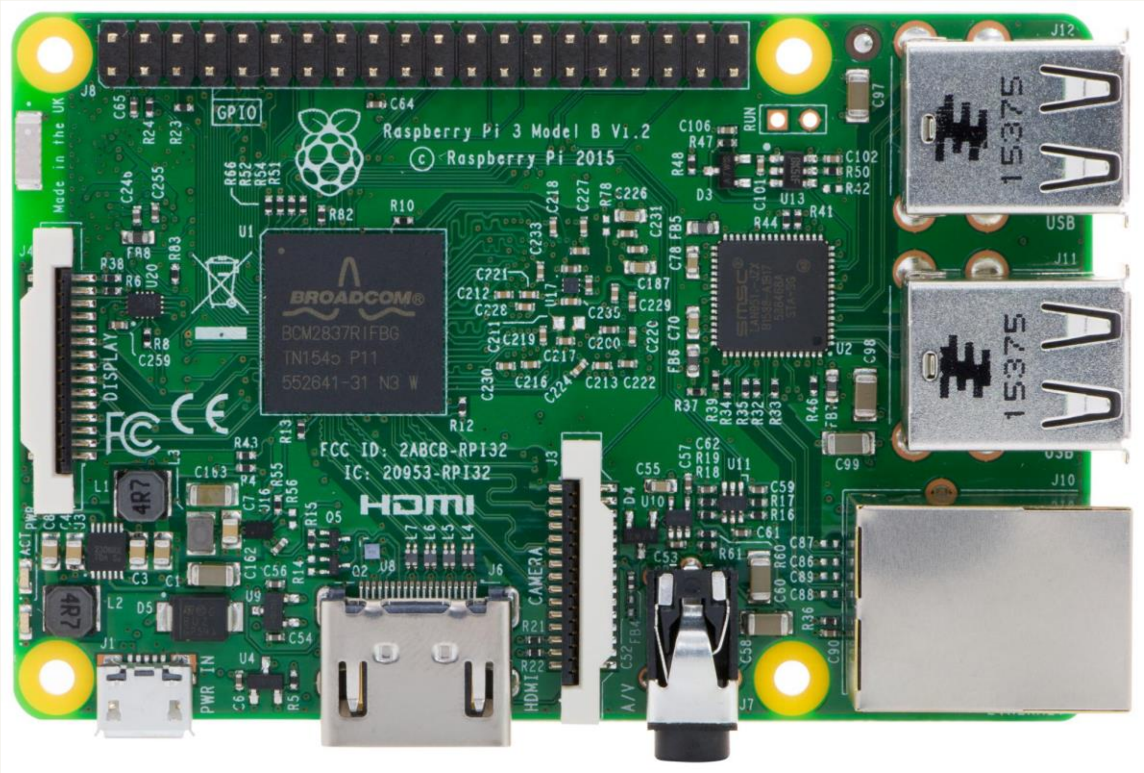
RASPBERRY PI

DAY 1

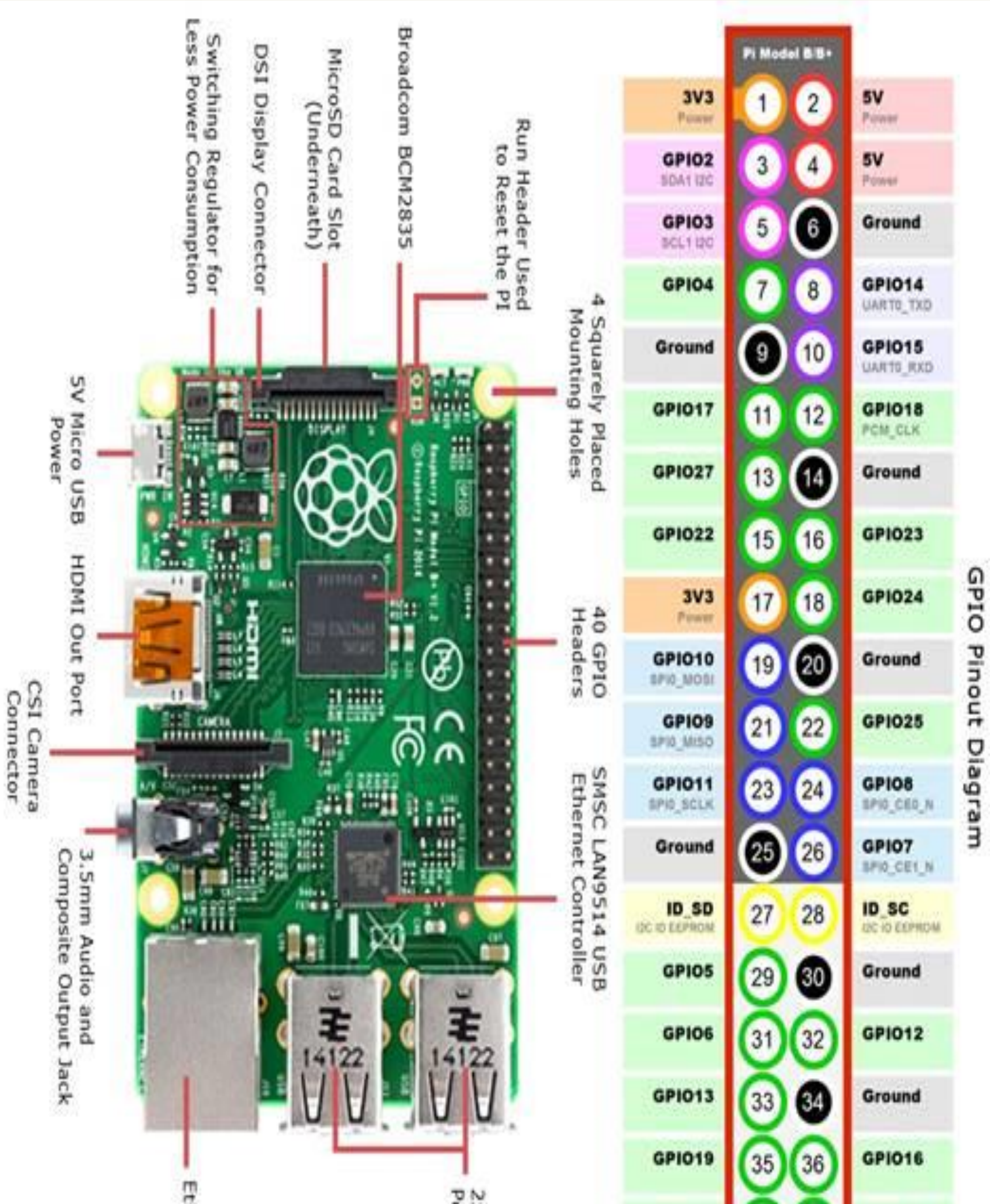
BY EMILY WENG

RASPBERRY PI?

- A credit card sized computer
- No monitor or keyboard/ mouse
- Only comes with a motherboard

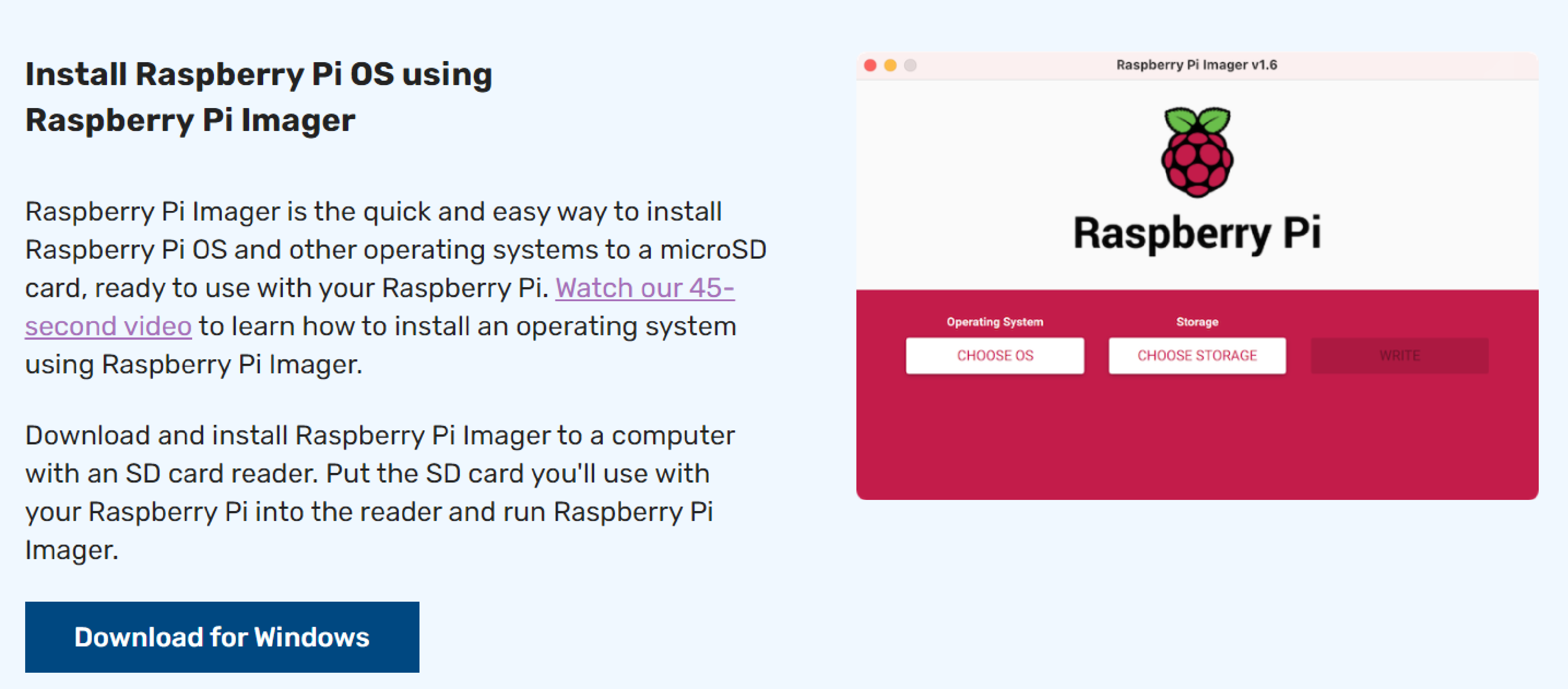


HARDWARE FO
MODEL B


Run Header Used to Reset the Pi
Broadcom BCM2835
MicroSD Card Slot (Underneath)
DSI Display Connector
Switching Regulator for Less Power Consumption
4 Squarely Placed Mounting Holes
5V Micro USB Power
HDMI Out Port
40 GPIO Headers
CSI Camera Connector
3.5mm Audio and Composite Output Jack
SMSC LAN9514 USB Ethernet Controller
2
P
Et
GPIO Pinout Diagram
Pi Model B/B+

3V3 Power	1	2	5V Power
GPIO2 SDA1 I2C	3	4	5V Power
GPIO3 SCL1 I2C	5	6	Ground
GPIO4	7	8	GPIO14 UART0_TXD
Ground	9	10	GPIO15 UART0_RXD
GPIO17	11	12	GPIO18 PCM_CLK
GPIO27	13	14	Ground
GPIO22	15	16	GPIO23
3V3 Power	17	18	GPIO24
GPIO10 SPI0_MOSI	19	20	Ground
GPIO9 SPI0_MISO	21	22	GPIO25
GPIO11 SPI0_SCLK	23	24	GPIO8 SPI0_CE0_N
Ground	25	26	GPIO7 SPI0_CE1_N
ID_SD I2C ID EEPROM	27	28	ID_SC I2C ID EEPROM
GPIO5	29	30	Ground
GPIO6	31	32	GPIO12
GPIO13	33	34	Ground
GPIO19	35	36	GPIO16

MAIN OPERATING SYSTEM THAT WILL BE USED TODAY

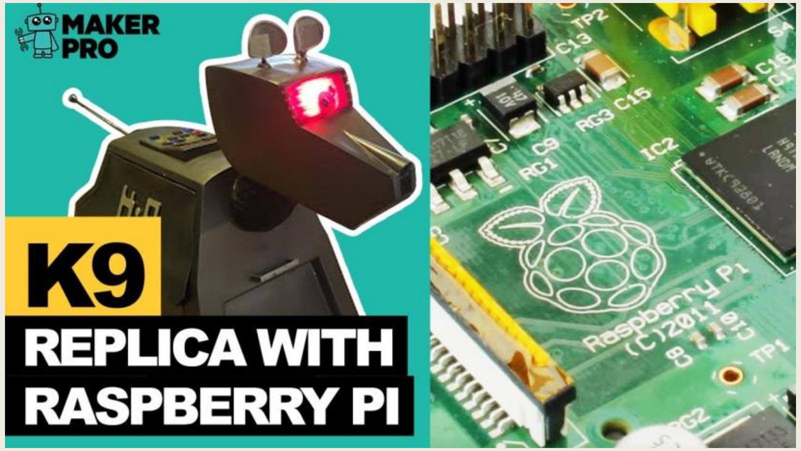


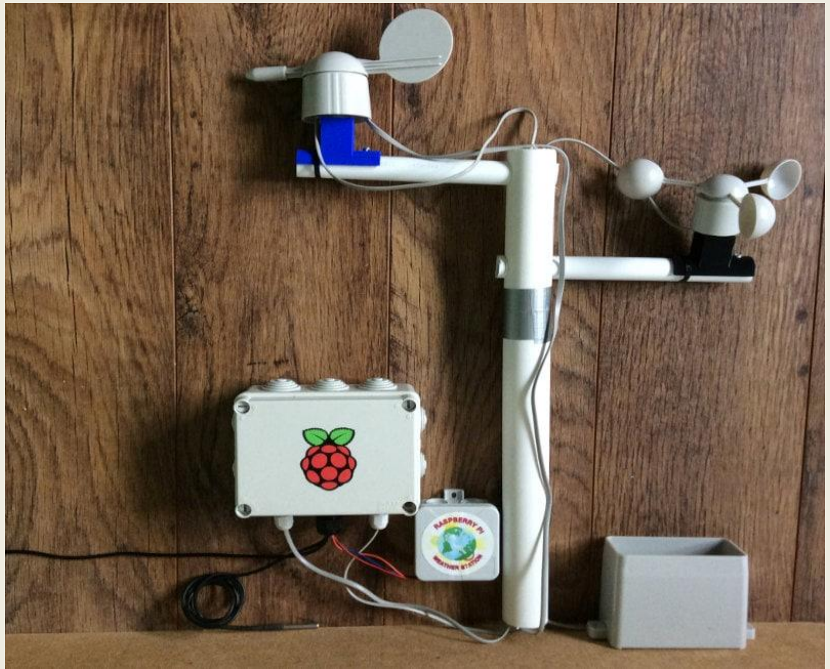
Install Raspberry Pi OS using Raspberry Pi Imager

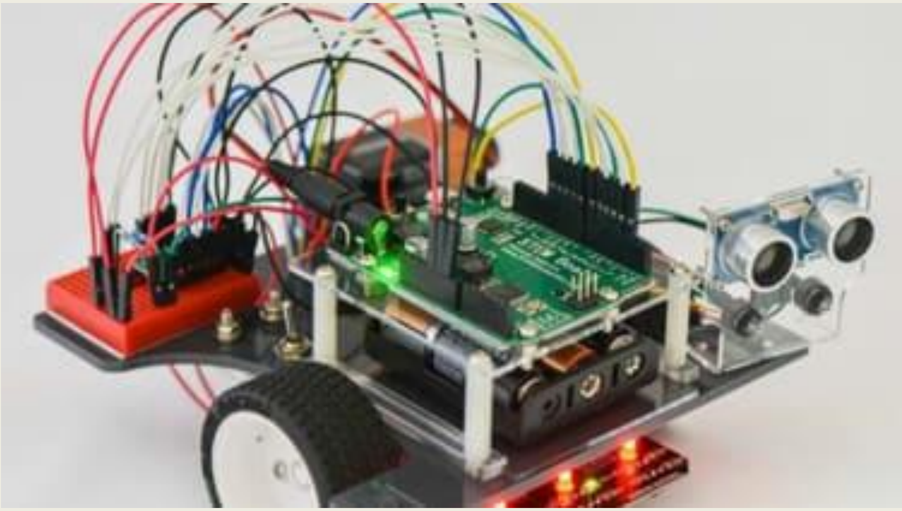
Raspberry Pi Imager is the quick and easy way to install Raspberry Pi OS and other operating systems to a microSD card, ready to use with your Raspberry Pi. [Watch our 45-second video]() to learn how to install an operating system using Raspberry Pi Imager.

Download and install Raspberry Pi Imager to a computer with an SD card reader. Put the SD card you'll use with your Raspberry Pi into the reader and run Raspberry Pi Imager.

WHAT CAN YOU DO WITH RASPBERRY PI?







WHAT WE NEED TODAY

- Our raspberry pi kit:
  - Inside should have a raspberry pi, a car kit, a camera and a camera holder, screws
- Computer
- Tools (screwdrivers)
- SD card (best with 32 GB, more than 8 is fine)
- USB charger that is able to send data
- Portable charger
- Batteries (4AA)

NO MONITOR, ONLY A MOTHERBOARD?

- If we don't have a monitor, how do we connect to the internet??
- There are many ways to connect to our raspberry pi!
  - Use HDMI and connect to telelvision
  - Use a laptop and usb cord (what we will be doing)
- What about internet?
  - Mobile hotspot
  - Ethernet cable
  - Wifi

INTERNET

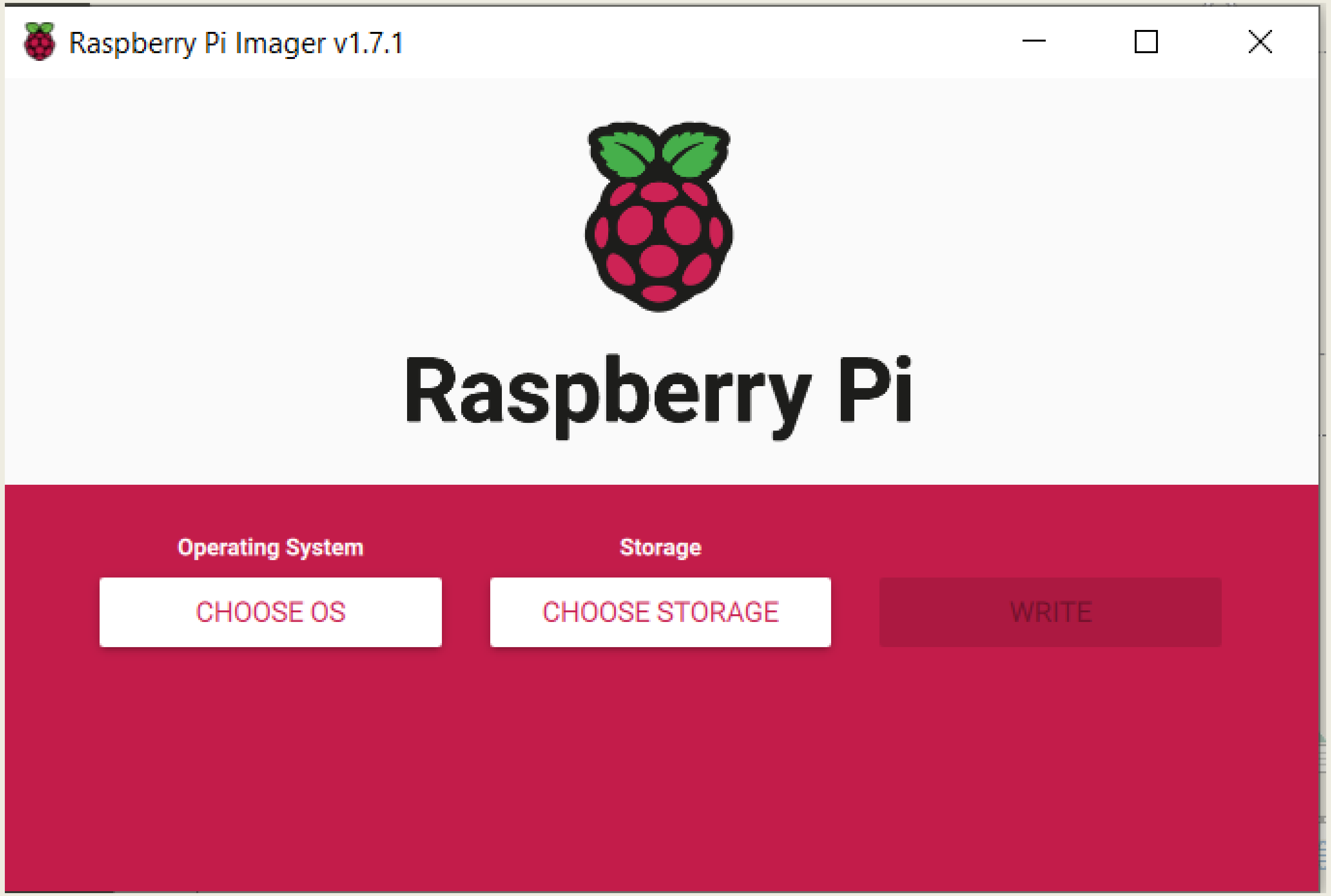
- We'll be connecting our raspberry pi to the wifi
- So we need the ssid and password of our wifi
- What is ssid?
  - It stands for Service Set Identifier
  - Basically it is our wifi's name
- So let's get started on the setting up part!

