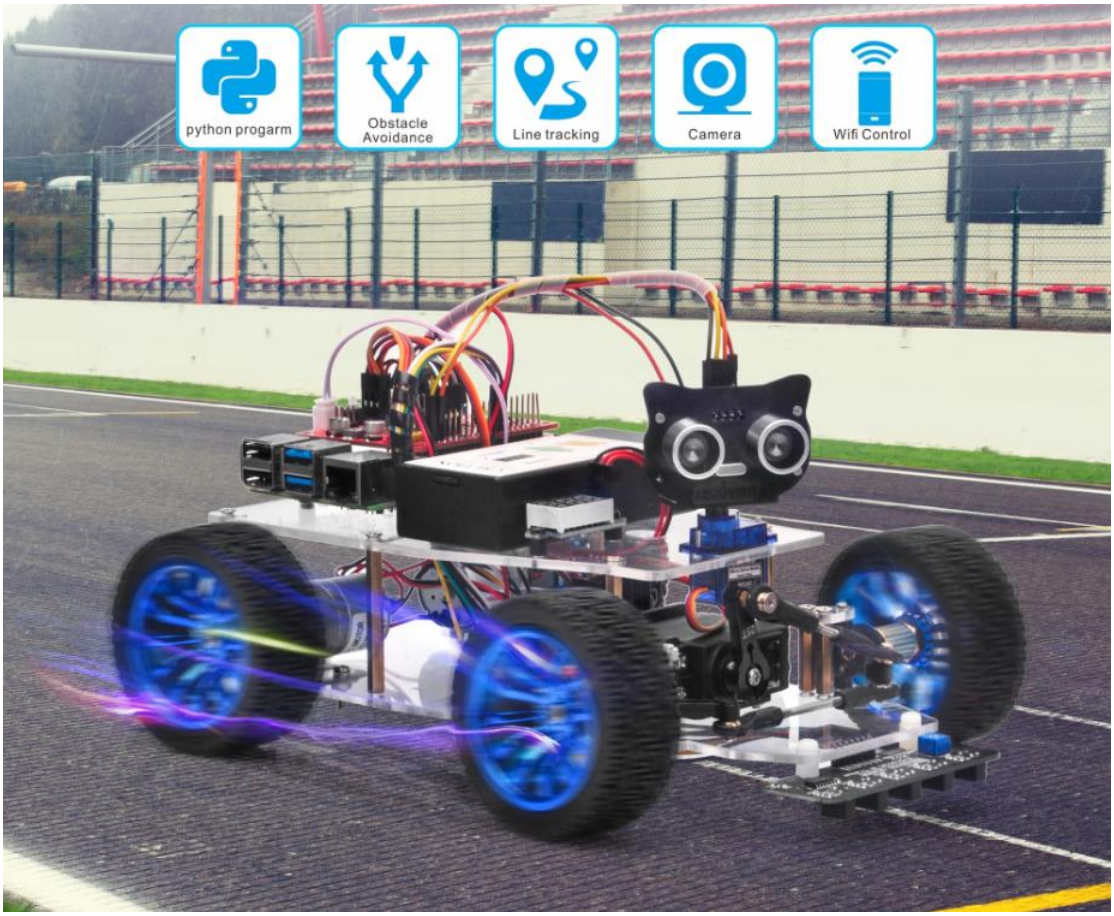
OSOYOO Servo Steer Smart Car for Raspberry Pi



Online Tutorial Index: https://osoyoo.com/?p=40236

If you have any problem, please feel free to contact us, our email address is support@vership.com

Table of Contents

Preface

Safety Instruction

- Do not plug or unplug any wire or module when power is on,
- Do not even touch the board when power is on or your hand has static charge (you can move static charge by touching metal tap).
- Reversing +/- when connecting modules with board, or incorrectly connecting can destroy your electronics and cause fire hazard.
- Please make sure polar direction is correct when you install batteries in your battery box, otherwise it can destroy your device and cause fire hazard.
- Please don't use Carbon zinc batteries as power, as output current of this kind of battery is too low and it is unable to load the robot car
- Do not leave batteries in battery box if you don't use it for long time due to the risk of fire and malfunction.

About OSOYOO

OSOYOO brand owned by Pinetree Electronics Ltd, the only Canadian Owned Science Fair Supplier, Circuit Builder, IOT, Autonomous Smart Machine, Home Electronics Workshop and Accessory Company.

Pinetree Electronics Ltd established since 2009 in Vancouver, Canada. We are Engineers and Programmers Ourselves, So We Know How Important It Is to CREATE Your Dreams! Not All Robot Kits Are Created Equal: Pay For True Quality & You Will Not Be Disappointed.

For more information and tutorial of OSOYOO products, please visit: www.osoyoo.com

About This Kit

OSOYOO Servo Steer Smart Car For Raspberry Pi is a perfect combination of challenge and excitement, learning and fun. The kit comes with step by step ONLINE tutorial with text, picture and video. The kit is great for any skill level — whether you're a pro, enthusiast, or a beginner.

Online Tutorial Index:

General Introduction: https://osoyoo.com/?p=40236

OSOYOO robot car introduction video: https://osoyoo.com/2021004700.html

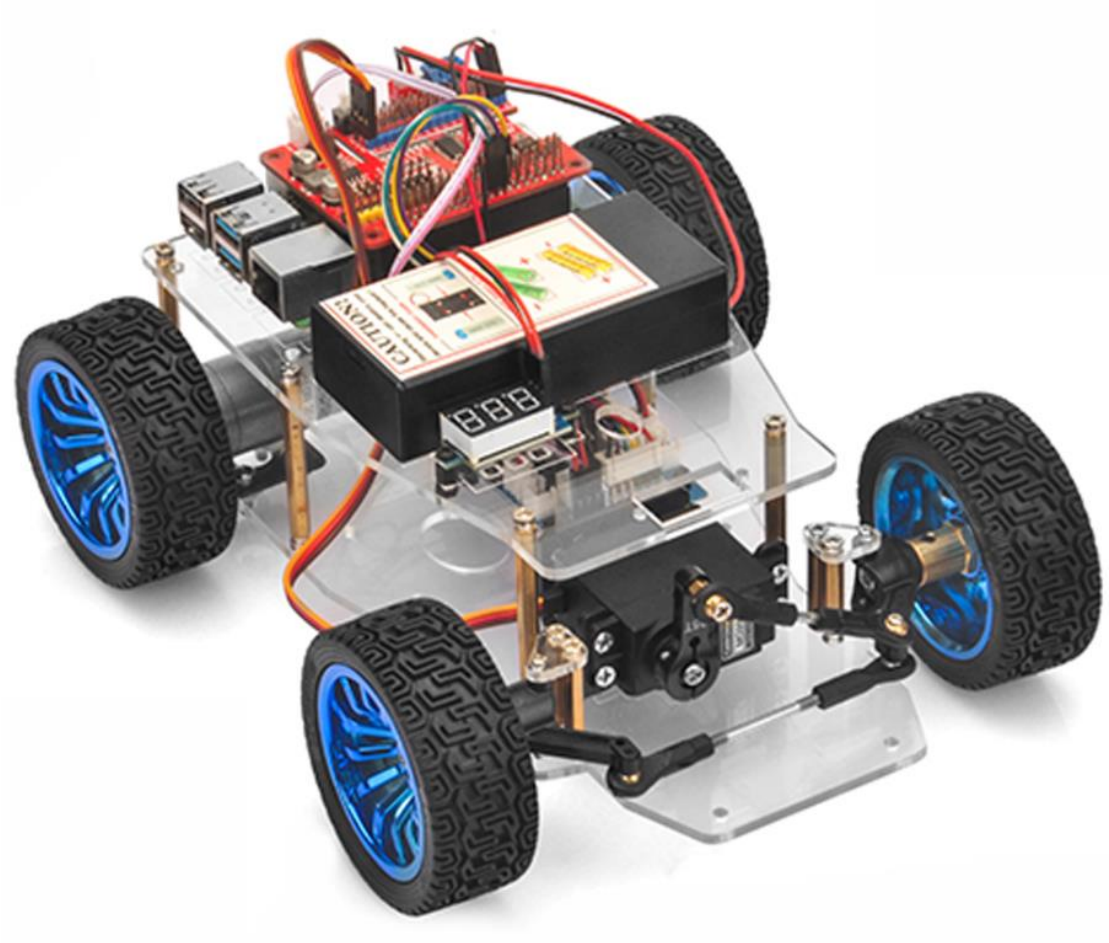
OSOYOO robot car tutorial PDF download link: https://osoyoo.com/manual/2021004700.pdf

Customer Service and Tech Support

You have following two options to ask for help:

1) Leave comments on our tutorial https://osoyoo.com/?p=40236
2) Send email to support@osoyoo.info. We guarantee that all inquiries will be replied in 24 hours.

OSOYOO V2.1 Robot car kit Lesson 0: Introduction



There are many entry level Robot Car Kits in the market, most of them are controlled by Arduino Boards. You can check our tutorial blog for such Arduino Robot kit in https://osoyoo.com/2017/08/06/osoyoo-robot-car-diy-introduction .

The advantage of Arduino Robot Car kit is that Arduino has no Operation System and programming is simple and easily. For some basic robot application which needs only simply logic to handle sensor data and control actuators, Arduino-controlled robot car is a good choice.

However, for some more complex robot applications which need more complex functions such as computer vision (CV), Internet of Things (IoT), web server control etc, Arduino board's ability is too weak to reach the target.

In order to help intermediate students to complete some complex Robotic project. We developed a more advanced Raspberry Pi Robot Car learning Kit.

Why Raspberry Pi is so important to the Robot Car DIY learning kit?

Because Raspberry Pi is a real computer which has Linux OS (Raspbian) and therefore much powerful than Arduino Board which is simply a micro-controller (MCU).

With Raspbian OS and its huge open-source software community , people can make much complicated Robot projects,i.e web appliation, database, A.I, machine learning, IoT, Computer Vision etc.

Unlike Arduino board, Raspberry Pi programming environment is much more complex and flexible. It supports almost all programming language as long as the language is supported by Rasbian Open Source community. The most commonly used languages for access Raspberry Pi GPIO pins are C and Python. If you want to learn some Raspberry Pi hardware GPIO programming, you can read our tutorial in following links:

https://osoyoo.com/2017/10/09/raspberry-pi-starter-kit-v1-introduction/

Tutorial and sample projects

We have developed a step-by-step tutorial which evolves from a simple car without any control to a multi-function robotic car controlled by mobile APP. Every lesson has detailed sample code with comments, circuit graph, assembly instruction and video. Even if you have no programming experience, you can follow the step-by-step instruction and gradually become a master.
Our robotic car is 100% open source. If you are an intermediate player and have time to read our code comments, you can easily customize this robotic car to make your own project for science fair, college homework or even commercial applications.

Robot car Sample Projects:

- Lesson 1 Basic robot car assembly: URL https://osoyoo.com/?p=36370
- Lesson 2 Line Tracking: URL https://osoyoo.com/?p=36411
- Lesson 3 Obstacle Avoidance: URL https://osoyoo.com/?p=36426
- Lesson 4 robot car controlled by phone: URL https://osoyoo.com/?p=39617
- Lesson 5 Make a simple website server in Pi: URL https://osoyoo.com/?p=40284
- Lesson 6 Web-Camera Controlled: URL https://osoyoo.com/?p=36440

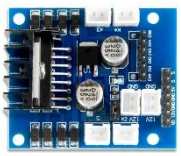
OSOYOO Servo Steer Smart Car for Raspberry Pi lesson 1: Hardware Installation

OBJECTIVE

Welcome to the first lesson of OSOYOO Servo Steer Smart Car for Raspberry Pi!
In this lesson, we will install the framework of the OSOYOO Servo Steer Smart Car for Raspberry Pi and simply introduce the hardware of this robot.
All lessons are based on the frame work of this lesson. Please follow this lesson carefully.

Please enter the link to watch the video: https://youtu.be/85tj4KS5_po

PARTS & DEVICES

No.	Picture	Device	Qty.	Link
1		Raspberry pi board 2/3/4 (**not in package**)	1	Click here to buy
2		OSOYOO PWM HAT v1.0	1	Click here to buy
3		OSOYOO model X motor driver module	1	Click here to buy
4		Servo motor	1	Click here to buy

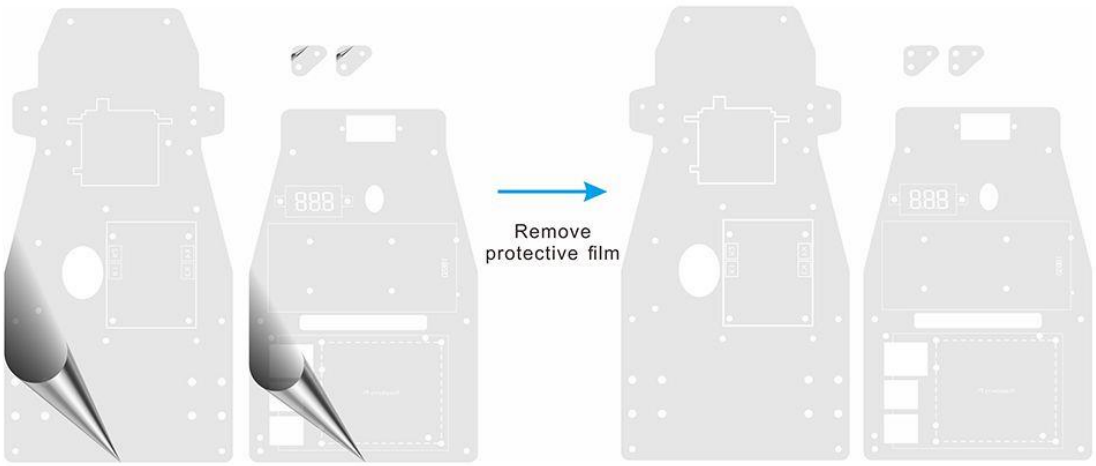
5		Servo motor holder	1	Click here to buy
6		Servo horn	1	Click here to buy
7		Voltage meter	1	Click here to buy
8		Motor with cable	2	Click here to buy
9		Motor holder	2	Click here to buy
10		Wheel	4	Click here to buy
11		Motor Flexible Coupler	1	Click here to buy
12		RC Steering Cup	2	Click here to buy

13		Connecting rod	2	Click here to buy
14		Rod radial end bearing	2	Click here to buy
15		Chassis	1	Click here to buy
16		3pin female to 3pin female jumper wire	1	Click here to buy
17		6Pin female to female jumper wire	1	Click here to buy
18		2Pin 20cm XH2.54 female jumper wire	1	Click here to buy
19		18650 battery box	1	Click here to buy
20		Battery charger for 18650 battery (Optional)	1	Click here to buy

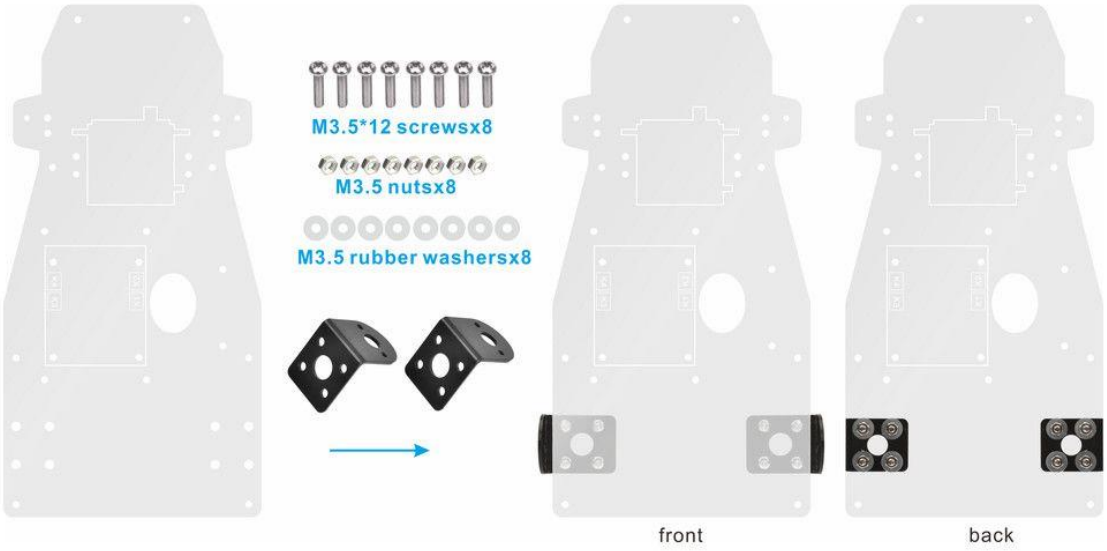
21		1Pair 18650 batteries (Optional)	1	Click here to buy
22		Philips screwdriver	1	Click here to buy
23		L Type hex wrench	1	Click here to buy

HARDWARE INSTALLATION

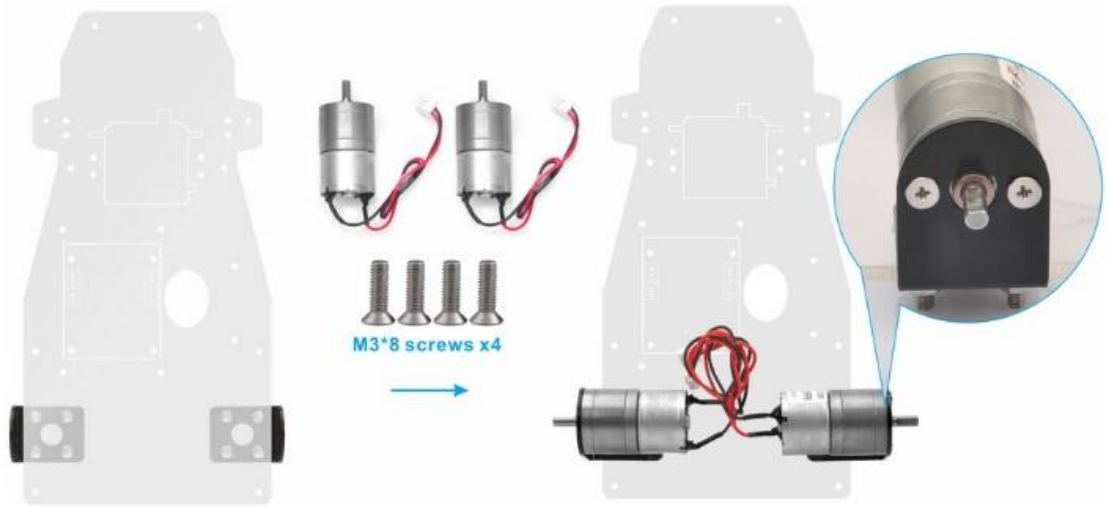
1. Remove the protective film from the chassis.



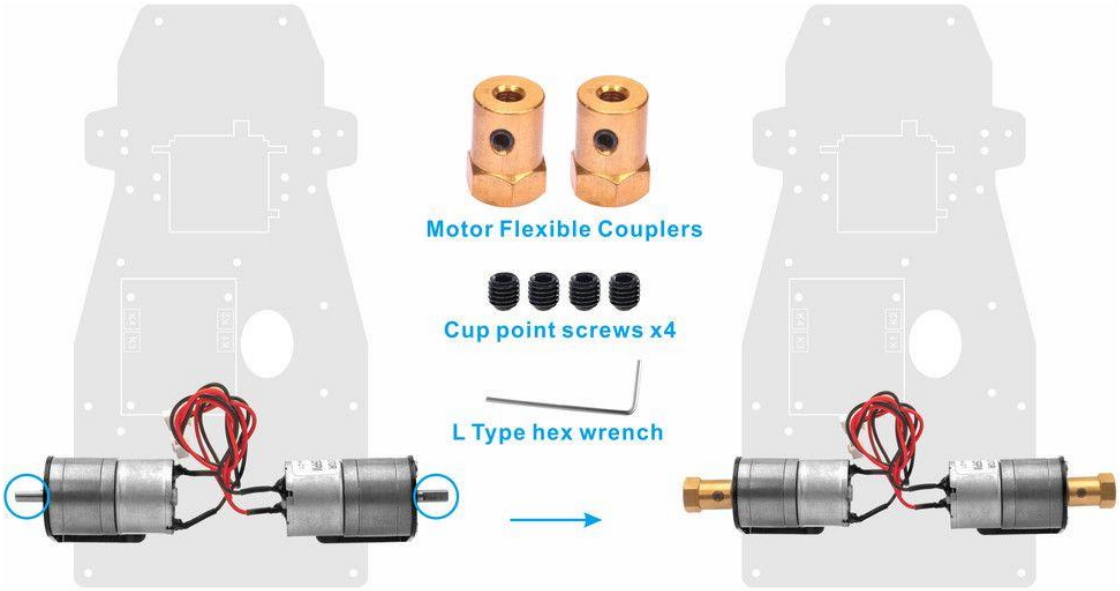
2. Fix motor holders on the lower chassis with 8pcs M3.5*12 screws, M3.5 nuts and M3.5 rubber washers. From top to bottom, use M3.5*12 screws cross lower chassis, Motor holder, 3.5 rubber washer and then 3.5 nut as follow:



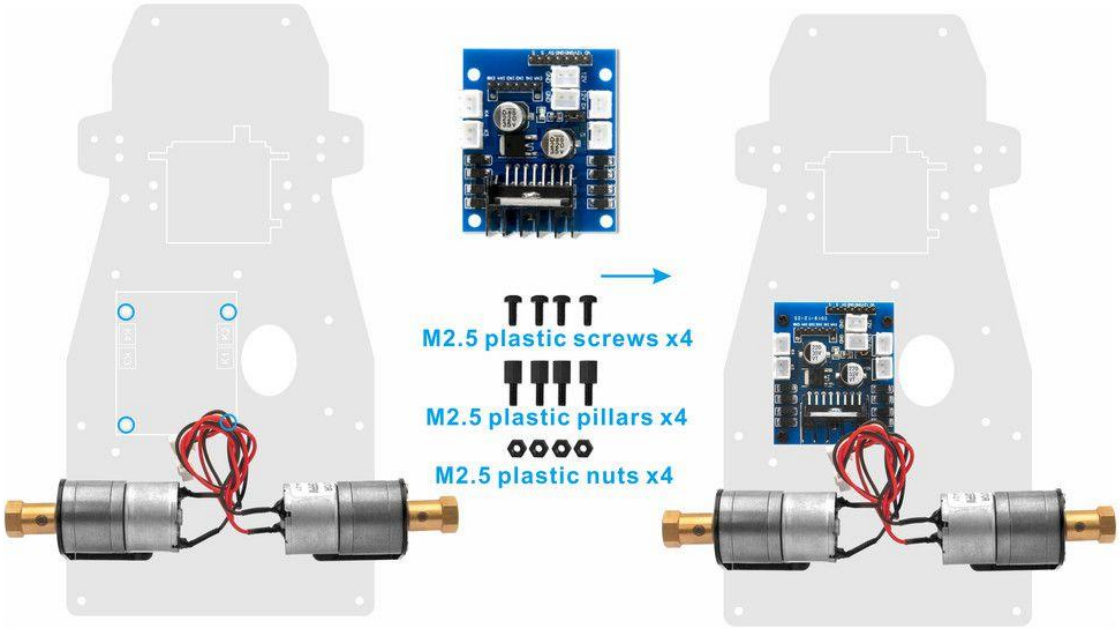
3. Cross motor holder, use M3*8 screws to fix motor on motor holder as follow:



4. Install motor flexible couplers on motors and use cup point screws to fix motor flexible couplers on motor.
The shaft from the motor has a flat area on it. Make sure that cup point screws are positioned on this flat, and tightened both screws on the shaft.

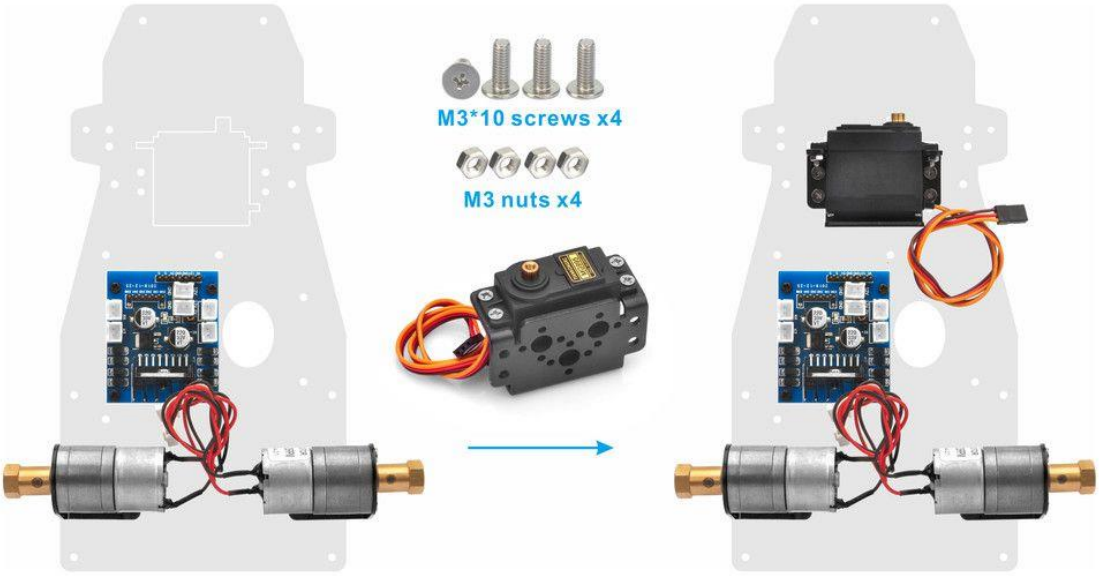


5. Install OSOYOO MODEL X motor driver module to lower car chassis with 4pcs M2.5 plastic screws, plastic pillars and plastic nuts. **(Please make sure you install the OSOYOO MODEL X motor driver module in correct direction.)**

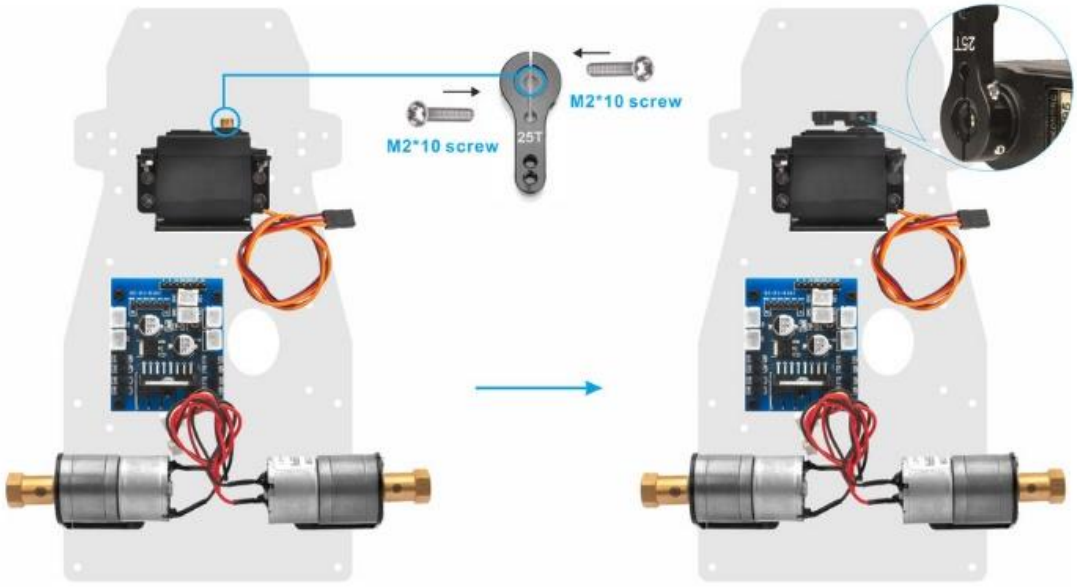


6. Fix servo motor on servo motor holder with M3*10 screws and nuts as follow:

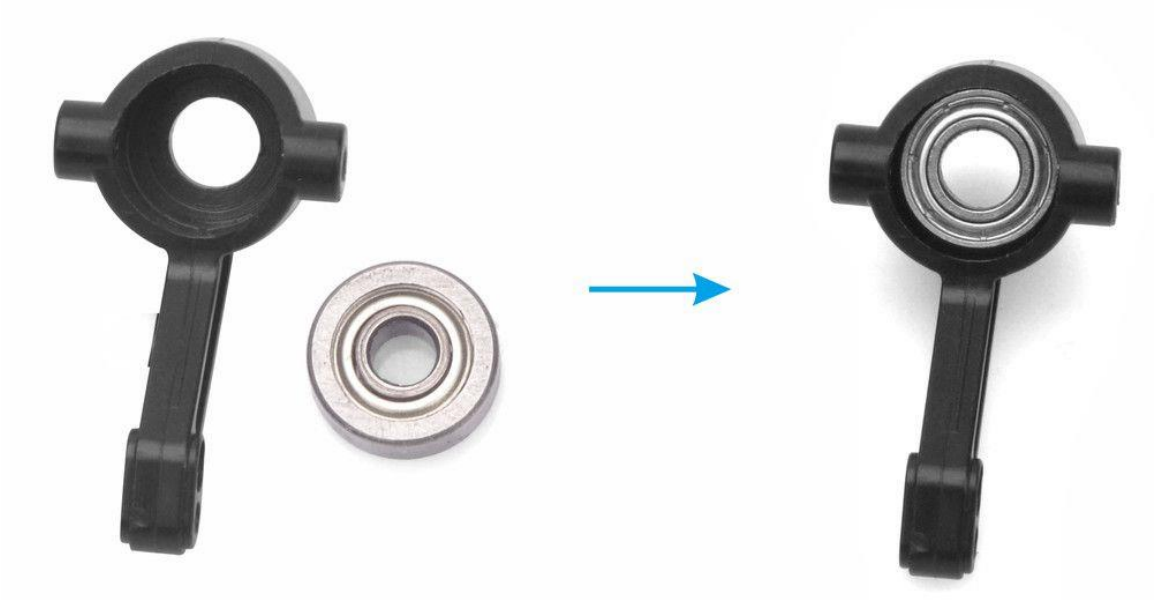
7. Use M3*10 screws cross servo motor holder with servo motor, lower chassis and nut to fix servo motor on lower chassis as follow:



8. Install servo horn on servo motor with 2pcs M2*10 screws as follow:



9. Push 4x12x4 roller bearing into front of RC steering cup tightly as follow:

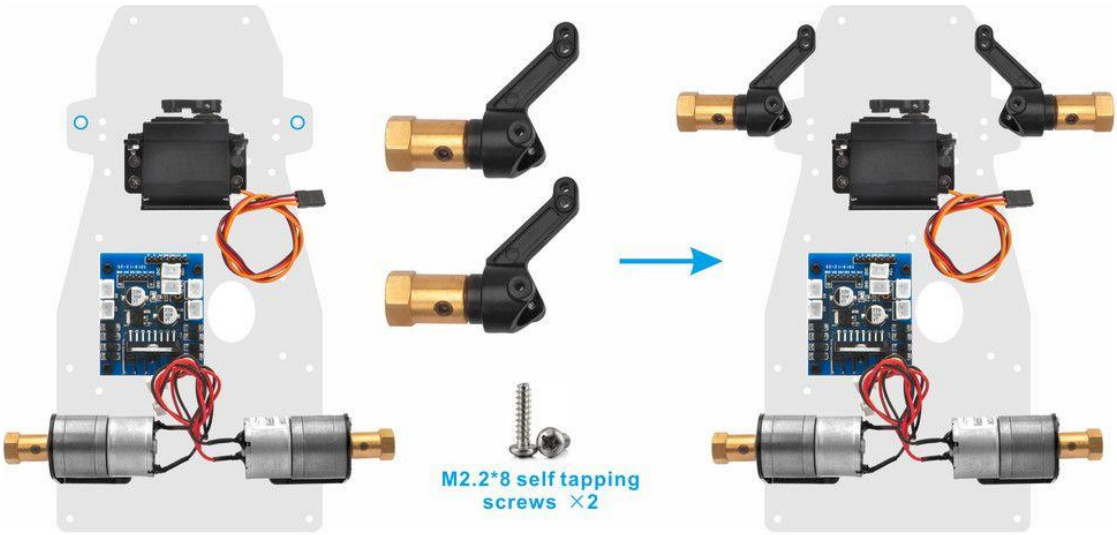


10. Push 4x8x4 roller bearing into back of RC steering cup tightly as follow:

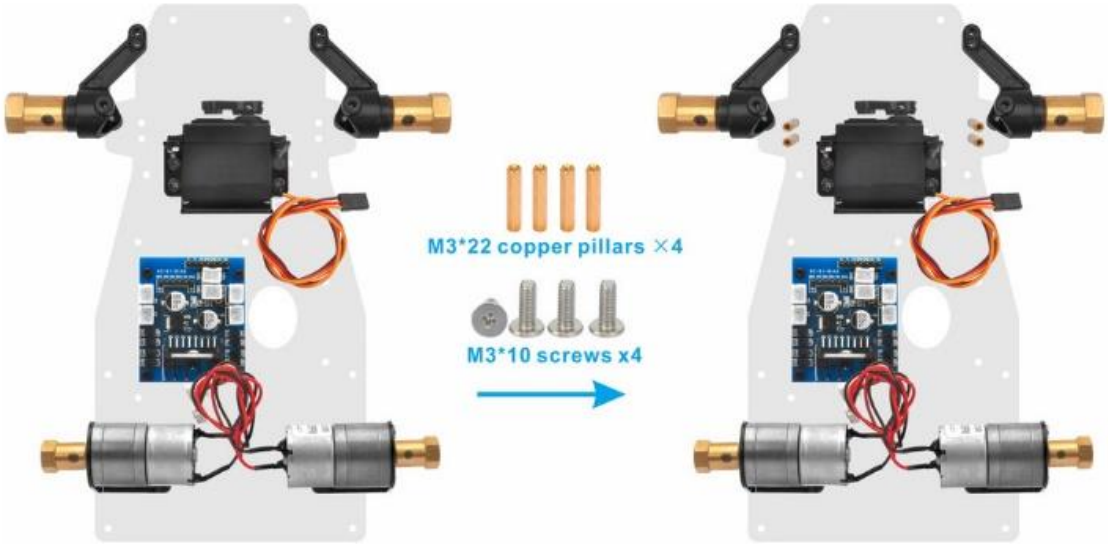
11. Use M4*20 screw cross RC steering cup as follow:

12. Fix Motor Flexible Couplers on M4*20 screw with Cup point screws as follow:

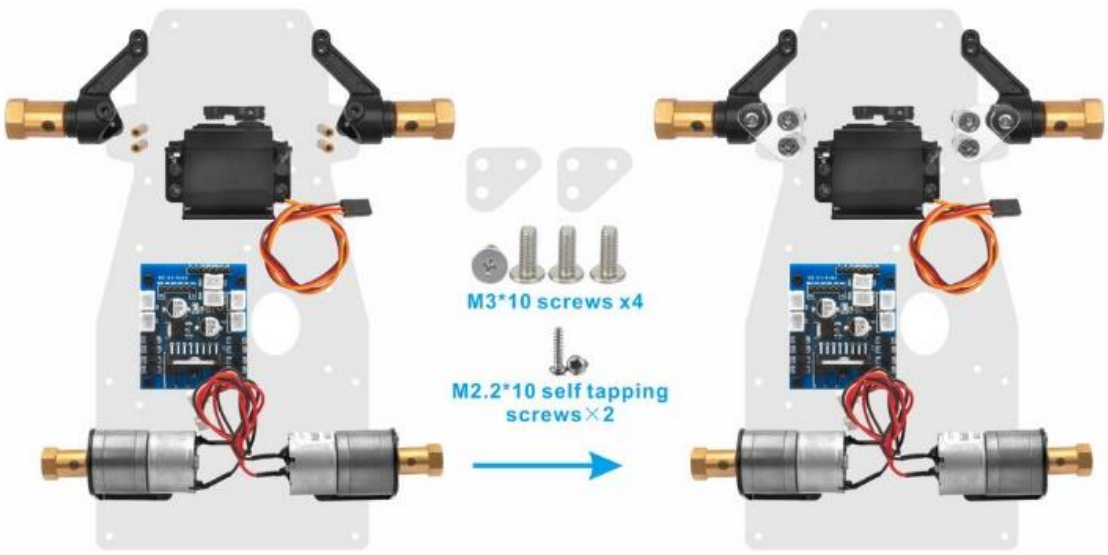
13. Fix RC steering cups on lower car chassis with M2.2*8 self-tapping screws as follow:



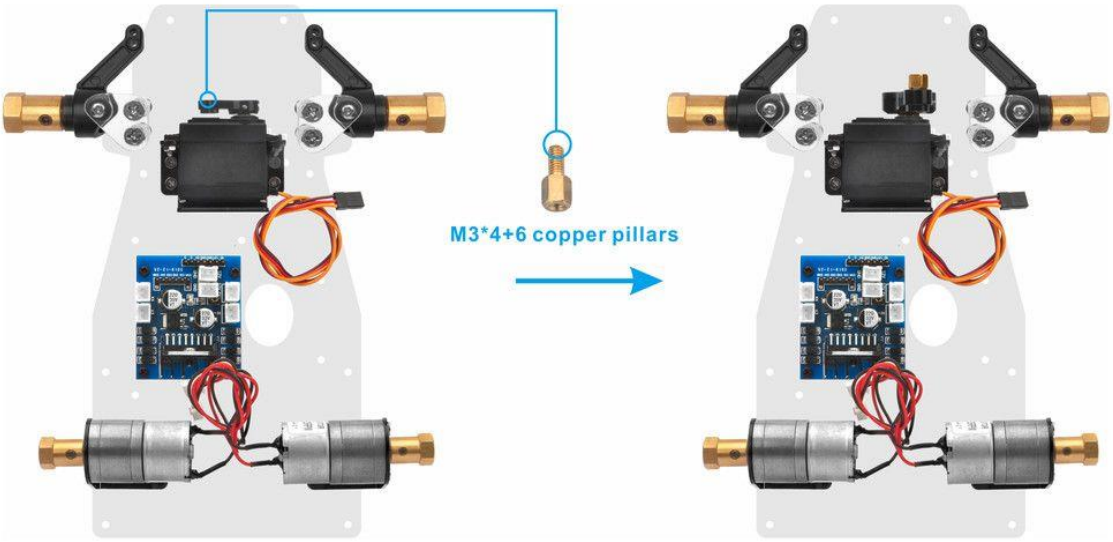
14. Install M3*22 copper pillars beside of RC steering cups with M3*10 screws as follow:



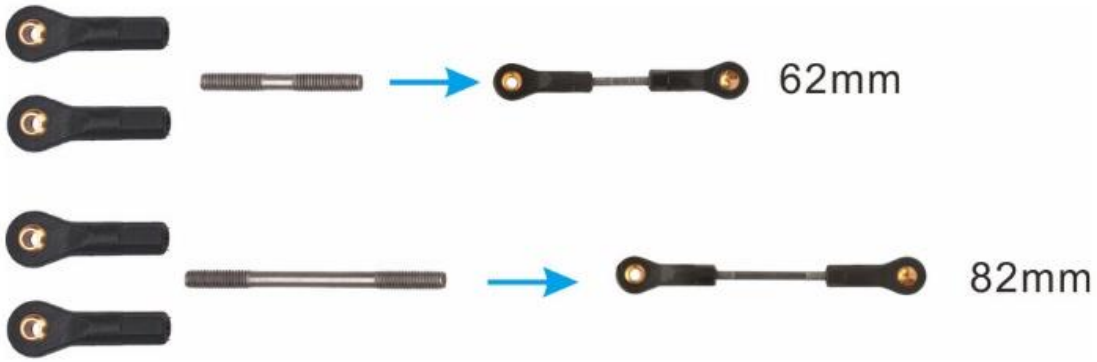
15. Install triangular Acrylic sheets on the M3*22 copper pillars and fix these with M2.2*8 self-tapping screws and M3*10 screws as follow:



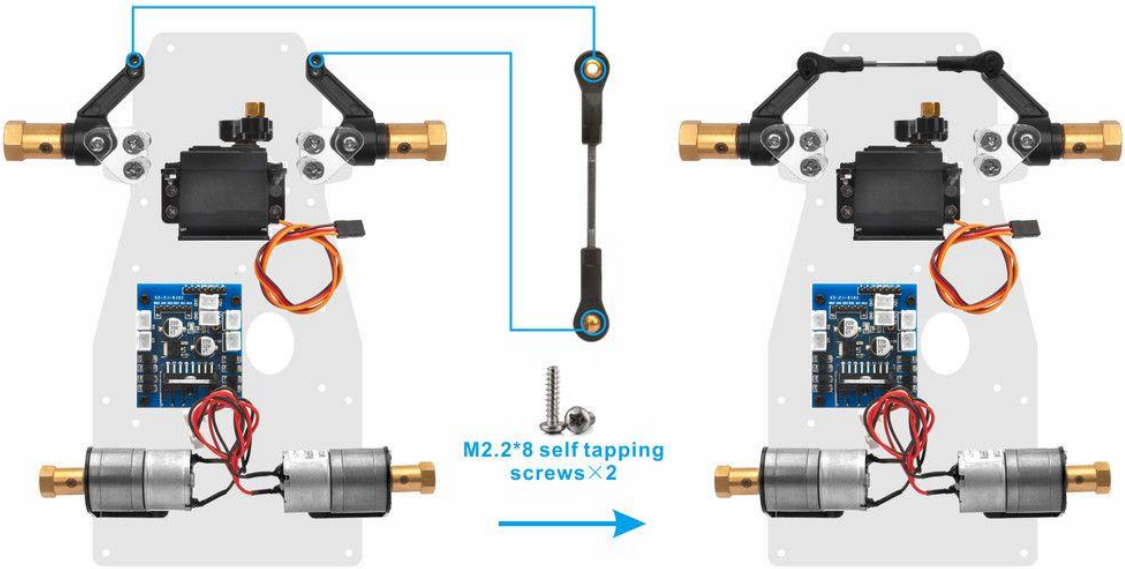
16. Install M3*4+6 copper pillar on servo horn as follow:



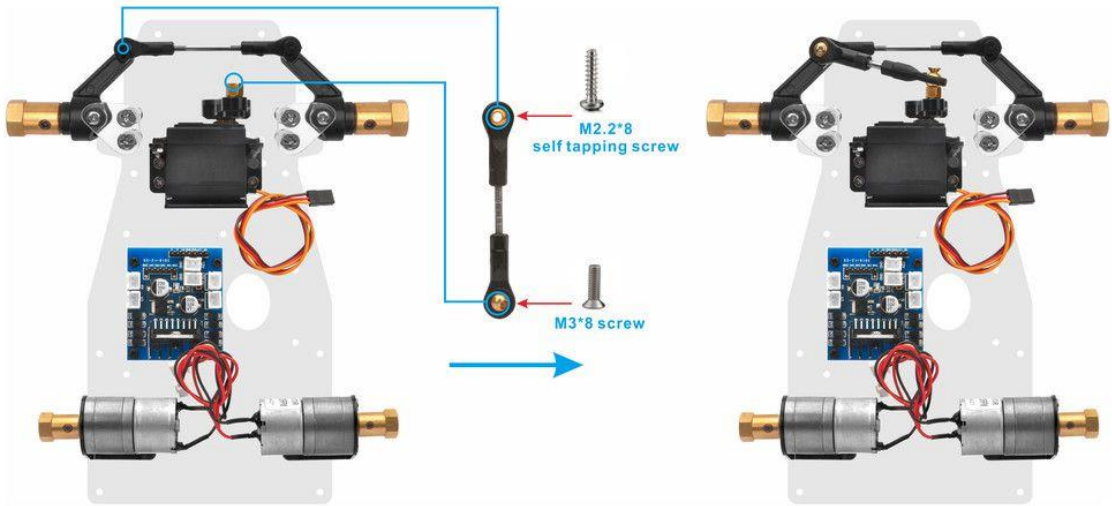
17. Fix 2 connecting rods with Rod radial end bearings, and make sure the lengths of these are about 62mm and 82mm



18. Install two ends of 82mm connecting rod under RC steering cups with 2pcs M2.2*8 self-tapping screws as follow (Note: When installing 82mm connecting rod, please keep the heads of two RC steering cups parallel, or you need to adjust the length of this connecting rod) :

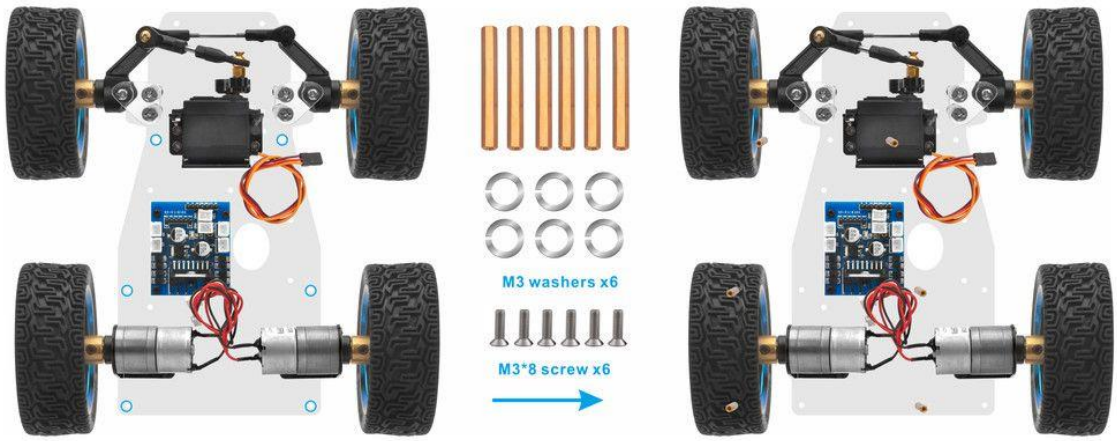


19. Install one end of 62mm connecting rod on left RC steering cup with M2.2*8 self-tapping screw and the other end on M3*4+6 copper pillar with M3*8 screw as follow (Note: When installing 62mm connecting rod, please keep the servo horn perpendicular to the micro servo, or you need to adjust the length of this connecting rod) :



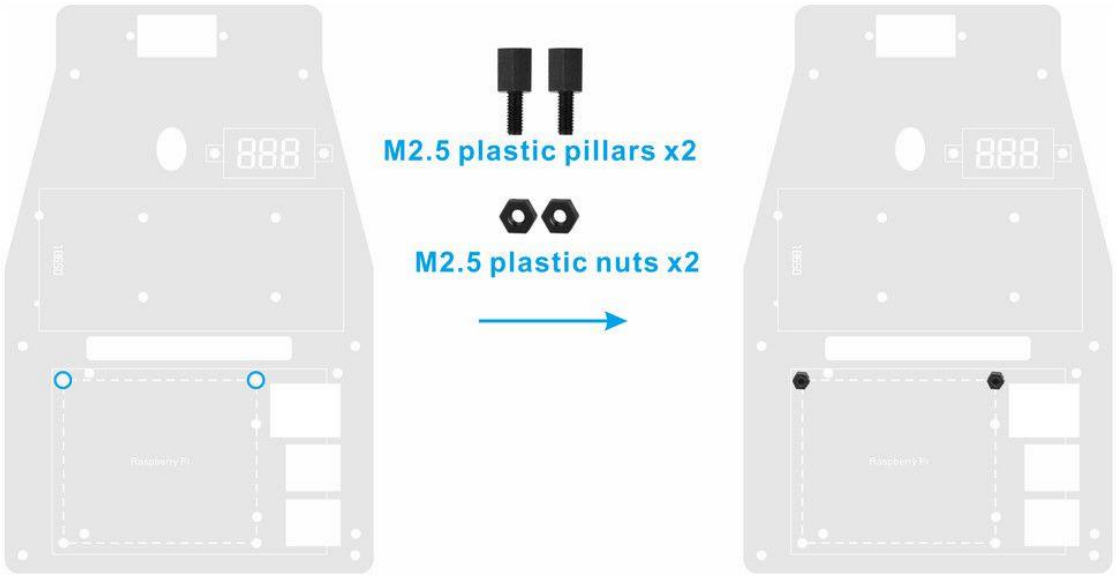
20. Install 4 wheels on motor flexible couplers with M4*6 screws as follow:

21. From bottom to top, use M3*8 cross M3 wash, lower chassis and M3*45 copper pillar to fix M3*45 copper pillar on low chassis as follow:

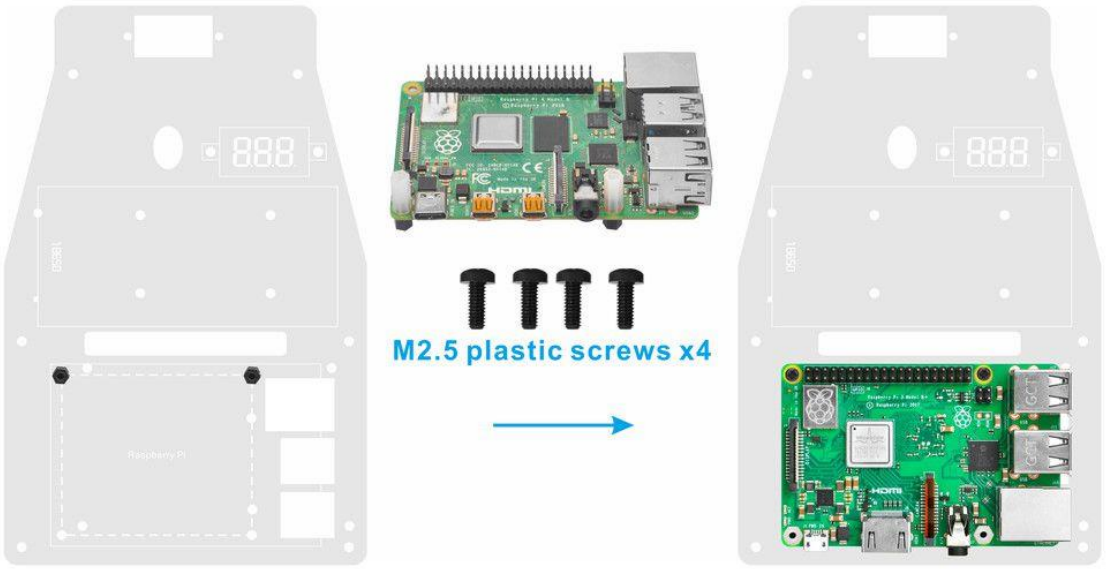


22. Use M2.5 plastic pillar cross Raspberry Pi and M2.5*13 plastic pillar from bottom to top and fix 2pcs M2.5*13 plastic pillars on Raspberry Pi as follow:

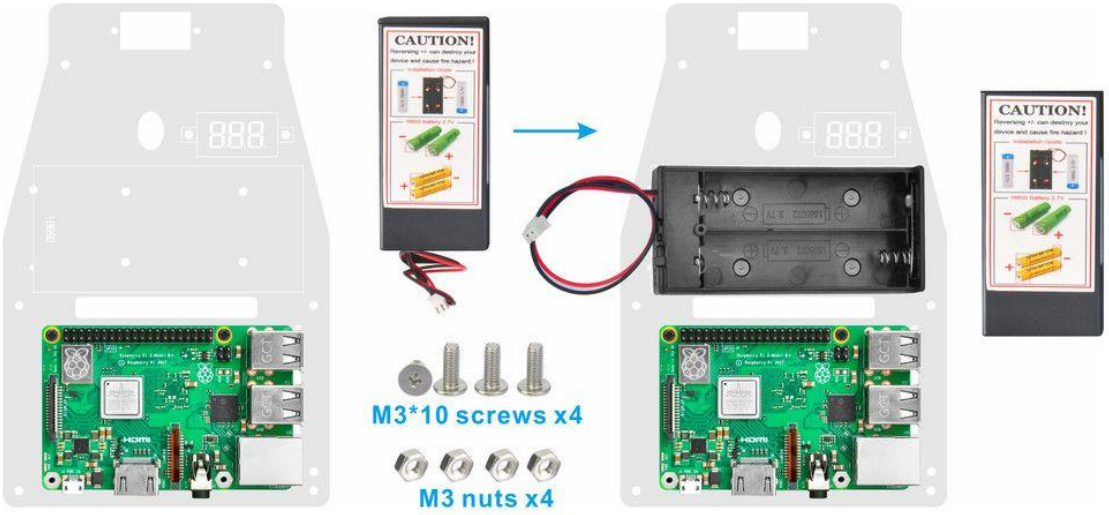
23. Fix 2pcs M2.5 plastic pillars on upper chassis as follow:



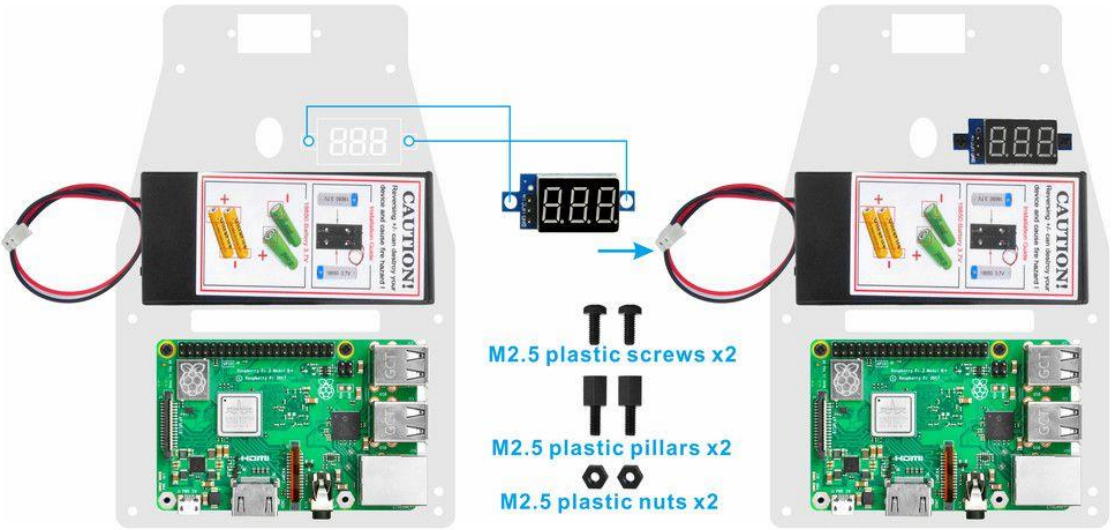
24. Install 2pcs M2.5 plastic screws under the chassis and 2pcs M2.5 plastic screws on Raspberry Pi



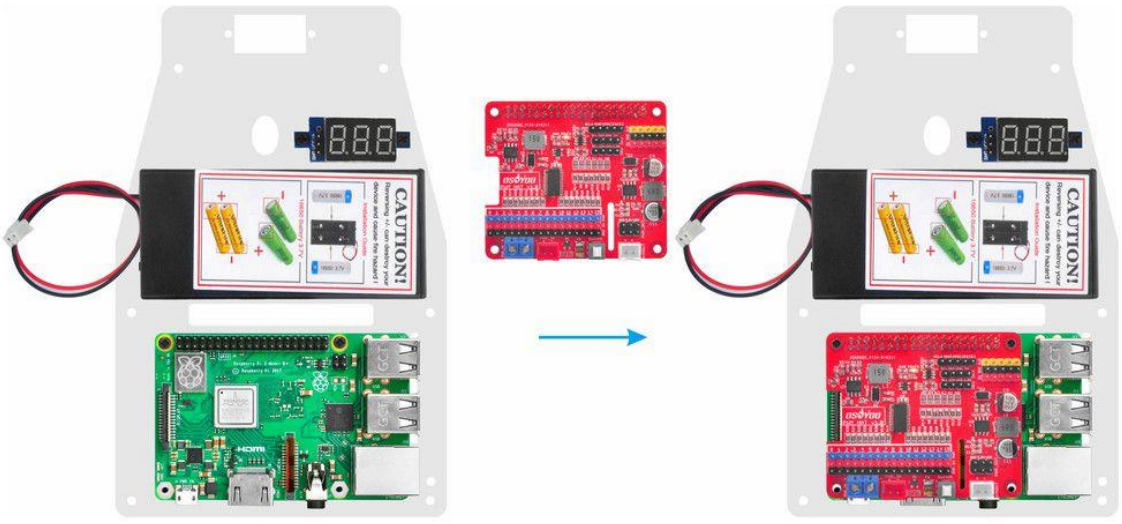
25. Fix 18650 battery box on upper chassis with M3*10 screws and M3 nuts



26. Install voltage meter on low car chassis with 2pcs M2.5 plastic screws, plastic pillars and plastic nuts



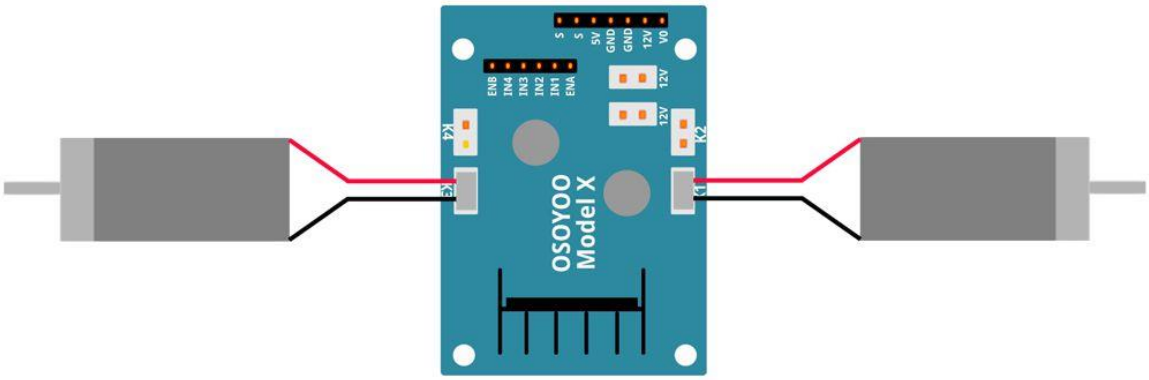
27. Insert OSOYOO PWM Hat V1.0 on Raspberry Pi as follow:



