
SunFounder PiCar-V

www.sunfounder.com

Feb 04, 2024

CONTENTS

1	About the PiCar-V	1
1.1	Components List	2
1.2	Introduction	10
1.3	Building the Car	11
1.4	Circuits Building	23
1.5	Get Started with Raspberry Pi	27
1.6	Servo Configuration	57
1.7	Continue to Assemble	60
1.8	Getting on the Road!	66
1.9	Appendix	80
1.10	Thank You	91
2	Copyright Notice	93

ABOUT THE PICAR-V

The car PiCar-V is equipped with a wide-angle USB webcam, three whole new circuit boards with less but more simple wiring, some acrylic plates with an adjusted structure, and new code suitable for almost all platforms to control the car.

In this book, we will show you how to build the PiCar-V via description, illustrations of physical components, in both hardware and software respects. You will enjoy learning how all this work. You can view the latest PDF user manual or clone the code by click the link: https://github.com/sunfounder/SunFounder_PiCar-V/tree/V3.0.

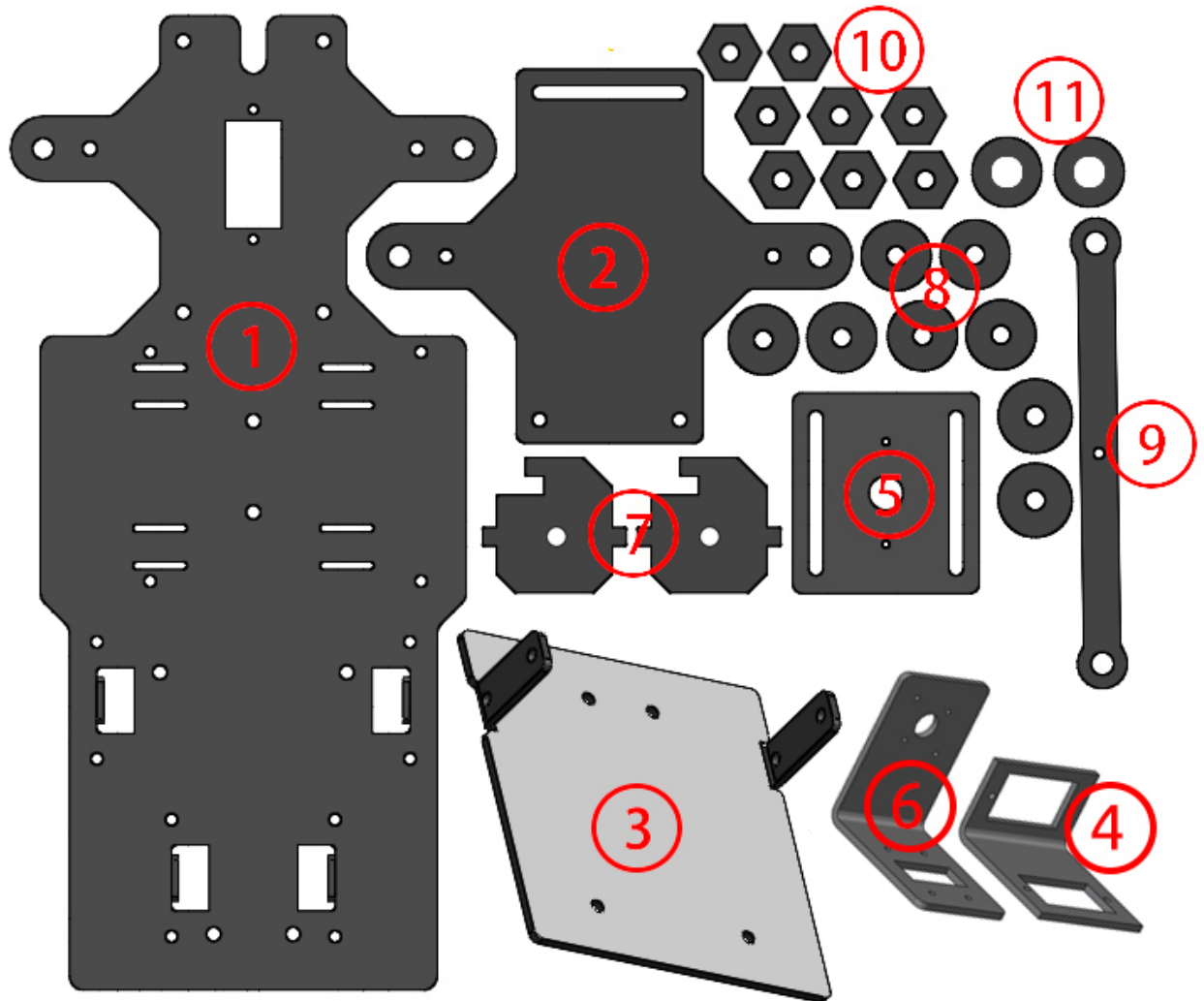
You are welcomed to post requests and issues on our page on Github.

If you want to learn another projects which we don't have, please feel free to send Email and we will update to our online tutorials as soon as possible, any suggestions are welcomed.

Here is the Email: cs@sunfounder.com.

1.1 Components List

1.1.1 Plates



1. Upper Plate x 1
2. Front Half Plate x 1
3. Back Half Plate x 1
4. Pan-and-tilt Plate x 1
5. Pan-and-tilt Base Plate x 1
6. Camera Mount Plate x 1
7. Steering Connector Plate x 2
8. Bearing Shield x 8
9. Steering Linkage Plate x 1
10. Hex Front Wheel Fixing Plate x 8

11. Gasket Plate x 2

Note: The Bearing Shield is similar to the Gasket Plate, but the aperture of Gasket Plate is large than the Bearing Shield.




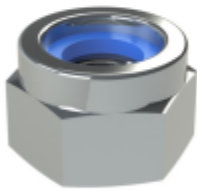


1.1.2 Servo x 3







1. Servo
2. 1-arm Rocker Arm
3. 2-arm Rocker Arm
4. 4-arm Rocker Arm
5. Rocker Arm Fixing Screw
6. Rocker Arm Screw

1.1.3 Mechanical Fasteners

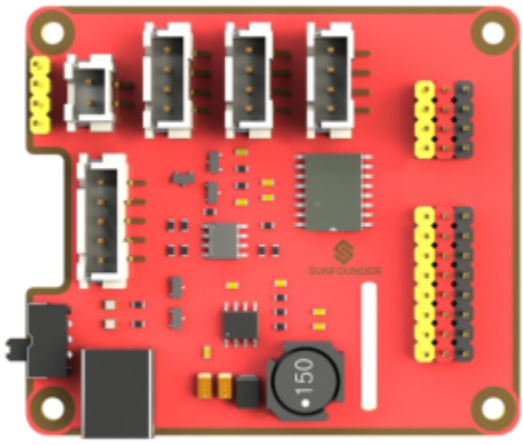
Name	Component	Qty.
M1.5x4 Self-tapping Screw		10
M2x8 Screw		6
M2.5x6 Screw		4
M2.5x12 Screw		8
M3x8 Screw		8
M3x8 Countersunk Screw		2
M3x10 Screw		7
M3x25 Screw		4
M4x25 Screw		2

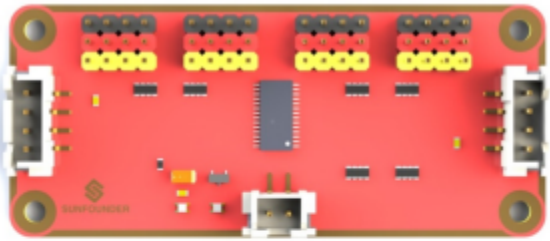
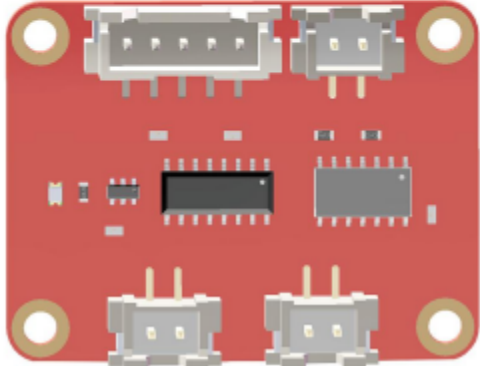
M2 Nut		6
M2.5 Nut		12
M3 Nut		21
M4 Self-locking Nut		2
M2.5x8 Copper Standoff		8
M3x25 Copper Standoff		8

1.1.4 Wires


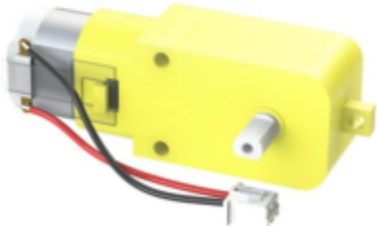
100mm HX2.54 5-Pin Jumper Wire		1
50mm HX-2.54 4-Pin Jumper Wire		1
50mm HX-2.54 2-Pin Jumper Wire		1
100mm HX-2.54 2-Pin Jumper Wire		1

1.1.5 PCB

Robot HATS		1
------------	--------------------------------------------------------------------------------------	---





PCA9685 PWM Driver		1
Motor Driver Module		1

1.1.6 Other Components

2x18650 Battery Holder		1
DC Gear Motor		2

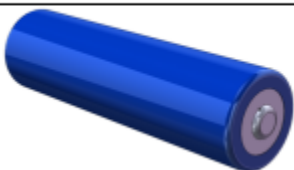
120° Wide-angle USB Camera		1
Rear Wheel		2
Front Wheel		2
Ribbon (30cm)		1

1.1.7 Tools

Cross Screwdriver		1
Cross Socket Wrench		1
M2.5/M4 Small Wrench		1
M2/M3 Small Wrench		1

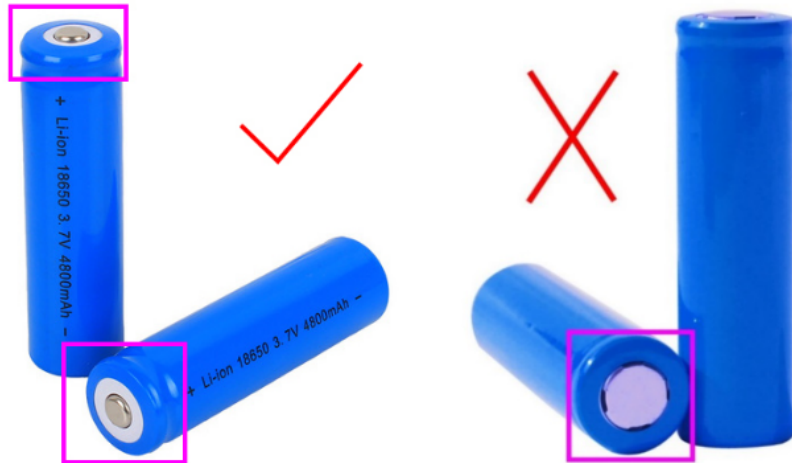
1.1.8 Self-provided Components

The following components are not included in this kit.

Component	Name	Qty.
	18650 3.7V Rechargeable Li-ion Battery	2

Note:

1. You are recommend to use 18650 batteries **without** a protective board. Otherwise, the car may be cut power and stop running because of the overcurrent protection of the protective board.
2. For unprotected batteries, please purchase those with the anode bulged out (as shown below), so that it can ensure the well connection with the battery holder.



-
3. In order to keep the car working for a long time, use large-capacity batteries as much as possible
-

1.2 Introduction

The PiCar-V is developed based on the previous Smart Video Car Kit. It's equipped with the new PCA9685 PWM driver, smaller and better-performing DC motor driver module, a Robot Hat with power and interface integrated, a wider-angle camera with clearer vision, some better wheels and tires, and so on. Later we'll check more details.

The basic functions of this new version still stay the same, including controlling the PiCar-V remotely and streaming the video data it captures back. Then what's new? Let's unveil the mysteries!



1.3 Building the Car

Are you excited when you open the box and see so many components? Keep your patience and take it easy. Please note that some details in the following steps need careful observation. You should double-check your work based on the figures in the manual after finishing each step. let's start!

1.3.1 Front Wheels

Insert an **M4x25 screw** through a **Steering Connector Plate**, 3 **Bearing Shields**, 3 **Hex Front Wheel Fixing Plates**, and a **front wheel**, into an **M4 Self-locking Nut** as shown below:

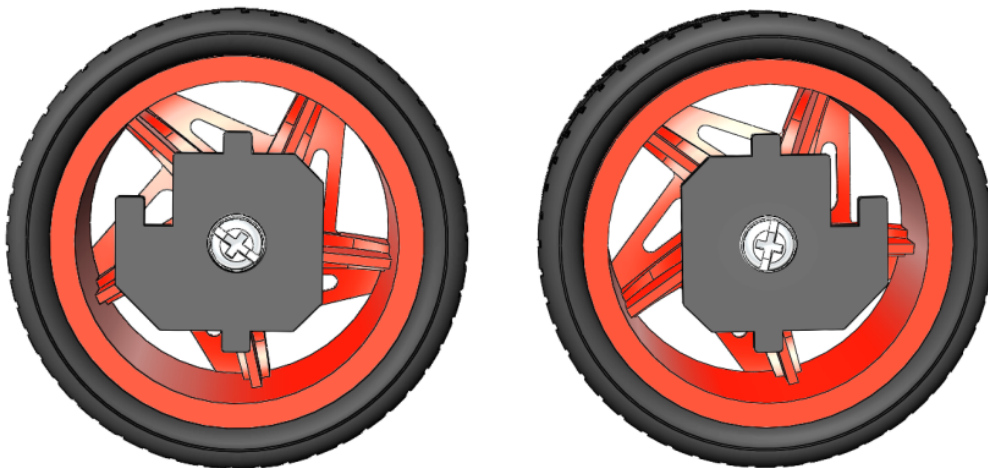


You can use the Cross Socket Wrench to secure the **M4 Self-locking Nut**, then use the screwdriver to tighten the **M4x25 screw**.



Note: The Self-locking Nut should be screwed tight enough. It would be better to tighten the screw until the wheel and Steering Connector cannot move first, then loosen the screw a little, so that the Steering Plate can just move. Thus, the wheel can turn flexibly when the connection would not be too loose.

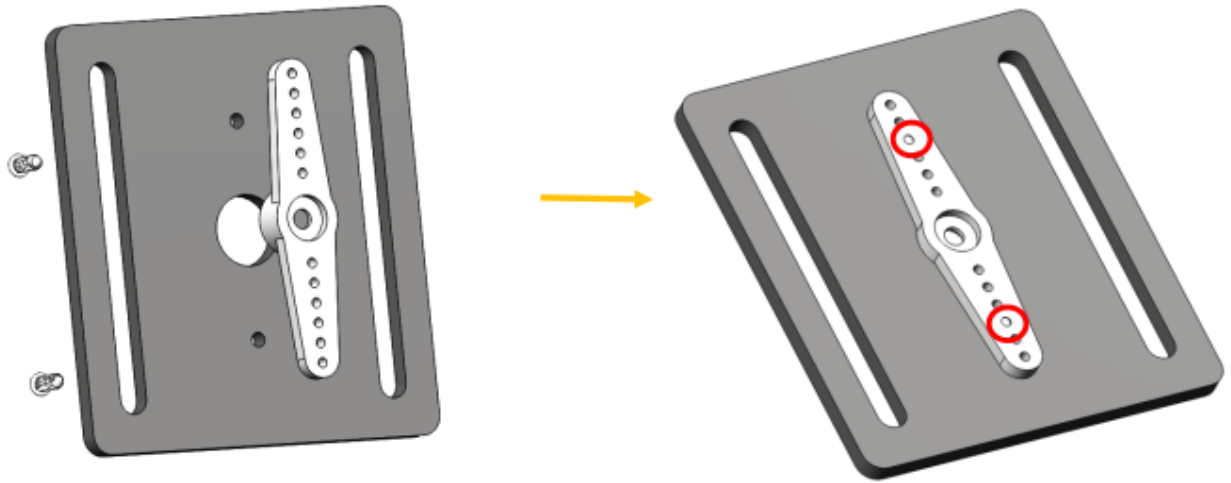
Assemble the other front wheel in the same way, but bear in mind the Steering Connector plate on the wheel should be symmetric with the previous one.



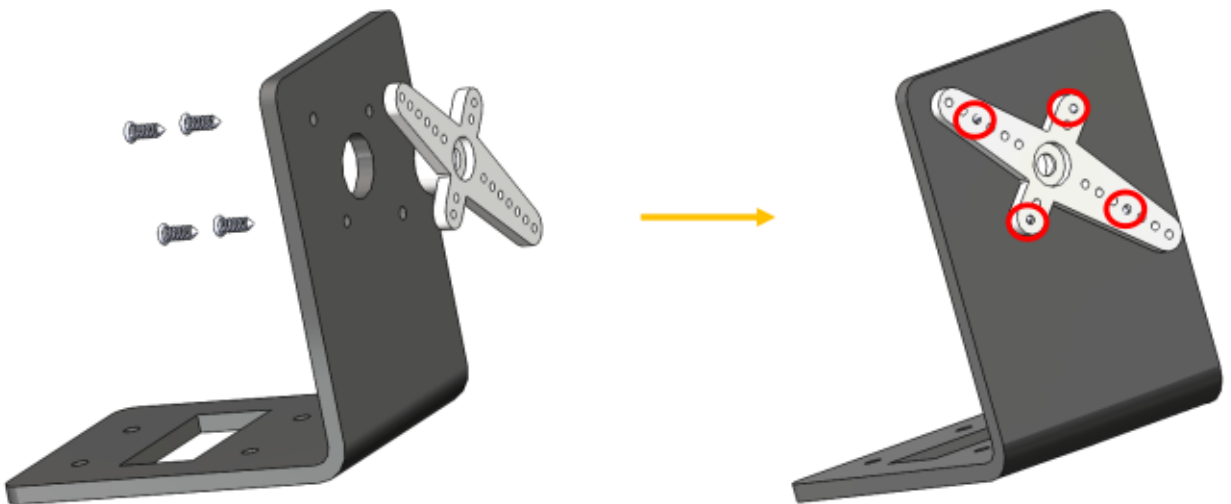
Now two front wheels have finished assembly.

1.3.2 Pan-and-Tilt

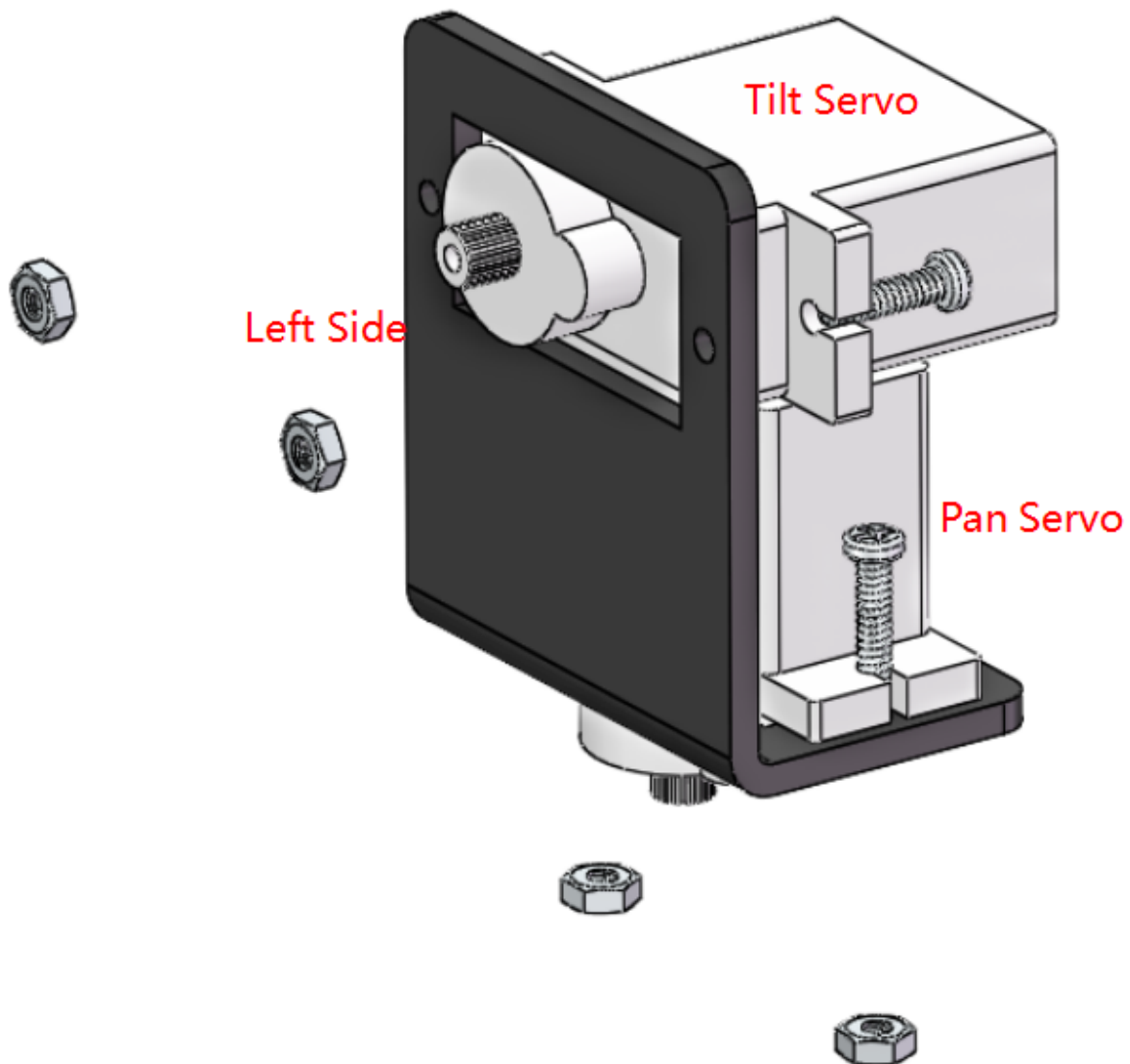
Take out the **Cross rocker arm** and mount it onto the **Pan-and-tilt Base Plate** with 2 **M1.5x4** screws (into the hole like this). Pay attention to the holes on the round rocker arm to be fastened.



Do the same operation to the **Camera Mount Plate**.



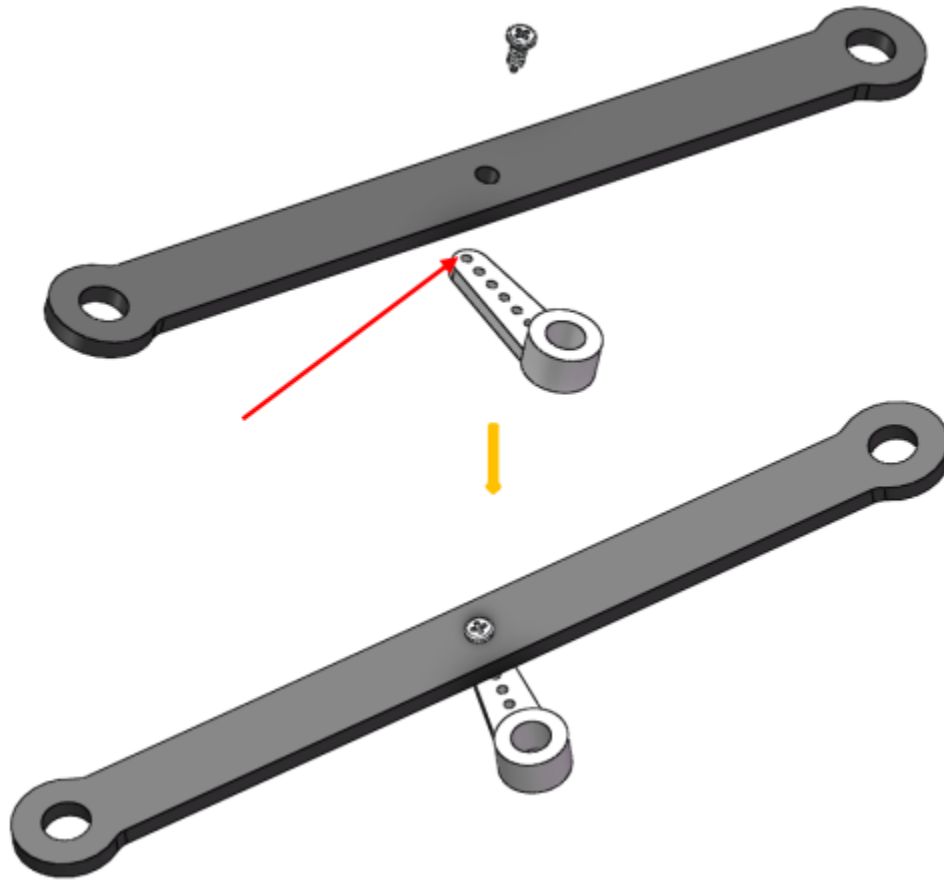
Assemble the two servos to the **Pan-and-tilt Plate** with four **M2x8** screws and the **M2 nuts** (The 2 servo shaft are close to the left side):



1.3.3 Steering Part

Connect the **Steering Linkage** and the **1-arm Rocker Arm** with the **M1.5x4 Self-tapping Screw**.

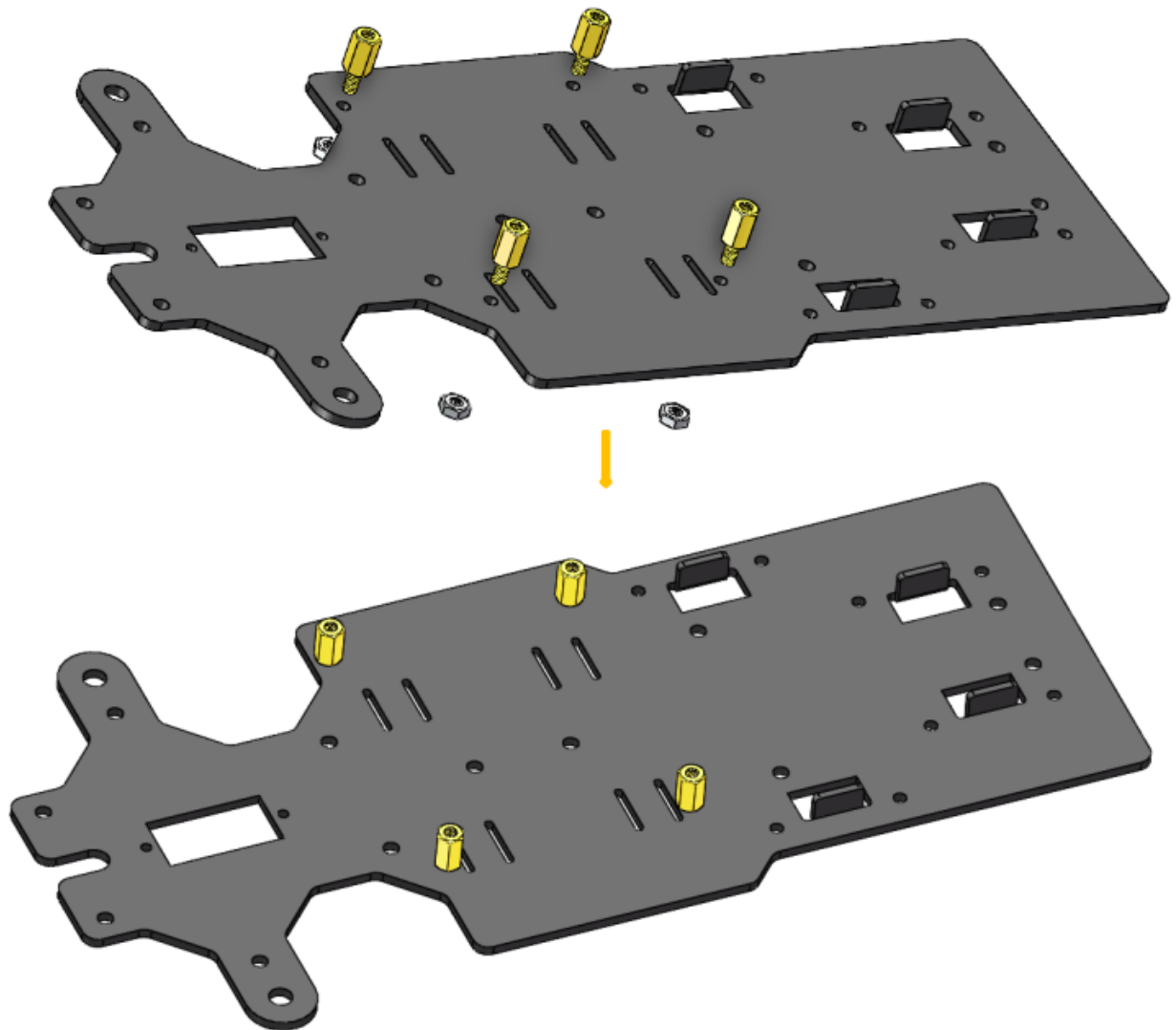
Note: Insert it into the **FIRST** hole of the arm (as indicated by the **arrow** below) which is the farthest from the gears.



And also fasten them as tightly as possible, and then loosen the screw a little so the Steering Linkage can move flexibly.

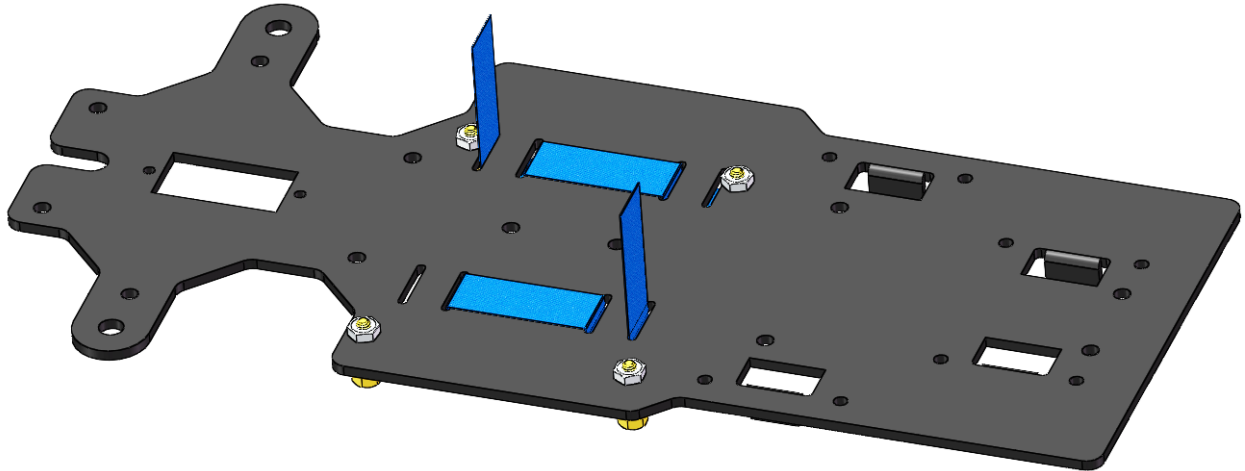
1.3.4 Upper Plate

Mount the **M2.5x8 copper standoffs** and **M2.5 nuts** into the **upper plate** first. Pay attention that the side the protruding prop should face up.

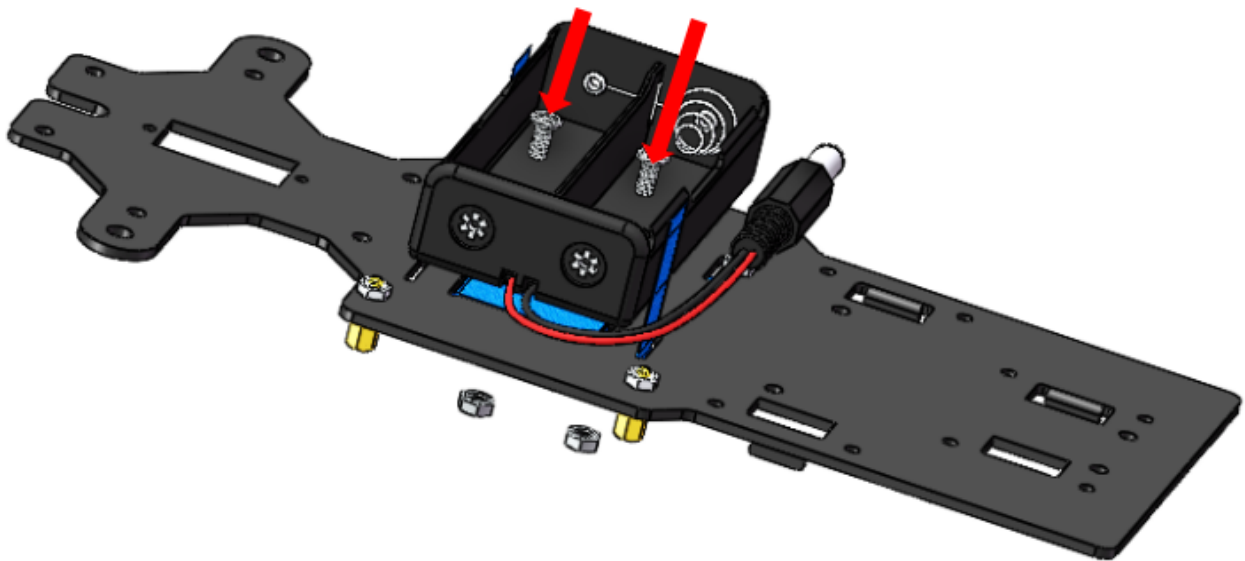


1.3.5 Battery Holder

Turn the Upper Plate upside down. Cut the **ribbon** into two halves. Thread them through the holes on the plate. Pay attention to the direction and leave one end longer out of the plate for each to remove the battery easily later.

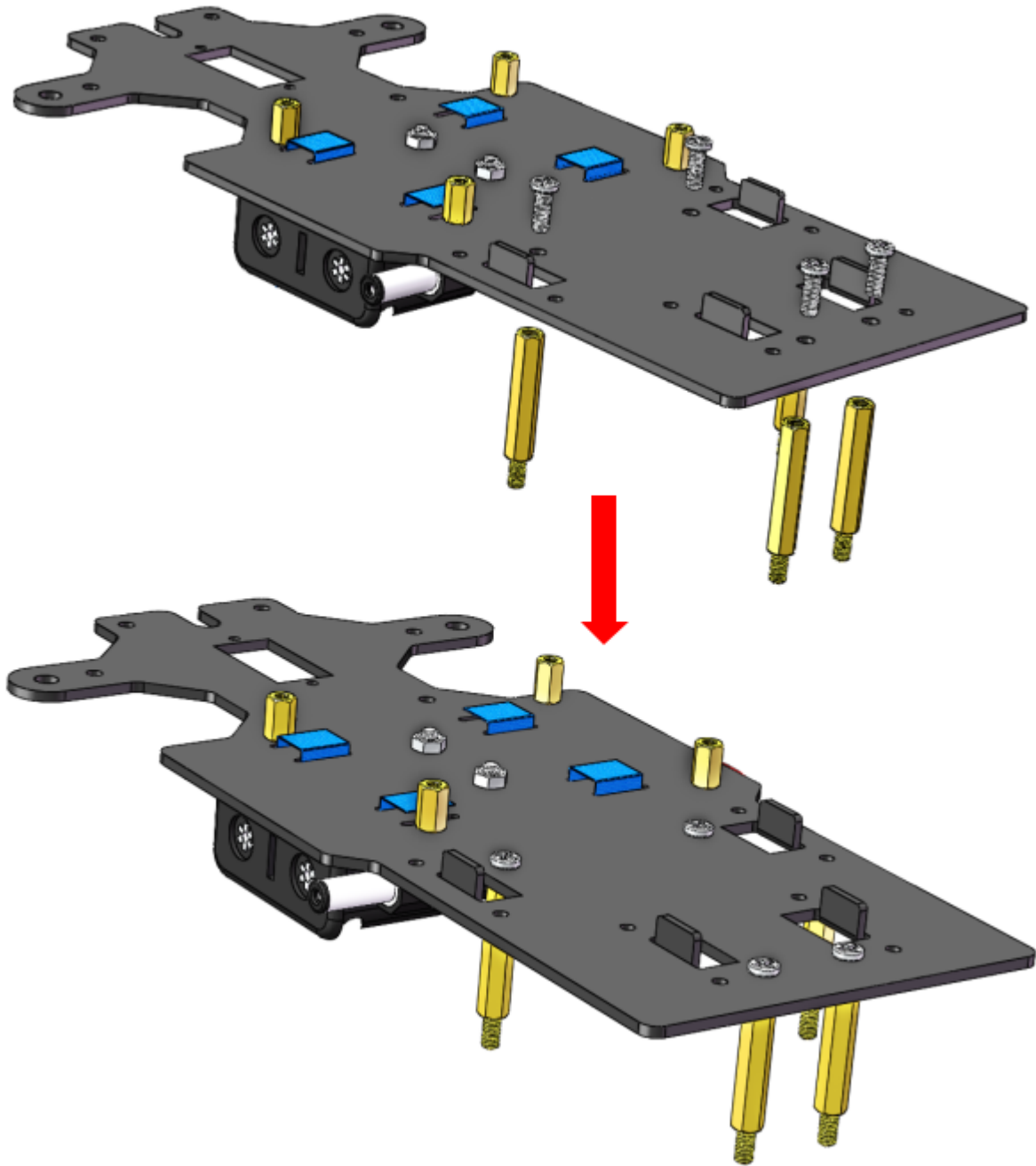


Fasten the battery holder with two **M3x8 countersunk screws** and **M3 nuts**. Pay attention to the direction of battery holder's wire.