CONNECTING TO WIFI

- First get your sd card ready and insert it into your computer
- Go to this website and download raspberry pi imager:
  - https://www.raspberrypi.com/software/
- After downloading load the program!
- Install the program and save it somewhere you will remember

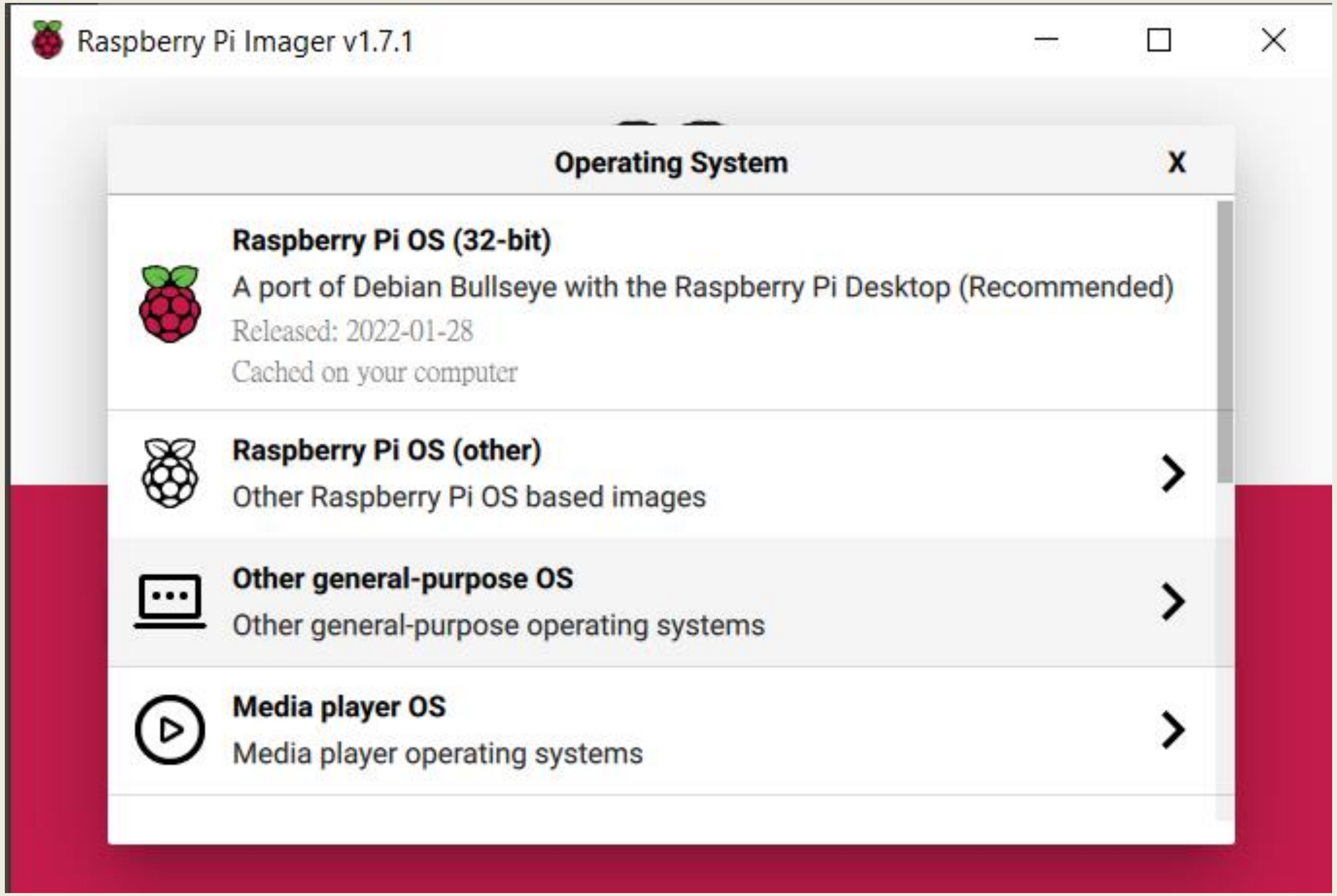
IMAGER

- Should look something like this



STEPS

- Step 1: Choose OS
- Step 2: Pick the first one


Raspberry Pi Imager v1.7.1

Operating System

Raspberry Pi OS (32-bit)
A port of Debian Bullseye with the Raspberry Pi Desktop (Recommended)
Released: 2022-01-28
Cached on your computer

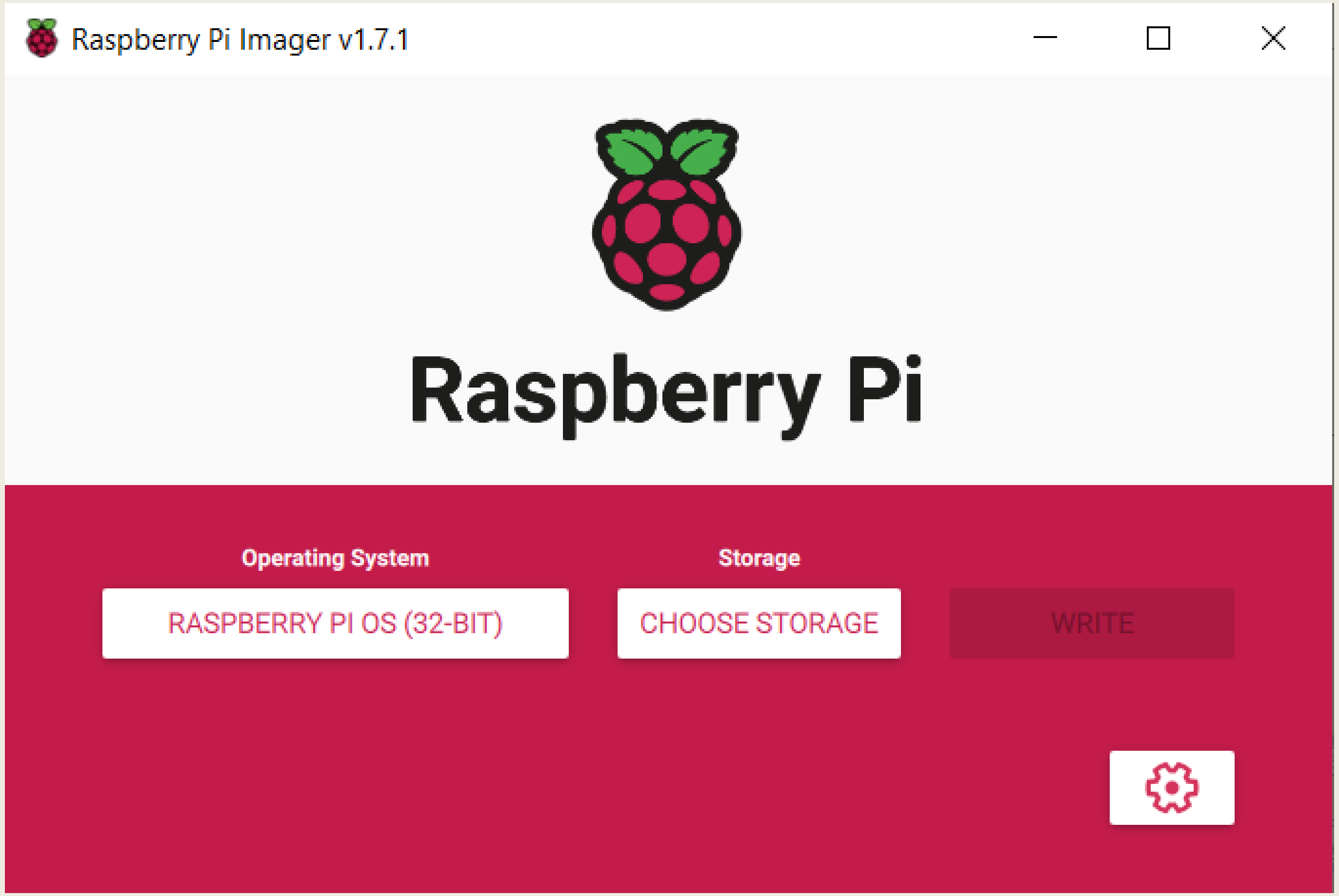
Raspberry Pi OS (other)
Other Raspberry Pi OS based images

Other general-purpose OS
Other general-purpose operating systems

Media player OS
Media player operating systems

STEPS

- Step 3: You should see that we have Raspberry Pi OS in the operating system part
- Step 4: Choose storage
  - If you sd card was inserted properly, there should be a hard drive listed



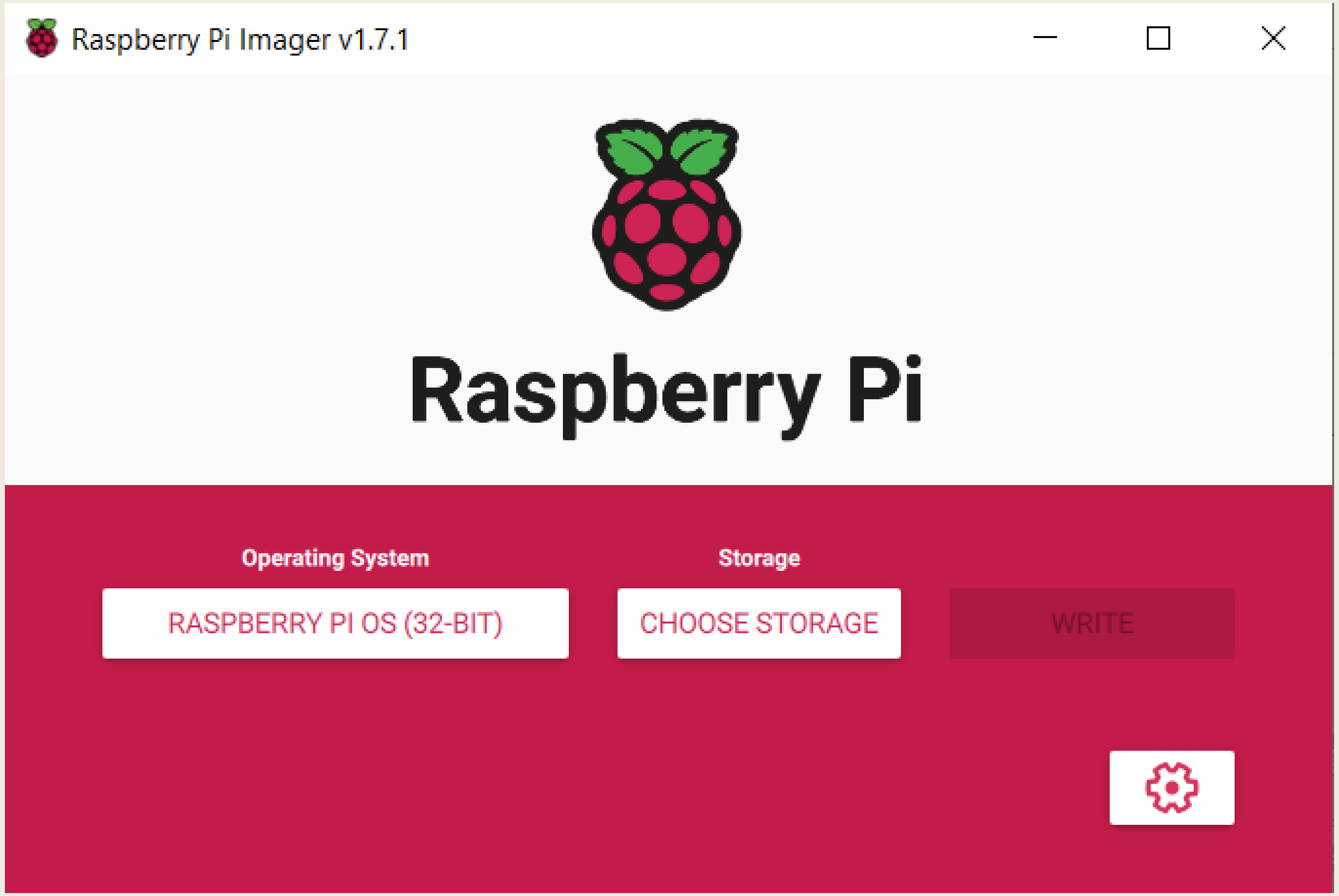
Raspberry Pi Imager v1.7.1

Operating System: RASPBERRY PI OS (32-BIT)

Storage: CHOOSE STORAGE

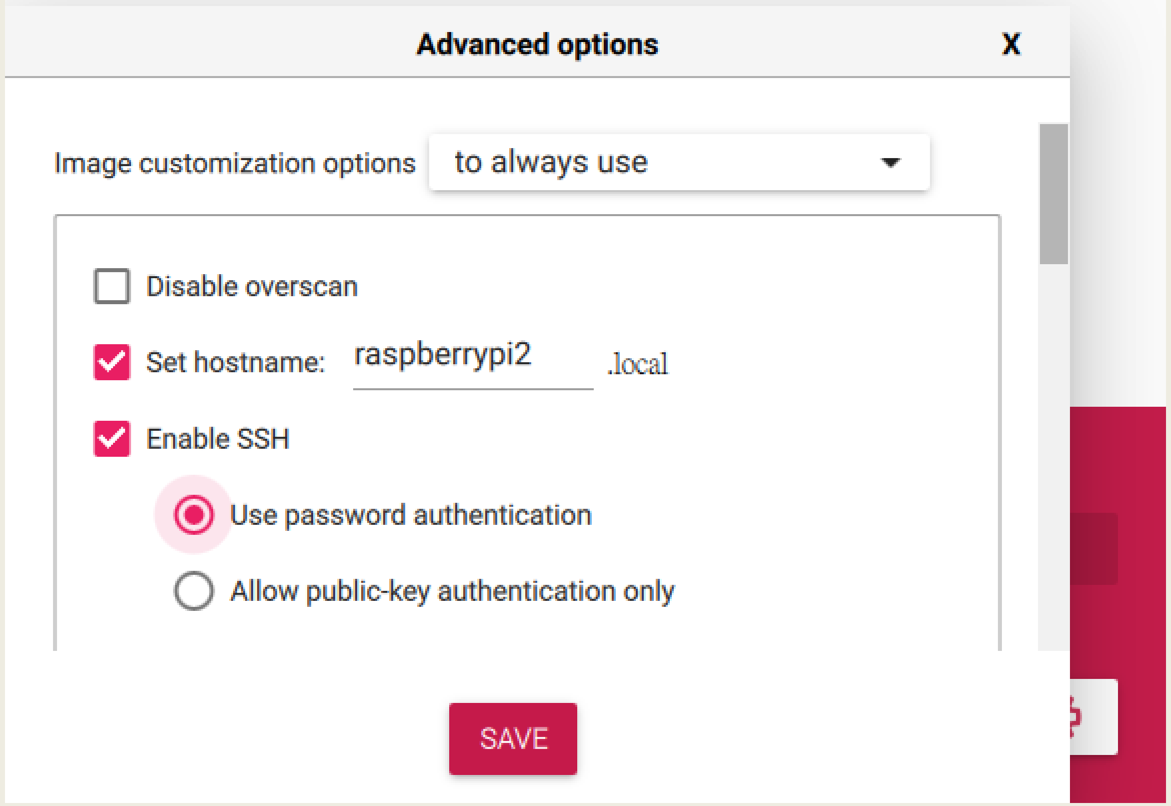
STEPS

- Step 5: IMPORTANT! Settings
- There is a setting icon on the bottom right, click on it



STEPS

- Step 6: Advanced options
- This step is important when we first step up
- Change customization options to
  - Always use
- Step 7: Set hostname is raspberrypi (leave it as it is)
- Step 8: Check Enable SSH



Advanced options

Image customization options: to always use

☐ Disable overscan

☑ Set hostname: raspberrypi2 .local

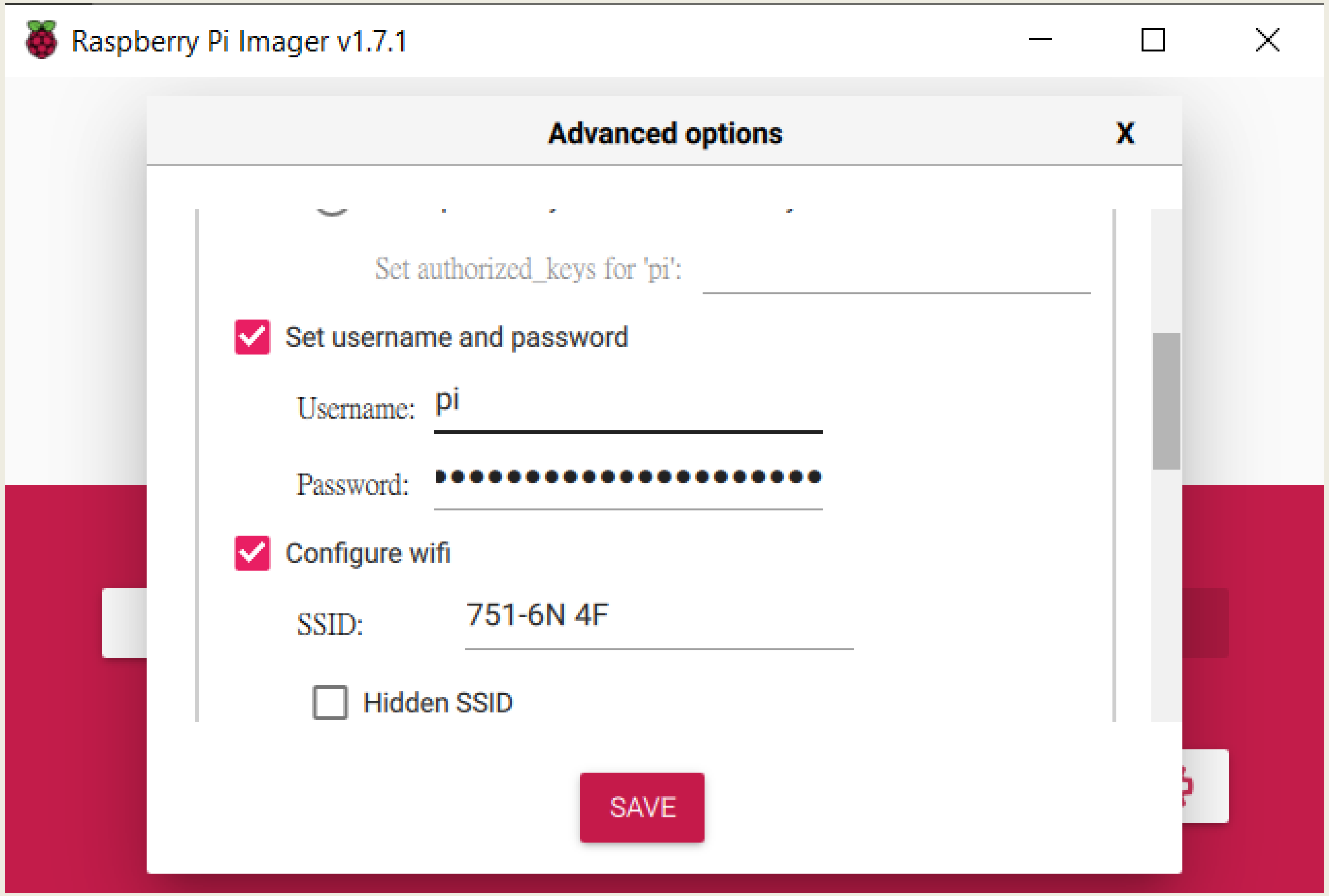
☑ Enable SSH

● Use password authentication

○ Allow public-key authentication only

STEPS

- Step 9: Check and set username and password
  - To make things easier just set user as pi and password as 123
- Step 10: Check Configure wifi
- Step 11: Find ssid of your wifi and password and type it in



