Welcome

Thank you for choosing Freenove products!

About Battery

First, read the document **About_Battery.pdf** in the unzipped folder.
If you did not download the zip file, please download it and unzip it via link below.
https://github.com/Freenove/Freenove_Robot_Dog_Kit_for_Raspberry_Pi/archive/master.zip

Get Support and Offer Input

Freenove provides free and responsive product and technical support, including but not limited to:

- Product quality issues
- Product use and build issues
- Questions regarding the technology employed in our products for learning and education
- Your input and opinions are always welcome
- We also encourage your ideas and suggestions for new products and product improvements

For any of the above, you may send us an email to:

support@freenove.com

Safety and Precautions

Please follow the following safety precautions when using or storing this product:

- Keep this product out of the reach of children under 6 years old.
- This product should be used only when there is adult supervision present as young children lack necessary judgment regarding safety and the consequences of product misuse.
- This product contains small parts and parts, which are sharp. This product contains electrically conductive parts. Use caution with electrically conductive parts near or around power supplies, batteries and powered (live) circuits.
- When the product is turned ON, activated or tested, some parts will move or rotate. To avoid injuries to hands and fingers, keep them away from any moving parts!
- It is possible that an improperly connected or shorted circuit may cause overheating. Should this happen, immediately disconnect the power supply or remove the batteries and do not touch anything until it cools down! When everything is safe and cool, review the product tutorial to identify the cause.
- Only operate the product in accordance with the instructions and guidelines of this tutorial, otherwise parts may be damaged or you could be injured.
- Store the product in a cool dry place and avoid exposing the product to direct sunlight.
- After use, always turn the power OFF and remove or unplug the batteries before storing.

About Freenove

Freenove provides open source electronic products and services worldwide.

Freenove is committed to assist customers in their education of robotics, programming and electronic circuits so that they may transform their creative ideas into prototypes and new and innovative products. To this end, our services include but are not limited to:

- Educational and Entertaining Project Kits for Robots, Smart Cars and Drones
- Educational Kits to Learn Robotic Software Systems for Arduino, Raspberry Pi and micro: bit
- Electronic Component Assortments, Electronic Modules and Specialized Tools
- **Product Development and Customization Services**

You can find more about Freenove and get our latest news and updates through our website:

http://www.freenove.com

Copyright

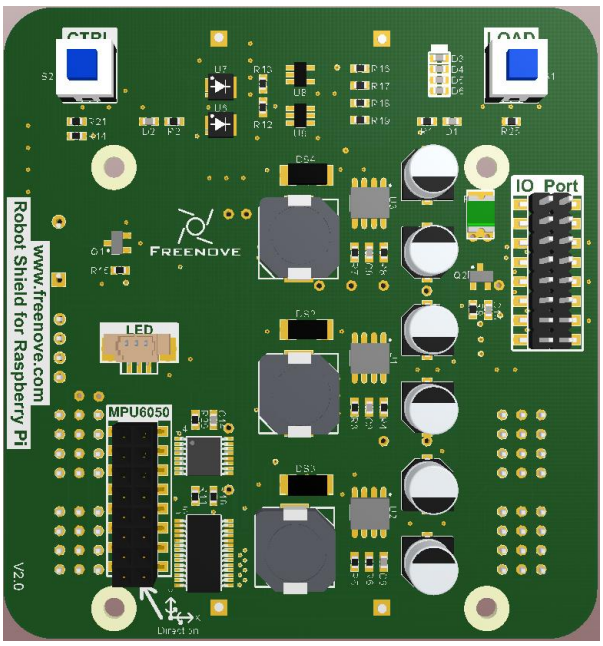
Contents

List

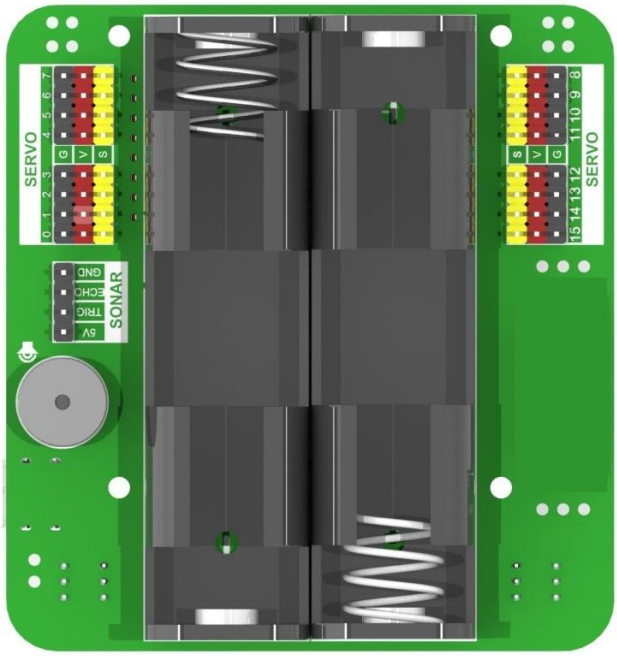
If you have any concerns, please feel free to contact us at support@freenove.com

Robot Shield for Raspberry Pi

Top



Bottom

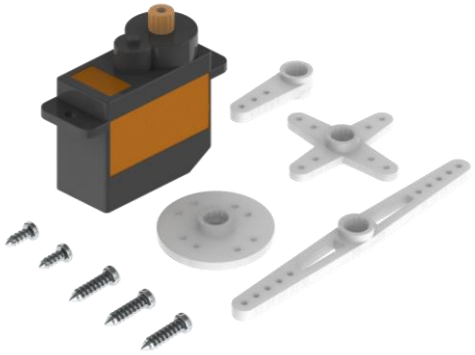
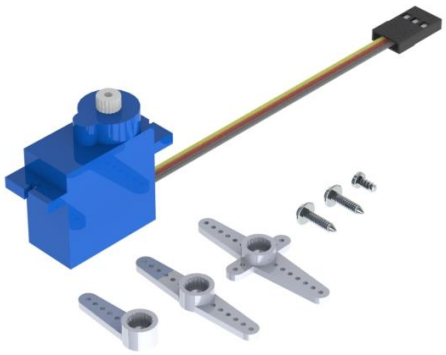


Machinery Parts

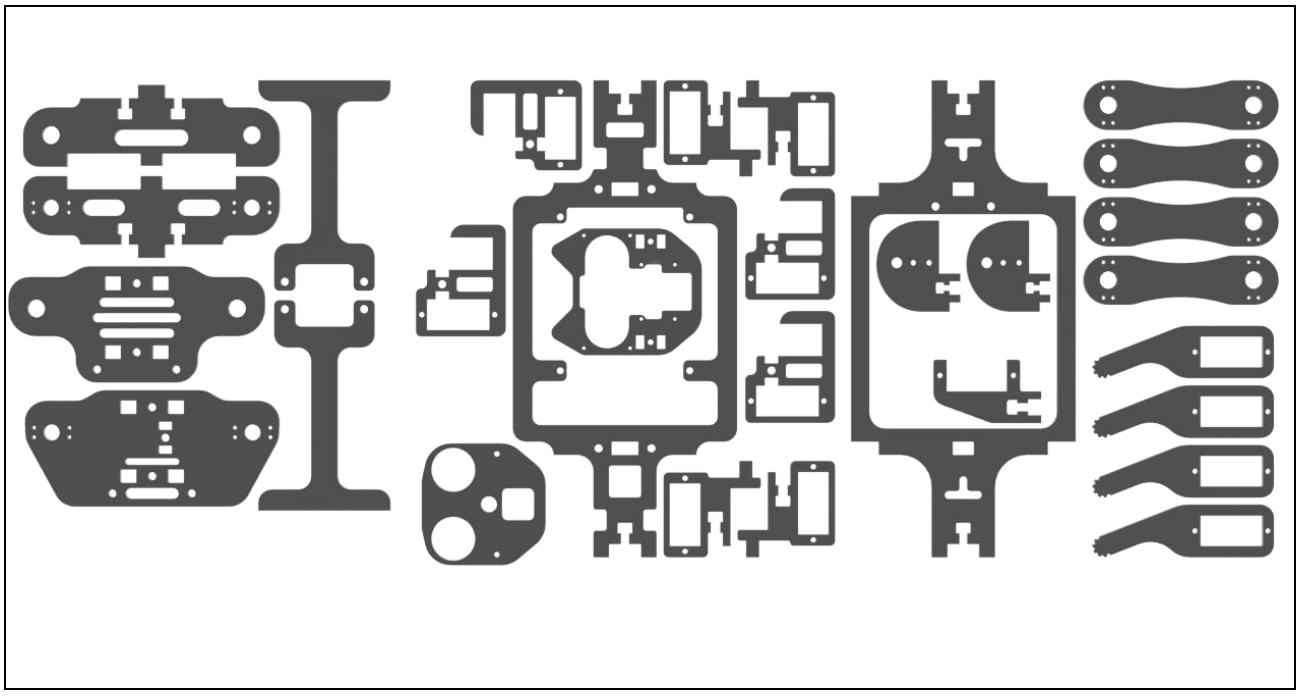
M2.5*14+6 Copper Standoff  x5 Freenove	M2.5*14 Copper Standoff  x5 Freenove	M2.5*8 Screw  x10 Freenove
M2*14 Screw  x32 Freenove	M3*12 Screw  x22 Freenove	M1.2*7 Self-tapping Screw  x60 Freenove
M2 Nut  x32 Freenove	M3 Nut  x22 Freenove	M1.4*4 Self-tapping Screw  x12 Freenove

Note: You may receive M1.4*4 or M1.4*5. Both are suitable.

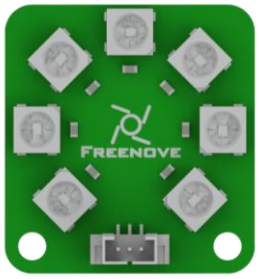
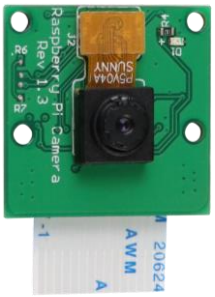
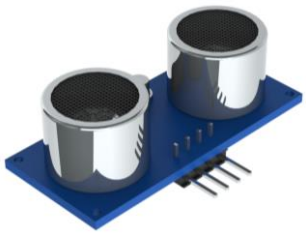
Transmission Parts

ES08MA II servo package x12	S90 servo package x1
	

Acrylic Parts



Electronic Parts and Tools

LED module 	Camera 	HC-SR04 ultrasonic module 	Connector 
Jumper wire F/F(4) for ultrasonic module 			
10cm 3Pin LED cable (same direction) 			
25cm 15Pin camera cable (reversed direction) 			
Cross screwdriver (3mm) x1 Cross screwdriver (2mm) x1 	Cable tidy x80cm 	Red ball 	

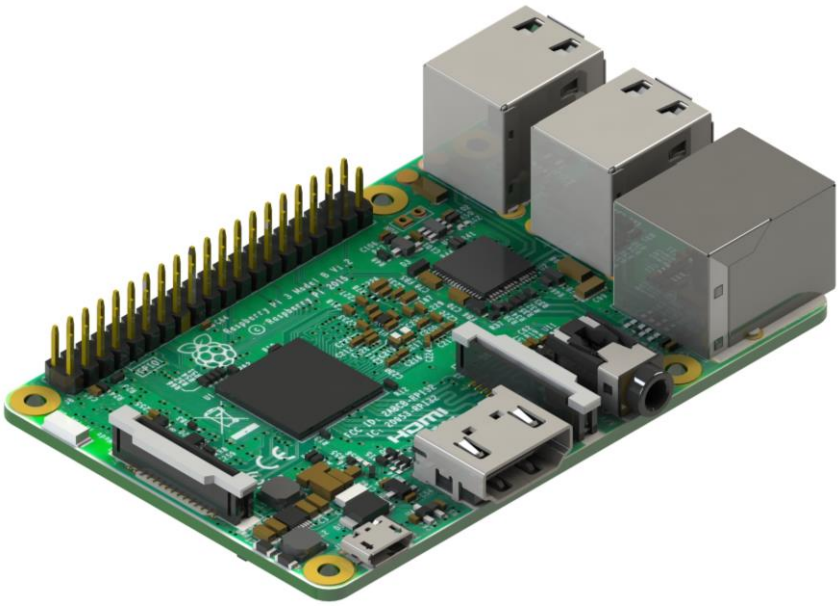
Required but NOT Contained Parts

Two 18650 lithium batteries without protected board.

The continuous discharge current >10A

It is not easy to find proper batteries on Amazon. **Search 18650 3.7V high drain on eBay** or other websites.

Raspberry Pi (Recommended model: Raspberry 4B / 3B+/ 3A+ /3B) x1



Preface

Welcome to use Freenove Robot Dog Kit for Raspberry Pi. Following this tutorial, you can make a very cool robot dog with many functions.

This kit is based on Raspberry Pi, a popular control panel, so you can share and exchange your experience and design ideas with many enthusiasts all over the world. This kit contains all electronic components, modules, and mechanical components required for making the robot dog. And all of them are packaged individually. There are detailed instructions for assembly and configuration in this book.

If you encounter any problems, please feel free to contact us for quick and free technical support.

support@freenove.com

This book can help enthusiasts with little technical knowledge to make a robot dog. If you are very interested in Raspberry Pi, and want to learn how to program and build the circuit, please visit our website www.freenove.com or contact us to buy the kits designed for beginners:
Freenove Basic\LCD1602\Super\Ultrasonic\RFID\Ultimate Starter Kit for Raspberry Pi

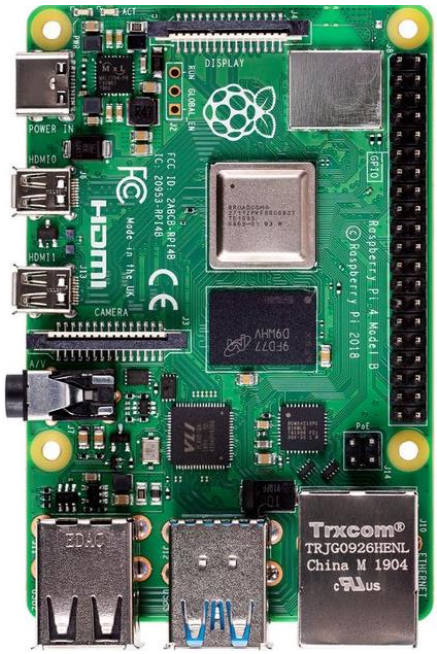
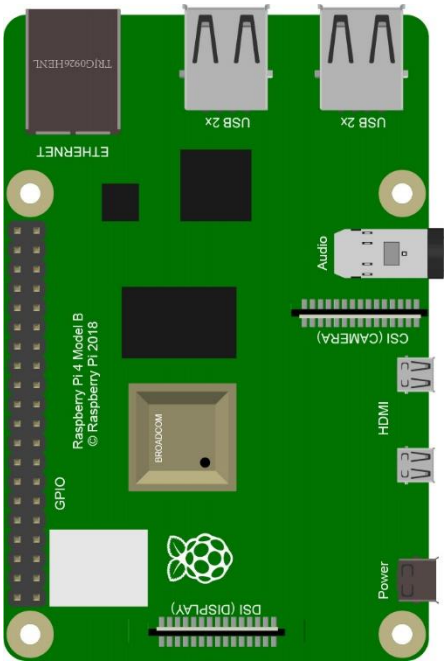
Introduction to Raspberry Pi

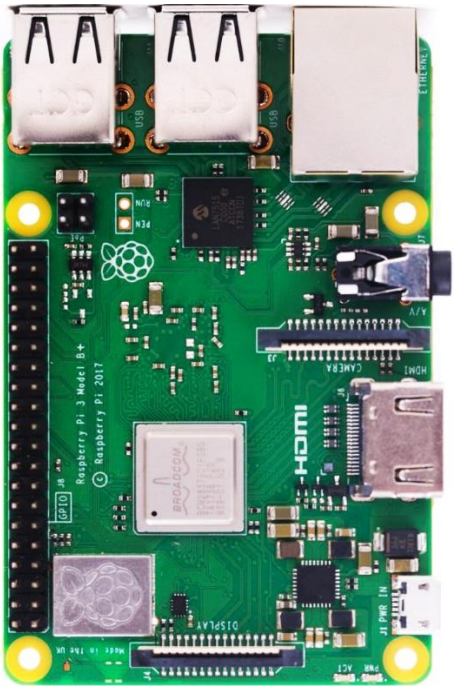
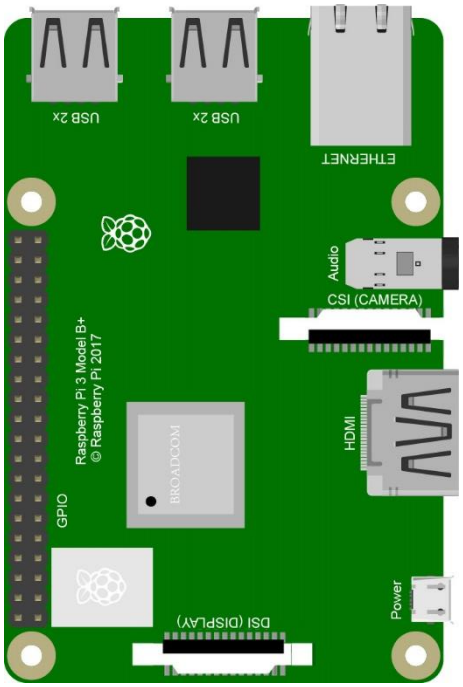
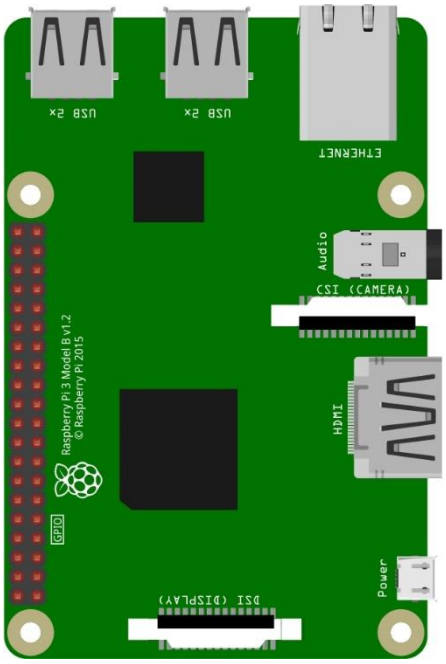
Raspberry Pi (or RPi, RPI, RasPi, which will be also referenced in this tutorial), a micro-computer with size of a card, quickly swept the world since it was launched. It is widely used in desktop workstation, media center, smart household, robots, and even the servers, etc. It can do almost everything, which continues to attract fans to explore it. Raspberry Pi is used to be running with Linux system and along with the release of windows 10 IoT, we can also run it with Windows. Raspberry Pi (with interfaces USB, network, HDMI, camera, audio, display and GPIO), as a microcomputer, can be run in command line mode and desktop system mode. Additionally, it is easy to operate just like Arduino, and you can even directly operate the GPIO of CPU.

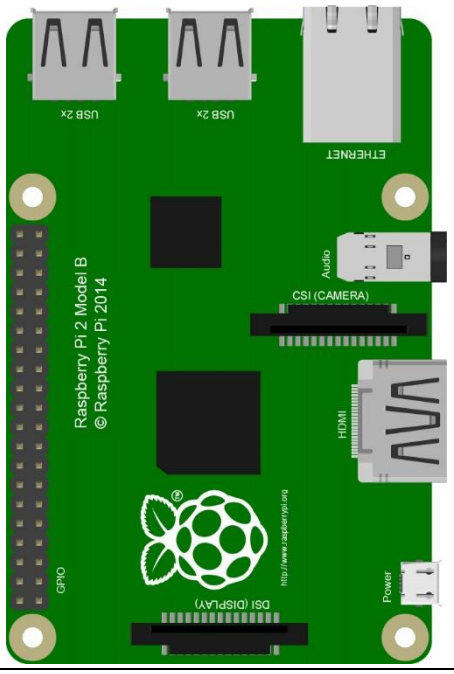
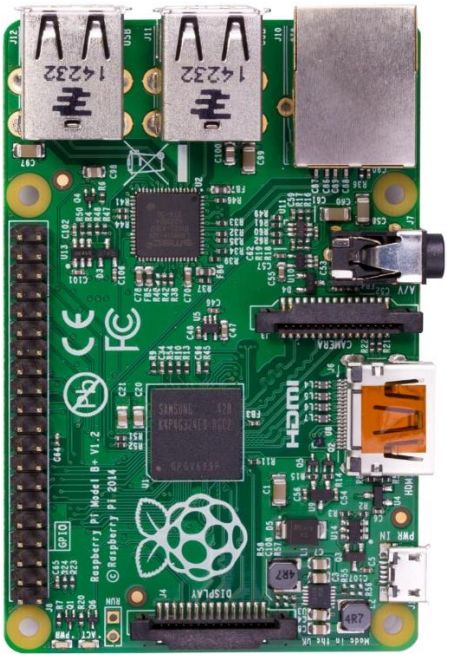
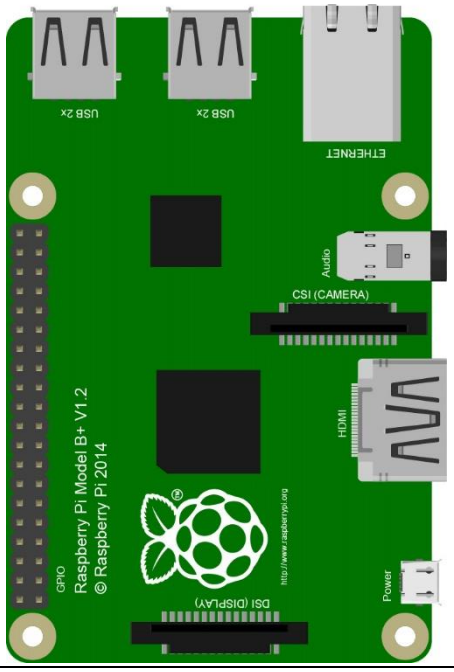
So far, at this writing, Raspberry Pi has advanced to its fourth generation product offering. Version changes are accompanied by increases in upgrades in hardware and capabilities.

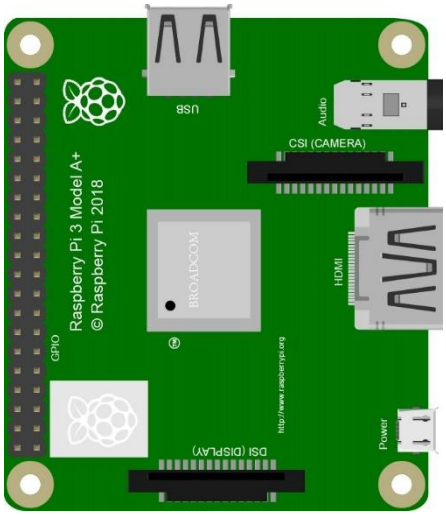
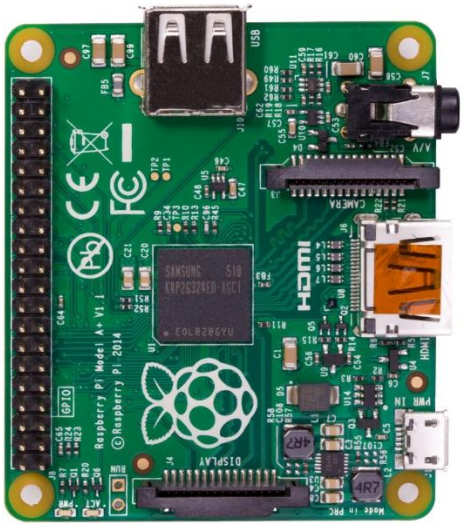
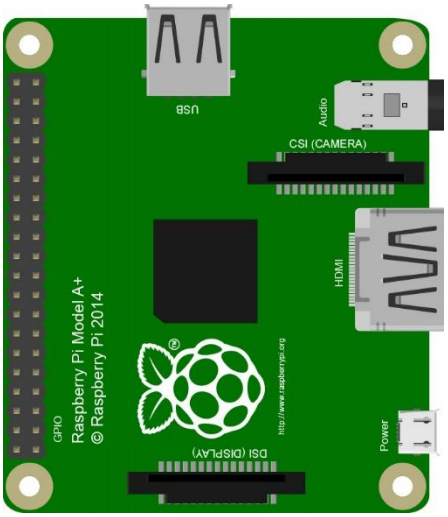
The A type and B type versions of the first generation products have been discontinued due to various reasons. What is most important is that other popular and currently available versions are consistent in the order and number of pins and their assigned designation of function, making compatibility of peripheral devices greatly enhanced between versions.

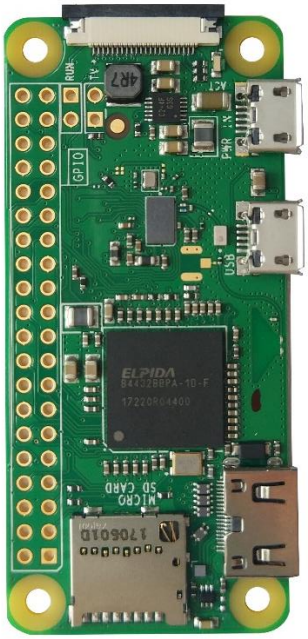
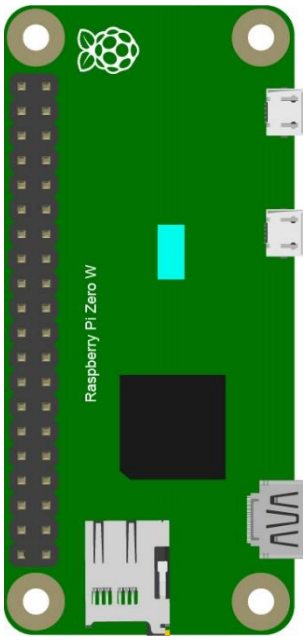
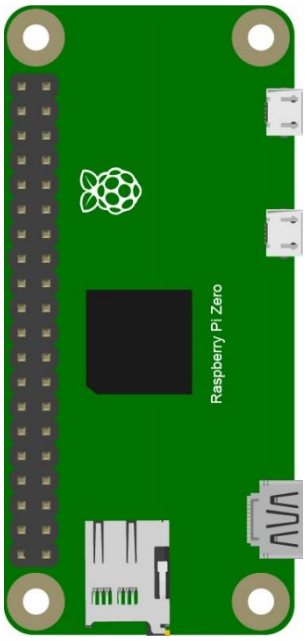
Below are the Raspberry Pi pictures and model pictures supported by this product.

Practicality picture of Raspberry Pi 4 Model B:	Model diagram of Raspberry Pi 4 Model B:
	

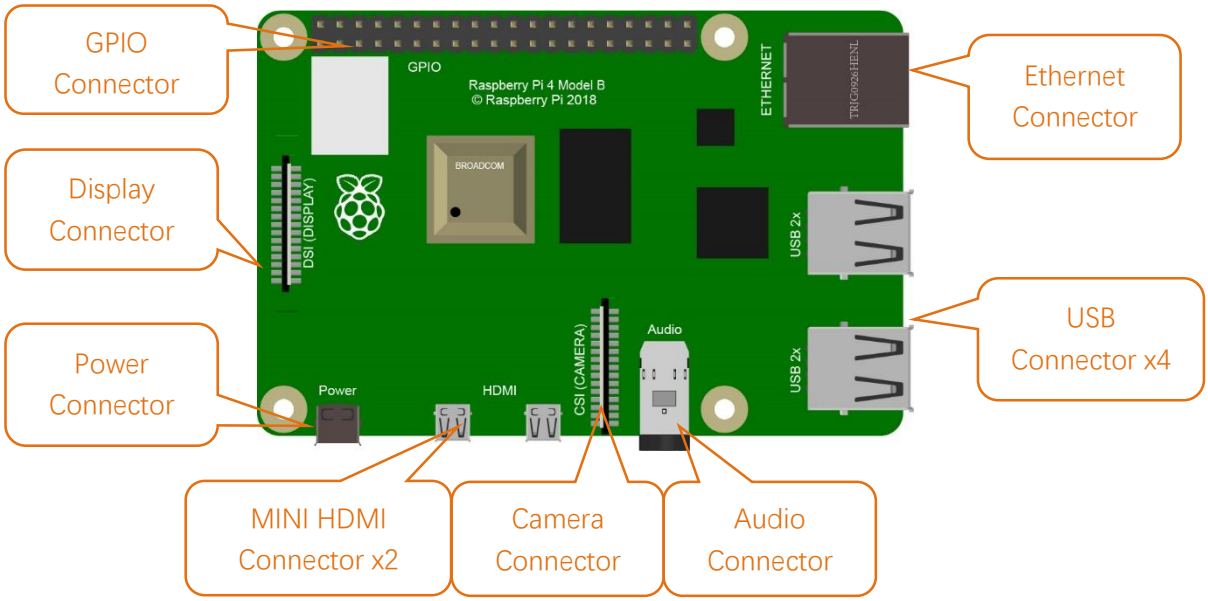
Practicality picture of Raspberry Pi 3 Model B+:	Model diagram of Raspberry Pi 3 Model B+:
	
Practicality picture of Raspberry Pi 3 Model B:	Model diagram of Raspberry Pi 3 Model B:
	

Practicality picture of Raspberry Pi 2 Model B: 	Model diagram of Raspberry Pi 2 Model B: 
Practicality picture of Raspberry Pi 1 Model B+: 	Model diagram of Raspberry Pi 1 Model B+: 

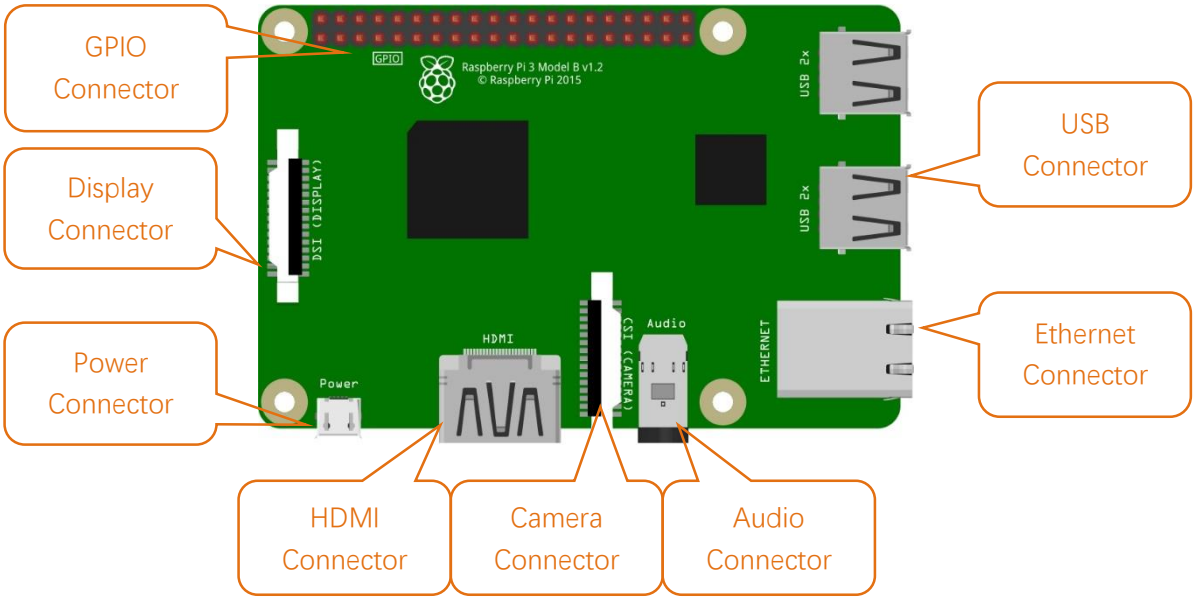
Practicality picture of Raspberry Pi 3 Model A+:	Model diagram of Raspberry Pi 3 Model A+:
	
Practicality picture of Raspberry Pi 1 Model A+:	Model diagram of Raspberry Pi 1 Model A+:
	

Practicality picture of Raspberry Pi Zero W: 	Model diagram of Raspberry Pi Zero W: 
Practicality picture of Raspberry Pi Zero: 	Model diagram of Raspberry Pi Zero: 

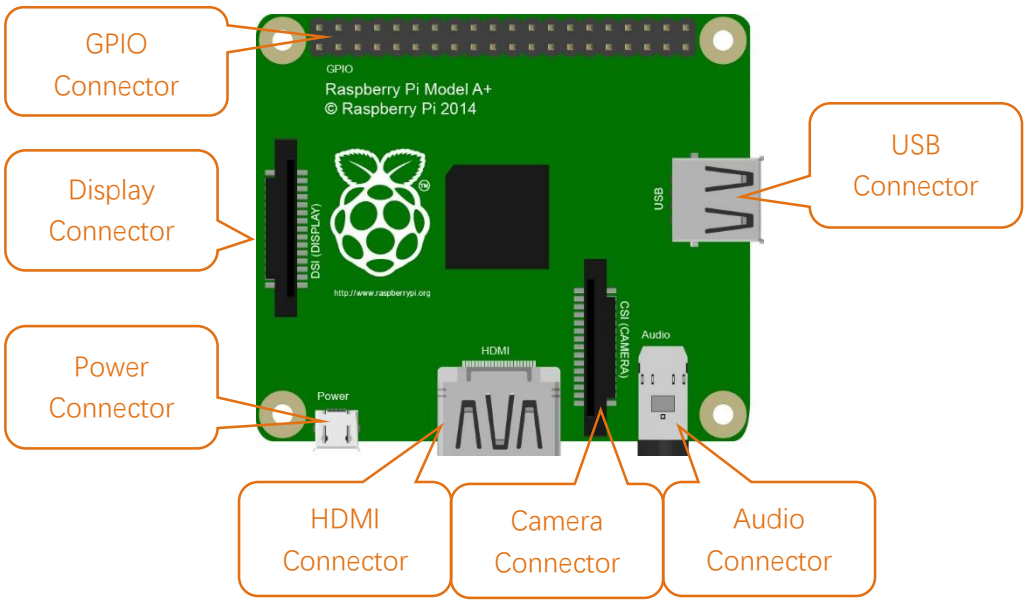
Hardware interface diagram of RPi 4B is shown below:



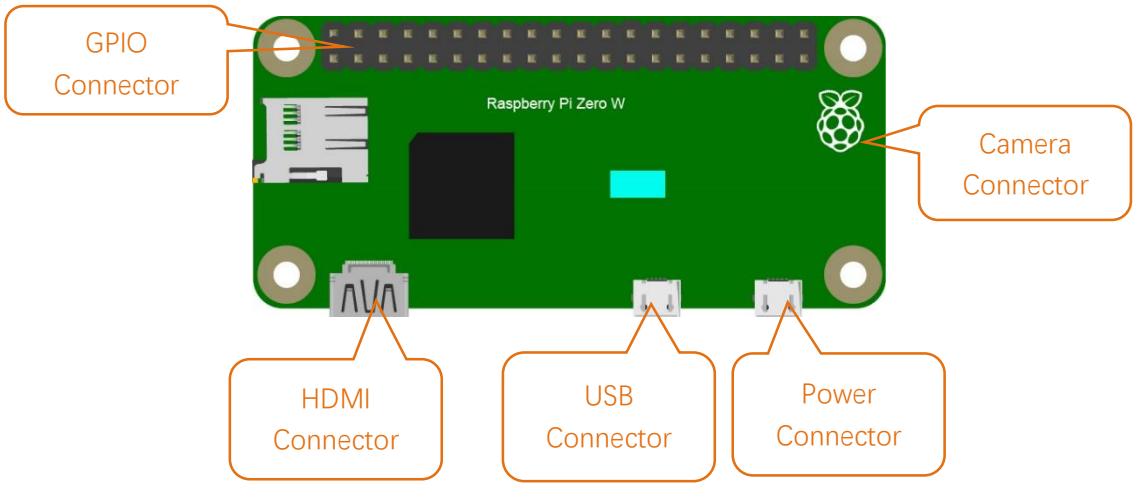
Hardware interface diagram of RPi 3B+/3B/2B/1B+ are shown below:



Hardware interface diagram of RPi 3A+/A+ is shown below:



Hardware interface diagram of RPi Zero/Zero W is shown below:

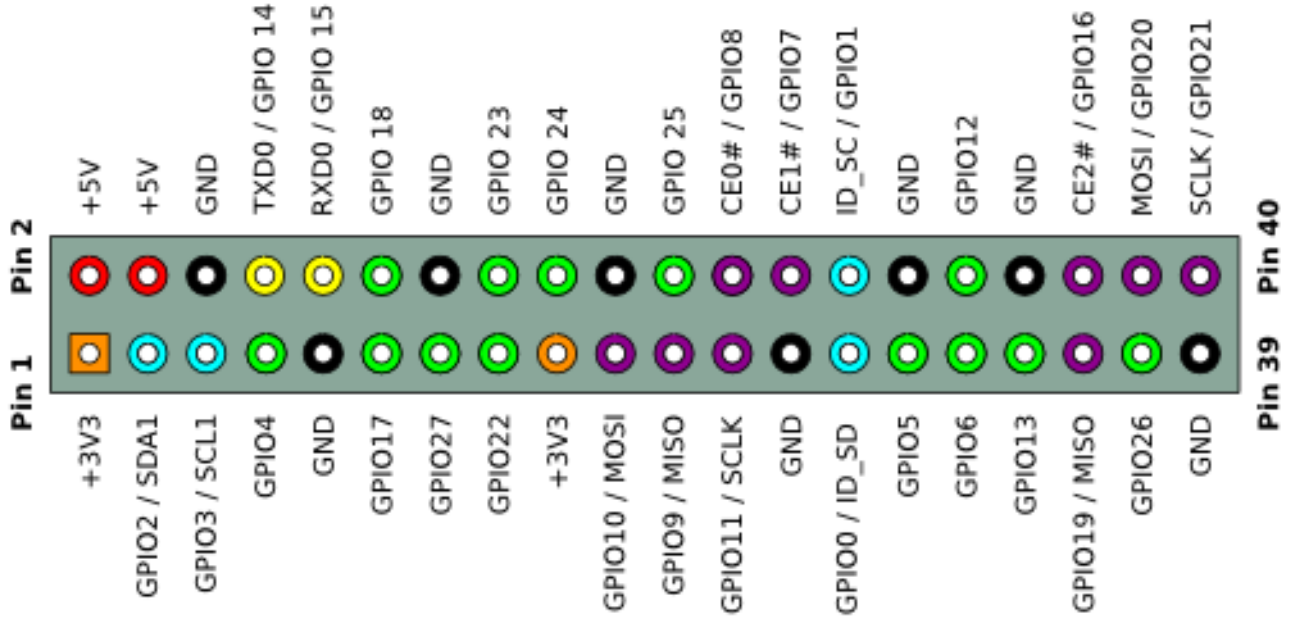


GPIO

GPIO: General purpose input/output. We will introduce the specific feature of the pins on the Raspberry Pi and how you can utilize them in all sorts of ways in your projects. Most RPi Module pins can be used as either an input or output, depending on your program and its functions. When programming the GPIO pins, there are 3 different ways to reference them: GPIO numbering, physical numbering, WiringPi GPIO Numbering.

BCM GPIO Numbering

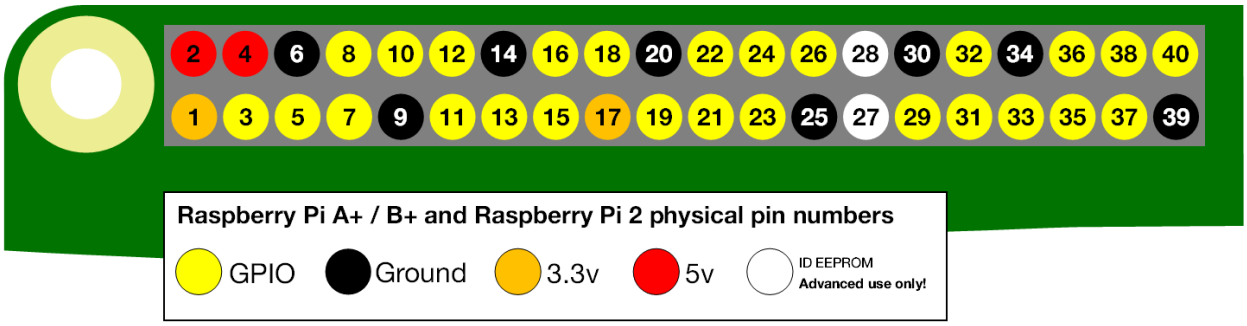
The Raspberry Pi CPU uses Broadcom (BCM) processing chips BCM2835, BCM2836 or BCM2837. GPIO pin numbers are assigned by the processing chip manufacturer and are how the computer recognizes each pin. The pin numbers themselves do not make sense or have meaning as they are only a form of identification. Since their numeric values and physical locations have no specific order, there is no way to remember them, so you will need to have a printed reference or a reference board that fits over the pins.
Each pin is defined as below:



For more details about pin definition of GPIO, please refer to http://pinout.xyz/

PHYSICAL Numbering

Another way to refer to the pins is by simply counting across and down from pin 1 at the top left (nearest to the SD card). This is 'physical numbering', as shown below:



WiringPi GPIO Numbering

Different from the previous two types of GPIO serial numbers, RPi GPIO serial number of the WiringPi are numbered according to the BCM chip use in RPi.

wiringPi Pin	BCM GPIO	Name	Header	Name	BCM GPIO	wiringPi Pin
—	—	3.3v	1 | 2	5v	—	—
8	R1:0/R2:2	SDA	3 | 4	5v	—	—
9	R1:1/R2:3	SCL	5 | 6	0v	—	—
7	4	GPIO7	7 | 8	TxD	14	15
—	—	0v	9 | 10	RxD	15	16
0	17	GPIO0	11 | 12	GPIO1	18	1
2	R1:21/R2:27	GPIO2	13 | 14	0v	—	—
3	22	GPIO3	15 | 16	GPIO4	23	4
—	—	3.3v	17 | 18	GPIO5	24	5
12	10	MOSI	19 | 20	0v	—	—
13	9	MISO	21 | 22	GPIO6	25	6
14	11	SCLK	23 | 24	CE0	8	10
—	—	0v	25 | 26	CE1	7	11
30	0	SDA.0	27 | 28	SCL.0	1	31
21	5	GPIO.21	29 | 30	0V		
22	6	GPIO.22	31 | 32	GPIO.26	12	26
23	13	GPIO.23	33 | 34	0V		
24	19	GPIO.24	35 | 36	GPIO.27	16	27
25	26	GPIO.25	37 | 38	GPIO.28	20	28
		0V	39 | 40	GPIO.29	21	29
wiringPi Pin	BCM GPIO	Name	Header	Name	BCM GPIO	wiringPi Pin

For Pi B

For A+, B+, 2B, 3B, 3B+, 4B, Zero

(For more details, please refer to https://projects.drogon.net/raspberry-pi/wiringpi/pins/)

You can also use the following command to view their correlation.

```
gpio readall
```

```
pi@raspberrypi:~ $ gpio readall
 +-----+-----+---------+------+---+---Pi 3---+---+------+---------+-----+-----+
 | BCM | wPi |   Name  | Mode | V | Physical | V | Mode | Name    | wPi | BCM |
 +-----+-----+---------+------+---+----++----+---+------+---------+-----+-----+
 |     |     |    3.3v |      |   |  1 || 2  |   |      | 5v      |     |     |
 |   2 |   8 |   SDA.1 | ALT0 | 1 |  3 || 4  |   |      | 5V      |     |     |
 |   3 |   9 |   SCL.1 | ALT0 | 1 |  5 || 6  |   |      | 0v      |     |     |
 |   4 |   7 | GPIO. 7 |   IN | 1 |  7 || 8  | 1 | ALT5 | TxD     | 15  | 14  |
 |     |     |      0v |      |   |  9 || 10 | 1 | ALT5 | RxD     | 16  | 15  |
 |  17 |   0 | GPIO. 0 |   IN | 0 | 11 || 12 | 0 | IN   | GPIO. 1 | 1   | 18  |
 |  27 |   2 | GPIO. 2 |   IN | 0 | 13 || 14 |   |      | 0v      |     |     |
 |  22 |   3 | GPIO. 3 |   IN | 0 | 15 || 16 | 0 | IN   | GPIO. 4 | 4   | 23  |
 |     |     |    3.3v |      |   | 17 || 18 | 0 | IN   | GPIO. 5 | 5   | 24  |
 |  10 |  12 |    MOSI | ALT0 | 0 | 19 || 20 |   |      | 0v      |     |     |
 |   9 |  13 |    MISO | ALT0 | 0 | 21 || 22 | 0 | IN   | GPIO. 6 | 6   | 25  |
 |  11 |  14 |    SCLK | ALT0 | 0 | 23 || 24 | 1 | OUT  | CE0     | 10  | 8   |
 |     |     |      0v |      |   | 25 || 26 | 1 | OUT  | CE1     | 11  | 7   |
 |   0 |  30 |   SDA.0 |   IN | 1 | 27 || 28 | 1 | IN   | SCL.0   | 31  | 1   |
 |   5 |  21 | GPIO.21 |   IN | 1 | 29 || 30 |   |      | 0v      |     |     |
 |   6 |  22 | GPIO.22 |   IN | 1 | 31 || 32 | 0 | IN   | GPIO.26 | 26  | 12  |
 |  13 |  23 | GPIO.23 |   IN | 0 | 33 || 34 |   |      | 0v      |     |     |
 |  19 |  24 | GPIO.24 |   IN | 0 | 35 || 36 | 0 | IN   | GPIO.27 | 27  | 16  |
 |  26 |  25 | GPIO.25 |   IN | 0 | 37 || 38 | 0 | IN   | GPIO.28 | 28  | 20  |
 |     |     |      0v |      |   | 39 || 40 | 0 | IN   | GPIO.29 | 29  | 21  |
 +-----+-----+---------+------+---+----++----+---+------+---------+-----+-----+
 | BCM | wPi |   Name  | Mode | V | Physical | V | Mode | Name    | wPi | BCM |
 +-----+-----+---------+------+---+---Pi 3---+---+------+---------+-----+-----+
```

For more details about wiringPi, please refer to http://wiringpi.com/ .

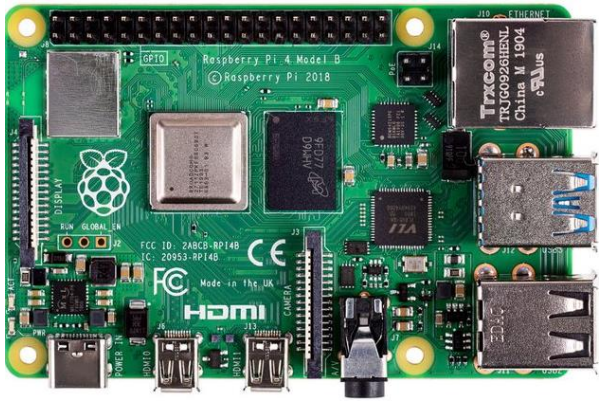
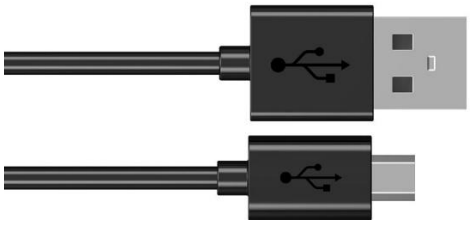
Chapter 0 Raspberry Pi Preparation

Install a System

Firstly, install a system for your RPi.

Component List

Required Components

Raspberry Pi 4B / 3B+/ 3B /3A+ (Recommended) 	5V/3A Power Adapter. Different versions of Raspberry Pi have different power requirements. 
Micro USB Cable x1 	Micro SD Card (TF Card) x1, Card Reader x1 

This robot also supports the following versions of the Raspberry Pi, but **additional accessories** need to be prepared by yourself.

Raspberry	Additional accessories
Raspberry Pi Zero W	Camera cable(>25cm) for zero w, 15 Pin 1.0mm Pitch to 22 Pin 0.5mm https://www.amazon.com/dp/B076Q595HJ/
Raspberry Pi Zero 1.3	wireless network adapter, Camera cable(>25cm) for zero w, 15 Pin 1.0mm Pitch to 22 Pin 0.5mm, OTG cable (USB Type micro B to USB Type A)
Raspberry Pi 2 Model B	wireless network adapter
Raspberry Pi 1 Model A+	wireless network adapter
Raspberry Pi 1 Model B+	wireless network adapter

Power requirements of various versions of Raspberry Pi are shown in following table:

Product	Recommended PSU current capacity	Maximum total USB peripheral current draw	Typical bare-board active current consumption
Raspberry Pi Model A	700mA	500mA	200mA
Raspberry Pi Model B	1.2A	500mA	500mA
Raspberry Pi Model A+	700mA	500mA	180mA
Raspberry Pi Model B+	1.8A	600mA/1.2A (switchable)	330mA
Raspberry Pi 2 Model B	1.8A	600mA/1.2A (switchable)	350mA
Raspberry Pi 3 Model B	2.5A	1.2A	400mA
Raspberry Pi 3 Model A+	2.5A	Limited by PSU, board, and connector ratings only.	350mA
Raspberry Pi 3 Model B+	2.5A	1.2A	500mA
Raspberry Pi 4 Model B	3.0A	1.2A	600mA
Raspberry Pi Zero W	1.2A	Limited by PSU, board, and connector ratings only.	150mA
Raspberry Pi Zero	1.2A	Limited by PSU, board, and connector ratings only	100mA

For more details, please refer to https://www.raspberrypi.org/help/faqs/#powerReqs

In addition, RPi also needs an Ethernet network cable used to connect it to a WAN (Wide Area Network).
All these components are necessary for any of your projects to work. Among them, the power supply of at least 5V/2.5A, because a lack of a sufficient power supply may lead to many functional issues and even damage your RPi, we STRONGLY RECOMMEND a 5V/2.5A power supply. We also recommend using a SD Micro Card with a capacity 16GB or more (which, functions as the RPI's "hard drive") and is used to store the operating system and necessary operational files.

In future projects, the components list with a RPi will contain these required components, using only RPi as a representative rather than presenting details.

Optional Components

Under normal circumstances, there are two ways to login to Raspberry Pi: 1) Using a stand-alone monitor. 2) Using a remote desktop computer monitor "sharing" the PC monitor with your RPi.

Required Accessories for Monitor

If you want to use an independent monitor, mouse and keyboard, you also need the following accessories.

1. A Display with HDMI interface
2. A Mouse and a Keyboard with an USB interface

As to Pi Zero and Pi Zero W, you also need the following accessories:

1. A Mini-HDMI to HDMI Adapter and Cable.
2. A Micro-USB to USB-A Adapter and Cable (Micro USB OTG Cable).
3. A USB HUB.
4. USB to Ethernet Interface or USB Wi-Fi receiver.

For different Raspberry Pi, the optional items are slightly different. But they all aim to convert the interfaces to Raspberry Pi standards.

Item	Pi Zero	Pi Zero W	Pi A+	Pi 3A+	Pi B+/2B	Pi 3B/3B+/4B
Monitor	Yes	Yes	Yes	Yes	Yes	Yes
Mouse	Yes	Yes	Yes	Yes	Yes	Yes
Keyboard	Yes	Yes	Yes	Yes	Yes	Yes
Micro-HDMI to HDMI cable	Yes	Yes	No	No	No	No
Micro-USB to USB-A OTG cable	Yes	Yes	No	No	No	No
USB HUB	Yes	Yes	Yes	Yes	No	No
USB transferring to Ethernet interface	select one from two or select two from two	optional	select one from two or select two from two	optional	Internal Integration	Internal Integration
USB Wi-Fi receiver		Internal Integration		Internal Integration	optional	

Required Accessories for Remote Desktop

If you do not have an independent monitor, or if you want to use a remote desktop, first you need to login to Raspberry Pi through SSH, then open the VNC or RDP service. This requires the following accessories.

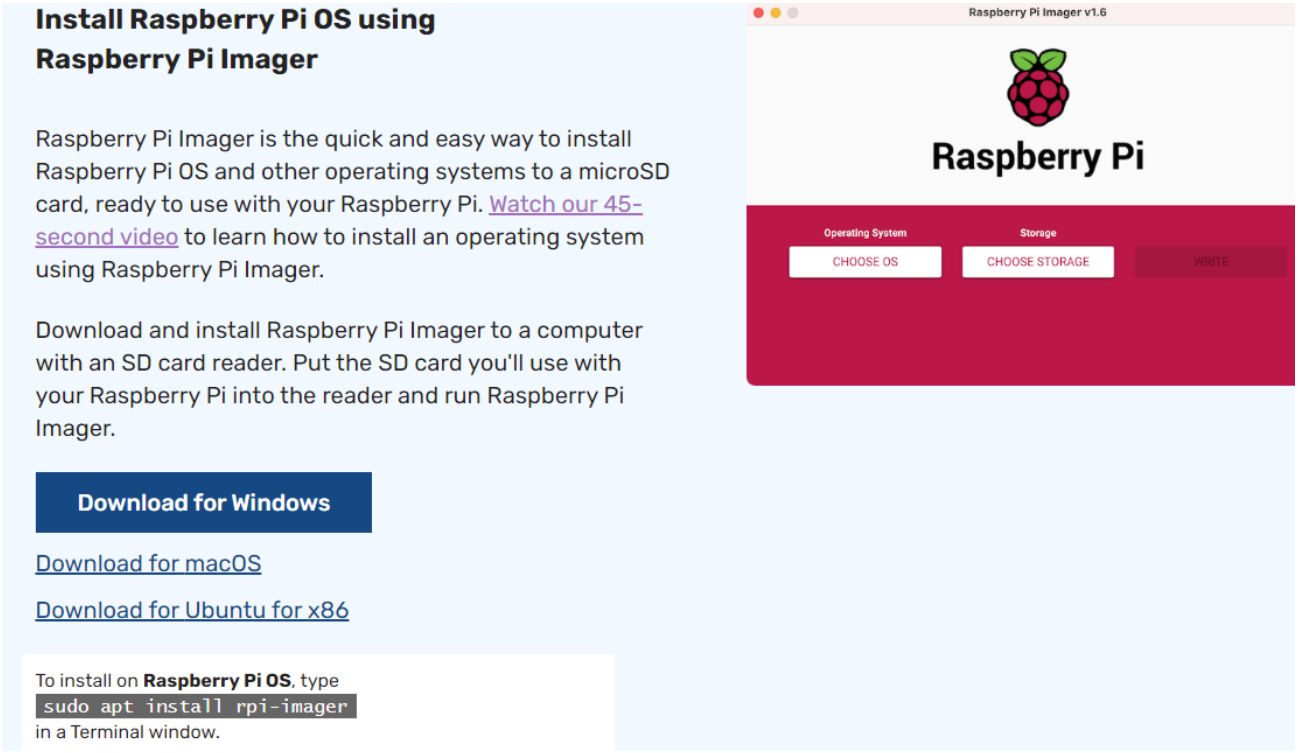
Item	Pi Zero	Pi Zero W	Pi A+	Pi 3A+	Pi B+/2B	Pi 3B/3B+/4B
Micro-USB to USB-A OTG cable	Yes	Yes	No	NO		
USB transferring to Ethernet interface	Yes	Yes	Yes			

Raspberry Pi OS

Install imager tool

Visit this website to install imager tool.

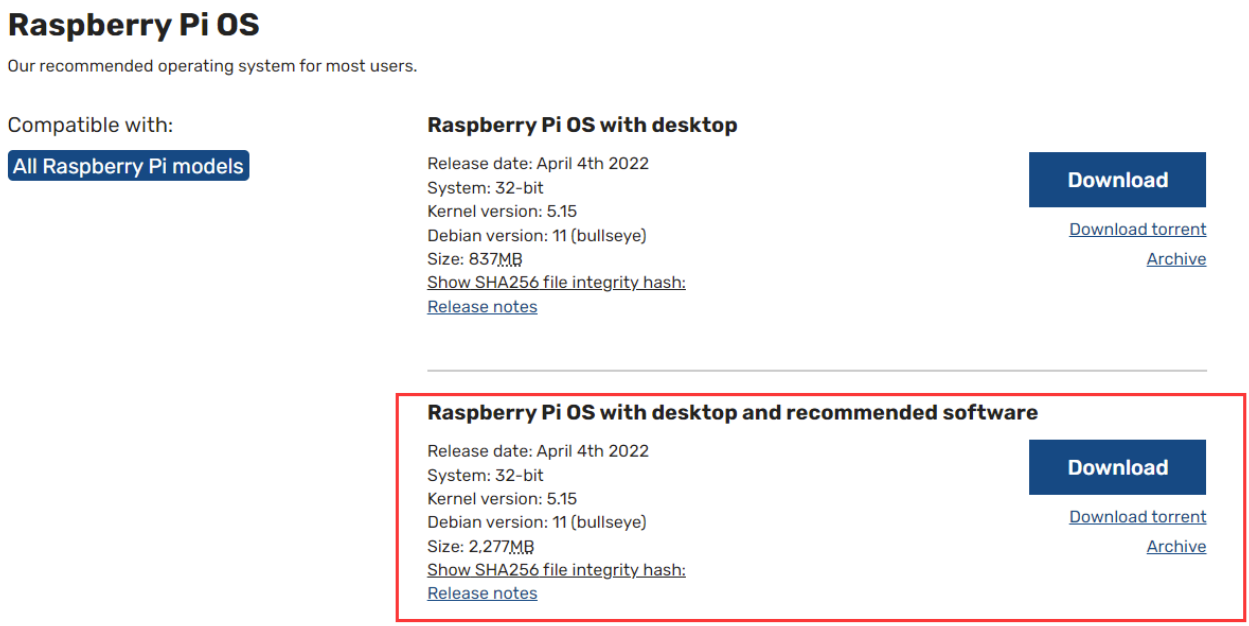
https://www.raspberrypi.com/software/



Download OS file

Visit following website to download the OS file.

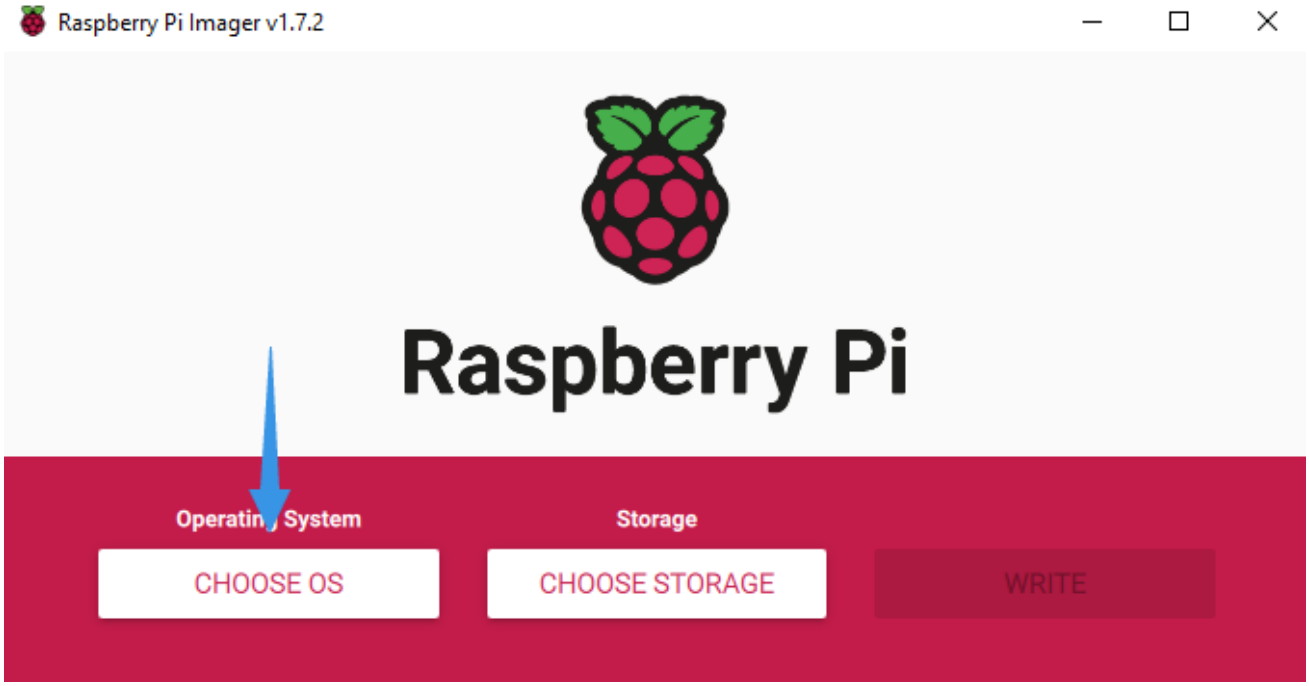
https://www.raspberrypi.com/software/operating-systems/

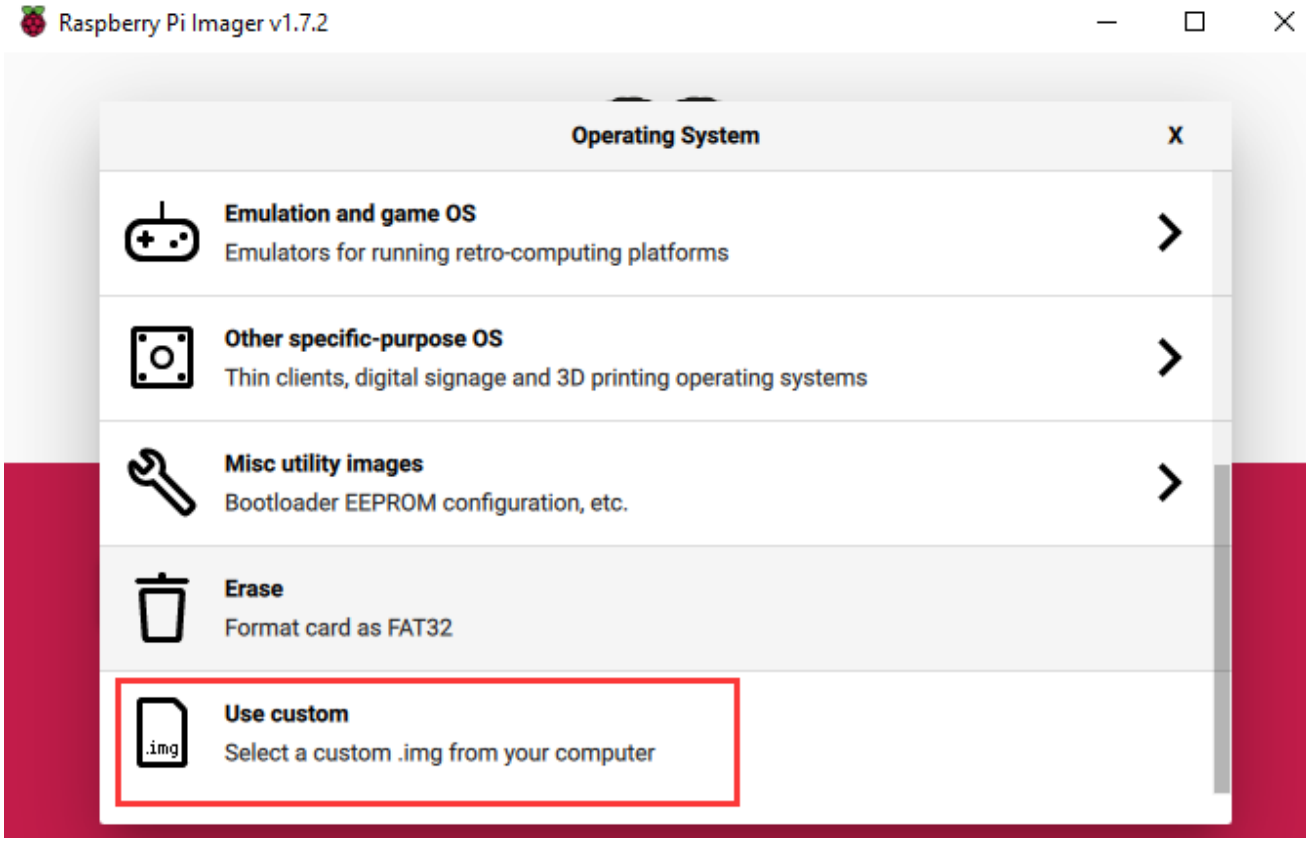


Write System to Micro SD Card

First, insert your Micro **SD card** into **card reader** and connect it to USB port of **PC**. Then open imager tool.

Choose system that you just downloaded in Use custom.





Choose the SD card.

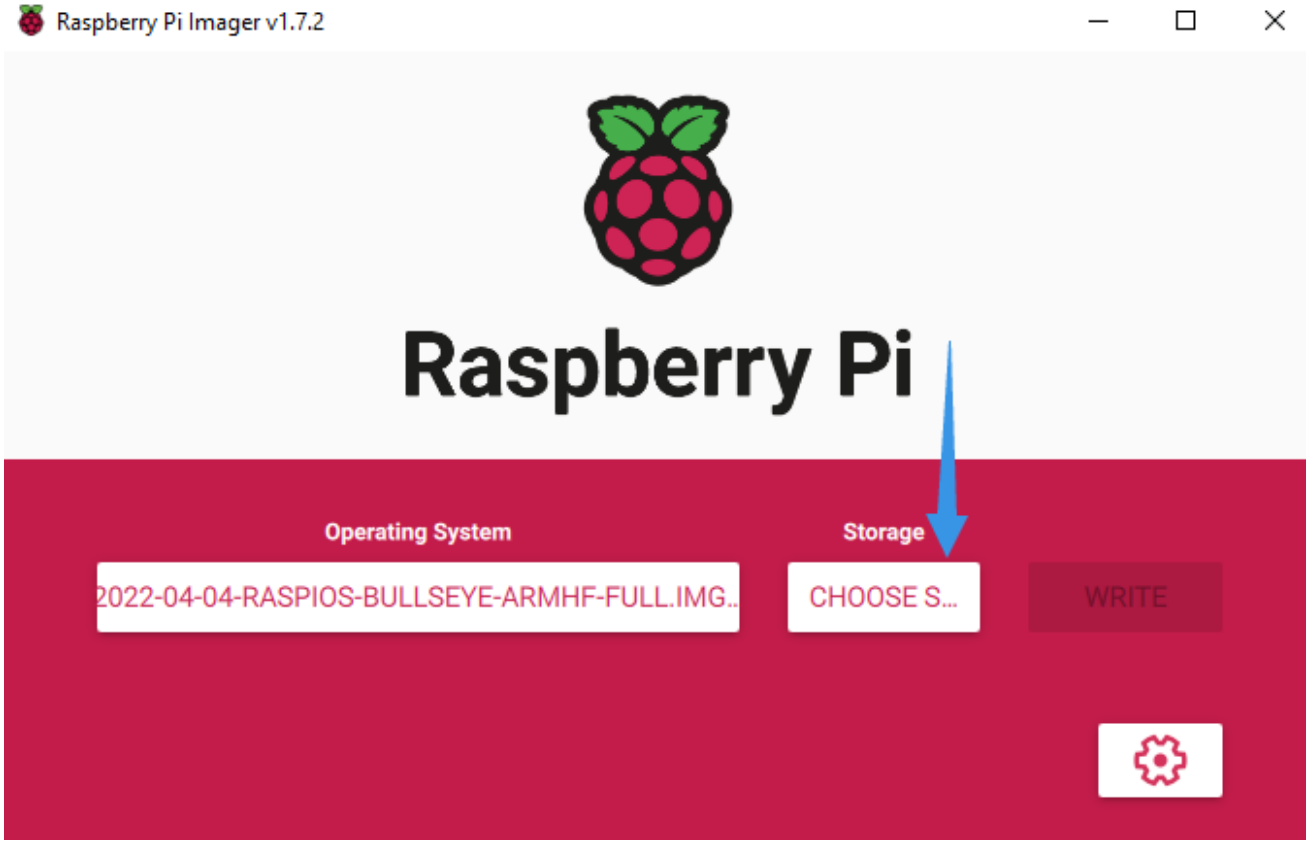
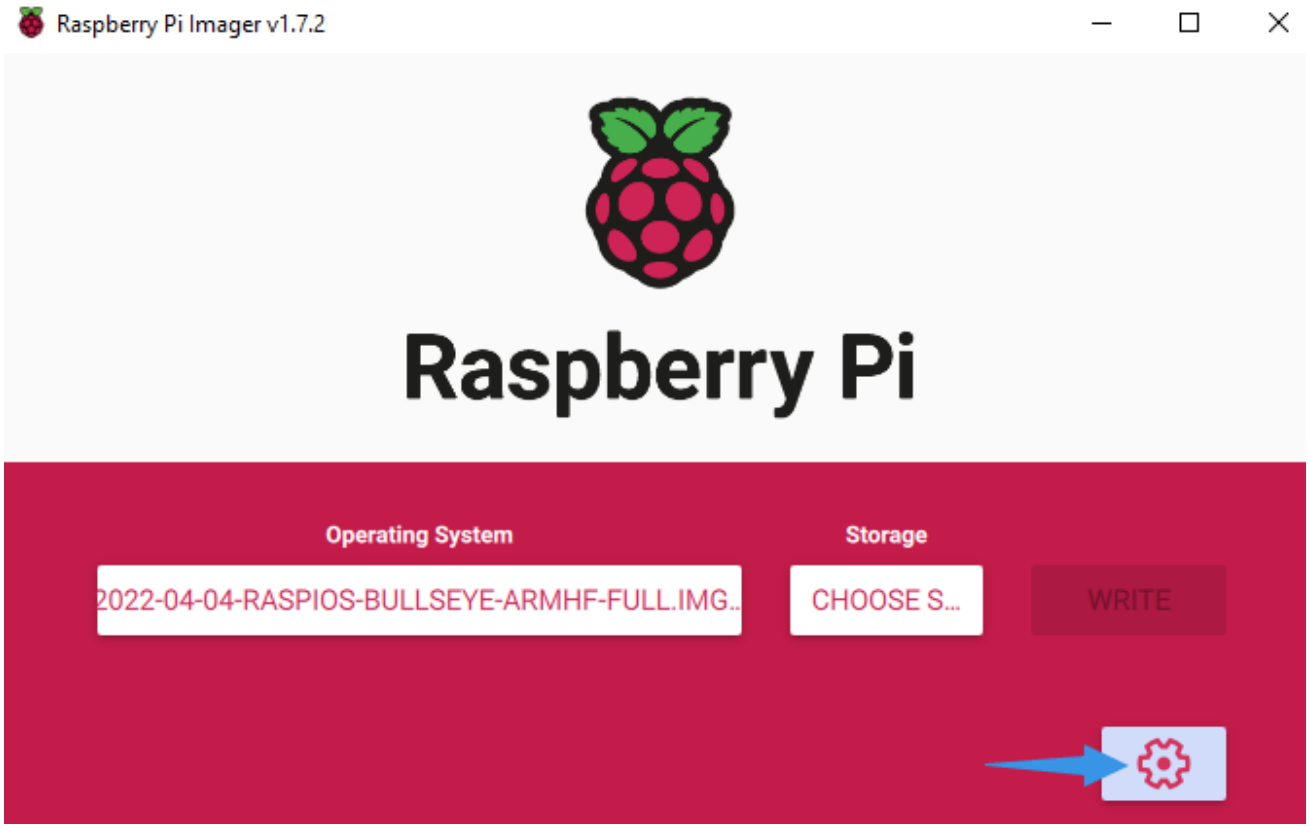
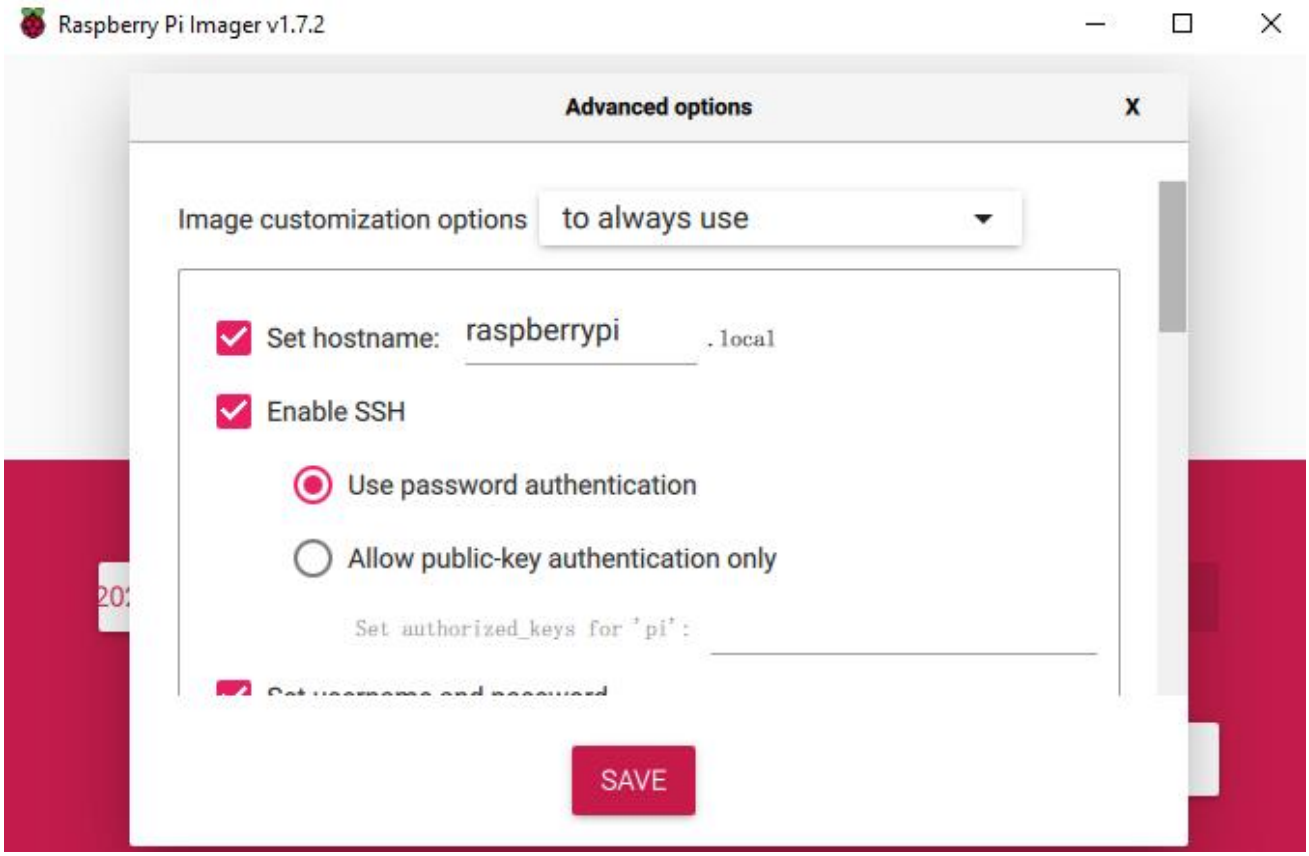


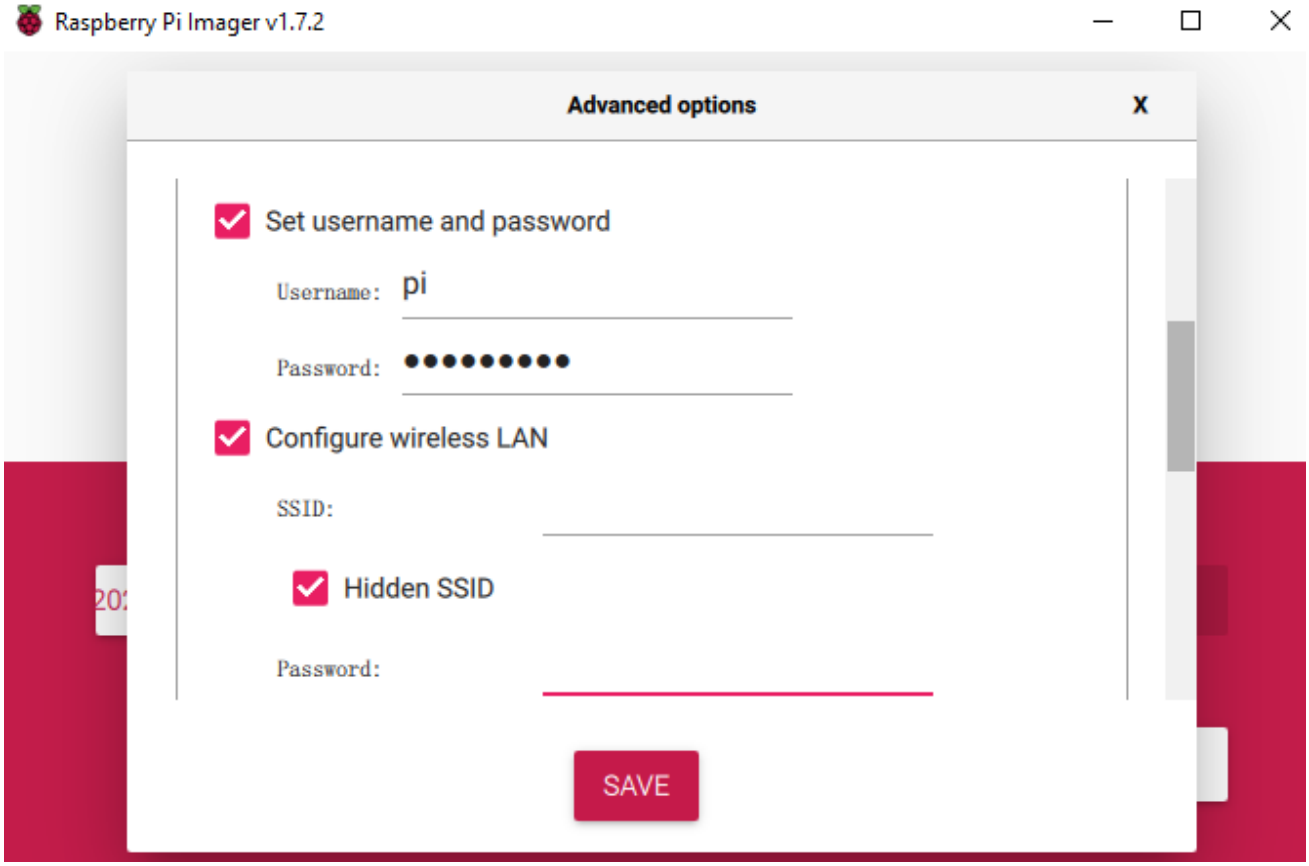
Image option.



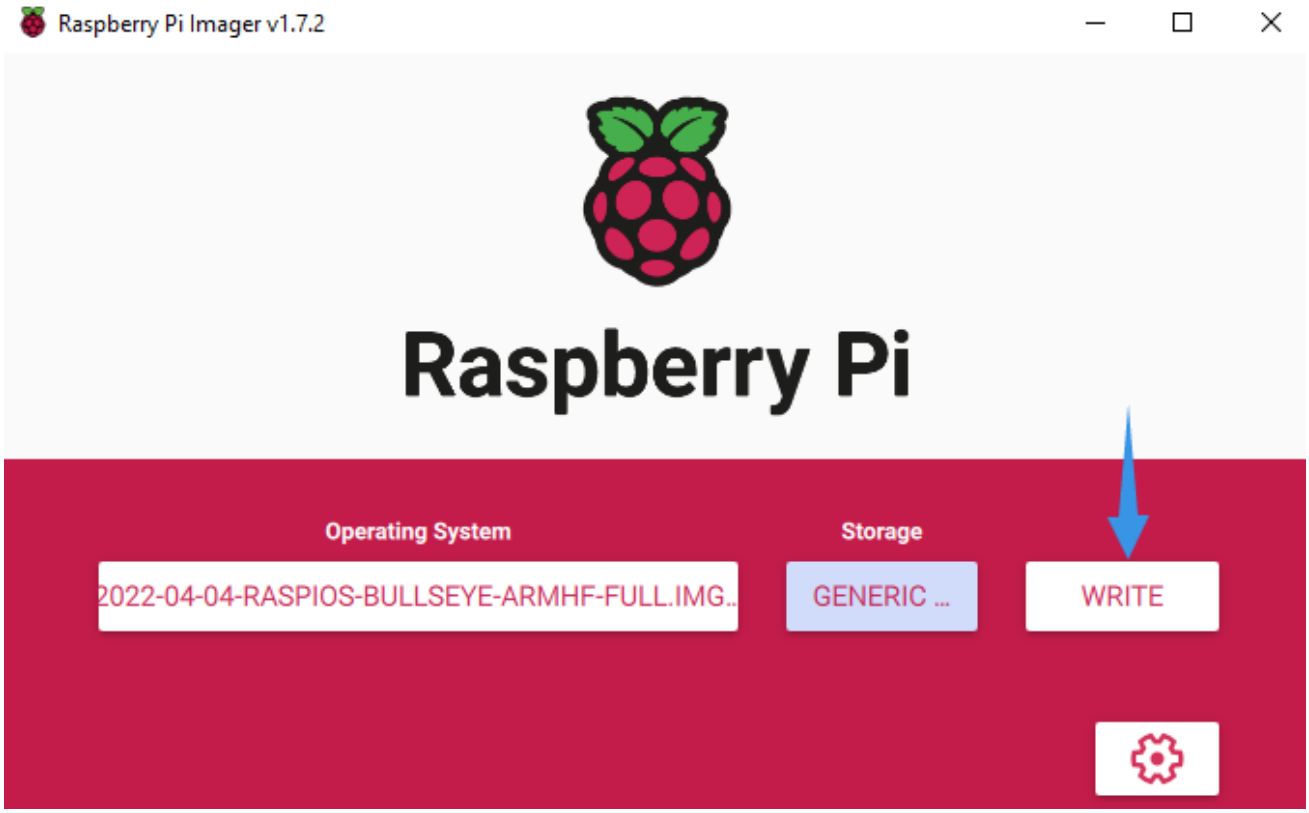
Enable SSH.



Configure WiFi and location. Here we set username as **pi**, password as **raspberry**



Finally WRITE.

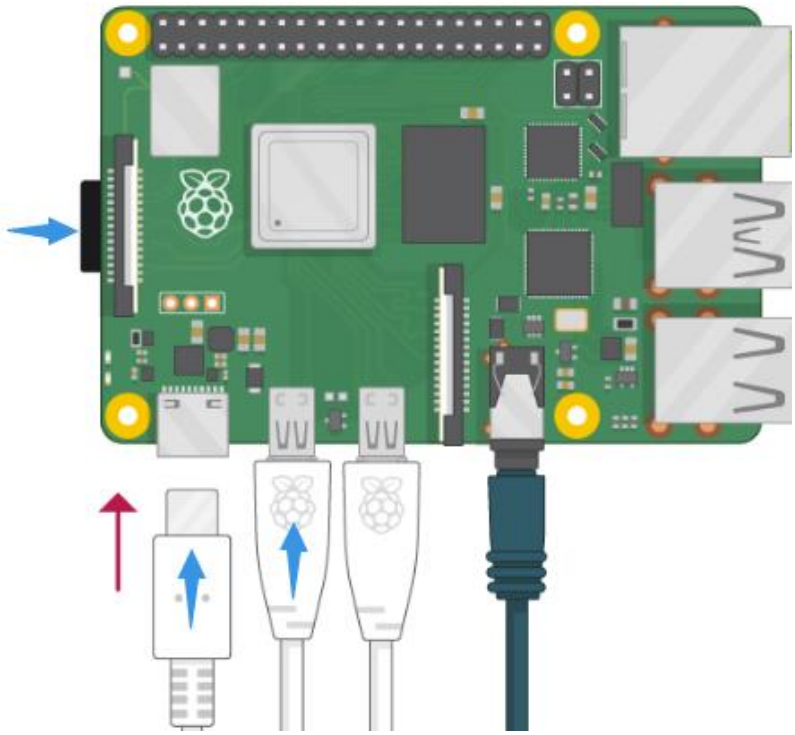

Raspberry Pi Imager v1.7.2
Raspberry Pi
Operating System
Storage
2022-04-04-RASPIOS-BULLSEYE-ARMHF-FULL.IMG..
GENERIC ...
WRITE

Start Raspberry Pi

If you don't have a spare monitor, please skip to next section.

If you have a spare monitor, please follow the steps in this section.

After the system is written successfully, put SD card into the SD card slot of RPi. Then connect your RPi to monitor through HDMI cable, attach your mouse and keyboard through the USB ports,



Later, after setup, you will need to enter your user name and password to login. The default user name: pi; password: raspberry. After login, you should see the following screen.

You can connect WiFi on the right corner if WiFi is connected successfully.

Now you can skip to VNC Viewer.

Remote desktop & VNC

After you log in Raspberry Pi, please use VNC Viewer to connect Raspberry Pi for this robot. Other remote ways may not support GUI. **If you have logged in Raspberry Pi please skip to VNC Viewer.**

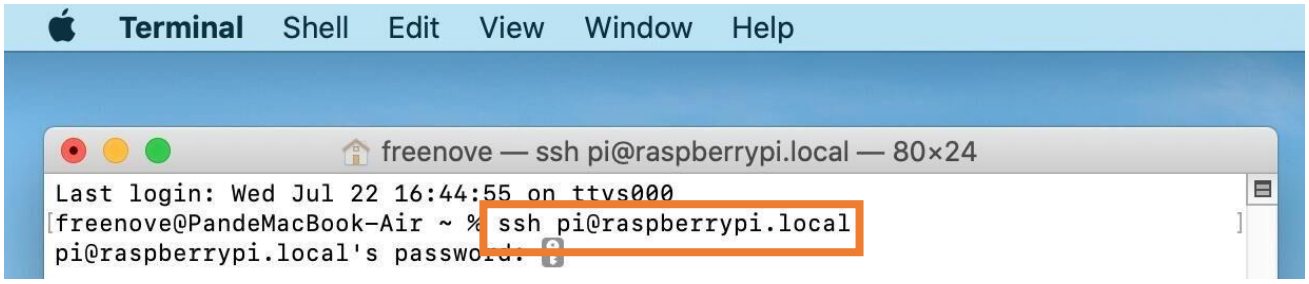
If you don't have a spare monitor, mouse and keyboard for your RPi, you can use a remote desktop to share a display, keyboard, and mouse with your PC. Below is how to use:
MAC OS remote desktop and Windows OS remote desktop.

MAC OS Remote Desktop

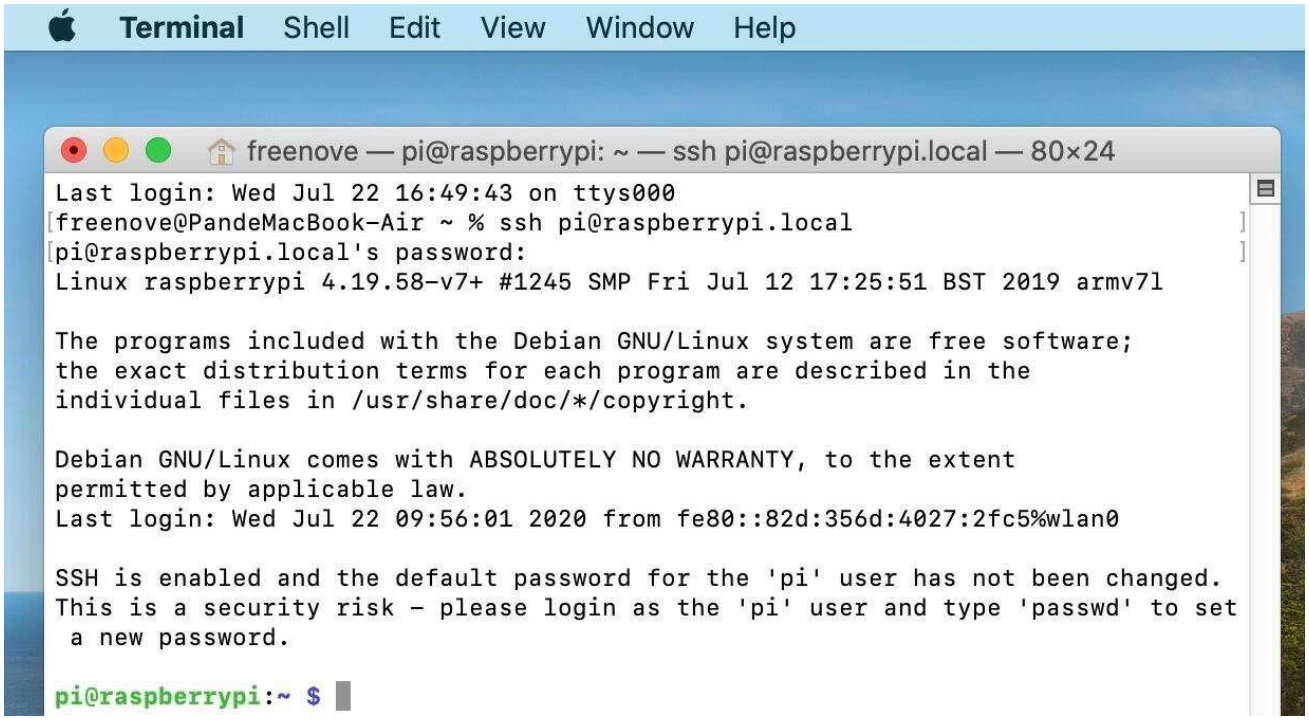
Open the terminal and type following command. **If this command doesn't work, please move to next page.**

```
ssh pi@raspberrypi.local
```

The password is **raspberry** by default, case sensitive.



You may need to type **yes** during the process.



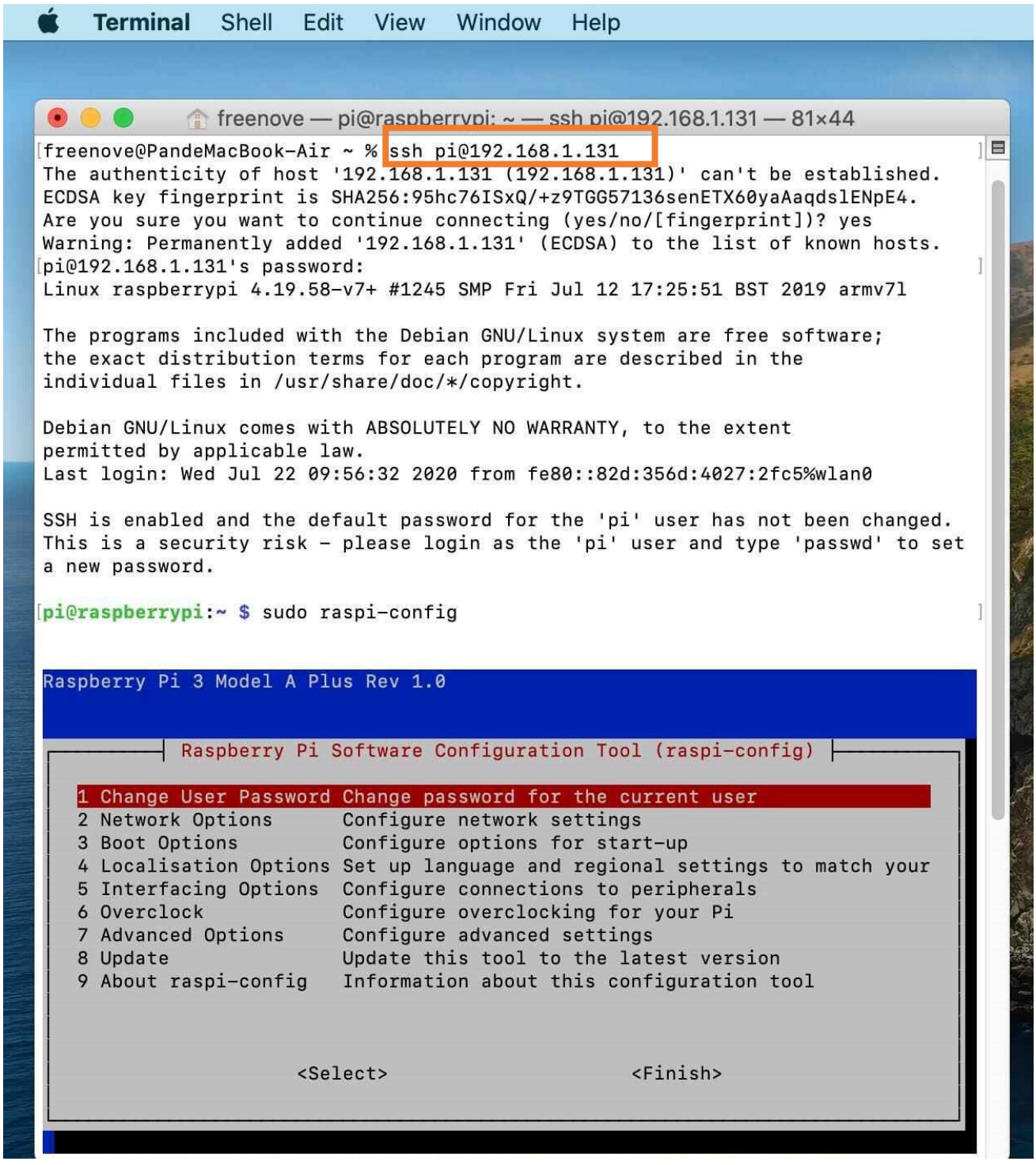
You can also use the IP address to log in Pi.

Enter **router** client to **inquiry IP address** named "raspberry pi". For example, I have inquired to **my RPi IP address, and it is "192.168.1.131".**

Open the terminal and type following command.

```
ssh pi@192.168.1.131
```

When you see pi@raspberrypi:~ $, you have logged in Pi successfully. Then you can skip to next section.



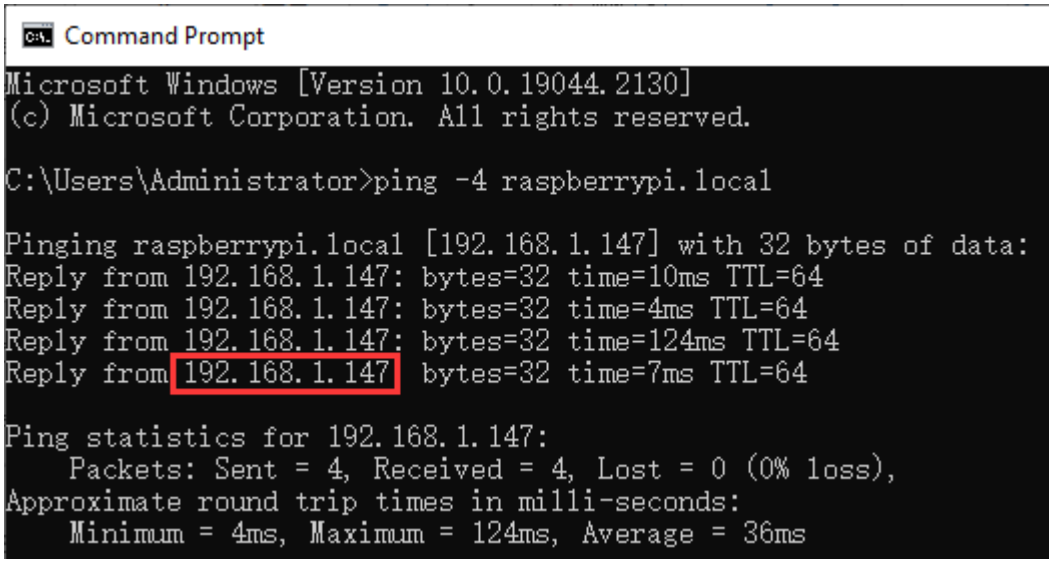
Then you can skip to VNC Viewer.

Windows OS Remote Desktop

If you are using win10, you can use follow way to login Raspberry Pi without desktop.

Press **Win+R**. Enter **cmd**. Then use this command to check IP:

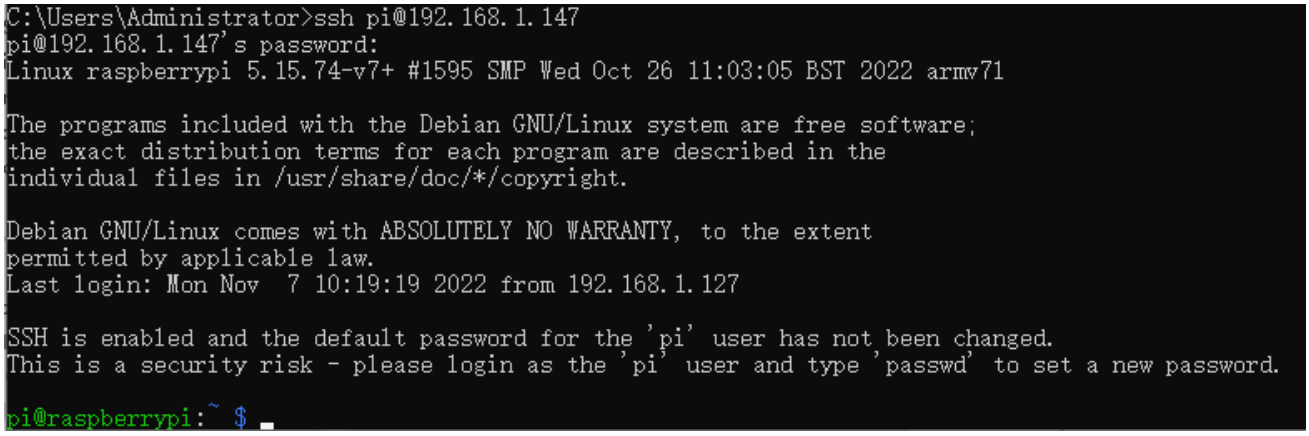
```
ping -4 raspberrypi.local
```



Then 192.168.1.147 is my Raspberry Pi IP.

Or enter **router** client to **inquiry IP address** named "**raspberrypi**". For example, I have inquired to **my RPi IP address, and it is "192.168.1.147".**

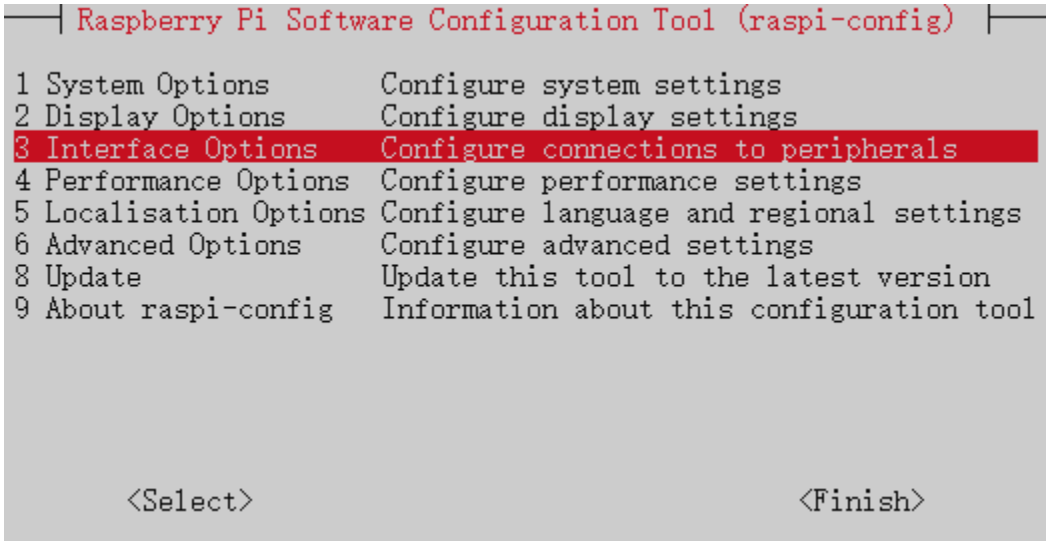
```
ssh pi@192.168.1.147
```

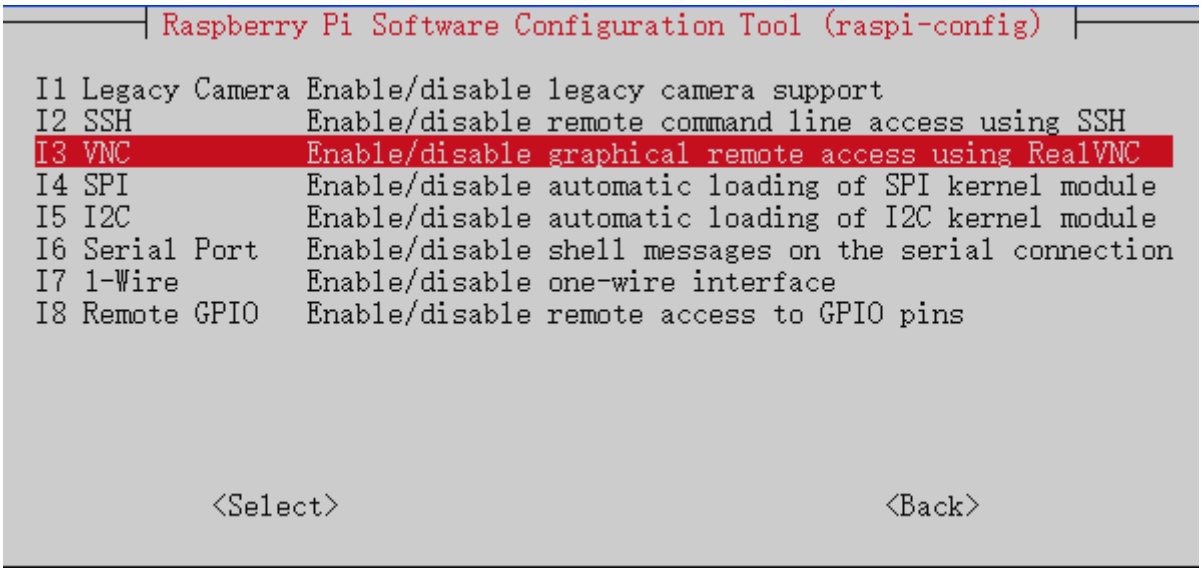


VNC Viewer & VNC

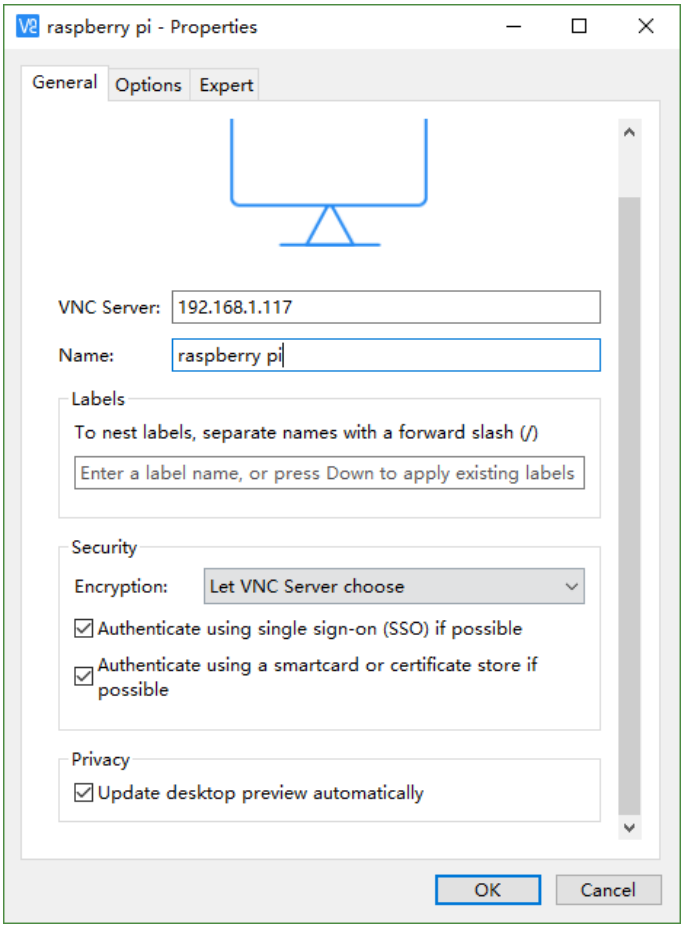
Type the following command. And select Interfacing **Options→ VNC →Yes→OK→Finish**. Here Raspberry Pi may need to be restarted, and choose ok. Then open VNC interface.

```
sudo raspi-config
```

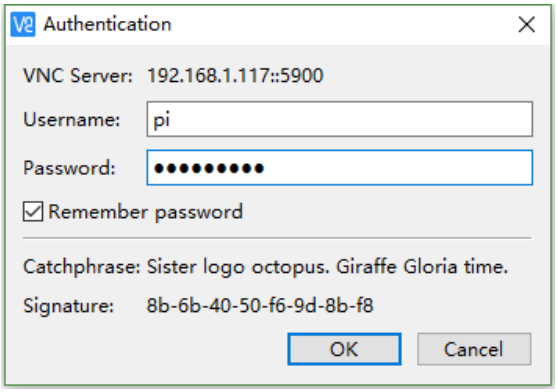




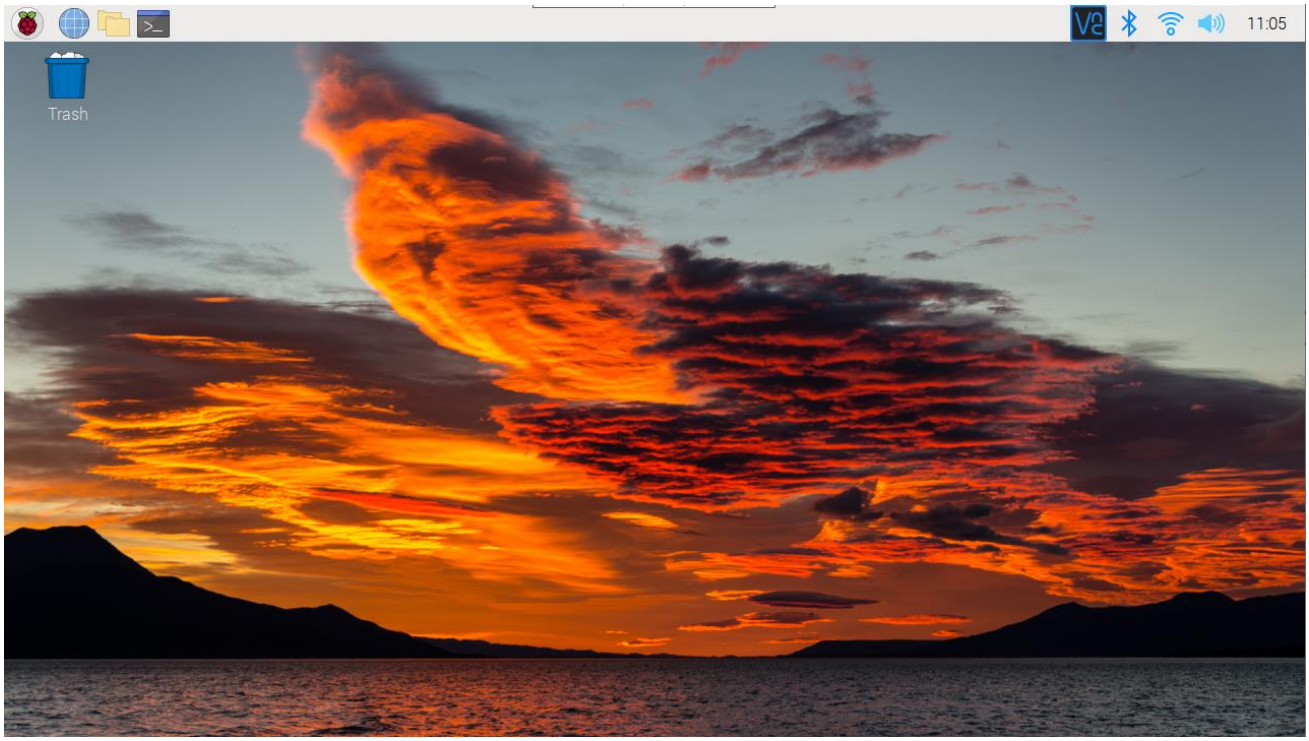
Then download and install VNC Viewer according to your computer system by clicking following link:
https://www.realvnc.com/en/connect/download/viewer/
After installation is completed, open VNC Viewer. And click File → New Connection. Then the interface is shown below.



Enter IP address of your Raspberry Pi and fill in a Name. And click OK.
Then on the VNC Viewer panel, double-click new connection you just created, and the following dialog box pops up.



Enter username: **pi** and Password: **raspberry**. And click OK.

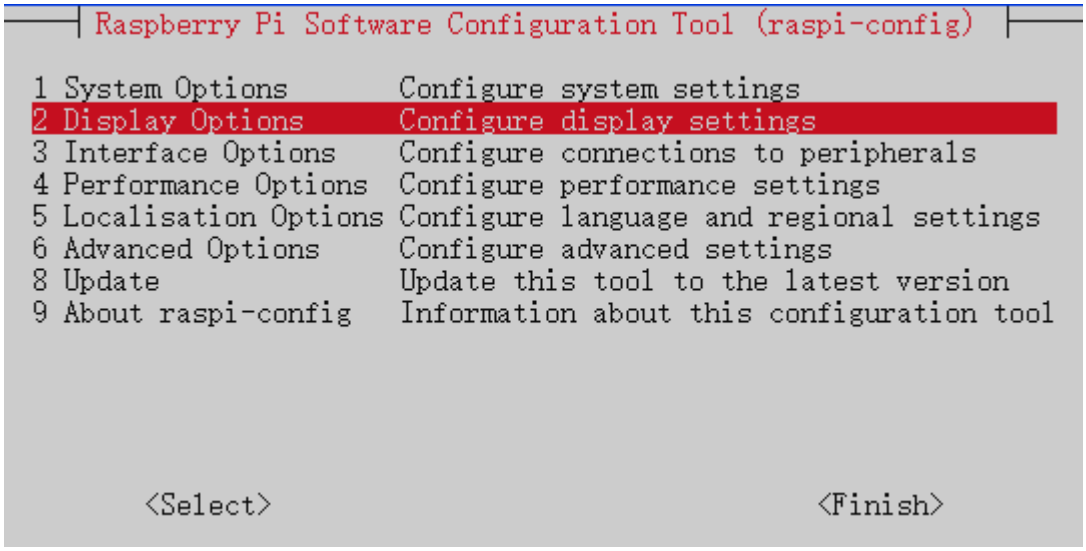


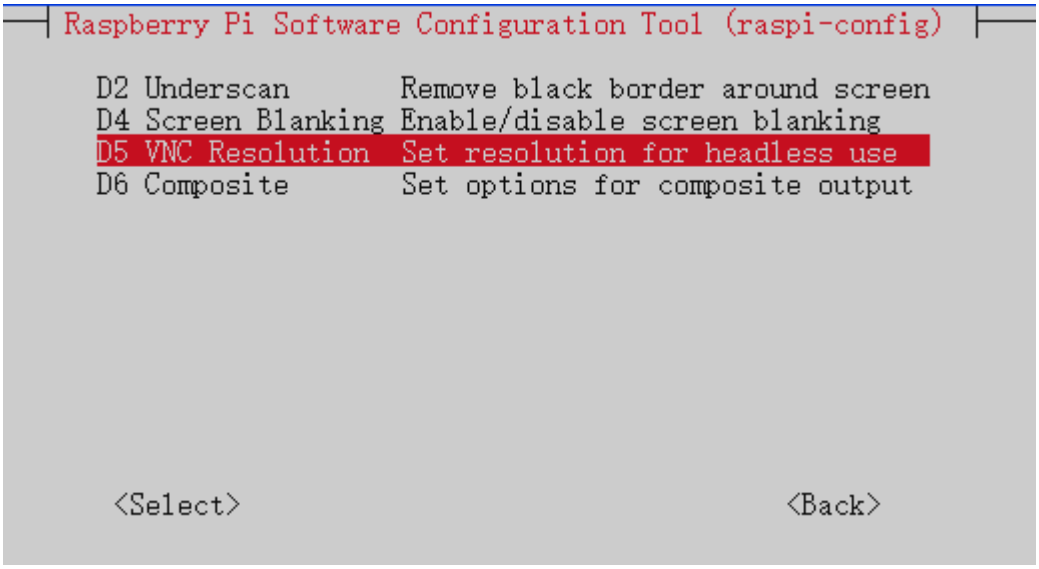
Here, you have logged in to Raspberry Pi successfully by using VNC Viewer

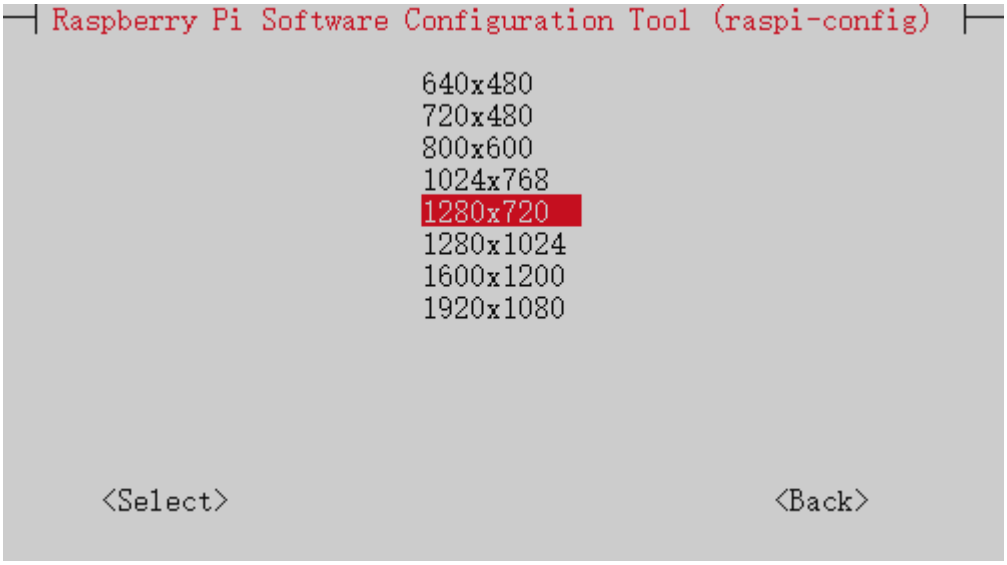
If the resolution ratio is not great or there is just a **little window**, you can set a proper resolution ratio via steps below.

```
sudo raspi-config
```

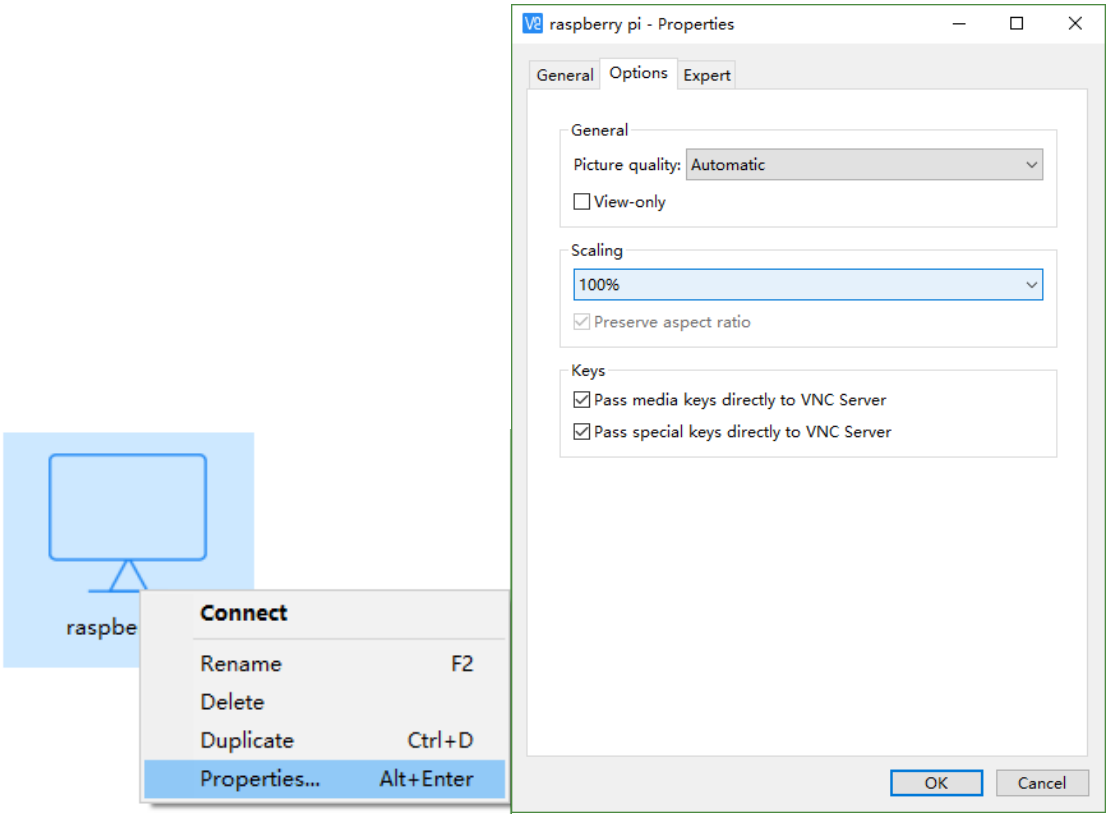
Select **Display Options→VNV Resolution→Proper resolution ratio (set by yourself)→OK→Finish→Yes**. And then reboot Raspberry Pi.







In addition, your VNC Viewer window may zoom your Raspberry Pi desktop. You can change it. On your VNC View control panel, click right key. And select Properties->Options label->Scaling. Then set proper scaling.



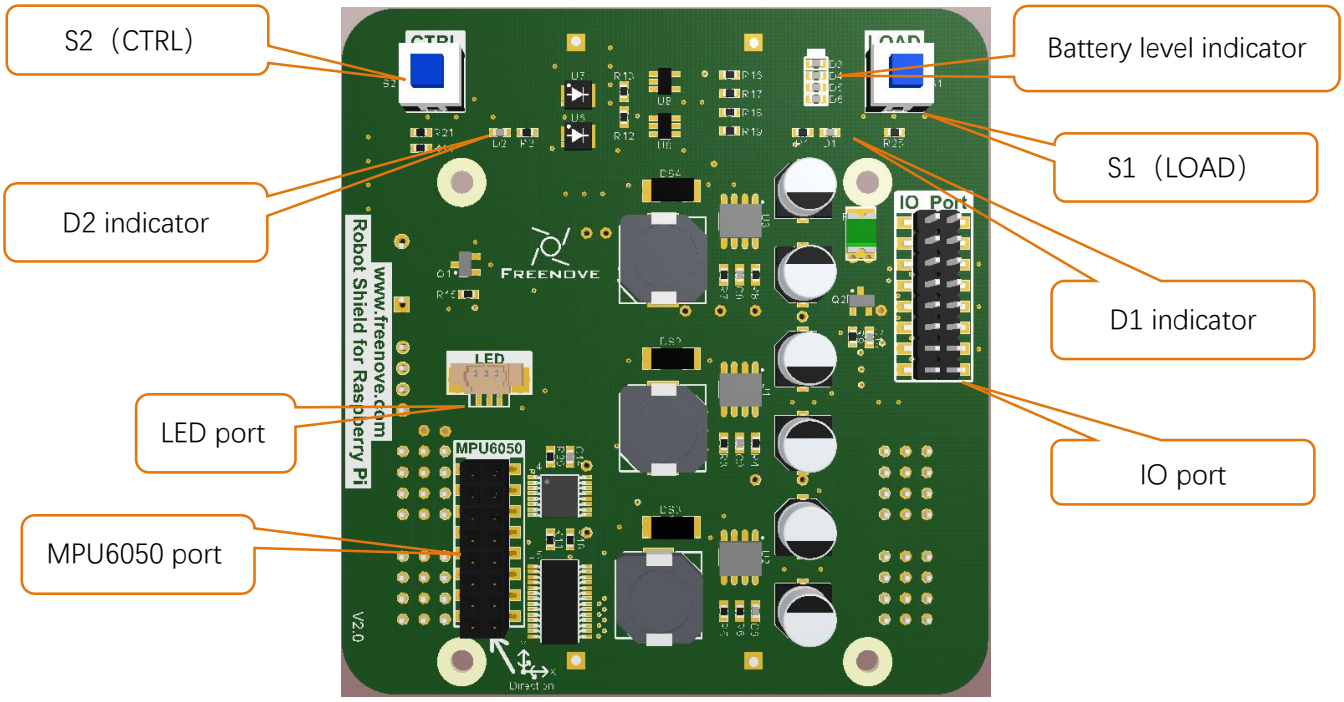
Here, you have logged in to Raspberry Pi successfully by using VNC Viewer and operated proper setting.

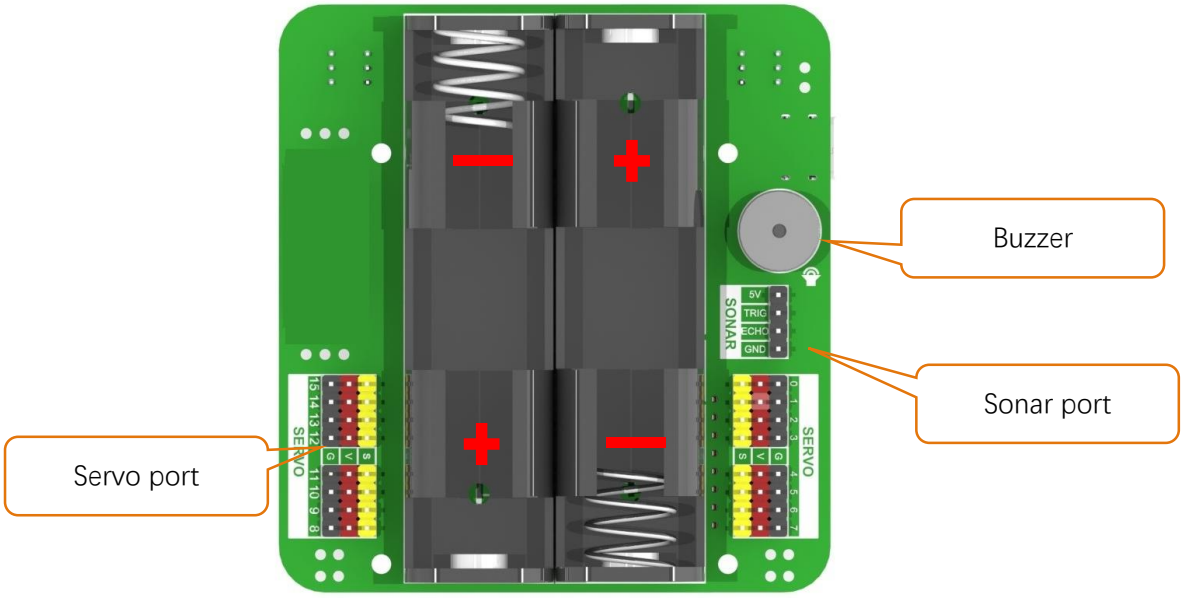
Raspberry Pi 4B/3B+/3B integrates a Wi-Fi adaptor. If you did not connect Pi to WiFi. You can connect it to wirelessly control the robot.

Robot Shield for Raspberry Pi

Shield Introduction

The shield is connected to the Raspberry Pi through the IO Port on the board. The positioning holes on the board are suitable for the Raspberry Pi. The features and functions are as follows.





- **Robot Shield board**: It requires two 18650 3.7v batteries.
- **S1(LOAD) Switch:** It mainly controls the power supply of servos, buzzer, ultrasonic module and LED module. The D1 indicator will light up when S1 is pressed.
- **S2(CTRL) Switch:** It mainly controls the power of chips such as PCF9685, ADS7830 and Raspberry Pi. The D2 indicator will light up when S2 is pressed.
- **LED port**: It is used to connect LED module which is controlled by Raspberry Pi.
- **MPU6050 port**: It is used to connect MPU6050 which is controlled by Raspberry Pi.
- **Battery level indicator**: It consists of four LEDs. When the battery power decreases gradually, the LEDs will get dimmer gradually until it's turned off.
- **IO port**: It is used to connect shield with Raspberry Pi conveniently.
- **Servo port**: There are 16 servo ports with control accuracy of 0.09 degrees.
- **Sonic module port**: The interface of HC-SR04 Ultrasound Module is provided.
- **Buzzer**: It is directly controlled by raspberry pi, which is turned on at high level and off at low level.

Chapter 1 Install Python Libraries (Required)

If you have any concerns, please feel free to contact us at support@freenove.com

In this chapter, we will do some necessary foundational preparation work: Start your Raspberry Pi and install some necessary libraries. And in next chapter, we will assemble the robot dog.

Note:

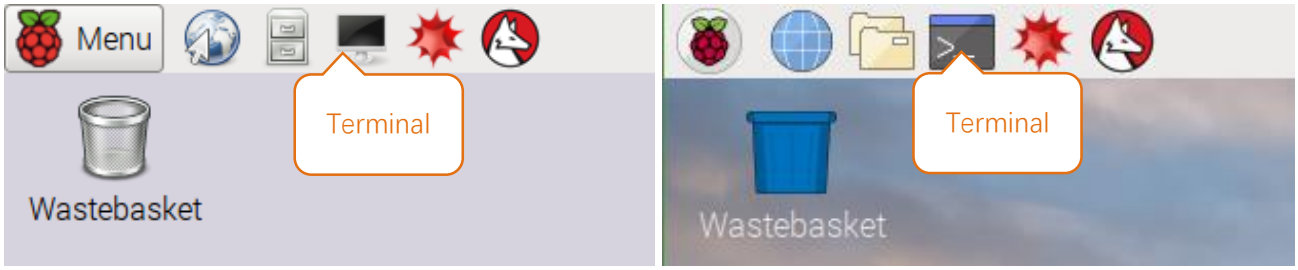
1, **Please use Raspberry Pi OS with Desktop**

2, The installation of libraries takes much time. **You can power Raspberry Pi with a power supply Cable**.

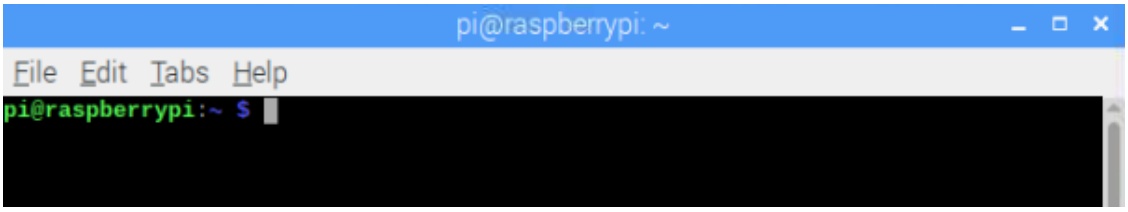
3, If you are using **remote desktop** to login Raspberry Pi, you need to use **VNC viewer**.

Step 1 Obtain the Code and Set python3 as default

Start the Raspberry Pi and open the terminal. You can click the terminal as shown below, or press "CTAL+ALT+T" on the desktop.



The terminal is shown below:



You can view YouTube video as a reference.

https://www.youtube.com/watch?v=Lq-OERj2WZM

Type following command to get robot dog code and place it in user directory "Pi".

Please execute commands below one by one in turn.

```
cd ~
git clone https://github.com/Freenove/Freenove_Robot_Dog_Kit_for_Raspberry_Pi
```

Downloading takes much time. Please wait with patience.

You can also find and download the code by visiting our official website (http://www.freenove.com) or our GitHub repository (https://github.com/freenove).

Please note that all codes for this robot dog is written with **Python3**. If executed under python 2, there will be error messages.

Set Python3 as default python

First, execute python to check the default python on your raspberry Pi. Press Ctrl-Z to exit.

```
pi@raspberrypi:~ $ python --version
```

If it is python3, you can skip this section.

If it is python2, you need to execute the following commands to set default python to python3.

1. Enter directory /usr/bin

```
cd /usr/bin
```

2. Delete the old python link.

```
sudo rm python
```

3. Create new python links to python.

```
sudo ln -s python3 python
```

4. Check python. Press Ctrl-Z to exit.

```
python --version
```

```
pi@raspberrypi:~ $ cd /usr/bin
pi@raspberrypi:/usr/bin $ sudo rm python
pi@raspberrypi:/usr/bin $ sudo ln -s python3 python
```

If you want to set python2 as default python in **other projects**, just repeat the above command and change python3 to python2.

```
pi@raspberrypi:~ $ cd /usr/bin
pi@raspberrypi:/usr/bin $ sudo rm python
pi@raspberrypi:/usr/bin $ sudo ln -s python2 python
```

Shortcut Key

Now, we will introduce several shortcuts that are very **useful** and **commonly used** in terminal.

1. **up and down arrow keys**. History commands can be quickly brought back by using up and down arrow keys, which are very useful when you need to reuse certain commands.

When you need to type command, pressing "↑" (the Up key) will go backwards through the command history and pressing "↓" (the Down Key) will go forwards through the command history.

2. **Tab key**. The Tab key can automatically complete the command/path you want to type. When there are multiple commands/paths conforming to the already typed letters, pressing Tab key once won't have any result. And pressing Tab key again will list all the eligible options. However, when there is only one eligible option, the command/path will be completely typed as soon as you press the Tab key..

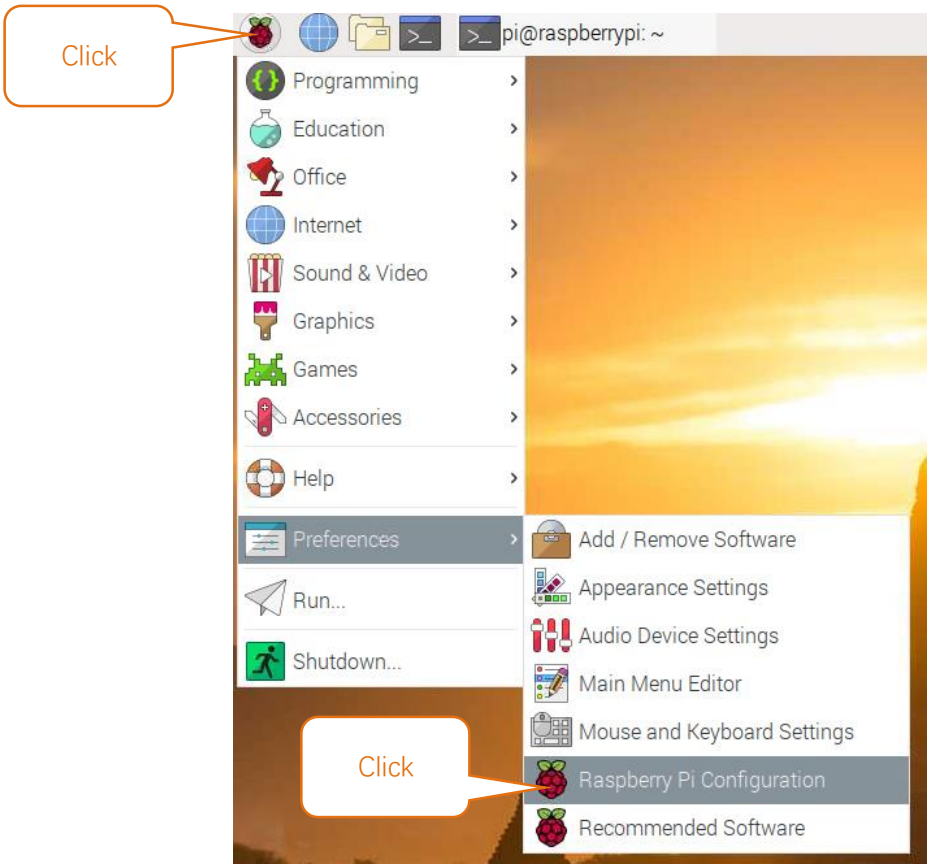
As shown below, under the '~'directory, enter the Documents directory with the "cd" command. After typing "cd D", press Tab key, there is no response. Press Tab key again, all the files/folders that begin with "D" is listed. Continue to type the character "oc", then press the Tab key, and then "Documents" is typed automatically.

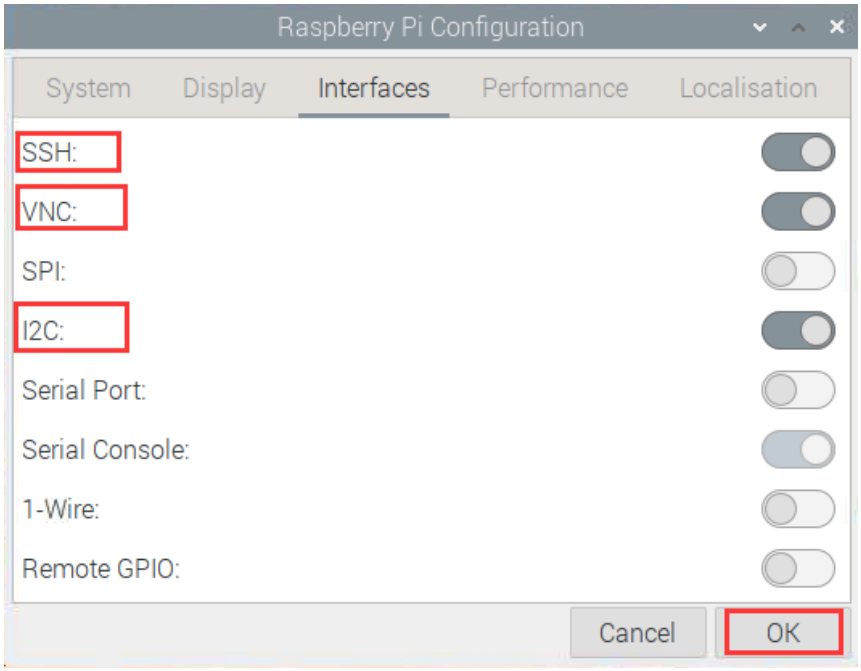
```
pi@raspberrypi:~ $ cd D
Desktop/   Documents/ Downloads/
pi@raspberrypi:~ $ cd Doc
```

```
pi@raspberrypi:~ $ cd D
Desktop/   Documents/ Downloads/
pi@raspberrypi:~ $ cd Documents/
```

Step 2 Configuration

Enable i2c

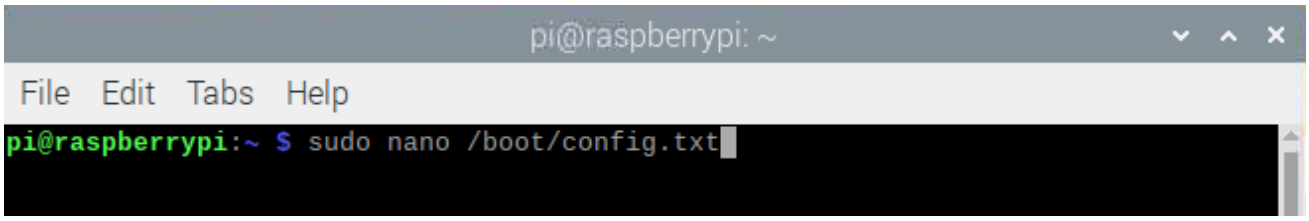




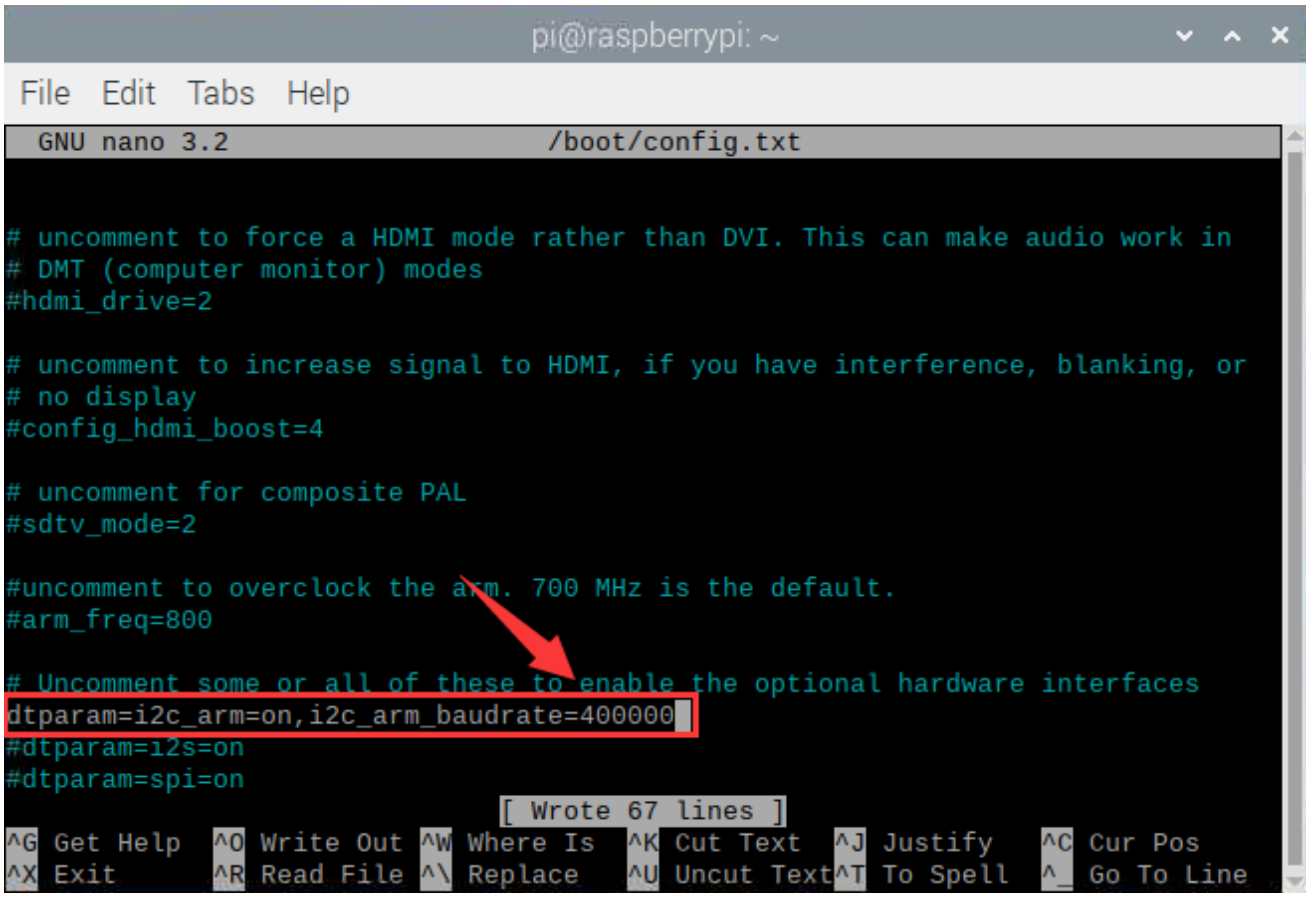
Set I2C Baud Rate

Open the terminal and enter the following command.

```
sudo nano /boot/config.txt
```



The default I2C Baud Rate is 100000. Now we change it to 400000, because this can speed up the response speed of the servos to make robot dog walk faster. If the baud rate is 100,000, the robot walks slowly. Scrolling the middle of the mouse to find **dtparam=i2c_arm=on, and add "i2c_arm_baudrate=400000".**



Press "CTRL"+"O" and then "Enter" to save the modified content. Then press "CTRL"+"X" to exit editing.
After the modification is completed, reboot Raspberry Pi to make the change work. You can also reboot the Raspberry Pi after completing the step 3 below.

Additional supplement

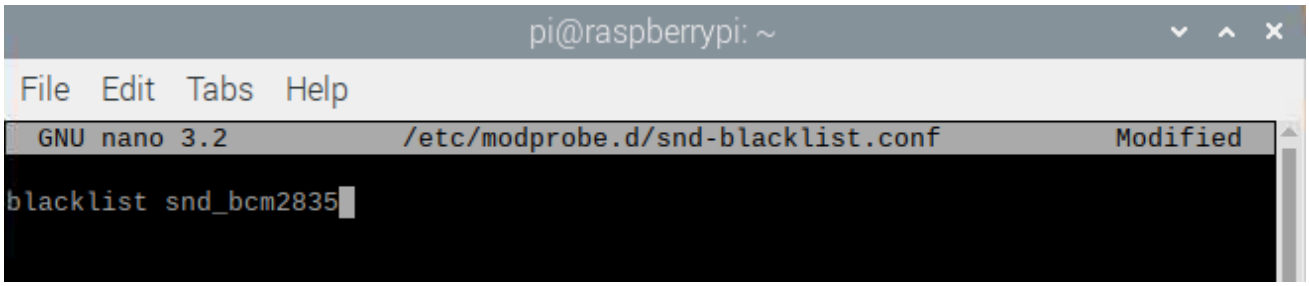
Raspbery Pi, other than 4B and 400, needs to disable the audio module, otherwise the LED will not work properly.

1. Create a new snd-blacklist.conf and open it for editing

```
sudo nano /etc/modprobe.d/snd-blacklist.conf
```

Add following content: After adding the contents, you need to press Ctrl+O, Enter, Ctrl+Z.

```
blacklist snd_bcm2835
```



2. We also need to edit config file.

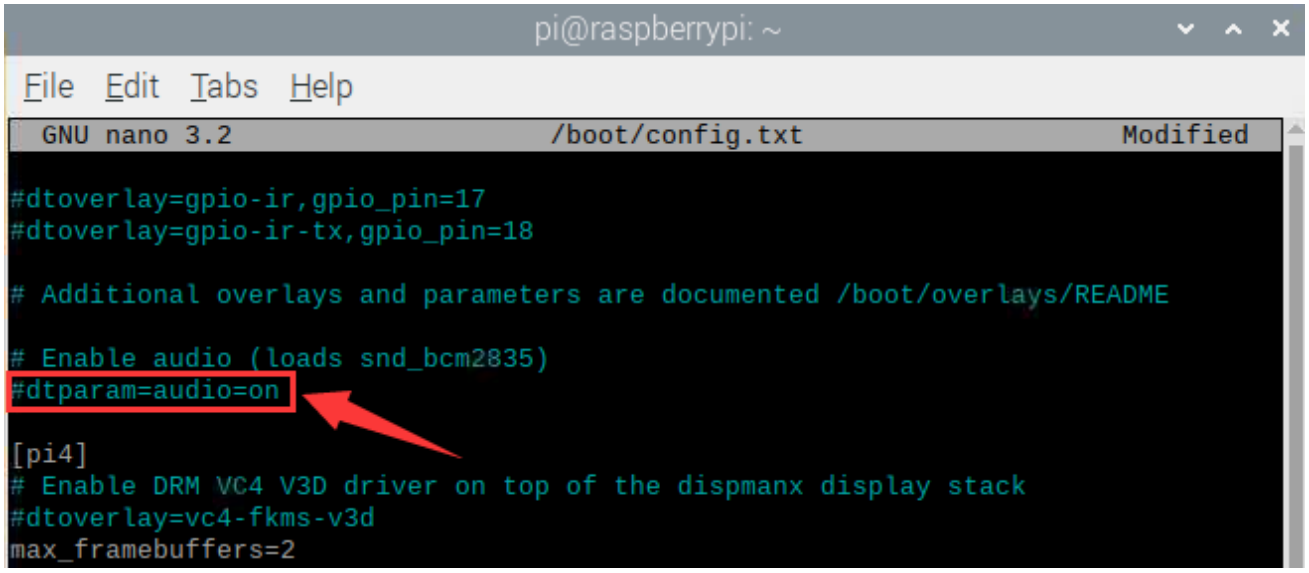
```
sudo nano /boot/config.txt
```

Find the contents of the following two lines (with Ctrl + W you can search):

```
# Enable audio (loads snd_bcm2835)
dtparam=audio=on
```

Add # to comment out the second line. Press Ctrl+O, Enter, Ctrl+Z.

```
# Enable audio (loads snd_bcm2835)
# dtparam=audio=on
```



It will take effect after restarting, and you can restart after executing the next section.

If you want to restart the audio module, just restore the content modified in the above two steps.

Step 3 Run the Installation Program

1. Execute following commands to enter directory of “setup.py”.

```
cd ~/Freenove_Robot_Dog_Kit_for_Raspberry_Pi/Code
```

2. Run setup.py

```
sudo python setup.py
```

This program will automatically install the pca9685, rpi_ws281x, PyQt5 library, etc. Please **reboot** the Raspberry Pi after the installation is completed, as shown below.

If the installation fails, please rerun setup.py. After the installation is completed, restart the Raspberry Pi. Most installation failures are caused by network reasons.

```
sudo python setup.py
```

Chapter 2 Assemble Robot

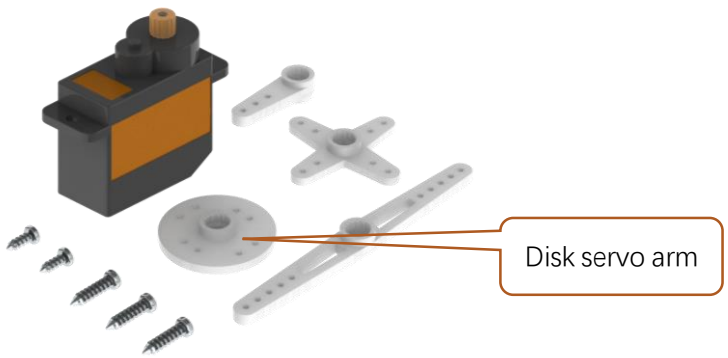
If you have any concerns, please feel free to contact us at support@freenove.com

It is recommended to assemble and use the robot dog according to the tutorial. Otherwise, there may be installation errors, device damage, etc.

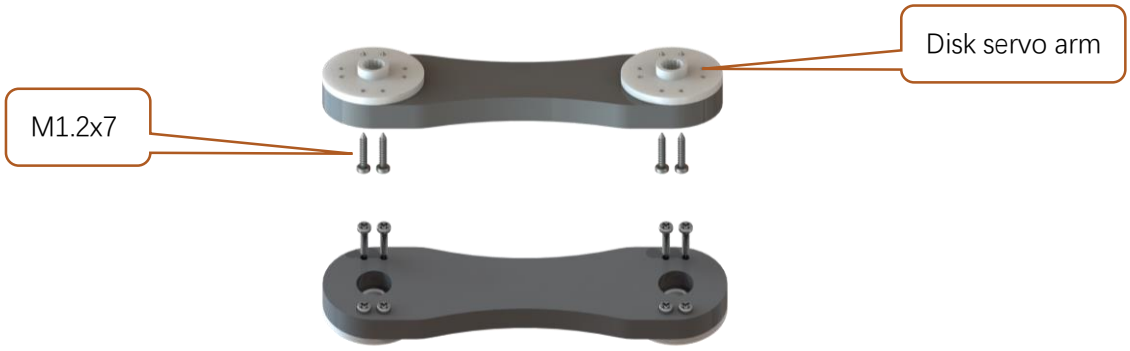
Don't reverse batteries. Or the board will be damaged.

Step 1 Install Disk Servo Arms

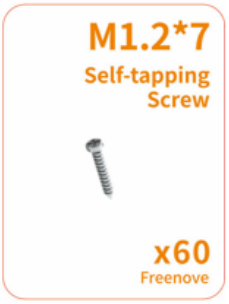
Take out 12 disk servo arms from the servo packages.

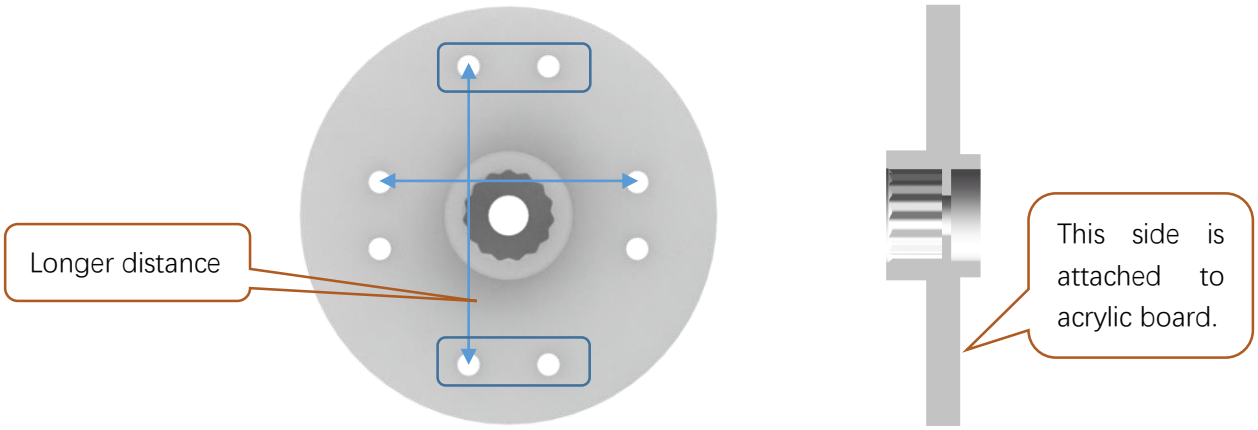


There are 4 pairs of opposite holes on the disk servo arm, and the distance between each pair is different. Please use the 2 pairs of holes with **longer distance**.

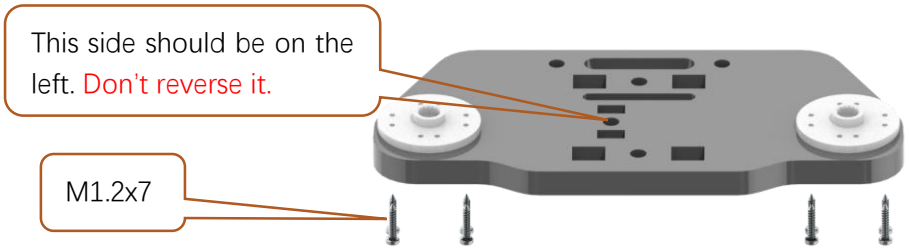


M1.2*7 screws are contained in following plastic bag.

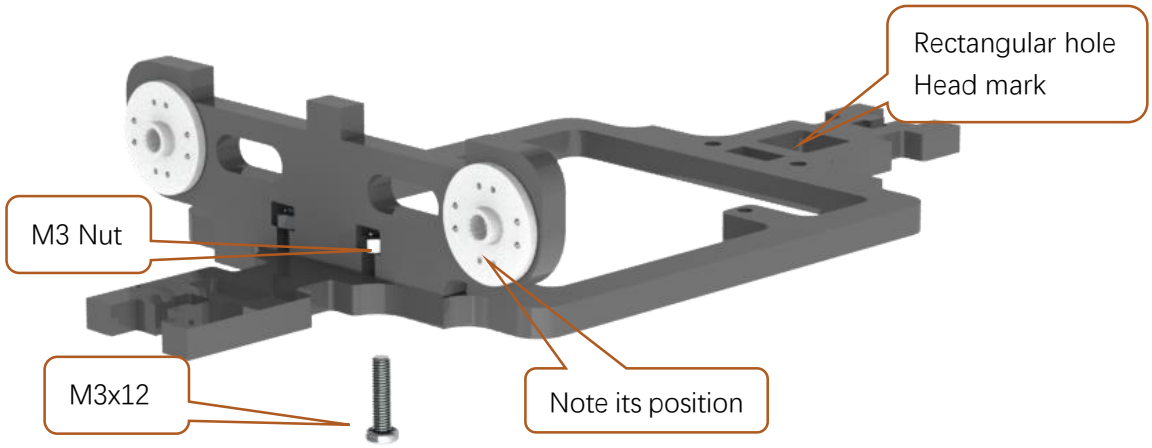


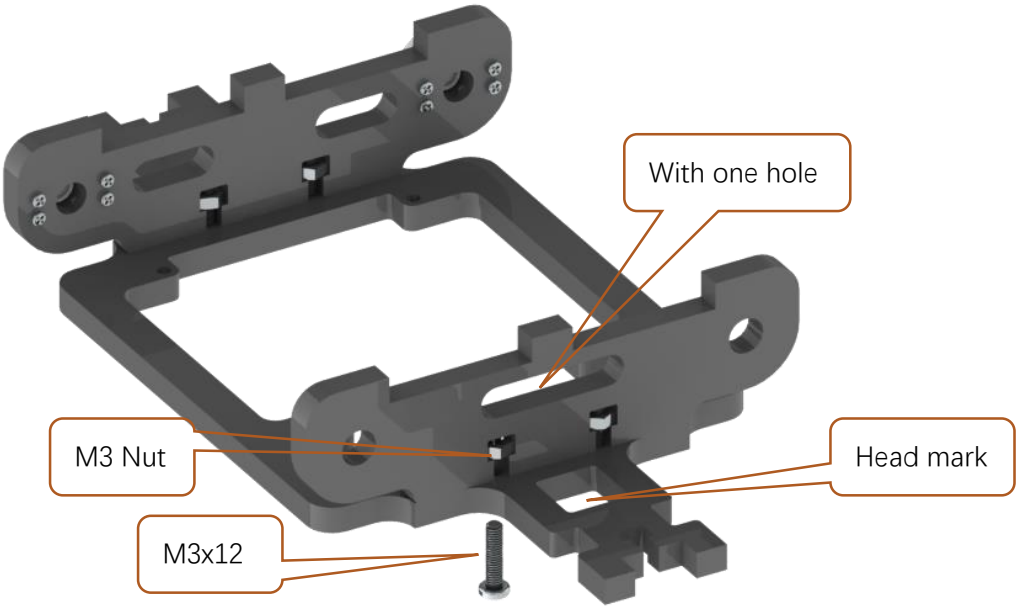


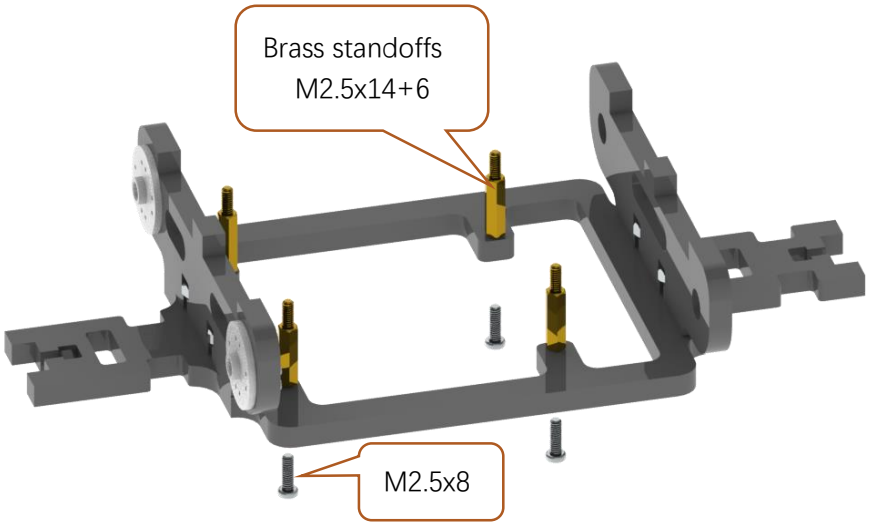
Get following 4 parts.



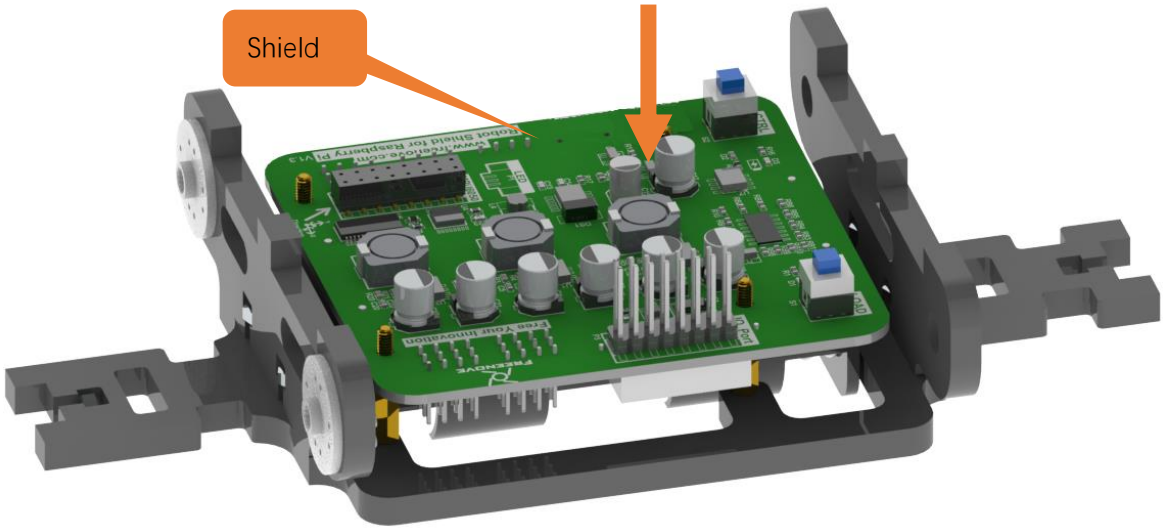
Step 2 Install Body Bracket

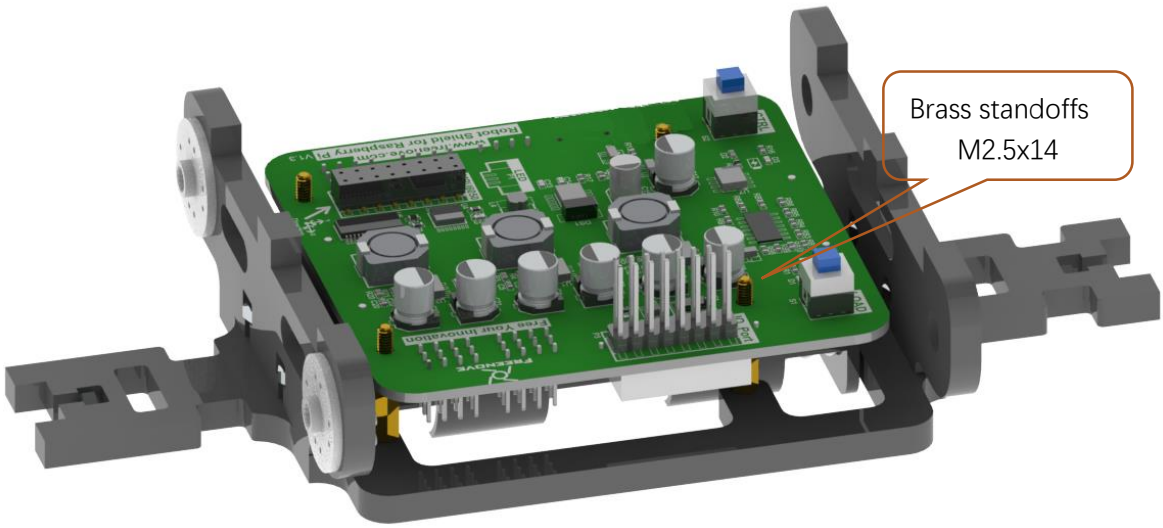






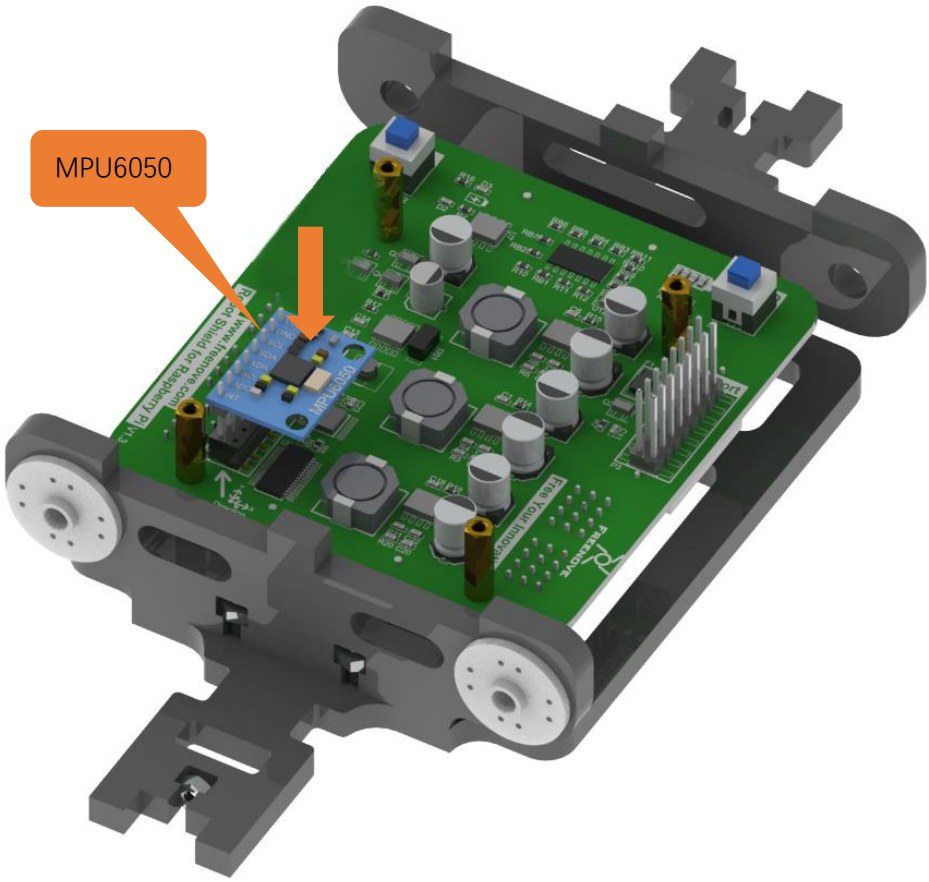
Step 3 Install Shield



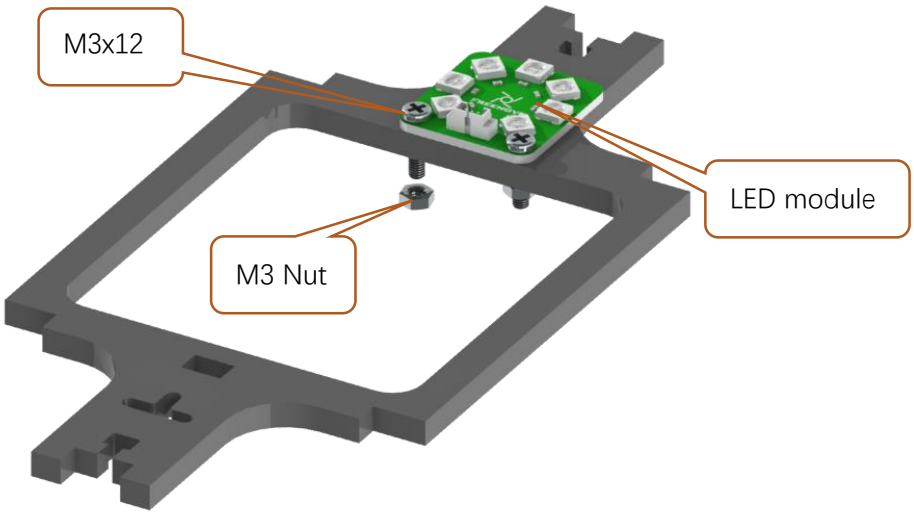


Step 4 Install MPU6050

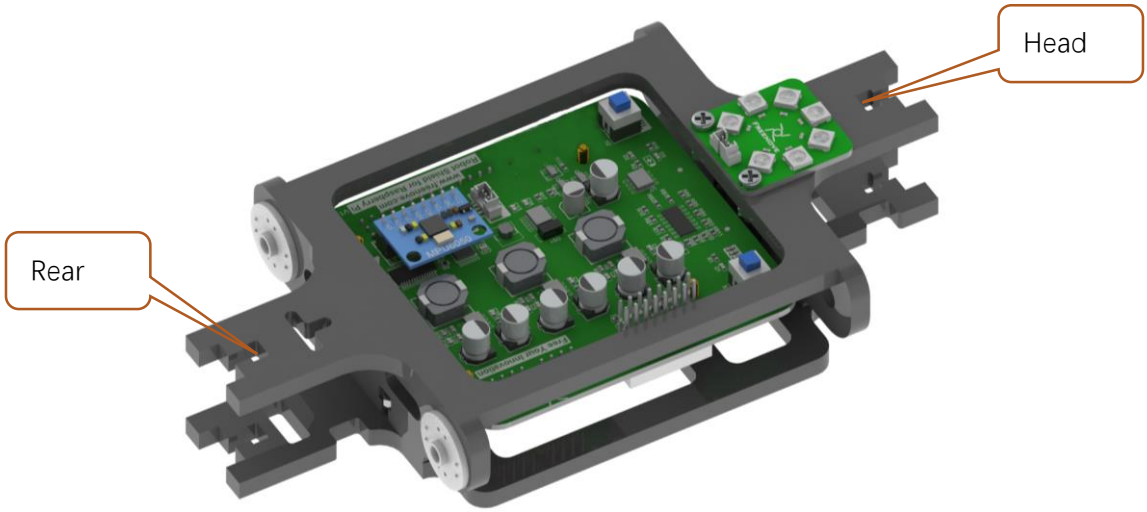
Note: There are two rows of headers. Plug the MPU6050 into the outer Row.



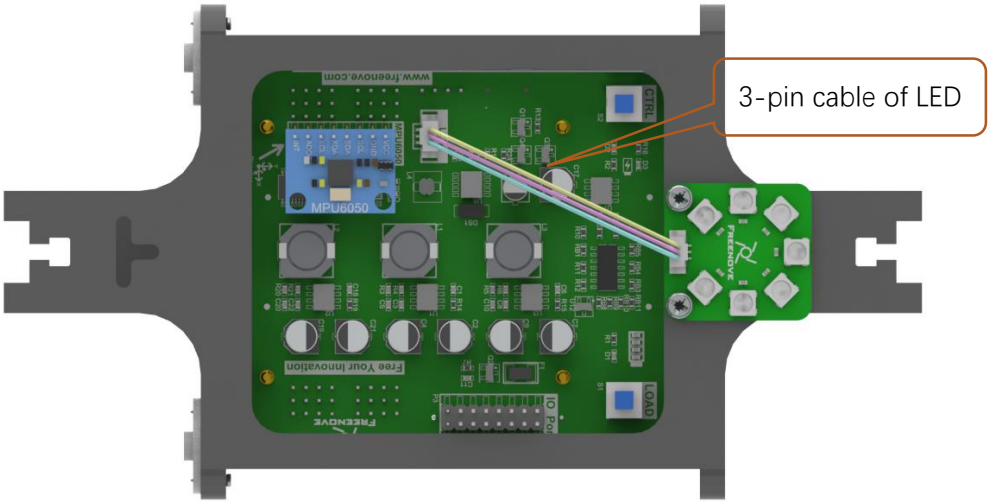
Step 5 Install LED module



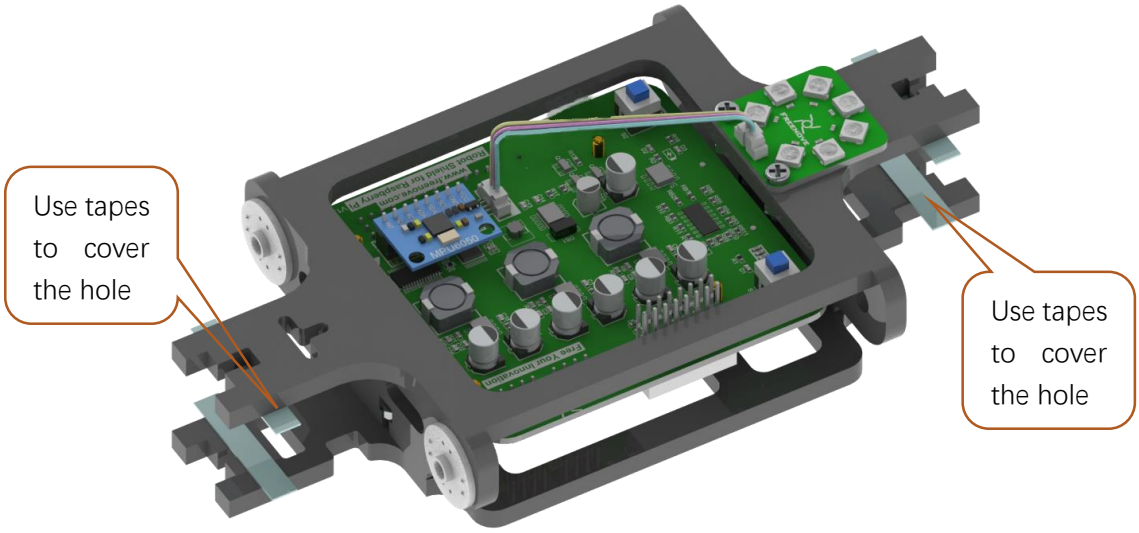
Install the top bracket.



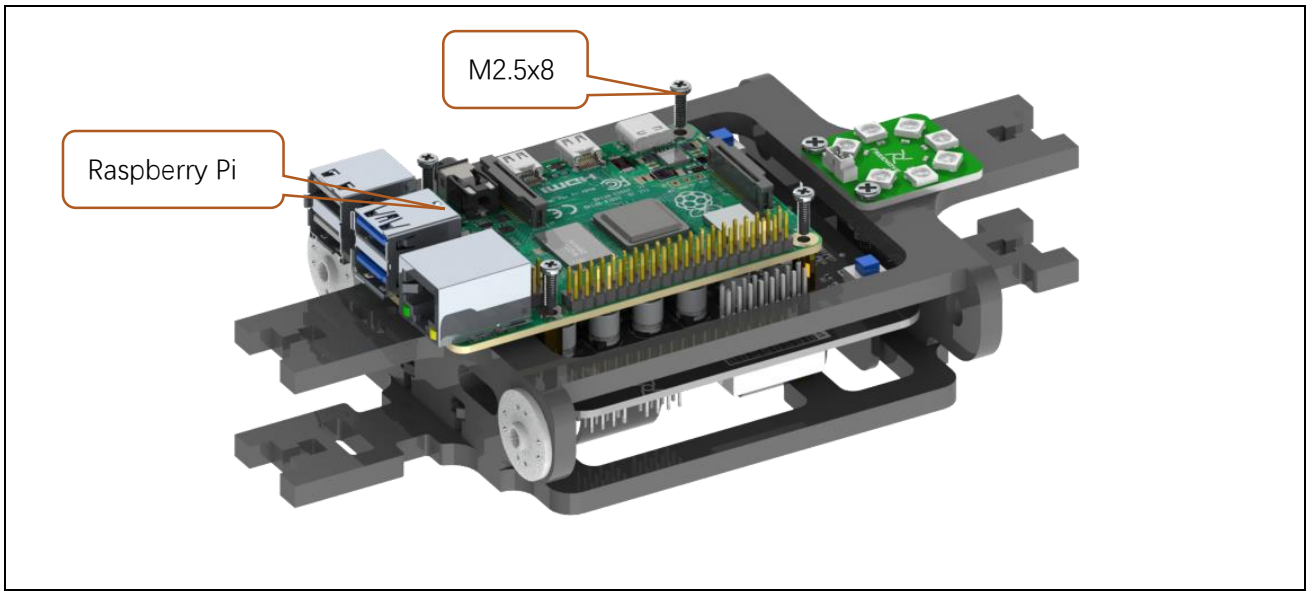
Connect shield with LED module.



This step will be very helpful for later assembly.

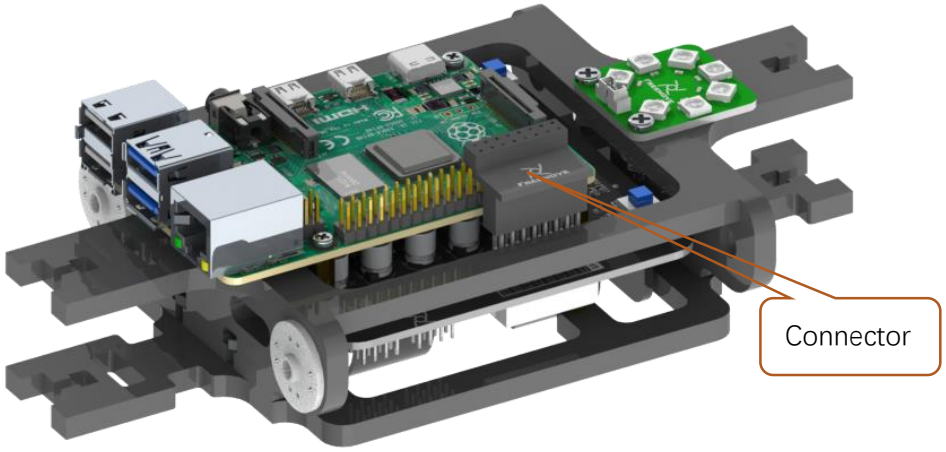


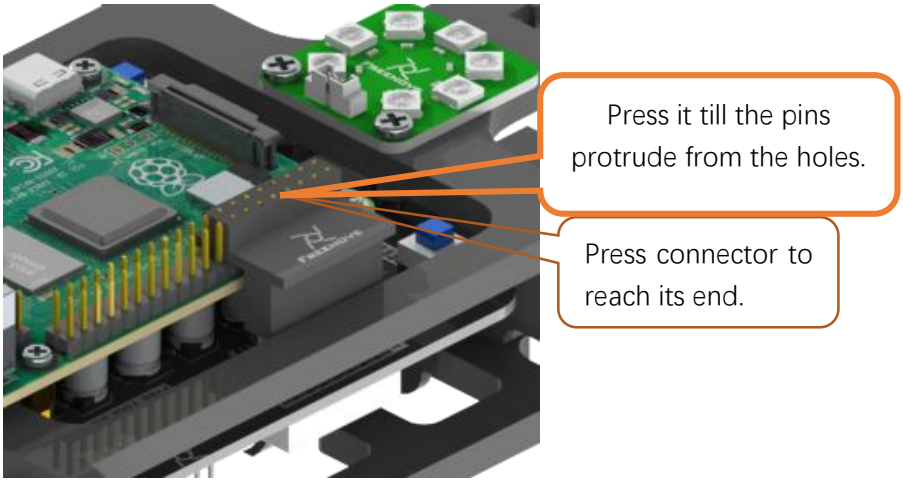
Step 6 Install Raspberry Pi

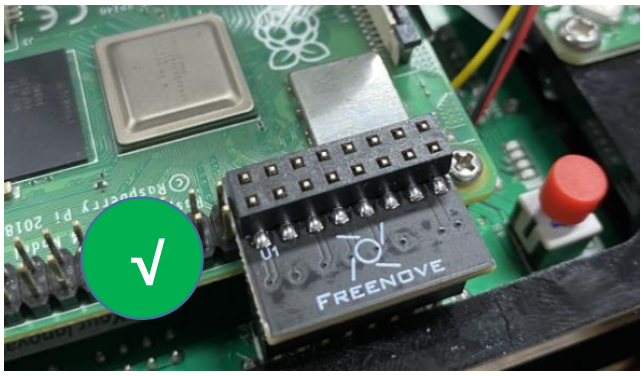


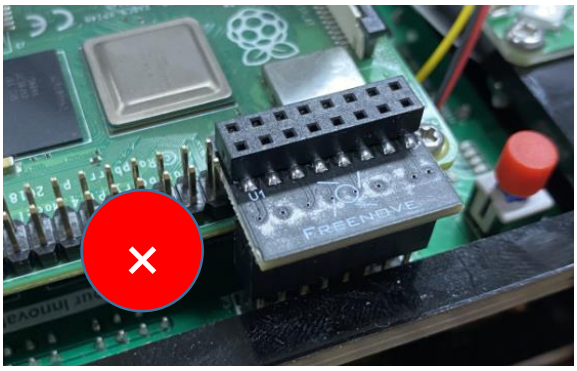
Step 7 Install Connector

Install connector to connect Raspberry Pi and shield.

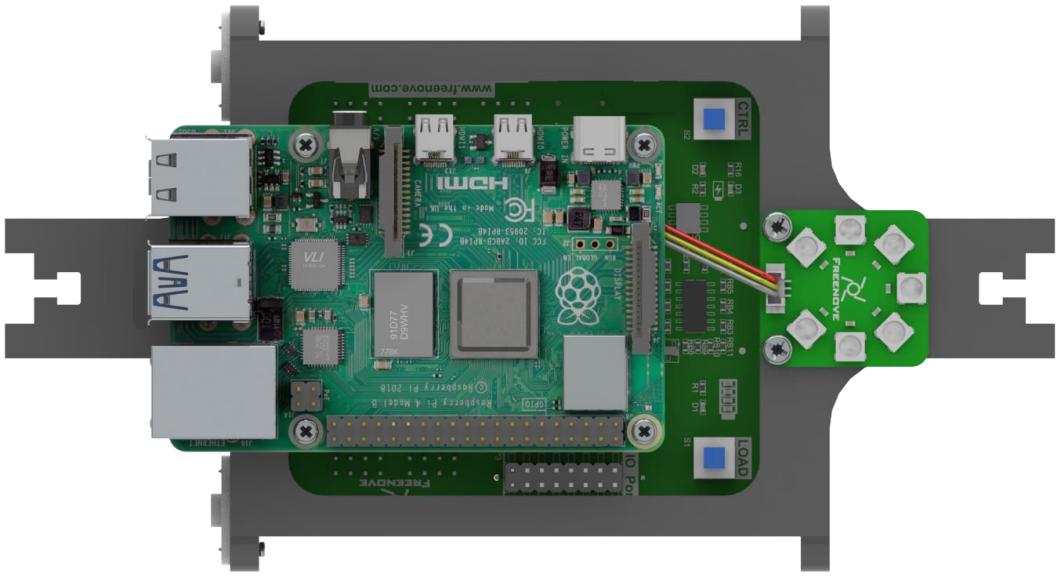






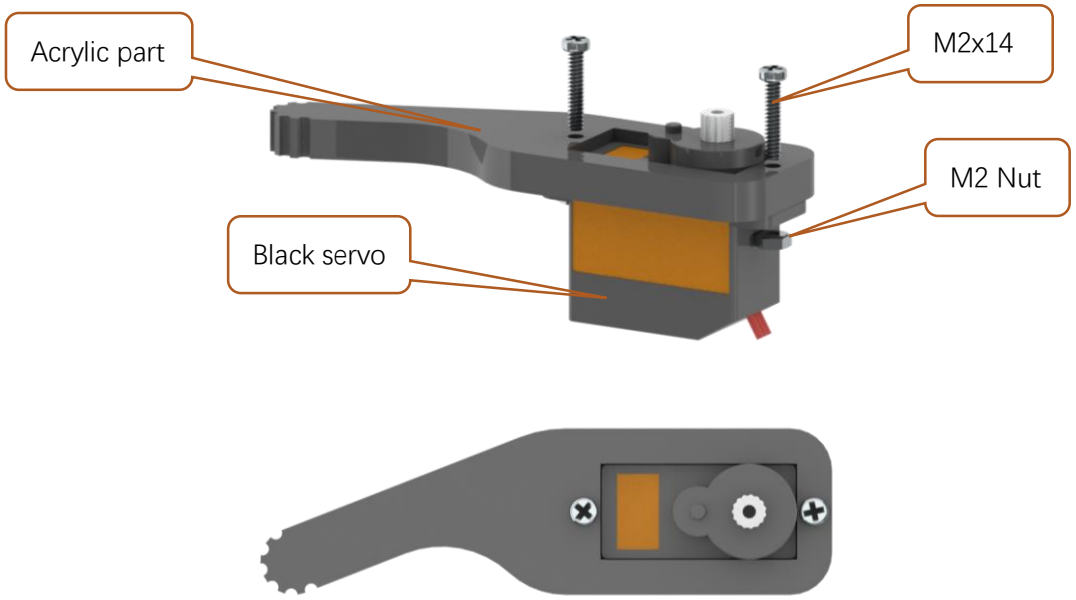


If you have any concerns, please feel free to contact us via support@freenove.com
We will offer you satisfying solution.

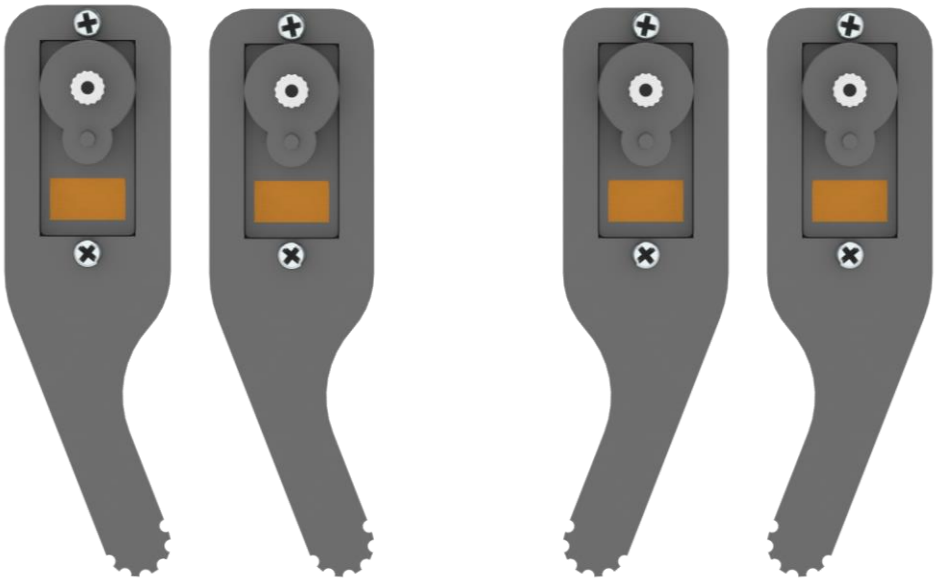


Now we have assembled the main body part. We will assemble leg parts later.

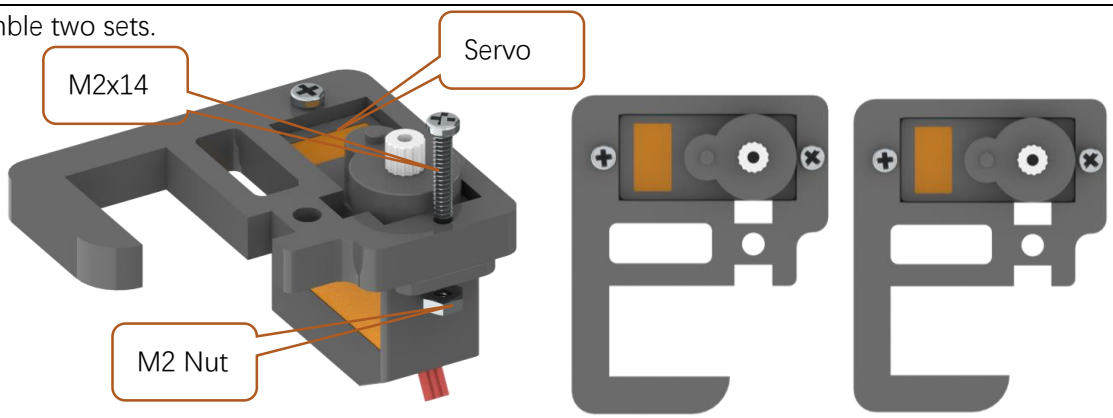
Step 8 Install Servo to Acrylic Board



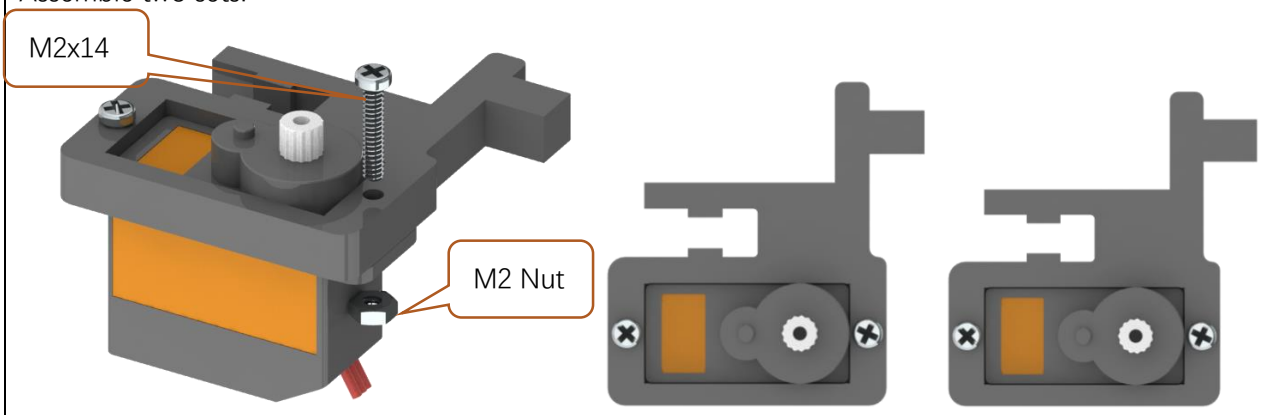
Note: There are four parts. They are different .



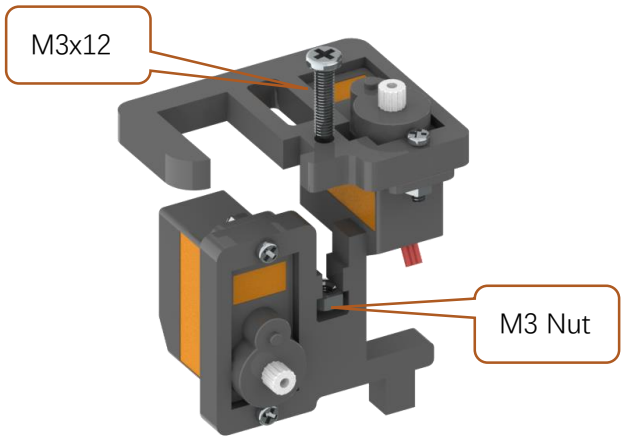
Assemble two sets.



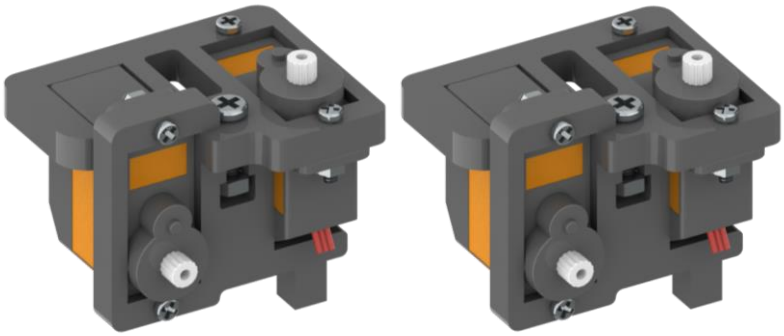
Assemble two sets.



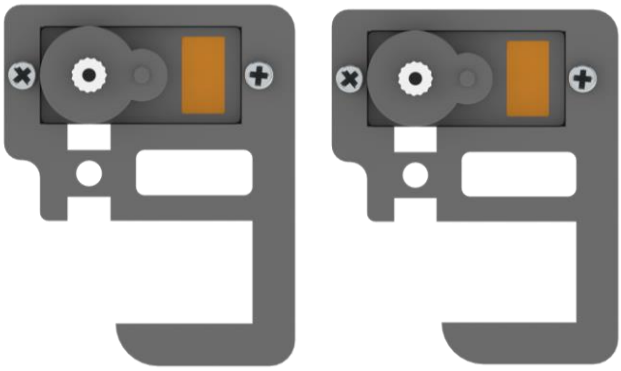
Assemble parts to the top.



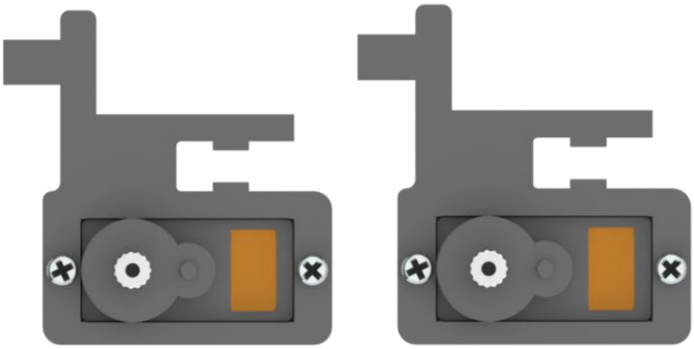
Now you get two sets of joint parts.



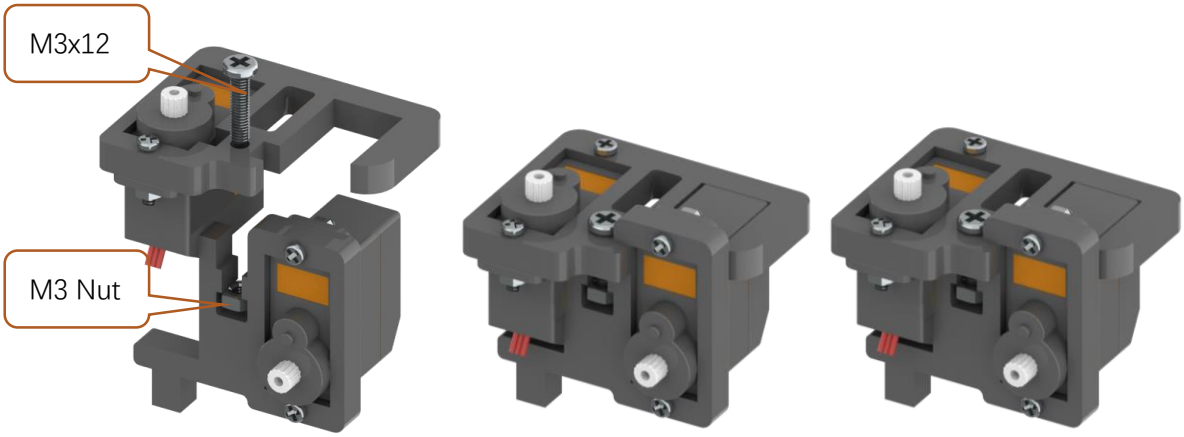
Assemble another two parts. **Note: they are different from parts above.**



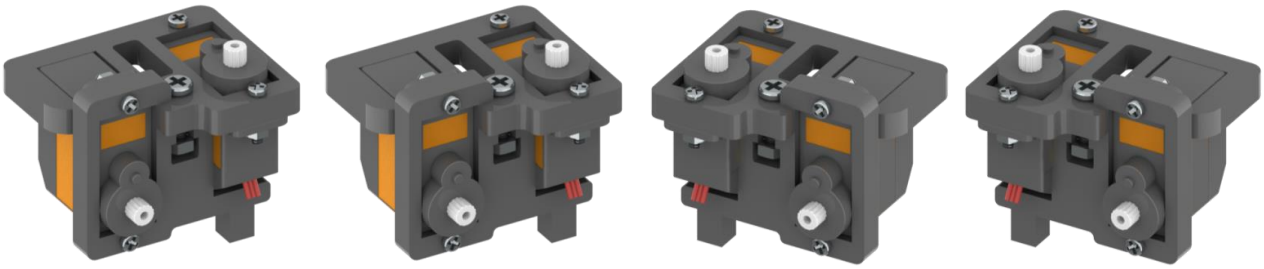
Assemble another two parts. **Note: they are different from parts above.**



Asseble them.

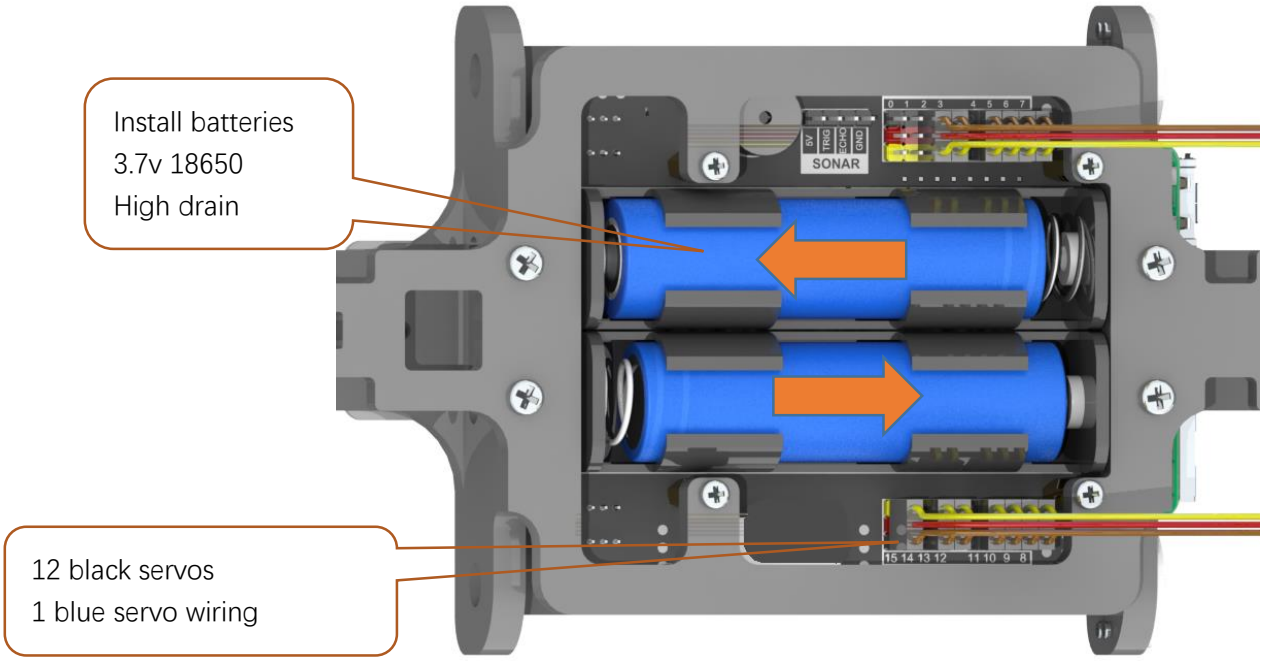


Now you will get four parts as below. **Note, they are different.**

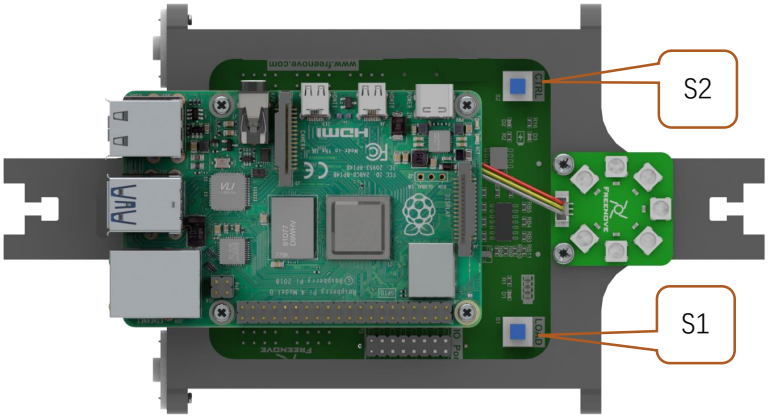


Step 9 Run Servo Program (Necessary)

Connect **All** 12 **black** servos and 1 **blue** servo to servo port **Randomly**. And install **batteries**. Push batteries to "+" ends.



You can still use cable to power Raspberry Pi with switches ON. If you disconnect cable, the batteries will power Raspberry Pi. Press **S1 and S2** and there will be two indicators ON.



Turn on the Raspberry Pi. Make sure all three steps in **Chapter 1** have been performed correctly. If you haven't done that, please perform the steps in **Chapter 1 first**.

a. Type the following command to enter servo code folder "Server".

```
cd ~/Freenove_Robot_Dog_Kit_for_Raspberry_Pi/Code/Server
```

b. Run Servo.py

```
sudo python Servo.py
```

After running the program, all servos will keep at 90°.

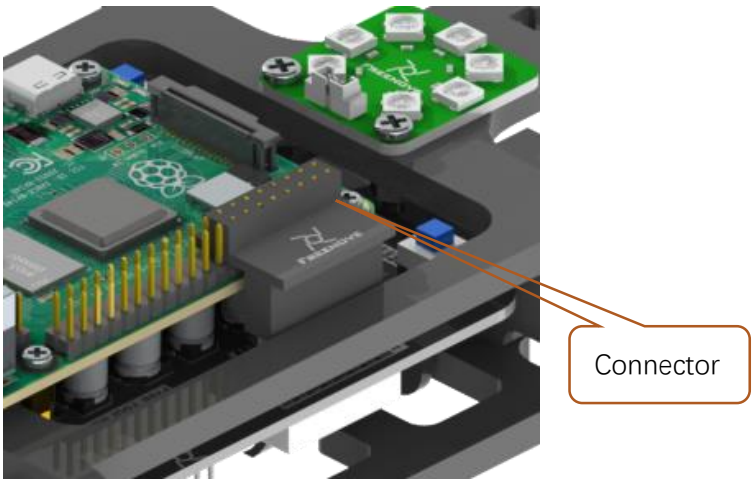
Next we will connect the servos to disk arm.

Keep the **power on** and all servos connected **during the assembly.** Don't disconnect **servos**.

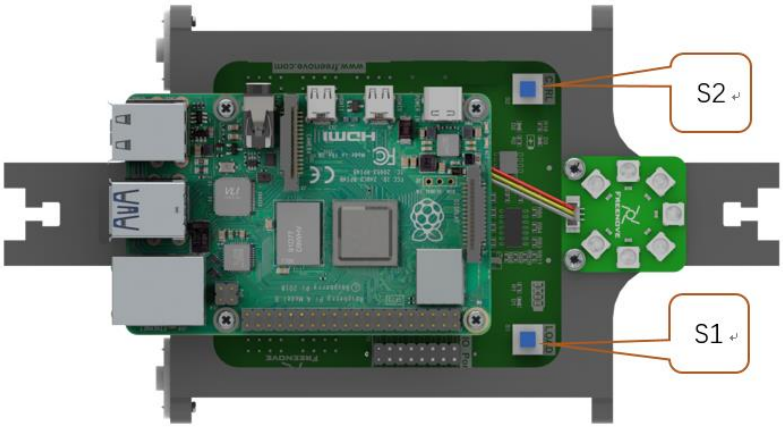
https://www.youtube.com/watch?v=7GkMYdVrong

When running the code, if it reports "Remote I/O error", please check the following three aspects:

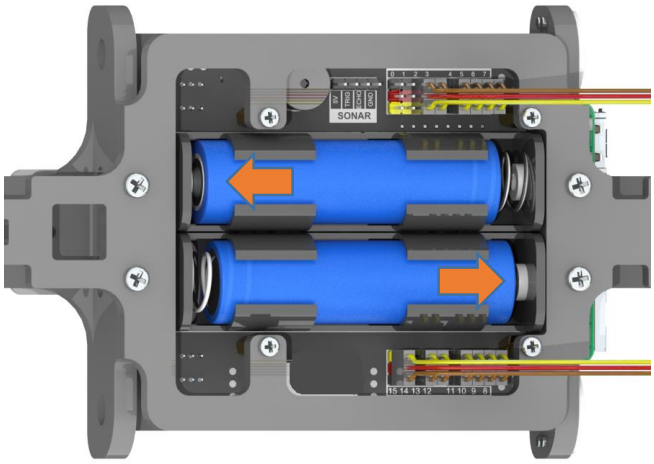
1. **Whether the connector is connected well**



2. **Whether both S1 and S2 are turned ON**



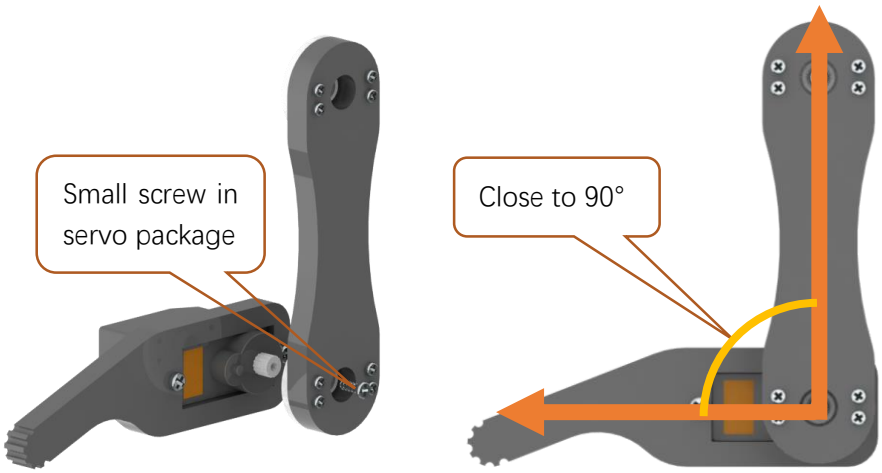
3. Whether the batteries are contacted well with the battery holder



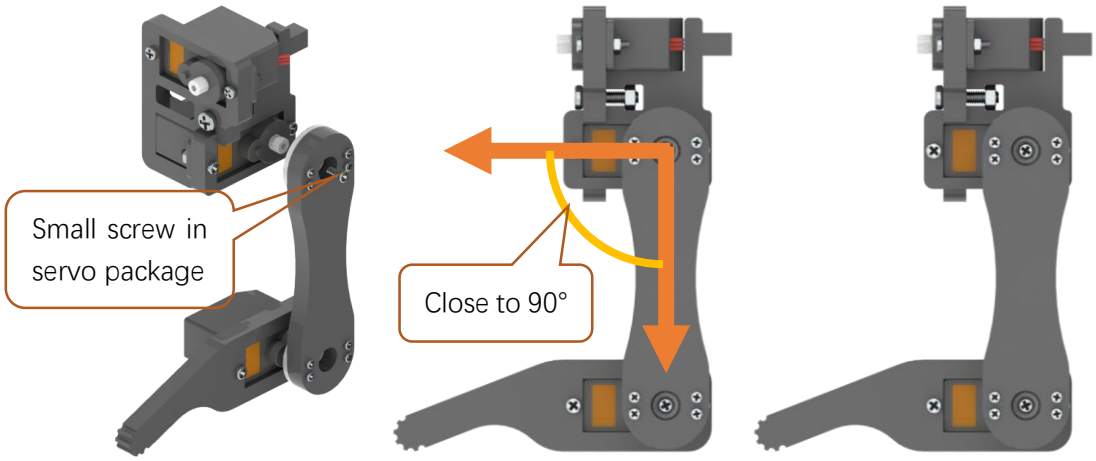
Step 10 Assemble Legs to Body

Keep the power ON and all servos connected to shield. Do NOT disconnect wiring. We need to keep the servo at 90° when connected to them to servo arms. (servo wires are not shown in the following content.)

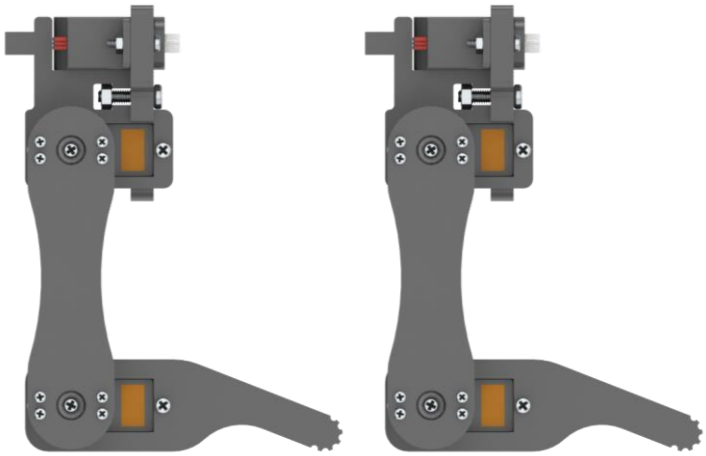
Try to install them close to 90°. The angles are acceptable within 65~115.



Assemble legs of one side. Try to install them at 90°. The angles are acceptable within 65~115.

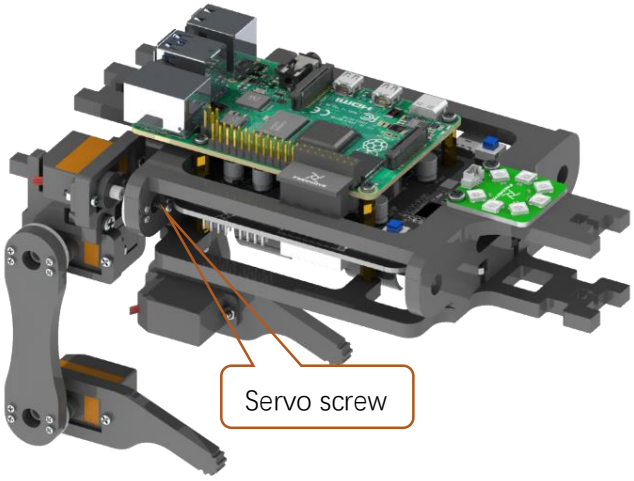


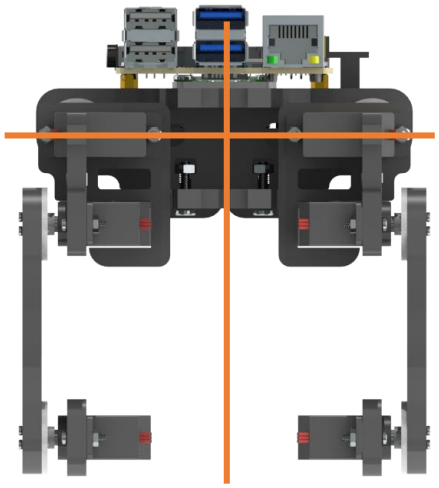
Assemble legs of the other side.



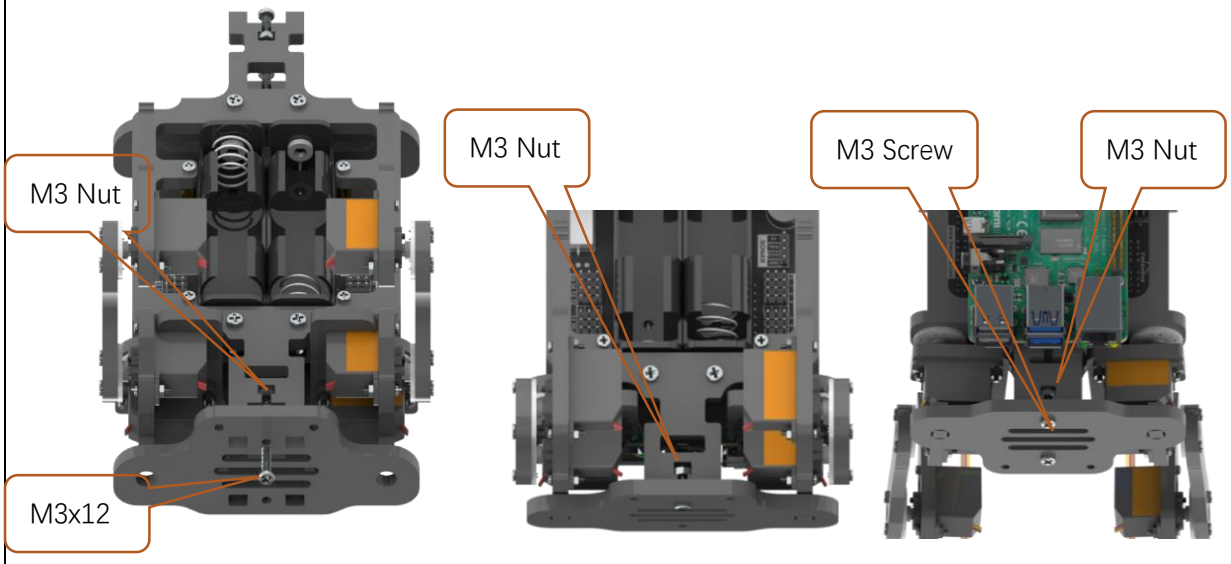
Keep the power ON and all servos connected to shield. Do NOT disconnect wiring.

Select two legs of **different sides** as rear legs. And install them as below. Then install the servo screws.

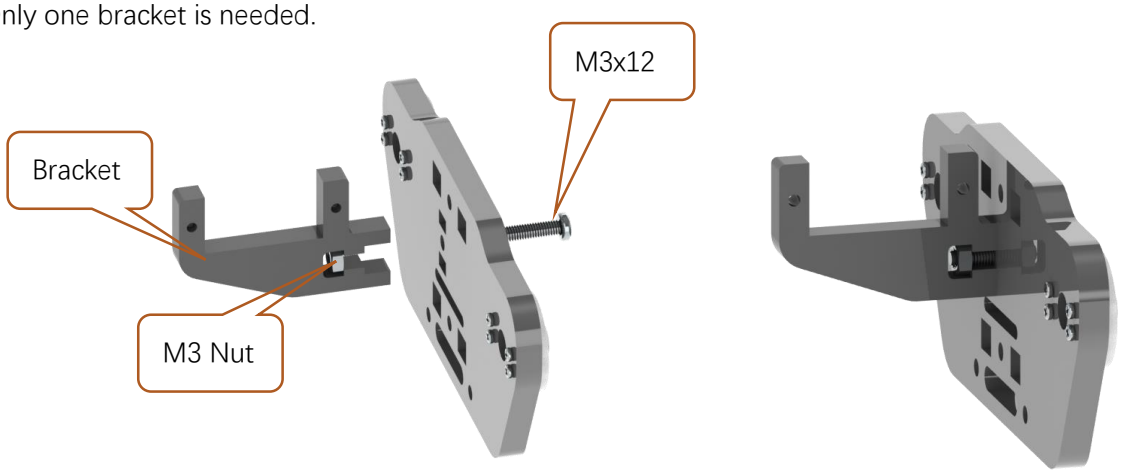




In this step, if you did not cover tape to the holes before, you can turn off **Load S1** to install the screw. After the screw is installed, turn on the **Load S1**.

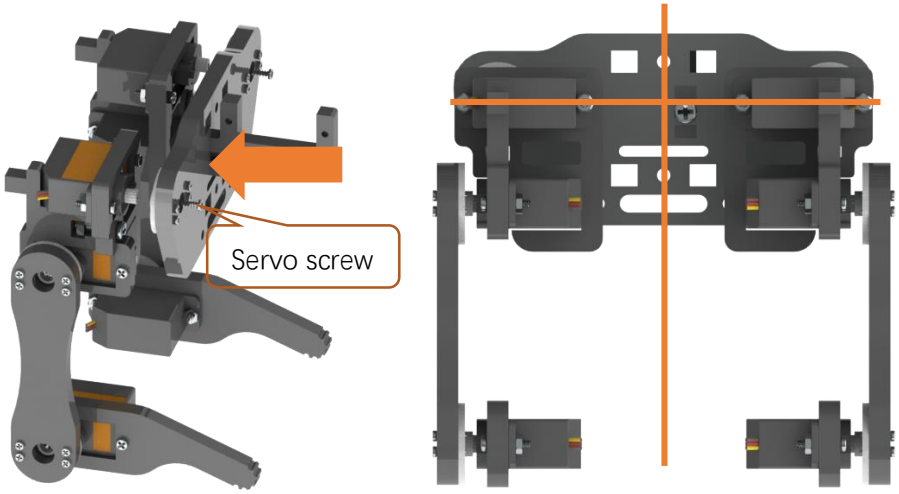


Only one bracket is needed.

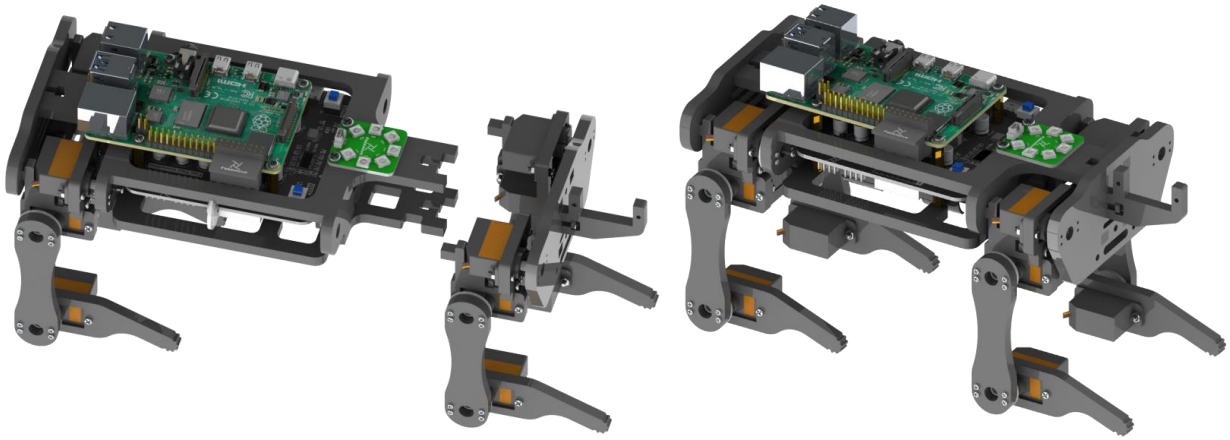


Keep the power ON and all servos connected to shield. Do NOT disconnect wiring.

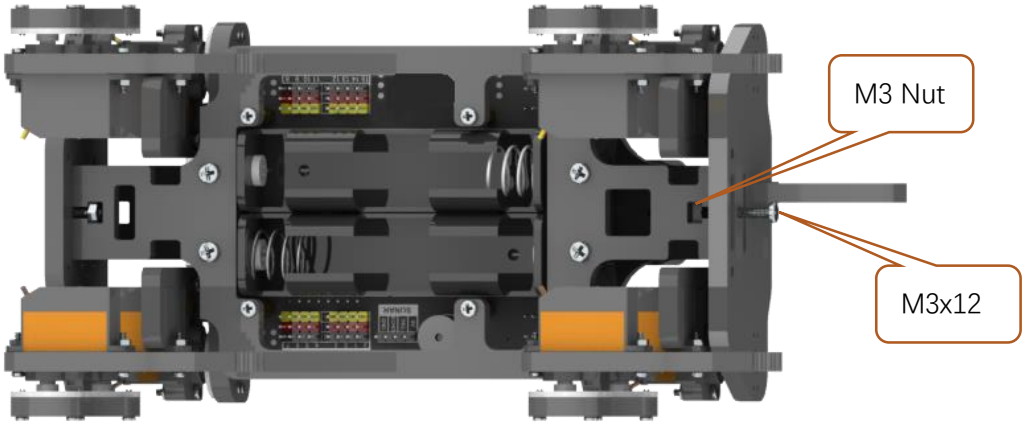
Assemble front legs and install the servo screw.

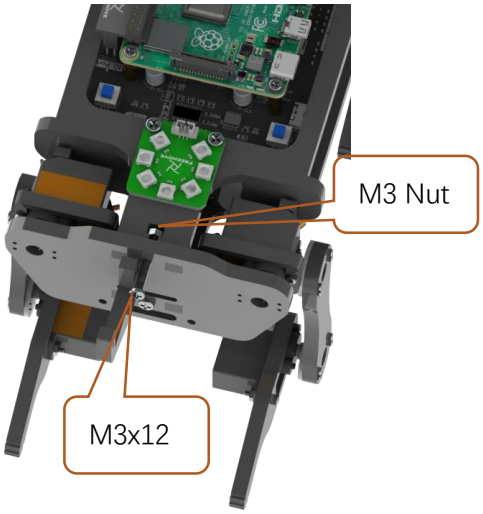


Assemble them.

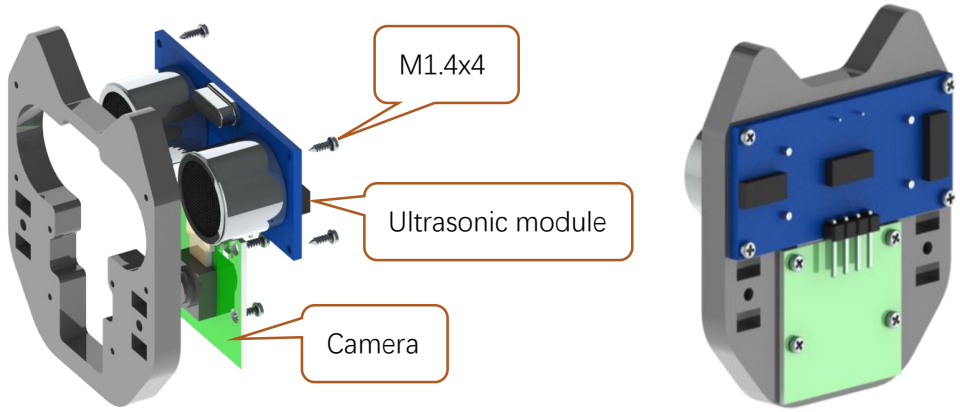


In this step, if you did not cover tape to the holes before, you can turn off **Load S1** to install the screw. After the screw is installed, turn on the **Load S1**.

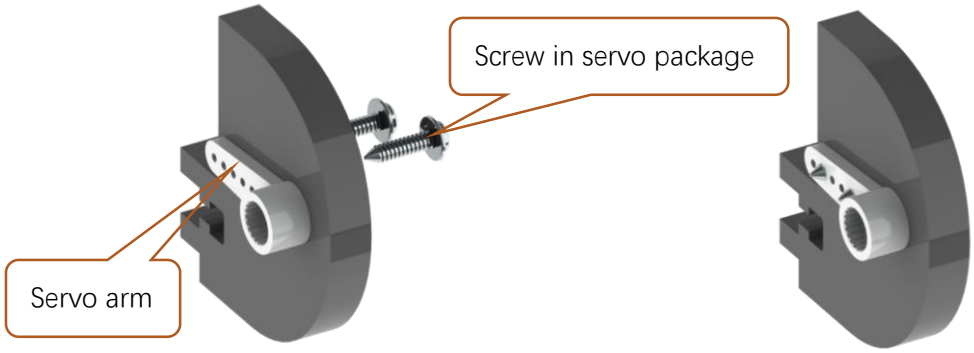


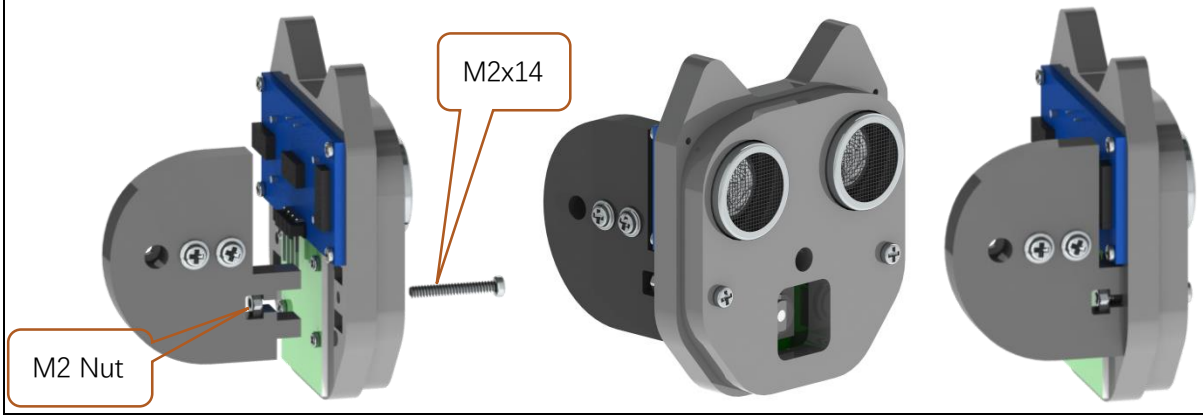

M3 Nut
M3x12

Step 11 Assemble Head

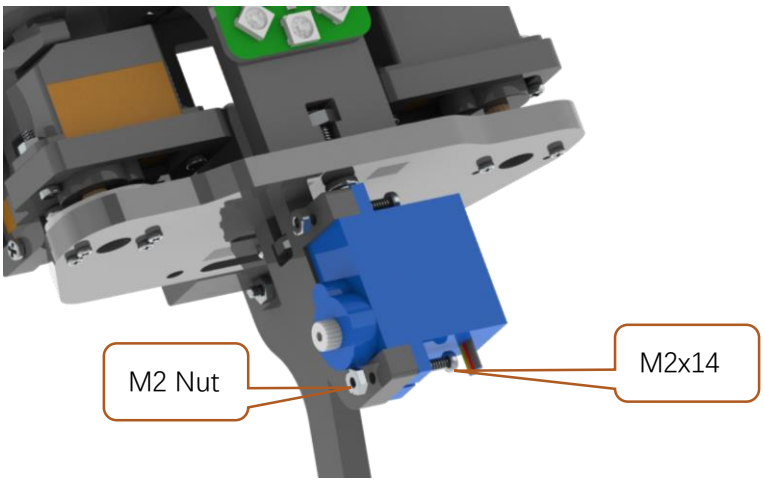


The servo arm and screws are included in the blue servo package.

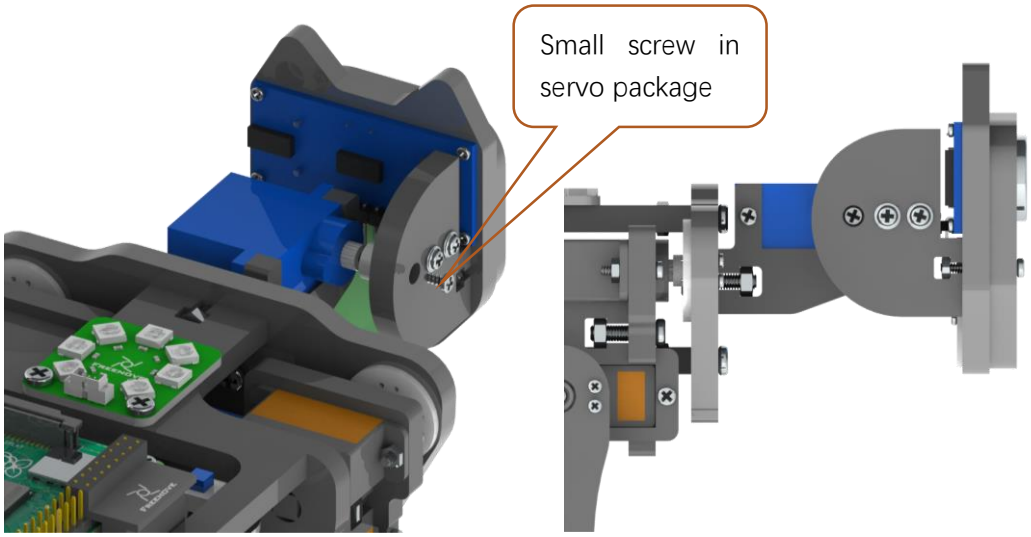


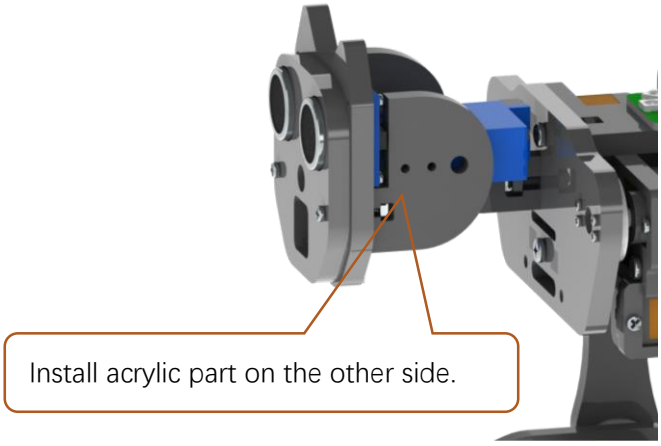


Step 12 Assemble Head to Body



Keep the power ON and servos connected when installing arm to servo.

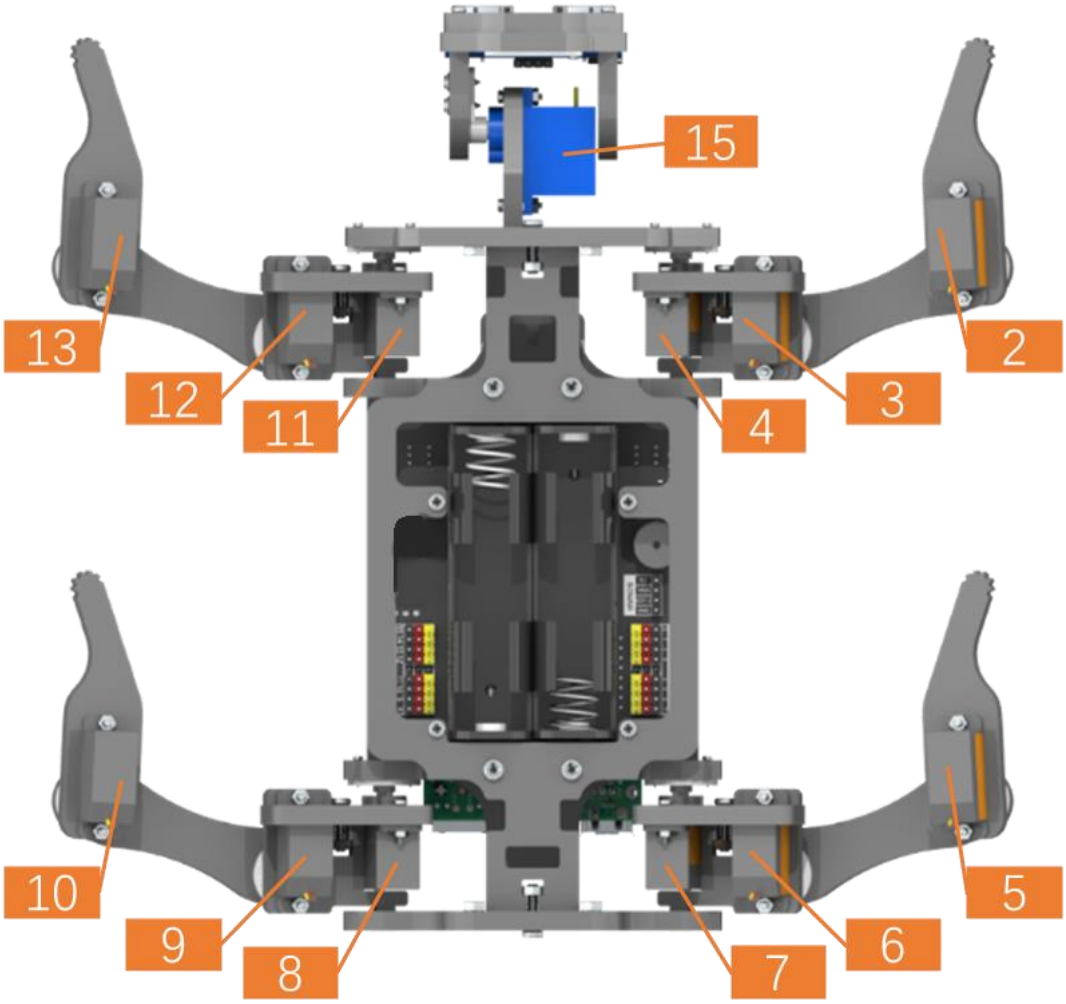


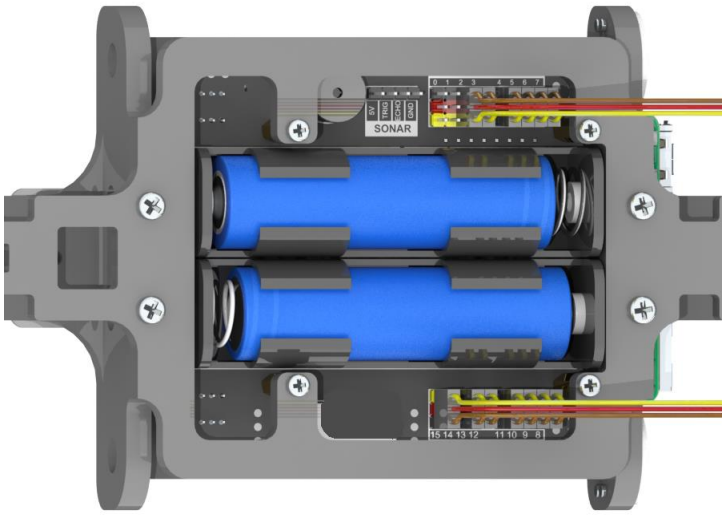


We will complete wiring in next section. Shut down the Raspberry Pi first. Do NOT turn on Raspberry Pi until chapter 3.

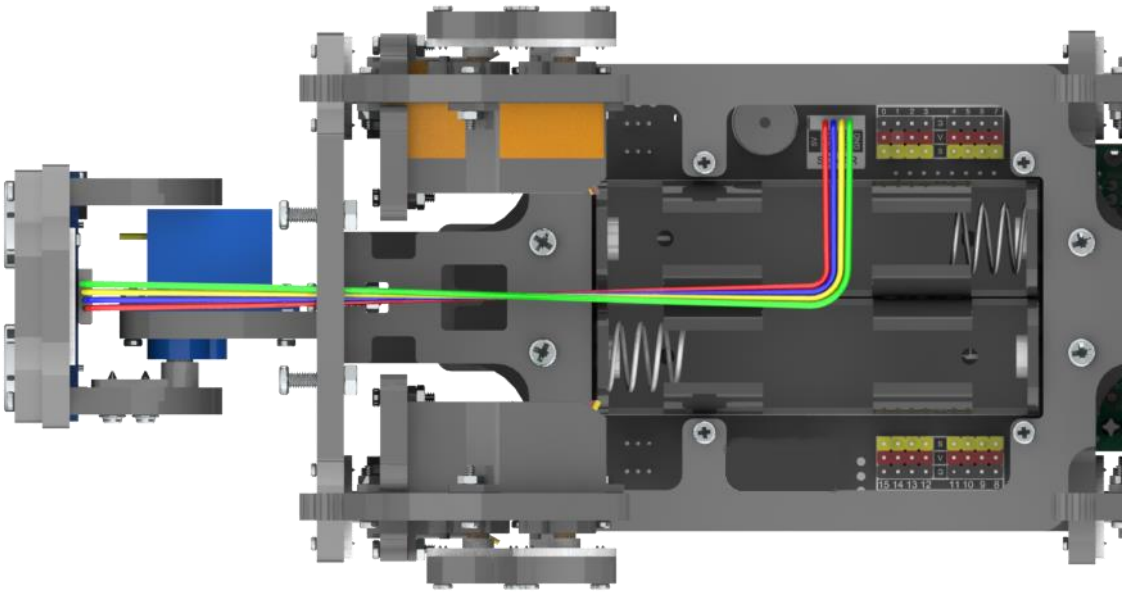
Step 13 Wiring

Connect servos according to the numbers. **Note servo ports 0, 1, 14 are not connected to servo**. They are spare.



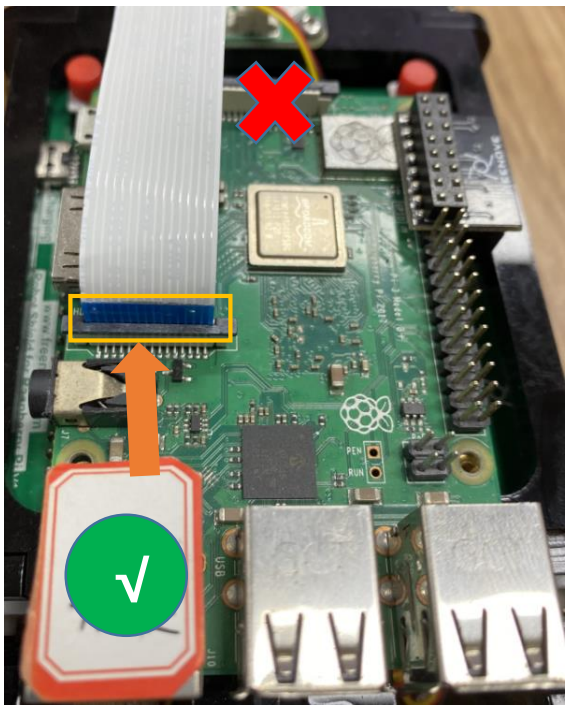


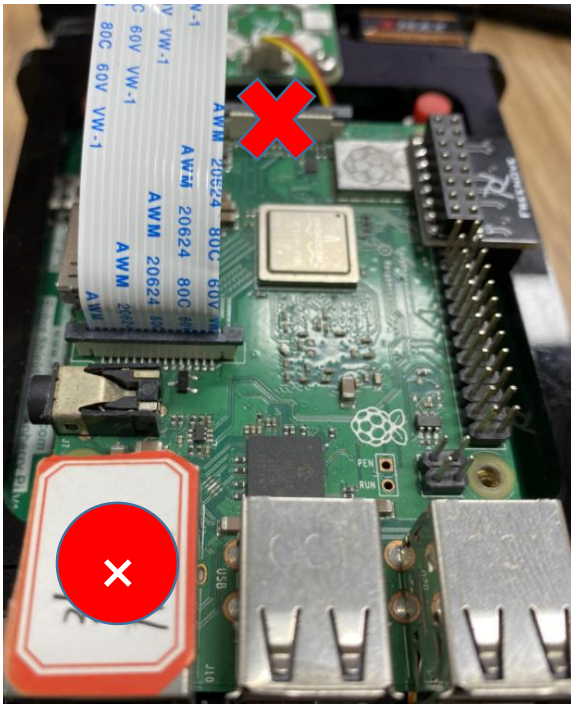
Wiring of Ultrasonic module (Note: **Do NOT connect wrongly. If you connect 5V to GND, it may damage the ultrasonic module**)



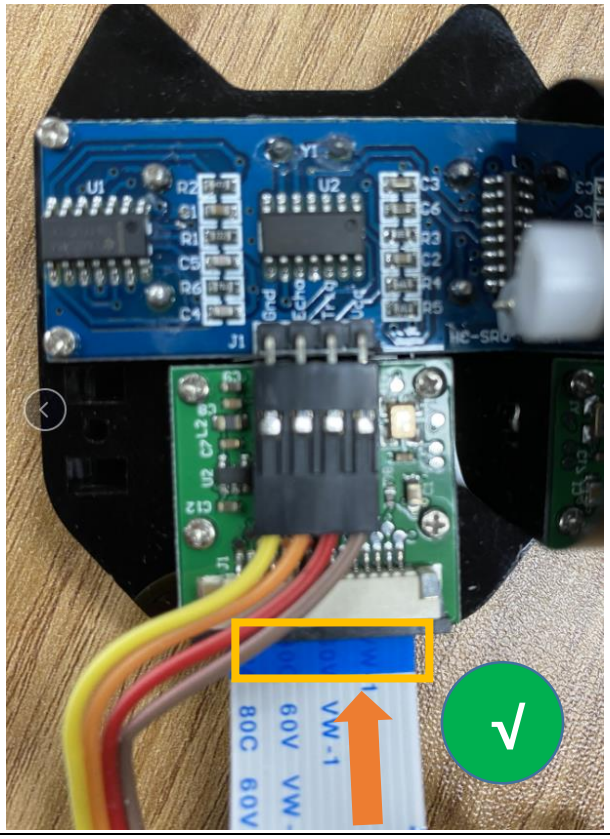
Pay attention to the **blue side** of camera cable.

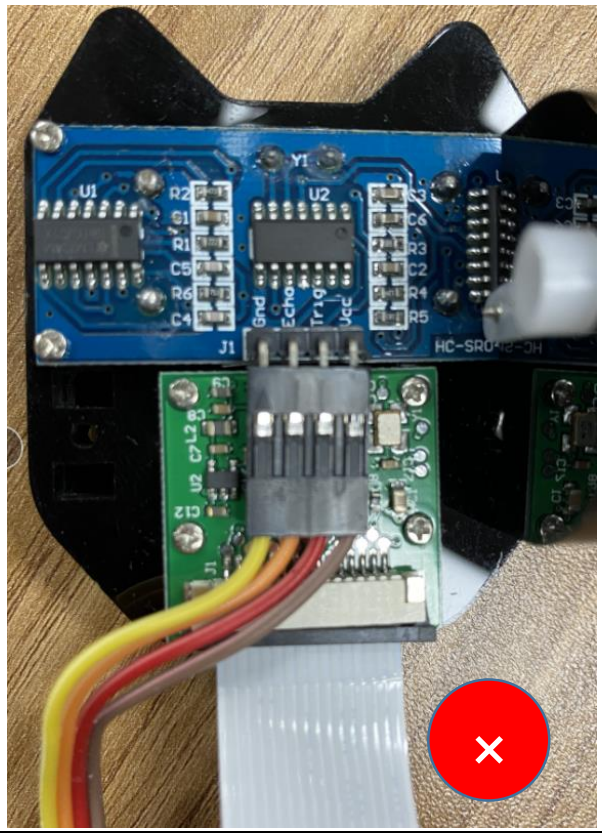
(Note: plugging and unplugging the cable requires the Raspberry Pi to be powered off, otherwise the camera module may be burned.)



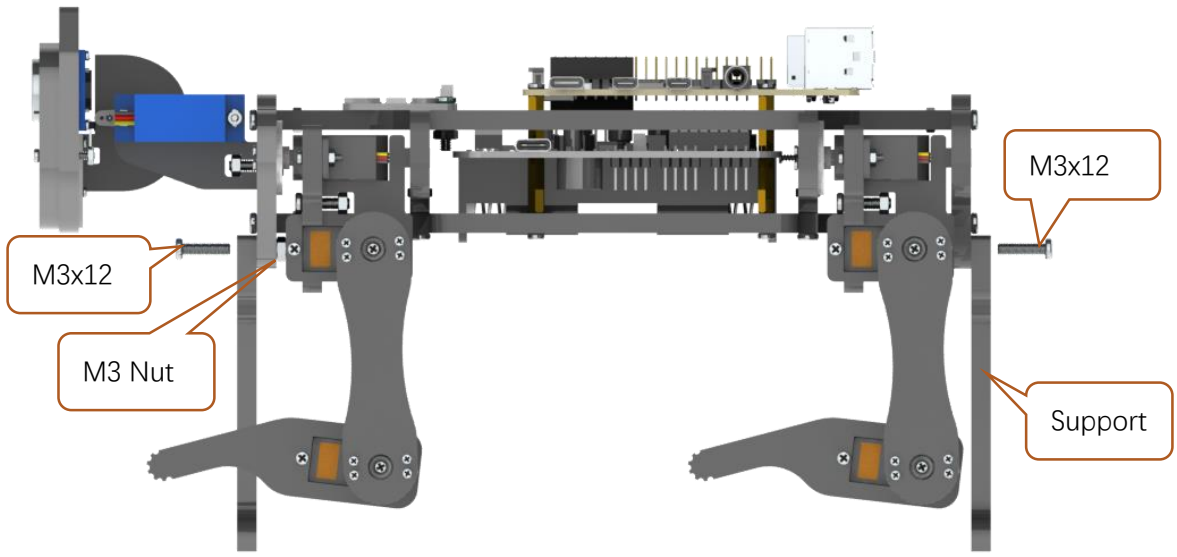


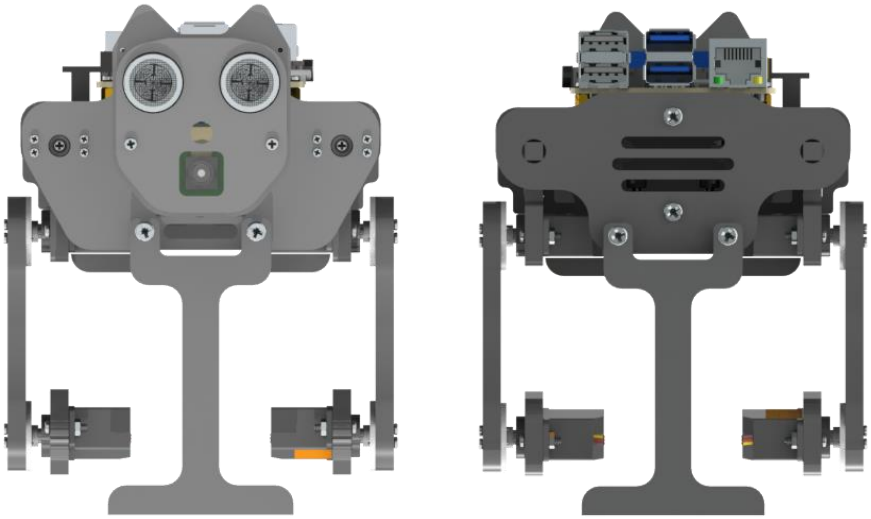
Pay attention to the **blue side** of camera cable.





Step 14 Install Calibration Support





Step 15 Verify Assembly

Turn on two switches and run following two commands again.

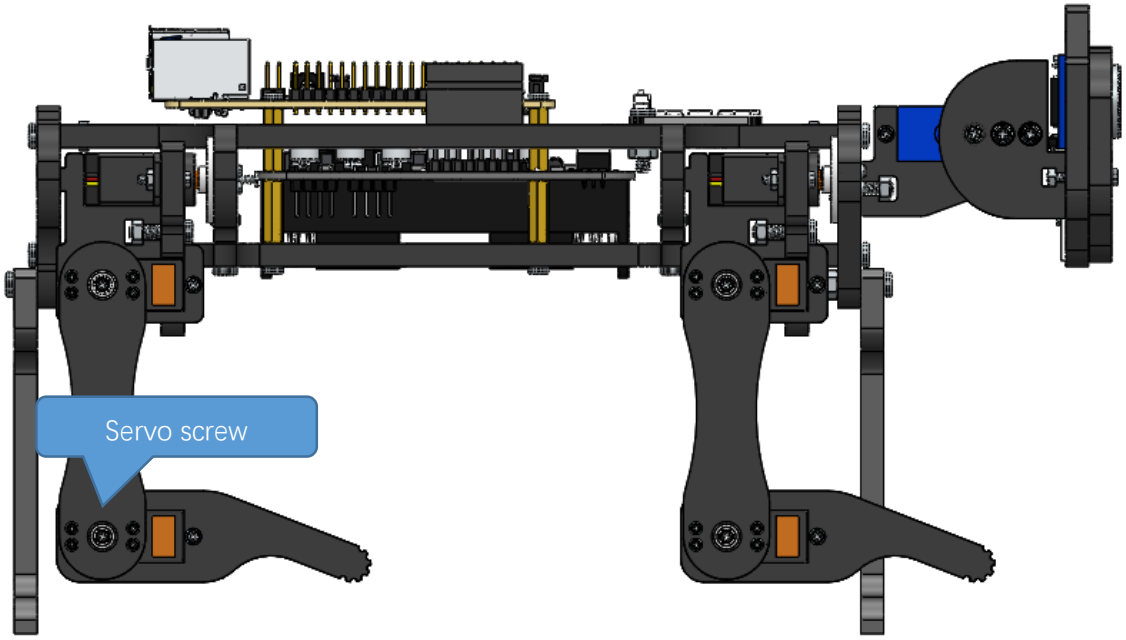
a. Type the following command to enter servo code folder “Server”.

```
cd ~/Freenove_Robot_Dog_Kit_for_Raspberry_Pi/Code/Server
```

b. Run Servo.py

```
sudo python Servo.py
```

The robot should show the following posture.



If it does not look like this, you need to disassemble servo screws of incorrect servos. Then reassemble the incorrect servo to make the robot look like posture above.

Chapter 3 Module test (Required)

If you have any concerns, please feel free to contact us at support@freenove.com

The robot dog has been assembled in the previous chapter. This chapter will test each module of the robot dog. It is a necessary check for later control.

Before starting, please install the battery for the robot dog and **turn on the S1 and S2 switches**. Place the robot dog with the calibration supports on a horizontal table to ensure that the wiring of each servo is correct.

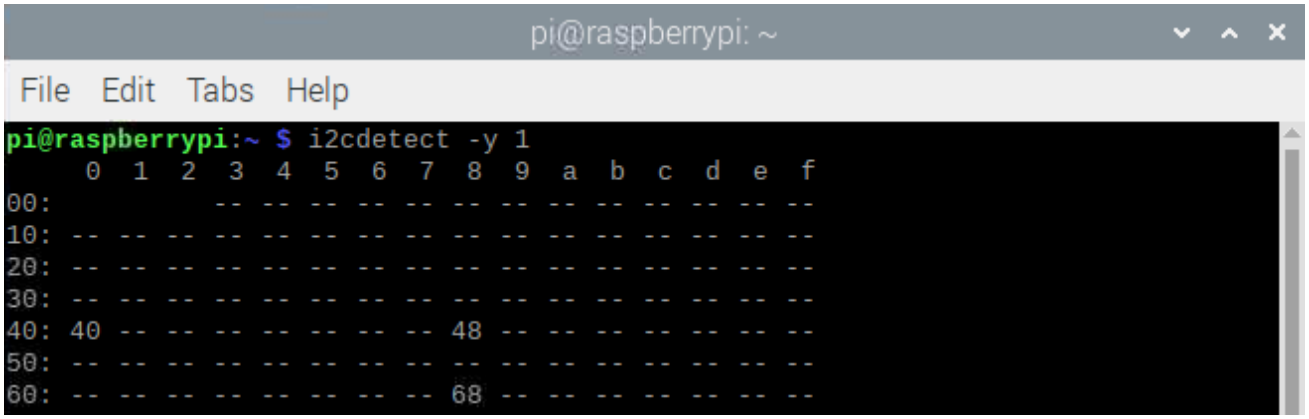
Please use VNC Viewer to build this robot when you use remote mode to login RPi, or there will be errors. Because other desktop remote doesn't support GUI.

Servo

Execute following command to check i2c address.

```
i2cdetect -y 1
```

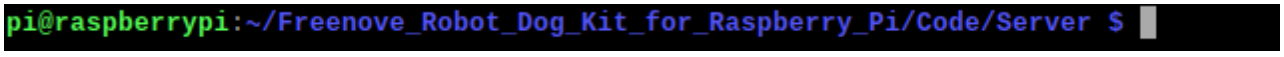
As shown in the figure below, the addresses 0x40, 0X48, and 0X68 corresponds to the PCA9685 chip, ADS7830 chip, and MPU6050 module, respectively.



Run program

Enter following commands in the terminal to test servos.

If the terminal displays the directory as below (where test.py is located). You can **directly** execute the test.py command.



1. If not, execute the cd command:

```
cd ~/Freenove_Robot_Dog_Kit_for_Raspberry_Pi/Code/Server
```

2. Execute test.py command:

```
sudo python test.py Servo
```

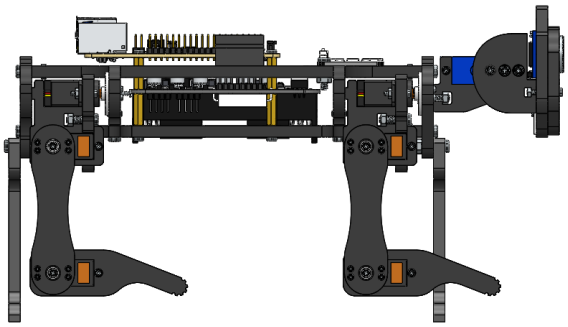
```
pi@raspberrypi: ~/Freenove_Robot..._Kit_for_Raspberry_Pi/Code/Server
File  Edit  Tabs  Help
pi@raspberrypi:~ $ cd ~/Freenove_Robot_Dog_Kit_for_Raspberry_Pi/Code/Server
pi@raspberrypi:~/Freenove_Robot_Dog_Kit_for_Raspberry_Pi/Code/Server $ sudo pyth
on test.py Servo
Program is starting ...
```

Result:

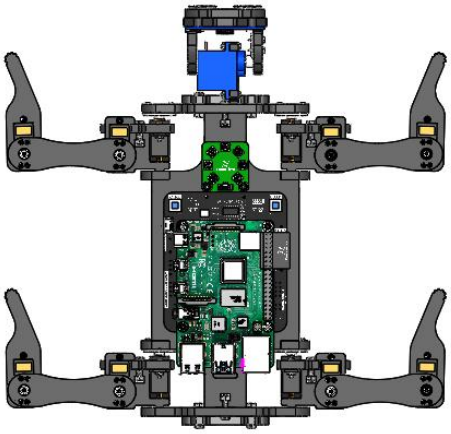
After assembly in the previous chapter, the robot should look as shown in picture A.

After servo test program is executed, the robot's posture will change to A, B, C, D gradually, which indicates the servo channel works normally.

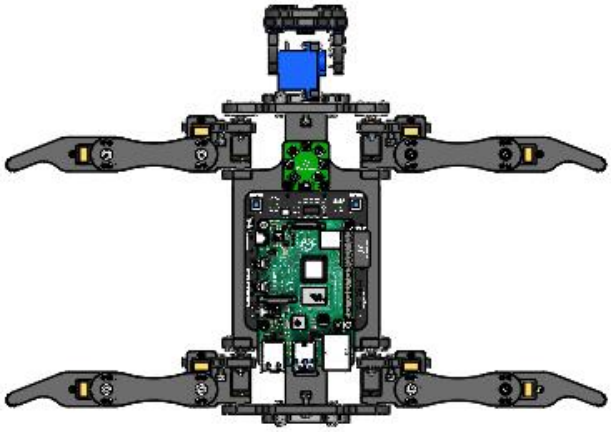
If the situation is not correct, check the servo wiring.



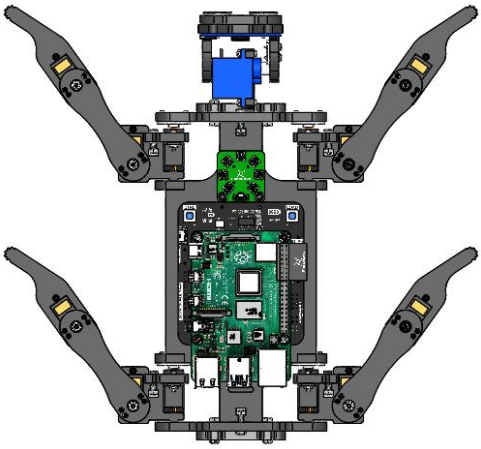
(A)



(B)



(C)



(D)

The code is as below:

```
from Servo import *
servo=Servo()
def test_Servo():
    try:
        for i in range(90):
            servo.setServoAngle(4,90-i)
            servo.setServoAngle(7,90-i)
            servo.setServoAngle(8,90+i)
            servo.setServoAngle(11,90+i)
            time.sleep(0.01)
        for i in range(90):
            servo.setServoAngle(2,90-i)
            servo.setServoAngle(5,90-i)
            servo.setServoAngle(10,90+i)
            servo.setServoAngle(13,90+i)
            time.sleep(0.01)
        for i in range(60):
            servo.setServoAngle(3,90-i)
            servo.setServoAngle(6,90-i)
            servo.setServoAngle(9,90+i)
            servo.setServoAngle(12,90+i)
            time.sleep(0.01)
        print ("\nEnd of program")
    except KeyboardInterrupt:
        print ("\nEnd of program")
```

Reference

setServoAngle(channel, angle)

This function has two parameters to control the Servo of the corresponding channel to the corresponding angle.

The first parameter indicates the servo channel

The second parameter is the rotation angle

Example:

setServoAngle (0,60) The Servo of channel 0 is turned to 60 degrees

setServoAngle (15,160) The Servo of channel 15 is turned to 160 degrees

ADC Module

Run program

Enter following commands to test ADC module.

If the terminal displays the directory as below (where test.py is located). You can **directly** execute the second command test.py.

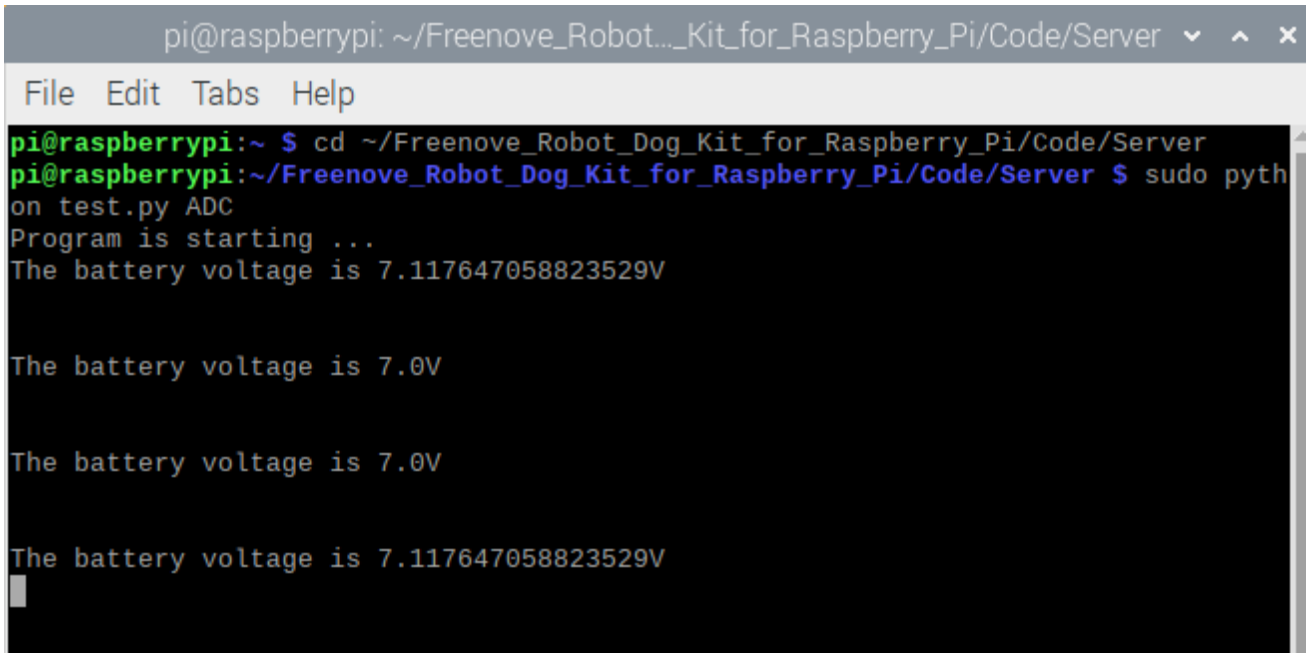
```
pi@raspberrypi:~/Freenove_Robot_Dog_Kit_for_Raspberry_Pi/Code/Server $
```

1.If not, execute the cd command:

```
cd ~/Freenove_Robot_Dog_Kit_for_Raspberry_Pi/Code/Server
```

2.Execute test.py command:

```
sudo python test.py ADC
```



Result:

Print voltage value of batteries per second. You can press "Ctrl + C" to end the program.

The code is as below:

```
from ADS7830 import *
adc=ADS7830()
def test_Adc():
    try:
        while True:
            Power=adc.readAdc(0)/255.0*5.0*3
            print ("The battery voltage is "+str(Power)+"V")
            time.sleep(1)
            print ('\n')
    except KeyboardInterrupt:
        print ("\nEnd of program")
```

Reference

readAdc (channel)

This function has one parameter for reading the ADC value of the corresponding channel.

For example:
readAdc (0)
Read the ADC value of the batteries voltage. The returned number ranges between 0-255. Divide the obtained value by 255.0 and then multiply by the reference voltage of 5.0V, and then triple the value to get the batteries voltage.

Ultrasonic module

Run program

If the terminal displays the directory as below (where test.py is located). You can **directly** execute the test.py command.

```
pi@raspberrypi:~/Freenove_Robot_Dog_Kit_for_Raspberry_Pi/Code/Server $
```

1. If not, execute the cd command:

```
cd ~/Freenove_Robot_Dog_Kit_for_Raspberry_Pi/Code/Server
```

2. Execute test.py command:

```
sudo python test.py Ultrasonic
```

```
pi@raspberrypi: ~/Freenove_Robot_...g_Kit_for_Raspberry_Pi/Code/Server
File  Edit  Tabs  Help
pi@raspberrypi:~ $ cd ~/Freenove_Robot_Dog_Kit_for_Raspberry_Pi/Code/Server
pi@raspberrypi:~/Freenove_Robot_Dog_Kit_for_Raspberry_Pi/Code/Server $ sudo python
test.py Ultrasonic
Program is starting ...
Obstacle distance is 40CM
Obstacle distance is 60CM
Obstacle distance is 63CM
Obstacle distance is 61CM
Obstacle distance is 3CM
Obstacle distance is 13CM
Obstacle distance is 14CM
Obstacle distance is 42CM
^C
End of program
pi@raspberrypi:~/Freenove_Robot_Dog_Kit_for_Raspberry_Pi/Code/Server $
```

Result:

Every 1s, the distance between the obstacle and the ultrasonic module will be printed in, and you can press "Ctrl + C" to end the program.

The code is as below:

```
1  from Ultrasonic import *
2  ultrasonic=Ultrasonic()
3  def test_Ultrasonic():
4      try:
5          while True:
6              data=ultrasonic.getDistance()   #Get the value
7              print ("Obstacle distance is "+str(data)+"CM")
8              time.sleep(1)
9      except KeyboardInterrupt:
10         print ("\nEnd of program")
```

Reference

getDistance()
This function is used to obtain the distance between ultrasonic module and the obstacle in front of it in centimeters.

LED

There is an RGB LED module on the robot as below:

Run program

Enter the following commands to test LED.

If the terminal displays the directory as below (where test.py is located). You can **directly** execute the test.py command.

```
pi@raspberrypi:~/Freenove_Robot_Dog_Kit_for_Raspberry_Pi/Code/Server $
```

1.If not, execute the cd command:

```
cd ~/Freenove_Robot_Dog_Kit_for_Raspberry_Pi/Code/Server
```

2.Execute test.py command:

```
sudo python test.py Led
```

```
pi@raspberrypi: ~/Freenove_Robot_...g_Kit_for_Raspberry_Pi/Code/Server
File  Edit  Tabs  Help
pi@raspberrypi:~ $ cd ~/Freenove_Robot_Dog_Kit_for_Raspberry_Pi/Code/Server
pi@raspberrypi:~/Freenove_Robot_Dog_Kit_for_Raspberry_Pi/Code/Server $ sudo python
test.py Led
Program is starting ...

Red wipe

Green wipe

Blue wipe

White wipe

End of program
pi@raspberrypi:~/Freenove_Robot_Dog_Kit_for_Raspberry_Pi/Code/Server $
```

Result:

The LED module will show red, green, blue and white in turn. Then all LED are turned OFF.

You can end the program ahead of time by pressing "ctrl+c".

If the LED color display order is not correct, open the "**Led.py**" file in the current directory and modify the value of the "self.ORDER" variable on line 16.

The code of test.py is as below:

```
from Led import *
led=Led()
def test_Led():
    try:
        #Red wipe
        led.colorWipe(led.strip, Color(255, 0, 0))
        time.sleep(1)
        print ("\nRed wipe")

        #Green wipe
        led.colorWipe(led.strip, Color(0, 255, 0))
        time.sleep(1)
        print ("\nGreen wipe")

        #Blue wipe
        led.colorWipe(led.strip, Color(0, 0, 255))
        time.sleep(1)
        print ("\nBlue wipe")

        #White wipe
        led.colorWipe(led.strip, Color(255, 255, 255))
        time.sleep(1)
        print ("\nWhite wipe")

        led.colorWipe(led.strip, Color(0, 0, 0))    #turn off the light
        print ("\nEnd of program")
    except KeyboardInterrupt:
        led.colorWipe(led.strip, Color(0, 0, 0))    #turn off the light
        print ("\nEnd of program")
```

Reference

colorWipe(strip, color, wait_ms)

This function erases the color of one pixel at a time. It has three input parameters, among which, strip represents the Neopixel object, color represents the color to be erased, and wait_ms represents the erasure interval. The default is 50ms. For example, colorWipe(strip, Color(255,0,0),20) means that the LED0 color is red first, waiting for 20ms, and then the LED1 color is also red, and repeat the same to all 8 LEDs until they are all red.

Buzzer

Run the program

Enter following command in the terminal to test buzzer.
If the terminal displays the directory as below (where test.py is located). You can **directly** execute the test.py command.

```
pi@raspberrypi:~/Freenove_Robot_Dog_Kit_for_Raspberry_Pi/Code/Server $
```

1.If not, execute the cd command:

```
cd ~/Freenove_Robot_Dog_Kit_for_Raspberry_Pi/Code/Server
```

2.Execute test.py command:

```
sudo python test.py Buzzer
```

```
pi@raspberrypi: ~/Freenove_Robot_...g_Kit_for_Raspberry_Pi/Code/Server
File  Edit  Tabs  Help
pi@raspberrypi:~ $ cd ~/Freenove_Robot_Dog_Kit_for_Raspberry_Pi/Code/Server
pi@raspberrypi:~/Freenove_Robot_Dog_Kit_for_Raspberry_Pi/Code/Server $ sudo python
test.py Buzzer
Program is starting ...
1S
2S
3S

End of program
pi@raspberrypi:~/Freenove_Robot_Dog_Kit_for_Raspberry_Pi/Code/Server $
```

Result:

The buzzer will be turned ON lasting for 3s. Then the program will automatically end or you can press "Ctrl + C" to end the program.

The code is as below:

```
from Buzzer import *
buzzer=Buzzer()
def test_Buzzer():
    try:
        buzzer.run('1')
        time.sleep(1)
        print ("1S")
        time.sleep(1)
        print ("2S")
        time.sleep(1)
        print ("3S")
        buzzer.run('0')
        print ("\nEnd of program")
    except KeyboardInterrupt:
        buzzer.run('0')
        print ("\nEnd of program")
```

Reference

buzzer.run(cmd)

This function has one input parameter. If the input is '1', the buzzer will be turned on. If the input is '0', the buzzer will be turned off.

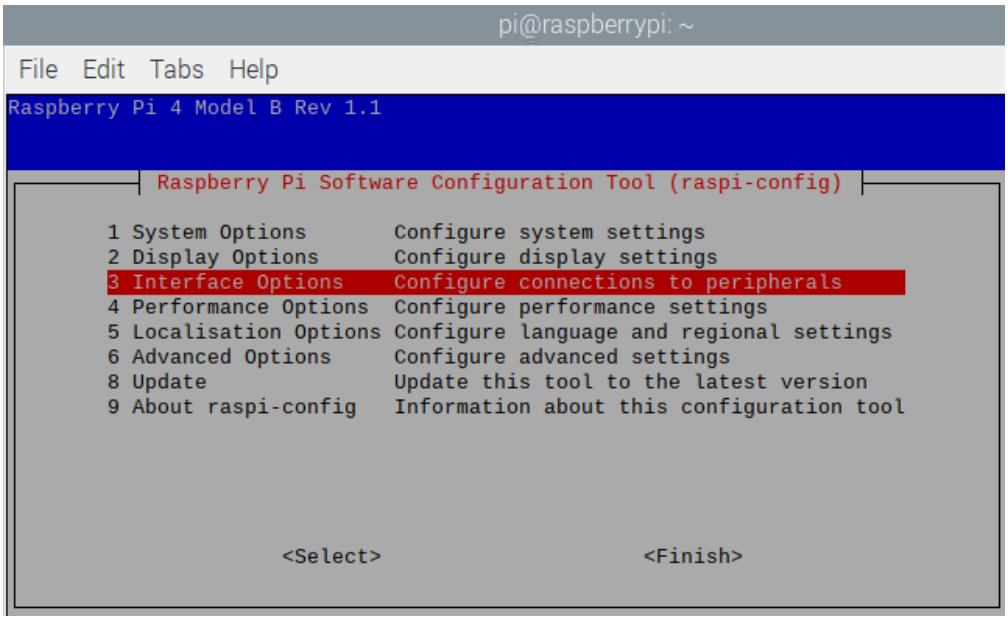
Camera

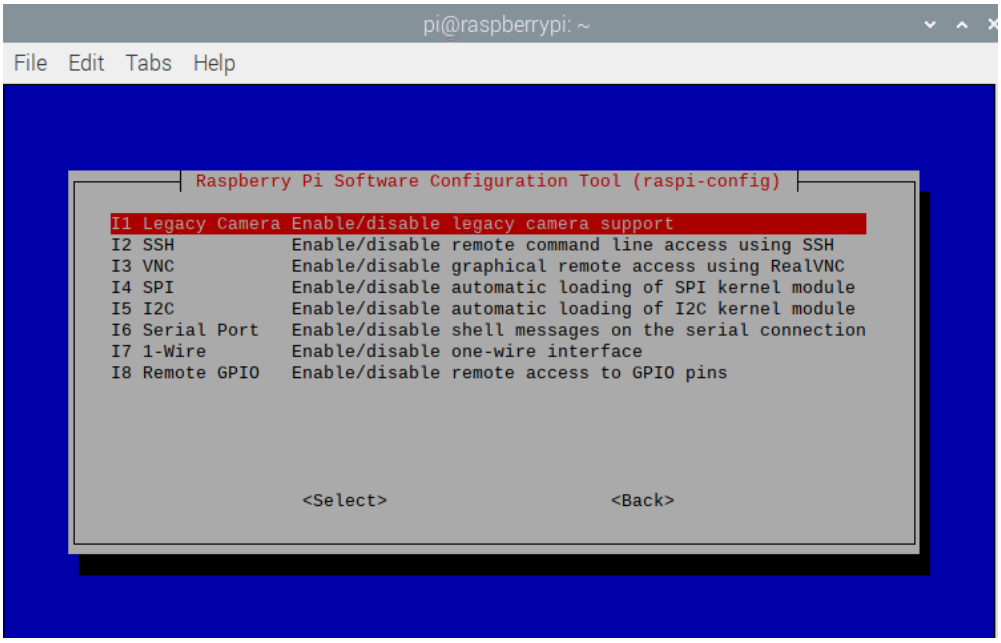
To use the camera, you need to disable legacy camera, which is disabled by default on the latest Raspberry Pi OS. If it is not disabled, please do it as below.

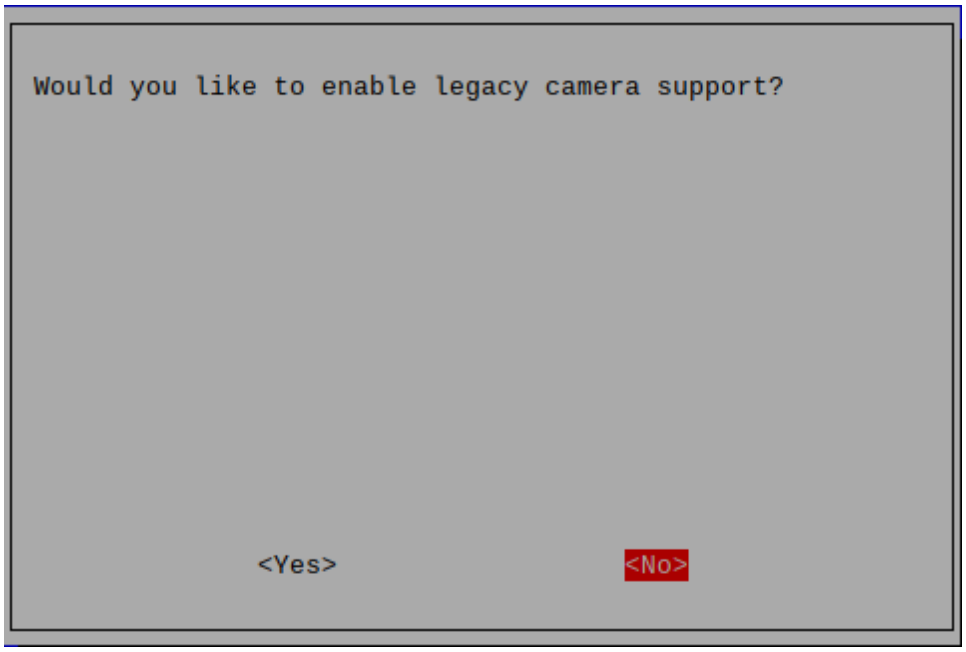
Enter the following command. Choose **Interface Options→ Legacy Camera →No→OK→Finish,**and then restart the Raspberry Pi.

```
sudo raspi-config
```

```
pi@raspberrypi:~ $ sudo raspi-config
```



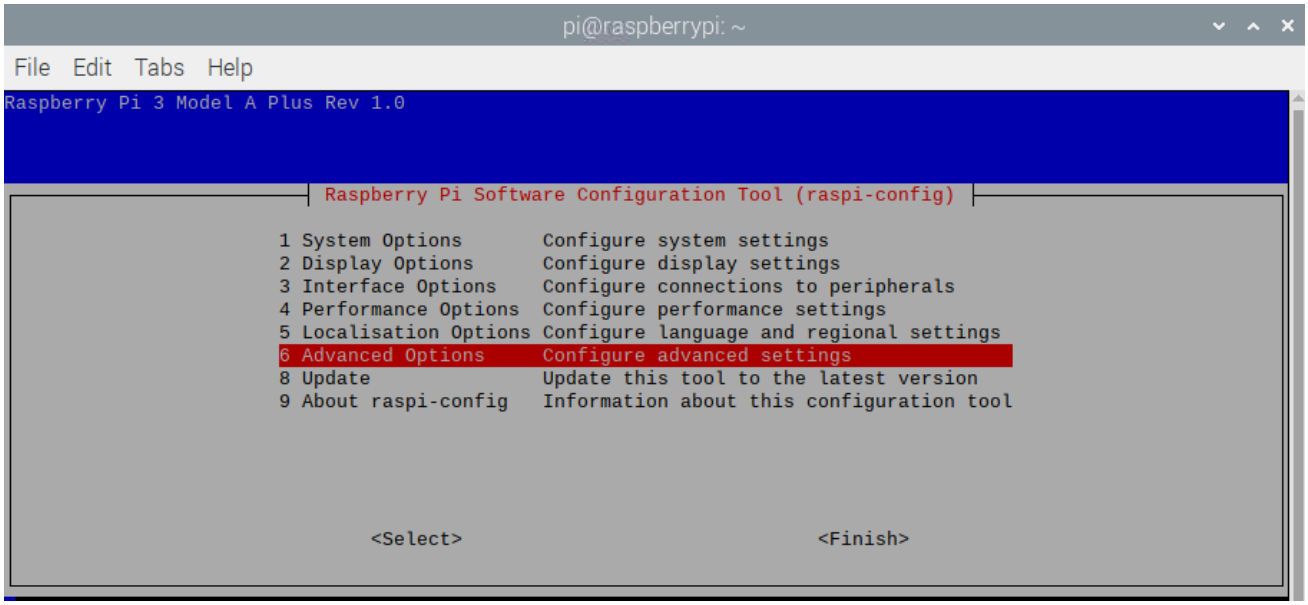


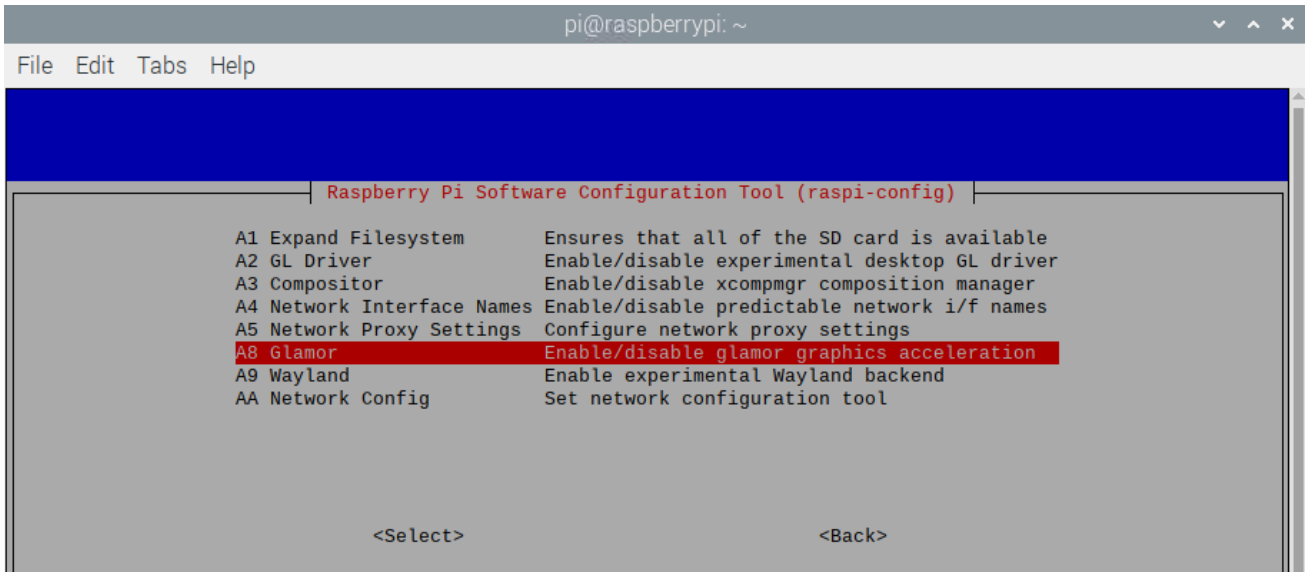


libcamera-apps does not work properly on Pi 0 to 3 devices when running the latest Bullseye images.
A workaround is to open a terminal, run "**sudo raspi-config**", navigate to "**Advanced Options**" and enable "**Glamor**" graphic acceleration. Then reboot your Pi.

```
sudo raspi-config
```

```
pi@raspberrypi:~ $ sudo raspi-config
```


pi@raspberrypi: ~
File Edit Tabs Help
Raspberry Pi 3 Model A Plus Rev 1.0
Raspberry Pi Software Configuration Tool (raspi-config)
1 System Options Configure system settings
2 Display Options Configure display settings
3 Interface Options Configure connections to peripherals
4 Performance Options Configure performance settings
5 Localisation Options Configure language and regional settings
6 Advanced Options Configure advanced settings
8 Update Update this tool to the latest version
9 About raspi-config Information about this configuration tool
<Select>
<Finish>


pi@raspberrypi: ~
File Edit Tabs Help
Raspberry Pi Software Configuration Tool (raspi-config)
A1 Expand Filesystem Ensures that all of the SD card is available
A2 GL Driver Enable/disable experimental desktop GL driver
A3 Compositor Enable/disable xcompmgr composition manager
A4 Network Interface Names Enable/disable predictable network i/f names
A5 Network Proxy Settings Configure network proxy settings
A8 Glamor Enable/disable glamor graphics acceleration
A9 Wayland Enable experimental Wayland backend
AA Network Config Set network configuration tool
<Select>
<Back>


pi@raspberrypi: ~
File Edit Tabs Help
Would you like glamor acceleration to be enabled?
<Yes>
<No>

Then reboot your Pi.

If the result above does not appear, please check the camera Wiring

(Note: plugging and unplugging the cable requires the Raspberry Pi to be powered off, otherwise the camera module may be burned.)

Run program

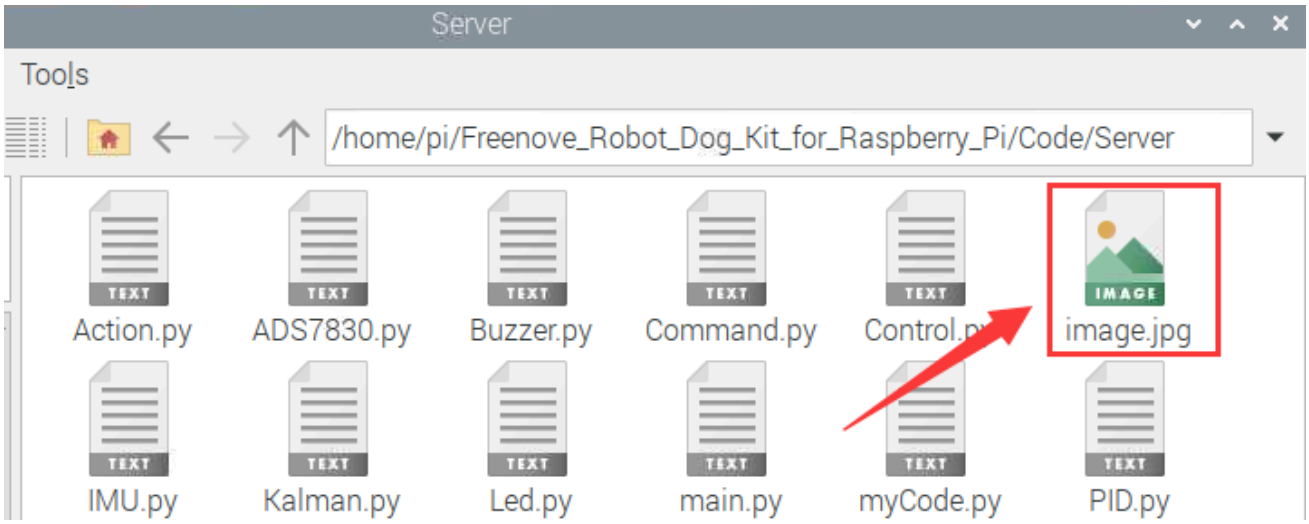
1. execute the cd command:

```
cd ~/Freenove_Robot_Dog_Kit_for_Raspberry_Pi/Code/Server
```

2. Execute command below:

```
python camera.py
```

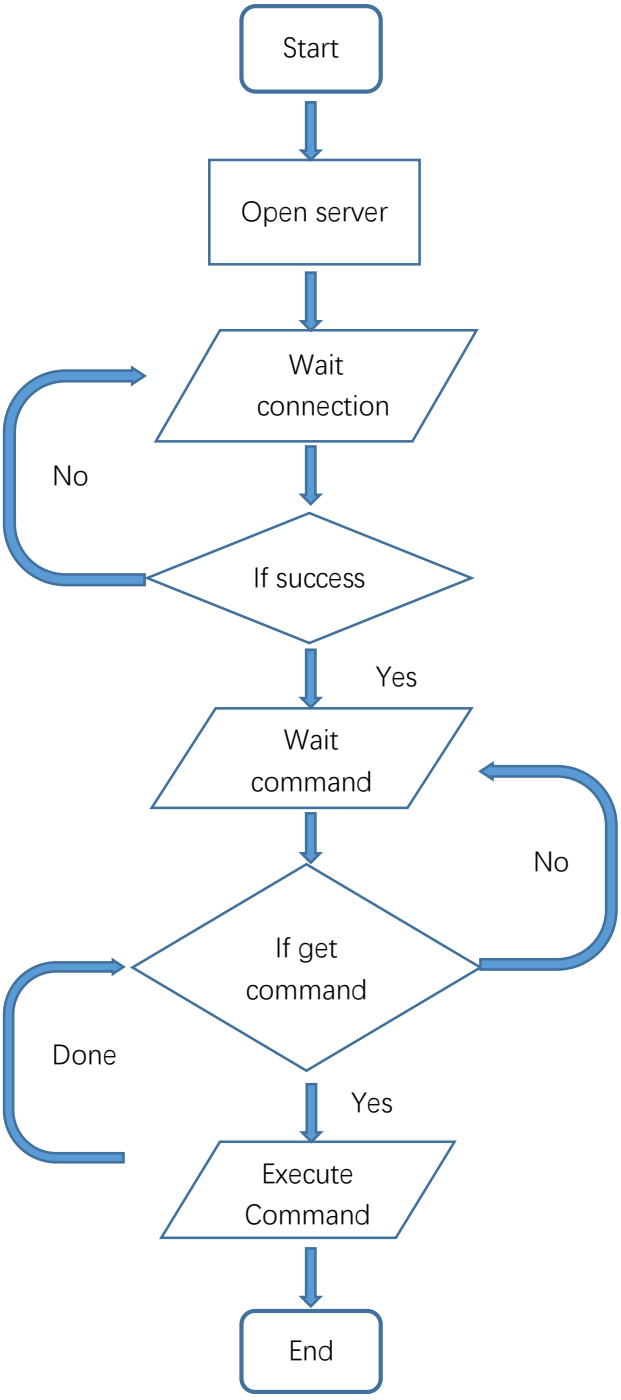
Then please open and check the generated image.jpg under /Freenove_Robot_Dog_Kit_for_Raspberry_Pi/Code/Server.

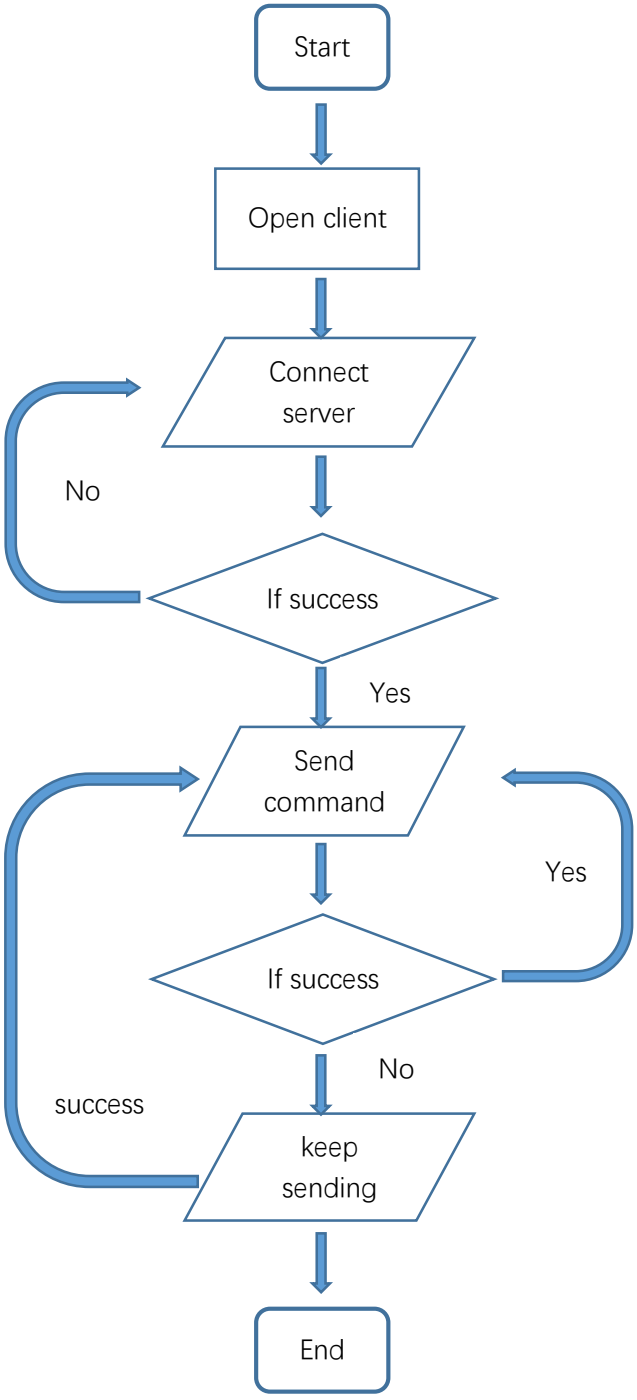


Chapter 4 Robot Dog

If you have any concerns, please feel free to contact us at support@freenove.com

This robot has rich functions, such as real-time video, LED, ultrasonic ranging. The server and client are established, based on Python3 and PyQt5. They communicate via TCP/IP protocol. The robot can be controlled remotely within a local area network (LAN).





Server

The server runs on the Raspberry Pi. It sends the camera data and ultrasonic module data to the client, and receive commands from client.

Part of server code is as below:

```
    def get_interface_ip(self):
        s = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)
        return socket.inet_ntoa(fcntl.ioctl(s.fileno(),
                                            0x8915,
                                            struct.pack('256s',b'wlan0'[:15])
                                            )[20:24])
    def turn_on_server(self):
        #ip adress
        HOST=self.get_interface_ip()
        #Port 8000 for video transmission
        self.server_socket = socket.socket()
        self.server_socket.setsockopt(socket.SOL_SOCKET,socket.SO_REUSEPORT,1)
        self.server_socket.bind((HOST, 8000))
        self.server_socket.listen(1)

        #Port 5000 is used for instruction sending and receiving
        self.server_socket1 = socket.socket()
        self.server_socket1.setsockopt(socket.SOL_SOCKET,socket.SO_REUSEPORT,1)
        self.server_socket1.bind((HOST, 5000))
        self.server_socket1.listen(1)

        print('Server address: '+HOST)

    def turn_off_server(self):
        try:
            self.connection.close()
            self.connection1.close()
        except :
            print ('\n'+"No client connection")
```

Reference

For more code details, please open “Server.py” in Server folder to check.

get_interface_ip()
This function can get the IP address of WLAN0 of the Raspberry Pi.
turn_on_server()
This function is used to open TCP and wait for a client connection. The channel of port 5000 is mainly used to send and receive commands between the client and server. The channel of port 8000 is used to send video data to the client.
turn_off_server()
This function is used to close TCP.
send_data()
This function is used to send commands to the client.
reset_serve()
This function is used to close TCP and restart.
transmission_video()
This function is used to transmit video data to the client.
receive_instruction()
This function is used to receive commands from the client

Open Server

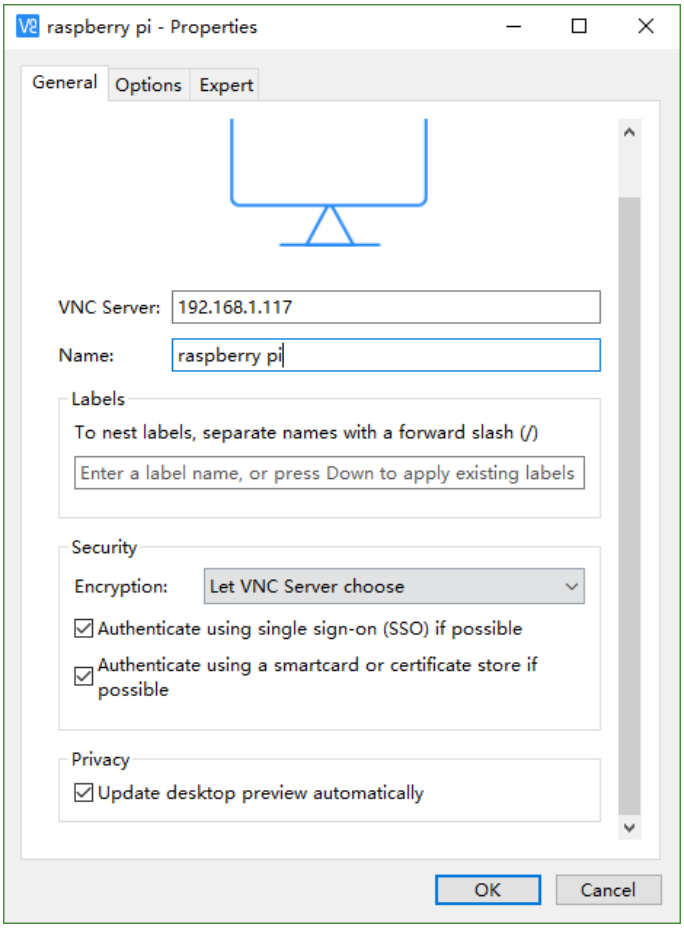
Step 1 Login Raspberry Pi via VNC viewer

Because server and client use GUI. You need use VNC viewer as remote desktop way.

Download and install VNC Viewer according to your computer system by clicking following link:

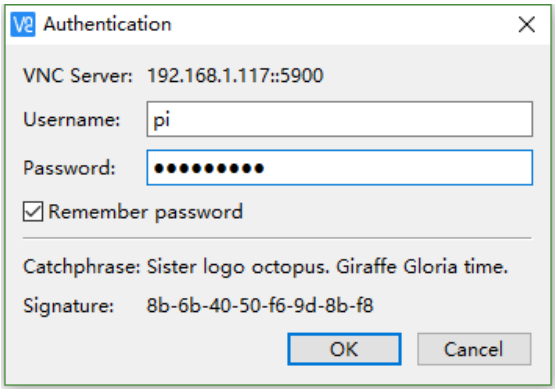
https://www.realvnc.com/en/connect/download/viewer/

After installation is completed, open VNC Viewer. And click File → New Connection. Then the interface is shown below.



Enter IP address of your Raspberry Pi and fill in a Name. And click OK.

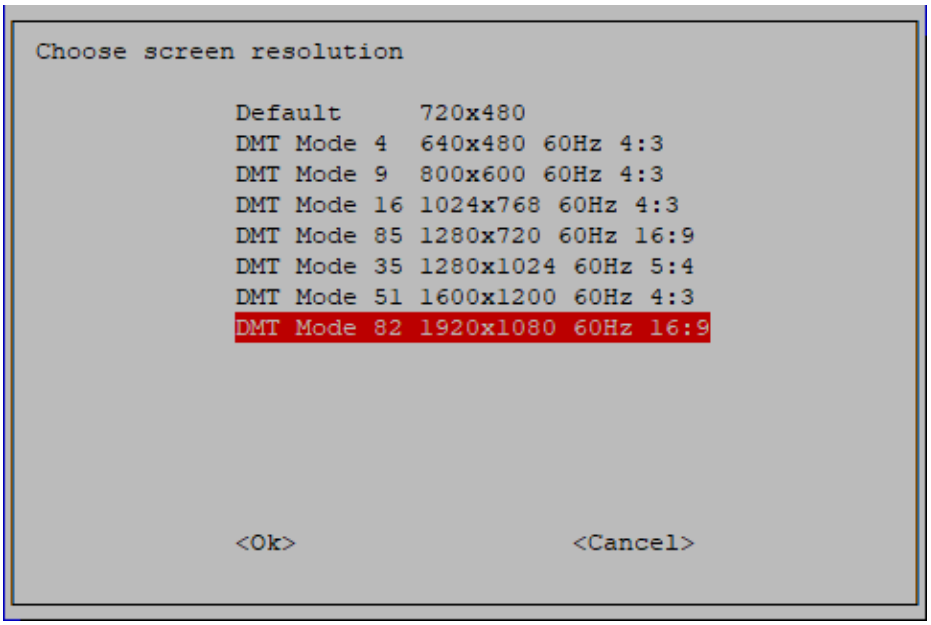
Then on the VNC Viewer panel, double-click new connection you just created, and the following dialog box pops up. Enter username: **pi** and Password: **raspberry**. And click OK.



If the resolution ratio is not great or there is just a **little window**, you can set a proper resolution ratio via steps below.

```
sudo raspi-config
```

Select 2 Display Options →D5 VNC Resolution→proper resolution ratio (set by yourself)→OK→Finish. And then reboot Raspberry Pi.



In addition, your VNC Viewer window may zoom your Raspberry Pi desktop. You can change it. On your VNC View control panel, click right key. And select Properties->Options label->Scaling. Then set proper scaling.

Step 2 Run commands

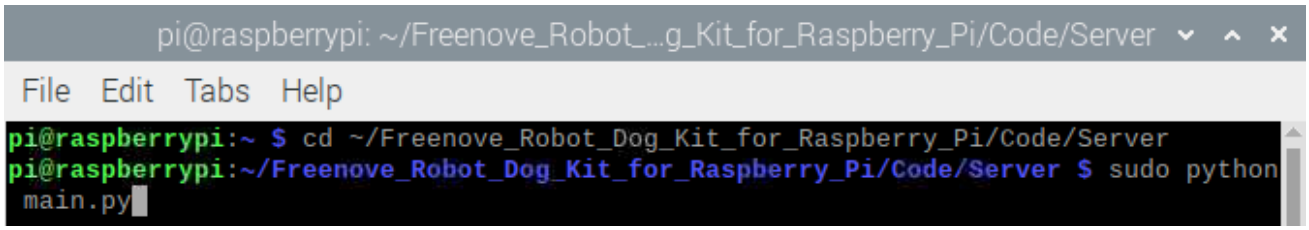
Enter following command in the terminal.

1. Use cd command to enter directory where main.py is located:

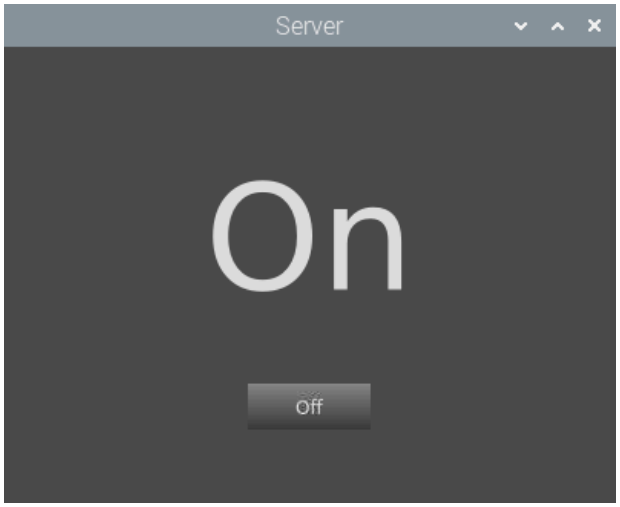
```
cd ~/Freenove_Robot_Dog_Kit_for_Raspberry_Pi/Code/Server
```

2. Run main.py:

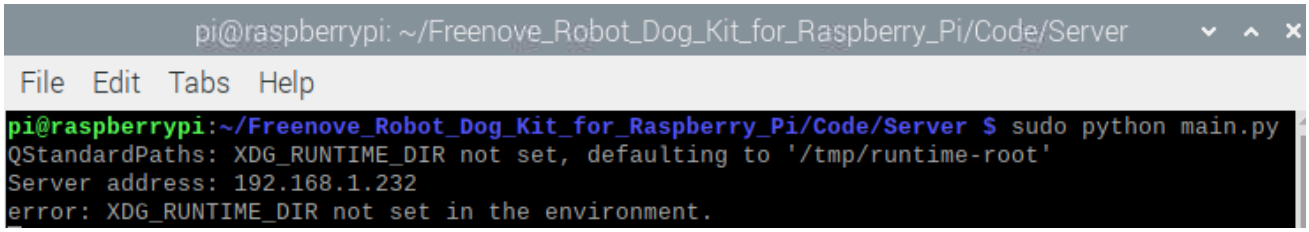
```
sudo python main.py
```



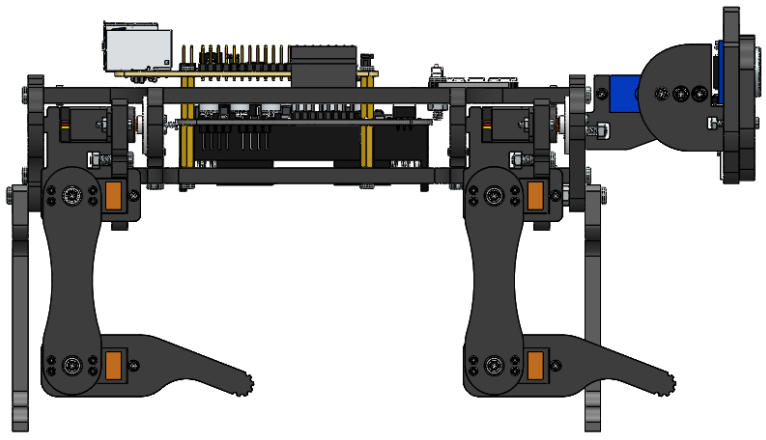
The interface is as below:



You will see your server address (Raspberry Pi IP), Which is the LAN IP address of your Raspberry Pi. In the next chapter, you will need to enter this IP address to connect the Raspberry Pi client with the server.



The robot will start from this posture. We define it as **Relax mode**. It will also be referenced later.



If you don't like the interface, you can also enter the commands to open the server. It is more convenient.

1. Use cd command to enter directory where main.py is located:

```
cd ~/Freenove_Robot_Dog_Kit_for_Raspberry_Pi/Code/Server
```

2. Run main.py:

```
sudo python main.py -t -n
```

or Run main,py with following command:

```
sudo python main.py -tn
```

"-t" means open TCP communication. "-n" means don't show interface.

Sever Auto Start

1 Open the terminal and execute the following two commands respectively to create a "start.sh" file.

```
cd ~
sudo touch start.sh
```

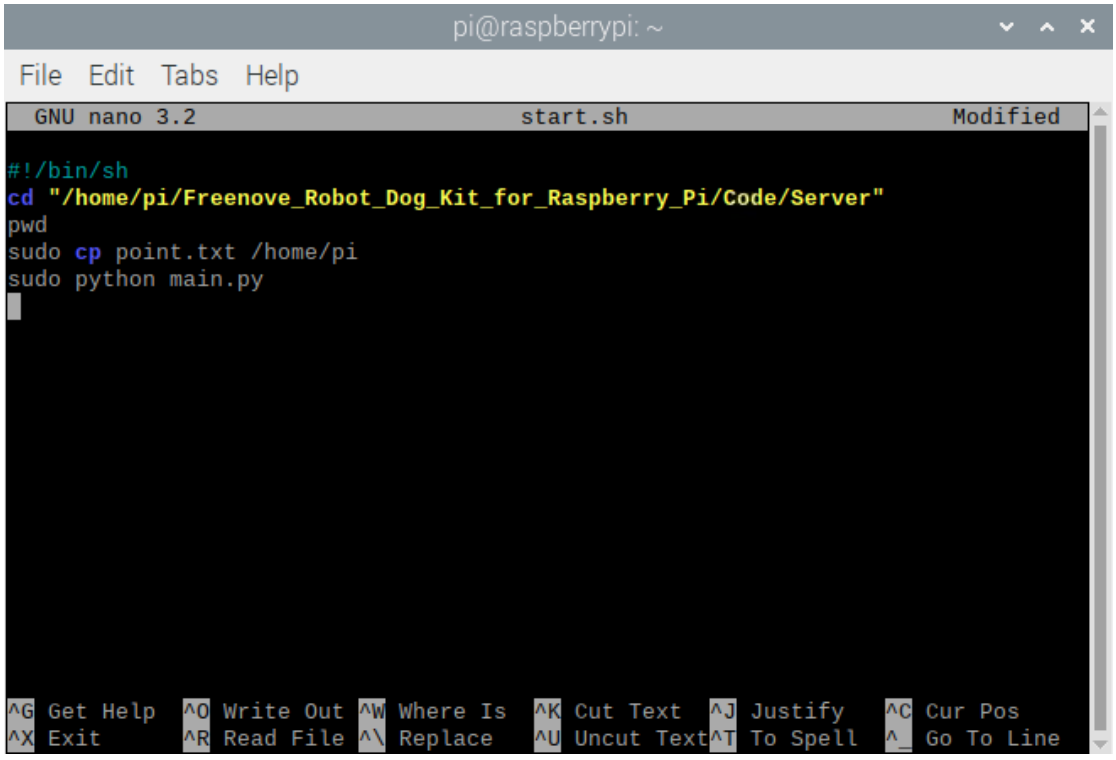
2 Open "start.sh".

```
sudo nano start.sh
```

3 Add the following contents to "start.sh" file.

```
#!/bin/sh
cd "/home/pi/Freenove_Robot_Dog_Kit_for_Raspberry_Pi/Code/Server"
pwd
sleep 10
sudo cp point.txt /home/pi
sudo python main.py
```

Press Ctrl + O and then press Enter to save it. Press Ctrl+X to exit.



4 Modify permissions.

```
sudo chmod 777 start.sh
```

5 Enter the following command to create a directory.

```
mkdir ~/.config/autostart/
```

6 create and open “start.desktop” file

```
sudo nano .config/autostart/start.desktop
```

7 Add the following content to “start.desktop” file.

```
[Desktop Entry]
Type=Application
Name=start
NoDisplay=true
Exec=/home/pi/start.sh
```

Press Ctrl + O and then press Enter to save it. Press Ctrl+X to exit.

8 Modify permissions.

```
sudo chmod +x .config/autostart/start.desktop
```

9 Finally enter the following content to reboot Raspberry Pi.

```
sudo reboot
```

Note: To cancel auto start, please delete the files “start.sh” and “start.desktop” created above.

Client

The client can receive video data and commands from the server, and can send commands to the server. And it can run on different systems, such as windows, macOS and so on. However, you need to install related software and libraries when running it.

Part of client code is as below:

```
    def turn_on_client(self,ip):
        self.client_socket1 = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        self.client_socket = socket.socket(socket.AF_INET, socket.SOCK_STREAM)
        print (ip)
    def turn_off_client(self):
        try:
            self.client_socket.shutdown(2)
            self.client_socket1.shutdown(2)
            self.client_socket.close()
            self.client_socket1.close()
        except Exception as e:
            print(e)
    def receiving_video(self,ip):
        stream_bytes = b' '
        try:
            self.client_socket.connect((ip, 8000))
            self.connection = self.client_socket.makefile('rb')
        except:
            #print ("command port connect failed")
            pass
        while True:
            try:
                stream_bytes= self.connection.read(4)
                leng=struct.unpack('L', stream_bytes[:4])
                jpg=self.connection.read(leng[0])
                if self.is_valid_image_4_bytes(jpg):
                    if self.video_flag:
                        self.image = cv2.imdecode(np.frombuffer(jpg, dtype=np.uint8),
cv2.IMREAD_COLOR)
                        if self.ball_flag:
                            self.Looking_for_the_ball()
                        self.video_flag=False
            except BaseException as e:
                print (e)
                break
```

Reference

For more code details, please open “Client.py” in Client folder to check.

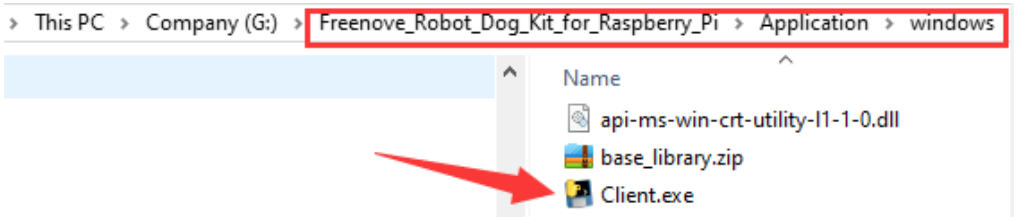
turn_on_client ()
This function is used to connect client to the server.
turn_off_client ()
This function is used to disconnect from the server.
receiving_video()
This function is used to receive video data sent from the server.
is_valid_image_4_bytes ()
This function is used to check whether each frame of video data is complete.
Looking_for_the_ball ()
This function is used to find the small ball in the video image.
send_data ()
This function is used to send commands to the server.
receive_data ()
This function is used to receive commands from the server.

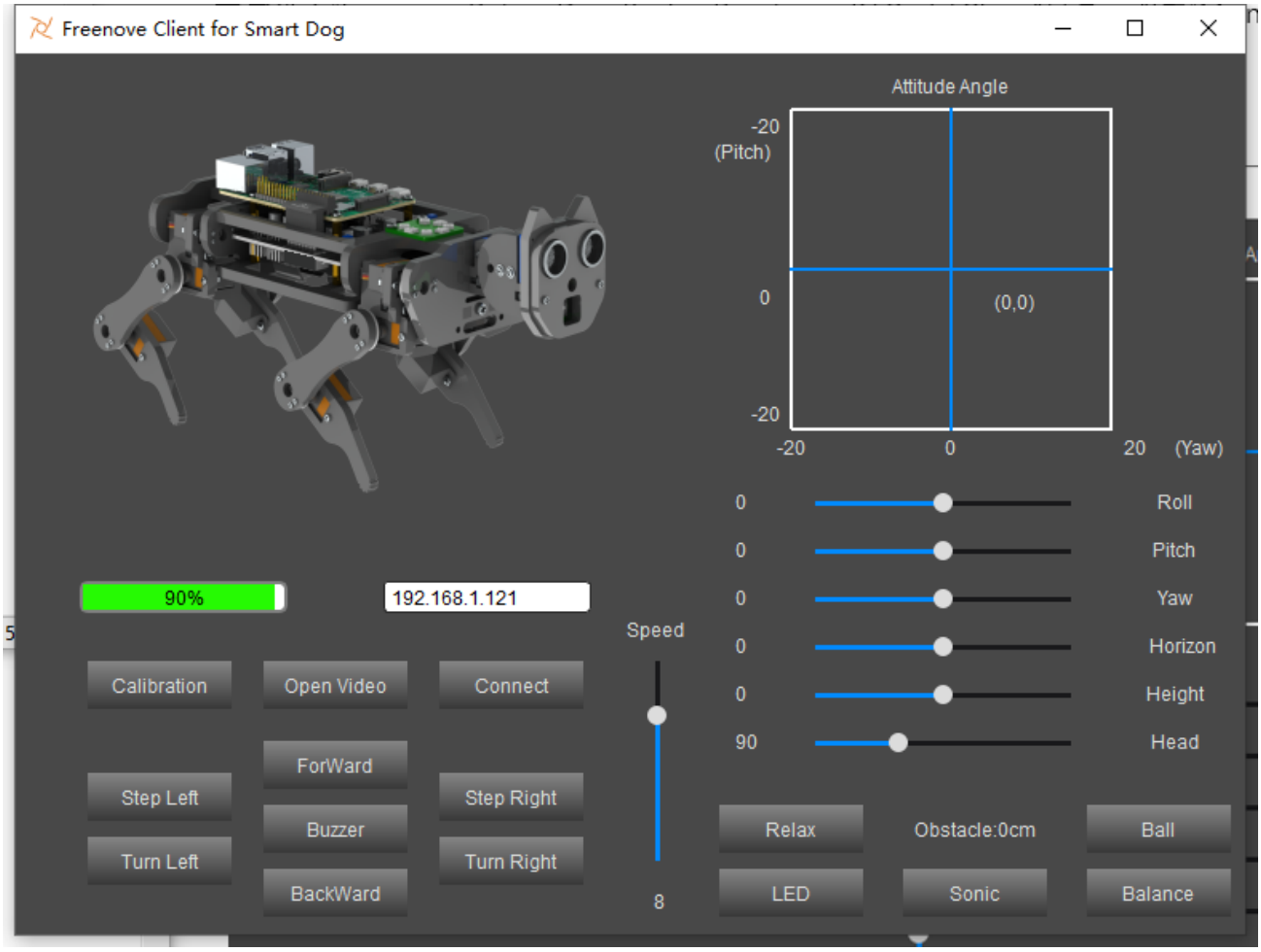
Run Client on Windows system

There are two ways to run Client on Windows.

Option 1 Running executable file directly

Find the "Client.exe" file in the specified directory, double click it and the Client is opened.





After the client is successfully opened,

1.You need to open the Raspberry Pi and Turn on the server,

2.Enter the Raspberry Pi's IP address in the white IP edit box,

3.Click "**Connect**" to connect client to the Raspberry Pi.

After the connection is successful, you need to calibrate the robot in Calibration section. After the calibration is completed, the robot dog can be controlled to move.

You can refer to this video: https://youtu.be/l2v9PdwQdvY

Note: when Raspberry Pi is shut down, server will be closed. You need open server again the next time.

Option 2 Install python3 and some related python libraries to run client

If you want to modify the client, you can follow this section.

This section will be completed in your **computer with windows system, not Raspberry Pi.**

To run the client on a Windows system, you need to install some software and libraries, which takes some time. During this process, it does not need to run a server and a Raspberry Pi. You can turn off the Raspberry Pi first. After the installation is completed, turn on the Raspberry Pi and the server.

Install python3

Download the installation file via the below link:

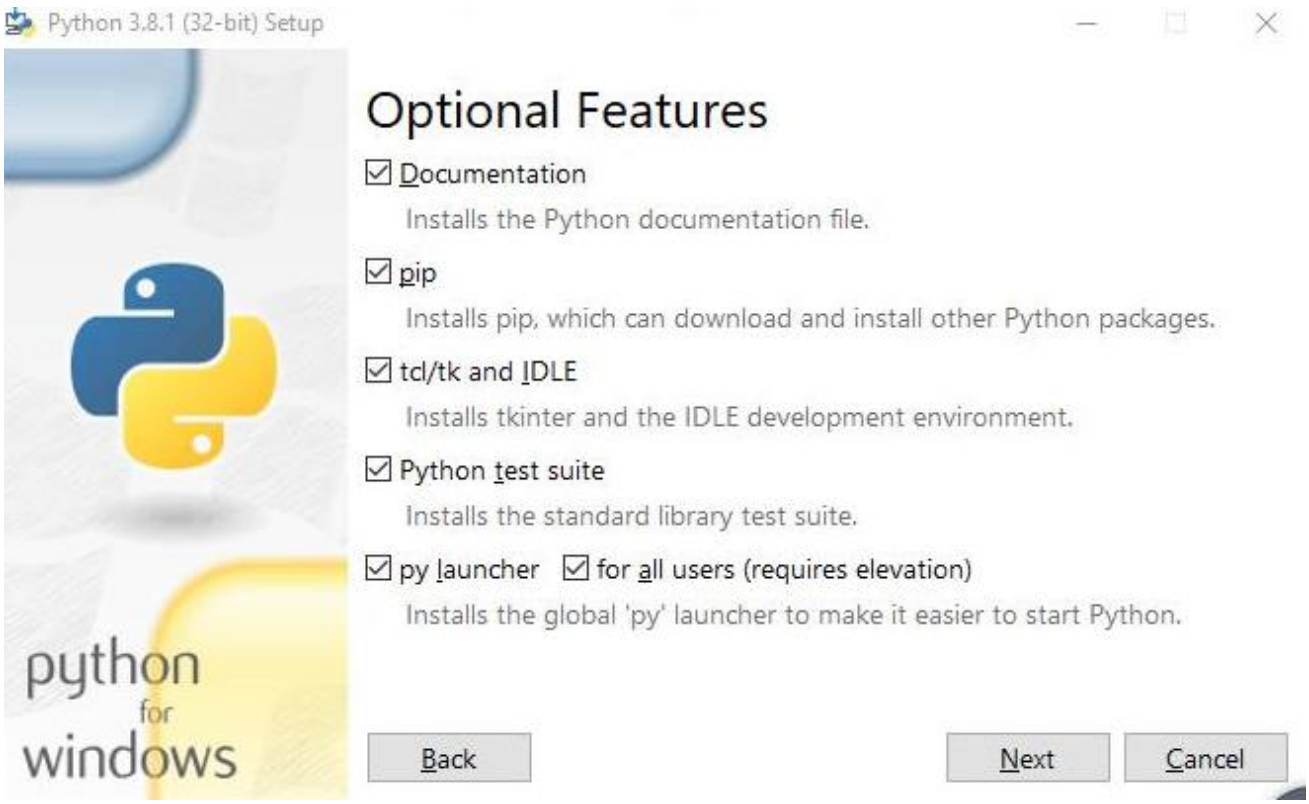
https://www.python.org/downloads/windows/

Click **Latest Python 3 Release** - Python 3.8.1 (as an example).

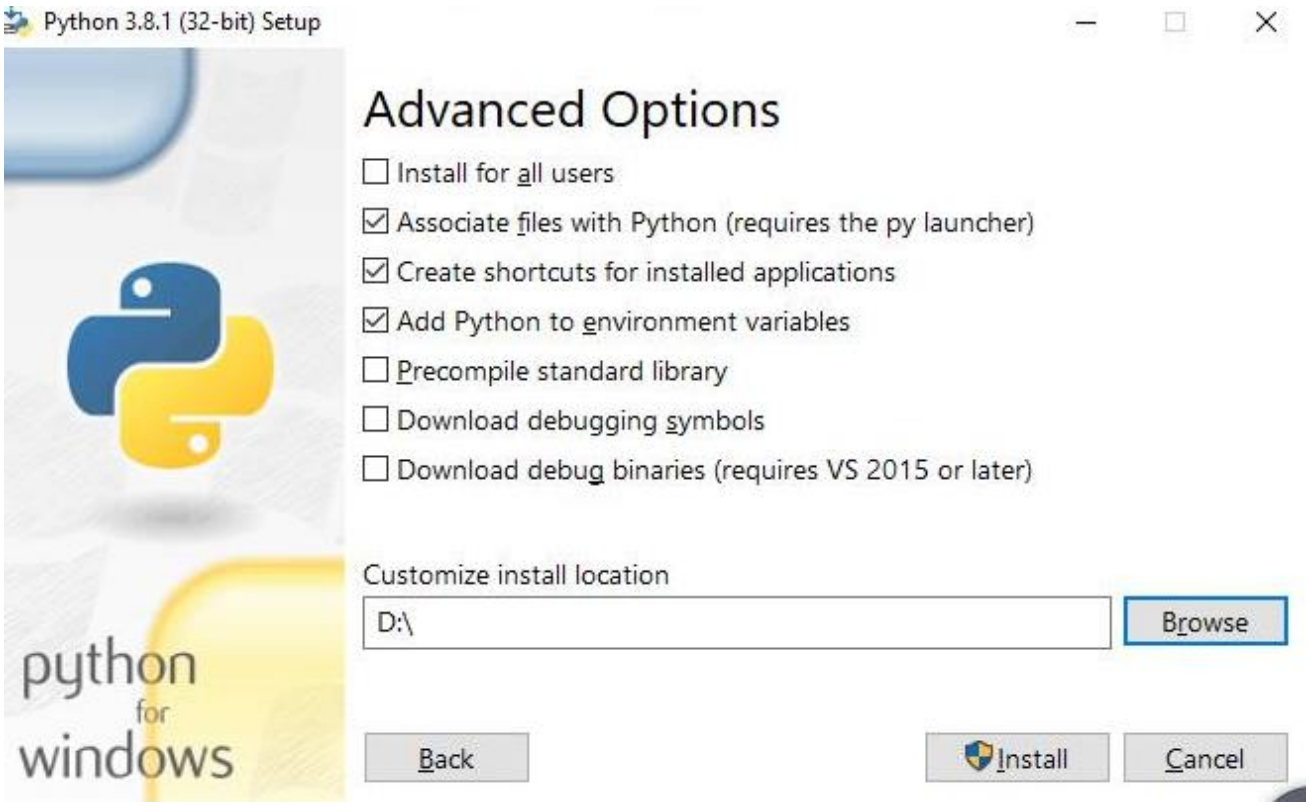
Version	Operating System	Description
Gzipped source tarball	Source release	
XZ compressed source tarball	Source release	
macOS 64-bit installer	Mac OS X	for OS X 10.9 and later
Windows help file	Windows	
Windows x86-64 embeddable zip file	Windows	for AMD64/EM64T/x64
Windows x86-64 executable installer	Windows	for AMD64/EM64T/x64
Windows x86-64 web-based installer	Windows	for AMD64/EM64T/x64
Windows x86 embeddable zip file	Windows	
Windows x86 executable installer	Windows	
Windows x86 web-based installer	Windows	

Choose “Windows x86-64 executable installer” to download and install.

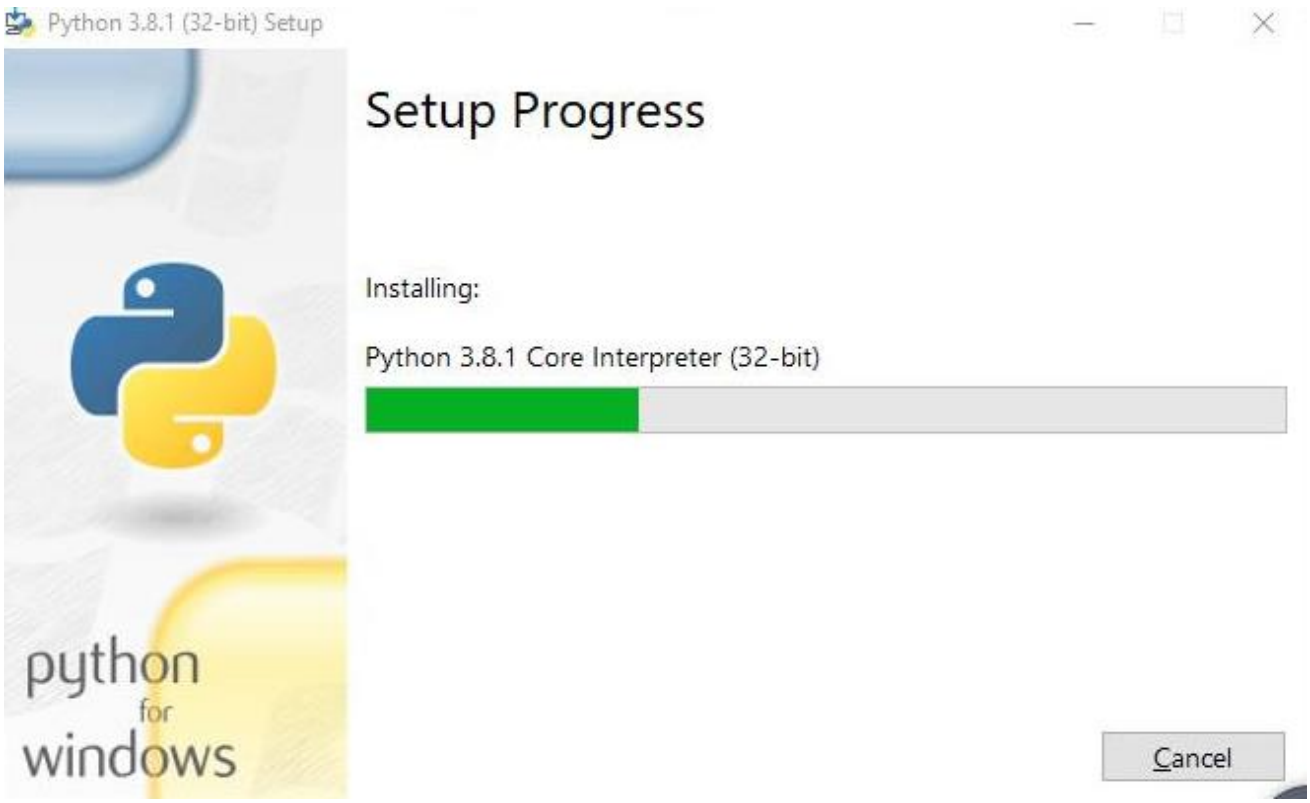
Select "Add Python 3.8 to PATH". And choose Customize installation.



Select all options and click Next.



Here python is installed into D disk as an example (You can choose your own installation path). Click Install.



Wait for installation.

Installation is successful.

Install PyQt5, opencv, numpy and other libraries

If you have not download the zip file, download it via below link:

https://github.com/Freenove/Freenove_Robot_Dog_Kit_for_Raspberry_Pi/archive/master.zip

Then unzip it and delete "-master" to rename it to "Freenove_Robot_Dog_Kit_for_Raspberry_Pi".

Then put it into D disk, for example.

You can also place it into other disks (like E), but the path in following command should be modified accordingly (replace D: by E:).

Press "win + R" and enter cmd, and click ok. Then enter following commands.

1.Enter D disk. If you put it into E, it should be E:

```
D:
```

2.Enter directory where setup_windows.py is located:

```
cd D:\Freenove_Robot_Dog_Kit_for_Raspberry_Pi\Code
```

3.Run: setup_windows.py

```
Python setup_windows.py
```

```
C:\Users\Freenove>D:

D:\>cd D:\Freenove_Robot_Dog_Kit_for_Raspberry_Pi\Code

D:\Freenove_Robot_Dog_Kit_for_Raspberry_Pi\Code>Python setup_windows.py
```

Or double-click "**setup_windows.py**" to execute the installation program, under following path:

Freenove_Robot_Dog_Kit_for_Raspberry_Pi\Code

Or use python3 to run "setup_windows.py".

Note: If the default python in your windows system is not python3 but python2, then change all "python" in "setup_windows.py" to "python3" and run the command "setup_windows.py" as shown below.

```
import os
import sys
import time
flag=0x00
for x in range(1,4):
    if os.system("python -m pip install --upgrade pip") == 0:
        flag=flag | 0x01
        break
for x in range(1,4):
    if os.system("pip3 install PyQt5") == 0:
        flag=flag | 0x02
```

```
Python3 setup_windows.py
```

Installation takes some time, please be patient. If all installations are successful, it will prompt “Press any key to continue…” and "All libraries installed successfully".

```
Package       Version
------------- -------------
click         7.1.1
numpy         1.18.2
opencv-python 4.2.0.32
Pillow        7.0.0
pip           20.0.2
PyQt5         5.13.2
PyQt5-sip     12.7.1
pyqt5-tools   5.13.2.1.6rc1
python-dotenv 0.12.0
setuptools    41.2.0
Press any key to continue . . .

All libraries installed successfully
```

If some libraries are not installed successfully, it will prompt "Some libraries have not been installed yet. Please run '**python setup_windows.py**' again". Then you need to execute the python3 setup_windows.py command again. Most installation failures are caused by a poor network. You can check the network before installing.

Open client

Press “win + R” and enter cmd, and click ok. Then enter following commands.

1.Enter D disk. If you put it into E, it should be E:

```
D:
```

2.Enter directory where Main.py is located:

```
cd D:\Freenove_Robot_Dog_Kit_for_Raspberry_Pi\Code\Client
```

3.Run Main.py:

```
Python Main.py
```

```
C:\Users\Freenove>D:

D:\>cd D:\Freenove_Robot_Dog_Kit_for_Raspberry_Pi\Code\Client

D:\Freenove_Robot_Dog_Kit_for_Raspberry_Pi\Code\Client>Python Main.py
```

Or double-click "Main.py". under following path:

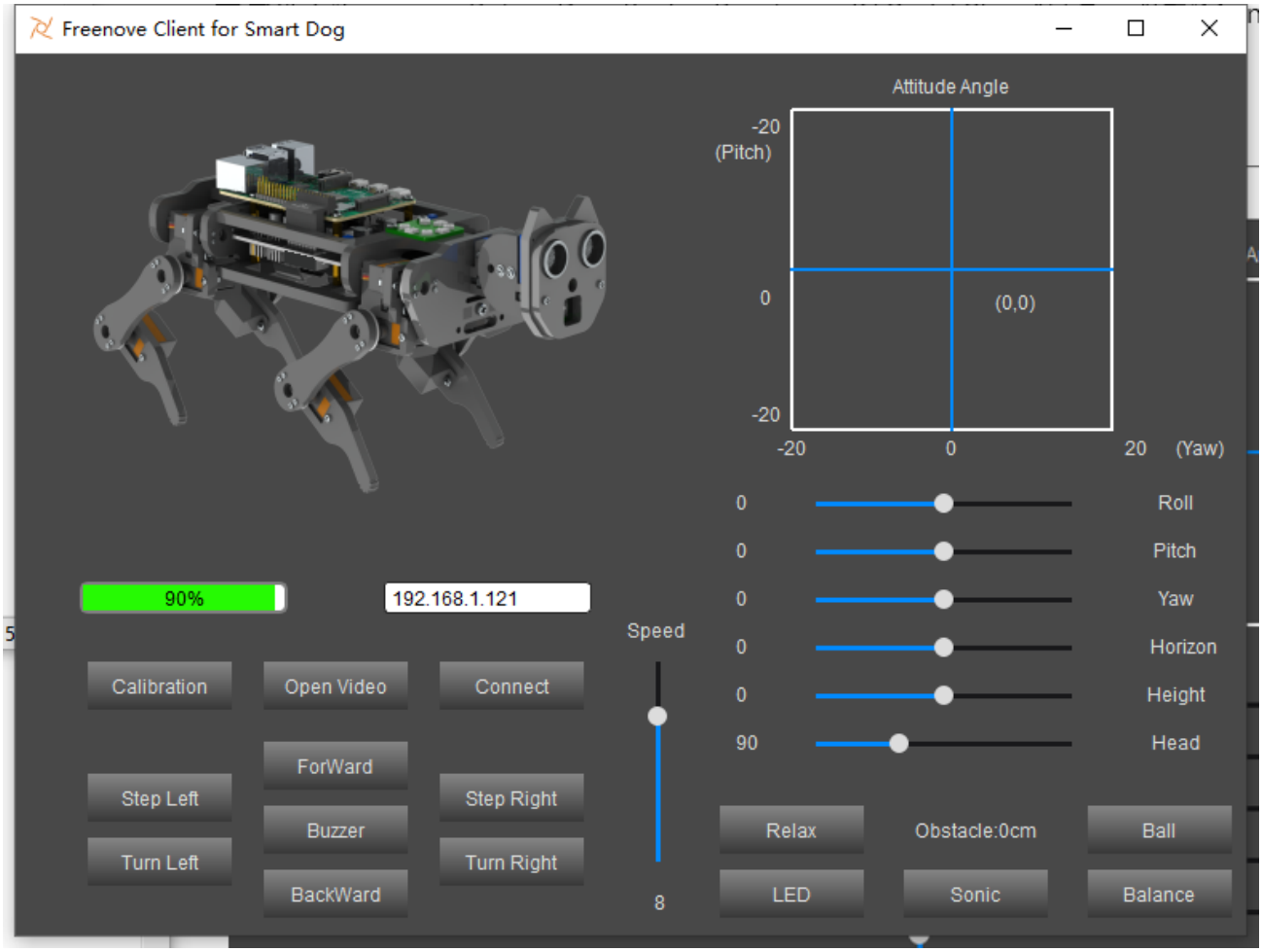
Freenove_Robot_Dog_Kit_for_Raspberry_Pi\Code\Client

Or use python3 to run "Main.py".

Note: If the default python in your windows system is not python3, then change the command to run Main.py as shown below.

```
Python3 Main.py
```

The client interface is shown below:



After the client is successfully opened,

1.You need to open the Raspberry Pi and Turn on the server,

2.Enter the Raspberry Pi's IP address in the white IP edit box,

3.Click "**Connect**" to connect client to the Raspberry Pi.

After the connection is successful, you need to calibrate the robot in Calibration section. After the calibration is completed, the robot dog can be controlled to move.

Note: when Raspberry Pi is shut down, server will be closed. You need open server again the next time.

Run Client on macOS system

Here take MacOS 10.13 as an example. To run the client on MacOS, you need to install some software and libraries. During the installation, it does not need to run a server and use a Raspberry Pi. You can turn off the Raspberry Pi first. After the installation is completed, turn on the Raspberry Pi and run the server. MacOS 10.13 comes with python2, but no python3. The programs of this robot can only run on python3.

Install python3

Download installer, link https://www.python.org/downloads/

Python 3.8.1	Dec. 18, 2019	Download
Python 3.7.6	Dec. 18, 2019	Download

Click Python 3.8.2.

If your macOS is 11. Like 11.0, please install python 3.9.

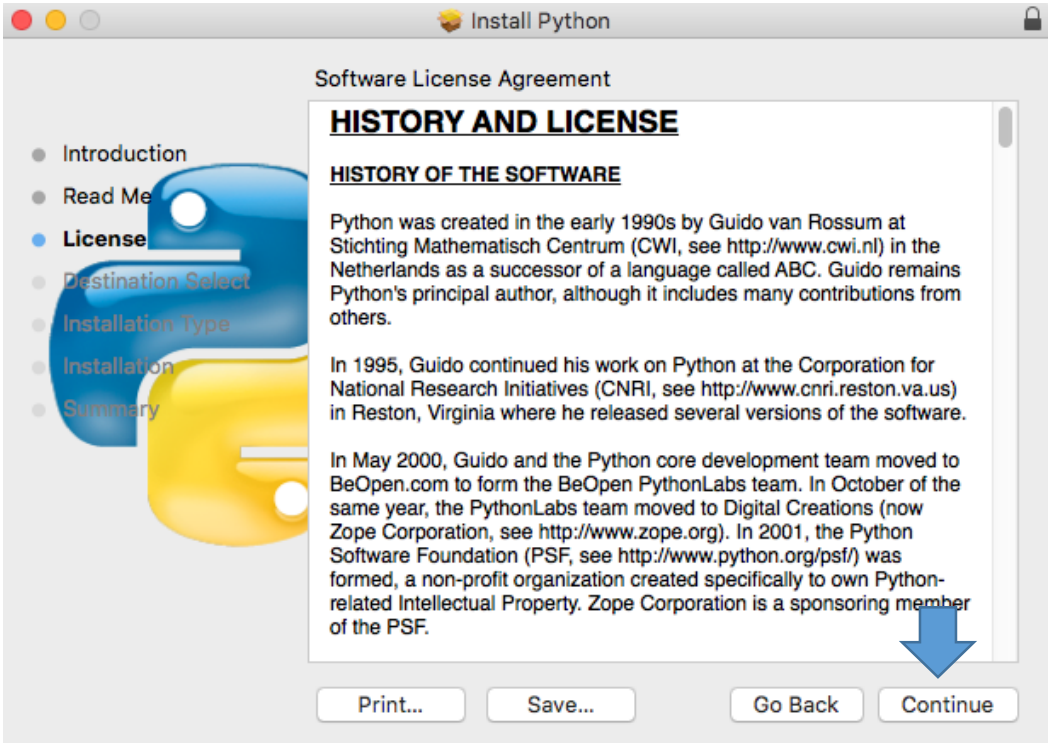
If your macOS is NOT 11, like 10.15, please install python 3.8. If you have installed python 3.9. You need uninstall it first.

Version	Operating System	Description
Gzipped source tarball	Source release	
XZ compressed source tarball	Source release	
macOS 64-bit installer	Mac OS X	for OS X 10.9 and later
Windows help file	Windows	
Windows x86-64 embeddable zip file	Windows	for AMD64/EM64T/x64
Windows x86-64 executable installer	Windows	for AMD64/EM64T/x64
Windows x86-64 web-based installer	Windows	for AMD64/EM64T/x64
Windows x86 embeddable zip file	Windows	
Windows x86 executable installer	Windows	
Windows x86 web-based installer	Windows	

On the bottom of the page, click macOS 64-bit installer to download installer.

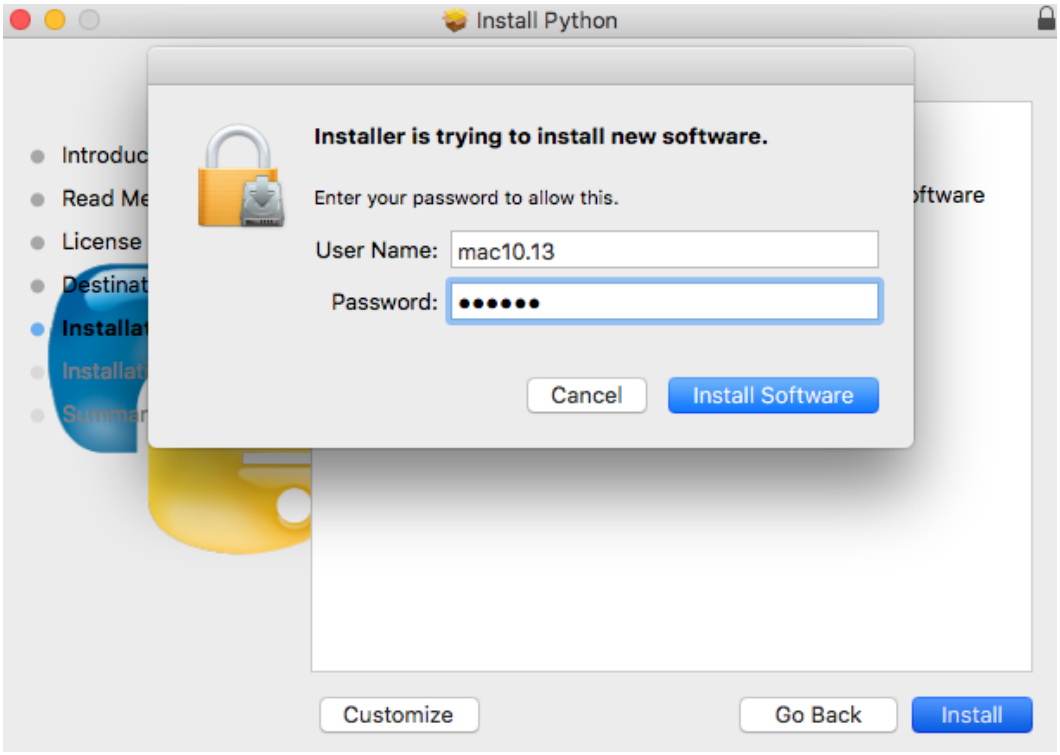
Then install python.

Click Continue.



Click Continue.

Click Agree.



Click Install. If your computer has a password, enter the password and click Install Software.

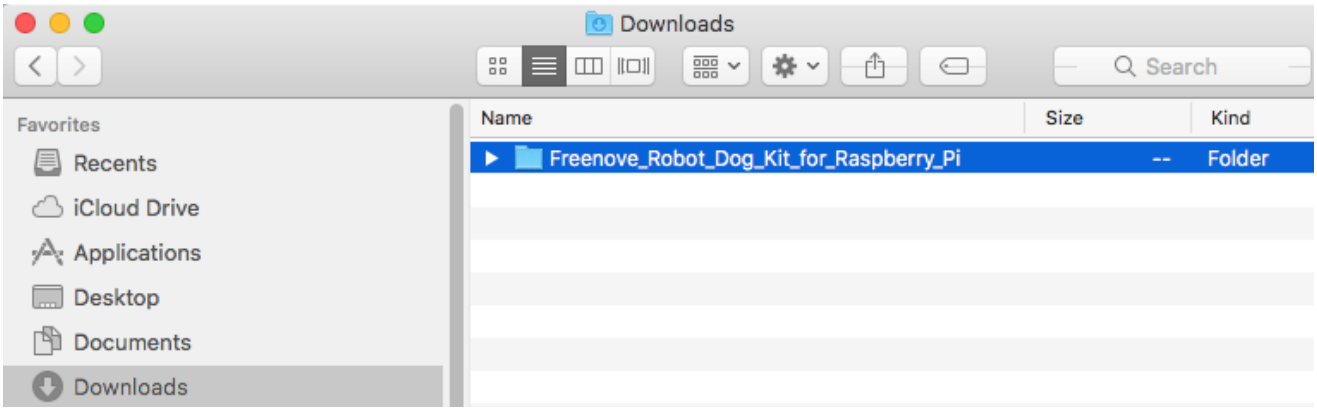
Now the installation is completed successfully.

You can find in Applications.

Install PyQt5、opencv、numpy and other libraries

If there is no robot dog code in your macOS, you can download it via link below:
https://github.com/Freenove/Freenove_Robot_Dog_Kit_for_Raspberry_Pi/archive/master.zip

After downloading successfully, you can find it under "Downloads".



Open "Terminal".

Type following command in Terminal.

1. Enter “Downloads” where the code is located. If your path is different, please modify the command.

```
cd Downloads
```

2. Enter directory where setup_macos.py is located:

```
cd Freenove_Robot_Dog_Kit_for_Raspberry_Pi/Code/
```

3. Run setup_macos.py:

```
python3 setup_macos.py
```

Installation takes some time, please wait patiently.

```
Package                 Version
----------------------- --------
numpy                   1.18.1
opencv-python-headless  4.2.0.32
Pillow                  7.0.0
pip                     20.0.2
PyQt5                   5.14.1
PyQt5-sip               12.7.1
setuptools              41.2.0

All libraries installed successfully
mac13deMac:Code mac10.13$
```

If some libraries are not installed successfully, it will prompt "Some libraries have not been installed yet. Please run 'python3 setup_windows.py' again". Then you need to execute the python3 setup_windows.py command again. Most installation failures are caused by a poor network. You can check the network before installing.

If you are using **macOS under 11.0, like 10.15.** **Just skip to “Open client”.**

If you are using **macOS 11.0 or later version**. Please run commands below:

```
pip3 uninstall PyQt5
pip3 install PyQt5
```

Open client

After installation is completed in previous step, now it is in the directory that setup_macos.py is located.

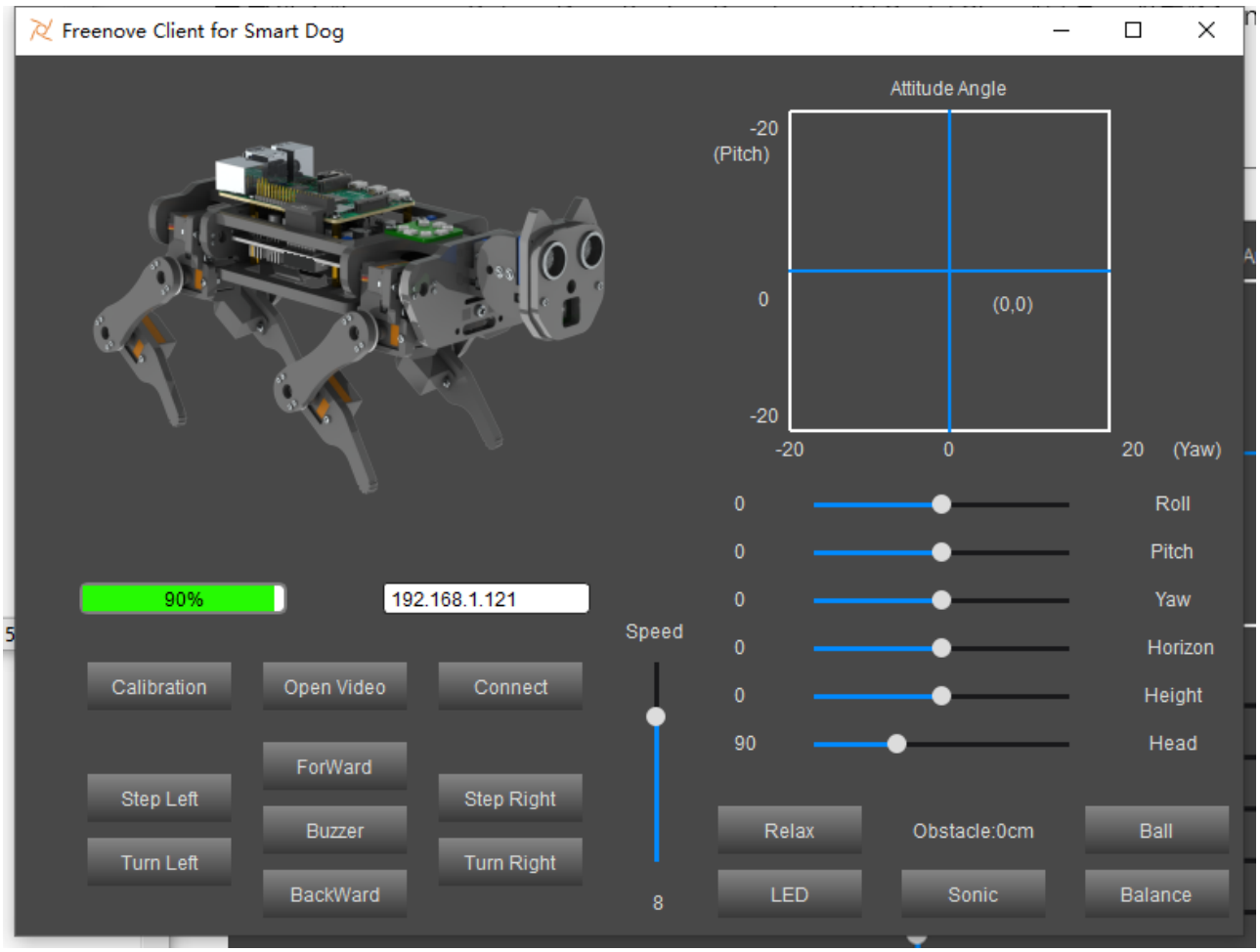
```
Package                 Version
----------------------- --------
numpy                   1.18.1
opencv-python-headless  4.2.0.32
Pillow                  7.0.0
pip                     20.0.2
PyQt5                   5.14.1
PyQt5-sip               12.7.1
setuptools              41.2.0

All libraries installed successfully
mac13deMac:Code mac10.13$
```

1. Type following command to enter where the program is located.

```
cd Client/
```

2. Type following command to run the program.

```
python3 Main.py
```



After the client is successfully opened,
1.You need to open the Raspberry Pi and Turn on the server,
2.Enter the Raspberry Pi's IP address in the white IP edit box,
3.Click "Connect" to connect client to the Raspberry Pi.
Then you need calibrate the robot in the Calibration section. After the calibration is completed, robot dog can be controlled to move.
Note: when Raspberry Pi is shut down, server will be closed. You need open server again the next time.

Run client on Raspberry Pi(Linux) system

Install openCv library

Execute following 3 commands in turn.

```
sudo apt-get install -y libopencv-dev python3-opencv

sudo pip3 install opencv-contrib-python

sudo apt-get install -y libatlas-base-dev libjasper-dev
```

Open client

Enter the following commands at the terminal.

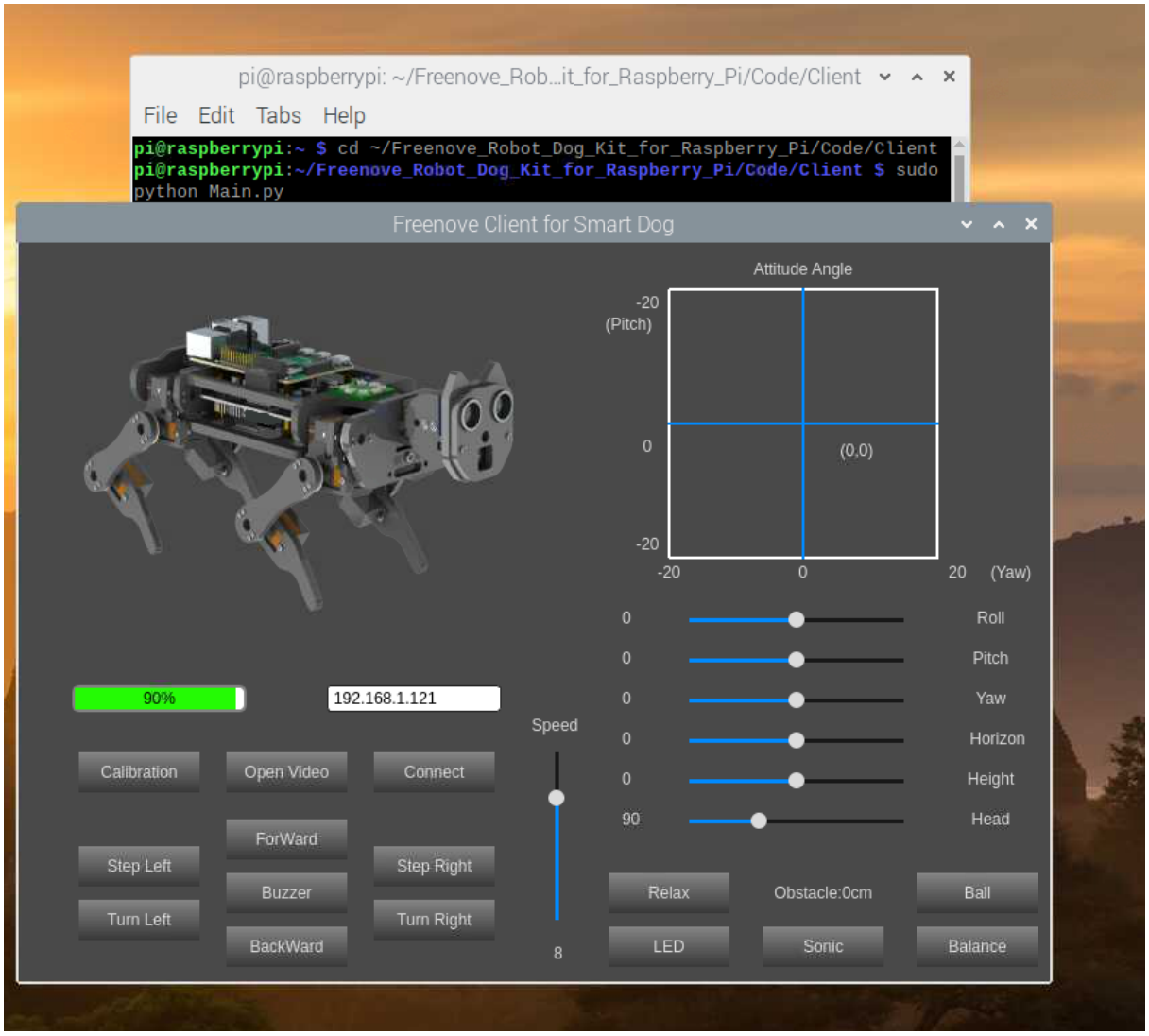
1. Use the cd command to go to the directory where Main.py is located.

```
cd ~/Freenove_Robot_Dog_Kit_for_Raspberry_Pi/Code/Client
```

2. Run Main.py:

```
sudo python Main.py
```

The interface is shown below. The face recognition cannot work well because it need more computing power than Raspberry Pi has.



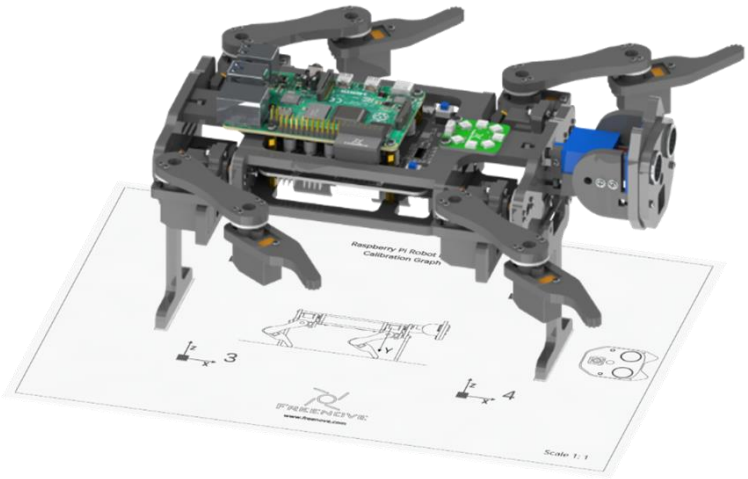
After the client is successfully opened,
1.You need to open the Raspberry Pi and Turn on the server,
2.Enter the Raspberry Pi's IP address in the white IP edit box,
3.Click "Connect" to connect client to the Raspberry Pi.
After the connection is successful, you need to calibrate the four legs of the robot dog in Calibration section.
After the calibration is completed, the robot dog can be controlled to move.
Note: when Raspberry Pi is shut down, server will be closed. You need open server again the next time.

Calibration

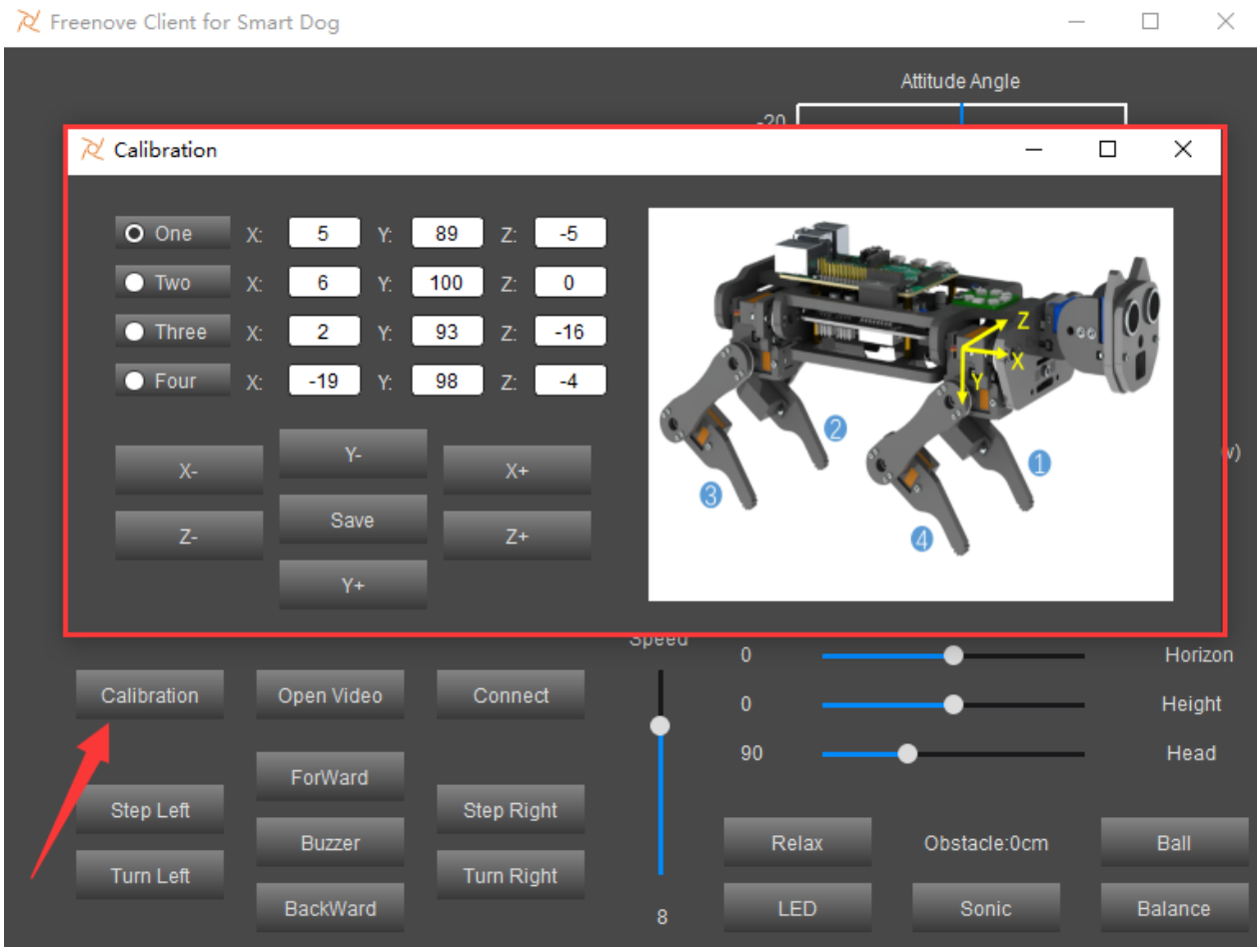
You can refer to this video: https://youtu.be/l2v9PdwQdvY

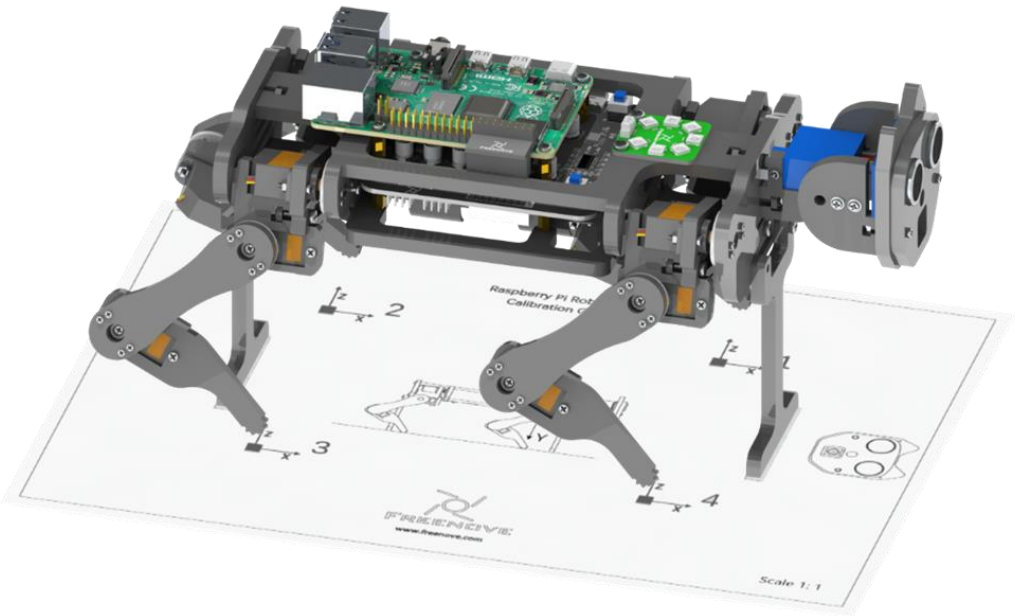
Calibrate the robot.

1. Lay the calibration paper on a horizontal hard table.
2. Install the calibration supports on the robot dog and place it on the corresponding position on the calibration paper.

