

* Resources Link

RobotName: Adeept\_Rasptank

RobotURL: https://github.com/adeept/adeept\_rasptank RobotGit: https://github.com/adeept/adeept\_rasptank.git

[Official Raspberry Pi website] https://[www.raspberrypi.org/downloads/](http://www.raspberrypi.org/downloads/) [Official website] https://[www.adeept.com/](http://www.adeept.com/)

[GitHub] https://github.com/adeept/adeept\_rasptank/

[Image file and Documentation for structure assembly] https://[www.adeept.com/learn/detail-34.html](http://www.adeept.com/learn/detail-34.html)

**Content**

* 1. Premise 1
  1. [STEAM and Raspberry Pi 1](#_TOC_250024)
  2. [About the Documentation 1](#_TOC_250023)
* 2. Raspberry Pi System Installation and Development Environment Establishment 3
  1. [Install an operating system for the Raspberry Pi 3](#_TOC_250022)
     1. [Method A: Write 'Raspbian' to the SD card by Raspberry Pi Imager 3](#_TOC_250021)
     2. [Method B: Download the image file Raspbian and write it to the SD card manually 5](#_TOC_250020)
     3. [Method C: Manually download the image file provided by us and write it to the SD card (not recommended)](#_TOC_250019)

.......................................................................................................................................................................................8 2.2 Enable SSH Server of Raspberry Pi............................................................................................................................... 10

* + 1. [Method A: Enable SSH with peripherals 11](#_TOC_250018)
    2. [Method A: Enable SSH without peripherals 11](#_TOC_250017)
  1. [Configure WiFi on Raspberry Pi 12](#_TOC_250016)
     1. [Method A: WiFi connection with peripherals 12](#_TOC_250015)
     2. [Method A: WiFi connection without peripherals 12](#_TOC_250014)
* 3 Log In to The Raspberry Pi and Install The App 14
  1. [Log into Raspberry Pi (Windows 10) 14](#_TOC_250013)
  2. [Log into Raspberry Pi (Linux or Mac OS) 15](#_TOC_250012)
  3. [Log into Raspberry Pi (Windows 8 or previous version) 15](#_TOC_250011)
  4. [Download Program of the Raspberry Pi Robot 16](#_TOC_250010)
  5. [Install Corresponding Dependent Libraries 17](#_TOC_250009)
  6. [Run the Raspberry Pi Robot's Program 17](#_TOC_250008)
* 4 Raspberry Pi Structure Assembly and Precautions 18
  1. [Documentation for structure assembly 18](#_TOC_250007)
  2. [Tips for structural assemblage 81](#_TOC_250006)
  3. [Tips for power provision 82](#_TOC_250005)
* 5 Controlling Robot via WEB App 83
* 6 Common Problems and Solutions(Q&A) 85
* 7 Set The Program to Start Automatically 89
  1. [Set the specified program to run automatically at boot 89](#_TOC_250004)
  2. [Change the program that starts automatically 89](#_TOC_250003)
* 8 Remote Operation of Raspberry Pi Via MobaXterm 90
* 9 How to Control WS2812 RGB LED 92
* 10 How to Control The Servo 94
  1. [Control the steering gear to rotate to a certain angle 94](#_TOC_250002)
  2. [Control the slow motion of the steering gear 96](#_TOC_250001)
  3. [Non-blocking control 96](#_TOC_250000)
* 11 How to Control DC Motor 97
* 12 Ultrasonic Module 100
* 13 Line Tracking 102
* 14 Make A Police Light or Breathing Light 105
  1. Multi-threading introduction 105
  2. Realization of police lights / breathing lights 105
  3. Warning lights or breathing lights in other projects 110
* 15 Real-Time Video Transmission 111
* 16 Automatic Obstacle Avoidance 115
* 17 Why OpenCV Uses Multi-threading to Process Video Frames 117
  1. Single thread processing of video frames 117
  2. Multi-thread processing of video frames 118
* 18 OpenCV Learn to Use OpenCV 122
* 19 Using OpenCV to realize color recognition and tracking 126
  1. Color recognition and color space 126
  2. Process of color identifying and tracking 126
  3. Specific code 127
  4. HSV color component range in OpenCV 130
* 20 Machine Line Tracking Based on OpenCV 131
  1. Visual line inspection process 131
  2. Specific code 132
* 21 Create A WiFi Hotspot on The Raspberry Pi 134
* 22 Install GUI Dependent Item under Window 136

22.1 Install GUI dependencies 136

* 23 How to Use GUI 137
* 24 Control The WS2812 LED via GUI 137
* 25 Real-time video transmission based on OpenCV 142
* 26 Use OpenCV to Process Video Frames on The PC 145
* 27 Enable UART 149
  1. Mini UART and CPU core frequency 149
  2. Disabling Linux's use of console UART 150
  3. UART output on GPIO pins 150
  4. UARTs and Device Tree 150
  5. Relevant differences between PL011 and mini UART 151
* 28 Control Your RaspTank with An Android Device 151

Conclusion 153

* 1. Premise

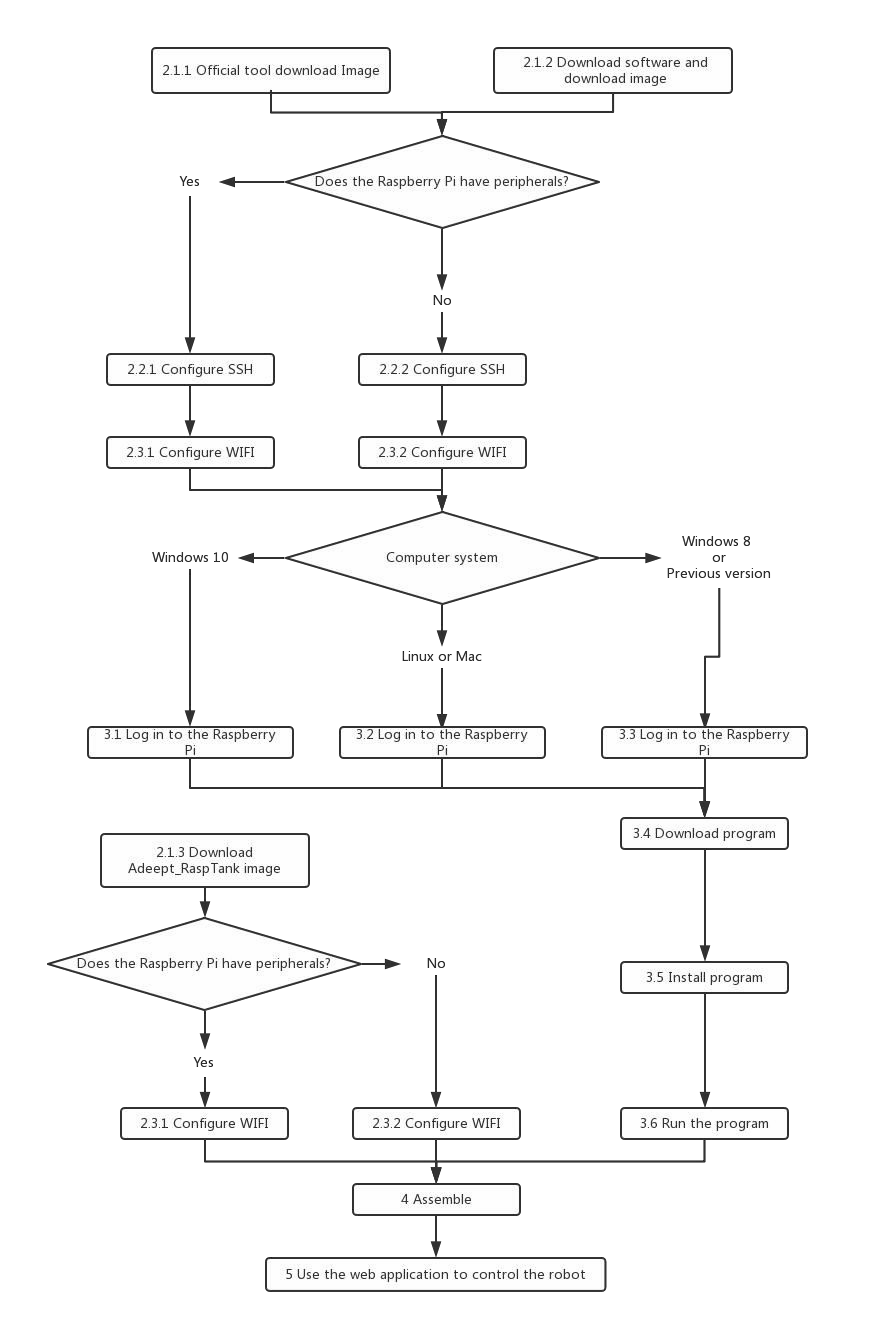
# STEAM and Raspberry Pi

STEAM stands for Science, Technology, Engineering, Arts and Mathematics. It's a type of trans disciplinary education idea focused on practice. As a board designed for computer programming education, Raspberry Pi has lots of advantages over other robot development boards. Therefore, Raspberry Pi is used for function control of the robot.

# About the Documentation

This documentation is for software installation and operation guide for the Python robot product. It describes every detail of the whole process of fulfilling the robot project by Python and Raspberry Pi from scratch as well as some precautions. Hope you can get started with the Raspberry Pi robot on Python and make more creations with this documentation.

According to the different situations of different users, there will be some changes in the process of this document, you can refer to the following process:



* 2. Raspberry Pi System Installation and Development Environment Establishment

# 2.1 Install an operating system for the Raspberry Pi

### Method A: Write 'Raspbian' to the SD card by Raspberry Pi Imager

is an image writing tool to SD card developed by the Raspberry Pi Organization. It

Raspberry Pi Imager

comes with many versions working on different systems and it's quite easy to use; all you need is choose the operating system and SD card, Raspberry Pi Imager will download the corresponding image file for the system and install it to the SD card.

**Step-by-Step Overview**

* + - 1. Prepare an SD card (16G or larger) and an SD card reader
      2. Download the `Raspberry Pi Imager` on the official website
* [Raspberry Pi Imager for Windows]
* [Raspberry Pi Imager for macOS]
* [Raspberry Pi Imager for Ubuntu]
  + - 1. Install the `Raspberry Pi Imager`
      2. Write the operating system for Raspberry Pi to the SD card with `Raspberry Pi Imager` `Raspbian Full - A port of Debian with desktop and recommended application`
      3. Leave the SD card connected after writing is completed, we'll use for configuring SSH and WiFi connection later.

**Detailed Steps:**

⃗Open a web browser on your computer, go to the Raspberry Pi website [Official Raspberry Pi website], find and

download the

Raspberry Pi Imager

for your computer OS, or click on the links above for the corresponding

system to directly download and install.



⃗Insert the SD card into the card reader, connect the card reader with your computer.

⃗Run the Raspberry Pi Imager, select -> ->

Raspbian Full - A port of Debia

n

CHOOSE OS

Raspbian(other)

with desktop and recommended applications.

⃗Click on

CHOOSE SD CARD

for the SD card to write the Raspbian Full, please be noted that the image

writing will automatically delete all files on the SD card if any.

Raspbian

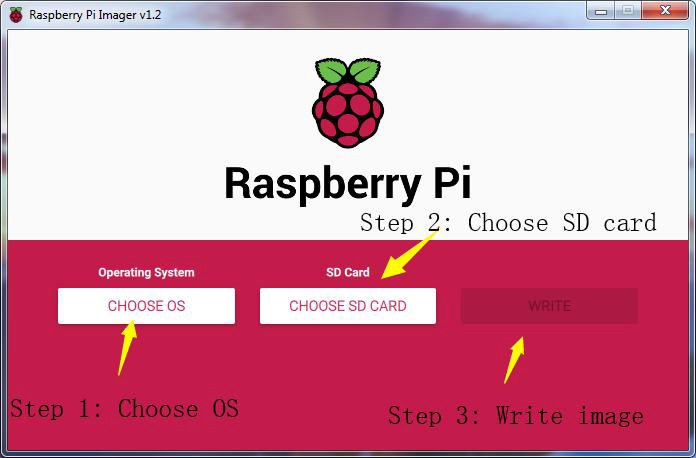
⃗Click on WRITE, wait for the writing. The

Raspberry Pi Imager

needs to download the

image file

during the process. You can download the file following the step in 2.1.2.



⃗Do not remove the SD card connected after writing is completed, we'll use for configuring SSH and WiFi

connection later. Otherwise, if you remove the card, insert it into the Raspberry Pi and boot, WiFi configuration without any peripherals may fail in the following process.

### Method B: Download the image file Raspbian and write it to the SD card manually

⃗Since the image file is downloaded with Raspberry Pi Imager in 2.1.1, it can take a long time due to a slow network in some places. You may then manually download the image file Raspbian and write it to the SD card with th Raspberry Pi Imager.

**Step-by-Step Overview**

* + - 1. Prepare an SD card (16G or larger) and an SD card reader
      2. Download the `Raspberry Pi Imager` on the official website [Official Raspberry Pi website]
* [Raspberry Pi Imager for Windows]
* [Raspberry Pi Imager for macOS]
* [Raspberry Pi Imager for Ubuntu]
  + - 1. Install the `Raspberry Pi Imager`
      2. Download the image file `Raspbian`
* Torrent file:

[Raspbian-Raspbian Buster with desktop and recommended software]

-Zip file: [Raspbian - Raspbian Buster with desktop and recommended software]

* + - 1. Unzip the file, be noted that the path should be in English for the `.img` file extracted, no special characters allowed.
      2. Write the image file `Raspbian` downloaded to the SD card with `Raspberry Pi Imager`
      3. Leave the SD card connected after writing is completed, we'll use for configuring SSH and WiFi connection later.

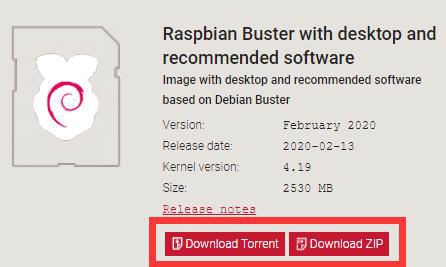
**Detailed Steps:**

⃗Open a web browser on your computer, go to the Raspberry Pi website[Official Raspberry Pi website], find and download the Raspberry Pi Imager for your computer OS, or click on the links above for the corresponding system to directly download and install.



⃗On the Raspberry Pi website [Official Raspberry Pi website], select through Downloads -> Raspbian ->

Raspbian Buster with desktop and recommended software, and click on the torrent or zip file to download. Unzip the file after download, be noted that the path should be in English for the .img file extracted, no special characters allowed; otherwise Raspberry Pi Imager may not open the .img file. It's recommended to save the .img file to the root directory of the C:\ or D:\ disk, but do not save .img on the SD card.



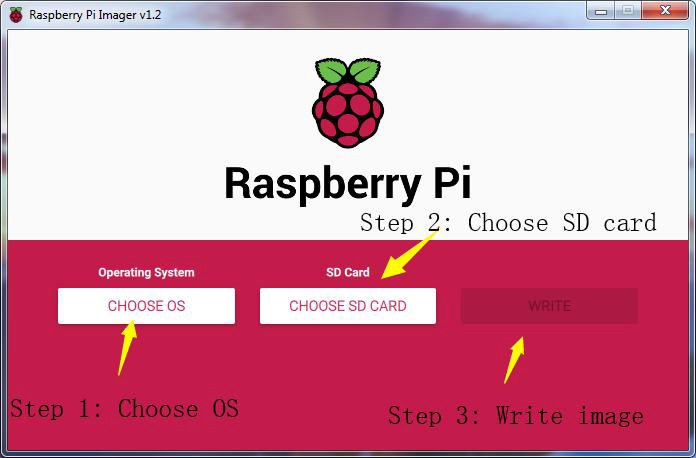
⃗Insert the SD card into the card reader, connect the card reader and your computer.

⃗Run the Raspberry Pi Imager, select CHOOSE OS, and then Use custom to find the .img extracted, click Open.

⃗Select CHOOSE SD CARD for the SD card to write the Raspbian, please be noted that the image writing

will automatically delete all files on the SD card if any.

⃗Click on WRITE, wait for the writing.



⃗Do not remove the SD card connected after writing is completed, we'll use for configuring SSH and WiFi

connection later. Otherwise, if you remove the card, insert it into the Raspberry Pi and boot it up, WiFi configuration without any peripherals may fail in the following process.

* + 1. **Method C: Manually download the image file provided by us and write it to the SD card (not recommended)**

⃗ The Raspbian image file downloaded in **2.1.1** and **2.1.2** is the official source with some preinstalled software. To operate the robot, you may need many dependent libraries. Though we provide the simple script to install them (see details later), failure can happen during installation if the library is not the latest version. Therefore, despite we provide the downloading of the Raspbian image file, it may happen that our image file and the dependent libraries are not most updated versions. Please only use when you encounter the most troublesome situation.

⃗Step-by-Step Overview

* + - 1. Prepare an SD card (16G or larger) and an SD card reader
      2. Download the `Raspberry Pi Imager` from the official website [Official Raspberry Pi website]
* [Raspberry Pi Imager for Windows]
* [Raspberry Pi Imager for macOS]
* [Raspberry Pi Imager for Ubuntu]
  + - 1. Install the `Raspberry Pi Imager`
      2. Download the image file `Adeept\_RaspTank`
* [Image file for the Adeept\_RaspTank Robot]
  + - 1. Unzip the file, be noted that the path should be in English for the `.img` file extracted, no special characters allowed.
      2. Write the image file `Raspbian` downloaded to the SD card with `Raspberry Pi Imager`
      3. Leave the SD card connected after writing is completed, we'll use for configuring SSH and WiFi connection later.

#### Detailed Steps:

⃗Open a web browser on your computer, go to the Raspberry Pi website [Official Raspberry Pi website], find and download the Raspberry Pi Imager for your computer OS, or click on the links above for the corresponding system to directly download and install.



⃗Go to our [official website], find and download the image file [Image file for the Adeept\_RaspTank Robot]. Unzip the file, be noted that the path should be in English for the .img file extracted, no special characters allowed. otherwise Raspberry Pi Imager may not open the .img file. It's recommended to save the .img file to the root directory of the C:\ or D:\ disk, but do not save .img on the SD card.

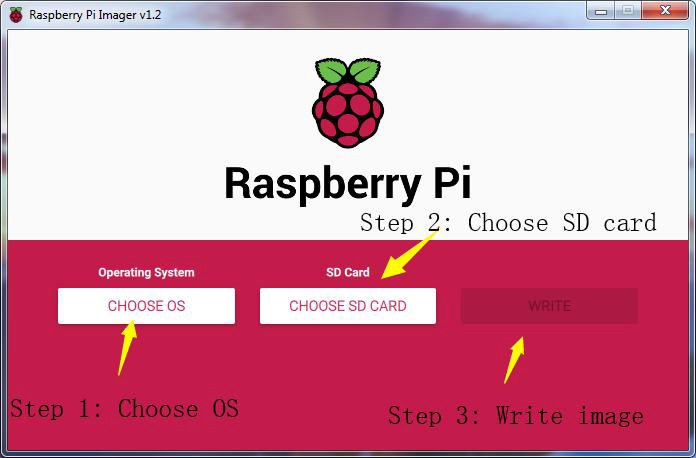
⃗Insert the SD card into the card reader, connect the card reader and your computer.