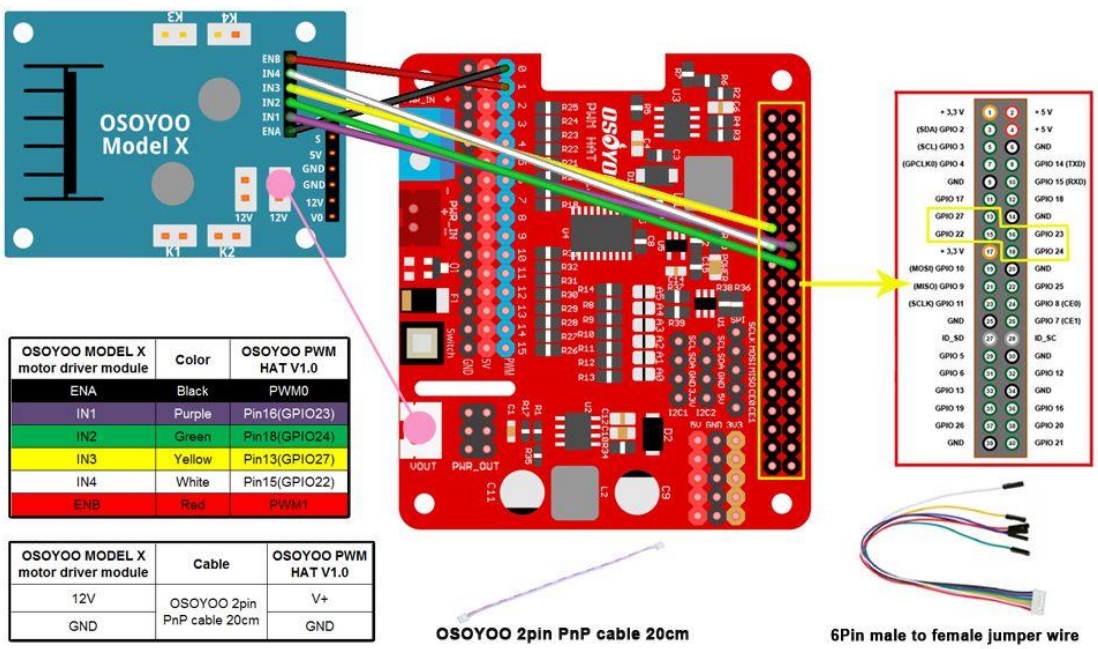
CIRCUIT CONNECTION

Note: before fix upper chassis on lower chassis, please connect the parts. To learn more about the GPIO pins of Raspberry Pi, please visit: https://osoyoo.com/2017/06/26/introduction-of-raspberry-pi-gpio/

1) Connect 2 motors to OSOYOO MODEL X motor driver module K1 and K3 sockets as following graph:



2) Connect OSOYOO MODEL X motor driver module to OSOYOO PWM Hat V1.0 with 6Pin female to female jumper wire as following graph

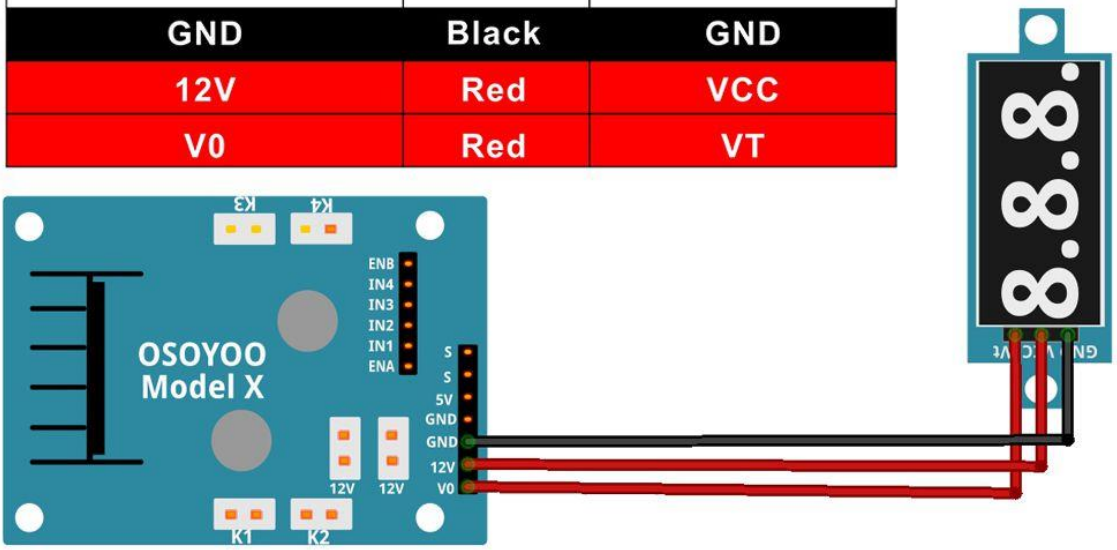


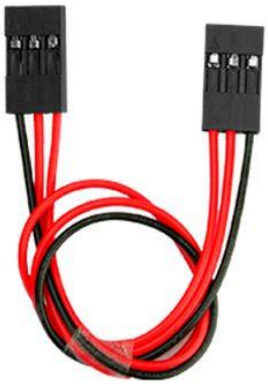
OSOYOO MODEL X motor driver module	Color	OSOYOO PWM HAT V1.0
ENA	Black	PWM0
IN1	Purple	Pin16(GPIO23)
IN2	Green	Pin18(GPIO24)
IN3	Yellow	Pin13(GPIO27)
IN4	White	Pin15(GPIO22)
ENB	Red	PWM1

OSOYOO MODEL X motor driver module	Cable	OSOYOO PWM HAT V1.0
12V	OSOYOO 2pin PnP cable 20cm	V+
GND		GND

3) Connect Voltage Meter to OSOYOO MODEL X motor driver module with 3pin female to female jumper wires as below connection diagram

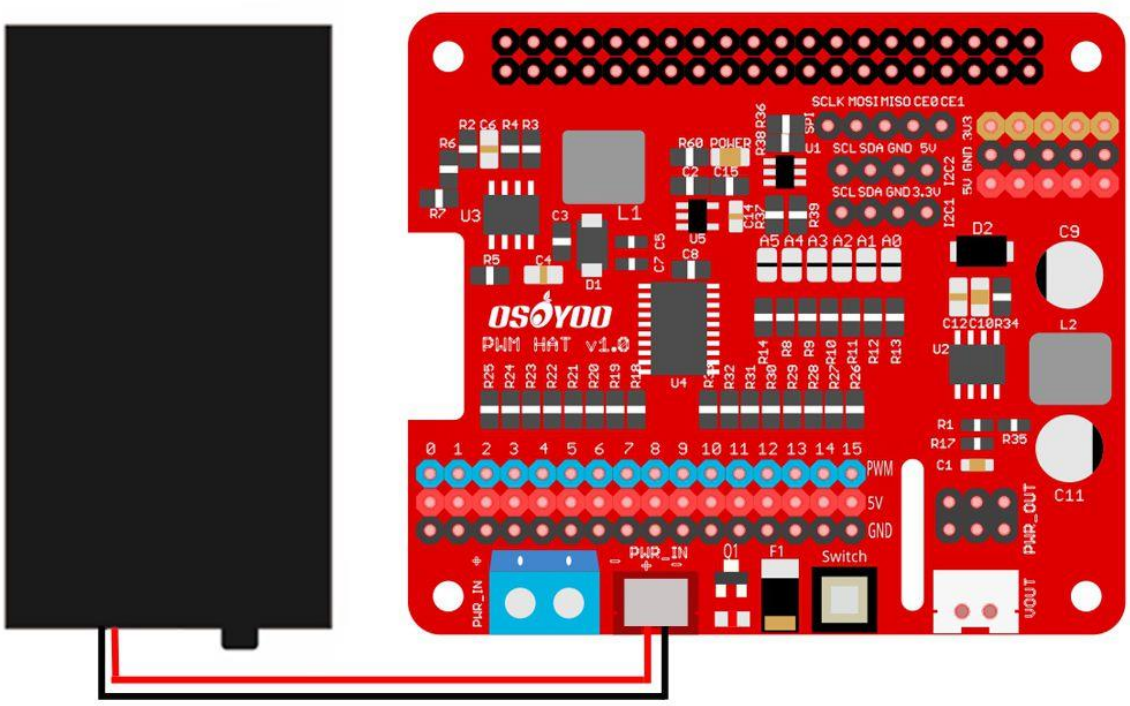
OSOYOO MODEL X motor dirver module	Colors	Voltage Mete
GND	Black	GND
12V	Red	VCC
V0	Red	VT





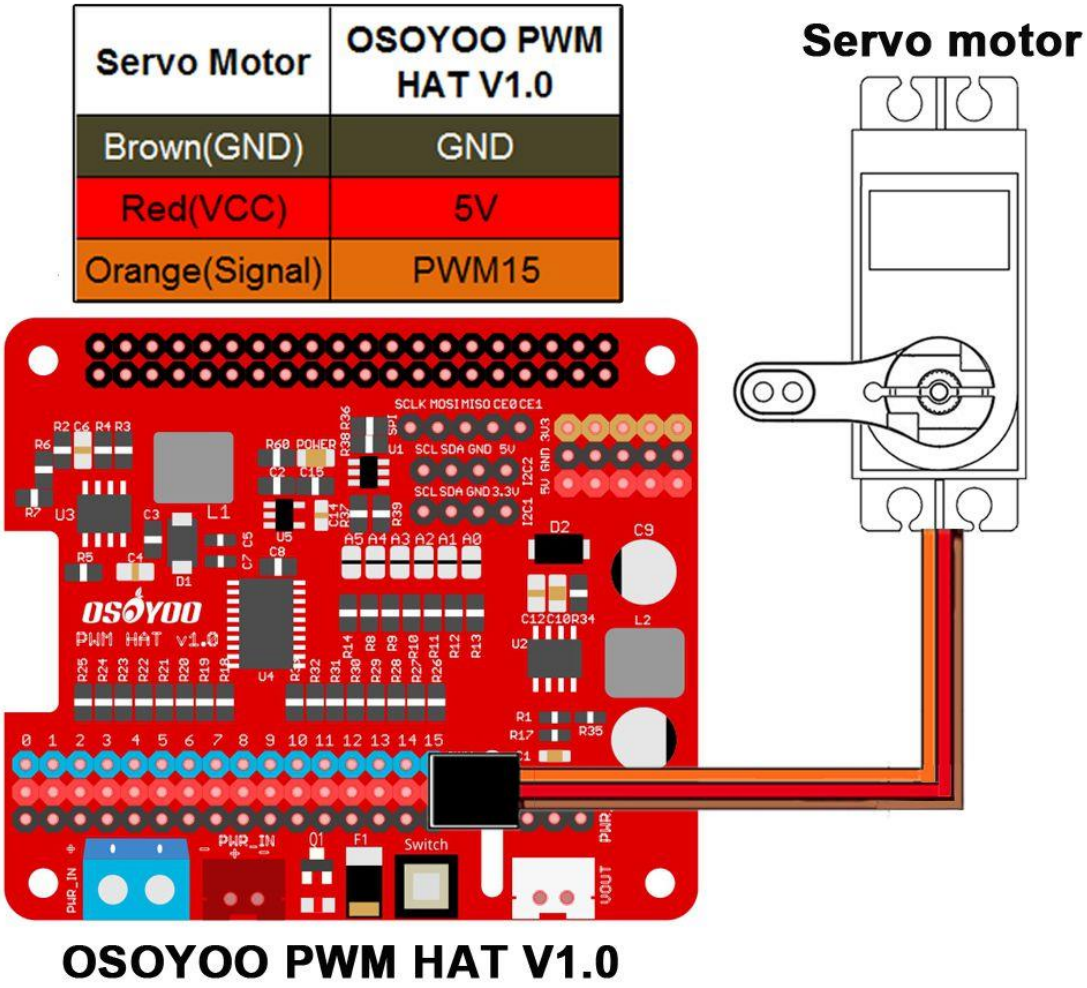
3Pin female to 3Pin female Jumper wire

4) Connect 18650 battery box to OSOYOO PWM Hat V1.0 as below connection diagram

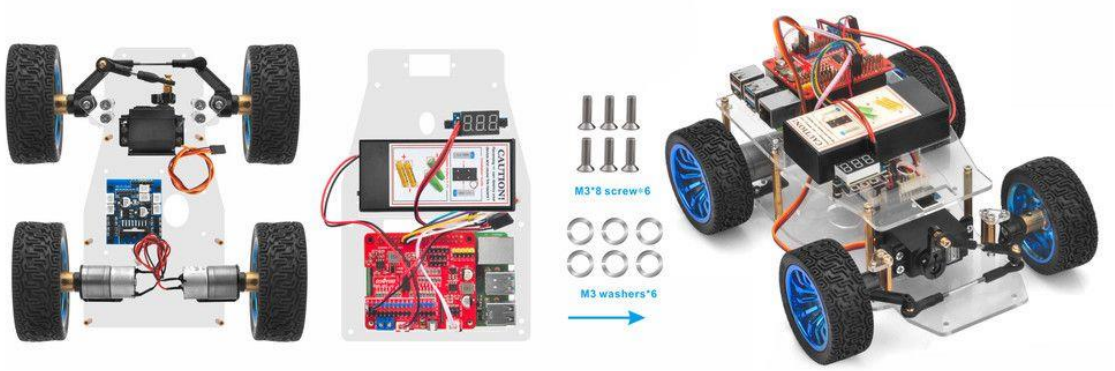


5) Connect Servo motor to PWM15 of OSOYOO PWM Hat V1.0 as following:

Servo Motor	OSOYOO PWM HAT V1.0
Brown(GND)	GND
Red(VCC)	5V
Orange(Signal)	PWM15

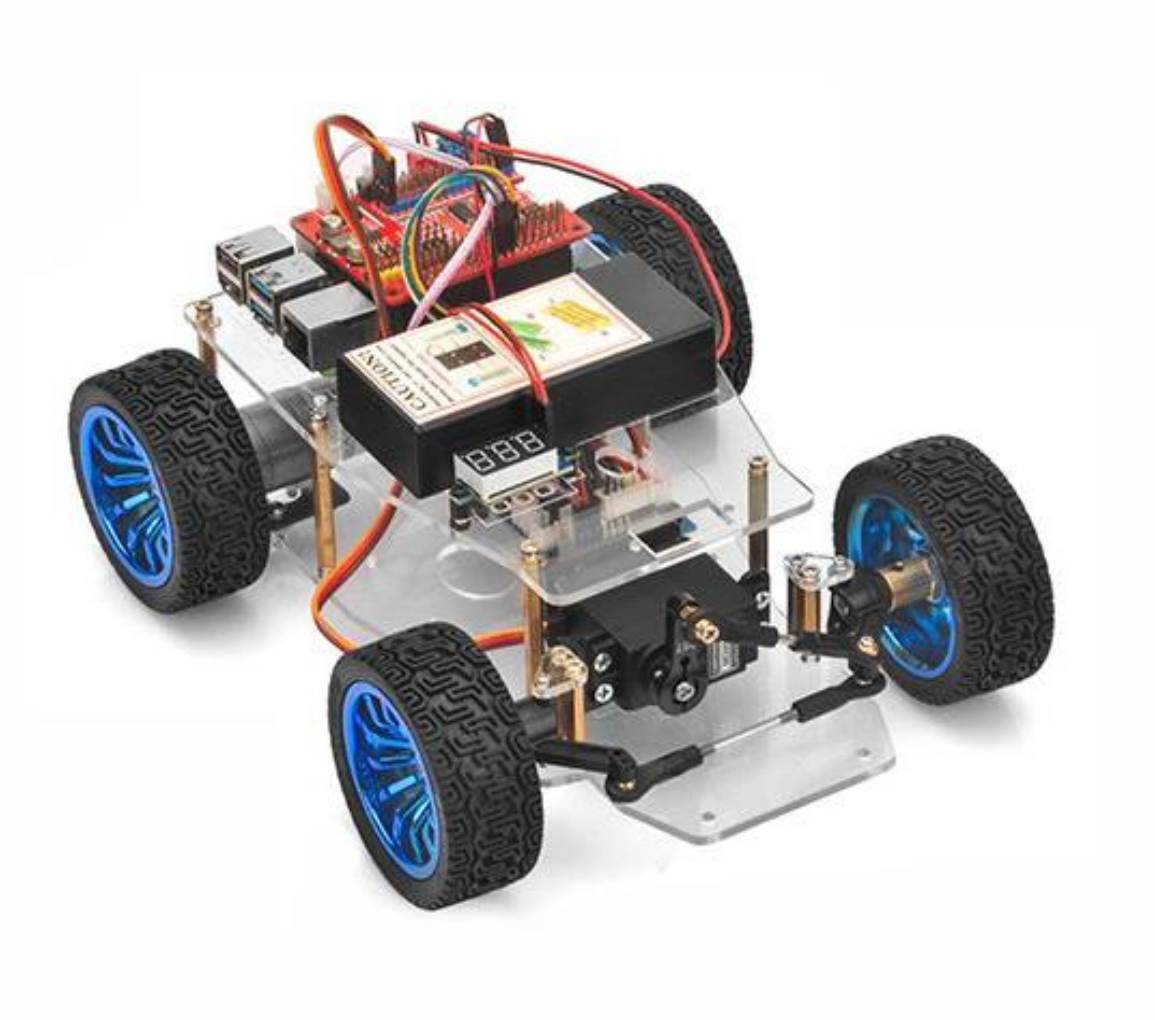


6) Connect upper chassis to lower chassis with 6 copper pillars and fix copper pillars with M3*8 screws and M3 washes as following:



7) Please install your 18650 batteries in battery box for 18650 as per following instruction **(Note: Check the box instruction and make sure polar direction is correct, otherwise it can destroy your device and cause fire hazard.):**

Now hardware installation is almost down.



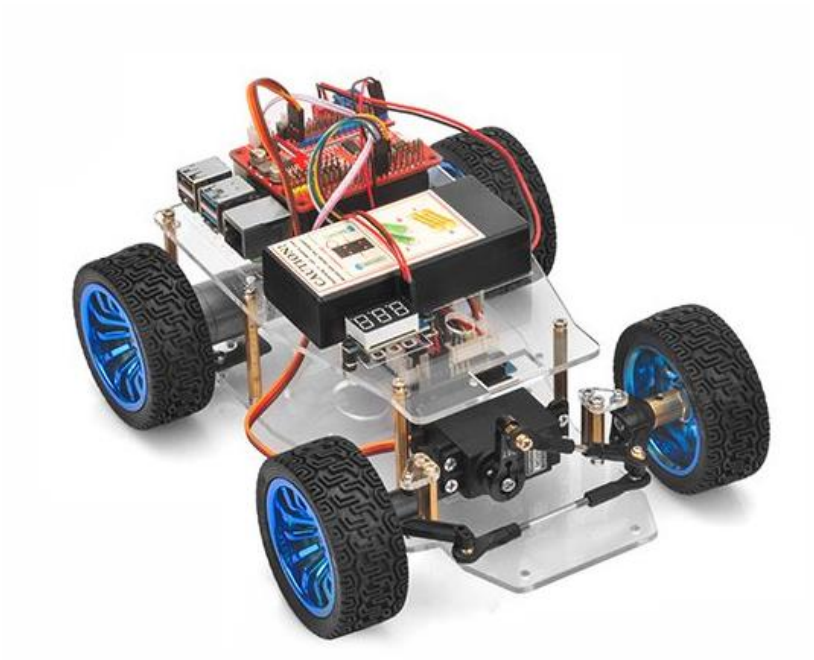
OSOYOO Servo Steer Smart Car for Raspberry Pi lesson 2: Software Installation and Basic Movement

OBJECTIVE

In this tutorial, we will simply tell you how to install Raspberry Pi OS for the Raspberry pi and how to use console to control raspberry pi. We'll use OSOYOO Servo Steer Smart Car for Raspberry Pi to do some simple movements. Once the car installation is completed, it is very important to test the installation and sample code. If you have passed the test movement of this lesson, it means Raspberry pi, motors, battery, model X board, servo motor and wire connections between these parts are all functioning well, and you can move on other sample lessons

If you don't complete the frame of car, please review lesson 1

Please enter the link to watch the video: https://youtu.be/YCcgr-RUON4



PARTS & DEVICES

- Servo Steer Smart Car for Raspberry Pi (with raspberry pi board) x1
- Micro SD card (more than 8GB) x1
- Micro SD card reader x1

- OSOYOO 5 inches DSI touch screen for Raspberry Pi x1
- PC x1

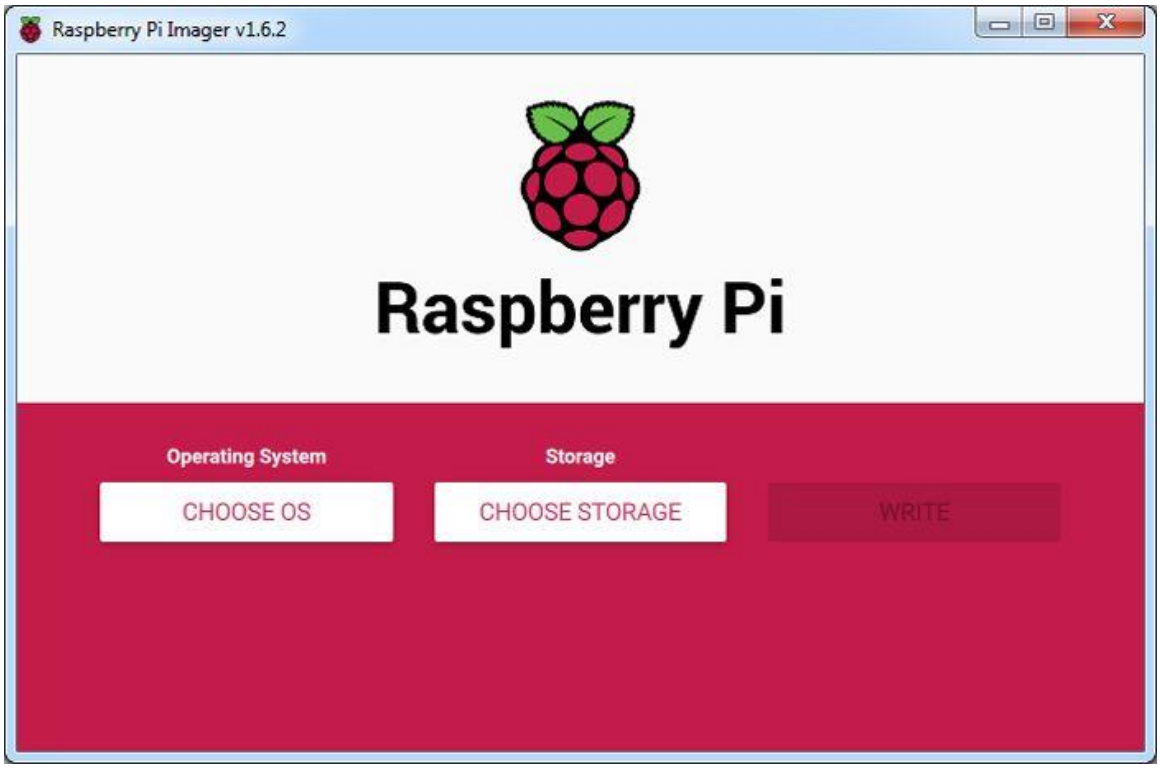
SOFTWARE PREPARATION

Raspberry pi OS (Recommend use desktop version)	Download Raspberry pi OS from: https://www.raspberrypi.org/software/operating-systems/#raspberry-pi-os-32-bit
7 zip is a free zip utility that un-zips zip file	Download 7 zip from: https://www.7-zip.org/
Raspberry Pi Imager is a imager utility	Download Raspberry Pi imager from: https://www.raspberrypi.org/software/
Notepad++ is a free source code editor and Notepad replacement that supports several languages.	Download Notepad++ from: https://notepad-plus-plus.org/downloads/
PuTTY is a SSH tool for Windows users	Download PuTTY from: https://www.chiark.greenend.org.uk/~sgtatham/putty/

INSTALLATION RASPBERRY PI OS

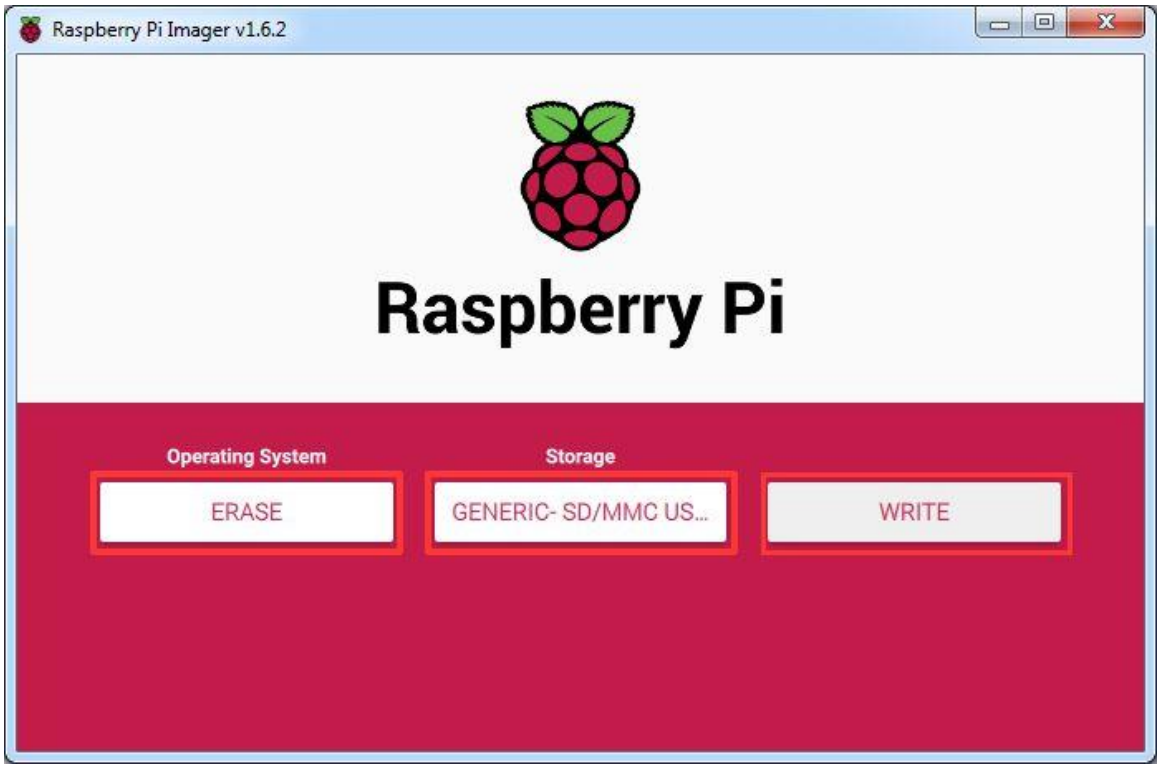
Note: there is a 16GB micro SD card which is burn 2021-05-07-raspios-buster-armhf.img in the package of OSOYOO Servo Steer Smart Car for Raspberry Pi. If you use this micro SD card, please skip the steps of Raspberry Pi OS

Step 1: Download Raspberry Pi Imager from https://www.raspberrypi.org/software/ and then install this software.

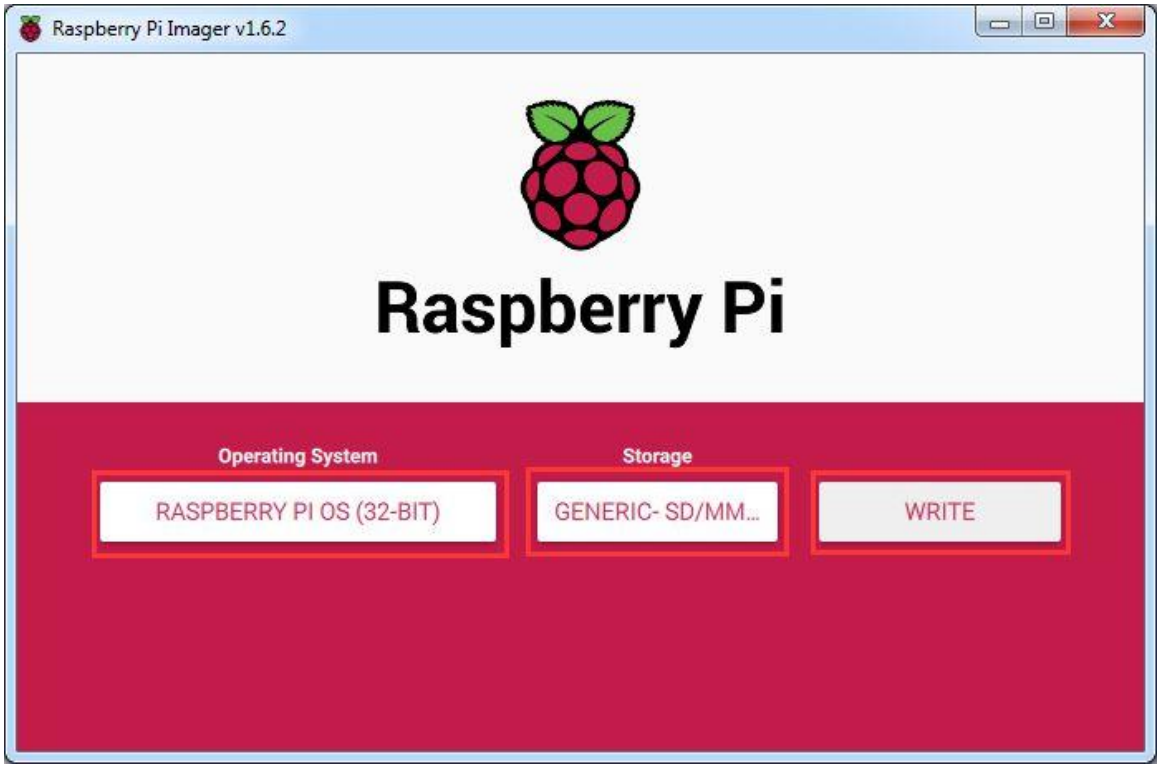


Step 2: Prepare a new Micro SD card (more than 8GB) and insert it in USB micro SD card reader, and connect USB micro SD card reader with your PC.

