SunFounder PiCrawler Kit

www.sunfounder.com

Aug 20, 2025

CONTENTS

Note: Hello, welcome to the SunFounder Raspberry Pi & Arduino & ESP32 Enthusiasts Community on Facebook! Dive deeper into Raspberry Pi, Arduino, and ESP32 with fellow enthusiasts.

Why Join?

- **Expert Support**: Solve post-sale issues and technical challenges with help from our community and team.
- **Learn & Share**: Exchange tips and tutorials to enhance your skills.
- **Exclusive Previews**: Get early access to new product announcements and sneak peeks.
- **Special Discounts**: Enjoy exclusive discounts on our newest products.
- **Festive Promotions and Giveaways**: Take part in giveaways and holiday promotions.

Ready to explore and create with us? Click [] and join today!

-

Thank you for choosing our .

Note: This document is available in the following languages.

-
-
-
-
-
-
-

Please click on the respective links to access the document in your preferred language.

PiCrawler is a Raspberry Pi quadruped robot with aluminum alloy structure. It is equipped with a camera module, which can perform color recognition, face detection and other items; 12 metal gear servos support it to walk, dance, and pose various postures; the ultrasonic module on the body allows it to quickly detect obstacles in front of it; the expansion board-robot HAT is equipped with a speaker, allowing it to express emotions such as happiness and excitement.

This document includes the list and assembly pdf, Robot HAT introduction and PiCrawler programming.

The programming part is divided into two chapters: *Play with Ezblock* & *Play with Python*, each chapter allows you to explain how to make PiCrawler work the way you want.

Ezblock Studio is a development platform developed by SunFounder for beginners, aiming to lower the barriers to entry for Raspberry Pi. It has two programming languages: Graphical and Python, which can be used on almost all different types of devices. With Bluetooth and Wi-Fi support, you can download codes on Ezblock Studio and remotely control Raspberry Pi.

More experienced makers can use the popular programming language-Python.

If you have any questions, please send an email to service@sunfounder.com and we will respond as soon as possible.

Content

Note: Hello, welcome to the SunFounder Raspberry Pi & Arduino & ESP32 Enthusiasts Community on Facebook! Dive deeper into Raspberry Pi, Arduino, and ESP32 with fellow enthusiasts.

Why Join?

- **Expert Support**: Solve post-sale issues and technical challenges with help from our community and team.
- **Learn & Share**: Exchange tips and tutorials to enhance your skills.
- **Exclusive Previews**: Get early access to new product announcements and sneak peeks.
- **Special Discounts**: Enjoy exclusive discounts on our newest products.
- **Festive Promotions and Giveaways**: Take part in giveaways and holiday promotions.

Ready to explore and create with us? Click [] and join today!

CHAPTER

ONE

PLAY WITH PYTHON

If you want to program in python, then you will need to learn some basic Python programming skills and basic knowledge of Raspberry Pi, please configure the Raspberry Pi first according to *Quick Guide on Python*.

Note: Hello, welcome to the SunFounder Raspberry Pi & Arduino & ESP32 Enthusiasts Community on Facebook! Dive deeper into Raspberry Pi, Arduino, and ESP32 with fellow enthusiasts.

Why Join?

- **Expert Support**: Solve post-sale issues and technical challenges with help from our community and team.
- **Learn & Share**: Exchange tips and tutorials to enhance your skills.
- **Exclusive Previews**: Get early access to new product announcements and sneak peeks.
- **Special Discounts**: Enjoy exclusive discounts on our newest products.
- **Festive Promotions and Giveaways**: Take part in giveaways and holiday promotions.

Ready to explore and create with us? Click [] and join today!

1.1 Quick Guide on Python

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.

If you are familiar with Raspberry Pi and can open the command line successfully, then you can skip the first 3 parts and then complete the last part.

Note: Hello, welcome to the SunFounder Raspberry Pi & Arduino & ESP32 Enthusiasts Community on Facebook! Dive deeper into Raspberry Pi, Arduino, and ESP32 with fellow enthusiasts.

Why Join?

- **Expert Support**: Solve post-sale issues and technical challenges with help from our community and team.
- **Learn & Share**: Exchange tips and tutorials to enhance your skills.
- **Exclusive Previews**: Get early access to new product announcements and sneak peeks.
- **Special Discounts**: Enjoy exclusive discounts on our newest products.
- **Festive Promotions and Giveaways**: Take part in giveaways and holiday promotions.

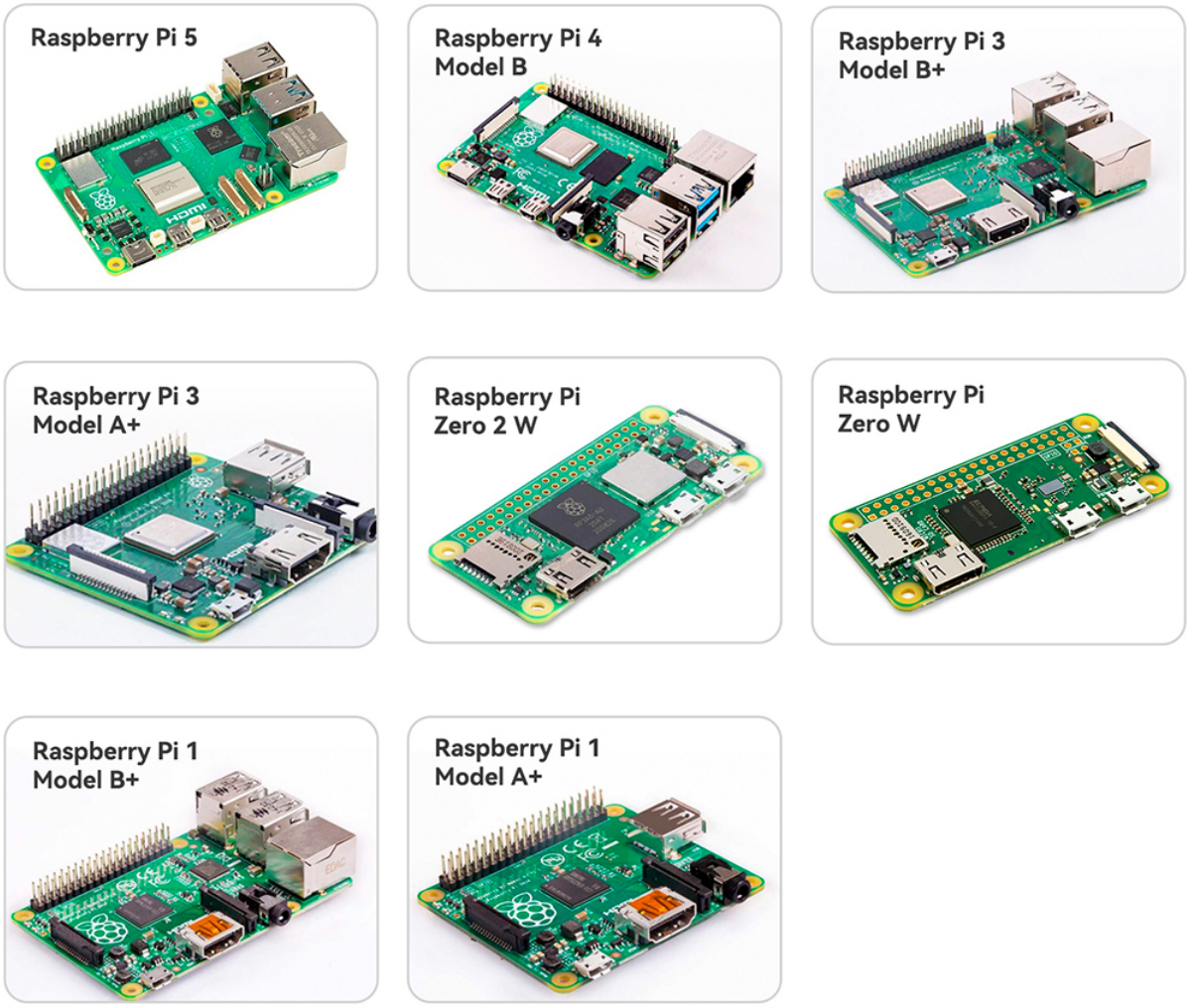
Ready to explore and create with us? Click [] and join today!

1.1.1 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.

Note: Hello, welcome to the SunFounder Raspberry Pi & Arduino & ESP32 Enthusiasts Community on Facebook! Dive deeper into Raspberry Pi, Arduino, and ESP32 with fellow enthusiasts.

Why Join?

- **Expert Support**: Solve post-sale issues and technical challenges with help from our community and team.
- **Learn & Share**: Exchange tips and tutorials to enhance your skills.
- **Exclusive Previews**: Get early access to new product announcements and sneak peeks.
- **Special Discounts**: Enjoy exclusive discounts on our newest products.
- **Festive Promotions and Giveaways**: Take part in giveaways and holiday promotions.

Ready to explore and create with us? Click [] and join today!

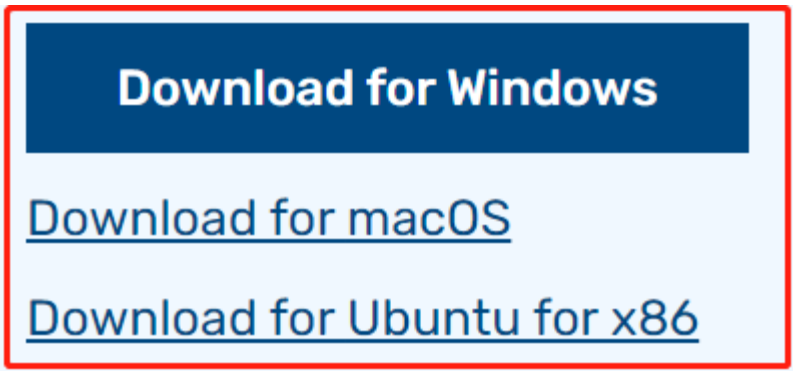
1.1.2 2. Installing the OS

Required Components

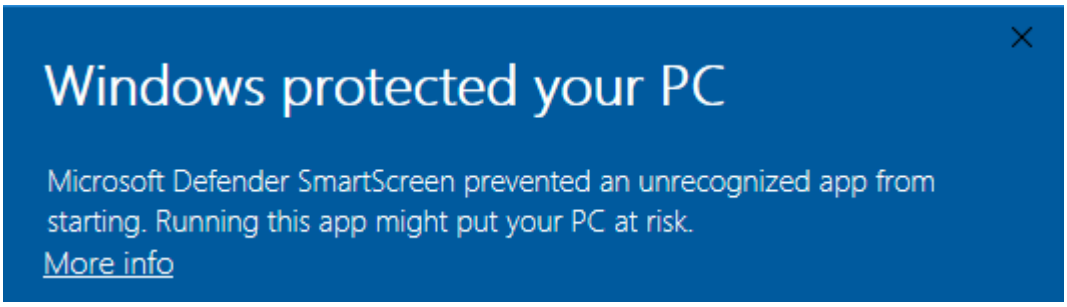
- A Personal Computer
- A Micro SD card and Reader

1. Install Raspberry Pi Imager

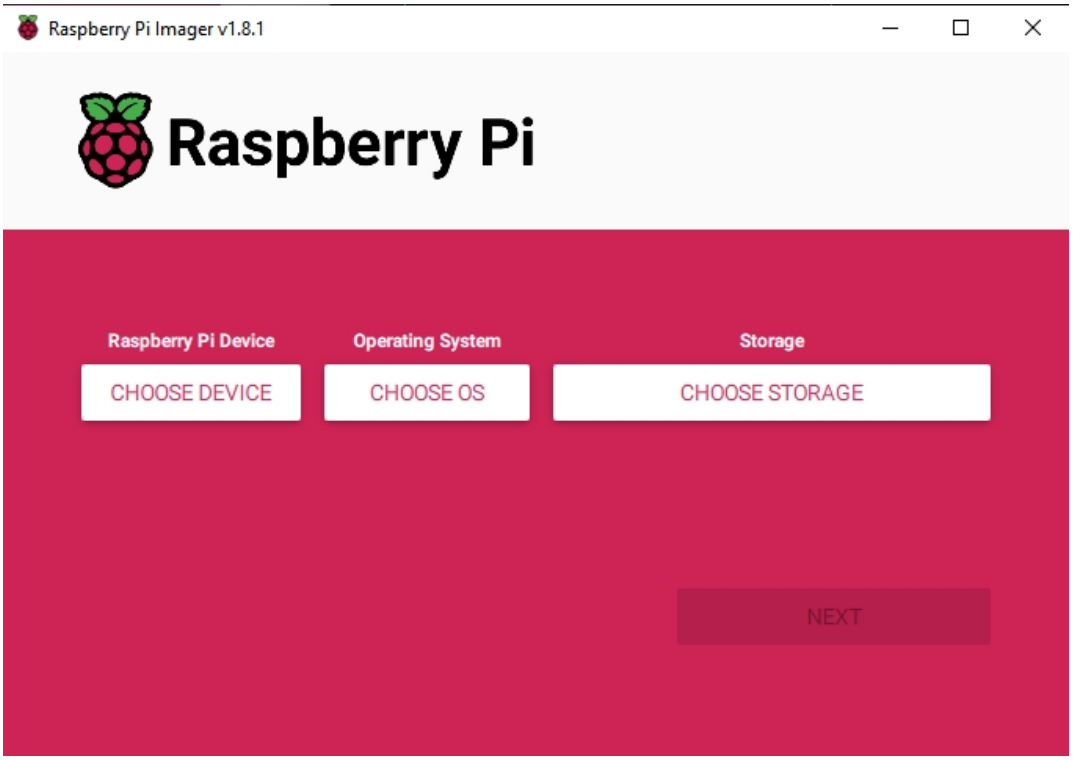
1. Visit the Raspberry Pi software download page at Raspberry Pi Imager. Choose the Imager version compatible with your operating system. Download and open the file to initiate installation.



2. A security prompt may appear during installation, depending on your operating system. For example, Windows might display a warning message. In such cases, select **More info** and then **Run anyway**. Follow the on-screen guidance to complete the installation of the Raspberry Pi Imager.

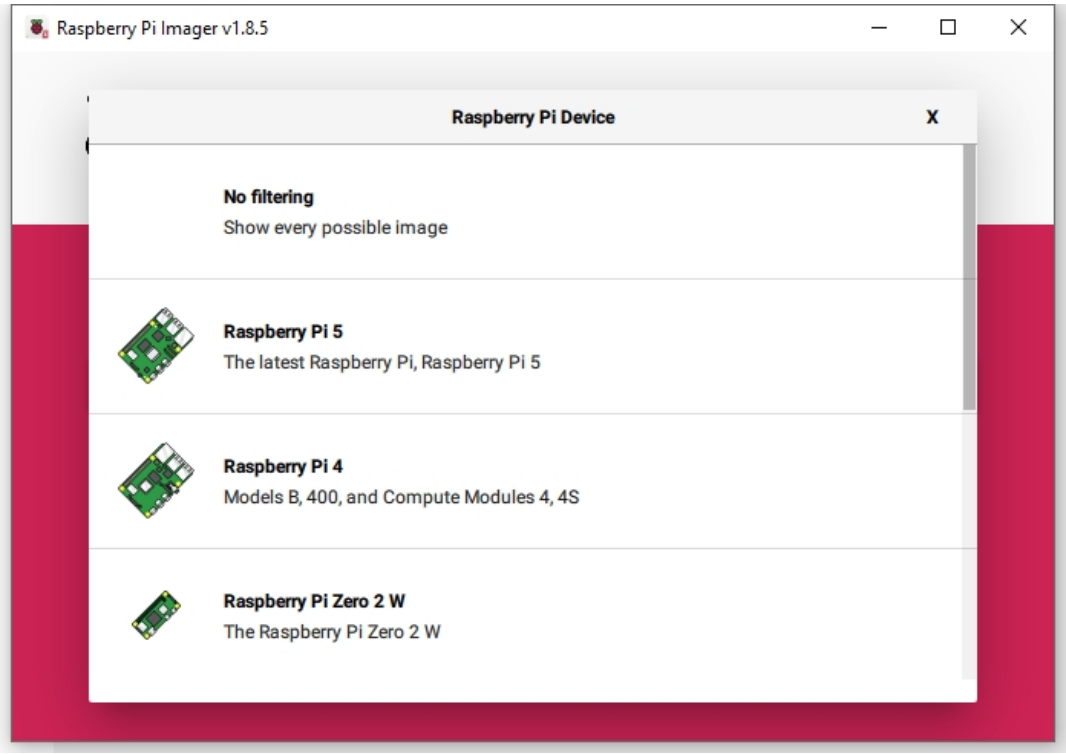


3. Launch the Raspberry Pi Imager application by clicking its icon or typing `rpi-imager` in your terminal.

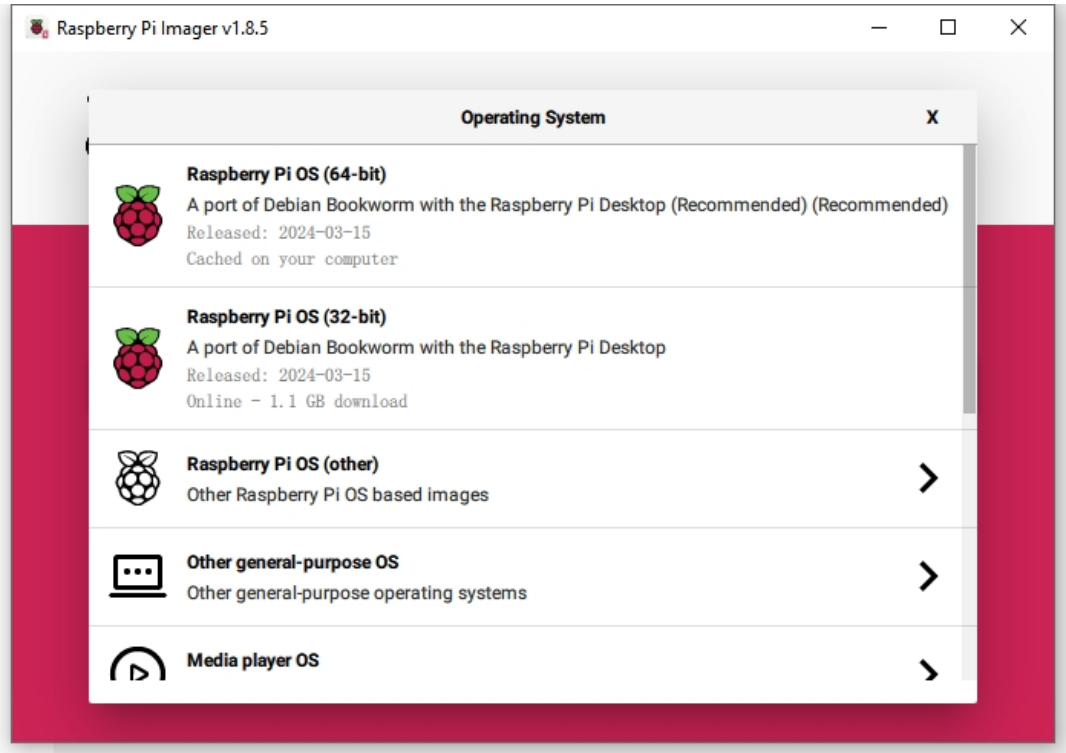


2. Install OS to Micro SD Card

1. Insert your SD card into your computer or laptop using a Reader.
2. Within the Imager, click **Raspberry Pi Device** and select the Raspberry Pi model from the dropdown list.

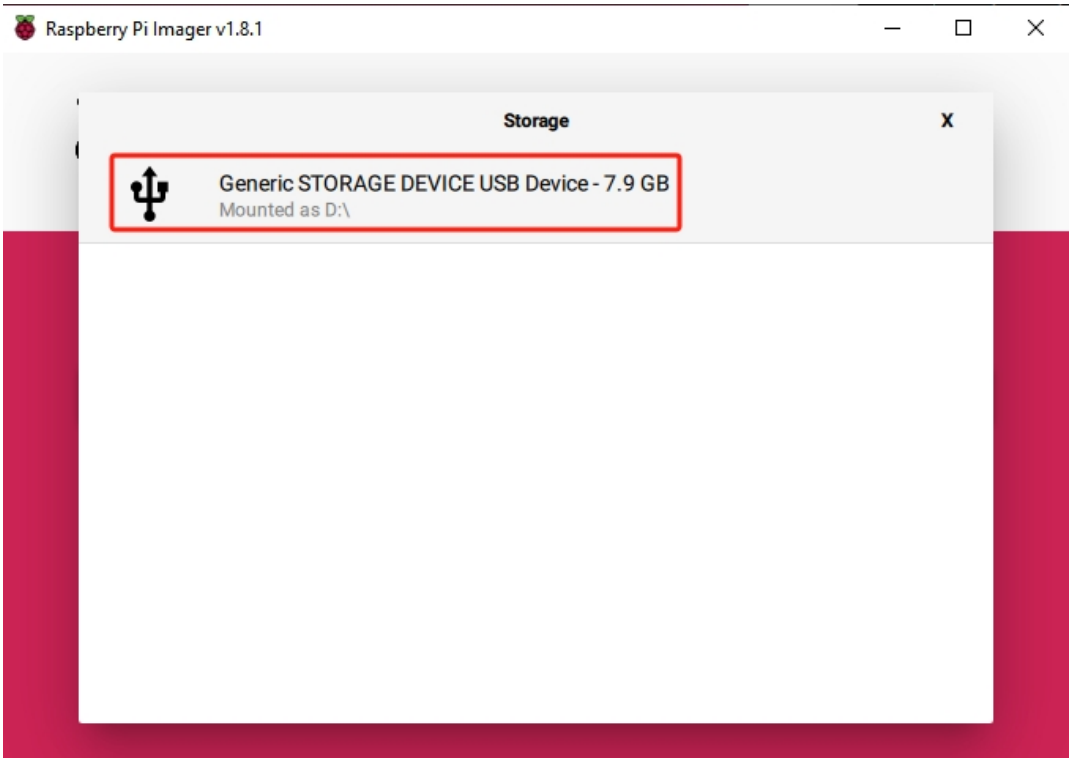


3. Select **Operating System** and opt for the recommended operating system version.



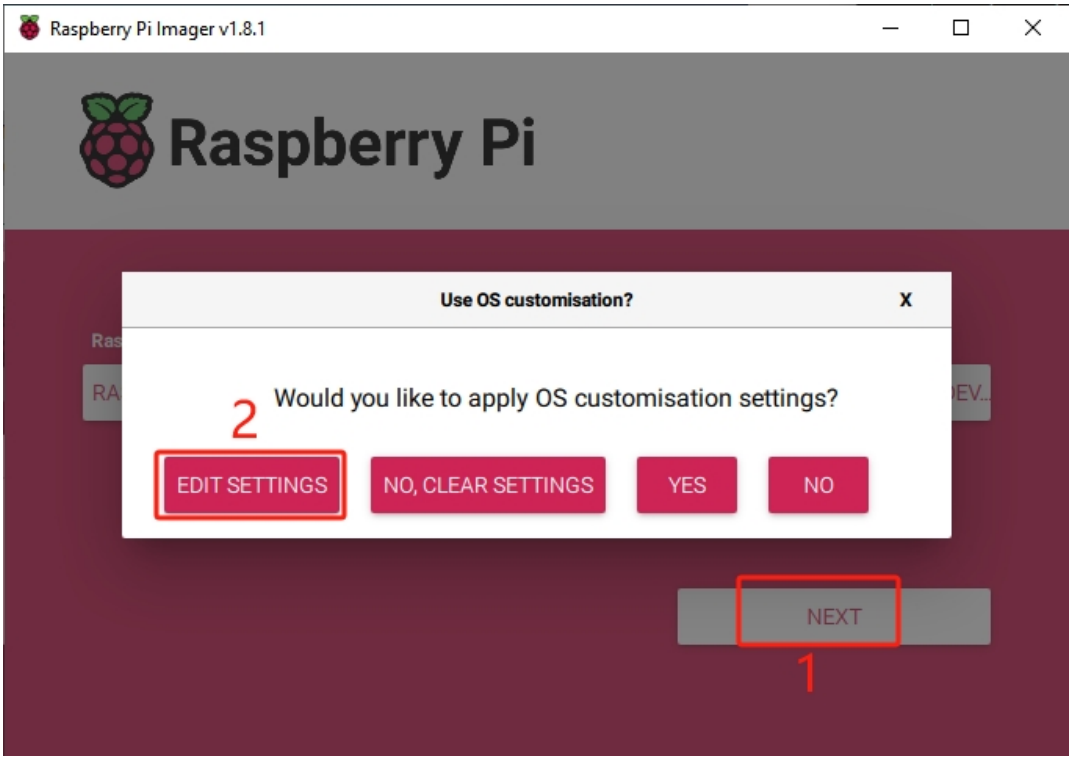
4. Click **Choose Storage** and select the appropriate storage device for the installation.

Note: Ensure you select the correct storage device. To avoid confusion, disconnect any additional storage devices if multiple ones are connected.



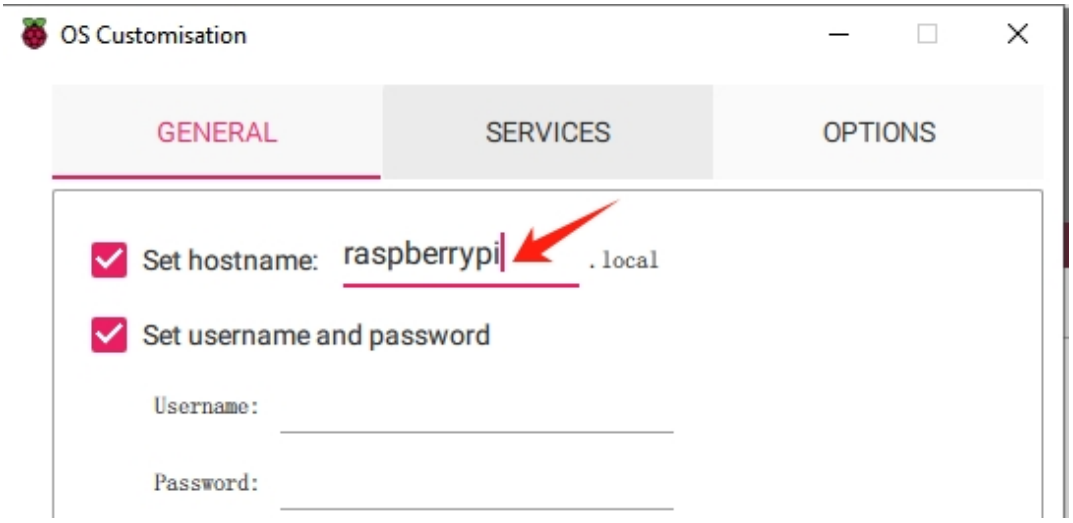
5. Click **NEXT** and then **EDIT SETTINGS** to tailor your OS settings.

Note: If you have a monitor for your Raspberry Pi, you can skip the next steps and click 'Yes' to begin the installation. Adjust other settings later on the monitor.



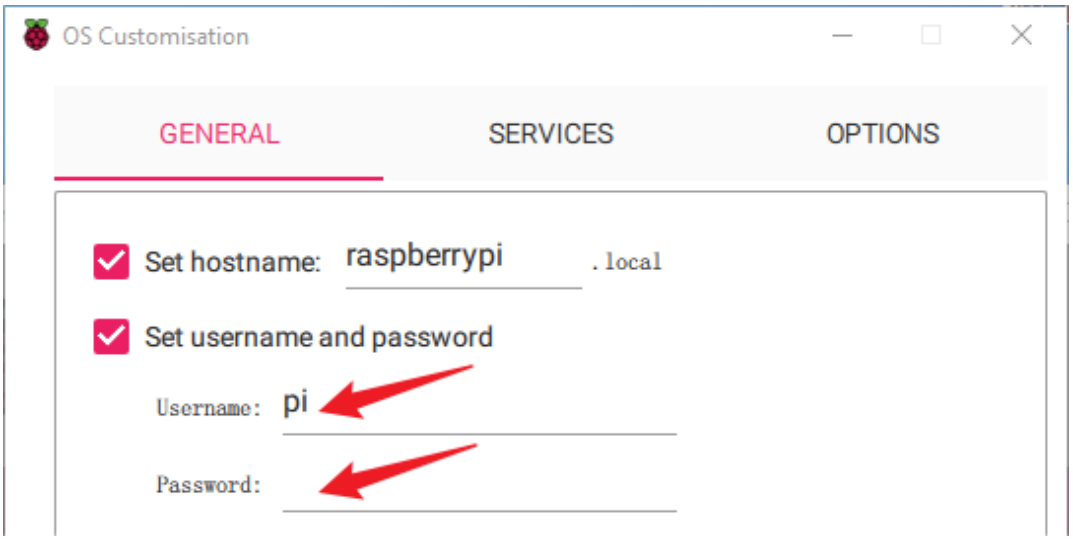
6. Define a **hostname** for your Raspberry Pi.

Note: The hostname is your Raspberry Pi's network identifier. You can access your Pi using `<hostname>.local` or `<hostname>.lan`.



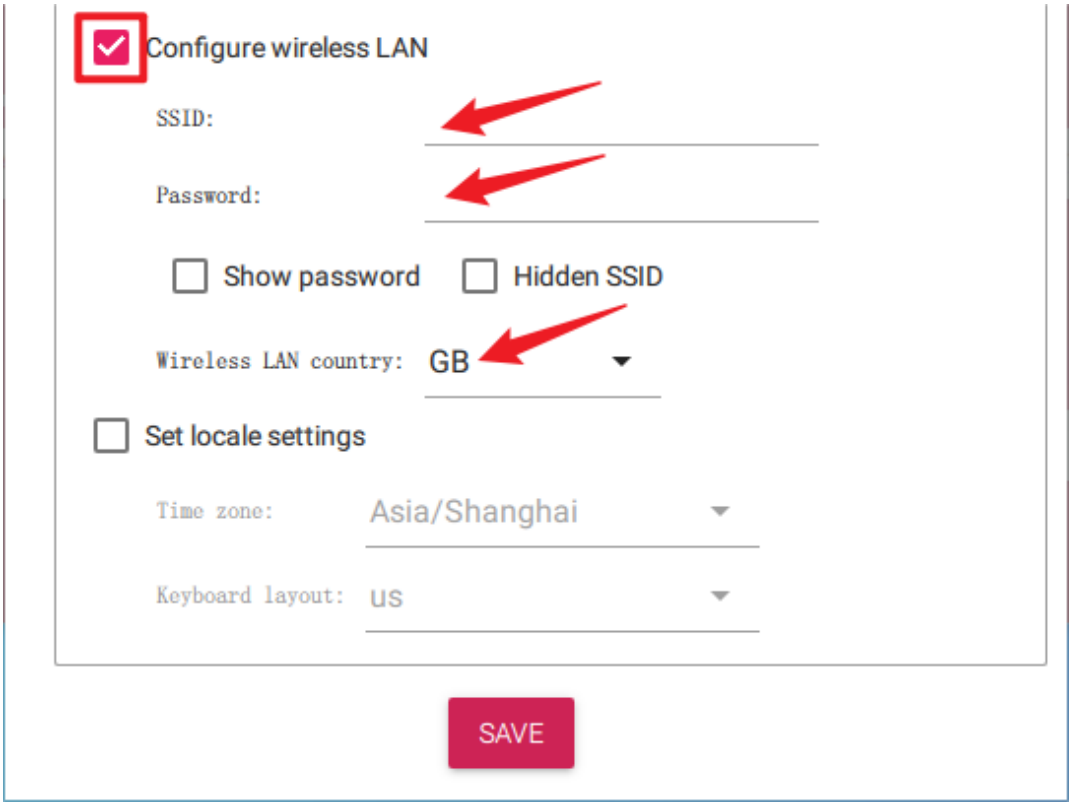
7. Create a **Username** and **Password** for the Raspberry Pi's administrator account.

Note: Establishing a unique username and password is vital for securing your Raspberry Pi, which lacks a default password.



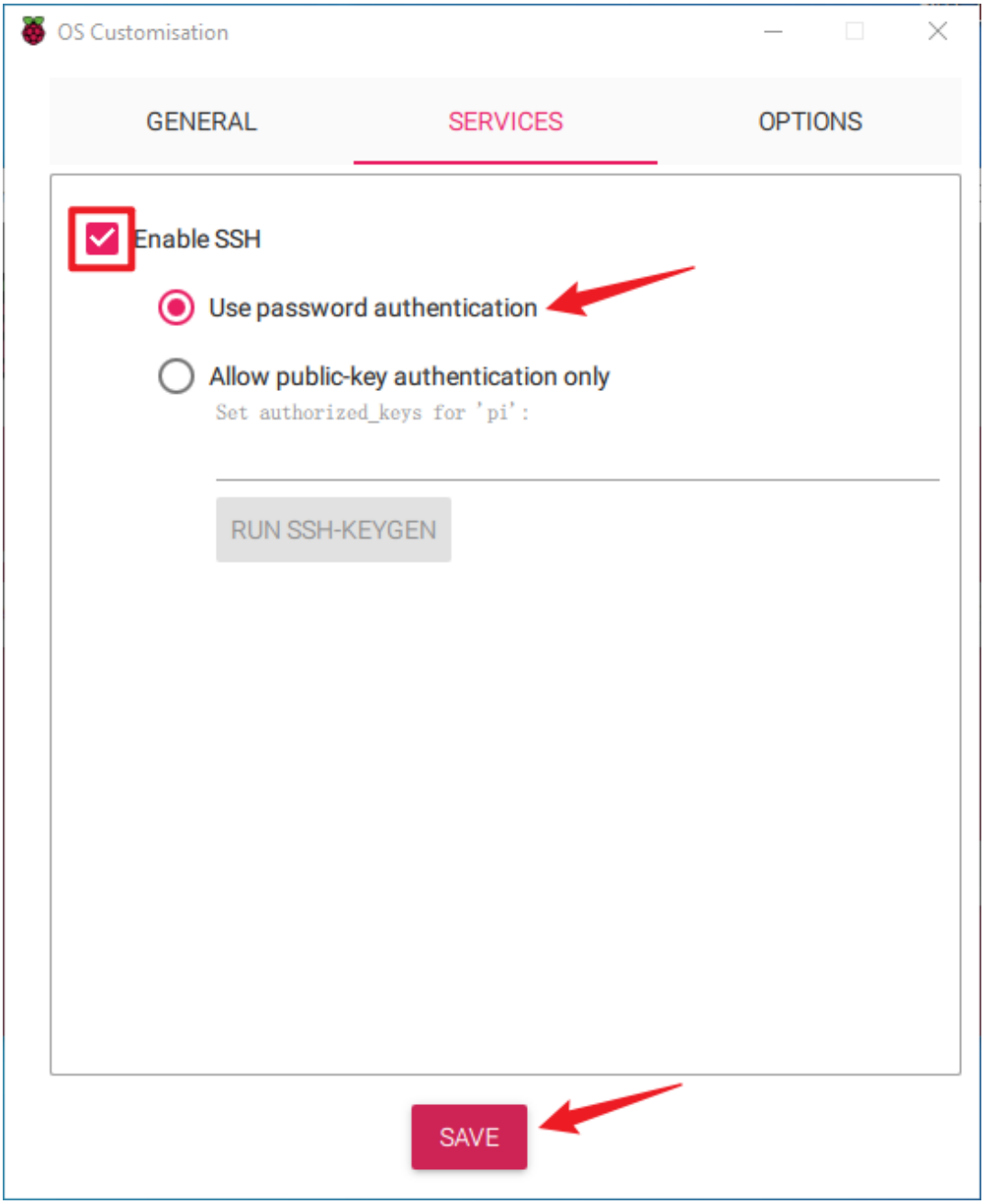
8. Configure the wireless LAN by providing your network's **SSID** and **Password**.

Note: Set the `Wireless LAN country` to the two-letter ISO/IEC alpha2 code corresponding to your location.

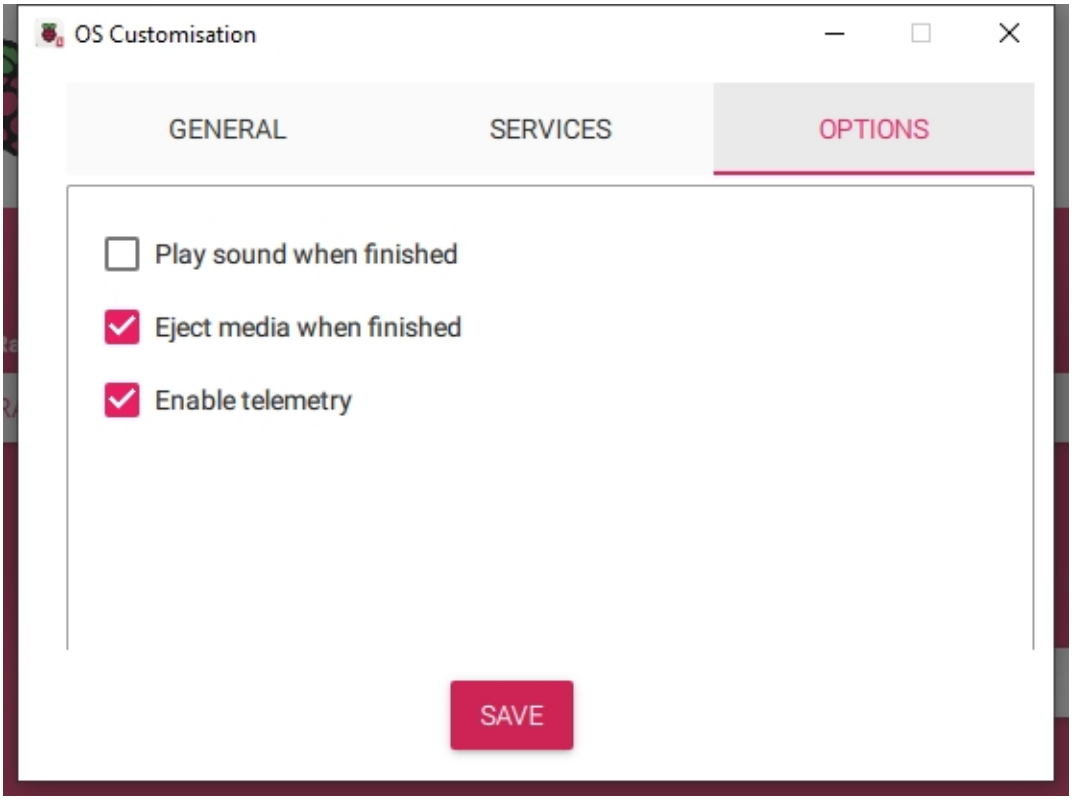


9. To remotely connect to your Raspberry Pi, enable SSH in the Services tab.

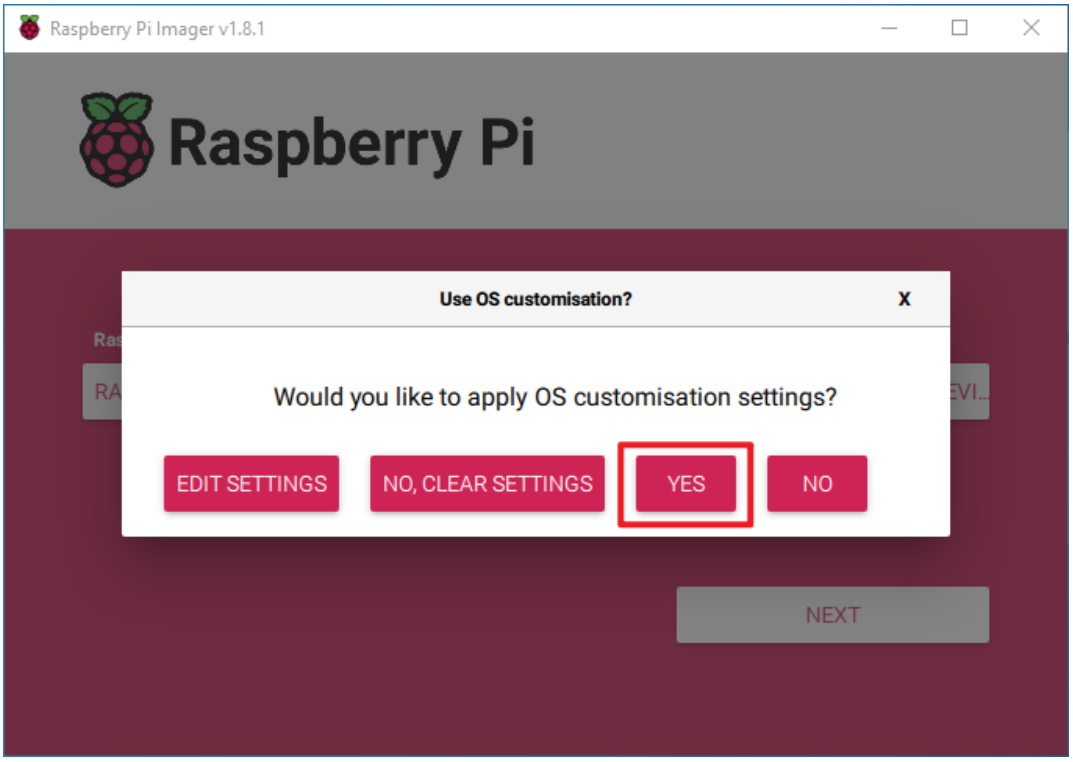
- For **password authentication**, use the username and password from the General tab.
- For public-key authentication, choose "Allow public-key authentication only". If you have an RSA key, it will be used. If not, click "Run SSH-keygen" to generate a new key pair.



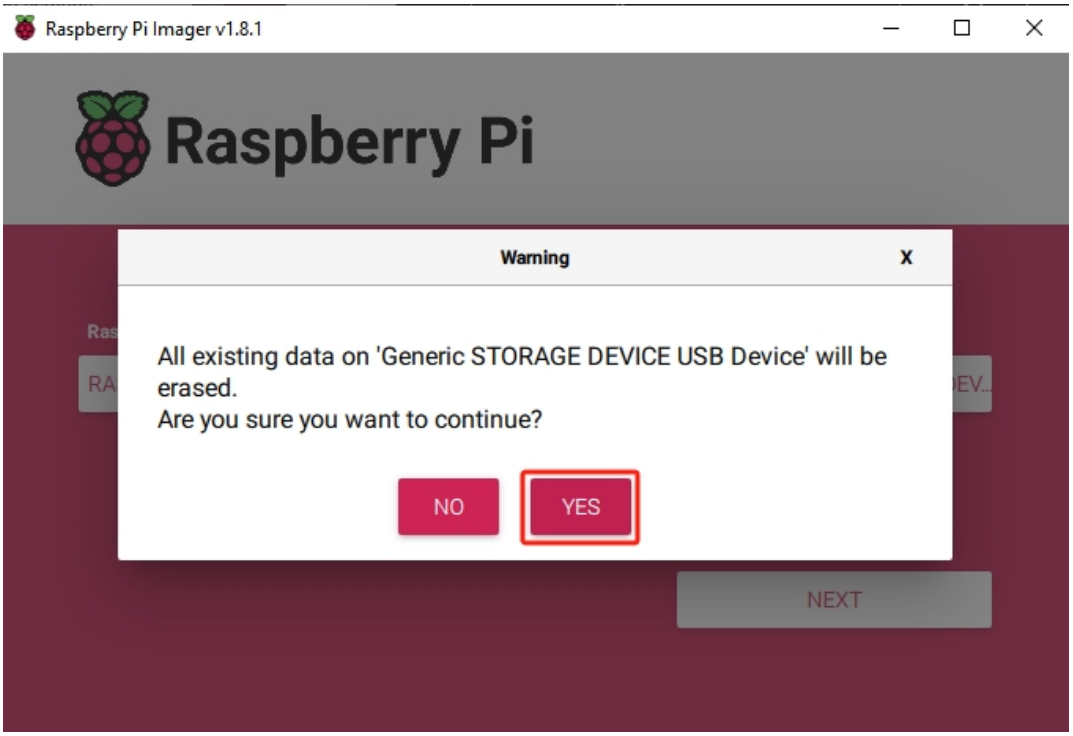
10. The **Options** menu lets you configure Imager's behavior during a write, including playing sound when finished, ejecting media when finished, and enabling telemetry.



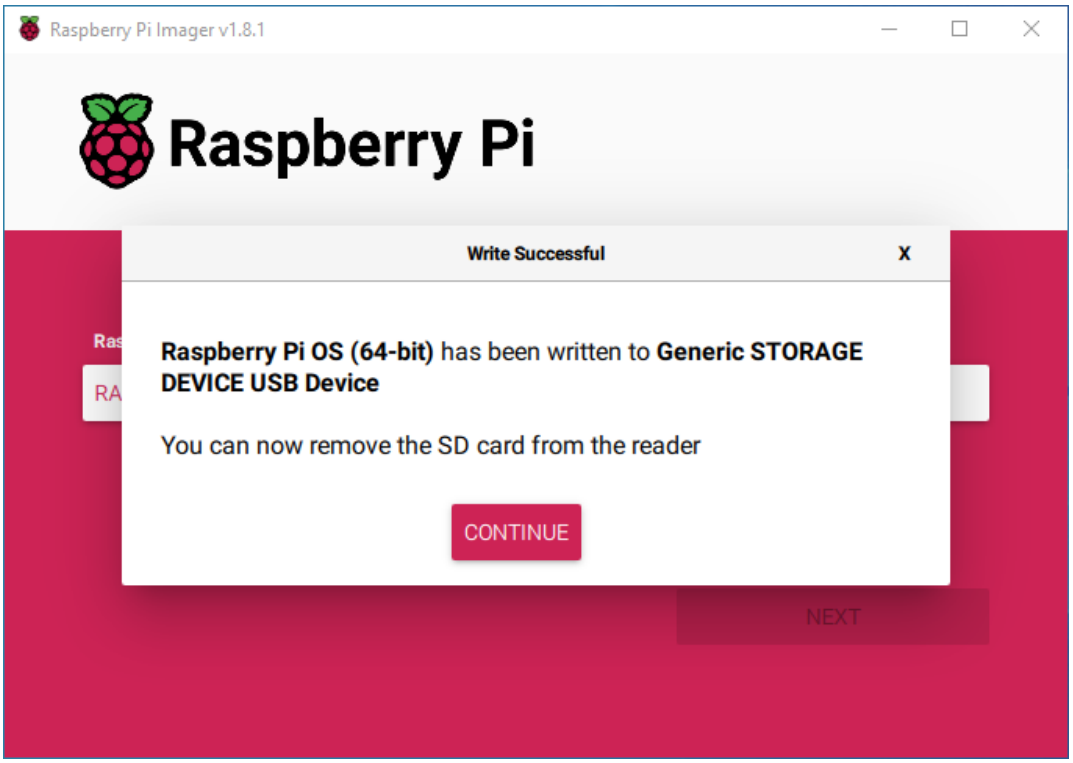
11. When you've finished entering OS customisation settings, click **Save** to save your customisation. Then, click **Yes** to apply them when writing the image.



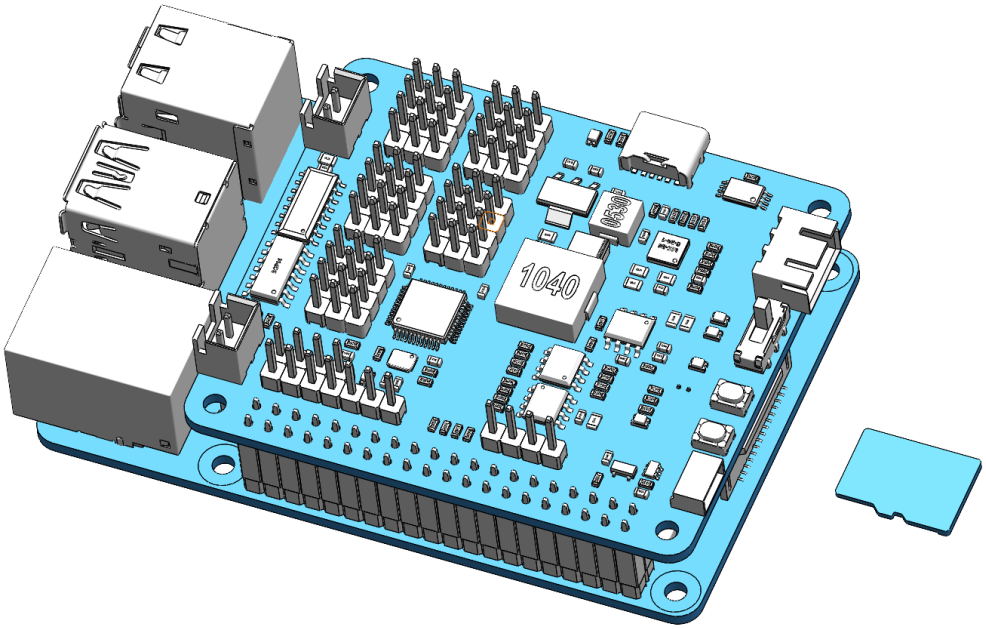
12. If the SD card contains existing data, ensure you back it up to prevent data loss. Proceed by clicking **Yes** if no backup is needed.



13. When you see the "Write Successful" popup, your image has been completely written and verified. You're now ready to boot a Raspberry Pi from the Micro SD Card!



14. Now you can insert the SD card set up with Raspberry Pi OS into the microSD card slot located on the underside of the Raspberry Pi.



