Practical No. 01

Aim: Familiarizing with Raspberry Pi components and Interface, connecting to Ethernet, monitor and USB

Name: Ankit Singh Chauhan

Roll No: 64

Class: T.Y.BSc.IT

Sub: IOT

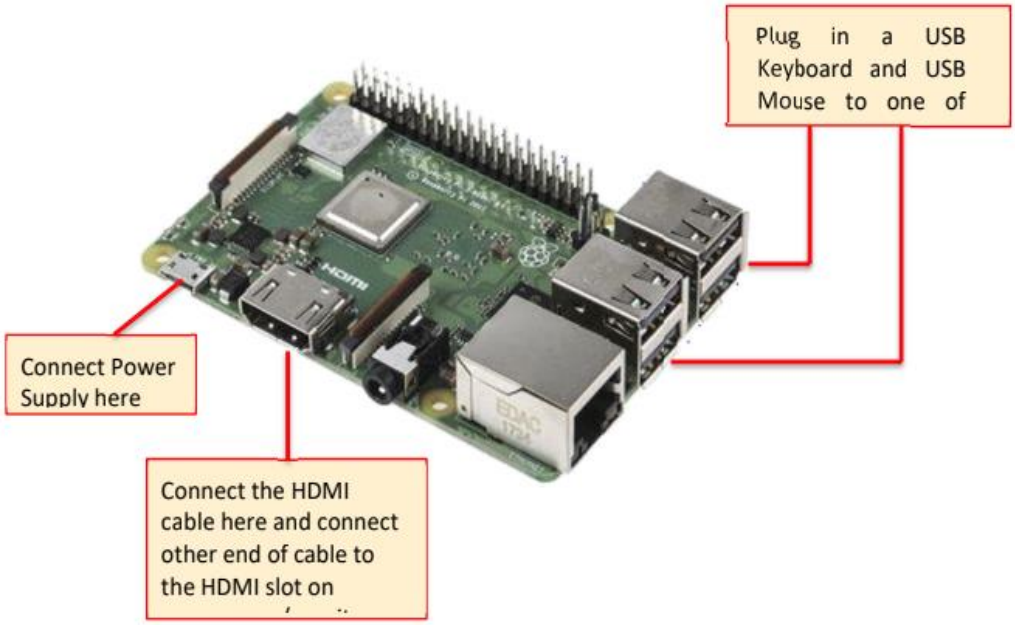
Grade:

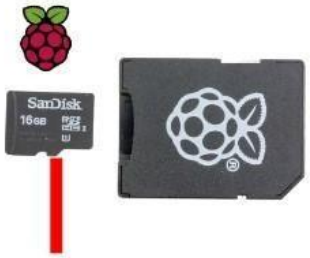
Sign:

Name: Ankit Singh Chauhan
Roll No: 64

Aim: Familiarizing with Raspberry Pi components and Interface, connecting to Ethernet, monitor and USB:

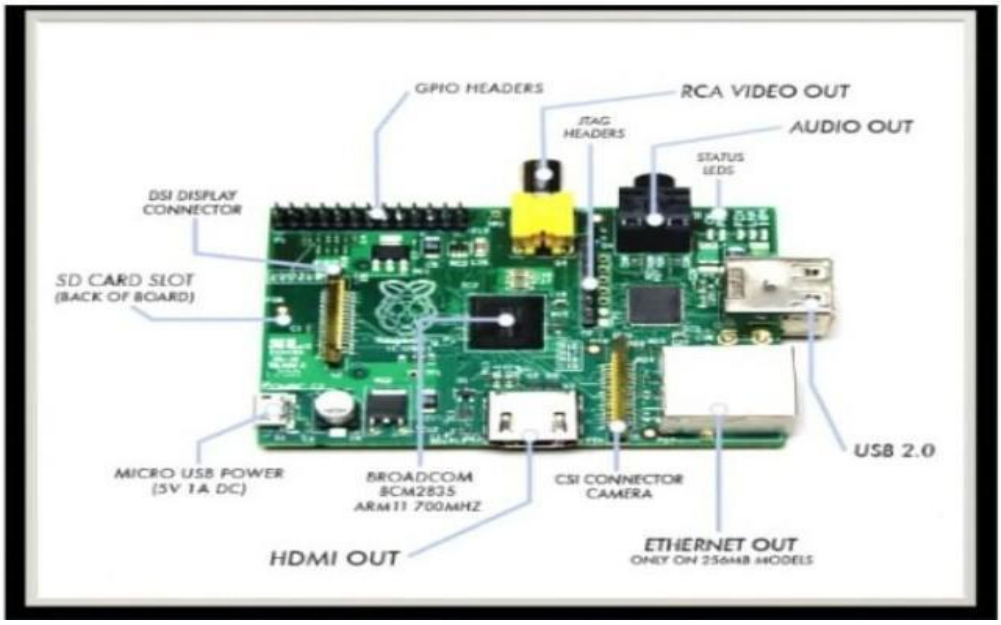
The quickest way to start working on Raspberry Pi is to set-up and begin experimenting with the various features. Connect the leads that came with your Raspberry Pi (HDMIcable and Power Supply) as shown plus a USB keyboard and USB Mouse, which you need to supply:
Finally, the micro SD card needs to be inserted. This card comes in an SD converter and needs to be removed to fit into the micro SD card socket on the underside of your Pi:





You are now ready to start the software set-up.

Name: Ankit Singh Chauhan
Roll No: 64



How to make it work

Monitor or TV:

A monitor or TV with HDMI in can be used as a display with a Raspberry Pi. Most modern television sets and monitors have an HDMI port, and are the easiest to get working with the.Raspberry Pi. You can use an HDMI cable to connect the Raspberry Pi directly to the television or monitor. Some older monitors have a DVI port. These work well with the Raspberry Pi, although you will need an HDMI- to-DVI adapter to attach to an HDMI cable, or a one-piece HDMI-to-DVI cable. Some old monitors have a VGA port. These

can be trickier to use as you will need an HDMI-to-VGA converter, which can change digital video to analogue video. A simple port adapter won't work.

HDMI to HDMI cable:

Connect Raspberry Pi to Monitor or TV with a HDMI to

HDMI cable. Ethernet Cable:

Ethernet cable will allow your Pi connect with the internet. It is also useful for headless setup of Raspberry Pi.

USB Keyboard and Mouse:

Any standard USB Keyboard and mouse can be used with the Raspberry Pi. This plug and play device will work without any additional driver. Simply plug them into the Raspberry Pi and they should be recognized when it starts up.