Raspberry Pi Imager v1.7.1

Advanced options

Set authorized_keys for 'pi':

☑ Set username and password

Username: pi

Password: ••••••••••••••••••••••

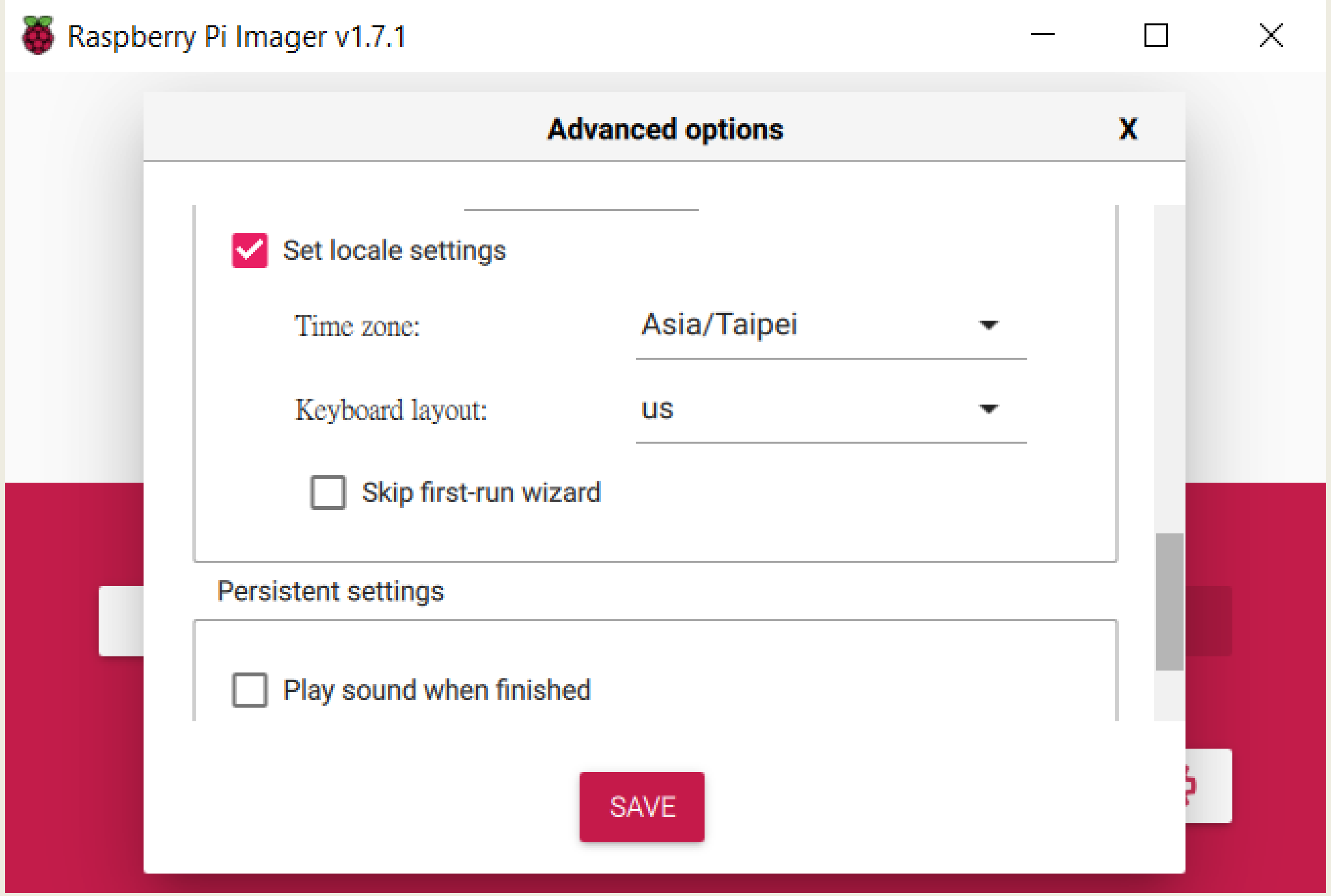
☑ Configure wifi

SSID: 751-6N 4F

□ Hidden SSID

STEPS

- Step 12: Check set locale settings and change time zone to asia/Taipei
- Step 13: Save when you're finished



Raspberry Pi Imager v1.7.1

Advanced options

☑ Set locale settings

Time zone: Asia/Taipei

Keyboard layout: us

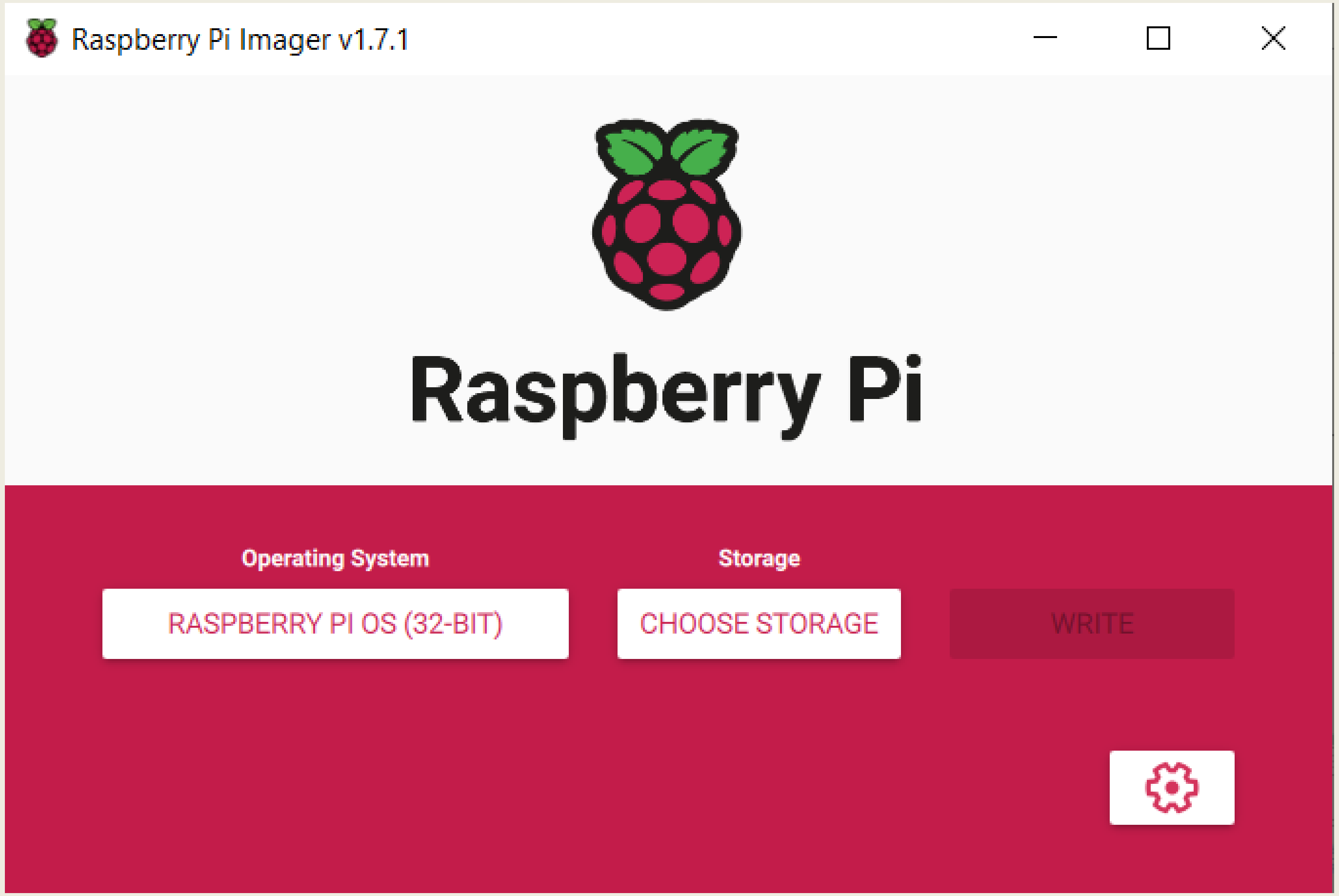
□ Skip first-run wizard

Persistent settings

□ Play sound when finished

STEPS

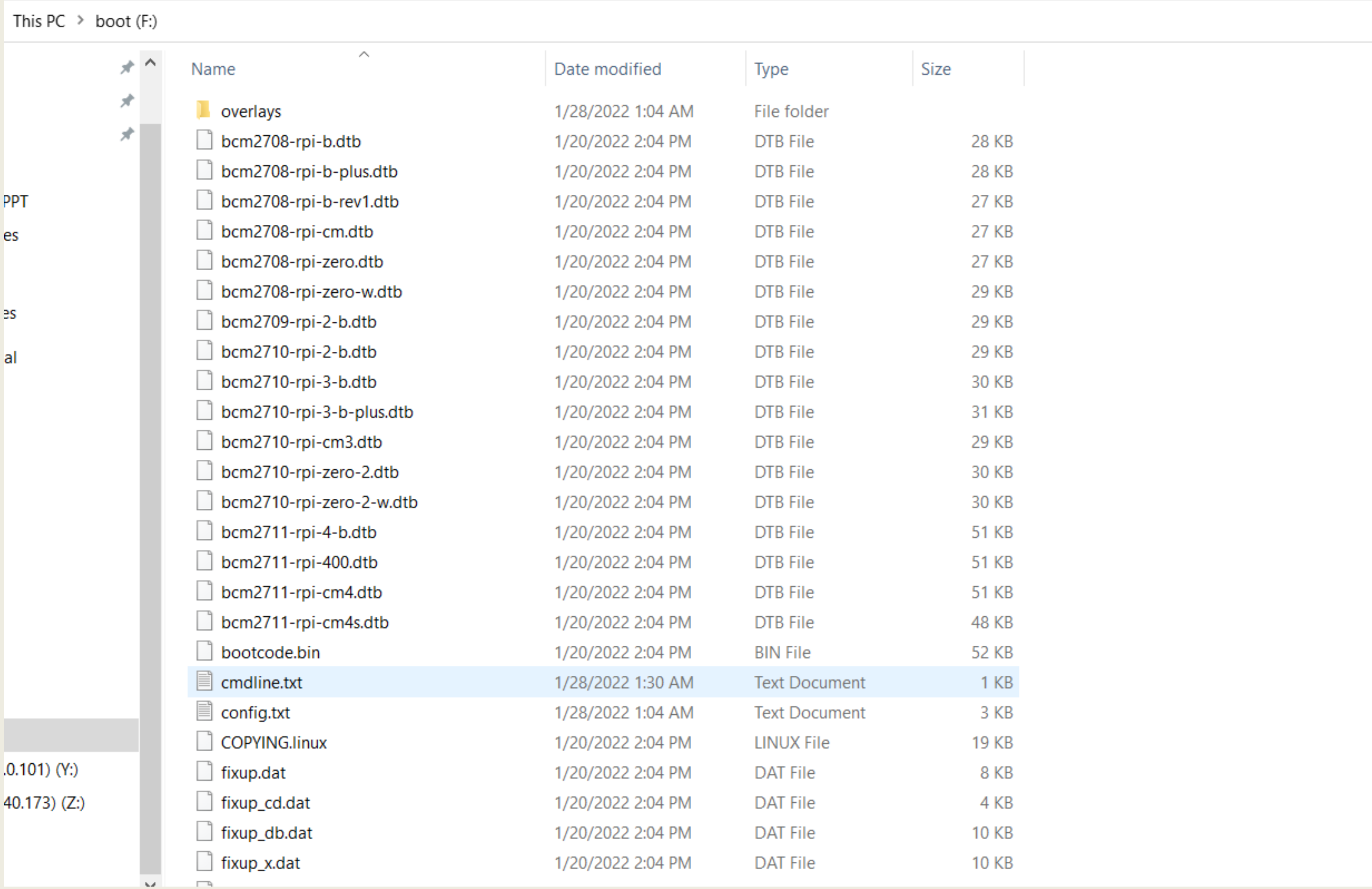
- After everything is set up, click write and the system should be written into your sd card.
- This will take a while so please be patient.



STEPS

- After everything is done, you should get a "You can remove sd card now" message and this means the writing is complete.
- Next, open visual studio code or Notepad++
- For this one, using notepad++ or VSC is fine but we will need Notepad ++ for conversion
- You can always switch programs
- If you don't have Notepad++ or Visual Studio Code please download it!

CURRENT SD CARD



Name	Date modified	Type	Size
overlays	1/28/2022 1:04 AM	File folder	
bcm2708-rpi-b.dtb	1/20/2022 2:04 PM	DTB File	28 KB
bcm2708-rpi-b-plus.dtb	1/20/2022 2:04 PM	DTB File	28 KB
bcm2708-rpi-b-rev1.dtb	1/20/2022 2:04 PM	DTB File	27 KB
bcm2708-rpi-cm.dtb	1/20/2022 2:04 PM	DTB File	27 KB
bcm2708-rpi-zero.dtb	1/20/2022 2:04 PM	DTB File	27 KB
bcm2708-rpi-zero-w.dtb	1/20/2022 2:04 PM	DTB File	29 KB
bcm2709-rpi-2-b.dtb	1/20/2022 2:04 PM	DTB File	29 KB
bcm2710-rpi-2-b.dtb	1/20/2022 2:04 PM	DTB File	29 KB
bcm2710-rpi-3-b.dtb	1/20/2022 2:04 PM	DTB File	30 KB
bcm2710-rpi-3-b-plus.dtb	1/20/2022 2:04 PM	DTB File	31 KB
bcm2710-rpi-cm3.dtb	1/20/2022 2:04 PM	DTB File	29 KB
bcm2710-rpi-zero-2.dtb	1/20/2022 2:04 PM	DTB File	30 KB
bcm2710-rpi-zero-2-w.dtb	1/20/2022 2:04 PM	DTB File	30 KB
bcm2711-rpi-4-b.dtb	1/20/2022 2:04 PM	DTB File	51 KB
bcm2711-rpi-400.dtb	1/20/2022 2:04 PM	DTB File	51 KB
bcm2711-rpi-cm4.dtb	1/20/2022 2:04 PM	DTB File	51 KB
bcm2711-rpi-cm4s.dtb	1/20/2022 2:04 PM	DTB File	48 KB
bootcode.bin	1/20/2022 2:04 PM	BIN File	52 KB
cmdline.txt	1/28/2022 1:30 AM	Text Document	1 KB
config.txt	1/28/2022 1:04 AM	Text Document	3 KB
COPYING.linux	1/20/2022 2:04 PM	LINUX File	19 KB
fixup.dat	1/20/2022 2:04 PM	DAT File	8 KB
fixup_cd.dat	1/20/2022 2:04 PM	DAT File	4 KB
fixup_db.dat	1/20/2022 2:04 PM	DAT File	10 KB
fixup_x.dat	1/20/2022 2:04 PM	DAT File	10 KB

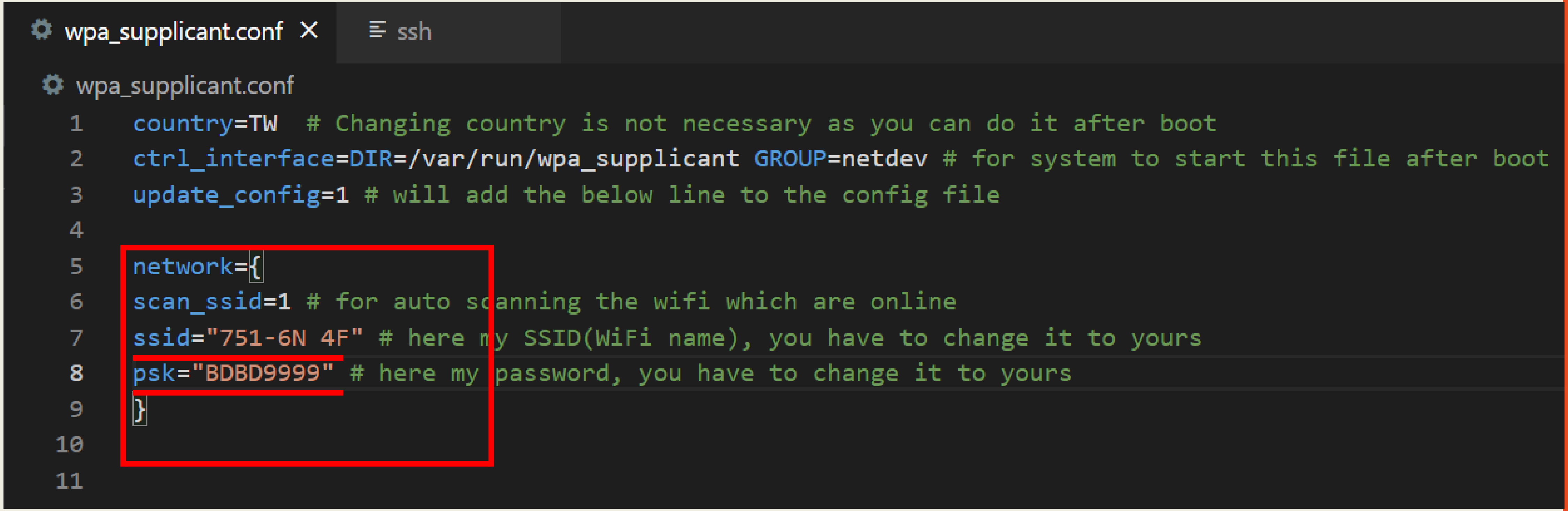
FILES

- We’re going to add two files into our sd card
  - ssh
  - wpa_supplicant.conf
- SSH File: Secure Shell
  - It is used to let your raspberry pi connect to ssh
  - Secure Shell is a network communication protocol that enables two computers to communicate
  - For the ssh file, create it without any extensions

WPA_SUPPLICANT.CONF

- Please type the name of the file as shown
- For this file, we’ll be coding on the network we want our raspberry pi to connect to
- This file is for auto-connecting to your android phone's hotspot or Wi-Fi.
- We’ll be changing to our wifi’s name and password

