Preface

About SunFounder

SunFounder is a company focused on STEAM education with products like open source robots and smart devices distributed globally. In SunFounder, we strive to help elementary and middle school students as well as hobbyists, through STEAM education, strengthen their hands-on practices and problem-solving abilities. In this way, we hope to disseminate knowledge and provide skill training in a full-of-joy way, thus fostering your interest in programming and making, and exposing you to a fascinating world of science and engineering. To embrace the future of artificial intelligence, it is urgent and meaningful to learn abundant STEAM knowledge.

About This Kit

PiCar-4WD is a cost-effectively developmental smart car kit of Raspberry Pi, suitable for the Raspberry Pi 3 model B/B+ and 4 Model B. With the grayscale module, ultrasonic module, Photo-interrupter sensor module, the car can realize obstacle avoidance, line following, object following, cliff detection, velocity measurement, mileage calculation and so on.

Besides, the kits comes with operational examples of Web terminal, a rookie-friendly design. We also provide the original python code for in-depth learning.

Quickly Support

Thanks for using the PiCar-4WD Smart Robot Car Kit for Raspberry Pi. You can contact us at the email address below. Please don't hesitate to let us know if you need any support from us. We love helping our customers to learn and play with our products.

If you have any **TECHNICAL question**, add a topic under **FORUM** section on our website and we'll reply as soon as possible.

For **NON-TECH questions** like order and shipment issues, please **send an email to service@sunfounder.com**. You're also welcomed to share your projects on FORUM.

Copyright Notice

All contents including but not limited to texts, images, and code in this manual are owned by the SunFounder Company. You should only use it for personal study, investigation, enjoyment, or other non-commercial or nonprofit purposes, under the related regulations and copyrights laws, without infringing the legal rights of the author and relevant right holders. For any individual or organization that uses these for commercial profit without permission, the Company reserves the right to take legal action.

V1.0.0 Mar. 2020

Warning

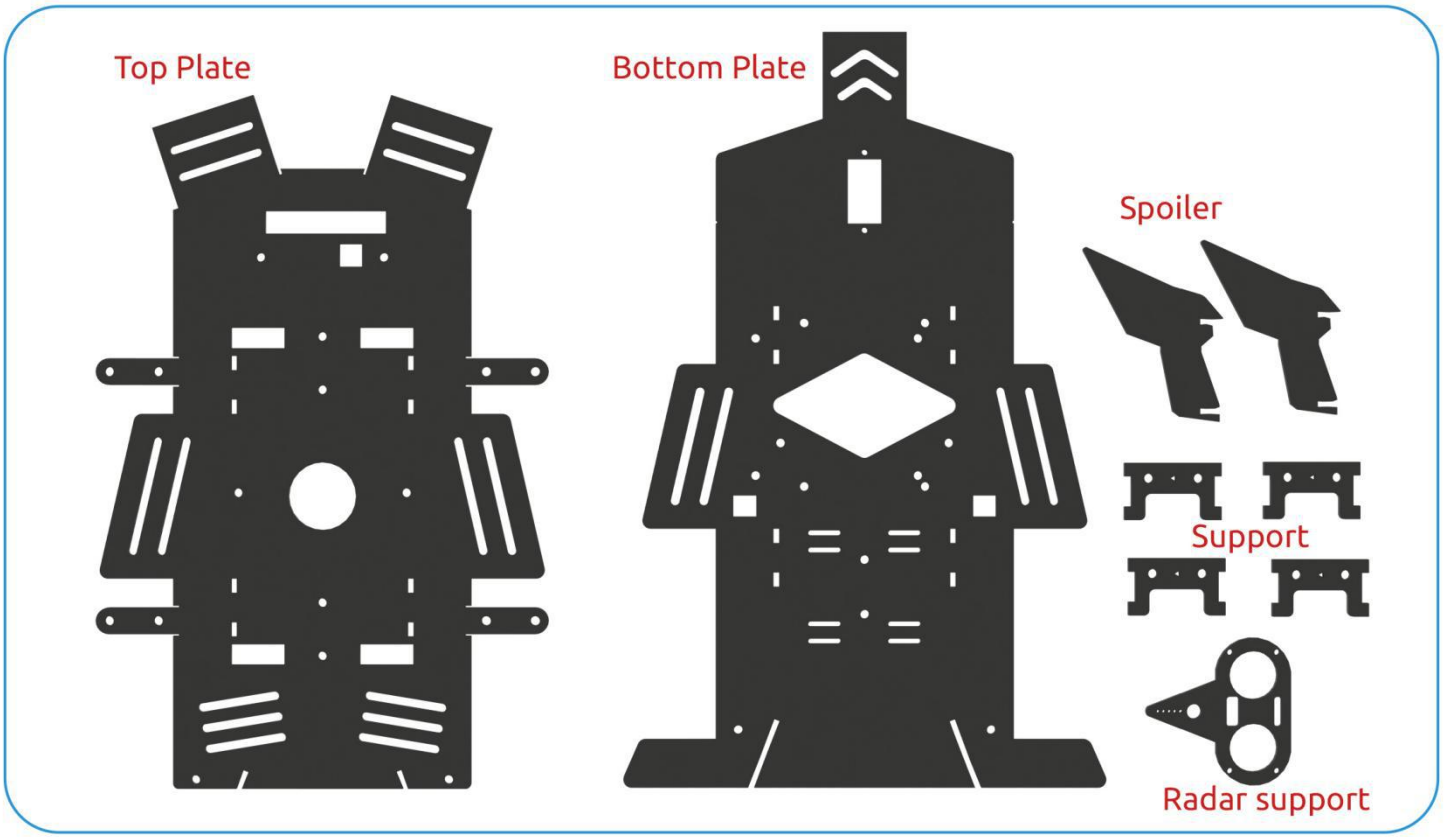
When you purchase or use this product, please note the following points:

- This product contains many small parts. Swallowing or improper operation may cause serious infections and death. Seek immediate medical attention once the accident happens.
- Strictly prohibit use this product and its parts near any AC electrical outlet or other circuits in case of the potential risk of electric shock.
- Strictly prohibit use this product near any liquid or fire. Keep conductive materials away from this product.
- Do not allow children under 3 years old to use this product without adult supervision. Please place this product in the position where children under 3 years old cannot reach.
- Do not store or use this product in any extreme environments such as extreme hot or cold, high humidity and etc.
- Remember to break the circuit when it is not needed.
- Some parts of this product may become warm to touch when used in certain circuit designs, which is normal.
- Using components not in accordance with the specification may cause damage to the product.

Contents

Component List

Structural Plates



Mechanical Parts

M1.5 x 4mm Self - tapping Screw
4 PCS

M1.4 x 8mm Screw
6 PCS

M2 x 10mm Screw
4 PCS

M2.5x6mm Screw
10 PCS

M3 x 8mm Flat-head Screw
4 PCS

M3 x 8mm Screw
16 PCS

M3 x 28mm Screw
10 PCS

M1.4 Nut
6 PCS

M2 Nut
4 PCS

M3 Nut
18 PCS

M2.5 x 11mm Bi-pass Nylon Standoff
6 PCS

Encoding Disk
2 PCS

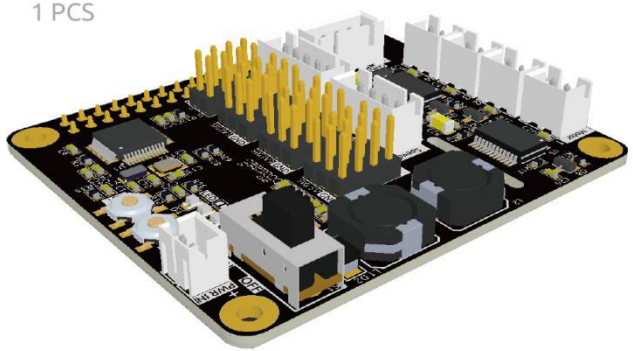
M2.5x8+6mm Single-pass Nylon Standoff
6 PCS

M3 x 30mm Bi-pass Nylon Standoff
6 PCS

Electronic Parts

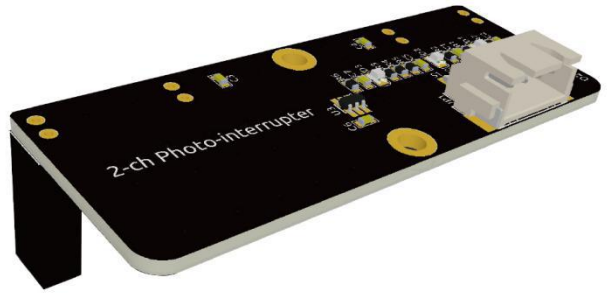
4WD Hat Module

1 PCS



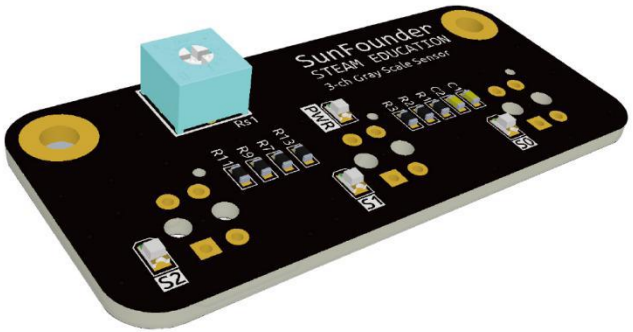
2-ch Photo-interrupter Module

1 PCS



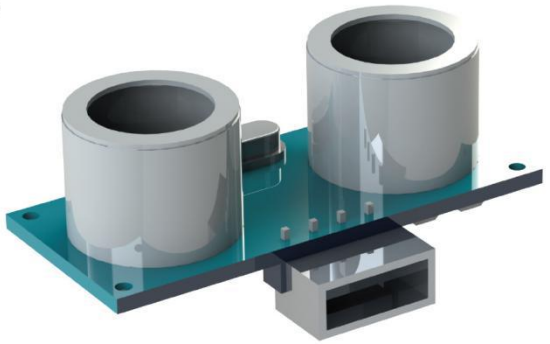
3-ch Grayscale Sensor Module

1 PCS



3.3V Ultrasonic Module

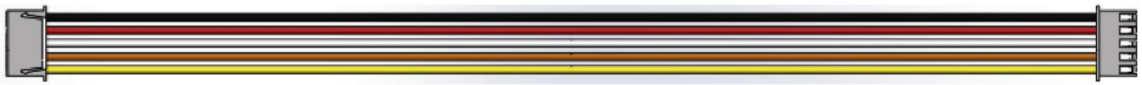
1 PCS



Other Parts

Connection Wire(Grayscale Sensor Module)

1 PCS



Connection Wire(Ultrasonic Module)

1 PCS

Connection Wire(Photo-interrupter Module)

1 PCS

Ribbon

1 PCS

Cable Tie

4 PCS

Screwdriver

1 PCS

Wheels

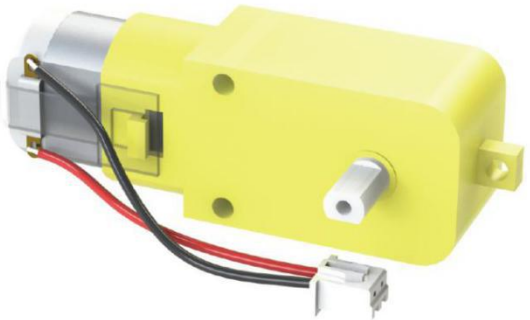
4 PCS

Battery Holder

1 PCS

Driving Parts

Motor
4 PCS



Servo
1 PCS

Self- prepared Part

18650 Batteries
2 PCS

Note: After opening the package, please check whether the quantity of components is compliance with product description and whether all components are in good condition.

Get Started with Raspberry Pi

In this chapter, configure your Raspberry Pi first.

For the Raspberry Pi OS (previously called Raspbian) system, You are recommend to install the version: **Raspberry Pi OS (32-bit) with desktop and recommended software.**

If you have installed the other two versions, you need to install the following softwares manually and then skip this chapter.

```
sudo apt-get update
sudo apt-get install git-core
sudo apt-get install python3-pip
sudo pip3 install setuptools
```

If you are a beginner, please follow this chapter to let your Raspberry Pi work.

Depending on the different devices you use, you can start up the Raspberry Pi in different methods. If you have a separate screen for Raspberry Pi, follow the instructions eariler in this chapter. Otherwise, please find the corresponding steps in later parts of this chapter.

If You Have a Screen

If you have a screen, you can use the NOOBS to install the Raspberry Pi OS system.

Required Components

Any Raspberry Pi	1 * RPi Power Adapter
1 * Monitor	1 * Monitor Power Adapter
1 * HDMI cable	1 * Micro SD card
1 * Mouse	1 * Keyboard
1 * Personal Computer	

Procedures

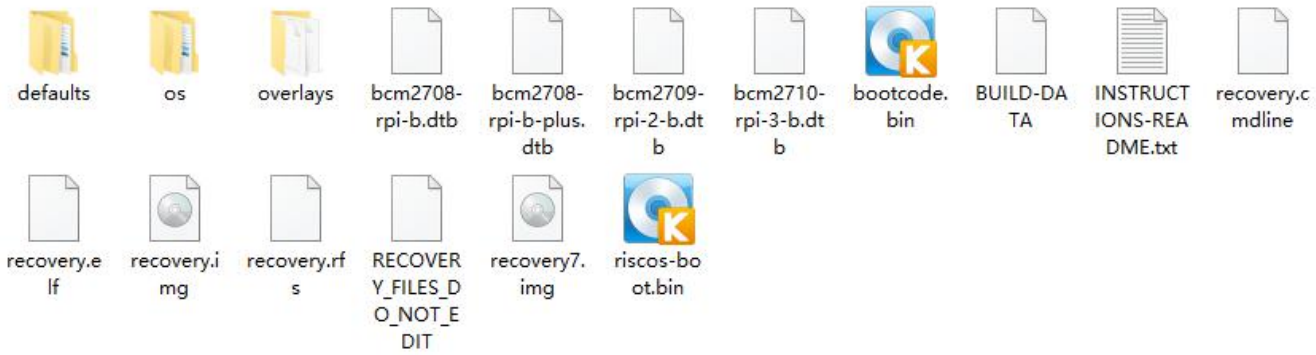
Step 1: To download NOOBS from your PC, you can choose **NOOBS** or **NOOBS LITE -** the only difference is that there is a built-in offline Raspberry Pi OS installer in **NOOBS**, while the **NOOBS LITE** can only be operated online. Here, you are suggested to use the former. Here is the download address of Noobs: *https://www.raspberrypi.org/downloads/noobs/*

Step 2: Plug in the Micro SD reader and format the Micro SD card with the **SD Formatter** (https://www.sdcard.org/downloads/formatter/index.html). If there are some important files in the Micro SD card, please backup them first.

Step 3: Next, you will need to extract the files from the NOOBS zip archive you downloaded from the Raspberry Pi website.

- Find the downloaded archive — by default, it should be in your Downloads folder.
- Double-click on it to extract the files, and keep the resulting Explorer/Finder window open.

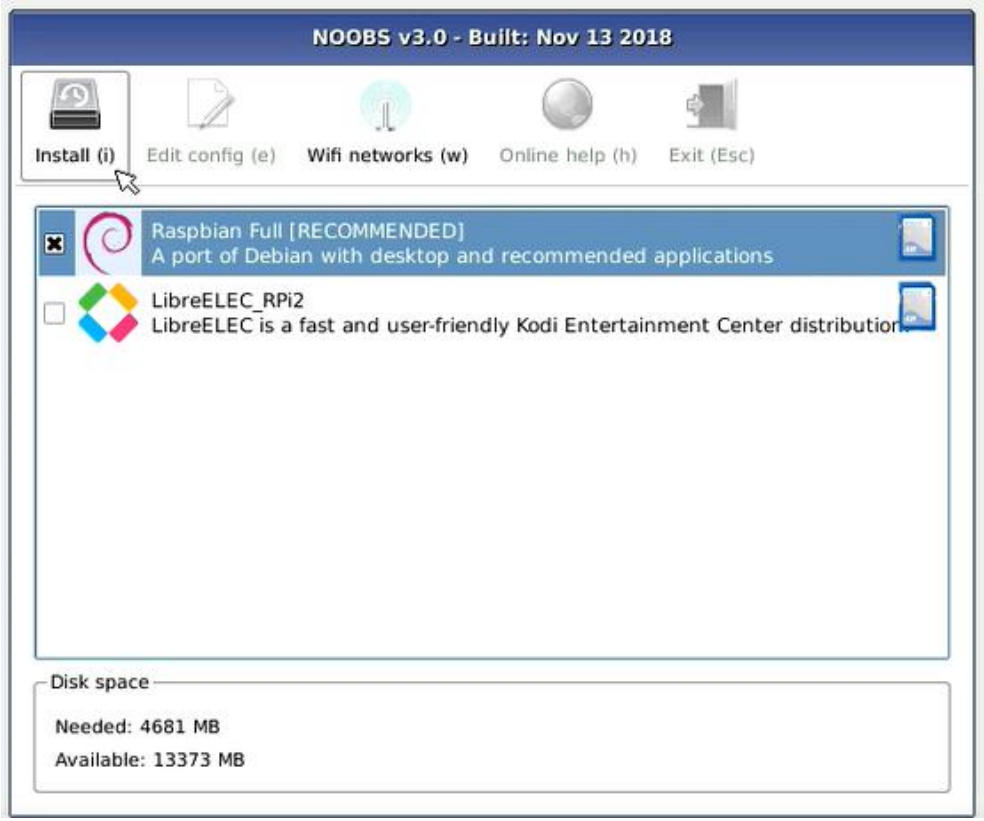
Finally select all the files in the NOOBS folder and copy them to the Micro SD card.



