ZEAL EDUCATION SOCIETY'S

ZEAL COLLEGE OF ENGINEEIRNG AND RESEARCH,
NARHE, PUNE

DEPARTMENT OFINFORMATION TECHNOLOGY
ENGINEERING
SEMESTER-II

[A.Y.: 2023 - 2024]

Internet of Things
(314448D)

LABORATORY MANUAL

Institute and Department Vision and Mission

INSTITUTE VISION	To impart value added technological education through pursuit of academic excellence, research and entrepreneurial attitude.
INSTITUTE MISSION	**M1:** To achieve academic excellence through innovative teaching and learning process. **M2:** To imbibe the research culture for addressing industry and societal needs**.** **M3:** To provide conducive environment for building the entrepreneurial skills. **M4:** To produce competent and socially responsible professionals with core human values.

DEPARTMENT VISION	To emerge as a department of repute in Computer Engineering which produces competent professionals and entrepreneurs to lead technical and betterment of mankind.
DEPARTMENT MISSION	**M1:** To strengthen the theoretical and practical aspects of the learning process by teaching applications and hands on practices using modern tools and FOSS technologies. **M2:** To endavor innovative interdisciplinary research and entrepreneurship skills to serve the needs of Industry and Society. **M3:** To enhance industry academia dialog enabling students to inculcate professional skills. **M4:** To incorporate social and ethical awareness among the students to make them conscientious professionals.

Department
Program Educational Objectives(PEOs)

PEO1:	To Impart fundamentals in science, mathematics and engineering to cater the needs of society and Industries.
PEO2:	Encourage graduates to involve in research, higher studies, and/or to become entrepreneurs.
PEO3:	To Work effectively as individuals and as team members in a multidisciplinary environment with high ethical values for the benefit of society.

Savitribai Phule Pune University

Third Year of Information Technology Engineering (2019 Course)
314456: Computer Network & Security

Teaching Scheme:	Credit	Examination Scheme:
PR: 04 Hours/Week	02	TW: 25 Marks OR: 25 Marks

Prerequisite Courses:

• Programming Skill Development Lab

Course Objectives:

1. To learn interfacing of sensor and actuators using Arduino Uno/Raspberry Pi.
2. To learn and understand IoT platforms and its significance for real time applications
3. To learn and understand the steps involved in python programming for IoT applications

Course Outcomes:

On completion of the course, student will be able to-

CO 1 Design and implement realtime applications with sensors and actuators.

CO 2 Design and develop real time IoT based application by cloud interfacing.

List of Assignments

TITLE
Group A
1. Design and implement IoT system using Arduino Uno/ Raspberry Pi using 'Ultrasonic sensor and Servo motor' such as 'Door opener in home automation'. **2. Design and implement parameter monitoring IoT system keeping records on Cloud such as 'environment humidity and temperature monitoring'.** **3. Design and implement real time monitoring system using android phone (Blynk App.) such as 'soil parameter monitoring'.** **4. Design and implement IoT system for one of the applications like: Traffic Application, Medical/Health application, Social Application etc.**

IoT Practical 1

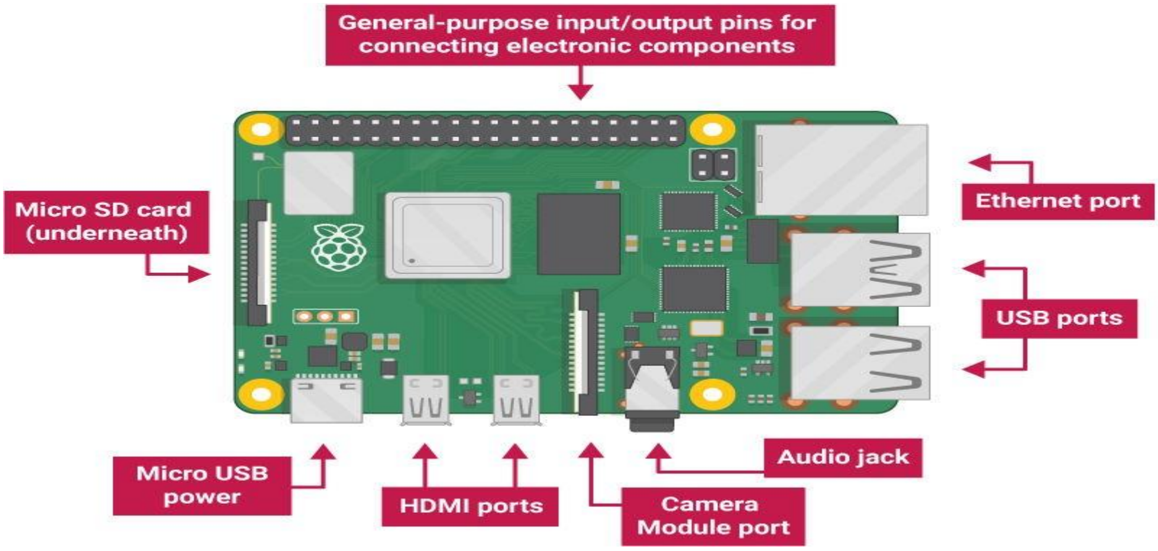
Aim : Design and implement IoT system using Arduino Uno/ Raspberry Pi using 'Ultrasonic sensor and Servo motor' such as 'Door opener in home automation'

Objective : To connect and implement Raspberry Pi, Ultrasonic Sensor using Servo Motor used as a Door Opener System.

Theory :
Raspberry Pi:-

You are going to take a first look at Raspberry Pi! You should have a Raspberry Pi computerin front of you for this. The computer shouldn't be connected to anything yet.

- Look at your Raspberry Pi. Can you find all the things labelled on the diagram?



- USB ports — these are used to connect a mouse and keyboard. You can also connectother components, such as a USB drive.
- SD card slot — you can slot the SD card in here. This is where the operating systemsoftware and your files are stored.
- Ethernet port — this is used to connect Raspberry Pi to a network with a cable.Raspberry Pi can also connect to a network via wireless LAN.
- Audio jack — you can connect headphones or speakers here.

- HDMI port — this is where you connect the monitor (or projector) that you are usingto display the output from the Raspberry Pi. If your monitor has speakers, you can alsouse them to hear sound.
- Micro USB power connector — this is where you connect a power supply. You shouldalways do this last, after you have connected all your other components.
- GPIO ports — these allow you to connect electronic components such as LEDs and buttons to Raspberry Pi.

Set up your SD card

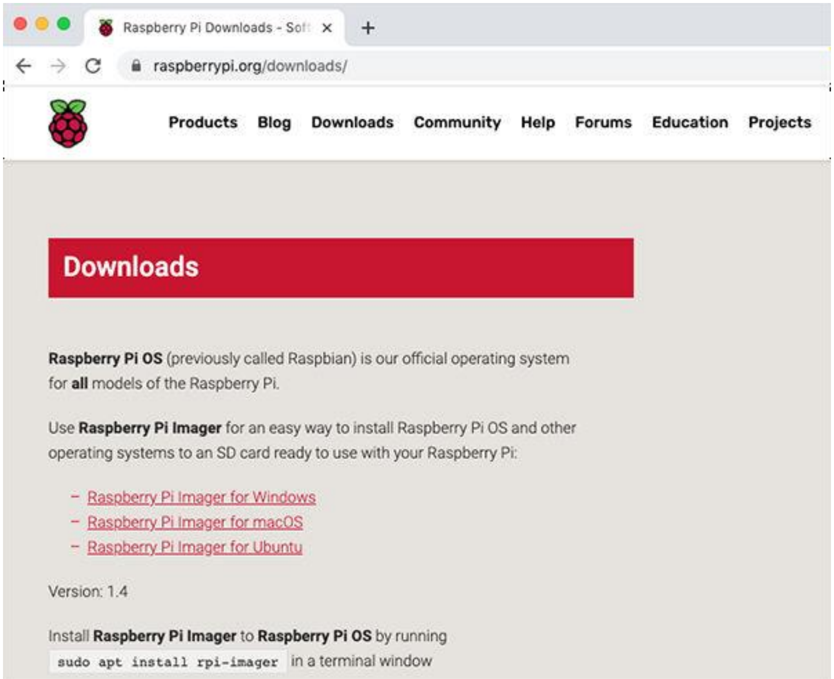
If you have an SD card that doesn't have the Raspberry Pi OS operating system on it yet, or ifyou want to reset your Raspberry Pi, you can easily install Raspberry Pi OS yourself. To do so, you need a computer that has an SD card port — most laptop and desktop computers haveone.

The Raspberry Pi OS operating system via the Raspberry Pi Imager

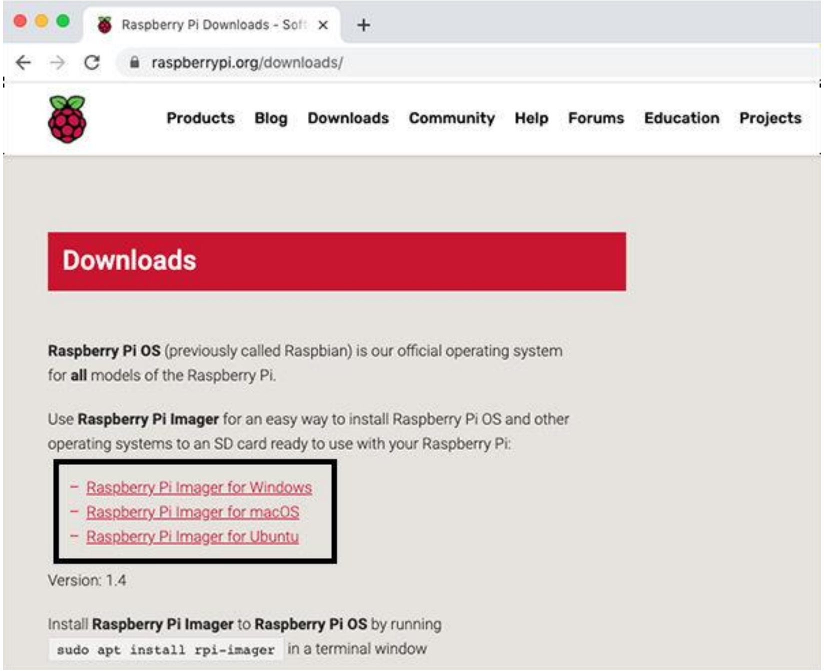
Using the Raspberry Pi Imager is the easiest way to install Raspberry Pi OS on your SD card. Note: More advanced users looking to install a particular operating system should use this guide to installing operating system images.

Download and launch the Raspberry Pi Imager

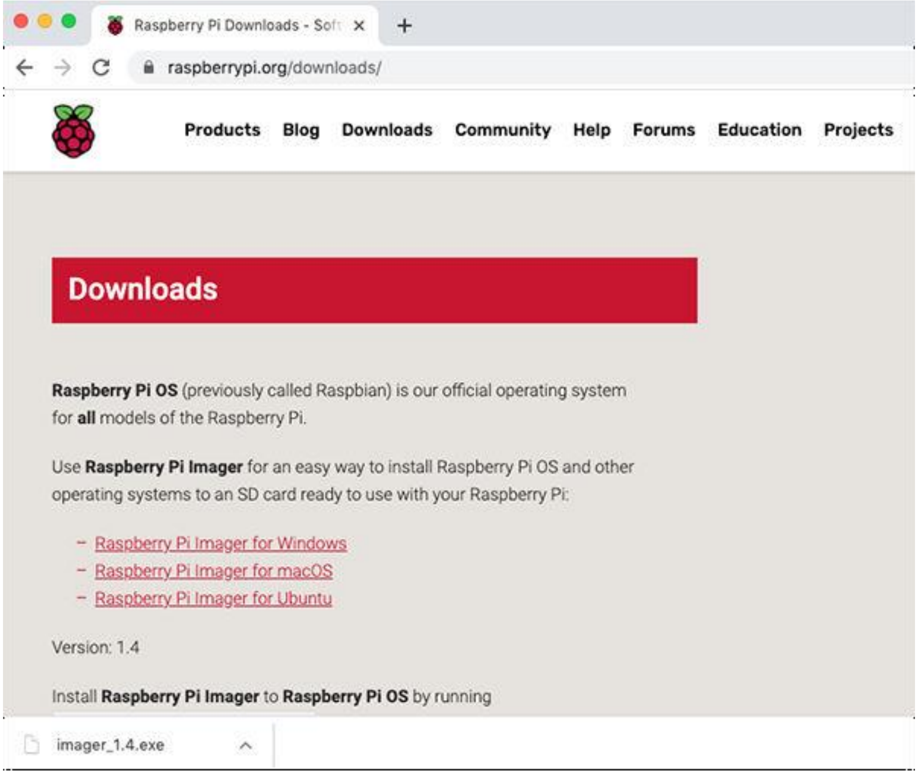
- Visit the Raspberry Pi downloads page



- Click on the link for the Raspberry Pi Imager that matches your operating system