Note: Hello, welcome to the SunFounder Raspberry Pi & Arduino & ESP32 Enthusiasts Community on Facebook! Dive deeper into Raspberry Pi, Arduino, and ESP32 with fellow enthusiasts.

Why Join?

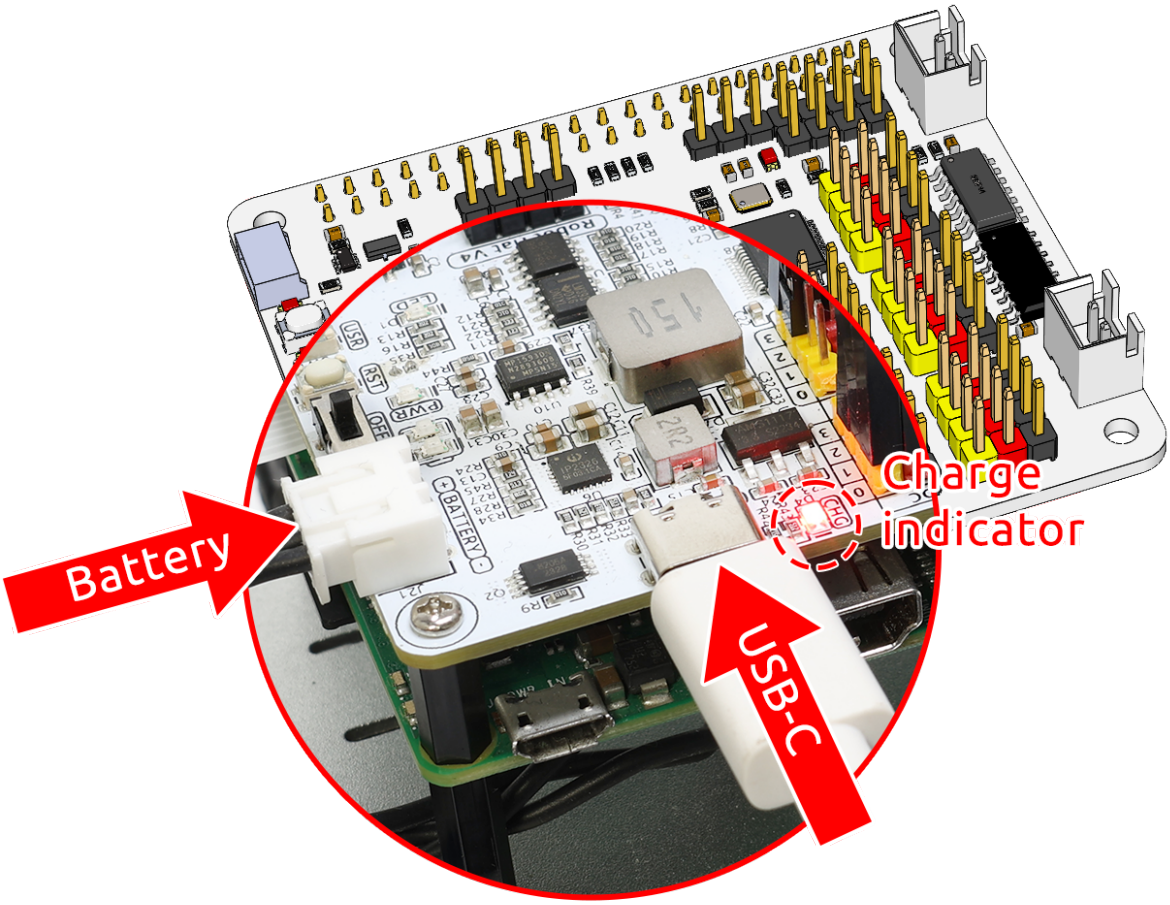
- **Expert Support**: Solve post-sale issues and technical challenges with help from our community and team.
- **Learn & Share**: Exchange tips and tutorials to enhance your skills.
- **Exclusive Previews**: Get early access to new product announcements and sneak peeks.
- **Special Discounts**: Enjoy exclusive discounts on our newest products.
- **Festive Promotions and Giveaways**: Take part in giveaways and holiday promotions.

Ready to explore and create with us? Click [] and join today!

1.1.3 3. Power Supply for Raspberry Pi (Important)

Charge

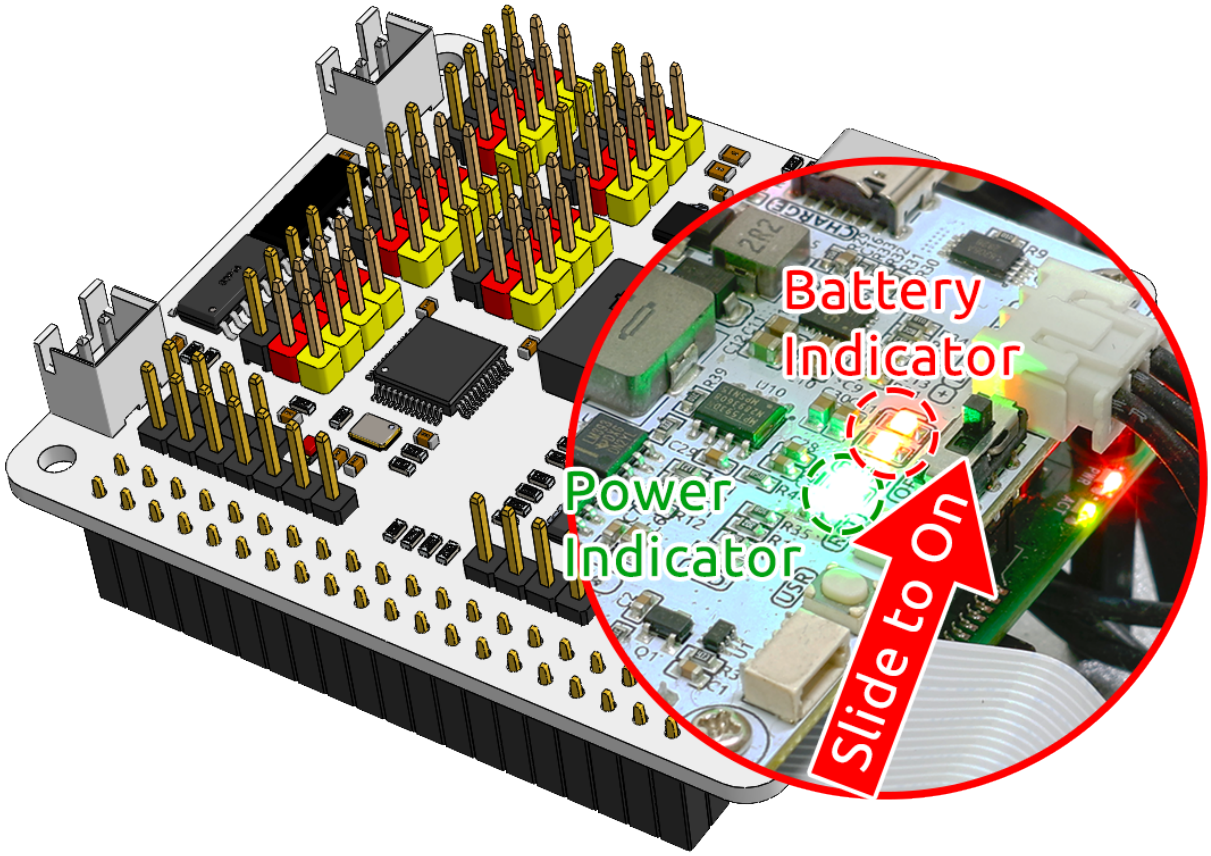
Insert the battery cable. Next, insert the USB-C cable to charge the battery. You will need to provide your own charger; we recommend a 5V 3A charger, or your commonly used smartphone charger will suffice.



Note: Connect an external Type-C power source to the Type-C port on the robot hat; it will immediately start charging the battery, and a red indicator light will illuminate.When the battery is fully charged, the red light will automatically turn off.

Power ON

Turn on the power switch. The Power indicator light and the battery level indicator light will illuminate.



Wait for a few seconds, and you will hear a slight beep, indicating that the Raspberry Pi has successfully booted.

Note: If both battery level indicator lights are off, please charge the battery. When you need extended programming or debugging sessions, you can keep the Raspberry Pi operational by inserting the USB-C cable to charge the battery simultaneously.

Note: Hello, welcome to the SunFounder Raspberry Pi & Arduino & ESP32 Enthusiasts Community on Facebook! Dive deeper into Raspberry Pi, Arduino, and ESP32 with fellow enthusiasts.

Why Join?

- **Expert Support**: Solve post-sale issues and technical challenges with help from our community and team.
- **Learn & Share**: Exchange tips and tutorials to enhance your skills.
- **Exclusive Previews**: Get early access to new product announcements and sneak peeks.
- **Special Discounts**: Enjoy exclusive discounts on our newest products.
- **Festive Promotions and Giveaways**: Take part in giveaways and holiday promotions.

Ready to explore and create with us? Click [] and join today!

1.1.4 4. Set up Your Raspberry Pi

If You Have a Screen

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

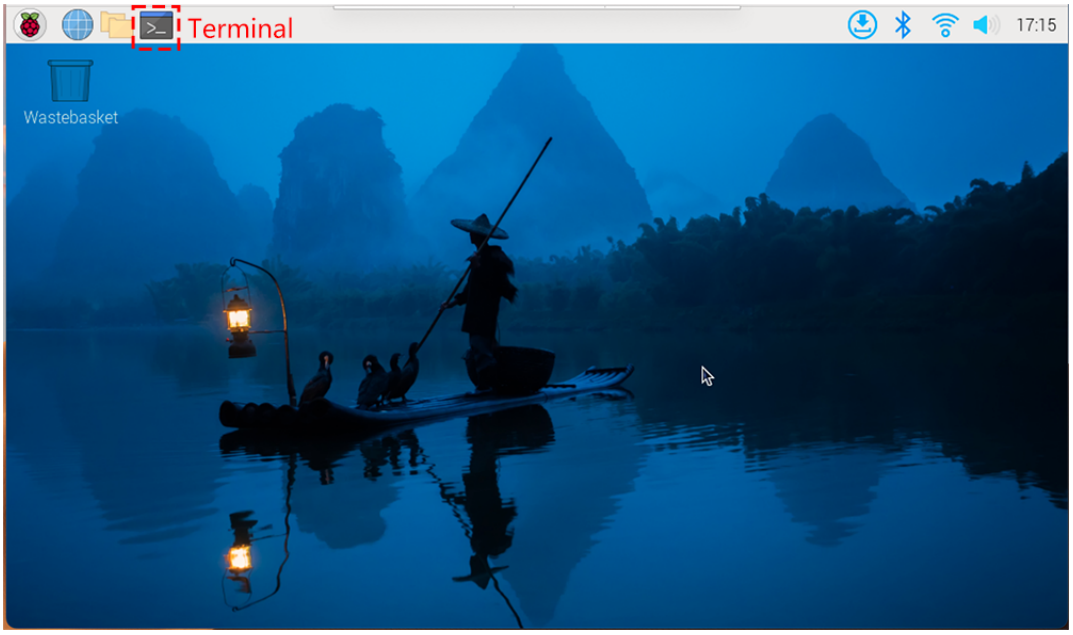
Required Components

- Any Raspberry Pi
- 1 * Power Adapter
- 1 * Micro SD card
- 1 * Screen Power Adapter
- 1 * HDMI cable
- 1 * Screen
- 1 * Mouse
- 1 * Keyboard

1. Insert the SD card you've set up with Raspberry Pi OS into the micro SD card slot on the underside of your Raspberry Pi.
2. Plug in the Mouse and Keyboard.
3. 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).

4. Use the power adapter to power the Raspberry Pi.
5. After a few seconds, the Raspberry Pi OS desktop will be displayed. Now you can open the Terminal to start entering commands.



If You Have No Screen

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

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.

Note: Hello, welcome to the SunFounder Raspberry Pi & Arduino & ESP32 Enthusiasts Community on Facebook! Dive deeper into Raspberry Pi, Arduino, and ESP32 with fellow enthusiasts.

Why Join?

- **Expert Support**: Solve post-sale issues and technical challenges with help from our community and team.
- **Learn & Share**: Exchange tips and tutorials to enhance your skills.
- **Exclusive Previews**: Get early access to new product announcements and sneak peeks.
- **Special Discounts**: Enjoy exclusive discounts on our newest products.
- **Festive Promotions and Giveaways**: Take part in giveaways and holiday promotions.

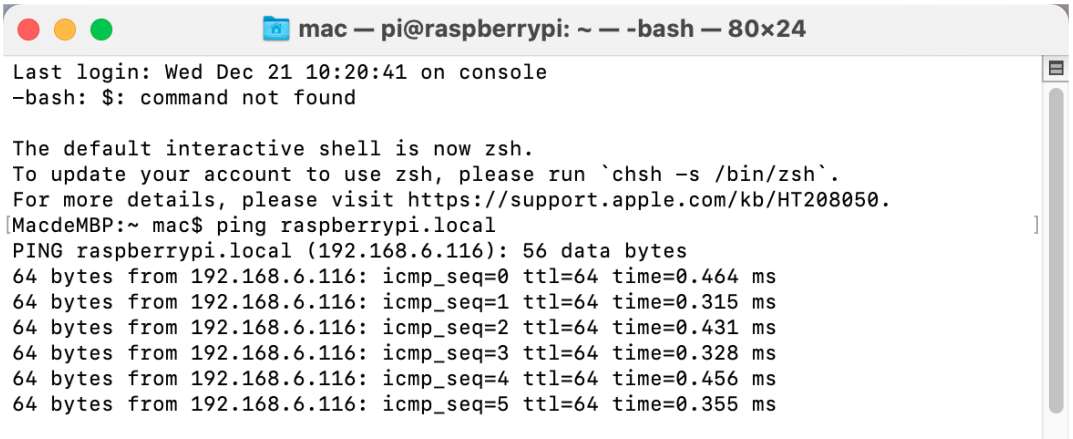
Ready to explore and create with us? Click [] and join today!

For Mac OS X Users

For Mac OS X users, SSH (Secure Shell) offers a secure and convenient method to remotely access and control a Raspberry Pi. This is particularly handy for working with the Raspberry Pi remotely or when it's not connected to a monitor. Using the Terminal application on a Mac, you can establish this secure connection. The process involves an SSH command incorporating the Raspberry Pi's username and hostname. During the initial connection, a security prompt will ask for confirmation of the Raspberry Pi's authenticity.

1. To connect to the Raspberry Pi, type the following SSH command:

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



2. A security message will appear during your first login. Respond with **yes** to proceed.

```
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.
Are you sure you want to continue connecting (yes/no/[fingerprint])?
```

3. Input the password for the Raspberry Pi. Be aware that the password won't display on the screen as you type, which is a standard security feature.

```
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:~ $
```

Note: Hello, welcome to the SunFounder Raspberry Pi & Arduino & ESP32 Enthusiasts Community on Facebook! Dive deeper into Raspberry Pi, Arduino, and ESP32 with fellow enthusiasts.

Why Join?

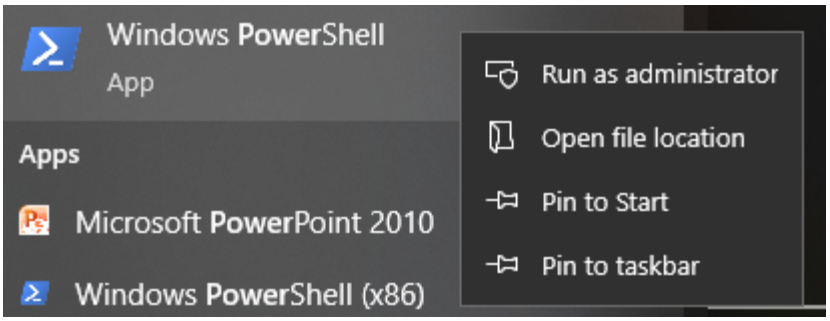
- **Expert Support**: Solve post-sale issues and technical challenges with help from our community and team.
- **Learn & Share**: Exchange tips and tutorials to enhance your skills.
- **Exclusive Previews**: Get early access to new product announcements and sneak peeks.
- **Special Discounts**: Enjoy exclusive discounts on our newest products.
- **Festive Promotions and Giveaways**: Take part in giveaways and holiday promotions.

Ready to explore and create with us? Click [] and join today!

For Windows Users

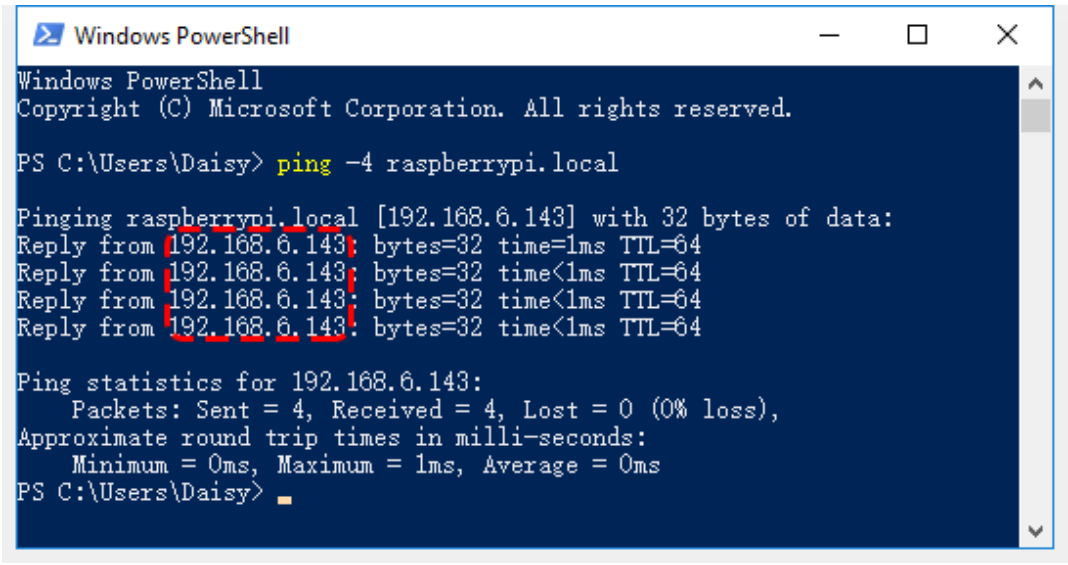
For Windows 10 or higher users, remote login to a Raspberry Pi can be achieved through the following steps:

1. Search for `powershell` in your Windows search box. Right-click on `Windows PowerShell` and select `Run as administrator`.



2. Determine your Raspberry Pi's IP address by typing `ping -4 <hostname>.local` in PowerShell.

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



The Raspberry Pi's IP address will be displayed once it's connected to the network.

- If the terminal displays `Ping request could not find host pi.local. Please check the name and try again.`, verify the hostname you've entered is correct.
- If the IP address still isn't retrievable, check your network or WiFi settings on the Raspberry Pi.

3. Once the IP address is confirmed, log in to your Raspberry Pi using `ssh <username>@<hostname>.local` or `ssh <username>@<IP address>`.

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

> **Warning:** If an error appears stating `The term 'ssh' is not recognized as the name of a cmdlet...`, your system may not have SSH tools pre-installed. In this case, you need to manually install OpenSSH following *Install OpenSSH via Powershell*, or use a third-party tool, like PuTTY.

4. A security message will appear on your first login. Enter `yes` to proceed.

```
The authenticity of host 'raspberrypi.local␣
↪(2400:2410:2101:5800:635b:f0b6:2662:8cba)' can't be established.
```

(continues on next page)

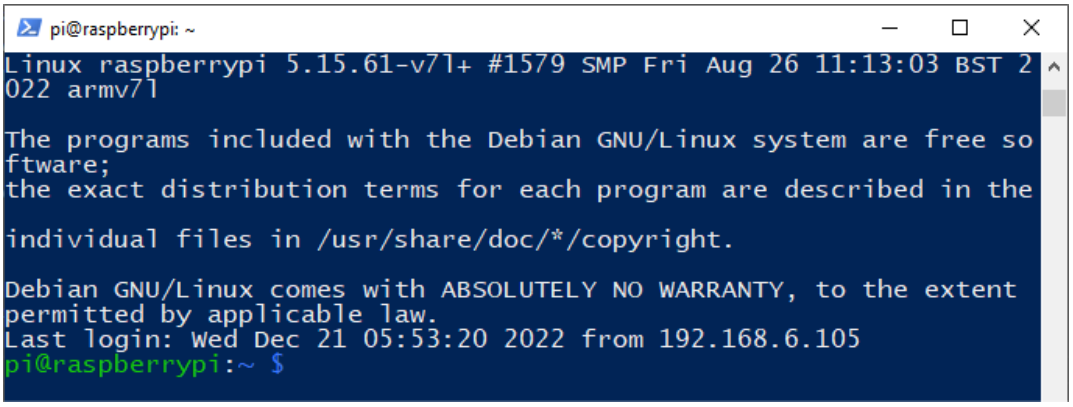
(continued from previous page)

```
ED25519 key fingerprint is SHA256:oo7x3ZSgAo032wD1tE8eW0fFM/
↪kmewIvRwkBys6XRwg.
Are you sure you want to continue connecting (yes/no/[fingerprint])?
```

5. Enter the password you previously set. Note that the password characters won't be displayed on the screen, which is a standard security feature.

   Note: The absence of visible characters when typing the password is normal. Ensure you input the correct password.

6. Once connected, your Raspberry Pi is ready for remote operations.



Note: Hello, welcome to the SunFounder Raspberry Pi & Arduino & ESP32 Enthusiasts Community on Facebook! Dive deeper into Raspberry Pi, Arduino, and ESP32 with fellow enthusiasts.

Why Join?

- **Expert Support**: Solve post-sale issues and technical challenges with help from our community and team.
- **Learn & Share**: Exchange tips and tutorials to enhance your skills.
- **Exclusive Previews**: Get early access to new product announcements and sneak peeks.
- **Special Discounts**: Enjoy exclusive discounts on our newest products.
- **Festive Promotions and Giveaways**: Take part in giveaways and holiday promotions.

Ready to explore and create with us? Click [] and join today!

For Linux/Unix Users

1. Locate and open the **Terminal** on your Linux/Unix system.
2. Ensure your Raspberry Pi is connected to the same network. Verify this by typing `ping <hostname>.local`. For example:

   ```
   ping raspberrypi.local
   ```

   You should see the Raspberry Pi's IP address if it's connected to the network.

- If the terminal shows a message like `Ping request could not find host pi.local. Please check the name and try again.`, double-check the hostname you've entered.
- If you're unable to retrieve the IP address, inspect your network or WiFi settings on the Raspberry Pi.

3. Initiate an SSH connection by typing `ssh <username>@<hostname>.local` or `ssh <username>@<IP address>`. For instance:

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

4. On your first login, you'll encounter a security message. Type `yes` to proceed.

```
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.
Are you sure you want to continue connecting (yes/no/[fingerprint])?
```

5. Enter the password you previously set. Note that for security reasons, the password won't be visible as you type.

> **Note:** It's normal for the password characters not to display in the terminal. Just ensure to enter the correct password.

6. Once you've successfully logged in, your Raspberry Pi is now connected, and you're ready to proceed to the next step.

Note: Hello, welcome to the SunFounder Raspberry Pi & Arduino & ESP32 Enthusiasts Community on Facebook! Dive deeper into Raspberry Pi, Arduino, and ESP32 with fellow enthusiasts.

Why Join?

- **Expert Support**: Solve post-sale issues and technical challenges with help from our community and team.
- **Learn & Share**: Exchange tips and tutorials to enhance your skills.
- **Exclusive Previews**: Get early access to new product announcements and sneak peeks.
- **Special Discounts**: Enjoy exclusive discounts on our newest products.
- **Festive Promotions and Giveaways**: Take part in giveaways and holiday promotions.

Ready to explore and create with us? Click [] and join today!

Remote Desktop Access for Raspberry Pi

For those preferring a graphical user interface (GUI) over command-line access, the Raspberry Pi supports remote desktop functionality. This guide will walk you through setting up and using VNC (Virtual Network Computing) for remote access.

We recommend using VNC® Viewer for this purpose.

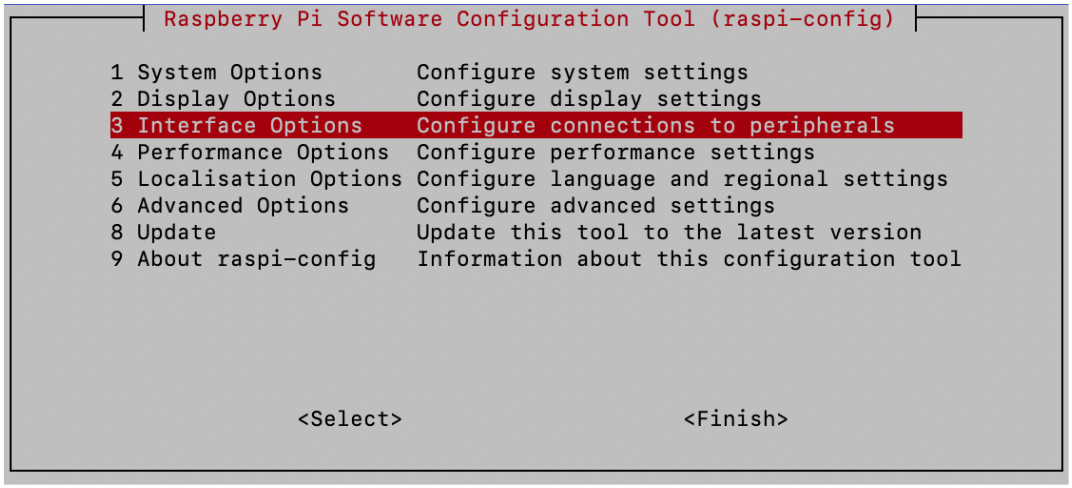
Enabling VNC Service on Raspberry Pi

VNC service comes pre-installed in the Raspberry Pi OS but is disabled by default. Follow these steps to enable it:

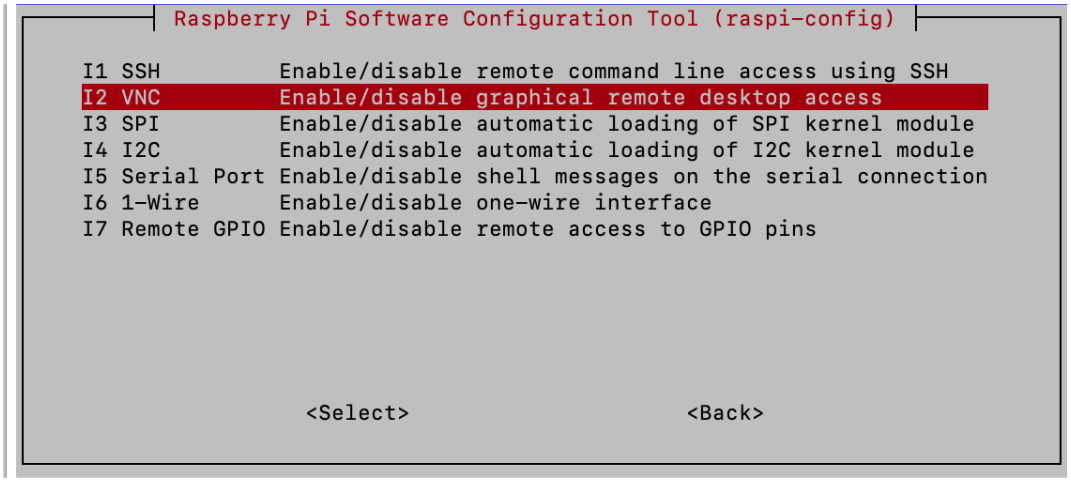
1. Enter the following command in the Raspberry Pi terminal:

```
sudo raspi-config
```

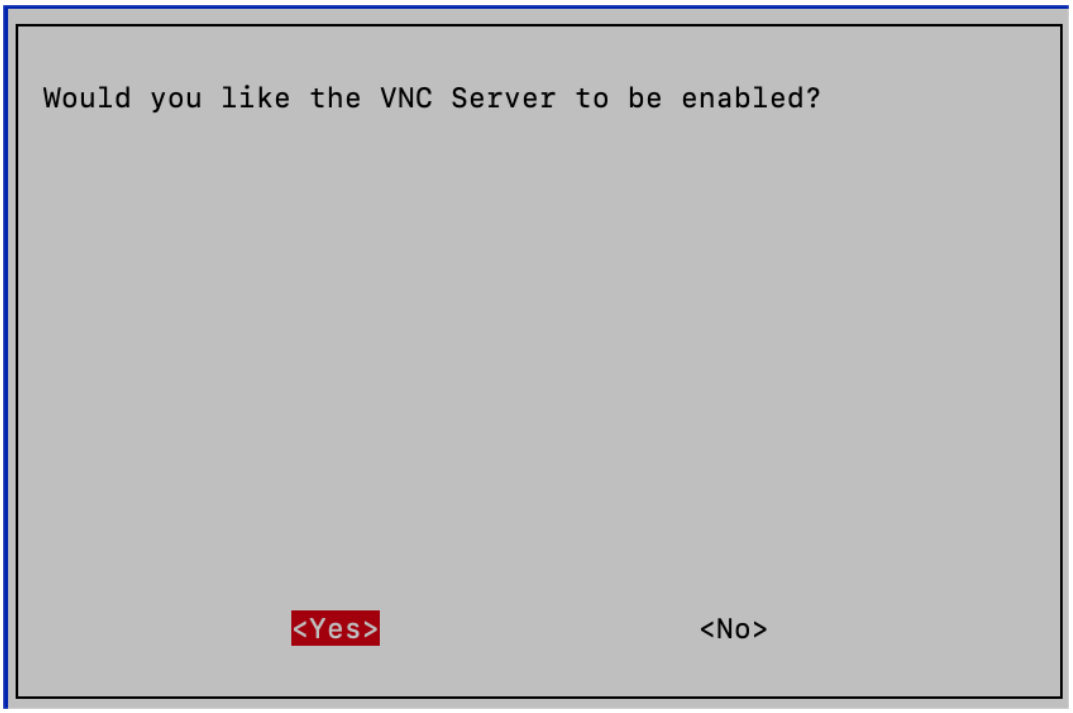
2. Navigate to **Interfacing Options** using the down arrow key, then press **Enter**.



3. Select **VNC** from the options.

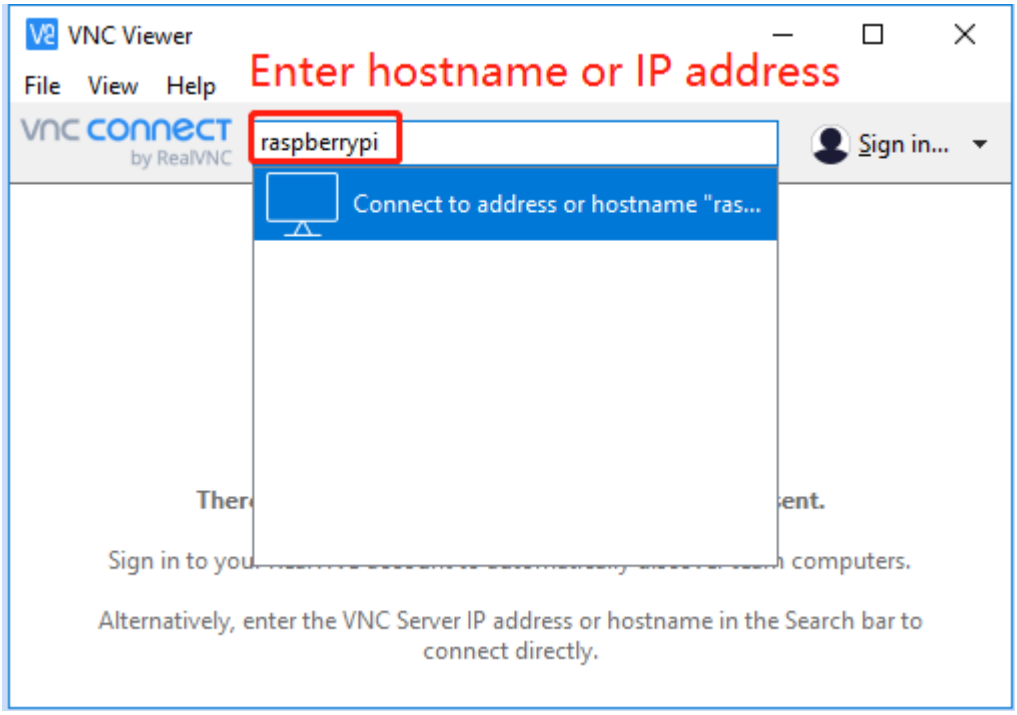


4. Use the arrow keys to choose **<Yes>** -> **<OK>** -> **<Finish>** and finalize the VNC service activation.

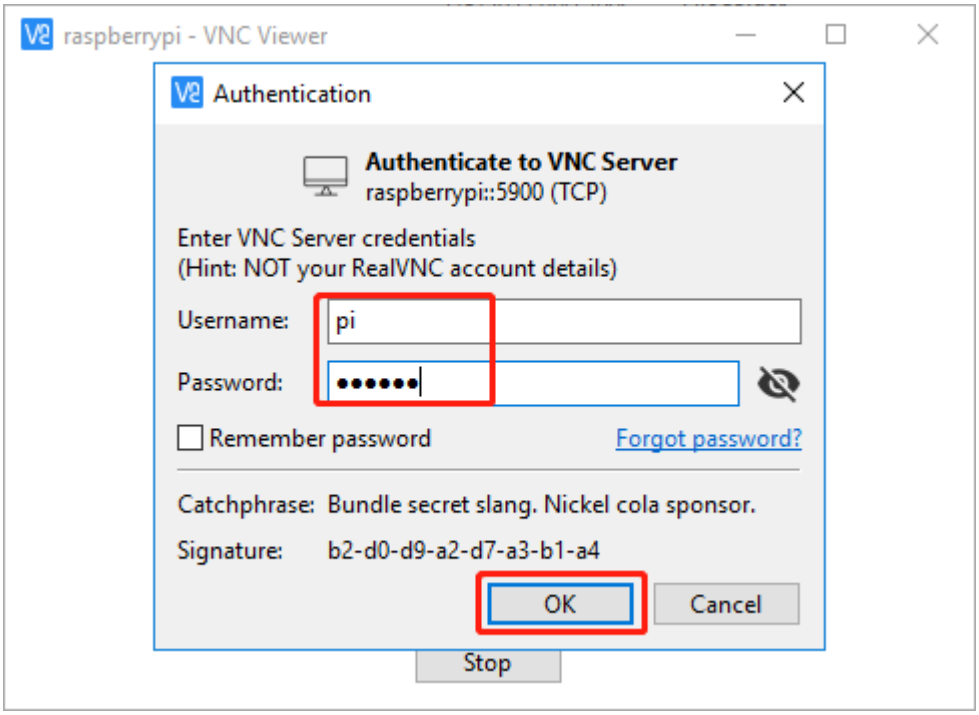


Logging in via VNC Viewer

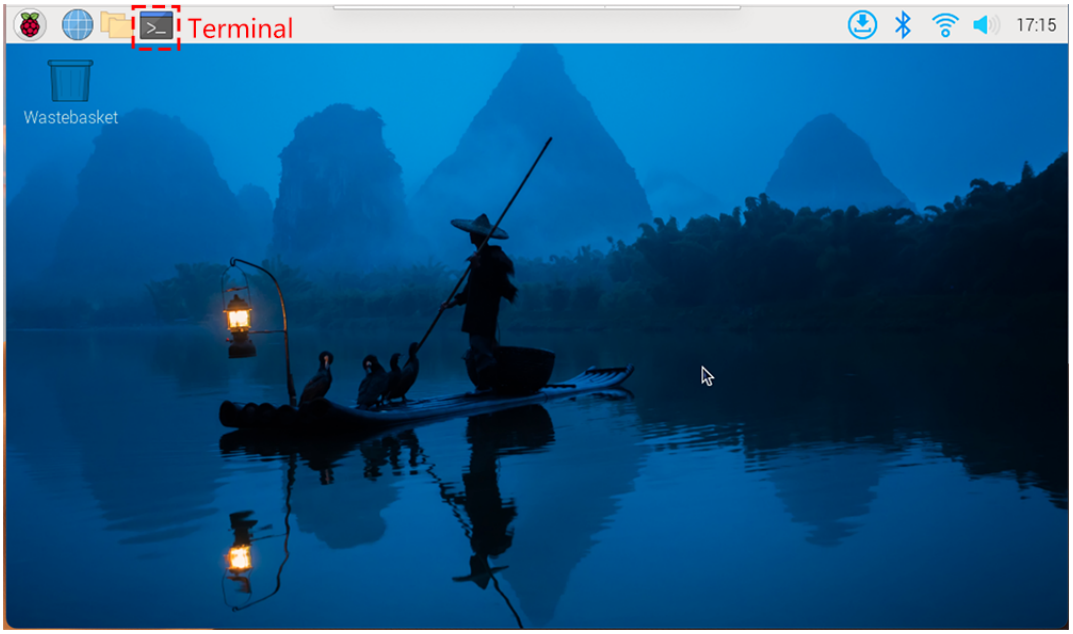
1. Download and install VNC Viewer on your personal computer.
2. Once installed, launch VNC Viewer. Enter the hostname or IP address of your Raspberry Pi and press Enter.



3. When prompted, enter your Raspberry Pi's username and password, then click **OK**.



4. After a few seconds, the Raspberry Pi OS desktop will be displayed. Now you can open the Terminal to start entering commands.



Note: Hello, welcome to the SunFounder Raspberry Pi & Arduino & ESP32 Enthusiasts Community on Facebook! Dive deeper into Raspberry Pi, Arduino, and ESP32 with fellow enthusiasts.

Why Join?

- **Expert Support**: Solve post-sale issues and technical challenges with help from our community and team.
- **Learn & Share**: Exchange tips and tutorials to enhance your skills.

- **Exclusive Previews**: Get early access to new product announcements and sneak peeks.
- **Special Discounts**: Enjoy exclusive discounts on our newest products.
- **Festive Promotions and Giveaways**: Take part in giveaways and holiday promotions.

Ready to explore and create with us? Click [] and join today!

1.1.5 5. Install All the Modules (Important)

Make sure you are connected to the Internet and update your system:

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

Note: Python3 related packages must be installed if you are installing the Lite version OS.

```
sudo apt install git python3-pip python3-setuptools python3-smbus
```

Install `robot-hat` module.

```
cd ~/
git clone -b v2.0 https://github.com/sunfounder/robot-hat.git
cd robot-hat
sudo python3 setup.py install
```

Then download the code and install `vilib` module.

```
cd ~/
git clone -b picamera2 https://github.com/sunfounder/vilib.git
cd vilib
sudo python3 install.py
```

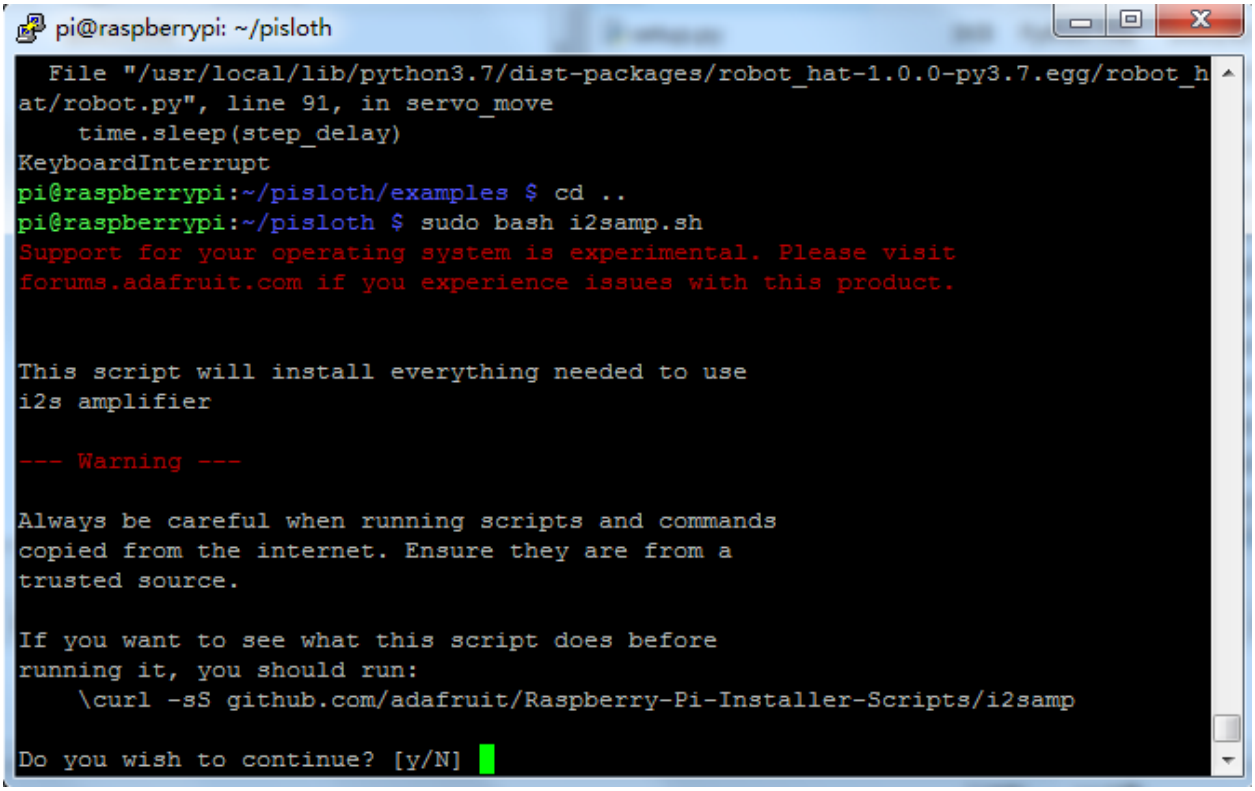
Then download the code and install `picrawler` module.

```
cd ~/
git clone https://github.com/sunfounder/picrawler.git --depth 1
cd picrawler
sudo python3 setup.py install
```

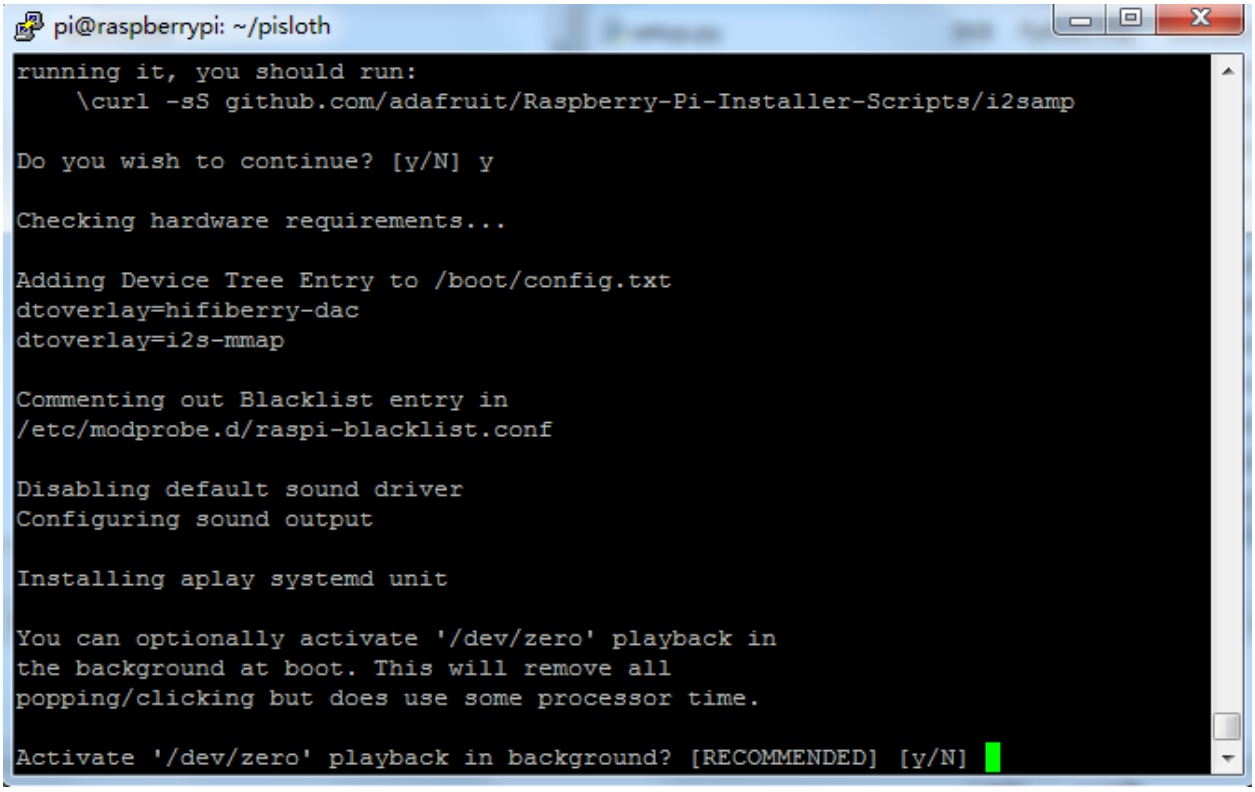
This step will take a little time, so please be patient.

Finally, you need to run the script `i2samp.sh` to install the components required by the i2s amplifier, otherwise the pislot will have no sound.

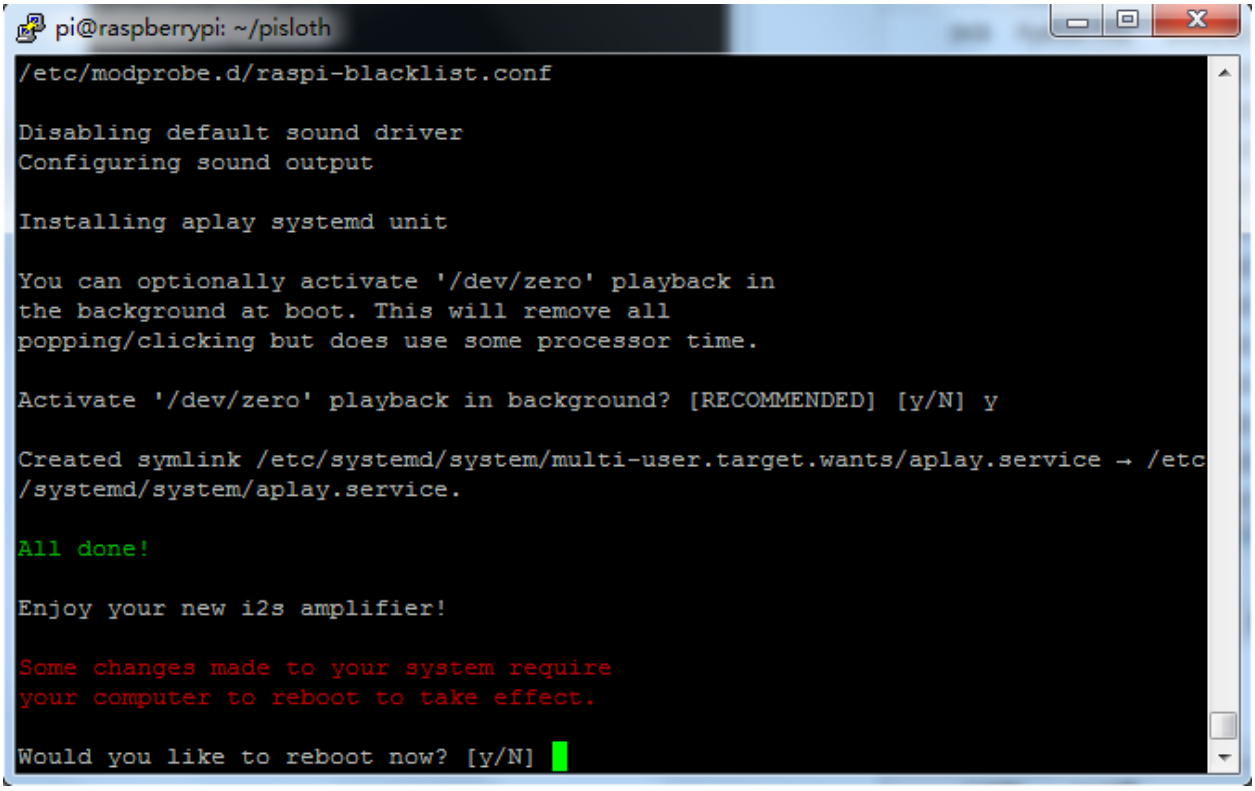
```
cd ~/picrawler
sudo bash i2samp.sh
```



Type `y` and press `Enter` to continue running the script.



Type `y` and press `Enter` to run `/dev/zero` in the background.



Type `y` and press `Enter` to restart the machine.

Note: If there is no sound after restarting, you may need to run the `i2samp.sh` script multiple times.

Note: Hello, welcome to the SunFounder Raspberry Pi & Arduino & ESP32 Enthusiasts Community on Facebook! Dive deeper into Raspberry Pi, Arduino, and ESP32 with fellow enthusiasts.

Why Join?

- **Expert Support**: Solve post-sale issues and technical challenges with help from our community and team.
- **Learn & Share**: Exchange tips and tutorials to enhance your skills.
- **Exclusive Previews**: Get early access to new product announcements and sneak peeks.
- **Special Discounts**: Enjoy exclusive discounts on our newest products.
- **Festive Promotions and Giveaways**: Take part in giveaways and holiday promotions.

