PRACTICAL - 1

AIM:-

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

ABSTRACT:-

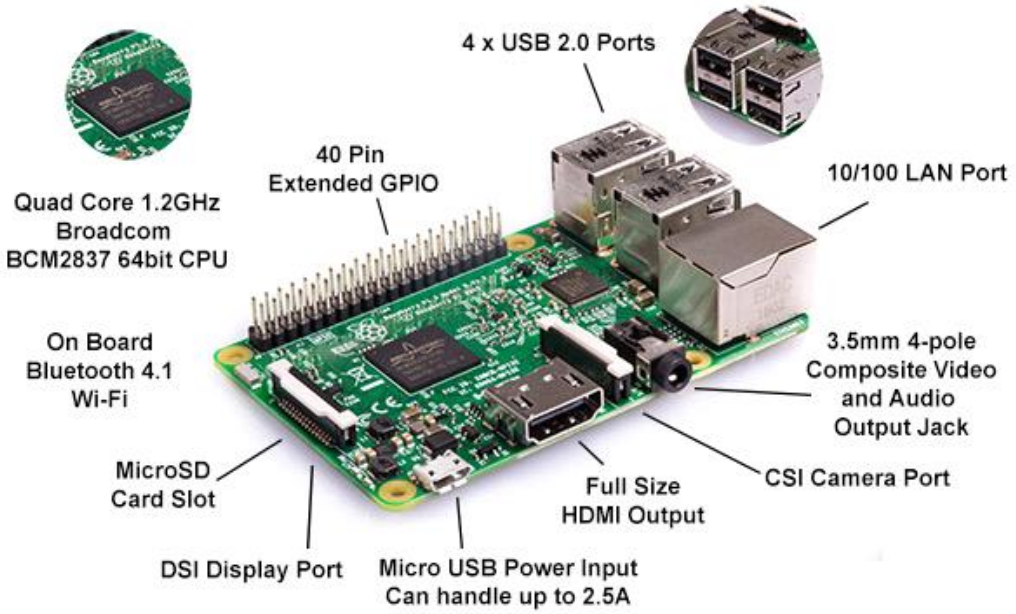
The Raspberry Pi is a mini computer specifically created to make tech learning easier. It has a loT components for computer-based projects, like USB ports, an Ethernet port, an SD card slot, Wi-Fi antenna ports, and more.

REQUIRED COMPONENTS:-

1. **Hardware:**

Particular	Quantity
Raspberry Pi3	1
Power Supply	1
USB Keyboard	1
USB Mouse	1
Micro SD card	1
Micro SD USB card reader	1
A Monitor that supports HDMI	1
An HDMI cable	1
An Ethernet cable	1

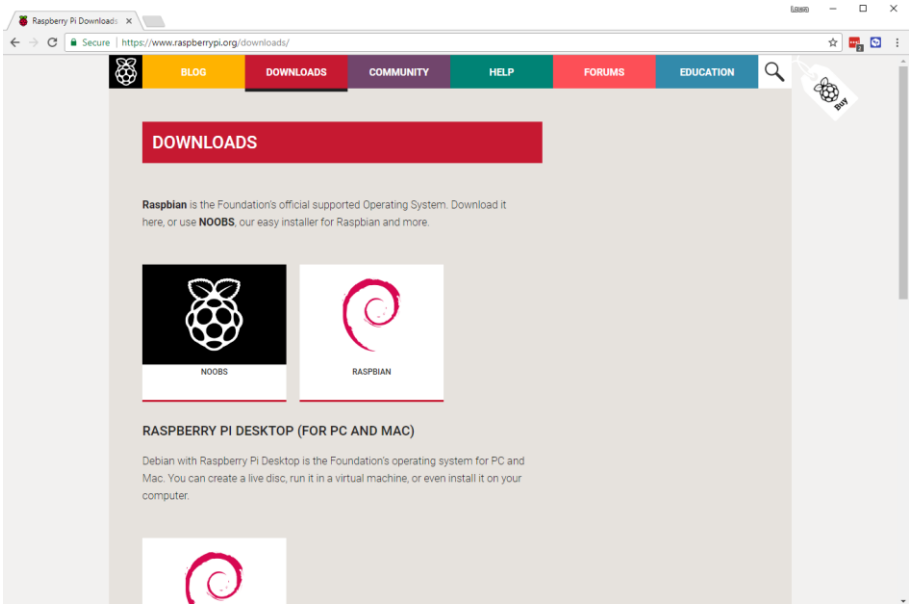
2. **Software :**
   Raspbian, installed via NOOBS

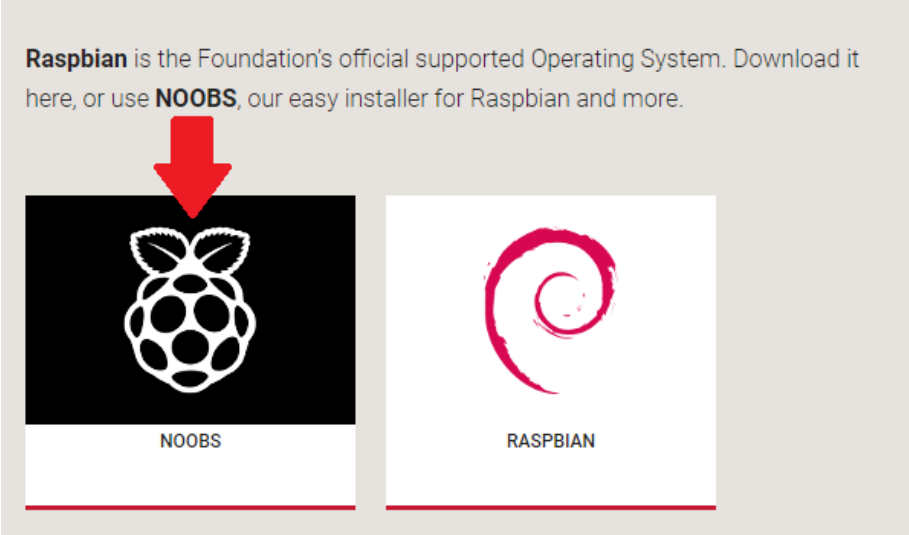


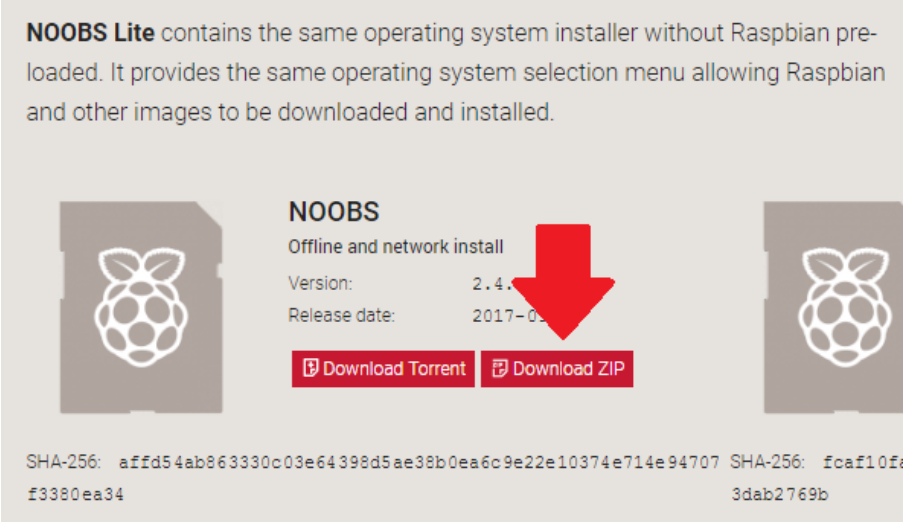
1. **USB ports** - these are used to connect a mouse and keyboard. You can also connect other components, such as a USB drive.
2. **SD card slot** - you can slot the SD card in here. This is where the operating system software and your files are stored.
3. **Ethernet port** - this is used to connect the Raspberry Pi to a network with a cable. The Raspberry Pi can also connect to a network via wireless LAN.
4. **Audio jack** - you can connect headphones or speakers here.
5. **HDMI port**- this is where you connect the monitor (or projector) that you are using to display the output from the Raspberry Pi. If your monitor has speakers, you can also use them to hear sound.
6. **Micro USB power connector** - this is where you connect a power supply. You should always do this last, after you have connected all your other components.
7. **GPIO ports** -these allow you to connect electronic components such as LEDs and buttons to the Raspberry Pi.

PROCEDURE:-

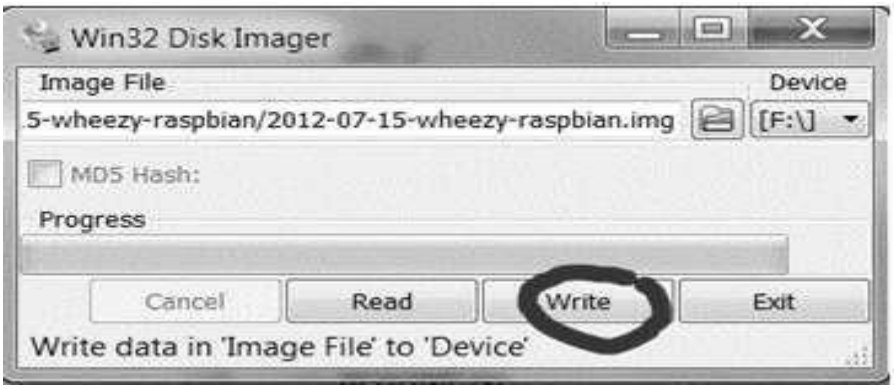
1. Downloading NOOBS
   Using NOOBS is the easiest way to install Raspbian on your SD card.
   To get a copy of NOOBS: Visit www.raspberrypi.org/downloads/



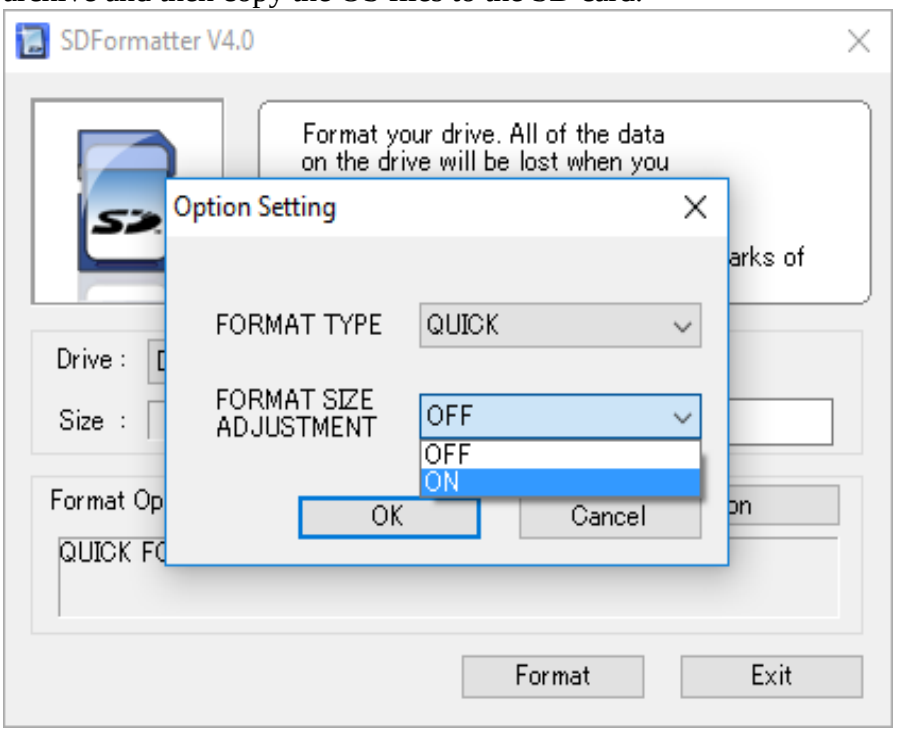




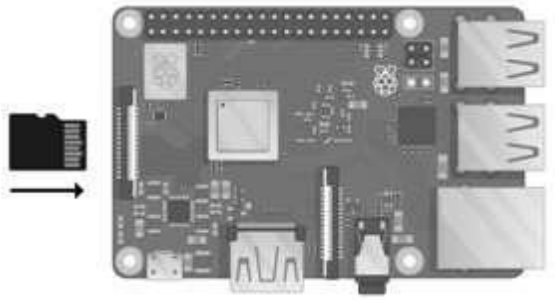
2. Insert the SD card into your SD card reader. Note the drive letter assigned to the SD card. You can see the drive letter in the right hand column of Windows Explorer.
3. Run the Win32DiskImager utility from desktop-menu then select the image file and click on Write Button.



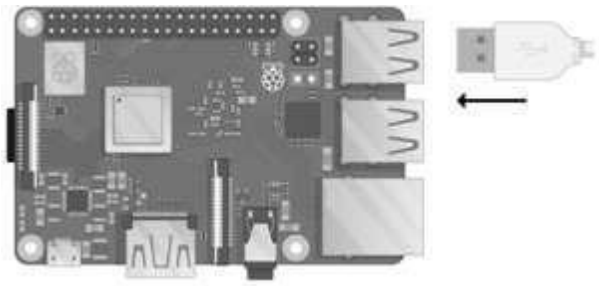
Or else run the SD card formatter, format the SD card and extract NOOBS from the zip archive and then copy the OS files to the SD card.



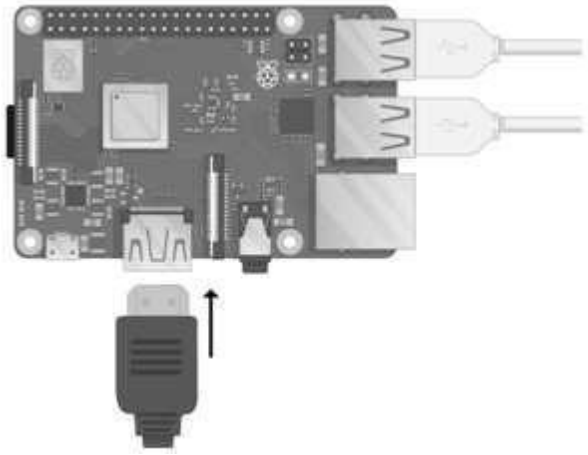
4. Remove SD Card & Insert in Raspberry Pi



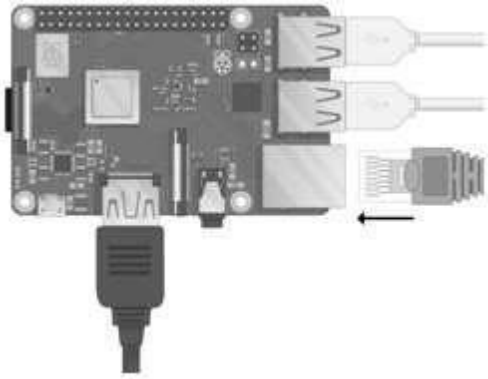
5. Connect the mouse & Keyboard to USB port of Raspberry Pi.



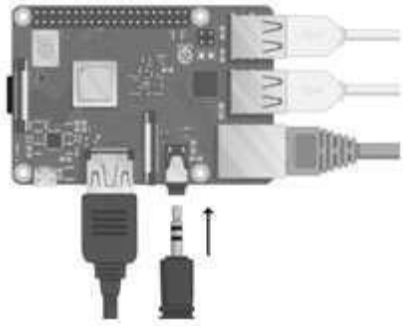
6. Connect Raspberry Pi to HDMI port directly or use HDMI to VGA connector.



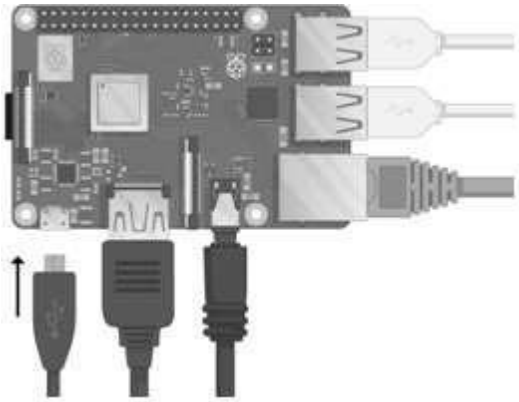
7. Connect the Pi to the internet via Ethernet, use an Ethernet cable to connect the RaspberryPi.



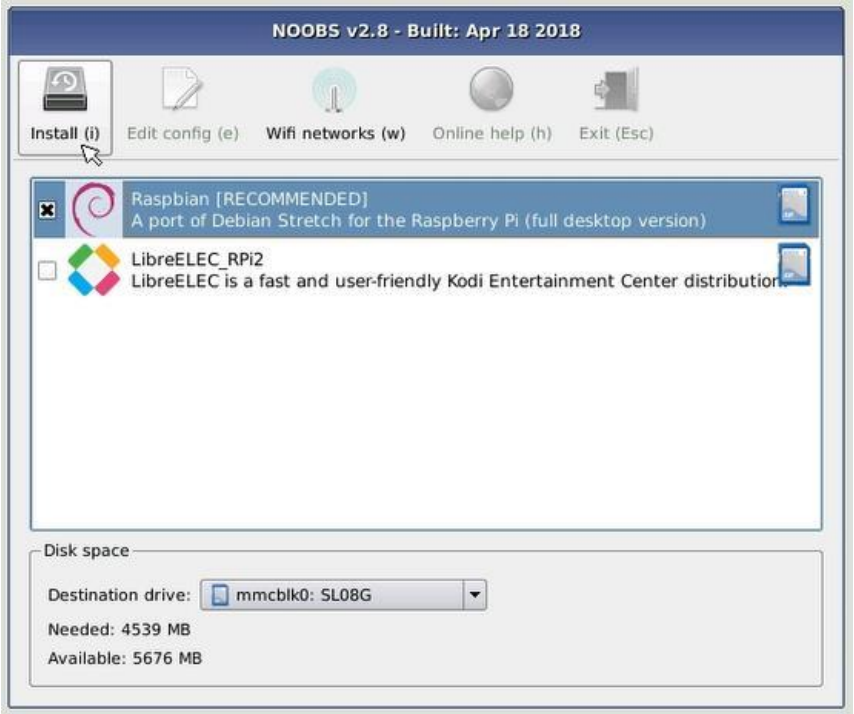
8. Sound will come if it has speakers or connect headphones or speakers to the Audio jack if necessary



9. Plug the power supply into a socket and connect it to the micro USB power port.



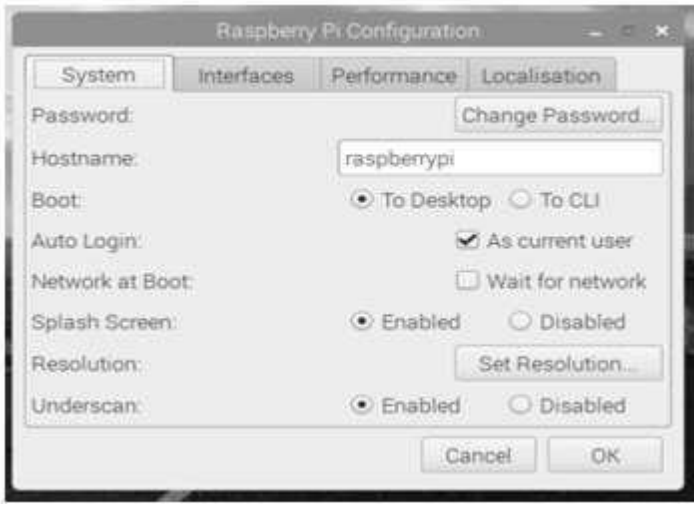
10. Once the files have been copied over, insert the micro SD Card into your Raspberry Pi, and plug the Pi into a power source.You will be offered a choice when the installer has loaded. You should check the box for Raspbian, and then click Install.



11. Red light will indicate Power On and Green Will Indicate the booting of the RaspberryPi. The Pi will boot up into a graphical desktop

12. **Configuring your Pi:**
    a. You can control most of your Raspberry Pi's settings, such as the password, through the Raspberry Pi Configuration application found in Preferences on the menu

b. System:
In this tab you can change basic system settings of your Pi.



Password - set the password of the pi user (it is a good idea to change the password from the factory default 'raspberry')

Boot - select to show the Desktop or CLI (command line interface) when your Raspberry Pi starts

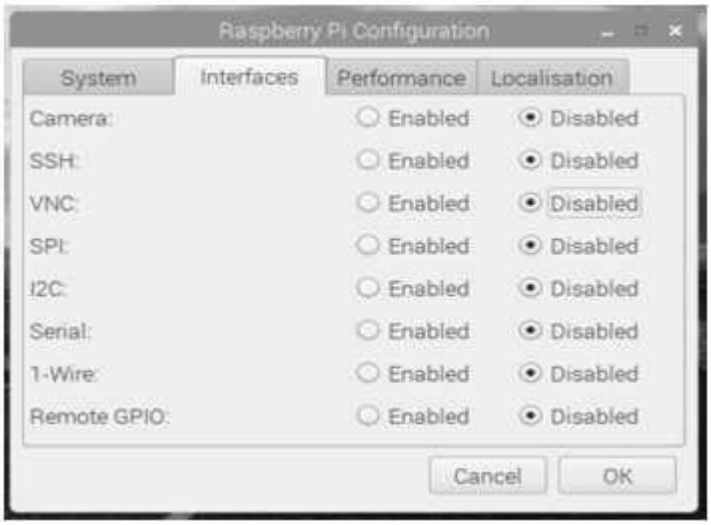
Auto Login- enabling this option will make the Raspberry Pi automatically log in whenever it starts

Network at Boot-selecting this option will cause your Raspberry Pi to wait until a network connection is available before starting
Splash Screen -choose whether or not to show the splash (startup) screen when your Raspberry Pi boots

Resolution - you can set the screen resolution here

Underscan - choose whether your Pi should show black bars at the top and bottom of the screen when it can't match the screen resolution

c. Interfaces:
You can link devices and components to the Raspberry Pi using a lot of different types of connections. The Interfaces tab is where you turn these different connections on or off, so that the Pi recognizes that you've linked something to it via a particular type of connection.



