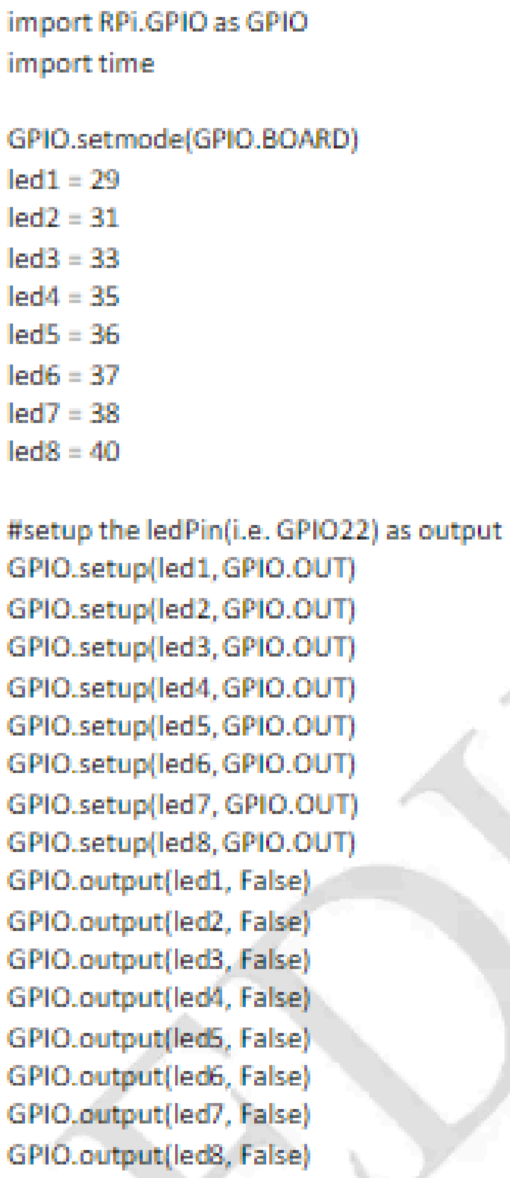
Practical No:1

Aim:
Displaying different LED patterns with Raspberry Pi.

Hardware Requirements:
1. LED
2. Resistor
3. Connecting wires
4. Breadboard

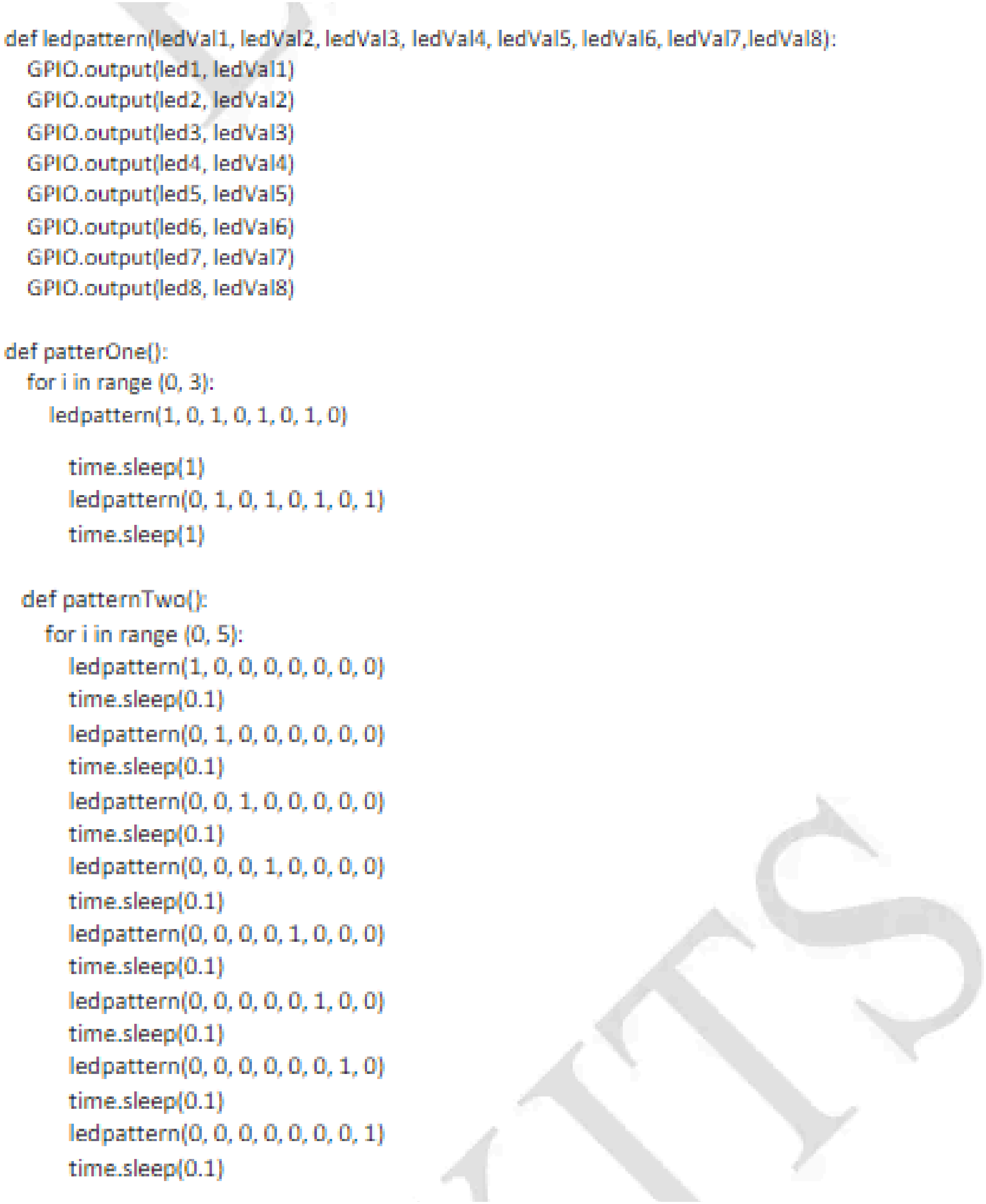
Code:



```
import RPi.GPIO as GPIO
import time

GPIO.setmode(GPIO.BOARD)
led1 = 29
led2 = 31
led3 = 33
led4 = 35
led5 = 36
led6 = 37
led7 = 38
led8 = 40

#setup the ledPin(i.e. GPIO22) as output
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)
GPIO.setup(led7, GPIO.OUT)
GPIO.setup(led8, GPIO.OUT)
GPIO.output(led1, False)
GPIO.output(led2, False)
GPIO.output(led3, False)
GPIO.output(led4, False)
GPIO.output(led5, False)
GPIO.output(led6, False)
GPIO.output(led7, False)
GPIO.output(led8, False)
```