⃗Run the Raspberry Pi Imager, select CHOOSE OS, and then Use custom to find the .img extracted, click Open.

⃗Select CHOOSE SD CARD for the SD card to write the Raspbian, please be noted that the image writing

will automatically delete all files on the SD card if any.

⃗Click on WRITE, wait for the writing.



⃗Do not remove the SD card connected after writing is completed, we'll use for configuring WiFi connection

later. Otherwise, if you remove the card, insert it into the Raspberry Pi and boot it up, WiFi configuration without any peripherals may fail in the following process.

# 2.2 Enable SSH Server of Raspberry Pi

### ⃗By SSH (Secure Shell) server, you can use the command line of Raspberry Pi remotely on another device. In the subsequent operation and when using the Raspberry Pi, you don't have to connect a mouse, keyboard, or monitor to it, but simply control it on a computer in the same LAN.

⃗As of the November 2016 release, Raspbian has the SSH server disabled by default. You will have to enable it manually.

⃗The method to enable the SSH in this documentation can be referred to the Raspberry Pi

official website SSH(Secure Shell)

**2.2.1 Method A: Enable SSH with peripherals**

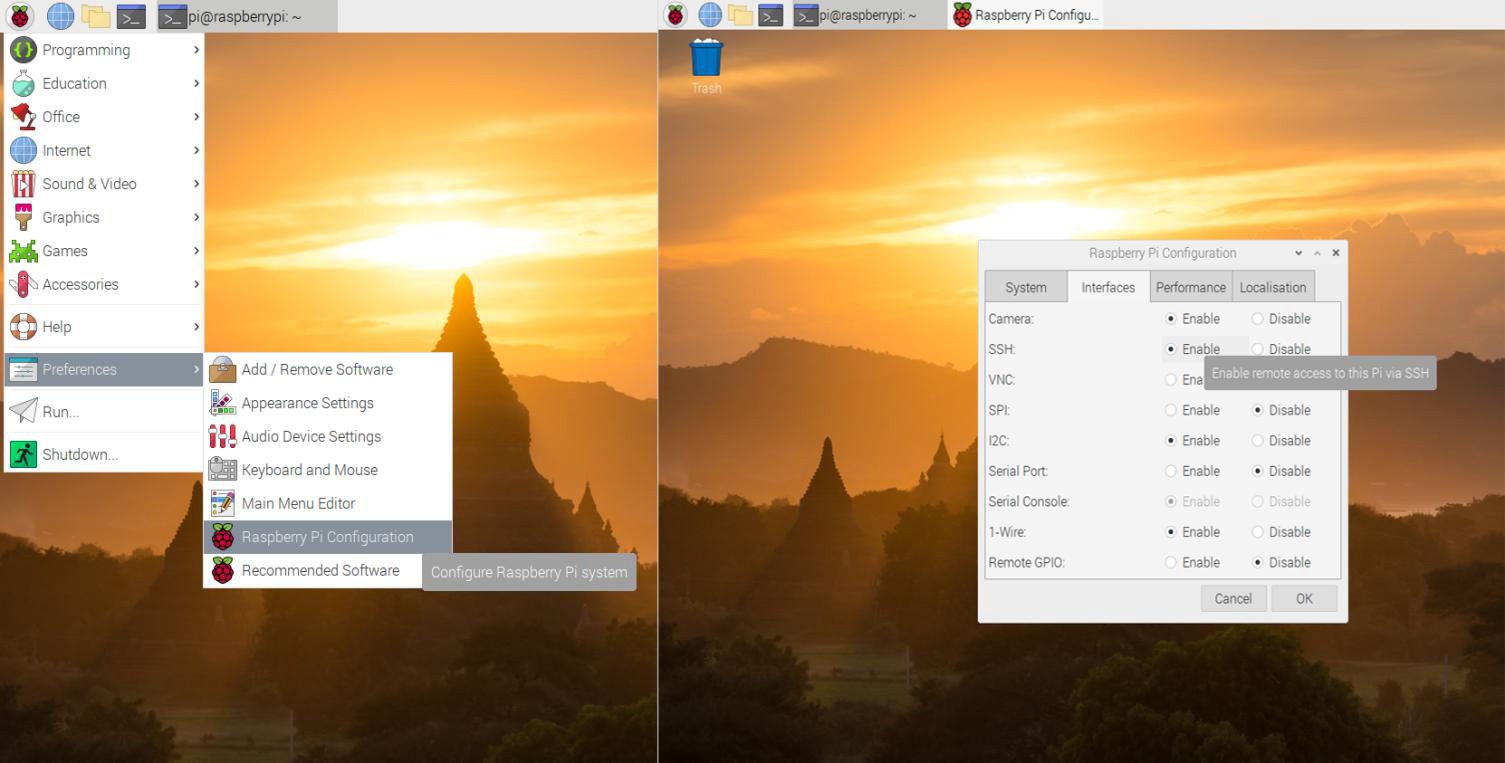
⃗If you use (**2.1.3 to manually download the image file we provide and write it to the SD card**) to write the operating system of the Raspberry Pi to the SD card, you do not need to refer to this section to open SSH, because The SSH service in the image is already enabled.

⃗If you've connected a mouse, keyboard, or monitor to the Raspberry Pi, follow these steps to enable SSH. 1.Remove the SD card from the computer, insert it to the Raspberry Pi, connect a mouse, keyboard, and

monitor to the Raspberry Pi, boot it up.

2.Go to Preferences menu, select Raspberry Pi Configuration. 3.Go to Interfaces option.

4.Select Enable next to SSH. 5.Click on OK.



### Method A: Enable SSH without peripherals

⃗If you use (**2.1.3 to manually download the image file we provide and write it to the SD card**) to write

the operating system of the Raspberry Pi to the SD card, you do not need to refer to this section to open SSH, because The SSH service in the image is already enabled.

⃗If you haven't connected any monitor to the Raspberry Pi, follow these steps to enable SSH.

* + - 1. Do not remove the SD card after `Raspberry Pi Imager` writes the image file.
      2. Create a file named `ssh` under any directory, without any extension name. You may create a `ssh.txt` and delete the `.txt` (make sure under Folder Options the box of Hide extensions for known file types is unchecked. Then you have an `ssh` file without extension name.
      3. Copy the `ssh` file and paste to the root directory of the SD card. The Raspberry Pi will auto search for the `ssh` file when booting, and enable SSH if the file is found. You only need to copy for one time because the Raspberry Pi then will automatically enable SSH at every boot.
      4. Do not remove the SD card if you need to configure WiFi.

# Configure WiFi on Raspberry Pi

⃗There are many ways to connect WiFi for Raspberry Pi. Two methods are provided in this documentation; you may visit the official Raspberry Pi website for more: [Wireless connectivity]

## Method A: WiFi connection with peripherals

WiFi.

⃗ If you've connected a mouse, keyboard, or monitor to the Raspberry Pi, follow these steps to configure

1. Remove the SD card from the computer, insert it to the Raspberry Pi, connect a mouse, keyboard, and

monitor to the Raspberry Pi, boot it up.

1. Select the WiFi icon at the top right corner on the monitor, find the WiFi to connect and select.
2. Type in the password for the WiFi, connect.
3. After it's connected successfully, the WiFi will be saved and the Raspberry Pi will auto connect for next boot, so you don't need to connect peripherals every time.

## Method A: WiFi connection without peripherals

⃗If you haven't connected any monitor to the Raspberry Pi, follow these steps to configure WiFi.

⃗This method is based on the [official documentation]

* + - 1. Do not remove the SD card after `Raspberry Pi Imager` has written the image file. (This method works for the situation that the Raspbian image file has just been written to the SD card; if you've already plugged the SD card into the Raspberry Pi and got it rebooted after the image file being written, the configuration may fail.)
      2. Create a file named `wpa\_supplicant.conf` anywhere in your computer.
      3. Open the file `wpa\_supplicant.conf` created with Textbook, enter the following code: ctrl\_interface=DIR=/var/run/wpa\_supplicant GROUP=netdev update\_config=1

country=Insert country code here network={

ssid="Name of your WiFi" psk="Password for your WiFi"

}

* + - 1. Type in your own information for `Insert country code here`, `Name of your WiFi`, and `Password for

your WiFi`. Pay attention to the capitalization. Refer to the example below: ctrl\_interface=DIR=/var/run/wpa\_supplicant GROUP=netdev update\_config=1

country=US network={ ssid="MyName" psk="12345678"

}

* + - 1. Save and exit. Copy the `wpa\_supplicant.conf` to the root directory of the SD card.
      2. If you've alreay copied the file `ssh` to the SD card as instructed in \*\*2.2\*\*, then both the WiFi and SSH

settings without peripherals are done. You may remove the SD card, insert it into the Raspberry Pi, and boot it up.

* + - 1. For more about the file `wpa\_supplicant.conf`, refer to the official documentation [WIRELESS-CLI]
* 3 Log In to The Raspberry Pi and Install The App

⃗ If you followed the steps in **2.2.1** and **2.3.1** for SSH and WiFi configuration, you may remove the peripherals now and use SSH to remotely control the Raspberry Pi later on.

⃗If you followed the steps in **2.2.2** and **2.3.2**, you may now insert the SD card into the Raspberry Pi and boot it up. The Raspberry Pi will auto boot and connect WiFi when powered on, with no need of peripherals.

⃗If you use the operation steps of **2.1.3** to write to the SD card, you only need to refer to **2.3.1** or **2.3.2** to configure the WIFI, you can install the SD card into the Raspberry Pi, and the robot product program will Automatic operation, you can skip some content, refer to **5 Use WEB application to control the robot** after the structure is assembled.

⃗Some steps mentioned below are based on the official Raspberry Pi documentation SSH.

⃗For power supply of the Raspberry Pi, refer to the official documentation Power supply.

⃗The Robot HAT board of the Adeept Raspberry Pi Robot can supply power for the Raspberry Pi via GPIO port. However, since it may take a long time to install software on the Raspberry Pi, it's not recommended to supply with the batteries during this process. You may skip the installation of the Robot HAT board or camera during software installation; though you need to make sure the driver board and camera for the Raspberry Pi when it's ready to run the software installed, or a program error will occur.

# Log into Raspberry Pi (Windows 10)

⃗For Windows 10, SSH is built in the versions after October 2018, so you don't need any third-party software.

⃗For lower versions of Windows OS, SSH is not built in, and you may log into the Raspberry Pi by

referring to the official documentation [SSH using Windows].

⃗Before connecting the Raspberry Pi via SSH, you need to know the IP address of the Raspberry Pi.

Check the Management interface for your router, or download the app `Network Scanner` -> search for a device named `RASPBERRY` or `Raspberry Pi Foundation` to get the IP address.

⃗For other methods of obtaining the IP address of Raspberry Pi, refer to the official documentation [IP

Address]

⃗Press the keys ‘win’+’R’, type in `cmd`, and press ‘enter’.

⃗The default user is `pi`, and the password is `raspberry`.

⃗ Type in `ssh pi@<IP>` in the command line, replace the `<IP>` with the IP address of your Raspberry Pi, as shown below:

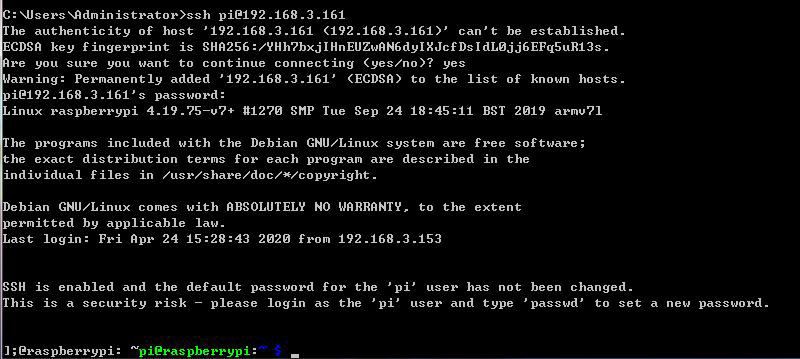
ssh [pi@192.168.3.161](mailto:pi@192.168.3.161)

⃗Press Enter key and a prompt will appear: `Are you sure you want to continue connecting (yes/no)?`

⃗Type in `yes`, press Enter and it'll show `pi@192.168.3.161's password:`, type in the initial password

of the Raspberry Pi, `raspberry` (pay attention to capitalization). There's no change on the screen when you're typing in, but it doesn't mean you're not entering the information. Press ‘enter’ after you finish typing in.

⃗So now you've logged into the Raspberry Pi.



# Log into Raspberry Pi (Linux or Mac OS)

⃗Before connecting the Raspberry Pi via SSH, you need to know the IP address of the Raspberry Pi. Check the Management interface for your router, or download the app `Network Scanner` -> search for a device named `RASPBERRY` or `Raspberry Pi Foundation` to get the IP address.

⃗For other methods of obtaining the IP address of Raspberry Pi, refer to the official documentation [IP

Address]

⃗Open the terminal window (or command line)

⃗The default user is `pi`, and the password is `raspberry`.

⃗Type in `ssh pi@<IP>` in the command line, replace `<IP>` with the IP address of your Raspberry Pi as shown below:

ssh [pi@192.168.3.161](mailto:pi@192.168.3.161)

⃗Press Enter key and a prompt will appear: `Are you sure you want to continue connecting (yes/no)?`

⃗Type in `yes`, press Enter and it'll show `pi@192.168.3.161's password:`, type in the initial password of the Raspberry Pi, `raspberry` (pay attention to capitalization). There's no change on the screen when you're typing in, but it doesn't mean you're not entering the information. Press ‘enter’ after you finish typing in.

⃗So now you've logged into the Raspberry Pi.

# Log into Raspberry Pi (Windows 8 or previous version)

⃗For lower versions of Windows OS, SSH is not built in, and you may log into the Raspberry Pi by

referring to the official documentation Raspberry Pi[SSH using Windows].

⃗Before connecting the Raspberry Pi via SSH, you need to know the IP address of the Raspberry Pi. Check the Management interface for your router, or download the app `Network Scanner` -> search for a device named `RASPBERRY` or `Raspberry Pi Foundation` to get the IP address.

⃗For other methods of obtaining the IP address of Raspberry Pi, refer to the official documentation [IP Address]

⃗You may need to download the `PuTTY` version for your OS and log into Raspberry Pi with the tool. [Click here to download PuTTY]

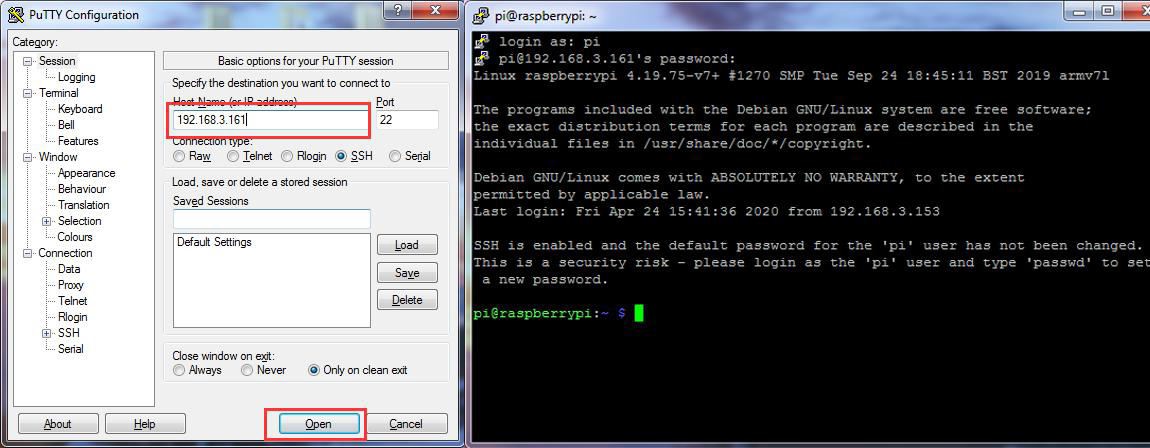
⃗Run `PuTTY`, type in the IP address of Raspberry Pi for `Host Name`, and click ‘Open’.

⃗If a prompt of `Network error: Connection timed out` appears, possibly you've entered an incorrect IP address.

⃗When the connection works you will see the security warning shown below. You can safely ignore it, and click the 'Yes' button. You will only see this warning the first time PuTTY connects to a Raspberry Pi that it has not seen before.

⃗You will now see the usual login prompt. Log in with the same username and password you would use on the Pi itself. The default login for Raspbian is `pi` with the password `raspberry`.

⃗ You should now have the Raspberry Pi prompt which will be identical to the one found on the Raspberry Pi itself.



# Download Program of the Raspberry Pi Robot

⃗ The code for the robot product has been uploaded to [GitHub], you may need to download to your Raspberry Pi and install the corresponding dependent libraries to run the program.

⃗In the previous section you've logged into the Raspberry Pi, and here type in the follow command in the

terminal window:

sudo git clone https://github.com/adeept/adeept\_rasptank.git

⃗Press ‘enter’ to start downloading the program of the robot from GitHub. It may take some time, please

wait until it's done.

# Install Corresponding Dependent Libraries

⃗Follow the steps below to install the libraries if you wrote the image file to the SD card based on 2.1.1 Write 'Raspbian' to the SD card by `Raspberry Pi Imager` and 2.1.2 Download the image file `Raspbian` and write it to the SD card manually.

⃗ Here a script is provided for installing all dependent libraries needed and configuration of starting the

camera and other auto start programs.

⃗Type in the code below in the terminal window to run the dependent libraries for the script `setup.py`: sudo python3 adeept\_rasptank/setup.py

⃗ Press ‘enter’ and the script will auto run. This may take minutes or hours, depending on the network status. Please wait until it's done.

⃗After installation is completed, the following prompts will appear:

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 (Robot HAT). After turning on again, the Raspberry Pi will automatically run the program to set the servos port signal to turn the servos to the middle position, which is convenient for mechanical assembly.

⃗ When installation is completed, the Raspberry Pi will automatically disconnect SSH and reboot. If you

used puTTy to connect the Raspberry Pi, there can be an error prompt like `Network error:Software caused connection abort`. You can just ignore and close it.

# Run the Raspberry Pi Robot's Program

⃗ Raspberry Pi auto runs the program for the robot when rebooting every time, which is the part

`[RobotName]/server/webServer.py` (replace `[RobotName]` for the name of the folder for your robot product's program). However, if the Raspberry Pi camera or RobotHAT is not connected, the `webServer.py` can't run well. It makes sense because the robot's program needs the camera and the chipset PCA9685. RobotHAT controls servo with PCA9685; the Raspberry Pi communicates with PCA9685 via I2C; so if Robot HAT is not connected to Raspberry Pi, a program error will occur when instantiating dependent libraries for PCA9685 due to communication failure.