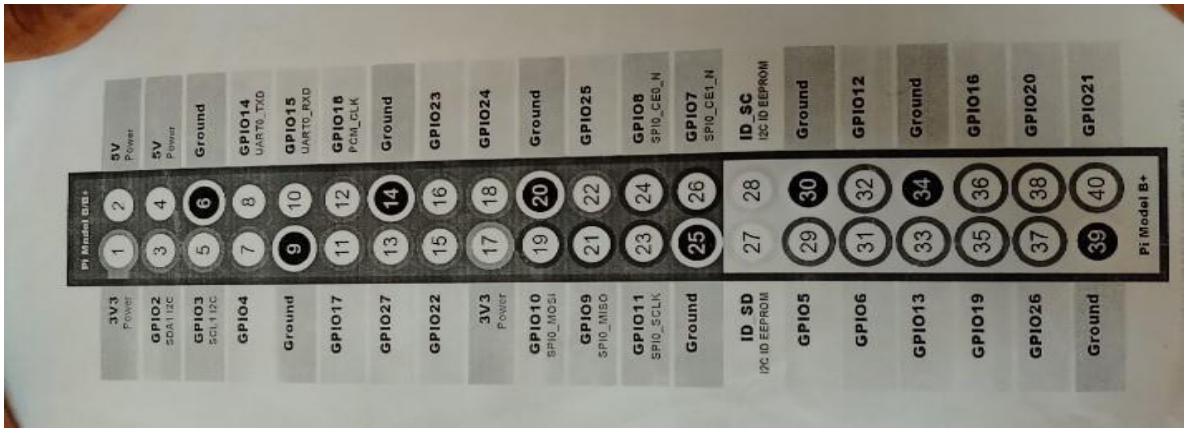
Power Supply:

It is recommended that you use a 5V, 2A USB power supply for all models of Raspberry Pi.

SD card:

The latest version of Raspbian, the default operating system recommended for the Raspberry Pi, requires an 8GB or larger micro SD card. SD card will store the operating systems as well as all the file and applications created by you.

40 GPIO PINS:



Practical No. 02

Aim: Starting Raspbian Operating System. Installation of Raspbian Operating System.

Name: Ankit Singh Chauhan

Roll No: 64

Class: T.Y.BSc.IT

Sub: IOT

Grade:

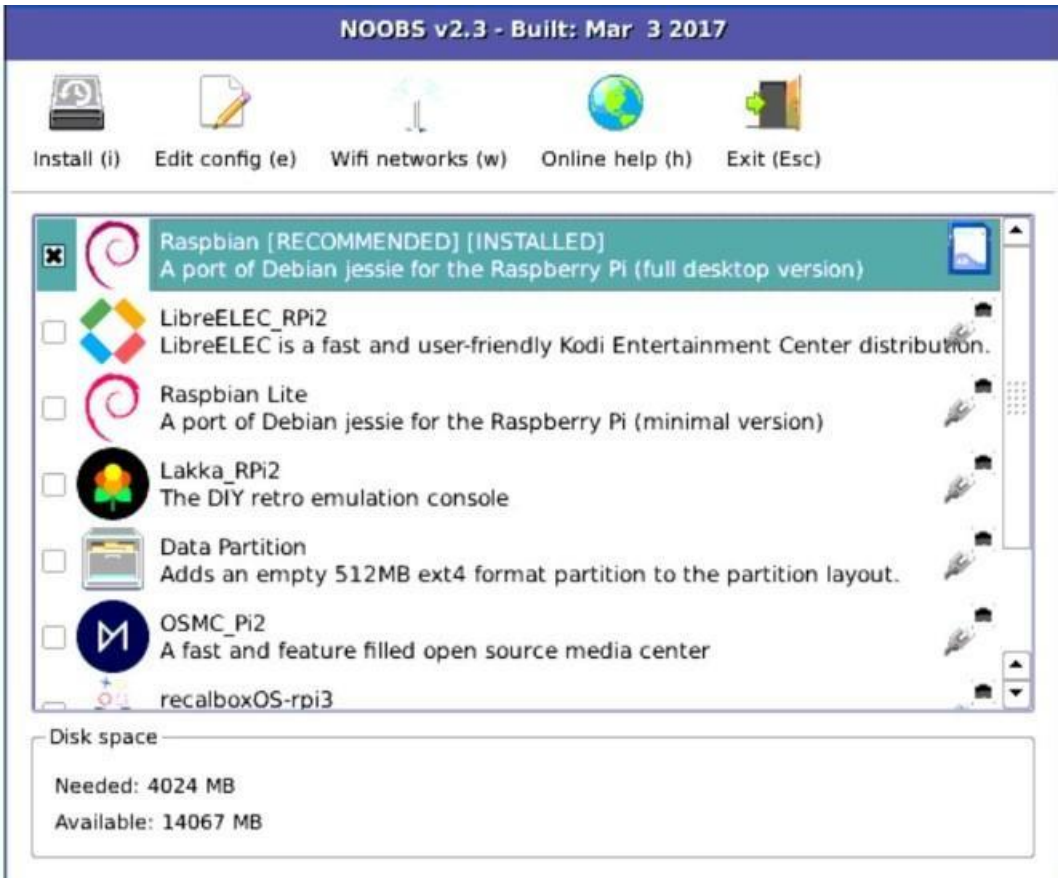
Sign:

Name: Ankit Singh Chauhan
Roll No: 64

Aim: Starting Raspbian Operating System. Installation of Raspbian Operating System:

As with any computer you need an operating system (OS) to allow you to interact with the machine. The micro-SD card you inserted into your Pi comes preinstalled with NOOBS, a program to guide you through the OS installation procedure.

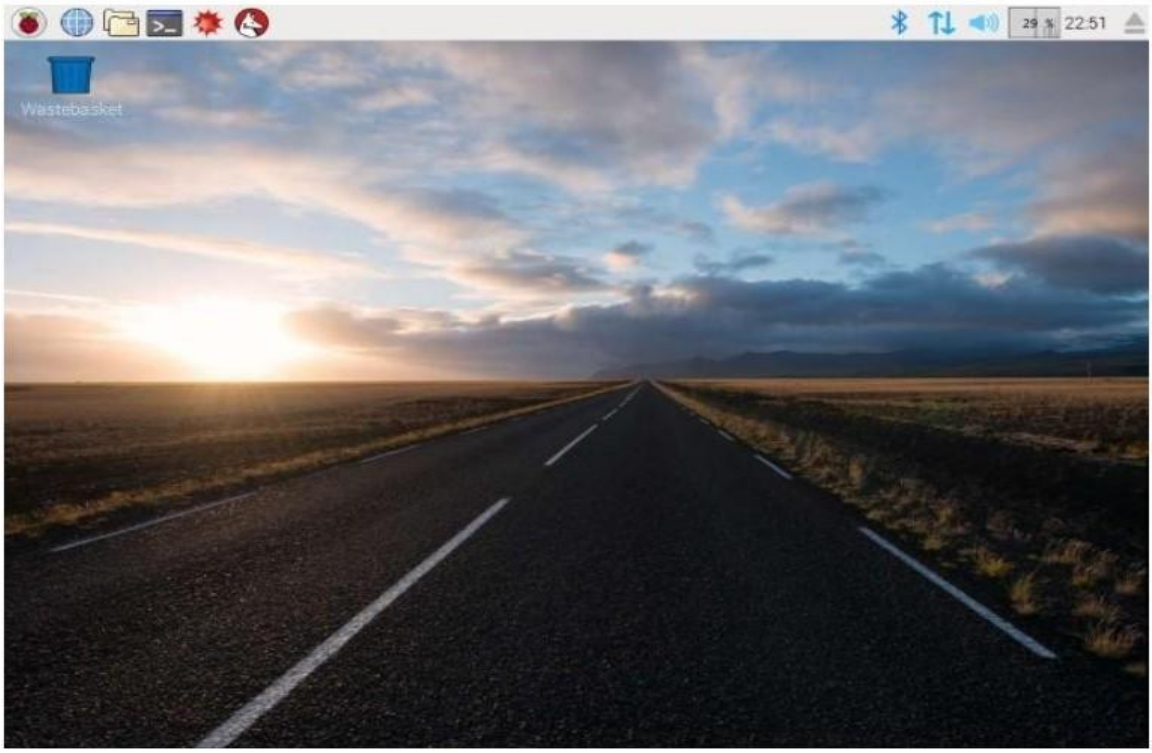
On powering up your Pi and switching to the HDMI source you are connected to on your screen or monitor you should see this screen:



Firstly, click on Wifi networks (w) and connect to your WiFi. Next check the box next to Raspbian (Recommended) and click the install button. The Raspbian Operating System will now be installed. This may take a little while.

Eventually you will be presented with the Home Screen:

Name: Ankit Singh Chauhan
Roll No: 64



It is now time for you to start investigating your machine and its software to become completely familiar with operating your Raspberry Pi. Your first point of access is the Raspberry button in the top left-hand corner of the screen, start investigating interacting with the various features and installed software on your Pi via this Graphical User Interface (GUI). When you are comfortable navigating and interacting with the various software you could try interacting with your machine at a lower level. At a very low level, computers work by interpreting ON and OFF signals as 0's and 1's; each digit is a bit of data. This would be extremely tedious and time consuming to read and write. It is for this reason an interface is used to convert this binary into something easier to read from and write to. The GUI you have used so far is a higher-level way to interact. At a lower level a shell can, be used interact with your operating system. As Raspbian (the OS) is based on an Operating System called Linux you interact with it through a Linux or Terminal Shell. To open a terminal shell, click on the 4th icon along from the left on the top bar of your screen. To see a full list of commands, you can use here, type help. Now experiment interacting with your Pi using a Terminal Shell. A list of commands can also be found at: https://www.raspberrypi.org/documentation/linux/usage/commands.md In the next task we will start writing simple programs to run on your Raspberry Pi.

Installation Guide:

Name: Ankit Singh Chauhan
Roll No: 64

How to install NOOBS on the Raspberry Pi

The Raspberry Pi is an incredible device, but it won't do much of anything without an operating system. Luckily, choosing and installing an appropriate operating system on your Raspberry Pi has never been easier. One simple method is to use NOOBS, or "New Out of Box Software." As the name suggests, NOOBS is perfect for Pi newbies. It lets you choose your preferred operating system and install it right then and there. But how do you load NOOBS itself? Here's our complete guide on how to install NOOBS on the Raspberry Pi.

We've called our article "How to install NOOBS on the Raspberry Pi," but what we're technically doing is installing it on a flash drive, booting to the drive on the Raspberry Pi, and then using NOOBS to choose and install an operating system.

NOOBS has plenty of operating systems for us to choose from when we reach that step – the most notable of which is Raspbian. For now, though let's concentrate on how to install NOOBS on the Raspberry Pi. We will briefly discuss the operating system installations later, in our final step.

The optional easy route: buy a NOOBS SD card.

Installing NOOBS on an SD card isn't hard, but it also isn't necessary. If you'd like, you can choose to buy an SD card that comes pre-loaded with NOOBS. If you go that route, you can skip all the way to the final step!

This project is pretty simple. Besides your Raspberry Pi and essential peripherals, here's all you'll need:

- A computer with an SD card slot
- An SD or microSD card of at least 8 GB

Step 1: Download NOOBS and extract it.

You're going to use your computer to put NOOBS on an SD card – so step one is to get NOOBS onto your computer!

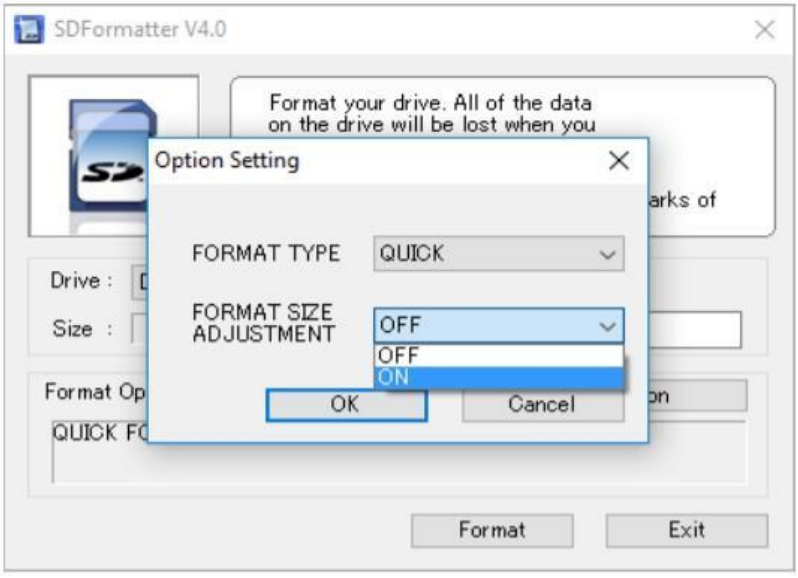
The NOOBS download page will let you choose between NOOBS and "NOOBS Lite." NOOBS includes a full version of Raspbian, so you can install that particular operating system without using the internet at all. With NOOBS Lite, on the other hand, you'll need a network connection to install any of the operating systems NOOBS makes available – even Raspbian.

Go ahead and choose whichever version you would like. NOOBS will download as a .zip file, so before you do anything else, go ahead and extract it.

Step 2: Format an SD card

Now you're going to want to go ahead and stick your SD card into the corresponding slot on your computer. You're going to want to format it as FAT. There are a few ways to do this:

On Mac or Windows, use the SD Association's Formatting Tool (Mac users can also just use the disk utility). Make sure the "Format size adjustment" option is set to "on." Then erase it in FAT (or MS-DOS) format.