Camera - enable the Raspberry Pi Camera Module

SSH - allow remote access to your Raspberry Pi from another computer using SSH

VNC - allow remote access to the Raspberry Pi Desktop from another computer using VNC

SPI -enable the SPI GPIO pins

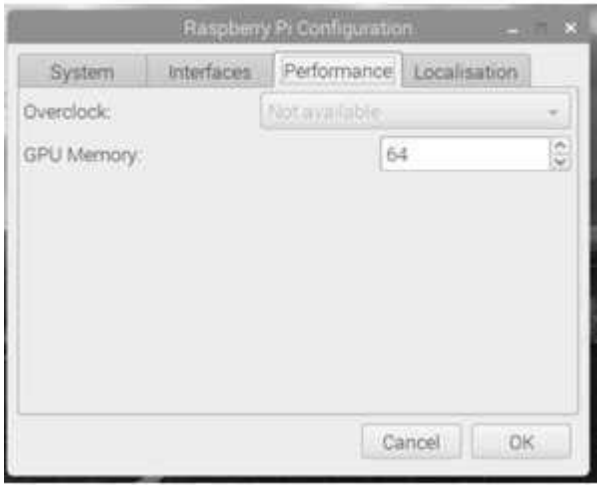
I2C - enable the I2C GPIO pins

Serial -enable the Serial (Rx, Tx) GPIO pins

1- Wire - enable the 1-Wire GPIO pin

Remote GPIO - allow access your Raspberry Pi's GPIO pins from another computer.

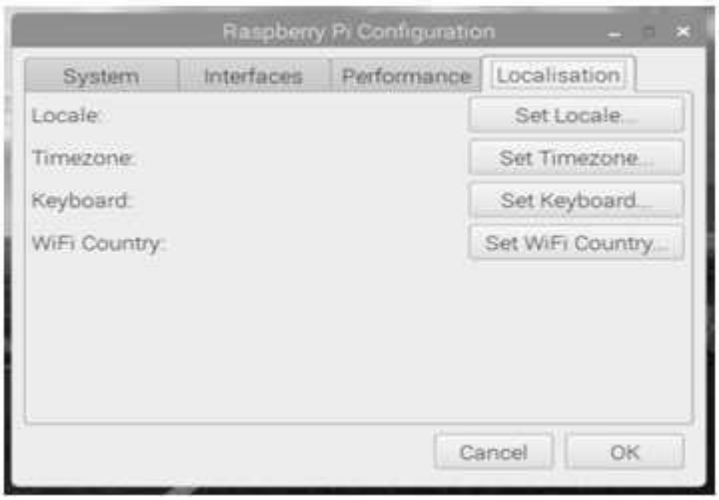
d. Performance:
If you need to do so for a particular project you want to work on, you can change the performance settings of your Pi in this tab.



Overclock - change the CPU speed and voltage to increase performance

GPU Memory - change the allocation of memory given to the GPU

e. Localisation



13. This tab allows you to change your Raspberry Pi settings to be specific to a country or location.
    Locale -set the language, country, and character set used by your Raspberry Pi
    Time zone- set the time zone
    Keyboard - change your keyboard layout
    Wi-Fi Country - set the Wi-Fi country code

14. After starting with Raspberry Pi for the first time, the Welcome to RaspberryPi application will pop up and it will guide through the initial setup.

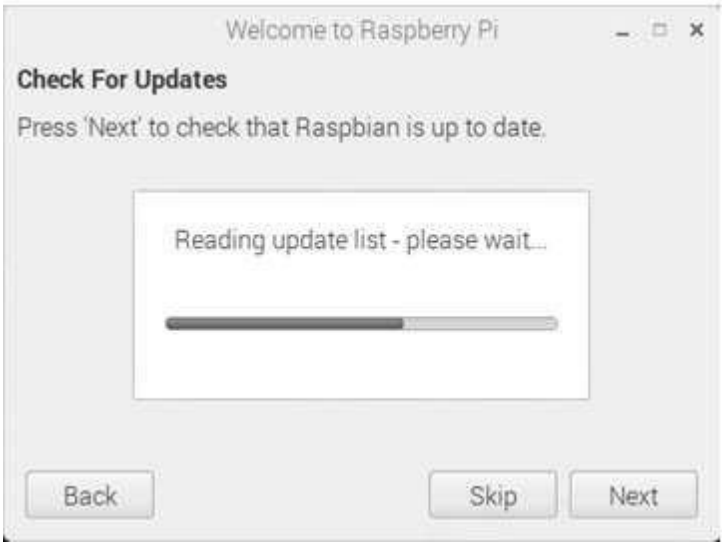
15. Click Next to start the setup.

16. Set respective Country, Language, and Time zone, then click next again.

17. Enter a new password for Raspberry Pi and click next.

18. Connect to your Wi-Fi network by selecting its name, entering the password, and clicking next

19. Click next let the wizard check for updates to Raspbian and install.



20. Click done or Reboot to finish the setup.

Linux commands

File system

1. **ls:** The ls command lists the content of the current directory
2. **cd:** Using cd changes the current directory to the one specified. You can use relative (i.e. cd directoryA) or absolute (i.e. cd /home/pi/directoryA) paths.
3. **pwd:** The pwd command displays the name of the present working directory entering pwd willoutput something like /home/pi.
4. **mkdir:** You can use mkdir to create a new directory, e.g. mkdir newDir would create the directory newDir in the present working directory.
5. **rmdir:** To remove empty directories, use rmdir. So, for example, rmdir oldDir will remove the directory oldDir only if it is empty.

6. **rm:** The command rm removes the specified file (or recursively from a directory when used with -r).
7. **cp:** Using cp makes a copy of a file and places it at the specified.
8. **ssh:** SSH denotes the secure shell. Connect to another computer using an encrypted network connection.
9. **scp:** The scp command copies a file from one computer to another using ssh.
10. **sudo:** The sudo command enables you to run a command as a superuser, or another user. Use sudo -s for a superuser shell. For more details see Root user / sudo
11. **dd:** The dd command copies a file converting the file as specified. It is often used to copy an entire disk to a single file or back again.
12. **df:** Use df to display the disk space available and used on the mounted filesystems. Use df -h to see the output in a human-readable format using M for MBs rather than showing number of bytes.
13. **unzip:** The unzip command extracts the files from a compressed zip file.
14. **wget:** Download a file from the web directly to the computer with wget. So wget https://www.raspberrypi.org/documentation/linux/usage/commands.mdwill download this file to your computer as commands.md
15. **grep:** Use grep to search inside files for certain search patterns.
16. **awk:** awk is a programming language useful for searching and manipulating text files.
17. **find:** The find command searches a directory and subdirectories for files matching certain patterns.

Networking

1. **ping:** The ping utility is usually used to check if communication can be made with another host. It can be used with default settings by just specifying a hostname (e.g. ping raspberrypi.org) or an IP address (e.g. ping 8.8.8.8). It can specify the number of packets to send with the -c flag.
2. **nmap:** nmap is a network exploration and scanning tool. It can return port and OS information about a host or a range of hosts. Running just nmap will display the options available as well as example usage.
3. **hostname:** The hostname command displays the current hostname of the system. A privileged(super) user can set the hostname to a new one by supplying it as an argument (e.g. hostname newhost).
4. **ifconfig:** Use ifconfig to display the network configuration details for the interfaces on the current system when run without any arguments (i.e. ifconfig). By supplying the command with the name of an interface (e.g. eth0 or lo) you can then alter the configuration: check the manual page for more details.

CONCLUSION:-

Thus Raspbian OS was installed, Raspberry Pi Components and Interface were studied and implemented, Raspberry was also connected to ethernet, Monitor, USB.

PRACTICAL - 2

AIM:-

Displaying different LED patterns with Raspberry Pi.

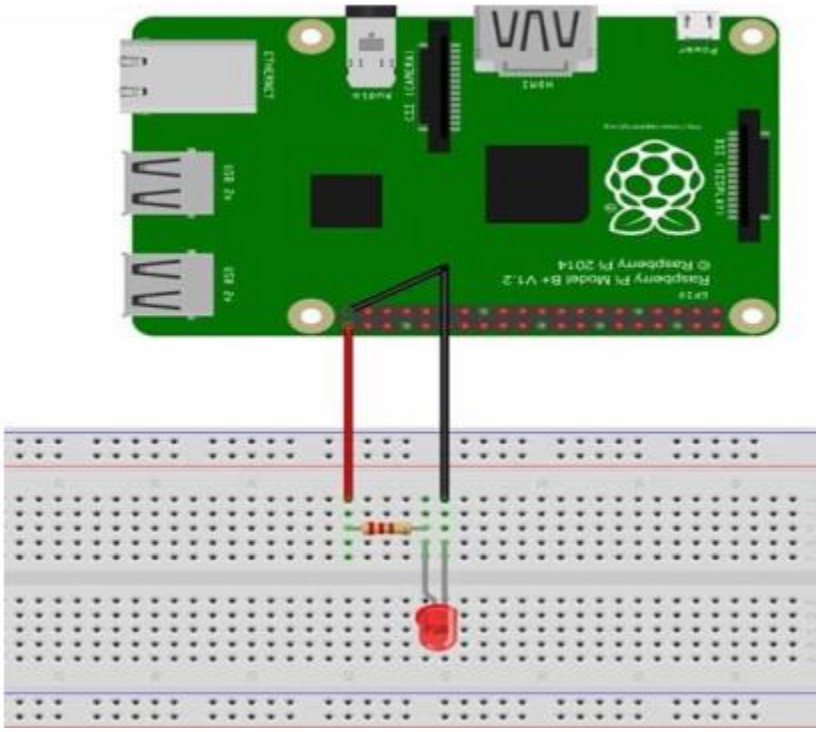
ABSTRACT:-

This project deals with displaying different LED pattern.

REQUIRED COMPONENTS:-

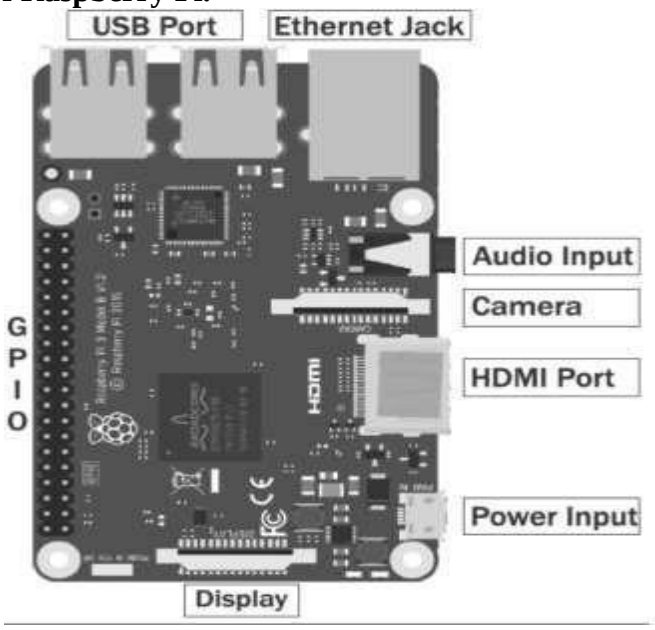
Particular	Quantity
Raspberry Pi3	1
Power Supply	1
USB Keyboard	1
USB Mouse	1
Micro SD card	1
LED	1
Resistor 330ohm	1

INTERFACING DIAGRAM:-



PROCEDURE:-

Step 1:- Connection of Raspberry Pi.



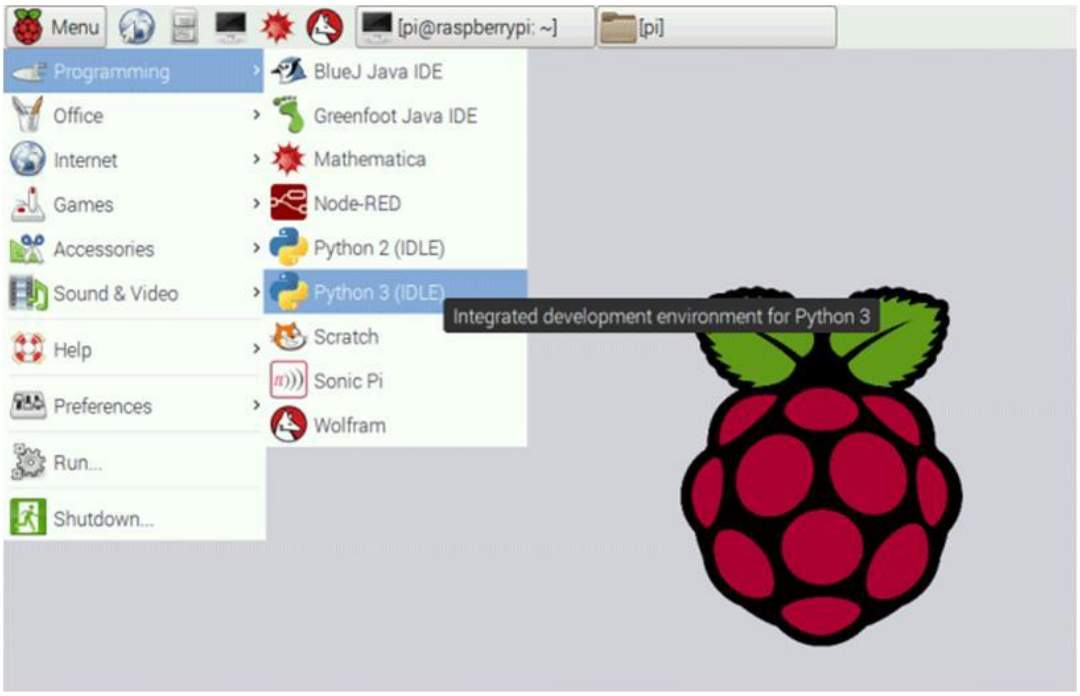
a. Connect Keyboard and mouse to USB Ports of Raspberry Pi.
b. Connect HDMI Port of Raspberry Pi to HDMI monitor via HDMI cable or HDMI to VGA convertor.
c. Connect the LED to the Raspberry Pi as per the interfacing diagram

Step 2:- Switch ON power supply.

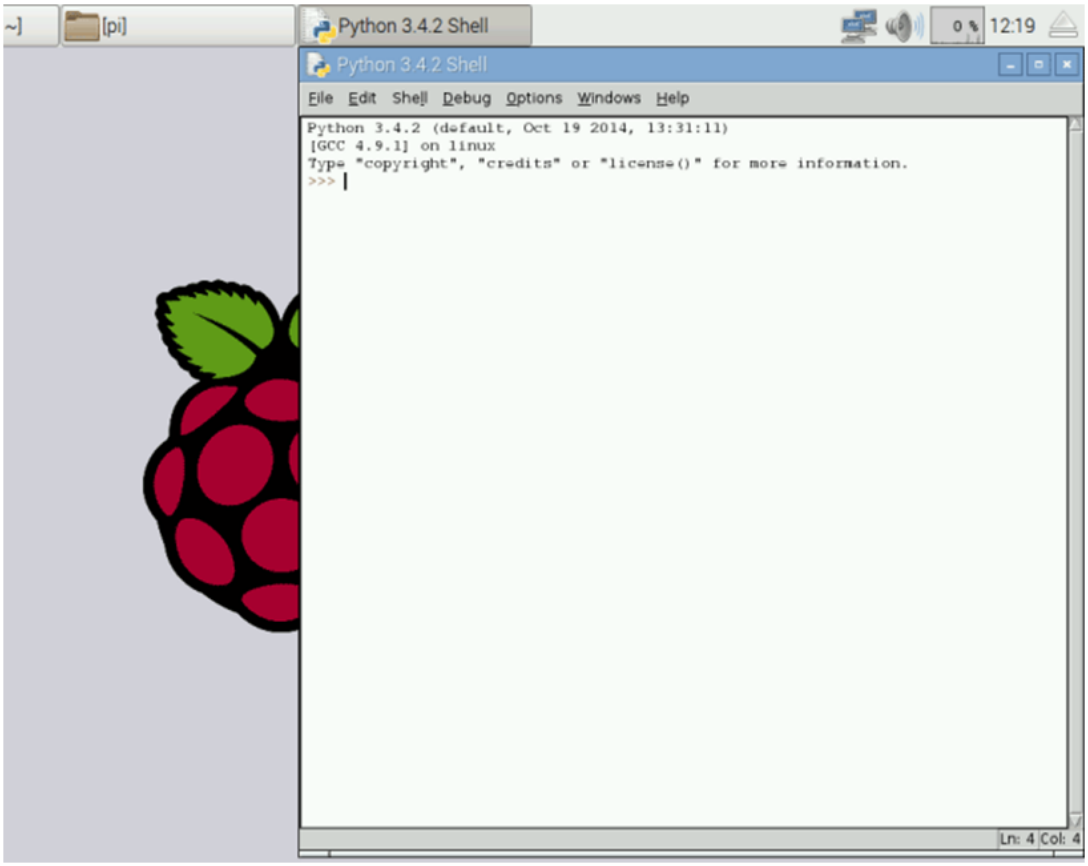
Step 3:- Login to Raspberry Pi

a. Username: pi
b. Password: raspberry

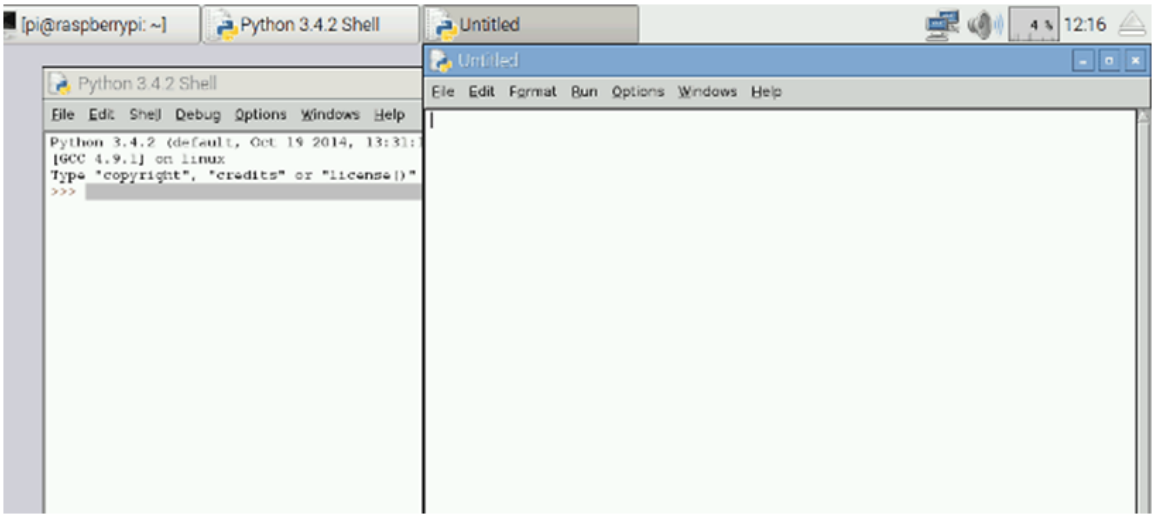
Step 4:- On the desktop, go the Start Menu and choose for the PYTHON 3, as shown in figure below



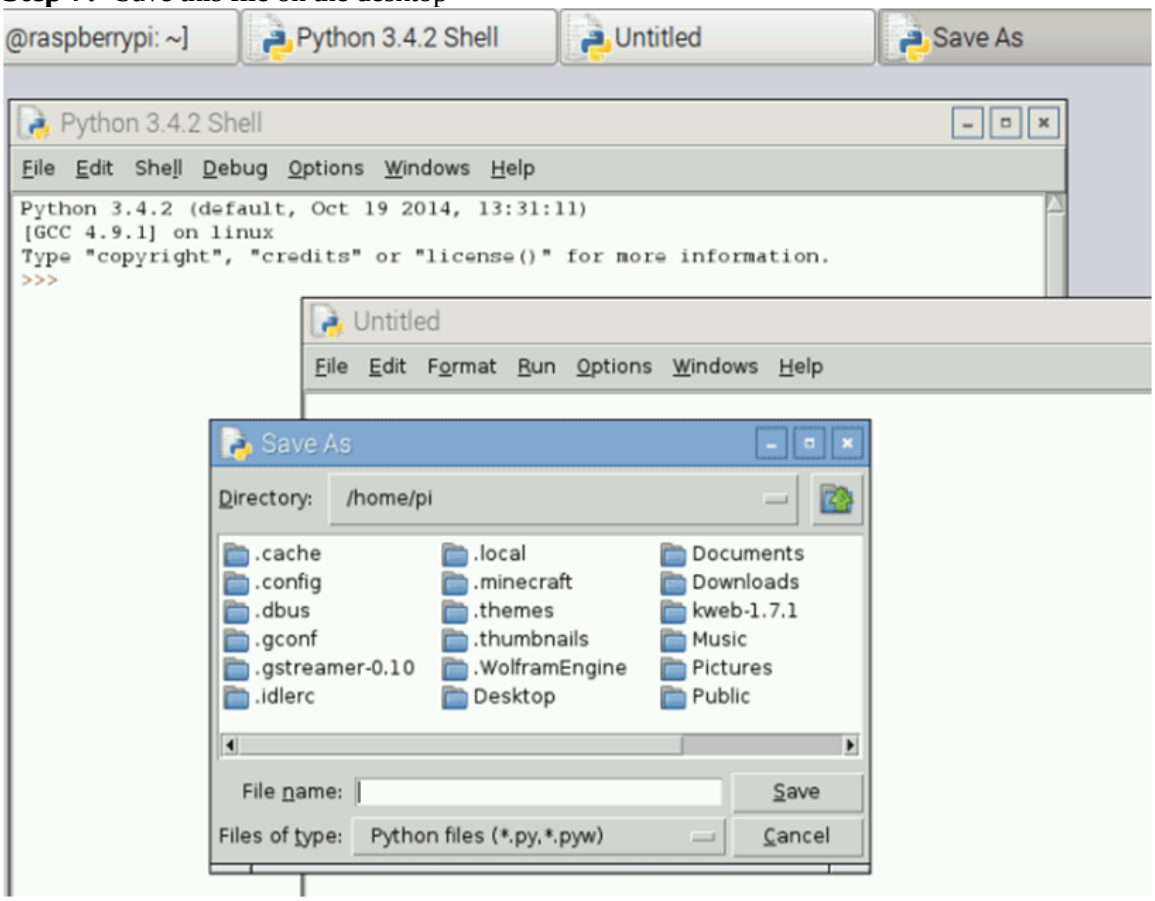
Step 5:- PYTHON will run and you will see a window as shown in below figure



Step 6:- After that, click on New File in File Menu, You will see a new Window open.



