

**BHARTIYAVIDYABHAVAN’S**

**M.M.COLLEGEOFARTS,N.M.INSTITUTEOFSCIENCE**

**H.R.J.COLLEGEOFCOMMERCE BHAVAN’SCOLLEGE**

**MUNSHINAGAR,ANDHERIWEST,**

**MUMBAI-400058**

**CERTIFICATE**

This is to certify that Mr. Om Gadade Seat No.TYIT10 of T.Y.BSC Information Technology has satisfactorily completed the practical course in Internet of Things as prescribed by the University of Mumbai during the academic year 2023-2024.

Internal Examiner External Examiner

Signature Signature

Co-ordinator’s College Stamp   
 Signature

**Index**

|  |  |  |  |
| --- | --- | --- | --- |
| **Practical**  **No** | **Date** | **Practical Name** | **Sign** |
| **1** | **19/07/2023** | **Starting Raspbian OS, Familiarising with Raspberry Pi Components and interface, Connecting to ethernet, Monitor, USB** |  |
| **2** | **26/07/2023** | **Displaying different LED patterns with Raspberry Pi.** |  |
| **3** | **02/08/2023** | **Displaying time over 4-Digit 7-Segment display using Raspberry Pi.** |  |
| **4** | **09/08/2023** | **Controlling Raspberry Pi with Telegram**. |  |
| **5** | **06/09/2023** | **Implement Raspberry Pi based Oscilloscope** |  |
| **6** | **27/09/2023** | **Visitor Monitoring with Raspberry Pi and Pi Camera** |  |

**Practical 3 Date :- 02/08/2023**

**Aim:-Displaying time over 4-Digit 7-Segment display using Raspberry Pi.**

**Step 1: Creating connection.**

Connect seven segment module pins from GND to pin 14, VCC to pin 4, DIO to pin 18, CLK to pin 16.



**Step 2: create new folder.**

Open raspberry Pi operating system. Open file explorer.

Create a folder in the home directory inside that bhavans in that select Desktop. Right click and

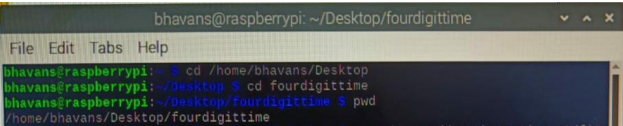
create a new folder. Give the folder name as fourdigittime.

**Step3:**

Open the terminal. Use cd command to change the directory give the following command.

# cd /home/bhavans/Desktop/fourdigittime

Use pwd command to print the current working directory.

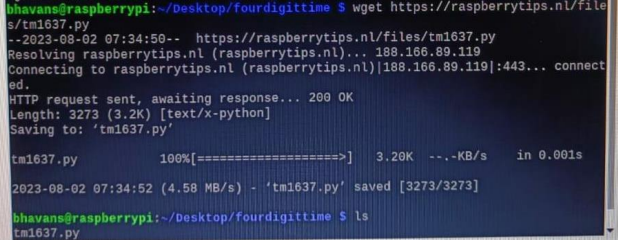


**Step 4: Download the script using wget command**

Use the following command and a new file get created inside the fourdigittime folder

tm1637.py.

# wget <https://raspberrytips.nl/files/tm1637.py>



**Step5: install the python library tm1637**

To install the library give the following command to install this library

# pip install raspberrypi.tm1367

**Step 6: create a new python file**

Create a new file clock.py in fourdigittime folder. Open this file with Thoppy and write the

following code.

**clock.py**

import sys

import time

import datetime

import RPi.GPIO as GPIO

import tm1637

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

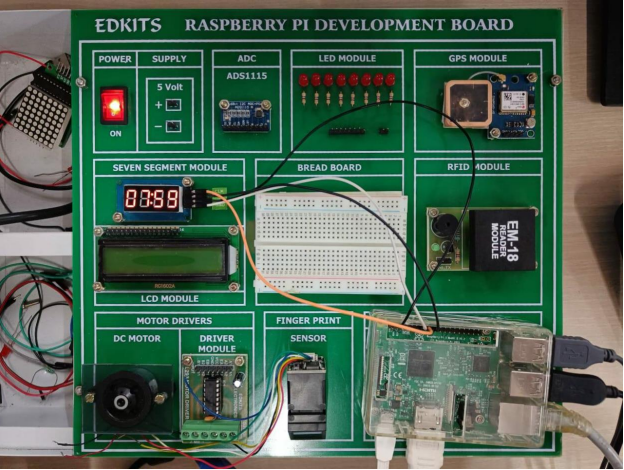
second-now.second

currenttime=[int (hour/10), hour %10, int (minute/10), minutes%10]

