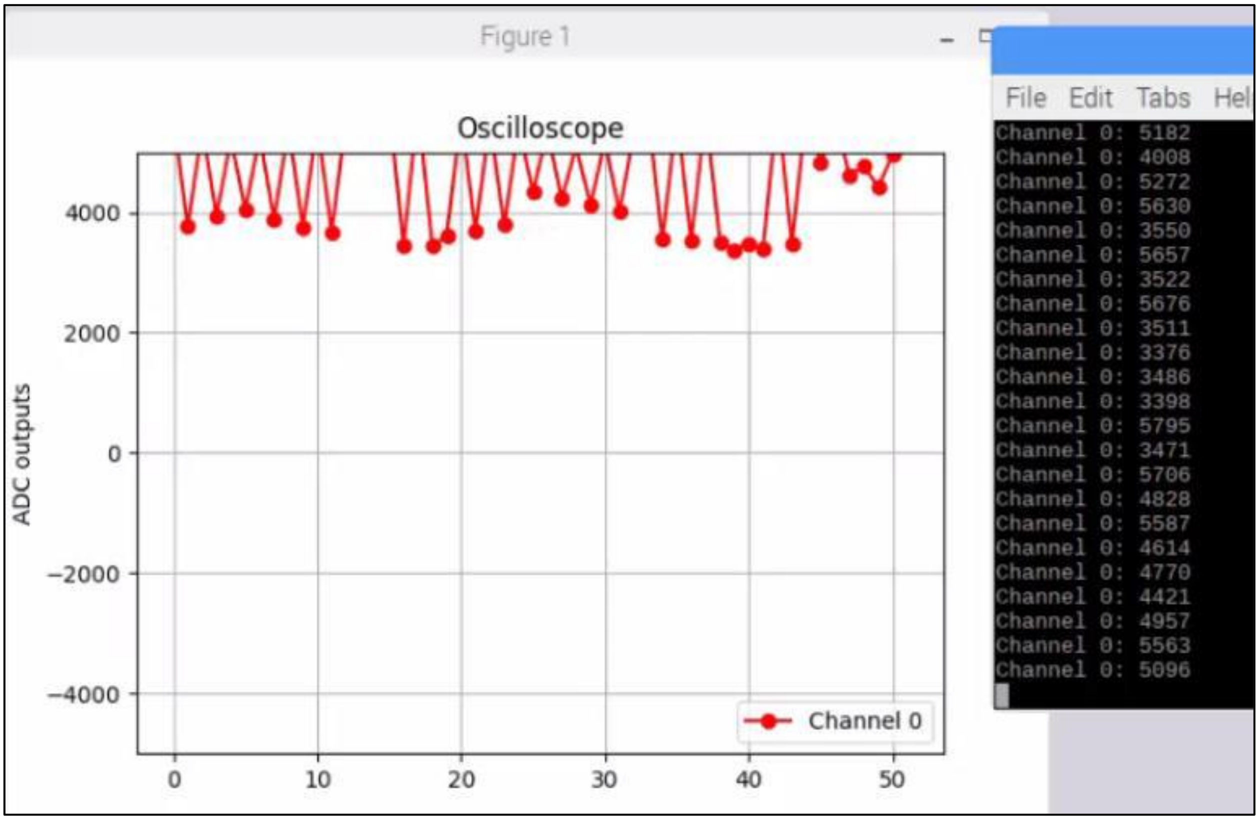
```
import matplotlib.pyplot as plt
from matplotlib.animation import FuncAnimation
import Adafruit_ADS1x15
# Create an ADS1115 ADC (16-bit) instance
adc = Adafruit_ADS1x15.ADS1115()
GAIN = 1
# Start continuous ADC conversions on channel 0 using the previous gain value
adc.start_adc(0, gain=GAIN)
print('Reading ADS1x15 channel 0')
# Set up the plot
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')
# Store values for later
val = []
def update(cnt):
    # Read the last ADC conversion value
    value = adc.get_last_result()
    print('Channel 0: {0}'.format(value))
    # Store the new value
    val.append(int(value))
    # Limit the number of points displayed
    if len(val) > 50:
        val.pop(0)
    # Set new data to line
    line.set_data(range(len(val)), val)
    ax.relim()
    ax.autoscale_view()
# Create animation
ani = FuncAnimation(fig, update, interval=500)
plt.show()
```

Save the code and run using

```
  $ python3 osilloscopedemo.py
```

That's all !!!

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



```
Channel 0: 5182
Channel 0: 4008
Channel 0: 5272
Channel 0: 5630
Channel 0: 3550
Channel 0: 5657
Channel 0: 3522
Channel 0: 5676
Channel 0: 3511
Channel 0: 3376
Channel 0: 3486
Channel 0: 3398
Channel 0: 5795
Channel 0: 3471
Channel 0: 5706
Channel 0: 4828
Channel 0: 5587
Channel 0: 4614
Channel 0: 4770
Channel 0: 4421
Channel 0: 4957
Channel 0: 5563
Channel 0: 5096
```

Practical - 5

Aim: Controlling Raspberry Pi with Telegram.