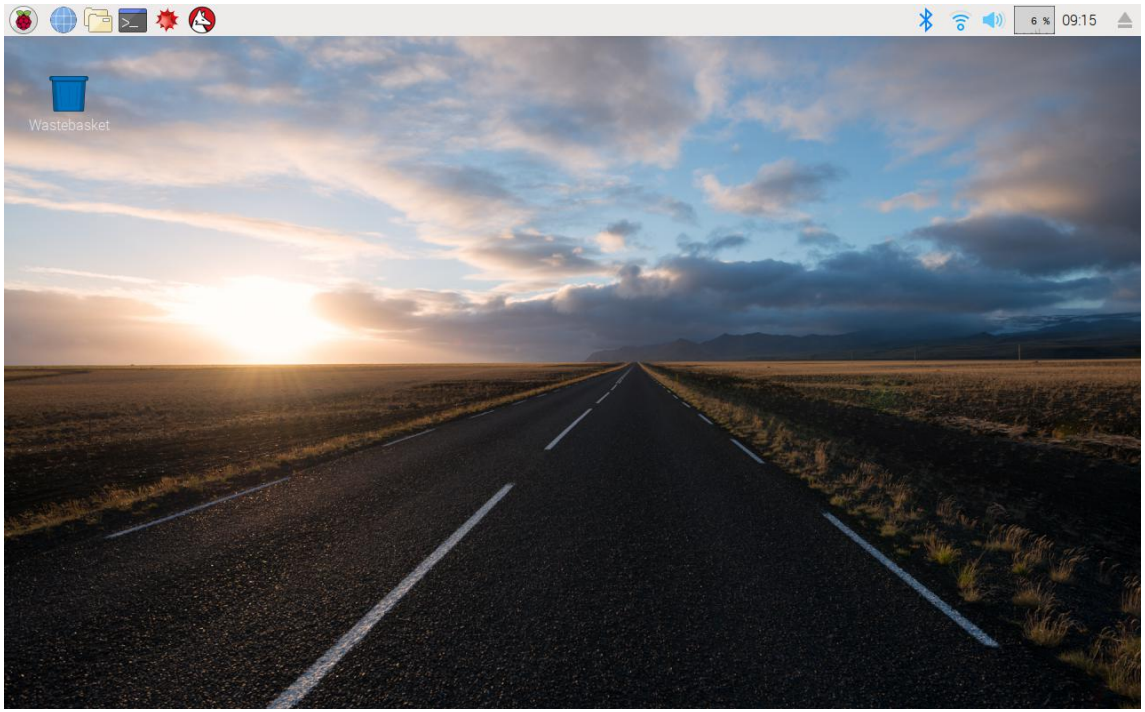
Step 4: All the files transferred, the Micro SD card pops up.

Step 5: Insert the Micro SD card into the Raspberry Pi. In addition, connect the screen, keyboard and mouse to it. Finally plug a power adapter into the Raspberry Pi.

Step 6: It will go to the NOOBS interface after starting up. If you use NOOBS LITE, you need to select Wi-Fi networks (w) first. Tick the checkbox of the Raspbian and click Install in the top left corner. The NOOBS will help to conduct the installation automatically. This process will take a few minutes.



Step 7: When the installation is done, the system will restart automatically and the desktop of the system will appear.



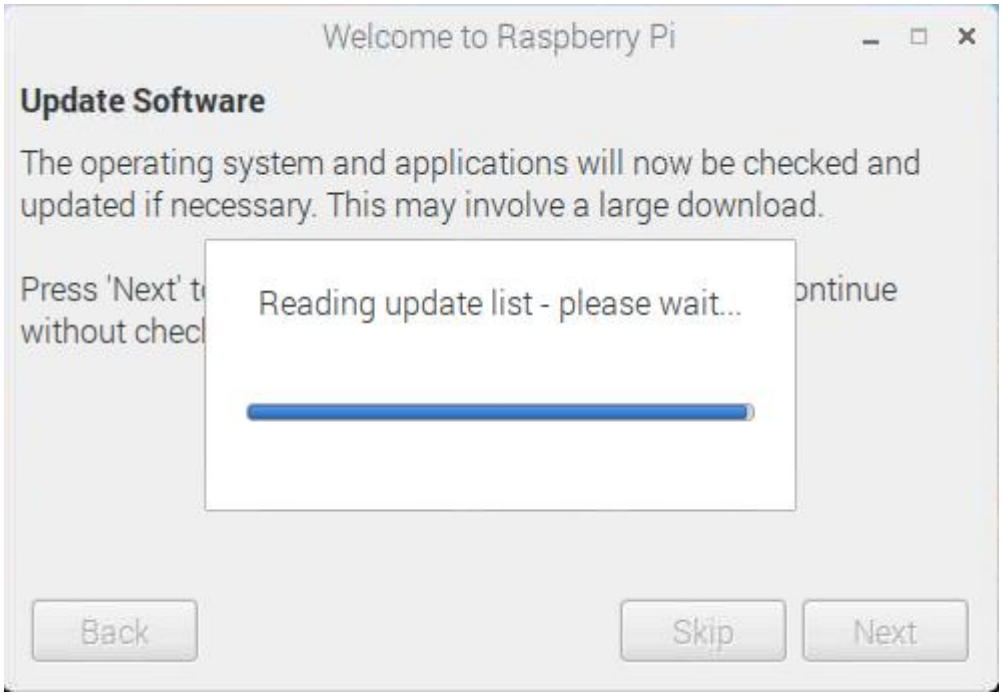
Step 8: If you run Raspberry Pi for the first time, the application of "Welcome to Raspberry Pi" pops up and guides you to perform the initial setup.

Step 9: Set country/region, language and time zone, and then click “next” again.

Step 10: Input the new password of Raspberry Pi and click “Next”.

Step 11: Connect the Raspberry Pi to WIFI and click "Next".

Step 12: Retrieve update.



Step 13: Click "Done" to complete the Settings.

Now we can run the Raspberry Pi.

Note: You can check the complete tutorial of NOOBS on the official website of the Raspberry Pi: *https://www.raspberrypi.org/help/noobs-setup/*.

If You Have No Screen

If we don't have a screen, we can directly write the Raspberry Pi OS system to the Micro SD card and we can control the Raspberry Pi on PC remotely by directly modifying the configuration file of the network settings in the Micro SD card.

Required Components

Any Raspberry Pi	1 * Power Adapter
1 * Micro SD card	1 * Personal computer

Installing System

There are 2 ways to install the system, **Using Raspberry Pi Imager** or **Using Raspberry Pi OS**. **Using Raspberry Pi Imager** is a kind of method recommended by Raspberry Pi official website for beginners with which you can directly write the Raspberry Pi OS into SD card after downloading Raspberry Pi Imager. However, each time the system is reinstalled, this method can take several hours.

In the later method, you need to download Raspberry Pi OS image at first, then use the tool to write it to your SD card, which can be confusing. But once you successfully finish the flashing at the first time, it only takes about 10 minutes to flash again.

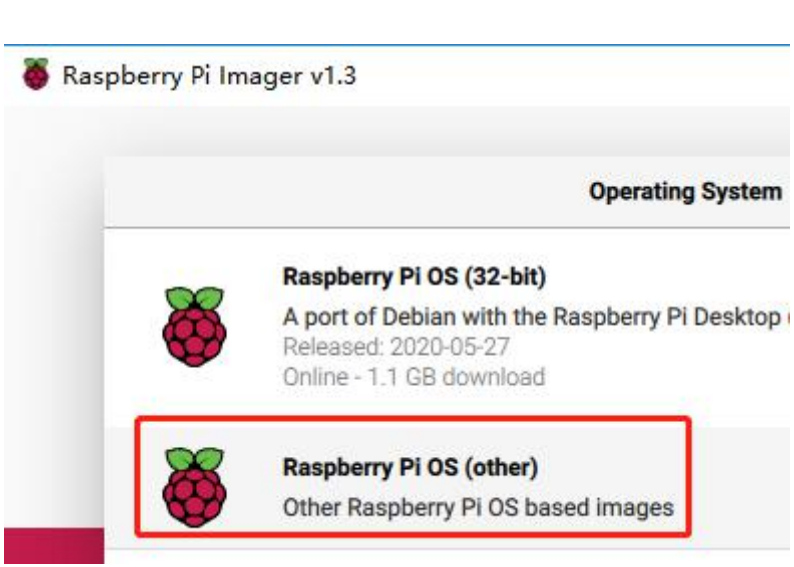
➢ Using Raspberry Pi Imager

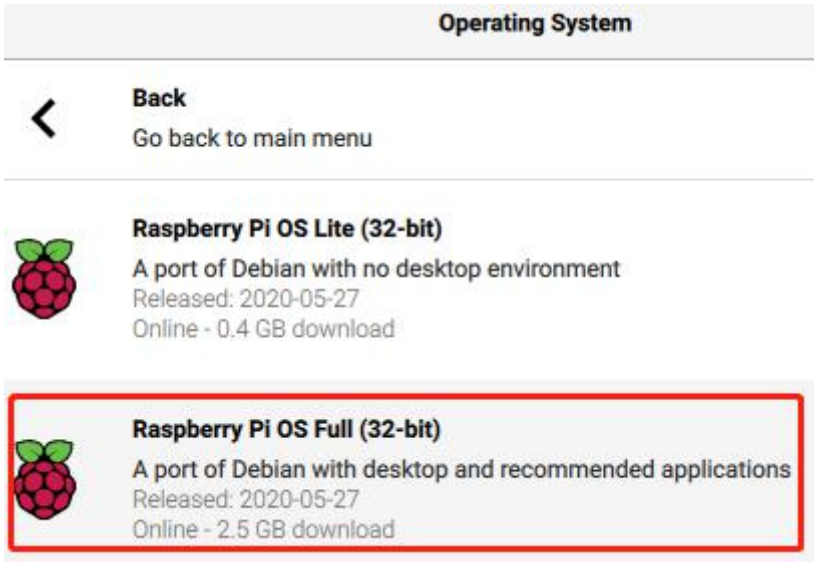
Raspberry Pi have developed a graphical SD card writing tool that works on Mac OS, Ubuntu 18.04 and Windows, and is the easiest option for most users as it will download the image and install it automatically to the SD card.

1) Download the latest version of **Raspberry Pi Imager**(https://www.raspberrypi.org/downloads/) and install it.

2) Connect an SD card reader with the SD card inside.

3) Open Raspberry Pi Imager and choose **Raspberry Pi OS (other) -> Raspberry Pi OS Full (32-bit)**.





4) Choose the SD card you wish to write your image to.

5) Review your selections and click 'WRITE' to begin writing data to the SD card.

Note: If using the Raspberry Pi Imager on Windows 10 with Controlled Folder Access enabled, you will need to explicitly allow the Raspberry Pi Imager permission to write the SD card. If this is not done, Raspberry Pi Imager will fail with a "failed to write" error.

➢ Using Raspberry Pi OS

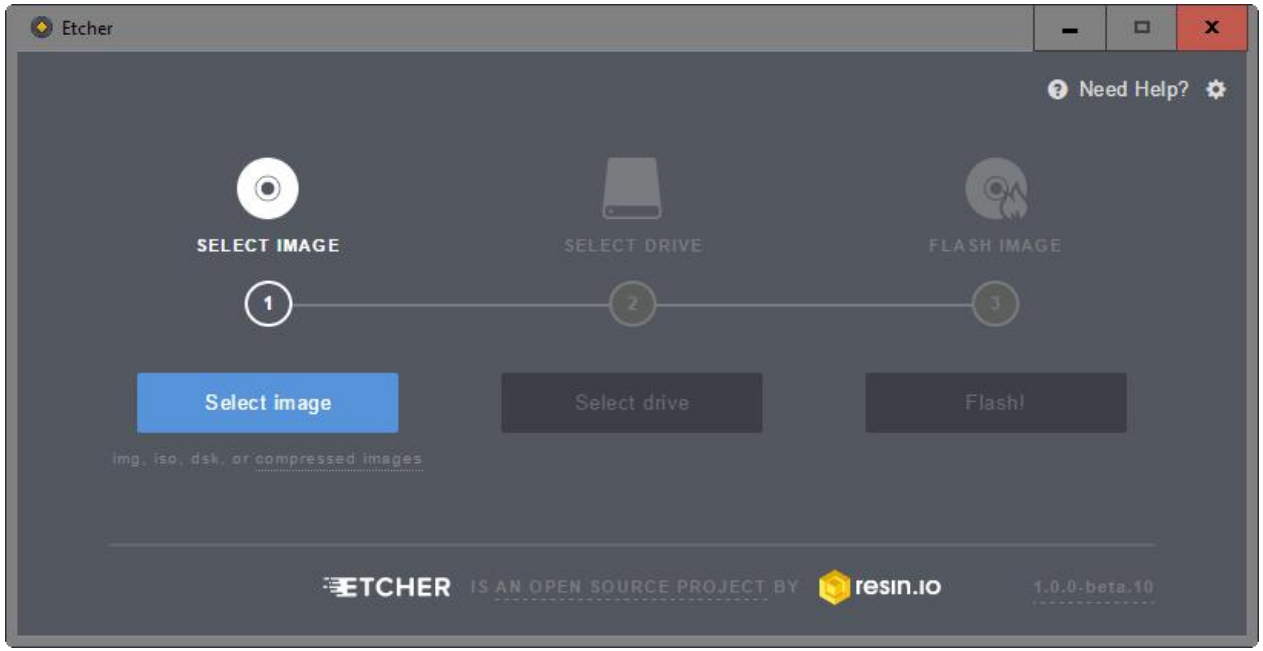
Step 1: Prepare the tool of image burning. Here we use the **balenaEtcher**. You can download the software from the link: *https://www.balena.io/etcher/*

Step 2: Download the complete image on the official website by clicking this link: *https://www.raspberrypi.org/downloads/raspbian/*. There are three different kinds of Raspberry Pi OS system available, You are recommend to install the version : **Raspberry Pi OS with desktop and recommended software**.

Step 3: Unzip the package downloaded and you will see the *.img* file inside.

Note: The Raspberry Pi OS with desktop image contained in the ZIP archive is over 4GB in size and uses the ZIP64 format. To uncompress the archive, a unzip tool that supports ZIP64 is required. The following zip tools support ZIP64: 7-Zip (Windows), The Unarchiver (Mac) and Unzip (Linux).

Step 4: Plug the USB card reader into the computer, then you can burn the **.img** file with the Etcher.



At this point, Raspberry Pi OS is installed. **Keep the USB card reader plug in your computer**. If you want to apply it, next you need to complete the settings accordingly.

Connect the Raspberry Pi to the Internet

There are two methods to help get the Raspberry Pi connected to the network: the first one is using a network cable, the other way is using WIFI. We will talk in detail about how to connect via WIFI as below.

If you want to use the WIFI function, you need to modify a WIFI configuration file wpa-supplicant.conf in the Micro SD card by your PC that is located in the directory **/etc/wpa_supplicant/**.

If your personal computer is working on a linux system, you can access the directory directly to modify the configuration file; however, if your PC use Windows system, then you can't access the directory and what you need next is to go to the directory, */boot/* to create a new file with the same name, **wpa_supplicant. conf**.

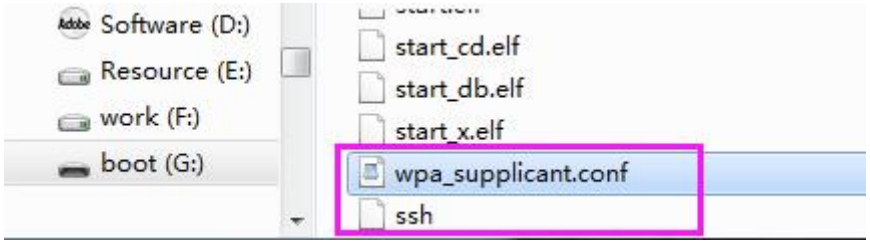
Input the following content in the file.

```
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
update_config=1
country=GB
network={
ssid="WiFi-A"
psk="Sunfounder"
key_mgmt=WPA-PSK
priority=1
}
```

You need to replace "**WiFi-A**" with your custom name of WiFi and "**Sunfounder**" with your password. By doing these, the Raspberry Pi OS system will move this file to the target directory automatically to overwrite the original WIFI configuration file when it runs next time. After doing this step, you also need to **keep the USB card reader plug in your computer.**

Start SSH

To use the function of remote control of the Raspberry Pi, you need to start SSH firstly that is a more reliable protocol providing security for remote login sessions and other network services. Generally, SSH of Raspberry Pi is in a disabled state. Additionally, if you want to run it, you need to create a file named SSH under directory /boot/.



Now, the Raspberry Pi OS system is configured.

Power on the Raspberry Pi

Plug out the USB card reader and then plug the Micro SD card into the Raspberry Pi. Finally plug a power adapter into the Raspberry Pi.

Get the IP Address

Here, we need to get the IP address of it. There are many ways to know the IP address, and two of them are listed as follows.

1. Checking via the router

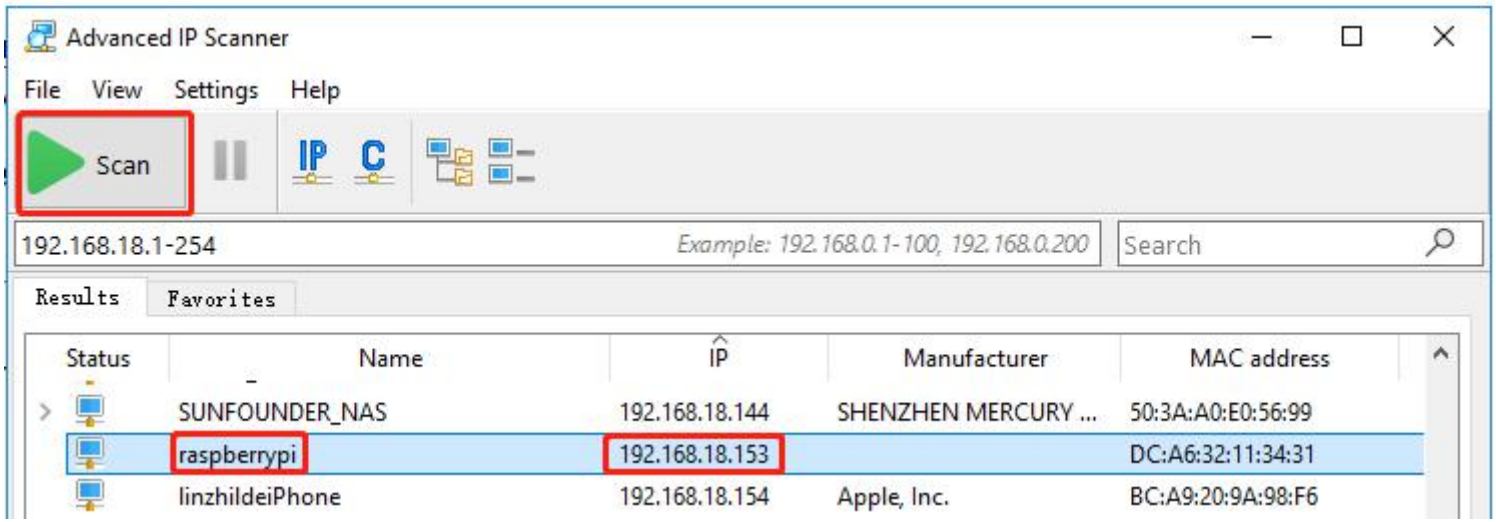
If you have permission to log in the router(such as a home network), you can check the addresses assigned to Raspberry Pi on the admin interface of router.

The default hostname of the system, Raspberry Pi OS is **raspberrypi**, and you need to find it. (If you are using ArchLinuxARM system, please find alarmpi.)

2. Network Segment Scanning

You can also use network scanning to look up the IP address of Raspberry Pi. You can apply the software, Advanced IP scanner(download from Google).

Click **Scan** and the name of all connected devices will be displayed. Similarly, the default hostname of the Raspberry Pi OS system is **raspberrypi**, now you need to find the hostname and its IP.



Use the SSH Remote Control

We can open the Bash Shell of Raspberry Pi by applying SSH. Bash is the standard default shell of Linux. The Shell itself is a program written in C that is the bridge linking the customers and Unix/Linux. Moreover, it can help to complete most of the work needed.

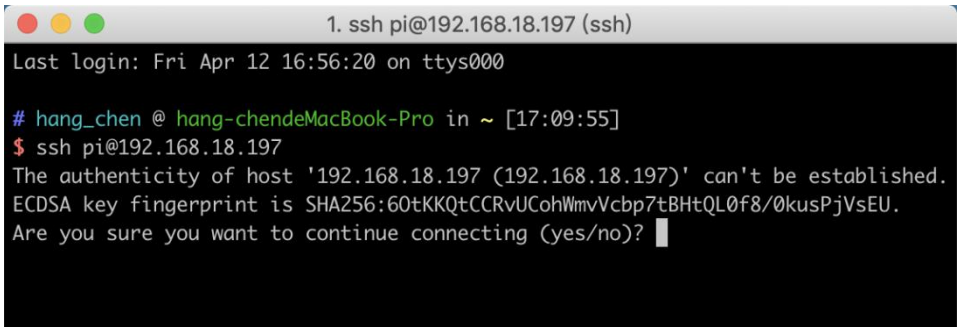
➢ For Linux or/Mac OS X Users

Step 1: Go to **Applications**->**Utilities**, find the **Terminal**, and open it.

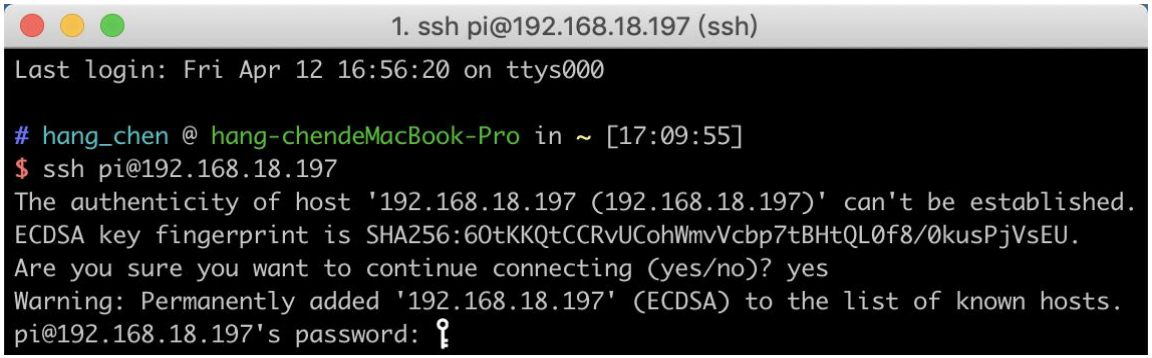
Step 2: Type in **ssh pi@ip_address** . "pi"is your username and "ip_address" is your IP address. For example:

```
ssh pi@192.168.18.197
```

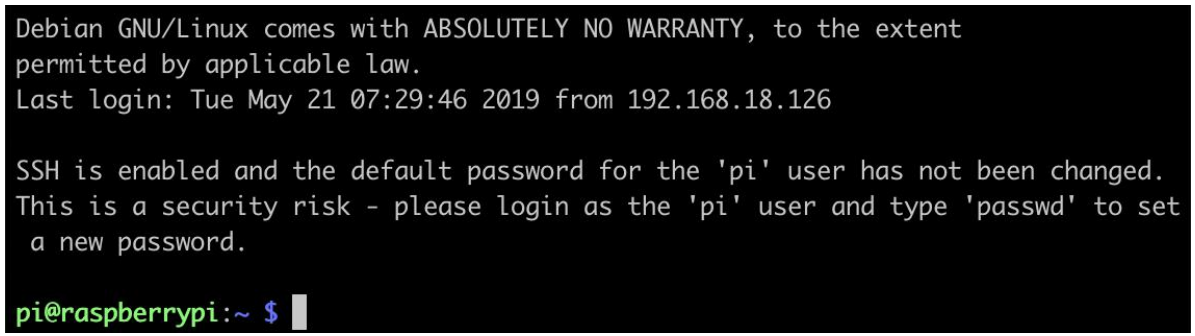
Step 3: Input"yes".



Step 4: Input the passcode and the default password is **raspberry**.



Step 5: We now get the Raspberry Pi connected and are ready to go to the next step.

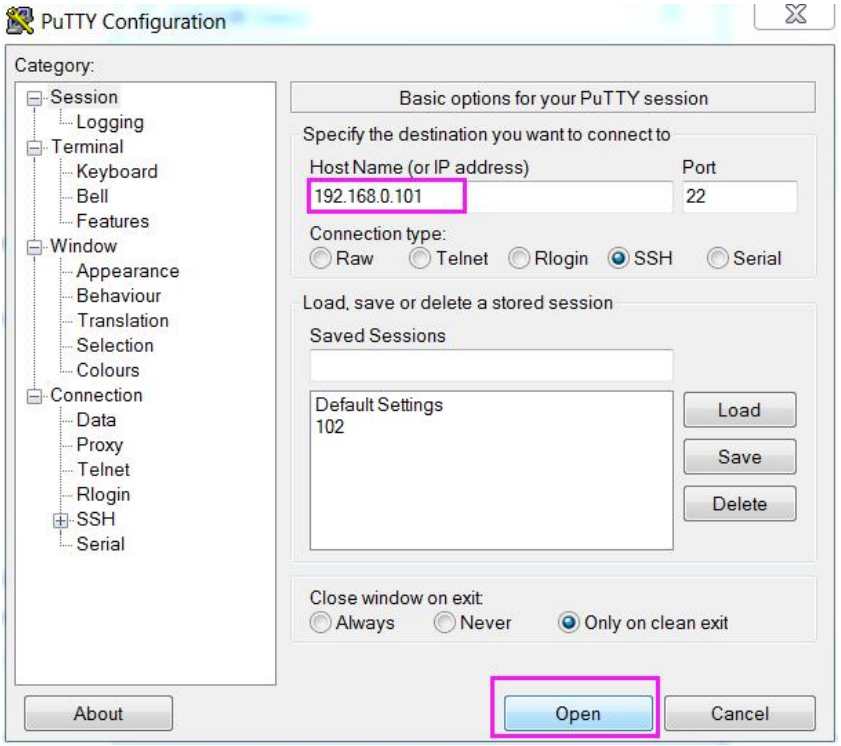


Note: When you input the password, the characters do not display on window accordingly, which is normal. What you need is to input the correct passcode.

➢ For Windows Users

If you're a Windows user, you can use SSH with the application of some software. Here, we recommend **PuTTY**.

Step 1: Download PuTTY. Open PuTTY and click **Session** on the left tree-alike structure. Enter the IP address of the RPi in the text box under **Host Name (or IP address)** and **22** under **Port** (by default it is 22).



Step 2: Click **Open**. Note that when you first log in to the Raspberry Pi with the IP address, there prompts a security reminder. Just click **Yes**.

Step 3: When the PuTTY window prompts "**login as:**", type in "**pi**"(the user name of the RPi), and **password:** "raspberry" (the default one, if you haven't changed it).

Step 4: Here, we get the Raspberry Pi connected and it is time to conduct the next steps.

Note: When you input the password, the characters do not display on window accordingly, which is normal. What you need is to input the correct password.

Download the Code

The codes in the disk don't need downloading, only for your fast study. If you need the latest codes, please go to github to download as following methods.

① Change directory to */home/pi.*

```
cd /home/pi/
```

Note: cd, short for change directory is to change from the current path to the intended directory. Informally, here is to go to the path /home/pi/.

② Clone the repository from github.

```
git clone https://github.com/sunfounder/picar-4wd
```

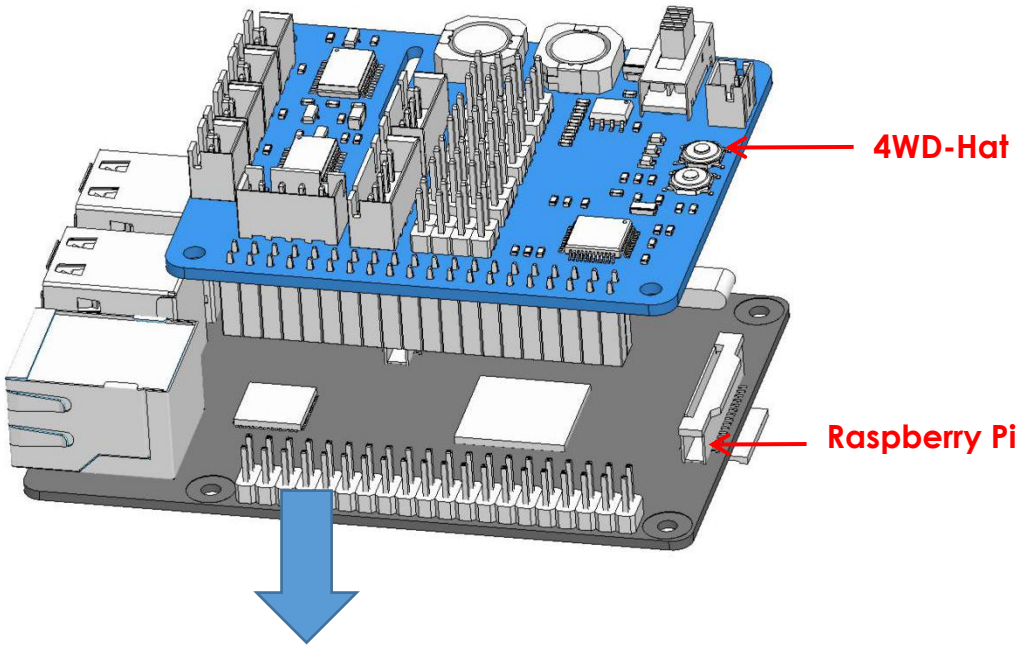
Test the Modules

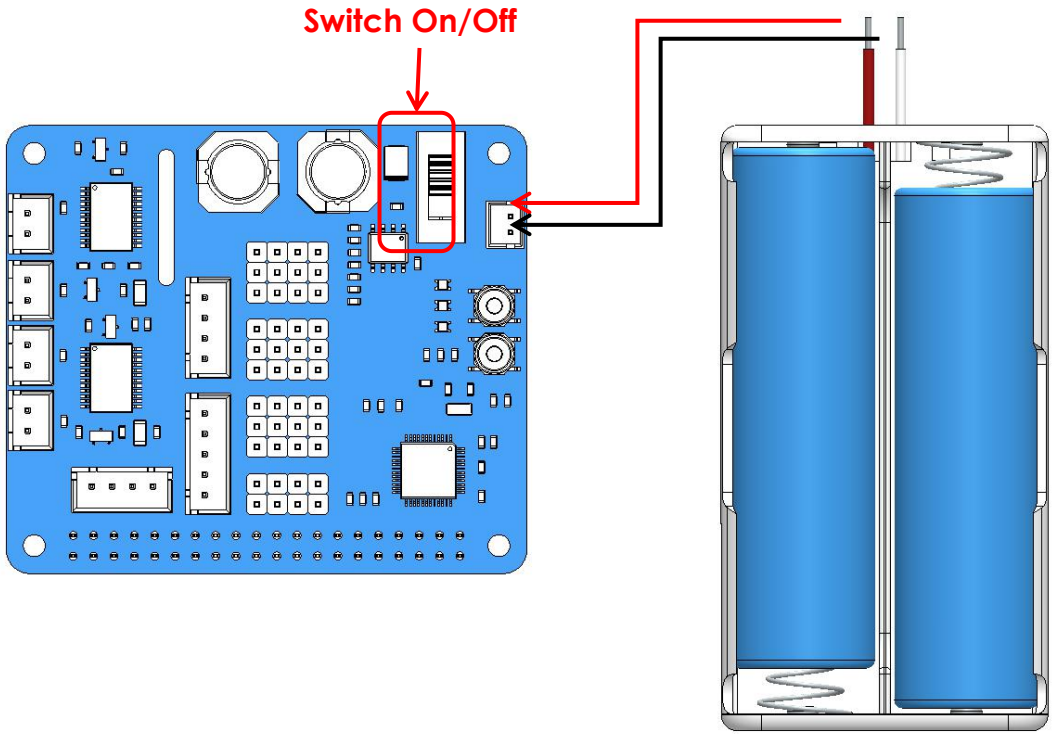
In case of anything wrong with the components, you can firstly debug them, then start your assembling and use.

Initialize the Environment

Initialize the environment before using the product example. The methods are as follows.

① Assemble 4WD-Hat and boot the Raspberry Pi.





Note: Due to the long test time, you can also use an external power supply to power the Raspberry Pi at the same time. But 4WD-Hat can only be powered by battery, you can't just use external power supply without batterries, otherwise the following error will occur when you are running the command: ***picar-4wd test motor***.

---- *picar-4wd test motor returns OSError: [Errno 121] Remote I/O error* ----

② Enter the folder picar-4wd.

```
cd /home/pi/picar-4wd/
```

③ Start up the initialization function.

```
sudo python3 setup.py install
```

The main functions of setup.py may be:

A. Download and configure necessary library files, including pip, sysstat, i2c-tools.

B. Provide entry points called「picar-4wd」to call the function of example code.

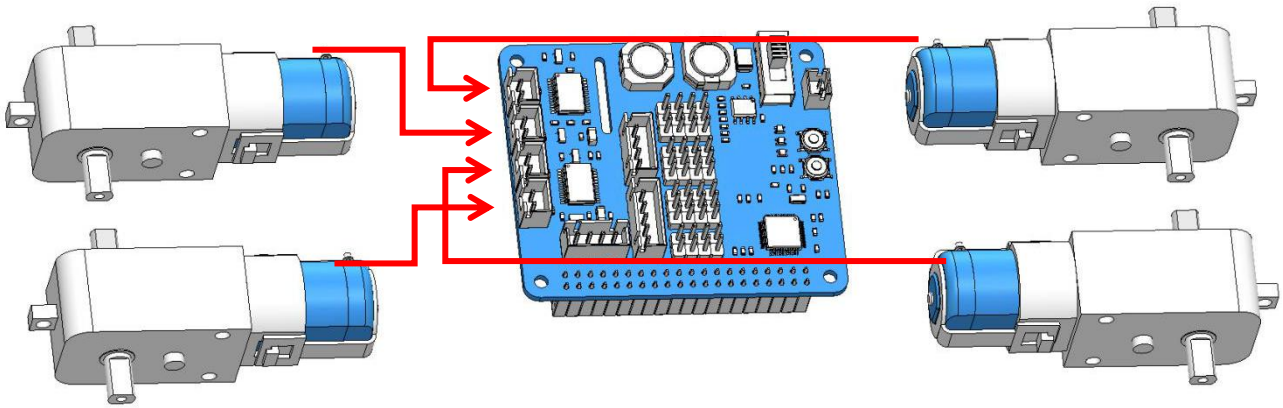
Note: Initialization needs connecting to the internet and spending a lot of time. The final line will appear「finish」after configuration is finished.

Module Testing

To make sure that the module works well, you can test the module by these means.

① Testing the motor of the car.

Wire up the motor according to the diagram.



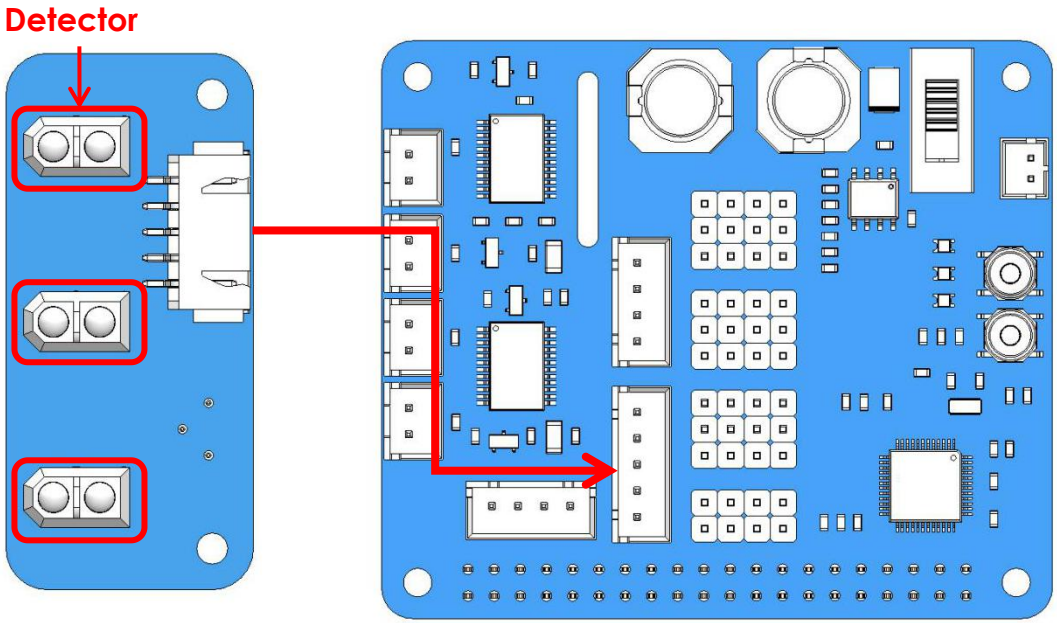
Input the command.

```
picar-4wd test motor
```

After you input the command, the four motors will last working till you press on Ctrl+C. Among the motors, two of them rotate clockwise, and the rest of them rotate anticlockwise.

② Testing the grayscale sensor module.

Finishing the wire up according to the diagram.



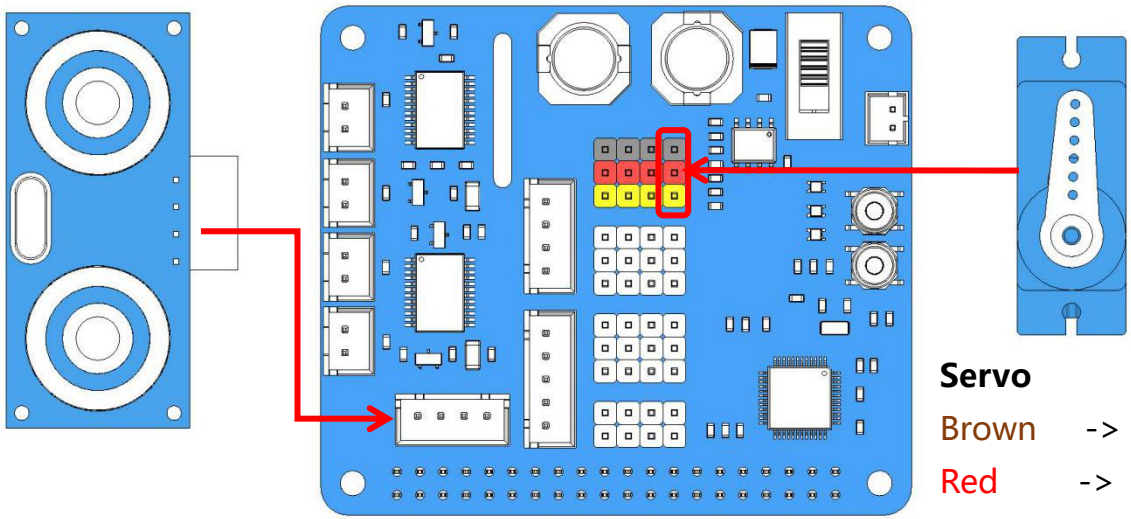
Input the following command.

```
picar-4wd test grayscale
```

After you input the command, the monitor will print the readings of three detectors of line-following module once.

③ Testing the servo and the ultrasonic module.

Start your wire up:



Input the commands:

```
picar-4wd test servo
```

The arm of servo will rotate to the angle of 0, meanwhile the monitor will print the readings of ultrasonic module once.

Note: You need to boot this function to check the direction of servo's output axis when you assemble the car and the servo.

Power Supply Detection

You can use the following command to get the supply voltage.

```
picar-4wd power-read
```

The supply voltage ought to be 6V~8.5V to ensure that the car works well.

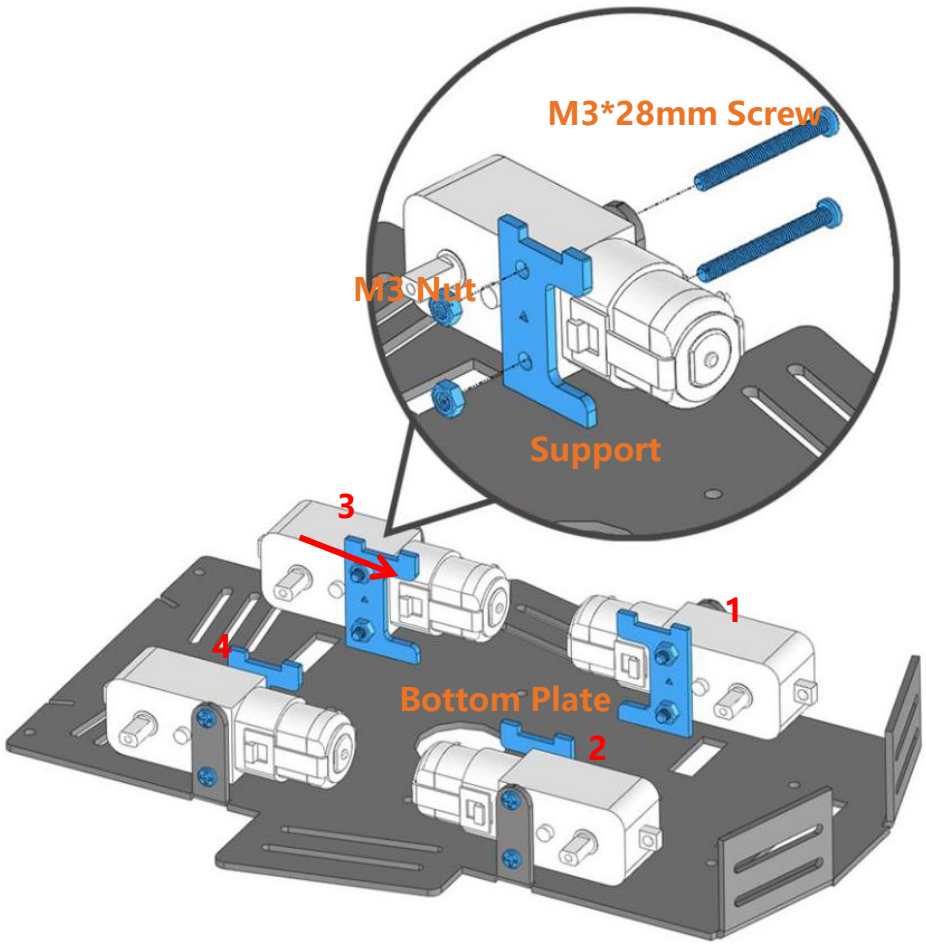
Reset Controller

If you encounter some errors during your use, for example, the car don't make response to your control command of control terminal, you can try to reset HATS by using the following command.

```
picar-4wd soft-reset
```

Hardware Assembling

Assembling Motor



Assemble 4 motor to the bottom plate.

Note:

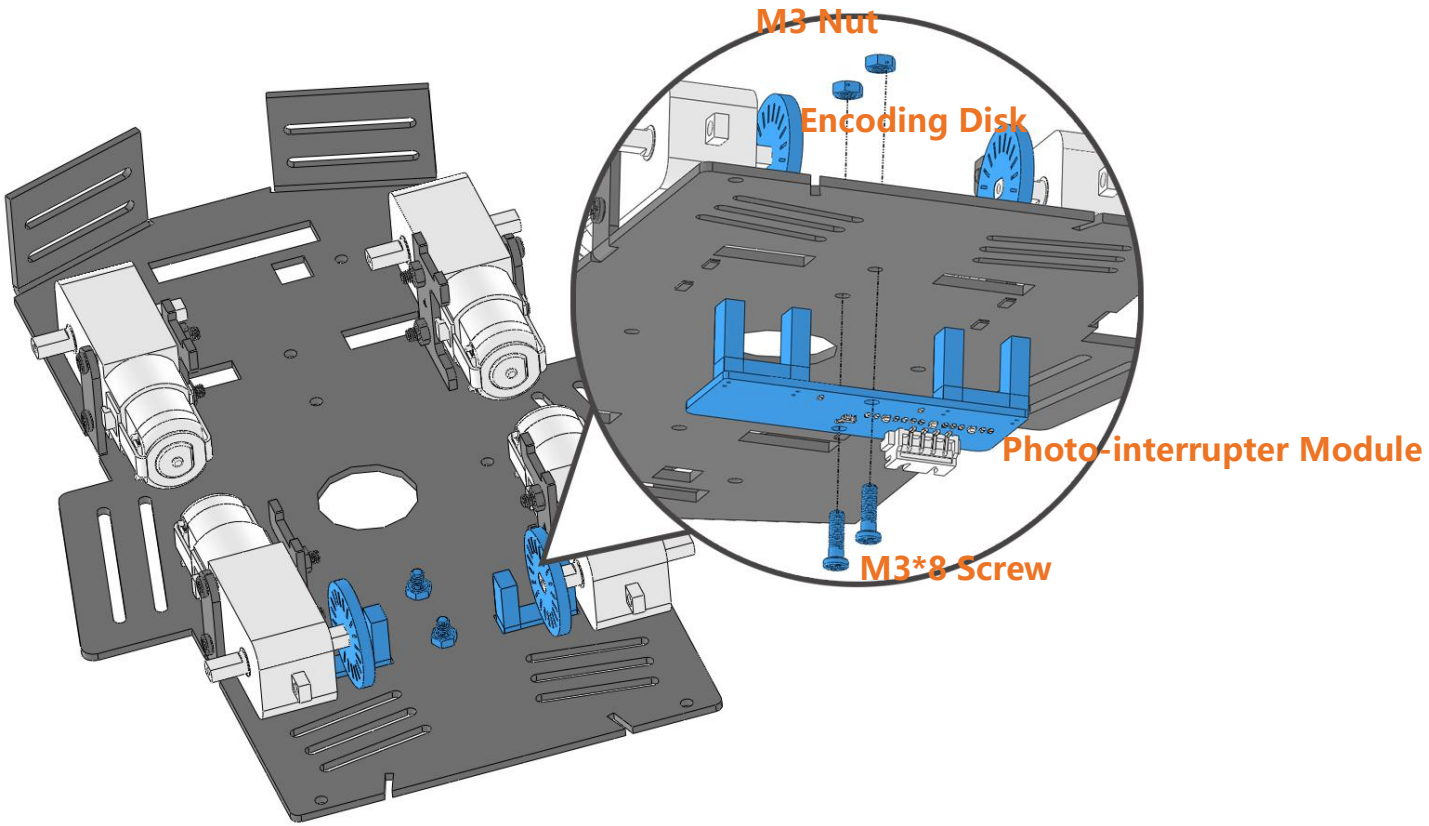
Keep the vertex angle of the triangle sticking upward.

For the convenience of later assembly, please mark the number(1, 2, 3, 4) on the socket of the 4 motors.

Photo-interrupter Module + Encoding Disk

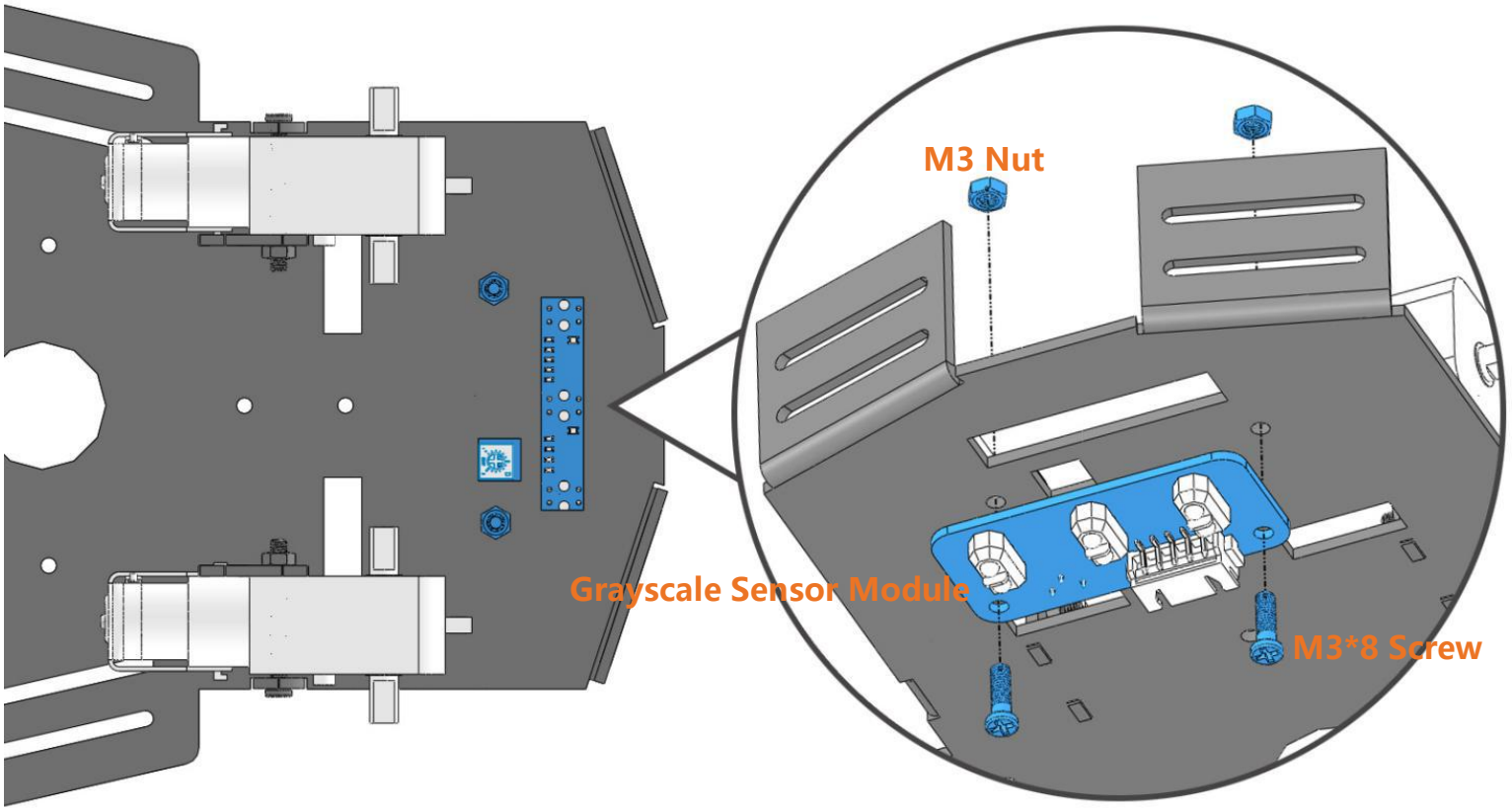
Insert the Encoding Disk into the motor, then assemble the Photo-interrupter module from bottom to top.

Note: Keep the Encoding Disk on the motor away from the Photo-interrupter module to avoid the potential abrasion.



Assembling Grayscale Sensor Module

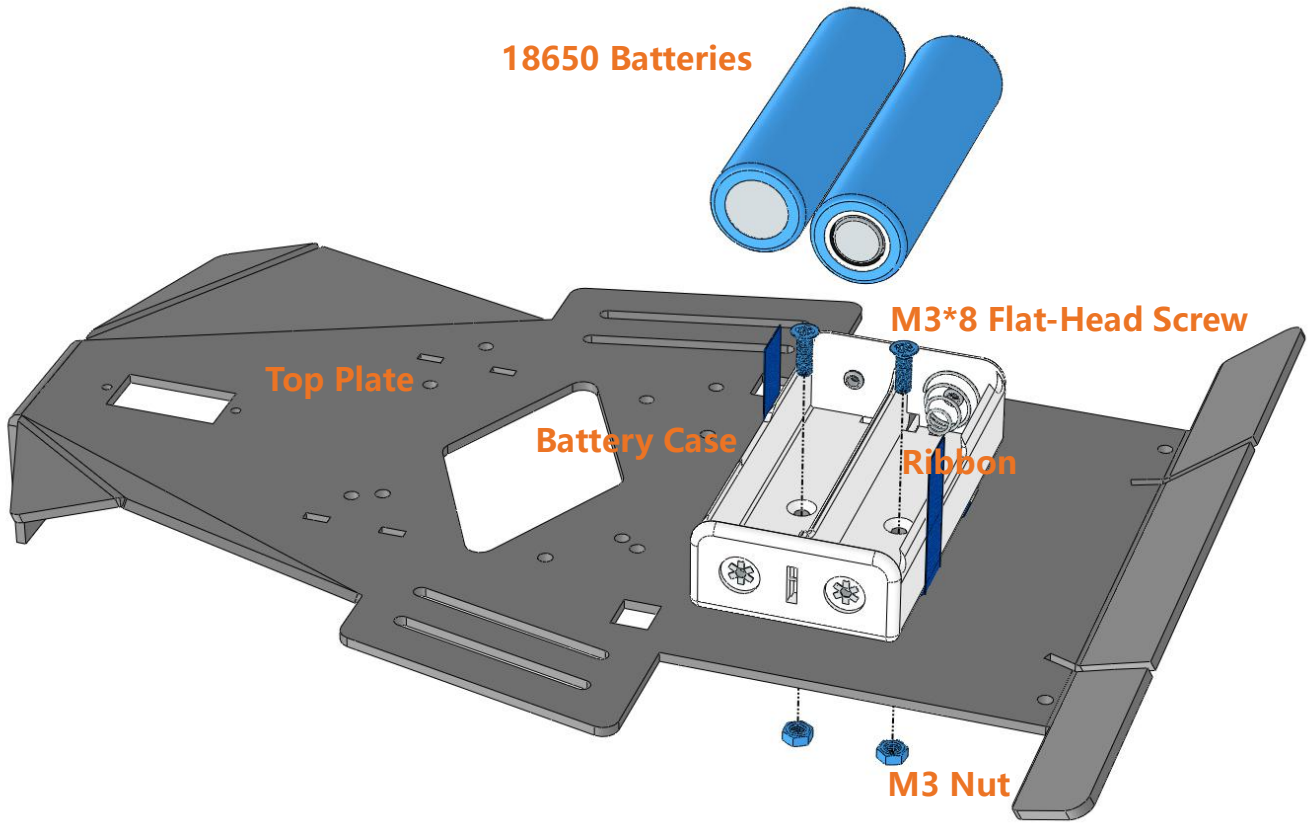
Have the Grayscale sensor module be under the head of the car.



Assembling Battery Holder

Assemble the battery holder to the top plate and put the battery in.

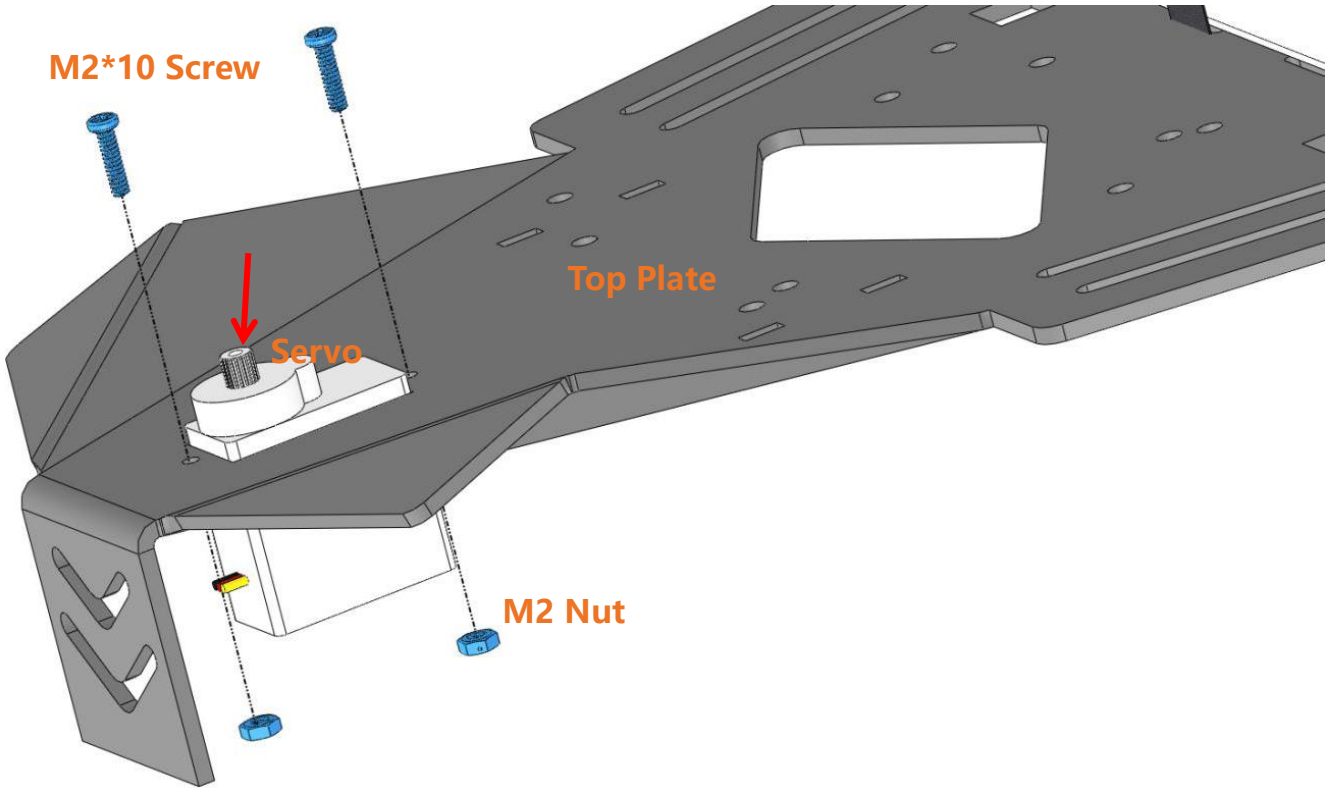
Note: To get the battery out of the holder easily, you can wind the battery with ribbon like this.



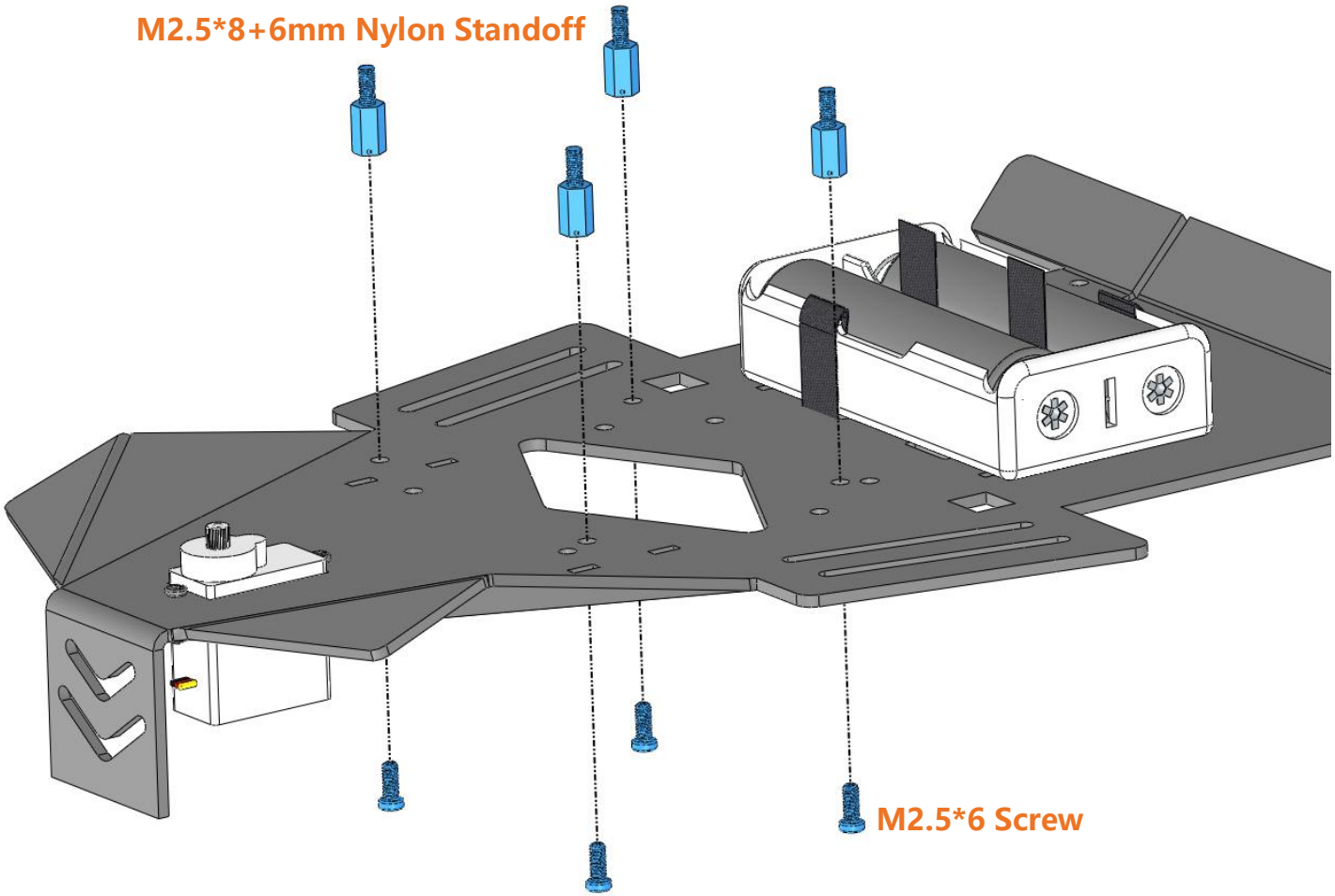
Assembling Servo

Assemble the servo from bottom to top.

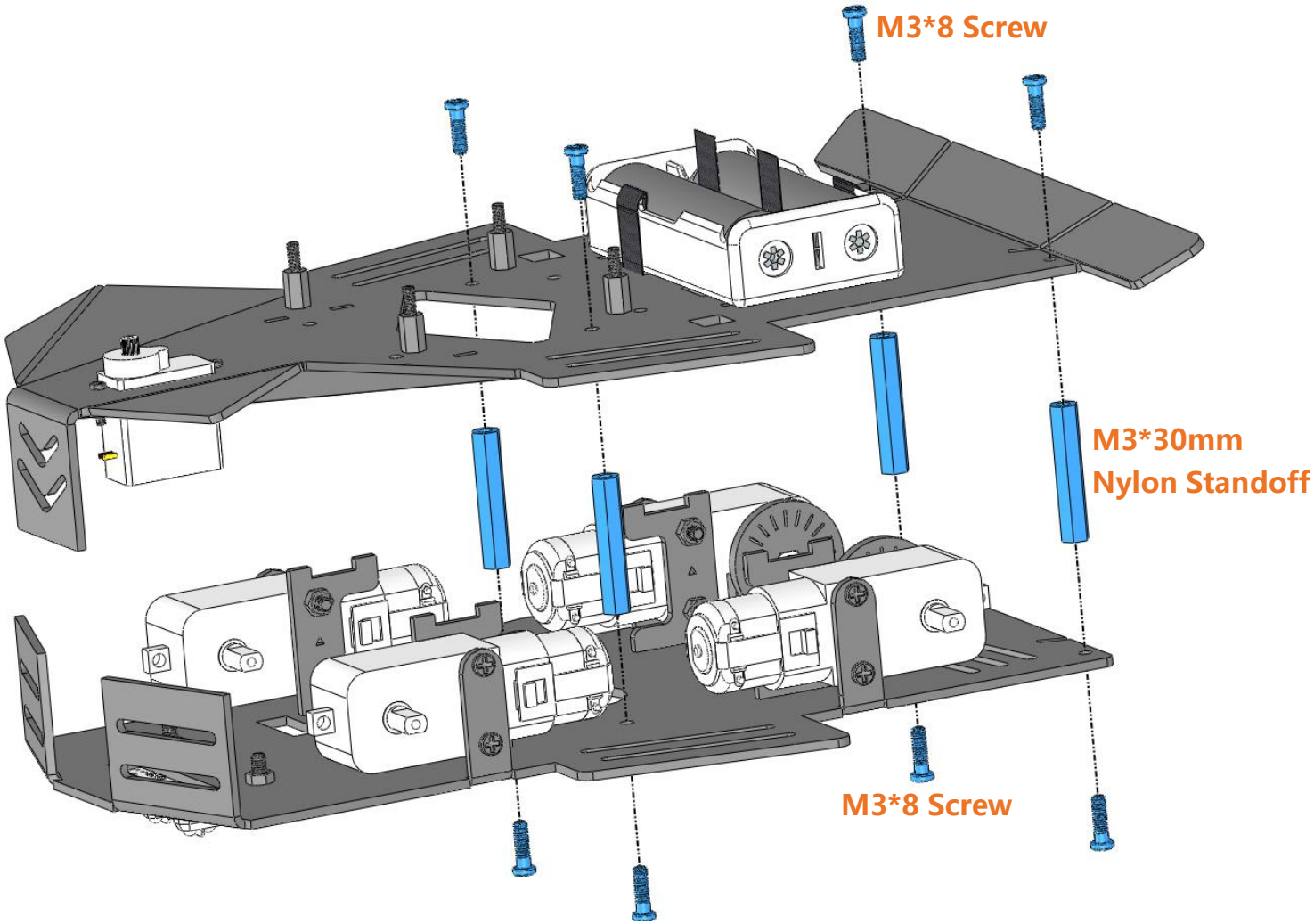
> **Note:** The output axis is to face outwards.



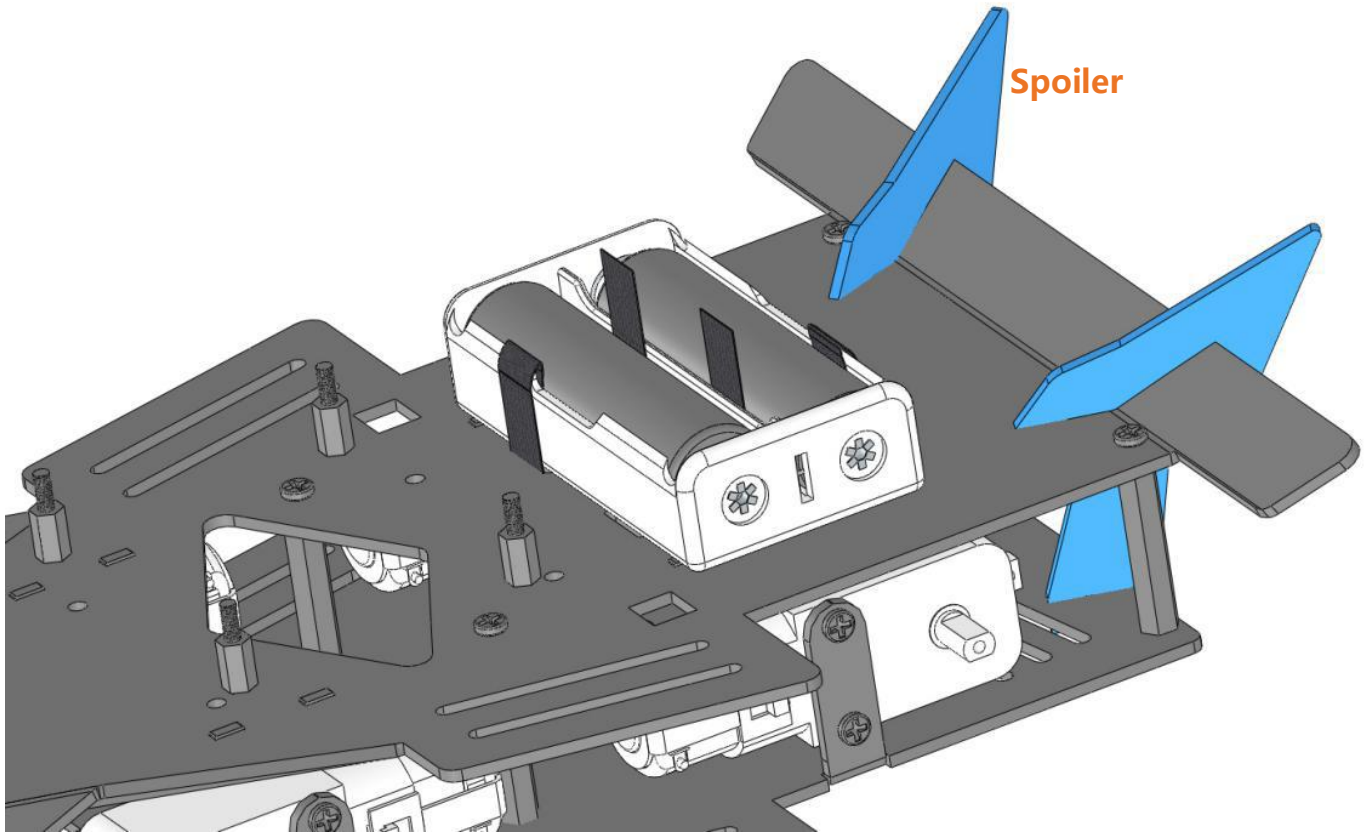
Assembling Nylon Standoffs



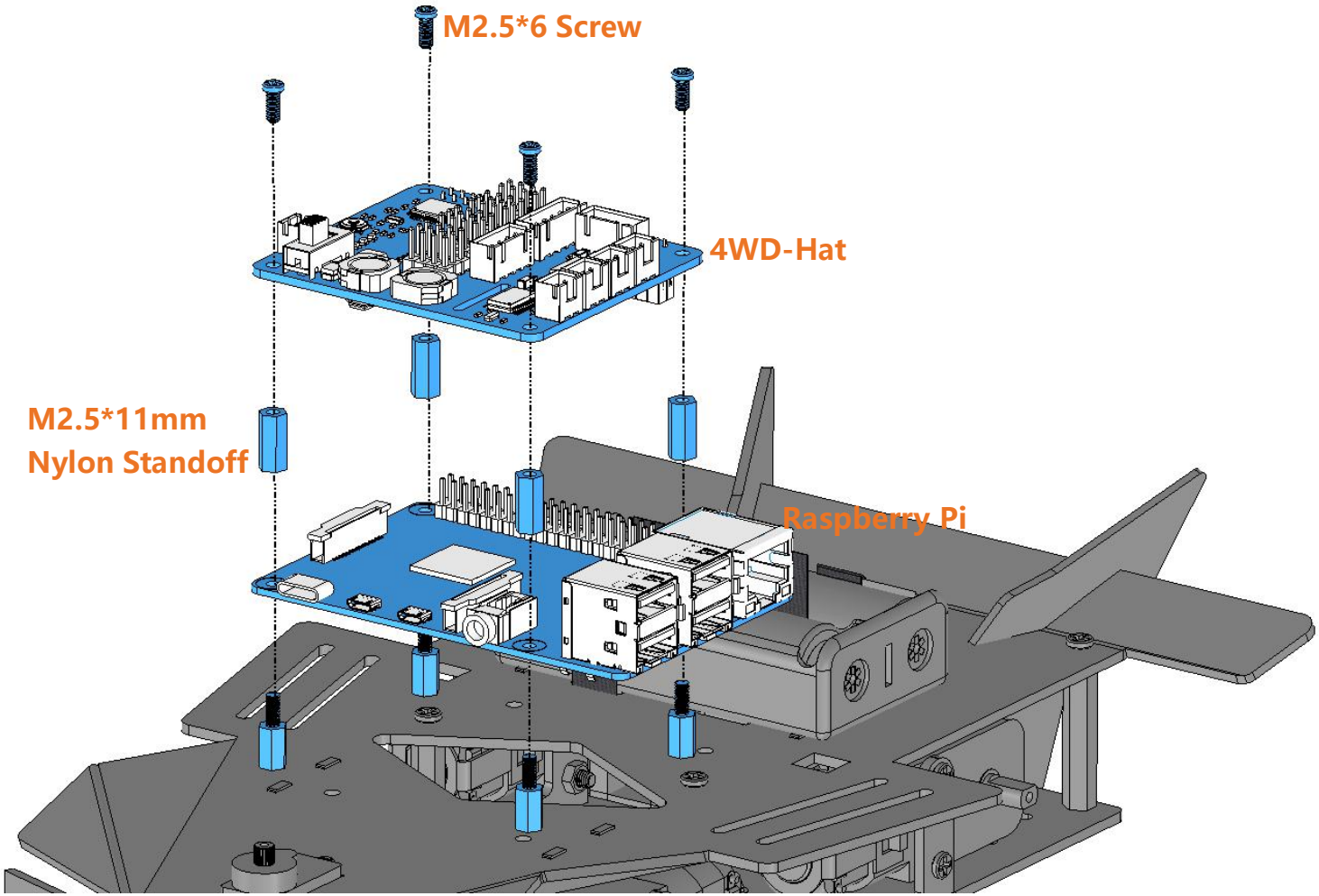
Assembling Top Plate and Bottom Plate



Assembling Spoiler



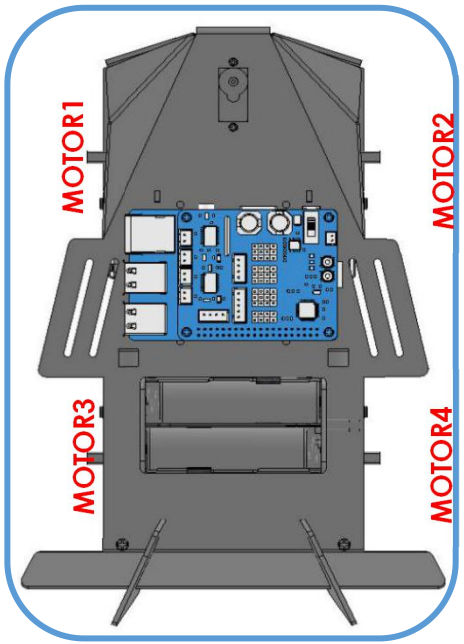
Assembling Control Board

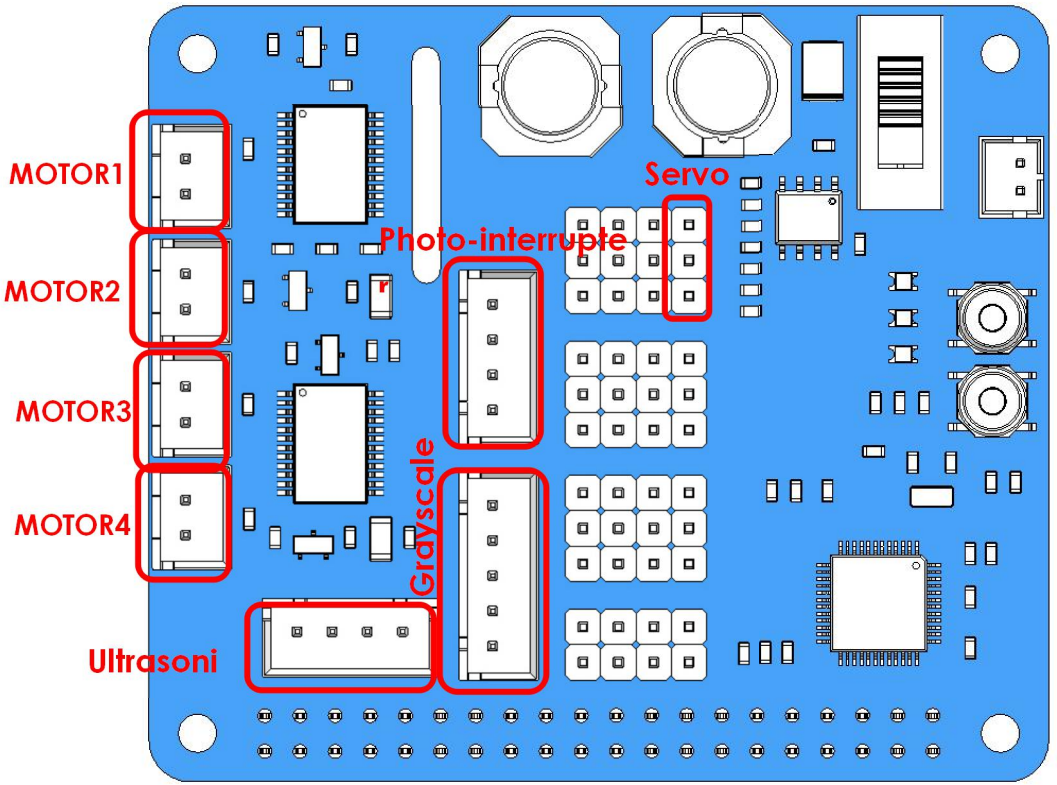


Connect Wires

Connect the modules properly with the guidance of the marks on the 4WD-Hat board.

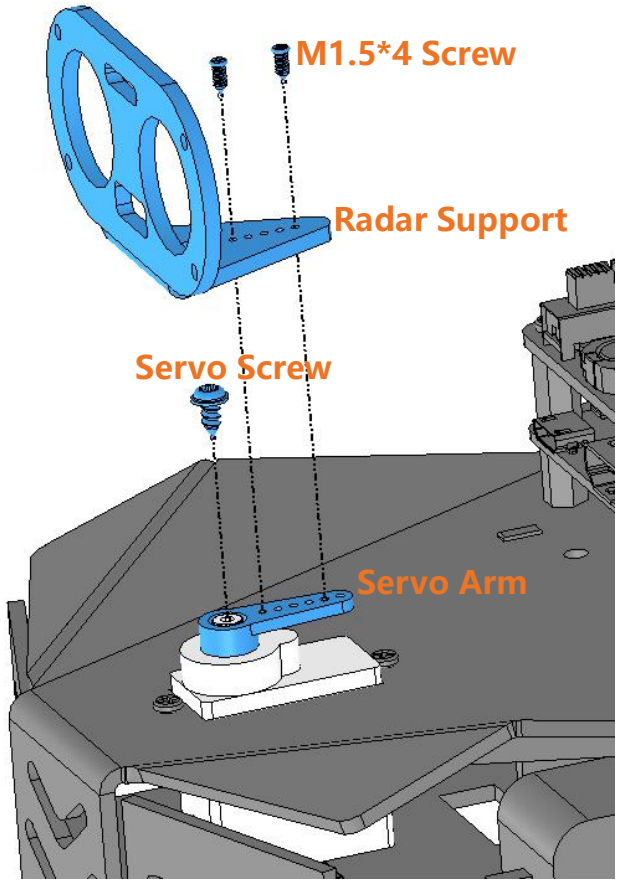
Note: For good appearance, you need to make sure that all the wires go through the bottom plate of the car to the HAT on the upper plate of the car.



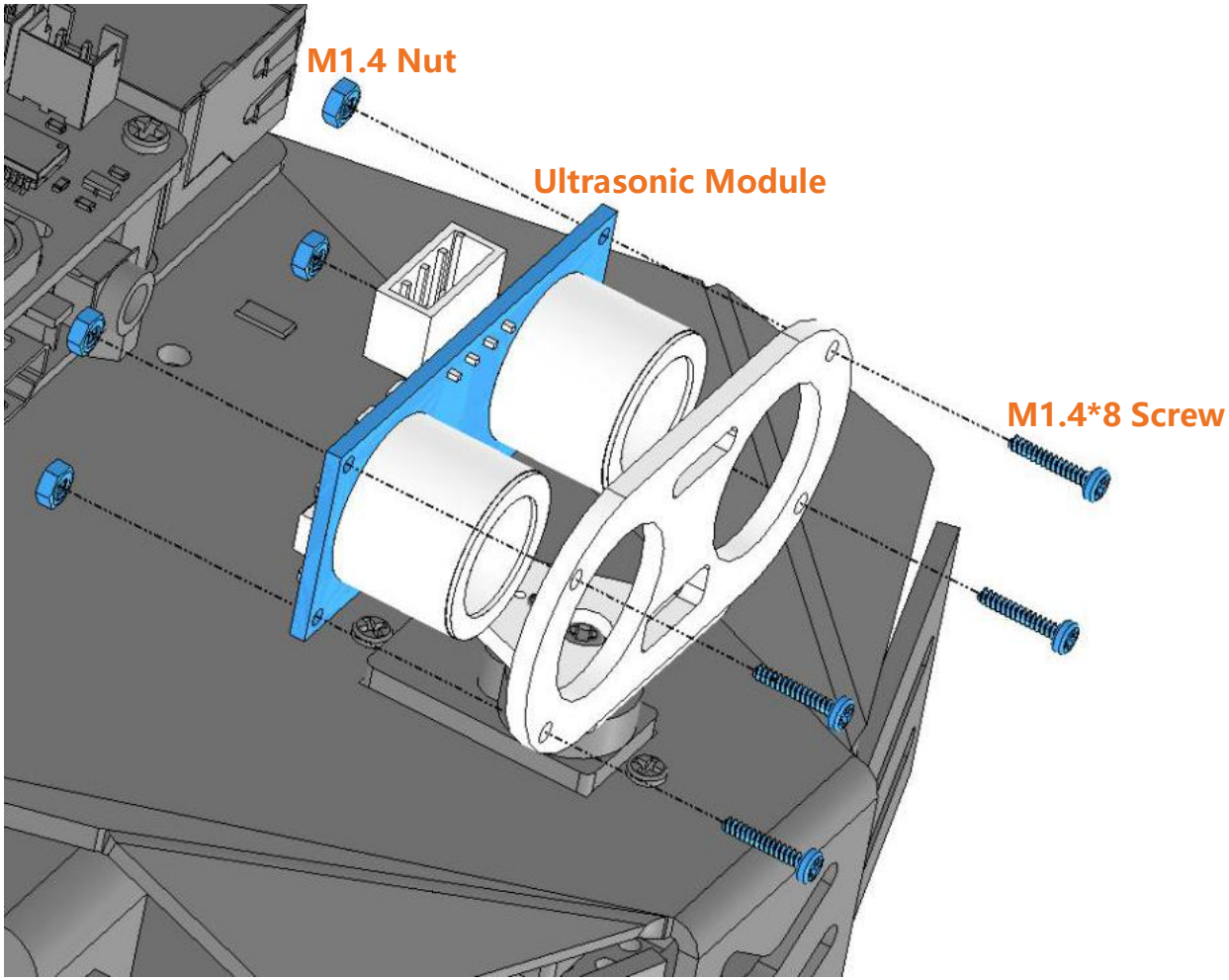


Assembling Radar Support

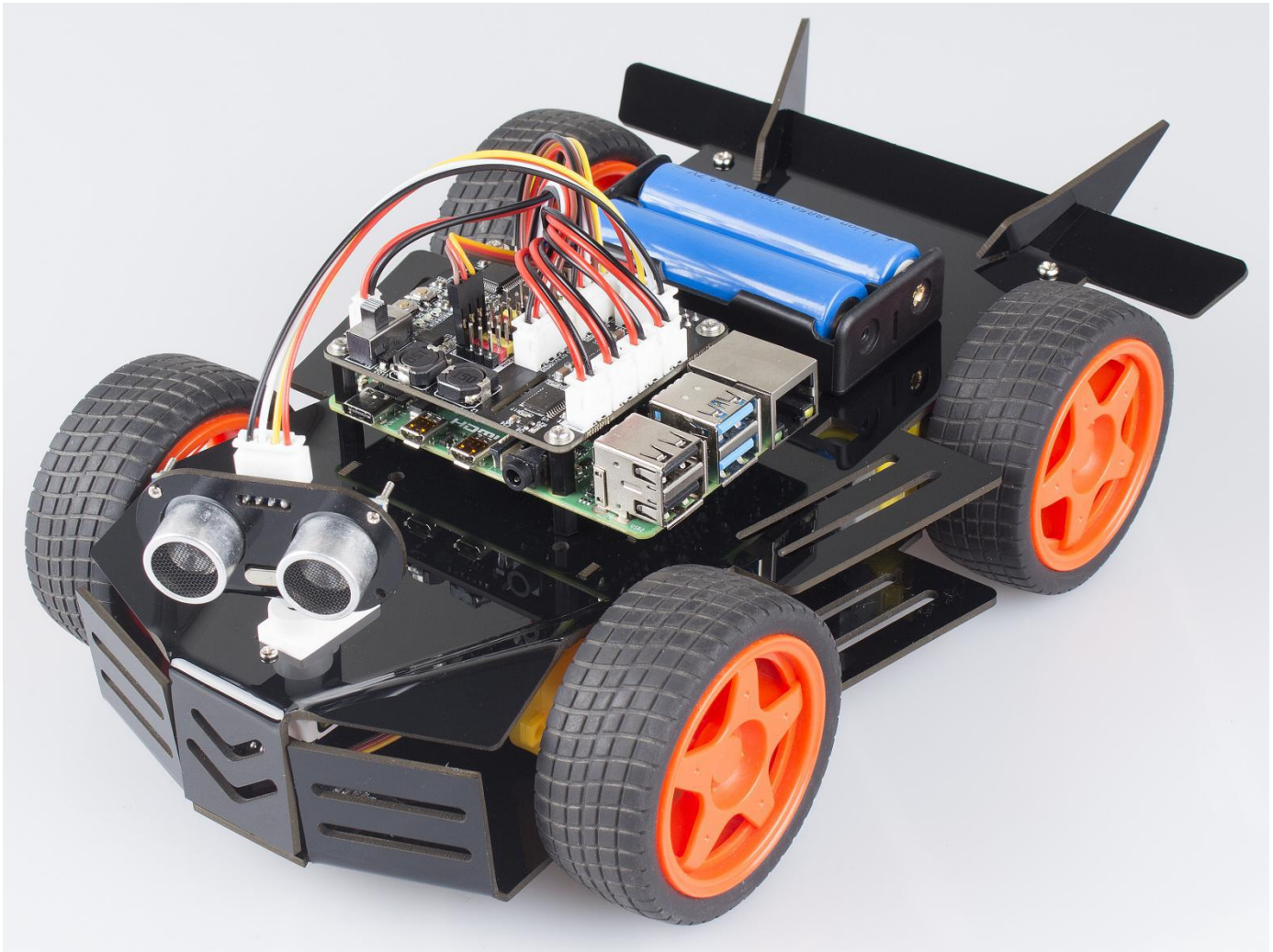
Note: Assemble the servo arm only when the Raspberry Pi is energized and the command 「picar-4wd test servo」is running.



