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_Big_Hexapod_Robot_Kit_for_Raspberry_Pi/archive/master.zip

Get Support and Offer Input

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

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

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

support@freenove.com

Safety and Precautions

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

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

About Freenove

Freenove provides open source electronic products and services worldwide.

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

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

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

http://www.freenove.com

Copyright

FREENOVE®

Free your innovation

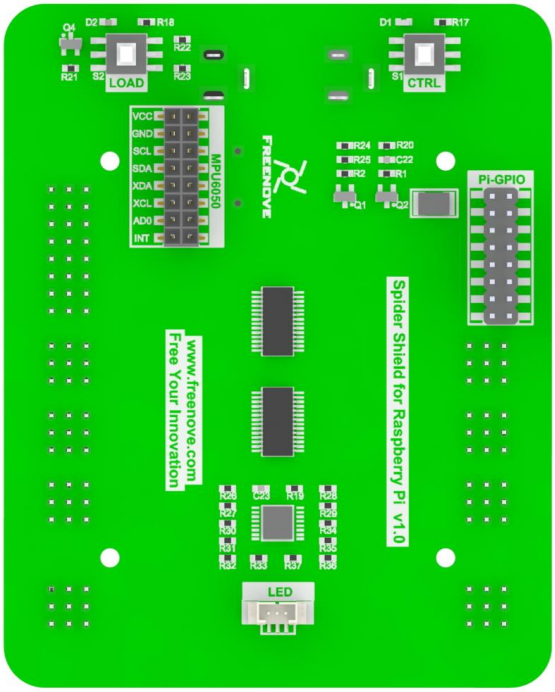
Raspberry Pi® is a trademark of Raspberry Pi Foundation (https://www.raspberrypi.org/).

Contents

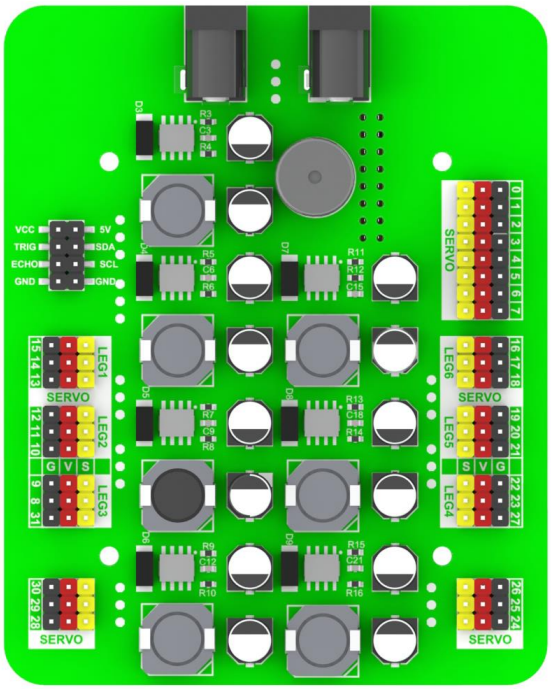
List

Spider Shield for Raspberry Pi

Top



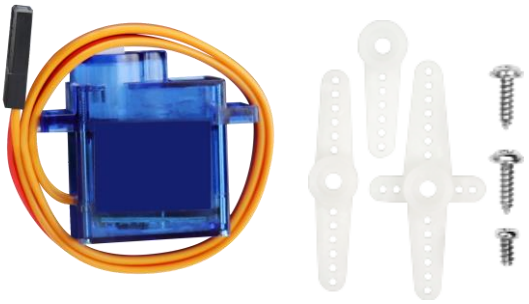
Bottom



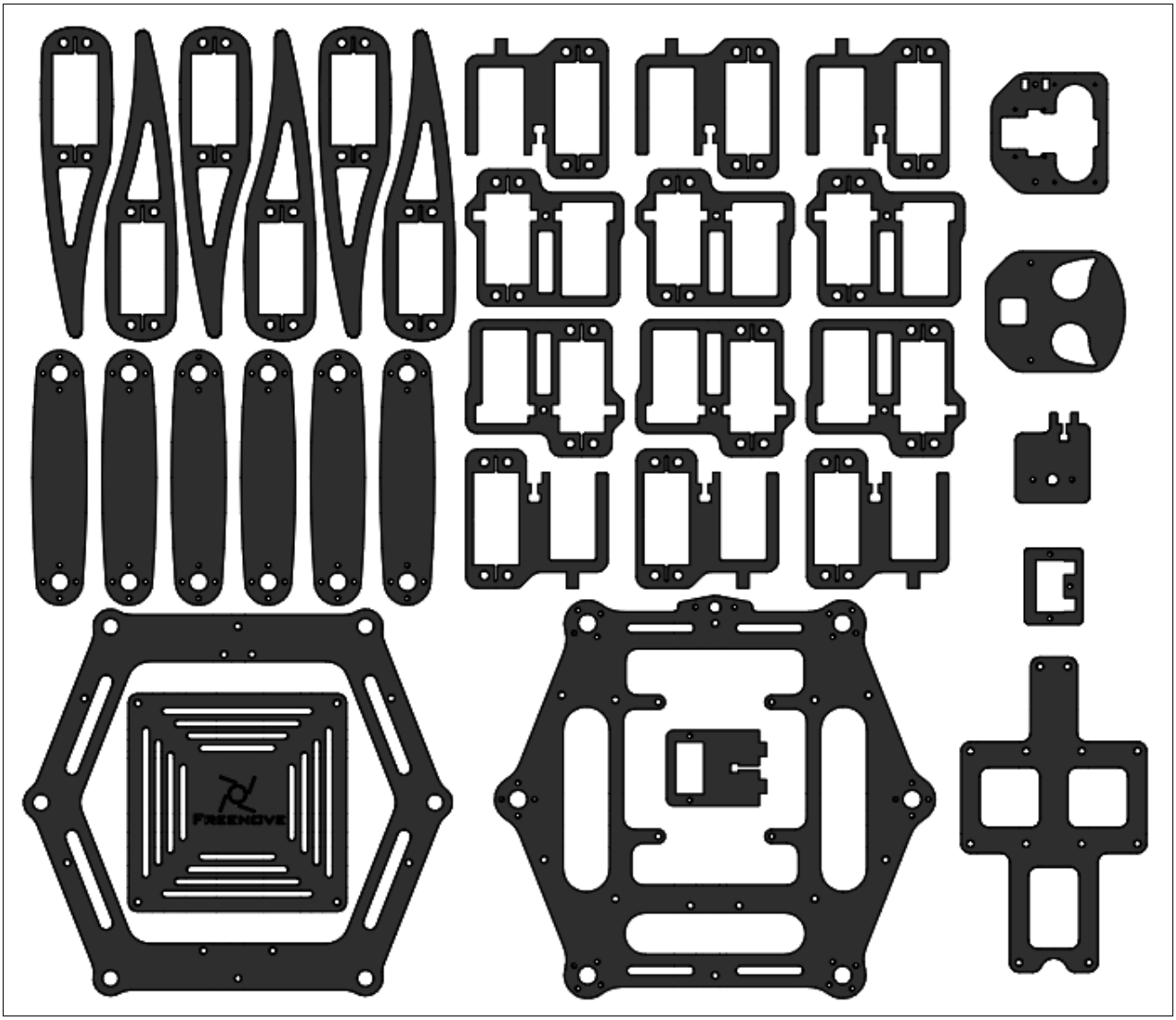
Machinery Parts

M3*55 Copper Standoff x8 Freenove 	M3*30 Copper Standoff x6 Freenove 	M3*20 Copper Standoff x6 Freenove 	M2.5*8 Copper Standoff x6 Freenove 	M1.8*8 Screw x80 Freenove 	M1.4*6 Screw x12 Freenove 
M4*14 Screw x74 Freenove 	M3*12 Screw x40 Freenove 	M2.5*8 Screw x10 Freenove 	M2*16 Screw x8 Freenove 	M3*10 Sunk Screw x10 Freenove 	M4 Non-slip nuts x74 Freenove 
M3 Nut x20 Freenove 	M2 Nut x8 Freenove 				

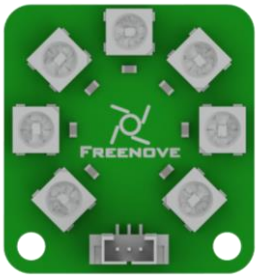
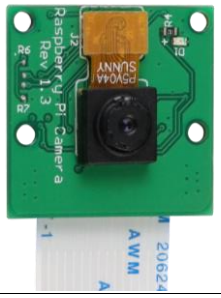
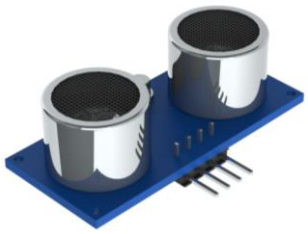
Transmission Parts

MG996R servo package x18	S90 servo package x2
	

Acrylic Parts



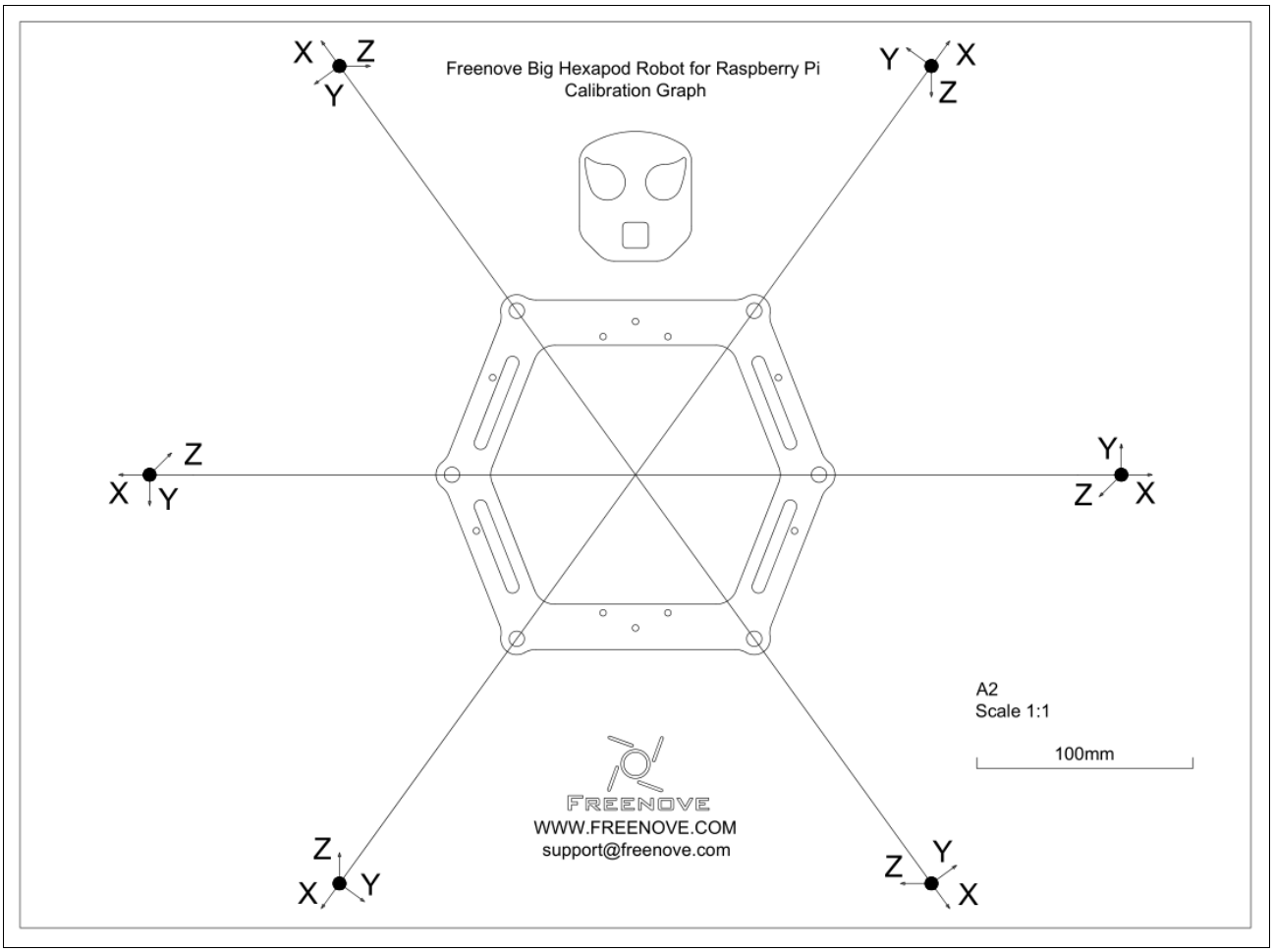
Electronic Parts

LED module x1 	Camera x1 	HC-SR04 Ultrasonic Module x1 	Connection board x1 
Jumper Wire F/F(4) x1 			
1.25mm 3Pin LED cable (same direction) x1 			
FPC 25cm 15Pin camera cable (reversed direction) x1 			
Battery holder x2 			

Tools

Cross screwdriver (2mm) x1 	Cross screwdriver (4.5mm) x1 
Cable Tidy x150cm 	Multifunctional Spanner x1 

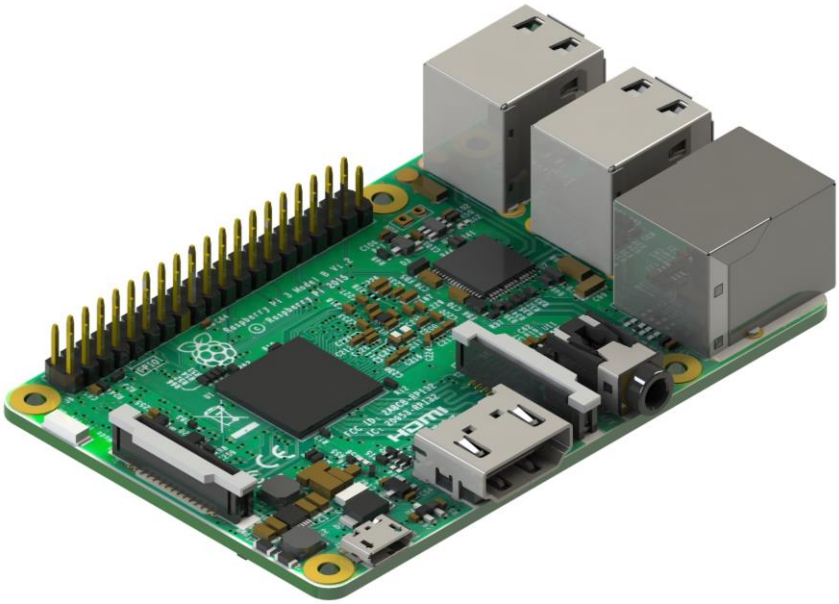
Calibration Graph



Required but NOT Contained Parts

Four 18650 lithium batteries without protection board. Discharge current is greater than 15A

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



Preface

Welcome to use this robot. Following this tutorial, you can make a very cool robot with many functions.

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

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

support@freenove.com

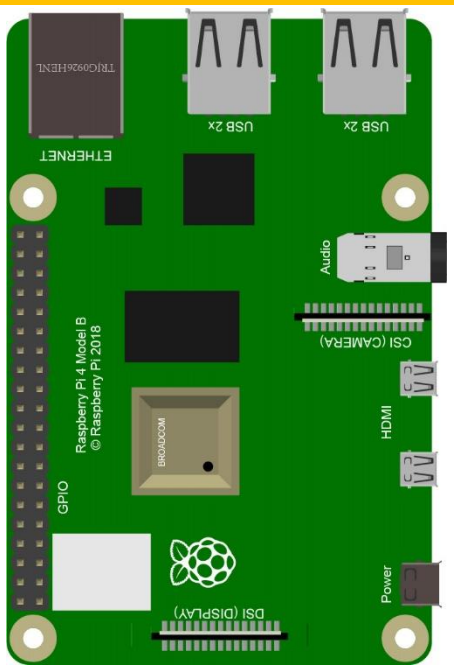
The contents in this book can help enthusiasts with little technical knowledge to make a robot. If you are very interested in Raspberry Pi, and want to learn how to program and build the circuit, please visit our website www.freenove.com or contact us to buy the kits designed for beginners:
Freenove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

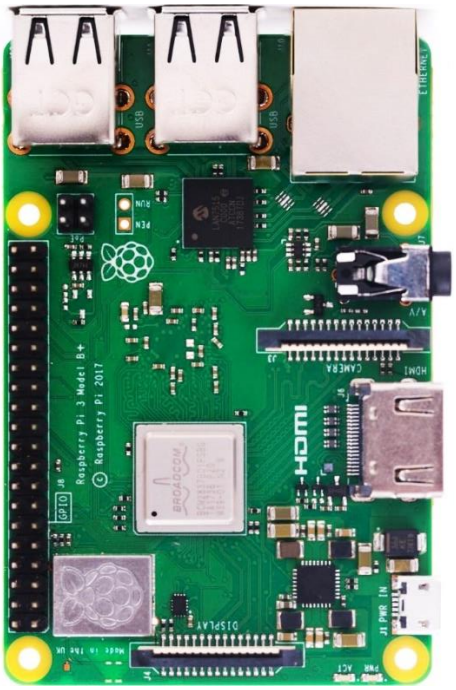
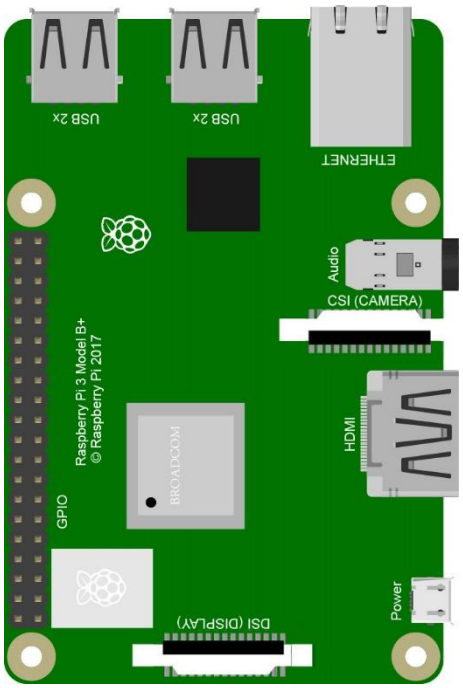
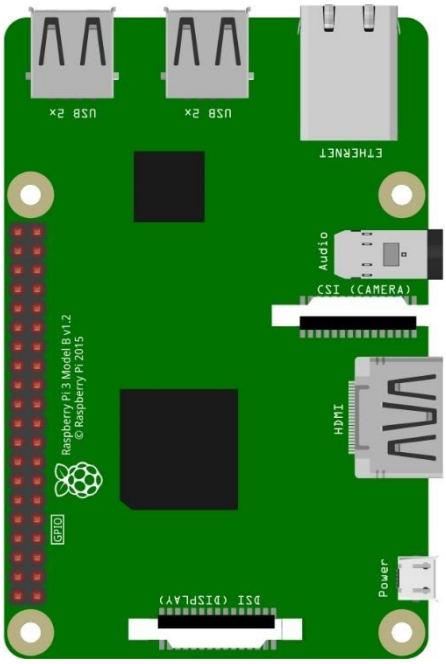
Raspberry Pi

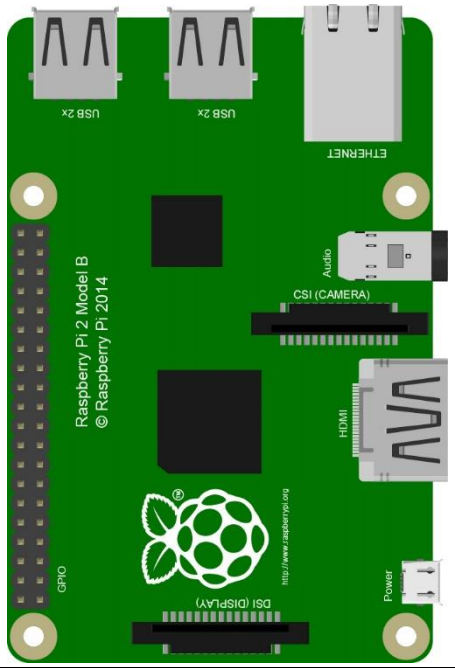
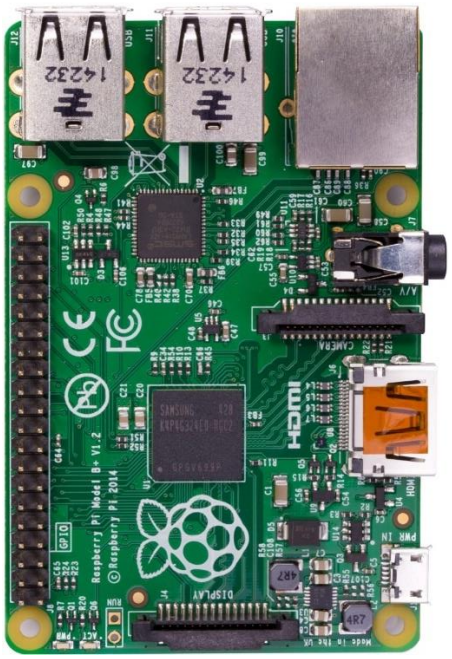
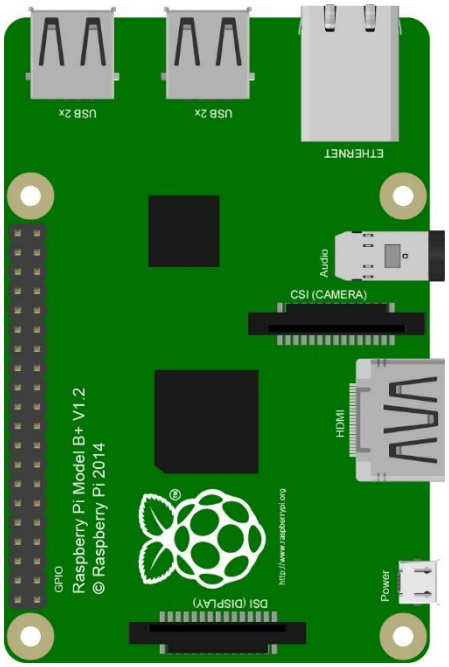
So far, at this writing, Raspberry Pi has advanced to its fourth generation product offering. Version changes are accompanied by increases in upgrades in hardware and capabilities.

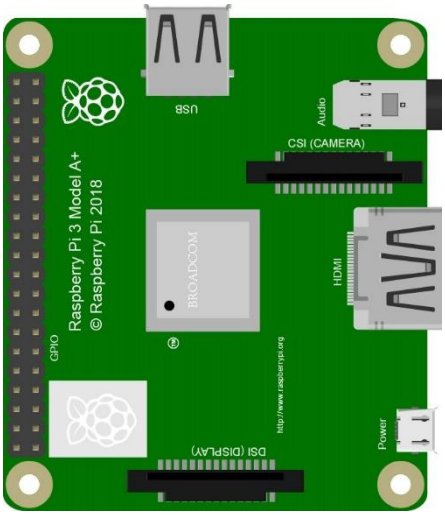
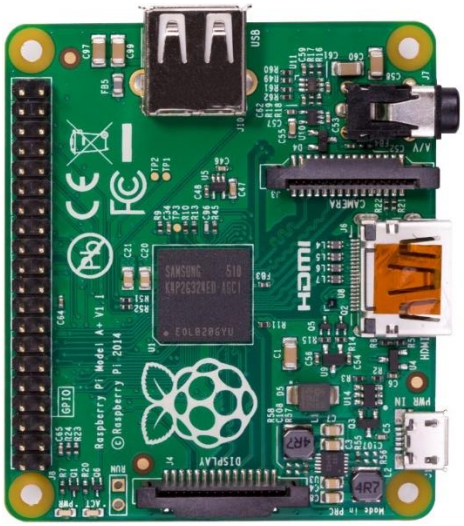
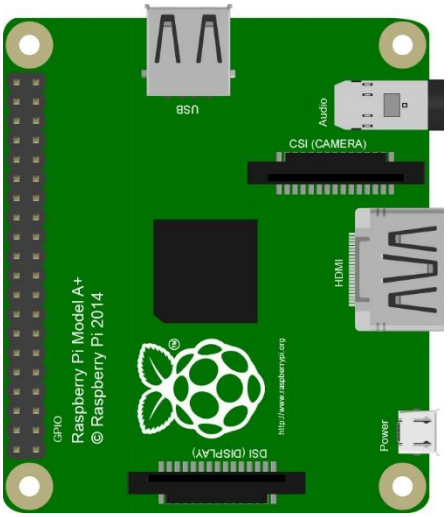
The A type and B type versions of the first generation products have been discontinued due to various reasons. What is most important is that other popular and currently available versions are consistent in the order and number of pins and their assigned designation of function, making compatibility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

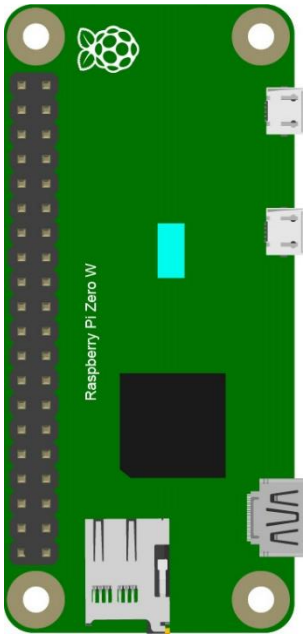
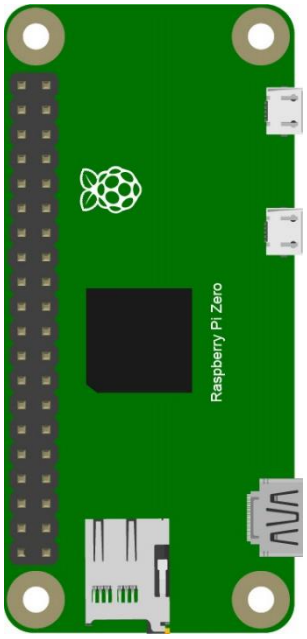
Below are the raspberry pi pictures and model pictures supported by this product. They have 40 pins.

Actual image of Raspberry Pi 4 Model B	CAD image of Raspberry Pi 4 Model B
	

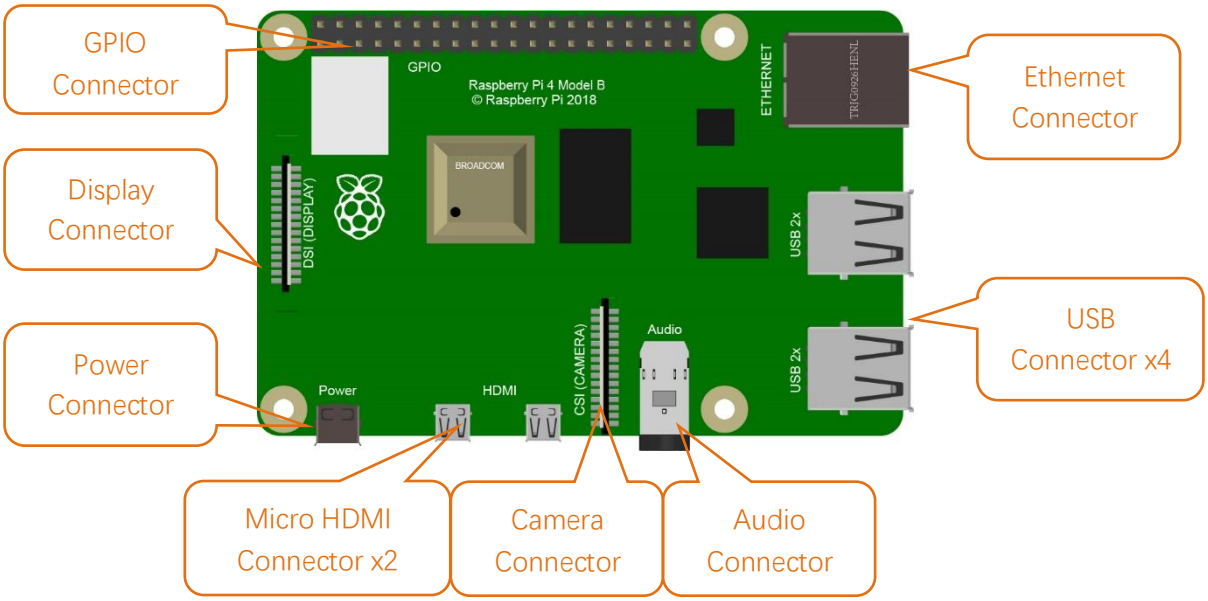
Actual image of Raspberry Pi 3 Model B+	CAD image of Raspberry Pi 3 Model B+
	
Actual image of Raspberry Pi 3 Model B:	CAD image of Raspberry Pi 3 Model B:
	

Actual image of Raspberry Pi 2 Model B: 	CAD image of Raspberry Pi 2 Model B: 
Actual image of Raspberry Pi 1 Model B+: 	CAD image of Raspberry Pi 1 Model B+: 

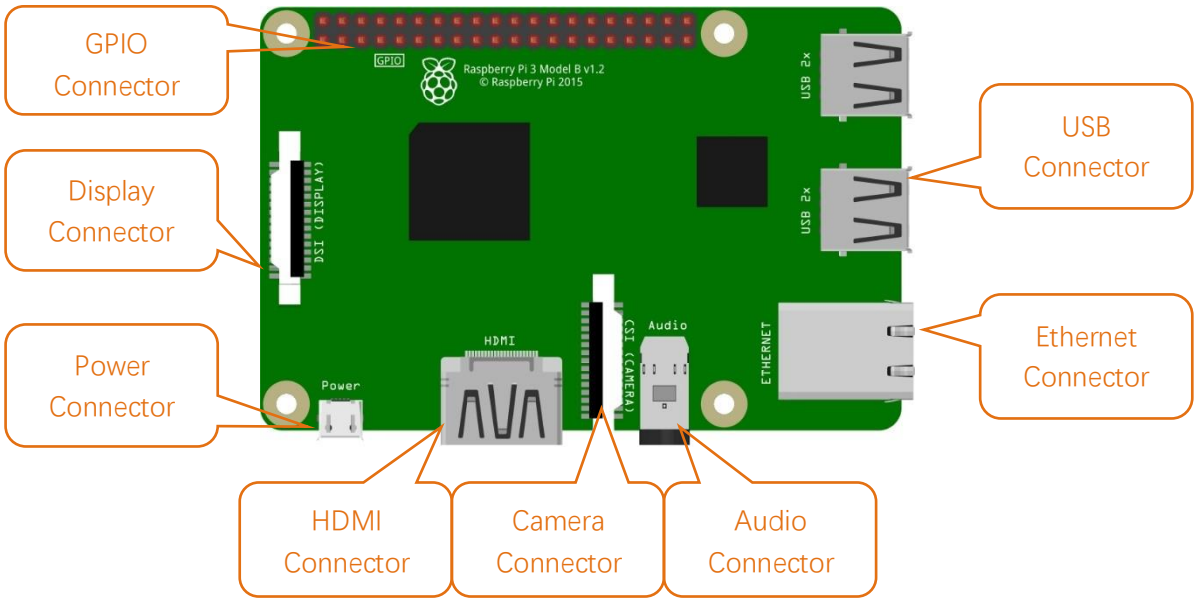
Actual image of Raspberry Pi 3 Model A+: 	CAD image of Raspberry Pi 3 Model A+: 
Actual image of Raspberry Pi 1 Model A+: 	CAD image of Raspberry Pi 1 Model A+: 

Actual image of Raspberry Pi Zero W:	CAD image of Raspberry Pi Zero W:
	
Actual image of Raspberry Pi Zero：	CAD image of Raspberry Pi Zero：
	

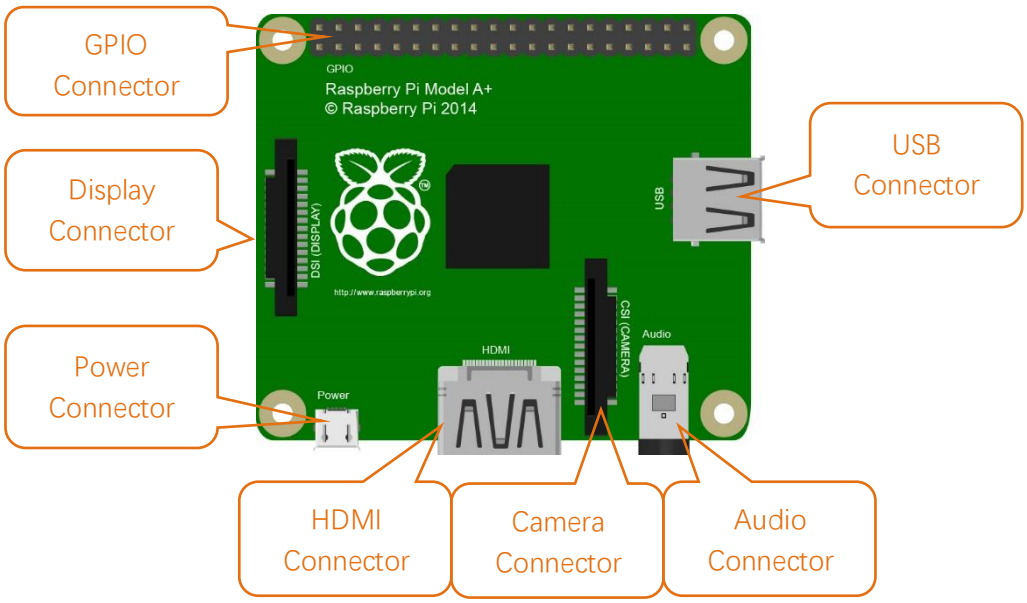
Hardware interface diagram of RPi 4B:



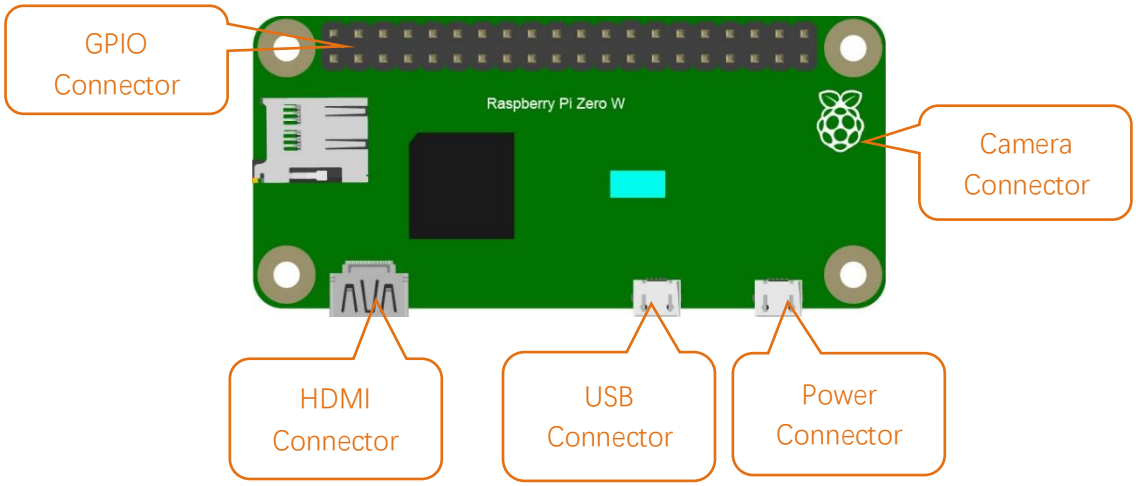
Hardware interface diagram of RPi 3B+/3B/2B/1B+:



Hardware interface diagram of RPi 3A+/A+:



Hardware interface diagram of RPi Zero/Zero W:

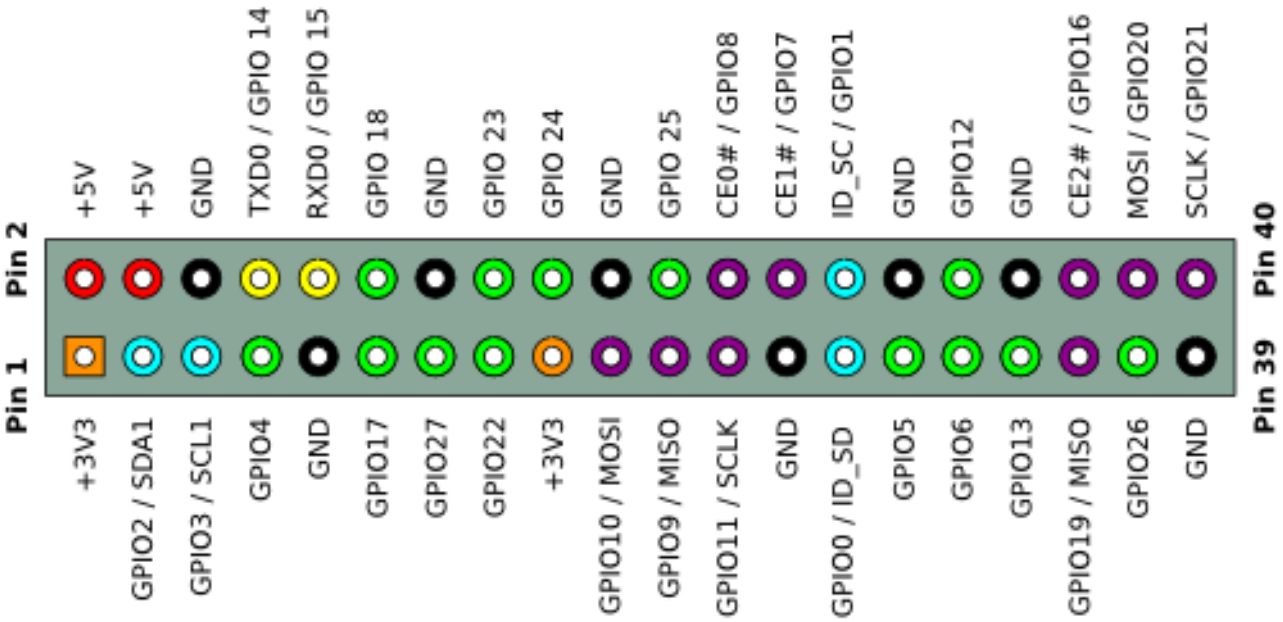


GPIO

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

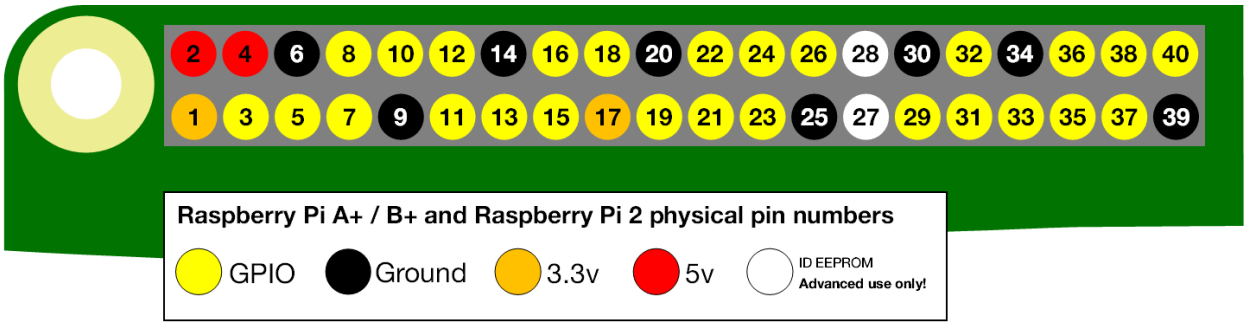
Raspberry Pi CPU use BCM2835/BCM2836/BCM2837of Broadcom. GPIO pin number is set by chip manufacturer. These are the GPIO pins as that computer recognizes. The numbers are unordered and don't make any sense to humans. You will need a printed reference or a reference board that fits over the pins.
Each 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	—	—	For Pi B
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 A+, B+, 2B, 3B, 3B+, 4B, Zero

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

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

```
gpio readall
```

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

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

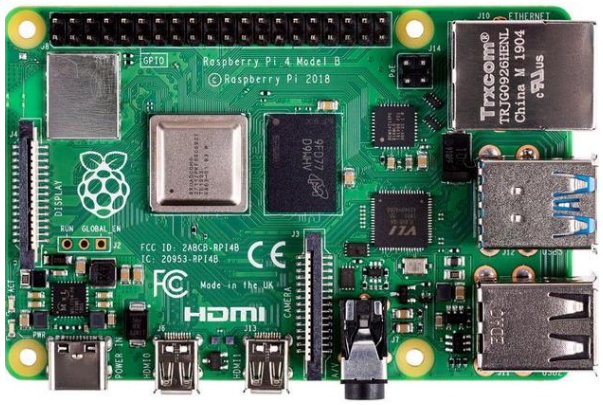
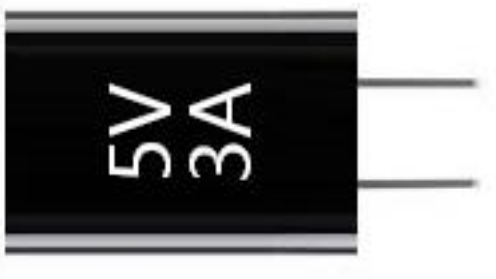
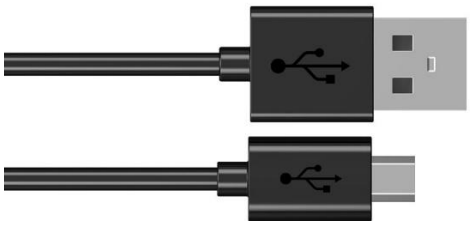
Chapter 0 Raspberry Pi Preparation

Install a System

Firstly, install a system for your RPi.