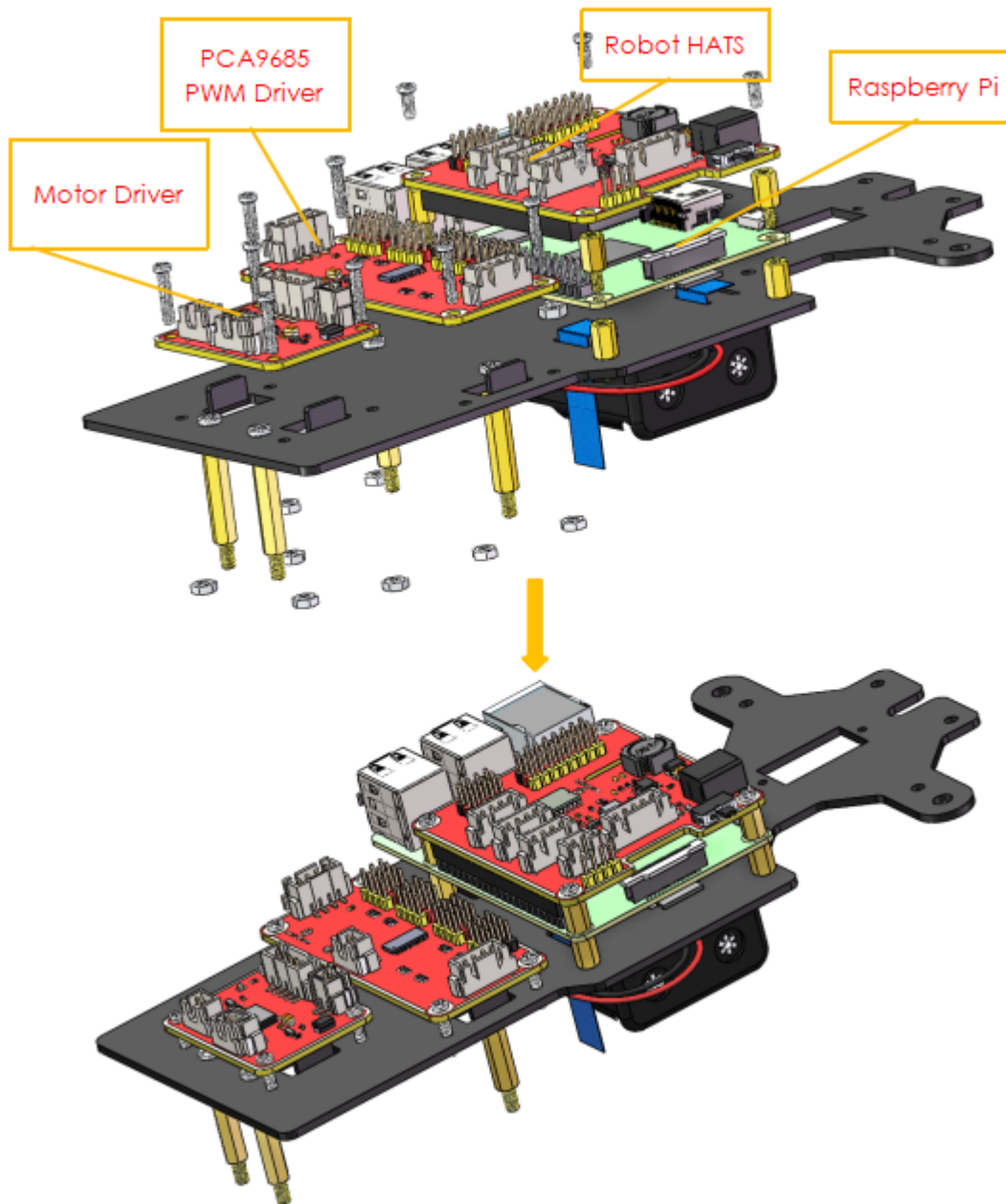
1.3.6 Rear Wheels (Screws)

Insert 4 **M3x8 screws** with four **M3x25 copper standoffs**:



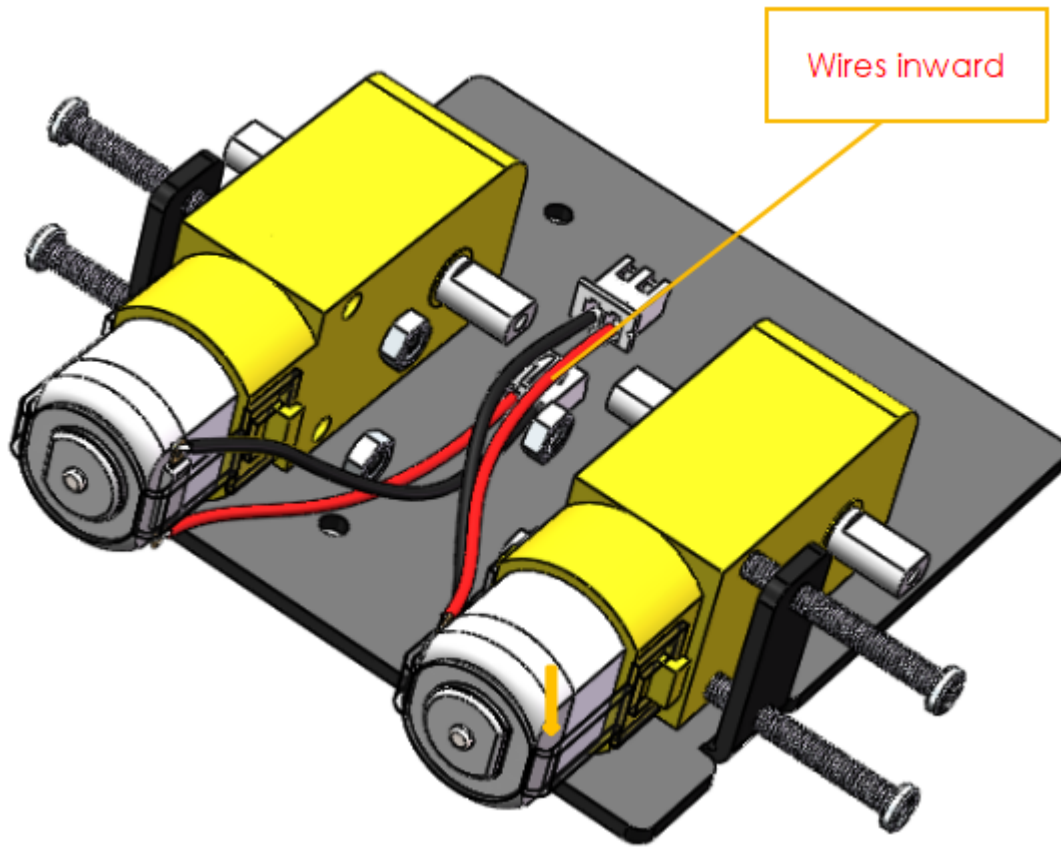
1.3.7 PCB Assembly

- 1) Assemble the **Raspberry Pi** (TF Card inserted) with 8 **M2.5x8 single pass copper standoffs**, then plug the **Robot HATS** onto it.
- 2) Fix the **Robot HATS** with 4 **M2.5x6 screws**.
- 3) Fix The **PCA9685 PWM Driver** and the **Motor Driver** with 8 **M2.5x12 screws and M2.5 nuts** into the **down plate**

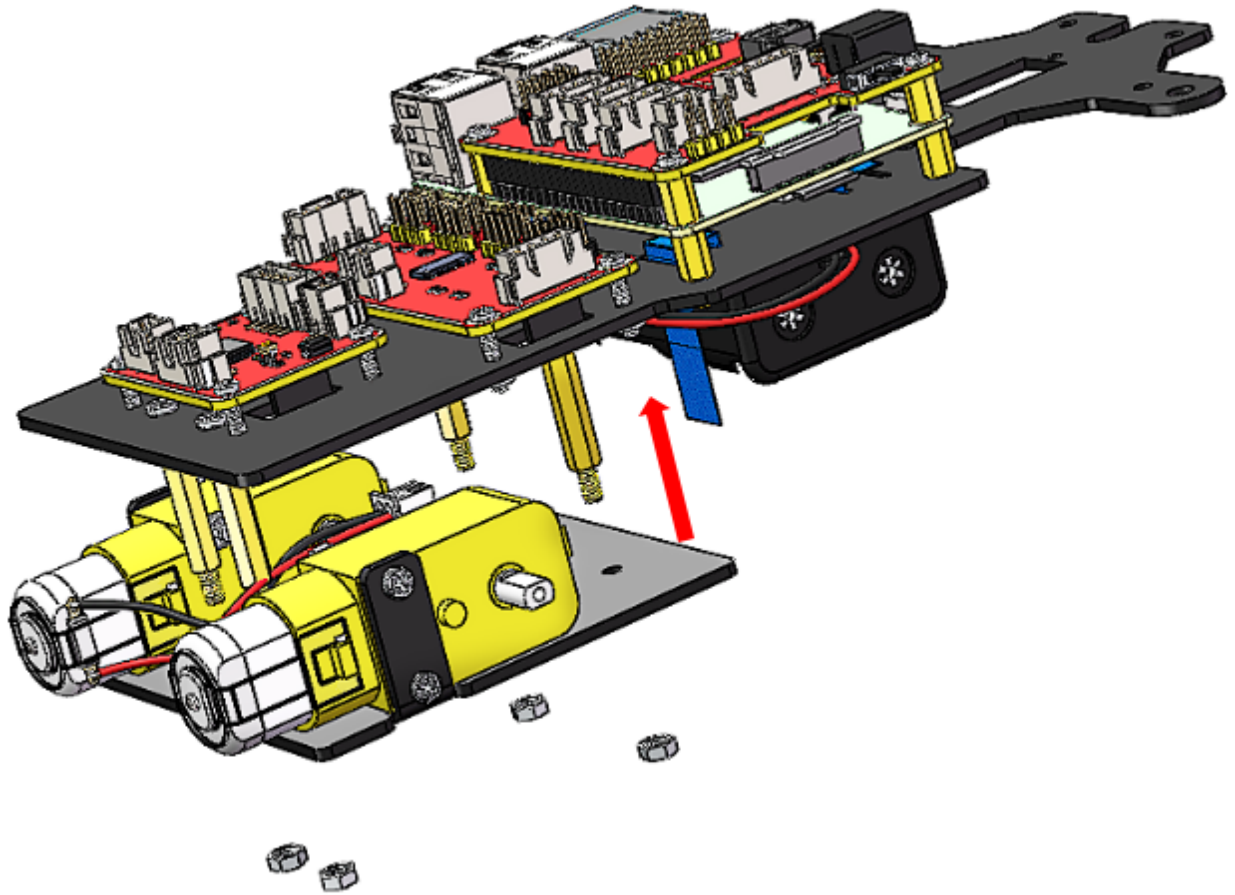


1.3.8 Fixing Rear Wheels

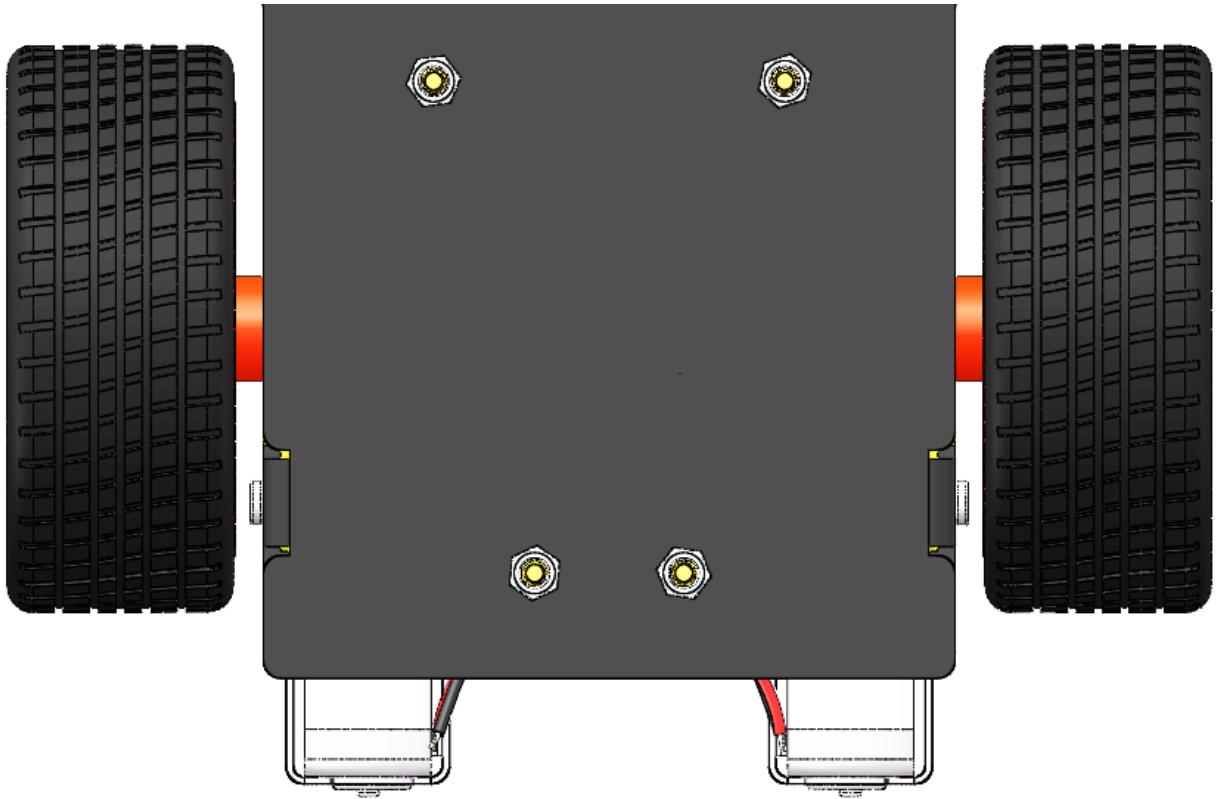
Assemble the two motors with four **M3x25 screws** and **M3 nuts**. Pay attention to place the motors with wires inward, providing convenience for connecting the circuit.



Assemble the rear wheels with 4 M3 nuts.

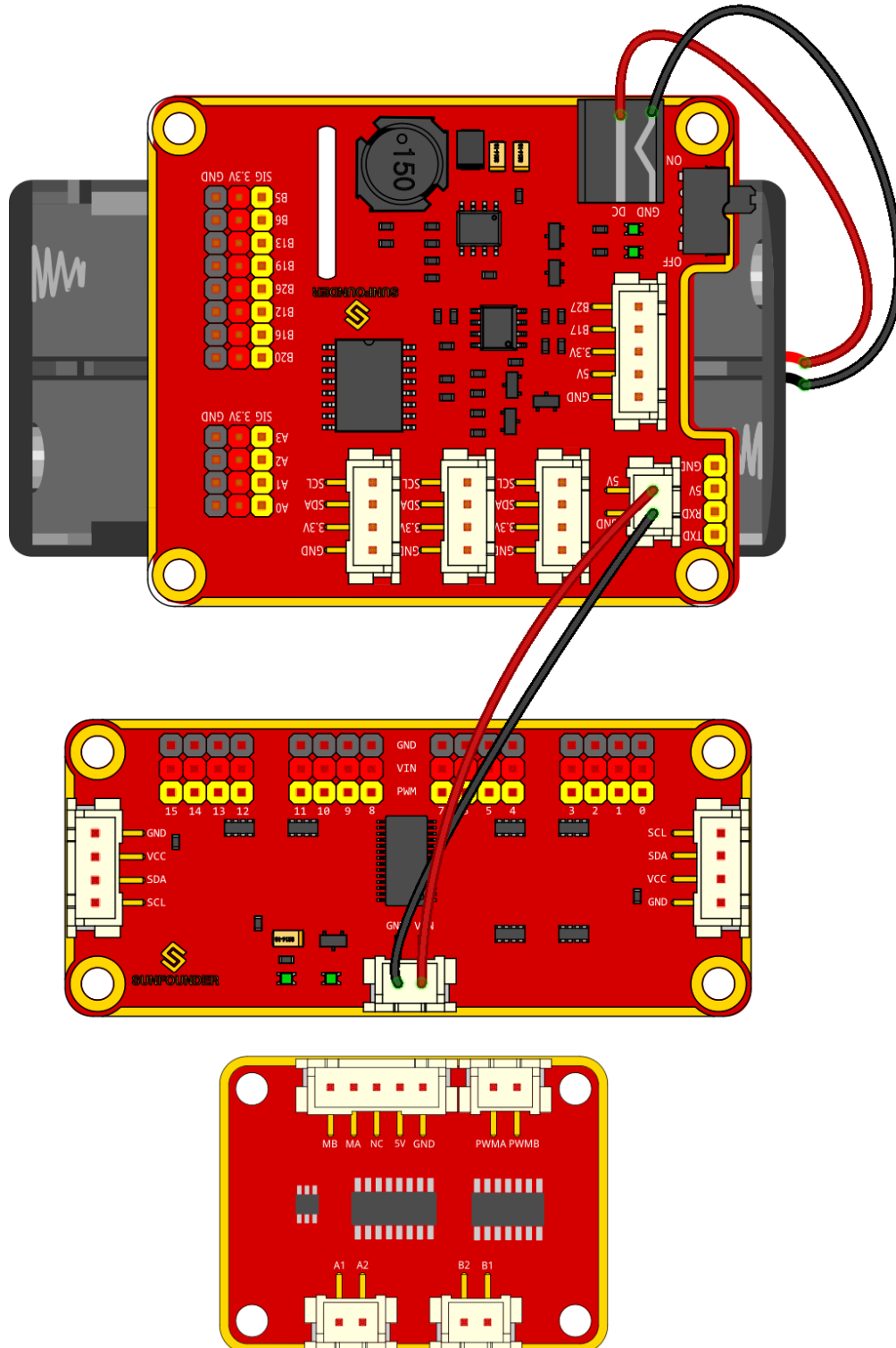


Align the **rear wheels** with the motor shaft, and rotate to insert them gently.

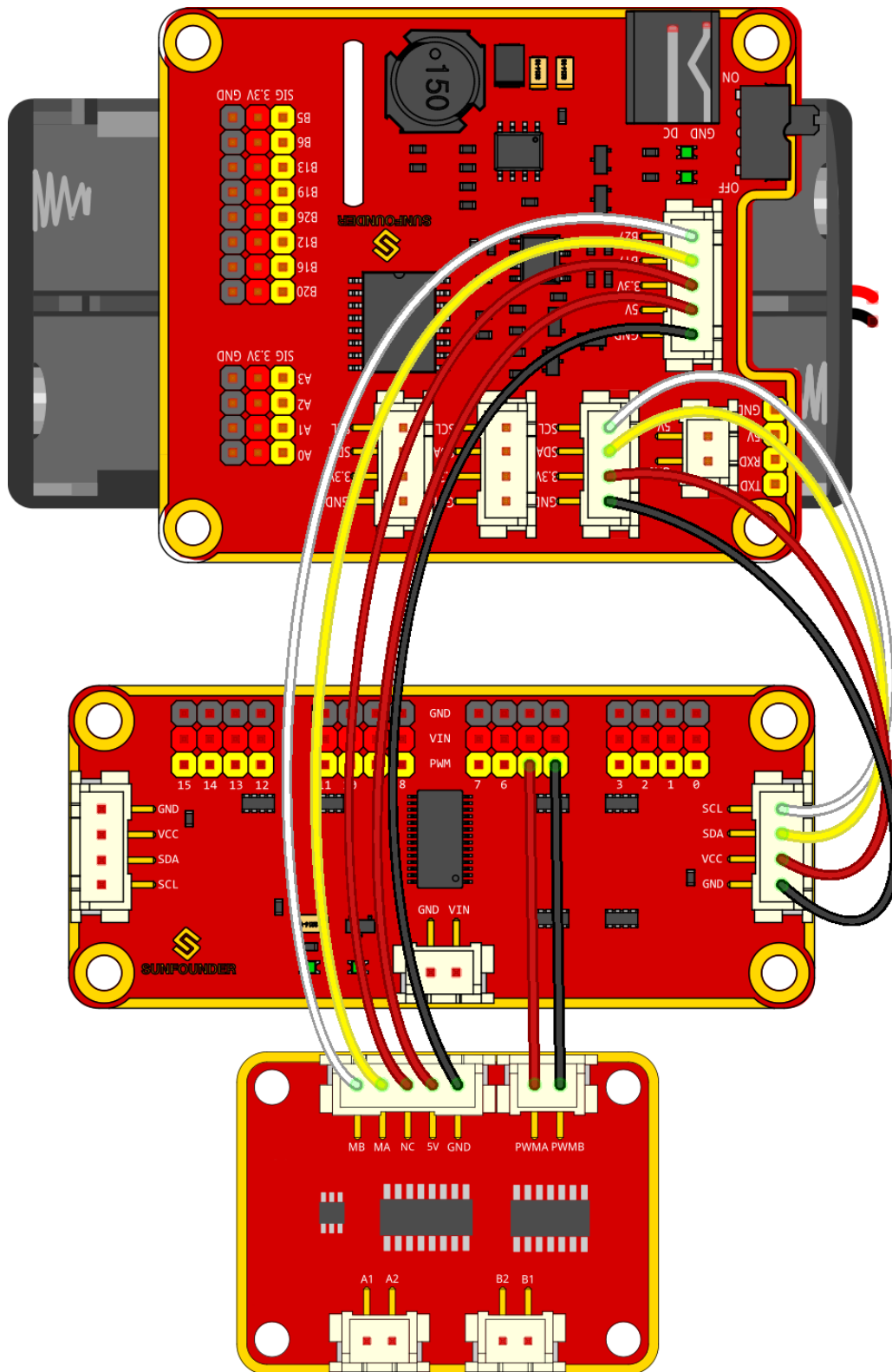


1.4 Circuits Building

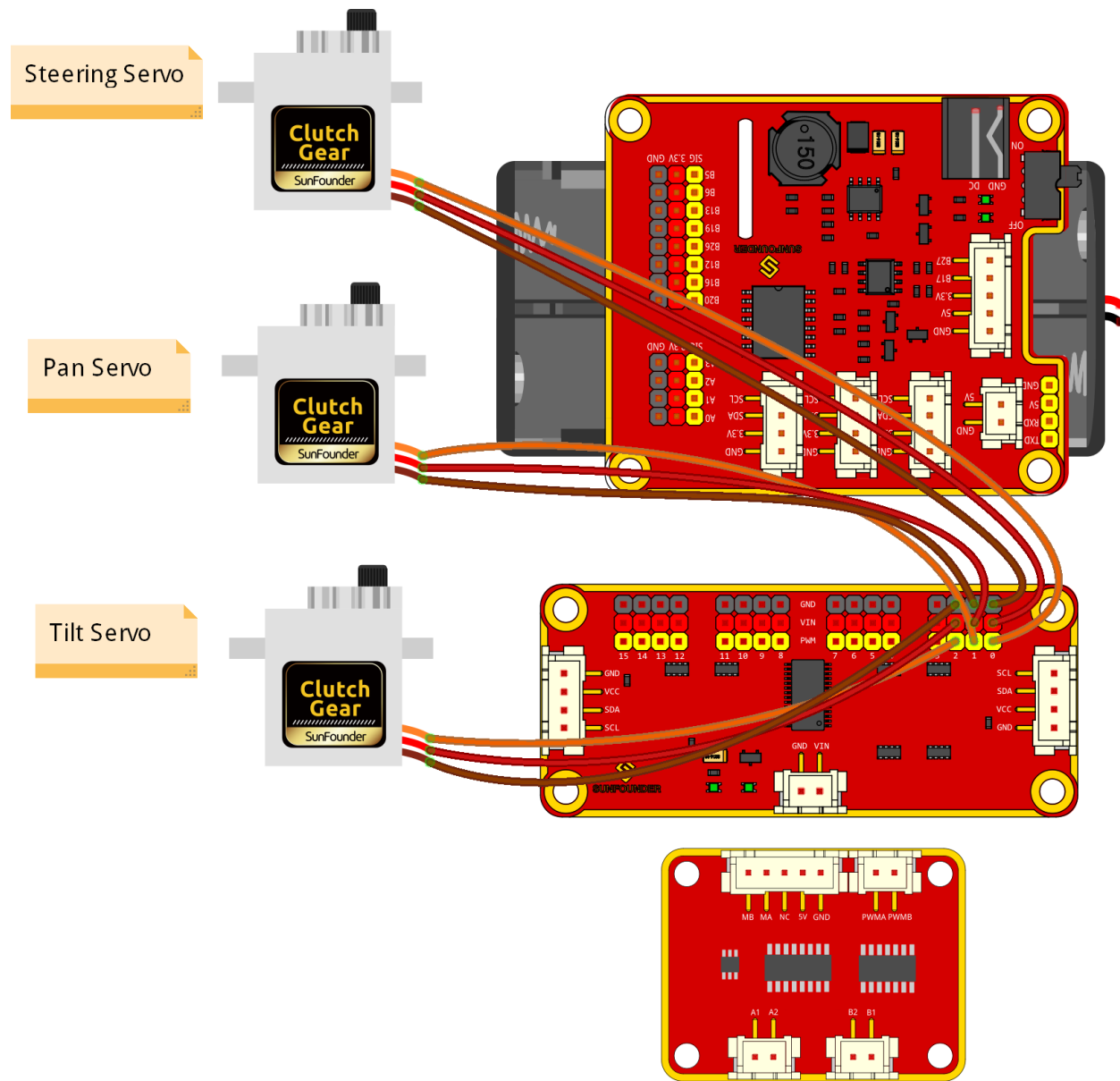
1.4.1 Connect the Power



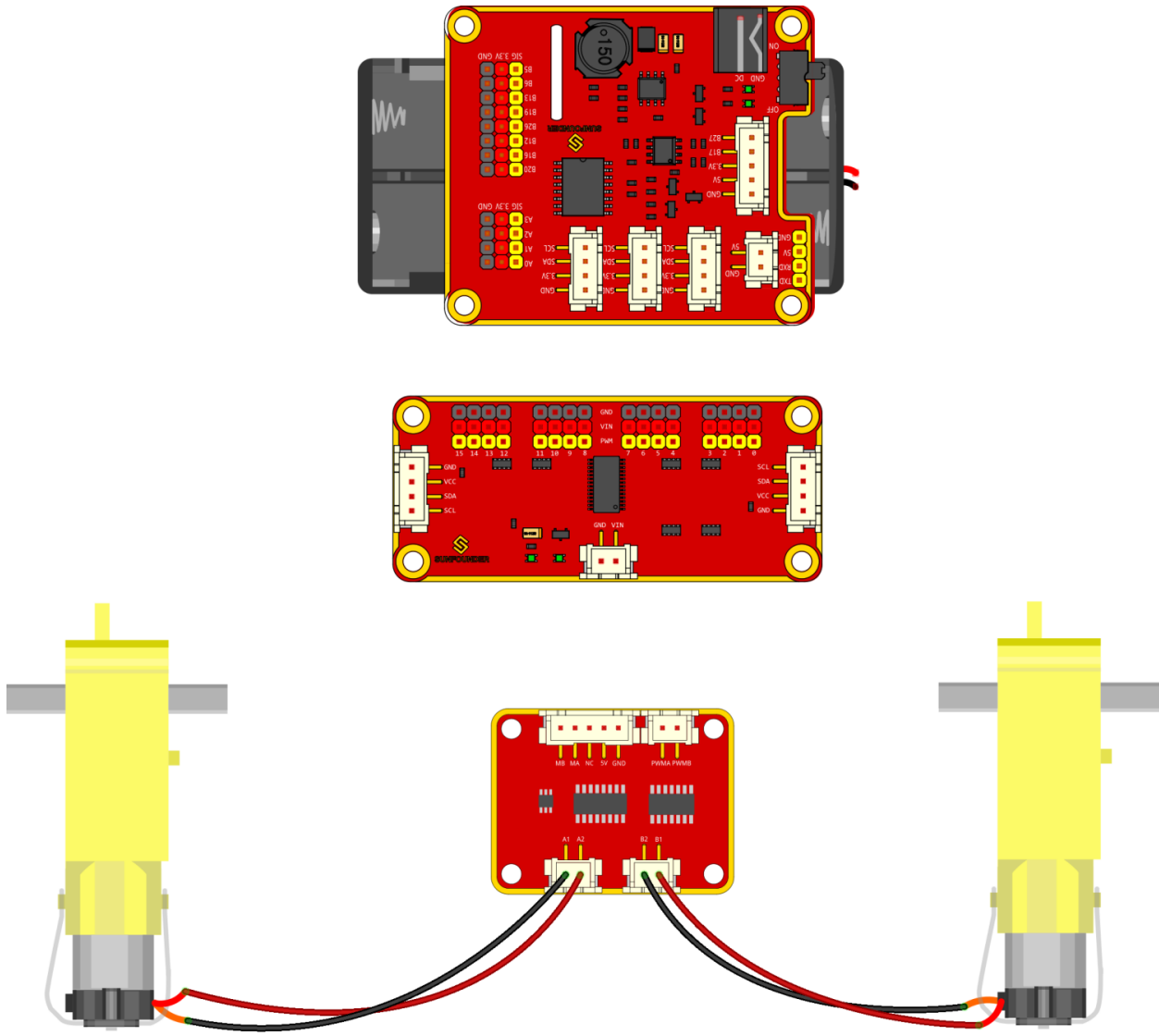
1.4.2 Connect the Modules



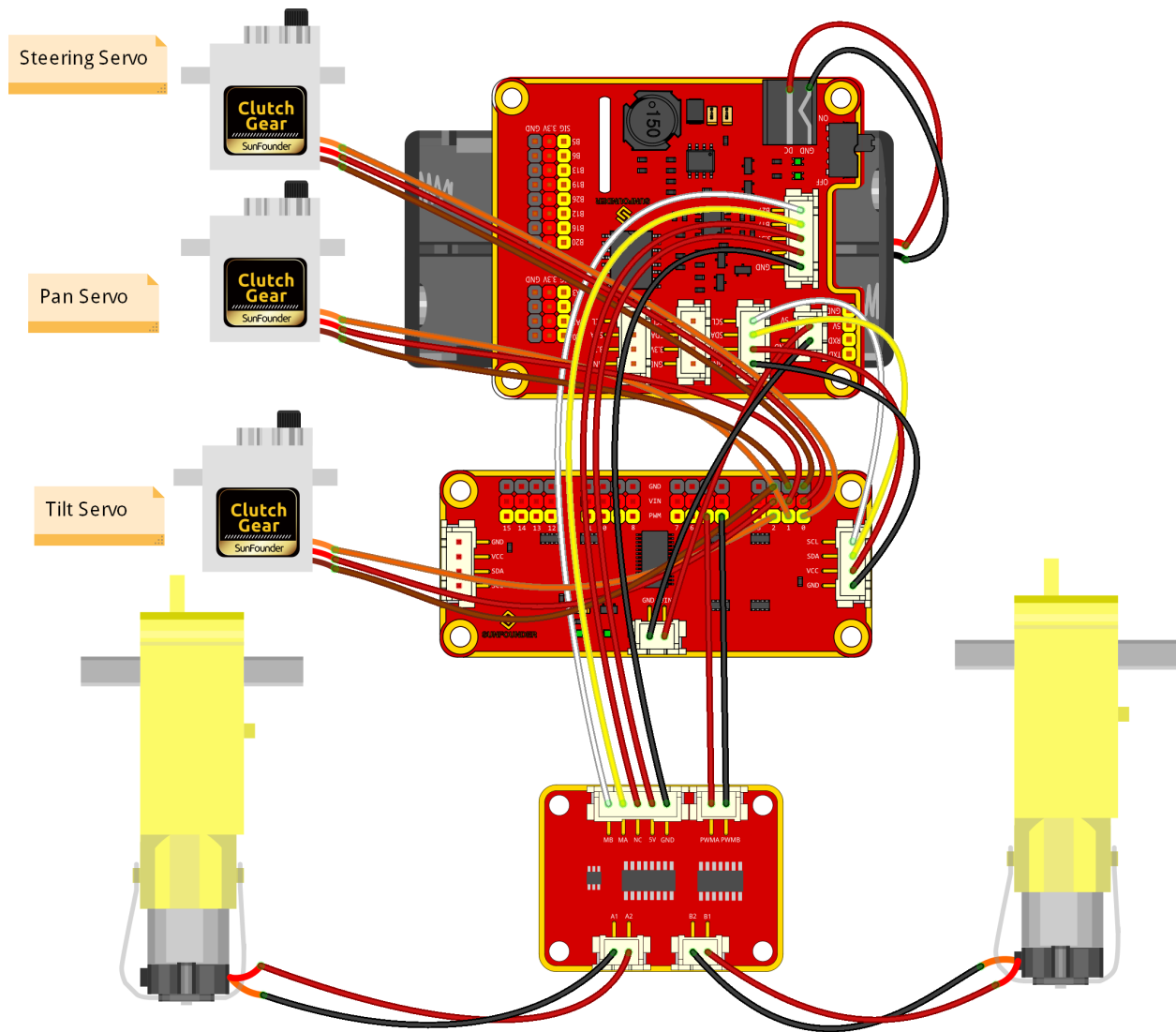
1.4.3 Connect the Servos



1.4.4 Connect the Motor



The complete connection is shown as follows.



1.5 Get Started with Raspberry Pi

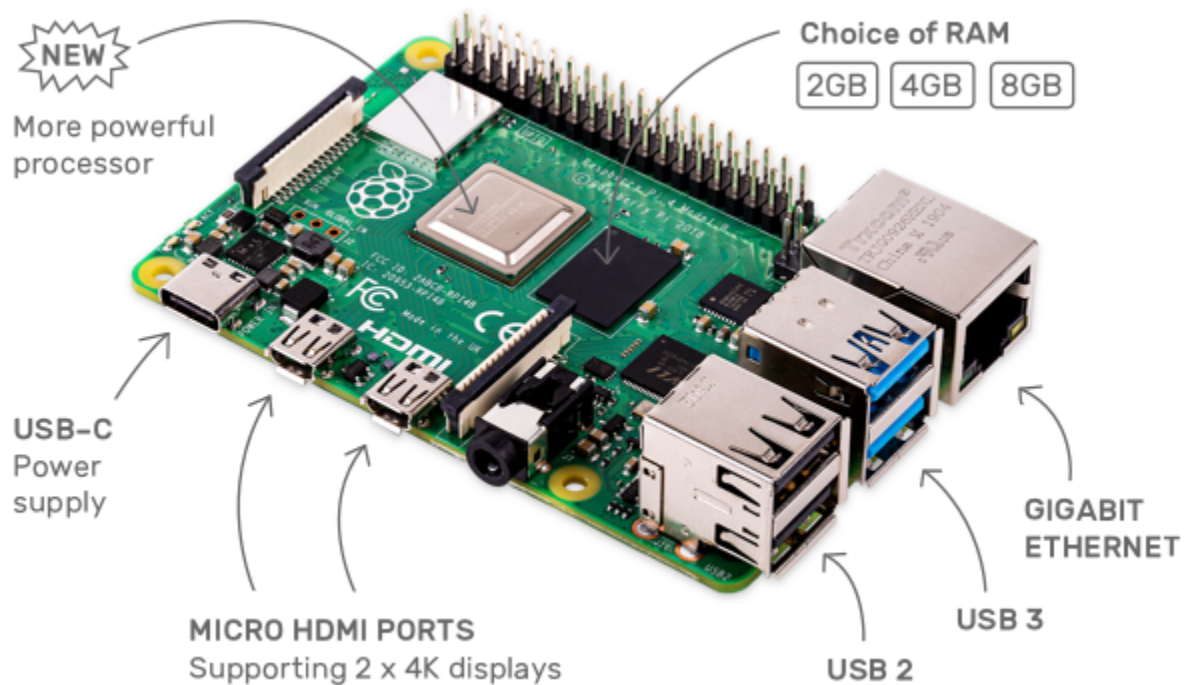
This section is to teach you how to install Raspberry Pi OS, configure wifi to Raspberry Pi, remote access to Raspberry Pi to run the corresponding code.

1.5.1 What Do We Need?

Required Components

Raspberry Pi

The Raspberry Pi is a low cost, credit-card sized computer that plugs into a computer monitor or TV, and uses a standard keyboard and mouse. It is a capable little device that enables people of all ages to explore computing, and to learn how to program in languages like Scratch and Python.



Power Adapter

To connect to a power socket, the Raspberry Pi has a micro USB port (the same found on many mobile phones). You will need a power supply which provides at least 2.5 amps.

Micro SD Card

Your Raspberry Pi needs an Micro SD card to store all its files and the Raspberry Pi OS. You will need a micro SD card with a capacity of at least 8 GB

Optional Components

Screen

To view the desktop environment of Raspberry Pi, you need to use the screen that can be a TV screen or a computer monitor. If the screen has built-in speakers, the Pi plays sounds via them.

Mouse & Keyboard

When you use a screen , a USB keyboard and a USB mouse are also needed.

HDMI

The Raspberry Pi has a HDMI output port that is compatible with the HDMI ports of most modern TV and computer monitors. If your screen has only DVI or VGA ports, you will need to use the appropriate conversion line.

Case

You can put the Raspberry Pi in a case; by this means, you can protect your device.

Sound or Earphone

The Raspberry Pi is equipped with an audio port about 3.5 mm that can be used when your screen has no built-in speakers or when there is no screen operation.

