RASPBERRY PI

DAY 2

EMILY WENG

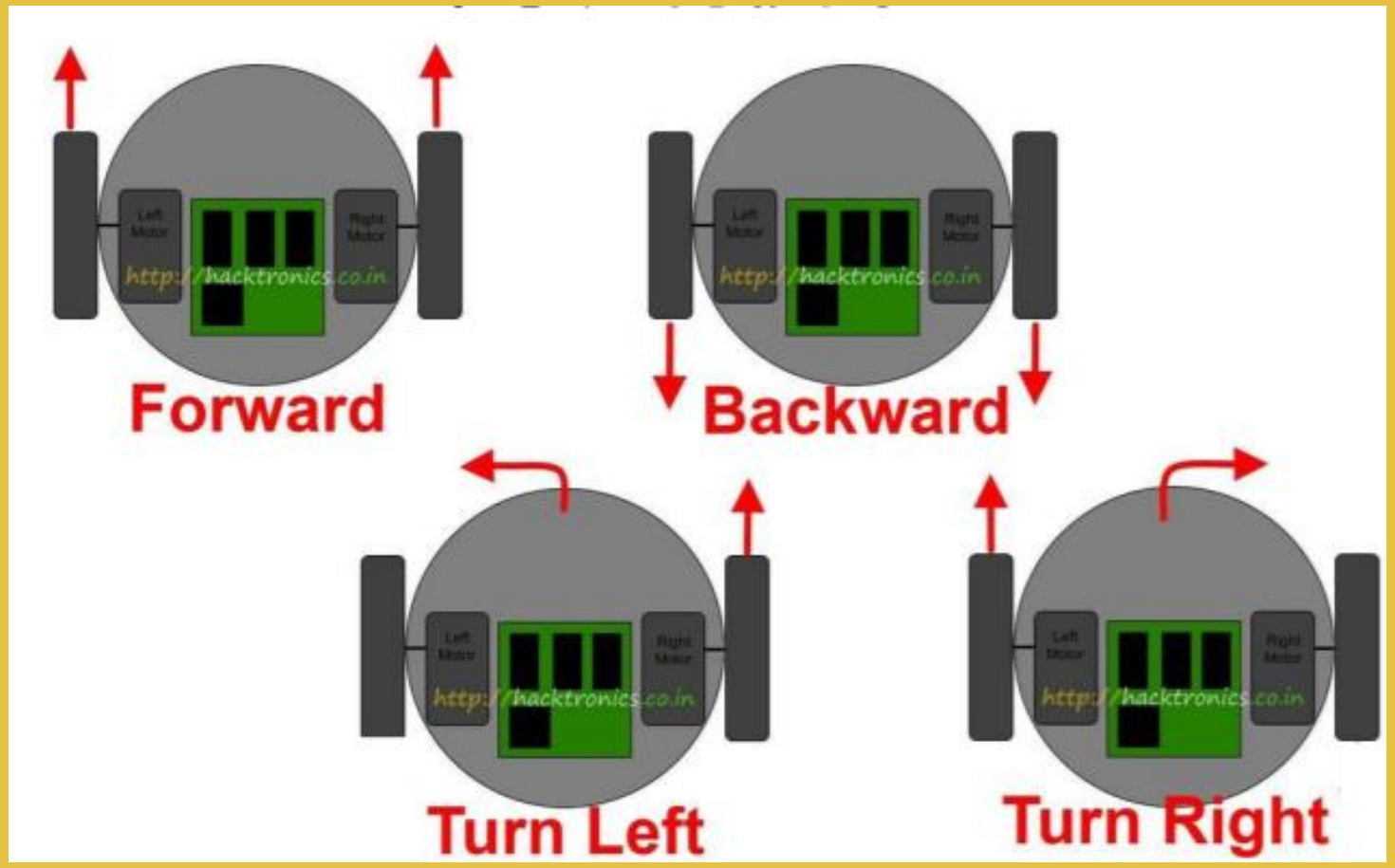
WHAT HAVE WE DONE SO FAR?

- Connected our pi to the internet
- We learned a bit how the coding in raspberry pi works
- We put together our car

WHAT IS NEXT?

- CODING
- We will be connect our car to raspberry pi so it can move
- We will also be learning how to use our camera
- Final setup and debugging

HOW DOES THE CAR MOVE WITH L298N MOTOR?