Component List

Required Components

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

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

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

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

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

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

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

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

Optional Components

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

Required Accessories for Monitor

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

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

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

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

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

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

Required Accessories for Remote Desktop

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

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

Raspberry Pi OS

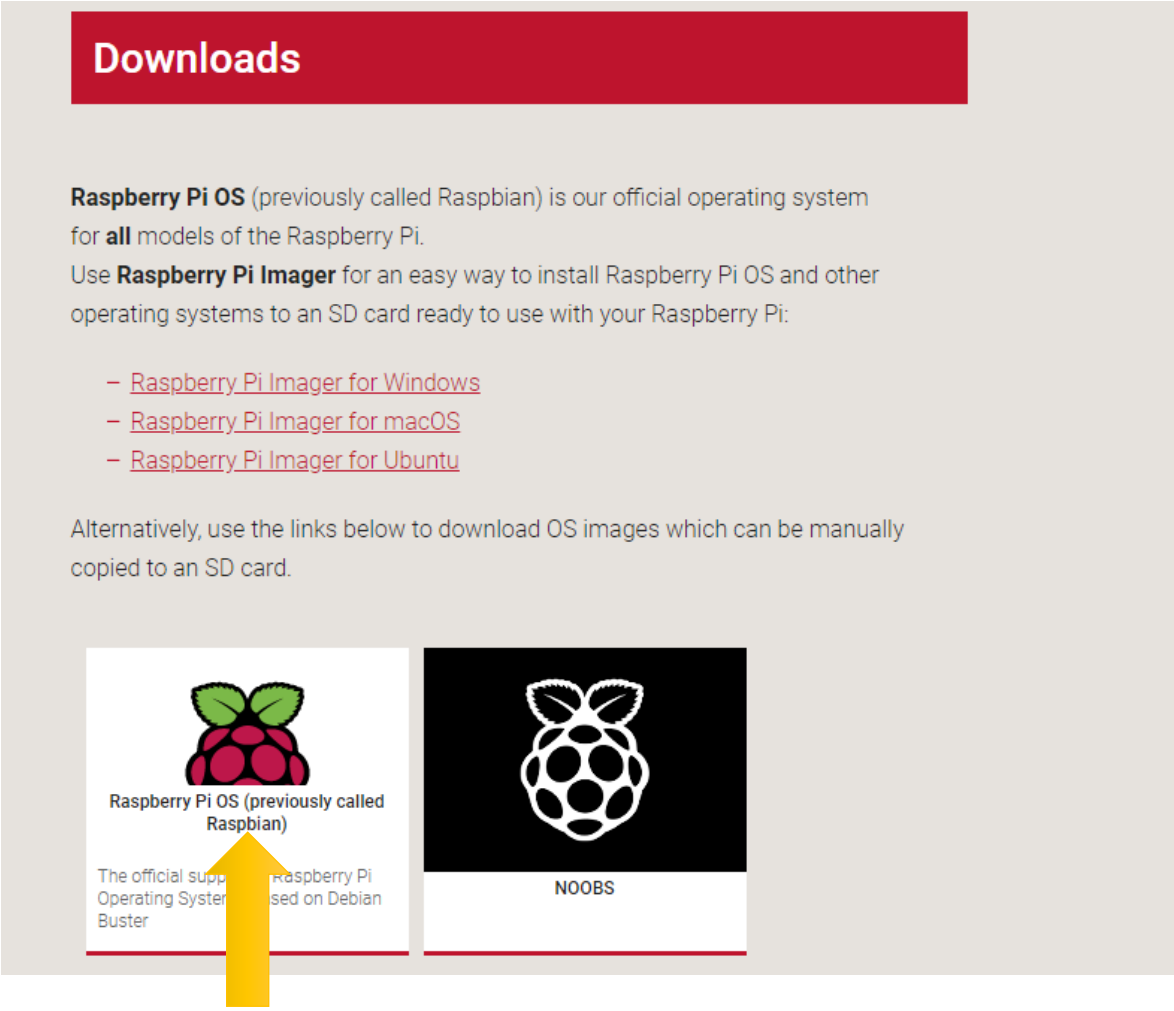
Official Method (Recommended)

You can follow the official method to install the system for raspberry pi via visiting link below:
https://projects.raspberrypi.org/en/projects/raspberry-pi-setting-up/2
In this way, the system will be downloaded **automatically** via the application.

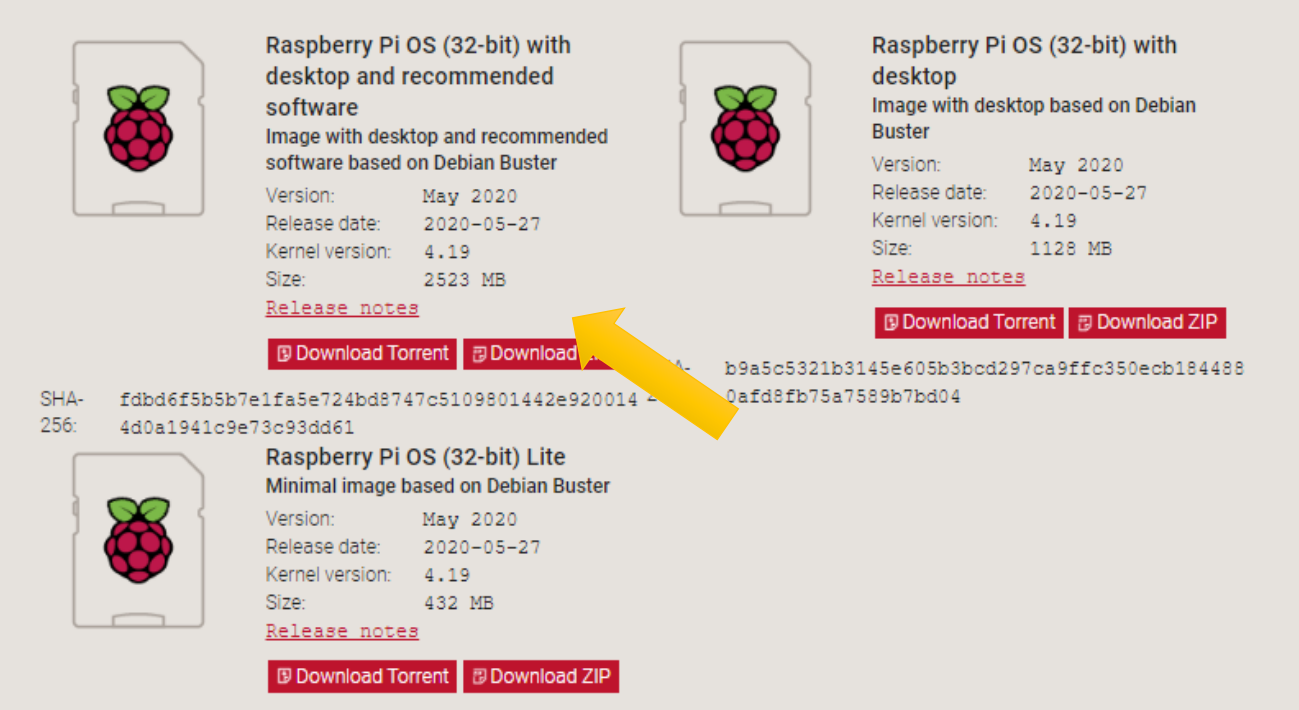
Download system manually (Optional)

After installing the Imager Tool in the **link above**. You can **also** download the system **manually** first.
Visit https://www.raspberrypi.org/downloads/

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



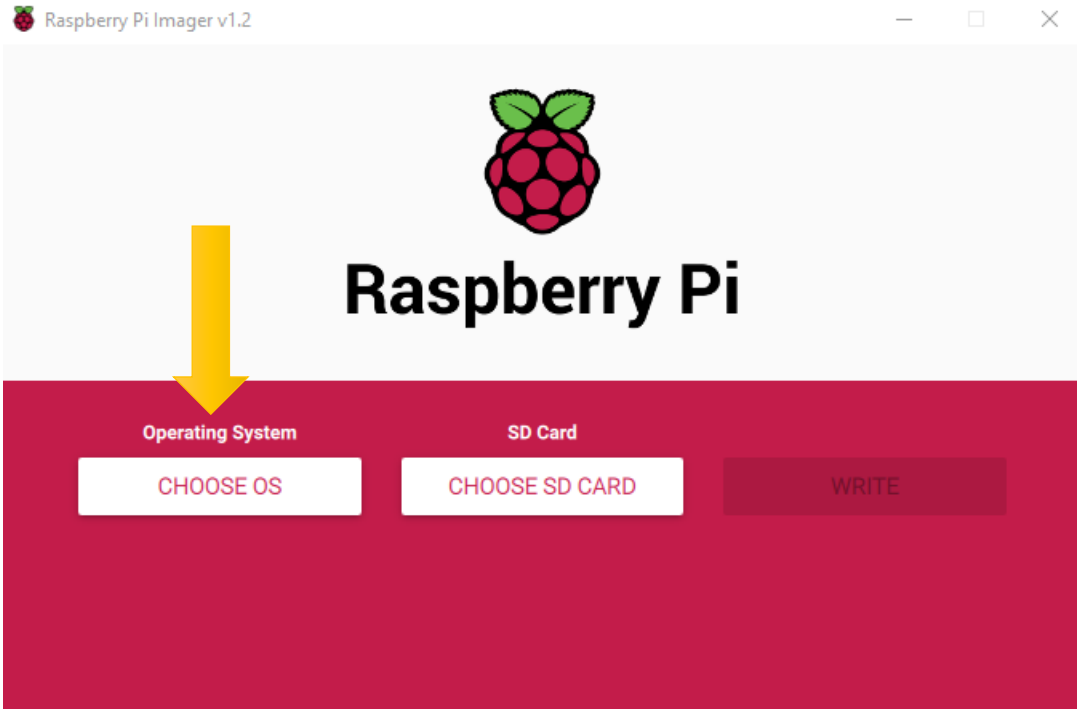
https://www.raspberrypi.org/downloads

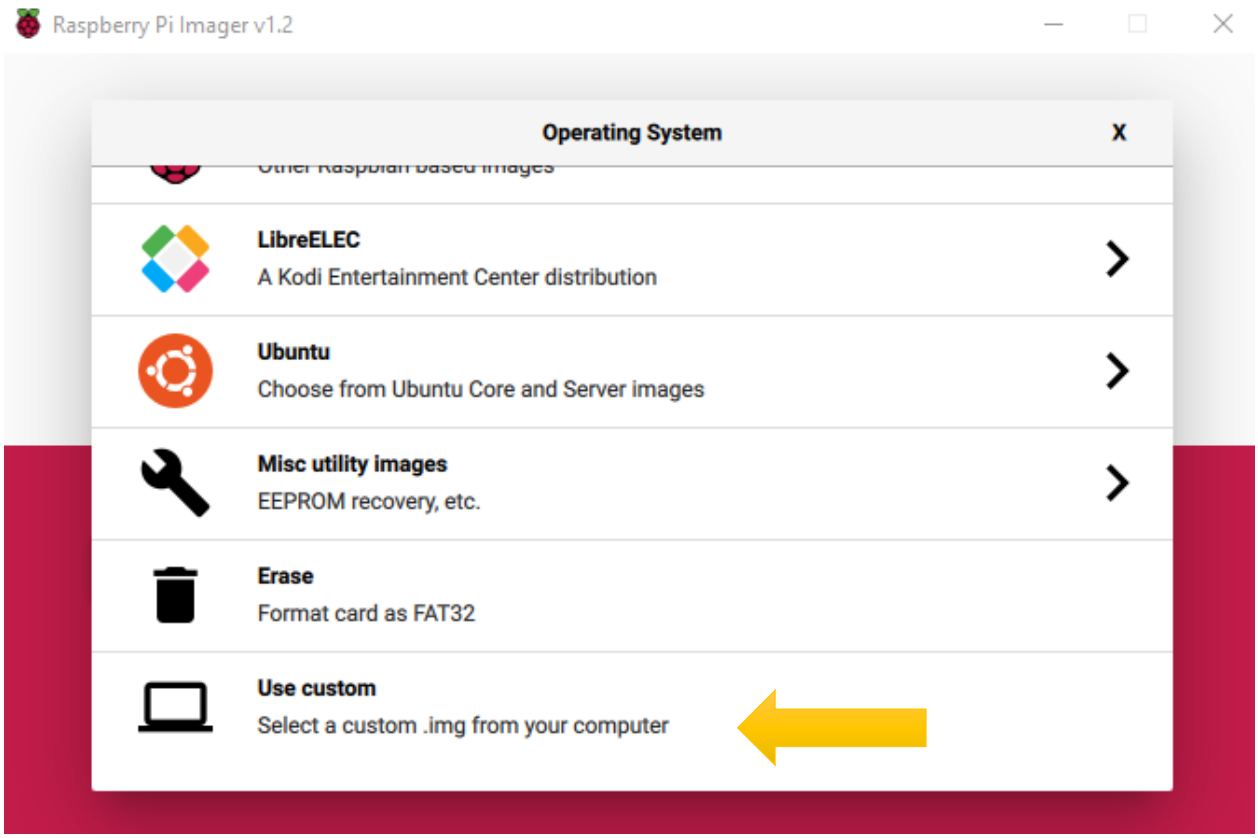


And then the zip file is download.

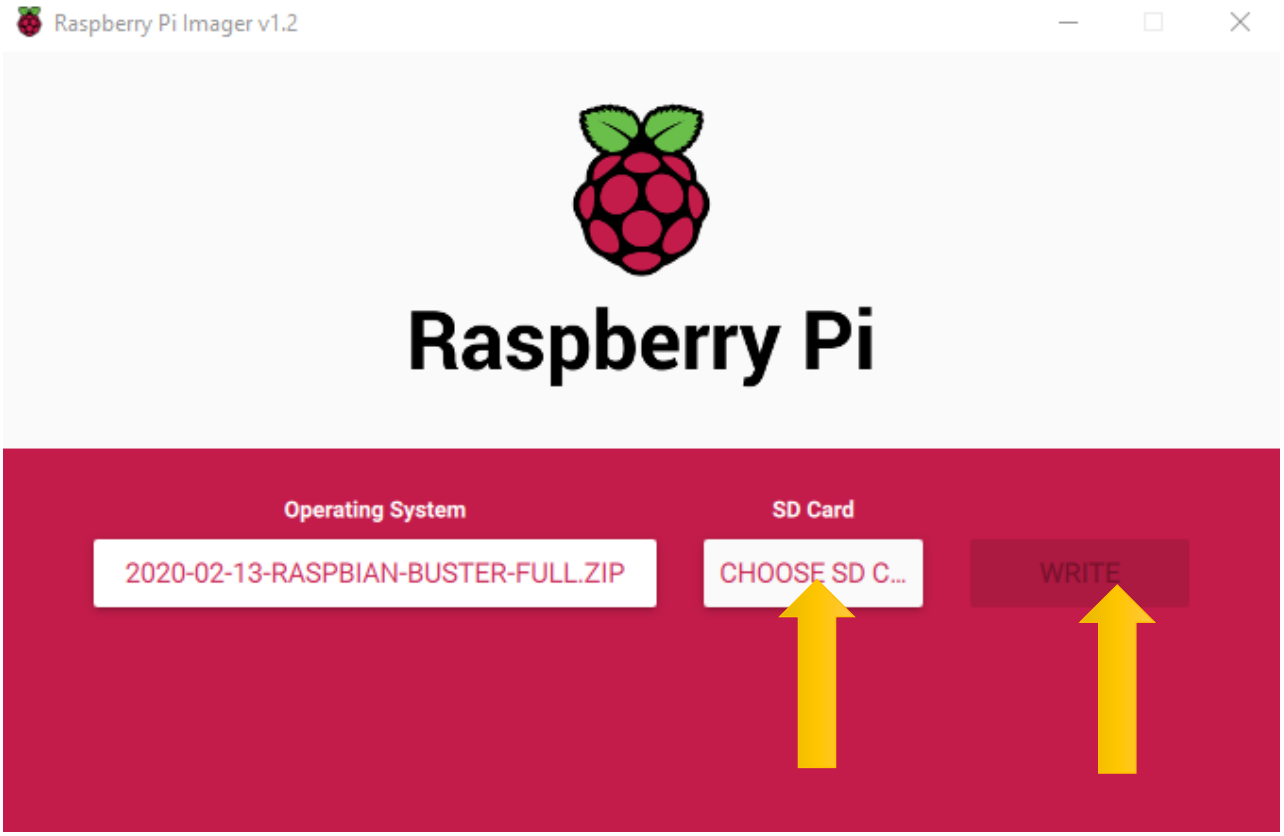
Write System to Micro SD Card

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 skip to next section.

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

After the system is written successfully, take out Micro SD Card and put it into the SD card slot of RPi. Then connect your RPi to send video through the HDMI port, attach your mouse and keyboard through the USB ports, attach a network cable to the network card port and finally, connect your power supply (making sure that it meets the specifications required by your RPi Module Version. Your RPi should start (power up). Later, after setup, you will need to enter your user name and password to login. The default user name: pi; password: raspberry. After login, you should see the following screen.

Congratulations! You have successfully installed the RASPBERRY PI OS operating system on your RPi. Then you can connect WiFi on the right corner.

Now you can skip to VNC Viewer.

Remote desktop & VNC

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

If you don't have a spare 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 and mac OS.

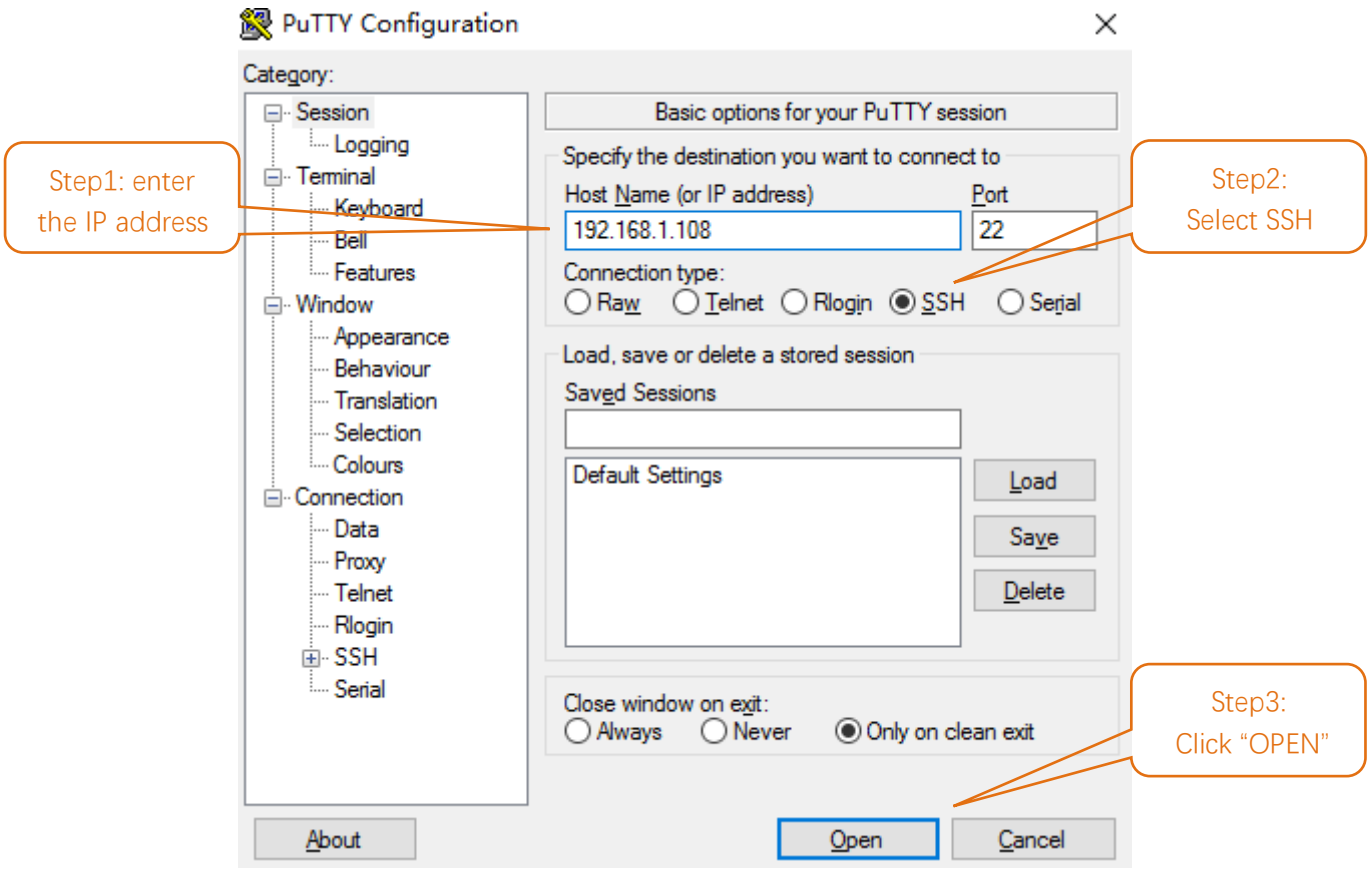
Enable SSH

Under the latest version of Raspberry Pi OS, SSH is closed by default. So you need to open it first.
After write system**, create a folder named "ssh" under generated boot disk of Micro SD Card,** then the SSH connection will be opened.

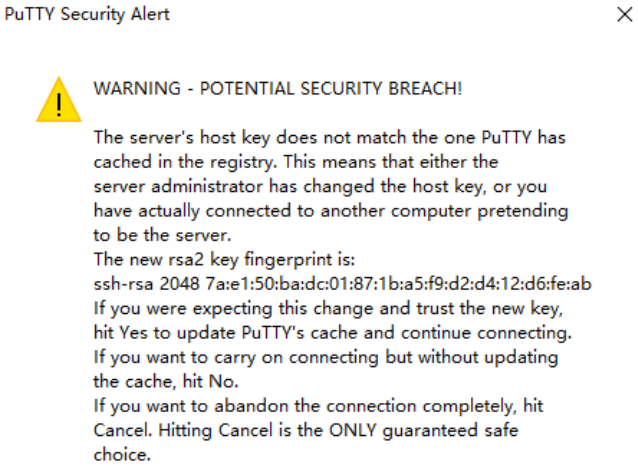
Windows OS Remote Desktop

The windows built-in application remote desktop corresponds to the Raspberry Pi xrdp service.
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

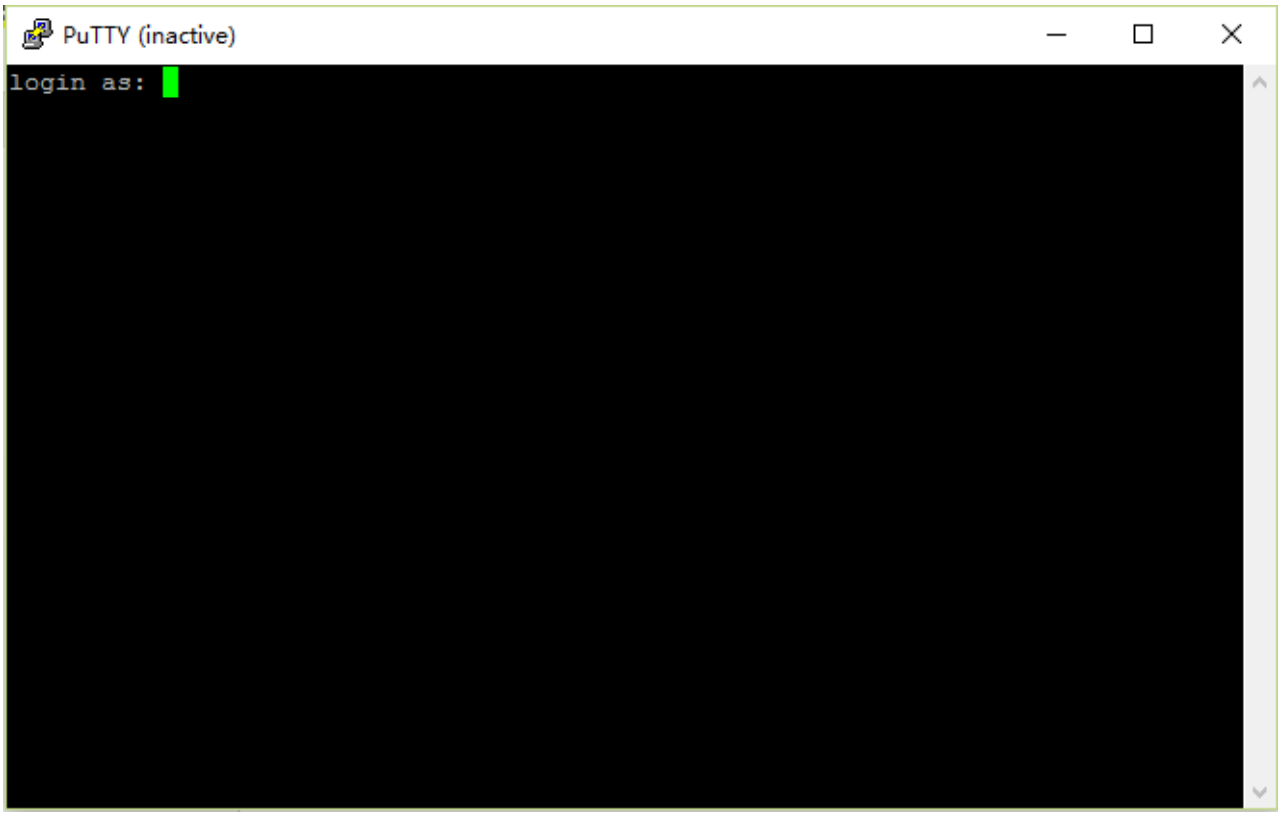
Then use net cable to connect your RPi to the same router with your PC. Then put the system Micro SD Card prepared before into the slot of the RPi and turn on the power supply. Enter router client to inquiry IP address named "raspberry pi". For example, my RPi IP address is "192.168.1.108".
Then open Putty, enter the address, select SSH, and then click "OPEN", as shown below:



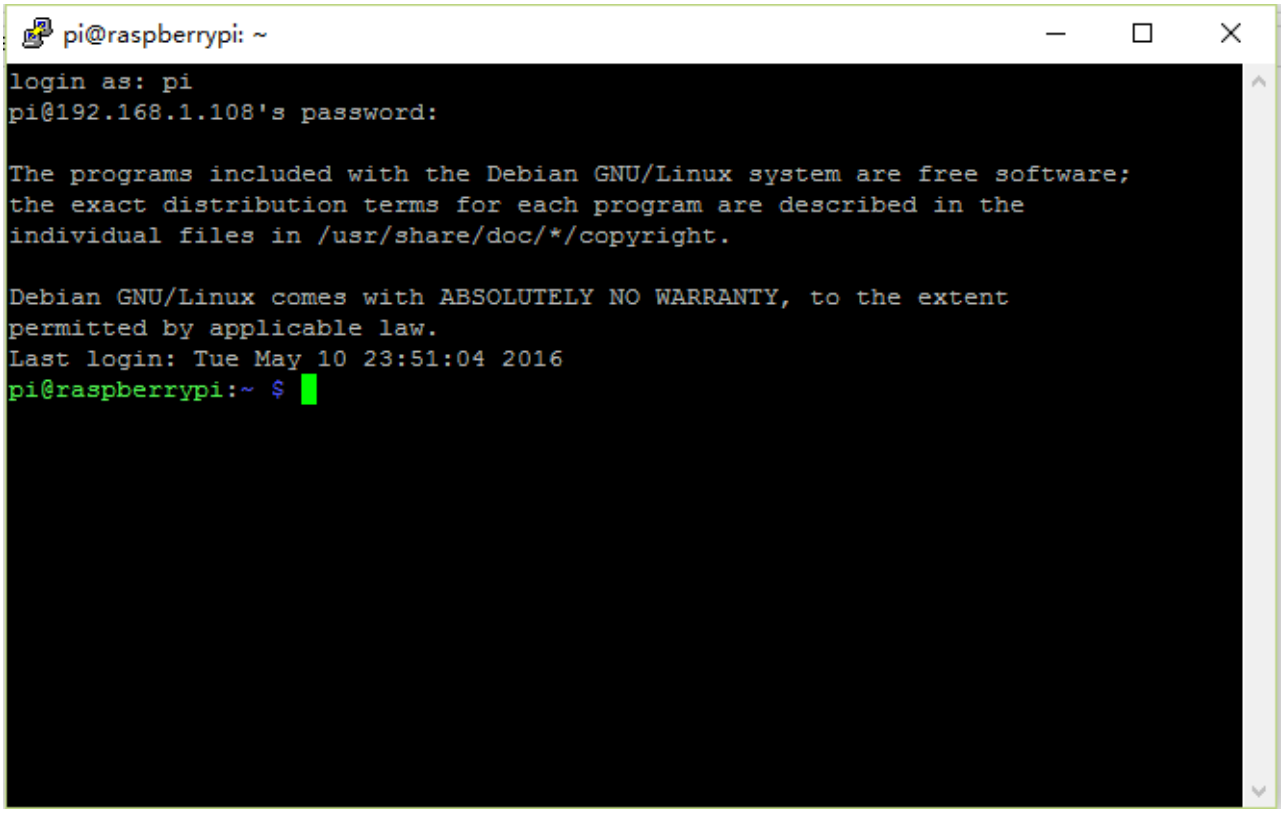
There will appear a security warning at first login. Just click "YES".



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.

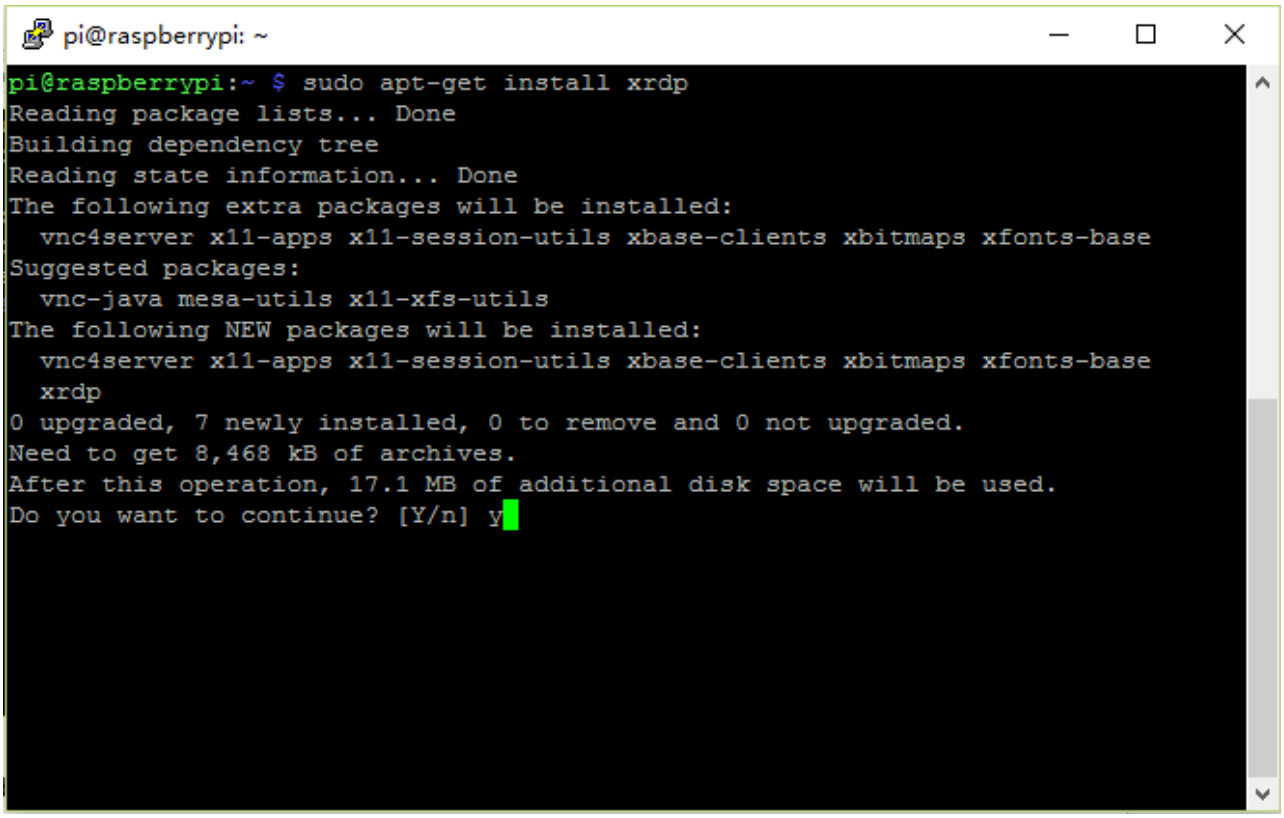


If you want to use built-in Remote Desktop Connection under Windows, you need install xrdp service on Raspberry Pi.

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.

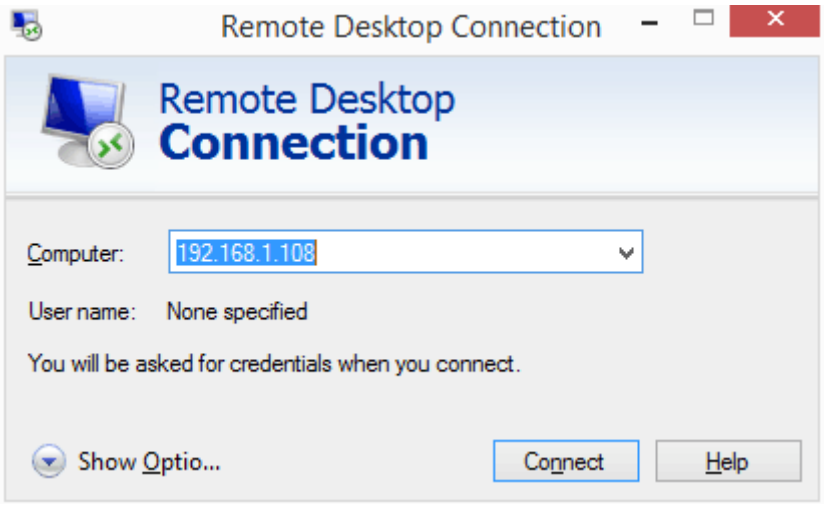


Enter "Y", press key “Enter” to confirm.

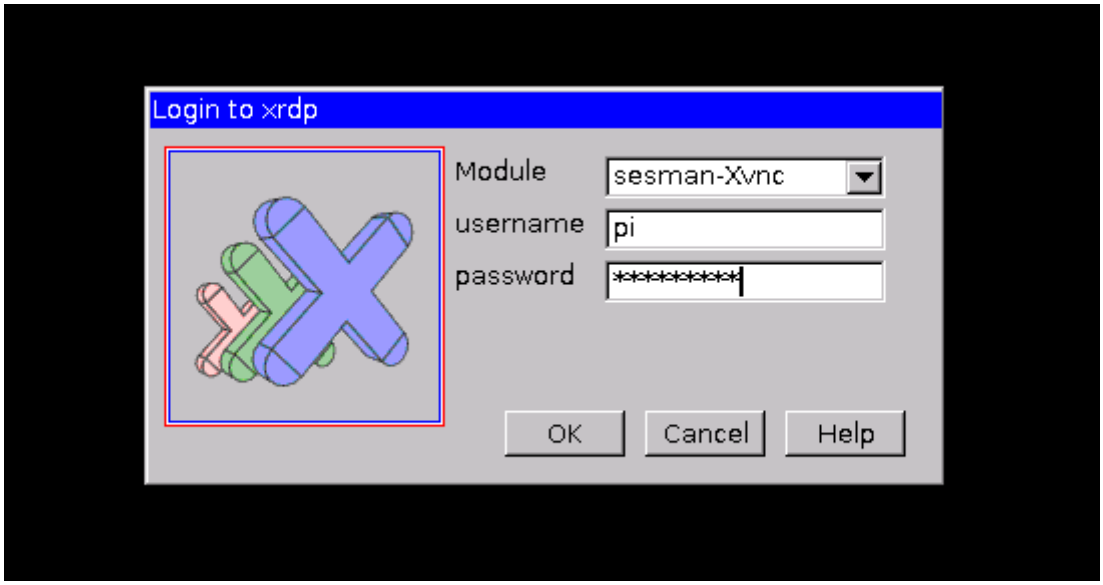
After the installation is completed, you can use Windows remote desktop applications to login to your RPi.

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.

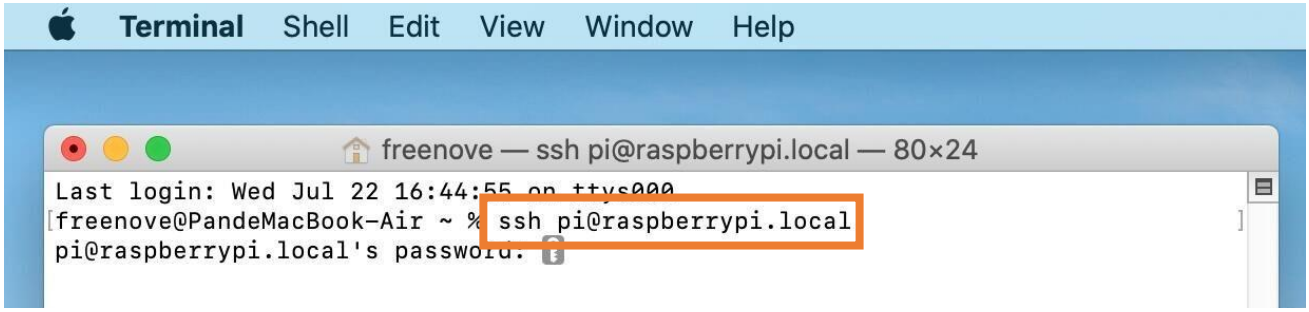
Raspberry Pi 4B/3B+/3B integrates a Wi-Fi adaptor. You can use it to connect to your Wi-Fi. Then you can use the wireless remote desktop to control your RPi. This will be helpful for the following operation. Raspberry Pi of other models can use wireless remote desktop through accessing an external USB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

MAC OS Remote Desktop
Connect your pi to the router via a network cable.

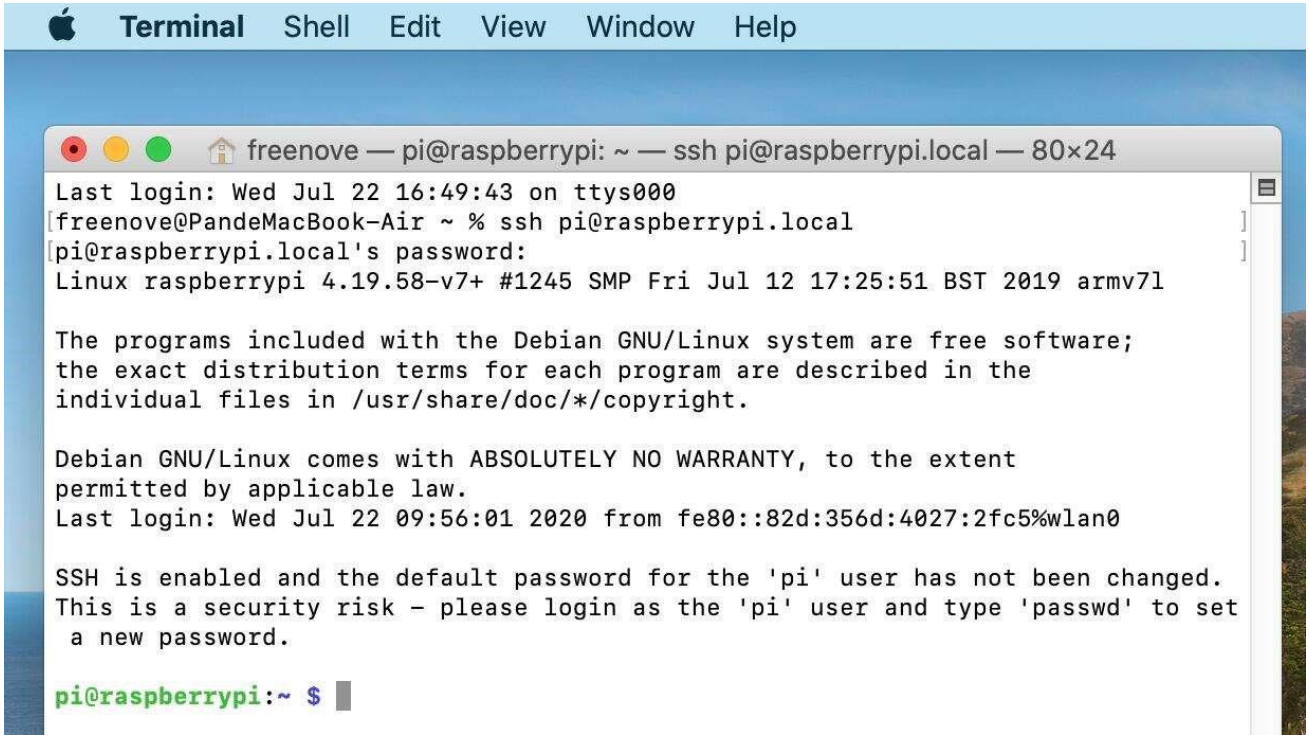
Open the terminal and type following command.