- When the download finishes, click it to launch the installer



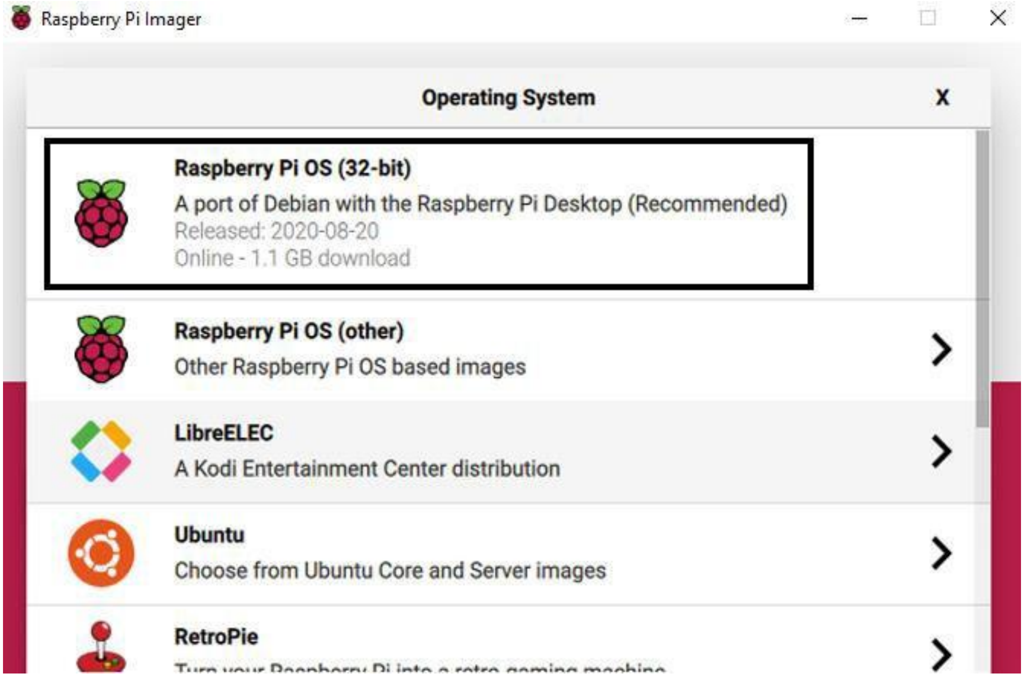
Using the Raspberry Pi Imager

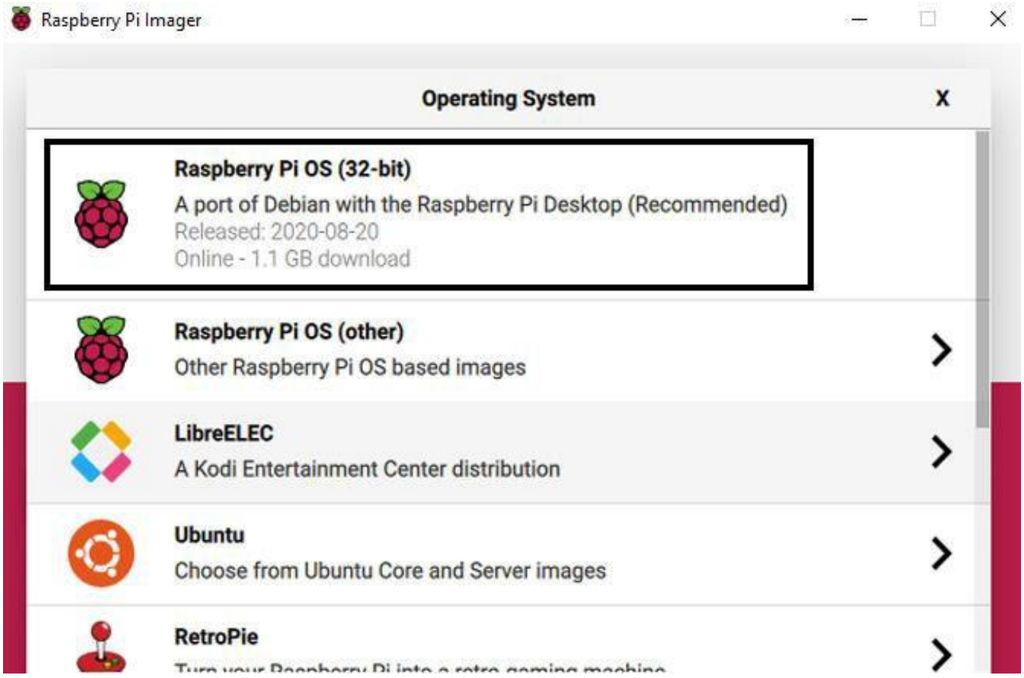
Anything that's stored on the SD card will be overwritten during formatting. If your SD card currently has any files on it, e.g. from an older version of Raspberry Pi OS, you may wish to back up these files first to prevent you from permanently losing them.

When you launch the installer, your operating system may try to block you from running it. For example, on Windows I receive the following message:

- If this pops up, click on More info and then Run anyway
- Follow the instructions to install and run the Raspberry Pi Imager
- Insert your SD card into the computer or laptop SD card slot
- In the Raspberry Pi Imager, select the OS that you want to install and the SD card youwould like to install it on

Note: You will need to be connected to the internet the first time for the the Raspberry Pi Imager to download the OS that you choose. That OS will then be stored for future offline use. Being online for later uses means that the Raspberry Pi imager will always give you the latest version.


Raspberry Pi Imager
Operating System
X
Raspberry Pi OS (32-bit)
A port of Debian with the Raspberry Pi Desktop (Recommended)
Released: 2020-08-20
Online - 1.1 GB download
Raspberry Pi OS (other)
Other Raspberry Pi OS based images
LibreELEC
A Kodi Entertainment Center distribution
Ubuntu
Choose from Ubuntu Core and Server images
RetroPie


Raspberry Pi Imager
Operating System
X
Raspberry Pi OS (32-bit)
A port of Debian with the Raspberry Pi Desktop (Recommended)
Released: 2020-08-20
Online - 1.1 GB download
Raspberry Pi OS (other)
Other Raspberry Pi OS based images
LibreELEC
A Kodi Entertainment Center distribution
Ubuntu
Choose from Ubuntu Core and Server images
RetroPie

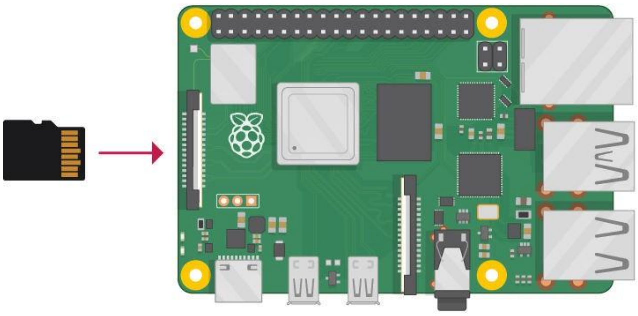
- Then simply click the WRITE button
- Wait for the Raspberry Pi Imager to finish writing
- Once you get the following message, you can eject your SD card

Connect your Raspberry Pi

Let’s connect up your Raspberry Pi and get it running.

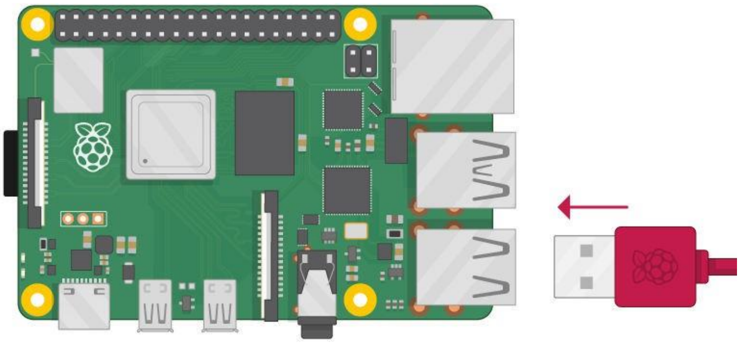
- Check the slot on the underside of your Raspberry Pi to see whether an SD card is inside.If no SD card is there, then insert an SD card with Raspbian installed (via

NOOBS).

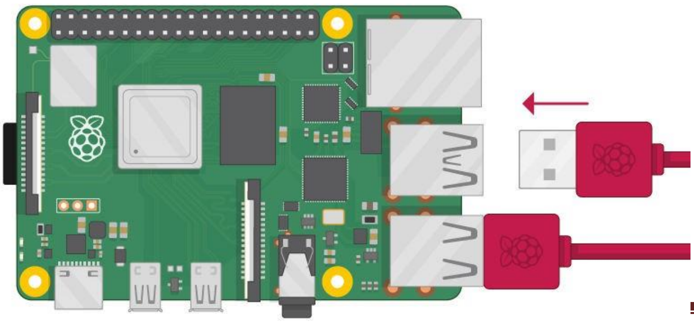


Note: Many microSD cards come inside a larger adapter — you can slide the smaller card outusing the lip at the bottom.

- Find the USB connector end of your mouse’s cable, and connect the mouse to a USB porton your Raspberry Pi (it doesn’t matter which port you use).



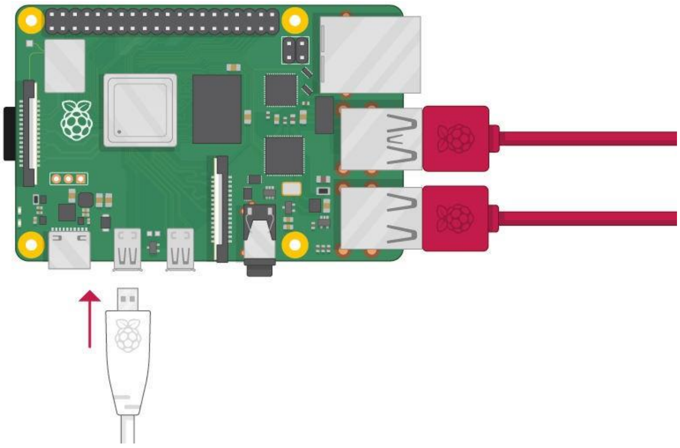
- Connect the keyboard in the same way.



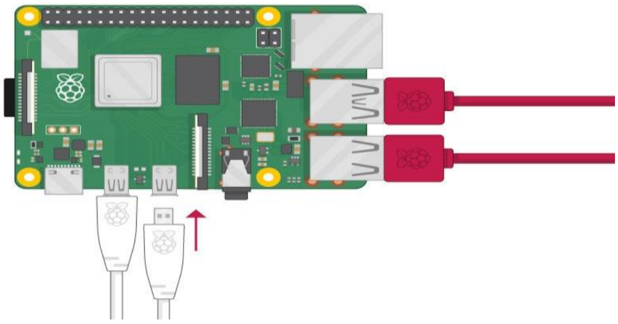
- Make sure your screen is plugged into a wall socket and switched on.
- Look at the HDMI port(s) on your Raspberry Pi — notice that they have a flat side on top.
- Use a cable to connect the screen to the Raspberry Pi's HDMI port — use an adapter if necessary.

  Raspberry Pi 4

  Connect your screen to the first of Raspberry Pi 4's HDMI ports, labelled HDMI0.

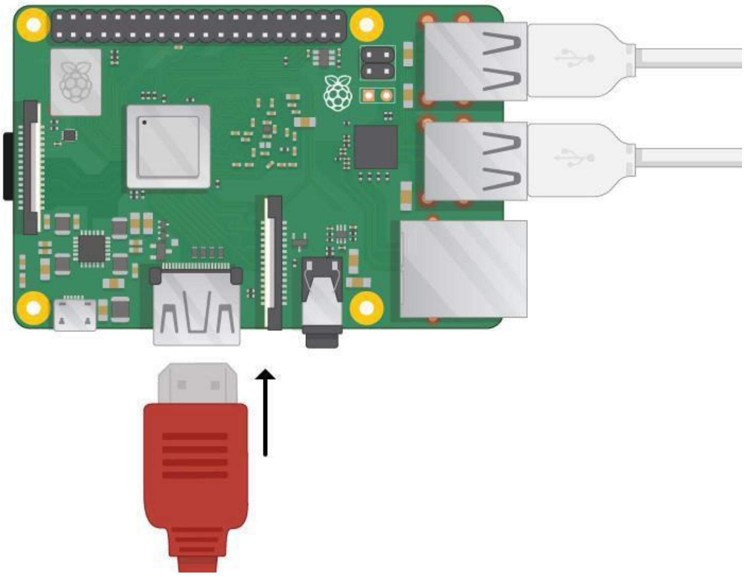


  You could connect an optional second screen in the same way.



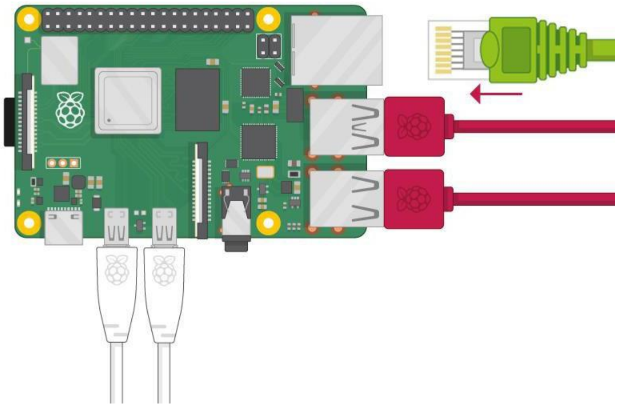
  Raspberry Pi 1, 2, 3

  Connect your screen to the single HDMI port.