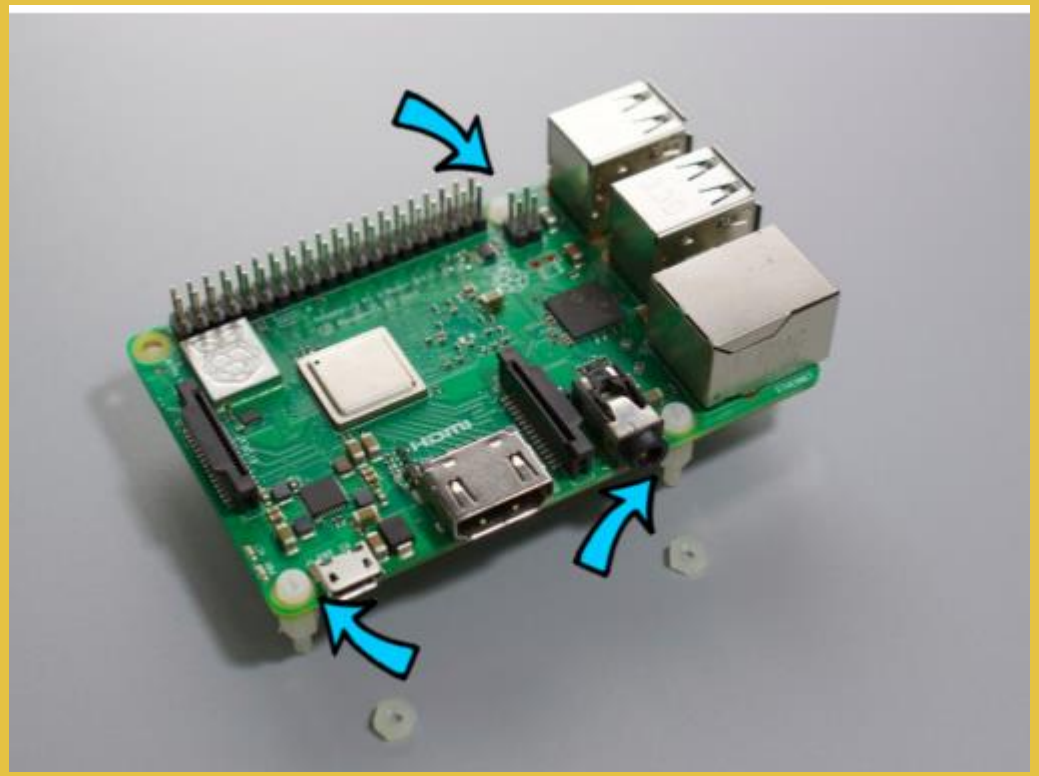


NEXT STEP

- We'll need four of this
- We'll connect this to L298N and Raspberry pi.
- This is important because you need to make sure the pin you connect to is in the right place on the motor as well

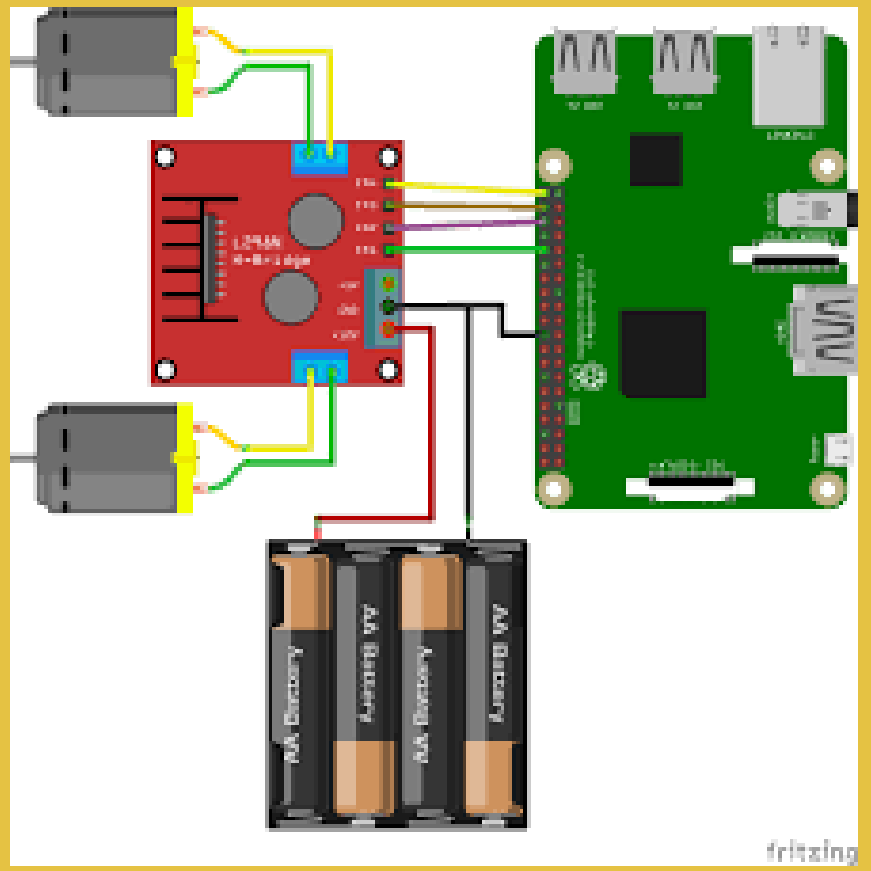
CONNECT TO CAR

- We can now put our raspberry pi onto our car so it is easier to work the car with the motor

