Adeept

PiCar-Pro

www.adeept.com

About Adeept

Adeept is a company dedicated to open source hardware and STEAM education services. The Adeept creative technology team is constantly developing new technologies, using excellent products as technology and service carriers, providing complete tutorials and after-sales technical support to help users enjoy learning and entertainment. The main products include Arduino, Raspberry Pi and various learning kits and robots of BBC micro:bit. Adeept is committed to providing users with high-quality STEAM education products, services and technical support.

Technical Support: support@adeept.com

Customer Service: service@adeept.com

Catalog

Introduction of PiCar Pro Products

1. About PiCar Pro products

PiCar Pro is an open source intelligent robot product for artificial intelligence and robot enthusiasts and students. It is also an open Raspberry Pi-based robot development platform. It has the following features:

Easy to assemble: adopting structural modular design, open hardware list and detailed assembly tutorial.

Easy to learn: Complete and detailed development learning tutorials and sample codes are provided from algorithms to applications.

Multi-form: It can be transformed into different types of carts by different combinations, such as robotic arm carts and ultrasonic carts.

Rich functions: automatic obstacle avoidance, color recognition, color tracking, moving object detection, Web remote control, OLED display, lighting indicator, line patrol module.

Aluminum alloy structure: strong and durable.

Extensible: the structure can be expanded and DIY.

Web remote control: Regardless of mobile phone, tablet, computer, Windows, Linux, Mac OS, as long as you can install Google Chrome browser to control the robot.

Support multiple versions of Raspberry Pi: support Raspberry Pi 3B, Raspberry Pi 3B+ and Raspberry Pi 4.

Supporting Python.

2. Two types of PiCar Pro robots

PiCar Pro can be assembled into two different forms of robots, namely the robotic arm form and the ultrasonic form.

2.1 Robotic arm shape

The PiCar Pro robotic arm has the functions of grabbing objects and visual line tracking, color recognition, moving object detection, Web remote control, OLED display, and lighting indicators; PiCar Pro uses a decelerating DC motor as a power unit, and its advantage is fast speed ; The use of large-size wheels makes PiCar Pro have excellent off-road performance and can be applied to complex terrain; the program only needs to control the high and low levels of the corresponding GPIO ports to control the car to grab items, which is convenient for novices and makers to learn and use the control methods of car products;

2.2 Ultrasonic form

PiCar Pro has visual line patrol and ultrasonic line patrol, automatic obstacle avoidance, color recognition, moving object detection, Web remote control, OLED display, lighting indicator and other functions; PiCar Pro uses a decelerating DC motor as a power unit, and its advantage is fast; using large-size wheels, PiCar Pro has excellent off-road performance and can be applied to complex terrain; the program only needs to control the high and low levels of the corresponding GPIO ports to control the motor rotation, which is convenient for novice makers to learn quickly about the control method of using trolley products;

Raspberry Pi

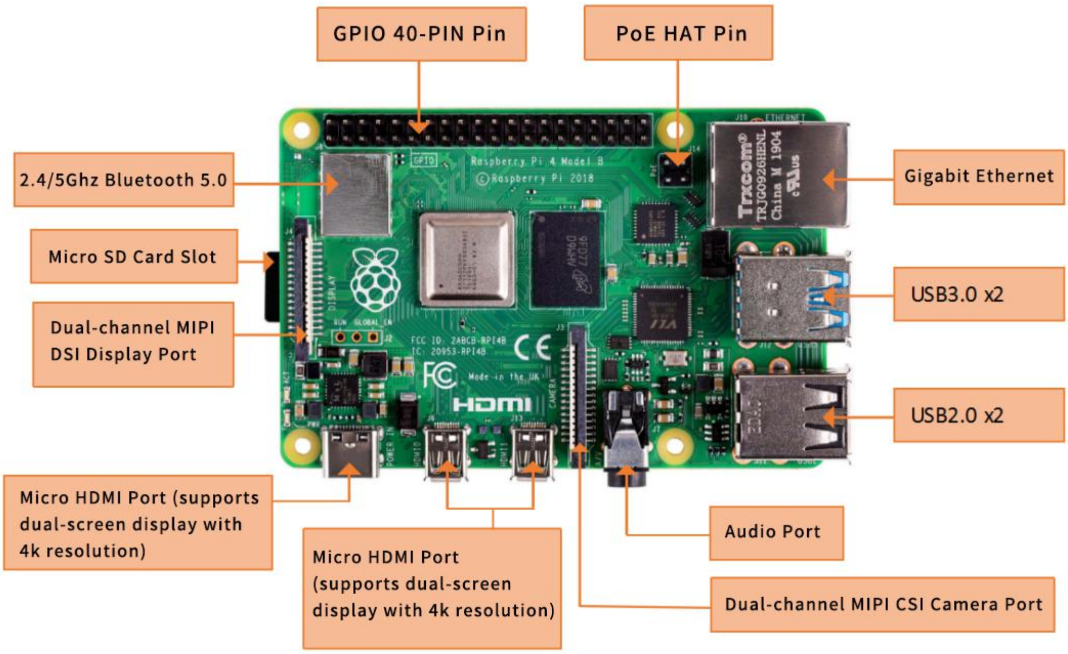
1. Introduction to Raspberry Pi

(1) Raspberry Pi

Raspberry Pi (Raspberry Pi, RasPi/RPi) is developed by the British charity organization "Raspberry Pi Foundation", based on ARM microcomputer motherboard, only the size of a credit card, but has the basic functions of a personal computer. The original purpose of the Foundation's development of the Raspberry Pi was to improve the teaching level of the school's computer science and related disciplines, and cultivate the youth's computer programming interest and ability. Nowadays, most people use the Raspberry Pi for embedded development, which is mostly used in the Internet of Things, smart home and artificial intelligence.

(2) Raspberry Pi motherboard

In our lessons, we will use the Raspberry Pi 4 motherboard. Let's take a look at the structure of the Raspberry Pi 4 motherboard. As shown in the following figure:



The following contents will briefly explain the main structure ports of the Raspberry Pi 4 motherboard:

(1) GPIO 40-PIN pin:

The General Purpose Input Output (GPIO) is designed as a slot with two rows of pins on the Raspberry Pi motherboard. GPIO can be used to connect various peripheral electronic devices and sensors to control or monitor these devices through input/output level signals. For example, you can use GPIO to control the speed of a DC motor, or read the measured distance of an ultrasonic sensor. These functional characteristics of GPIO make the Raspberry Pi different from ordinary computer motherboards because it gives developers the freedom to operate manually. We will further introduce GPIO in the subsequent chapters and use them extensively.

(2) Gigabit Ethernet port:

The Ethernet interface allows the Raspberry Pi to connect to the computer network in a wired manner, which allows us to easily access the Internet or log in to the Raspberry Pi remotely. The Raspberry Pi's Ethernet interface is implemented using a USB bus, and data is transferred through the USB bus. Most models of Raspberry Pi provide an Ethernet interface

(3) Micro HDMI port:

High-definition multimedia interface (High Definition Multimedia Interface, HDMI) is a fully digital video and sound transmission interface, used to transmit uncompressed audio and video signals. By connecting it to a display (or TV) equipped with an HDMI interface, the content of the Raspberry Pi can be displayed. The HDMI interface can transmit video and audio signals at the same time, so when we use it, we don't need to connect speakers to the audio interface of the Raspberry Pi. If we really need to play sound through the audio interface, we need to modify the operating system configuration accordingly.

(4) USB2.0/3.0 port:

The Universal Serial Bus (USB) interface is the most common interface on a computer. You can use it to connect devices such as keyboards, mice, USB flash drives, and wireless network cards. When the number of USB ports is not enough, we can also increase the number of USB ports through a USB hub.

(5) Audio port:

Audio interface (3.5mm headphone jack) When HDMI connection is not used, you can use the standard 3.5mm headphone jack speakers or headphones to play audio. At the same time, the interface also integrates a composite video interface with a composite audio and video output function, which is generally used to connect to old models of TVs, and is currently rarely used.

(6) MIPI CSI camera port:

The CSI interface can be used to connect the CSI camera to the Raspberry Pi via a ribbon cable for easy video recording and image capture. Compared with the USB camera, this camera module has better performance.

(7) USB-C 5V/3A power supply port:

The Micro USB power supply interface is one of the main power supply methods of the Raspberry Pi. The rated voltage is 5V. The standard current requirements of different versions of the Raspberry Pi are slightly different. For example: the 1B type only needs 700mA, and the 3B+ type requires 2.5A. The chargers of many Android mobile phones can provide the necessary voltage and current for the Raspberry Pi. The current demand of the Raspberry Pi is also related to the connected external device. It is recommended that it should be calculated in advance when using it. Choose a suitable current (power) power supply for the Raspberry Pi. When the external device has a large power, an independent power supply should be used Power supply for external devices.

(8) Micro SD card slot:

The SD card slot is located on the back of the Raspberry Pi motherboard. The SD/MicroSD card is an essential storage part of the Raspberry Pi. It is used to install the operating system and store data. The capacity of the SD card should be above 2GB. In order to have a better experience, it is recommended to equip your Raspberry Pi with a large-capacity (above 16G) high-speed (Class10 or above) SD card.

(9) Bluetooth port:

The Bluetooth function allows the Raspberry Pi to connect with Bluetooth-enabled devices (such as a mouse, keyboard, and handle).

(10) PoE HAT port:

Active Ethernet (Power Over Ethernet, PoE) refers to a technology that uses Ethernet for power transmission. On the basis of the original Micro USB and GPIO power supply, the Raspberry Pi 3B+ type adds a new power supply method over Ethernet. Users can use the network cable to supply power to the Raspberry Pi without the need to configure an additional power supply, which is convenient for certain application scenarios.

(11) MIPI DSI display port:

You can connect the LCD display to the Raspberry Pi, which is generally used for embedded product development. Under normal circumstances, the HDMI interface can already meet the demand.

(3)Operating system

The Raspberry Pi supports a variety of operating systems, mainly based on Liunx and Windows, and most of them can be found on the official website of the Raspberry Pi Foundation (www.raspberrypi.org). The following briefly introduces two representative operating systems.

(1) Raspbian

Raspbian is the official operating system of the Raspberry Pi Foundation. It is customized based on Debian GNU/Linux and can run on all versions of the Raspberry Pi motherboard. According to the experience, Raspbian and Raspberry Pi combine the best, stable operation, powerful, easy to use, can basically meet various application needs, so it is strongly recommended to use Raspbian as the preferred operating system for Raspberry Pi. In the following chapters, we will further introduce the use of Raspbian in detail, and develop various applications on it.

(2) Windows 10 IoT Core

Windows 10 IoT Core is an operating system specifically created by Microsoft for the Internet of Things ecosystem. Windows 10 IoT Core is the core version of the Windows 10 IoT operating system. It has relatively simple functions and can run on the Raspberry Pi of type 2B or above. The installation and use of Windows 10 IoT Core will not be described in detail here. If you are interested, you can visit Microsoft's website for more information.

In addition to the two operating systems described above, there are several operating systems that support the Raspberry Pi, such as Ubuntu MATE, OSMC, LibreELEC, PiNet, RISC OS, etc. As for which one to choose, it depends on whether you want to use Raspberry What to do. If you want to use the Raspberry Pi as an ordinary computer or for electronic project development, then Raspbian is a very good choice. If you plan to use the Raspberry Pi as a media center, you can consider using OSMC or LibreELEC.

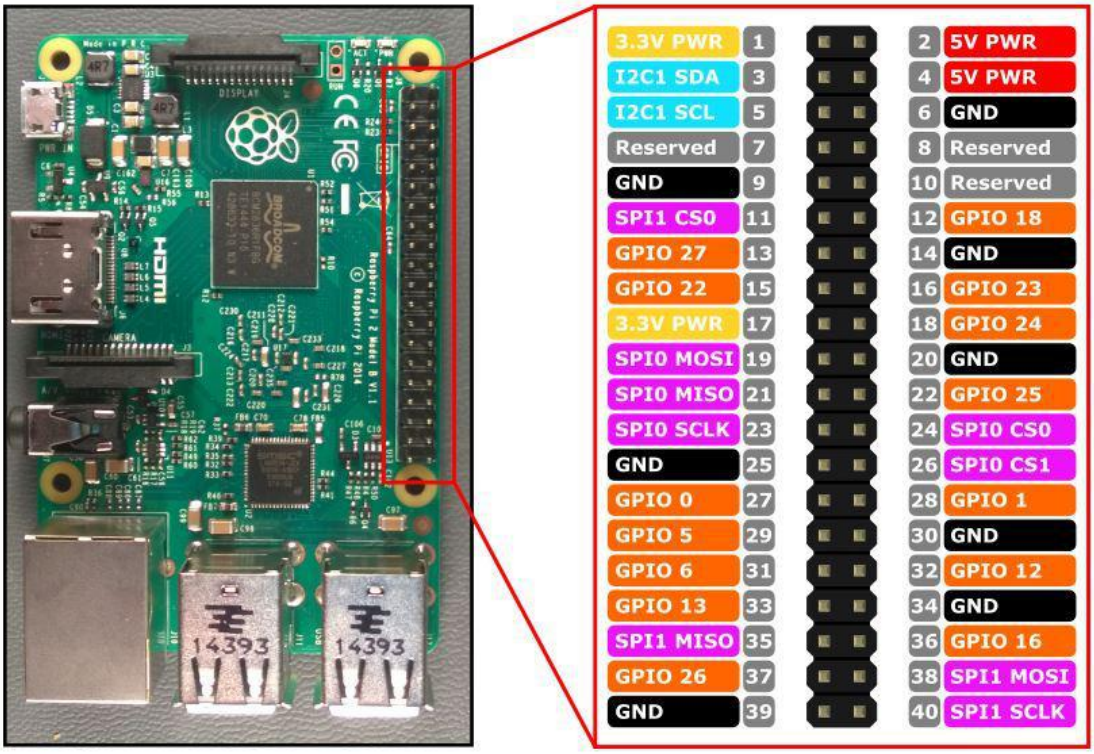
(4)Programming language

For the Raspberry Pi, there are many programming languages available. In fact, any language that can be compiled for the ARM architecture (such as the C language) can be used for the Raspberry Pi. The most popular language should be Python. In fact, the Pi in the name of the Raspberry Pi was inspired by the word Python. Python is an interpretive, object-oriented, and dynamic data type high-level programming language with powerful functions, good compatibility, and high reliability. Python programs are easy to write and read. At present, there are two major versions of Python: Python 2 and Python 3. Both versions have been updated and maintained, but people still have disputes about which version to use. You can visit Python's official website (www.python.org) to understand more related content, in the future we will mainly use Python 3 for development introduction. In addition, because the compatibility of the Raspberry Pi is splendid, the program we wrote on the 3B+ model can be run on the Zero W model with little modification.

2. Introduction to GPIO

(1) What is GPIO

GPIO (General Purpose I/O Ports) are general-purpose input/output ports. In layman's terms, they are some pins with two rows of pins. They can be used to output high and low levels or to read the state of the pins-whether it is high or low. Users can interact with the hardware through the GPIO port (such as UART), control the work of the hardware (such as LED, buzzer, etc.), read the working status signal of the hardware (such as interrupt signal), etc.



(2) Introduction of GPIO pins

(1) GPIO pin comparison table

Raspberry Pi 40Pin Pin Comparison Table

wiringPi Encoding	BCM Encoding	Function Name	BOARD Encoding of Physical Pins		Function Name	BCM Encoding	wiringPi Encoding
		3.3V	1	2	5V		
8	2	SDA.1	3	4	5V		
9	3	SCL.1	5	6	GND		
7	4	GPIO.7	7	8	TXD	14	15
		GND	9	10	RXD	15	16
0	17	GPIO.0	11	12	GPIO.1	18	1
2	27	GPIO.2	13	14	GND		
3	22	GPIO.3	15	16	GPIO.4	23	4
		3.3V	17	18	GPIO.5	24	5
12	10	MOSI	19	20	GND		
13	9	MISO	21	22	GPIO.6	25	6
14	11	SCLK	23	24	CE0	8	10
		GND	25	26	CE1	7	11
30	0	SDA.0	27	28	SCL.0	1	31
21	5	GPIO.21	29	30	GND		
22	6	GPIO.22	31	32	GPIO.26	12	26
23	13	GPIO.23	33	34	GND		
24	19	GPIO.24	35	36	GPIO.27	16	27
25	26	GPIO.25	37	38	GPIO.28	20	28
		GND	39	40	GPIO.29	21	29

【Form description】:

(1) Three naming (coding) methods for Raspberry Pi pins

Three ways to name the Raspberry Pi pins:

The WiringPi number is the pin number of the functional wiring (such as TXD, PWM0, etc.); the BCM number is the Broadcom pin number, also known as GPIO; the physical number is the number corresponding to the physical location of the pin on the Raspberry Pi motherboard (1 ~40).

(2) 3.3V/5V pin and GND pin

3.3V/5V pin and GND pin are commonly known as power and ground pins. The power and ground pins allow your Raspberry Pi to power some external components, such as LED lights. It should be noted that before using these pins to power any external modules or components, care should be taken. Excessive operating current or peak voltage may damage the Raspberry Pi. Do not use voltages greater than 5V!

(3) SDA and SCL pins

The SDA and SCL pins constitute the I2C interface. I2C is a simple, bidirectional two-wire synchronous serial bus developed by Philips. It only requires two wires to transfer information between devices connected to the bus. The Raspberry Pi can control multiple sensors and components through the I2C interface. Their communication is done through SDA (data pin) and SCL (clock speed pin). Each slave device has a unique address, allowing rapid communication with many devices. The ID_EEPROM pin is also an I2C protocol, which is used to communicate with HATs.

(4) SCLK, MOSI and MISO pins

SCLK, MOSI and MISO pins form the SPI interface. SPI is a serial peripheral interface, used to control components with a master-slave relationship, and works in a slave-in, master-out and master-in-slave manner. The SPI on the Raspberry Pi consists of SCLK, MOSI, and MISO interfaces, and SCLK is used for controlling data speed, MOSI sends data from the Raspberry Pi to the connected device, while MISO does the opposite.

(5) TXD and RXD pins

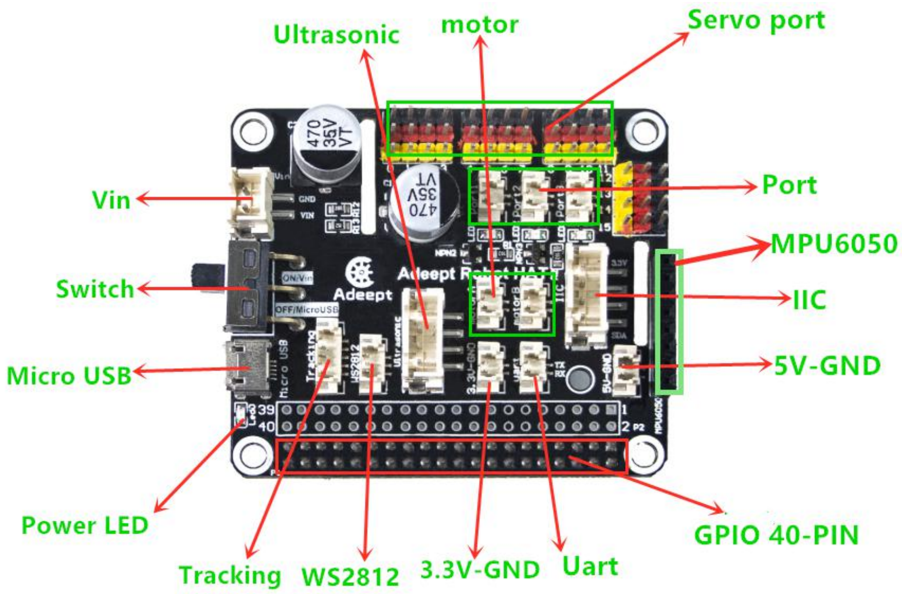
TXD and RXD form a UART interface. TXD is a pin to send data, and RXD is a pin to receive data. A friend who uses Arduino must have heard of UART or Serial. The Universal Asynchronous Receiver/Transmitter interface is used to connect the Arduino to the computer for which it is programmed. It is also used for communication between other devices and the RX and TX pins. If the Raspberry Pi has a serial terminal enabled in raspi-config, you can use these pins to control the

Raspberry Pi through a computer or directly to control the Arduino.

Introduction of Robot HAT Driver Board

1.Introduction of Robot HAT driver board

When you get the PiCar Pro product, you will see a board with its name printed on it called: Adeept Robot HAT, which is an important part of PiCar Pro. There are many interfaces on the Robot HAT driver board. By these interfaces, you can connect some sensors and electronic hardware modules, so that you can achieve many extended functions. Our PiCar Pro products need to be used in conjunction with the Raspberry Pi. Let's first get to know the Robot HAT driver board.



【Vin】： The vin interface is an interface for external power supply.

【Switch】：Switch is the switch of Robot HAT driver board, ON is to open, and OFF is to close.

【Micro USB】：The Micro USB interface can connect the Robot HAT driver board to a computer or other equipment, and can also supply power for the Robot HAT driver board.

【Power LED】 Power LED is used to indicate the power status of Robot HAT driver board. If the LED is on, it means that the Robot HAT driver board is powered on and can run; if the LED is off, it means that the Robot HAT driver board is not powered on.

【Tracking】 is the pin interface of Tracking Module.

【WS2812】 is the pin interface of WS2812 Module.

【3.3V-GND】 3.3V power supply interface.

【Uart】 Uart interface.

【GPIO 40-PIN】 General Purpose Input Output (GPIO) is designed as a slot with two rows of pins on the Robot HAT driver board. GPIO can be used to connect various peripheral electronic devices and sensors and control or monitor these devices with input/output level signals. In PiCar Pro products, this GPIO interface is connected to the GPIO pins on the Raspberry Pi driver board.

【5V-GND】 5V power supply interface.

【IIC】IIC interface. It is also the interface of the OLED screen module.

【MPU6050】The interface of MPU6050 sensor.

【Port】is divided into Port1, Port2, and Port3 interfaces, which are commonly used to connect Small LED light.

【Servo port】Servo interface.

【motor】 is divided into motor1, motor2 interfaces.

【Ultrasonic】Ultrasonic interface.

2. Precautions for the use of Robot HAT driver board

When you are performing software installation, structural assembly or program debugging, you can use a USB cable to power the Raspberry Pi. If the Raspberry Pi is equipped with Robot HAT, you can connect the USB cable to the USB port on the Robot HAT. Robot HAT will power the Raspberry Pi by the GPIO interface.

Different Raspberry Pi have different current requirements. For example, the Raspberry Pi 3B needs at least 2A to boot up, and the Raspberry Pi 4 needs 3A to boot normally. When you use the power adapter to power the Raspberry Pi, you can check the specifications on your power adapter.

When the Robot HAT is connected to a load, such as a motor or multiple servos, you need to use a high-current power supply to connect to the Vin on the Robot HAT. You can use two 18650 batteries that support high-current to power the Robot HAT. For power supply, our product will provide a dual 18650 battery box with a 2pin interface. You can directly connect it to the Robot HAT.

When the USB interface on the Robot HAT is used for power supply, the switch of the Robot HAT does not control whether to supply power. The switch of the Robot HAT can only control the power supply of Vin.

Do not use the USB port on the Robot HAT and Vin to supply power at the same time. If you need to debug the program for a long time and don't want to remove the battery, you can set the switch on the Robot HAT to OFF, so that when the USB cable is used to connect the Robot HAT, the Robot HAT is powered by USB.

If your robot restarts automatically after it is turned on, or after it is turned on normally, it is disconnected and restarted at the moment when the robot starts to move, it is likely that your power supply does not output enough current. The robot will automatically restart when it is turned on. Run the program to place all the servos in the neutral position. The voltage drop generated during this process causes the Raspberry Pi to restart.

We have tested that the peak current of the robot is around 3.75A when powered by 7.4V, so you need to use a battery that supports 4A output.

You can also use the power lithium battery to power the Robot HAT. Robot HAT supports power supply below 15V.

When assembling and installing the servo rocker arm, you can use a USB cable to power the Robot HAT. After the Raspberry Pi with the robot software is installed, it will control the Robot HAT to set all the servo ports to output neutral signals. You can connect the servo to any port. The gear of the servo will rotate to the neutral position, and then you can install the servo rocker arm according to the specified angle. After the rocker arm is installed, you can disconnect the servo from the Robot HAT , When you need to install the rocker arm of the second servo, connect the second servo to any servo port on the drive board.

Lesson 1 Installing and Logging in to the Raspberry Pi System

In this lesson, we will learn how to install and remotely log in to the Raspberry Pi system under Windows. And we will download the code program to control the robot.

1.1 Preparation

(1) When studying this lesson, you need to prepare the following components first:

One SD card that has been formatted (we recommend using an SD card with memory above 16G), 1 card reader, Raspberry Pi development board.

(2) You need to insert the SD card into the card reader first, and then connect the card reader to the computer.

1.2 Downloading the Raspberry Pi system Raspbian

Raspbian is the official operating system of the Raspberry Pi Foundation. It is customized based on Debian GNU/Linux and can run on all versions of the Raspberry Pi motherboard. According to the experience, Raspbian combines Raspberry Pi the best. It is stable, powerful, and easy to use. It can basically meet the needs of various applications. This course uses Raspbian as the preferred operating system for the Raspberry Pi. Next, we will teach you how to download the Raspberry Pi system Raspbian. Now there are two ways to download Raspbian for Raspberry Pi system (we recommend method one first)

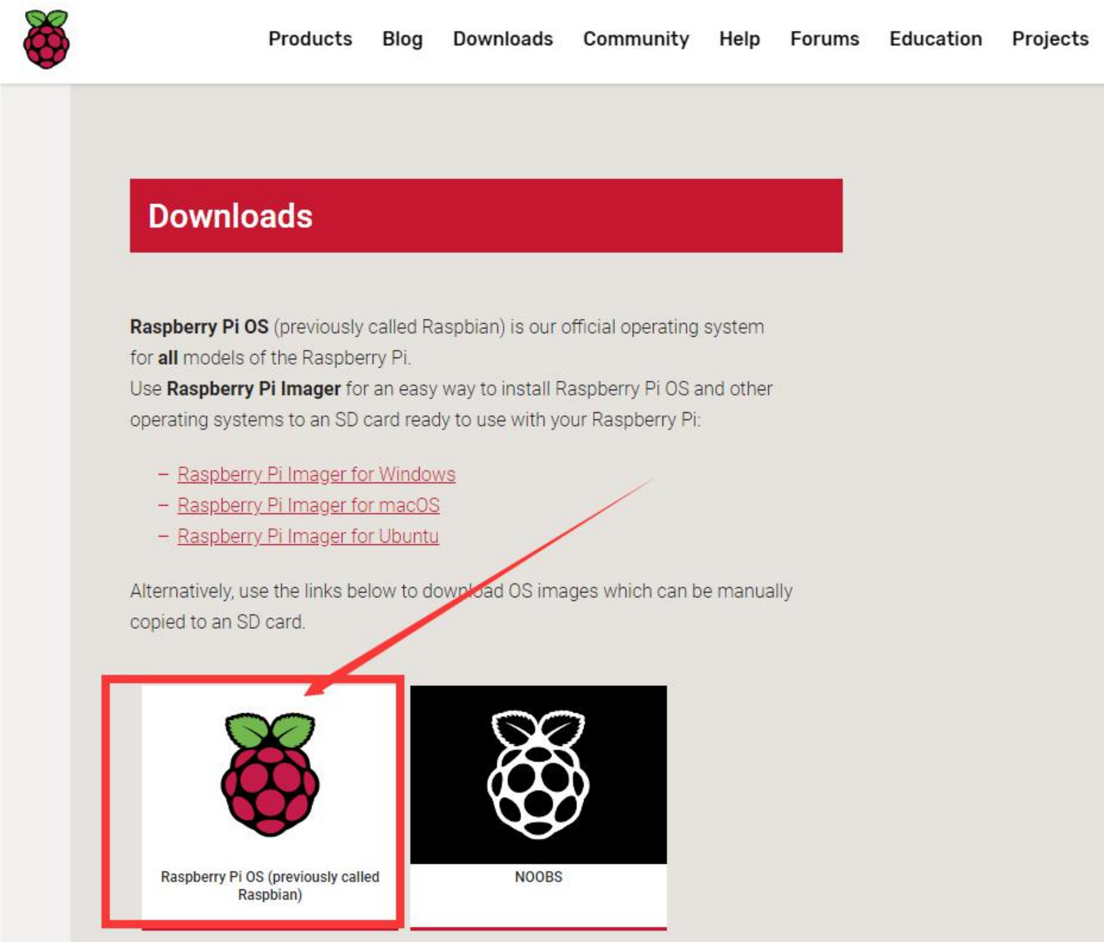
Method one :

(1) visit the official website of the Raspberry Pi through a browser to download

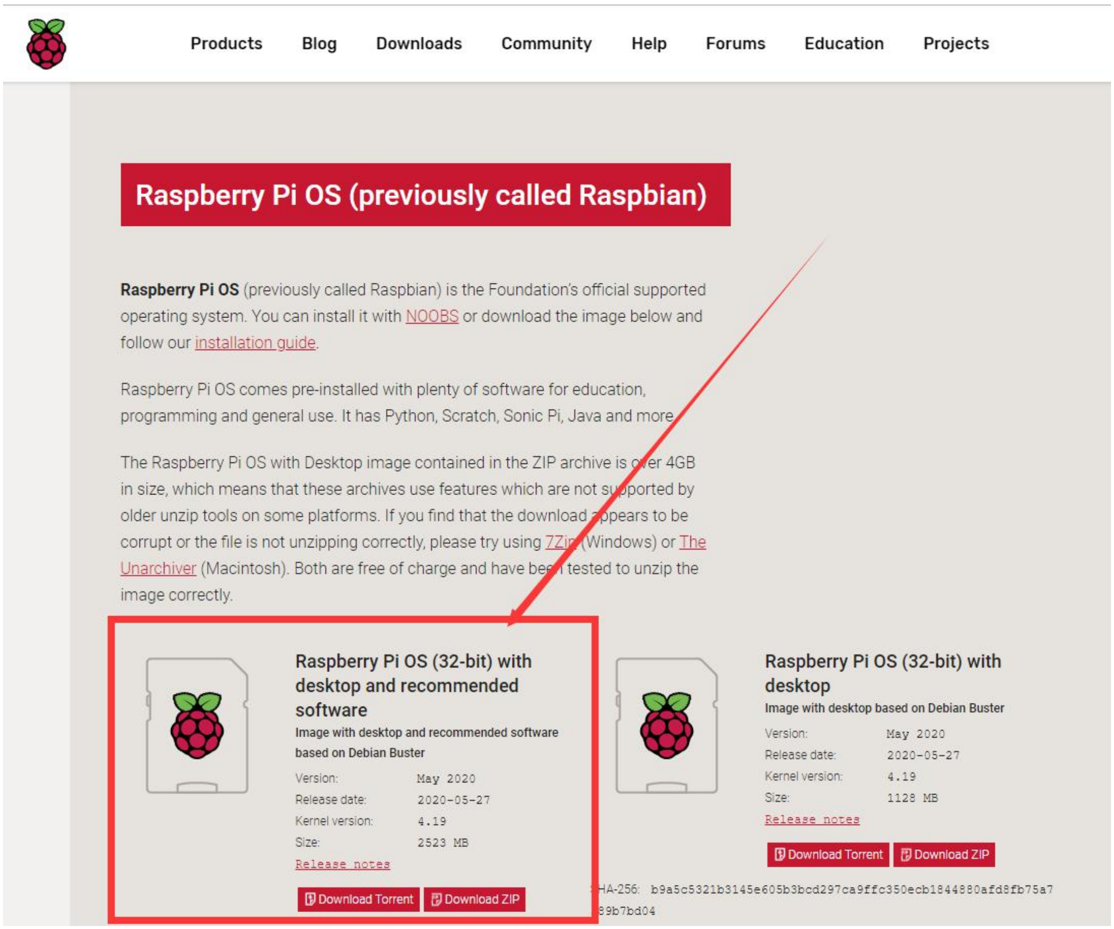
Raspbian:

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

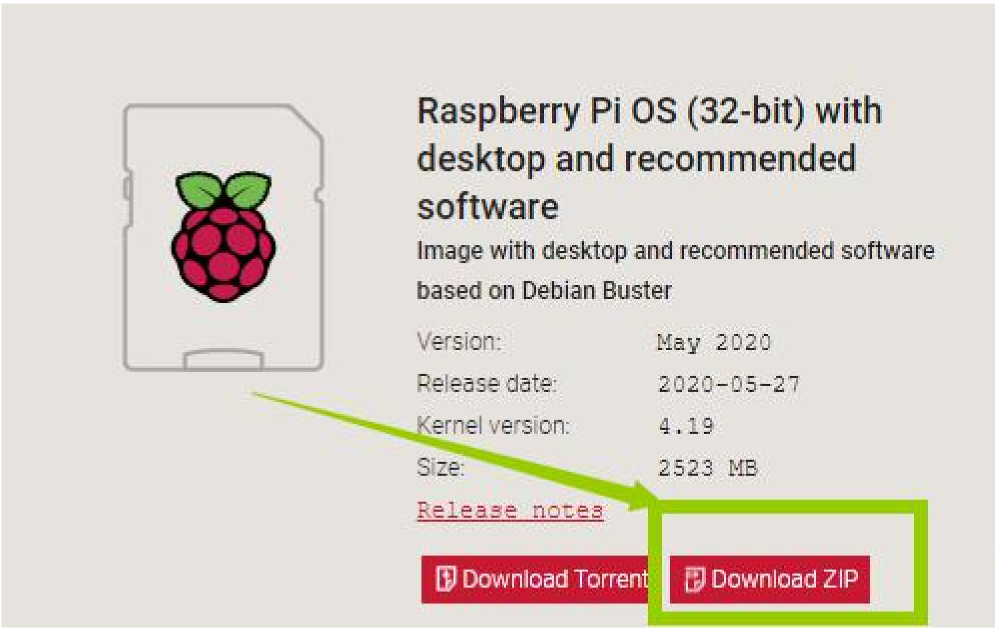
After logging in to the official website, click on the location shown below:



(2) We need to find out the Raspberry Pi OS (32-bit) with desktop and recommended software. It contains a complete desktop system and recommended software packages.



(3) Choose to download the ".ZIP" file and wait for the download to complete:



(4) Find the ".ZIP" file you just downloaded, double-click to open it, and extract it. The uncompressed file format of the file is ".img". Pay attention, you must name

the path of the uncompressed .img file all English letters without special characters.

Method two:

Manually downloading the image file we provide and write it to the SD card (not recommended).

The image downloaded according to the method one is the latest official version of Raspbian and comes with some pre-installed software. At the same time, the normal operation of the robot product requires many other dependent libraries, although we provide a script to install these simple methods of relying on libraries (will be introduced in detail later), occasionally encountering dependent library updates may cause the installation of dependent libraries to fail, so we provide a Raspbian mirror file pre-installed with dependent libraries. The disadvantage of this method is that the mirror files and related dependent libraries we provide cannot be kept updated at any time. Only when you encounter a very difficult problem, you can try this method to solve it.Download the Raspbian image file address we provide:

https://www.adeept.com/learn/detail-50.html

After the download is complete, decompress it. The path of the decompressed .img file must be all English letters and no special characters.

1.3 Burning the downloaded Raspberry Pi system to the SD card

We recommend using the Raspberry Pi Imager tool officially provided by the

Raspberry Pi. Raspberry Pi Imager is a new image burning tool launched by the Raspberry Pi Foundation. Users can download and run this tool on Windows, macOS and Ubuntu to burn the system image for the Raspberry Pi. Its usage is similar to Etcher and win32diskimager.

1.3.1 Downloading Raspberry Pi Imager

(1) Visit the official website of Raspberry Pi to download through a browser: https://www.raspberrypi.org/downloads/.

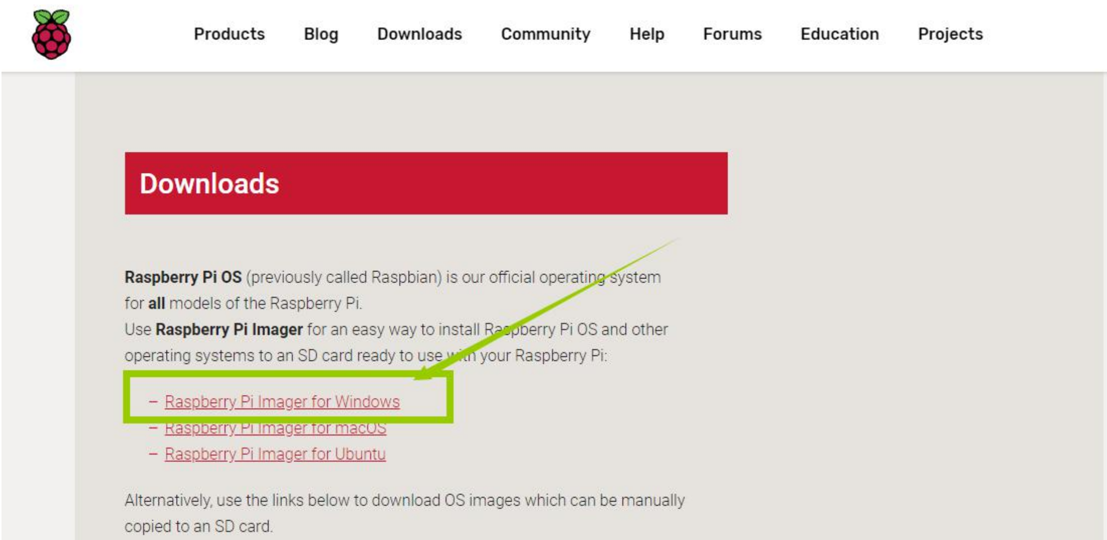
For Windows system, click "Raspberry Pi Imager for Windows" to download.

For Mac OS, click "Raspberry Pi Imager for macOS" to download.

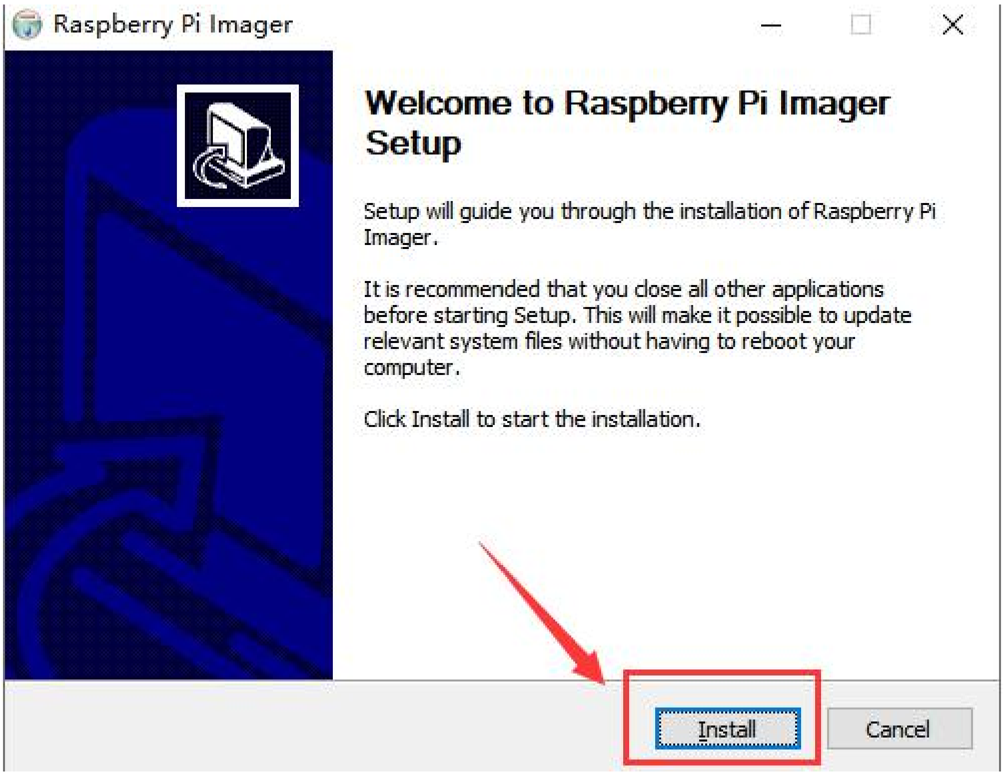
Linux Click "Raspberry Pi Imager for Ubuntu" to download.

After the download is complete, install the software and burn the Raspberry Pi system.

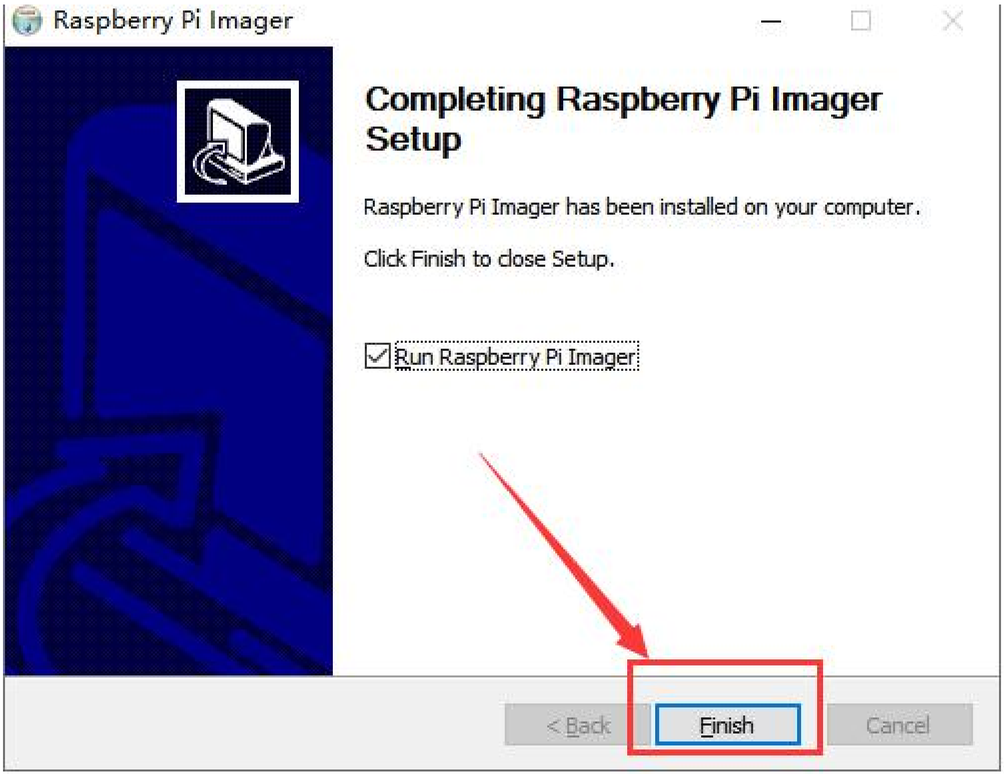
Now take Windows as an example.



(2) Open the downloaded file "imager.exe" and click "Install".



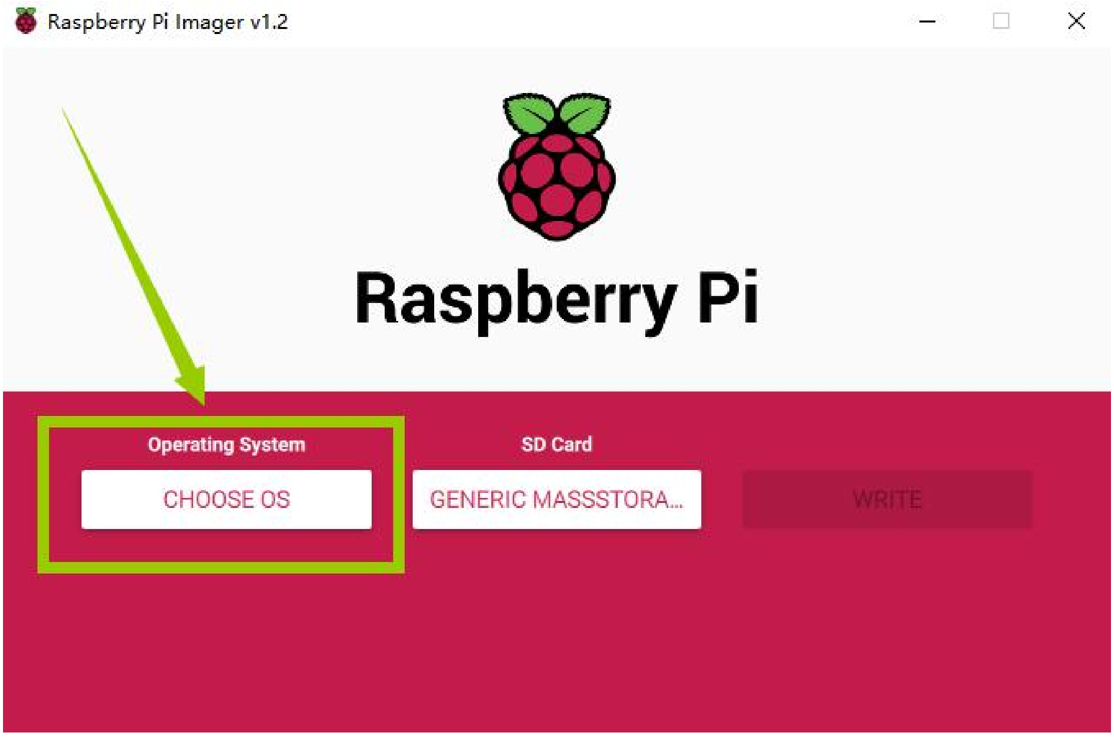
(3) Then click "Finish".



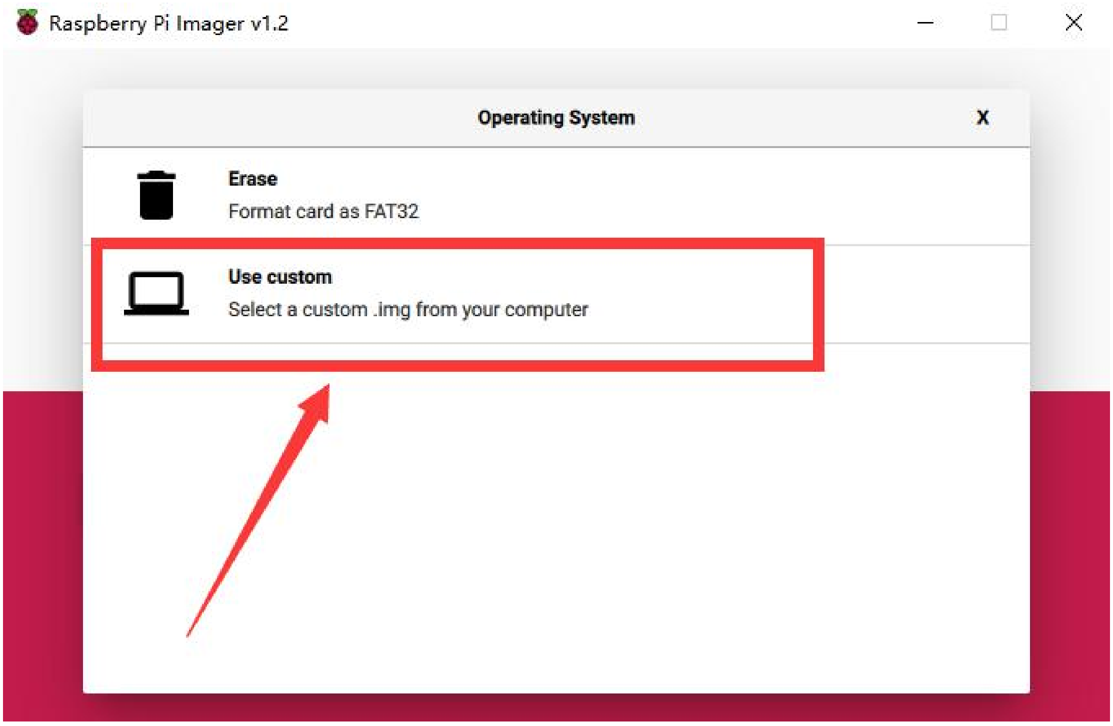
(4) The software interface after opening is as shown below:

1.3.2 Burning Raspberry Pi system to SD card with Raspberry Pi Imager

(1) Click "CHOOSE OS" on the opened Raspberry Pi Imager software interface.