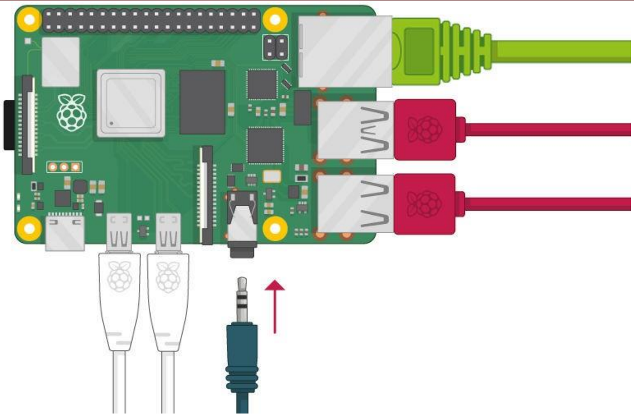


Note: nothing will display on the screen, because the Raspberry Pi is not running yet.

- If you want to connect the Pi to the internet via Ethernet, use an Ethernet cable to connect the Ethernet port on the Raspberry Pi to an Ethernet socket on the wall or on your internet router. You don’t need to do this if you want to use wireless connectivity, or if you don’t want to connect to the internet.



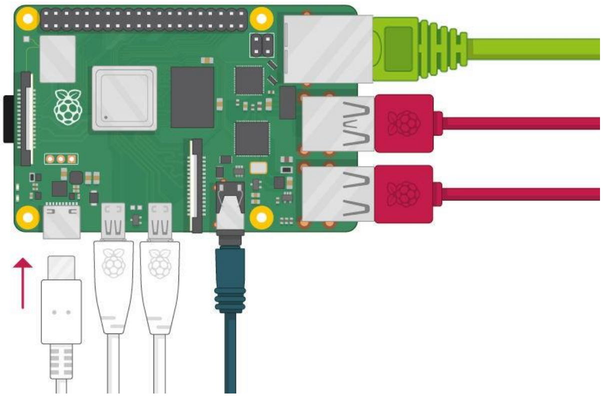
- If your screen has speakers, your Raspberry Pi can play sound through these. Or you couldconnect headphones or speakers to the audio port.



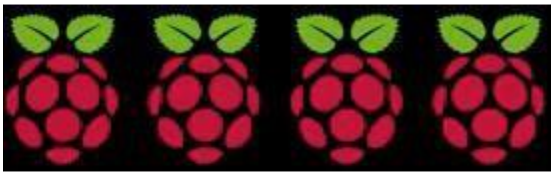
Start up your Raspberry Pi

Your Raspberry Pi doesn't have a power switch. As soon as you connect it to a power outlet, it will turn on.

- Plug the power supply into a socket and connect it to your Raspberry Pi's power port.



You should see a red LED light up on the Raspberry Pi, which indicates that Raspberry Pi is connected to power. As it starts up (this is also called booting), you will see raspberries appear in the top left-hand corner of your screen.



After a few seconds the Raspberry Pi OS desktop will appear.

Finish the setup

When you start your Raspberry Pi for the first time, the Welcome to Raspberry Pi applicationwill pop up and guide you through the initial setup.

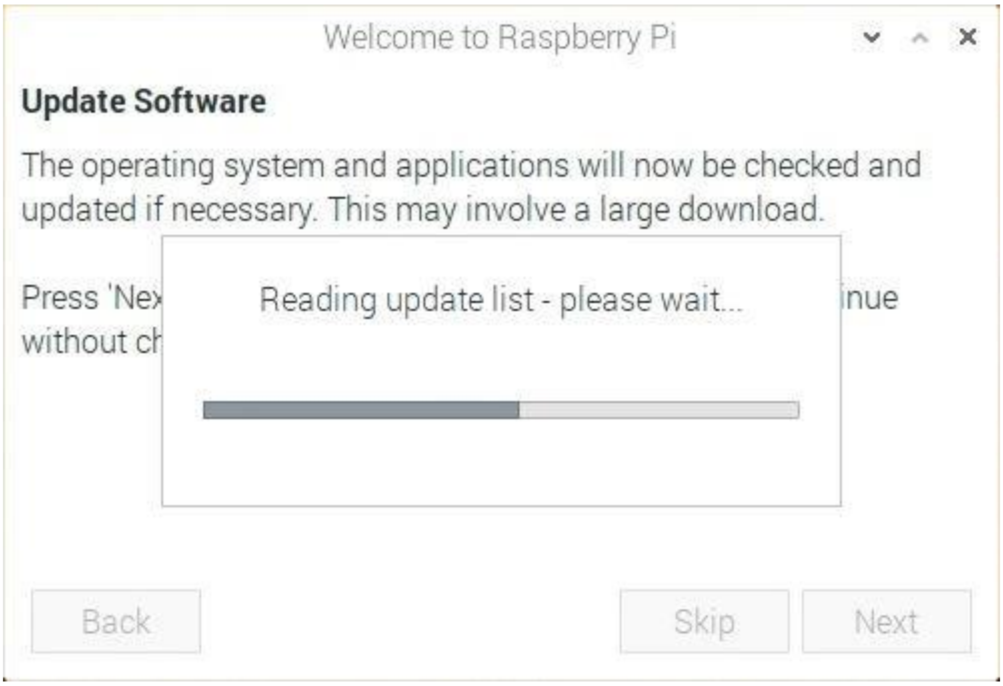
- Click Next to start the setup.
- Set your Country, Language, and Timezone, then click Next again.

- Enter a new password for your Raspberry Pi and click Next.

- Connect to your WiFi network by selecting its name, entering the password, andclicking Next.

Note: if your Raspberry Pi model doesn't have wireless connectivity, you won't see thisscreen.

- Click Next let the wizard check for updates to Raspbian and install them (this might take alittle while).



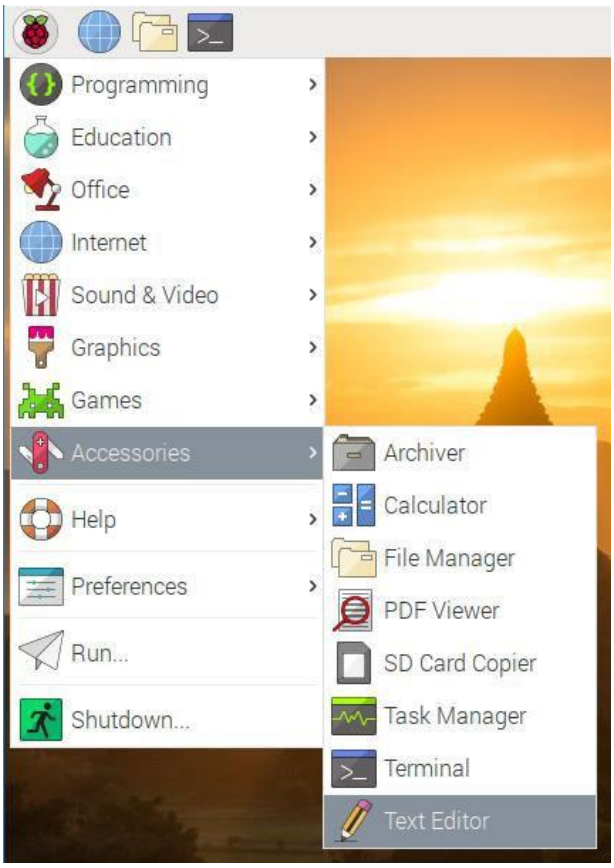
- Click Done or Reboot to finish the setup.

Note: you will only need to reboot if that's necessary to complete an update.

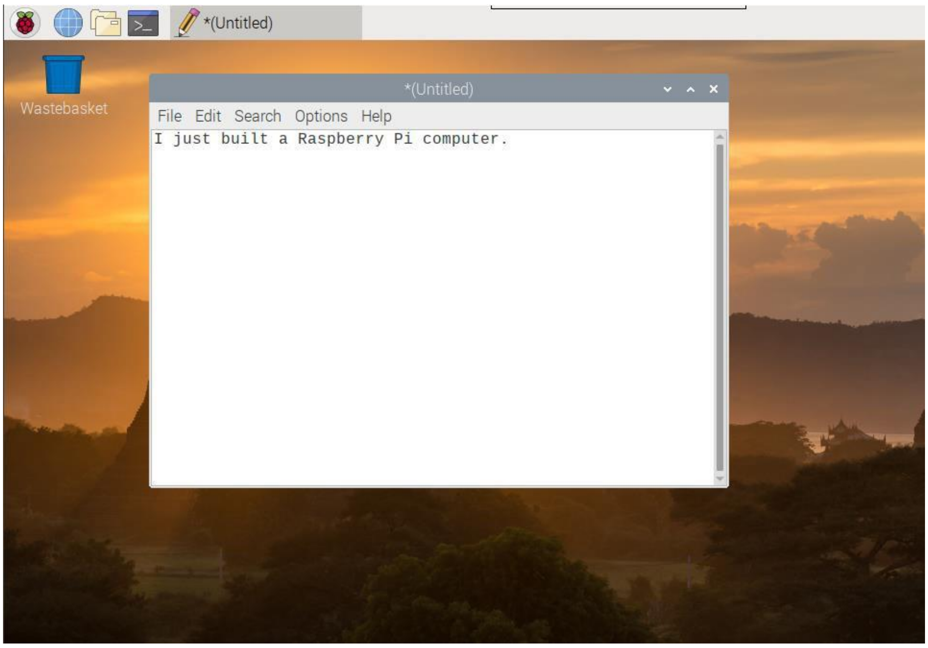
A tour of Raspberry Pi

Now it's time to take a tour of your Raspberry Pi.

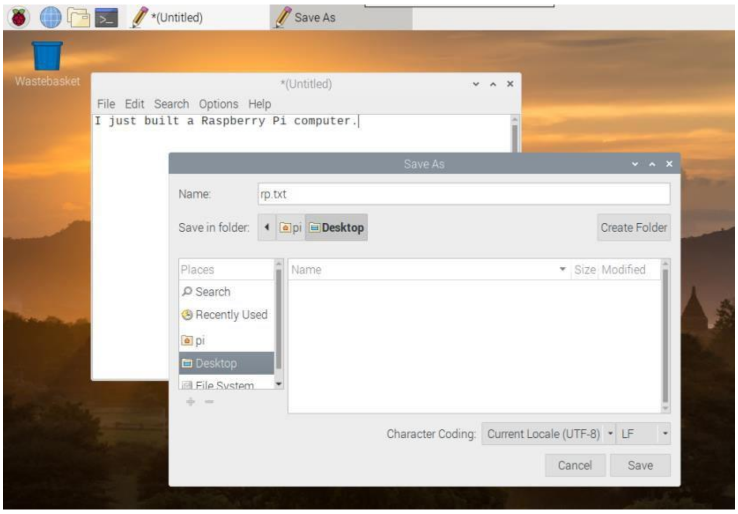
- Do you see the raspberry symbol in the top left-hand corner? That's where you access themenu: click on it to find lots of applications.
- Click on Accessories, and then click on Text Editor.



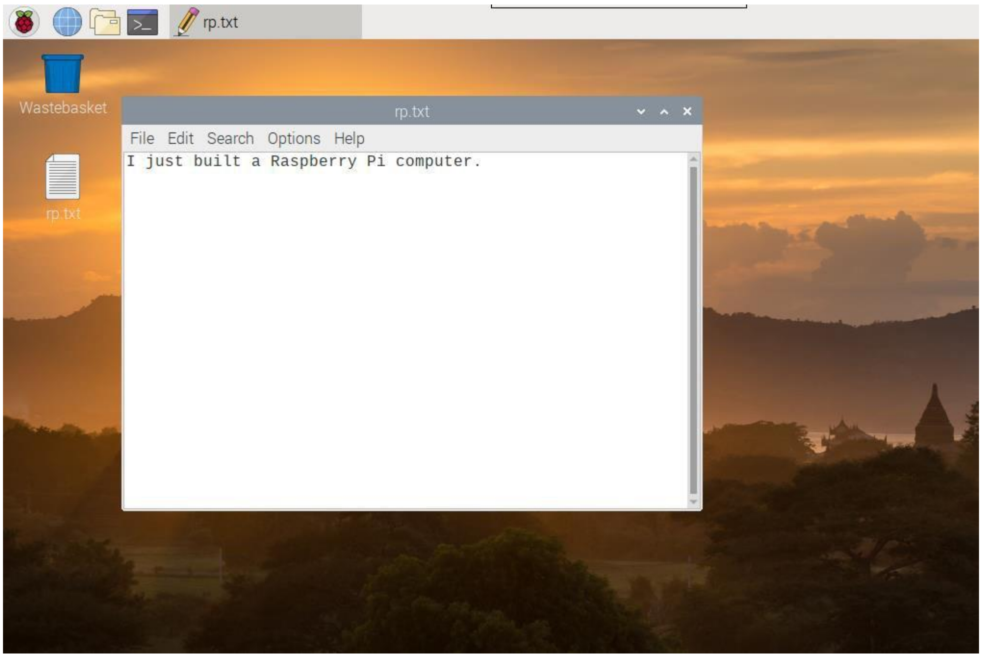
- Type I just built a Raspberry Pi computer in the window that appears.



