Assignment 1 – Programming Raspberry Pi to Read/Write Sensors and Actuators

Controlling LED with a switch, an LDR sensor, and a DHT sensor

Outline

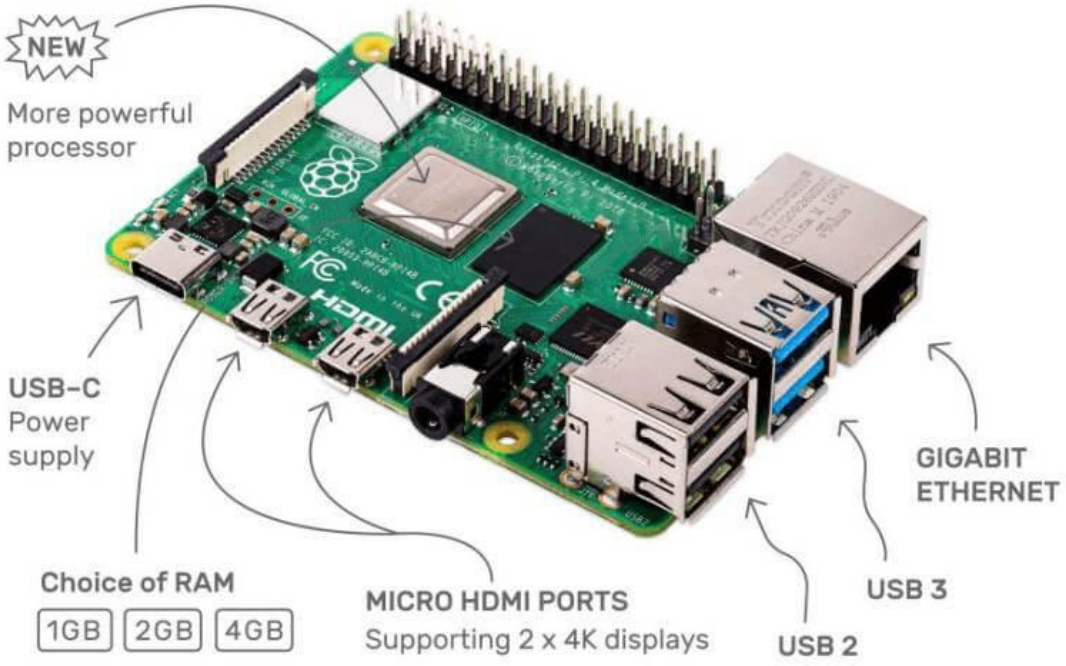
- Objectives
- Basic Raspberry Pi
- OS installation
- A simple IoT application
  - Controlling LED
    - With Raspberry Pi
    - With Switch
    - With Light Sensor (LDR)
    - With Temperature and Humidity Sensor (DHT)
- Assignment 1 - Specification

Objectives

- Get to know Raspberry Pi
- Capable to install the Raspbian operating system
- Connecting sensors and actuators to Raspberry Pi
- Writing the code to run the sensors and actuators

Basic Raspberry Pi – What is a Raspberry Pi?

A low cost, credit-card sized computer

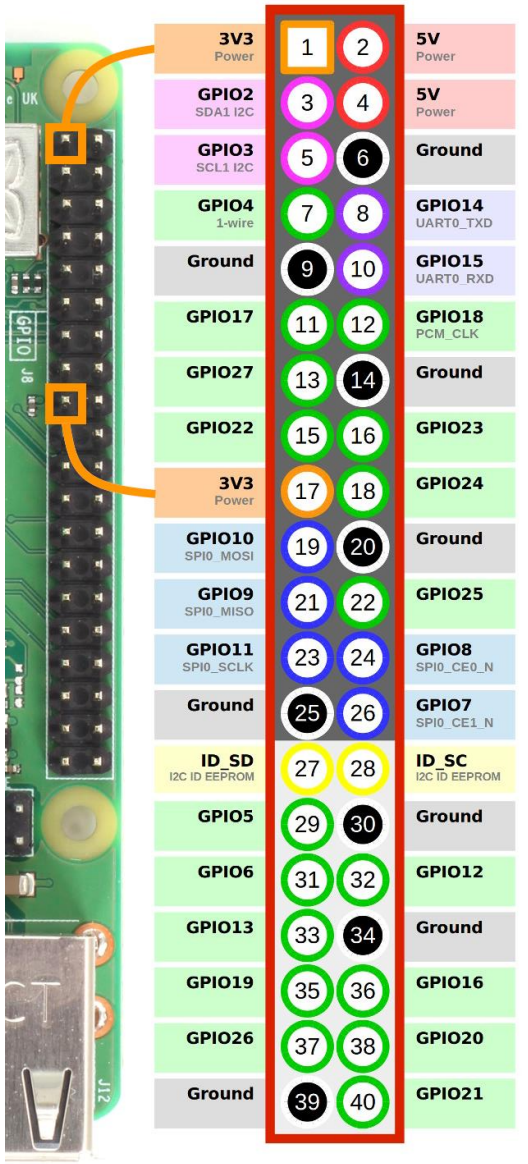


Processor:	Broadcom BCM2711, quad-core Cortex-A72 (ARM v8) 64-bit SoC @ 1.5GHz
Memory:	2GB, 4GB or 8GB LPDDR4 (depending on model)
Connectivity:	2.4 GHz and 5.0 GHz IEEE 802.11b/g/n/ac wireless LAN, Bluetooth 5.0, BLE Gigabit Ethernet 2 × USB 3.0 ports 2 × USB 2.0 ports.
GPIO:	Standard 40-pin GPIO header (fully backwards-compatible with previous boards)

Basic Raspberry Pi – GPIO Pins

- GPIO: a physical interface between the Raspberry Pi and the outside world
- To monitor and control the sensors and actuators
- 5 parts of GPIO pins:

  - GPIO: Input/output pins
  - Ground: Zero volts
  - 3.3v: These pins provide 3.3V power
  - 5v: These pins provide 5V power
  - ID EEPROM: Advanced use only

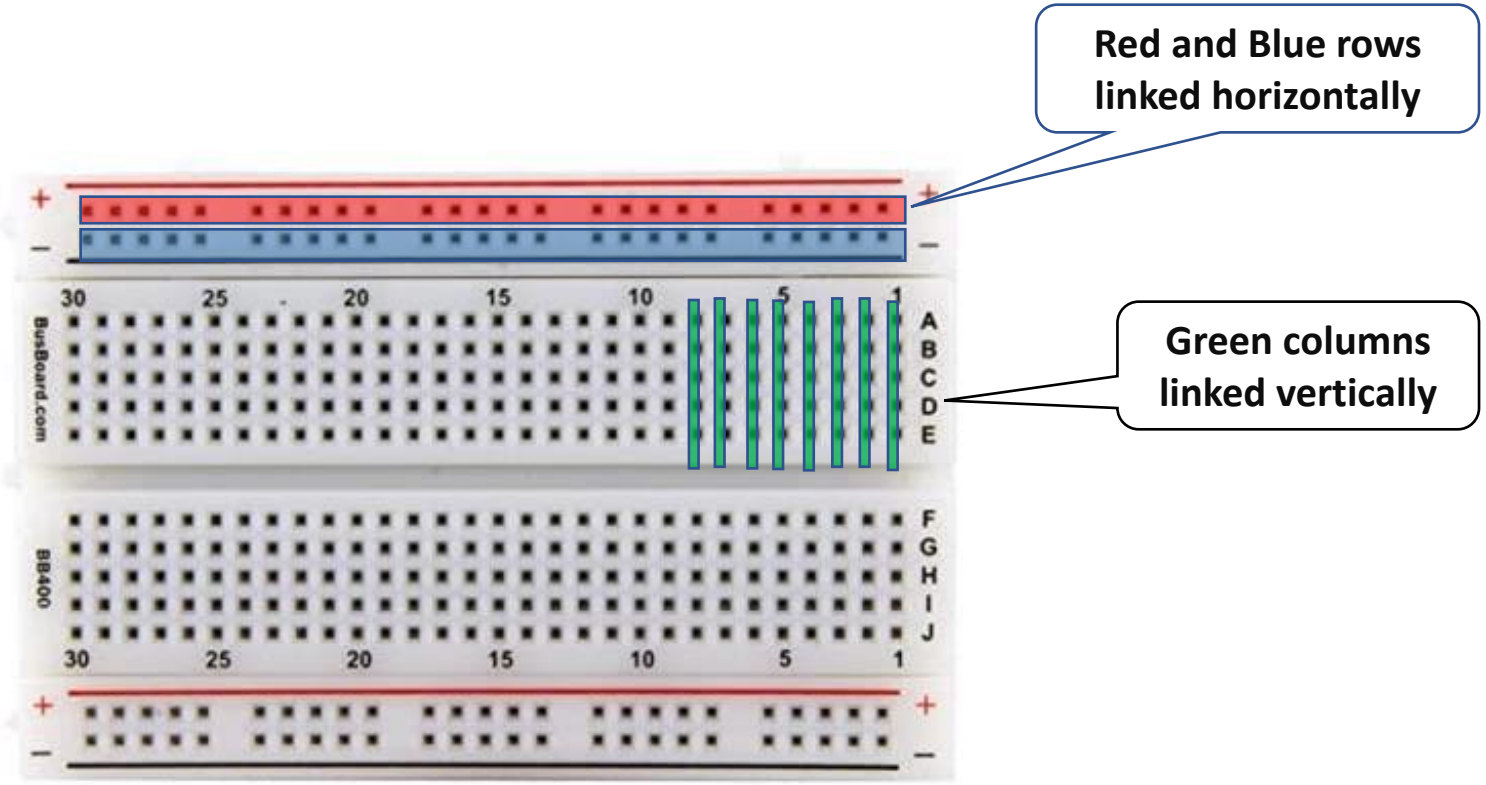


Basic Raspberry Pi – Connecting Raspberry Pi

- Using micro HDMI ports
  - Connect the Micro HDMI into LCD Monitor
- Using network
  - By wire/wireless
  - Connect by using SSH or VNC