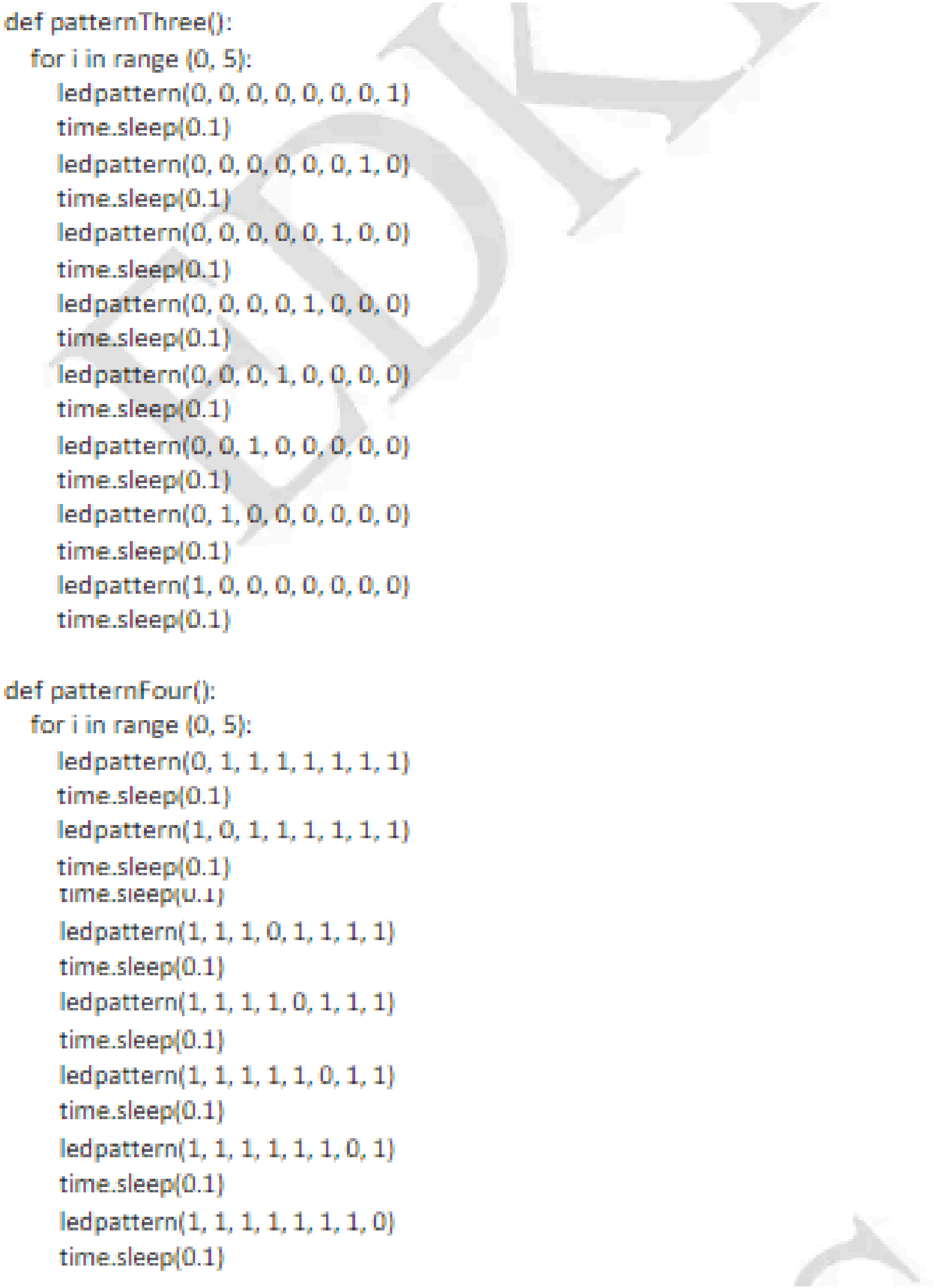


```
def ledpattern(ledVal1, ledVal2, ledVal3, ledVal4, ledVal5, ledVal6, ledVal7,ledVal8):
   GPIO.output(led1, ledVal1)
   GPIO.output(led2, ledVal2)
   GPIO.output(led3, ledVal3)
   GPIO.output(led4, ledVal4)
   GPIO.output(led5, ledVal5)
   GPIO.output(led6, ledVal6)
   GPIO.output(led7, ledVal7)
   GPIO.output(led8, ledVal8)

def patterOne():
   for i in range (0, 3):
      ledpattern(1, 0, 1, 0, 1, 0, 1, 0)

        time.sleep(1)
        ledpattern(0, 1, 0, 1, 0, 1, 0, 1)
        time.sleep(1)

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



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

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

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


try:
   while True:
      patterOne()
      patternTwo()
      patternThree()
      patternFour()
      patternFive()

finally:
   #reset the GPIO Pins
   GPIO.cleanup()
```