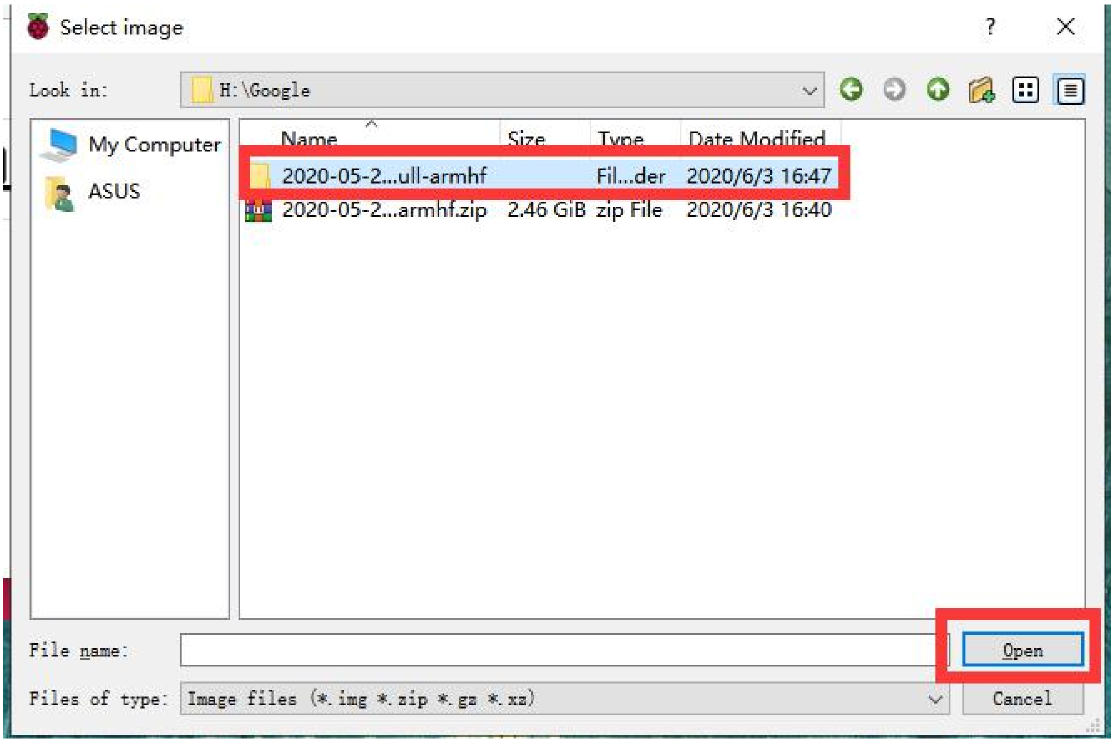


(2) Click "Use custom" and select a custom ".img" file from your computer, which is the ".img" file of the Raspberry Pi system that we downloaded and decompressed before.

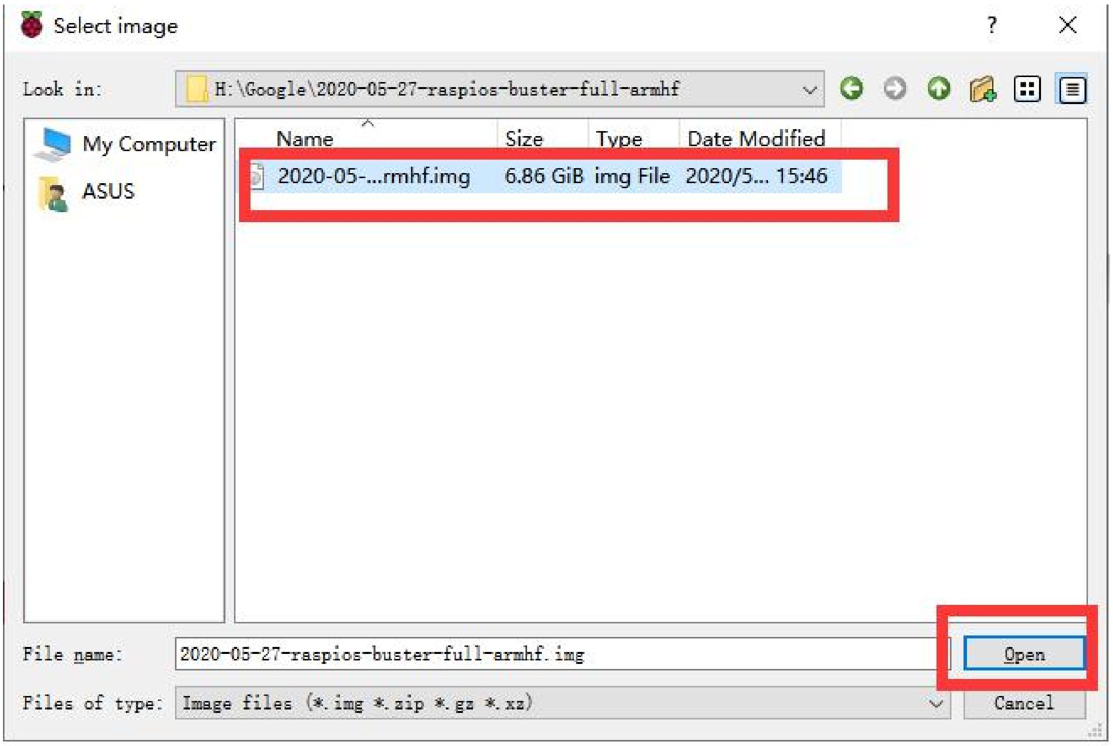


(3) Find the ".img" file of the Raspberry Pi system that we downloaded and

decompressed before. Click "Open".

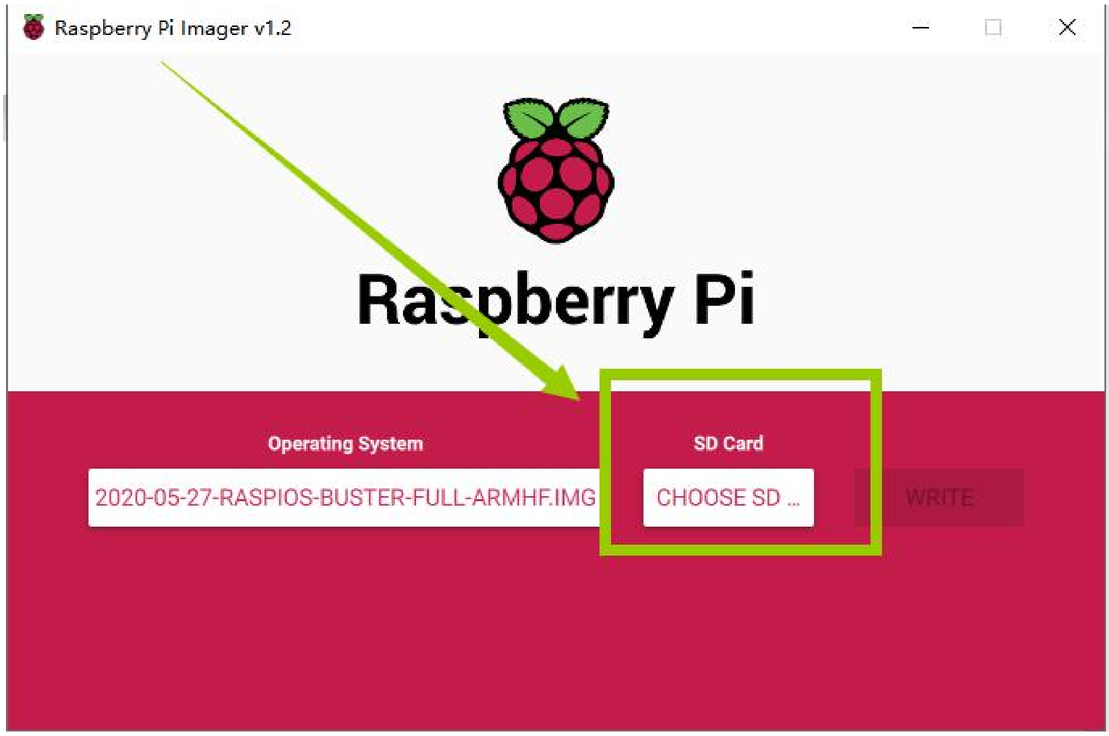


(4) Select the ".img" file and click "Open".

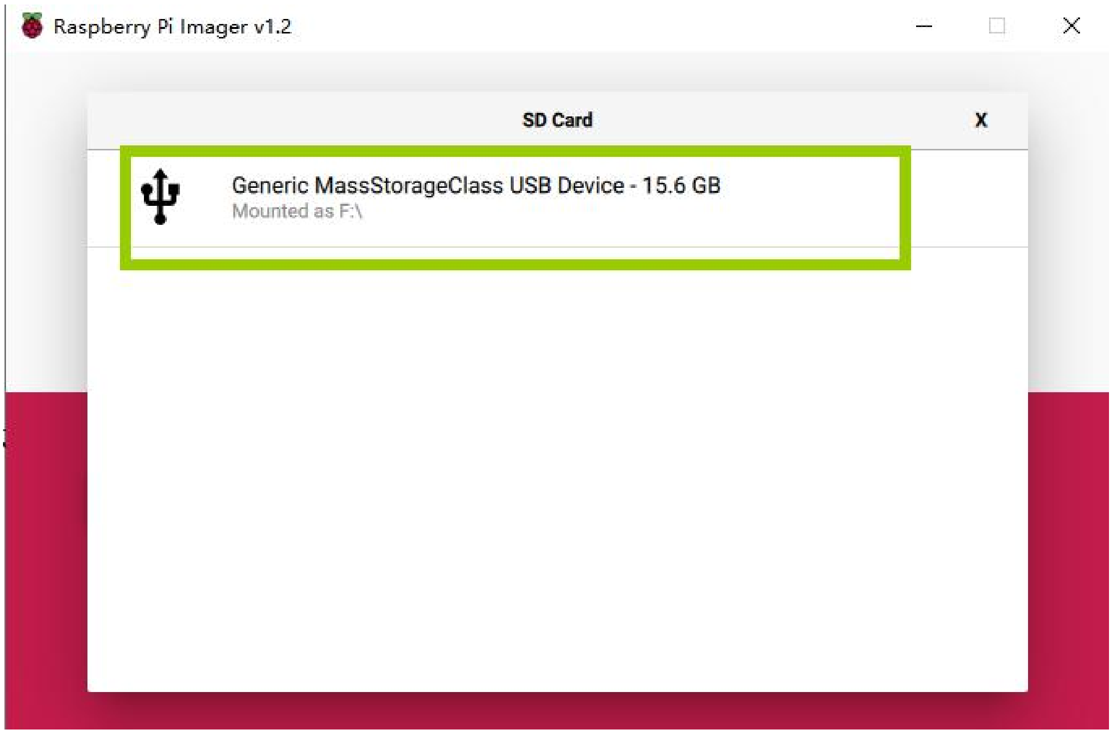


(5) Then on the interface of Raspberry Pi Imager, the ".img" file of our selected Raspberry Pi system will appear.

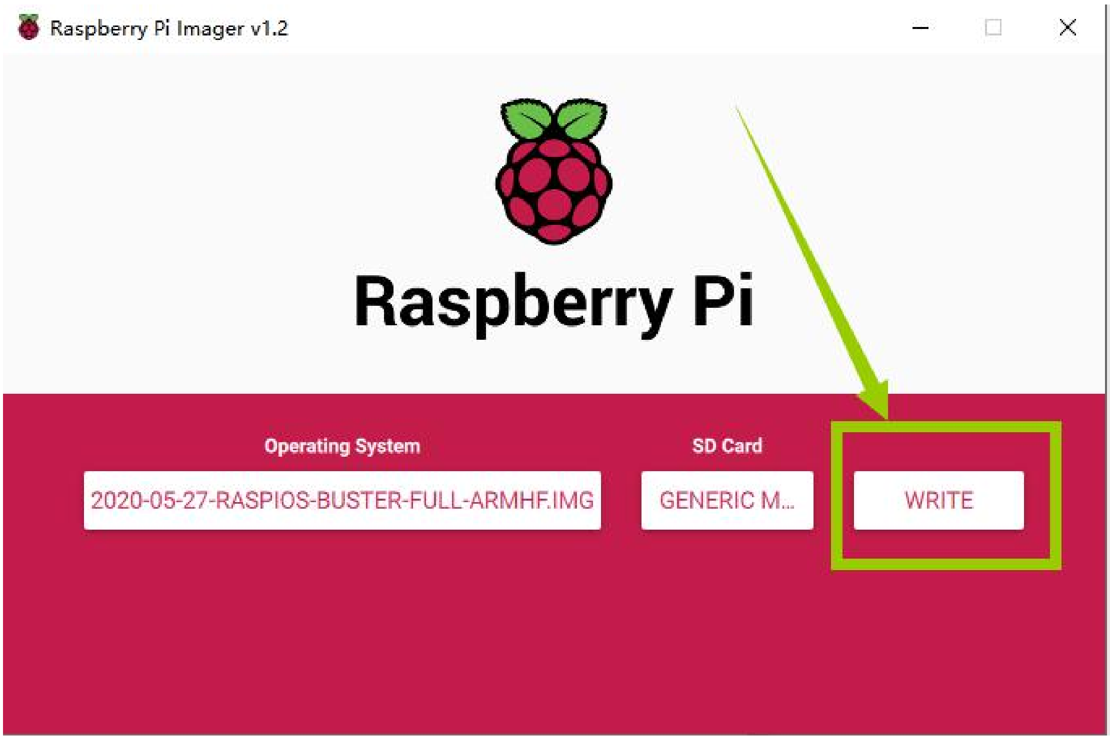
(6) Click "CHOOSE SD".



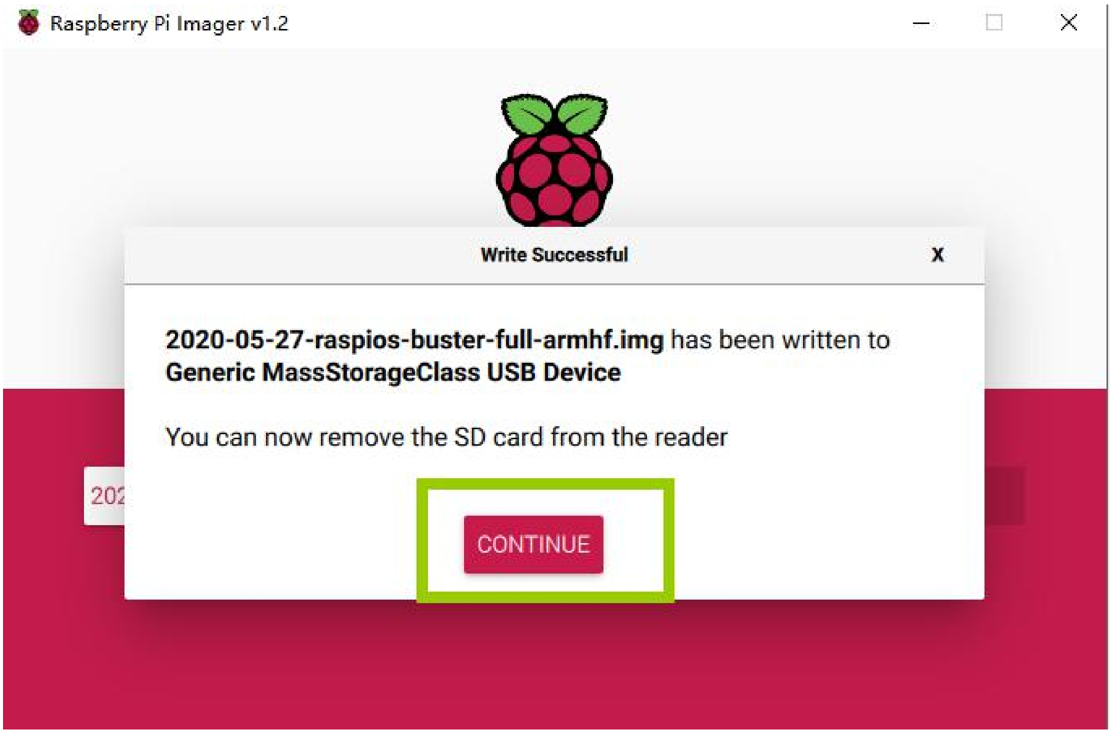
(7) Then select the SD card we need to burn.



(8) Click "WRITE" to write it to the SD card. Wait for the burn to complete.



(9) After the burning is completed, the following message will be prompted,indicating that the burning is finished, click "CONTINUE".



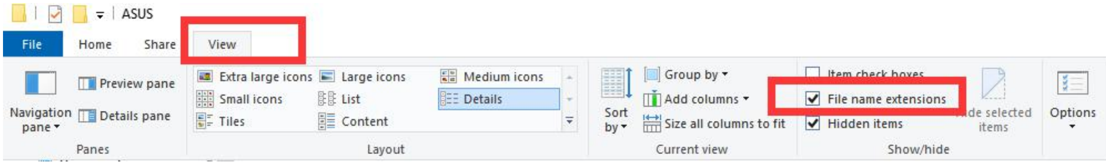
【Pay Attention】

Don't remove the SD card after burning! After the Raspberry Pi Imager is burned, the memory card will be ejected in the program. This will cause the subsequent copy operation to prompt that the SD card has not been found. You can unplug the card reader from the computer and then plug it into the computer again.It is necessary to configure SSH and WIFI connection later. At this time, once the SD card is put into the Raspberry Pi to boot, it may cause subsequent headless WIFI configuration failure.

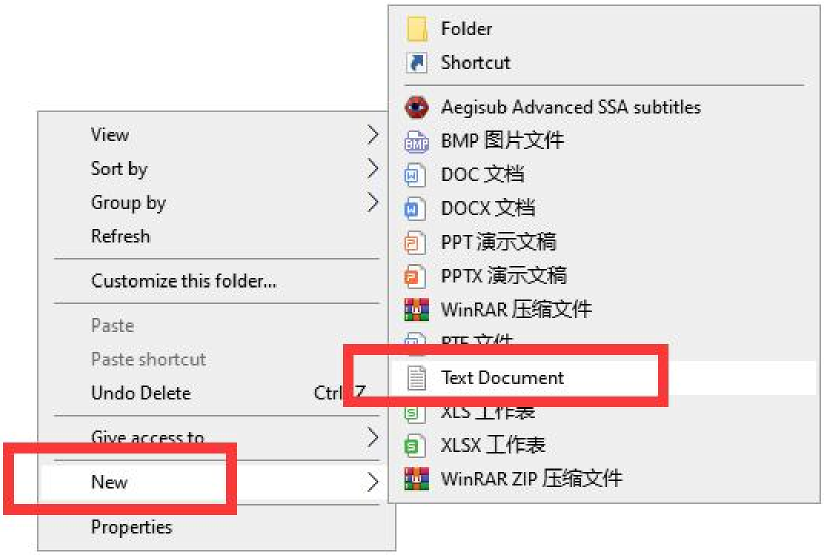
1.4 Starting the Raspberry Pi SSH service

SSH is a protocol designed to provide security for remote login sessions and other network services. Through the SSH service, you can remotely use the command line of the Raspberry Pi on another machine. In the subsequent operations and the process of using the Raspberry Pi, you can control the Raspberry Pi through another machine in the same local area network without connecting the mouse, keyboard and monitor to the Raspberry Pi. After 2016, Raspbian distributions disable the SSH service by default, so we need to manually enable it.

(1) We first enter the driver D of the computer, click "View" in the upper left corner, and select "File Extension", as shown below:

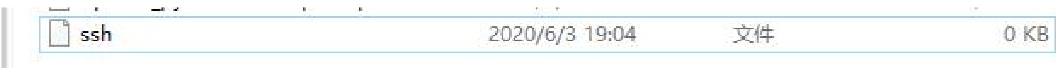


(2) Right-click on the blank space of the D drive, select "New", and select "Text Document".



(3) Name the file "ssh", as shown below:

(4) Then delete the suffix ".txt". We will get an ssh file without any extension. As shown below:



(5) Copy this ssh file to the root directory of the SD card of the Raspberry Pi system. When the Raspberry Pi starts, it will automatically find this ssh file. If it is found, it will start SSH. This method only needs to be used once. After that, every time you start the Raspberry Pi, it will automatically start SSH without repeating the

above operations. Copy the ssh file to the Raspberry Pi as shown below:

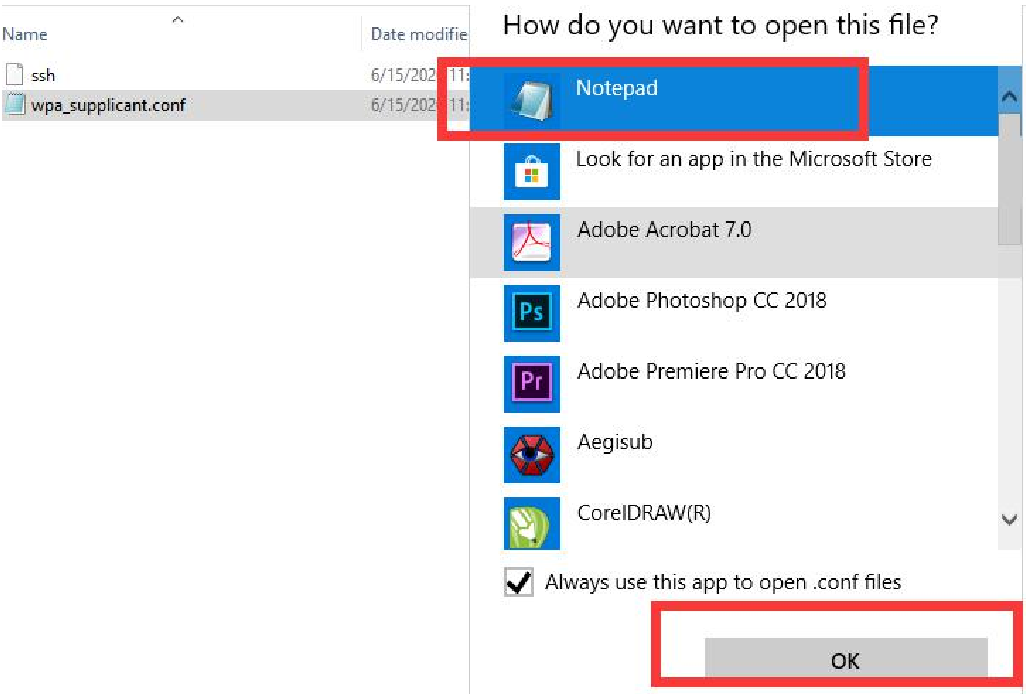
1.5 Setting up Raspberry Pi WIFI wireless connection

Next, we also need to set up a WIFI wireless connection for the Raspberry Pi.

(1) Create a new file named wpa_supplicant.conf in the root directory of the D driver of the computer.

(2) Click to select the wpa_supplicant.conf file, right-click the mouse, and select "Open Mode (H)".

(3) Select "Notepad" to open it.



(4) Write the following contents:

```
country=US
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
update_config=1
network={
ssid="WIFI"
psk="PASSWORD"
key_mgmt=WPA-PSK
priority=1
}
```

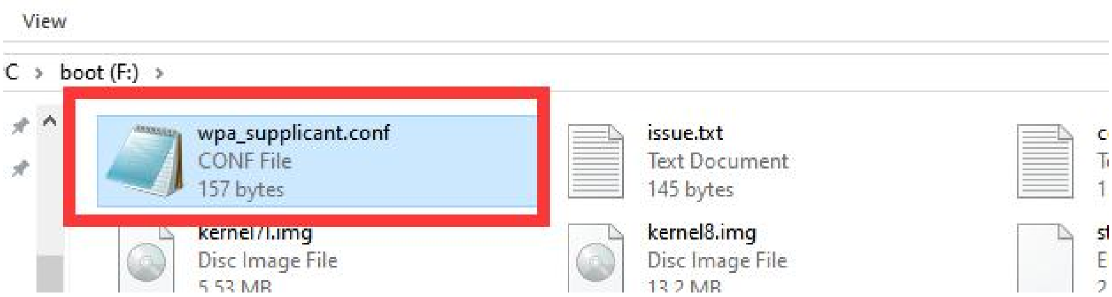
```
*wpa_supplicant.conf - Notepad
File Edit Format View Help
country=US
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
update_config=1
network={
ssid="WIFI"
psk="PASSWORD"
key_mgmt=WPA-PSK
priority=1
}
```

"Country" is your country code, do not modify it, the default is US; "ssid" needs to be changed to the name of the WIFI you want to connect; "psk" needs to be changed to the password of the WIFI you want to connect; other parts do not need to do any modifications.

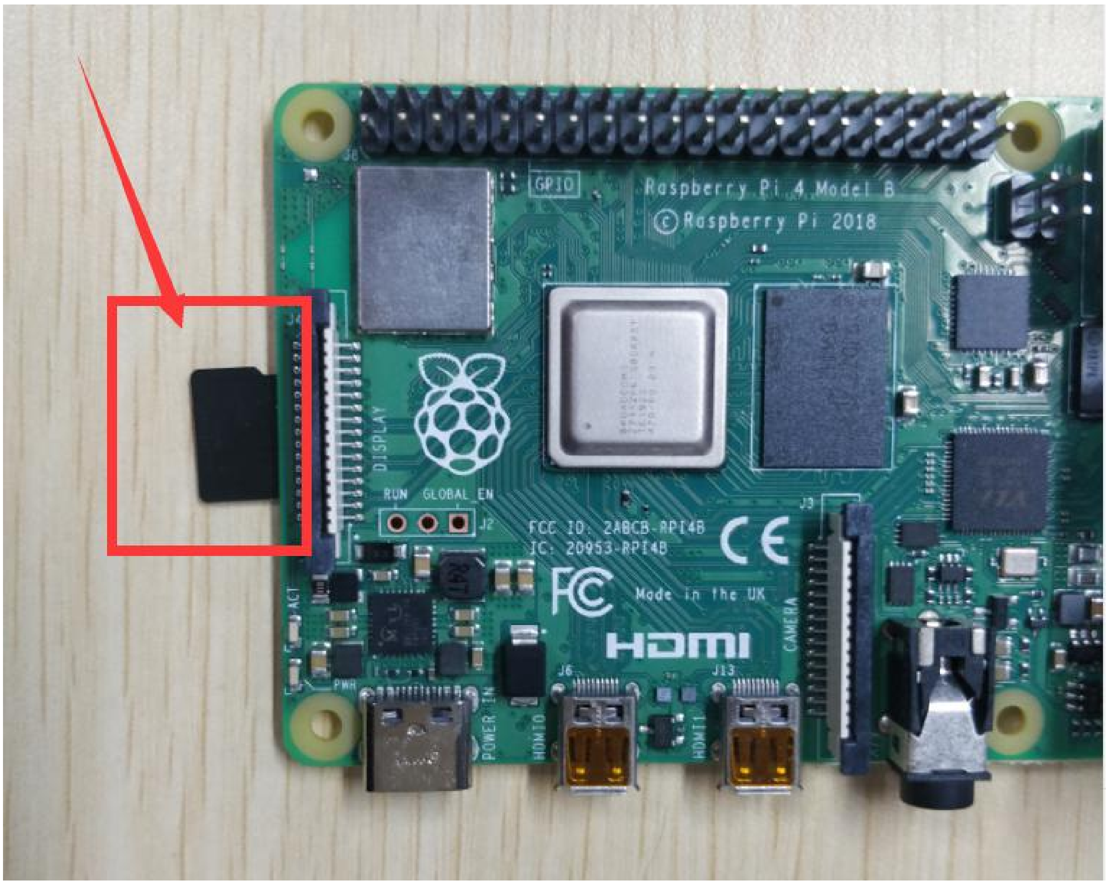
For example, our company's WIFI name is Adeept, WIFI password is 123456, and the modified wpa_supplicant.conf file is as shown below:

```
*wpa_supplicant.conf - Notepad
File Edit Format View Help
country=US
ctrl_interface=DIR=/var/run/wpa_supplicant GROUP=netdev
update_config=1
network={
ssid="Adeept"
psk="123456"
key_mgmt=WPA-PSK
priority=1
}
```

(5) Save the set wpa_supplicant.conf file, and then copy it to the root directory of the SD card of the Raspberry Pi system. As shown below:



(6) Now we can take out the SD card and put it into the "MICRO SD CARD" card slot on the Raspberry Pi development board, and use the Type-C data cable to supply power to the Raspberry Pi. And then the Raspberry Pi will start up and run.



1.6 Remotely logging in to the Raspberry Pi system

1.6.1 Obtaining the IP address of the Raspberry Pi

1.6.1.1 Obtaining an IP address with an external display

We provide a simple and fast way to get the Raspberry Pi IP address. You need to prepare the following components:

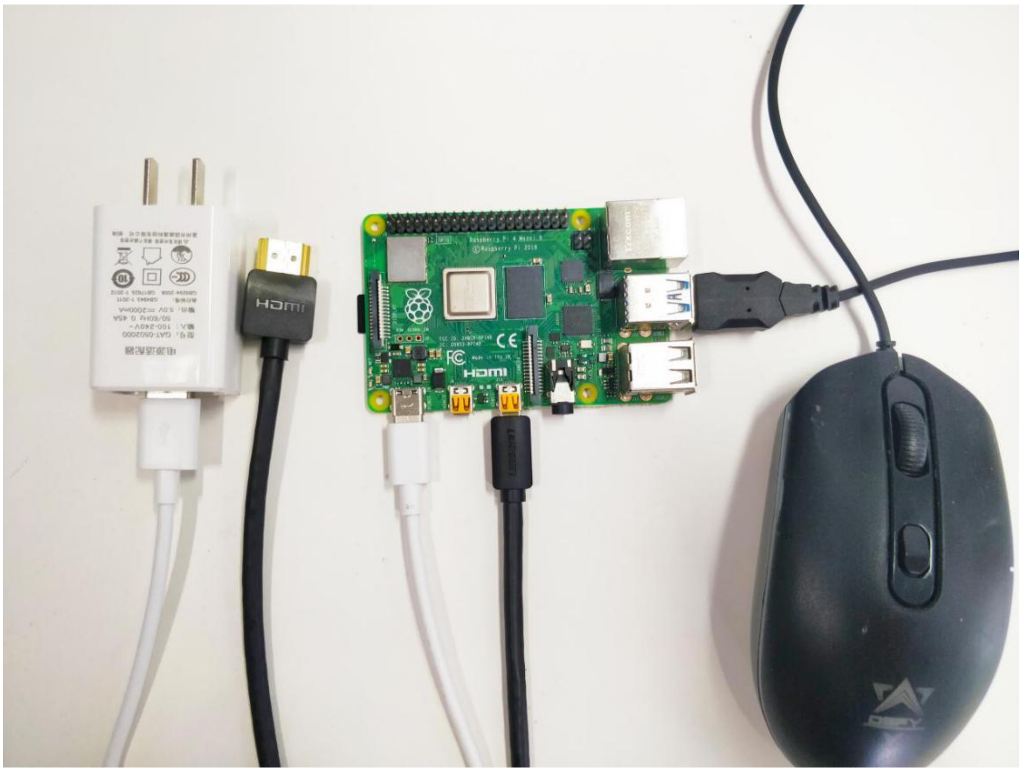
(1) One Type-C data cable: used to supply power to the Raspberry Pi.

(2) One HDMI cable: used to connect the monitor.

(3) One mouse: used to operate.

(4) One monitor

(5) One Raspberry Pi

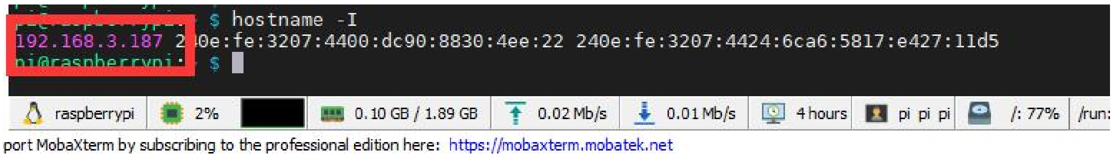


Connect the HDMI cable to the HDMI port of the monitor:

1.Turn on the monitor switch, and connect the mouse to the USB port of the Raspberry Pi, supply power to the Raspberry Pi with the Type-C data cable, then the Raspberry Pi starts. After entering the system interface, we move the mouse cursor to the "  " in the upper right corner, then it will display the IP address of the Raspberry Pi: 192.168.3.157 (the IP address of each Raspberry Pi is different). It is necessary for you to record this IP address for it is needed to log in to the Raspberry Pi system later.

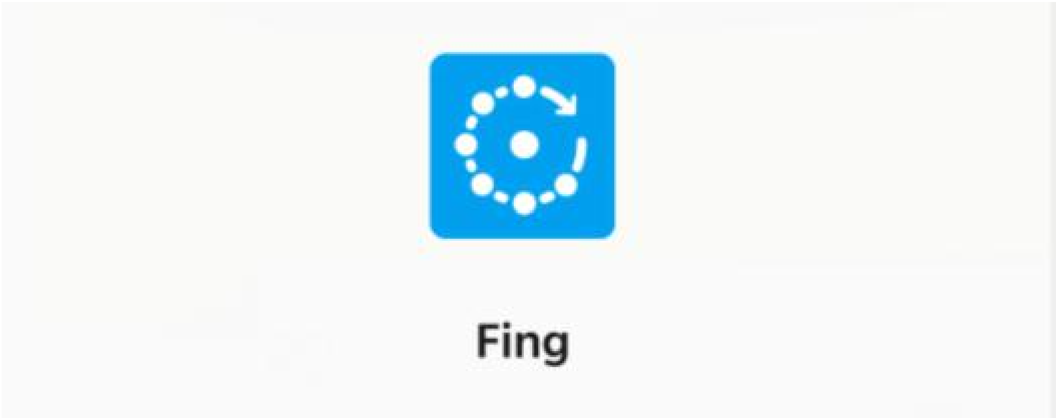
2. You can also check the following IP address by opening the command window of the Raspberry Pi and entering the following command, you need to write it down:

hostname -I

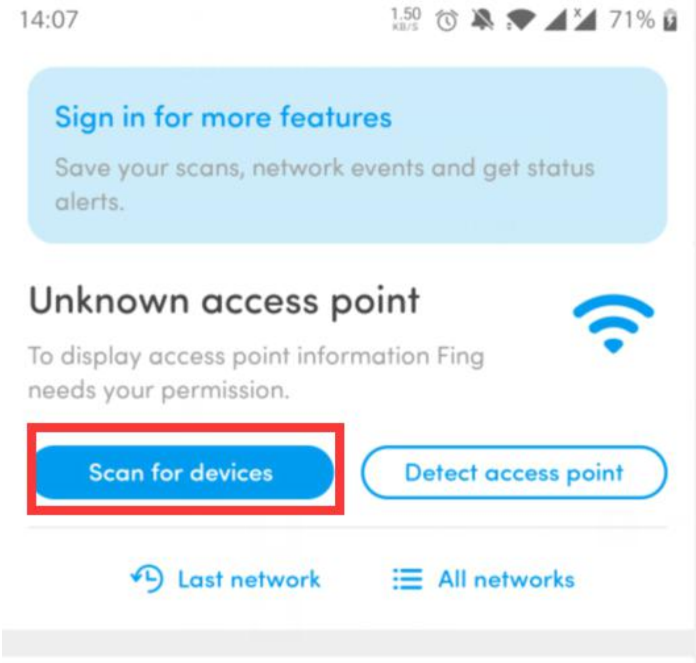


1.6.1.2 Obtaining an IP address with a mobile phone

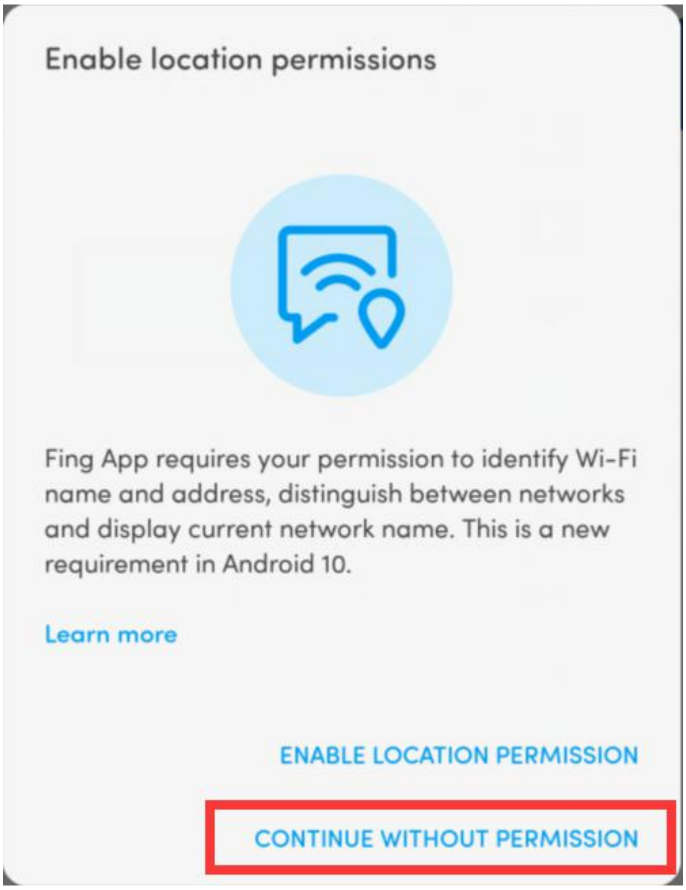
1. You need to download an APP called "Fing" on your phone, as shown below:



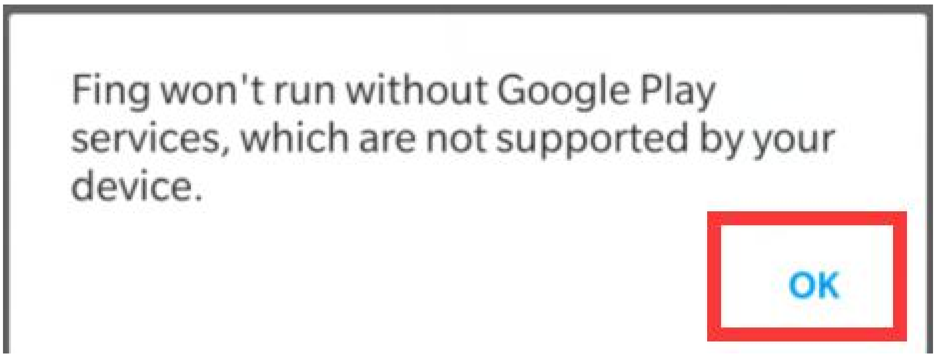
2. After the download is complete, your phone and Raspberry Pi need to be in the same local area network, that is, your phone and Raspberry Pi are connected to the same WIFI, then open "Fing" and click "Scan for devices":



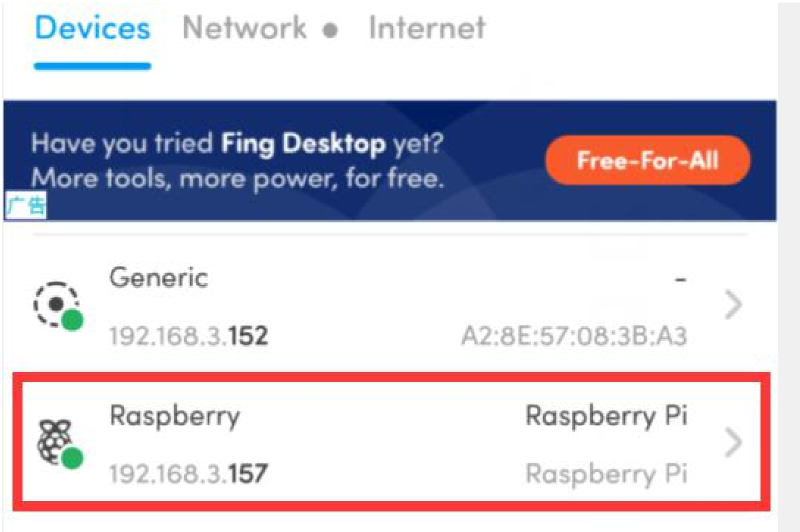
Click“CONTINUE WITHOUT PERMISSION”:



Click OK：



3. Wait for the scan to complete. In the list, you find a device named "Raspberry Pi". In the lower left corner, you will see the IP address of the Raspberry Pi: 192.168.3.157. You need to write down this IP address.



1.6.2 Remotely logging in to the Raspberry Pi system

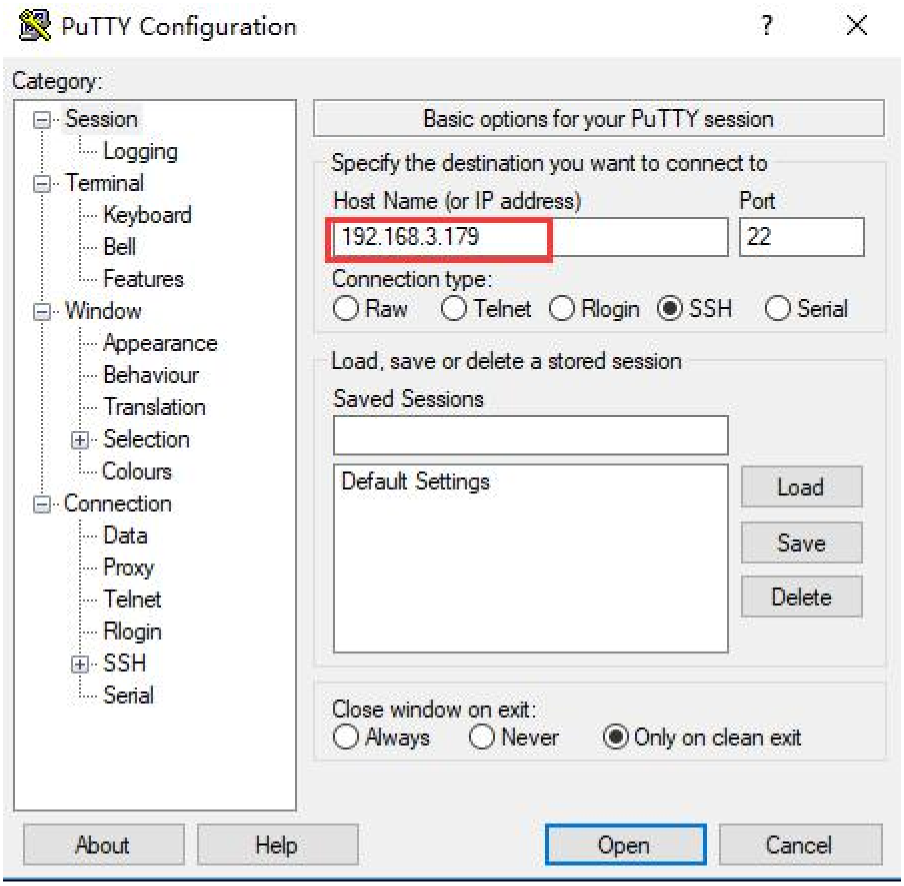
This course recommends two kinds of software for SSH login to Raspberry Pi. In actual use, you only need to download one. LInux or Mac OS comes with SSH function, you can log in to the Raspberry Pi remotely with the terminal without downloading software.

1.6.2.1 Putty

You need to download and install PuTTY corresponding to your computer system version, and use it to log in to the Raspberry Pi. PuTTY download address:

https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html

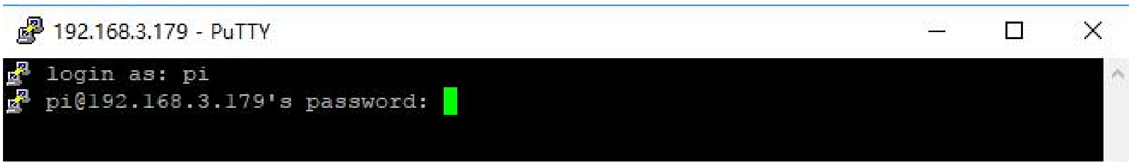
Run PuTTY, enter the IP address of the Raspberry Pi into the Host Name, and click Open.



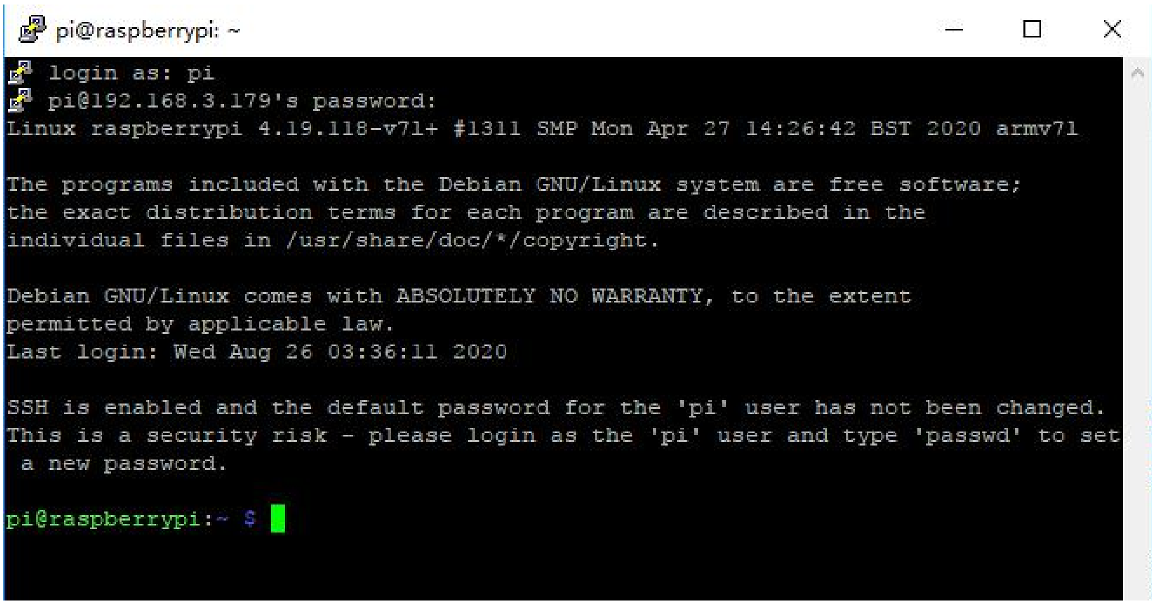
• If it prompts Network error: Connection timed out, it means you probably entered the wrong IP address.

• When the connection is normal, you will see a security warning. You can safely ignore it and click the "Yes" button. You will see this warning when PuTTY connects to a Raspberry Pi that has not been connected before.

• You will now see the usual login prompt. Log in with the same username and password as the Pi itself. The default login name of Raspbian is pi and the password is raspberry. When entering the password, the screen will not display the entered password. After entering raspberry, press Enter to confirm.



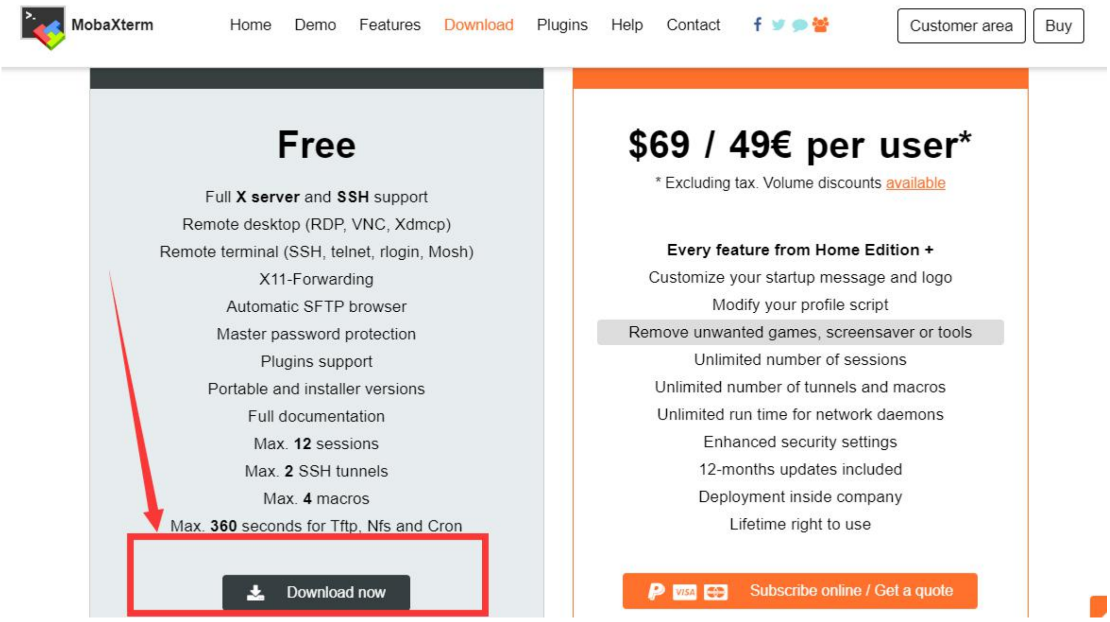
• You should now have the Raspberry Pi prompt, which will be the same as the prompt on the Raspberry Pi itself.