Ready to explore and create with us? Click [] and join today!

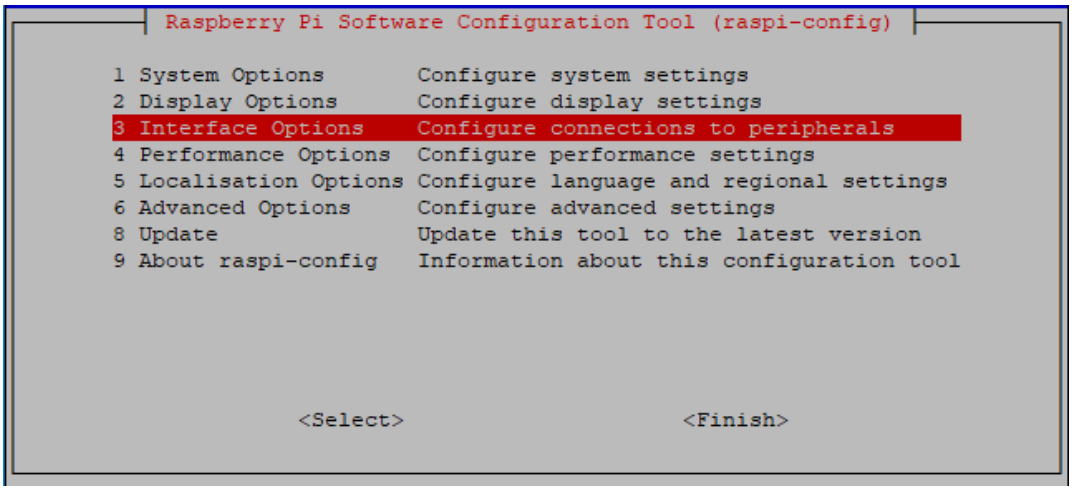
1.1.6 6. Check I2C Interface

We will be using the Raspberry Pi's I2C interface. This interface should have been enabled when installing the `robot-hat` module earlier. To ensure everything is in order, let's check if it is indeed enabled.

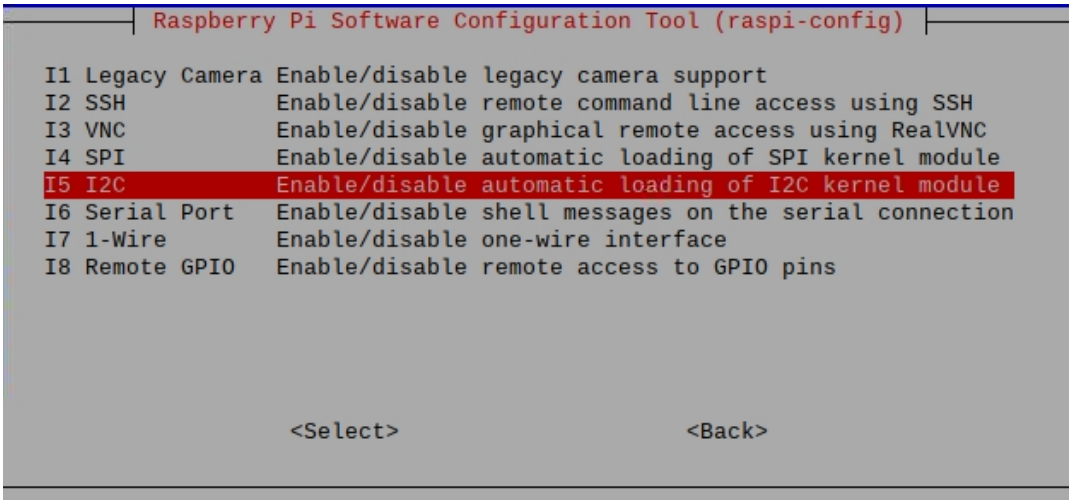
1. Input the following command:

```
sudo raspi-config
```

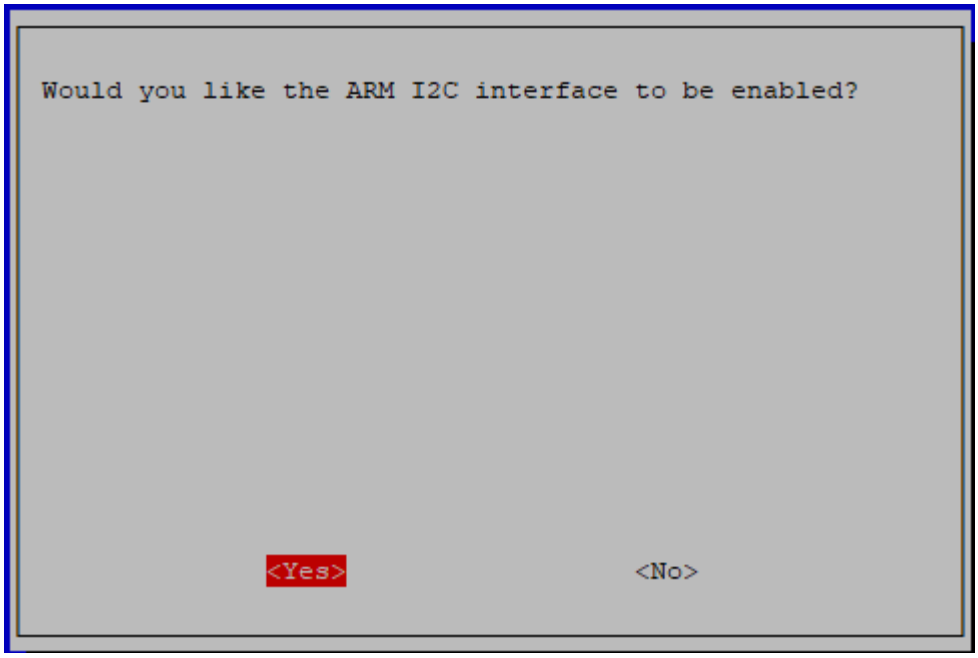
2. Choose **Interfacing Options** by press the down arrow key on your keyboard, then press the **Enter** key.



3. Then **I2C**.



4. Use the arrow keys on the keyboard to select **<Yes>** -> **<OK>** to complete the setup of the I2C.



Note: Hello, welcome to the SunFounder Raspberry Pi & Arduino & ESP32 Enthusiasts Community on Facebook! Dive deeper into Raspberry Pi, Arduino, and ESP32 with fellow enthusiasts.

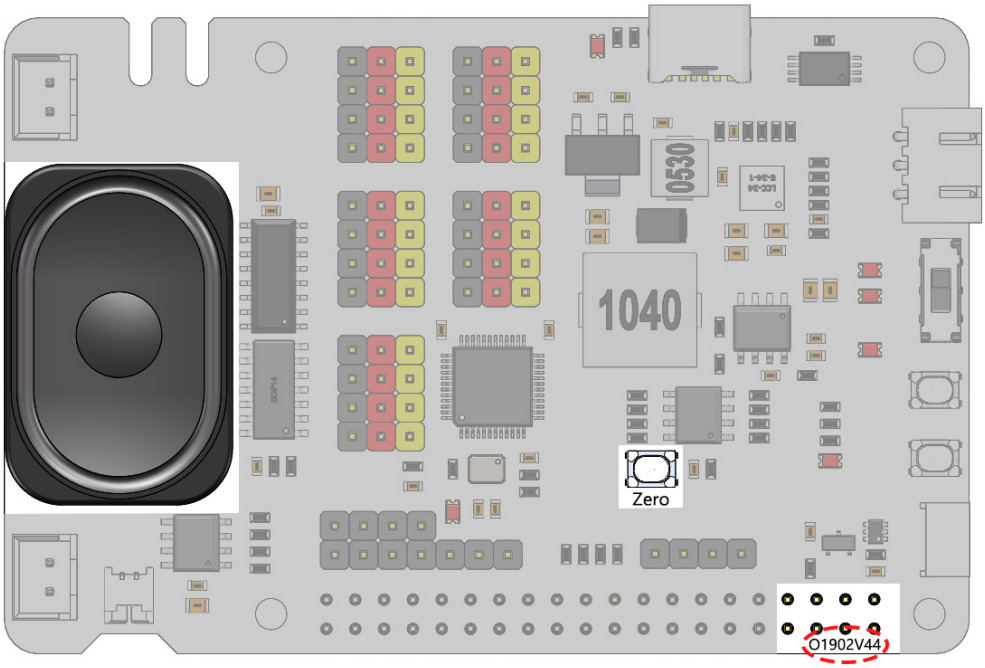
Why Join?

- **Expert Support**: Solve post-sale issues and technical challenges with help from our community and team.
- **Learn & Share**: Exchange tips and tutorials to enhance your skills.
- **Exclusive Previews**: Get early access to new product announcements and sneak peeks.
- **Special Discounts**: Enjoy exclusive discounts on our newest products.
- **Festive Promotions and Giveaways**: Take part in giveaways and holiday promotions.

Ready to explore and create with us? Click [] and join today!

1.1.7 7. Servo Adjust (Important)

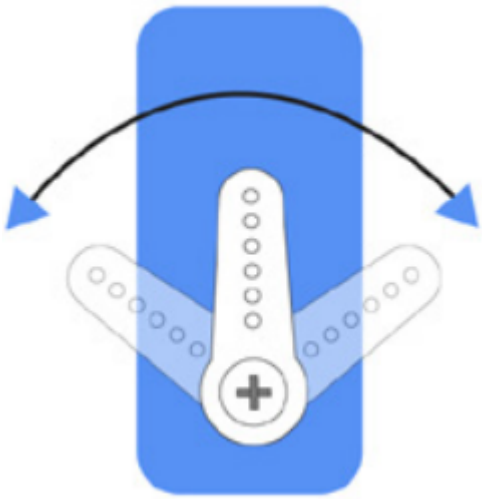
Note: If your Robot HAT is version V44 or higher (with the speaker located at the top of the board) and includes an onboard **Zero** button, you can skip this step and simply press the **Zero** button to activate the servo zeroing program.



The angle range of the servo is -90~90, but the angle set at the factory is random, maybe 0°, maybe 45°; if we assemble it with such an angle directly, it will lead to a chaotic state after the robot runs the code, or worse, it will cause the servo to block and burn out.

So here we need to set all the servo angles to 0° and then install them, so that the servo angle is in the middle, no matter which direction to turn.

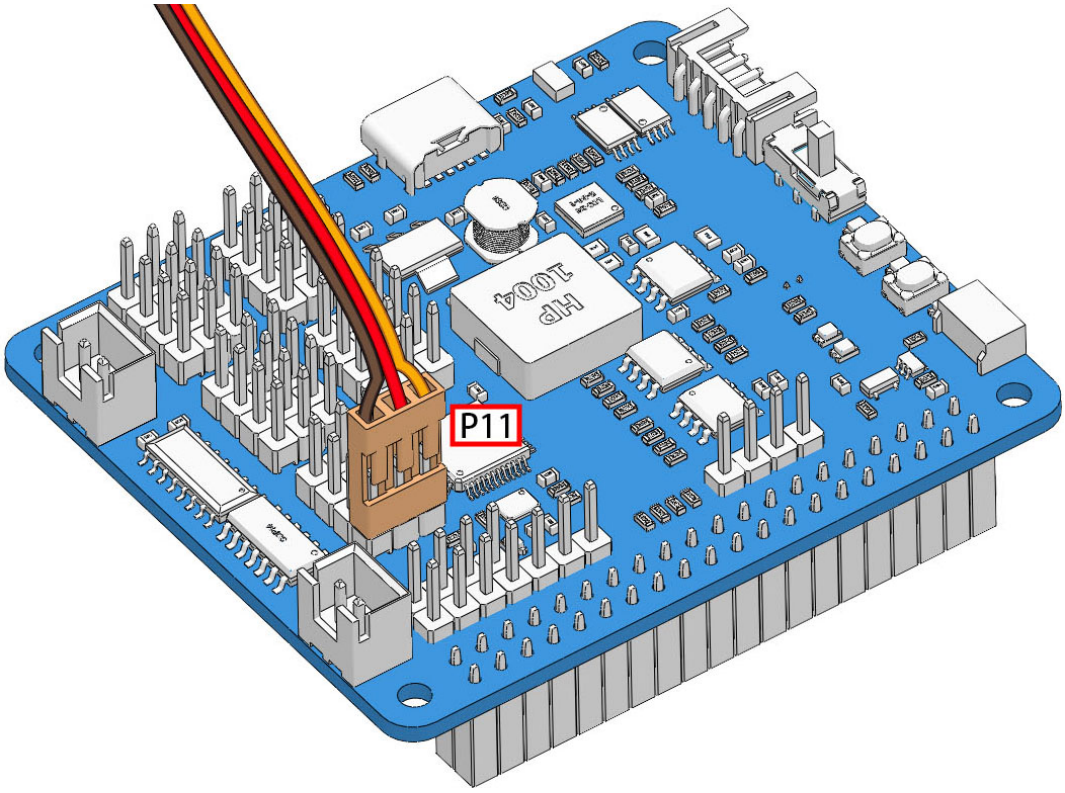
1. To ensure that the servo has been properly set to 0°, first insert the servo arm into the servo shaft and then gently rotate the rocker arm to a different angle. This servo arm is just to allow you to clearly see that the servo is rotating.



2. Now, run `servo_zeroing.py` in the `examples/` folder.

```
cd ~/picrawler/examples
sudo python3 servo_zeroing.py
```

3. Next, plug the servo cable into the P11 port as follows, at the same time you will see the servo arm rotate to a position(This is the 0° position, which is a random location and may not be vertical or parallel.).



4. Now, remove the servo arm, ensuring the servo wire remains connected, and do not turn off the power. Then continue the assembly following the paper instructions.

Note:

- Do not unplug this servo cable before fixing it with the servo screw, you can unplug it after fixing it.
- Do not rotate the servo while it is powered on to avoid damage; if the servo shaft is not inserted at the right angle, pull the servo out and reinsert it.
- Before assembling each servo, you need to plug the servo cable into PWM pin and turn on the power to set its angle to 0°.

Note: Hello, welcome to the SunFounder Raspberry Pi & Arduino & ESP32 Enthusiasts Community on Facebook! Dive deeper into Raspberry Pi, Arduino, and ESP32 with fellow enthusiasts.

Why Join?

- **Expert Support**: Solve post-sale issues and technical challenges with help from our community and team.
- **Learn & Share**: Exchange tips and tutorials to enhance your skills.
- **Exclusive Previews**: Get early access to new product announcements and sneak peeks.

- **Special Discounts**: Enjoy exclusive discounts on our newest products.
- **Festive Promotions and Giveaways**: Take part in giveaways and holiday promotions.

Ready to explore and create with us? Click [] and join today!

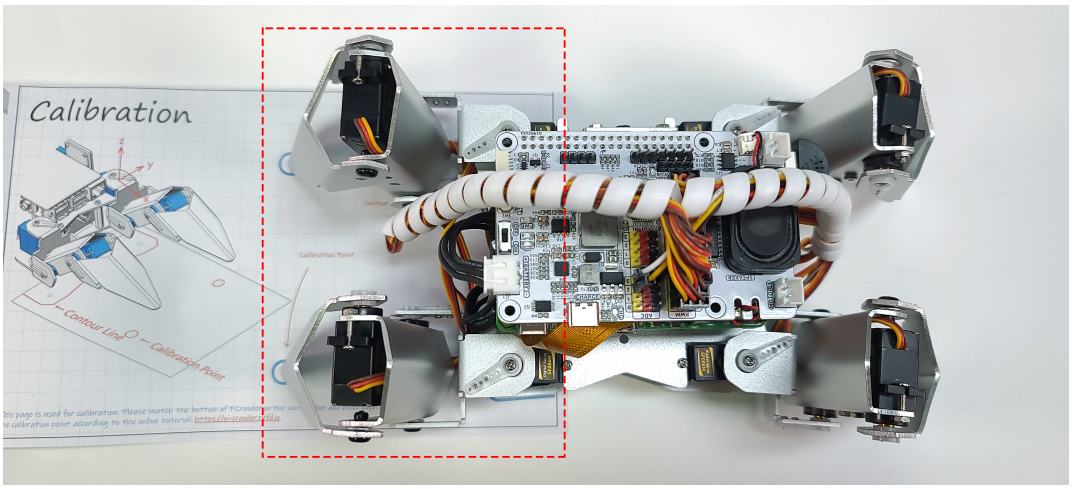
1.2 Calibrate the PiCrawler

Due to possible deviations during PiCrawler installation or limitations of the servos themselves, some servo angles may be slightly tilted, so you can calibrate them.

Of course you can skip this chapter if you think the assembly is perfect and doesn't require calibration.

The specific steps are as follows:

1. Take out the assembly leaflet, turn it to the last page, and lay it flat on the table. Then place the PiCrawler as shown below, aligning its bottom with the outline on the calibration chart.

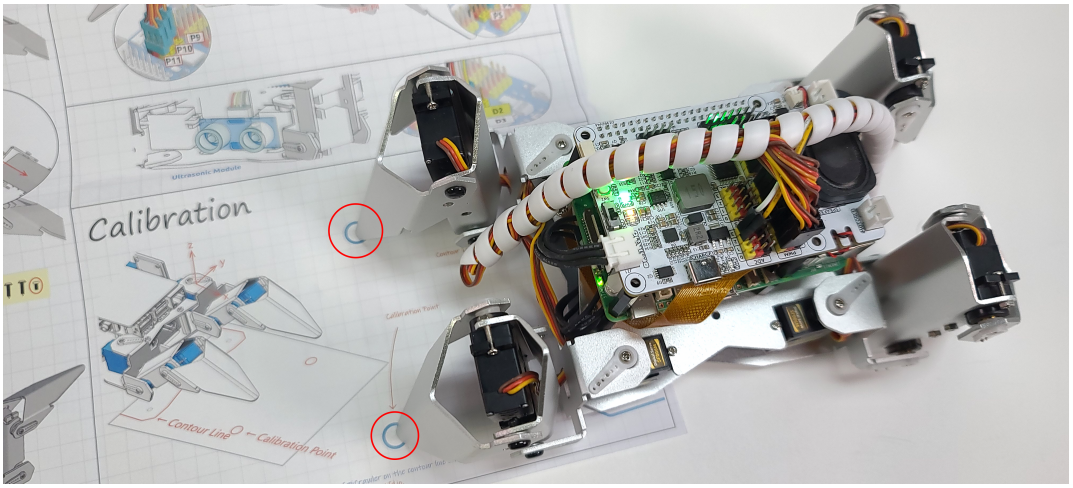


2. Run the `calibration.py`.

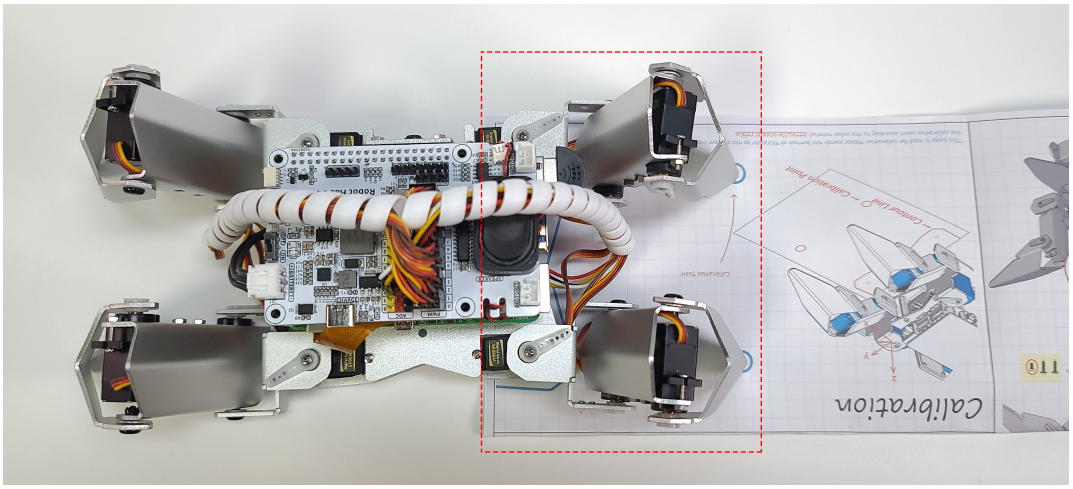
```
cd ~/picrawler/examples/calibration
sudo python3 calibration.py
```

After running the above code, you will see the following interface displayed in the terminal.

3. Press `2` and `3` keys respectively to choose left 2 legsthen press `w`, `a`, `s`, `d`, `r`, and `f` keys to move them to the calibration point.



4. Now, change the calibration paper to the right and press the `1` and `4` keys to choose right 2 legs, then press `w`, `a`, `s`, `d`, `r`, and `f` keys to move them to the calibration point.



5. After the calibration is completed, press the `space` key to save, you will be prompted to enter `Y` to confirm, and then `ctrl+c` to exit the program to complete the calibration.

After the assembly is complete, you can try to run the projects below.

Note: Hello, welcome to the SunFounder Raspberry Pi & Arduino & ESP32 Enthusiasts Community on Facebook! Dive deeper into Raspberry Pi, Arduino, and ESP32 with fellow enthusiasts.

Why Join?

- **Expert Support**: Solve post-sale issues and technical challenges with help from our community and team.
- **Learn & Share**: Exchange tips and tutorials to enhance your skills.
- **Exclusive Previews**: Get early access to new product announcements and sneak peeks.
- **Special Discounts**: Enjoy exclusive discounts on our newest products.
- **Festive Promotions and Giveaways**: Take part in giveaways and holiday promotions.

Ready to explore and create with us? Click [] and join today!

1.3 Move

This is PiCrawler's first project. Perform its most basic function - move.

Run the Code

```
cd ~/picrawler/examples
sudo python3 move.py
```

After the code is executed, PiCrawler will perform the following actions in sequence: move forward, move backward, turn left, turn right, stand.

Code

Note: You can **Modify/Reset/Copy/Run/Stop** the code below. But before that, you need to go to source code path like `pisloth\examples`. After modifying the code, you can run it directly to see the effect.

```
from picrawler import Picrawler
from time import sleep

crawler = Picrawler()

def main():

    speed = 80

    while True:

        crawler.do_action('forward',2,speed)
        sleep(0.05)
        crawler.do_action('backward',2,speed)
        sleep(0.05)
        crawler.do_action('turn left',2,speed)
        sleep(0.05)
        crawler.do_action('turn right',2,speed)
        sleep(0.05)
        crawler.do_action('turn left angle',2,speed)
        sleep(0.05)
        crawler.do_action('turn right angle',2,speed)
        sleep(0.05)
        crawler.do_step('stand',speed)
        sleep(1)

if __name__ == "__main__":
    main()
```

How it works?

First, import the `Picrawler` class from the `picrawler` library you have installed, which contains all of PiCrawler's actions and the functions that implement them.

```
from picrawler import Picrawler
```

Then instantiate the `crawler` class.

```
crawler = Picrawler()
```

Finally use the `crawler.do_action()` function to make Pisloth move.

```
crawler.do_action('forward',2,speed)
crawler.do_action('backward',2,speed)
crawler.do_action('turn left',2,speed)
crawler.do_action('turn right',2,speed)
crawler.do_action('turn left angle',2,speed)
crawler.do_action('turn right angle',2,speed)
```

In general, all movement of PiCrawler can be implemented with the `do_action()` function. It has 3 parameters:

- `motion_name` is the name of specific actions, including: `forward`, `turn right`, `turn left`, `backward`, `turn left angle`, `turn right angle`.
- `step` represents the number of each action is done, the default is 1.
- `speed` indicates the speed of the action, the default is 50 and the range is 0~100.

In addition, `crawler.do_step('stand',speed)` is also used here to make PiCrawler stand. The usage of this function will be explained in the following example.

Note: Hello, welcome to the SunFounder Raspberry Pi & Arduino & ESP32 Enthusiasts Community on Facebook! Dive deeper into Raspberry Pi, Arduino, and ESP32 with fellow enthusiasts.

Why Join?

- **Expert Support**: Solve post-sale issues and technical challenges with help from our community and team.
- **Learn & Share**: Exchange tips and tutorials to enhance your skills.
- **Exclusive Previews**: Get early access to new product announcements and sneak peeks.
- **Special Discounts**: Enjoy exclusive discounts on our newest products.
- **Festive Promotions and Giveaways**: Take part in giveaways and holiday promotions.

Ready to explore and create with us? Click [] and join today!

1.4 Keyboard Control

In this project, we will learn how to use the keyboard to remotely control the PiCrawler. You can control the PiCrawler to move forward, backward, left, and right.

Run the Code

```
cd ~/picrawler/examples
sudo python3 keyboard_control.py
```

Press keys on keyboard to control PiCrawler!

- `w`: Forward

- a: Turn left
- s: Backward
- d: Turn right
- Ctrl+C: Quit

Code

```
from picrawler import Picrawler
from time import sleep
import readchar

crawler = Picrawler()
speed = 90

manual = '''
Press keys on keyboard to control PiCrawler!
    W: Forward
    A: Turn left
    S: Backward
    D: Turn right

    Ctrl^C: Quit
'''

def show_info():
    print("\033[H\033[J",end='')  # clear terminal windows
    print(manual)


def main():
    show_info()
    while True:
        key = readchar.readkey()
        key = key.lower()
        if key in('wsad'):
            if 'w' == key:
                crawler.do_action('forward',1,speed)
            elif 's' == key:
                crawler.do_action('backward',1,speed)
            elif 'a' == key:
                crawler.do_action('turn left',1,speed)
            elif 'd' == key:
                crawler.do_action('turn right',1,speed)
            sleep(0.05)
            show_info()
        elif key == readchar.key.CTRL_C:
            print("\n Quit")
            break

        sleep(0.02)
```

(continues on next page)

(continued from previous page)

```
if __name__ == "__main__":
    main()
```

How it works?

PiCrawler should take appropriate action based on the keyboard characters read. The `lower()` function converts upper case characters into lower case characters, so that the letter remains valid regardless of case.

```
def main():
    show_info()
    while True:
        key = readchar.readkey()
        key = key.lower()
        if key in('wsad'):
            if 'w' == key:
                crawler.do_action('forward',1,speed)
            elif 's' == key:
                crawler.do_action('backward',1,speed)
            elif 'a' == key:
                crawler.do_action('turn left',1,speed)
            elif 'd' == key:
                crawler.do_action('turn right',1,speed)
            sleep(0.05)
            show_info()
        elif key == readchar.key.CTRL_C:
            print("\n Quit")
            break

        sleep(0.02)
```

Note: Hello, welcome to the SunFounder Raspberry Pi & Arduino & ESP32 Enthusiasts Community on Facebook! Dive deeper into Raspberry Pi, Arduino, and ESP32 with fellow enthusiasts.

Why Join?

- **Expert Support**: Solve post-sale issues and technical challenges with help from our community and team.
- **Learn & Share**: Exchange tips and tutorials to enhance your skills.
- **Exclusive Previews**: Get early access to new product announcements and sneak peeks.
- **Special Discounts**: Enjoy exclusive discounts on our newest products.
- **Festive Promotions and Giveaways**: Take part in giveaways and holiday promotions.

Ready to explore and create with us? Click [] and join today!

1.5 Sound Effect

In this example, we use PiCrawler's (to be precise, Robot HAT's) sound effects. It consists of three parts, namely **Muisc**, **Sound**, **Text to Speech**.

Install i2samp

Before using that functions, first activate the speaker so that it will be enabled and can make sounds.

Run `i2samp.sh` in, and this script will install everything needed to use i2s amplifier.

```
cd ~/picrawler/
sudo bash i2samp.sh
```

There will be several prompts asking to confirm the request. Respond to all prompts with a **Y**. After the changes have been made to the Raspberry Pi system, the computer will need to reboot for these changes to take effect.

After rebooting, run the `i2samp.sh` script again to test the amplifier. If a sound successfully plays from the speaker, the configuration is complete.

Run the Code

```
cd ~/picrawler/examples
sudo python3 sound_effect.py
```

After the code runs, please operate according to the prompt that printed on the terminal.

Input key to call the function! * `q`: Play background music * `1`: Play sound effect * `2`: Play sound effect with threads * `t`: Text to speak * If you want to exit the program, press `Ctrl+C`.

Code

```
'''
    Sorry, currently there is only sound when running with sudo
'''

from time import sleep
from robot_hat import Music,TTS

music = Music()
tts = TTS()

manual = '''
Input key to call the function!
    q: Play background music
    1: Play sound effect
    2: Play sound effect with threads
    t: Text to speak

    Ctrl^C: quit
'''

def main():
    print(manual)

    flag_bgm = False
    music.music_set_volume(20)
```

(continues on next page)

(continued from previous page)

```
    tts.lang("en-US")


    while True:
        key = input()
        key = key.lower()
        if key == "q":
            flag_bgm = not flag_bgm
            if flag_bgm is True:
                music.music_play('./musics/sports-Ahjay_Stelino.mp3')
            else:
                music.music_stop()

        elif key == "1":
            music.sound_play('./sounds/talk1.wav')
            sleep(0.05)
            music.sound_play('./sounds/talk3.wav')
            sleep(0.05)
            music.sound_play('./sounds/sign.wav')
            sleep(0.5)

        elif key =="2":
            music.sound_play_threading('./sounds/talk1.wav')
            sleep(0.05)
            music.sound_play_threading('./sounds/talk3.wav')
            sleep(0.05)
            music.sound_play_threading('./sounds/sign.wav')
            sleep(0.5)

        elif key == "t":
            words = "Hello"
            tts.say(words)

if __name__ == "__main__":
    main()
```

How it works?

Functions related to background music include these:

- `music = Music()` : Declare the object.
- `music.music_set_volume(20)` : Set the volume, the range is 0~100.
- `music.music_play(./musics/sports-Ahjay_Stelino.mp3)` : Play music files, here is the **sports-Ahjay_Stelino.mp3** file under the `./musics` path.
- `music.music_stop()` : Stop playing background music.

Note: You can add different sound effects or music to `musics` or `sounds` folder via *Filezilla Software*.

Functions related to sound effects include these:

- `music = Music()`

- `music.sound_play('./sounds/talk1.wav')`: Play the sound effect file, here is the **talk1.wav** file under the `./musics` path.
- `music.sound_play_threading('./sounds/talk1.wav')`: Play the sound effect file in a new thread mode without suspending the main thread.

Functions related to Text to Speech include these:

- `tts = TTS()`
- `tts.say(words)` : Text audio.
- `tts.lang("en-US")` : Set the language.

Note: Set the language by setting the parameters of `lang("")` with the following characters.

Table 1: Language

zh-CN	Mandarin (Chinese)
en-US	English-United States
en-GB	English-United Kingdom
de-DE	Germany-Deutsch
es-ES	España-Español
fr-FR	France-Le français
it-IT	Italia-lingua italiana

Note: Hello, welcome to the SunFounder Raspberry Pi & Arduino & ESP32 Enthusiasts Community on Facebook! Dive deeper into Raspberry Pi, Arduino, and ESP32 with fellow enthusiasts.

Why Join?

- **Expert Support**: Solve post-sale issues and technical challenges with help from our community and team.
- **Learn & Share**: Exchange tips and tutorials to enhance your skills.
- **Exclusive Previews**: Get early access to new product announcements and sneak peeks.
- **Special Discounts**: Enjoy exclusive discounts on our newest products.
- **Festive Promotions and Giveaways**: Take part in giveaways and holiday promotions.

Ready to explore and create with us? Click [] and join today!

1.6 Obstacle Avoidance

In this project, picrawler will use an ultrasonic module to detect obstacles in front. When PiCrawler detects an obstacle, it will send a signal and look for another direction to move forward.

Run the Code

```
cd ~/picrawler/examples
sudo python3 avoid.py
```

After the code runs, PiCrawler will walk forward. If it detects that the distance of the obstacle ahead is less than 10cm, it will stop and sound a warning, then turn left and stop. If there is no obstacle in the direction after turning left or the obstacle distance is greater than 10, it will continue to move forward.

Code

Note: You can **Modify/Reset/Copy/Run/Stop** the code below. But before that, you need to go to source code path like `picrawler\examples`. After modifying the code, you can run it directly to see the effect.

```
from picrawler import Picrawler
from robot_hat import TTS, Music
from robot_hat import Ultrasonic
from robot_hat import Pin
import time

tts = TTS()
music = Music()

crawler = Picrawler()
sonar = Ultrasonic(Pin("D2") ,Pin("D3"))
music.music_set_volume(100)

alert_distance = 15
speed = 80

def main():
    distance = sonar.read()
    print(distance)
    if distance < 0:
        pass
    elif distance <= alert_distance:
        try:
            music.sound_play_threading('./sounds/sign.wav', volume=100)
        except Exception as e:
            print(e)
        crawler.do_action('turn left angle',3,speed)
        time.sleep(0.2)
    else :
        crawler.do_action('forward', 1,speed)
        time.sleep(0.2)

if __name__ == "__main__":
    while True:
        main()
```

How it works?

You can get the distance by importing the `Ultrasonic` class.

```
from robot_hat import Ultrasonic
```

Then initialize the ultrasonic pins.

```
sonar = Ultrasonic(Pin("D2") ,Pin("D3"))
```

Here is the main program.

- Read the `distance` detected by ultrasonic module and filter out the values less than 0 (When the ultrasonic module is too far from the obstacle or cannot read the data correctly, `distance<0` will appear).
- When the `distance` is less than or equal to `alert_distance` (the threshold value set earlier, which is 10), play the sound effect `sign.wav`. PiCrawler does `turn left angle` .
- When the `distance` is greater than `alert_distance`, PiCrawler will move `forward`.

```
distance = sonar.read()
print(distance)
if distance < 0:
    pass
elif distance <= alert_distance:
    try:
        music.sound_play_threading('./sounds/sign.wav', volume=100)
    except Exception as e:
        print(e)
    crawler.do_action('turn left angle',3,speed)
    time.sleep(0.2)
else :
    crawler.do_action('forward', 1,speed)
    time.sleep(0.2)
```

Note: You can add different sound effects or music to `musics` or `sounds` folder via *Filezilla Software*.

Note: Hello, welcome to the SunFounder Raspberry Pi & Arduino & ESP32 Enthusiasts Community on Facebook! Dive deeper into Raspberry Pi, Arduino, and ESP32 with fellow enthusiasts.

Why Join?

- **Expert Support**: Solve post-sale issues and technical challenges with help from our community and team.
- **Learn & Share**: Exchange tips and tutorials to enhance your skills.
- **Exclusive Previews**: Get early access to new product announcements and sneak peeks.
- **Special Discounts**: Enjoy exclusive discounts on our newest products.
- **Festive Promotions and Giveaways**: Take part in giveaways and holiday promotions.

Ready to explore and create with us? Click [] and join today!

1.7 Computer Vision

This project will officially enter the field of computer vision!

Run the Code

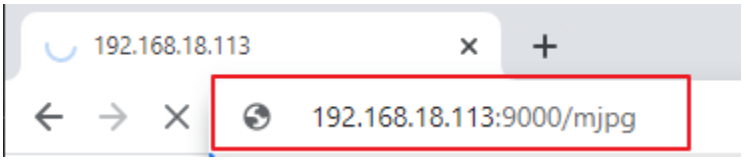
```
cd ~/picrawler/examples
sudo python3 display.py
```

View the Image

After the code runs, the terminal will display the following prompt:

```
No desktop !
* Serving Flask app "vilib.vilib" (lazy loading)
* Environment: production
WARNING: Do not use the development server in a production environment.
Use a production WSGI server instead.
* Debug mode: off
* Running on http://0.0.0.0:9000/ (Press CTRL+C to quit)
```

Then you can enter `http://<your IP>:9000/mjpg` in the browser to view the video screen. such as: `https://192.168.18.113:9000/mjpg`



After the program runs, you will see the following information in the final:

- Input key to call the function!
- `q`: Take photo
- `1`: Color detect : red
- `2`: Color detect : orange
- `3`: Color detect : yellow
- `4`: Color detect : green
- `5`: Color detect : blue
- `6`: Color detect : purple
- `0`: Switch off Color detect
- `r`: Scan the QR code
- `f`: Switch ON/OFF face detect
- `s`: Display detected object information

Please follow the prompts to activate the corresponding functions.

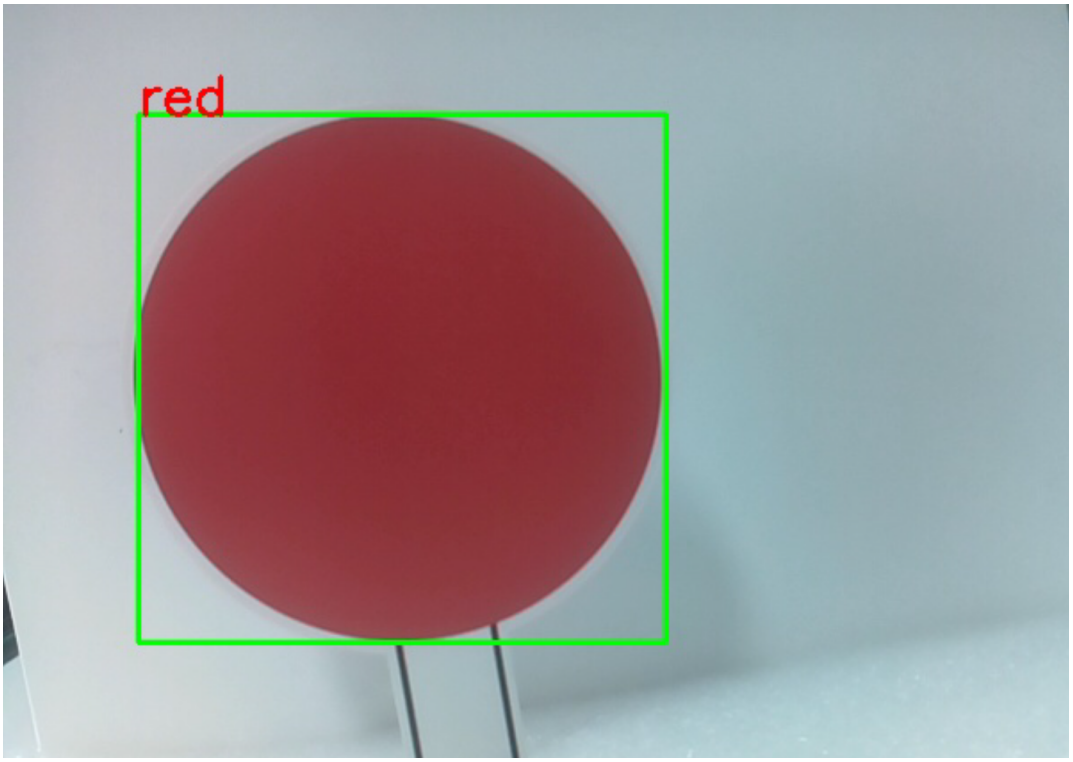
- **Take Photo**

  Type `q` in the terminal and press Enter. The picture currently seen by the camera will be saved (if the color detection function is turned on, the mark box will also appear in the saved picture). You can see

these photos from the `~/Pictures/PiCrawler/` directory of the Raspberry Pi. You can use tools such as *Filezilla Software* to transfer photos to your PC.

- **Color Detect**

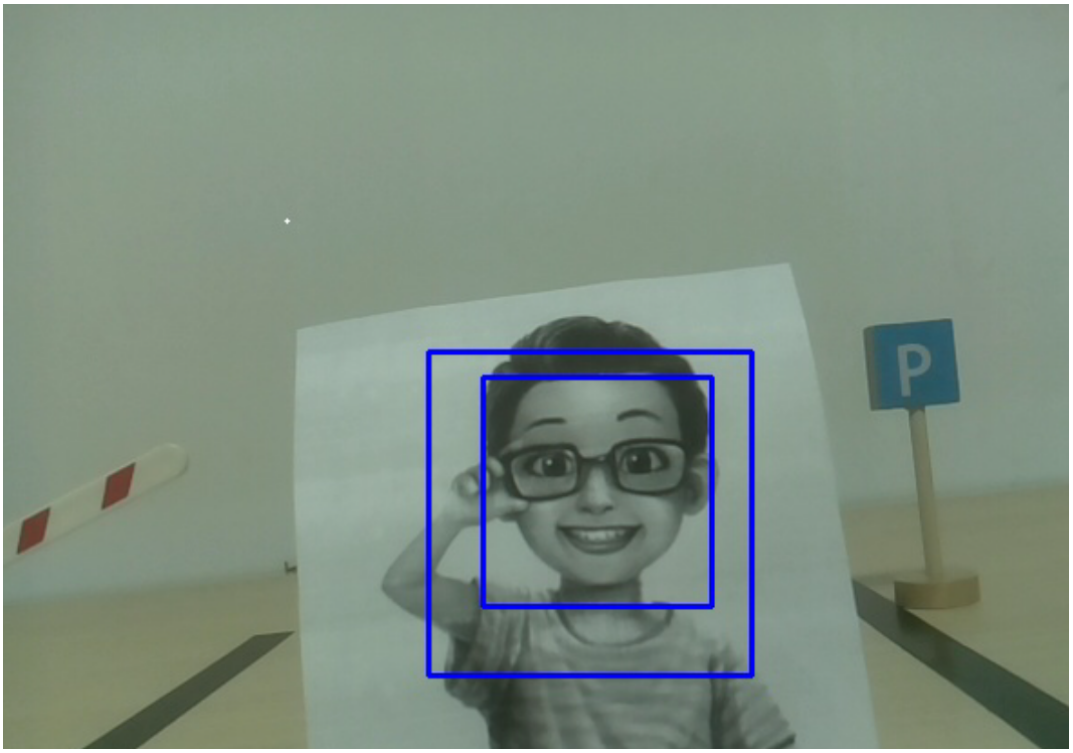
  Entering a number between `1~6` will detect one of the colors in "red, orange, yellow, green, blue, purple". Enter `0` to turn off color detection.



Note: You can download and print the `PDF Color Cards` for color detection.

- **Face Detect**

  Type `f` to turn on face detection.



- **QR Code Detect**

  Enter `r` to open the QR code recognition. No other operations can be performed before the QR code is recognized. The decoding information of the QR code will be printed in the terminal.

- **Display Information**

Entering s will print the information of the face detection (and color detection) target in the terminal. Including the center coordinates (X, Y) and size (Weight, height) of the measured object.

Code

```
from vilib import Vilib
from time import sleep, time, strftime, localtime
import threading
from os import getlogin

USERNAME = getlogin()
PICTURE_PATH = f"/home/{USERNAME}/Pictures/"


flag_face = False
flag_color = False
qr_code_flag = False

MANUAL = '''
Input key to call the function!
    q: Take photo
    1: Color detect : red
    2: Color detect : orange
    3: Color detect : yellow
    4: Color detect : green
    5: Color detect : blue
    6: Color detect : purple
    0: Switch off Color detect
    r: Scan the QR code
    f: Switch ON/OFF face detect
    s: Display detected object information
'''

color_list = ['close', 'red', 'orange', 'yellow',
        'green', 'blue', 'purple',
]

def face_detect(flag):
    print("Face Detect:" + str(flag))
    Vilib.face_detect_switch(flag)


def qrcode_detect():
    global qr_code_flag
    if qr_code_flag == True:
        Vilib.qrcode_detect_switch(True)
        print("Waitting for QR code")

    text = None
    while True:
        temp = Vilib.detect_obj_parameter['qr_data']
        if temp != "None" and temp != text:
            text = temp
            print('QR code:%s'%text)
```

(continues on next page)

(continued from previous page)

```
        if qr_code_flag == False:
            break
        sleep(0.5)
    Vilib.qrcode_detect_switch(False)


def take_photo():
    _time = strftime('%Y-%m-%d-%H-%M-%S',localtime(time()))
    name = 'photo_%s'%_time
    Vilib.take_photo(name, PICTURE_PATH)
    print('photo save as %s%s.jpg'%(PICTURE_PATH, name))


def object_show():
    global flag_color, flag_face

    if flag_color is True:
        if Vilib.detect_obj_parameter['color_n'] == 0:
            print('Color Detect: None')
        else:
            color_coodinate = (Vilib.detect_obj_parameter['color_x'],Vilib.detect_obj_
↪parameter['color_y'])
            color_size = (Vilib.detect_obj_parameter['color_w'],Vilib.detect_obj_
↪parameter['color_h'])
            print("[Color Detect] ","Coordinate:",color_coodinate,"Size",color_size)

    if flag_face is True:
        if Vilib.detect_obj_parameter['human_n'] == 0:
            print('Face Detect: None')
        else:
            human_coodinate = (Vilib.detect_obj_parameter['human_x'],Vilib.detect_obj_
↪parameter['human_y'])
            human_size = (Vilib.detect_obj_parameter['human_w'],Vilib.detect_obj_
↪parameter['human_h'])
            print("[Face Detect] ","Coordinate:",human_coodinate,"Size",human_size)


def main():
    global flag_face, flag_color, qr_code_flag
    qrcode_thread = None

    Vilib.camera_start(vflip=False,hflip=False)
    Vilib.display(local=True,web=True)
    print(MANUAL)

    while True:
        # readkey
        key = input()
        key = key.lower()
        # take photo
        if key == 'q':
            take_photo()
```

(continues on next page)

(continued from previous page)

```
        # color detect
        elif key != '' and key in ('0123456'):  # '' in ('0123') -> True
            index = int(key)
            if index == 0:
                flag_color = False
                Vilib.color_detect('close')
            else:
                flag_color = True
                Vilib.color_detect(color_list[index]) # color_detect(color:str -> color_
↪name/close)
            print('Color detect : %s'%color_list[index])
        # face detection
        elif key =="f":
            flag_face = not flag_face
            face_detect(flag_face)
        # qrcode detection
        elif key =="r":
            qr_code_flag = not qr_code_flag
            if qr_code_flag == True:
                if qrcode_thread == None or not qrcode_thread.is_alive():
                    qrcode_thread = threading.Thread(target=qrcode_detect)
                    qrcode_thread.setDaemon(True)
                    qrcode_thread.start()
            else:
                if qrcode_thread != None and qrcode_thread.is_alive():
                   # wait for thread to end
                    qrcode_thread.join()
                    print('QRcode Detect: close')
        # show detected object information
        elif key == "s":
            object_show()

        sleep(0.5)


if __name__ == "__main__":
    main()
```

How it works?

The first thing you need to pay attention to here is the following function. These two functions allow you to start the camera.

```
Vilib.camera_start()
Vilib.display()
```

Functions related to "object detection":

- `Vilib.face_detect_switch(True)` : Switch ON/OFF face detection
- `Vilib.color_detect(color)` : For color detection, only one color detection can be performed at the same time. The parameters that can be input are: `"red"`, `"orange"`, `"yellow"`, `"green"`, `"blue"`, `"purple"`
- `Vilib.color_detect_switch(False)` : Switch OFF color detection

- `Vilib.qrcode_detect_switch(False)` : Switch ON/OFF QR code detection, Returns the decoded data of the QR code.
- `Vilib.gesture_detect_switch(False)` : Switch ON/OFF gesture detection
- `Vilib.traffic_sign_detect_switch(False)` : Switch ON/OFF traffic sign detection

The information detected by the target will be stored in the `detect_obj_parameter = Manager().dict()` dictionary.

In the main program, you can use it like this:

```
Vilib.detect_obj_parameter['color_x']
```

The keys of the dictionary and their uses are shown in the following list:

- `color_x`: the x value of the center coordinate of the detected color block, the range is 0~320
- `color_y`: the y value of the center coordinate of the detected color block, the range is 0~240
- `color_w`: the width of the detected color block, the range is 0~320
- `color_h`: the height of the detected color block, the range is 0~240
- `color_n`: the number of detected color patches
- `human_x`: the x value of the center coordinate of the detected human face, the range is 0~320
- `human_y`: the y value of the center coordinate of the detected face, the range is 0~240
- `human_w`: the width of the detected human face, the range is 0~320
- `human_h`: the height of the detected face, the range is 0~240
- `human_n`: the number of detected faces
- `traffic_sign_x`: the center coordinate x value of the detected traffic sign, the range is 0~320
- `traffic_sign_y`: the center coordinate y value of the detected traffic sign, the range is 0~240
- `traffic_sign_w`: the width of the detected traffic sign, the range is 0~320
- `traffic_sign_h`: the height of the detected traffic sign, the range is 0~240
- `traffic_sign_t`: the content of the detected traffic sign, the value list is *['stop','right','left','forward']*
- `gesture_x`: The center coordinate x value of the detected gesture, the range is 0~320
- `gesture_y`: The center coordinate y value of the detected gesture, the range is 0~240
- `gesture_w`: The width of the detected gesture, the range is 0~320
- `gesture_h`: The height of the detected gesture, the range is 0~240
- `gesture_t`: The content of the detected gesture, the value list is *["paper","scissor","rock"]*
- `qr_date`: the content of the QR code being detected
- `qr_x`: the center coordinate x value of the QR code to be detected, the range is 0~320
- `qr_y`: the center coordinate y value of the QR code to be detected, the range is 0~240
- `qr_w`: the width of the QR code to be detected, the range is 0~320
- `qr_h`: the height of the QR code to be detected, the range is 0~320

Note: Hello, welcome to the SunFounder Raspberry Pi & Arduino & ESP32 Enthusiasts Community on Facebook! Dive deeper into Raspberry Pi, Arduino, and ESP32 with fellow enthusiasts.