Step 7:- Save this file on the desktop



Step 7:- After that write the program as given below and execute the program by clicking on "RUN" on 'DEBUG' option

Step 8:- Type and run the program and see the output on LED.

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

successfully. By this time you should see the LED blinking three times. If there were any errors in the program, the execution tells to correct it. Once the error is corrected execute the program again.

We will see the PYTHON program Code for LED Blinking, in detail, below

PROGRAM:-

```
import RPi.GPIO as GPIO                    # Import Package import import time
#Import Time Class

GPIO.setmode(GPIO.BOARD)                   # Set Mode BOARD or BCM
GPIO.setwarnings(False)                    # Disable Warnings
GPIO.setup(40,GPIO.OUT)                    #Set Pin 11 as Output


while True:
        print ("LED on")                   # 'on' Print Message
        GPIO.output(40,GPIO.HIGH)          #Set GPIO pin High Turn on LED
        time.sleep(1)                      # wait for 1 Second
        print ("LED off")                  # 'off' Print Message
        GPIO.output(40,GPIO.LOW)           #Set GPIO pin Low Turn off LED
        time.sleep(1)                      # wait for 1 Second
```

CONCLUSION:-

Thus we have studied and displayed different LED patterns with Raspberry Pi.

PRACTICAL- 3

AIM:-

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

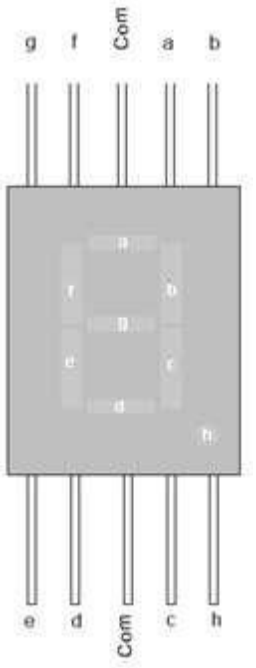
ABSTRACT:-

This project deals with Interfacing of 4-digit Seven Segment Display Module with Raspberry Pi and display Time over it**.** Only two data wires are required for I2C – a data line (SDA) and a clock line (SCL).

REQUIRED COMPONENTS:-

Particular	Quantity
Raspberry Pi3	1
Power Supply	1
USB Keyboard	1
USB Mouse	1
Micro SD card	1
4-Digit 7-Segment Display Module	1
Jumper wires	4

INTERFACING DIAGRAM:



4-digit 7-Segment Display Module

PROCEDURE:-

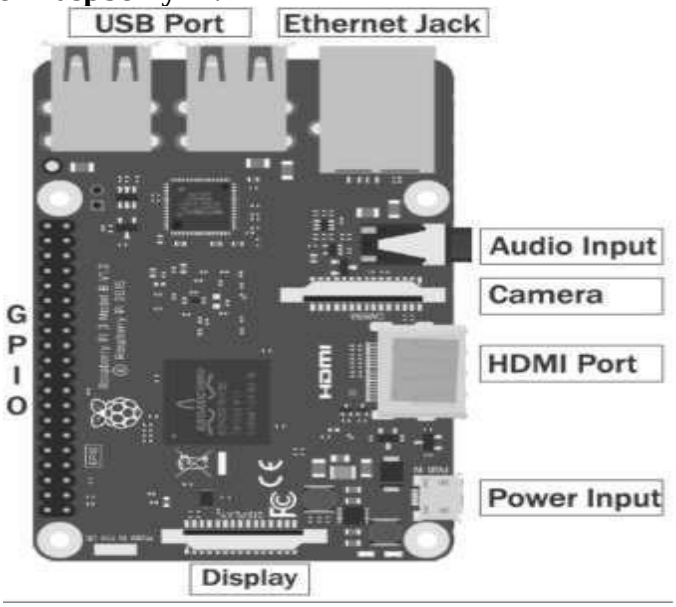
Step 1:- Connection of Discover Development Board

a. Connect 4-Digit 7 segment display board to your Raspberry Pi according to the following table:

TM1637 Board Pin	Position	RPI Pin	Raspberry Function
GND	Ground	14	GND
VCC	+5V Power	4	5V
D10	Data In	24	GPIO 24
CLK	Clock	23	GPIO 23

b. Connect Power Supply

Step 2:- Connection of Raspberry Pi.



a. Connect Keyboard and mouse to USB Ports of Raspberry Pi.
b. Connect HDMI Port of Raspberry Pi to HDMI monitor via HDMI cable or HDMI to VGA convertor.

Step 3:- Switch ON power supply.

Step 4:- Login to Raspberry Pi

Step 5:- Create a new file with an extension .py.Also copy and keep the files TM1637.py and time.py file in the same folder.

Step 6:- Open the file with Python 2 IDLE only.

Step 7:- Type and run the program, and see the output on 4-Digit 7 segment display board.

PROGRAM:-

```
import sys
import time
import datetime
import RPi.GPIO as GPIO
import tm1637


#CLK -> GPIO23 (Pin 16)
#Di0 -> GPIO24 (Pin 18)

Display = tm1637.TM1637(23,24,tm1637.BRIGHT_TYPICAL)
Display.Clear( )
Display.SetBrightnes(1)
while(True):
        now = datetime.datetime.now()
        hour = now.hour minute = now.minute second = now.second
        currenttime = [ int(hour / 10), hour % 10, int(minute / 10), minute % 10 ]

        Display.Show(currenttime)
        Display.ShowDoublepoint(second % 2)

        time.sleep(1)
```

CONCLUSION:-

Thus we have studied and displayed Time over 4-Digit 7-Segment Display using Raspberry Pi.

PRACTICAL - 4

AIM:-

Control Raspberry Pi via Telegram Messenger.

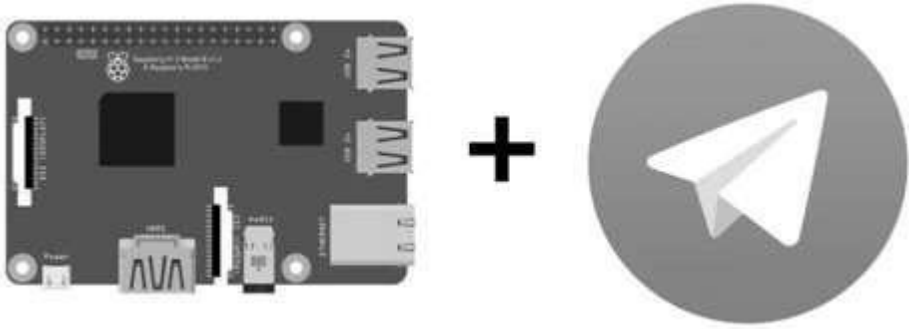
ABSTRACT:-

This project deals with how to automatically run scripts for text commands via Telegram. This allows you to, for example, easily query the status of your Pi's on the go or simply perform actions remotely via your messenger.

REQUIRED COMPONENTS:-

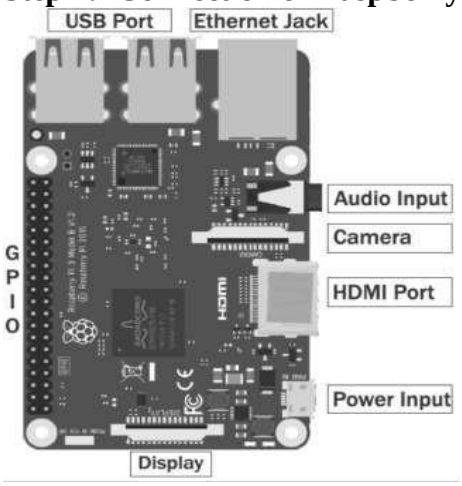
Particular	Quantity
Raspberry Pi3	1
Power Supply	1
USB Keyboard	1
USB Mouse	1
Micro SD card	1
Telegram Messenger	1

INTERFACING DIAGRAM:-



PROCEDURE:-

Step 1:- Connection of Raspberry Pi.



a. Connect Keyboard and mouse to USB Ports of Raspberry Pi.

b. Connect HDMI Port of Raspberry Pi to HDMI monitor via HDMI cable or HDMI to VGA convertor.

Step 2:- Switch ON power supply.

Step 3:- Login to Raspberry Pi

a. Username: pi
b. Password: raspberry

Procedure to install “Telegram App” on your smart phone and preparing “Raspberry Pi” to control Lamp, Fan, etc.

Step 4:- Installing Telegram App on your smart phone(Only for Android smartphone, WIFI or mobile data should be ON)

Open “playstore” app in your android mobile.

Search for “Telegram” in search option.

Install the “Telegram” app by clicking on install button highlighted in green color.

Open the telegram app in your mobile.

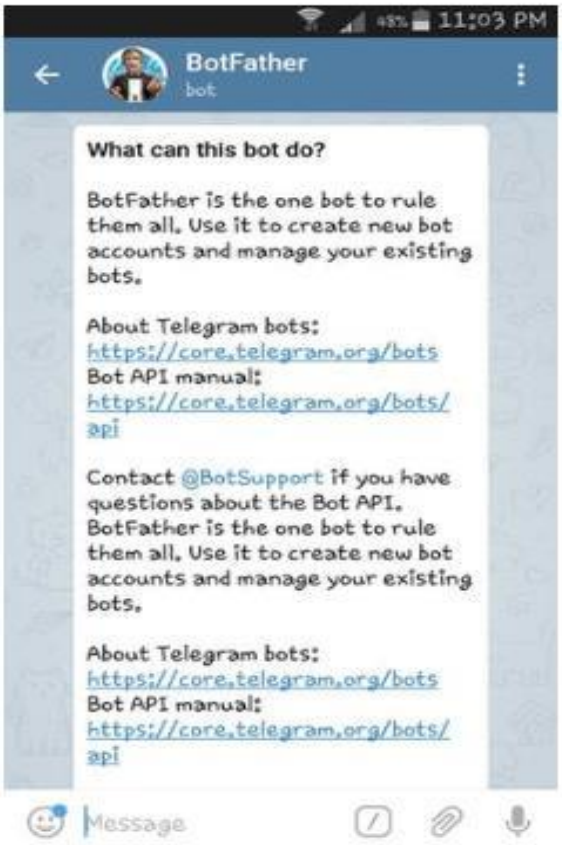
Click on “START MESSAGING” button.

Enter your “mobile number” to register with telegram service.

Step 5:- Now, configure “Telegram App” for controlling raspberry pi.

Now, go in “Search option” in your telegram app and search for the name “BotFather”

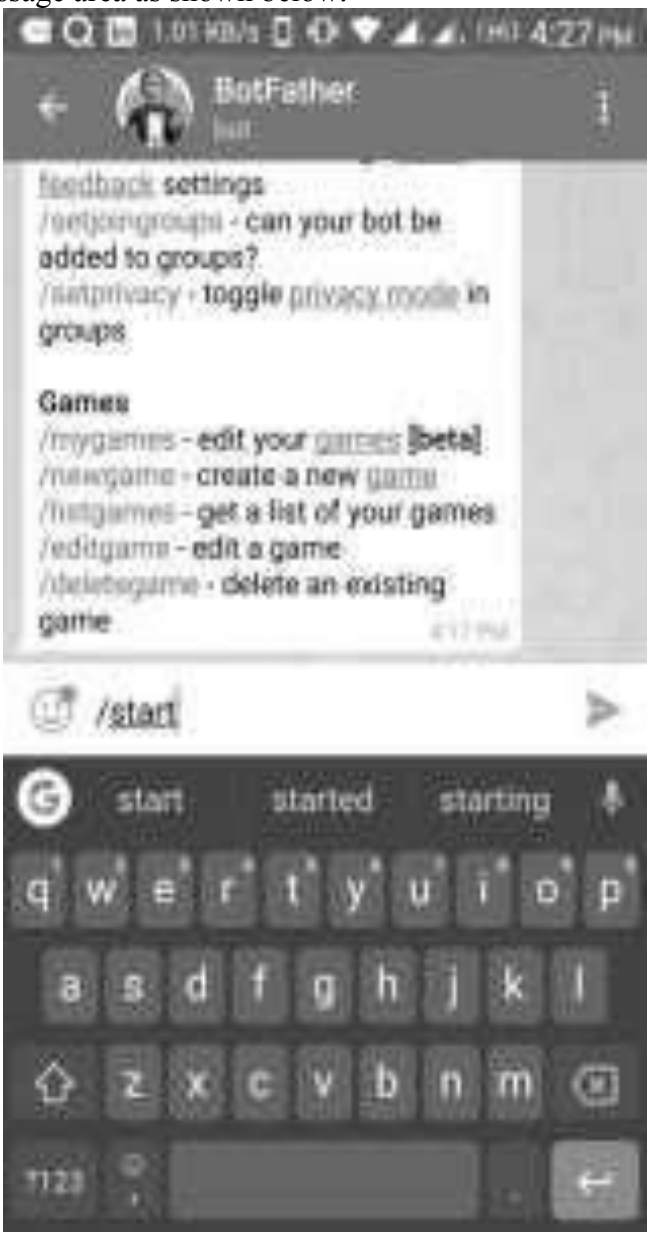
Click on “BotFather” and you will see the screen below



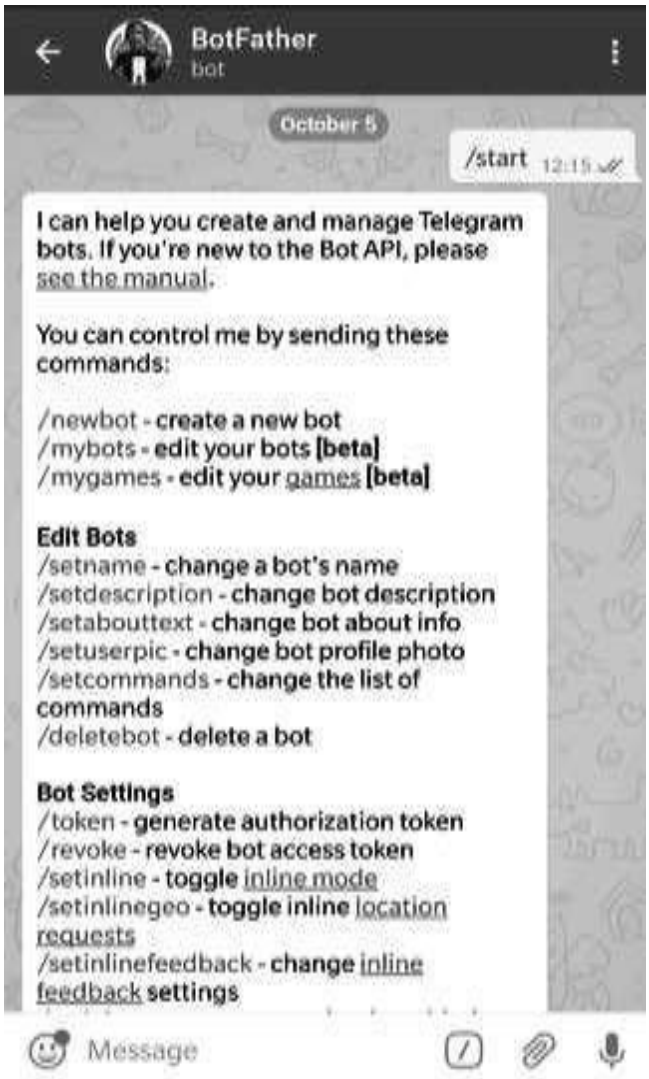
- Click on “START” and you will get some chat information from telegram read it carefully and understand it properly for using the telegram. More information you will find on telegram website and its online tutorials.
- Now carefully follow the steps to generate “Token Key” so that you can connect with your raspberry pi.
- The same “Token Key” will be added to your python program script to control relays.

NOTE: *All Telegram commands starts with “/” { example:* /start }

- Now type “/start” in message area as shown below.



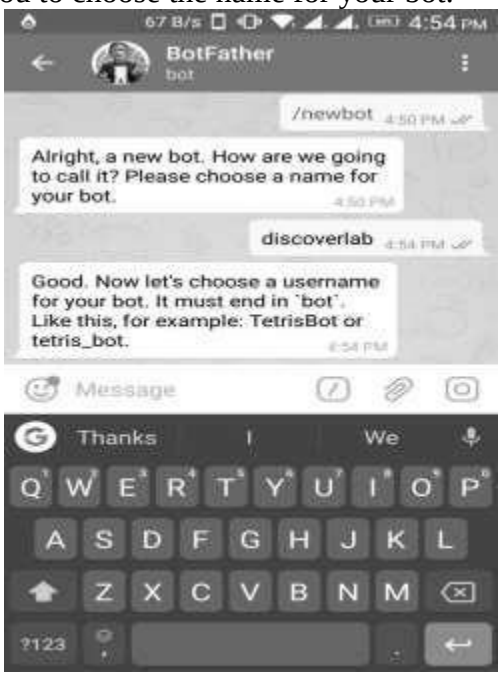
- You will see something as shown below after above step.



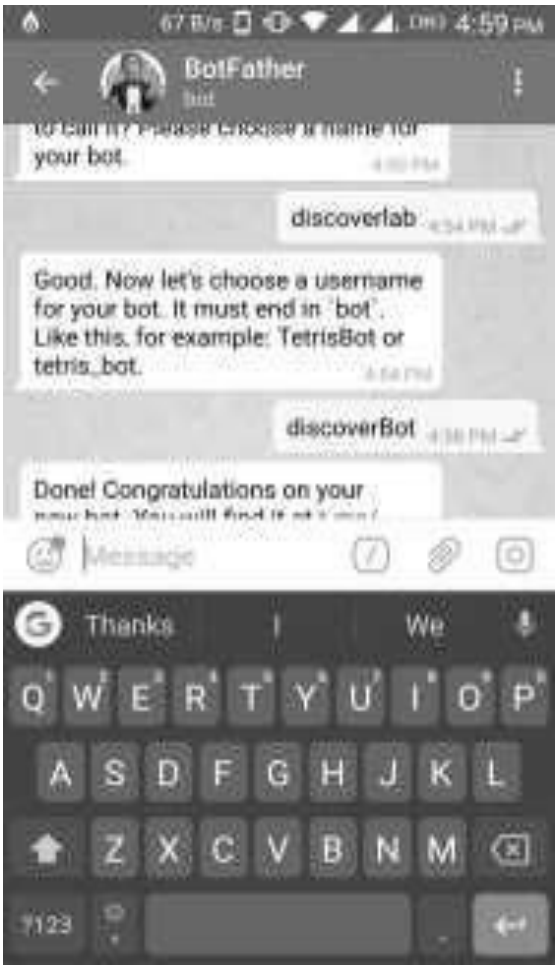
- Now, type "/newbot" in message area as shown below and enter.

- You will see screen as shown below.