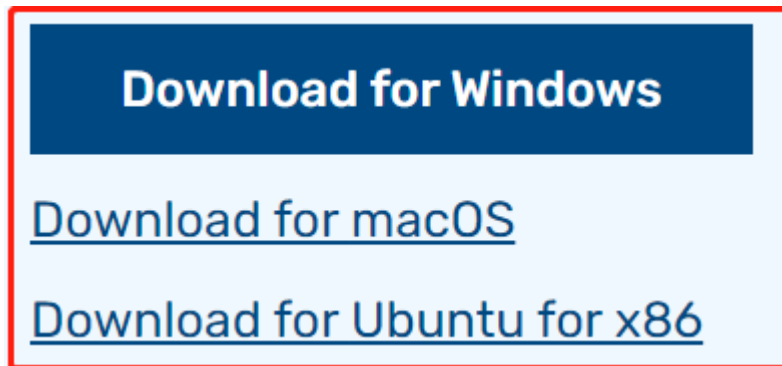
1.5.2 Installing the OS

Required Components

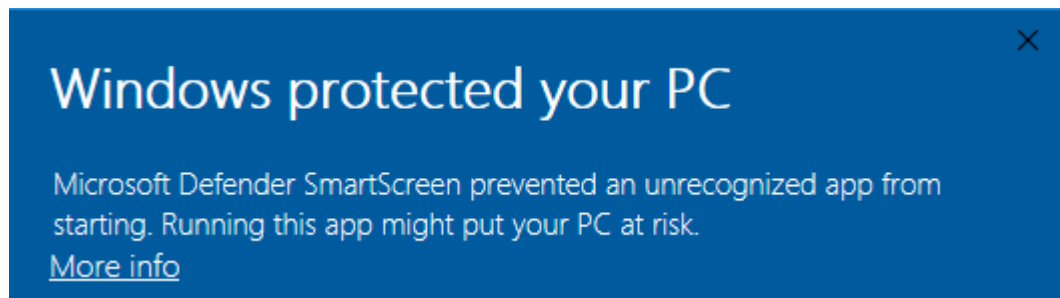
- Raspberry Pi 4B/Zero 2 w/3B 3B+/2B/Zero W
- 1 x Personal Computer
- 1 x Micro SD card

Steps

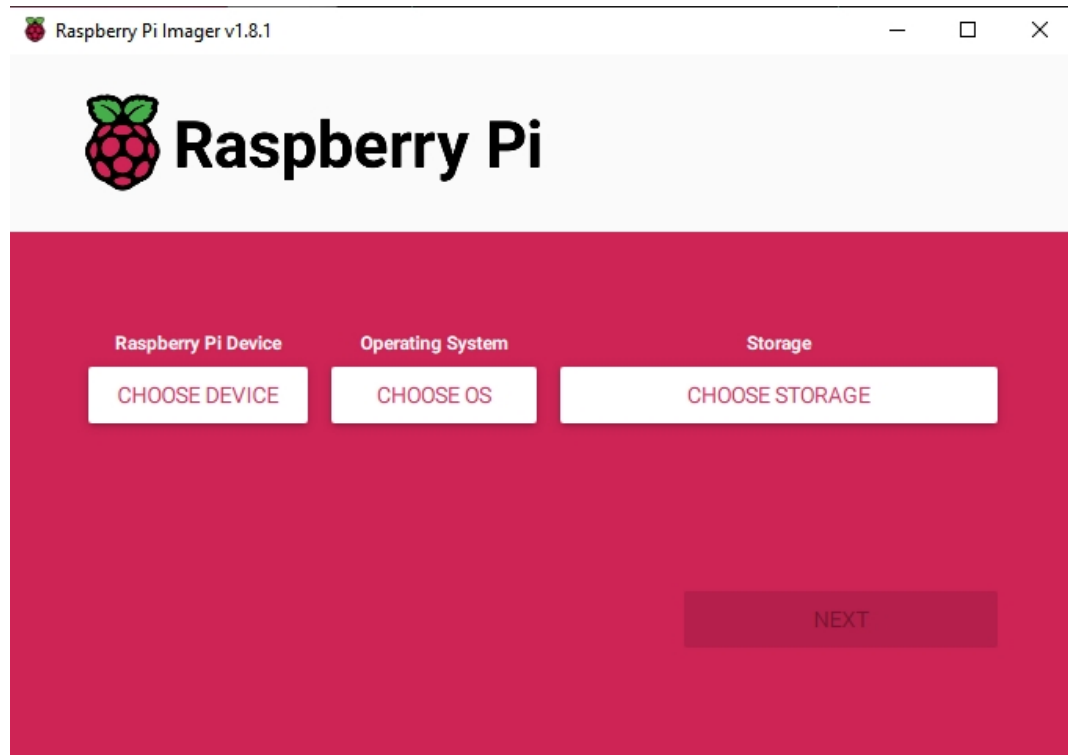
1. Go to the Raspberry Pi software download page: [Raspberry Pi Imager](#). Select the Imager version for your operating system. After downloading, open the file to start the installation.



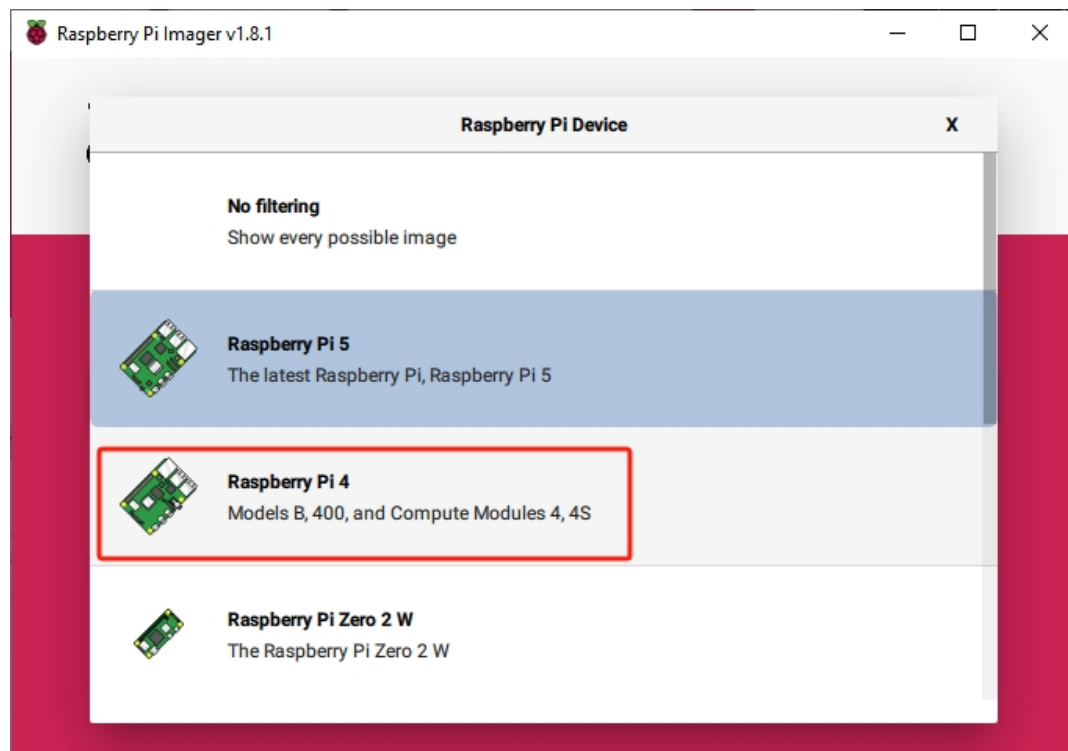
2. Upon launching the installer, your OS might display a security warning. For instance, Windows may show a caution message. If this occurs, select **More info** and then **Run anyway**. Follow the on-screen instructions to install the Raspberry Pi Imager.



3. Insert your SD card into the computer or laptop SD card slot.
4. Open the Raspberry Pi Imager application either by clicking its icon or executing `rpi-imager` in your terminal.



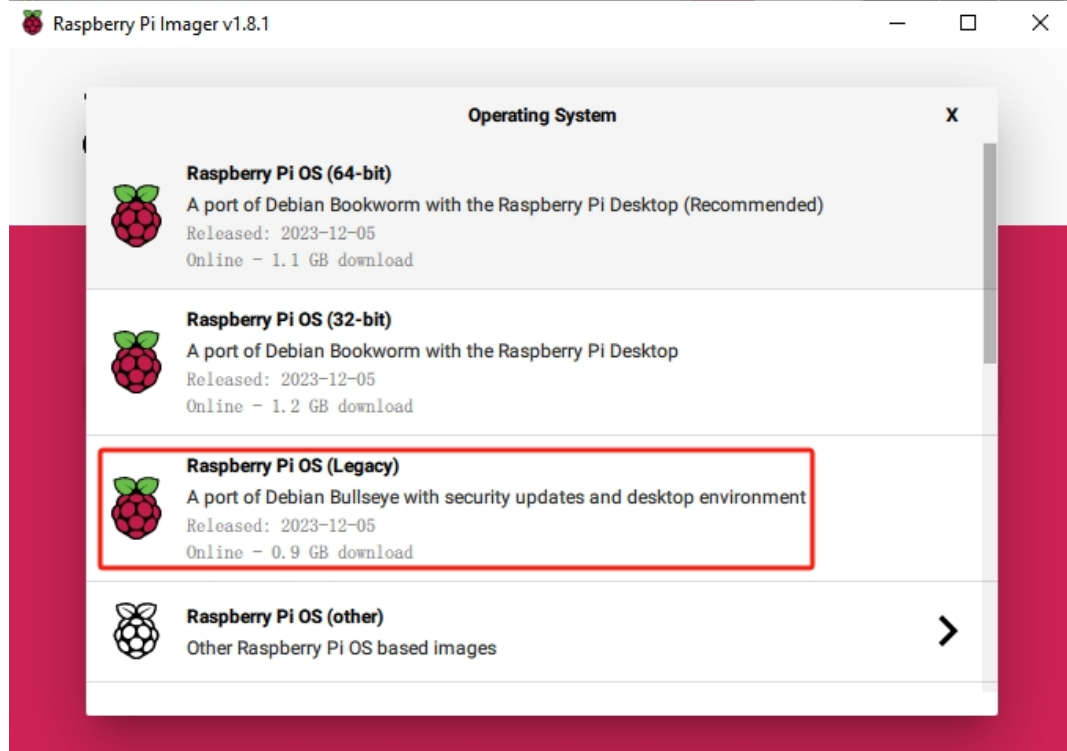
5. Click **CHOOSE DEVICE** and select your specific Raspberry Pi model from the list (Note: Raspberry Pi 5 is not applicable).



6. Select **CHOOSE OS** and then choose **Raspberry Pi OS (Legacy)**.

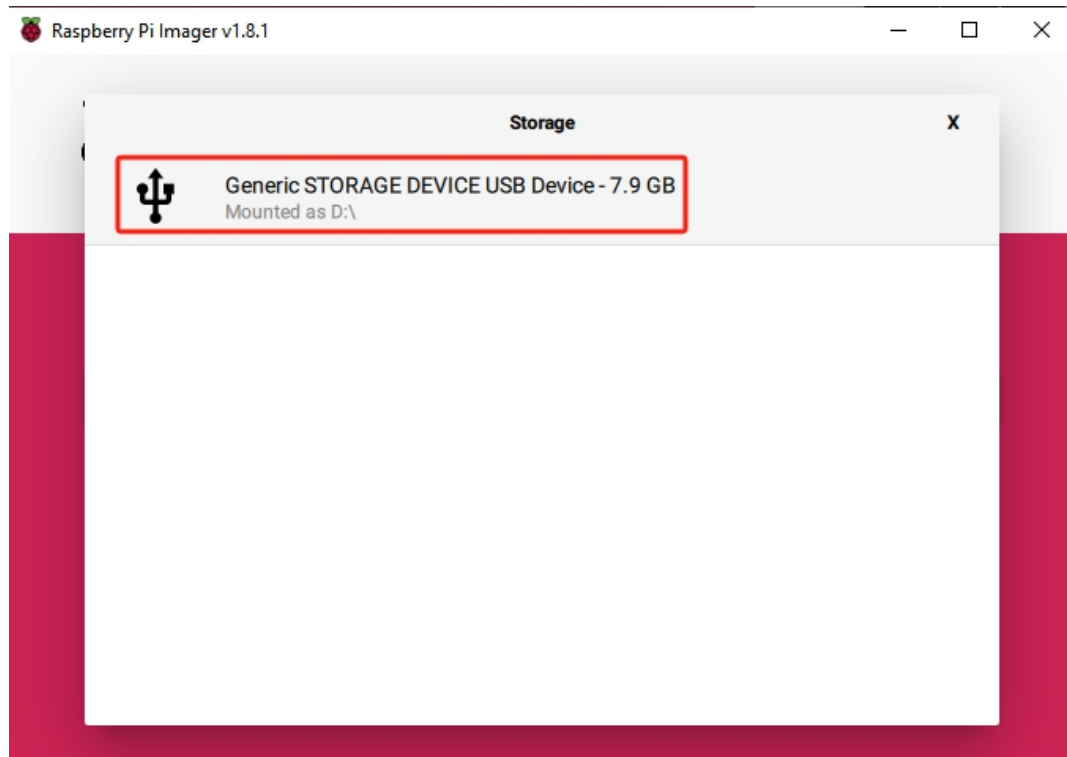
Warning:

- Please do not install the **Bookworm** version as the speaker will not work.
- You need to install the **Raspberry Pi OS (Legacy)** version - **Debian Bullseye**.

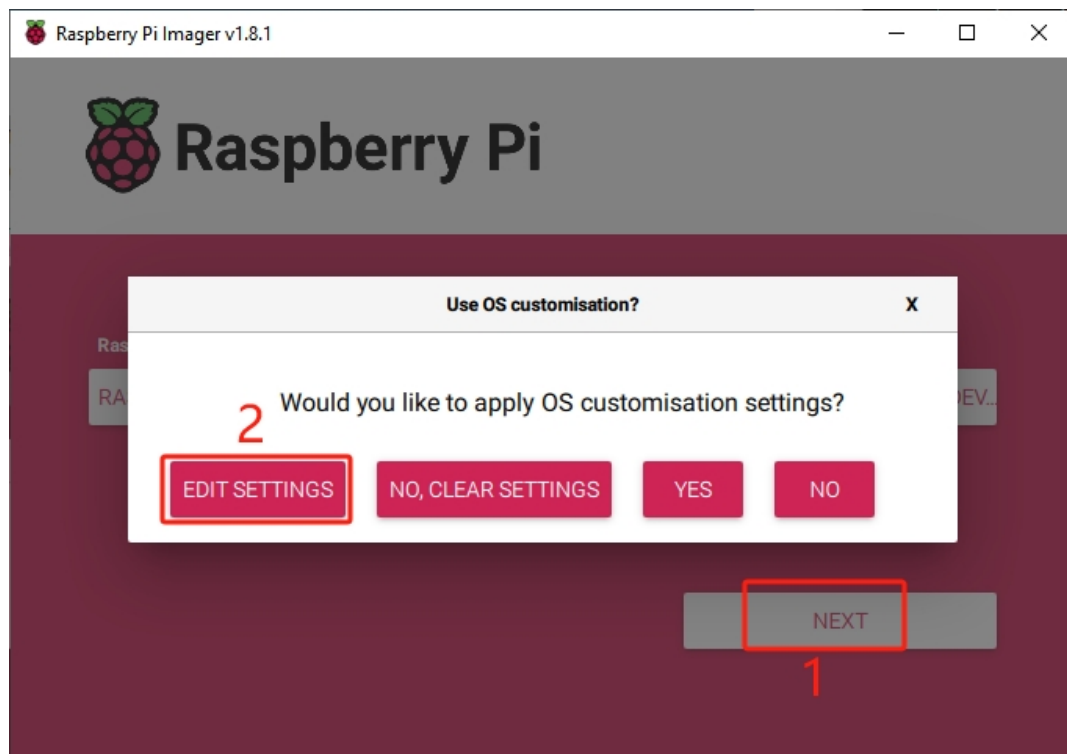


7. Click **Choose Storage** and pick the correct storage device for the installation.

Note: Be sure to select the correct device, especially if multiple storage devices are connected. Disconnect others if you're unsure.



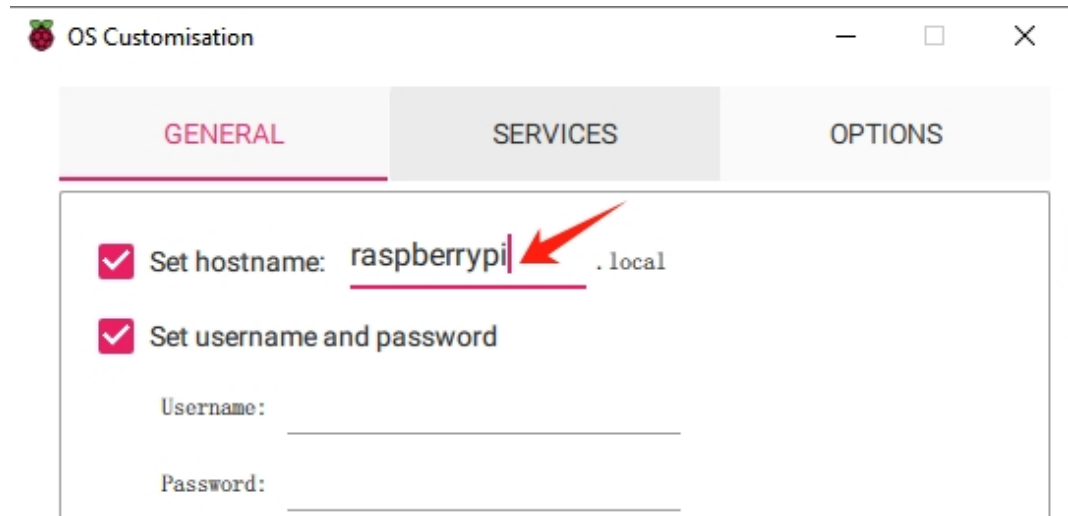
8. Press **NEXT** and select **EDIT SETTINGS** to customize your OS settings.



9. Set your Raspberry Pi's **hostname**.

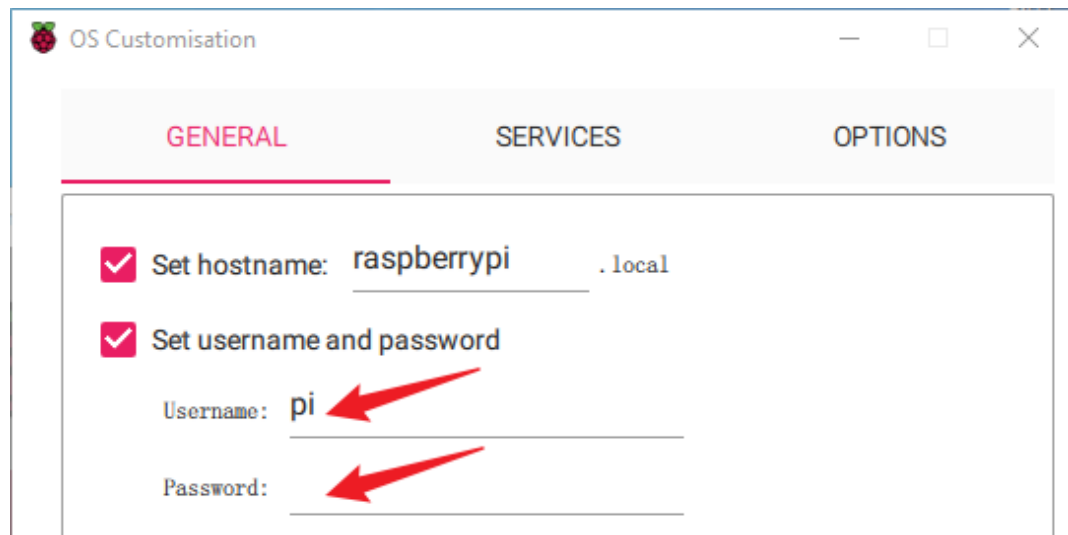
Note: The hostname is what your Raspberry Pi uses to identify itself on the network. You can

connect to your Pi using `<hostname>.local` or `<hostname>.lan`.



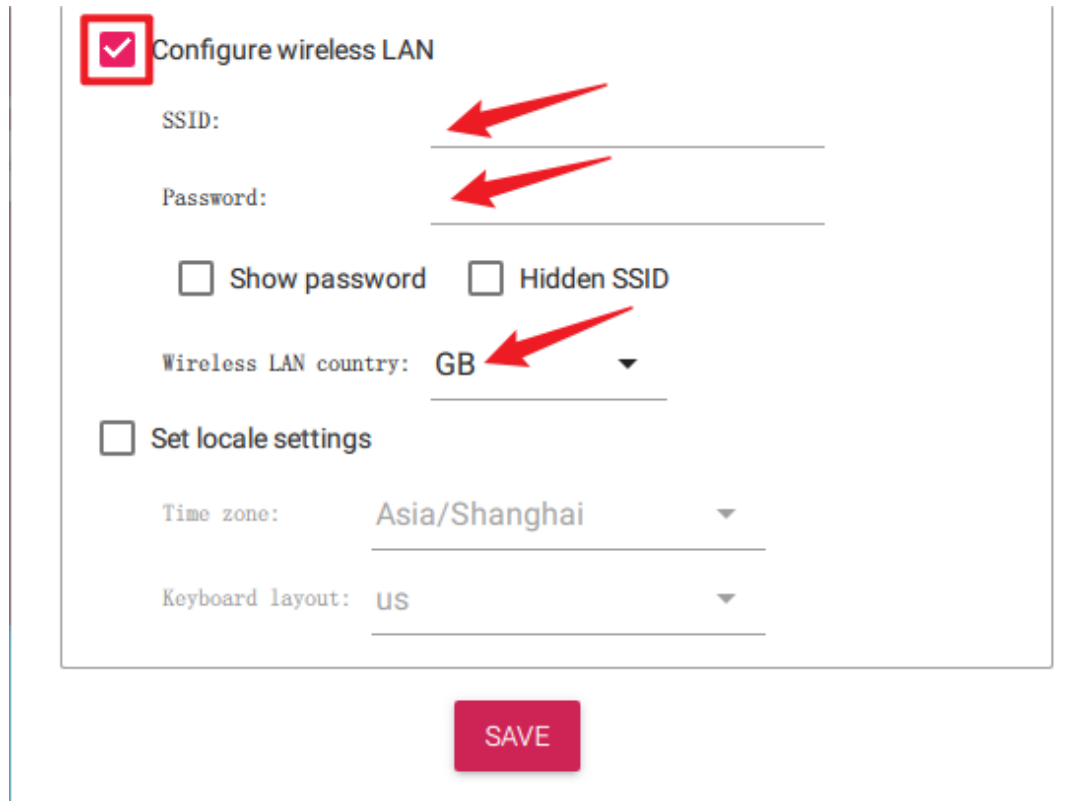
10. Create a **Username** and **Password** for the Raspberry Pi's administrator account.

Note: Setting a unique username and password is crucial for security, as the Raspberry Pi does not have a default password.



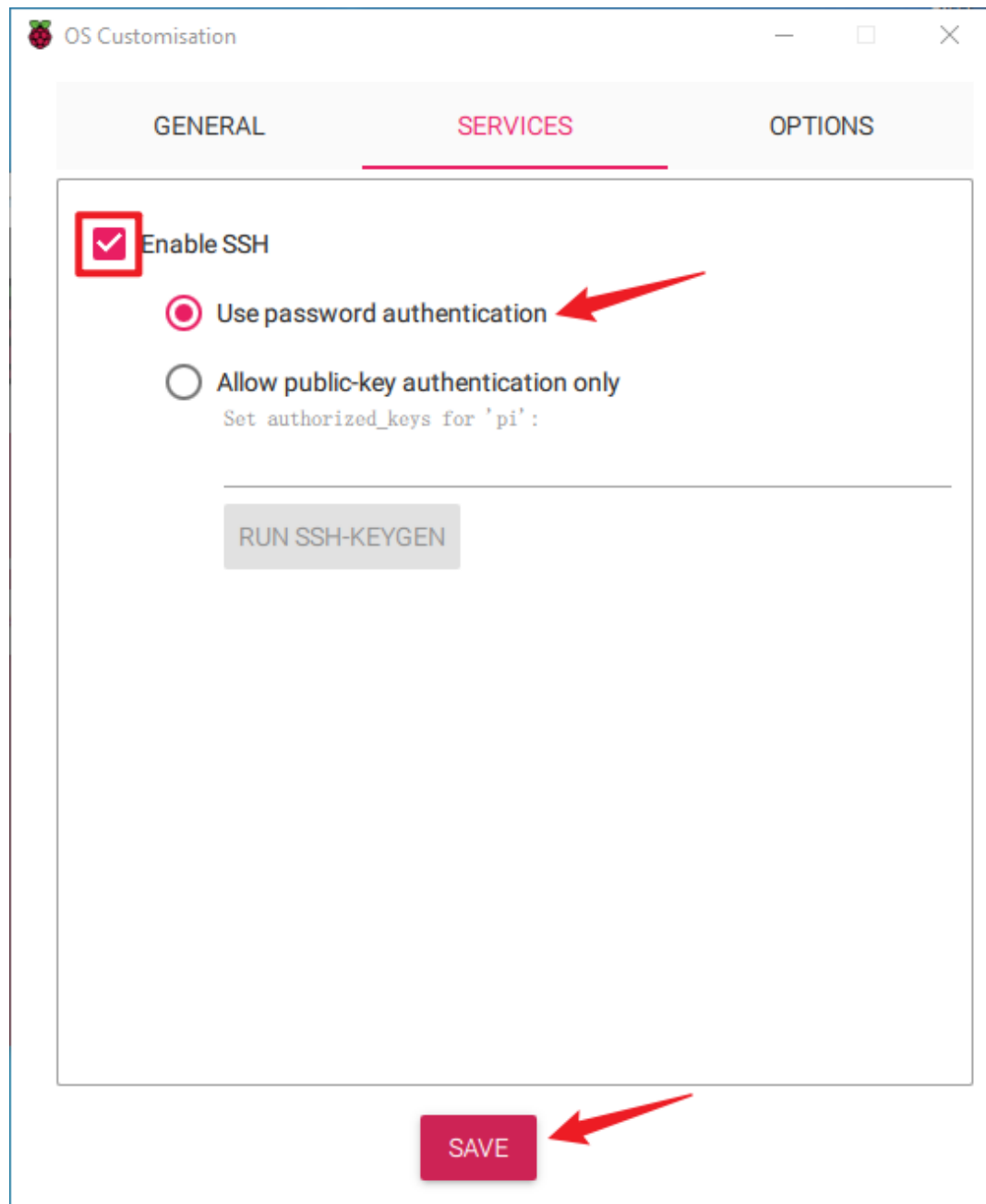
11. Set up wireless LAN by inputting your network's **SSID** and **Password**.

Note: Wireless LAN country should be set the two-letter [ISO/IEC alpha2 code](#) for the country in which you are using your Raspberry Pi.

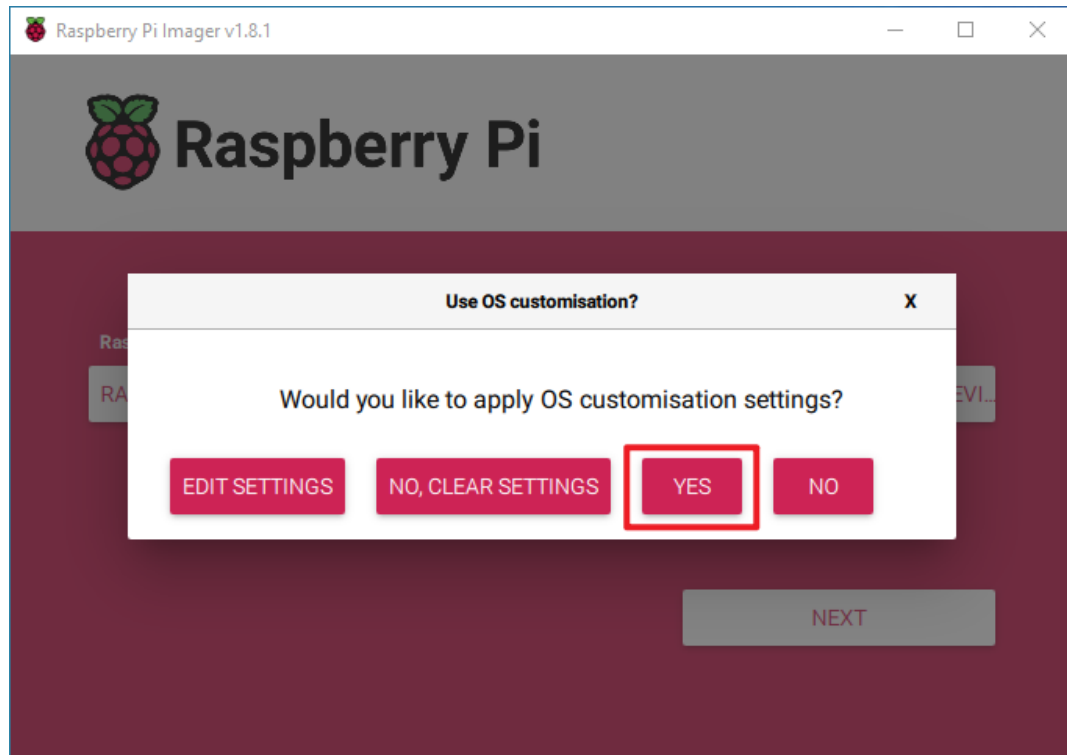


The screenshot displays a configuration window for the SunFounder PiCar-V. At the top, the option "Configure wireless LAN" is checked, indicated by a red square with a white checkmark. Below this, there are input fields for "SSID:" and "Password:", each with a red arrow pointing to it. Further down, there are two unchecked checkboxes: "Show password" and "Hidden SSID". Below these is a dropdown menu for "Wireless LAN country:" with "GB" selected, also indicated by a red arrow. At the bottom of the configuration section, there is an unchecked checkbox for "Set locale settings". Below this, there are two more dropdown menus: "Time zone:" with "Asia/Shanghai" selected, and "Keyboard layout:" with "US" selected. A red arrow points to the "GB" selection in the "Wireless LAN country:" dropdown. At the bottom center of the window is a red "SAVE" button.

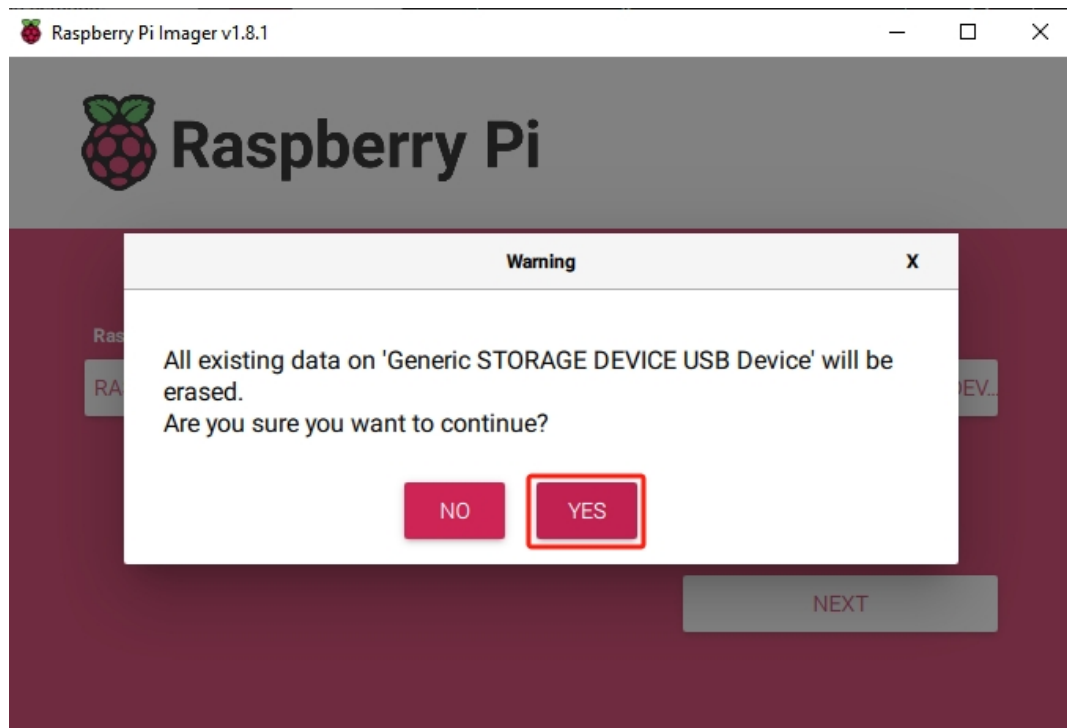
12. Click **SERVICES** and enable **SSH** for password-based remote access. Remember to click **Save**.



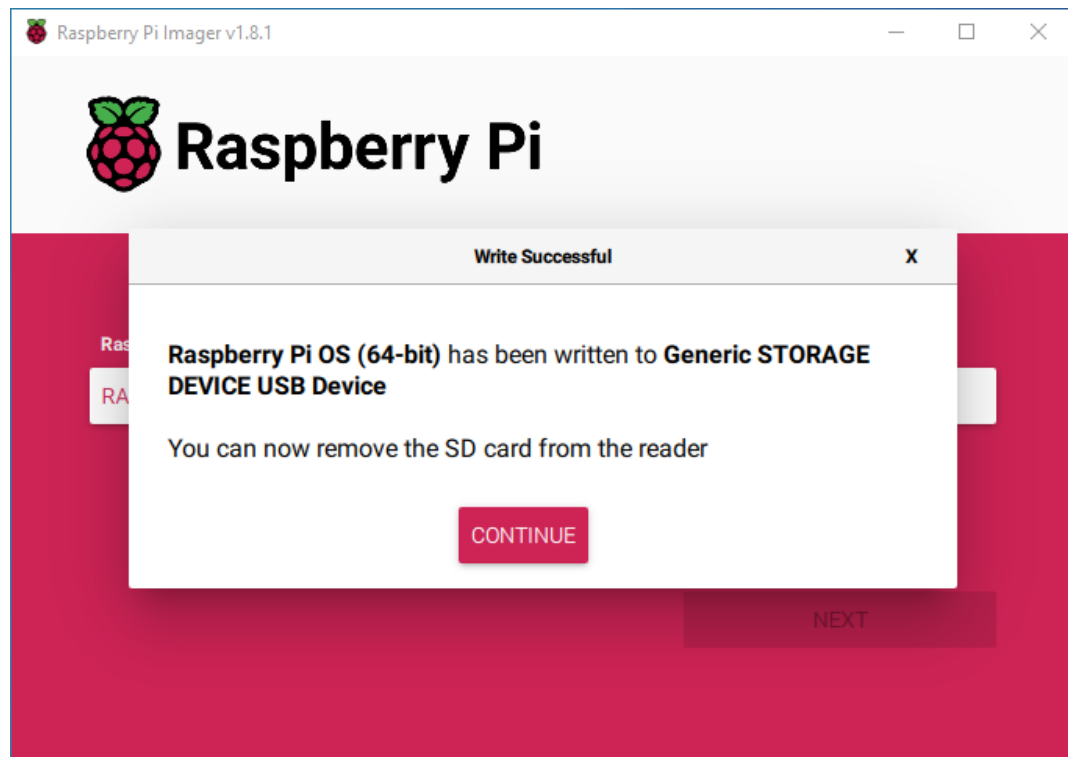
13. Confirm your choices by clicking **Yes**.



14. If your SD card has existing files, back them up to avoid data loss. Click **Yes** to proceed if no backup is necessary.



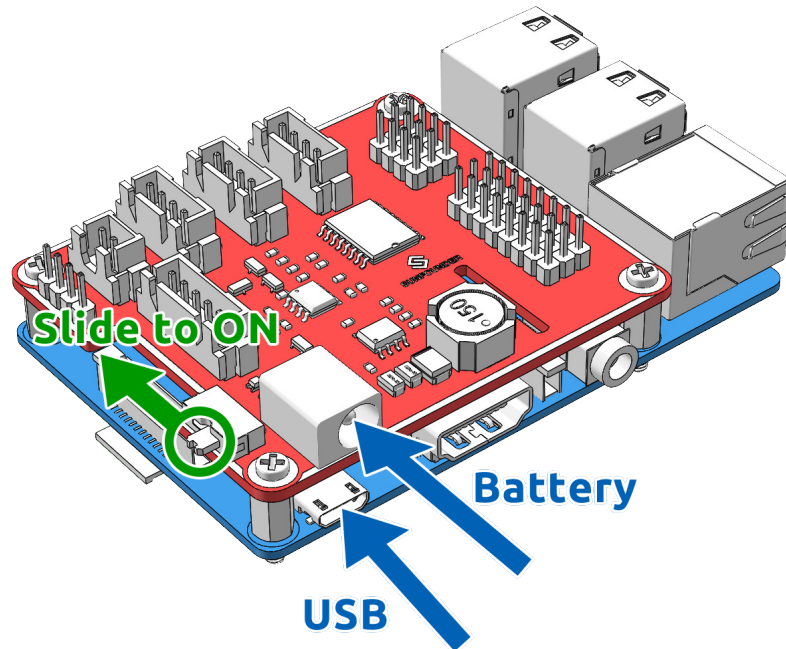
15. Wait as the OS is written to the SD card. Once completed, a confirmation window will appear.



1.5.3 Set up Your Raspberry Pi

Power Supply for Raspberry Pi (Important)

1. Insert the SD card set up with Raspberry Pi OS into the microSD card slot located on the underside of the Raspberry Pi.
2. **Following the assembly instructions, insert the battery cable and turn on the power switch. Next, insert the USB cable to Raspberry Pi.**



If You Have a Screen

Note: The Raspberry Pi ZERO installed on the Robot is not easy to connect to the screen, please use the method without a screen to set it up.

If you have a screen, it will be easy for you to operate on the Raspberry Pi.