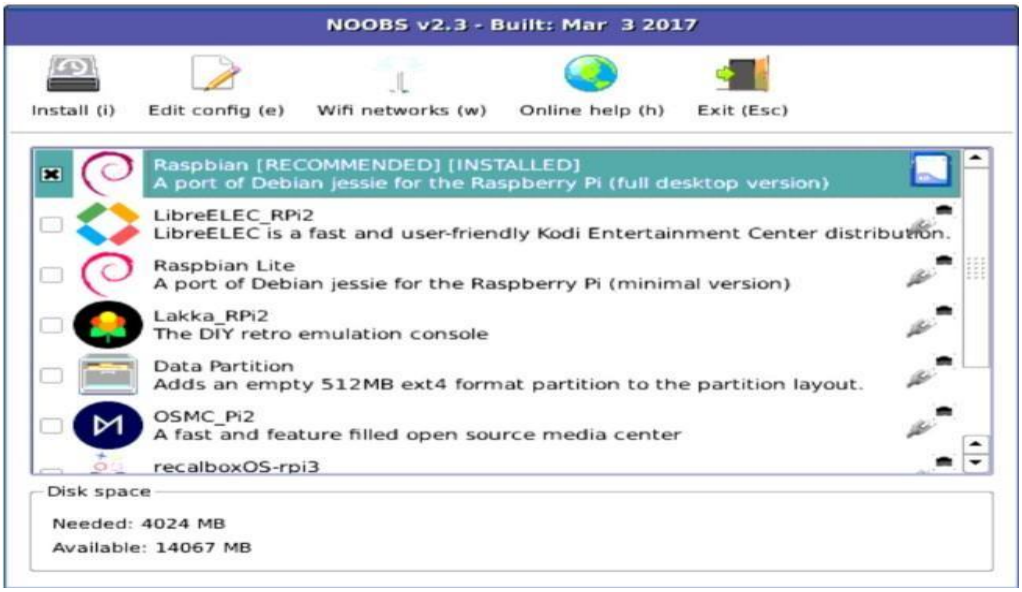


Step 3: Put the NOOBS files on the SD card

Now, just drag and drop the NOOBS files into your newly formatted SD card. You want the files only, so if your .zip extracted to a folder, open that folder up and select only the stuff inside of it.

Step 4: Put your SD card into your Raspberry Pi and boot it up

Name: Ankit Singh Chauhan
Roll No: 64



Once you have NOOBS on your SD card, using it is incredibly easy. Just put the SD card into your Raspberry Pi and start that sucker up. As we said before, while this guide is called “How to install NOOBS on the Raspberry Pi,” the endgame here is actually to install an operating system like Raspbian, LibreELEC, OSMC, or any of the other’s NOOBS gives you access to.

This is the step in which that happens. After booting to NOOBS, you’ll be greeted with a menu that will let you choose which operating system you’d like to install on your Pi. Your menu may look a little bit different than the one in the screenshot above, because NOOBS ingeniously adapts to your generation and model of Raspberry Pi.

Which OS should you choose? Well, that’s up to you. Raspbian is probably the most frequently used, and you’ll find plenty of projects here on our site that utilize it. OSMC acts as a media centre, and LibreELEC boots directly to the popular media centre app Kodi. Ultimately, it’s all a matter of personal preference!

Once you’ve decided, just hit “Install” and sit back. From now on, your Pi will boot directly to that operating system. Easy, right?

And if you’re not happy with the operating system you pick, you’re not stuck. Just hold down the SHIFT key while booting up, and you’ll be back in the NOOBS menu ready to try out a different option.

Practical No. 03

Aim: Displaying different LED with Raspberry Pi.

Name: Ankit Singh Chauhan

Roll No: 64

Class: T.Y.BSc.IT

Sub: IOT

Grade:

Sign:

Name: Ankit Singh Chauhan
Roll No: 64

Aim: Displaying different LED with Raspberry Pi.

The LED:

A Red LED stands for Light Emitting Diode, and glows when electricity is passed through it. When you pick up the LED, you will notice that one leg is longer than the other. The longer leg is known as ‘anode;, is always connected to the positive supply of the circuit. The shorter leg is known as ‘cathode’, and is always connected to the negative side of the power supply, known as ground.

The Resistor:

You must always use resistors to connect LEDs up to the GPIO pins of the Raspberry Pi. You will be using 330 Ohm Resistor. Yu can identify the 330 ohm resistor by the colour bands along the body.

REQUIREMENTS:

1. Raspberry Pi, SD Card
2. LED
3. Bread Board
4. Jumper Wires
5. Power Supply
6. Resistor
7. Monitor, Keyboard, Mouse

A) LED Blinking

PIN DIAGRAM:

NAME	PIN NUMBER
GPIO	40
GND	6

Name: Ankit Singh Chauhan
Roll No: 64

Step 1: Connect the led's using the pin connection table

Step 2: Open the terminal

Step 3: Open the file

- Sudo nano led.py

Step 4: Write the python code.

Input:

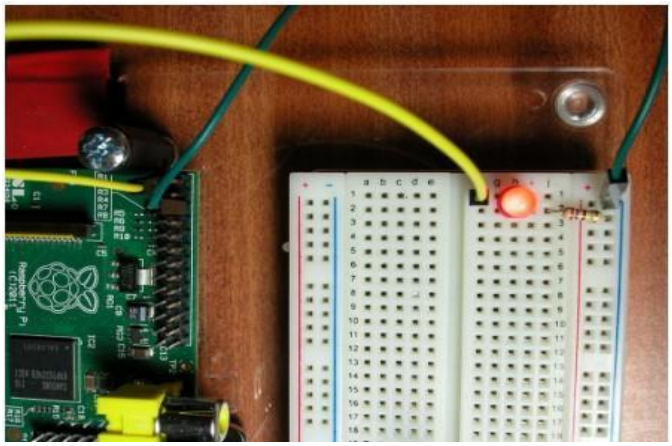
```
import RPi.GPIO as GPIO
import time
GPIO.setwarnings(False)
GPIO.setmode(GPIO.BOARD)
GPIO.setup(11,GPIO.OUT)
print(&amp;quot;press q to exist&amp;quot;);
try:
while True:
for num in range(0,2):
GPIO.output(11,GPIO.HIGH)
time.sleep(0.05)
except KeyboardInterrupt:
pass
GPIO.output(11,GPIO.LOW)
GPIO.cleanup()
```

Step 5: Run the code

sudo python led.py

Name: Ankit Singh Chauhan
Roll No: 64

Output:



B) LED Pattern Blinking

PIN DIAGRAM:

NAME	PIN DIAGRAM
GPIO	40
GPIO	37
GPIO	35
GPIO	33
GND	6

Step 1: Connect the led's using the pin connection table

Step 2: Open the terminal

Step 3: Open the file as

Sudo nano led.py

Step 4: Write the python code

Input:

```
import RPi.GPIO as GPIO
import time
GPIO.setmode(GPIO.BOARD)
led1=29
led2=31
```

```
led3=33
led4=35
led5=36
led6=37
GPIO.setup(led1,GPIO.OUT)
GPIO.setup(led2,GPIO.OUT)
GPIO.setup(led3,GPIO.OUT)
GPIO.setup(led4,GPIO.OUT)
GPIO.setup(led5,GPIO.OUT)
GPIO.setup(led6,GPIO.OUT)
def ledpattern(ledVal1,ledVal2,ledVal3,ledVal4,ledVal5,ledVal6):
    GPIO.output(led1, ledVal1)
    GPIO.output(led2, ledVal2)
    GPIO.output(led3, ledVal3)
    GPIO.output(led4, ledVal4)
    GPIO.output(led5, ledVal5)
   GPIO.output(led6, ledVal6)
def patterOne():
   for i in range(0,3):
      ledpattern(1,0,1,0,1,0)
      time.sleep(1)
     ledpattern(0,1,0,1,0,1)
     time.sleep(1)
def patternTwo():
   for i in range(0,5):
       ledpattern(1,0,0,0,0,0)
       time.sleep(0.1)
```

```
        ledpattern(0,1,0,0,0,0)
        time.sleep(0.1)
        ledpattern(0,0,1,0,0,0)
        time.sleep(0.1)
        ledpattern(0,0,0,1,0,0)
        time.sleep(0.1)
        ledpattern(0,0,0,0,1,0)
        time.sleep(0.1)
        ledpattern(0,0,0,0,0,1)
        time.sleep(0.1)
def patternThree():
    for i in range(0,5):
        ledpattern(0,0,0,0,0,1)
        time.sleep(0.1)
        ledpattern(0,0,0,0,1,0)
        time.sleep(0.1)
        ledpattern(0,0,0,1,0,0)
        time.sleep(0.1)
        ledpattern(0,0,1,0,0,0)
        time.sleep(0.1)
        ledpattern(0,1,0,0,0,0)
        time.sleep(0.1)
        ledpattern(1,0,0,0,0,0)
        time.sleep(0.1)
def patternFour():
    for i in range(0,5):
        ledpattern(0,1,1,1,1,1)
```

