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 & Custommer Service

You may find somethings missing or broken, or some difficulty to learn the kit.
Freenove provides free and quick support, including but not limited to:

- Quality problems of products
- Problems in using products
- Questions for learning and technology
- Opinions and suggestions
- Ideas and thoughts

If you have any concerns, please send email to us:

support@freenove.com

And suggestions and feedbacks are welcomed. Many customers offered great feedbacks. According to that, we are keeping updating the kit and the tutorial to make it better. Thank you.

Safety

Pay attention to safety when using and storing this product:

- Do not expose children under 6 years of age to this product. Put it out of their reach.
- Children lack safety ability should use this product under the guardianship of adults.
- This product contains small and sharp parts. Do not swallow, prick and scratch to avoid injury.
- This product contains conductive parts. Do not hold them to touch power supply and other circuits.
- Some parts will rotate or move when it works. Do not touch them to avoid being bruised or scratched.
- The wrong operation may cause overheat. Do not touch and disconnect the power supply immediately.
- Operate in accordance with the requirements of the tutorial. Otherwise, the parts may be damaged.
- Store the product in a dry place and avoid direct sunlight.
- Turn off the power of the circuit before leaving.

About

Freenove provides open source electronic products and services.

Freenove is committed to helping customers learn programming and electronic knowledge, quickly realize their creative ideas and product prototypes and launching innovative products. Our services include:

- Kits of robots, smart cars and drones
- Kits for learning Arduino, Raspberry Pi and micro:bit
- Electronic components and modules, tools
- **Product customization service**

You can learn more about us or get our latest information through our website:

http://www.freenove.com

Copyright

FREENOVE®

FREE YOUR INNOVATION

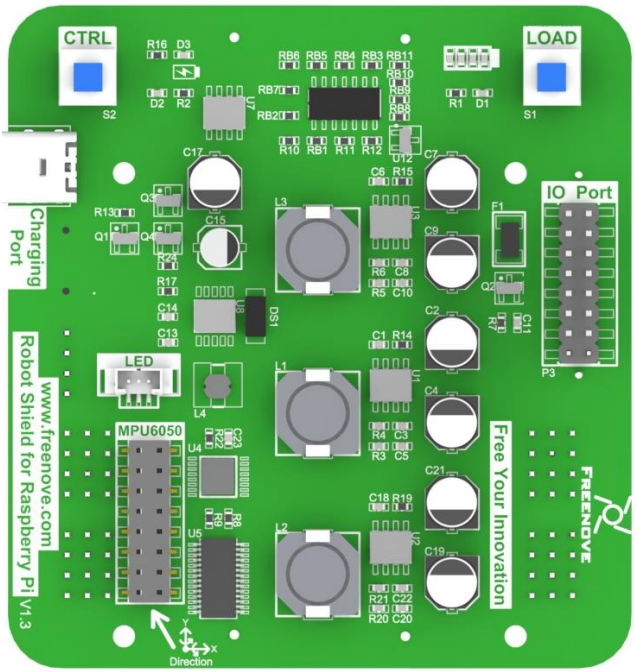
Contents

List

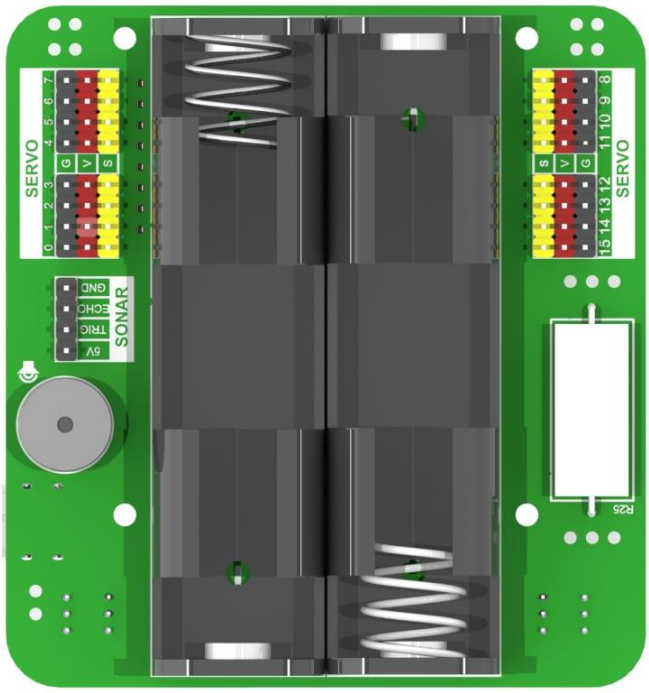
If you have any concerns, please feel free to contact us at support@freenove.com

Robot Shield for Raspberry Pi

Top



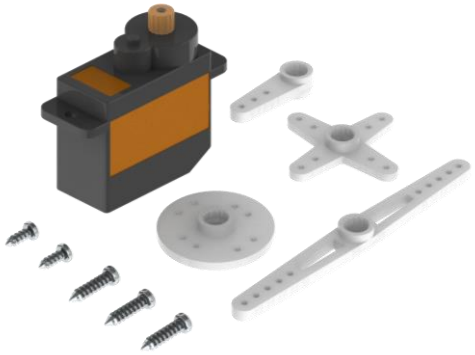
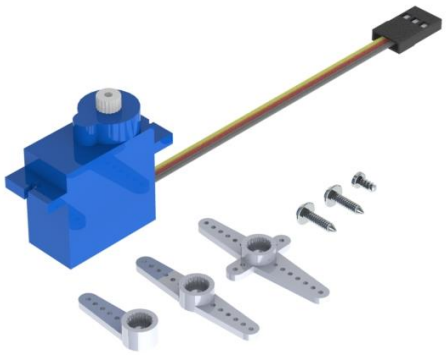
Bottom



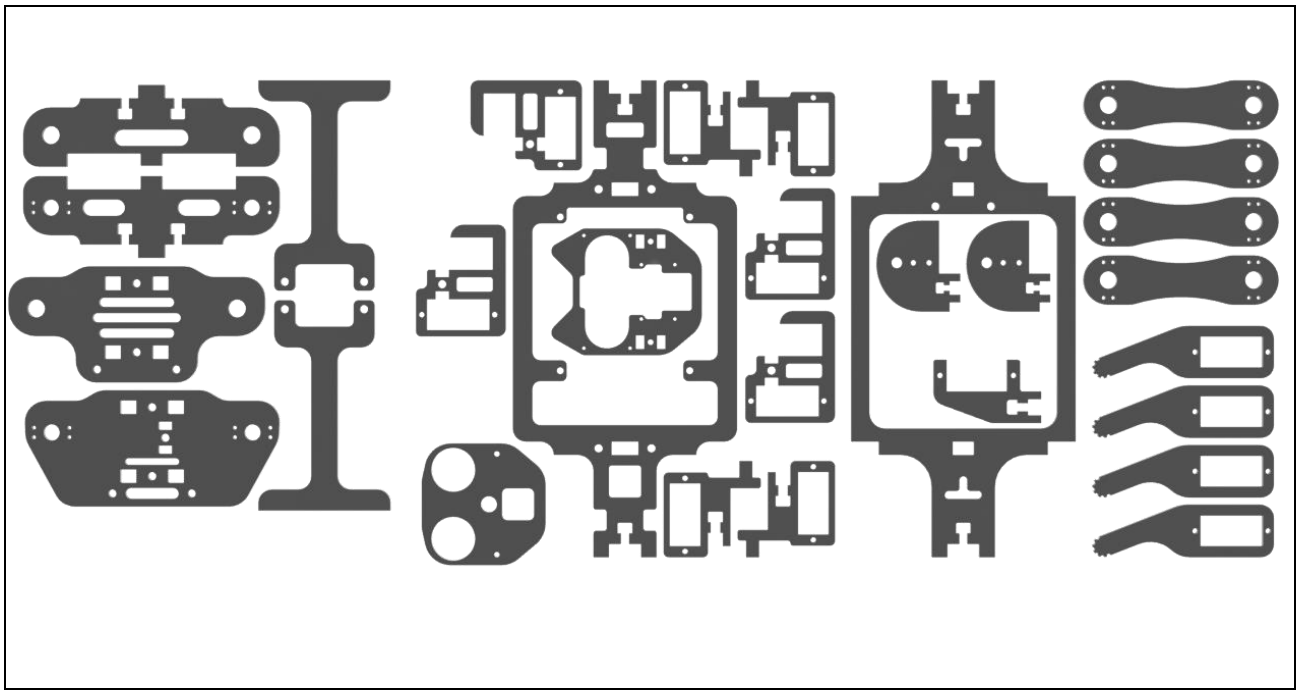
Machinery Parts

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

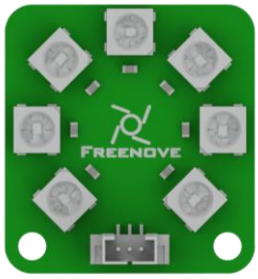
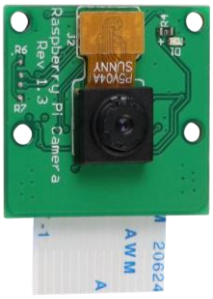
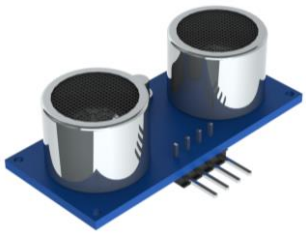
Transmission Parts

ES08MA II servo package x12	S90 servo package x1
	

Acrylic Parts



Electronic Parts

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) 			

Tools

Cross screwdriver (3mm) x1 Cross screwdriver (2mm) x1 	Cable tidy x80cm 	Red ball 

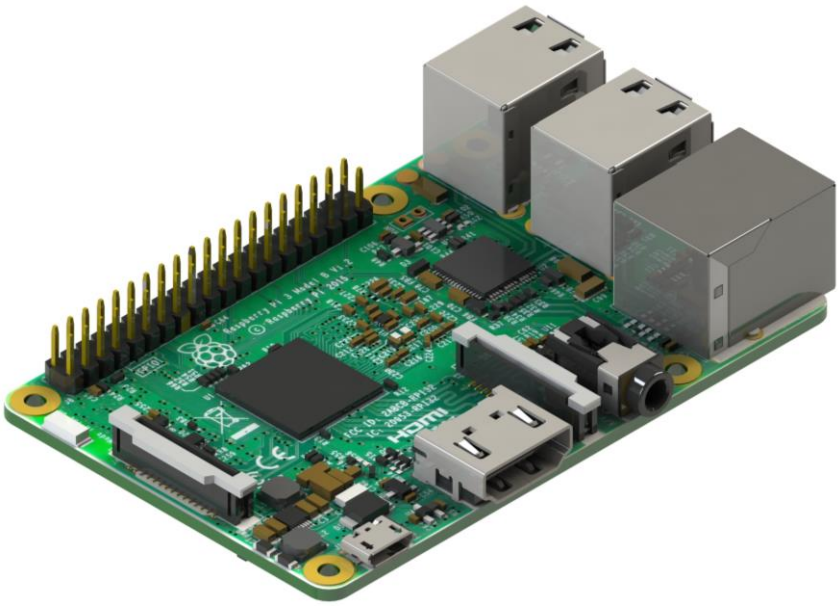
Self-prepared Parts

Two 18650 lithium batteries without protection board.

The continuous discharge current >3A

It is not easy to find proper battery 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. By using this tutorial, you will make a very cool Robot Dog with many functions.

This kit is based on the popular control board Pi Raspberry and program language Python3. Some libraries are needed to be installed. You can share and exchange your experience and design ideas with many enthusiasts all over the world.

The robot needs assembly. It will take much time. Please be patient. The parts in this kit include all electronic components, modules, and mechanical components required for making the robot. And all of them are packaged individually. There are detailed assembly and test instructions in this tutorial.

And if you encounter any problems, please feel free to contact us for fast and free technical support.

support@freenove.com

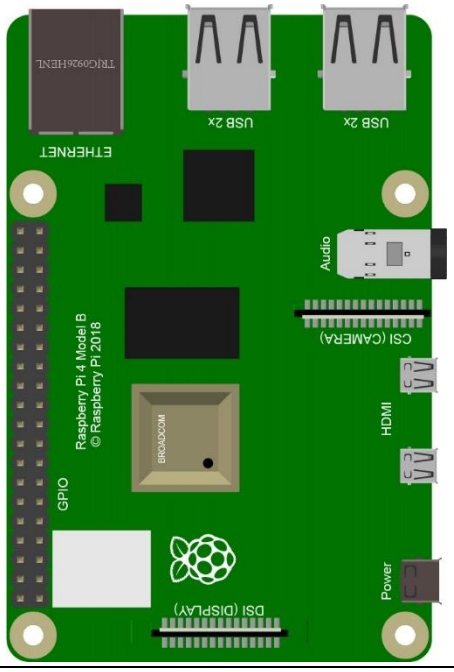
The contents in this tutorial can make users with little technical knowledge to make a Robot Dog. If you are very interested in Raspberry Pi, and want to learn how to program and build the circuit, please visit our website www.freenove.com or contact us for the kits designed for beginners:
Freenove Basic\LCD1602\Super\Ultrasonic\RFID\Ultimate Starter Kit for Raspberry Pi

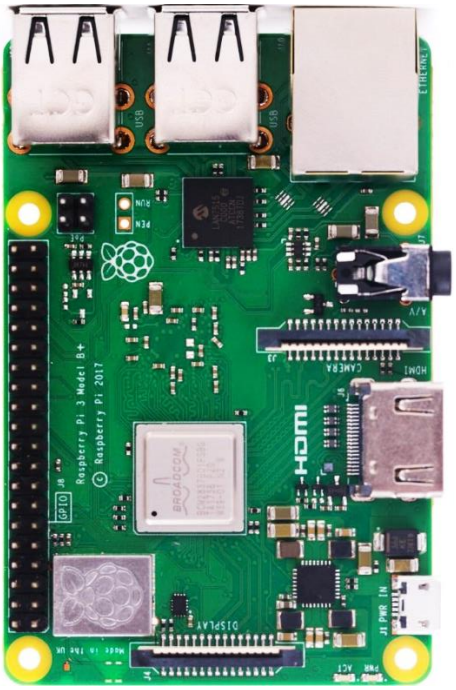
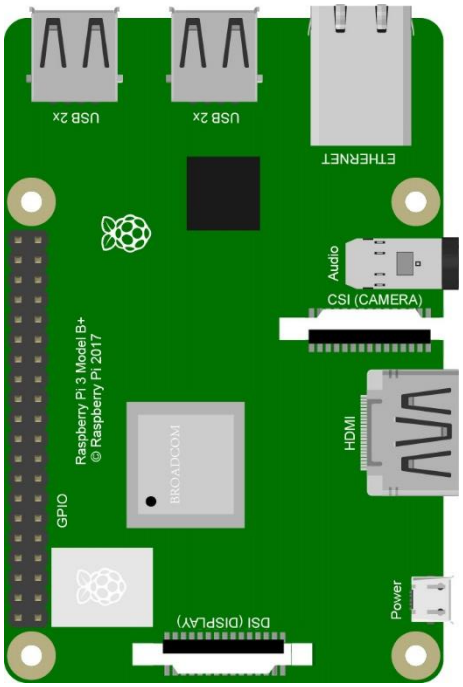
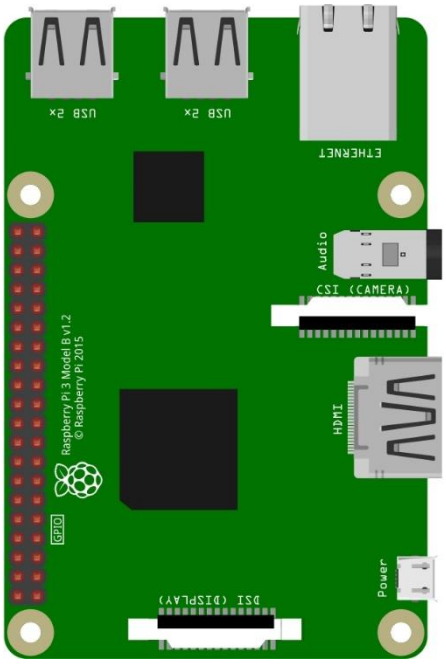
Raspberry Pi Introduction

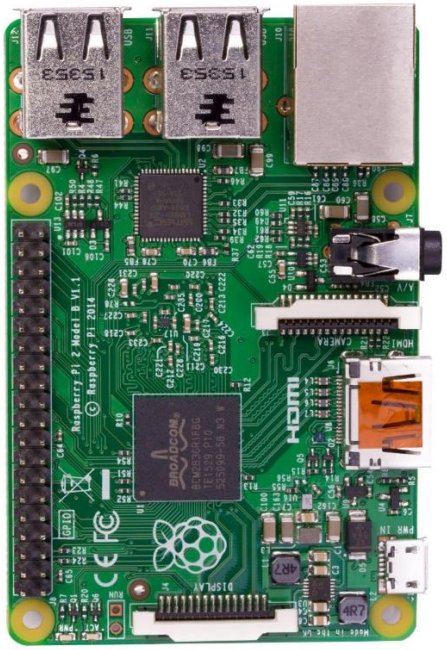
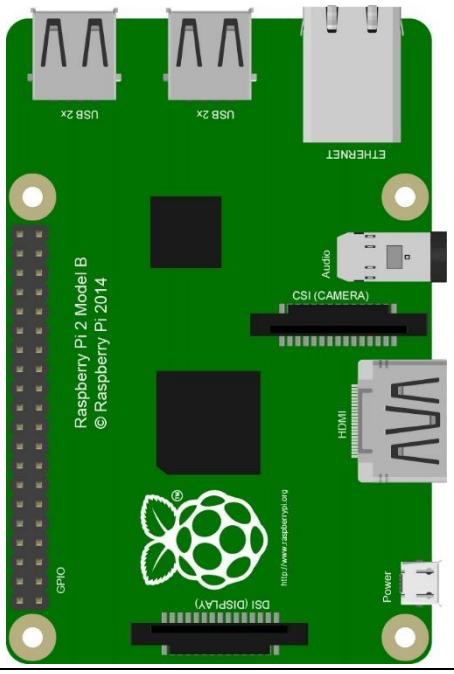
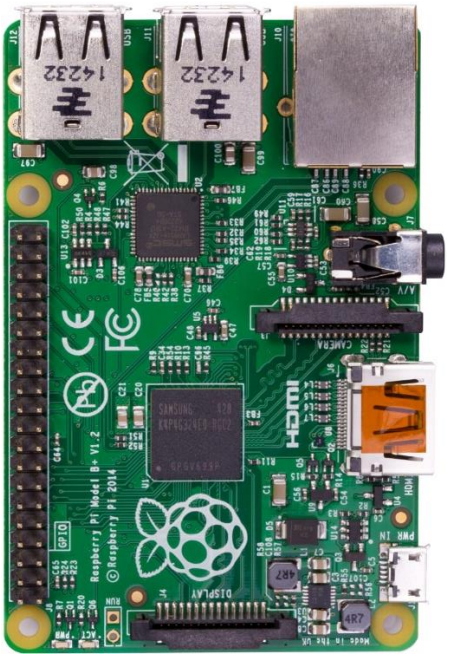
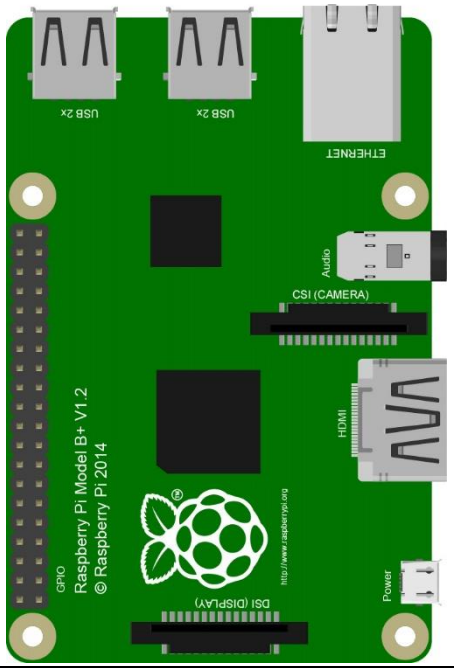
Raspberry Pi (called RPi, RPI, RasPi, the text these words will be used alternately later), a micro-computer with size of a card, quickly swept the world since its debut. It is widely used in desktop workstation, media center, smart home, robots, and even the servers, etc. It can do almost anything, which continues to attract fans to explore it. Raspberry Pi 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 running 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.

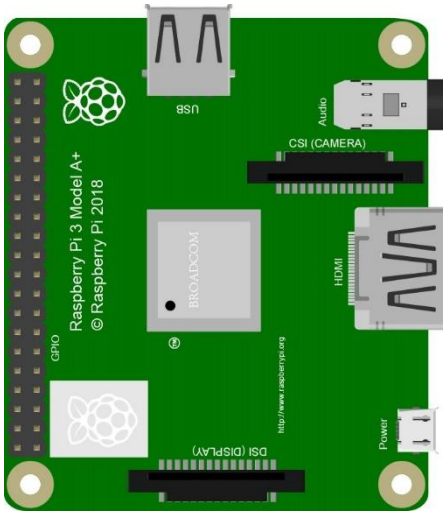
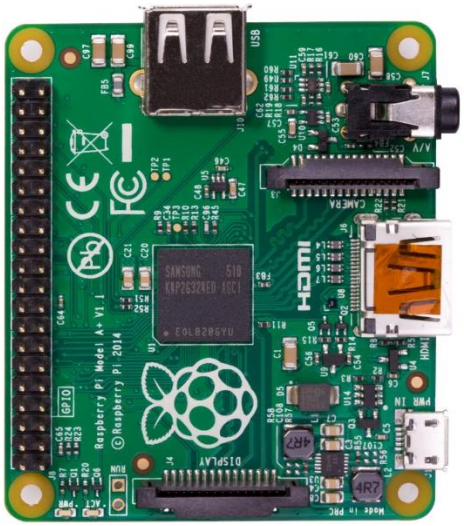
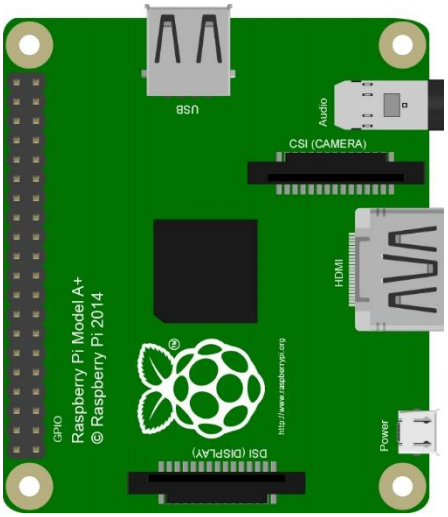
So far, Raspberry Pi has developed to the fourth generation. Changes in versions are accompanied by increase and upgrades in hardware. A type and B type, the first generation of products, have been stopped due to various reasons. Other versions are popular and active and the most important is that they are consistent in the order and number of pins, which makes the compatibility of peripheral devices greatly enhanced between different versions.

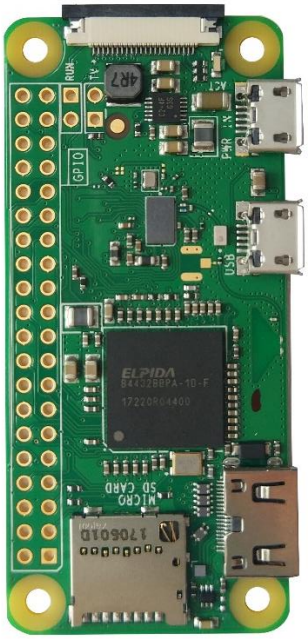
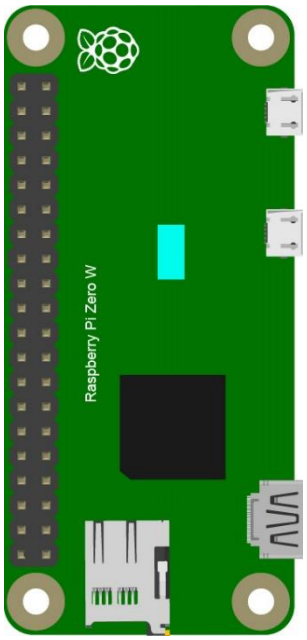
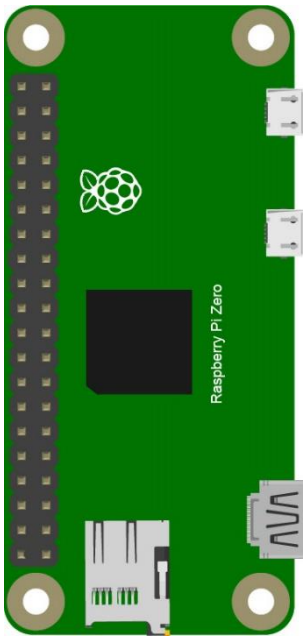
Below are the raspberry pi pictures and model pictures supported by this product.

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

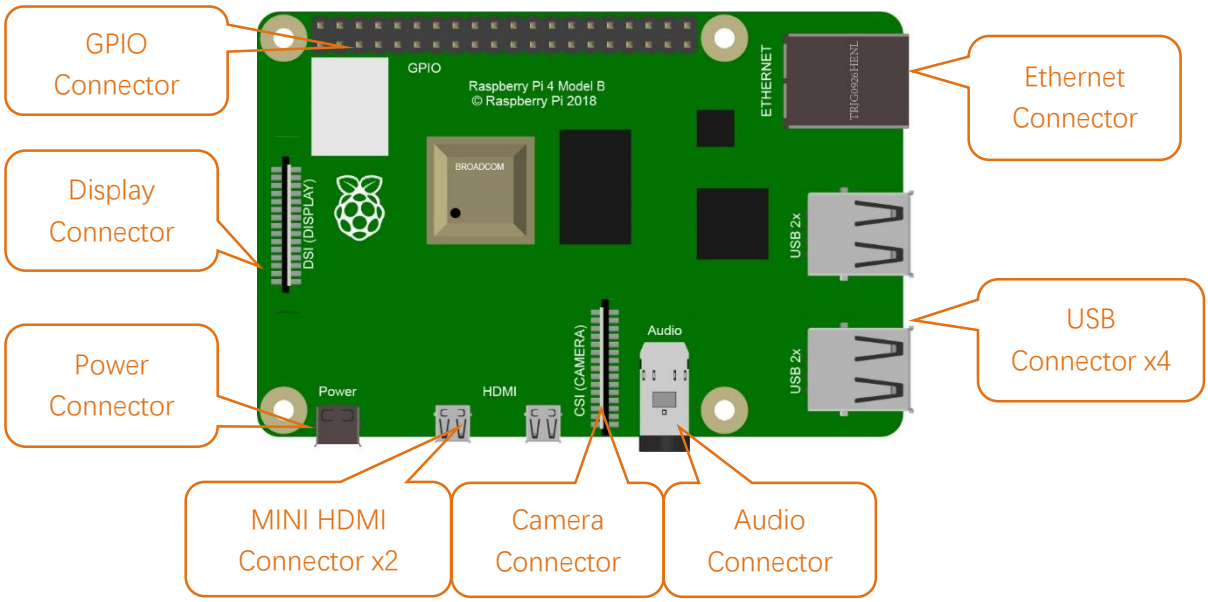
Practicality picture of Raspberry Pi 3 Model B+:	Model diagram of Raspberry Pi 3 Model B+:
	
Practicality picture of Raspberry Pi 3 Model B:	Model diagram of Raspberry Pi 3 Model B:
	

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+: 

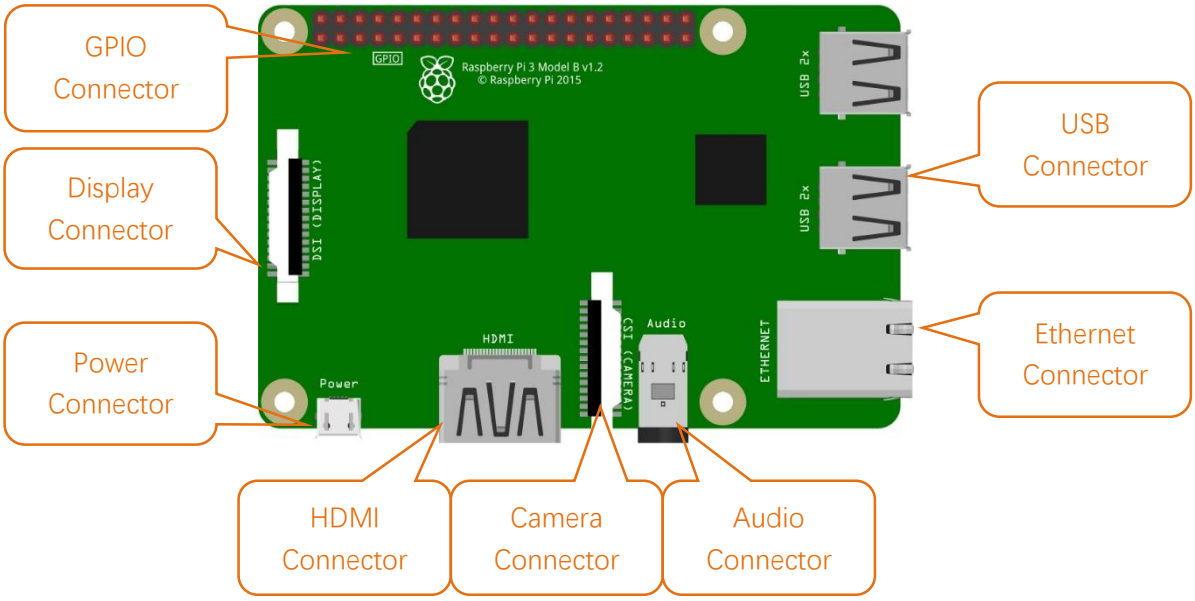
Practicality picture of Raspberry Pi 3 Model A+:	Model diagram of Raspberry Pi 3 Model A+:
	
Practicality picture of Raspberry Pi 1 Model A+:	Model diagram of Raspberry Pi 1 Model A+:
	

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: 

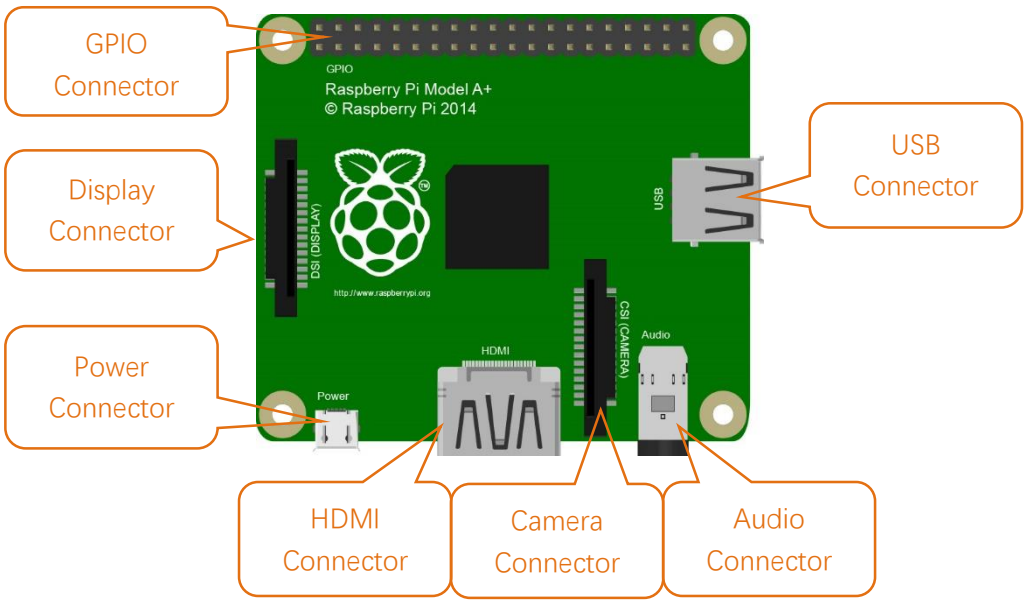
Hardware interface diagram of RPi 4B is shown below:



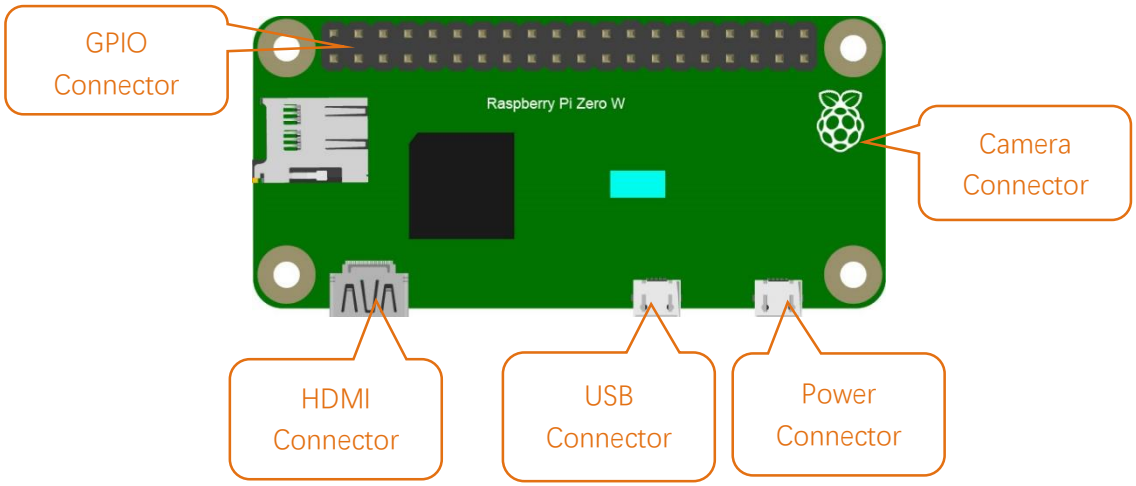
Hardware interface diagram of RPi 3B+/3B/2B/1B+ are shown below:



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



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

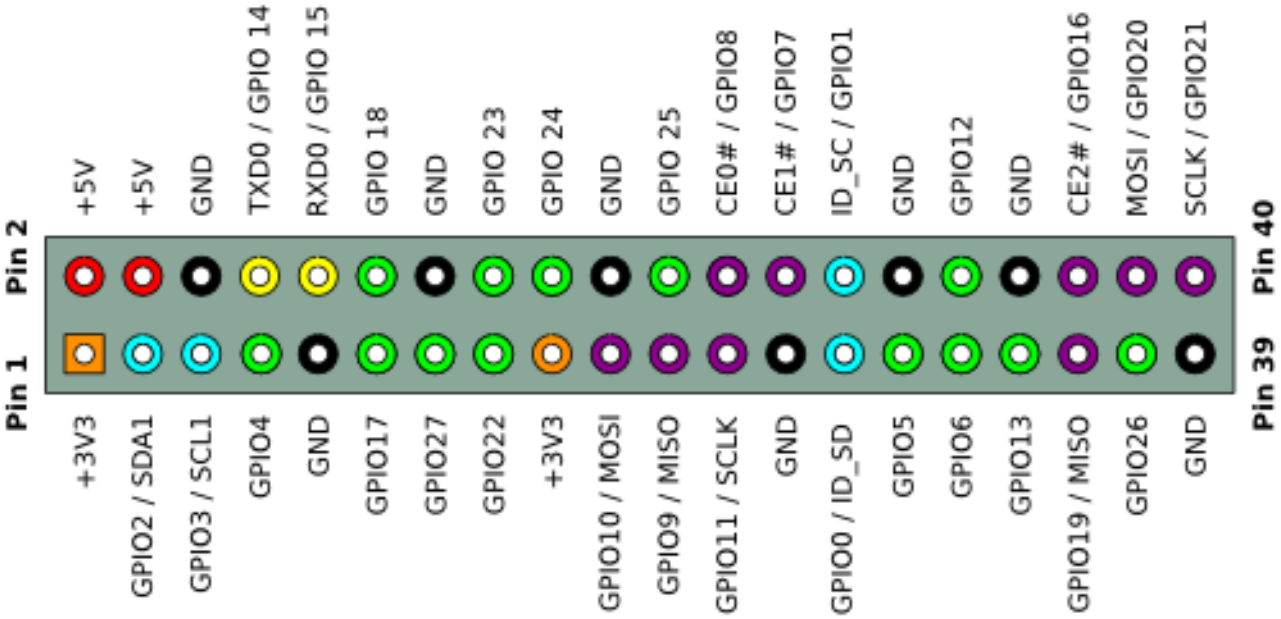


GPIO

GPIO: General purpose input/output. We will introduce the specific feature of the pins on the Raspberry Pi and what you can do with them. You can use them for all sorts of purposes. Most of them can be used as either inputs or outputs, depending on your program.
When programming the GPIO pins there are 3 different ways to refer to them: GPIO numbering, physical numbering, WiringPi GPIO Numbering.

BCM GPIO Numbering

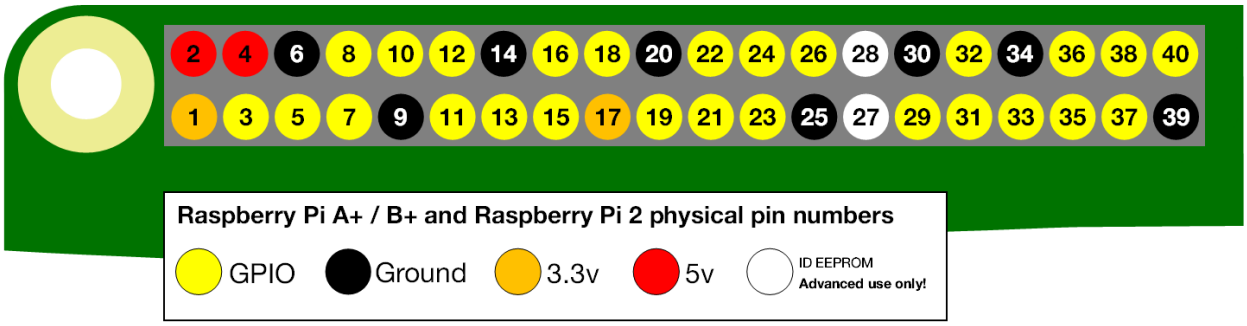
Raspberry Pi CPU use BCM2835/BCM2836/BCM2837of Broadcom. GPIO pin number is set by chip manufacturer. These are the GPIO pins as that computer recognizes. The numbers are unordered and don't make any sense to humans. You will need a printed reference or a reference board that fits over the pins.
Each 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 mentioned two kinds of GPIO serial numbers, RPi GPIO serial number of the WiringPi was renumbered. Here we have three kinds of GPIO number mode: based on the number of BCM chip, based on the physical sequence number and based on wiringPi. The correspondence between these three GPIO numbers is shown below:

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/)

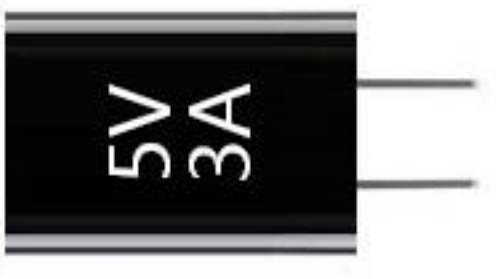
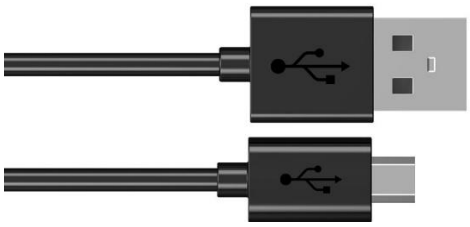
Chapter 0 Raspberry Pi Preparation

Install the 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 requirement of different versions of Raspberry Pi is 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 a network cable used to connect it to wide area network.