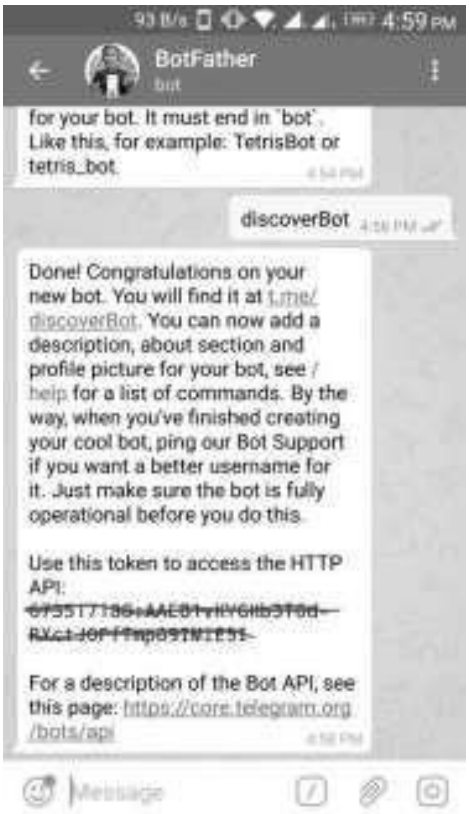
- Now, “botfather ” is asking you to choose the name for your bot.



- Now, “botfather” is asking username for your bot. I have reply with username “discoverBot”.
- You use your own “username”



- Thus, you will see reply from “botfather” as shown below.
- Same way you have to choose name for your “Bot”. Here I have use “discoverBot”.



- Thus for security reason, I have hidden my token key as shown above.
- And note down your token key somewhere in text editor or book so that you can copy it to your python script.

Step 6:- Setup Raspberry Pi

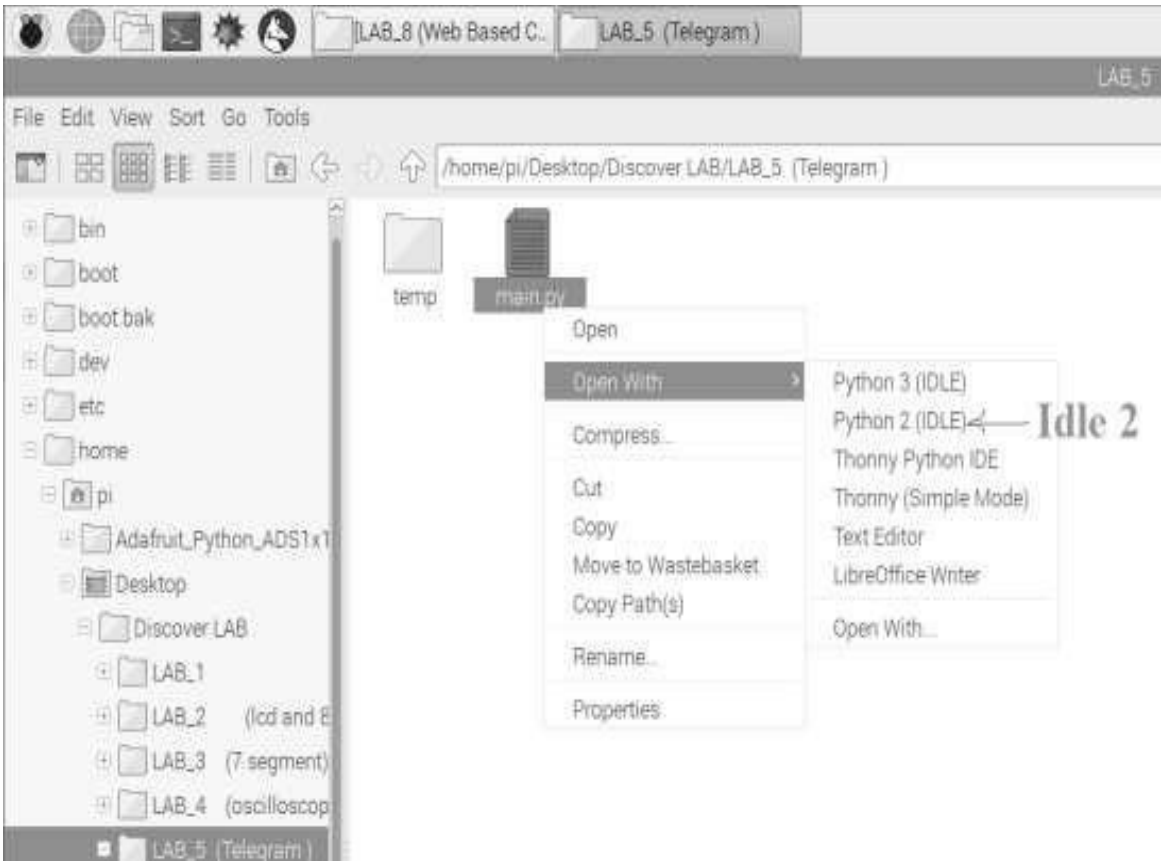
- Install "Python Package Index and teleport using:
  - sudo apt-get update
  - sudo apt-get upgrade
  - sudo apt-get install python-pip
  - sudo pip install telepot
- Test your bot using python 2 IDLE and type:

```
Import telepot
bot=teleport.Bot('BoT Token')
bot.getMe ()
```

If it prints your bot details, means everything is correct. If not token is wrong.

Step 7:- Run the program in Raspberry Pi

- Right click and create python file with name "main.py" which steps are shown below and open with python idle 2 as shown below in picture.



Make the below change in the program

- Replace current Token Key with your own key.
- Finally, you check all the hardware connection and run the program and enjoy controlling your devices like fan and light.

PROGRAM:-

```
import datetime

import teleport

import time

from telepot.loop import MessageLoop

import RPi.GPIO as IO

from time import sleep

IO.setmode(IO.BCM)

relay = 21

relay2= 20

IO.setup(relay,IO.OUT)

IO.setup(relay2,IO.OUT)

now=datetime.datetime.now()
```

```
def handle(msg):
    chat_id = msg['chat']['id']
    command=msg['text']

    print('Received')
    print(command)

    if command == '/hi':
        telegram_bot.sendMessage (chat_id, str("Hi!"))
    elif command == '/time':
        telegram_bot.sendMessage(chat_id, str(now.hour)+str(":")+str(now.minute))
    elif command == '/relayon':
        telegram_bot.sendMessage (chat_id, str("Relay1 On"))
        IO.output(relay,True)
    elif command == '/relayoff':
        telegram_bot.sendMessage (chat_id, str("Relay1 Off"))
        IO.output(relay,False)
telegram_bot = telepot.Bot('Token ID')
print (telegram_bot.getMe())
MessageLoop(telegram_bot, action).run_as_thread()
print ('Up and Running....')
while 1:
    time.sleep(10)
```

CONCLUSION:-

Thus we have configured Telegram Messenger to Raspberry Pi.

PRACTICAL - 4

AIM:-

Control Raspberry Pi via Telegram Messenger.

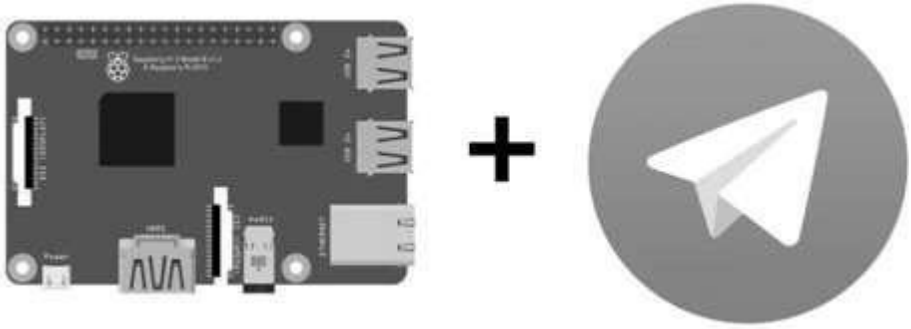
ABSTRACT:-

This project deals with how to automatically run scripts for text commands via Telegram. This allows you to, for example, easily query the status of your Pi's on the go or simply perform actions remotely via your messenger.

REQUIRED COMPONENTS:-

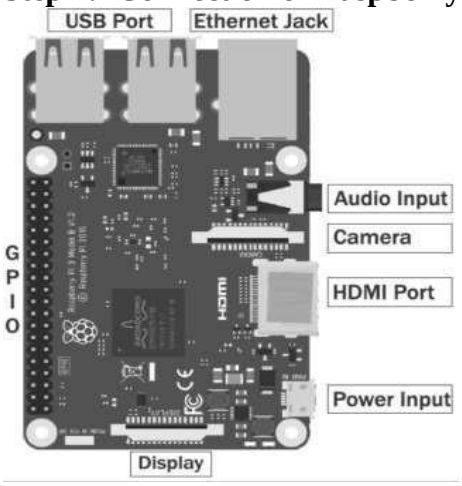
Particular	Quantity
Raspberry Pi3	1
Power Supply	1
USB Keyboard	1
USB Mouse	1
Micro SD card	1
Telegram Messenger	1

INTERFACING DIAGRAM:-



PROCEDURE:-

Step 1:- Connection of Raspberry Pi.



a. Connect Keyboard and mouse to USB Ports of Raspberry Pi.

b. Connect HDMI Port of Raspberry Pi to HDMI monitor via HDMI cable or HDMI to VGA convertor.

Step 2:- Switch ON power supply.

Step 3:- Login to Raspberry Pi

a. Username: pi
b. Password: raspberry

Procedure to install “Telegram App” on your smart phone and preparing “Raspberry Pi” to control Lamp, Fan, etc.

Step 4:- Installing Telegram App on your smart phone(Only for Android smartphone, WIFI or mobile data should be ON)

Open “playstore” app in your android mobile.

Search for “Telegram” in search option.

Install the “Telegram” app by clicking on install button highlighted in green color.

Open the telegram app in your mobile.

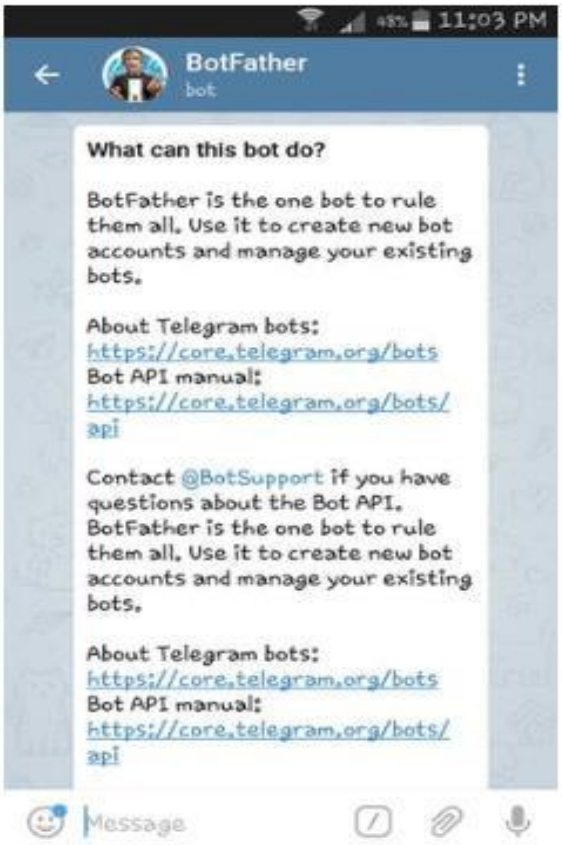
Click on “START MESSAGING” button.

Enter your “mobile number” to register with telegram service.

Step 5:- Now, configure “Telegram App” for controlling raspberry pi.

Now, go in “Search option” in your telegram app and search for the name “BotFather”

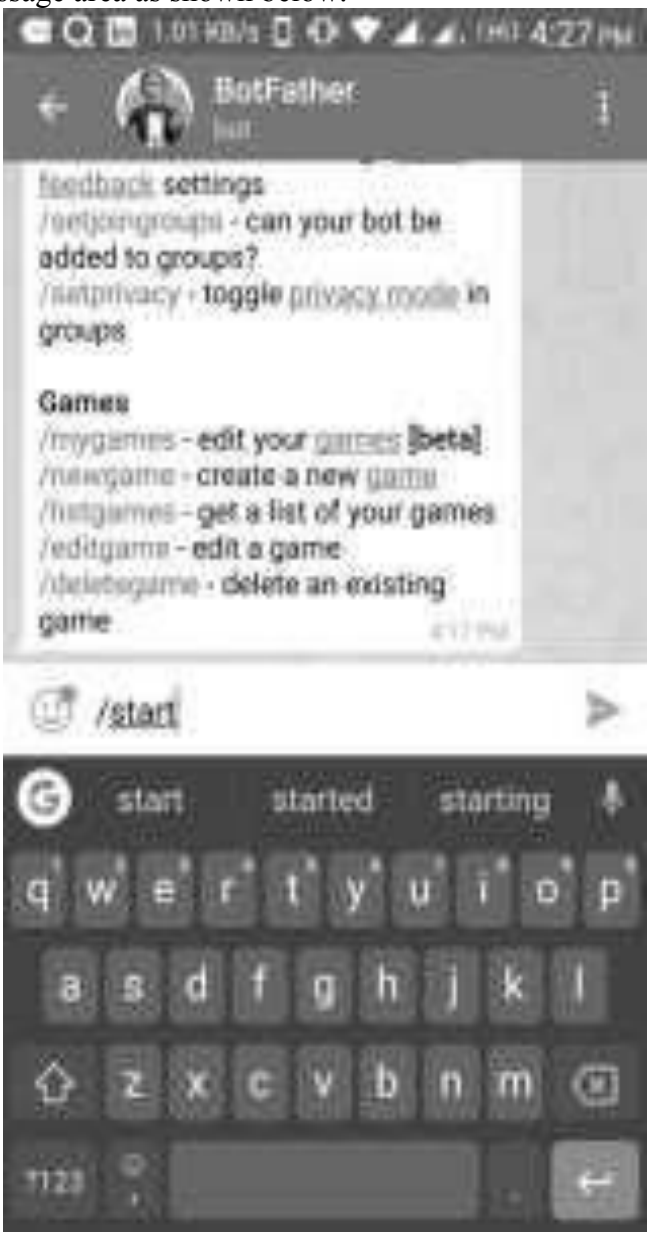
Click on “BotFather” and you will see the screen below



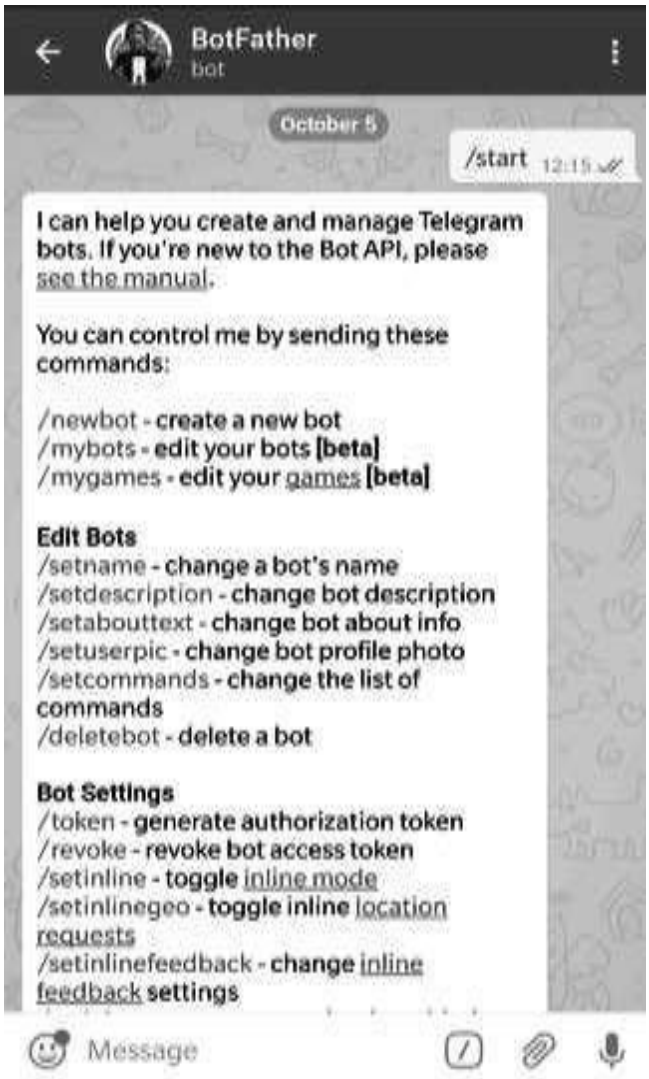
- Click on “START” and you will get some chat information from telegram read it carefully and understand it properly for using the telegram. More information you will find on telegram website and its online tutorials.
- Now carefully follow the steps to generate “Token Key” so that you can connect with your raspberry pi.
- The same “Token Key” will be added to your python program script to control relays.

NOTE: *All Telegram commands starts with “/” { example:* /start }

- Now type “/start” in message area as shown below.



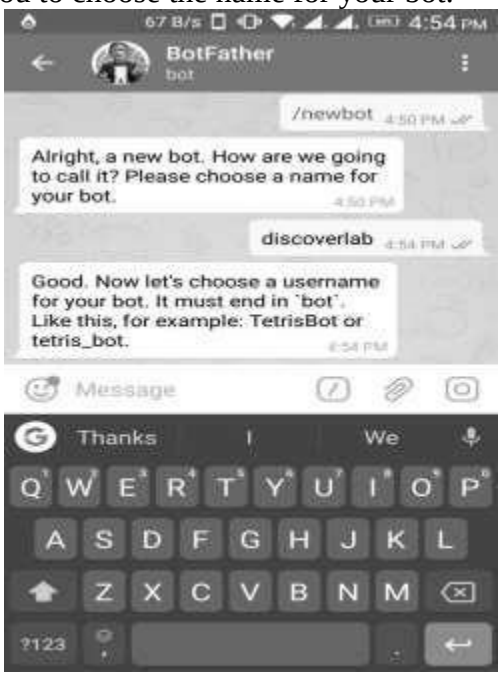
- You will see something as shown below after above step.



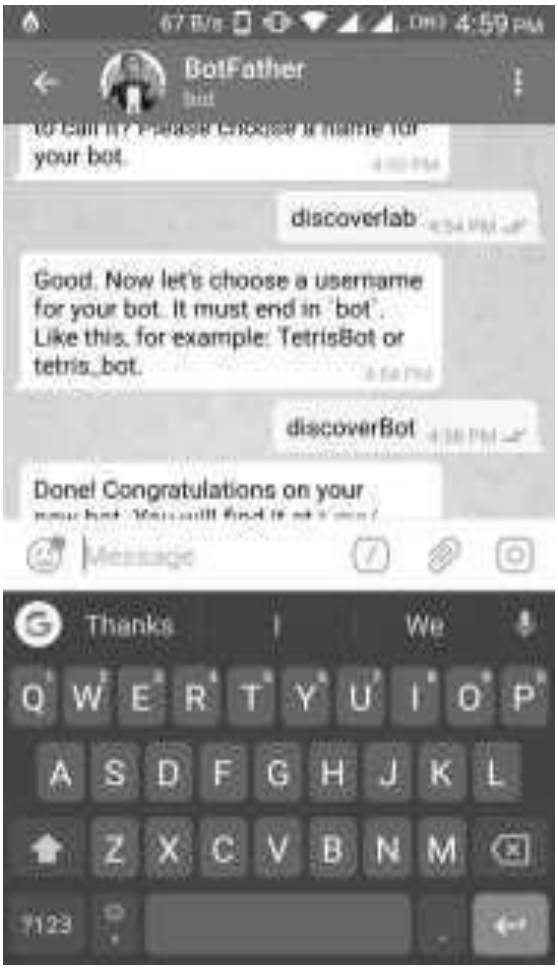
- Now, type "/newbot" in message area as shown below and enter.

- You will see screen as shown below.

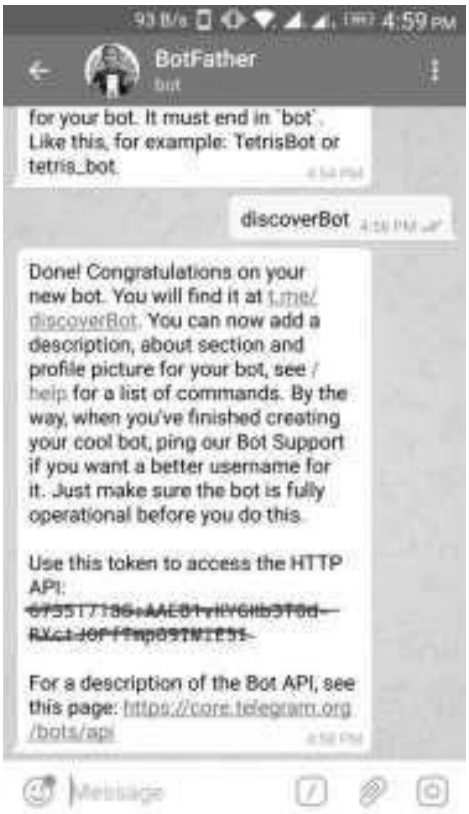
- Now, “botfather ” is asking you to choose the name for your bot.