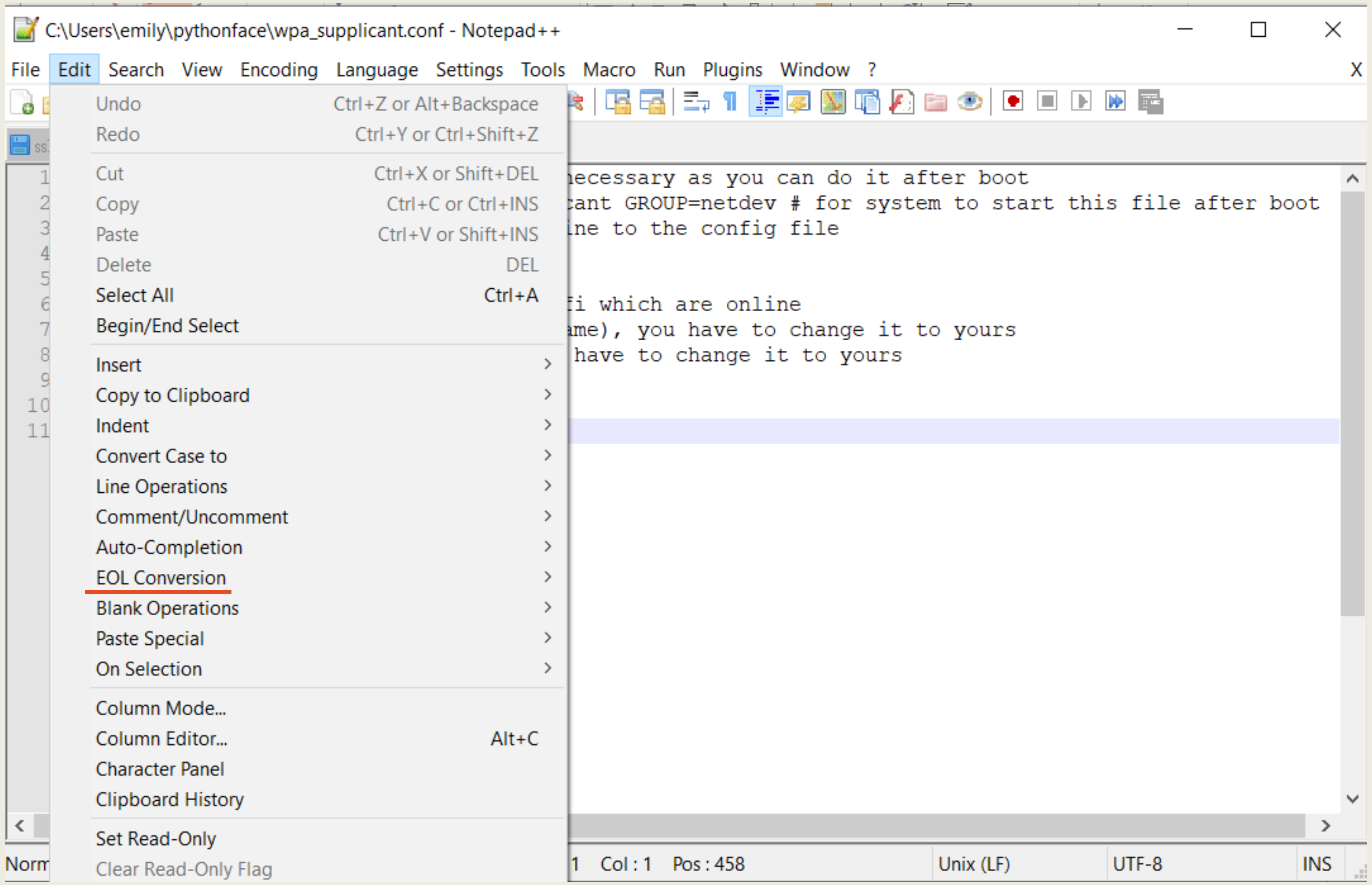
WPA_SUPPLICANT.CONF



```
country=TW  # Changing country is not necessary as you can do it after boot
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev # for system to start this file after boot
update_config=1 # will add the below line to the config file

network={
scan_ssid=1 # for auto scanning the wifi which are online
ssid="751-6N 4F" # here my SSID(WiFi name), you have to change it to yours
psk="BDBD9999" # here my password, you have to change it to yours
}


```

STEPS

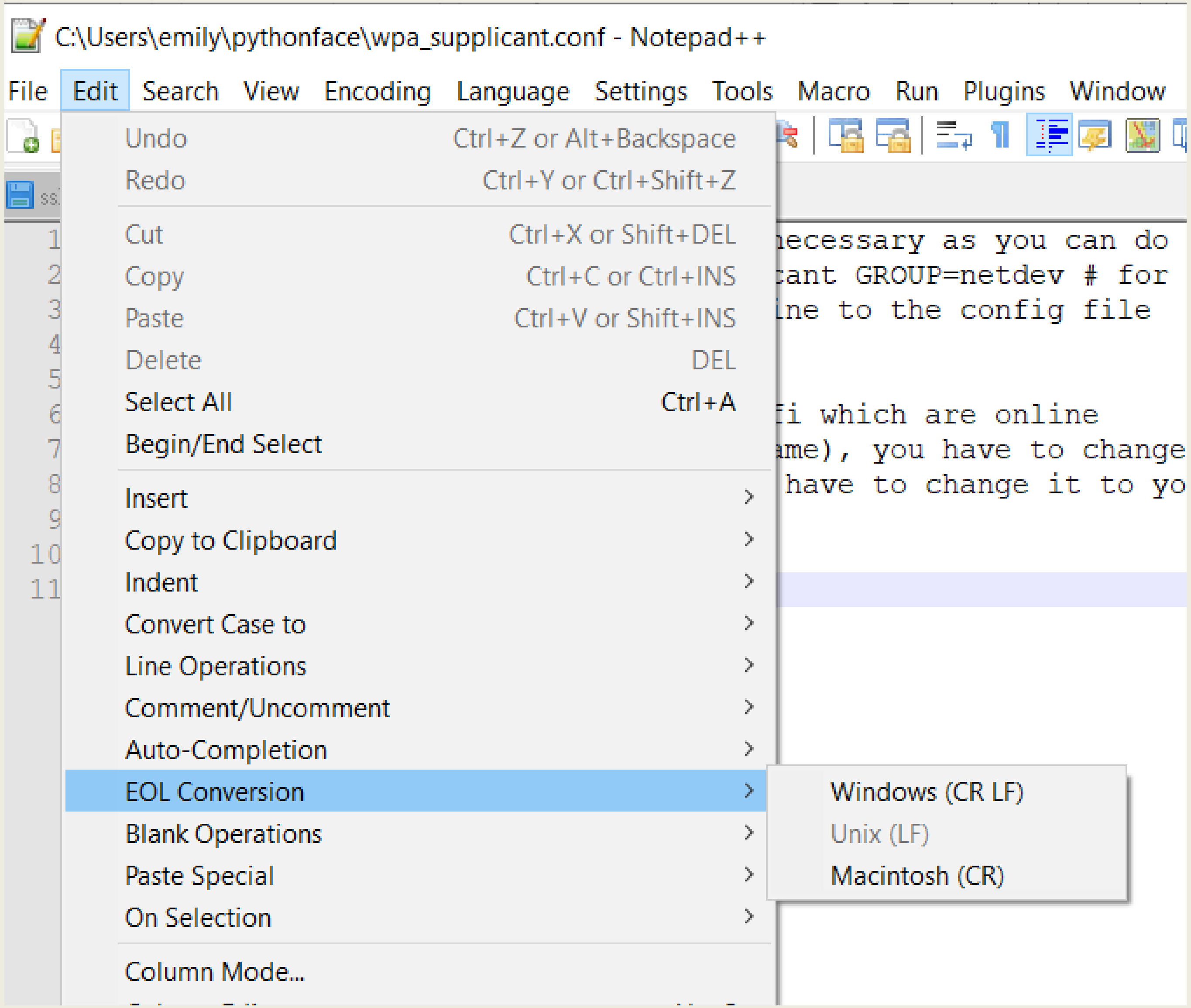
- After you've created the wpa_supplicant.conf file, open it with Notepad++
- Open Edit and look for EOL Conversion



```
ecessary as you can do it after boot
ant GROUP=netdev # for system to start this file after boot
ine to the config file


i which are online
me), you have to change it to yours
have to change it to yours
```

STEPS

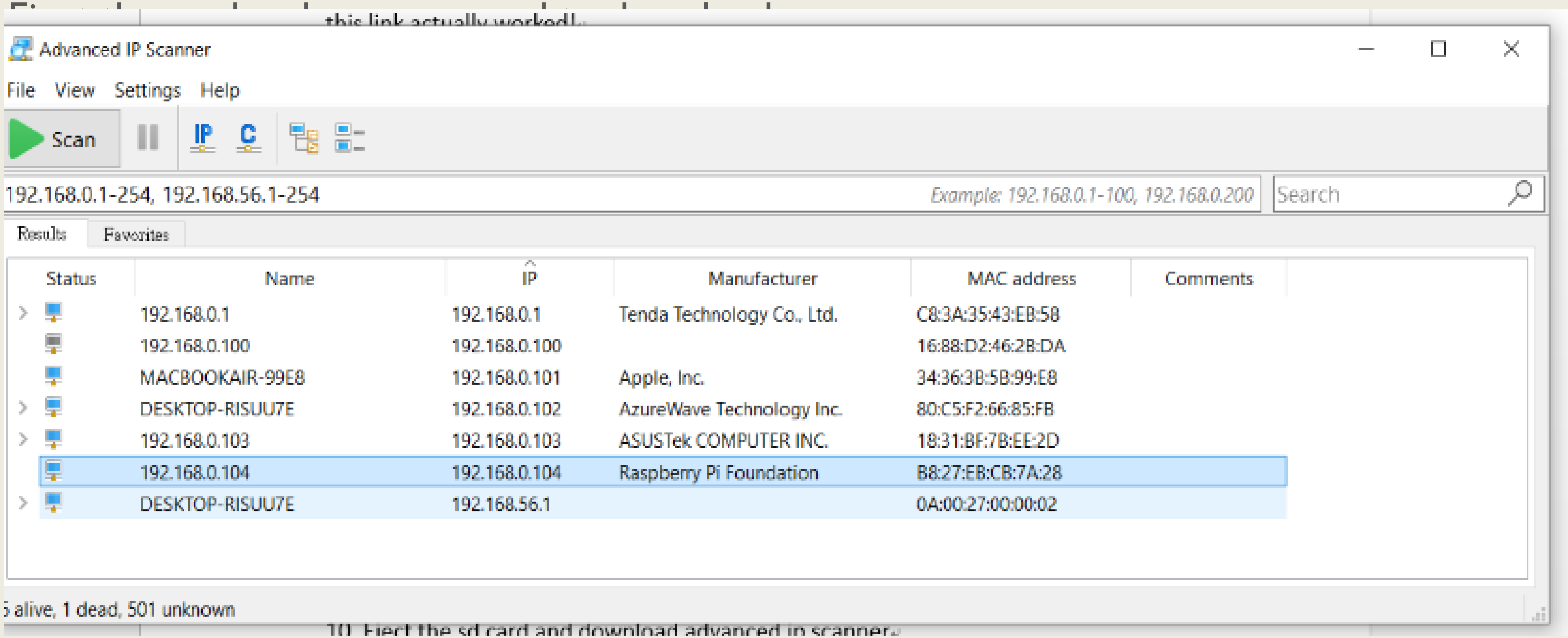
- Hover over EOL Conversion and click on Unix
- Save your file and we should be good to go
- Exit Notepad++ and put these two files into your sd card



C:\Users\emily\pythonface\wpa_supplicant.conf - Notepad++

```
necessary as you can do
cant GROUP=netdev # for
ine to the config file


fi which are online
ame), you have to change
 have to change it to yo
```

STEPS



this link actually worked!

Advanced IP Scanner

192.168.0.1-254, 192.168.56.1-254

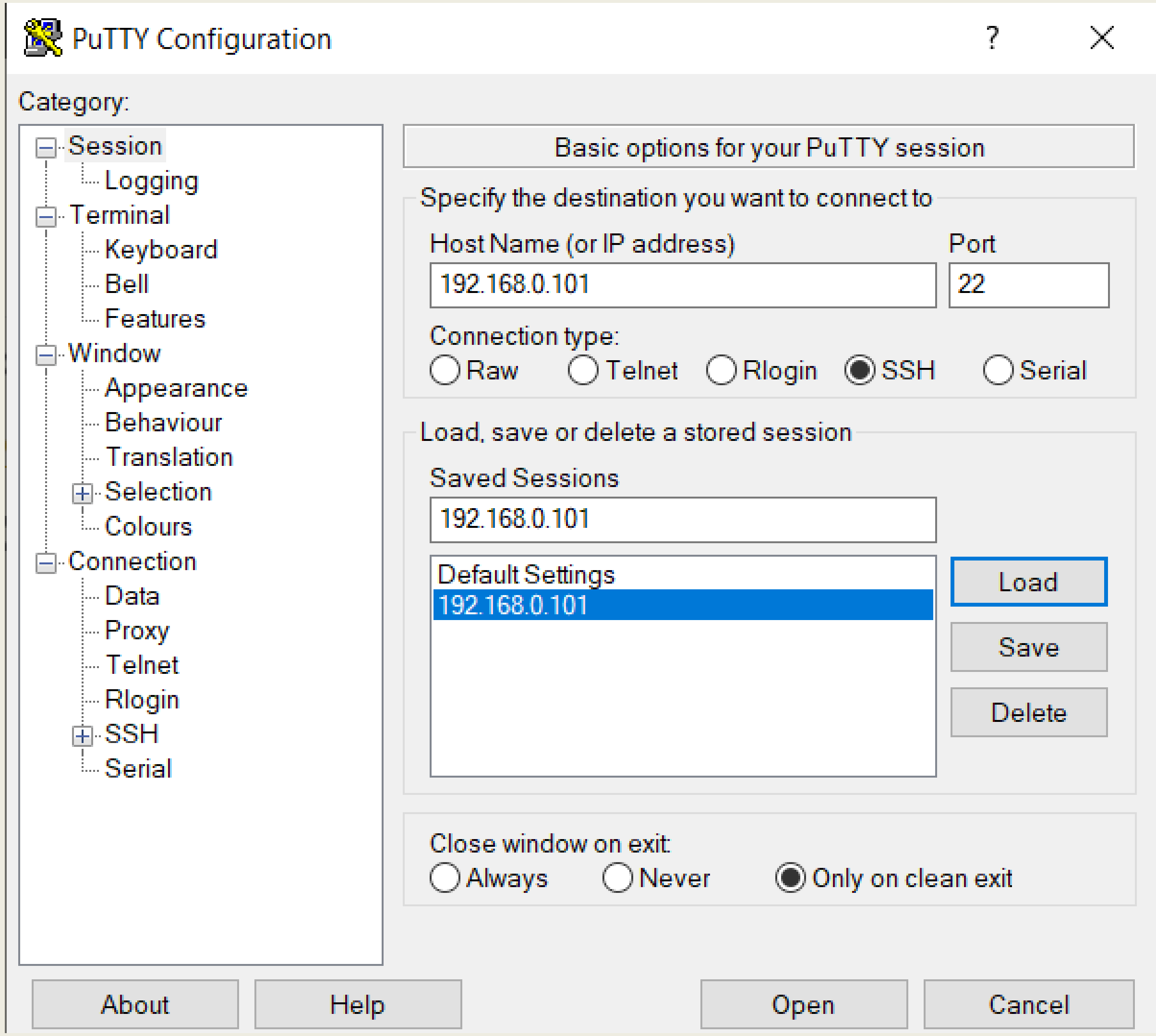
Status	Name	IP	Manufacturer	MAC address	Comments
	192.168.0.1	192.168.0.1	Tenda Technology Co., Ltd.	C8:3A:35:43:EB:58	
	192.168.0.100	192.168.0.100		16:88:D2:46:2B:DA	
	MACBOOKAIR-99E8	192.168.0.101	Apple, Inc.	34:36:3B:5B:99:E8	
	DESKTOP-RISUU7E	192.168.0.102	AzureWave Technology Inc.	80:C5:F2:66:85:FB	
	192.168.0.103	192.168.0.103	ASUSTek COMPUTER INC.	18:31:BF:7B:EE:2D	
	192.168.0.104	192.168.0.104	Raspberry Pi Foundation	B8:27:EB:CB:7A:28	
	DESKTOP-RISUU7E	192.168.56.1		0A:00:27:00:00:02	

5 alive, 1 dead, 501 unknown

10. Eject the sd card and download advanced ip scanner.

STEPS

- Next download Putty
  - https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html
- Type in your ip address for your raspberry pi and open



PuTTY Configuration

Category:

Basic options for your PuTTY session

Specify the destination you want to connect to

Host Name (or IP address): 192.168.0.101

Port: 22

Connection type: ○ Raw ○ Telnet ○ Rlogin ● SSH ○ Serial

Load, save or delete a stored session

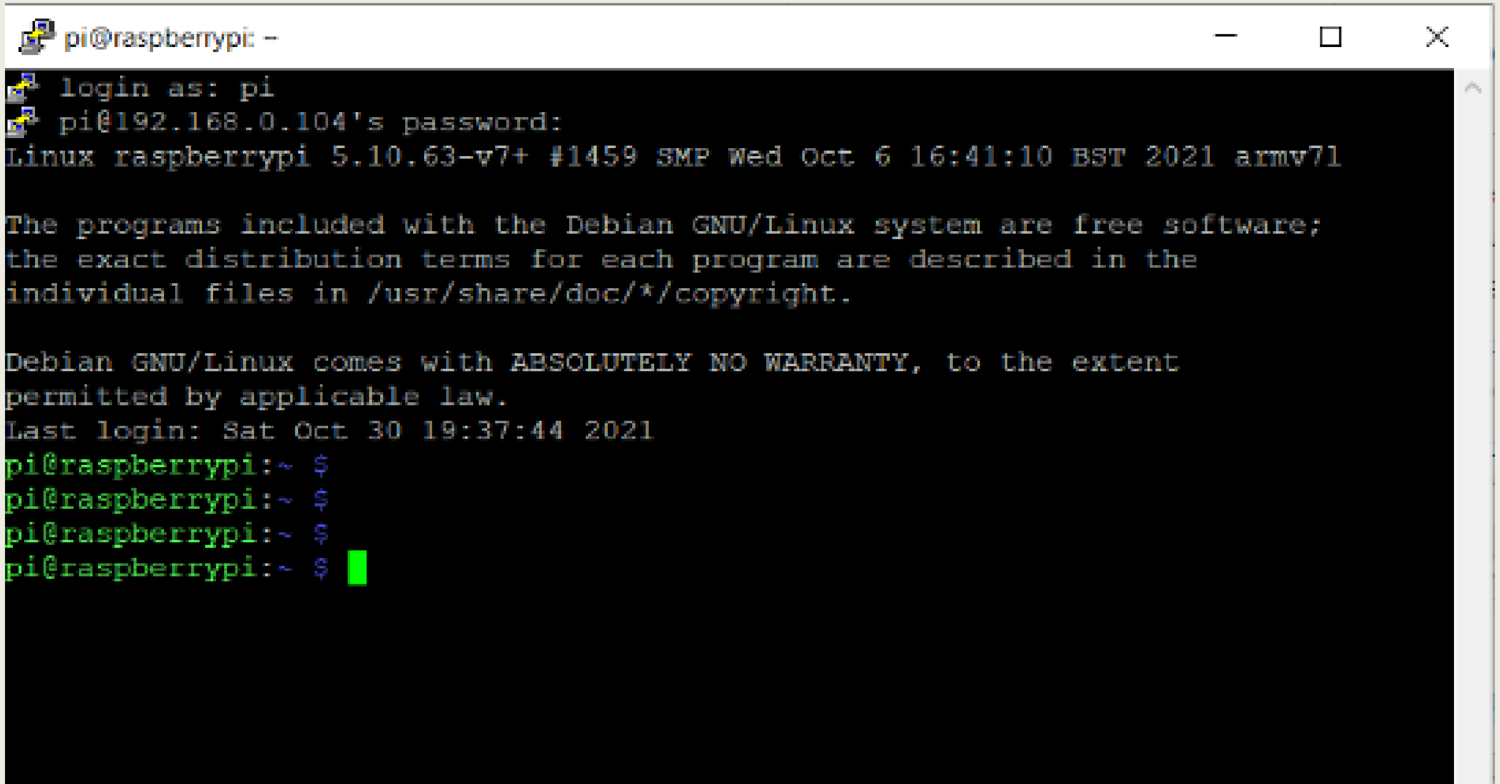
Saved Sessions: 192.168.0.101

Default Settings
192.168.0.101

Close window on exit: ○ Always ○ Never ● Only on clean exit

STEPS

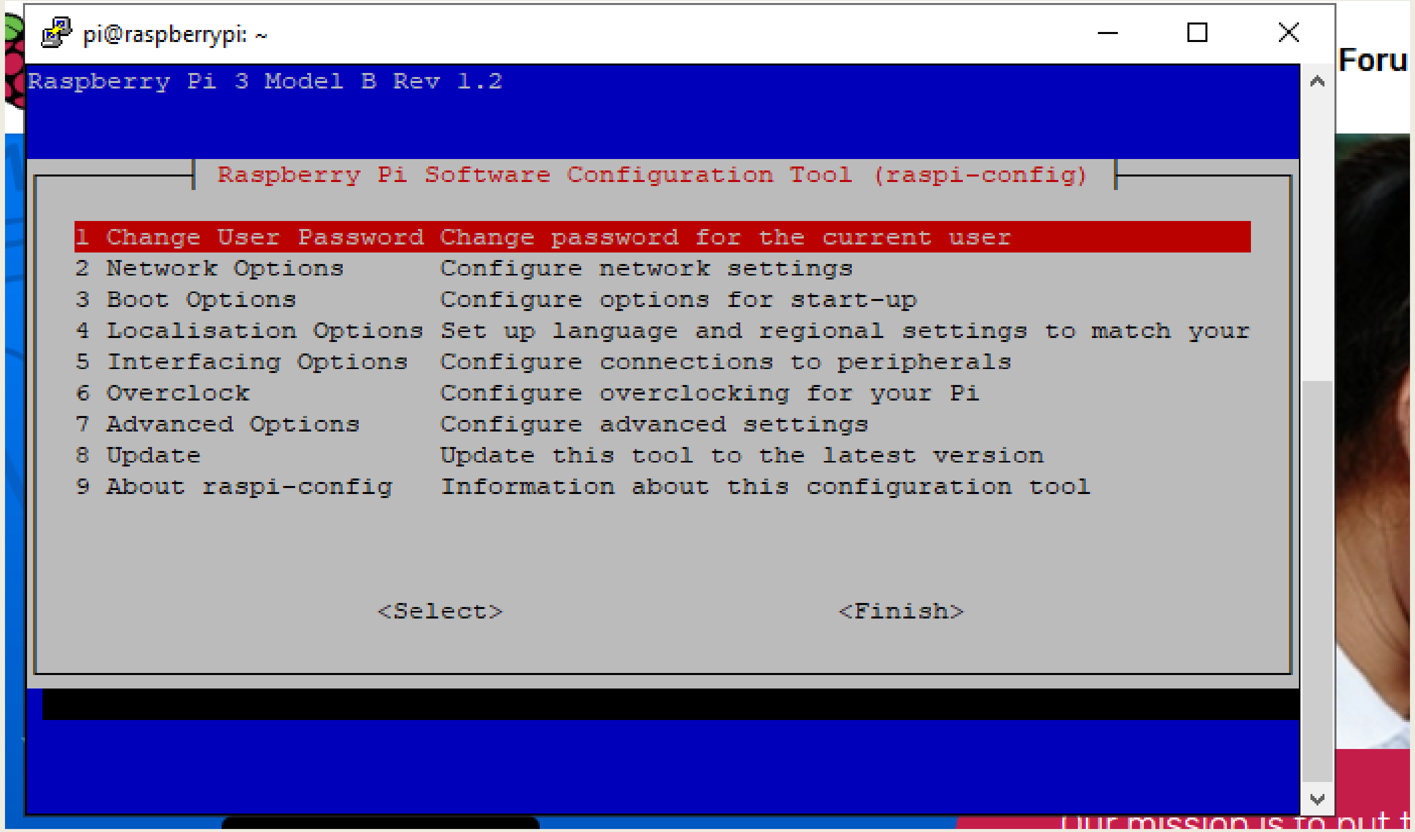
- It will ask you login as: type in the user pi
- It will
- also ask for password, which we have set during the raspberry pi imager
- If all is correct then we should be able to see this



```
login as: pi
pi@192.168.0.104's password:
Linux raspberrypi 5.10.63-v7+ #1459 SMP Wed Oct 6 16:41:10 BST 2021 armv7l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sat Oct 30 19:37:44 2021
pi@raspberrypi:~ $
pi@raspberrypi:~ $
pi@raspberrypi:~ $
pi@raspberrypi:~ $
```

STEPS

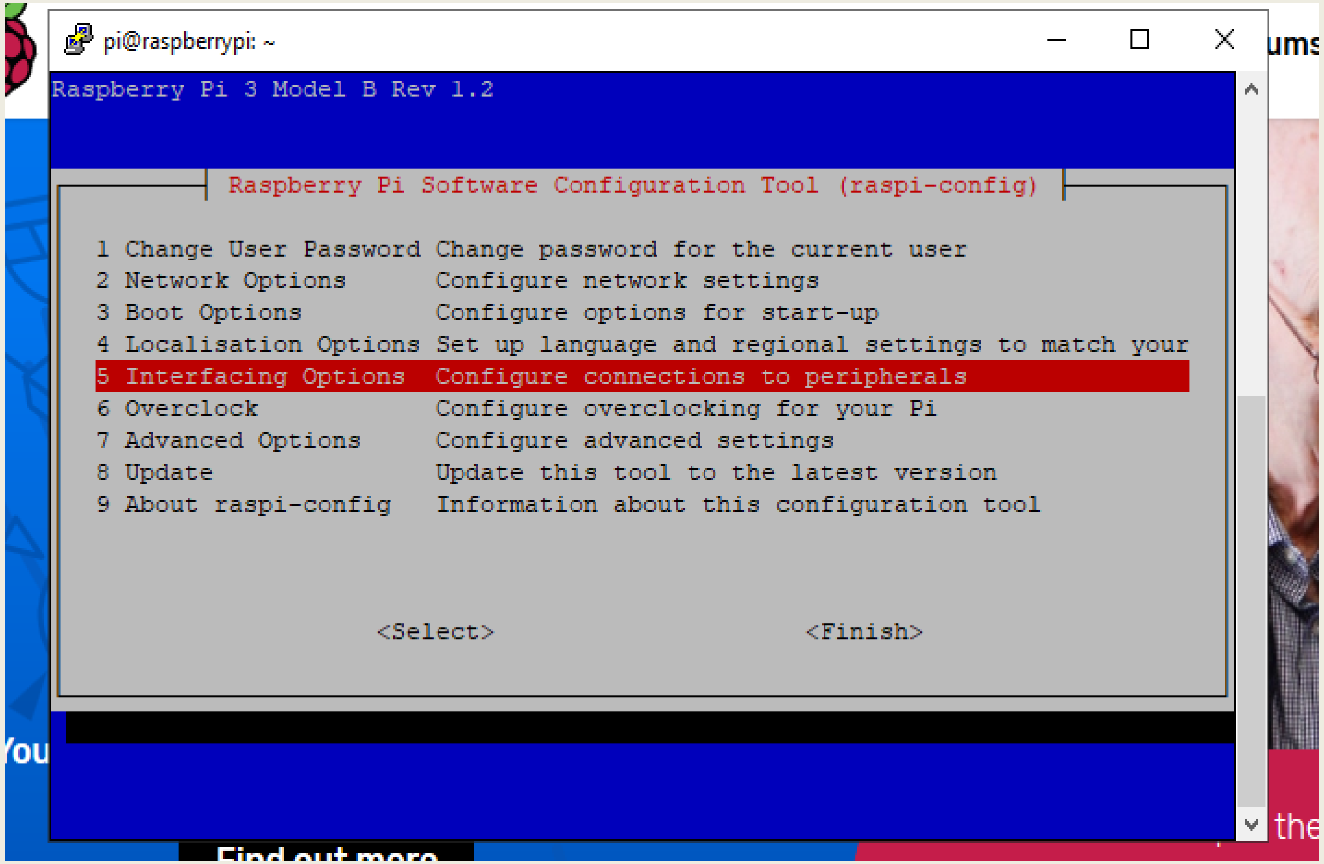
- Next, we want to type in the command:
  - sudo raspi-config
- You should see this interface



```
Raspberry Pi 3 Model B Rev 1.2

         Raspberry Pi Software Configuration Tool (raspi-config)

 1 Change User Password Change password for the current user
 2 Network Options      Configure network settings
 3 Boot Options         Configure options for start-up
 4 Localisation Options Set up language and regional settings to match your
 5 Interfacing Options  Configure connections to peripherals
 6 Overclock            Configure overclocking for your Pi
 7 Advanced Options     Configure advanced settings
 8 Update               Update this tool to the latest version
 9 About raspi-config   Information about this configuration tool

                  <Select>                     <Finish>
```

STEPS

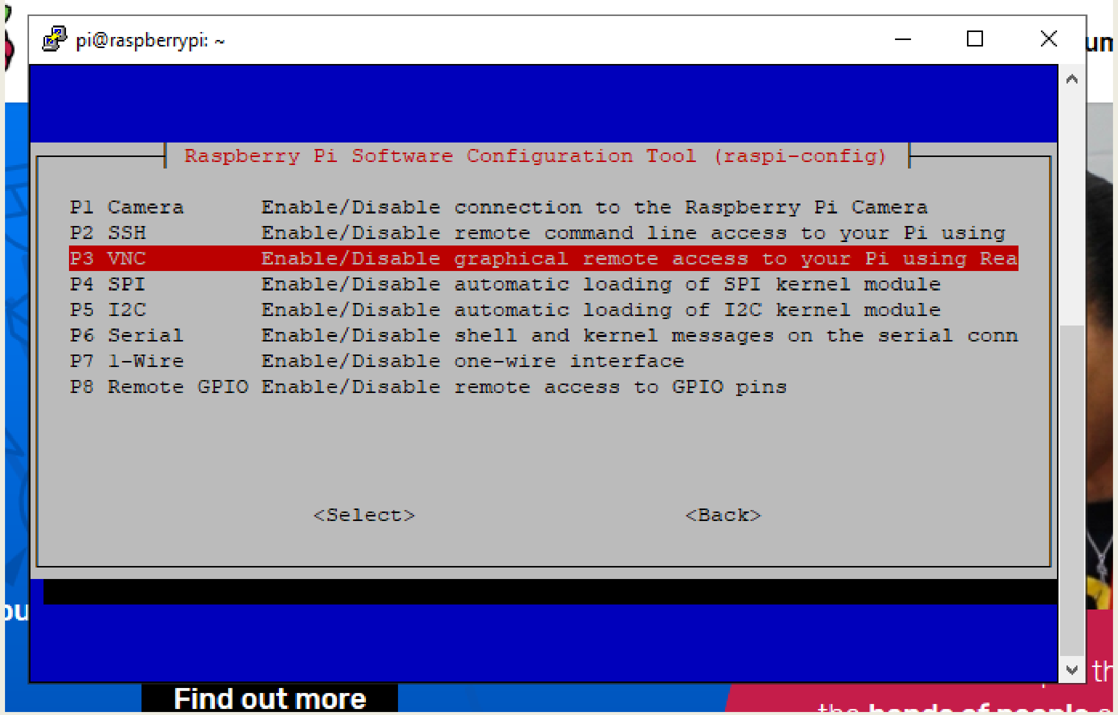
- Go down to interfacing options
- Press enter



```
Raspberry Pi 3 Model B Rev 1.2

        Raspberry Pi Software Configuration Tool (raspi-config)

 1 Change User Password Change password for the current user
 2 Network Options      Configure network settings
 3 Boot Options         Configure options for start-up
 4 Localisation Options Set up language and regional settings to match your
 5 Interfacing Options  Configure connections to peripherals
 6 Overclock            Configure overclocking for your Pi
 7 Advanced Options     Configure advanced settings
 8 Update               Update this tool to the latest version
 9 About raspi-config   Information about this configuration tool

                <Select>                     <Finish>
```

STEPS

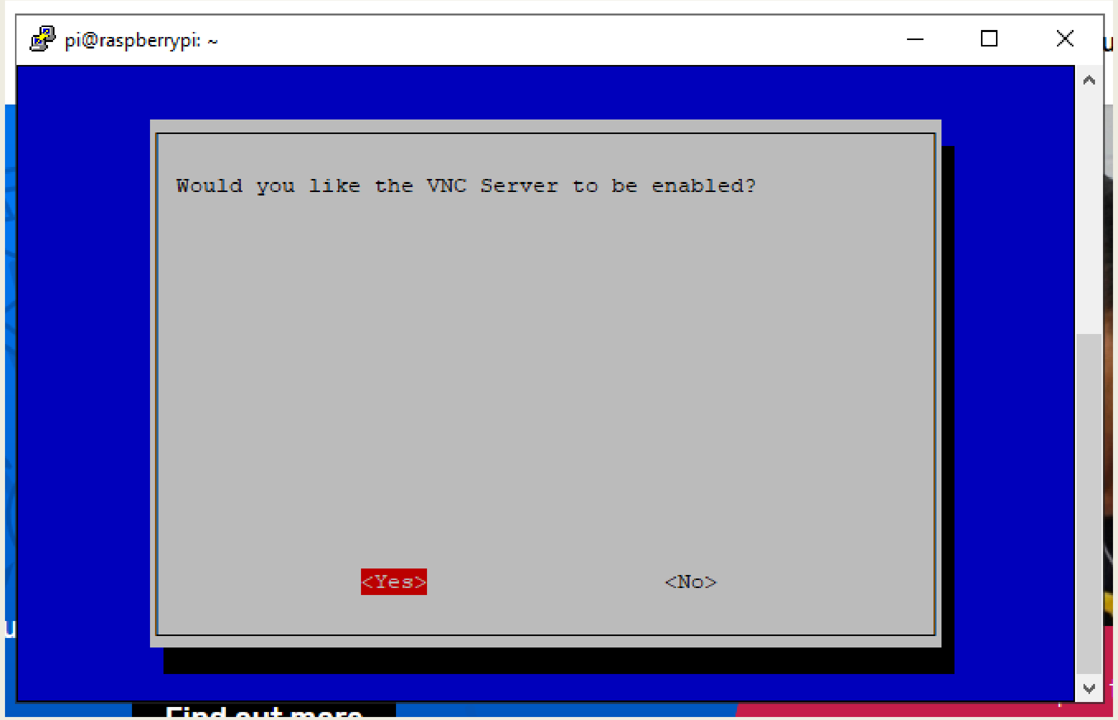
- You should see VNC
- Enter on VNC



```
Raspberry Pi Software Configuration Tool (raspi-config)

P1 Camera      Enable/Disable connection to the Raspberry Pi Camera
P2 SSH         Enable/Disable remote command line access to your Pi using
P3 VNC         Enable/Disable graphical remote access to your Pi using Rea
P4 SPI         Enable/Disable automatic loading of SPI kernel module
P5 I2C         Enable/Disable automatic loading of I2C kernel module
P6 Serial      Enable/Disable shell and kernel messages on the serial conn
P7 1-Wire      Enable/Disable one-wire interface
P8 Remote GPIO Enable/Disable remote access to GPIO pins

                <Select>                     <Back>
```

STEPS

- It would ask if you would like to enable VNC
- Press yes
- After than we should be done and exit



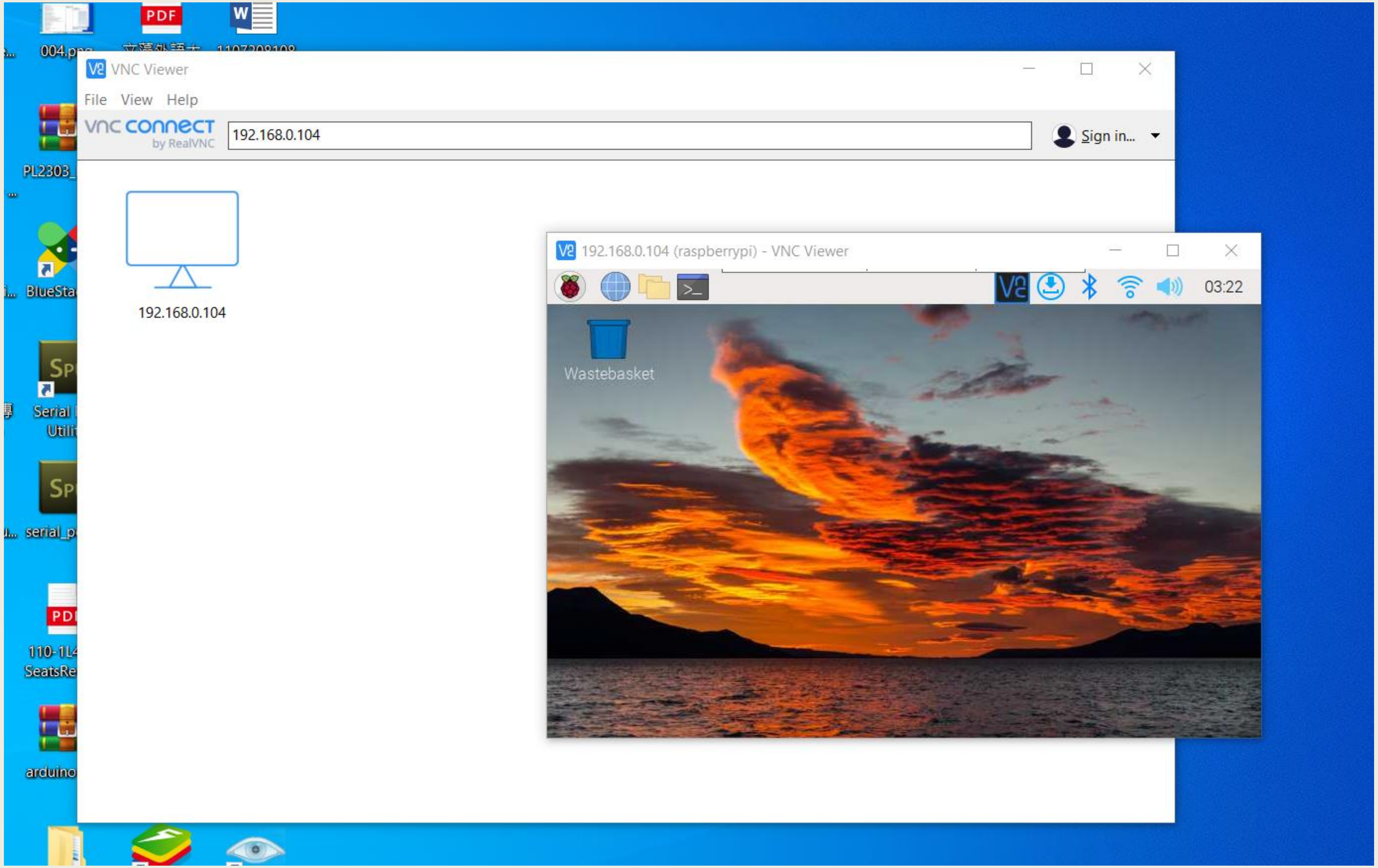
pi@raspberrypi: ~

Would you like the VNC Server to be enabled?