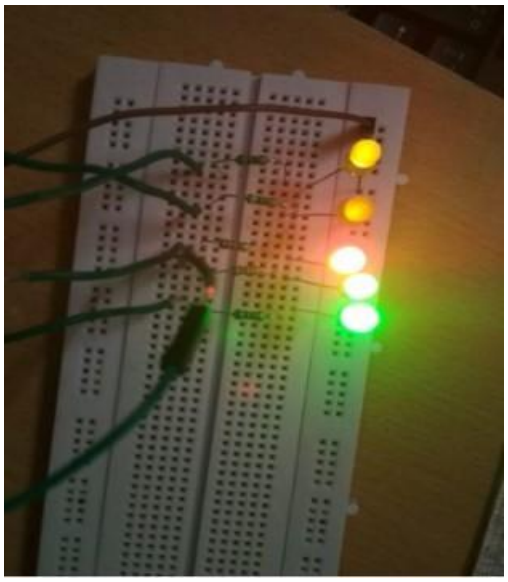
```
        time.sleep(0.1)
        ledpattern(1,0,1,1,1,1)
        time.sleep(0.1)
        ledpattern(1,1,0,1,1,1)
        time.sleep(0.1)
        ledpattern(1,1,1,0,1,1)
        time.sleep(0.1)
        ledpattern(1,1,1,1,0,1)
        time.sleep(0.1)
        ledpattern(1,1,1,1,1,0)
        time.sleep(0.1)
def patternFive():
    for i in range(0,5):
        ledpattern(1,1,1,1,1,0)
        time.sleep(0.1)
        ledpattern(1,1,1,1,0,1)
        time.sleep(0.1)
        ledpattern(1,1,1,0,1,1)
        time.sleep(0.1)
        ledpattern(1,1,0,1,1,1)
        time.sleep(0.1)
        ledpattern(1,0,1,1,1,1)
        time.sleep(0.1)
        ledpattern(0,1,1,1,1,1)
        time.sleep(0.1)
try:
    while True:
```

```
        patterOne()
        patternTwo()
        patternThree()
        patternFour()
        patternFive()
finally:
    #reset the GPIO pins
    GPIO.cleanup()
```

Step 5: Run the code

sudo python led.py

Output:



Practical No. 04

Aim: Displaying time over 4-digit 7- segment display using Raspberry Pi.

Name: Ankit Singh Chauhan

Roll No: 64

Class: T.Y.BSc.IT

Sub: IOT

Grade:

Sign:

Name: Ankit Singh Chauhan
Roll No: 64

Aim: Displaying time over 4-digit 7- segment display using Raspberry Pi.

To display small amount of data with Raspberry Pi, we can use 4 digit 7-segment

Display. 7 segment displays has seven segments in it and each segment has one LED inside it

to display the numbers by lighting up the corresponding segments.

REQUIREMENTS:

1. Raspberry Pi Model
2. Raspbian Stretch OS
3. 4-digit 7 Segment Display
4. Jumper wires
5. Monitor, Keyboard, Mouse

PIN DIAGRAM:

TM1637 Board Pin	RPI Physical Pin	Raspberry Function
GND (Ground)	14	GND
VCC (+5V Power)	4	5V
DI0 (Data In)	18	GPIO 24
CLK (Clock)	16	GPIO 13

Step 1: Download python script.

- ❖ **Create a folder 4digitTimer under /home/pi.**
- pwd
- ❖ **Download the script using wget command**

- Wget https://raspberrytips.nl/files/tm1637.py
- Ls

Step 2: Write the python code.

- Sudo nano clock.py

Input:

```
import sys
import time
import datetime
import RPi.GPIO as GPIO
import tm1637
#CLK -&gt; GPIO23(Pin 16)
#DI0 -&gt; 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)
```

Step 3: The above script needs the tm1637.py script to work, so place both files in the same

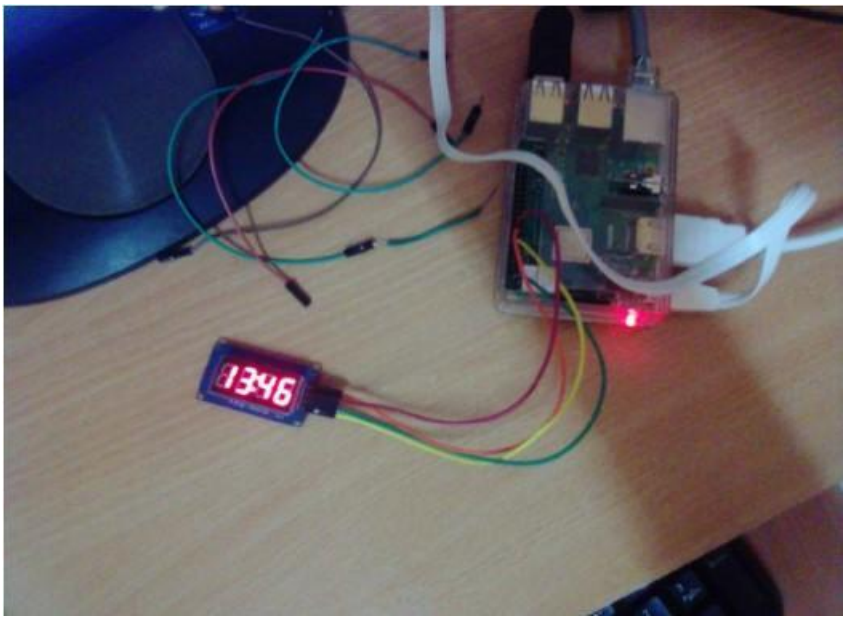
folder.

- Ls

Step 4: Run the code.