Why Join?

- **Expert Support**: Solve post-sale issues and technical challenges with help from our community and team.
- **Learn & Share**: Exchange tips and tutorials to enhance your skills.
- **Exclusive Previews**: Get early access to new product announcements and sneak peeks.
- **Special Discounts**: Enjoy exclusive discounts on our newest products.
- **Festive Promotions and Giveaways**: Take part in giveaways and holiday promotions.

Ready to explore and create with us? Click [] and join today!

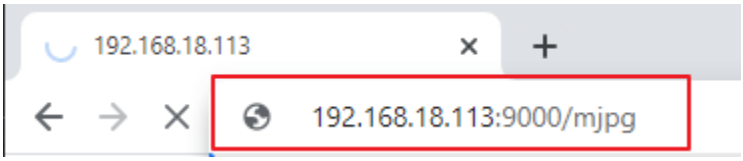
1.8 Record Video

This example will guide you how to use the recording function.

Run the Code

```
cd ~/picrawler/examples
sudo python3 record_video.py
```

After the code runs, you can enter `http://<your IP>:9000/mjpg` in the browser to view the video screen. such as: `http://192.168.18.113:9000/mjpg`



Recording can be stopped or started by pressing the keys on the keyboard.

- Press `q` to begin recording or pause/continue, `e` to stop recording or save.
- If you want to exit the program, press `Ctrl+C`.

Code

```
from time import sleep,strftime,localtime
from vilib import Vilib
import readchar
from os import getlogin

USERNAME = getlogin()
VIDEO_PATH = f"/home/{USERNAME}/Videos/"

MANUAL = '''
Press keys on keyboard to control recording:
    Q: record/pause/continue
    E: stop
    Ctrl^C: Quit
```

(continues on next page)

(continued from previous page)

```
'''

def print_overwrite(msg,  end='', flush=True):
    print('\r\033[2K', end='',flush=True)
    print(msg, end=end, flush=True)

def main():
    rec_flag = 'stop' # start,pause,stop
    vname = None
    Vilib.rec_video_set["path"] = VIDEO_PATH

    Vilib.camera_start(vflip=False,hflip=False)
    Vilib.display(local=True,web=True)
    sleep(0.8)  # wait for startup

    print(MANUAL)
    while True:
        # read keyboard
        key = readchar.readkey()
        key = key.lower()
        # start,pause
        if key == 'q':
            key = None
            if rec_flag == 'stop':
                rec_flag = 'start'
                # set name
                vname = strftime("%Y-%m-%d-%H.%M.%S", localtime())
                Vilib.rec_video_set["name"] = vname
                # start record
                Vilib.rec_video_run()
                Vilib.rec_video_start()
                print_overwrite('rec start ...')
            elif rec_flag == 'start':
                rec_flag = 'pause'
                Vilib.rec_video_pause()
                print_overwrite('pause')
            elif rec_flag == 'pause':
                rec_flag = 'start'
                Vilib.rec_video_start()
                print_overwrite('continue')
        # stop
        elif key == 'e' and rec_flag != 'stop':
            key = None
            rec_flag = 'stop'
            Vilib.rec_video_stop()
            print_overwrite("The video saved as %s%s.avi"%(Vilib.rec_video_set["path"],
↪vname),end='\n')
        # quit
        elif key == readchar.key.CTRL_C:
            Vilib.camera_close()
            print('\nquit')
            break
```

(continues on next page)

(continued from previous page)

```
        sleep(0.1)

if __name__ == "__main__":
    main()
```

How it works?

Functions related to recording include the following:

- `Vilib.rec_video_run(video_name)`: Started the thread to record the video. `video_name` is the name of the video file, it should be a string.
- `Vilib.rec_video_start()`: Start or continue video recording.
- `Vilib.rec_video_pause()`: Pause recording.
- `Vilib.rec_video_stop()`: Stop recording.

`Vilib.rec_video_set["path"] = "~/video/test/"` sets the storage location of video files.

Note: Hello, welcome to the SunFounder Raspberry Pi & Arduino & ESP32 Enthusiasts Community on Facebook! Dive deeper into Raspberry Pi, Arduino, and ESP32 with fellow enthusiasts.

Why Join?

- **Expert Support**: Solve post-sale issues and technical challenges with help from our community and team.
- **Learn & Share**: Exchange tips and tutorials to enhance your skills.
- **Exclusive Previews**: Get early access to new product announcements and sneak peeks.
- **Special Discounts**: Enjoy exclusive discounts on our newest products.
- **Festive Promotions and Giveaways**: Take part in giveaways and holiday promotions.

Ready to explore and create with us? Click [] and join today!

1.9 Bull Fight

Make PiCrawler an angry bull! Use its camera to track and rush the red cloth!

Run the Code

```
cd ~/picrawler/examples
sudo python3 bull_fight.py
```

View the Image

After the code runs, the terminal will display the following prompt:

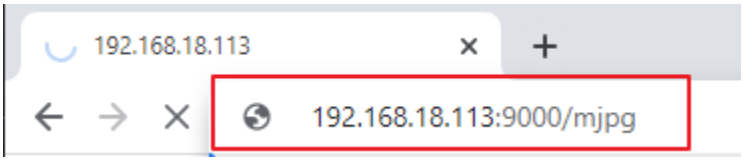
```
No desktop !
* Serving Flask app "vilib.vilib" (lazy loading)
* Environment: production
WARNING: Do not use the development server in a production environment.
Use a production WSGI server instead.
```

(continues on next page)

(continued from previous page)

```
* Debug mode: off
* Running on http://0.0.0.0:9000/ (Press CTRL+C to quit)
```

Then you can enter `http://<your IP>:9000/mjpg` in the browser to view the video screen. such as: `https://192.168.18.113:9000/mjpg`



Code

Note: You can **Modify/Reset/Copy/Run/Stop** the code below. But before that, you need to go to source code path like `picrawler\examples`. After modifying the code, you can run it directly to see the effect.

```
from picrawler import Picrawler
from time import sleep
from robot_hat import Music
from vilib import Vilib


crawler = Picrawler()

music = Music()

def main():
    Vilib.camera_start()
    Vilib.display()
    Vilib.color_detect("red")
    speed = 80

    while True:
        if Vilib.detect_obj_parameter['color_n']!=0:
            coordinate_x = Vilib.detect_obj_parameter['color_x']
            music.sound_play_threading('./sounds/talk1.wav')

            if coordinate_x < 100:
                crawler.do_action('turn left',1,speed)
                sleep(0.05)
            elif coordinate_x > 220:
                crawler.do_action('turn right',1,speed)
                sleep(0.05)
            else :
                crawler.do_action('forward',2,speed)
                sleep(0.05)
        else :
            crawler.do_step('stand',speed)
            sleep(0.05)
```

(continues on next page)

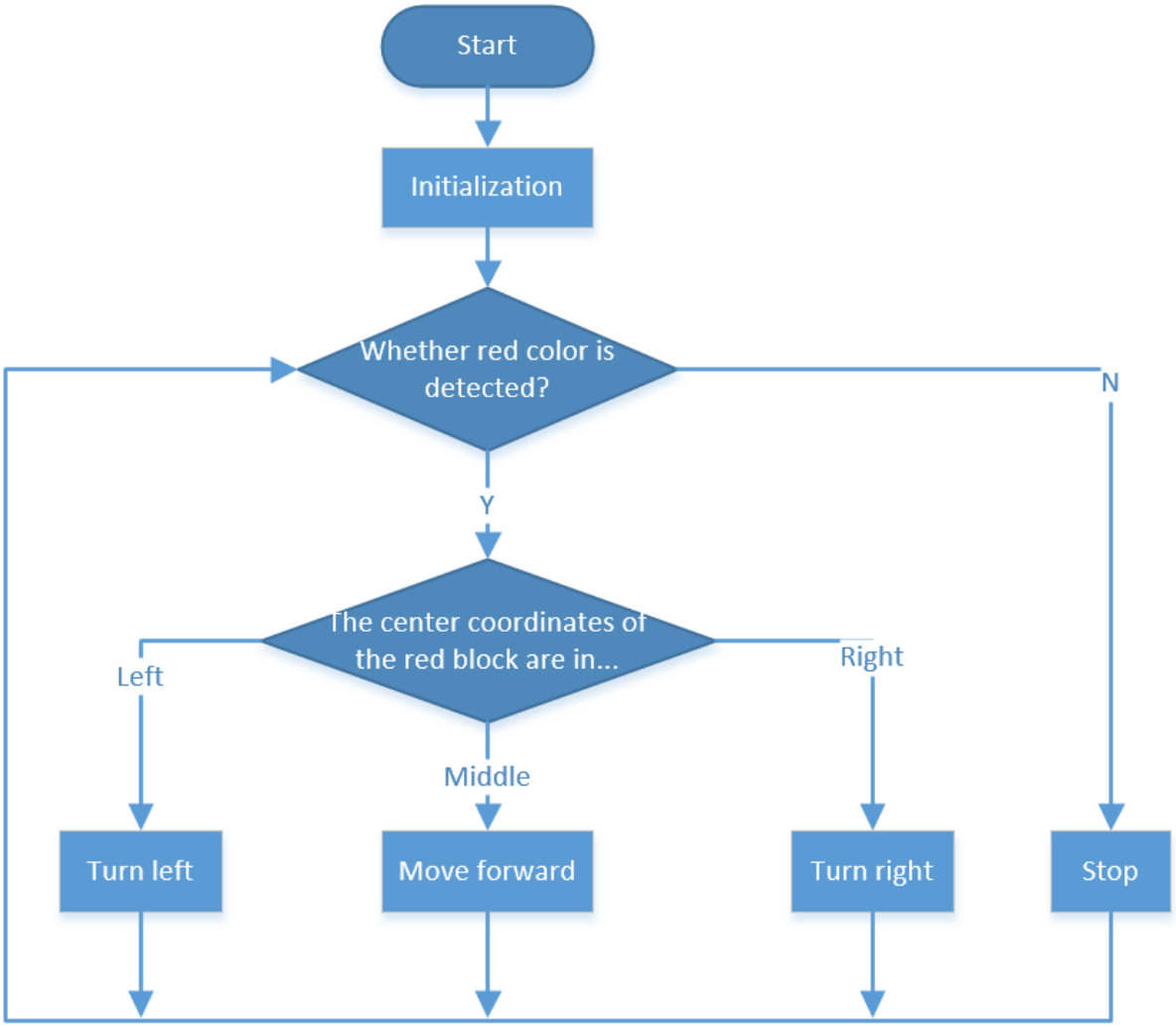
(continued from previous page)

```
if __name__ == "__main__":
    main()
```

How it works?

In general, this project combines the knowledge points of *Move*, *Computer Vision* and *Sound Effect*.

Its flow is shown in the figure below:



Note: Hello, welcome to the SunFounder Raspberry Pi & Arduino & ESP32 Enthusiasts Community on Facebook! Dive deeper into Raspberry Pi, Arduino, and ESP32 with fellow enthusiasts.

Why Join?

- **Expert Support**: Solve post-sale issues and technical challenges with help from our community and team.
- **Learn & Share**: Exchange tips and tutorials to enhance your skills.
- **Exclusive Previews**: Get early access to new product announcements and sneak peeks.
- **Special Discounts**: Enjoy exclusive discounts on our newest products.

- **Festive Promotions and Giveaways**: Take part in giveaways and holiday promotions.

Ready to explore and create with us? Click [] and join today!

1.10 Treasure Hunt

Arrange a maze in your room and place six different color cards in six corners. Then control PiCrawler to search for these color cards one by one!

Note: You can download and print the `PDF Color Cards` for color detection.

Run the Code

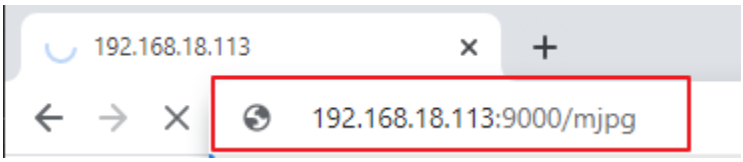
```
cd ~/picrawler/examples
sudo python3 treasure_hunt.py
```

View the Image

After the code runs, the terminal will display the following prompt:

```
No desktop !
* Serving Flask app "vilib.vilib" (lazy loading)
* Environment: production
WARNING: Do not use the development server in a production environment.
Use a production WSGI server instead.
* Debug mode: off
* Running on http://0.0.0.0:9000/ (Press CTRL+C to quit)
```

Then you can enter `http://<your IP>:9000/mjpg` in the browser to view the video screen. such as: `http://192.168.18.113:9000/mjpg`



Code

```
from picrawler import Picrawler
from time import sleep
from robot_hat import Music,TTS
from vilib import Vilib
import readchar
import random
import threading

crawler = Picrawler()


music = Music()
tts = TTS()
```

(continues on next page)

(continued from previous page)

```
manual = '''
Press keys on keyboard to control Picrawler!
    w: Forward
    a: Turn left
    s: Backward
    d: Turn right
    space: Say the target again
    Ctrl^C: Quit
'''

color = "red"
color_list=["red","orange","yellow","green","blue","purple"]
key_dict = {
    'w': 'forward',
    's': 'backward',
    'a': 'turn_left',
    'd': 'turn_right',
}
def renew_color_detect():
    global color
    color = random.choice(color_list)
    Vilib.color_detect(color)
    tts.say("Look for " + color)

key = None
lock = threading.Lock()
def key_scan_thread():
    global key
    while True:
        key_temp = readchar.readkey()
        print('\r',end='')
        with lock:
            key = key_temp.lower()
            if key == readchar.key.SPACE:
                key = 'space'
            elif key == readchar.key.CTRL_C:
                key = 'quit'
                break
        sleep(0.01)

def main():
    global key
    action = None
    Vilib.camera_start(vflip=False,hflip=False)
    Vilib.display(local=False,web=True)
    sleep(0.8)
    speed = 80
    print(manual)

    sleep(1)
    _key_t = threading.Thread(target=key_scan_thread)
```

(continues on next page)

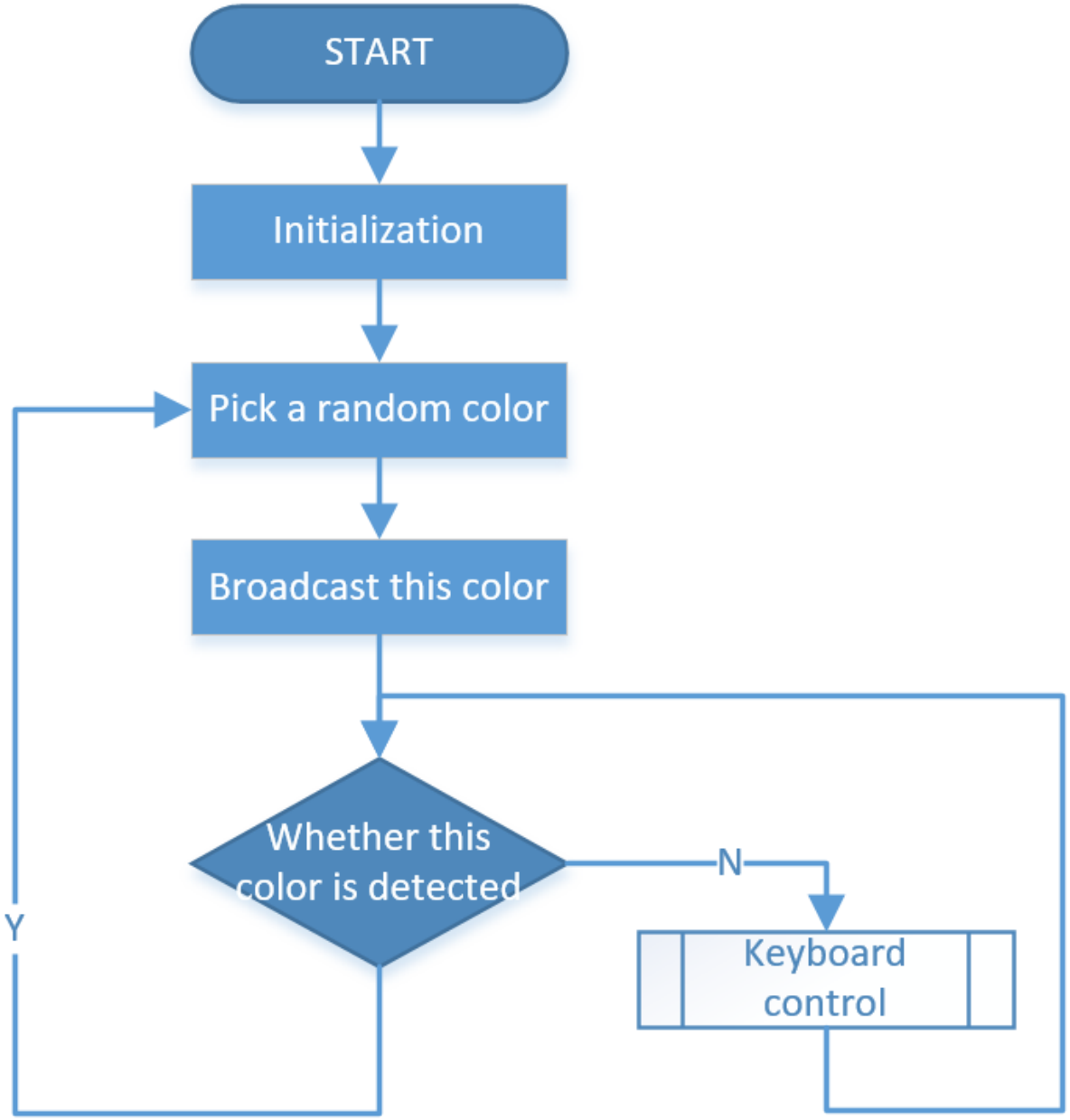
(continued from previous page)

```
    _key_t.setDaemon(True)
    _key_t.start()

    tts.say("game start")
    sleep(0.05)
    renew_color_detect()
    while True:

        if Vilib.detect_obj_parameter['color_n']!=0 and Vilib.detect_obj_parameter[
↪'color_w']>100:
            tts.say("will done")
            sleep(0.05)
            renew_color_detect()

        with lock:
            if key != None and key in ('wsad'):
                action = key_dict[str(key)]
                key =  None
            elif key == 'space':
                tts.say("Look for " + color)
                key =  None
            elif key == 'quit':
                _key_t.join()
                Vilib.camera_close()
                print("\n\rQuit")
                break

        if action != None:
            crawler.do_action(action,1,speed)
            action = None

        sleep(0.05)


if __name__ == "__main__":
    main()
```

How it works?

In general, this project combines the knowledge points of *Keyboard Control*, *Computer Vision* and *Sound Effect*.

Its flow is shown in the figure below:



Note: Hello, welcome to the SunFounder Raspberry Pi & Arduino & ESP32 Enthusiasts Community on Facebook! Dive deeper into Raspberry Pi, Arduino, and ESP32 with fellow enthusiasts.

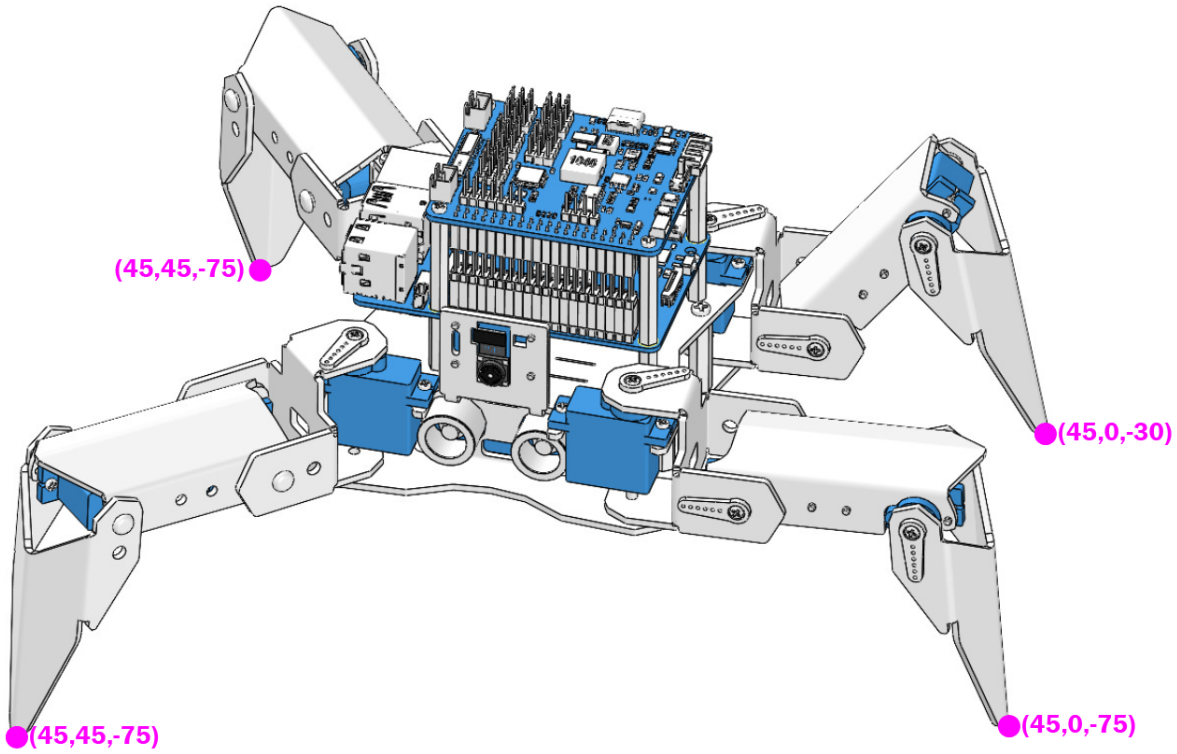
Why Join?

- **Expert Support**: Solve post-sale issues and technical challenges with help from our community and team.
- **Learn & Share**: Exchange tips and tutorials to enhance your skills.
- **Exclusive Previews**: Get early access to new product announcements and sneak peeks.
- **Special Discounts**: Enjoy exclusive discounts on our newest products.
- **Festive Promotions and Giveaways**: Take part in giveaways and holiday promotions.

Ready to explore and create with us? Click [] and join today!

1.11 Pose

PiCrawler can assume a specific posture by writing a coordinate array. Here it assumes a raised right rear foot posture.



Run the Code

```
cd ~/picrawler/examples
sudo python3 do_step.py
```

Code

```
from picrawler import Picrawler
from time import sleep

crawler = Picrawler()

## [right front],[left front],[left rear],[right rear]
new_step=[[45, 45, -75], [45, 0, -75], [45, 0, -30], [45, 45, -75]]
stand_step = crawler.move_list['stand'][0]

def main():
    while True:
        speed = 80
```

(continues on next page)

(continued from previous page)

```
        print(f"stand step: {stand_step}")
        crawler.do_step(stand_step, speed)
        sleep(3)
        print(f"new step: {new_step}")
        crawler.do_step(new_step,speed)
        sleep(3)


if __name__ == "__main__":
    main()
```

How it works?

In this code, the code you need to pay attention to is this `crawler.do_step()`.

Similar to `do_action()`, `do_step()` can also manipulate PiCrawler's behavior. The difference is that the former can perform the continuous behavior of `move forward`, while the latter can be used to make separate gestures of `stand` and `sit`.

It has two uses:

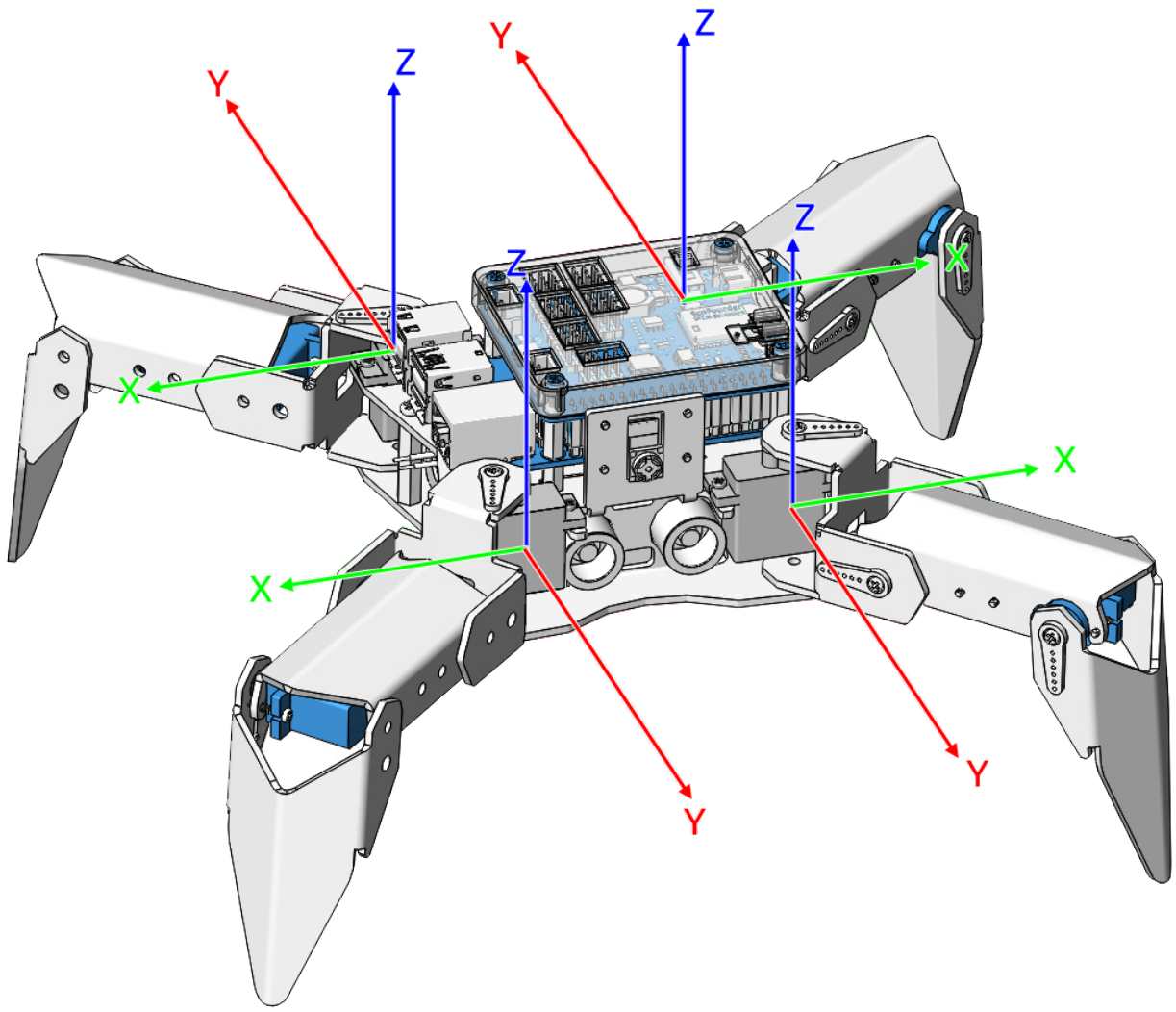
One: It can write strings, directly use the `step_list` dictionary in the `picrawler` library.

```
crawler.do_step('stand',speed)
# "speed" indicates the speed of the step, the range is 0~100.
```

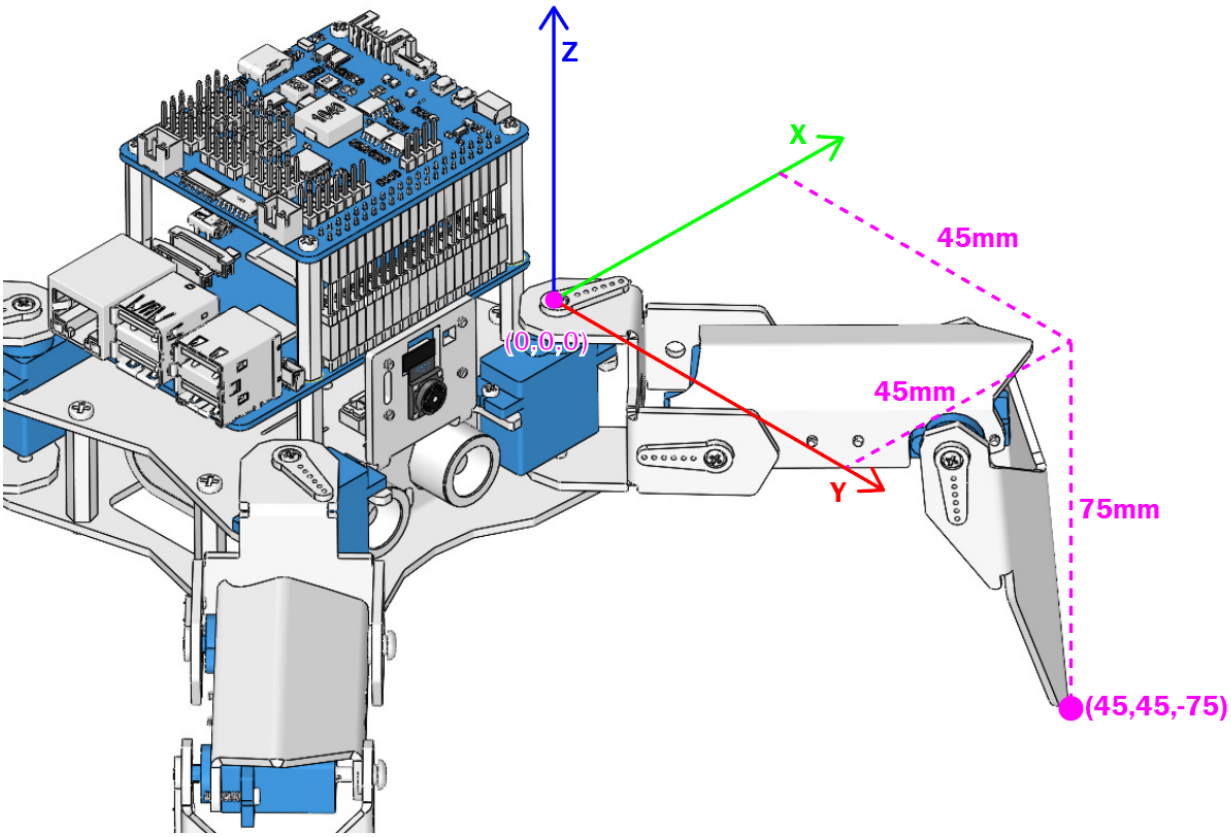
Second: It can also write an array of 4 coordinate values.

```
new_step=[[45, 45, -75], [45, 0, -75], [45, 0, -30], [45, 45, -75]]
# These four coordinates are used to control the four legs of right front, left front,␣
↪left rear, and left rear respectively.
```

Each foot has an independent coordinate system. As shown below:



You need to measure the coordinates of each toe individually. As shown below:



By the way: the `step_list` called in the first method also consists of an array containing 4 coordinate values.

```
step_list = {

    "stand":[
        [45, 45, -50],
        [45, 45, -50],
        [45, 45, -50],
        [45, 45, -50]
    ],
    "sit":[
        [45, 45, -30],
        [45, 45, -30],
        [45, 45, -30],
        [45, 45, -30]
    ],

}
```

Note: Hello, welcome to the SunFounder Raspberry Pi & Arduino & ESP32 Enthusiasts Community on Facebook! Dive deeper into Raspberry Pi, Arduino, and ESP32 with fellow enthusiasts.

Why Join?

- **Expert Support**: Solve post-sale issues and technical challenges with help from our community and team.

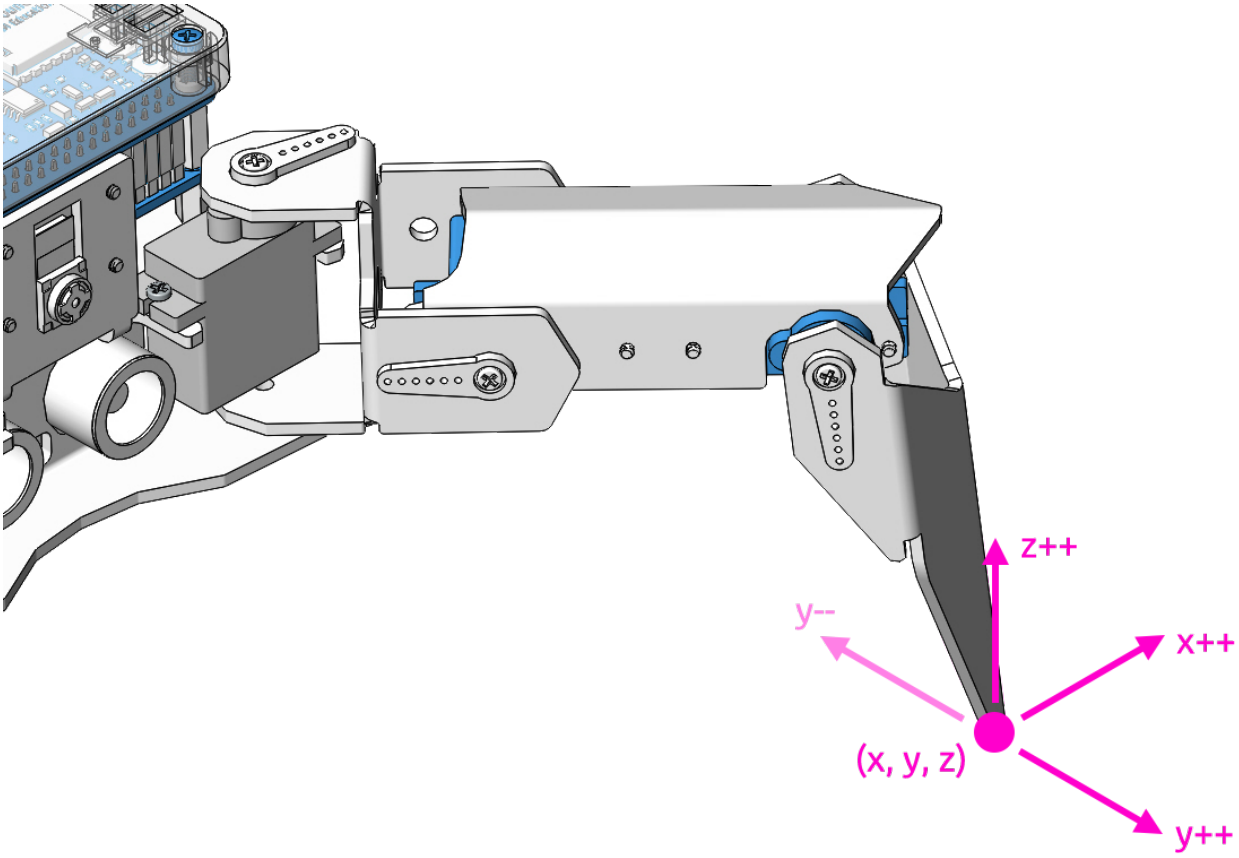
- **Learn & Share**: Exchange tips and tutorials to enhance your skills.
- **Exclusive Previews**: Get early access to new product announcements and sneak peeks.
- **Special Discounts**: Enjoy exclusive discounts on our newest products.
- **Festive Promotions and Giveaways**: Take part in giveaways and holiday promotions.

Ready to explore and create with us? Click [] and join today!

1.12 Adjust Posture

In this example, we use the keyboard to control the PiCrawler foot by foot and assume the desired posture.

You can press the space bar to print out the current coordinate values. These coordinate values come in handy when you create unique actions for PiCrawler.



Run the Code

```
cd ~/picrawler/examples
sudo python3 do_single_leg.py
```

After the code runs, please operate according to the prompt that pops up in the terminal.

- Press `1234` to select the feet separately, `1`: right front foot, `2`: left front foot, `3`: left rear foot, `4`: right rear foot
- Press `w`, `a`, `s`, `d`, `r`, and `f` to slowly control the PiCrawler's coordinate values.

- Press Ctrl+C to exit.

Code

```
from picrawler import Picrawler
from time import sleep
import readchar

crawler = Picrawler()
speed = 80


manual = '''
-------- PiCrawler Controller ---------
       .......           .......
    <=|   2   |--|   1   |=>
       `````````  |       `````````
                  |
       .... | ....
    <=| 3 | |-------| 4 |=>
       `````````   `````````
    1: Select right front leg
    2: Select left front leg
    3: Select left rear leg
    4: Select right rear leg

    W: Y++   R: Z++
    A: X--   F: Z--
    S: Y--
    D: X++   Ctrl+C: Quit
'''
legs_list = ['right front', 'left front', 'left rear', 'right rear']

def main():
    leg = 0
    speed = 80
    step = 2
    print(manual)
    crawler.do_step('stand', speed)
    sleep(0.2)
    coordinate=crawler.current_step_all_leg_value()

    def show_info():
        print("\033[H\033[J", end='') # clear terminal windows
        print(manual)
        print('%s : %s'%(leg+1, legs_list[leg]))
        print('coordinate: %s'%(coordinate))

    show_info()

    while True:
        # readkey
        key = readchar.readkey()
        key = key.lower()
        # select the leg
```

(continues on next page)

(continued from previous page)

```
        if key in ('1234'):
            leg = int(key) - 1
            show_info()
        # move
        elif key in ('wsadrf'):
            if 'w' == key:
                coordinate[leg][1]=coordinate[leg][1] + step
            elif 's' == key:
                coordinate[leg][1]=coordinate[leg][1] - step
            elif 'a' == key:
                coordinate[leg][0]=coordinate[leg][0] - step
            elif 'd' == key:
                coordinate[leg][0]=coordinate[leg][0] + step
            elif 'r' == key:
                coordinate[leg][2]=coordinate[leg][2] + step
            elif 'f' == key:
                coordinate[leg][2]=coordinate[leg][2] - step

            crawler.do_single_leg(leg,coordinate[leg],speed)
            sleep(0.1)
            # coordinate=crawler.current_step_all_leg_value()
            show_info()

        sleep(0.05)


if __name__ == "__main__":
    main()
```

- `current_step_all_leg_value()`: Returns the coordinate values of all legs.
- `do_single_leg(leg,coordinate[leg],speed)`: Modify the coordinate value of a certain leg individually.

Note: Hello, welcome to the SunFounder Raspberry Pi & Arduino & ESP32 Enthusiasts Community on Facebook! Dive deeper into Raspberry Pi, Arduino, and ESP32 with fellow enthusiasts.

Why Join?

- **Expert Support**: Solve post-sale issues and technical challenges with help from our community and team.
- **Learn & Share**: Exchange tips and tutorials to enhance your skills.
- **Exclusive Previews**: Get early access to new product announcements and sneak peeks.
- **Special Discounts**: Enjoy exclusive discounts on our newest products.
- **Festive Promotions and Giveaways**: Take part in giveaways and holiday promotions.

Ready to explore and create with us? Click [] and join today!

1.13 Record New Step

We use the keyboard to control PiCrawler to make several poses in turn, and record these poses. Replay them later.

Run the Code

```
cd ~/picrawler/examples
sudo python3 record_new_step_by_keyboard.py
```

After the code runs, please operate according to the prompt that pops up in the terminal.

- Press `1234` to select the feet separately, `1`: right front foot, `2`: left front foot, `3`: left rear foot, `4`: right rear foot
- Press `w`, `a`, `s`, `d`, `r`, and `f` to slowly control the PiCrawler's coordinate values.
- Press `space` to print all coordinate values.
- Press `p` to have PiCrawler replay the recorded action.
- Press `esc` to exit.

Code

```
from picrawler import Picrawler
from time import sleep
import sys
import tty
import termios
import copy

crawler = Picrawler()
speed = 80

def readchar():
    fd = sys.stdin.fileno()
    old_settings = termios.tcgetattr(fd)
    try:
        tty.setraw(sys.stdin.fileno())
        ch = sys.stdin.read(1)
    finally:
        termios.tcsetattr(fd, termios.TCSADRAIN, old_settings)
    return ch


manual = '''
Press keys on keyboard to control!
    w: Y++
    a: X--
    s: Y--
    d: X++
    r: Z++
    f: Z--
    1: Select right front leg
    2: Select left front leg
    3: Select left rear leg
    4: Select right rear leg
```

(continues on next page)

(continued from previous page)

```
    Space: Print all leg coodinate & Save this step
    p: Play all saved step
    esc: Quit
'''


new_step=[]

def save_new_step():
    new_step.append(copy.deepcopy(crawler.current_step_all_leg_value()))
    print(new_step)

def play_all_new_step():
    for step in new_step:
        crawler.do_step(step,speed)
        sleep(0.6)

def main():

    speed = 80
    print(manual)
    crawler.do_step('sit',speed)
    leg = 0
    coodinate=crawler.current_step_leg_value(leg)
    while True:
        key = readchar()
        key = key.lower()
        # print(key)
        if 'w' == key:
            coodinate[1]=coodinate[1]+2
        elif 's' == key:
            coodinate[1]=coodinate[1]-2
        elif 'a' == key:
            coodinate[0]=coodinate[0]-2
        elif 'd' == key:
            coodinate[0]=coodinate[0]+2
        elif 'r' == key:
            coodinate[2]=coodinate[2]+2
        elif 'f' == key:
            coodinate[2]=coodinate[2]-2
        elif '1' == key:
            leg=0
            coodinate=crawler.current_step_leg_value(leg)
        elif '2' == key:
            leg=1
            coodinate=crawler.current_step_leg_value(leg)
        elif '3' == key:
            leg=2
            coodinate=crawler.current_step_leg_value(leg)
        elif '4' == key:
            leg=3
            coodinate=crawler.current_step_leg_value(leg)
```

(continues on next page)

(continued from previous page)

```
        elif chr(32) == key:
            print("[[right front],[left front],[left rear],[right rear]]")
            print("saved new step")
            print(crawler.current_step_all_leg_value())
            save_new_step()
        elif 'p' == key:
            play_all_new_step()
        elif chr(27) == key:# 27 for ESC
            break

        sleep(0.05)
        crawler.do_single_leg(leg,coodinate,speed)
    print("\n q Quit")


if __name__ == "__main__":
    main()
```

How it works?

This project was born out of *Adjust Posture*. Added recording and replay functions.

The recording function is implemented by the following code.

```
new_step=[]

def save_new_step():
    new_step.append(copy.deepcopy(crawler.current_step_all_leg_value()))
    print(new_step)
```

Note: The assignment here needs to use the Deep Copy function, otherwise the `new_step` will not get a new array object when appending.

The replay function is implemented by the following code.

```
def play_all_new_step():
    for step in new_step:
        crawler.do_step(step,speed)
        sleep(0.6)
```

Note: Hello, welcome to the SunFounder Raspberry Pi & Arduino & ESP32 Enthusiasts Community on Facebook! Dive deeper into Raspberry Pi, Arduino, and ESP32 with fellow enthusiasts.

Why Join?

- **Expert Support**: Solve post-sale issues and technical challenges with help from our community and team.
- **Learn & Share**: Exchange tips and tutorials to enhance your skills.
- **Exclusive Previews**: Get early access to new product announcements and sneak peeks.
- **Special Discounts**: Enjoy exclusive discounts on our newest products.
- **Festive Promotions and Giveaways**: Take part in giveaways and holiday promotions.

Ready to explore and create with us? Click [] and join today!

1.14 Twist

We already know how to make PiCrawler assume a specific pose, the next step is to combine the poses to form a continuous action.

Here, PiCrawler's four feet are up and down in twos, jumping with the music.

Run the Code

```
cd ~/picrawler/examples
sudo python3 twist.py
```

Code

Note: You can **Modify/Reset/Copy/Run/Stop** the code below. But before that, you need to go to source code path like `picrawler\examples`. After modifying the code, you can run it directly to see the effect.

```
from picrawler import Picrawler
from robot_hat import Music

music = Music()
crawler = Picrawler()


def twist(speed):
    new_step=[[50, 50, -80], [50, 50, -80],[50, 50, -80], [50, 50, -80]]
    for i in range(4):
        for inc in range(30, 60, 5):
            rise = [50,50,(-80+inc*0.5)]
            drop = [50,50,(-80-inc)]

            new_step[i]=rise
            new_step[(i+2)%4] = drop
            new_step[(i+1)%4] = rise
            new_step[(i-1)%4] = drop
            # print(new_step)
            crawler.do_step(new_step,speed)


def main():

    music.music_play('./musics/sports-Ahjay_Stelino.mp3')
    music.music_set_volume(20)

    while True:
        twist(speed=100)
```

(continues on next page)

(continued from previous page)

```
if __name__ == "__main__":
    main()
```

How it works?

In this code, you need to pay attention to this part:

```
def twist(speed):
    ## [right front],[left front],[left rear],[right rear]
    new_step=[[50, 50, -80], [50, 50, -80],[50, 50, -80], [50, 50, -80]]
    for i in range(4):
        for inc in range(30,60,5):
            rise = [50,50,(-80+inc*0.5)]
            drop = [50,50,(-80-inc)]

            new_step[i]=rise
            new_step[(i+2)%4] = drop
            new_step[(i+1)%4] = rise
            new_step[(i-1)%4] = drop
            crawler.do_step(new_step,speed)
```

Simply put, it uses two layers of for loops to make the `new_step` array produce continuous and regular changes, and at the same time, `crawler.do_step()` executes the posture to form a continuous action.

You can intuitively get the coordinate value array corresponding to each pose from *Adjust Posture*.

In addition, the example also played background music. The implementation method is as follows.

Play music by importing the following libraries.

```
from robot_hat import Music
```

Declare a Music object.

```
music = Music()
```

Play the background music in the `picrawler/examples/musics` directory and set the volume to 20. You can also add music to the `musics` folder via *Filezilla Software*.

```
music.music_play('./musics/sports-Ahjay_Stelino.mp3')
music.music_set_volume(20)
```

Note: You can add different sound effects or music to `musics` or `sounds` folder via *Filezilla Software*.

Note: Hello, welcome to the SunFounder Raspberry Pi & Arduino & ESP32 Enthusiasts Community on Facebook! Dive deeper into Raspberry Pi, Arduino, and ESP32 with fellow enthusiasts.

Why Join?

- **Expert Support**: Solve post-sale issues and technical challenges with help from our community and team.
- **Learn & Share**: Exchange tips and tutorials to enhance your skills.
- **Exclusive Previews**: Get early access to new product announcements and sneak peeks.
- **Special Discounts**: Enjoy exclusive discounts on our newest products.

- **Festive Promotions and Giveaways**: Take part in giveaways and holiday promotions.

Ready to explore and create with us? Click [] and join today!

1.15 Emotional Robot

This example shows several interesting custom actions of PiCrawler.

Run the Code

```
cd ~/picrawler/examples
sudo python3 emotional_robot.py
```

Code

Note: You can **Modify/Reset/Copy/Run/Stop** the code below. But before that, you need to go to source code path like `picrawler\examples`. After modifying the code, you can run it directly to see the effect.

```
from picrawler import Picrawler
from time import sleep

crawler = Picrawler()

def handwork(speed):

    basic_step = []
    basic_step = crawler.step_list.get("sit")
    left_hand  = crawler.mix_step(basic_step,0,[0,50,80])
    right_hand  = crawler.mix_step(basic_step,1,[0,50,80])
    two_hand = crawler.mix_step(left_hand,1,[0,50,80])

    crawler.do_step('sit',speed)
    sleep(0.6)
    crawler.do_step(left_hand,speed)
    sleep(0.6)
    crawler.do_step(two_hand,speed)
    sleep(0.6)
    crawler.do_step(right_hand,speed)
    sleep(0.6)
    crawler.do_step('sit',speed)
    sleep(0.6)

def twist(speed):
    new_step=[[50, 50, -80], [50, 50, -80],[50, 50, -80], [50, 50, -80]]
    for i in range(4):
        for inc in range(30,60,5):
            rise = [50,50,(-80+inc*0.5)]
            drop = [50,50,(-80-inc)]

            new_step[i]=rise
```

(continues on next page)

(continued from previous page)

```
            new_step[(i+2)%4] = drop
            new_step[(i+1)%4] = rise
            new_step[(i-1)%4] = drop
            crawler.do_step(new_step,speed)

##"[[right front], [left front], [left rear], [left rear]]")

def pushup(speed):
    up=[[80, 0, -100], [80, 0, -100],[0, 120, -60], [0, 120, -60]]
    down=[[80, 0, -30], [80, 0, -30],[0, 120, -60], [0, 120, -60]]
    crawler.do_step(up,speed)
    sleep(0.6)
    crawler.do_step(down,speed)
    sleep(0.6)

def swimming(speed):
    for i in range(100):
        crawler.do_step([[100-i,i,0],[100-i,i,0],[0,120,-60+i/5],[0,100,-40-i/5]],speed)

# main
def main():
    speed = 100

    swimming(speed)
    pushup(speed)
    handwork(speed)
    twist(speed)

    sleep(0.05)

if __name__ == "__main__":
    main()
```

Note: Hello, welcome to the SunFounder Raspberry Pi & Arduino & ESP32 Enthusiasts Community on Facebook! Dive deeper into Raspberry Pi, Arduino, and ESP32 with fellow enthusiasts.

Why Join?

- **Expert Support**: Solve post-sale issues and technical challenges with help from our community and team.
- **Learn & Share**: Exchange tips and tutorials to enhance your skills.
- **Exclusive Previews**: Get early access to new product announcements and sneak peeks.
- **Special Discounts**: Enjoy exclusive discounts on our newest products.
- **Festive Promotions and Giveaways**: Take part in giveaways and holiday promotions.

Ready to explore and create with us? Click [] and join today!