All of these components are necessary. Among them, the power supply is required at least 5V/2.5A, because lack of power supply will lead to many abnormal problems, even damage to your RPi. So power supply with 5V/2.5A is highly recommend. SD Card Micro (recommended capacity 16GB or more) is a hard drive for RPi, which is used to store the system and personal files. In later projects, the components list with a RPi will contains 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: using independent monitor, or remote desktop to share a monitor with your PC.

Required Accessories for Monitor

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

1.Display with HDMI interface

2.Mouse and Keyboard with USB interface

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

1. Mini-HDMI to HDMI converter wire.
2. Micro-USB to USB-A Receptacles converter wire (Micro USB OTG wire).
3. USB HUB.
4. USB transferring to Ethernet interface or USB Wi-Fi receiver.

For different Raspberry Pi, the optional items are slightly different. But all of their aims are to convert the special interface to standard interface of standard Raspberry Pi.

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
Mini-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 don't have an independent monitor, or you want to use a remote desktop, first you need to login to Raspberry Pi through SSH, then open the VNC or RDP service. So you need 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			

Raspbian System

Official Method

It is recommended to use this method.

You can follow the official method to install the system for raspberry pi

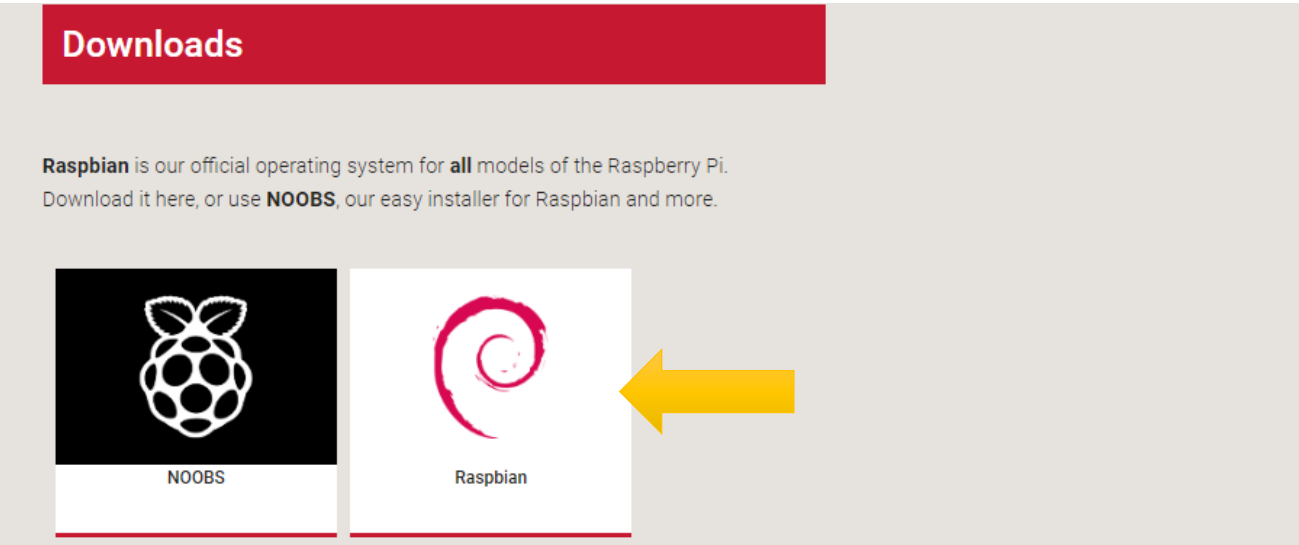
https://projects.raspberrypi.org/en/projects/raspberry-pi-setting-up/2

In this way, the system will be download **automatically** via the application.

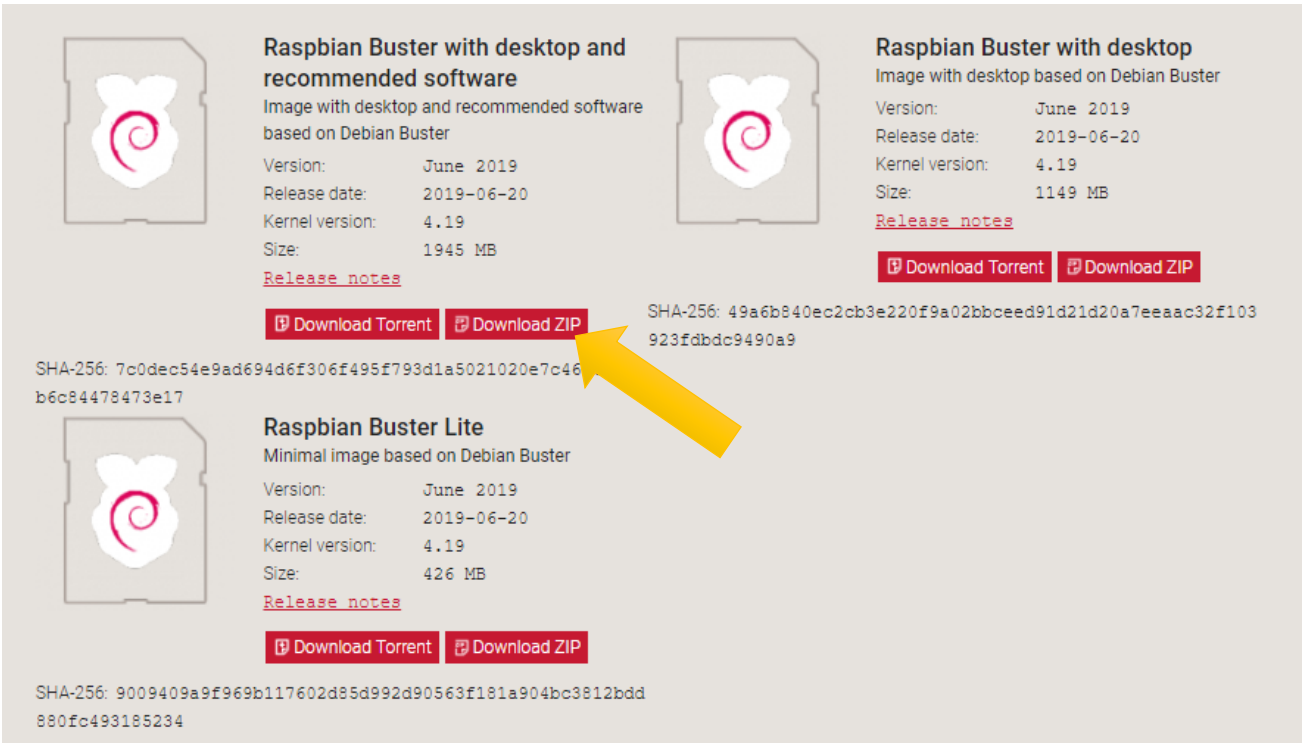
Download system manually (optional)

After installing the Imagage Tool in **link above**. You can also download the system **manually** first.

Visit RPi official website (https://www.RaspberryPi.org/), click "Downloads" and choose to download "RASPBIAN". RASPBIAN supported by RPI is an operating system based on Linux, which contains a number of contents required for RPi. We recommended RASPBIAN system to beginners. All projects in this tutorial are operated under the RASPBIAN system.



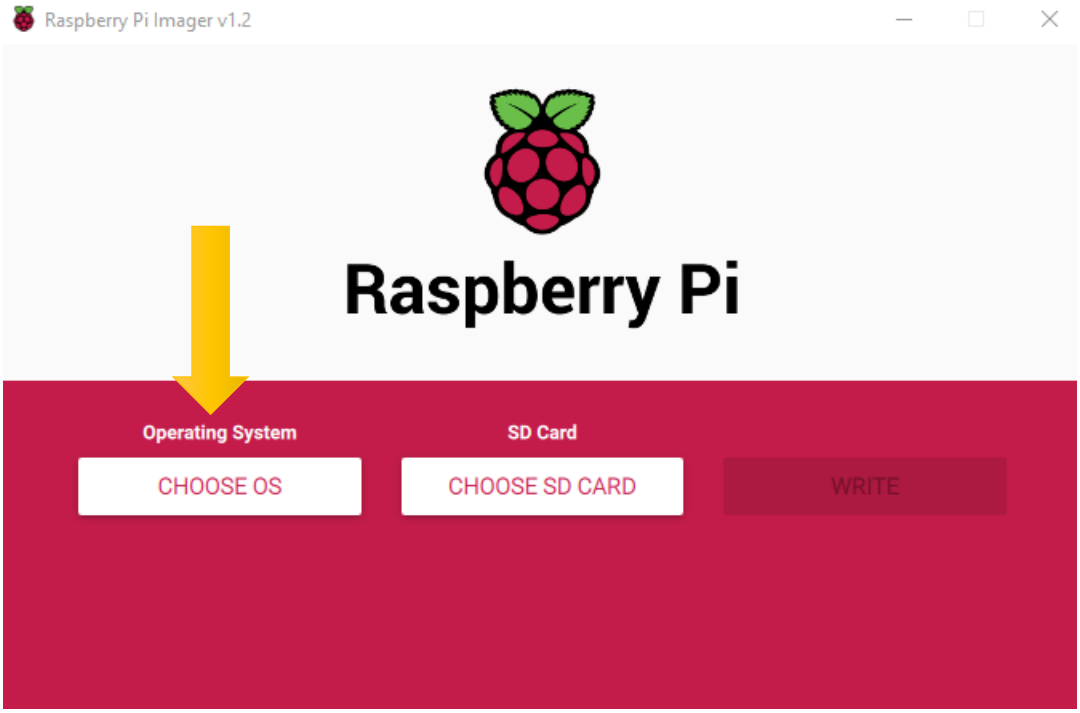
https://www.raspberrypi.org/downloads/raspbian/

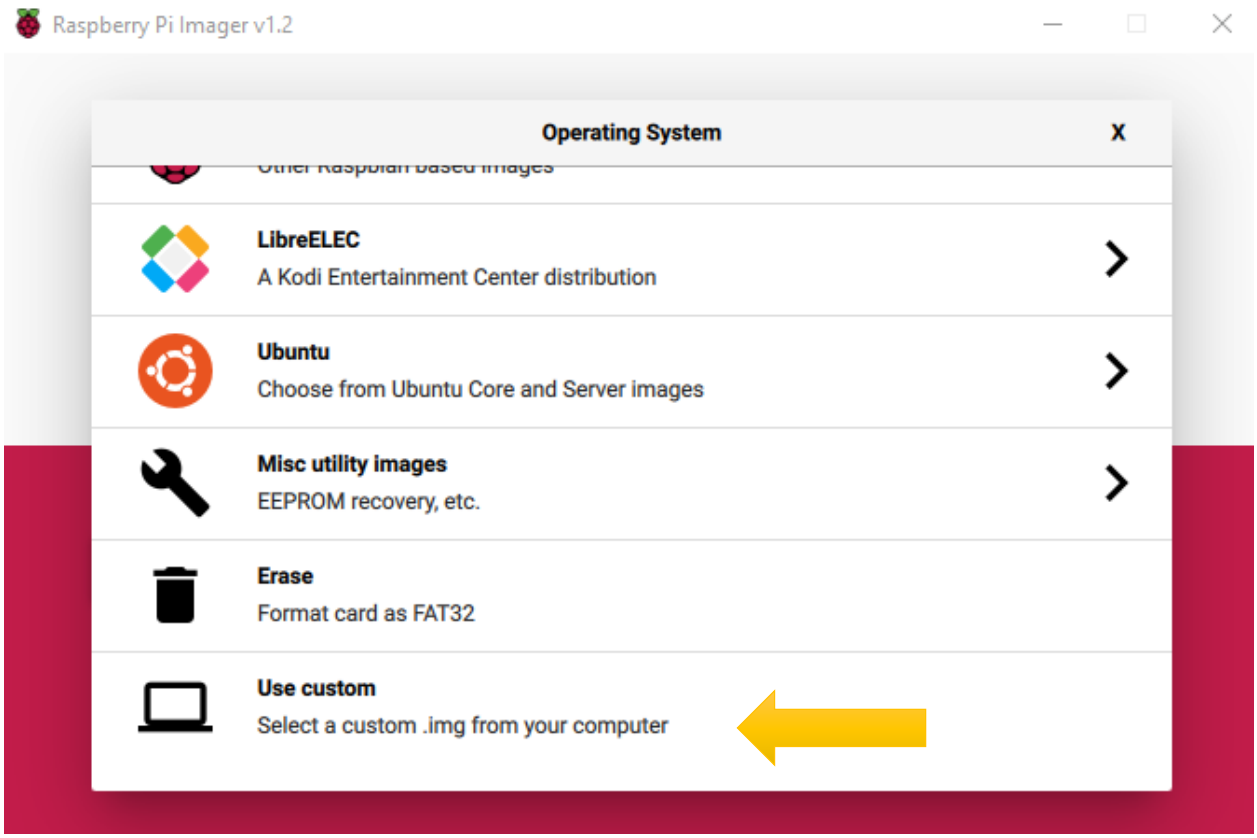


After the zip file is download.

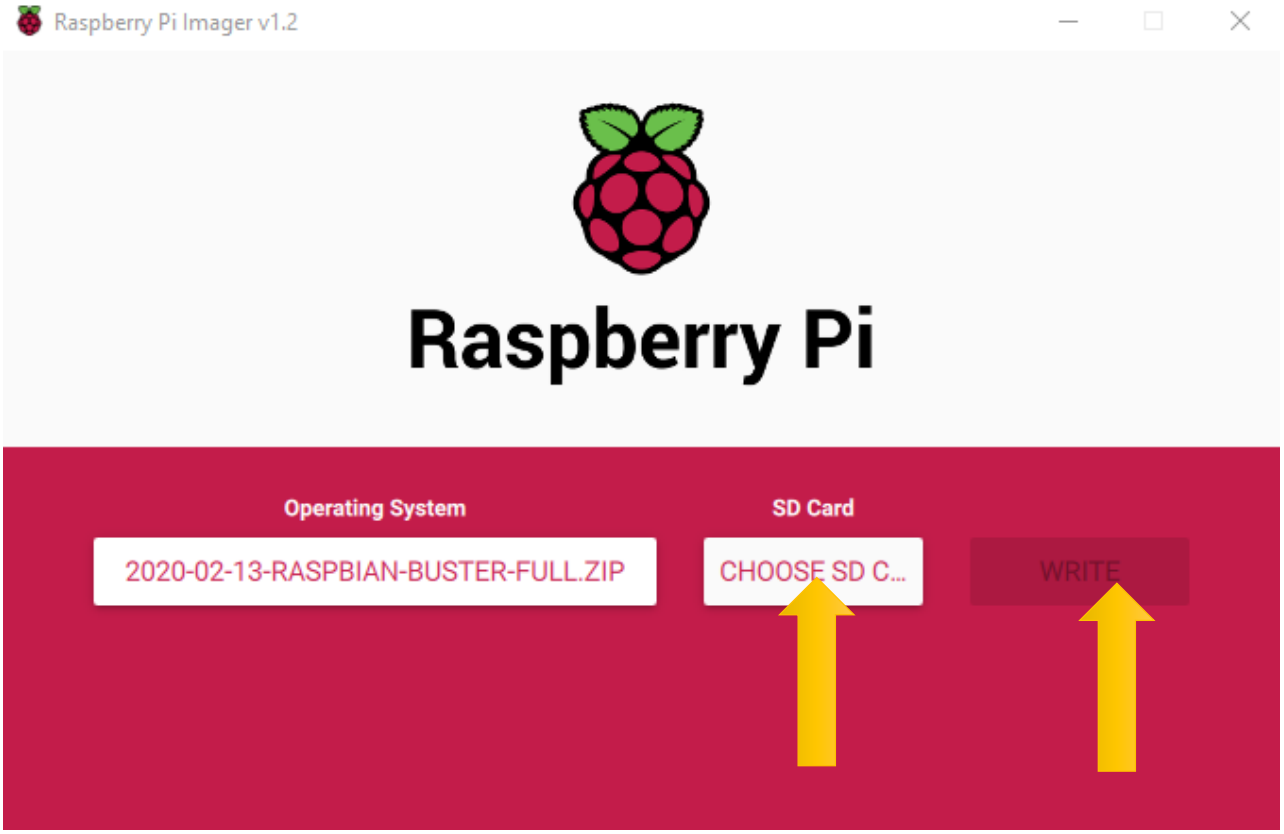
Write System to Micro SD Card

First, put your Micro **SD card** into card reader and connect it to USB port of PC. Then open imager toll, choose Choose system that you just download in Use custom.





Choose the SD card. Then click “WRITE”.



Start Raspberry Pi

If you don't have a spare monitor, please jumper to next section. If you have a spare monitor, please follow steps in this section.

After the system is written successfully, take out Micro SD Card and put it into the card slot of RPi. Then connect RPi to screen through the HDMI, to mouse and keyboard through the USB port, to network cable through the network card interface and to the power supply. Then your RPi starts initially. Later, you need to enter the user name and password to login. The default user name: pi; password: raspberry. Enter and login. After login, you can enter the following interface.

Now, you have successfully installed the RASPBIAN operating system for your RPi.

Then you can connect WiFi on the right corner.

Now you can jumper to VNC Viewer.

Remote desktop & VNC

If you don't have a spare display, 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 remote desktop under the Windows operating system to control RPi.

Under windows, Raspberry Pi can be generally accessed remotely through two applications. The first one is the windows built-in application remote desktop, which corresponds to the Raspberry Pi xrdp service. The second one is the free application VNC Viewer, which corresponds to the VNC interface of Raspberry Pi. Each way has its own advantages. You can choose either one or two.

Windows	Raspberry Pi
Remote Desktop Connection	Xrdp
VNC Viewer	VNC

VNC Viewer can not only run under Windows, but also under system MAC, Linux, IOS, Android and so on.
Some remote connection tools like Xrdp, it does not support opencv and pyqt window display. So it is recommended to use VNC Viewer to connect Raspberry Pi for this robot.

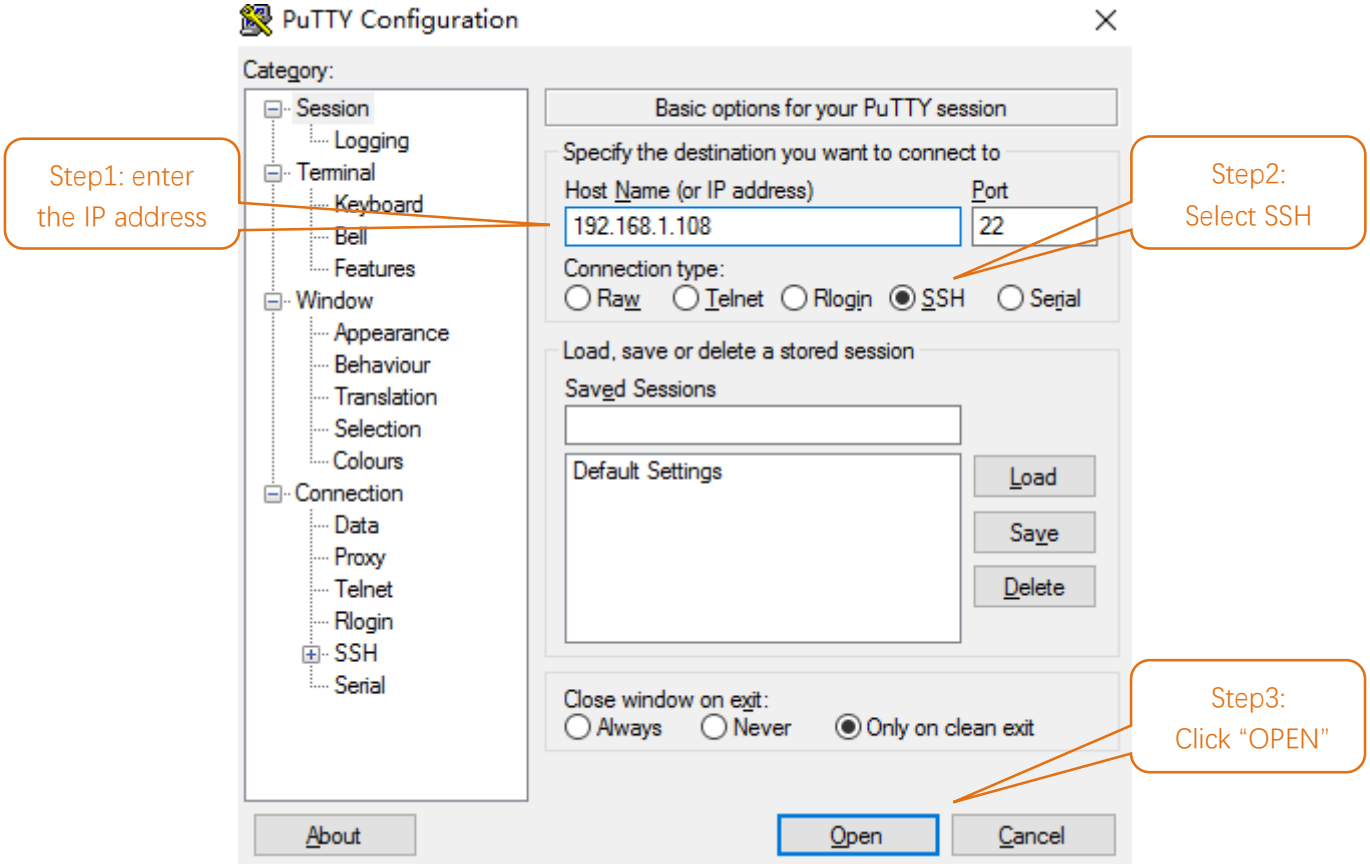
SSH

Under previous Raspbian system, SSH is opened by default. Under the latest version of Raspbian system, it is closed by default. So you need to open it first.

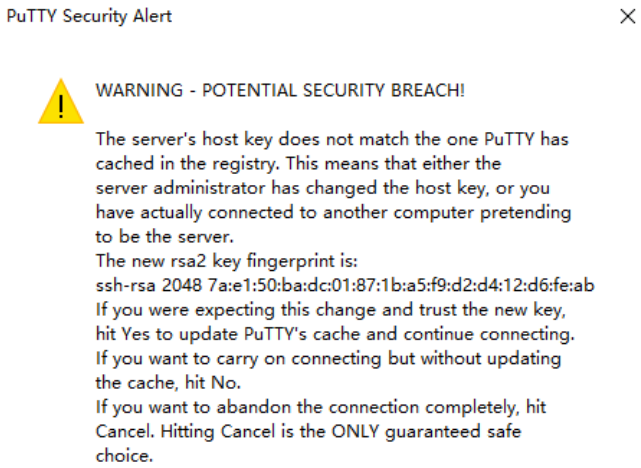
Method: after the system is written. Create a folder named “ssh” under generated boot disk, then the SSH connection will be opened.

And then, download the tool software Putty. Its official address: http://www.putty.org/
Or download it here: http://www.chiark.greenend.org.uk/~sgtatham/putty/download.html

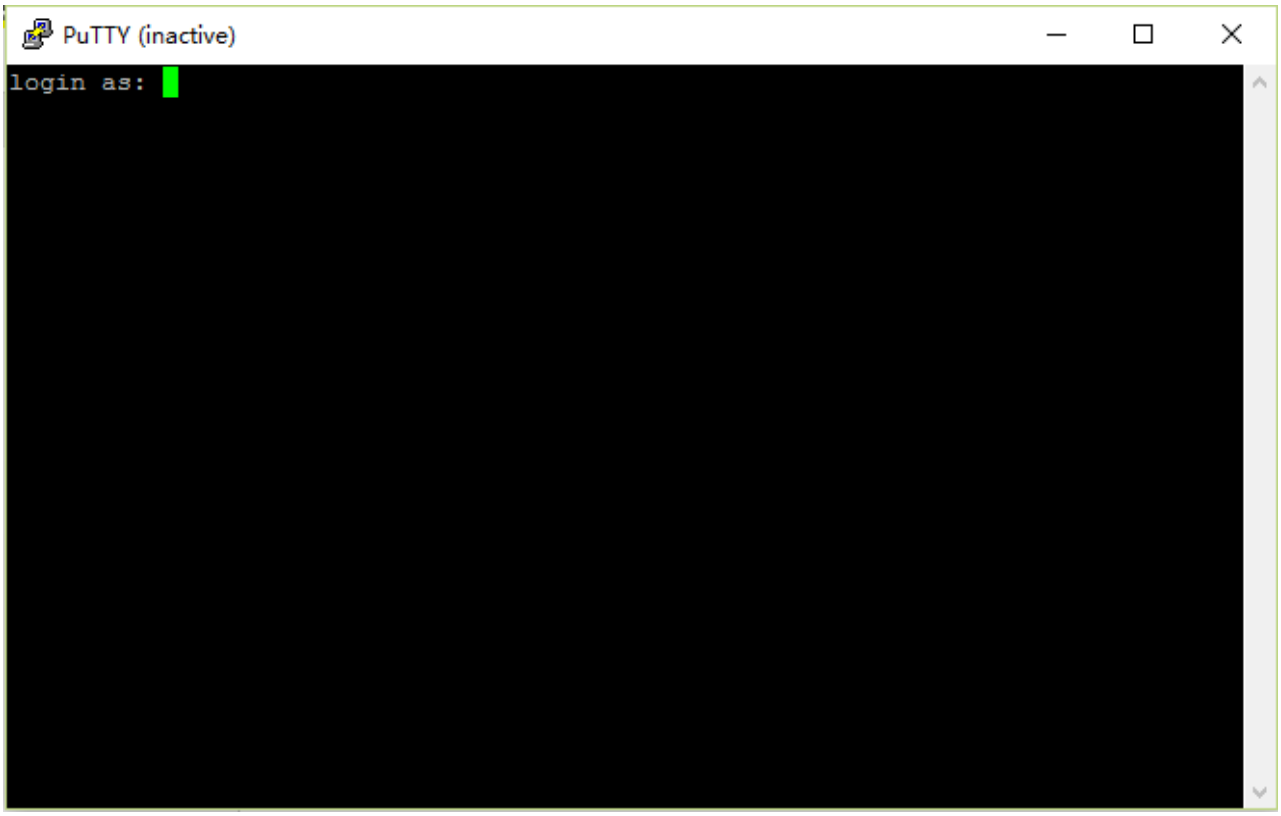
Then use cable to connect your RPi to the routers of your PC LAN, to ensure your PC and your RPi in the same LAN. Then put the system Micro SD Card prepared before into the slot of the RPi and turn on the power supply waiting for starting RPi. Later, enter control terminal of the router to inquiry IP address named “raspberry pi”. For example, I have inquired to my RPi IP address, and it is “192.168.1.108". Then open Putty, enter the address, select SSH, and then click "OPEN", as shown below:



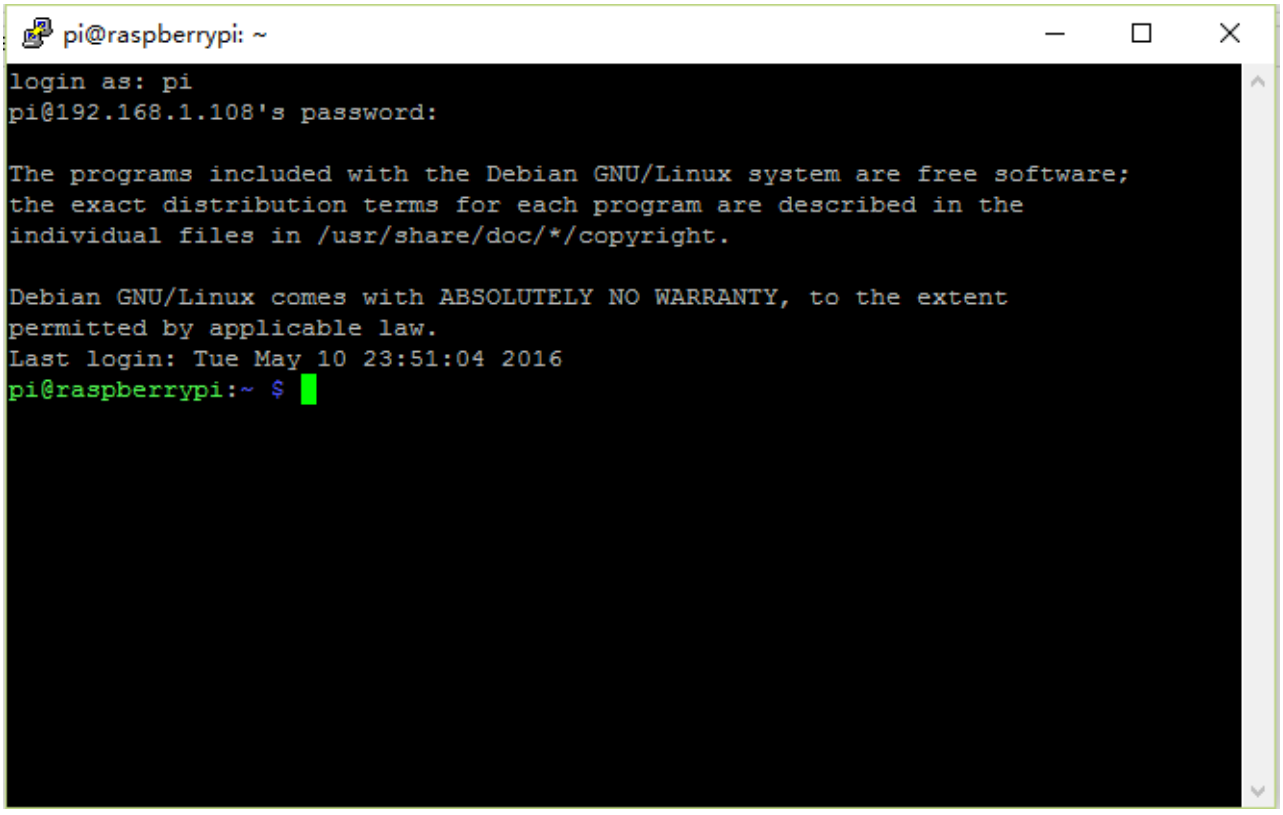
There will appear a security warning at first login. Just click “YES”.



Then there will be a login interface (RPi default user name: **pi**; the password: **raspberry**). When you enter the password, there will be **no display** on the screen. This is normal. After the correct input, press "Enter" to confirm.



Then enter the command line of RPi, which means that you have successfully login to RPi command line mode.



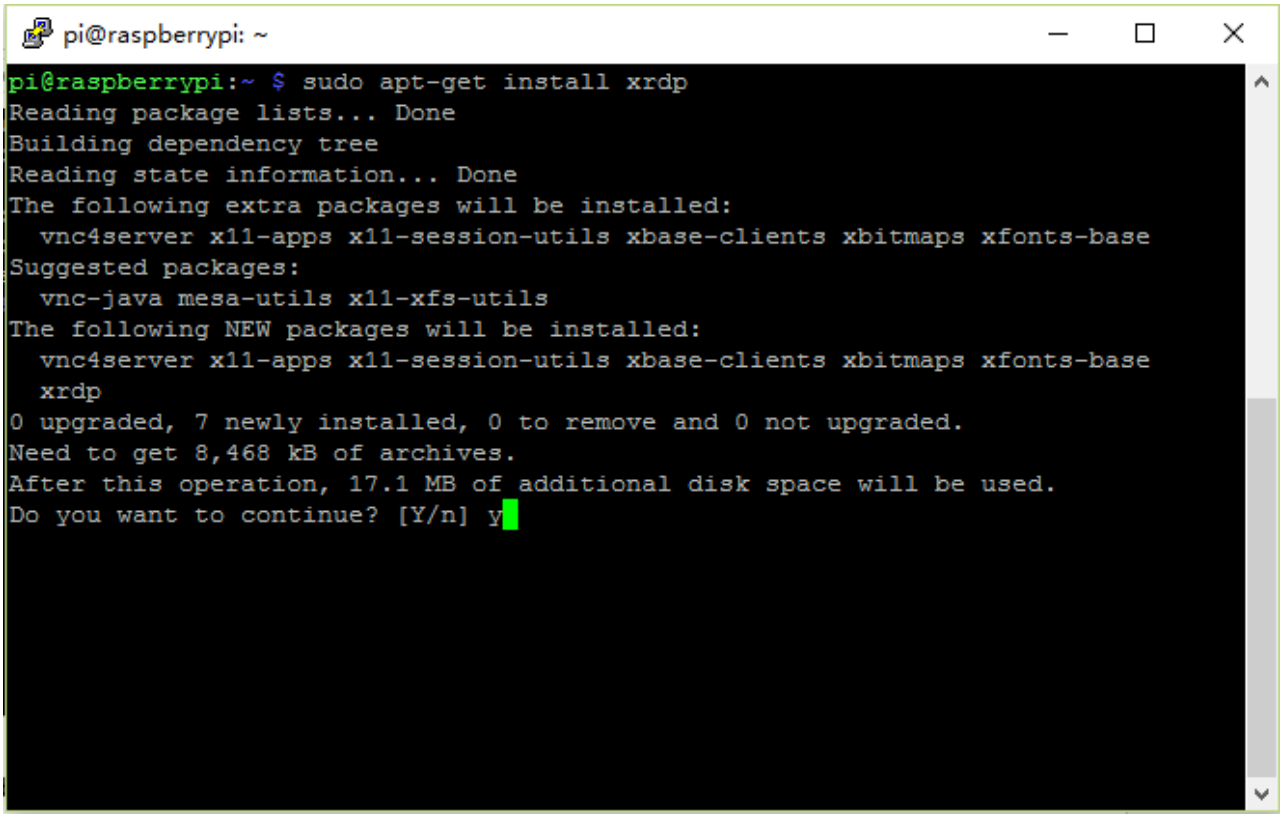
Remote Desktop Connection & xrdp

If you want to use built-in Remote Desktop Connection under Windows, you need install xrdp service on Raspberry Pi.

Next, install a xrdp service, an open source remote desktop protocol(xrdp) server, for RPi. Type the following command, then press enter to confirm:

```
sudo apt-get install xrdp
```

Later, the installation starts.

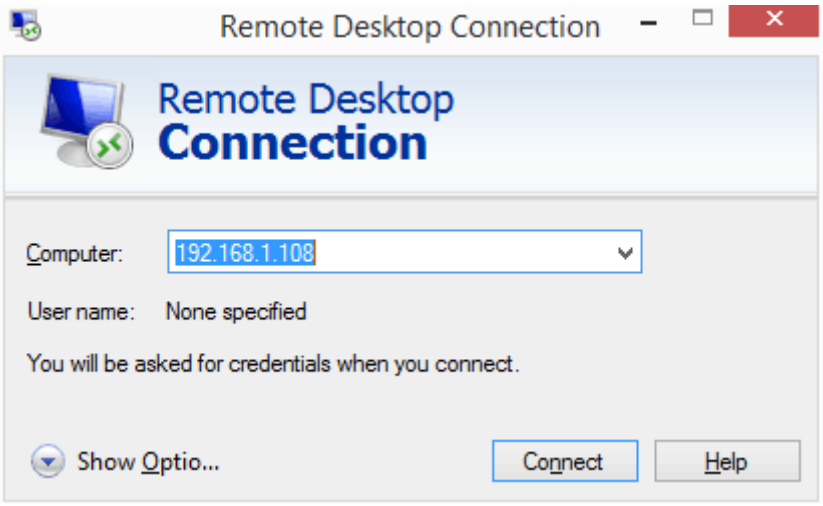


Enter "Y", press key “Enter” to confirm.

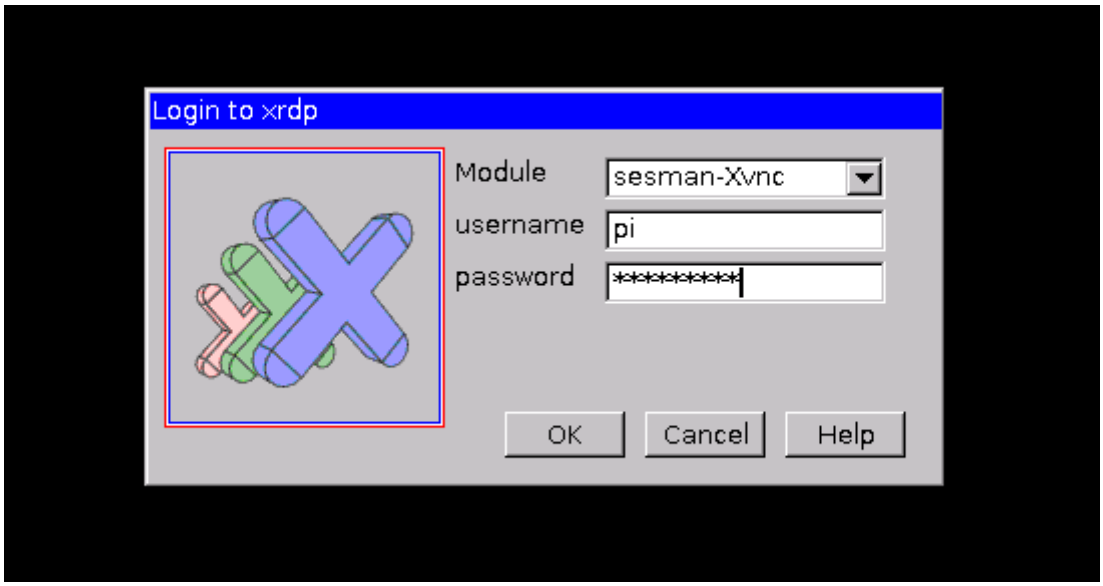
After the installation is completed, you can use Windows remote desktop applications to login to your RPi.

Login to Windows remote desktop

Use "WIN+R" or search function, open the remote desktop application "mstsc.exe" under Windows, enter the IP address of RPi and then click “Connect”.



Later, there will be xrdp login screen. Enter the user name and password of RPi (RPi default user name: pi; password: raspberry) and click “OK”.