- sudo python clock.py

Output:



Practical No. 05

Aim: Pi Camera Module Interface with Raspberry Pi. (Photo Capturing)

Name: Ankit Singh Chauhan

Roll No: 64

Class: T.Y.BSc.IT

Sub: IOT

Grade:

Sign:

Name: Ankit Singh Chauhan
Roll No: 64

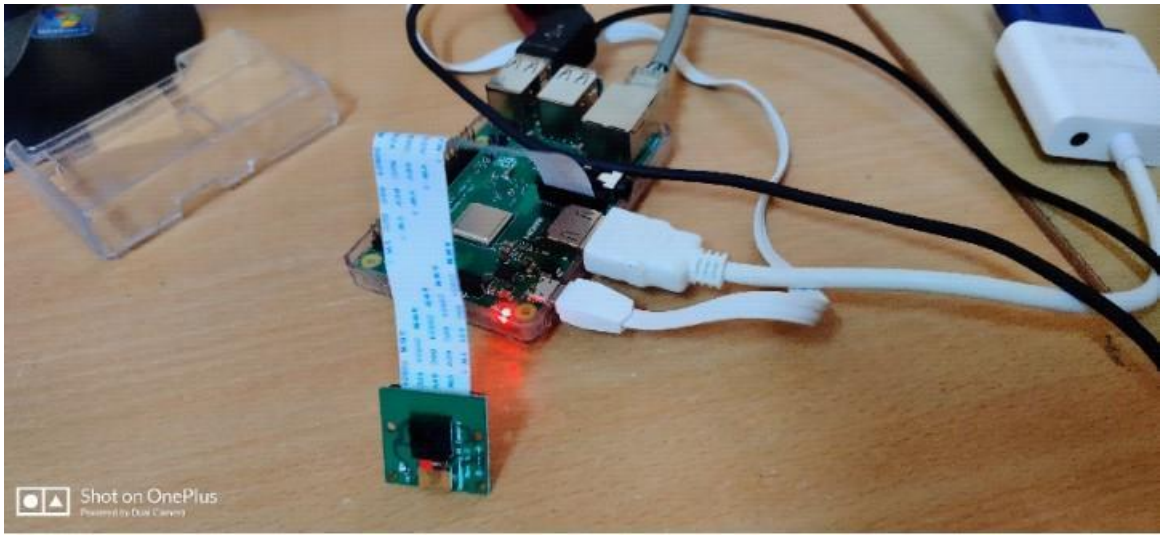
Aim: Pi Camera Module Interface with Raspberry Pi. (Photo Capturing):

Pi camera module is a camera which can be used to take pictures and high definition videos. 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

Features of Pi Camera

1. Resolution -5MP
2. HD video recording
3. It can capture wide, motionless images of resolution 2592x1944 pixels.
4. CSI Interface enabled.

Step 1: Connect the Pi Camera Module



Step 2: Open the terminal.

sudo raspi-config

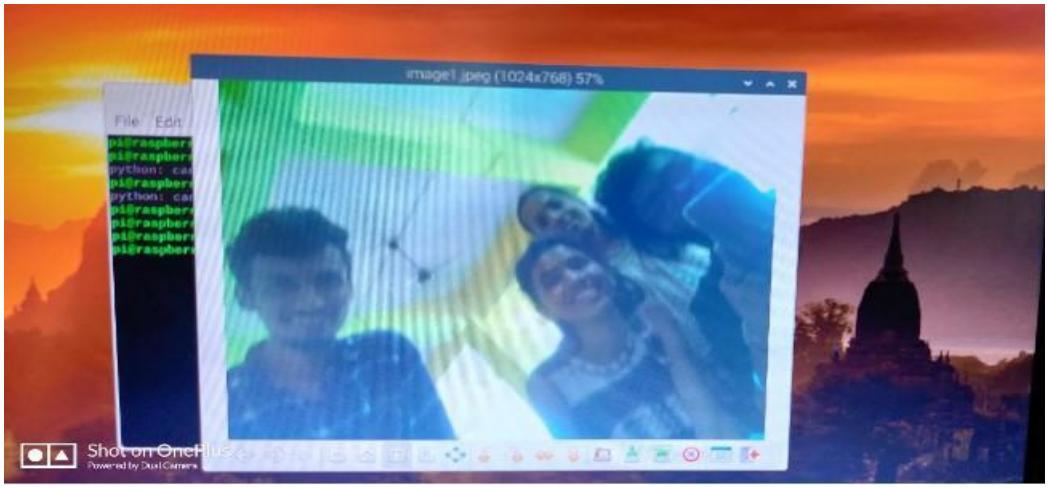
Then select Interfacing options in which select Camera option to enable its functionality.

Step 3: Write Python code

For Image Capture:

```
import picamera
from time import sleep
camera = picamera.PiCamera()
camera.resolution=(1024,768)
camera.brightness=60
camera.start_preview()
camera.annotate_text=&amp;quot;hi&amp;quot;
sleep(10)
camera.capture(&amp;#39;image1.jpeg&amp;#39;)
camera.stop_preview()
```

Output:



Name: Ankit Singh Chauhan
Roll No: 64

Practical No: 06

Aim: Pi Camera Module Interface with Raspberry Pi. (Video Recording)

Name: Ankit Singh Chauhan

Roll No: 64

Class: T.Y.BSc.IT

Sub: IOT

Grade:

Sign:

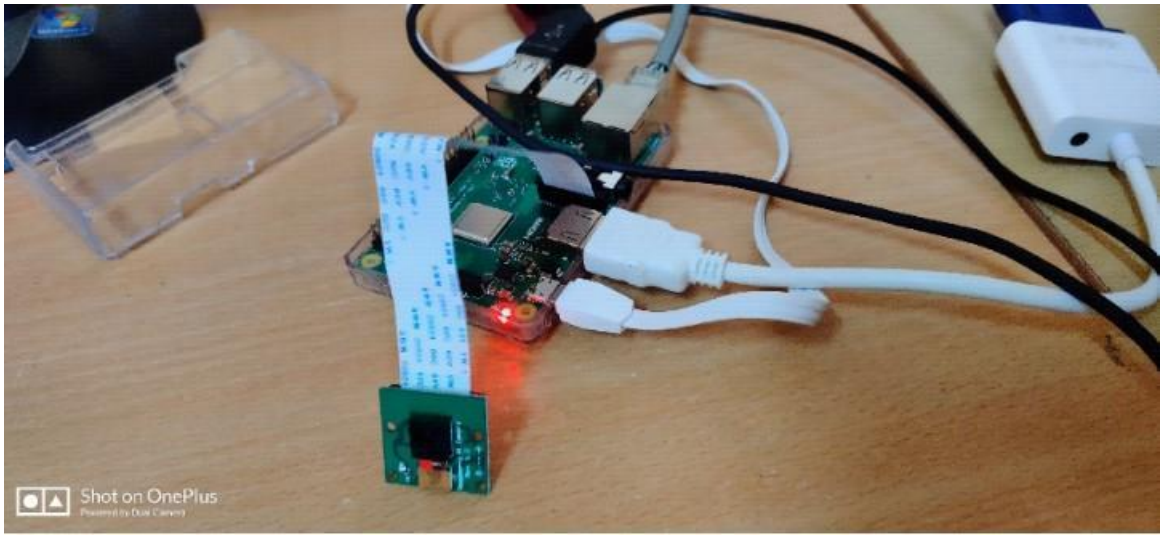
Name: Ankit Singh Chauhan
Roll No: 64

Aim: Pi Camera Module Interface with Raspberry Pi. (Video Recording):

Pi camera module is a camera which can be used to take pictures and high-definition videos. 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&#39's CSI port using 15-pin ribbon cable.