- Now, “botfather” is asking username for your bot. I have reply with username “discoverBot”.
- You use your own “username”



- Thus, you will see reply from “botfather” as shown below.
- Same way you have to choose name for your “Bot”. Here I have use “discoverBot”.



- Thus for security reason, I have hidden my token key as shown above.
- And note down your token key somewhere in text editor or book so that you can copy it to your python script.

Step 6:- Setup Raspberry Pi

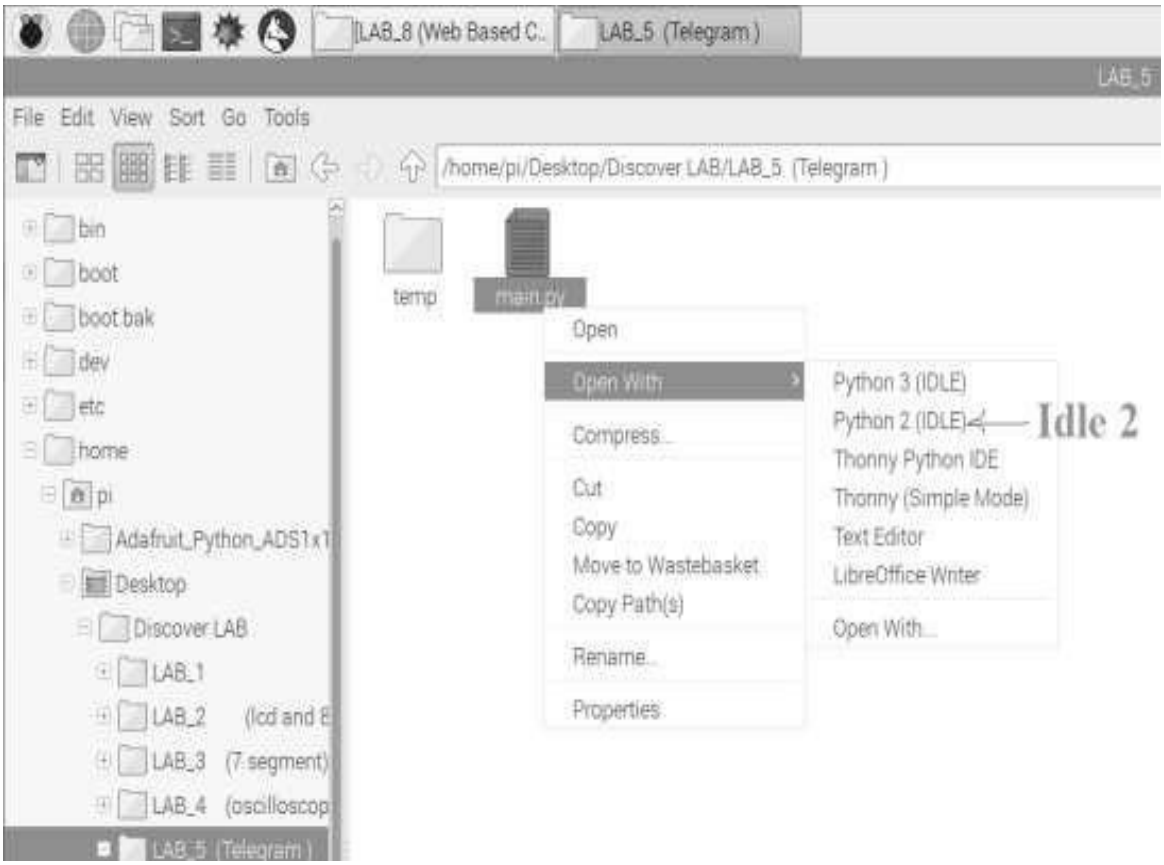
- Install "Python Package Index and teleport using:
  - sudo apt-get update
  - sudo apt-get upgrade
  - sudo apt-get install python-pip
  - sudo pip install telepot
- Test your bot using python 2 IDLE and type:

```
Import telepot
bot=teleport.Bot('BoT Token')
bot.getMe ()
```

If it prints your bot details, means everything is correct. If not token is wrong.

Step 7:- Run the program in Raspberry Pi

- Right click and create python file with name "main.py" which steps are shown below and open with python idle 2 as shown below in picture.



Make the below change in the program

- Replace current Token Key with your own key.
- Finally, you check all the hardware connection and run the program and enjoy controlling your devices like fan and light.

PROGRAM:-

```
import datetime

import teleport

import time

from telepot.loop import MessageLoop

import RPi.GPIO as IO

from time import sleep

IO.setmode(IO.BCM)

relay = 21

relay2= 20

IO.setup(relay,IO.OUT)

IO.setup(relay2,IO.OUT)

now=datetime.datetime.now()
```

```
def handle(msg):
    chat_id = msg['chat']['id']
    command=msg['text']

    print('Received')
    print(command)

    if command == '/hi':
        telegram_bot.sendMessage (chat_id, str("Hi!"))
    elif command == '/time':
        telegram_bot.sendMessage(chat_id, str(now.hour)+str(":")+str(now.minute))
    elif command == '/relayon':
        telegram_bot.sendMessage (chat_id, str("Relay1 On"))
        IO.output(relay,True)
    elif command == '/relayoff':
        telegram_bot.sendMessage (chat_id, str("Relay1 Off"))
        IO.output(relay,False)
telegram_bot = telepot.Bot('Token ID')
print (telegram_bot.getMe())
MessageLoop(telegram_bot, action).run_as_thread()
print ('Up and Running....')
while 1:
    time.sleep(10)
```

CONCLUSION:-

Thus we have configured Telegram Messenger to Raspberry Pi.

PRACTICAL- 5

AIM:-

Setting up Wireless Access Point using Raspberry Pi.

ABSTRACT:-

To access the Raspberry pi desktop with a remote Desktop connection.

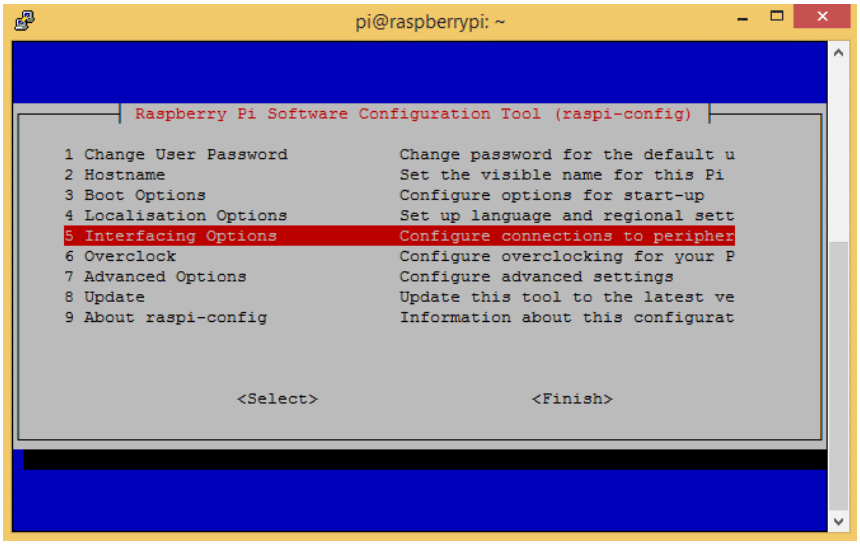
REQUIRED COMPONENTS:-

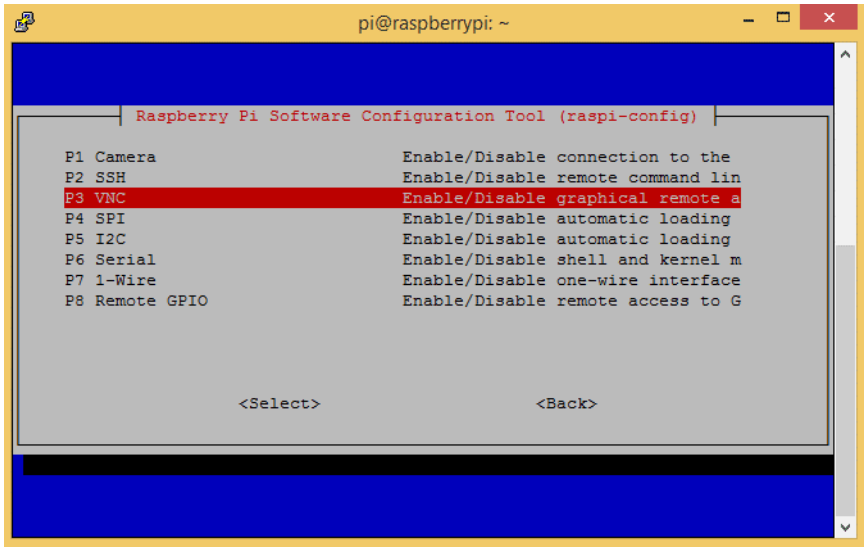
Particular	Quantity
Raspberry Pi3	1
Power Supply	1
USB Keyboard	1
USB Mouse	1
Micro SD card	1

PROCEDURE:-

Step 1:- Enabling VNC in RPi

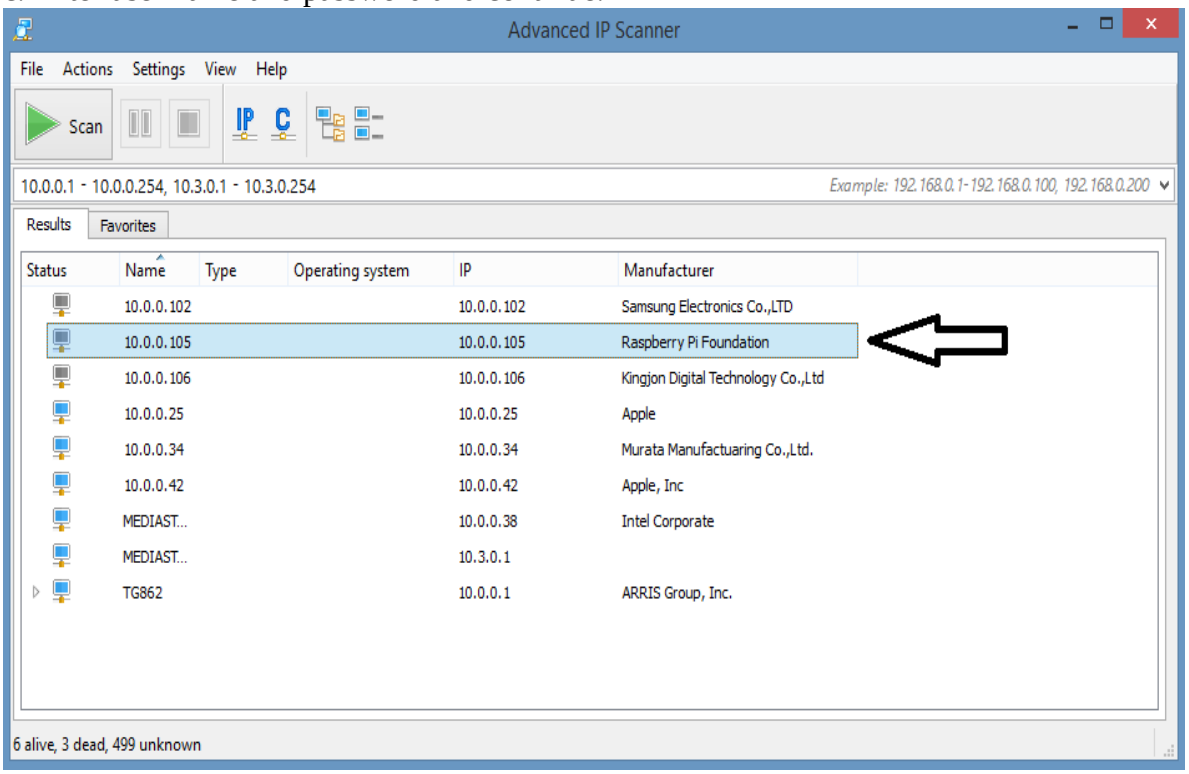
a. Enter sudo raspi-config at the command prompt to access it.
b. Then select “Interfacing Options” from the menu
c. Then select “VNC”, to enable VNC
d. Now reboot the Pi with sudo reboot

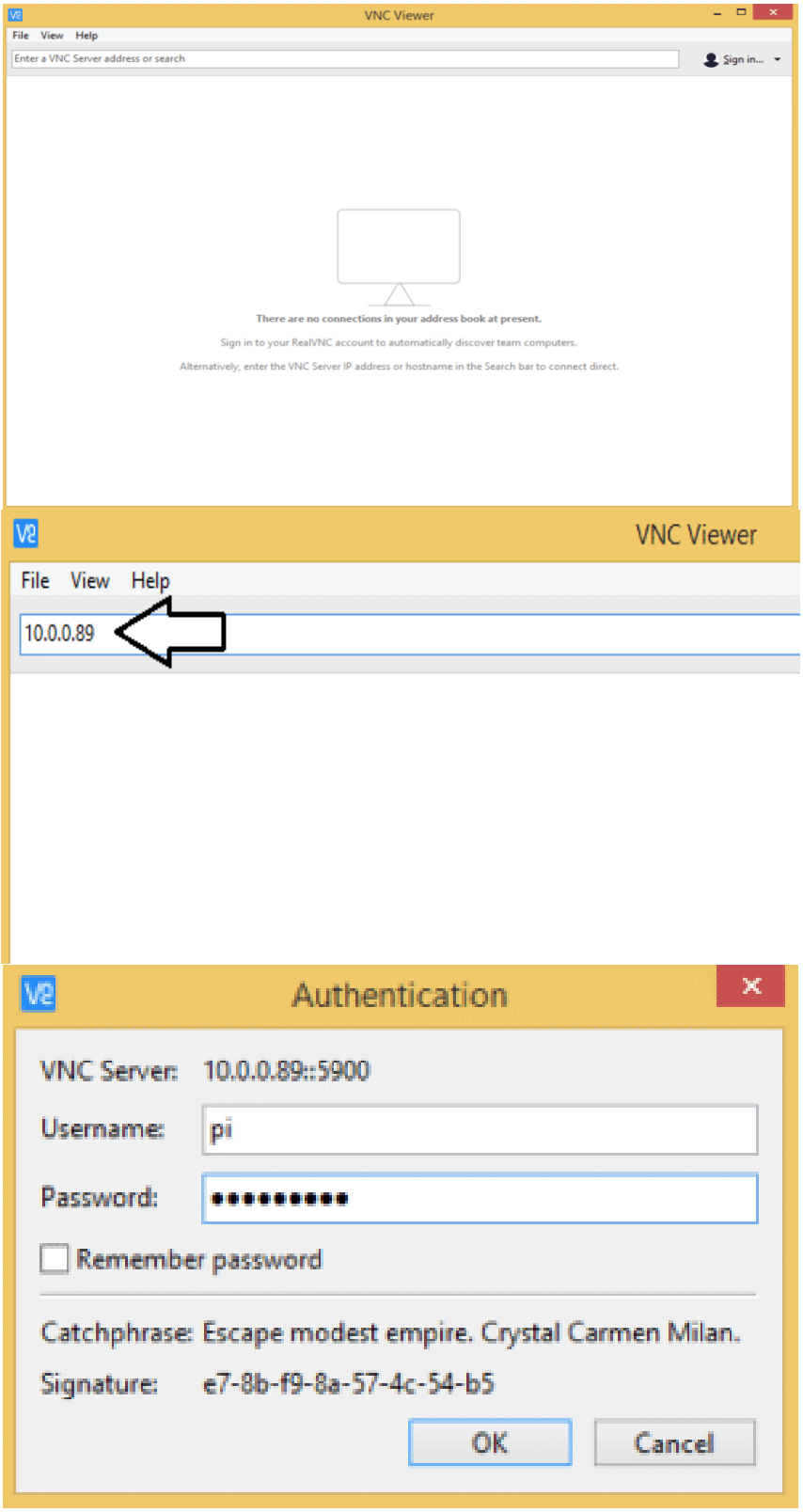


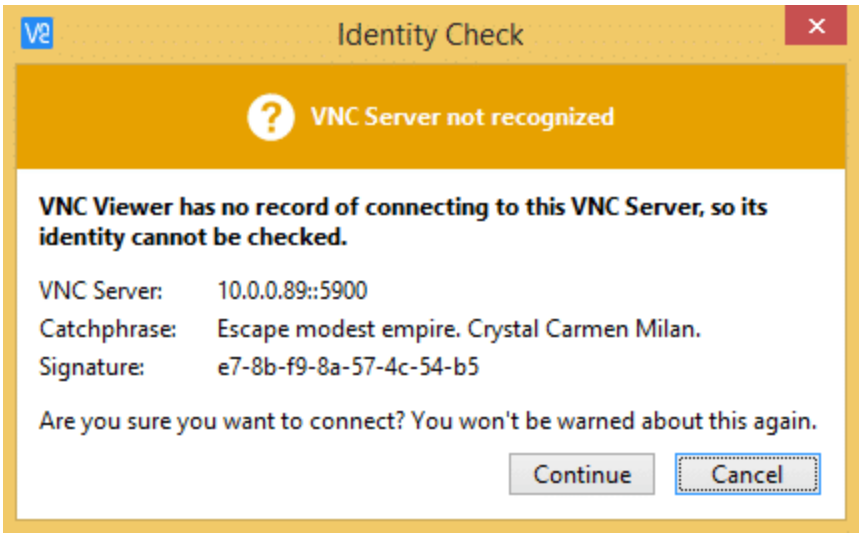


Step 2:- Install RealVNC Viewer in Raspberry Pi

a. Get the IP address of your Pi using IP scanner.
b. Enter IP address and you'll be prompted for the Pi's username and password.
c. Enter user name and password and continue.




VNC Viewer
File View Help
Enter a VNC Server address or search
Sign in...
There are no connections in your address book at present.
Sign in to your RealVNC account to automatically discover team computers.
Alternatively, enter the VNC Server IP address or hostname in the Search bar to connect direct.
VNC Viewer
File View Help
10.0.0.89
Authentication
VNC Server: 10.0.0.89::5900
Username: pi
Password: ••••••••••
Remember password
Catchphrase: Escape modest empire. Crystal Carmen Milan.
Signature: e7-8b-f9-8a-57-4c-54-b5
OK
Cancel



Step 3:- The Raspbian Desktop will open up in a VNC Viewer window:

CONCLUSION:-

Thus we have studied setting up Wireless Access Point using Raspberry Pi.

PRACTICAL - 6

AIM:-

Fingerprint Sensor interfacing with Raspberry Pi.

ABSTRACT:-

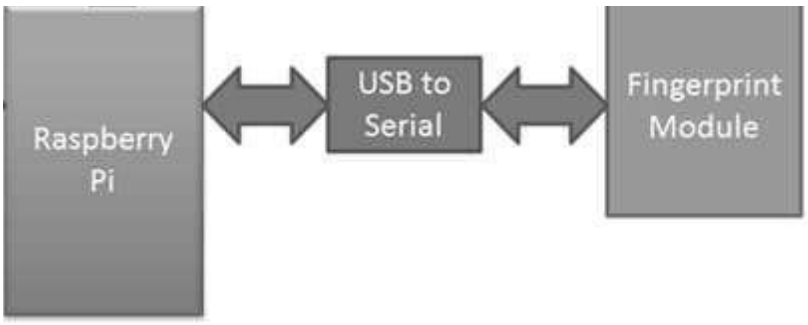
Finger Print Sensor, is now become very common to verify the identity of a person for various purposes. In present time we can see fingerprint-based systems everywhere in our daily life like for attendance in offices, employee verification in banks, for cash withdrawal or deposits in ATMs, for identity verification in government offices etc. Using this Raspberry Pi Fingerprint System, we can enrol new finger prints in the system and can delete the already fed finger prints.