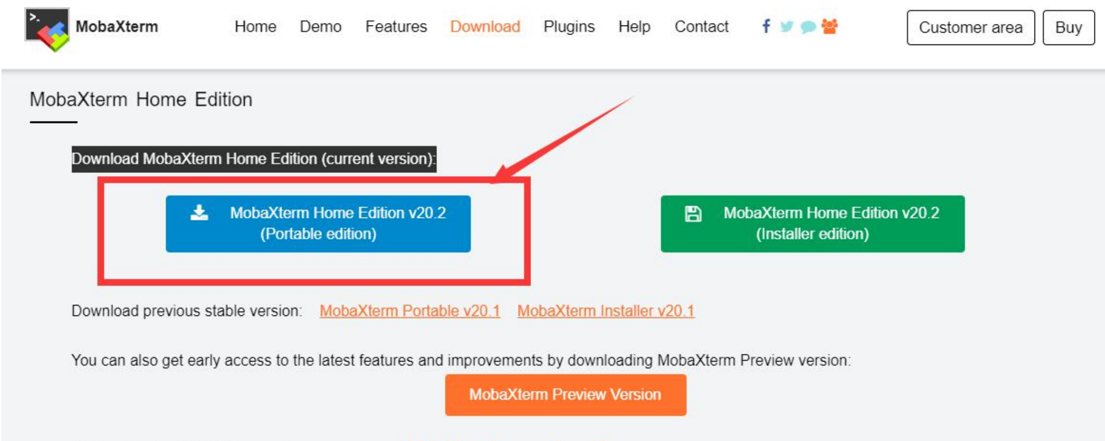
1.6.2.2 MobaXterm

MobaXterm is a terminal tool software that can be used to remotely control the Raspberry Pi.

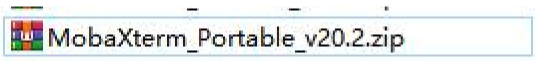
(1) Log in to the official website with a browser to download: https://mobaxterm.mobatek.net/download.html. Choose the Free version to download.

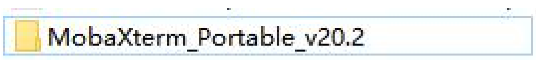


(2) Download the Portable edition of MobaXterm Home Edition (current version):

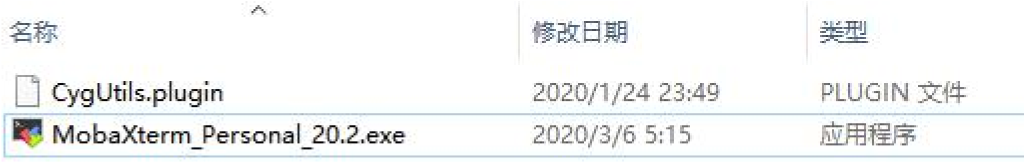


(3) Find the downloaded file MobaXterm_Portable_v20.2.zip, double-click to open it, unzip it to get a new file.

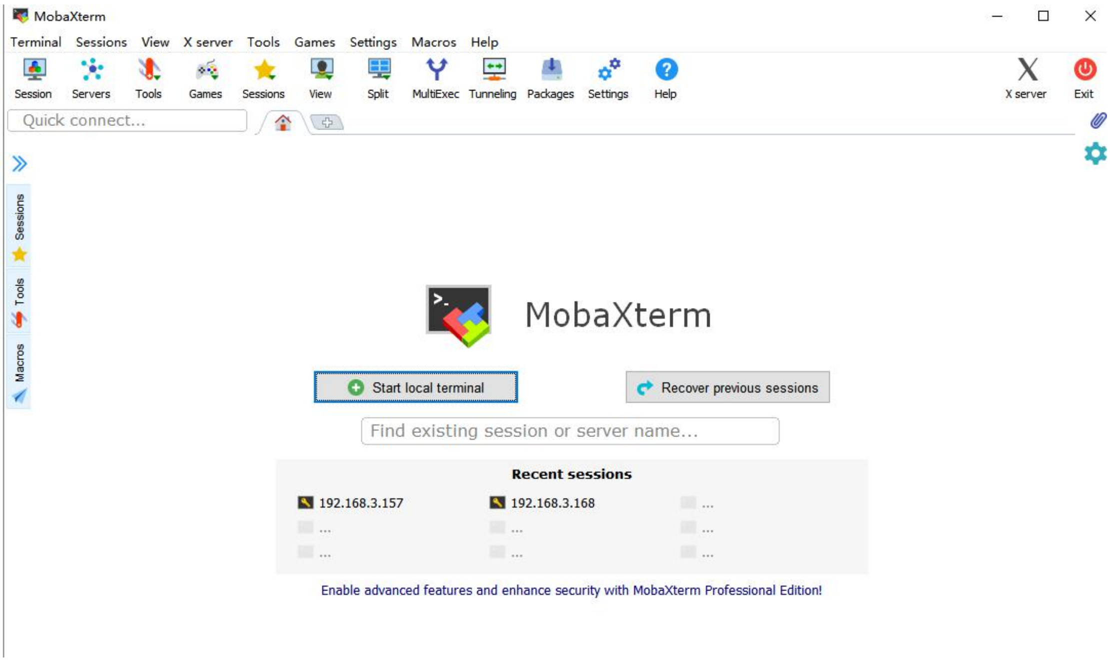




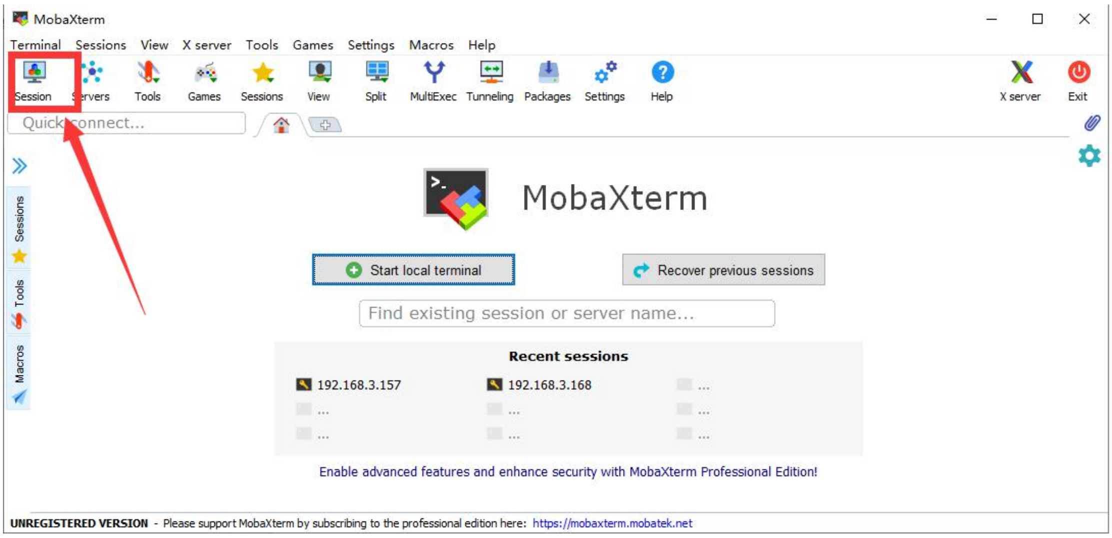
(4) Open the unzipped folder, there is a file MobaXterm_Personal_20.2.exe inside.



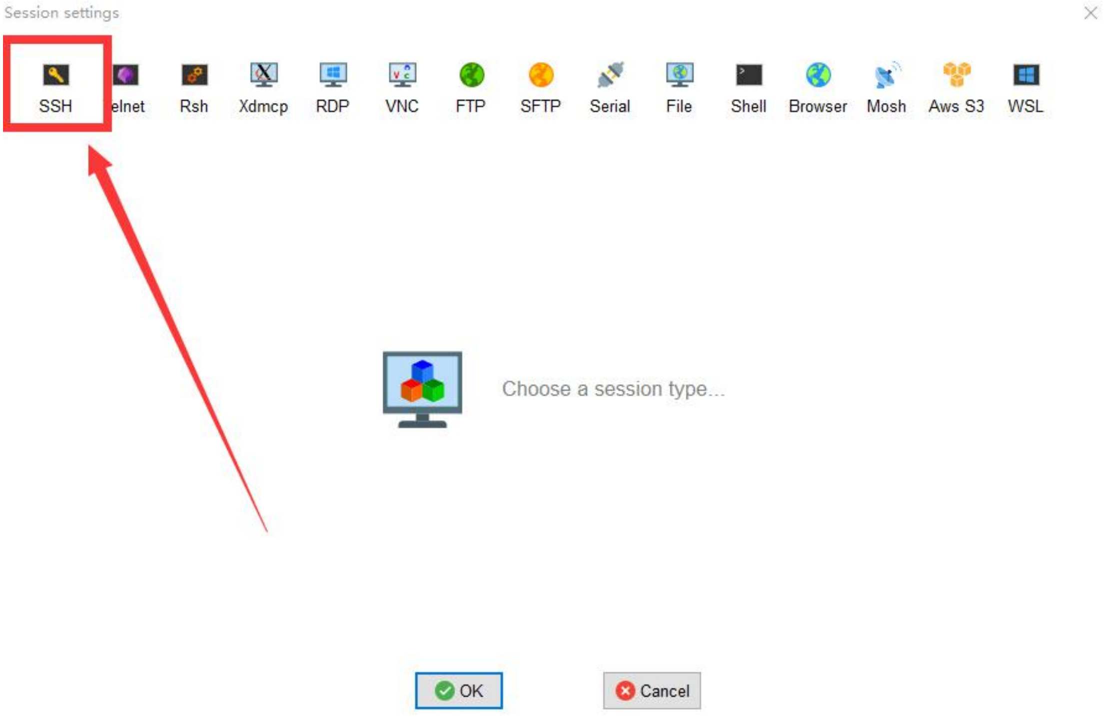
(5) Double-click to open MobaXterm_Personal_20.2.exe, and then directly open the MobaXterm software. The interface is as follows:



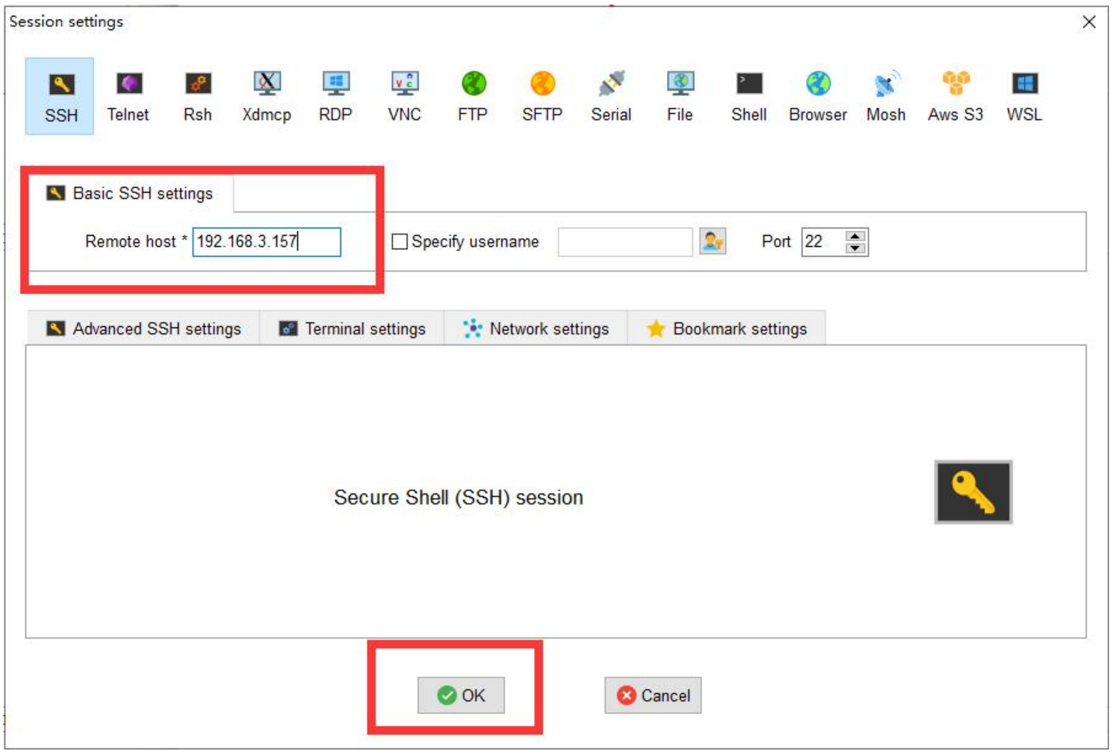
(6) Click "Session" in the upper left corner.



(7) Click "SSH".

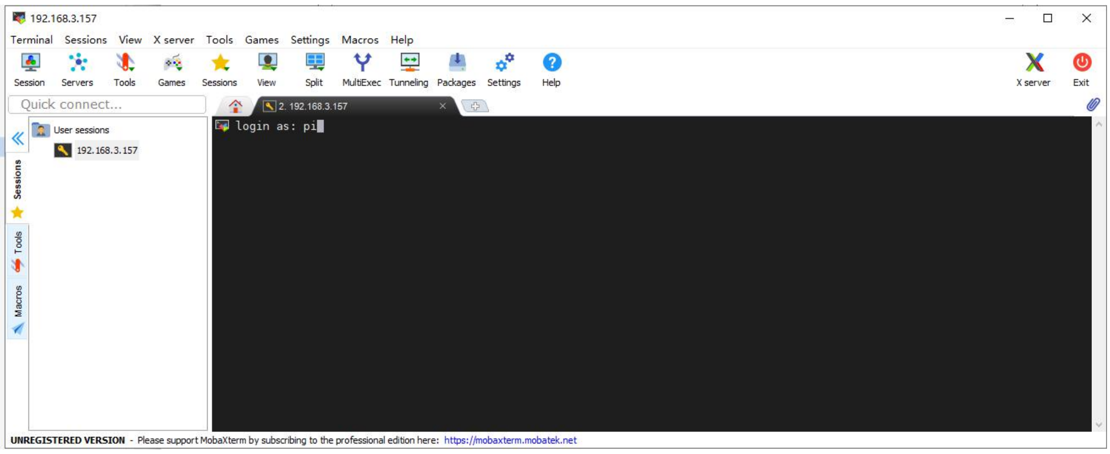


(8) Enter the IP address of the Raspberry Pi queried before: 192.168.3.157, and click "OK" to confirm.

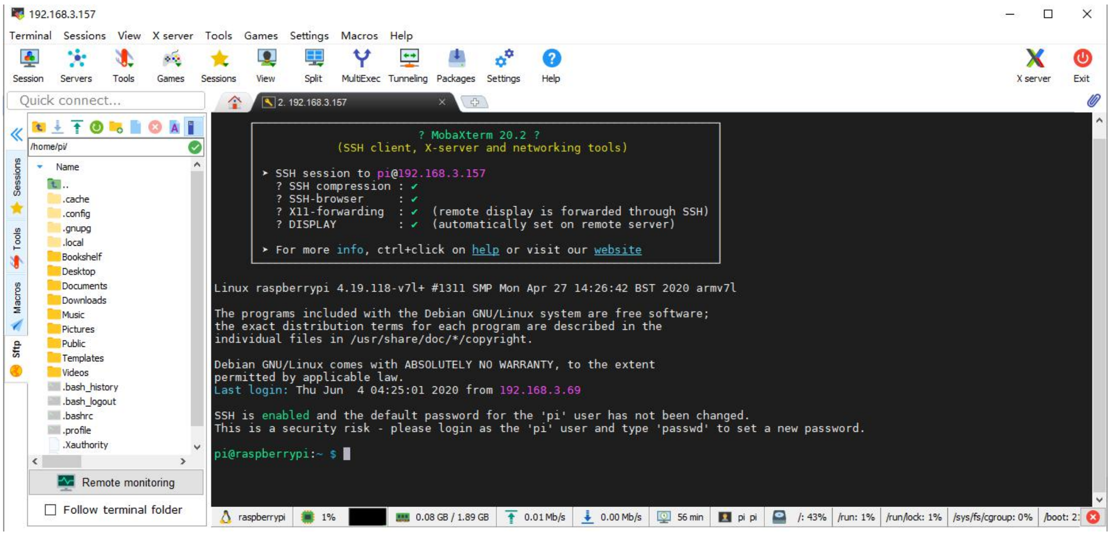


(9) Enter the Raspberry Pi default account: pi, then press the Enter key, and then enter the Raspberry Pi default password: raspberry. Press Enter to log in to the

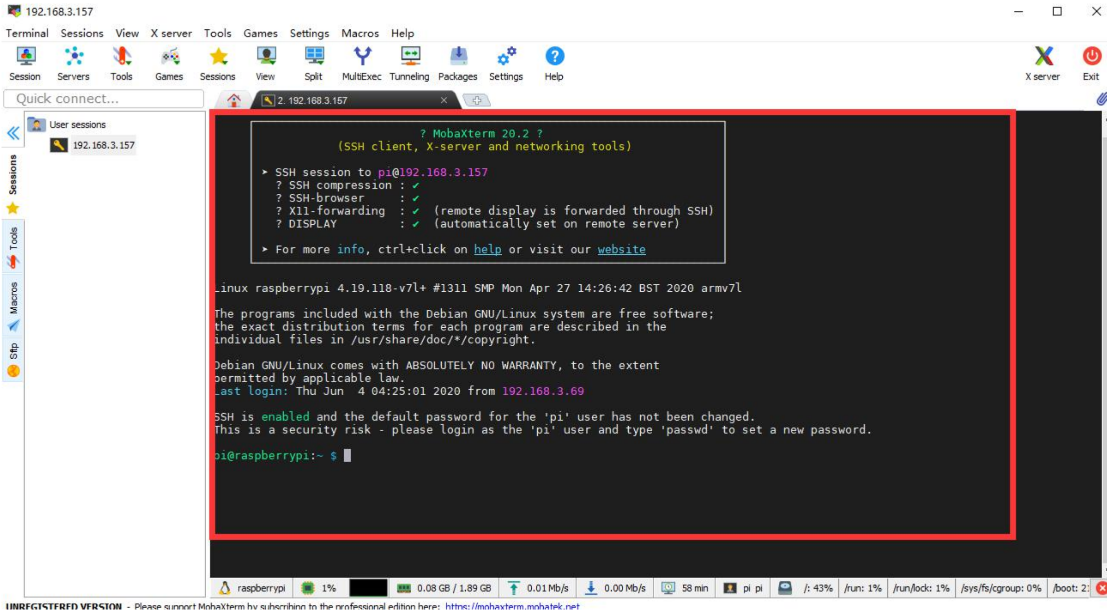
Raspberry Pi system.



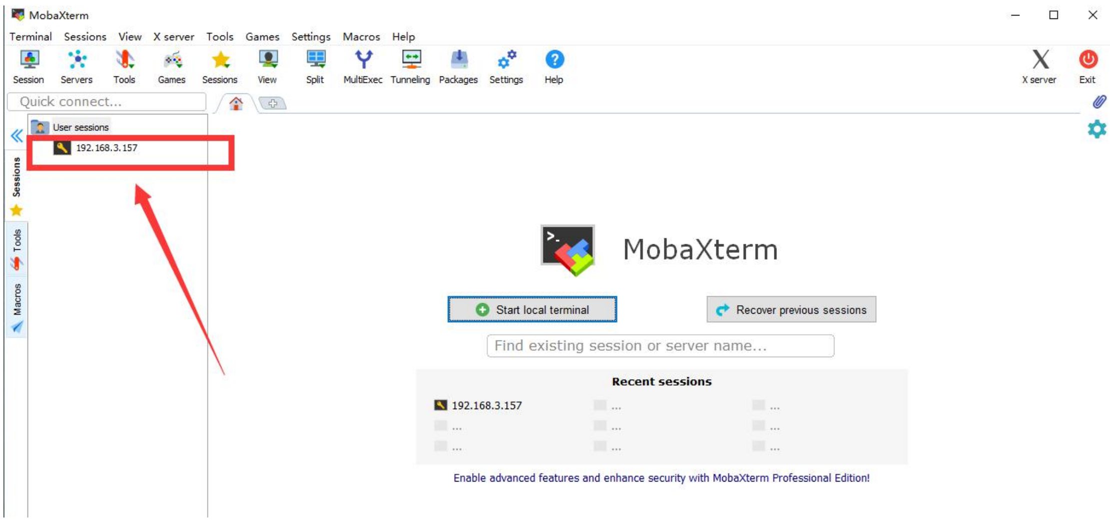
(10) After successfully logging in to the Raspberry Pi system, the following interface will appear:



(11) The red box in the figure below is the command window, where you can control the Raspberry Pi by entering commands.



(12) When we close the MobaXterm software and open MobaXterm again to connect to the Raspberry Pi, we can double-click the IP address under "User sessions" on the left: 192.168.3.157, enter the account name: pi, and you can directly connect to the Raspberry Sent.



1.6.2.3 LInux or Mac OS comes with SSH function, and it is not necessary to download the above two software

Steps to connect to Raspberry Pi via SSH:

1. Open a console terminal window.

2. The initial user name of the Raspberry Pi is pi and the initial password is raspberry.

3. Enter ssh pi@<IP> in the command line and replace <IP> with your Raspberry Pi IP address, as shown in the following example:

ssh pi@192.168.3.157

4. Press Enter, and the prompt Are you sure you want to continue connecting (yes/no)?

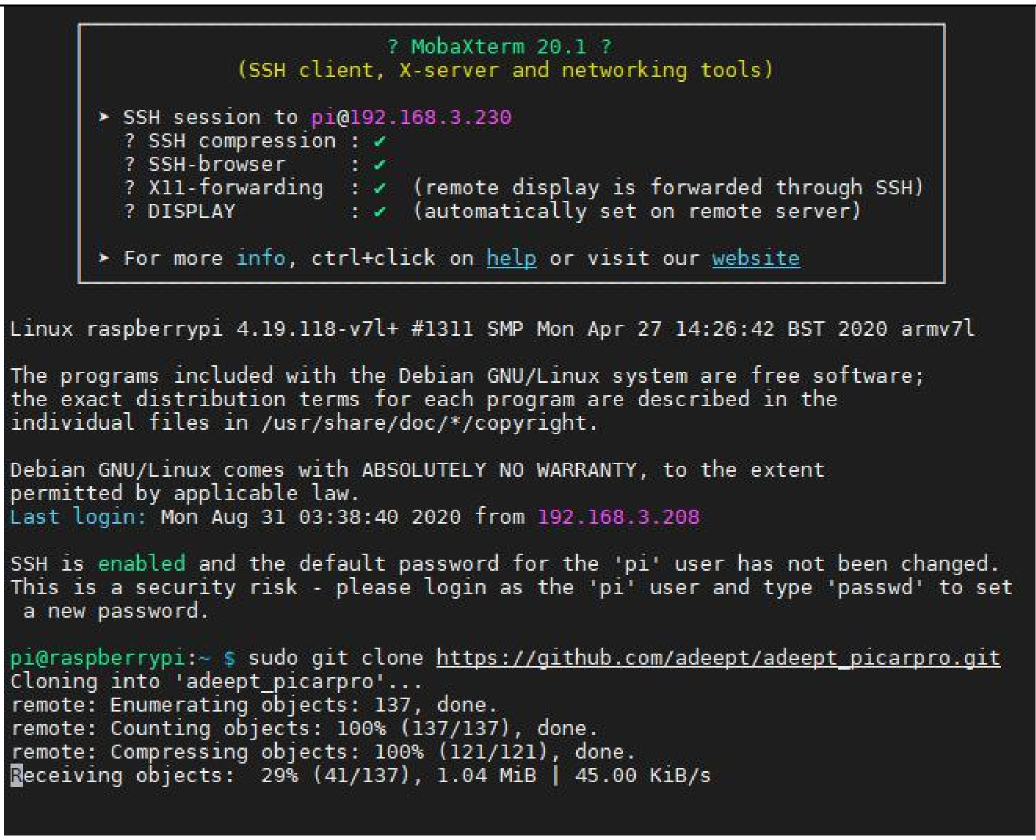
5. Enter yes, press Enter, pi@192.168.3.157's password: appears, fill in the initial password raspberry of the Raspberry Pi, pay attention to the case, there will be no changes on the screen during the password input, but it does not Indicates that the input was not successful, press enter after the input is complete.

1. 6. Now you have logged in to the Raspberry Pi.

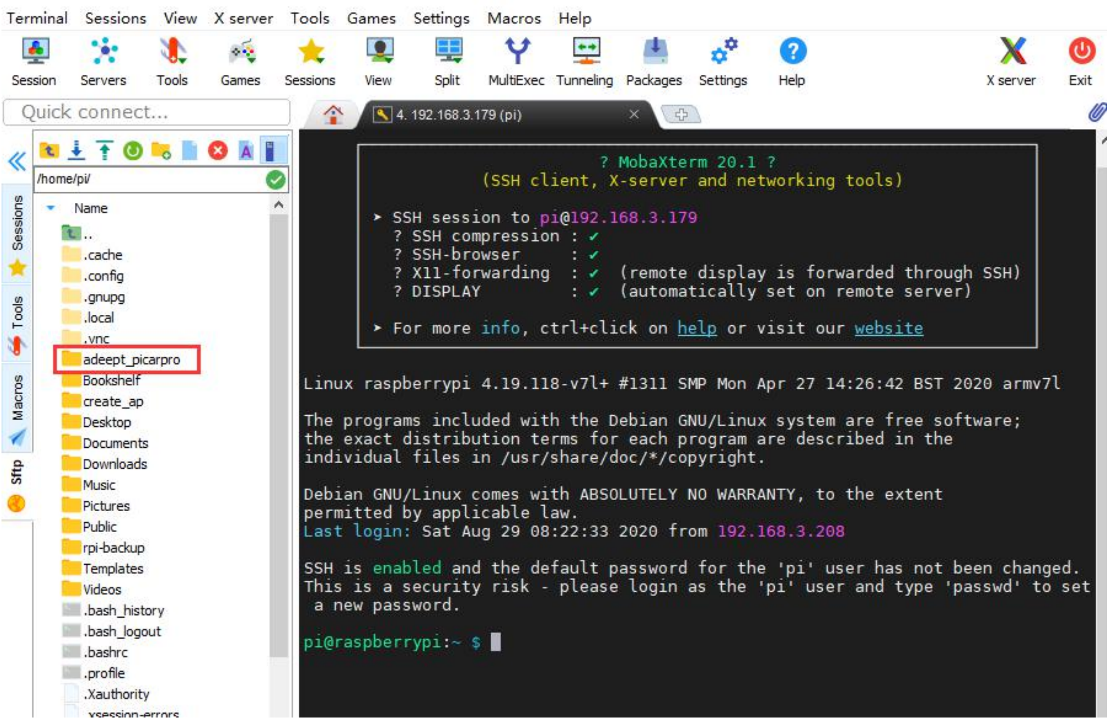
1.7 Downloading the code program to control the robot

After successfully logging in to the Raspberry Pi, we need to download the code program to control the robot. We enter the following command in the console:

sudo git clone https://github.com/adeept/adeept_picarpro.git



After successfully downloading, you will see a new folder adept_picarpro in the file resource list on the left. This folder stores some very important program codes. We will teach you how to use it in detail in the following courses.



1.8 Installing the dependency library of PiCar Pro program

We have prepared a script to install all the dependent libraries that need to be used and set up operations such as turning on the camera and automatically running on startup.

Steps:

1. Enter the following code in the console and run the script setup.py to install the required dependent libraries:

sudo python3 adeept_picarpro/setup.py

```
pi@raspberrypi:~ $ sudo python3 adeept_picarpro/setup.py
```

2. Press Enter to confirm. The following operations are automatically completed by the script program. Depending on the network environment, this process may last for tens of minutes or several hours, just wait patiently.

3. After the installation is complete, the console will show text:

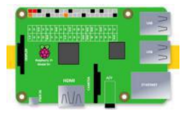
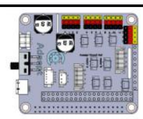
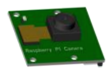
```
The program in Raspberry Pi has been installed, disconnected and restarted.
You can now power off the Raspberry Pi to install the camera and driver board (R
obot HAT).
After turning on again, the Raspberry Pi will automatically run the program to s
et the servos port signal to turn the servos to the middle position, which is co
nvenient for mechanical assembly.
restarting...
```

After the installation is complete, the Raspberry Pi will automatically disconnect the SSH connection and restart. At this time, if you are using a Raspberry Pi connected by software such as Putty, there will be an error message such as Network error: Software caused connection abort, which is normal, just close the window.

Lesson 2 How to Use the Web Controller

In this lesson, we will learn how to control the Raspberry Pi with the web. The web controller is a web interface used to control the robot products to perform various actions. It can be used on PCs, mobile phones, tablets, and any device that can run a browser.

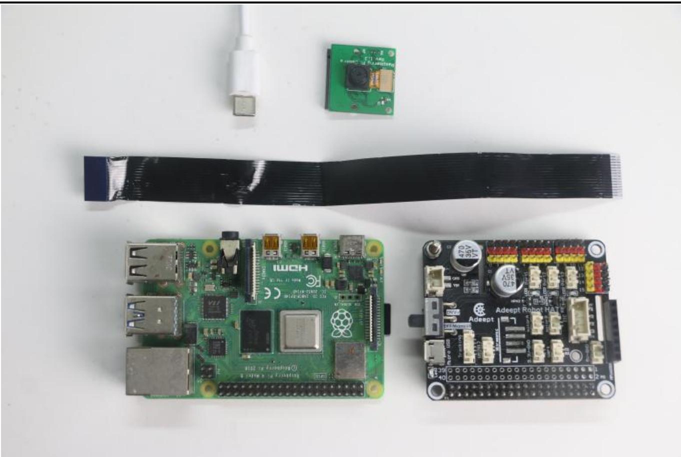
2.1 Components needed for this lesson

Components	Quantity	Picture
Raspberry Pi	1	
Robot HAT	1	
camera	1	
Camera long cable(black)	1	

2.2 Installing components and run web programs

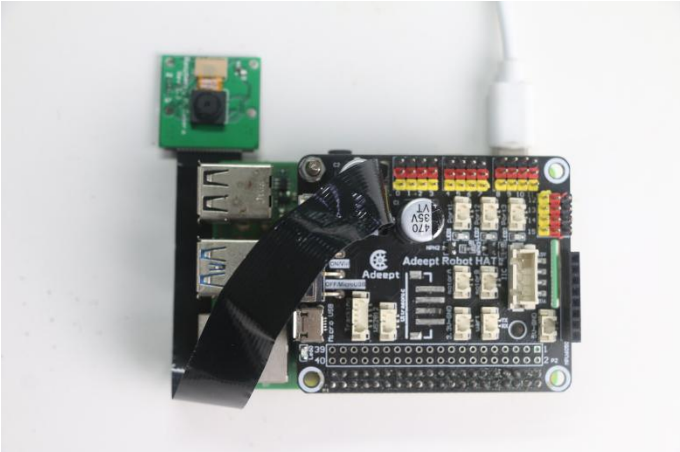
Only by successfully running the webServer.py program on the Raspberry Pi, can the Raspberry Pi be accessed with the IP on the browser.(After installing the dependent libraries, the Raspberry Pi will automatically run webServer.py).

Prepare the components required for installation, and turn off the power of the Raspberry Pi during installation.



Install the camera cable, contact the metal surface of the cable with the metal surface of the Raspberry Pi (the same applies to installing the camera).

Install Robot HAT and camera, and connect the Raspberry Pi power supply.



After the Raspberry Pi is turned on (about 30-50s), you can access the Raspberry Pi with a browser.

2.3 What is a web controller

The web controller is a web interface used to control the robot products to perform various actions. It can be used on PCs, mobile phones, tablets, and any device that can run a browser.

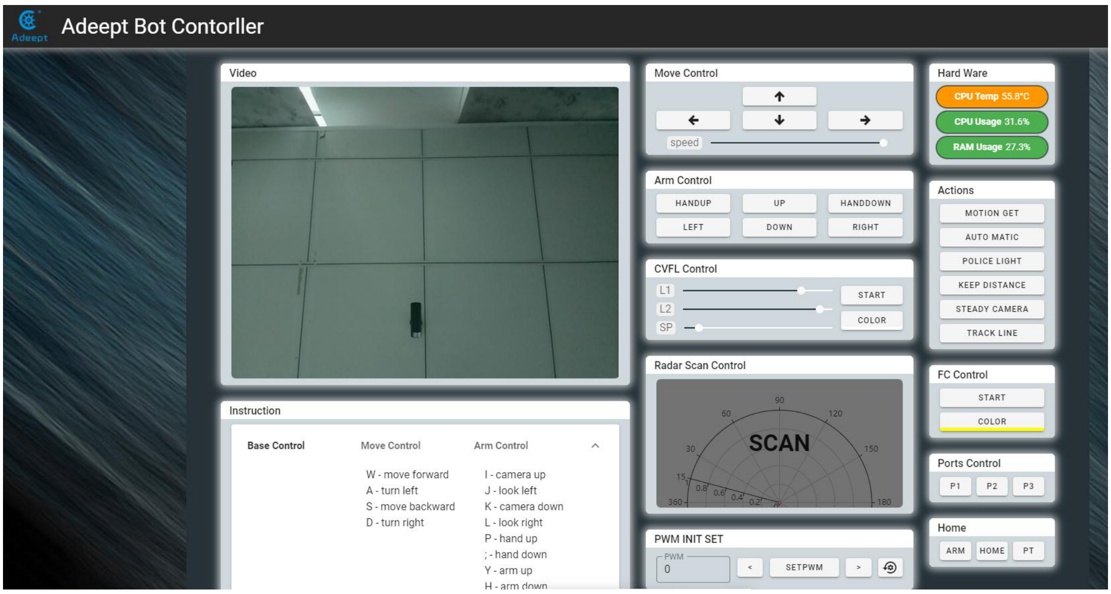
If you install the contents of the software package according to the document, it is very simple to open the web controller.

1. Make sure your device is in the same local area network as the Raspberry Pi.

2. Obtain the IP address of the Raspberry Pi (refer to the software installation section).

3. Open the browser on the device (chrome browser is recommended to avoid possible browser compatibility issues), enter the IP address of your Raspberry Pi in the address bar, and visit port 5000, for example: 192.168.3.44 :5000

The web controller will then be loaded into the browser.



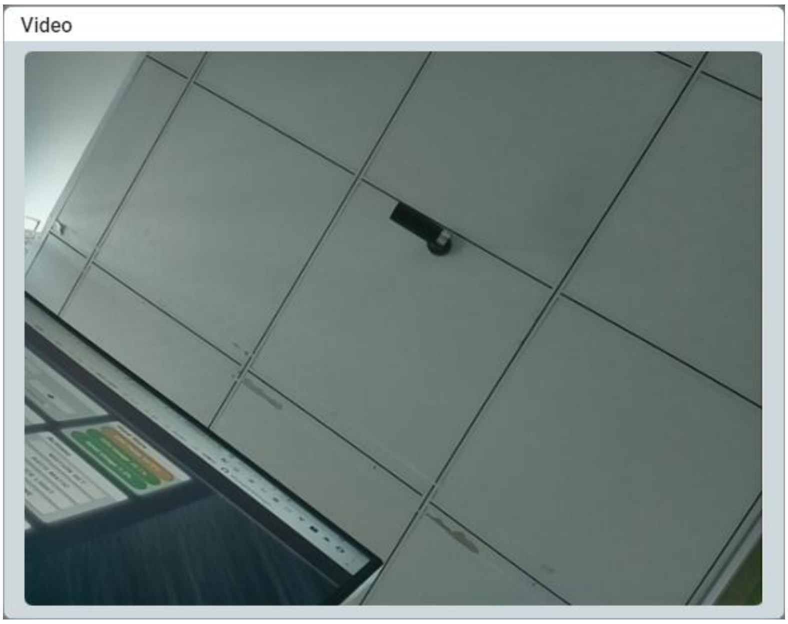
Depending on the product, the modules on the web controller are also different. Below is a list of most of the modules and their usage. You can compare the modules displayed in your browser to understand their functions and usage.

2.4 Basic module

Basic modules can be found in almost all products. These modules are used to control the core functions of robot products. At this time, only the camera can work, and other functions can only be realized after PiCarPro is installed.

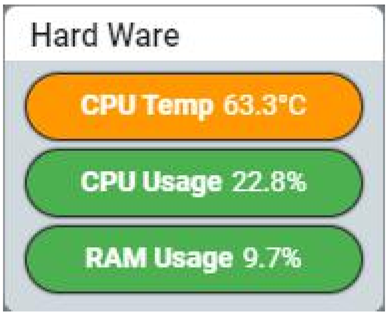
2.4.1 Vedio module

Showing the screen captured by the camera. Depending on the product type, the window rendering method may be different, and some products can also interact with this window.



2.4.2 Hard wave module

Displaying the hardware information of the robot product.



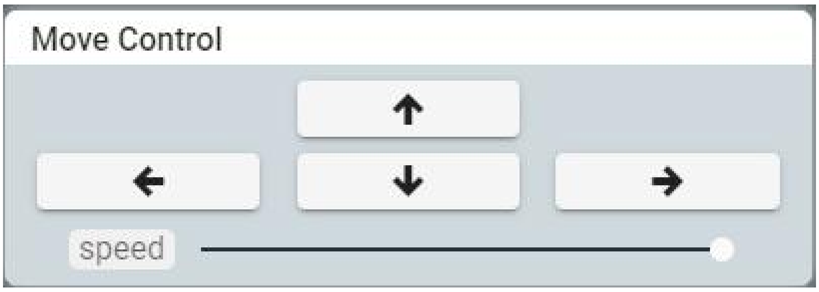
CPU Temp: Displaying the Raspberry Pi CPU temperature;

CPU Usage: Displaying the CPU occupancy rate of the Raspberry Pi;

RAM Usage: Displaying the RAM usage of Raspberry Pi

2.4.3 Move Control Module

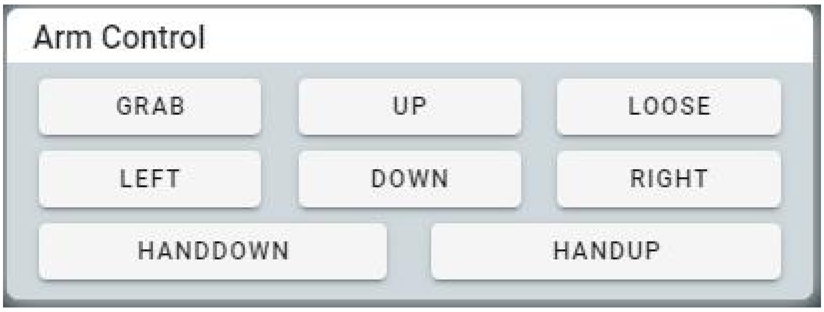
Controlling the movement of the robot product back and forth.



speed: Using the slider to control the speed when the robot moves.

2.4.4 Arm Control Module

Controlling PTZ and robotic arm. Depending on the product, the number of buttons and manipulation methods are also different.



UP: The camera angle of view moves upward

DOWN: the camera angle of view moves downward

GRAB: clamping chuck

LOOSE: Loosening the chuck

LEFT: The robotic arm turns to the left

RIGHT: The robotic arm turns to the right

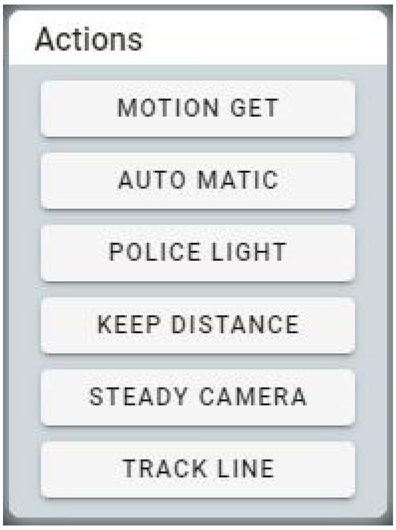
HANDDOWN: The robotic arm rotates downward

HANDUP: The robotic arm rotates upward

2.5 Advanced Function Module

Advanced function modules refer to those modules used to perform advanced functions of robot products. Also called function button group. Used to switch the function of the robot.

2.5.1 Actions Module



MOTION GET: switching watchdog mode. In this mode, the robot product stops moving and reacts to the moving objects detected in the camera, and the moving objects are framed in the video of the Vedio module.

AUTO MATIC: switching automatic obstacle avoidance mode. In this mode, the robot product will automatically advance and use the ultrasonic module to detect obstacles. When encountering obstacles, try to find other ways.

POLICE LIGHT: switching the police light mode. In this mode, the LED lights of the robot product will flash like a warning light.

STEADY CAMERA: switching camera stabilization mode. In this mode, the robot product will try to maintain the vertical stability of the camera.

TRACK LINE: switching hunting mode. In this mode, the robot product will try to move along the white lines on the black ground.

2.5.2 FC Control Module

Control the switch of the color tracking function and the setting of the color to be tracked.

COLROR: choosing the color you want to track.

START: switching color tracking mode.

2.5.3 CVFL Control Module

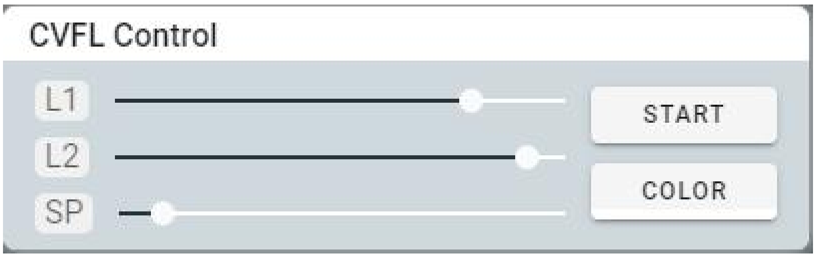
The switch that controls the visual patrol function.

L1, L2: the two parameters L1 and L2 are used to set the height of the two auxiliary lines. Robot products will only recognize the lines in the screen surrounded by two auxiliary lines.

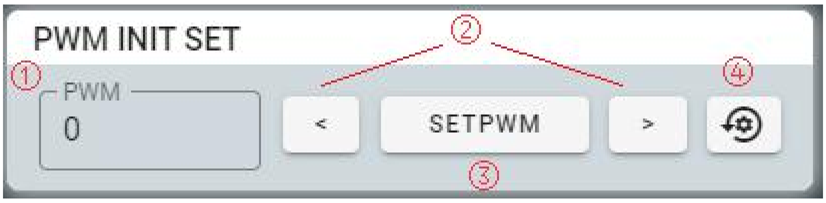
SP: deflection tolerance. The larger the value set, the more the robot tends to go straight.

COLOR: switching to search for white lines on black or black lines on white.

START: switching the visual tracking function.



2.5.4 PWM INIT SET module



It is used to fine-tune the default angle of the servo, and can correct the angle error generated when the servo is installed. When returning to the center, the rudder will return to this default angle.

①Enter the number of the PWM port connected to the servo you want to fine-tune in the PWM input box.

②Click these two buttons to make the servo rotate slightly clockwise or counterclockwise.

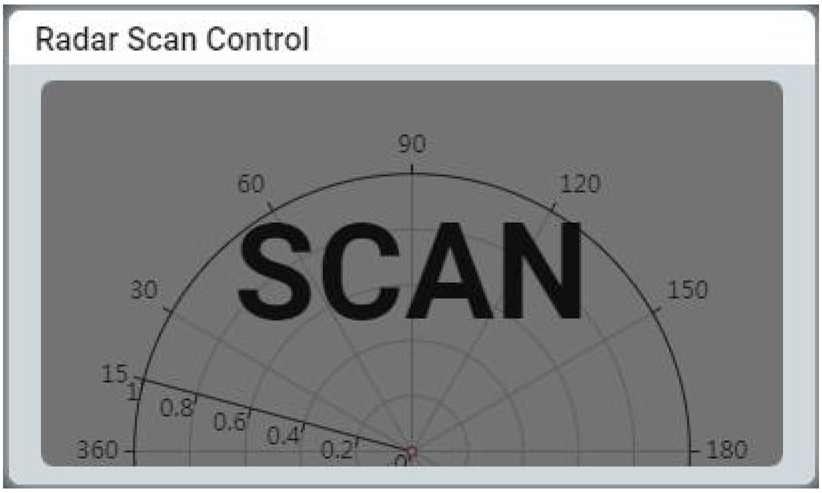
③Click this button to save the current servo angle as the default angle.

④ Click this button to initialize the default angle of all servos to factory settings.

2.5.5 Home Module

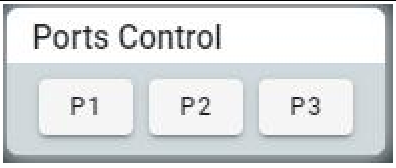
HOME: Returning all servos to their initial positions.

2.5.6 Radar Scan Control Module



Used to perform the ultrasound scan function and display the scan results.

2.5.7 Ports Control Module

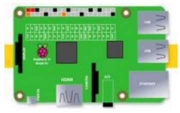
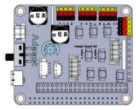


Controlling the switches of Port1, Port2, Port3 on the development board.

Lesson 3 How to Control LED

In this lesson, we will learn how to control LED with Raspberry Pi.

3.1 Components used in this course

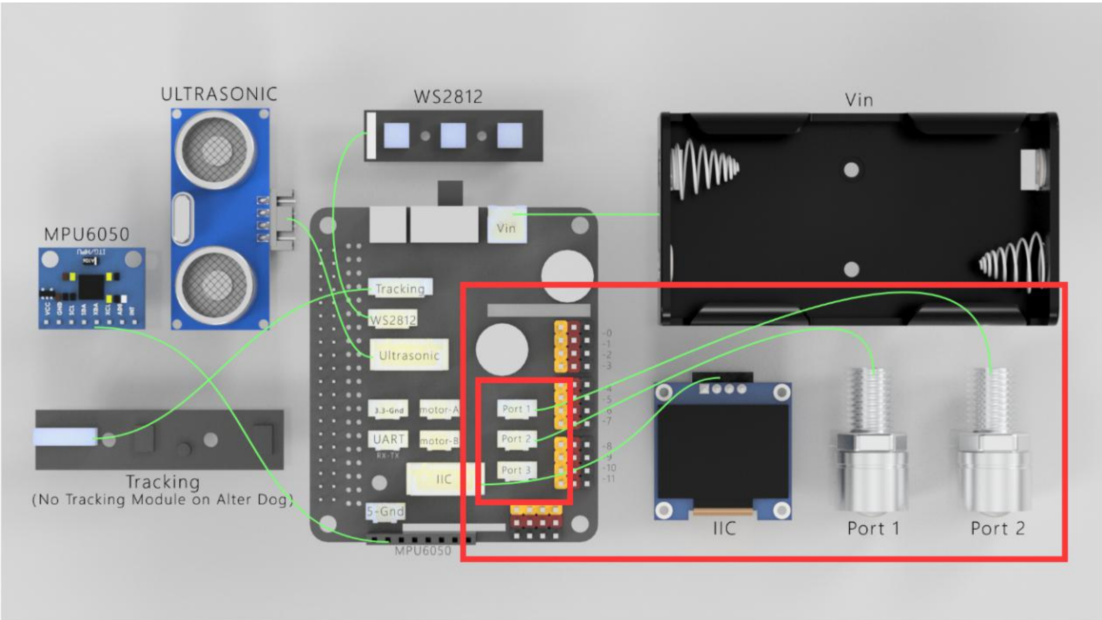
Components	Quantity	Picture
Raspberry Pi	1	
Robot HAT	1	
Small LED light	1	

3.2 Introduction of small warm color LED

The small warm color LED is a LED that can emit warm white light after being lit. It has a working power of 0.5W, a working voltage of 5V/12V, and a light-emitting angle of 60°. The red line represents the positive pole and the black line represents the negative pole. The small LED needs to be connected to Port1, Port2, and Port3 on the Robot HAT driver board to use

3.3 Wiring diagram (Circuit diagram)

When the small warm color LED is used, it needs to be connected to the Port1, Port2, and Port3 on the Robot HAT driver board, as shown in the red box:



3.4 How to control the small warm color LED

3.4.1Run the program of this course

1. Log in to your Raspberry Pi via SSH (the method to log in to the Raspberry Pi has been introduced in Lesson 1):

```
Linux raspberrypi 4.19.118-v7l+ #1311 SMP Mon Apr 27 14:26:42 BST 2020 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: Sat Aug 29 08:17:49 2020 from 192.168.3.208

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:~ $
```

2. Run the command to enter the picarpro/server folder. This folder stores the sample code program for controlling the robot. Enter the following command and press Enter:

cd adeept_picarpro/server

```
pi@raspberrypi:~ $ cd adeept_picarpro/server/
pi@raspberrypi:~/adeept_picarpro/server $
```

3. Enter the command to view the contents of the current directory:

ls

```
pi@raspberrypi:~ $ cd adeept_picarpro/server/
pi@raspberrypi:~/adeept_picarpro/server $ ls
app.py             findline.py    instruction.txt   OLED.py         servo.py
appserver.py       FPV.py         Kalman_filter.py  PID.py          switch.py
base_camera.py     FPVtest.py     LEDapp.py         __pycache__     ultra.py
camera_opencv.py   functions.py   LED.py            robotLight.py   webServer.py
dist               info.py        move.py           RPIservo.py
pi@raspberrypi:~/adeept_picarpro/server $
```

4. switch.py is a python program, you can run this program on the Raspberry Pi by directly typing the following commands:

sudo python3 switch.py

```
pi@raspberrypi:~/adeept_picarpro/server $ sudo python3 switch.py
Light on...
Light off...
```

5. After successfully running the program, you will observe that the small warm LED light will be on for 1 second and be off for 1 second. When you want to terminate the running program, you can press the shortcut key "Ctrl + C" on the keyboard.