⃗Power off the Raspberry Pi, connect camera module and RobotHAT, reboot it up and now `webServer.py`

can run.

⃗ Generally you don't need to manually run `webServer.py` since it's auto run by Raspberry Pi at every

boot.

⃗ Open a web browser (Google Chrome for example), in the address bar type in the IP address of the

Raspberry Pi, add `:5000` to the end as shown below, and press ‘enter’ to redirect to the web page of the Raspberry Pi:

http://192.168.3.157:5000/

⃗ If it fails to enter the page, log into the Raspberry Pi via SSH, type in the command below to end the

program auto run at boot to release resources, or else issues like camera initialization failure or occupied ports. sudo killall python3

⃗Type in the command below to run `webServer.py`:

sudo python3 adeept\_rasptank/server/webServer.py

⃗Check whether there's any error and solve them based on instructions in the **Q&A** section below.

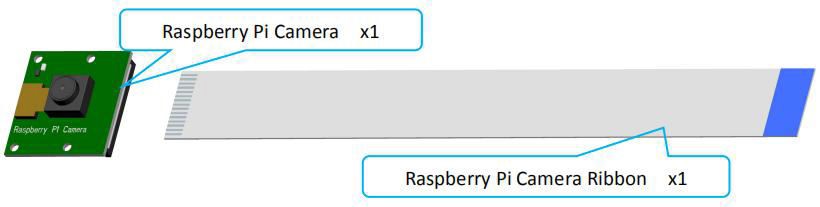
* 4 Raspberry Pi Structure Assembly and Precautions

# Documentation for structure assembly



Preparations before Assembly

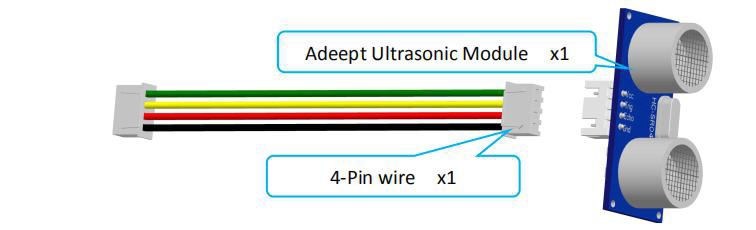
1. Connect the Raspberry Pi Camera and the ribbon. Assemble the following components



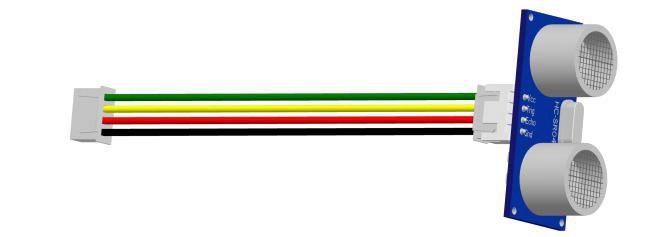
Effect diagram after assembling



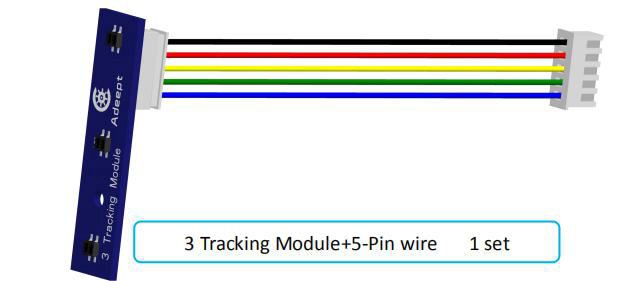
1. Connect the Adeept Ultrasonic Module with 4-Pin wire. Assemble the following components

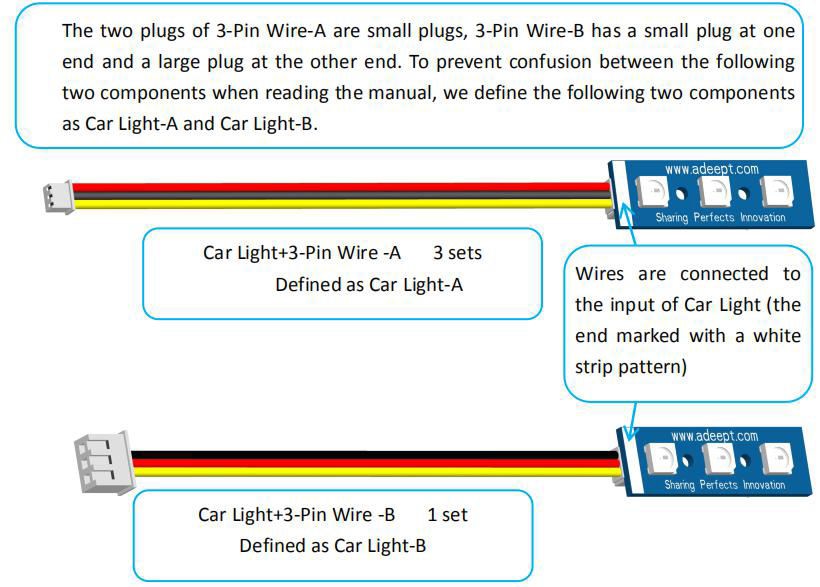


Effect diagram after assembling



The connection diagrams of the remaining modules and wires are as follows：



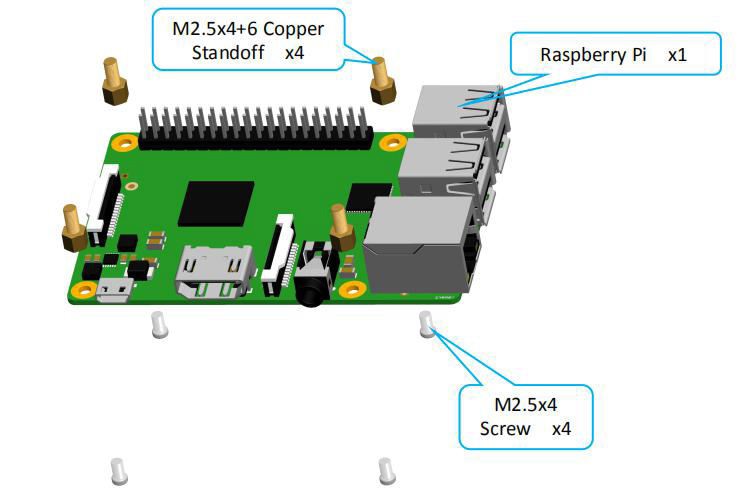


1. Screw the M4\*4 Locking Screw into the S12D4 Coupling (2 sets). Assemble the following components

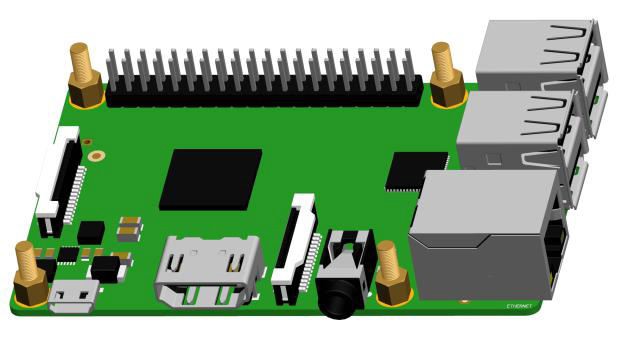


Servo debugging

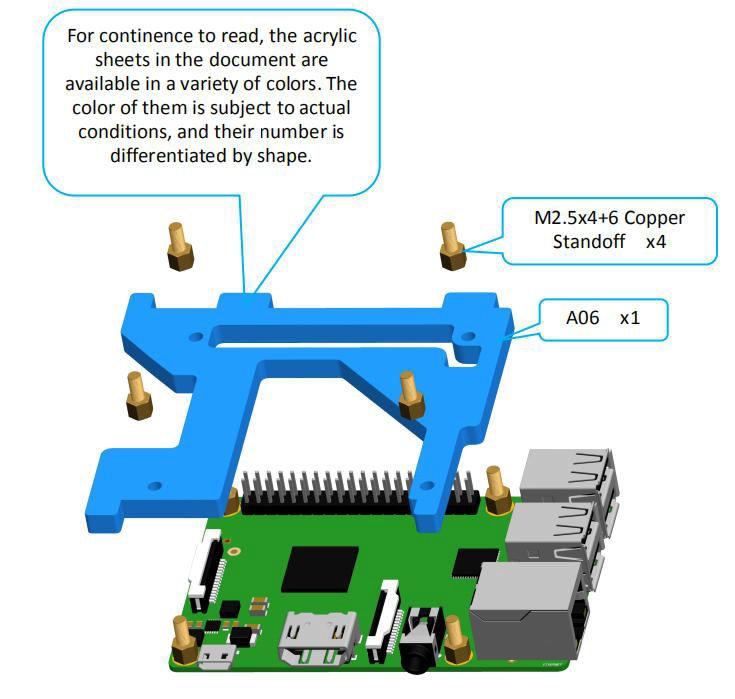
1. Fix four M2.5x4+6 Copper Standoffs on Raspberry Pi. Assemble the following components



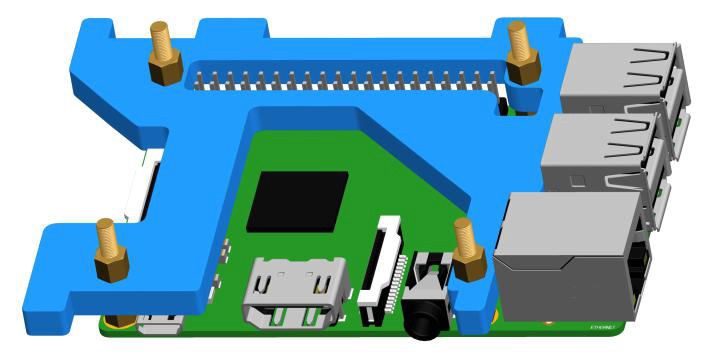
Effect diagram after assembling



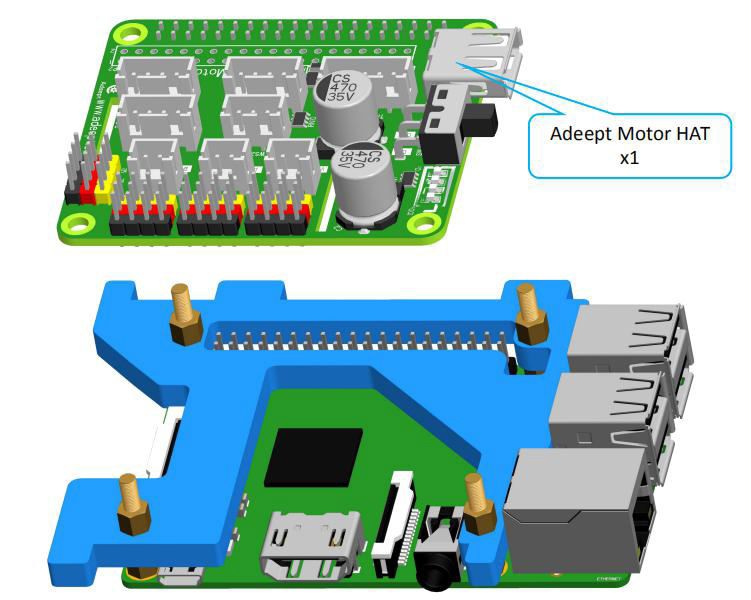
1. Fix A06 to Raspberry Pi. Assemble the following components



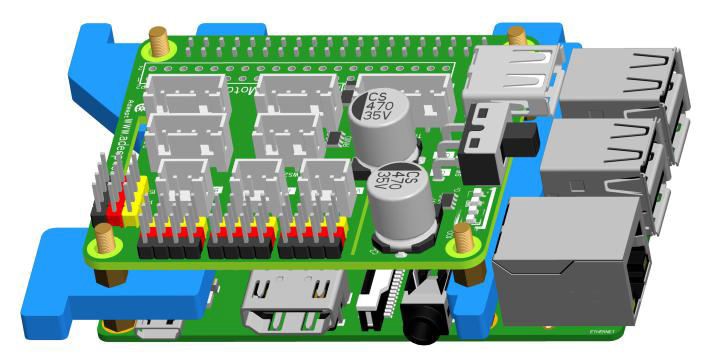
Effect diagram after assembling



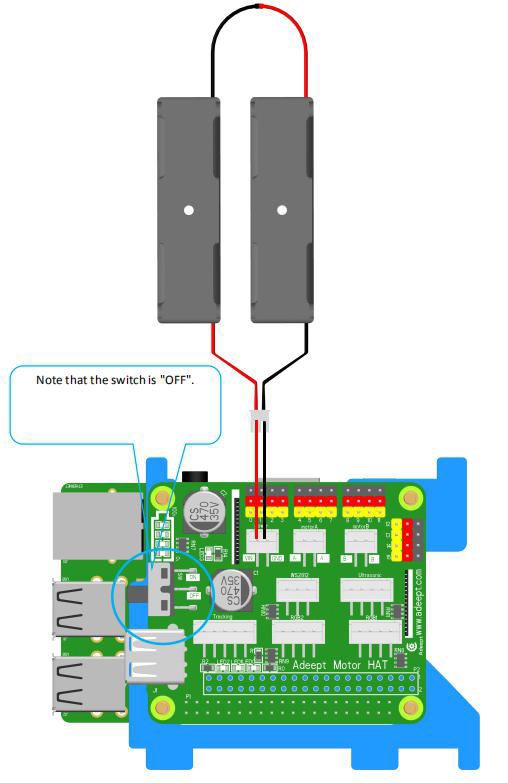
1. Insert the Adeept Motor HAT into Raspberry Pi. Assemble the following components



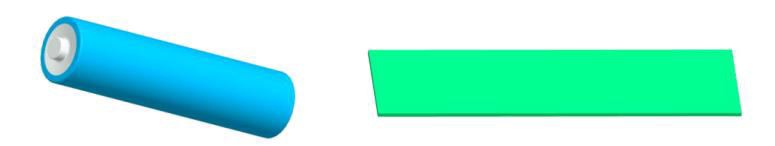
Effect diagram after assembling



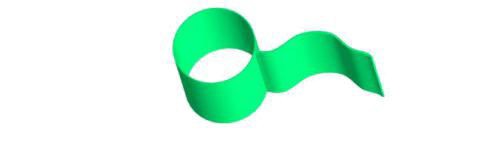
1. Connect the 18650 Battery Holder Set to the Adeept Motor HAT.



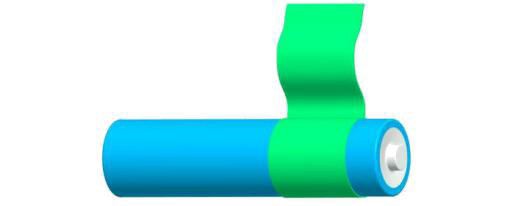
1. Put two 18650 batteries in 18650 Battery Holder Set according to the following method. Take out 1 ribbons and 1 batteries.



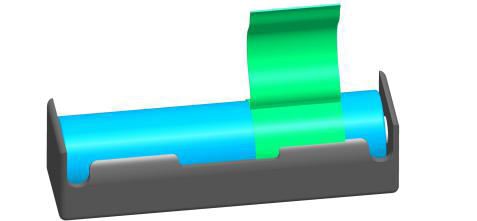
Roll one end of the ribbon to let through a battery and fix.



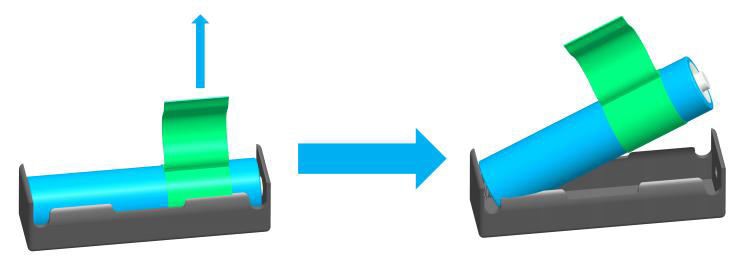
Insert the batteries into the rings - ribbon closer to the anode.



Install the batteries into the holder based on the pole.



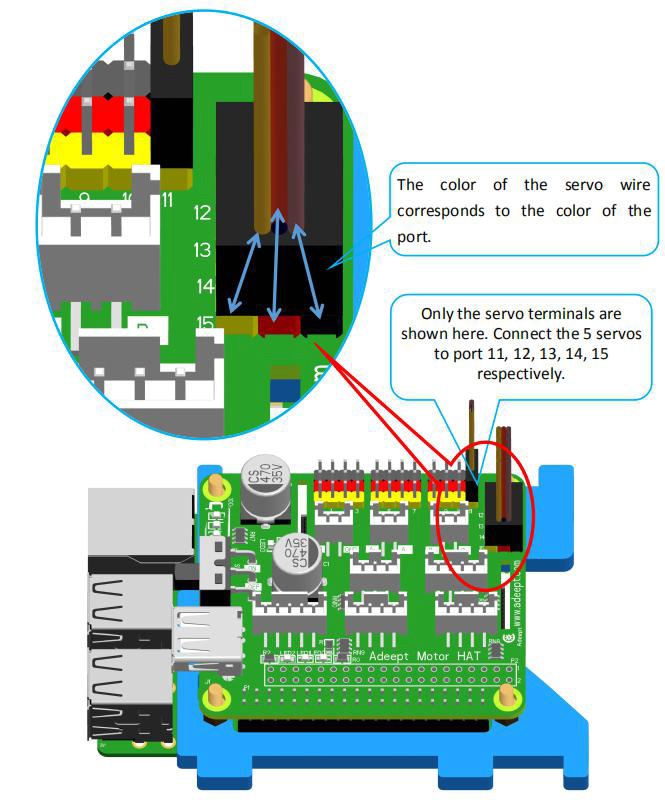
To remove the batteries, just pull the ribbon and take them out.



Effect diagram after assembling

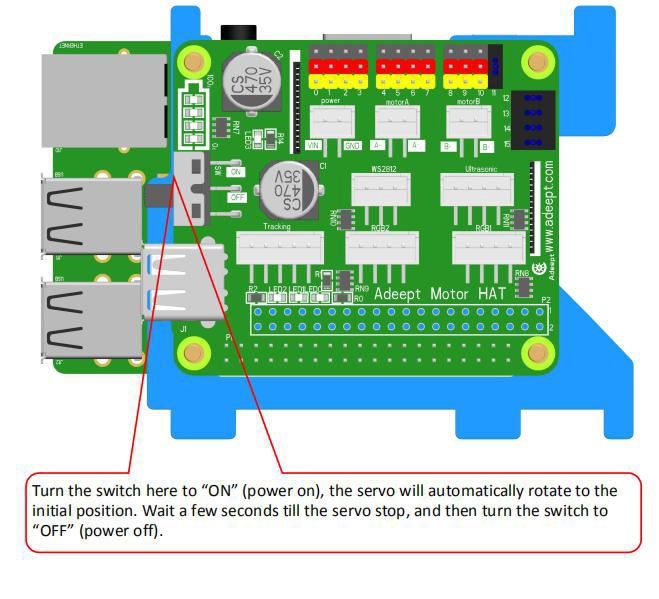


1. Connect five servos to Adeept Motor HAT.



1. Turn on the power and debug the servo.

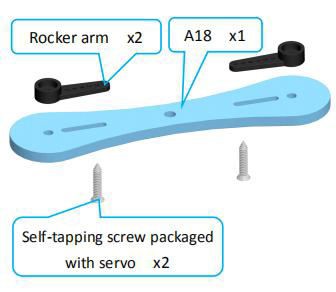
Before switching on, you need to insert the configured SD card into the Raspberry Pi. For details, please refer to the third chapter of the document. **That in the next operation, the Pi Camera of the Raspberry Pi should always be connected to the Raspberry Pi, and do not reverse the wires of the Raspberry Pi.** Otherwise, the servo will not rotate to the middle position after booting. If SD card is not inserted, the servo needs to be rotated to the middle position manually.