Click on File, then choose Save, and then click on Desktop and save the file as

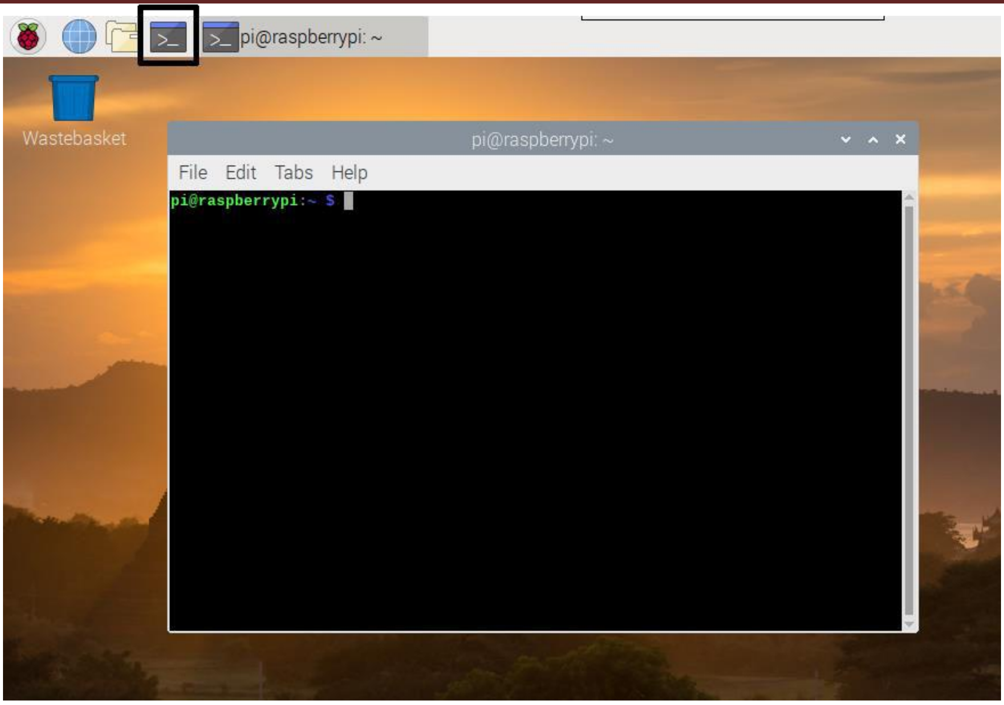


- You should see an icon named rp.txt appear on the desktop.



Your file has been saved to your Raspberry Pi's SD card.

- Close the text editor by clicking the X in the top right-hand corner of the window.
- Return to the menu, click on Shutdown, and then click on Reboot.
- When Raspberry Pi has rebooted, your text file should still be there on the desktop.
- Raspberry Pi runs a version of an operating system called Linux (Windows and macOS are other operating systems). This operating system allows you to make things happen by typing in commands instead of clicking on menu options. To try this out, click on the Terminal symbol at the top of the screen:



- In the window that appears, type:

```
ls
```

and then press Enter on the keyboard.

You can now see a list of the files and folders in your directory.

- Now type this command to change directory to the Desktop:

You have to press the Enter key after every command.

Then type:

```
ls
```

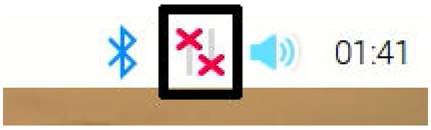
Can you see the text file you created?

- Close the terminal window by clicking on the X.
- Now drag to the Wastebasket on the desktop so the Raspberry Pi will be tidy for the next person using it.

Browsing the web

You might want to connect your Raspberry Pi to the internet. If you didn't plug in an ethernetcable or connect to a WiFi network during the setup, then you can connect now.

- Click the icon with red crosses in the top right-hand corner of the screen, and select your network from the drop-down menu. You may need to ask an adult which network you should choose.



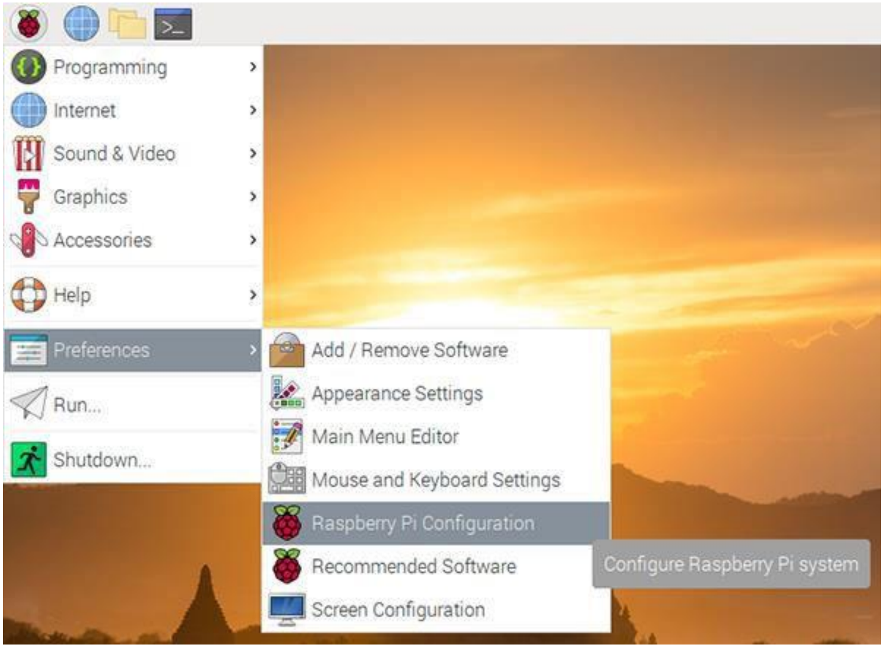
- Type in the password for your wireless network, or ask an adult to type it for you, thenclick OK.

- When your Pi is connected to the internet, you will see a wireless LAN symbol instead ofthe red crosses.

- Click the web browser icon and search for .

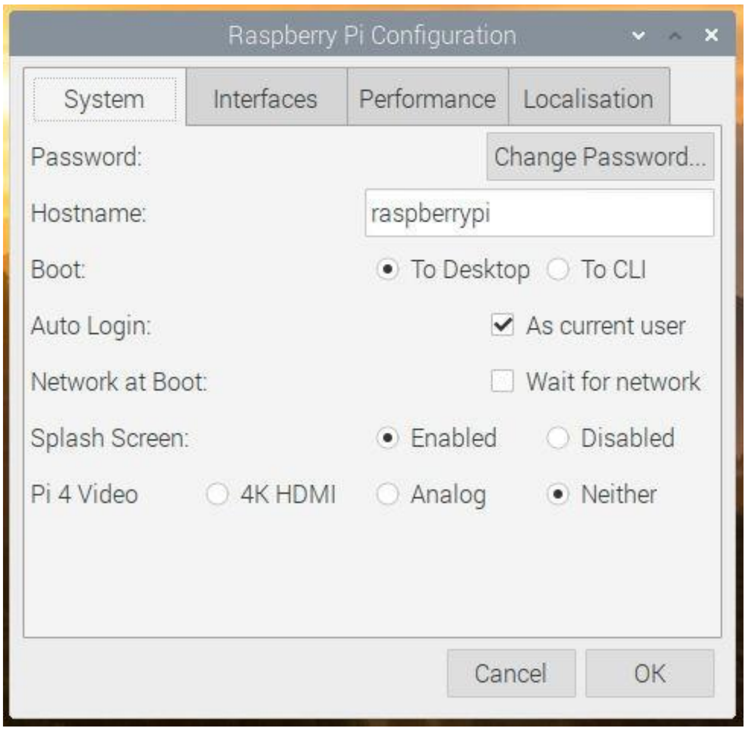
Configuring your Raspberry Pi

You can control most of your Raspberry Pi's settings, such as the password, throughthe Raspberry Pi Configuration application found in Preferences on the menu.



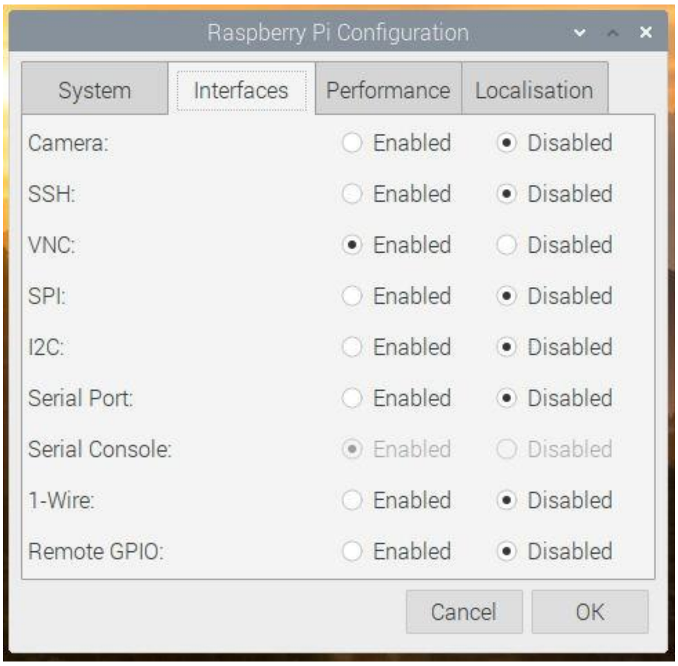
- System

  In this tab you can change basic system settings of your Raspberry Pi.



- Password — set the password of the pi user (it is a good idea to change the passwordfrom 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 inwhenever it starts
- Network at Boot — selecting this option will cause your Raspberry Pi to wait until anetwork connection is available before starting
- Splash Screen — choose whether or not to show the splash (startup) screen when yourRaspberry Pi boots
- Interfaces

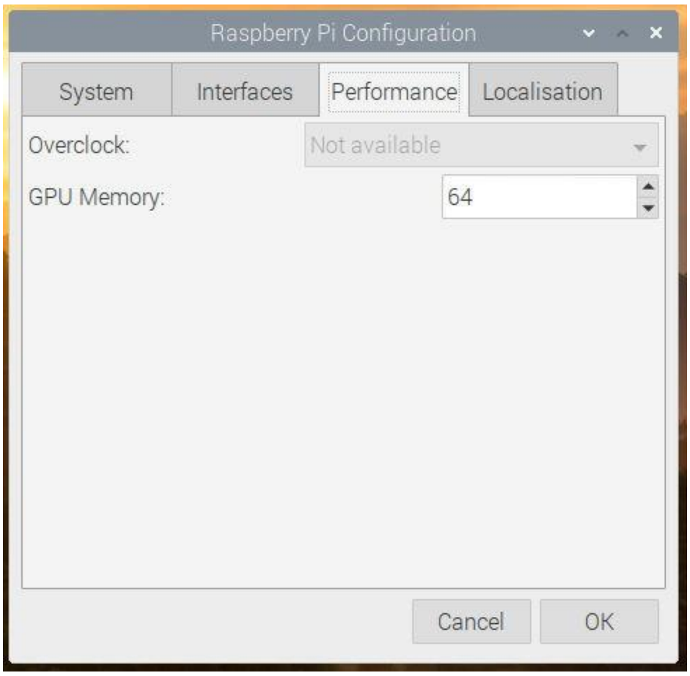
You can link devices and components to your Raspberry Pi using a lot of different types of connections. The Interfaces tab is where you turn these different connections on or off, so that your Raspberry Pi recognises that you've linked something to it via a particular type of connection.



- Camera — enable the [Raspberry Pi Camera Module]
- SSH — allow remote access to your Raspberry Pi from another computer usingSSH
- VNC — allow remote access to the Raspberry Pi Desktop from another computerusing 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 to your Raspberry Pi's GPIO pins from anothercomputer
- Performance

If you need to do so for a particular project you want to work on, you can change theperformance settings of your Raspberry Pi in this tab.

Warning: Changing your Raspberry Pi's performance settings may result in it behaving erratically or not working.



- Overclock — change the CPU speed and voltage to increase performance
- **GPU Memory** — change the allocation of memory given to the GPU
- Localisation

This tab allows you to change your Raspberry Pi settings to be specific to a country orlocation.