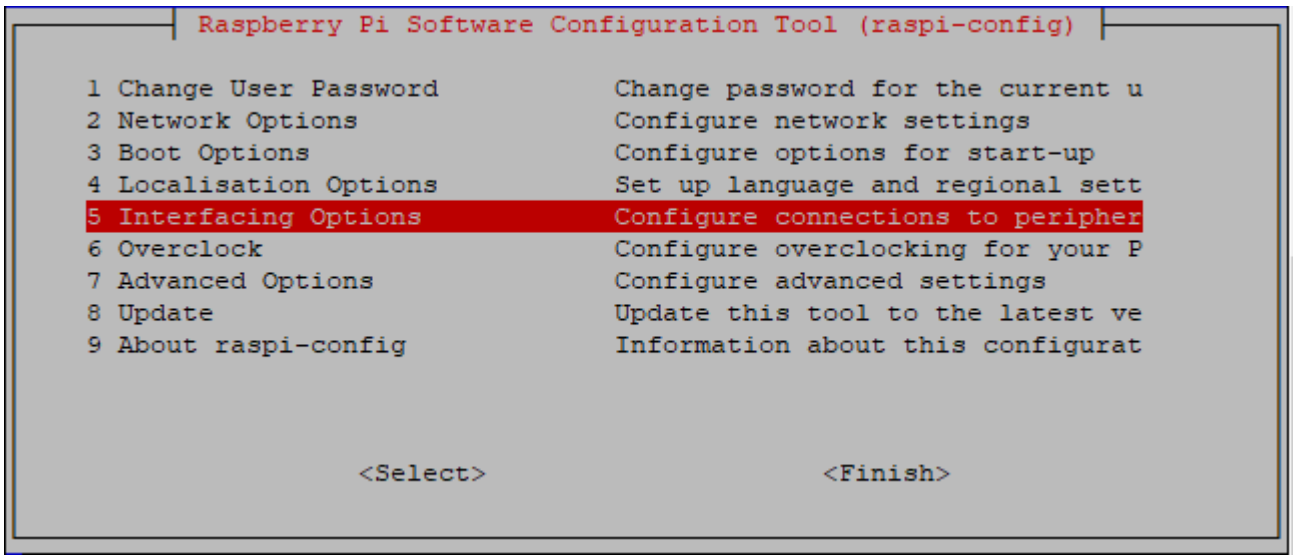
Later, you can enter the RPi desktop system.

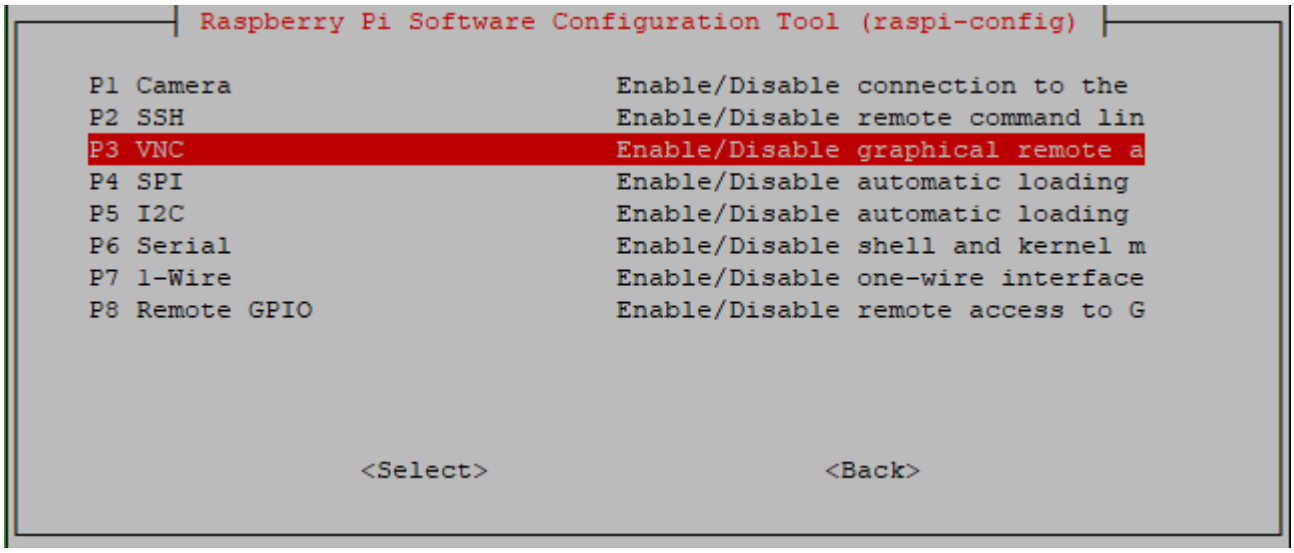
Here, you have successfully used the remote desktop login to RPi.

VNC Viewer & VNC

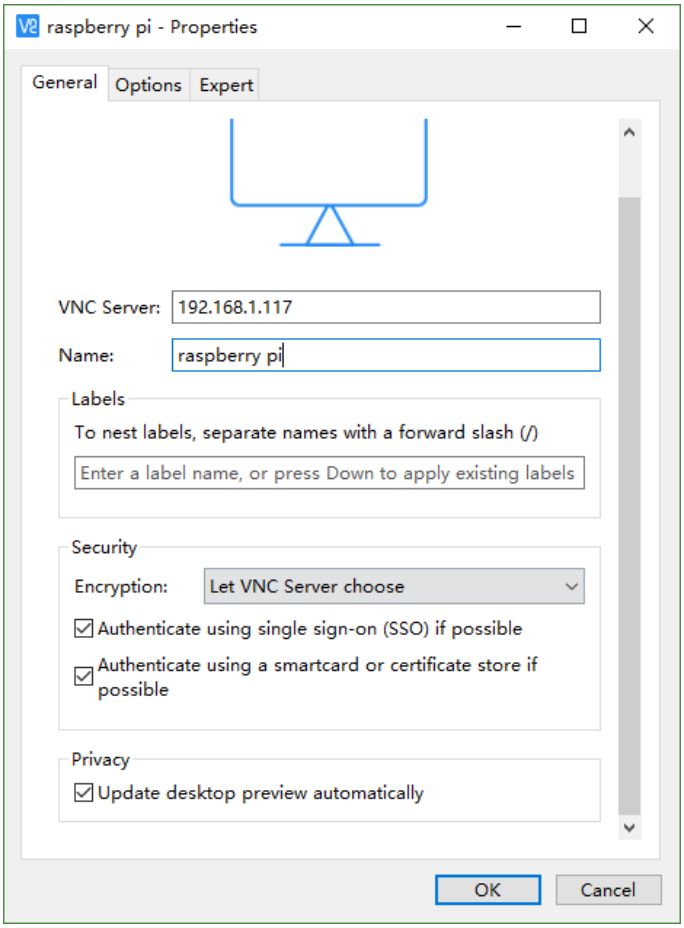
Type the following command. And select 5 Interfacing Options→P3 VNC →Yes→OK→Finish. Here Raspberry Pi may need be restarted, and choose ok. Then open VNC interface.

```
sudo raspi-config
```

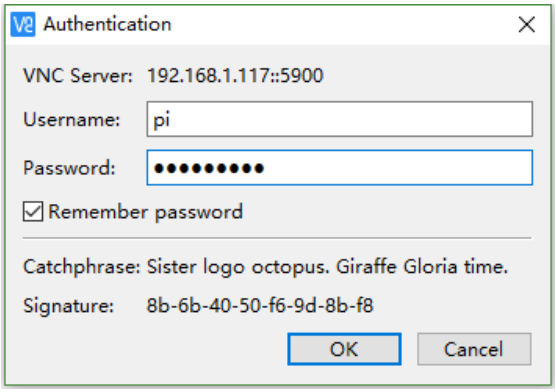




Then download and install VNC Viewer by click following link:
https://www.realvnc.com/en/connect/download/viewer/windows/
After installation is completed, open VNC Viewer. And click File → New Connection. Then the interface is shown below.



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.



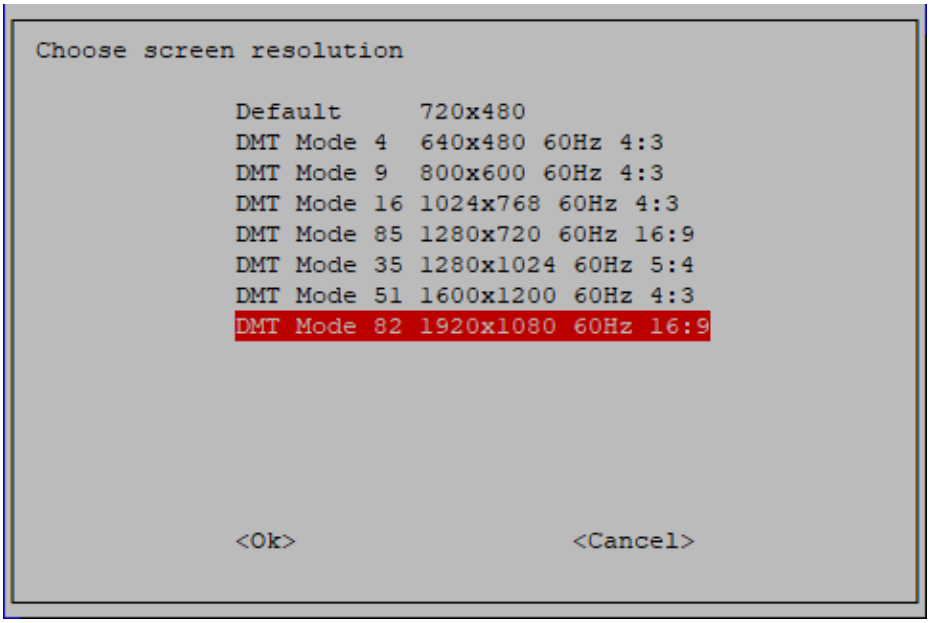
Enter username: **pi** and Password: **raspberry**. And click OK.

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

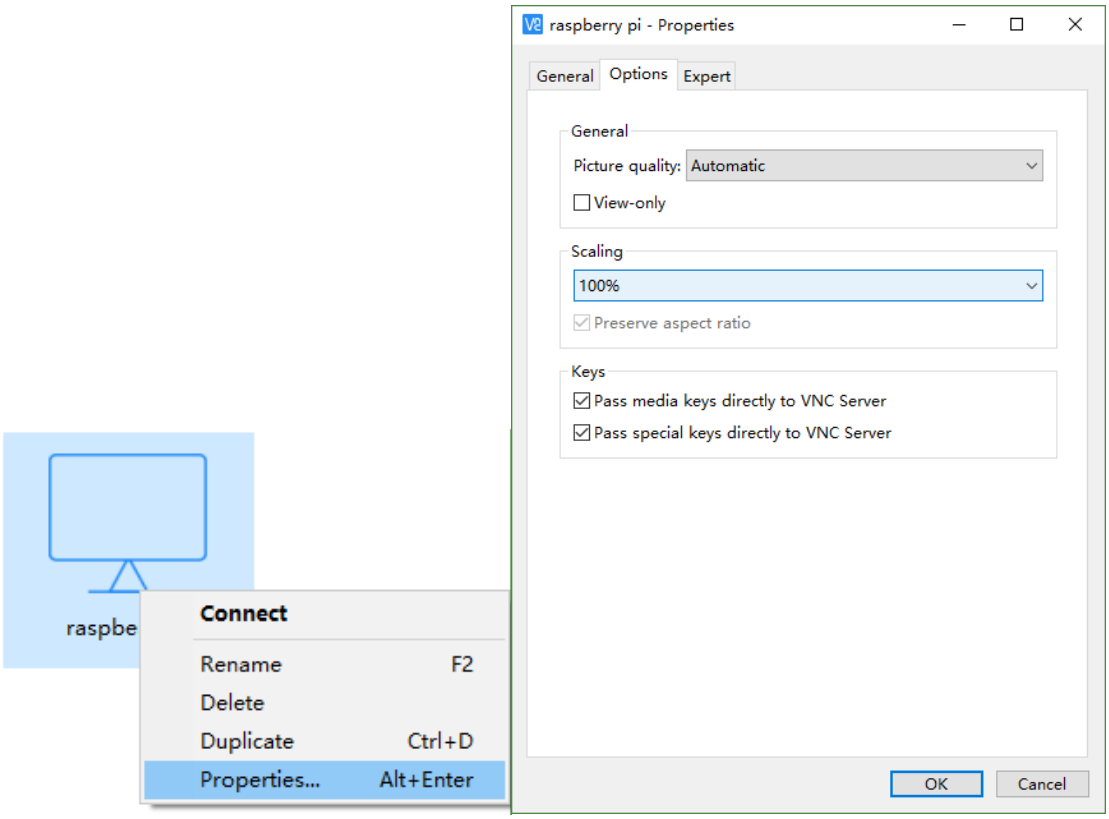
If the resolution ratio is not great or there is just a **black little window**, you can set a proper resolution ratio via steps below.

```
sudo raspi-config
```

Select 7 Advanced Options→A5 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.



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

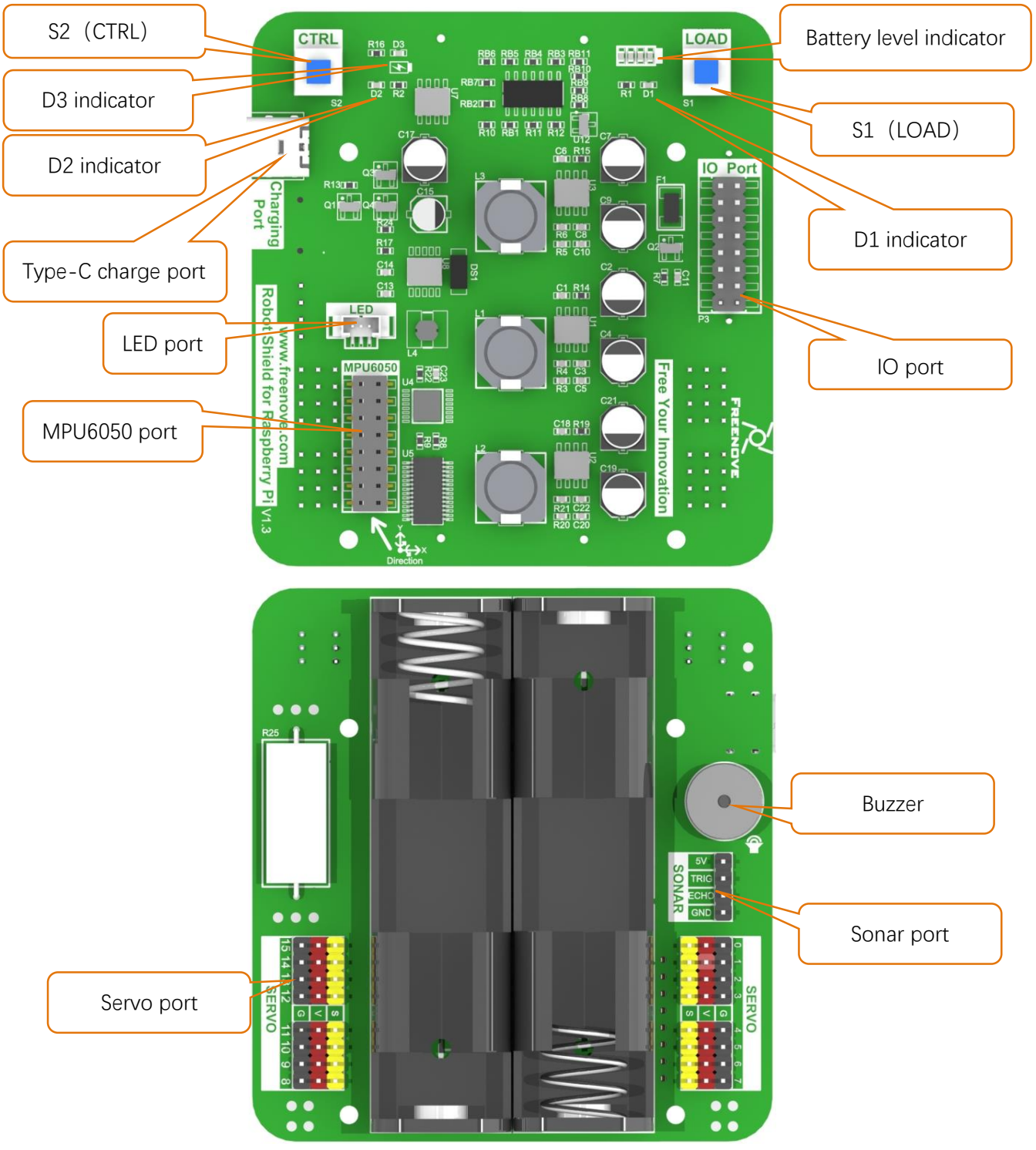
Wi-Fi

Raspberry Pi 4B/3B+/3B integrates a Wi-Fi adaptor. You can use it to connect to your Wi-Fi. Then you can use the wireless remote desktop to control your RPi. This will be helpful for the following work. Raspberry Pi of other models can use wireless remote desktop through accessing an external USB wireless card.

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 need two 18650 3.7v batteries. It has charger function.
- **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.
- **Type-C port**: It is used to connect cable for charging. It only works when S1 and S2 are turned off.
- **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 be gradually 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.

Charging Function (necessary for using charging function)

Description

1 The control board has a complete battery charging protection function and uses a complete CC/CV charging mode. Two 18650 batteries can be charged through the TYPE-C port, but **not a single battery**.

2 When voltage of the two batteries is less than 2V, they are charged with 50mA. When voltage of two batteries is greater than 2V and less than 5.6V, they are charged with 100mA. When voltage of two batteries is larger than 5.6V, they are charged with 1A.

3 When the batteries are nearly full, they will be charged at constant voltage mode. At this time, the current will be small and the charging will be slow.

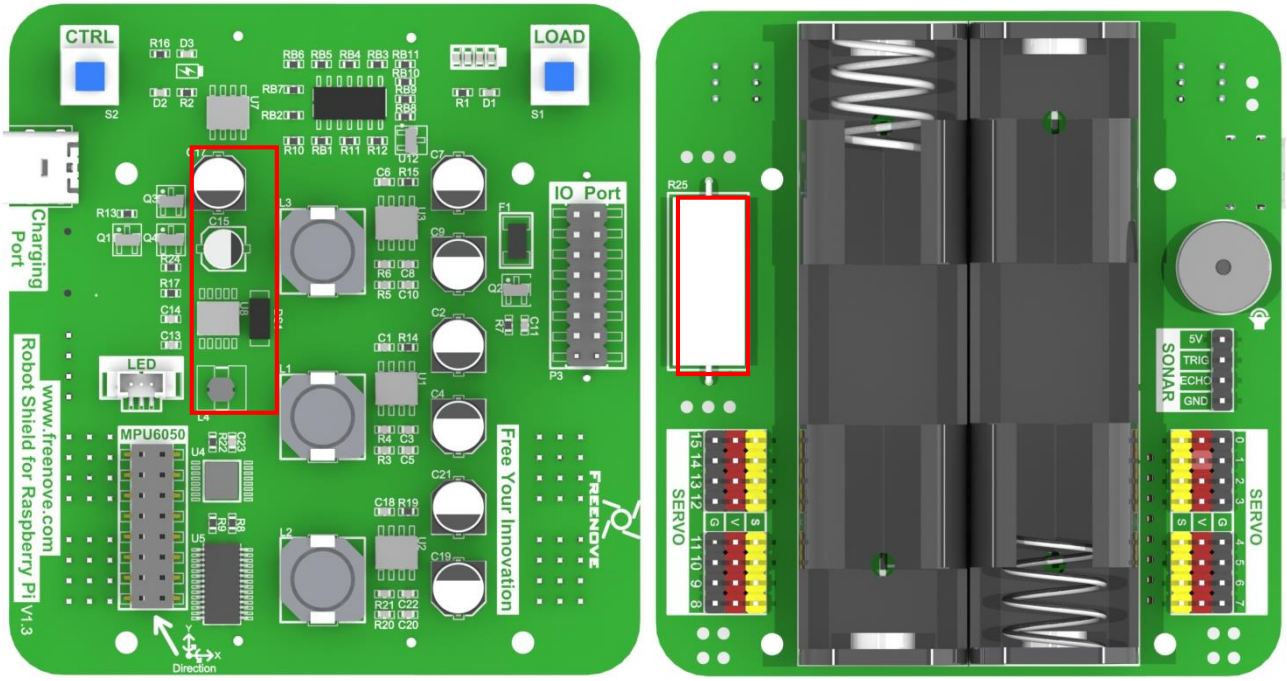
4 When voltage of each battery is greater than 3.6V and the sum voltage of the two batteries is greater than 7.2V, and the voltage drop is greater than 0.06V, the balanced charging function will be activated, and the maximum balanced current is about 370mA. If it is detected that the voltage difference between the two batteries is greater than 0.5V, it is judged that the batteries are damaged and the battery charging will stop.

How to use the charging function

1. To use the charging function, first **turn off the two switches S1 and S2** (D1, D2 and the power indicator are all off).
2. Insert the TYPE-C data cable into the TYPE-C port. Please use an adapter of 5V 2A or more. If there is no battery in the battery box, and the D3 charging indicator will blink continuously. Put **two batteries** into the battery holder (single one doesn't work). If they are already there, just continue reading. The D3 indicator will stop blinking and stay on (charging), and will turn off when the batteries are fully charged.
3. When there are no power indicator leds on when using, it is time to charge. Because when the voltage of two batteries are less than 5.6V. The charge current is very small. Therefore, we recommend to charging the battery beyond 3V.
4. It will take too long to get the batteries full charged with D3 off. You can turn on S2 (no need to disconnect charge cable) to check the power indicator after charge some hours. If there are less than 3 LED on, you can turn off S2. The charging will continue.
5. If the battery contact is abnormal, the output is short-circuited, the charging time is over, or the battery temperature is abnormal, the D3 indicator will flash and charging will stop.

Note:

In the process of charging with 1A current, the two positions of the shield will be hot, which is normal to use. Please avoid touching them with your hands.



Chapter 1 Install Python Libraries (Necessary)

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

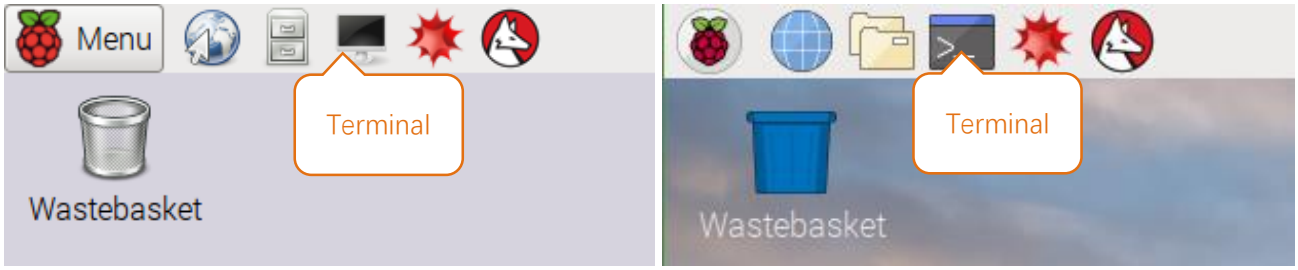
In this chapter, we will do some necessary preparations, start the Raspberry Pi and install some necessary python libraries. The next chapter will assemble the robot dog.

Note: The library installation needs a lot of time. You can use the power cable to power the Raspberry Pi (no battery power required).

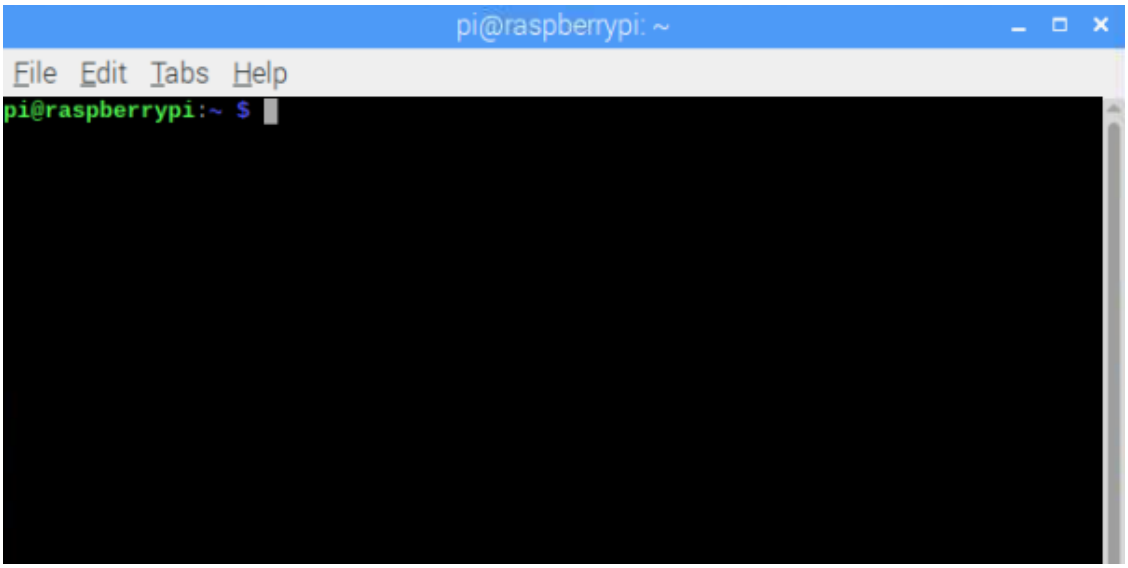
It is recommended to use **VNC Viewer** to build this robot when you use R**emote Desktop** to login RPi or there will be errors.

Step 1.1 Obtain the Code

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



The terminal is shown below:



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
```

```
pi@raspberrypi:~ $ cd ~
pi@raspberrypi:~ $ git clone https://github.com/Freenove/Freenove_Robot_Dog_Kit_for_Raspberry_Pi
```

Downloading need 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 code 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 default python on your raspberry Pi. Press Ctrl-Z to exit.

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

If it is python3, you can skip this section.
If it is python2, you need execute 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
```

```
pi@raspberrypi:~ $ cd /usr/bin
pi@raspberrypi:/usr/bin $ sudo rm python
pi@raspberrypi:/usr/bin $ sudo ln -s python3 python
pi@raspberrypi:/usr/bin $ python
Python 3.7.3 (default, Apr  3 2019, 05:39:12)
[GCC 8.2.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

If you want to set python2 as default python in **other projects**.
Just repeat command above 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
pi@raspberrypi:/usr/bin $ python
Python 2.7.16 (default, Apr  6 2019, 01:42:57)
[GCC 8.2.0] on linux2
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

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 "↑" will bring back the previous command, and pressing "↓" will bring back the latter command.

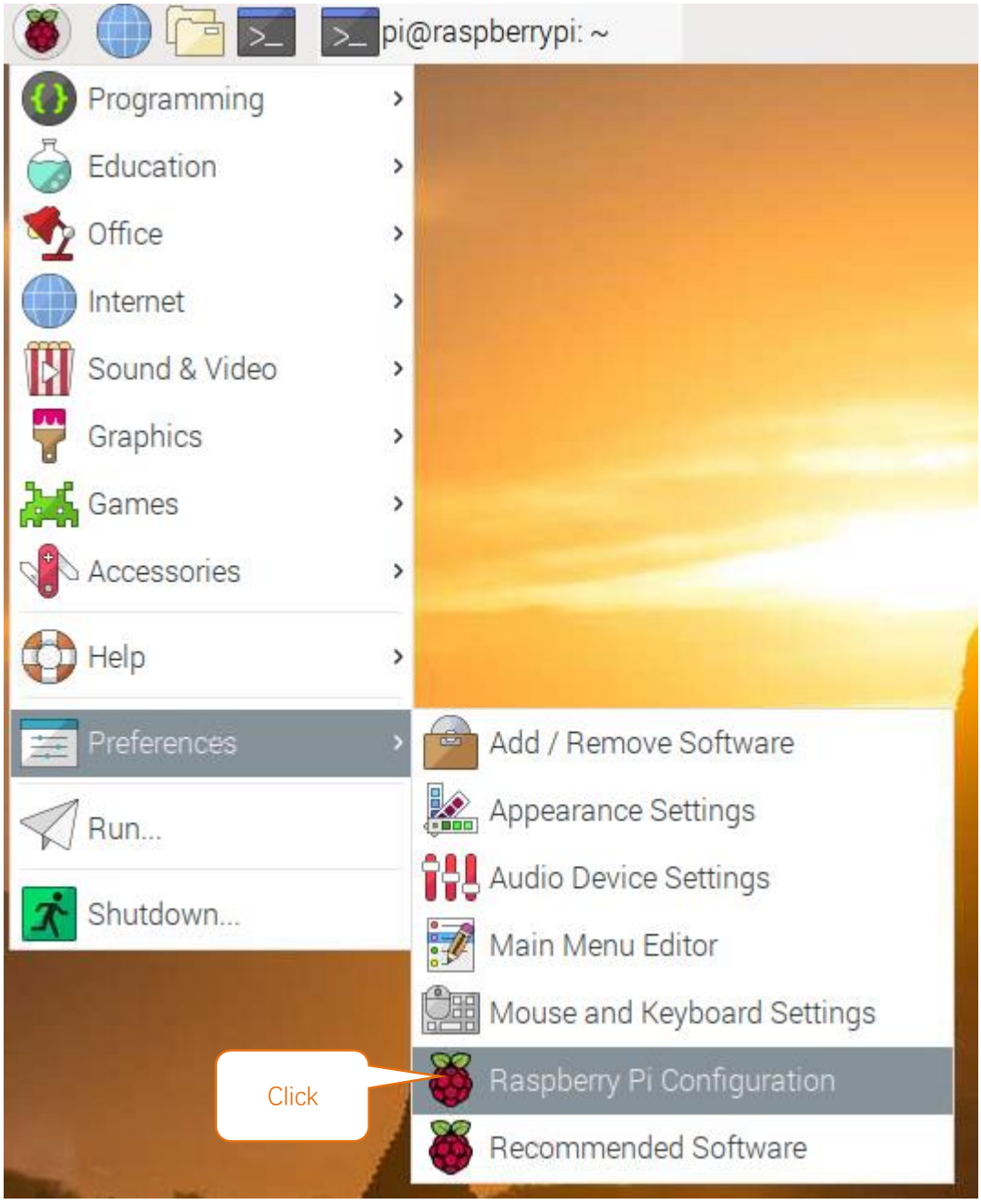
2. **Tab key**. The Tab key can automatically complete the command/path you want to type. When there are multiple commands/paths conforming to the already typed letter, pressing Tab key once won't have any result. And pressing Tab key again will list all the eligible options. This command/path will be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

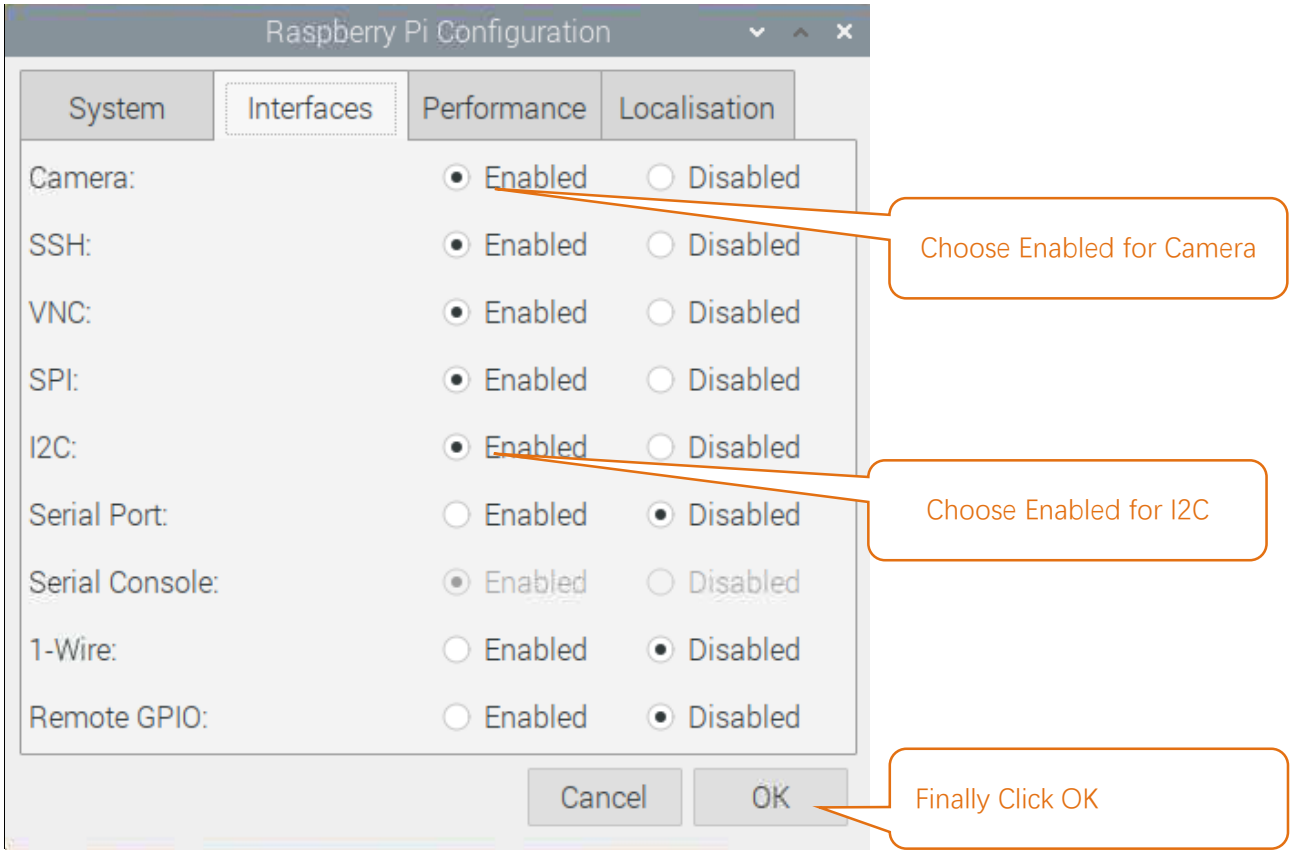
As shown below, under the '~'directory, enter the Documents directory with the "cd" command. After typing "cd D", press Tab key, then there is no response. Press Tab key again, then all the files/folders that begin with "D" is listed. Continue to type the character "oc", then press the Tab key, and then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

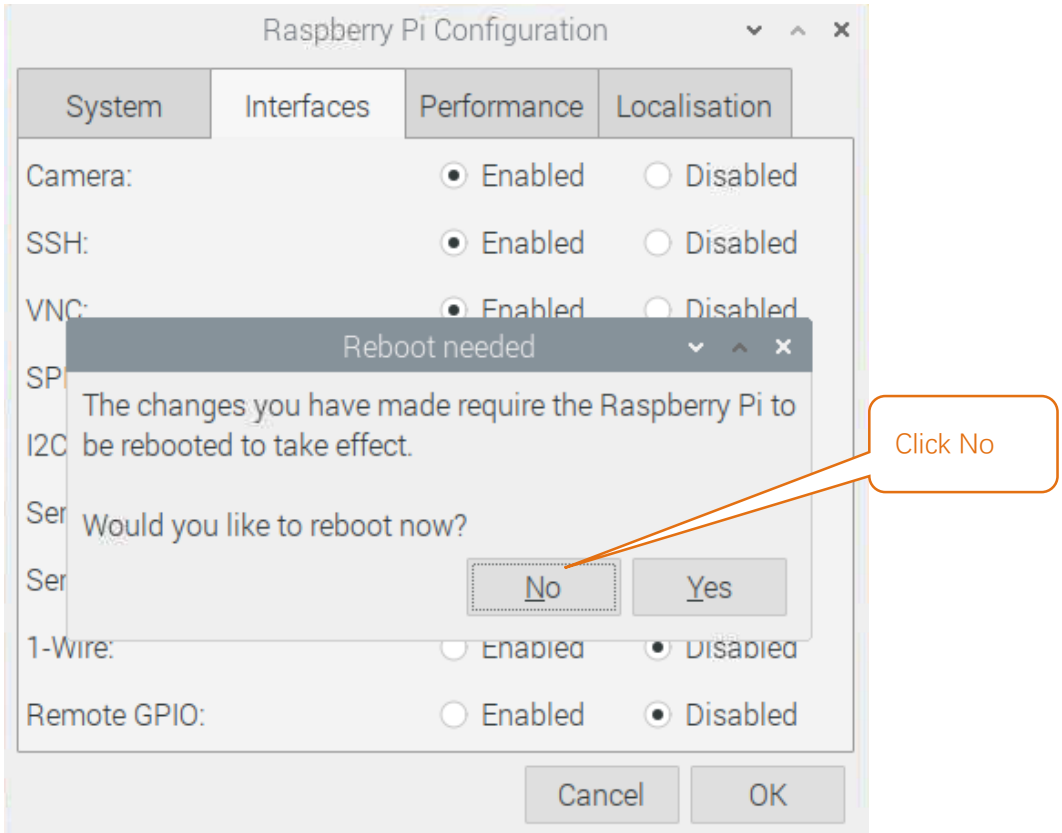
```
pi@raspberrypi:~ $ cd D
Desktop/   Documents/ Downloads/
pi@raspberrypi:~ $ cd Doc
```

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

Step 2 Enable I2C and Camera



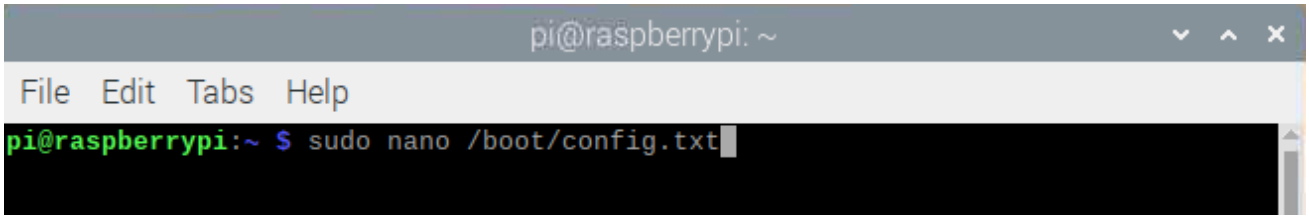

Raspberry Pi Configuration
System
Interfaces
Performance
Localisation
Camera: Enabled Disabled
SSH: Enabled Disabled
VNC: Enabled Disabled
SPI: Enabled Disabled
I2C: Enabled Disabled
Serial Port: Enabled Disabled
Serial Console: Enabled Disabled
1-Wire: Enabled Disabled
Remote GPIO: Enabled Disabled
Cancel
OK
Choose Enabled for Camera
Choose Enabled for I2C
Finally Click OK


Raspberry Pi Configuration
System
Interfaces
Performance
Localisation
Camera: Enabled Disabled
SSH: Enabled Disabled
Reboot needed
The changes you have made require the Raspberry Pi to be rebooted to take effect.
Would you like to reboot now?
No
Yes
Remote GPIO: Enabled Disabled
Cancel
OK
Click No

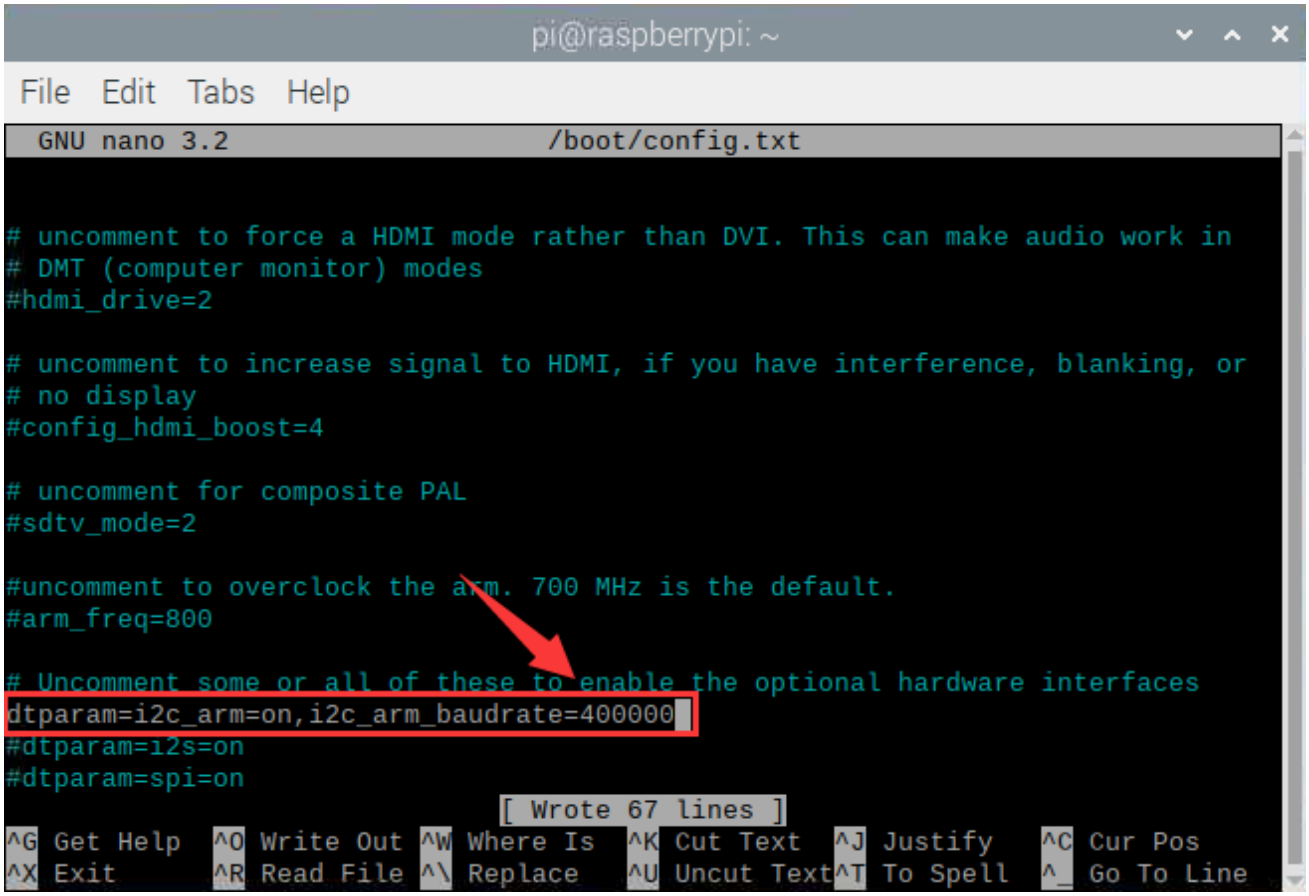
Set I2C Baud Rate

Open the terminal and enter 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. Scroll 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.

Step 3 Run the Installation Program

All the commands are based on python3. If the default python is python2, please refer to the Step1 to set python3 to default python.

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 completes, as shown below.