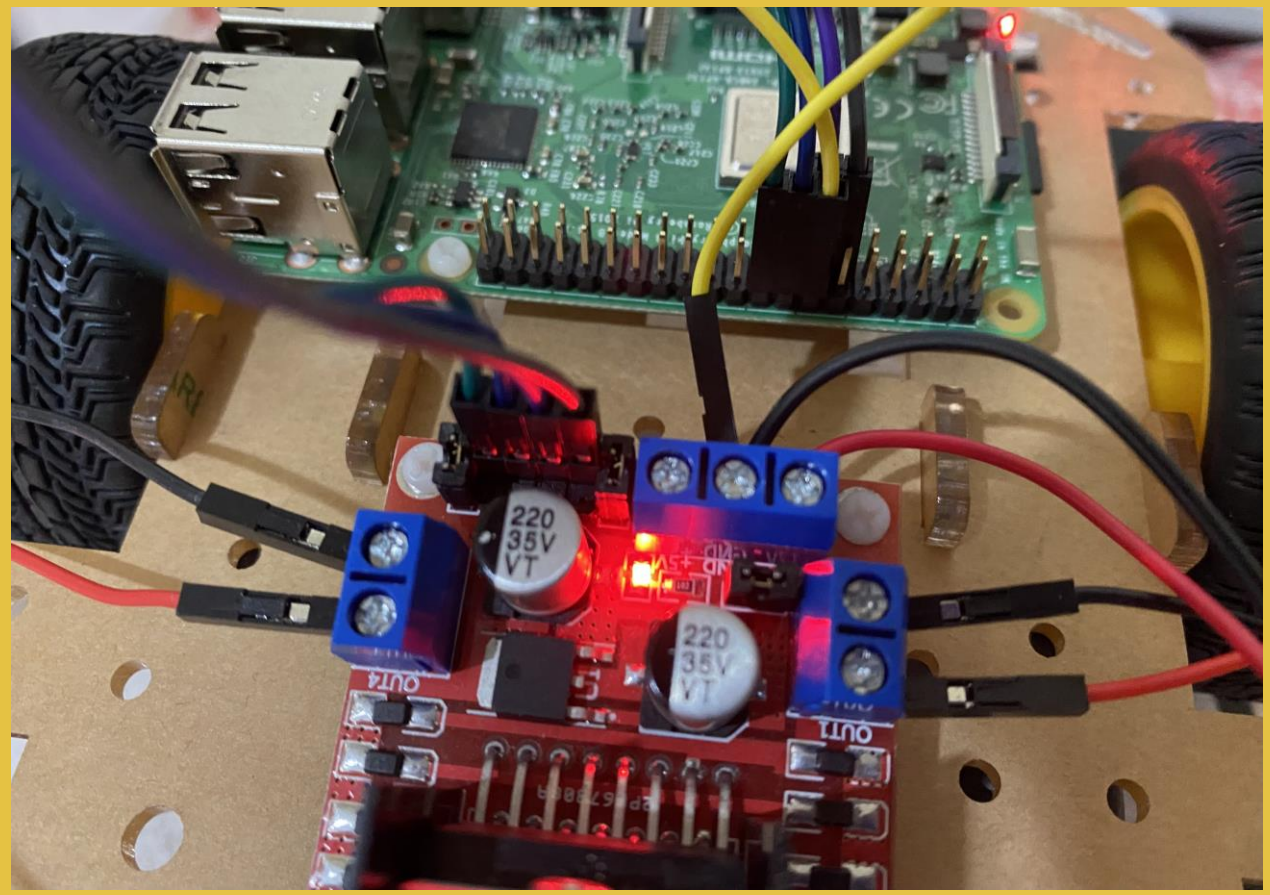


CONNECTING

- We need to connect to pin: 16, 18, 11, 13
- But we need to match it to the correct IN as well
  - INs are the ones on L298N
- IN1(Green) Pin16
- IN2(Yellow) Pin18
- IN3(Purple) Pin11
- IN4(Orange) Pin13
- Colors may vary



END RESULTS



CODING

- We're going to code how to move our car
- Import 3 modules
  - Do you remember the other two?
  - We're going to add one we didn't add last time:
  - readchar
  - This allows the program to know which key on the keyboard to execute

MOVE_CAR.PY

- We're going to create 5 functions and loops.
- I will show you how to define moving forward
- First off, we need to list out the pins we used in our raspberry pi and set as variables
- For example:
  - Motor_R1_Pin = 16
  - Motor_R2_Pin = 18
  - Try the motor for the left side
- Set a timer
  - t=0.5

MOVE_CAR.PY

- We need to set up our GPIO.BOARD too
- Next we set up our motor pin as output
- We'll need to set at the start that GPIO is turned off

```
29 GPIO.setmode(GPIO.BOARD)
30 GPIO.setup(Motor_R1_Pin, GPIO.OUT, initial=GPIO.LOW)
31 GPIO.setup(Motor_R2_Pin, GPIO.OUT, initial=GPIO.LOW)
32 GPIO.setup(Motor_L1_Pin, GPIO.OUT, initial=GPIO.LOW)
33 GPIO.setup(Motor_L2_Pin, GPIO.OUT, initial=GPIO.LOW)
```

DEF FORWARD():

- First we need to define how does forward work.
- We need to make sure if it is output or input
- In this case it's output cause you give it command from raspberry pi to move
- But we have two pin for one wheel, we only need one to work
- We also need it to pause for a little bit so we can give it time to move.

```
def forward():
    GPIO.output(Motor_R1_Pin, True)
    GPIO.output(Motor_R2_Pin, False)
    GPIO.output(Motor_L1_Pin, True)
    GPIO.output(Motor_L2_Pin, False)
    time.sleep(t)
    stop()
```

TRY BACKWARDS

- Backwards look the same but we change the True and False of the pin
- Switch the true and false and remember time.sleep
- How about right? And left?
- How do you think it will look like?
- We also need a function for stop
- What would stop look like?

RESULTS

```
def backward():
    GPIO.output(Motor_R1_Pin, False)
    GPIO.output(Motor_R2_Pin, True)
    GPIO.output(Motor_L1_Pin, False)
    GPIO.output(Motor_L2_Pin, True)
    time.sleep(t)
    stop()
```

```
def turnRight():
    GPIO.output(Motor_R1_Pin, True)
    GPIO.output(Motor_R2_Pin, False)
    GPIO.output(Motor_L1_Pin, False)
    GPIO.output(Motor_L2_Pin, False)
    time.sleep(t)
    stop()
```

```
def turnLeft():
    GPIO.output(Motor_R1_Pin, False)
    GPIO.output(Motor_R2_Pin, False)
    GPIO.output(Motor_L1_Pin, True)
    GPIO.output(Motor_L2_Pin, False)
    time.sleep(t)
    stop()
```

```
def stop():
    GPIO.output(Motor_R1_Pin, False)
    GPIO.output(Motor_R2_Pin, False)
    GPIO.output(Motor_L1_Pin, False)
    GPIO.output(Motor_L2_Pin, False)
```

READCHAR

- Now, we're not done yet
- Add a if __name__ == "__main__"
- This is used just to check if the program is used as a module or not, and therefore decides whether to run the code.
- Print("press q to quit")
- We'll use a while True:
  - Ch=readchar.readkey()
  - This uses the function from readchar module

READCHAR

- We'll use w,a,s,d as our arrow keys
- If the key equals the key we pressed then it will execute a function
- So w= the function forward
- With our module it'll look like this:
  - If ch=='w':
    - Forward()
- Now try the rest with backward(), left(), right()

MOVE_CAR.PY

- After w, a, s, d, we also have q for quit.
- So when ch==‘q’:
  - Print(“quit”)
  - Cleanup
  - And quit()

RESULTS

```
if __name__ == "__main__":

    print ("Press 'q' to quit...")

    while True:
        ch = readchar.readkey()

        if ch == 'w':
            forward()

        elif ch == 's':
            backward()

        elif ch == 'd':
            turnRight()

        elif ch == 'a':
            turnLeft()

        elif ch == 'q':
            print ("\nQuit")
            GPIO.cleanup()S
            quit()
```