Assembling Ultrasonic Sensor Module



Assembling Wheels(finished)



Web Control

In the kit, there are two control methods: Web control and Python program control.

In this chapter, we are provided with method of web control. So we can open the browser on different devices, including computer, **cellphone（Recommended）**, tablet and so on. You can use most functions of Picar-4wd, such as obstacle avoidance, line-following, object following and voltage check.

Note: Check the next chapter if you want to use python to control Picar-4wd.

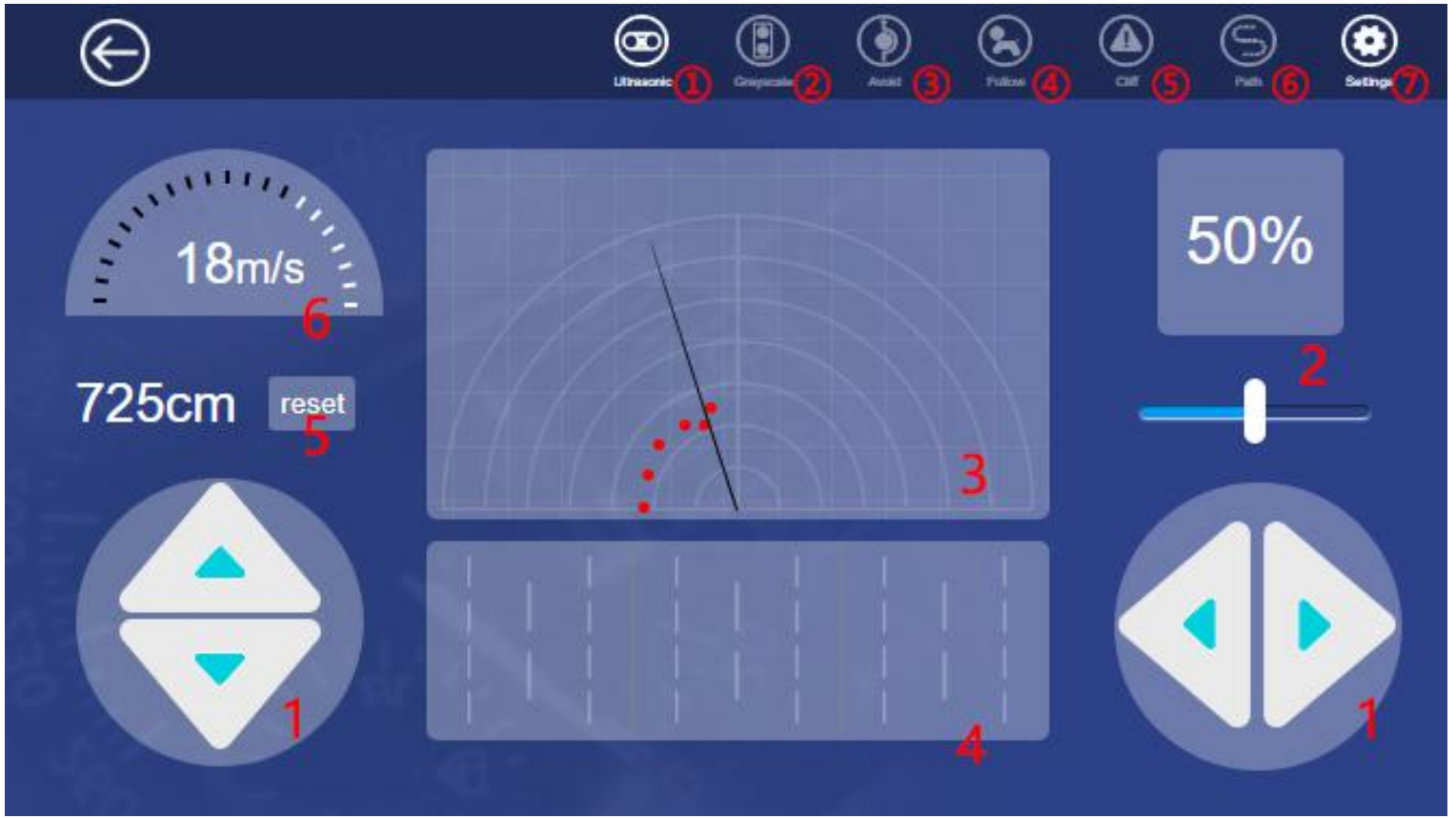
Boot Web Control

Input the command and start up the web control program.

```
cd /home/pi/picar-4wd/
picar-4wd web-example
```

As the program is running, on the browser, type the IP of the Raspberry Pi, and you can enter the Web interface.

Interface Introduction



1. Arrow Key

2. Power Regulator

3. Obstacle Monitor: checking the obstacle ahead within 35cm and 180°. Red dot represents there is an obstacle ahead detected by ultrasonic module.

4. Grayscale Monitor: displaying the color condition of the detected surface. If the module detects the black lines and then the cliff, the different colors will display on the monitor.

5. Mileage

6. Speed Monitor

①. Ultrasonic: turning on/off the obstacle monitor and the ultrasonic measurement function.

②. Grayscale: turning on/off the grayscale monitor and the line following function.

③. Avoid: turning on/off obstacle avoidance function can make the car get around the obstacles automatically or not.

④. Follow: turning on/off line following function can let the car follow the objects ahead or not.

⑤. Cliff: turn on/off edge detection. The car retreats once it detects the cliff or not. If the car doesn't realize the effect, please adjust the threshold on the setting page.

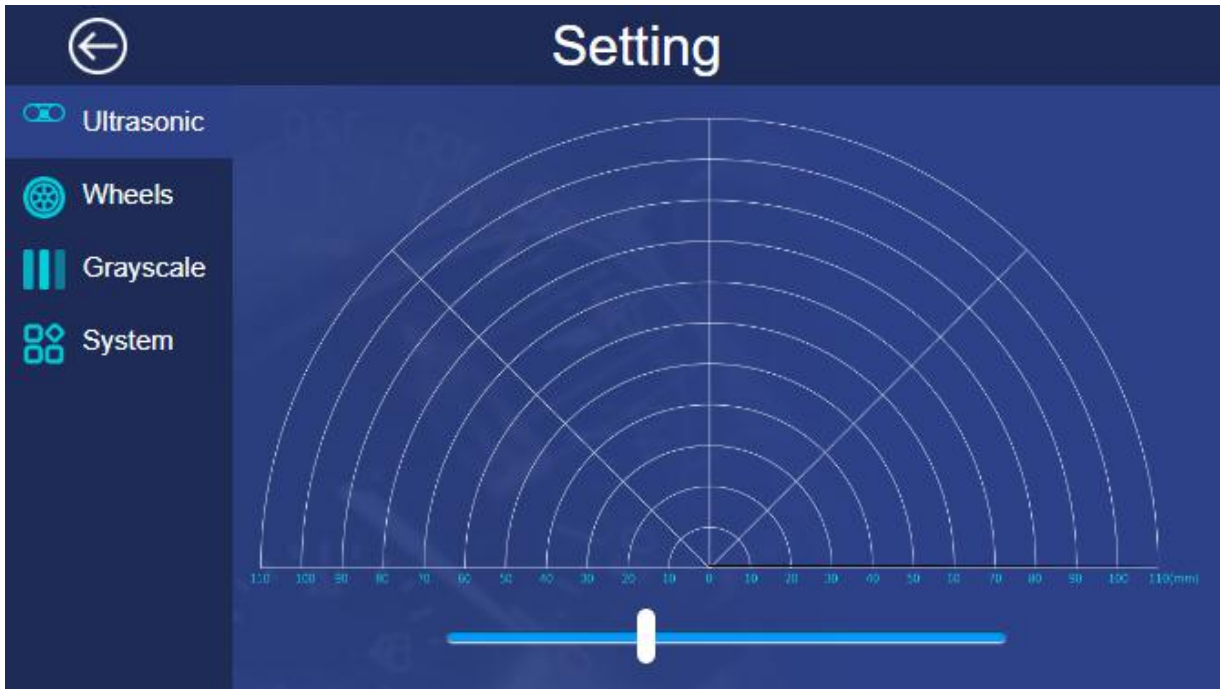
⑥. Path: to turn on/off line following function. Once you start up the line following function, the car will drive along the black lines on the ground. If not, please adjust the threshold on the setting page.

⑦. Setting: You can enter the Setting page by clicking this one.

Setting Page

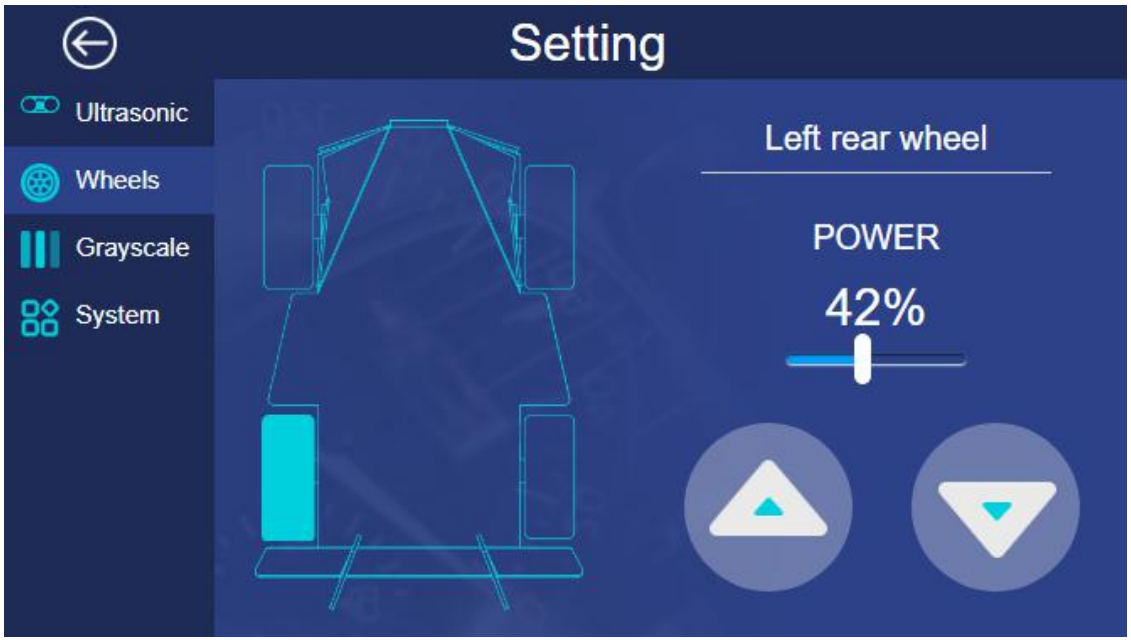
Ultrasonic

This page is used to test whether the ultrasonic sensor works well. The slider below can be used to adjust the current direction and angle of the ultrasonic sensor.



Wheels

You can test the working condition of the wheels on this page.



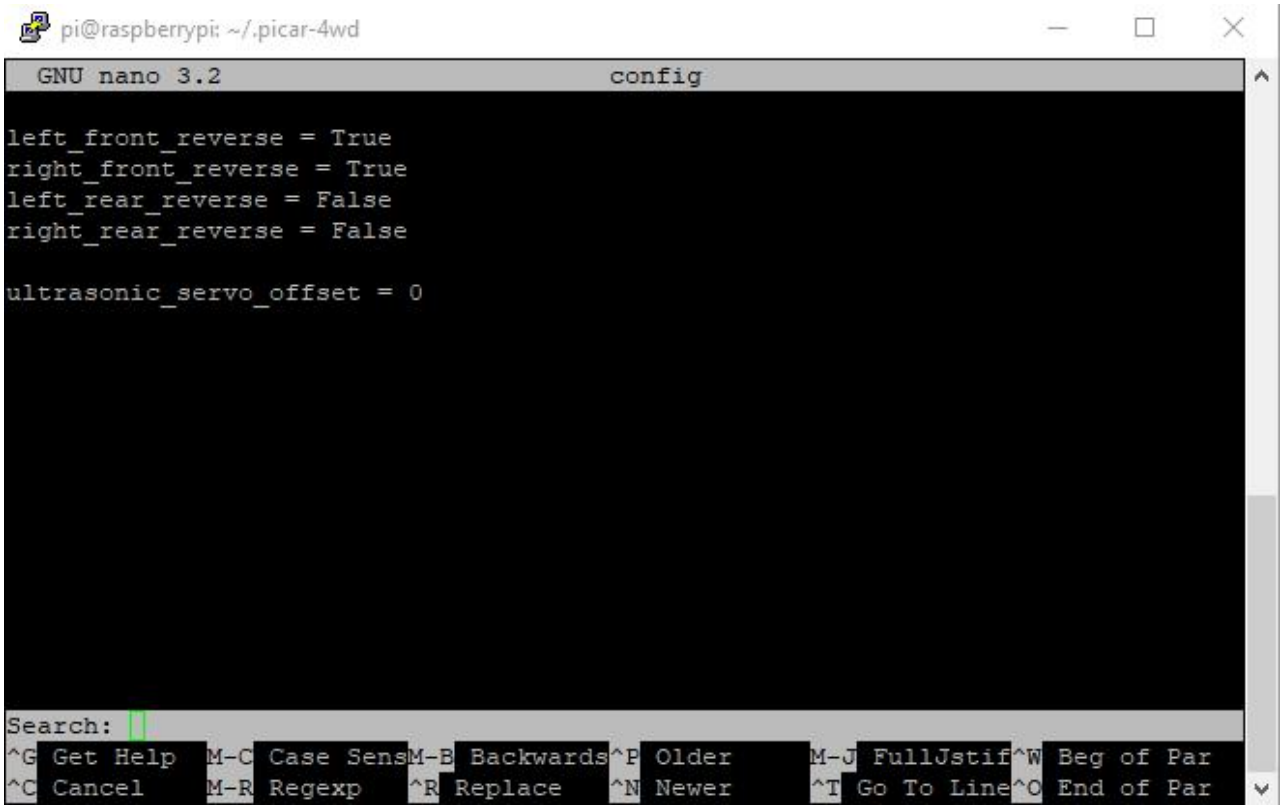
Click a wheel then set the POWER, click the two buttons in bottom right corner. When you press the button at left, the wheel goes forward; press the button on the right, the wheel go backward. If the car doesn't follow your operation like that, you should adjust the direction. The methods are as follows:

1) Input the following command to open the configure file.

```
cd ..
ls -a
```

```
cd .picar-4wd/
nano config
```

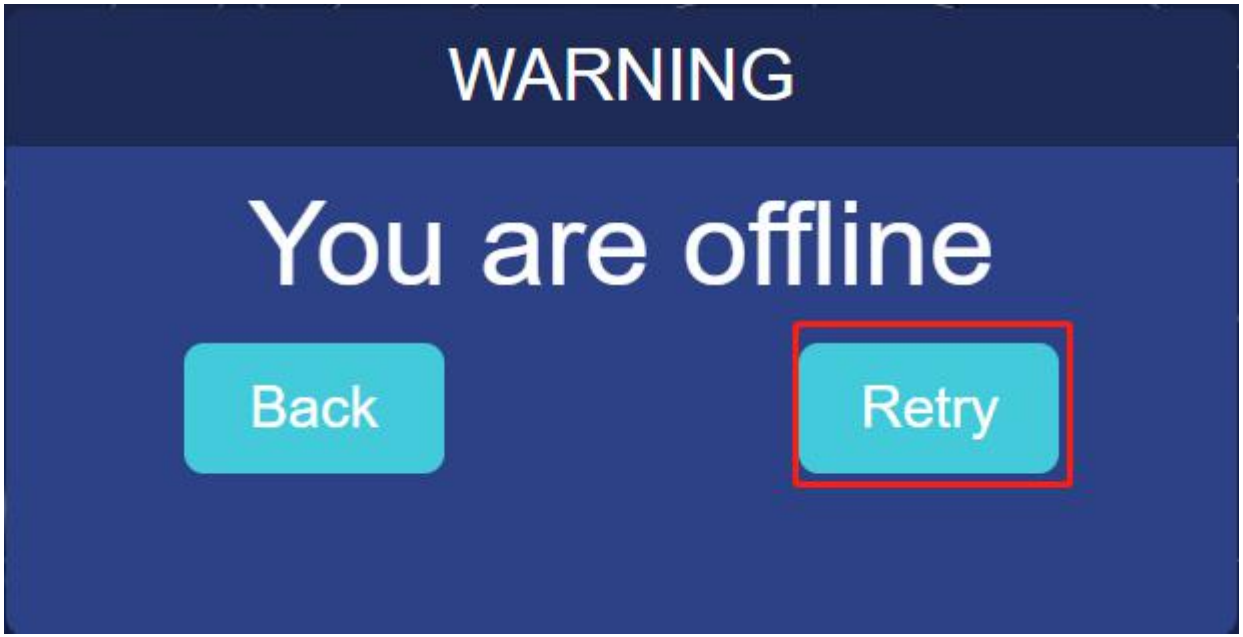
2) In the configure file, the first 4 lines of codes are used to control the turning of the motor. The last line is used to adjust the angle of the servo. For example, if in the above test, the direction of the motor at bottom left is reverse, so we change the code in the third line to left_rear_reverse = **True**.