```
Now the installation is successful.

Please restart raspberry pi
```

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

```
sudo python setup.py
```

Chapter 2 Assemble Robot

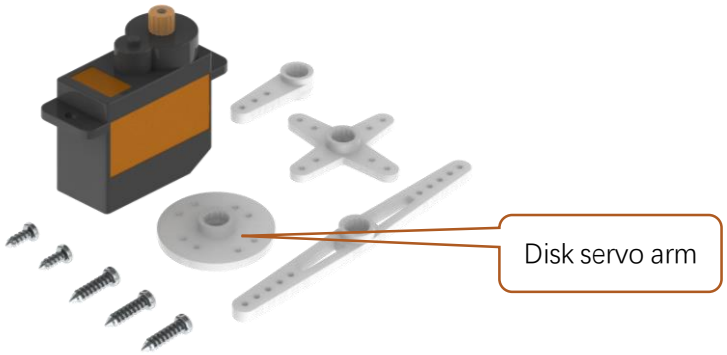
If there are any concerns, please feel free to contact us at support@freenove.com

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 driver the screws too tight in the assembly.

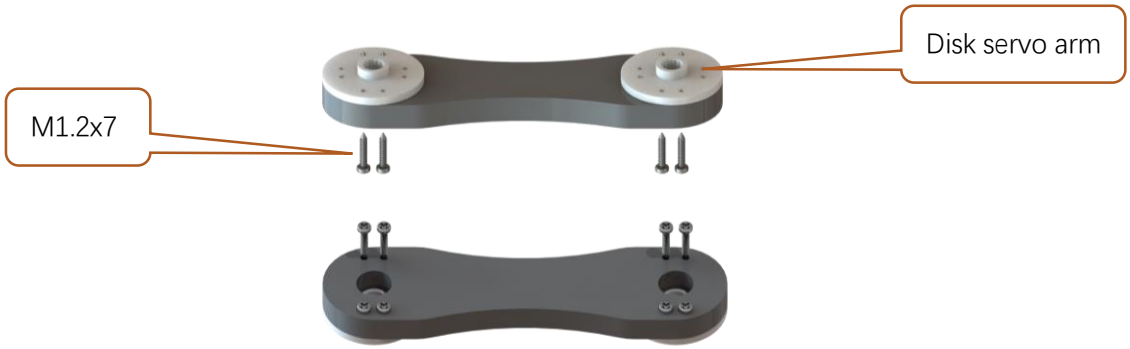
Step 1 Install Disk Servo Arms

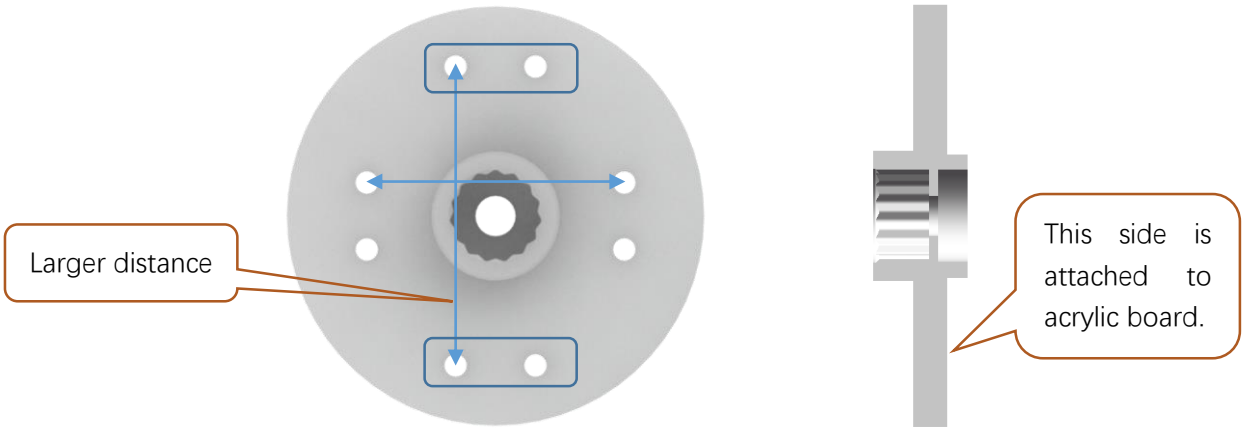
Take out 12 disk servo arms from the servo package.



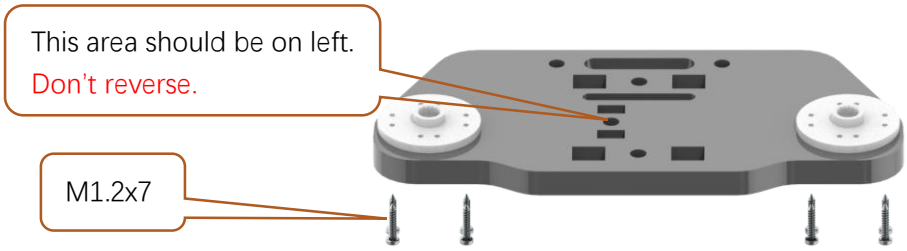
The distance between two holes are different.

Please use the holes with **larger distance** between them.

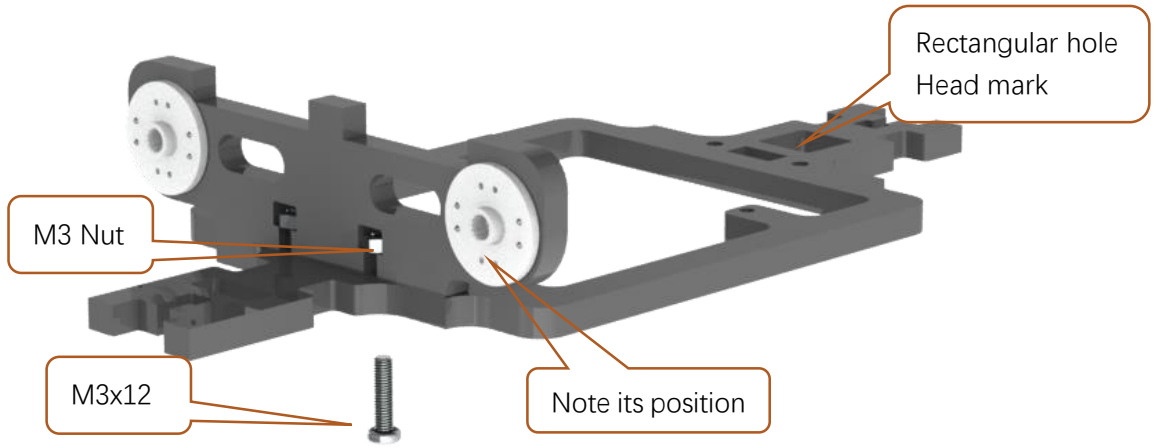


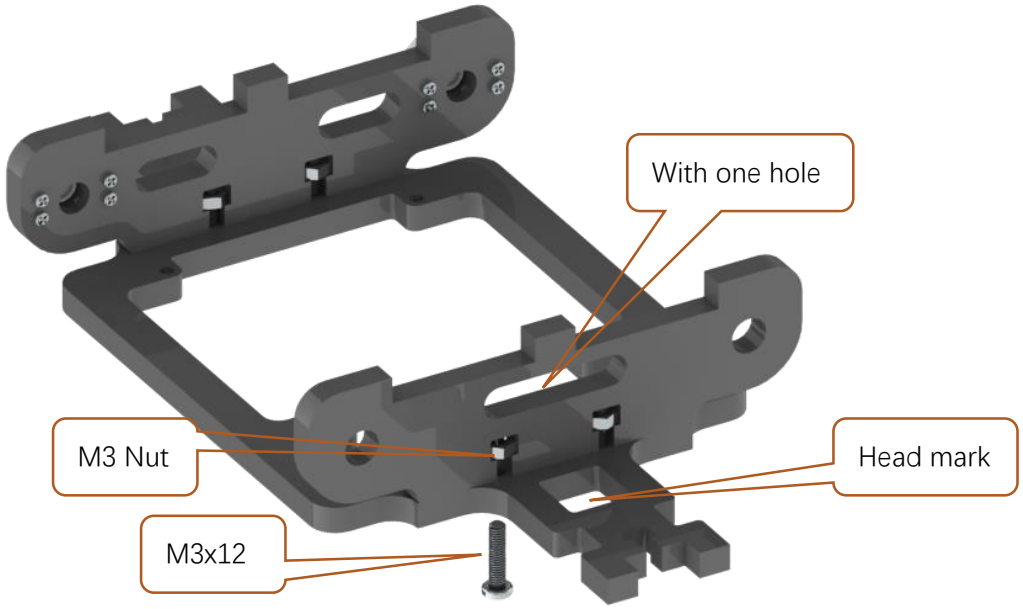


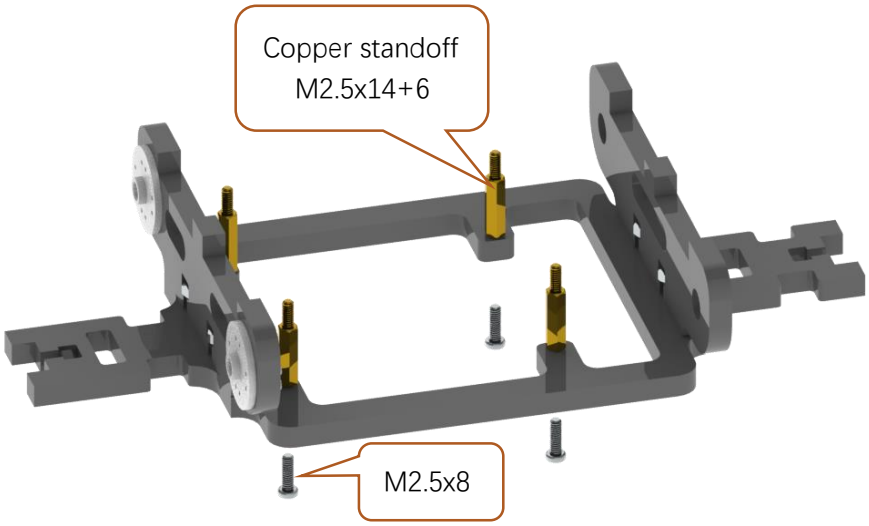
Get following 4 parts.



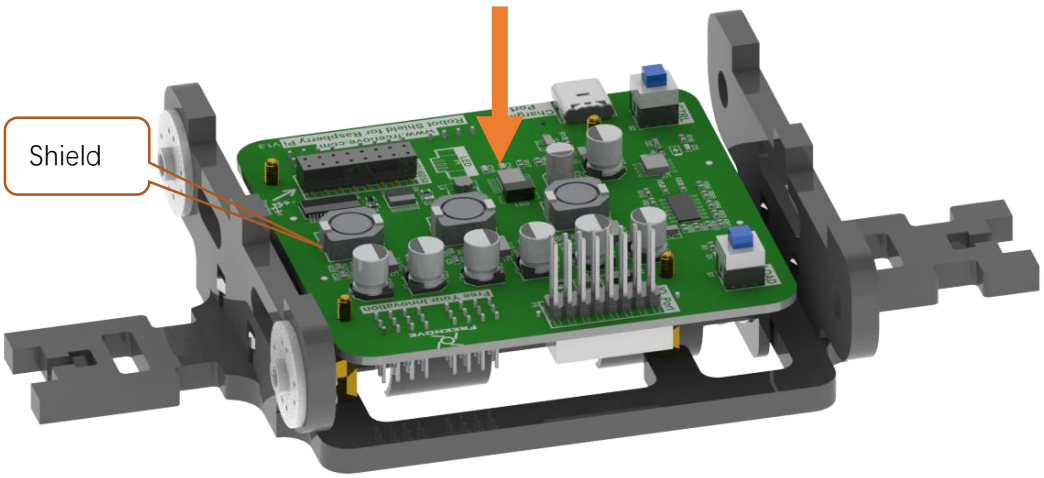
Step 2 Install Body Bracket

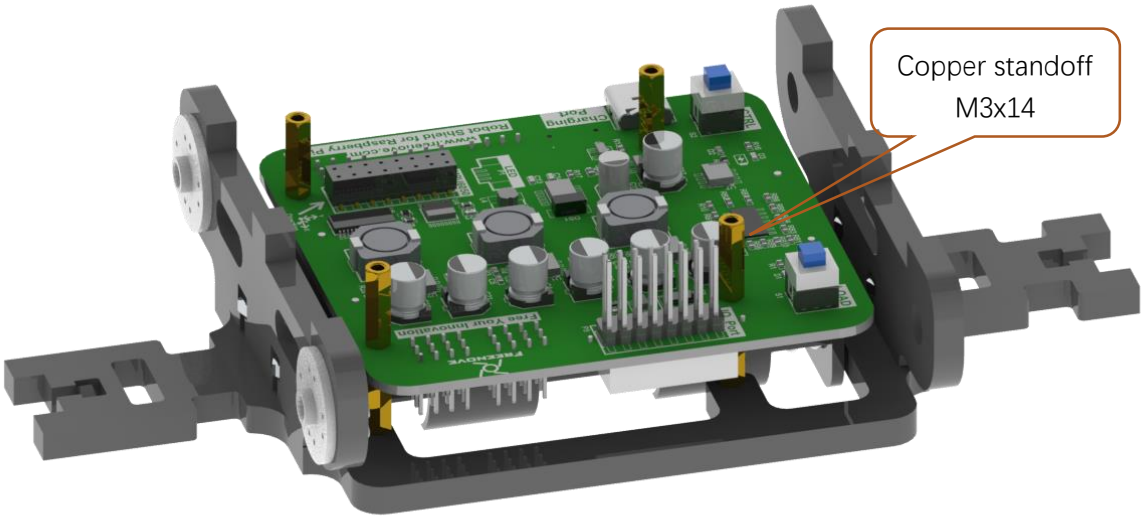






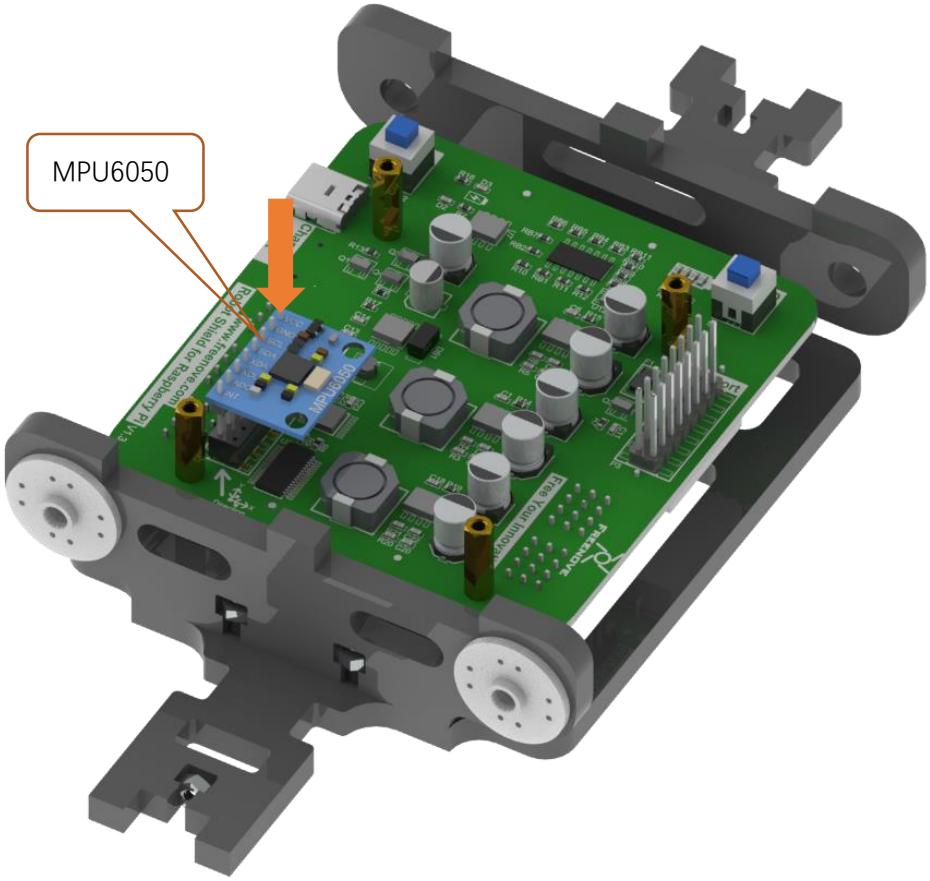
Step 3 Install Shield



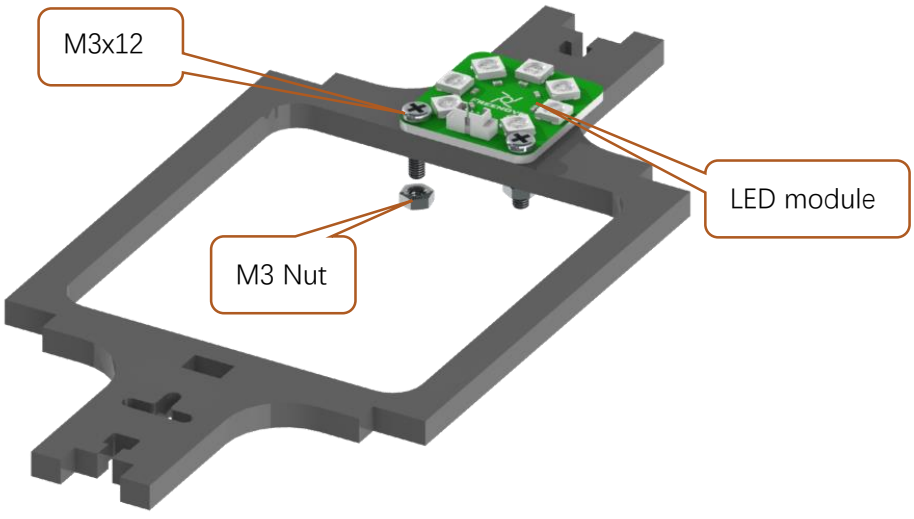


Step 4 Install MPU6050

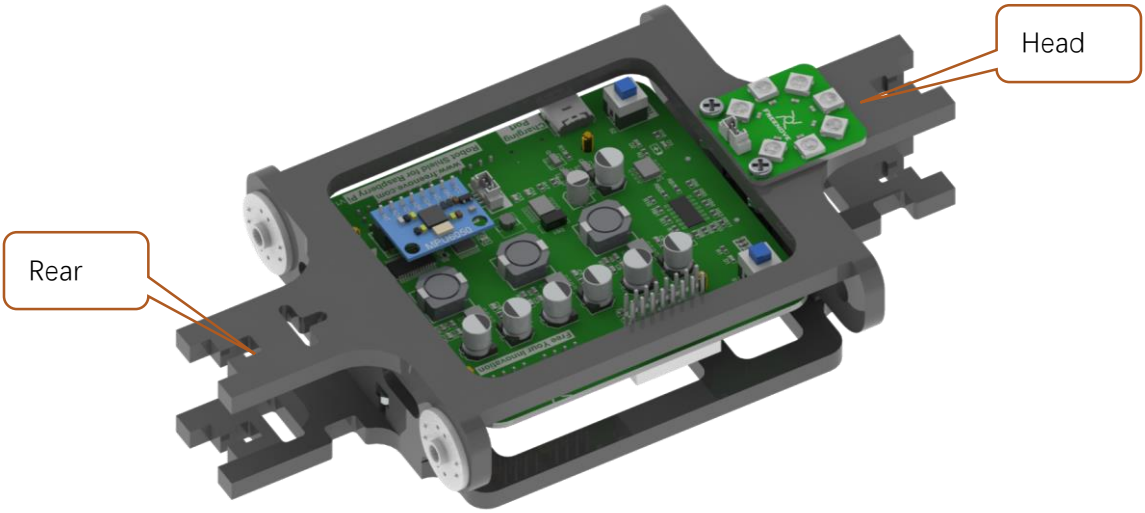
Note: There are two rows of row headers, fit MPU6050 to the Row toward outside.



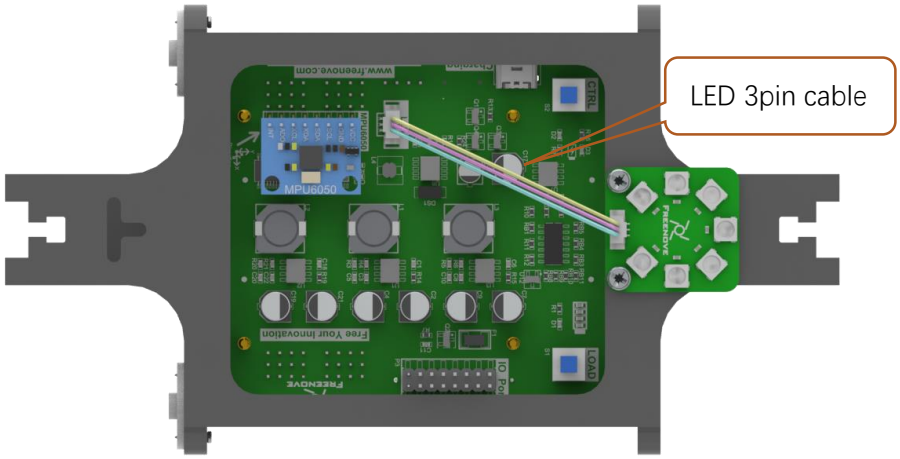
Step 5 Install LED module



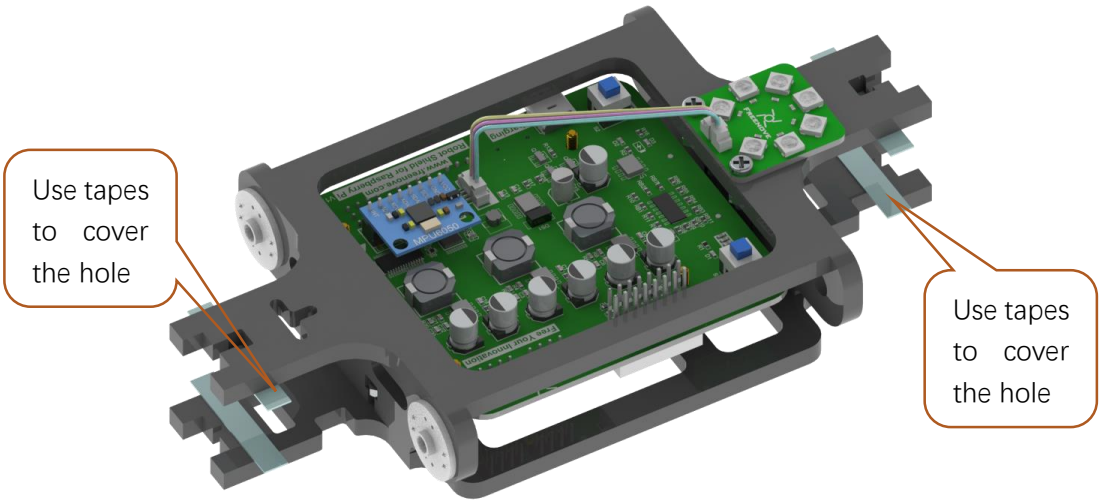
Put the top board on the body part.



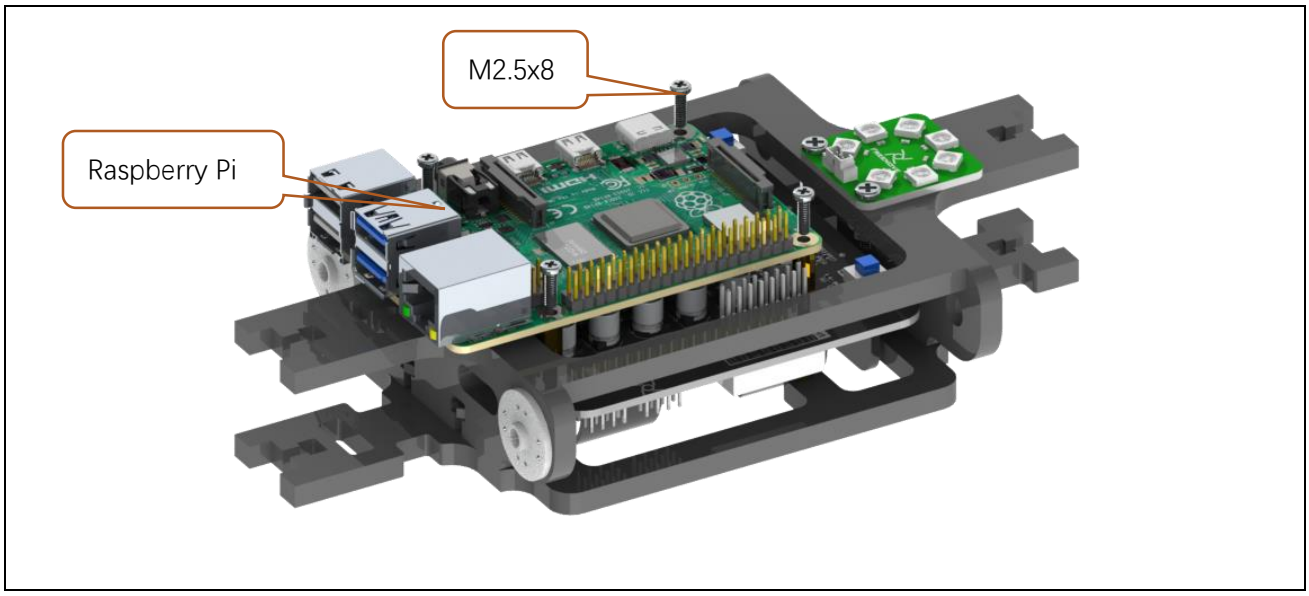
Connect shield with LED module with shield.



This step will be very helpful for later assembly.

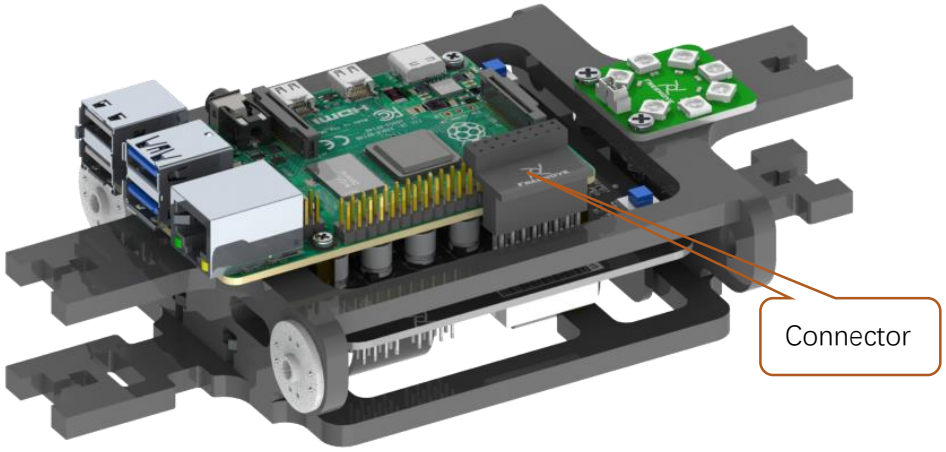


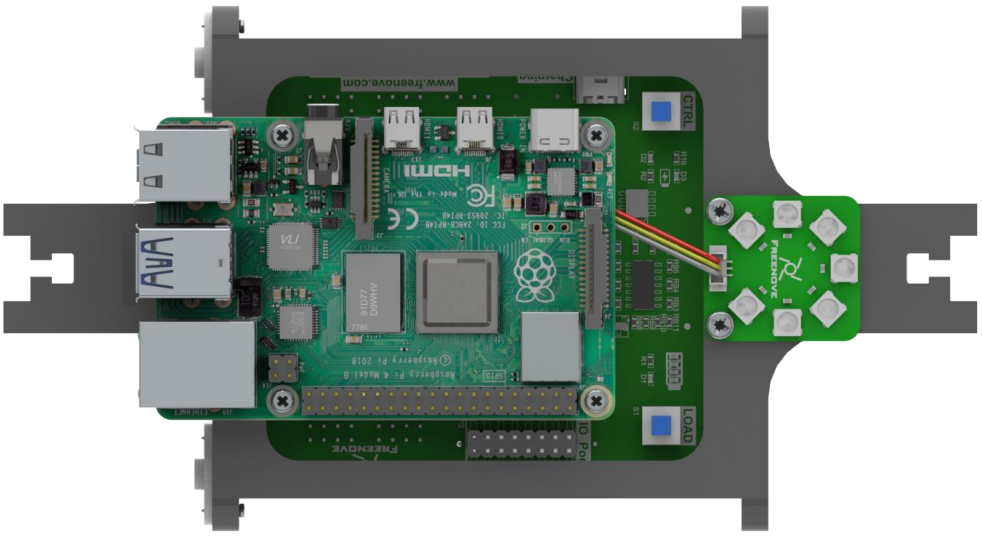
Step 6 Install Raspberry Pi



Step 7 Install Connector

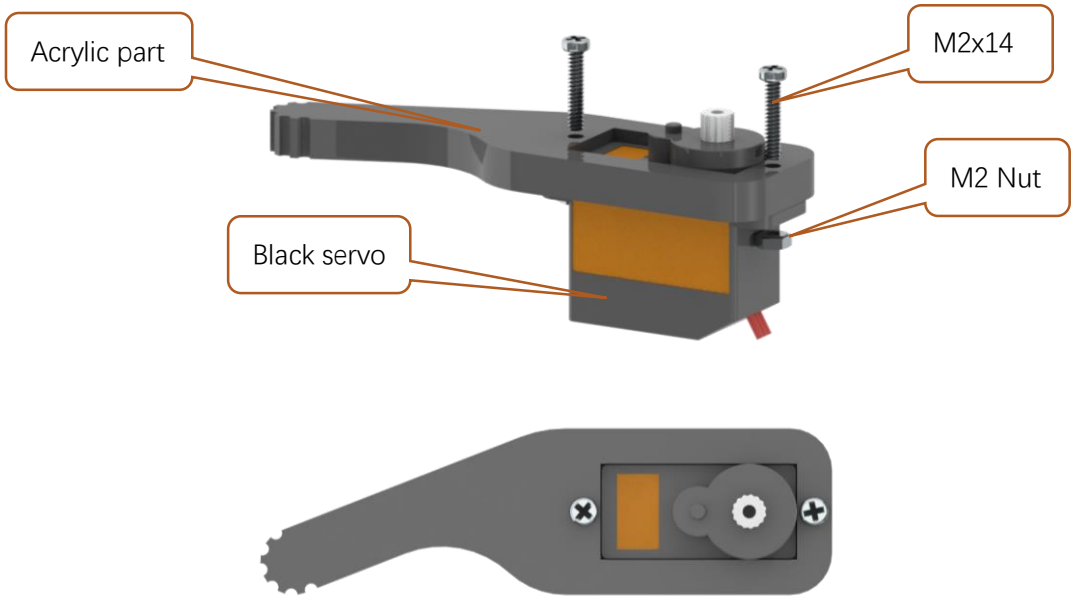
Install connector to connect Raspberry Pi and shield.



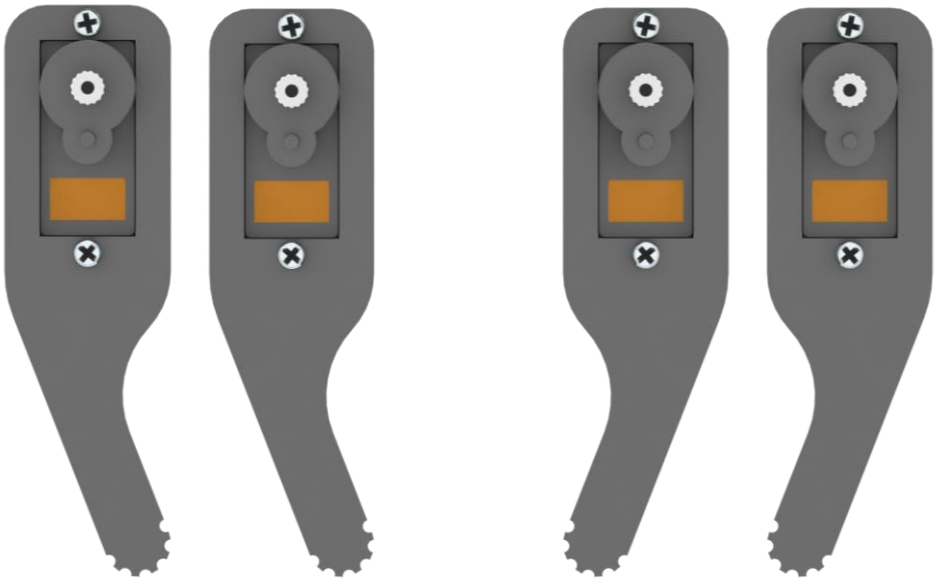


Now we have assembled the core body part. We will assemble leg parts later.

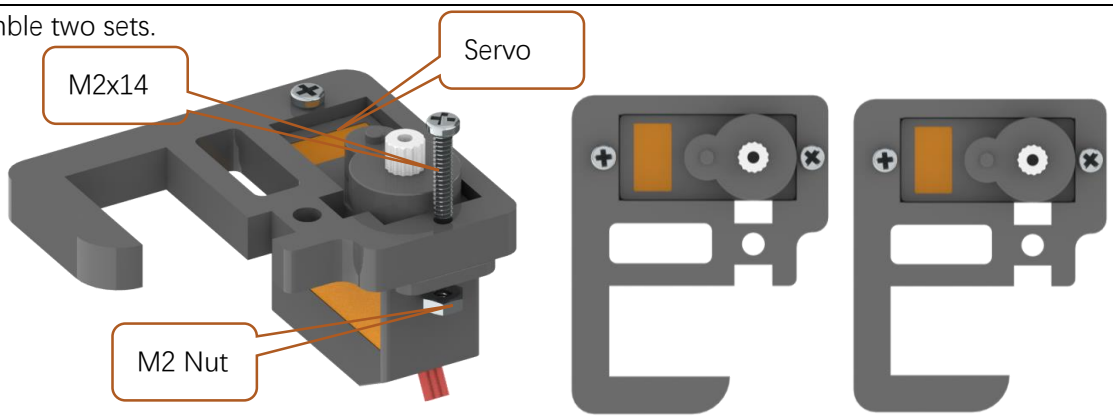
Step 8 Install Servo to Acrylic Board



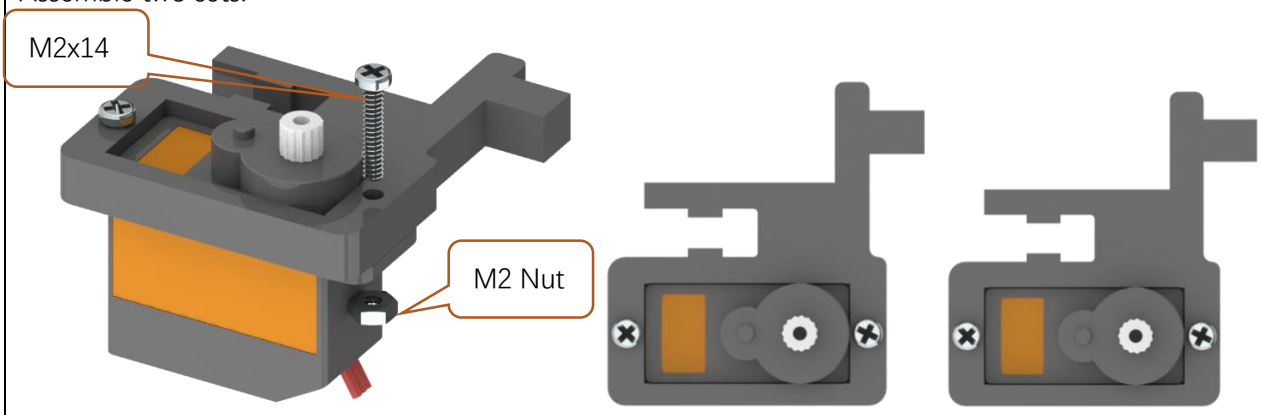
Note: There are four parts. They are different .



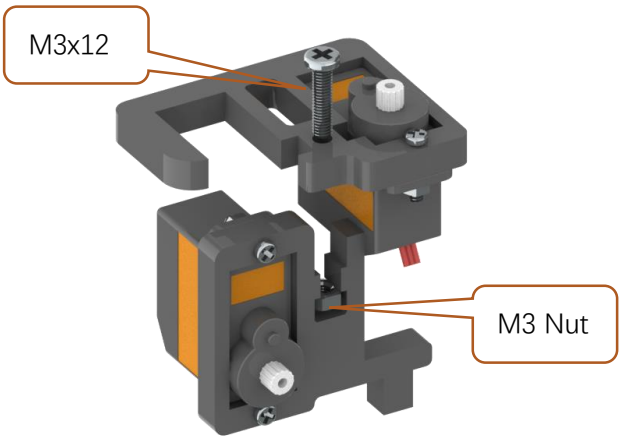
Assemble two sets.



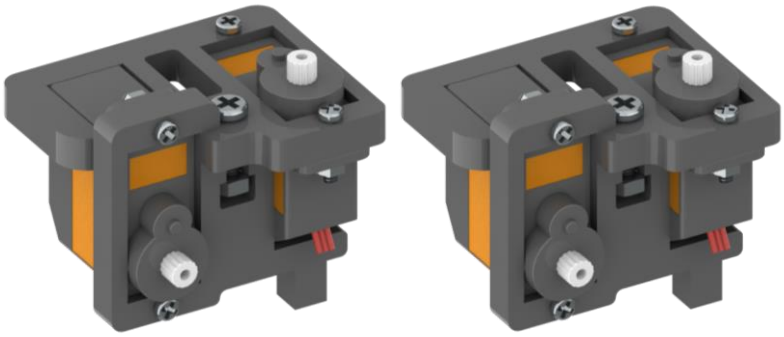
Assemble two sets.



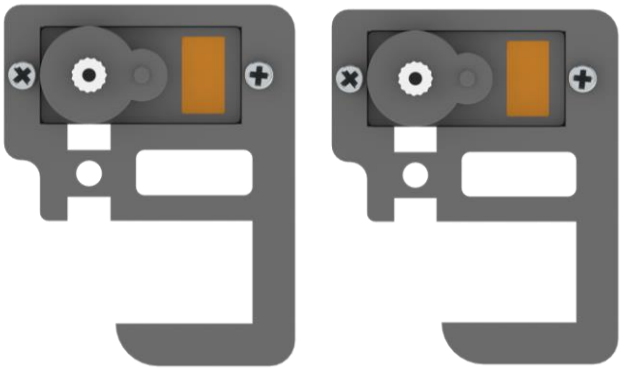
Assemble parts above.



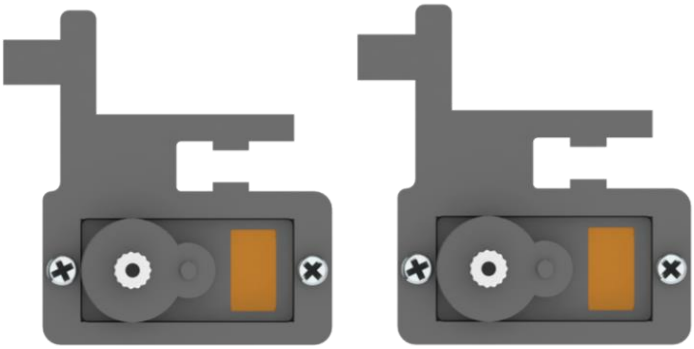
Now you get two sets joint parts.



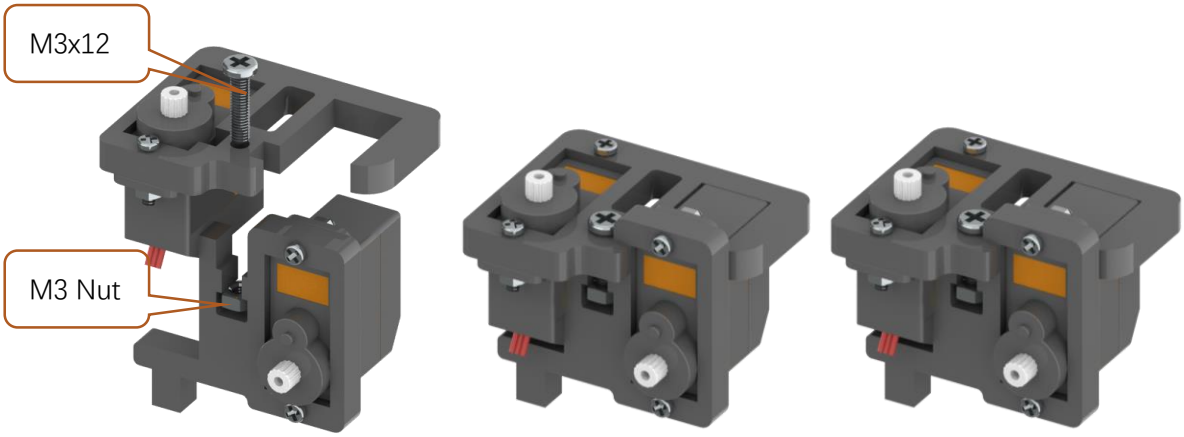
Assemble another two parts. **Note: they are different from parts above.**



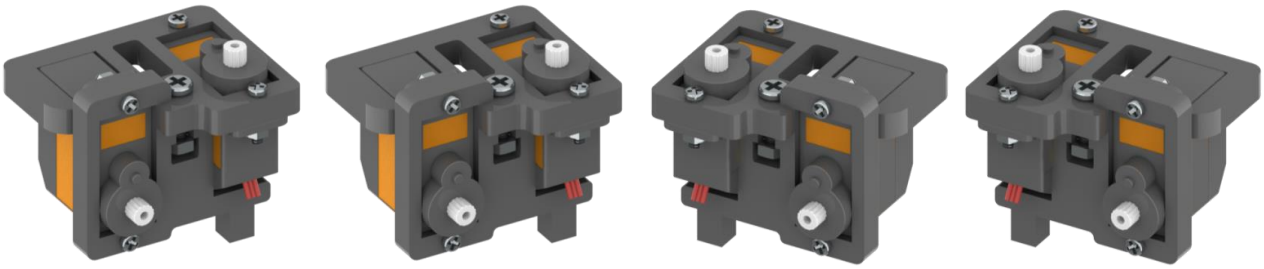
Assemble another two parts. **Note: they are different from parts above.**



Asseble them.

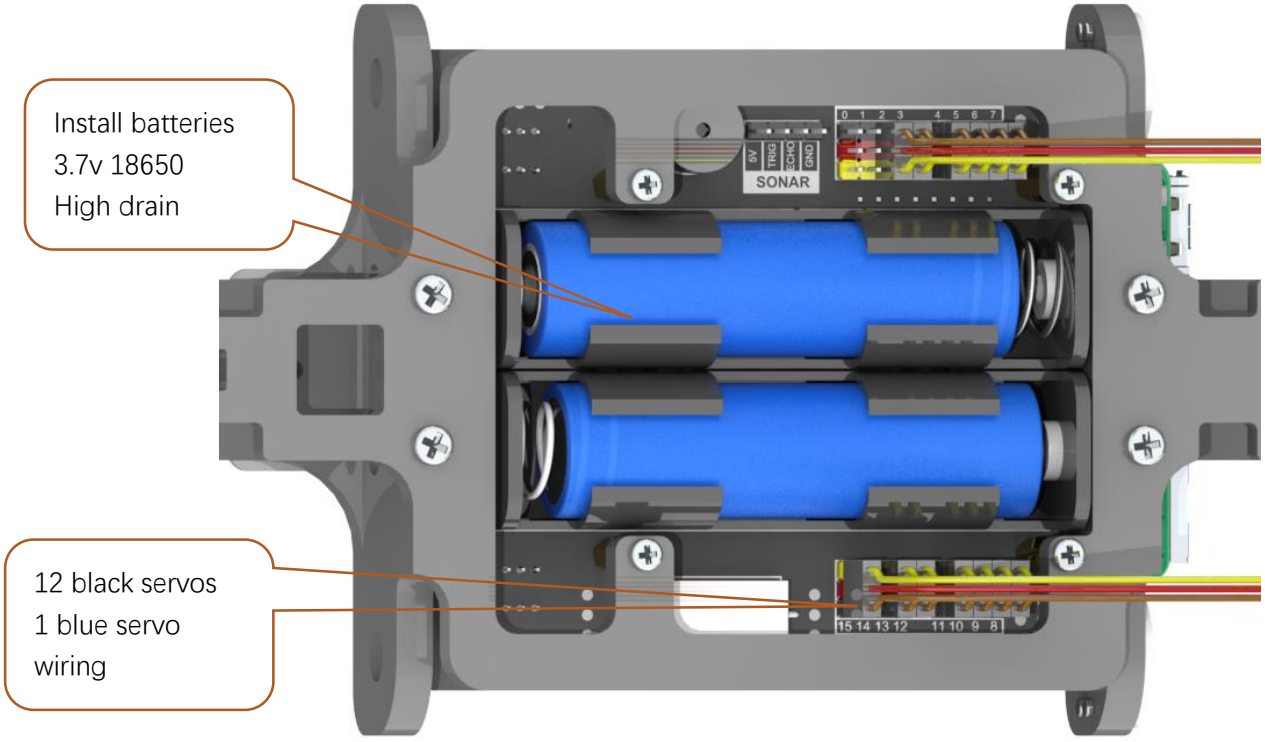


Now you will get four parts like below. **Note, they are different.**

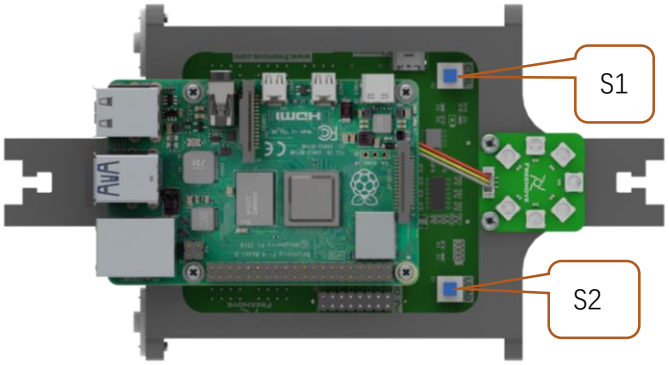


Step 9 Run Servo Program (Necessary)

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



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**. There will be two indicators on.



Turn on the Raspberry Pi. Make sure all three steps in Chapter 1 have be performed correctly. Otherwise, please perform the steps in Chapter 1 first.

a. Type command below to enter servo code folder "Server".

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

b. Run Servo.py