Step 3: If the micro SD card is not new, please open Raspberry Pi Imager, select "Erase" as Operating system, then select your micro SD card and then click "write" to format the SD card firstly as following:



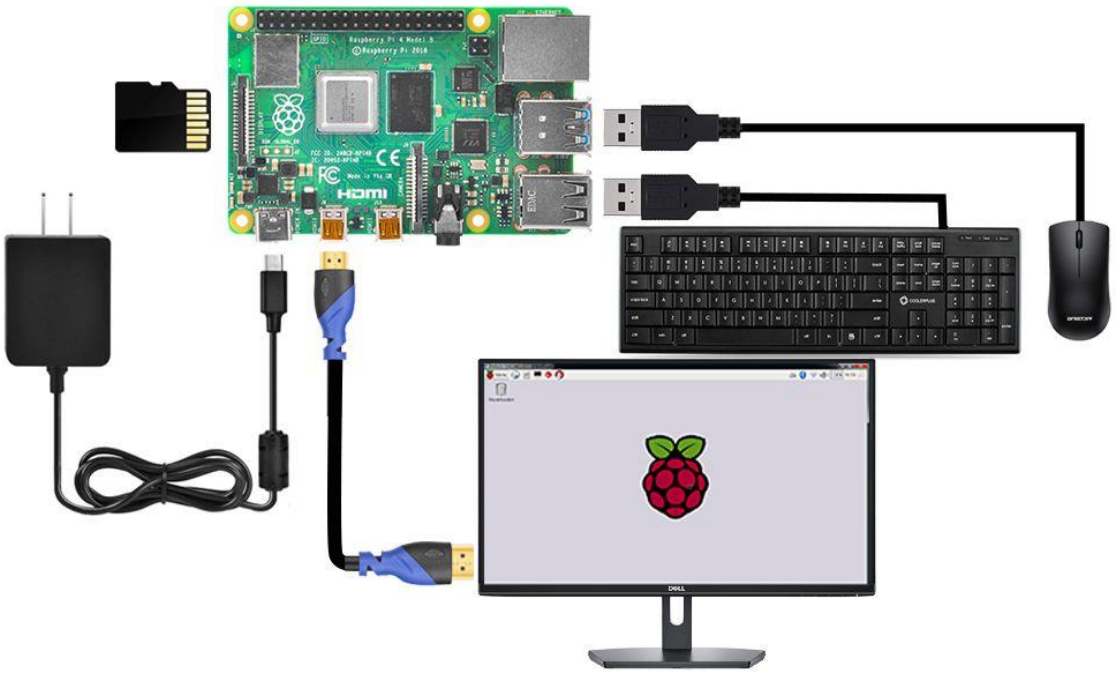
Step 4: Open Raspberry Pi Imager, select “Raspberry Pi OS (32-bit)” as Operating system, then select your micro SD card and then click “write” to burn the OS in your micro SD card *(Note: if you want to burn the OS you have download, please select “Use custom” and then browse the OS in your PC)*



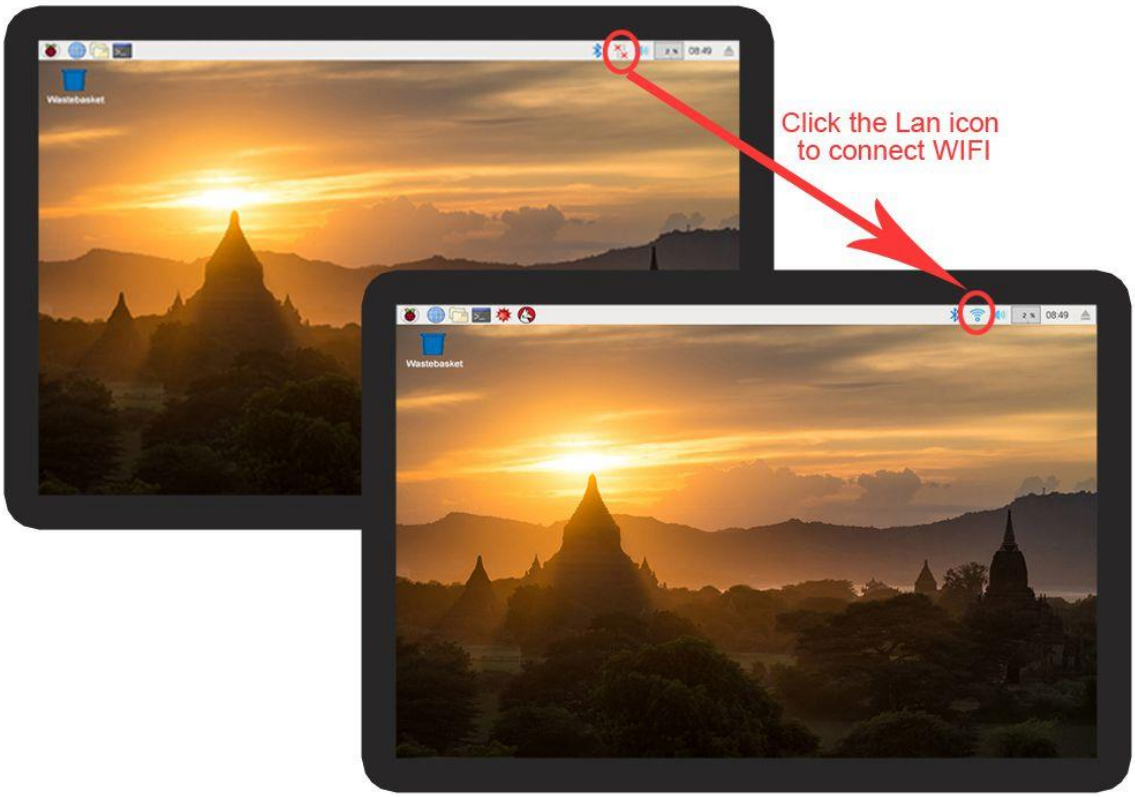
A. Using console directly

Note: A screen monitor is needed when you use the console directly.

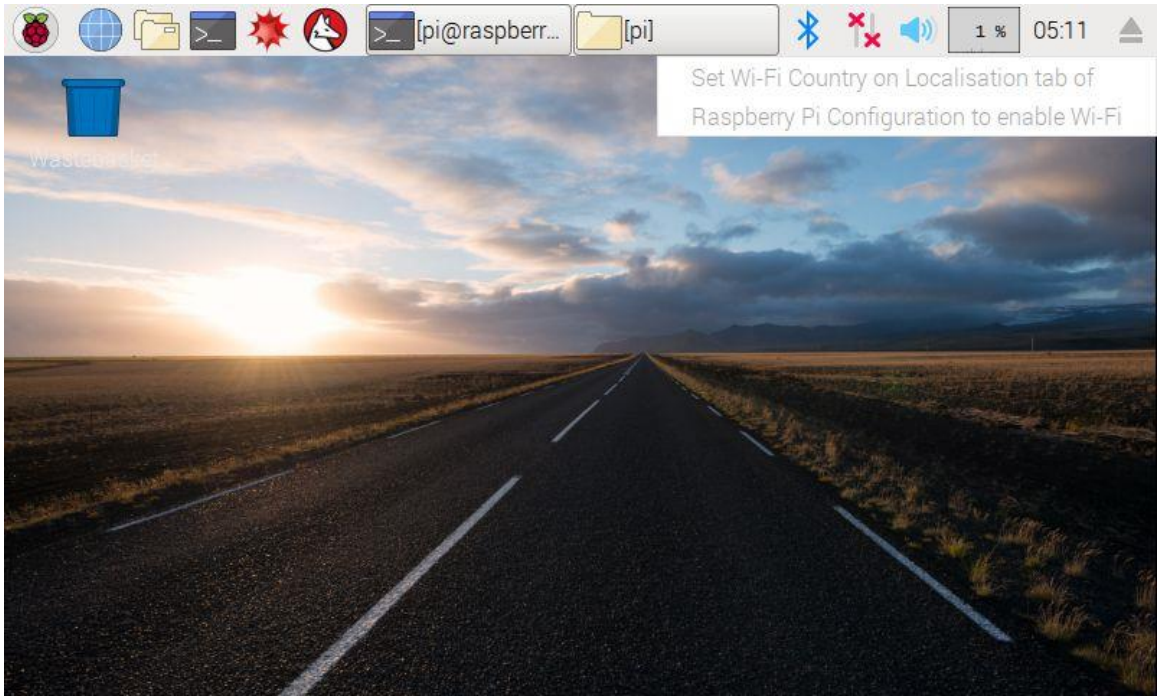
1. Connect Raspberry Pi to your HDMI monitor or TV. Put a keyboard and mouse into Raspberry Pi USB ports. Insert SD card into the slot on your Raspberry Pi as following:



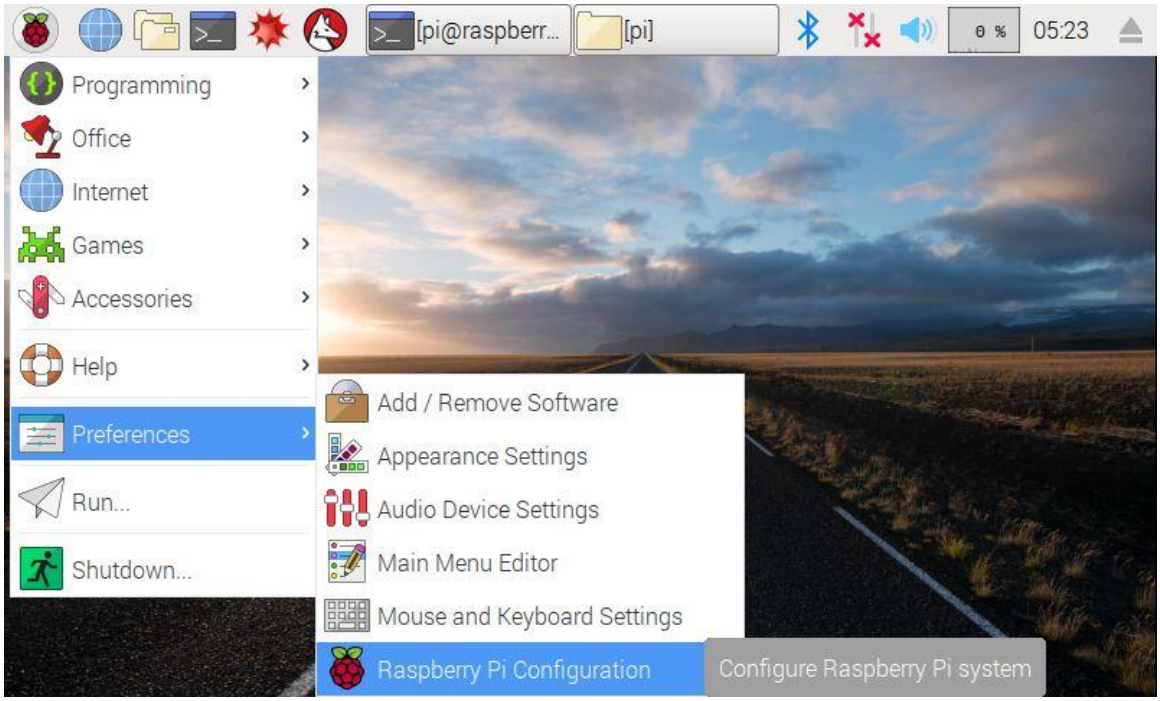
2. Please click the Lan icon at the right-upper corner, and select your WIFI SSID, and enter the pass word of your wifi to connect wifi hotspot. Then you can see the console full screen.

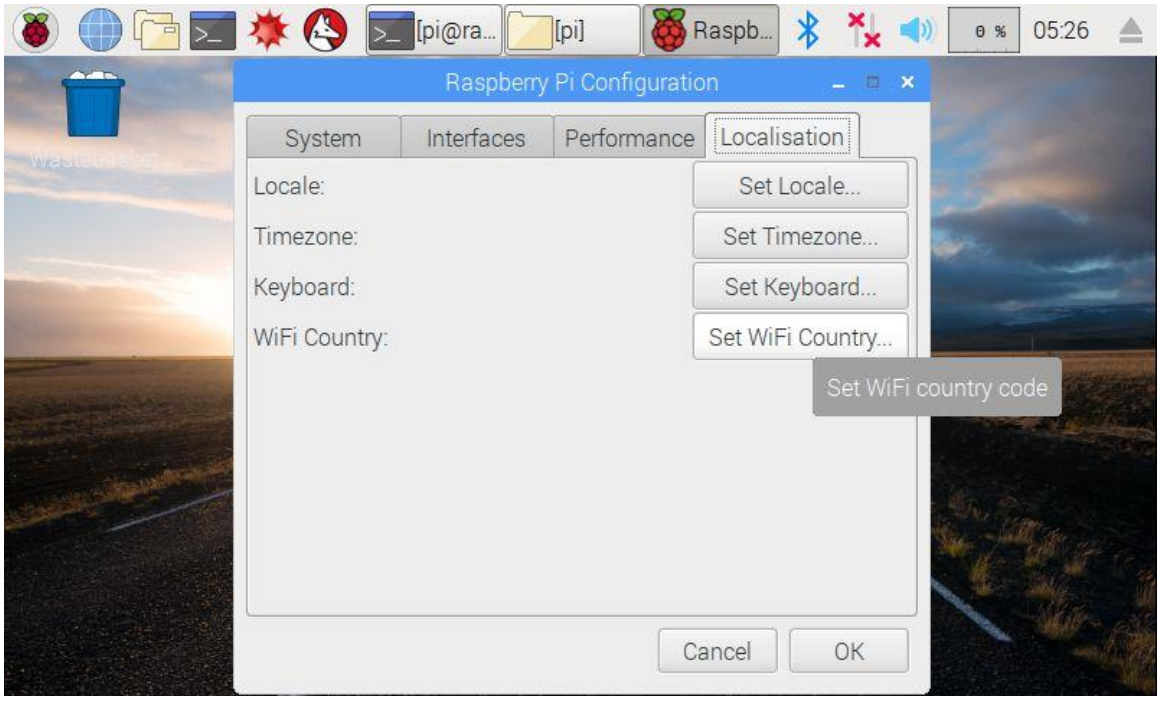


3. If you click the Lan icon, and get the notice as following:

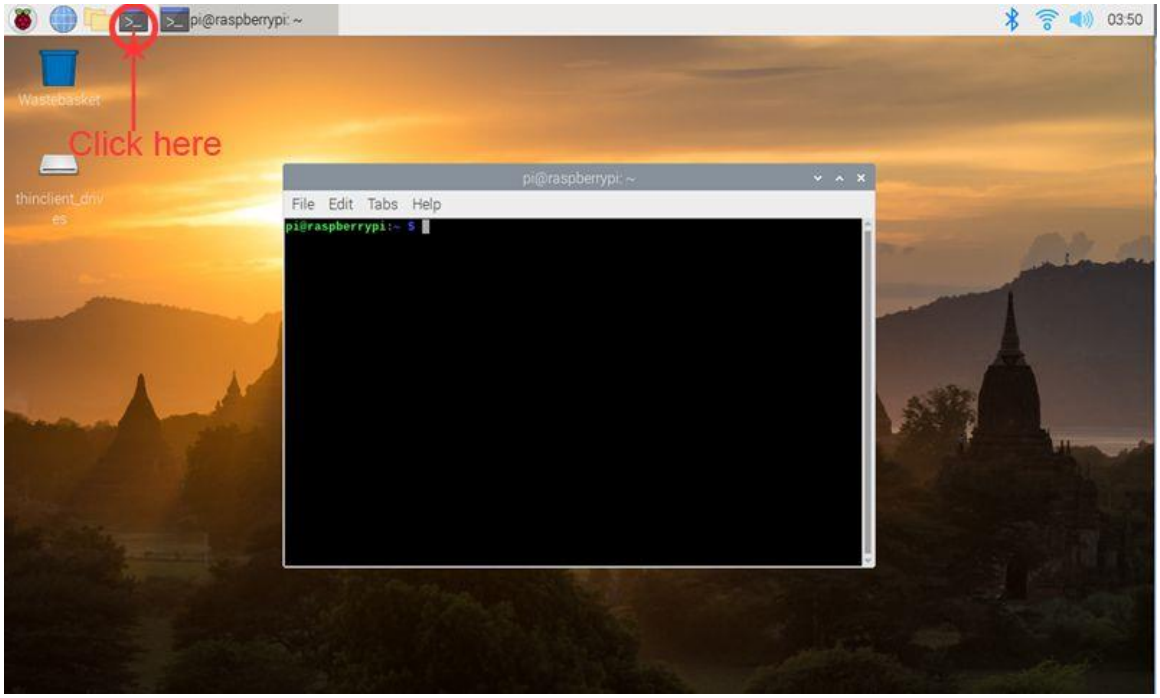


4. Please click configure the WIFI localisation as following:





5. Click the icon of **Terminal** on the screen, or press **CTRL**+**ALT**+**T** simultaneously, then a terminal will pop up as follows:

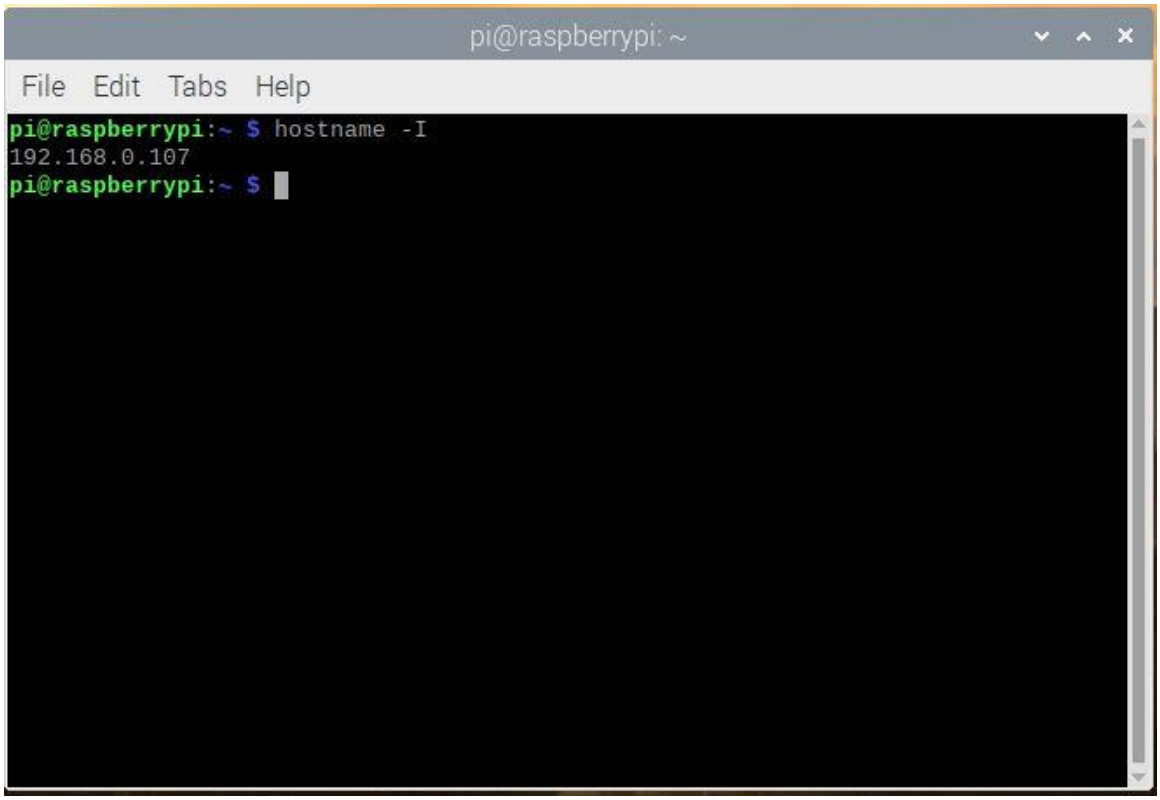


6. Find out the IP address of the RPi.

Method A: Connect your Pi to monitor and mouse, click LAN or Wifi icon to get the IP address as following photo

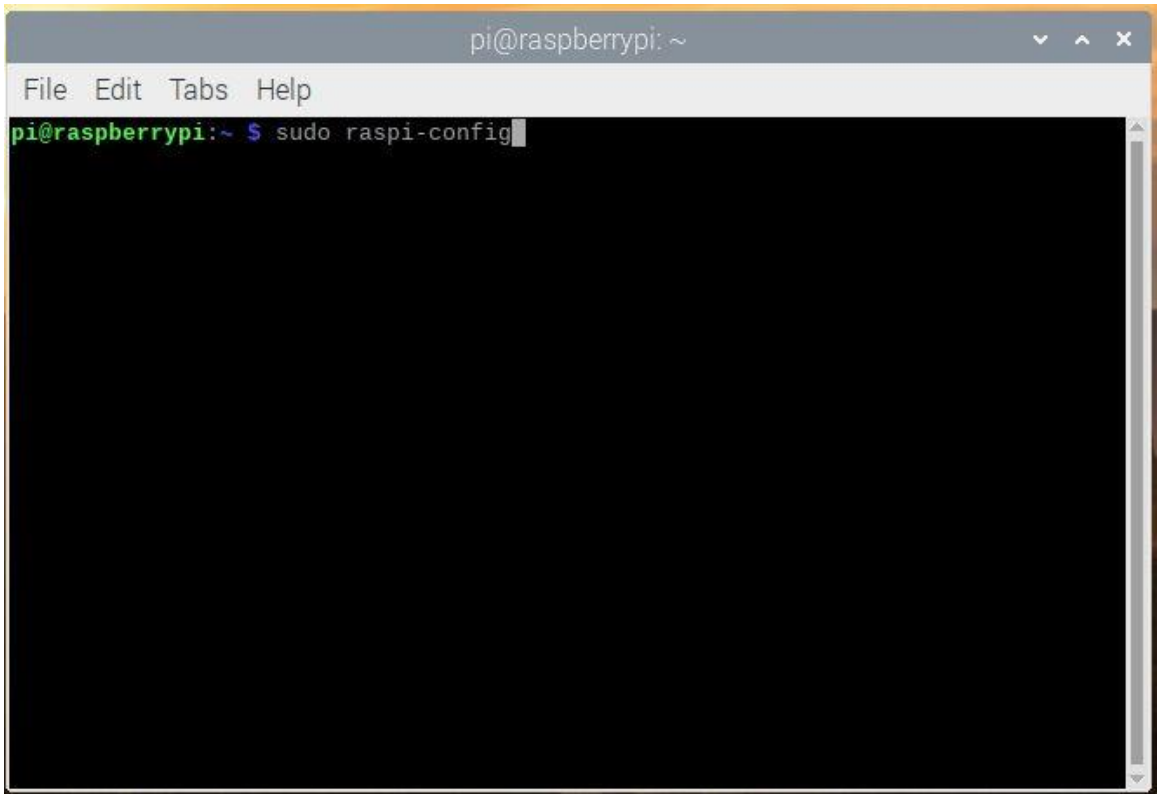
Method B: you can also find the IP address by typing terminal command:

hostname -I

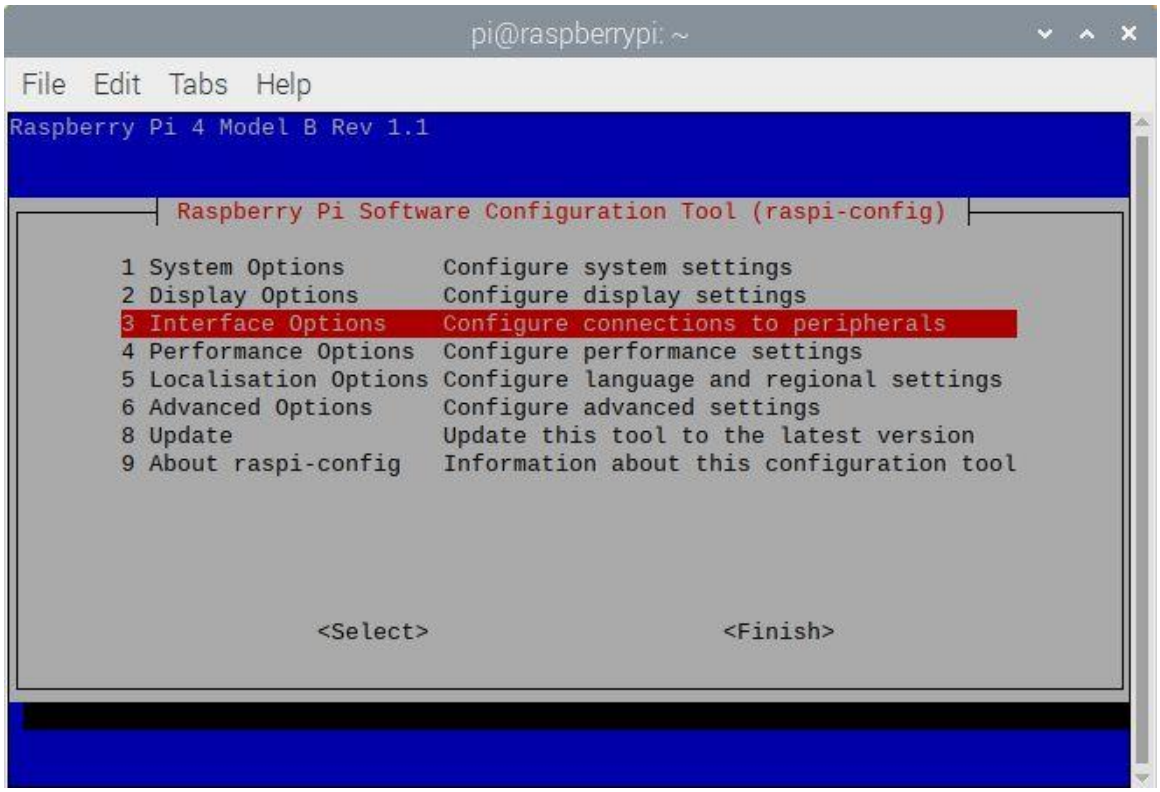


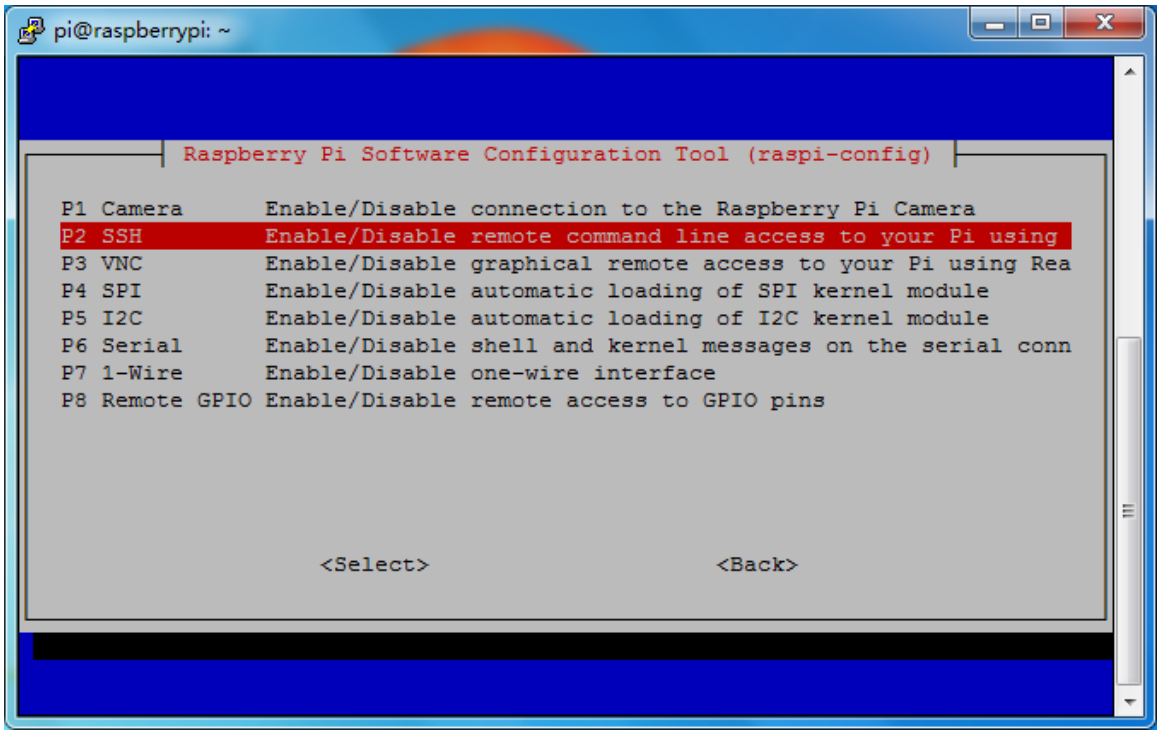
7. Enable SSH

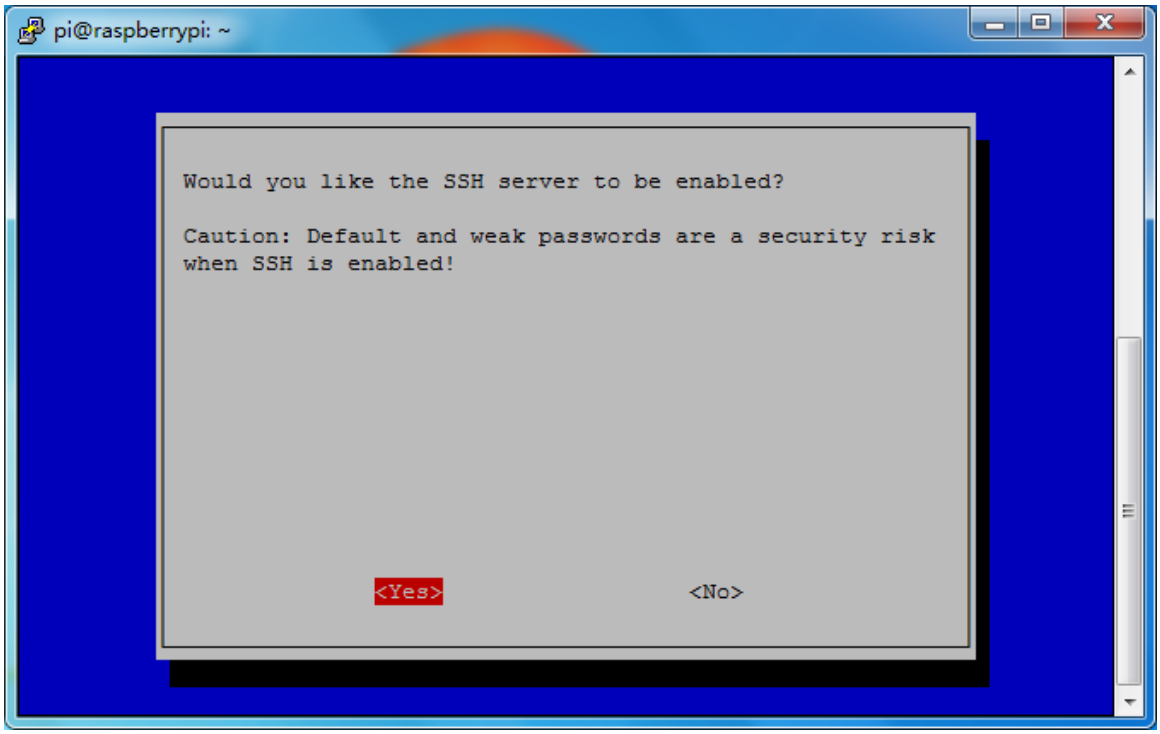
Typing terminal command: *sudo raspi-config*