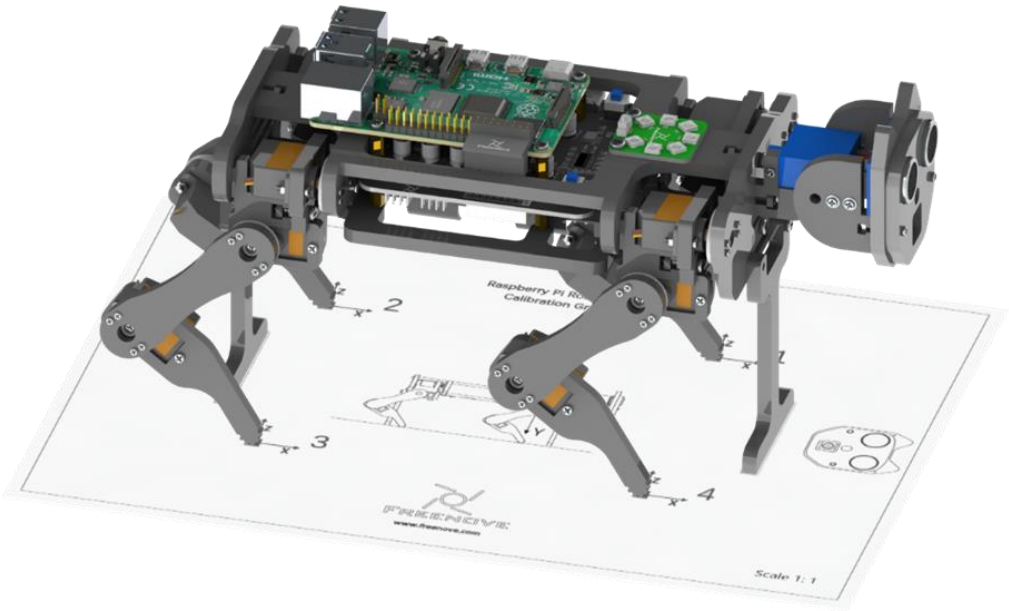


3. Connect client with server successfully, click the "Calibration" button on the client, then a calibration window pops up. And the robot's legs will automatically put to the following position.





4. Calibrate the four legs to make 4 foot points fall to the corresponding positions. Feel the contact force between the robot dog and the ground during calibration. Future control depends on this step.



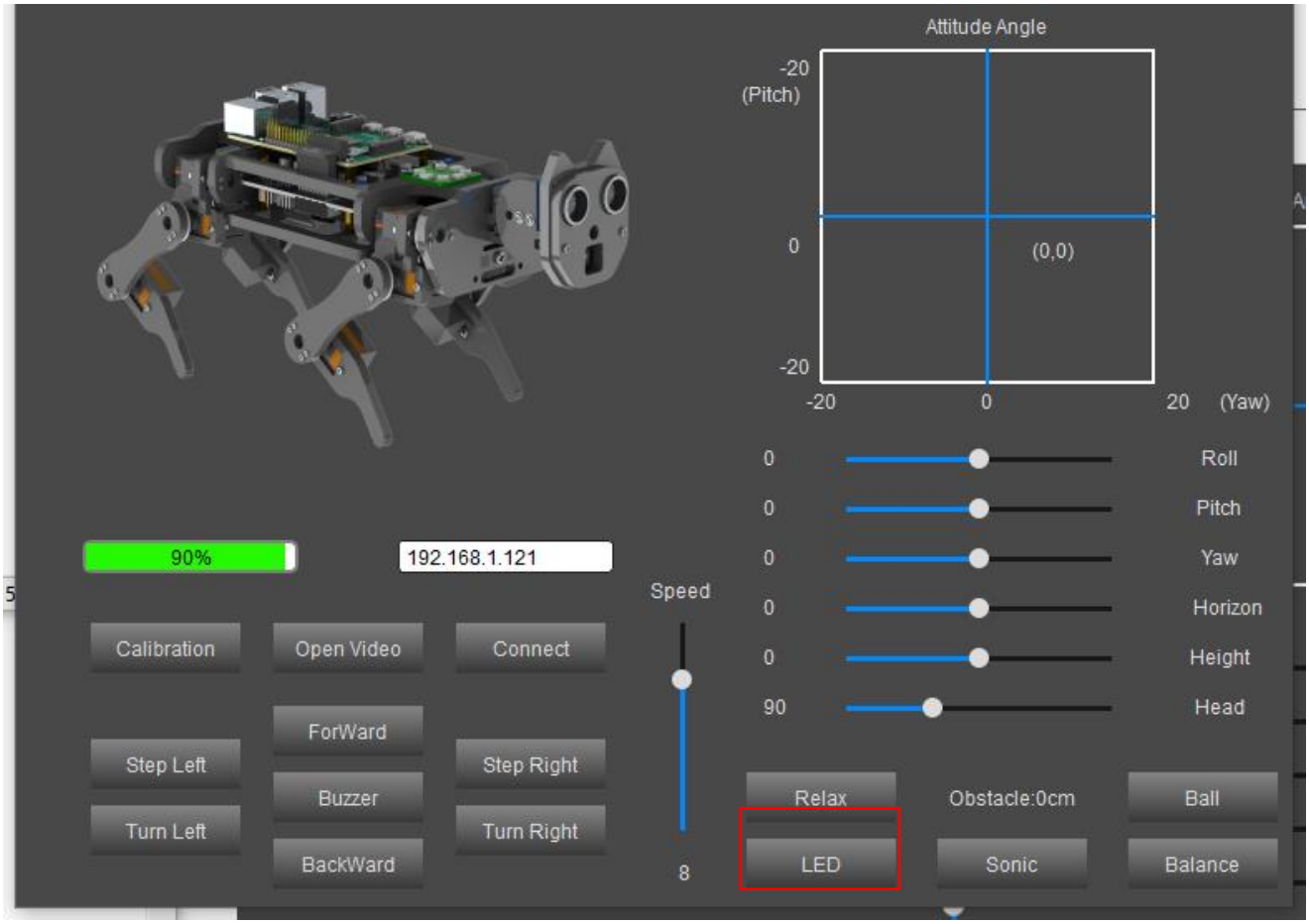
5. After all the 4 points have reached the designated position, click "Save" to save. Now calibration is completed.

NOTE:

1. It is best to have the robot dog walk on a **flat hard** floor. The robot dog may not walk well on other grounds.
2. **There is an offset when controlling the robot dog to walk straightly, which is normal.** It is caused by the installation error and the error of the servo itself. As a result, the left and right legs cannot be completely symmetrical, and may differ by 1mm or 2mm. As the walking distance accumulates, this error is constantly superimposed. Then there will be a large offset from the original position.
3. If the calibration is not good enough, it will affect walking. You can **recalibrate** the robot with same steps.

Control

After calibrating successfully, you can control the robot dog to move.

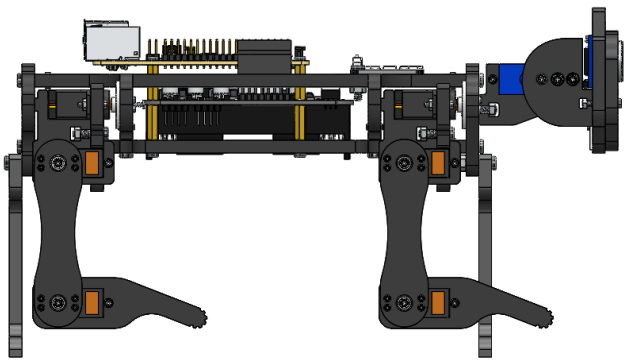


Relax mode.

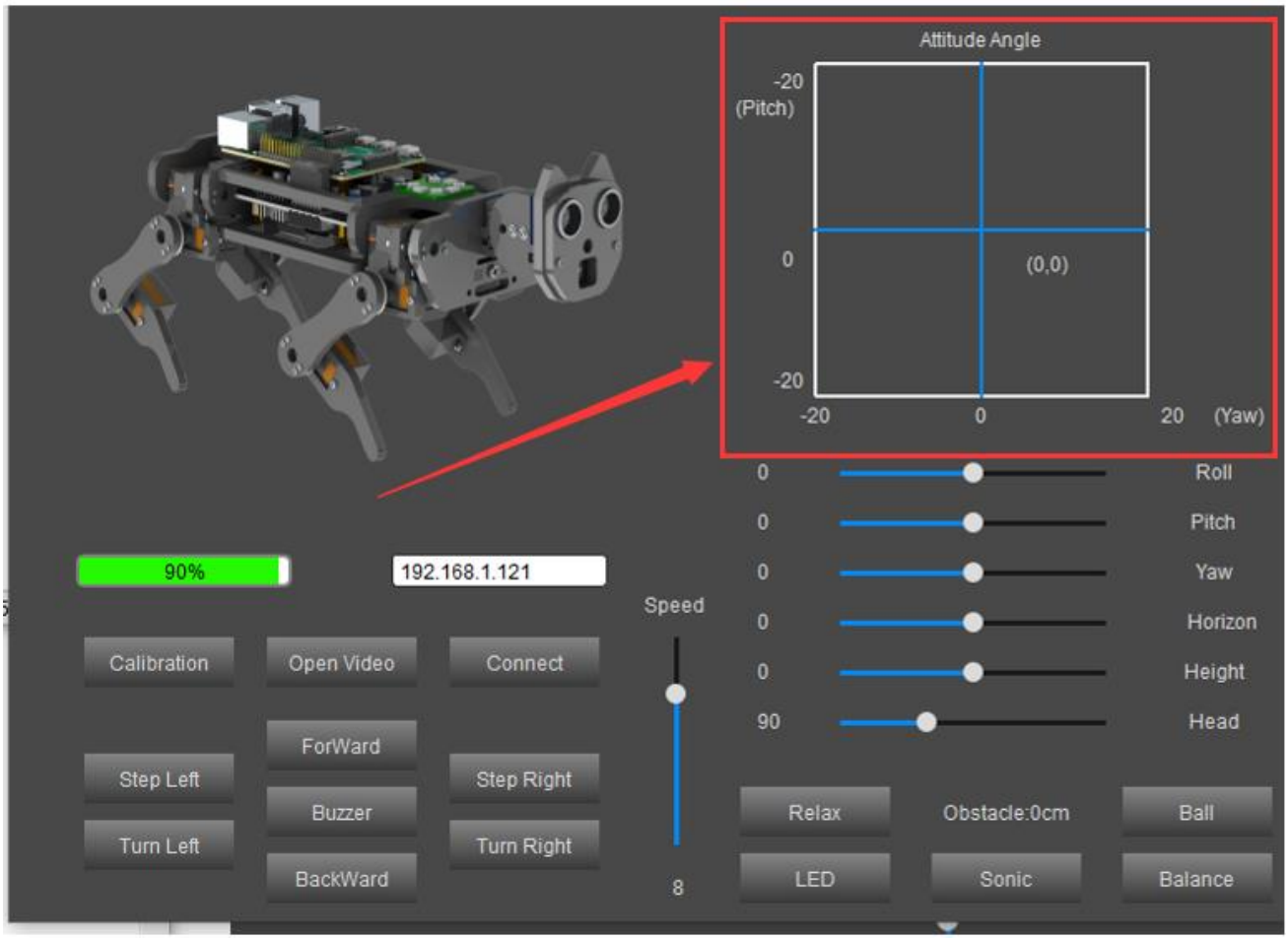
a) When the robot dog moves for 3 minutes at a time, it will feel tired (the servo will get hot). In order to protect the servo, the robot will get into relax mode for 1 minute. **During this time, it won't respond to any motion command.** You can still use the functions of LED, buzzer, real-time video and so on.

b) When the robot dog moves for less than3 minutes and rests for 1 minute. The timer will start from 0. Then the robot can move for 3 minutes again.

c) If the robot isn't tired and is standing, when the robot does not receive motion command for 10s, it will get into relax mode. In this situation, it will respond to all commands.



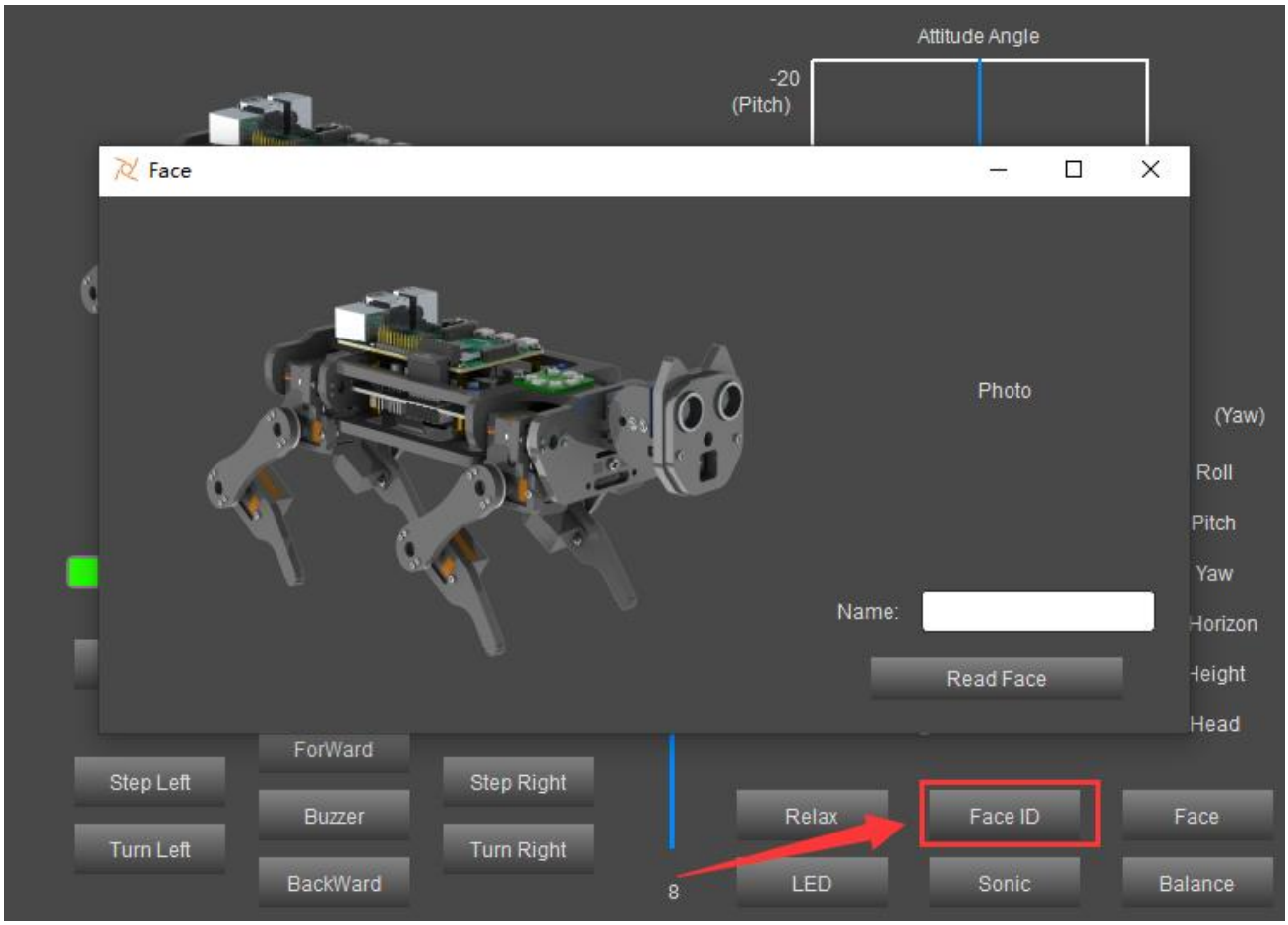
You can click and drag the cross cursor in the coordinate system to adjust the yaw and pitch angle of body.



The following is the corresponding operation of the buttons and keyboards.

Button on Client	Key	Action
ForWard	W	Move
BackWard	S	Back off
Turn Left	A	Turn left
Turn Right	D	Turn right
Step Left	Q	Step Left
Step Right	E	Step Right
Connect/ Disconnect	C	On/Off Connection
Open Video/ Close Video	V	On/Off Video
Calibration	M	Open calibration interface
Buzzer/Noise	Space	On/Off Buzzer
Relax	R	Relax station
LED	L	Open LED control interface
Sonic	U	Measure and display distance from obstacle
Balance	B	Open/Close balance mode
Face/Ball/Close	F	Face recognition and ball tracking
Face ID		Input images of human faces

Input images of human faces and recognize them.



1. Click "Face ID".
2. Enter name in the box of the pop-up window.
3. Click "Read Face".
4. A picture is taken and recorded every two seconds, and a total of 30 images will be taken. **Try to show faces at different distances and angles during the shooting process.** (When the face is not recognized, the picture will not be recorded until the face is recognized again.)
5. **After the 30 images are recorded, you can close the window** and click "Face" on the Right to identify the face.

Note: Do NOT use different names when re-entering images of the same person to avoid client working abnormally.

The function of SliderBar is below:

SliderBar	Function
Head	Adjust head angle.
Height	Control body height.
Horizon	Make body move forward or back.
Yaw	Adjust Yaw angle of body
Pitch	Adjust Pitch angle of body
Roll	Adjust Roll angle of body

Note:

If the robot work abnormally, please check following contents.

1. Check the battery level. Low battery level will make the server shut down.
2. Check if the wireless connection is disconnected.
3. Check if the Raspberry Pi is stuck.
4. If all three points above work well, please restart server and client. If there still is any abnormality, please contact us by email (**support@freenove.com**). We will help you.

Free your innovation

If you have any concerns, please feel free to contact us via support@freenove.com

If you want to write your own program to control the robot dog, you can follow this chapter.

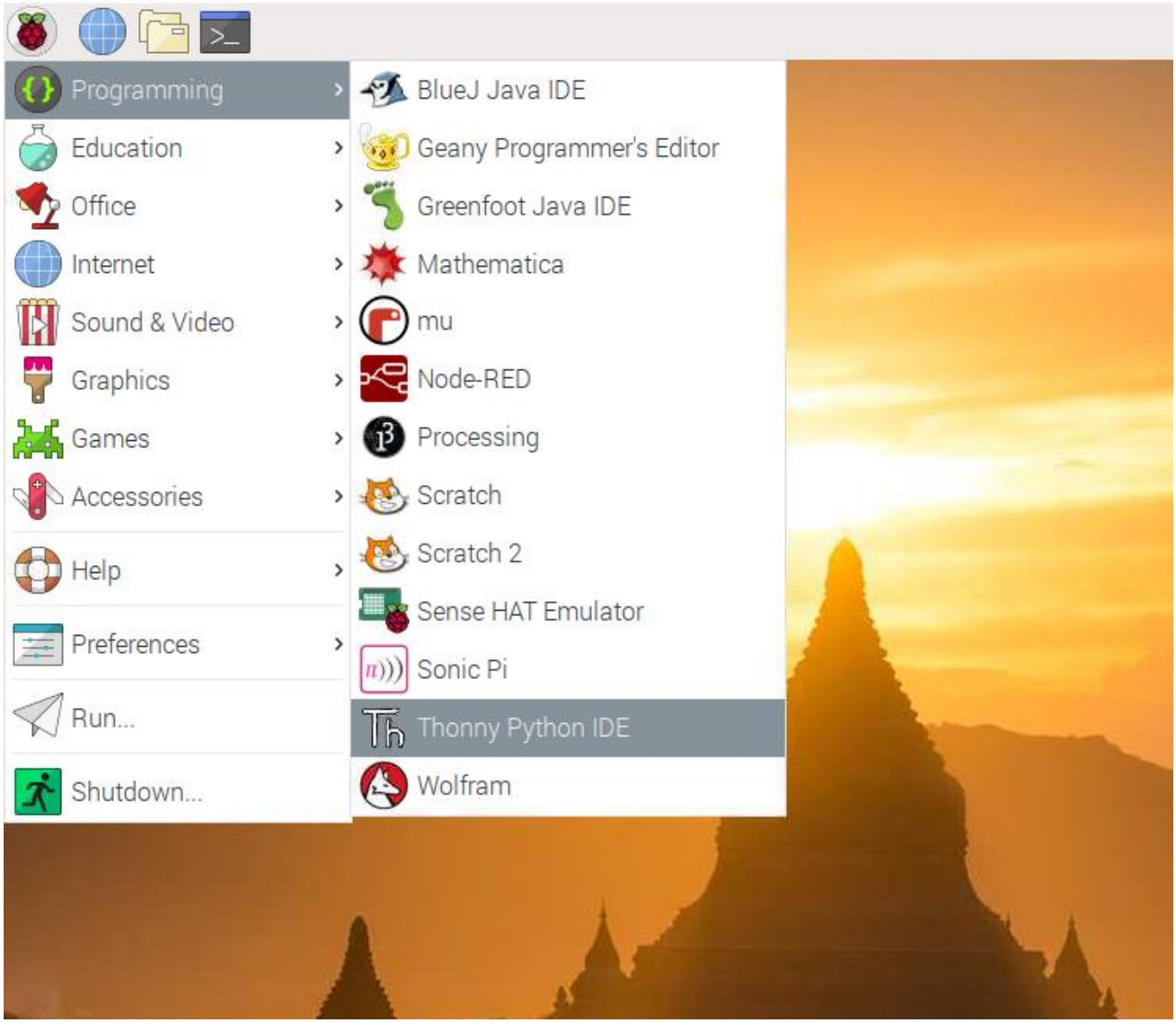
The robot dog program is based on python3. If your python is python2 by default, please change to python3.

If you have never learned python before, you can learn some basics through the following links:

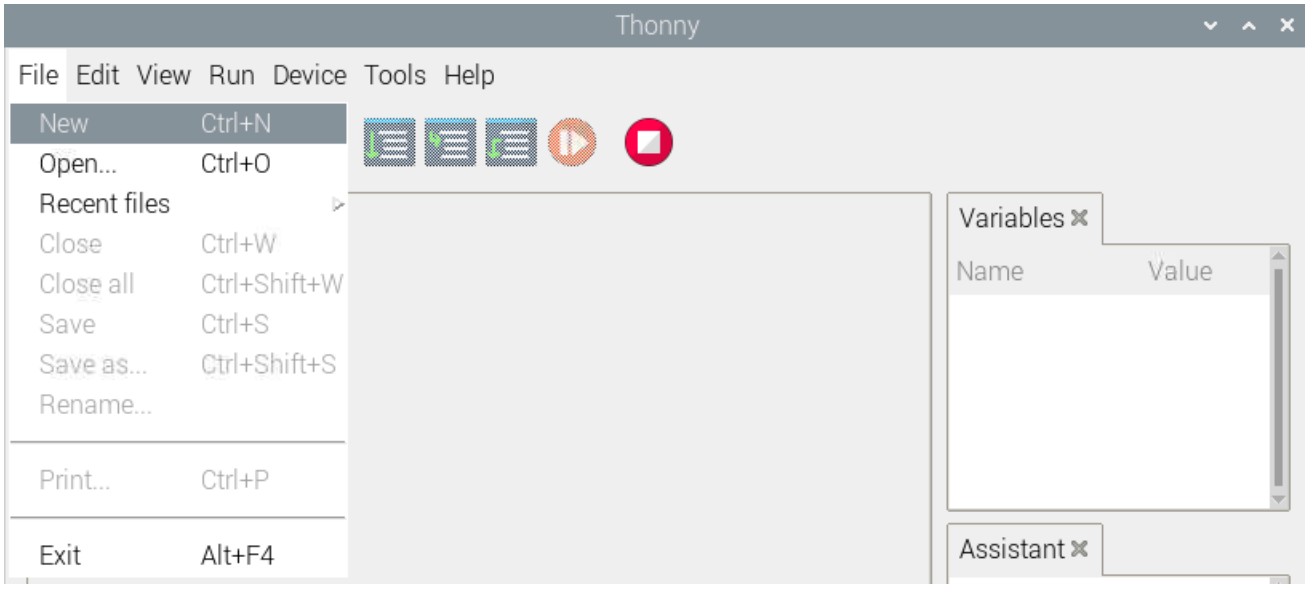
https://python.swaroopch.com/basics.html

Program

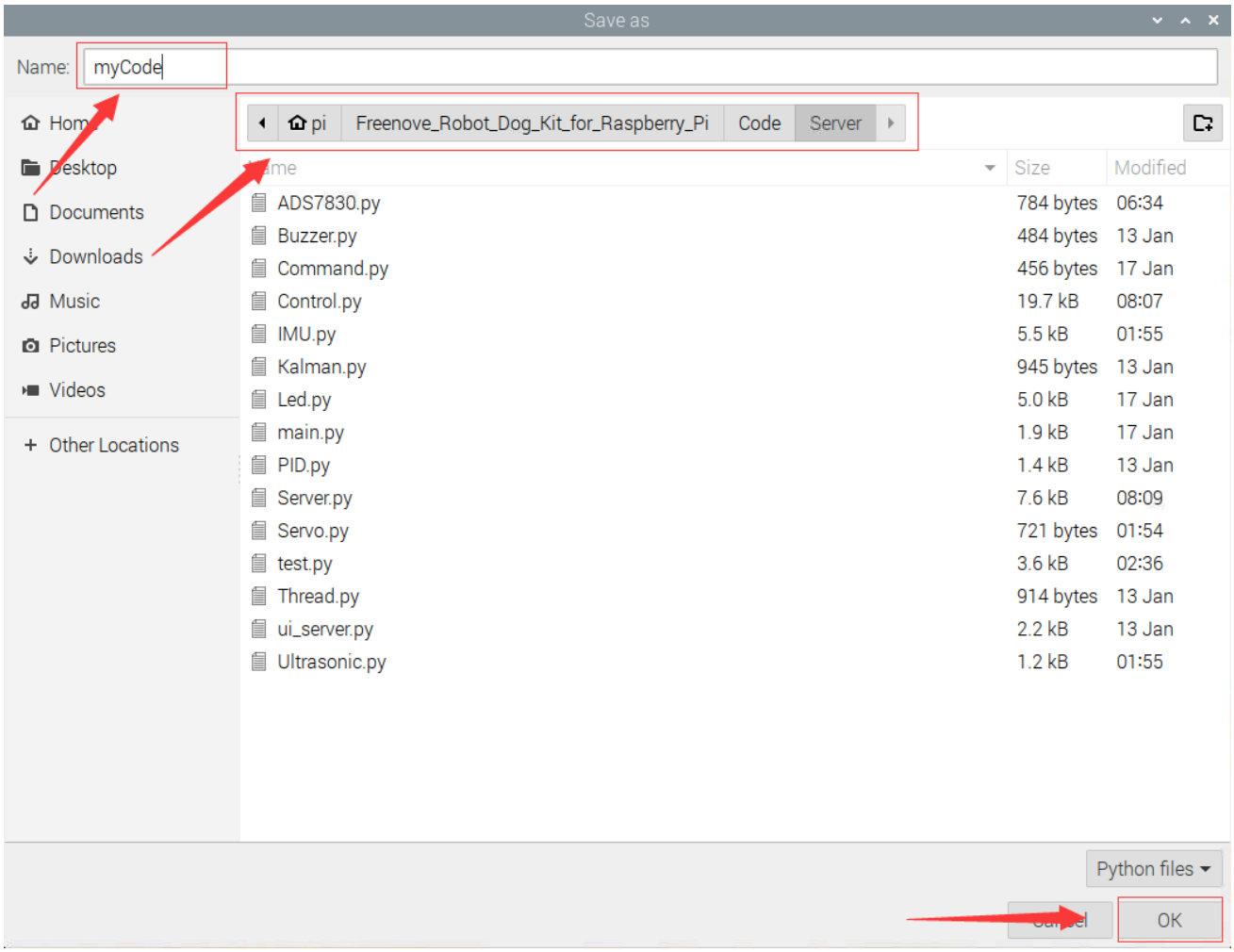
First, open Thonny Python IDE which is easy to use for beginners.



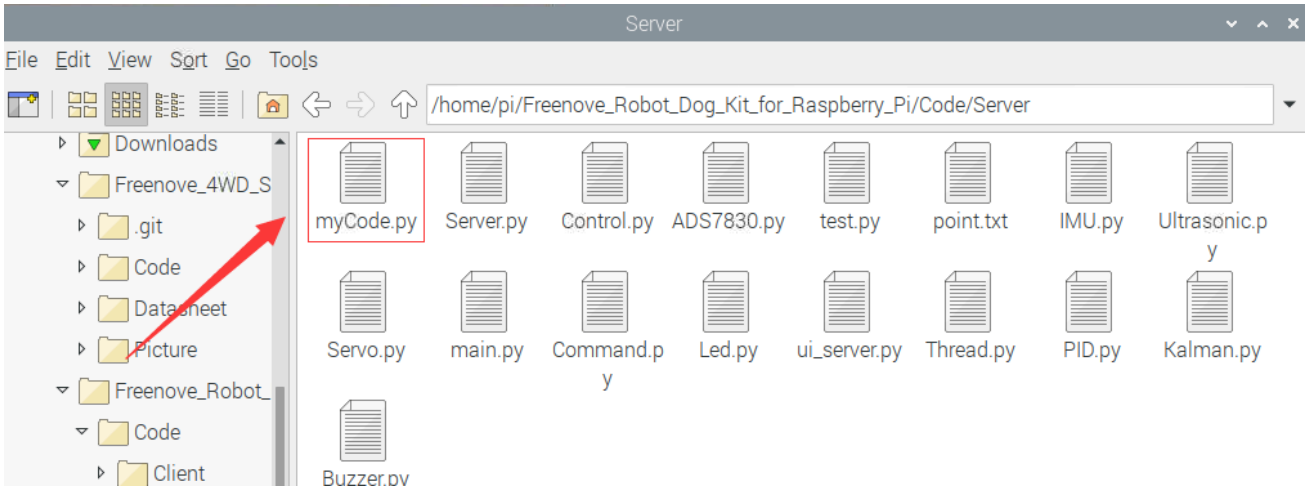
Create a new file.



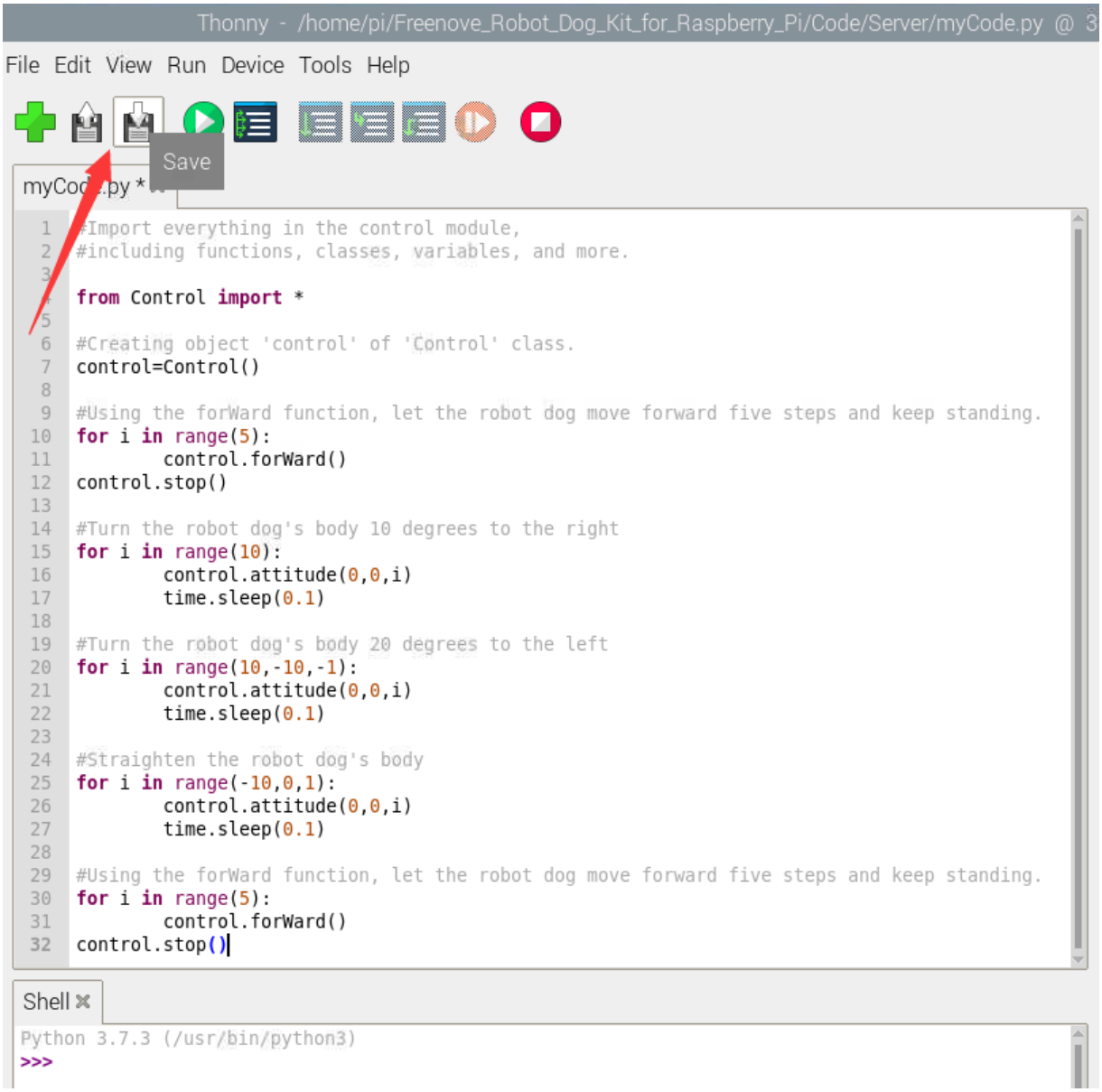
Name it with myCode as an example and save it in Server folder of robot code folder.



Open the Server folder of the robot dog code and you can see the file you created.



Write the code in myCode.py as below and click save after finished, as shown below.

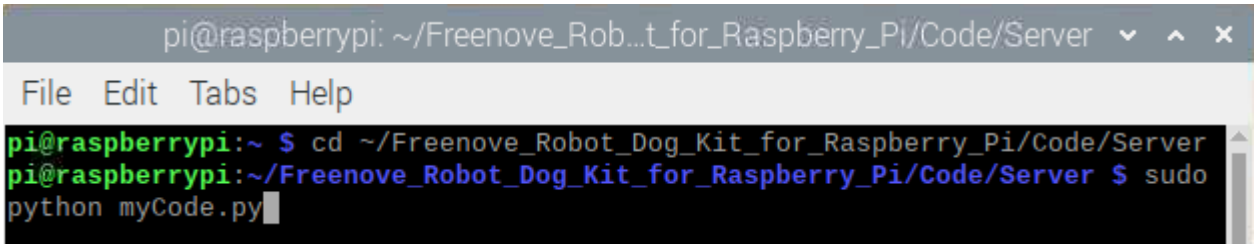


Type the following command to enter the directory where myCode.py is located.

```
cd ~/Freenove_Robot_Dog_Kit_for_Raspberry_Pi/Code/Server
```

Run myCode.py

```
sudo python myCode.py
```



You can see that your robot dog moves forward for 5 steps, then twists its body at 10 degrees to the right, then 20 degrees to the left, then returns to the original attitude and move forward 5 steps and then stops.

Related py files

Some important functions included in the py files are listed below. If you want to see more detailed code content, please open the corresponding py file to check directly.

ADS7830.py

Function	Description
power(channel)	Enter 0 to return the battery voltage value.

Buzzer.py

Function	Description
run(command)	Enter 0, then the buzzer will not sound. Enter 1, the buzzer will sound.

Contorl.py

Function	Description
coordinateToAngle(x,y,z)	Enter the foot coordinate then return servo angle of each joint in one leg.
angleToCoordinate(a,b,c)	Enter the leg joint angle then return the foot coordinate.
Condition()	Execute corresponding action commands
map()	Mapping a value from one range to another
backWard ()	Move backward one step
forward()	Move forward one step
turnLeft()	Turn left one step
turnRight()	Turn right one step
stop()	Stop.
setpLeft()	Move to left one step
setpRight()	Move to right one step
upAndDown()	Adjust height of body
beforeAndAfter()	Make body move forward or backward
attitude(r,p,y)	Set roll, pitch, and yaw angle of the robot.

IMU.py

Function	Description
imuUpdate()	Update the current roll, pitch, and yaw angle of the robot.

Led.py

Function	Description
ledIndex(Index, R, G, B)	Turn on one LED and set color.
wheel(pos)	Create different RGB values.
LED_TYPR(order,R_G_B)	Change the order in which the LED color data is transmitted. When the value of the order parameter is "RGB", the order of data transmission should be: R-G-B; when the value of the order parameter is "GBR", and the order of data transmission should be: G-B-R
theaterChaseRainbow(strip, wait_ms)	The function is used to make 7 LEDs show one color at the same time, and change various colors to blink. The blinking interval is wait_ms, and the default value is 50ms
rainbow(strip, wait_ms)	This function achieves the effect of rainbow breathing. It makes 8 LEDs display same color at the same time, and then change all various colors like breathing. The interval is wait_ms. The default value is 20ms.
rainbowCycle(strip, wait_ms)	This function also achieves the effect of rainbow breathing. but unlike rainbow(), it makes eight LEDs to display different colors at the same time, and then change various color separately. The interval is wait_ms. The default value is 20ms

Servo.py

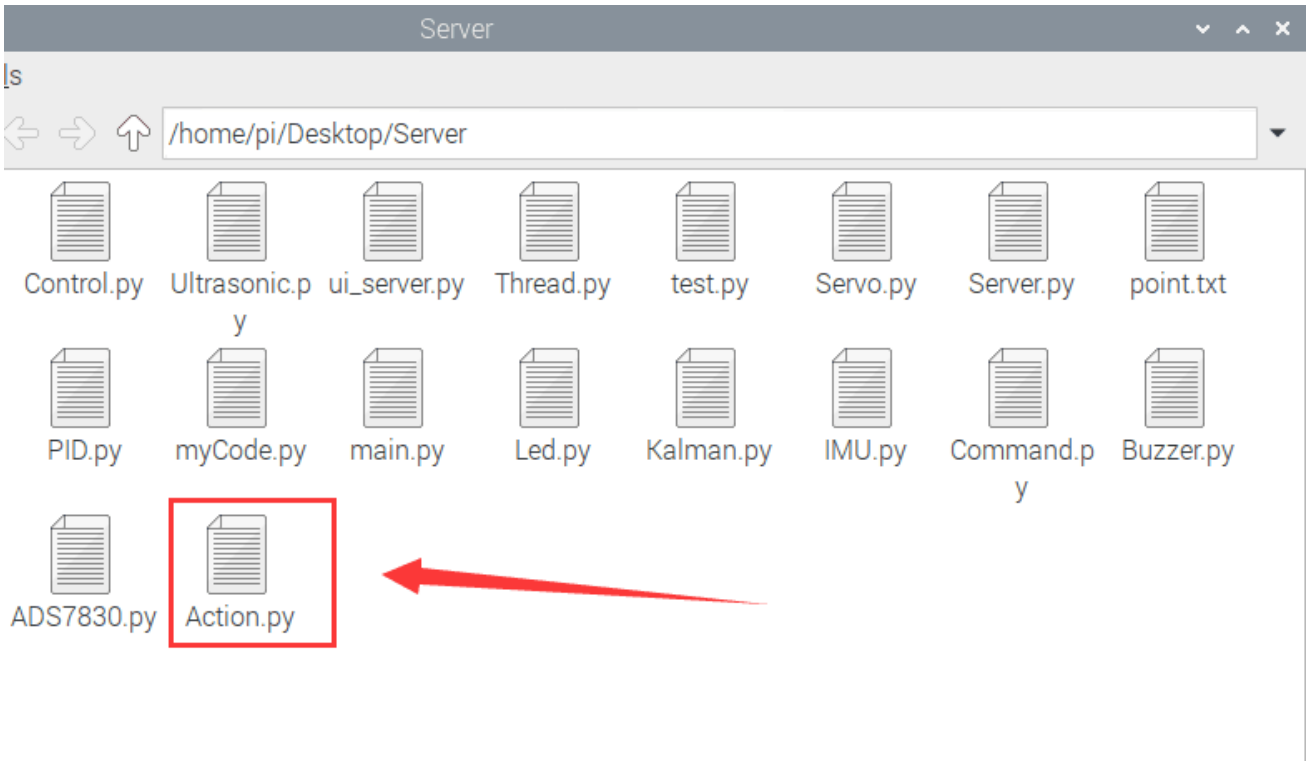
Function	Description
setServoAngle(channel, angle)	Turn the servo of the specified channel to specified angle.

Ultrasonic.py

Function	Description
getDistance()	Obtain the distance between the obstacle in front and the ultrasound.

Custom Action

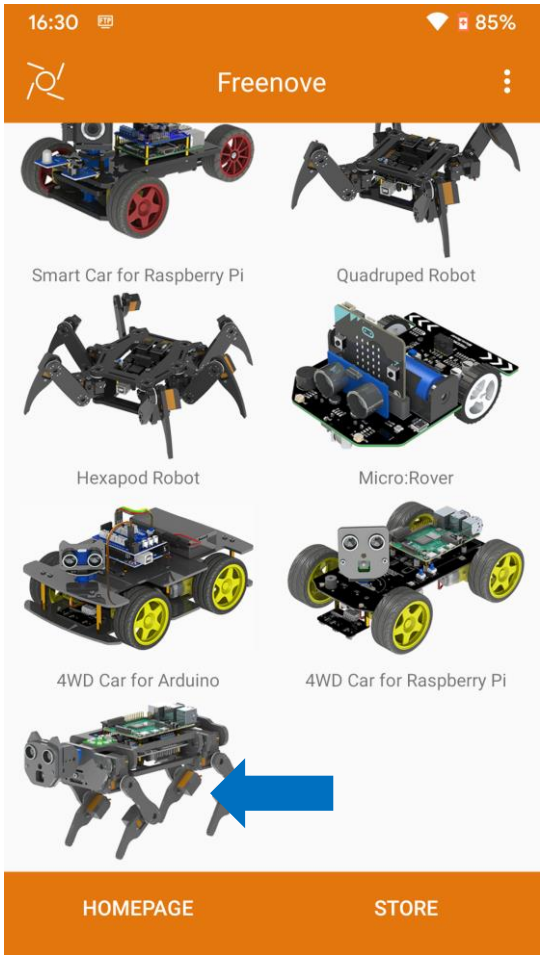
We have also customized some specific actions, such as push-up, greeting, etc. You can run "Action.py" to play.



Android and iOS app

You can download and install the Freenove **Android app** from below links:
On Google play:
https://play.google.com/store/apps/details?id=com.freenove.suhayl.Freenove
On GitHub:
https://github.com/Freenove/Freenove_App_for_Android
In this GitHub repository, you can find the App instruction (Tutorial.pdf).

You can download and install the Freenove **iOS app** by searching **freenove** in app store.



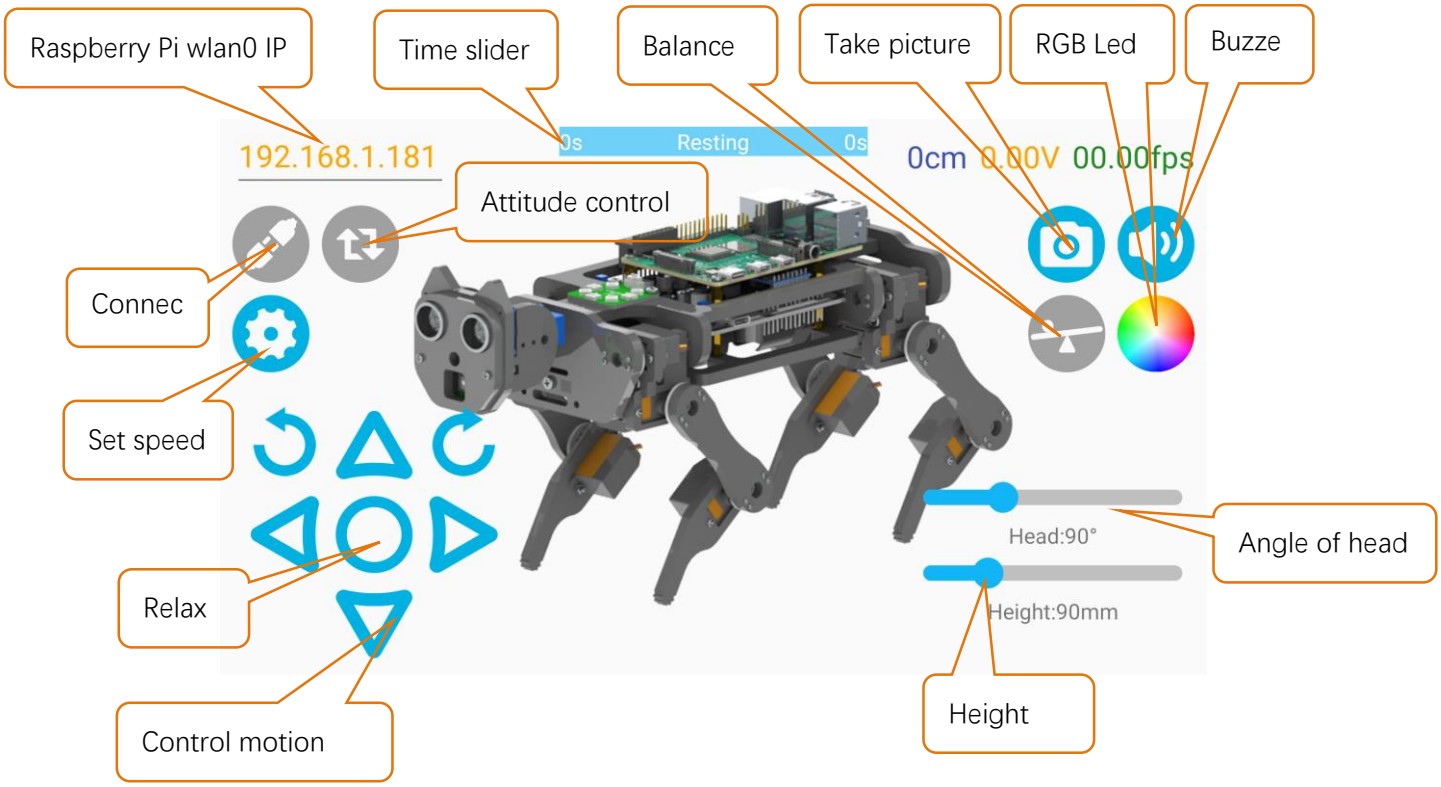
Relax mode.

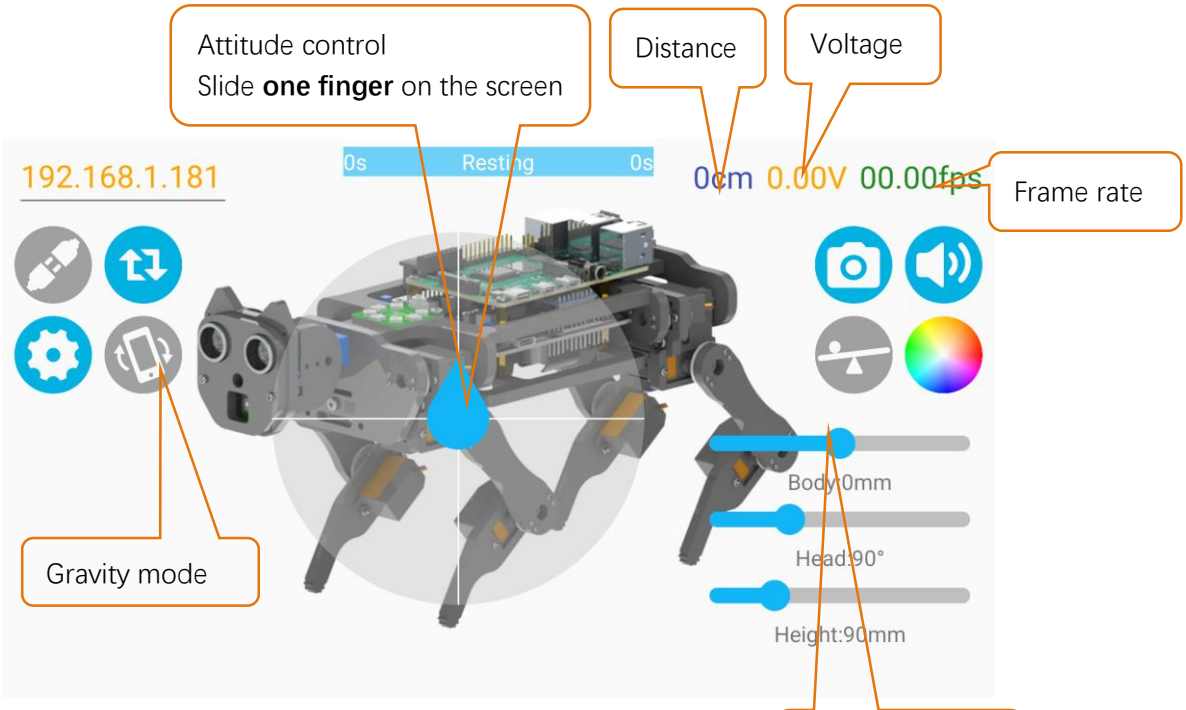
a) When the robot dog moves for 3 minutes in total, it will feel tired (the servo will get hot). In order to protect the servo, the robot will get into relax mode for 1 minute. **During this time, it won't respond to motion command.** You can still use the functions of LED, buzzer, real-time video and so on.
b) When the robot dog moves for <3 minutes and then the robot rest for 1 minute. The timer will start from 0. Then the robot can move for 3 minutes again.
c) If the robot isn't tired and is standing, when the robot does not receive motion command for 10s, it will get into relax mode. In this situation, it will responds to any commands.

The followings are the features of this app.

First, you need to turn on the Server. Then enter your raspberry pi IP address and click connect icon.

On the top of the interface, there is a timer slider to indicate the time for moving or resting.





What's Next?

THANK YOU for participating in this learning experience!

We have reached the end of this Tutorial. If you find errors, omissions or you have suggestions and/or questions about the Tutorial or component contents of this Kit, please feel free to contact us: **support@freenove.com**
We will make every effort to make changes and correct errors as soon as feasibly possible and publish a revised version.

If you want to learn more about Arduino, Raspberry Pi, Smart Cars, Robotics and other interesting products in science and technology, please continue to visit our website. We will continue to launch fun, cost-effective, innovative and exciting products.
http://www.freenove.com/

Thank you again for choosing Freenove products.