Features of Pi Camera

1. Resolution -5MP
2. HD video recording
3. It can capture wide, motionless images of resolution 2592x1944 pixels.
4. CSI Interface enabled.



Step 1: Connect the Pi Camera Module

Step 2: Open the terminal.

- sudo raspi-config

Then select Interfacing options in which select Camera option to enable its functionality.

Step 3: Write Python code

Name: Ankit Singh Chauhan
Roll No: 64

Code:

For Video Recording

```
import picamera
from time import sleep
camera=picamera.PiCamera()
camera.resolution=(640,480)
print()
camera.start_preview()
camera.start_recording(&amp;quot;/home/pi/demo.h264&amp;quot;)
camera.wait_recording(20)
camera.stop_recording()
camera.close()
print(&amp;quot;Video recording stopped&amp;quot;)
```

Step 4: Run the Code.

sudo python camera.py

Output:

Practical No 7

Aim: Interfacing Raspberry Pi with RFID.

Name: Ankit Singh Chauhan

Roll No: 64

Class: T.Y.BSc.IT

Sub: IOT

Grade:

Sign:

Name: Ankit Singh Chauhan
Roll No: 64

Aim: Interfacing Raspberry Pi with RFID.

In this practical, we will learn about the RFID Reader and Raspberry Pi RFID Reader Interface. We will be interfacing the RFID Reader Module with Raspberry Pi and access information from a few RFID cards through Python script.

RFID:

- RFID or Radio Frequency Identification is a way of communication over electromagnetic wave. RFID Tags and RFID Cards are often used for authentication and access control.
- The RFID RC522 is a very low-cost RFID (Radio-frequency identification) reader and writer that is based on the MFRC522 microcontroller.
- This microcontroller provides its data through the SPI protocol and works by creating a 13.56MHz electromagnetic field that it uses to communicate with the RFID tags.

REQUIREMENTS:

Raspberry Pi Model

RFID RC522

Few RFID Cards or RFID Tags

Power Supply for RFID Reader

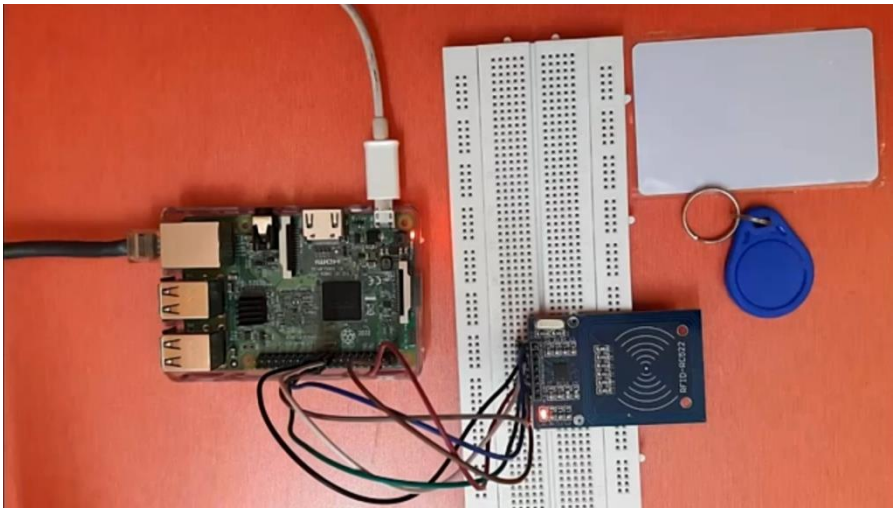
Connecting Wires

Name: Ankit Singh Chauhan
Roll No: 64

PIN DIAGRAM:

RFID Reader Board Pin	RPI Physical Pin	Raspberry Function
SDA	24	GPIO8 (SPI_CE0_N)
SCK	23	GPIO11 (SPI0_CLK)
MOSI	19	GPIO10 (SPI0_MOSI)
MISO	21	GPIO9 (SPI0_MISO)
IRQ	UNUSED	UNUSED
GND	6	GND
RST	22	GPIO25 (GPIO_GEN6)
3.3V	1	3.3V PWR

Step 1: Connect the RFID Reader using the pin connection table



Step 2: Open the terminal

Getting Python ready for the RFID RC522

- sudo apt-get update
- sudo apt-get upgrad

Setting up Raspbian for the RFID RC522

- sudo raspi-config
- sudo reboot
- lsmod | grep spi

Install python3-dev, python-pip and git packages

- sudo apt-get install python3-dev python3-pip
- cd
- pwd
- git clone https://github.com/lthiery/SPI-Py.git
- ls
- cd SPI-Py/
- pwd
- ls
- sudo python setup.py install
- git clone https://github.com/pimylifeup/MFRC522-python.git
- ls
- cd MFRC522-python/
- ls
- pwd

Step 4: Write the python code.

- sudo nano write.py

Write:

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

Step 5:

```
Sudo python write.py
New data: IBSARIIMS
Written
```

Step 6:

- pwd

- sudo nano read.py

Read:

```
import RPi.GPIO as GPIO
from mfrc522 import SimpleMFRC522
reader = SimpleMFRC522()
try:
        id, text = reader.read()
        print(id)
        print(text)
finally:
        GPIO.cleanup()
```

Step 7:

- Sudo python read.py

Output:

```
01232425010
IBSARIIMS
```

Practical No 8

Aim: Controlling Raspberry Pi with Telegram.

Name: Ankit Singh Chauhan

Roll No: 64

Class: T.Y.BSc.IT

Sub: IOT

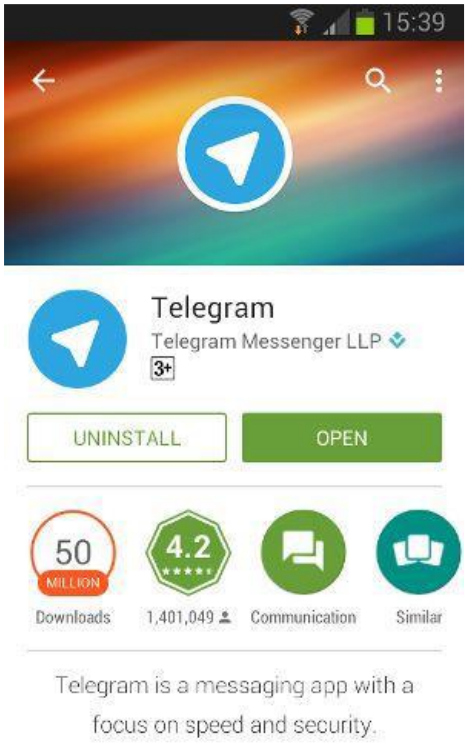
Grade:

Sign:

Name: Ankit Singh Chauhan
Roll No: 64

On 24 June 2015, Telegram published the BotAPI, enabling machines to talk Telegram. From that day on, not only can human use Telegram, so can machines.