```
sudo python Servo.py
```

Then all servos will rotate to 90°. It is a necessary initialization.

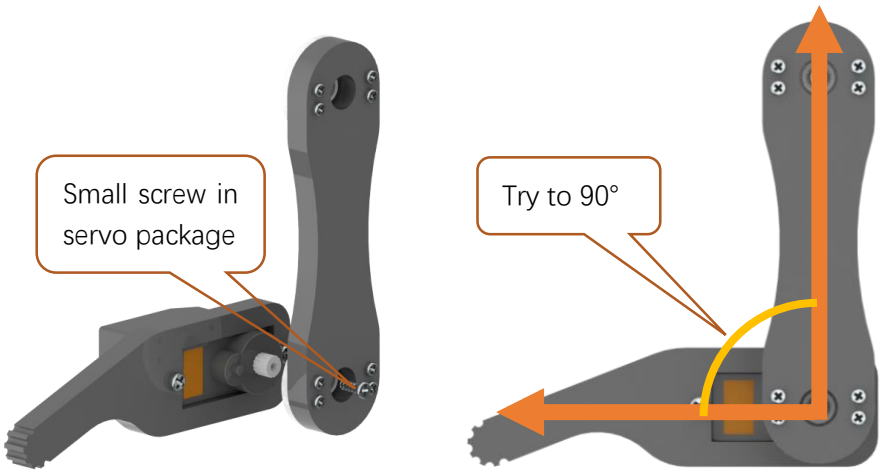
Next we will connect the servos to disk arm.

Keep the **power on** and all servo connected. Don't disconnect **servo**.

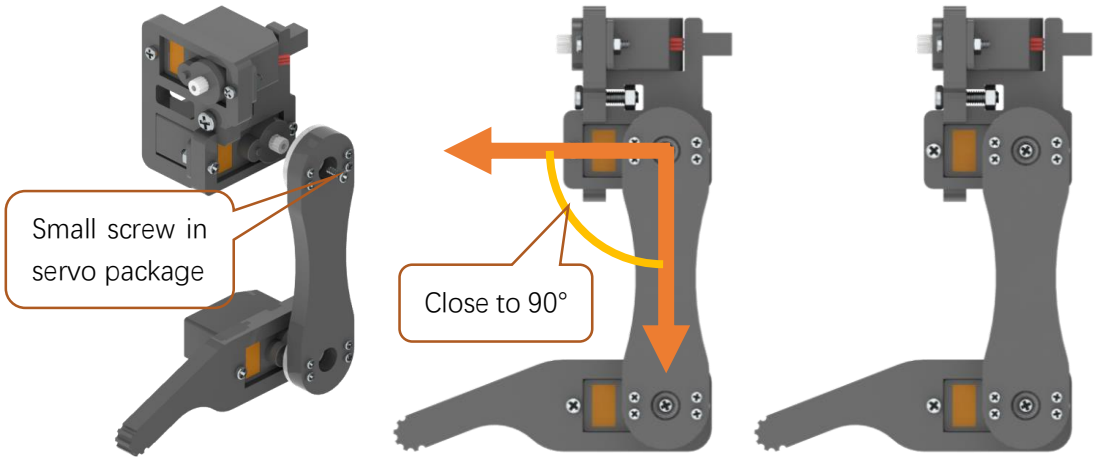
Step 10 Assemble Leg to Body

Keep the power on and all servo connected to shield. Don't disconnect wiring. We need keep the servo under 90° when connected to arm. (servo wires are not show later.)

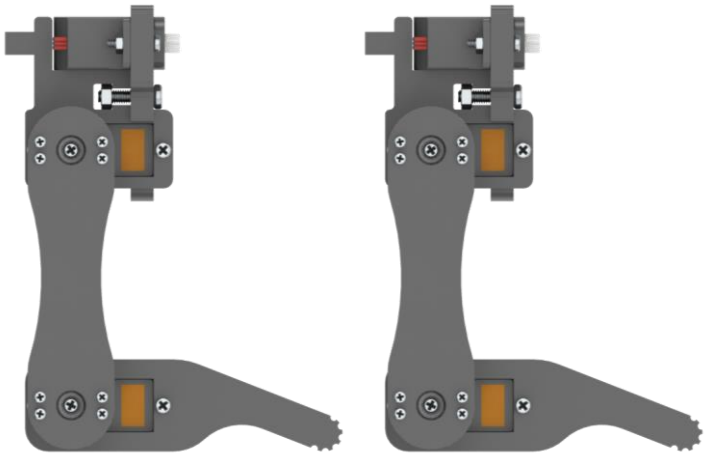
Try to install them colesed to 90°. The angles are reasonable within 65~115.



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

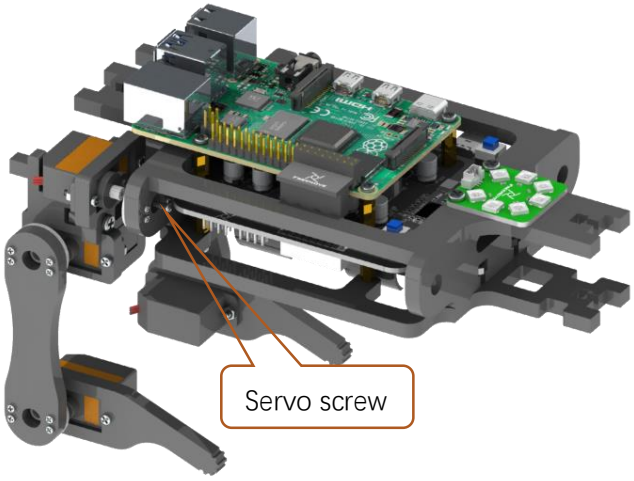


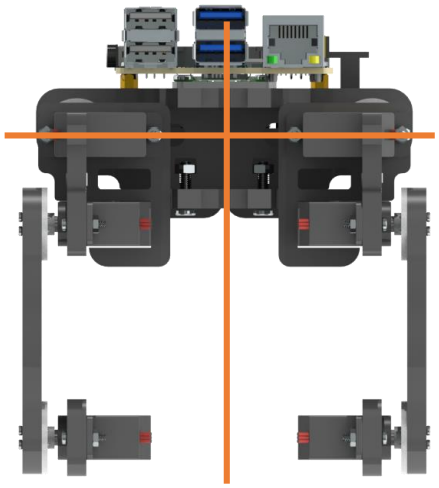
Assemble legs of another side.



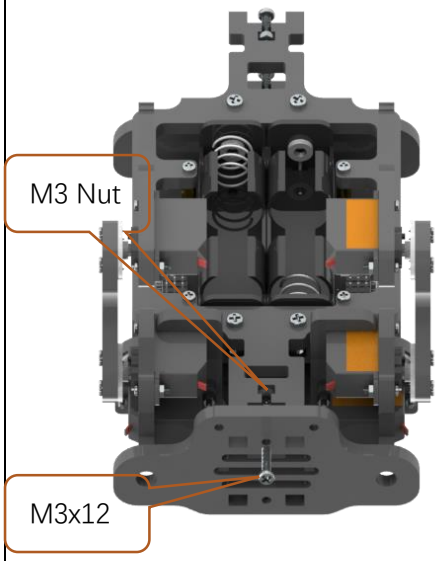
Keep the power on and all servo connected to shield. Don't disconnect wiring.

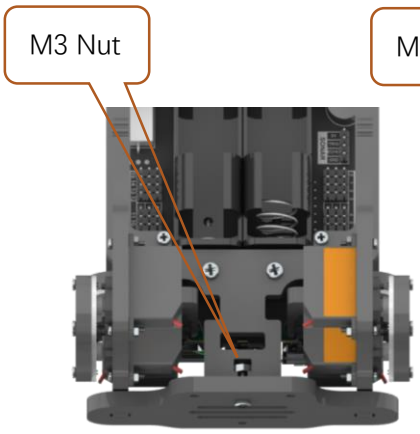
Select two legs of **different sides** as rear legs. And install them as below. Then install the servo screws.

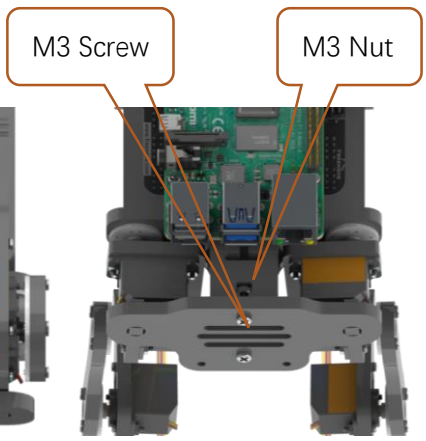




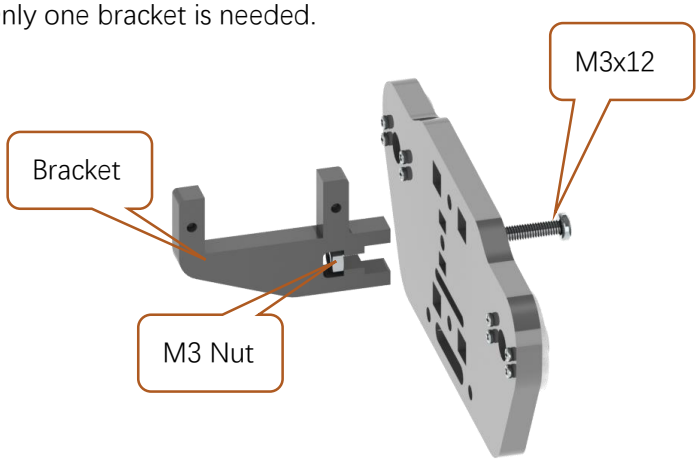
In this step, if you did not cover tape to the holes, you can turn off **Load S1** to in stall the screw. After the screw is installed, turn on the **Load S1**.

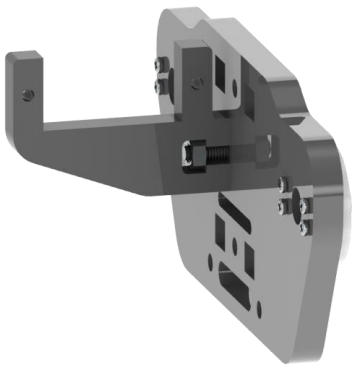






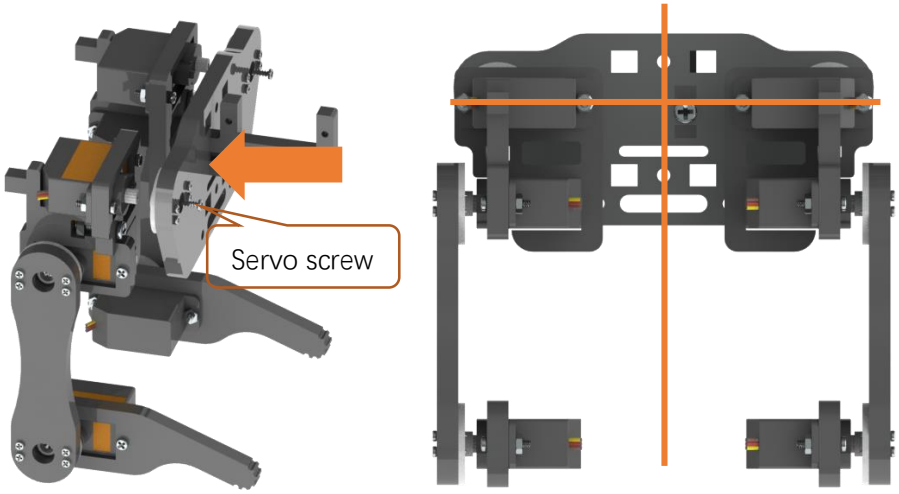
Only one bracket is needed.



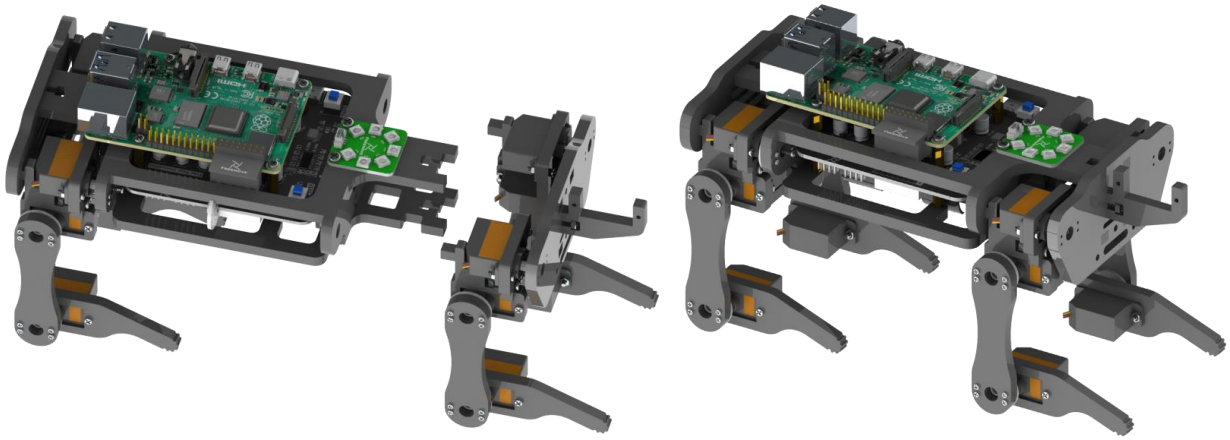


Keep the power on and all servo connected to shield. Don't disconnect wiring.

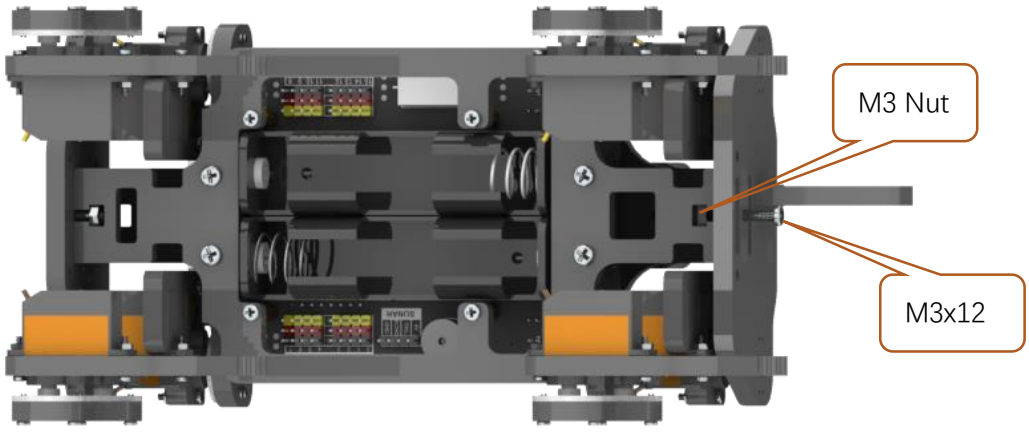
Assemble front legs and install the servo screw.

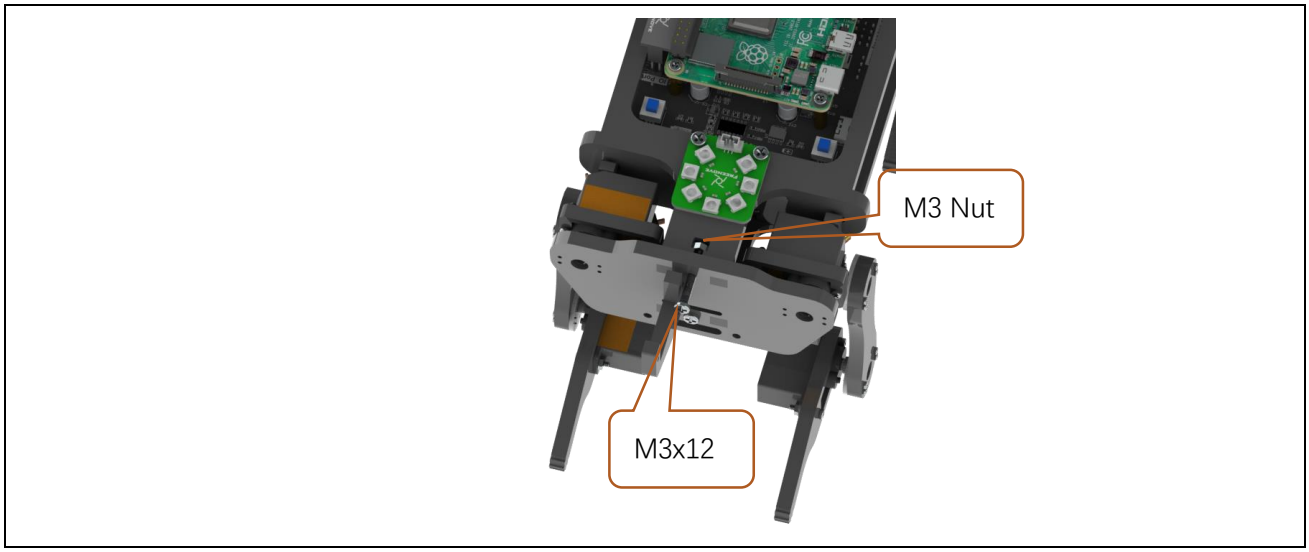


Assemble them.

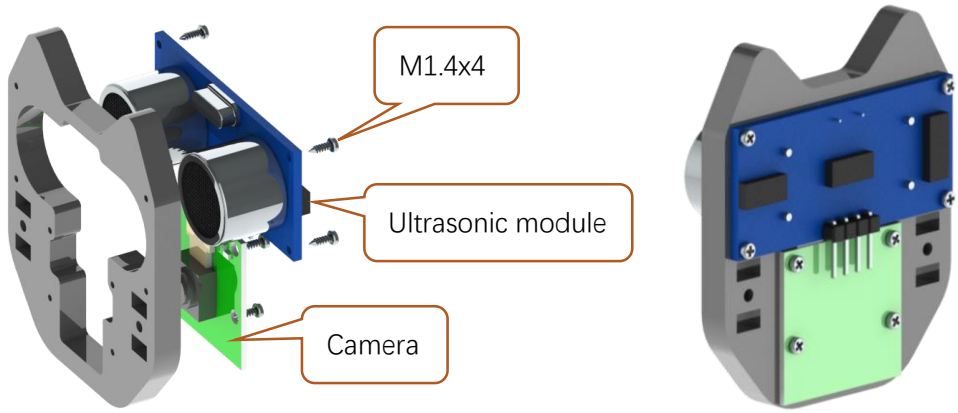


In this step, if you did not cover tape to the holes, you can turn off **Load S1** to in stall the screw. After the screw is installed, turn on the **Load S1**.

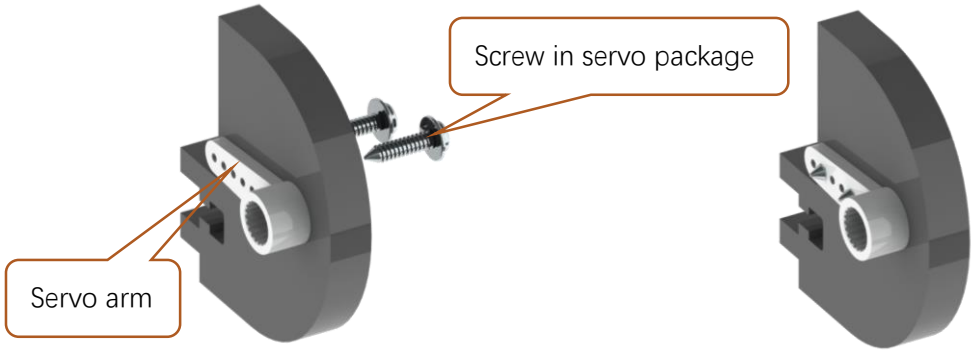


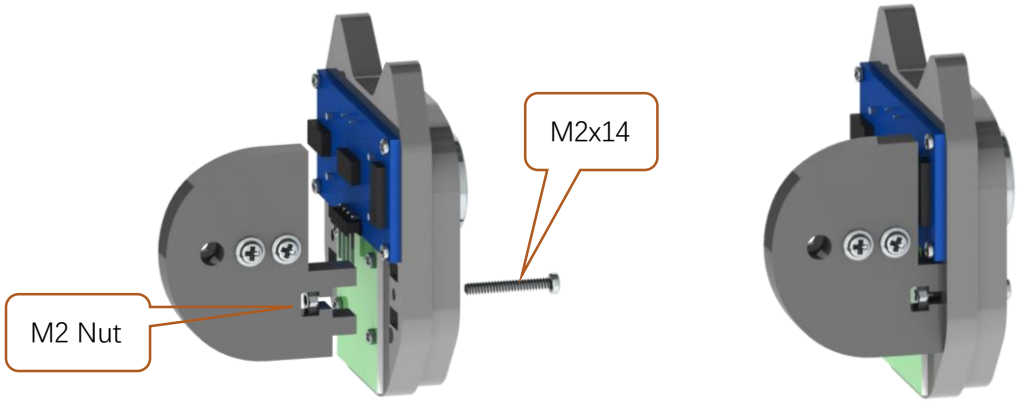

M3 Nut
M3x12

Step 11 Assemble Head

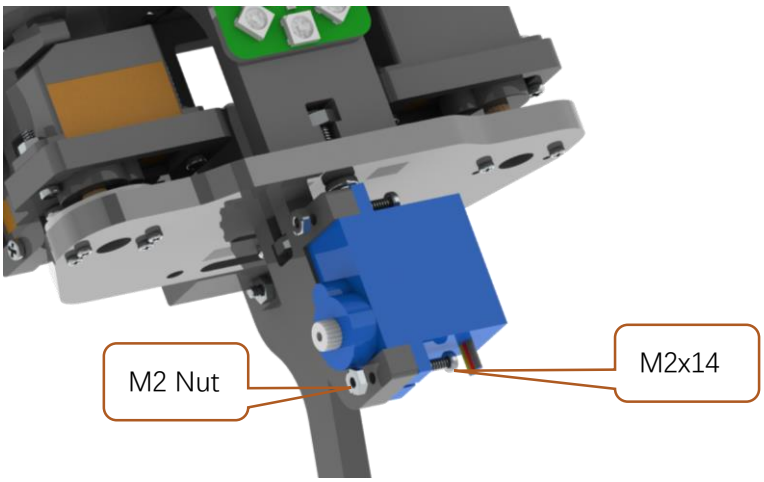


The servo arm and screw are included in the blue servo package.

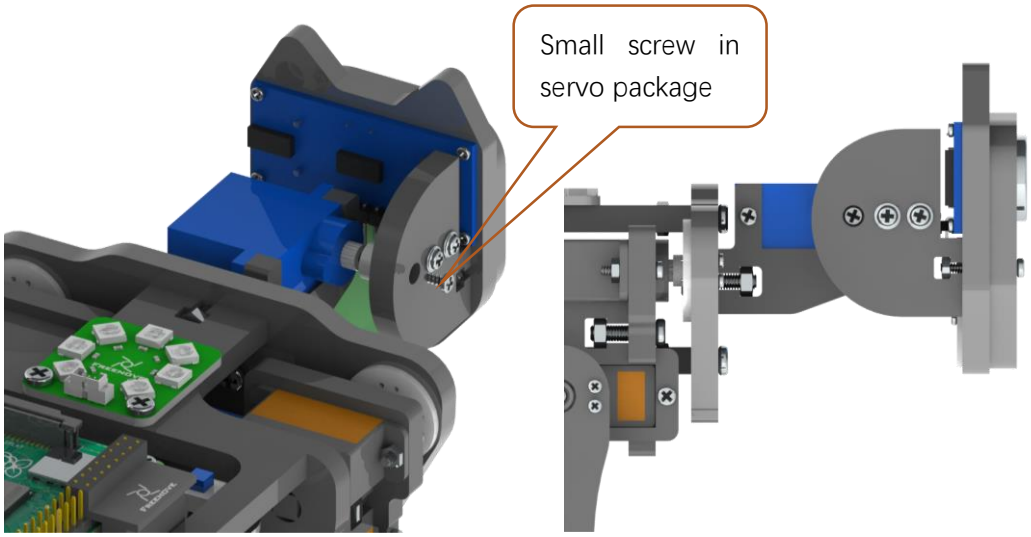


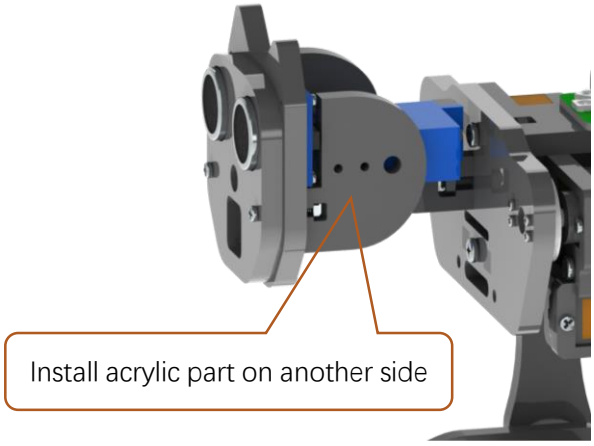


Step 12 Assemble Head to Body



Keep the power on and servo connected when install arm to servo.

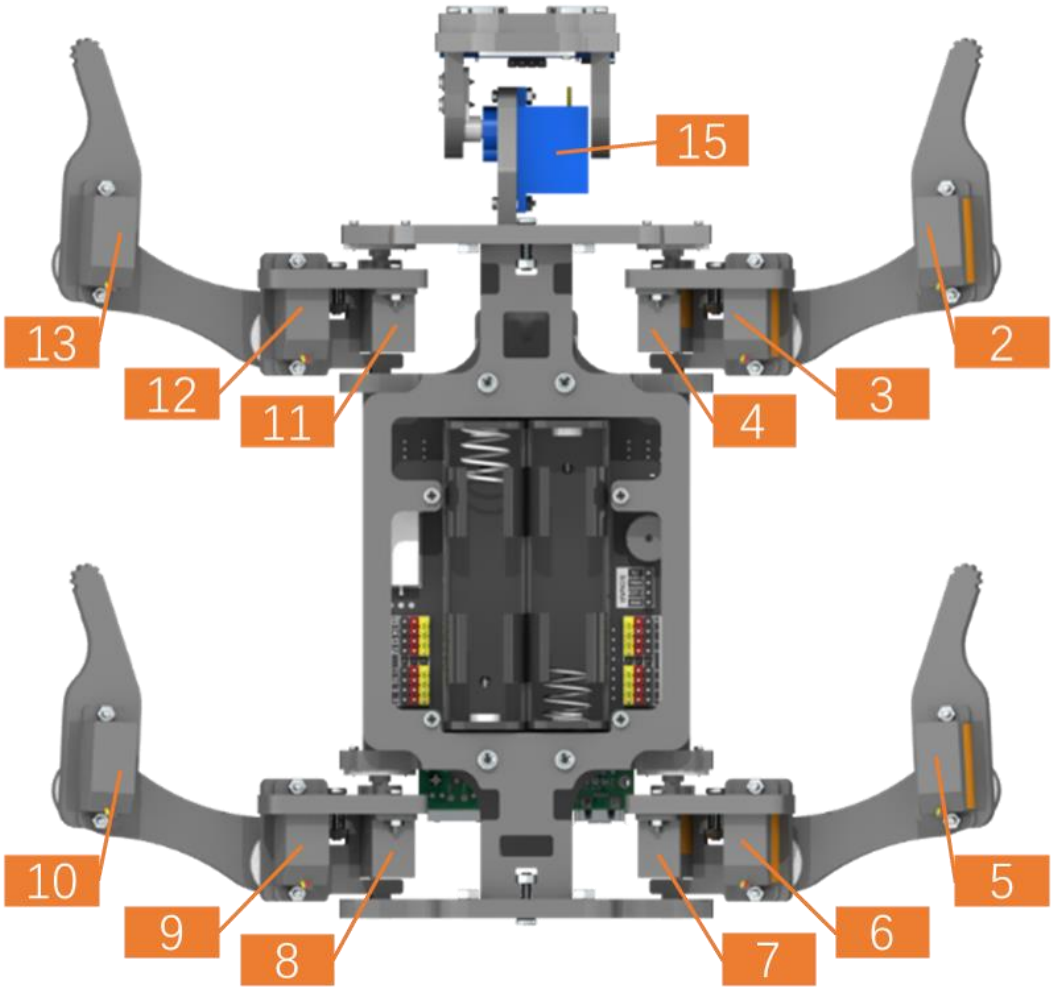


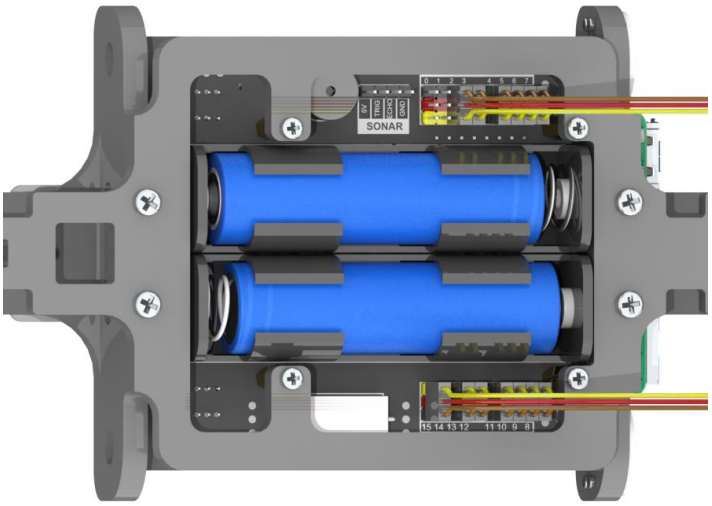


We will complete wiring in next section. Shut down the Raspberry Pi first. Don't turn on Raspberry Pi until chapter 3.

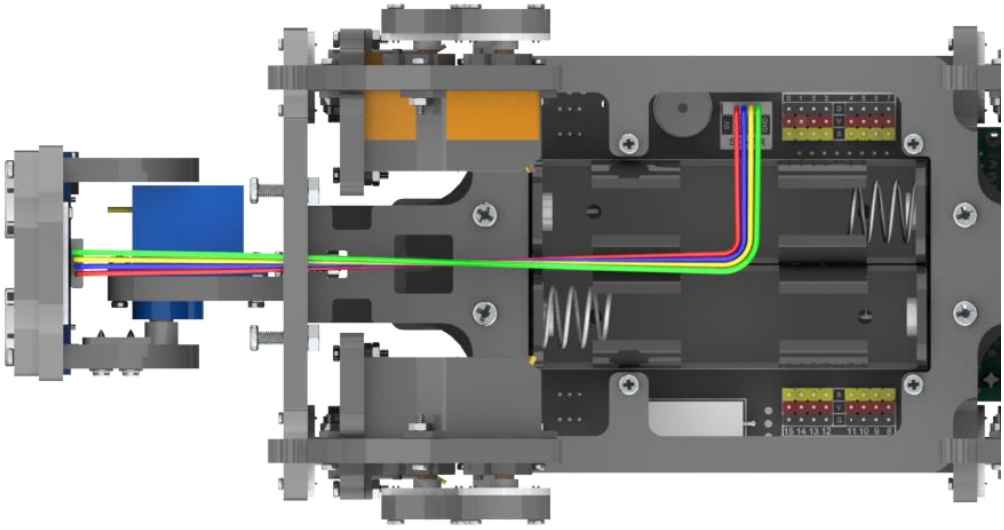
Step 13 Wiring

Connect servo according to the numbers. **Note servo port 0, 1, 14 have no servo**. They are spare.



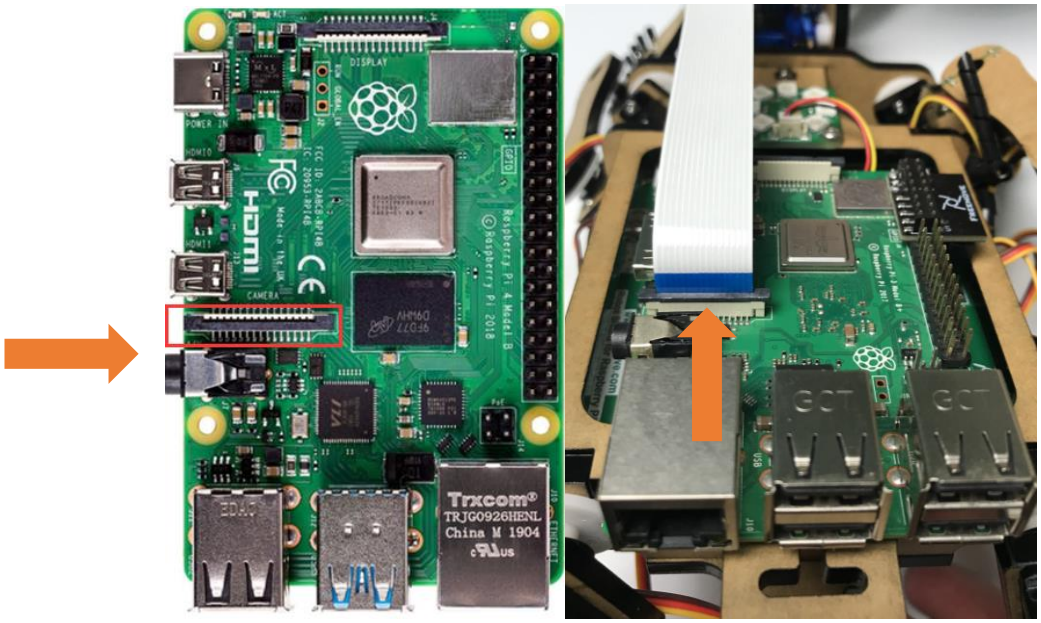


Ultrasonic module wiring (Note: **Do not connect wrongly. If you connect 5V to GND, this may damage the ultrasonic module**)

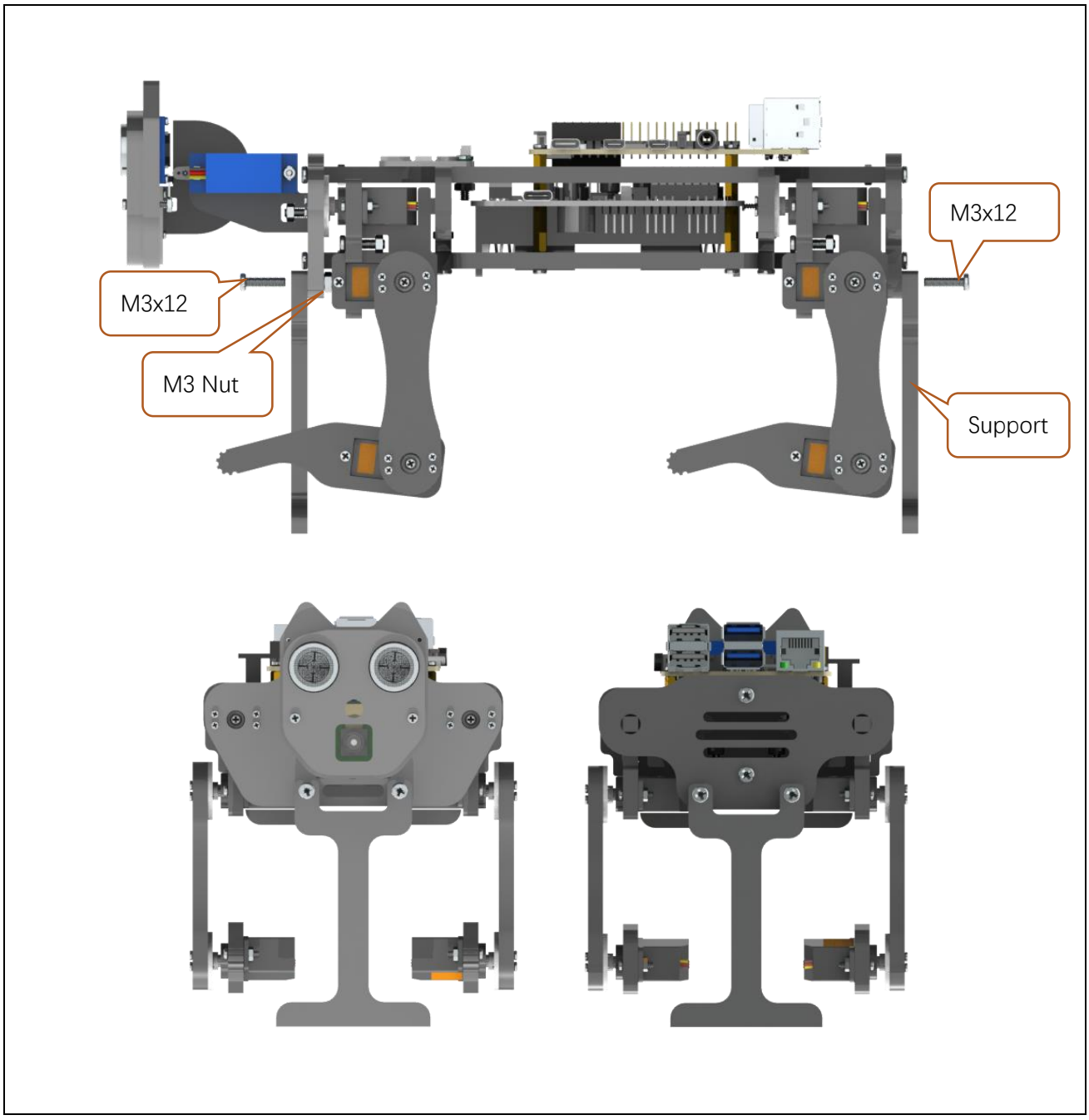


Pay attention to the **blue** side of camera cable.

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



Step 14 Install Calibration Support



Chapter 3 Module test (necessary)

If there are 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.

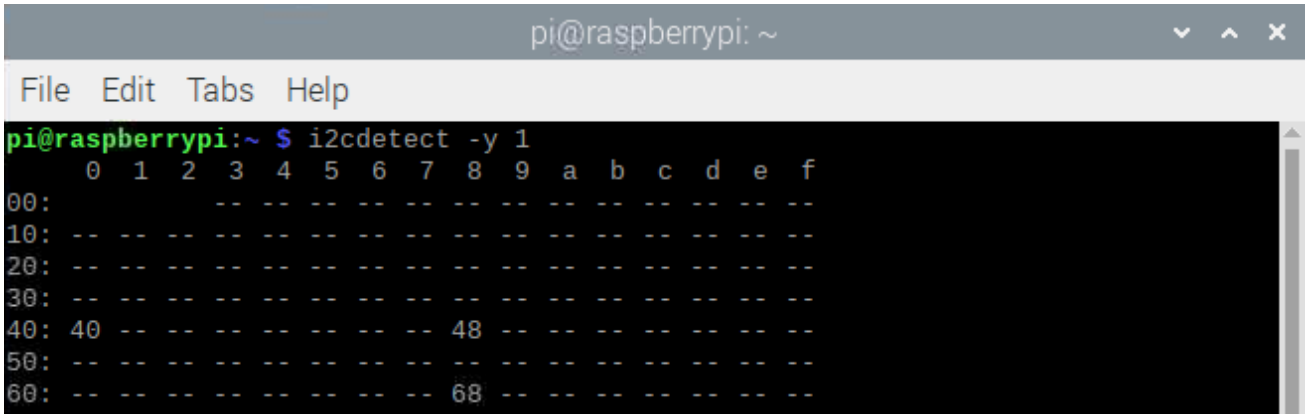
It is recommended to use VNC Viewer to build this robot when you use remote mode to login RPi, or there will be errors. Some remote way doesn't support GUI.

Servo

Execute following command to check i2c address.