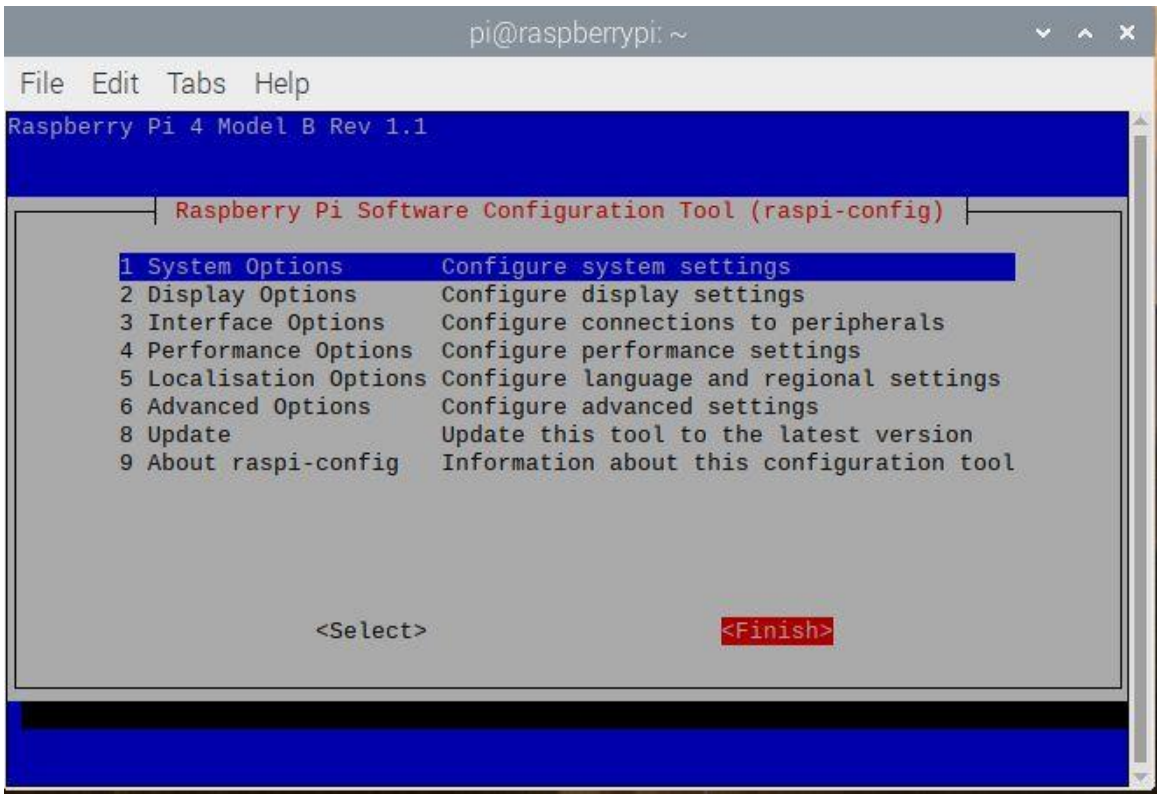


Go to configuration menu, go to **Interface Options ->SSH -> Yes -> Select ->Finish**
(Note: please use “arrows” on keyboard to position the cursor where you want to go and press “enter” to confirm the select)







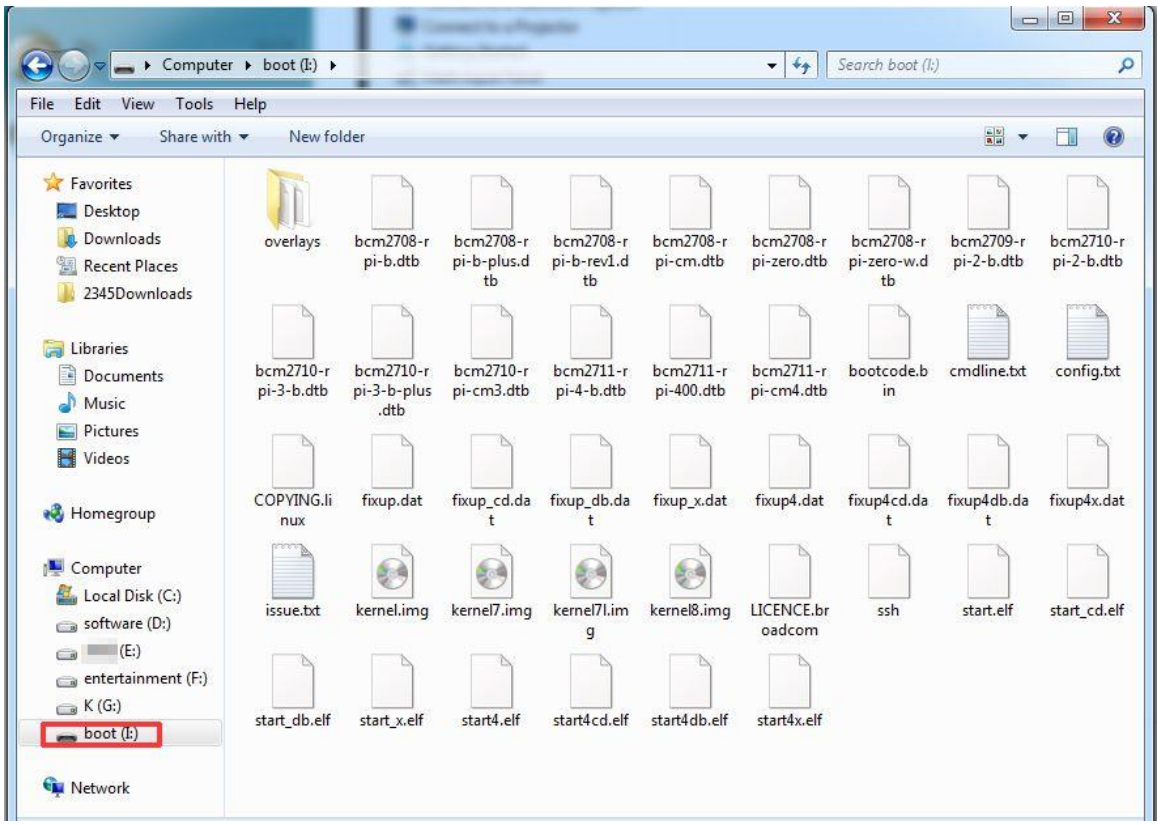


B. Using console remotely

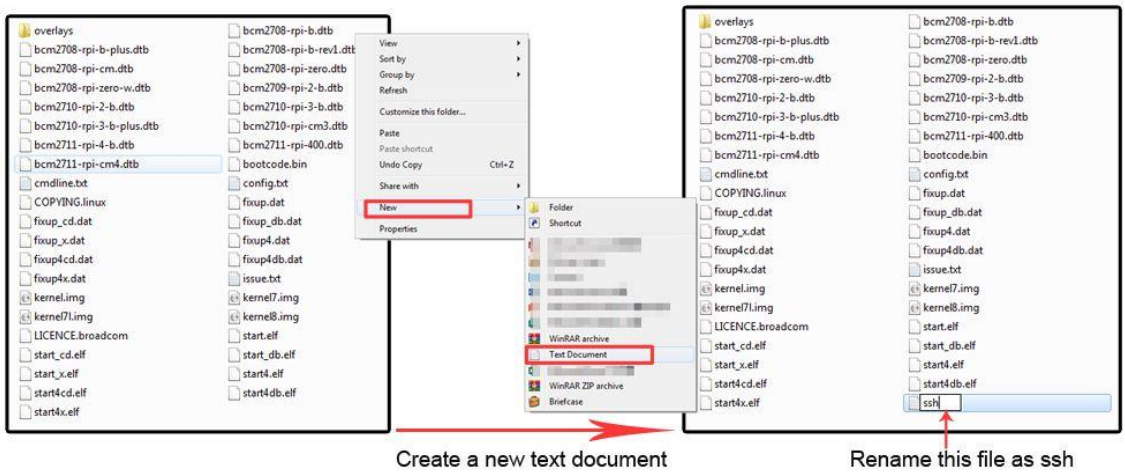
Note: For 2016-11-25 release or above, SSH (a protocol securing remote login session and other network service) is disabled by default. Therefore, when you need to log in remotely, you need to enable ssh firstly.

1. Insert Micro SD card which is burn with Raspberry Pi OS in USB micro SD card reader, and connect USB micro SD card reader with your PC.

2. Open this micro SD card in your computer



3. Create a txt. file and renamed “ssh” under /boot/ to enable it.



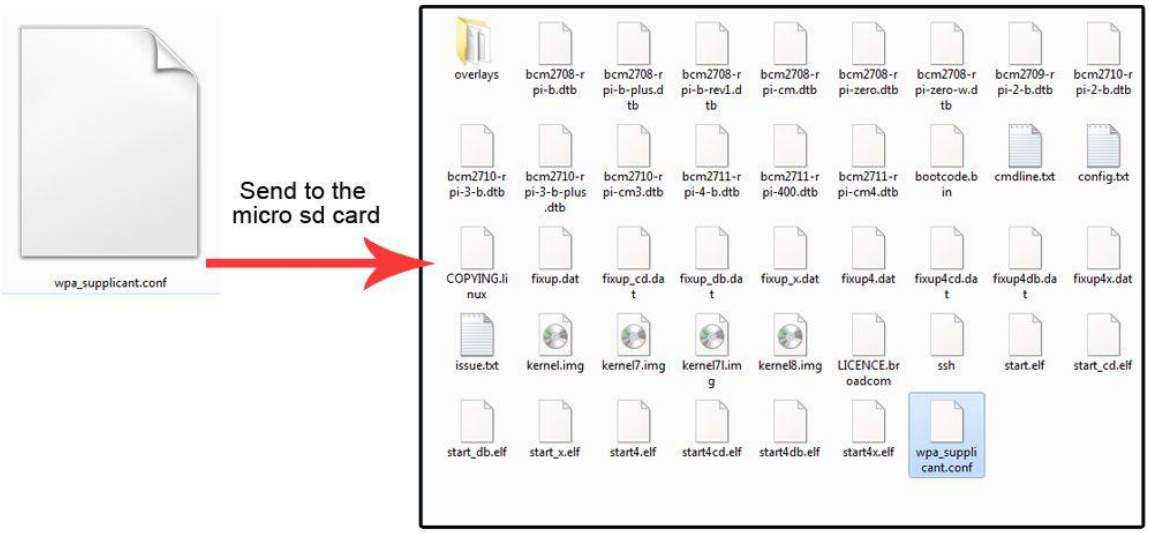
4. Download the file "wpa_supplicant.conf" and then use Notepad++ to open this file, and replace the SSID and PSK with your own wifi SSID and password and save this file:

```
ssid="Your wifi-A ssid"
psk="ssid password"
```

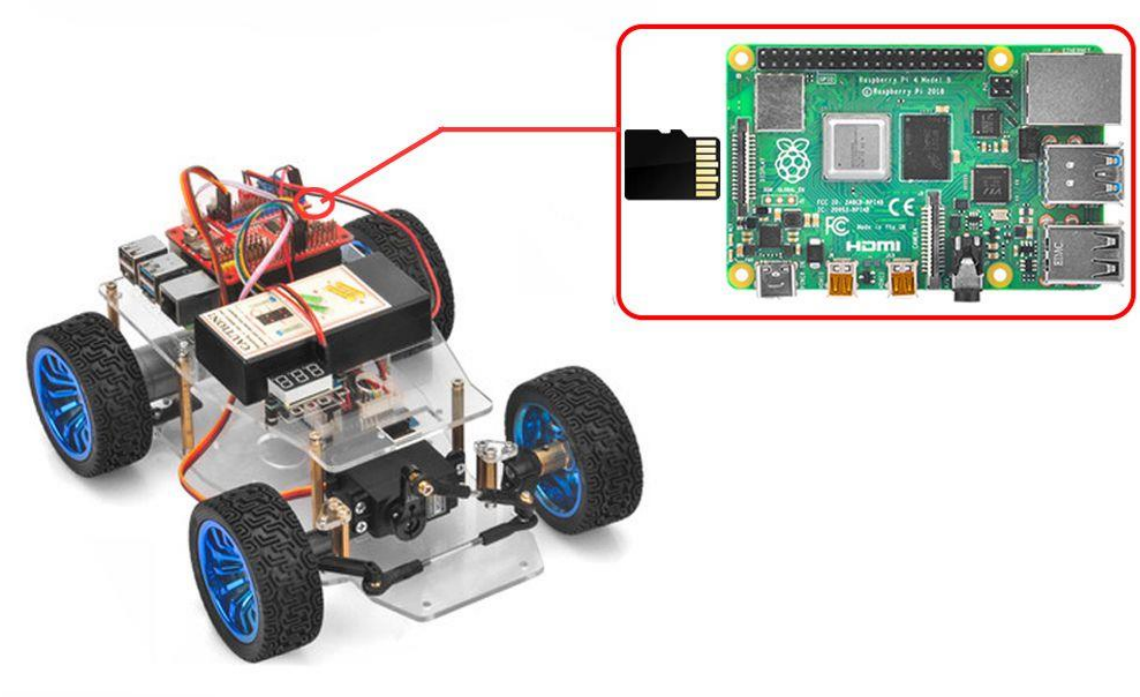
The content of "wpa_supplicant.conf" as following:

```
country=CN
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
update_config=1

network={
ssid="Your wifi-A ssid"
psk="ssid password"
key_mgmt=WPA-PSK
priority=10
}

network={
ssid="Your wifi-B ssid"
psk="ssid password"
key_mgmt=WPA-PSK
priority=2
scan_ssid=1
}
```

5. Send the file "wpa_supplicant.conf" from your PC to the micro SD card



6. Plug the MicroSD card into the Raspberry Pi and then power on the Raspberry Pi.

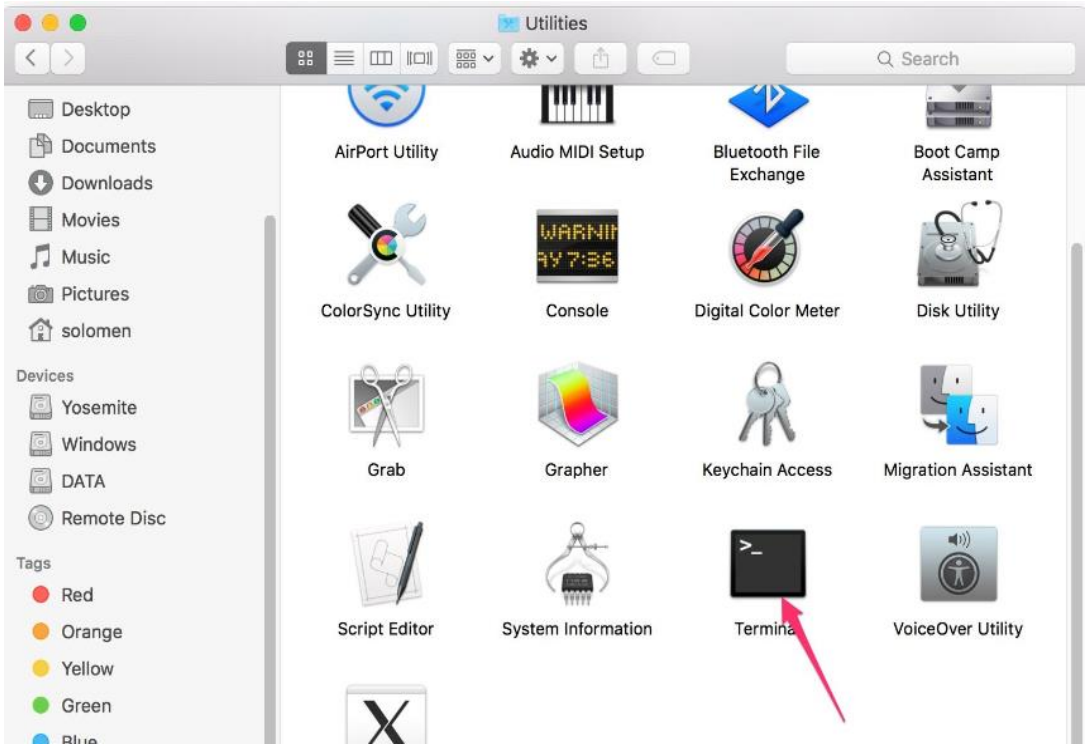


7. Login the Wifi router and check the IP address of your Raspberry Pi *(Or you can use some IP address scan APP to scan the IP address)*

8. Use ssh tool to control Raspberry Pi remotely

Remark: For three platforms: Windows, Mac and Linux, it might be a little bit different to do this.

(1) Linux and Mac users can easily log into the Raspberry Pi via ssh. On Linux or Mac, find Terminal and open it.

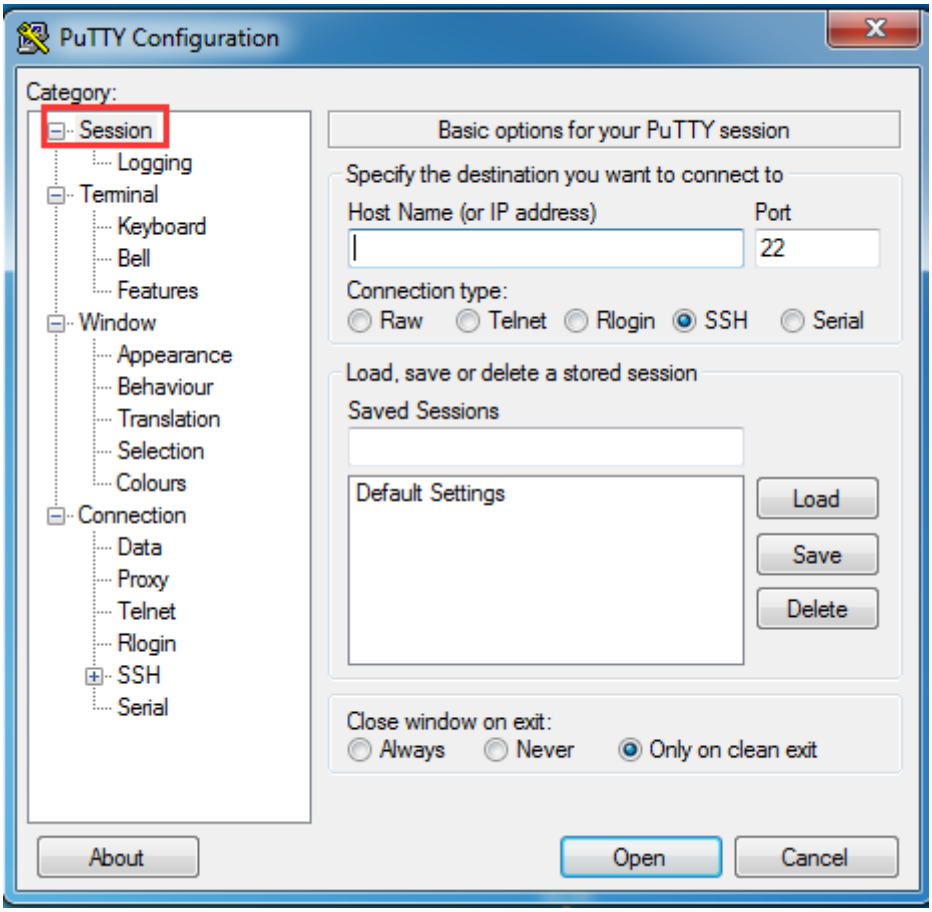


Type in ssh pi@IP address (ssh is the tool for remote login; pi is the user name, and as the name suggests, your RPi's IP address) and then press **Enter** to confirm. For example:
ssh pi@192.168.0.107

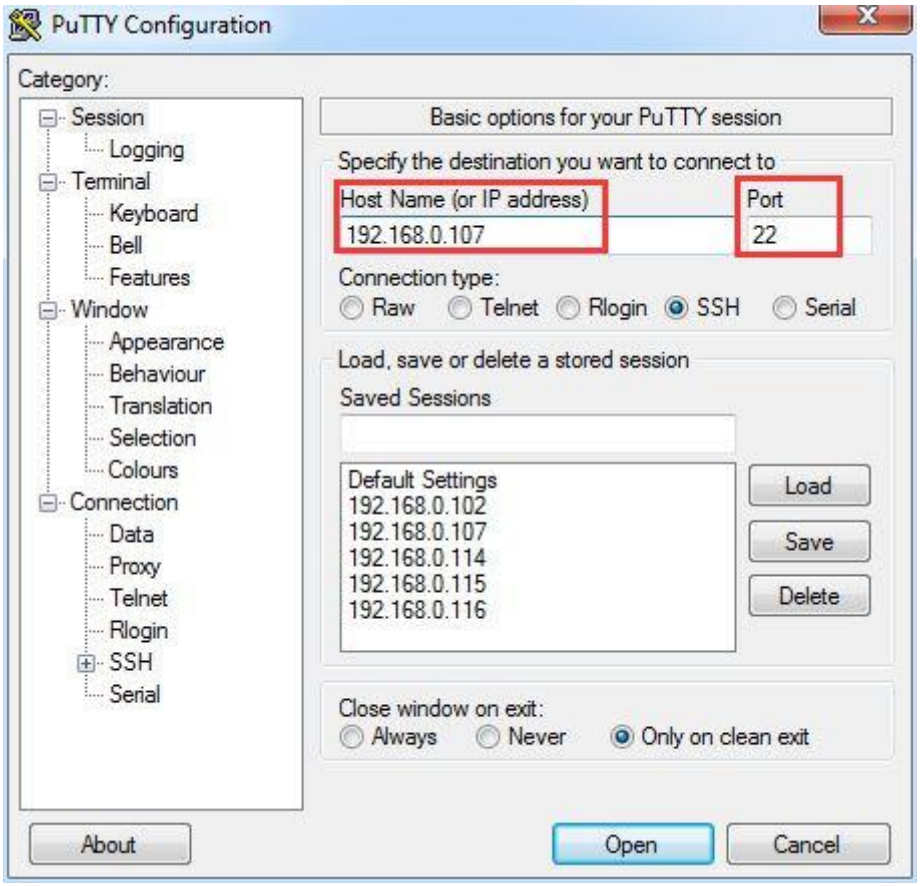
If you get a prompt that no ssh is found, you need to install a ssh tool like Ubuntu and Debian by yourself:
sudo apt-get install ssh

(2) For Windows users, you may use a ssh tool to log into Raspberry Pi remotely, like PuTTY.
Step 1. Download PuTTY from:
https://www.chiark.greenend.org.uk/~sgtatham/putty/ and install this exe. in your Windows PC

Step 2. Open PuTTY and click Session on the left tree-alike structure (generally it's collapsed upon PuTTY startup):

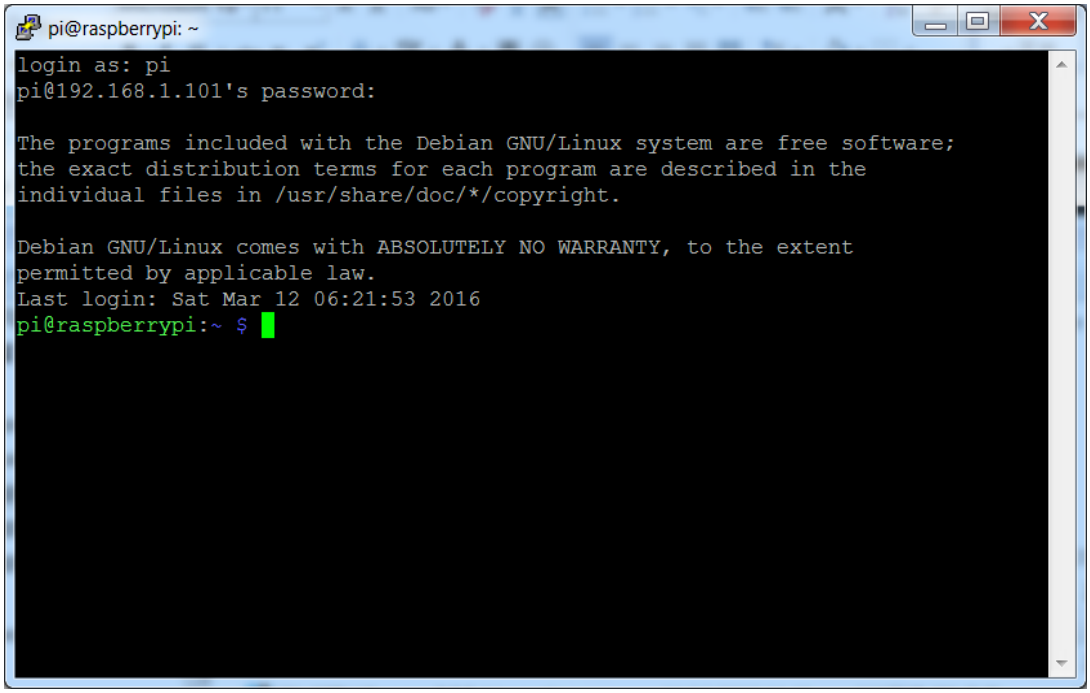


Step 3. Enter the IP address you got into the textbox under Host Name (or IP address) and 22 under Port (by default it is 22), then click open.



Step 4. Note that when you first log in to the Raspberry Pi with the IP address, you'll be prompted with a security reminder. Just click **Yes**. When the PuTTY window prompts login as: type in the user name: **pi**, and password: **raspberry** (the default one, if you haven't changed it).

Note: when you're typing the password in, the window shows nothing just null, but you're in fact is typing things in. So just focus on typing it right and press Enter. After you log in the RPi successfully, the window will display as follows:



For other platforms, please contact your supplier.

STEERING SERVO ALIGNMENT

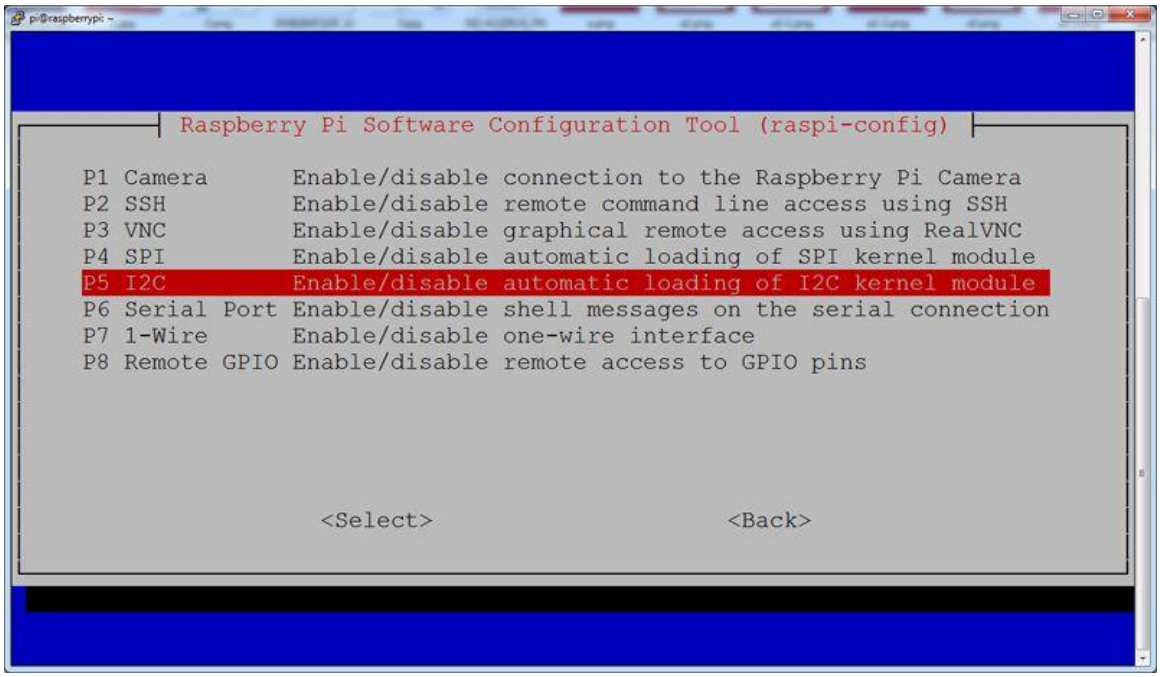
After learning how to install Raspberry Pi OS and use the console of Raspberry Pi, you can follow the next steps to test the frame work of the robot car in lesson1.