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

time.sleep(1)

**OUTPUT: -**



‘

**Practical 4 Date :- 09-08-2023**

**Aim: - Controlling Raspberry Pi with Telegram**.

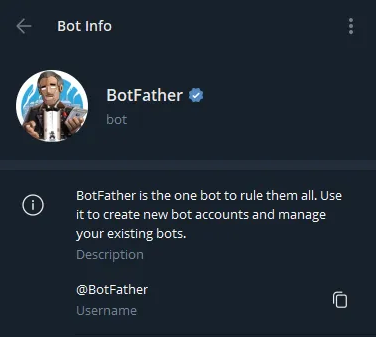
**Command: -**

Sudo apt-get install python-pip

Sudo pip install telepot

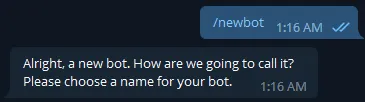
**Steps for creating Telegram bot using Bot Father: -**

1. Enter @Botfather in the search tab and Choose it (Official Telegram bots have a blue checkmark beside their name.)

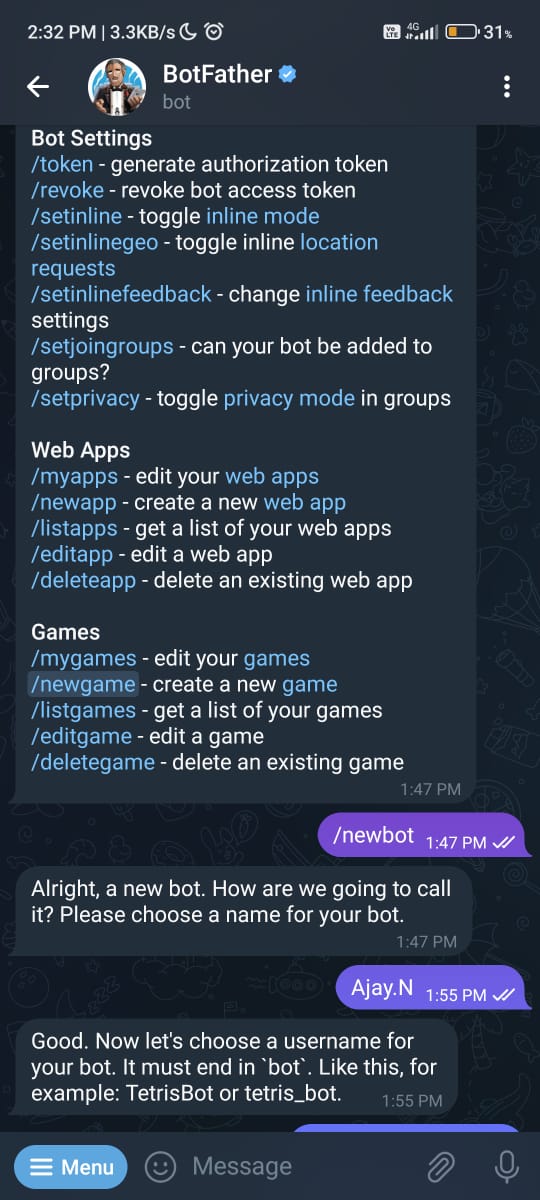


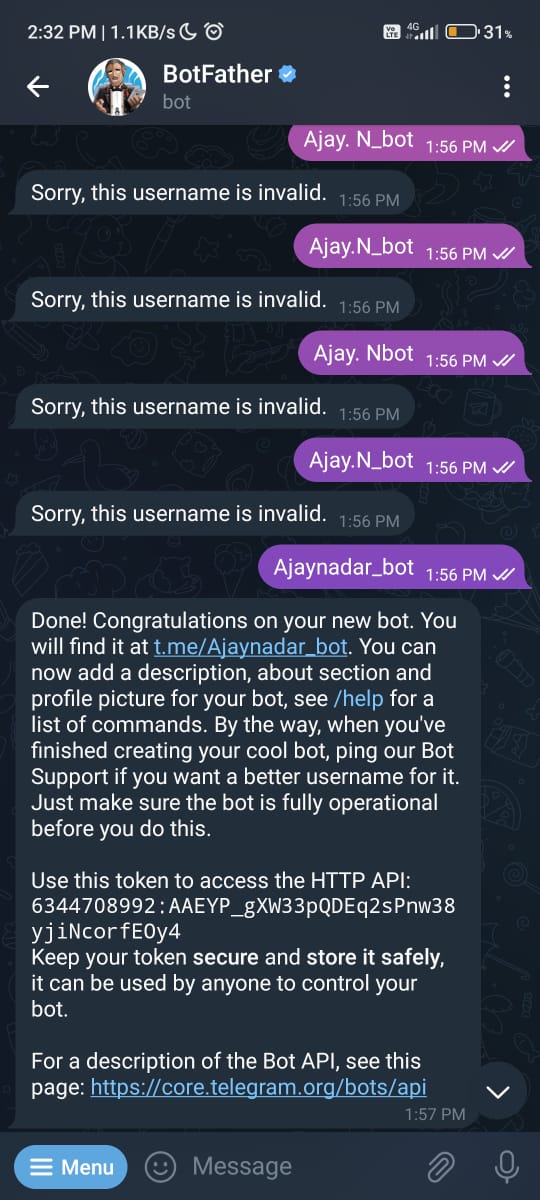
Click “Start” to activate BotFather bot

1. Choose or type the /newbot command and send it.



**Step 3.** Choose a name for your bot. And choose a username for your bot — the bot can be found by its username in searches. The username must be unique and end with the word “bot.”





 At this point, we create our Bot !

**Connection: -**

Connect LED module any one led to pin 11, and Ground to pin no 9.

**Code: -**

import sys

import time

import random

import datetime

import telepot

import RPi.GPIO as GPIO

#LED

def on(pin):

GPIO.output(pin,GPIO.HIGH)

return

def off(pin):

GPIO.output(pin,GPIO.LOW)

return

# to use Raspberry Pi board pin numbers

GPIO.setmode(GPIO.BOARD)

# set up GPIO output channel

GPIO.setup(11, GPIO.OUT)

def handle(msg):

chat\_id = msg['chat']['id']

command = msg['text']

print 'Got command: %s' % command

if command == 'on':

bot.sendMessage(chat\_id, on(11))

elif command =='off':

bot.sendMessage(chat\_id, off(11))

bot = telepot.Bot(‘6344708332:AAEYP\_Gwx33pQDEq2sPnw38yjiNcorfE0y4’)