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.
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: Go to telegram and search and open botfather

   i. Type the commands
   ii. /Start
   iii. /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 led's or to create a pattern.

Terminal Command:-

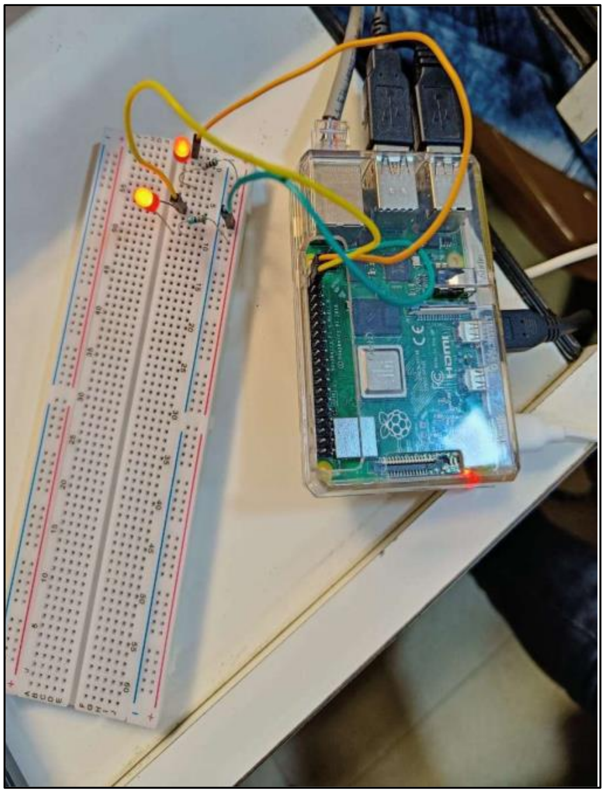
```
sudo pip3 install telepot
```

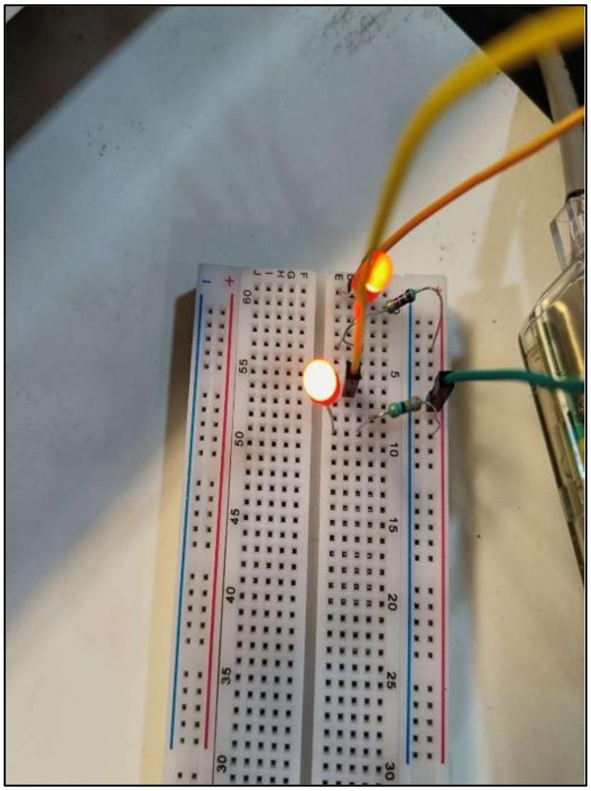
Code:-

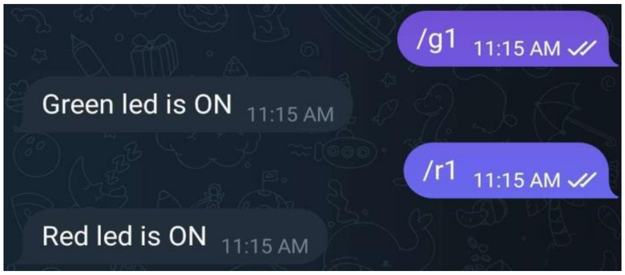
```
import datetime
import telepot
from telepot.loop import MessageLoop
import RPi.GPIO as GPIO
from time import sleep
# Define GPIO pins
red_led_pin = 21
green_led_pin = 20
# Set up GPIO
GPIO.setmode(GPIO.BCM)
GPIO.setup(red_led_pin, GPIO.OUT)
GPIO.setup(green_led_pin, GPIO.OUT)
# Create a bot instance
bot = telepot.Bot('YOUR_BOT_TOKEN')
print(bot.getMe())
def handle(msg):
    chat_id = msg['chat']['id']
    command = msg['text']
    print('Received:', command)
    if command == '/hi':
        bot.sendMessage(chat_id, "Hi!")
    elif command == '/r1':
        bot.sendMessage(chat_id, "Red LED is ON")
        GPIO.output(red_led_pin, True)
    elif command == '/r0':
        bot.sendMessage(chat_id, "Red LED is OFF")
        GPIO.output(red_led_pin, False)
    elif command == '/g1':
        bot.sendMessage(chat_id, "Green LED is ON")
        GPIO.output(green_led_pin, True)
    elif command == '/g0':
        bot.sendMessage(chat_id, "Green LED is OFF")
        GPIO.output(green_led_pin, False)
```

```
# Start listening for messages
MessageLoop(bot, handle).run_as_thread()
print('Listening...')
# Keep the program running
while True:
    sleep(10)
```