CAMERA

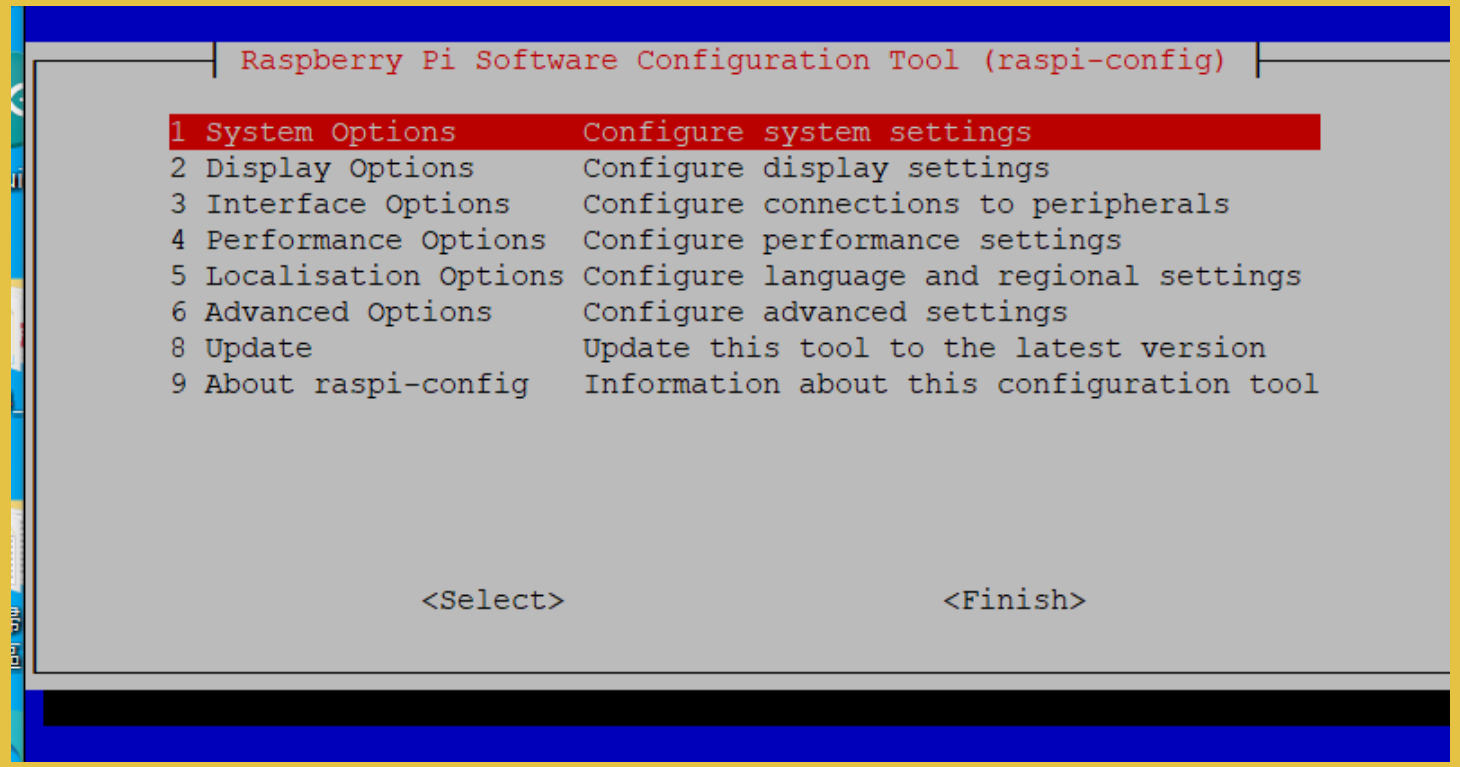
CAMERA

- First, we power off our raspberry pi
  - Sudo poweroff
- Insert the tip of the camera into your raspberry pi
- Today, we're using the raspberry pi camera module v1