```
i2cdetect -y 1
```

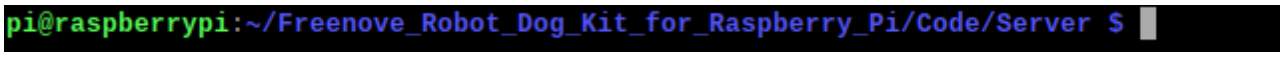
As shown in the figure below, the addresses 0x40, 0X48, and 0X68 correspond to the PCA9685 chip, ADS7830 chip, and MPU6050 module, respectively.



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
```

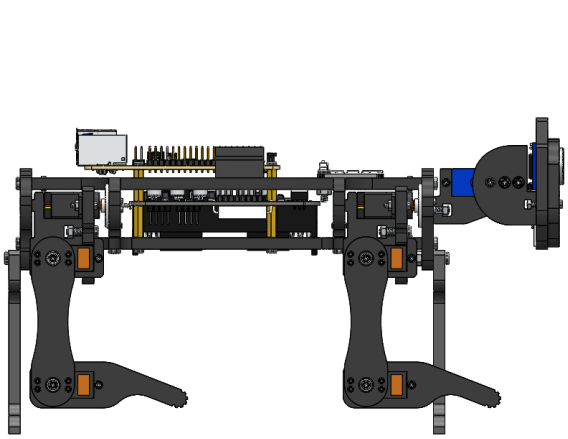
```
pi@raspberrypi: ~/Freenove_Robot..._Kit_for_Raspberry_Pi/Code/Server
File  Edit  Tabs  Help
pi@raspberrypi:~ $ cd ~/Freenove_Robot_Dog_Kit_for_Raspberry_Pi/Code/Server
pi@raspberrypi:~/Freenove_Robot_Dog_Kit_for_Raspberry_Pi/Code/Server $ sudo pyth
on test.py Servo
Program is starting ...
```

Result:

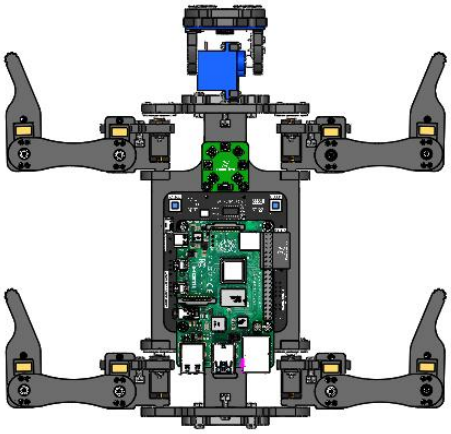
After assembly before in last chapter, the robot should be shown as picture A.

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

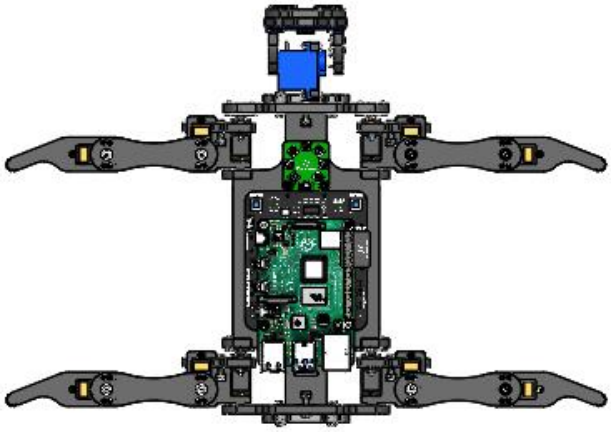
If the situation is not correct, check the servo wiring.



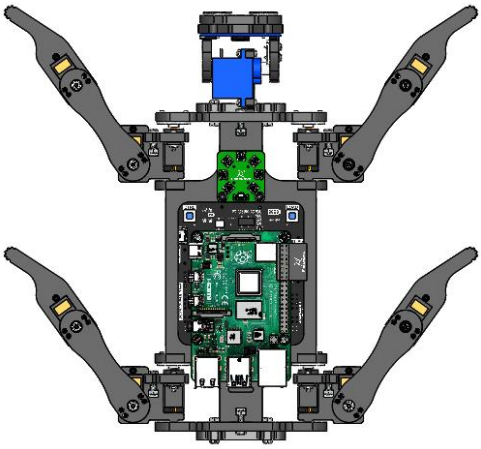
(A)



(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 second test.py command.

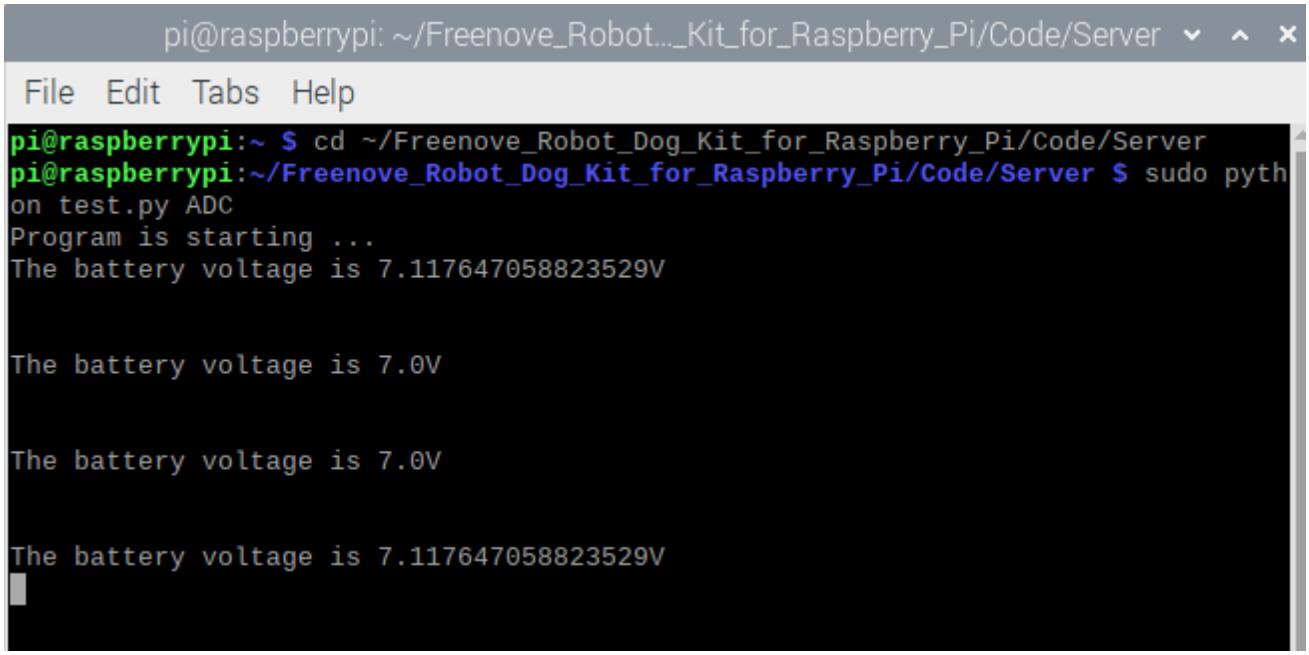
```
pi@raspberrypi:~/Freenove_Robot_Dog_Kit_for_Raspberry_Pi/Code/Server $
```

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 ADC
```



Result:

Print batteries voltage value 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 range is 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 of ultrasonic module and front obstacle, with unit CM.

Led

There is a 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, has three input parameters, 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, wait for 20ms, and then the LED1 color is also red, so that all eight LEDs are lit and 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 turn 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)

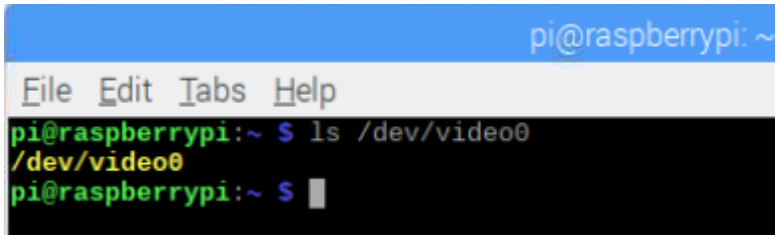
This function has one input parameter. If the input is '1', the buzzer will be turned on. If the input is '0', the buzzer will be turned off.

Camera

Enter following command to check if the camera is connected successfully.

```
ls /dev/video0
```

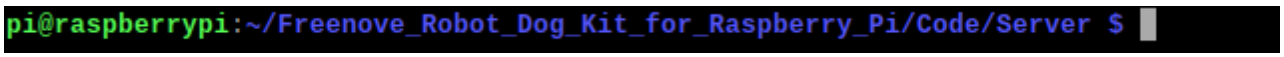
Then the device node will be shown below:



If the result above does not appear, please check the camera **Wiring** and if the camera interface is enabled.
(Note: plugging and unplugging the cable requires the Raspberry Pi to be powered off, otherwise the camera module may be burned.)

Run program

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

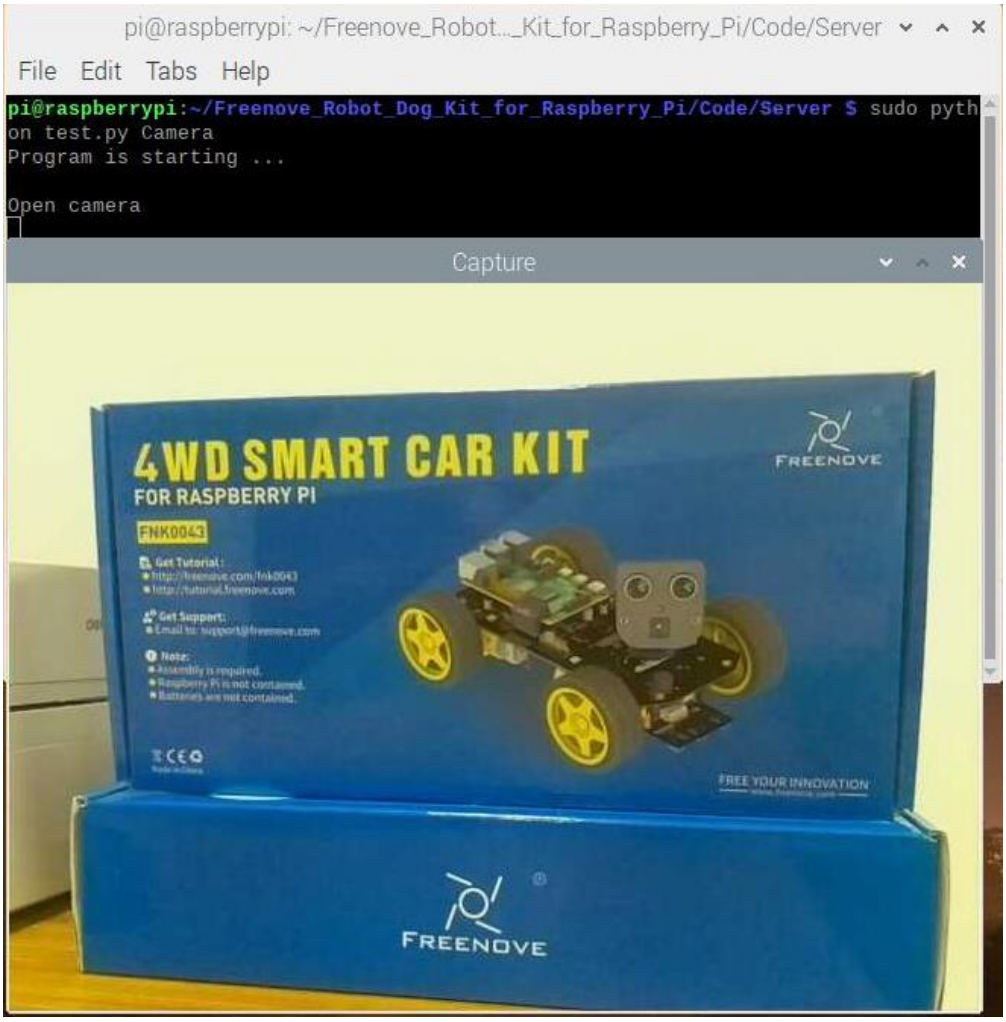


1. If not, execute the cd command:

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

2. Execute test.py command:

```
sudo python test.py Camera
```



Result:

A window will pop up and display the picture shot by the camera. Press “Ctrl + “C” in the **terminal** window to end the program. (Note: pressing "Ctrl + C" in the **camera** window doesn't work.)

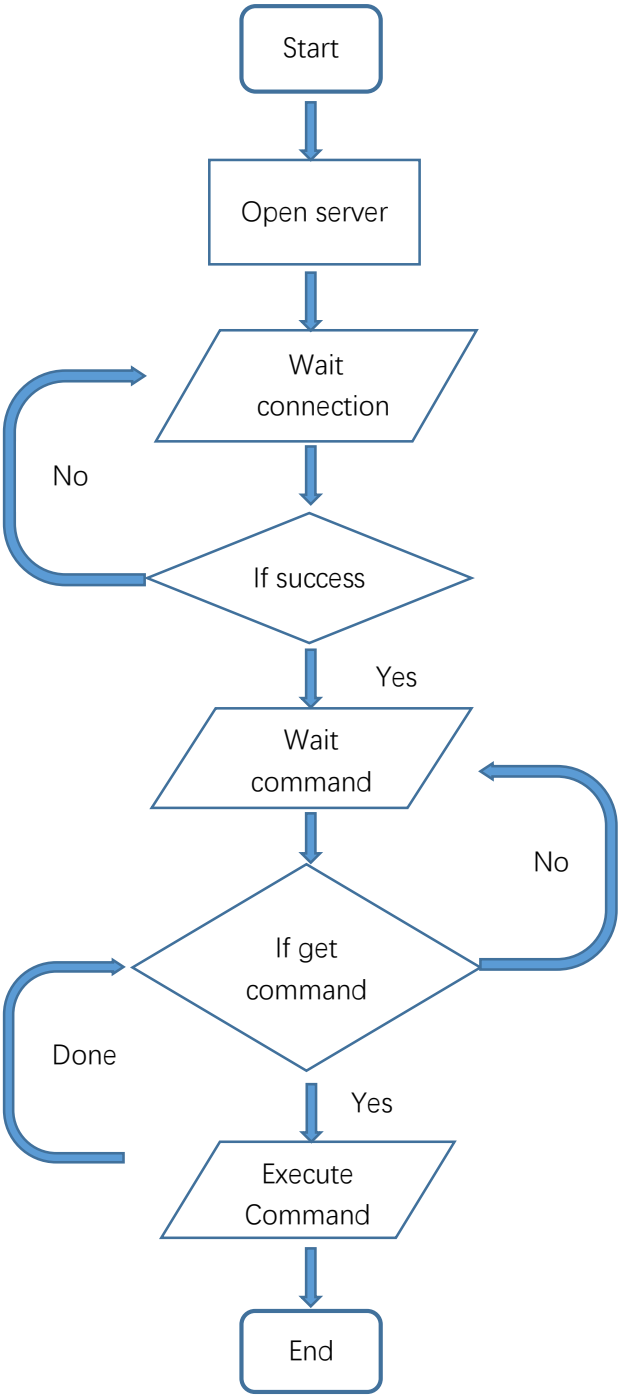
The coed is below:

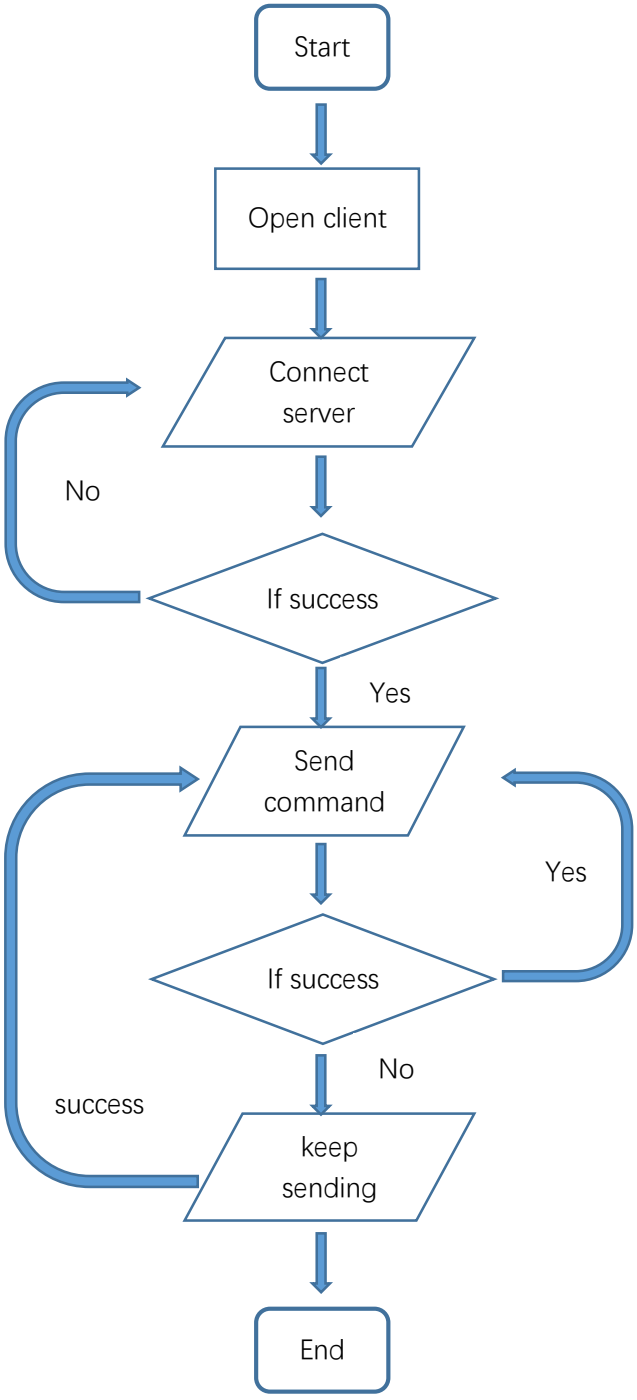
```
import cv2
def test_Camera():
        try:
            print ("\nOpen camera")
            capturing_Flag = True
            cap = cv2.VideoCapture(0)
            while(capturing_Flag):
                ret, frame = cap.read()
                cv2.imshow("Capture", frame)
                cv2.waitKey(5)
            cv2.destroyAllWindows()
        except KeyboardInterrupt:
                print ("\nClose camera")
                capturing_Flag = False
```

Chapter 4 Robot Dog

If there are 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.





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” under 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

Turn on Server

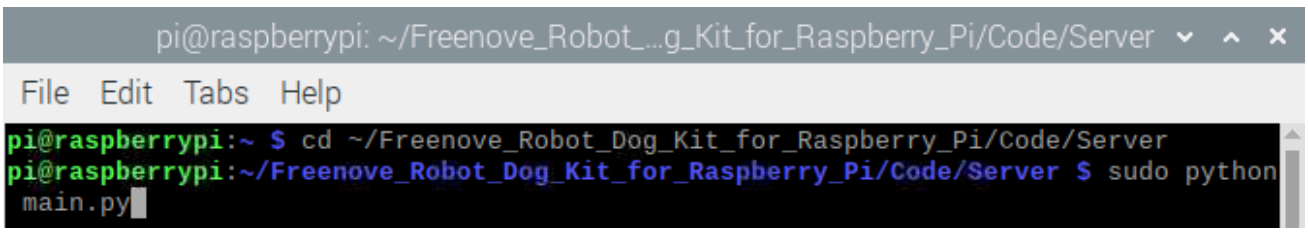
Enter following command in the terminal.

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

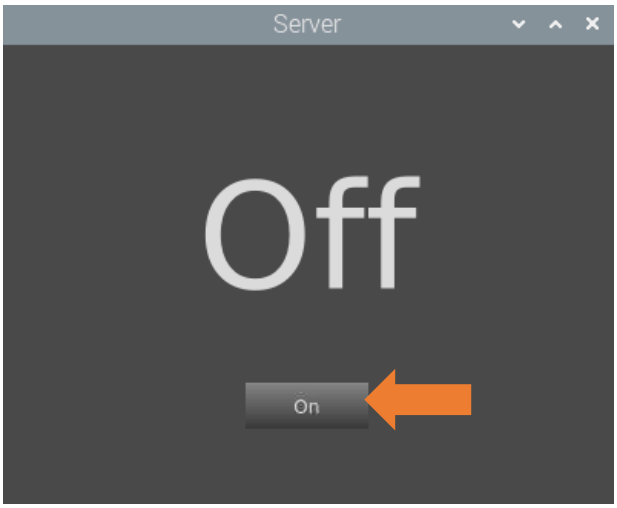
```
cd ~/Freenove_Robot_Dog_Kit_for_Raspberry_Pi/Code/Server
```

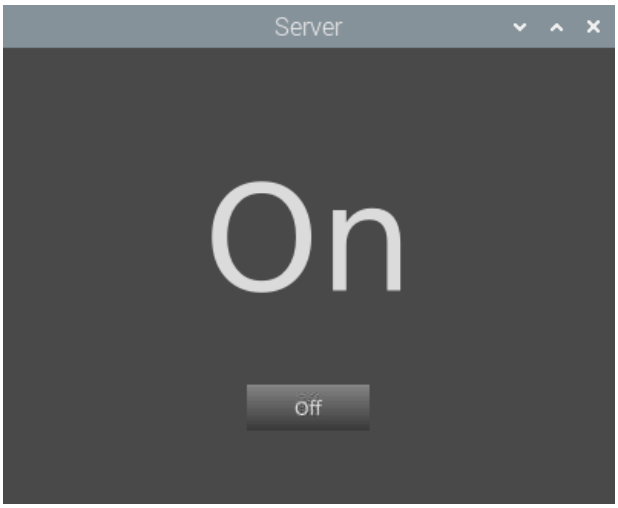
2. Run main.py:

```
sudo python main.py
```



The interface is as below: Click “On” to open the server. The result is show as below:

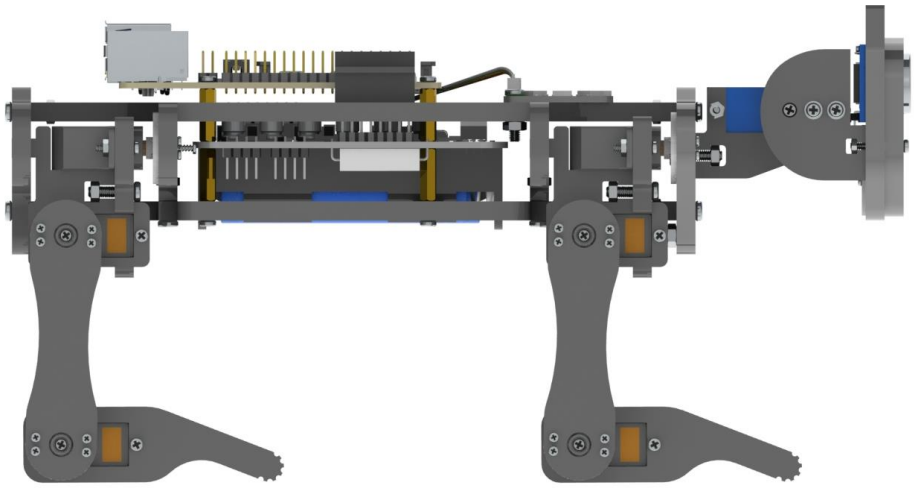




You will see your server address (Raspberry Pi IP).