3) Rerun the following commands.

```
cd /home/pi/picar-4wd/
picar-4wd web-example
```

4) In the browser, click Retry, and you can test the direction of the wheel ready to be used.



Grayscale

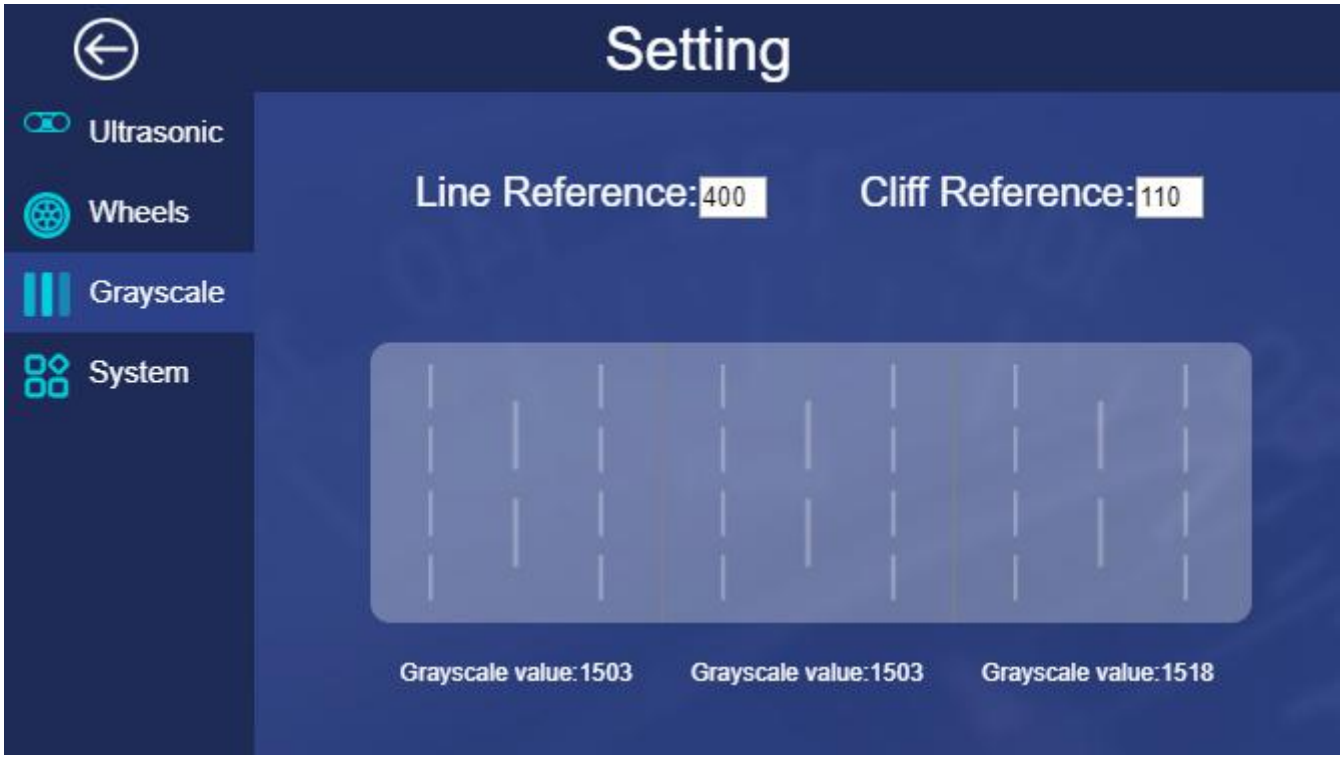
On this page, you can adjust the threshold of the line following and cliff detection.

Line Reference

Put the Picar-4wd on the white surface and record these three values of Grayscale. Now the value of Grayscale is large enough, about 1500. Then put the grayscale sensor module of Picar-4wd above the black surface and record these three values of Grayscale. Now, the value of Grayscale gets relatively small, about 150 as is predicted. So we set the value of Line Reference to 400. Of course, you can change the value. Once you finish doing this, you can see that the corresponding area turning into gray if the value of Grayscale is larger than 400; if not so, turning into black.

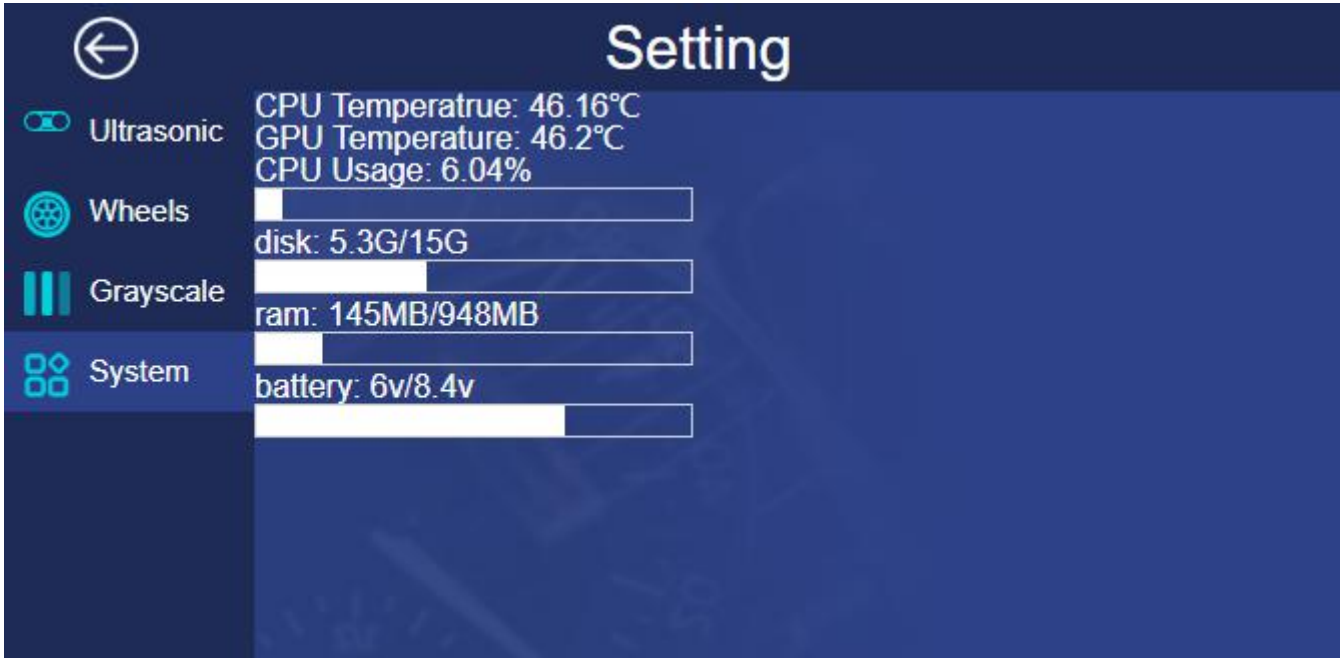
Cliff Reference

Put the car at the edge of the desk with the car's head sticking out from the desk. Record three values of Grayscale, and now the value is relatively small, less than 100. Please set the value of Cliff Reference to 110. When you finish doing this, the corresponding area appears red exclamation symbol if the value of Grayscale is less than 110.


Setting
Ultrasonic
Wheels
Grayscale
System
Line Reference:400
Cliff Reference:110
Grayscale value:1503
Grayscale value:1503
Grayscale value:1518

System

You can check the operating condition of the system on this page.



① Make the example program run automatically at startup.

```
picar-4wd web-example enable
```

② If you want to turn off the Start on Boot, please type in the following command.

```
picar-4wd web-example disable
```

Python Program Control

Python Example Program

Python Example Program is in the examples folder.

```
cd /home/pi/picar-4wd/examples
```

You can run them by using Python3.

- **keyboard_control.py**

After running the example, press「W」, 「A」, 「S」, 「D」, and you can get the car going ahead, back, left, right. And the keys「4」and「6」can increase or decrease the power of the motor.

Run it by using the following command.

```
python3 keyboard_control.py
```

- **obstacle_avoidance.py**

After running the example, the car goes ahead automatically.

It can also turn right to get around the obstacles if there is anything obstructive ahead.

```
python3 obstacle_avoidance.py
```

- **track_line.py**

If you paste a black lane (width: around 2.5cm) on the white floor, the car will move along the lane.

```
python3 track_line.py
```

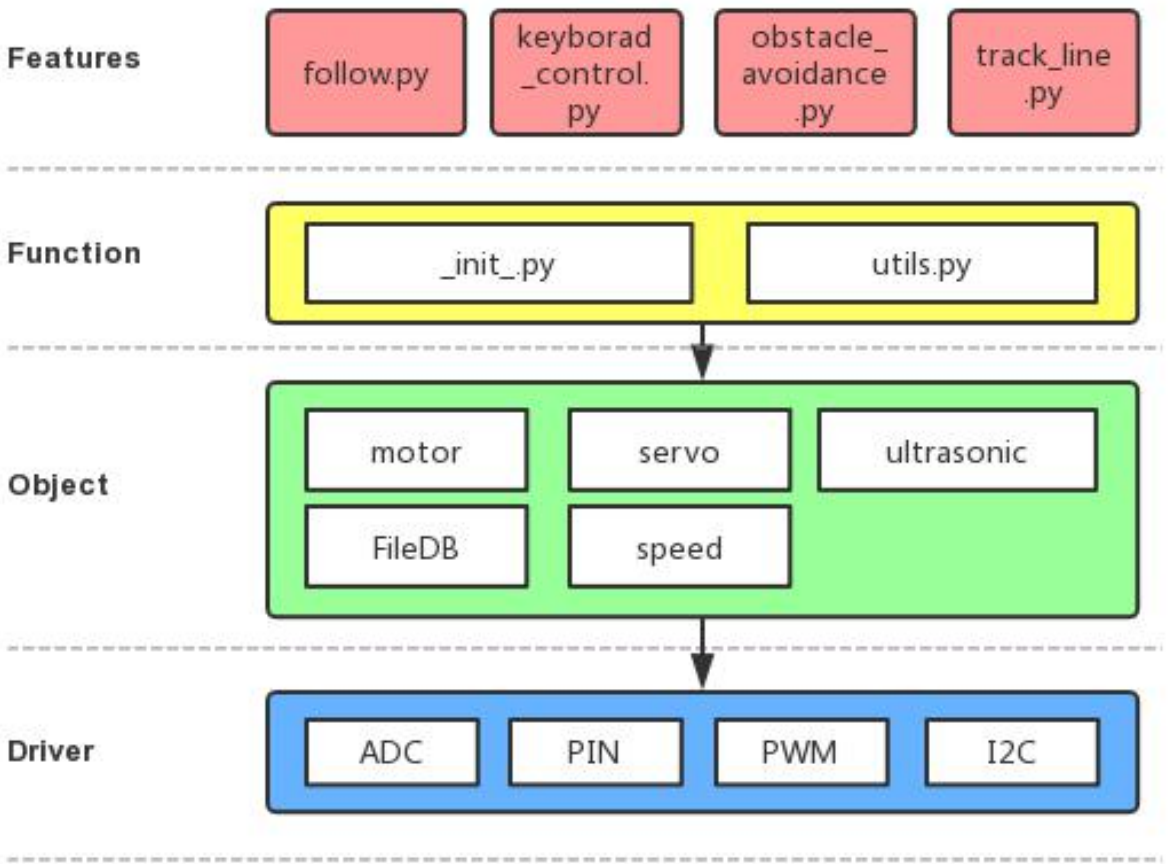
- **follow.py**

When the example runs, the car will follow the object ahead.

```
python3 follow.py
```

Python Example Code Analysis

Program Structure Framework



keyboard_control.py

In the code, we use readchar and readkey to get the input state of the key, then we assign some functions to the keys by using if statement.

```
import picar_4wd as fc
import sys
import tty
import termios
import asyncio

power_val = 50
key = 'status'

def readchar():
    ……

def readkey(getchar_fn=None):
    getchar = getchar_fn or readchar
    ……

def Keyborad_control():
    while True:
        global power_val
        key=readkey()
        if key=='6':
            ……
        elif key=='4':
            ……
        if key=='w':
            ……
        elif key=='a':
            ……
```

```
        elif key=='s':
            ……
        elif key=='d':
            ……
        else:
            ……
        if key=='q':
            print("quit")
            break
if __name__ == '__main__':
    Keyborad_control()
```

obstacle_avoidance.py

In the code, the distance measurement radar (ultrasonic + servo) will take sample for 11 times in total (once every 18°). Then it makes judgement on the distance states of the fourth and the seventh sampling. If radar indicates no obstacle(the returned value is 2), the car will continue moving ahead, or else the car turn right to get around the obstacle.

Note: Check 「scan_step()」, 「get_status_at()」 in the 「_init_.py」for more details about the judgement on radar sampling and distance state.

```
import picar_4wd as fc
speed = 30
def main():
    while True:
        scan_list = fc.scan_step(35)
        if not scan_list:
            continue

        tmp = scan_list[3:7]
```

```
        print(tmp)
        if tmp != [2,2,2,2]:
            fc.turn_right(speed)
        else:
            fc.forward(speed)

if __name__ == "__main__":
    try:
        main()
    finally:
        fc.stop()
```

track_line.py

In this code, grayscale sensor module will judge the gray level of the floor under the car's head. The car adjusts the directions according to the detection values of these three detectors to follow the black line.

Note: About Grayscale judgement, refer to 「get_line_status()」in 「_init_.py」for more details.

```
import picar_4wd as fc

Track_line_speed = 20

def Track_line():
    gs_list = fc.get_grayscale_list()
    if fc.get_line_status(400,gs_list) == 0:
        fc.forward(Track_line_speed)
    elif fc.get_line_status(400,gs_list) == -1:
        fc.turn_left(Track_line_speed)
    elif fc.get_line_status(400,gs_list) == 1:
        fc.turn_right(Track_line_speed)

if __name__=='__main__':
```

```
    while True:
        Track_line()
```

follow.py

In the code, the ultrasonic module will take sample to form an array containing 11 elements; the module detects once every 18° and puts the distance state (0，1，2) of every angle into the array. If there is something obstructive at the right front, the detection value is [22222211122].

The array takes the 2 as a delimiter to divide the array anew. The examples above will be divided into several null elements and a [111]- value element.

The corresponding angular range of the element with a value [111] is 108°~144° (0°+18°*6=108°, 0°+18°*8=144°), and the car will turn its angle into a median 126° （(144+108)/2)

Judge the distance, and keep the car away from the object about 10cm.

Note: Refer to 「scan_step()」and「get_status_at()」in 「_init_.py」to get more details about radar sampling and the distance state judgement.

```
import picar_4wd as fc
speed = 30
def main():
    while True:
        scan_list = fc.scan_step(23)
        if not scan_list:
            continue
        scan_list = [str(i) for i in scan_list]
        scan_list = "".join(scan_list)
        paths = scan_list.split("2")
```

```
        length_list = []
        for path in paths:
            length_list.append(len(path))
        if max(length_list) == 0:
            fc.stop()
        else:
            i = length_list.index(max(length_list))
            pos = scan_list.index(paths[i])
            pos += (len(paths[i]) - 1) / 2
            delta = len(scan_list) / 3
            if pos < delta:
                fc.turn_left(speed)
            elif pos > 2 * delta:
                fc.turn_right(speed)
            else:
                if scan_list[int(len(scan_list)/2-1)] == "0":
                    fc.backward(speed)
                else:
                    fc.forward(speed)

if __name__ == "__main__":
    try:
        main()
    finally:
        fc.stop()
```