3.4.2 Viewing the main code program of this lesson

Enter the program to edit the code, we provide two solutions:

Option One:

1. By running the command, press Enter and enter the edit:

sudo nano switch.py

```
pi@raspberrypi:~/adeept_picarpro/server $ sudo nano switch.py
```

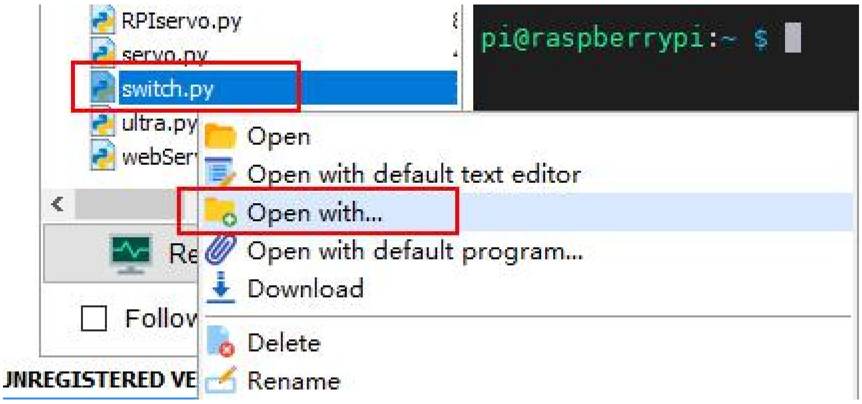
```
  GNU nano 3.2                          switch.py

#!/usr/bin/env/python
# File name   : switch.py
# Production  : HAT
# Website     : www.gewbot.com
# Author      : William
# Date        : 2018/08/22

import RPi.GPIO as GPIO

def switchSetup():
    GPIO.setwarnings(False)
    GPIO.setmode(GPIO.BCM)
    GPIO.setup(5, GPIO.OUT)
    GPIO.setup(6, GPIO.OUT)
    GPIO.setup(13, GPIO.OUT)

def switch(port, status):
    if port == 1:
        if status == 1:
            GPIO.output(5, GPIO.HIGH)
        elif status == 0:
            GPIO.output(5,GPIO.LOW)
        else:
            pass
    elif port == 2:
        if status == 1:
            GPIO.output(6, GPIO.HIGH)
        elif status == 0:
            GPIO.output(6,GPIO.LOW)
        else:
            pass
    elif port == 3:
                              [ Read 57 lines ]
^G Get Help   ^O Write Out  ^W Where Is   ^K Cut Text   ^J Justify    ^C Cur Pos
^X Exit       ^R Read File  ^\ Replace    ^U Uncut Text ^T To Spell   ^_ Go To Line
```

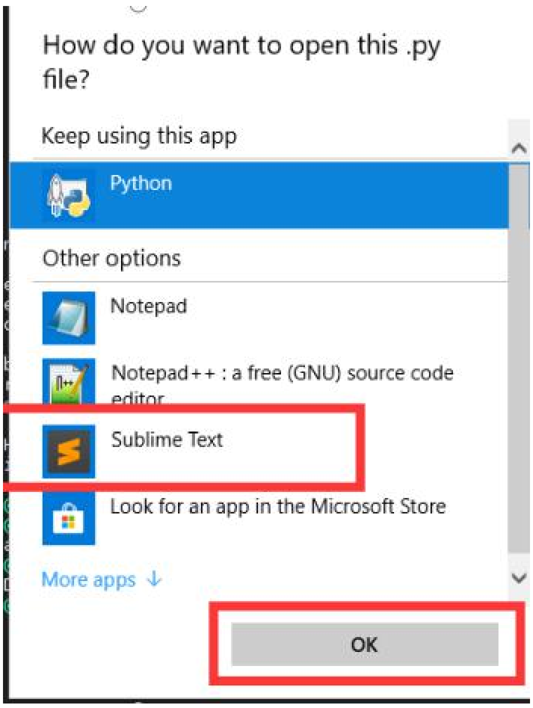
2. Move the cursor by using the up, down, left, and right keys. After editing the code, press "Ctrl + X" --> "Y" --> "Enter" to save the code. If you have not edited the code, just press "Ctrl + X" to exit.

Option two:

1. To install a third-party IDE tool, you can choose Sublime Text / Visual Studio Code / Atom, etc.

1. 2. Log in to the Raspberry Pi through MobaXterm software, you can find the file you want to edit in the file resource management system on the left side of MobaXterm, right-click on the file, click Open with, select your IDE, and edit in the IDE After finishing, save the file by "Ctrl+S".


RPIservo.py
servo.py
switch.py
ultra.py
pi@raspberrypi:~ $
Open
Open with default text editor
Open with...
Open with default program...
Download
Delete
Rename


How do you want to open this .py file?
Keep using this app
Python
Other options
Notepad
Notepad++ : a free (GNU) source code editor
Sublime Text
Look for an app in the Microsoft Store
More apps
OK

```python
#!/usr/bin/env/python
# File name   : switch.py
# Production  : HAT
# Website     : www.gewbot.com
# Author      : William
# Date        : 2018/08/22

import RPi.GPIO as GPIO

def switchSetup():
    GPIO.setwarnings(False)
    GPIO.setmode(GPIO.BCM)
    GPIO.setup(5, GPIO.OUT)
    GPIO.setup(6, GPIO.OUT)
    GPIO.setup(13, GPIO.OUT)

def switch(port, status):
    if port == 1:
        if status == 1:
            GPIO.output(5, GPIO.HIGH)
        elif status == 0:
            GPIO.output(5,GPIO.LOW)
        else:
            pass
    elif port == 2:
        if status == 1:
            GPIO.output(6, GPIO.HIGH)
        elif status == 0:
            GPIO.output(6,GPIO.LOW)
        else:
            pass
    elif port == 3:
        if status == 1:
            GPIO.output(13, GPIO.HIGH)
        elif status == 0:
            GPIO.output(13,GPIO.LOW)
        else:
            pass
    else:
        print('Wrong Command: Example--switch(3, 1)->to switch on port3')

def set_all_switch_off():
    switch(1,0)
    switch(2,0)
    switch(3,0)

```

3. When you need to save the modified file, you can press the shortcut key "Ctrl+S", then select "Yes". If there is a "Permission denied" error, just run the command in MobaXterm:

sudo chmod -R 777 adeept_picarpro/server

```
pi@raspberrypi:~ $ sudo chmod -R 777 picarpro/server
pi@raspberrypi:~ $
```

3.4.3 Main code program

Complete code reference file switch.py

1. First import the library used to control the Raspberry Pi GPIO, and instantiate it as a GPIO while importing it. Import the time library for code delay.

```python
import RPi.GPIO as GPIO
```

```python
import time
```

2. The switchSetup function sets the GPIO pin numbers of the Raspberry Pi corresponding to the interface to 5, 6, and 13, here we use BCM coding.

```python
def switchSetup():
    GPIO.setwarnings(False)
    GPIO.setmode(GPIO.BCM)
    GPIO.setup(5, GPIO.OUT)
    GPIO.setup(6, GPIO.OUT)
    GPIO.setup(13, GPIO.OUT)
```

3. The switch function sets the high and low levels of the interface. Port represents ports 1-3. When the status is 0, the corresponding interface light is off, and when the status is 1, the corresponding interface light is on.

```python
def switch(port, status):
    if port == 1:
        if status == 1:
            GPIO.output(5, GPIO.HIGH)
        elif status == 0:
            GPIO.output(5,GPIO.LOW)
        else:
            pass
    elif port == 2:
        if status == 1:
            GPIO.output(6, GPIO.HIGH)
        elif status == 0:
            GPIO.output(6,GPIO.LOW)
        else:
            pass
    elif port == 3:
        if status == 1:
            GPIO.output(13, GPIO.HIGH)
        elif status == 0:
            GPIO.output(13,GPIO.LOW)
        else:
            pass
    else:
        print('Wrong Command: Example--switch(3, 1)->to switch on port3')
```

4. The set_all_switch_off function sets the level of all interfaces to low level, that is, turns off the lights of all interfaces.

```
def set_all_switch_off():
    switch(1,0)
    switch(2,0)
    switch(3,0)
```

5. Instantiating the object and executing the method function **while 1** means always looping, switchSetup、switch、set_all_switch_off means calling the above function. This code keeps the light on for 1 second and then be off for 1 second.

```
if __name__ == "__main__":
    switchSetup()
    while 1:
        switch(1,1)
        switch(2,1)
        switch(3,1)
        print（"Light on...."）
        time.sleep(1)
        set_all_switch_off()
        print("Light off....")
        time.sleep(1)
```

3.5 How to control the LED on Robot HAT driver board

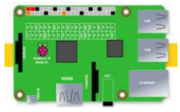
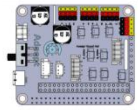
When you study the content of this course, even if you do not use the Small LED module, you can also learn this lesson by controlling the LED on the Robot HAT driver board,because there is a green LED next to each interface: when the interface is turned on, the LED next to the interface will light up, indicating that the interface is on.

You can observe the status of the LED on the Robot HAT driver board by using the method of "3.4 How to control the small warm color LED " above.

Lesson 4 How to Control 180° Servo

In this lesson, we will learn how to control 180° Servo.

4.1 Components used in this course

Components	Quantity	Picture
Raspberry Pi	1	
Robot HAT	1	
180°Servo	1	

4.2 Introduction of 180° Servo

What is a servo?

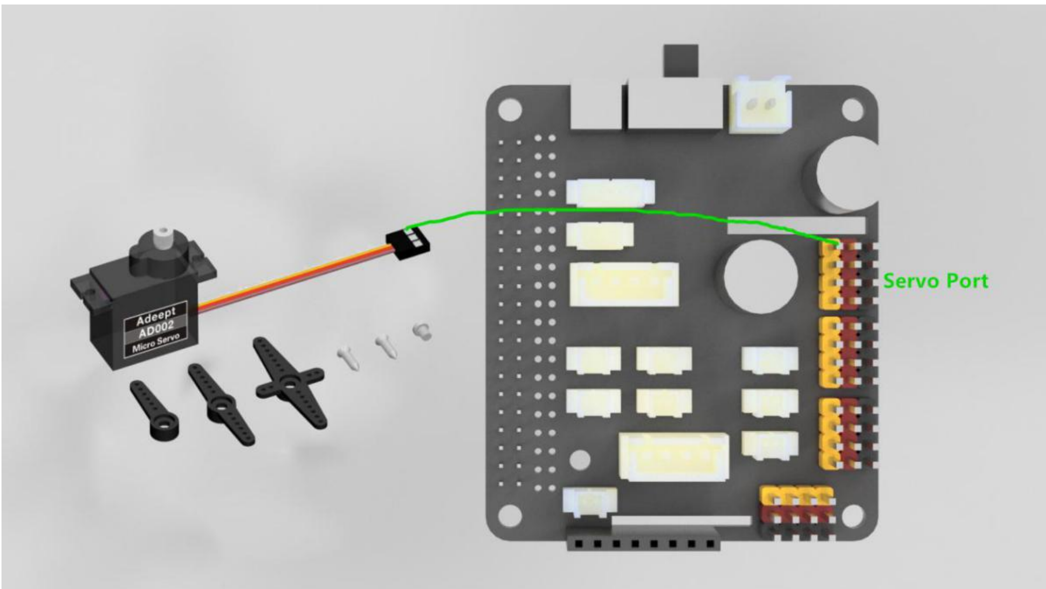
The servo is a position (angle) servo driver, which is suitable for those control systems that require constant angle changes and can be maintained. It has been widely used in high-end remote control toys, such as airplanes, submarine models, and remote control robots.

We use a 180° servo in this lesson, which can move between 0° and 180°.Since the 180° servo can use the PWM signal to control the rotation angle of a certain mechanism, it is a more commonly used module in robot products.

On the Raspberry Pi driver board Robot HAT, there is a PCA9685 chip specially used to control the servo. The Raspberry Pi uses I2C to communicate with the PCA9685. It controls the servo by sending pulse signals from the microcontroller. These pulses tell the servo mechanism of the servo where to move. The picture of the 180° servo is as follows:

4.3 Wiring diagram (Circuit diagram)

When the 180°Servo module is in use, it needs to be connected to the servo interface on the RobotHAT driver board. The yellow wire is connected to the yellow pin, the red wire is connected to the red pin, and the brown wire is connected to black pin,as shown below:



4.4 How to control 180°Servo

4.4.1Run the code

1. Remotely log in to the Raspberry Pi terminal.

```
Linux raspberrypi 4.19.118-v7l+ #1311 SMP Mon Apr 27 14:26:42 BST 2020 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: Sat Aug 29 08:17:49 2020 from 192.168.3.208

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:~ $
```

2. Enter the command and press Enter to enter the folder where the program is located:

cd adeept_picarpro/server/

```
pi@raspberrypi:~ $ cd adeept_picarpro/server/
pi@raspberrypi:~/adeept_picarpro/server $
```

3. View the contents of the current directory file:

ls

```
pi@raspberrypi:~/adeept_picarpro/server $ ls
app.py             findline.py    instruction.txt   OLED.py        servo.py
appserver.py       FPV.py         Kalman_filter.py  PID.py         switch.py
base_camera.py     FPVtest.py     LEDapp.py         __pycache__    ultra.py
camera_opencv.py   functions.py   LED.py            robotLight.py  webServer.py
dist               info.py        move.py           RPIservo.py
pi@raspberrypi:~/adeept_picarpro/server $
```

4. Enter the command and press Enter to run the program:

sudo python3 servo.py

```
pi@raspberrypi:~/adeept_picarpro/server $ sudo python3 servo.py
```

5. After running the program successfully, you will observe that the servo will rotate regularly.

6. When you want to terminate the running program, you can press the shortcut key "Ctrl + C" on the keyboard.

4.4.2 The main code program of this lesson

After the above hands-on practice, you already know how to use and run our course sample code program. Then you must be curious to know how our code program is programmed on the Raspberry Pi to control the 180° servo. Let's learn about the main code program together.

```
import Adafruit_PCA9685
import time
```

Import dependent libraries.

```
pwm = Adafruit_PCA9685.PCA9685()
pwm.set_pwm_freq(50)
```

set_pwm_freq(50) is used to set the PWM frequency to 50Hz. This setting depends on the model of the servo. The servos used in our robot products need to be controlled by a 50Hz PWM signal. If you use other servos, this A value needs to be set by referring to the specific servo document.

```
while 1:
    for i in range(0, 100):
        pwm.sety_pwm(3, 0, (300+i))
        time.sleep(0.05)
    for i in range(0,100):
        pwm.sety_pwm(3, 0, (400-i))
        time.sleep(0.05)
```

pwm.set_pwm(3, 0, 300+i) This method is used to control the rotation of a servo to a certain position, where 3 is the port number of the servo, which corresponds to the number marked on the Robot HAT drive board, but pay attention to the direction of the ground wire, VCC and signal wire when the servo is connected to the drive board. Don’t plug in the opposite direction. Brown to black, red to red, and yellow to yellow; 0 is the deviation value that controls the rotation of the servo, 300+i is the PWM duty cycle value to be set depends on the different servos. This value represents the angle of the servo. The PWM duty cycle range of the servos we use is about 100 to 560, corresponding to about 0° to 180° Rotation range.

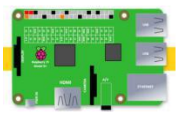
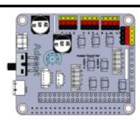
By the formula: (duty ratio -100)/2.55, the angle corresponding to the current duty ratio can be calculated.

Using the above code can make the servo rotate slowly back and forth between 300 and 400, but this method of controlling the servo also has great drawbacks. When the program is executed to the slow motion part of the servo, it will be blocked, which will seriously affect the program. Therefore, a multi-threaded solution is provided in our robot product program to solve this problem.

Lesson 5 How to Control WS2812 LED

In this lesson, we will learn how to control WS2812 LED.

5.1 Components used in this course

Components	Quantity	Picture
Raspberry Pi	1	
Robot HAT	1	
3 pin wire	1	
WS2812 RGB LED	1	

5.2 Introduction of WS2812 RGB LED

WS2812 RGB module is a low-power RGB tri-color lamp with integrated current control chip. Its appearance is the same as a 5050LED lamp bead, and each element is a pixel. The pixel contains an intelligent digital interface data latch signal shaping amplifier driving circuit, and also contains a high-precision internal oscillator and a 12V high-voltage programmable constant current control part, which effectively guarantees that the color of the pixel light is highly consistent.

WS2812 LED is a very commonly used module on our robot products. There are three WS2812 LEDS on each module. Pay attention to the direction of the signal line when connecting. The signal line needs to be connected to the “IN” port of WS2812 LED after being led from the Raspberry Pi. When the next WS2812 LED needs to be connected, we connect a signal wire drawn from the "OUT" port of the previous WS2812 LED with the "IN" port of the next WS2812 LED.

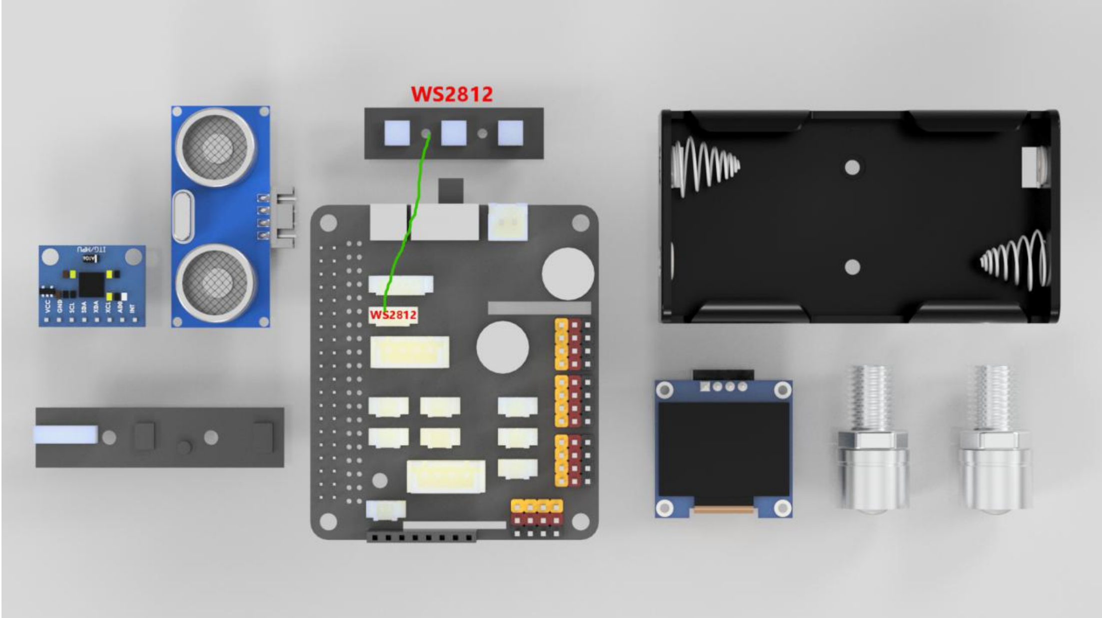
When using the Raspberry Pi to install the driver board RobotHAT, the WS2812 LED can be connected to the WS2812 interface on the RobotHAT using a 3pin cable.

We use a third-party library [rpi_ws281x] to control the WS2812 LED. You can learn about it via https://github.com/richardghirst/rpi_ws281x.

If you connect the WS2812 LED to the WS2812 interface of RobotHAT, the signal line is equivalent to connecting to the GPIO 12 of the Raspberry Pi.

5.3 Wiring diagram (Circuit diagram)

When the WS2812 LED is in use, the IN port needs to be connected to the WS2812 port on the RobotHAT driver board, as shown in the figure below:



5.4 How to control WS2812 LED

5.4.1Run the code

1. Remotely log in to the Raspberry Pi terminal.

```
Linux raspberrypi 4.19.118-v7l+ #1311 SMP Mon Apr 27 14:26:42 BST 2020 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: Sat Aug 29 08:17:49 2020 from 192.168.3.208

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:~ $
```

2. Enter the command and press Enter to enter the folder where the program is located:

cd adeept_picarpro/server/

```
pi@raspberrypi:~ $ cd adeept_picarpro/server/
pi@raspberrypi:~/adeept_picarpro/server $
```

3. View the contents of the current directory file:

ls

```
pi@raspberrypi:~/adeept_picarpro/server $ ls
app.py            findline.py    instruction.txt   OLED.py         servo.py
appserver.py      FPV.py         Kalman_filter.py  PID.py          switch.py
base_camera.py    FPVtest.py     LEDapp.py         __pycache__     ultra.py
camera_opencv.py  functions.py   LED.py            robotLight.py   webServer.py
dist              info.py        move.py           RPIservo.py
pi@raspberrypi:~/adeept_picarpro/server $
```

4. Enter the command and press Enter to run the program:

sudo python3 LED.py

```
pi@raspberrypi:~/adeept_picarpro/server $ sudo python3 LED.py
```

5. After running the program successfully, you will observe that the WS2812 light turns red.

5.4.2 The main code program of this lesson

For the complete code, please refer to the file LED.py.

```
import time
from rpi_ws281x import *
```

The third-party library rpi_ws281x can be used to control the ws_2812 module used on our products. The package name imported here is inconsistent with the official routine, but this third-party library can be successfully used after testing and using the import method here.

```
class LED:
  def __init__(self):
    self.LED_COUNT      = 16 # 3. Set to the total number of LED lights on the robot product
    self.LED_PIN        = 12 # Set as the input pin number of the LED lamp group
    self.LED_FREQ_HZ    = 800000
    self.LED_DMA        = 10
    self.LED_BRIGHTNESS = 255
    self.LED_INVERT     = False
    self.LED_CHANNEL    = 0
    # Use the above configuration items to create a strip
    self.strip = Adafruit_NeoPixel(
      self.LED_COUNT,
      self.LED_PIN,
      self.LED_FREQ_HZ,
      self.LED_DMA,
      self.LED_INVERT,
```

```
            self.LED_BRIGHTNESS,
            self.LED_CHANNEL
        )
        self.strip.begin()
    def colorWipe(self,color): # This function is used to change the color of the LED lamp
        for i in range(self.strip.numPixels()): #Only one LED lamp color can be set at a time, so a cycle is required
            self.strip.setPixelColor(i, color)
        self.strip.show() #After calling the show method, the color will really change
```

Build LED control class.

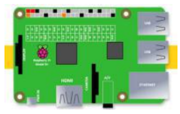
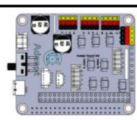
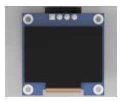
```
if __name__ == '__main__':
    led = LED()
    led.colorWipe(0,0,255)
```

①The colorWipe() method must be used to pack the RGB value, and then pass it to the function for color switching.

Lesson 6 Displaying Text on the OLED Screen

In this lesson, we will learn how to display text on the OLED screen.

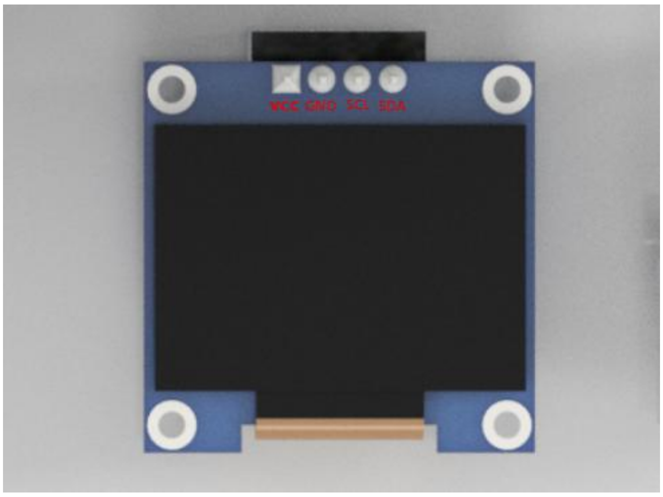
6.1 Components used in this course

Components	Quantity	Picture
Raspberry Pi	1	
Robot HAT	1	
4 pin wire	1	
OLED screen	1	

6.2 Introduction of OLED Screen

OLED (Organic Light-Emitting Diode), also known as organic electric laser display, organic light emitting semiconductor (Organic Electroluminesence Display, OLED). OLED is a kind of current-type organic light-emitting device, which produces light by the injection and recombination of carriers, and the luminous intensity is proportional to the injected current. The PiCar Pro robot uses an OLED screen to display the expressions or some parameters of the robot. OLED Screen is a commonly used module on robot products. Due to the black non-luminous feature of OLED Screen, this type of screen has extremely high contrast. Even if the ambient light is strong, you can see the information on the OLED Screen clearly, and the power consumption is relatively low. We only need to connect the power supply and the GPIO port to control it.

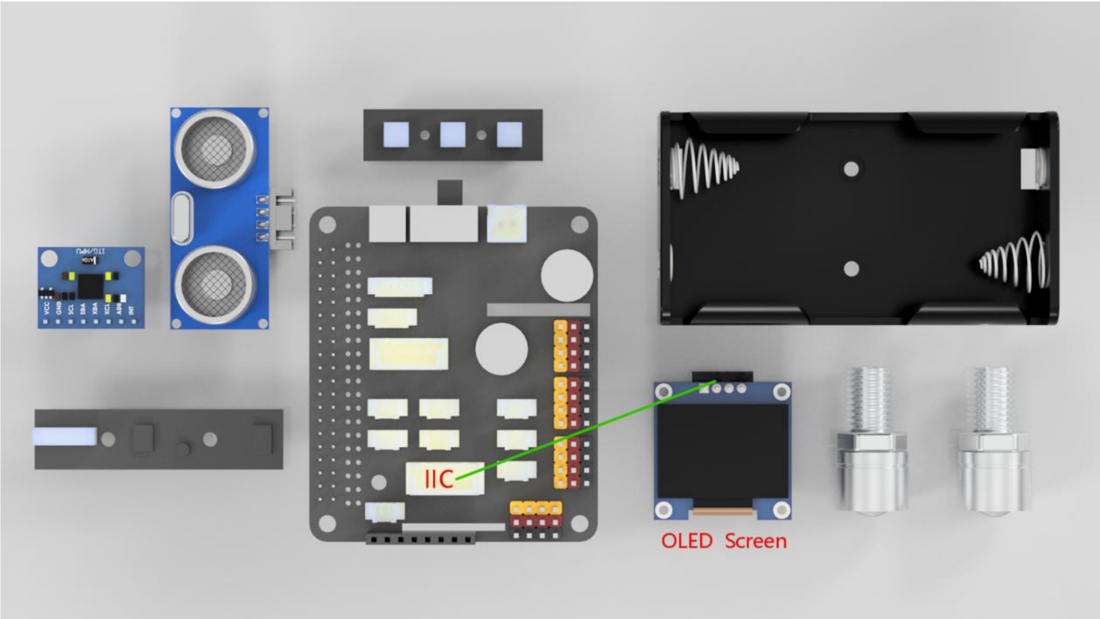
If you want to use OLED Screen, you need to use a 4-pin cable with anti-reverse interface to connect the OLED screen to the IIC interface on the Robot HAT.



If you do not use Robot HAT driver board to connect with Raspberry Pi driver board, then you need to connect Vin of OLED screen to 5V or 3.3V of Raspberry Pi, and connect GND of OLED screen to GND of Raspberry Pi. Connect SCL of Robot HAT to SCL of OLED, and SCA of Robot HAT to SCA of Raspberry Pi. Please refer to the pin definition diagram of Raspberry Pi for specific pins.

6.3 Wiring diagram (Circuit diagram)

If you want to use the OLED Screen module, you need to connect the IIC interface on the Robot HAT driver board, as shown in the figure below:



6.4 How to display text on the OLED screen

6.4.1Run the code

1. Remotely log in to the Raspberry Pi terminal.

```
Linux raspberrypi 4.19.118-v7l+ #1311 SMP Mon Apr 27 14:26:42 BST 2020 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: Sat Aug 29 08:17:49 2020 from 192.168.3.208

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:~ $
```

2. Enter the command and press Enter to enter the folder where the program is

located:

cd adeept_picarpro/server/

```
pi@raspberrypi:~ $ cd adeept_picarpro/server/
pi@raspberrypi:~/adeept_picarpro/server $
```

3. View the contents of the current directory file:

ls

```
pi@raspberrypi:~ $ cd adeept_picarpro/server/
pi@raspberrypi:~/adeept_picarpro/server $ ls
app.py            findline.py    instruction.txt   OLED.py          servo.py
appserver.py      FPV.py         Kalman_filter.py  PID.py           switch.py
base_camera.py    FPVtest.py     LEDapp.py         __pycache__      ultra.py
camera_opencv.py  functions.py   LED.py            robotLight.py    webServer.py
dist              info.py        move.py           RPIservo.py
pi@raspberrypi:~/adeept_picarpro/server $
```

4. Enter the command and press Enter to run the program:

sudo python3 OLED.py

```
pi@raspberrypi:~/adeept_picarpro/server $ sudo python3 OLED.py
```

5. After running the program successfully, you will observe that the word "GEWBOT.COM" will be displayed on the OLED screen.

6. When you want to terminate the running program, you can press the shortcut key "Ctrl + C" on the keyboard.

6.4.2 The main code program of this lesson

For the complete code, please refer to the file OLED.py.

```python
from luma.core.interface.serial import i2c
from luma.core.render import canvas
from luma.oled.device import ssd1306
import time

serial = i2c(port=1, address=0x3C)
device = ssd1306(serial, rotate=0)

text_1 = 'GEWBOT.COM' # Define the text displayed on the OLED screen, up to 6 lines.
text_2 = 'IP:CONNECTING'
text_3 = '<ARM> OR <PT> MODE'
text_4 = 'MPU6050 DETECTING'
text_5 = 'FUNCTION OFF'
text_6 = 'Message:None'
```

Import dependent libraries and initialize.

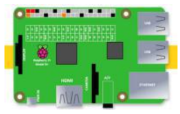
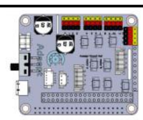
```
def run():
    with canvas(device) as draw:
        draw.text((0, 0), text_1, fill="white")
        draw.text((0, 10), text_2, fill="white")
        draw.text((0, 20), text_3, fill="white")
        draw.text((0, 30), text_4, fill="white")
        draw.text((0, 40), text_5, fill="white")
        draw.text((0, 50), text_6, fill="white")

if __name__ == '__main__':
    run()
```

Light up the OLED screen.

Lesson 7 How to Control DC Motor

In this lesson, we will learn how to control DC Motor.

7.1 Components used in this course

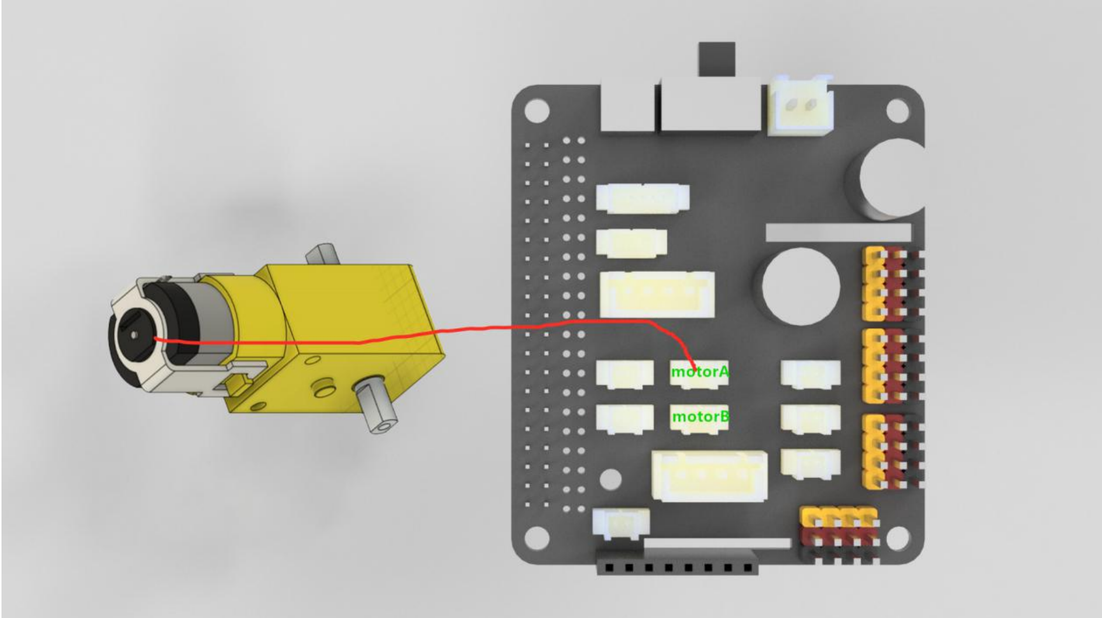
Components	Quantity	Picture
Raspberry Pi	1	
Robot HAT	1	
DC Motor	1	

7.2 The introduction of DC Motor

PiCarPro robot products use DC motor as a power device. DC motor is a device that converts DC electrical energy into mechanical energy. It is widely used to drive various equipment, such as electric fans, remote control cars, electric windows, etc.The DC motor is very suitable as the walking mechanism of the robot.

7.3 Wiring diagram (Circuit diagram)

When the DC Motor module is in use, it needs to be connected to the motorA or motorB interface on the Robot HAT drive board. The yellow wire is connected to the yellow pin, the red wire is connected to the red pin, and the brown wire is connected to the black pin, as shown below:



7.4 How to control Motor

7.4.1Run the code

1. Remotely log in to the Raspberry Pi terminal.

```
Linux raspberrypi 4.19.118-v7l+ #1311 SMP Mon Apr 27 14:26:42 BST 2020 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: Sat Aug 29 08:17:49 2020 from 192.168.3.208

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:~ $
```

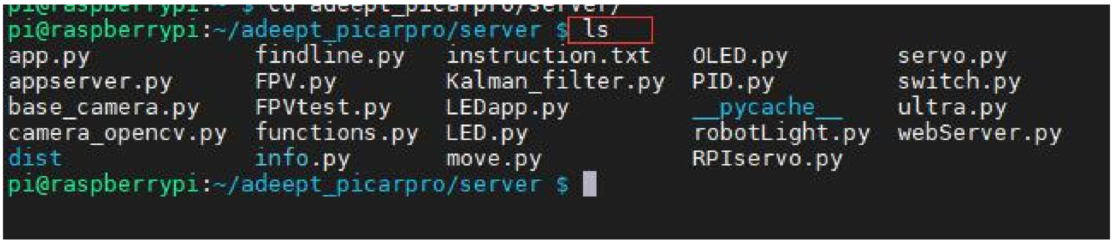
2. Enter the command and press Enter to enter the folder where the program is located:

cd adeept_picarpro/server/

```
pi@raspberrypi:~ $ cd adeept_picarpro/server/
pi@raspberrypi:~/adeept_picarpro/server $
```

3. View the contents of the current directory file:

ls



4. Enter the command and press Enter to run the program:

sudo python3 move.py

5. After running the program successfully, you will observe that the Motor will rotate for about 1 second and then stop, and the program will also stop. If you need the motor to rotate again, you need to run the program again.

7.4.2 The main code program of this lesson

For the complete code, please refer to the file move.py.

```
import time
import RPi.GPIO as GPIO

Motor_A_EN    = 4
Motor_A_Pin1  = 26
Motor_A_Pin2  = 21
```

Import the dependent library, where Motor_EN = 4, Motor_Pin1 = 26, Motor_Pin2 = 21 are the parameters of the corresponding interface motorA.

```
def setup():#Motor initialization
    global pwm_A
    GPIO.setwarnings(False)
    GPIO.setmode(GPIO.BCM)
    GPIO.setup(Motor_A_EN, GPIO.OUT)
    GPIO.setup(Motor_A_Pin1, GPIO.OUT)
    GPIO.setup(Motor_A_Pin2, GPIO.OUT)

    motorStop()
    try:
        pwm_A = GPIO.PWM(Motor_A_EN, 1000)
    except:
        pass
```

Set the motor initialization function.

```
def destroy():
    motorStop()
    GPIO.cleanup()             # Release resource

```

Set the motor stop function.

```
def move(speed, direction, turn, radius=0.6):   # 0 < radius <= 1
    #speed = 100
    if direction == 'forward':
        if turn == 'right':
            motor_left(0, left_backward, int(speed*radius))
            motor_right(1, right_forward, speed)
        elif turn == 'left':
            motor_left(1, left_forward, speed)
            motor_right(0, right_backward, int(speed*radius))
        else:
            motor_left(1, left_forward, speed)
            motor_right(1, right_forward, speed)
    else:
        pass
```

Set the motor rotation function.

```
if __name__ == '__main__':
```

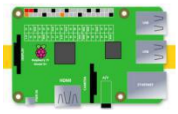
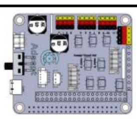
```
    try:
        speed_set = 60
        setup()
        move(speed_set, 'forward', 'no', 0.8)
        time.sleep(1.3)
        motorStop()
        destroy()
    except KeyboardInterrupt:
        destroy()
```

Instantiate the object.

Lesson 8 How to Control the Ultrasonic Module

In this lesson, we will learn how to read the data of the RGB ultrasonic ranging module.

8.1Components used in this course

Components	Quantity	Picture
Raspberry Pi	1	
Robot HAT	1	
ultrasonic module	1	
4 pin wire	1	

8.2 Introduction of RGB ultrasonic ranging module

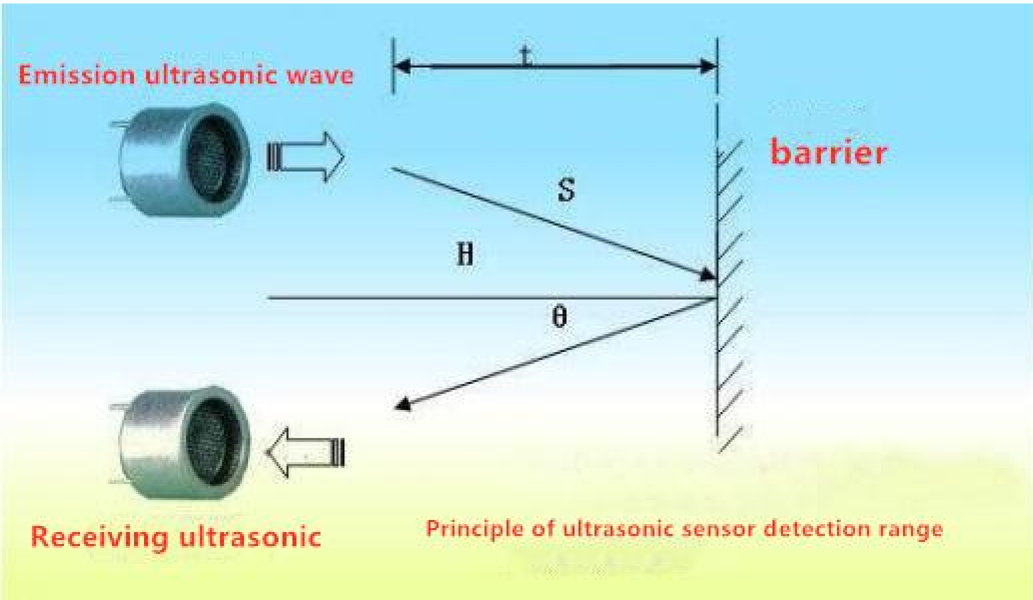
The ultrasonic ranging module used in our PiCarPro product has four pins, namely VCC, GND, Echo and Trig. The HC-SR04 can provide a non-contact distance sensing function of 2cm-400cm, and the ranging accuracy can reach 3mm; The module includes an ultrasonic transmitter, receiver and control circuit. The basic working principle is as follows:

Use IO port TRIG to trigger distance measurement, and give a high level signal of at least 10us.

The module automatically sends eight 40khz square waves, and automatically detects whether there is a signal return.

There is a signal return, and a high level is output with the IO port ECHO. The duration of the high level is the time from emission to return of the ultrasonic wave.

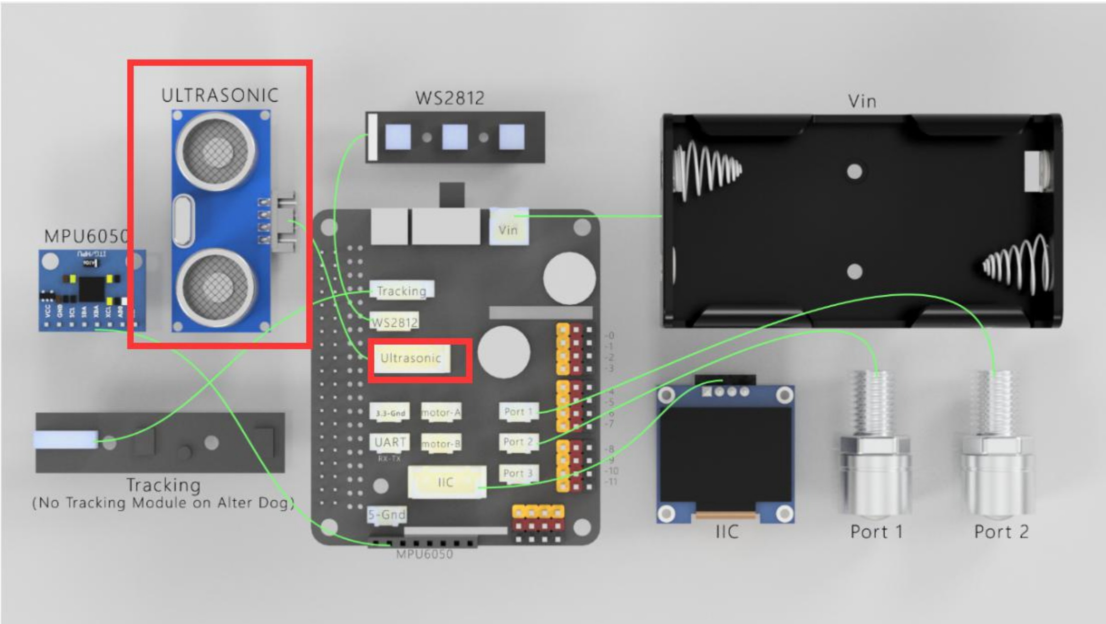
The principle of distance detection by ultrasonic ranging sensor: the method of detecting distance by ultrasonic is called echo detection method, that is, the ultrasonic transmitter emits ultrasonic waves in a certain direction, and the timer starts timing at the same time as the launch time. The ultrasonic waves propagate in the air and encounter obstacles on the way. When the object surface (object) is blocked, it will be reflected back immediately, and the ultrasonic receiver will immediately stop timing when the reflected ultrasonic wave is received. The propagation speed of ultrasonic waves in the air is 340m/s. According to the time t recorded by the timer, the distance s from the launch point to the obstacle surface can be calculated, namely: s=340t/2. Using this principle of ultrasound, the ultrasonic ranging module is widely used in practical applications, such as car reversing radar, unmanned aerial vehicle, and smart car.



When using Robot HAT driver board, the ultrasonic sensor needs to be connected to the Ultrasonic interface on the driver board, and must not be connected to the IIC port to avoid burning the ultrasonic module. (IIC is an interface used to connect I2C devices, and the pin positions of VCC and GND are different from Ultrasonic).

8.3Wiring diagram (Circuit diagram)

When using Robot HAT driver board, you need to connect the ultrasonic sensor to the Ultrasonic interface on the driver board. Do not connect it to the IIC port to avoid burning the ultrasonic module. (IIC is an interface used to connect I2C devices, and the pin positions of VCC and GND are different from Ultrasonic).



8.4 Obtain data from ultrasonic sensors

8.4.1Run the code

1. Remotely log in to the Raspberry Pi terminal.

```
Linux raspberrypi 4.19.118-v7l+ #1311 SMP Mon Apr 27 14:26:42 BST 2020 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: Sat Aug 29 08:17:49 2020 from 192.168.3.208

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:~ $
```

2. Enter the command and press Enter to enter the folder where the program is located:

cd adeept_picarpro/server/

```
pi@raspberrypi:~ $ cd adeept_picarpro/server/
pi@raspberrypi:~/adeept_picarpro/server $
```

3. View the contents of the current directory file:

ls

```
pi@raspberrypi:~/adeept_picarpro/server $ ls
app.py             findline.py    instruction.txt   OLED.py        servo.py
appserver.py       FPV.py         Kalman_filter.py  PID.py         switch.py
base_camera.py     FPVtest.py     LEDapp.py         __pycache__    ultra.py
camera_opencv.py   functions.py   LED.py            robotLight.py  webServer.py
dist               info.py        move.py           RPIservo.py
pi@raspberrypi:~/adeept_picarpro/server $
```

4. Enter the command and press Enter to run the program:

sudo python3 ultra.py

```
pi@raspberrypi:~/adeept_picarpro/server $ sudo python3 ultra.py
```

5. After successfully running the program, the command window will display the distance data of the obstacle detected by the ultrasonic sensor. When you use an object to approach it directly in front of the ultrasonic sensor, the detected distance data will change.

6. When you want to terminate the running program, you can press the shortcut key "Ctrl + C" on the keyboard.

8.4.2 The main code program of this lesson

For the complete code, please refer to the file ultra.py.

```
import RPi.GPIO as GPIO
import time

Tr = 11 # The pin number of the input end of the ultrasonic module
Ec = 8 # Pin number of the output end of the ultrasonic module

GPIO.setmode(GPIO.BCM)
GPIO.setup(Tr, GPIO.OUT,initial=GPIO.LOW)
GPIO.setup(Ec, GPIO.IN)
```

Import the dependent library and initialize it.

```
def checkdist():
    GPIO.output(Tr, GPIO.HIGH) # Set the input terminal of the module to high level and send out an initial sound wave.
    time.sleep(0.000015)
    GPIO.output(Tr, GPIO.LOW)

    while not GPIO.input(Ec): # When the module no longer receives the initial sound wave ①
        pass
    t1 = time.time() # Write down the time when the initial sound wave was emitted.
    while GPIO.input(Ec): # When the module receives the return sound wave.
        pass
    t2 = time.time() # Write down the time when the return sound wave was captured.

    return round((t2-t1)*340/2,2) # Calculate the distance.
```

Define the ranging function. When the ultrasound module captures ultrasound, its output terminal will output high frequency. The GPIO.input method can determine whether the output terminal is outputting high-frequency signals.