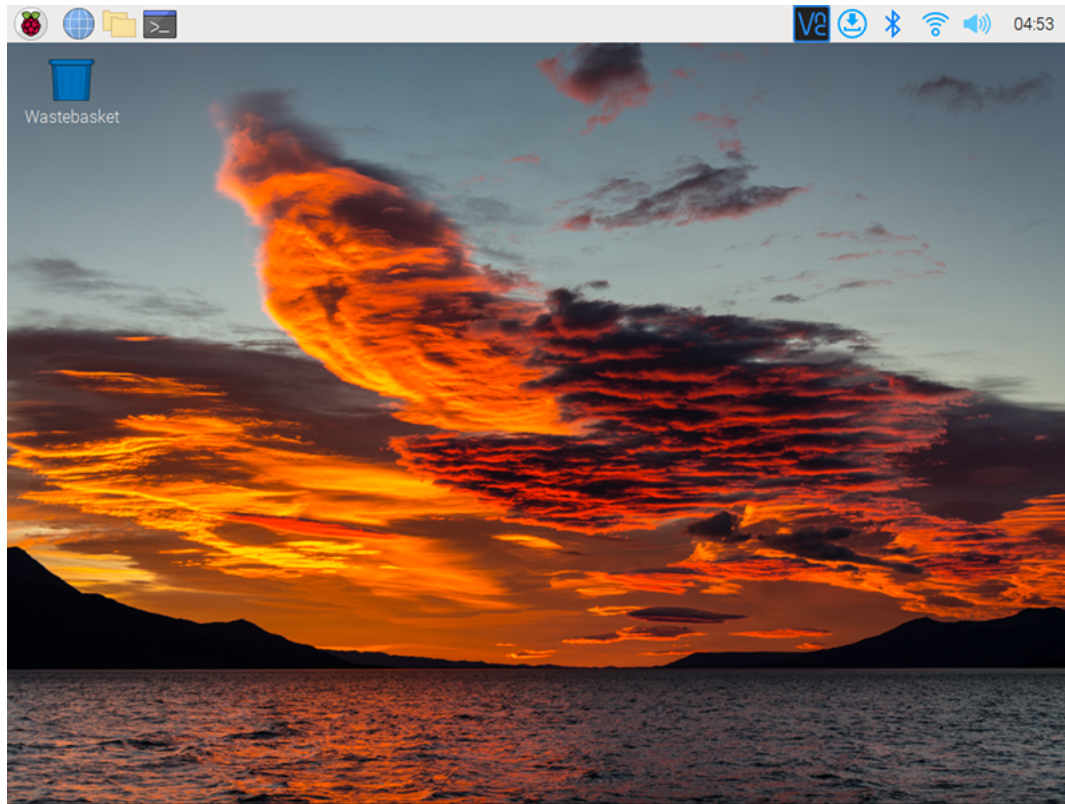
Required Components

- Raspberry Pi 4B/3B 3B+/2B
- 1 * Power Adapter
- 1 * Micro SD card
- 1 * Screen Power Adapter
- 1 * HDMI cable
- 1 * Screen
- 1 * Mouse
- 1 * Keyboard

1. Plug in the Mouse and Keyboard.
2. Connect the screen to Raspberry Pi's HDMI port and make sure your screen is plugged into a wall socket and switched on.

Note: If you use a Raspberry Pi 4, you need to connect the screen to the HDMI0 (nearest the power in port).

3. Use the power adapter to power the Raspberry Pi. After a few seconds, the Raspberry Pi OS desktop will be displayed.



If You Have No Screen

If you don't have a monitor, you can remotely log into your Raspberry Pi.

Required Components

- – Raspberry Pi 4B/Zero 2 w/3B 3B+/2B/Zero W
- 1 * Power Adapter
- 1 * Micro SD card

You can apply the SSH command to open the Raspberry Pi's Bash shell. Bash is the standard default shell for Linux. The shell itself is a command (instruction) when the user uses Unix/Linux. Most of what you need to do can be done through the shell.

If you're not satisfied with using the command window to access your Raspberry Pi, you can also use the remote desktop feature to easily manage files on your Raspberry Pi using a GUI.

See below for detailed tutorials for each system.

Mac OS X user

For Mac users, accessing the Raspberry Pi desktop directly via VNC is more convenient than from the command line. You can access it via Finder by entering the set account password after enabling VNC on the Raspberry Pi side.

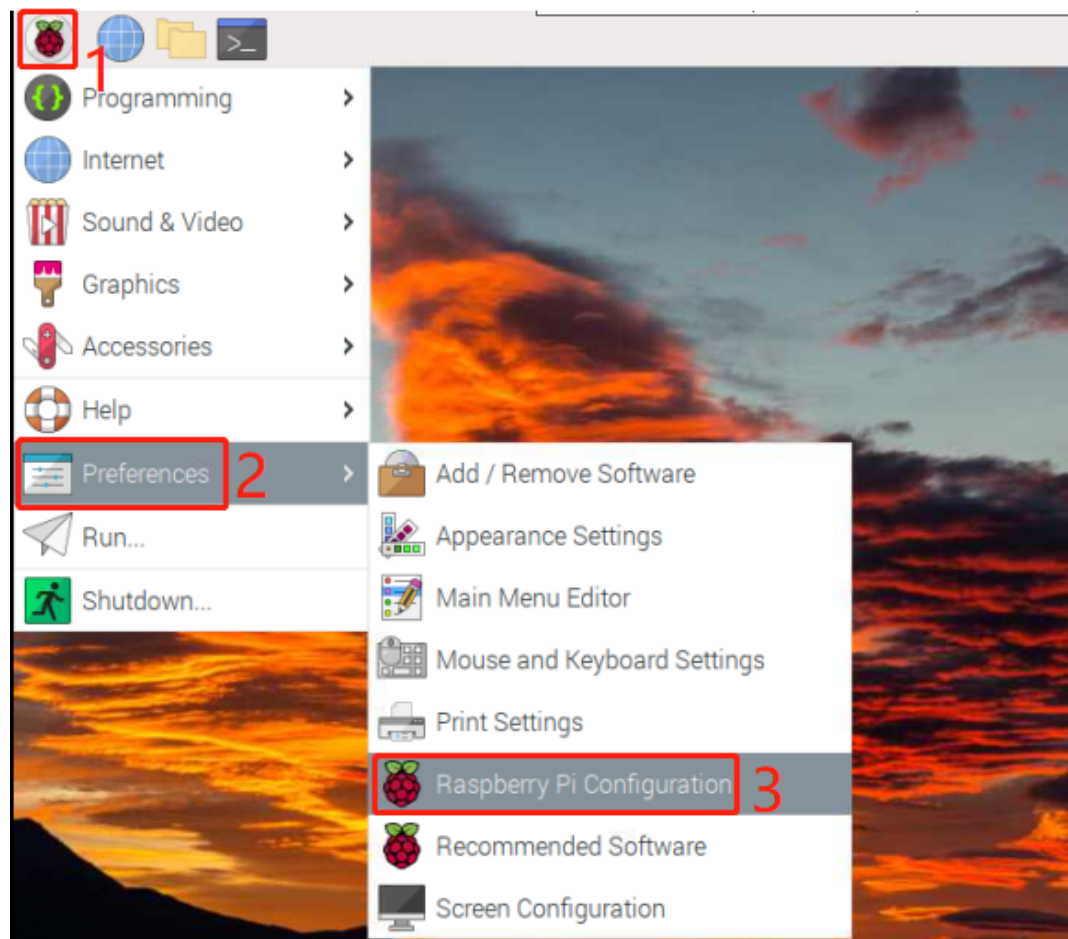
Note that this method does not encrypt communication between the Mac and Raspberry Pi. The communication will take place within your home or business network, so even if it's unprotected, it won't be an issue. However, if you are concerned about it, you can install a VNC application such as [VNC® Viewer](#).

Alternatively it would be handy if you could use a temporary monitor (TV), mouse and keyboard to open the Raspberry Pi desktop directly to set up VNC. If not, it doesn't matter, you can also use the SSH command to open the Raspberry Pi's Bash shell and then using the command to set up the VNC.

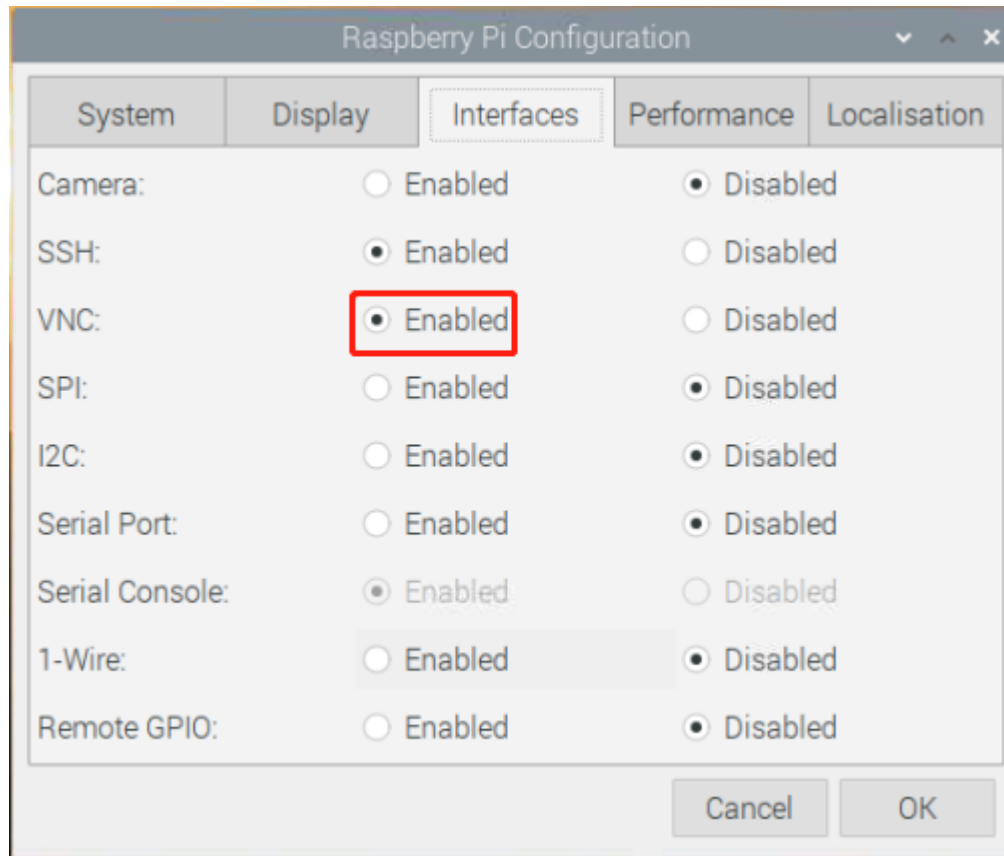
- *Have Temporarily Monitor (or TV)?*
- *Don't Have Temporarily Monitor (or TV)?*

Have Temporarily Monitor (or TV)?

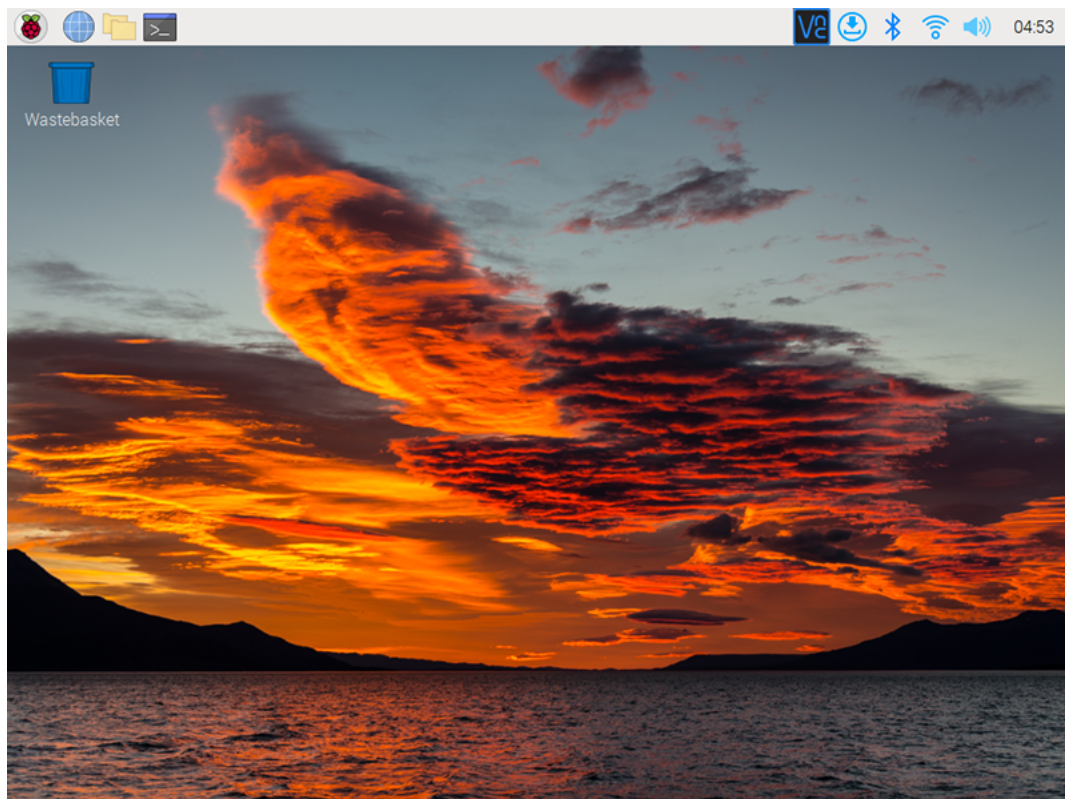
1. Connect a monitor (or TV), mouse and keyboard to the Raspberry Pi and power it on. Select the menu according to the numbers in the figure.



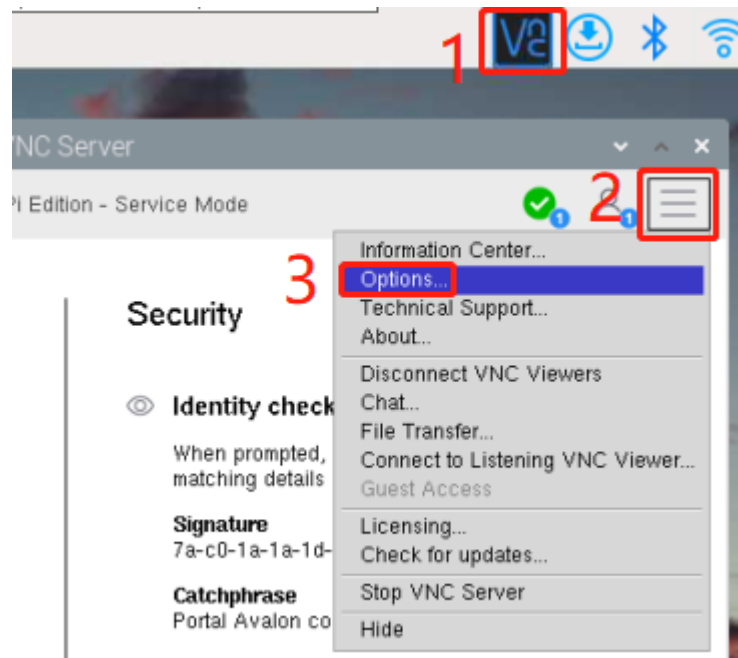
2. The following screen will be displayed. Set **VNC** to **Enabled** on the **Interfaces** tab, and click **OK**.



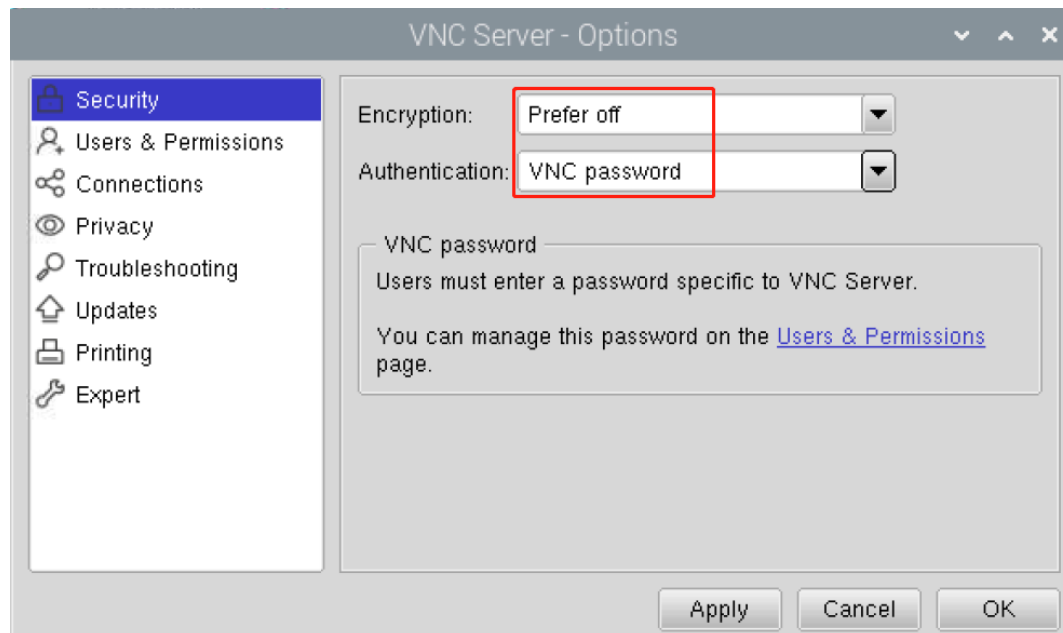
3. A VNC icon appears on the upper right of the screen and the VNC server starts.



4. Open the VNC server window by clicking on the **VNC** icon, then click on the **Menu** button in the top right corner and select **Options**.

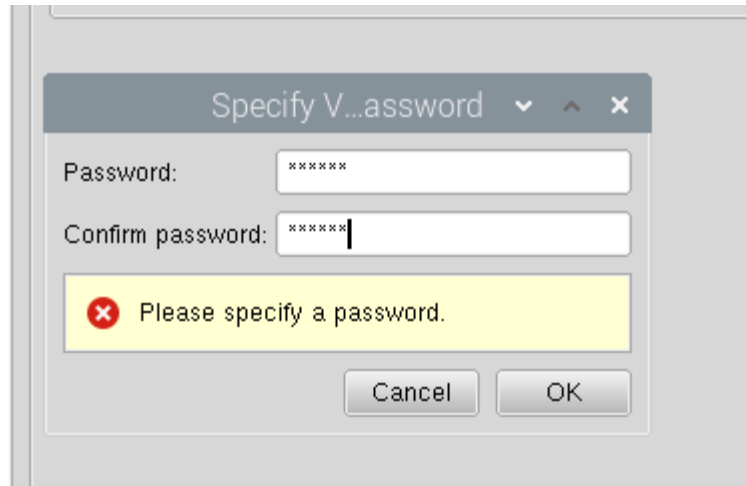


5. You will be presented with the following screen where you can change the options.



Set **Encryption** to **Prefer off** and **Authentication** to **VNC password**.

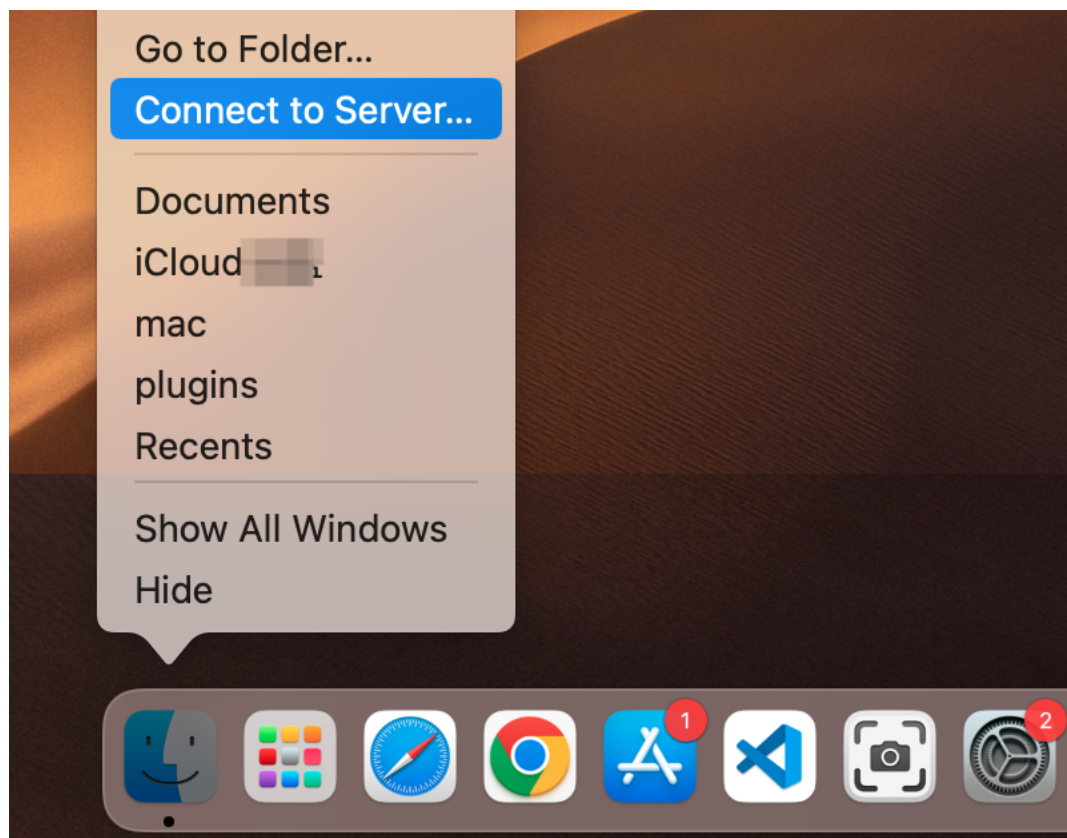
6. When you click the **OK** button, the password input screen is displayed. You can use the same password as the Raspberry pi password or a different password, so enter it and click **OK**.



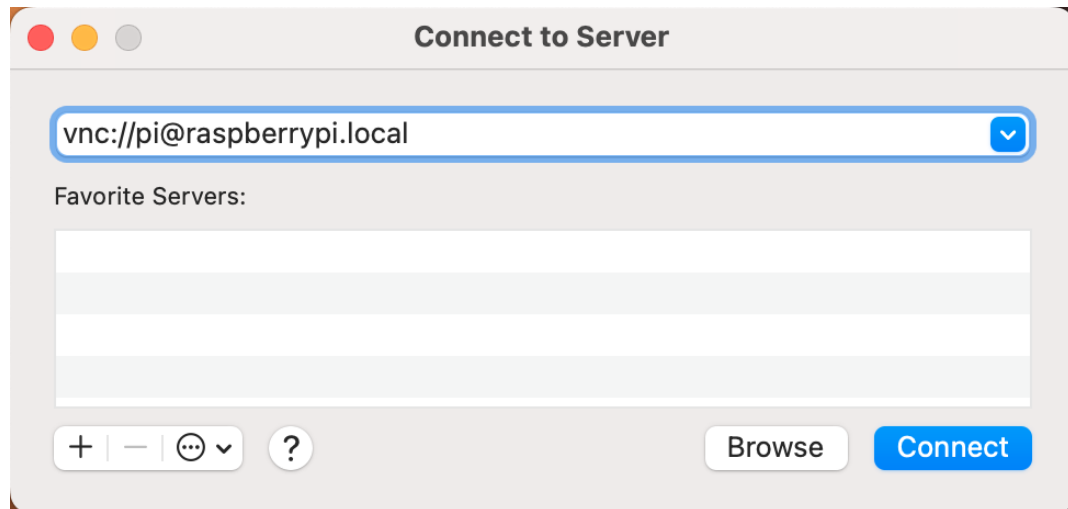
You are now ready to connect from your Mac. It's okay to disconnect the monitor.

From here, it will be the operation on the Mac side.

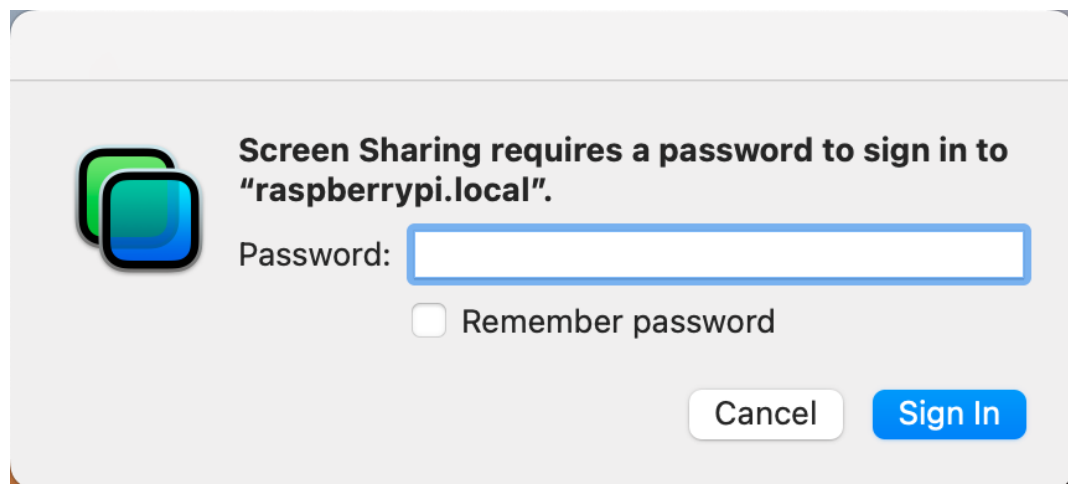
1. Now, select **Connect to Server** from the Finder's menu, which you can open by right-clicking.



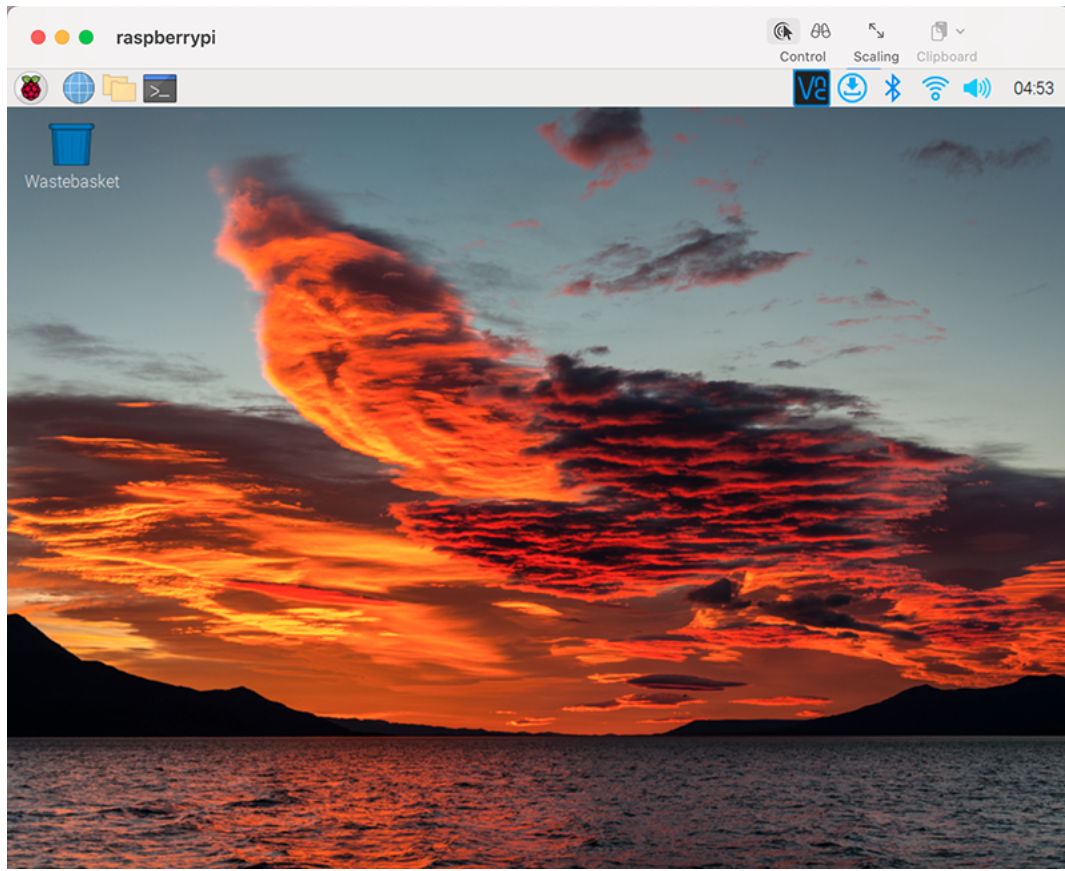
2. Type in `vnc://<username>@<hostname>.local` (or `vnc://<username>@<IP address>`). After entering, click **Connect**.



3. You will be asked for a password, so please enter it.



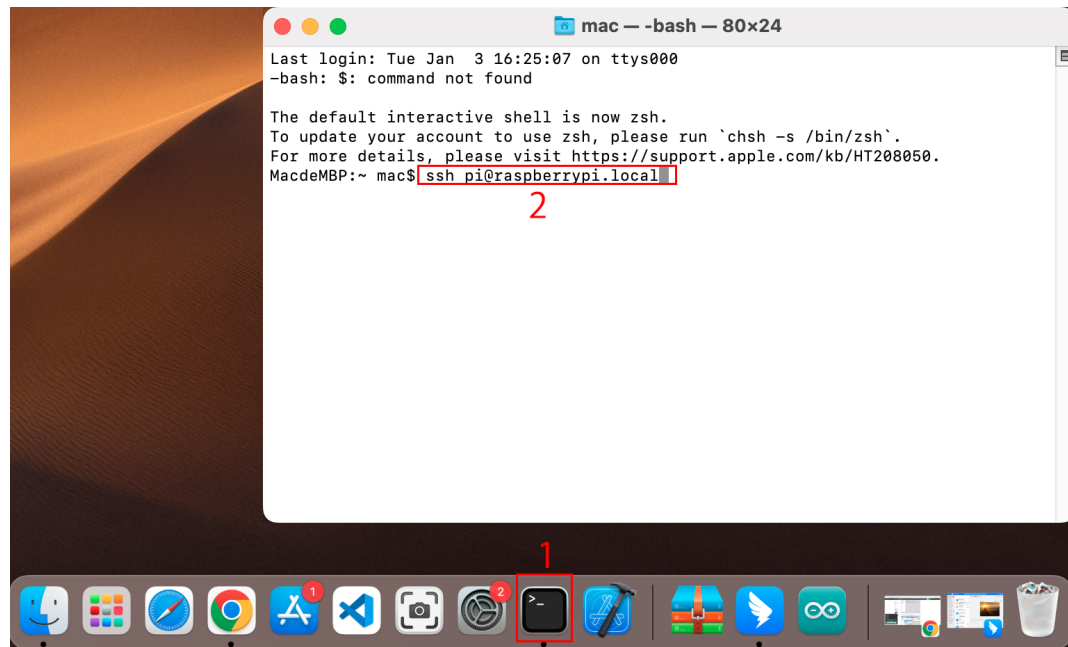
4. The desktop of the Raspberry pi will be displayed, and you will be able to operate it from the Mac as it is.



Don't Have Temporarily Monitor (or TV)?

- You can apply the SSH command to open the Raspberry Pi's Bash shell.
 - Bash is the standard default shell for Linux.
 - The shell itself is a command (instruction) when the user uses Unix/Linux.
 - Most of what you need to do can be done through the shell.
 - After setting up the Raspberry pi side, you can access the desktop of the Raspberry Pi using the **Finder** from the Mac.
1. Type `ssh <username>@<hostname>.local` to connect to the Raspberry Pi.

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

2. The following message will be displayed only when you log in for the first time, so enter **yes**.

```
The authenticity of host 'raspberrypi.local
(2400:2410:2101:5800:635b:f0b6:2662:8cba)' can't be established.
ED25519 key fingerprint is SHA256:oo7x3ZSgAo032wD1tE8eW0fFM/
kmewIvRwkBys6XRwg.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])?
```

3. Enter the password for the Raspberry pi. The password you enter will not be displayed, so be careful not to make a mistake.

```
pi@raspberrypi.local's password:
Linux raspberrypi 5.15.61-v8+ #1579 SMP PREEMPT Fri Aug 26 11:16:44 BST
2022 aarch64

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Thu Sep 22 12:18:22 2022
pi@raspberrypi:~ $
```

4. Set up your Raspberry Pi so that you can log in via VNC from your Mac once you have successfully logged into it. The first step is to update your operating system by running the following commands.

```
sudo apt update
sudo apt upgrade
```

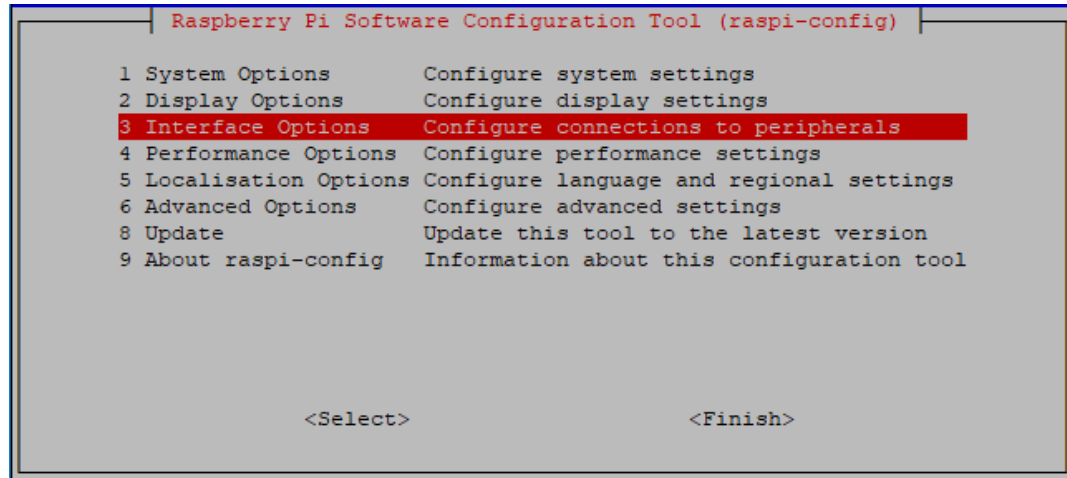
Do you want to continue? [Y/n], Enter Y when prompted.

It may take some time for the update to finish. (It depends on the amount of updates at that time.)

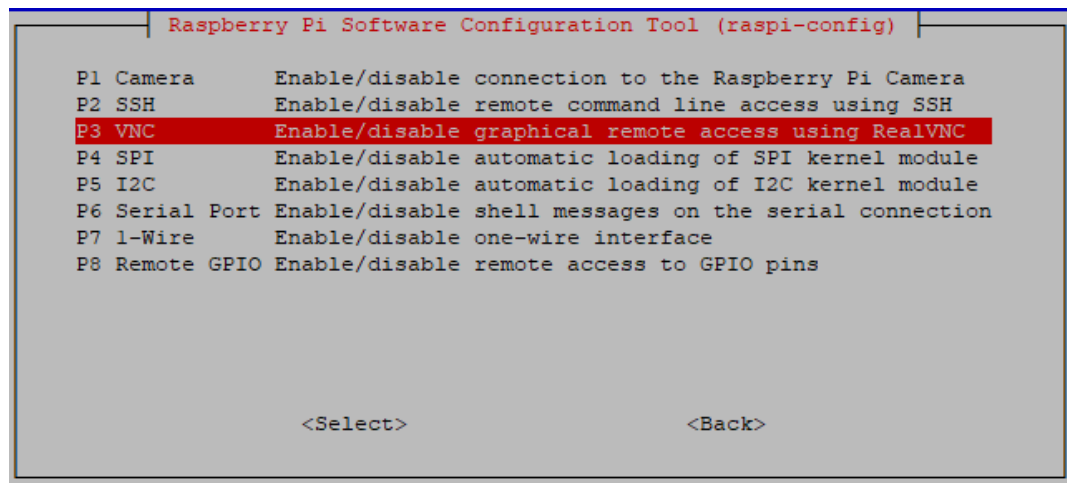
5. Enter the following command to enable the **VNC Server**.

```
sudo raspi-config
```

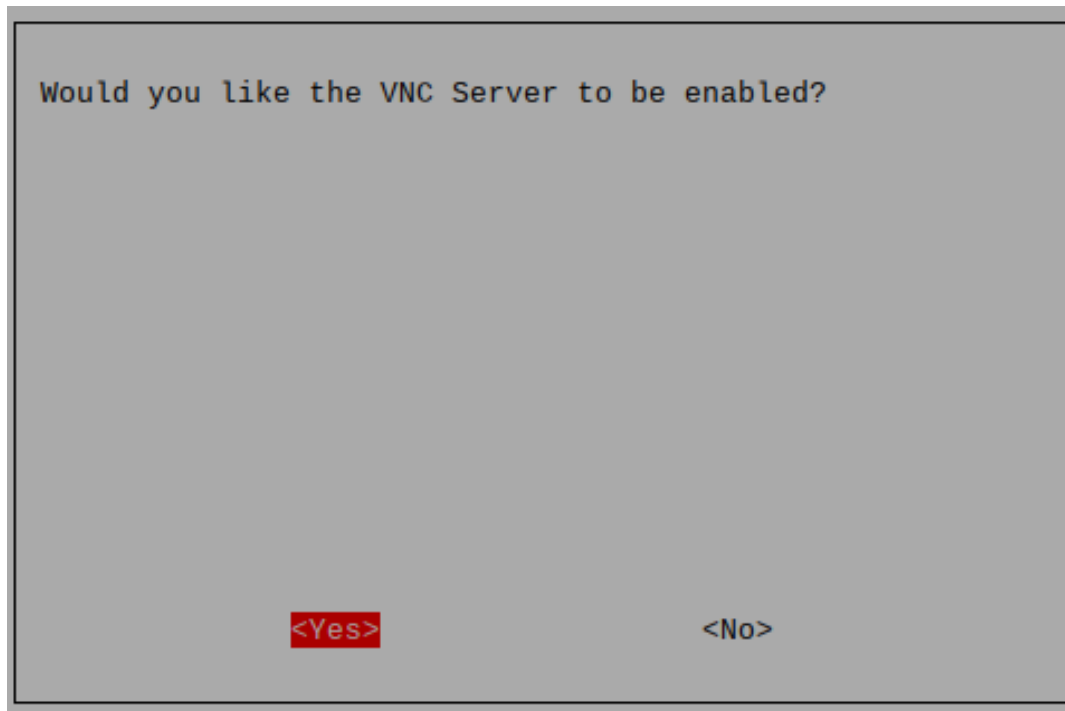
6. The following screen will be displayed. Select **Interface Options** with the arrow keys on the keyboard and press the **Enter** key.