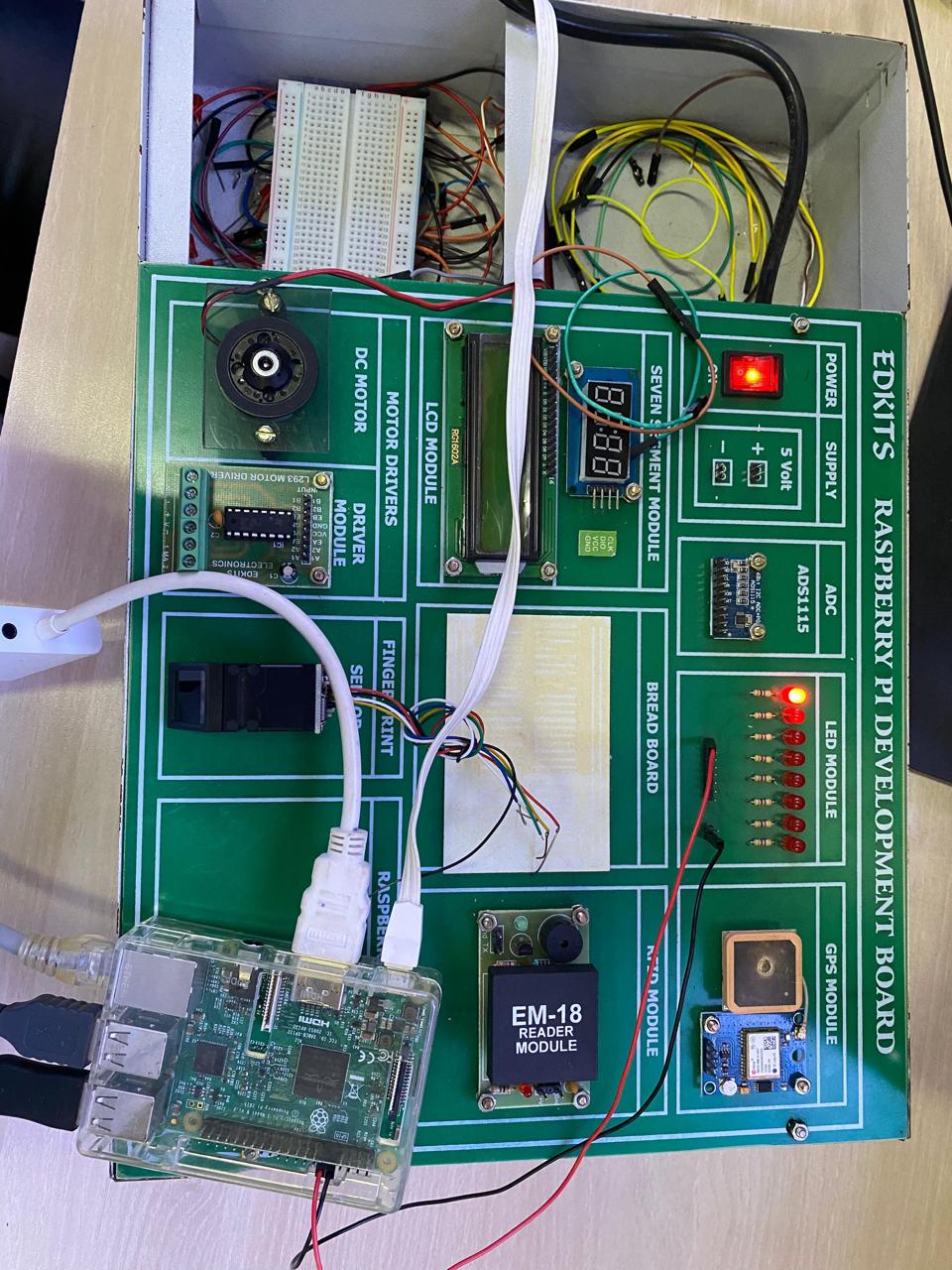
bot.message\_loop(handle)