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

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



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

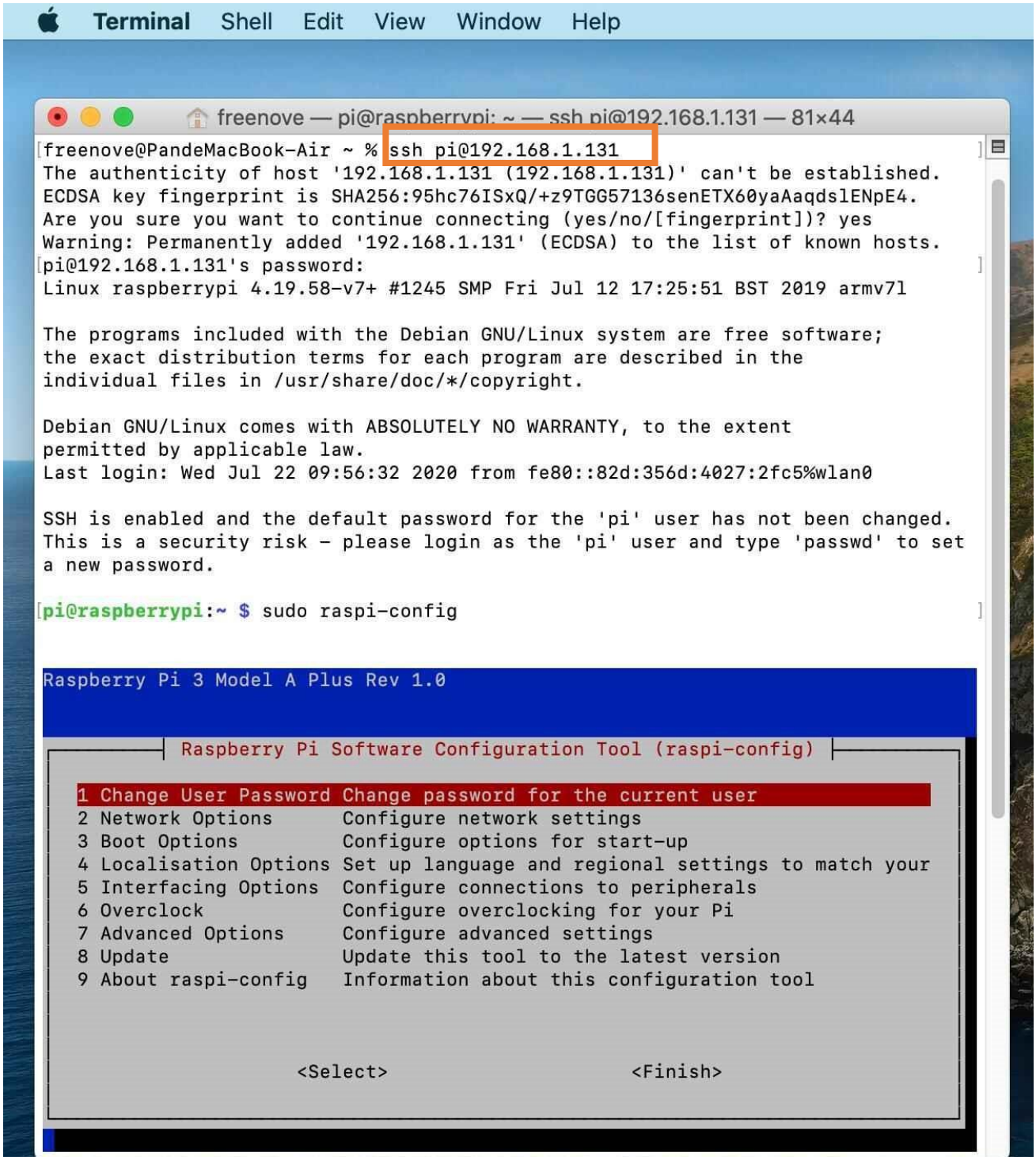


When you see pi@raspberrypi:~ $, you have logged in Pi successfully. Then you can skip to next section.

You can also use the IP address to log in Pi.
Enter router client to inquiry IP address named “raspberry pi”. For example, I have inquired to my RPi IP address, and it is “192.168.1.131".

Open the terminal and type following command.

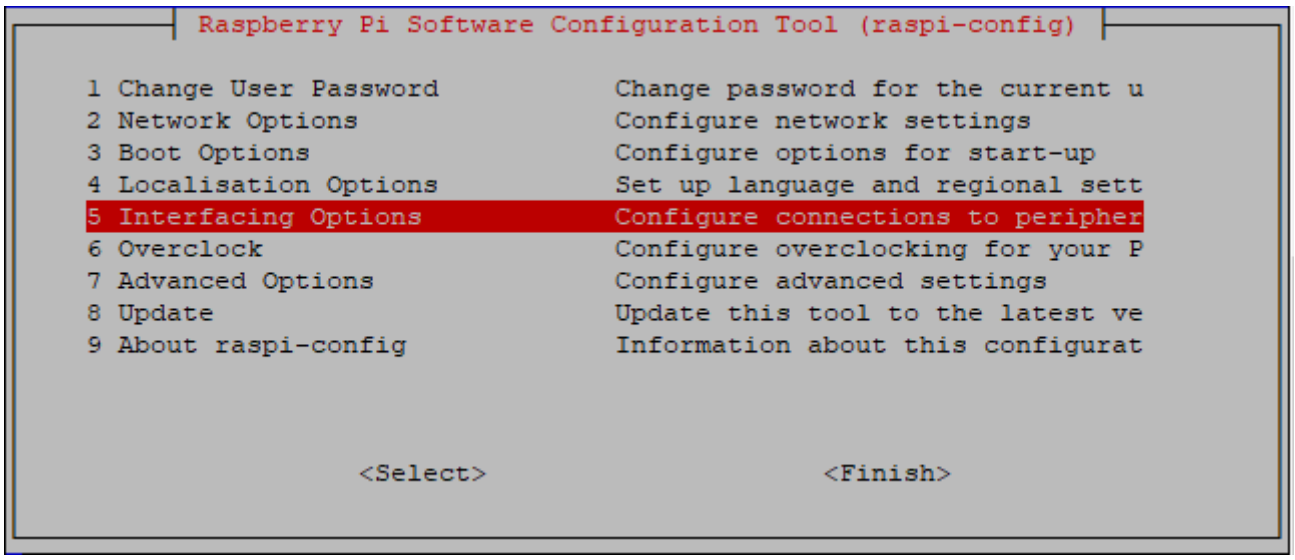
```
ssh pi@192.168.1.131
```

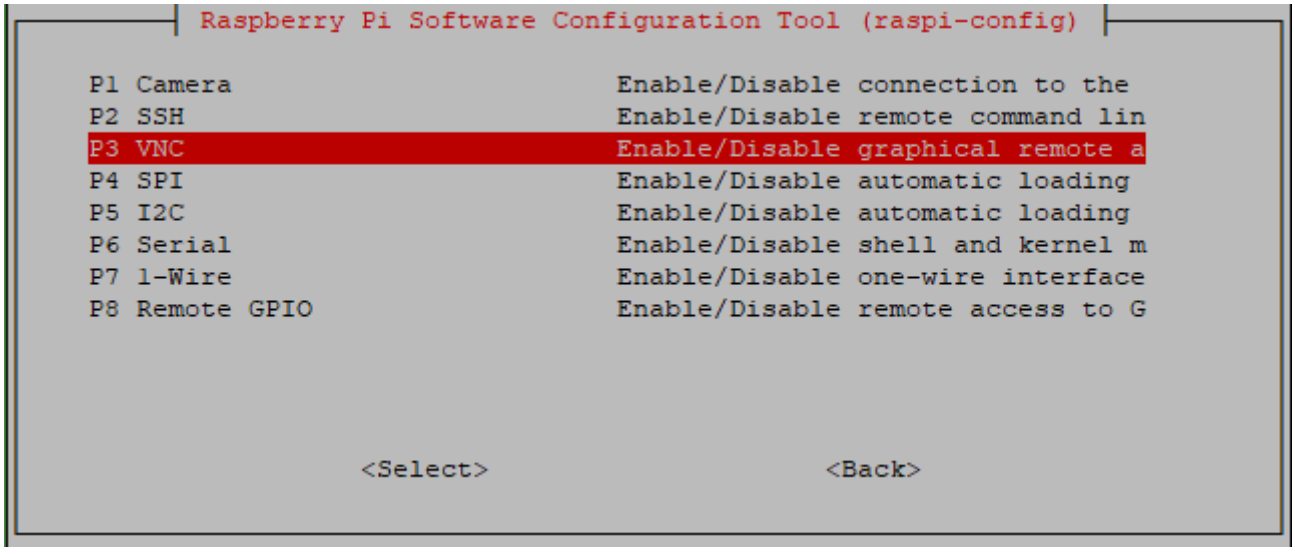

Terminal Shell Edit View Window Help
freenove — pi@raspberrypi: ~ — ssh pi@192.168.1.131 — 81×44
freenove@PandeMacBook-Air ~ % ssh pi@192.168.1.131
The authenticity of host '192.168.1.131 (192.168.1.131)' can't be established.
ECDSA key fingerprint is SHA256:95hc76ISxQ/+z9TGG57136senETX60yaAaqdslENpE4.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '192.168.1.131' (ECDSA) to the list of known hosts.
pi@192.168.1.131's password:
Linux raspberrypi 4.19.58-v7+ #1245 SMP Fri Jul 12 17:25:51 BST 2019 armv7l
The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.
Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Wed Jul 22 09:56:32 2020 from fe80::82d:356d:4027:2fc5%wlan0
SSH is enabled and the default password for the 'pi' user has not been changed.
This is a security risk - please login as the 'pi' user and type 'passwd' to set
a new password.
pi@raspberrypi:~ $ sudo raspi-config
Raspberry Pi 3 Model A Plus Rev 1.0
Raspberry Pi Software Configuration Tool (raspi-config)
1 Change User Password Change password for the current user
2 Network Options Configure network settings
3 Boot Options Configure options for start-up
4 Localisation Options Set up language and regional settings to match your
5 Interfacing Options Configure connections to peripherals
6 Overclock Configure overclocking for your Pi
7 Advanced Options Configure advanced settings
8 Update Update this tool to the latest version
9 About raspi-config Information about this configuration tool
<Select>
<Finish>

VNC Viewer & VNC

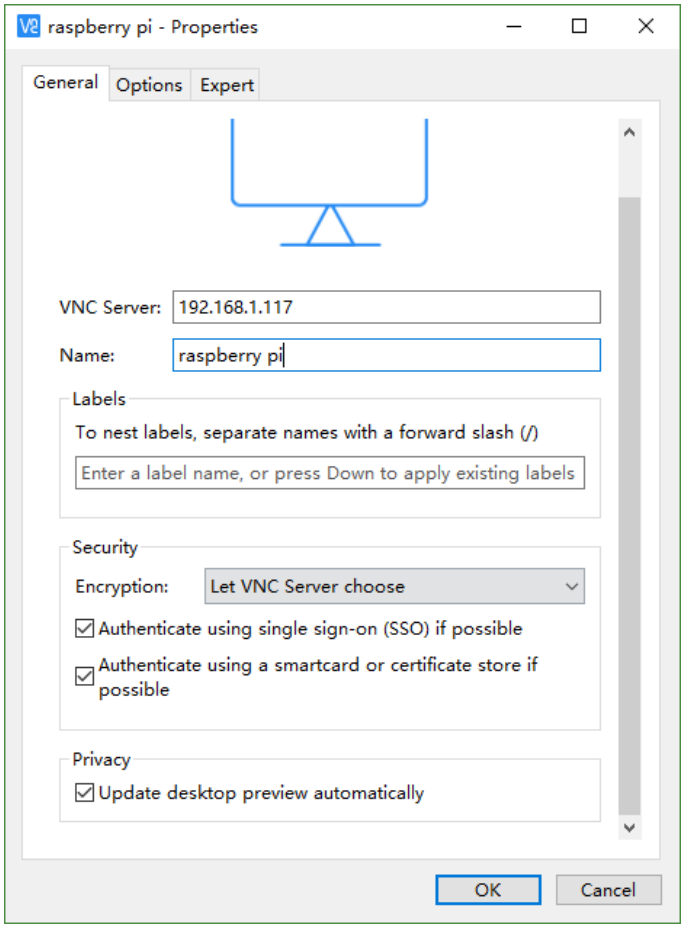
Type the following command. And select 5 Interfacing Options→P3 VNC →Yes→OK→Finish. Here Raspberry Pi may need to be restarted, and choose ok. Then open VNC interface.

```
sudo raspi-config
```

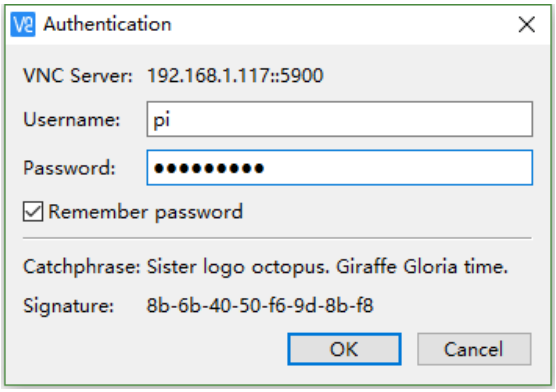




Then download and install VNC Viewer according to your computer system by clicking following link:
https://www.realvnc.com/en/connect/download/viewer/
After installation is completed, open VNC Viewer. And click File → New Connection. Then the interface is shown below.



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



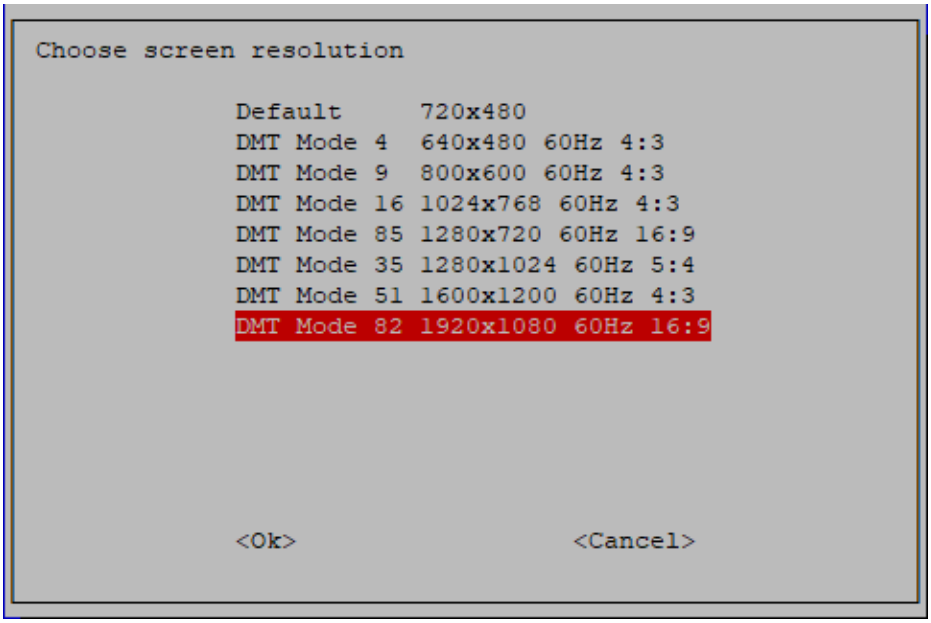
Enter username: **pi** and Password: **raspberry**. And click OK.

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

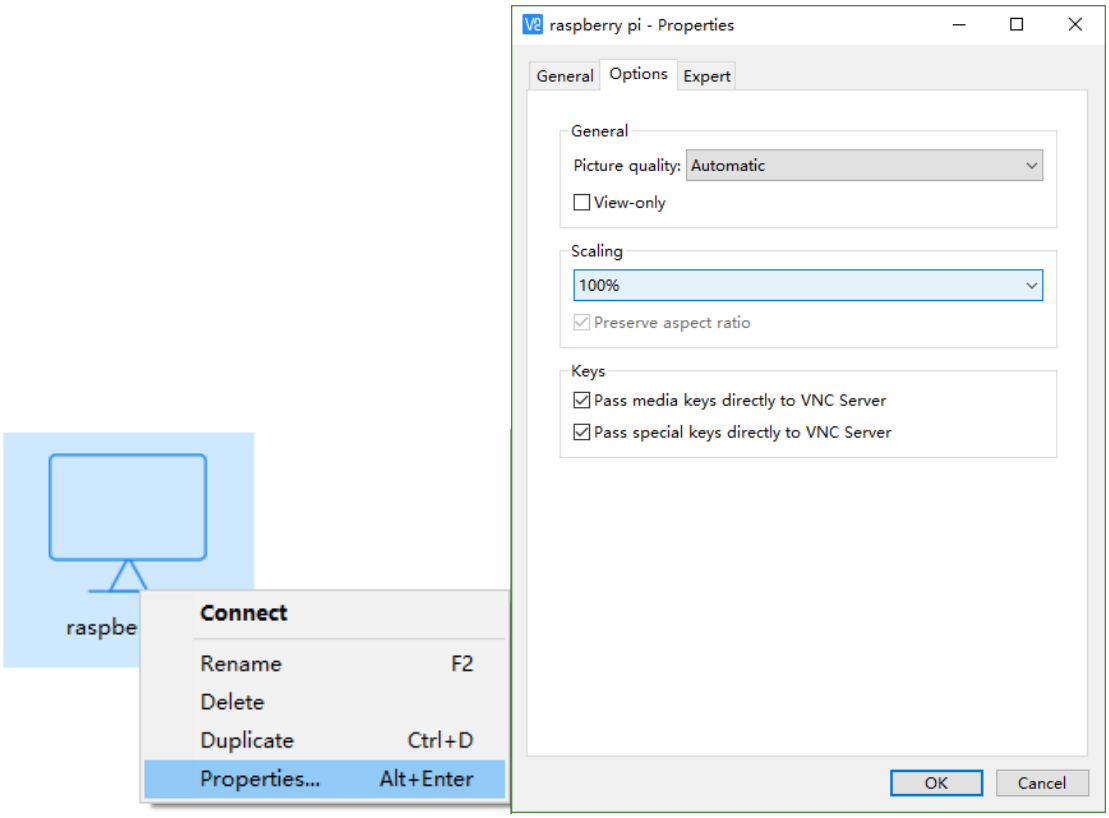
If the resolution ratio is not great or there is just a **little window**, you can set a proper resolution ratio via steps below.

```
sudo raspi-config
```

Select 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.



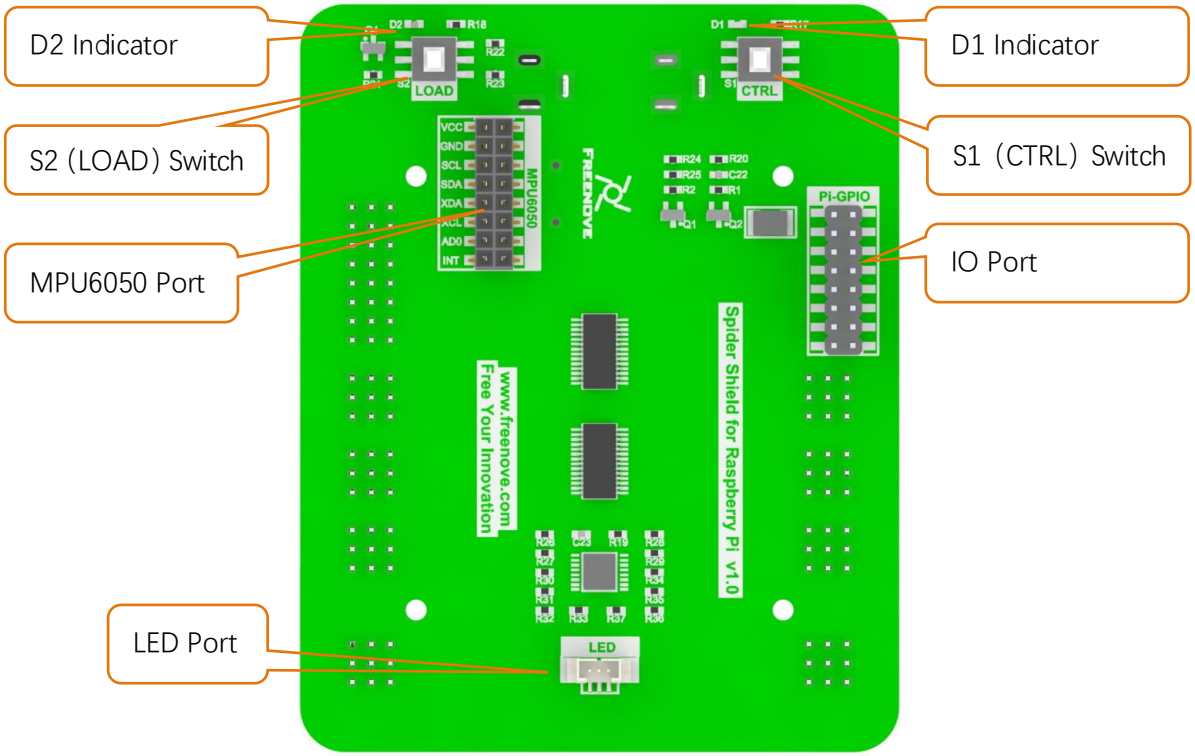
Here, you have logged in to Raspberry Pi successfully by using VNC Viewer and operated proper setting.

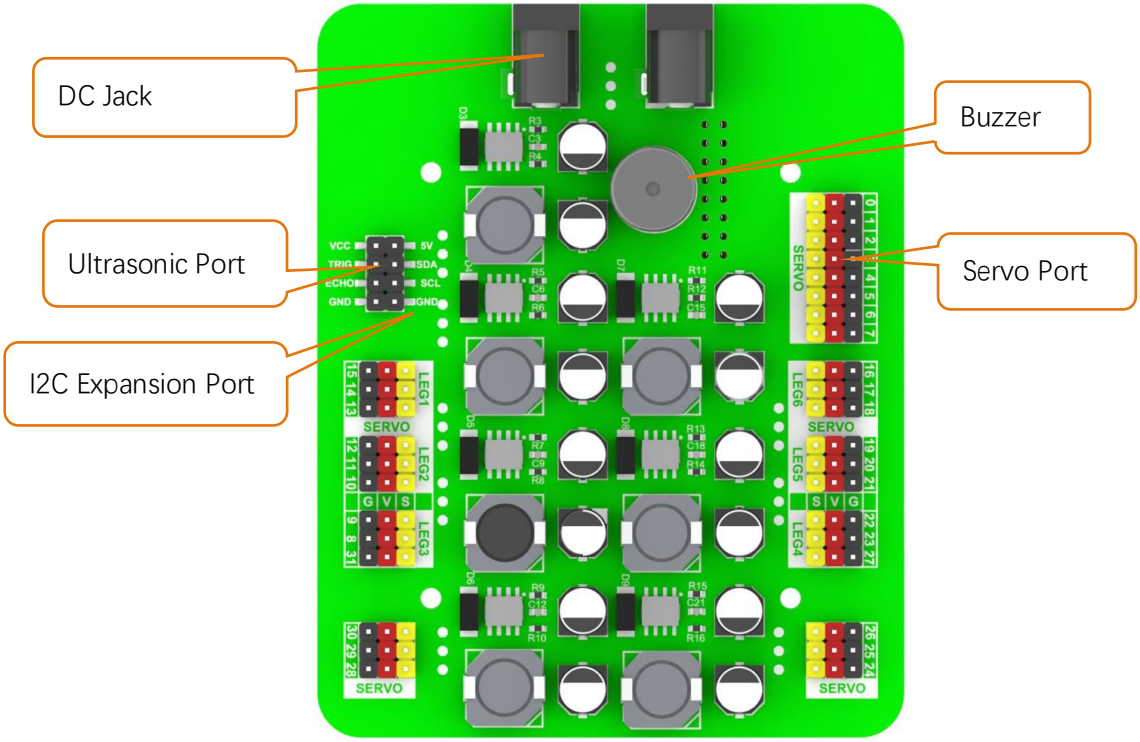
Raspberry Pi 4B/3B+/3B integrates a Wi-Fi adaptor. If you did not connect Pi to WiFi. You can connect it to wirelessly control the robot.

Spider Shield for Raspberry Pi

Description of Robot Control Board

The control board is mainly connected the Raspberry Pi through the "PI-GPIO" port and the positioning holes on the board are suitable for the Raspberry Pi. The features and functions of the control panel are as follows.





- **Spider Shield board**: powered by two sets of batteries without protection plates, with each set containing two 18650 batteries and voltage ranging from 7V-8.4V.
- **DC socket**: Two DC sockets available, each connecting to one set of batteries, each containing two 18650 batteries.
- **S1(CTRL) Switch:** It mainly controls chips like PCA9685 and ADS7830 and the power supply of the Raspberry Pi. When it is pressed, D1 indicator will turn ON.
- **S2(LOAD) Switch:** It mainly controls the power supply of servos. When it is pressed, D2 indicator will turn ON.
- **LED port**: Used to connect the LED Module, controlled directly by RPi.
- **MPU6050 Module port**: Used to connect MPU6050, controlled directly by RPi.
- **PI-GPIO port**: Used to connect control board with RPi, facilitating RPi to control the robot
- **I2CExtension Board**: Used for the extension of other I2C devices.
- **Servo port**: 32 servo ports available.
- **Sonic module port**: The interface of HC-SR04 Ultrasonic Module is provided.
- **Buzzer**: Turning ON when it is at high level and OFF at low level, directly controlled by RPi

Chapter 1 Installation of Python Libraries(Important)

In this chapter, we will do some necessary foundational preparation work: Start your Raspberry Pi and install some necessary libraries. And we will assemble the robot in the next chapter. Assembling and using the robot following the tutorial allows you to have a good experience of the fun brought by the robot.

Caution: Do NOT install it without referring to the tutorial. Otherwise, it may cause installation errors and damage the components.

Note:

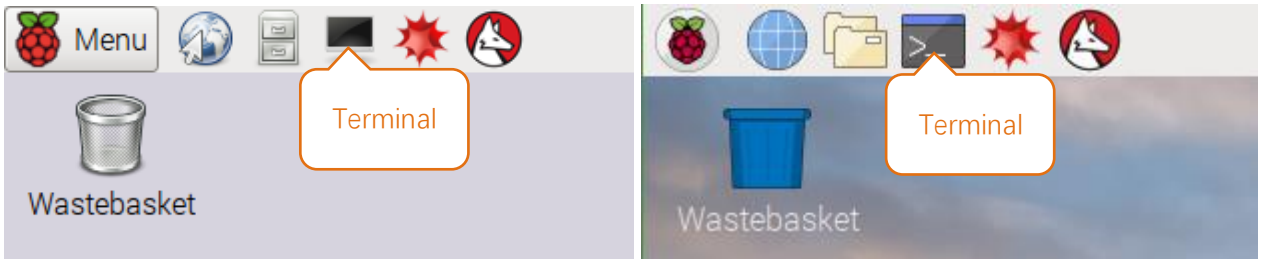
1, **Please use Raspberry Pi OS with Desktop**

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

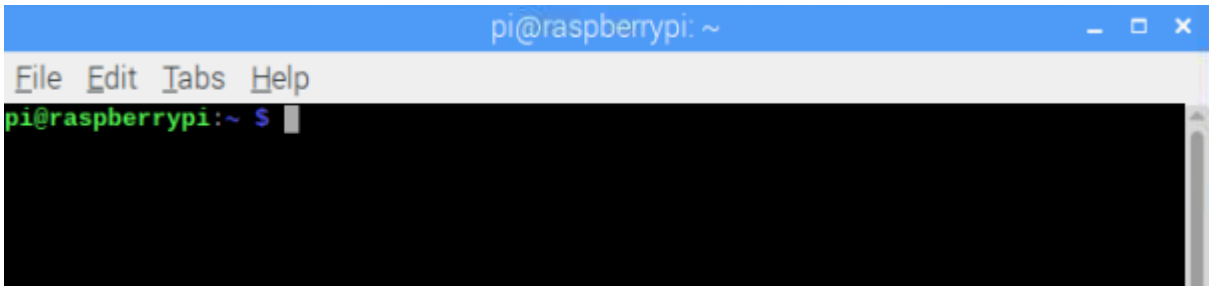
3, If you are using **remote desktop** to login Raspberry Pi, you need to use VNC viewer.

Step 1 Obtain the Code

Start the Raspberry Pi and open the terminal. You can open the terminal as follows or press "CTRL+ALT+T" on the desktop.



The terminal is shown below:



Enter the following commands in the terminal to obtain the robot's code and save it to the user's directory" Pi". (Note that here are two commands, please enter them one by one in order.)

```
cd ~

git clone https://github.com/Freenove/Freenove_Big_Hexapod_Robot_Kit_for_Raspberry_Pi.git
```

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).

Note: all codes of the robot are written with Python3. If run with Python2, it may cause errors.

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.

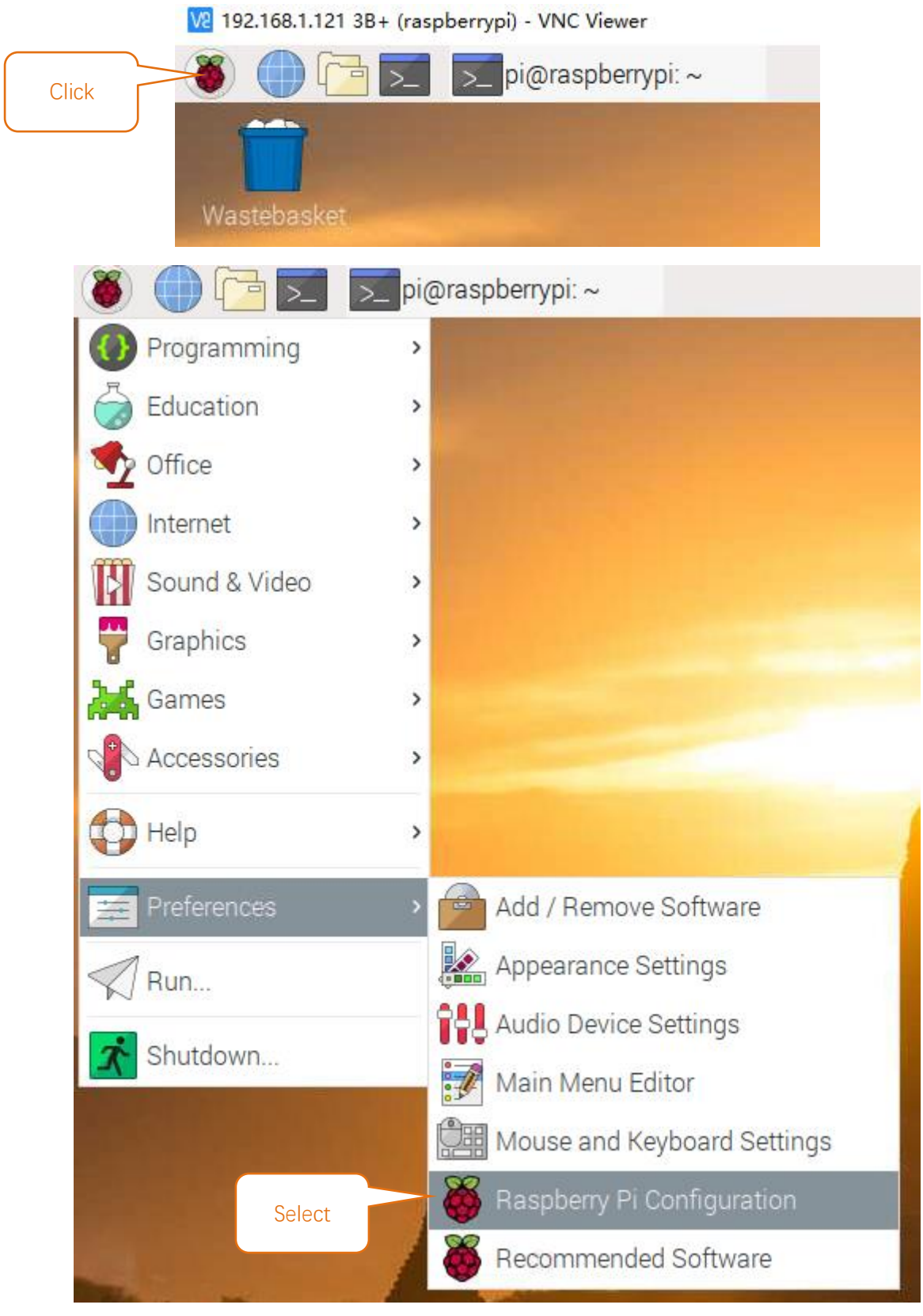
2. **Tab key**. The Tab key can automatically complete the command/path you want to type. When there are multiple commands/paths conforming to the already typed letter, pressing Tab key once won't have any result. And pressing Tab key again will list all the eligible options. This command/path will be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

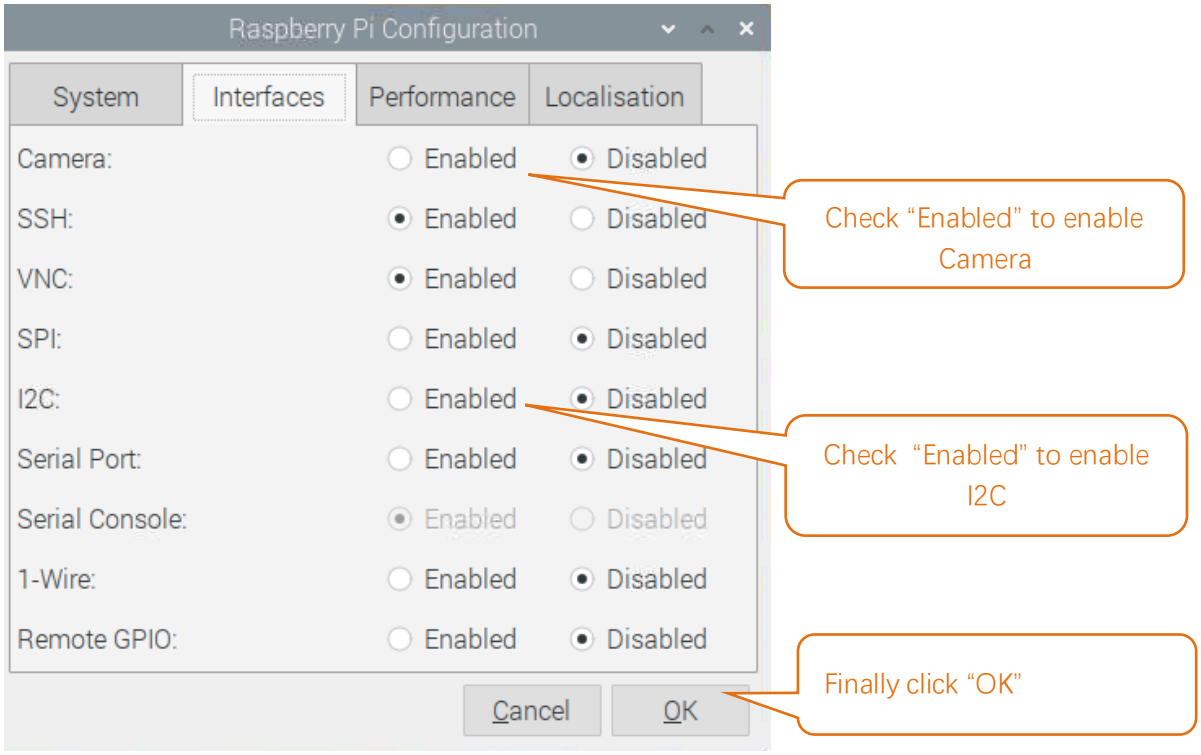
As shown below, under the '~'directory, enter the Documents directory with the "cd" command. After typing "cd D", press Tab key, then there is no response. Press Tab key again, then all the files/folders that begin with "D" is listed. Continue to type the character "oc", then press the Tab key, and then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

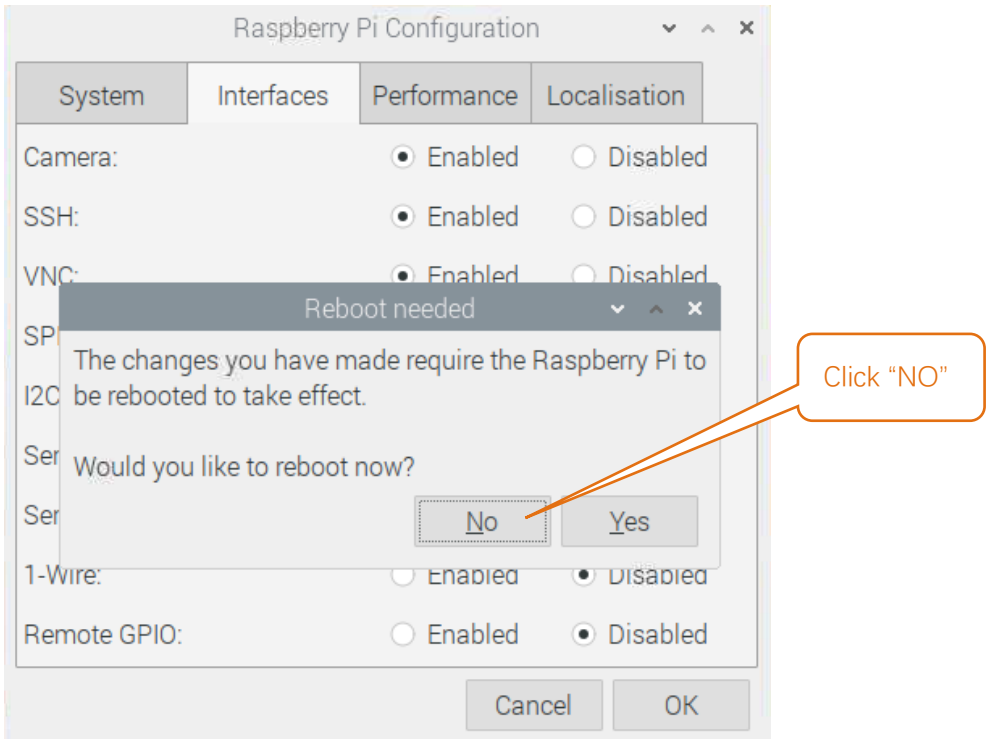
```
pi@raspberrypi:~ $ cd D
Desktop/   Documents/ Downloads/
pi@raspberrypi:~ $ cd Doc
```

```
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
Check “Enabled” to enable Camera
Check “Enabled” to enable 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”

Additional supplement

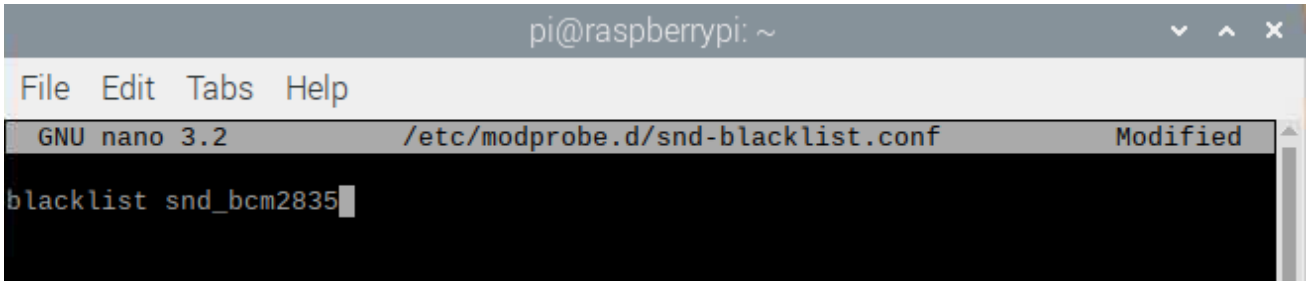
Raspbery Pi, other than 4B and 400, nees to disable the audio module, othewise the LED will not work properly.

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

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

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

```
blacklist snd_bcm2835
```



2. We also need to edit config file.

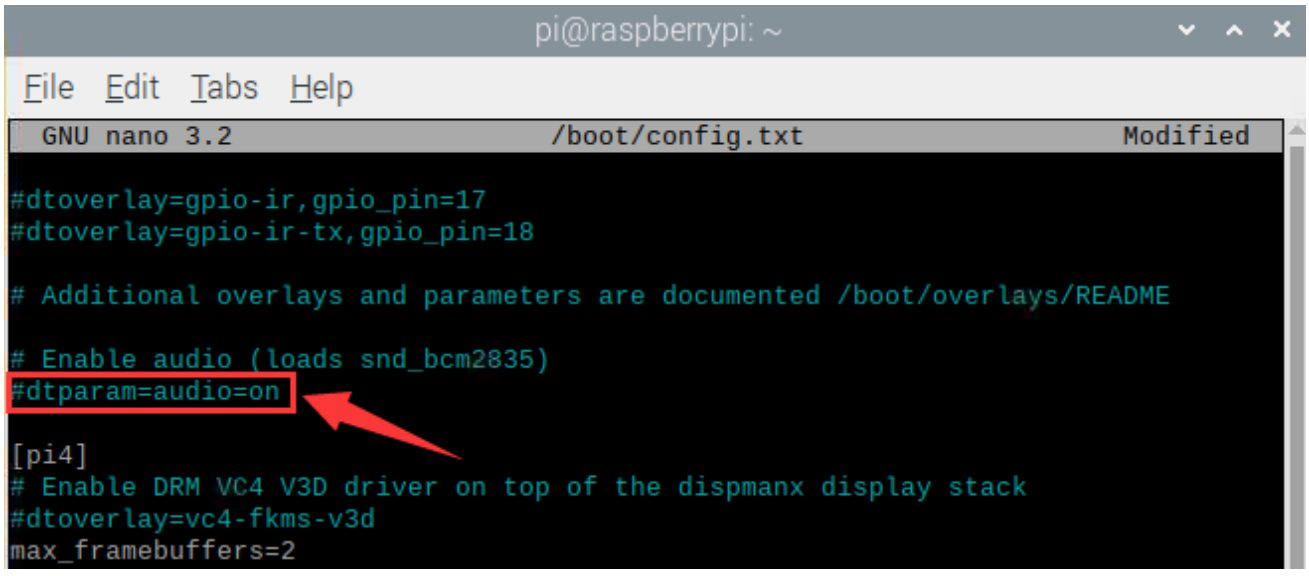
```
sudo nano /boot/config.txt
```

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

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

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

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



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

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

Step 3 Execute the Installed Program

All commands are based On python3. If the default python of your system is Python2, please change it to Python3 referring to Step1.

1. Execute following commands to enter directory of “build.sh”.

```
cd ~/Freenove_Big_Hexapod_Robot_Kit_for_Raspberry_Pi/Code
```

2. Run build.sh

```
sh ./build.sh
```

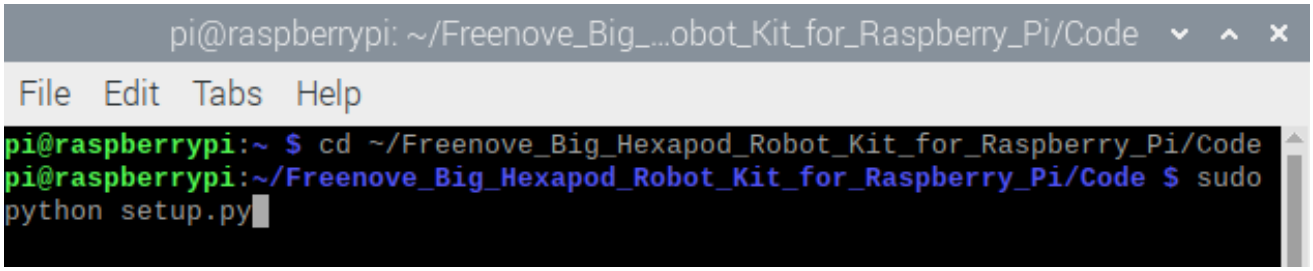
```
Processing dependencies for rpi-ws281x==4.2.6
Finished processing dependencies for rpi-ws281x==4.2.6
The installation is complete!
```

1. Execute the following command to enter the directory where robot code “setup.py” locates.