Practical No:2

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

Hardware Requirements:
1. TM1637 4-digit seven segment Display board
2. Connecting wires

Wiring up your Circuit: Hook up your circuit as follows:
1. Connect the Pin2 (5V) of Rpi to Vcc pin of Module
2. Connect Pin 6 (GND) of Rpi to GND of Module
3. Connect Pin38 (GPIO20) of Rpi to DIO of Module
4. Lastly connect Pin 40 (GPIO21) of Rpi to CLK of Module

Code:

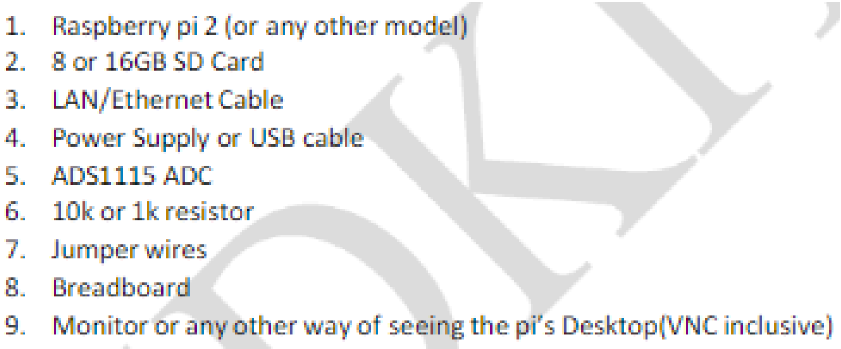
```
#Program to display Time on 4-digit Seven segment display
from time import sleep
import tm1637
try:
  import thread
except ImportError:
  import _thread as thread
# Initialize the clock (GND, VCC=3.3V, Example Pins are DIO-20 and CLK21)
Display = tm1637.TM1637(CLK=21, DIO=20, brightness=1.0)
try:
  print "Starting clock in the background (press CTRL + C to stop):"
  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:3

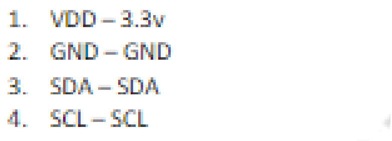
Aim:
Raspberry Pi Based Oscilloscope.

Hardware Requirements:



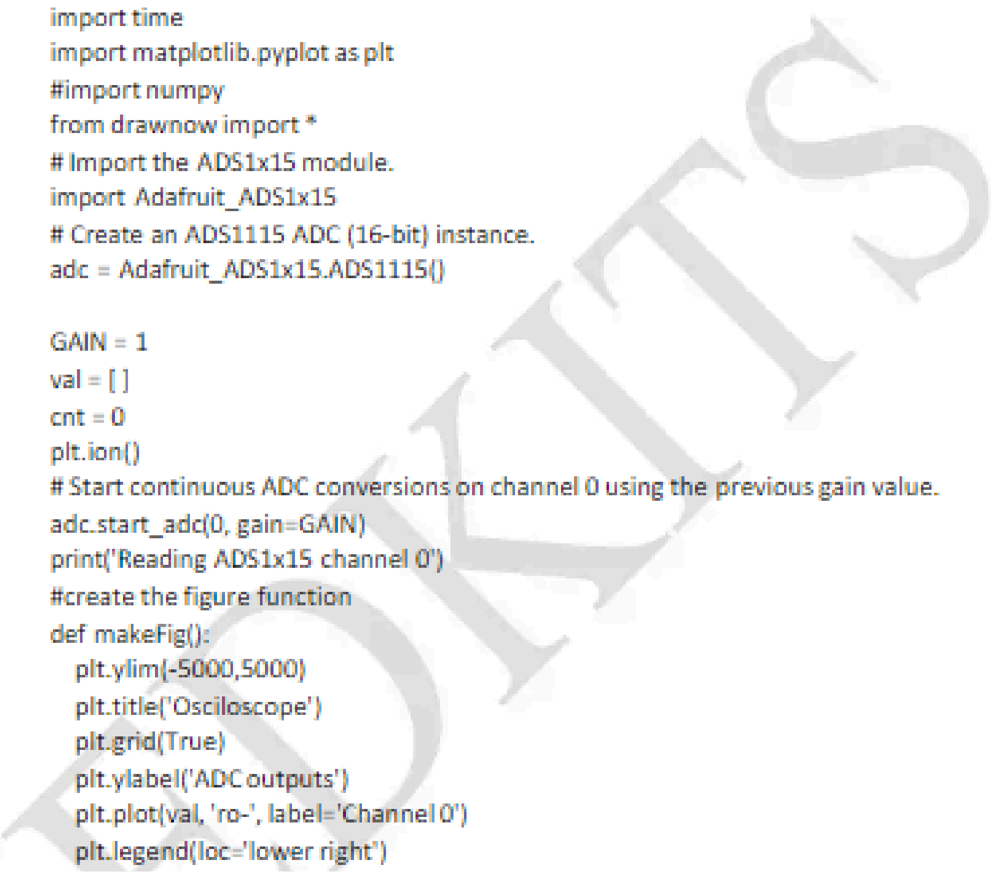
1. Raspberry pi 2 (or any other model)
2. 8 or 16GB SD Card
3. LAN/Ethernet Cable
4. Power Supply or USB cable
5. ADS1115 ADC
6. 10k or 1k resistor
7. Jumper wires
8. Breadboard
9. Monitor or any other way of seeing the pi's Desktop(VNC inclusive)

Wiring up your Circuit: Hook up your circuit as follows:



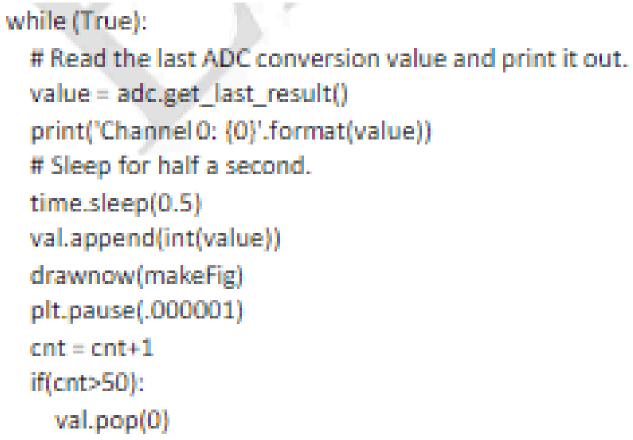
1. VDD – 3.3v
2. GND – GND
3. SDA – SDA
4. SCL – SCL

Code:



```
import time
import matplotlib.pyplot as plt
#import numpy
from drawnow import *
# Import the ADS1x15 module.
import Adafruit_ADS1x15
# 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)
```

Practical No:4

Aim:
Controlling Raspberry Pi with WhatsApp.

Step 1: Installation

Update the packages with

```
sudo apt-get update
sudo apt-get upgrade
```

Update firmware

```
sudo rpi-update
```

Prepare the system with the necessary components to Yowsup

```
sudo apt-get install python-dateutil
sudo apt-get install python-setuptools
sudo apt-get install python-dev
sudo apt-get install libevent-dev
sudo apt-get install ncurses-dev
```

Download the library with the command

```
git clone git://github.com/tgalal/yowsup.git
```

navigate to the folder

```
cd yowsup
```

and install the library with the command

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

Step 2: Registration

After installing the library we have to register the device to use WhatsApp. Yowsup comes with a cross platform command-line frontend called yowsup-cli. It provides you with the options of registration, and provides a few demos such as a command line client.
WhatsApp registration involves 2 steps. First you need to request a registration code.
And then you resume the registration with the code you got.
Request a code with command