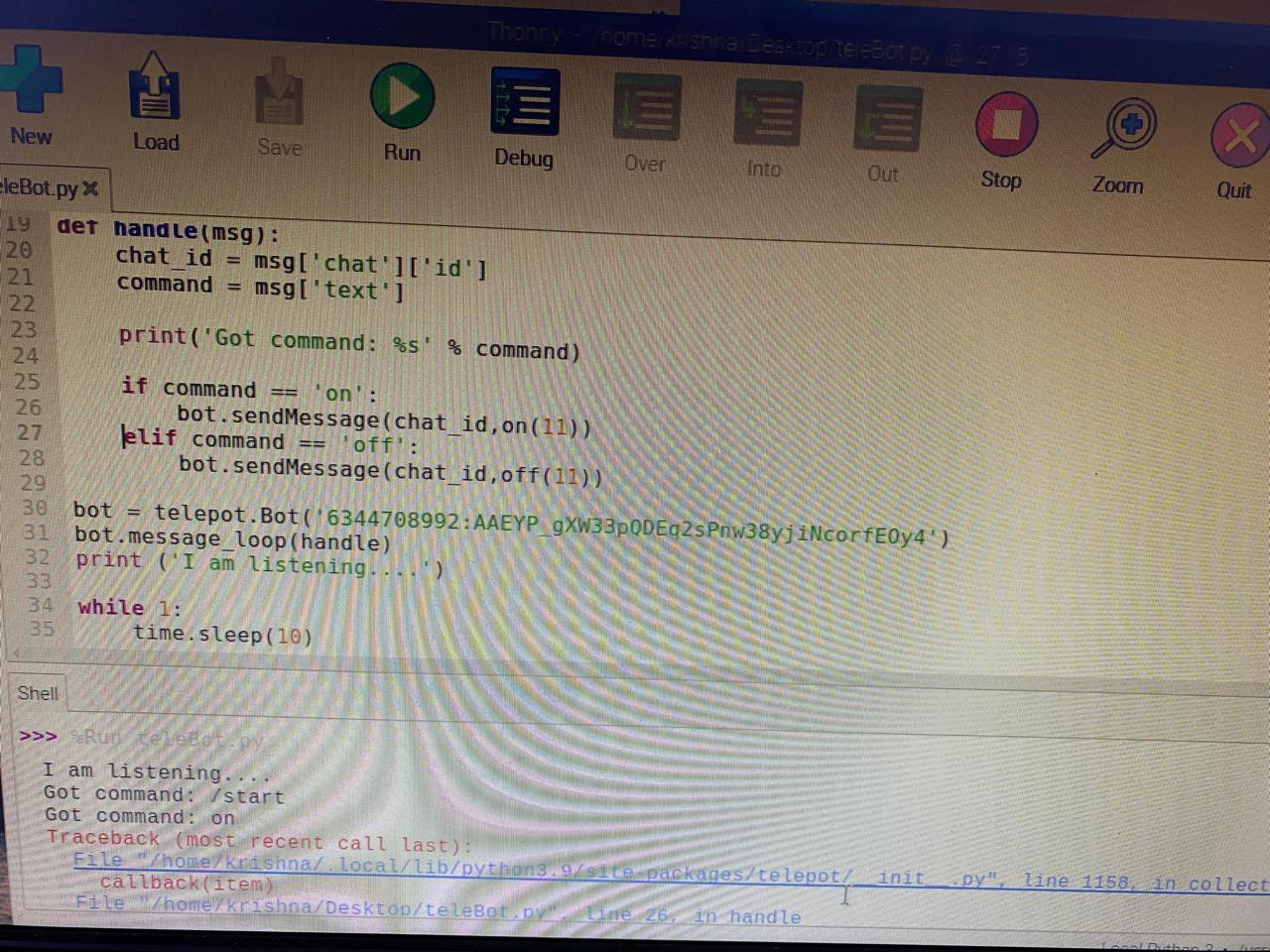
print 'I am listening...'

while 1:

time.sleep(10)

**Output: -**





**Practical 5 Date:-06/09/2023**

**Aim:** **Implement Raspberry Pi based Oscilloscope**

|  |  |  |
| --- | --- | --- |
| **ADS115 ADC** | **Pin Number** | **GPIO Number** |
| VDD | Pin 17 | 3.3v |
| GND | Pin 9 | GND |
| SCL | Pin 5 | GPIO 3 |
| SDA | Pin 3 | GPIO 2 |

**Step 1: Enable Raspberry Pi I2C interface**.

I2C is a serial communication protocol to enable simple low level communication between chips.

**sudo raspi-config** OR

**Go to Interfacing Options -> I2C->Enable(Yes)**

**Step 2: Update the Raspberry Pi**

**sudo apt-get update**

**sudo apt-get upgrade**

**Step 3: Install the Adafruit ADS115 library for ADC**

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

**sudo apt-get install build-essential python-dev python-smbus git**

In next line sudo apt install python3-smbus

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

**sudo git clone** [**https://github.com/adafruit/Adafruit\_Python\_ADS1x15.git**](https://github.com/adafruit/Adafruit_Python_ADS1x15.git)

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

**Cd /home/pi/Adafruit\_Python\_ADS1x15/**

**sudo python setup.py install OR**

**sudo python3 setup.py install.**

**Step 4: Test the library and I2C communication.**

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

$cd examples

$python simpletest.py

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

**Step 5: Install Matplotlib (in home directory)**

**sudo apt-get install python3-matplotlib**

**Step 6: Python Code for Raspberry Pi Oscilloscope**

import matplotlib.pyplot as plt

from matplotlib.animation import FuncAanimation

import Adafruit\_ADS1x15

**#Create an ADS1115 ADC (16-bit) instance.**

adc= Adafruit\_ADS1x15.ADS1115()

GAIN = 1

val = [ ]