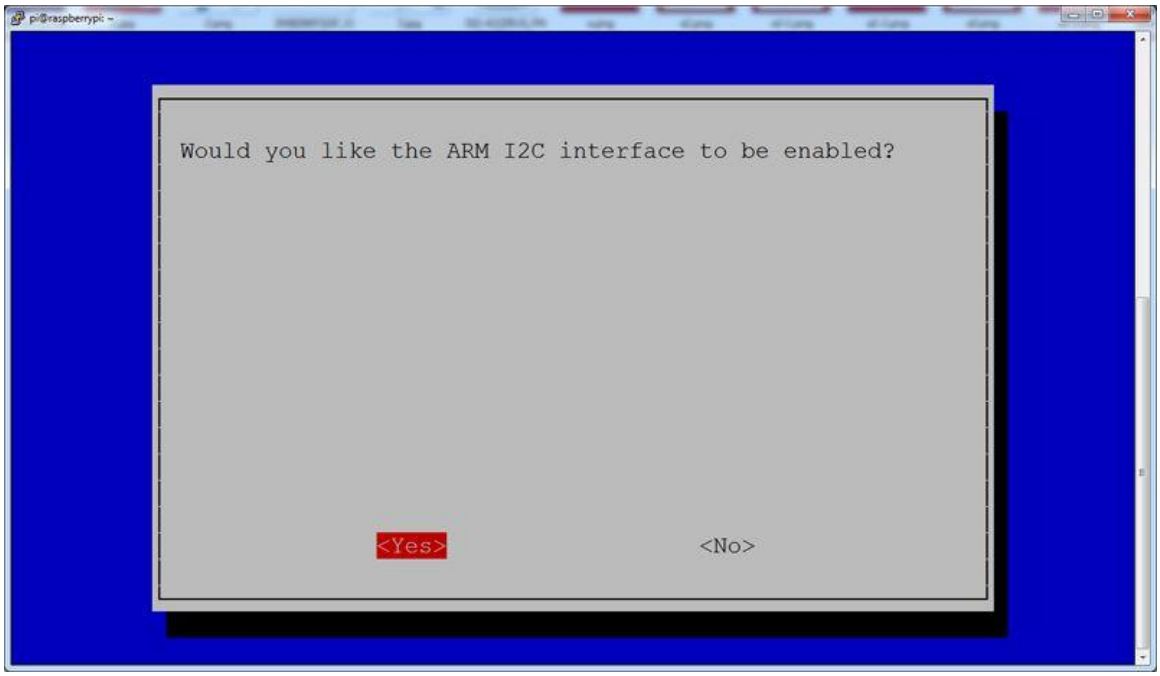
Note: In our sample lessons, we use Windows PC as the remote console device and PuTTY as the ssh tool

Step 1: Power the car and run the following command and enable the I2C (I2C is a protocol which will be used to exchange data with I2C device)

sudo raspi-config

Then select Interfacing Options->I2C->Yes->Select->Finish





Step 2: Install GPIO and pca9685 PWM Library by running the following commands:

```
cd ~
sudo apt-get install rpi.gpio
sudo pip install adafruit-pca9685
```

```
pi@raspberrypi:~ $ cd ~
pi@raspberrypi:~ $ sudo apt-get install rpi.gpio
Reading package lists... Done
Building dependency tree
Reading state information... Done
Note, selecting 'python-rpi.gpio-dbgsym' for regex 'rpi.gpio'
Note, selecting 'python3-rpi.gpio' for regex 'rpi.gpio'
Note, selecting 'rpi.gpio-common' for regex 'rpi.gpio'
Note, selecting 'python-rpi.gpio' for regex 'rpi.gpio'
Note, selecting 'python3-rpi.gpio-dbgsym' for regex 'rpi.gpio'
python-rpi.gpio is already the newest version (0.7.0-0.1~bpo10+4).
python3-rpi.gpio is already the newest version (0.7.0-0.1~bpo10+4).
rpi.gpio-common is already the newest version (0.7.0-0.1~bpo10+4).
rpi.gpio-common set to manually installed.
The following NEW packages will be installed:
  python-rpi.gpio-dbgsym python3-rpi.gpio-dbgsym
0 upgraded, 2 newly installed, 0 to remove and 0 not upgraded.
Need to get 88.2 kB of archives.
After this operation, 126 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
```

```
pi@raspberrypi:~ $ sudo pip install adafruit-pca9685
Looking in indexes: https://pypi.org/simple, https://www.piwheels.org/simple
Collecting adafruit-pca9685
  Downloading https://files.pythonhosted.org/packages/d1/3a/be3c9b67b3b5bf44b956
7853d968c0979268ffef5f18c8287a47dd6fa60f/Adafruit_PCA9685-1.0.1-py2-none-any.whl
Collecting Adafruit-GPIO>=0.6.5 (from adafruit-pca9685)
  Downloading https://files.pythonhosted.org/packages/db/1c/2dc8a674514219f287fa
344e44cadfd77b3e2878d6ff602a8c2149b50dd8/Adafruit_GPIO-1.0.3.tar.gz
Collecting adafruit-pureio (from Adafruit-GPIO>=0.6.5->adafruit-pca9685)
  Downloading https://files.pythonhosted.org/packages/93/e4/e6e25699445b4d8aafa9
7ed705ed43b39bcd6db17127ea7073aeaa76aad8/Adafruit_PureIO-1.0.1.tar.gz
```

Step 3: Run following commands to find the front value of the servo:

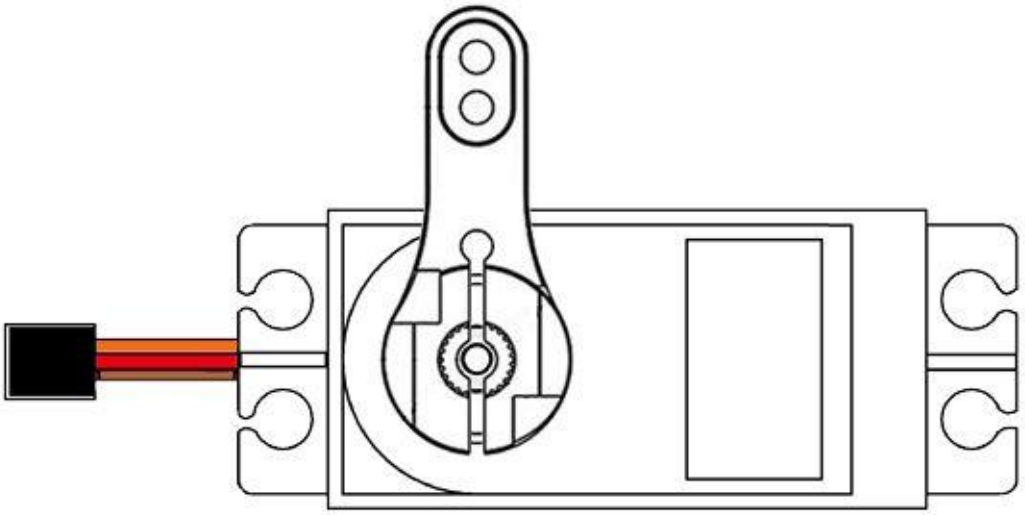
wget http://osoyoo.com/driver/servo-steer-car/servo.py
python servo.py

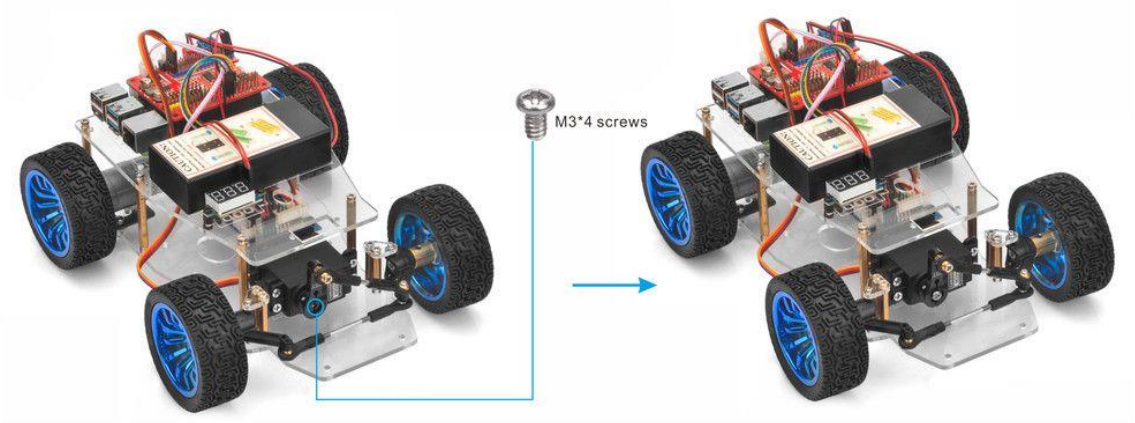
```
pi@raspberrypi:~ $ wget http://osoyoo.com/driver/servo-steer-car/servo.py
--2021-06-23 07:55:43--  http://osoyoo.com/driver/servo-steer-car/servo.py
Resolving osoyoo.com (osoyoo.com)... 51.79.21.80
Connecting to osoyoo.com (osoyoo.com)|51.79.21.80|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1020 [text/x-python]
Saving to: 'servo.py'

servo.py            100%[==================>]    1020  --.-KB/s    in 0s

2021-06-23 07:55:43 (24.2 MB/s) - 'servo.py' saved [1020/1020]

pi@raspberrypi:~ $ python servo.py
pi@raspberrypi:~ $
```

Step 4: If your steering servo does not facing face to center front direction, then you have two options:

Option 1) Turn off the power and remove the horn from the servo motor. Install the arm as the follow picture. And then install M3*4 screws on the middle of the horn to

fix the horn.





Option 2) Install M3*4 screws on the middle of the horn to fix the horn and enter the following command in terminal:

```
sudo nano servo.py
```

You can change the default value of **line 20**

```
CENTER= 425 #Steer servo car go forward
```

```
  GNU nano 3.2                          servo.py

# Import the PCA9685 module.
import Adafruit_PCA9685
import RPi.GPIO as GPIO
# Initialise the PCA9685 using the default address (0x40).
pwm = Adafruit_PCA9685.PCA9685()

servo_pin = 15 #  servo connect to PWM 15

RIGHT = 465 #Steer servo car turn right
CENTER= 425 #Steer servo car go forward
LEFT = 385 #Steer servo car turn left

# Set frequency to 60hz, good for servos.
pwm.set_pwm_freq(60)
pwm.set_pwm(servo_pin, 0, LEFT)
time.sleep(1)
pwm.set_pwm(servo_pin, 0, RIGHT)
time.sleep(1)
pwm.set_pwm(servo_pin, 0, CENTER)
                [ Read 29 lines (Converted from DOS format) ]
^G Get Help  ^O Write Out ^W Where Is  ^K Cut Text  ^J Justify   ^C Cur Pos
^X Exit      ^R Read File ^\ Replace   ^U Uncut Text^T To Spell  ^  Go To Line
```

If your steering wheels tends to left, increase the value from 425 to 430, 435, 440 …
If your steering wheels tends to right, decrease the value from 425 to 420, 415, 410 …

Then click “ctrl” + “x”, and then Y to save this file and run the command: *python servo.py* again. Repeat these steps until your steering wheels finally faces to front, and write down this value to change the lesson2,3,4,5,6 code and make steering servo always facing front at default FRONT value

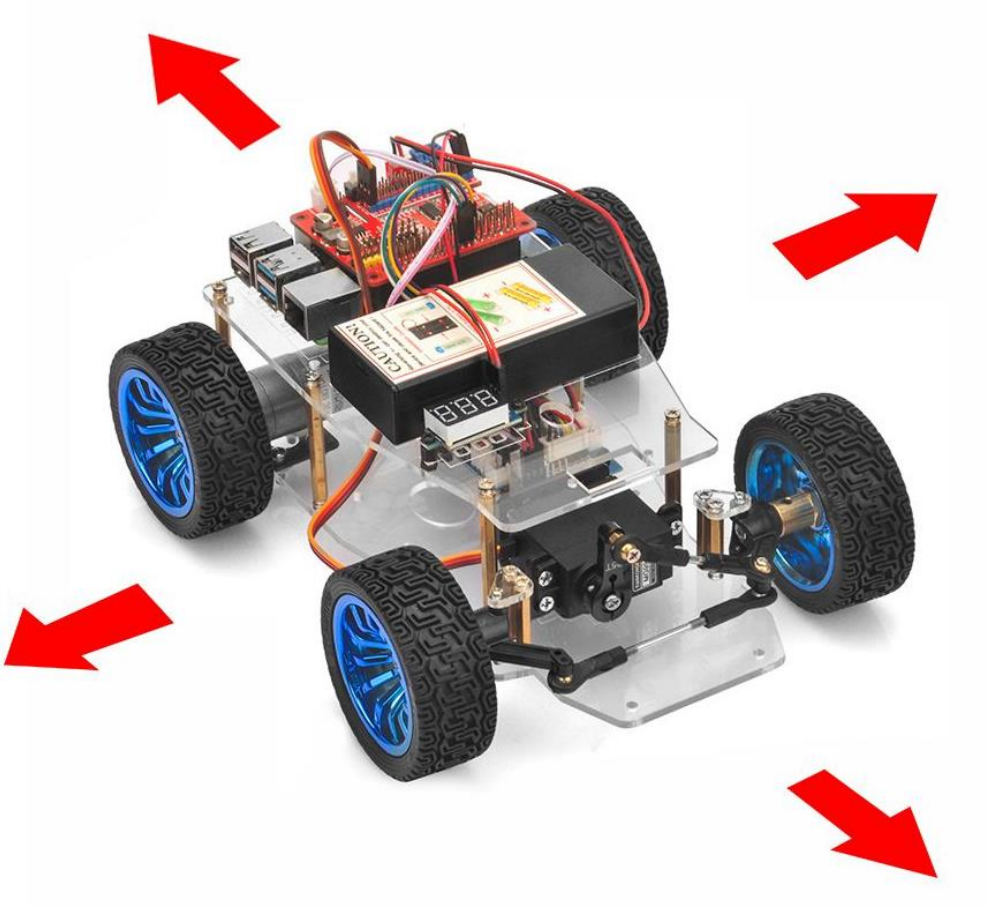
BASIC MOVEMENT

1. Power the car and type the following commands in the terminal:

```
wget http://osoyoo.com/driver/servo-steer-car/pi-basic.py
python pi-basic.py
```

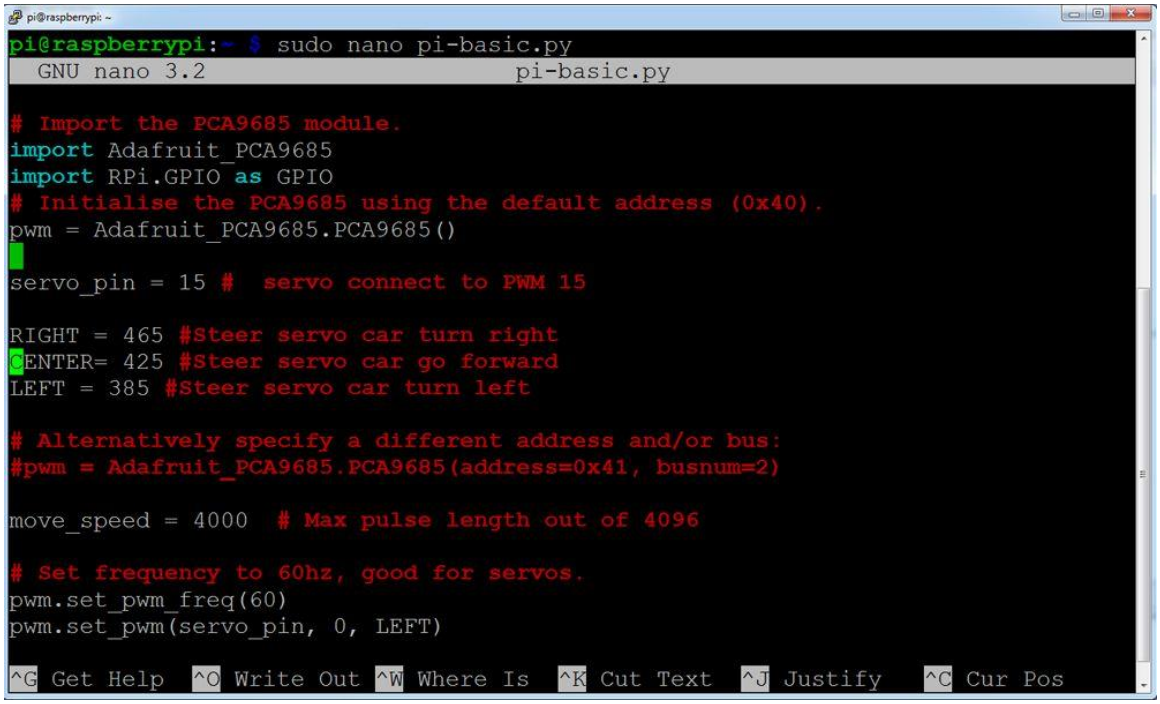
```
pi@raspberrypi:~ $ wget http://osoyoo.com/driver/servo-steer-car/pi-basic.py
--2021-06-30 03:35:39--  http://osoyoo.com/driver/servo-steer-car/pi-basic.py
Resolving osoyoo.com (osoyoo.com)... 51.79.21.80
Connecting to osoyoo.com (osoyoo.com)|51.79.21.80|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 3031 (3.0K) [text/x-python]
Saving to: 'pi-basic.py'

pi-basic.py         100%[==================>]   2.96K  --.-KB/s    in 0.002s

2021-06-30 03:35:39 (1.22 MB/s) - 'pi-basic.py' saved [3031/3031]

pi@raspberrypi:~ $ python pi-basic.py
pi@raspberrypi:~ $
```

2. After above python is running, you motors will move forward and then move backward , and turn left for then turn right, then back to right and finally back to left.



3. Please run the following command to change the default value 425 in line 20 as the FRONT value. If you align the wheel direction manually, please skip this step.

```
sudo nano pi-basic.py
```



You can download the sample python code from https://osoyoo.com/driver/p3-car/v3car-basic.py and read it via Notepad++. If you have some basic python knowledge, you can easily understand how to customize the code for your own application.

OSOYOO Servo Steer Smart Car for Raspberry Pi Lesson 3: Line Tracking

OBJECTIVE

In this lesson, we use our Raspberry Pi robot car to automatically drive along a black line in white ground . We will use 5-point IR tracking sensors to detect the line.

PARTS & DEVICES

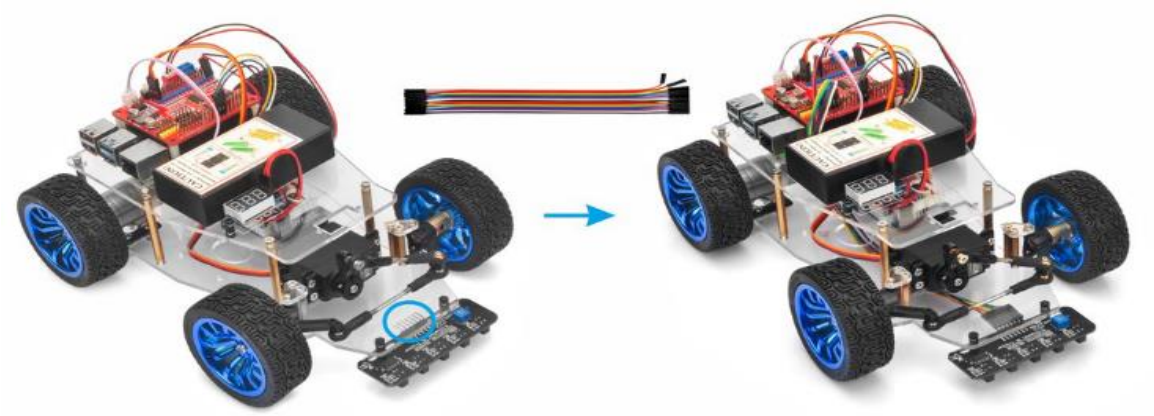
No.	Picture	Device	Qty.	Accessories	Link
1		Tracking sensor module	1	M2.5 Plastic Screw x 2 M2.5 Plastic Nut x2 M2.5 Plastic Pillar x 2	Click here to buy
2		7pin 25cm Female to Female Cable	1		Click here to buy
3		Philips screwdriver	1		Click here to buy

HARDWARE INSTALLATION

Step 1: You must complete Lesson 2 basic frame work

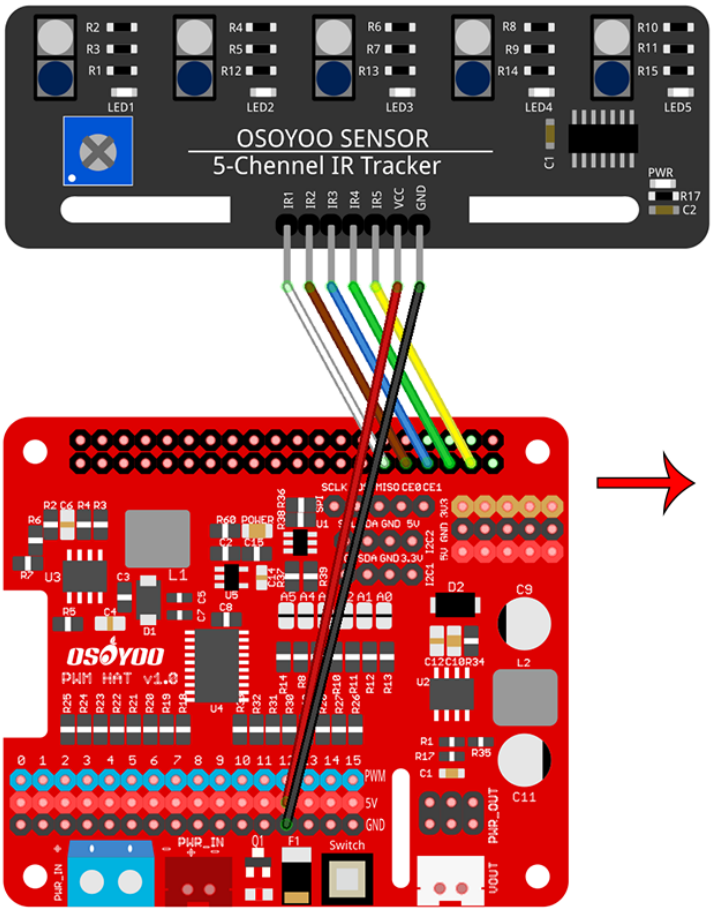
Step 2: Install tracking sensor modules under lower car chassis with 2pcs M2.5 plasctic screws, M2.5 plastic pillars and M2.5 plastic

nuts.

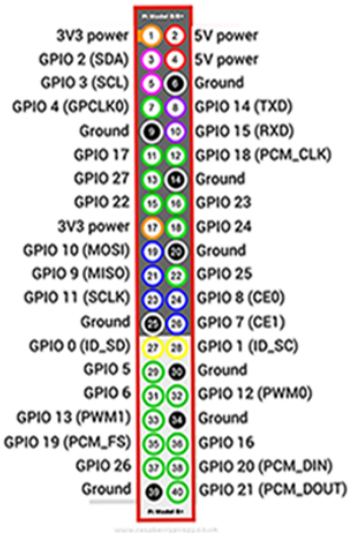


CIRCUIT CONNECTION

Step1: Connect GND-VCC pin of tracking sensor module to GND-5V of osoyoo PWM HAT board ; connect IR1, IR2, IR3, IR4, IR5 pins to GPIO5, GPIO6, GPIO13, GPIO19, GPIO26 of Raspberry pi with 7pin 25cm female to female cable as the following photo shows (Remember : DO NOT remove any existing wires installed in Lesson 1):

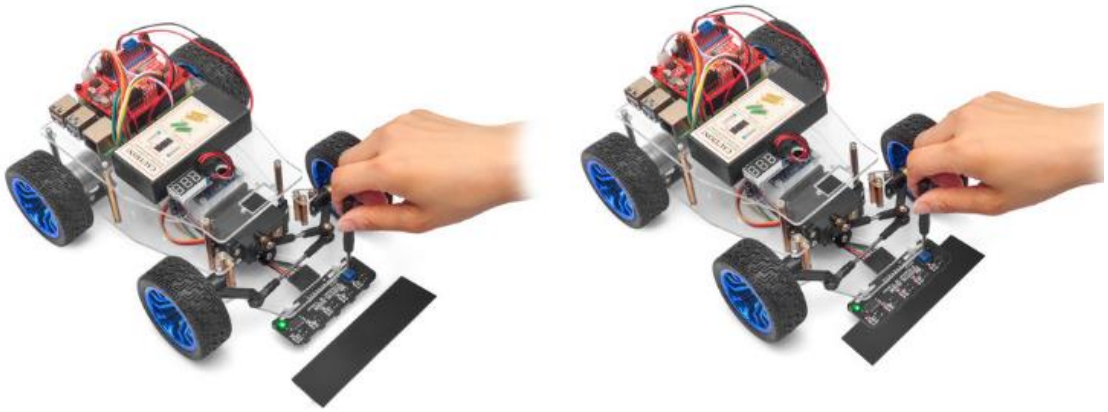


Tracking sensor	OSOYOO PWM hat v1.0
IR1	GPIO5
IR2	GPIO6
IR3	GPIO13
IR4	GPIO19
IR5	GPIO26
VCC	5V
GND	GND



Step 2: Adjust the sensitivity of tracking sensor modules.

Turn on and hold the car and adjust the potentiometer on the tracking sensor with Philips screwdriver until you get the best sensitivity status: the signal indicate LED light will turn on when sensor is above black track, and the signal LED will turn off when the sensor is above white ground.



SOFTWARE INSTALLATION

Download the python code by typing following command in your Raspberry Pi terminal:

*wget http://osoyoo.com/driver/servo-steer-car/pi-*tracking.py

```
pi@raspberrypi:~ $ wget http://osoyoo.com/driver/servo-steer-car/pi-tracking.py
--2021-05-17 10:08:04--  http://osoyoo.com/driver/servo-steer-car/pi-tracking.py
Resolving osoyoo.com (osoyoo.com)... 51.79.21.80
Connecting to osoyoo.com (osoyoo.com)|51.79.21.80|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 4690 (4.6K) [text/x-python]
Saving to: 'pi-tracking.py'

pi-tracking.py      100%[===================>]   4.58K  --.-KB/s    in 0.001s

2021-05-17 10:08:05 (3.72 MB/s) - 'pi-tracking.py' saved [4690/4690]
```

Turn on the battery of your car and put the car on the black track line:

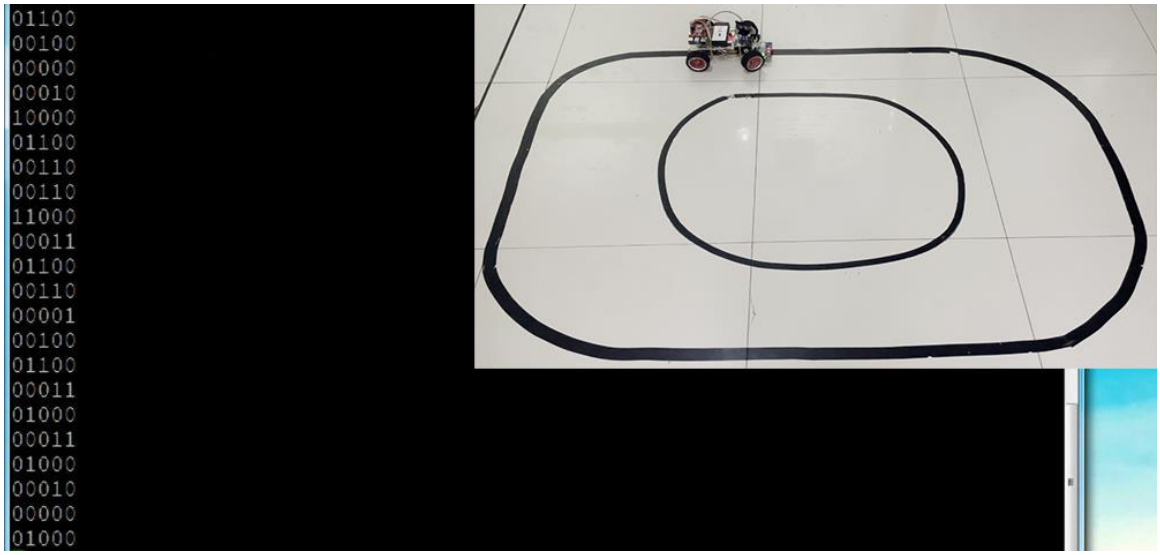
Now you can run the line tracking python program by typing following command:

python pi-tracking.py

```
pi@raspberrypi:~ $ python pi-tracking.py
```

```
pi@raspberrypi: ~
00110
00110
00110
00110
00110
00110
00110
00110
00110
00110
00110
00110
00110
00110
00110
00110
00110
00110
00110
00110
00110
00110
00110
00110
00110
00110
00110
00110
00110
00110
00110
```

Your car will move along the black track line.


01100
00100
00000
00010
10000
01100
00110
00110
11000
00011
01100
00110
00001
00100
01100
00011
01000
00011
01000
00010
00000
01000

OSOYOO Servo Steer Smart Car for Raspberry Pi Lesson 4: Obstacle Avoidance

OBJECTIVE

In this lesson, our Raspberry Pi robot car will use Ultrasonic sensor to detect obstacles and make automatic driving and avoid collision.