```
cd ~/Freenove_Big_Hexapod_Robot_Kit_for_Raspberry_Pi/Code
```

2.Run setup.py

```
sudo python setup.py
```



The installation program will automatically install pca9685 library, rpi_ws281x library and PyQt5 library, etc. When the installation is successful, please restart Raspberry Pi.

```
Now the installation is successful.

Please restart raspberry pi
```

If the installation fails as shown below

```
Some libraries have not been installed yet. Please run 'sudo python setup.py' again
```

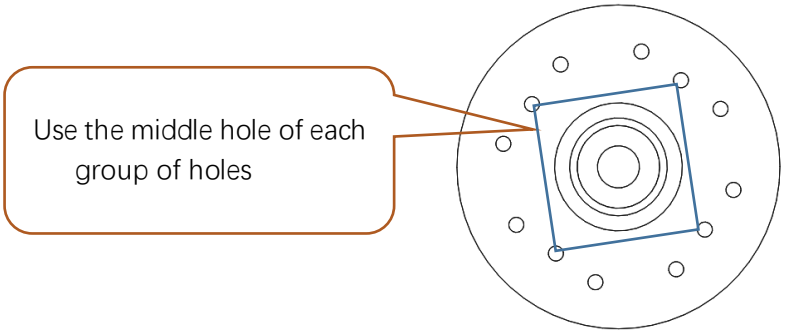
Our solution is: Re-execute setup.py. After installing successfully, restart Raspberry Pi. Most installation failures are caused by a poor network.

```
sudo python setup.py
```

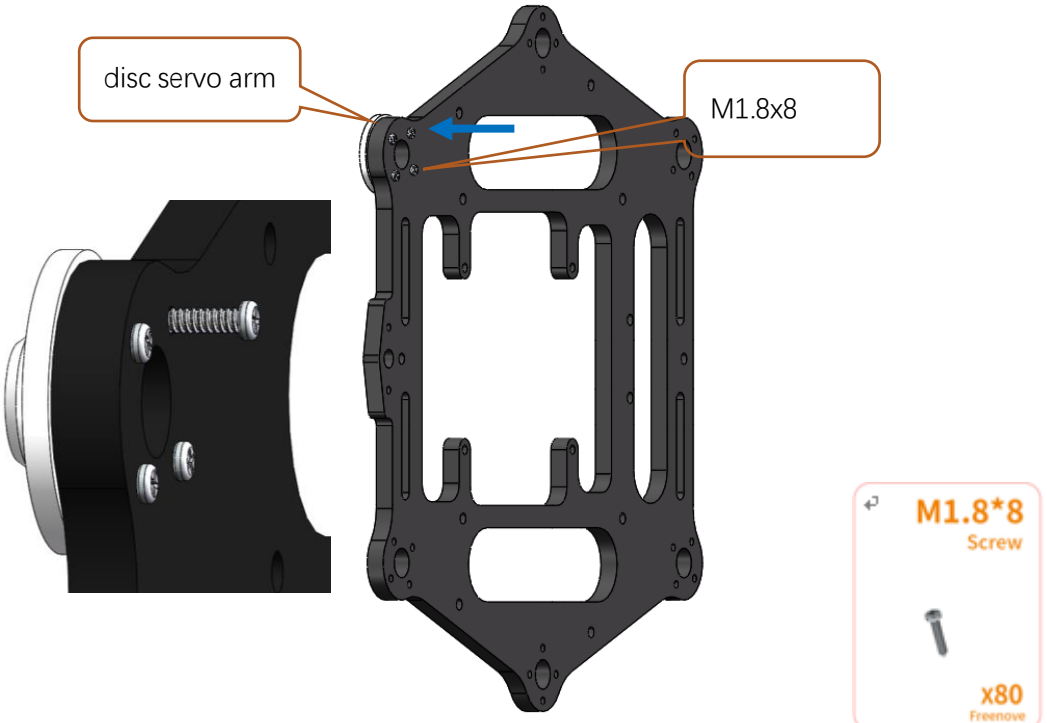
Chapter 2 Assembly of the Robot (Important)

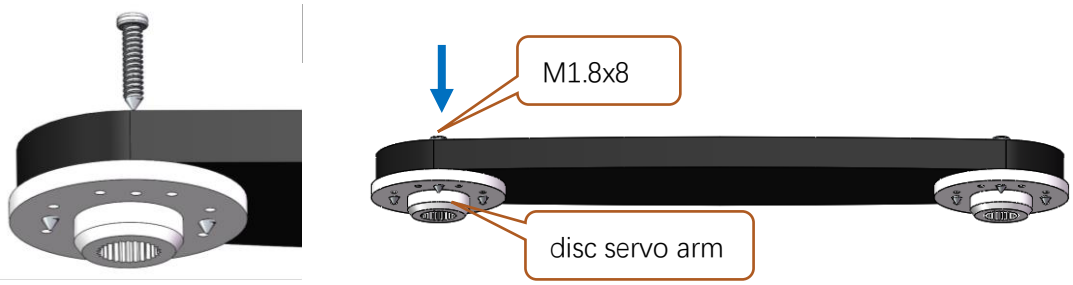
Step 1 Installing Servo Base

Take out the disc servo arms from the servo package.



Fix each disc servo arm to the joints.



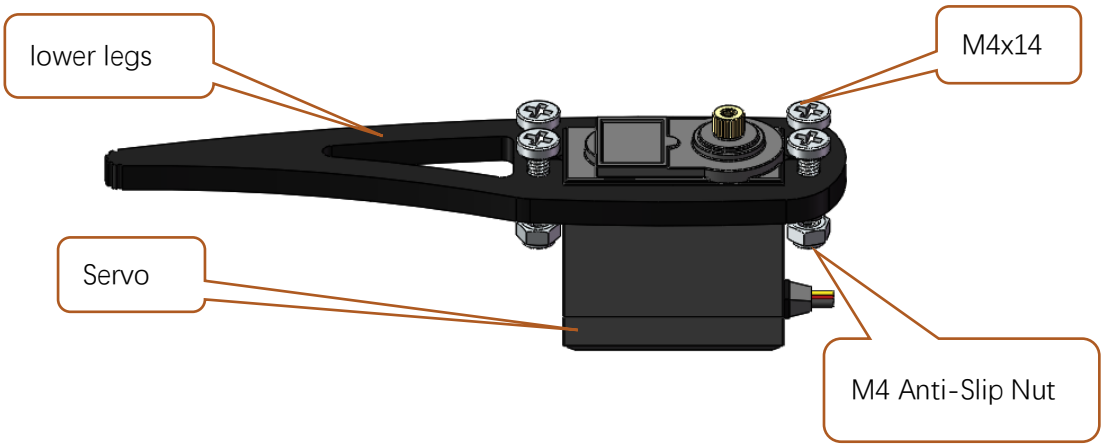


After all disc servo arms are installed successfully, it should look as follows:

Step 2 Installing Servos

Installing Servos on the Legs

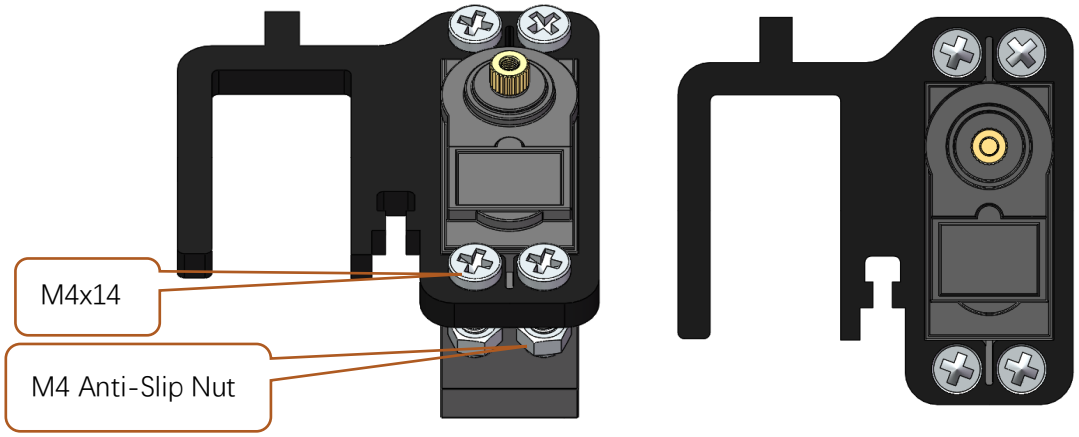
Install servos on the lower legs



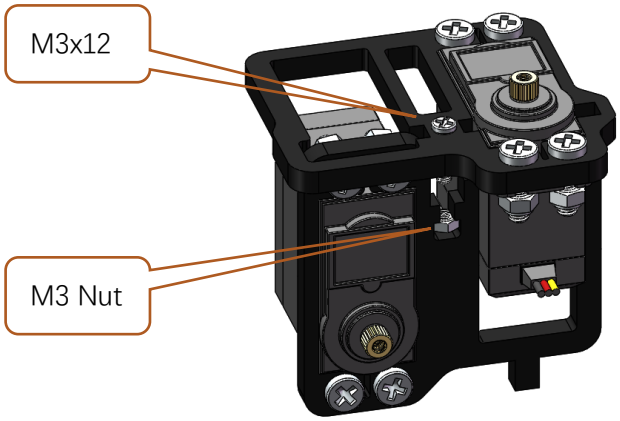
The other 5 leg parts are installed in the same way.

Note: The robot has six legs and its left lower legs and right lower legs are in different directions.

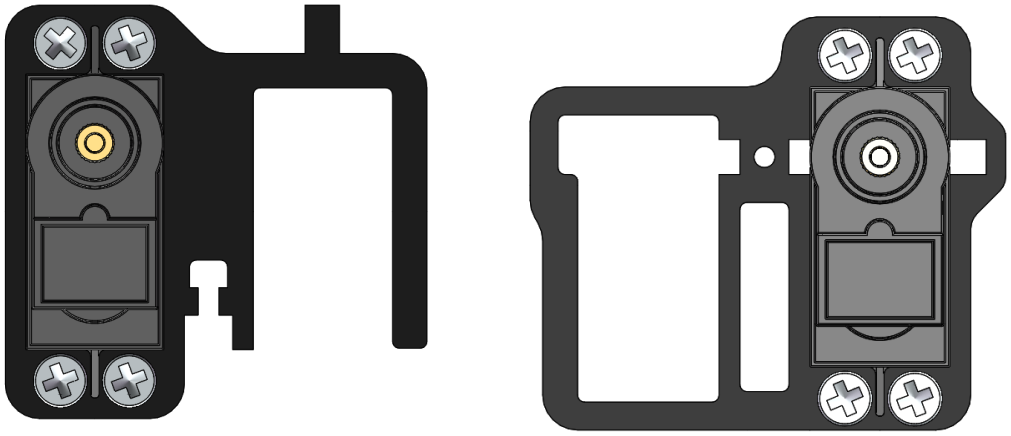
Installation of servos on the hip joints of the three left legs.



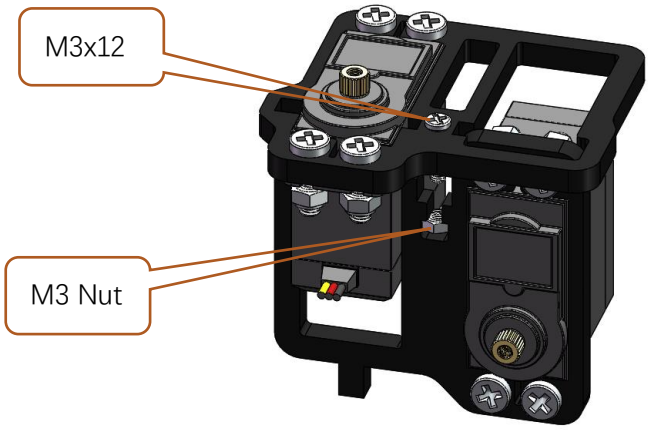
Assembly



Installation of servos on the hip joints of the three right legs.



Assembly

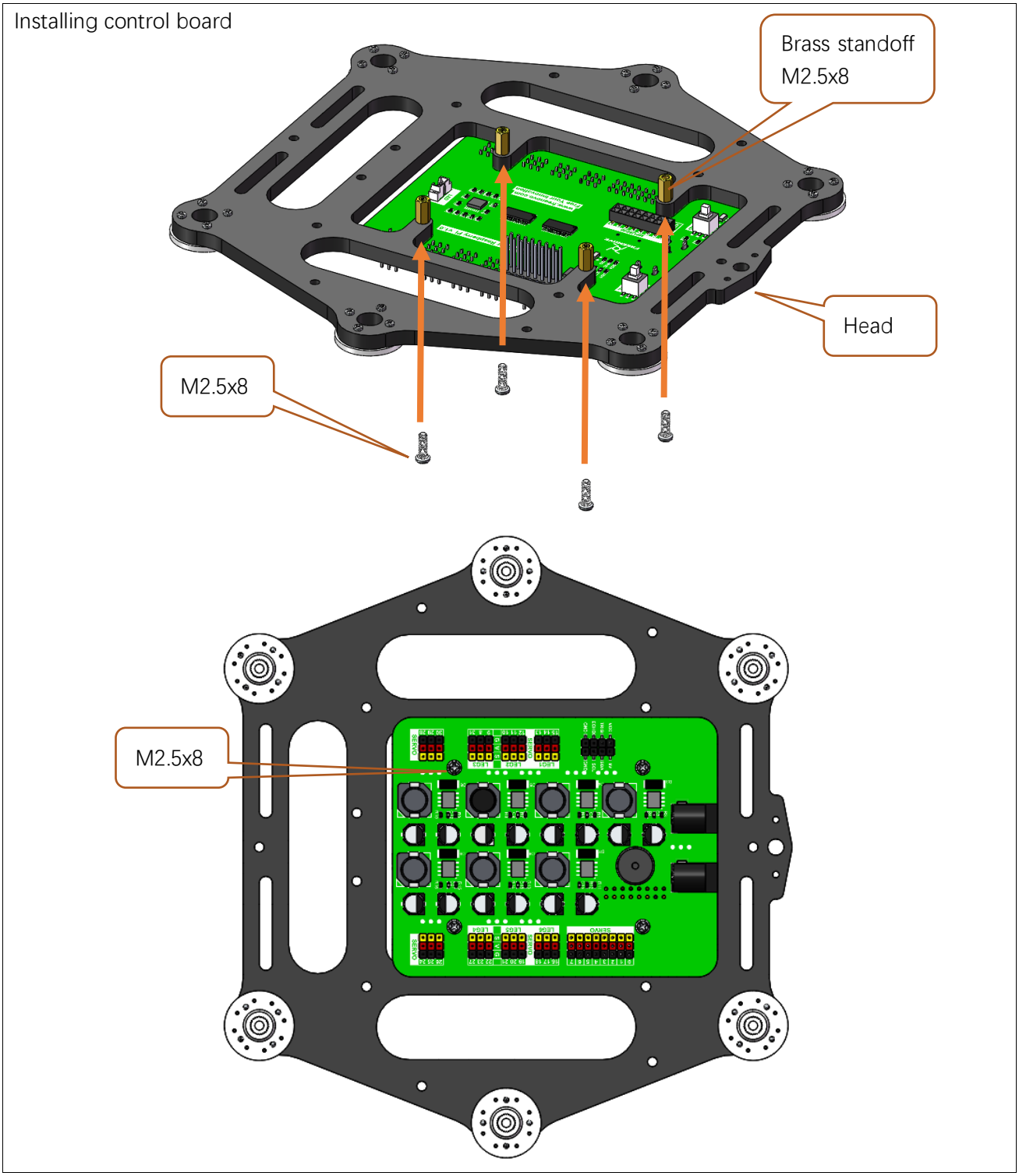


Servos at left and right hip joints (Note: the direction of servos on each side is different.)

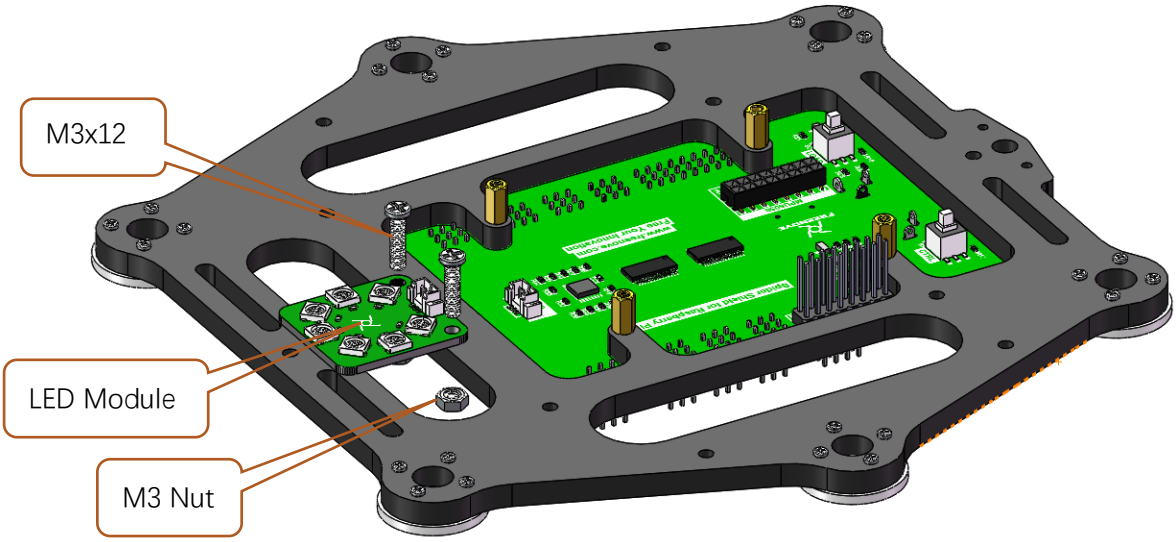
Step 3 Assembling the center of the body

Installing Control Board

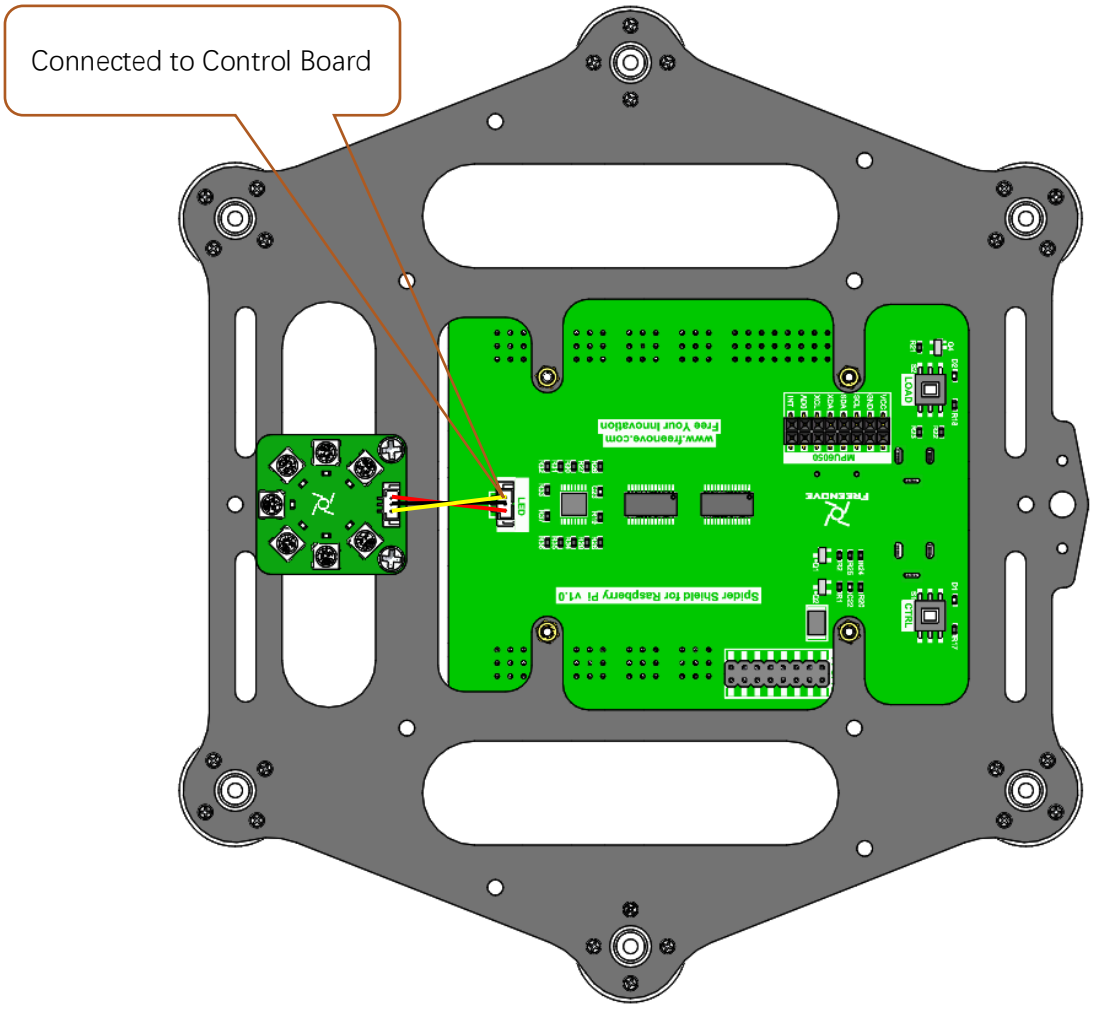
Installing control board



Installing LED Module

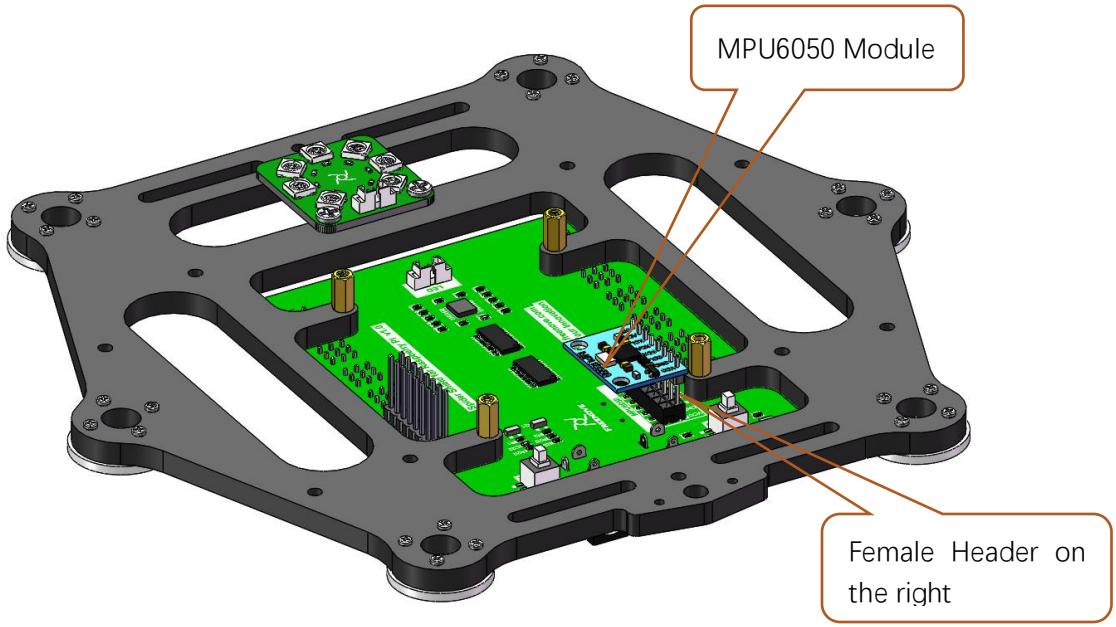


Wiring

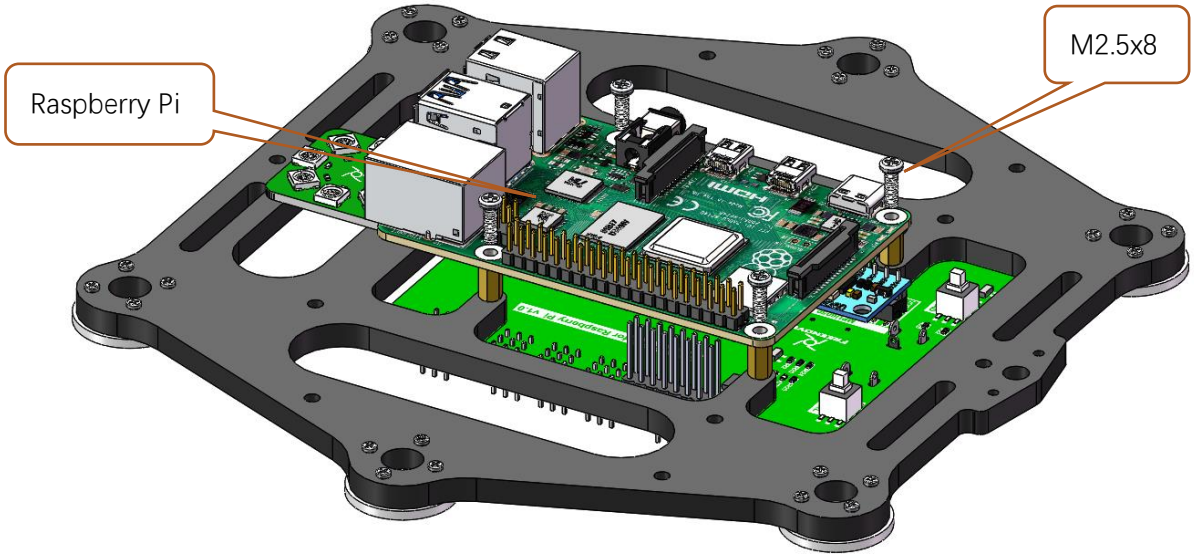


Installing MPU6050 Module

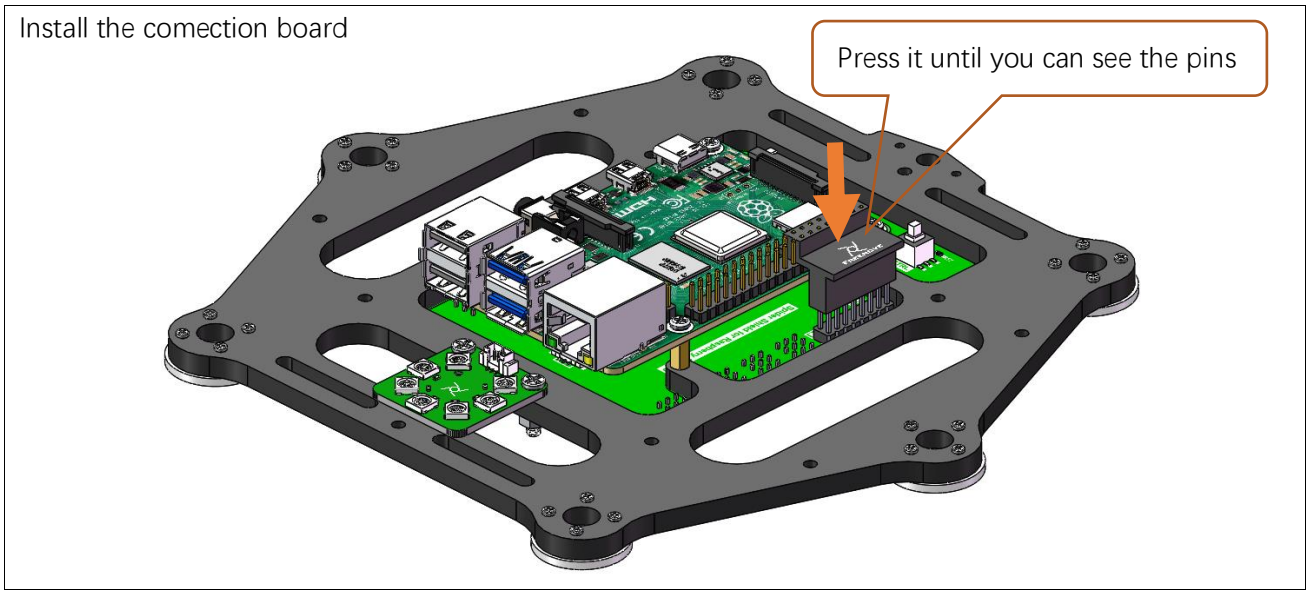
Note: There are two rows of female header, please plug the module into the right.



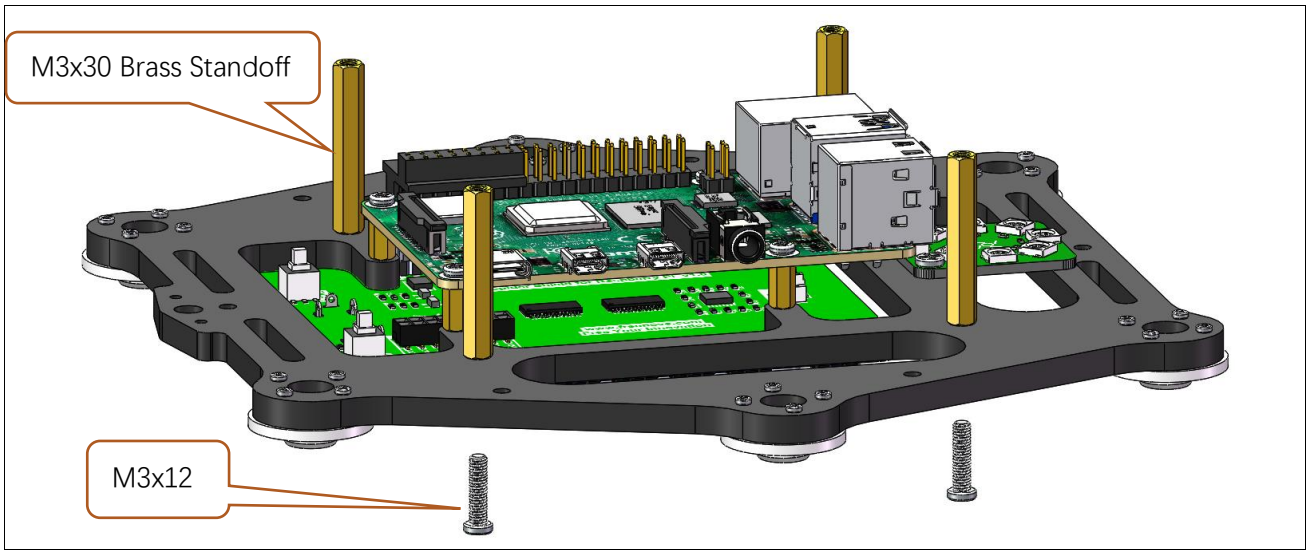
Installing Raspberry Pi



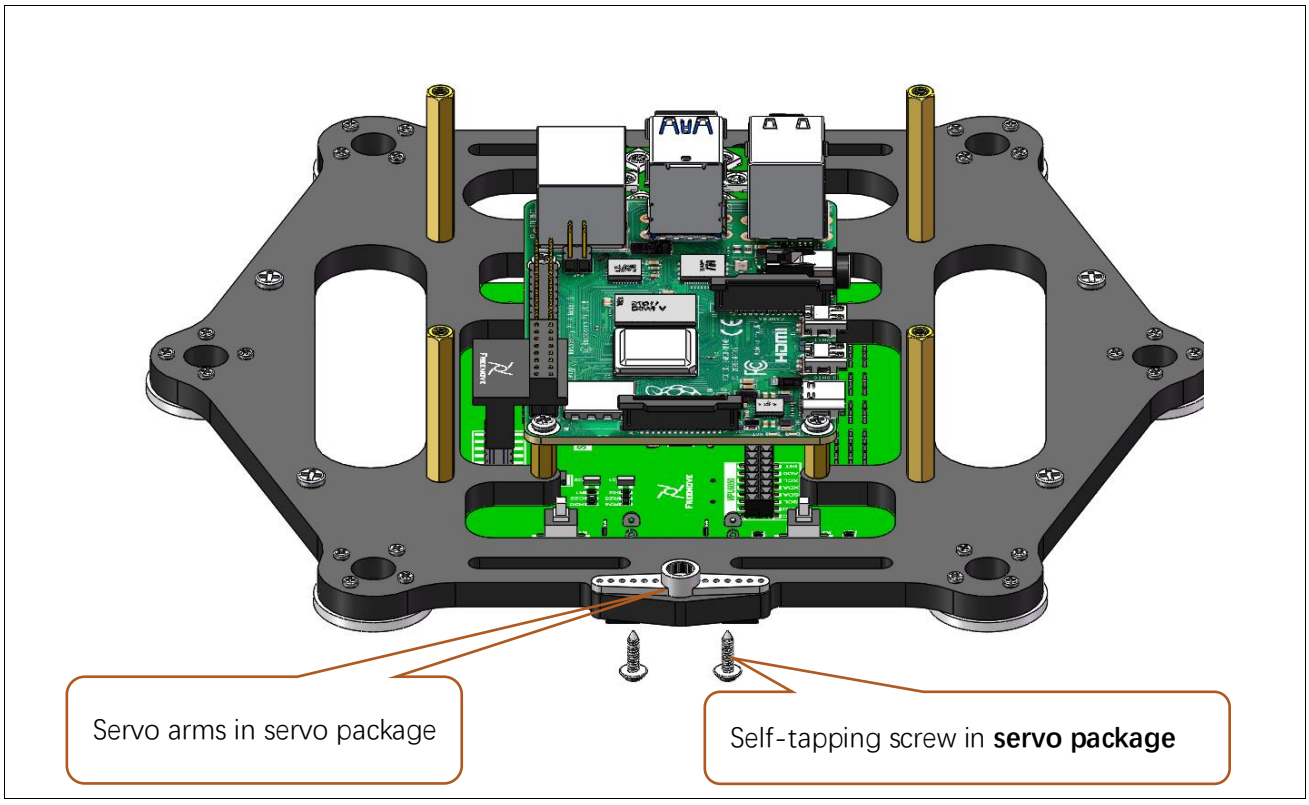
Installing Connector



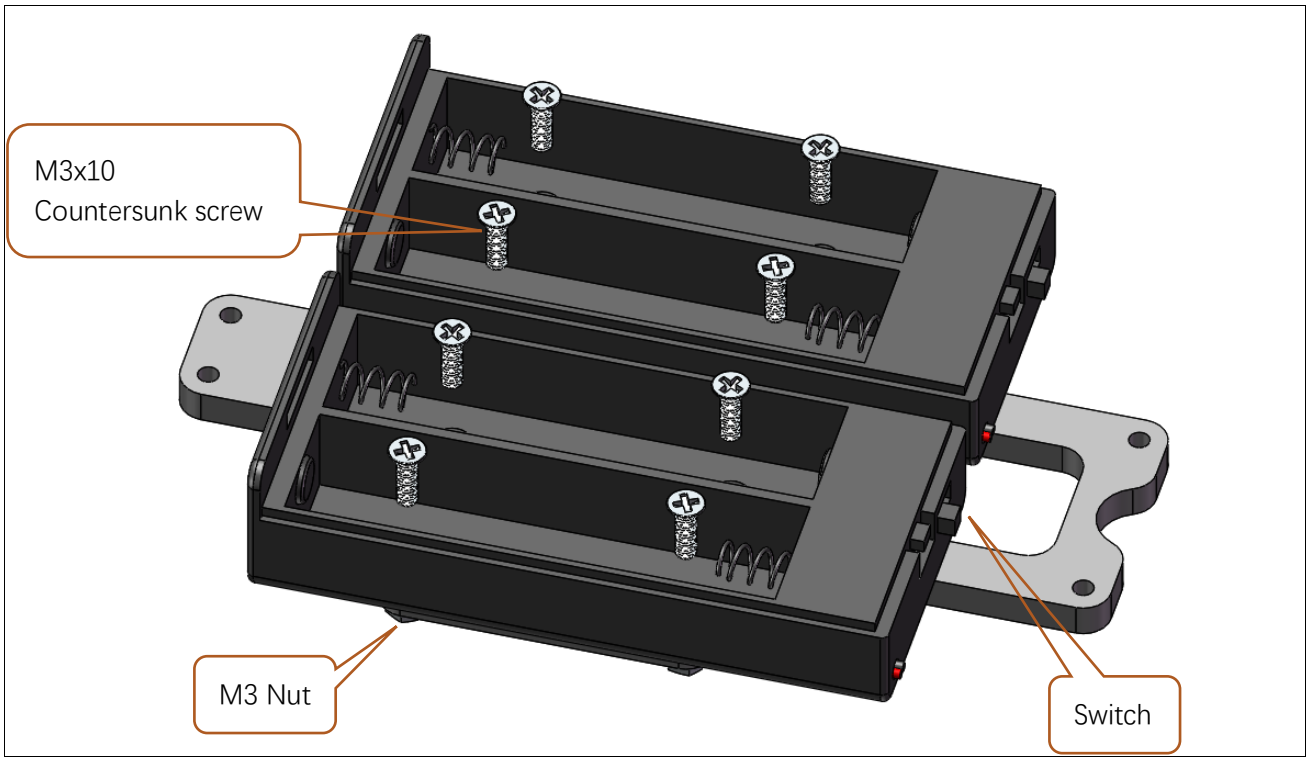
Installing Brass Standoff

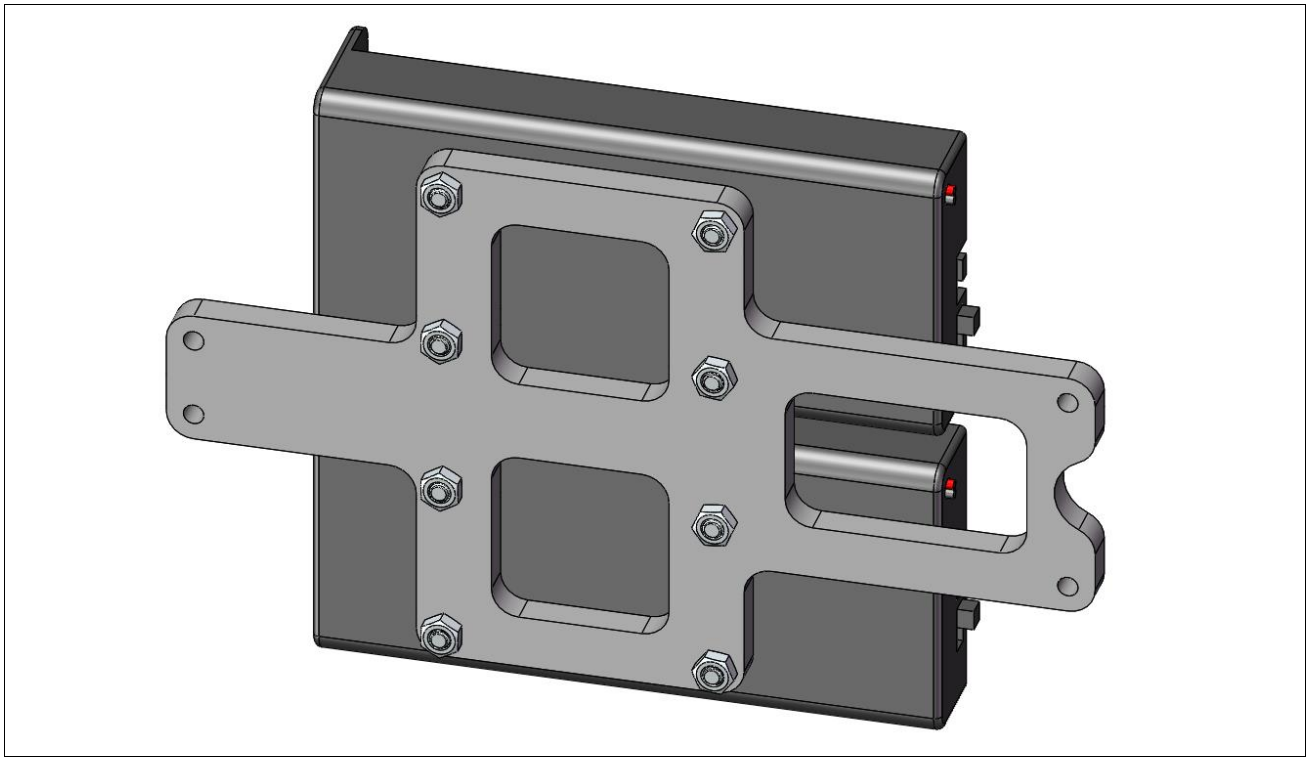


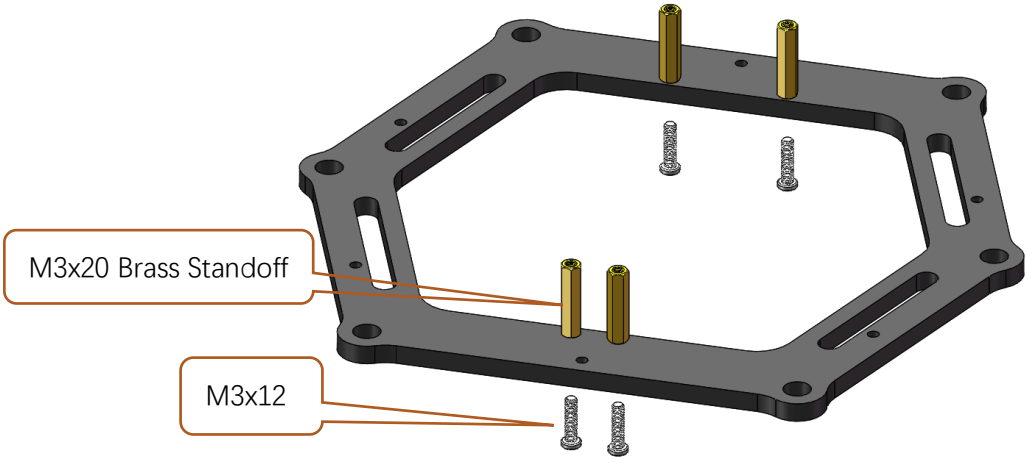
Installing Servo Arms

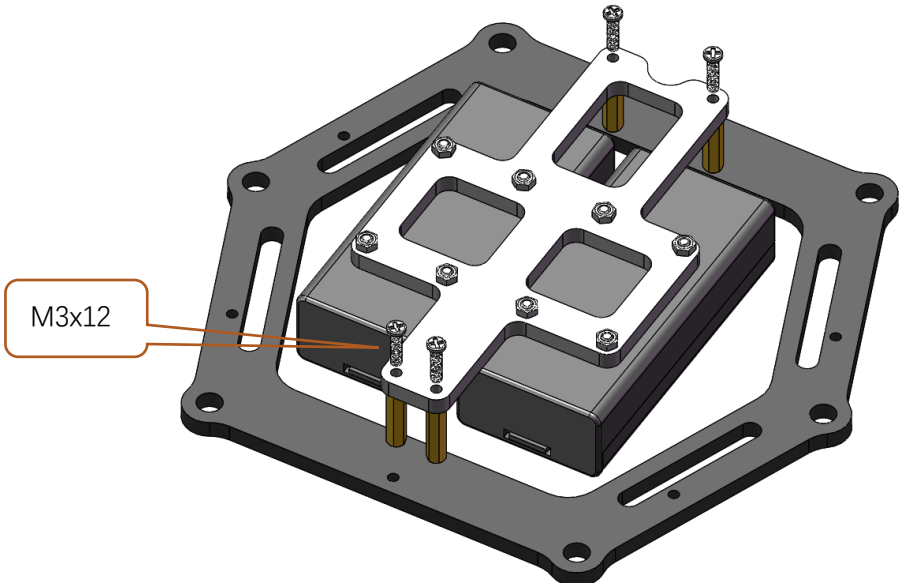


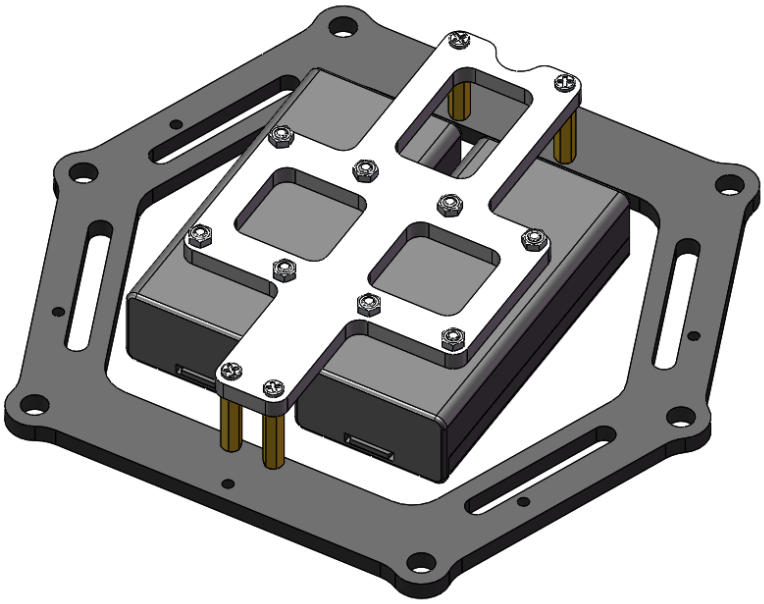
Step 4 Installing Battery Holder






M3x20 Brass Standoff
M3x12

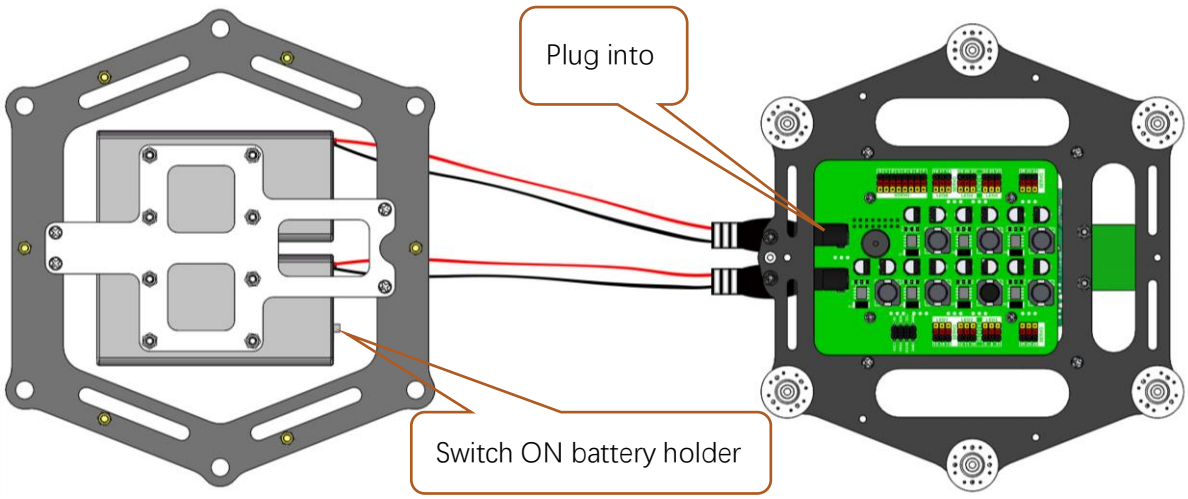

M3x12



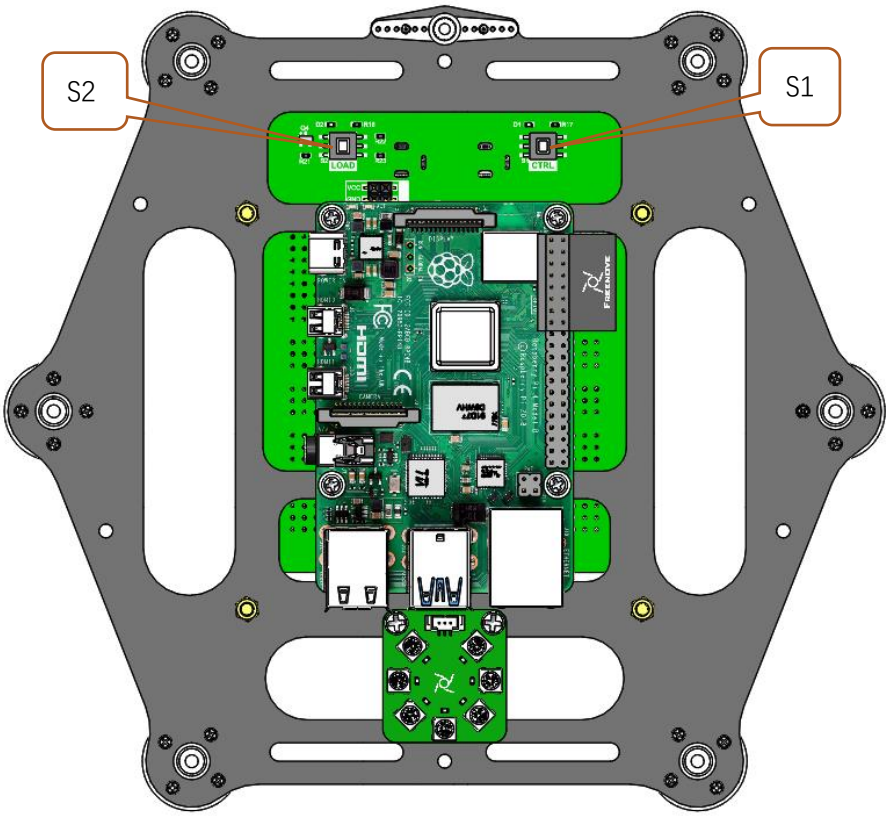
Step 5 Assembling Legs and Head to the Body

Run the servo installation program

Before installing, fix the servos at the specified angle. At this point you need to run the servo program.
Plug batteries into the battery holder and switch it on. And then plug the battery holder to control board.



Press S1 and S2.



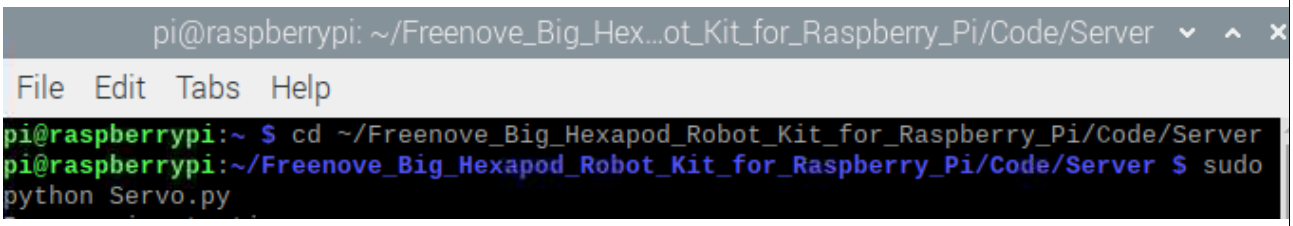
Open Raspberry Pi. (All contents in Chapter 1 should be executed correctly first. If not, please return Chapter 1 to finish it first.)

Enter the following command to enter the directory where the servo program locates.