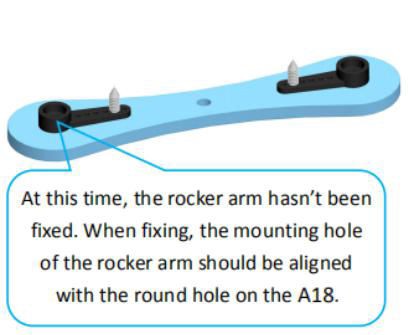
After debugging, remove the servo and battery holder, and take the 18650 batteries out of the Holder Set. Do not rotate the rotation axis before the servo fixed to the rocker arm. Otherwise, you need to re-debug the servo.

Rocker arm assembly.

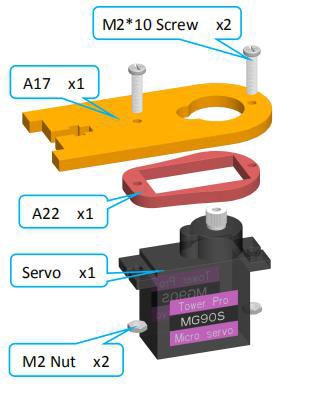
1. Take two rocker arms as in the illustration and connect them to A18. Assemble the following components



Effect diagram after assembling



1. Fix a debugged servo to A17. Assemble the following components

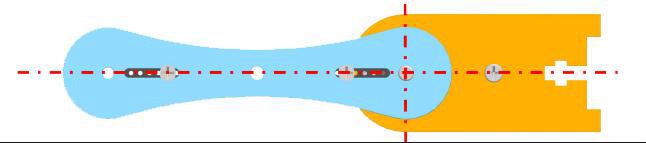


Effect diagram after assembling

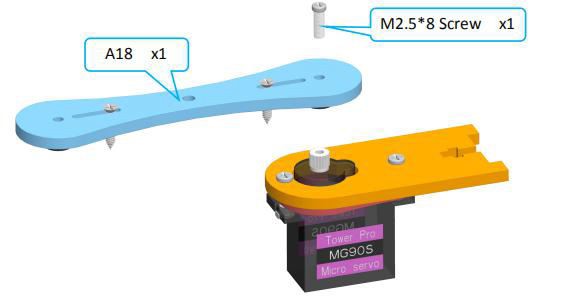


1. Then fix one end of A18 to the servo on A17.

First install the rocker arm on the A18 into the servo. When installing, the mounting hole of the rocker arm should be aligned with the round hole on the A18. Install it at the angle shown below.



Assemble the following components



Effect diagram after assembling



1. Fix a debugged servo to A20. Assemble the following components

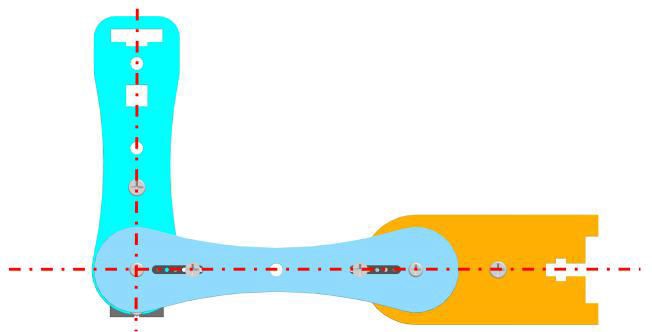


Effect diagram after assembling



1. Then fix the other end of the A18 to the servo on the A20.

First install the rocker arm on the A18 into the servo. When installing, the mounting hole of the rocker arm should be aligned with the round hole on the A18. Install it at the angle shown below.



Assemble the following components

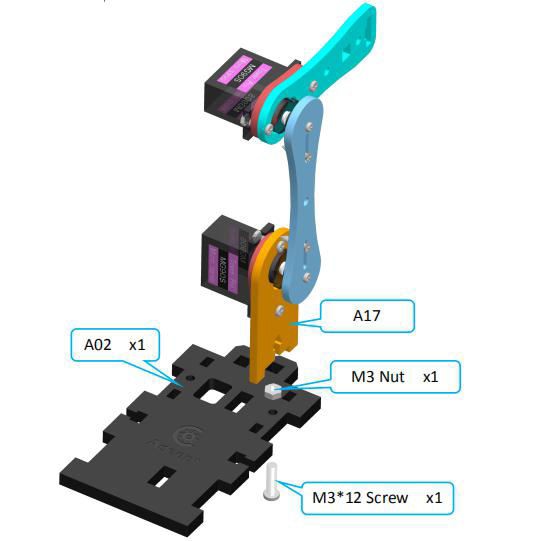


Effect diagram after assembling

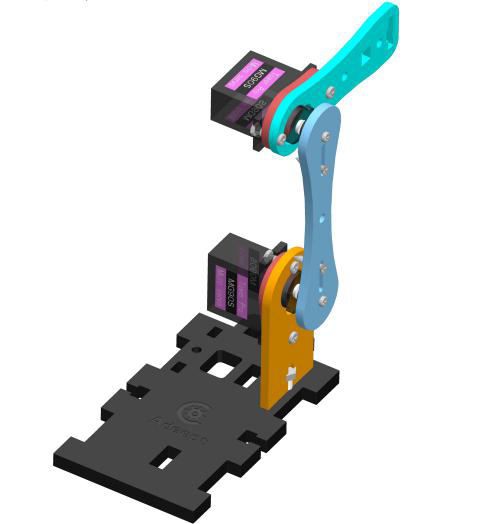


1. Then fix A17 to A02.

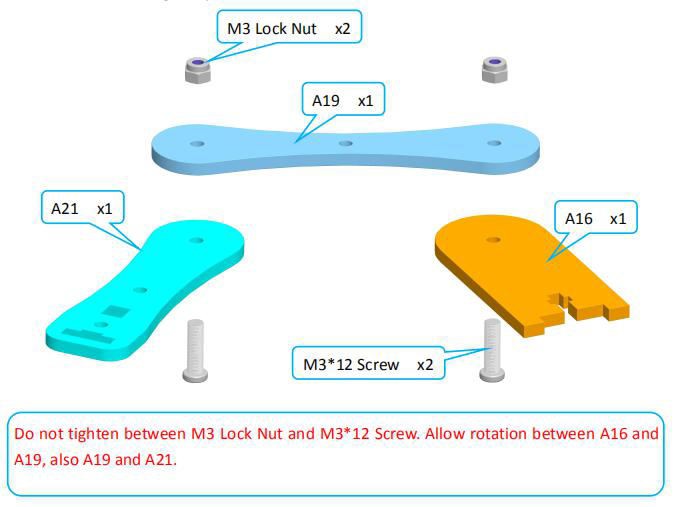
Assemble the following components



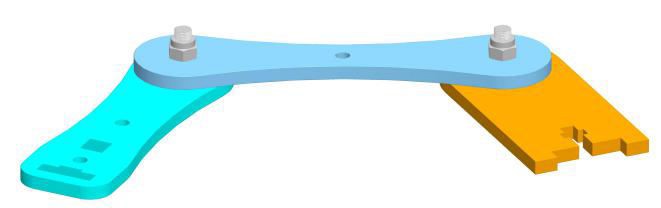
Effect diagram after assembling



1. Connect A16 with A19 and A21. Assemble the following components

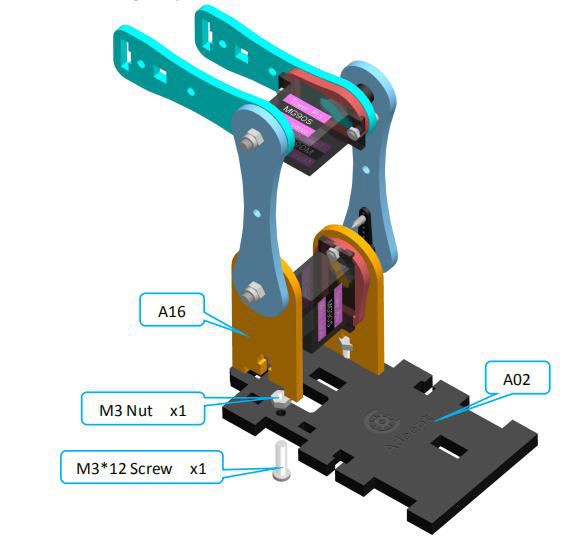


Effect diagram after assembling

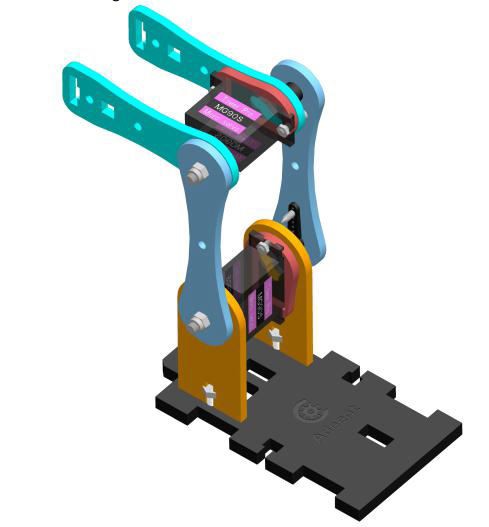


1. Fix A16 to A02.

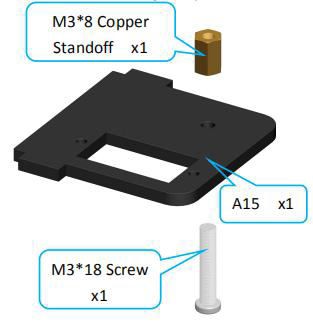
Assemble the following components



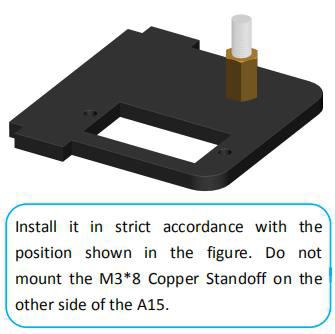
Effect diagram after assembling



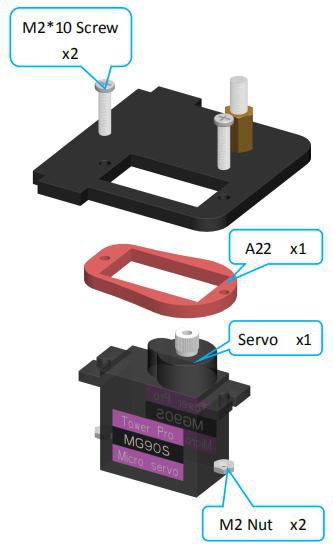
1. Fix one M3\*8 Copper Standoff to A24. Assemble the following components



Effect diagram after assembling



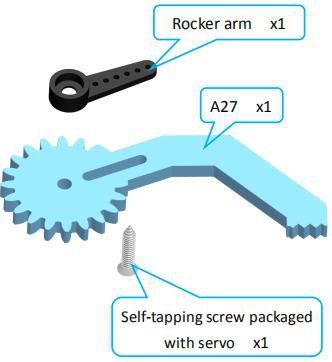
1. Fix a debugged servo to A24. Assemble the following components



Effect diagram after assembling



1. Fix one rocker arm of the servo to A27. Assemble the following components

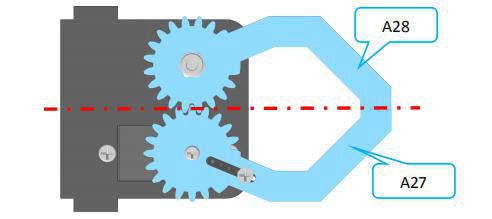


Effect diagram after assembling

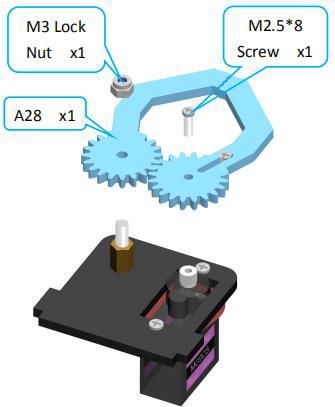


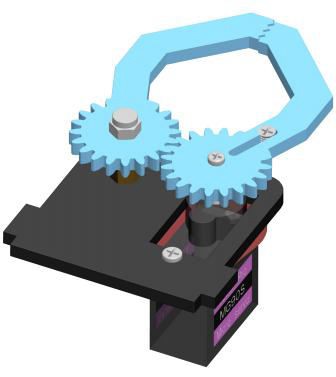
1. Assemble A27 and A28.

Install A27 and A28 as shown below.

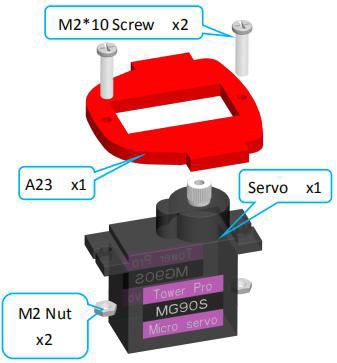


Assemble the following components



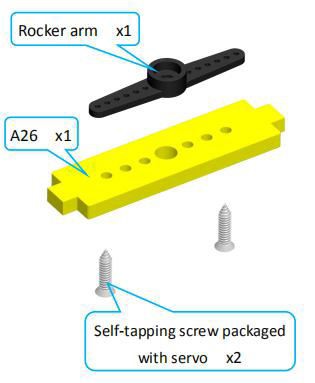


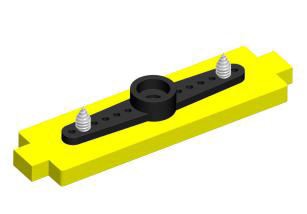
1. Fix a debugged servo to A23. Assemble the following components



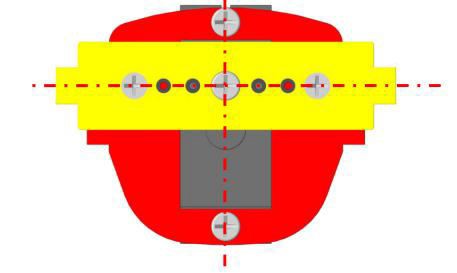


1. Fix a rocker arm to A26. Assemble the following components

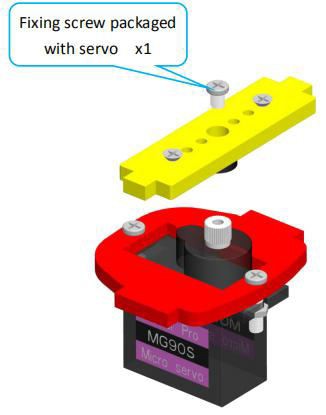




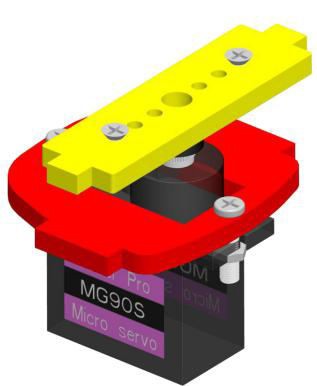
1. Fix the rocker arm on the A26 to the servo on the A23. Install as shown below.



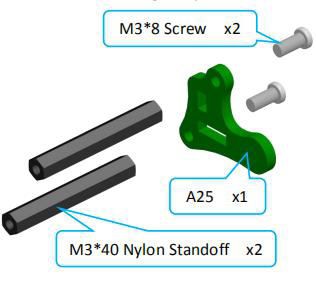
Assemble the following components



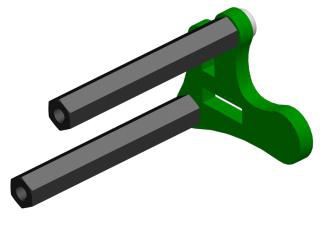
Effect diagram after assembling



1. Fix one A25 with two M3\*40 Nylon Standoffs. Assemble the following components



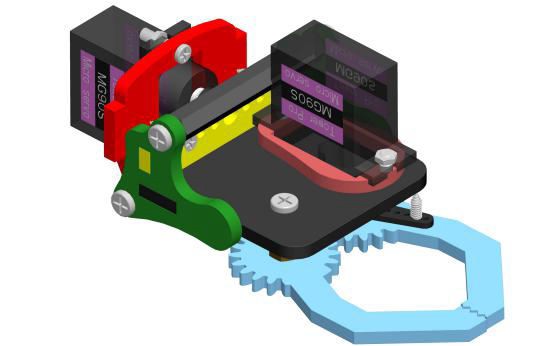
Effect diagram after assembling



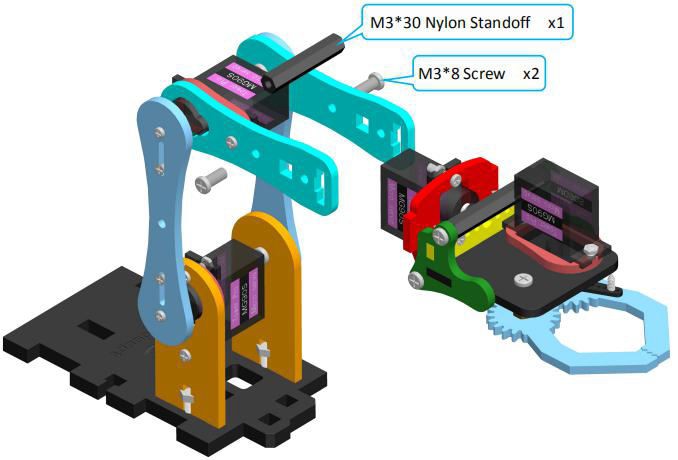
1. Complete assembly of the clamp section. Assemble the following components