STEPS

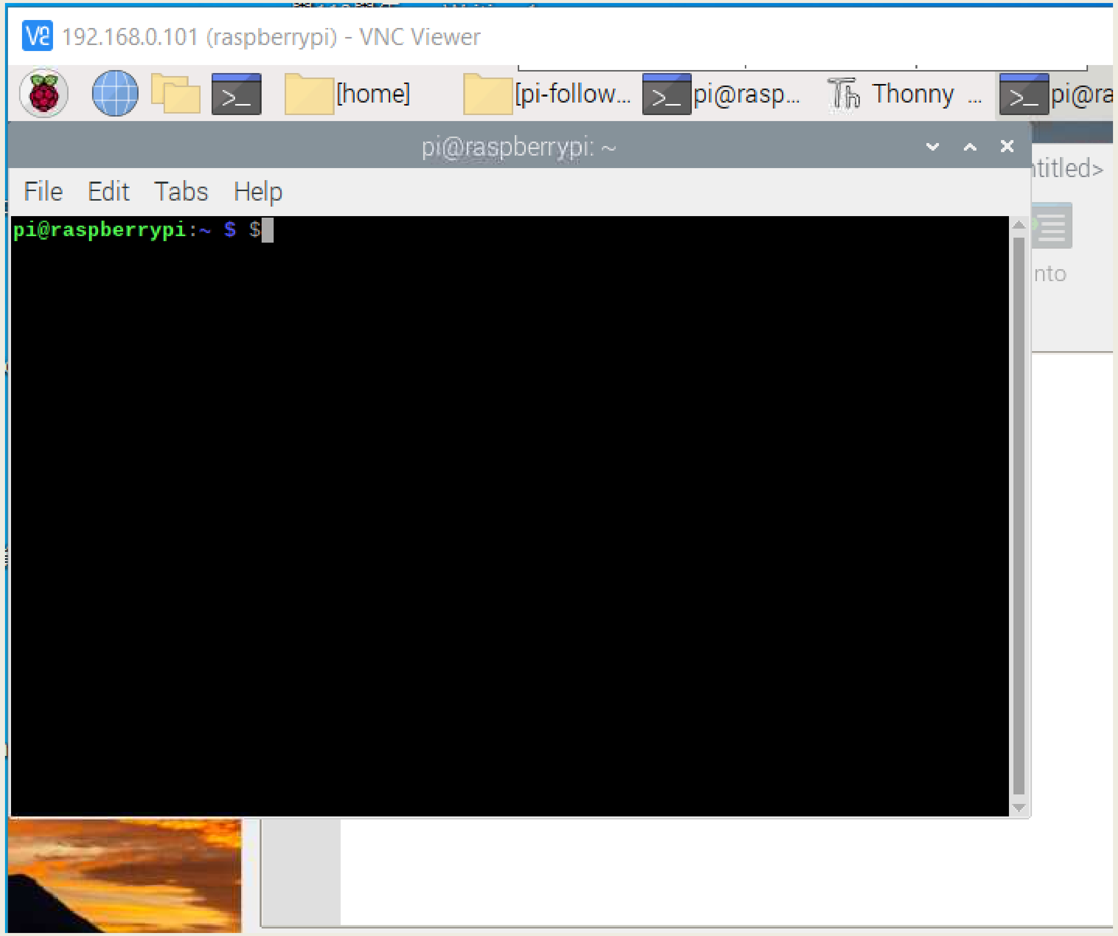
- Next, download VNC viewer
  - – https://www.realvnc.com/en/connect/download/viewer/
- Type in your IP address and it should find it for you
- Type in your user (pi) and password (123)
- And now we should have an interface that looks like a mini computer

GOOD JOB! WE ARE NOW INSIDE



INSTALLATIONS

- Next is installing programs
- Enter your raspberry pi and open terminal



```
pi@raspberrypi:~ $ $
```

INSTALLATIONS

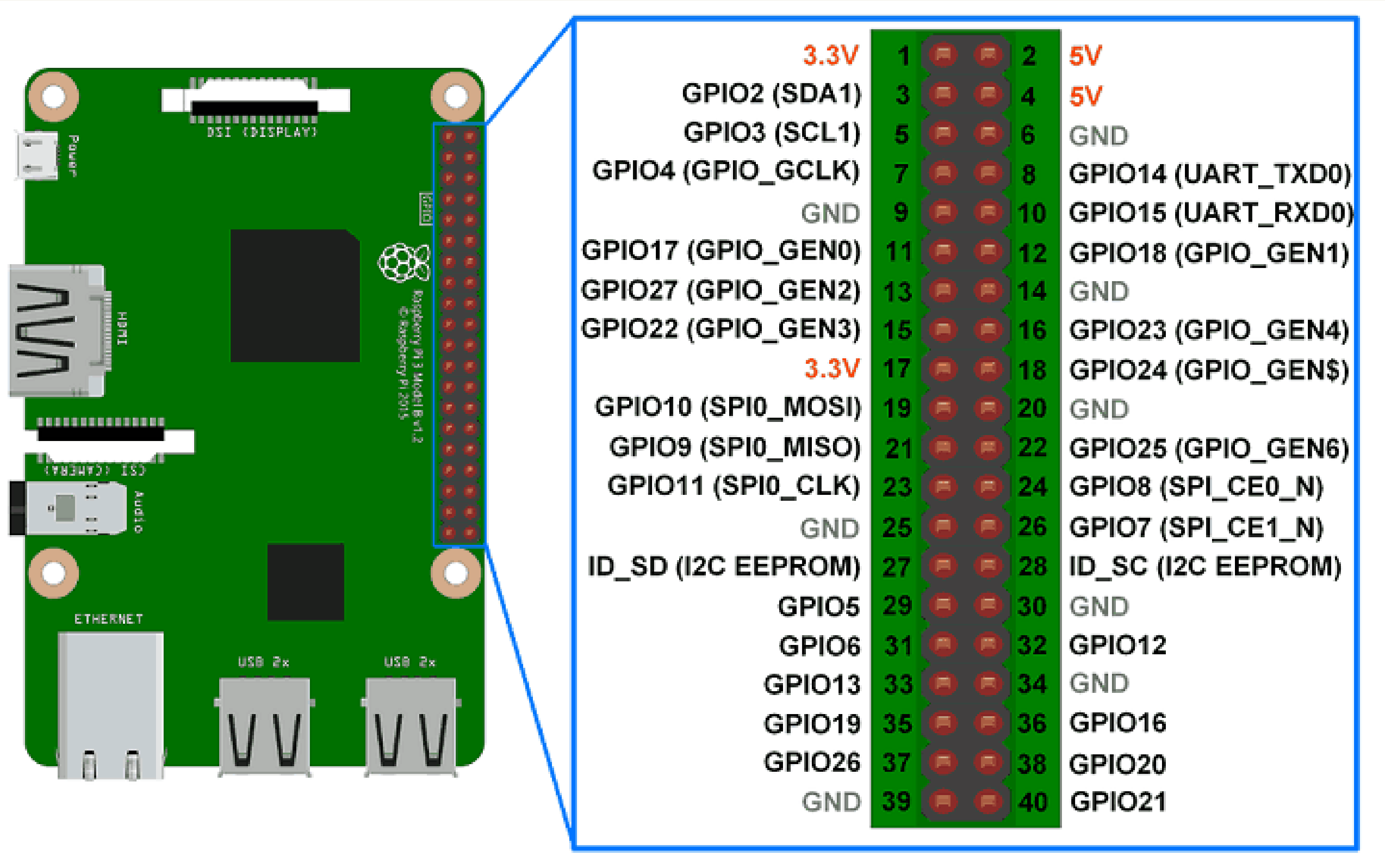
- Using terminal we will start coding
- First off, type in:
  - sudo apt-get update
- Next, we want to upgrade
  - sudo apt full-upgrade
- Then we download opencv
  - sudo apt install -y python3-opencv
- This may take a little while, let it run

INSTALLATIONS

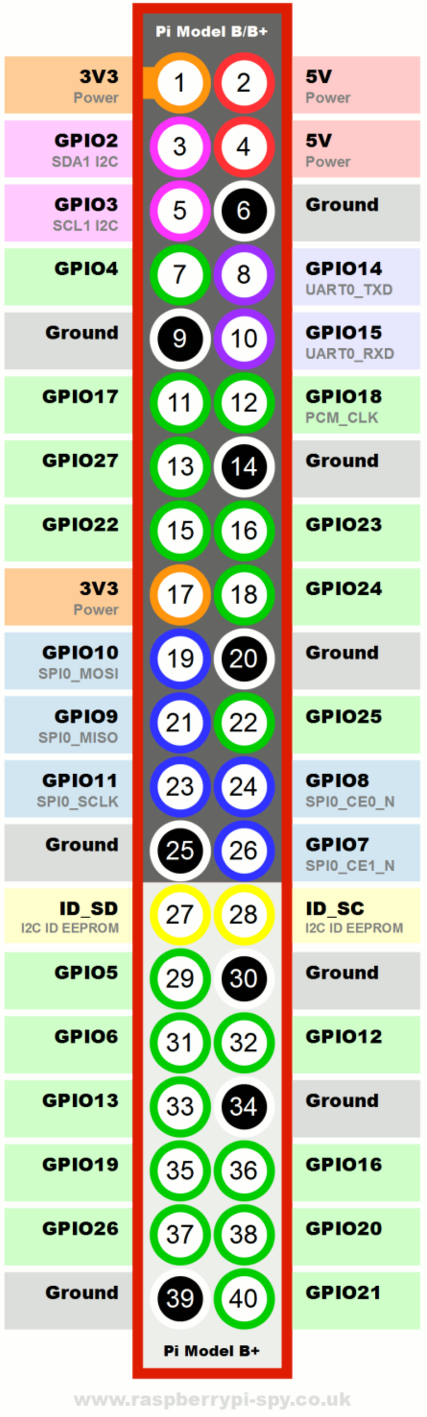
- Then we download
  - sudo apt install –y python-dev
  - sudo apt install –y python3-pip
  - sudo pip install readchar
  - sudo pip install bottle
- After installing the programs we need, we can now start coding for our raspberry pi

GPIO

- Stands for General Purpose Input Output
- They are the pins along the edge of the board
- These can be used for connecting and communicating with all manner of electronic components, acting as a physical interface between the Raspberry Pi and the outside world.
- We will use python to control Raspberry pi’s gpio



3.3V	1	2	5V
GPIO2 (SDA1)	3	4	5V
GPIO3 (SCL1)	5	6	GND
GPIO4 (GPIO_GCLK)	7	8	GPIO14 (UART_TXD0)
GND	9	10	GPIO15 (UART_RXD0)
GPIO17 (GPIO_GEN0)	11	12	GPIO18 (GPIO_GEN1)
GPIO27 (GPIO_GEN2)	13	14	GND
GPIO22 (GPIO_GEN3)	15	16	GPIO23 (GPIO_GEN4)
3.3V	17	18	GPIO24 (GPIO_GEN$)
GPIO10 (SPI0_MOSI)	19	20	GND
GPIO9 (SPI0_MISO)	21	22	GPIO25 (GPIO_GEN6)
GPIO11 (SPI0_CLK)	23	24	GPIO8 (SPI_CE0_N)
GND	25	26	GPIO7 (SPI_CE1_N)
ID_SD (I2C EEPROM)	27	28	ID_SC (I2C EEPROM)
GPIO5	29	30	GND
GPIO6	31	32	GPIO12
GPIO13	33	34	GND
GPIO19	35	36	GPIO16
GPIO26	37	38	GPIO20
GND	39	40	GPIO21



Pi Model B/B+

3V3 Power	1	2	5V Power
GPIO2 SDA1 I2C	3	4	5V Power
GPIO3 SCL1 I2C	5	6	Ground
GPIO4	7	8	GPIO14 UART0_TXD
Ground	9	10	GPIO15 UART0_RXD
GPIO17	11	12	GPIO18 PCM_CLK
GPIO27	13	14	Ground
GPIO22	15	16	GPIO23
3V3 Power	17	18	GPIO24
GPIO10 SPI0_MOSI	19	20	Ground
GPIO9 SPI0_MISO	21	22	GPIO25
GPIO11 SPI0_SCLK	23	24	GPIO8 SPI0_CE0_N
Ground	25	26	GPIO7 SPI0_CE1_N
ID_SD I2C ID EEPROM	27	28	ID_SC I2C ID EEPROM
GPIO5	29	30	Ground
GPIO6	31	32	GPIO12
GPIO13	33	34	Ground
GPIO19	35	36	GPIO16
GPIO26	37	38	GPIO20
Ground	39	40	GPIO21

Pi Model B+

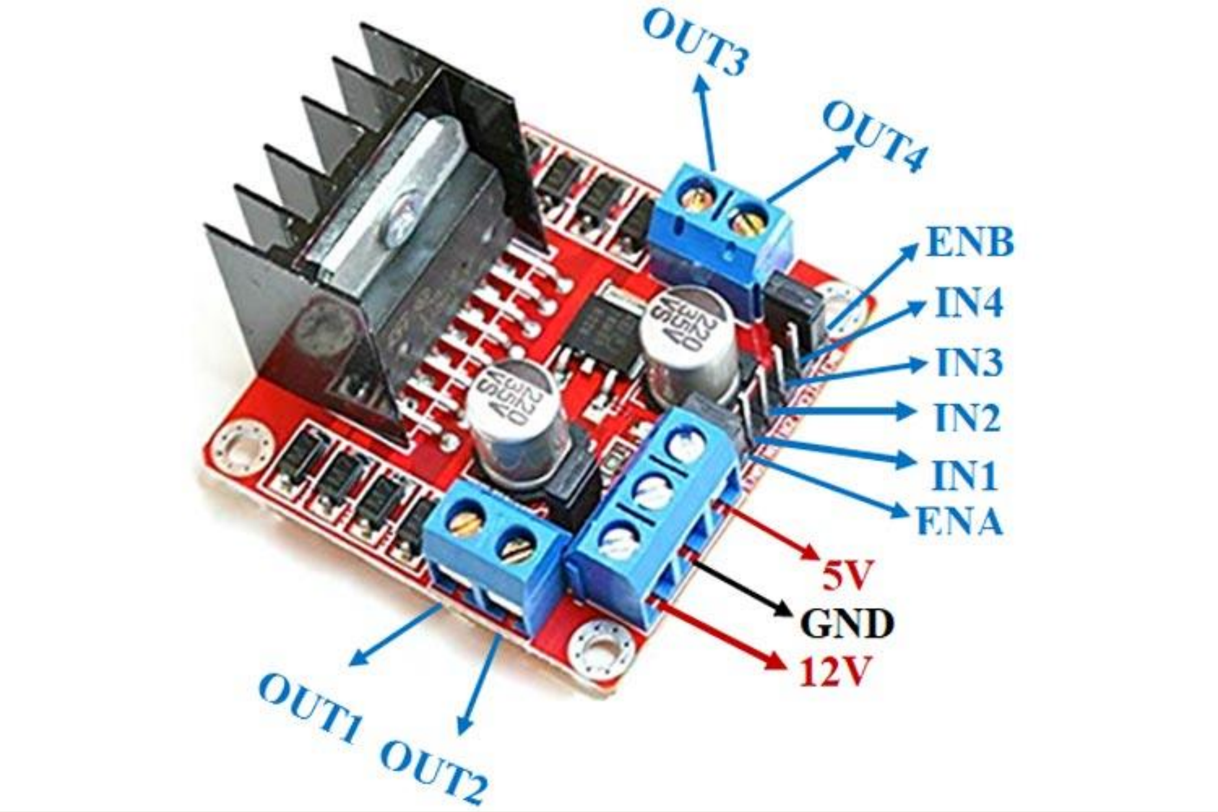
www.raspberrypi-spy.co.uk

BASICS OF PYTHON CODING

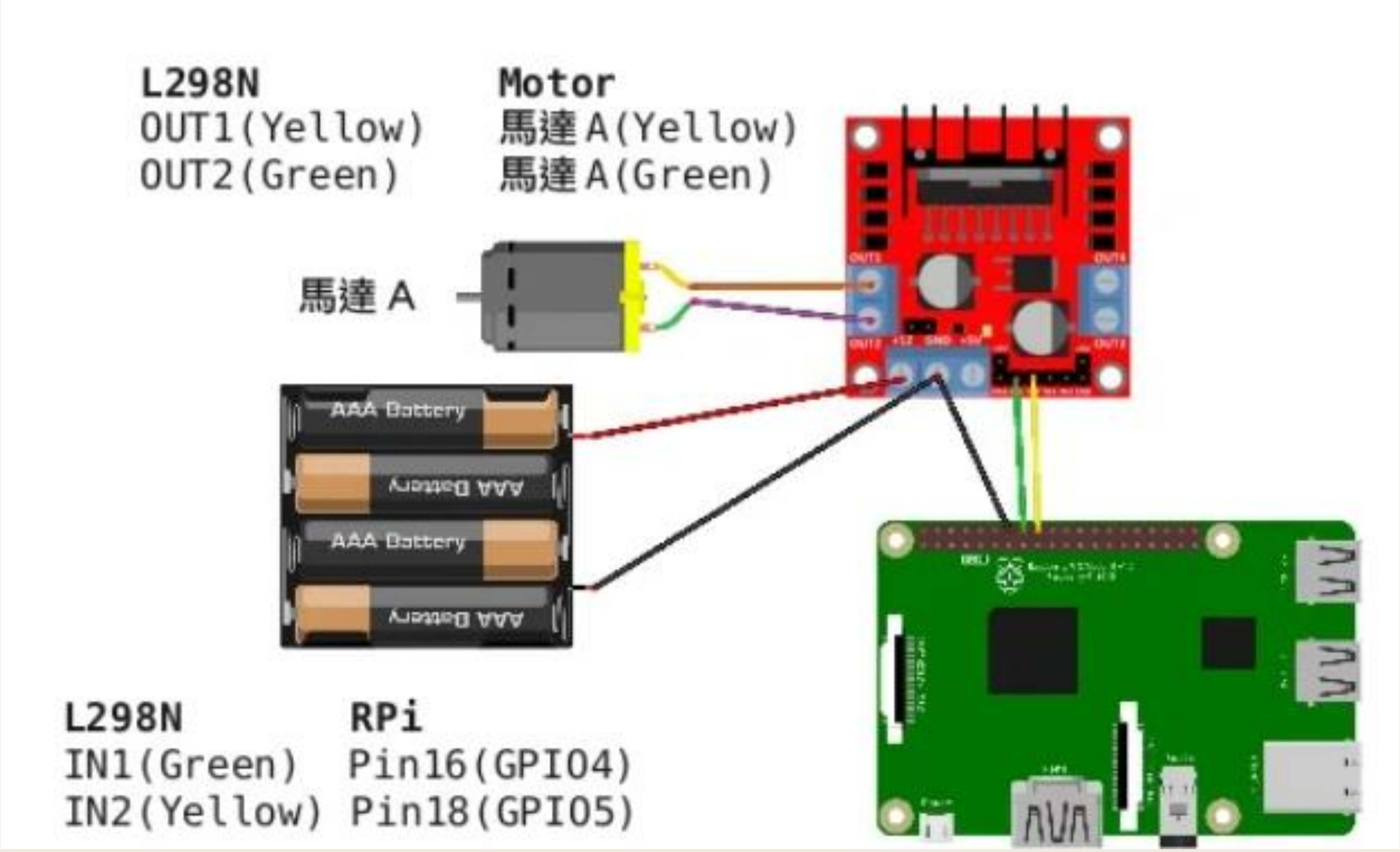
- For GPIO, some common things are:
  - Import module
  - Define pin number
  - Setup a channel
  - Input/Output
  - Cleanup

L298N MOTOR

Pin Name	Description
IN1 & IN2	Motor A input pins. Used to control the spinning direction of Motor A
IN3 & IN4	Motor B input pins. Used to control the spinning direction of Motor B
ENA	Enables PWM signal for Motor A
ENB	Enables PWM signal for Motor B
OUT1 & OUT2	Output pins of Motor A
OUT3 & OUT4	Output pins of Motor B
12V	12V input from DC power Source
5V	Supplies power for the switching logic circuitry inside L298N IC
GND	Ground pin



SO HOW TO CODE TO MAKE THE MOTOR MOVE?