```
pi@raspberrypi:~/Desktop/20200327Freenove_Robot_Dog_Kit_for_Raspberry_Pi/Code/Se
rver $ sudo python main.py
libEGL warning: DRI2: failed to authenticate
QStandardPaths: XDG_RUNTIME_DIR not set, defaulting to '/tmp/runtime-root'
Server address: 192.168.1.131
```

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



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 run 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” under 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

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, and it takes some time. At this time, it does not need to run a server and a Raspberry Pi. You can turn off the Raspberry Pi first. After the installation is complete, turn on the Raspberry Pi and the server.

Install python3

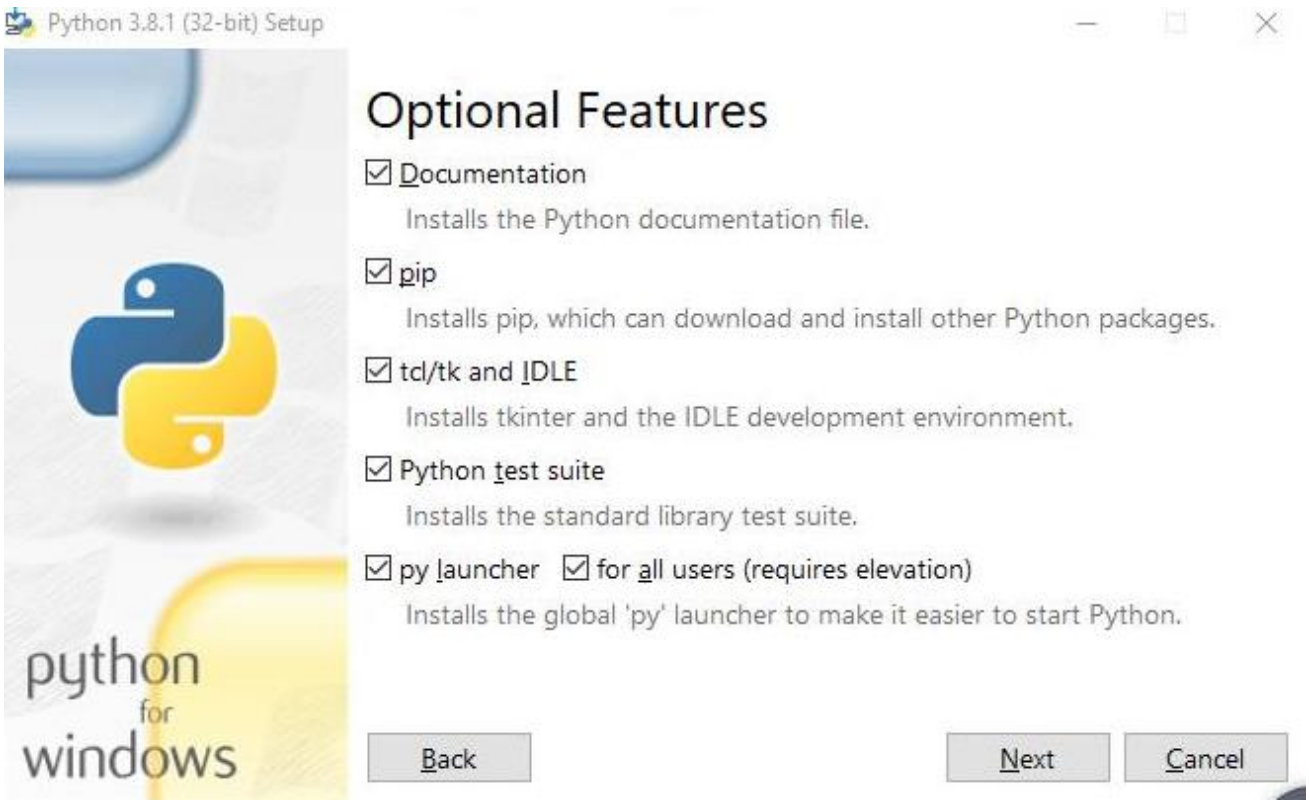
Download the installation file via below:
https://www.python.org/downloads/windows/

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

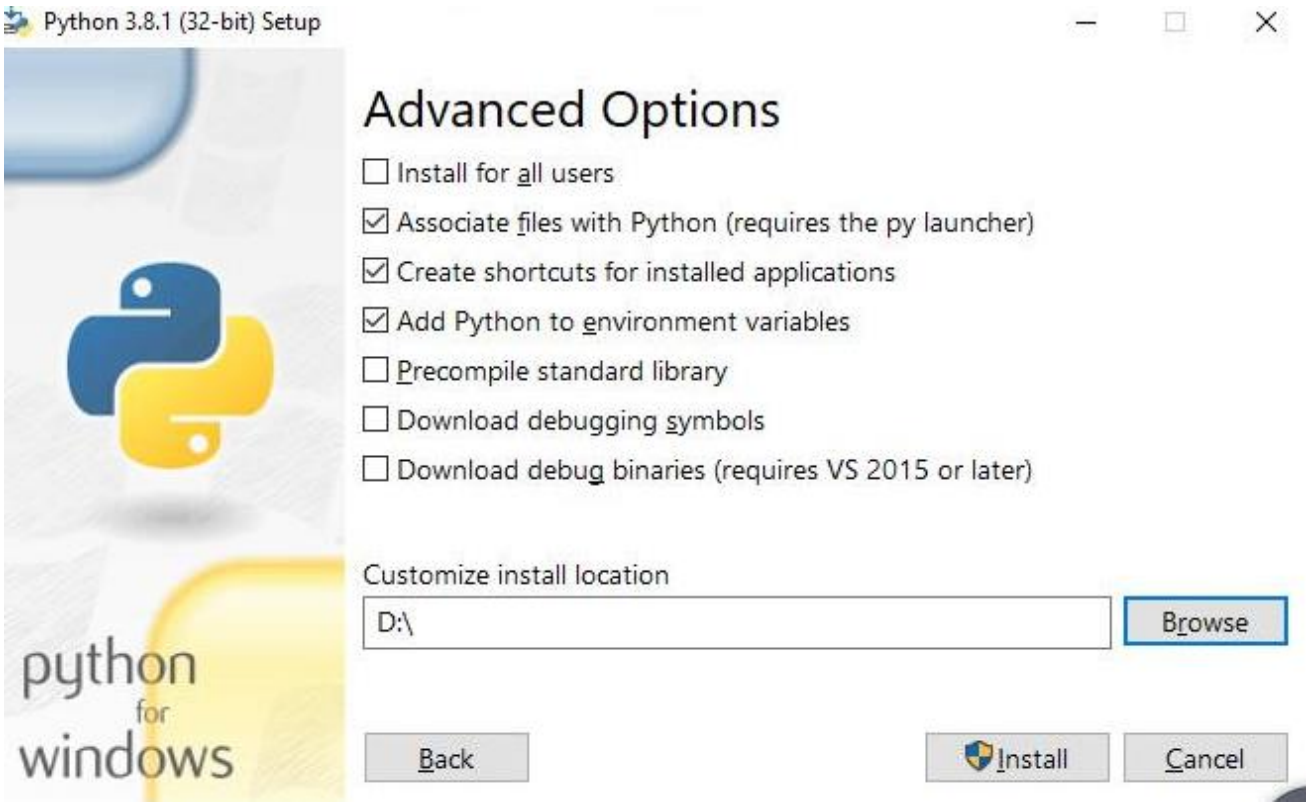
Version	Operating System	Description
Gzipped source tarball	Source release	
XZ compressed source tarball	Source release	
macOS 64-bit installer	Mac OS X	for OS X 10.9 and later
Windows help file	Windows	
Windows x86-64 embeddable zip file	Windows	for AMD64/EM64T/x64
Windows x86-64 executable installer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

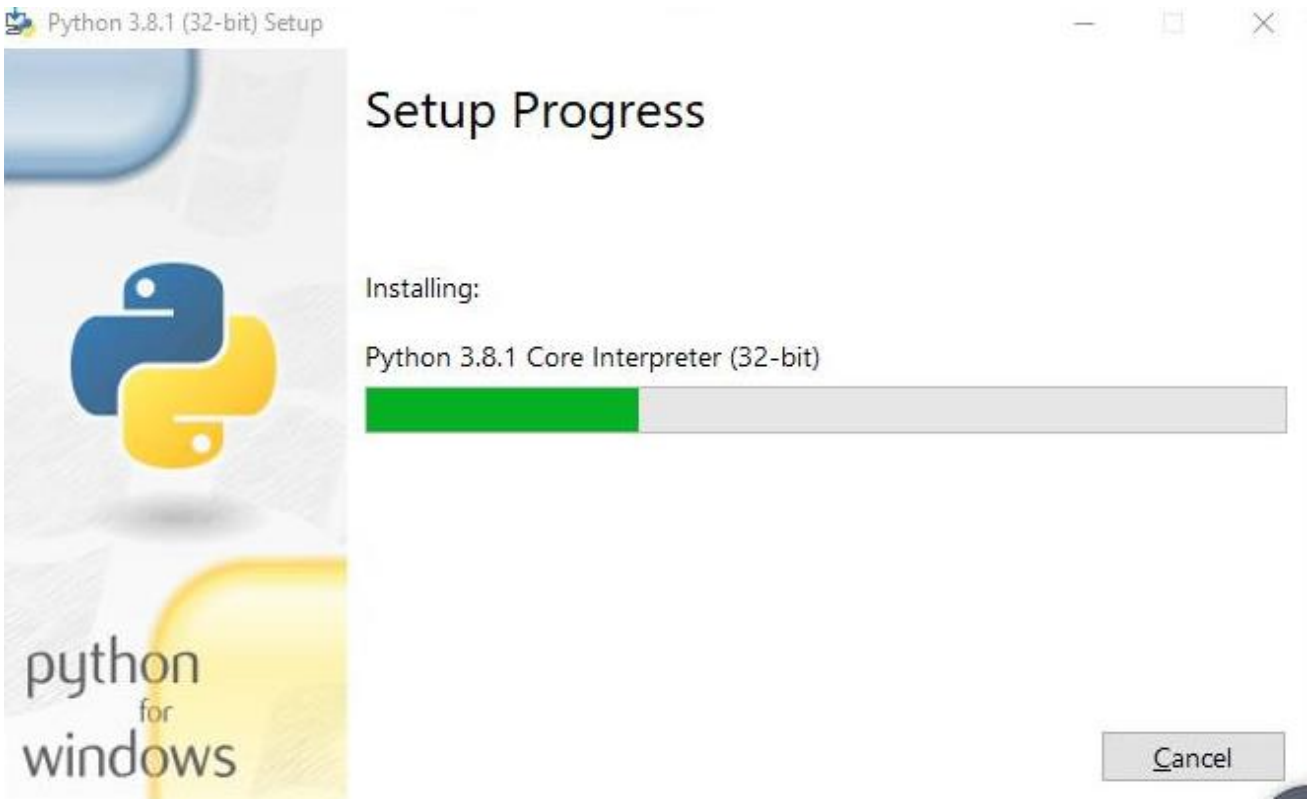
Select "Add Python 3.8 to PATH". And choose Customize installation.



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 installation.

Installation is successful.

Install PyQt5, opencv, numpy and other libraries

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

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 your windows system defaults to python 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 installation is successful, it will prompt “Press any key to continue…” and "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 slow 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:

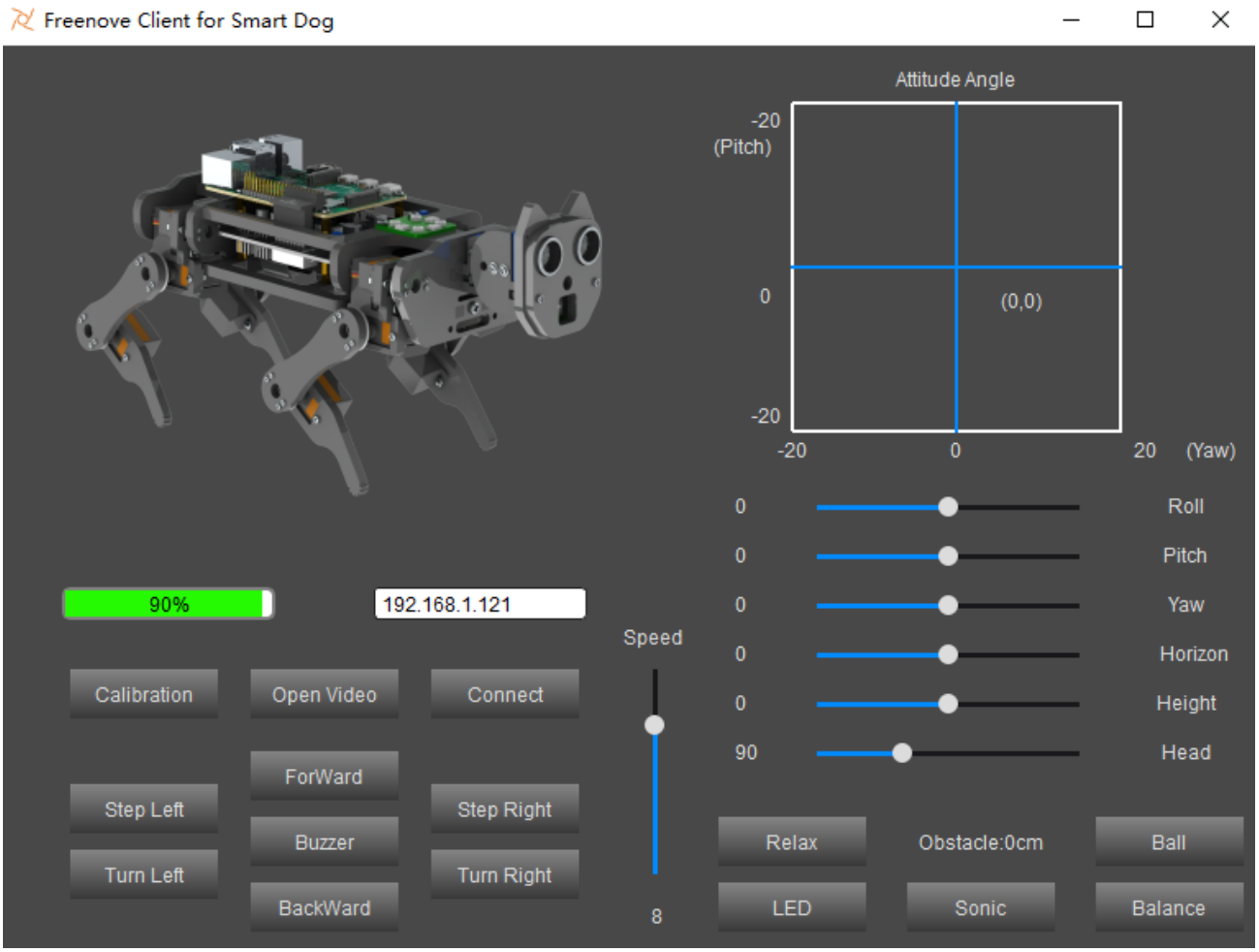
Freenove_Robot_Dog_Kit_for_Raspberry_Pi\Code\Client

Or use python3 to run "Main.py".

Note: If 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 as 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. At this time, 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 complete, 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/

Release version	Release date	
Python 3.8.2	Feb. 24, 2020	Download
Python 3.8.1	Dec. 18, 2019	Download
Python 3.7.6	Dec. 18, 2019	Download
Python 3.6.10	Dec. 18, 2019	Download
Python 3.5.9	Nov. 2, 2019	Download
Python 3.5.8	Oct. 29, 2019	Download

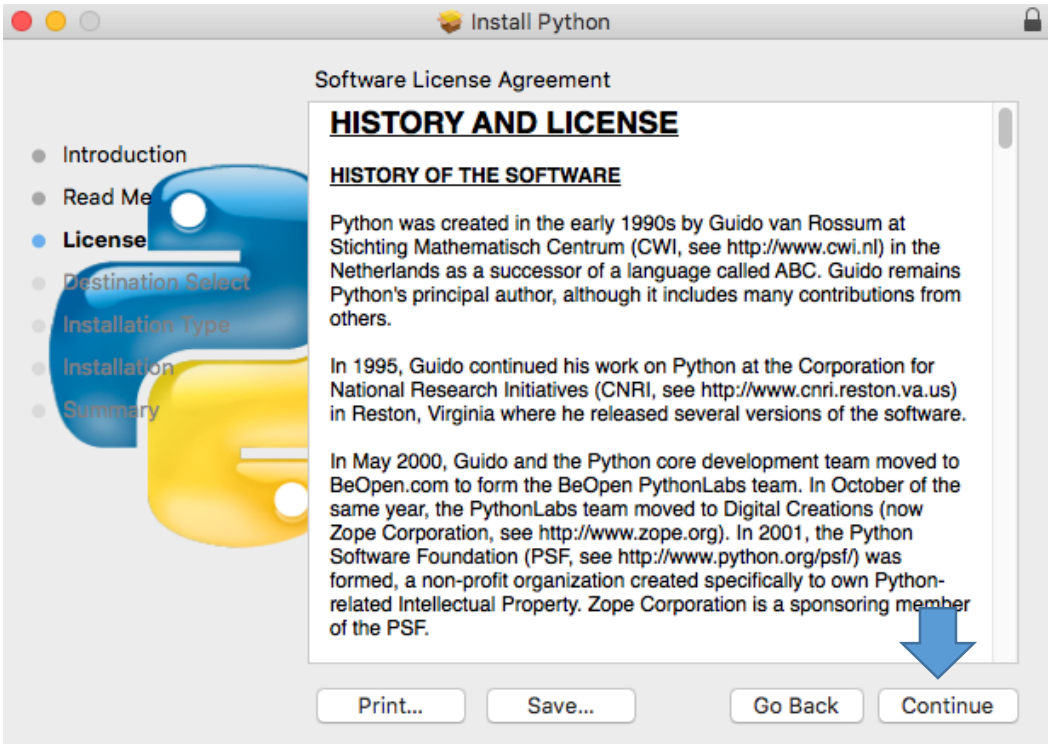
Click Python 3.8.2.

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.

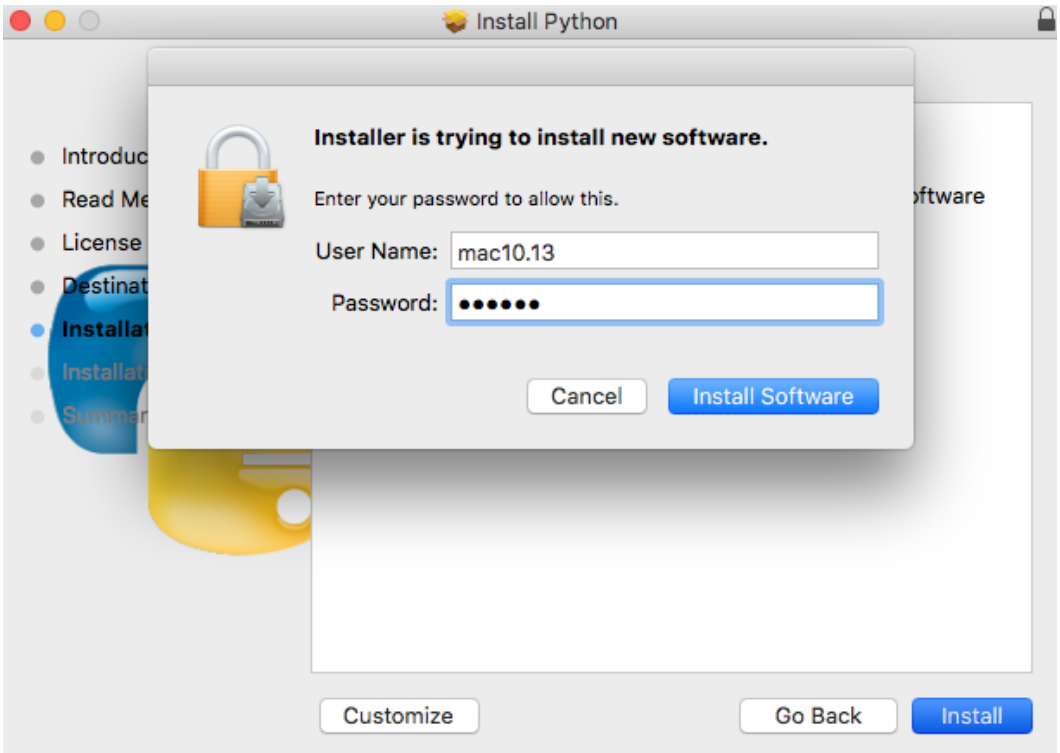
Then install python.

Click Continue.

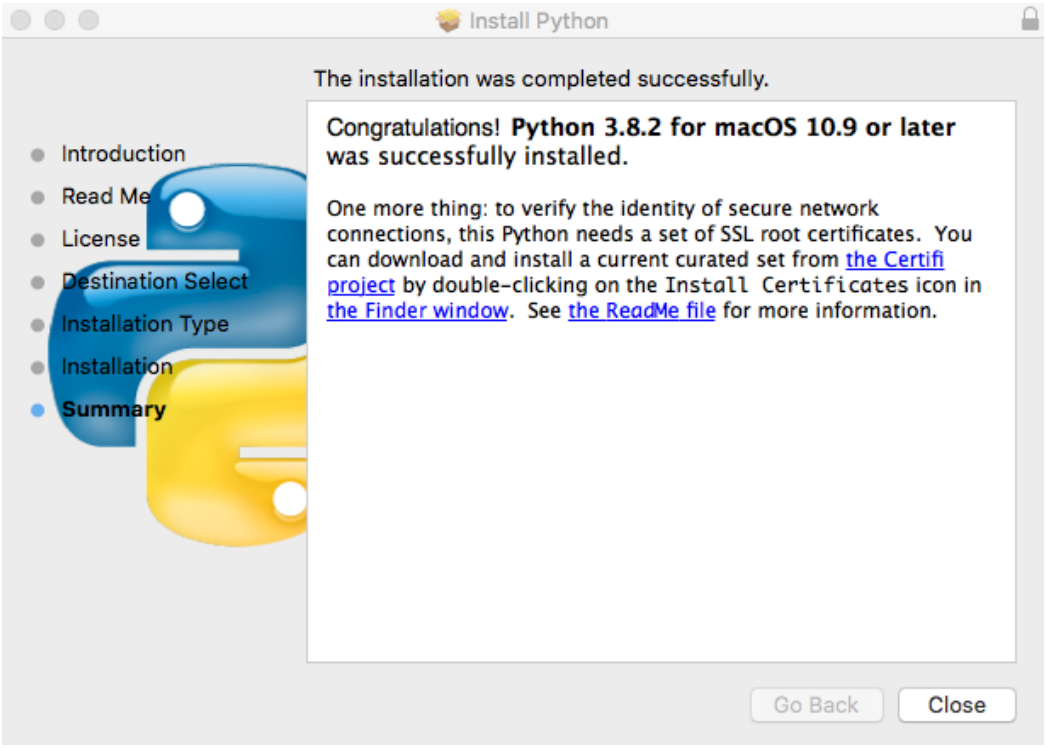


Click Continue.

Click Agree.



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



Now the installation is completed successfully.

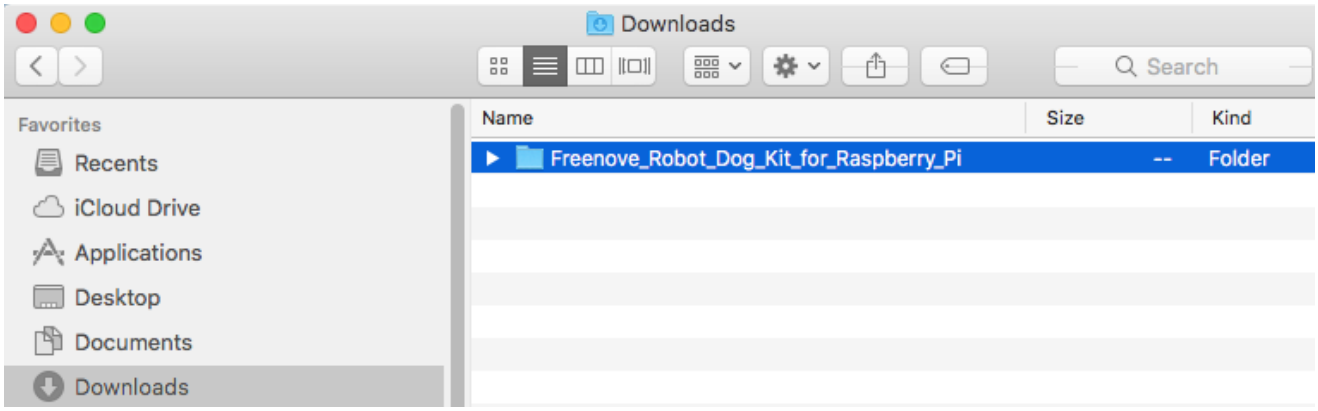
You can find in Applications.

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 downloaded 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 need 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 slow network. You can check the network before installing.

Open client

After installation is complete in previous step, now it is in the directory where setup_macos.py is located.

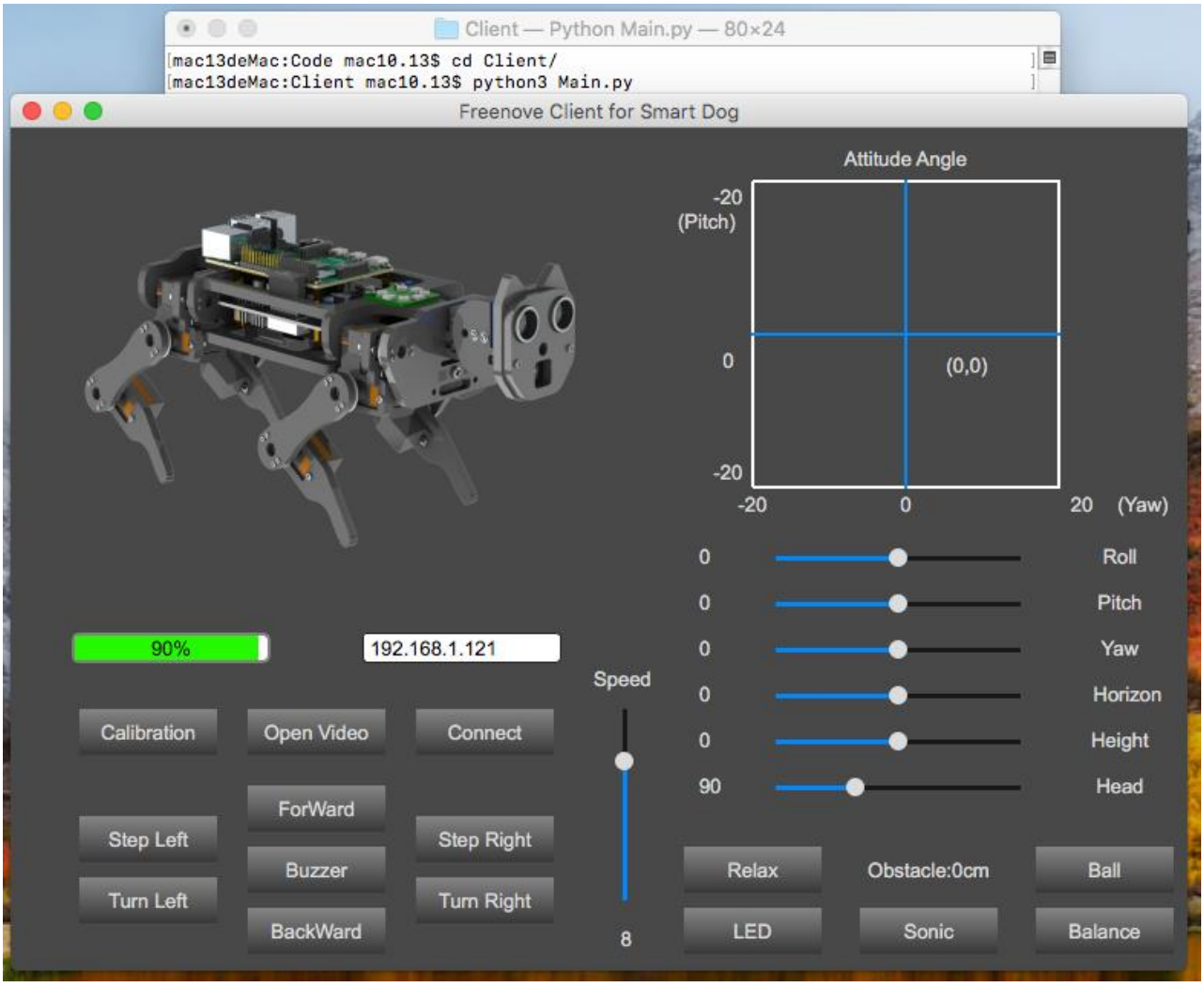
```
Package                 Version
----------------------- --------
numpy                   1.18.1
opencv-python-headless  4.2.0.32
Pillow                  7.0.0
pip                     20.0.2
PyQt5                   5.14.1
PyQt5-sip               12.7.1
setuptools              41.2.0

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, then 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

Open client

Enter the following commands at the terminal.