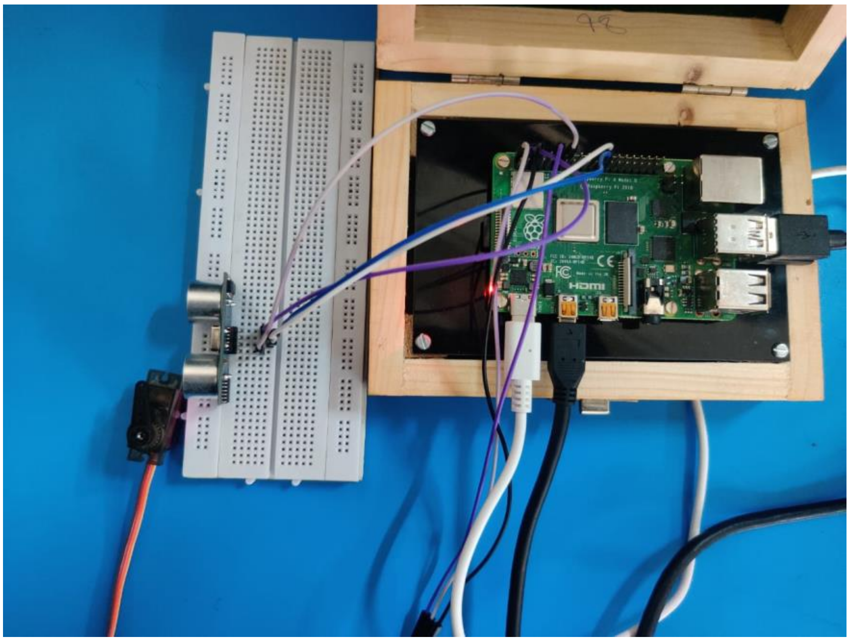
- Locale — set the language, country, and character set used by your Raspberry Pi
- Timezone — set the time zone
- Keyboard — change your keyboard layout
- WiFi Country — set the WiFi country code

Code

```
import RPi.GPIO as GPIO
import time

# Setup GPIO
GPIO.setwarnings(False)
TRIG = 11
ECHO = 8
servoPIN = 18
GPIO.setmode(GPIO.BCM)

# Ultrasonic sensor pin initialization
GPIO.setup(TRIG, GPIO.OUT)
GPIO.setup(ECHO, GPIO.IN)  # Corrected line: Set ECHO pin as input

# Servo motor pin initialization
GPIO.setup(servoPIN, GPIO.OUT)
servo = GPIO.PWM(servoPIN, 50)
servo.start(2.5)

try:
        while True:
         GPIO.output(TRIG, False)
         time.sleep(0.000002)

         GPIO.output(TRIG, True)
        time.sleep(0.00001)
        GPIO.output(TRIG, False)

         startTime = time.time()
        stopTime = time.time()

        while GPIO.input(ECHO) == 0:
        startTime = time.time()

        while GPIO.input(ECHO) == 1:
        stopTime = time.time()
        GPIO.output(TRIG, True)

        timeElapsed = stopTime - startTime
        distance = (timeElapsed * 34300) / 2
        distance = int(distance)
        print("Distance: {} cm".format(distance))
```

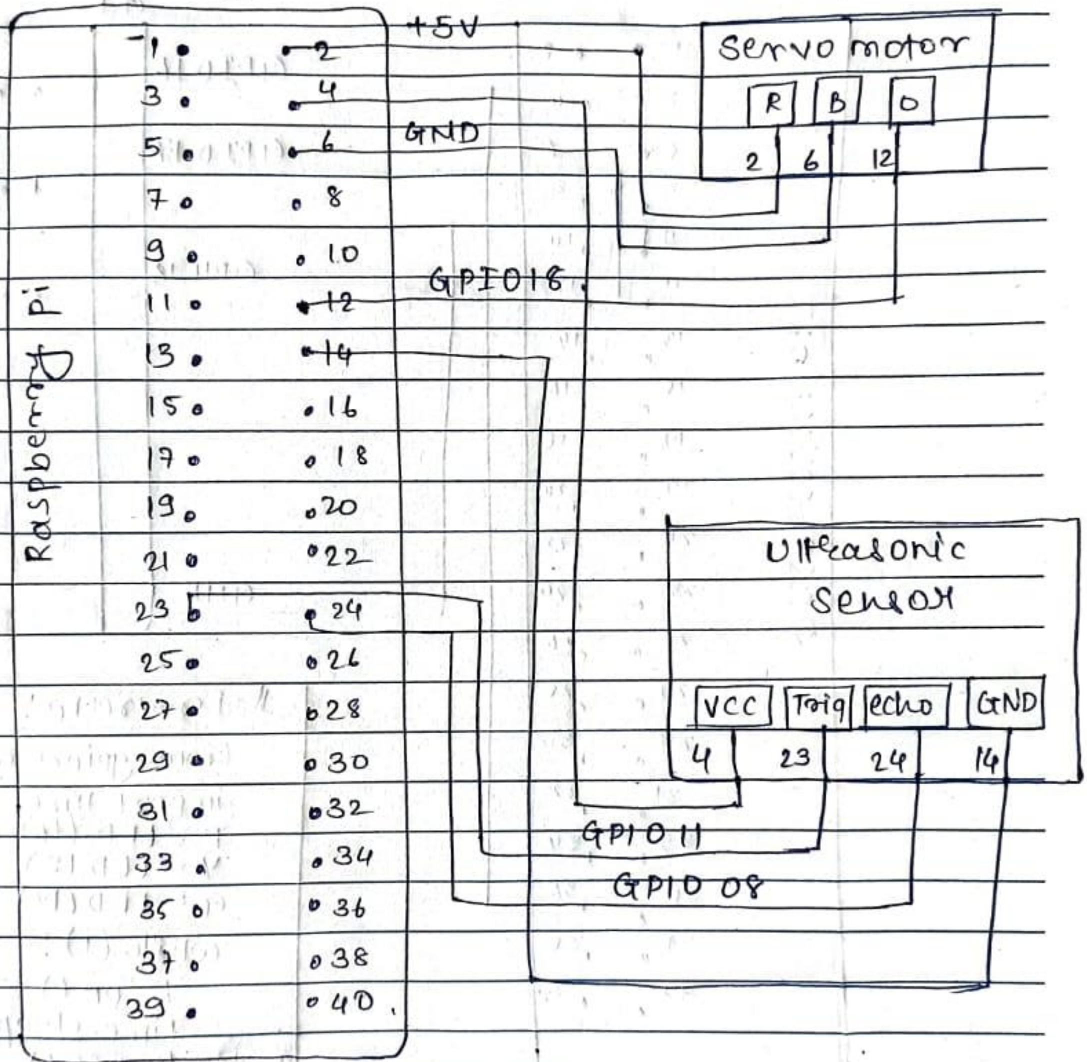
```
if distance <= 20:
duty_cycle = 12.5  # Adjust this value for desired servo position
servo.ChangeDutyCycle(duty_cycle)
print("Motor Rotated")
time.sleep(0.1)
 else:
        duty_cycle = 2.5  # Adjust this value for desired servo position
         servo.ChangeDutyCycle(duty_cycle)
         print("Motor is at orignal position")
          time.sleep(0.1)

        except KeyboardInterrupt:
         servo.stop()
         GPIO.cleanup()
```

Hardware Connection: -



R - Vcc
B - GND
O - o/p

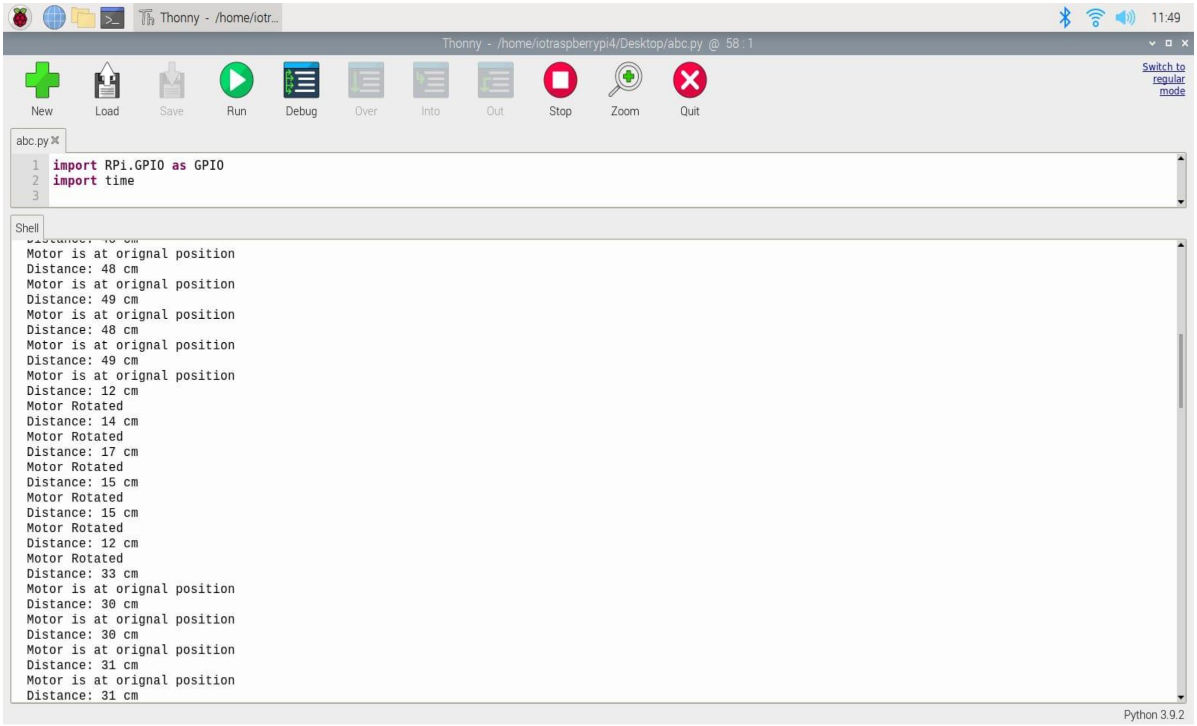
Circuit Diagram:



** 23 = GPIO 11
24 = GPIO 08
12 = GPIO 18

Zeal Education Institutes

Output:


Thonny - /home/iotr...
11:49
Thonny - /home/iotraspberrypi4/Desktop/abc.py @ 58 : 1
New
Load
Save
Run
Debug
Over
Into
Out
Stop
Zoom
Quit
Switch to regular mode
abc.py
import RPi.GPIO as GPIO
import time
Shell
Motor is at orignal position
Distance: 48 cm
Motor is at orignal position
Distance: 49 cm
Motor is at orignal position
Distance: 48 cm
Motor is at orignal position
Distance: 49 cm
Motor is at orignal position
Distance: 12 cm
Motor Rotated
Distance: 14 cm
Motor Rotated
Distance: 17 cm
Motor Rotated
Distance: 15 cm
Motor Rotated
Distance: 15 cm
Motor Rotated
Distance: 12 cm
Motor Rotated
Distance: 33 cm
Motor is at orignal position
Distance: 30 cm
Motor is at orignal position
Distance: 30 cm
Motor is at orignal position
Distance: 31 cm
Motor is at orignal position
Distance: 31 cm
Python 3.9.2

IoT PRACTICAL 2

Aim : Design and implement parameter monitoring IoT system keeping records on Cloud such as 'environment humidity and temperature monitoring'.

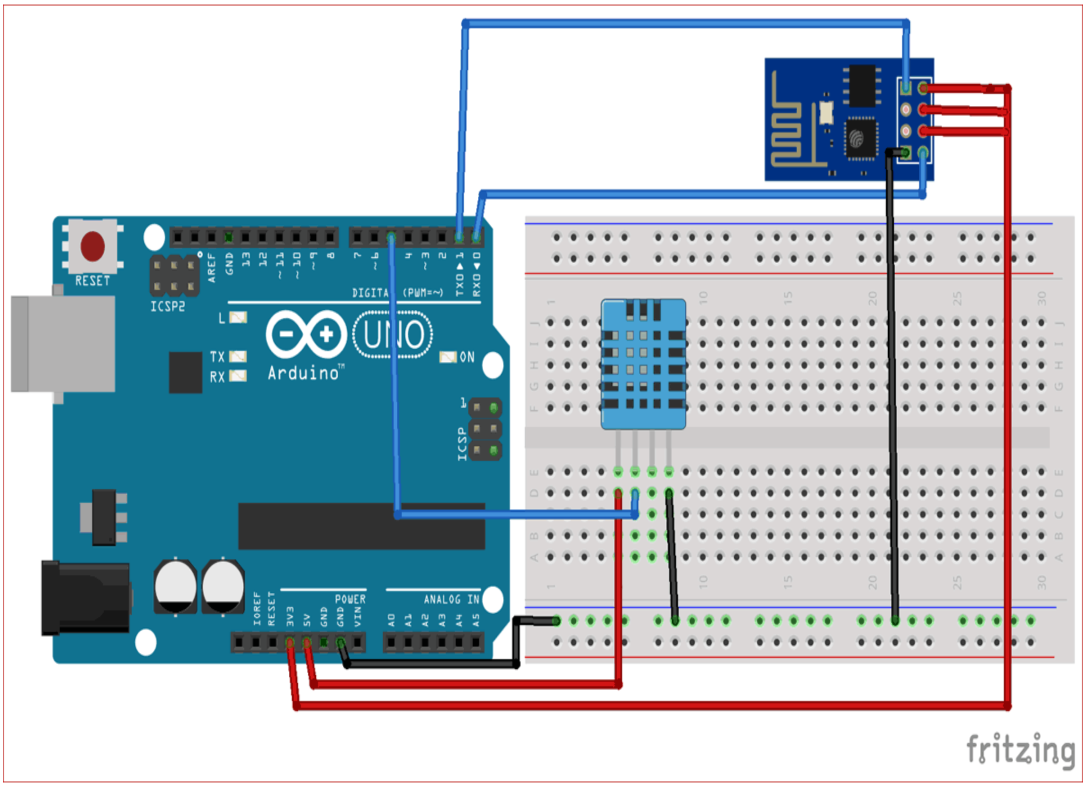
Objective : To design and implement Environment humidity & Temperature Monitoring.

Theory :
we are using the **DHT11 sensor for sending Temperature and Humidity data to Thingspeak using Arduino and ESP8266**. By this method, we can monitor our DHT11 sensor's temperature and humidity data over the internet using the ThingSpeak IoT server. And we can view the logged data and graph overtime on the Thingspeak website.

Components Required

- Arduino Uno
- ESP8266 WiFi Module
- DHT11 Sensor
- Breadboard
- Jumper Wires

Temperature and Humidity Monitoring System Circuit Diagram

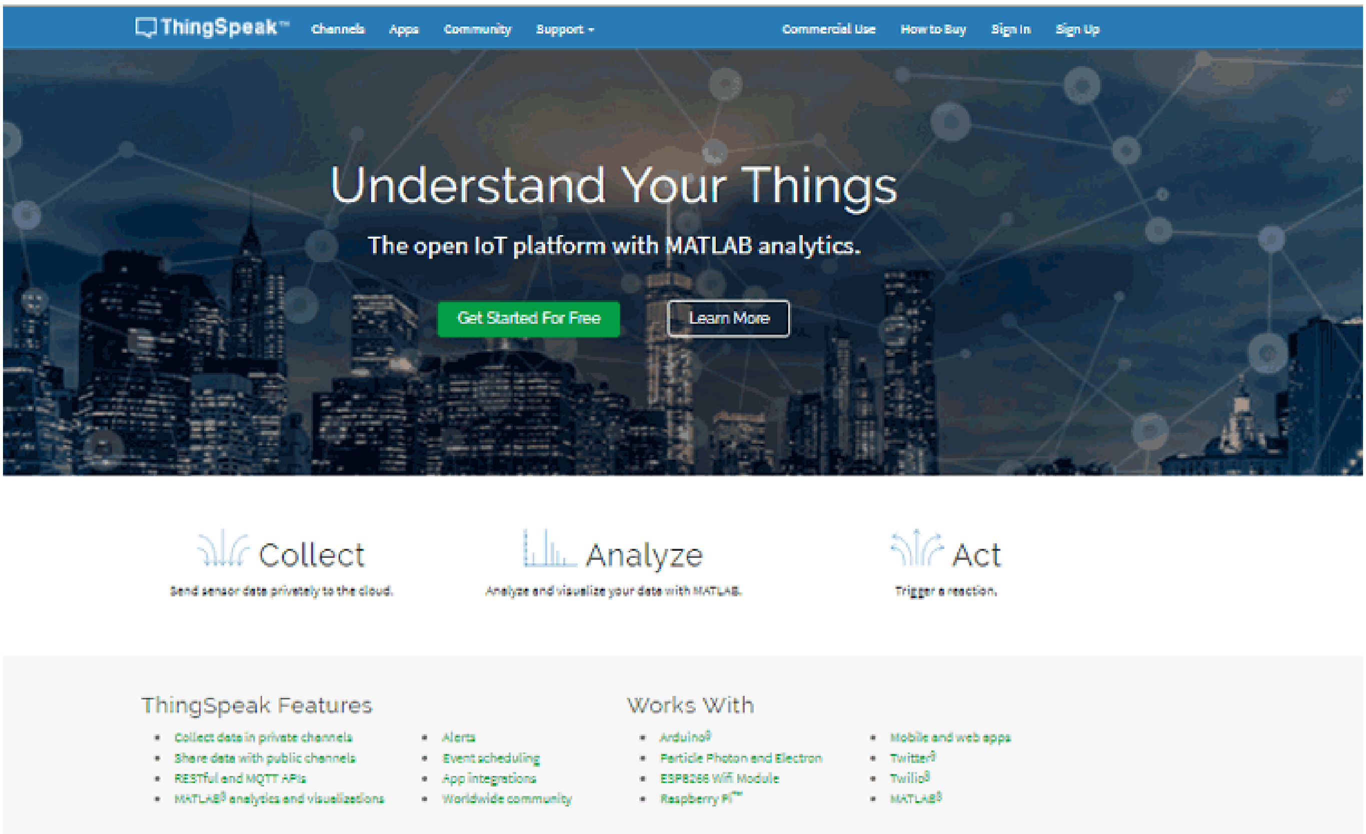


S.NO.	**Pin Name**	**Arduino Pin**
1	ESP8266 VCC	3.3V
2	ESP8266 RST	3.3V
3	ESP8266 CH-PD	3.3V
4	ESP8266 RX	TX
5	ESP8266 TX	RX
6	ESP8266 GND	GND
7	DHT-11 VCC	5V
8	DHT-11 Data	5
9	DHT-11 GND	GND

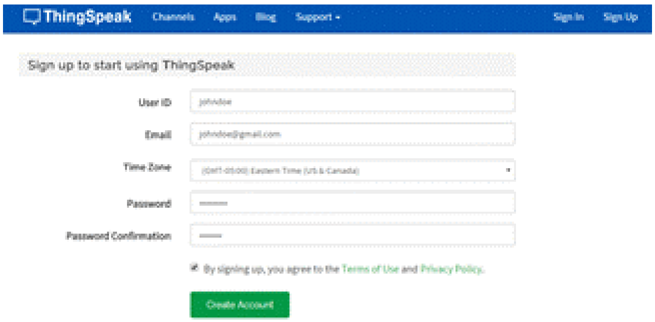
Step 1: ThingSpeak Setup for Temperature and Humidity Monitoring

For creating your channel on Thingspeak, you first need to Sign up on Thingspeak. In case if you already have an account on Thingspeak, just sign in using your id and password.

For creating your account go to www.thinspeak.com

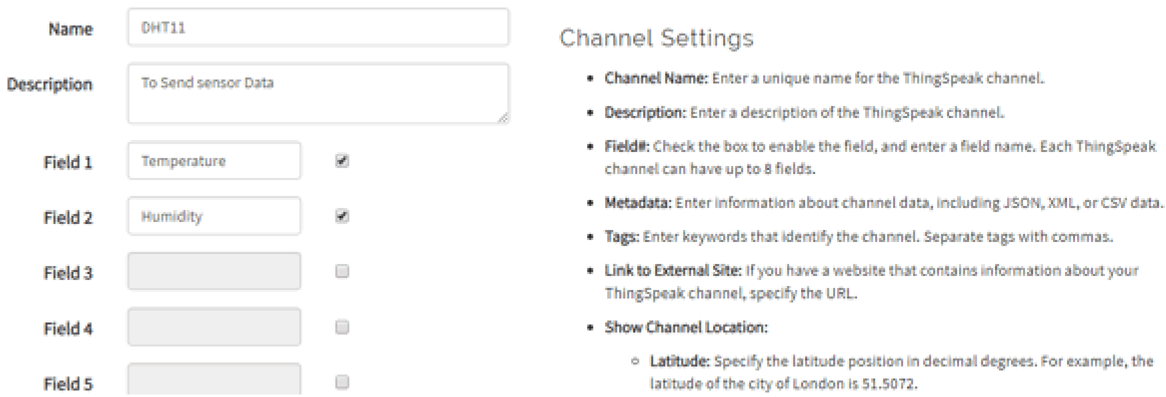


Click on Sing up if you don't have account and if you already have an account, then click on sign in.After clicking on signup, fill in your details.



Step 2: Create a Channel for Your Data

Once you Sign in after your account verification, Create a new channel by clicking "New Channel" button.



After clicking on "New Channel", enter the Name and Description of the data you want to upload on this channel. For example, I am sending my DHT11 sensor data, so I named it DHT11 data.

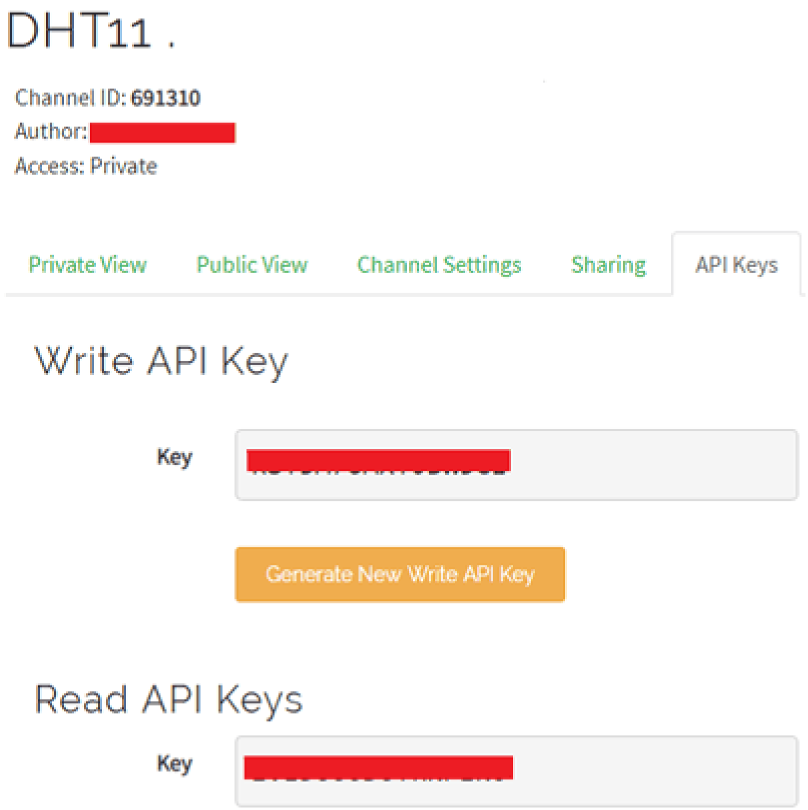
Enter the name of your data 'Temperature' in Field1 and 'Humidity' in Field2. If you want to use more Fields, you can check the box next to Field option and enter the name and description of your data.

After this, click on the save channel button to save your details.

Step 3: API Key

To send data to Thingspeak, we need a unique API key, which we will use later in our code to upload our sensor data to Thingspeak Website.

Click on "API Keys" button to get your unique API key for uploading your sensor data.



Now copy your "Write API Key". We will use this API key in our code.

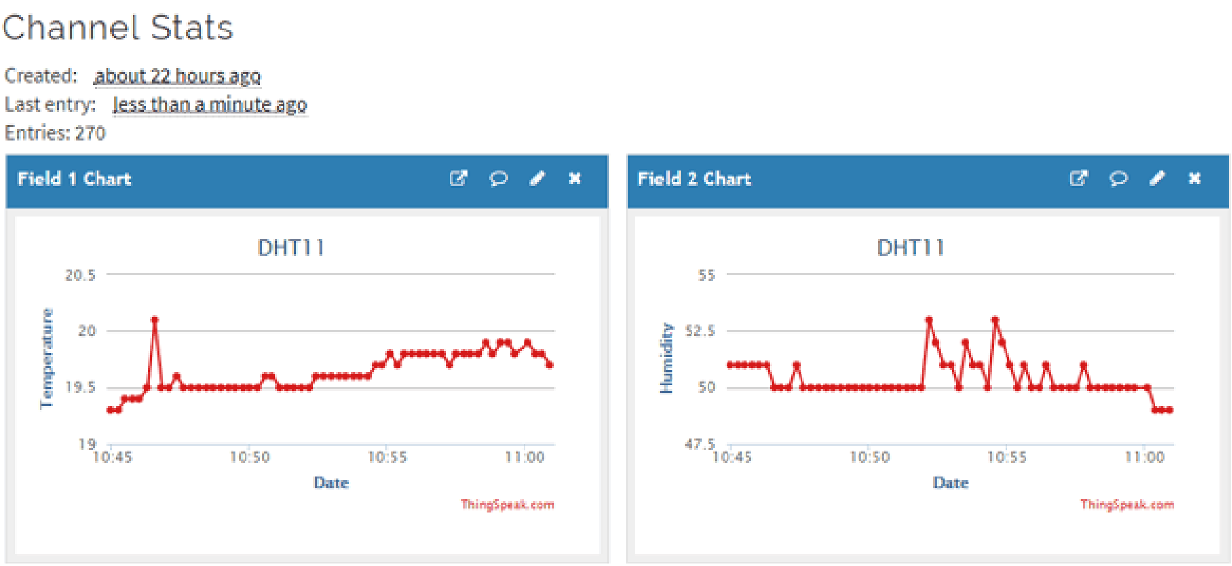
Programming Arduino for Sending data to ThingSpeak

To program Arduino, open Arduino IDE and choose the correct board and port from the *'tool'* menu.

Upload it in Arduino UNO. If you successfully upload your program, Serial monitor will look like this:

```
AT
AT+CWMODE=1
AT+CWJAP="CircuitLoop","circuitdigest101"
AT+CIPSTART="TCP","184.106.153.149",80
AT+CIPSEND=56
AT+CIPCLOSE
AT+CIPSTART="TCP","184.106.153.149",80
AT+CIPSEND=56
GET /update?key=9B6ILVOYMUSVOADA&field1=19.7&field2=50
AT+CIPSTART="TCP","184.106.153.149",80
AT+CIPSEND=56
GET /update?key=9B6ILVOYMUSVOADA&field1=19.7&field2=50
AT+CIPSTART="TCP","184.106.153.149",80
AT+CIPSEND=56
GET /update?key=9B6ILVOYMUSVOADA&field1=19.7&field2=50
AT+CIPSTART="TCP","184.106.153.149",80
AT+CIPSEND=56
GET /update?key=9B6ILVOYMUSVOADA&field1=19.7&field2=50
AT+CIPSTART="TCP","184.106.153.149",80
AT+CIPSEND=56
GET /update?key=9B6ILVOYMUSVOADA&field1=19.8&field2=50
AT+CIPSTART="TCP","184.106.153.149",80
AT+CIPSEND=56
```

After this navigate to your Thingspeak page and open your channel at Thingspeak and output will be shown as below:



Hence, we have successfully monitored Temperature and Humidity data over ThingSpeak using Arudino and ESP32.

Code:

```
#include <stdlib.h>

#include <DHT.h>

#define DHTPIN 5         // DHT data pin connected to Arduino pin 5

#define DHTTYPE DHT11     // DHT11 (DHT Sensor Type )

DHT dht(DHTPIN, DHTTYPE); // Initialize the DHT sensor

#define SSID "WiFi Name"     // "WiFi Name"

#define PASS "WiFi Password"       // "Password"

#define IP "184.106.153.149"// thingspeak.com ip

String msg = "GET /update?key=Your API Key"; //change it with your key...

float temp;

int hum;

String tempC;

int error;

void setup()

{

  Serial.begin(115200); // use default 115200.

  Serial.println("AT");

  delay(5000);

  if(Serial.find("OK")){

    connectWiFi();

  }

}
```

```
void loop(){
  start:
 error=0;
 temp = dht.readTemperature();
 hum = dht.readHumidity();
 char buffer[10];
 tempC = dtostrf(temp, 4, 1, buffer);
 updateTemp();
 if (error==1){
   goto start;
 }

 delay(5000);
}
void updateTemp(){
 String cmd = "AT+CIPSTART=\"TCP\",\"";
 cmd += IP;
 cmd += "\",80";
 Serial.println(cmd);
 delay(2000);
 if(Serial.find("Error")){
   return;
 }
 cmd = msg ;
```

```
  cmd += "&field1=";
  cmd += tempC;
  cmd += "&field2=";
  cmd += String(hum);
  cmd += "\r\n";
  Serial.print("AT+CIPSEND=");
  Serial.println(cmd.length());
  if(Serial.find(">")){
    Serial.print(cmd);
  }
  else{
    Serial.println("AT+CIPCLOSE");
    //Resend...
    error=1;
  }
}
  boolean connectWiFi(){
  Serial.println("AT+CWMODE=1");
  delay(2000);
  String cmd="AT+CWJAP=\"";
  cmd+=SSID;
  cmd+="\",\"";
  cmd+=PASS;
  cmd+="\"";
```

```
  Serial.println(cmd);

  delay(5000);

  if(Serial.find("OK")){

    return true;

  }else{

    return false;

  }

}
```

Conclusion:

This is we have run program successfully. We get Temperature on Thing-speak.

IoT Practical 3

Aim: Design and implement real time monitoring system using android phone (Blynk App.) such as 'soil-parameter monitoring'.

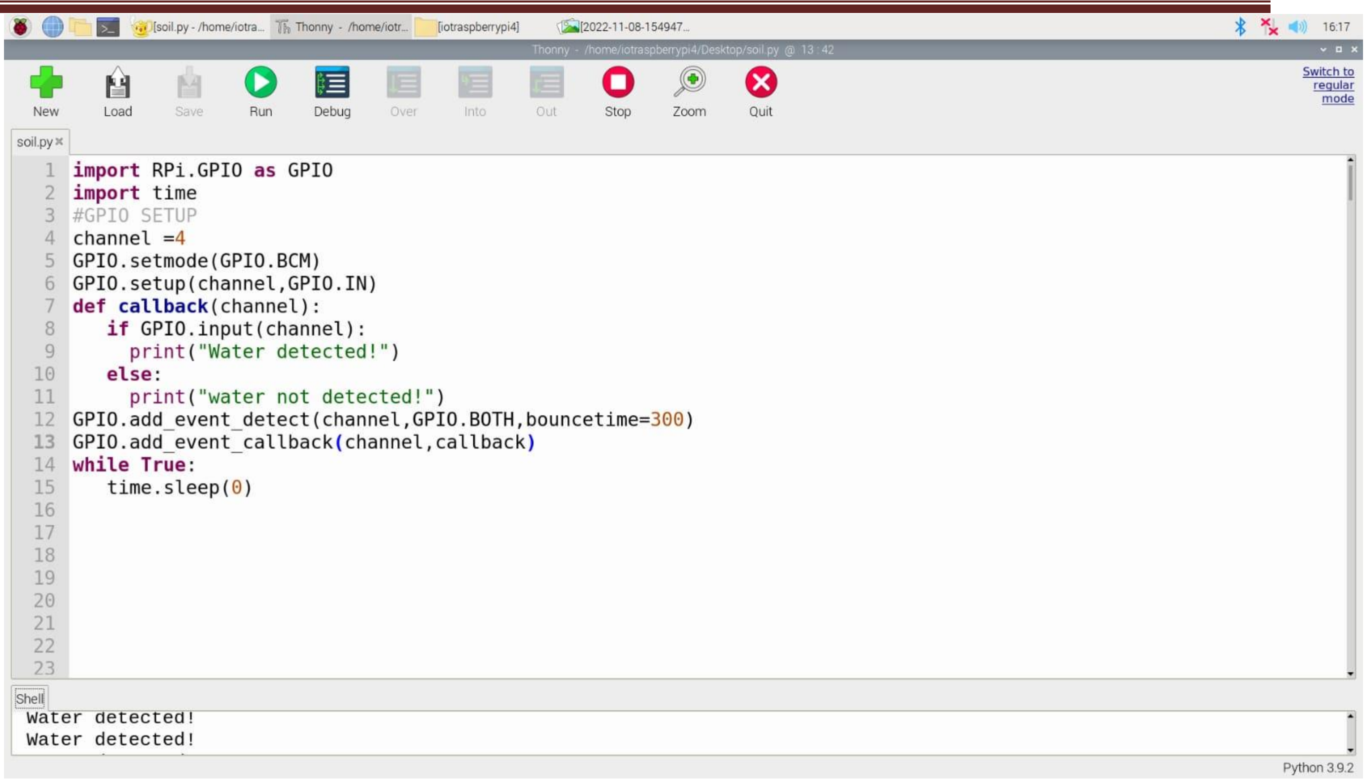
Objective: Real Time Monitoring of Temperature on Android Phone using Soil Parameter.

Parameters: Raspberry Pi , BLYNK ,Soil Moisture Sensor .

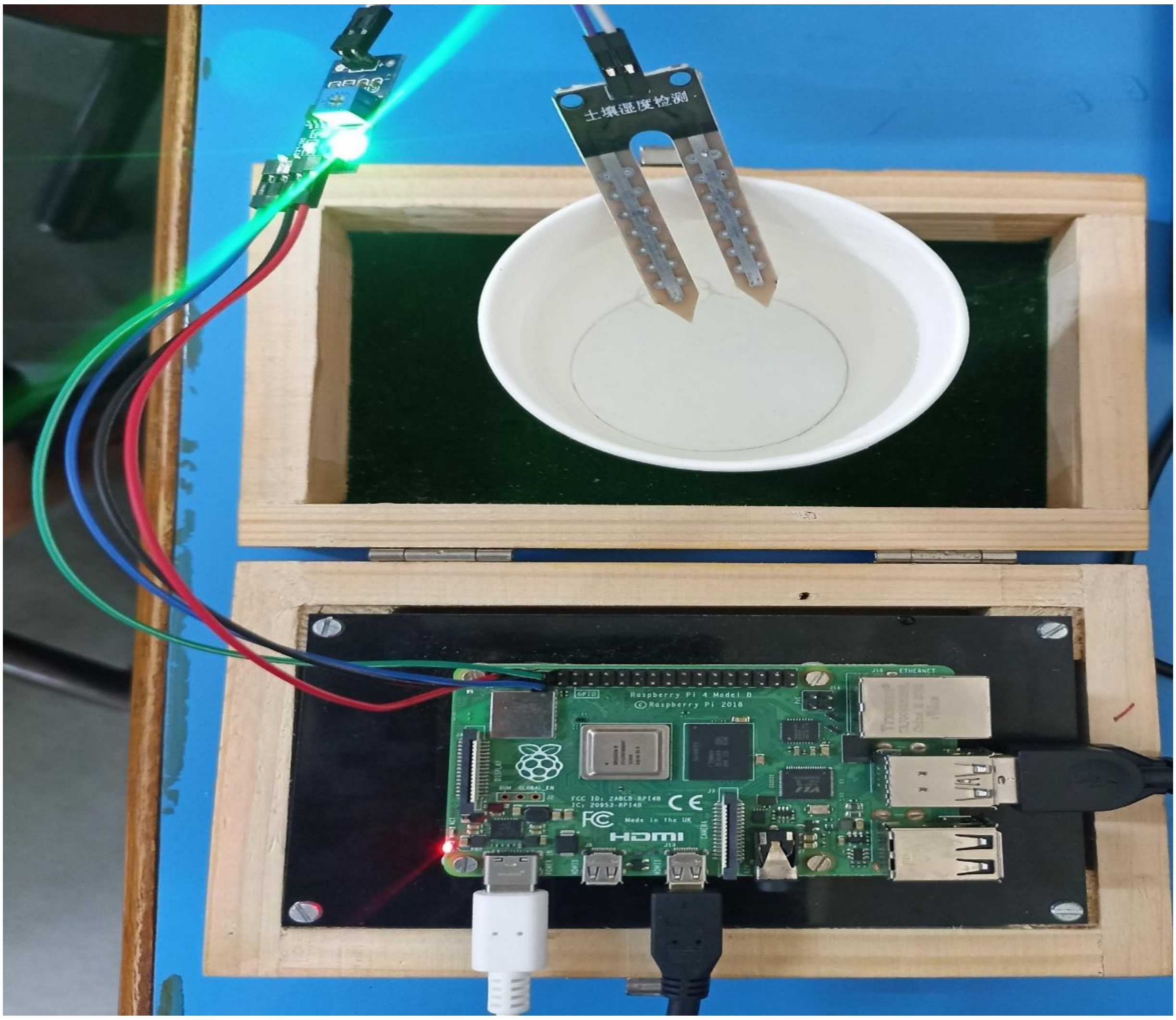
Theory:

- Agriculture has been the most important practice from very beginning of the human civilization. It has seen many iterations of development in technology with time. A good agricultural practice is still an art.
- In last few decades changing weather condition, increase in global temperature and pollution, has led to abnormal environmental conditions like raining. Traditional way of farming is unable to cope with these environmental changes.
- Good control over Environmental parameters like temperature, humidity, and moisture plays important role in growth of the plant. Temperature affects many of plant activities such as pollination, germination etc.
- It is observed that, at higher temperature, respiration rate increases that result in reduction of sugar contents of fruits and vegetables. At lower temperatures photosynthesis activity is slowed down. Till date many methods have come into existence where water can be limitedly consumed.
- A method where monitoring water status and based on status of water whether it is high or low irrigation is scheduled which is based on canopy temperature of plant, which was captured with thermal imaging. Another method is making use of information on volumetric water content of soil, using dielectric moisture sensors to control actuators and save water, instead of the scheduled irrigation at a particular time of day and supplying water only for a specific duration. This above method just opens the valve and supply water to bedding plants when volumetric content of soil will drop below threshold value. In this paper a use of the second method where sensors are placed and based on that water is supplied to the field and intimated to the farmer using software application.
- Wireless sensor networks is also called as wireless sensors and actor network, are distributed spatially autonomous sensors to monitor physical or environmental conditions as temperature, pressure sound, moisture etc. and it co-operatively passes these data via network to the main location.

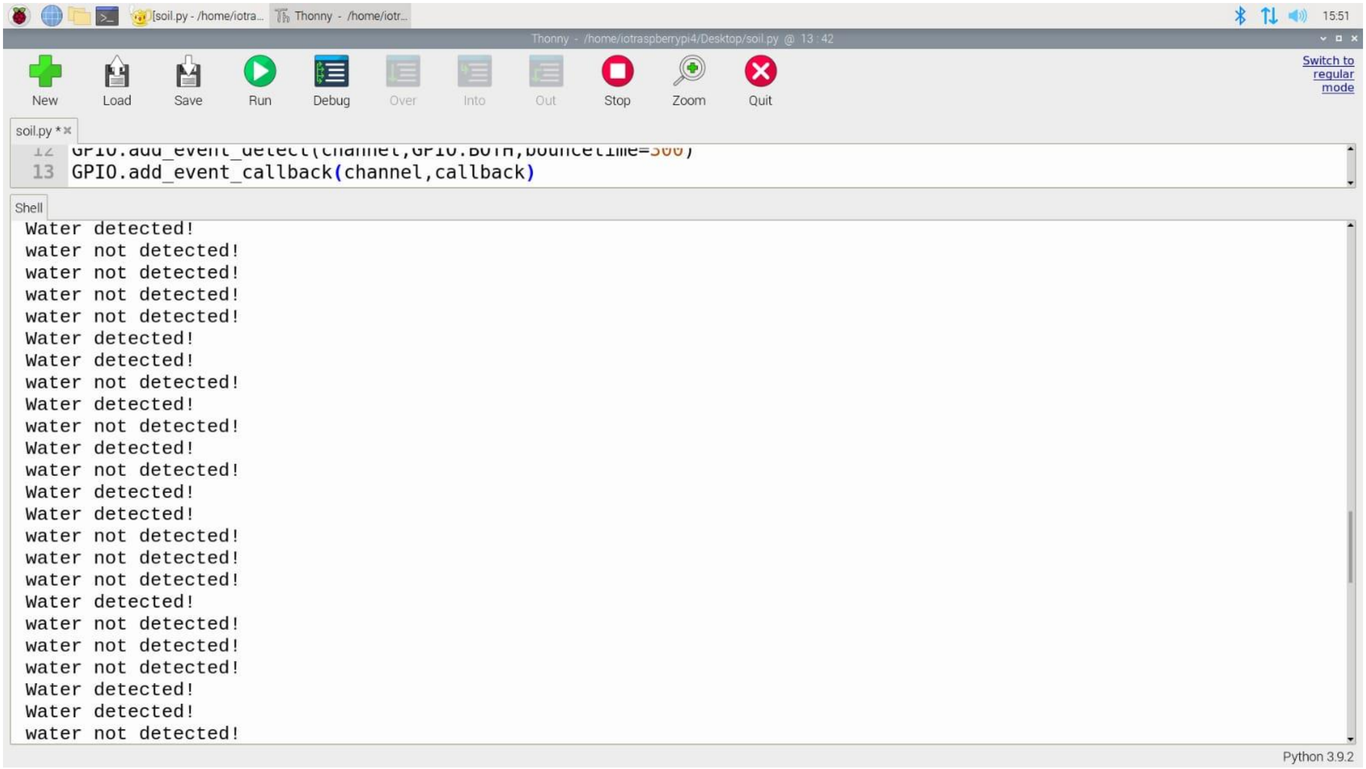
Code


[soil.py - /home/iotra...
Thonny - /home/iotr...
[iotraspberrypi4]
[2022-11-08-154947...
16:17
Thonny - /home/iotraspberrypi4/Desktop/soil.py @ 13 : 42
New
Load
Save
Run
Debug
Over
Into
Out
Stop
Zoom
Quit
Switch to regular mode
soil.py
import RPi.GPIO as GPIO
import time
#GPIO SETUP
channel =4
GPIO.setmode(GPIO.BCM)
GPIO.setup(channel,GPIO.IN)
def callback(channel):
   if GPIO.input(channel):
     print("Water detected!")
   else:
     print("water not detected!")
GPIO.add_event_detect(channel,GPIO.BOTH,bouncetime=300)
GPIO.add_event_callback(channel,callback)
while True:
   time.sleep(0)
Shell
Water detected!
Water detected!
Python 3.9.2

Hardware Diagram:



Output:


Thonny - /home/iotraspberrypi4/Desktop/soil.py @ 13 : 42
New
Load
Save
Run
Debug
Over
Into
Out
Stop
Zoom
Quit
Switch to regular mode
soil.py
13 GPIO.add_event_callback(channel,callback)
Shell
Water detected!
water not detected!
water not detected!
water not detected!
water not detected!
Water detected!
Water detected!
water not detected!
Water detected!
water not detected!
Water detected!
water not detected!
Water detected!
Water detected!
water not detected!
water not detected!
water not detected!
Water detected!
water not detected!
water not detected!
water not detected!
Water detected!
Water detected!
water not detected!
Python 3.9.2

IoT Practical 4

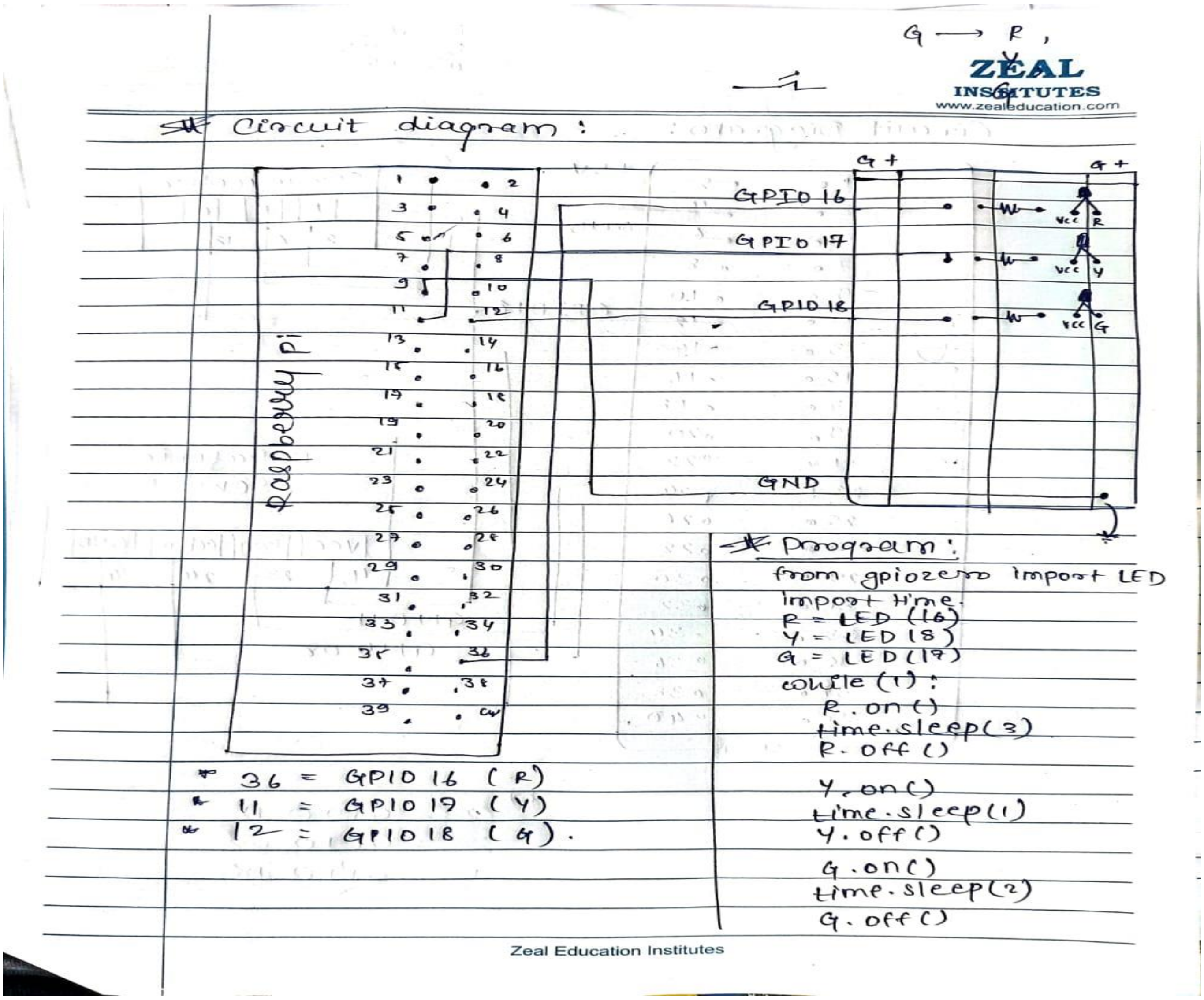
Aim: Design and implement IoT system for one of the applications like: Traffic Application, Medical/Health application, Social Application etc.

Objective: Using for the Manage Traffic Light Signal.

Parameters: LED R,G,B, Breadboard, Raspberry Pi Kit.

Diagram:

Circuit Diagram:



Theory:

To get started, you'll need to place all the components on the breadboard and connect them to the appropriate GPIO pins on the Raspberry Pi.

- First, you need to understand how each component is connected:
  - An LED requires GPIO pin 36,11,12 .
  - A ground pin 3 of Raspberry Pi.

  Each component requires its own individual GPIO pin, but components can share a ground pin. We will use the breadboard to enable this.

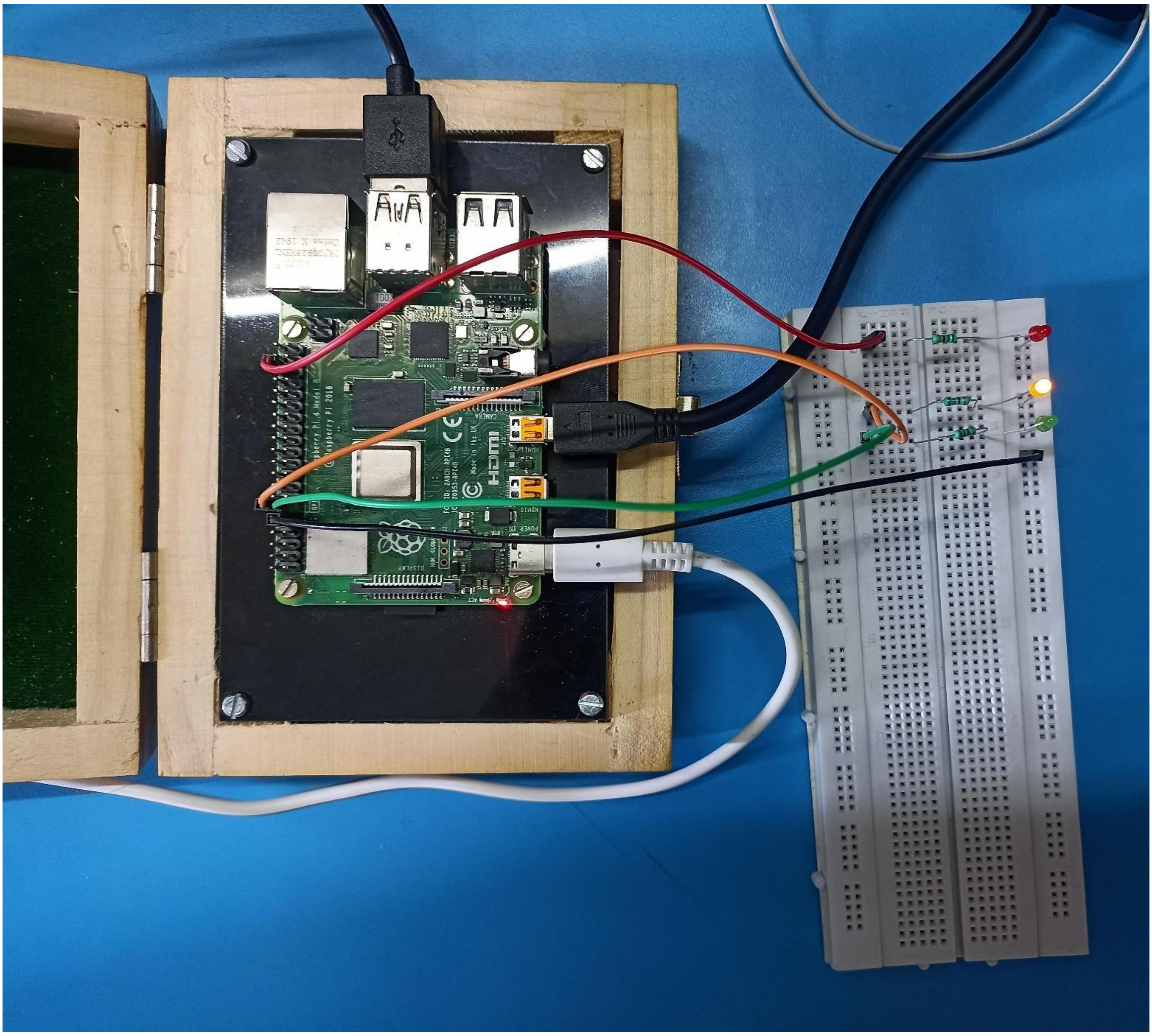
- Place the components on the breadboard and connect them to the Raspberry Pi GPIO pins, according to the above diagram:

Component	GPIO pin
Button	21
Red LED	16
Yellow LED	18
Green LED	17
Breadboard	

Code:

```
from gpiozero import LED
import time
R=LED(16)
Y=LED(18)
G=LED(17)
while(1) :
    R.on()
    time.sleep(3)
    R.off()

    Y.on()
    time.sleep(2)
    Y.off()

    G.on()
    time.sleep(3)
    G.off()
```

Output:



Conclusion:

Thus we have executed successfully the Traffic Signal Using Raspberry Pi.