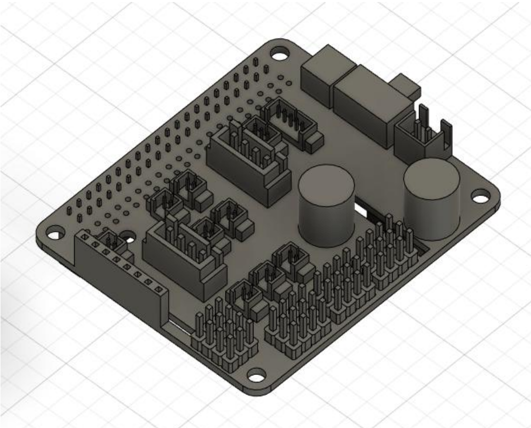
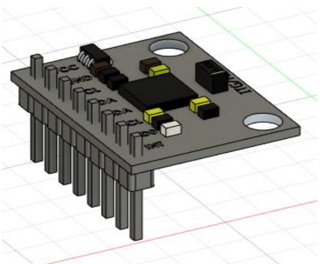
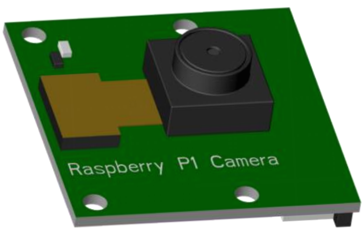
```
for i in range(10):
    print(checkdist())
    time.sleep(1)
```

The ultrasonic module measures the distance and displays the data on the console.

Lesson 9 How to Assemble PiCar Pro

9.1Components used in this course

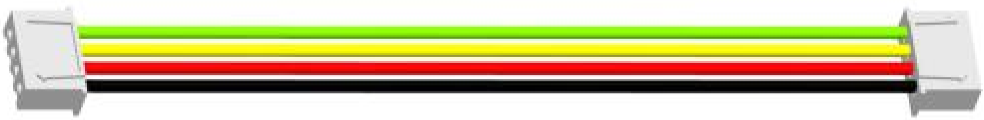
The components we provide:

RobotHAT driver board x 1 	**LED headlight** x 1 
18650 battery holder x 1 	**LED strip** x 1 
MPU6050（pin header） x 1 	**Camera** x 1 

180 degree AD002 servo x 3

3pin cable（for ws2812） x 1

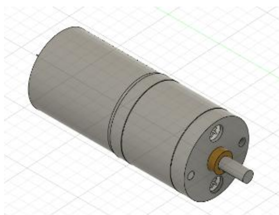
4pin cable x 2



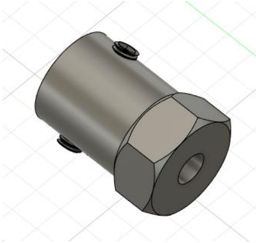
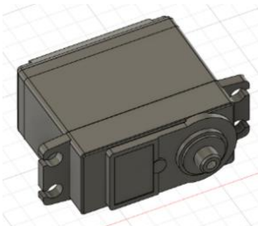
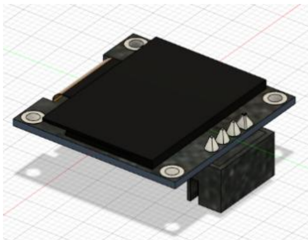
40CM camera cable（black） x1

Small LED light x2

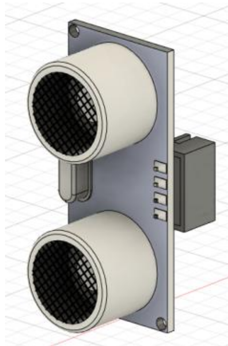
DC gear motor JGA25-370 DC12V130RPM（with 2pin cable） x2



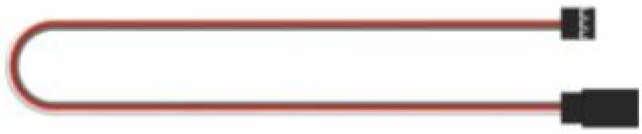
Internal 4mm coupling (with screws) X2


Big wheels x4 
180 degree MG946R servo x4 
Inner 4mm long 22mm plastic tube x2 
Red metal rocker x1 
OLED screen（anti-reverse connection interface） x1 

ultrasonic module x1

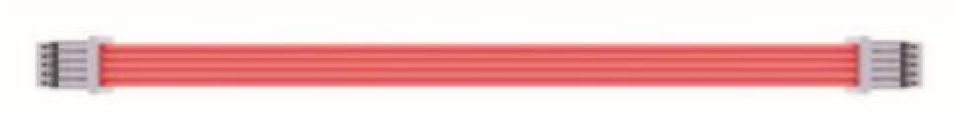


Servo extension cable x2



Tracking module x1

5pin cable x1



Copper pillar

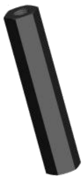
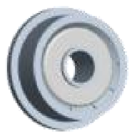
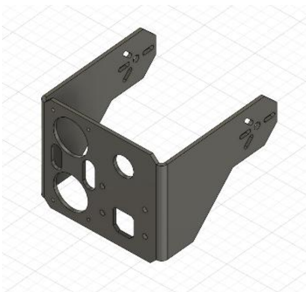
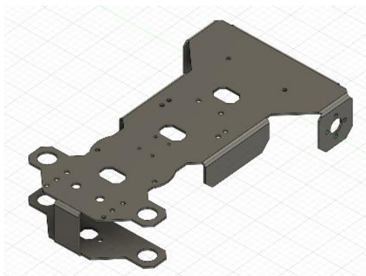
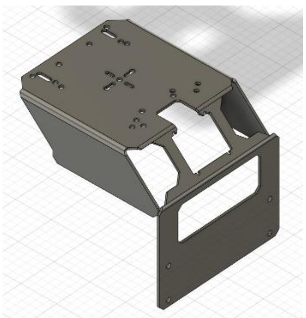
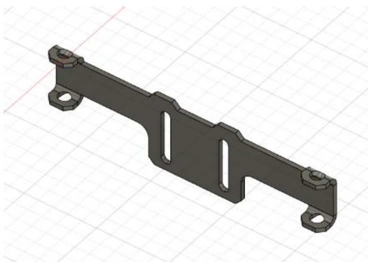
M2.5*14	x4
M3*60	x4

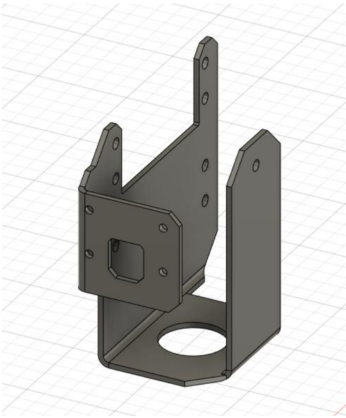
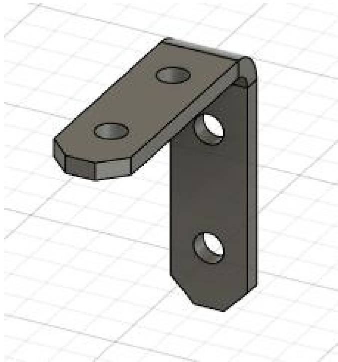
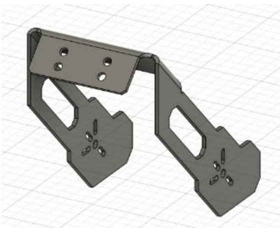
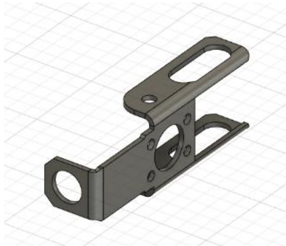
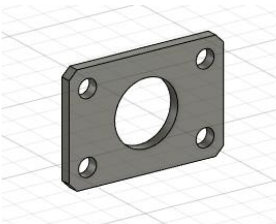
Gasket

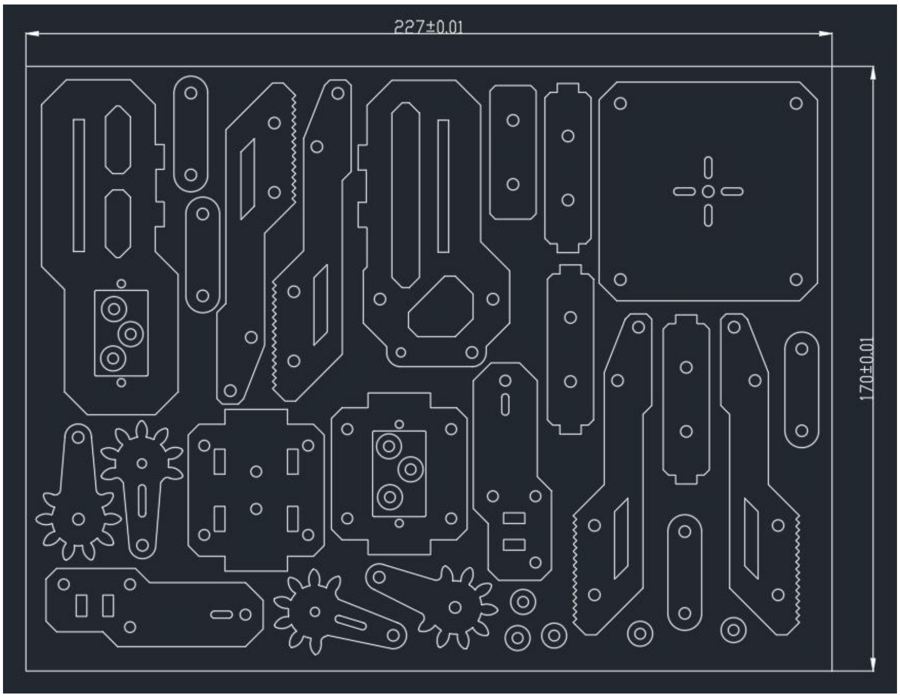
M4*8*1	x2

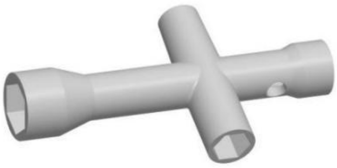
Round head screw (black)

M2*8	x25
M2*14	x2
M2.5*4	x6（silver）
M2.5*8	x4
M2.5*12	x2
M3*4	x23
M3*16	x5
M3*10	x43
M4*6	x1（silver）
M4*25	x1（silver）
M4*40	x4（silver）

Double pass nylon column M3*15 x2 M3*20 x4 M3*40 x3 	**Flat head screw** M3*6 x1 
self-tapping screw M1.4*6 x5 M1.7*6*6 x10 	**Hex nuts** M2 x23（black） M3 x13（black） M4 x1（silver) 
Locknut M3 x11 M4 x1 	**Mechanical parts bearings** MF84ZZ x2 
A set of aluminum alloy	
 x1	
 x1	 x1

 x1	 x1
 x1	 x1
 x1	
A set of acrylic	



Small Cross-head Screwdriver x 1 	**Cross Socket Wrench x 1** 
Large Cross-head Screwdriver x 1 	
Large straight Screwdriver x 1 	
20CM Winding Pipe x 1 	

Bring your own components:

Raspberry Pi x1

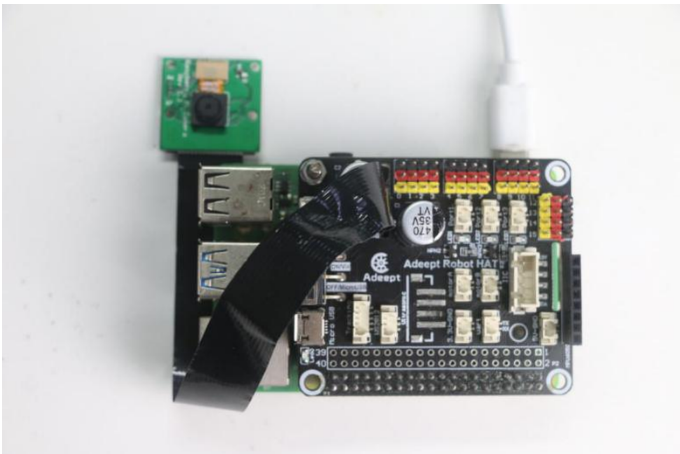
18650 battery (high current power battery) x2

9.2 Precautions for assembly

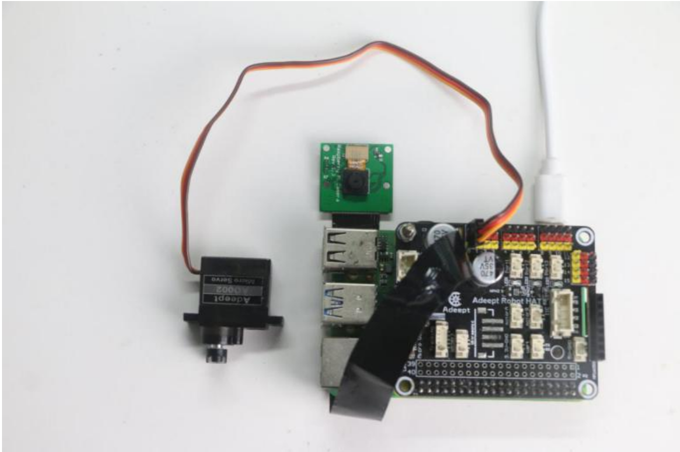
Due to the large number of servo used in this product, the installation of the servo has a greater impact on the performance of the product, so you need to power on the servo and control the servo to rotate to the middle position before installing the servo rocker arm, in this way, the rocker arm of the servo installed at the specified angle can be in the middle position of it.

Steps to power on the servo:

1.Boot the Raspberry Pi.



2.Install the servo. Pay attention to the direction of the interface during installation. The yellow wire is connected to the yellow pin, the red wire is connected to the red pin, and the brown wire is connected to the black pin.



3.After the Raspberry Pi is powered on, it will automatically run webServer.py, and after webServer.py is running, it will control all the servo ports to send signals to move to the middle position. When you install the servo rocker arm, you can connect the servo to any port at any time. When the servo is connected to the port, the servo will rotate to the middle position. Install the rocker arm according to the specified angle then you can disconnect the servo from the port and plug in a new servo that has not been installed with a rocker arm. This new servo without a rocker arm will also rotate to the middle position.

4. It takes a while to boot the Raspberry Pi to control the PCA9685 to set all the servo ports to the signal to rotate to the middle position. The boot process of the Raspberry Pi is about 30-50s.

5. All the servo rocker arm installation angles shown in the document are the middle position of the servo rotation. When the servo is turned on to the middle position, install the servo rocker arm at the angle shown in the document.

6. Note: all lock nuts involved in this document should not be tightened.

9.3 Precautions for power supply during assembly

1. When you are performing software installation, structural assembly or program debugging, you can use a USB cable to power the Raspberry Pi. If the

Raspberry Pi is installed with RobotHAT, you can connect the USB cable to the USB port on the RobotHAT. RobotHAT will power the Raspberry Pi with the GPIO interface.

2. Different raspberry parties have different current requirements. For example, the Raspberry Pi 3B needs at least 2A to boot up, and the Raspberry Pi 4 needs 3A to boot normally. You can check before you use the power adapter to power the Raspberry Pi. The specifications on your power adapter.

3. When RobotHAT is connected to a load, such as a motor or multiple servos, a high-current power supply is required to connect to the Vin on the RobotHAT. You can use two 18650 batteries that support high-current to power the RobotHAT.Our product will provide a 18650 battery holder with 2pin interface, you can directly connect it with RobotHAT.

4. When using the USB interface on RobotHAT to supply power, the switch of RobotHAT does not control whether to supply power, the switch of RobotHAT can only control the power supply of Vin.

5. Do not use the USB port and Vin on the RobotHAT to supply power at the same time. If you need to debug the program for a long time and do not want to remove the battery, you can turn the switch on the RobotHAT to OFF, so that when you use the USB cable to connect to the RobotHAT , RobotHAT is powered by USB.

6. If your robot restarts automatically after booting, or disconnects and restarts at the moment the robot starts moving after normal booting, it is most likely because your power supply does not output enough current. The robot will automatically restart when it is turned on. Run the program to place all the servos in the neutral position. The voltage drop generated during this process causes the Raspberry Pi to restart.

7. We tested that the peak current of the robot is around 3.75A when using 7.4V voltage power supply, so you need to use a battery that supports 4A output.

8. You can also use power lithium battery to power RobotHAT, RobotHAT

supports power supply below 15V.

9. When installing the servo rocker arm in the structure assembly, you can use the USB cable to power the RobotHAT. After the Raspberry Pi with the robot software installed, it will control the RobotHAT to set all the servo ports to output neutral signals. You can connect the servo to any servo port, the gear of the servo will turn to the neutral position, and then you can install the servo rocker arm according to the specified angle. After the rocker arm is installed, you can disconnect the servo and RobotHAT. When you need to install the rocker arm of the second servo, connect the second servo to any servo port on the drive board.

9.4 Assembly

9.4.1 Screw color description

•In order to make the structural assembly process more intuitive, we dye the screws used in the product. During the assembly process, you can refer to the fastener color in the tutorial to determine which type of screw and nut to use.

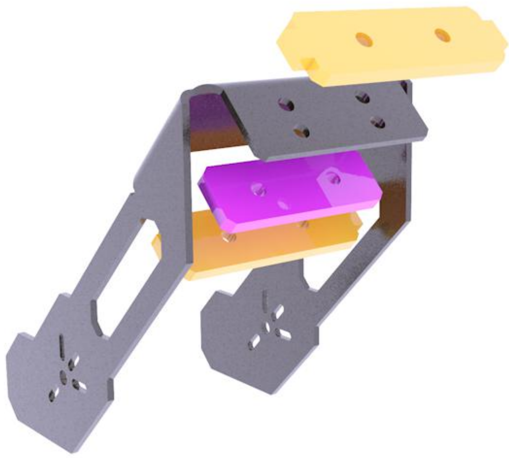
•The actual color of the product is subject to the product, and the actual screws are not colored.

9.4.2 Robotic arm assembly

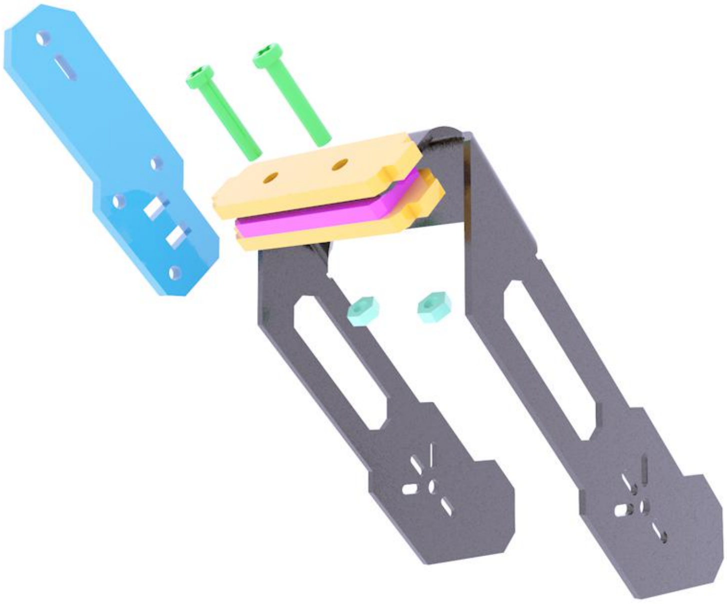
note:

1. When installing the servo, the gear of the servo should be in the middle of the servo rotation range.

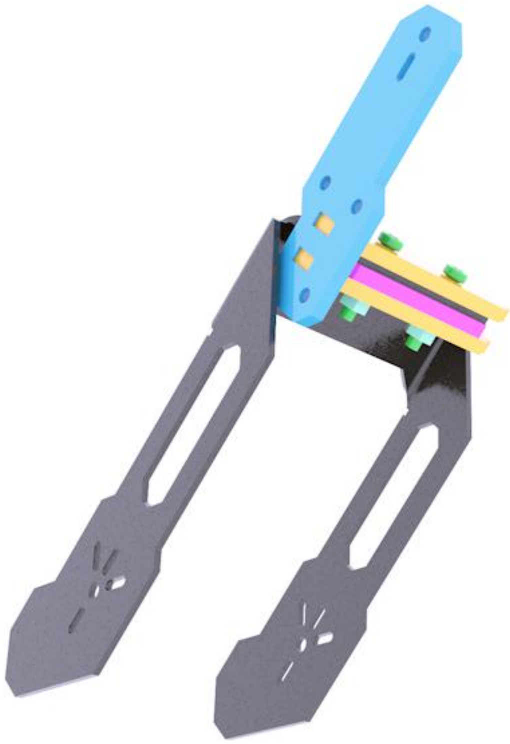
1. 2. After installing the software in the Raspberry Pi, the Raspberry Pi will automatically run the robot product program when the Raspberry Pi is turned on. After powering on the servo, the drive board will automatically place the servo in the middle position.



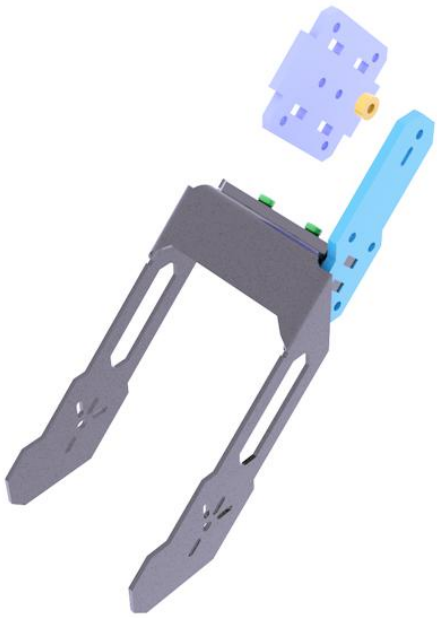
Use two M3X16 screws and two M3 nuts to fix the three acrylic panels shown above on the aluminum alloy base of the robotic arm.



Prepare the acrylic panel shown on the right side of the above picture. There is no need to fix it on the assembled component. Since the acrylic panel on the right has directions, the rendering diagram will fix it on the component to indicate the subsequent assembly position.

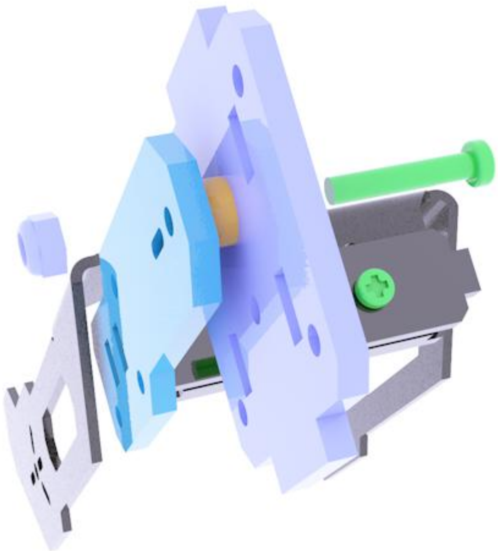


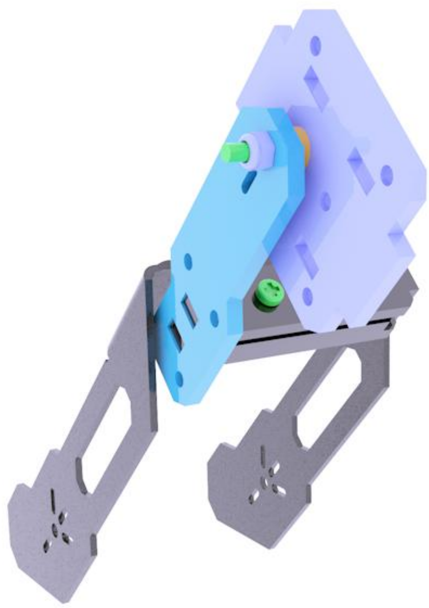
Prepare the acrylic panel and acrylic gasket as shown in the picture above.

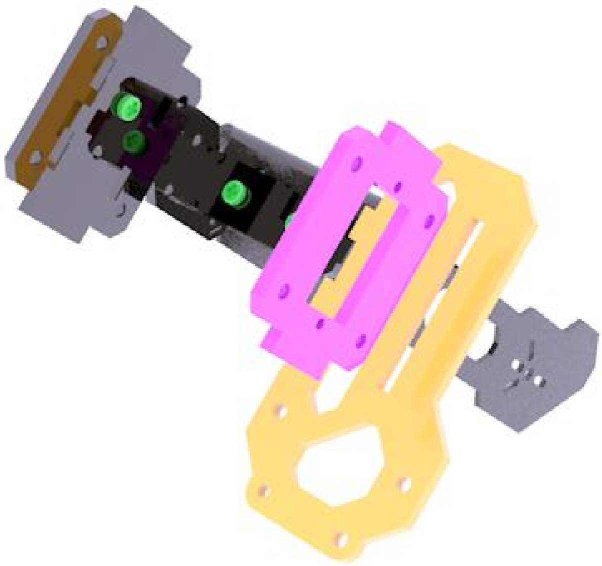


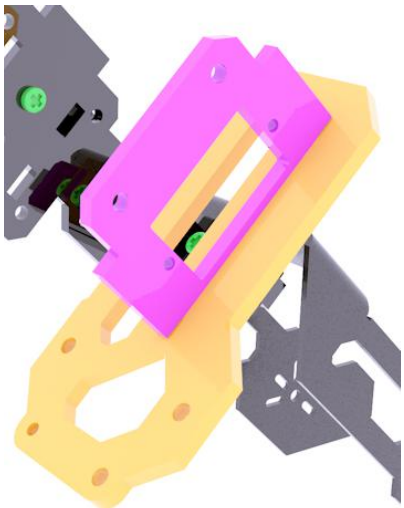
Use M3X16 screws and M3-LOCK lock nuts to fix the acrylic panel and acrylic washer as shown in the figure on the side panel. It should be noted that the lock nut should not be tightened too tightly, because this is the rotating pair of the movable part. After tightening, it will not be able to rotate and the servo will be blocked.

- Note: All lock nuts involved in this document should not be tightened.

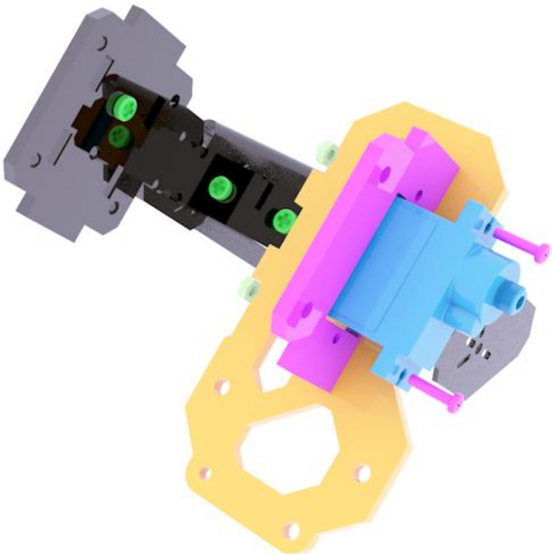


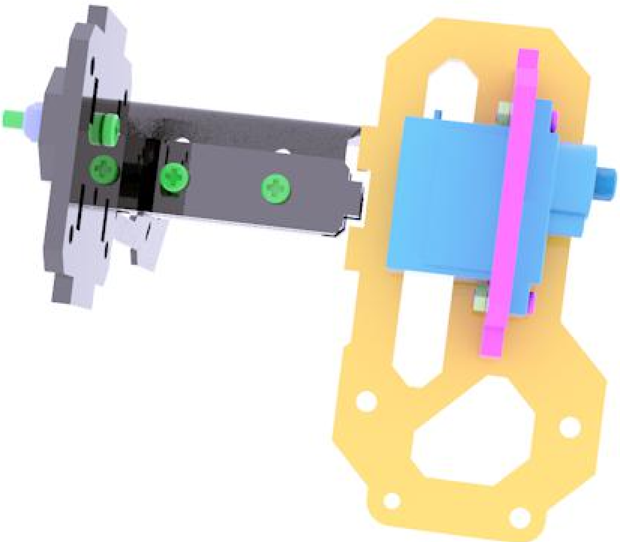






Use M2X8 screws and M2 nuts to fix the AD002 servo on the acrylic panel shown in the figure above. The acrylic panel here is similar in shape to the acrylic panel just installed, except that there are more mounting holes for fixing the servo.

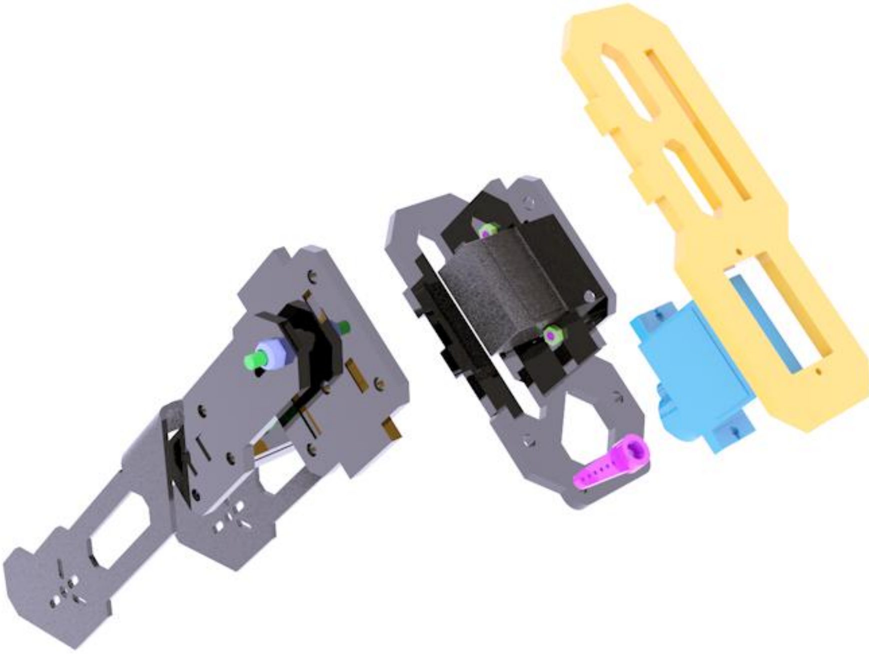


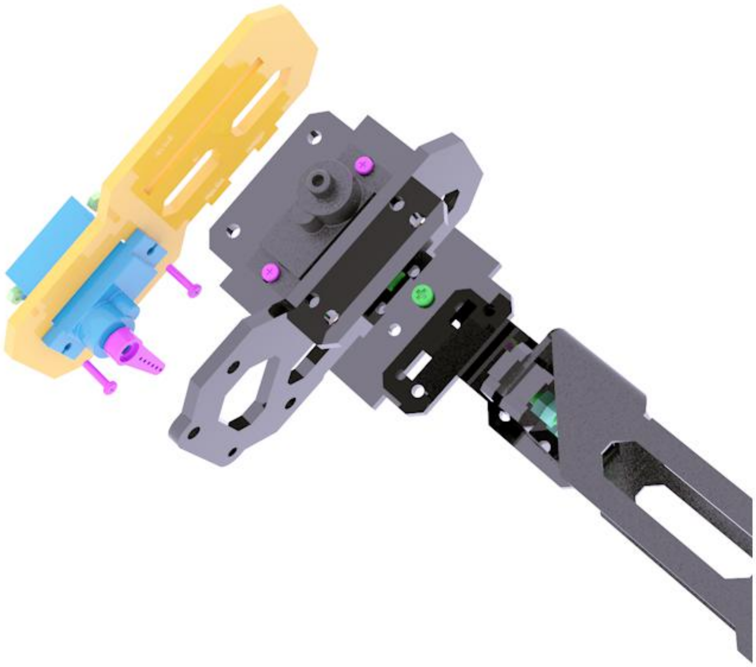


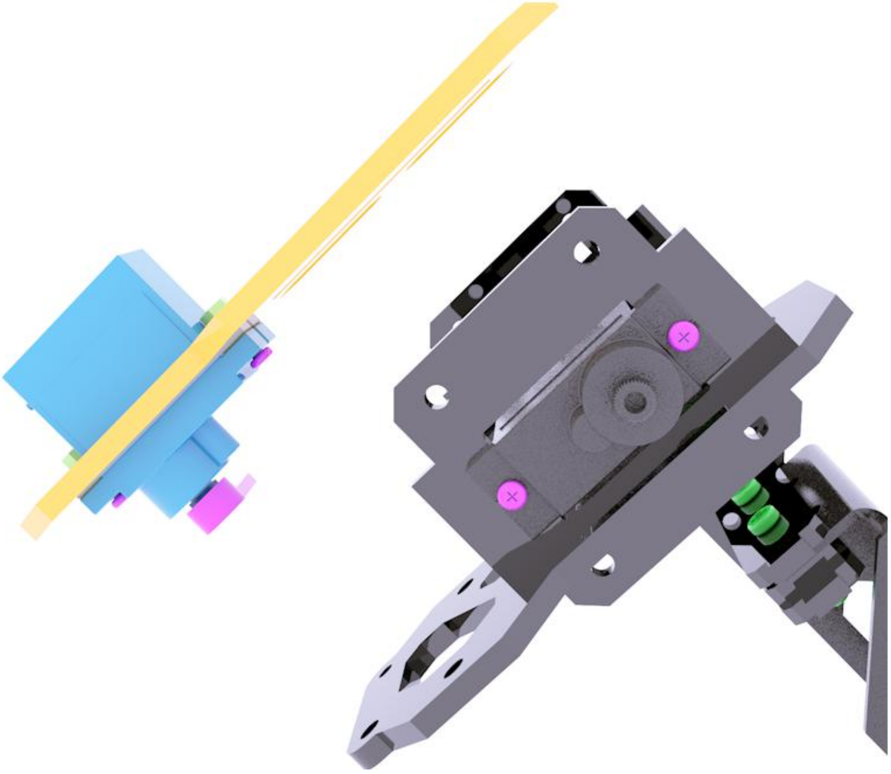
It looks like the picture above after installation.

Prepare the acrylic panel (top panel of GRIPPER) and AD002 servo as shown in the picture.

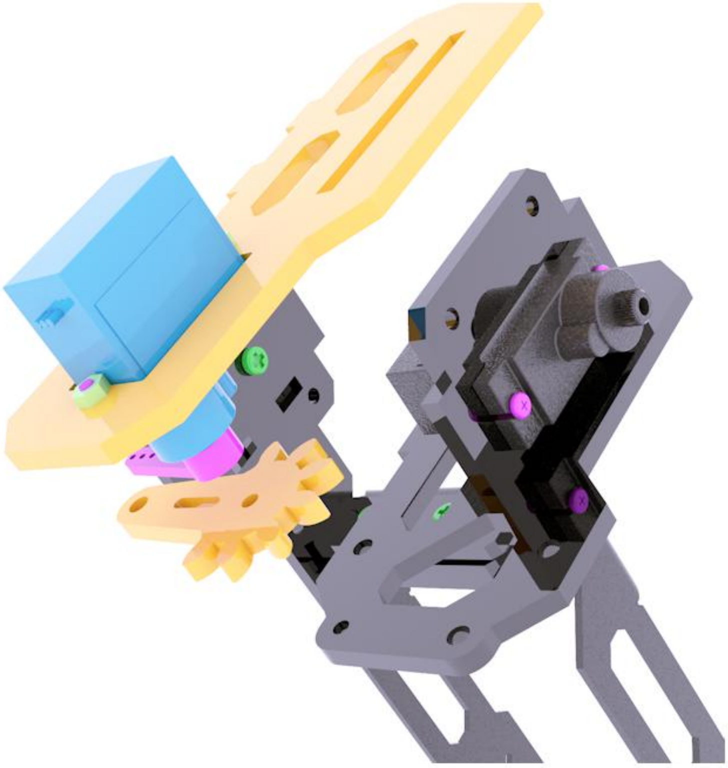
Use M2X8 screws and M2 nuts to fix AD002 servo to the upper panel of the arm chuck.



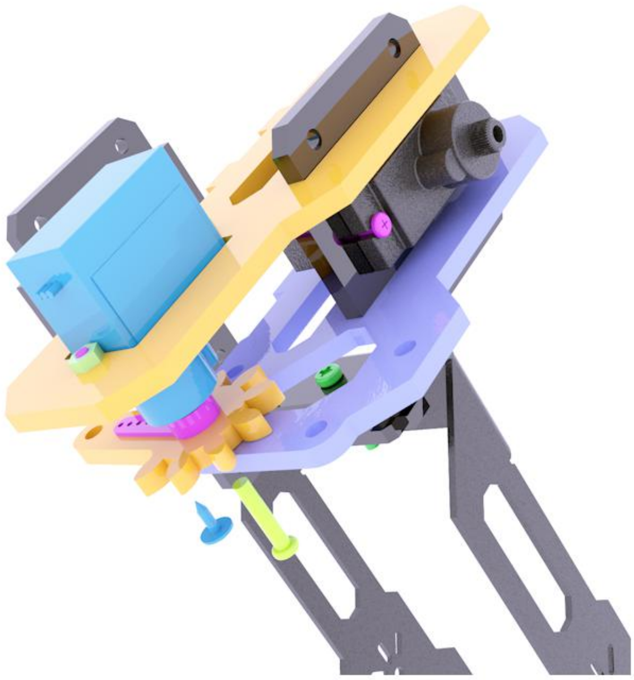




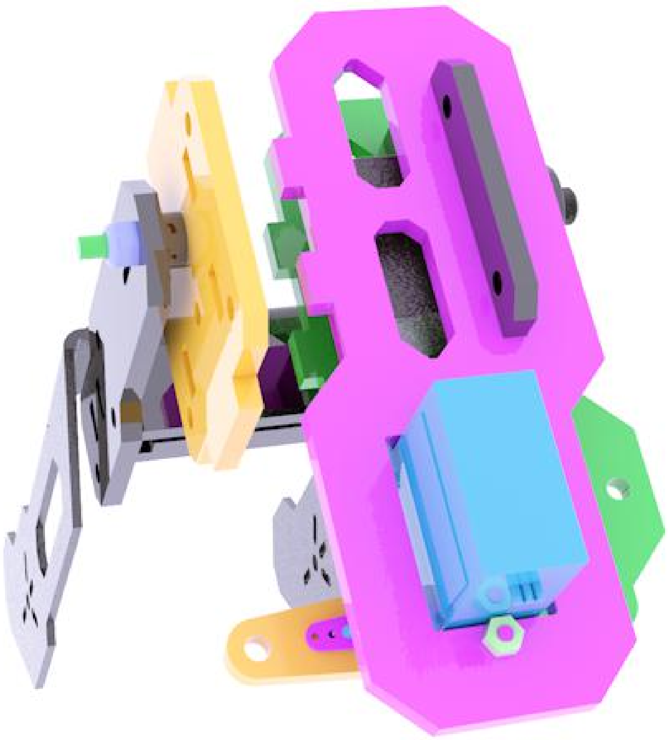
Assemble the upper panel and lower panel of the chuck. Here you can load the gear shown above between the servo arm and the lower panel. For the time being, you don’t need to fix it. You only need to know that the gear is installed here. The gear is clamped in this position to avoid the gear cannot be installed due to too small gap after assembly.

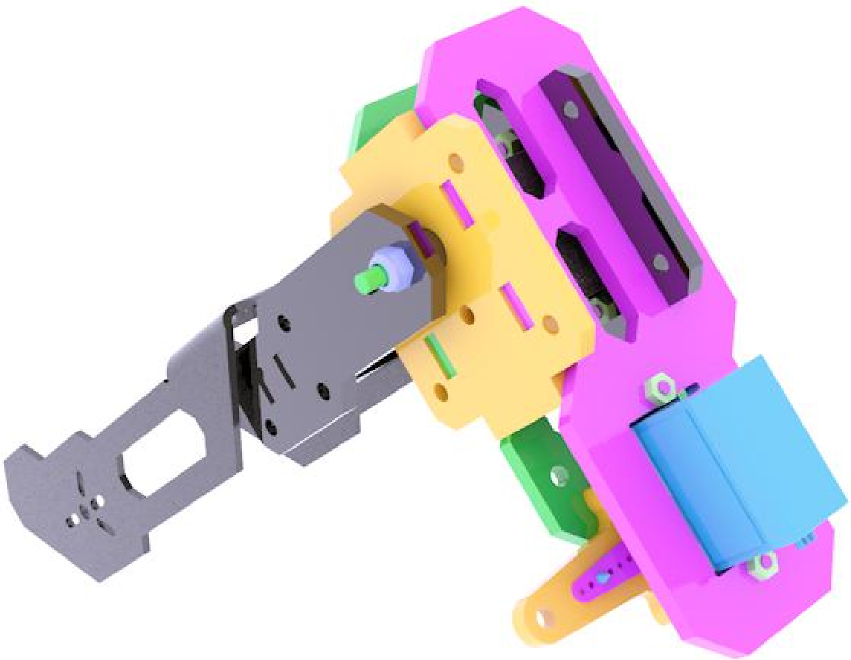


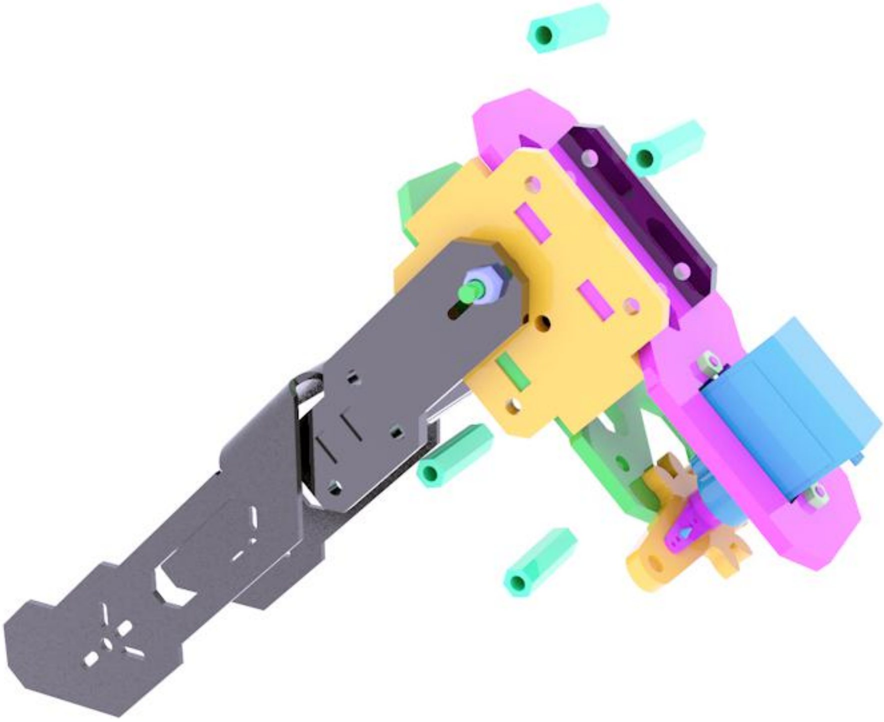
Use M2.5*12 screws and M1.7*6*6 self-tapping screws to fix

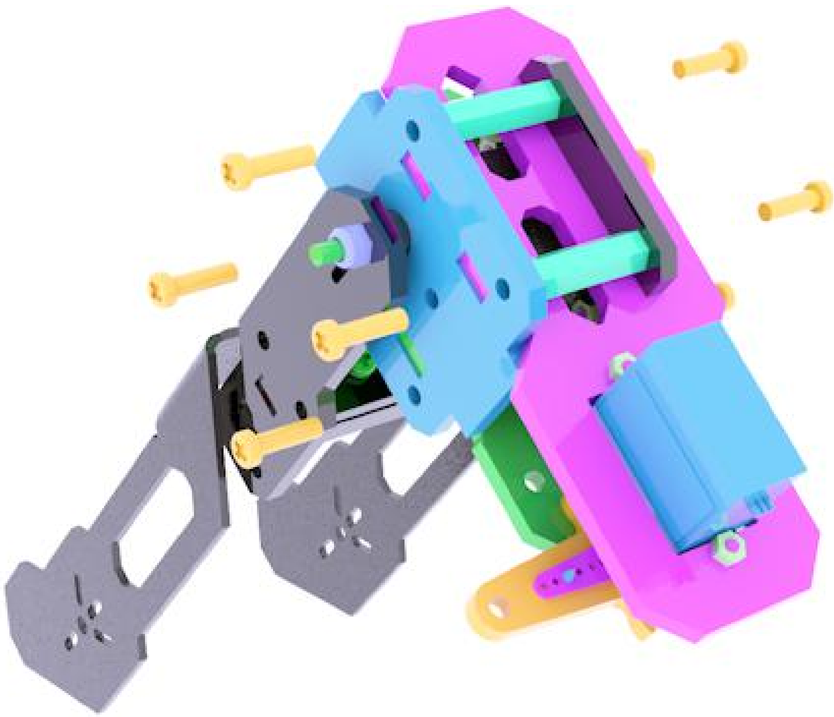


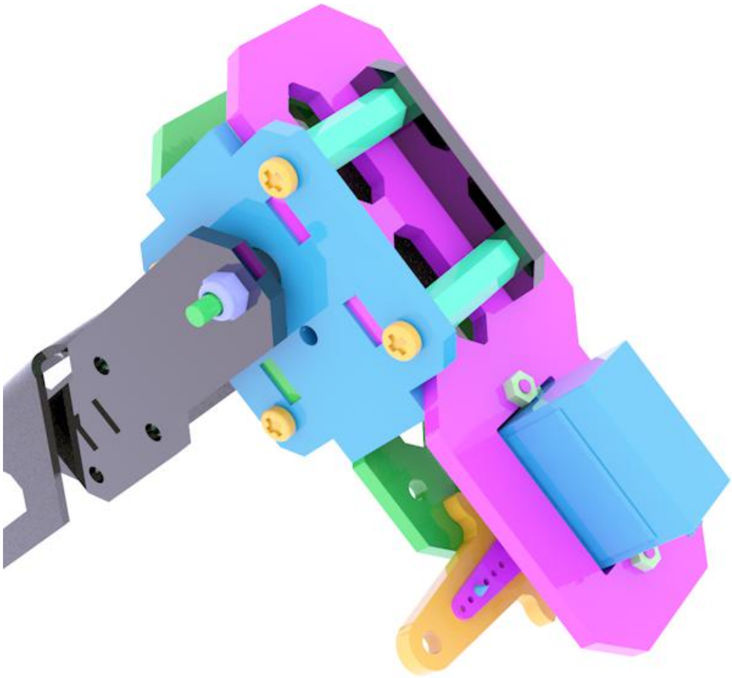
Use four M3X20 nylon posts and eight M3X10 screws to fix the left and right panels of the chuck together.