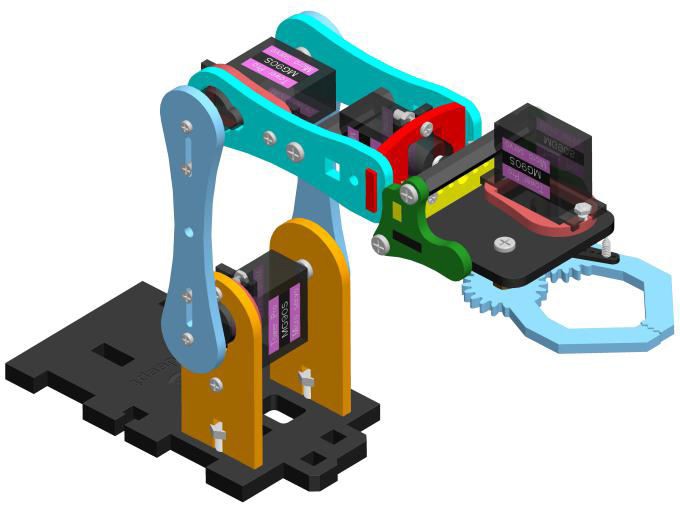
Effect diagram after assembling



1. Install the clamp section on the robotic arm. Assemble the following components

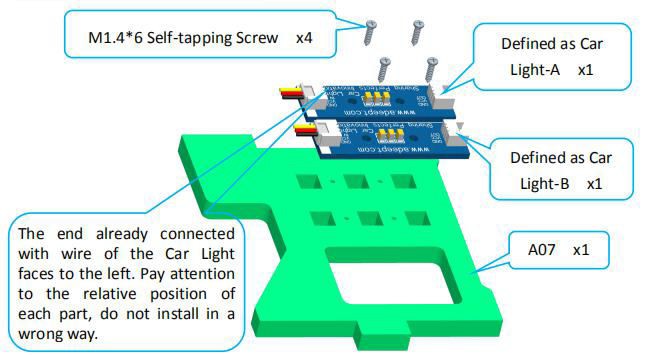


Effect diagram after assembling

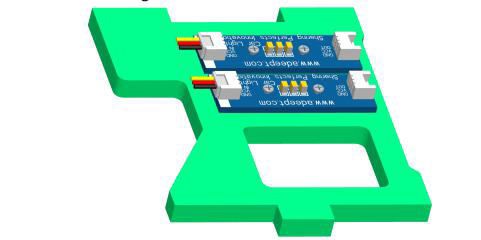


Body part assembly

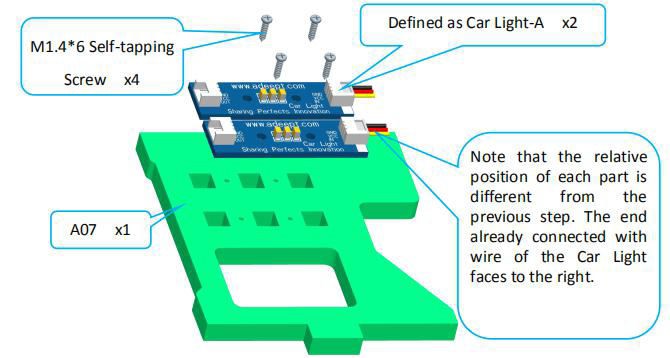
1. Fix two Car Lights on A07. Assemble the following components



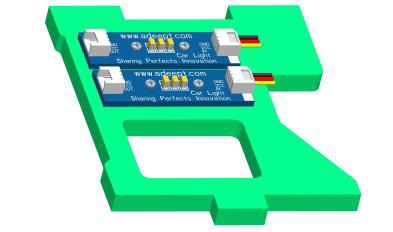
Effect diagram after assembling



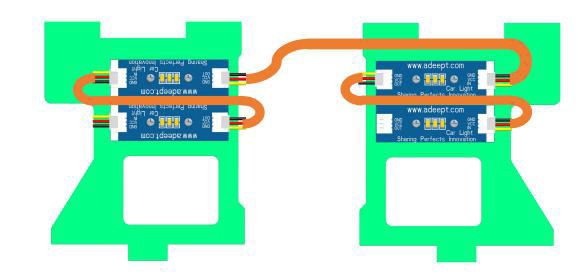
1. Fix two Car Lights on the other A07. Assemble the following components



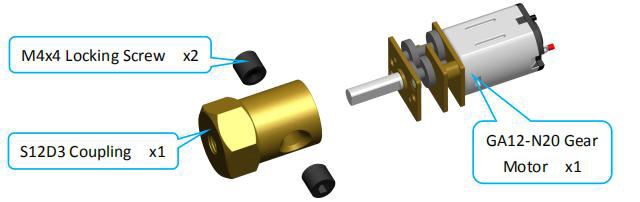
Effect diagram after assembling



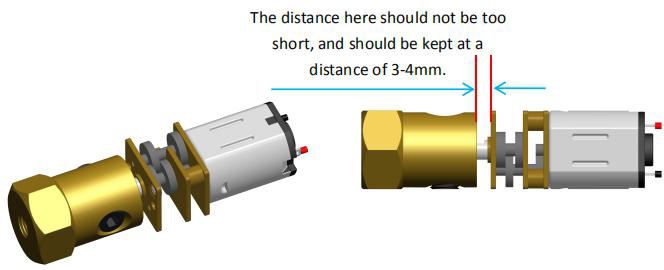
1. Connect the Car Light as shown below.



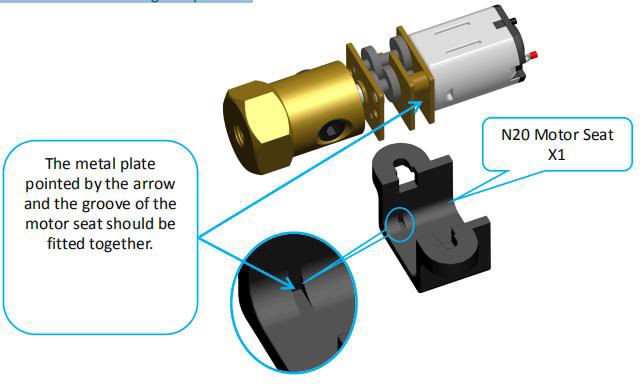
1. Fix S12D3 Coupling on GA12-N20 Gear Motor (2 sets). Assemble the following components



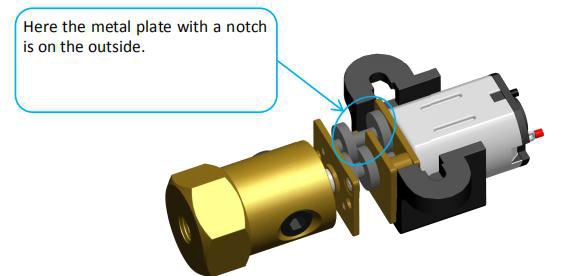
Effect diagram after assembling



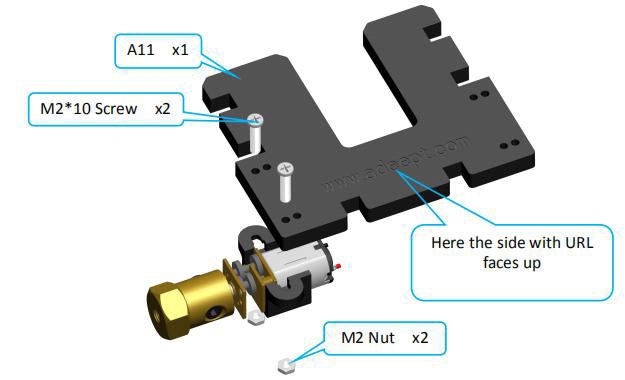
1. Put the GA12-N20 Gear Motor on N20 Motor Seat (2 sets). Assemble the following components



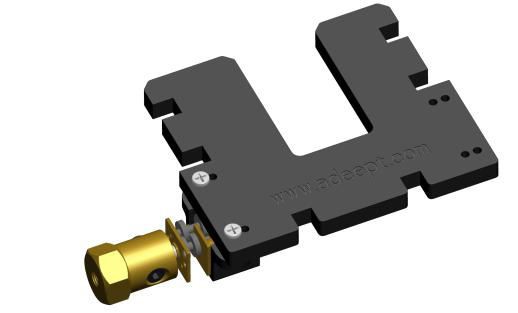
Effect diagram after assembling



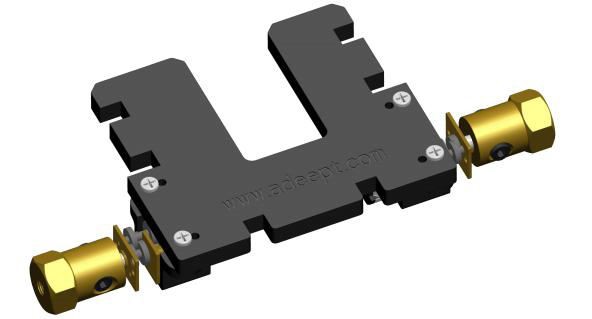
1. Fix N20 Motor Seat on A11. Assemble the following components



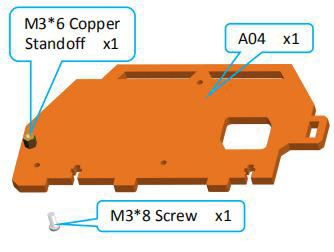
Effect diagram after assembling



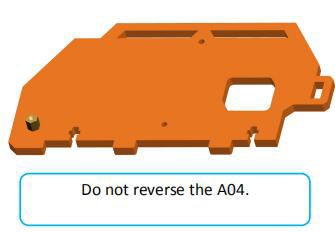
Fix the other N20 Motor Seat on A11.



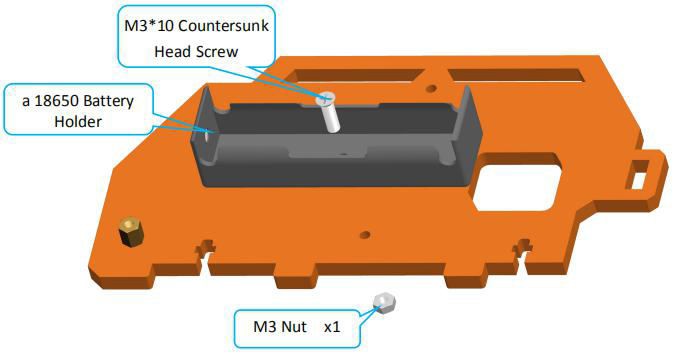
1. Fix one M3\*6 Copper Standoff on A04. Assemble the following components



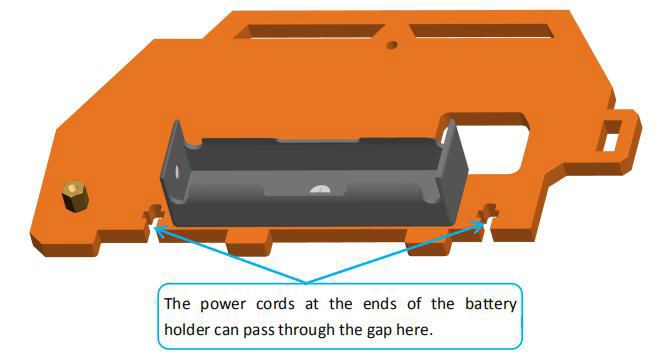
Effect diagram after assembling



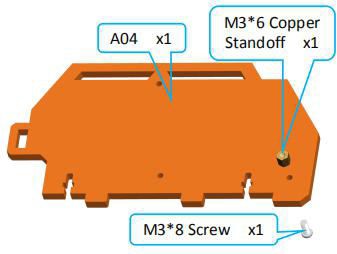
1. Fix one section of the 18650 Battery Holder Set to A04. Assemble the following components



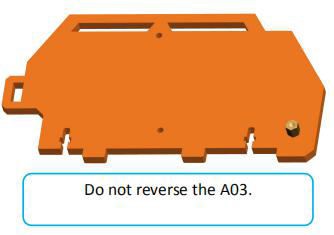
Effect diagram after assembling



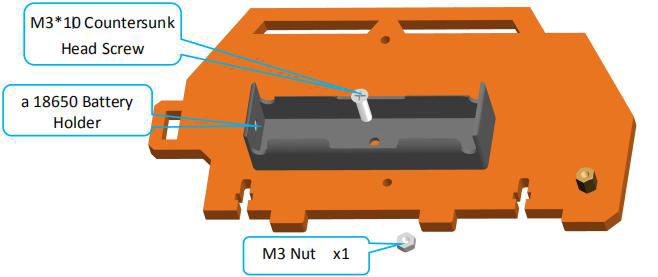
1. Fix one M3\*6 Copper Standoff on A03. Assemble the following components



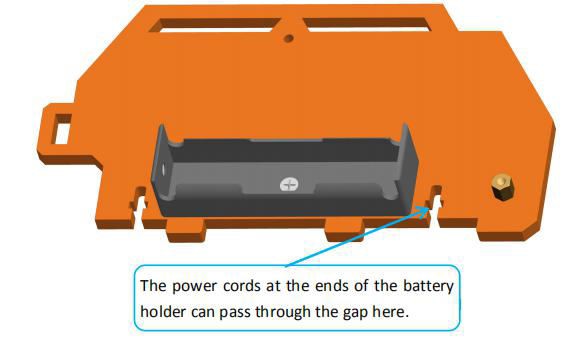
Effect diagram after assembling



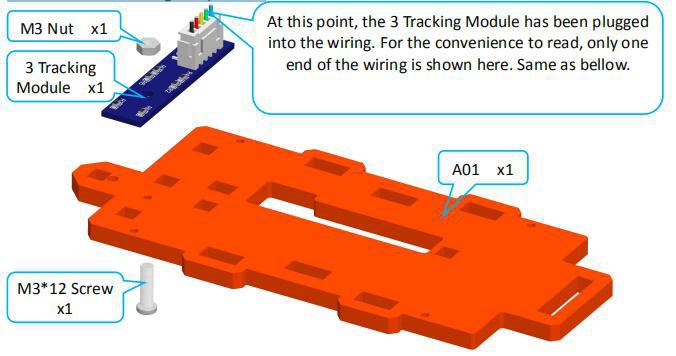
1. Fix the other section of the 18650 Battery Holder Set to A04. Assemble the following components



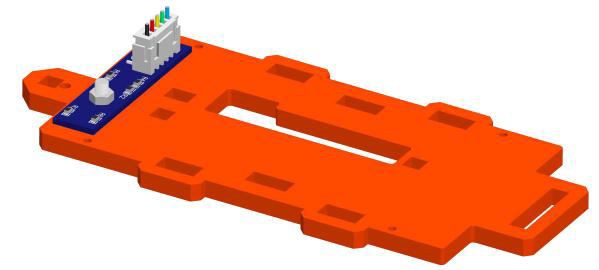
Effect diagram after assembling



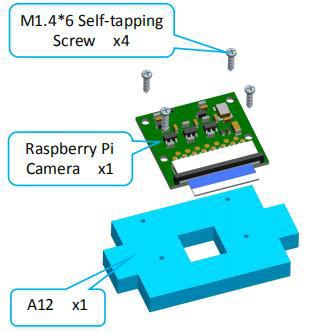
1. Fix 3 Tracking Module on A01. Assemble the following components



Effect diagram after assembling

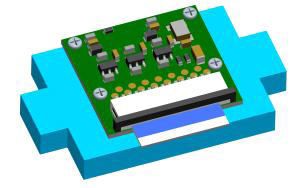


1. Fix Raspberry Pi Camera on A12. Assemble the following components



Effect diagram after assembling

When the screw is obstructed or cannot be screwed in, turn over the plate and screw in the other side.

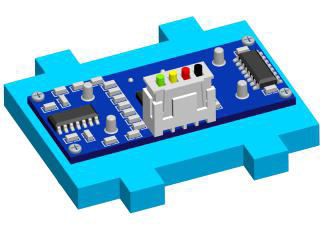


1. Fix Adeept Ultrasonic Module on A10. Assemble the following components

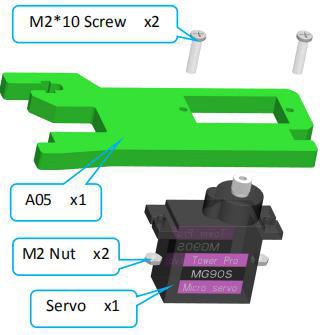


Effect diagram after assembling

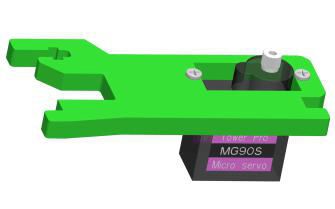
When the screw is obstructed or cannot be screwed in, turn over the plate and screw in the other side.



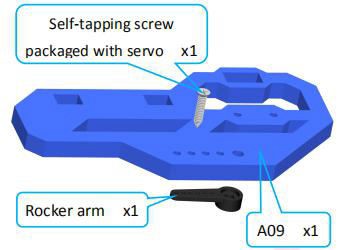
1. Fix a debugged servo to A05. Assemble the following components



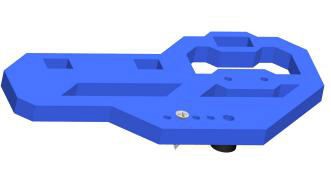
Effect diagram after assembling



1. Fix one rocker arm of the servo on A09. Assemble the following components

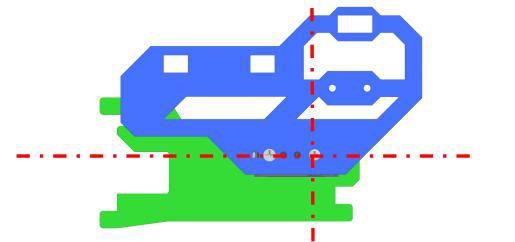


Effect diagram after assembling



1. Fix the rocker arm on A09 to the servo on A05.

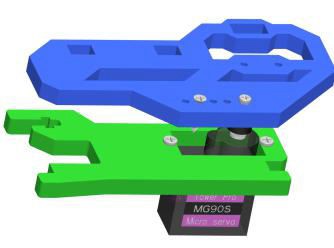
Buckle the rocker arm into the servo at the angle shown in the figure below.