Output:





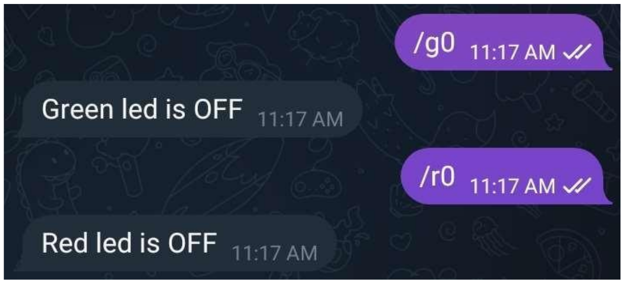


/g1 11:15 AM

Green led is ON 11:15 AM

/r1 11:15 AM

Red led is ON 11:15 AM



/g0 11:17 AM

Green led is OFF 11:17 AM

/r0 11:17 AM

Red led is OFF 11:17 AM

Practical - 6

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, 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.
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 end 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 10

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 super user. If you can’t, add the pi-user to the dialout group:

```
sudo adduser pi dialout
```

- ➢ Now it’s finally time to determine the position of the Raspberry Pi. Type the following command to stop the gpsd service that got started automatically when you installed gpsd earlier. You have to do this because the default options aren’t correct for the Pi:

```
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

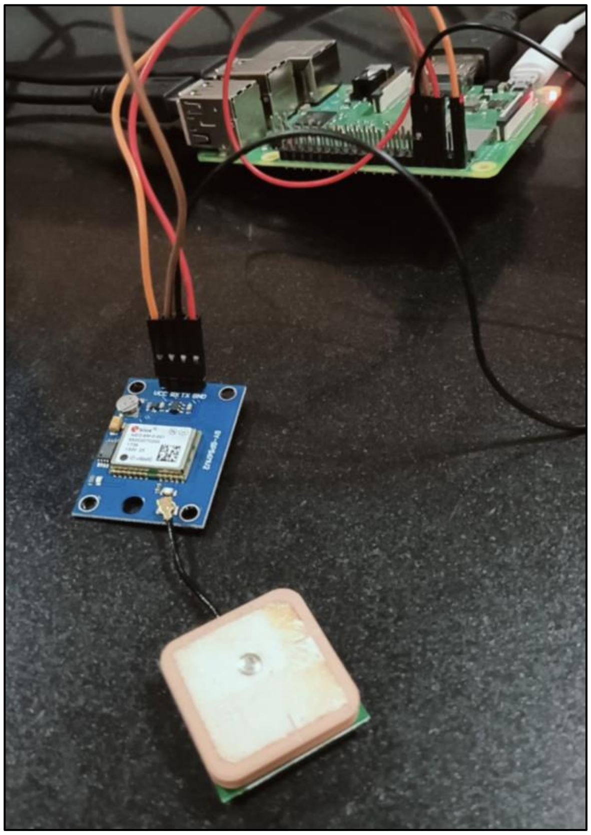
sudo cgps -s

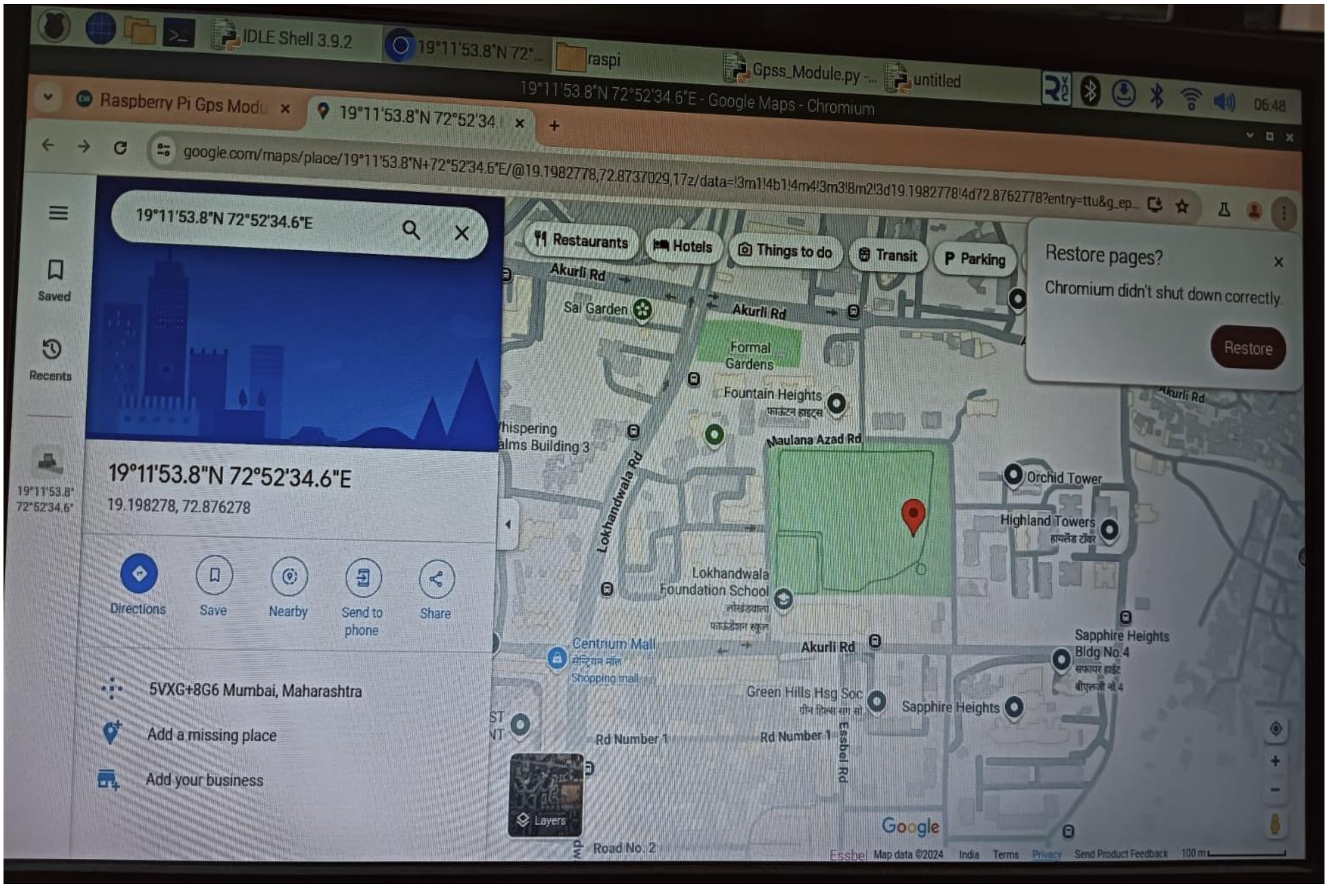
2. Now it will show your latitude and longitude of your current locations on the screen.

Code:

```
import serial               # Import serial package
from time import sleep
import webbrowser           # Import package for opening link in browser
import sys                  # Import system package

def GPS_Info(NMEA_buff):
    # Extract and print GPS information