PARTS & DEVICES

No.	Picture	Device	Qty.	Accessories	Link
1		Ultrosonic Sensor	1	M1.4*8 Screw x 4 M1.4 Nut x 4	Click here to buy
2		Servo Motor	1	M2.2*8 Self Tapping Screw x 2 M2*4 Self Tapping Screw x 1	Click here to buy
3		Mount Holder for Ultrasonic Sensor	1	M1.4*8 Screw x 4 M1.4 Nut x 4 M2*4 Self Tapping Screw x 1	Click here to buy
4		20Pin jumper wire female to female 20cm	some		

5		Philips screwdriver	1		Click here to buy

HARDWARE INSTALLATION

Step 1: You must complete Lesson1 and Lesson 2

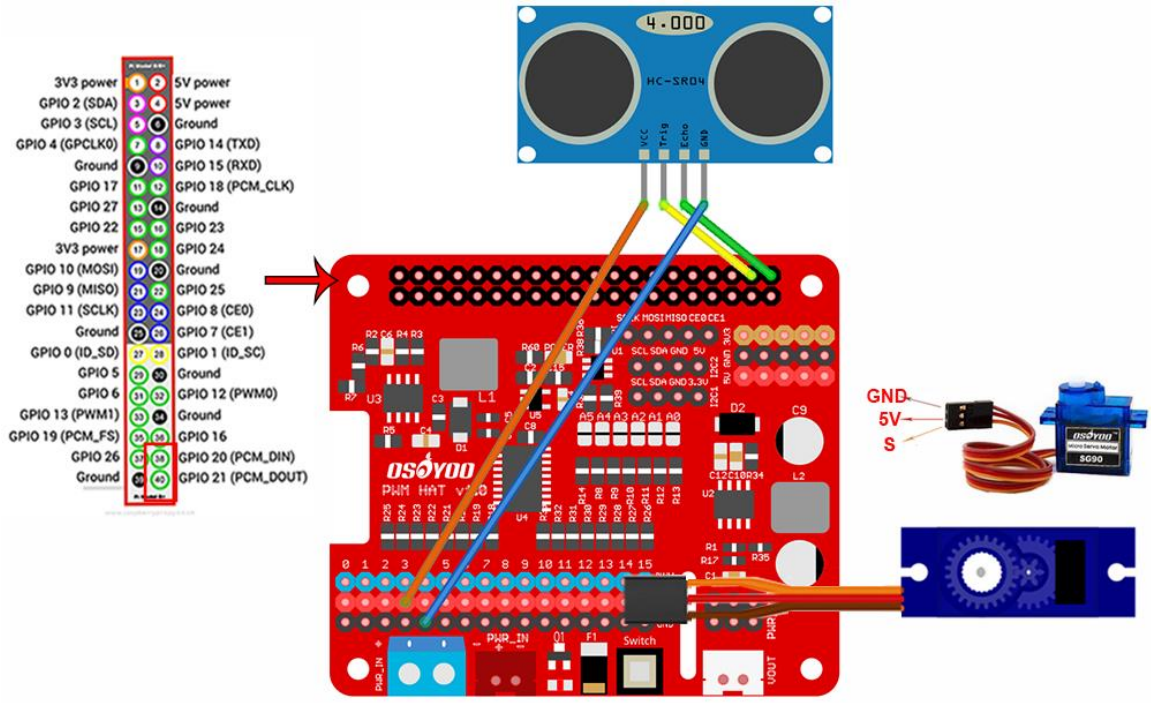
Step 2: Install servo motor at the front of upper car chassis with 2pcs M2.2*8 self tapping screws.

Step 3: Install ultrasonic module to mount holder with 4pcs M1.4*8 screw and M1.4 nuts.

Step 4: Install ultrasonic holder on micro servo motor with 1pcs M2*4 self-tapping screw.

CIRCUIT CONNECTION

Connect ultrasonic sensor holder SG90 servo motor to **PWM 14** port of OSOYOO PWM HAT board, and connect GND and VCC of ultrasonic module to GND and 5V of OSOYOO PWM HAT board, then connect TRIG and ECHO of ultrasonic module to GPIO20 and GPIO21 of raspberry pi as following graph (Remember: DO NOT remove any existing wires installed in Lesson 1):



Ultrasonic sensor	VCC	Trig	Echo	GND
OSOYOO PWM HAT V1.0	5V	GPIO20	GPIO21	GND

Micro Servo Motor	GND	VCC	S
OSOYOO PWM hat v1.0	GND	5V	PWM14

SOFTWARE INSTALLATION

Download the python code by typing following command in your Raspberry Pi terminal:

wget http://osoyoo.com/driver/*servo-steer-car*/pi-obstacle.py

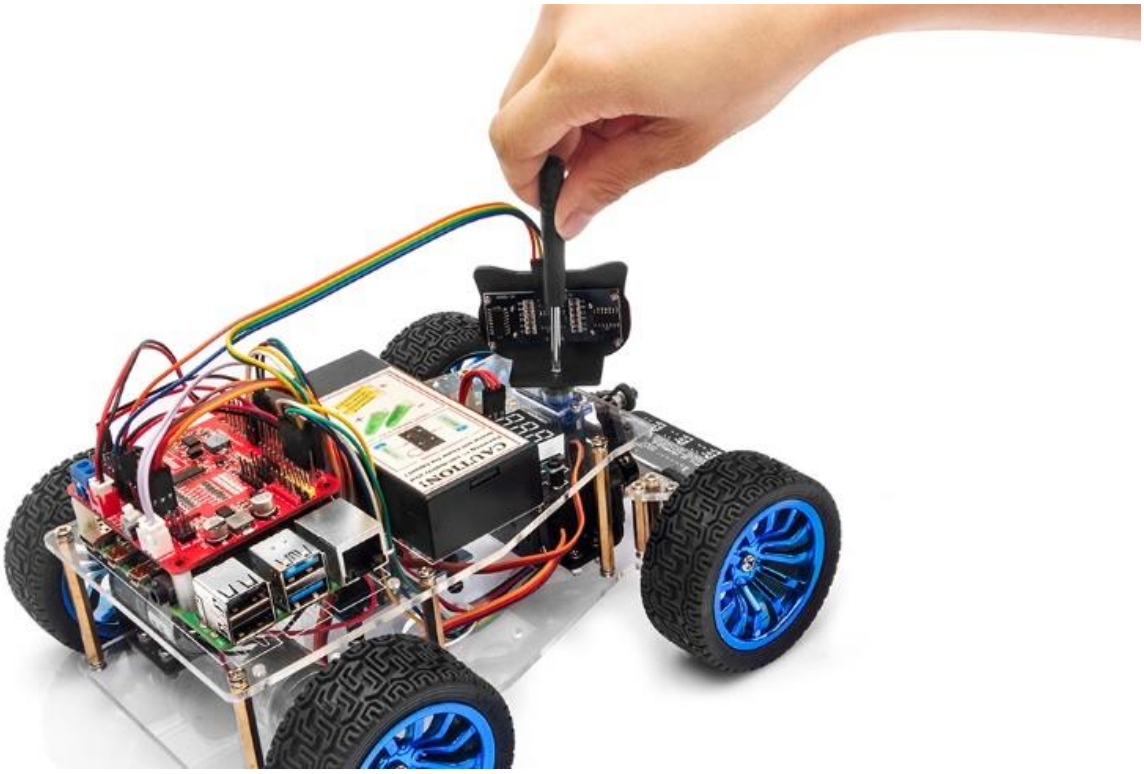
```
pi@raspberrypi:~ $ wget http://osoyoo.com/driver/servo-steer-car/pi-obstacle.py
--2021-05-19 09:42:31--  http://osoyoo.com/driver/servo-steer-car/pi-obstacle.py
Resolving osoyoo.com (osoyoo.com)... 51.79.21.80
Connecting to osoyoo.com (osoyoo.com)|51.79.21.80|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 7989 (7.8K) [text/x-python]
Saving to: 'pi-obstacle.py'

pi-obstacle.py      100%[===================>]   7.80K  --.-KB/s    in 0.001s

2021-05-19 09:42:32 (7.99 MB/s) - 'pi-obstacle.py' saved [7989/7989]
```

Turn on the battery of your car and your servo and run the program, If you are using Python 2 in Raspberry Pi, type:

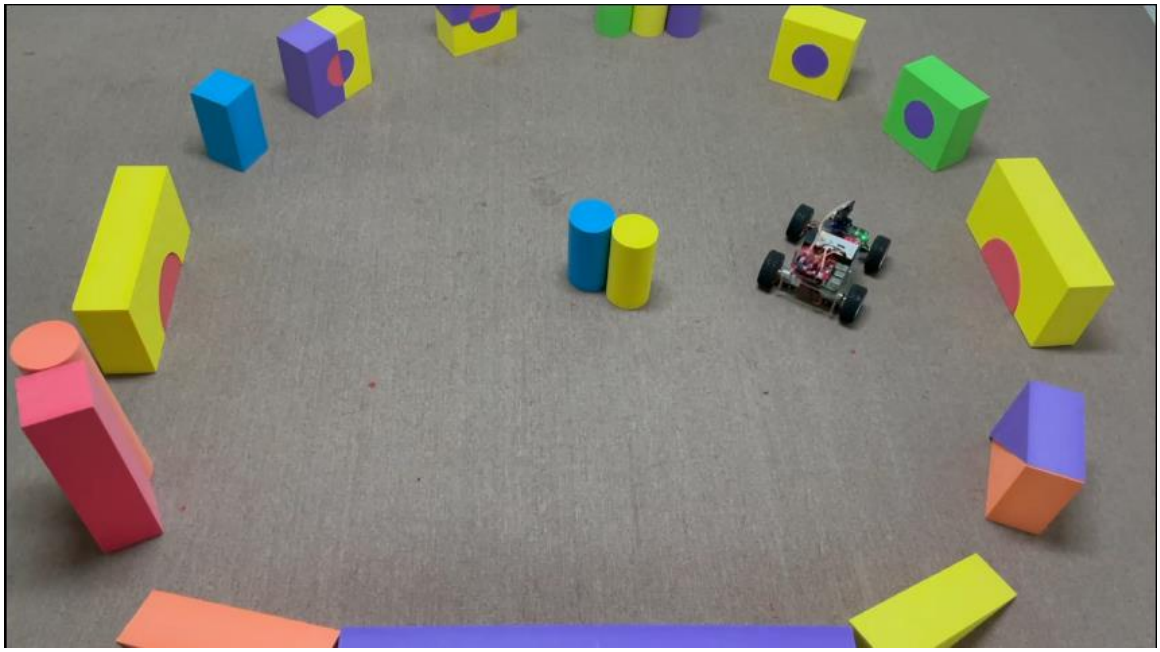
python pi-obstacle.py

```
pi@raspberrypi:~ $ python pi-obstacle.py
Moving servo on channel 0, press Ctrl-C to quit...
00001 turn left
11000 back to left
00011 back to right
00000 forward
00000 forward
11000 back to left
```

Your servo will rotate the ultrasonic sensor to front position for 3 seconds. If your sensor is not facing front direction, please turn off the battery or press Ctrl-C key to stop the program. Then remove the sensor from servo and re-install it, make sure it faces front and fix the position with screw, now you can type same command *python pi-obstacle.py* and run the program again.



Now the car will automatically make obstacle avoidance auto driving.



OSOYOO Servo Steer Smart Car for Raspberry Pi Lesson 5: Wifi UDB control by phone

OBJECTIVE

In this lesson, we will teach you how to use mobile APP to control Robot car through UDP protocol. The Raspberry Pi will run a Python program to get UDP packet from APP.

PARTS & DEVICES

No.	Picture	Device	Qty.	Accessories	Link
1		Ultrosonic Sensor	1	M1.4*8 Screw x 4 M1.4 Nut x 4	Click here to buy
2		Servo Motor	1	M2.2*8 Self Tapping Screw x 2 M2*4 Self Tapping Screw x 1	Click here to buy
3		Mount Holder for Ultrasonic Sensor	1	M1.4*8 Screw x 4 M1.4 Nut x 4 M2*4 Self Tapping Screw x 1	Click here to buy
4		Tracking sensor module	1	M2.5 Plastic Screw x 2 M2.5 Plastic Nut x2 M2.5 Plastic Pillar x 2	Click here to buy

5		7pin 25cm Female to Female Cable	1		Click here to buy
6		20Pin jumper wire female to female 20cm	some		
7		Philips screwdriver	1		Click here to buy

HARDWARE INSTALLATION

Must Install Lesson 3 (Line Tracking) and Lesson 4 (Obstacle Avoidance) First.

CIRCUIT CONNECTION

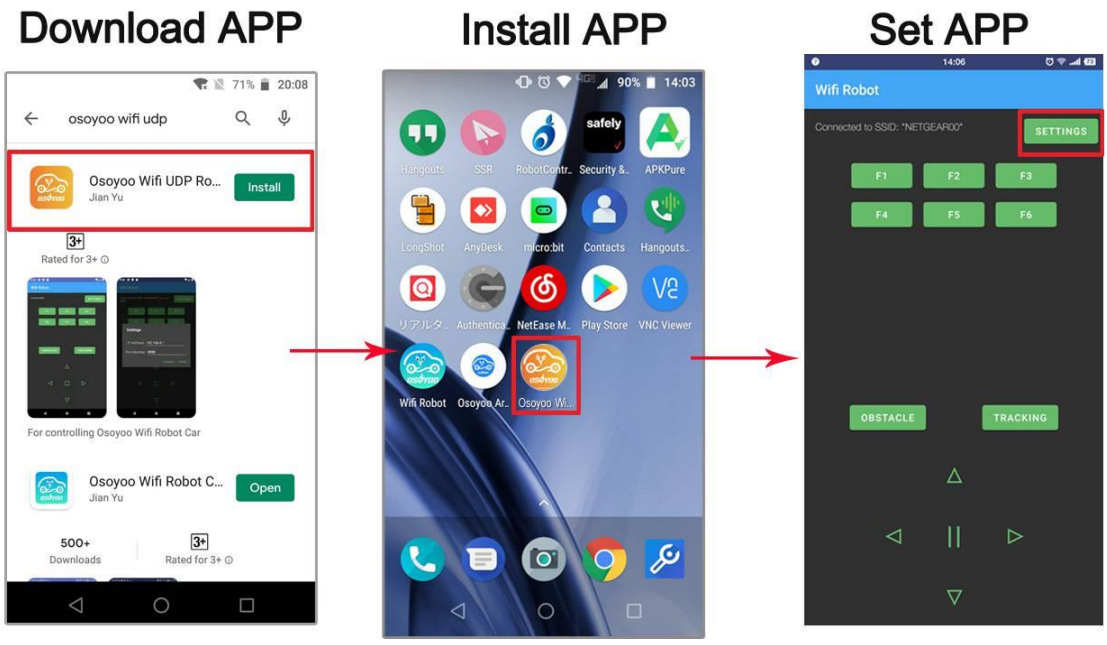
Remember: Keeping all existing wires installed in Lesson 1 -lesson4.

SOFTWARE INSTALLATION

Osoyoo Wifi Robot APP	OSOYOO	search “Osoyoo Wifi UDP Robot APP” in Google Play or Apple Store

Step 1) Download OSOYOO Wifi UDP Robot Car control APP

In Google Play or Apple Store, please search key words “OSOYOO Wifi UDP Robot Car”, you will find an orange icon APP as following:



Step 2) Make sure you have installed rpi.gpio and adafruit-pca9685 library in lesson 1 .

Step 3) Type following command to download the sample code:

wget http://osoyoo.com/driver/servo-steer-car/picar-udp.py

```
pi@raspberrypi:~ $ wget http://osoyoo.com/driver/servo-steer-car/picar-udp.py
--2021-05-26 08:11:36--  http://osoyoo.com/driver/servo-steer-car/picar-udp.py
Resolving osoyoo.com (osoyoo.com)... 51.79.21.80
Connecting to osoyoo.com (osoyoo.com)|51.79.21.80|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 10054 (9.8K) [text/x-python]
Saving to: 'picar-udp.py'

picar-udp.py        100%[===================>]   9.82K  --.-KB/s    in 0.04s

2021-05-26 08:11:37 (244 KB/s) - 'picar-udp.py' saved [10054/10054]
```

Step 4) Type following command to run the sample code:

python picar-udp.py

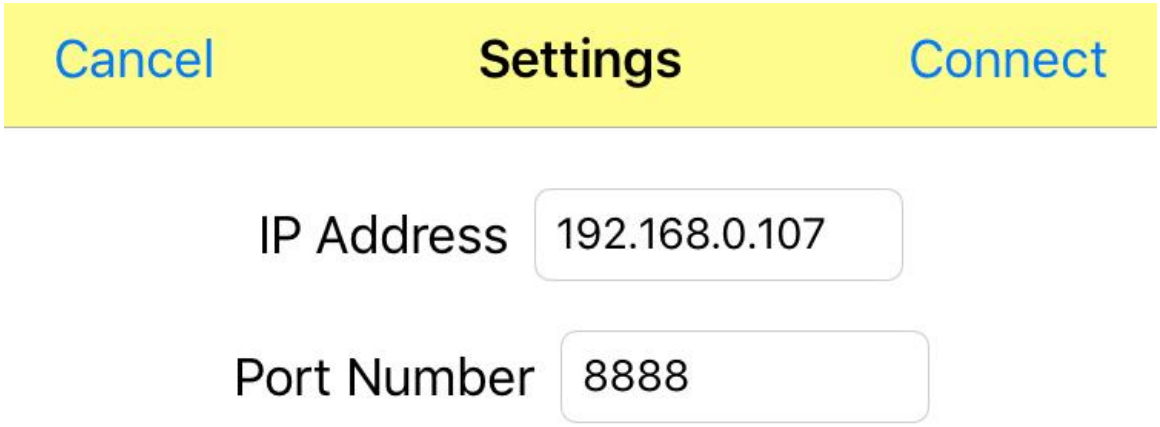
```
pi@raspberrypi:~ $ python picar-udp.py
Moving servo on channel 0, press Ctrl-C to quit...
```

or if you want to use Python3, typing:

python3 picar-udp.py

After above python is running, your car is waiting for command from your cell phone.

Step 5) Connect your phone with the same router wifi SSID your raspberry pi use. Open the APP, click Settings, and enter your **Raspberry Pi IP address** and **Port** to **8888** in settings:



Now you can click the < > ^ v direction keys to make the car move. Use || pause key to stop the car movement.

If you click Obstacle key, the car will do obstacle avoidance auto driving similar to Lesson 4

If you click Tracking key, the car will do link tracking auto driving similar to lesson 3

Note: F1~F6 are further development functions in the future.

FAQ about the Wifi UDP APP and sketch Code:

Q 1) How to tune the robot car speed?
A: If you want change the speed performance of the robot car, please change values following parameters in line 19-21 in picar-udp-control.py file :

```
high_speed = 3500  # Max pulse length out of 4096
mid_speed = 1900  # Max pulse length out of 4096
low_speed = 1700  # Max pulse length out of 4096
```

Q 2) What happened when you press buttons in OSOYOO WiFi UDP Robot Car APP?
A: When you press a button of the APP, APP will send a single-letter message through UDP protocol to target device Raspberry Pi

Button	UDP message
F1	F
F2	G
F3	H
F4	I
F5	J
F6	K
▲	A
▼	B
▸	R
◂	L
square	E
obstacle	O
tracking	T

Q 3) How does Raspberry Pi python program handle the UDP command?
Line 290 to 318 *while loop* receives UDP data from APP and give it to viable **cur_status**, ticker function in line 266 – 280 handle the **cur_status** :

```
def ticker():
    if cur_status=='R':
        turnRight(high_speed,0)
    if cur_status=='L':
        turnLeft(0,high_speed)
    if cur_status=='A':
        forward(mid_speed,mid_speed)
    if cur_status=='B':
        backward(mid_speed,mid_speed)
    if cur_status=='E':
        stopcar()
    if cur_status=='T':
```

```
        line_tracking()
    if cur_status=='O':
        obstacle_avoid()
```

For example , when APP ▲key is pressed , **cur_status** value is A , then ticker() function call *forward(mid_speed,mid_speed)* function to make car moving forward.

OSOYOO Servo Steer Smart Car for Raspberry Pi Lesson 6: Web Camera Control

OBJECTIVE

In this lesson, we will show you how to use Python3 Flask and M-Jpeg Streamer software to control a Raspberry Pi Robot Car through Internet. You will monitor the car's real-time movement through its' eye (front camera).

PARTS & DEVICES

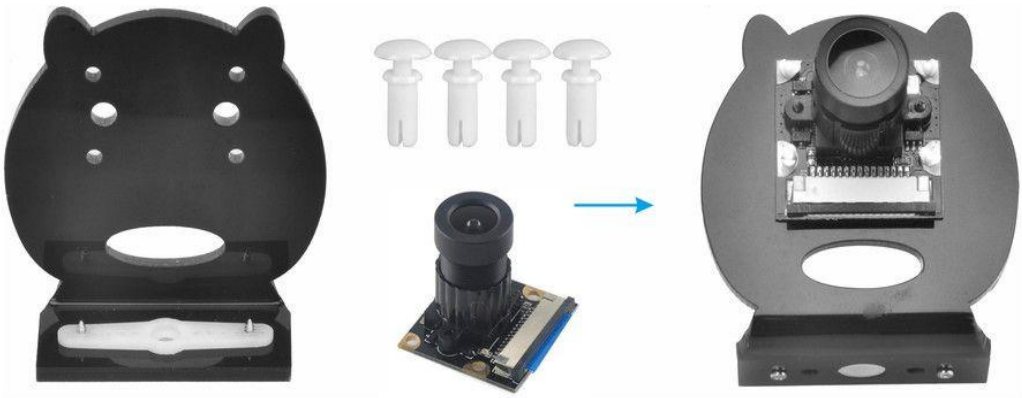
No.	Picture	Device	Qty.	Accessories	Link
1		CSI camera	1	M2 push pin rivets x 4	Click here to buy
2		Servo Motor	1	M2.2*8 Self Tapping Screw x 2 M2*4 Self Tapping Screw x 1	Click here to buy
3		Mount Holder for CSI camera	1	M2 push pin rivets x 4 M1.5*6 Self Tapping Screws x2 M2*4 Self Tapping Screw x 1	click here to buy
4		Philips screwdriver	1		Click here to buy

HARDWARE INSTALLATION

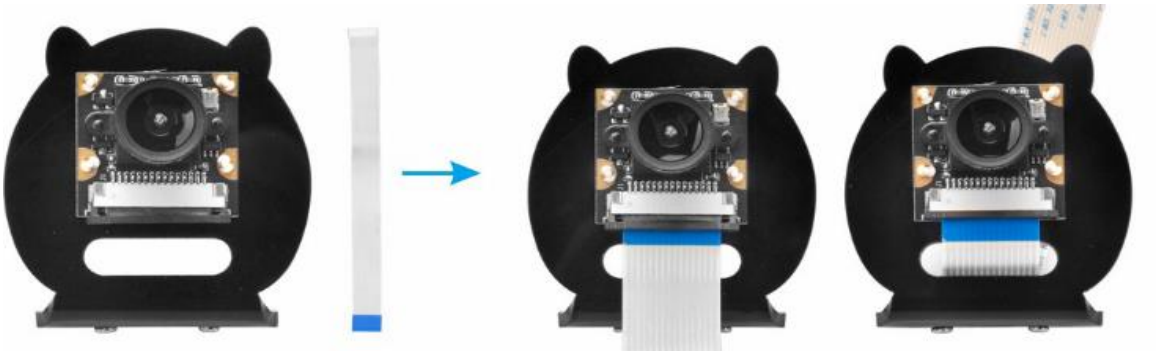
Step 1: You must complete and test lesson 2 before you continue on with this lesson, then install a new SG90 blue servo onto the servo position. Connect SG90 servo to OSOYOO PWM HAT board port 14.

Step 2: Choose slotted bracket in servo motor to cross Camera holder from top to bottom and fix blade with M1.5*6 Self Tapping Screws

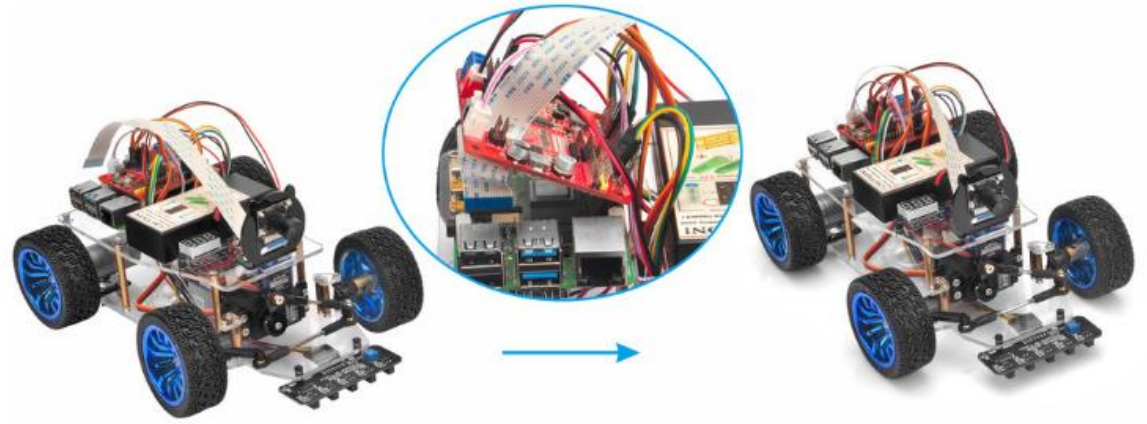
Step 3: Install CSI camera to holder with 4pcs M2 push pin rivets.



Step 4: Connect CSI camera with CSI ribbon cable (Please pay attention the connections of the cable before you install it.)

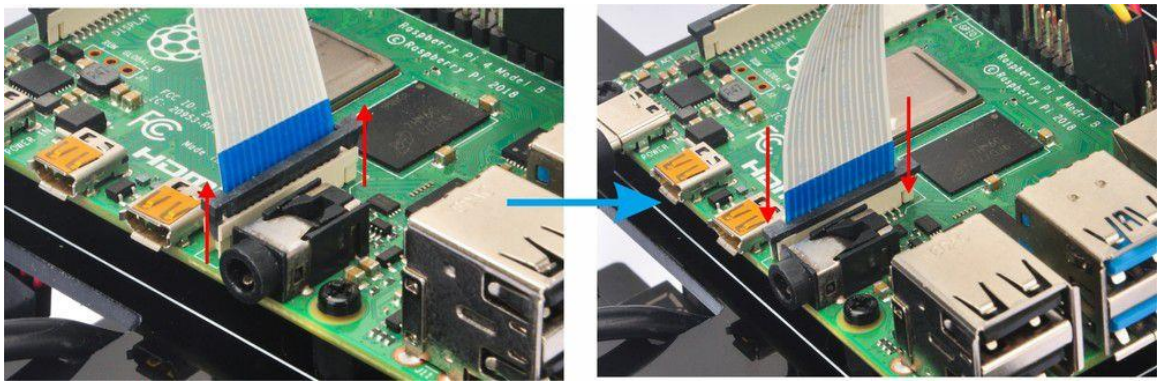


Step 5: Install Camera holder on servo motor with 1pc M2*4 self-tapping screws.



CIRCUIT CONNECTION

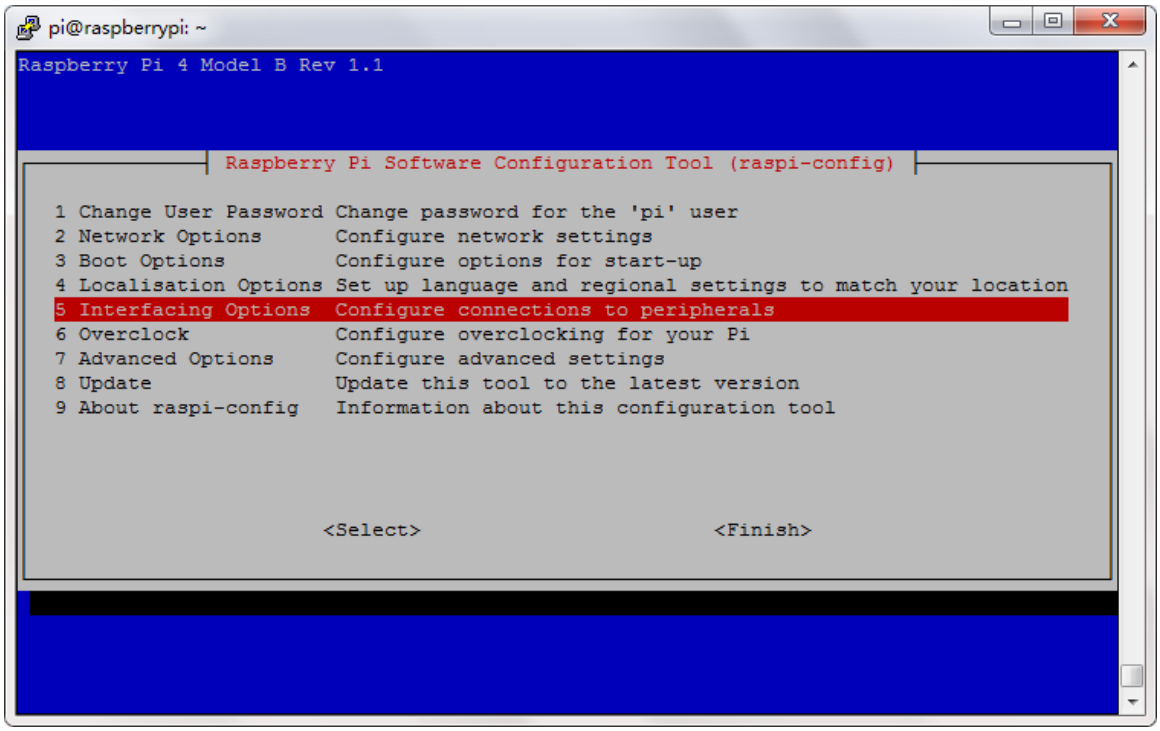
Servo must be installed and connected OSOYOO PWM HAT board 15 port. Connect CSI camera to CSI Slot of Raspberry Pi with CSI ribbon cable (Please pay attention the connections of the cable before you install it.)