7. Then select **VNC**.



8. Use the arrow keys on the keyboard to select **<Yes>** -> **<OK>** -> **<Finish>** to complete the setup.

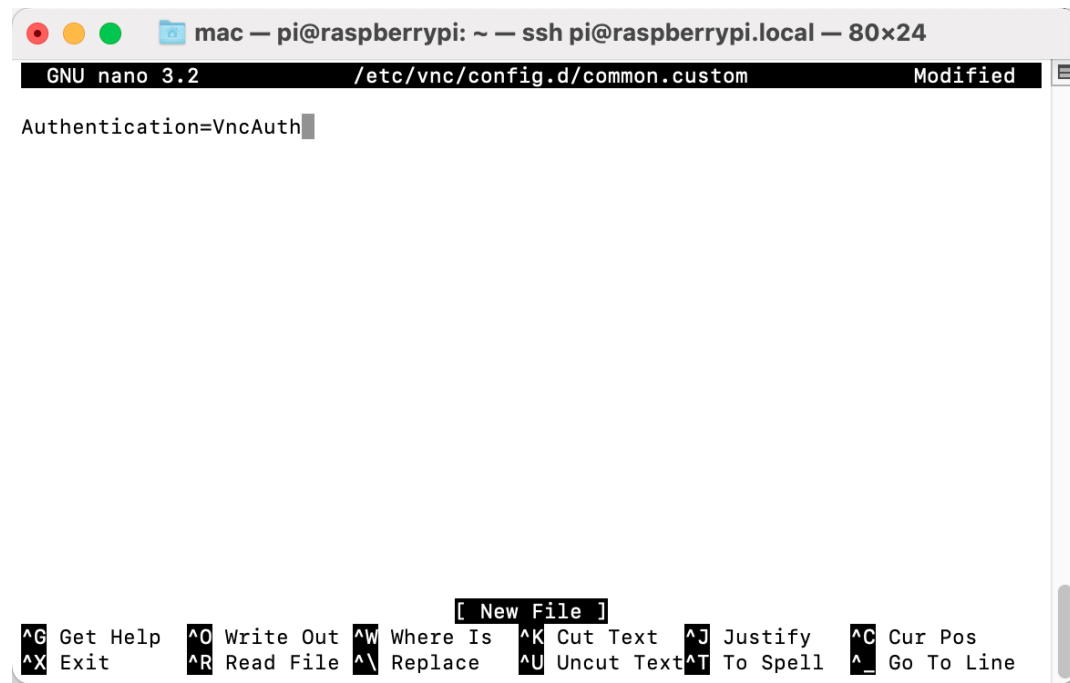


9. Now that the VNC server has started, let's change the settings for connecting from a Mac.

To specify parameters for all programs for all user accounts on the computer, create `/etc/vnc/config.d/common.custom`.

```
sudo nano /etc/vnc/config.d/common.custom
```

After entering `Authentication=VncAuthenter`, press `Ctrl+X` -> `Y` -> `Enter` to save and exit.



10. In addition, set a password for logging in via VNC from a Mac. You can use the same password as the Raspberry

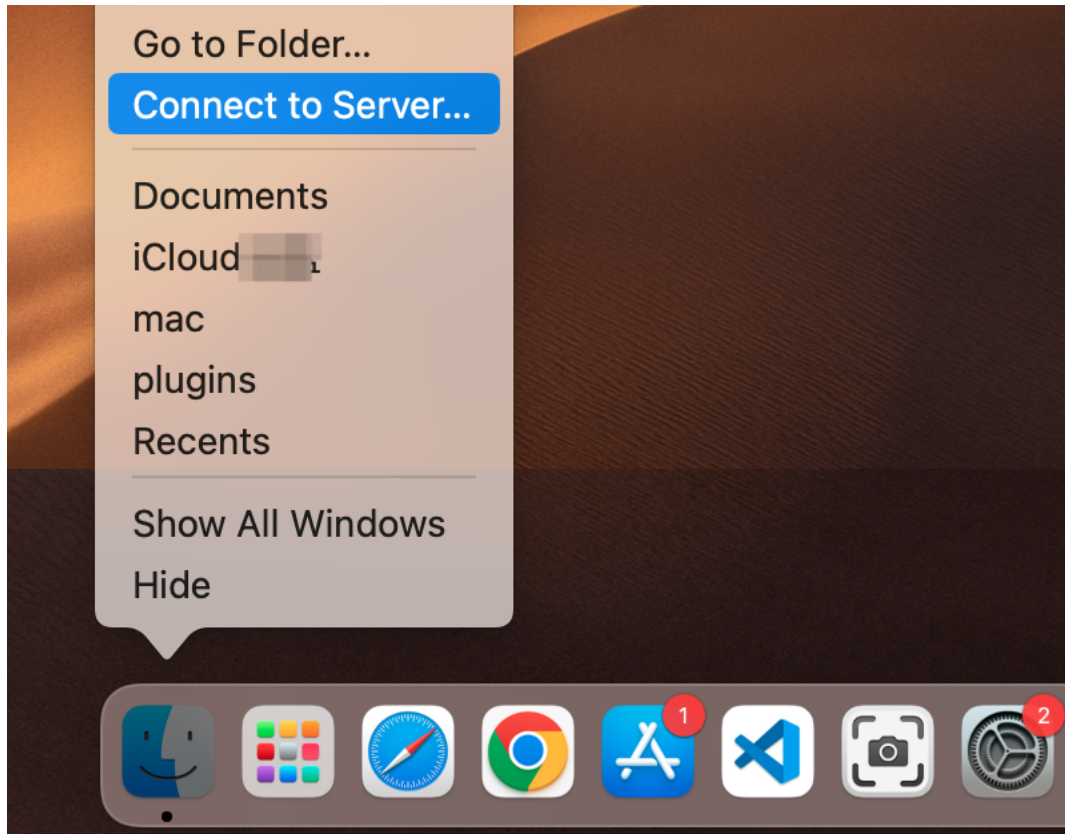
pi password or a different password.

```
sudo vncpasswd -service
```

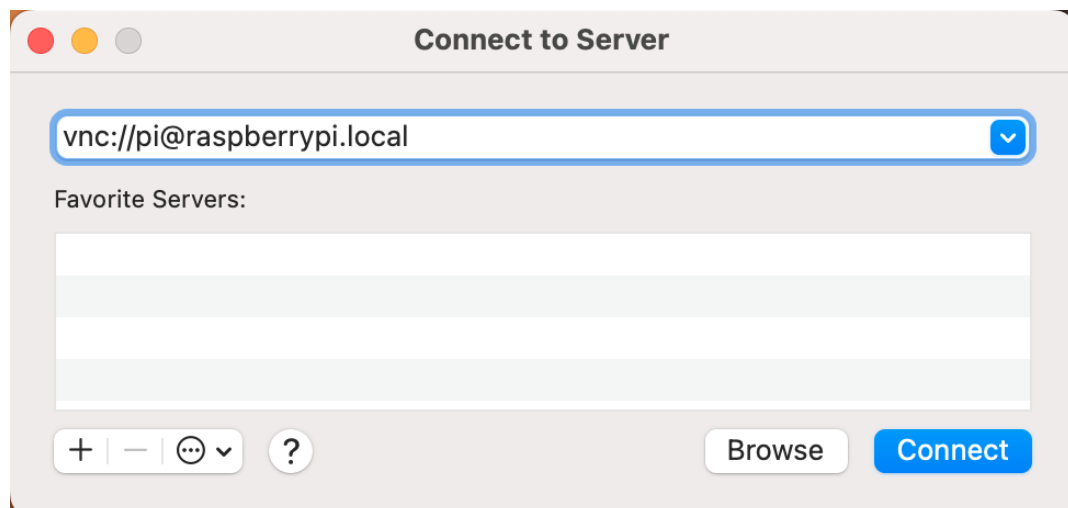
11. Once the setup is complete, restart the Raspberry Pi to apply the changes.

```
sudo sudo reboot
```

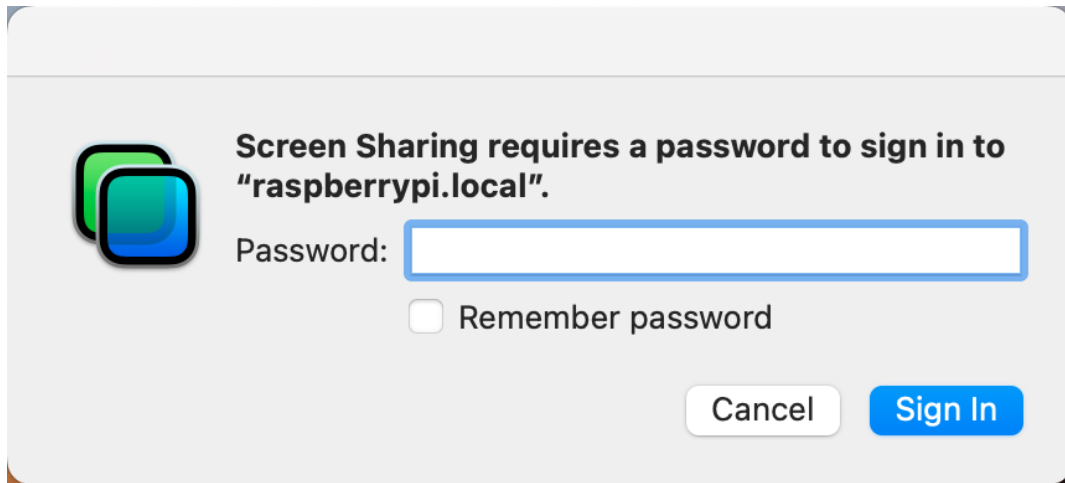
12. Now, select **Connect to Server** from the **Finder**'s menu, which you can open by right-clicking.



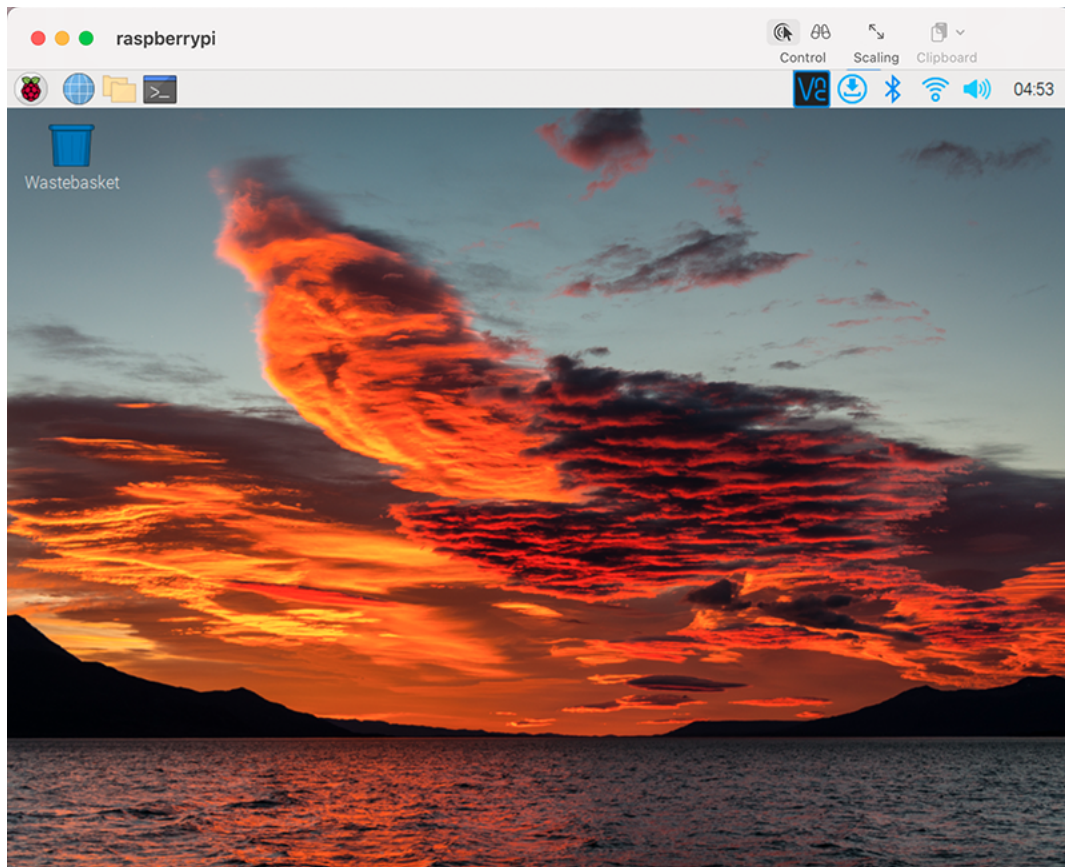
13. Type in `vnc://<username>@<hostname>.local` (or `vnc://<username>@<IP address>`). After entering, click **Connect**.



14. You will be asked for a password, so please enter it.



15. The desktop of the Raspberry pi will be displayed, and you will be able to operate it from the Mac as it is.

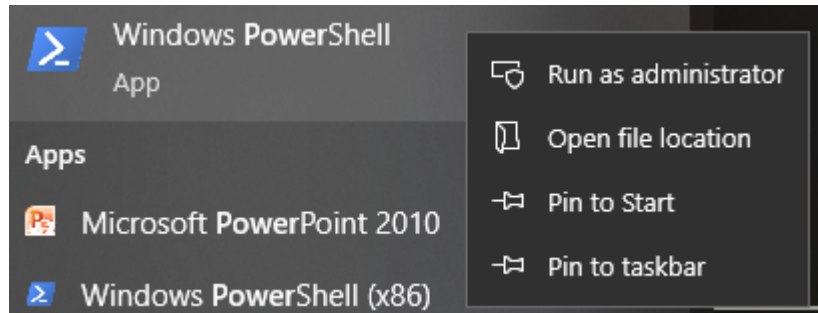


Windows Users

Login Raspberry Pi Remotely

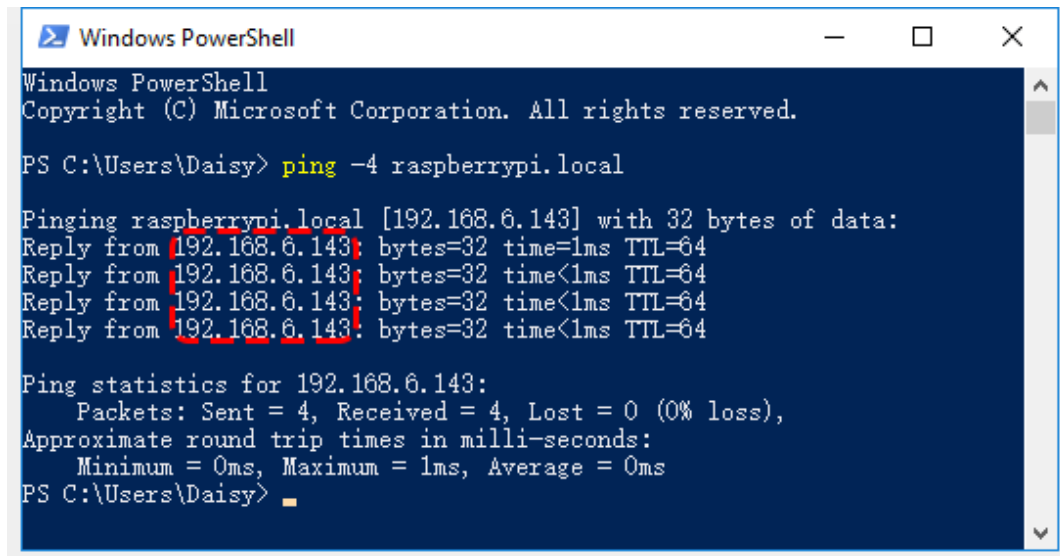
If you are using win10, you can use follow way to login Raspberry Pi remotely.

1. Type powershell in the search box of your Windows desktop, right click on the Windows PowerShell, and select Run as administrator from the menu that appears.



2. Then, check the IP address of your Raspberry Pi by typing in `ping -4 <hostname>.local`.

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



As shown above, you can see the Raspberry Pi's IP address after it has been connected to the network.

- If terminal prompts Ping request could not find host pi.local. Please check the name and try again.. Please follow the prompts to make sure the hostname you fill in is correct.
 - Still can't get the IP? Check your network or WiFi configuration on the Raspberry Pi.
3. At this point you will be able to log in to your Raspberry Pi using the `ssh <username>@<hostname>.local` (or `ssh <username>@<IP address>`).

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

Warning: If a prompt appears The term 'ssh' is not recognized as the name of a cmdlet....

It means your system is too old and does not have ssh tools pre-installed, you need to manually *Install OpenSSH via Powershell*.

Or use a third party tool like *PuTTY*.

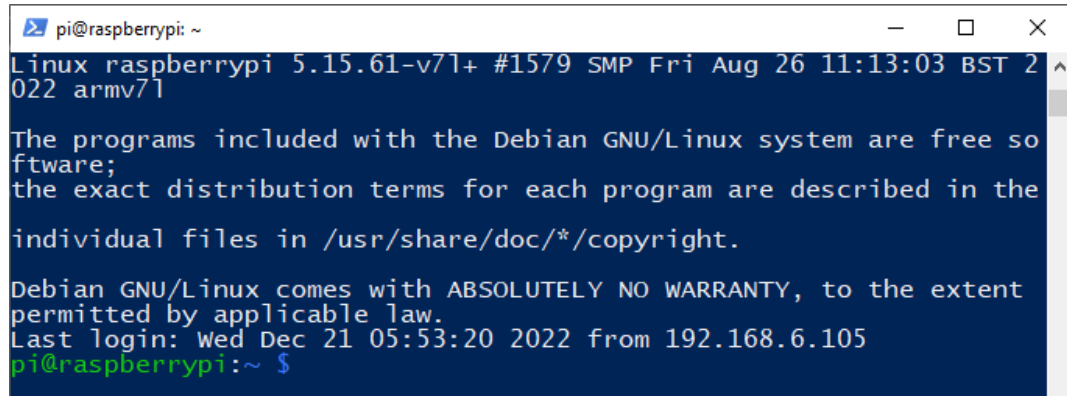
- The following message will be displayed only when you log in for the first time, so enter yes.

```
The authenticity of host 'raspberrypi.local
(2400:2410:2101:5800:635b:f0b6:2662:8cba)' can't be established.
ED25519 key fingerprint is SHA256:oo7x3ZSgAo032wD1tE8eW0fFM/
kmewIvRwkBys6XRwg.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])?
```

- Input the password you set before. (Mine is raspberry.)

Note: When you input the password, the characters do not display on window accordingly, which is normal. What you need is to input the correct password.

- We now get the Raspberry Pi connected and are ready to go to the next step.



```
pi@raspberrypi: ~
Linux raspberrypi 5.15.61-v7l+ #1579 SMP Fri Aug 26 11:13:03 BST 2022 armv7l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Wed Dec 21 05:53:20 2022 from 192.168.6.105
pi@raspberrypi:~ $
```

Remote Desktop

If you're not satisfied with using the command window to access your Raspberry Pi, you can also use the remote desktop feature to easily manage files on your Raspberry Pi using a GUI.

Here we use [VNC® Viewer](#).

Enable VNC service

The VNC service has been installed in the system. By default, VNC is disabled. You need to enable it in config.

- Input the following command:

```
sudo raspi-config
```

```

pi@raspberrypi: ~
login as: pi
pi@192.168.0.234's password:

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Mon Feb 20 09:18:17 2017 from daisy-pc.lan
pi@raspberrypi:~ $ sudo raspi-config

```

2. Choose **3 Interfacing Options** by press the down arrow key on your keyboard, then press the **Enter** key.

```

Raspberry Pi Software Configuration Tool (raspi-config)

1 System Options          Configure system settings
2 Display Options         Configure display settings
3 Interface Options       Configure connections to peripherals
4 Performance Options     Configure performance settings
5 Localisation Options    Configure language and regional settings
6 Advanced Options        Configure advanced settings
8 Update                 Update this tool to the latest version
9 About raspi-config      Information about this configuration tool

<Select>                  <Finish>

```

3. Then **VNC**.

```

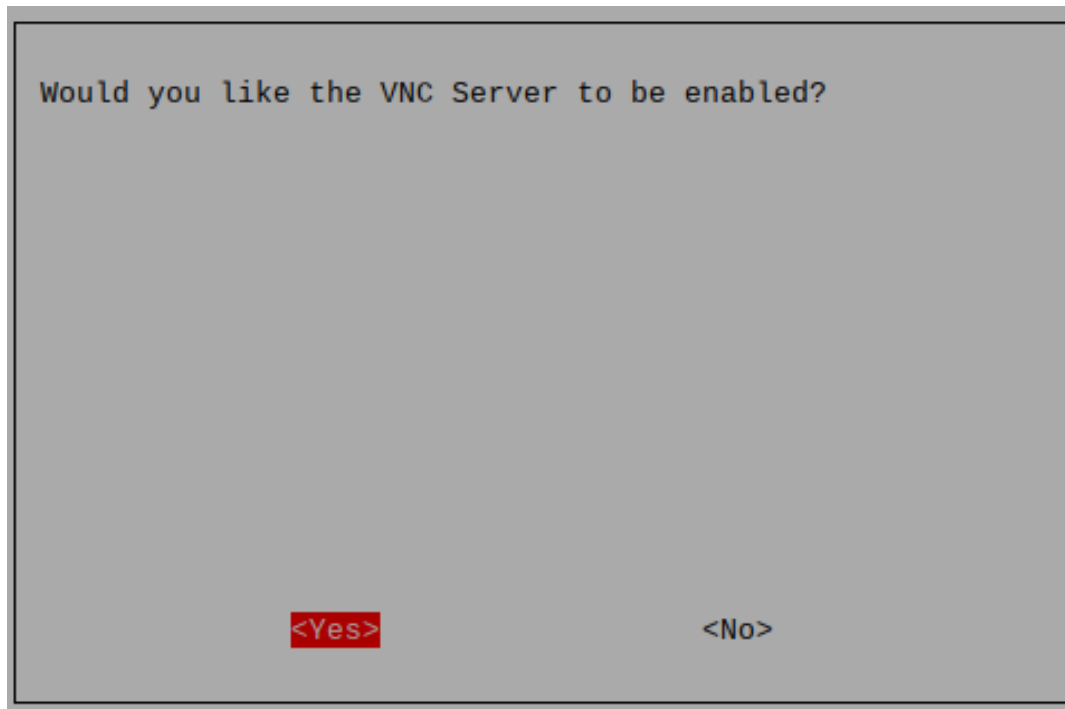
Raspberry Pi Software Configuration Tool (raspi-config)

P1 Camera                Enable/disable connection to the Raspberry Pi Camera
P2 SSH                   Enable/disable remote command line access using SSH
P3 VNC                   Enable/disable graphical remote access using RealVNC
P4 SPI                   Enable/disable automatic loading of SPI kernel module
P5 I2C                   Enable/disable automatic loading of I2C kernel module
P6 Serial Port           Enable/disable shell messages on the serial connection
P7 1-Wire                Enable/disable one-wire interface
P8 Remote GPIO           Enable/disable remote access to GPIO pins

<Select>                  <Back>

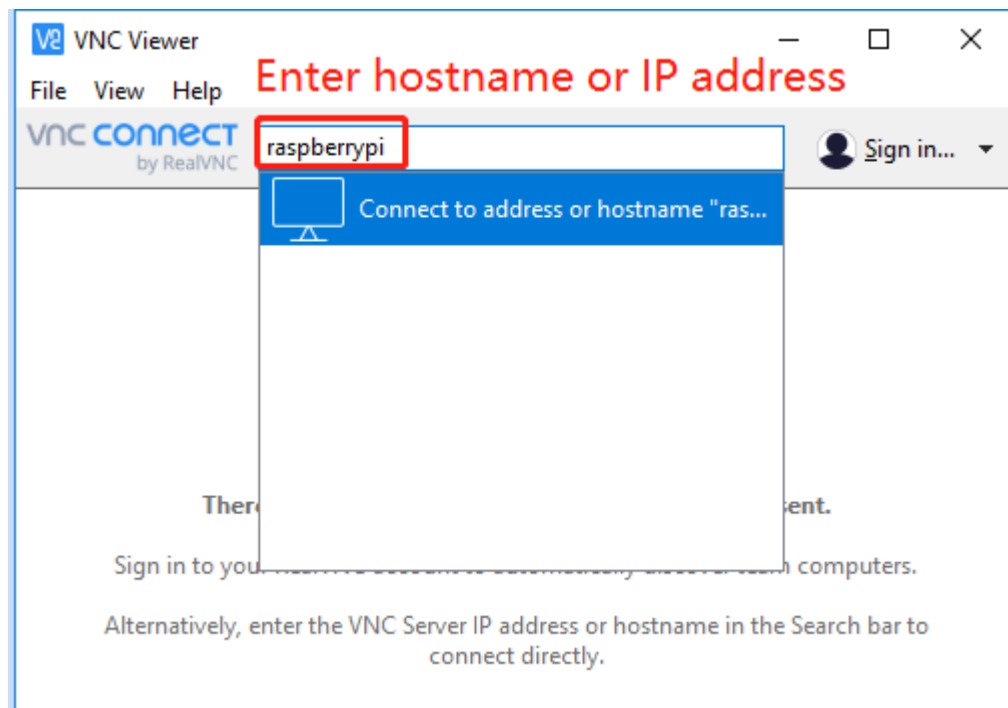
```

4. Use the arrow keys on the keyboard to select **<Yes>** -> **<OK>** -> **<Finish>** to complete the setup.

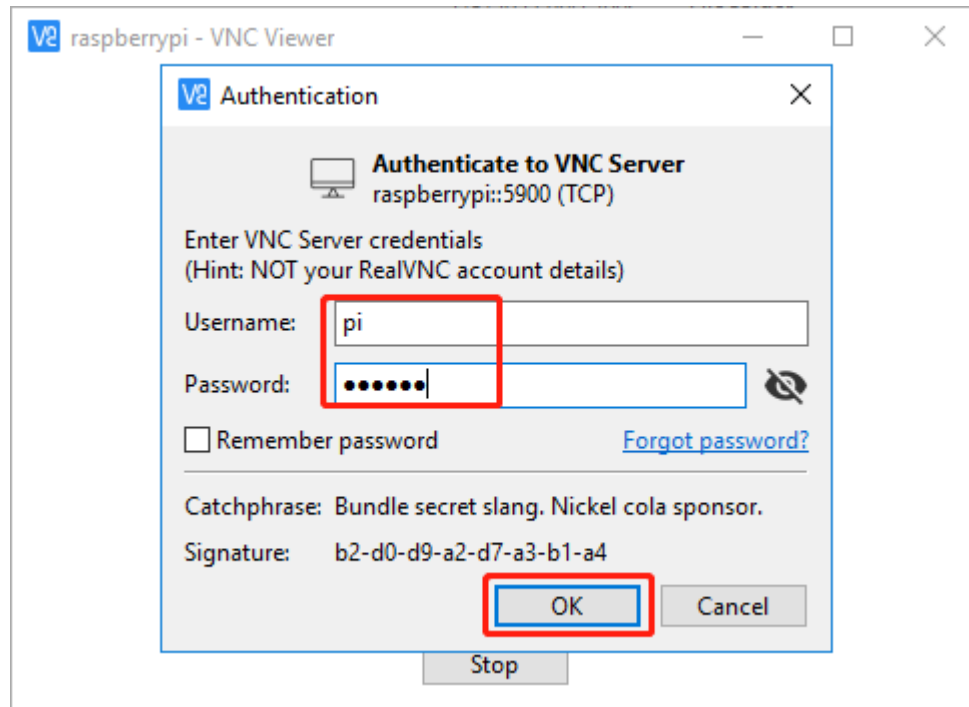


Login to VNC

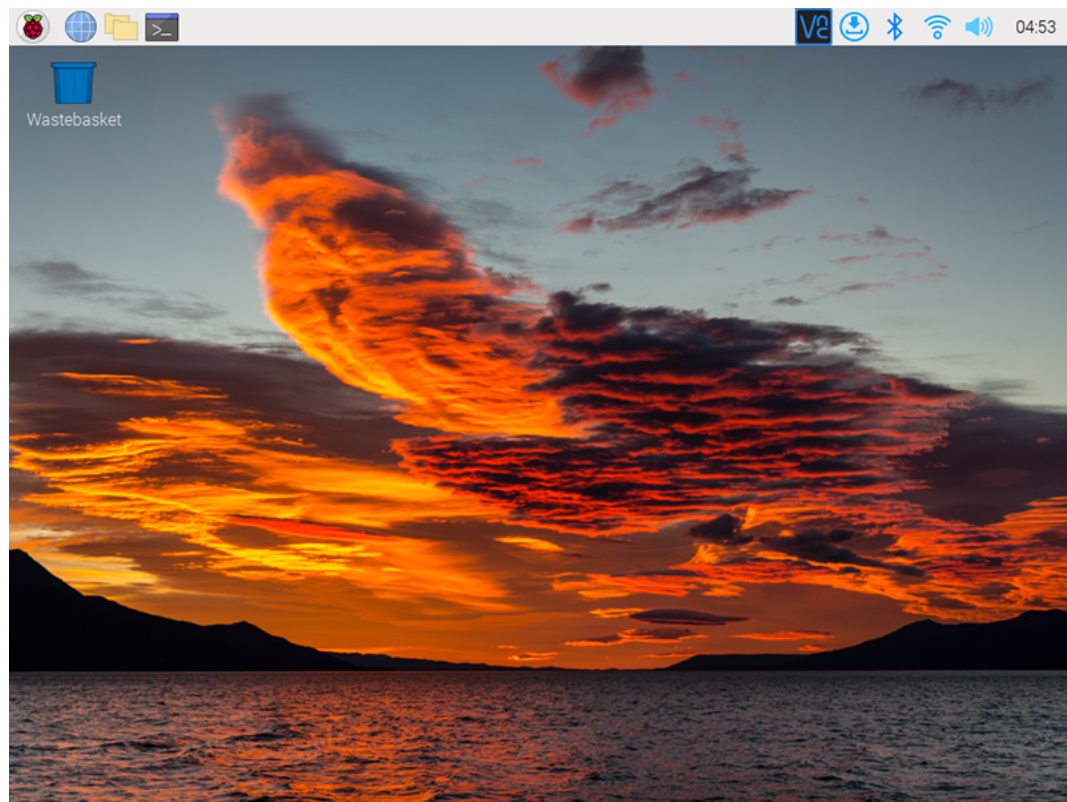
1. You need to download and install the [VNC Viewer](#) on personal computer.
2. Open it once the installation is complete. Then, enter the host name or IP address and press Enter.



3. After entering your Raspberry Pi name and password, click **OK**.

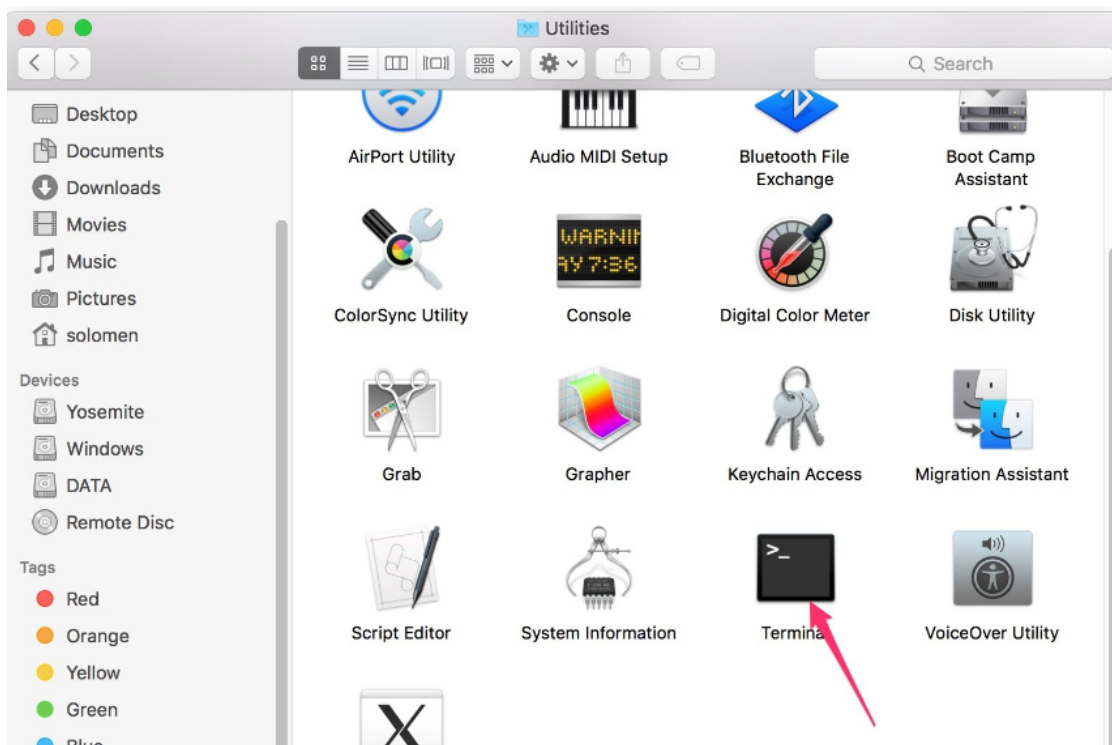


4. Now you can see the desktop of the Raspberry Pi.



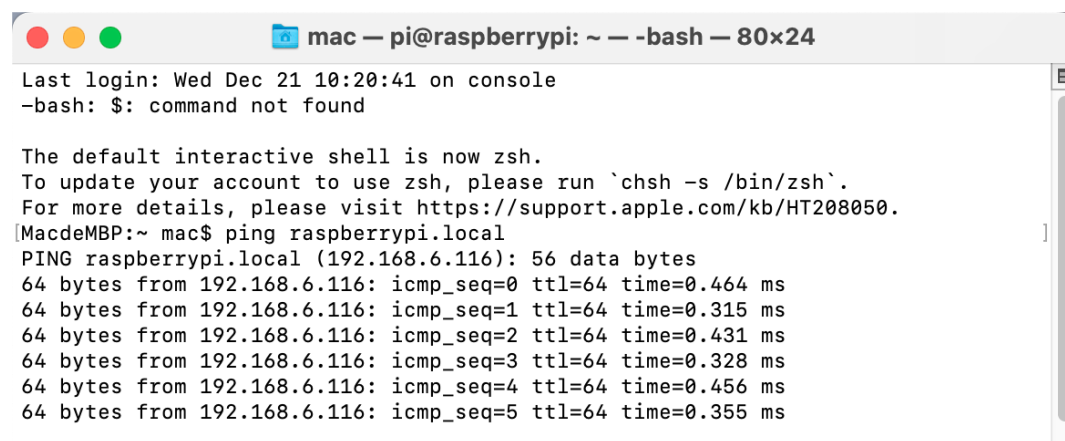
Linux /Unix Users

#. Go to **Applications->Utilities**, find the **Terminal**, and open it.



1. Check if your Raspberry Pi is on the same network by type in `ping <hostname>.local`.

```
ping raspberrypi.local
```



As shown above, you can see the Raspberry Pi's IP address after it has been connected to the network.

- If terminal prompts `Ping request could not find host pi.local`. Please check the name and try again.. Please follow the prompts to make sure the hostname you fill in is correct.
- Still can't get the IP? Check your network or WiFi configuration on the Raspberry Pi.

2. Type in `ssh <username>@<hostname>.local` (or `ssh <username>@<IP address>`).


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

Note: If a prompt appears The term 'ssh' is not recognized as the name of a cmdlet....

It means your system is too old and does not have ssh tools pre-installed, you need to manually *Install OpenSSH via Powershell*.

Or use a third party tool like *PuTTY*.