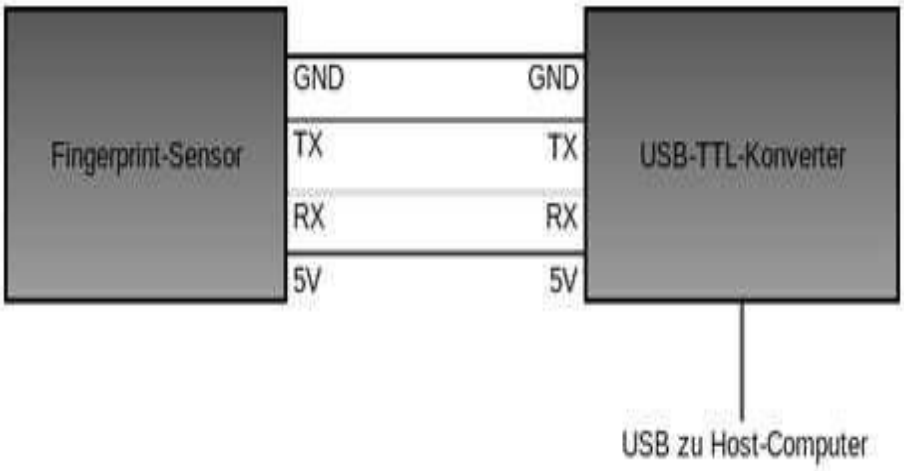
REQUIRED COMPONENTS:-

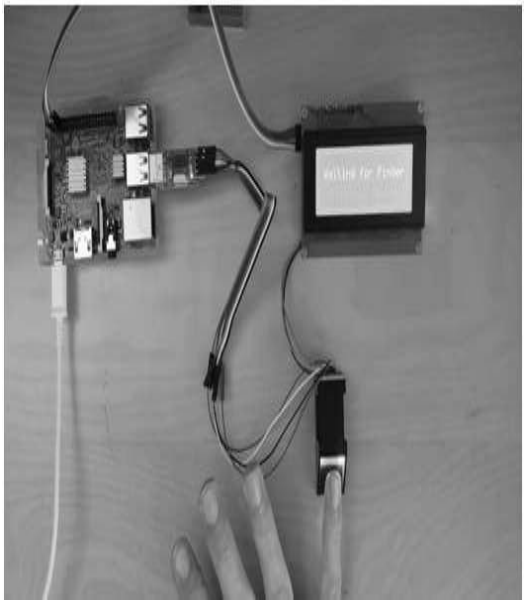
Particular	Quantity
Raspberry Pi3	1
Power Supply	1
USB Keyboard	1
USB Mouse	1
Micro SD card	1
Discover Board	1
USB to TTL converter	1
Fingerprint Module	1

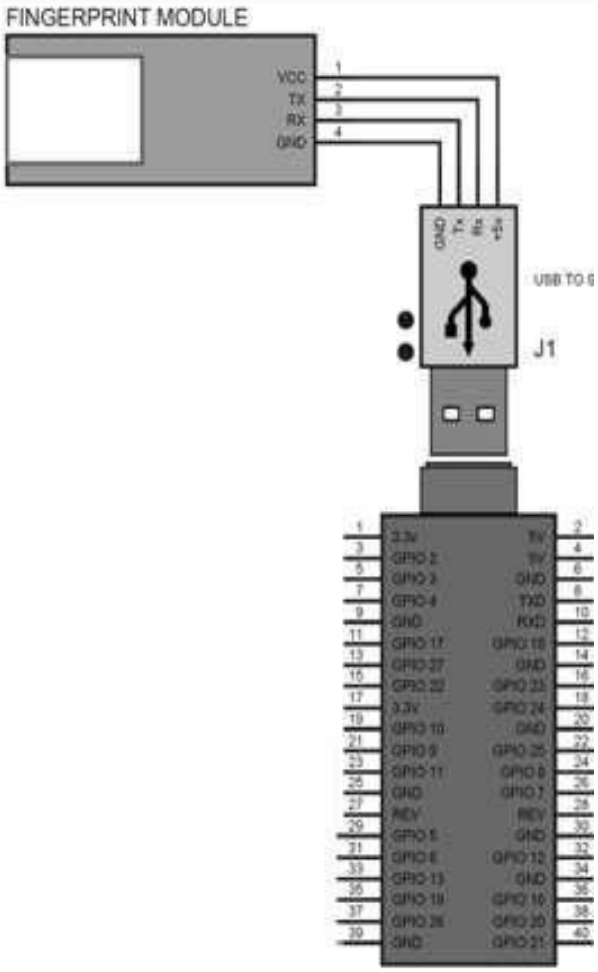
The Raspberry Pi has two pins (pin 8 / GPIO14 and pin 10 / GPIO 15), but they work with 3.3V. Since there are different fingerprint sensors, which do not all work with 3.3V, a USB UART converter is recommended.

INTERFACING DIAGRAM:-



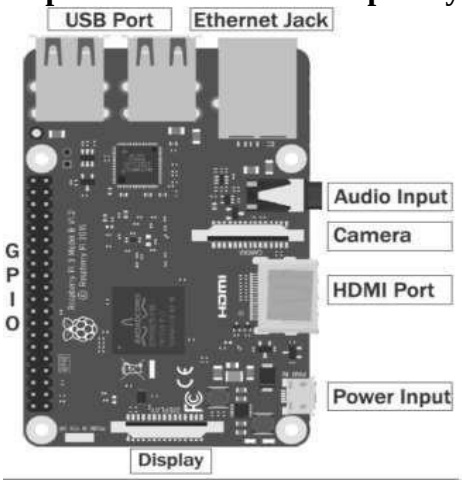

Fingerprint-Sensor
GND
GND
TX
TX
RX
RX
5V
5V
USB-TTL-Konverter
USB zu Host-Computer




FINGERPRINT MODULE
VCC
TX
RX
GND
J1

PROCEDURE:-

Step 1:- Connection of Raspberry Pi.



a. Connect Keyboard and mouse to USB Ports of Raspberry Pi.

b. Connect HDMI Port of Raspberry Pi to HDMI monitor via HDMI cable or HDMI to VGA convertor.

Step 2:- Switch ON power supply.

Step 3:- Login to Raspberry Pi

a. Username: pi
b. Password: raspberry

Step 4:- Step to check whether finger print sensor is installed properly.

To check whether the cabling is correct and whether the sensor is detected, you can open your console and perform the following before and after connecting:

ls /dev/ttyUSB*

If no other serial devices are connected via USB, nothing should be displayed first and afterwards /dev/ttyUSB0.

Step 4:- Installation steps for Raspberry Pi finger print module

- **Installation of the Raspberry Pi Fingerprint Library**

For some commands of the installation, root privileges are required. Therefore we start a terminal session and type the following, which executes all the following commands as root:

sudo bash

- **Add the necessary package**

  wget -O - http://apt.pm-codeworks.de/pm-codeworks.de.gpg | apt-key add -

  wget http://apt.pm-codeworks.de/pm-codeworks.list -P /etc/apt/sources.list.d/

- **Need to update the Raspberry pi and install the downloaded finger print sensor library:**

  sudo apt-get update

  sudo apt-get install python-fingerprint

- **If an error has occurred (in particular, that not all dependent packages have been installed), then execute the following:**

  sudo apt-get -f install

- **To return to the normal shell (under the Pi user), type exit.**

- **Attached are sample files for storing a new fingerprint, reading out and deleting stored fingerprints. Let's begin by recording a finger. Call the following:**
  - **python /usr/share/doc/python-fingerprint/examples/example_enroll.py**

- **Put your finger on the glass surface, wait for the instruction in the terminal and remove your finger as soon as it is written there. Afterwards you have to put your finger a second time for the verification and the imprint is stored in the next number.**

- **Let us also see whether our finger is recognized. So remove your finger from the sensor and call the following script**
  - **python /usr/share/doc/python-fingerprint/examples/example_search.py**

- **Put your finger on it again. If the fingerprint on the Raspberry Pi is detected, a message like this appears:**

```
Currently stored templates: 2
Waiting for finger...
Found template at position #1
The accuracy score is: 63
SHA-2 hash of template: 3aa1b01149abf0a7ad0d7803eaba65c22ba084009700c3c7f5f4ec
```

CONCLUSION:-

Thus we have configured finger print module to Raspberry Pi.

PRACTICAL - 6

AIM:-

Fingerprint Sensor interfacing with Raspberry Pi.

ABSTRACT:-

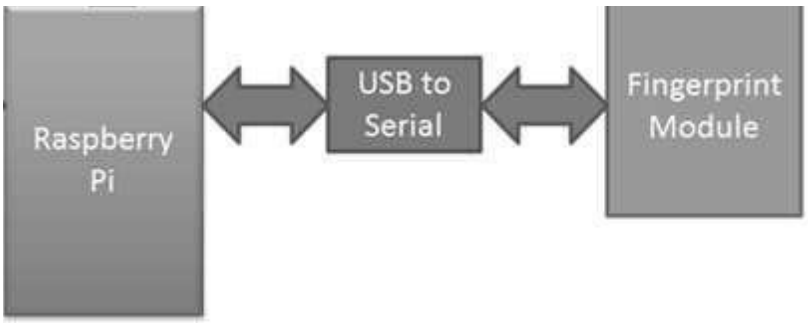
Finger Print Sensor, is now become very common to verify the identity of a person for various purposes. In present time we can see fingerprint-based systems everywhere in our daily life like for attendance in offices, employee verification in banks, for cash withdrawal or deposits in ATMs, for identity verification in government offices etc. Using this Raspberry Pi Fingerprint System, we can enrol new finger prints in the system and can delete the already fed finger prints.

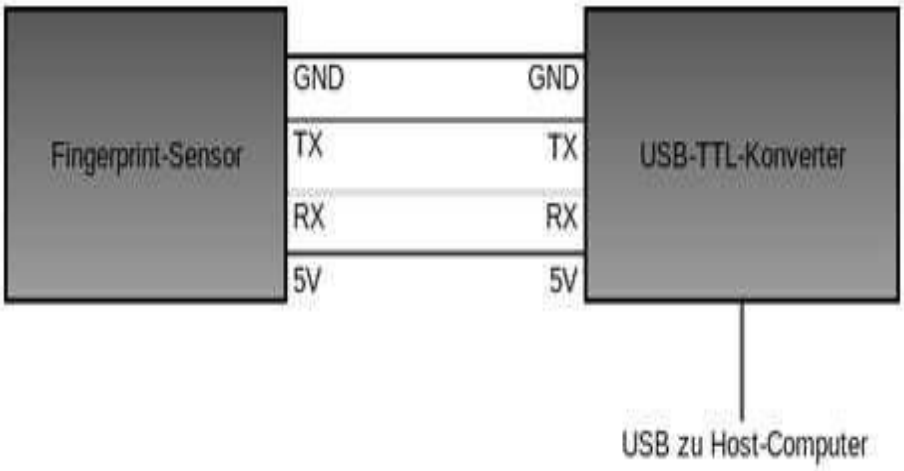
REQUIRED COMPONENTS:-

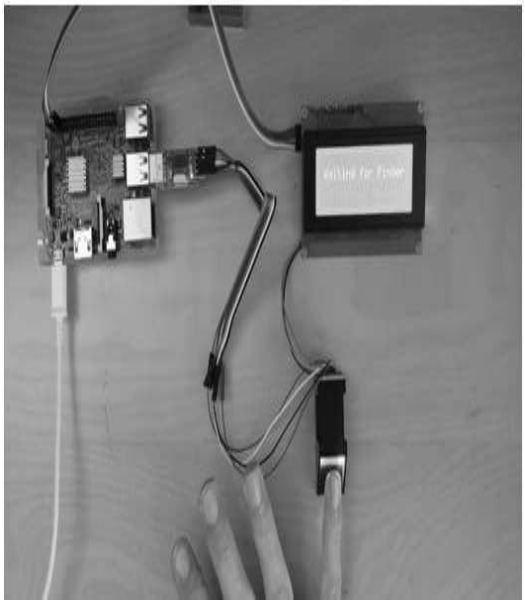
Particular	Quantity
Raspberry Pi3	1
Power Supply	1
USB Keyboard	1
USB Mouse	1
Micro SD card	1
Discover Board	1
USB to TTL converter	1
Fingerprint Module	1

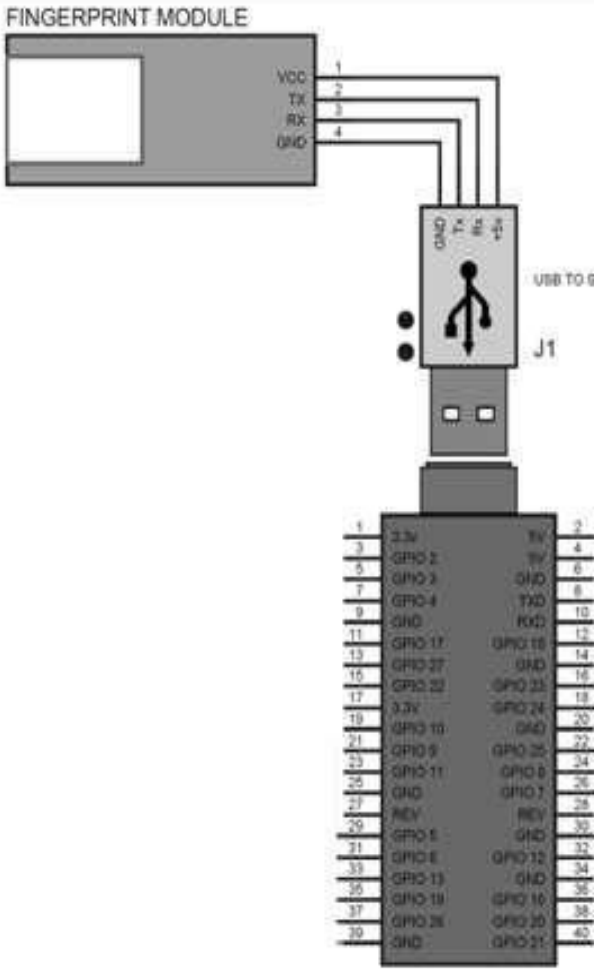
The Raspberry Pi has two pins (pin 8 / GPIO14 and pin 10 / GPIO 15), but they work with 3.3V. Since there are different fingerprint sensors, which do not all work with 3.3V, a USB UART converter is recommended.

INTERFACING DIAGRAM:-



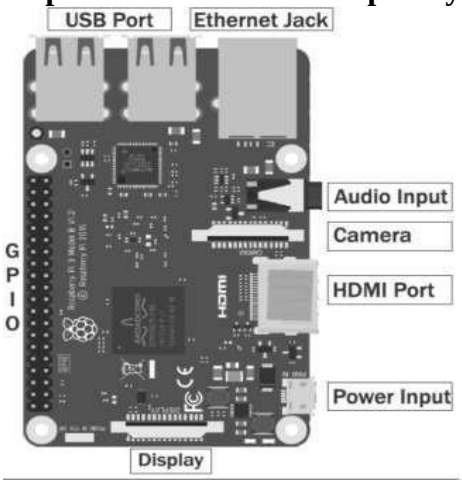

Fingerprint-Sensor
GND
GND
TX
TX
RX
RX
5V
5V
USB-TTL-Konverter
USB zu Host-Computer




FINGERPRINT MODULE
VCC
TX
RX
GND
J1

PROCEDURE:-

Step 1:- Connection of Raspberry Pi.



a. Connect Keyboard and mouse to USB Ports of Raspberry Pi.

b. Connect HDMI Port of Raspberry Pi to HDMI monitor via HDMI cable or HDMI to VGA convertor.

Step 2:- Switch ON power supply.

Step 3:- Login to Raspberry Pi

a. Username: pi
b. Password: raspberry

Step 4:- Step to check whether finger print sensor is installed properly.

To check whether the cabling is correct and whether the sensor is detected, you can open your console and perform the following before and after connecting:

ls /dev/ttyUSB*

If no other serial devices are connected via USB, nothing should be displayed first and afterwards /dev/ttyUSB0.

Step 4:- Installation steps for Raspberry Pi finger print module

- **Installation of the Raspberry Pi Fingerprint Library**

For some commands of the installation, root privileges are required. Therefore we start a terminal session and type the following, which executes all the following commands as root:

sudo bash

- **Add the necessary package**

  wget -O - http://apt.pm-codeworks.de/pm-codeworks.de.gpg | apt-key add -

  wget http://apt.pm-codeworks.de/pm-codeworks.list -P /etc/apt/sources.list.d/

- **Need to update the Raspberry pi and install the downloaded finger print sensor library:**

  sudo apt-get update

  sudo apt-get install python-fingerprint

- **If an error has occurred (in particular, that not all dependent packages have been installed), then execute the following:**

  sudo apt-get -f install

- **To return to the normal shell (under the Pi user), type exit.**

- **Attached are sample files for storing a new fingerprint, reading out and deleting stored fingerprints. Let's begin by recording a finger. Call the following:**
  - **python /usr/share/doc/python-fingerprint/examples/example_enroll.py**

- **Put your finger on the glass surface, wait for the instruction in the terminal and remove your finger as soon as it is written there. Afterwards you have to put your finger a second time for the verification and the imprint is stored in the next number.**

- **Let us also see whether our finger is recognized. So remove your finger from the sensor and call the following script**
  - **python /usr/share/doc/python-fingerprint/examples/example_search.py**

- **Put your finger on it again. If the fingerprint on the Raspberry Pi is detected, a message like this appears:**

```
Currently stored templates: 2
Waiting for finger...
Found template at position #1
The accuracy score is: 63
SHA-2 hash of template: 3aa1b01149abf0a7ad0d7803eaba65c22ba084009700c3c7f5f4ec
```

CONCLUSION:-

Thus we have configured finger print module to Raspberry Pi.

PRACTICAL - 7

AIM:-

IOT based Web Controlled Home Automation using Raspberry Pi.

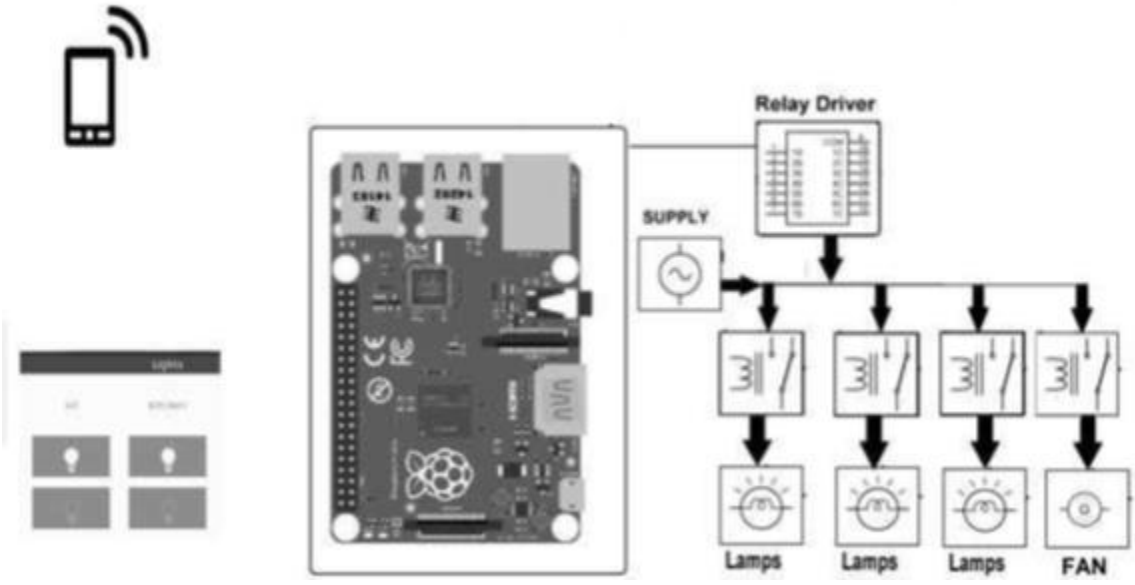
ABSTRACT:-

Using this IOT based home automation system, you can control your Home appliances from anywhere in the world. This web server can be run from any device which can run HTML applications, like Smart Phone, tablet, computer etc.

REQUIRED COMPONENTS:-

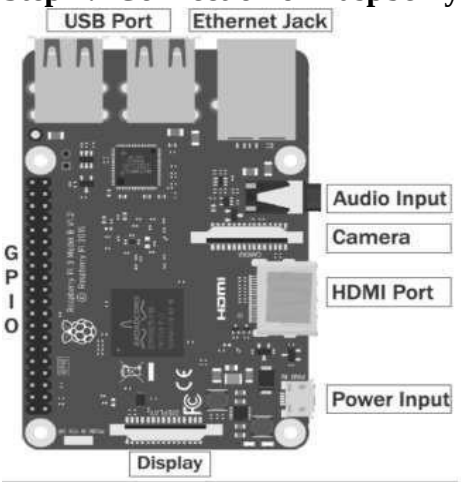
Particular	Quantity
Raspberry Pi3	1
Power Supply	1
USB Keyboard	1
USB Mouse	1
Micro SD card	1
Telegram Messenger	1
Relay board	1

INTERFACING DIAGRAM:-



PROCEDURE:-

Step 1:- Connection of Raspberry Pi.



a. Connect Keyboard and mouse to USB Ports of Raspberry Pi.

b. Connect HDMI Port of Raspberry Pi to HDMI monitor via HDMI cable or HDMI to VGA convertor.

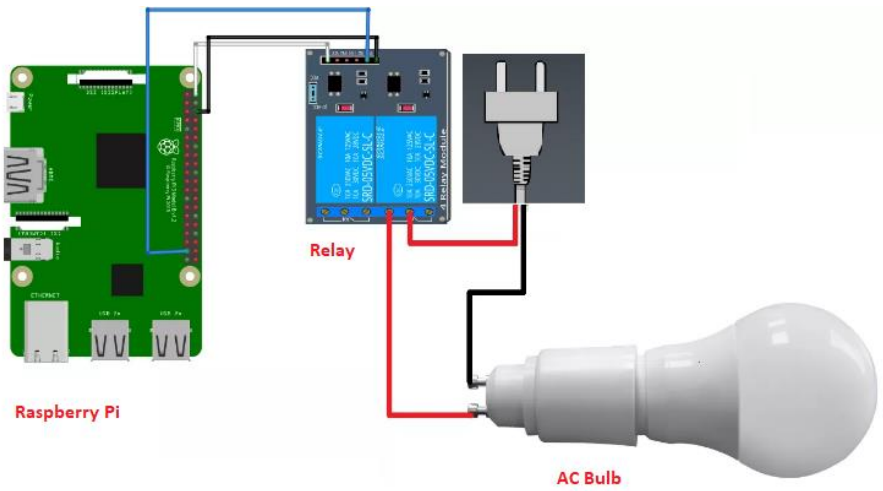
Step 2:- Switch ON power supply.