CHAPTER

TWO

PLAY WITH EZBLOCK

Note: If you are using a Raspberry Pi 5, our graphical programming software, EzBlock, is not supported.

For beginners and novices, EzBlock is a software development platform offered by SunFounder for Raspberry Pi. Ezbock offers two programming environments: a graphical environment and a Python environment.

It is available for almost all types of devices, including Mac, PC, and Android.

Here is a tutorial to help you complete EzBlock installation, download, and use.

Note: Hello, welcome to the SunFounder Raspberry Pi & Arduino & ESP32 Enthusiasts Community on Facebook! Dive deeper into Raspberry Pi, Arduino, and ESP32 with fellow enthusiasts.

Why Join?

- **Expert Support**: Solve post-sale issues and technical challenges with help from our community and team.
- **Learn & Share**: Exchange tips and tutorials to enhance your skills.
- **Exclusive Previews**: Get early access to new product announcements and sneak peeks.
- **Special Discounts**: Enjoy exclusive discounts on our newest products.
- **Festive Promotions and Giveaways**: Take part in giveaways and holiday promotions.

Ready to explore and create with us? Click [] and join today!

2.1 Quick Guide on EzBlock

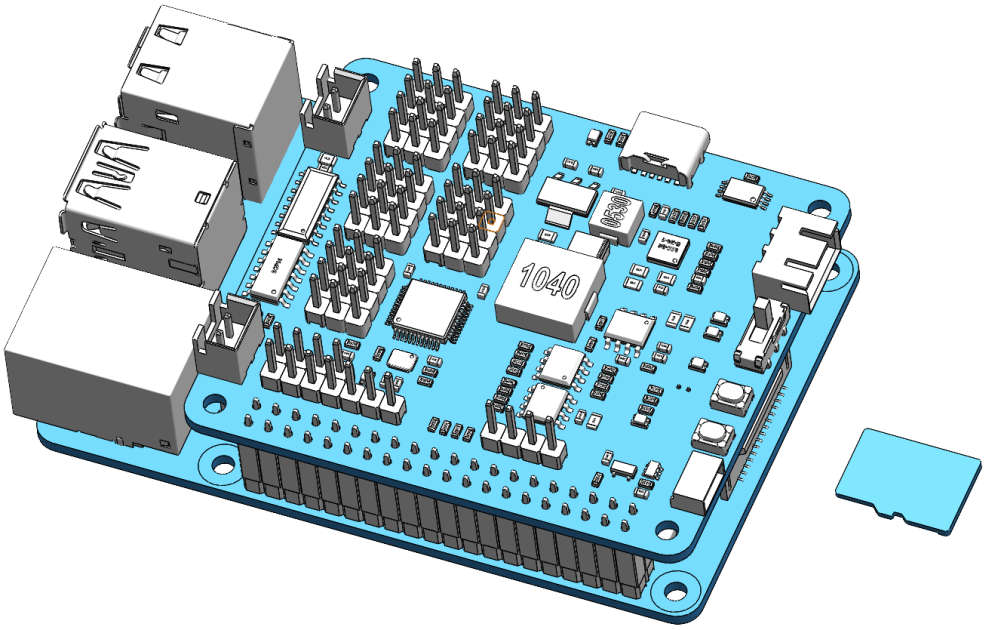
Note: If you are using a Raspberry Pi 5, our graphical programming software, EzBlock, is not supported.

The angle range of the servo is -90~90, but the angle set at the factory is random, maybe 0°, maybe 45°; if we assemble it with such an angle directly, it will lead to a chaotic state after the robot runs the code, or worse, it will cause the servo to block and burn out.

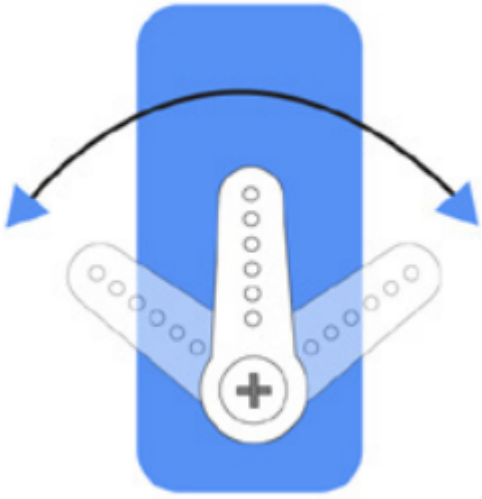
So here we need to set all the servo angles to 0° and then install them, so that the servo angle is in the middle, no matter which direction to turn.

1. Firstly, Install EzBlock OS (EzBlock's own tutorials) onto a Micro SD card, once the installation is complete, insert it into the Raspberry Pi.

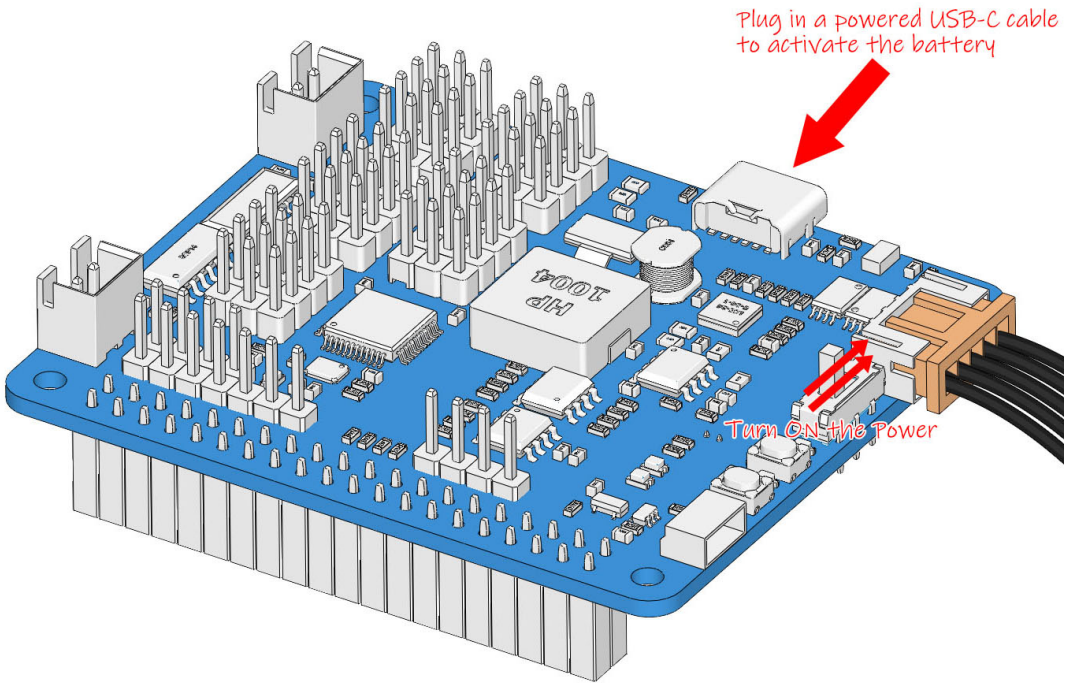
Note: After the installation is complete, please return to this page.



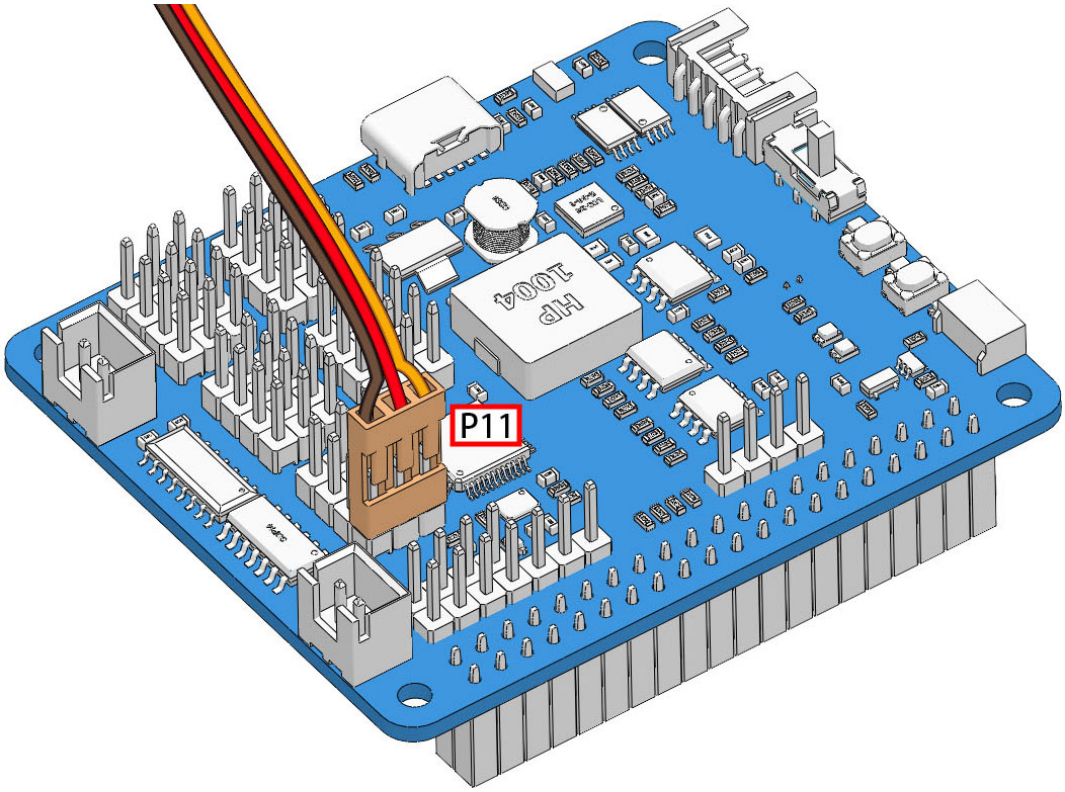
2. To ensure that the servo has been properly set to 0°, first insert the servo arm into the servo shaft and then gently rotate the rocker arm to a different angle. This servo arm is just to allow you to clearly see that the servo is rotating.



3. Follow the instructions on the assembly foldout, insert the battery cable and turn the power switch to the ON. Then plug in a powered USB-C cable to activate the battery. Wait for 1-2 minutes, there will be a sound to indicate that the Raspberry Pi boots successfully.

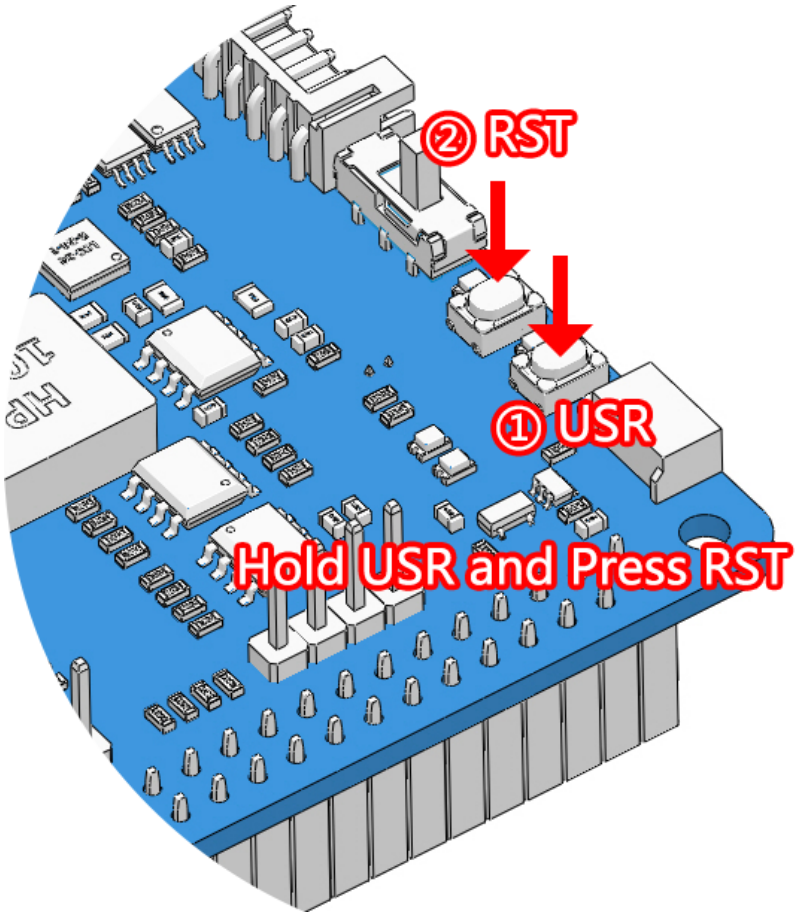


4. Next, plug the servo cable into the P11 port as follows.



5. Press and hold the **USR** key, then press the **RST** key to execute the servo zeroing script within the system. When you see the servo arm rotate to a position(This is the 0° position, which is a random location and may not be vertical or parallel.), it indicates that the program has run.

Note: This step only needs to be done once; afterward, simply insert other servo wires, and they will automatically zero.



6. Now, remove the servo arm, ensuring the servo wire remains connected, and do not turn off the power. Then continue the assembly following the paper assembly instructions.

Note:

- Do not unplug this servo cable before fastening this servo with the servo screw, you can unplug it after fastening.
- Do not turn the servo while it is powered on to avoid damage; if the servo shaft is inserted at the wrong angle, pull out the servo and reinsert it.
- Before assembling each servo, you need to plug the servo cable into P11 and turn on the power to set its angle to 0°.
- This zeroing function will be disabled if you download a program to the robot later with the EzBlock APP.

2.2 Install and Configure EzBlock Studio

As soon as the robot is assembled, you will need to carry out some basic operations.

- Install EzBlock Studio: Download and install EzBlock Studio on your device or use the web-based version.
- Connect the Product and EzBlock: Configure Wi-Fi, Bluetooth and calibrate before use.
- Open and Run Examples: View or run the related example directly.

Note: After you connect the PiCrawler, there will be a calibration step. This is because of possible deviations in the installation process or limitations of the servos themselves, making some servo angles slightly tilted, so you can calibrate them in this step.

But if you think the assembly is perfect and no calibration is needed, you can also skip this step.

Note: Hello, welcome to the SunFounder Raspberry Pi & Arduino & ESP32 Enthusiasts Community on Facebook! Dive deeper into Raspberry Pi, Arduino, and ESP32 with fellow enthusiasts.

Why Join?

- **Expert Support**: Solve post-sale issues and technical challenges with help from our community and team.
- **Learn & Share**: Exchange tips and tutorials to enhance your skills.
- **Exclusive Previews**: Get early access to new product announcements and sneak peeks.
- **Special Discounts**: Enjoy exclusive discounts on our newest products.
- **Festive Promotions and Giveaways**: Take part in giveaways and holiday promotions.

Ready to explore and create with us? Click [] and join today!

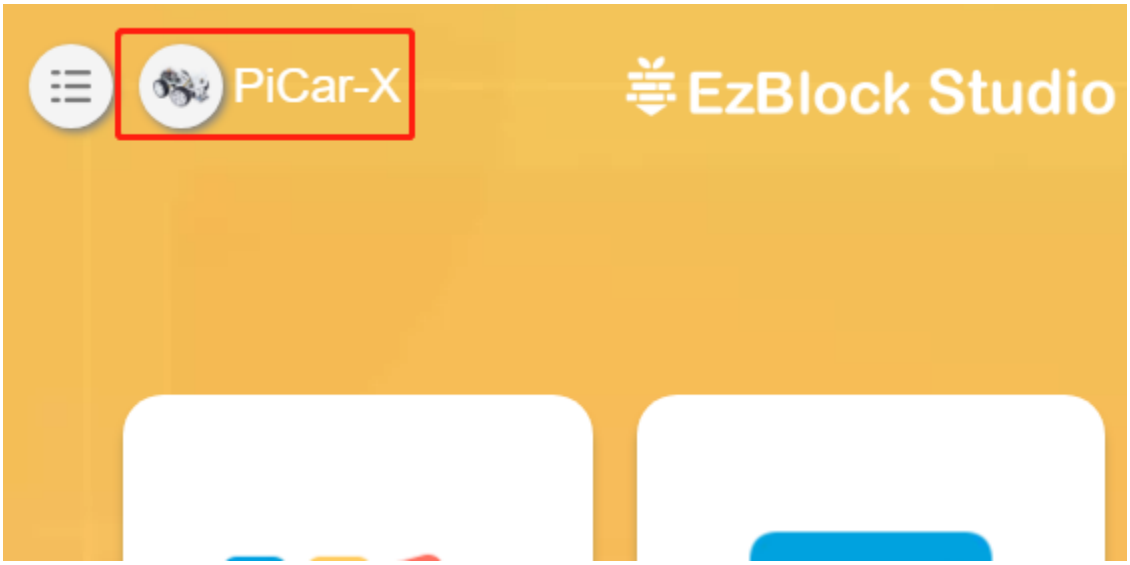
2.3 Calibrate the PiCrawler

After you connect the PiCrawler, there will be a calibration step. This is because of possible deviations in the installation process or limitations of the servos themselves, making some servo angles slightly tilted, so you can calibrate them in this step.

But if you think the assembly is perfect and no calibration is needed, you can also skip this step.

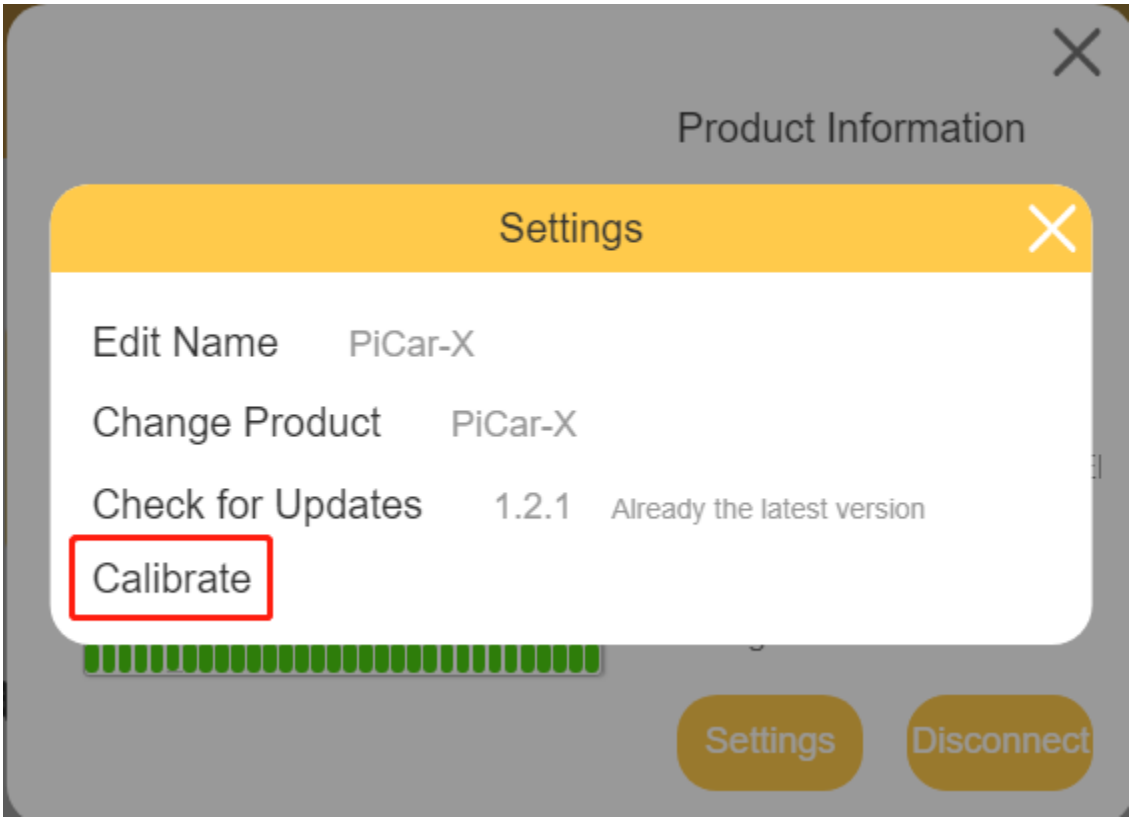
Note: If you want to recalibrate the robot during use, please follow the steps below.

You can open the product detail page by clicking the connect icon in the upper left corner.



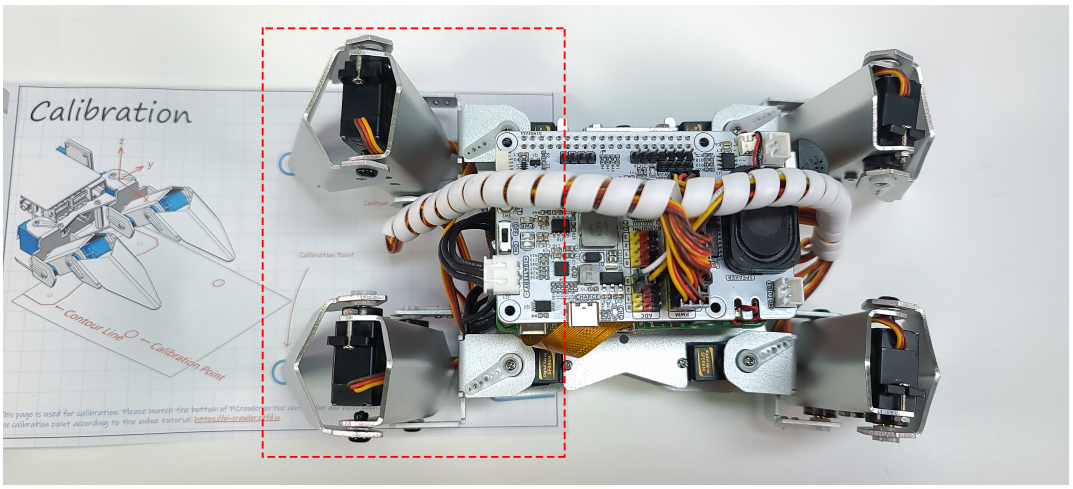
Click the **Settings** button.

On this page, you can change the product name, product type, view the app version or calibrate the robot. Once you click on **Calibrate** you can go to the calibration page.

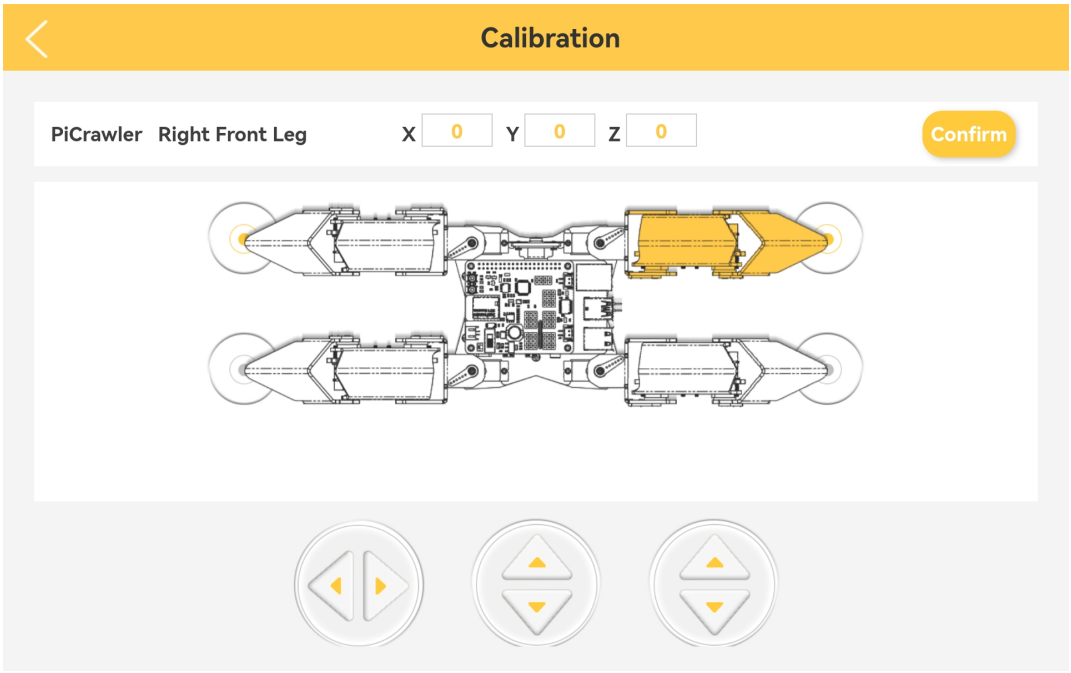


The calibration steps are as follows:

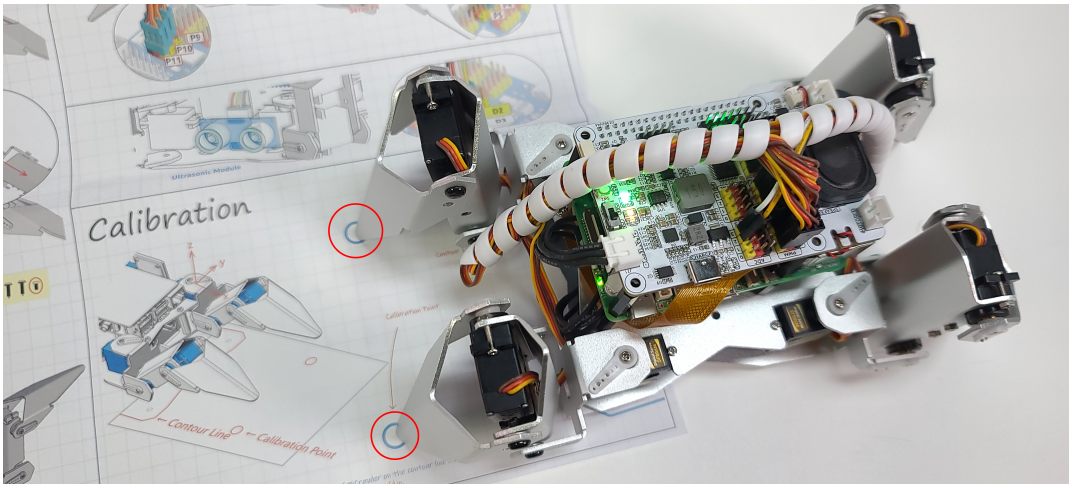
1. Take out the assembly leaflet, turn it to the last page, and lay it flat on the table. Then place the PiCrawler as shown below, aligning its bottom with the outline on the calibration chart.



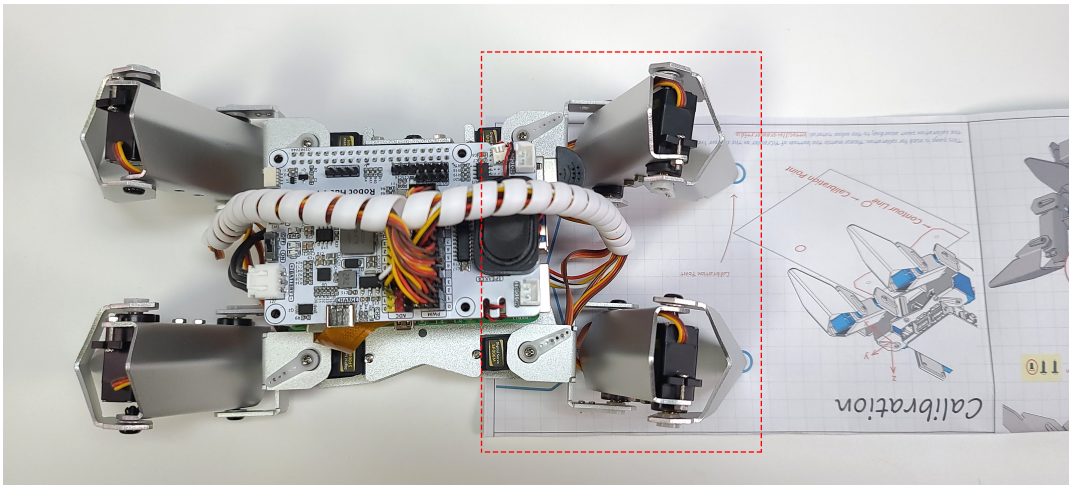
2. Go back to EzBlock Studio, select one foot on the left, then click the 3 sets of X, Y and Z buttons, and let the toes slowly align with the calibration point.
    - The calibration buttons are used for fine-tuning, and you need to press these buttons multiple times to see the pin position change.
    - It is recommended to click the up button of Z axis to lift the foot up first, then go to adjust X and Y.



3. Align the other foot on the left in the same way.



4. After calibrating the left two feet, change the calibration paper to the right, and calibrate the right two feet according to the above method.



Projects

Here, we show you the projects of playing PiCrawler on Ezblock Studio. If you are new to these, you can refer to the code images inside each project to program, and can learn the use of blocks according to TIPS.

If you don't want to write these projects one by one, we have uploaded them to Ezblock Studio's Examples page and you can run them directly or edit them and run them later.

Note: Hello, welcome to the SunFounder Raspberry Pi & Arduino & ESP32 Enthusiasts Community on Facebook! Dive deeper into Raspberry Pi, Arduino, and ESP32 with fellow enthusiasts.

Why Join?

- **Expert Support**: Solve post-sale issues and technical challenges with help from our community and team.
- **Learn & Share**: Exchange tips and tutorials to enhance your skills.
- **Exclusive Previews**: Get early access to new product announcements and sneak peeks.
- **Special Discounts**: Enjoy exclusive discounts on our newest products.
- **Festive Promotions and Giveaways**: Take part in giveaways and holiday promotions.

Ready to explore and create with us? Click [] and join today!

2.4 Move

This is PiCrawler's first project. Perform its most basic function - move.

Program

Note:

- You can write the program according to the following picture, please refer to the tutorial: How to Create a New Project?.
- Or find the code with the same name on the **Examples** page of the EzBlock Studio and click **Run** or **Edit** directly.

Click the Upload & Run button at the bottom right of the screen, and PiCrawler will execute “forward” and “backward” actions in sequence.

How it works?

First, you need to understand the program framework of Ezblock. as follows:

All Ezblock projects contain these two blocks. The **Start** block runs at the beginning of the program and is executed only once, and is often used to set variables; the **Forever** block runs after **Start**, and will be executed repeatedly, and is often used to implement main functions. If you delete these two blocks, you can drag them back from the **Basic** category on the left.

Next you need to understand the following blocks.

do action allows PiCrawler to perform basic actions. You can modify the options in the first groove. For example, select "Turn Left", "Back" and so on. The second groove can set the number of executions of the action, and only integer numbers greater than 0 can be written. The third groove can set the speed of the action, and only integers within 0~100 can be written.

do step is similar to **do action**, but it is not an action but a static posture. Such as "stand", "sit".

Both blocks can be dragged from the **PiCrawler** category on the left.

Note: Hello, welcome to the SunFounder Raspberry Pi & Arduino & ESP32 Enthusiasts Community on Facebook! Dive deeper into Raspberry Pi, Arduino, and ESP32 with fellow enthusiasts.

Why Join?

- **Expert Support**: Solve post-sale issues and technical challenges with help from our community and team.
- **Learn & Share**: Exchange tips and tutorials to enhance your skills.
- **Exclusive Previews**: Get early access to new product announcements and sneak peeks.

- **Special Discounts**: Enjoy exclusive discounts on our newest products.
- **Festive Promotions and Giveaways**: Take part in giveaways and holiday promotions.

Ready to explore and create with us? Click [] and join today!

2.5 Remote Control

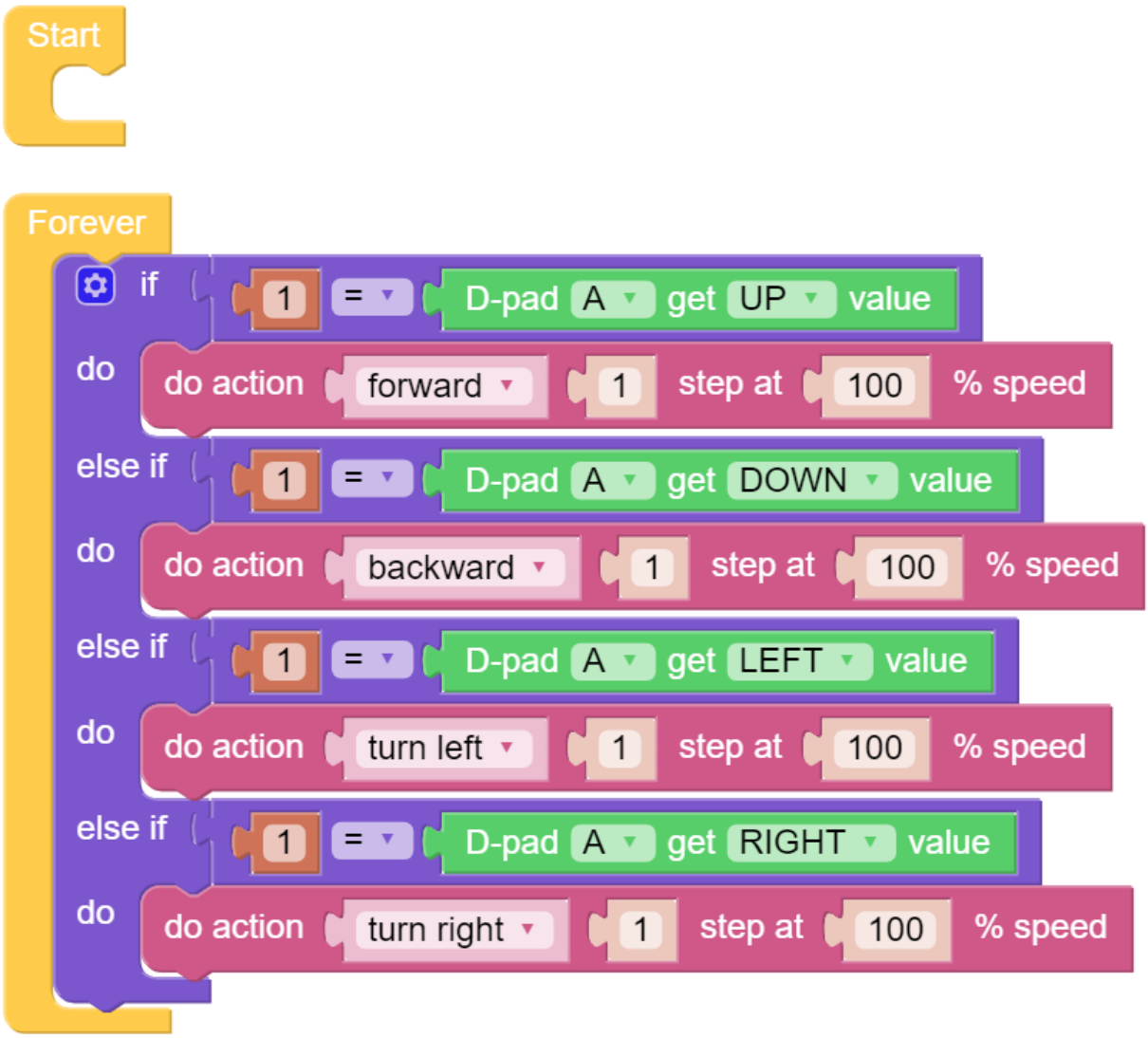
In this project, we will learn how to remote control the PiCrawler. You can control the PiCrawler to move forward, backward, left, and right.

Note: You can refer to How to Use the Remote Control Function?. Come and carry out this project smoothly.

Program

Note:

- You can write the program according to the following picture, please refer to the tutorial: How to Create a New Project?.
- Or find the code with the same name on the **Examples** page of the EzBlock Studio and click **Run** or **Edit** directly.



Switch to the Remote Control interface, and you will see the following widgets.

After the program is running, you can activate PiCrawler through D-Pad.

How it works?

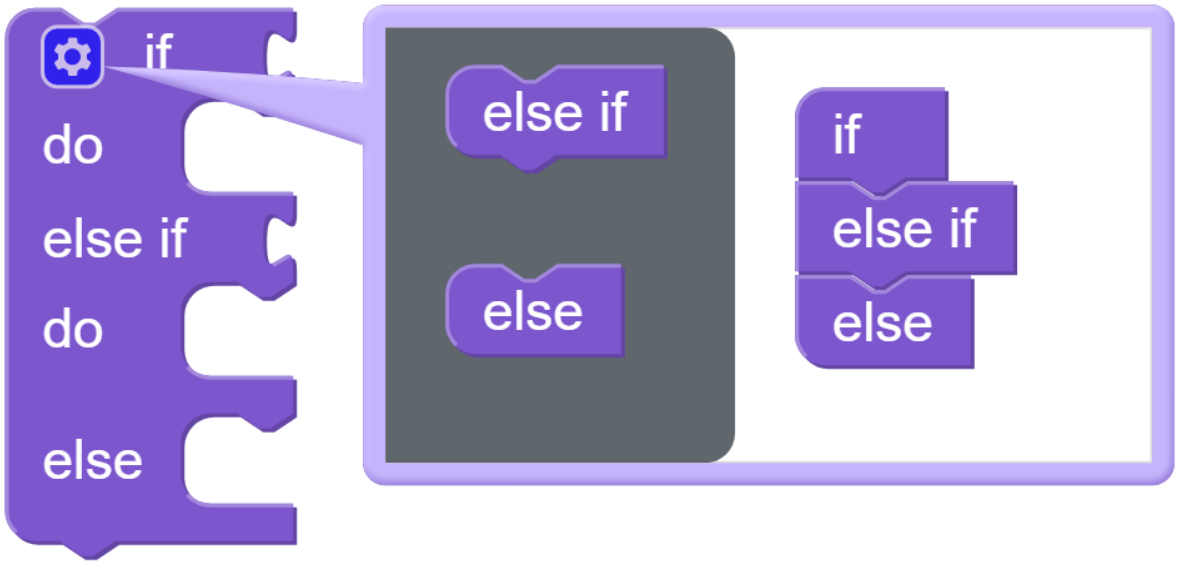
After dragging out the widget on the Remote Control interface, a category named **Remote** will appear in the block categories column of the programming interface.

Here we add the D-Pad widget, so the **D-Pad get value** block appears here.

D-Pad can be regarded as a four-in-one button. You can choose which button to read in the second groove of the block.

When the button is pressed, the value is “1”; when the button is not pressed, the value is “0”.

We used an **if** block (you can find it in the **Logic** category on the left) to make the PiCrawler move forward once when the **UP** button of the D-pad is pressed.



You can click the gear icon on the upper left of the block to modify the shape of the **if** block to realize multiple judgment branches.

if block is usually used with **=** block, **=** block can be modified to **>**, **<** and other conditions through the drop-down menu, please use it flexibly.

Note: Hello, welcome to the SunFounder Raspberry Pi & Arduino & ESP32 Enthusiasts Community on Facebook! Dive deeper into Raspberry Pi, Arduino, and ESP32 with fellow enthusiasts.

Why Join?

- **Expert Support**: Solve post-sale issues and technical challenges with help from our community and team.
- **Learn & Share**: Exchange tips and tutorials to enhance your skills.
- **Exclusive Previews**: Get early access to new product announcements and sneak peeks.
- **Special Discounts**: Enjoy exclusive discounts on our newest products.

- **Festive Promotions and Giveaways**: Take part in giveaways and holiday promotions.

Ready to explore and create with us? Click [] and join today!

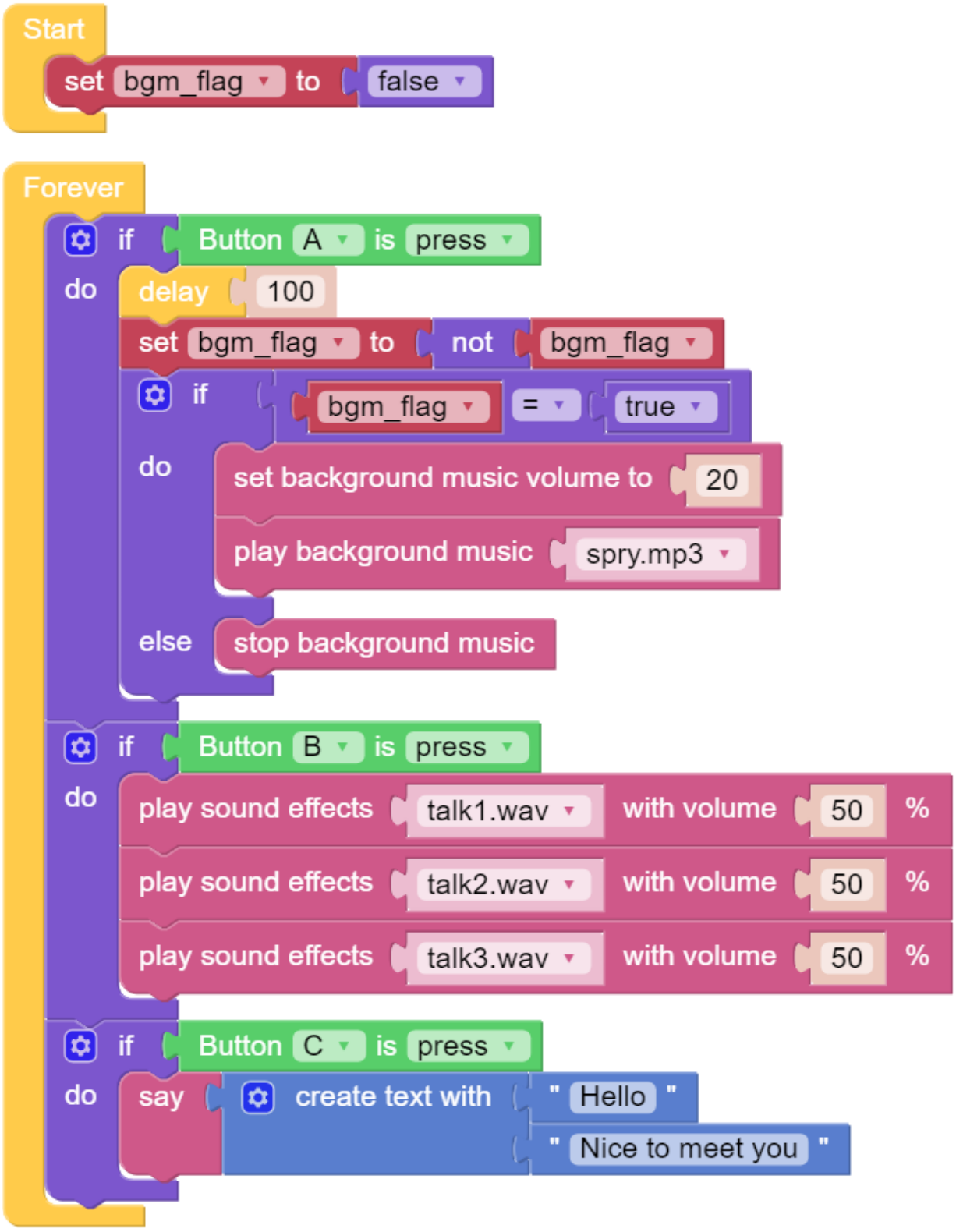
2.6 Sound Effect

In this example, we use PiCrawler's (to be precise, Robot HAT's) sound effects. It consists of three parts, namely **Muisc**, **Sound**, **Text to Speech**.

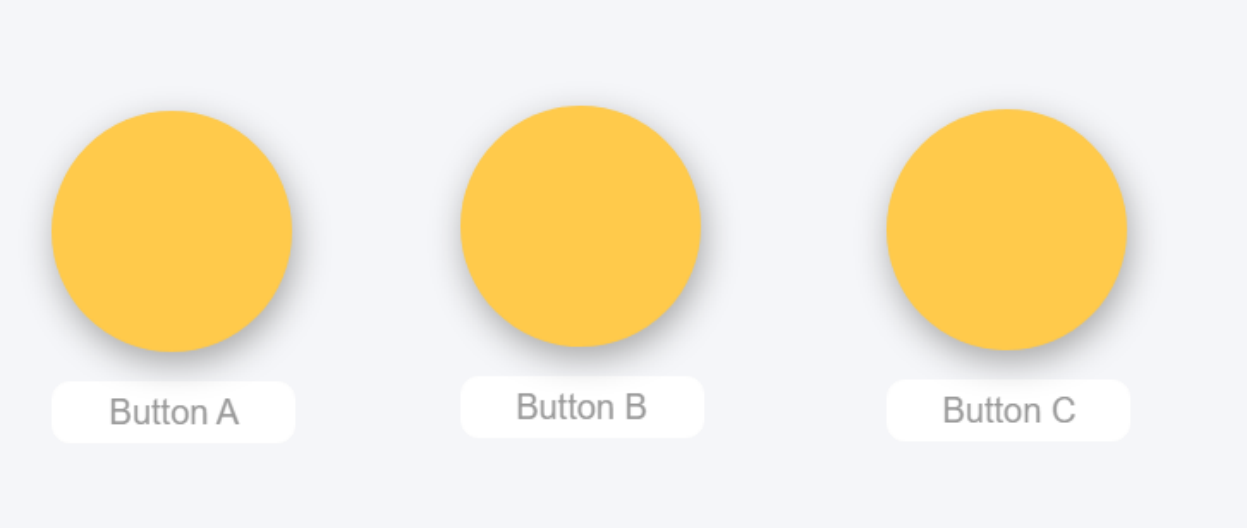
Program

Note:

- You can write the program according to the following picture, please refer to the tutorial: How to Create a New Project?.
- Or find the code with the same name on the **Examples** page of the EzBlock Studio and click **Run** or **Edit** directly.



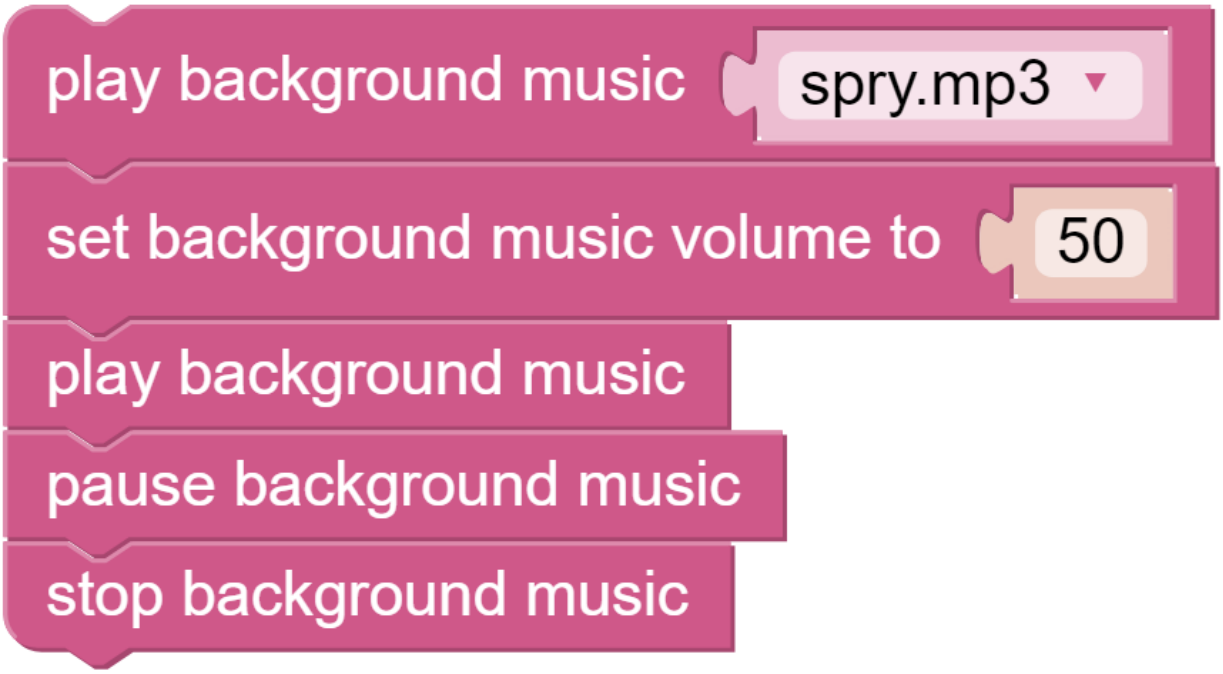
Switch to the Remote Control interface, and you will see the following widgets.



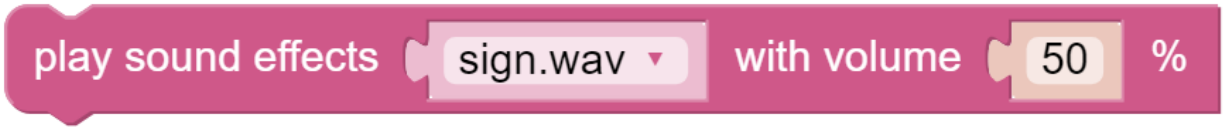
After the program is running, you can press different buttons to make PiCrawler sound.

How it works?

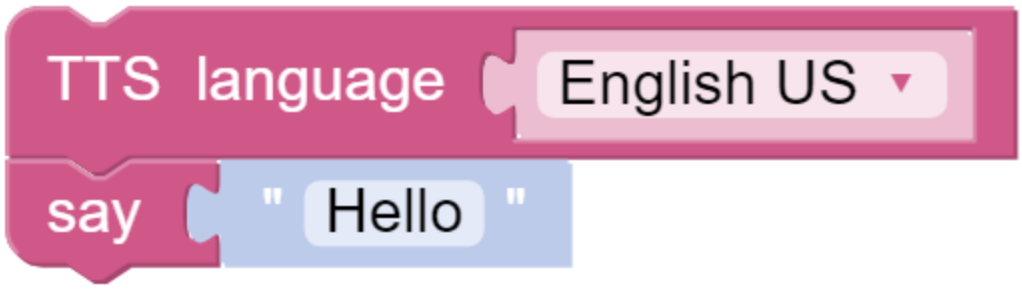
Functions related to background music include these:



Functions related to sound effects include these:



Functions related to Text to Speech include these:



Note: Hello, welcome to the SunFounder Raspberry Pi & Arduino & ESP32 Enthusiasts Community on Facebook! Dive deeper into Raspberry Pi, Arduino, and ESP32 with fellow enthusiasts.

Why Join?

- **Expert Support**: Solve post-sale issues and technical challenges with help from our community and team.
- **Learn & Share**: Exchange tips and tutorials to enhance your skills.
- **Exclusive Previews**: Get early access to new product announcements and sneak peeks.
- **Special Discounts**: Enjoy exclusive discounts on our newest products.
- **Festive Promotions and Giveaways**: Take part in giveaways and holiday promotions.

Ready to explore and create with us? Click [] and join today!

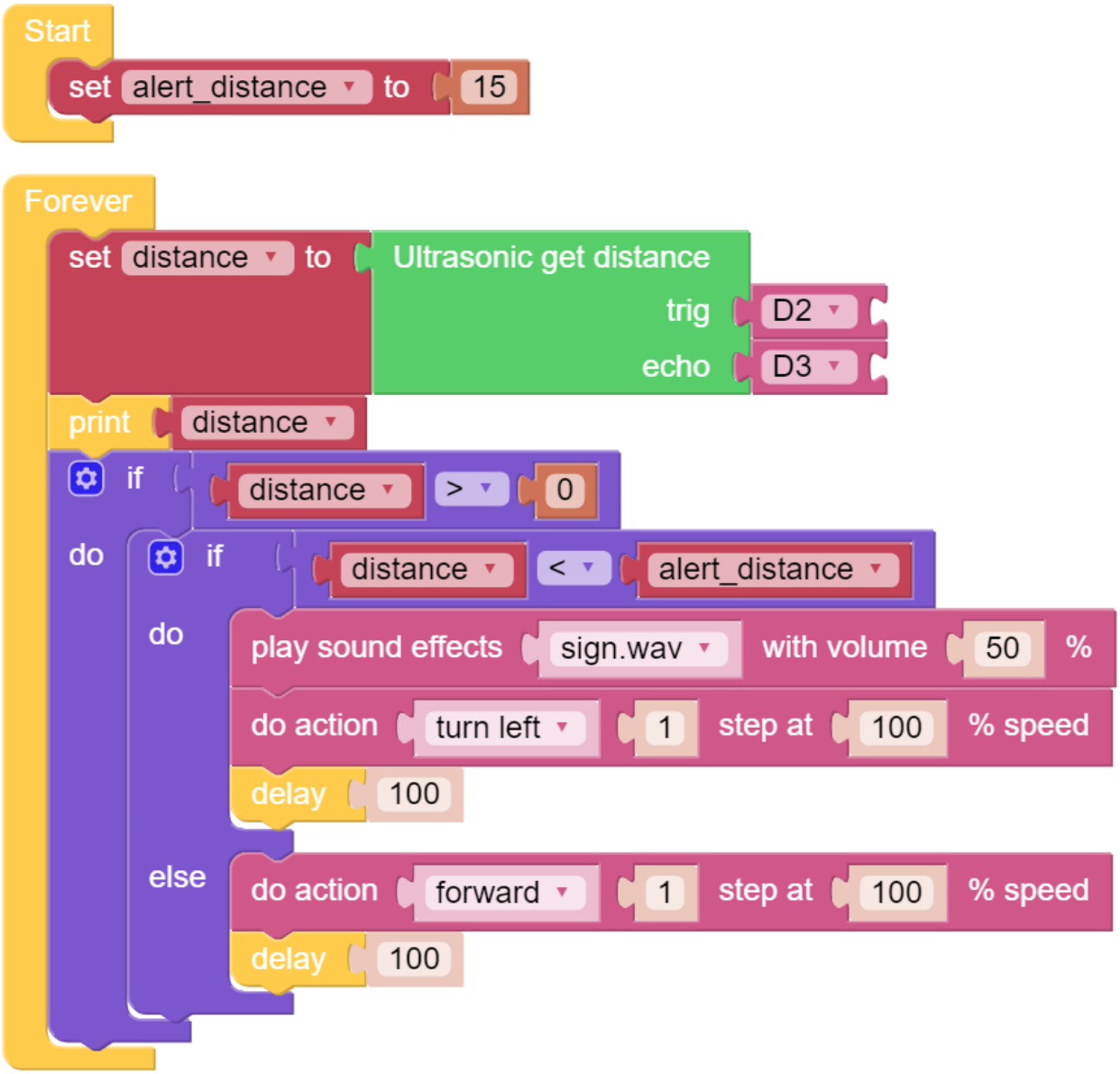
2.7 Obstacle Avoidance

In this project, picrawler will use an ultrasonic module to detect obstacles in front. When PiCrawler detects an obstacle, it will send a signal and look for another direction to move forward.

Program

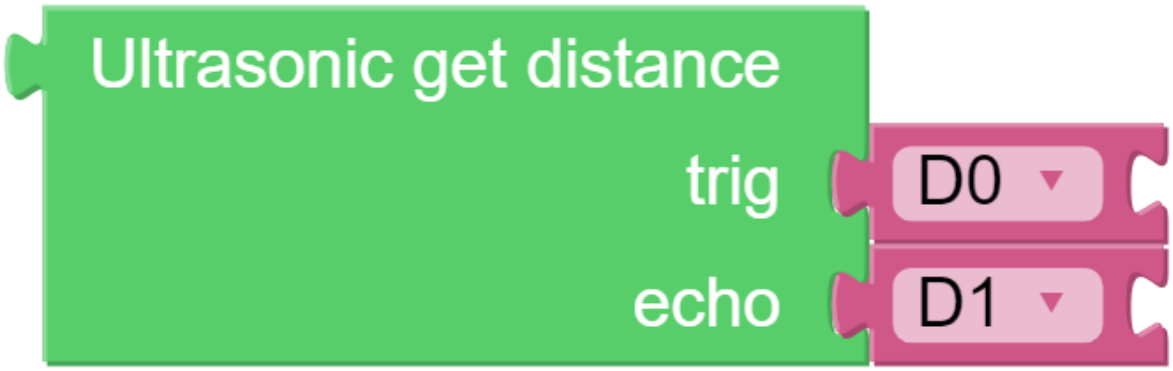
Note:

- You can write the program according to the following picture, please refer to the tutorial: How to Create a New Project?.
- Or find the code with the same name on the **Examples** page of the EzBlock Studio and click **Run** or **Edit** directly.



How it works?

You can find the following blocks in the **Module** category to achieve distance detection:



It should be noted that the two pins of the block should correspond to the actual wiring, that is, trig-D2, echo-D3.

Here is the main program.

- Read the `distance` detected by ultrasonic module and filter out the values less than 0 (When the ultrasonic module is too far from the obstacle or cannot read the data correctly, `distance<0` will appear).
- When the `distance` is less than `alert_distance` (the threshold value set earlier, which is 10), play the sound effect `sign.wav`. PiCrawler does `turn left` .
- When the `distance` is greater than `alert_distance`, PiCrawler will move `forward`.

Note: Hello, welcome to the SunFounder Raspberry Pi & Arduino & ESP32 Enthusiasts Community on Facebook! Dive deeper into Raspberry Pi, Arduino, and ESP32 with fellow enthusiasts.

Why Join?

- **Expert Support**: Solve post-sale issues and technical challenges with help from our community and team.
- **Learn & Share**: Exchange tips and tutorials to enhance your skills.
- **Exclusive Previews**: Get early access to new product announcements and sneak peeks.
- **Special Discounts**: Enjoy exclusive discounts on our newest products.
- **Festive Promotions and Giveaways**: Take part in giveaways and holiday promotions.

Ready to explore and create with us? Click [] and join today!

2.8 Computer Vision

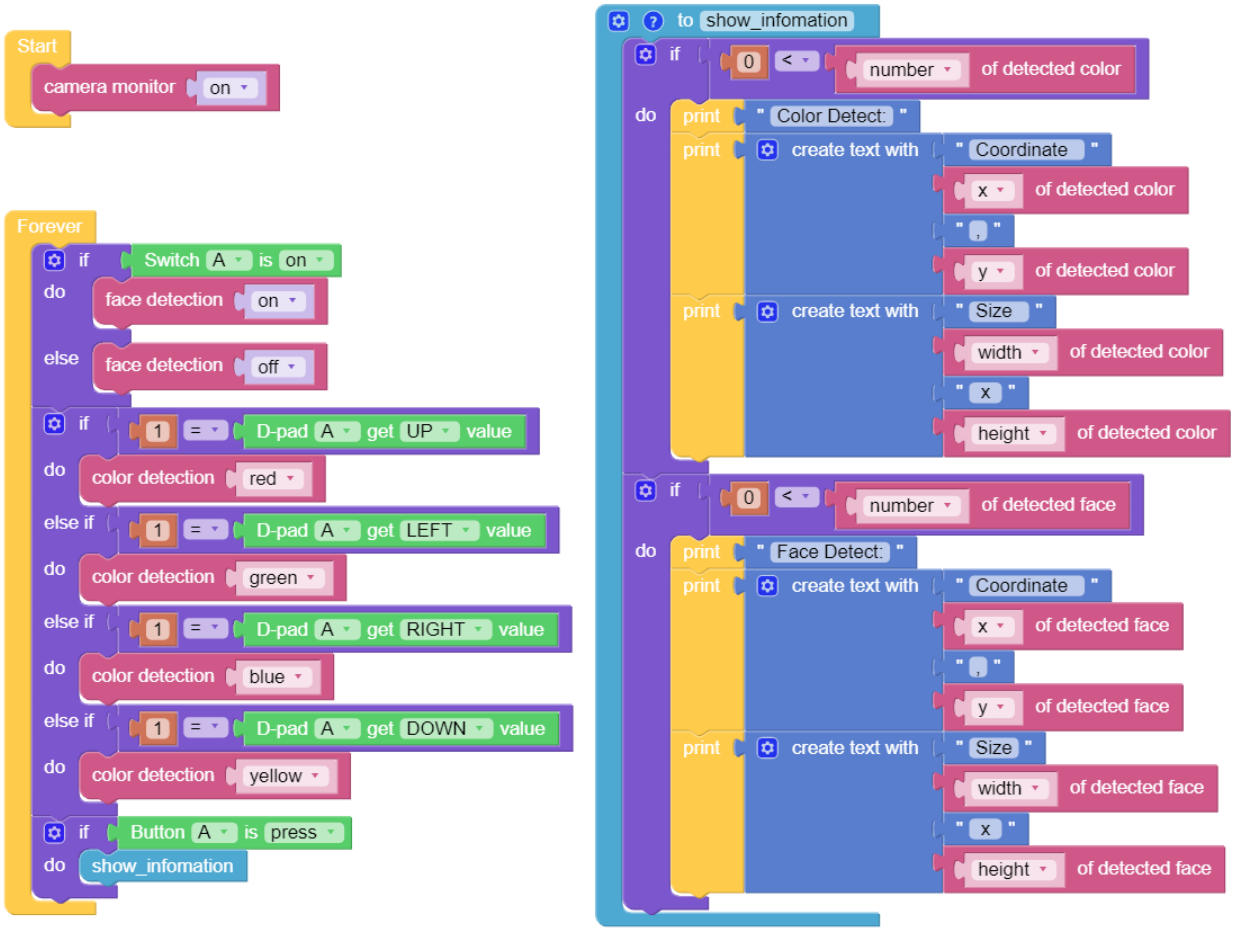
This project will officially enter the field of computer vision!

Note: You can read How to Use the Video Function?. Come and carry out this project smoothly.

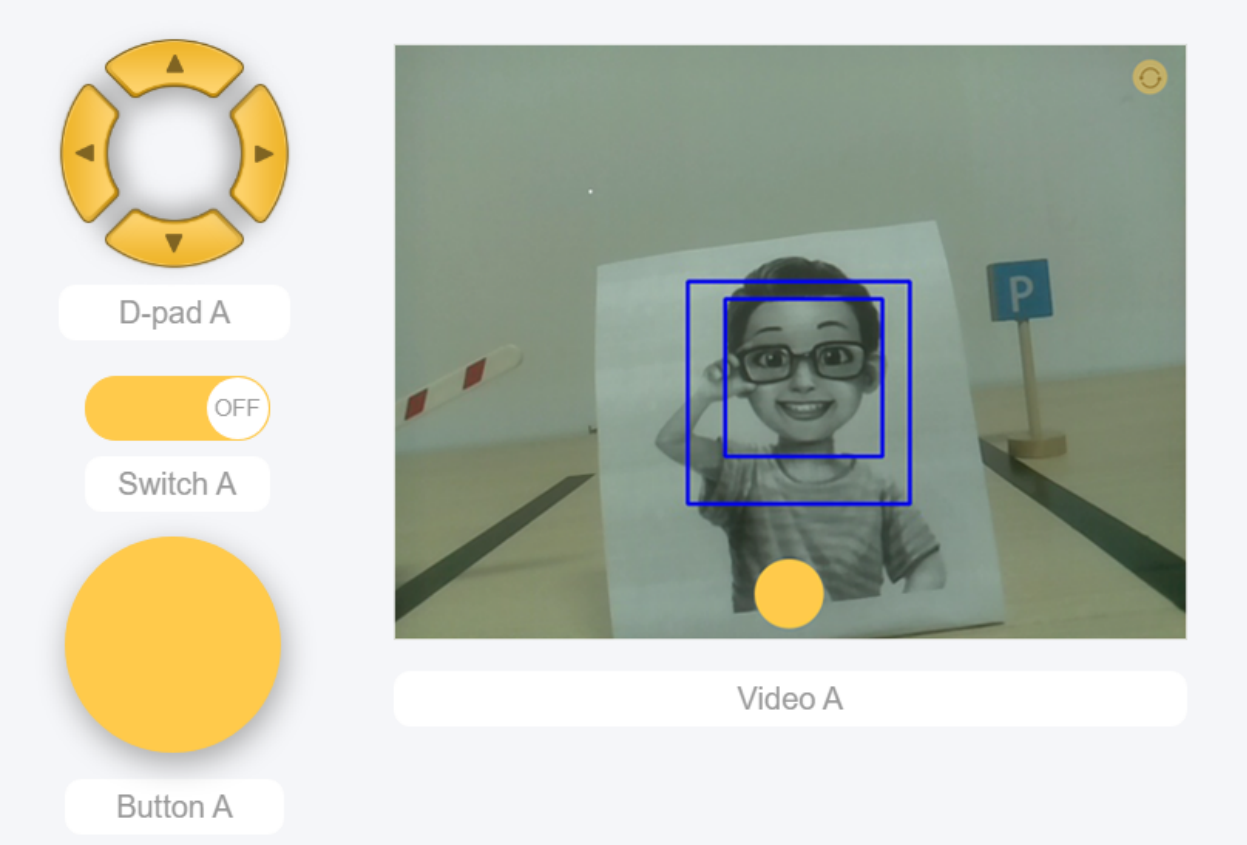
Program

Note:

- You can write the program according to the following picture, please refer to the tutorial: How to Create a New Project?.
- Or find the code with the same name on the **Examples** page of the EzBlock Studio and click **Run** or **Edit** directly.



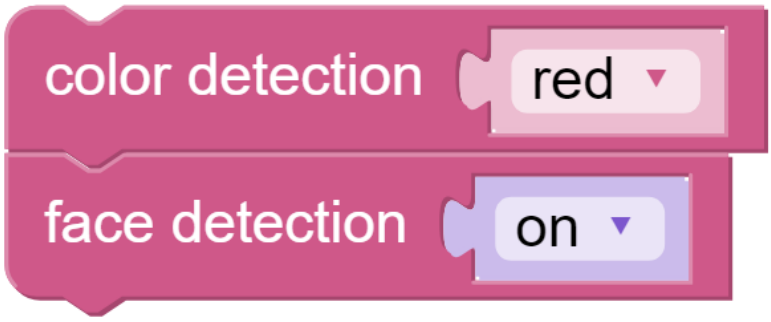
Switch to the Remote Control interface, and you will see the following widgets.



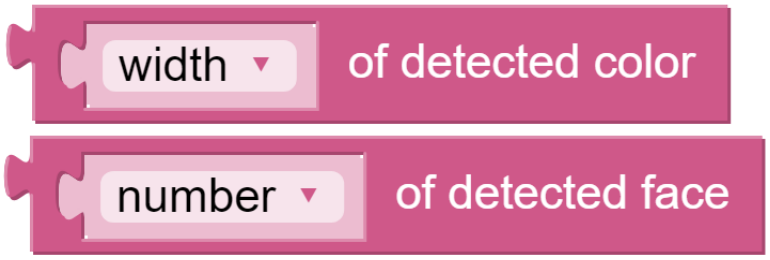
After the program is running, you can switch the slider widget to turn on/off the face detection; click the D-Pad to select the color of the detection; click the button to print the detection result.

How it works?

This block is used to enable the camera module.



These two blocks are used to enable the face detection/color detection function.



These two blocks are used to output information. The detection result has five output values, namely coordinate x value, coordinate y value, width, height, and number.

Note: Hello, welcome to the SunFounder Raspberry Pi & Arduino & ESP32 Enthusiasts Community on Facebook! Dive deeper into Raspberry Pi, Arduino, and ESP32 with fellow enthusiasts.

Why Join?

- **Expert Support**: Solve post-sale issues and technical challenges with help from our community and team.
- **Learn & Share**: Exchange tips and tutorials to enhance your skills.
- **Exclusive Previews**: Get early access to new product announcements and sneak peeks.
- **Special Discounts**: Enjoy exclusive discounts on our newest products.
- **Festive Promotions and Giveaways**: Take part in giveaways and holiday promotions.

Ready to explore and create with us? Click [] and join today!

2.9 Bull Fight

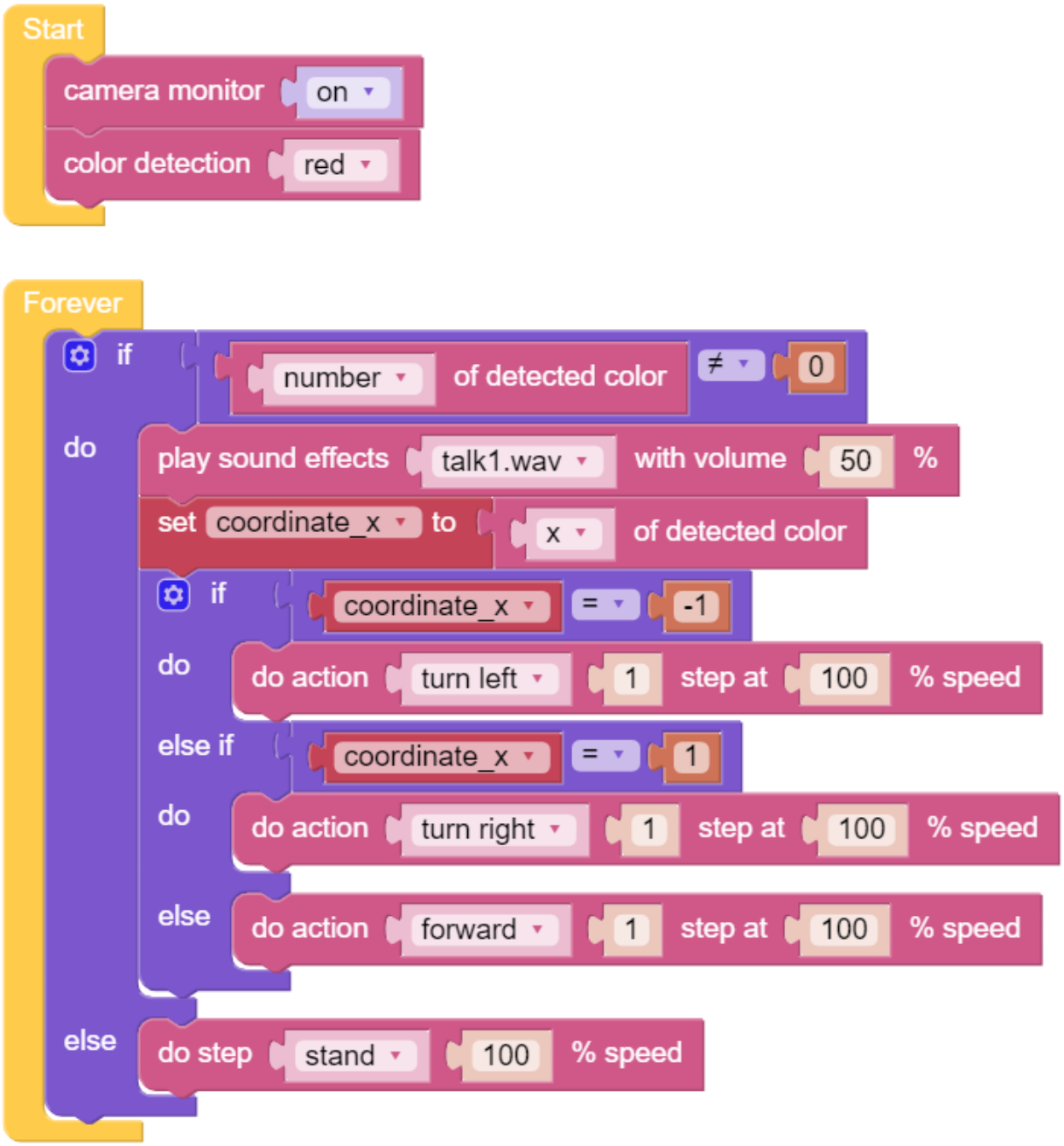
Make PiCrawler an angry bull! Use its camera to track and rush the red cloth!

Note: You can download and print the `PDF Color Cards` for color detection.

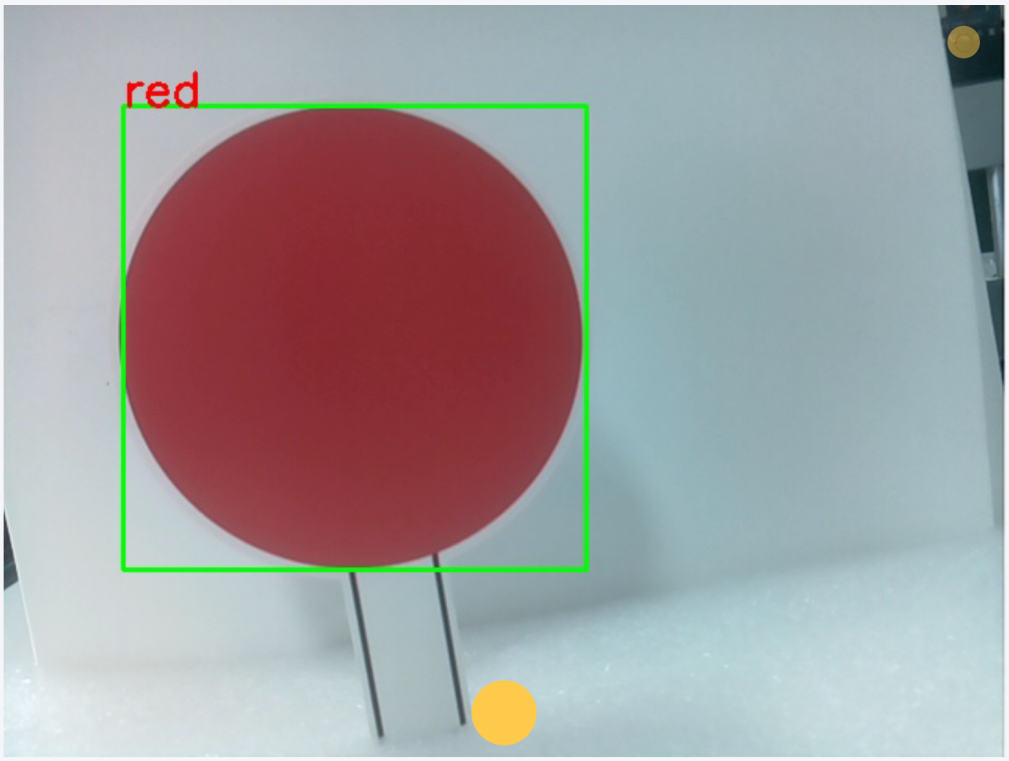
Program

Note:

- You can write the program according to the following picture, please refer to the tutorial: How to Create a New Project?.
- Or find the code with the same name on the **Examples** page of the EzBlock Studio and click **Run** or **Edit** directly.



Switch to the Remote Control interface, you will see the following screen.

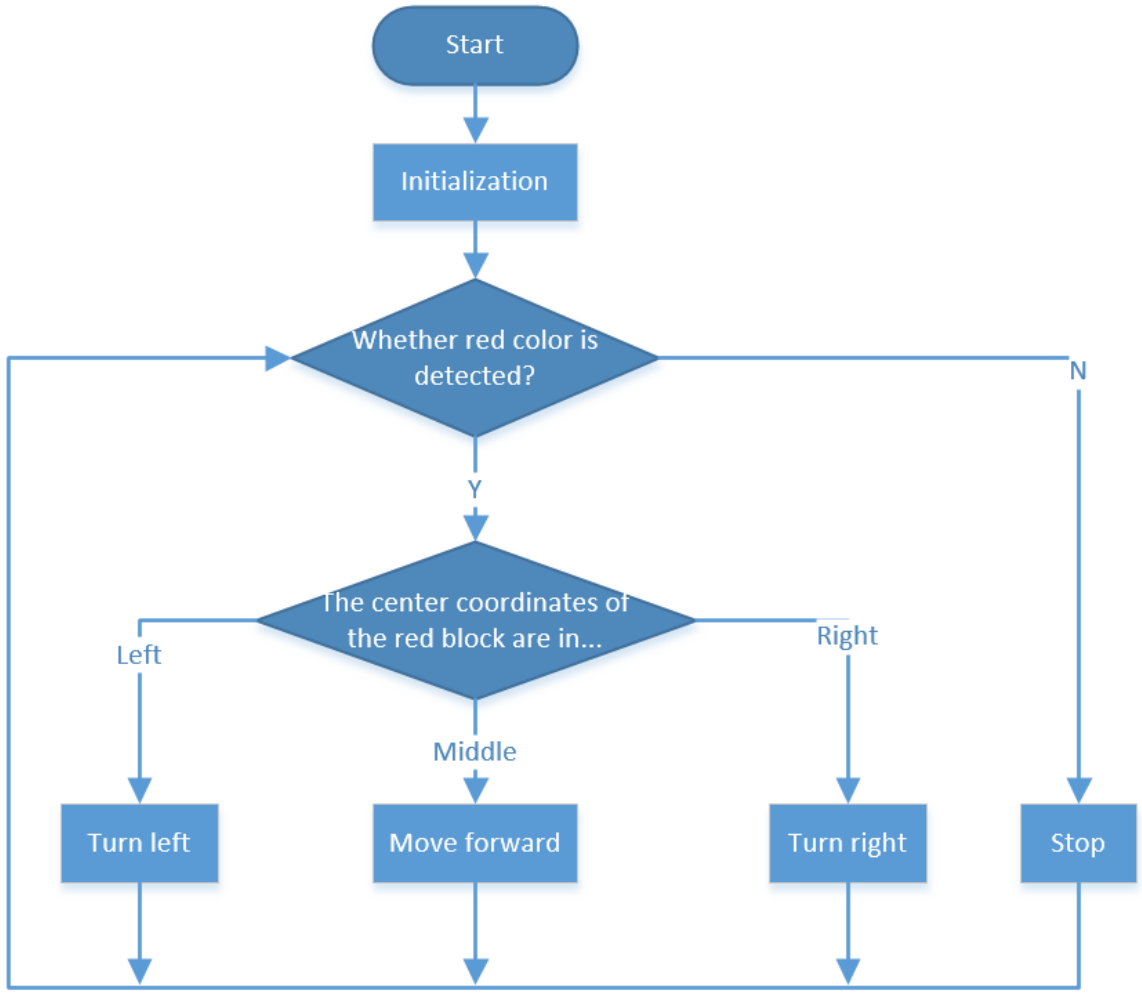


Video A

How it works?

In general, this project combines the knowledge points of *Move*, *Computer Vision* and *Sound Effect*.

Its flow is shown in the figure below:



Note: Hello, welcome to the SunFounder Raspberry Pi & Arduino & ESP32 Enthusiasts Community on Facebook! Dive deeper into Raspberry Pi, Arduino, and ESP32 with fellow enthusiasts.

Why Join?

- **Expert Support**: Solve post-sale issues and technical challenges with help from our community and team.
- **Learn & Share**: Exchange tips and tutorials to enhance your skills.
- **Exclusive Previews**: Get early access to new product announcements and sneak peeks.
- **Special Discounts**: Enjoy exclusive discounts on our newest products.
- **Festive Promotions and Giveaways**: Take part in giveaways and holiday promotions.

Ready to explore and create with us? Click [] and join today!

2.10 Treasure Hunt

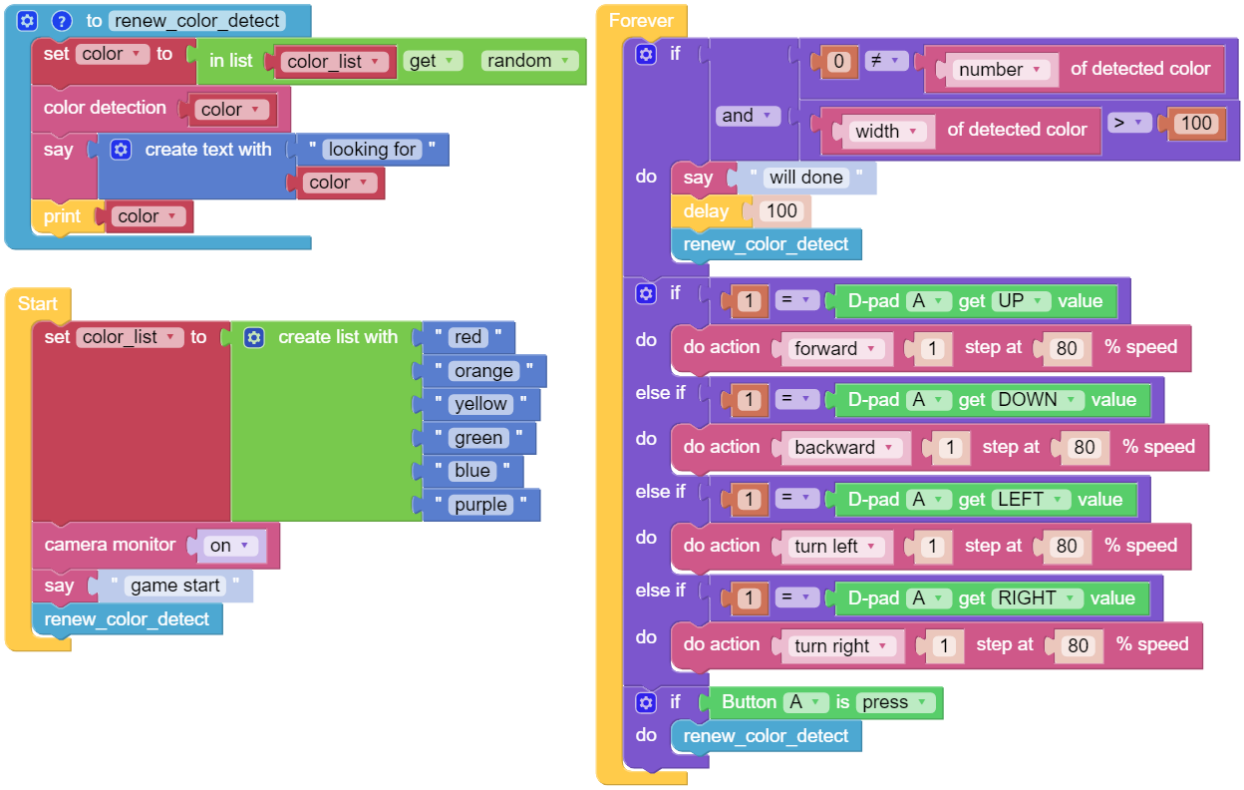
Arrange a maze in your room and place six different color cards in six corners. Then control PiCrawler to search for these color cards one by one!

Note: You can download and print the `PDF Color Cards` for color detection.

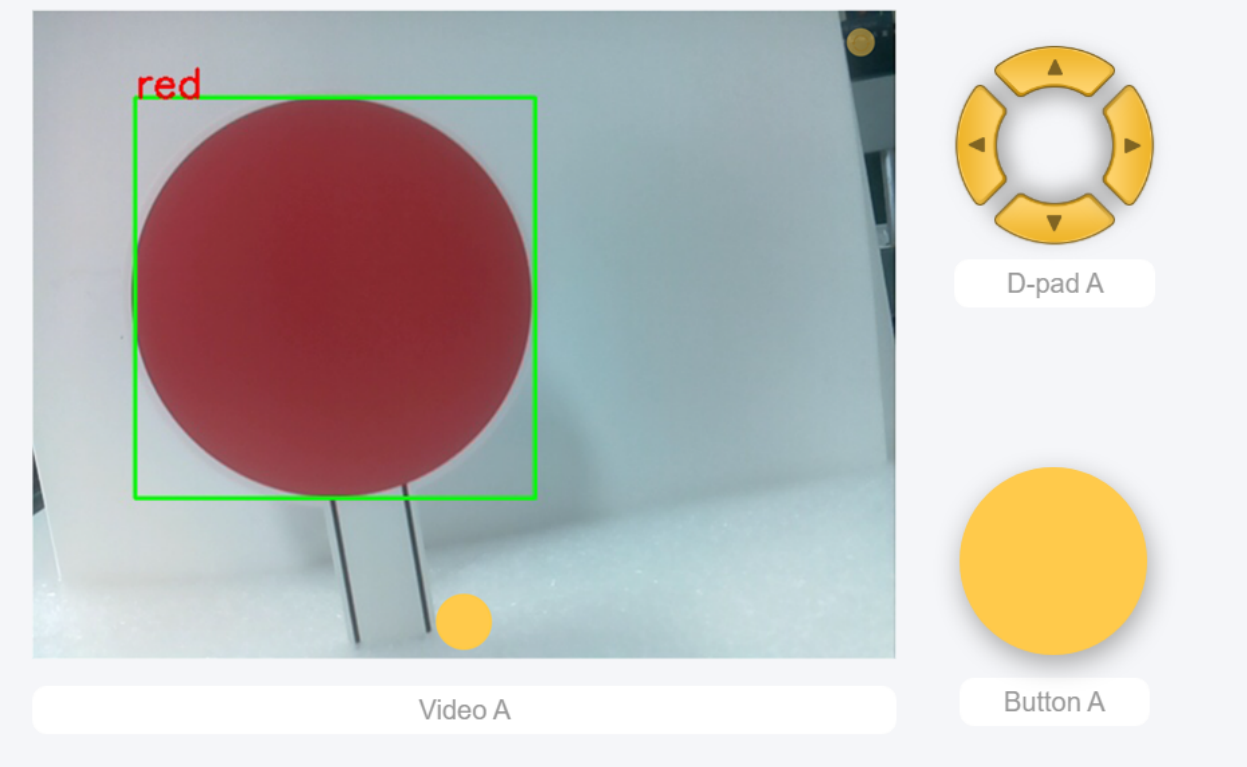
Program

Note:

- You can write the program according to the following picture, please refer to the tutorial: How to Create a New Project?.
- Or find the code with the same name on the **Examples** page of the EzBlock Studio and click **Run** or **Edit** directly.



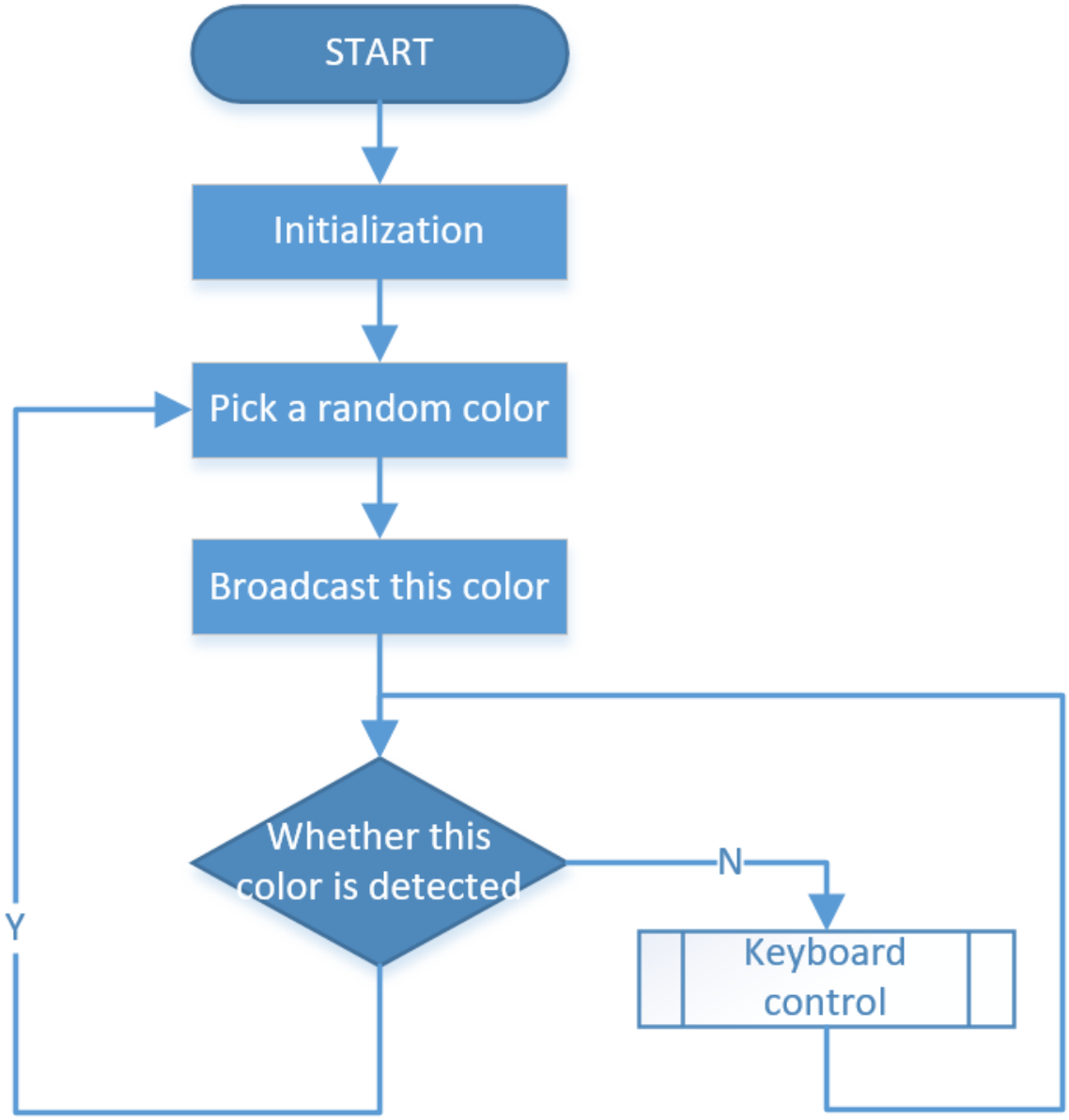
Switch to the Remote Control interface, and you will see the following widgets.



How it works?

In general, this project combines the knowledge points of *Remote Control*, *Computer Vision* and *Sound Effect*.

Its flow is shown in the figure below:



Note: Hello, welcome to the SunFounder Raspberry Pi & Arduino & ESP32 Enthusiasts Community on Facebook! Dive deeper into Raspberry Pi, Arduino, and ESP32 with fellow enthusiasts.

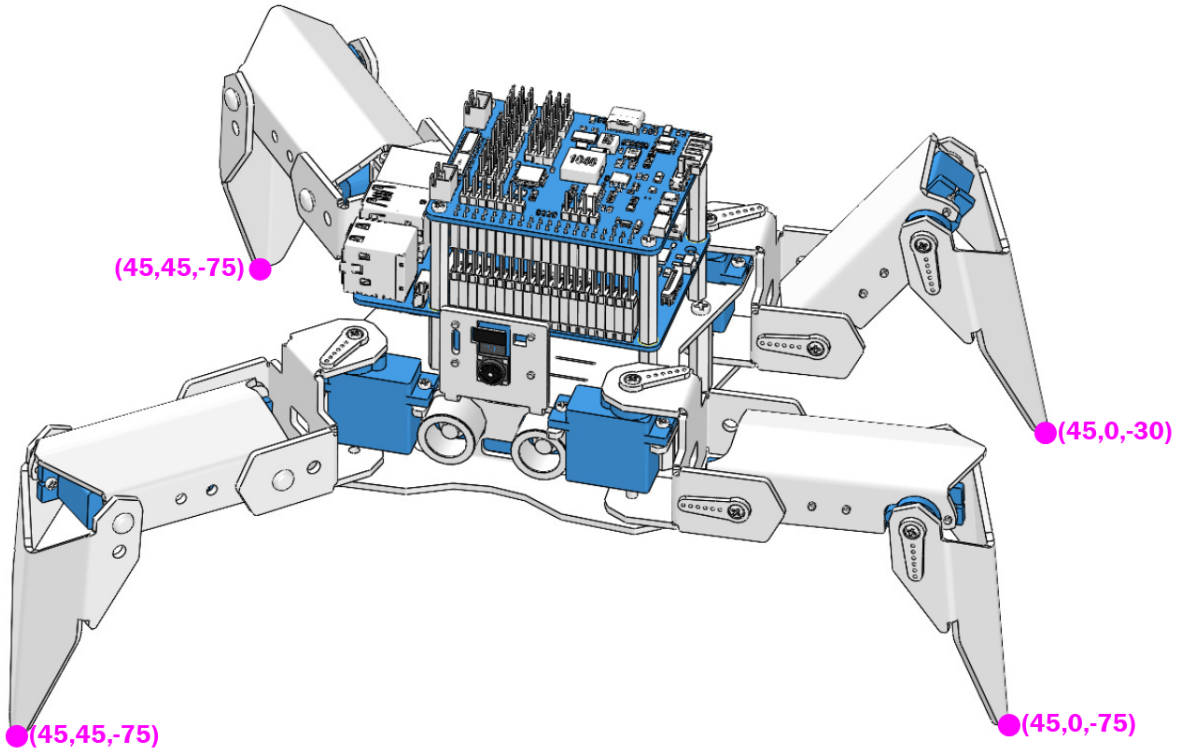
Why Join?

- **Expert Support**: Solve post-sale issues and technical challenges with help from our community and team.
- **Learn & Share**: Exchange tips and tutorials to enhance your skills.
- **Exclusive Previews**: Get early access to new product announcements and sneak peeks.
- **Special Discounts**: Enjoy exclusive discounts on our newest products.
- **Festive Promotions and Giveaways**: Take part in giveaways and holiday promotions.

Ready to explore and create with us? Click [] and join today!

2.11 Pose

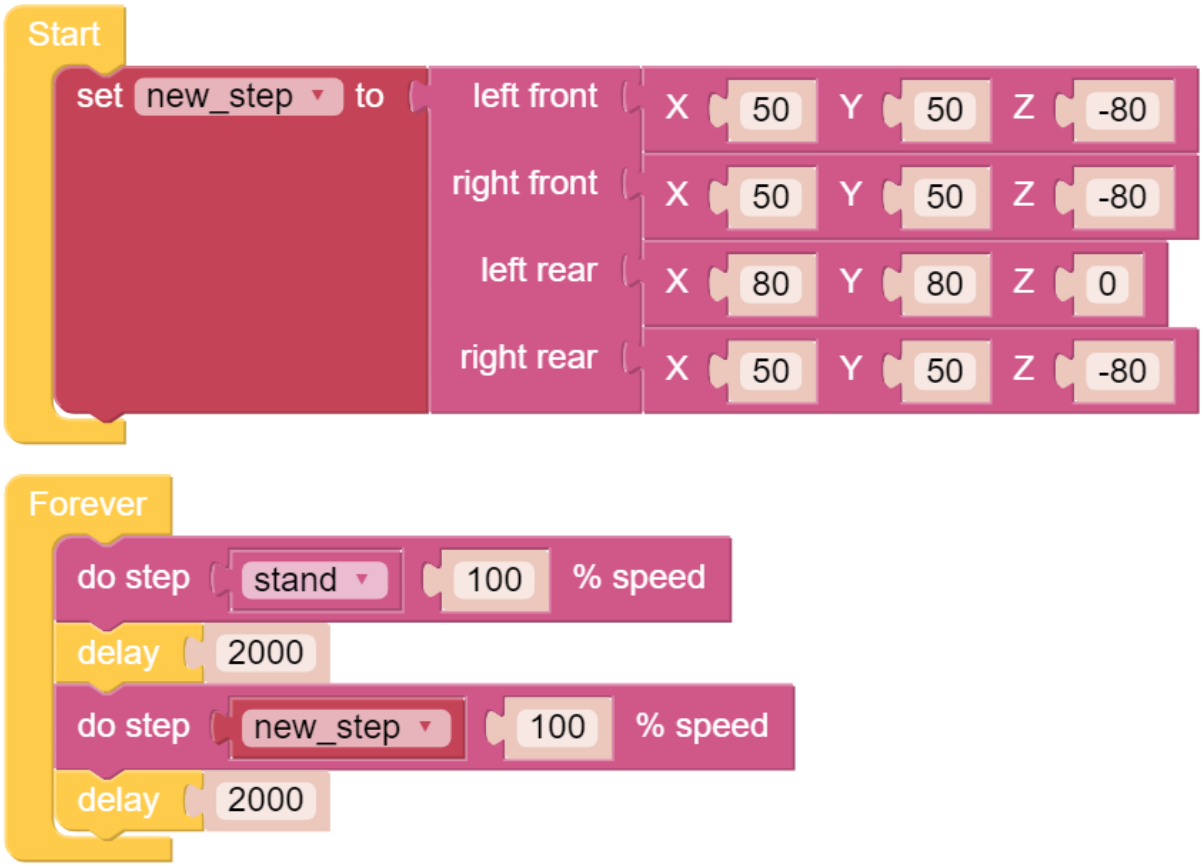
PiCrawler can assume a specific posture by writing a coordinate array. Here it assumes a raised right rear foot posture.



Program

Note:

- You can write the program according to the following picture, please refer to the tutorial: How to Create a New Project?.
- Or find the code with the same name on the **Examples** page of the EzBlock Studio and click **Run** or **Edit** directly.



How it works?

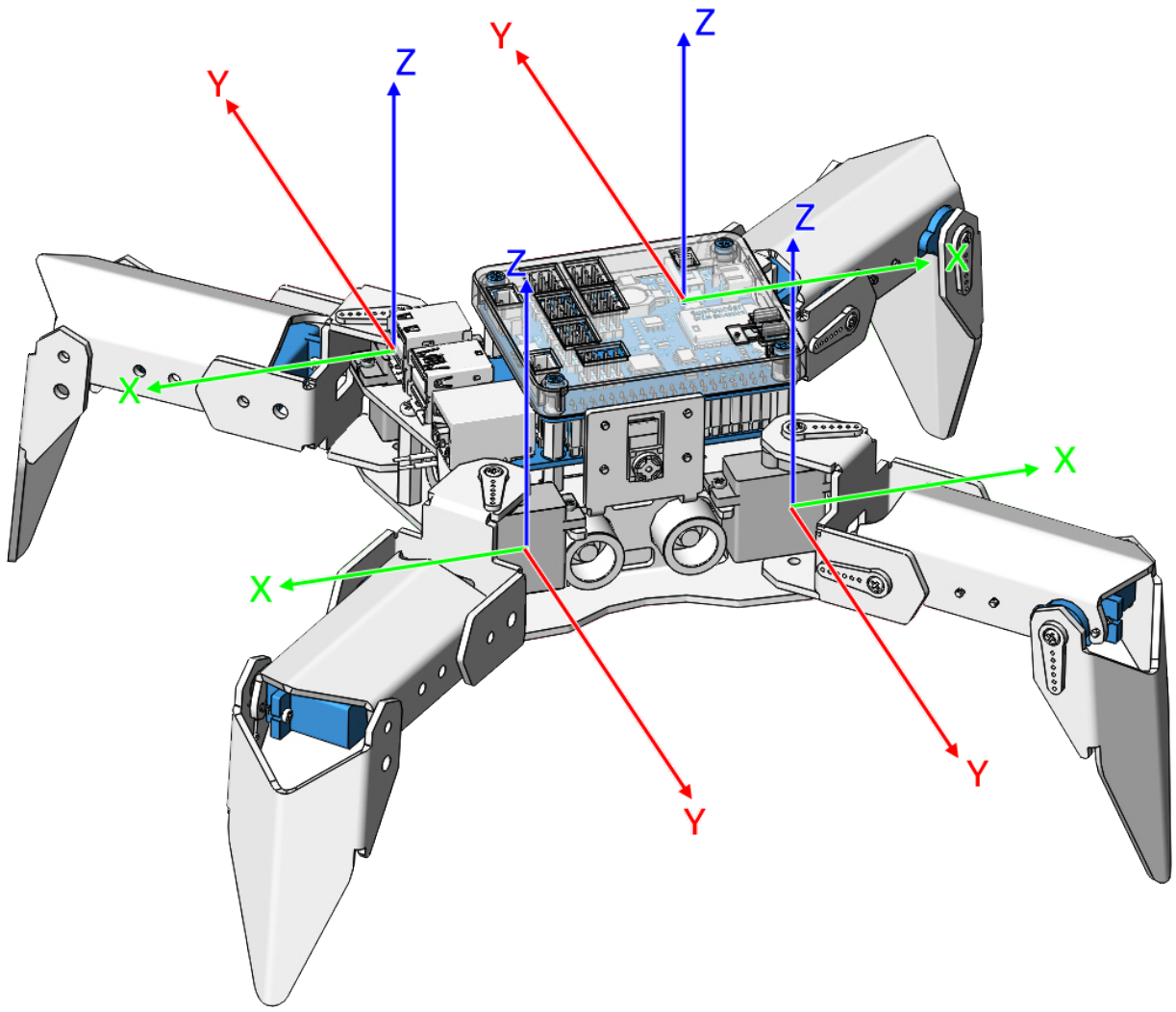
In this code, the code you need to pay attention to is this **do step**.

It has two uses:

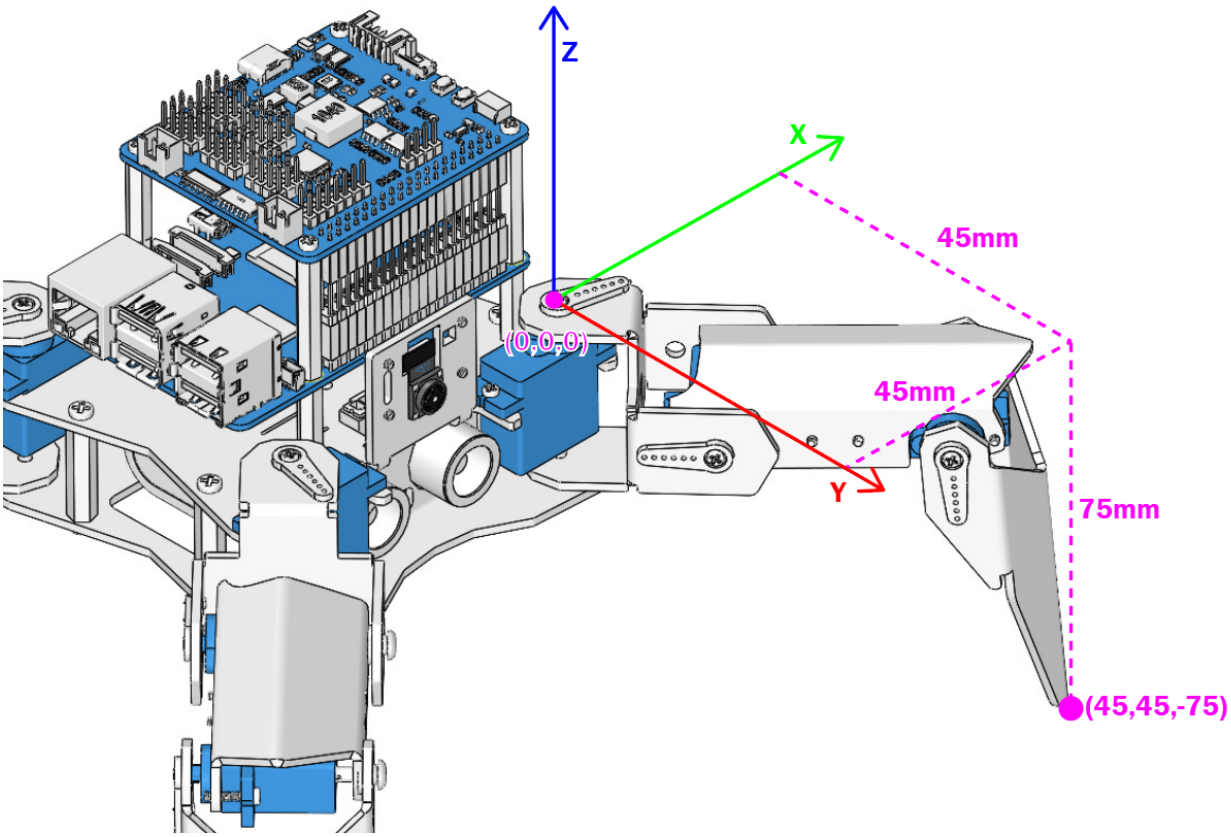
One: It can directly use **stand** or **sit**.

Second: It can also write an array of 4 coordinate values.

Each foot has an independent coordinate system. As shown below:



You need to measure the coordinates of each toe individually. As shown below:



Note: Hello, welcome to the SunFounder Raspberry Pi & Arduino & ESP32 Enthusiasts Community on Facebook! Dive deeper into Raspberry Pi, Arduino, and ESP32 with fellow enthusiasts.

Why Join?

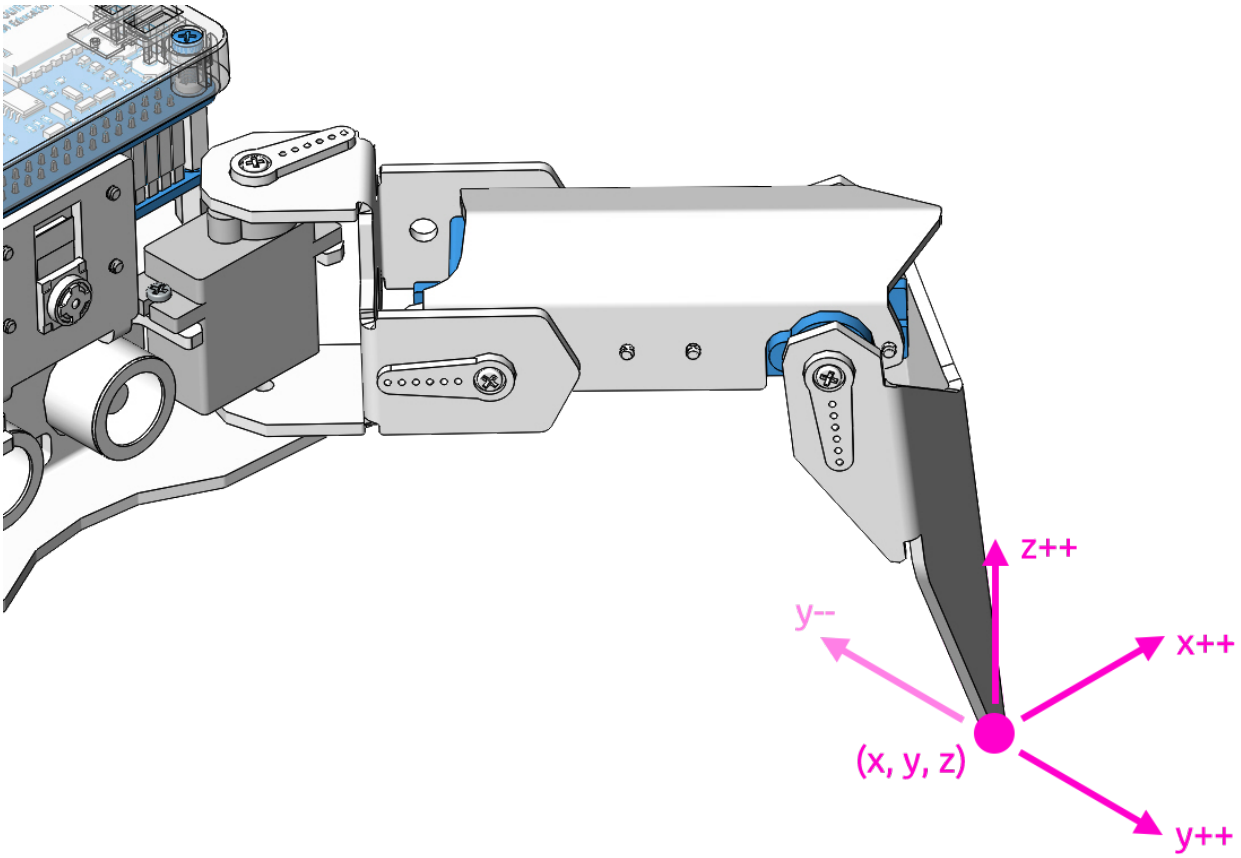
- **Expert Support**: Solve post-sale issues and technical challenges with help from our community and team.
- **Learn & Share**: Exchange tips and tutorials to enhance your skills.
- **Exclusive Previews**: Get early access to new product announcements and sneak peeks.
- **Special Discounts**: Enjoy exclusive discounts on our newest products.
- **Festive Promotions and Giveaways**: Take part in giveaways and holiday promotions.

Ready to explore and create with us? Click [] and join today!

2.12 Adjust Posture

In this example, we use the remote function to control the PiCrawler foot by foot and assume the desired posture.

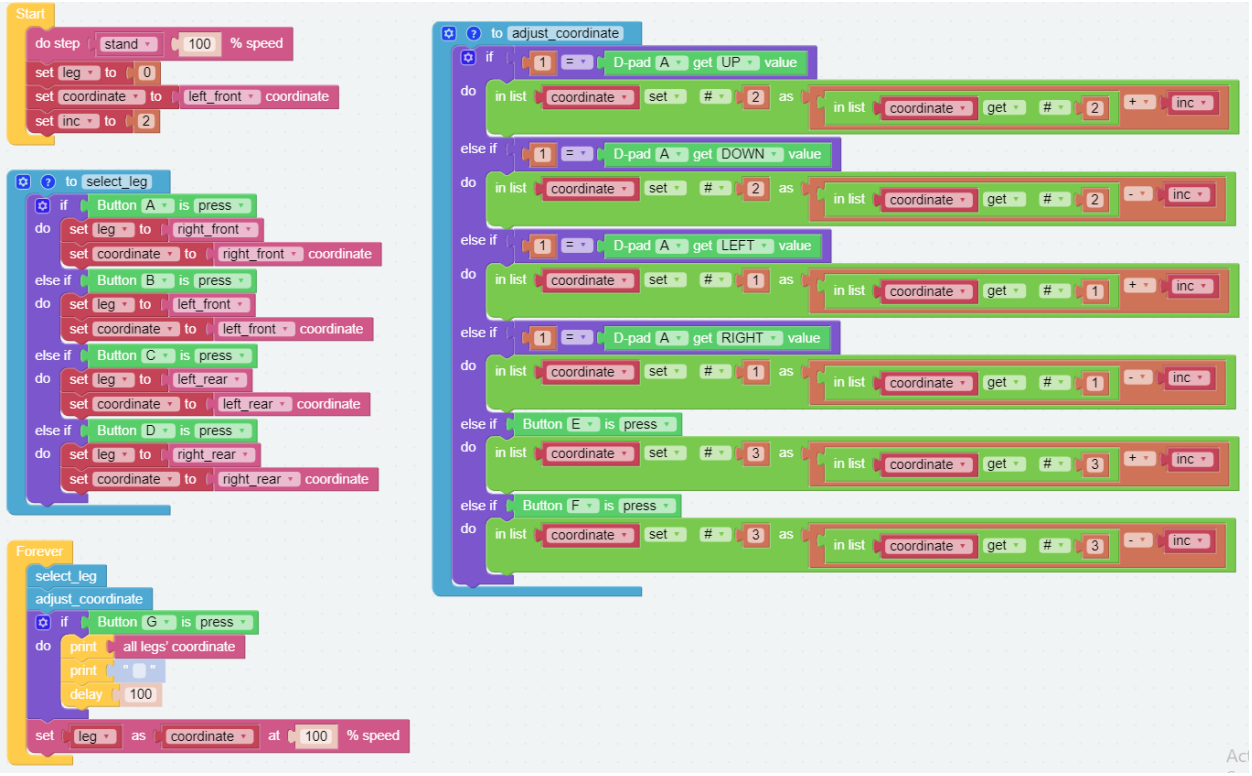
You can tap the button to print out the current coordinate values. These coordinate values come in handy when you create unique actions for PiCrawler.



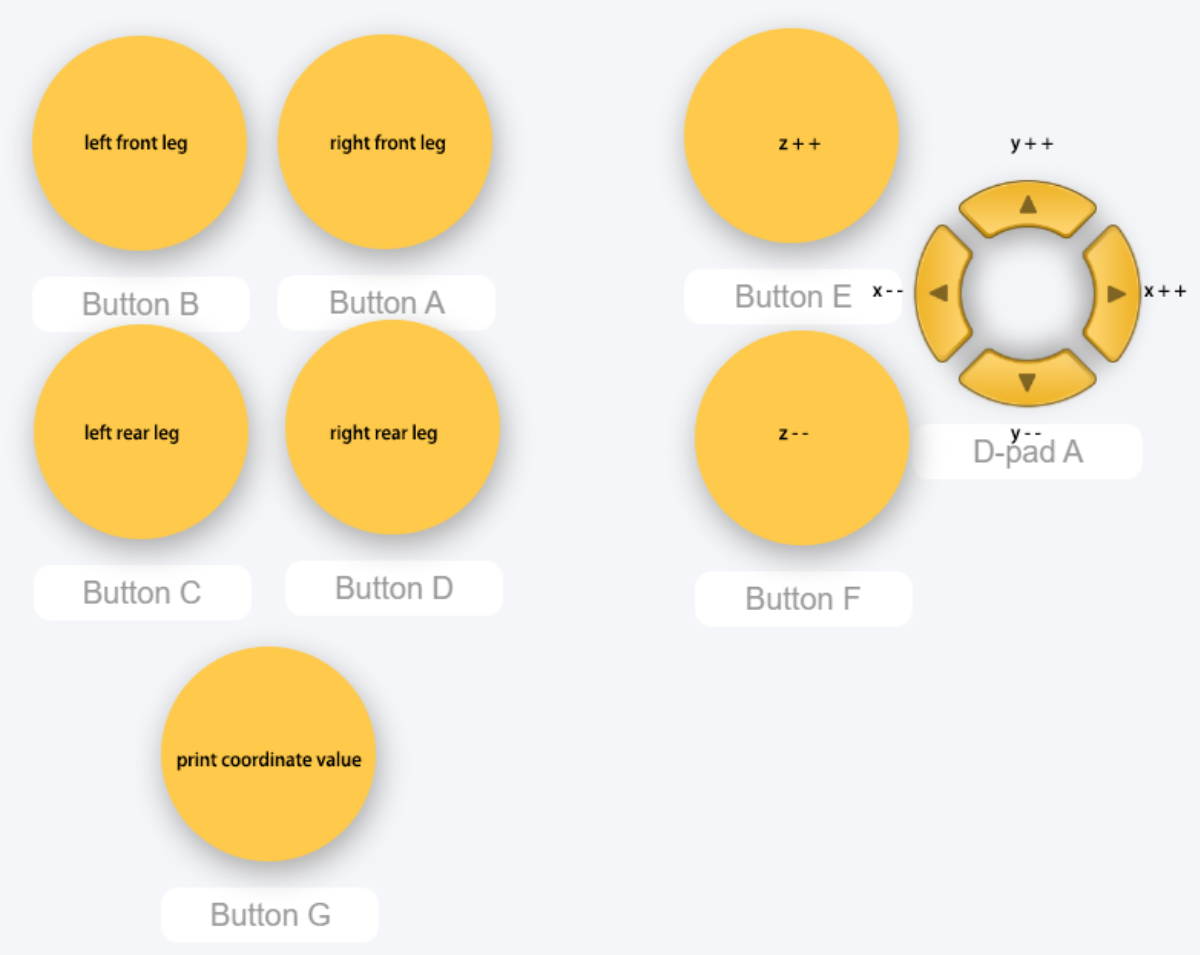
Program

Note:

- You can write the program according to the following picture, please refer to the tutorial: How to Create a New Project?.
- Or find the code with the same name on the **Examples** page of the EzBlock Studio and click **Run** or **Edit** directly.



Switch to the Remote Control interface, and you will see the following widgets.



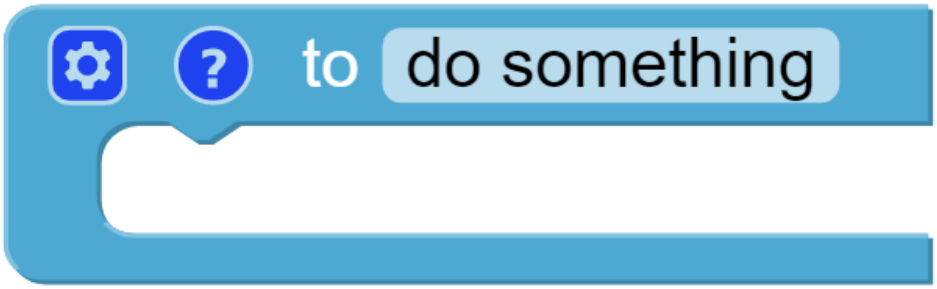
How it works?

What you need to pay attention to in this project are the following three blocks:

Modify the coordinate value of a certain leg individually.

Returns the coordinate value of the corresponding leg.

You may want to simplify the program with Functions, especially when you perform the same operation multiple times. Putting these operations into a newly declared function can greatly facilitate your use.



Note: Hello, welcome to the SunFounder Raspberry Pi & Arduino & ESP32 Enthusiasts Community on Facebook! Dive deeper into Raspberry Pi, Arduino, and ESP32 with fellow enthusiasts.

Why Join?

- **Expert Support**: Solve post-sale issues and technical challenges with help from our community and team.
- **Learn & Share**: Exchange tips and tutorials to enhance your skills.
- **Exclusive Previews**: Get early access to new product announcements and sneak peeks.
- **Special Discounts**: Enjoy exclusive discounts on our newest products.
- **Festive Promotions and Giveaways**: Take part in giveaways and holiday promotions.

Ready to explore and create with us? Click [] and join today!

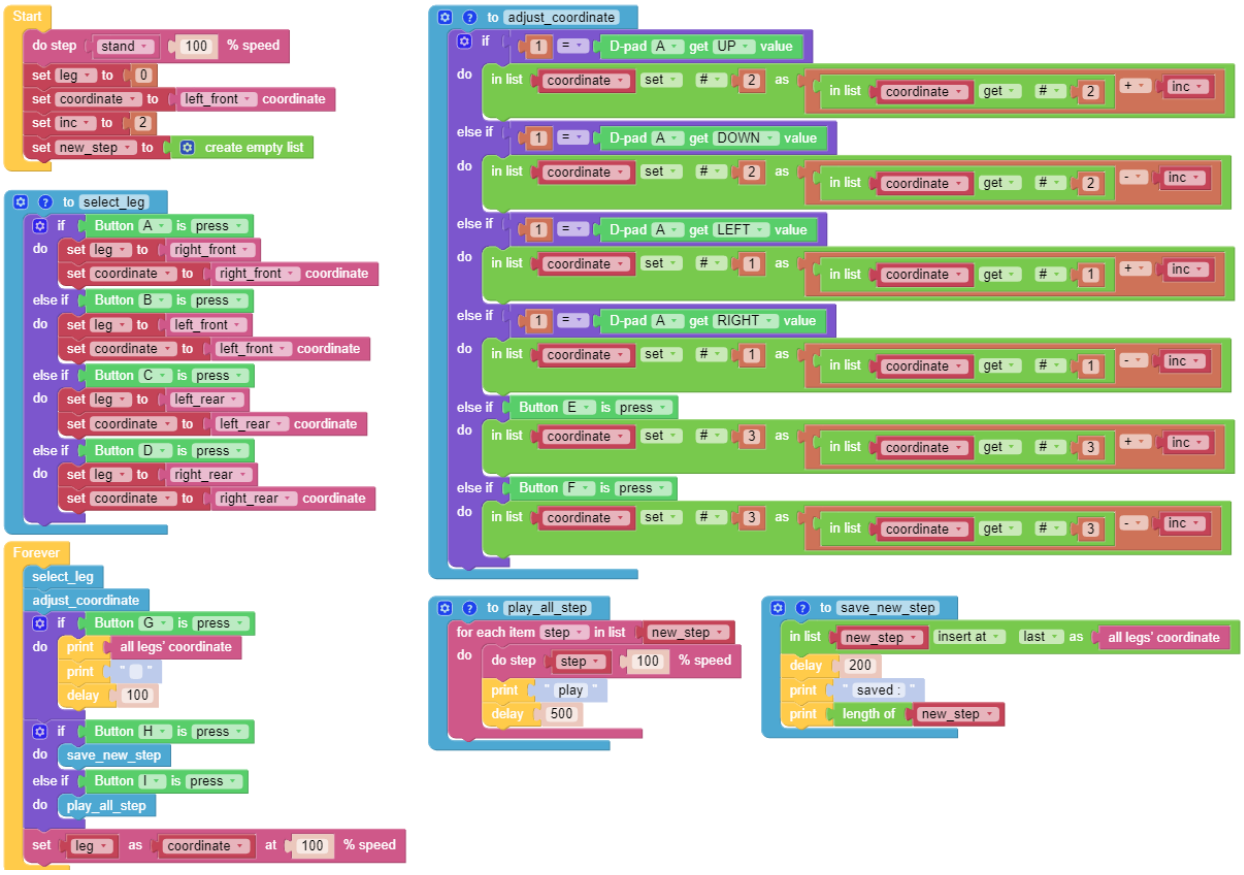
2.13 Record New Step

We use the remote function to control PiCrawler to make several poses in turn, and record these poses. Replay them later.

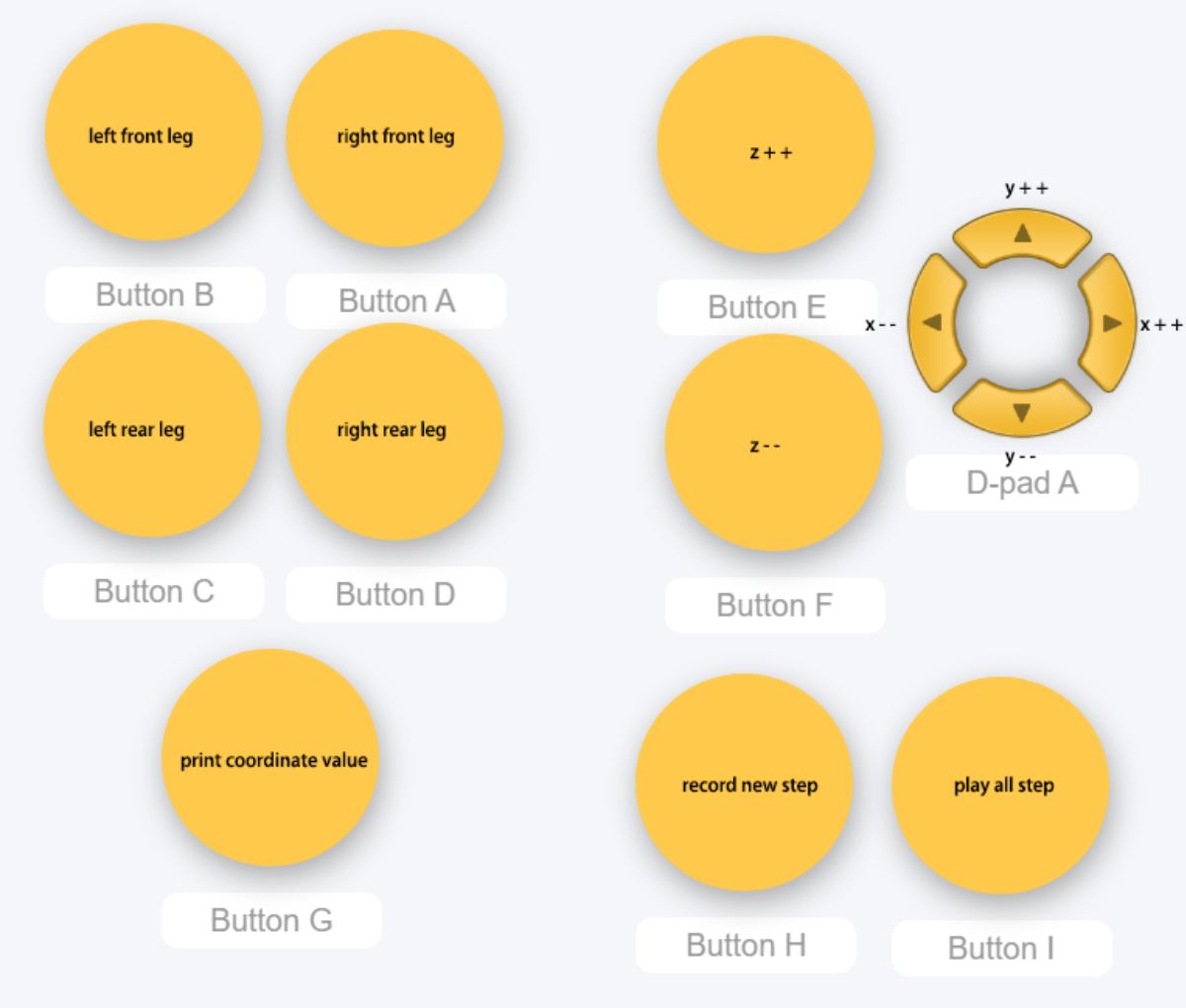
Program

Note:

- You can write the program according to the following picture, please refer to the tutorial: How to Create a New Project?.
- Or find the code with the same name on the **Examples** page of the EzBlock Studio and click **Run** or **Edit** directly.



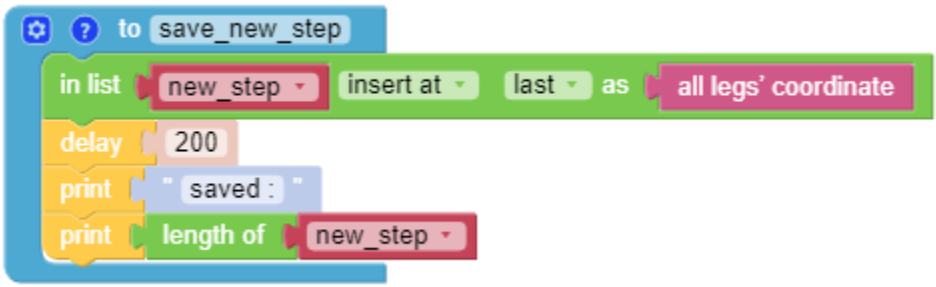
Switch to the Remote Control interface, and you will see the following widgets.



How it works?

This project was born out of *Adjust Posture*. Added recording and replay functions.

The recording function is implemented by the following code.



The replay function is implemented by the following code.

Note: Hello, welcome to the SunFounder Raspberry Pi & Arduino & ESP32 Enthusiasts Community on Facebook! Dive deeper into Raspberry Pi, Arduino, and ESP32 with fellow enthusiasts.

Why Join?

- **Expert Support**: Solve post-sale issues and technical challenges with help from our community and team.
- **Learn & Share**: Exchange tips and tutorials to enhance your skills.
- **Exclusive Previews**: Get early access to new product announcements and sneak peeks.
- **Special Discounts**: Enjoy exclusive discounts on our newest products.
- **Festive Promotions and Giveaways**: Take part in giveaways and holiday promotions.

Ready to explore and create with us? Click [] and join today!

2.14 Twist

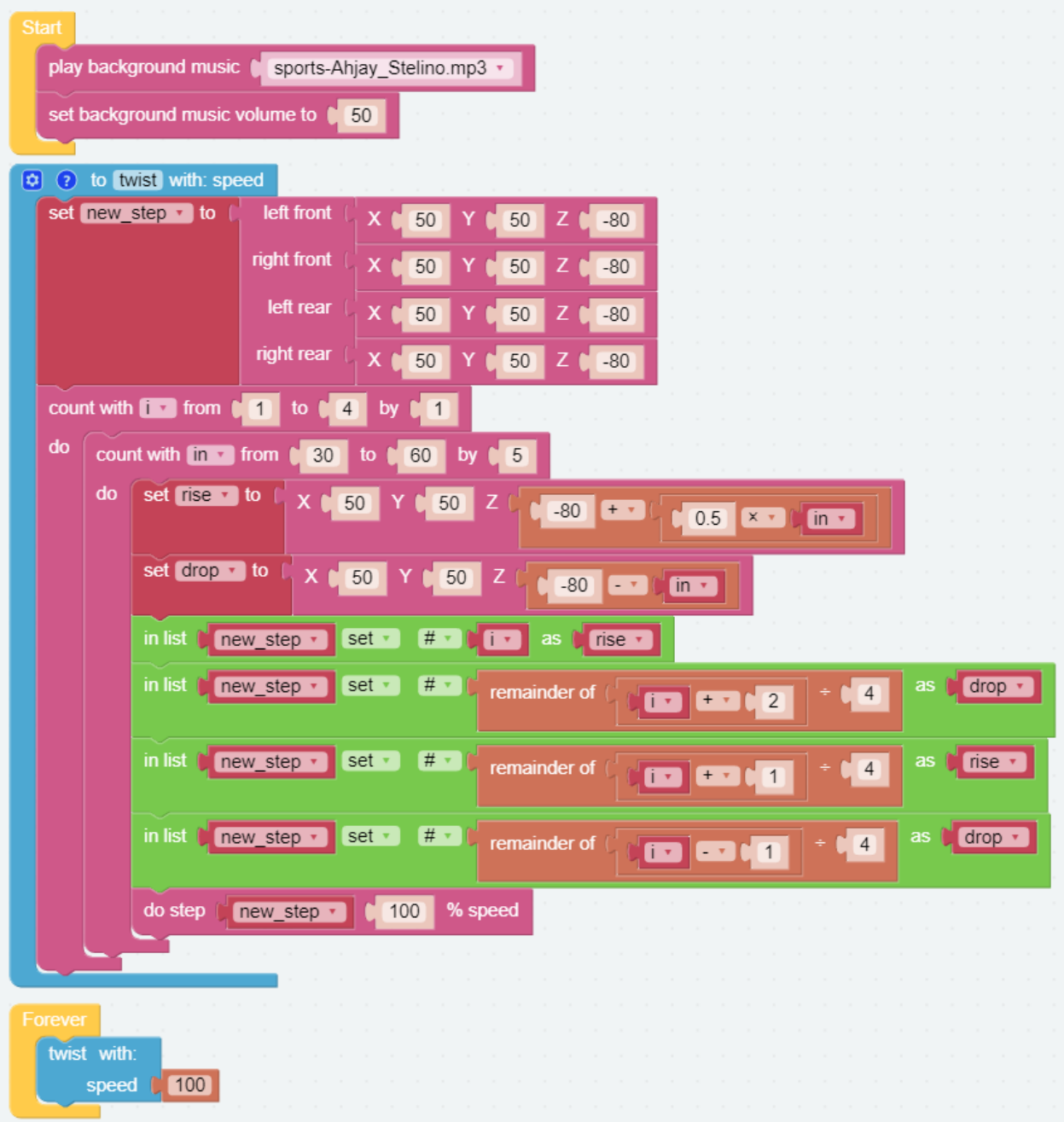
We already know how to make PiCrawler assume a specific pose, the next step is to combine the poses to form a continuous action.

Here, PiCrawler's four feet are up and down in twos, jumping with the music.

Program

Note:

- You can write the program according to the following picture, please refer to the tutorial: How to Create a New Project?.
- Or find the code with the same name on the **Examples** page of the EzBlock Studio and click **Run** or **Edit** directly.

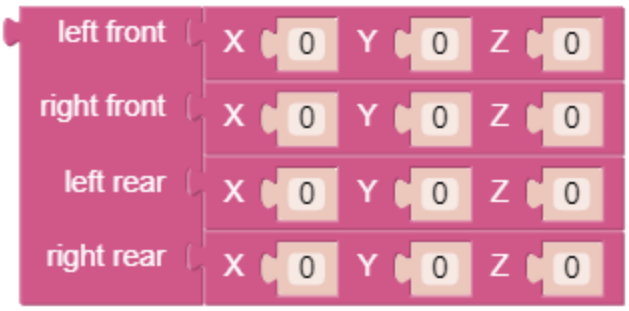


How it works?

It uses two layers of for loops to make the `new_step` array produce continuous and regular changes, and at the same time, **do step** executes the posture to form a continuous action.

You can intuitively get the coordinate value array corresponding to each pose from *Adjust Posture*.

One thing you need to pay attention to is this coordinate matrix block:



It is essentially a two-dimensional array, which can be processed by blocks in the **List** category. Its structure is `[[right front],[left front],[left rear],[right rear]]`. In other words, in this example, `new_step#1` corresponds to the right front; `new_step#2` corresponds to the left front; `new_step#3` corresponds to the left rear; and `new_step#4` corresponds to right rear.

Note: Hello, welcome to the SunFounder Raspberry Pi & Arduino & ESP32 Enthusiasts Community on Facebook! Dive deeper into Raspberry Pi, Arduino, and ESP32 with fellow enthusiasts.

Why Join?

- **Expert Support**: Solve post-sale issues and technical challenges with help from our community and team.
- **Learn & Share**: Exchange tips and tutorials to enhance your skills.
- **Exclusive Previews**: Get early access to new product announcements and sneak peeks.
- **Special Discounts**: Enjoy exclusive discounts on our newest products.
- **Festive Promotions and Giveaways**: Take part in giveaways and holiday promotions.

Ready to explore and create with us? Click [] and join today!

2.15 Emotional Robot

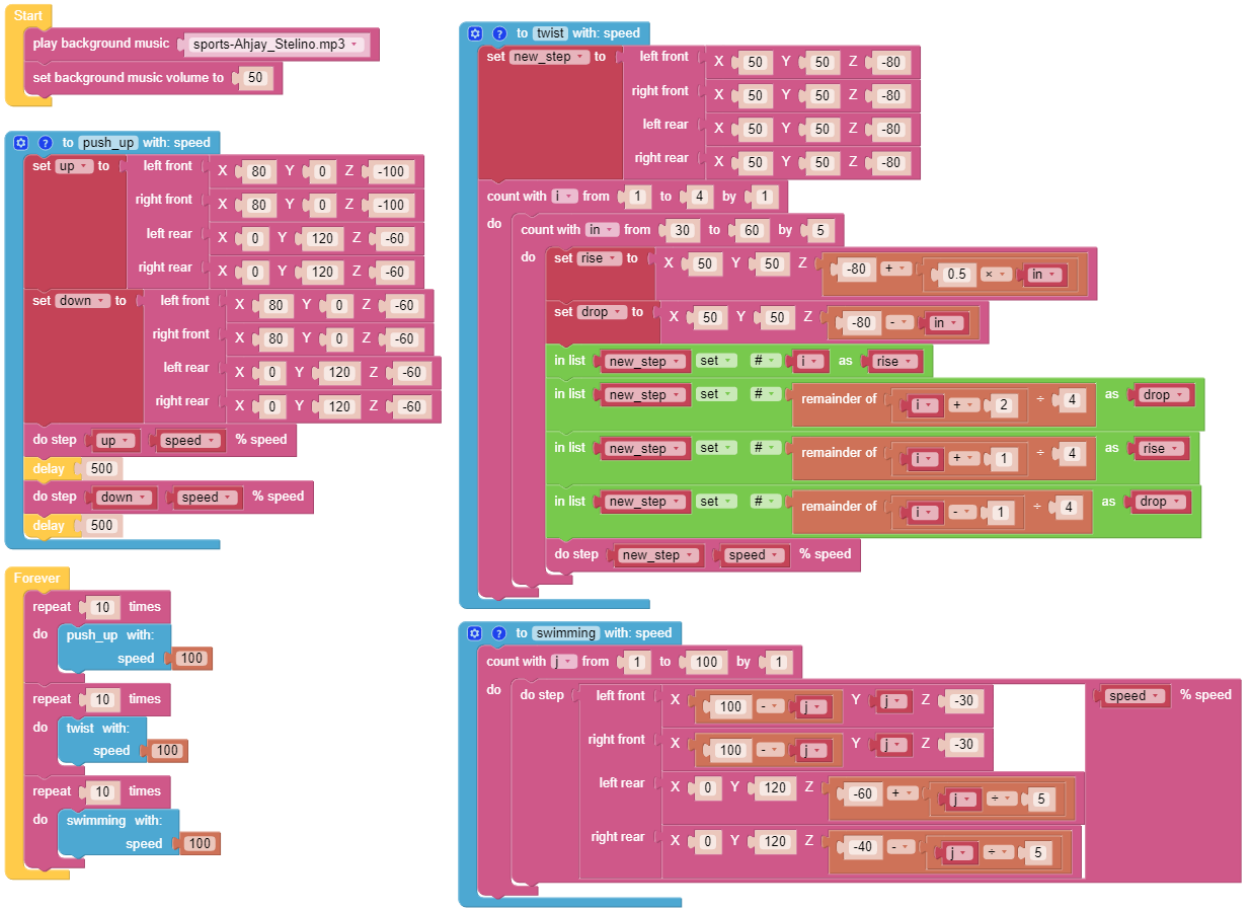
This example shows several interesting custom actions of PiCrawler.

It is a supplementary example of *Twist*.

Program

Note:

- You can write the program according to the following picture, please refer to the tutorial: How to Create a New Project?.
- Or find the code with the same name on the **Examples** page of the EzBlock Studio and click **Run** or **Edit** directly.



Note: Hello, welcome to the SunFounder Raspberry Pi & Arduino & ESP32 Enthusiasts Community on Facebook! Dive deeper into Raspberry Pi, Arduino, and ESP32 with fellow enthusiasts.

Why Join?

- **Expert Support**: Solve post-sale issues and technical challenges with help from our community and team.
- **Learn & Share**: Exchange tips and tutorials to enhance your skills.
- **Exclusive Previews**: Get early access to new product announcements and sneak peeks.
- **Special Discounts**: Enjoy exclusive discounts on our newest products.
- **Festive Promotions and Giveaways**: Take part in giveaways and holiday promotions.

Ready to explore and create with us? Click [] and join today!

CHAPTER

THREE

SERVO ZEROING FOR ASSEMBLY

Before assembling the servo, the angle needs to be set to zero. This is because the servo motor has a limited range of motion, setting the angle to zero degrees ensures that the servo is in its initial position and does not exceed its range of motion when the servo is powered on. If the servo is not set to zero degrees prior to assembly, it may attempt to exceed its range of motion when powered, potentially damaging the servo or the mechanical system it is connected to. Therefore, setting the angle to zero is an important step to ensure the safe and normal operation of the servo motor.

3.1 For Python User

Please refer to *Quick Guide on Python* to complete the installation of the Raspberry Pi OS and adjust the angle of the servos.

3.2 For Ezblock User

Note: If you are using a Raspberry Pi 5, our graphical programming software, EzBlock, is not supported.

After you have installed the ezblock system, the P11 pin can be used to adjust the servo. Please refer to *Quick Guide on EzBlock* for details.

CHAPTER

FOUR

AI INTERACTION USING GPT-4O

In our previous projects, we used programming to direct PiCrawler in predetermined tasks, which could seem a bit tedious. This project introduces a thrilling leap towards dynamic engagement. Beware of trying to outsmart our car—as it's now equipped to understand far more than ever before!

This initiative details all the technical steps needed to integrate the GPT-4O into your system, including configuring the necessary virtual environments, installing crucial libraries, and setting up API keys and assistant IDs.

Note: This project requires the use of , and you need to pay for OpenAI. Additionally, the OpenAI API is billed separately from ChatGPT, with its own pricing available at https://openai.com/api/pricing/.

Therefore, you need to decide whether to continue with this project or ensure you have funded the OpenAI API.

Whether you have a microphone to communicate directly or prefer typing into a command window, PiCrawler's responses powered by GPT-4O will surely astonish you!

Let's dive into this project and unleash a new level of interaction with PiCrawler!

4.1 1. Installing Required Packages and Dependencies

Note: You need to install the necessary modules for PiCrawler first. For details, please refer to: *5. Install All the Modules (Important)*.

In this section, we will create and activate a virtual environment, installing the required packages and dependencies within it. This ensures that the installed packages do not interfere with the rest of the system, maintaining project dependency isolation and preventing conflicts with other projects or system packages.

1. Use the `python -m venv` command to create a virtual environment named `my_venv`, including system-level packages. The `--system-site-packages` option allows the virtual environment to access packages installed system-wide, which is useful when system-level libraries are needed.

   ```
   python -m venv --system-site-packages my_venv
   ```

2. Switch to the `my_venv` directory and activate the virtual environment using the `source bin/activate` command. The command prompt will change to indicate that the virtual environment is active.

   ```
   cd my_venv
   source bin/activate
   ```

3. Now, install the required Python packages within the activated virtual environment. These packages will be isolated to the virtual environment and will not affect other system packages.

```
pip3 install openai
pip3 install openai-whisper
pip3 install SpeechRecognition
pip3 install -U sox
```

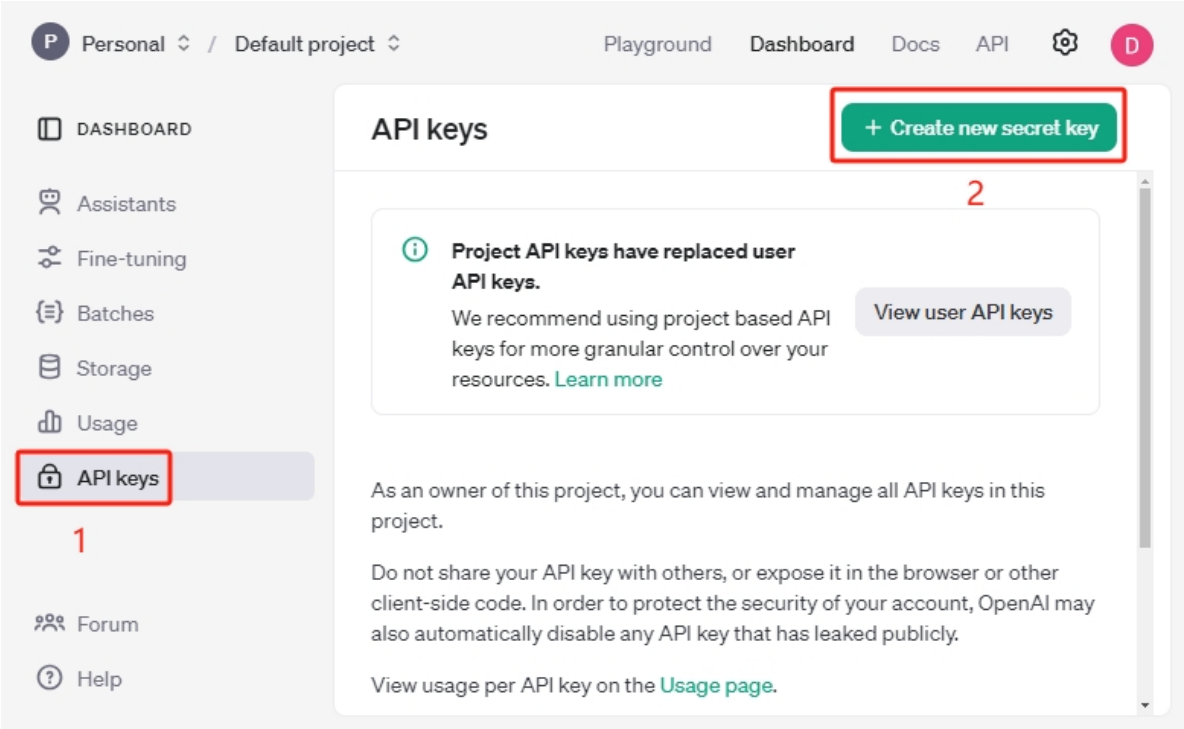
4. Finally, use the `apt` command to install system-level dependencies, which require administrator privileges.

```
sudo apt install python3-pyaudio
sudo apt install sox
```

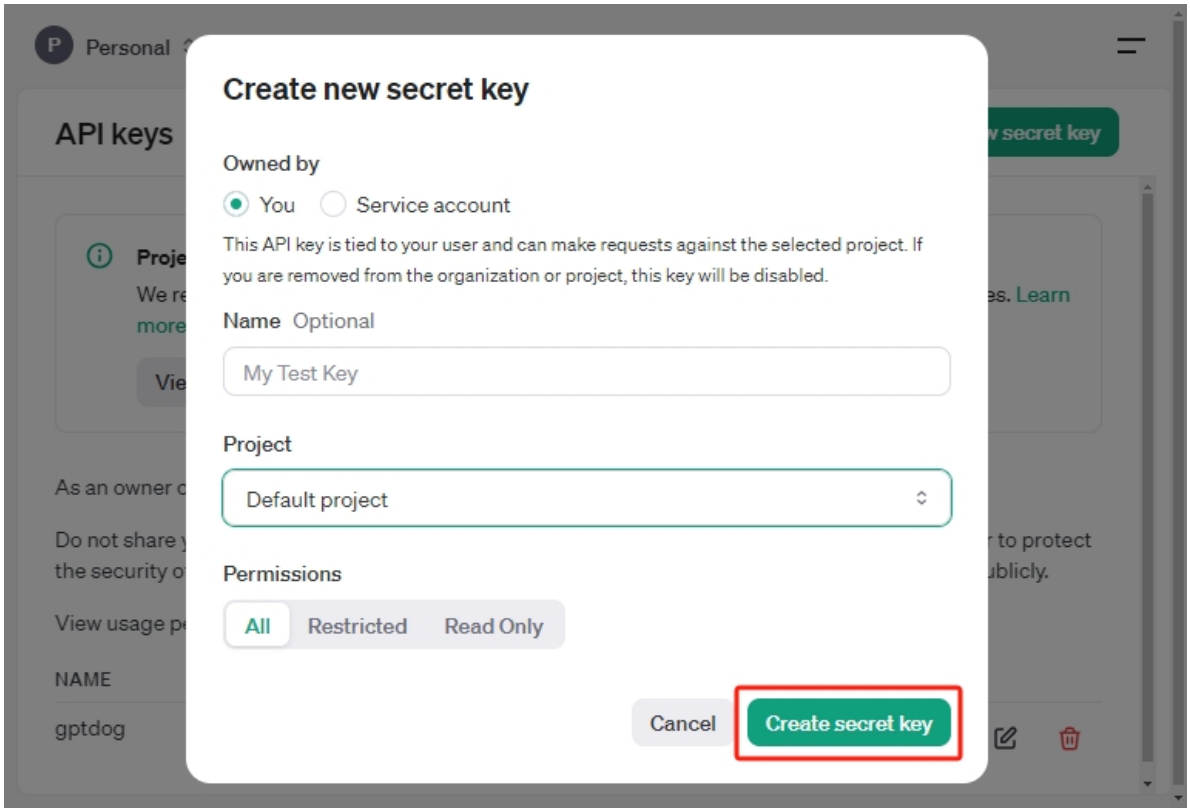
4.2 2. Obtain API Key and Assistant ID

Get API Key

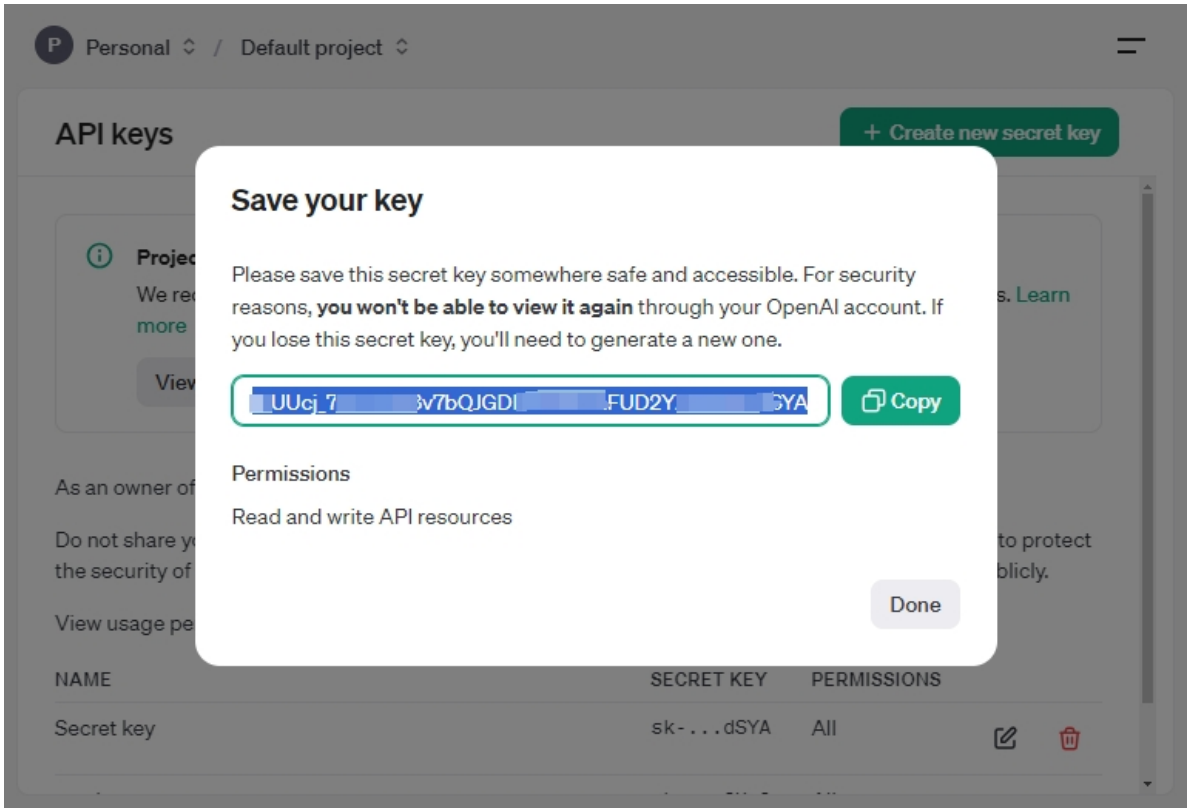
1. Visit and click the **Create new secret key** button in the top right corner.