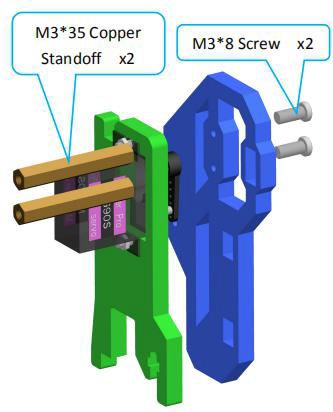
Assemble the following components



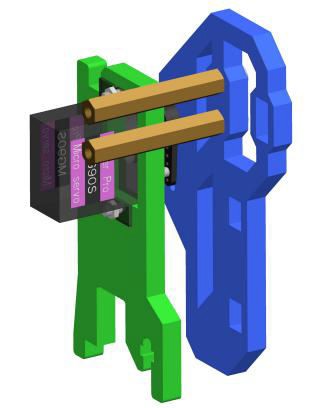
Effect diagram after assembling



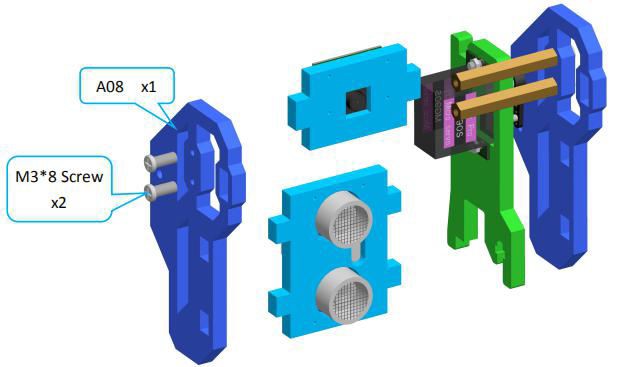
1. Fix two M3\*35 Copper Standoffs on A09. Assemble the following components



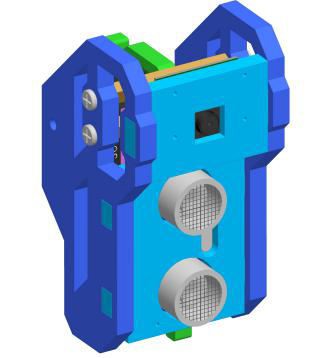
Effect diagram after assembling



1. Fix A10 and A12 between A08 and A09. Assemble the following components

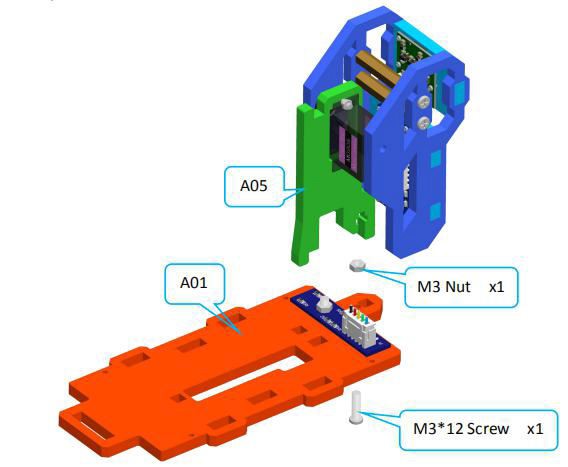


Effect diagram after assembling

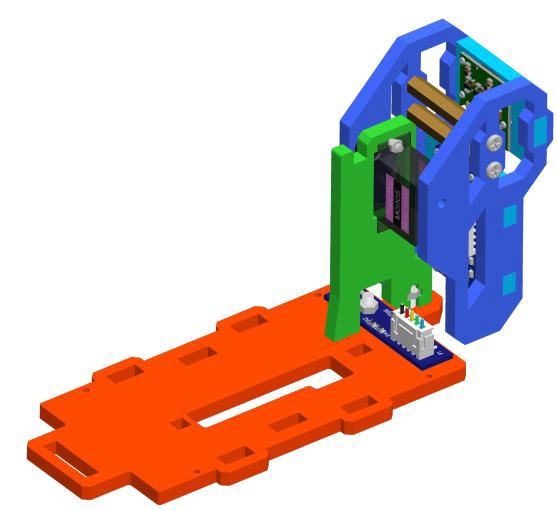


1. Fix A05 on A01.

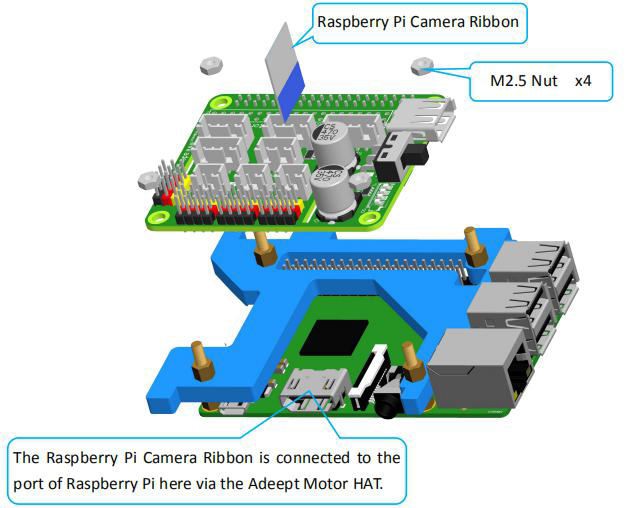
Assemble the following components



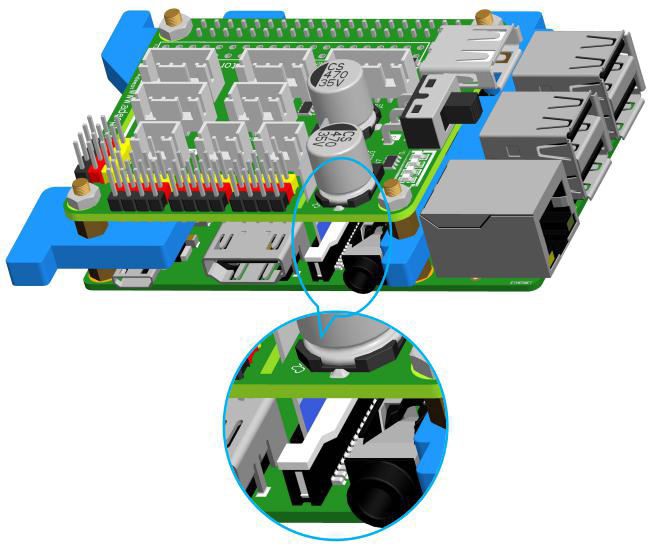
Effect diagram after assembling



1. Connect the other end of Raspberry Pi Camera Ribbon with Raspberry Pi. Assemble the following components

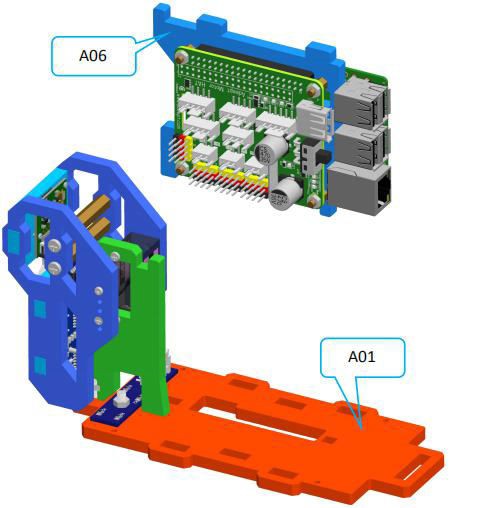


Effect diagram after assembling

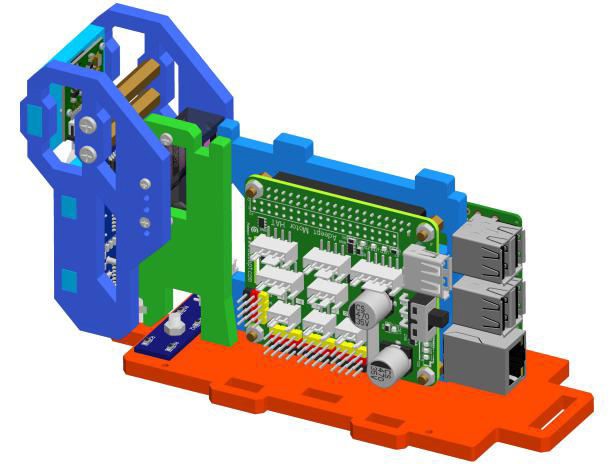


1. Insert A06 to A01.

Assemble the following components

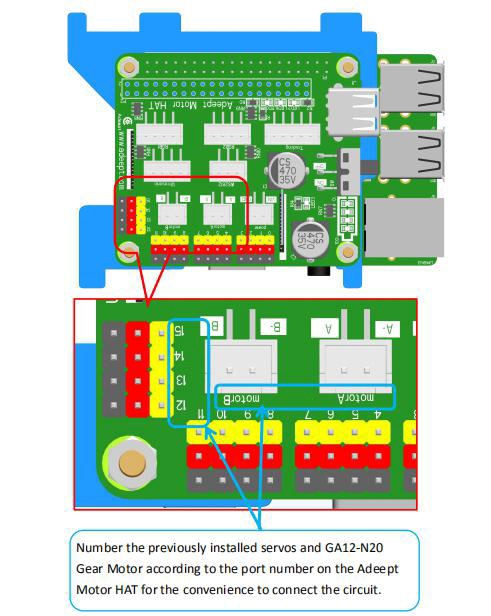


Effect diagram after assembling

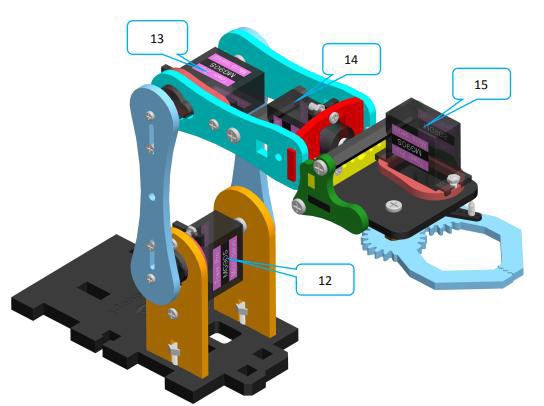


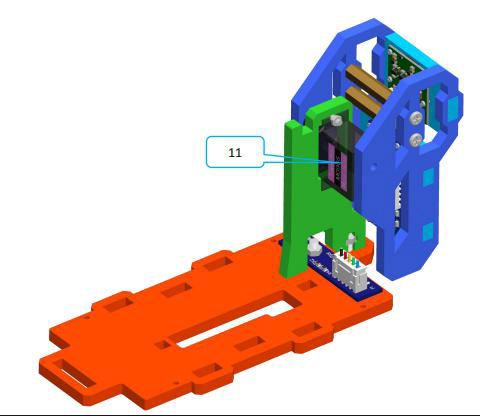
1. Connect the circuit.

Connect the servo with GA12-N20 Gear Motor.



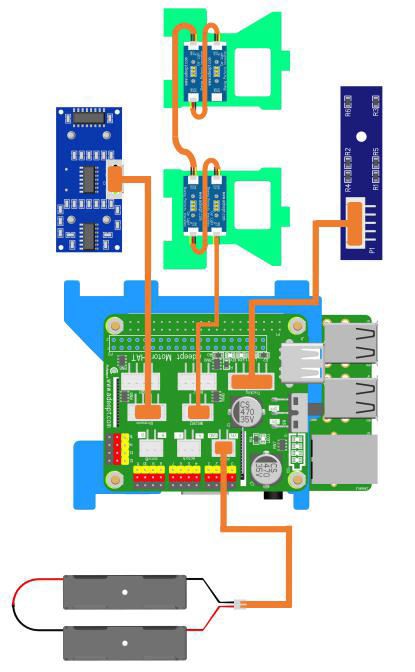






Connect the servo and the GA12-N20 Gear Motor to the corresponding port numbers of Adeept Motor HAT. Connect the Adeept Ultrasonic Module, Car Light, 3 Tracking Module and 18650 Battery Holder Set as shown

below.

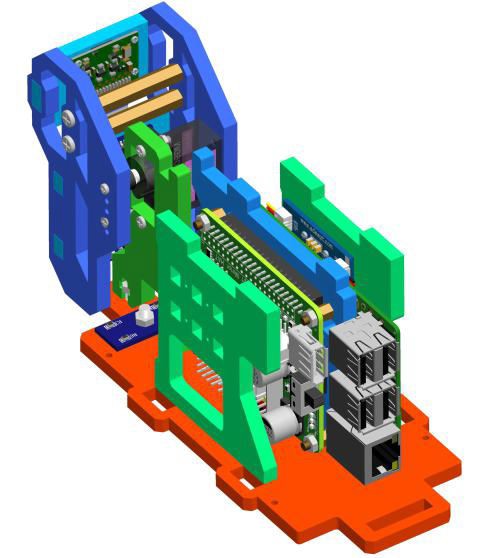


1. Insert A07 to A01.

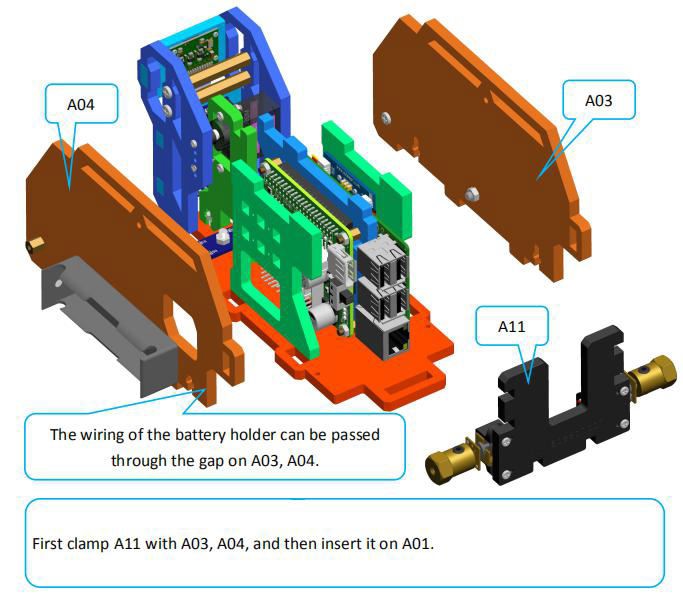
Assemble the following components



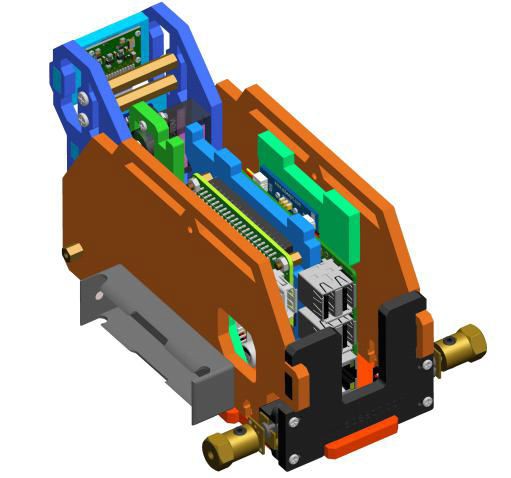
Effect diagram after assembling



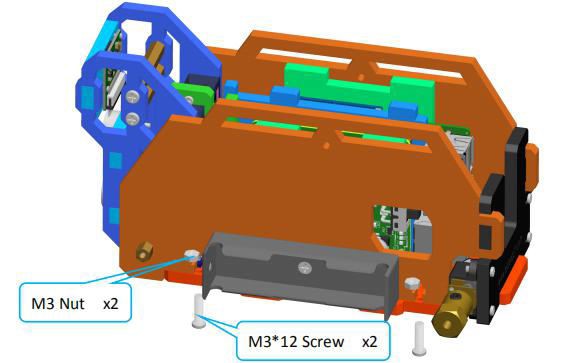
1. Insert A03, A04 and A11 on A01. Assemble the following components



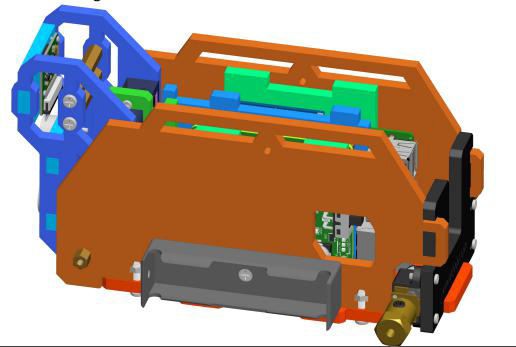
Effect diagram after assembling



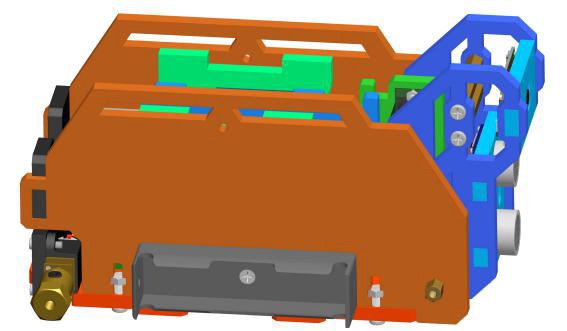
1. Fix A04 with M3\*12 Screw. Assemble the following components



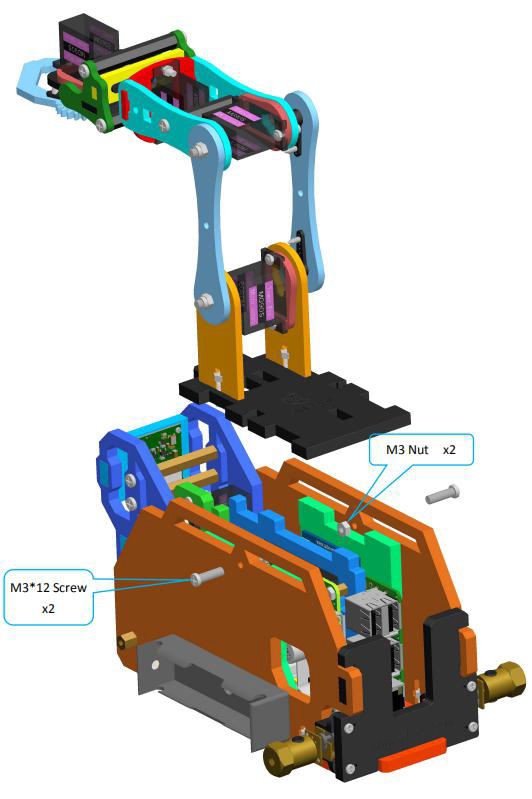
Effect diagram after assembling



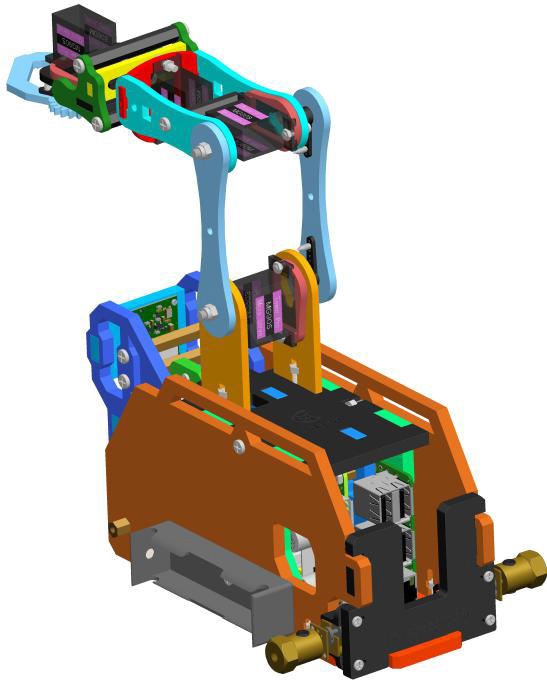
Fix A03 with M3\*12 Screw.



#### 26. Fix A02 to A03 and A04. Assemble the following components



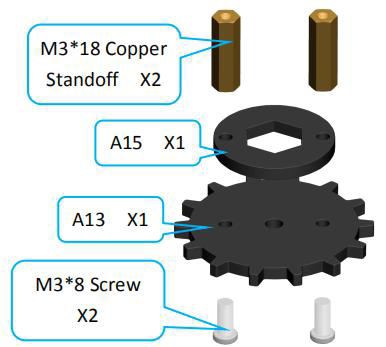
Effect diagram after assembling



Track assembly.