Step 3:- Login to Raspberry Pi

a. Username: pi
b. Password: raspberry

Step 4:- Connection of relay

a. Connect Relay Driver to AC Lamp and Fan as shown below.



Step 5:- Start Telegram app
Finally, you check all the hardware connection and run the program and enjoy controlling your devices like fan and light.

PROGRAM:-

```
import datetime

import teleport

import time

from telepot.loop import MessageLoop

import RPi.GPIO as IO

from time import sleep

IO.setmode(IO.BCM)

relay = 21

relay2= 20

IO.setup(relay,IO.OUT)

IO.setup(relay2,IO.OUT)

now=datetime.datetime.now()


def handle(msg):
```

```
    chat_id = msg['chat']['id']
    command=msg['text']

    print('Received')
    print(command)

    if command == '/hi':
        telegram_bot.sendMessage (chat_id, str("Hi!"))
    elif command == '/time':
        telegram_bot.sendMessage(chat_id, str(now.hour)+str(":")+str(now.minute))
    elif command == '/relayon':
        telegram_bot.sendMessage (chat_id, str("Relay1 On"))
        IO.output(relay,True)
    elif command == '/relayoff':
        telegram_bot.sendMessage (chat_id, str("Relay1 Off"))
        IO.output(relay,False)
telegram_bot = telepot.Bot('Token ID')
print (telegram_bot.getMe())
MessageLoop(telegram_bot, action).run_as_thread()
print ('Up and Running....')
while 1:
    time.sleep(10)
```

CONCLUSION:-

IOT based Web Controlled Home Automation is made using Raspberry Pi.

PRACTICAL - 8

AIM:-

Raspberry Pi module interfacing to camera.

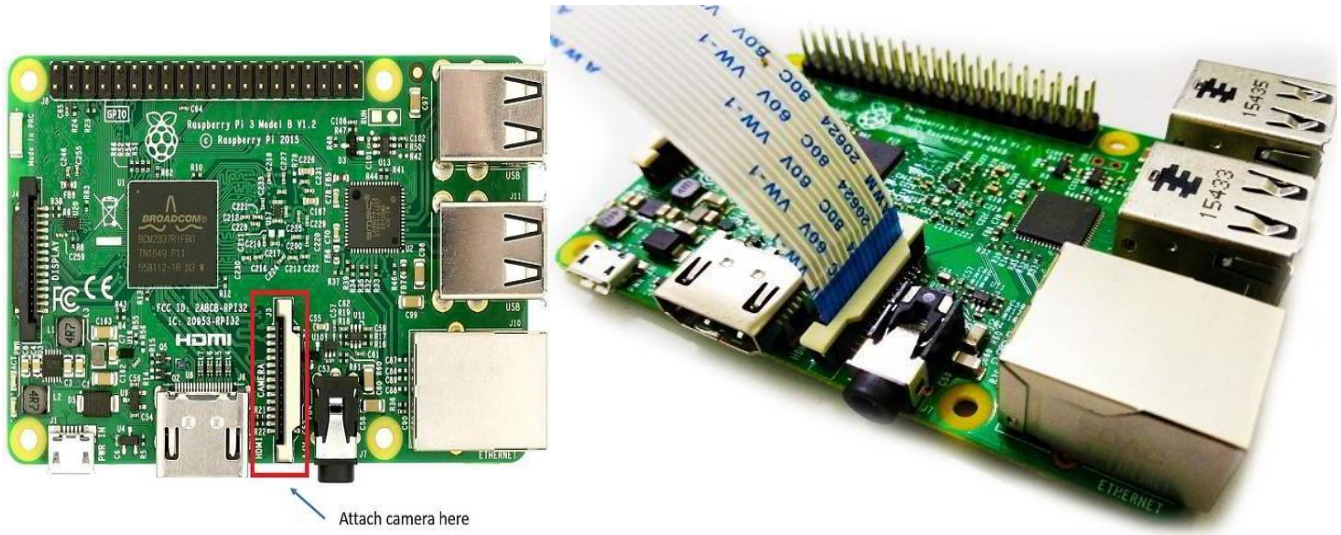
ABSTRACT:-

Raspberry Pi module interfacing to camera. Raspberry Pi Board has CSI (Camera Serial Interface) interface to which we can attach Pi Camera module directly. This Pi Camera module can attach to the Raspberry Pi's CSI port using 15-pin ribbon cable.

REQUIRED COMPONENTS:-

Particular	Quantity
Raspberry Pi3	1
Power Supply	1
USB Keyboard	1
USB Mouse	1
Micro SD card	1
Pi camera module	1

INTERFACING DIAGRAM:-



PROCEDURE:-

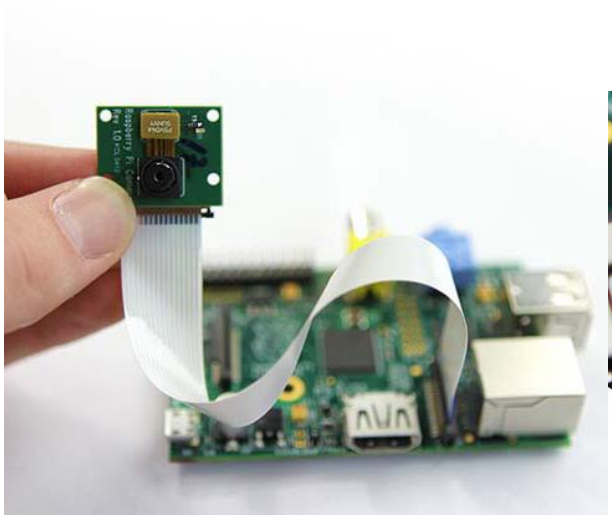
Step 1:- Setting up camera hardware

a. The camera board attaches to the Raspberry Pi via a 15-way ribbon cable.

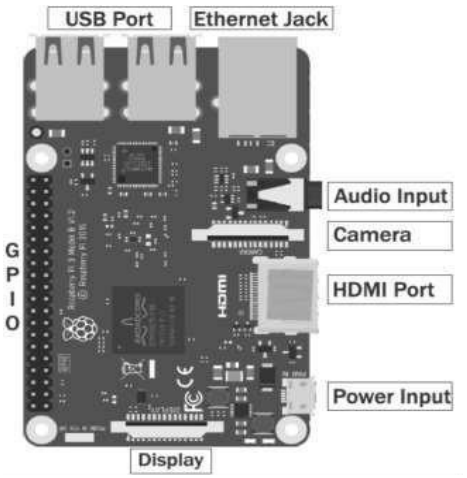
b. There are only two connections to make: the ribbon cable need to be attached to the camera PCB and the Raspberry Pi itself.

c. You need to get it the right way round, or the camera will not work.

d. On the camera PCB, the blue backing on the cable should be facing away from the PCB, and on the Raspberry Pi it should be facing towards the Ethernet connection (or where the Ethernet connector would be if you are using a model A).



Step 2:- Connection of Raspberry Pi.



a. Connect Keyboard and mouse to USB Ports of Raspberry Pi.

b. Connect HDMI Port of Raspberry Pi to HDMI monitor via HDMI cable or HDMI to VGA convertor.

Step 3:- Switch ON power supply.

Step 4:- Login to Raspberry Pi

a. Username: pi
b. Password: raspberry

Step 5:- Enable Camera Functionality.

For enabling camera in Raspberry Pi, open raspberry pi configuration using following command,

sudo raspi-config

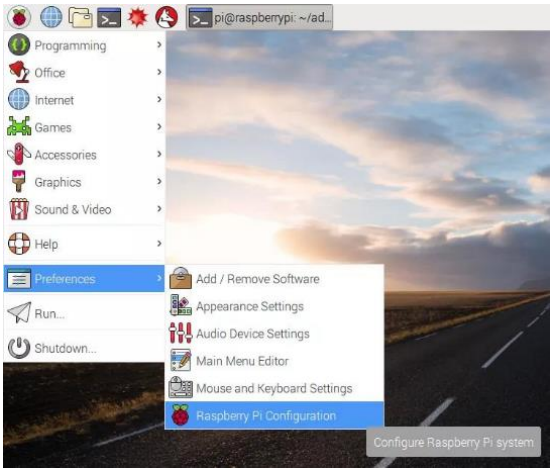
then select Interfacing options in which select camera option to enable its functionality.

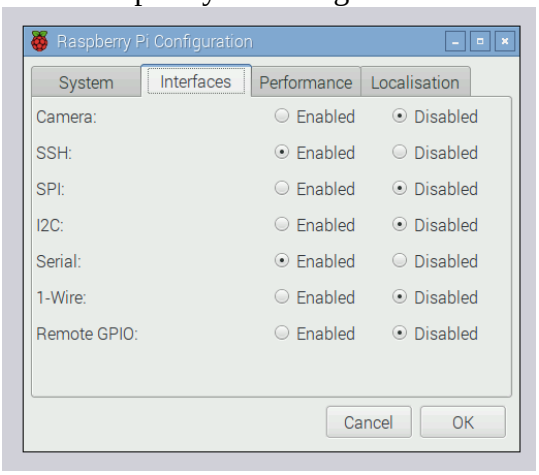
reboot Raspberry Pi.

Now we can access camera on Raspberry Pi.

Now we can capture images and videos using Pi Camera on Raspberry Pi

Ensure the camera software is enabled in the Raspberry Pi Configuration tool:





Step 6:- Setting up the Camera software

sudo apt-get update
sudo apt-get upgrade

Step 7:- Capturing image using camera

To capture image we have to execute the following command:

raspistill -v -o test.jpg
The display should show a 5-second preview from the camera and then take a picture, saved to the file test.jpg

```
raspistill -t 2000 -o image.jpg
raspistill -t 2000 -o image.jpg -w 640 -h 480
raspistill -t 2000 -o image.jpg -q 5
raspistill -t 2000 -o image.jpg –p 100,100,300,200
raspistill -t 2000 -o image.png –e png
```

Image parameter commands

-?, --help : This help information

-w, --width : Set image width

-h, --height : Set image height

-q, --quality : Set jpeg quality <0 to 100>

-r, --raw : Add raw bayer data to jpeg metadata

-o, --output : Output filename (to write to stdout, use '-o -'). If not specified, no file is saved

-v, --verbose : Output verbose information during run

-t, --timeout : Time (in ms) before takes picture and shuts down (if not specified, set to 5s)

-th, --thumb : Set thumbnail parameters (x:y:quality)

-d, --demo : Run a demo mode (cycle through range of camera options, no capture)

-e, --encoding : Encoding to use for output file (jpg, bmp, gif, png)

-x, --exif : EXIF tag to apply to captures (format as 'key=value')

-tl, --timelapse : Timelapse mode. Takes a picture every ms

-sh, --sharpness : Set image sharpness (-100 to 100)

-co, --contrast : Set image contrast (-100 to 100)

-br, --brightness : Set image brightness (0 to 100)

-sa, --saturation : Set image saturation (-100 to 100)

-ISO, --ISO : Set capture ISO

-vs, --vstab : Turn on video stablisation

-ev, --ev : Set EV compensation

-ex, --exposure : Set exposure mode (see Notes)

-awb, --awb : Set AWB mode (see Notes)

-ifx, --imxfx : Set image effect (see Notes)

-cfx, --colfx : Set colour effect (U:V)

-mm, --metering : Set metering mode (see Notes)

-rot, --rotation : Set image rotation (0-359)

-hf, --hflip : Set horizontal flip

-vf, --vflip : Set vertical flip

Step 8:- Capturing video using camera

To capture video we have to execute the following commands:
raspivid -t 5000 -o video.h264
raspivid -t 5000 -o video.h264 -b 3500000
raspivid -t 5000 -o video.h264 -f 5

Display camera output to display, and optionally saves an H264 capture at requested bitrate

Video parameter commands

-?, --help : This help information

-w, --width : Set image width . Default 1920

-h, --height : Set image height . Default 1080

-b, --bitrate : Set bitrate. Use bits per second (e.g. 10MBits/s would be -b 10000000)

-o, --output : Output filename (to write to stdout, use '-o -')

-v, --verbose : Output verbose information during run

-t, --timeout : Time (in ms) before takes picture and shuts down. If not specified, set to 5s

-d, --demo : Run a demo mode (cycle through range of camera options, no capture)

-fps, --framerate : Specify the frames per second to record

-e, --penc : Display preview image *after* encoding (shows compression artifacts)

Preview parameter commands

-p, --preview : Preview window settings

-f, --fullscreen : Fullscreen preview mode

-n, --nopreview : Do not display a preview window

Image parameter commands

-sh, --sharpness : Set image sharpness (-100 to 100)

-co, --contrast : Set image contrast (-100 to 100)

-br, --brightness : Set image brightness (0 to 100)

-sa, --saturation : Set image saturation (-100 to 100)

-ISO, --ISO : Set capture ISO

-vs, --vstab : Turn on video stablisation

-ev, --ev : Set EV compensation

-ex, --exposure : Set exposure mode (see Notes)

-awb, --awb : Set AWB mode (see Notes)

-ifx, --imxfx : Set image effect (see Notes)

-cfx, --colfx : Set colour effect (U:V)

-mm, --metering : Set metering mode (see Notes)

-rot, --rotation : Set image rotation (0-359)

-hf, --hflip : Set horizontal flip

-vf, --vflip : Set vertical flip

CONCLUSION:-

We have interfaced camera module to RaspberryPi.

PRACTICAL - 9

AIM:-

RFID interfacing with Raspberry Pi.

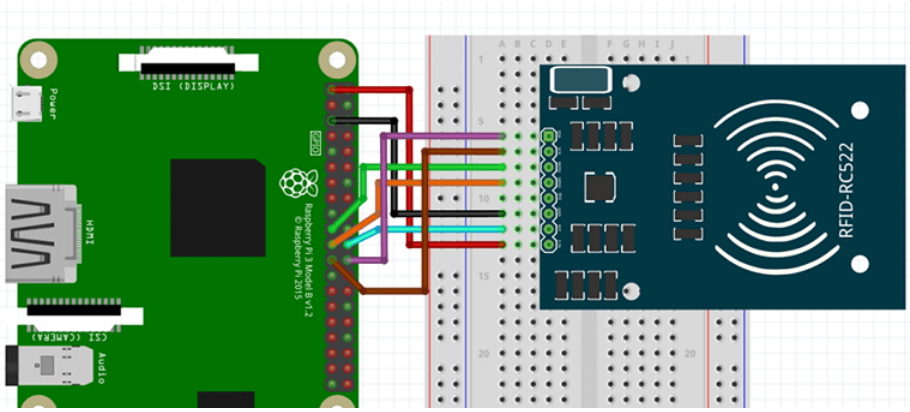
ABSTRACT:-

Radio-frequency identification (RFID) uses electromagnetic fields to automatically identify and track tags attached to objects. The tags contain electronically stored information. The RFID technology enables remote and automated gathering and sending of information between RFID tags or transponders and readers using a wireless link. Using RFID the exchange of data between tags and readers is rapid, automatic and does not require direct contact or line of sight. An RFID system consists of a reader device and a transponder. A transponder or tag has a unique serial number which is identified by the reader. The RFID module can be interfaced with the board using the same serial reception program used for testing the UART. When a Card (tag) is brought near the RFID reader, it tries to communicate with the tag, receives the data and decodes it. Finally it send the data over the TX line. The UART module in MCU receives the data and thus used for further applications. RFID tag is applied to products, individuals or animals to identify and track them through this number. By employing RFID, much secured entry systems can be developed without incurring huge costs.

REQUIRED COMPONENTS:-

Particular	Quantity
Raspberry Pi3	1
Power Supply	1
USB Keyboard	1
USB Mouse	1
Micro SD card	1
RFID Reader	1

INTERFACING DIAGRAM:-



PROCEDURE:-

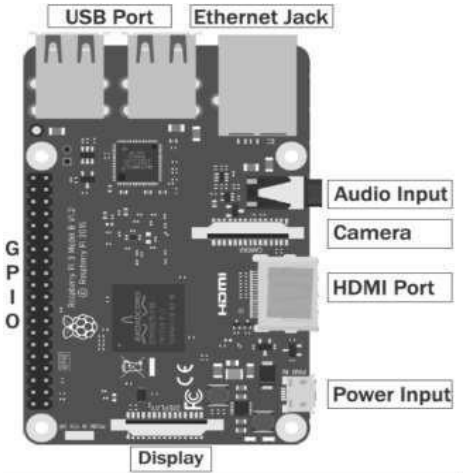
Step 1:- Setting up RFID module with Raspberry Pi

RFID RC522 there are 8 possible connections on it.

Connections of Raspberry Pi to RFID

- SDA (Serial Data Signal) - 24
- SCK (Serial Clock) -23
- MOSI (Master Out Slave In)-Line for the master to send data to the slave -19
- MISO (Master In Slave Out)-Line for the slave to send data to the master -21
- IRQ (Interrupt Request) –Not used
- GND (Ground Power)-6
- RST (Reset-Circuit) -22
- 3.3v (3.3v Power In)-1

Step 2:- Connection of Raspberry Pi.



a. Connect Keyboard and mouse to USB Ports of Raspberry Pi.

b. Connect HDMI Port of Raspberry Pi to HDMI monitor via HDMI cable or HDMI to VGA convertor.

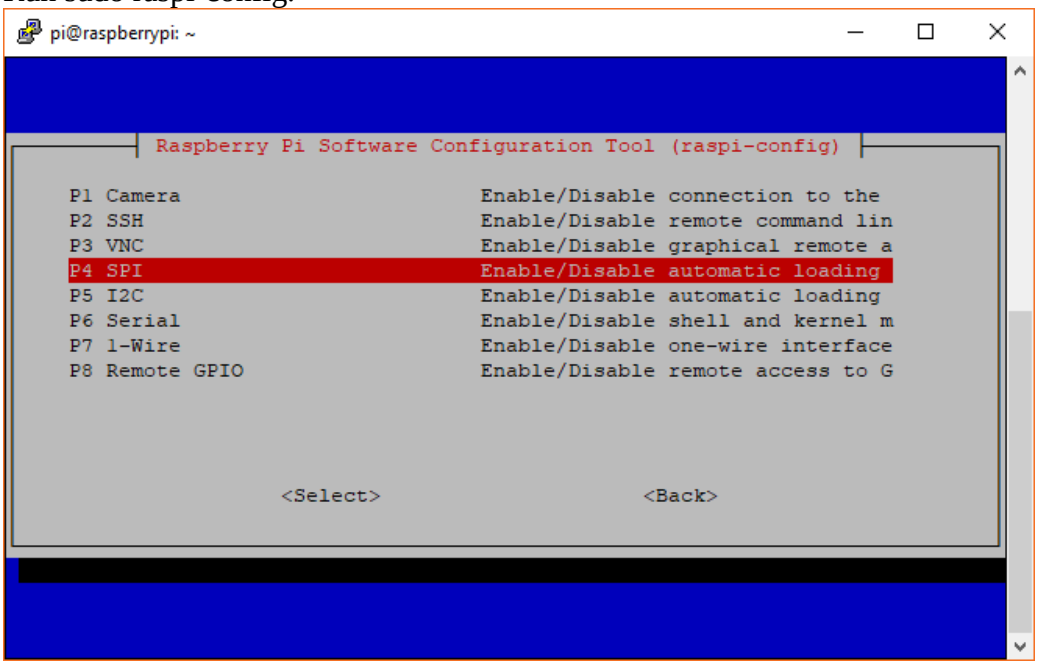
Step 3:- Switch ON power supply.

Step 4:- Login to Raspberry Pi

a. Username: pi
b. Password: raspberry

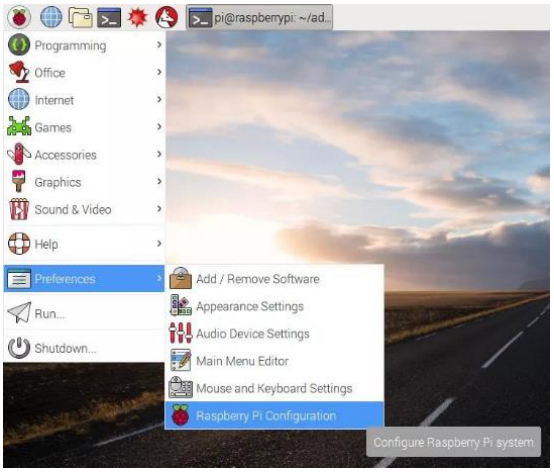
Step 5:-Enable SPI on Raspberry PI

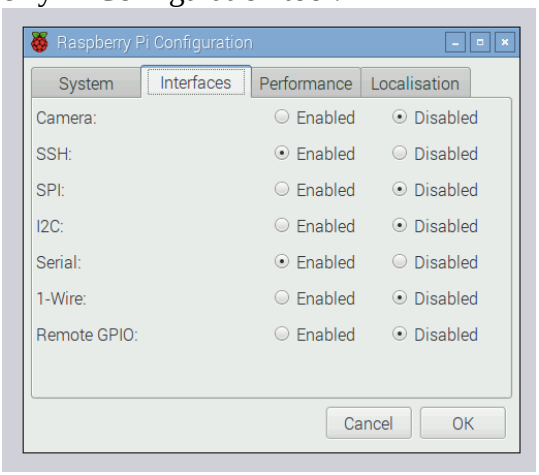
- Run sudo raspi-config.