```
python yowsup-cli registration --requestcode sms --phone 39xxxxxxxxxx --cc 39 --mcc 222 --mnc 10
```

Replace with your data ,
cc is your country code in this example 39 is for Italy,
mcc is Mobile Country Code
mnc is Mobile Network Code
You should receive on your phone a sms message with a code like xxx-xxx
Send a message to request registration with this command, (replace xxx-xxx with code you received)

```
python yowsup-cli registration --register xxx-xxx --phone 39xxxxxxxxxx --cc 39
```

Step 3: Utilization

Create a file to save your credentials

```
sudo nano /home/pi/yowsup/config
```

with this content

```
## Actual config starts below ##
cc=39 #if not specified it will be autodetected
phone=39xxxxxxxxxx
password=xxxxxxxxxxxxxxx=
```

Ok, we're ready for the test, Yowsup has a demo application in /home/pi/yowsup/yowsup/demos

Navigate to yowsup folder

```
cd /home/pi/yowsup
```

Start yowsup-cli demos with the command

```
yowsup-cli demos --yowsup --config config
```

You can see Yowsup prompt

If type "/help" you can see all available commands

First use the '/L' command for login; to send a message type

```
/message send 39xxxxxxxxxx "This is a message sent from Raspberry Pi"
```

replace xxx with the recipient number

If you respond with a message it will be displayed on Raspberry.

Practical No:5

Aim:
Raspberry Pi GPS Module Interfacing

Hardware Requirements:
1.GPS module
2. USB to TTL converter
3. Connecting wire

Wiring up your Circuit:
1. Connect the VCC Pin of GPS Module to 3.3V Pin of USB to TTL converter
2. Connect the GND Pin of GPS Module to GND Pin of USB to TTL converter
3. Connect the Tx Pin of GPS Module to Rx Pin of USB to TTL converter
4. Connect the Rx Pin of GPS Module to the Tx Pin of USB to TTL converter.
5. Lastly connect the USB to TTL converter to USB port of Raspberry Pi

Code:

```
import serial            #import serial package
from time import sleep
import webbrowser        #import package for opening link in browser
import sys               #import system package
def GPS_Info():
  global NMEA_buff
  global lat_in_degrees
  global long_in_degrees
  nmea_time = []
  nmea_latitude = []
  nmea_longitude = []
  nmea_time = NMEA_buff[0]                #extract time from GPGGA string
  nmea_latitude = NMEA_buff[1]              #extract latitude from GPGGA string
  nmea_longitude = NMEA_buff[3]             #extract longitude from GPGGA string
  print("NMEA Time: ", nmea_time,'\n')
 print ("NMEA Latitude:", nmea_latitude,"NMEA Longitude:", nmea_longitude,'\n')
  try:
        lat = float(nmea_latitude)                #convert string into float for calculation
        longi = float(nmea_longitude)              #convertr string into float for calculation
  Except ValueError:
        print(“Error converting latitude and longitude”)
        return
lat_in_degrees = convert_to_degrees(lat)    #get latitude in degree decimal format
  long_in_degrees = convert_to_degrees(longi) #get longitude in degree decimal format
#convert raw NMEA string into degree decimal format
def convert_to_degrees(raw_value):
  decimal_value = raw_value/100.00
  degrees = int(decimal_value)
```

```
    mm_mmmm = (decimal_value - int(decimal_value))/0.6
    position = degrees + mm_mmmm
    position = "%.4f" %(position)
    return position

gpgga_info = "$GPGGA,"
ser = serial.Serial ("/dev/ttyUSB0")              #Open port with baud rate
GPGGA_buffer = 0
NMEA_buff = 0
lat_in_degrees = 0
long_in_degrees = 0
try:
    while True:
        received_data = (str)(ser.readline())                  #read NMEA string received
        GPGGA_data_available = received_data.find(gpgga_info)   #check for NMEA GPGGA
string
        if (GPGGA_data_available>0):
            GPGGA_buffer = received_data.split("$GPGGA,",1)[1]  #store data coming after
"$GPGGA,"
            NMEA_buff = (GPGGA_buffer.split(','))               #store comma separated data in buffer
            GPS_Info()                                    #get time, latitude, longitude
             print("lat in degrees:", lat_in_degrees," long in degree: ", long_in_degrees, '\n')
            map_link = 'http://maps.google.com/?q=' + lat_in_degrees + ',' + long_in_degrees
#create link to plot location on Google map
            print("<<<<<<<<press ctrl+c to plot location on google maps>>>>>>\n")
#press ctrl+c to plot on map and exit
            print("---------------------------------------------------------\n")

except KeyboardInterrupt:
    webbrowser.open(map_link)        #open current position information in google map
    sys.exit(0)
#end of file
```

Practical No:6

Aim:
Interfacing Pi Camera with Raspberry Pi.

Hardware Requirements:

1. Camera Module
2. Connecting wires

Code:

```
#Camera Program
# import time and picamera library
from time import sleep
from picamera import PiCamera
camera = PiCamera()
camera.resolution = (1280, 720)      # selecting resolution 1280x720 px
camera.start_preview()
# Camera warm-up time
sleep(2)
camera.capture('/home/pi/Pictures/newImage.jpg') #capture and save image at specified location
camera.stop_preview()
#end of code
```

Practical No:7

Aim:

Interfacing Raspberry Pi with RFID.

Hardware Requirements:

1. Raspberry Pi 3 Model B
2. EM-18 RFID Reader Module
3. RS232 – to – USB Adapter
4. Few RFID Cards or RFID Tag
5. Power Supply for RFID Reader
6. 5V Supply for Raspberry Pi and RS232-USB Adapter
7. Connecting Wires
8. 680Ω Resistor (1/4 Watt)
9. 1.5KΩ Resistor (1/4 Watt)

Code:

For writing data into card:

```
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()#!
```

For reading data from card:

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

Practical No:8

Aim:

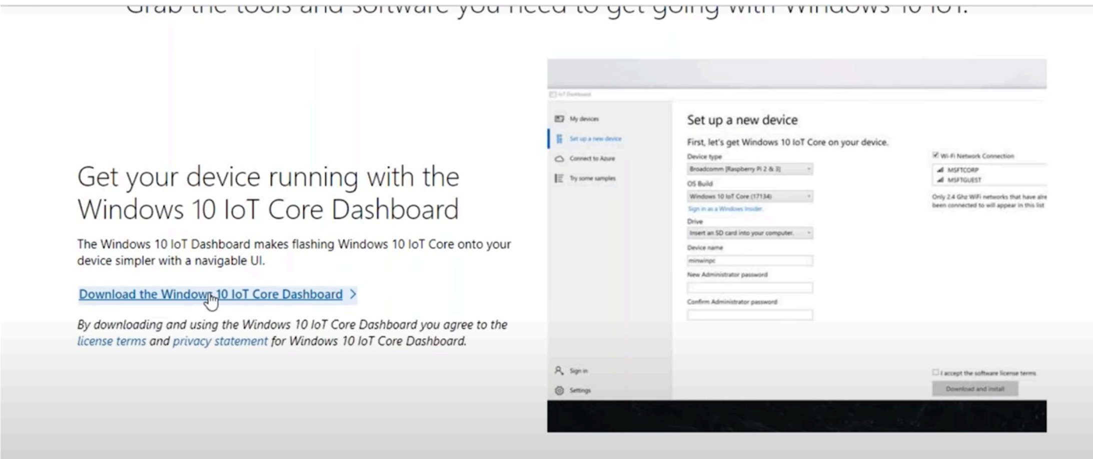
Installing Windows 10 IoT Core on Raspberry Pi .

Hardware Requirements:

1. Raspberry Pi 3.
2. 5V 2A microUSB power supply.
3. 8GB or larger Class 10 microSD card with full-size SD adapter.
4. HDMI cable.
5. Access to a PC.
6. USB WiFi adapter (older models of Raspberry Pi) or Ethernet cable.

Steps for installing windows:

1. Go to the Windows 10 developer center.
2. Click Get Windows 10 IoT Core Dashboard to download the necessary application.



Grab the tools and software you need to get going with Windows 10 IoT.