```
cd ~/Freenove_Big_Hexapod_Robot_Kit_for_Raspberry_Pi/Code/Server
```

b. Execute the following command to run "Servo.py".

```
sudo python Servo.py
```

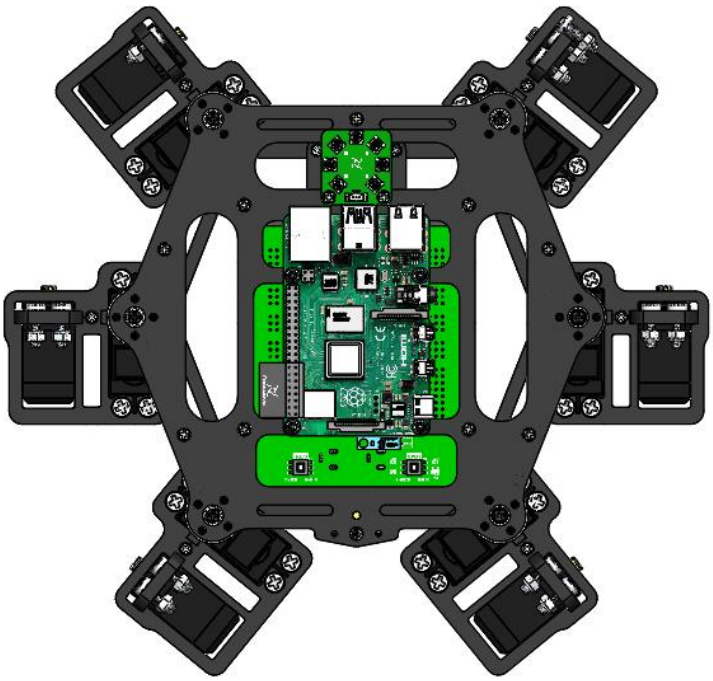


Next, fix each servo to the corresponding disc servo arm. Do NOT pull the servo wires during the installation process.

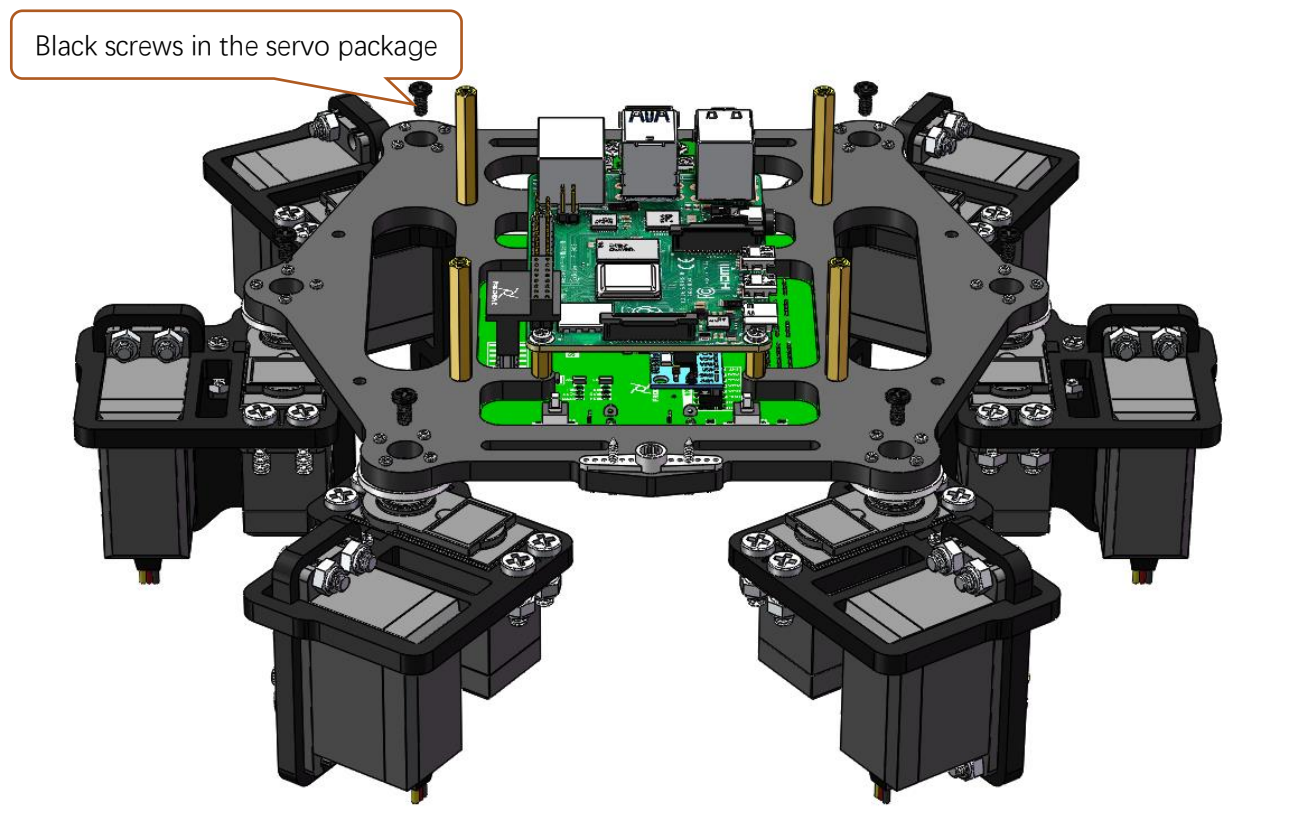
Installing Hip Joint

Connect the servo wires to corresponding servo port.

Bottom

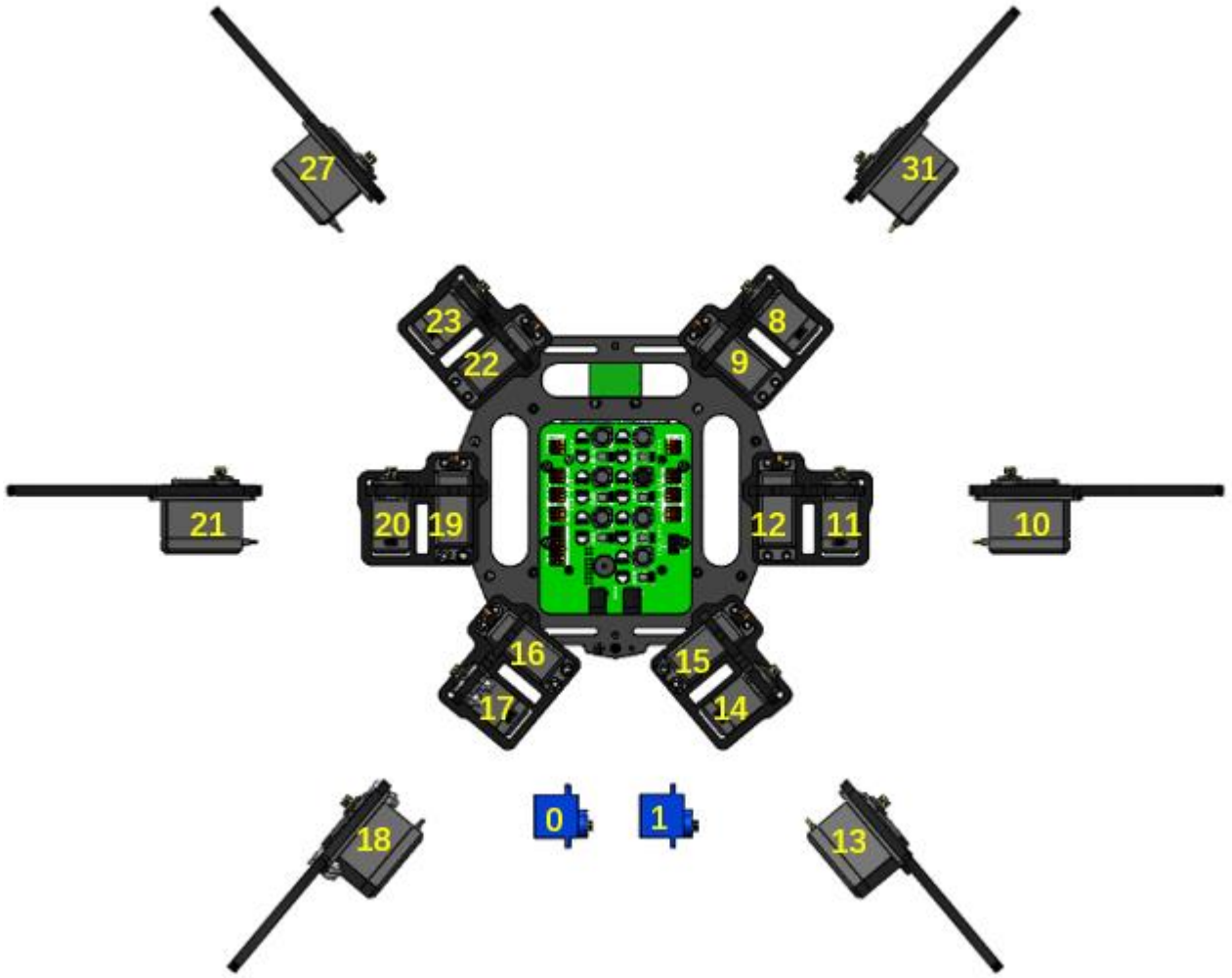


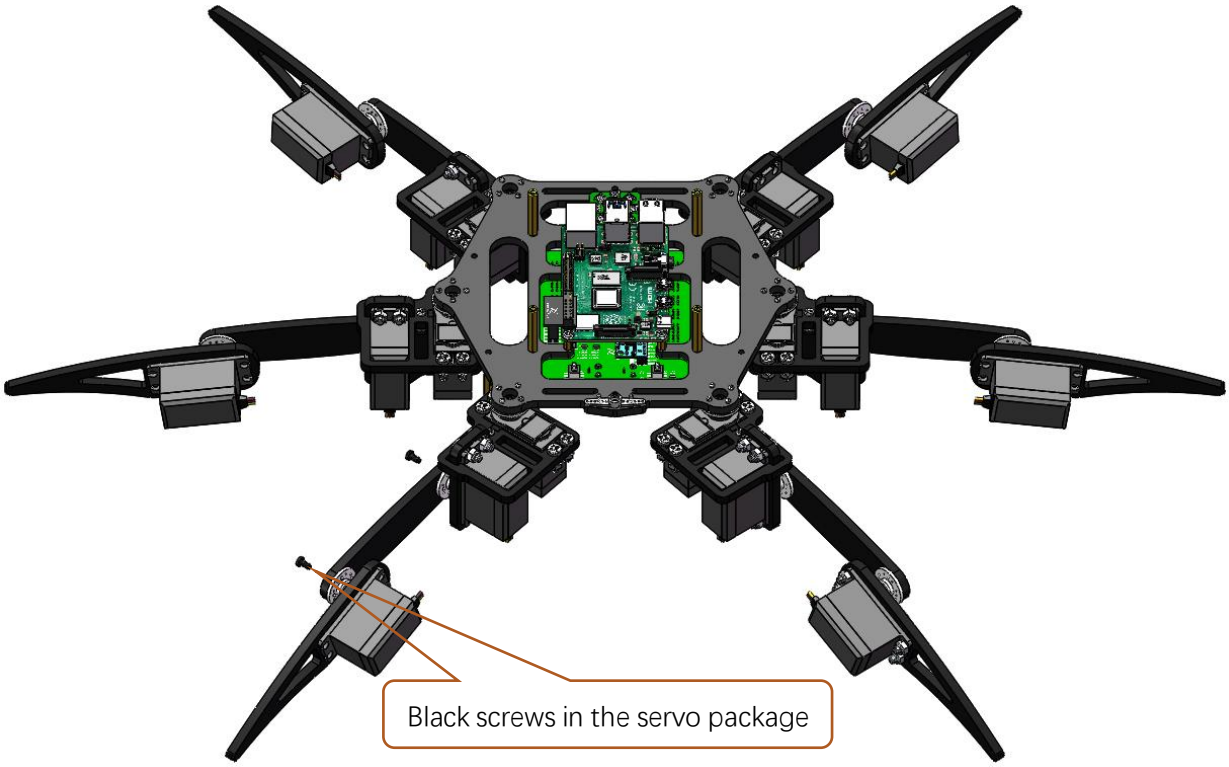
Top


Black screws in the servo package

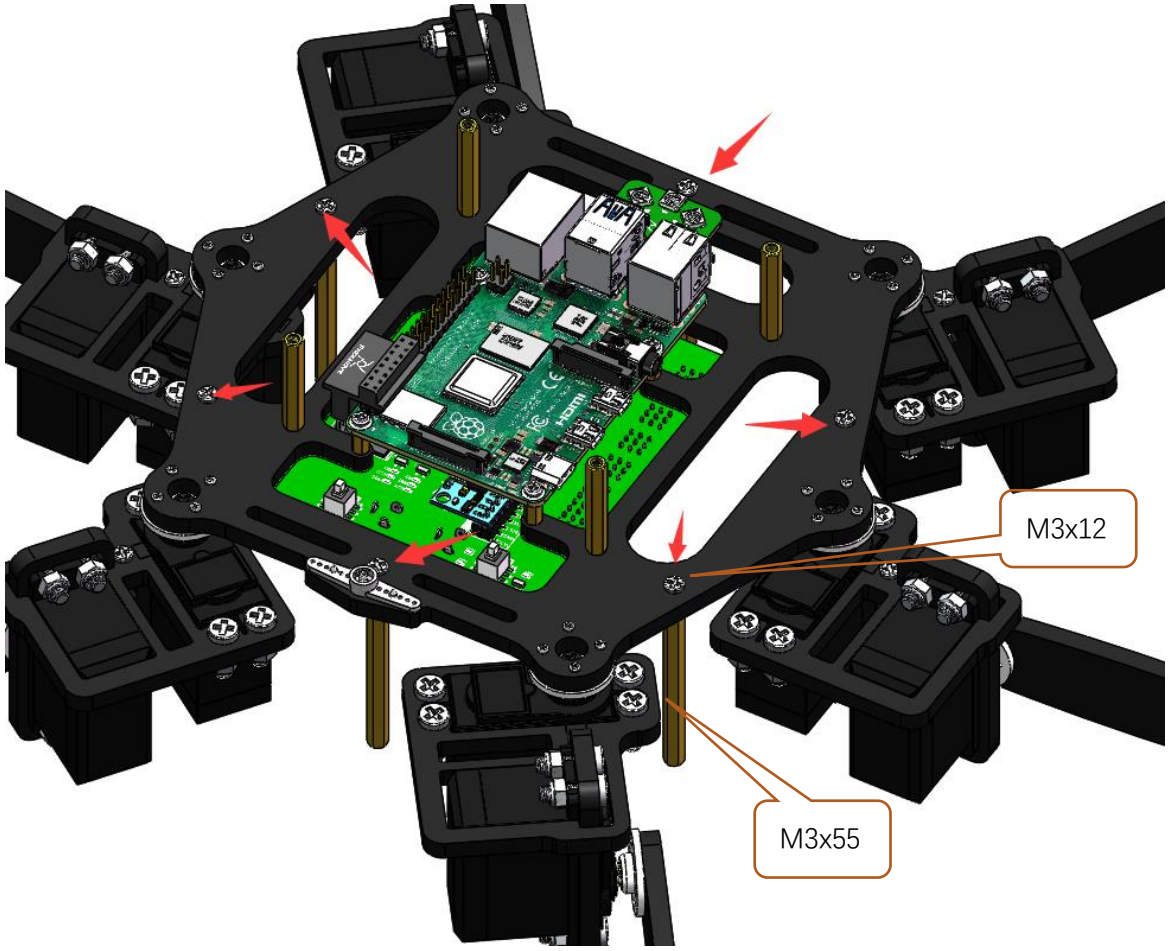
Installing Lower Leg and the Head

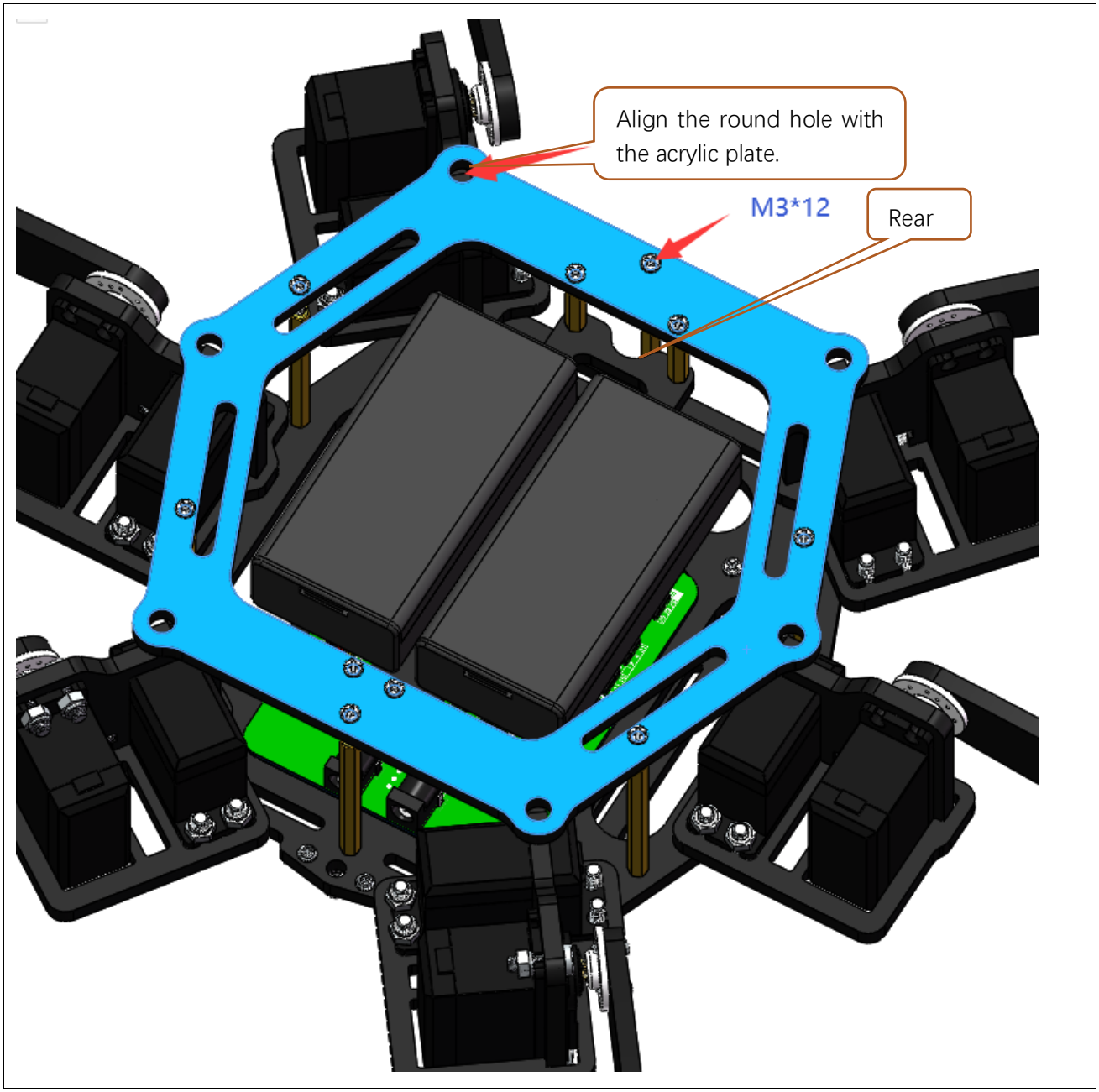
Connect the servo wires to corresponding servo port.



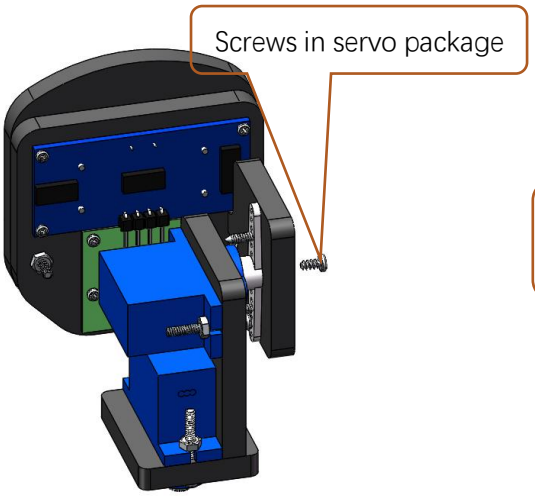


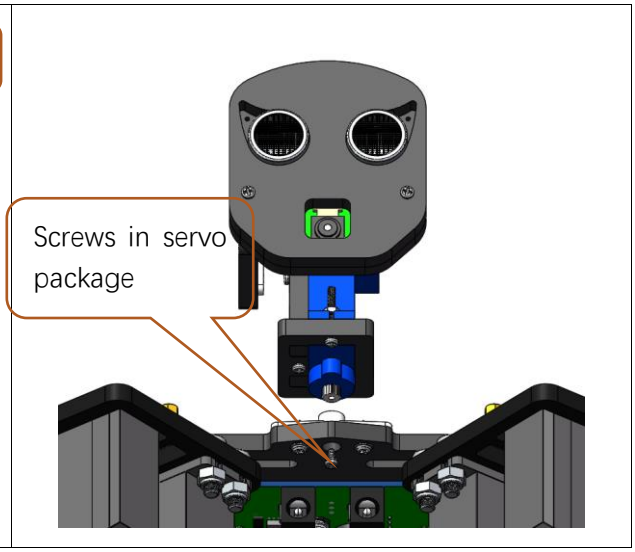
Asseble 6 standoff




Align the round hole with the acrylic plate.
M3*12
Rear

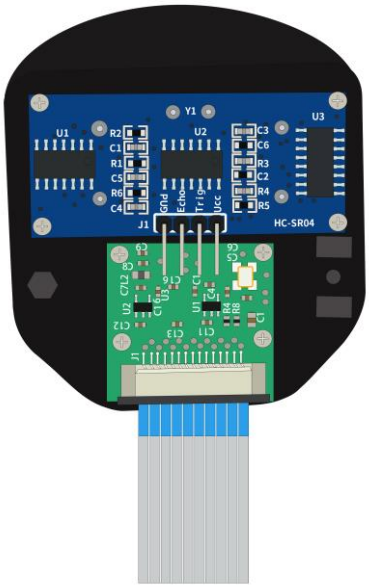
Installing head

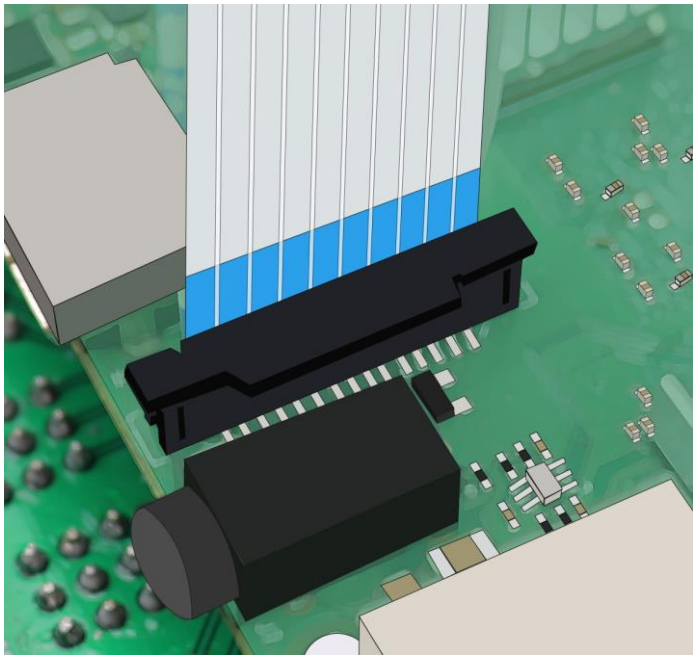




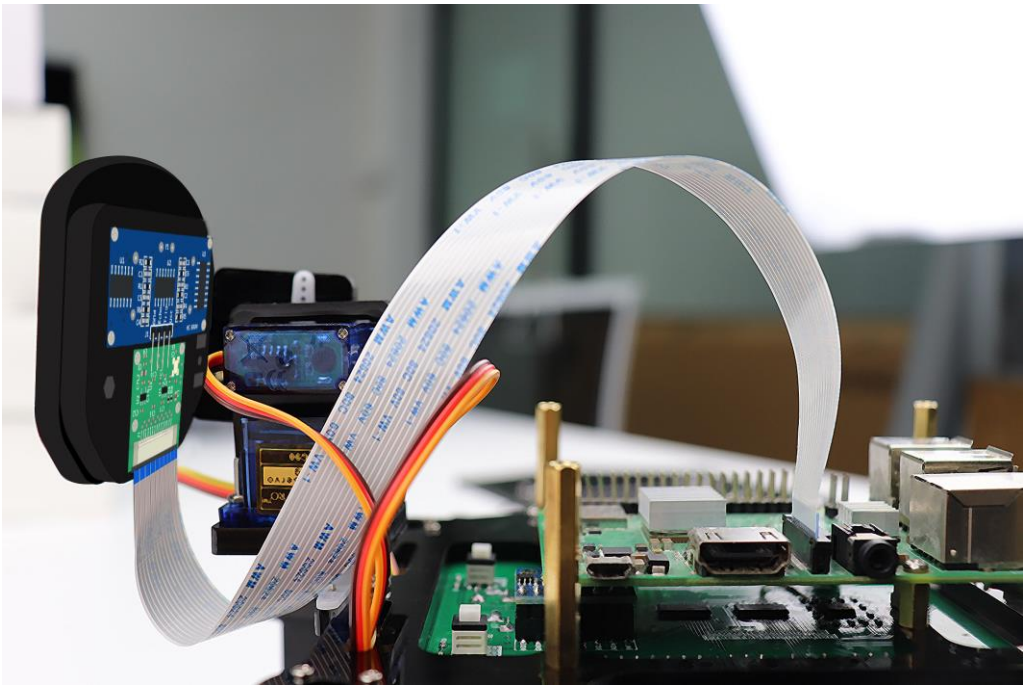
Connecting Wires of Camera and Ultrasonic Module

Connect camera wire



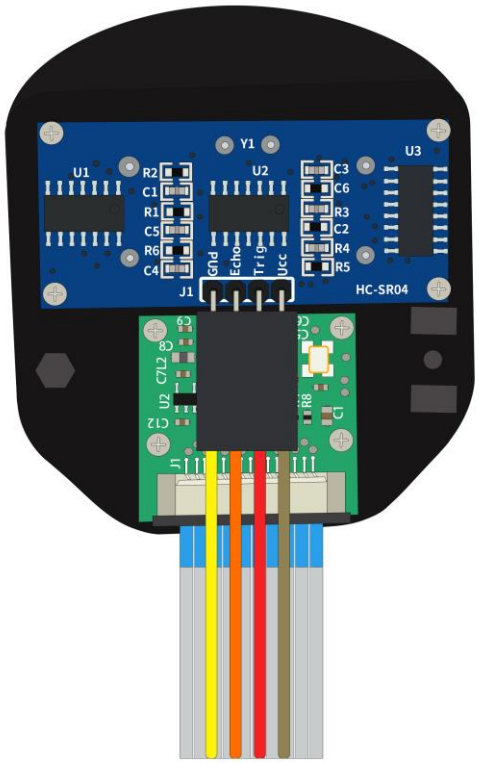


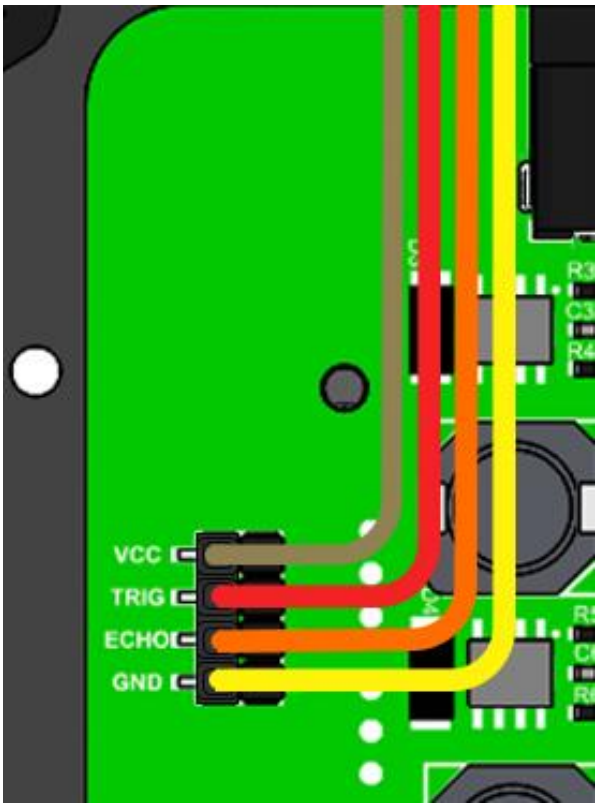
After connected, it should be as follows



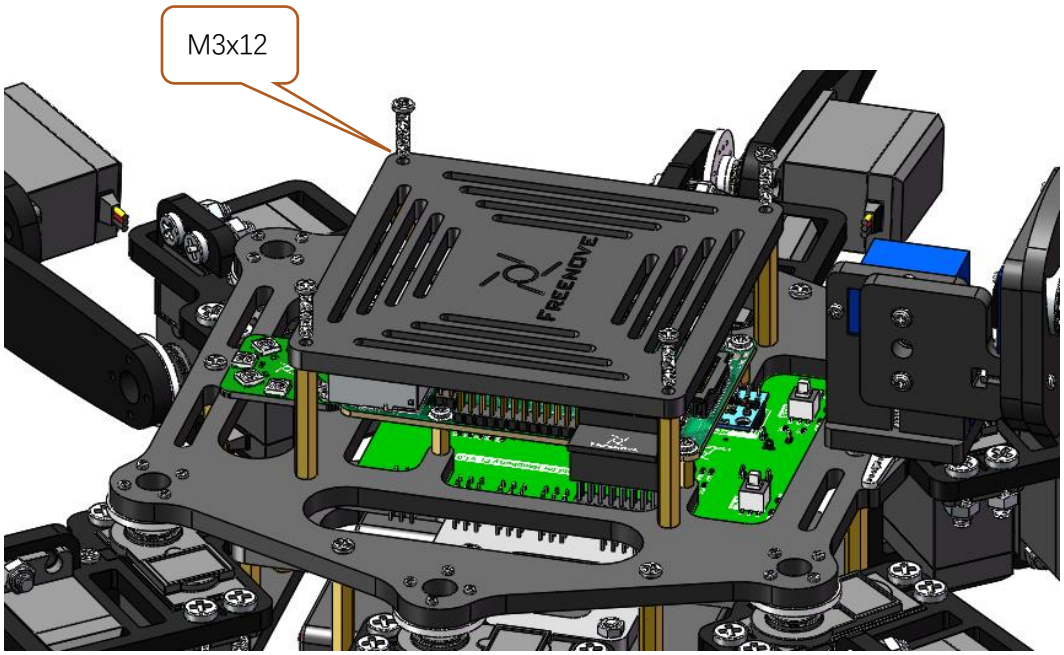
Connecting Ultrasonic Module

Note: VCC, Echo, Trig, GND are connected to VCC, Echo, Trig and GND on the connection board respectively. Do NOT connect them wrongly. Otherwise, it may burn out the Ultrasonic module.





Step 6 Installing Base and Cover Plates



Congratulations! You have finished the assembly of the robot.

Chapter 3 Module Test (necessary)

In the previous chapter, we have assembled the robot. In this chapter, we will test all modules of the robot to see whether they work.

Before starting, please plug in the batteries, and switch ON S1 and S2. Place the assembled robot on a horizontal table and make sure that the wiring of each servo is correct.

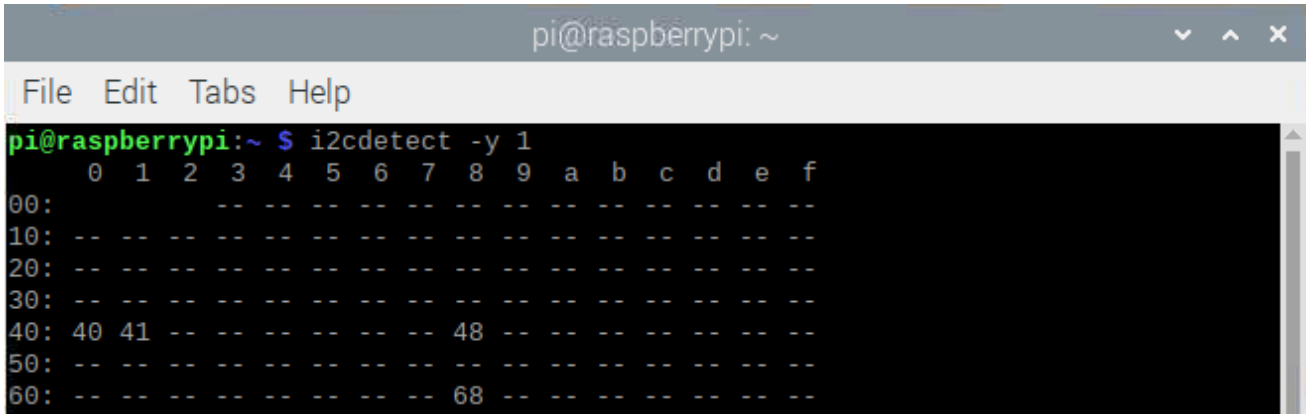
If you have never learned python before, you can learn some basic knowledge via the link below: https://python.swaroopch.com/basics.html

Servo

Enter the following command to check I2C.