**#Start continuous ADC conversions on channel 0 using the previous gain value**.

adc.start\_adc(0,gain=GAIN)

print(‘Reading ADS1x15 channel 0’)

fig,ax = plt.subplots()

ax.set\_ylim(-5000,5000)

ax.set\_title(‘Oscilloscope’)

ax.grid(True)

ax.set\_ylabel(‘ADCoutputs’)

line, = ax.plot([],’ro-’,label=’Channel0’)

ax.legend(loc=’lowerright’)

def update(cnt):

**#Read the last ADC conversion value and print it out.**

value = adc.get\_last\_result()

print(‘Channel 0:{0}’.format(value))

**#Set new data to line**

line.set\_data(list(range(len(val))),val)

ax.relim()

ax.autoscale\_view()

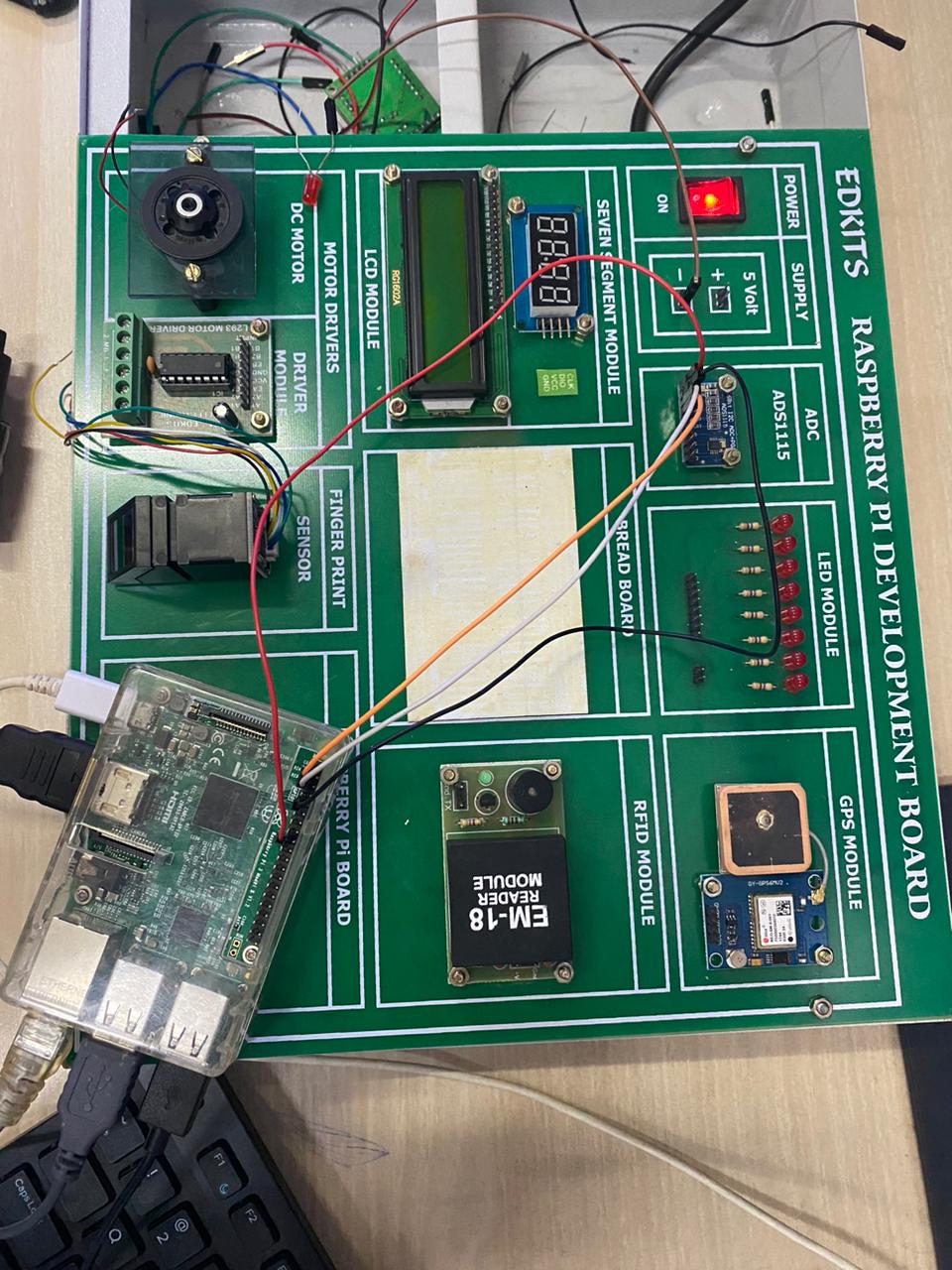
**#Store values for later**

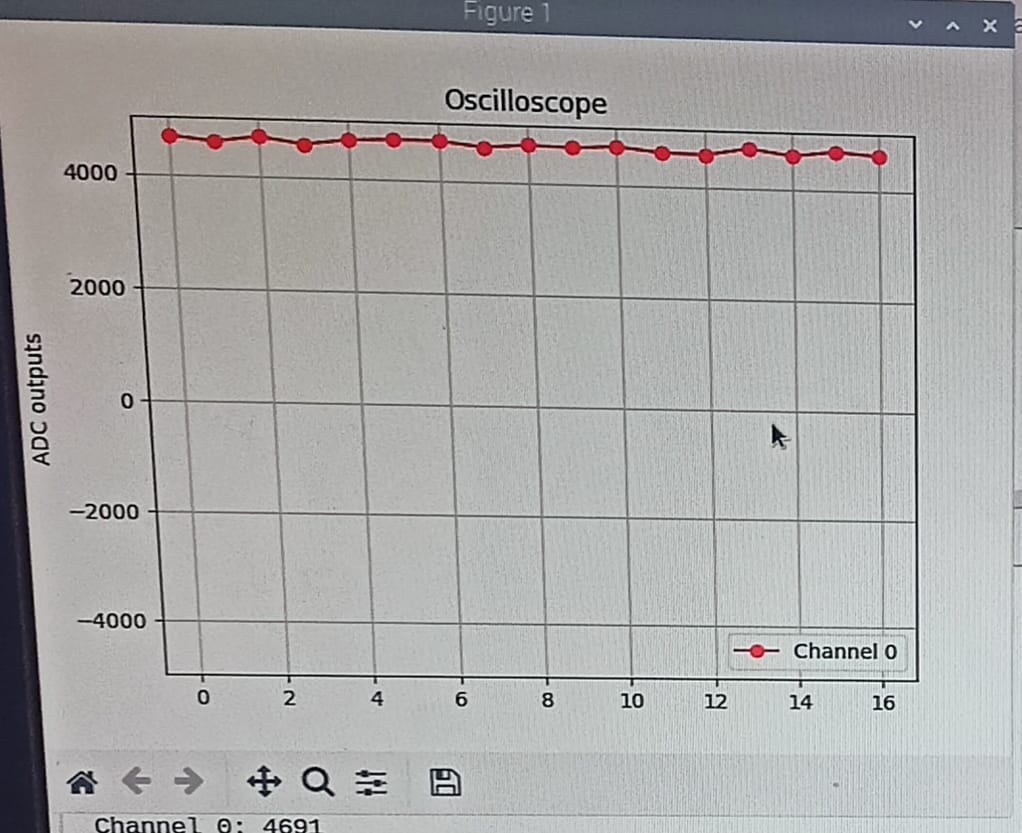
val.append(int(value))

if(cnt > 50):

val.pop(0)