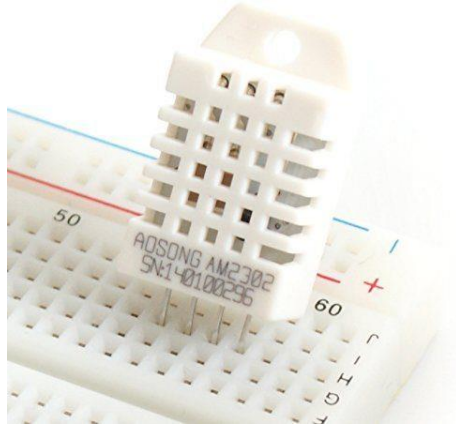
Basic Raspberry Pi – Breadboard

- 3 parts of breadboard:
  - The red line or positive: commonly used for Power
  - The blue line or negative: commonly used for Ground
  - The main part in the middle: to put and assemble your sensors or actuators



Basic Raspberry Pi – Sensors

- What are sensors
  - Add almost-human sensing capabilities
  - Take real-world events
  - Convert them to analogue or digital signals
  - Read by Raspberry Pi



Basic Raspberry Pi – Sensors

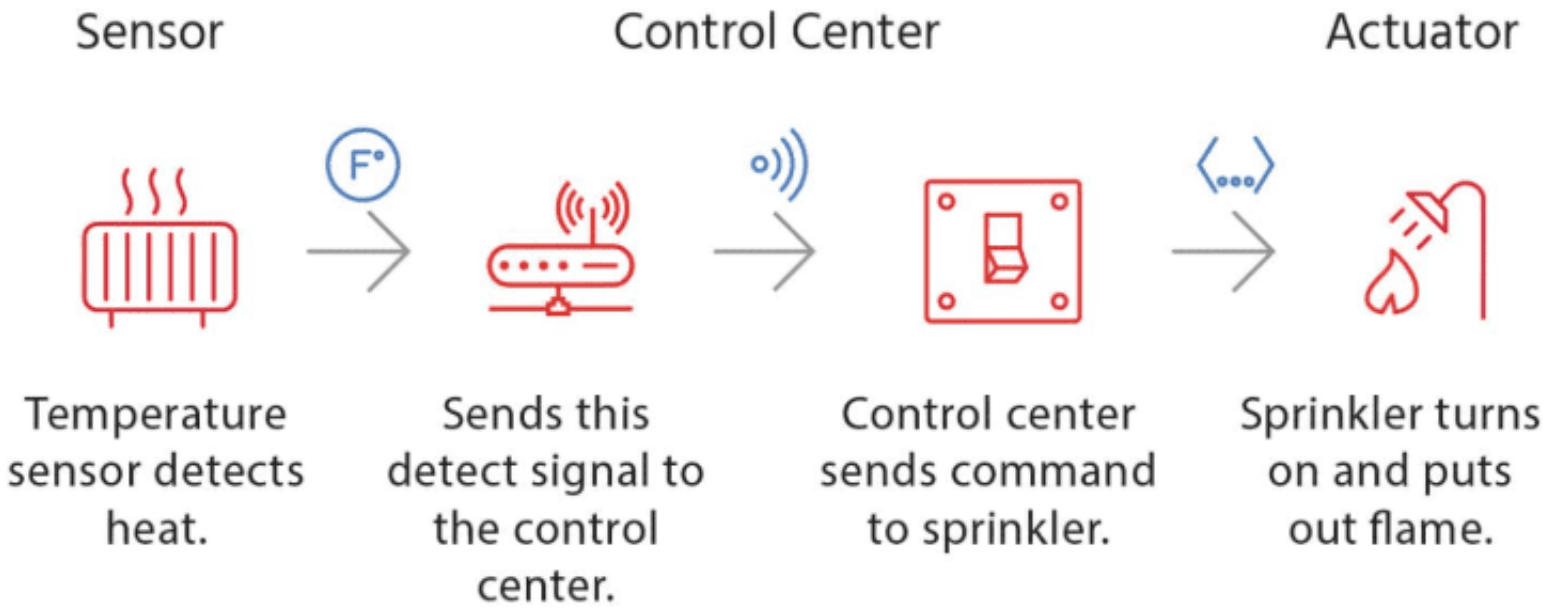
- Sensor categories
  - Temperature / Humidity / Air Pressure / Gas
  - Motion Sensors
  - Navigation Modules
  - Wireless / Infrared (IR) / Bluetooth
  - Analogue Sensors
  - Current Supply
  - Other Modules, Components and Sensors

Basic Raspberry Pi – Actuators

- What are actuators
  - Convert an electrical signal into a corresponding physical quantity
    - Example: movement, force, sound etc.
  - Controlled by Raspberry Pi

Basic Raspberry Pi – Sensors vs Actuators

- Difference in sensors and actuators
  - Sensors: read and get the information from sensors
  - Actuators: write and control some tools based on the previous information



Basic Raspberry Pi – Sensors and Actuators How to get it ?

- Borrow from us
  - We have a limited number of sensors and actuators
- Buy it by yourself
  - Save the receipt and reimburse to us
  - The limit of amount to reimburse : 1,000 NTD per team
  - The receipts should show the following title or number
    - 抬頭 : 國立陽明交通大學
    - 統編 : **87557573**

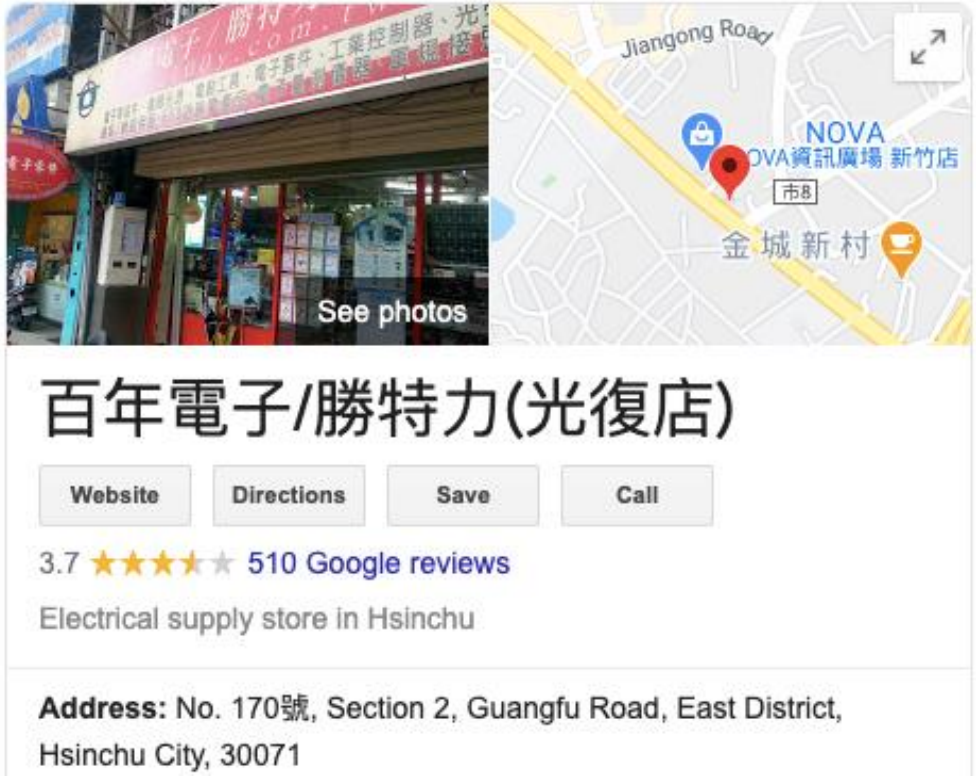
Basic Raspberry Pi – Sensors and Actuators
Our Collections of IoT Equipments

Item	Total	Note
Mandatory Components		
Raspberry Pi	37	Tiny computer that can be used as the main IoT system
Power Supply	37	To supply power to Raspberry Pi
SD Card	37	OS and storage holder
SD Card Reader	13	Converter from SD Card to micro SD
Sensors		
BME680	1	Measures the temperature & humidity
DHT 11	28	
HW484	1	Sound Detector
FC22	1	Gas Leakage Detection
IR-08H	2	Object Detection
MH Flying Fish	1	
SR04	8	Measure the Distance
LDR	61	Measures the Light Intensity
PIR	20	Motion Detection
Water Sensor	1	Detect the presence of water
Soil Sensor	3	Measure the amount of water in the soil
Rain Sensor	1	Switching device when rain is detected
Touch Sensor	1	When touch is closed switch, untouch is open switch
Raspberry Pi Cam	5	Capturing any activities

Item	Total	Note
Sensors		
Mic	1	Capturing any sound
RFID Reader	2	A device used to gather information from an RFID tag
RFID Tags	3	RFID tags are a type of tracking system
Push ON button	17	Switching device, closed circuit when pushed, open circuit when unpushed
Switch	7	ON/OFF Switch
Keypad	2	Record number that is inputed
Actuators		
Display	8	To display any infromation given
Number Display	10	To display number information
Buzzer	20	To give ring sound
Fan	9	Small fan to cool small area
LED	50	To provide lighting
Other Components		
Bread Board	19	To help developing IoT environment
Relay	5	Controls the opening and closing of the circuit
Potentiometer	2	Measures the distance or displacement of an object in a linear
Capacitor	50	Device that stores electrical energy in an electric field
Resistor	50	Limit the flow of electric current
Battery	10	To give power

Basic Raspberry Pi – Sensors and Actuators

- Where to buy the sensors and actuators?



OS Installation – Things you need at first

Make sure you already get all of them below:

- Raspberry Pi 4 Model B
- USB type-C power supply
- microSD card

Something you also need:

- card reader (for microSD)
- network cable (Ethernet RJ45)
- laptop or PC

OS Installation – Download the OS

- Download Raspbian from here:

https://www.raspberrypi.org/downloads/raspbian/

- Choose the version you like and unzip the .zip file

- Here, we choose **Raspberry Pi OS (32-bit) with desktop and recommended software** since we have 32GB SD card

OS Installation – Tools to Flash the OS (1)

- Right now pi 4b only supports booting from SD card, so we need to download a tool to flash OS image to SD card

- Rufus (Windows only) or balenaEtcher (Windows / macOS / Linux)

- Flash .img file into your SD card

(You need an SD card reader to help you complete this step.)