```
i2cdetect -y 1
```

As shown in the following illustration, the address of two PCA9685 chips are 0x40 and 0x41 respectively, the address of ADS7830 chip is 0x48 and the address of MPU6050 module is 0x68。



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.

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

1. If not, execute the cd command:

```
cd ~/Freenove_Big_Hexapod_Robot_Kit_for_Raspberry_Pi/Code/Server
```

2. Execute test.py command:

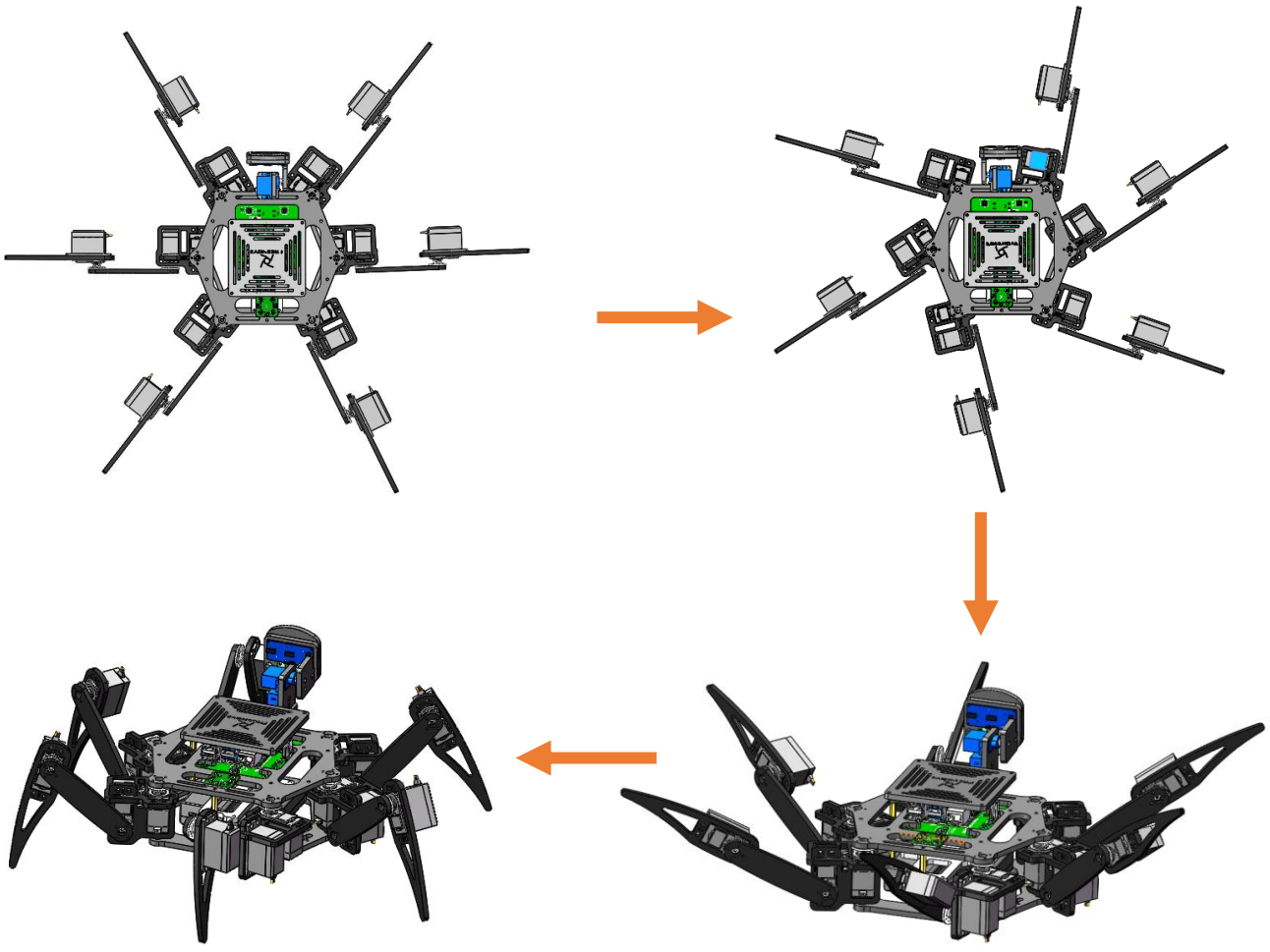
```
sudo python test.py Servo
```

```
pi@raspberrypi: ~/Freenove_Big_Hex...ot_Kit_for_Raspberry_Pi/Code/Server
File  Edit  Tabs  Help
pi@raspberrypi:~ $ cd ~/Freenove_Big_Hexapod_Robot_Kit_for_Raspberry_Pi/Code/Server
pi@raspberrypi:~/Freenove_Big_Hexapod_Robot_Kit_for_Raspberry_Pi/Code/Server $ sudo
python test.py Servo
Program is starting ...
```

Result:

In the previous chapter, we have assembled the robot and executed the servo installation program, which makes the six legs of the robot straight.

After running the servo test program, the robot slowly rotates its six legs clockwise from the original straight state, and then lifts the legs up, and finally stands with the lower leg extended back. If the sequence of the movement of your robot is different, please check whether the wiring is correct.



The code is as below:

```
from Servo import *
servo=Servo()
def test_Servo():
    try:
        for i in range(50):
            servo.setServoAngle(15,90+i)
            servo.setServoAngle(12,90+i)
            servo.setServoAngle(9,90+i)
            servo.setServoAngle(16,90+i)
            servo.setServoAngle(19,90+i)
            servo.setServoAngle(22,90+i)
            time.sleep(0.05)
        for i in range(60):
            servo.setServoAngle(14,90+i)
            servo.setServoAngle(11,90+i)
            servo.setServoAngle(8,90+i)
            servo.setServoAngle(17,90-i)
            servo.setServoAngle(20,90-i)
            servo.setServoAngle(23,90-i)
            time.sleep(0.05)
        for i in range(150):
            servo.setServoAngle(13,i)
            servo.setServoAngle(10,i)
            servo.setServoAngle(31,i)
            servo.setServoAngle(18,180-i)
            servo.setServoAngle(21,180-i)
            servo.setServoAngle(27,180-i)
            time.sleep(0.05)
        print ("\nEnd of program")
    except KeyboardInterrupt:
        print ("\nEnd of program")
```

Reference

setServoAngle(channel, angle)

There are two parameters in this function to rotate the servo of specified channel to a specified angle.
The first parameter indicates the servo channel
The second parameter indicates the rotation angle
For example:

setServoAngle(0,60)	Rotate servo of channel 0 to 60 degrees
setServoAngle(15,160)	Rotate servo of channel 15 to 160 degrees

ADC Module

Run program

Enter following commands to test ADC module.

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

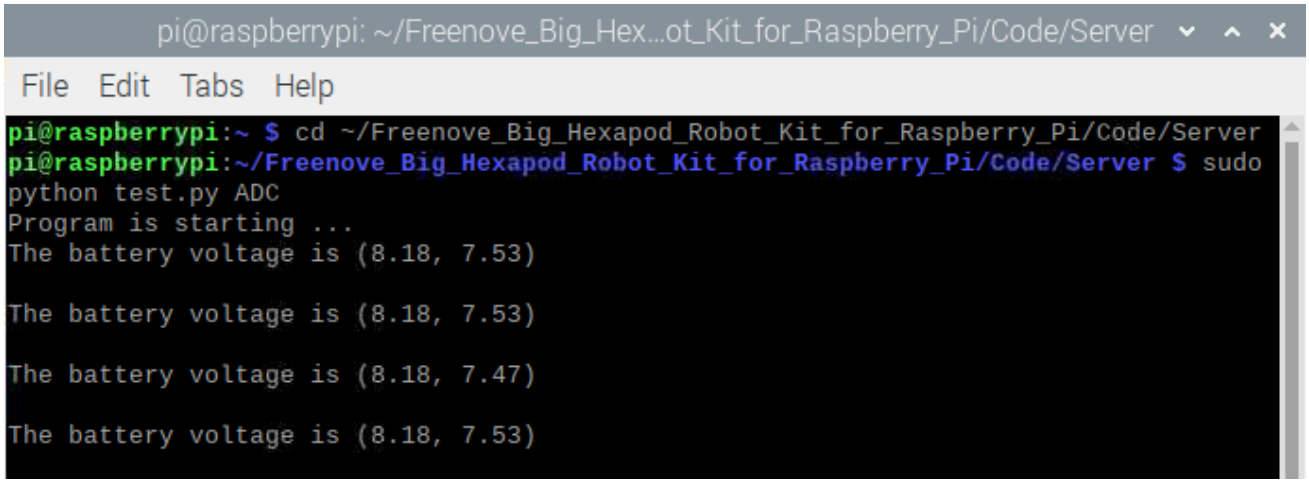
```
pi@raspberrypi:~/Freenove_Big_Hexapod_Robot_Kit_for_Raspberry_Pi/Code/Server $
```

1.If not, execute the cd command:

```
cd ~/Freenove_Big_Hexapod_Robot_Kit_for_Raspberry_Pi/Code/Server
```

2.Execute test.py command:

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



Result:

Print out the voltage value of two sets of batteries every one second,You can press "Ctrl + C" to end program.

The code is as below:

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

Reference

batteryPower()
Return the voltage value of the battery, retaining two decimal places.

Ultrasonic module

Run program

Enter the following command in the terminal to test the ultrasonic module:

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

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

1. If not, execute the cd command:

```
cd ~/Freenove_Big_Hexapod_Robot_Kit_for_Raspberry_Pi/Code/Server
```

2. Execute test.py command:

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

```
pi@raspberrypi: ~/Freenove_Big_Hex...ot_Kit_for_Raspberry_Pi/Code/Server
File  Edit  Tabs  Help
pi@raspberrypi:~ $ cd ~/Freenove_Big_Hexapod_Robot_Kit_for_Raspberry_Pi/Code/Server
pi@raspberrypi:~/Freenove_Big_Hexapod_Robot_Kit_for_Raspberry_Pi/Code/Server $ sudo
python test.py Ultrasonic
Program is starting ...
Obstacle distance is 32CM
Obstacle distance is 32CM
Obstacle distance is 9CM
Obstacle distance is 15CM
Obstacle distance is 14CM
Obstacle distance is 7CM
Obstacle distance is 31CM
^C
End of program
pi@raspberrypi:~/Freenove_Big_Hexapod_Robot_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:

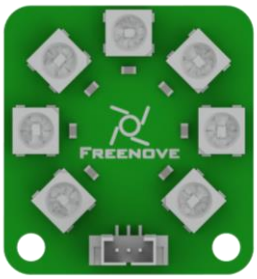
```
1  from Ultrasonic import *
2  ultrasonic=Ultrasonic()
3  def test_Ultrasonic():
4      try:
5          while True:
6              data=ultrasonic.getDistance()   #Get the value
7              print ("Obstacle distance is "+str(data)+"CM")
8              time.sleep(1)
9      except KeyboardInterrupt:
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 an LED module on the robot, as shown 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_Big_Hexapod_Robot_Kit_for_Raspberry_Pi/Code/Server $
```

1.If not, execute the cd command:

```
cd ~/Freenove_Big_Hexapod_Robot_Kit_for_Raspberry_Pi/Code/Server
```

2.Execute test.py command:

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

```
pi@raspberrypi: ~/Freenove_Big_Hex...ot_Kit_for_Raspberry_Pi/Code/Server
File  Edit  Tabs  Help
pi@raspberrypi:~ $ cd ~/Freenove_Big_Hexapod_Robot_Kit_for_Raspberry_Pi/Code/Server
pi@raspberrypi:~/Freenove_Big_Hexapod_Robot_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_Big_Hexapod_Robot_Kit_for_Raspberry_Pi/Code/Server $
```

Result:

The LED lights up with the color of red, green, blue and white in turn and then it turns 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
        print ("\nRed wipe")
        led.colorWipe(led.strip, Color(255, 0, 0))
        time.sleep(1)


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


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


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

        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.

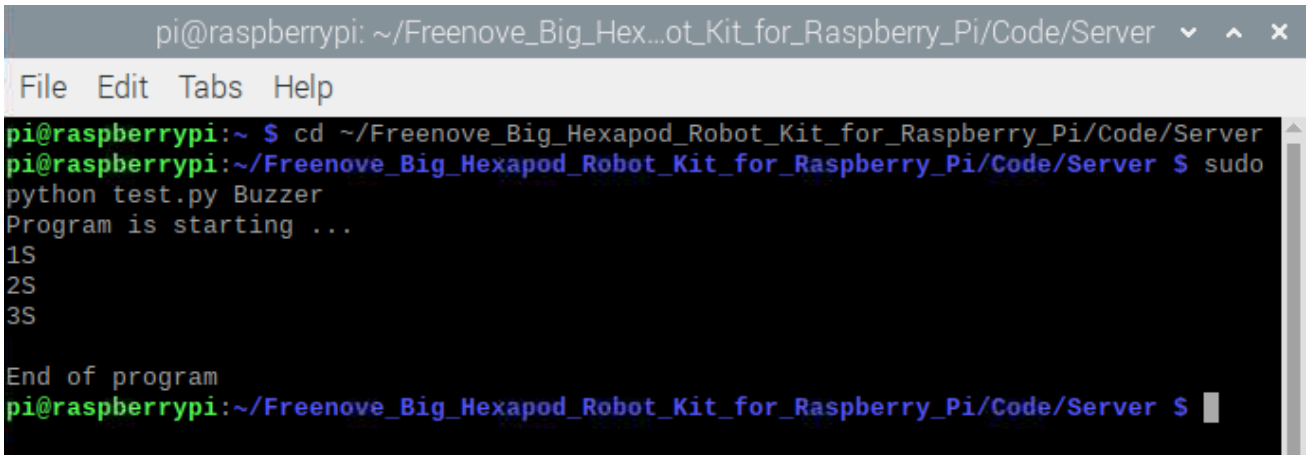
```
pi@raspberrypi:~/Freenove_Big_Hexapod_Robot_Kit_for_Raspberry_Pi/Code/Server $
```

1.If not, execute the cd command:

```
cd ~/Freenove_Big_Hexapod_Robot_Kit_for_Raspberry_Pi/Code/Server
```

2.Execute test.py command:

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



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)

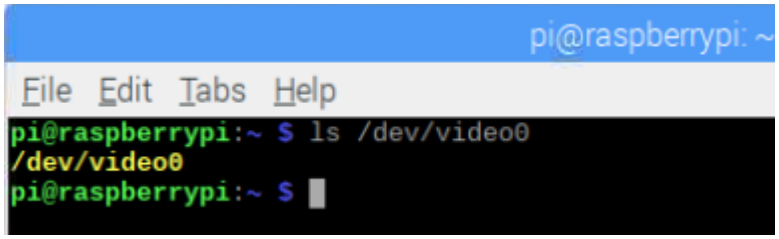
This function has one input parameter. If the input is '1', the buzzer will be turned on. If the input is '0', the buzzer will be turned off.

Camera

Enter the following command to check whether the camera is installed successfully:

```
ls /dev/video0
```

Then the device node will be shown below:



If you do not get the result shown in the above illustration, please check whether the camera wire is connected correctly and whether Camera has been enabled.

(Note: When plugging in or pulling out camera wire, please make sure Raspberry Pi is turned OFF. Otherwise, it may burn out the camera.)

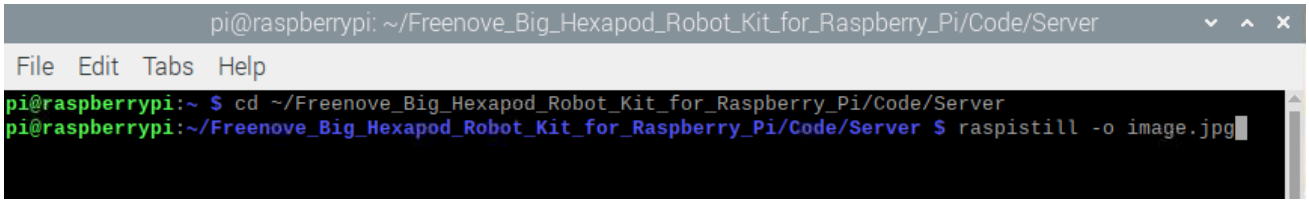
Run program

1. execute the cd command:

```
cd ~/Freenove_Big_Hexapod_Robot_Kit_for_Raspberry_Pi/Code/Server
```

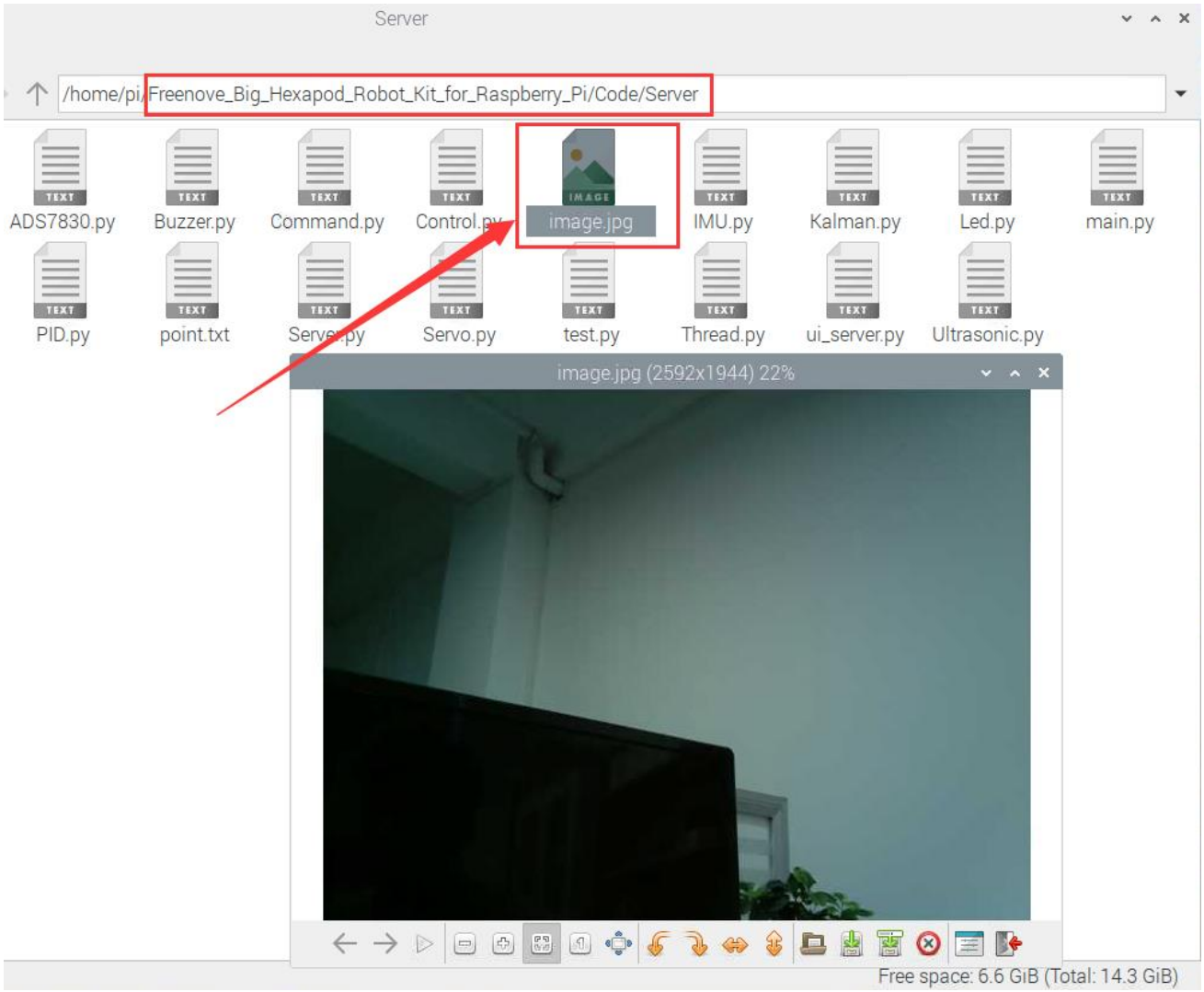
2. Execute test.py command:

```
raspistill -o image.jpg
```



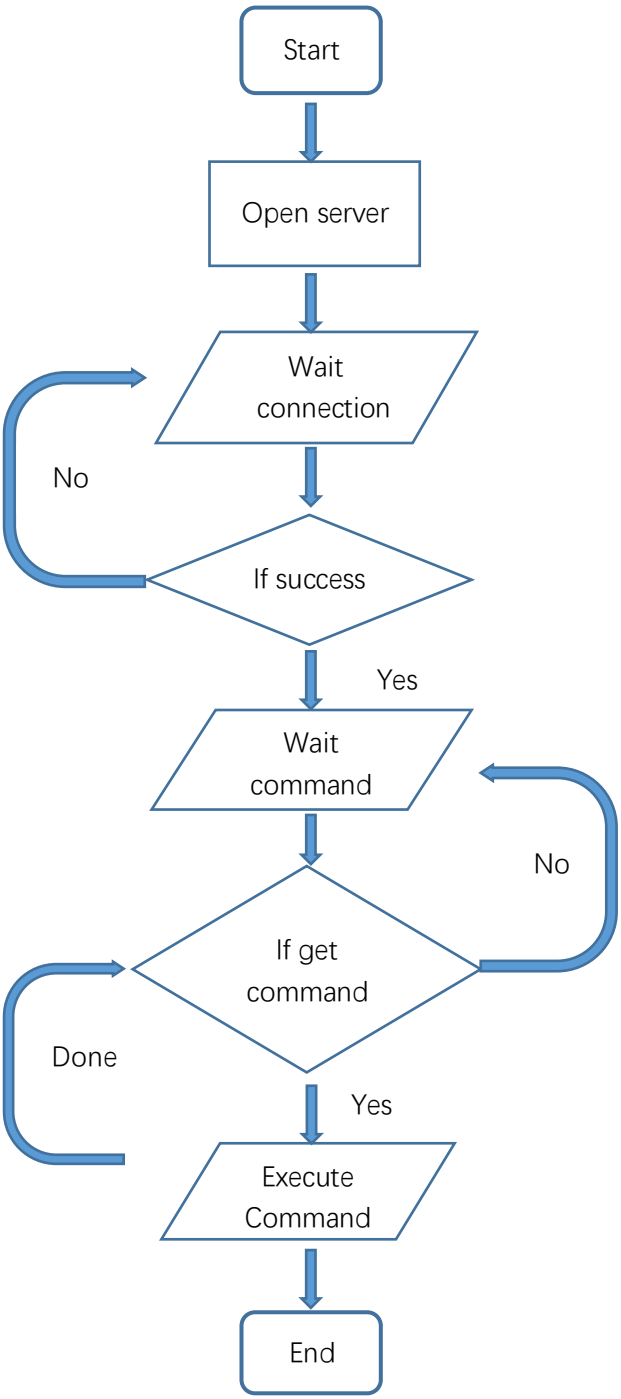
Result:

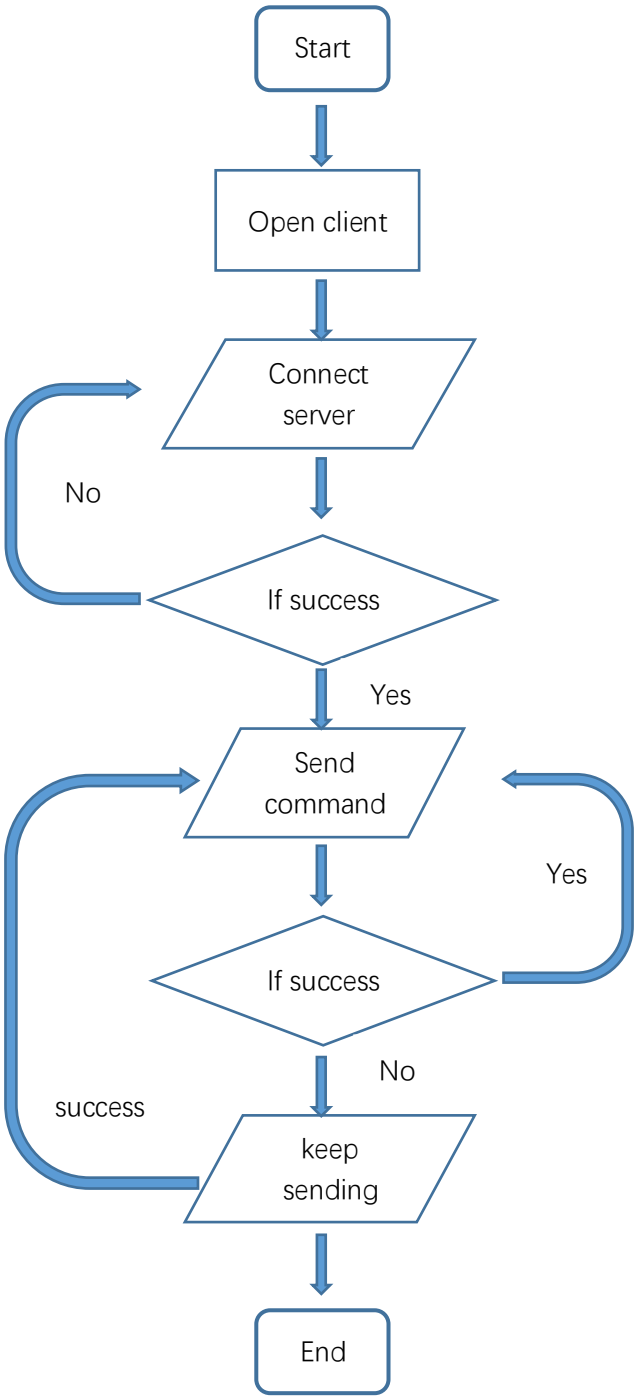
After executing the commands, you can see image.jpg in the corresponding directory.



Chapter 4 Hexapod Robot

The robot features with live video, ultrasonic ranging and other functions. Based on Python3 and PyQt5, it has also been built with server and client, which communicate with each other through TCP/IP protocol and can be controlled remotely with the same LAN.





Server

Run on RPi, the server can send image data obtained by camera and ultrasonic data to client as well as receive commands from client.

Part of the server code is as follows:

```
    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 address
        HOST=self.get_interface_ip()
        #Port 8002 for video transmission
        self.server_socket = socket.socket()
        self.server_socket.setsockopt(socket.SOL_SOCKET,socket.SO_REUSEPORT,1)
        self.server_socket.bind((HOST, 8002))
        self.server_socket.listen(1)
        #Port 5002 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, 5002))
        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")
    def reset_server(self):
        self.turn_off_server()
        self.turn_on_server()
        self.video=threading.Thread(target=self.transmission_video)
        self.instruction=threading.Thread(target=self.receive_instruction)
        self.video.start()
        self.instruction.start()
```

Reference

For more code, please check “Server.py” in the Server directory.

get_interface_ip()
To obtain the IP address of RPi’s WLAN0
turn_on_server()
To turn ON TCP to wait for the connection of Client
turn_off_server()
To turn OFF TCP
send_data()
To send commands to client
reset_server ()
To restart TCP
transmission_video()
To send video data to client
receive_instruction()
To receive commands from client

Open Server

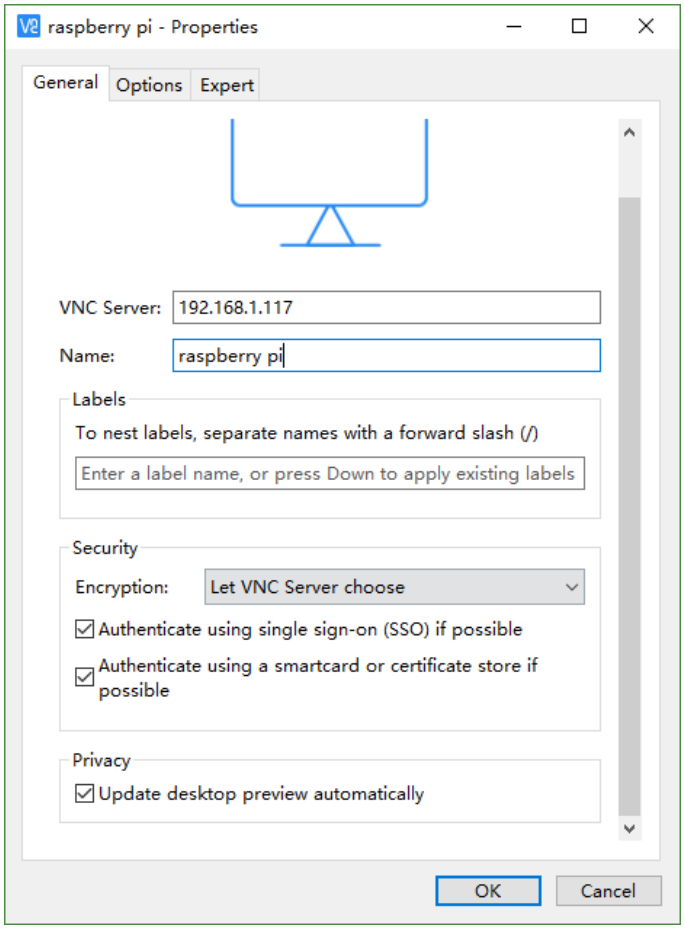
Step 1 Login Raspberry Pi via VNC viewer

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

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

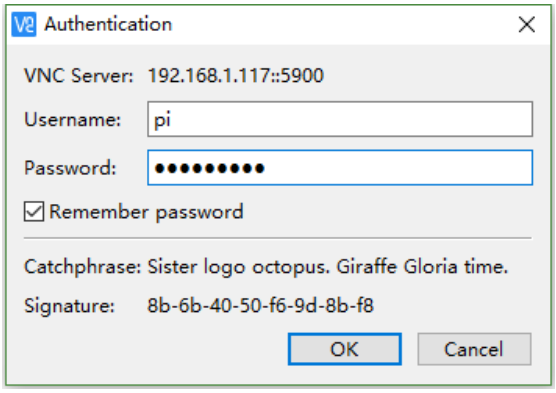
https://www.realvnc.com/en/connect/download/viewer/

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



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

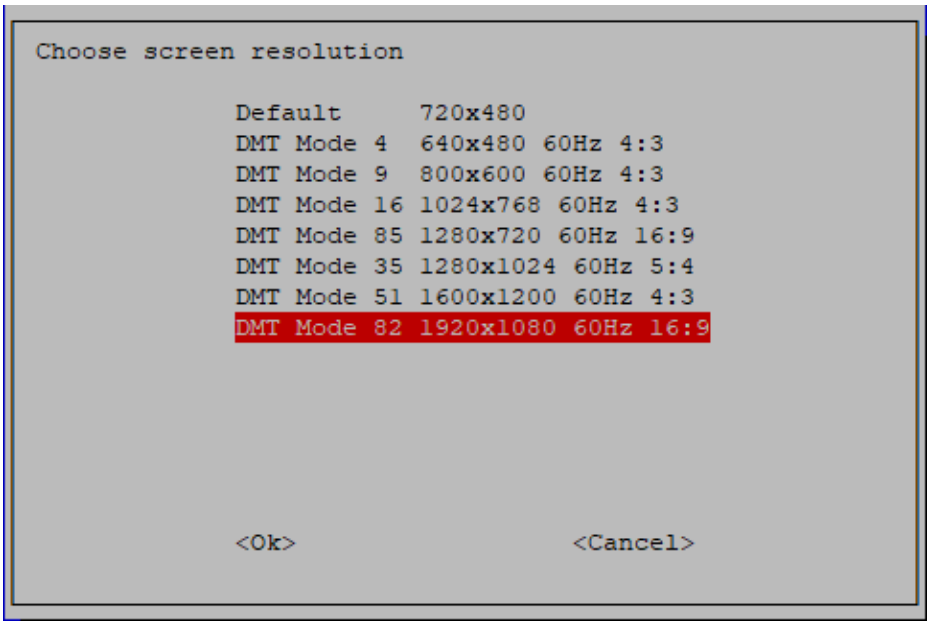
Then on the VNC Viewer panel, double-click new connection you just created, and the following dialog box pops up. Enter username: **pi** and Password: **raspberry**. And click OK.



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

```
sudo raspi-config
```

Select 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.

Step 2 Run commands

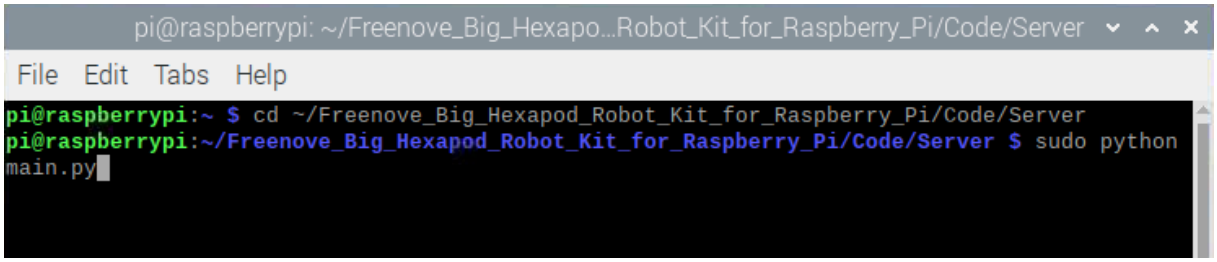
Enter following command in the terminal.

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

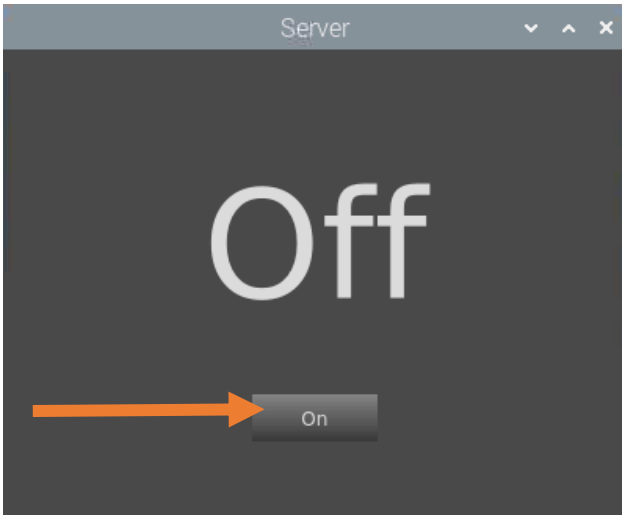
```
cd ~/Freenove_Big_Hexapod_Robot_Kit_for_Raspberry_Pi/Code/Server
```

2. Run main.py:

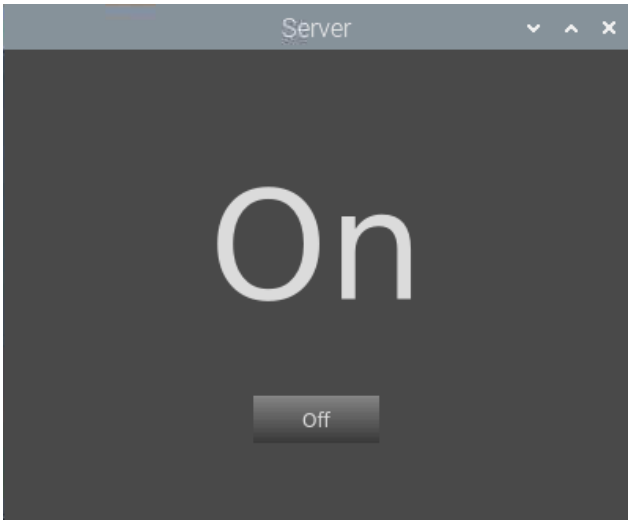
```
sudo python main.py
```



The interface is as below:



Click “On” to open the server



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

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

```
cd ~/Freenove_Big_Hexapod_Robot_Kit_for_Raspberry_Pi/Code/Server
```

2. Run main.py:

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

or Run main,py with following command:

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

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

Sever Auto Start

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

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

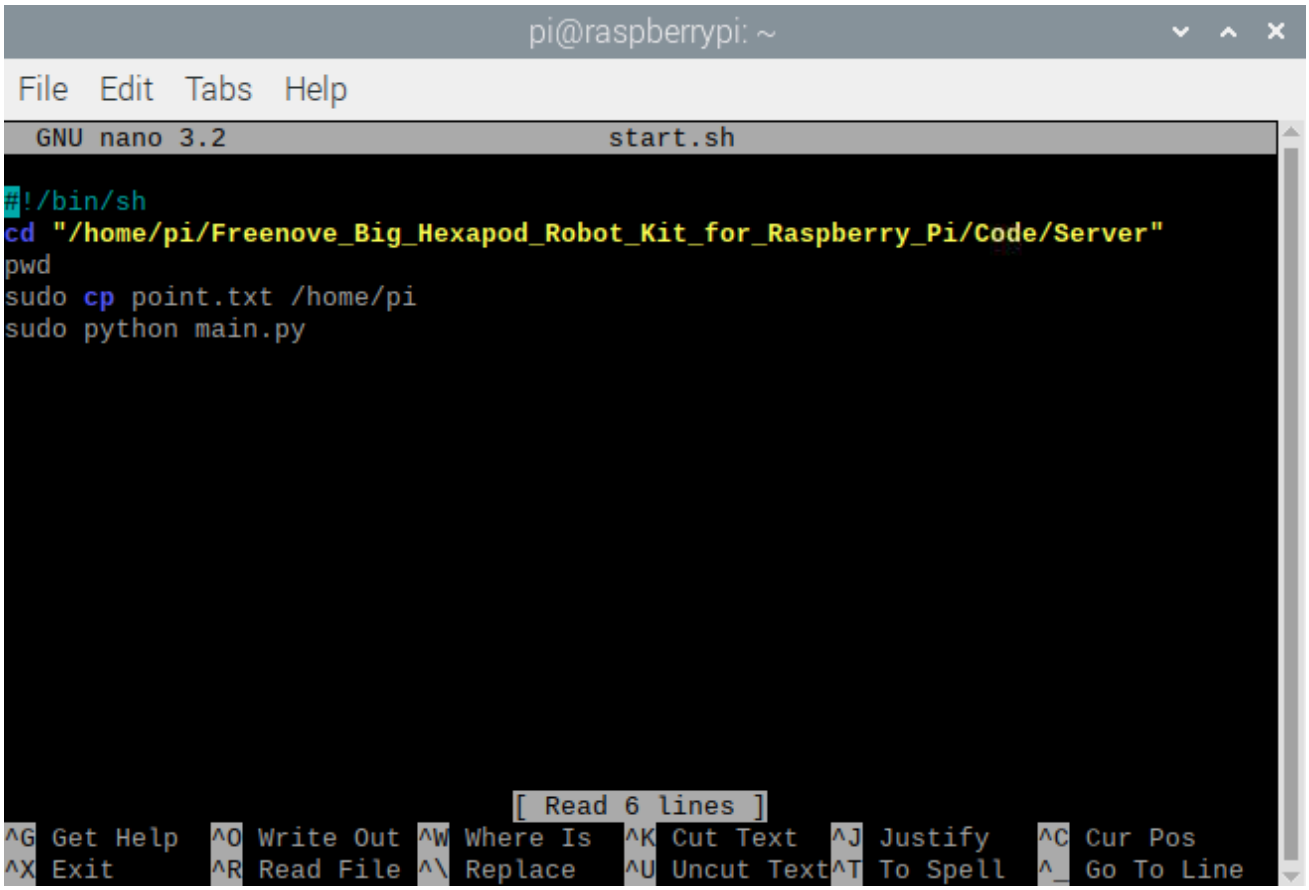
2 Open "start.sh".

```
sudo nano start.sh
```

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

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

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



4 Modify permissions.

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

5 Enter the following command to create a directory.

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

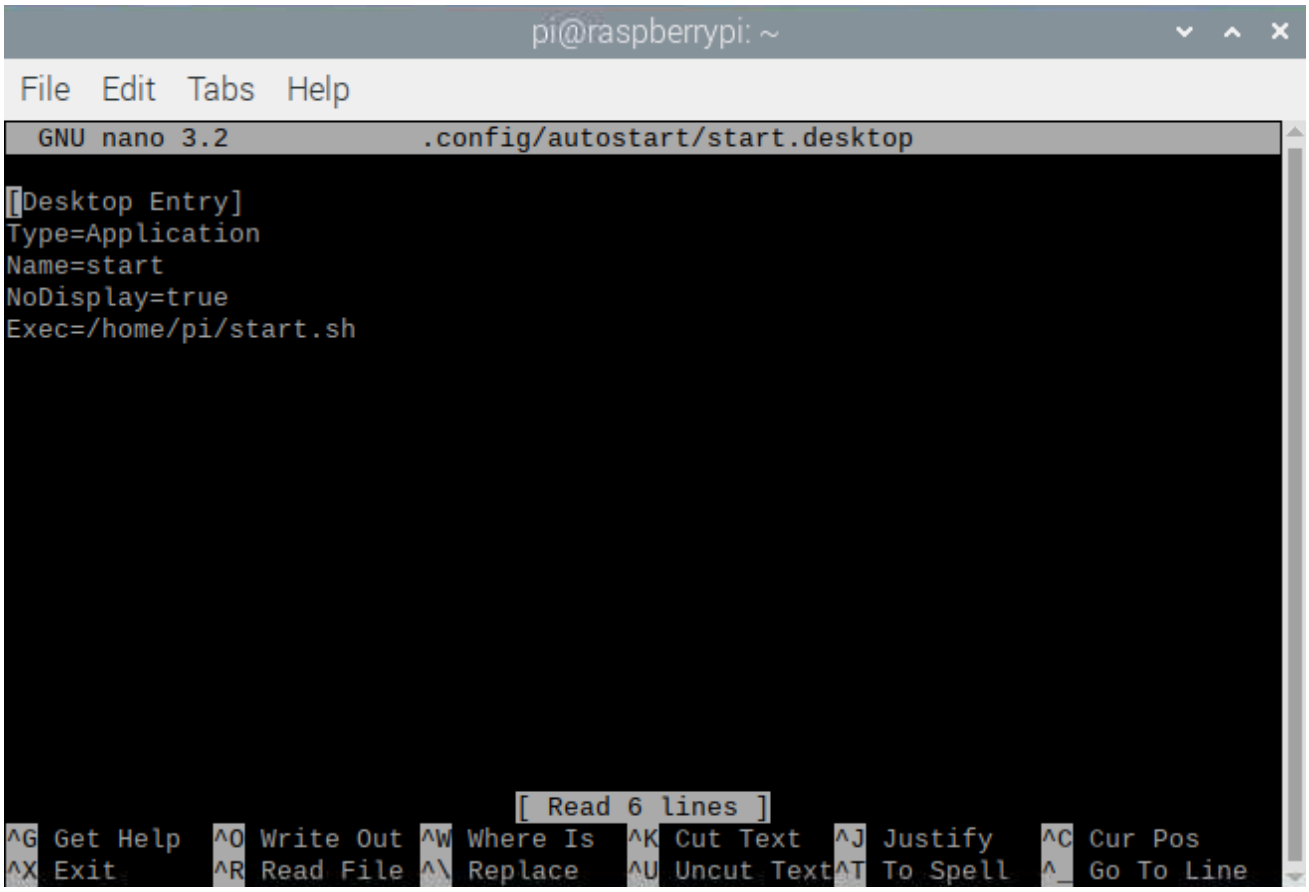
6 create and open “start.desktop” file

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

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

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

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



8 Modify permissions.

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

9 Finally enter the following content to reboot Raspberry Pi.

```
sudo reboot
```

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

Client

Client can receive video data and commands from the server as well as send commands to server. It can also be run in different systems including Windows and macOS as long as you install the related software and libraries.