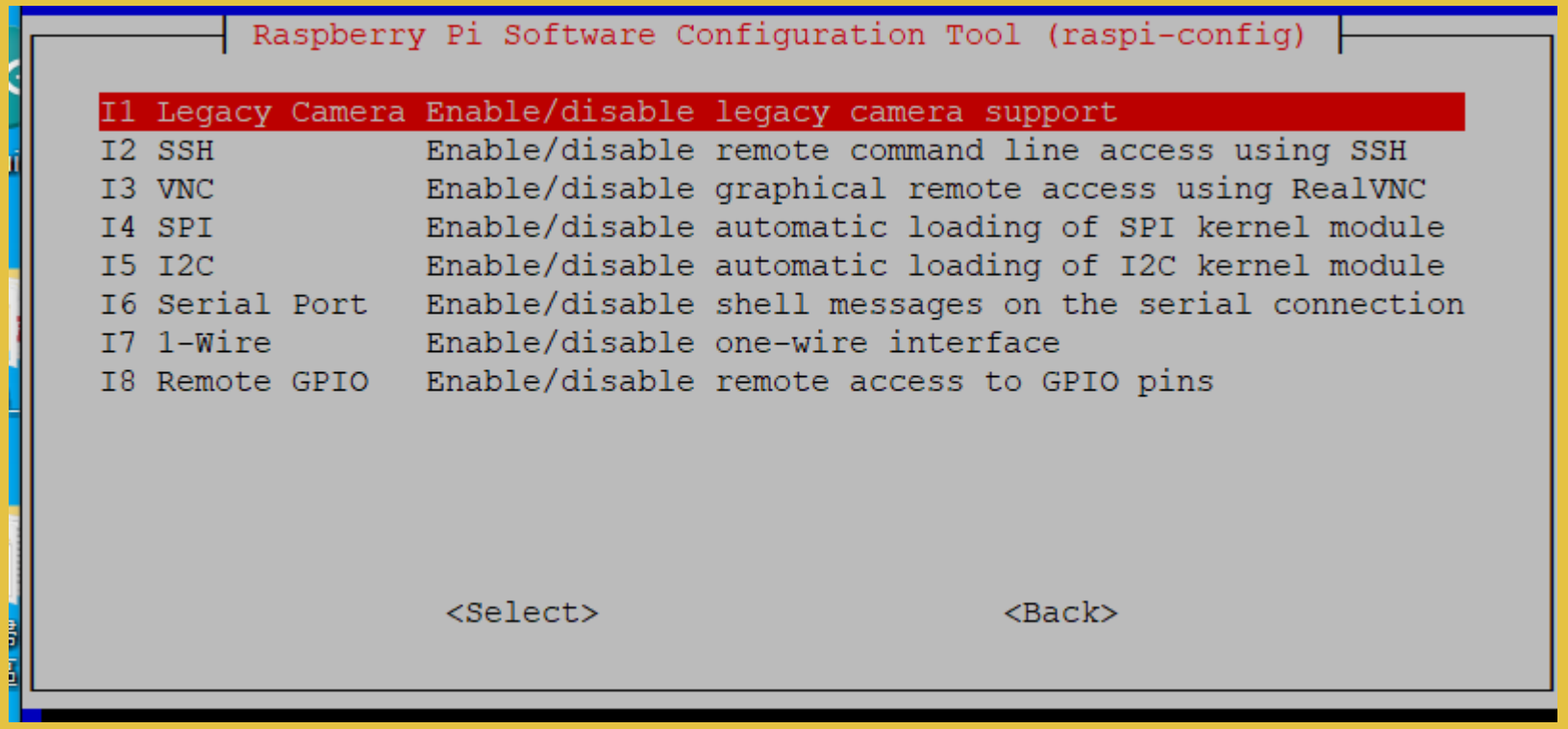
SETTING UP

- The camera should light up when you power on
- Next we can go back to VNC and open up our Raspberry window
- Enter the command:
  - sudo raspi-config
- You'll be taken to a settings menu

FIRST WE GO TO INTERFACING OPTIONS AND ENTER (THIRD ONE)



NEXT WE'LL SEE A CAMERA OPTIONS



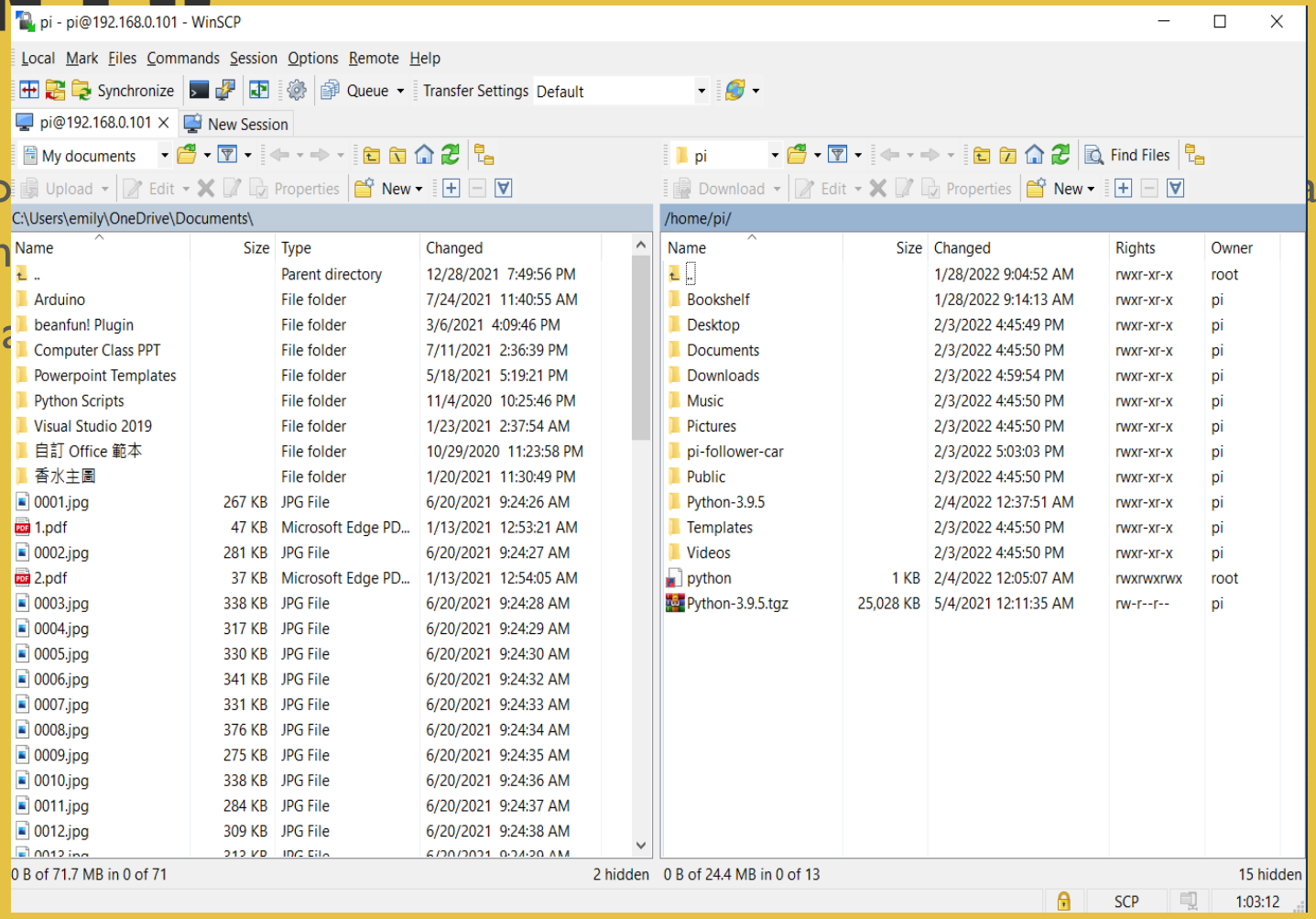
PRESS ENTER AND IT WILL ASK IF YOU WANT TO ENABLE IT

SETTING UP

- Once you enter yes, it will confirm
- Next we close it and reboot
- For our camera, we need to download winscp from this link:
- https://winscp.net/eng/download.php
-