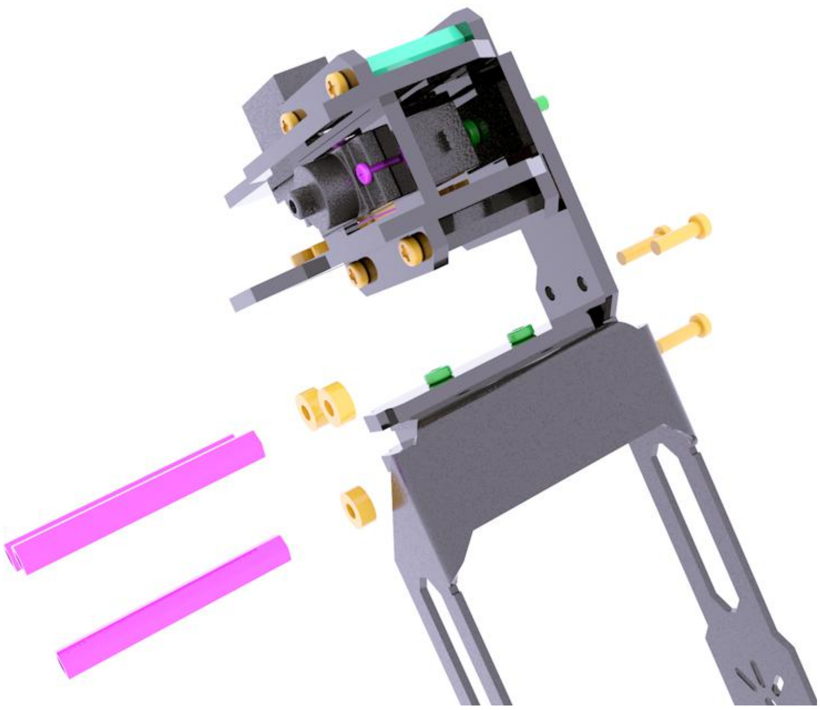


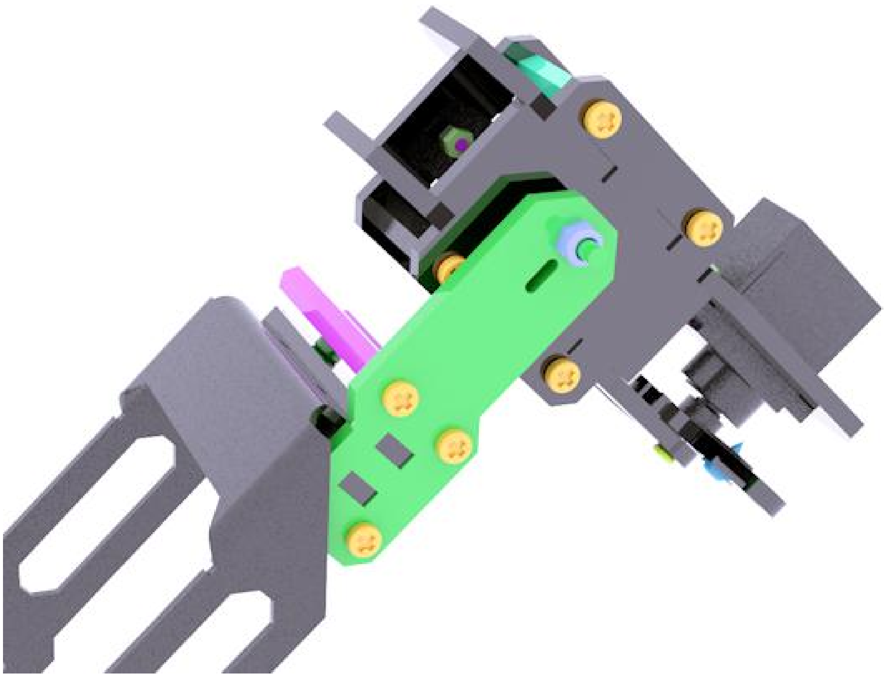


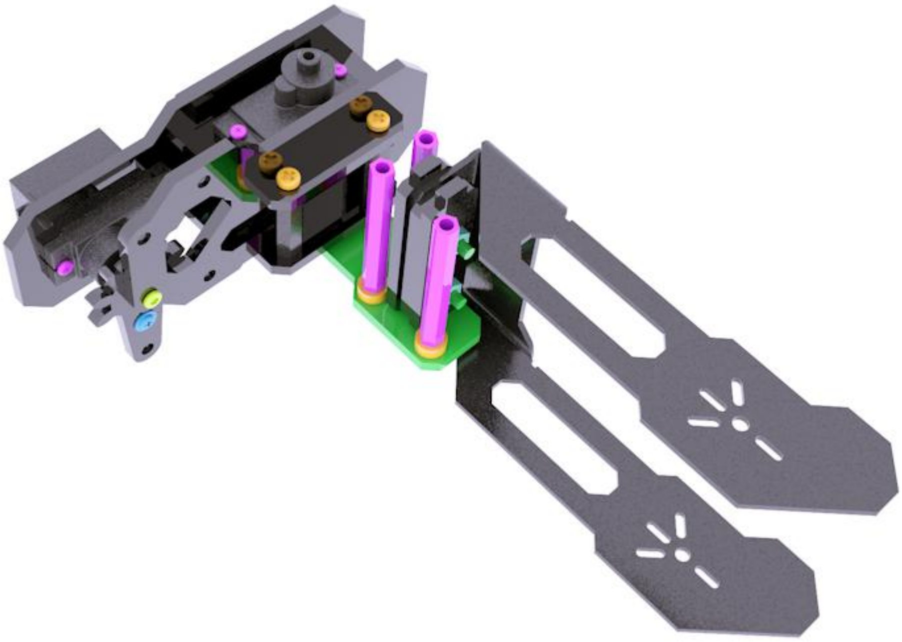




Use three M3X10 screws to fix the acrylic washer and M3X40 nylon column to the side panel of the robot arm.

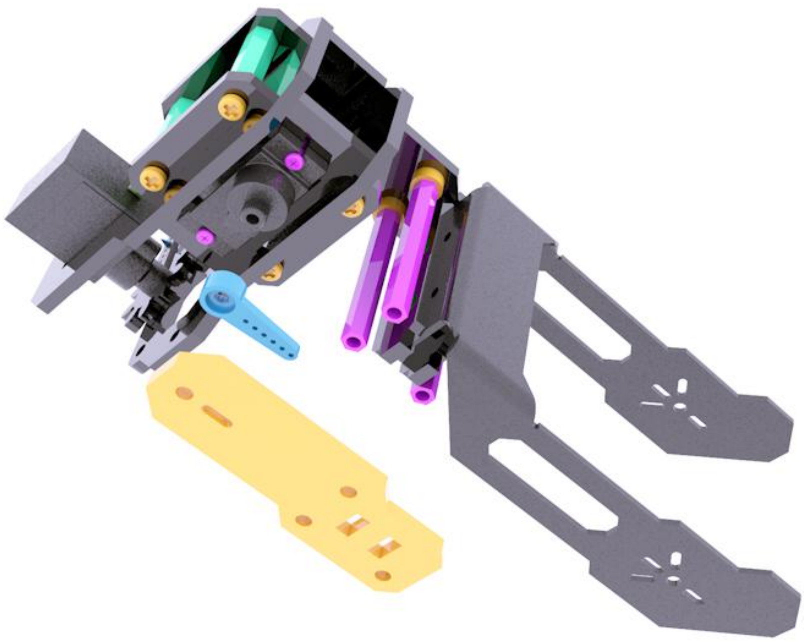


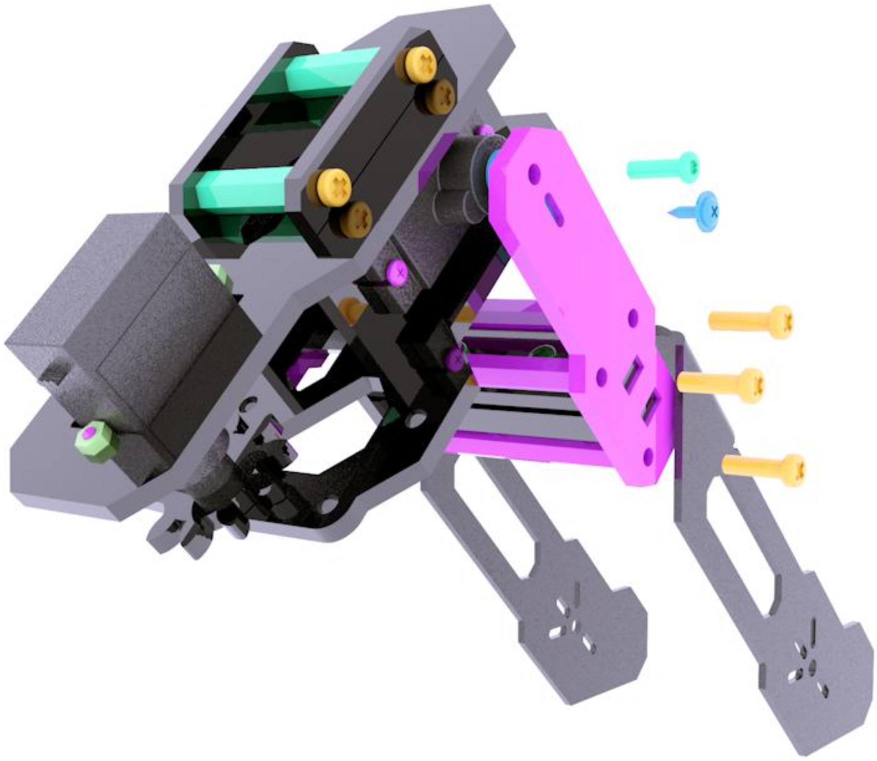


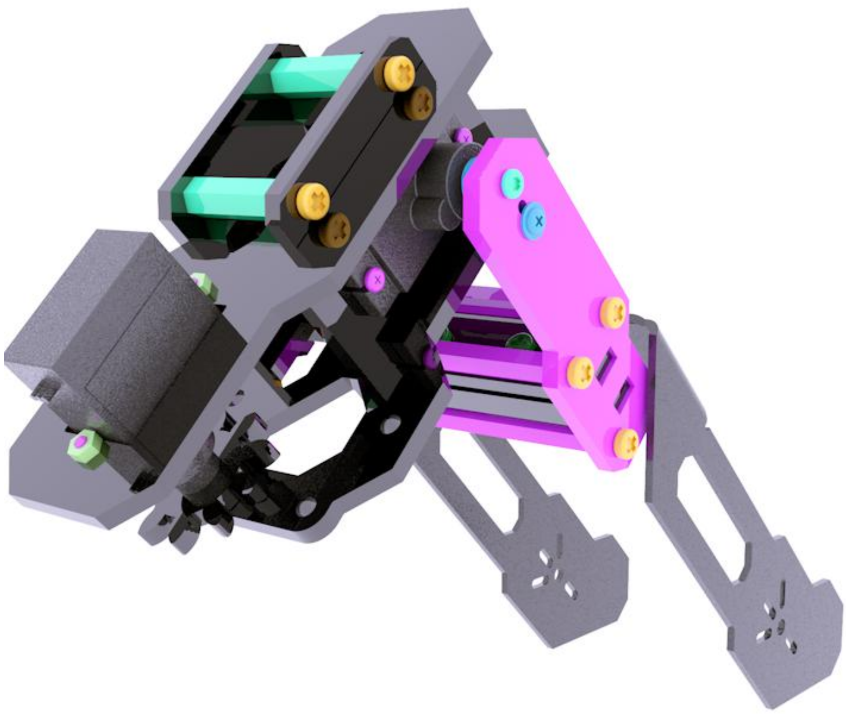


Use three M3X10 screws to fix the panel on the other side.

Use M1.7X6X6 self-tapping screws to fix the side panel and the servo arm, and use M2.5X8 screws to fix the servo to the side panel.

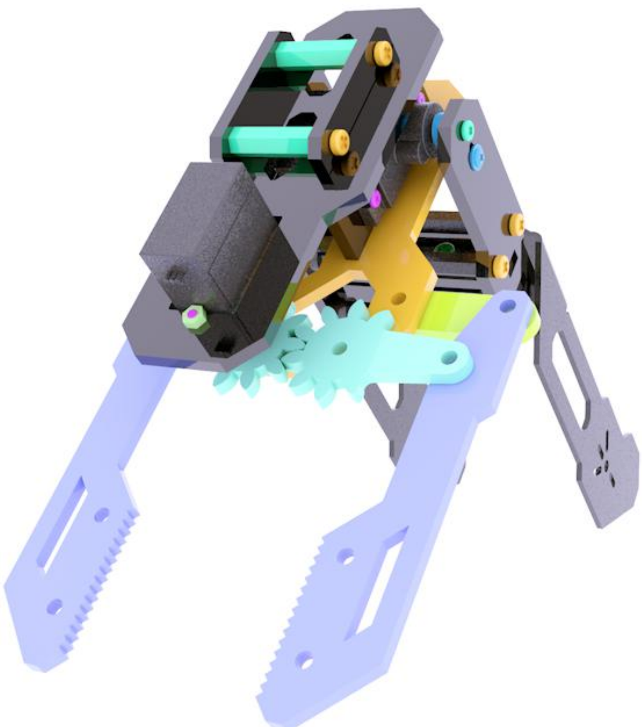


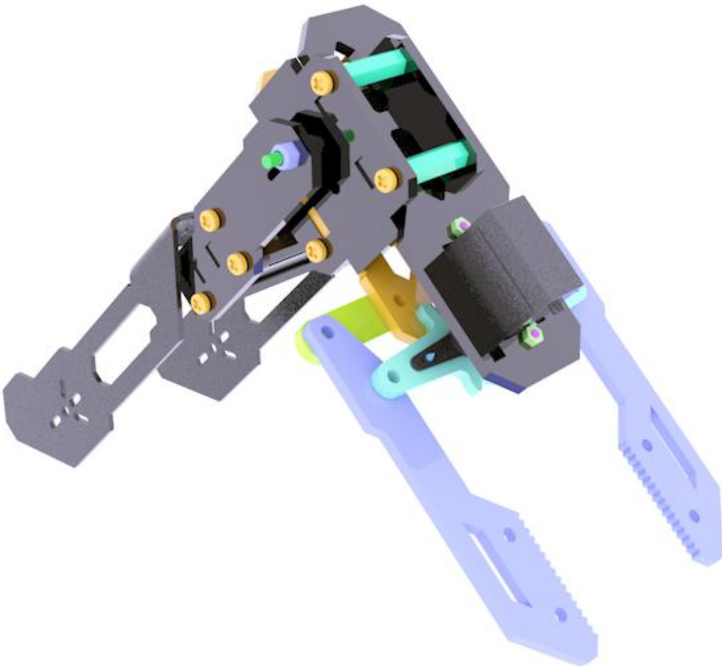


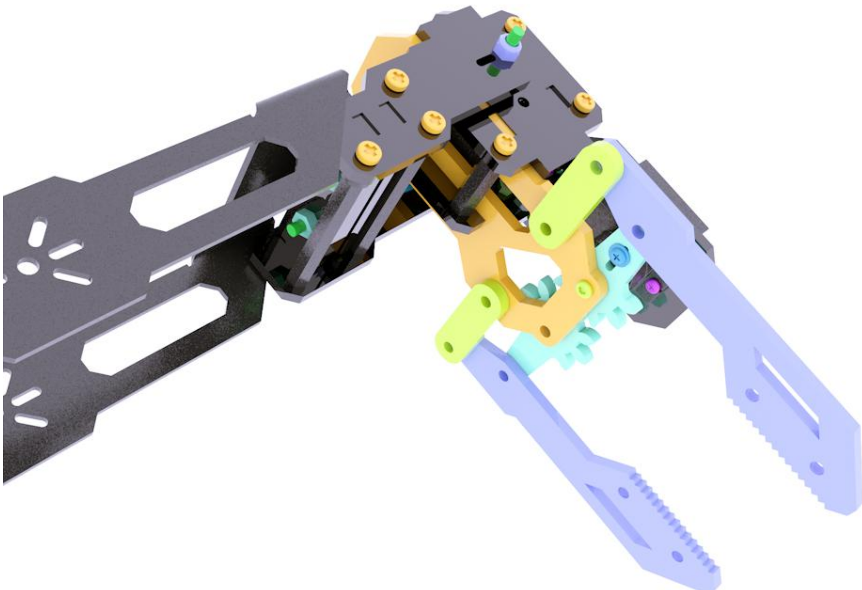


The assembled look is shown in the figure above.

Prepare the gear on the other side and the connecting rod parts for the chuck.

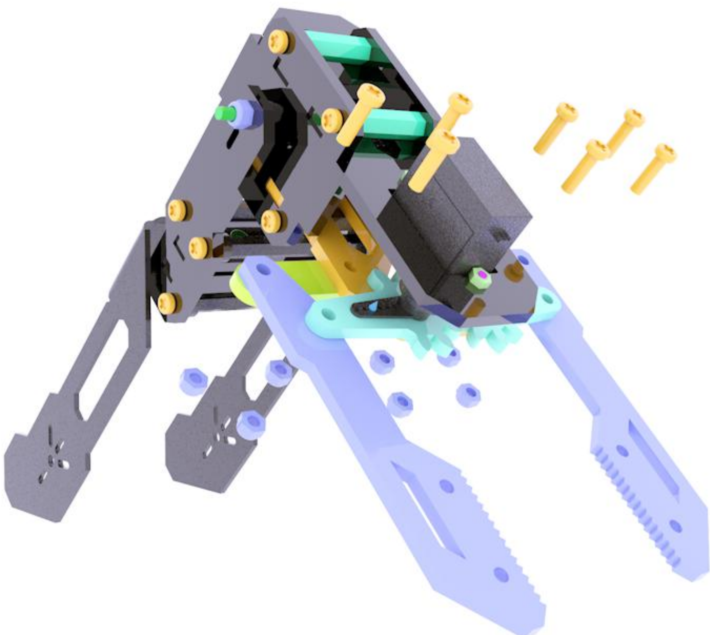


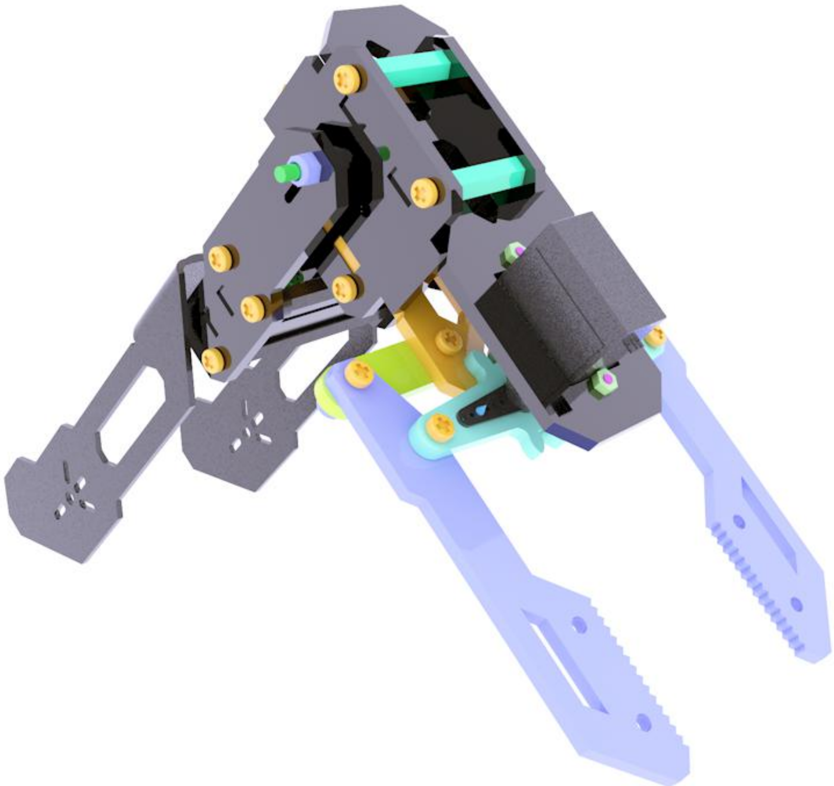




The installation method is shown in the figure above.

Use M3X10 screws and M3-LOCK locknuts to assemble the connecting rods.





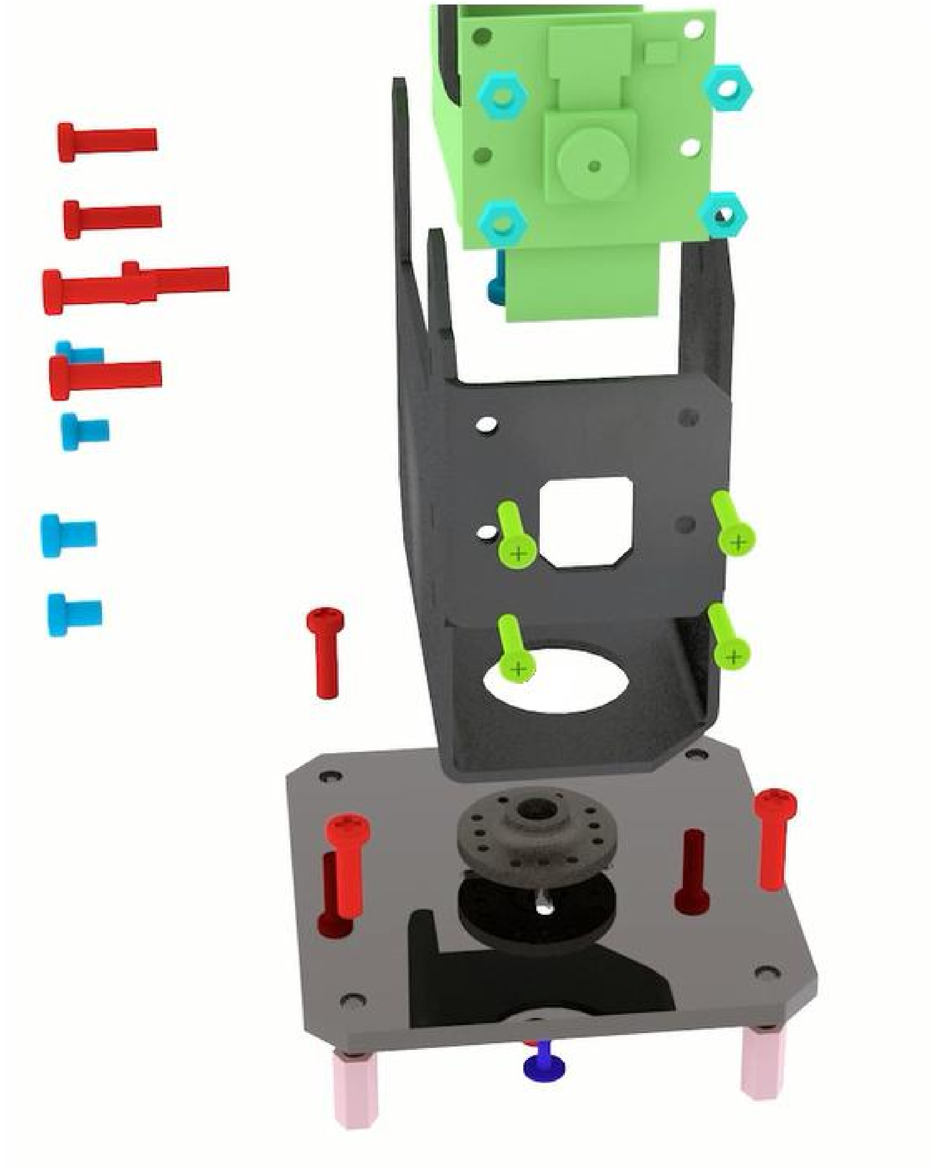
The robot arm after assembly is as shown in the figure above.

9.4.3 PTZ installation

note:

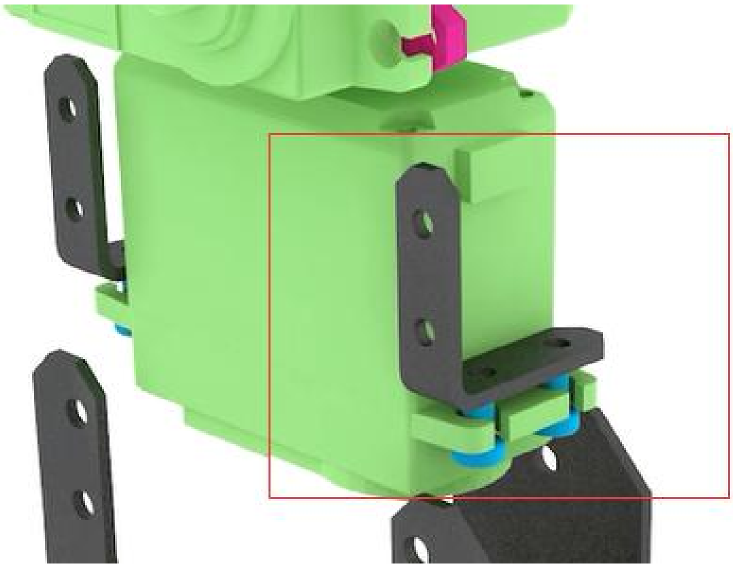
1. When installing the servo, the gear of the servo should be in the middle of the servo rotation range.
2. After installing the software in the Raspberry Pi, the Raspberry Pi will automatically run the robot product program when the Raspberry Pi is turned on. After powering on the servo, the drive board will automatically place the servo in the middle position.

Prepare 2 large servos (MG946R), a camera, and the following parts.

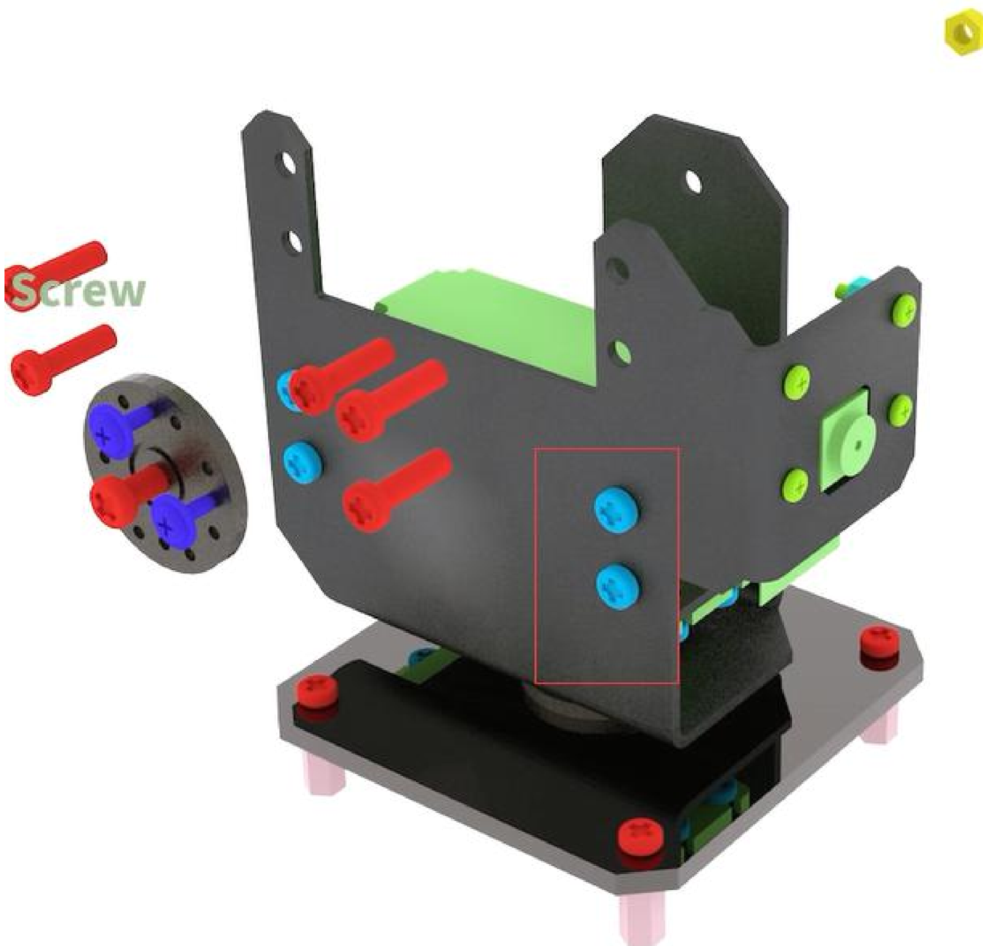


Use 8 M2 nuts on the front and back sides of the camera mounting hole, use M2*8 screws to pass through the aluminum alloy and the camera and fix them with M2 nuts.

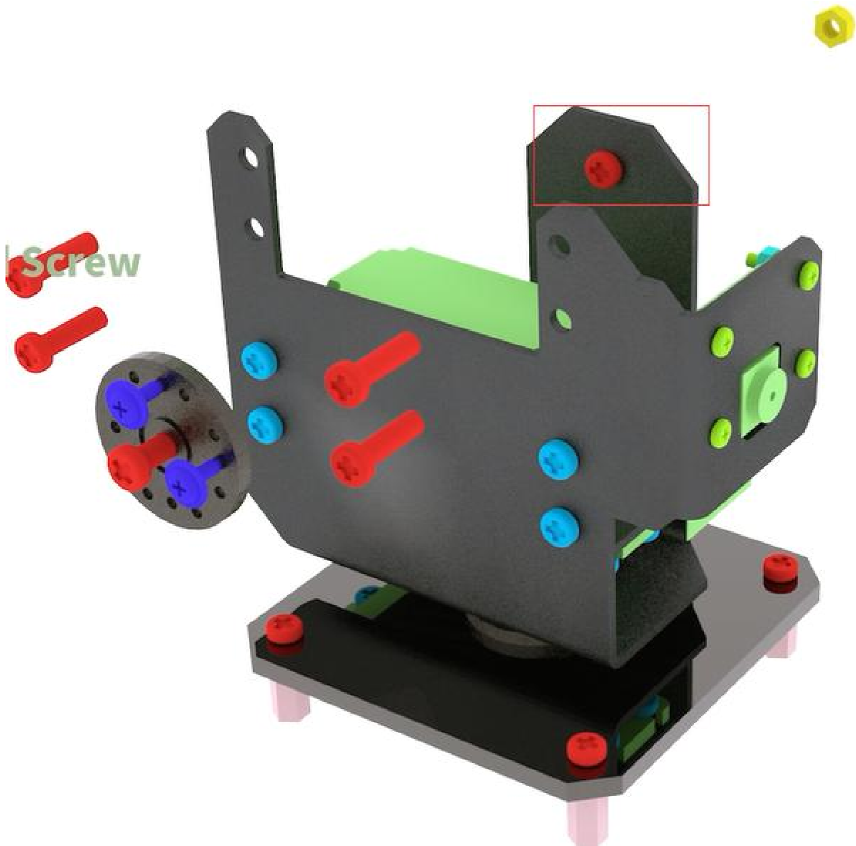
Use M3*4 screws to fix two "L"-shaped aluminum alloys.



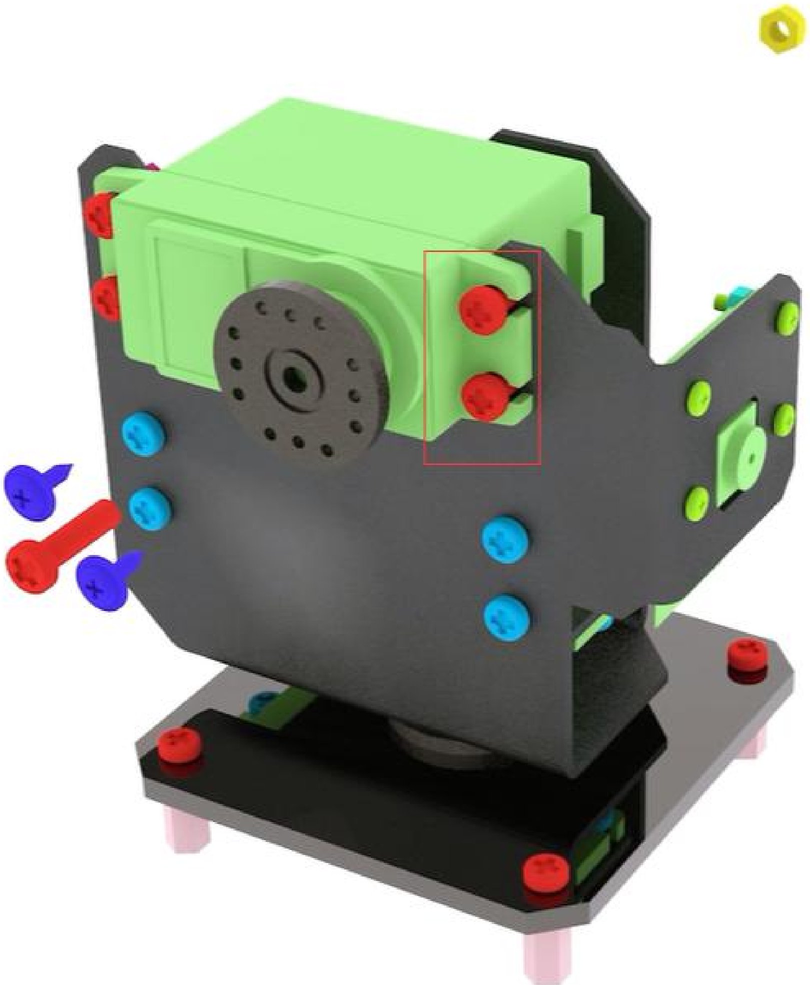
Use M3*4 screws to pass through the aluminum alloy and "L" aluminum alloy to fix the servo.



Install an M3*10 screw, no need to fix, for the connection of the PTZ and the robot arm.



Use M3*10 screws to pass through the aluminum alloy and the servo and fix them with M3 nuts. Install the steering wheel.

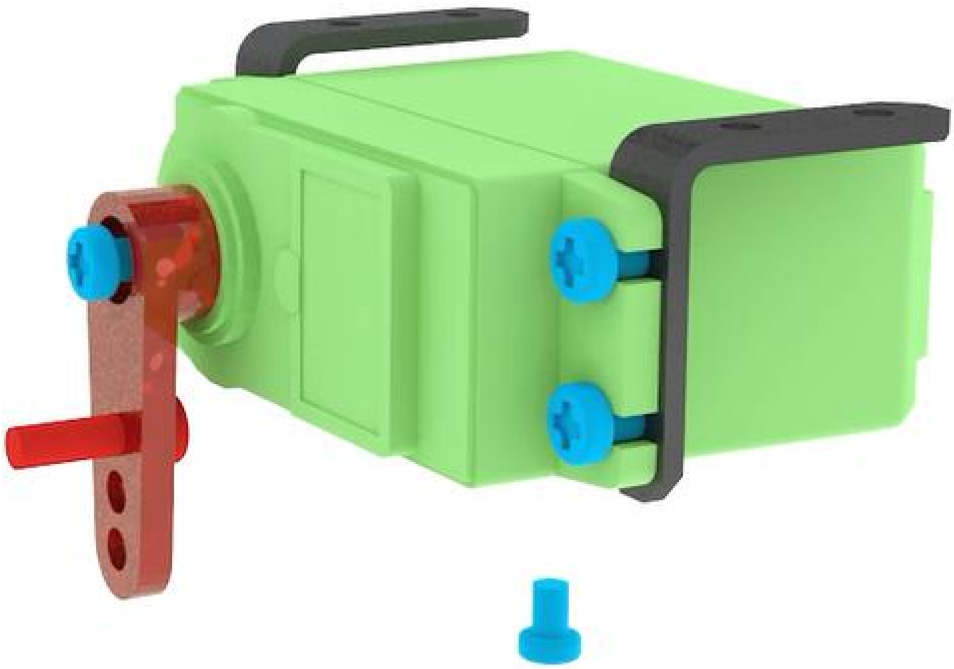


9.4.4 Body installation

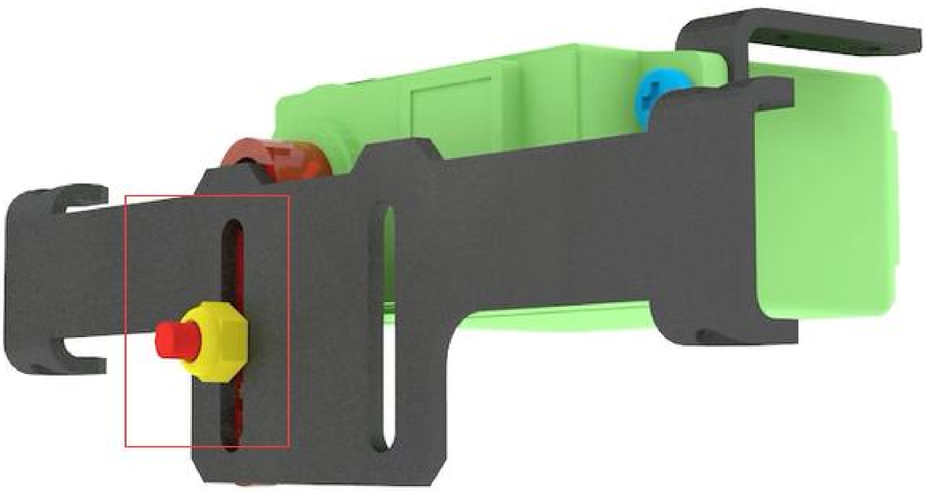
note:

1. When installing the rocker arm of the servo, the gear of it should be in the middle of the rotation range of the servo, and then install the servo as shown in the figure below.
2. After installing the software in the Raspberry Pi, the Raspberry Pi will automatically run the robot product program when the Raspberry Pi is turned on. After powering on the servo, the drive board will automatically place the servo in the middle position.

Use M3*4 screws to fix the "L" type aluminum alloy on the servo, install the rocker arm, fix it with M3*4 screws, and use M3*10 screws to pass through the rocker arm. (Due to the issue of the production version of the servo, if the M3*4 screw cannot fix the rocker arm, please use the M3*10 screw to fix it)

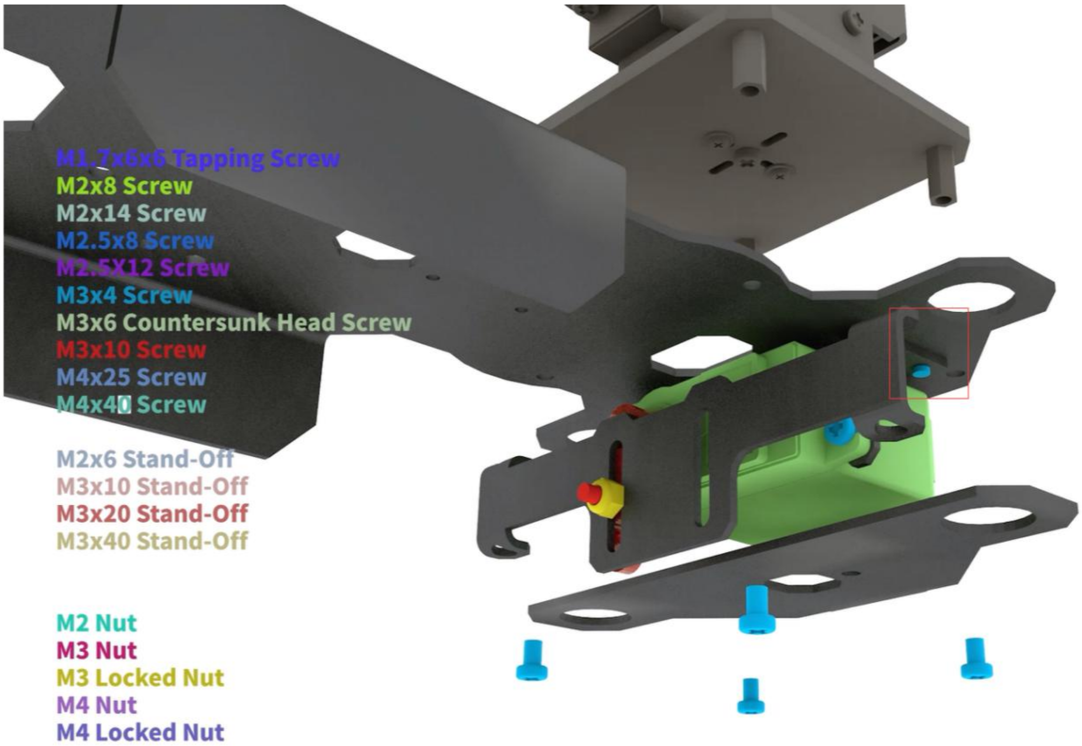


Install the aluminum alloy and fix it with M3 lock nuts (all lock nuts in this course do not need to be tightened).

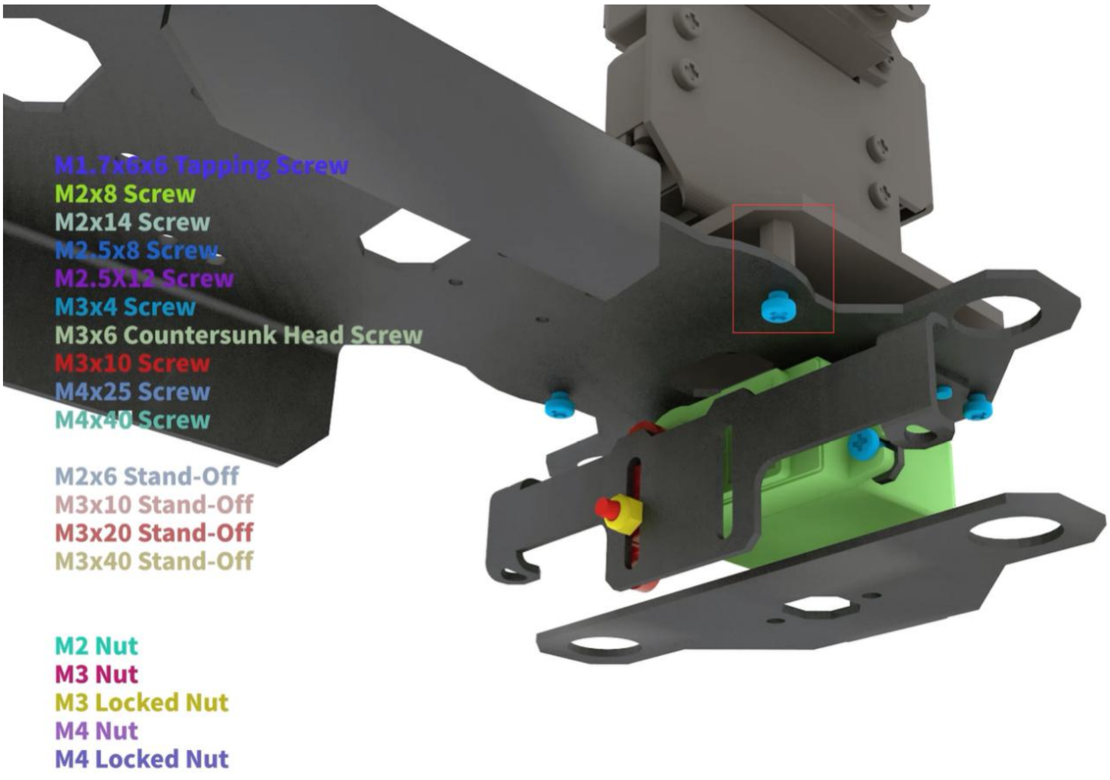


Prepare the aluminum alloy as shown below.

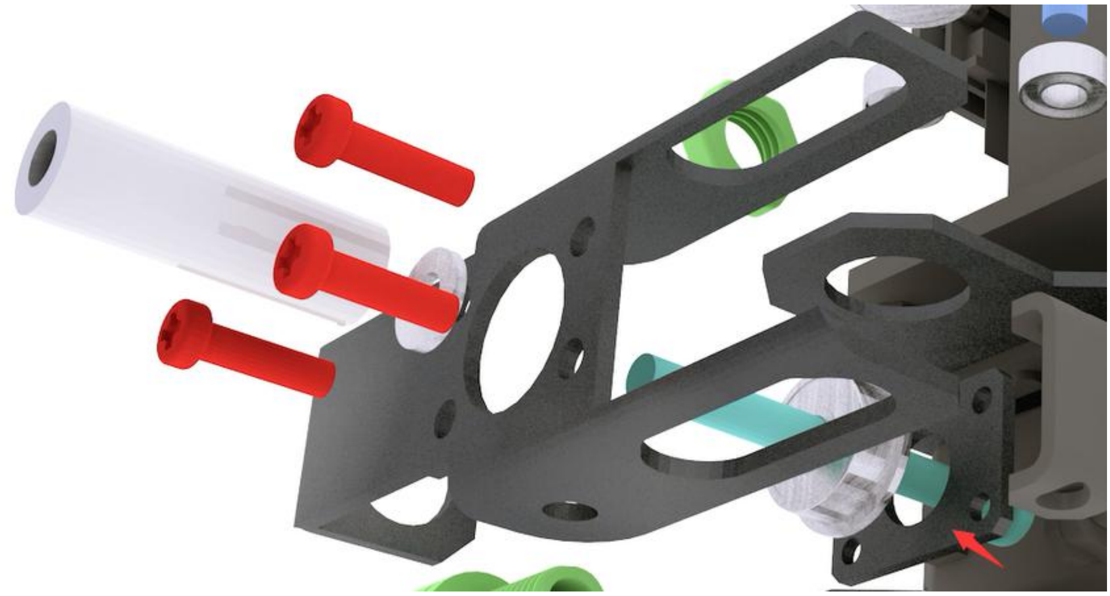
Install the aluminum alloy pictured above. Use 4 M3*4 screws to pass through the aluminum alloy and the "L" aluminum alloy on the servo to fix the servo.



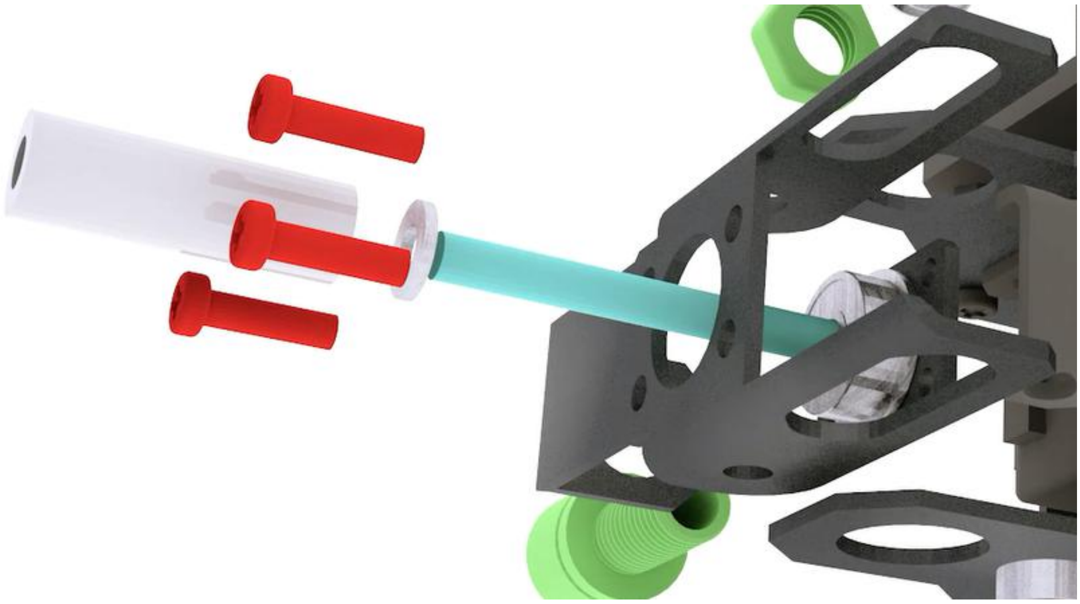
Install the PTZ. Use M3*4 screws to pass through the aluminum alloy and fix with M3*15 nylon posts.

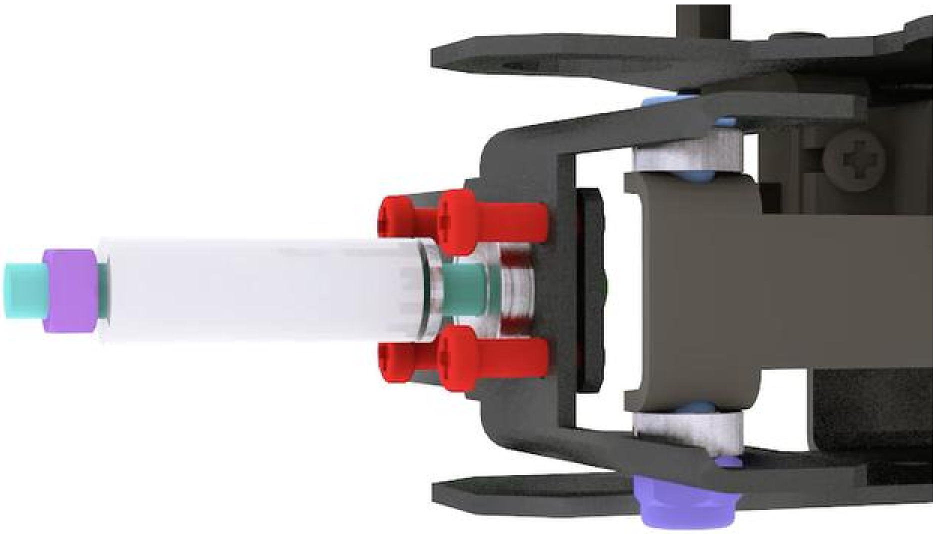


Prepare the aluminum alloy sheet (as shown by the arrow in the figure below), M4*40 screws, F624ZZ bearing (large), aluminum alloy, M4*8*1 gasket, plastic tube, and M4 nut.

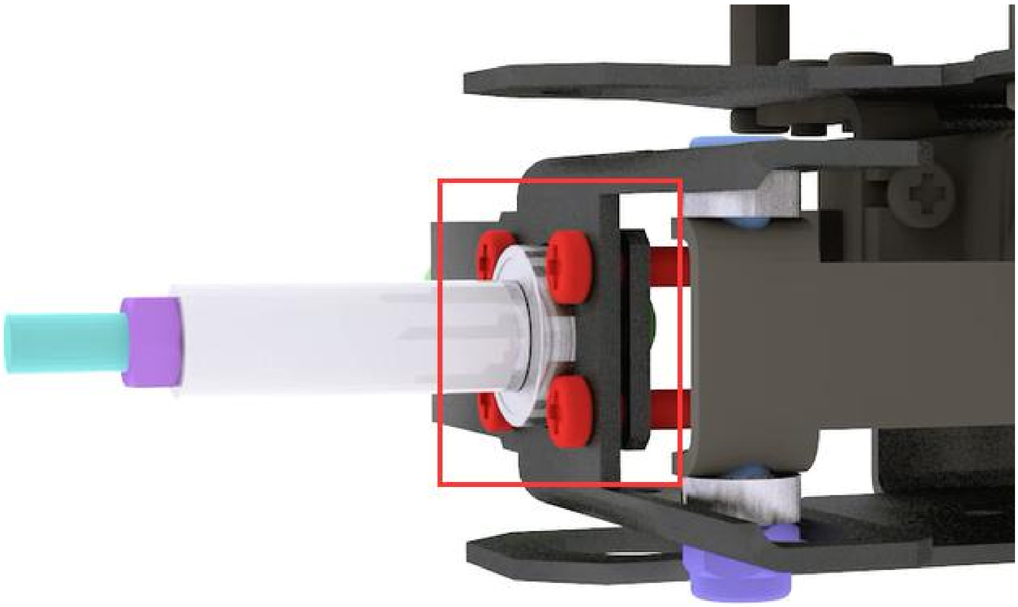


Use M4*40 screws to pass through the aluminum alloy sheet, F624ZZ bearing, aluminum alloy, gasket, plastic tube, and M4 nut from the inside to the outside. As shown below.