1. Fix A13 to A15 (4 sets).

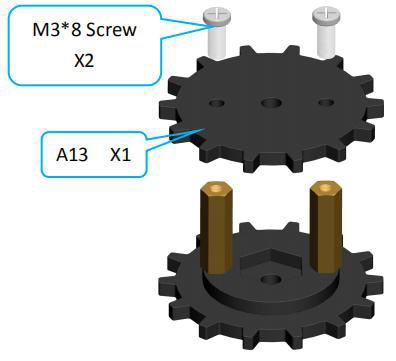
Assemble the following components



Effect diagram after assembling



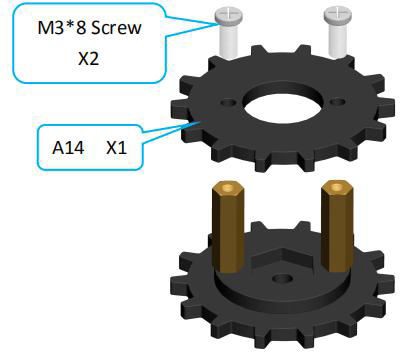
1. Assemble driving wheel (2 sets). Assemble the following components



Effect diagram after assembling



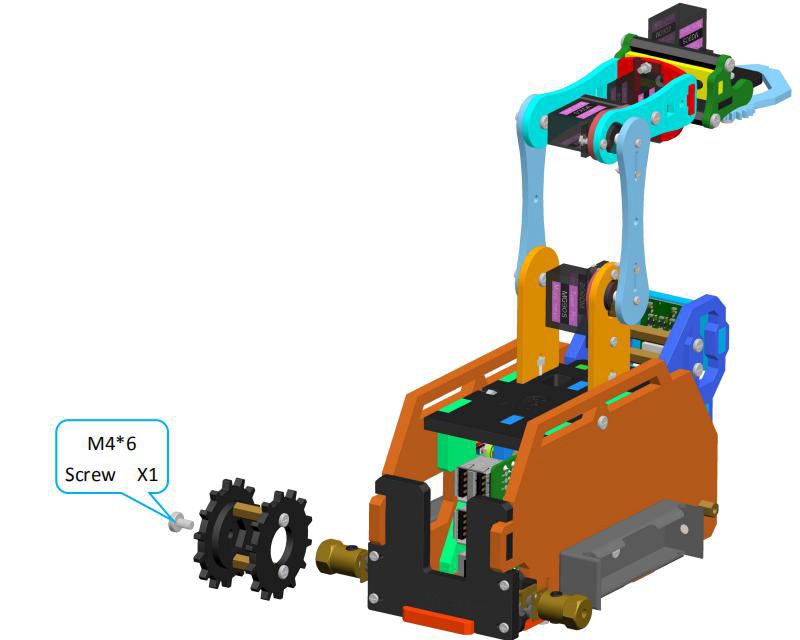
1. Assemble driven wheel (2 sets). Assemble the following components



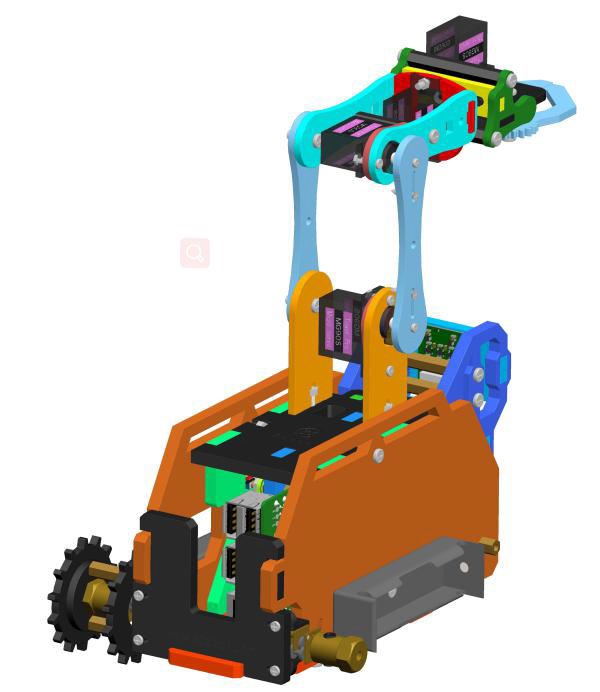
Effect diagram after assembling



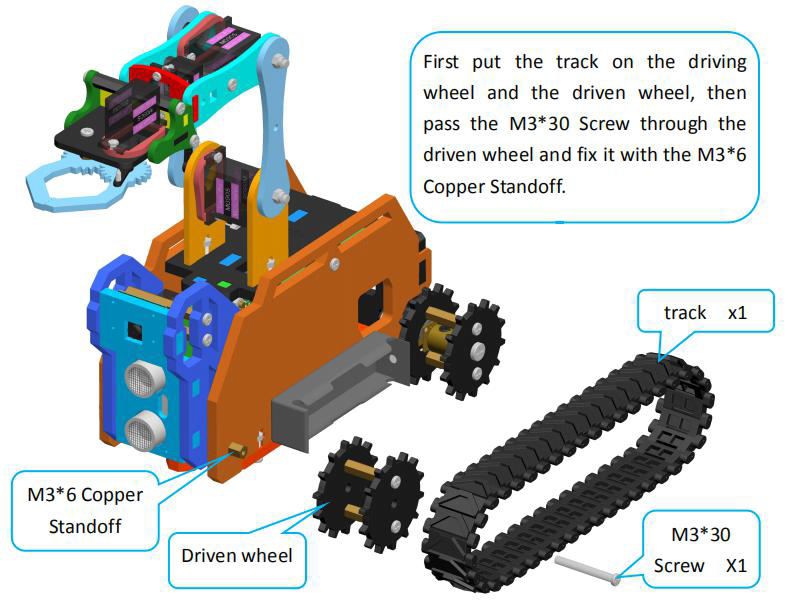
1. Install the driving wheel on the S12D4 Coupling. Assemble the following components



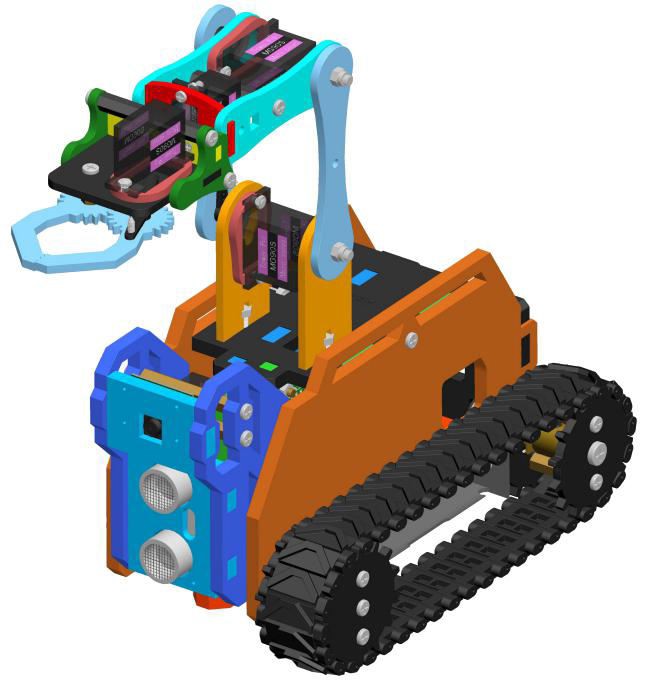
Effect diagram after assembling



1. Install the left track of the tank. Assemble the following components



Effect diagram after assembling



1. Install the track of the other side as described above. Effect diagram after assembling



# Tips for structural assemblage

⃗Since many servos are used in the product, the servo installation is critical for the robot. Before installing the rocker arm to the servo, you need to connect the servo to power and make the servo shaft rotate to the central position, so the rocker arm installed at the designated degree will be in the central position.

⃗Generally Raspberry Pi will auto run `webServer.py` when booting, when `webServer.py` will control all the

ports connected to servos to send a signal of rotating to the central position. When assembling the servo, you can connect it to any servo port anytime. After connecting the servo to the port, the gears will rotate to the central position; assemble the rocker arm to the servo, disconnect the servo from the port, and insert more servos to repeat rocker arm assembly (all servos will be in the central position).

⃗ When the servo is connected to power, try moving the rocker arm. If it can't be moved, it indicates the

program for the servo works; otherwise there's error for the servo program. Run the line

`[RobotName]/initPosServos.py` (replace `[RobotName]` with the folder name of your robot's program) to make the servo rotate to the central position.

⃗When booting (it may take 30-50s), it takes a while for the Raspberry Pi to control PCA9685 to set signal of

all servo ports for central position rotating.

# Tips for power provision

⃗When you install the software, assemble the structure and debug the program, you can use a USB cable to power the raspberry pi. If the raspberry pi is installed with Robot HAT, you can connect the USB cable to the USB interface on the Robot HAT. Robot HAT will supply power to the raspberry pi through the GPIO interface.

⃗ The demand of different raspberry pi for electric current is different. For example, the raspberry pi 3B

needs at least 2A of current to start up, and the raspberry pi 4 needs 3A to start up normally. You can check the specifications on your power adapter before you use the power adapter to power the raspberry pi.

⃗When Robot HAT is connected to a load (such as connecting to a motor or multiple servos), you need to

use a power supply that supports high current to connect to **Vin** on Robot HAT. You can use two 18650 batteries that support high current to power Robot HAT. Our product will provide a dual 18650 battery box with 2pin interface, you can directly connect it to Robot HAT.

⃗ When you use the USB interface on Robot HAT to supply power, Robot HAT's switch does not control

whether to supply power, Robot HAT's switch can only control the power supply of **Vin**.

⃗Do not use the USB interface and Vin on the Robot HAT 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 set the switch on the Robot HAT to OFF. So, when using a USB cable to connect Robot HAT, Robot HAT is powered by USB.

⃗If your robot restarts automatically after starting up or the robot disconnects and restarts suddenly when it starts moving after been started up normally, it is most likely because your power supply does not output enough current. When the robot is starting up, it will automatically run the program to put all the servos in the neutral position. The voltage drop caused by this process causes the raspberry pi to restart.

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

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

⃗You can use a USB cable to supply power to Robot HAT when the installing the rocker arm of the servo during structural assembly. After the robot software is installed, the raspberry pi will control Robot HAT to set all servo ports to output the neutral signal after it is started up. At this time, you can connect the servo to any servo port, the servo gear will turn to the neutral position, and then you can install the rocker arm of the servo according to the specified angle. After the rocker arm is installed, the servo can be disconnected from Robot HAT. When you need to install the rocker arm of the second servo, you only need to connect the second servo to any servo port on the drive board.

* 5 Controlling Robot via WEB App

⃗The WEB app is developed for common users to control the robot in an easier way. It's convenient to use

WEB app; you may use it to wirelessly control the robot on any device with a web browser (Google Chrome was used for testing).

⃗Generally Raspberry Pi will auto run `webServer.py` when booting and establish a web server in the LAN.

You may then use any other computer, mobile or tablet in the same LAN to visit the web page and control the robot.

⃗How to tell whether the robot has run the `webServer.py` or not: If the WS2812-LED lights up with the

breathing effect, it means the robot has booted and runs the program automatically.

⃗If the program is not run when the robot is booted, try to connect Raspberry Pi via SSH, manually run

`webServer.py` with code and check the errors. Refer to the **Q&A** below or email us for help (before manually running `webServer.py`, you need to end the program possibly auto run in the back end to release resources.

sudo killall python3

sudo python3 [RobotName]/server/webServer.py

⃗If the `webServer.py` is auto run successfully, open a web browser (here Google Chrome), type in the IP address of the Raspberry Pi, with `:5000` added to the end, and go to the next step, as shown below:

192.168.3.157:5000

⃗If no image is displayed, try manual running `webServer.py` as described in the step above.

⃗If image is shown, you can control the robot to move now. You may check the description for keyboard shortcuts `Instruction` at the bottom and control the robot based on its general functions with the keyboard.

⃗`Video` window shows the image captured by the robot's camera in real time.

⃗`Move Control` window is to control the basic movements of the robot.

⃗`Arm Control` Gwindow controls the servo movement.

·`**RAB**` `**LOOSE**`: Control the claws of the robotic arm to open and close

·`**HANDUP**` `**HANDDOWN**`: Control the robotic arm to move up and down

·`**LEFT**` `**RIGHT**`: Control the claws rotation of the robotic arm

⃗`CVFL Control` window is to control the visual line following function of the robot. Here only an overview for the function is described; more details will be provided in the OpenCV section:

·`**START**`: Enable or disable the visual line following function.

·`**COLOR**`: Switch between white and black line following. By default the robot follows white lines; click the button to switch to black line following.

·The line following function analyzes two pixels in parallel and utilizes the information detected; the positions of these two pixels are `L1` and `L2`.

·`SP` is the threshold of the turning command based on the visual analysis results. A bigger `SP` value means a big deviation; though a particularly small `SP` value may stop the robot from moving as it can't aim the target and find the direction.

⃗ When the visual line following function is enabled, the video screen will automatically become

binarized results, making the visual analysis clearer.

⃗ `Hard Ware` window displays CPU temperature, CPU occupancy rate, and memory usage of the Raspberry Pi.

⃗`**Actions**`window control unique functions of the robot:

·`**MOTION GET**`: Motion detection function based on OpenCV. When objects move in the view of the camera, the program will circle the part in the `Video` window, and the LED light on the robot will show respective changes.

·`**AUTO MATIC**`: Obstacle avoidance function based on ultrasonic. When the ultrasonic module on the robot detects an obstacle, it will automatically turn left, and take a step backward before turning if it's too close to the obstacle.

·`**POLICE LIGHT**`: WS2812-LED light control based on multi-threading. It makes the WS2812-LED light on the robot blink red and blue alternately.

⃗`**TRACK LINE**`: Line tracking function by using the 3-channel infrared module. By default it tracks black

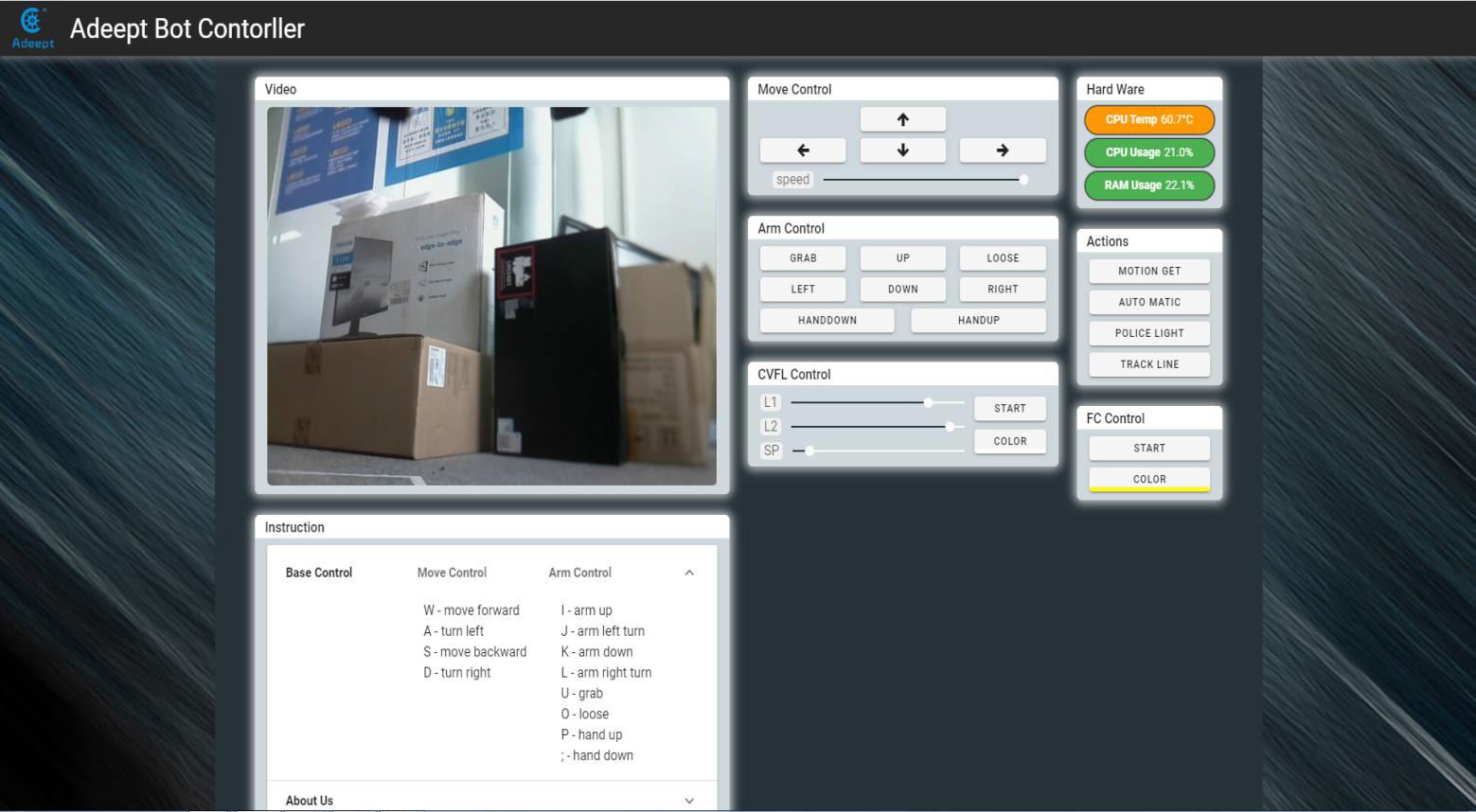
lines on a white surface (a white background that reflects infrared, and 1-cm wide black lines that do not reflects infrared). Performance of the line tracking varies from surface and line materials as well as the height of the robot chassis; you may need a cross screwdriver to adjust the potentiometer on the line tracking module.

⃗`**FC Control**` window control`s the color lock function of the robot:

·`**START**`: Enable or disable color searching and tracking function.

·`**COLOR**`: Select the color to track.

·When the function is on, the robot will automatically lock one particular color in the camera view. By default it tracks bright yellow objects. You can change the color as you want. When an object is locked, the LED on the robot will turn orange. As the robot's head can only move up and down, the program does not involve tracking colors horizontally. If you have interest in this part, you may add the motor control based on the openCV section to realize effect.



* 6 Common Problems and Solutions(Q&A)

⃗Where to find the IP address of the Raspberry Pi?

Before connecting the Raspberry Pi via SSH, you need to know the IP address of the Raspberry Pi. Check the Management interface for your router, or download the app `Network Scanner` -> search for a device named `RASPBERRY` or `Raspberry Pi Foundation` to get the IP address.

For other methods of obtaining the IP address of Raspberry Pi, refer to the official documentation [IP Address]

⃗Errors occur with `permission denied` prompt when I manually run `server.py` or `webServer.py`.

The Raspberry Pi needs the root permission to run the dependent libraries for WS2812 LED lights control. You need to add `sudo` to the beginning of `server.py` or `webServer.py` to run the program.

sudo python3 [PATH]/server.py sudo python3 [PATH]/webServer.py

⃗I can't create the hots pot for the robot.