- Use the down arrow to select 5 Interfacing Options
- Arrow down to P4 SPI.
- Select yes when it asks you to enable SPI,
- Also select yes if it asks about automatically loading the kernel module.
- Use the right arrow to select the <Finish> button.
  Select yes when it asks to reboot

  Ensure the SPI is enabled in the Raspberry Pi Configuration tool:





Step 6:- Getting Python ready for the RFID RC522

sudo apt-get update
sudo apt-get upgrade

- Install python3-dev, python-pip and git packages
  sudo apt-get install python3-dev python3-pip

- Install the Python Library spidev to our Raspberry Pi using the python "pip" to the spidev library helps handle interactions with the SPI
  sudo pip3 install spidev

- we have installed the spidev library to our Raspberry Pi we can now now proceed to installing the MFRC522 library using pip
  There are two files that are included within our MFRC522 library that we make use of:
  - MFRC522.py which is an implementation of the RFID RC522 interface, thislibrary handles all the heavy lifting for talking with the RFID over the Pi's SPI Interface.
  - SimpleMFRC522.py that takes the MFRC522.py file and greatly simplifies it by making you only have to deal with a couple of functions instead of several.

    sudo pip3 install mfrc522

Step 7:- Writing with the RFID RC522

Program:-

```
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()
```

Step 8:-Reading the RFID tag

Program:-

```
import RPi.GPIO as GPIO
from mfrc522 import SimpleMFRC522

reader = SimpleMFRC522()

try:
      id, text = reader.read()
      print(id)
      print(text)
finally:
```

```
GPIO.cleanup()
```

CONCLUSION:-

We have interfaced RFID module to Raspberry Pi.

PRACTICAL - 10

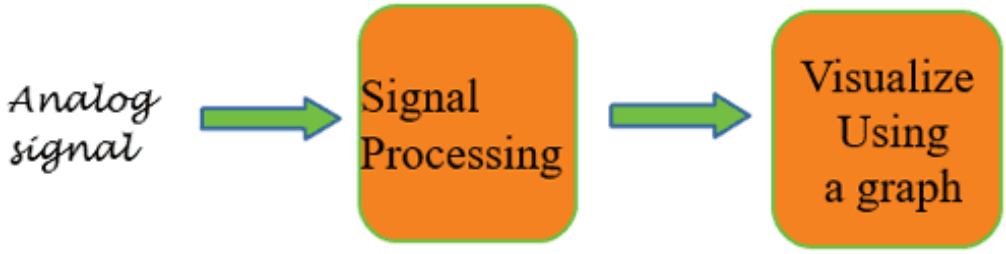
AIM:-

Raspberry Pi Based Oscilloscope.

ABSTRACT:-

The oscilloscope is an electronic test instrument that allows the visualization and observation of varying signal voltages, usually as a two dimensional plot with one or more signals plotted against time. Today's project will seek to replicate the signal visualization capabilities of the oscilloscope using the Raspberry Pi and an analog to digital converter module.

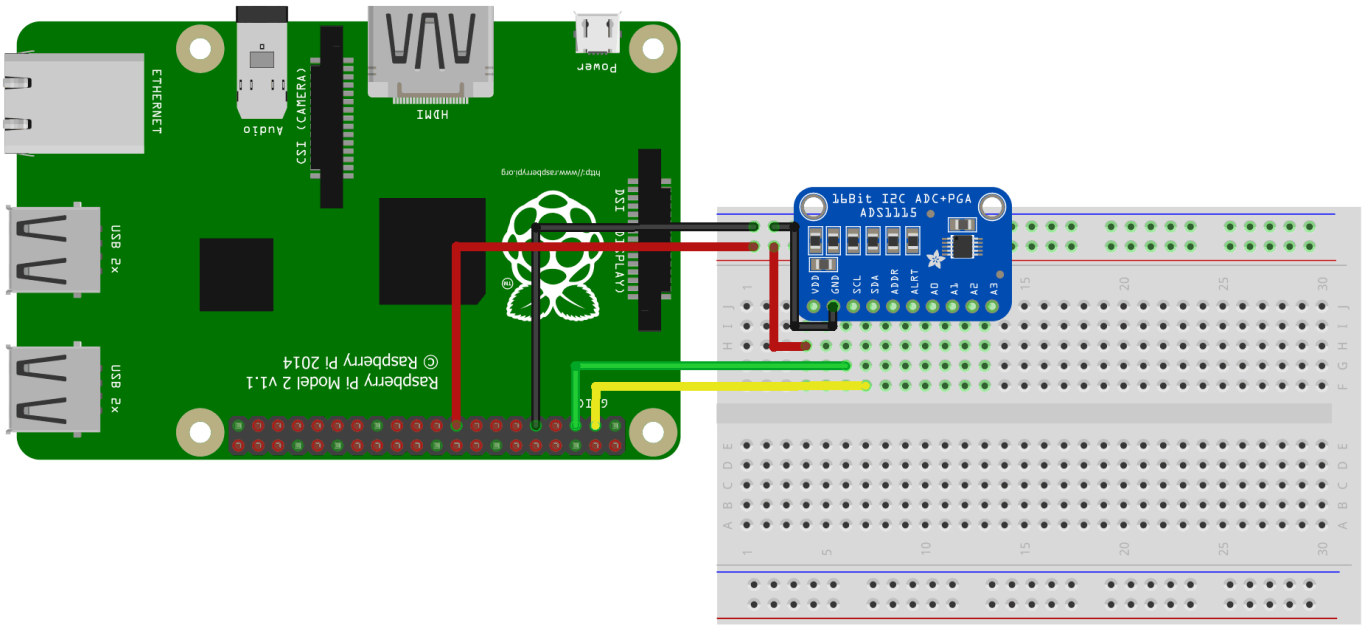
Replicating the signal visualization of the oscilloscope using the Raspberry Pi will require the following steps:
1. Perform Digital to analog conversion of the Input signal
2. Prepare the resulting data for representation
3. Plot the data on a live time graph



REQUIRED COMPONENTS:-

Particular	Quantity
Raspberry Pi3	1
Power Supply	1
USB Keyboard	1
USB Mouse	1
Micro SD card	1
ADS1115ADC	1
Analog Input As per availability	1

INTERFACING DIAGRAM:-



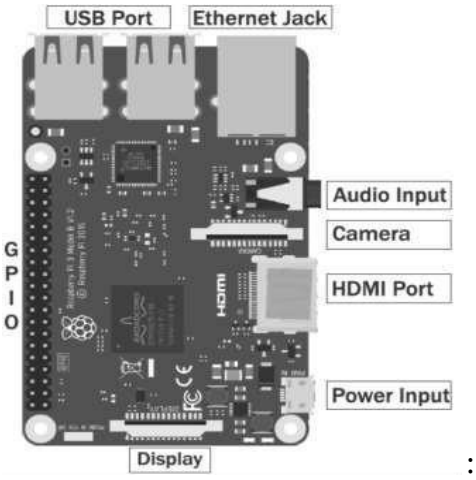
PROCEDURE:-

Step 1:- ADS1115 and Raspberry Pi Connections:

- VDD – 3.3v
- GND – GND
- SDA – SDA
- SCL – SCL

ADS1115 16-Bit ADC - 4 Channel with Programmable Gain Amplifier. The chip can be configured as 4 single-ended input channels, or two differential channels.

Step 2:- Connection of Raspberry Pi.



:

a. Connect Keyboard and mouse to USB Ports of Raspberry Pi.

b. Connect HDMI Port of Raspberry Pi to HDMI monitor via HDMI cable or HDMI to VGA convertor.

Step 3:- Switch ON power supply.

Step 4:- Login to Raspberry Pi

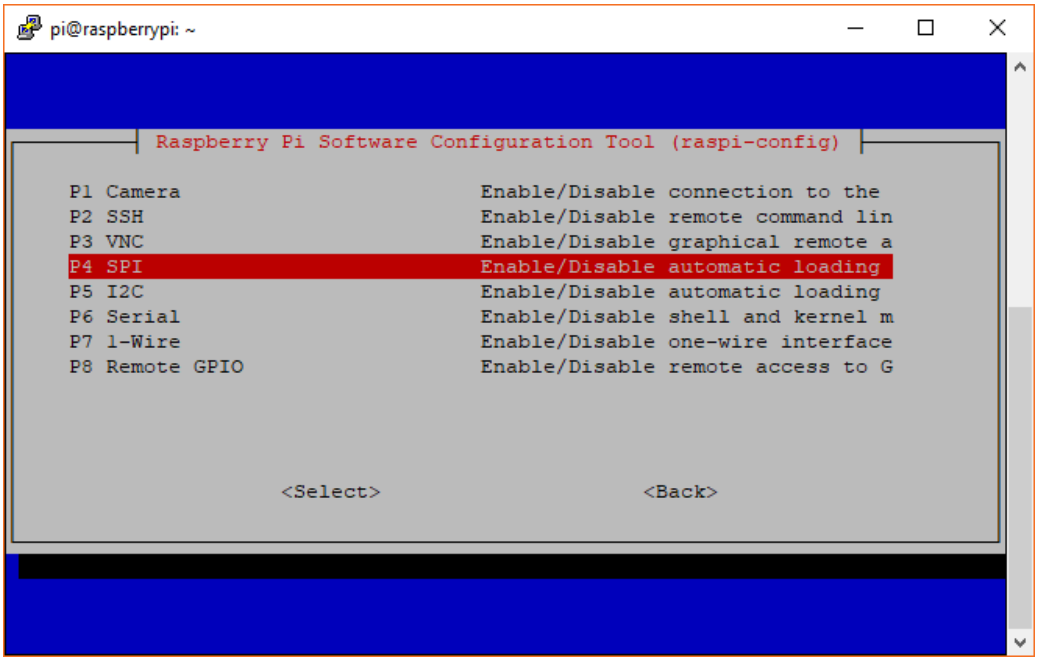
a. Username: pi

b. Password: raspberry

Step 5: Enable Raspberry Pi I2C interface

sudo raspi-config

- When the configuration panels open, select interface options, select I2C and click Enable



Step 6: Update the Raspberry pi

sudo apt-get update

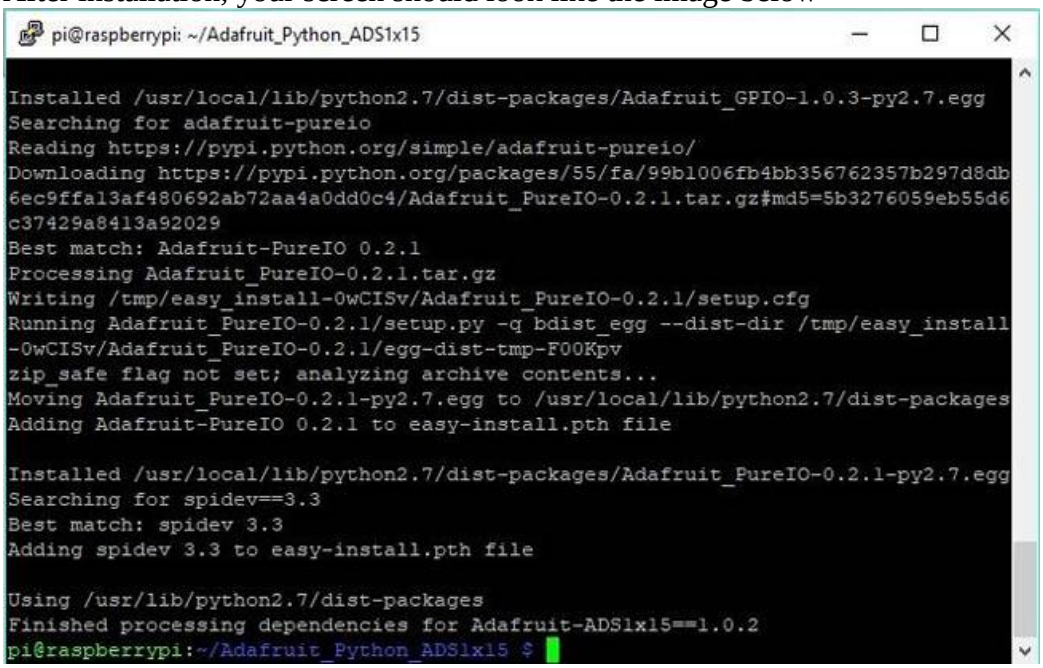
sudo apt-get upgrade

Step 7: Install the Adafruit ADS1115 library for ADC

- Ensure you are in the Raspberry Pi home directory by running;
  cd ~
- then install the build-essentials by running;
  sudo apt-get install build-essential python-dev python-smbus git
- Next, clone the Adafruit git folder for the library by running;

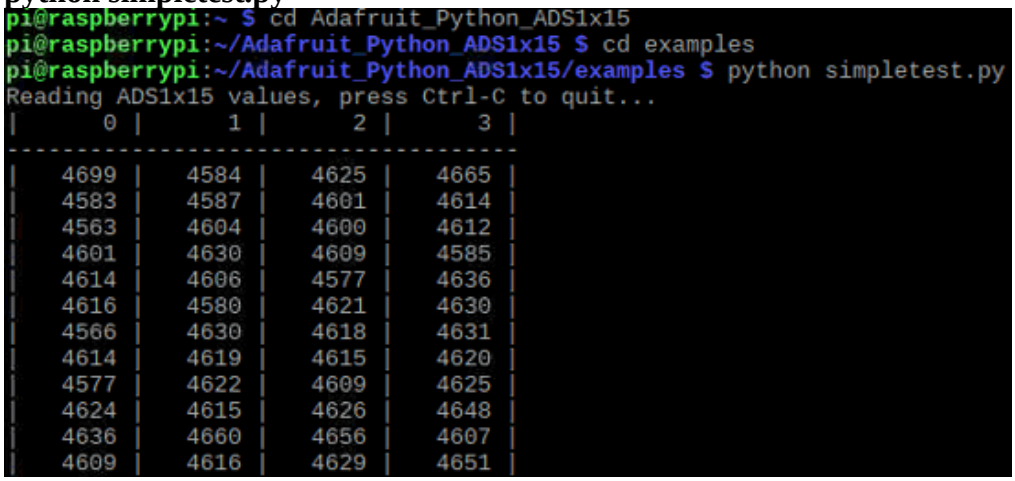
git clone https://github.com/adafruit/Adafruit_Python_ADS1x15.git

- Change into the cloned file's directory and run the setup file;
  cd Adafruit_Python_ADS1x1z
  sudo python setup.py install
  After installation, your screen should look like the image below



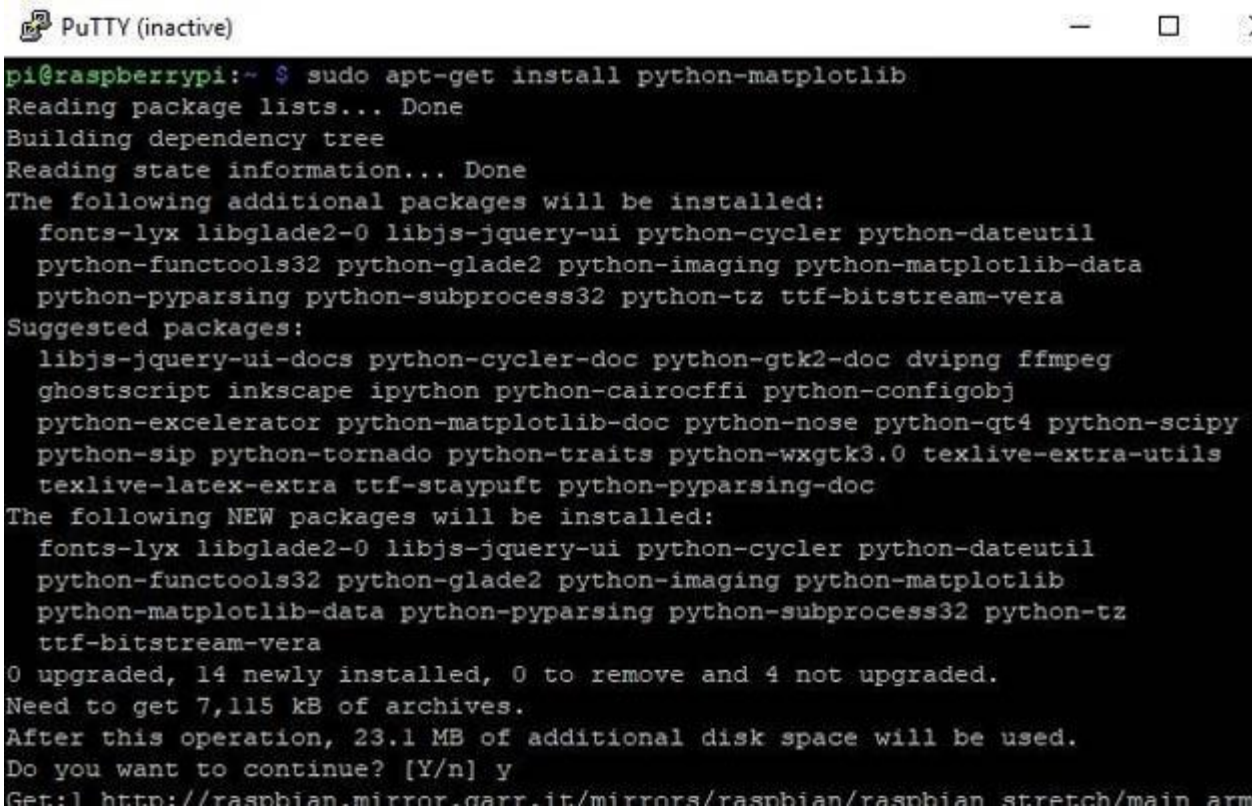
Step 8: Test the library and 12C communication.

- While still in the Adafruit_Python_ADS1x15 folder, change directory to the examples directory by running;
  cd examples
- Next, run the sampletest.py example which displays the value of the four channels on the ADC in a tabular form.
- Run the example using:
  python simpletest.py



Step 9: Install Matplotlib

- To visualize the data we need to install the matplotlib module which is used to plot all kind of graphs in python. This can be done by running;'
  sudo apt-get install python-matplotlib
  You should see an outcome like the image below



Step 10: Install the Drawnow python module

- Lastly, we need to install the drawnow python module. This module helps us provide live updates to the data plot.
- We will be installing drawnow via the python package installer; pip, so we need to ensure it is installed. This can be done by running;
  sudo apt-get install python-pip
- We can then use pip to install the drawnow package by running:
  sudo pip install drawnow
  You should get an outcome like the image below after running it.

```
pi@raspberrypi:~ $ sudo pip install drawnow
Collecting drawnow
  Downloading drawnow-0.71.3.tar.gz
Requirement already satisfied: matplotlib>=1.5 in /usr/lib/python2.7/dist-packag
es (from drawnow)
Building wheels for collected packages: drawnow
  Running setup.py bdist_wheel for drawnow ... done
  Stored in directory: /root/.cache/pip/wheels/83/90/79/cc7449a69f925bfbee33fc6f
582fa58febee3e2d0944ccb058
Successfully built drawnow
Installing collected packages: drawnow
Successfully installed drawnow-0.71.3
pi@raspberrypi:~ $
```

Step 11: Run python code for Raspberry Pi oscilloscope

Save the code and run using;

sudo python scope.py

PROGRAM:-

```
import time
import matplotlib.pyplot as plt
#import numpy
from drawnow import *
# Import the ADS1x15 module.
import Adafruit_ADS1x15
import warnings
import matplotlib.cbook
warnings.filterwarnings(“ignore”, category=matplotlib.cbook.mplDeprecation)
# Create an ADS1115 ADC (16-bit) instance.
adc = Adafruit_ADS1x15.ADS1115()

GAIN = 1
val = [ ]
cnt = 0
plt.ion()
# Start continuous ADC conversions on channel 0 using the previous gain value.
adc.start_adc(0, gain=GAIN)
print('Reading ADS1x15 channel 0')
#create the figure function
def makeFig():
   plt.ylim(-5000,5000)
   plt.title('Osciloscope')
   plt.grid(True)
   plt.ylabel('ADC outputs')
   plt.plot(val, 'ro-', label='Channel 0')
   plt.legend(loc='lower right')
```

```
while (True):
    # Read the last ADC conversion value and print it out.
    value = adc.get_last_result()
    print('Channel 0: {0}'.format(value))
    # Sleep for half a second.
    time.sleep(0.5)
    val.append(int(value))
    drawnow(makeFig)
    plt.pause(.000001)
    cnt = cnt+1
    if(cnt>50):
        val.pop(0)
```

CONCLUSION:-

Thus Raspberry Pi Based Oscilloscope has been studied and implemented.