**Output:**

****

****

**Practical 6 Date:-27/09/2023**

**Aim:- Visitor Monitoring with Raspberry Pi and Pi Camera**

**Connect Pi Camera to CS interface of Raspberry Pi board as shown below**

****

****

**How to enable camera functionality**

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

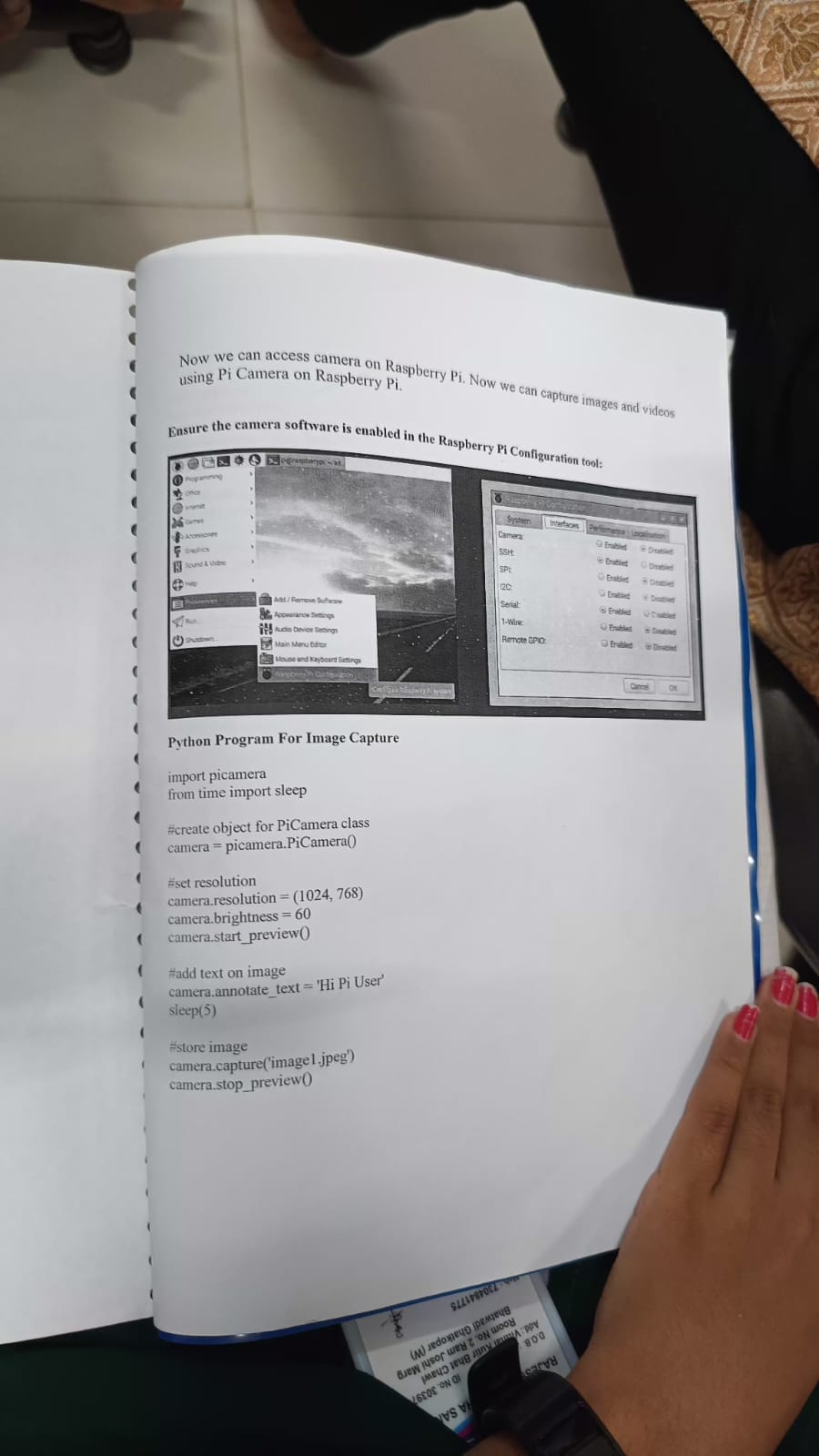
**Sudo raspi-config**

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:**

****

**Python program for Image Capture**

import picamera

from time import sleep

#create object for PiCamera class

camera = picamera.PiCamera()

#set resolution

camera.resolution = (1024,768)

camera.brightness = 60

camera.start\_preview()

#add text on image

camera.annotate\_text = ‘Hi Pi User’

Sleep(5)

#store image

camera.capture(‘image1.jpg’)

camera.stop\_preview()

**Python Program for Video Recording**

import picamera

from time import sleep

camera = picamera.PiCamera()

camera.resolution=(640,480)

print()

#start recording using pi camera c

amera.start\_recording(“/home/pi/demo.h264”)

#wait for video to record

camera.wait\_recording(20)

#stop recording

camera.stop\_recording()

camera.close()

print(“video recording stopped”)

**Output:**