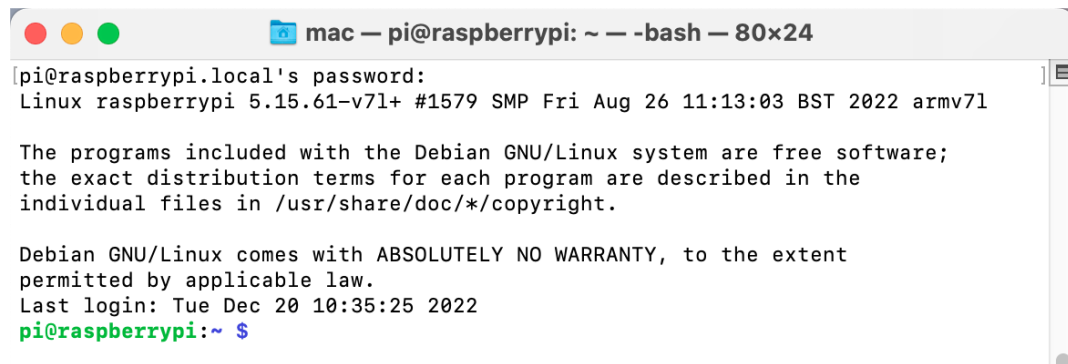
- The following message will be displayed only when you log in for the first time, so enter yes.

```
The authenticity of host 'raspberrypi.local'
(2400:2410:2101:5800:635b:f0b6:2662:8cba)' can't be established.
ED25519 key fingerprint is SHA256:oo7x3ZSgAo032wD1tE8eW0fFM/
kmewIvRwkBys6XRwg.
This key is not known by any other names
Are you sure you want to continue connecting (yes/no/[fingerprint])?
```

- Input the password you set before. (Mine is raspberry.)

Note: When you input the password, the characters do not display on window accordingly, which is normal. What you need is to input the correct password.

- We now get the Raspberry Pi connected and are ready to go to the nextstep.



```
mac — pi@raspberrypi: ~ — -bash — 80x24
pi@raspberrypi.local's password:
Linux raspberrypi 5.15.61-v7l+ #1579 SMP Fri Aug 26 11:13:03 BST 2022 armv7l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Tue Dec 20 10:35:25 2022
pi@raspberrypi:~ $
```

1.6 Servo Configuration

And since the servos used in this kit are adjusted by software and there's no such physical sticking point as other servos, here we need to configure the servo via software. First you need to finish some software installation before the configuration.

Note: Please do not forget to put in the battery and slide the power switch to ON in this chapter.

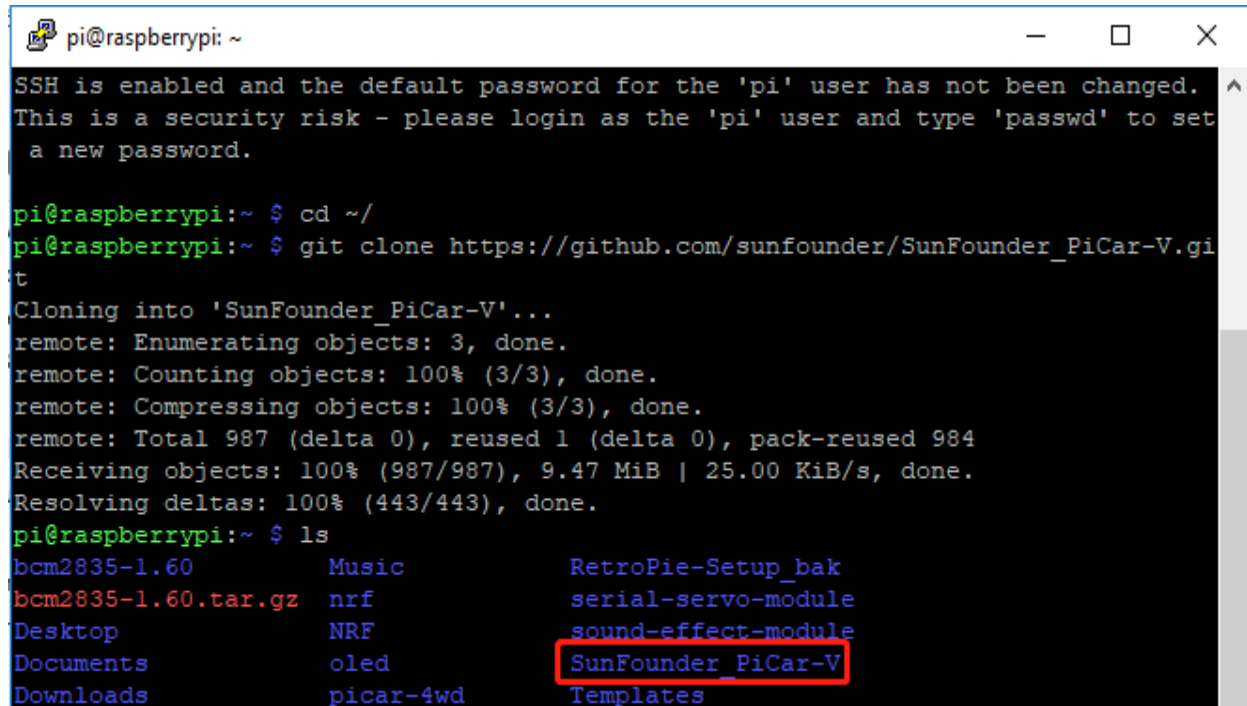
1.6.1 Get Source Code

You can find the source code in our Github repositories. Download the source code by *git clone*:

```
cd ~/
git clone https://github.com/sunfounder/SunFounder_PiCar-V -b V3.0
```

Note: Please pay attention to your typing – if you get the prompt of entering your user name and password, you may have typed wrong. If unluckily you did so, press Ctrl + C to exit and try again.

Check by the **ls** command, then you can see the code directory *SunFounder_PiCar-V*:



```
pi@raspberrypi: ~
SSH is enabled and the default password for the 'pi' user has not been changed.
This is a security risk - please login as the 'pi' user and type 'passwd' to set
a new password.

pi@raspberrypi:~ $ cd ~/
pi@raspberrypi:~ $ git clone https://github.com/sunfounder/SunFounder_PiCar-V.git
Cloning into 'SunFounder_PiCar-V'...
remote: Enumerating objects: 3, done.
remote: Counting objects: 100% (3/3), done.
remote: Compressing objects: 100% (3/3), done.
remote: Total 987 (delta 0), reused 1 (delta 0), pack-reused 984
Receiving objects: 100% (987/987), 9.47 MiB | 25.00 KiB/s, done.
Resolving deltas: 100% (443/443), done.
pi@raspberrypi:~ $ ls
bcm2835-1.60      Music           RetroPie-Setup_bak
bcm2835-1.60.tar.gz nrf            serial-servo-module
Desktop          NRF            sound-effect-module
Documents        oled          SunFounder_PiCar-V
Downloads        picar-4wd     Templates
```

1.6.2 Go to the Code Directory

```
cd ~/SunFounder_PiCar-V
```

Enter the code directory and you can see the installation script:

```
pi@raspberrypi:~ $ cd ~/SunFounder_PiCar-V
pi@raspberrypi:~/SunFounder_PiCar-V $ ls
ball_track  i2cHelper.py  LICENSE      remote_control
client      __init__.py   mjpg-streamer show
datasheet  install_dependencies README.md
pi@raspberrypi:~/SunFounder_PiCar-V $
```

1.6.3 Install the Environment via the Script

You can get all the required software and configuration done with the `install_dependencies` script. If you want to do step by step instead, please follow the instructions in the **Appendix 1: Function of the Server Installation Scripts**.

```
sudo ./install_dependencies
```

Note:

1. The installation script will install the required components and configure for the running environment. Make sure your Raspberry is connected to the Internet during the installation, or it would fail.
2. The Raspberry Pi will prompt you to reboot after the installation. You're recommended to type in **yes** to reboot.

1.6.4 Configure the Servo to 90 degrees

After reboot, run the `picar` tool:

```
cd ~/SunFounder_PiCar-V  
picar servo-install
```

Note: If the "OSError: [Errno 121] Remote I/O error" error message appears, open `raspi-config`:

```
sudo raspi-config
```

Then choose **3 Interfacing Options** → **P5 I2C** → **<YES>** → **OK** to enable I2C service. You can use the up, down, left, and right keys on the keyboard to select, and then press Enter to confirm.

After the code is running, insert the rocker arm into the servo. You will see the rocker arm is rotate in clockwise and counterclockwise, then stop at a specific location. It means the servo is good. If the any of the conditions below happened to your servo, your servo is bad:

- 1) Noisy, hot.
- 2) If unplug the servo line and rotate the rocker arm, it sounds like "ka" "ka" "ka" or there has no sounds of gear driving.
- 3) Rotate slowly but continuously.

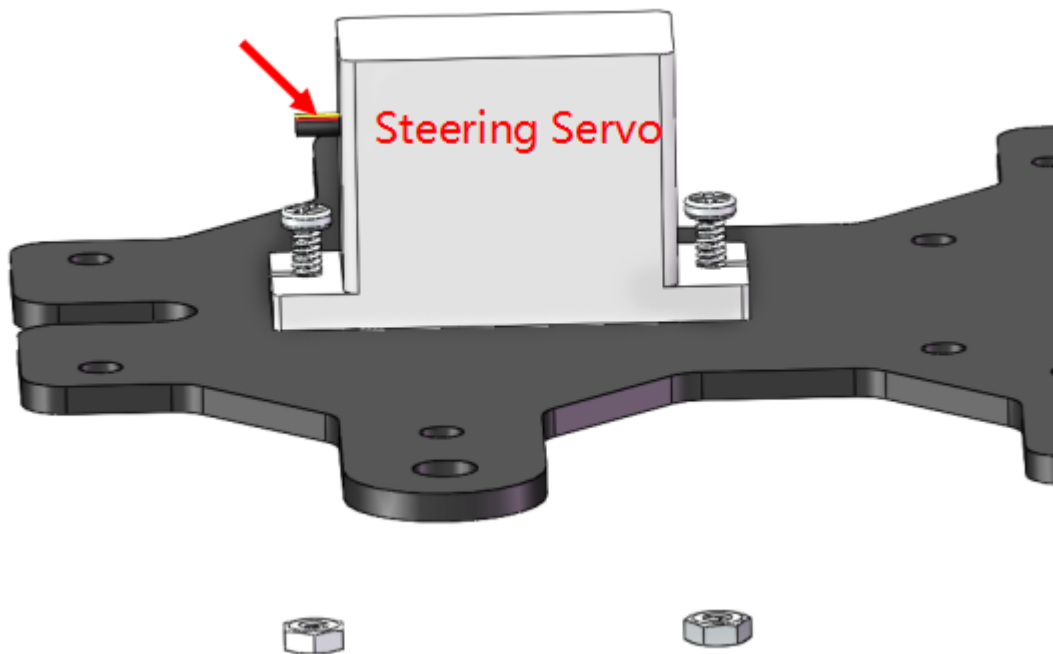
If you find one of the conditions above, please send e-mail to service@sunfounder.com. We will change a new one to you. If it is broken in the process of using or assembling, you should go to the official website www.sunfounder.com to buy.

1.7 Continue to Assemble

Warning: Please do forget to put in the battery and slide the power switch to ON, then keep servo-install running in the whole process of assembly.

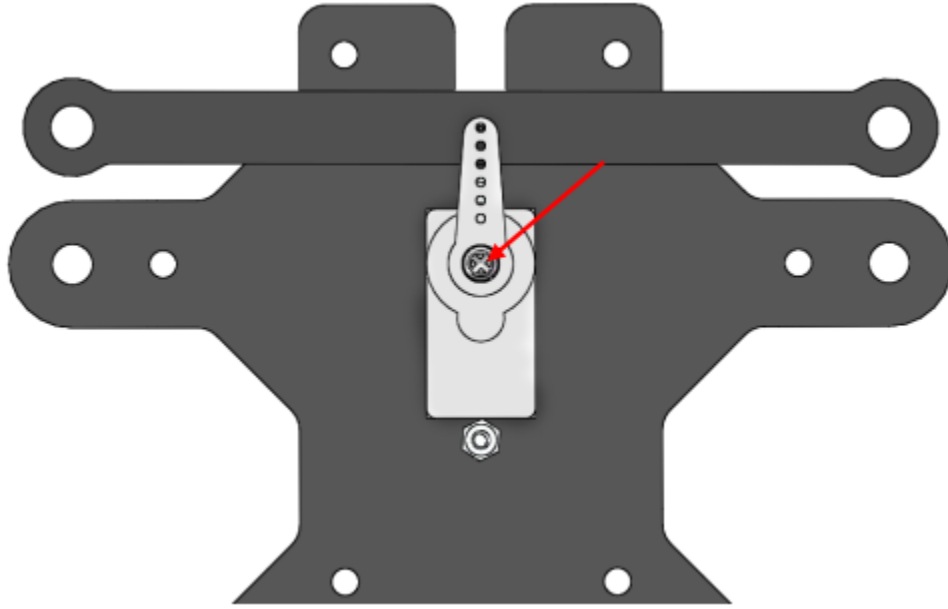
1.7.1 Assemble the Steering Servo

Mount the **Steering Servo** to the Upper Plate with two **M2x8 Screws** and **M2 nuts** (pay attention to the direction of the servo wires):

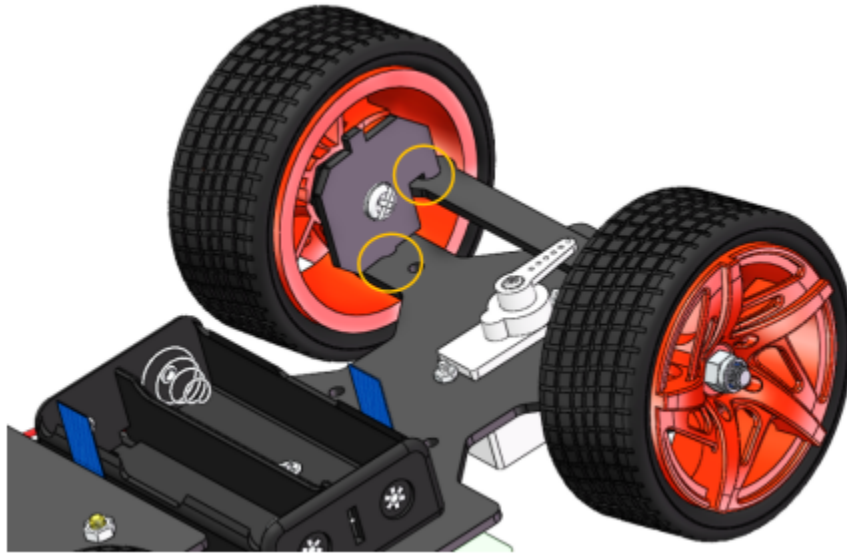


1.7.2 Front Half Chassis

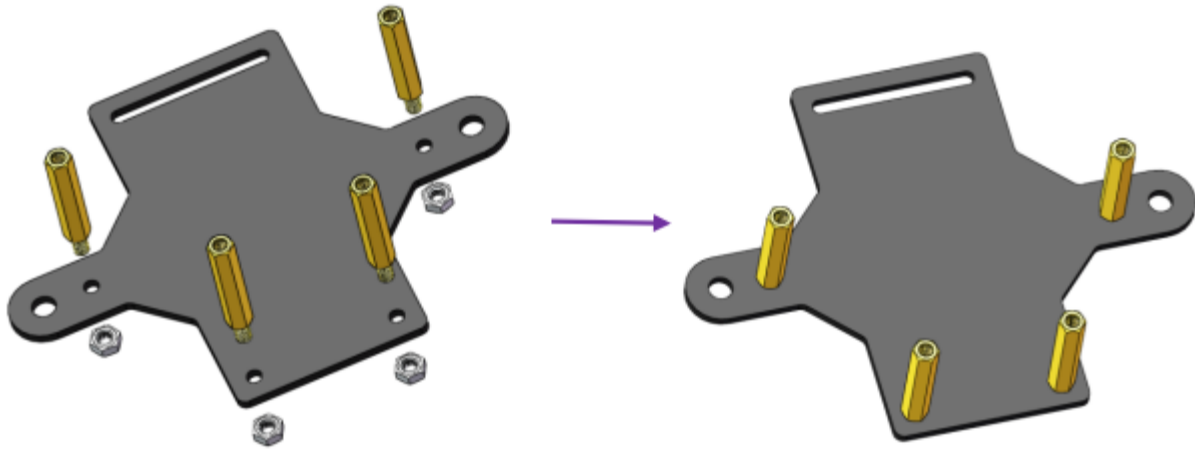
The **Rocker Arm** is facing straight ahead, and then inserted into the Servo shaft, and then fixed with **Rocker Arm Fixing Screw** (the shortest).



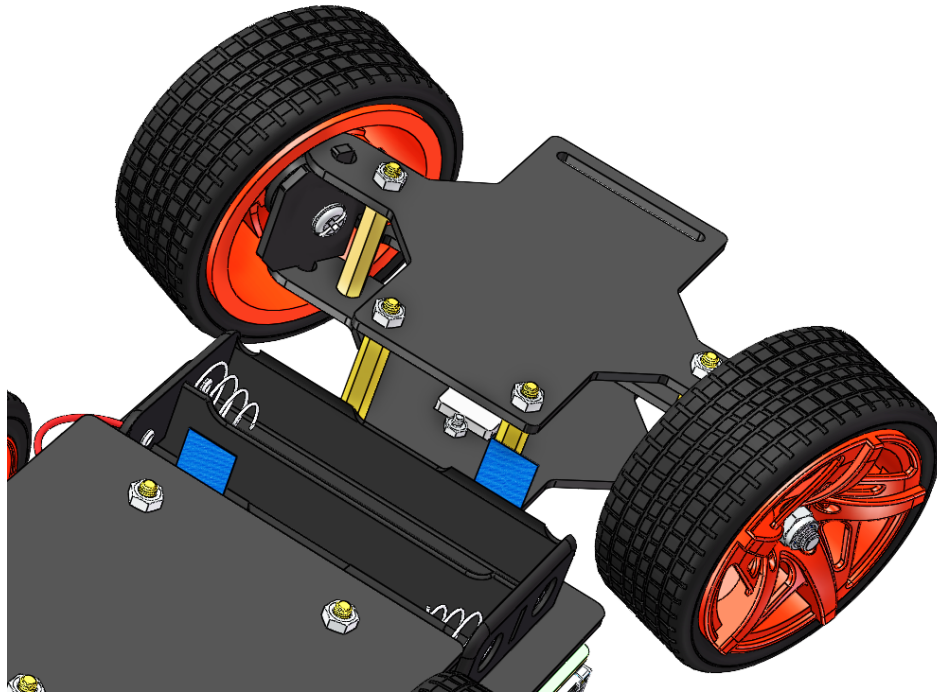
Mount the wheels onto the Upper Plate carefully.



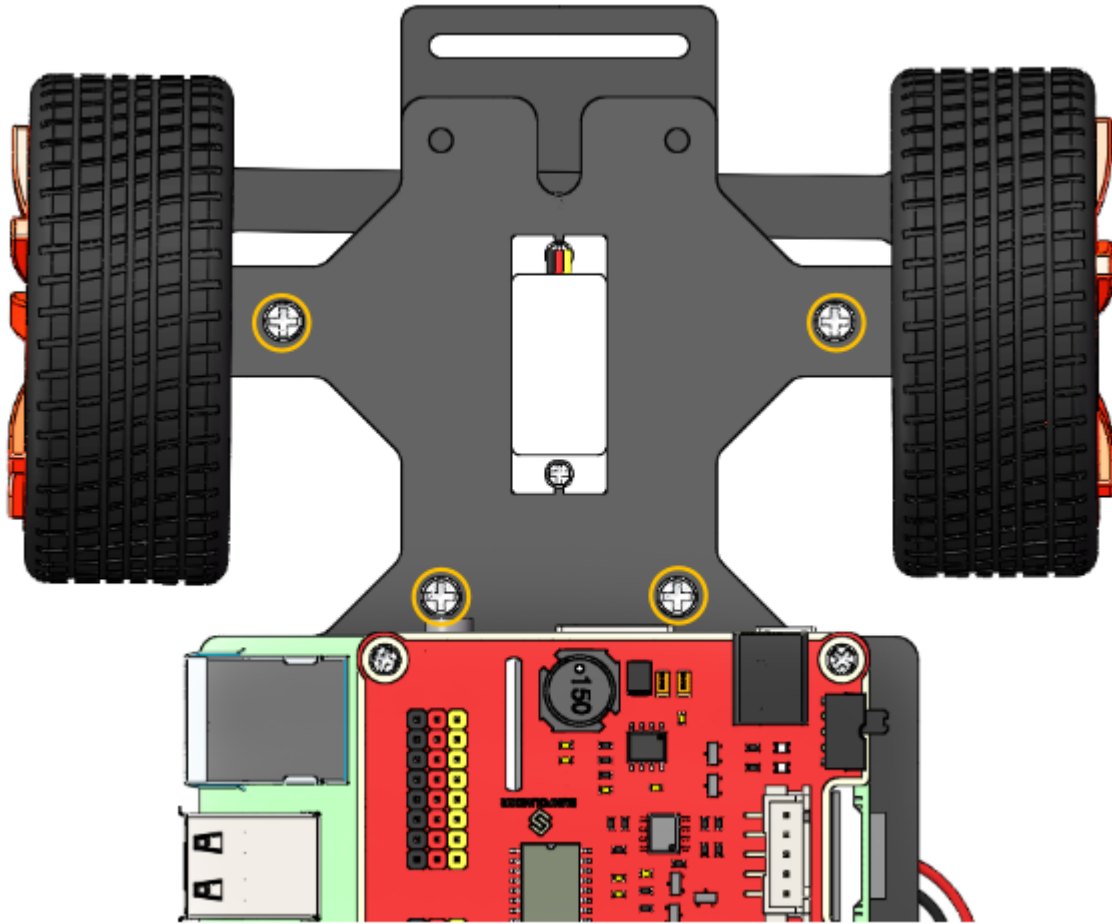
Assemble the **Front Half Chassis** with four **M3x25 copper standoffs** and **M3 nuts**.



Then put the assembled Front Half Chassis onto the Upper Plate with standoffs aligned with the holes.

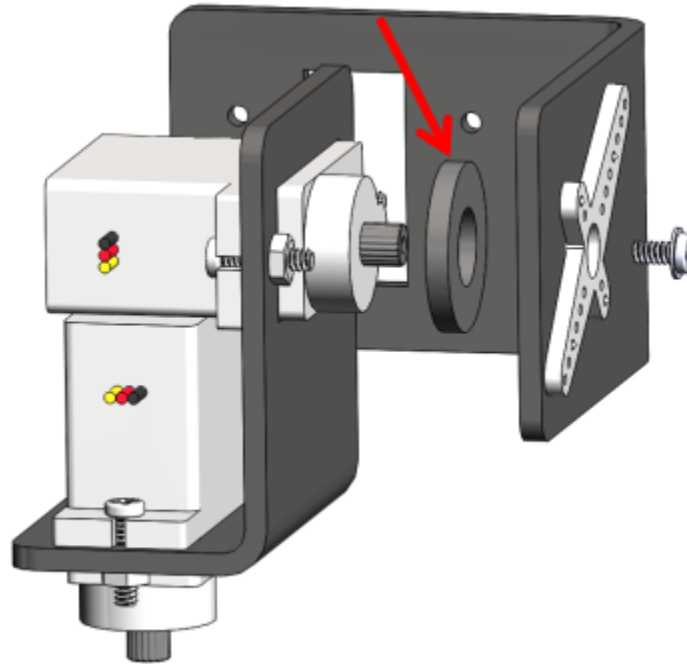


Hold them carefully, turn upside down, and fasten the standoffs and Upper Plate with four **M3x8 screws**.

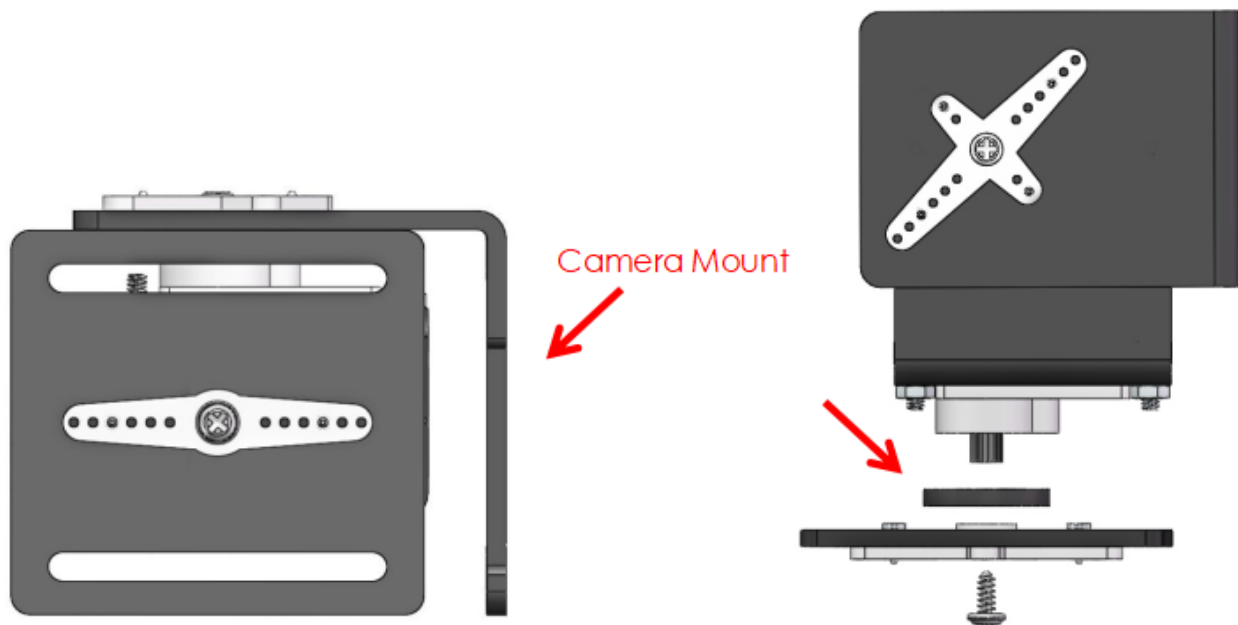


Assemble the assembled **Pan-and-tilt Plate**, **Gasket Plate** and **Camera Mount Plate** with the **Rocker Arm Fixing Screw** in a fixed angle as shown below:

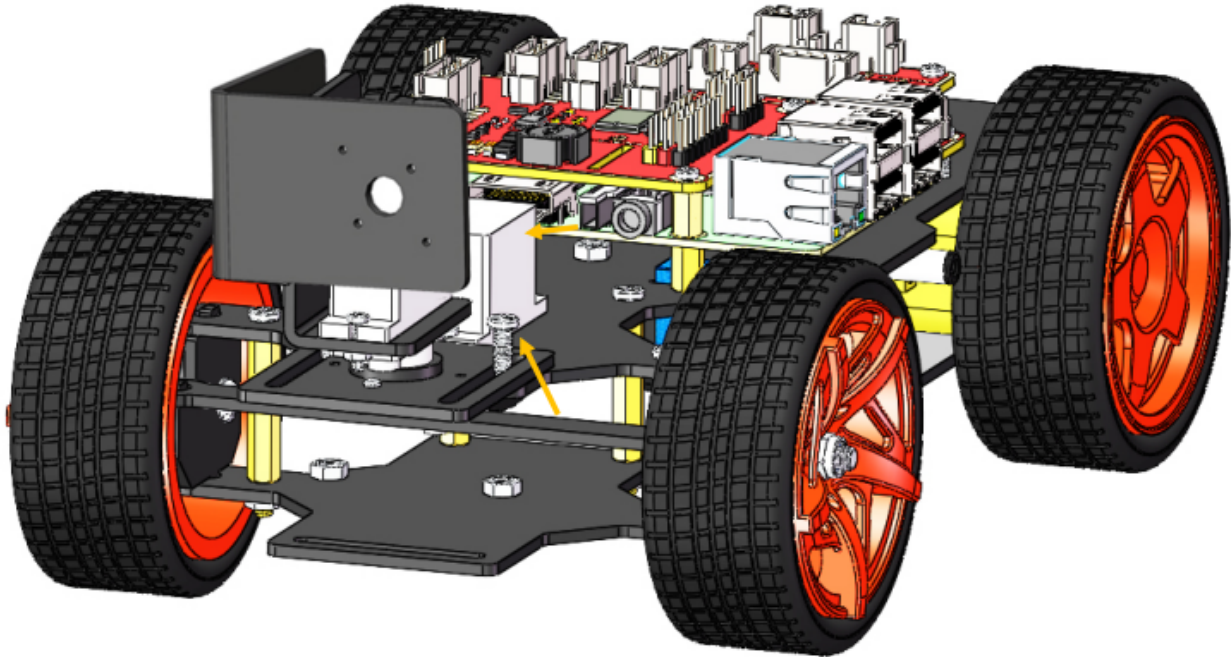
Note: Do not rotate the servo by hands in case of breaking the servodown.



Similarly check that the servo shaft has spun to 90 degrees. Then assemble the **Pan-and-tilt Base Plate** with the **Rocker Arm Fixing Screw** and the **Gasket Plate** in the angle as shown below.



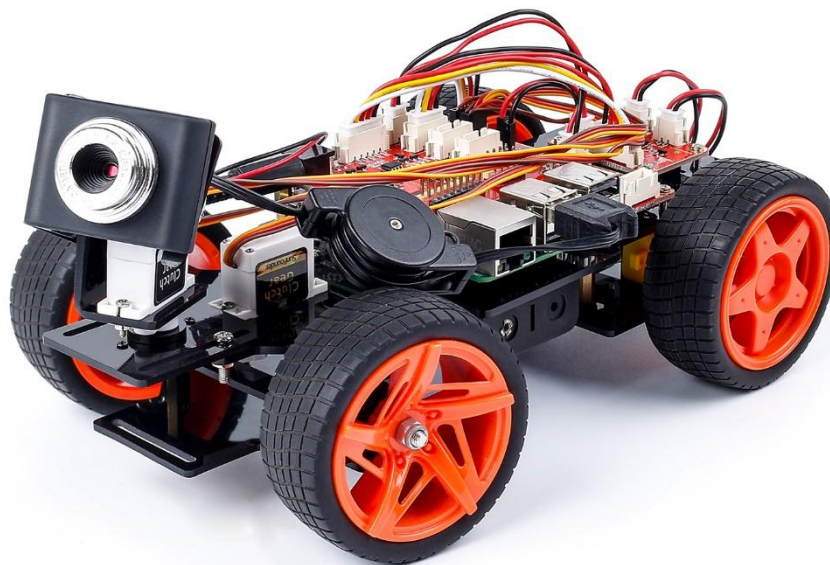
Assemble the **Pan-and-tilt Plate** to the car with two **M3x10 screws** and the **M3 nuts**.



1.7.3 Assemble the Camera

Take out the camera and nip it to the Camera Mount. Connect its USB cable to the USB port on the Raspberry Pi.

So now, the whole assembly is DONE! Congratulations! You can power off the car now! Don't forget to charge your batteries.



1.8 Getting on the Road!

Make sure all the procedures have been finished, and check there is no problem about the mechanical assembly and software installation.

1.8.1 Remote Control

Here is what we're going to do:

Use the Raspberry Pi as the **server**. Run a web server with an API for controlling the car and transmitting images captured by the camera.

Then take a PC, cell phone, or tablet as the **client**, and acquire the images from the camera and control the car by calling the API of the web server.

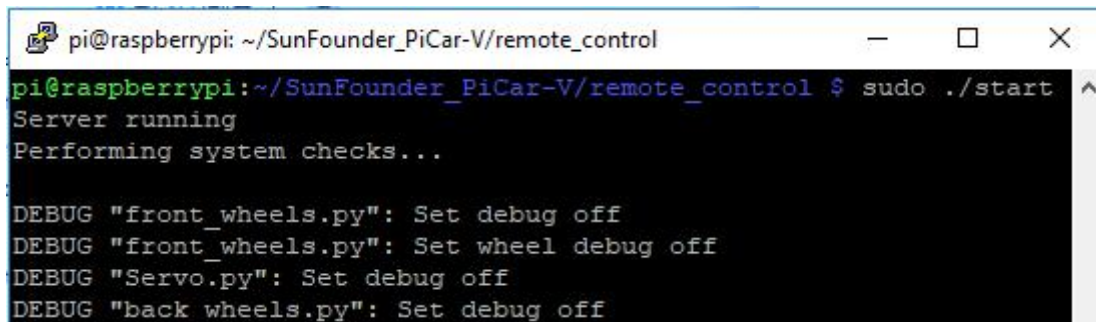
Now, you can power on the car again. You are also recommended to use the power adapter of Raspberry Pi to power your car for that the first test will take a long time.

Run the Server(Operation on Raspberry Pi)

Remotely log into the Raspberry Pi. Run the startup script **start** under the **remote_control** directory to start the web service.

```
cd ~/SunFounder_PiCar-V/remote_control
python3 manage.py migrate
sudo ./start
```

The script will enable the service and the corresponding data will appear. The hardware is initialized at the same time, so the servos connected to the front wheels and the pan-and-tilt will turn, indicating the hardware initialization is done.

A screenshot of a terminal window titled 'pi@raspberrypi: ~/SunFounder_PiCar-V/remote_control'. The terminal shows the command 'sudo ./start' being executed. The output is as follows:
pi@raspberrypi:~/SunFounder_PiCar-V/remote_control \$ sudo ./start
Server running
Performing system checks...

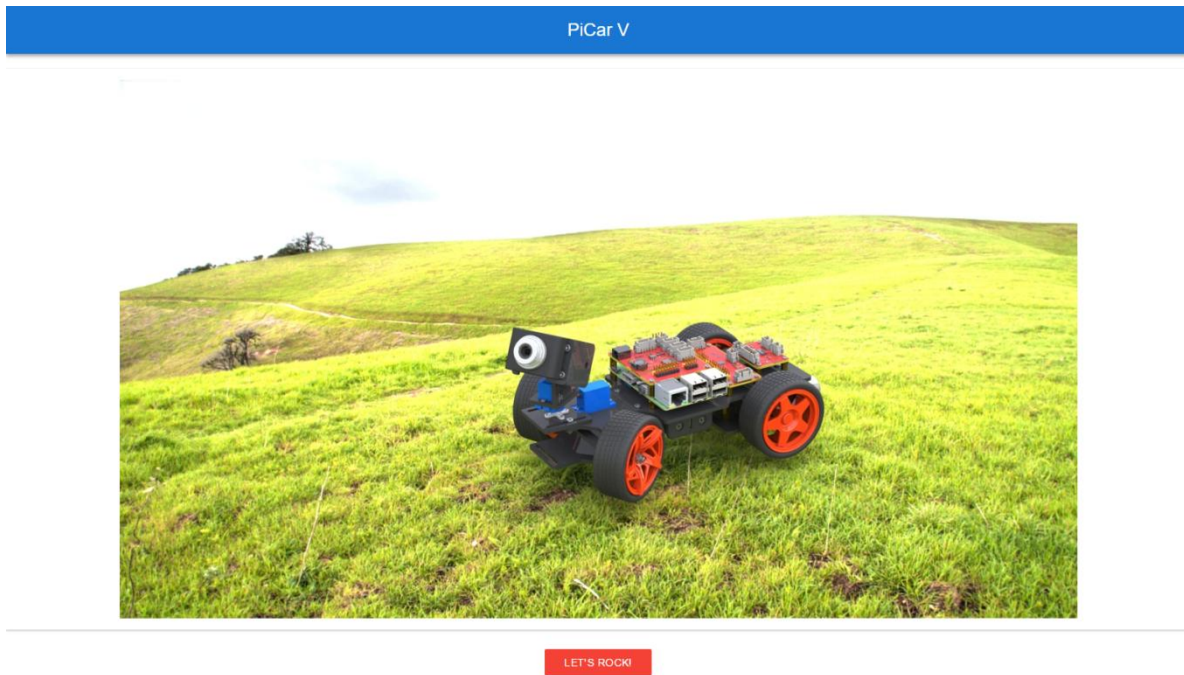
DEBUG "front_wheels.py": Set debug off
DEBUG "front_wheels.py": Set wheel debug off
DEBUG "Servo.py": Set debug off
DEBUG "back_wheels.py": Set debug off

If you get the result similar as shown above, the server is ready. Now move on to start the client.

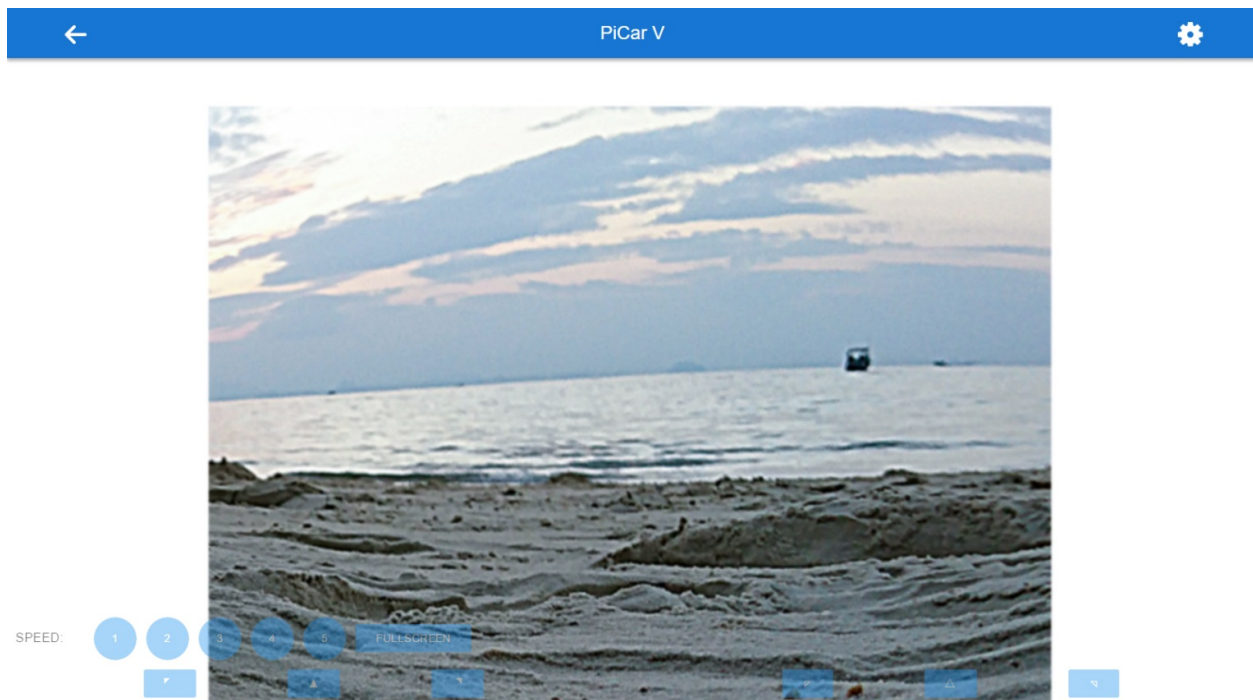
Warning: Keep the server being running all the time until you stop to run the Client.