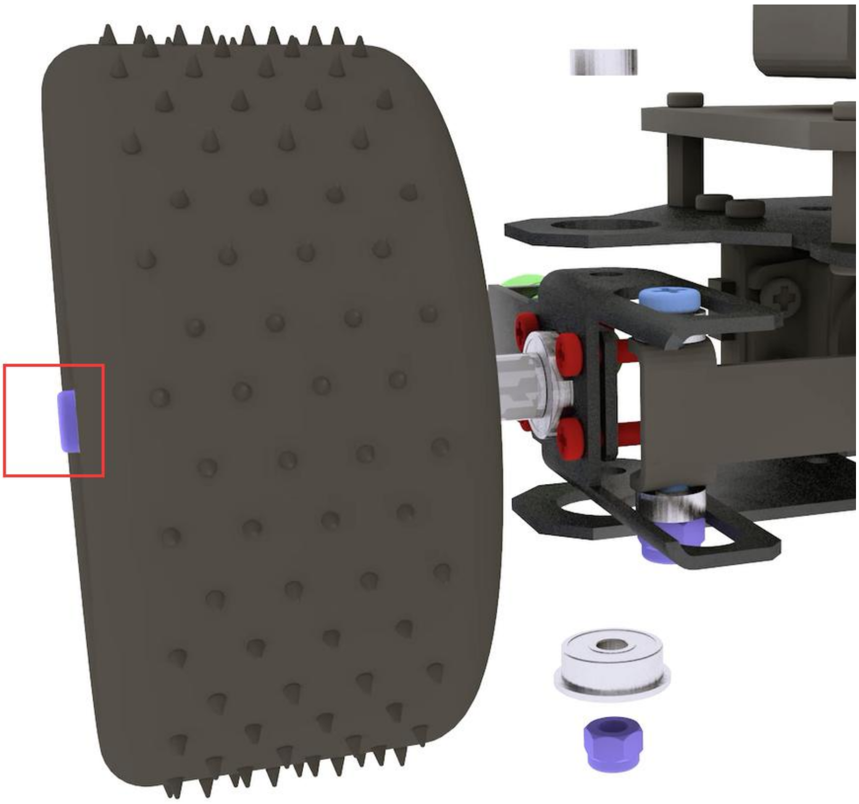




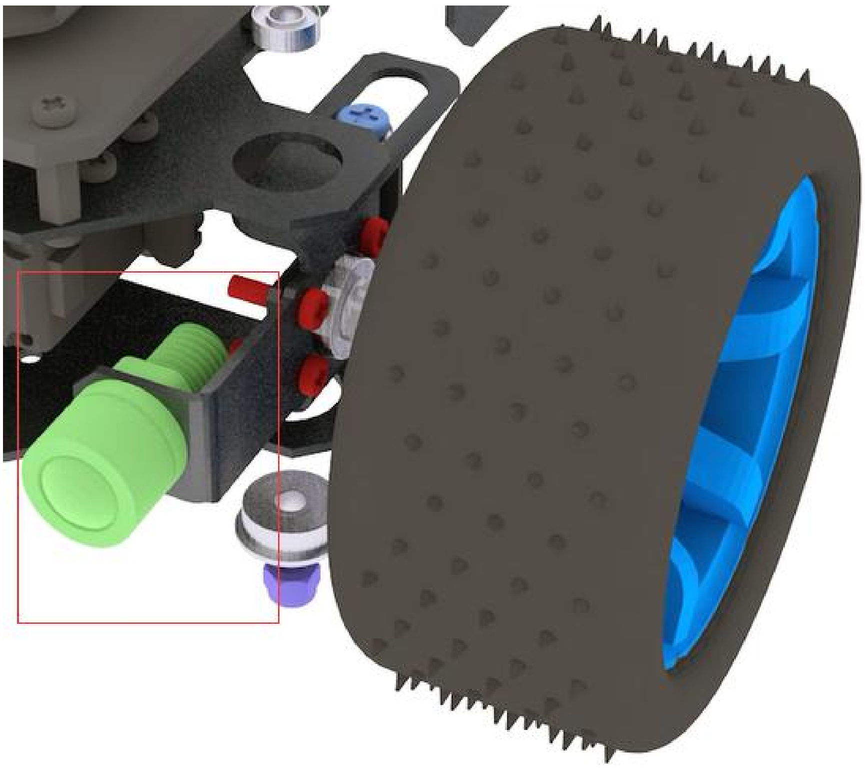
Use M3*10 screws to fix aluminum alloy and aluminum alloy sheet.



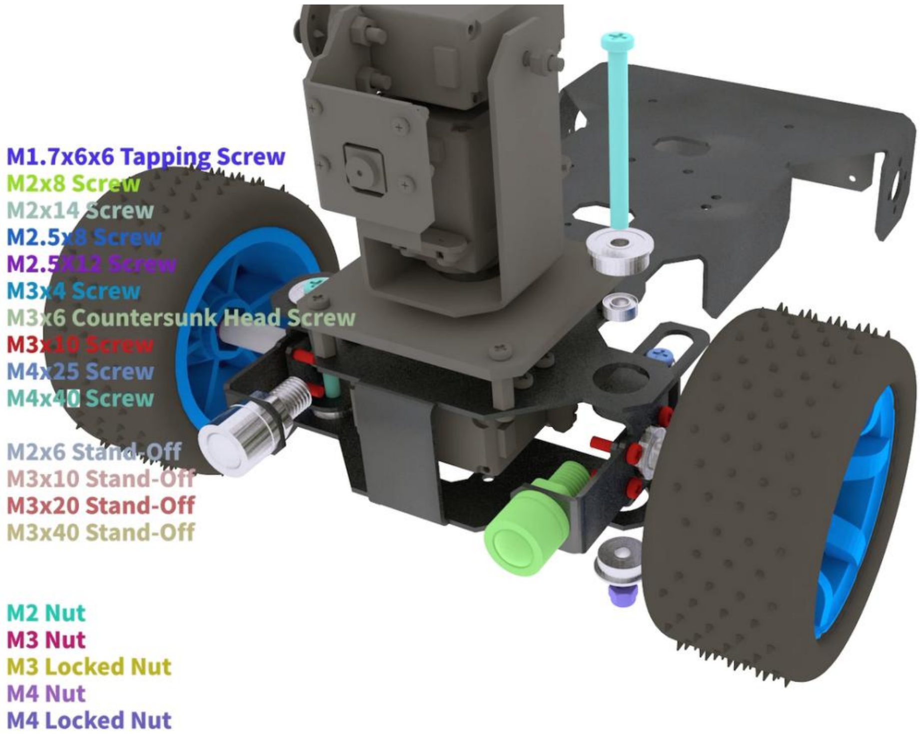
Install the tire and fix it with M4 locknut.



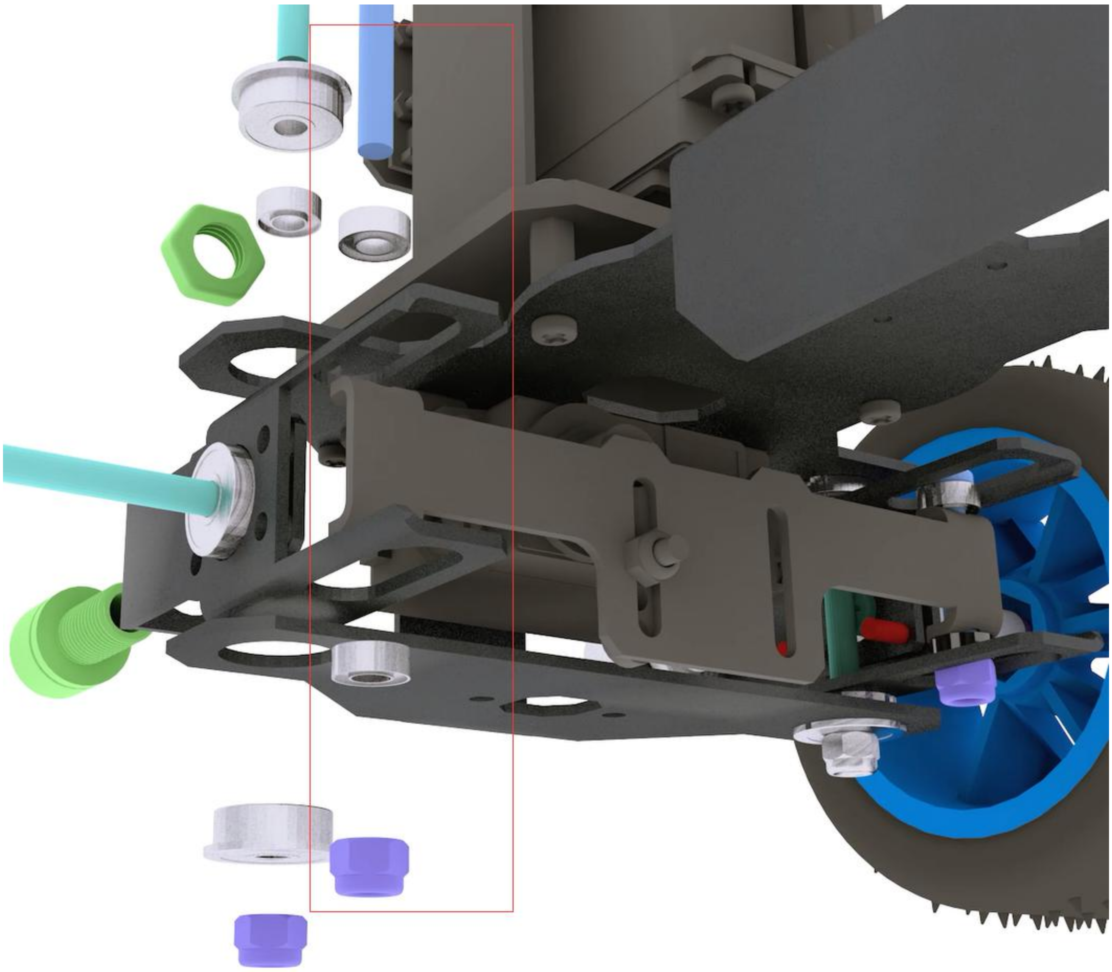
Install small LED lights.

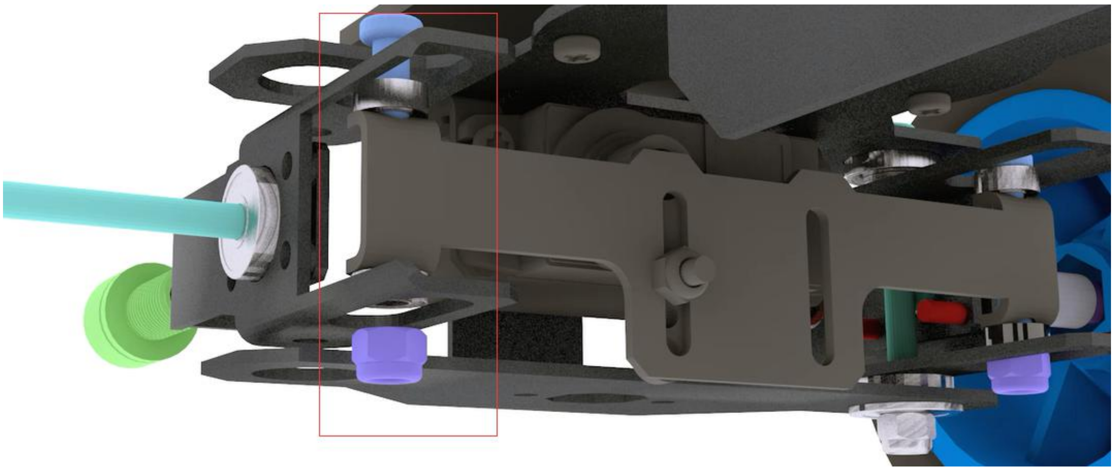


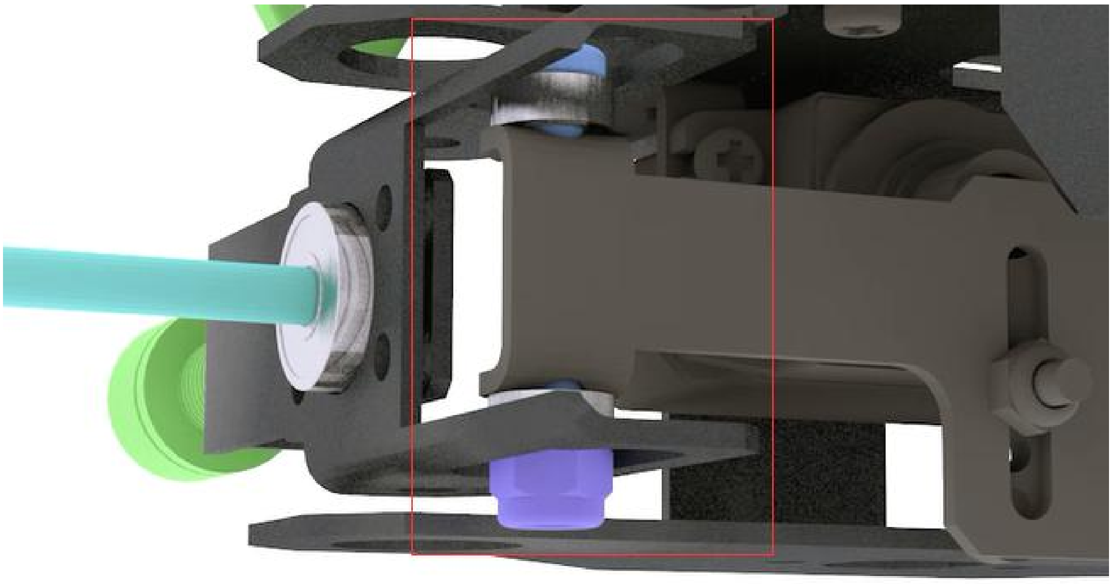
Repeat the above steps to assemble the structure on the other side.



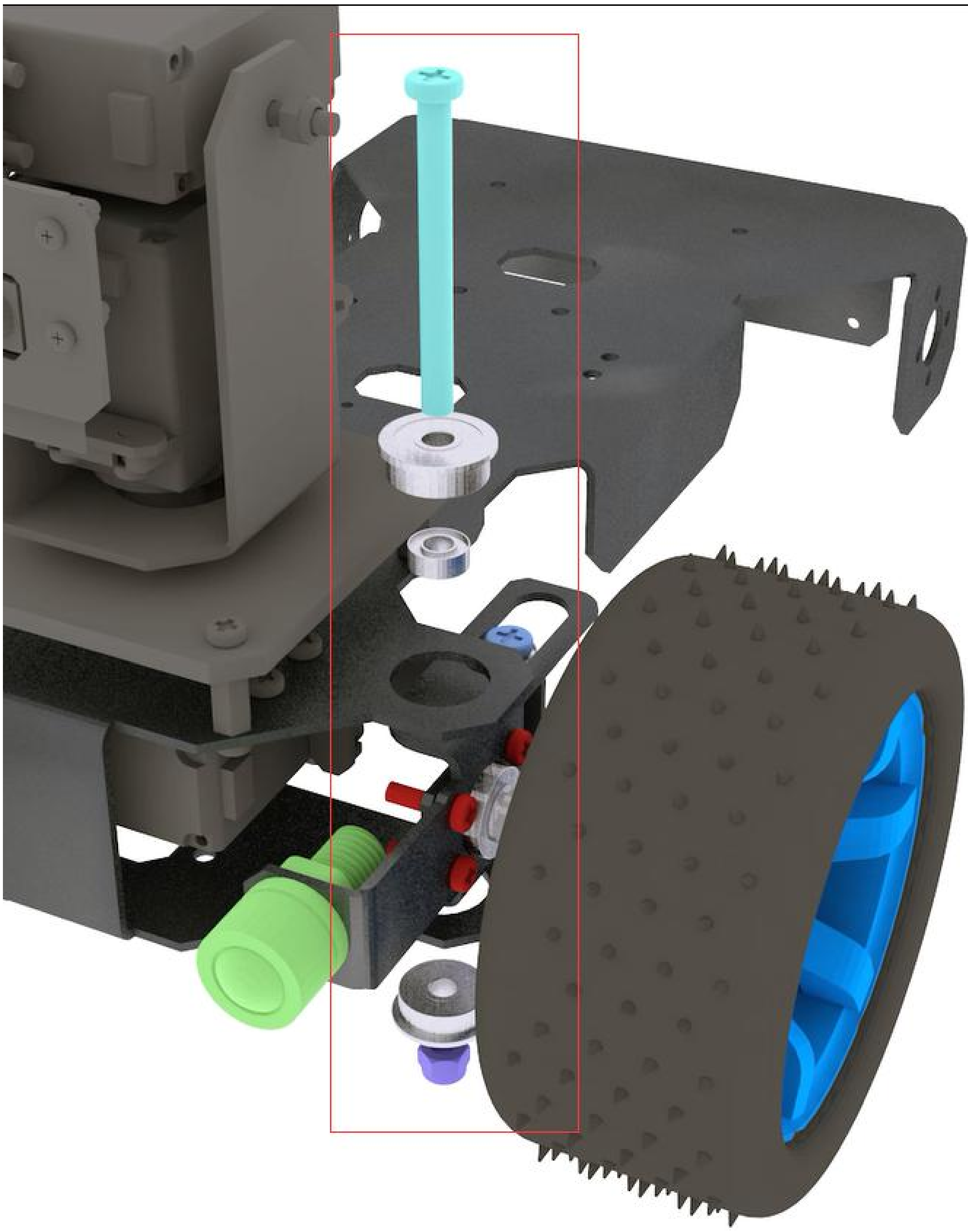
Use M4*40 screws to pass through these from top to bottom: MF84ZZ bearing (small), aluminum alloy (long large hole), aluminum alloy (small hole), aluminum alloy (small hole), aluminum alloy (long large hole) , MF84ZZ bearing (small), M4 nut. As shown below.

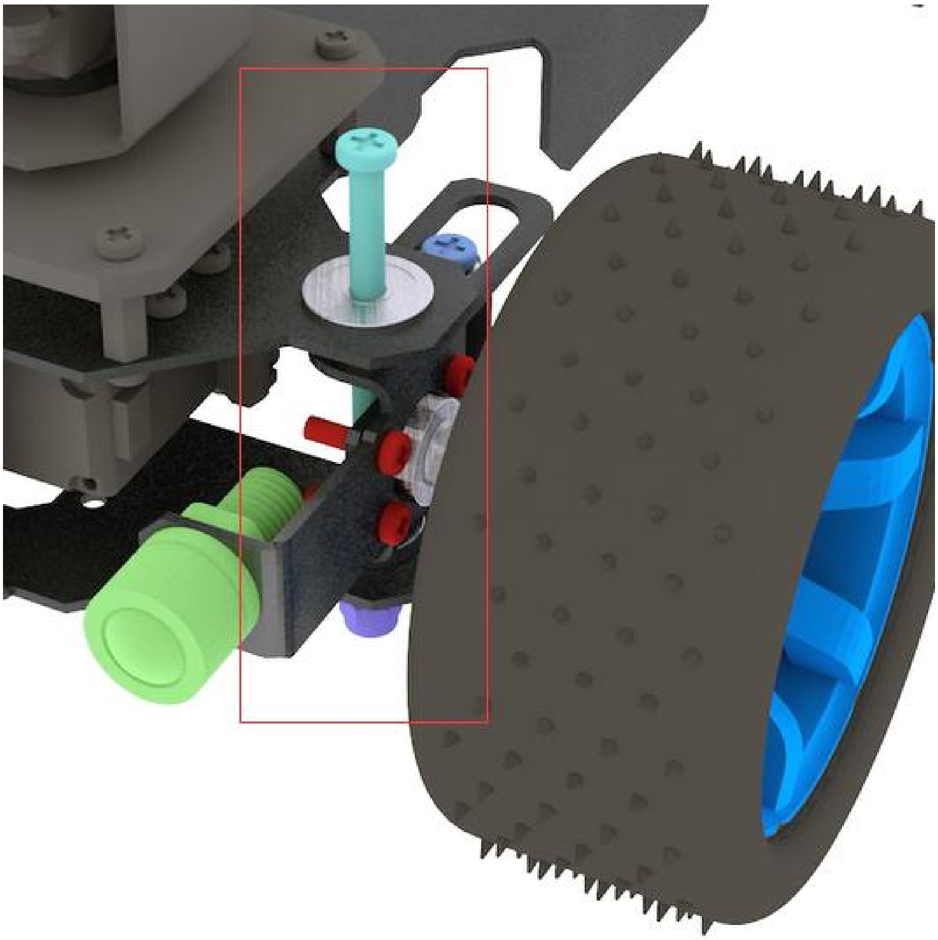


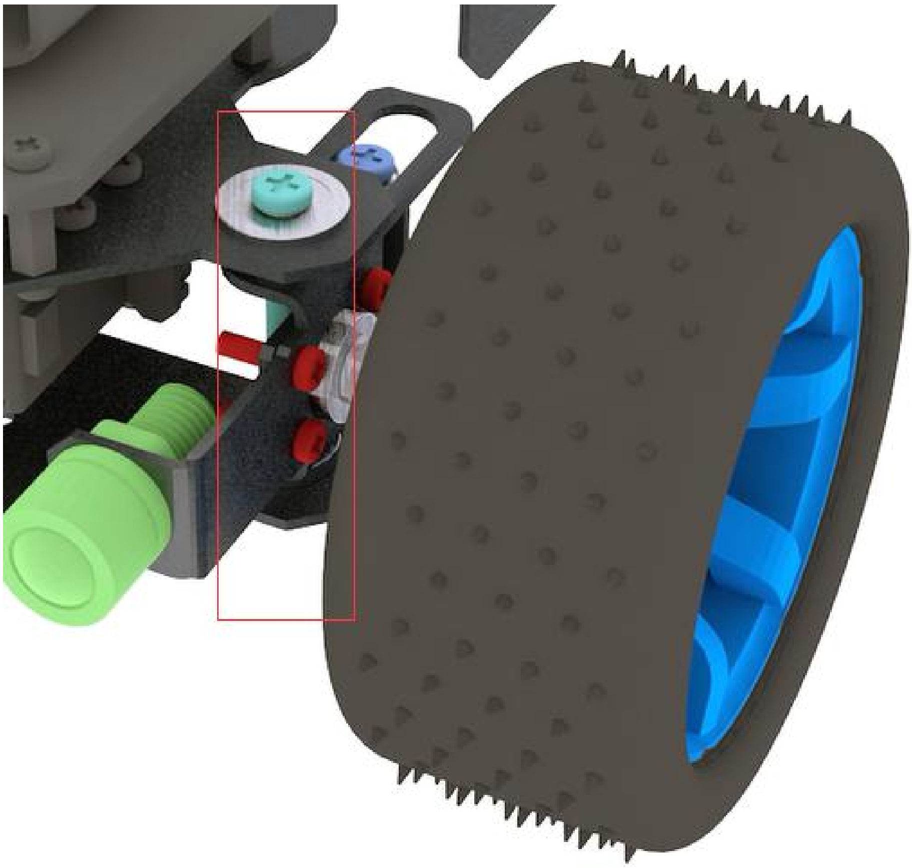




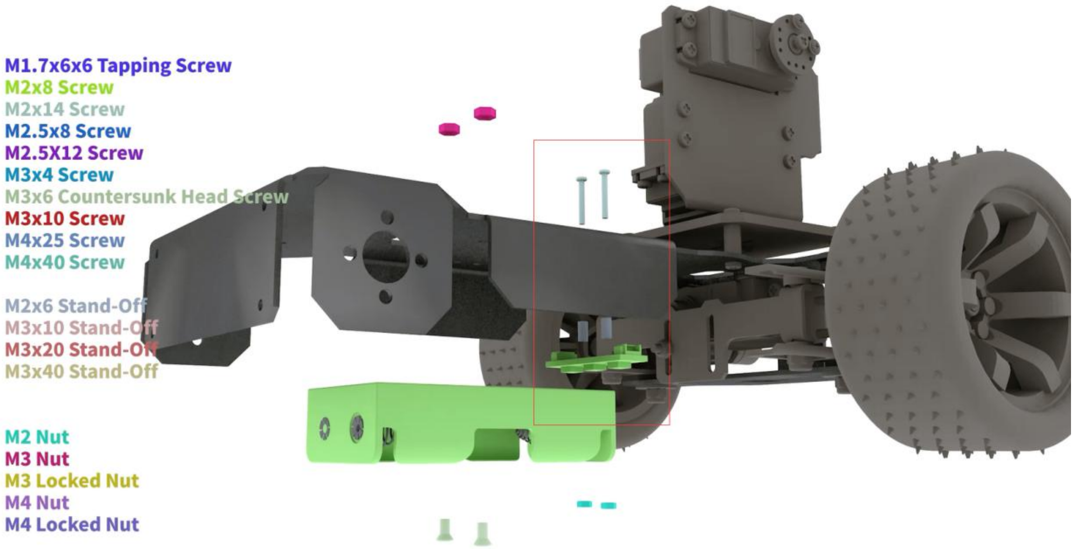
Use M4*40 screws to pass through these from top to bottom: F624ZZ bearing (large), MF84ZZ bearing (small), aluminum alloy (large hole), aluminum alloy (small hole), aluminum alloy (small hole), aluminum alloy (large hole), F624ZZ bearing (large), M4 nut. As shown below.



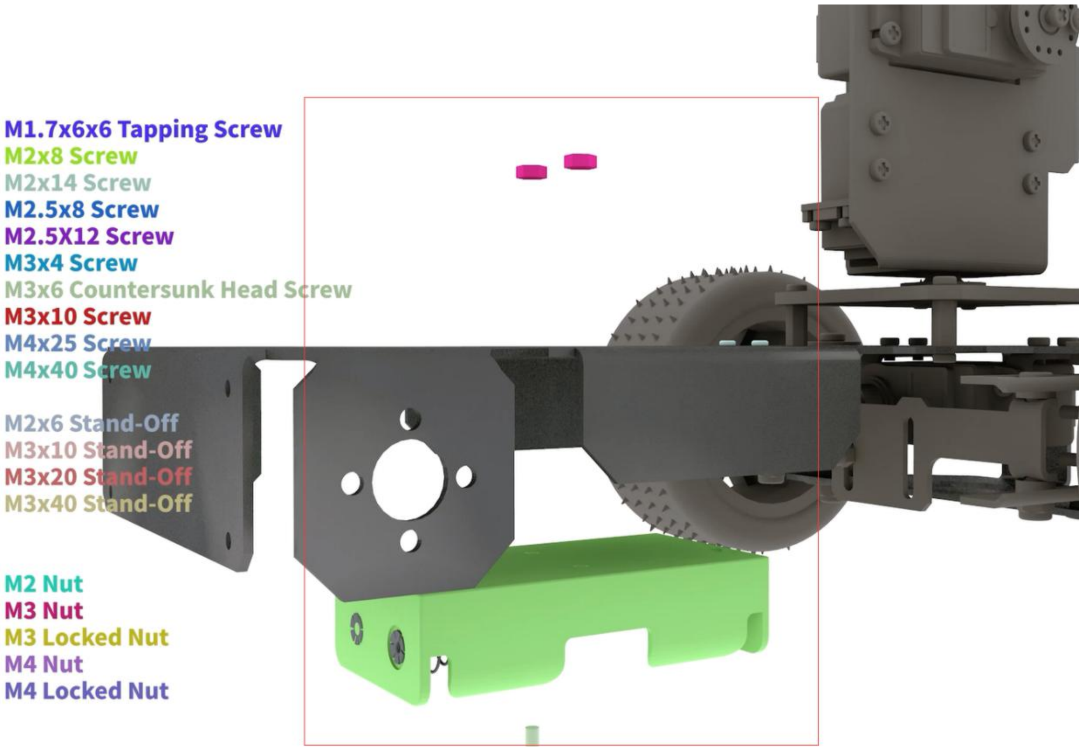




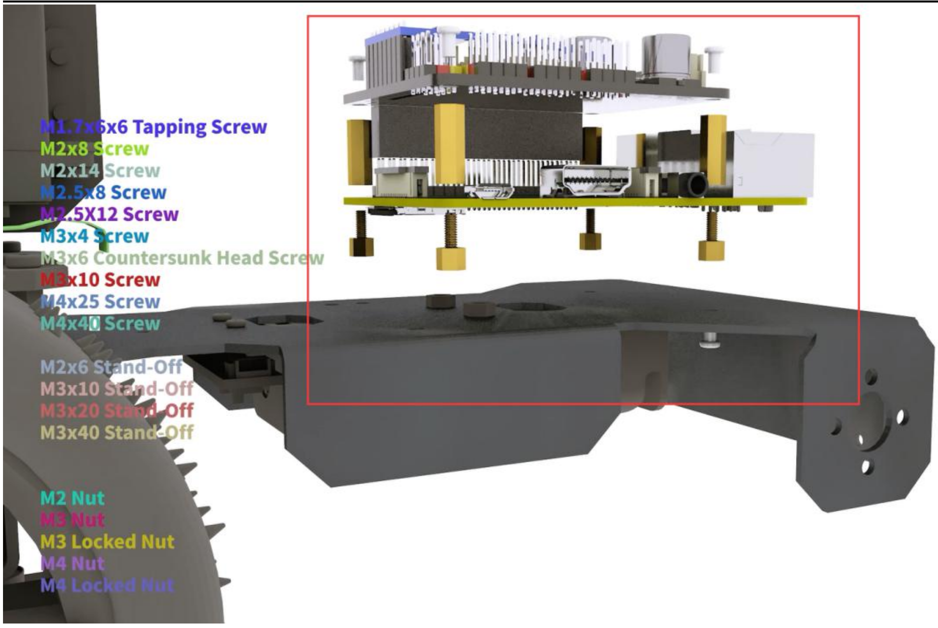
Install WS2812 LED lights. Use M2*14 screws to pass through the aluminum alloy, M2*6 copper column, ws2812 LED lights and fix them with M2 nuts.

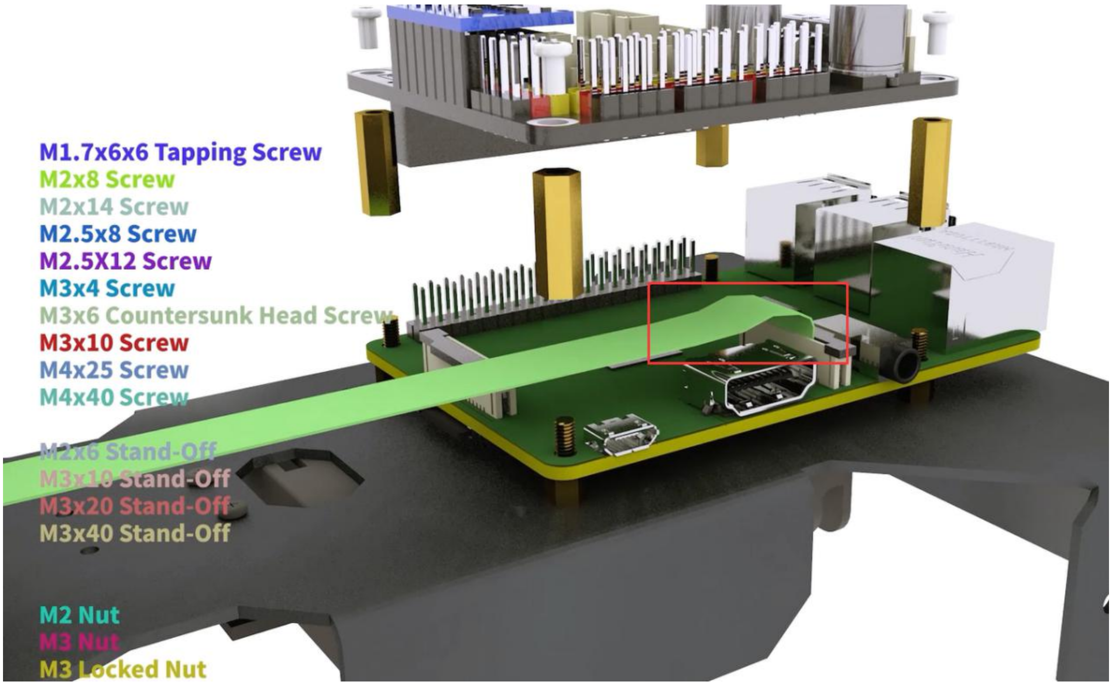


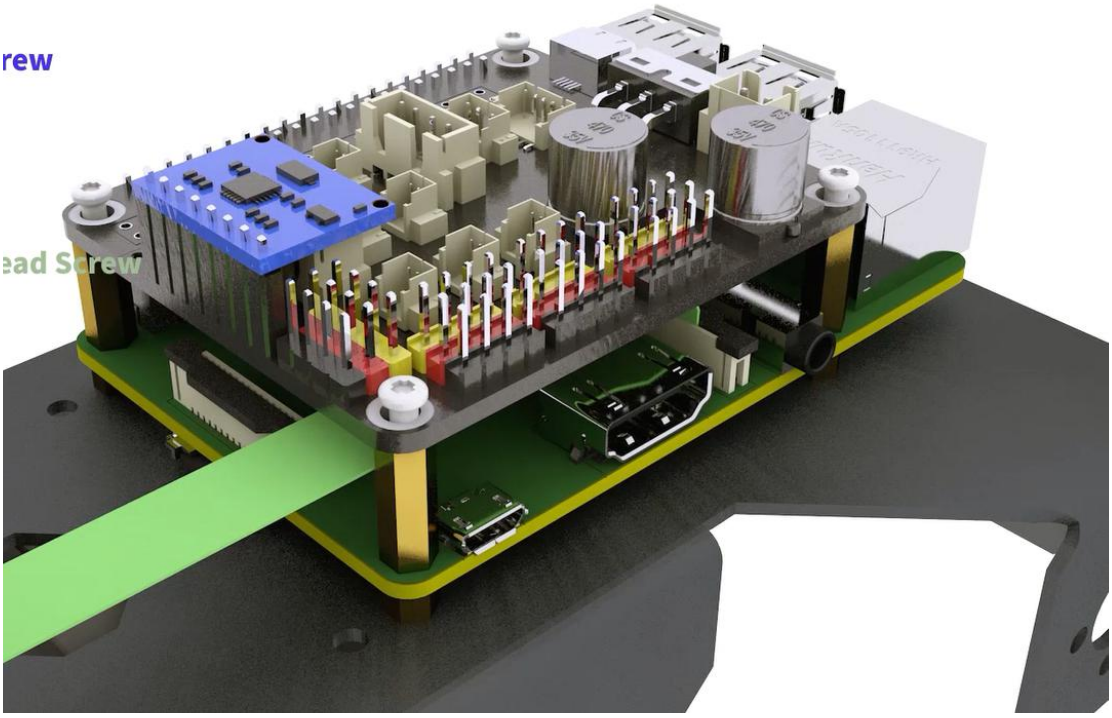
Install the battery holder. Use M3*6 countersunk screws to pass through the battery holder, aluminum alloy and fix them with M3 nuts.



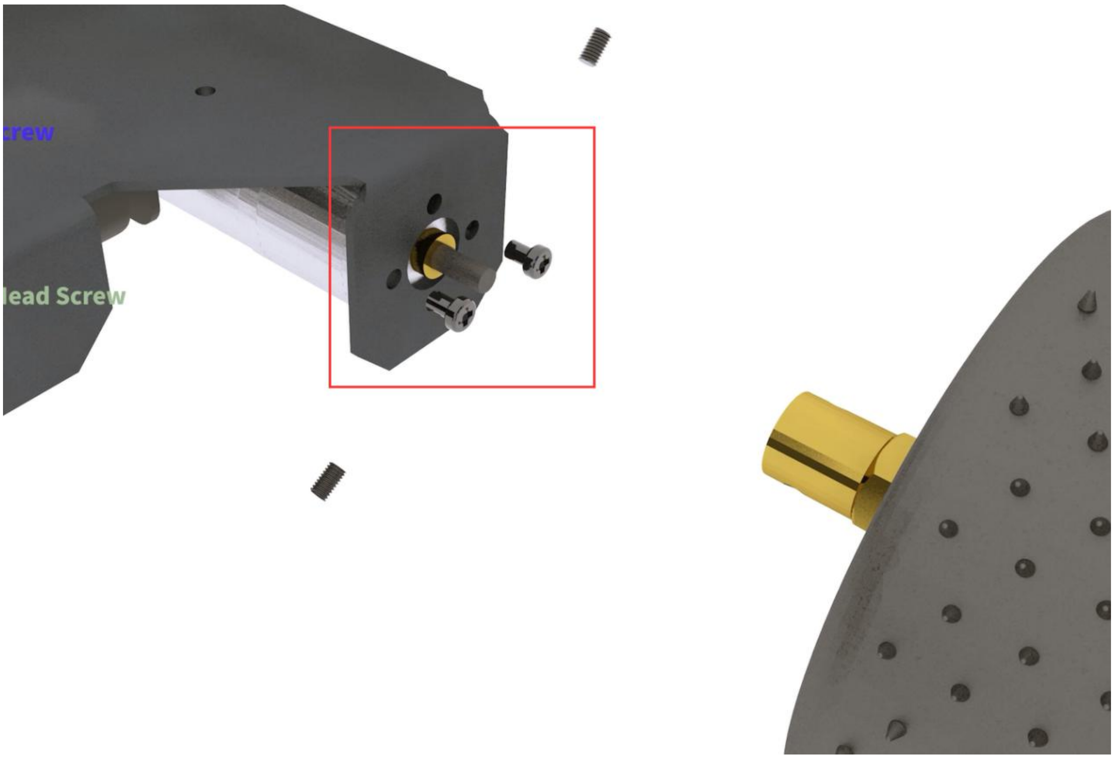
Install the Raspberry Pi and Robot HAT driver board. Use M2.5*4 screws to penetrate the bottom of the aluminum alloy and fix them with M2.5*4+6 copper pillars. Fix the Raspberry Pi with M2.5*4+6, install the camera cable, use M2.5*14 copper pillars to place between the Raspberry Pi and the Robot HAT board,use M2.5*4 screws to fix the robot HAT board.


M1.7x6x6 Tapping Screw
M2x8 Screw
M2x14 Screw
M2.5x8 Screw
M2.5X12 Screw
M3x4 Screw
M3x6 Countersunk Head Screw
M3x10 Screw
M4x25 Screw
M4x40 Screw
M2x6 Stand-Off
M3x10 Stand-Off
M3x20 Stand-Off
M3x40 Stand-Off
M2 Nut
M3 Nut
M3 Locked Nut
M4 Nut
M4 Locked Nut

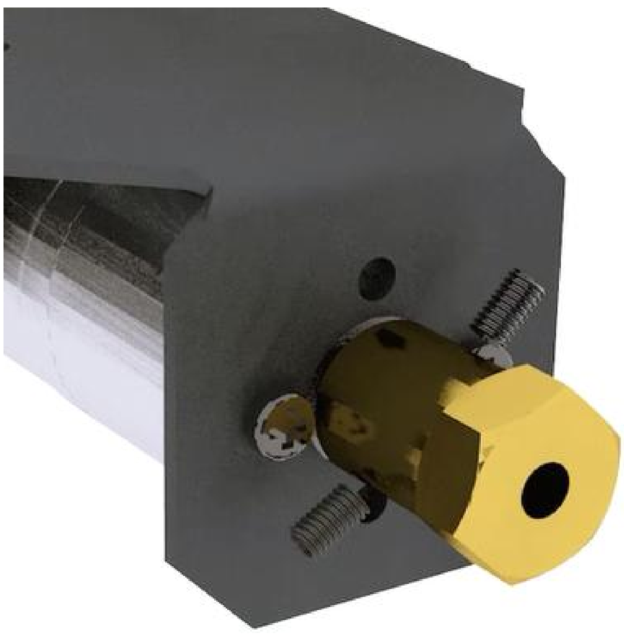

M1.7x6x6 Tapping Screw
M2x8 Screw
M2x14 Screw
M2.5x8 Screw
M2.5X12 Screw
M3x4 Screw
M3x6 Countersunk Head Screw
M3x10 Screw
M4x25 Screw
M4x40 Screw
M2x6 Stand-Off
M3x10 Stand-Off
M3x20 Stand-Off
M3x40 Stand-Off
M2 Nut
M3 Nut
M3 Locked Nut



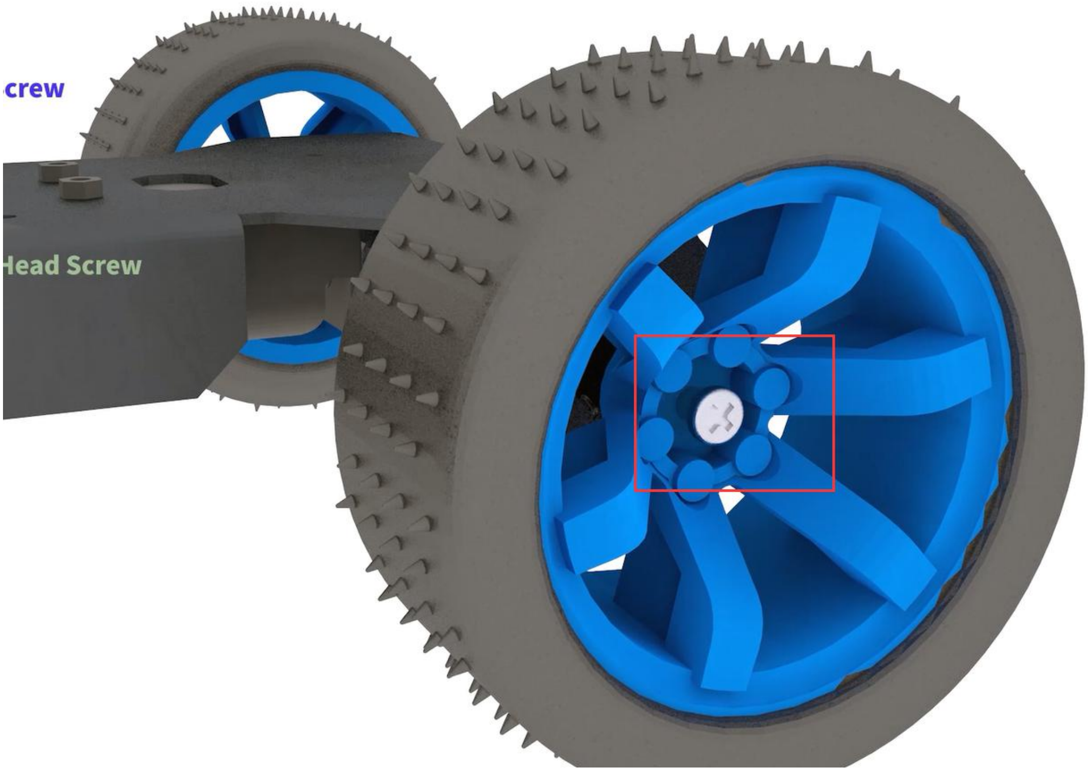
Install the rear wheel. Install the motor on the bottom of the aluminum alloy, and use M3*4 screws to pass through the aluminum alloy to fix the motor.



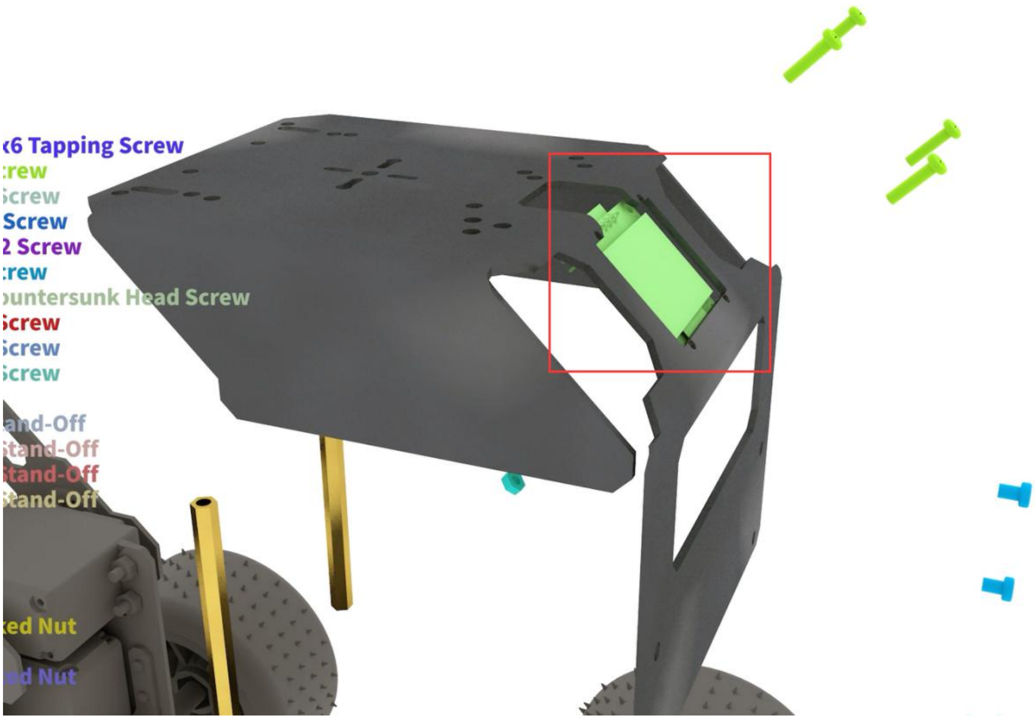
Install the inner 4mm coupling and fix it with screws.



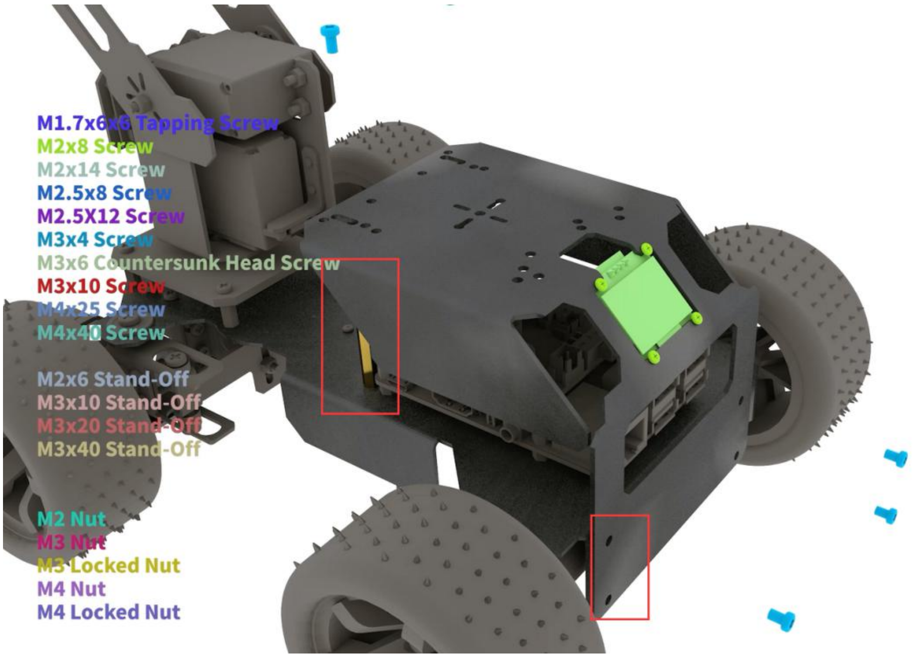
Install the rear wheel. Use M2.5*4 countersunk screws to fix.