L298N	Motor
OUT1(Yellow)	馬達A(Yellow)
OUT2(Green)	馬達A(Green)

L298N	RPi
IN1(Green)	Pin16(GPIO4)
IN2(Yellow)	Pin18(GPIO5)

CODING

- In your vnc, at home folder, make a new folder called: pi-follower-car
- Remember that naming and organizing is important in this project because we will be working with a lot of different files
- Inside your pi follower car folder, create another folder called 02-motor
- And inside 02-motor, create another folder called 02_2-l2908n_motor
- You will be saving your python file in 02_2-l2908n_motor

SO…

- In the basics we mentioned a few important steps that we need to remember cause it will be repeated many times
- First one is importing modules:
  - So in your new file that we will name: l298n_motor.py
  - import RPi.GPIO as GPIO
  - Import time
- We mentioned a little about time module and for here we will be using time.sleep() most of the time

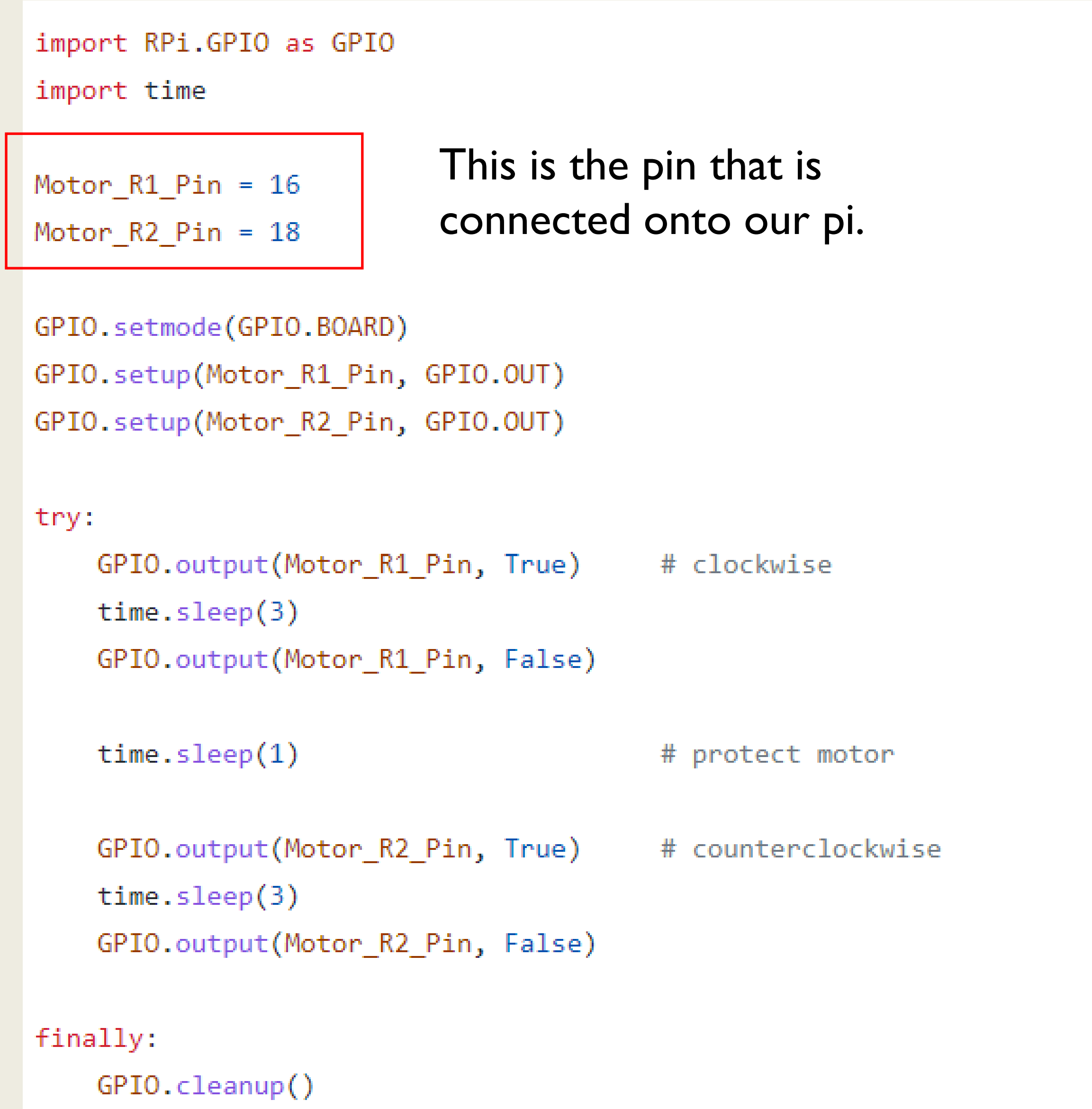
TRY EXCEPT ELSE FINALLY

- The try block lets you test a block of code for errors.
- The except block lets you handle the error.
- The else block lets you execute code when there is no error.
- The finally block lets you execute code, regardless of the result of the try- and except blocks.

EXAMPLE:

-
  ```
  try:
    print(x)
  except:
    print("An exception occurred")
  ```
- Will this code work?
- The answer should be no, the try part won’t work so it will print out “An exception occurred”
- It will still run even if there is an error.

EXAMPLE OF L298N_MOTOR.PY



```
import RPi.GPIO as GPIO
import time

Motor_R1_Pin = 16
Motor_R2_Pin = 18


GPIO.setmode(GPIO.BOARD)
GPIO.setup(Motor_R1_Pin, GPIO.OUT)
GPIO.setup(Motor_R2_Pin, GPIO.OUT)

try:
    GPIO.output(Motor_R1_Pin, True)     # clockwise
    time.sleep(3)
    GPIO.output(Motor_R1_Pin, False)

    time.sleep(1)                       # protect motor

    GPIO.output(Motor_R2_Pin, True)     # counterclockwise
    time.sleep(3)
    GPIO.output(Motor_R2_Pin, False)

finally:
    GPIO.cleanup()
```

This is the pin that is connected onto our pi.

- We have modules, variables and funcitons
- We also have try, finally loop
- You will see setmode, setup, output

- GPIO.setmode means setting up our board(the pins on the pi)
- Setup: meaning we tell raspberry which pin we are using and what we want to output
- Output, we're going to output and if it is true or false.
- Cleanup means when we're done, we clean up or reset.

HOW DO WE CONTROL THE MOTOR'S SPEED?

- We control it by defining how much voltage we input and output
- We can also change the amount of batteries
- We can add variable resistance
- We can change the pulse width modulation
  - This is done to reduce the average power delivered by an electrical signal
  - And we will be using the method.

CODING FOR PWM

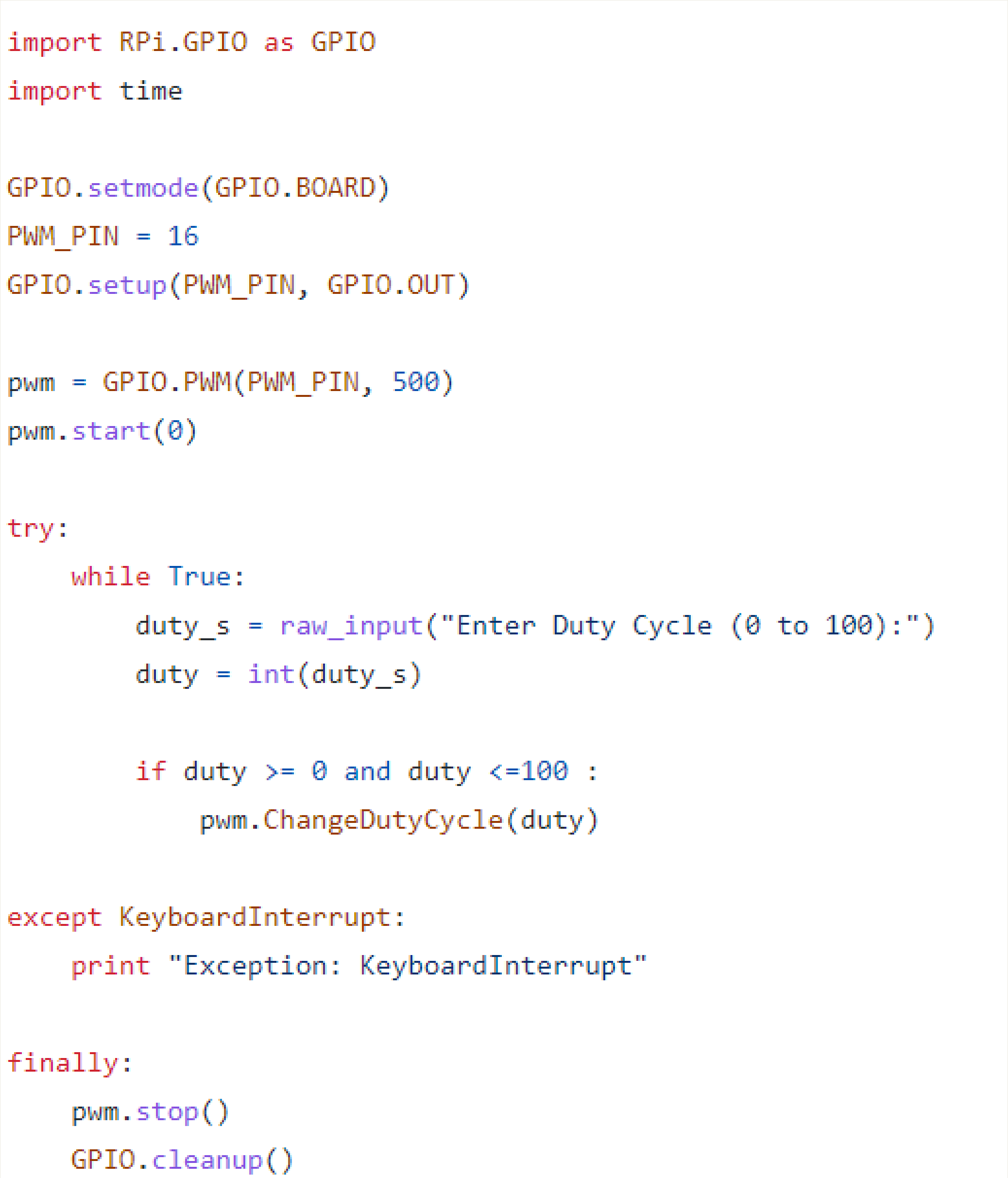
- Name your new file as pwn_l298n.py
- Let’s try a few lines ourselves since we will be repeating some of it from our previous file.
- Import two modules and setmode the board
- First we identify the pwm pin
- GPIO18 is a PWM pin as well so we will write:
    - PWM_PIN = 12
    - Then we set up GPIO

PWM

- Instead of setting up the GPIO board, we will set up the pwm pin and gpio.out
- We want to tell pwm how much signal we want going out
- Then we will start it
- Here we will use something called duty cycle
- The duty cycle describes the amount of time the signal is in a high (on) state as a percentage of the total time of it takes to complete one cycle.
- The frequency determines how fast the PWM completes a cycle.

PWM_L298N.PY

- This is what PWM file looks like
- KeyboardInterrupt is when a key is pressed then the program will stop running



```python
import RPi.GPIO as GPIO
import time

GPIO.setmode(GPIO.BOARD)
PWM_PIN = 16
GPIO.setup(PWM_PIN, GPIO.OUT)

pwm = GPIO.PWM(PWM_PIN, 500)
pwm.start(0)

try:
    while True:
        duty_s = raw_input("Enter Duty Cycle (0 to 100):")
        duty = int(duty_s)

        if duty >= 0 and duty <=100 :
            pwm.ChangeDutyCycle(duty)

except KeyboardInterrupt:
    print "Exception: KeyboardInterrupt"

finally:
    pwm.stop()
    GPIO.cleanup()
```

HOW TO CONTROL THE CAR?

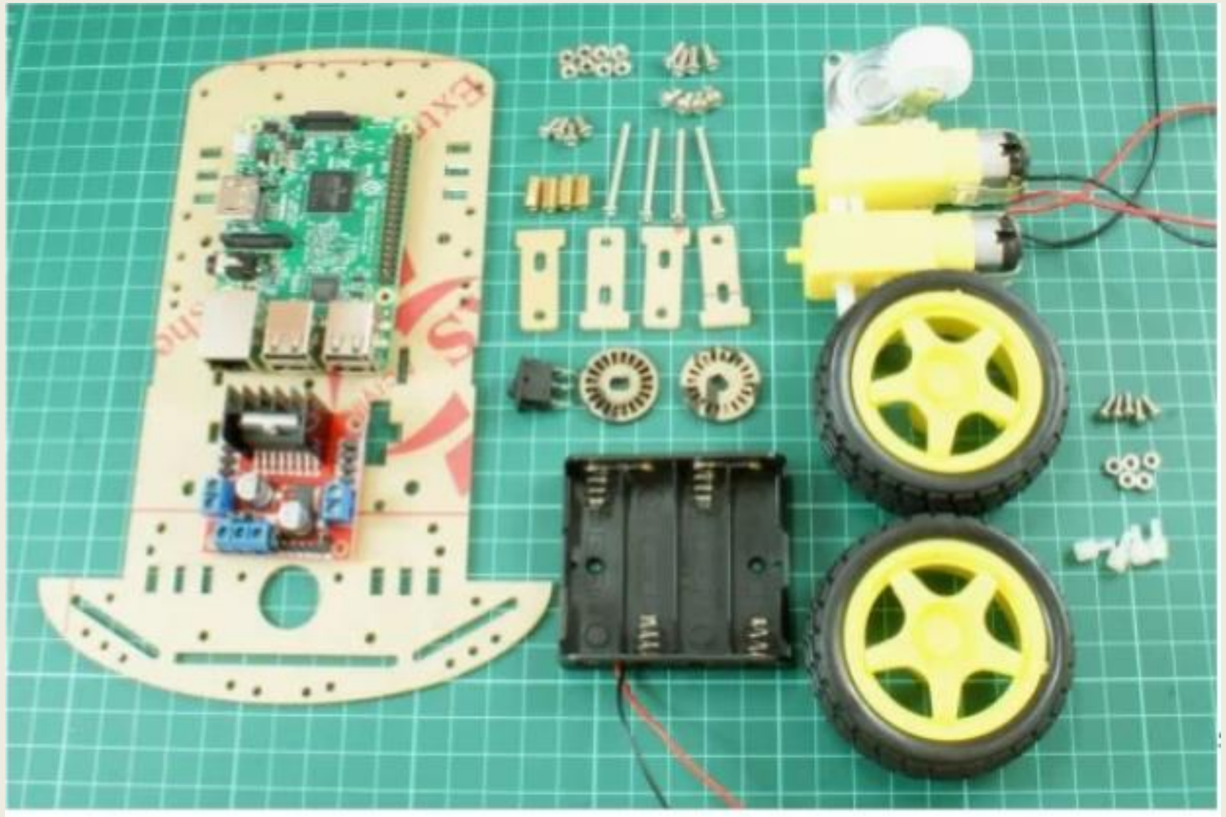
- By using:
  - Motor
  - Wiring
  - Battery(Power)
  - Control board (raspberry pi in our case)

LET’S PUT TOGETHER OUR CAR

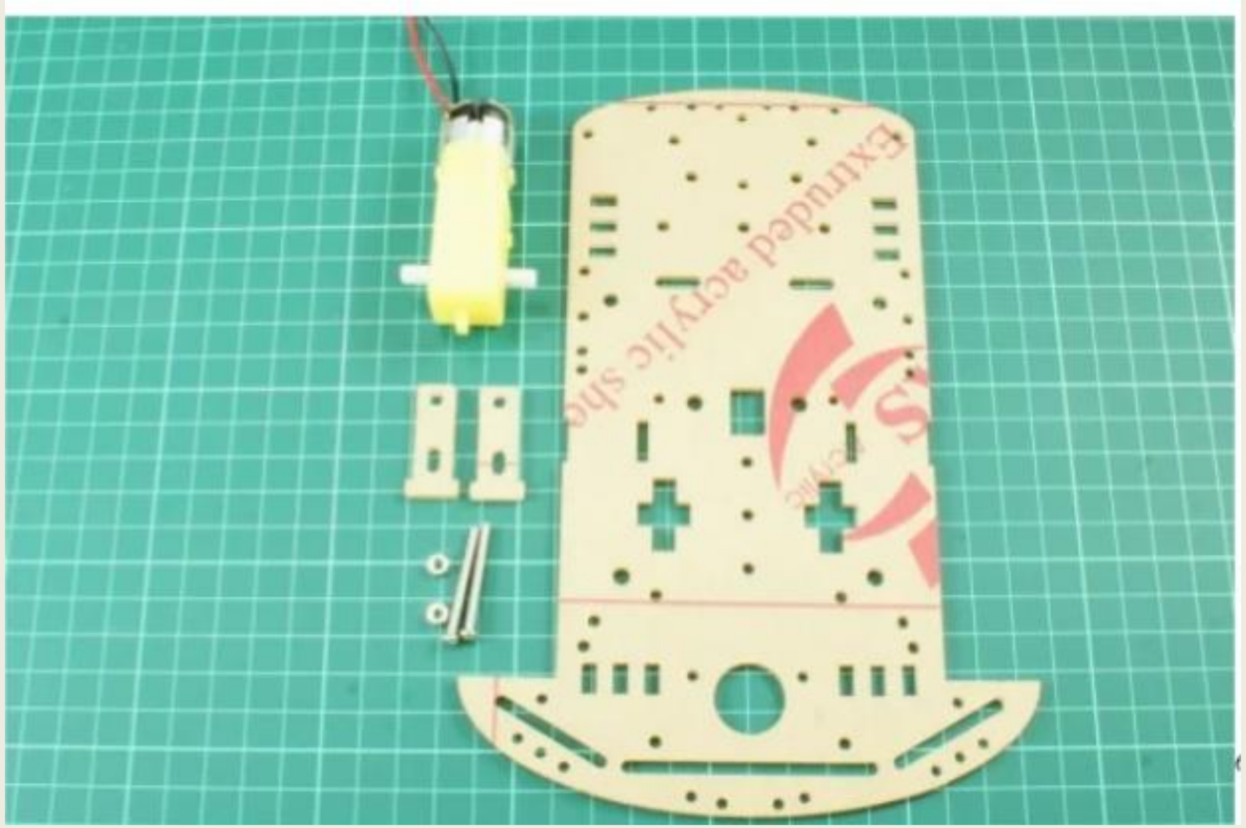
REMINDER

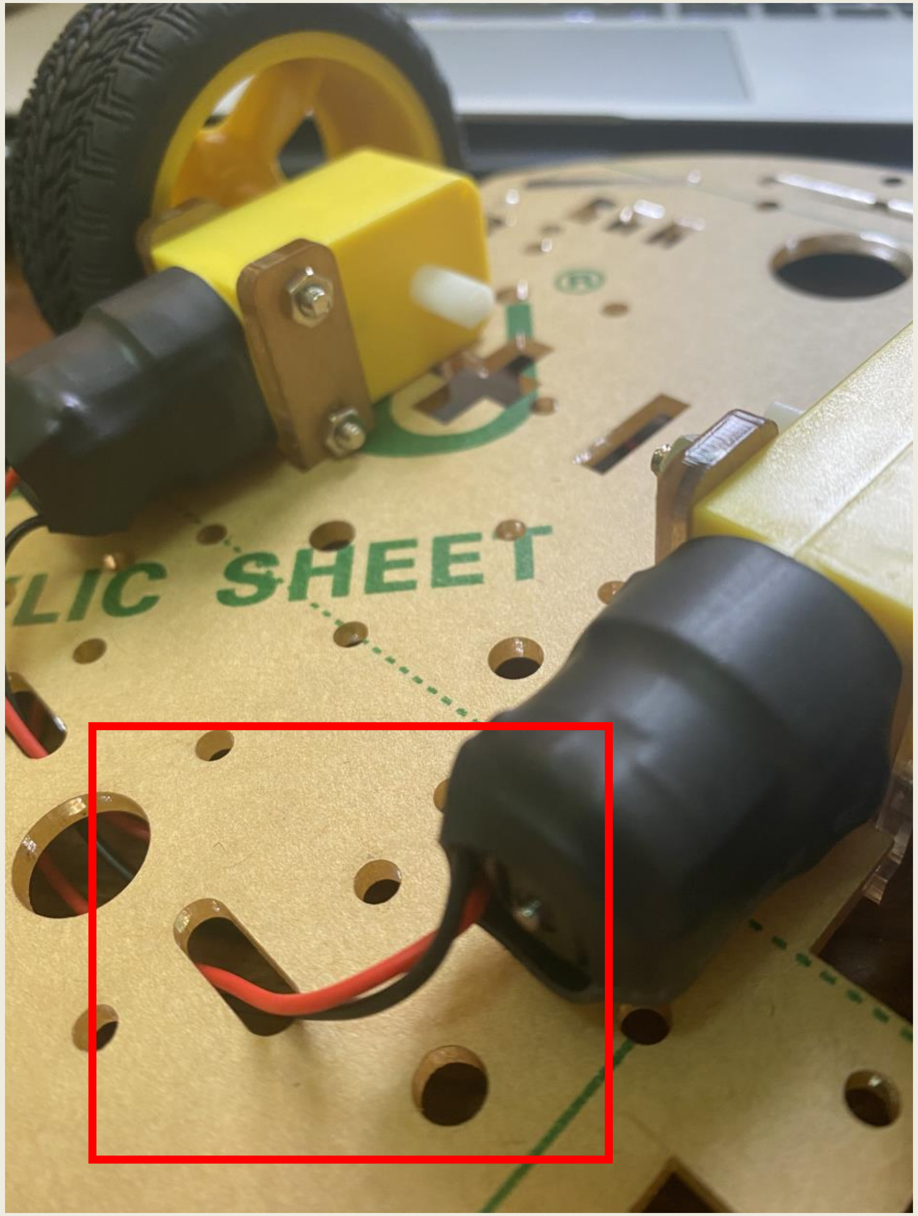
- Please remember that everything is small and delicate so be careful with your pieces!
- If you need help please tell us and we'll help you
- If we need to share the tools, wait patiently