Get your device running with the Windows 10 IoT Core Dashboard

The Windows 10 IoT Dashboard makes flashing Windows 10 IoT Core onto your device simpler with a navigable UI.

Download the Windows 10 IoT Core Dashboard >

By downloading and using the Windows 10 IoT Core Dashboard you agree to the license terms and privacy statement for Windows 10 IoT Core Dashboard.

Set up a new device

First, let's get Windows 10 IoT Core on your device.

Device type: Broadcomm [Raspberry Pi 2 & 3]

OS Build: Windows 10 IoT Core (17134)

Sign in as a Windows Insider.

Drive: Insert an SD card into your computer.

Device name: minwinpc

New Administrator password

Confirm Administrator password

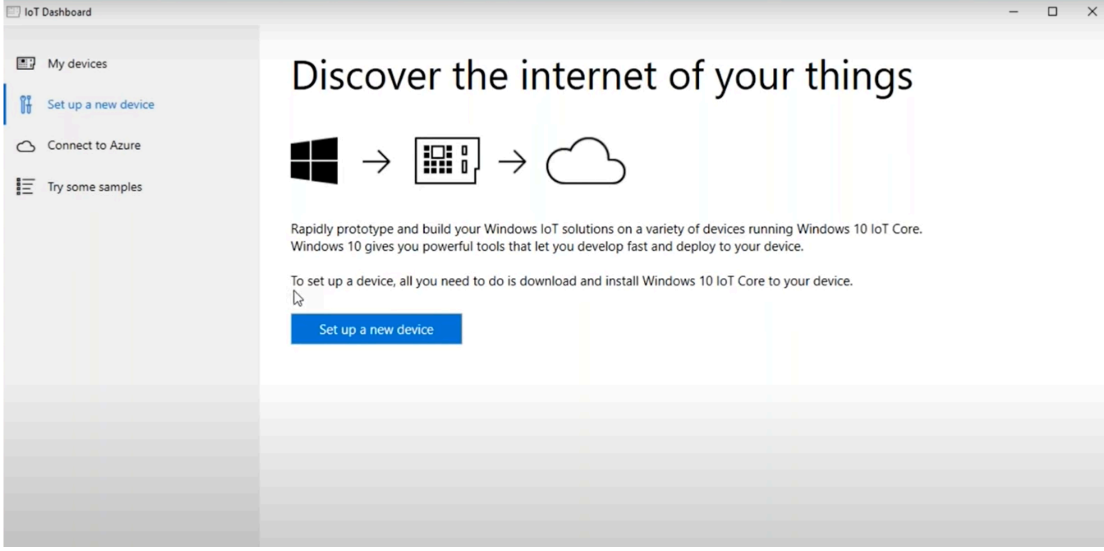
☑ Wi-Fi Network Connection

MSFTCORP

MSFTGUEST

Only 2.4 Ghz WiFi networks that have alre been connected to will appear in this list

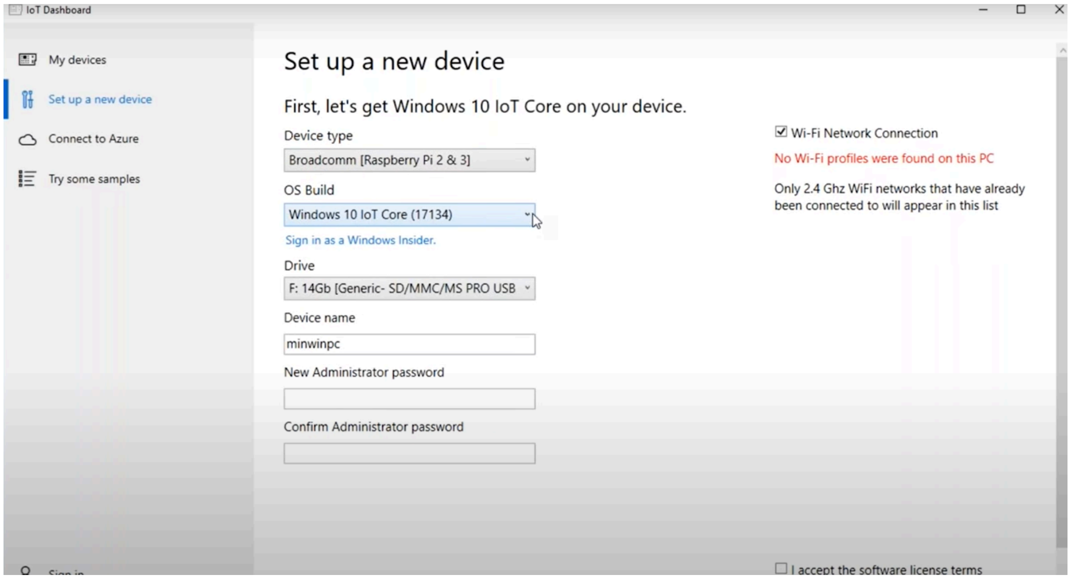
☐ I accept the software license terms



Discover the internet of your things

Rapidly prototype and build your Windows IoT solutions on a variety of devices running Windows 10 IoT Core. Windows 10 gives you powerful tools that let you develop fast and deploy to your device.

To set up a device, all you need to do is download and install Windows 10 IoT Core to your device.



Set up a new device

First, let's get Windows 10 IoT Core on your device.

Device type: Broadcomm [Raspberry Pi 2 & 3]

OS Build: Windows 10 IoT Core (17134)

Sign in as a Windows Insider.

Drive: F: 14Gb [Generic- SD/MMC/MS PRO USB

Device name: minwinpc