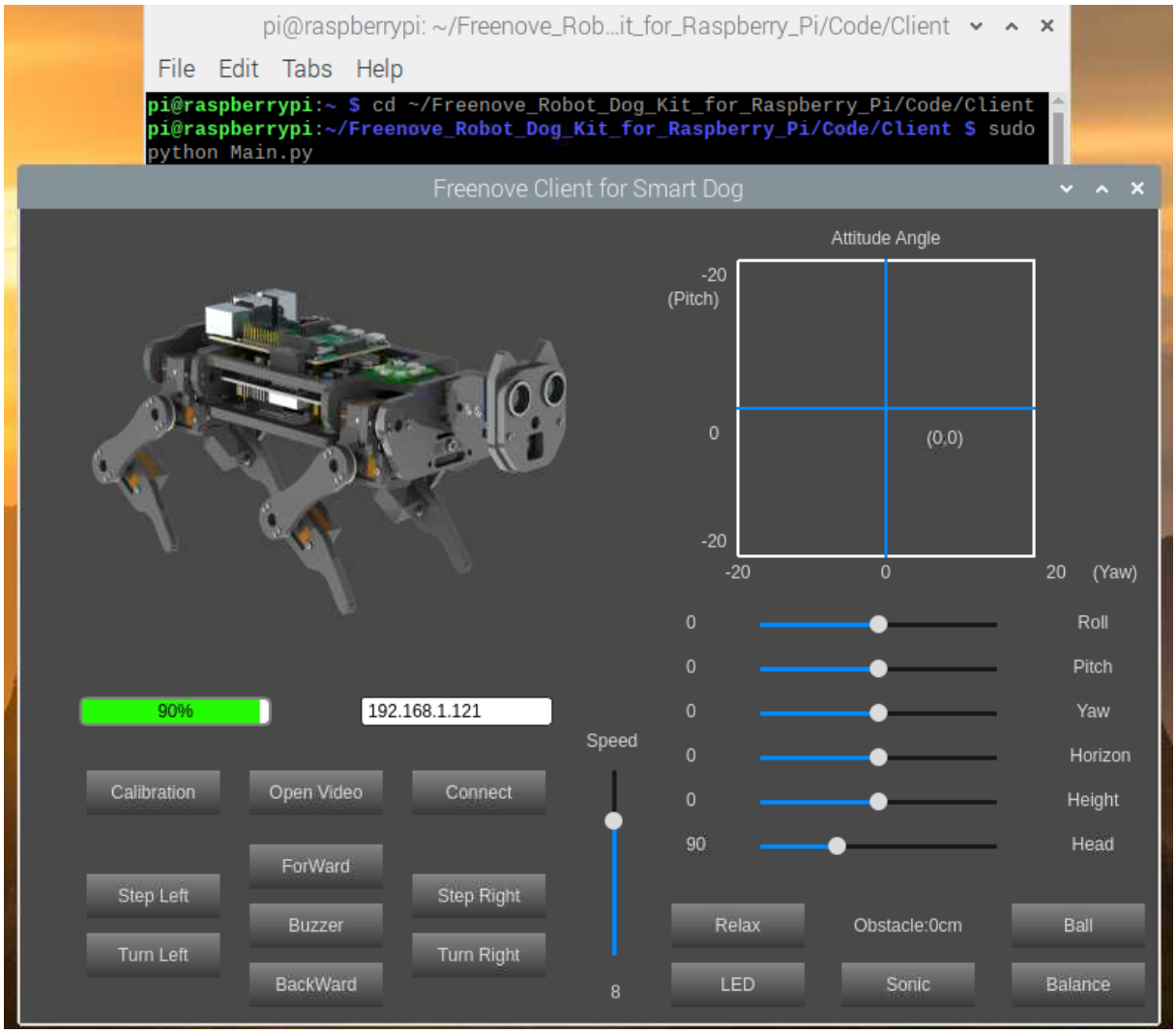
1. Use the cd command to go to the directory where Main.py is located.

```
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 have.



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.

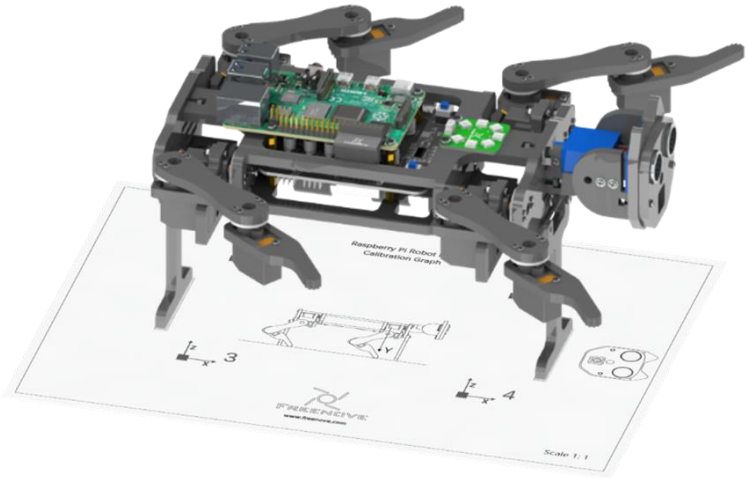
After the connection is successful, you need to calibrate the four legs of the robot dog in Calibration section. After the calibration is completed, then robot dog can be controlled to move.

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

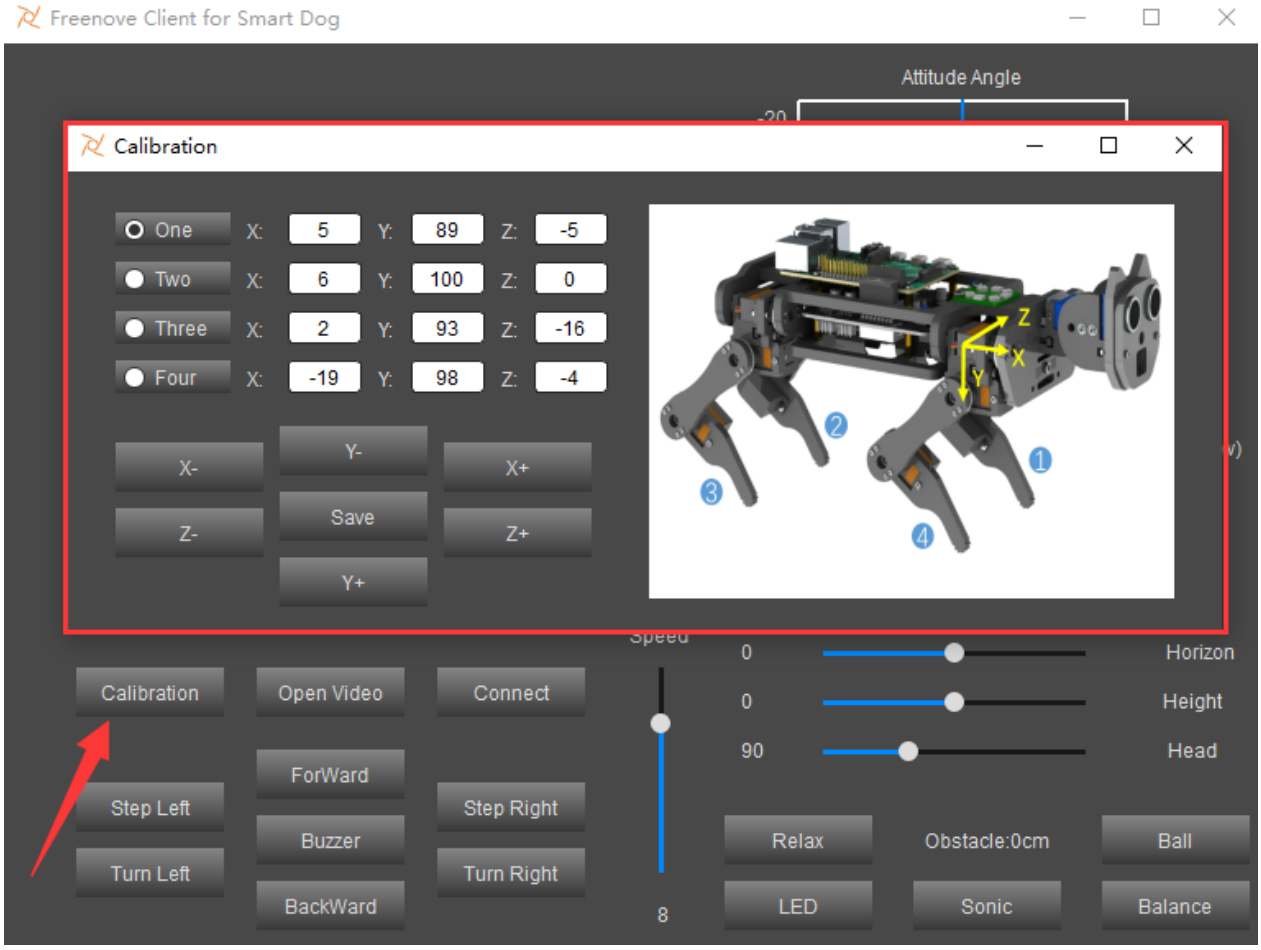
Calibration

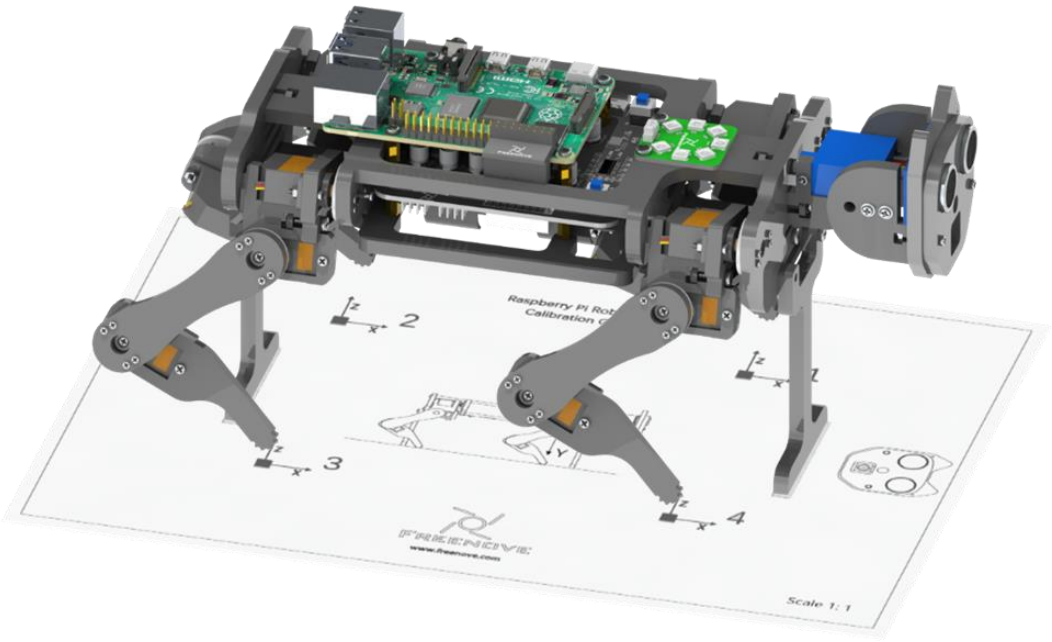
Calibrate the robot.

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

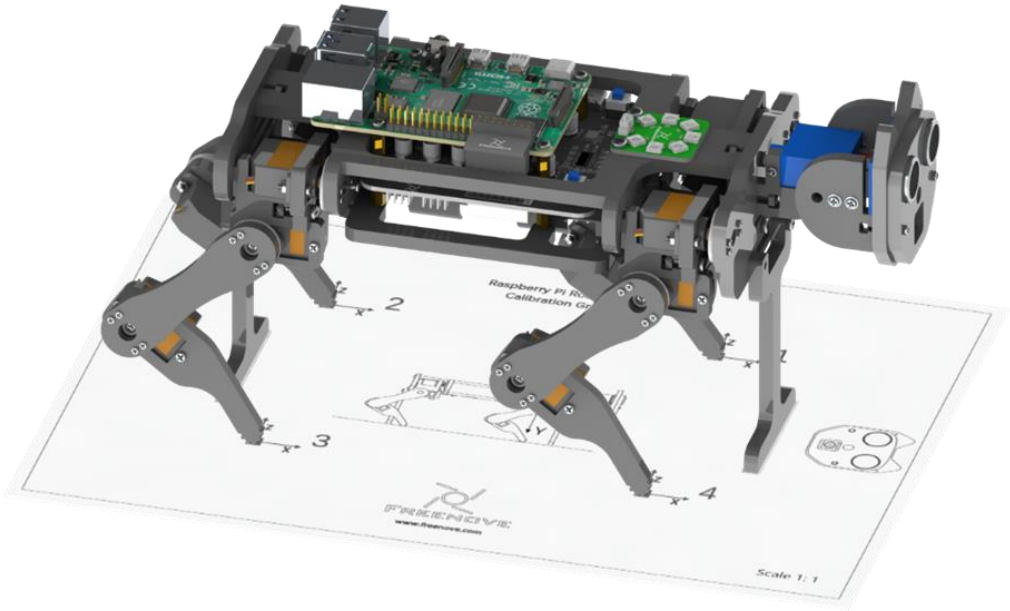


3. Connect client with server successfully, click the "Calibration" button on the client, then a calibration window pops up. And the robot leg will active to following position.





4. Calibrate the four legs to make 4 foot points fall to the corresponding positions. Feel the contact force with the ground during calibration. The later control depends on this step.



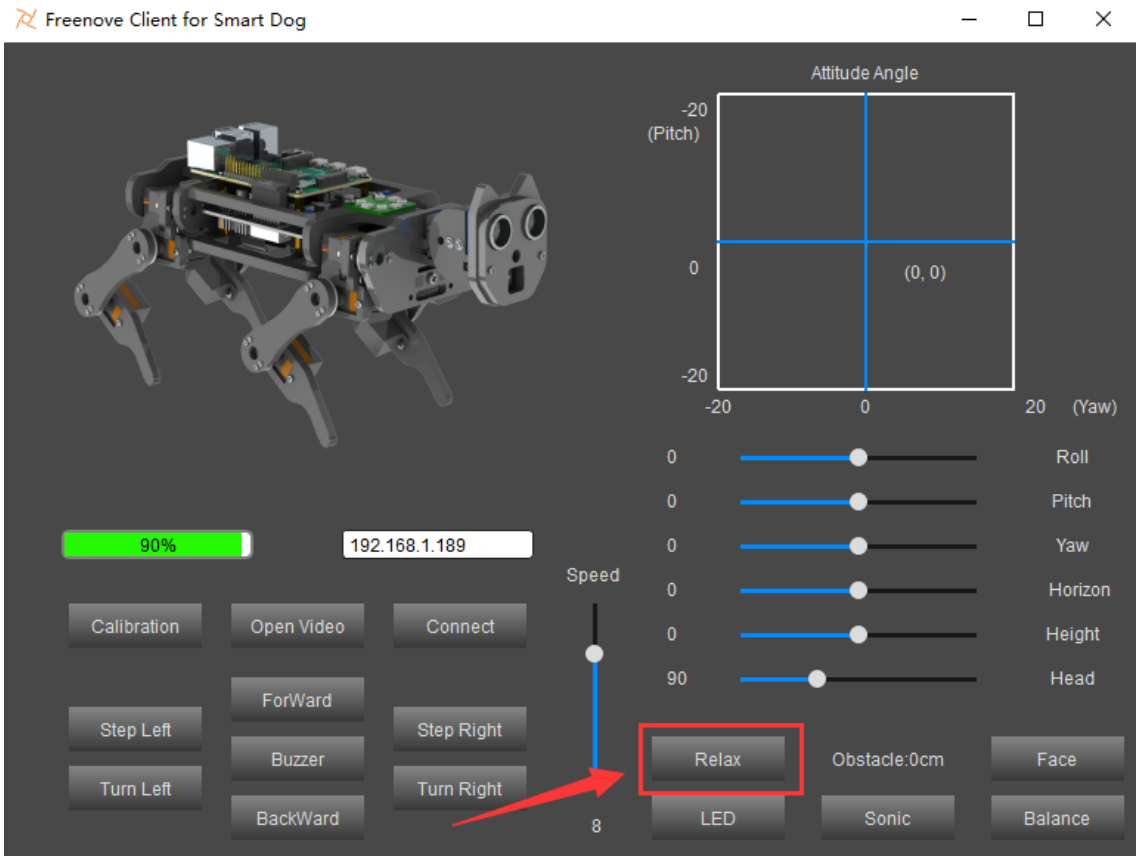
5. After all 4 points have reached the designated position, click "Save" to save. Now calibration is completed.

NOTE:

1. It is best to let the robot dog walk on a **flat hard** floor. The robot dog may not walk well on other roads.
2. **There is a offset when control 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 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 calibration is successfully, you can control the robot dog to move.

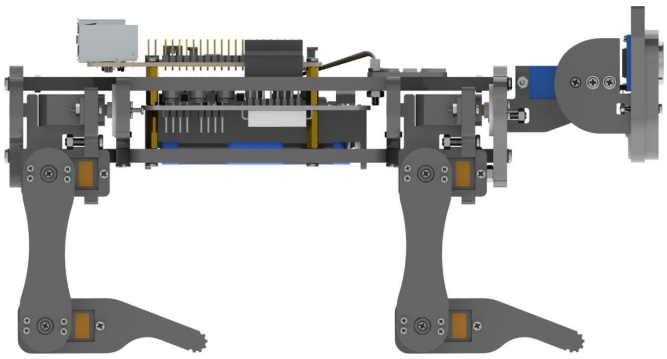


Relax mode.

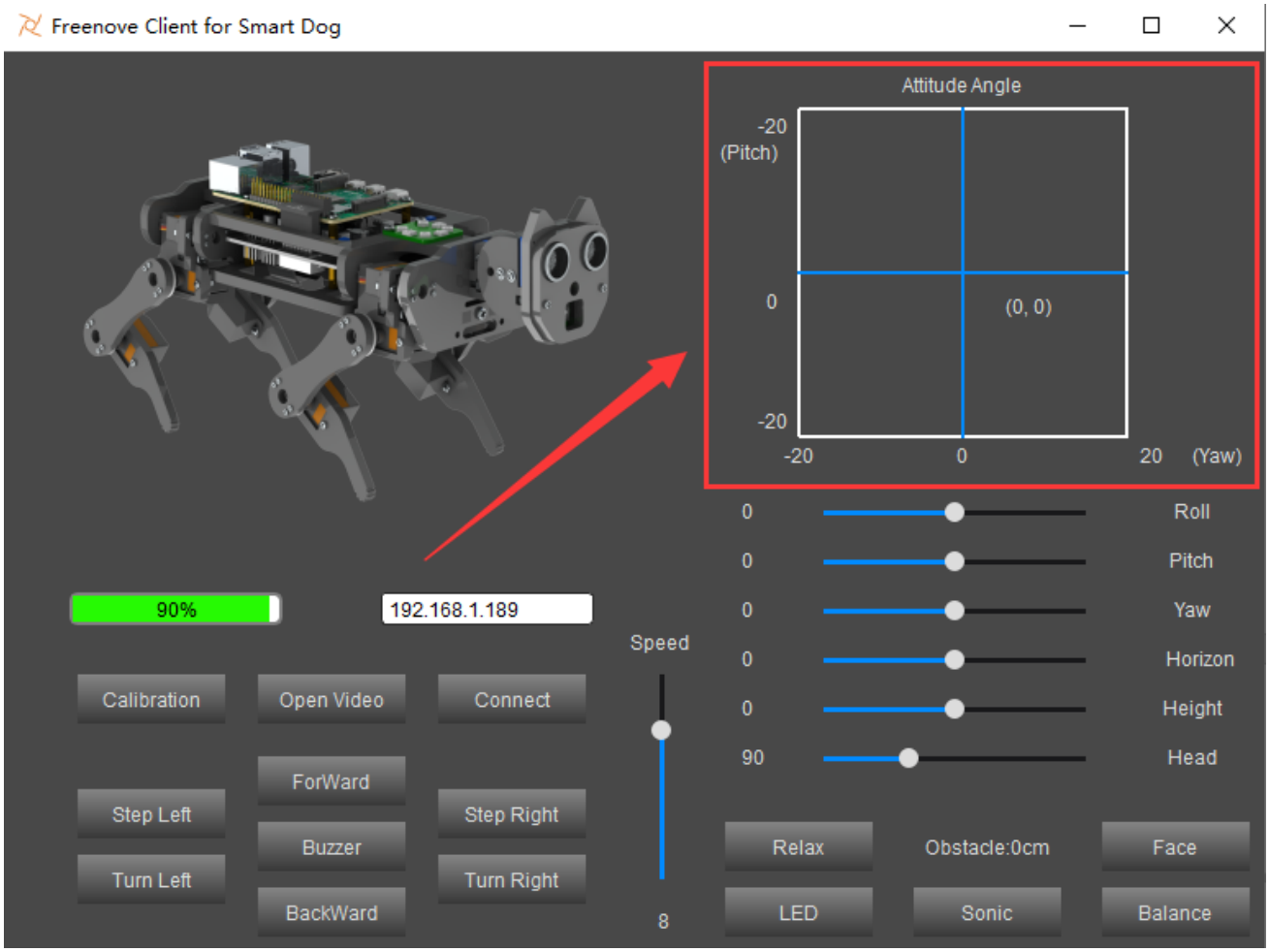
a) When the robot dog moves for 3 minutes in total, it will feel tired (the servo will get hot). In order to protect the servo, the robot will get into relax mode for 1 minute. **During this time, it won't responds to motion command.** You can still use the functions of LED, buzzer, real-time video and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

c) If the robot isn't tired and is standing, when the robot don't receive motion command for 10s, it will get into relax mode. In this situation, it will responds to any 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
Face/Ball/Close	F	Face recognition and ball tracking
Balance	B	Open/Close balance mode

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 to learn.

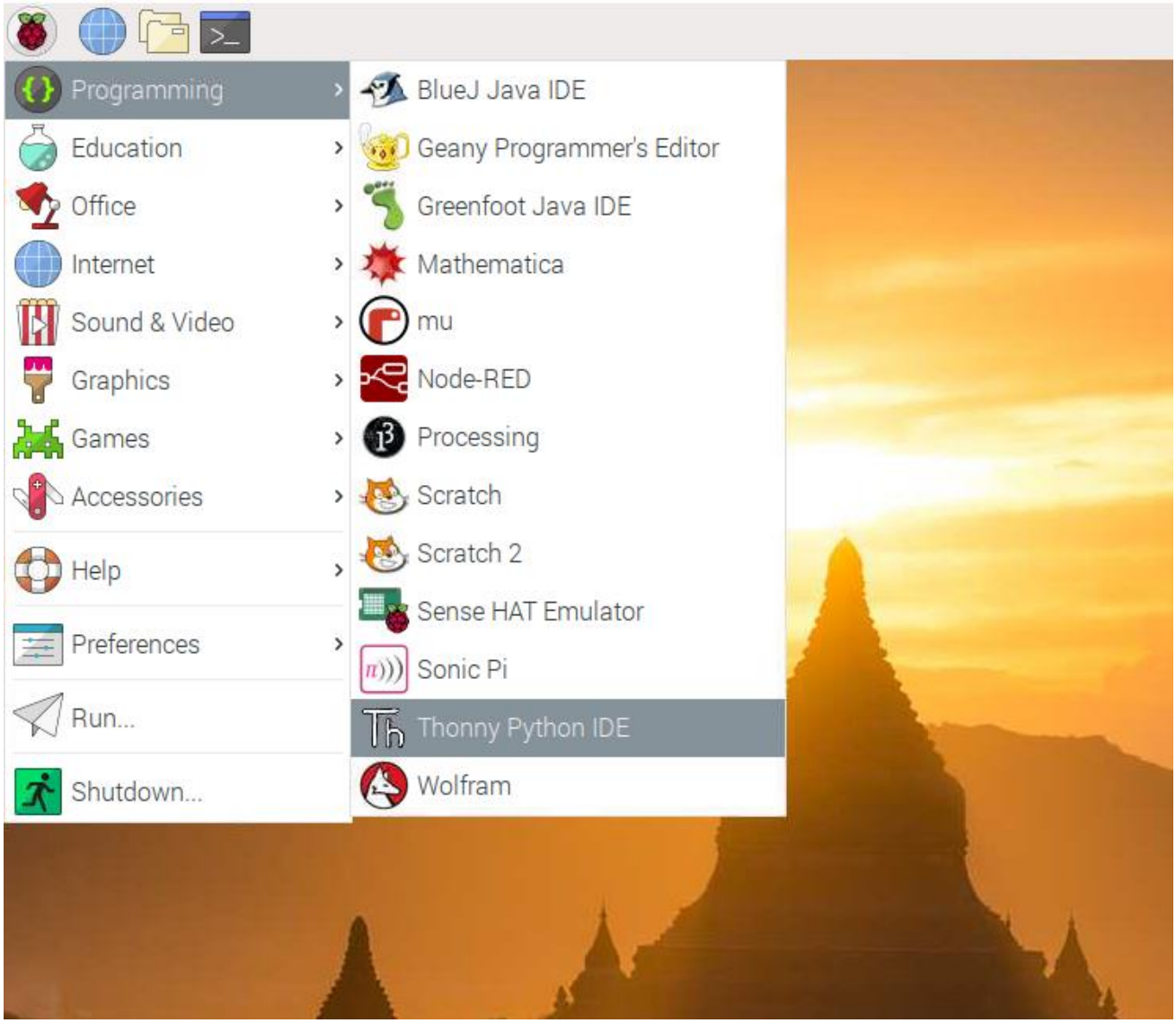
The robot dog program is based on python3. If your python is python2 by default, please change to python3.

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

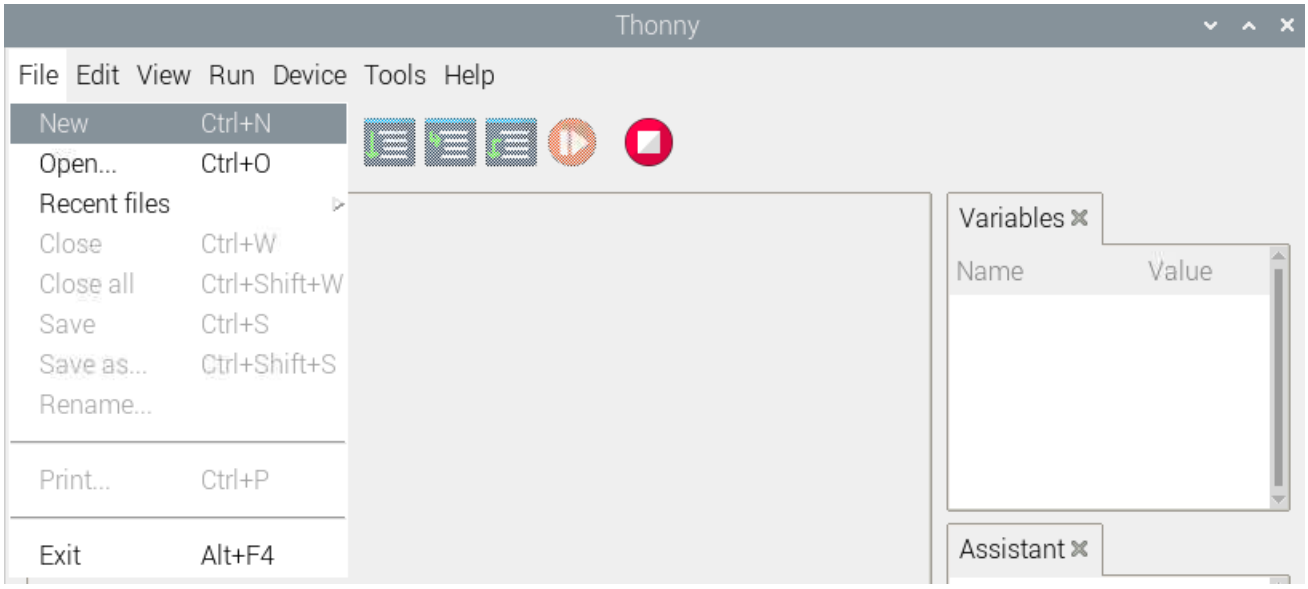
https://python.swaroopch.com/basics.html

Program

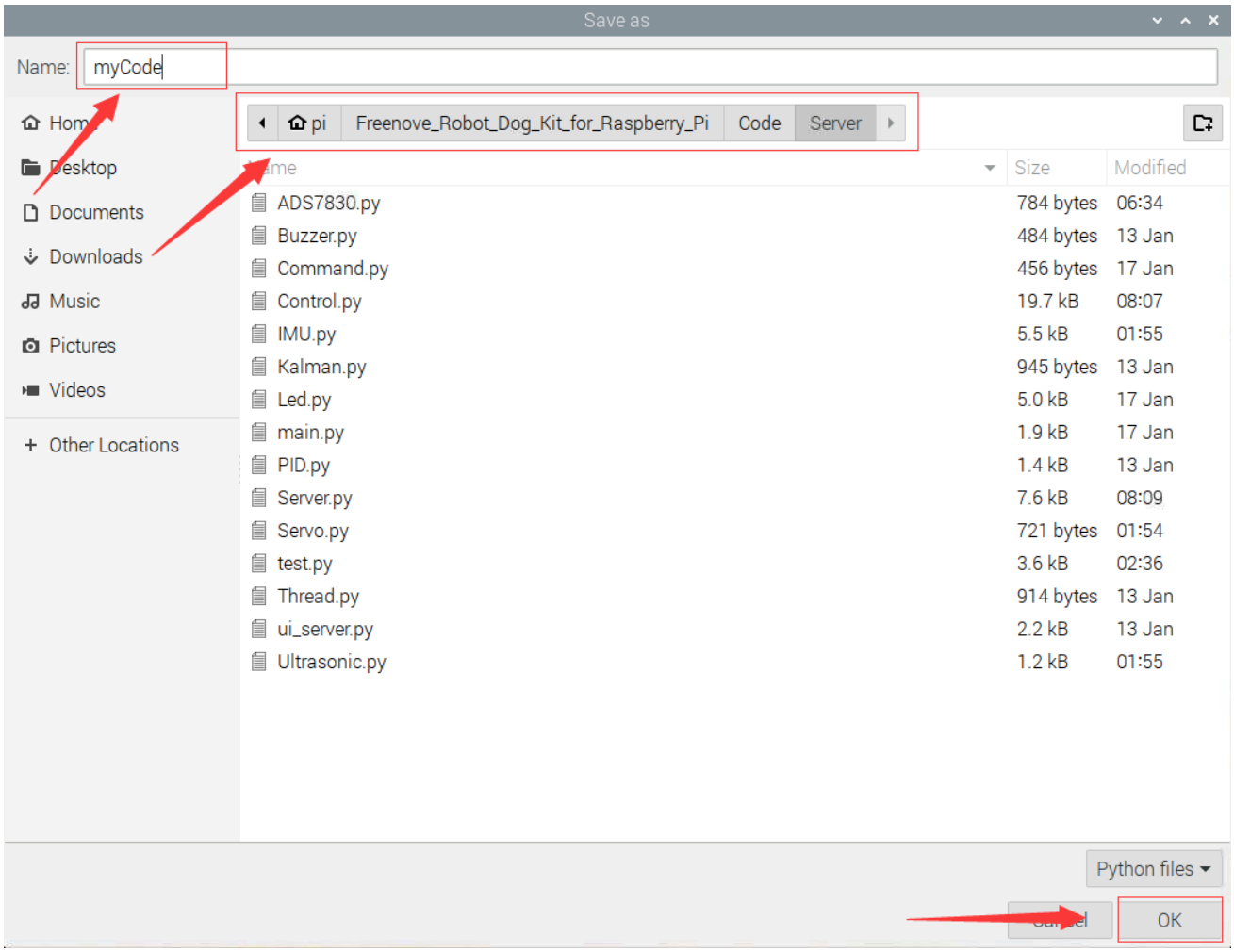
First, open Thonny Python IDE which is easy to use for beginners.



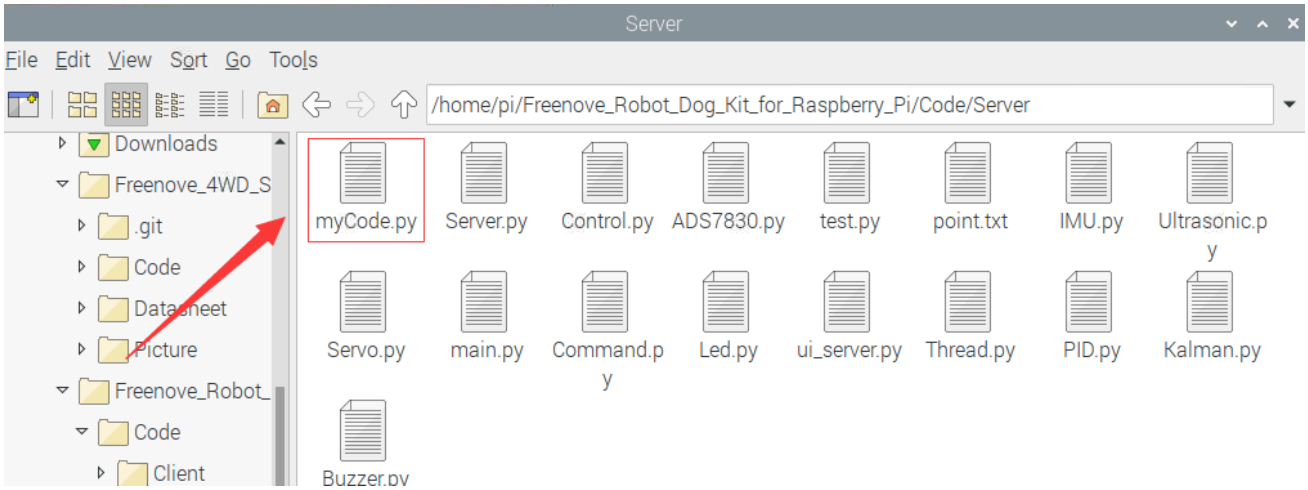
Create a new file.



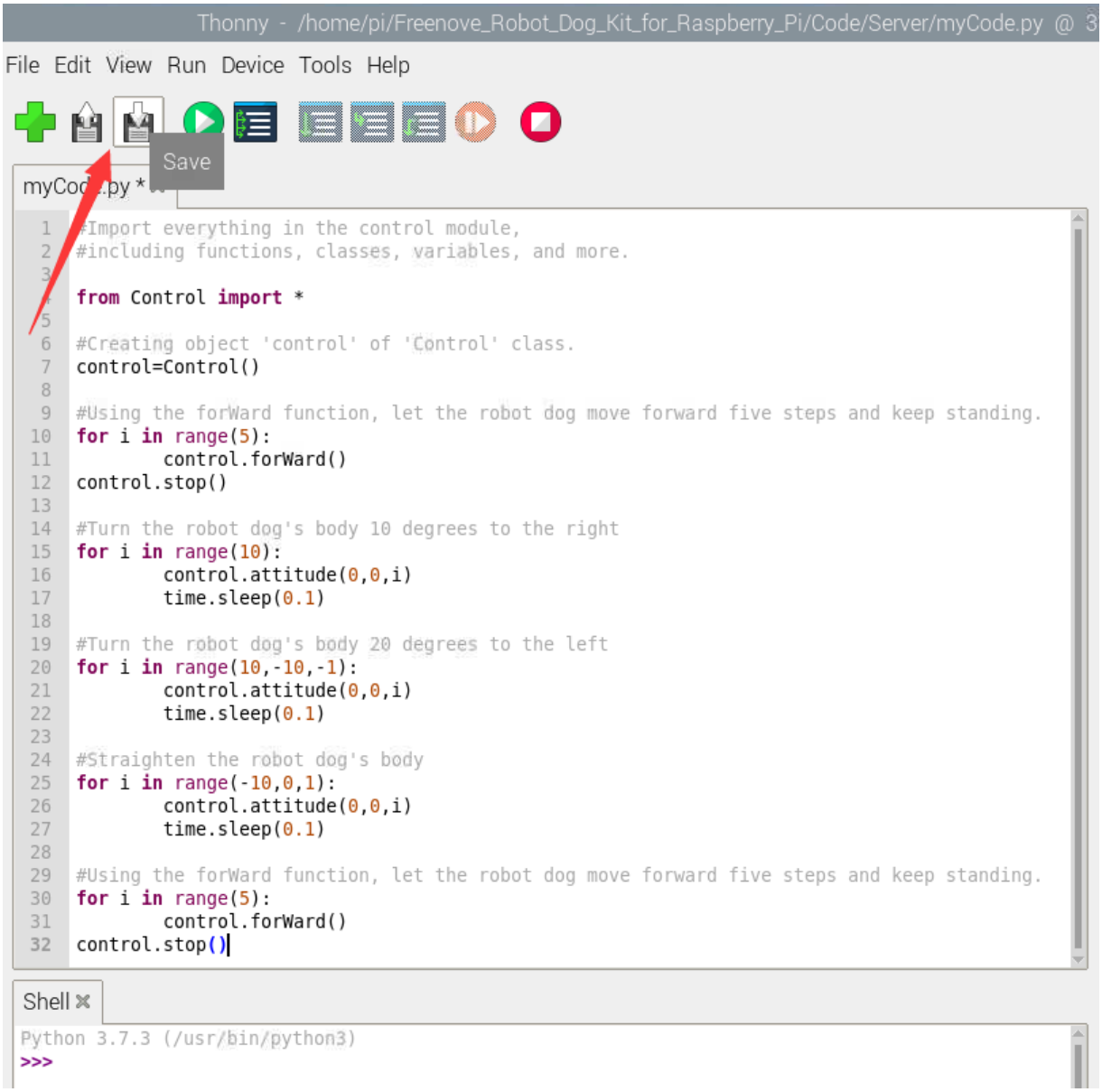
Name it with myCode as an example and save it under Server folder of robot code folder.



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



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

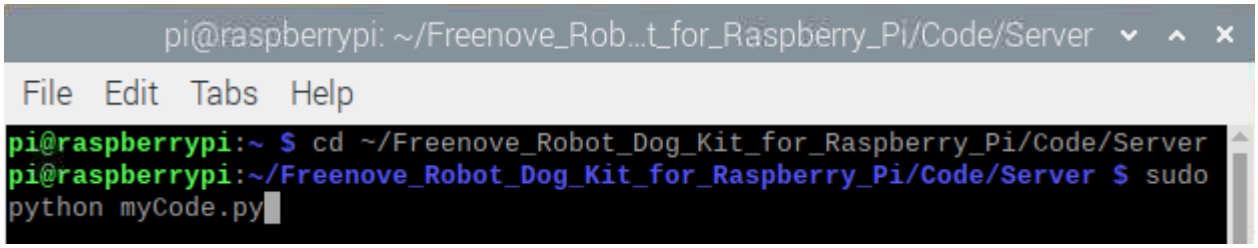


Type the following command to enter the directory where myCode.py is located.

```
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 5 steps, then twists body 10 degrees to the right, then 20 degrees to the left, then returns to the original attitude and move forwad 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

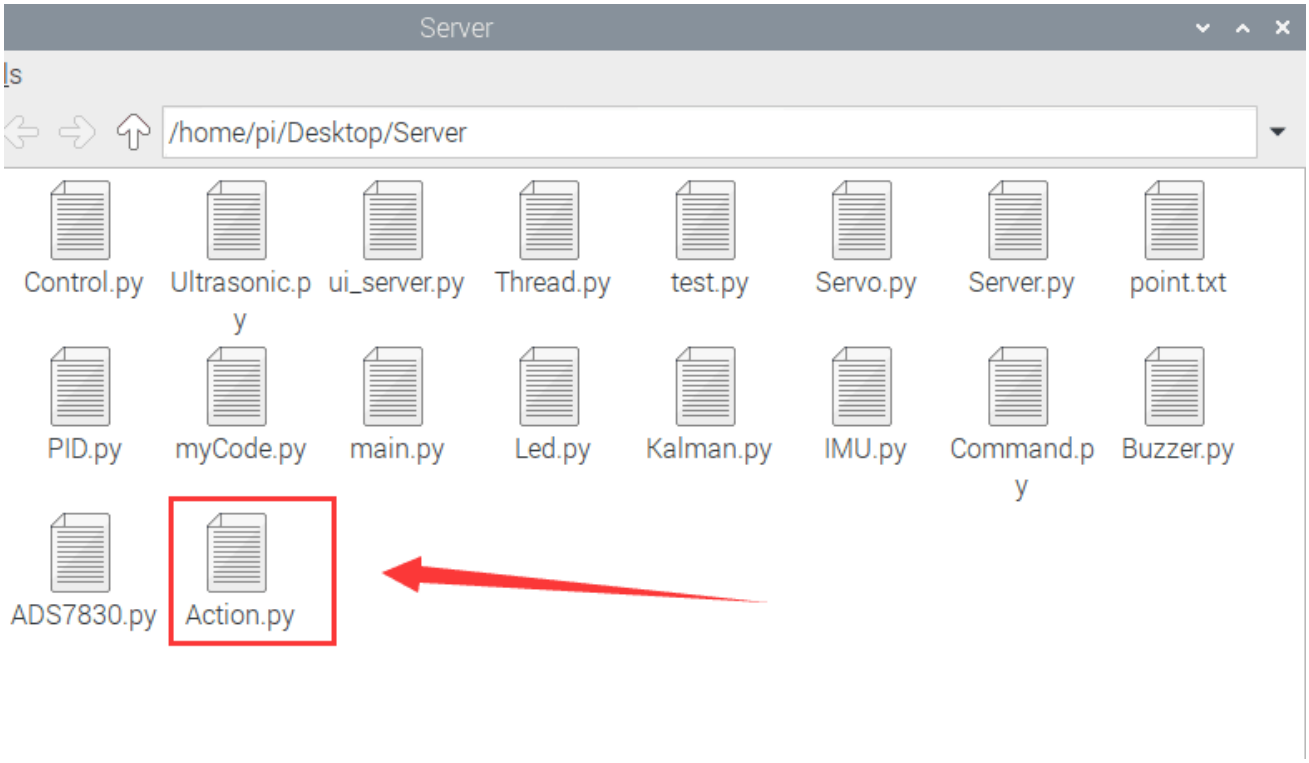
Function	Description
setServoAngle(channel, angle)	Turn the servo of the specified channel to specified angle.

Ultrasonic.py

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

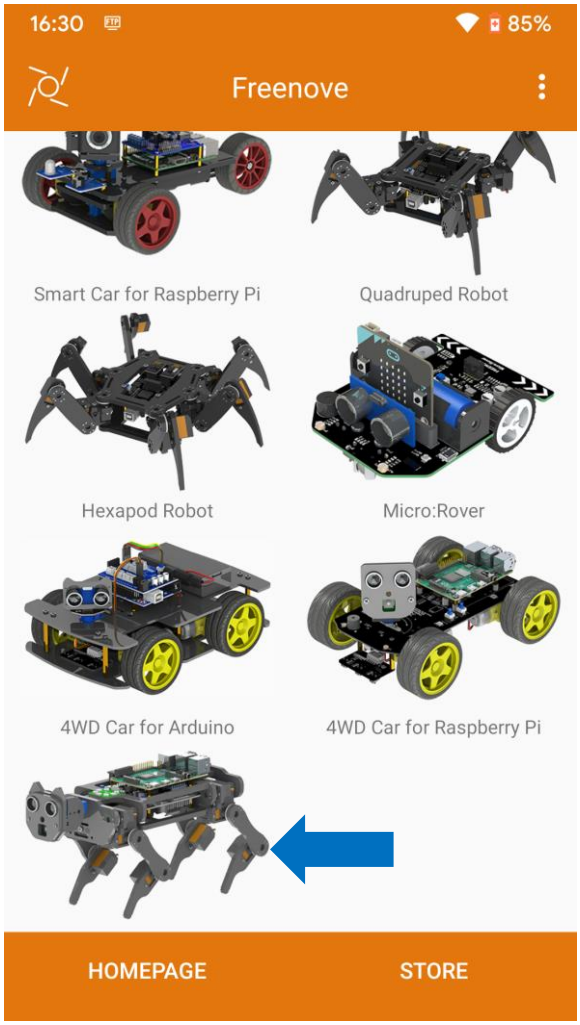
Custom Action

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



Android app

You can download and install the Freenove Android app from below:
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).



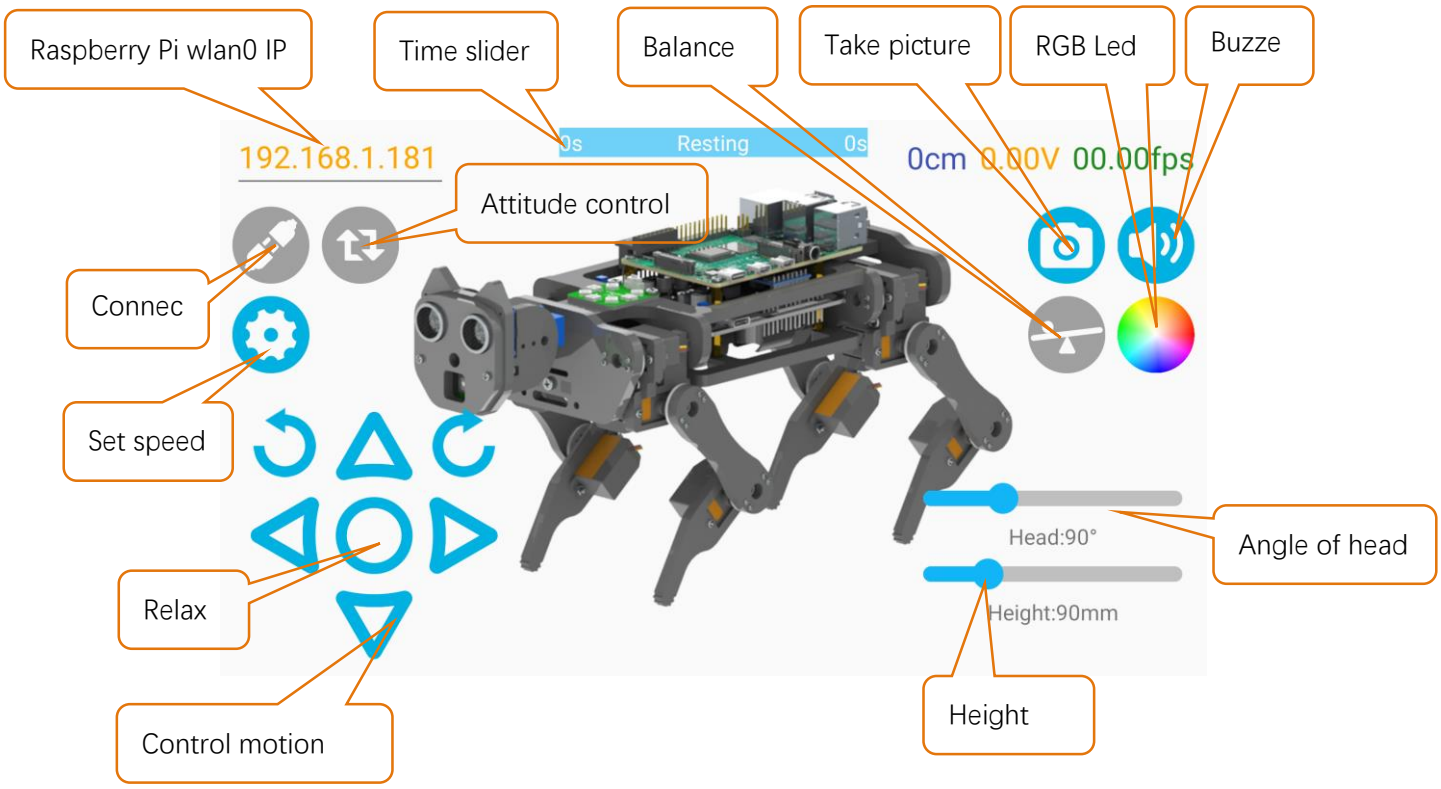
Relax mode.

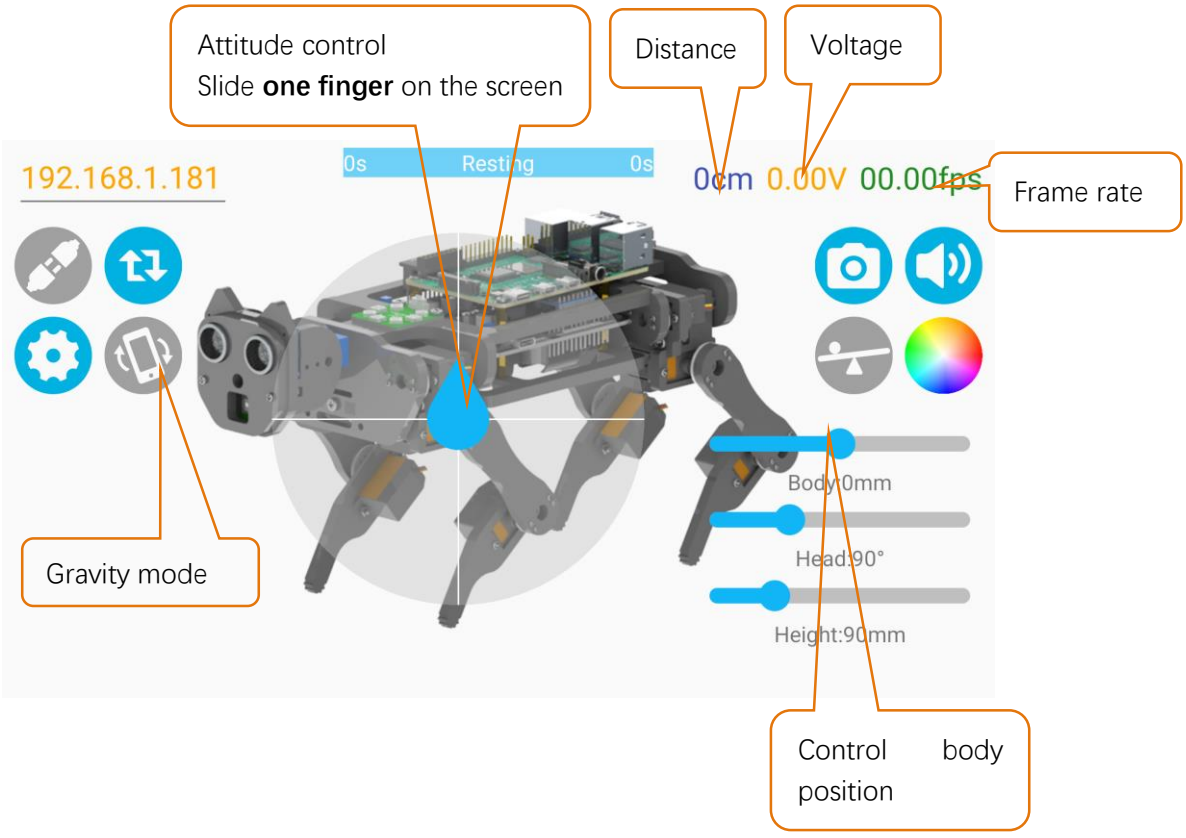
a) When the robot dog moves for 3 minutes in total, it will feel tired (the servo will get hot). In order to protect the servo, the robot will get into relax mode for 1 minute. **During this time, it won't responds to motion command.** You can still use the functions of LED, buzzer, real-time video and 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 moves for 3 minutes again.
c) If the robot isn't tired and is standing, when the robot don't receive motion command for 10s, it will get into relax mode. In this situation, it will responds to any commands.

Following are the features of his app.

First, you need 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?

Thanks for your reading.

This book is all over here. If you find any mistakes, missions or you have other ideas and questions about contents of this book or the kit and ect., please feel free to contact us at support@freenove.com and we will check and correct it as soon as possible.

After completing the contents in this book, you can try to reform this smart car, such as purchasing and installing other Freenove electronic modules, or improving the code to achieve different functions. We will also try our best to add more functions and update the code on our GitHub (https://github.com/freenove).

If you want to learn more about Arduino, Raspberry Pi, smart cars, robots and other interesting products in science and technology, please continue to focus on our website (www.freenove.com). We will continue to launch cost-effective, innovative and exciting products.

Thank you again for choosing Freenove products.