    nmea_time = NMEA_buff[0]                   # Extract time from GPGGA string
    nmea_latitude = NMEA_buff[1]               # Extract latitude from GPGGA string
    nmea_longitude = NMEA_buff[3]              # Extract longitude from GPGGA string

    print("NMEA Time: ", nmea_time, '\n')
    print("NMEA Latitude:", nmea_latitude, "NMEA Longitude:", nmea_longitude, '\n')

    lat = float(nmea_latitude)                 # Convert string to float
    longi = float(nmea_longitude)              # Convert string to float
    lat_in_degrees = convert_to_degrees(lat)    # Get latitude in degrees
    long_in_degrees = convert_to_degrees(longi) # Get longitude in degrees

    return lat_in_degrees, long_in_degrees

# Convert raw NMEA string into decimal degrees
def convert_to_degrees(raw_value):
    decimal_value = raw_value / 100.00
    degrees = int(decimal_value)
    mm_mmmm = (decimal_value - degrees) * 0.6
    position = degrees + mm_mmmm
    position = "%.4f" % position
    return position

gpgga_info = "$GPGGA,"
ser = serial.Serial("/dev/ttyS0")             # Open port with baud rate
GPGGA_buffer = ""
NMEA_buff = []
lat_in_degrees = 0
long_in_degrees = 0

try:
    while True:
        received_data = ser.readline().decode('ascii', errors='ignore').strip()  # Read and decode NMEA string
        GPGGA_data_available = received_data.find(gpgga_info)   # Check for NMEA GPGGA string