OS Installation - Tools to Flash the OS (2)

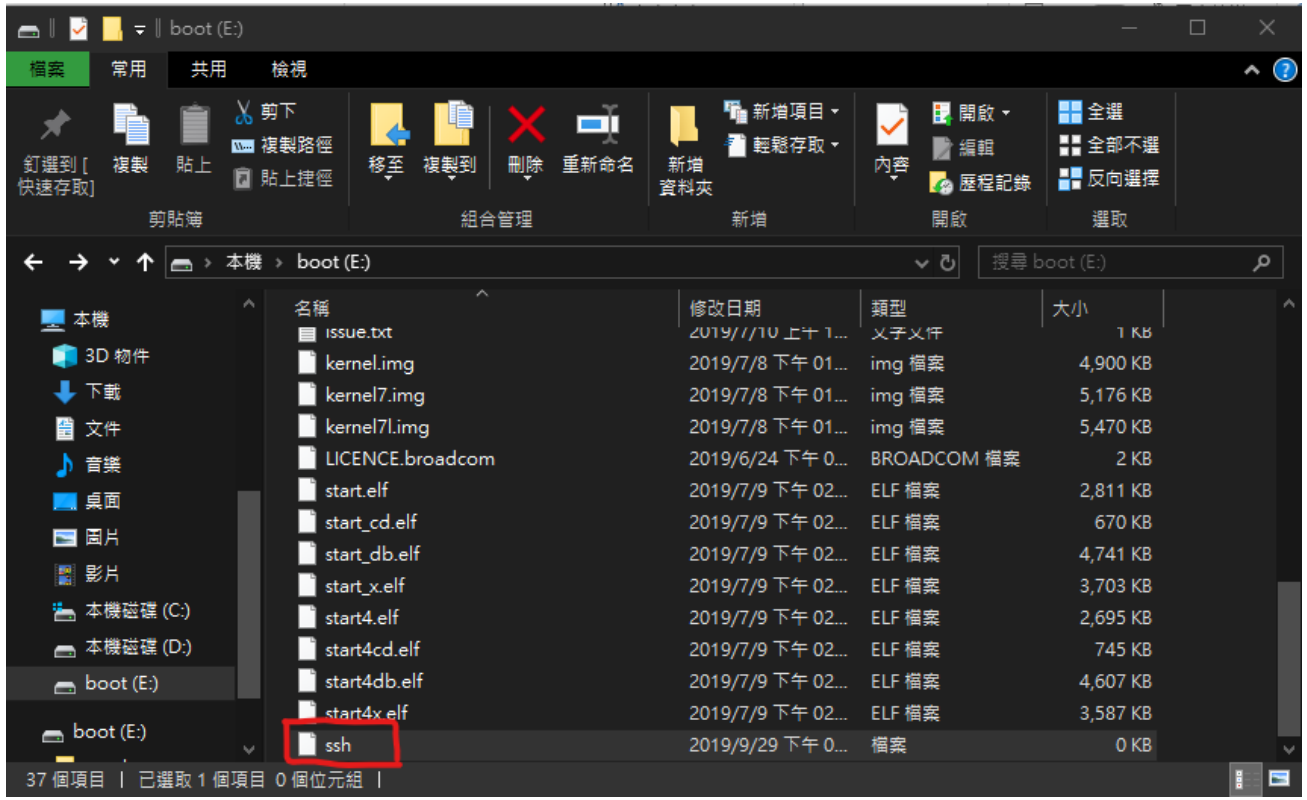
OS Installation – SSH (1)

- After process completes, add a **new raw file** called “ssh” into the “boot” disk.

Check:

Plug the microSD card into pi 4b, and connect type-C power cable and network cable.

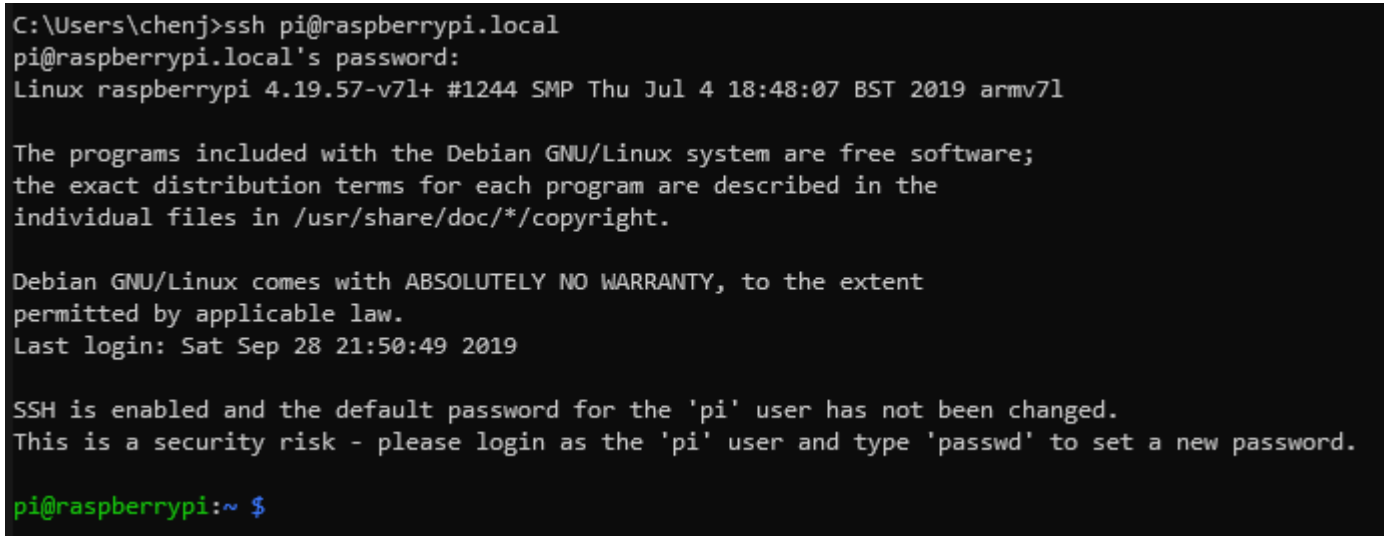
If green light is twinkling under the left corner of network cable slot, that means your pi 4b is using SSH now!!



OS Installation – SSH (2)

- There is a built-in tool for ssh in Windows 10. But if you cannot find it, you need to download PuTTY here.

- Use ssh command
  “ssh pi@raspberrypi.local”
  default password: raspberry



OS Installation – Enable VNC server

- **sudo raspi-config**

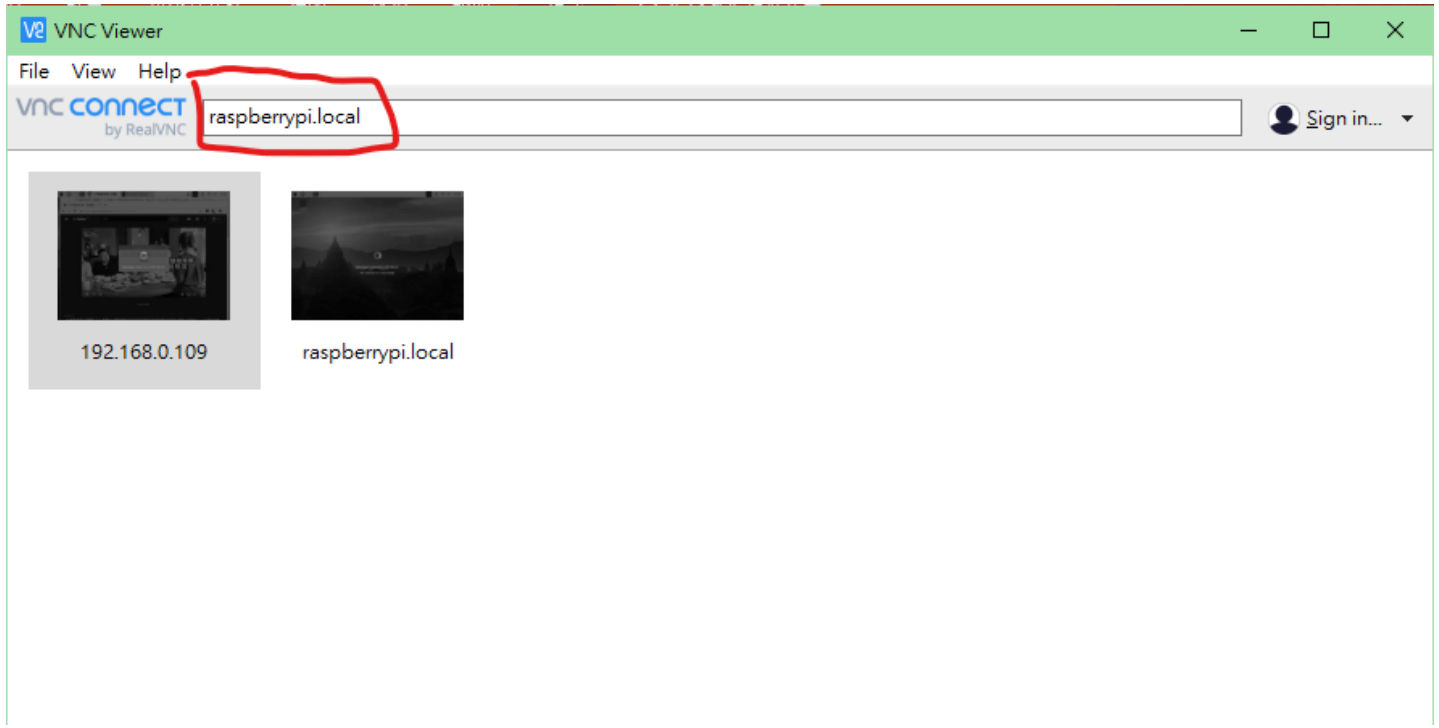
Step 1. Choose 5 Interfacing Options -> P3 VNC -> Yes(是)