New Administrator password:

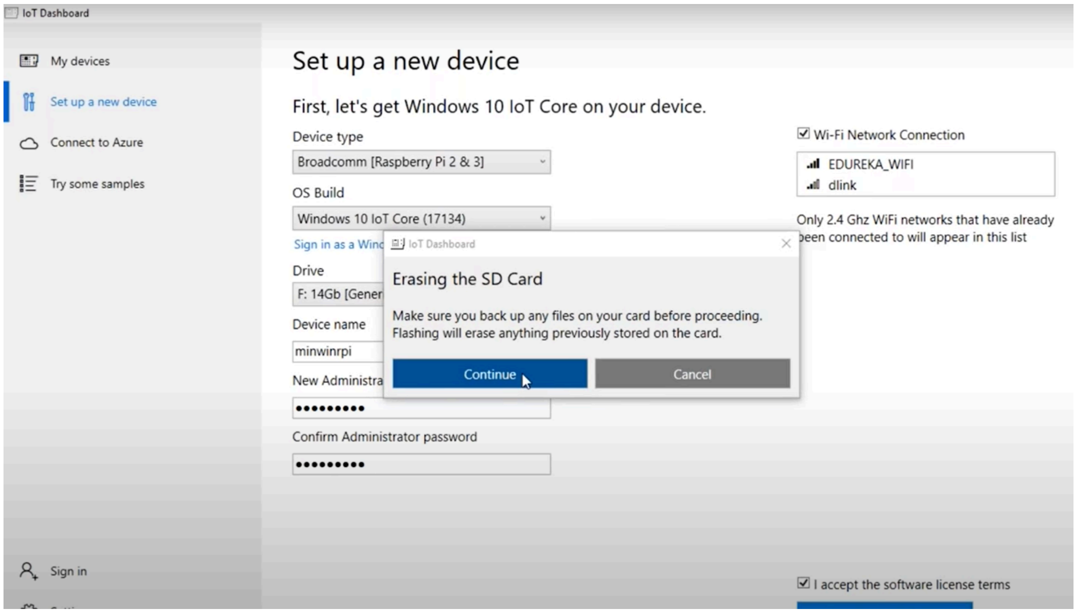
Confirm Administrator password:

☑ Wi-Fi Network Connection

No Wi-Fi profiles were found on this PC

Only 2.4 Ghz WiFi networks that have already been connected to will appear in this list

☐ I accept the software license terms



Set up a new device

First, let's get Windows 10 IoT Core on your device.

Device type: Broadcomm [Raspberry Pi 2 & 3]

OS Build: Windows 10 IoT Core (17134)

Sign in as a Winc

Drive: F: 14Gb [Gener

Device name: minwinrpi

New Administra: ●●●●●●●●●

Confirm Administrator password: ●●●●●●●●●

☑ Wi-Fi Network Connection

- EDUREKA_WIFI
- dlink

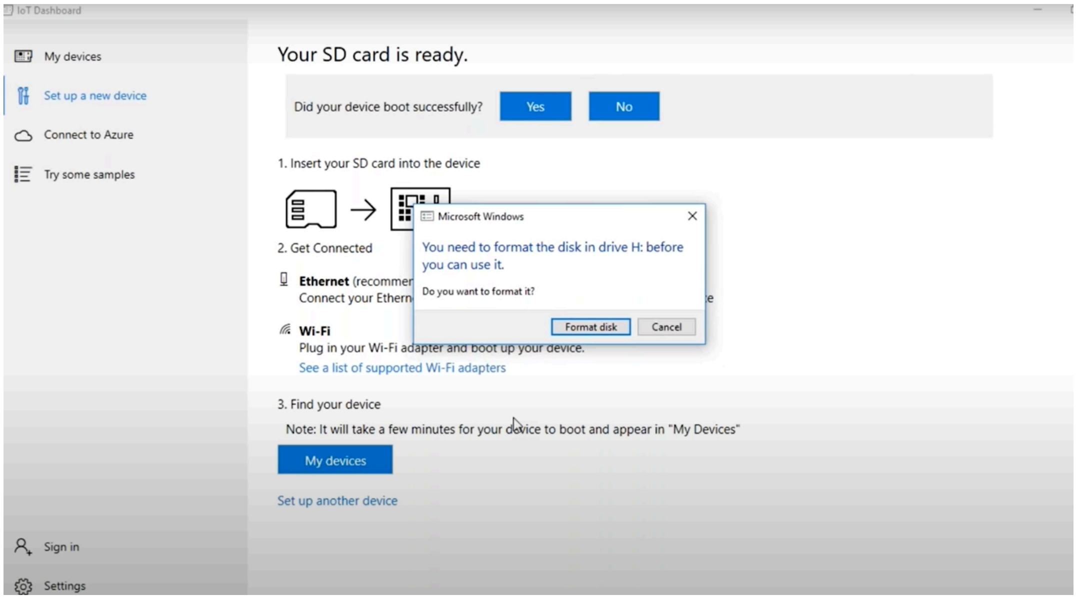
Only 2.4 Ghz WiFi networks that have already been connected to will appear in this list

IoT Dashboard

Erasing the SD Card

Make sure you back up any files on your card before proceeding. Flashing will erase anything previously stored on the card.

☑ I accept the software license terms



Your SD card is ready.

Did your device boot successfully? Yes No

1. Insert your SD card into the device

2. Get Connected

Ethernet (recommen

Connect your Ethern e

Wi-Fi

Plug in your Wi-Fi adapter and boot up your device.

See a list of supported Wi-Fi adapters

3. Find your device

Note: It will take a few minutes for your device to boot and appear in "My Devices"

My devices

Set up another device

Microsoft Windows

You need to format the disk in drive H: before you can use it.

Do you want to format it?

Format disk Cancel