Run the Client(Operation on PC)

Visit the server of the car at `http://<RPi_IP_address>:8000/`. You will see a welcome page:



Click LET'S ROCK to go to the operation interface:



On this page, you can press the keys **W**, **A**, **S**, and **D** on the keyboard to control the car to move **forward**, **backward**, **turn left**, and **turn right**, press the arrow keys to control the camera's movement, and number **1~5** to change the speed level.

Calibration

Click **FULLSCREEN** again to bring back the title bar. And then, tap the setting button at the top-right corner of the page to go to the calibration page:

← Calibration OK

Camera Calibration:

Front Wheels Calibration:

Rear Wheels Calibration:

There are three calibration parts: **Camera Calibration**, **Front Wheels Calibration** and **Rear Wheels Calibration**.

When you enter this page, the car will go forward; if not, click **Left** and **Right** in **Rear Wheels Calibration** to adjust the angle of the wheel.

Note: Every time you press a button, the angle will be changed slightly. For your larger change per time, you need to long press the button.

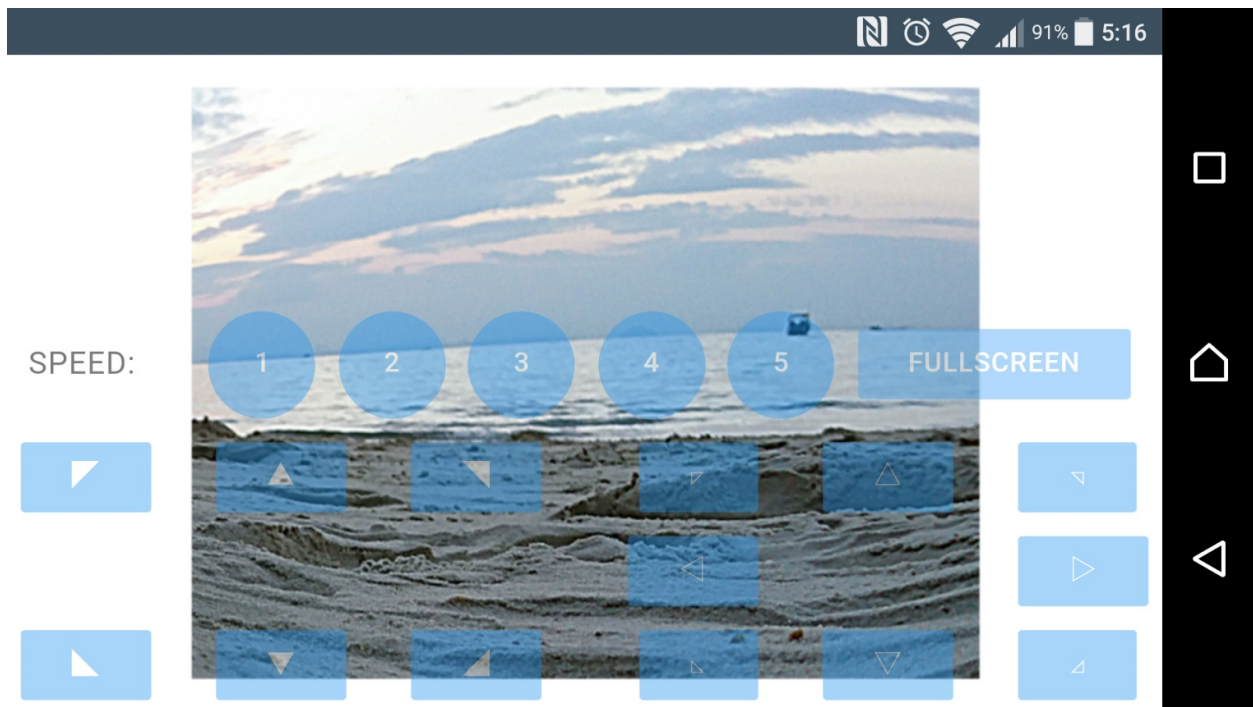
Click **OK** to save the result after all the calibration is done.

For Mobile Phones

Also for mobile phones, tap the **FULLSCREEN** button to have a better view and performance. Then, tap the buttons of 5 speed levels on the page to control the speed, and the arrow buttons to control the direction of the car and the pan-and-tilt. But you can only tab one touch point at **one** time.



Take this screenshot from an Android phone:



Though the appearance may not be as good as on the PC. Later updates may be released irregularly on **Github**. You are welcome to fork our repository and submit a Pull request with your changes. If there is no problem after testing, we are more than pleased to merge your request.

1.8.2 ball_tracker

In addition to using the keyboard to control the status of the car, we also wrote a ball tracking code.

Install OpenCV

Before installing OpenCV, you need to install the dependencies first. Different operating systems require different dependent libraries. (If you don't know your operating system version, you can try to install both.)

- If your operating system is **RASPBERRY PI BUSTER**, you need to install the following dependencies.

```
sudo apt-get install libatlas-base-dev libjasper-dev libqt4-test libwebp6 libtiff5
↪libopenexr23 libgstreamer1.0-0 libavcodec-dev libavformat-dev libswscale-dev libqtgui4
↪-y
```

- If your operating system is **RASPBERRY PI BULLSEYE**, you need to install the following dependencies.

```
sudo apt-get install libatlas-base-dev libjasper-dev libwebp6 libtiff5 libopenexr25
↪libgstreamer1.0-0 libavcodec-dev libavformat-dev libswscale-dev -y
```

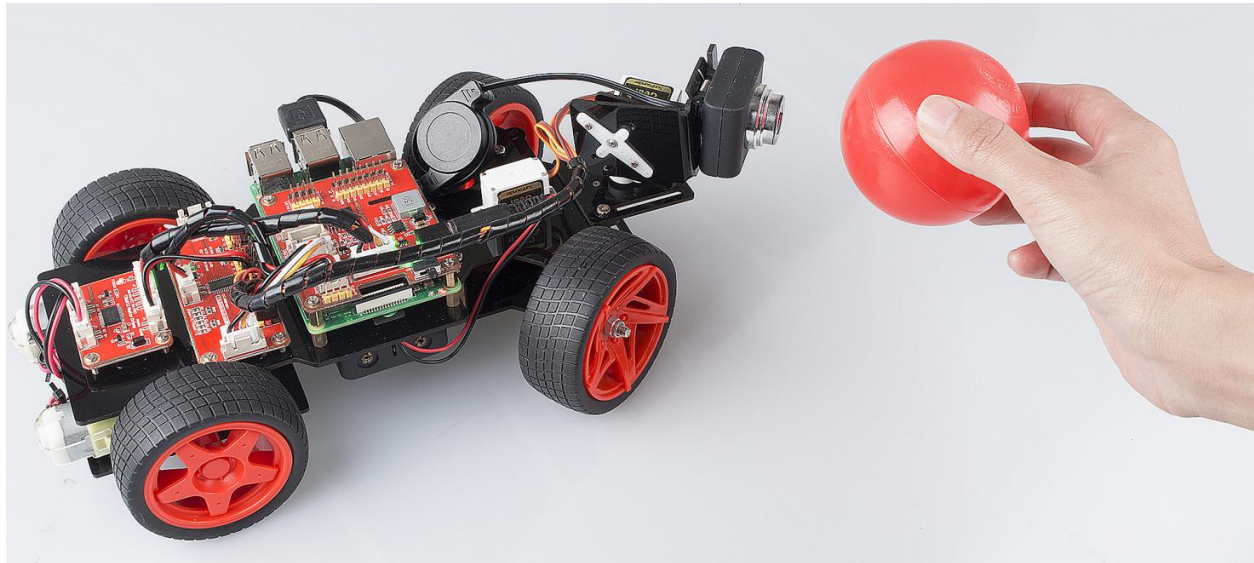
Next is to install OpenCV.

```
sudo pip3 install opencv-python==3.4.6.27
```

Run the Code

```
cd ~/SunFounder_PiCar-V/ball_track python3 ball_tracker.py
```

After the code runs, find a red ball (at least larger than the diameter of the camera) and put it within 10-50cm range in front of the camera to get the car to follow your ball.

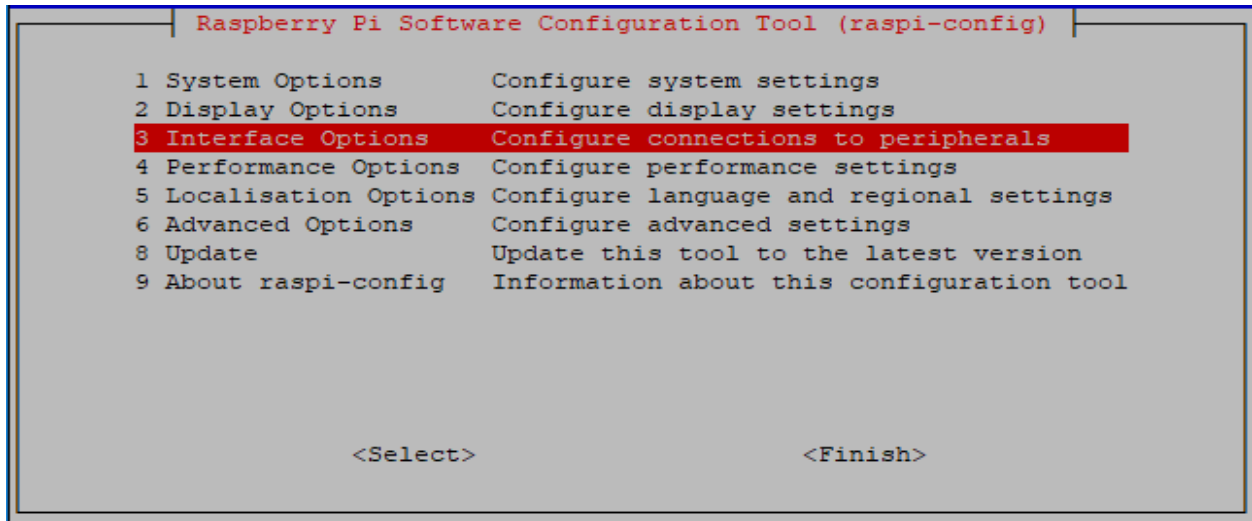


If you want to know whether the car has caught your ball, you can do as follows:

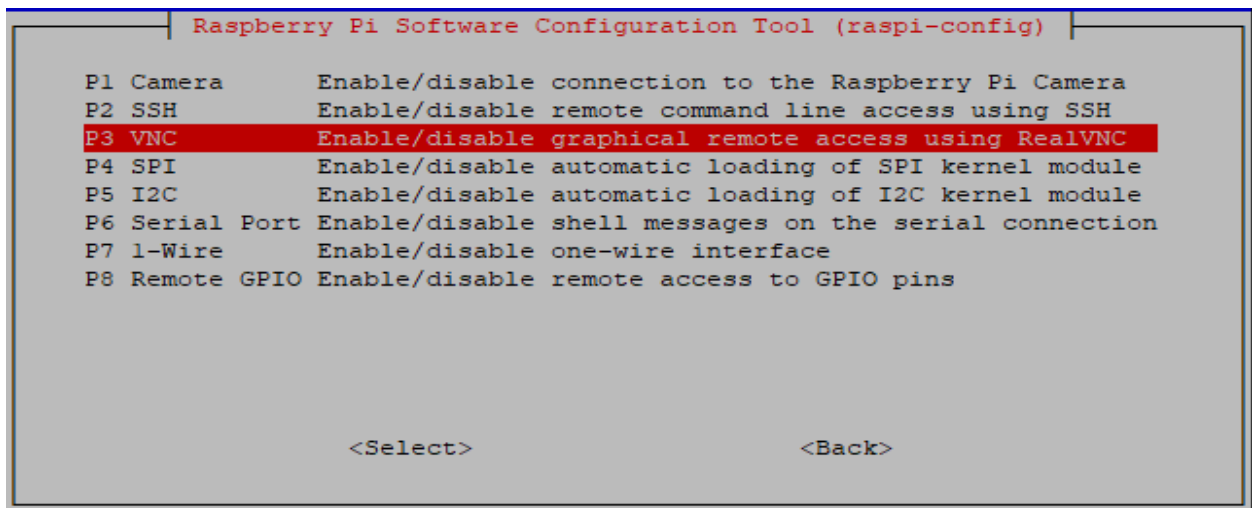
Step 1: Enable the VNC on the Raspberry Pi.

```
sudo raspi-config
```

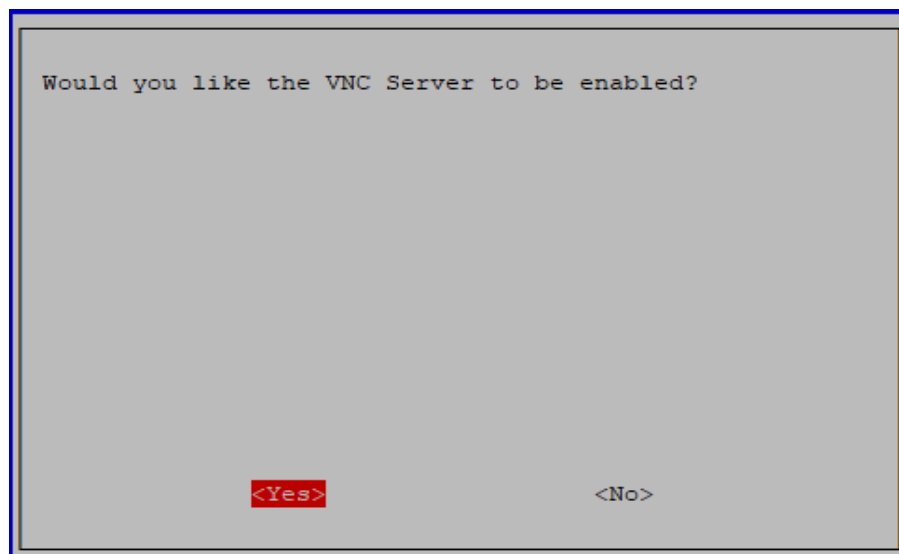
Choose **3 Interfacing Options** by press the down arrow key on your keyboard, then press the **Enter** key..



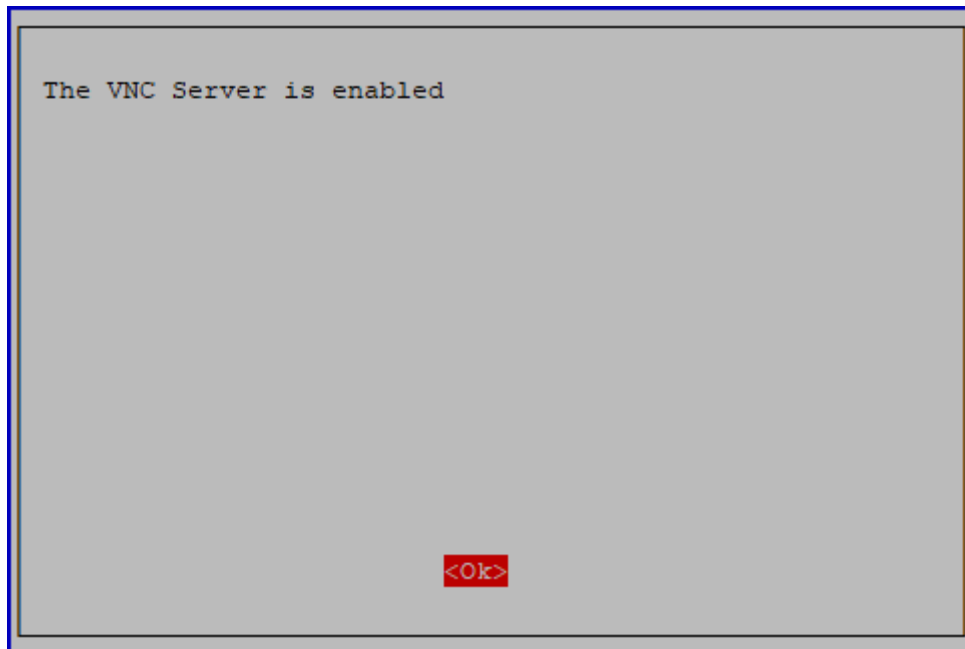
P3 VNC



Press the left arrow key on your keyboard to choose **Yes**.



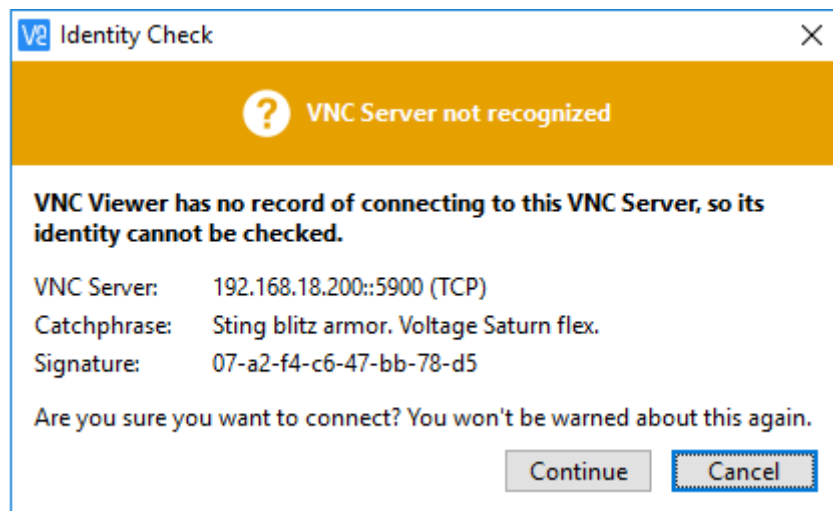
Finally select **OK->Finish** to exit the config.



Step 2: Download the VNC Viewer for your computer and install it.

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

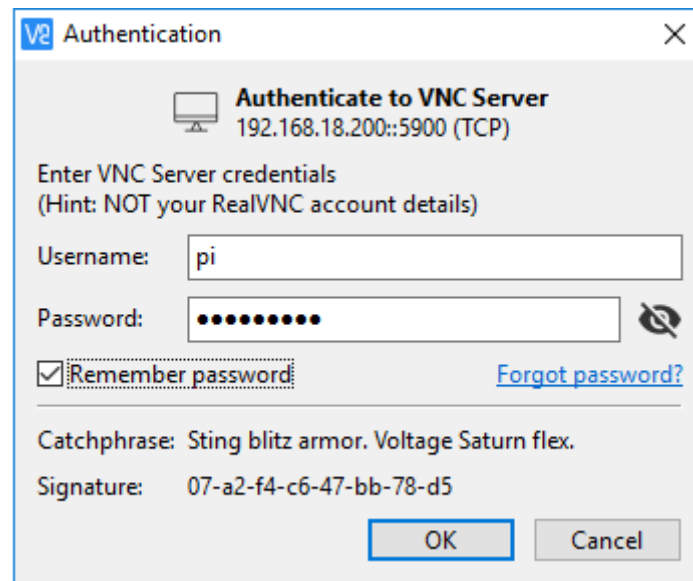
Note: During your installation on your Windows, if you encounter the following prompt, please click **Continue** to continue your installation.



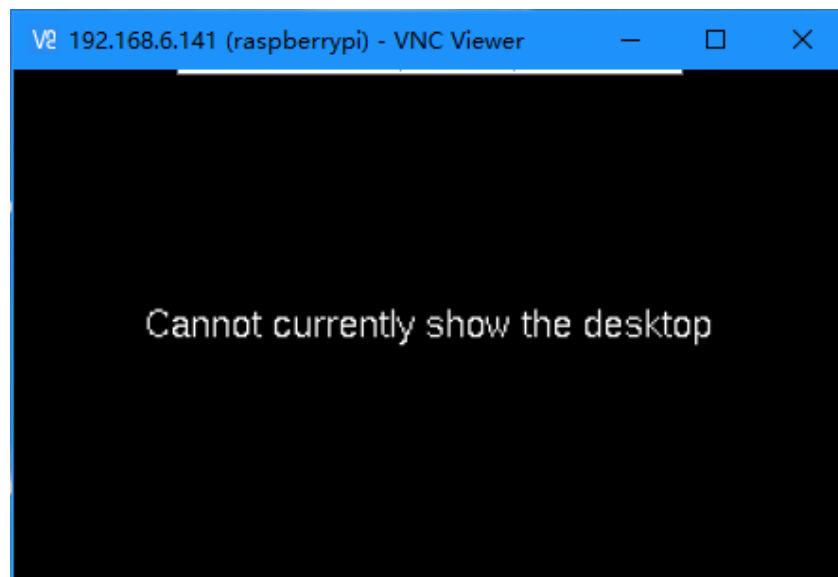
Step 3: Open VNC Viewer you have installed, type in your Raspberry Pi IP address, and press the Enter button.



Step 4: Enter the username and password of your Raspberry Pi on this page, by default, they are pi and raspberry; then click OK.



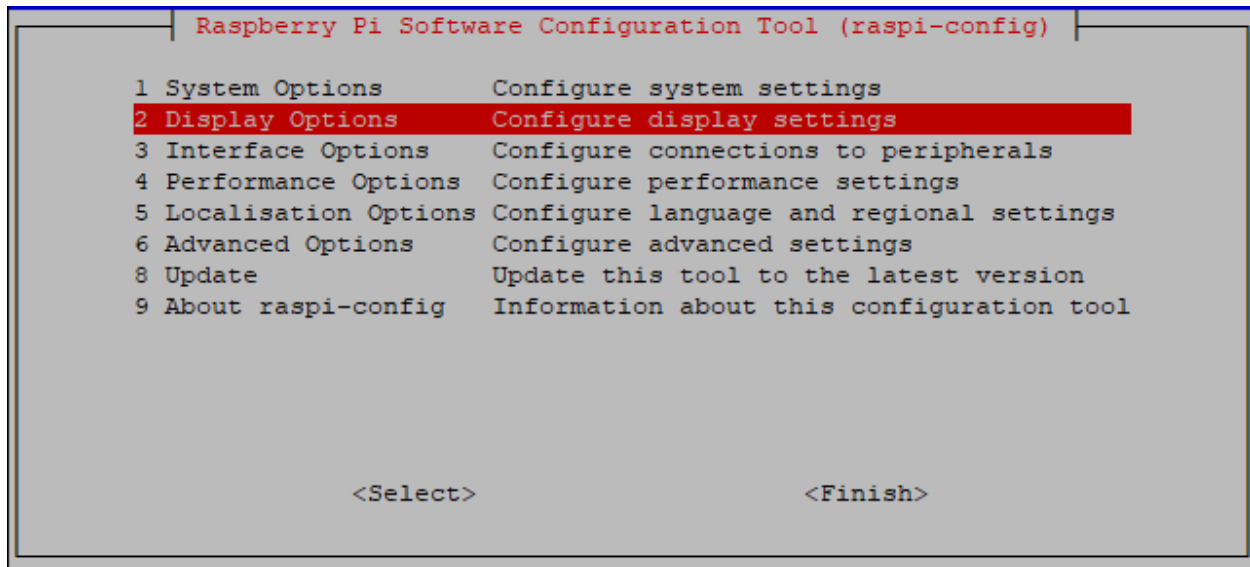
When you log in with VNC Viewer, the screen may be black.If your screen is working, skip to Step 5.



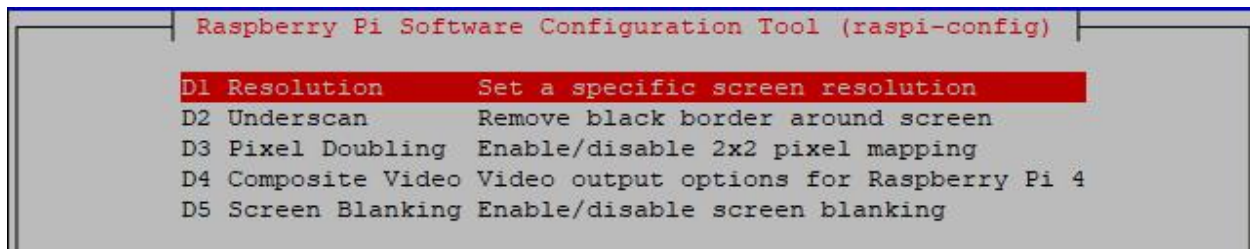
Changing the resolution to the highest on the Raspberry Pi.

```
sudo raspi-config
```

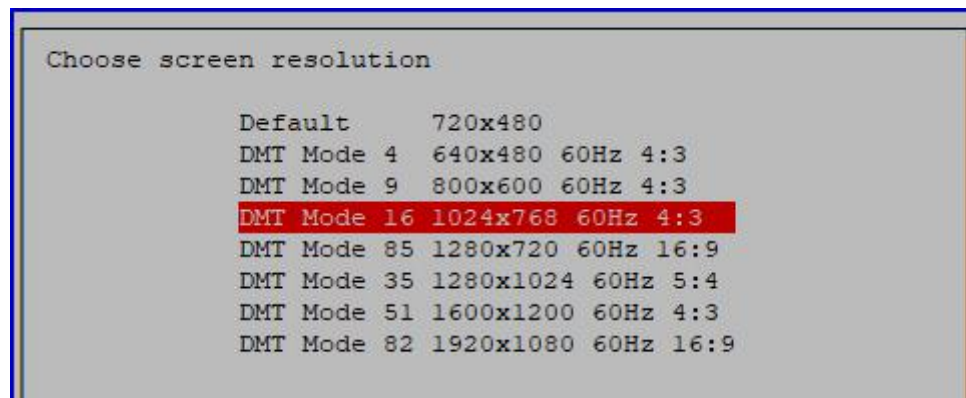
Choose **2 Display Options**.



Choose **D1 Resolution**.



Change the default to a higher resolution like **Mode 16**, then click **OK**.



Reboot the Raspberry Pi, log in again, and it will work.

```
sudo reboot
```

Step 5: Now, we enter the Raspberry Pi desktop, click Terminal and type the following commands to open the code ball_tracker.py.

```
cd ~/SunFounder_PiCar-V/ball_track
sudo nano ball_tracker.py
```

Then modify the code as follows:

```
Show_image_enable = True
Draw_circle_enble = True
```

Press Ctrl+X and Y to save the change.

The screenshot shows a VNC Viewer window titled '192.168.18.200 (raspberrypi) - VNC Viewer'. The terminal window displays the nano 3.2 text editor editing the file 'ball_tracker.py'. The code in the editor is as follows:

```
from picar import front_wheels, back_wheels
from picar.SunFounder_PCA9685 import Servo
import picar
from time import sleep
import cv2
import numpy as np
import picar
import os

picar.setup()
# Show image captured by camera, True to turn on, youwill need #DISPLAY and it$
show_image_enable = True
draw_circle_enable = True
scan_enable = False
```

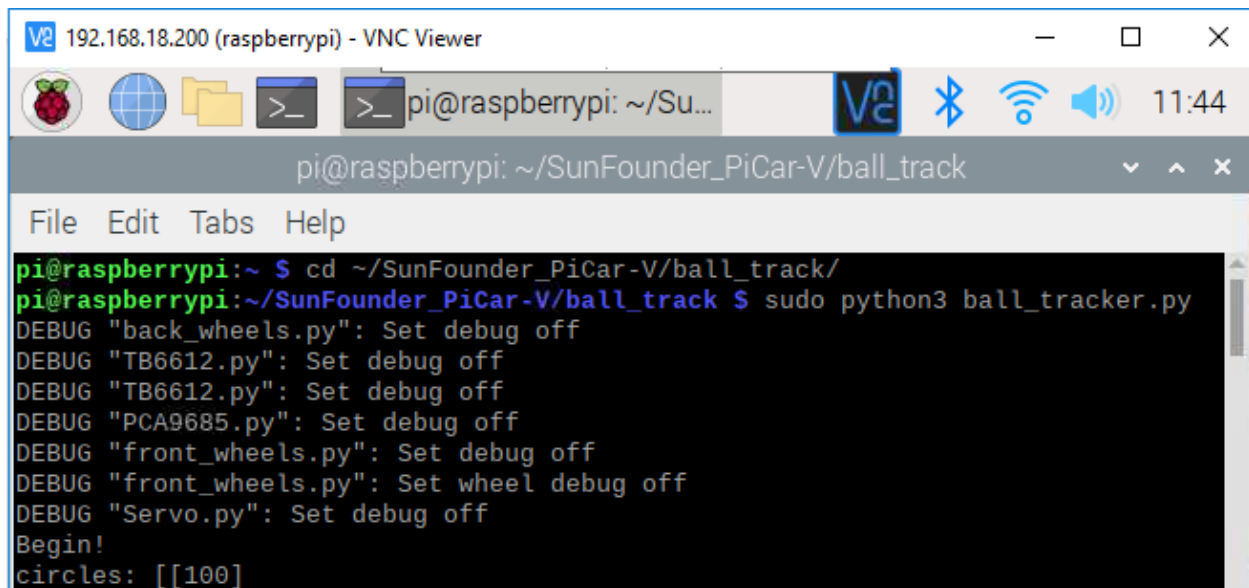
The values 'True' for 'show_image_enable' and 'draw_circle_enable' are highlighted with red boxes. The bottom of the nano editor shows keyboard shortcuts: ^G Get Help, ^O Write Out, ^W Where Is, ^K Cut Text, ^J Justify, ^X Exit, ^R Read File, ^\ Replace, ^U Uncut Text, ^T To Spell.

Note: After you finish doing this step, you must log in the Raspberry Pi via VNC, or you can use a monitor; if not, the warning is as follows:

```
pi@raspberrypi:~/SunFounder_PiCar-V/ball_track $ python3 ball_tracker.py
Warning: Display not found, turn off "show_image_enable" and "draw_circle_enable"
DEBUG "back_wheels.py": Set debug off
```

Step 6: Run the ball_tracker.py.

```
python3 ball_tracker.py
```

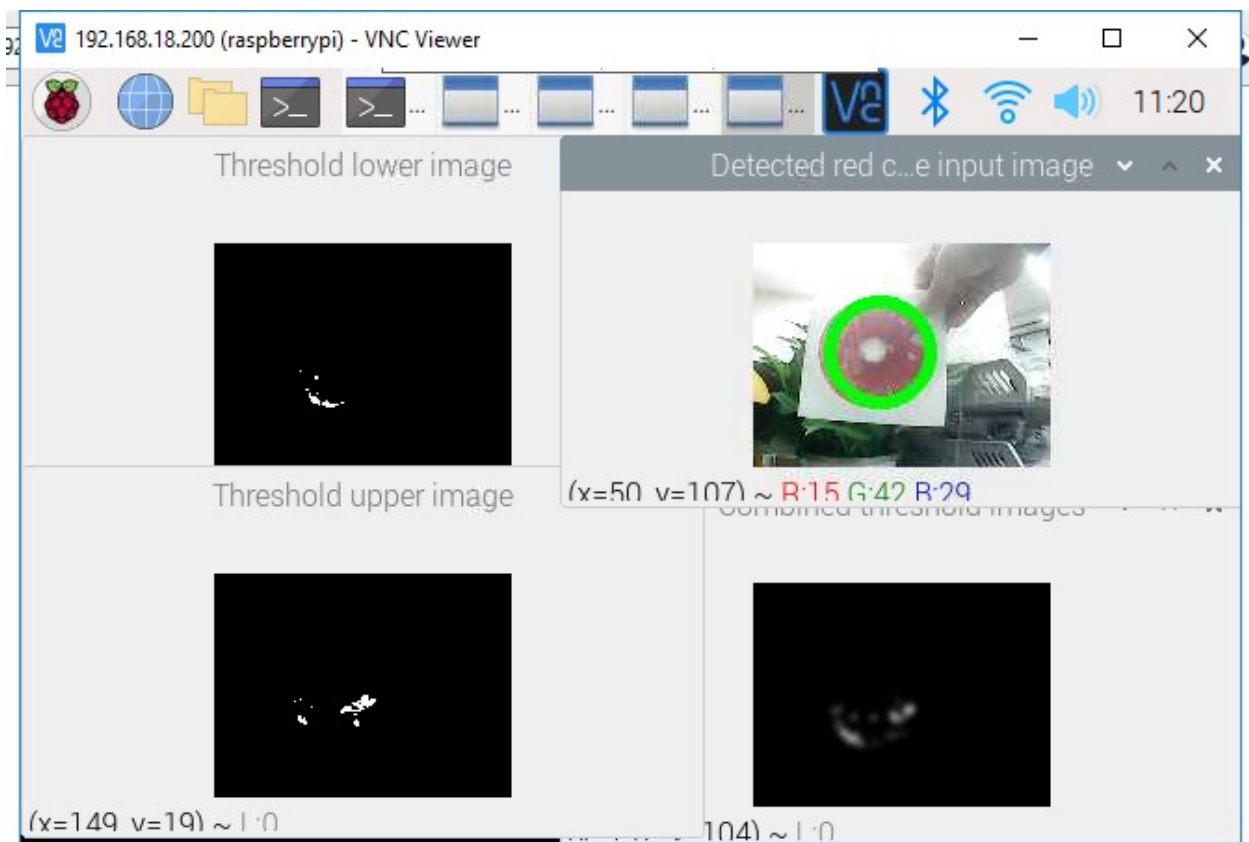


```

192.168.18.200 (raspberrypi) - VNC Viewer
pi@raspberrypi: ~/SunFounder_PiCar-V/ball_track
File Edit Tabs Help
pi@raspberrypi:~ $ cd ~/SunFounder_PiCar-V/ball_track/
pi@raspberrypi:~/SunFounder_PiCar-V/ball_track $ sudo python3 ball_tracker.py
DEBUG "back_wheels.py": Set debug off
DEBUG "TB6612.py": Set debug off
DEBUG "TB6612.py": Set debug off
DEBUG "PCA9685.py": Set debug off
DEBUG "front_wheels.py": Set debug off
DEBUG "front_wheels.py": Set wheel debug off
DEBUG "Servo.py": Set debug off
Begin!
circles: [[100]

```

Step 7: After running the code `ball_tracker.py`, these four images will appear. If you place a red ball in front of the camera, you can see a green circle which represents that the car has caught your red ball.

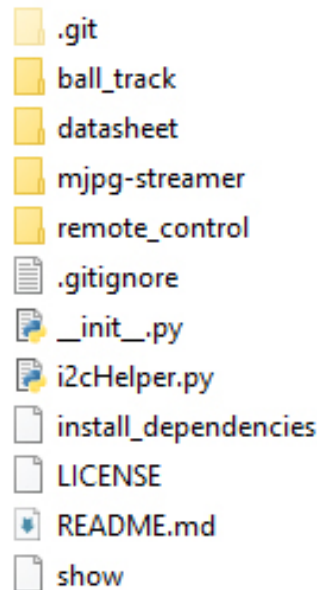


1.8.3 File Analysis

There are two folders under /home/pi: **SunFounder_PiCar** for controlling motors and steering and **SunFounder_PiCar-V** for pan/tilt and wireless control. Here look into the code of PiCar-V.

Since too many contents and instructions are involved in the code, we will not cover every detail. For more knowledge about Python 2, Python 3, Django, HTML, and CSS in the programs, you can visit related websites or buy books to learn by yourself. In the following part, we will go through the overall structure and the process in brief.

First, let's check the files in the code folder:



- `.git` is a hidden directory to store all the information of a Git repository; it's generated automatically after you create the repository.
- `Ball_track` is used to let the car follow the red ball.
- `datasheet` stores the manual of some chips used in the hardware modules.
- `mjpg-streamer` is an open source streaming media library, through which the data of the camera is transferred.
- `Remote_control` is to store the code of the web server; it controls the motor and servo of the car based on the API request.
- `.gitignore` records the requests of file types to be ignored when the Github repository is synchronized.
- `__init__.py` is automatically generated when you create a project which is a standard necessary document of python package, just leave it there.
- `i2cHelper.py` is a Python script written by Python 2 to configure and detect the I2C connection.
- `install_dependencies`, an executable bash script for simple installation and environment configuration.
- `LICENSE`, as the name suggests, is a text file of GNU V2 license.
- `README.md` and `show` record some information normally for statement and prompts.

Server Code

The server code is based on Django 1.10 (adapt to the latest release if needed). If you are interested in this, you can visit the Django website <https://www.djangoproject.com/>. Here we will just learn how the web server works.

Note: The code may be updated irregularly on Github to fix bugs and release some functions update. So the code here is the initial version. You can view the updates in the Github repository at:

https://github.com/sunfounder/SunFounder_PiCar-V/tree/V3.0

Open the code directory and check the file by `ls`:

```
pi@raspberrypi:~$ cd SunFounder_PiCar-V/
pi@raspberrypi:~/SunFounder_PiCar-V$ ls
datasheet  install_dependencies  mjpg-streamer  remote_control
client      i2cHelper.py          LICENSE         README.md show
```

`remote_control` is the main code directory of the web server. Open the code directory by `cd remote_control`, and check the file by `ls`:

```
pi@raspberrypi:~/SunFounder_PiCar-V$ cd remote_control
pi@raspberrypi:~/SunFounder_PiCar-V/remote_control$ ls
db.sqlite3  manage.py  remote_control  start  static
```

- `db.sqlite3` is generated when you create a Django project.
- `manage.py` is the main program of the Django project and is generated when the project is created. It normally does not need to be modified.
- `remote_control` includes the main code files.
- `start` is a small script written to run `sudo python manage.py runserver 0.0.0.0:8000`, and servo installation with attribute `install` just for convenience.
- `static` is to store some static pictures on the web.

The Django web server normally runs `sudo python manage.py runserver` to start. The address `0.0.0.0:8000` means the listening address covers all the addresses on the LAN, and the port number is 8000. Here we will just focus on the code in `remote_control` folder. Go to the directory via `cd remote_control`:

```
pi@raspberrypi:~/SunFounder_PiCar-V/remote_control$ cd remote_control
pi@raspberrypi:~/SunFounder_PiCar-V/remote_control/remote_control$ ls
driver  __init__.py  settings.py  templates  urls.py  views.py  wsgi.py
```

- `driver` stores the drivers of the car.
- `__init__.py` is automatically generated when you create a Django project which is a standard necessary document of python package, just leave it there.
- `settings.py` is automatically generated and stores the related settings.
- `templates` is a Django app for storing the webs in the html format.
- `urls.py` is generated automatically to configure the URL to associate with the code.
- `views.py` is the code for page control which is associated by the URL. It calls the templates to show the page and the driver to control the car.
- `wsgi.py` is generated automatically and does not need changes. For more, please visit the official website of Django.

So here is how the code works: Run the main program `manage.py` which will be associated with `urls.py` automatically, to respond to the URL. When you run the web browser like the Chrome to visit `http://<rpi_ip_address>:<port>` or visit the configured API via the client, the `manage.py` will turn to `views.py` due to the association of the `urls.py`. Then the `views.py` processes this and returns the templates to the browser. In addition, it will call the driver package on the basis of the parameters set in the browser to control the car.

Now open the folder `driver` and check:

```
pi@raspberrypi:~/SunFounder_PiCar-V/remote_control/remote_control $ cd driver/
pi@raspberrypi:~/SunFounder_PiCar-V/remote_control/remote_control/driver $ ls
camera.py  config  __init__.py  stream.py
```

The driver folder mainly includes the driver modules for controlling the pan and tilt and camera streamer.

`camera.py` for controlling the pan-and-tilt.

`config` stores the calibration data.

`__init__.py` is the essential file of the package and you can just leave it alone.

`stream.py` is a video streaming service based on the MJPG-streamer.

Exit and open the folder `templates` to view:

```
pi@raspberrypi:~/SunFounder_PiCar-V/remote_control/remote_control/driver $ cd ../
pi@raspberrypi:~/SunFounder_PiCar-V/remote_control/remote_control $ cd templates/
admin.py  __init__.py  models.py  tests.py
apps.py  migrations  templates  views.py
```

This folder is created by the `manage.py startapp` just for calling the templates conveniently. Therefore, the files have not been changed except for the templates. Open the templates again:

```
pi@raspberrypi:~/SunFounder_PiCar-V/remote_control/remote_control/templates $ cd templates
pi@raspberrypi:~/SunFounder_PiCar-V/remote_control/remote_control/templates/templates $ ls
base.html  cali.html  run.html
```

There are three HTML files that also consist of layers. The low level `base.html` stores contents contained on each page such as the `<head>` of HTML, the overall layout, and contents of the home page by default. The surface layer: `cali.html` for calibration and `run.html` to control the car.

1.8.4 Advanced

Now the car is running! You may try control it or view the image captured in Windows, Linux, or Mac OS X.

So what to do next?

Yes, sensors! There are 8 digital and 4 analog channels on the Robot HATS, as well as two I2C ports. Plus the 3mm slot on the acrylic plate in the car head, it enables the connection of various sensors, making the car smarter and more intriguing as you want!

And, coding! If you know programming and are capable of compiling code, you may change the code provided for the kit to make the car more fabulous. For instance, you can find the code that controls the front and back wheels in the folder `SunFounder_PiCar`, downloaded to the directory `/home/pi` if you've installed the program by `install-dependences`.

You may check what's in the file:

```
cd ~/SunFounder_PiCar
ls
```

After you've changed the code, you can go to the page of the Github repository for the car, and fork it to help improve it! Also you're quite welcome to post issues and request a pull on our Github page:

PiCar-V: https://github.com/sunfounder/SunFounder_PiCar-V/tree/V3.0

PiCar driver: https://github.com/sunfounder/SunFounder_PiCar

And of course, if you have any questions or ideas, please feel free to email us at service@sunfounder.com.

1.9 Appendix

1.9.1 Appendix 1: Function of the Server Installation Scripts

You may have a question: What do the installation scripts do when we install the server on the Raspberry Pi by them? So here let's check the detailed steps.

1. Install pip.

```
sudo apt-get install python-pip
```

2. Use pip to install django.

```
sudo pip install django
```

3. Install i2c-tools and python-smbus.

```
sudo pip3 install smbus2
```

4. Install PiCar drivers.

```
cd ~/
git clone --recursive https://github.com/sunfounder/SunFounder_PiCar.git
cd SunFounder_PiCar
sudo python setup.py install
```

5. Download source code.

```
cd ~/
git clone https://github.com/sunfounder/SunFounder_PiCar-V -b V3.0
```

6. Copy the MJPG-Streamer file to system directory.

```
cd ~/SunFounder_PiCar-V
sudo cp mjpg-streamer/mjpg_streamer /usr/local/bin
sudo cp mjpg-streamer/output_http.so /usr/local/lib/
sudo cp mjpg-streamer/input_file.so /usr/local/lib/
sudo cp mjpg-streamer/input_uvc.so /usr/local/lib/
sudo cp -R mjpg-streamer/www /usr/local/www
```

7. Export paths.

```
export LD_LIBRARY_PATH=/usr/local/lib/ >> ~/.bashrc
export LD_LIBRARY_PATH=/usr/local/lib/ >> ~/.profile
source ~/.bashrc
```


8. Enable I2C1.

Edit the file /boot/config.txt:

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

Add the line in the end:

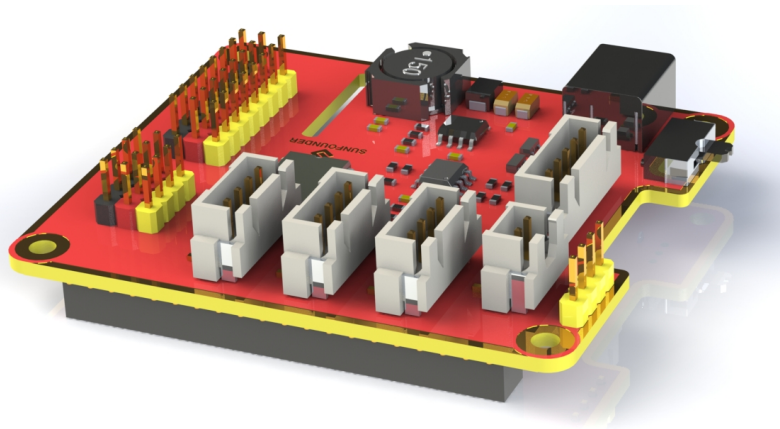
```
dtoverlay=i2c-arms
```

9. Reboot.

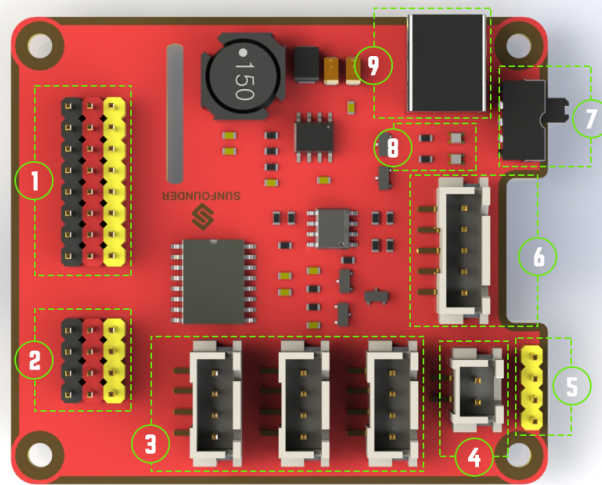
```
sudo reboot
```

1.9.2 Appendix 2: Components

Robot HATS

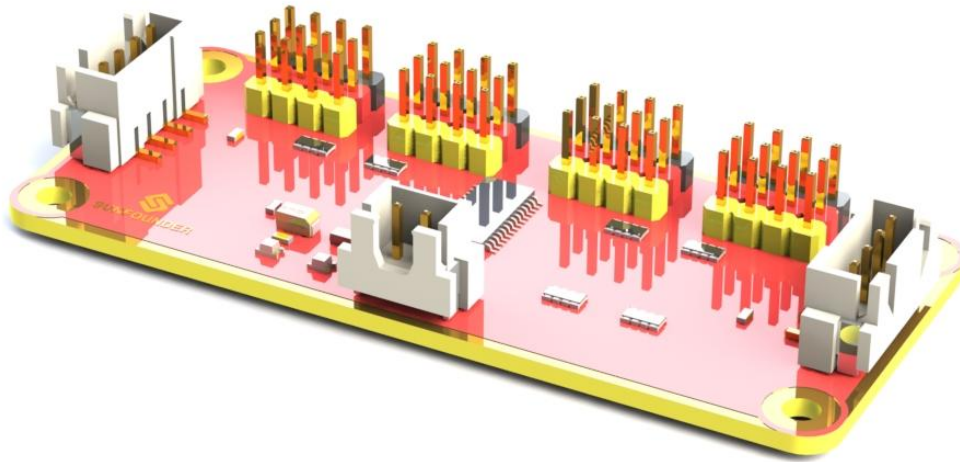


Robot HATS is a specially-designed HAT for a 40-pin Raspberry Pi and can work with Raspberry Pi model B+, 2 model B, 3 model B, 3 model B+, and 4 model B. It supplies power to the Raspberry Pi from the GPIO ports. Thanks to the design of the ideal diode based on the rules of HATS, it can supply the Raspberry Pi via both the USB cable and the DC port thus protecting it from damaging the TF card caused by batteries running out of power. The PCF8591 is used as the ADC chip, with I2C communication, and the address 0x48.

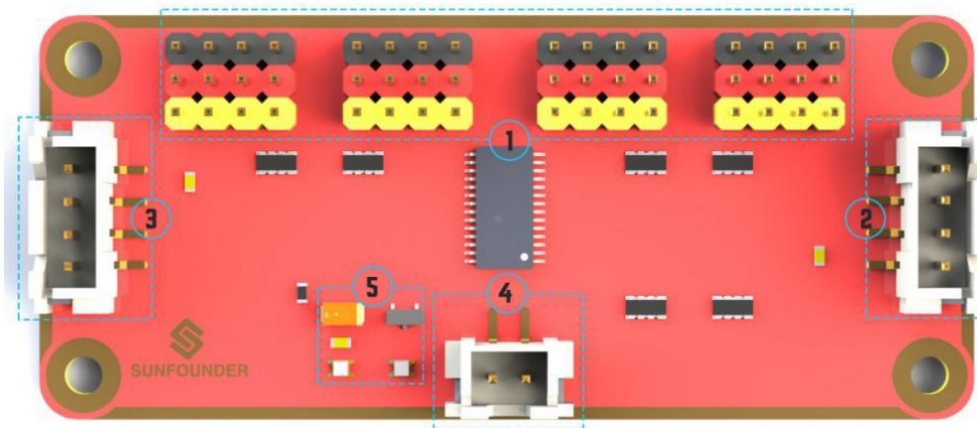


1. **Digital ports:** 3-wire digital sensor ports, signal voltage: 3.3V, VCC voltage: 3.3V.
2. **Analog ports:** 3-wire 4-channel 8-bit ADC sensor port, reference voltage: 3.3V, VCC voltage: 3.3V.
3. **I2C ports:** 3.3V I2C bus ports
4. **5V power output:** 5V power output to PWM driver.
5. **UART port:** 4-wire UART port, 5V VCC, perfectly working with SunFounder FTDI Serial to USB.
6. **Motor control ports:** 5V for motors, direction control of motors MA and MB and a floating pin NC ; working with SunFounder motor driver module.
7. **Switch:** power switch
8. **Power indicators:** indicating the voltage – 2 indicators on: >7.9V; 1 indicator on: 7.9V~7.4V; no indicator on: <7.4V. To protect the batteries, you're recommended to take them out for charge when there is no indicator on. The power indicators depend on the voltage measured by the simple comparator circuit; the detected voltage may be lower than normal depending on loads, so it is just for reference.
9. **Power port:** 5.5/2.1mm standard DC port, input voltage: 8.4~7.4V (limited operating voltage: 12V~6V).

PCA9685



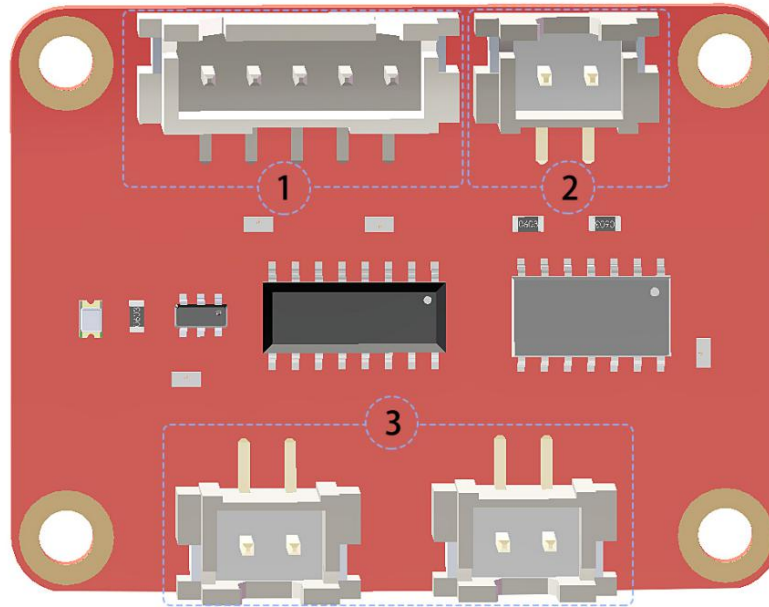
PCA9685 16-channel 12-bit I2C Bus PWM driver. It supports independent PWM output power and is easy to use 4-wire I2C port for connection in parallel, distinguished 3-color ports for PWM output.



1. **PWM output ports:** 3-color ports, independent power PWM output port, connect to the servo directly.
- 2 and 3. **I2C port:** 4-wire I2C port, can be used in parallel. Compatible with 3.3V/5.5V.
4. **PWM power input:** 12V max.
5. **LED:** power indicator for the chip and for the PWM power input.

Motor Driver Module

The motor driver module is a low heat generation one and small packaged motor drive.



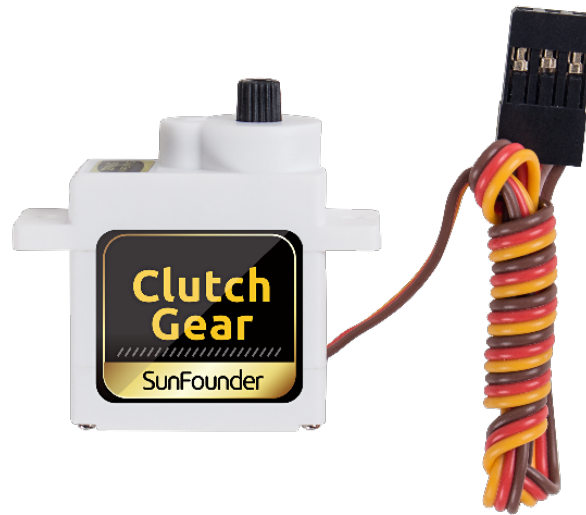
1. **Power and motor control port:** includes pins for supplying the chip and the motors and controlling the motors' direction.
2. **PWM input for the motors:** PWM signal input for adjusting the speed of the two motors.
3. **Motor output port:** output port for two motors.

USB Webcam



This camera supports a wide angle of 120°, which provides a wide and clear vision, thus giving better experience when you're using it on the PiCar-V.

SunFounder SF006C Servo

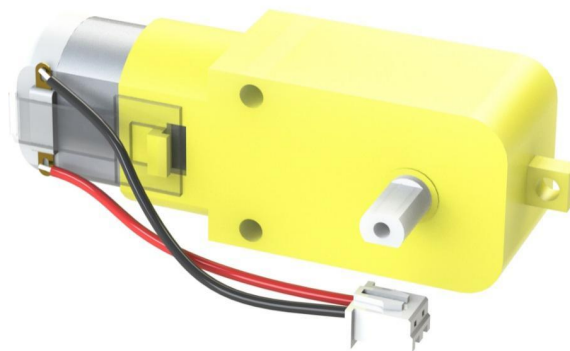


Clutch gear digital servo with a DC core motor inside, After a certain load, the steering gear reducer will automatically clutch and protect the product from damage and normal load.

Function of the Performance:

Item	V = 4.8V	V = 6.0V
Consumption Current* (No Load)	50mA	60mA
Stall Current	550mA	650mA
Rated Torque	0.6 kgf·cm	0.7 kgf·cm
Max. Torque	1.4 kgf·cm	1.6 kgf·cm
No Load Speed	0.14sec/60°	0.12sec/60°

DC Gear Motor



It's a DC motor with a speed reducing gear train. See the parameters below:

Motor	Model	F130SA-11200-38V
	Rated Voltage	4.5V-6V
	No-load Current	≤80mA
	No-load Speed	10000±10%
Gear Reducer	Gear Ratio	1:48
	Speed (no-load)	≈200rpm (≈180rpm in test)
	Current	≤120mA

1.9.3 Filezilla Software



The File Transfer Protocol (FTP) is a standard communication protocol used for the transfer of computer files from a server to a client on a computer network.

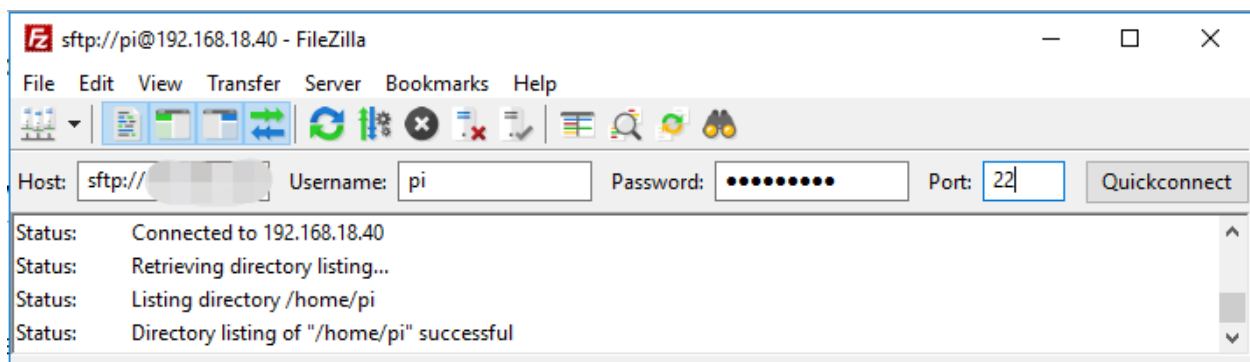
Filezilla is an open source software that not only supports FTP, but also FTP over TLS (FTPS) and SFTP. We can use Filezilla to upload local files (such as pictures and audio, etc.) to the Raspberry Pi, or download files from the Raspberry Pi to the local.

Step 1: Download Filezilla.

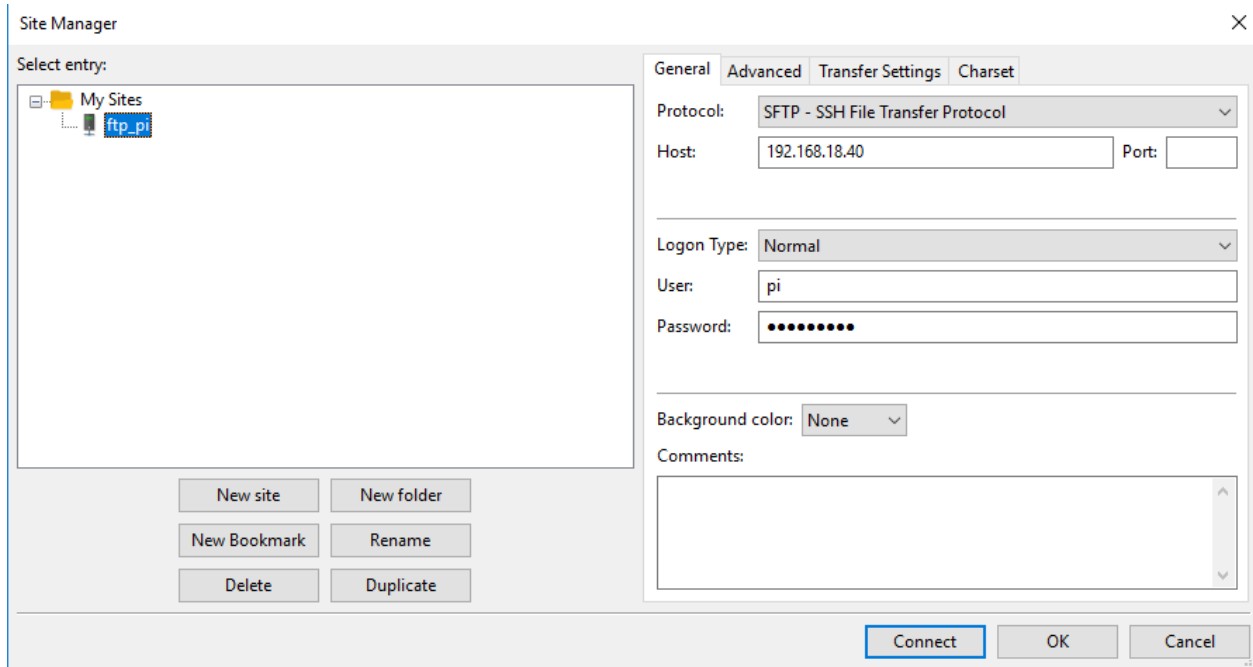
Download the client from [Filezilla's official website](#), Filezilla has a very good tutorial, please refer to: [Documentation - Filezilla](#).

Step 2: Connect to Raspberry Pi

After a quick install open it up and now [connect it to an FTP server](#). It has 3 ways to connect, here we use the **Quick Connect** bar. Enter the **hostname/IP**, **username**, **password** and **port (22)**, then click **Quick Connect** or press **Enter** to connect to the server.

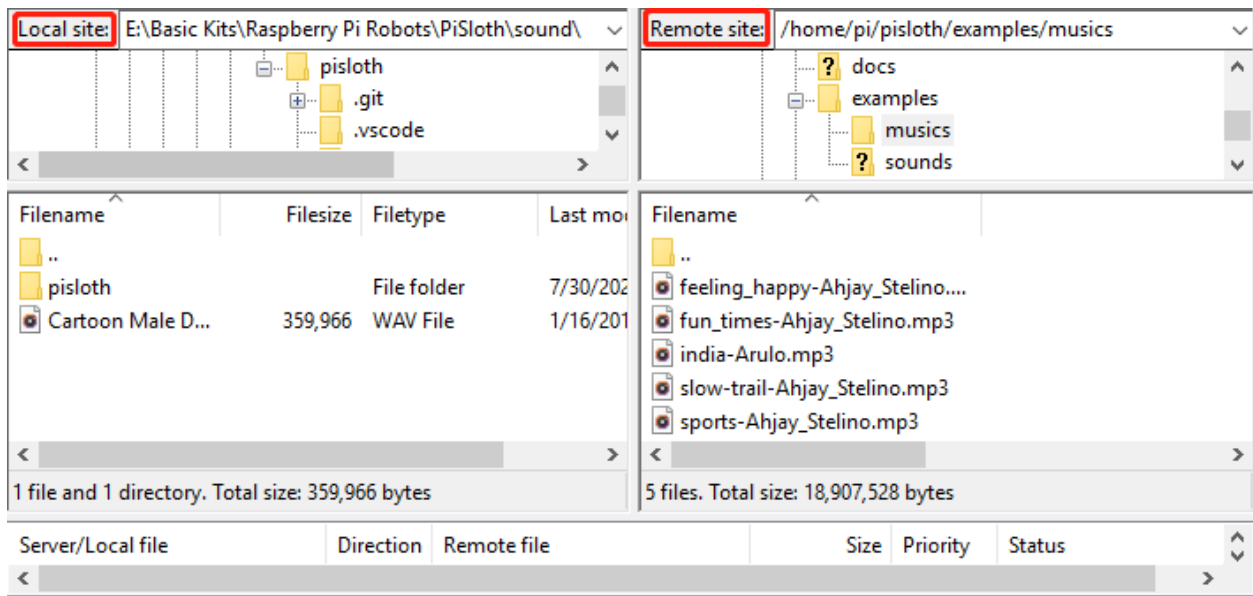


Note: Quick Connect is a good way to test your login information. If you want to create a permanent entry, you can select **File-> Copy Current Connection to Site Manager** after a successful Quick Connect, enter the name and click **OK**. Next time you will be able to connect by selecting the previously saved site inside **File -> Site Manager**.



Step 3: Upload/download files.

You can upload local files to Raspberry Pi by dragging and dropping them, or download the files inside Raspberry Pi files locally.



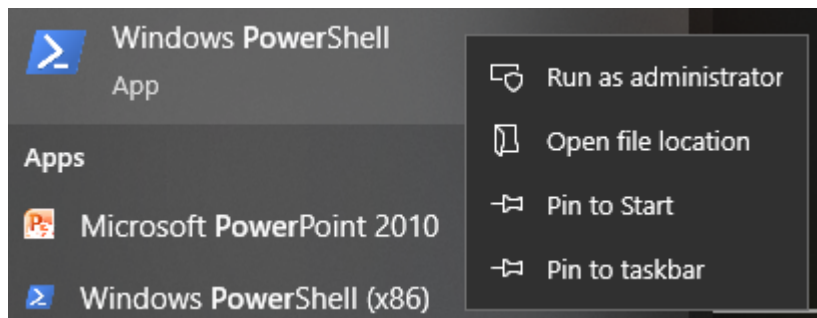
1.9.4 Install OpenSSH via Powershell

When you use `ssh <username>@<hostname>.local` (or `ssh <username>@<IP address>`) to connect to your Raspberry Pi, but the following error message appears.

```
ssh: The term 'ssh' is not recognized as the name of a cmdlet, function,
↪script file, or operable program. Check the
spelling of the name, or if a path was included, verify that the path is
↪correct and try again.
```

It means your computer system is too old and does not have OpenSSH pre-installed, you need to follow the tutorial below to install it manually.

1. Type powershell in the search box of your Windows desktop, right click on the Windows PowerShell, and select Run as administrator from the menu that appears.



2. Use the following command to install OpenSSH.Client.

```
Add-WindowsCapability -Online -Name OpenSSH.Client~~~~0.0.1.0
```

3. After installation, the following output will be returned.

```
Path      :
Online    : True
RestartNeeded : False
```

4. Verify the installation by using the following command.

```
Get-WindowsCapability -Online | Where-Object Name -like 'OpenSSH*'
```

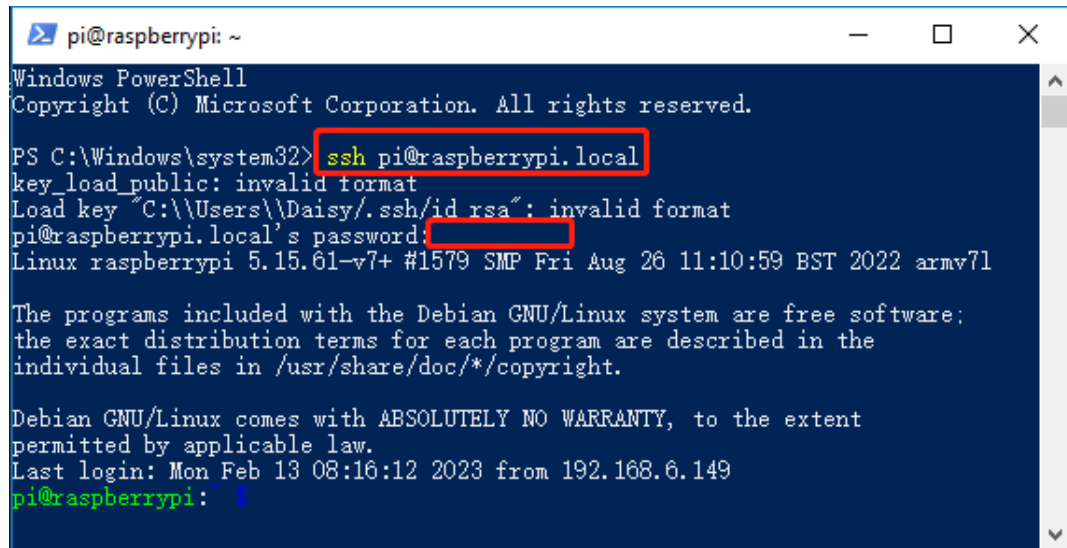
5. It now tells you that OpenSSH.Client has been successfully installed.

```
Name : OpenSSH.Client~~~~0.0.1.0
State : Installed

Name : OpenSSH.Server~~~~0.0.1.0
State : NotPresent
```

Warning: If the above prompt does not appear, it means that your Windows system is still too old, and you are advised to install a third-party SSH tool, like *PuTTY*.

6. Now restart PowerShell and continue to run it as administrator. At this point you will be able to log in to your Raspberry Pi using the `ssh` command, where you will be prompted to enter the password you set up earlier.



```
pi@raspberrypi: ~
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

PS C:\Windows\system32> ssh pi@raspberrypi.local
key_load_public: invalid format
Load key "C:\\Users\\Daisy\\.ssh\\id_rsa": invalid format
pi@raspberrypi.local's password:
Linux raspberrypi 5.15.61-v7+ #1579 SMP Fri Aug 26 11:10:59 BST 2022 armv7l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Mon Feb 13 08:16:12 2023 from 192.168.6.149
pi@raspberrypi: $
```

1.9.5 PuTTY

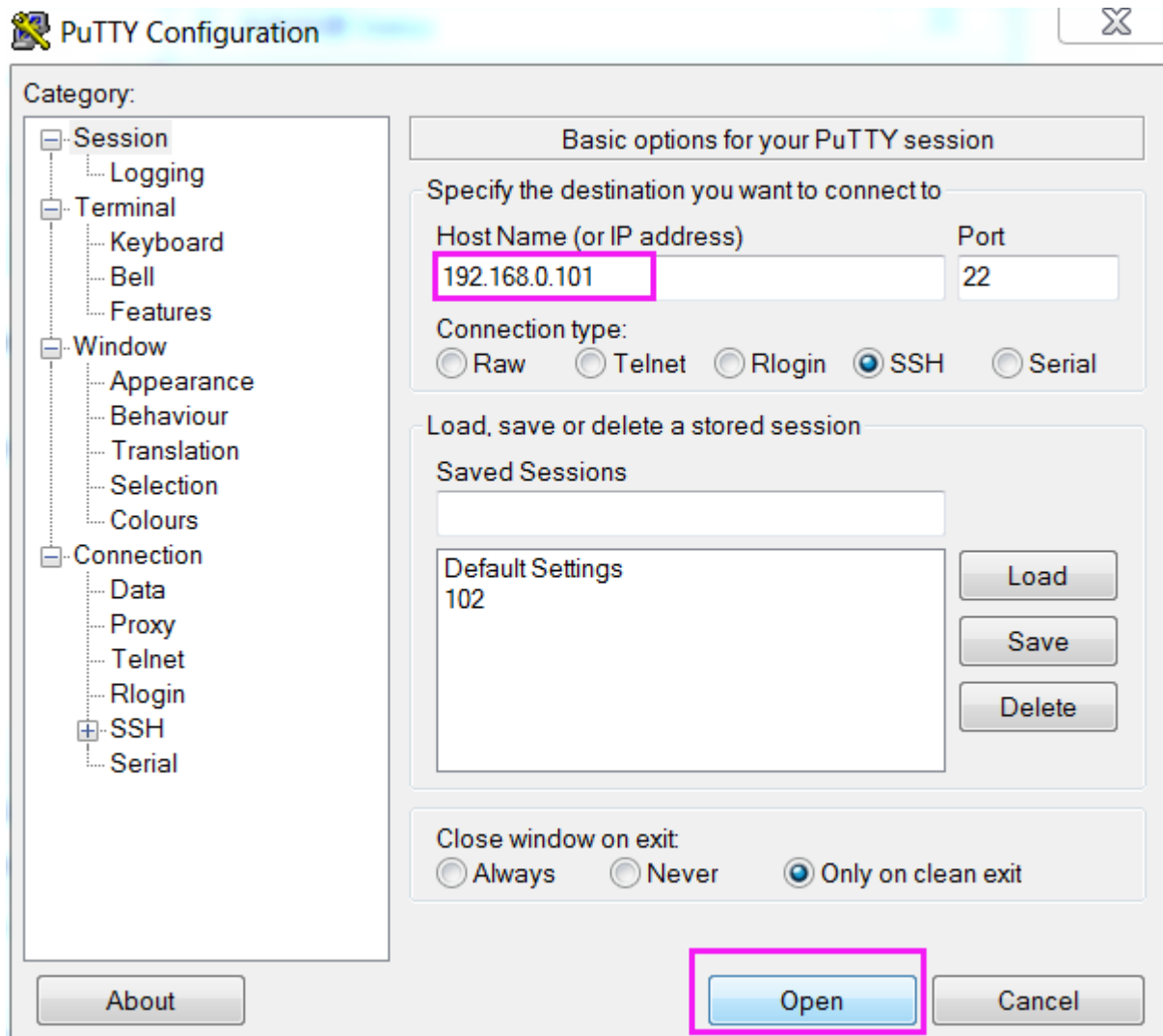
If you are a Windows user, you can use some applications of SSH. Here, we recommend PuTTY.

Step 1

Download PuTTY.

Step 2

Open PuTTY and click **Session** on the left tree-alike structure. Enter the IP address of the RPi in the text box under **Host Name (or IP address)** and **22** under **Port** (by default it is 22).



Step 3

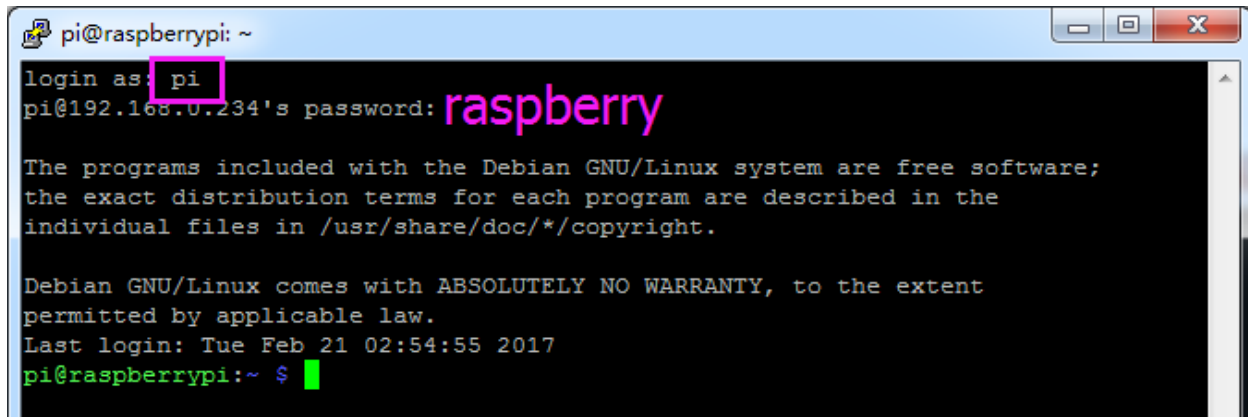
Click **Open**. Note that when you first log in to the Raspberry Pi with the IP address, there prompts a security reminder. Just click **Yes**.

Step 4

When the PuTTY window prompts "**login as:**", type in "**pi**" (the user name of the RPi), and **password:** "raspberrry" (the default one, if you haven't changed it).

Note: When you input the password, the characters do not display on window accordingly, which is normal. What you need is to input the correct password.

If inactive appears next to PuTTY, it means that the connection has been broken and needs to be reconnected.



```
pi@raspberrypi: ~
login as: pi
pi@192.168.0.234's password: raspberry

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Tue Feb 21 02:54:55 2017
pi@raspberrypi:~ $
```

Step 5

Here, we get the Raspberry Pi connected and it is time to conduct the next steps.

1.10 Thank You

Thanks to the evaluators who evaluated our products, the veterans who provided suggestions for the tutorial, and the users who have been following and supporting us. Your valuable suggestions to us are our motivation to provide better products!

Particular Thanks

- Len Davisson
- Kalen Daniel
- Juan Delacosta

Now, could you spare a little time to fill out this questionnaire?

Note: After submitting the questionnaire, please go back to the top to view the results.

COPYRIGHT NOTICE

All contents including but not limited to texts, images, and code in this manual are owned by the SunFounder Company. You should only use it for personal study, investigation, enjoyment, or other non-commercial or nonprofit purposes, under the related regulations and copyrights laws, without infringing the legal rights of the author and relevant right holders. For any individual or organization that uses these for commercial profit without permission, the Company reserves the right to take legal action.