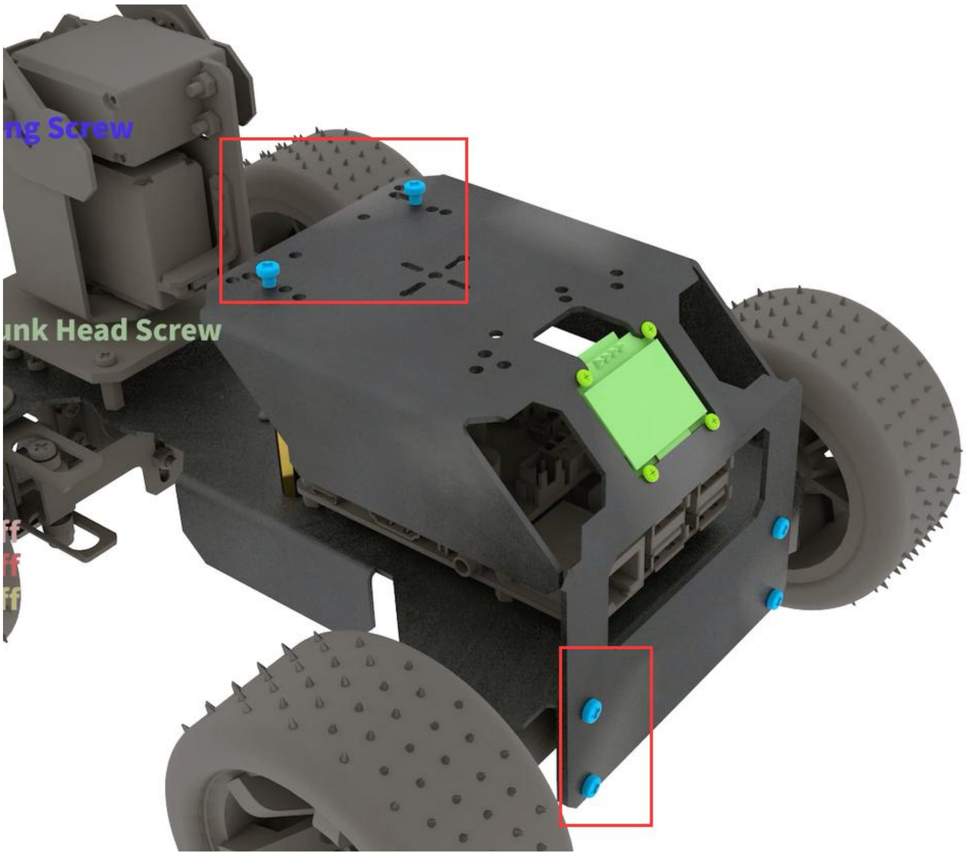


Install the OLED screen. Use M2*8 screws to pass through the aluminum alloy and OLED and fix them with M2 nuts.



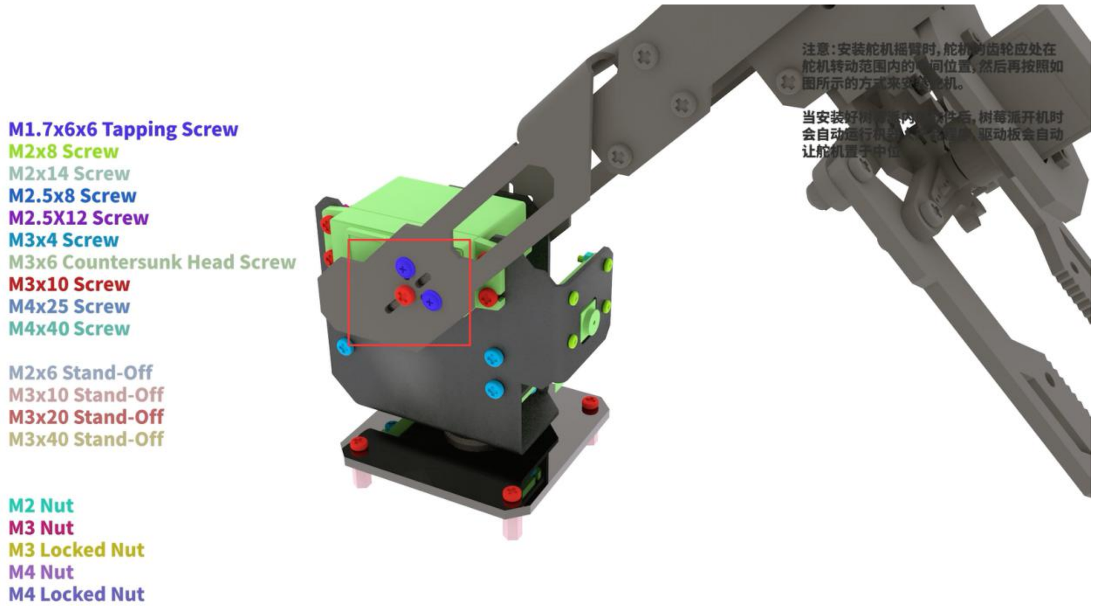
Use M3*4 screws to fix the M3*60 copper column with aluminum alloy, and use M3*4 screws to fix the aluminum alloy.





9.4.5 Installing the robotic arm

Use M3*10 screws to fix the position of the robot arm, and use two M1.7*6*6 self-tapping screws to fix the direction of the robot arm. Use M3 lock nut to fix the other end.

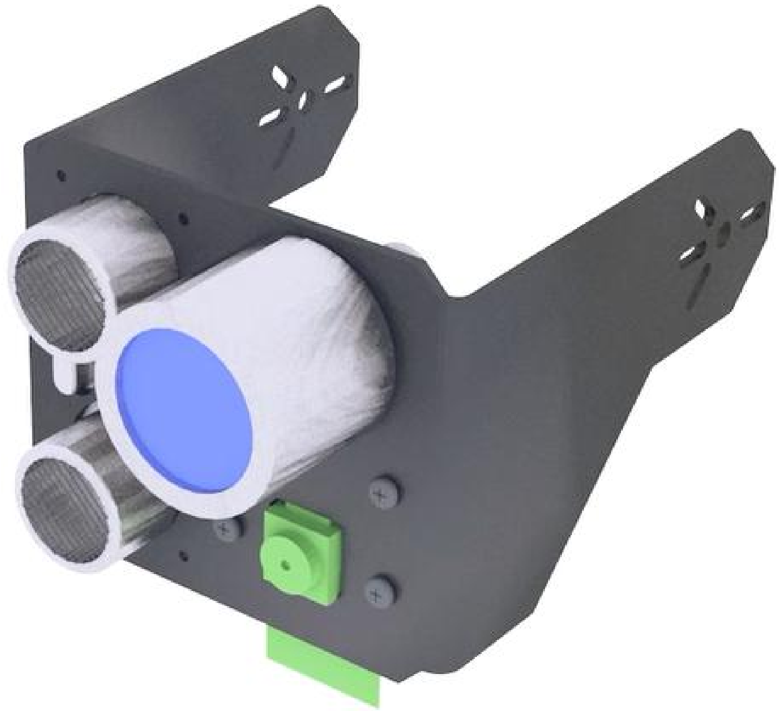


The installation is complete.

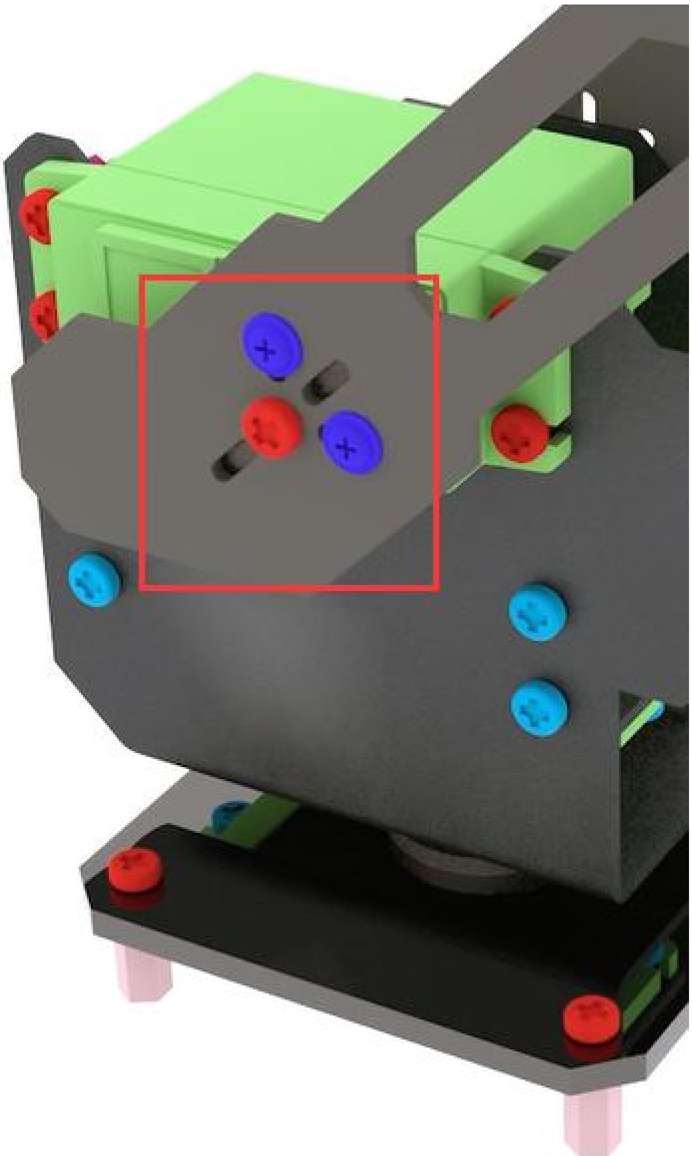
9.4.6 Install Ultrasonic Mode

When using the ultrasonic mode, the camera installation method is the same as the PTZ installation method. Because the shovel is equipped with a camera, when the ultrasonic mode is selected, the camera on the PTZ must be removed and installed in the designated position.

Use M1.4*6 to install the ultrasonic, and use M2*8 screws to pass through the aluminum alloy, M2 nut, camera, and M2 nut in turn (8 M2 nuts are required to install the camera). Install large LED lights.



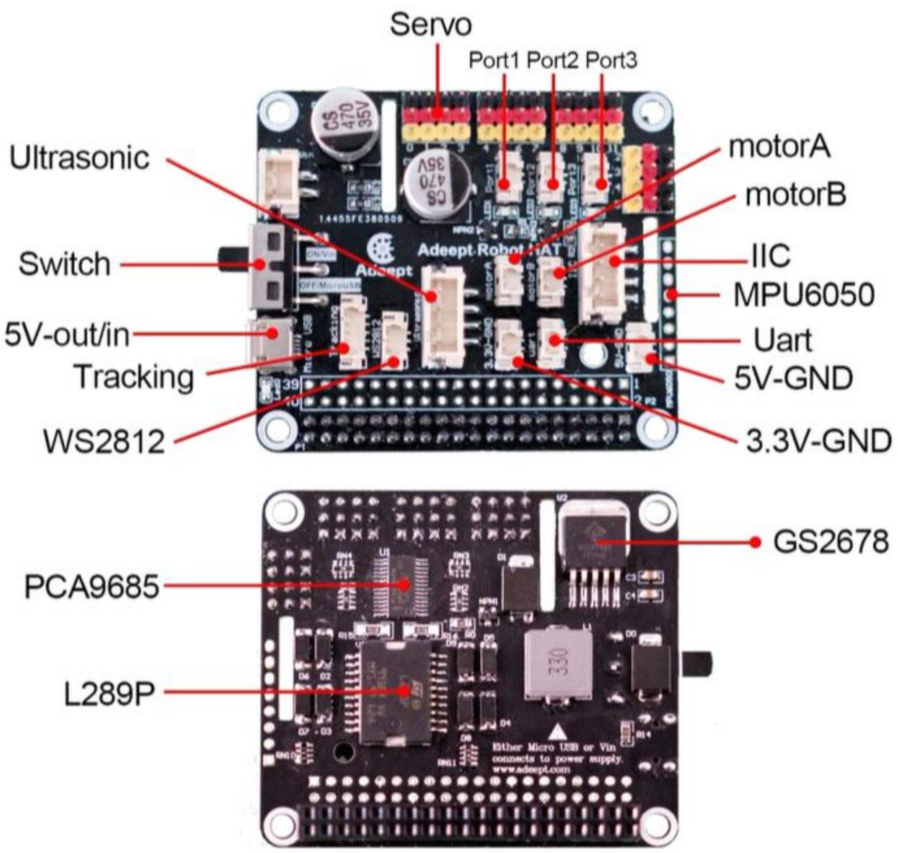
Install ultrasonic mode. It is the same as the method of installing the robot arm. Use M3*10 screws to fix the position of the robot arm, and use two M1.7*6*6 self-tapping screws to fix the direction of the robot arm. Use M3 lock nut to fix the other end.



The installation is complete.

9.5 Circuit Wiring Diagram

Robot HAT board interface diagram



Wiring method

1. If you don't know how to connect the camera, you can refer to the official documentation of the Raspberry Pi camera.

2. The motor on the left is connected to the MOTOR-B port; the motor on the right is connected to the MOTOR-A port.

3. The bottom servo that controls the left and right steering of the PiCarPro car is connected to the port0 servo port of the drive board.

4. Control the servo that swings the PTZ assembly to the left and right, and connect to the Port 1 servo port of the drive board.

5. The servo that controls the robotic arm or ultrasonic swing up and down on the PTZ is connected to the Port 2 servo port.

6. The servo that controls the up and down movement of the chuck is connected to the

Port 3 servo port.

7. The servo that controls the clamping action of the chuck is connected to the Port 4 servo port.

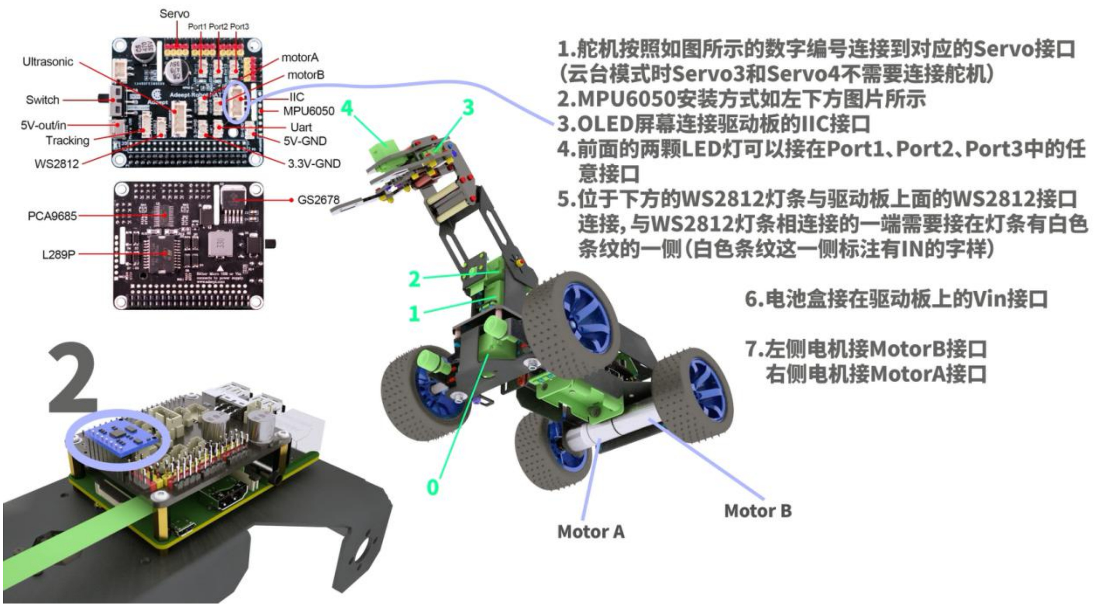
8. The ultrasonic module is connected to the Ultrasonic port with a 4pin wire. Never connect the ultrasonic to the IIC port, which will cause permanent damage to the ultrasonic module.

9. The WS2812 light bar is connected to the WS2812 port with a 3pin wire. Note here that the signal to control the light is sent by the Raspberry Pi and sent to the WS2812 light bar with the 3PIN wire with the driver board. It needs to be connected from the light bar with a white stripe. (The interface text on the back of this end is marked IN), connect from the other end of the light bar (the interface is marked OUT) and then connect to the input end of the next light bar with the 3PIN wire (It is painted with white stripes, and the interface is marked with IN).

10. The OLED screen is connected to the IIC port of the driver board with a 4PIN wire.

11. MPU6050 is inserted into the 8PIN DuPont port on the front of the driver board.

12. Connect the power supply to the VIN port.



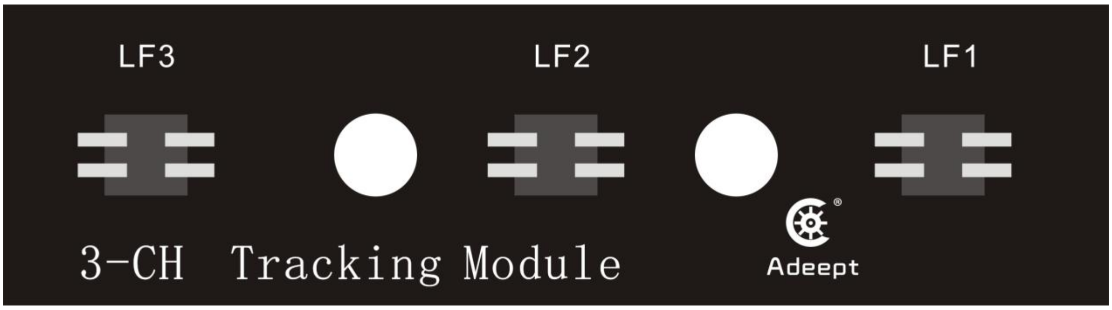
Lesson 10 Tracking Function

In this lesson, we will learn how to use tracking function of PiCarPro.

10.1 Tracking Module

Classic Car and Racing Car are driving on the white paper "road" with black lines. Because the black line and white paper have different reflection coefficients of light, the "road" can be judged according to the intensity of the received reflected light-black line . In the Tracking module, a more common detection method-infrared detection method is used.

Infrared detection method uses infrared rays to have different reflection properties on different colored physical surfaces. When the car is running, the infrared light is continuously emitted to the ground. When the infrared light meets the white ground, the diffuse emission occurs, and the reflected light is received by the receiving tube installed on the car;if it encounters a black line, the infrared light is absorbed, and the receiving tube on the car cannot receive the signal.

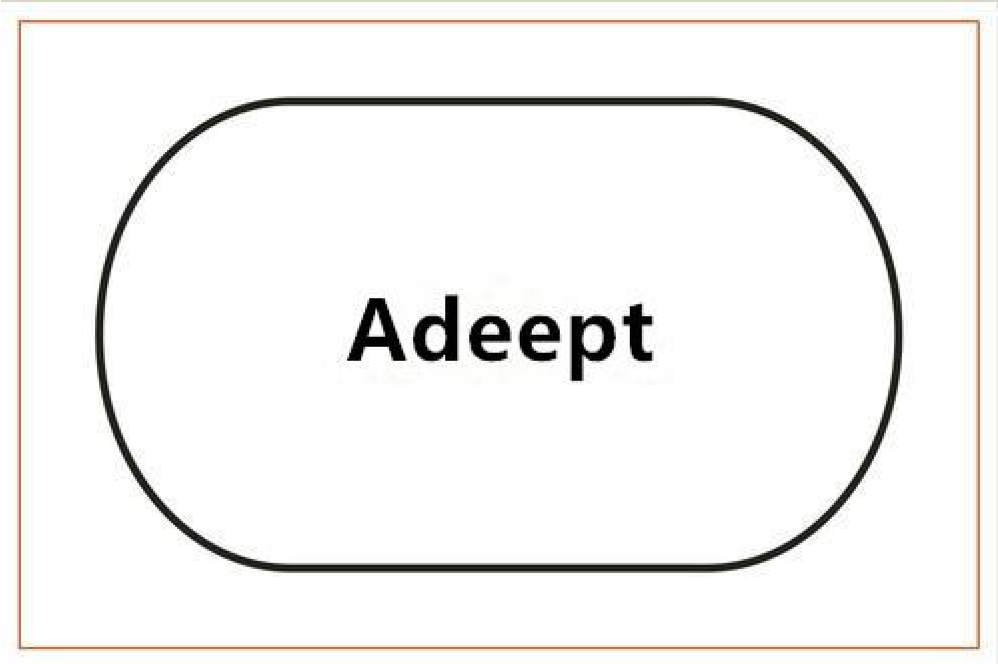


10.2 Preparation

1. The assembled PiCarPro. The tracking module is fixed under the center hole of the front wheel of the PiCar Pro chassis by a 15mm nylon column. Use M3*4 screws. Note: When not using the tracking function or driving on the ground with

obstacles, please disassemble the tracking module to avoid damage to the tracking module.

2. Making a patrol track. Due to the limitation of the steering range of the front wheel servo, the curve radius of the track cannot be too small.

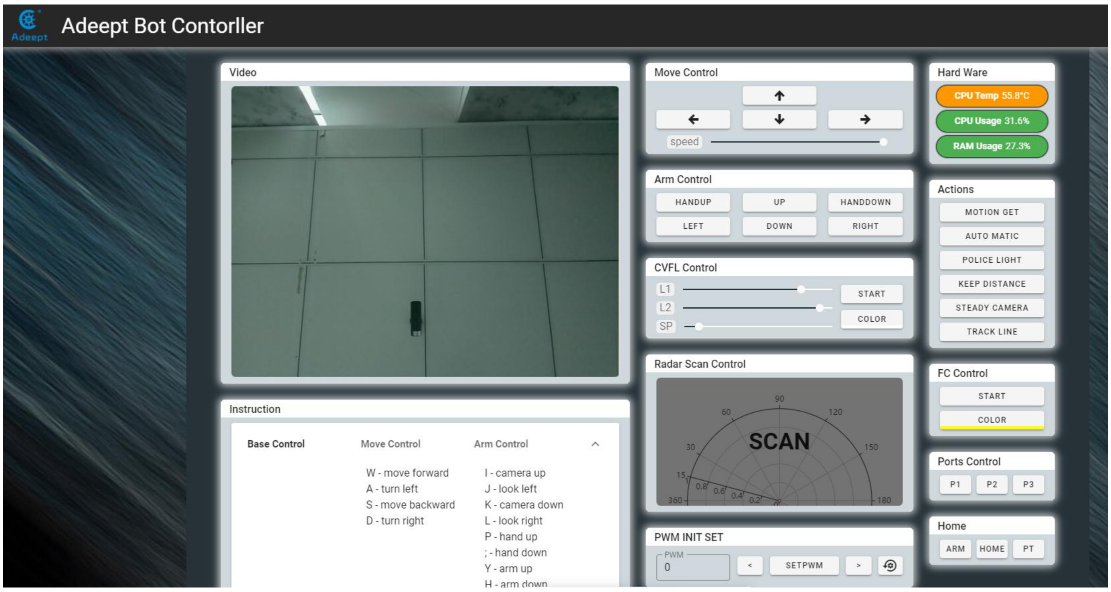


10.3 Turn on the tracking function

10.3.1 Running tracking program

1. Turn on PiCarPro, the boot time is about 1 minute.

2. After PiCarPro is turned on, enter the IP address of your Raspberry Pi with the Google browser of your mobile phone or computer, and access port 5000, for example: 192.168.3.44:5000 (see lesson 2 for detailed steps). The web controller will then be displayed on the browser.



3. Put the car on the ready-made patrol track.

4. Click "TRACK LINE", PiCarPro starts to drive along the black line.

5.When you want to terminate the tracking function, you can click "TRACK LINE" again.

6. The height of PiCar Pro's tracking module from the ground is about 7mm. When the Tracking module cannot be used normally, please refer to: Tracking module adjustment tutorial. Tutorial link:

https://www.adeept.com/learn/detail-50.html

10.3.2 Main code program

For the complete code, please refer to the file findline.py.

Import dependencies and initialize.

```
import RPi.GPIO as GPIO
import time

# The output pins of the line hunting module.
line_pin_right = 19
line_pin_middle = 16
line_pin_left = 20
```

```python

def setup():
    GPIO.setwarnings(False)
    GPIO.setmode(GPIO.BCM)
    GPIO.setup(line_pin_right,GPIO.IN)
    GPIO.setup(line_pin_middle,GPIO.IN)
    GPIO.setup(line_pin_left,GPIO.IN)
```

Define the main function of the hunt module.

```python
def run():
    status_right = GPIO.input(line_pin_right)
    status_middle = GPIO.input(line_pin_middle)
    status_left = GPIO.input(line_pin_left)

    # Detect whether the line hunting module senses lines.
    if status_middle == 1:
        print('forward')
    elif status_left == 1:
        print('left')
    elif status_right == 1:
        print('right')
    else:
            print('stop')
```

Execute function.

```python
if __name__ == '__main__':
    setup()
    while 1:
        run()
```

Lesson 11 Warning Light Function

In this lesson, we will learn how to use PiCarPro's warning light function.

11.1 Function introduction

This section of the course introduces the use of multi-threading to achieve some effects related to WS2812 LED lights. Multi-threading is a very common operation we use in robot projects, because robots have high requirements for real-time response. When performing a certain task, try not to block the main thread communication.

Multi-threading is similar to executing multiple different programs or tasks at the same time. Multi-threaded operation has the following advantages:

1. Using threads to put time-consuming tasks in the background for processing.

2. Improving the operating efficiency of the program. The subsequent real-time video and OpenCV processing video frames use multi-threading to greatly increase the frame rate.

3. The encapsulated multi-threaded task is more convenient to call, similar to the non-blocking control method, that is, the control method of the servo is encapsulated by multi-threading.

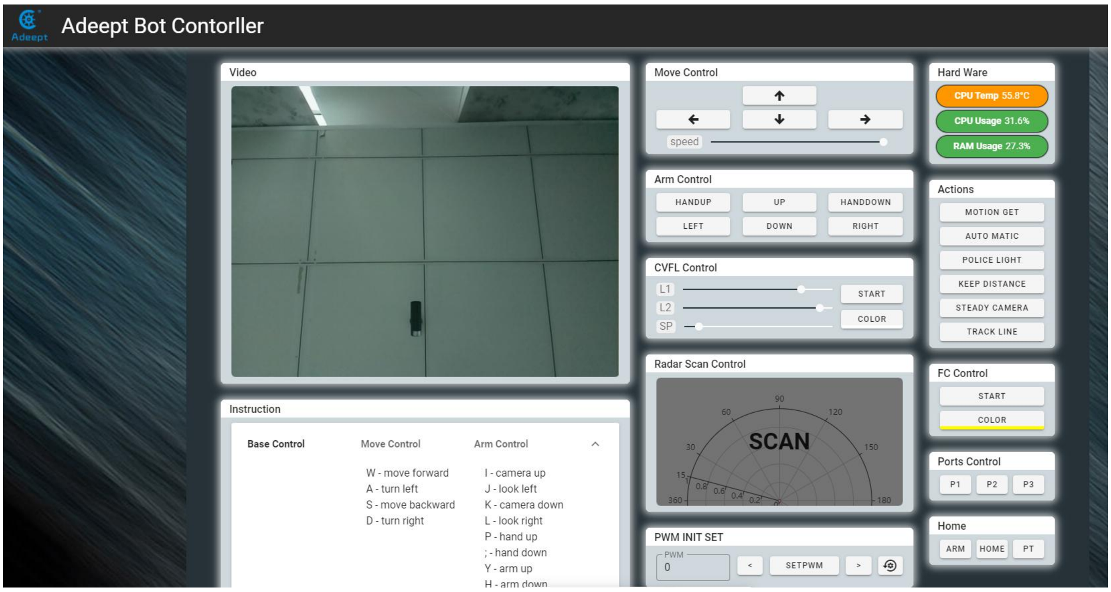
We use python's threading library to provide thread-related work, and thread is the smallest unit of work in an application. The current version of Python does not yet provide multi-threaded priority, thread groups, and threads cannot be stopped, suspended, resumed, or interrupted.

11.2 Realizing the WS2812 LED lighting effect with multithreading

11.2.1 Running the warning light program

1. Turn on PiCarPro, the boot time is about 1 minute.

2. After PiCarPro is turned on, enter the IP address of your Raspberry Pi with the Google browser of your mobile phone or computer, and access port 5000, for example: 192.168.3.44:5000 (see lesson 2 for detailed steps). The web controller will then be displayed on the browser.



3. Click "POLICE LIGHT", PiCarPro starts flashing lights of different colors.

4. When you want to stop the warning light function, click "POLICE LIGHT" again.

11.2.2 Main code program

For the complete code, please refer to the file robotLight.py.

Warning light mode method function

```
def policeProcessing(self):
        for i in range(0,3):
            self.colorWipe(Color(0,0,255))
            time.sleep(0.05)
            self.colorWipe(Color(0,0,0))
            time.sleep(0.05)
```

```
        time.sleep(0.1)
        for i in range(0,3):
            self.colorWipe(Color(255,0,0))
            time.sleep(0.05)
            self.colorWipe(Color(0,0,0))
            time.sleep(0.05)
        time.sleep(0.1)
```

Lesson 12 Ultrasonic Scanning Function

In this lesson, we will learn how to use PiCarPro's ultrasound scanning function.

12.1 Function introduction

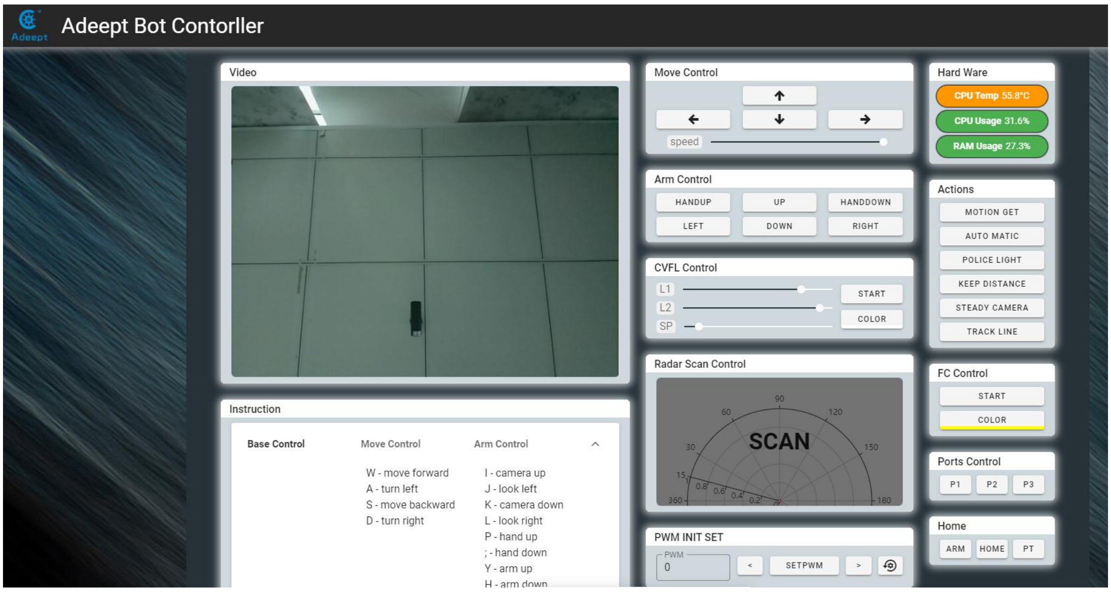
Ultrasonic scanning function means that after running the program, PiCarPro will rotate to detect the surrounding environment with the ultrasonic module. When there are obstacles around, it will display a red dot on the screen.

12.2 Turning on the ultrasonic scanning function

12.4.1 Running the ultrasonic scanning program

1. Turn on PiCarPro, the boot time is about 1 minute.

2. After PiCarPro is turned on, enter the IP address of your Raspberry Pi with the Google browser of your mobile phone or computer, and access port 5000, for example: 192.168.3.44:5000 (see lesson 2 for detailed steps). The web controller will then be displayed on the browser.



3. Place the car at the location to be detected.

4. Click the "SCAN" pattern, PiCarPro will start to rotate and detect the surrounding environment. After the detection is completed, there will be red dots in the pattern where there are obstacles.

12.4.2 Main code program

For the complete code, please refer to the file functions.py.

Import dependencies and initialize.

```python
import time
import RPi.GPIO as GPIO
import Adafruit_PCA9685

# Ultrasonic module pins
UPin1 = 11
Upin2 = 8

```

```
#Servo pin. Set as the pin of the servo on your product that can control the left and right rotation of the ultrasonic module.
servoPin = 1

GPIO.setmode(GPIO.BCM)
GPIO.setup(UPin1, GPIO.OUT,initial=GPIO.LOW)
GPIO.setup(Upin2, GPIO.IN)

pwm = Adafruit_PCA9685.PCA9685()
pwm.set_pwm_freq(50)

pwm0_init = 300
pwm0_max  = 450 # Set the maximum scanning angle
pwm0_min  = 150 # Set the minimum scanning angle
pwm0_pos  = pwm0_init
```

Define scan function.

```
def checkdist():
    ''' Refer to the basic module to control ultra.Py '''

def radarScan(): # Main function of ultrasonic scanning function
    global pwm0_pos
    scan_speed = 3 # Scanning speed

    # Move the servo to the maximum angle
    pwm0_pos = pwm0_max
    pwm.set_pwm(servoPin, 0, pwm0_pos)
    time.sleep(0.5)

    # Rotate the servo and continuously measure the distance during the period until it reaches the minimum angle.
    print(str(checkdist()), 'm')
    while pwm0_pos>pwm0_min:
        pwm0_pos-=scan_speed
        pwm.set_pwm(servoPin, 0, pwm0_pos)
        print(str(checkdist()), 'm')

    # Reply
    pwm.set_pwm(servoPin, 0, pwm0_init)
```

Execution function

```
if __name__ == '__main__':
    radar_scan()
    GPIO.cleanup()
```

Lesson 13 Camera Stabilization Function

In this lesson, we will learn how to use PiCarPro's camera stabilization function and its application scenarios.

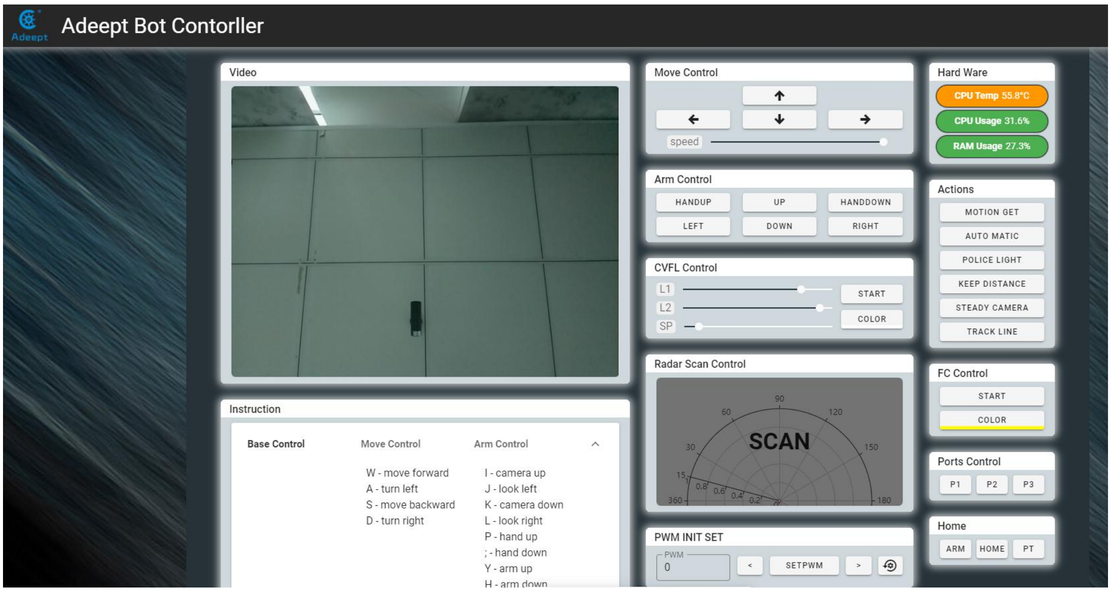
13.1 Camera application scenarios

When the car is driving on a non-level ground, in order for the camera to take a horizontal picture, the camera stabilization function is required.

13.2 Turning on camera stabilization

13.2.1 Running camera stabilization program

1. Turn on PiCarPro, the boot time is about 1 minute.
2. After PiCarPro is turned on, enter the IP address of your Raspberry Pi with the Google browser of your mobile phone or computer, and access port 5000, for example: 192.168.3.44:5000 (see lesson 2 for detailed steps). The web controller will then be displayed on the browser.



3. Place the car on the prepared patrol track.

4. Click "STEADY CAMERA", PiCarPro has the camera stabilization function.

5. When you want to terminate the camera stabilization function, you can click "STEADY CAMERA" again.

13.2.2 Main code program

For the complete code, please refer to the file steady.py.

Import dependencies and initialize.

```python
from mpu6050 import mpu6050
import Adafruit_PCA9685
import time

sensor = mpu6050(0x68)
pwm = Adafruit_PCA9685.PCA9685()

servoPort = 2       #The port number of the servo need to be controlled
servoPos  = 200     #Control the reference position of the servo
P = 1               #PID P value setting, proportional control parameter, the higher the faster
```

Main logic code

```python
def pwmGenOut(angleInput):
    """
    Convert the angle to PWM
    '''
    return int(round(23/9*angleInput))

while 1:
    print('steadyProcessing')
    xyzGet = sensor.get_accel_data()    #Get XYZ reading information from the sensor

    sGet = xyzGet['x']*10              #Get the X-axis angle information from the angle information
    #sGet = xyzGet['y']*10             #Get the Y-axis angle information from the angle information
    #sGet = xyzGet['z']*10             #Get the angle information of the Z-axis from the angle information
    pwm.set_pwm(2, 0, servoPos+pwmGenOut(sGet*P))   #Control the movement of the servo

    time.sleep(0.05)
```

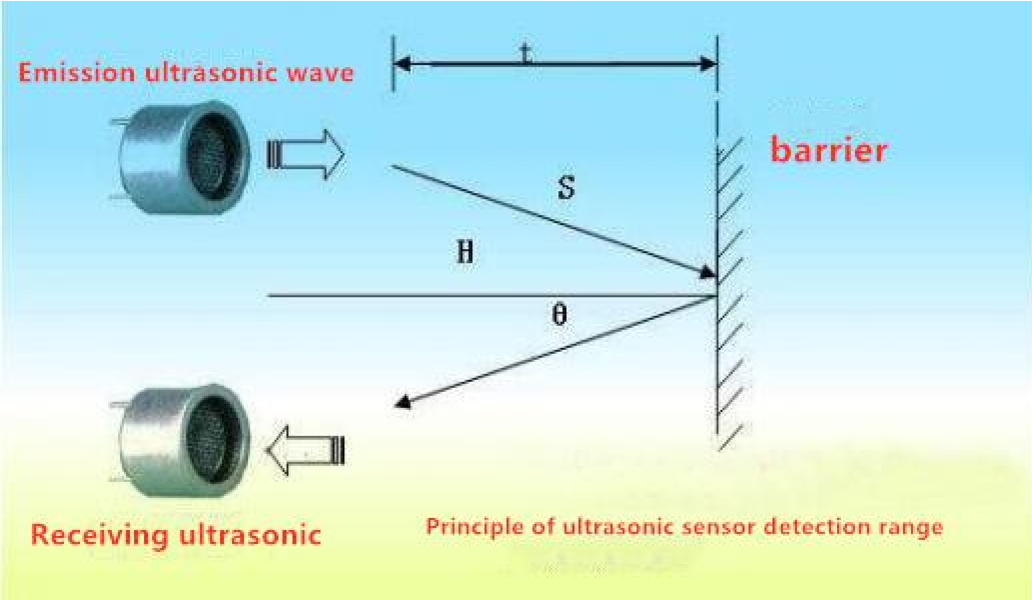
Lesson 14 Automatic Obstacle Avoidance Function

In this lesson, we will learn how to use PiCarPro to achieve automatic obstacle avoidance.

14.1 Introduction to Automatic Obstacle Avoidance

Since the camera used by our Raspberry Pi robot is a monocular camera and cannot collect depth information, many of our robot products use ultrasonic ranging modules to obtain depth information and detect whether there are obstacles in a certain direction to get the distance of the obstacle.

The principle of distance detection by ultrasonic ranging sensor: the method of detecting distance by ultrasonic is called echo detection method, that is, the ultrasonic transmitter emits ultrasonic waves in a certain direction, and the timer starts timing at the same time as the launch time. The ultrasonic waves propagate in the air and encounter obstacles on the way. When the object surface (object) is blocked, it will be reflected back immediately, and the ultrasonic receiver will immediately stop timing when the reflected ultrasonic wave is received. The propagation speed of ultrasonic waves in the air is 340m/s. According to the time t recorded by the timer, the distance s from the launch point to the obstacle surface can be calculated, namely: s=340t/2. Using this principle of ultrasound, the ultrasonic ranging module is widely used in practical applications, such as car reversing radar, unmanned aerial vehicle, and smart car.

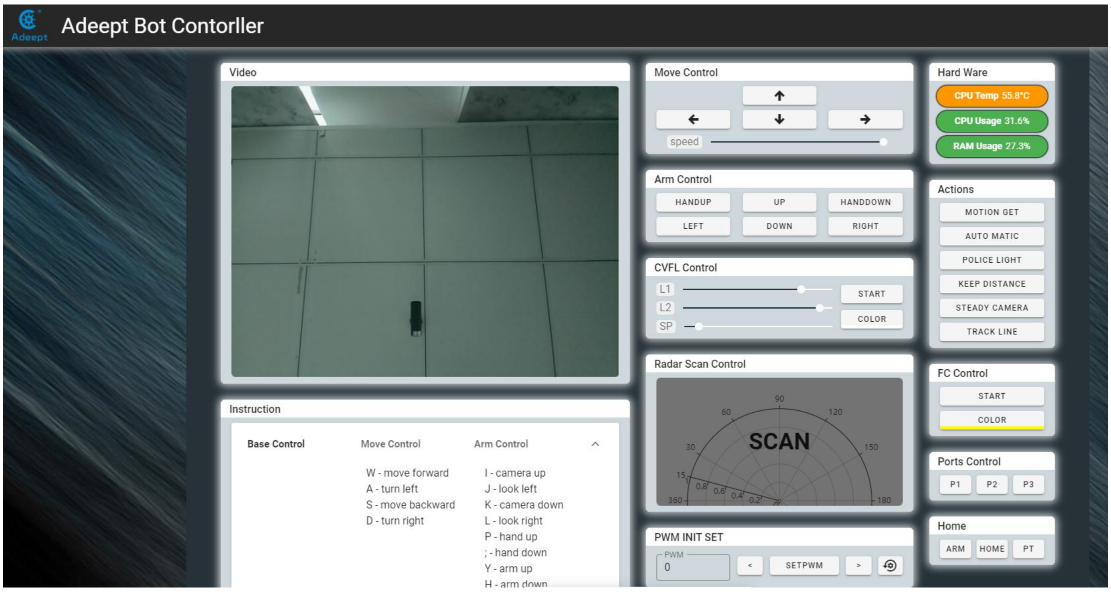


14.2 Turning on Automatic Obstacle Avoidance function

14.2.1 Running the Automatic Obstacle Avoidance program

1. Turn on PiCarPro, the boot time is about 1 minute.

2. After PiCarPro is turned on, enter the IP address of your Raspberry Pi with the Google browser of your mobile phone or computer, and access port 5000, for example: 192.168.3.44:5000 (see lesson 2 for detailed steps). The web controller will then be displayed on the browser.



3. Place the car on the prepared patrol track.

4. After clicking "AUTO MATIC", the robot will automatically avoid obstacles when encountering obstacles. .

5. When you want to terminate the Automatic Obstacle Avoidance function, you can click "AUTO MATIC" again.

14.2.2 The main code program of this lesson

For the complete code, please refer to the file functions.py.

Import dependencies and initialize.

```
import time
import RPi.GPIO as GPIO
import json
import Adafruit_PCA9685

pwm = Adafruit_PCA9685.PCA9685()
pwm.set_pwm_freq(50)

Tr = 11        #Pin No. of Ultrasonic Module Input
Ec = 8         # Pin number of the output end of the ultrasonic module
servoPort = 1      #The number of the servo that controls the horizontal rotation of the ultrasonic module
servoMiddle = 330   #The middle position of the servo
```

```
servoLeft = 180     #Left position of the servo
servoRight = 480    #The right position of the servo
rangeKeep = 0.3     #Avoidance distance

scanDir = 1     #Scan direction, 1 is from left to right, -1 is from right to left
scanPos = 1     #Store the current scan position (1 is the left, 2 is the middle, and 3 is the right)
scanNum = 3     #The number of scan positions (left, middle, and right, these are three positions)

scanList = [0,0,0]  #Save scan results

GPIO.setmode(GPIO.BCM)
GPIO.setup(Tr, GPIO.OUT,initial=GPIO.LOW)
GPIO.setup(Ec, GPIO.IN)
```

Main logic code

```
def checkdist():
    ''' Refer to the realization of basic functions-ultrasonic module '''

while 1:
    print('Automatic obstacle avoidance mode')
    if scanPos == 1:
        pwm.set_pwm(servoPort, 0, servoLeft)
        time.sleep(0.3)
        scanList[0] = checkdist()
    elif scanPos == 2:
        pwm.set_pwm(servoPort, 0, servoMiddle)
        time.sleep(0.3)
        scanList[1] = checkdist()
    elif scanPos == 3:
        pwm.set_pwm(servoPort, 0, servoRight)
        time.sleep(0.3)
        scanList[2] = checkdist()

    scanPos = scanPos + scanDir

    if scanPos > scanNum or scanPos < 1:
        if scanDir == 1:scanDir = -1
        elif scanDir == -1:scanDir = 1
        scanPos = scanPos + scanDir*2
    print(scanList)

```

```
    if min(scanList) < rangeKeep:
        if scanList.index(min(scanList)) == 0:  #The shortest distance detected on the left
            '''
            Turn right
            '''
            print('Turn right')
        elif scanList.index(min(scanList)) == 1:    #The shortest distance detected in the middle
            if scanList[0] < scanList[2]:
                '''
                If the detected distance on the left is shorter than the right, turn to the right
                '''
                print('Turn right')
            else:
                """
                Otherwise, turn left
                '''
                print('Turn left')
        elif scanList.index(min(scanList)) == 2:    #The shortest distance detected on the right
            '''
            Turn Left
            '''
            print('Turn Left')
        if max(scanList) < rangeKeep:
            '''
            If the distances in the left, center, and right directions are all closer than rangeKeep, reverse
            '''
            print('reverse')
    else:
        '''
        All three directions are farther than rangeKeep
        '''
        print('Go forward')
```

Lesson 15 Creating a WiFi Hotspot on Raspberry Pi

The method of turning on the WIFI hotspot in our robot product uses a project from GitHub create_ap. Our installation script will automatically install this program and related dependent libraries. If you have not run our installation script, you can use the following command to install create_ap:

sudo git clone https://github.com/oblique/create_ap.git

cd create_ap

sudo make install

1.Install related dependent libraries:

sudo apt-get install -y util-linux procps hostapd iproute2 iw haveged dnsmasq

2.Before turning on the hotspot, your Raspberry Pi cannot be connected to WIFI, and the WIFI module cannot be turned off, so when you test the hotspot function, you need to connect the necessary peripherals for the Raspberry Pi.

3.Under normal circumstances, if the robot program is not connected to the WIFI when it is turned on, it will automatically turn on the hotspot. You can use your phone or computer to search for the WIF named Adeept. The default password is 12345678. Once the connection is successful, you can log in to 192.168 .12.1: 5000 using a browser to open the WEB application to control the robot.

4.If your Raspberry Pi is connected to peripherals, and you want to test the Raspberry Pi ’s ability to turn on hotspots, you can click the WIFI icon in the upper right corner of the Raspberry Pi ’s desktop, click the name of the connected WIFI, click forget, and never turn Off WIFI, if it is already in the off state, you need to turn it on.

5.When the WIFI module of the Raspberry Pi is turned on and is not connected to any known network, you can enter the following command on the console to turn on the WIFI:

sudo create_ap wlan0 eth0 Adeept 12345678

Adeept is the name of the WIFI hotspot, 12345678 is the password of the WIFI hotspot.

Adeept

STEM Education Products and Service Provider

Shenzhen Adeept Technology Co.,Ltd.

: www.adeept.com
: support@adeept.com
: Rm.1315, Shihong Building, No.2095 Bixin Rd., Longgang Dist., Shenzhen CHINA