THE CAR

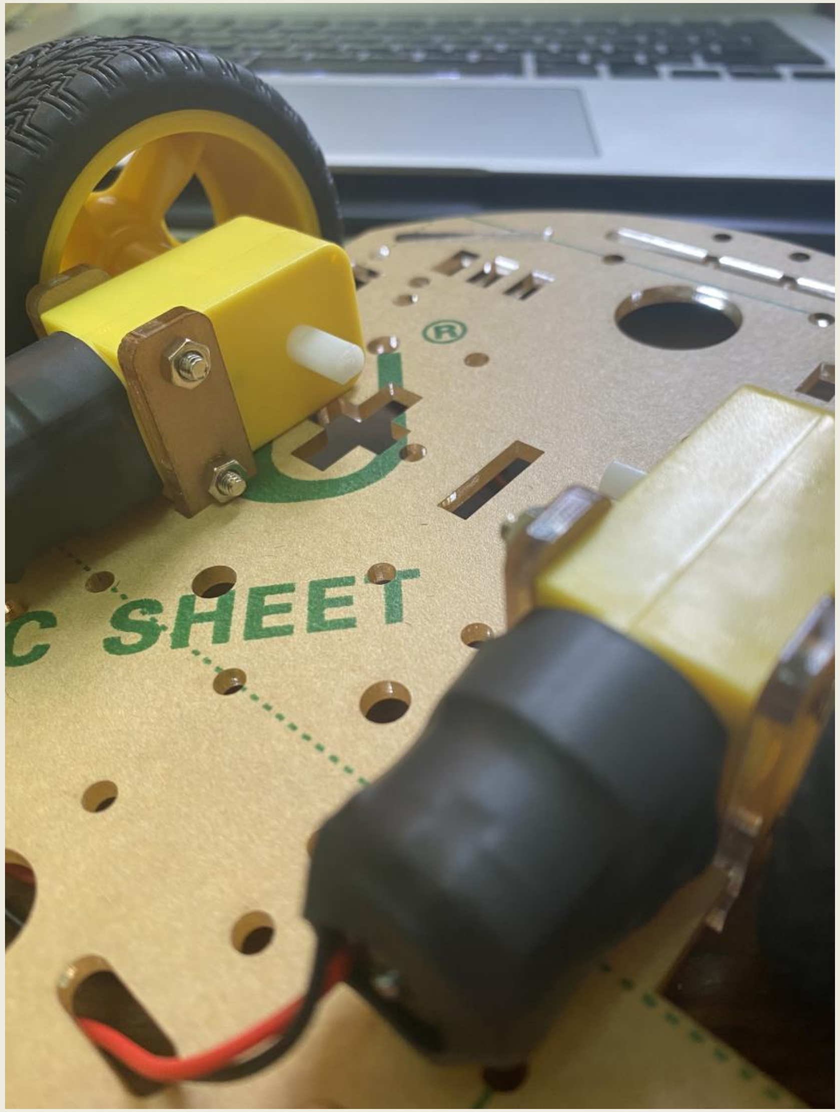


THE CAR

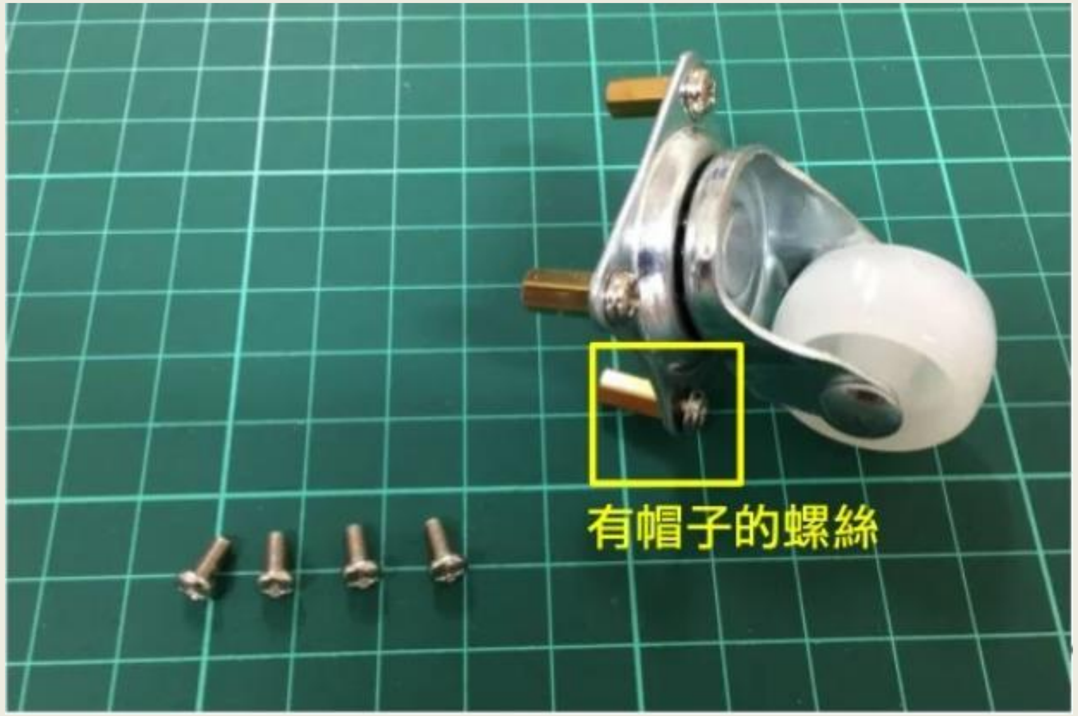


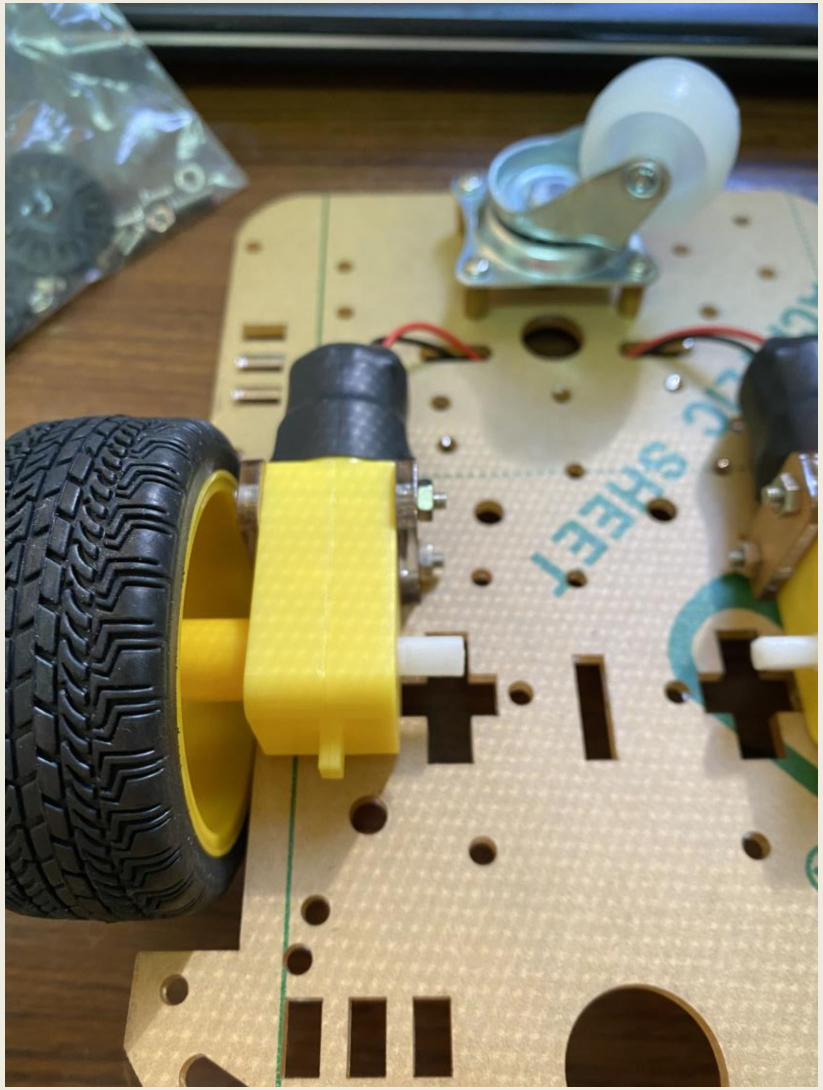


THE CAR

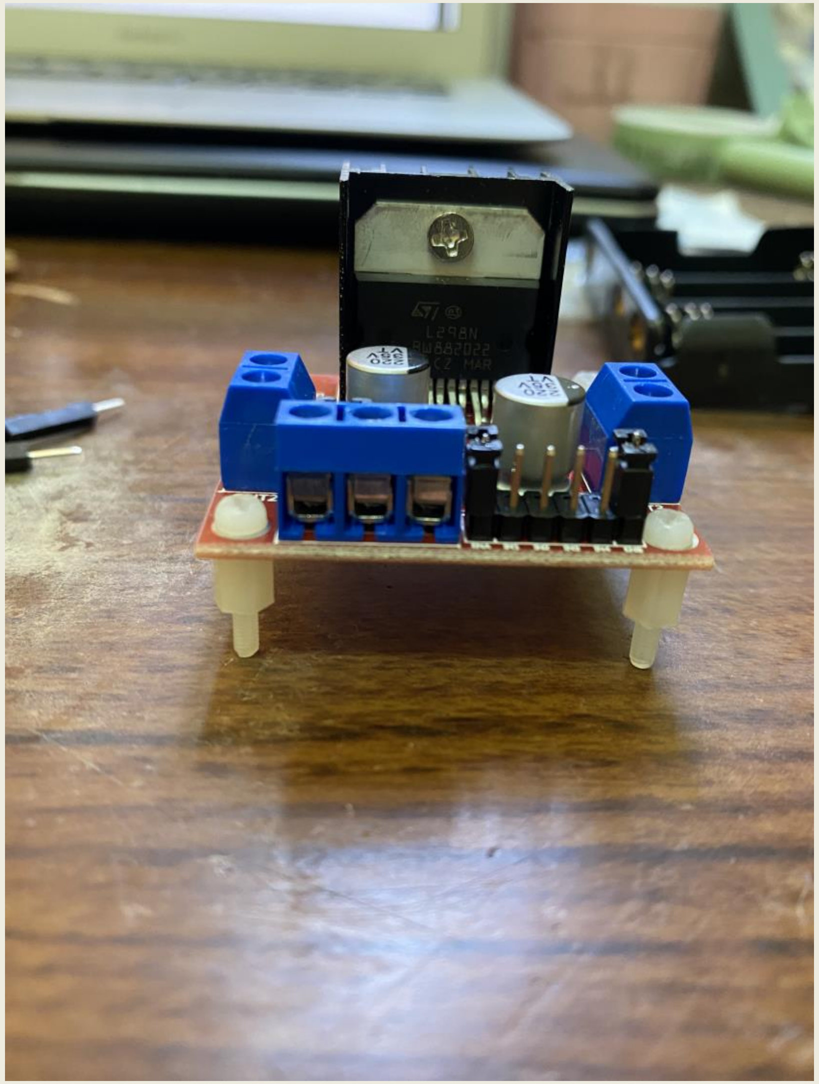


CAR

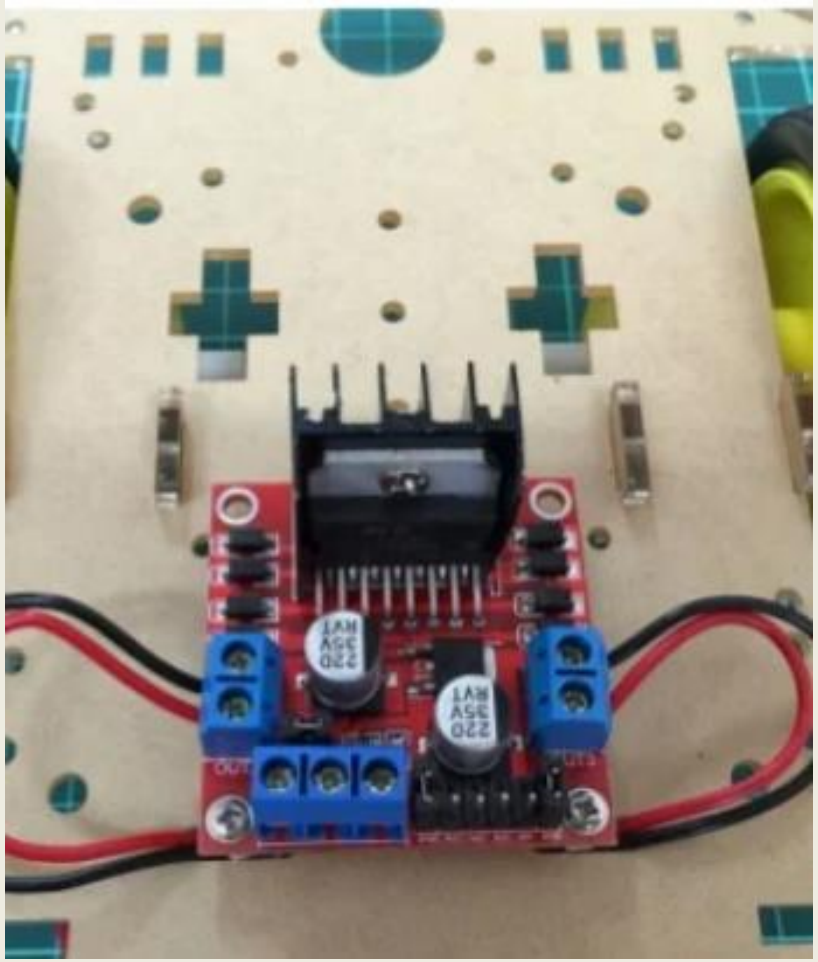




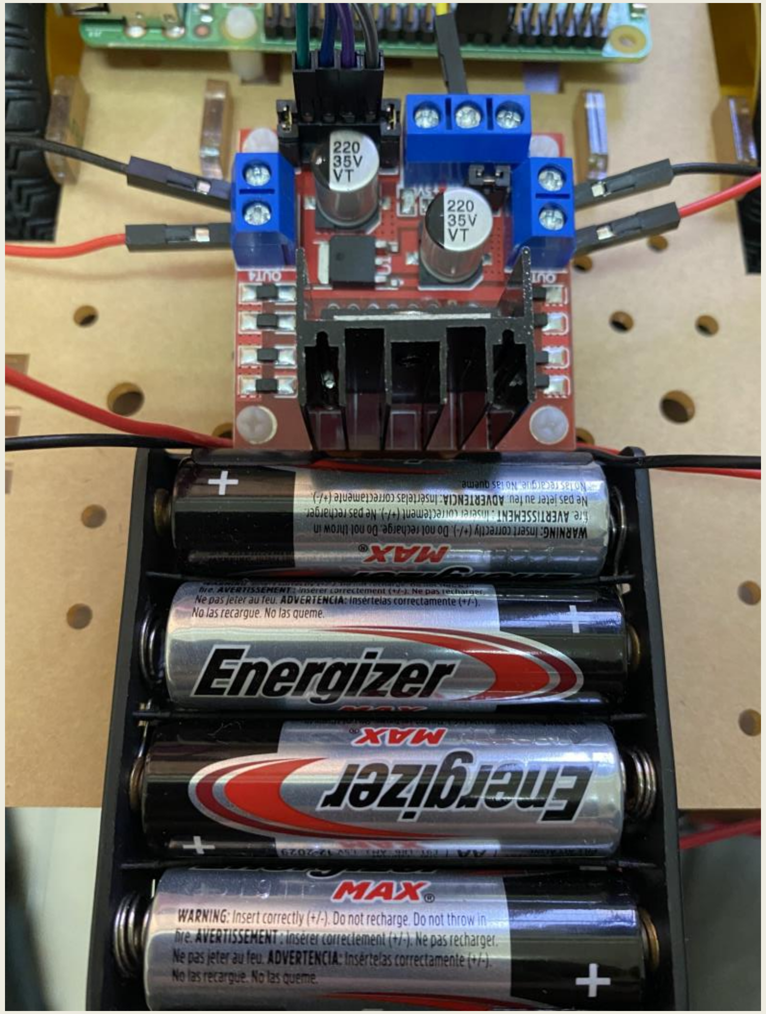
L298N

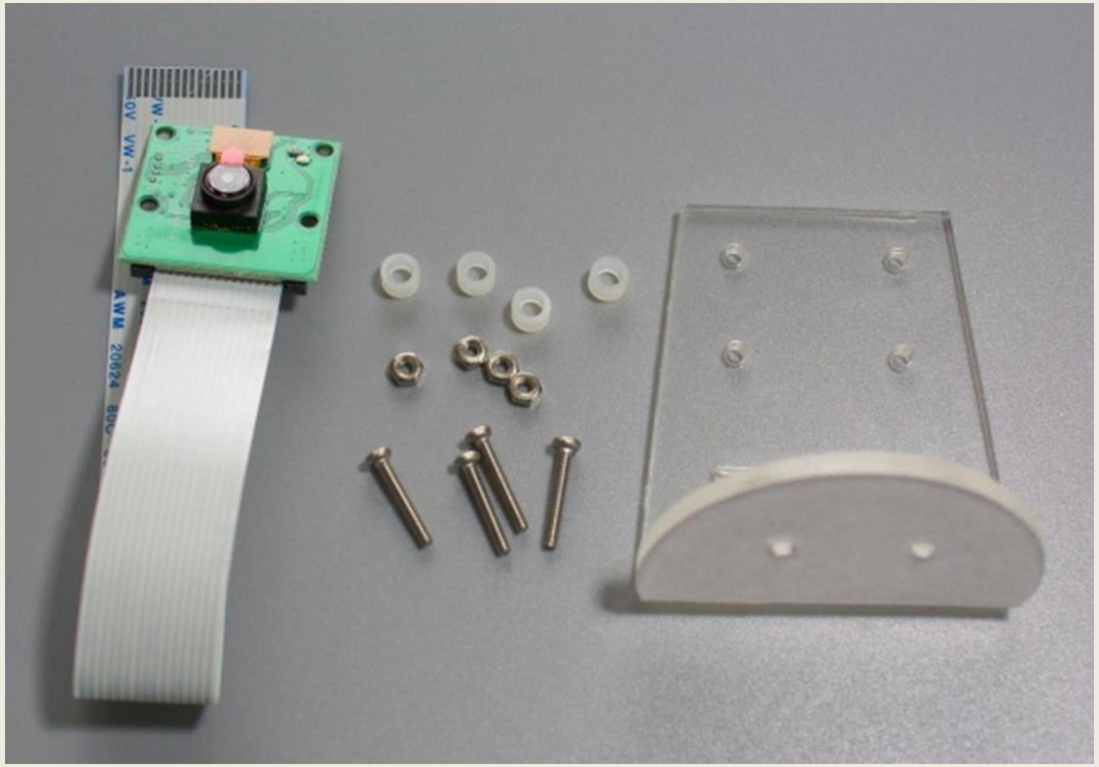


CAR



CAR


220
35V
VT
220
35V
VT
Energizer
MAX
WARNING: Insert correctly (+/-). Do not recharge. Do not throw in
fire. AVERTISSEMENT : Insérer correctement (+/-). Ne pas recharger.
Ne pas jeter au feu. ADVERTENCIA: Insértelas correctamente (+/-).
No las recargue. No las queme.



CAR

FINAL

- For now, we can put the camera and raspberry pi on the side
- We will start connecting them when we start coding our car