Part of the client code is as follows:

```
    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):
        try:
            self.client_socket.connect((ip, 8002))
            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.fece_id == False and self.fece_recognition_flag:
                            self.face.face_detect(self.image)
                        self.video_flag=False
            except BaseException as e:
                print (e)
                break
    def send_data(self,data):
        if self.tcp_flag:
```

```
37              try:
38                  self.client_socket1.send(data.encode('utf-8'))
39              except Exception as e:
40                  print(e)
41      def receive_data(self):
42          data=""
43          data=self.client_socket1.recv(1024).decode('utf-8')
44          return data
```

Reference

For more code, please check “Client.py” in the Client director.

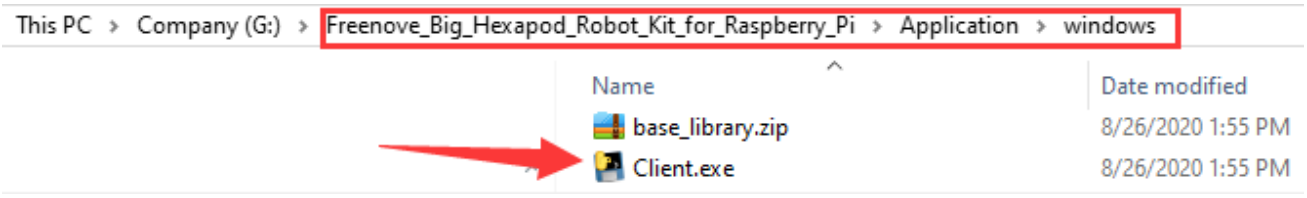
turn_on_client ()
To connect to the server
turn_off_client ()
To disconnect server。
receiving_video()
To receive video data from server
is_valid_image_4_bytes ()
To check the integrity of each frame of video data
send_data ()
To send commands to server
receive_data ()
To receive commands from server

Run Client on Windows system

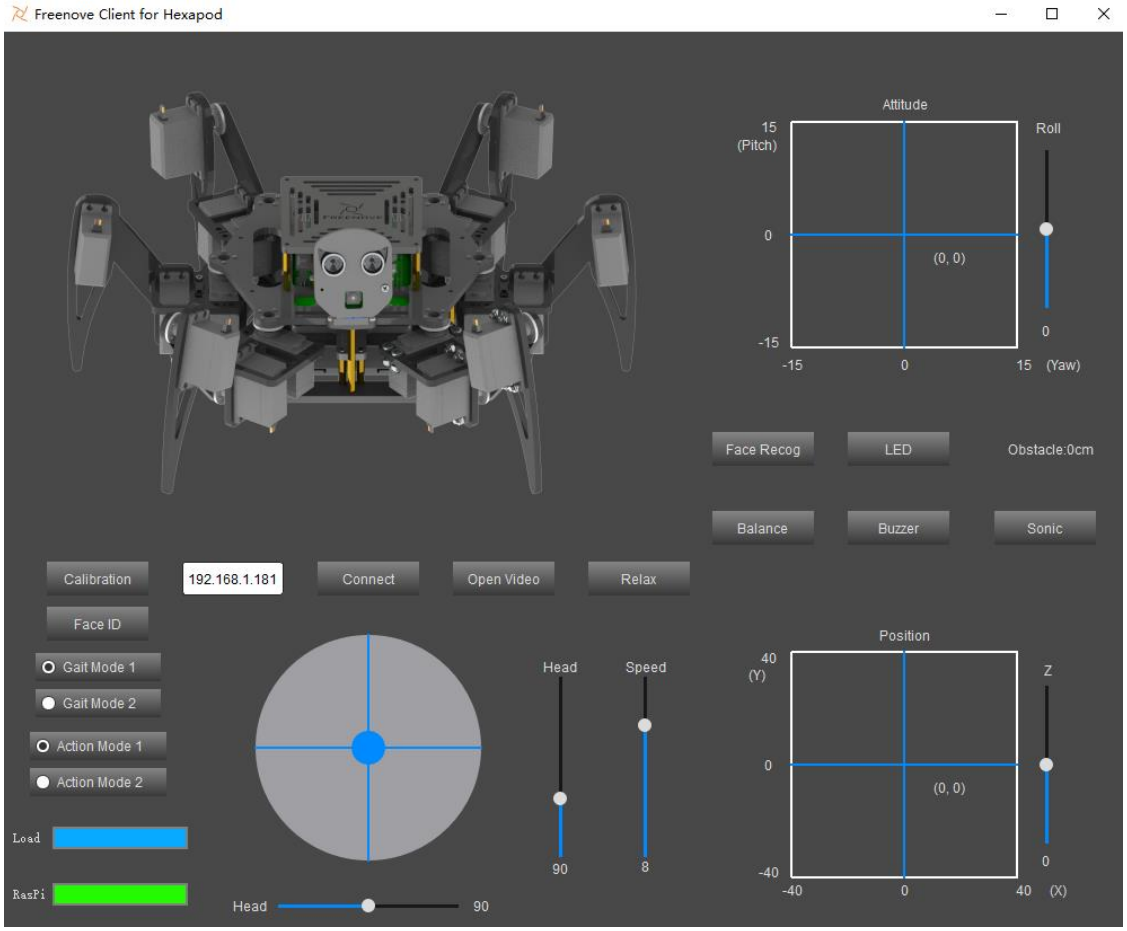
There are two ways to run Client on Windows.

1. Running executable file directly

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



The client interface is shown as below:



When the Client is opened successfully, you need to turn on Raspberry Pi and open the server. Enter Raspberry Pi’s IP address in the white IP editor and click “Connect” to connect the robot with RPi. After connecting successfully, you need to calibrate the robot's six legs before control it to move.

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

2.Install python3 and some related python libraries and execute the program through python3 editor.

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

Running client on Windows system requires the installation of some software and libraries, which take some time. At this point, it does not require to run server and use the Raspberry Pi, so you can turn OFF RPi temporarily. After installing successfully, turn ON the RPi and run the Server.

Install python3

Download the installation file via below:

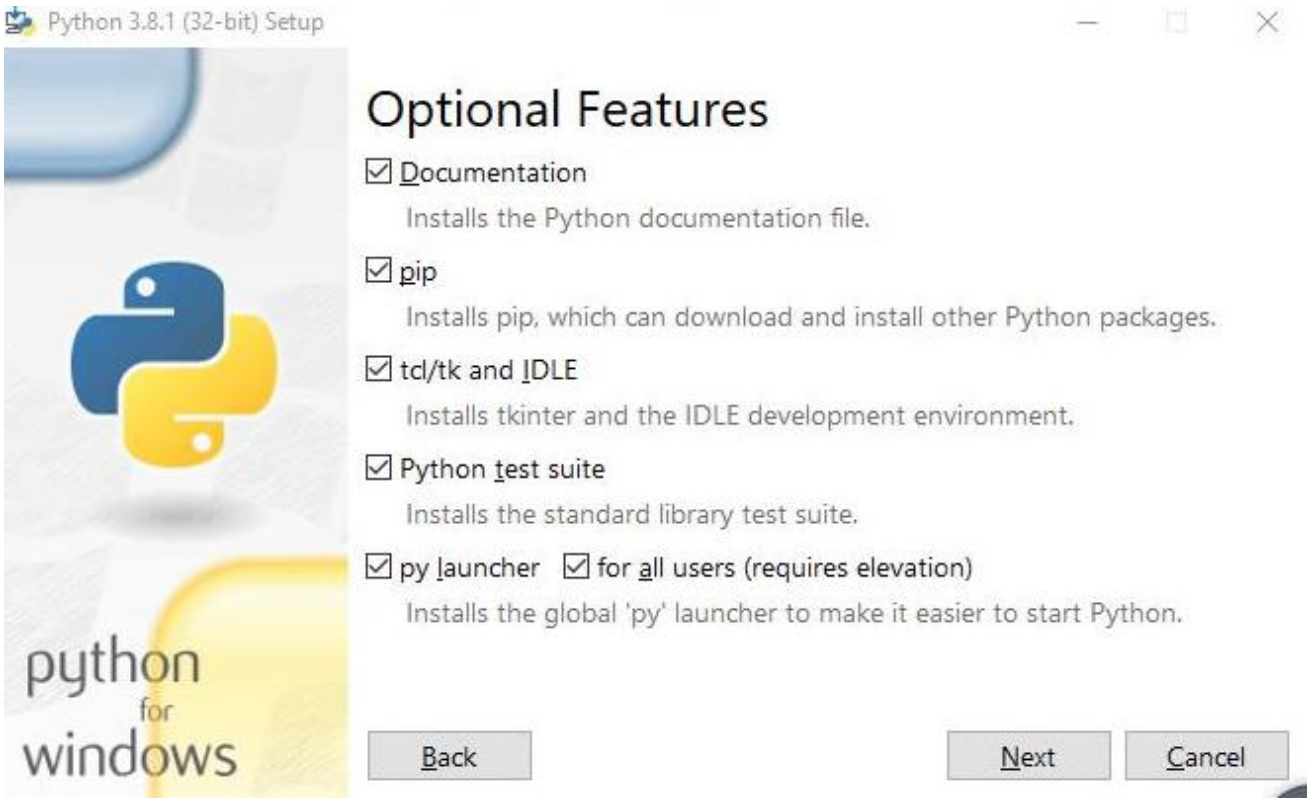
https://www.python.org/downloads/windows/

Click Latest Python Release - Python 3.8.1

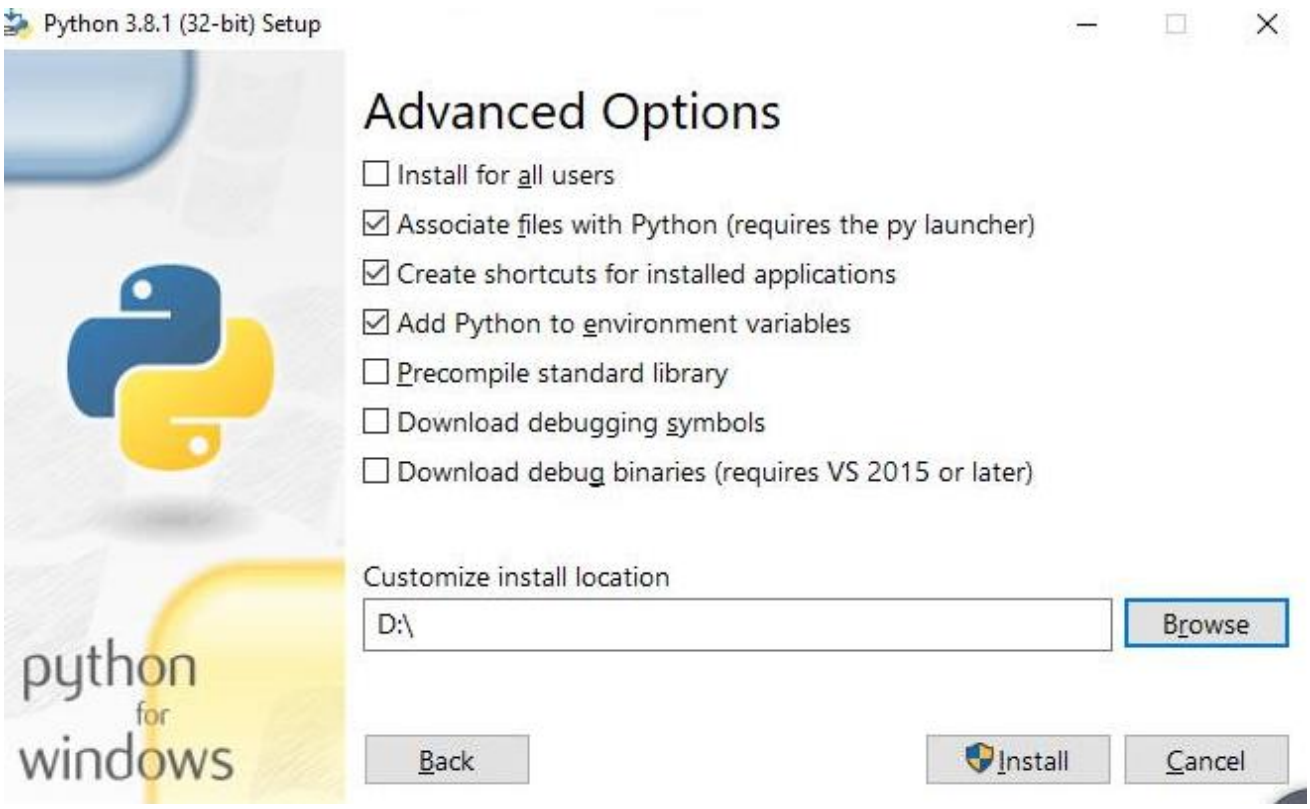
Version	Operating System	Description
Gzipped source tarball	Source release	
XZ compressed source tarball	Source release	
macOS 64-bit installer	Mac OS X	for OS X 10.9 and later
Windows help file	Windows	
Windows x86-64 embeddable zip file	Windows	for AMD64/EM64T/x64
Windows x86-64 executable installer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

Select “Windows x86 executable installer”. Execute the installer when completes downloading.。

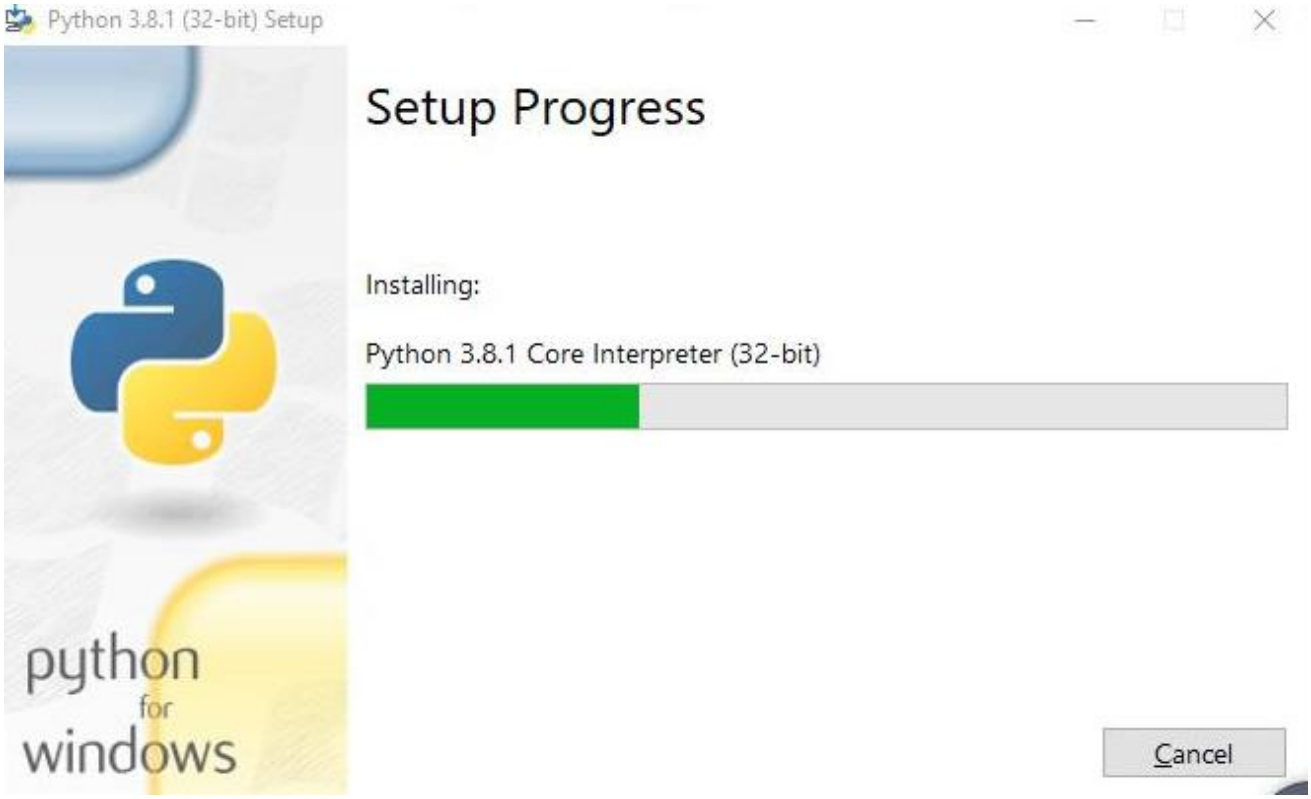
Check "Add Python 3.8 to PATH" and install according to your needs.



Check all the options and click "Next".



My installation path here is Disk D (You can choose whichever you prefer). Click Install.



Wait for the installation,

Installation finishes.

Install libraries including PyQt5 library, opencv library, numpy library, etc.

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

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

Then unzip it and delete "-master" to rename it to "Freenove_Big_Hexapod_Robot_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_Big_Hexapod_Robot_Kit_for_Raspberry_Pi\Code
```

3.Run: setup_windows.py

```
Python setup_windows.py
```

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

D:\>cd D:\Freenove_Big_Hexapod_Robot_Kit_for_Raspberry_Pi\Code

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

Or you can enter "Freenove_Big_Hexapod_Robot_Kit_for_Raspberry_Pi\Code\" directory and double click "setup_windows.py" to execute the installer.

Or open "setup_windows.py" with python3 and execute it.

Caution: If the default python of your Windows system is not python3 but python2, change the command "setup_windows.py" as follows and replace all "python" in setup_windows.py to "python3".

```
Python3 setup_windows.py
```

The installation takes some time, please wait with patience. When all the libraries are installed successfully, you can see the prompt "All libraries installed successfully" on the screen.

```
Package                Version
---------------------- ----------
click                  7.1.2
numpy                  1.18.5
opencv-contrib-python  4.3.0.36
opencv-python          4.3.0.36
Pillow                 7.1.2
pip                    20.2.1
PyInstaller            3.6
PyQt5                  5.15.0
PyQt5-sip              12.8.0
pyqt5-tools            5.15.0.1.7
python-dotenv          0.14.0
setuptools             41.2.0


All libraries installed successfully
```

If there is any library fails to install, it will print “Some libraries have not been installed yet. Please run 'python setup_windows.py' again” on the screen, then you need to execute the command python setup_windows.py again. Most installation fails because of poor network, so you can check your network before installing the libraries.

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_Big_Hexapod_Robot_Kit_for_Raspberry_Pi\Code\Client
```

3.Run Main.py:

```
Python Main.py
```

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

D:\>cd D:\Freenove_Big_Hexapod_Robot_Kit_for_Raspberry_Pi\Code\Client

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

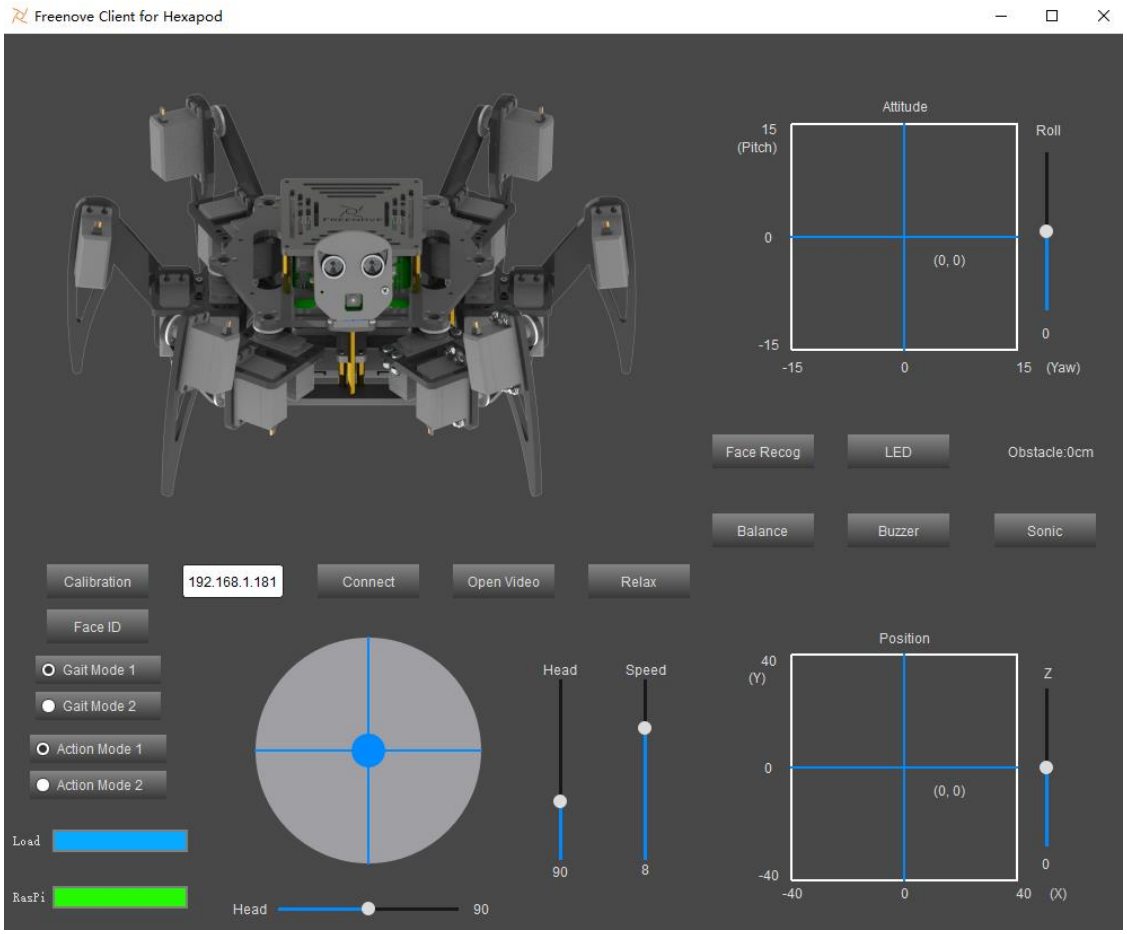
Or you can enter “Freenove_Big_Hexapod_Robot_Kit_for_Raspberry_Pi\Code\Client” directory and double-click “Main.py”.

Or use python3 to open and execute “Main.py”.

Note: If the default python of your Windows system is not Python3 but python2, change the command Main.py as follows:

```
Python3 Main.py
```

The client interface is shown as below:



When the Client is opened successfully, you need to turn on Raspberry Pi and open the server. Enter Raspberry Pi's IP address in the white IP editor and click "Connect" to connect the robot with RPi. After connecting successfully, you need to calibrate the robot's six legs before control it to move.

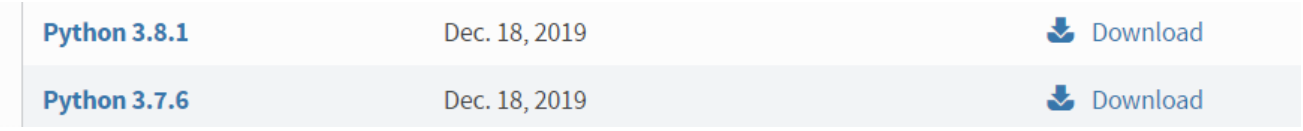
Note: when Raspberry Pi is shut down, server will be closed. You need open server again the next time.

Run Client on macOS system

Running client on macOS system requires the installation of some software and libraries, which take some time. At this point, it does not require to run server and use the Raspberry Pi, so you can turn OFF RPi temporarily. After installing successfully, turn ON the RPi and run the Server. The system comes with python2 instead of python3, so you need to install python3 first, as all the program of this robot are run with python3.

Install python3

Download the installation package via: https://www.python.org/downloads/

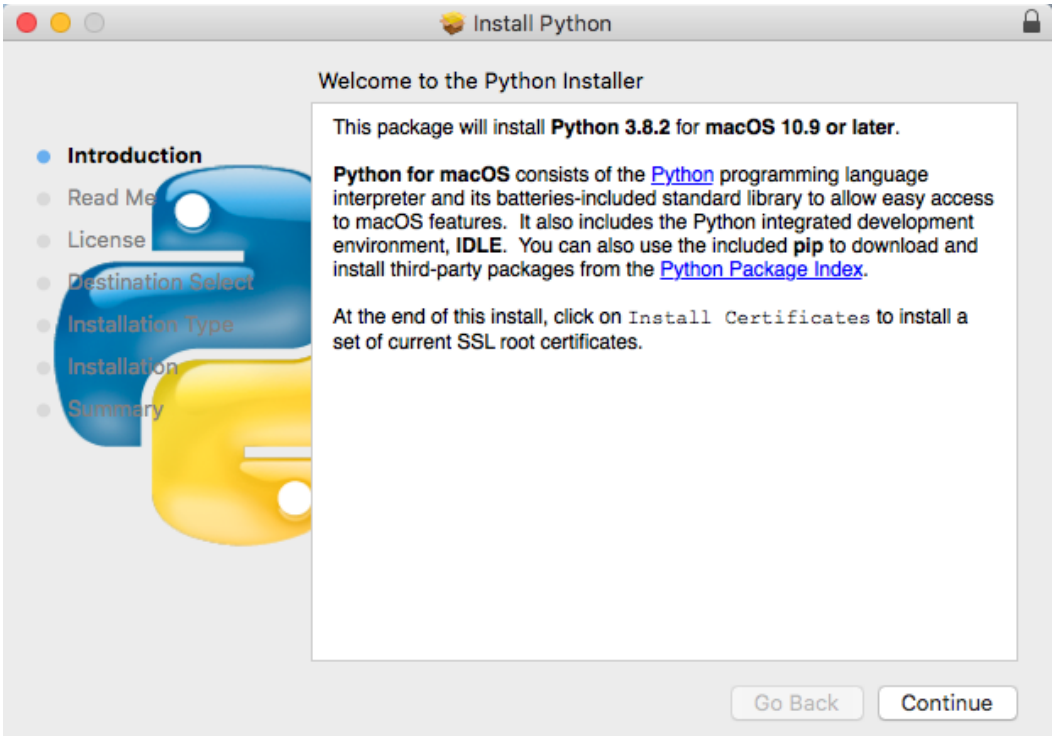


If your macOS is 11. Like 11.0, please install python 3.9.

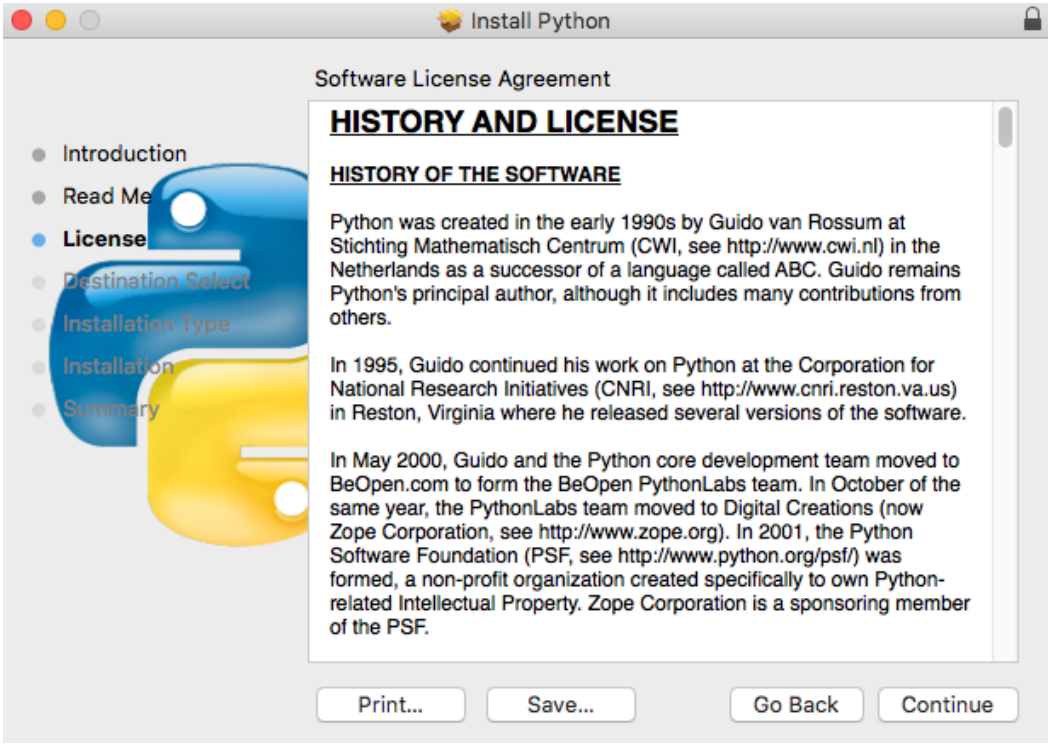
If your macOS is NOT 11, like 10.15, please install python 3.8. If you have installed python 3.9. You need uninstall it first.

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

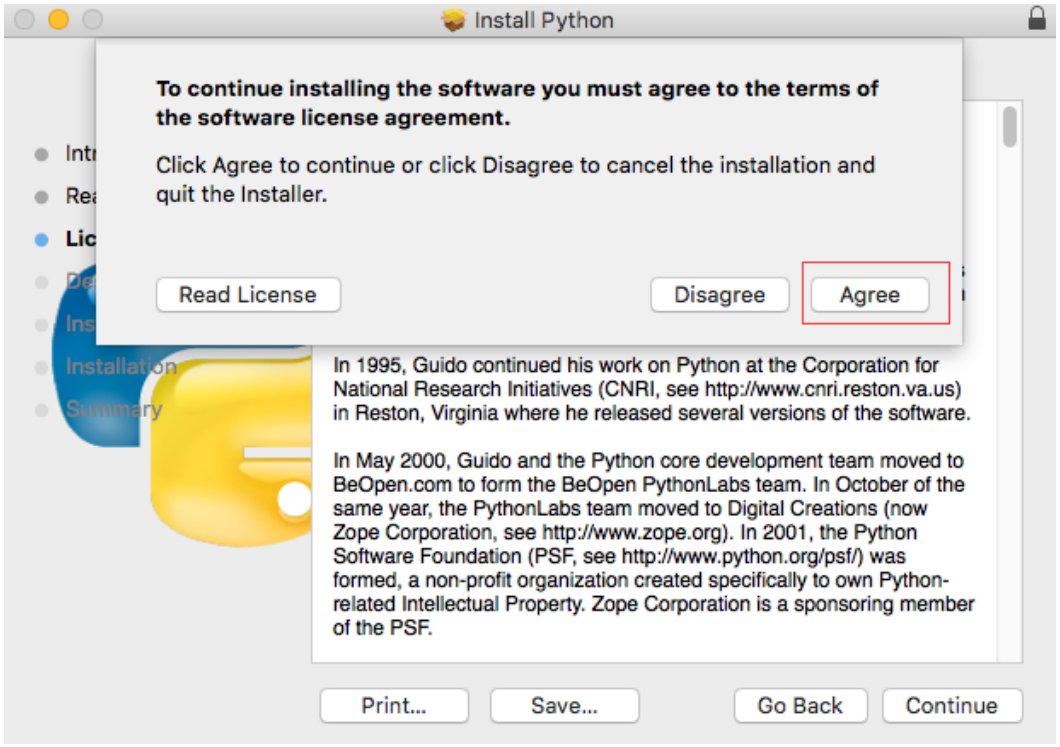
Slide to the bottom of the page and click macOS 64-bit installer.



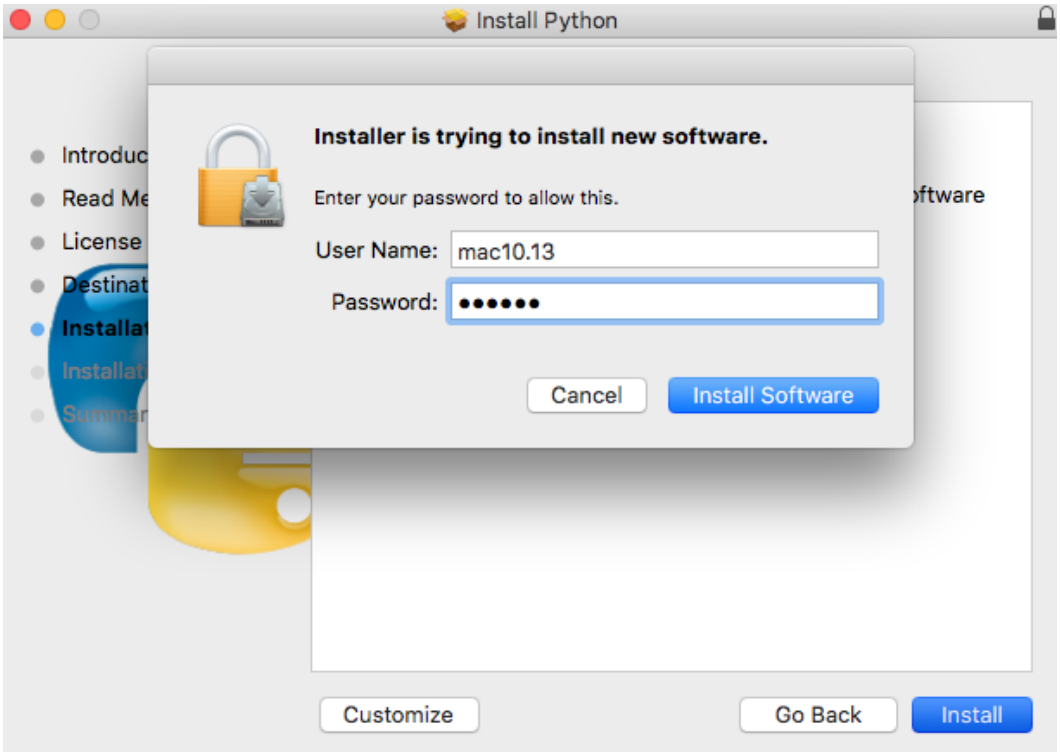
Click Continue



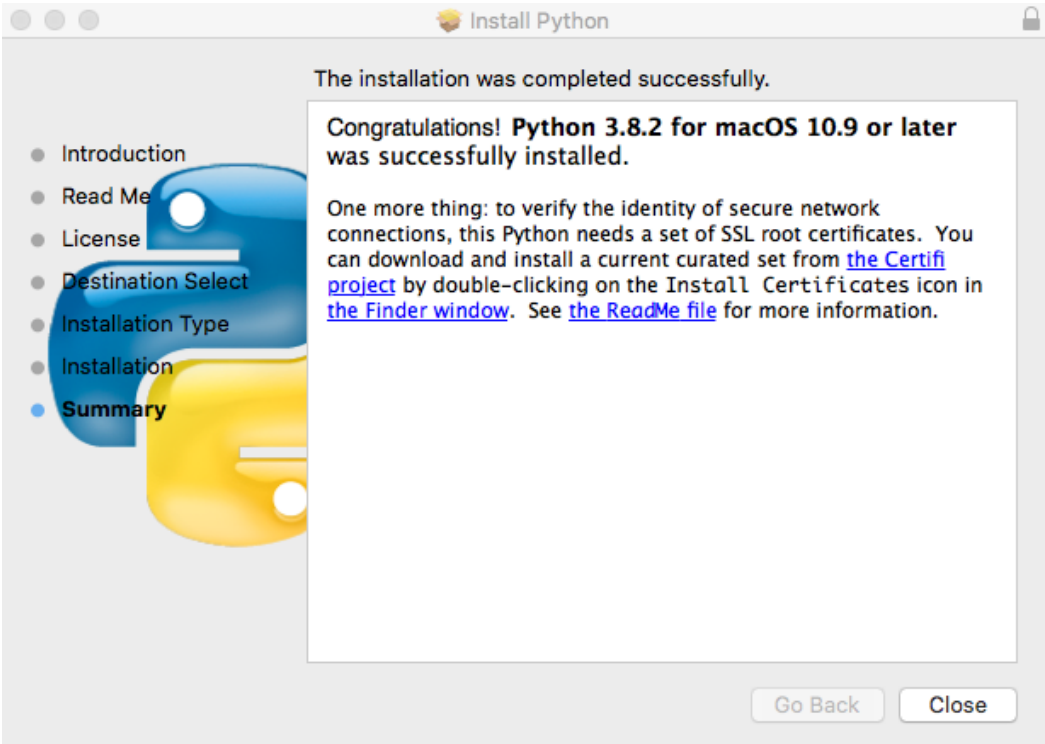
Click Continue



Click Agree



Click "Install", enter the computer password and click "Install Software".



Install successfully

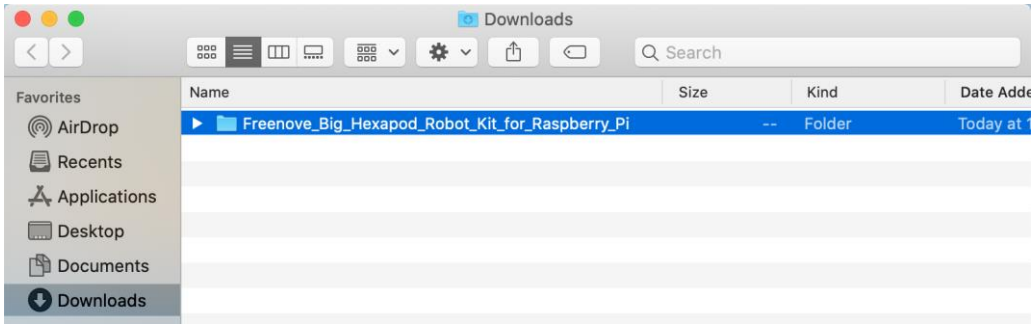
You can find it in your Applications.

Install PyQt5 library, opencv library, numpy library, etc.

If you have not yet downloaded the robot code, you can download it through the following link:

https://github.com/Freenove_Big_Hexapod_Robot_Kit_for_Raspberry_Pi/archive/master.zip

After downloading, you can find it in “Downloads”.



Open “Terminal”

Enter the following commands in “Terminal”

1.Enter “Downloads” (where you save the robot code. If your path is different, please change it.)

```
cd Downloads
```

2.Enter directory where setup_macos.py is located:

```
cd Freenove_Big_Hexapod_Robot_Kit_for_Raspberry_Pi/Code/
```

3.Run setup_macos.py:

```
python3 setup_macos.py
```

```
[freenove@PandeMacBook-Air ~ % cd Downloads                                                    ]
[freenove@PandeMacBook-Air Downloads % cd Freenove_Big_Hexapod_Robot_Kit_for_Raspberry_Pi/Code/  ]
 freenove@PandeMacBook-Air Code % python3 setup_macos.py
```

The installation takes some time, please wait with patience.

```
Package                       Version
----------------------------- --------
numpy                         1.19.0
opencv-contrib-python-headless 4.3.0.36
opencv-python-headless        4.3.0.36
Pillow                        7.2.0
pip                           20.2.1
PyQt5                         5.13.2
PyQt5-sip                     12.8.0
setuptools                    47.1.0
wheel                         0.34.2

All libraries installed successfully
freenove@PandeMacBook-Air Code %
```

When all the libraries are installed successfully, you can see the prompt "All libraries installed successfully" on the screen. If there is any library fails to install, it will print "Some libraries have not been installed yet. Please run ' python3 setup_macos.py' again" on the screen, then you need to execute the command python3 setup_macos.py again.

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

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

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

Most installation fails because of poor network, so you can check your network before installing the libraries.