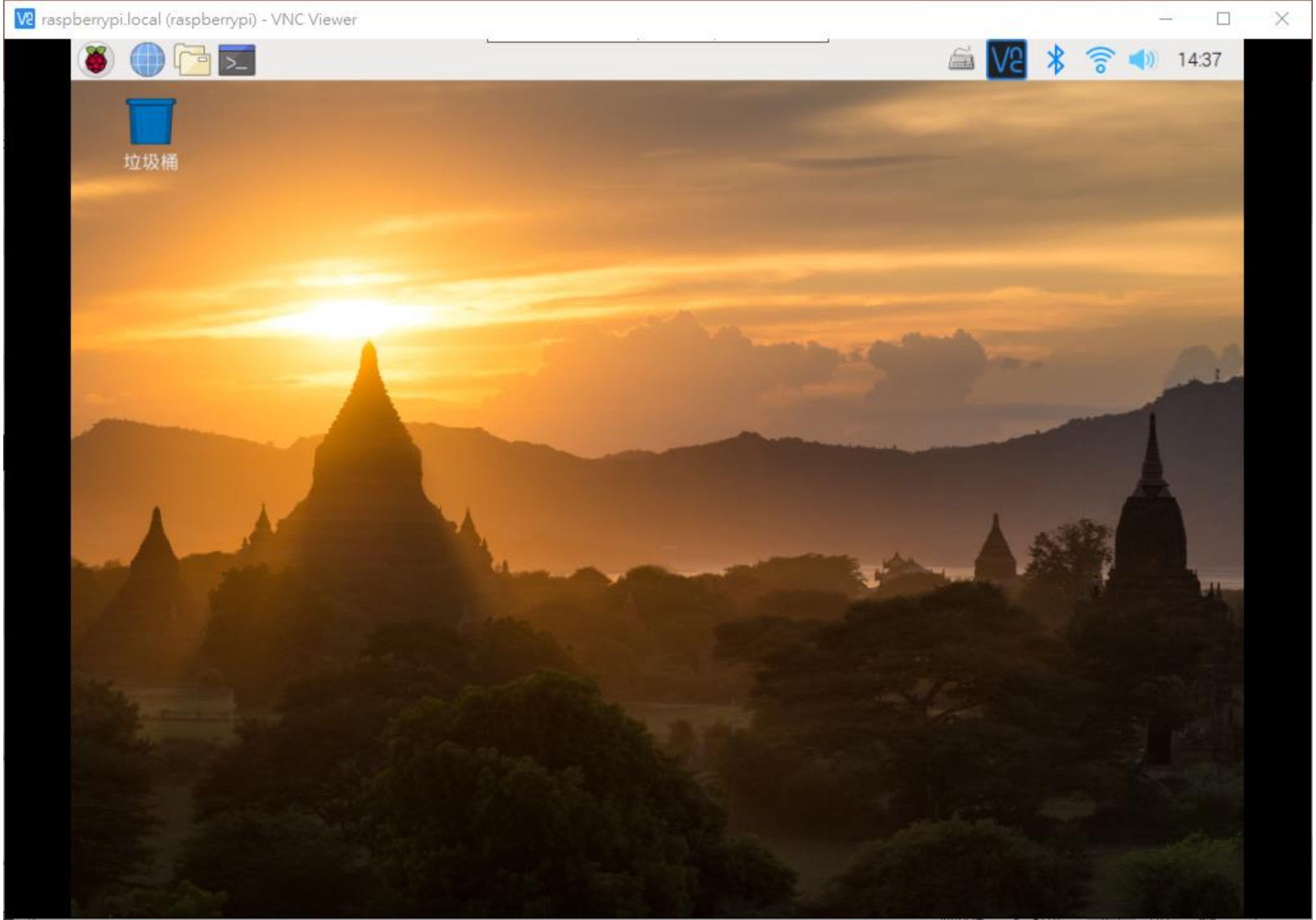
Step 2. Choose 7 Advanced Options -> A5 Resolution -> choose one other than Default

You need to reboot the system after the setting!

OS Installation – VNC client

- UltraVNC (Windows only) or RealVNC (Windows / macOS / Linux)
- Connect to the VNC Server “raspberrypi.local”




raspberrypi.local (raspberrypi) - VNC Viewer
14:37
垃圾桶

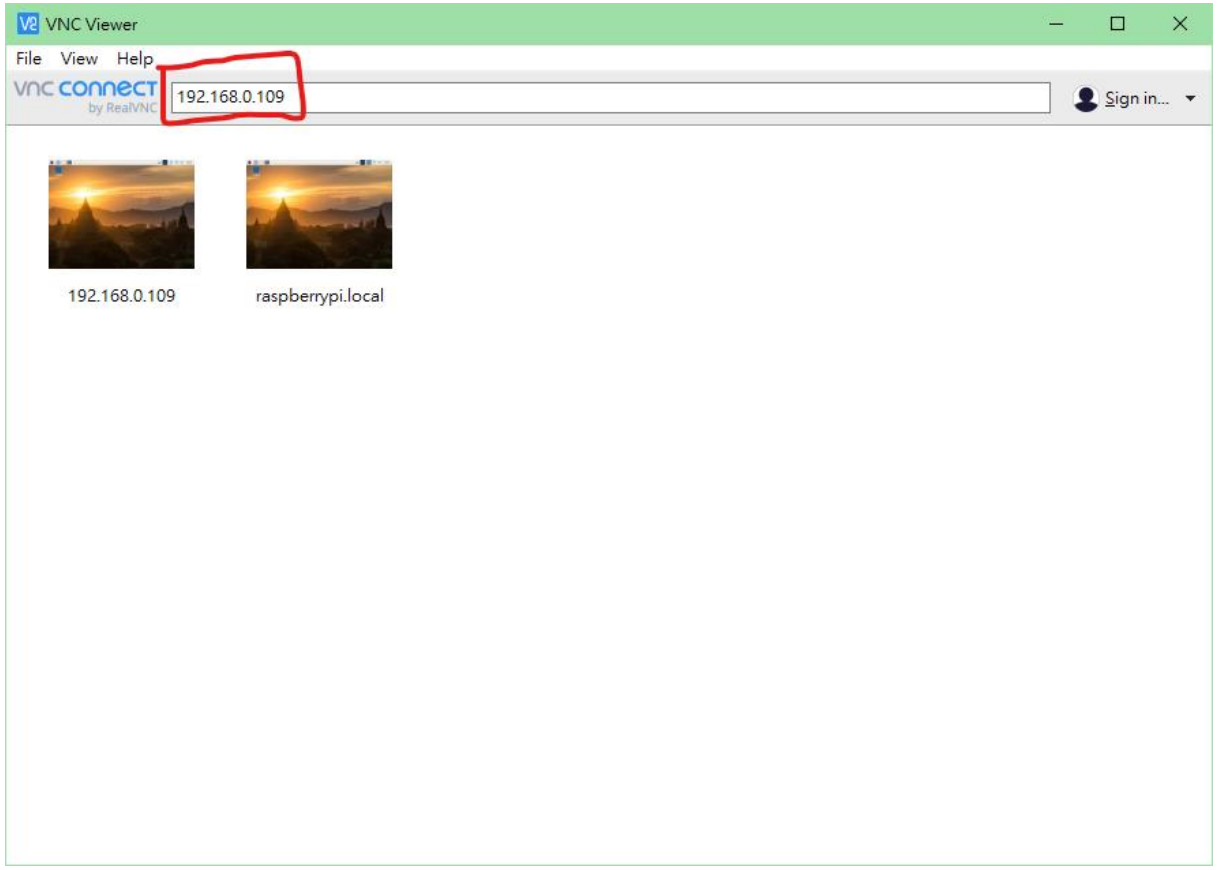
OS Installation - Notes

After you configure WiFi connection on Pi 4b, you can use VNC connect to Pi 4b without network cable.

Use command **ifconfig** to find what is the ip address on wlan.

OS Installation - Notes

```
pi@raspberrypi:~ $ ifconfig
eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 169.254.98.144  netmask 255.255.0.0  broadcast 169.254.255.255
        inet6 fe80::1b0b:8f29:bb0b:9fc2  prefixlen 64  scopeid 0x20<link>
        ether dc:a6:32:11:24:19  txqueuelen 1000  (Ethernet)
        RX packets 2722  bytes 303582 (296.4 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 3195  bytes 2441752 (2.3 MiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 17  bytes 1004 (1004.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 17  bytes 1004 (1004.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

wlan0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.0.109  netmask 255.255.255.0  broadcast 192.168.0.255
        inet6 fe80::b914:d332:250f:226c  prefixlen 64  scopeid 0x20<link>
        ether dc:a6:32:11:24:1a  txqueuelen 1000  (Ethernet)
        RX packets 1451  bytes 255163 (249.1 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 110  bytes 11534 (11.2 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0
```



OS Installation - Notes

If you re-install the Raspberry Pi again, maybe you will encounter some problem when you use SSH command.

Try to delete the “known_hosts” file or just delete the line related to “raspberrypi.local” in the known_hosts and use ssh command again!

OS Installation - Notes

```
C:\Users\chenj>ssh pi@raspberrypi.local
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@       WARNING: POSSIBLE DNS SPOOFING DETECTED!          @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
The ECDSA host key for raspberrypi.local has changed,
and the key for the corresponding IP address fe80::1b0b:8f29:bb0b:9fc2%7
is unknown. This could either mean that
DNS SPOOFING is happening or the IP address for the host
and its host key have changed at the same time.
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
@    WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED!     @
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
IT IS POSSIBLE THAT SOMEONE IS DOING SOMETHING NASTY!
Someone could be eavesdropping on you right now (man-in-the-middle attack)!
It is also possible that a host key has just been changed.
The fingerprint for the ECDSA key sent by the remote host is
SHA256:XgkT5//45sRrCkA2An2Jy/4g4w3HHDiWRYao4YbJTCo.
Please contact your system administrator.
Add correct host key in C:\\Users\\chenj/.ssh/known_hosts to get rid of this message.
Offending ECDSA key in C:\\Users\\chenj/.ssh/known_hosts:11
ECDSA host key for raspberrypi.local has changed and you have requested strict checking.
Host key verification failed.

C:\Users\chenj>rmdir /S .ssh
.ssh, 您確定要執行嗎 (Y/N)? Y
```

known_hosts (~\.ssh) - VIM

```
raspberrypi.local,fe80::1b0b:8f29:bb0b:9fc2%7 ecdsa-sha2-nistp256 AAAAE2VjZHNhLXNoYTItbmlzdHAyNTYAAAAIbmlzdHAyNTYAAABBBN
rEmKUrPo7rdLuobyyowUBlAqkKUuQjV4wzQzWmgA5KATGYTSIhGYQxlTjWOkd6N4jzqYvfd3VPNNwQRrxlrjE=
~
".ssh\known_hosts" 1L, 208C                                  1,1         全部
```

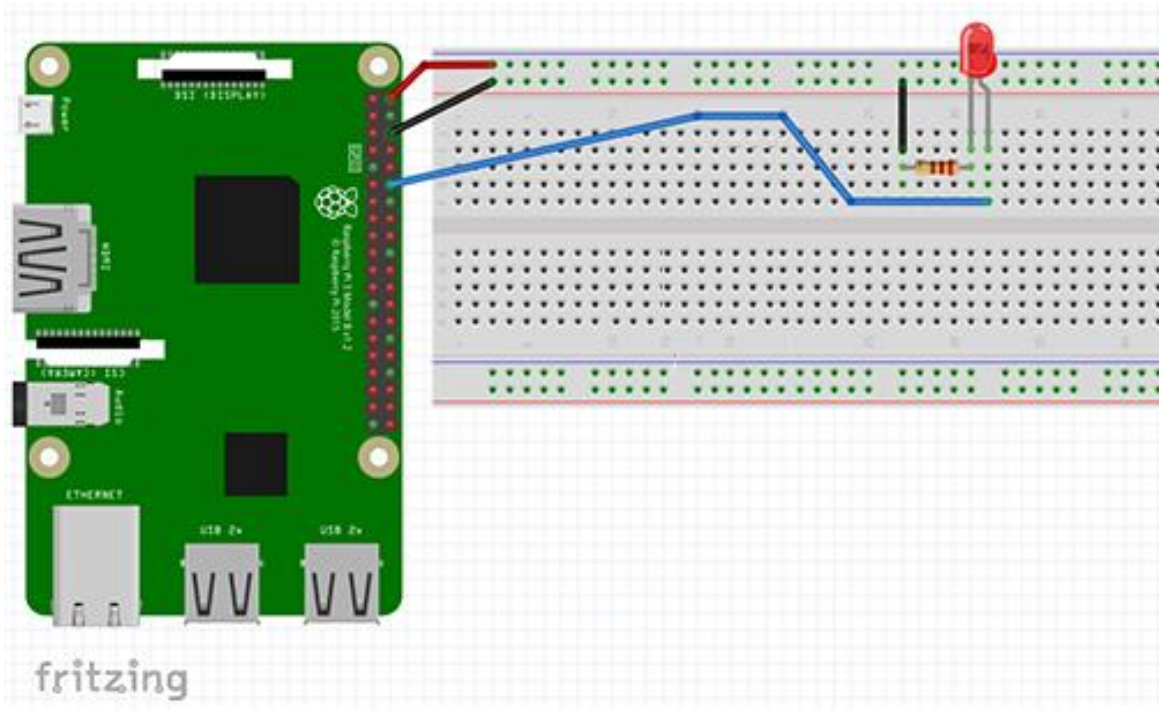
A Simple IoT Application

Controlling LED with a switch, an LDR sensor, and a DHT sensor

Controlling LED with Raspberry Pi

Components :

- LED
- A Resistor (Orange, Orange, Brown, Gold)



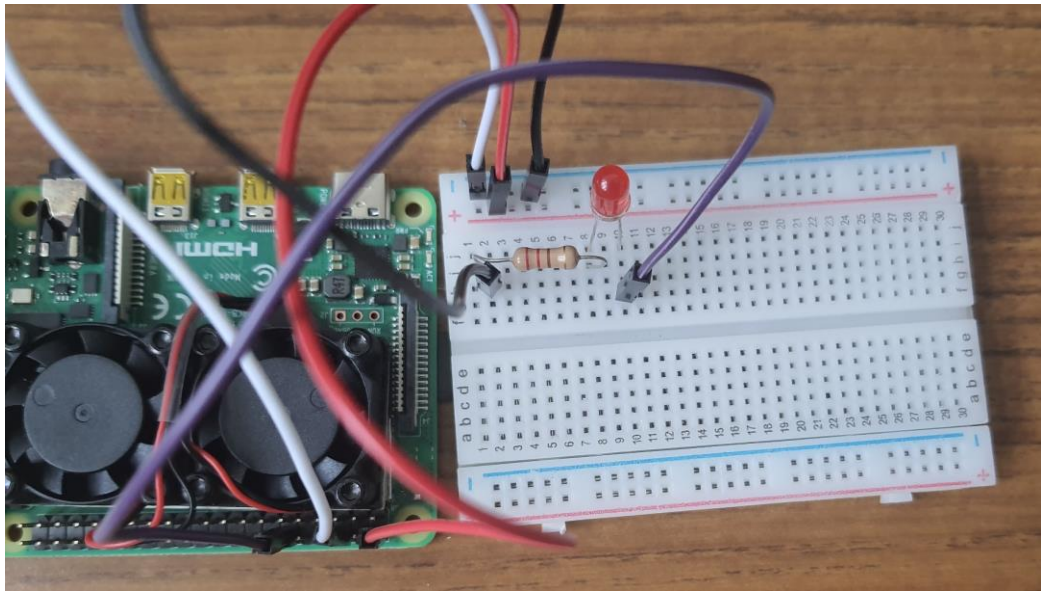

```
import RPi.GPIO as GPIO
import time

GPIO.setmode(GPIO.BOARD)
GPIO.setwarnings(False)

ledPin = 12

GPIO.setup(ledPin, GPIO.OUT)

for i in range(100):
        print("LED turning on.")
        GPIO.output(ledPin, GPIO.HIGH)
        time.sleep(1)
        print("LED turning off.")
        GPIO.output(ledPin, GPIO.LOW)
        time.sleep(1)
```

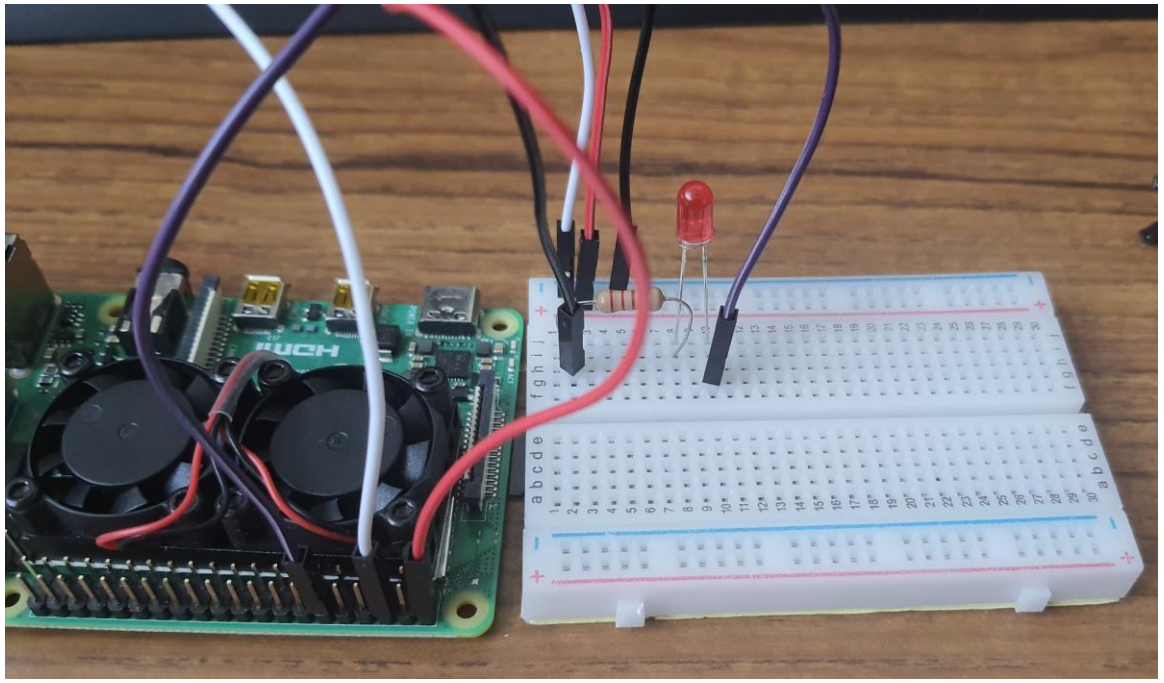
There are two different models of GPIO.setmode (pin numbering)

- GPIO.BOARD : using board numbering system (ex: pin 12)
- GPIO.BCM : using BCM numbers (ex : GPIO 18)

Controlling LED with Raspberry Pi

Demo video:
https://youtu.be/77u4gbw1fVw

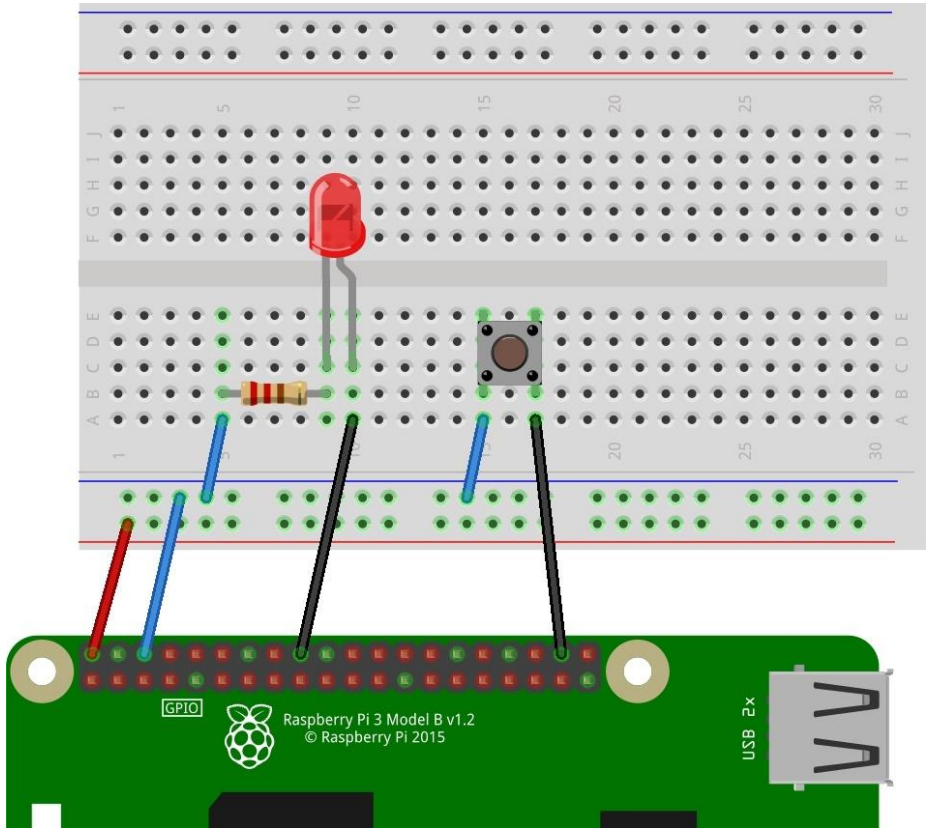




Controlling LED with Switch

Components :

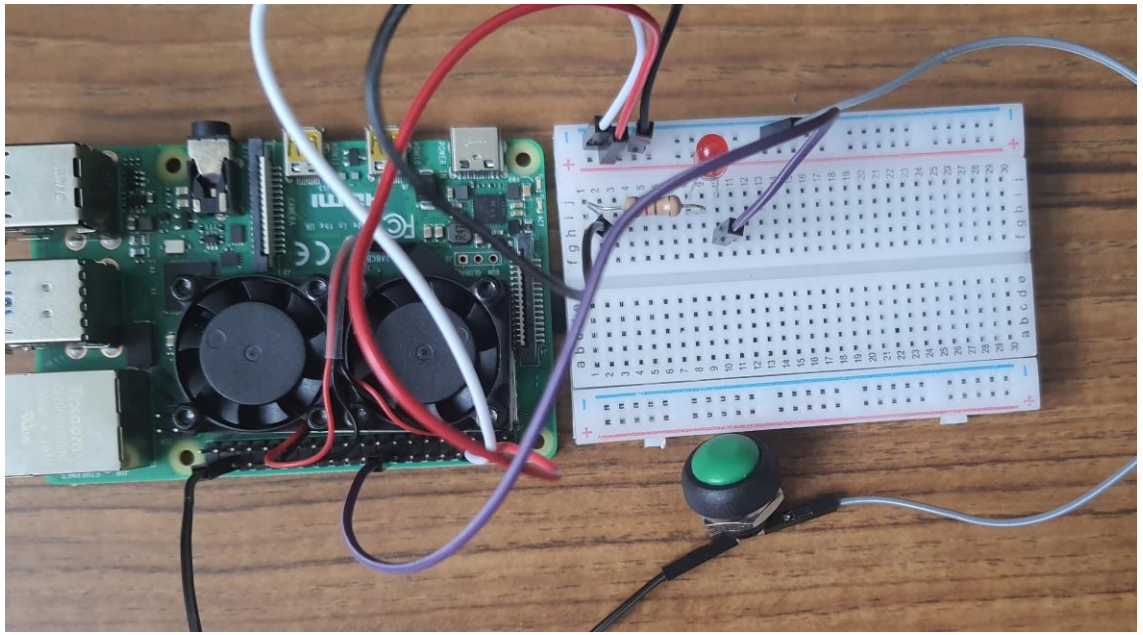
- LED
- Switch
- 1 Resistors (Orange, Orange, Brown, Gold)

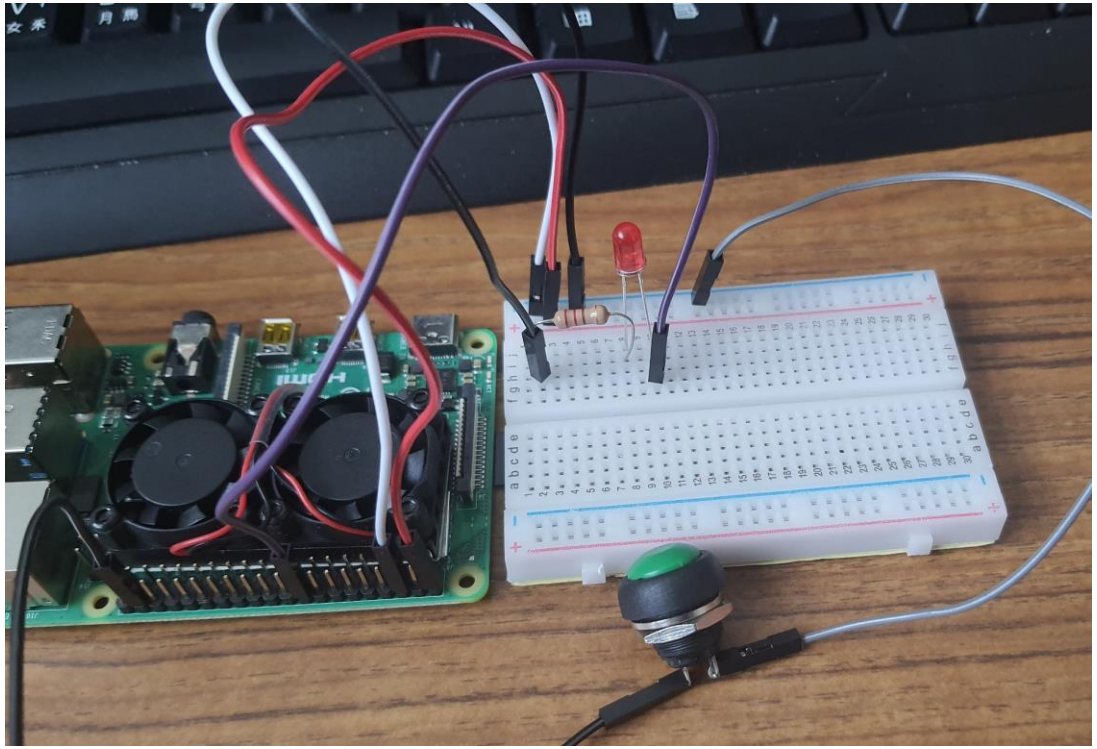



```
import RPi.GPIO as GPIO
import time

GPIO.setmode(GPIO.BCM)

GPIO.setup(20, GPIO.IN, pull_up_down=GPIO.PUD_UP)#Button to GPIO20
GPIO.setup(24, GPIO.OUT)  #LED to GPIO24

try:
    while True:
         button_state = GPIO.input(20)
         if button_state == False:
             GPIO.output(24, True)
             print('Button Pressed...')
             time.sleep(0.2)
         else:
             GPIO.output(24, False)
except:
    GPIO.cleanup()
```

Controlling LED with Switch

Demo video:
https://youtu.be/M9R75bi-ahA

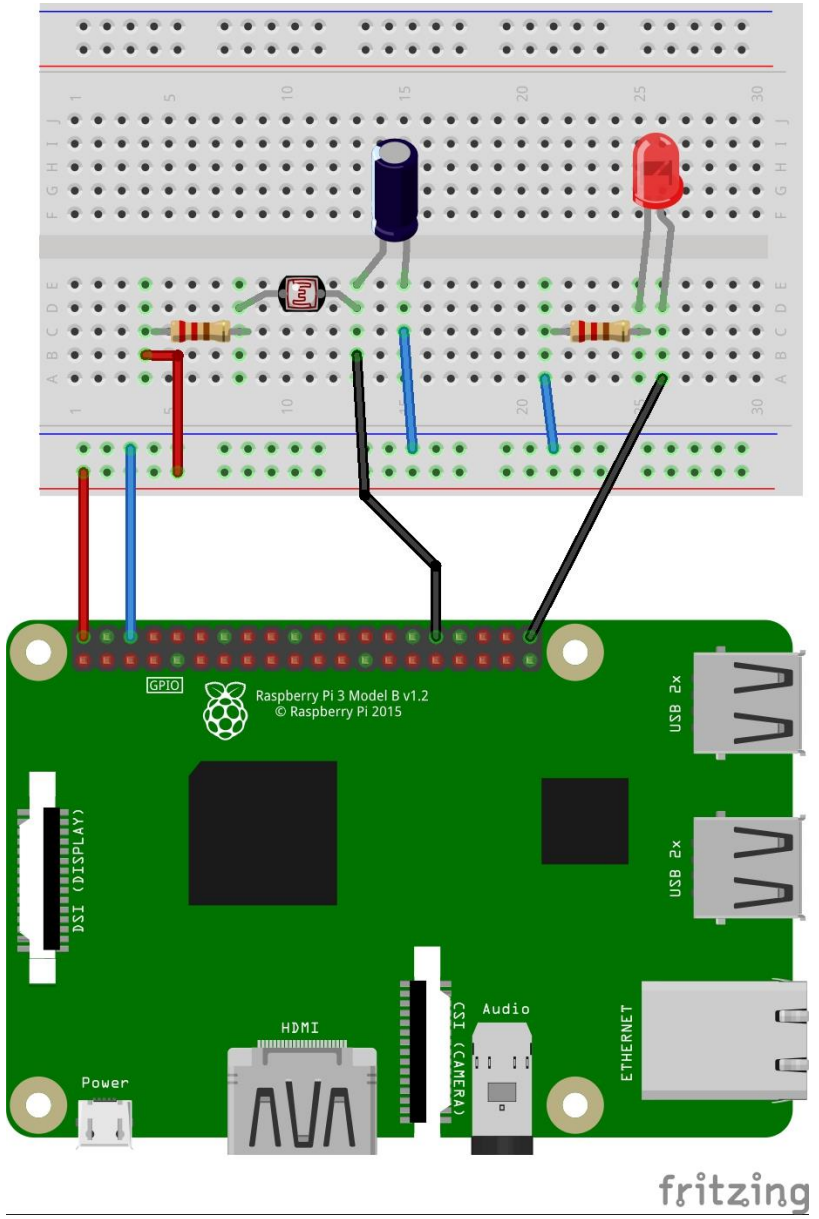




Controlling LED with LDR Sensor

Components :

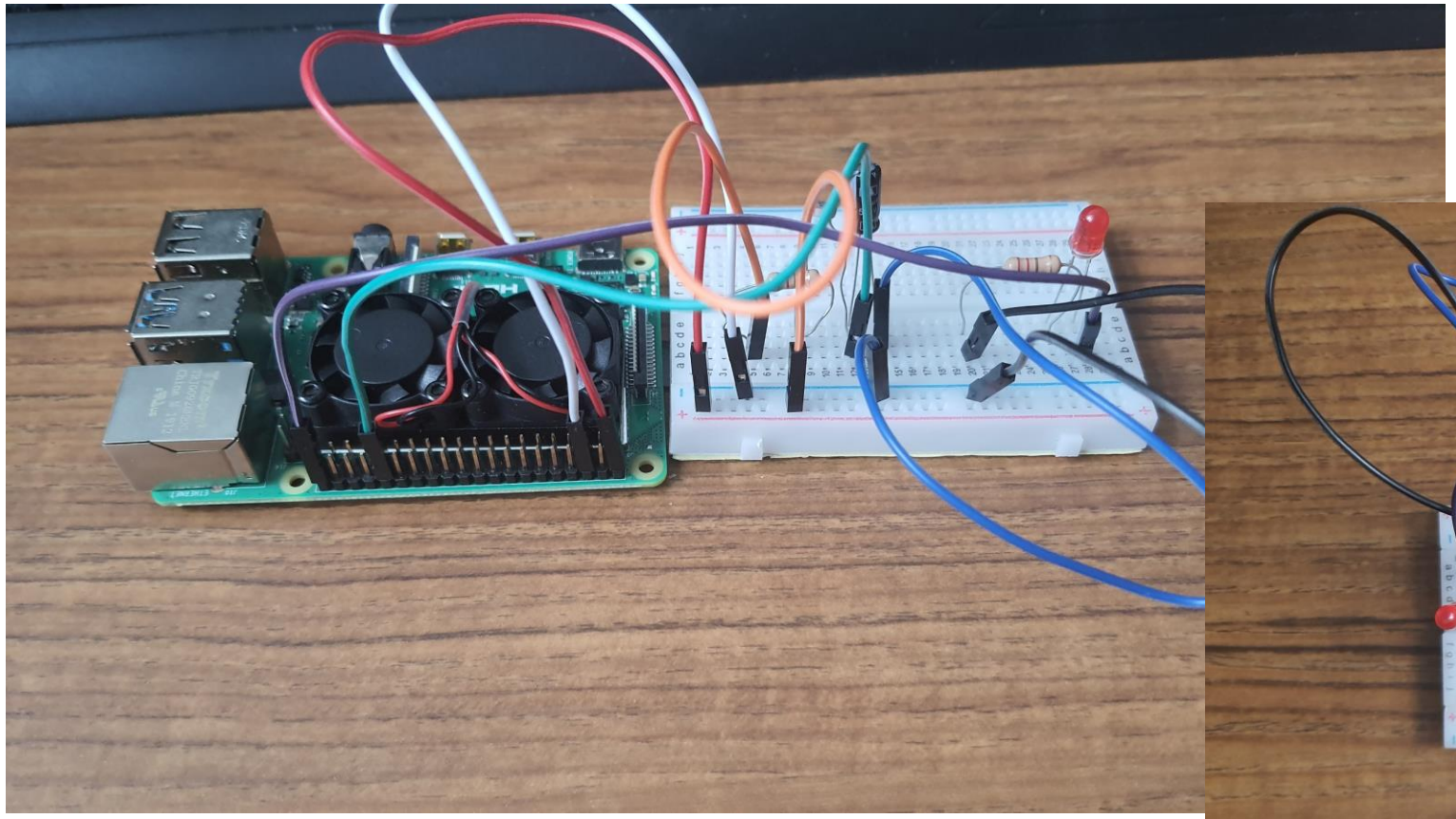
- LED
- LDR (Light Dependent Resistor)
- Capacitor 1µF
- 2 Resistors (Orange, Orange, Brown, Gold)

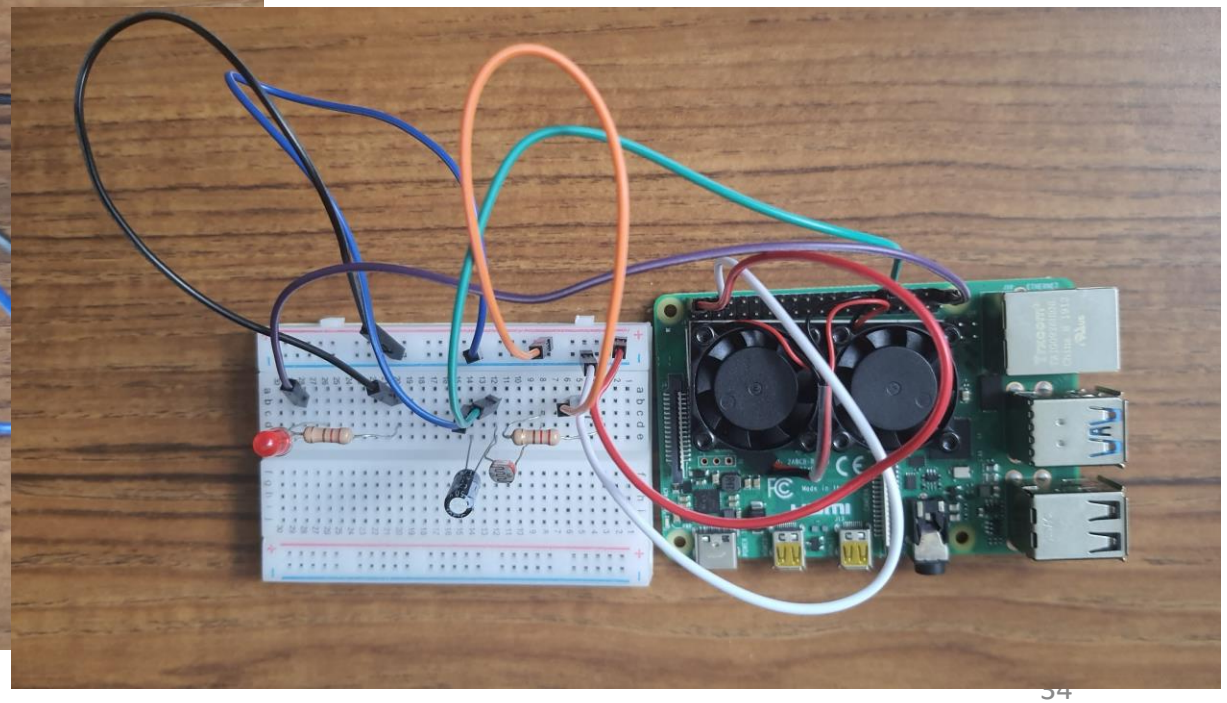



```
import RPi.GPIO as GPIO
import time

GPIO.setmode(GPIO.BCM)
ldr_threshold = 30000
LDR_PIN = 12
LIGHT_PIN = 21

def readLDR(PIN):
    reading=0
    GPIO.setup(PIN, GPIO.OUT)
    GPIO.output(PIN, False)
    time.sleep(0.1)
    GPIO.setup(PIN, GPIO.IN)
    while (GPIO.input(PIN)==False):
        reading=reading+1
    return reading

def switchOnLight(PIN):
    GPIO.setup(PIN, GPIO.OUT)
    GPIO.output(PIN, True)

def switchOffLight(PIN):
    GPIO.setup(PIN, GPIO.OUT)
    GPIO.output(PIN, False)


while True:
    try:
        ldr_reading = readLDR(LDR_PIN)
        print(ldr_reading)
        if ldr_reading > ldr_threshold:
            switchOnLight(LIGHT_PIN)
        else:
            switchOffLight(LIGHT_PIN)
        time.sleep(1)
    except KeyboardInterrupt:
        exit()
```

Controlling LED with LDR Sensor

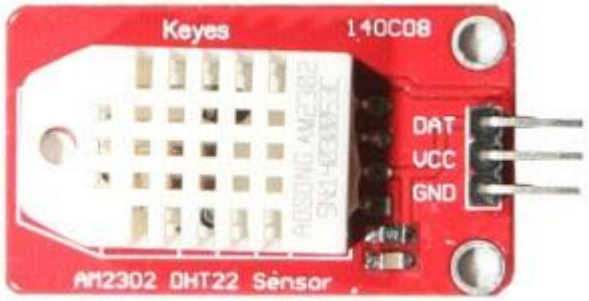
Demo video:
https://youtu.be/uyhHY2lFaq4





Controlling LED with DHT Sensor

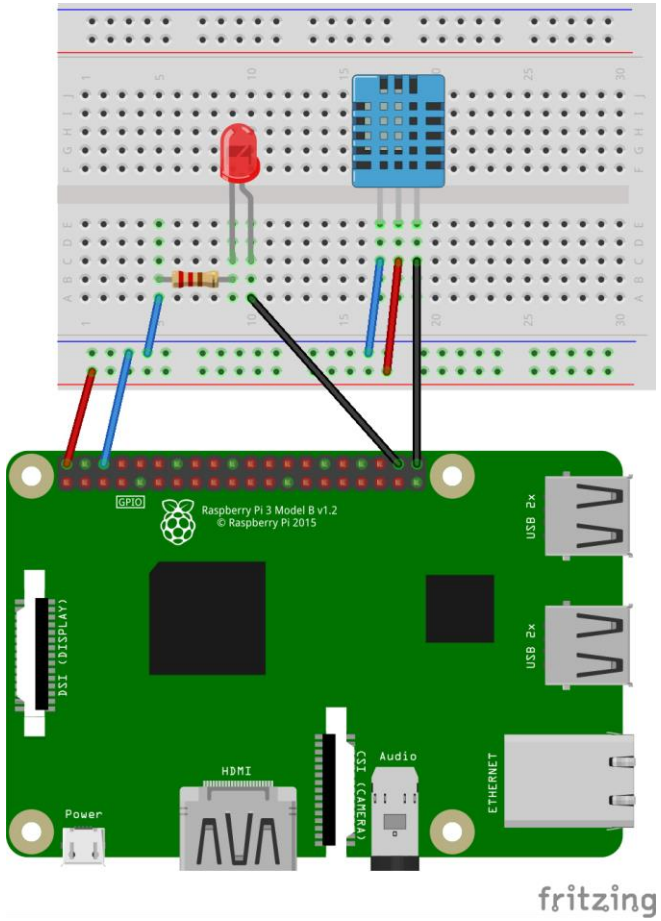
- Components :
  - DHT11 or DHT22 Sensor
    - VCC (+)
    - GND (-)
    - DAT (data)



- Install some libraries
  - sudo apt-get update
  - sudo apt-get upgrade
  - sudo apt-get install python3-dev python3-pip
  - sudo python3 -m pip install --upgrade pip setuptools wheel
  - sudo pip3 install Adafruit_DHT

Controlling LED with DHT Sensor

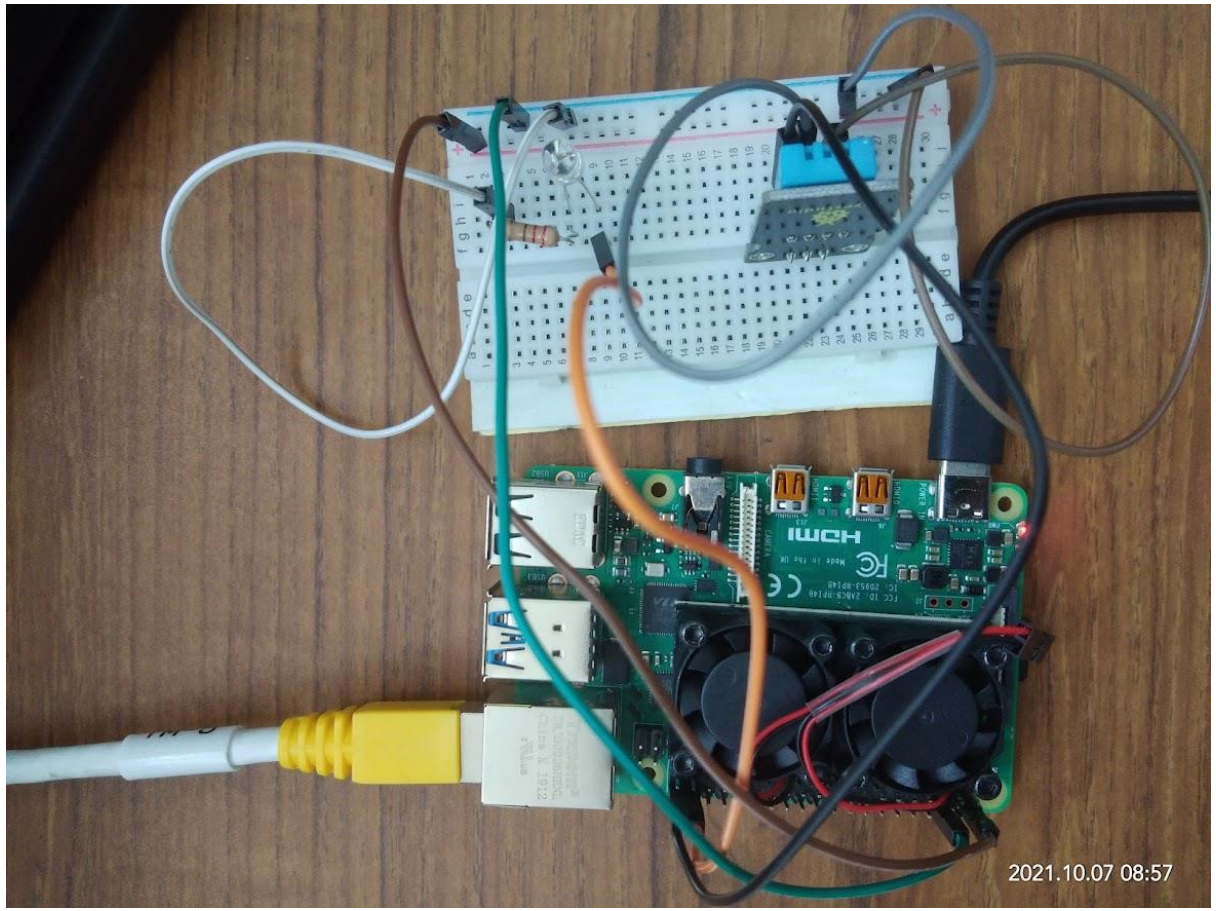
The circuit:
DAT : to GPIO
VCC : to Power
GND : to Ground




```
import Adafruit_DHT
import RPi.GPIO as GPIO
import time

GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)
DHT_SENSOR = Adafruit_DHT.DHT11
DHT_PIN = 21

humidity_threshold = 65
LIGHT_PIN = 20

def switchOnLight(PIN):
    GPIO.setup(PIN, GPIO.OUT)
    GPIO.output(PIN, True)

def switchOffLight(PIN):
    GPIO.setup(PIN, GPIO.OUT)
    GPIO.output(PIN, False)

while True:
    try:
        humidity, temperature = Adafruit_DHT.read_retry(DHT_SENSOR, DHT_PIN)
        if humidity is not None and temperature is not None:
            print("Temp={0:0.1f}*C  Humidity={1:0.1f}%".format(temperature, humidity))
        else:
           print("Failed to retrieve data from humidity sensor")

        if humidity > humidity_threshold:
            switchOnLight(LIGHT_PIN)
            print("SWITCH ON LIGHT")
        else:
            switchOffLight(LIGHT_PIN)

        time.sleep(1)
    except KeyboardInterrupt:
        exit()
```

Controlling LED with DHT Sensor

Demo video:
https://youtu.be/0fJ5V0zgsNo



Controlling LED with DHT Sensor Troubleshoot

- If you find an error like this one:

```
pi@raspberrypi:~/Documents $ python3 temperature.py
Traceback (most recent call last):
  File "temperature.py", line 8, in <module>
    humidity, temperature = Adafruit_DHT.read_retry(DHT_SENSOR, DHT_PIN)
  File "/usr/local/lib/python3.7/dist-packages/Adafruit_DHT/common.py", line 94,
 in read_retry
    humidity, temperature = read(sensor, pin, platform)
  File "/usr/local/lib/python3.7/dist-packages/Adafruit_DHT/common.py", line 80,
 in read
    platform = get_platform()
  File "/usr/local/lib/python3.7/dist-packages/Adafruit_DHT/common.py", line 60,
 in get_platform
    from . import Beaglebone_Black
  File "/usr/local/lib/python3.7/dist-packages/Adafruit_DHT/Beaglebone_Black.py"
, line 24, in <module>
    from . import Beaglebone_Black_Driver as driver
ImportError: cannot import name 'Beaglebone_Black_Driver' from 'Adafruit_DHT' (/
usr/local/lib/python3.7/dist-packages/Adafruit_DHT/__init__.py)
```

Controlling LED with DHT Sensor Troubleshoot

- After system updates, the hardware name in the /proc/cpuinfo on raspberry pi4 has been changed.

So, it is necessary to edit a **platform_detect.py** in **/usr/local/lib/python3.7/dist-packages/Adafruit_DHT/** directory. All you need it is add next few strings in **def pi_version():** function:

Code: Select all

```
elif match.group(1) == 'BCM2711':
    return 3
```

Assignment 1 - Specification

- Objectives:
  - Connect and read data from sensors
  - Connect and write data to control actuators
- Upload to E3 before 10/27 23:59PM
  - Assignment 1 – deliverables
    - Report (2-4 pages) in PDF and use the template that we provide
      - Explain the objective
      - Explain your source code and the detail of how your script can read and write your sensors and actuators, respectively
      - Link to a 3-minute demo video on YouTube
    - Source code
  - 1-page project proposal in PDF
    - Topic, objective, and sensors/actuators
    - Specs for Assignments 1, 2, and 3 (grow your IoT application instead of changing it for each assignment: Assignments 2 needs to store data to the cloud, while Assignment 3 needs to have two devices and run a program in the cloud)
  - Zip the above 3 files into one compressed file and upload
- Q&A? Post on E3 discussion board

Assignment 1 - Specification

- Note for Assignment 1:
  - You must use different combinations of sensors and actuators than the ones we present
    - If using the same combinations, your maximum score is only 65.
  - The report can be written in Chinese or English, but the video must be delivered in English.
  - In video, explain how you assemble your sensors and actuators, how you connect it, and also explain your source code, and show the results.
  - Upload your video to YouTube and put the link into your report. Don’t upload your video to E3.
  - In your report, make sure you have a diagram of connected sensors and actuators on your board (you can refer the diagram on page 29 and you can use “Fritzing” to draw it).

Enjoy it.