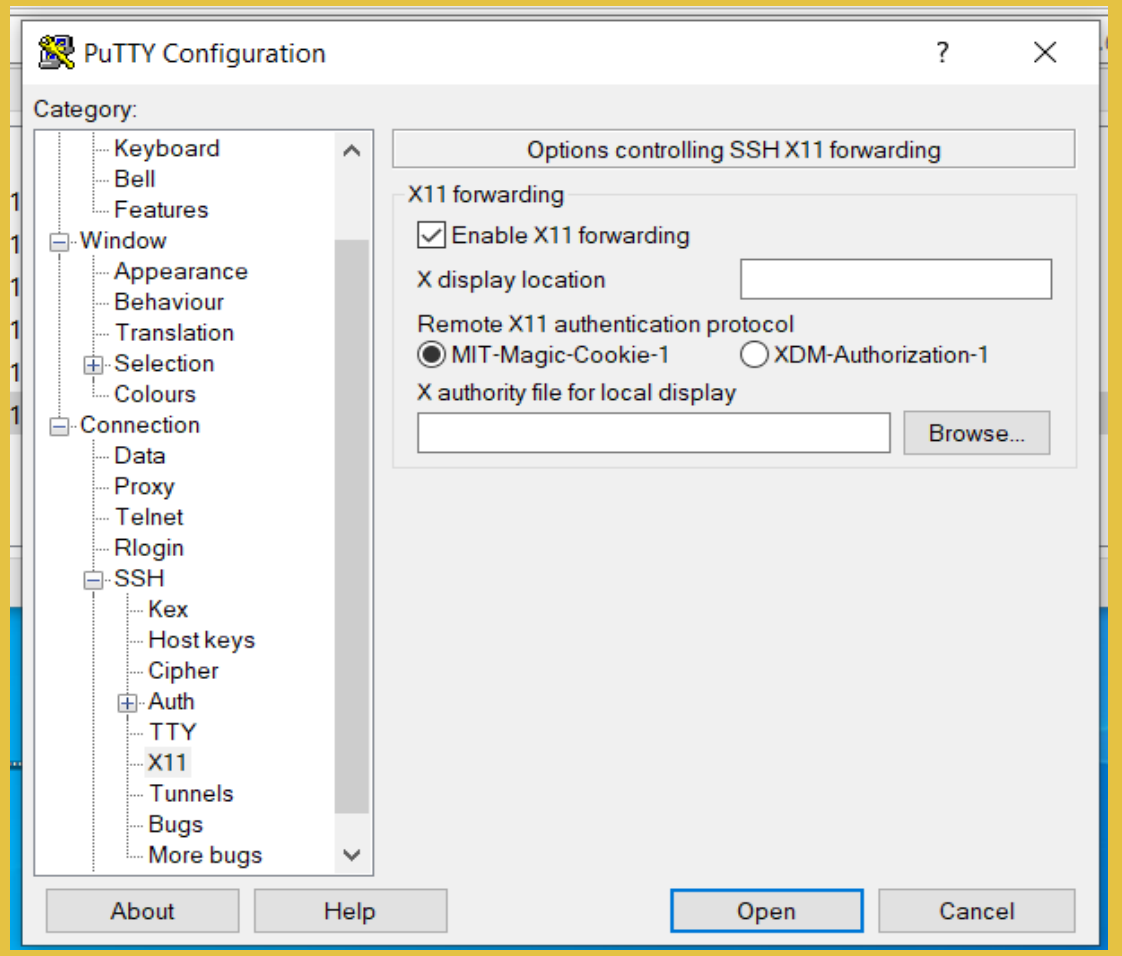
SENDING PI'S FILE BACK TO YOUR OWN PC

- After do … aspberry pi and usernam…
- After tha…



USING X11 FORWARDING

- Next we need to download another program
- https://sourceforge.net/projects/xming/
- After download, go to putty and on the left side, there is a menu
- Click on SSH, then X11, and check enable X11 forwarding



USING X11 FORWARDING

- Putty should open to pi's login page, enter your user and password
- Next type in the command:
  - gpicview test.jpg

- But technically, if you don't have a test.jpg then there won't be a picture. You can use a picture to test it out

DIRECT CAPTURE MODE FOR CAMERA

- Reboot raspberry pi
- Use VNC to log in but this time add 5900 behind your ip address
  - Ex: 192.168.0.101:5900
- Log in and go the options in VNC raspberry pi
- You should be able to right click it and see the options
- From the left side menu, go to trouble shooting and check:
  - Enable direct capture mode

DIRECT CAPTURE MODE FOR CAMERA

- Then, we can run a command called:
  - Raspistill
- We'll try two commands:
  - raspistill -o test.jpg
  - raspistill -t 3000 -o test.png -e png -w 640 -h 480
- This will allow your camera to take pictures
- To check you can go to files and you'll see the images you took

OPENCV

- Now, we'll be going into OpenCV
- Download this module in terminal
  - sudo modprobe bcm2835-v4l2
- We'll be using color in order to control the car
- We'll create three new files called follower_car.py, dc_motor.py and pwm_motor.py
- These three will help move our car
- Dc motor and pwm motor are considered modules and follower is the main program

DC_MOTOR.PY

- Dc motor is taking the move_car file without the if loop.
- So basically we can copy the functions from the move_car file
- This tells us which pin works when we want our car to move.

PWM_MOTOR.PY

- This controls the pulse of the motor.
- Import the modules as well and the pins variables
- Set up the pins and initial point as well
- This time we'll use a new variable called
  - Pwm_r1
  - Which is the pulse on the right motor. And we will set it with the right pin and to 500.
  - We will start at 0 each time.
  - Then we need to define the movements again.