Open client

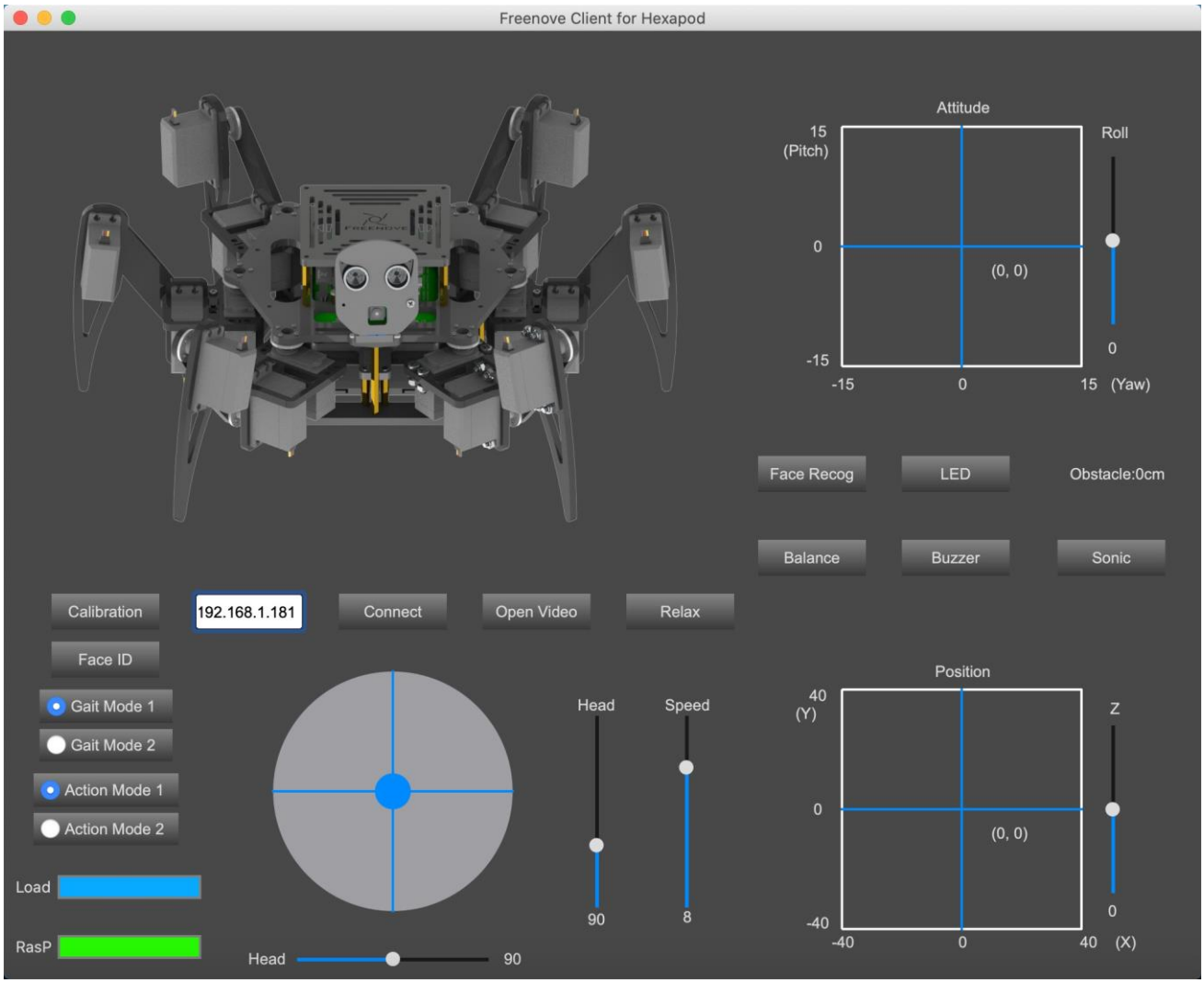
Installing successfully, now you are in the directory where setup_macos.py locates.

1. Enter the following command to enter the client program folder:

```
cd Client/
```

2. Enter the following command to run the program

```
python3 Main.py
```



When the Client is opened successfully, you need to turn on Raspberry Pi and open the server. Enter Raspberry Pi's IP address in the white IP editor and click "Connect" to connect the robot with RPi. After connecting successfully, you need to calibrate the robot's six legs before control it to move.

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

Run client on Raspberry Pi(Linux) system

Install opencv library

Enter the following three commands one by one

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

sudo pip3 install opencv-contrib-python==3.4.6.27

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

Open client

Enter the following commands at the terminal.

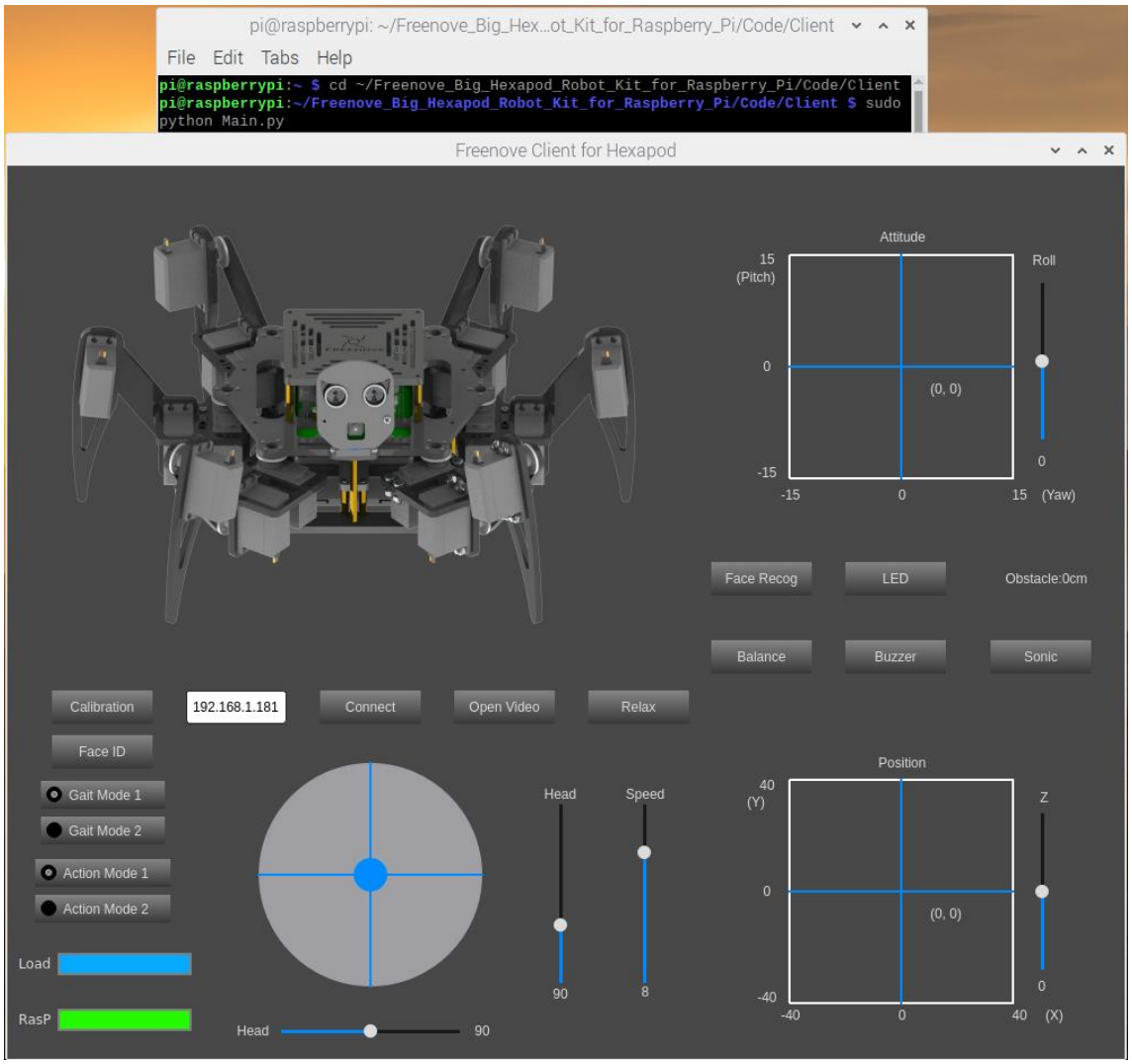
1. Use the cd command to go to the directory where Main.py is located.

```
cd ~/Freenove_Big_Hexapod_Robot_Kit_for_Raspberry_Pi/Code/Client
```

2. Run Main.py:

```
sudo python Main.py
```

The interface is shown below:



When the Client is opened successfully, you need to turn on Raspberry Pi and open the server. Enter Raspberry Pi's IP address in the white IP editor and click "Connect" to connect the robot with RPi. After connecting successfully, you need to calibrate the robot's six legs before control it to move.

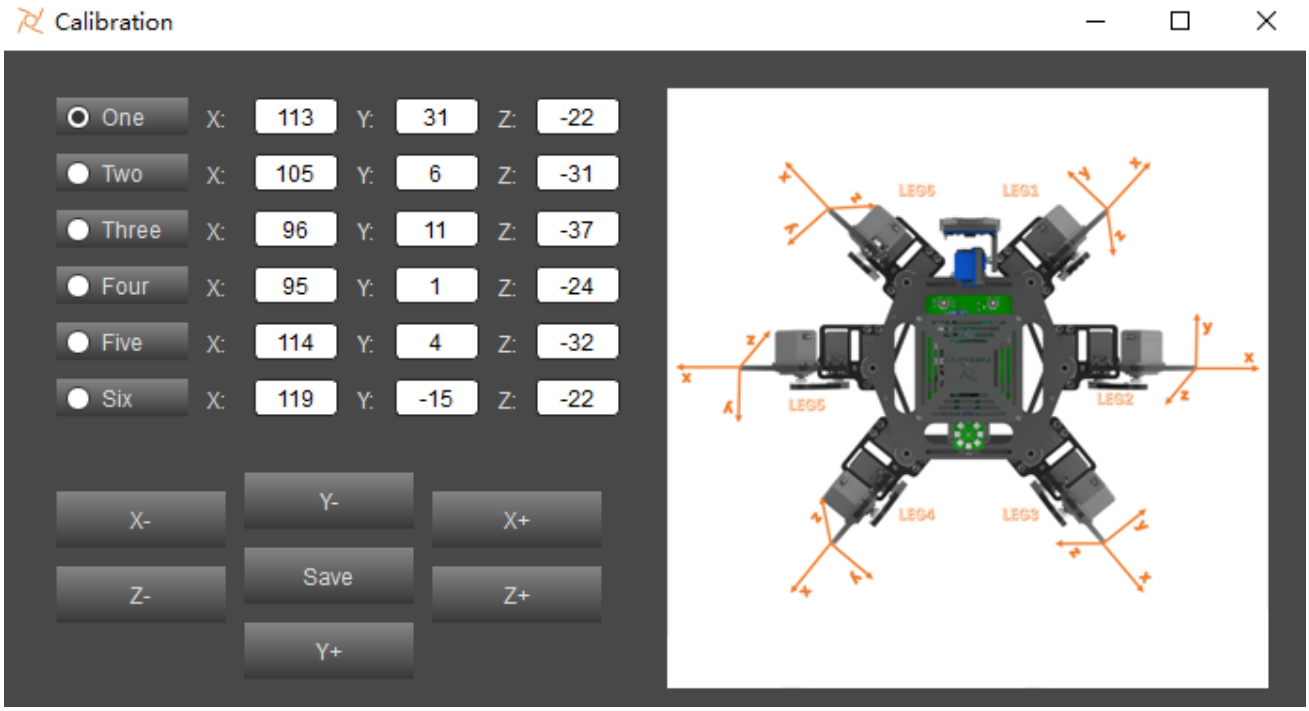
Note: when Raspberry Pi is shut down, server will be closed. You need open server again the next time.

Calibration

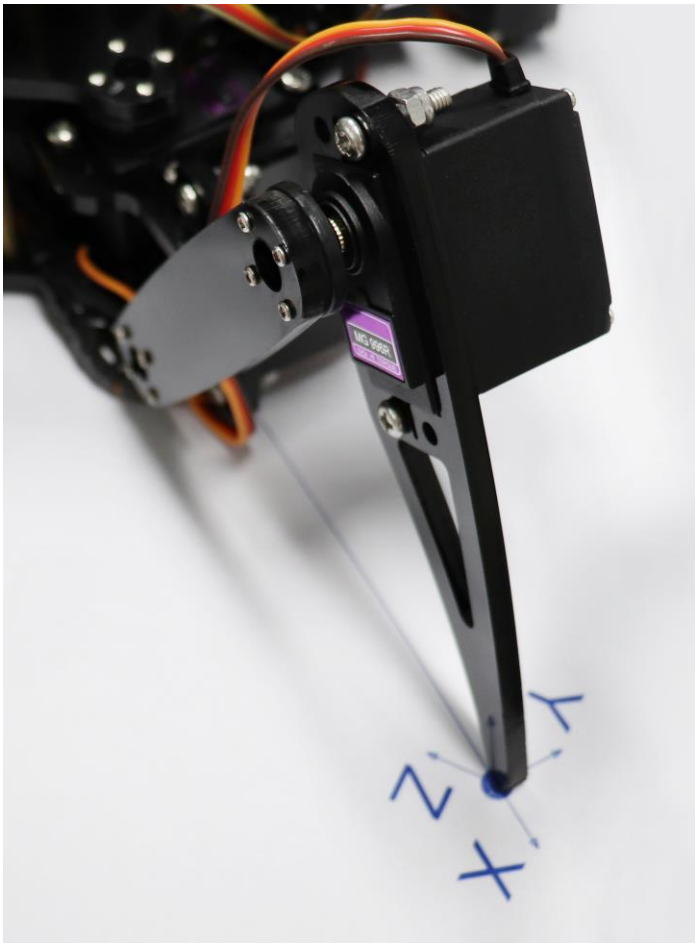
Calibrate the robot's six legs

1. Lay the calibration paper on a horizontal table.
2. Place the robot to the corresponding position on the calibration paper as shown below:

3. Connect client with server successfully, click the "Calibration" button on the client, then a calibration window pops up.



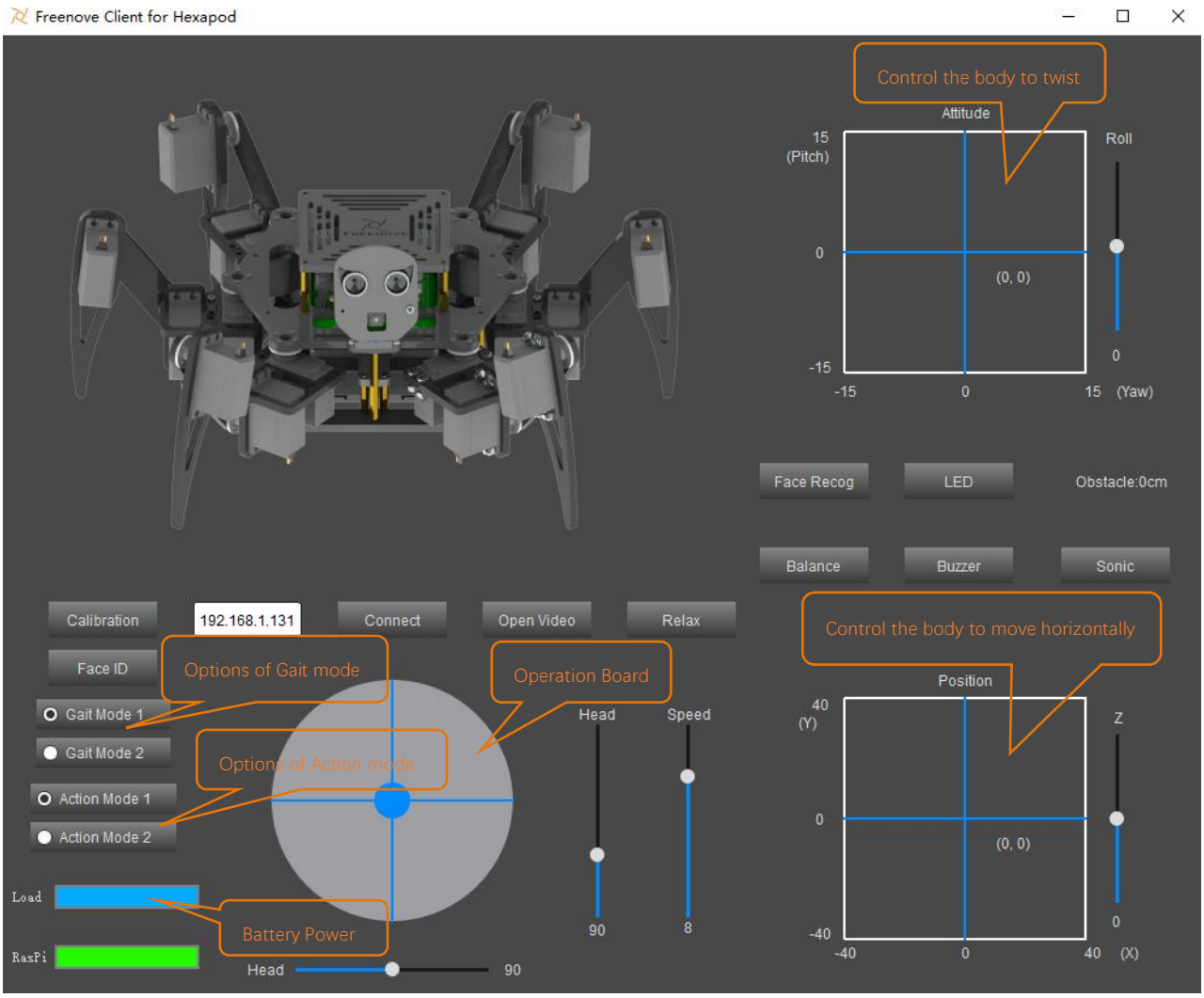
4. Calibrate the six legs to make the six foot points fall to the corresponding positions. (Due to the error of the installation and error of the servos themselves, the six foot points may fail to fall to the specified position perfectly, which is reasonable but we should try to make them as accurate as possible.)



5. When all the six foot points fall to the specified positions, click “Save” and you can control the robot to move.

Note: Poor calibration will affect the robot’s movement. You can recalibrate it when needed.

Control


Freenove Client for Hexapod
Control the body to twist
Attitude
15
(Pitch)
0
-15
-15
0
15
(Yaw)
Roll
0
(0, 0)
Face Recog
LED
Obstacle:0cm
Balance
Buzzer
Sonic
Calibration
192.168.1.131
Connect
Open Video
Relax
Face ID
Options of Gait mode
Operation Board
Gait Mode 1
Gait Mode 2
Options of Action mode
Action Mode 1
Action Mode 2
Head
Speed
90
8
Load
RasPi
Battery Power
Head
90
Control the body to move horizontally
Position
40
(Y)
0
-40
-40
0
40
(X)
Z
0
(0, 0)
0

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
Connect/ Disconnect	C	Connection ON/OFF
Open Video/ Close Video	V	Video ON/OFF
Relax	R	Relax/Activated
LED	L	Open LED control interface
Balance	B	balance mode ON/OFF
Face Recog	F	face recognition ON/OFF
Sonic	U	Ultrasonic ranging ON/OFF
Calibration	T	Open calibration interface
Face ID	I	Open the face reading interface
Buzzer/Noise	Y	Buzzer ON/OFF

The function of SliderBar is below:

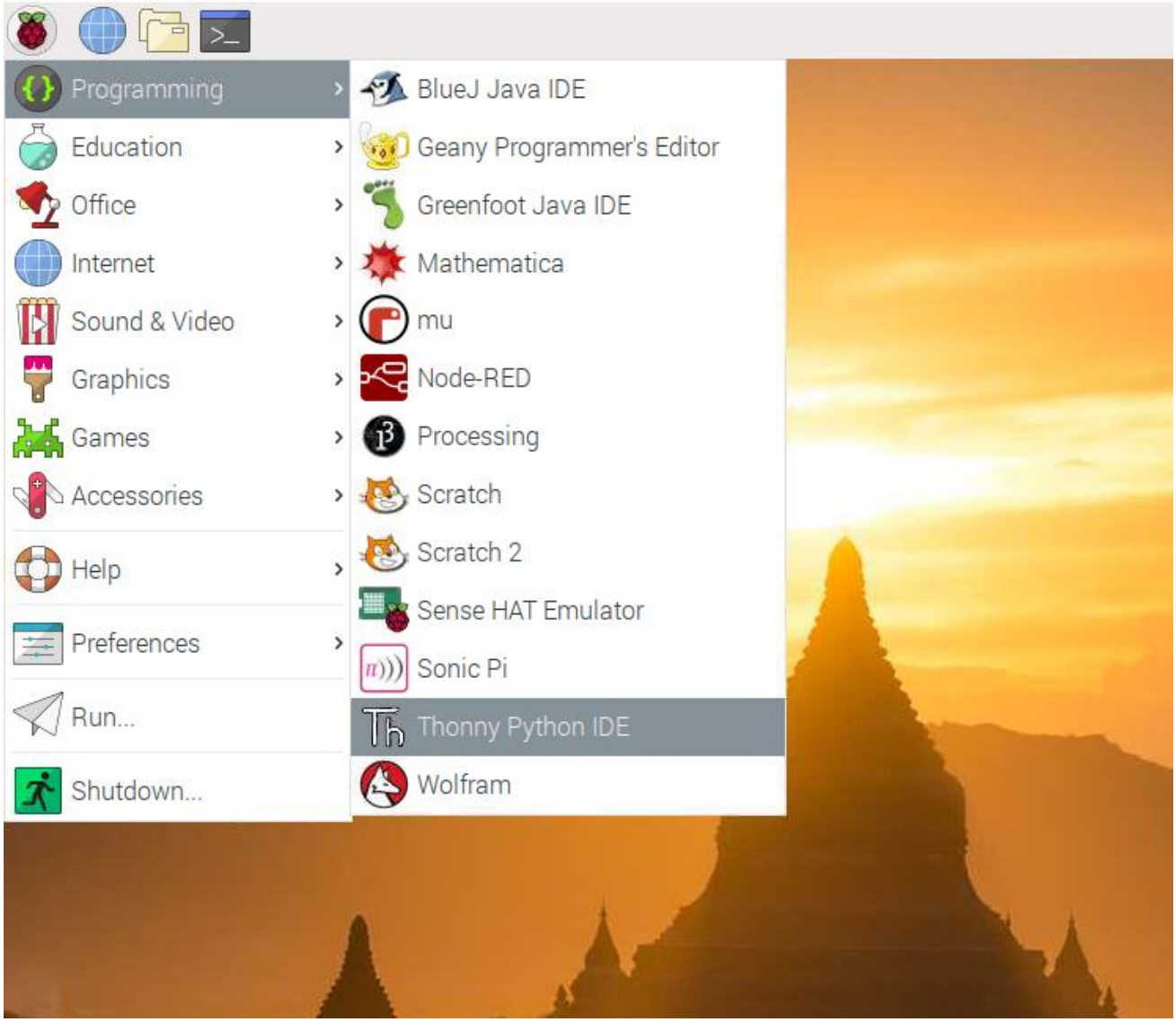
SliderBar	Function
Head	Rotate the robot's head
Speed	Control the robot's speed

Free your innovation

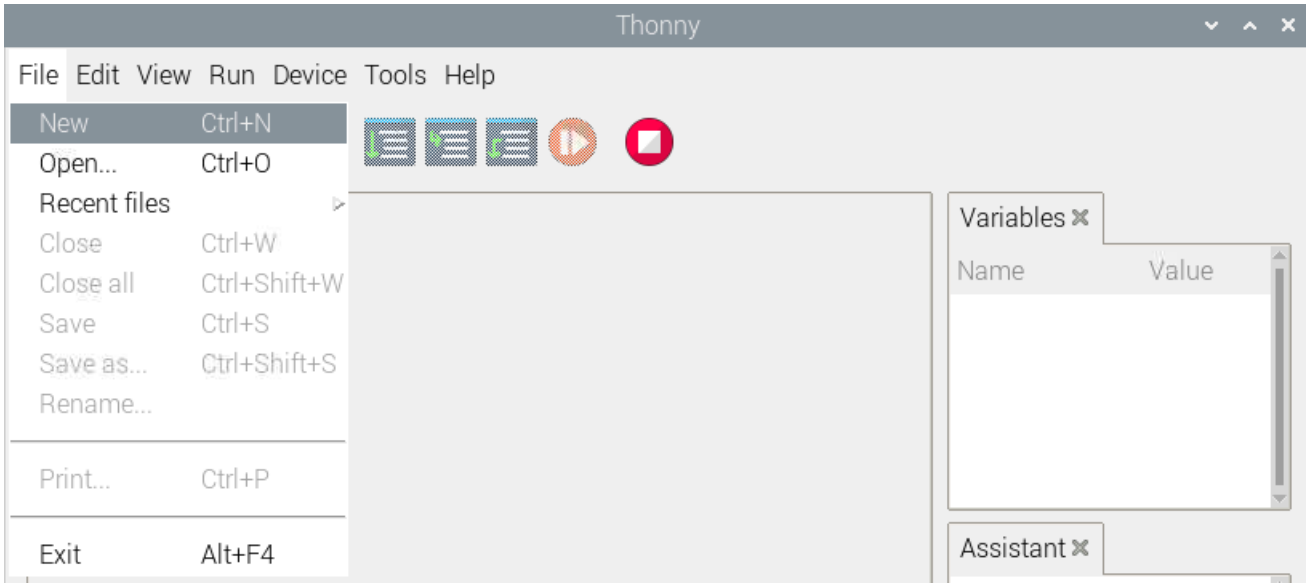
If you have any concerns, please feel free to contact us via support@freenove.com
If you want to write your own program to control the robot dog, you can follow this chapter.
The robot program is based on python3. If your python is python2 by default, please change to python3.
If you have never learned python before, you can learn some basics through the following links:
https://python.swaroopch.com/basics.html

Program

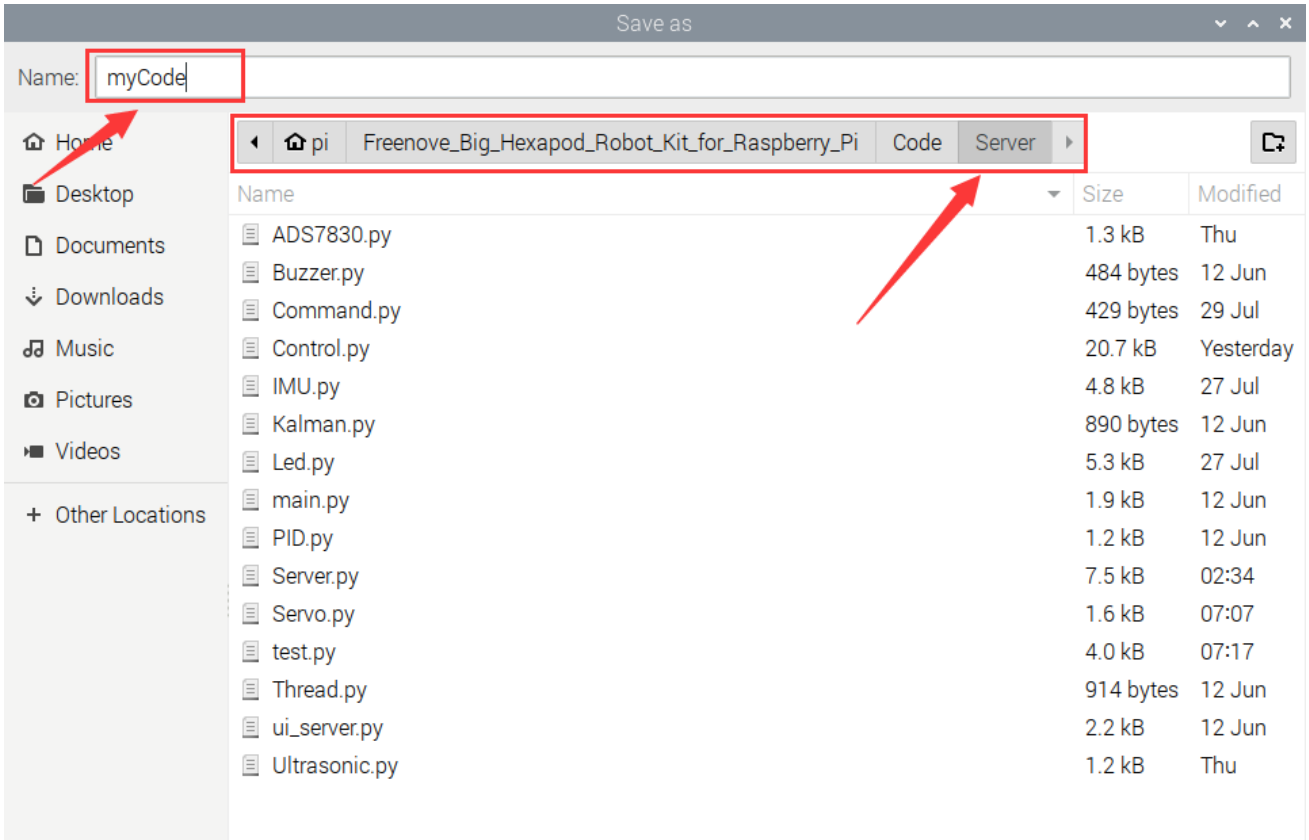
First, open Thonny Python IDE which is easy to use for beginners.



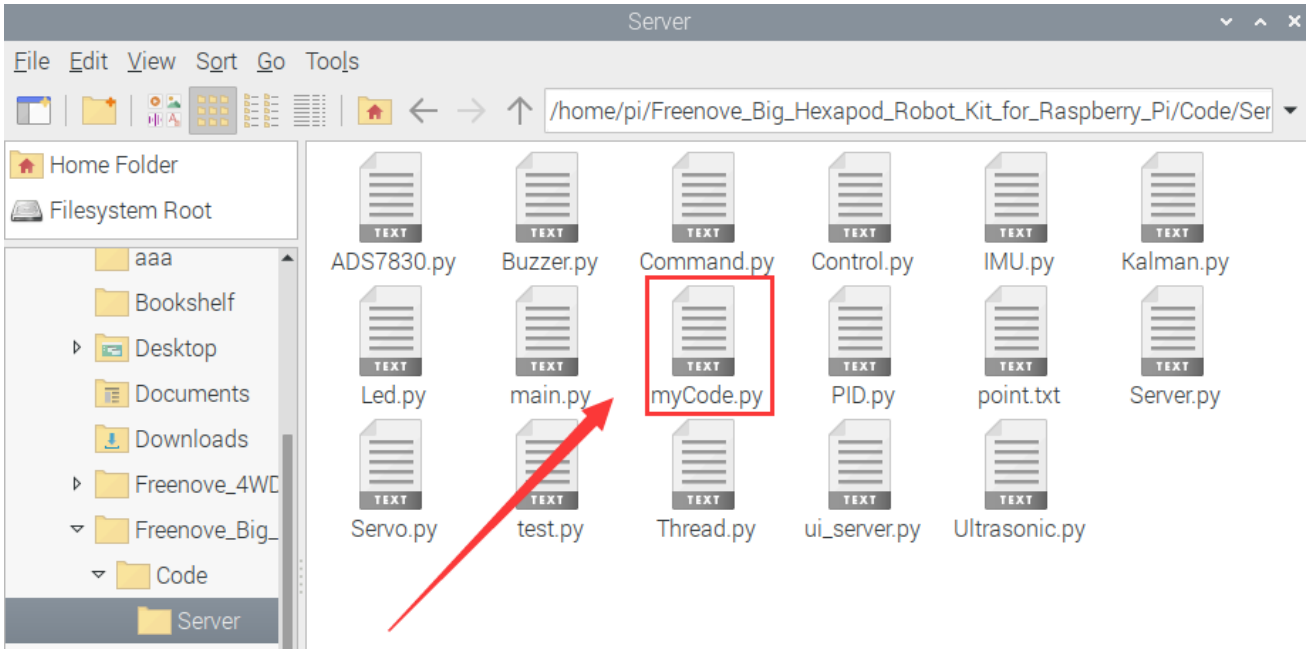
Create a new file.



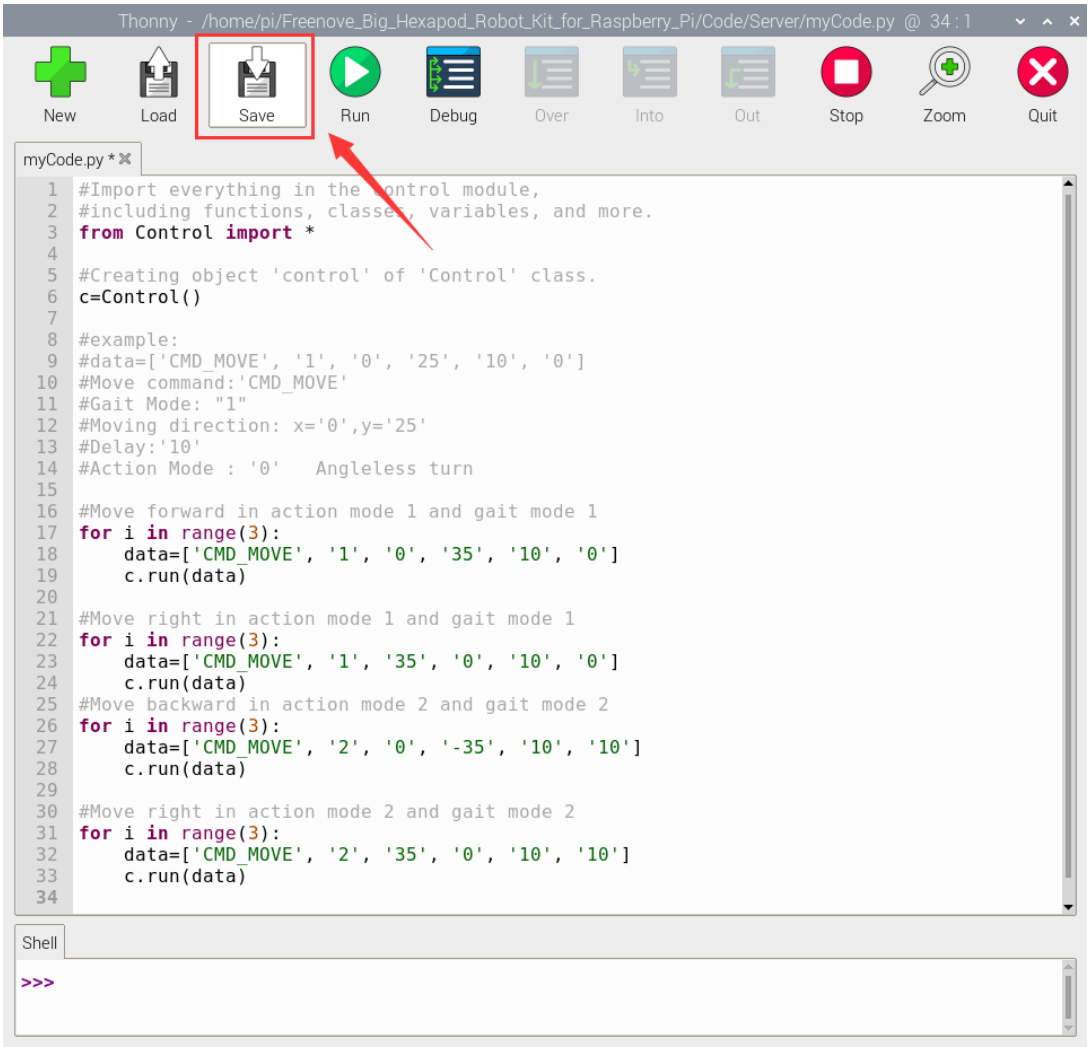
Name it myCode and save it in Server folder of robot code folder that you have downloaded.



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



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

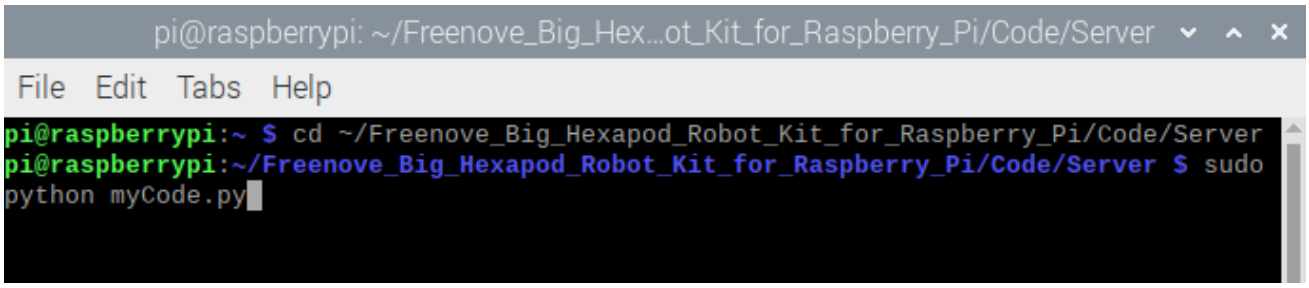


Type the following command to enter the directory where myCode.py is located.

```
cd ~/Freenove_Big_Hexapod_Robot_Kit_for_Raspberry_Pi/Code/Server
```

Run myCode.py

```
sudo python myCode.py
```



Result:

The robot moves straight forward first and then walks to the right (Gait Mode1 and Action Mode1), and then it moves back and turn right.(Gait Mode2 and Action Mode2)

Related py Files

Some important functions included in the py files are listed below. For more detailed code, please refer to the corresponding py file.

ADS7830.py

Function	Description
voltage(channel)	Returns the voltage value of corresponding channel
batteryPower()	Returns the voltage value of the two sets of batteries

Buzzer.py

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

Contorl.py

Function	Description
coordinateToAngle(x,y,z)	Enter the foot coordinate and it returns the rotation angle of servos on each joint.
angleToCoordinate(a,b,c)	Enter the joint angles and it returns the foot coordinate.
Condition()	Process the action command
coordinateTransformation()	Transformation of coordinate system
map()	Mapping a value from one range to another
Position(x,y,z)	Control the in-situ movement of the body in x,y,z directions
postureBalance(r,p,y)	Control in-situ twist of the body(deflection angle, roll Angle and course angle)
run()	Control the movement of the robot with two gait modes and two action modes.

IMU.py

Function	Description
imuUpdate()	Update the current deflection angle, roll angle and course angle of the robot

Led.py

Function	Description
ledIndex(Index, R, G, B)	Turn ON a single LED and set its color
wheel(pos)	Generate RGB value of different colors
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 7 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)	Rotate the servo of specified channel to a specified angle

Ultrasonic.py

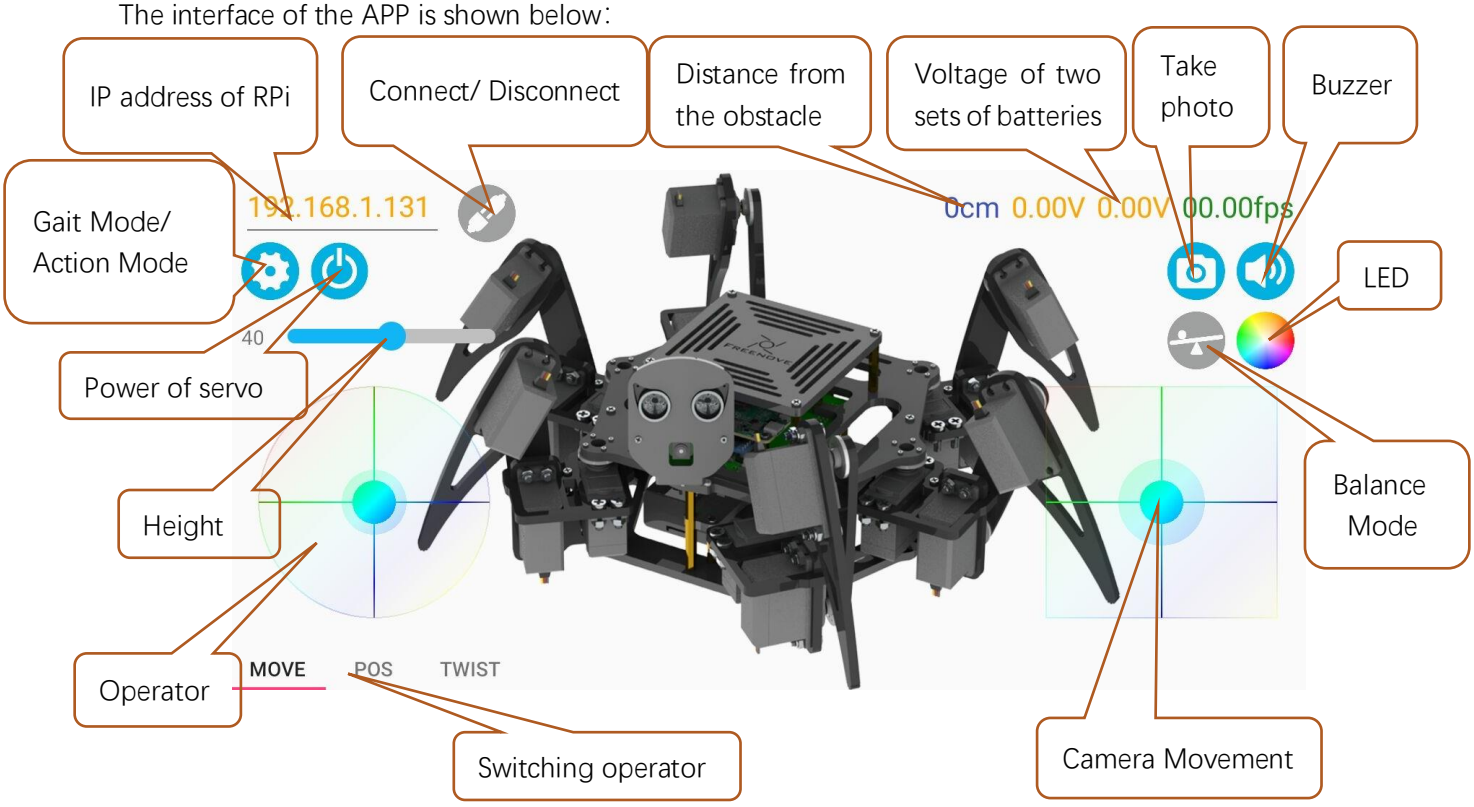
Function	Description
getDistance()	Obtain the distance between the ultrasonic module and the barrier in front of it.

Android and iOS 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.Freenovez
On GitHub:
https://github.com/Freenove/Freenove_App_for_Android
In this github repository, you can find the App instruction (Tutorial.pdf).

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

The interface of the APP is shown below:



The servos and the Raspberry Pi are powered by two separate, independent power supplies. When the servos are powered OFF, the power supply of RPi remains its original state.

What's Next?

THANK YOU for participating in this learning experience!

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

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

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

Thank you again for choosing Freenove products.