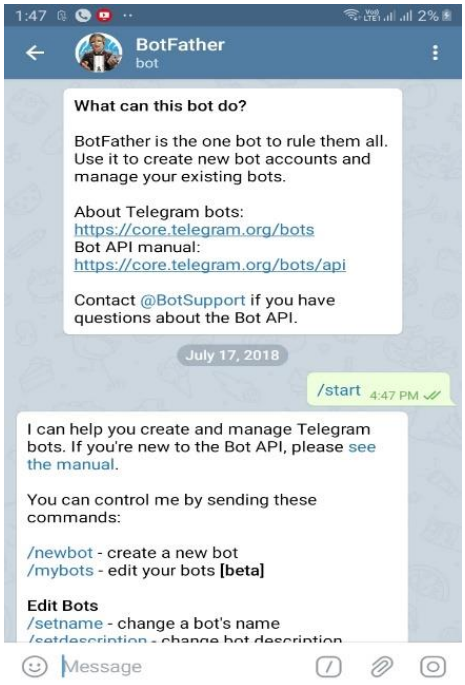
Step 1: Install Telegram on Cell Phone



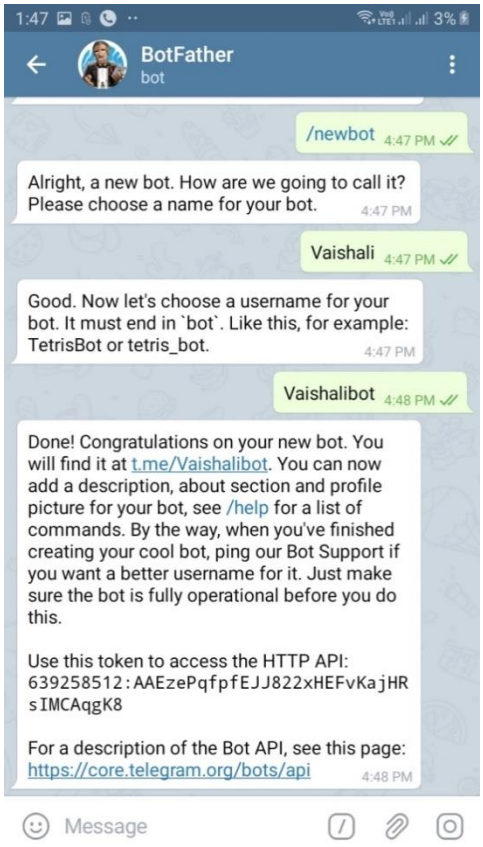
Step 2: Search for BotFather

Step 3: Click On BotFather

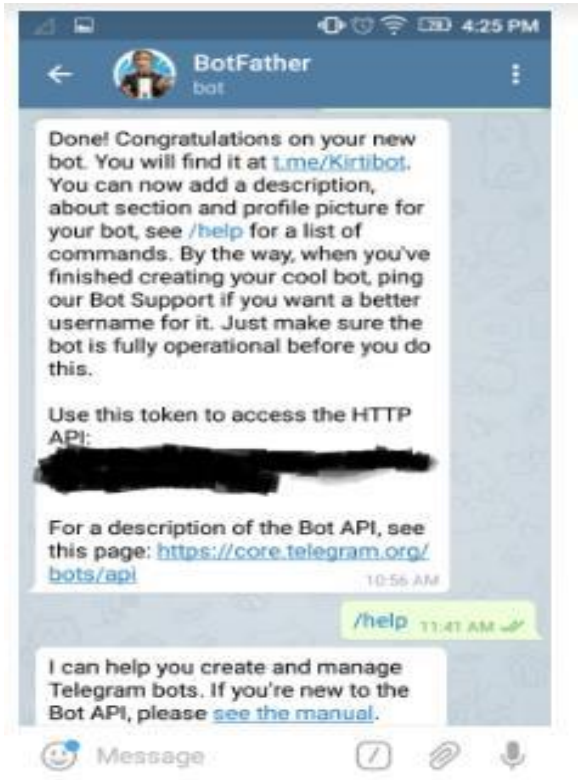
<u>Step 4</u>: To Start BotFather Type /start in Message



<u>Step 5</u>: Text /newbot to BotFather. Give name and username



<u>Step 6</u>: Obtain Access Token



Step 7: Setup on Raspberry Pi
1: Install "Python Package Index" and Teleport using:
sudo apt-get install python-pip
sudo pip install teleport

Step 8:

2: Test your bot using python 2 IDLE and type:
Import teleport
Bot - teleport.Bot('Bot Token')
Bot.getMe()

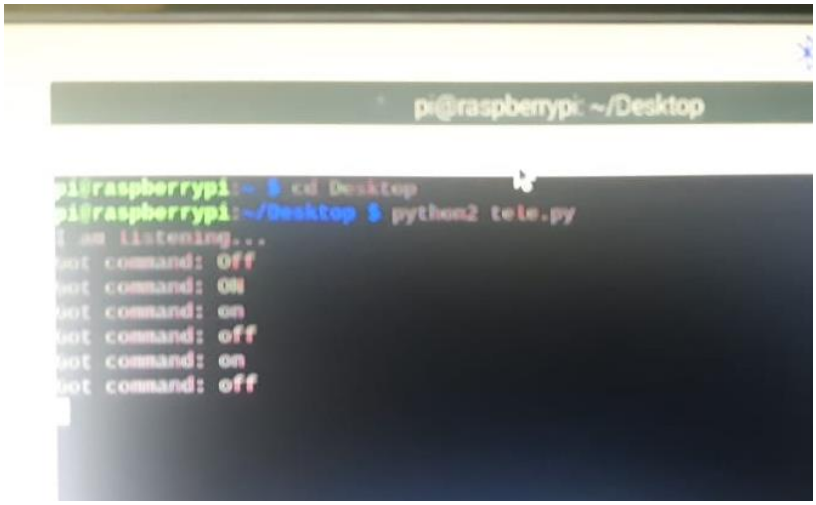
If it prints your bot details means everything is correct.
If not, then token is wrong.

Step 9: Write the Following Code in tele.py file

```
import sys
import time
import random
import datetime
import teleport
import RPi.GPIO as GPIO
GPIO.setmode(GPIO.BOARD)
GPIO.setup(7, GPIO.OUT)
def handle(msg):
    chat_id = msg['chat']['id']
    command = msg['text']
    print 'Got command:', command
    if command == 'on':
       bot.sendMessage(chat_id,"LED ON")
       GPIO.output(7,GPIO.HIGH)
    elif command =='off':
       bot.sendMessage(chat_id,"LED off")
       GPIO.output(7,GPIO.LOW)
    elif command =='stop':
          exit()
try:
  bot = telepot.Bot('639258512:AAEzePqfpfEJJ822xHEFvKajHRsIMCAqgK8')
  bot.message_loop(handle)
  print ('I am listening...')
  while 1:
    time.sleep(10)
```

```
except TelegramError:
    print('')
```

Step 10:



Step 11:

Name: Ankit Singh Chauhan
Roll No: 64

Output: