Practical No:-1

Aim:- Starting Raspbian os, Installing with Raspberry component

Theory:-

Steps for Installation of NOOBS and Raspbian are:-

Formating the SD card using SD formatter

Step 1:- Downloading NOOBS

using NOOB is the easiest way to install Raspbian on your SD card. To get copy of NOOBS:- www.raspberryPi.org/download

Step 2:- Installing NOOBS

Step 3:- Formatting the SD card

Step 4:- Extracting NOOBS from zip archieve

1. Go to your Download folder and find the zip file downloaded
2. Select and extract

Step 5:- Copying the files

1. Now open another Explorex window and Navigate SD card
2. Select all files and copy it.

Step 6:- Booting from NOOBS

1. Once copy is Done, plug in Power Supply.
2. Boot the software for first time.

Installing using Image file :-
Step 1 :- Download Raspbian
Step 2 :- Unzip the file
Step 3 :- Write the disc image to your card
Step 4 :- put SD card in your pi and boot up
Step 5 :- Initial configuration of Rasbian OS

Rasbian Variant :-

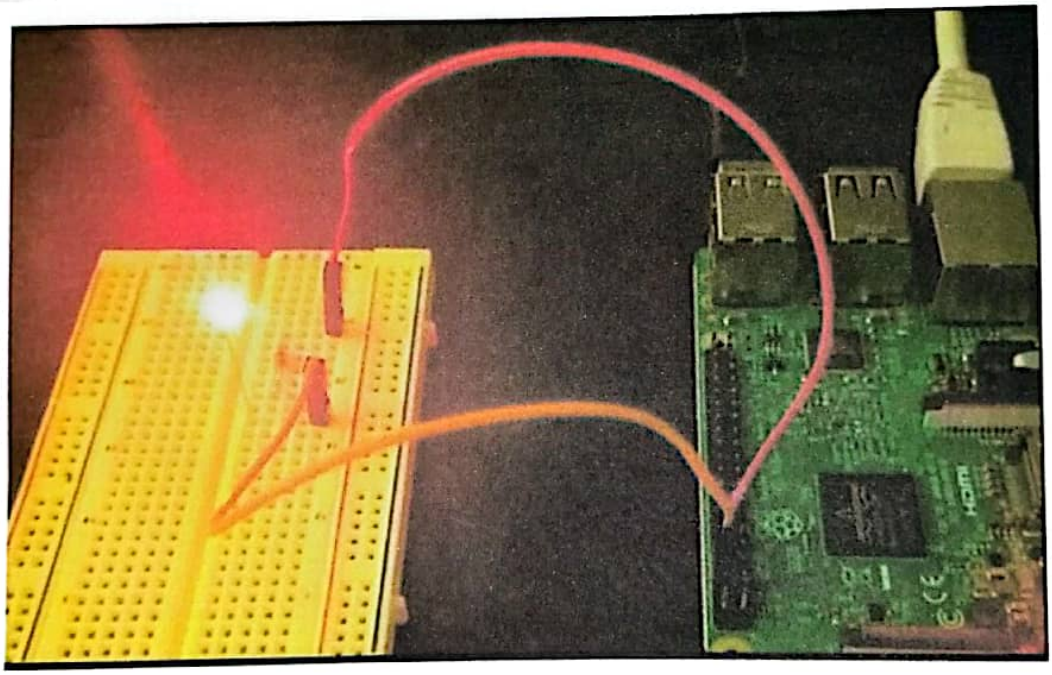
1. Rasbian stretch with Desktop and recommended software. This version comes with complete stack of Rasplian OS Desktop Environment (GUI) and also bundled software. It is recommended for most people starting with Raspberry Pi.

2. Rasphian Stretch with Desktop :-
This one does not come with bundled software tool. Just the Desktop Environment (GUI)

3. Rasphian Stretch lite :-
This version is base-bone and does not include Desktop environment. It is recommended for advanced class
Step 6 :- Set Country
Step 7 :- Set Password
Step 8 :- Set Network
Step 9 :- Set Screen
Step 10 :- Update Software
Step 11 :- Setup Complete

Pin diagram :-

3.3V	1	2	5V
GPIO2	3	4	5V
GPIO3	5	6	GND
GPIO4	7	8	GPIO14
GND	9	10	GPIO15
GPIO17	11	12	GPIO18
GPIO27	13	14	GND
GPIO22	15	16	GPIO23
3.3V	17	18	GPIO24
GPIO10	19	20	GND
GPIO9	21	22	GPIO25
GPIO11	23	24	GPIO8
GND	25	26	GPIO7
DNC	27	28	DNC
GPIO6	29	30	GND
GPIO13	31	32	GPIO12
GPIO19	33	34	GND
GPIO26	35	36	GPIO16
GPIO5	37	38	GPIO12
GND	39	40	GPIO21



Practical No :- 2

Aim :- GPIO : light the led with python

Theory :-

Hardware Requirements :-

Along with the basic setup you will require the following components to get started with the GPIO Pins as follows.

1. LED
2. Resistor
3. Connecting wires
4. Breadboard

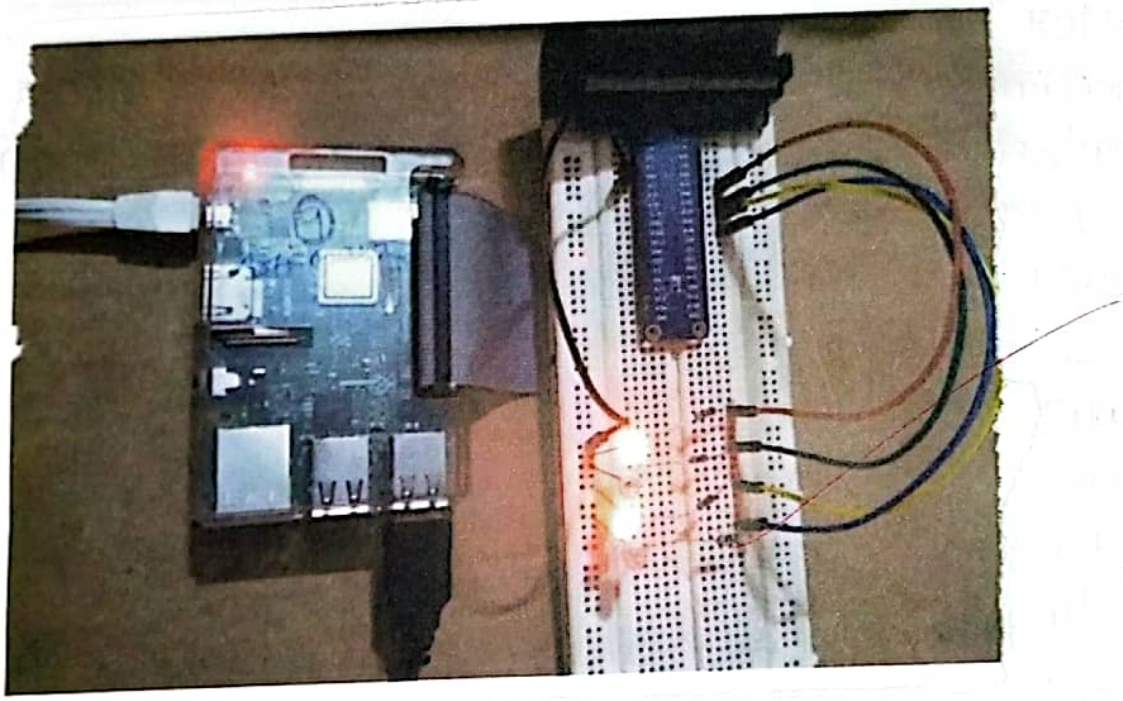
Before learning this lesson, you must understand the pin numbering system of the GPIO pins

Software Requirements :-

Raspbian OS comes with many pre installed instarted programming environments. Here we will be using python for coding

Description :-

To light on LED with python using GPIO on a Raspberry Pi, first connection the LED's anode to a GPIO pin and the cathode to a ground pin through a resistor. Then, in your python code, import the 'RPi.GPIO' library, set up the GPIO mode and configure the selected GPIO pin as an output. Finally use the 'GPIO.output()' function to send a HIGH signal to the pin, which will turn on the LED.



Pin Configuration :-

Step 1 :-

Connect the GPIO22 pin of raspberry pi to one end of resistor

Step 2 :-

connect the another end of resistor to the positive end of LED

Step 3 :-

Connect the negative end of LED to Ground of raspberry Pi.

Step 4 :-

Then power on your raspberry Pi

Code :-

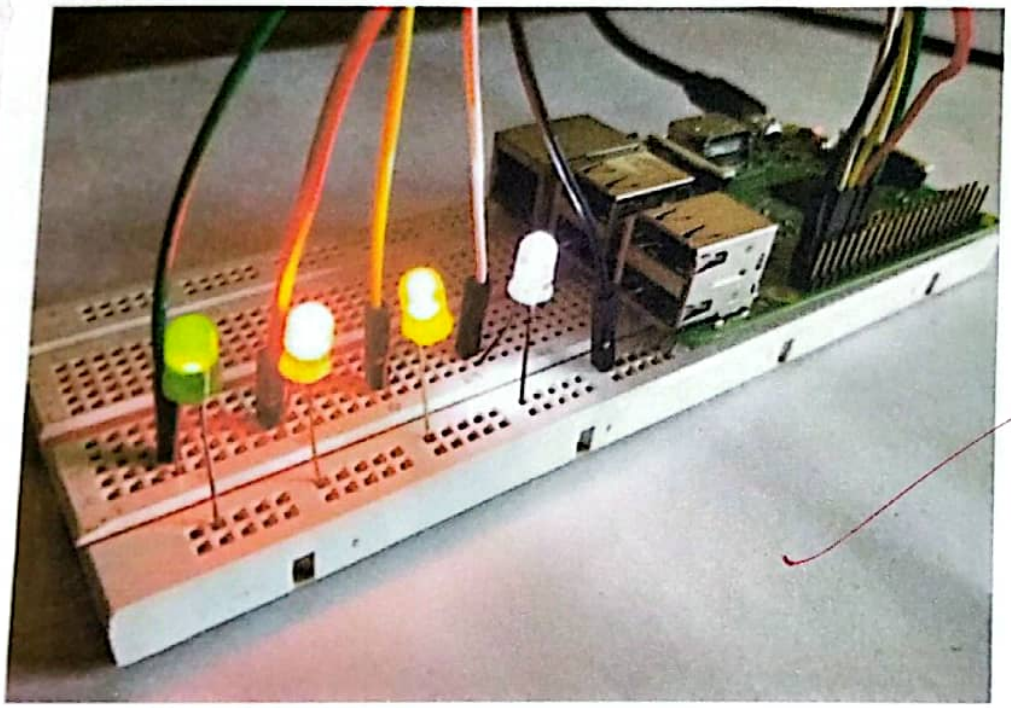
```
#Blink LED program
# Connect the LED to GPIO 22 pin
# LED Blink program
# Connect the LED to GPIO22

#import GPIO and time library
import RP ; GPIO as GPIO
from time import sleep

GPIO.setmode (GPIO.BCM)
ledPin = 22

#setup the led Pin (i.e. GPIO22) as output
GPIO.setup (ledpin, GPIO.OUT)
GPIO.output (ledpin, False)
```

```
try :
  While True :
  GPIO.output (ledPin, True)
  Print ("LED ON")
  sleep (1)
  GPIO.output (ledPin, False)
  print ("LED OFF")
  sleep (1)
finally
  GPIO.output (ledPin, False)
  GPIO.cleanup ()
```



Practical No:-3

Aim:- Displaying Different led pattern with Raspberry Pi

Theory :-

Hardware requirements :-

1 X Breadboard
1 X Raspberry Pi
1 X RGB LED
1 X 330 Ω Resistor
2 X Jumper wires

Software requirements :-

The Raspbian OS should be installed on your Raspberry Pi, as it provides the necessary environment for running python code.

Description :-

Using a Raspberry Pi to Create and Control various LED patterns by programming its GPIO pins with python.
By Configuring these pins, different LED sequences, such as blinking, chasing, or custom patterns, can be displayed. This display how the Raspberry Pi can be used to bridge the gap between coding and physical electronics

Code :-

```
import RPi.GPIO as GPIO
from time import sleep
GPIO.setWarnings(False)
GPIO.setmode(GPIO.BOARD)
```

```
led1 = 11
led2 = 12
led3 = 15
led4 = 16
GPIO.setup(led1, GPIO.OUT)
GPIO.setup(led2, GPIO.OUT)
GPIO.setup(led3, GPIO.OUT)
GPIO.setup(led4, GPIO.OUT)
GPIO.output(led1, False)
GPIO.output(led2, False)
GPIO.output(led3, False)
GPIO.output(led4, False)
def ledpattern(ledval1, ledval2, ledval3, ledval4):
    GPIO.output(led1, ledval1)
    GPIO.output(led2, ledval2)
    GPIO.output(led3, ledval3)
    GPIO.output(led4, ledval4)
def patternOne():
    for i in range(0,3):
        ledpattern(1,0,1,0)
        sleep(0.5)
        ledpattern(0,1,0,1)
        sleep(1)
    try:
        while True:
            patternOne()
    finally:
        GPIO.cleanup()
```

Practical No:-4

Aim:- Display time over four digit seven segment display using Raspberry Pi

Theory:-

Hardware requirements

For completing the lesson you will require following things along with your initial raspberry pi setup

Step 1:

TM1637 4-digit seven segment Display board

Step 2:

Connecting wires

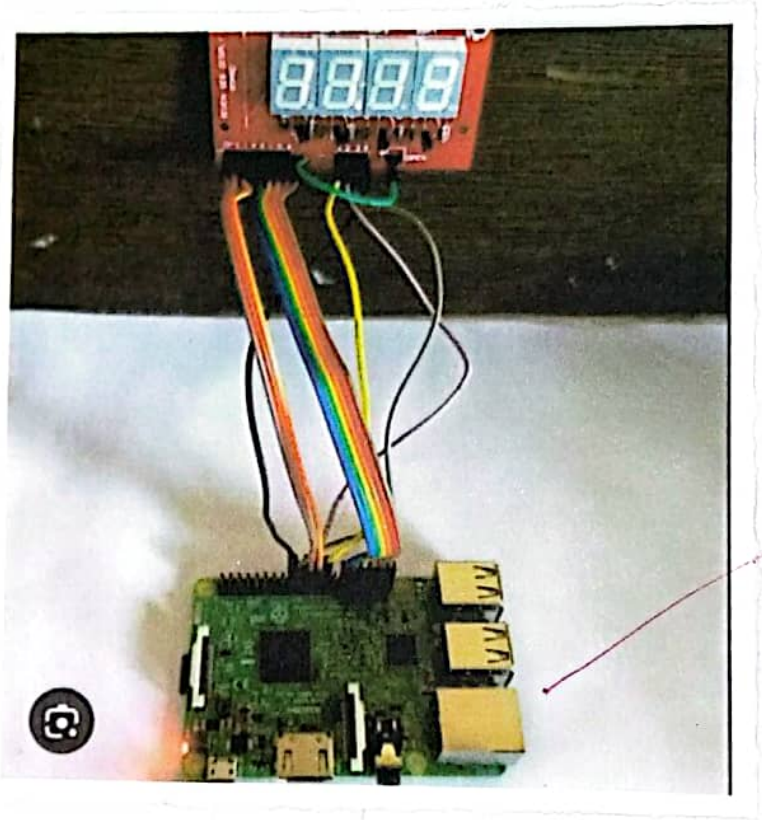
Software Requirements:-

Additionally, you may need the 'tm1637' library if you are using a TM1637-based 7 segment display, as it simplifies communication between the Raspberry Pi and the display module

Description:-

Using a Raspberry Pi to display the current time on a 4 digit 7 segment display.

The Raspberry Pi controls the display through its GPIO pins, with python scripts managing the timekeeping and updating the display in real time. By lighting up the appropriate segments the time is accurately shown.



Pin Configuration :-

step 1

connect the pin2 (5V) of Rpi to VCC pin of Module

step 2

connect the pin 6 (GND) of Rpi to GND of Module

step 3

connect Pin38 (GPIO20) of Rpi to DIO of module

step 4

Lastly connect pin40 (GPIO21) of RPi to CLK of Module.

Code :-

```
import RPi.GPIO as GPIO
from time import sleep
import tm1637
GPIO.setwarnings(False)

try:
    import thread
except ImportError:
    import _thread as thread
Display = tm1637.TM1637(CLK=21, DIO=20, brightness=1.0)
try:
    print("Starting clock in the background:")
    Display.startClock(military_time=True)
    Display.setBrightness(1.0)
while True:
```

```
        Display.showDoublepoint(True)
        sleep(1)
        Display.showDoublepoint(False)
        sleep(1)

        Display.StopClock()
        thread.interrupt_main()
except KeyboardInterrupt:
        print('properly closing the clock and open GPIO pins')
        Display.cleanup()
```

Practical No:-5

Aim:- Interfacing raspberry pi with telegram app.

Theory:-

Hardware Requirements :- Raspberry pi, Led (Generic)
Jumper wires (Generic), Breadboard (Generic)

Description:- On 24 June, 2015, Telegram published the Bot API, enabling machines to talk Telegram. From that day on, not only can human use Telegram, so can machines. For those whoe doesn't known how Telegram is, it is a messaging app, very much like whatsApp. This tutorial is going to teach how to send a Telegram message to your Raspberry Pi, and how to make your Pi telegram back. Make sure the Pi has ~~telegram~~ internet access.

Steps:-

Step 1: Install Telegram on your Phone, Obviously.

Step 2: Text /newbot to Botfather

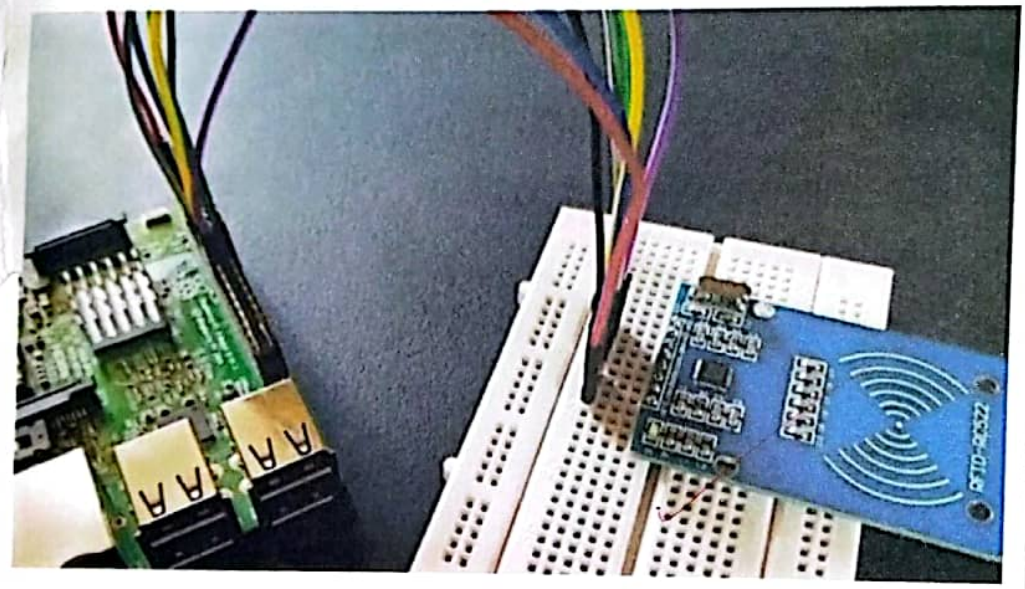
Open Telegram on your phone, search for a useetr called Botfather. As the name implies, he is the father of all Bots.

Step 3: Install Telepot on Raspberry Pi

On the command line, run these two commands

Sudo apt-get install python-pip

Sudo pip install telepot.



Step 4: Test token

In python interpreter, enter these three lines;

```
import telepot
bot = telepot.Bot('*** Copy bot token from botfather ***')
bot.getMe()
```

Step 5: Write the python code

Step 6: Connect red LED at pin 40 (GPIO 21)

Step 7: Run it and test it

Code:-

```
import sys
import time
import random
import datetime
import telepot
import RPI.GPIO as GPIO
from telepot.loop import MessageLoop
red = 40
now = datetime.datetime.now()
GPIO.setmode(GPIO.BOARD)
GPIO.setwarnings(False)
GPIO.setup(red, GPIO.OUT)
GPIO.output(red,0)
def action(msg):
    chat_id = msg['chat']['id']
    command = msg['text']
```

```
print ('Got command : %s' %command)
if 'On' in command:
    message = "Turn on"
    message = message + "red"
    GPIO.output (red, 1)
    bot.sendMessage (chat_id, message)
if 'off' in command:
    message = "Turn off"
    message = message + "red"
    GPIO.output (red, 0)
    bot.sendMessage (chat_id, message)
bot = telepot.Bot ('626665131:AAHsNzQbqSj9GZ9-w2t4r')
print (bot.getMe())
MessageLoop (bot, action).run_as_thread()
print ('I am listening...')
while 1:
    time.sleep(10)
```

Practical No:- 6

Aim:- Oscilloscope using Raspberry Pi

Hardware Requirements: Raspberry pi2
8 or 16 GB SD Card
LAN / Ethernet Cable
Power Supply or USB cable
ADS1115 ADC
10K or 1K resistor
Jumper Wires
Breadboard

Software Requirements:
1) matplotlib
2) drawnow

Description : The oscilloscope is an electronic test instrument that allows the visualization and observation of varying signal voltages, as a two dimensional plot with one or more signals plotted against time. Todays 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.

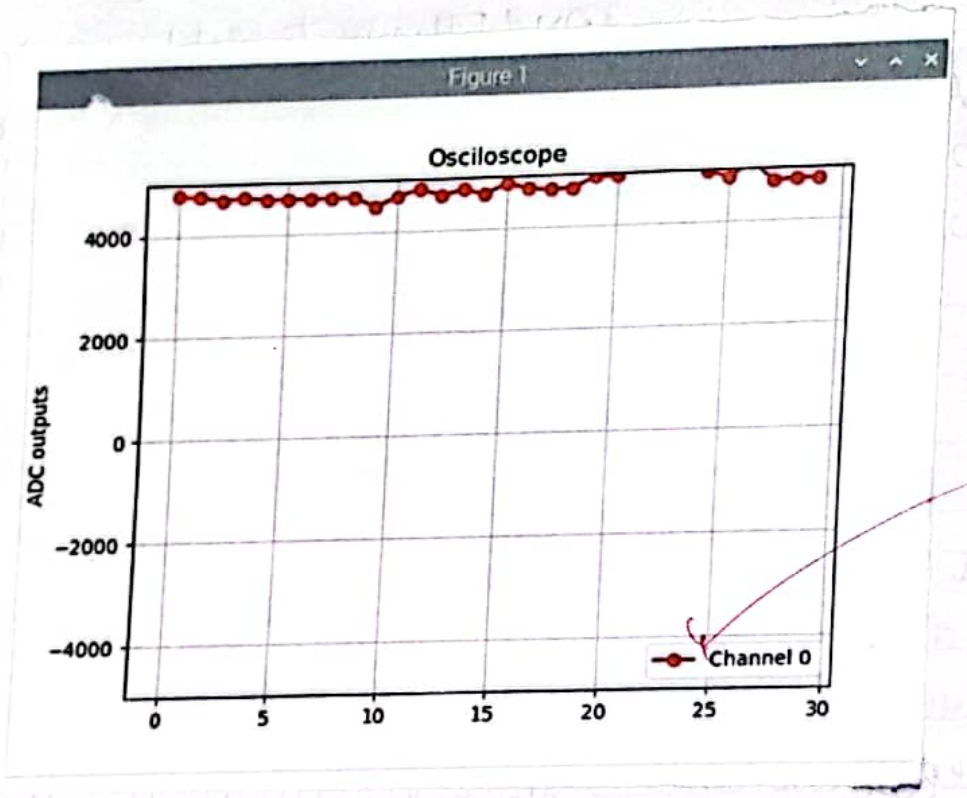


Figure 1
Osciloscope
ADC outputs
4000
2000
0
−2000
−4000
0
5
10
15
20
25
30
Channel 0

ADS115 and Raspberry Connections:
1. VDD – 3.3V Raspberry pi pin 1
2. GND = GND Raspberry pi pin 6
3. SCL = SCL Raspberry pi pin 5
4. SDA = SDA Raspberry Pi pin 3

Step1: Enable Raspberry pi I2C interface
Sudo raspi-config
[when the Configuration panels open, select interface options, select I2C and click enable]

Step2: Update the Raspberry pi
Sudo apt-get update
Sudo apt-get upgrade } commands

Step3: Install the Adafruit ADS115 library for ADC cal ~
[Ensure you are in the Raspberry Pi home directory by running]
Sudo apt-get install build-essential Python
[clone the Adafruit git folder for the library by running
git clone https://github.com/adafruit/Adafruit Python ADS1x15.git
i change into the clone file's directory and run setup file
cd Adafruit_Python_ADS1x1z
Sudo python setup.py install

Step 4: Test the library and I2C communication cd example
[Next run the Sampletest.py example which displays the value of four channels]
python simplest.py

Step 5: Install Matplotlib
sudo apt-get install python-matplotlib

Step 6: Install the Drawnow python module
sudo apt-get install python-pip
sudo pip install drawnow

CODE:

```
import time
import matplotlib.pyplot as plt
from drawnow import *
import Adafruit_ADS1x15
adc = Adafruit_ADS1x15.ADS1115()
GAIN = 1
val = []
int = 0
plt = 0
plt.ion()
adc.start_adc(0, gain=GAIN)
print('Reading ADS1x15 channel 0')
def makeFig():
plt.ylim(-5000, 5000)
plt.title('Oscilloscope')
```

```
plt.grid (True)
plt.plable ('ADC outputs')
plt.plot (val, 'no', label=channel 0')
plt.legend
(loc='lower right')
while (True):
  value = adc.get_last_result ()
print ('Channel 0: {0}': format (value))
# sleep for a half second
time.sleep (0.5)
val.append (int (value))
drawnow (makefig)
plt.pause (.0000001)
cnt = cnt+1
if (cnt>50):
  val.pop (0)
```

Practical No: 7

Aim: Interfacing using Telegram App.

Theory:
Interfacing hardware with telegram involves using a microcontroller or single board computer like the BOT API, facilitates the remote communication between the hardware and the Telegram App. By leveraging libraries such as: python-telegram-bot for Python you can send and receive messages to control devices or monitor sensor data remotely through the telegram interface, enabling simple and accessible IOT or automation solution.

Step 1:
Install telegram on your phone.

Step 2:
Text/Newbot to Botfather

Step 3:
Install Telepot on raspberry pi

On the Command line, run these two commands
Sudo apt-get install python-pip
Sudo pip install telepot

Step 4:
Test token

grptelepot.py

```
1  import sys
2  import time
3  import random
4  import datetime
5  import telepot
6  import RPi.GPIO as GPIO
7  from telepot.loop import Mess
8
9  red=40
10 now=datetime.datetime.now()
11 GPIO.setmode(GPIO.BOARD)
```

Shell

```
i am listening...
got command :/start
got command :Hi
got command :Helloo traouuuu
got command :On
got command :Off
got command :Heyyy everyone
got command :Good morning
```

Step 5:
python code

```
import sys
import time
import random
import datetime
import telepot
import RPi.GPIO as GPIO
from telepot.loop import MessageLoop

red = 40

now = datetime.datetime.now()
GPIO.setmode (GPIO.BOARD)
GPIO.setwarnings (False)

GPIO.setup (red, GPIO.OUT)
GPIO.output (red, 0)

def action (msg):
  chat_id = msg ['chat'] ['id']
  Command = msg ['text']
  print ('Got command: %s' % Command)
    if 'on' in command:
    message = "Turn on"
    message = message + "red"
```

```
if 'OFF' in command:
    message = "Turn off"
    message = message + "red"
    GPIO.output(red, 0)
    bot.SendMessage(chat_id, message)

bot = telepot.Bot('626665131:AAHsNzQbqSj9G29-W2+41')
print(bot.getMe())
MessageLoop(bot, action).run_as_thread()
print 'I am listening ...'

while 1:
  time.sleep(10)
```

Step 6:
Connect red LED at pin 40 (GPIO21)

Step 7:
Run it and Text it.

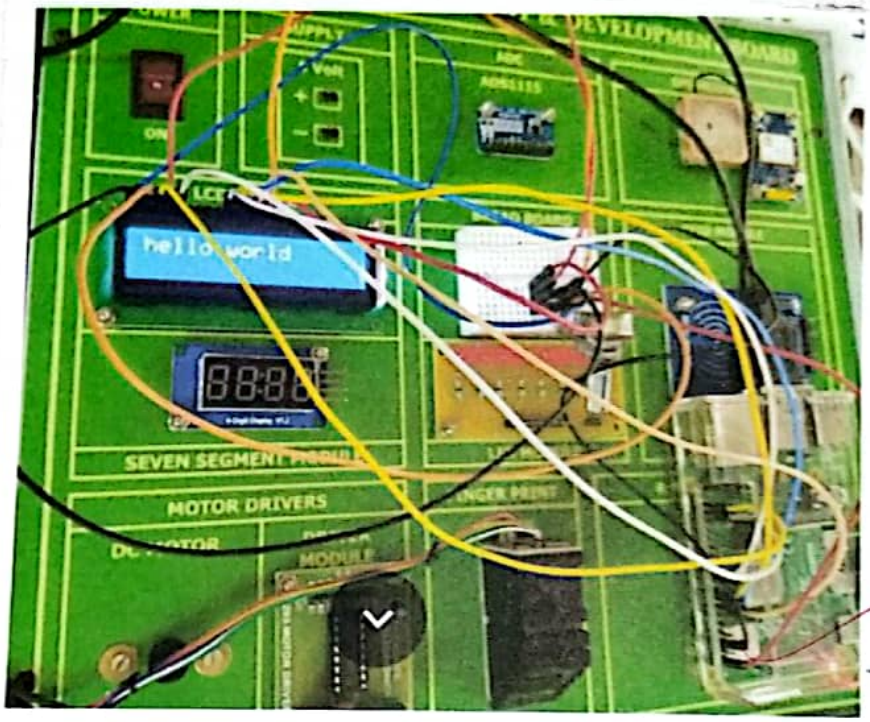
Practical No: 8

Aim:- 16X2 LCD Display using Raspberry

Theory:-

Hardware Requirements
1. Raspberry Pi 4
2. Micro SD card
3. 16X2 LCD display
4. Breadboard
5. Jumper Wires
6. Potentiometer
7. Resistor

Pin Configuration:
1. Raspberry Pi 5v (physical Pin 4) to LCD Pin 2, 3 and one side of the Potentiometer
2. Raspberry Pi GND to LCD pin 1, 5, 16 and the opposite side of the potentiometer.
3. Potentiometer output (middle pin) to LCD pin 3.
4. Raspberry Pi GPIO26 (Pin 37) to LCD Pin 4
5. Raspberry Pi GPIO19 (Pin 35) to LCD Pin 6
6. Raspberry Pi GPIO25 (Pin 22) to LCD Pin 11
7. Raspberry Pi GPIO24 (Pin 18) to LCD Pin 12
8. Raspberry Pi GPIO22 (Pin 15) to LCD Pin 13
9. Raspberry Pi GPIO27 (Pin 13) to LCD Pin 14



Source Code:

```
import time
import board
import digitalio
import adafruit_character_lcd.character_lcd as charlcd

lcd_column = 16
lcd_row = 2
lcd_rs = digitalio.DigitalInOut(board.D26)
lcd_en = digitalio.DigitalInOut(board.D19)
lcd_d4 = digitalio.DigitalInOut(board.D25)
d5 = digitalio.DigitalInOut(board.D24)
d6 = digitalio.DigitalInOut(board.D22)
d7 = digitalio.DigitalInOut(board.D27)
back = digitalio.DigitalInOut(board.D4)
lcd = charlcd.Character_LCD_Mono(lcd_rs, lcd_en, lcd_d4,
                                 d5, d6, d7, lcd_column, lcd_row, back)
lcd.message = "IOT Practical"
time.sleep(5)
lcd.clear()
```

Step1:

Install the required Packages.

```
Sudo pip3 install Adafruit-Blinka
Sudo pip3 install adafruit-circuitpython-charlcd
```

Practical No: 9

Aim: Capturing Images with Raspberry Pi and Pi Camera

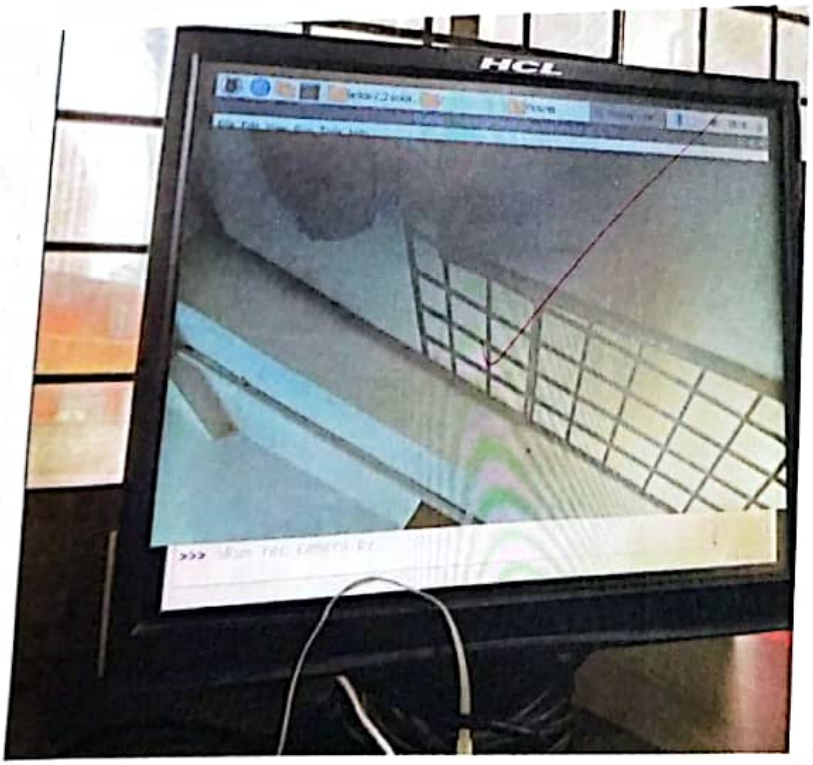
Theory:

Hardware Requirements:
Camera Module with your initial raspberry pi Setup.

Camera Module:
The Raspberry Pi Camera Board Plugs directly into the connector on the Raspberry Pi. The Camera is Supported in the latest version of Raspbian, the Raspberry Pi's Preferred operating System.

Connect to Camera Module:
Switch off the Raspberry Pi, you'll need to connect the camera module to the Raspberry Pi's camera Port, then start up the Pi and ensure the Software is enabled.

1. Local the camera port and connect the Camera.
2. Start up the Pi.
3. Open the Raspberry Configuration tool from main menu.
4. Ensure the camera software is enabled, if it is not enabled, then enable it and reboot your pi to began.



Software Requirements:

1. Open Terminal window and Type the command as follows:

$ sudo raspistill -o /home/pi/Desktop/image.jpg

This command will capture an image and store it to the specified location with specified name.

Source Code:

Image

```
from time import sleep
from picamera import PiCamera
camera = PiCamera()
camera.resolution = (1280, 720)
camera.start_preview()
sleep(2)
camera.capture("/home/Pi/Pictures/newimg.jpg")
Camera.stop_preview()
```

Video

```
From time import sleep
From picamera import PiCamera
camera = PiCamera()
camera.resolution = (1280,720)
camera.start_recording("/home/pi/video/newimage2.h264")
Camera.start_Preview()
Sleep(10)
Camera.stop_recording()
Camera.stop_Preview()
```

Practical No: 10

Aim:- Raspberry Pi based on LED Matrix

Theory:-

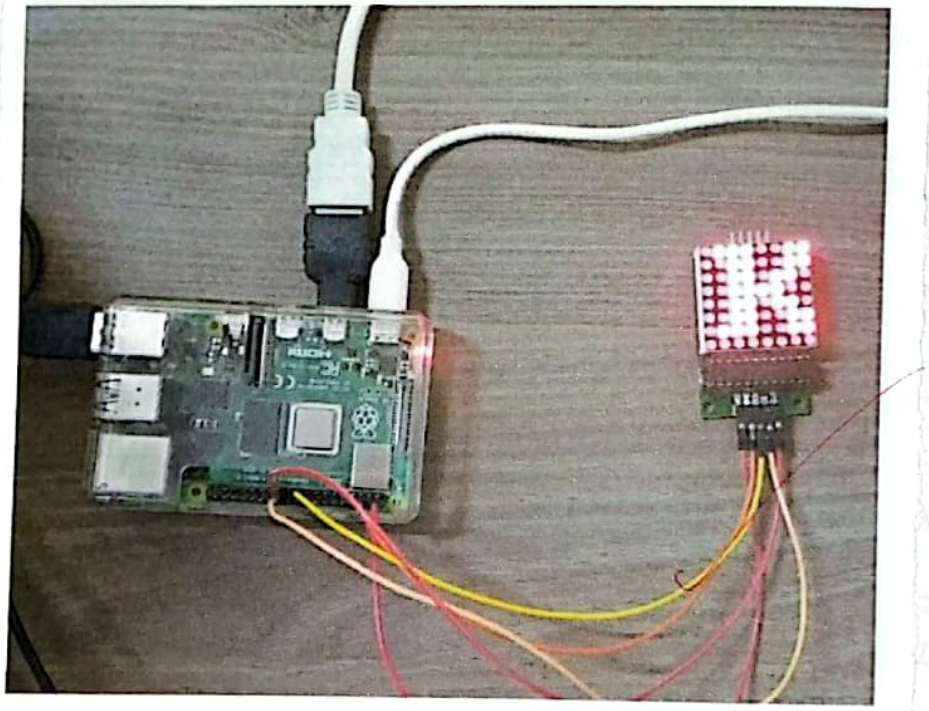
A Raspberry Pi-based LED Matrix system involves using the Raspberry Pi Microcontroller to control and display patterns or images on an LED matrix. LED matrices are grids of light emitting diodes arranged in rows and columns. Each LED represents a pixel that can be turned on or off to form patterns, numbers, letters, or animations.

Hardware Requirements:

1. Raspberry Pi : Main Controllers
2. LED Matrix : Display Panel
3. power Supply : for Raspberry Pi and matrix
4. Jumper wires : for GPIO connections
5. Driver (eg, MAX 7219) : Control the matrix

Software Requirements:

1. Raspbian OS : Raspberry Pi Operating System.
2. Python : Programming language.
3. Libraries.



4. GPIO libraries for Pin Control

Code:

```
import re
From time import sleep
From luma.led_matrix.device import max7219
From luma.core.interface.serial import spi, noop
From luma.core.legacy import show_message
From luma.core.legacy.font import proportional, CP437_FONT
def demo(n, block_orientation, rotate, inverse):
# create matrix device
Serial = spi(port=0, device=0 gpio=noop())
device = max7219(serial, cascaded=n or 1, block_orientation,
rotate=rotate or 0, block_arranged_in_reverse_order=inverse)
print("Created device")
msg = "TYIT A"
show_message(device, msg, fill="red", font=proportional
(CP437_FONT), scroll_delay=0.1)
sleep(1)
while True:
 demo(1,00, False)
```

Pin Configuration

1. VCC power → Connect to 5V on Raspberry Pi
2. GND (Ground) → Connect to GND on Raspberry Pi
3. DIN (Data IO) → Connect to GPIO (Pin 19) for data transmission.

4. CLK (Clock) → connect to GPIO 11 (C Pin23) for clock Signal

5. CS (Chip select) * * (if using SPI) * * → connect to GPIO 8 (Pin24) for SPI control.