PWM_MOTOR.PY

- We'll define stop, forward, backward, left and right again
- But this time we define the pulse of each motor.
- For example:
- Which one has dc which one doesn't? Finish the rest!

```
def forward():
    pwm_r1.ChangeDutyCycle(dc)
    pwm_r2.ChangeDutyCycle(0)
    pwm_l1.ChangeDutyCycle(dc)
    pwm_l2.ChangeDutyCycle(0)
    time.sleep(t)
    stop()
```

```
def forward():
    GPIO.output(Motor_R1_Pin, True)
    GPIO.output(Motor_R2_Pin, False)
    GPIO.output(Motor_L1_Pin, True)
    GPIO.output(Motor_L2_Pin, False)
    time.sleep(t)
    stop()
```

FOLLOWER_CAR.PY

- We'll need to import new modules into this file
- The first two will be the files we just made
  - Import dc_motor as motor
  - Import pwm_motor as motor
  - Import cv2 (this is opencv)
- Now, we'll need to set the color
- We have two kinds of color: upper and lower

UPPER AND LOWER COLOR

- Upper and lower means the upper and lower boundaries
- So you can set what color you want
- For me, I used green
- You can use red or blue
  - red: lower(160,20,70)
  - Upper: (190,255,255)
  - Blue lower(101,50,38)
  - Blue upper(110,255,255)

FOLLOWER_CAR.PY

- Next we set frame width and height:
  - Frame_Width=320
  - Frame_Height=240

```
import cv2
#import pwm_motor as motor
import dc_motor as motor

Color_Lower = (36, 130,46)
Color_Upper = (113, 255, 255)
Frame_Width  = 320
Frame_Height = 240
```

FOLLOWER_CAR.PY

- Next, we set video capture with cv2
    - This allows us to take videos using our camera
    - Now we set frame width and height
  - camera = cv2.VideoCapture(0)
  - camera.set(cv2.CAP_PROP_FRAME_WIDTH, Frame_Width)
  - camera.set(cv2.CAP_PROP_FRAME_HEIGHT, Frame_Height)

FOLLOWER_CAR.PY

- We're going to use the try loop
- Try:
  - While True:
    - (__, frame) = camera.read()
    - Frame = cv2.GaussianBlur(frame, (11,11),0)
    - hsv = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)
    - mask = cv2.inRange(hsv, Color_Lower, Color_Upper)
    - center = None
    - if len(contours) > 0:
      - c = max(contours, key=cv2.contourArea)
      - ((x, y), radius) = cv2.minEnclosingCircle(c)
      - M = cv2.moments(c)

FOLLOWER_CAR.PY

- Next we have to find the center
  - center = (int(M["m10"] / M["m00"]), int(M["m01"] / M["m00"]))
- And we need to process every frame:
  - cv2.circle(frame, (int(x), int(y)), int(radius),(0, 255, 255), 2)
  - cv2.circle(frame, center, 5, (0, 0, 255), -1)
- Forward and backward rule:
  - if radius < 90:
    - motor.forward()
  - elif radius > 100:
    - motor.backward()
  - else:
    - motor.stop()

FOLLOWER_CAR.PY

- turn right and turn left
  - if center[0] > Frame_Width/2 + 10:
    - motor.turnRight()
  - elif center[0] < Frame_Width/2 - 10:
    - motor.turnLeft()
  - else:
    - motor.stop()
  - Except:
    - pass

FOLLOWER_CAR.PY

- cv2.imshow("Frame", frame)
- key = cv2.waitKey(1) & 0xFF
- if key == ord("q"):
  - Break
- We can mark it up if we want to make our car faster
- So this is optional

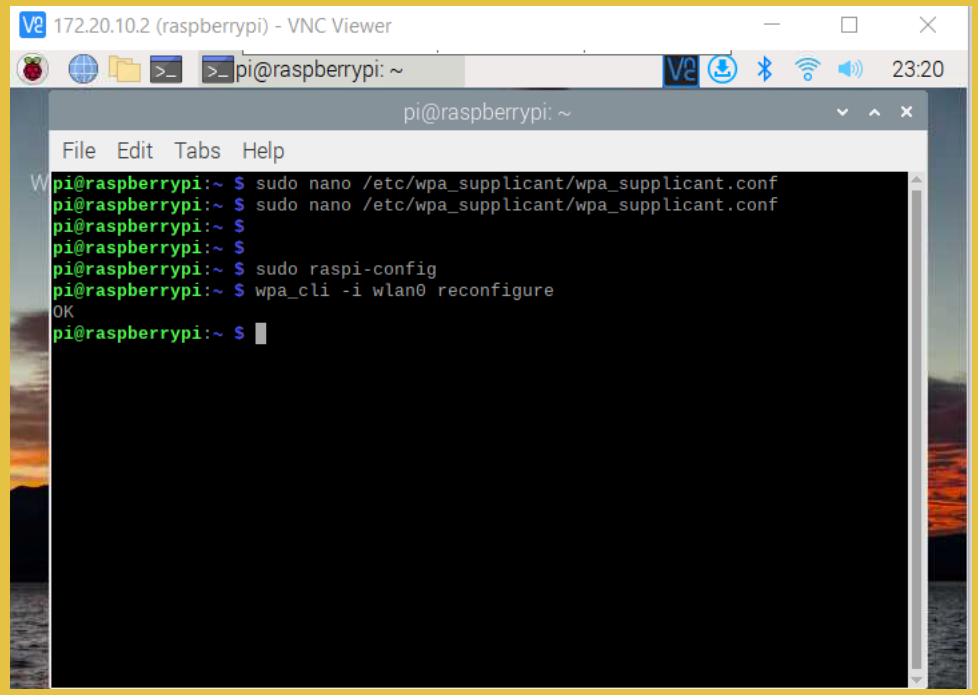
FOLLOWER_CAR.PY

- Now we've reached the end so we need to clean up, close the window and close the camera as well
  - finally:
    - motor.cleanup()
    - camera.release()
    - cv2.destroyAllWindows()

- Make sure the check for typos cause if you type it wrong it'll won't work.

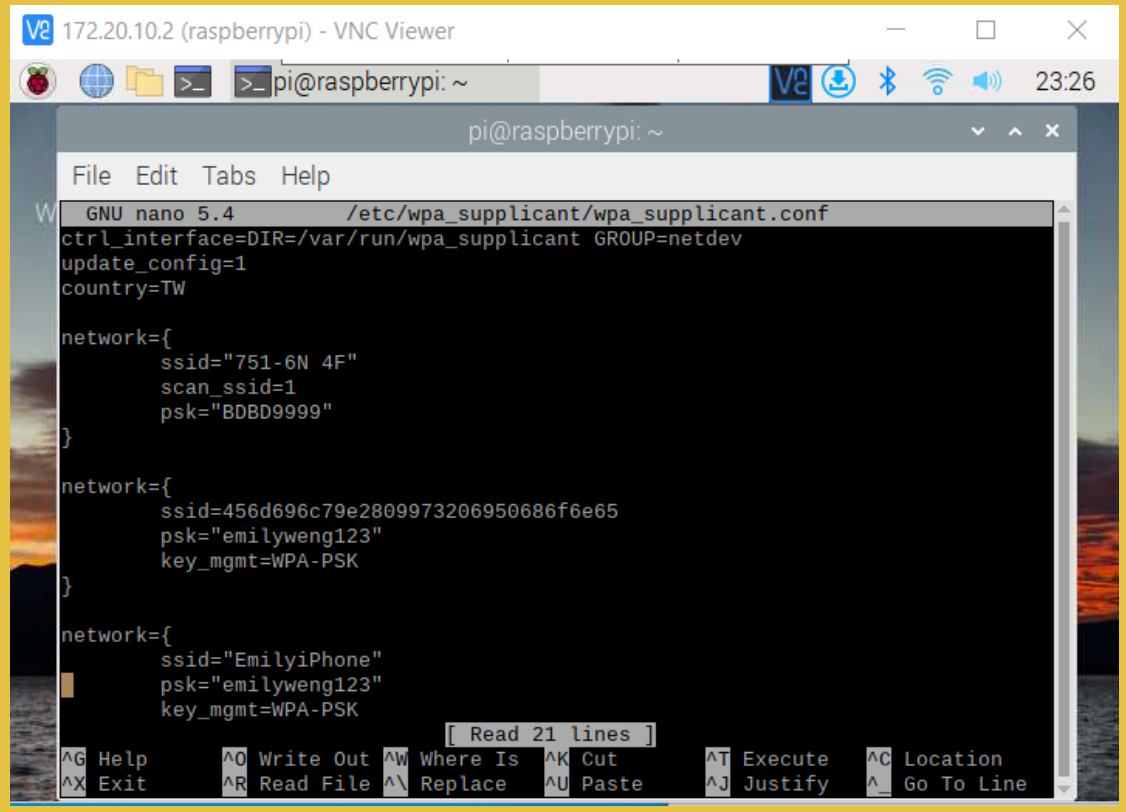
HOW TO SWITCH WIFI

- First, make sure your wifi's(phone's) name doesn't have any space or weird signs
- Second: check your hotspot opened
- Use the code:
- Sudo nano /etc/wpa_supplicant/wpa_supplicant.conf

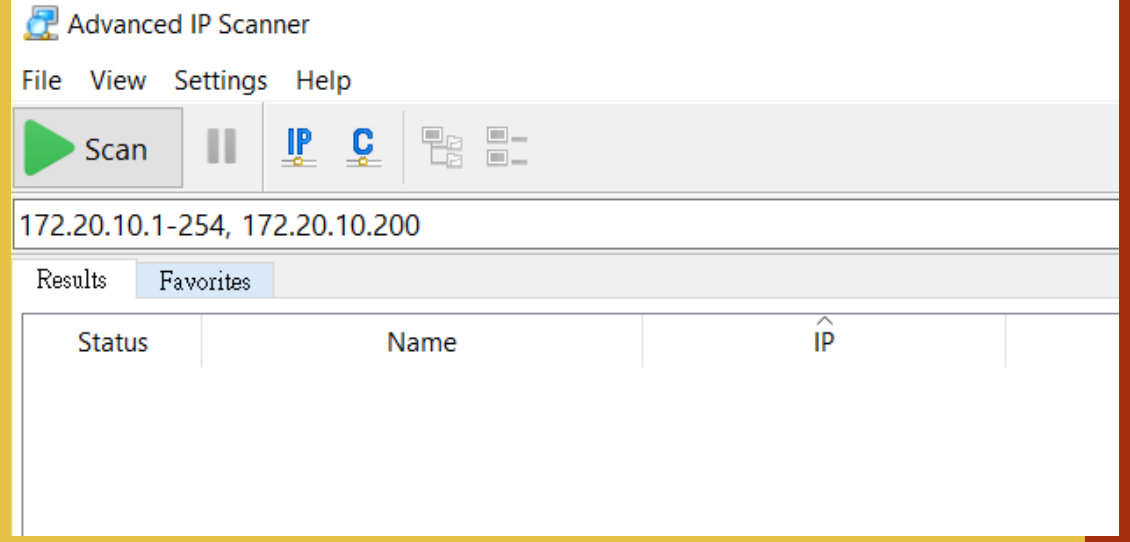


HOW TO SWITCH WIFI

- To open your wpa_supplicant.conf file
- In your wpa_supplicant file, add another network block and add your phone's name and your hotspot's password
- Save the file and exit
- Next, type the code:
  - Wpa_cli –i wlan0 reconfigure
- To take in the new network configuration
- After configuration, you should see an okay



SWITCH WIFI



- Then you can switch to your phone's hotspot
- Make sure your computer is connected to your phone's network
- Find your new ip address using properties from the network section
- Copy your new ip address to advanced ip scanner Should look something like this and scan it
- You should find your raspberry pi's new ip address and enter it into your vnc server
- VNC server should pop out the same system with the new ip address

- sudo apt-get install rpi.gpio