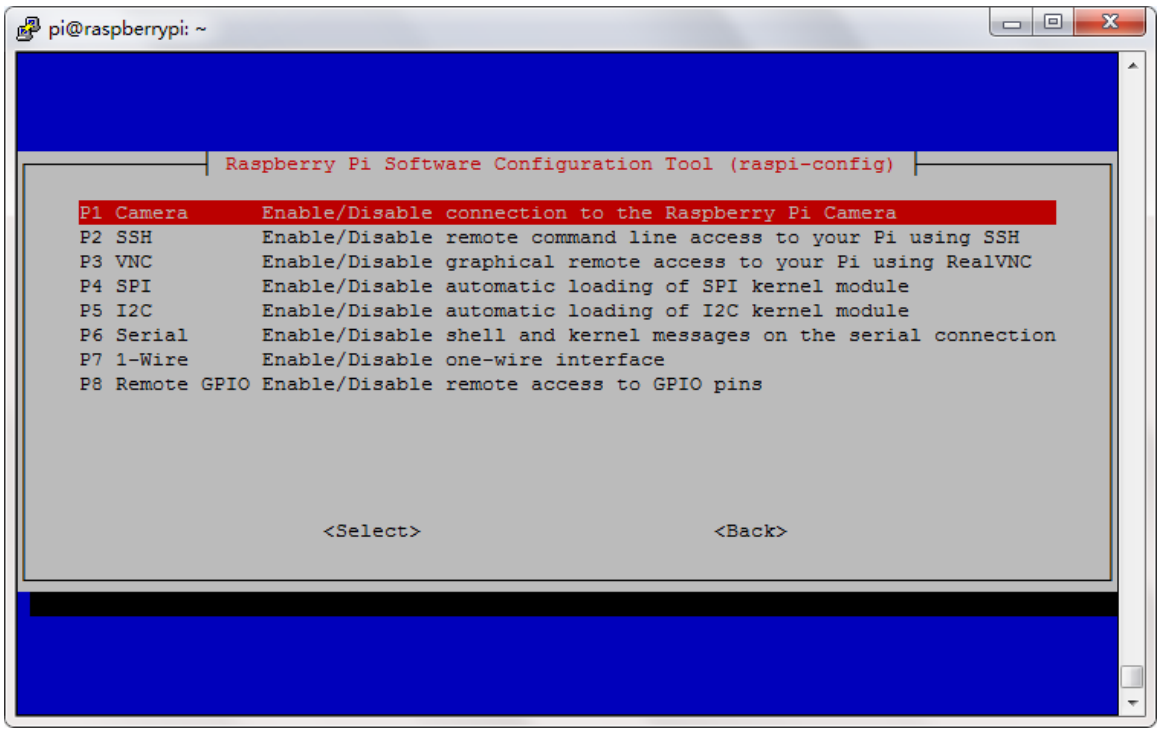


Step 1) please enable Camera in Raspberry Pi by typing following command

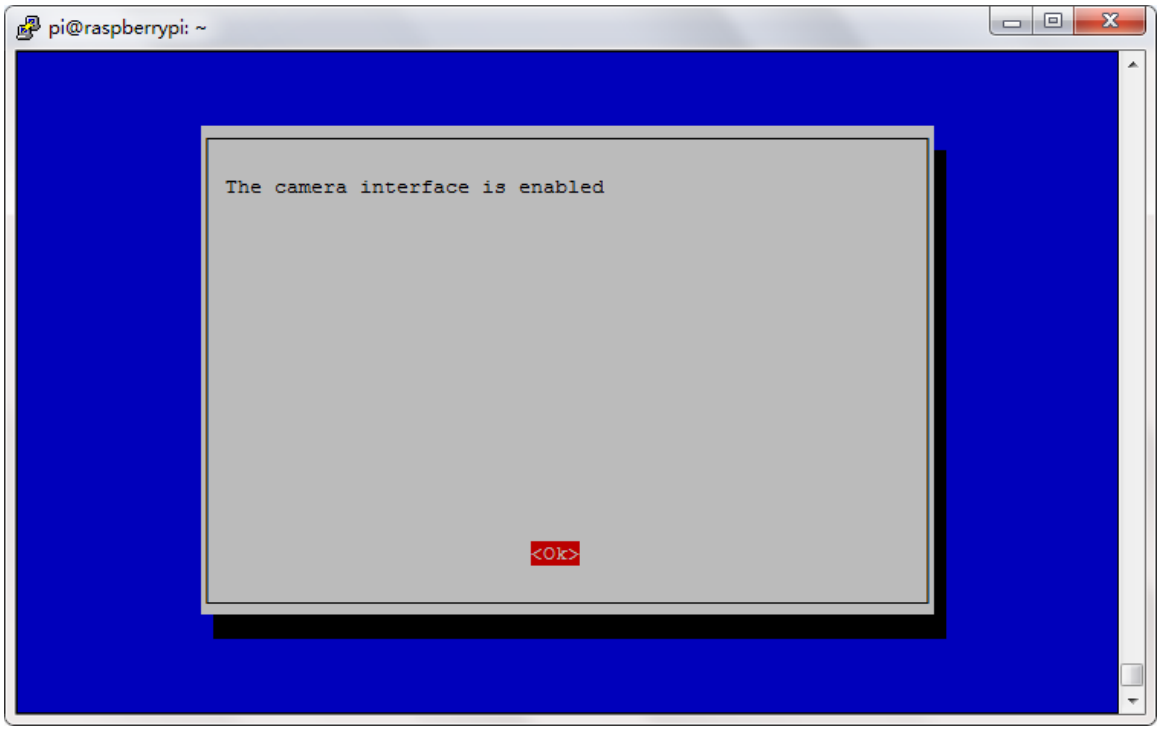
sudo raspi-config

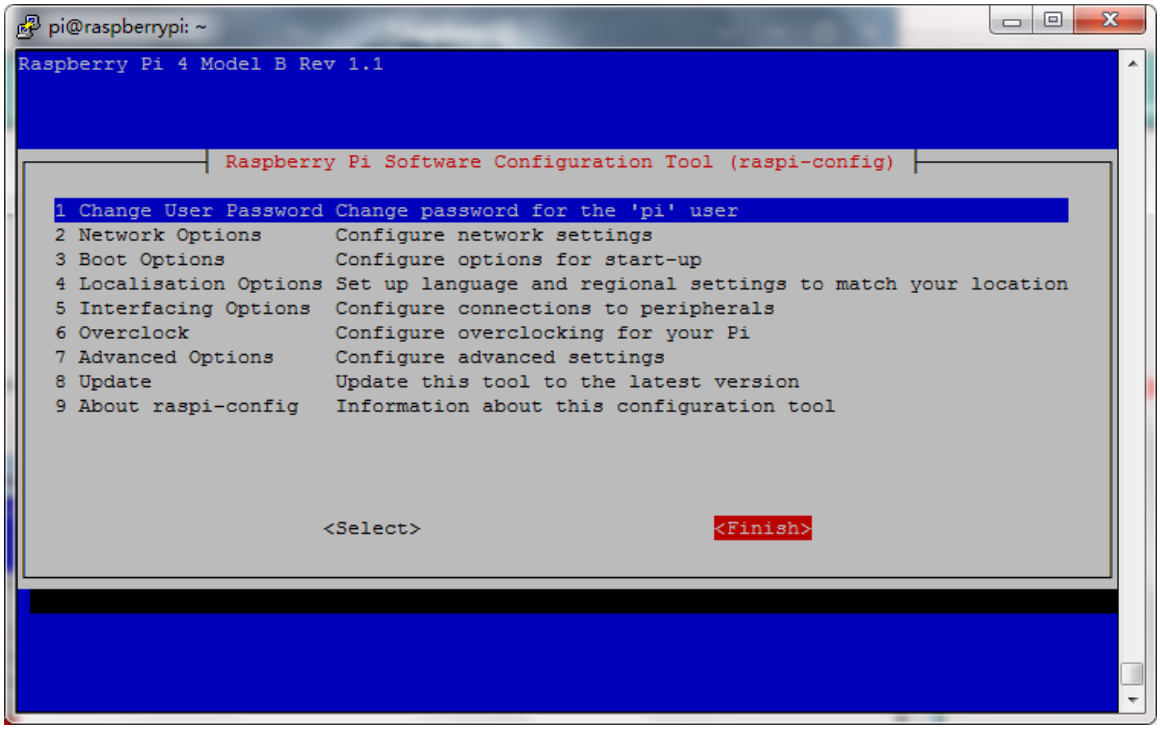
Then select ->5 Interfacing Options->P1 Camera->Yes->Ok->Finish (Please reboot the raspberry pi according to the notice.)






pi@raspberrypi: ~
Would you like the camera interface to be enabled?
<Yes>
<No>


pi@raspberrypi: ~
The camera interface is enabled
<Ok>



Step 2: Type following command to install mjpeg-streamer software:

wget http://osoyoo.com/driver/servo-steer-car/webcam.sh

```
pi@raspberrypi:~ $ wget http://osoyoo.com/driver/servo-steer-car/webcam.sh
--2021-05-18 04:52:15--  http://osoyoo.com/driver/servo-steer-car/webcam.sh
Resolving osoyoo.com (osoyoo.com)... 51.79.21.80
Connecting to osoyoo.com (osoyoo.com)|51.79.21.80|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 634 [text/x-sh]
Saving to: 'webcam.sh'

webcam.sh           100%[===================>]     634  --.-KB/s    in 0s

2021-05-18 04:52:15 (20.8 MB/s) - 'webcam.sh' saved [634/634]
```

And then type the following command:

wget http://osoyoo.com/driver/servo-steer-car/camstart.sh

```
pi@raspberrypi:~ $ wget http://osoyoo.com/driver/servo-steer-car/camstart.sh
--2021-05-18 04:53:51--  http://osoyoo.com/driver/servo-steer-car/camstart.sh
Resolving osoyoo.com (osoyoo.com)... 51.79.21.80
Connecting to osoyoo.com (osoyoo.com)|51.79.21.80|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 46 [text/x-sh]
Saving to: 'camstart.sh'

camstart.sh         100%[===================>]      46  --.-KB/s    in 0s

2021-05-18 04:53:51 (1.79 MB/s) - 'camstart.sh' saved [46/46]
```

And type the following command at last:

bash webcam.sh

```
pi@raspberrypi:~ $ bash webcam.sh
--2021-05-19 08:07:00--  http://osoyoo.com/driver/mjpg-streamer.tar.gz
Resolving osoyoo.com (osoyoo.com)... 51.79.21.80
Connecting to osoyoo.com (osoyoo.com)|51.79.21.80|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 558327 (545K) [application/x-gzip]
Saving to: 'mjpg-streamer.tar.gz.9'

mjpg-streamer.tar.g 100%[===================>] 545.24K   361KB/s    in 1.5s

2021-05-19 08:07:03 (361 KB/s) - 'mjpg-streamer.tar.gz.9' saved [558327/558327]
```

After running above commands, mjpeg-streamer software is installed in your raspberry pi. So Simple!

```
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  libapr1 libaprutil1 libserf-1-1 libsvn1 libutf8proc2
Suggested packages:
  db5.3-util libapache2-mod-svn subversion-tools
The following NEW packages will be installed:
  libapr1 libaprutil1 libserf-1-1 libsvn1 libutf8proc2 subversion
0 upgraded, 6 newly installed, 0 to remove and 0 not upgraded.
Need to get 2,159 kB/2,422 kB of archives.
After this operation, 9,064 kB of additional disk space will be used.
Err:1 http://raspbian.raspberrypi.org/raspbian buster/main armhf libsvn1 armhf 1
.10.4-1+deb10u1
  404  Not Found [IP: 93.93.128.193 80]
Err:2 http://raspbian.raspberrypi.org/raspbian buster/main armhf subversion armh
f 1.10.4-1+deb10u1
  404  Not Found [IP: 93.93.128.193 80]
E: Failed to fetch http://raspbian.raspberrypi.org/raspbian/pool/main/s/subversi
on/libsvn1_1.10.4-1+deb10u1_armhf.deb  404  Not Found [IP: 93.93.128.193 80]
E: Failed to fetch http://raspbian.raspberrypi.org/raspbian/pool/main/s/subversi
on/subversion_1.10.4-1+deb10u1_armhf.deb  404  Not Found [IP: 93.93.128.193 80]
E: Unable to fetch some archives, maybe run apt-get update or try with --fix-mis
sing?
--2021-05-19 08:07:08--  http://osoyoo.com/driver/statx.h
Resolving osoyoo.com (osoyoo.com)... 51.79.21.80
Connecting to osoyoo.com (osoyoo.com)|51.79.21.80|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1991 (1.9K) [text/x-chdr]
Saving to: 'statx.h'

statx.h             100%[===================>]   1.94K  --.-KB/s    in 0s

2021-05-19 08:07:09 (51.2 MB/s) - 'statx.h' saved [1991/1991]
```

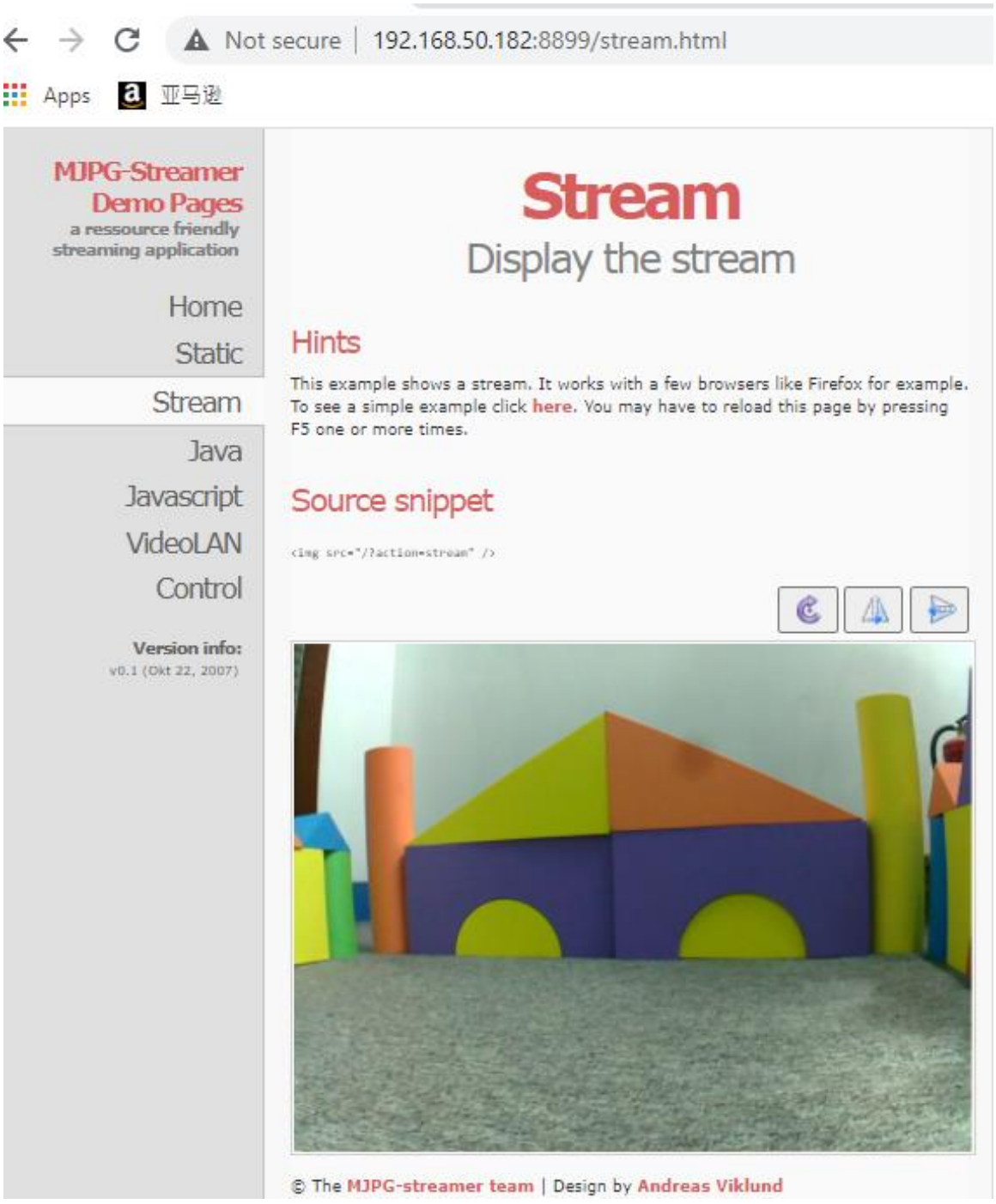
Step 3: Start jpeg-streamer server in your Raspberry Pi
Run following command in your Pi terminal will start your mjpeg-streamer server

bash camstart.sh

```
pi@raspberrypi:~ $ bash camstart.sh
MJPG Streamer Version: svn rev:
 i: Using V4L2 device.: /dev/video0
 i: Desired Resolution: 640 x 480
 i: Frames Per Second.: 30
 i: Format............: YUV
 i: JPEG Quality......: 80
 o: www-folder-path...: www/
 o: HTTP TCP port.....: 8899
 o: username:password.: disabled
 o: commands..........: enabled
```

These message means your video server is running at port 8899 in your Pi.

Step 4: Now we can use your browser to test the surveillance video. Now please visit http://your_raspberry_pi_ip:8899 (in our sample case http://192.168.0.34:8899).



Step 5: Open a new terminal window and run the following code to install the OSOYOO web camera controlled robot car software

wget http://osoyoo.com/driver/servo-steer-car/piwebcar.sh

```
pi@raspberrypi:~ $ wget http://osoyoo.com/driver/servo-steer-car/piwebcar.sh
--2021-06-01 09:08:41--  http://osoyoo.com/driver/servo-steer-car/piwebcar.sh
Resolving osoyoo.com (osoyoo.com)... 51.79.21.80
Connecting to osoyoo.com (osoyoo.com)|51.79.21.80|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 130 [text/x-sh]
Saving to: 'piwebcar.sh'

piwebcar.sh         100%[===================>]     130  --.-KB/s    in 0s

2021-06-01 09:08:41 (6.34 MB/s) - 'piwebcar.sh' saved [130/130]
```

and then type the following command:

bash piwebcar.sh

```
pi@raspberrypi:~ $ bash piwebcar.sh
Reading package lists... Done
Building dependency tree
Reading state information... Done
python3-flask is already the newest version (1.0.2-3).
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
--2021-05-24 08:58:09--  http://osoyoo.com/driver/p3-car/piwebcar.tar.gz
Resolving osoyoo.com (osoyoo.com)... 51.79.21.80
Connecting to osoyoo.com (osoyoo.com)|51.79.21.80|:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 360167 (352K) [application/x-gzip]
Saving to: 'piwebcar.tar.gz'

piwebcar.tar.gz     100%[===================>] 351.73K   168KB/s    in 2.1s

2021-05-24 08:58:12 (168 KB/s) - 'piwebcar.tar.gz' saved [360167/360167]

piwebcar/
piwebcar/webcar.py
piwebcar/templates/
piwebcar/templates/index.html
piwebcar/static/
piwebcar/static/style.css
piwebcar/static/images/
piwebcar/static/images/lightoff.png
piwebcar/static/images/lighton.png
piwebcar/static/images/robot52.png
piwebcar/static/images/forward.png
piwebcar/static/images/left.png
piwebcar/static/images/reverse.png
piwebcar/static/images/right.png
piwebcar/static/images/stop.png
pi@raspberrypi:~ $
```

Step 6: type following command to enter the folder osoyoowebcar

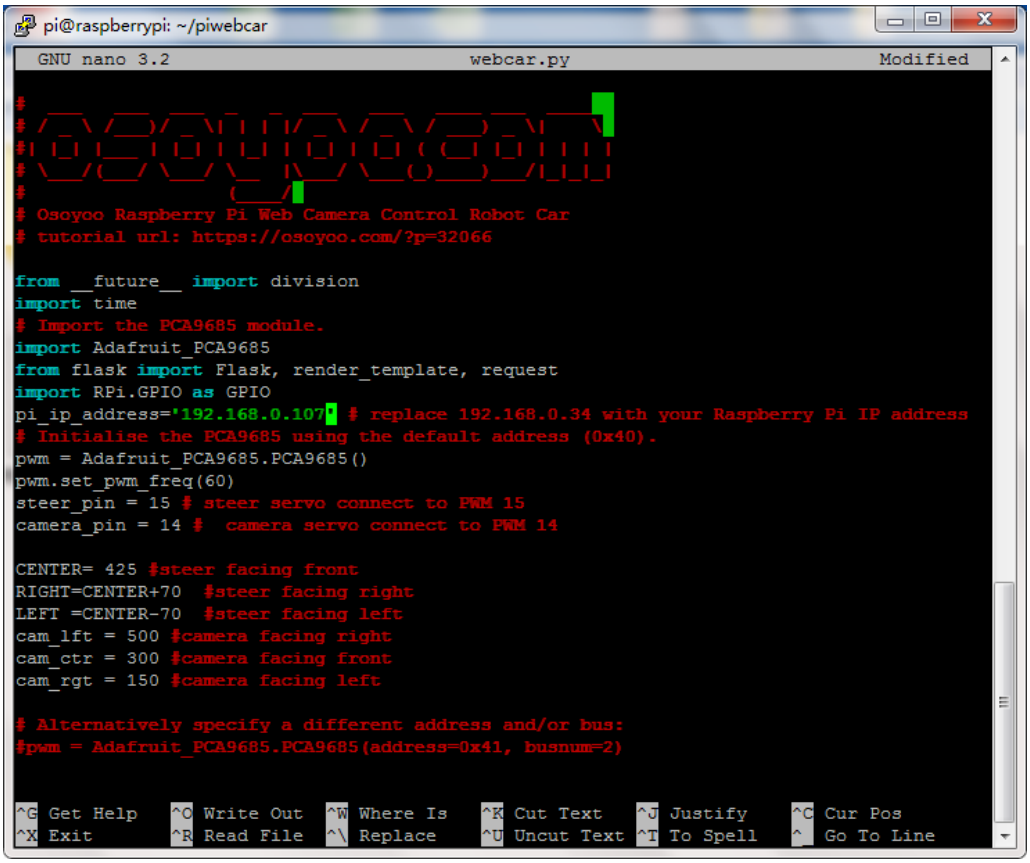
cd piwebcar

Step 7: Then type the following command to edit the file webcar.py

nano webcar.py

```
pi@raspberrypi:~ $ cd piwebcar
pi@raspberrypi:~/piwebcar $ nano webcar.py
```

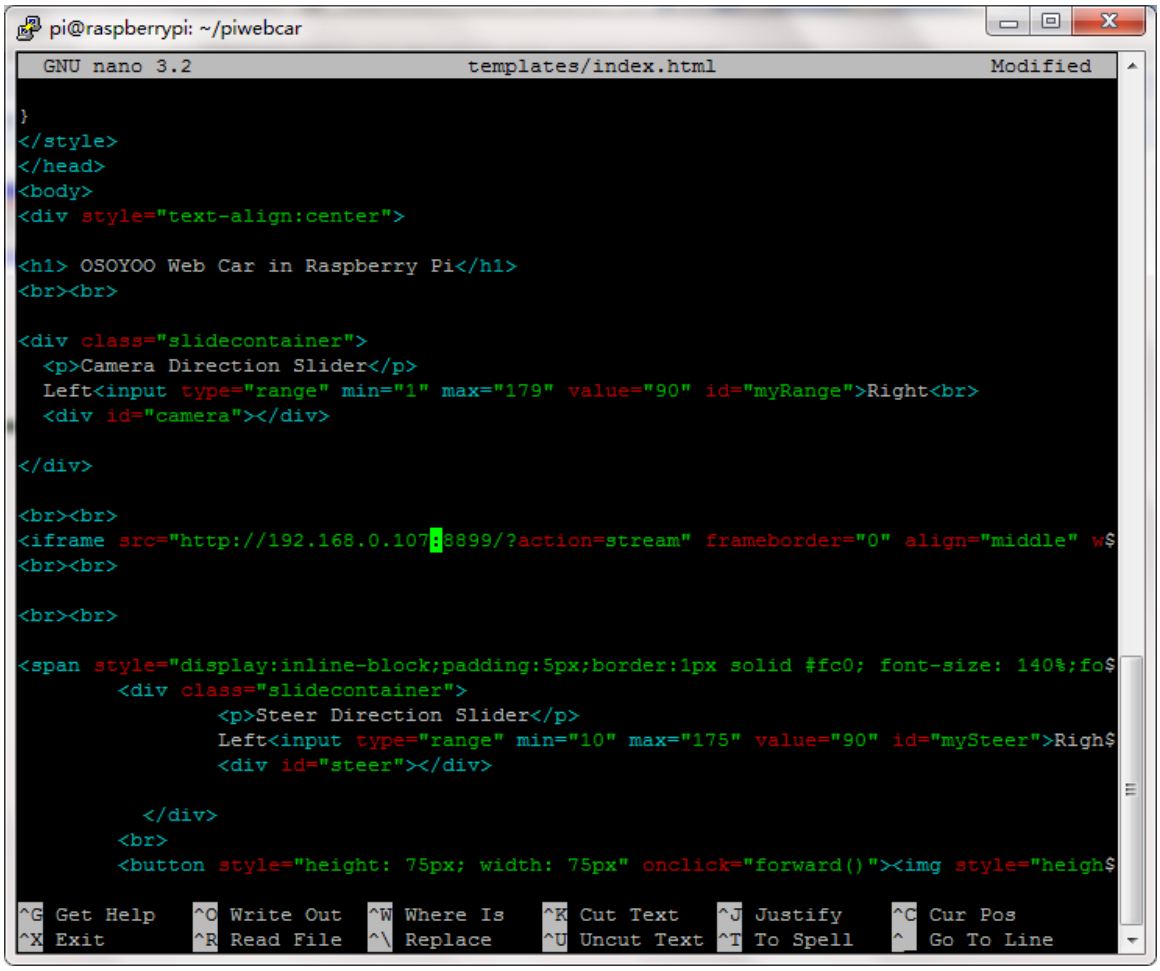
Replace ip address 192.168.0.34 in line 15 to your raspberry pi IP address, and click "Ctrl"+"X" then "Y" to save the file and then click "enter" to exit.



Step 8: Then type the following command to edit the file templates/index.html

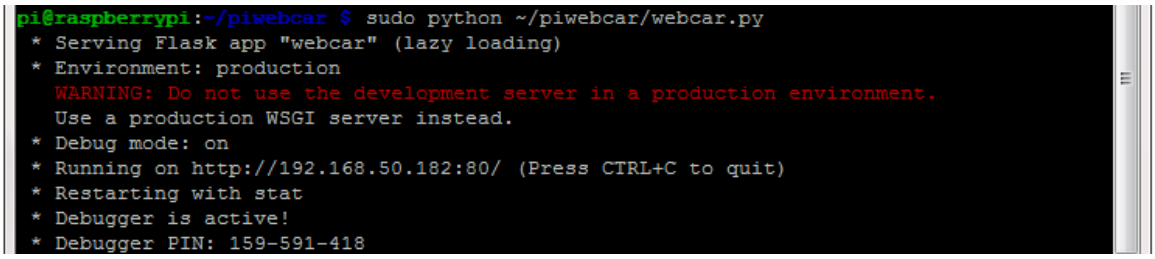
nano templates/index.html

Please change **192.168.0.34** in line **34** to your pi's ip address, and cick "Ctrl"+"X" then "Y" to save the file and then click "enter" to exit.

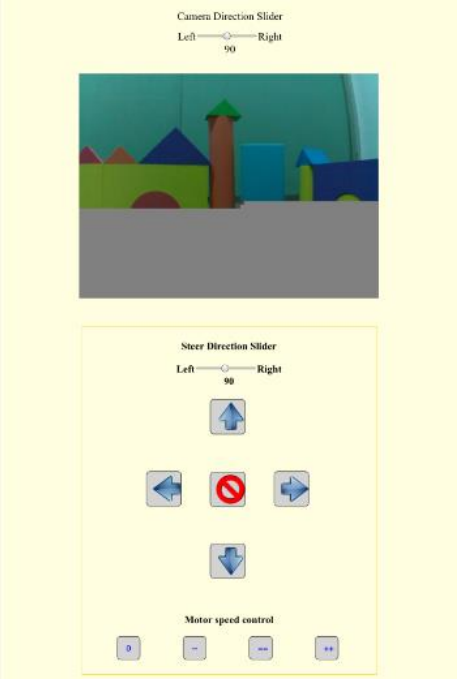


Step 9: Run the code by typing command

```
sudo python ~/piwebcar/webcar.py
```



Step 10: In your PC or cell phone which is the same wifi network of your Raspberry Pi, open the browser and visit http://your_RaspberryPi_ip_address, you will see following UI



The arrows buttons are direction control keys, red circle button in the middle is the STOP key. There are four speed control buttons in the bottom:

- 0 key means the slowest speed
- — key means the 2nd slow speed
- == key means regular speed
- ++ key means fastest speed

In the top of the page, there is a Camera Direction Slider, you can move the slider in order to rotate the servo and change the camera orientation.

Before the upper Arrow button, there is a Steer Direction Slider, you can move this slider to rotate your front wheel (just like rotate your steering wheel when driving a car).

Reference:

Marcelo Rovai : Python WebServer With Flask and Raspberry Pi https://towardsdatascience.com/python-webserver-with-flask-and-raspberry-pi-398423cc6f5d