        if GPGGA_data_available > 0:
            GPGGA_buffer = received_data.split("$GPGGA,", 1)[1]  # Store data after "$GPGGA,"
            NMEA_buff = GPGGA_buffer.split(',')              # Store comma-separated data in buffer

            lat_in_degrees, long_in_degrees = GPS_Info(NMEA_buff)  # Get time, latitude, longitude
        print("lat in degrees:", lat_in_degrees, "long in degrees:", long_in_degrees, '\n')
        map_link = 'http://maps.google.com/?q=' + lat_in_degrees + ',' + long_in_degrees    # Create link to plot location on Google Maps
        print("<<<<<<<< press ctrl+c to plot location on Google Maps >>>>>>\n")  # Press ctrl+c to plot on map and exit
        print("------------------------------------------------------------\n")

except KeyboardInterrupt:
    webbrowser.open(map_link)        # Open current position information in Google Maps
    sys.exit(0)
except Exception as e:
    print("An error occurred:", e)
    sys.exit(1)
```

Output:-





19°11'53.8"N 72°52'34.6"E - Google Maps - Chromium

google.com/maps/place/19°11'53.8"N+72°52'34.6"E/@19.1982778,72.8737029,17z/data=!3m1!4b1!4m4!3m3!8m2!3d19.1982778!4d72.8762778?entry=ttu&g_ep...

19°11'53.8"N 72°52'34.6"E

19.198278, 72.876278

5VXG+8G6 Mumbai, Maharashtra

Restore pages?

Chromium didn't shut down correctly.

Practical - 7

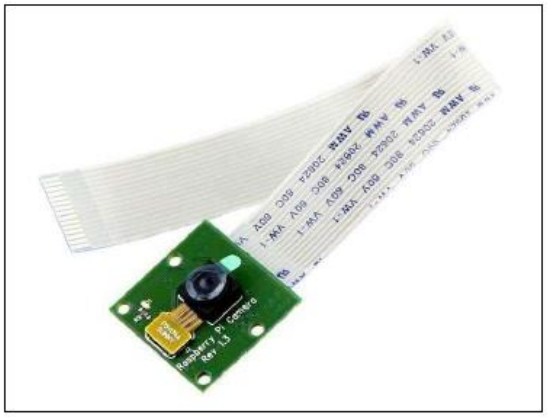
Aim: Interfacing Pi Camera with Raspberry Pi

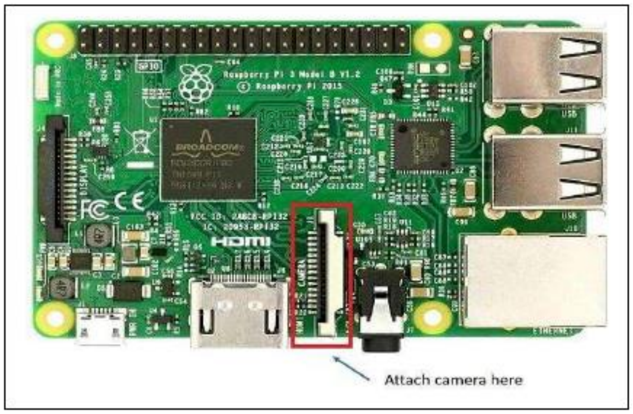
Hardware Requirements:

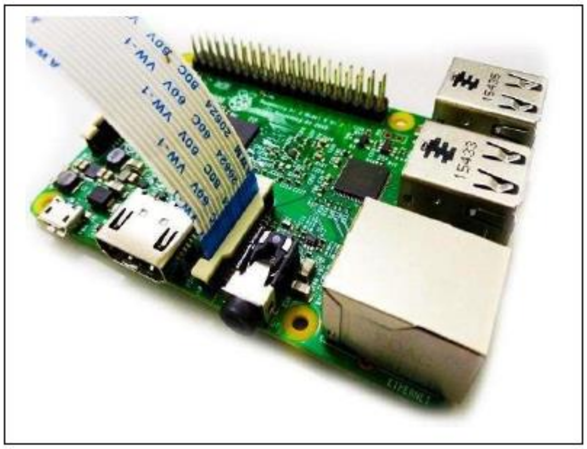
1. **Camera:** It will automatically record, monitor and alert the user.
2. **Raspberry Pi:** A Raspberry Pi is a small, affordable, single- board computer that's widely used for various DIY projects, education, and prototyping. It can run a variety of operating systems and is popular for programming, learning about computing, and building innovative electronics projects.
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:







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:

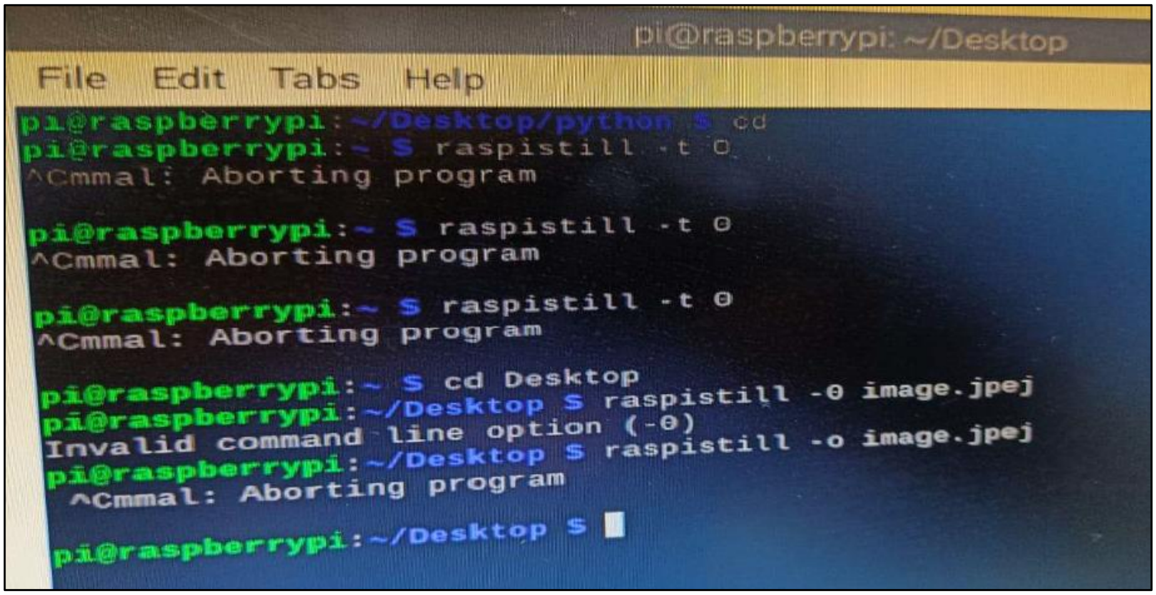
Video Recording (video 1.py):

```
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")
camera.stop_preview()
```

Image Capture (picam.py)

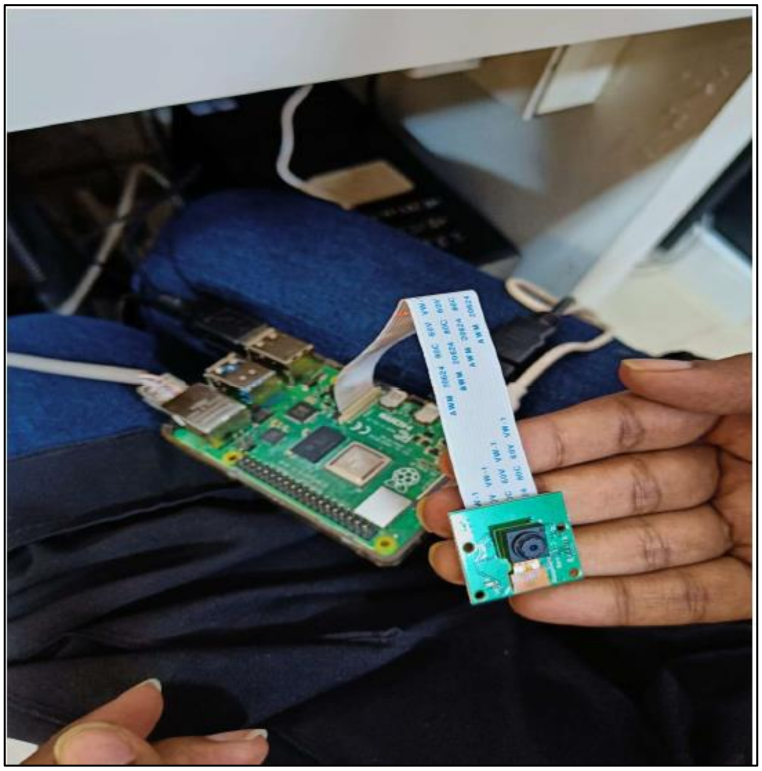
```
import time
from picamera import PiCamera
camera = PiCamera()
camera.resolution = (1280, 720)
camera.start_preview()
time.sleep(5)
camera.capture('/home/pi/Desktop/ty2.jpg')
camera.stop_preview()
```

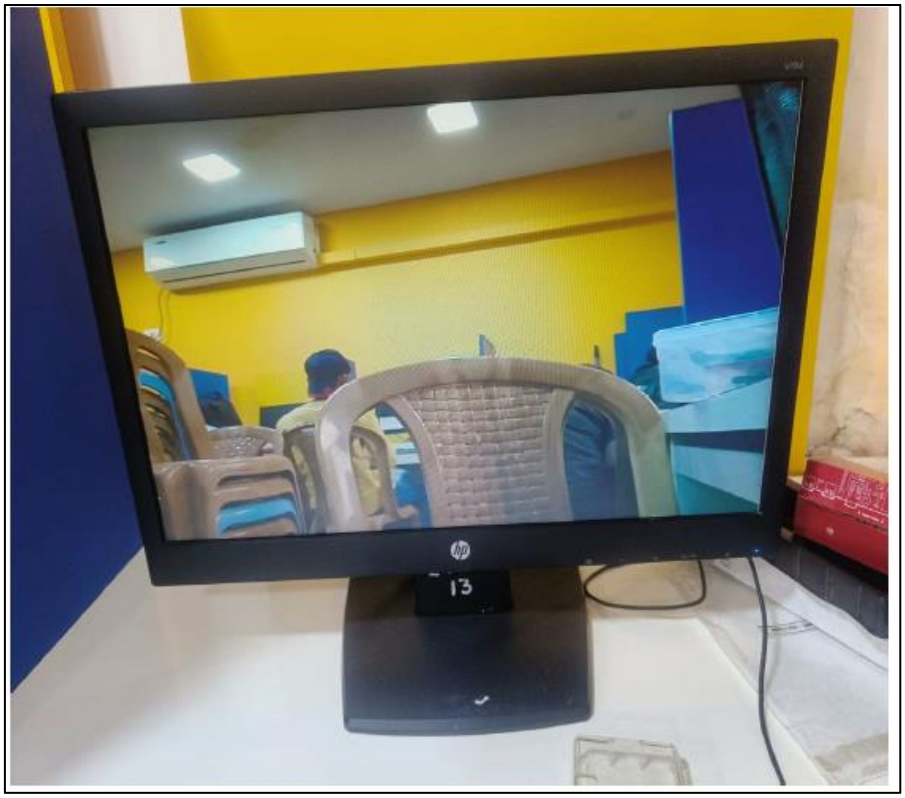
Note: You can also directly run the camera by using this command.



```
pi@raspberrypi:~/Desktop/python $ cd
pi@raspberrypi:~ $ raspistill -t 0
^Cmmal: Aborting program

pi@raspberrypi:~ $ raspistill -t 0
^Cmmal: Aborting program

pi@raspberrypi:~ $ raspistill -t 0
^Cmmal: Aborting program

pi@raspberrypi:~ $ cd Desktop
pi@raspberrypi:~/Desktop $ raspistill -0 image.jpej
Invalid command line option (-0)
pi@raspberrypi:~/Desktop $ raspistill -o image.jpej
 ^Cmmal: Aborting program

pi@raspberrypi:~/Desktop $ 
```

Output:





Practical - 8

Aim: Interfacing Raspberry Pi with RFID

Hardware Requirements:

1. **RFID Tag:** RFID tags are a type of tracking system that uses radiofrequency 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, 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.
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.

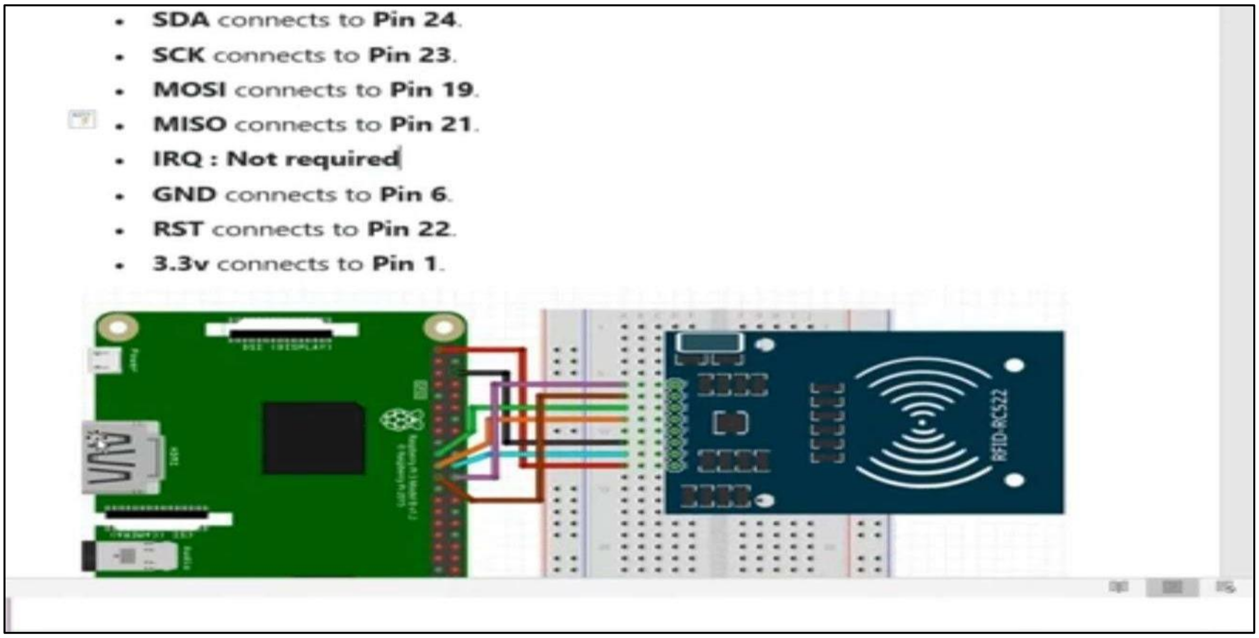
Code:

Read.py

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

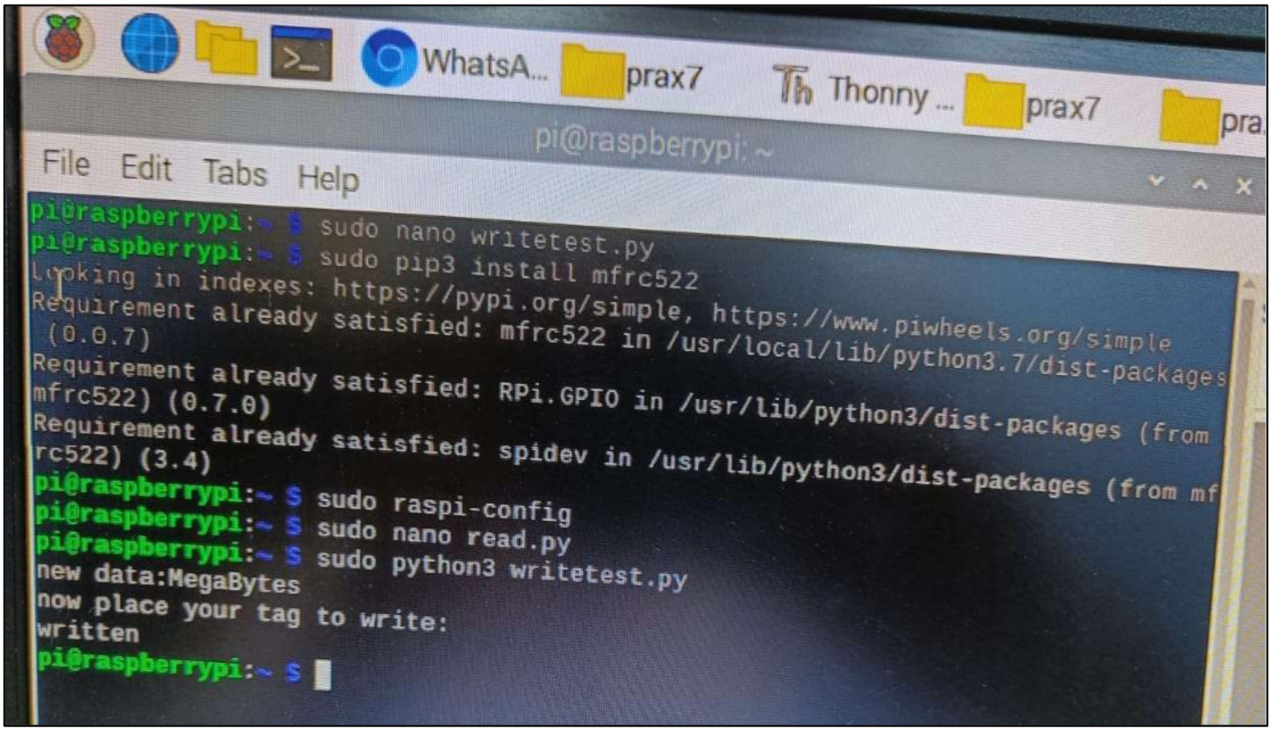
```
sudo nano read.py
```

- 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:



```
pi@raspberrypi:~ $ sudo nano writetest.py
pi@raspberrypi:~ $ sudo pip3 install mfrc522
Looking in indexes: https://pypi.org/simple, https://www.piwheels.org/simple
Requirement already satisfied: mfrc522 in /usr/local/lib/python3.7/dist-packages
 (0.0.7)
Requirement already satisfied: RPi.GPIO in /usr/lib/python3/dist-packages (from
mfrc522) (0.7.0)
Requirement already satisfied: spidev in /usr/lib/python3/dist-packages (from mf
rc522) (3.4)
pi@raspberrypi:~ $ sudo raspi-config
pi@raspberrypi:~ $ sudo nano read.py
pi@raspberrypi:~ $ sudo python3 writetest.py
new data:MegaBytes
now place your tag to write:
written
pi@raspberrypi:~ $ 
```