You need to use the open source project create\_ap to setup the robot's hotspot. Prior to use, disconnect WiFi network but DO NOT turn the WiFi module off, or the create\_ap will show an error of hardware being blocked.

⃗The servo rotates to an abnormal degree.

Before assembling the rocker arm and servo, you need to make the servo gears rotate to the central position of its rotating range. Then assemble the rocker arm based on the angle instructed in the documentation. There can be a deviation of less than 9° due to the structure of the servo (number of teeth is 20 for the servo gears). For better performance, you may refer to the servo control documentation for initial degree adjustment by code.

⃗The servo is shaking.

Probably the servo reducing gear is broken.

⃗Raspberry Pi can't boot.

Remove all parts on the driver board. Only connect the board to Raspberry Pi and power supply, reboot.

⃗"Remote side unexpectedly closed network connection" shows on a popup window.

There can be error prompts during installation because the Raspberry Pi will auto reboot after the installation, which will disconnect the board.

⃗Program crashes after double clicking on client.py or GUI.py.

Run the script by `python client.py` or `python GUI.py` in cmd and check the error reports. The GUI.py file is in adept/rasptank/server/GUI.

⃗How to initialize the servo's angle?

If you've finished software installation on the Raspberry Pi, just boot it up and the servo will be initialized.

⃗I can connect to the Raspberry Pi terminal via SSH \ Raspberry Pi failed to connect a WiFi.

The power supply methods will not influence control by SSH. Check whether you've created the file

`wpa\_supplicant.conf` for multiple times. If yes, that's problem causing SSH errors.

⃗Can I supply the Robot HAT and Raspberry Pi via USB?

A 2A output is required for a Raspberry Pi 3B, when at least 3A is needed for a Raspberry Pi 4. You can use the USB power for software installation and testing, but it's not suitable for high power module like servo or motor adjustment as it may result in low voltage. It's recommended to use battery for power here.

⃗After installation, the robot shows no response when booting.

The `server.py` or `webServer.py` may not run due to some reasons. Try to manually run `server.py` or

`webServer.py` and check whether there's any error prompt.

⃗The servo doesn't return to the central position when connected to the driver board.

In general, the Raspberry Pi will auto run `webServer.py` when booting, and `webServer.py` will run and control the servo ports to send a signal of rotating to the central position. When assembling the servo, you can connect it to any servo port anytime. After connecting the servo to the port, the gears will rotate to the central position; assemble the rocker arm to the servo, disconnect the servo from the port, and insert more servos to repeat rocker arm assembly (all servos will be in the central position).

When the servo is powered on, try moving the rocker arm. If it can't be moved, it indicates the program for the servo works; otherwise there's error for the servo program. Run the line `[RobotName]/initPosServos.py` (replace

`[RobotName]` with the folder name of your robot's program) to make the servo rotate to the central position.

When booting (it may take 30-50s), it takes a while for the Raspberry Pi to control PCA9685 to set signal of all servo ports for central position rotating.

⃗no cv2 error occurs when I manually run `server.py` or `webServer.py`.

OpenCV is not installed correctly. Type in the command sudo pip3 install opencv-contrib-python in the Raspberry Pi to manually install OpenCV.

⃗When using a computer to copy ssh and wpa\_supplicant.conf to the SD card, it prompts that there is no

SD card

If this happens, unplug the card reader and connect it to the computer.

⃗SSH can't connect, error WARNING: REMOTE HOST IDENTIFICATION HAS CHANGED!

Enter the following in the command line and press Enter ssh-keygen -R Add the Raspberry Pi's IP address

For example:

ssh-keygen -R 192.168.3.157

Then you can SSH to the Raspberry Pi again

⃗Raspberry Pi automatically restarts after booting / restart the robot once it starts to move

If your robot automatically restarts after powering on, or disconnects and restarts when the robot starts to move after normal power on, it is likely because your power supply does not output enough current, and the robot will automatically run the program when it starts Put all the servos in the neutral position, the voltage drop caused by this process causes the Raspberry Pi to restart.

We have tested that when using 7.4V power supply, the peak current of the robot is about 3.75A, so you need to use support 4A output battery.

⃗The direction of servo movement is incorrect

Due to the different batches of servos, when the same angle change trend is given to the servos, the actual direction of motion of the servos may be opposite. We have set an interface to adjust the direction of the servos in the program. You need to open RPIservo.py, Find the array sc\_direction in ServoCtrl. If the direction of the servo of port 3 is reversed, change the fourth 1 to -1.

(The serial number of the array starts from zero, so port 3 corresponds to the fourth 1).

If the servo direction of port 3 is not correct: Before modification:

self.sc\_direction = [1,1,1,1, 1,1,1,1, 1,1,1,1, 1,1,1,1]

After modification (the serial number of the array starts from zero, so port 3 corresponds to the fourth 1): self.sc\_direction = [1,1,1,-1, 1,1,1,1, 1,1,1,1, 1,1,1,1]

⃗Motor movement direction is incorrect

Due to the different batches of motors, when the same signal is given, the direction of rotation of the motor may be different. We have set an interface to adjust the direction of rotation of the motor in the program. You need to open move.py. In the program part, you can see To the following variable definitions:

|  |  |  |
| --- | --- | --- |
| Dir\_forward | = | 0 |
| Dir\_backward = | 1 |  |
| left\_forward | = | 0 |
| left\_backward = | 1 |  |
| right\_forward | = | 0 |
| right\_backward | = | 0 |

If all your motor actions are reversed, just change Dir\_forward = 0 to Dir\_forward = 1,Just change Dir\_backward

= 1 to Dir\_backward = 0.

If you only have one motor reversed, you only need to change the corresponding set of variables.

⃗After running the server, I get an error and can't find config.txt

This is because the installation script did not copy con ﬁ g.txt to the specified location due to permissions problems during installation. The new version of webServer will not use this file, only the old version of the server will use it. Copy the server folder of the Raspberry Pi to / etc / of the Raspberry Pi, use the following command

sudo cp -f //home/pi/adeept\_rasptank/server/conﬁg.txt //etc/conﬁg.txt

Just replace adeept\_rasptank above with your product name, we will take Rasptank as an example here.

⃗Running GUI.py reports ip.txt NotFound error

The correct mode of operation is to execute GUI.py directly in the GUI directory. If you execute GUI.py in another directory, it will look for ip.txt in your current directory.

* **7 Set The Program to Start Automatically**

# Set the specified program to run automatically at boot

⃗This section only introduces the auto-run method used by our products. If you need more information about the Raspberry Pi auto-run program, you can refer to this document from itechfythe document Auto-Run.

⃗If you have used the operation steps of 3.5 or 3.6, then the script program has been configured to

automatically run the program at startup. In this chapter, we explain how to set a program to start automatically at startup from scratch.

⃗First we use the following code to create a new startup.sh:

sudo touch //home/pi/startup.sh

⃗Edit startup.sh

sudo nano startup.sh

⃗Write the following content in startup.sh, where python3 is followed by the program you want to run automatically. Note that you must use an absolute path here. Let's take webServer.py as an example.

#!/bin/sh

sudo python3 [RobotName]/server/webServer.py

⃗After **Ctrl** + **X** To exit editing. Press Y Save. **Enter** Confirm and exit editing.

⃗Give startup.sh permissions, where \*\*\* is the Linux permission code, we do not recommend the use of permissions such as 777, but for novices 777 can avoid many account and permissions problems, of course, you can also set it to 700 , So that only the owner can read, write and execute startup.sh, you can learn more about Linux permissions through this article from maketecheasier the article link Understanding File Permissions.

sudo chmod 777 //home/pi/startup.sh

⃗Edit rc.local to configure the script to run automatically

sudo nano /etc/rc.local

⃗Add the following content under **fi** in the original document, save and exit:

//home/pi/startup.sh start

⃗ Of course, you can also replace the above script file path with other scripts you want to run automatically.

# Change the program that starts automatically

⃗After step **7.1**, you can already set the program to run automatically at boot. If you want to change the program to run automatically at boot, just edit startup.sh:

sudo nano //home/pi/startup.sh

⃗For example, if we want to replace webServer.py with server.py, we only need to edit the following:

Replace

sudo python3 [RobotName]/server/webServer.py

with

sudo python3 [RobotName]/server/server.py

⃗Save and exit so that the robot will automatically run server.py instead of webServer.py the next time the robot is turned on.

⃗server.py is a socket server used when using pythonGUI. We do not recommend it to novices here, because you need to manually install a lot of dependent libraries in the computer that controls it to allow the GUI to communicate with it normally. It is recommended to use the WEB application to control the Raspberry Pi robot.

**8**

**Remote**

**Operation**

**of**

**Raspberry**

**Pi**

**Via**

**MobaXterm**

⃗To make daily use of the Raspberry Pi more convenient, we usually do not connect peripherals such as

mouse, keyboard, and monitor to the Raspberry Pi. Since our Raspberry Pi is installed inside the robot, often with peripherals to control the Raspberry Pi, the efficiency of programming and testing will be seriously affected. Therefore, we introduce a method of programming in the Raspberry Pi.

⃗There are many ways to program in the Raspberry Pi. For example, you can use **3.x to log in to the**

**Raspberry Pi** without using a third-party tool. You can also create files in the Raspberry Pi. Almost all operations can use SSH to connect to the Raspberry Pi in the terminal, but for many people, it will be a disappointing experience when a lot of codes are written in the terminal. This chapter introduces a method that can facilitate the transfer of files to the Raspberry Pi. This method can directly edit programs in the Raspberry

Pi.

⃗This method requires the third-party software MobaXterm,Website address

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

control is available when SSH is on. For Raspberry Pi's method of enabling SSH and automatically connecting to WIFI, please refer to steps **2.2** and **2.3**.

⃗Download and install MobaXterm.

⃗ To obtain the IP address of the Raspberry Pi, you can refer to the method of **3.x and log into the Raspberry Pi** in this document to obtain the IP address of the Raspberry Pi.

⃗To run MobaXterm, firstly, create a new session, click Session in the upper left corner, click SSH in the

pop-up window, fill in the IP address of the Raspberry Pi behind Remote host, and finally click OK, the default

account name of the Raspberry Pi is pi , The default password is raspberry. Just the password doesn't appear on the screen when you enter it and the \* number doesn't mean nothing **Enter** successfully, press after login to log in to the Raspberry Pi, MobaXterm will remind you to save the password.You need to choose.

⃗If the user name and password are correct, you can change the user name and password according to

the prompt in the terminal, which is more secure.

⃗After the success of the login, MobaXterm will automatically save the conversation, when connected to

the raspberry pie again next time only need to double click on the left side of the IP address can be connected to the Raspberry Pi again, if there is no save username and password will need to input the user name and password, if the IP address of the Raspberry Pi changed, you need to start a new dialogue.

⃗ After a successful login, the left column is replaced with a file transfer system, which allows you to

interact with the system inside the Raspberry Pi. If you want to return to session selection, just click Sessions.

⃗Programs you write on other devices can be transferred to the Raspberry Pi by simple drag and drop,

and then the Raspberry Pi can be controlled in the terminal to execute the program, or the files in the raspberry Pi can be dragged to other devices.

⃗If you want to use another IDE to edit files in Raspberry Pi, you can find the file you want to edit in the file

transfer system on the left side of the MobaXterm. Right-click on this file and select your IDE so you can use your favorite on other devices IDE to edit the Raspberry Pi file, after editing is completed **CTRL**+**S** save the file.

⃗ However, it should be noted that when you use MobaXterm's file transfer system to edit files in the

Raspberry Pi, you need to pay attention to the permissions problem, because the file transfer system does not have root permissions, so if you are prompted to save after editing the file The permission denied error causes the file cannot be saved after editing. You need to use the following command to give the file you want to edit permission to be edited by MobaXterm:

sudo chmod 776 [FileName]

⃗You can learn more about Linux permissions through maketecheasier article from the article link Understanding File Permissions⃞

* **9 How to Control WS2812 RGB LED**

⃗WS2812 LED light is a commonly used module on our robot products. There are three WS2812 lights on

each module. Please pay attention when connecting. The signal line is different in direction, which needs to be connected to WS2812 after being led out from the Raspberry Pi. The IN end of the LED lamp module, when the next WS2812 LED module needs to be connected, the signal line is led out from the OUT end of the previous WS2812 module and connected to the IN end of the next WS2812 LED.

⃗When using the Raspberry Pi with the driver board RobotHAT installed, the WS2812 LED module 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 light, you can learn more about this project on GitHub.

⃗If you connect the WS2812 LED module to Robot HAT's WS2812 interface, the signal line is equivalent to the Raspberry Pi On GPIO 12, information about the pin number of the Raspberry Pi can refer to this official document GPIO

⃗Use the following command to install rpi\_ws281x for the Raspberry Pi. Since the Raspberry Pi has two versions of Python built in, the Python3 code is used as an example, so pip3 is adopted to install the library.

pip3 install rpi-ws281x

⃗ Next, we will explain the program. This program is written in the Raspberry Pi and executed in the Raspberry Pi. For the specific method, you can refer to **8 Programming in the Raspberry Pi.**

⃗Import dependencies

import time

from rpi\_ws281x import \*

⃗Construction of LED control class

class LED:

def init (self):

self.LED\_COUNT = 16 # Set to the total number of LED lights on the robot product.There are more LED lights on the Raspberry Pi

self.LED\_PIN = 12 # Set to the input pin number of the LED group

self.LED\_FREQ\_HZ = 800000

self.LED\_DMA = 10

self.LED\_BRIGHTNESS = 255

self.LED\_INVERT = False self.LED\_CHANNEL = 0

# Use the configuration item above 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, R, G, B): # This function is used to change the color of the LED light

color = Color(R, G, B)

for i in range(self.strip.numPixels()): # Only one LED light color can be set at a time, so we need to do a loop

self.strip.setPixelColor(i, color)

self.strip.show() # The color will only change after calling the show method

⃗Instantiate the object and execute the method function. The function colorWipe () needs to pass in three

parameters, namely R, G, and B, which correspond to the brightness of the three primary colors of red, green, and blue. The value range is 0- 255, the larger the value, the higher the brightness of the corresponding color channel. If the values of the three color channels are the same, white light is emitted. Specific examples are as follows:

if name == ' main ':

LED = LED()

try:

while 1:

LED.colorWipe(255, 0, 0) # All the lights turn red time.sleep(1)

LED.colorWipe(0, 255, 0) # All lights turn green time.sleep(1)

LED.colorWipe(0, 0, 255) # All lights turn blue time.sleep(1)

except:

LED.colorWipe(Color(0,0,0)) # Turn off all lights

⃗The above code will control all WS2812 lights to cycle through the three colors, press **CTRL**+**C** to exit the program.

⃗If you want to control the color of a single lamp, you can use the following code, where i is the serial

number of the lamp, the serial number of the first lamp connected from the driver board is 0, and the second lamp is 1. By analogy, R, G, B are the brightness of the corresponding three color channels:

LED.strip.setPixelColor(i, Color(R, G, B))

LED.strip.show()

⃗Note: You must use the Color () method to pack the RGB values and then pass them to setPixelColor ()

* **10 How to Control The Servo**

# Control the steering gear to rotate to a certain angle

⃗ Since the servo can use the PWM signal to control the rotation angle of a mechanism, it is a more

commonly used module on robot products. Walking robots, robotic arms and gimbals are all driven by the servo. In our Raspberry Pi The driver board Robot HAT has a dedicated PCA9685 chip for controlling the servo. The Raspberry Pi uses I2C to communicate with the PCA9685. You only need to install the Raspberry Pi

driver board Robot HAT on the Raspberry Pi, and the Raspberry Pi will be connected to the PCA9685. No other wires are required for connection.

⃗ The Raspberry Pi uses Python code to control the steering gear, and requires third-party libraries

Adafruit\_PCA9685, Adafruit-PCA9685Project address, if you run the installation script of the robot software, you do not have to manually install it again, if you do not have the security of running the robotTo install the script, use the following command to install Adafruit\_PCA9685 for Python3 in the Raspberry Pi:

sudo pip3 install adafruit-pca9685

⃗After the installation, you can use the Python3 code in the Raspberry Pi to control the servo:

import Adafruit\_PCA9685 # Import the library used to communicate with PCA9685 import time

pwm = Adafruit\_PCA9685.PCA9685() # Instantiate the object used to control the PWM pwm.set\_pwm\_freq(50) # Set the frequency of the PWM signal

while 1：# Make the servo connected to the No. 3 servo port on the Robot HAT drive board reciprocate pwm.set\_pwm(3, 0, 300)

time.sleep(1) pwm.set\_pwm(3, 0, 400) time.sleep(1)

⃗In the above code, set\_pwm\_freq (50) is used to set the PWM frequency to 50Hz. This setting depends

on the model of the servo. The servo used by our robot product needs to be controlled by a 50Hz PWM signal. If you use other The value of the servo needs to be set by referring to the specific servo documentation.

⃗pwm.set\_pwm (3, 0, 300) This method is used to control the rotation of a servo to a certain position,

where 3 is the servo port number, which corresponds to the number identified on the Robot HAT driver board, but pay attention to the rudder When the machine is connected to the drive board, do not insert the reverse direction of the ground wire, VCC and signal wire, brown to black, red to red, yellow to yellow; 0 is the deviation of controlling the rotation of the servo Our program does not use this function to correct the deviation (the reason for the error of the steering gear can refer to **4.2 Structural Assembly Note**); 300 is the PWM duty cycle value you want to set. According to the different servos, this value represents different servo angles.The PWM duty cycle range of the servos we use is approximately 100 to 560, which corresponds to a rotation range of approximately 0 ° to 180 °.

⃗The above code to control the steering gear does not control the rotation speed of the steering gear. If

we want to make a certain steering gear swing back and forth slowly between two positions, we need to use an increasing or decreasing variable method to control the steering gear.

# Control the slow motion of the steering gear

import Adafruit\_PCA9685 # Import the library used to communicate with PCA9685 import time

pwm = Adafruit\_PCA9685.PCA9685()# Instantiate the object used to control the PWM pwm.set\_pwm\_freq(50) # Set the frequency of the PWM signal

while 1:

for i in range(0,100): # Slowly move the servo from 300 to 400 pwm.set\_pwm(3, 0, (300+i))

time.sleep(0.05)

for i in range(0,100): # Slowly move the servo from 400 to 300 pwm.set\_pwm(3, 0, (400-i))

time.sleep(0.05)

⃗The above code can make the steering gear rotate slowly back and forth between 300 and 400, but this method of controlling the steering gear also has a lot of drawbacks. When the program is executed until the

slow movement of the steering gear will block, this will seriously affect the program Performance, so we provide a multi-threaded solution in our robot product program to solve this problem.

# Non-blocking control

⃗You can find the RPIservo.py file in the server folder of the robot product, copy it to the same folder as the program you want to run, and then you can use this method in your program.

import RPIservo # Import a library that uses multiple threads to control the steering gear import time

sc = RPIservo.ServoCtrl()# Instantiate the object that controls the steering gear sc.start() # Start this thread, when the servo does not move, the thread is suspended

while 1:

sc.singleServo(3, -1, 2) time.sleep(1)