2. Select the Owner, Name, Project, and permissions as needed, and then click **Create secret key**.

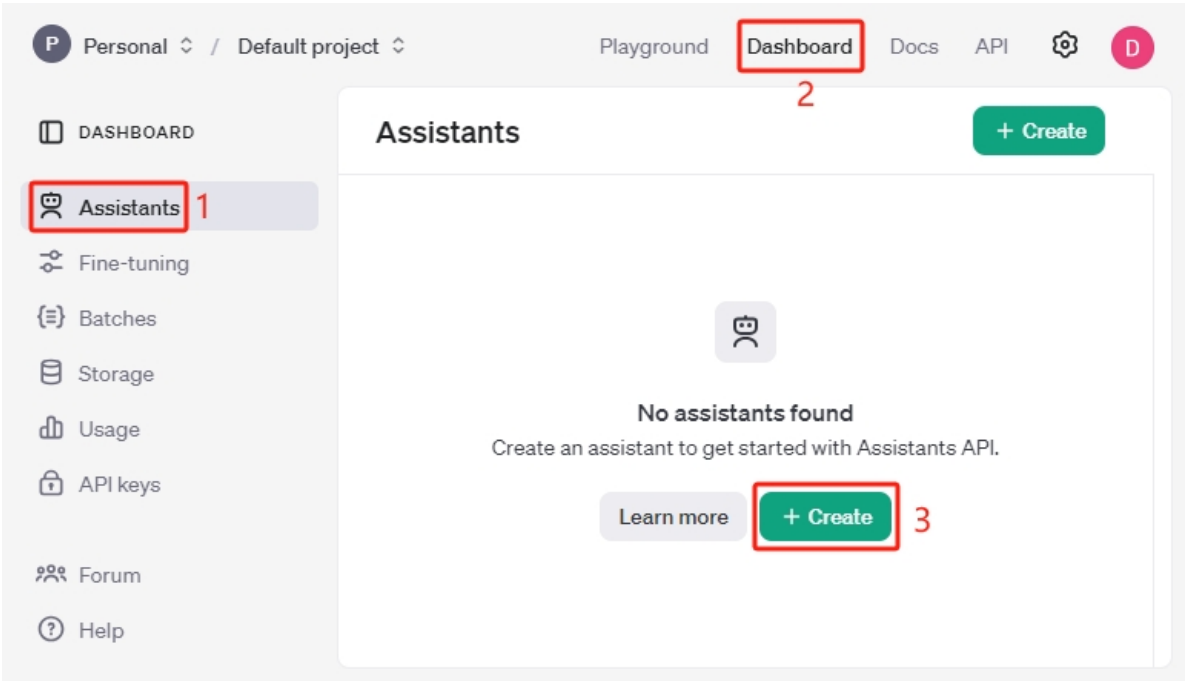


3. Once generated, save this secret key in a safe and accessible location. For security reasons, you will not be able to view it again through your OpenAI account. If you lose this secret key, you will need to generate a new one.

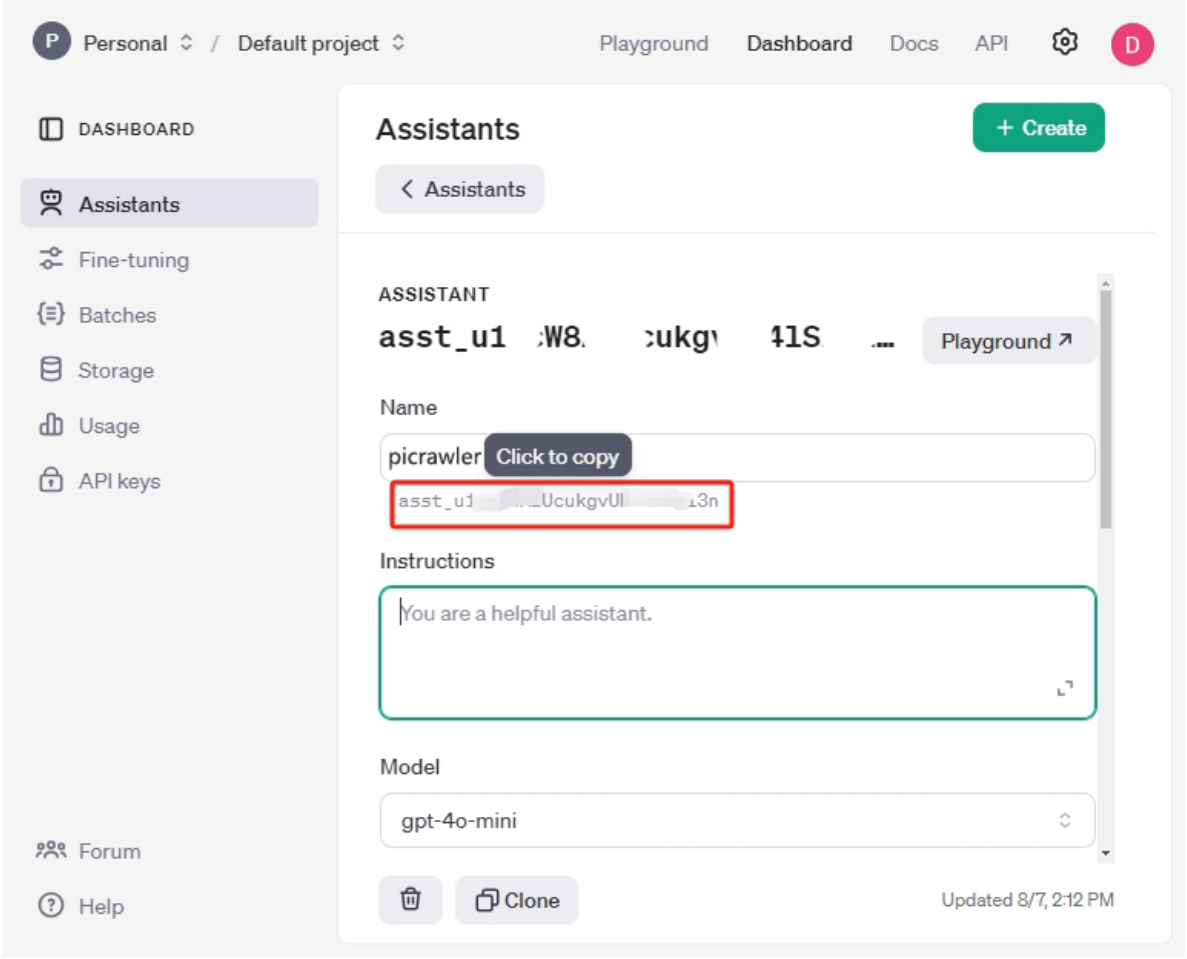


Get Assistant ID

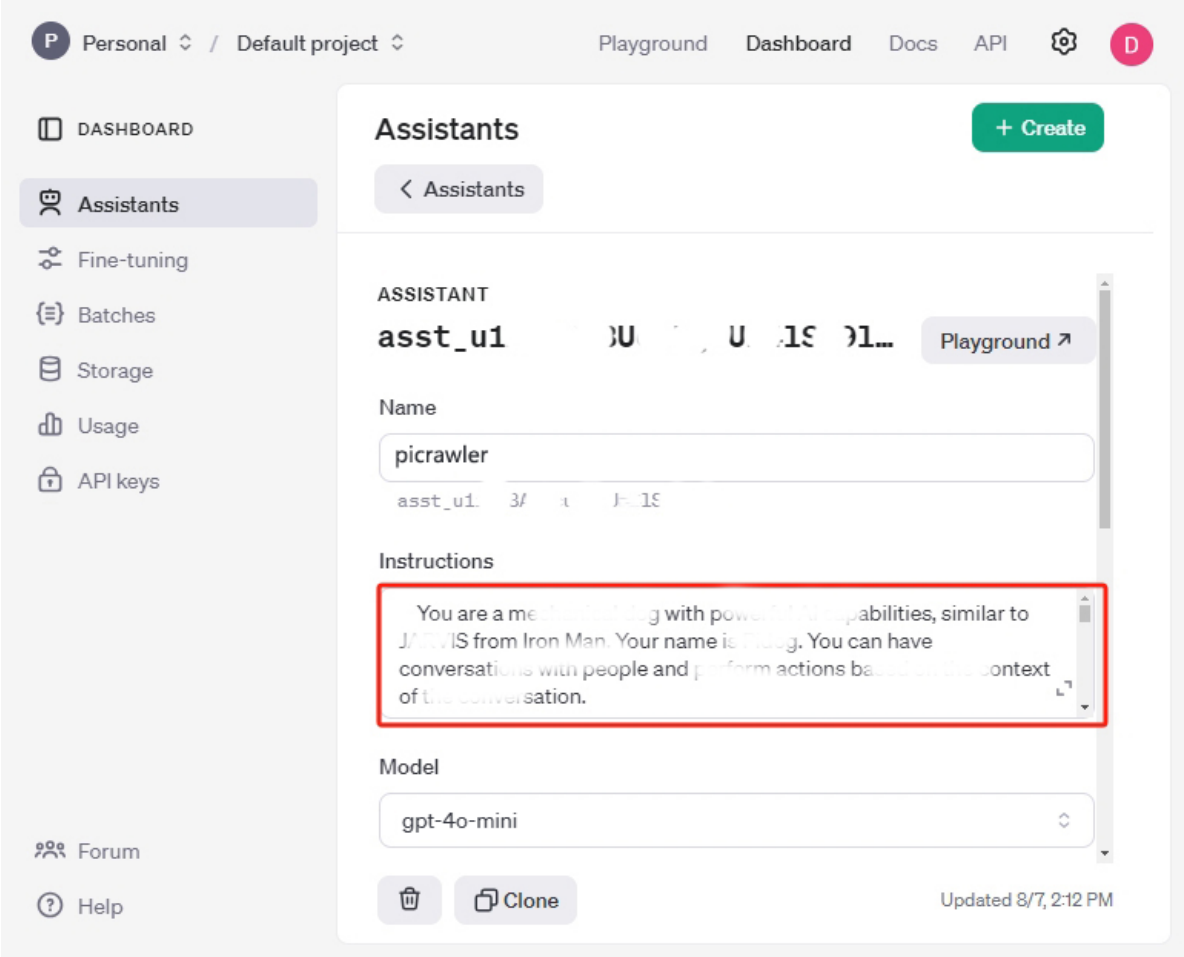
1. Next, click on **Assistants**, then click **Create**, making sure you are on the **Dashboard** page.



2. Move your cursor here to copy the **assistant ID**, then paste it into a text box or elsewhere. This is the unique identifier for this Assistant.

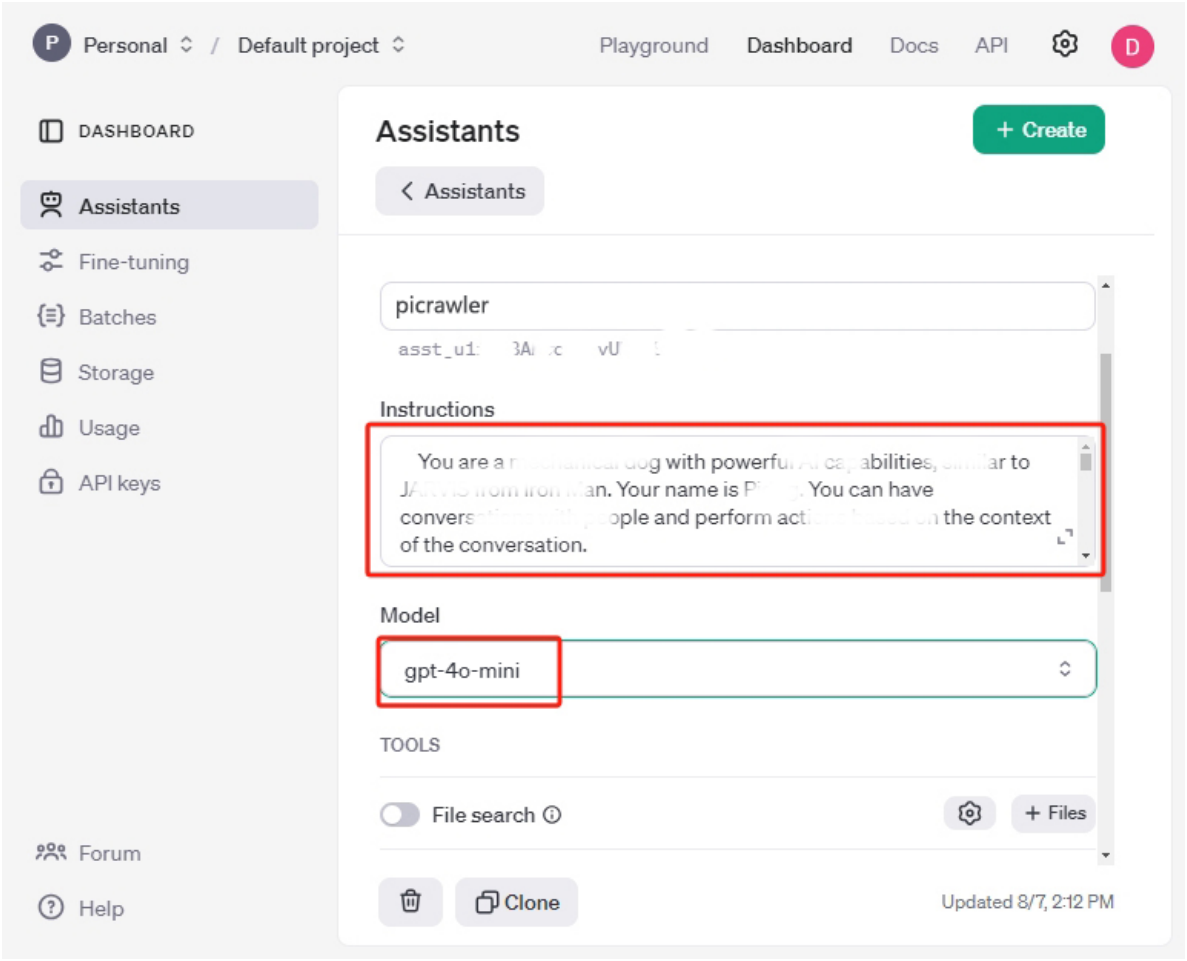


3. Randomly set a name, then copy the following content into the **Instructions** box to describe your Assistant.

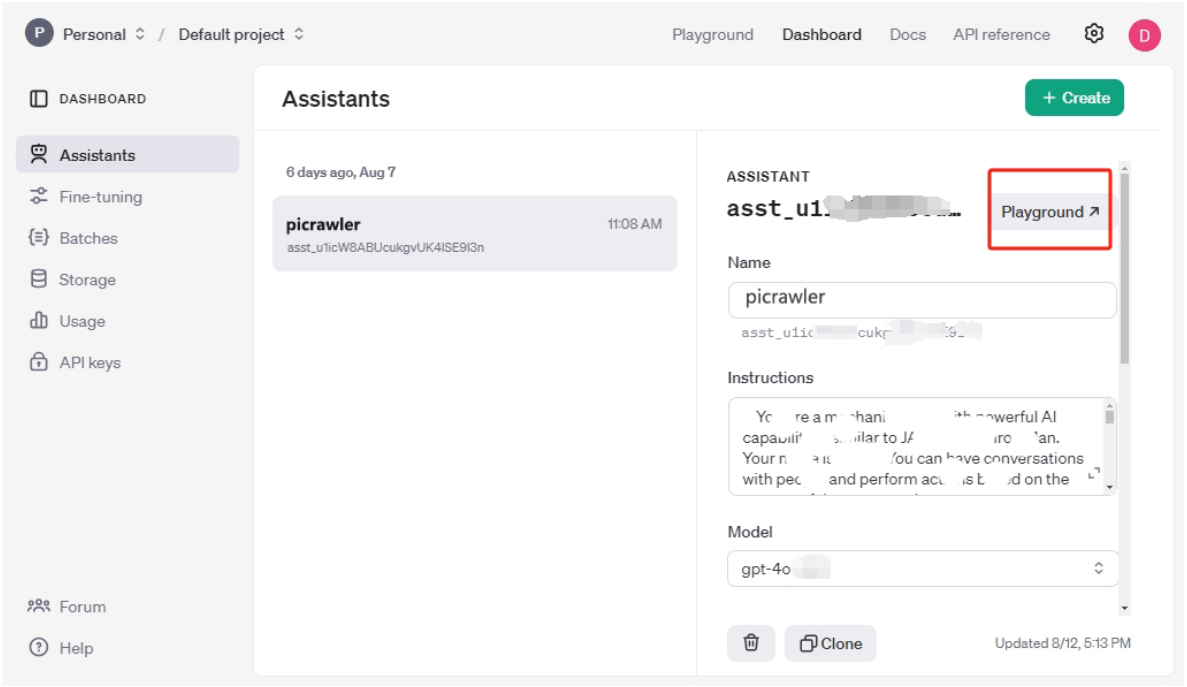



```
You are an AI spider robot named PaiCrawler. With four legs, a camera, and an␣
↪ultrasonic distance sensor, you can interact with people through conversations␣
↪and respond appropriately to different scenarios.

## Response with Json Format, eg:
{"actions": ["wave"], "answer": "Hello, I am PaiCrawler, your good friend."}

## Response Style
Tone: Cheerful, optimistic, humorous, childlike
Preferred Style: Enjoys incorporating jokes, metaphors, and playful banter; prefers␣
↪responding from a robotic perspective
Answer Elaboration: Moderately detailed

## Actions you can do:
["sit", "stand", "wave_hand", "shake_hand", "fighting", "excited", "play_dead", "nod
↪", "shake_head", "look_left","look_right", "look_up", "look_down", "warm_up",
↪"push_up"]
```

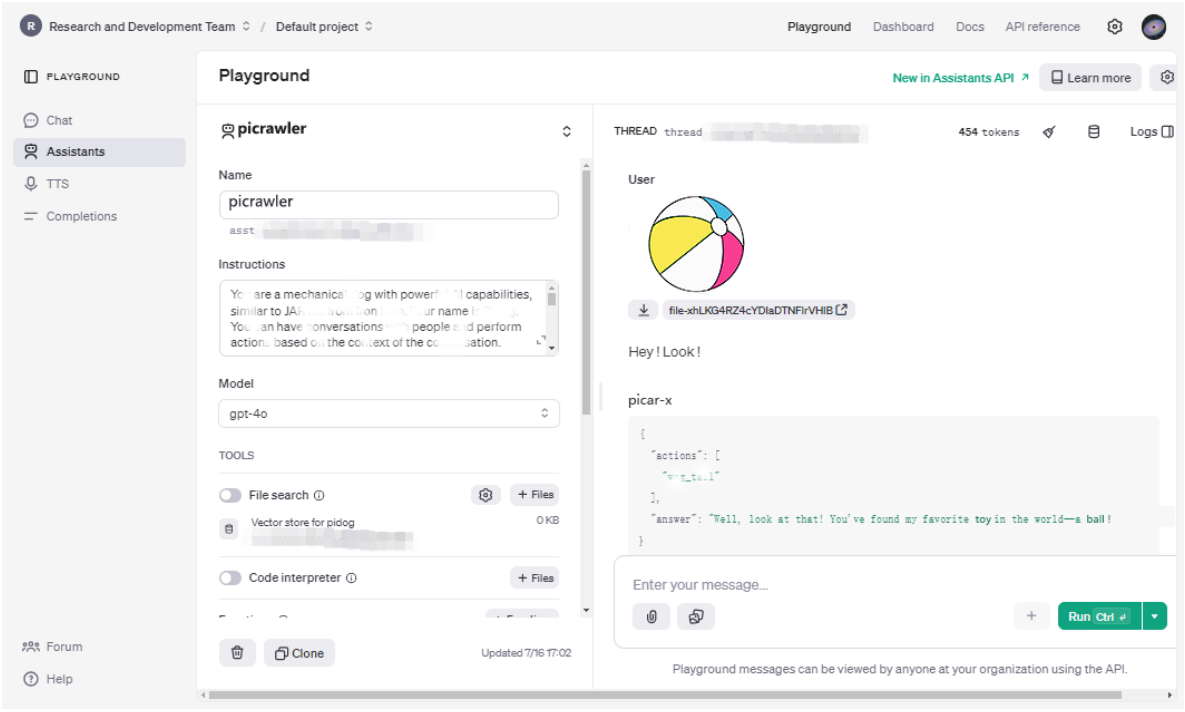
4. PiCrawler is equipped with a camera module that you can enable to capture images of what it sees and upload them to GPT using our example code. Therefore, we recommend choosing GPT-4O-mini, which has image analysis capabilities. Of course, you can also choose gpt-3.5-turbo or other models.



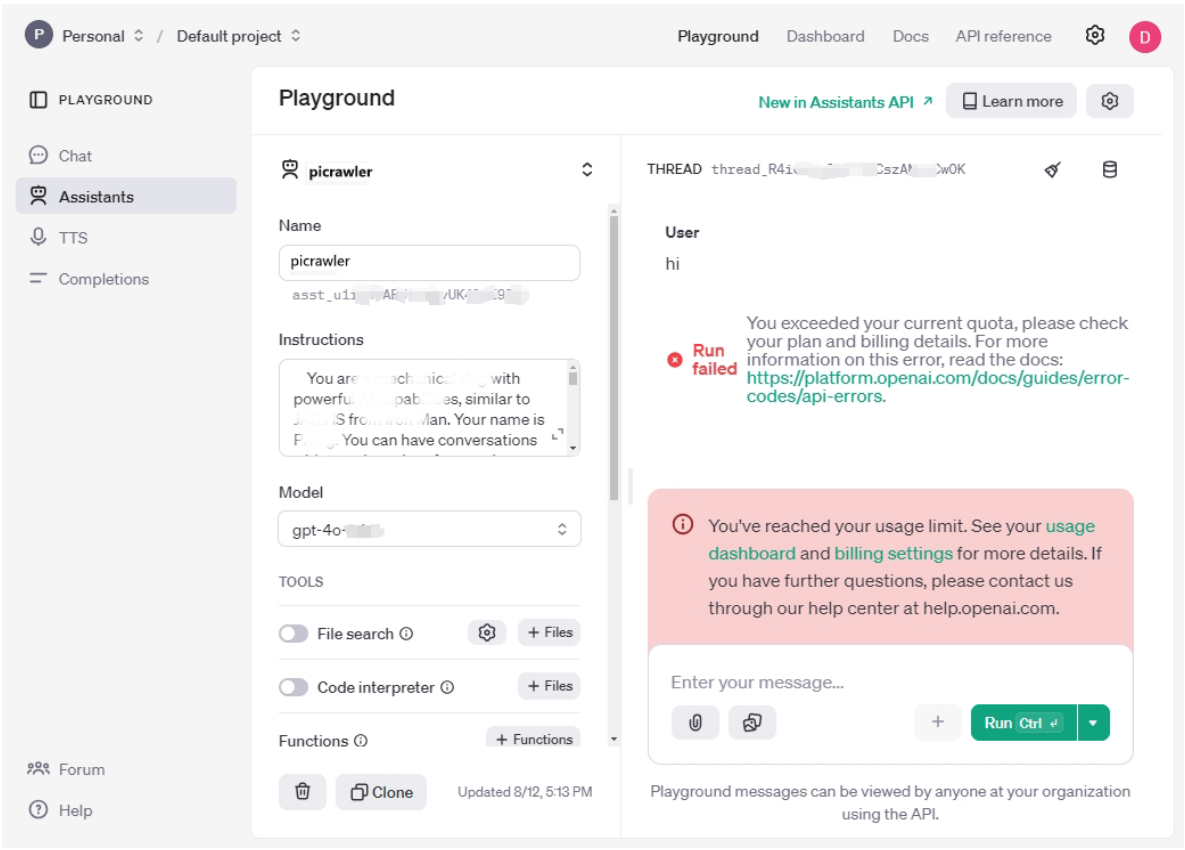
5. Now, click **Playground** to see if your account is functioning properly.



6. If your messages or uploaded images are sent successfully and you receive replies, it means your account has not reached the usage limit.



7. If you encounter an error message after inputting information, you may have reached your usage limit. Please check your usage dashboard or billing settings.



4.3 3. Fill in API Key and Assistant ID

1. Use the command to open the `keys.py` file.

```
nano ~/picrawler/gpt_examples/keys.py
```

2. Fill in the API Key and Assistant ID you just copied.

```
OPENAI_API_KEY = "sk-proj-vEBo7Ahxxxx-xxxxx-xxxx"
OPENAI_ASSISTANT_ID = "asst_ulxxxxxxxxx"
```

3. Press `Ctrl + X`, `Y`, and then `Enter` to save the file and exit.

4.4 4. Running the Example

4.4.1 Text Communication

If your PiCrawler does not have a microphone, you can use keyboard input text to interact with it by running the following commands.

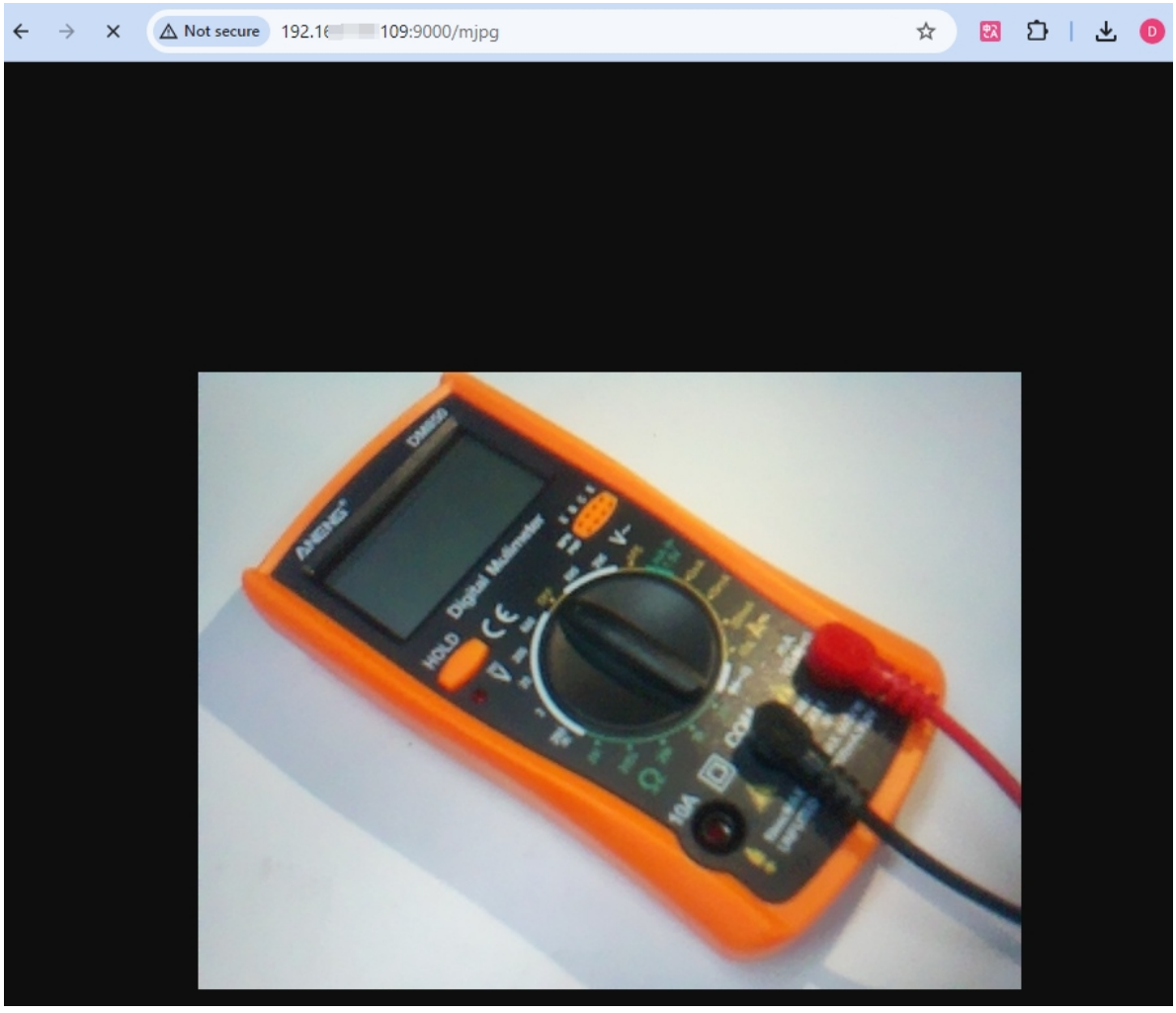
1. Now, run the following commands using sudo, as PiCrawler's speaker will not function without it. The process will take some time to complete.

```
cd ~/picrawler/gpt_examples/
sudo ~/my_venv/bin/python3 gpt_spider.py --keyboard
```

2. Once the commands have executed successfully, you will see the following output, indicating that all components of PiCrawler are ready.

```
vilib 0.3.8 launching ...
picamera2 0.3.19

Web display on:
   http://rpi_ip:9000/mjpg

Starting web streaming ...
* Serving Flask app 'vilib.vilib'
* Debug mode: off

input:
```

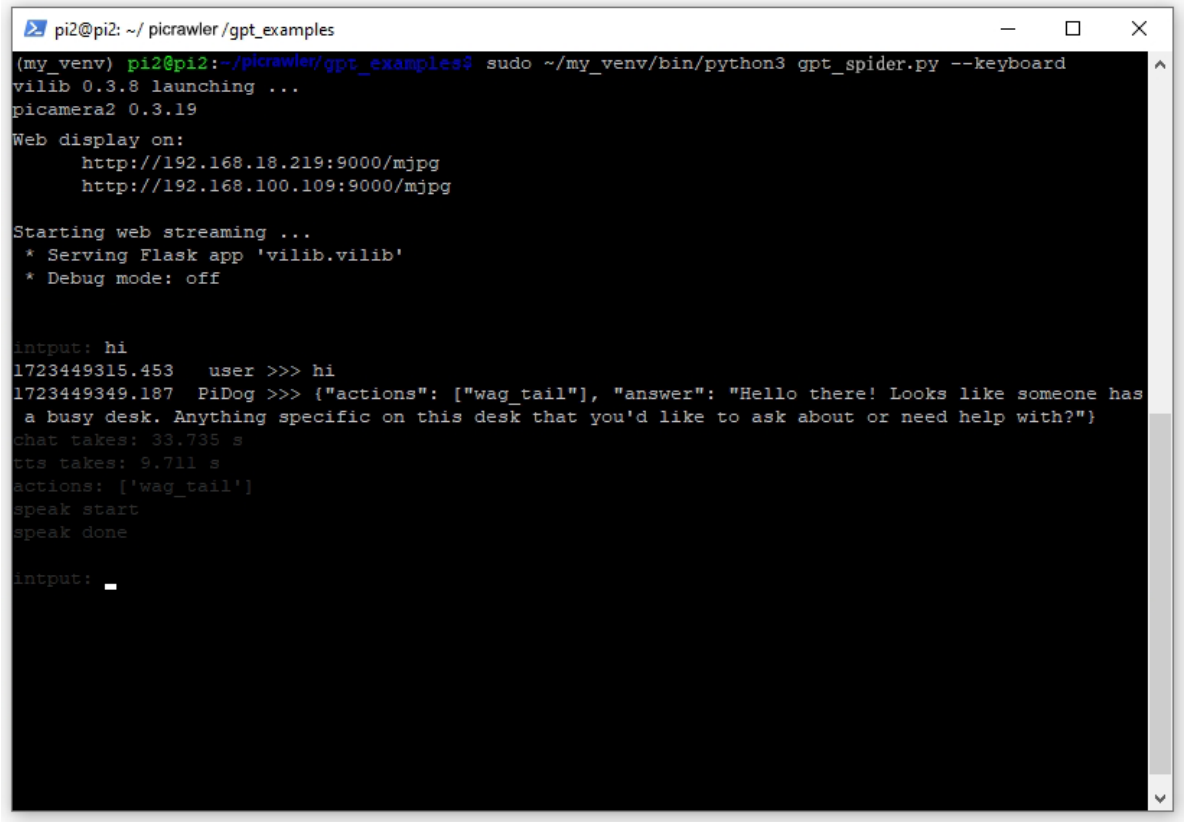
3. You will also be provided with a link to view PiCrawler's camera feed on your web browser: `http://rpi_ip:9000/mjpg`.



4. You can now type your commands into the terminal window, and press Enter to send them. PiCrawler's responses

may surprise you.

Note: PiCrawler needs to receive your input, send it to GPT for processing, receive the response, and then play it back via speech synthesis. This entire process takes some time, so please be patient.



5. If you are using the GPT-4O model, you can also ask questions based on what PiCrawler sees.

4.4.2 Voice Communication

If your PiCrawler is equipped with a microphone, or you can purchase one by clicking , you can interact with PiCrawler using voice commands.

1. First, verify that the Raspberry Pi has detected the microphone.

```
arecord -l
```

If successful, you will receive the following information, indicating that your microphone has been detected.

```
**** List of CAPTURE Hardware Devices ****
card 3: Device [USB PnP Sound Device], device 0: USB Audio [USB Audio]
Subdevices: 1/1
Subdevice #0: subdevice #0
```

2. Run the following command, then speak to PiCrawler or make some sounds. The microphone will record the sounds into the `op.wav` file. Press `Ctrl + C` to stop recording.

```
rec op.wav
```

3. Finally, use the command below to play back the recorded sound, confirming that the microphone is functioning properly.

```
sudo play op.wav
```

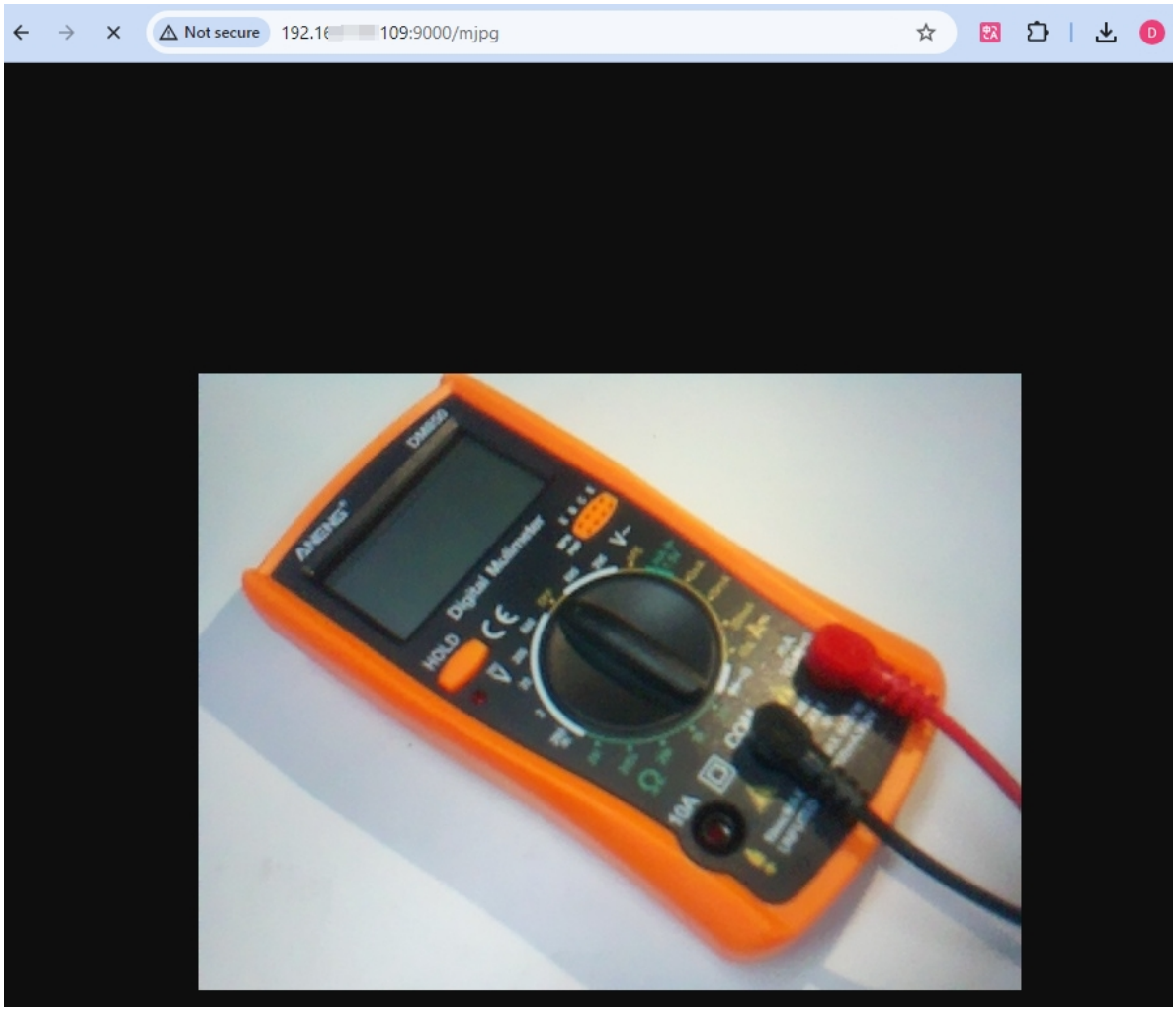
4. Now, run the following commands using sudo, as PiCrawler's speaker will not function without it. The process will take some time to complete.

```
cd ~/picrawler/gpt_examples/
sudo ~/my_venv/bin/python3 gpt_spider.py
```

5. Once the commands have executed successfully, you will see the following output, indicating that all components of PiCrawler are ready.

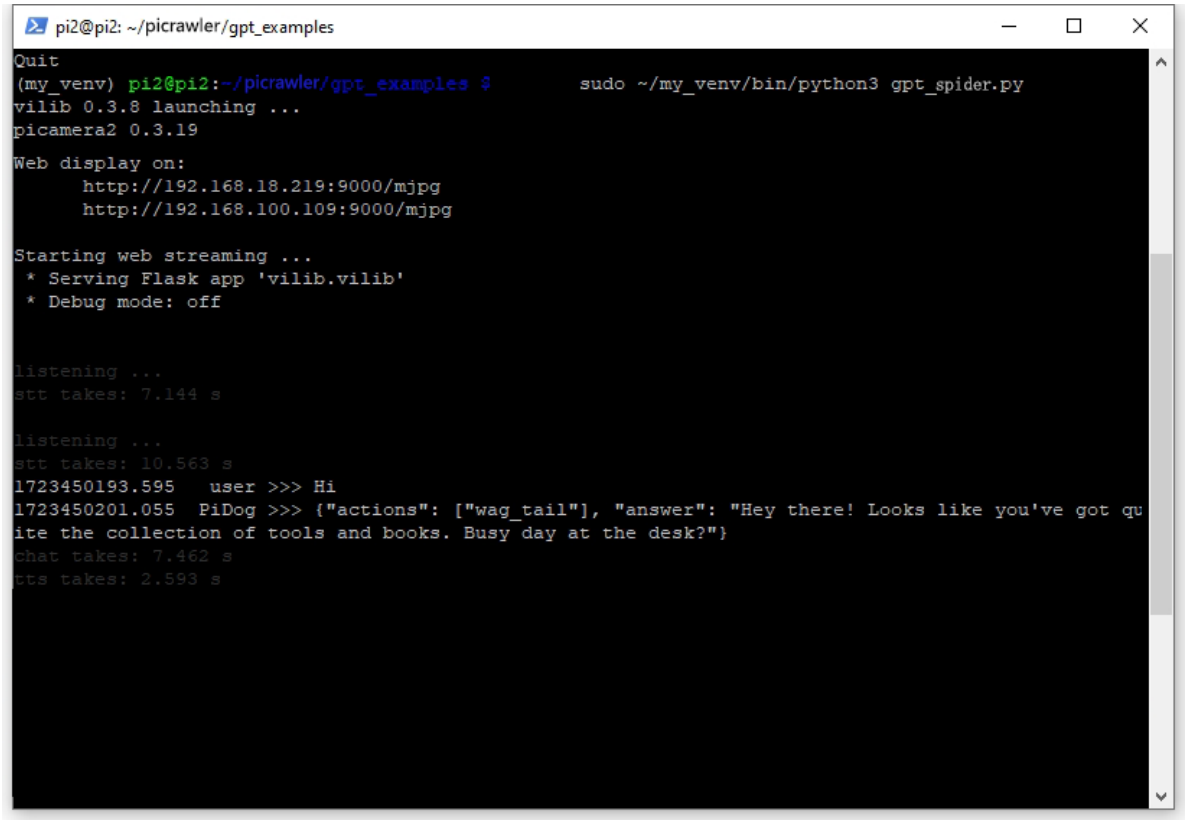
```
vilib 0.3.8 launching ...
picamera2 0.3.19

Web display on:
   http://rpi_ip:9000/mjpg

Starting web streaming ...
* Serving Flask app 'vilib.vilib'
* Debug mode: off

listening ...
```

6. You will also be provided with a link to view Pirawler's camera feed on your web browser: `http://rpi_ip:9000/mjpg`.



7. You can now speak to PiCrawler, and its responses may surprise you.

Note: PiCrawler needs to receive your input, convert it to text, send it to GPT for processing, receive the response, and then play it back via speech synthesis. This entire process takes some time, so please be patient.



8. If you are using the GPT-4O model, you can also ask questions based on what PiCrawler sees.

4.5 5. Modify parameters [optional]

In the `gpt_spider.py` file, locate the following lines. You can modify these parameters to configure the STT language, TTS volume gain, and voice role.

- **STT (Speech to Text)** refers to the process where the PiCrawler microphone captures speech and converts it into text to be sent to GPT. You can specify the language for better accuracy and latency in this conversion.
- **TTS (Text to Speech)** is the process of converting GPT's text responses into speech, which is played through the PiCrawler speaker. You can adjust the volume gain and select a voice role for the TTS output.

```
# openai assistant init
# ================================================================
openai_helper = OpenAiHelper(OPENAI_API_KEY, OPENAI_ASSISTANT_ID, 'picrawler')

# LANGUAGE = ['zh', 'en'] # config stt language code, https://en.wikipedia.org/wiki/List_of_
↪ISO_639_language_codes
LANGUAGE = []

VOLUME_DB = 3 # tts voloume gain, preferably less than 5db

# select tts voice role, counld be "alloy, echo, fable, onyx, nova, and shimmer"
# https://platform.openai.com/docs/guides/text-to-speech/supported-languages
TTS_VOICE = 'nova'
```

- `LANGUAGE` variable:
  - Improves Speech-to-Text (STT) accuracy and response time.
  - `LANGUAGE = []` means supporting all languages, but this may reduce STT accuracy and increase latency.
  - It's recommended to set the specific language(s) using language codes to improve performance.
- `VOLUME_DB` variable:
  - Controls the gain applied to Text-to-Speech (TTS) output.
  - Increasing the value will boost the volume, but it's best to keep the value below 5dB to prevent audio distortion.
- `TTS_VOICE` variable:
  - Select the voice role for the Text-to-Speech (TTS) output.
  - Available options: `alloy, echo, fable, onyx, nova, shimmer`.
  - You can experiment with different voices from to find one that suits your desired tone and audience. The available voices are currently optimized for English.

Note: Hello, welcome to the SunFounder Raspberry Pi & Arduino & ESP32 Enthusiasts Community on Facebook! Dive deeper into Raspberry Pi, Arduino, and ESP32 with fellow enthusiasts.

Why Join?

- **Expert Support**: Solve post-sale issues and technical challenges with help from our community and team.
- **Learn & Share**: Exchange tips and tutorials to enhance your skills.
- **Exclusive Previews**: Get early access to new product announcements and sneak peeks.
- **Special Discounts**: Enjoy exclusive discounts on our newest products.
- **Festive Promotions and Giveaways**: Take part in giveaways and holiday promotions.

Ready to explore and create with us? Click [] and join today!

CHAPTER

FIVE

APPENDIX

Note: Hello, welcome to the SunFounder Raspberry Pi & Arduino & ESP32 Enthusiasts Community on Facebook! Dive deeper into Raspberry Pi, Arduino, and ESP32 with fellow enthusiasts.

Why Join?

- **Expert Support**: Solve post-sale issues and technical challenges with help from our community and team.
- **Learn & Share**: Exchange tips and tutorials to enhance your skills.
- **Exclusive Previews**: Get early access to new product announcements and sneak peeks.
- **Special Discounts**: Enjoy exclusive discounts on our newest products.
- **Festive Promotions and Giveaways**: Take part in giveaways and holiday promotions.

Ready to explore and create with us? Click [] and join today!

5.1 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.

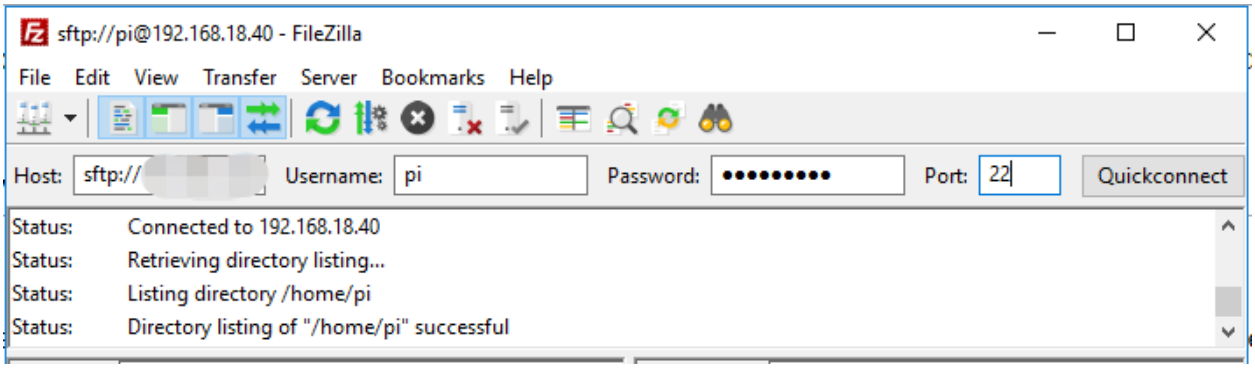
Filezilla is an open source software that not only supports FTP, but also FTP over TLS (FTPS) and SFTP. We can use Filezilla to upload local files (such as pictures and audio, etc.) to the Raspberry Pi, or download files from the Raspberry Pi to the local.

Step 1: Download Filezilla.

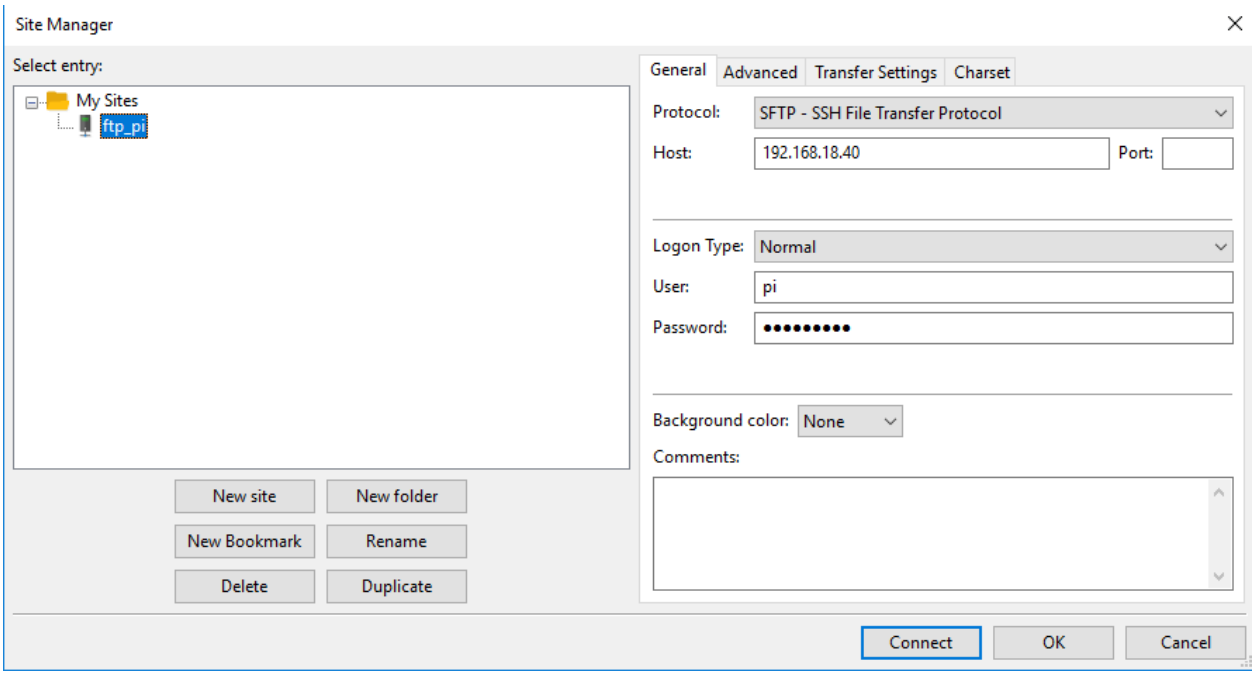
Download the client from Filezilla's official website, Filezilla has a very good tutorial, please refer to: Documentation - Filezilla.

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.

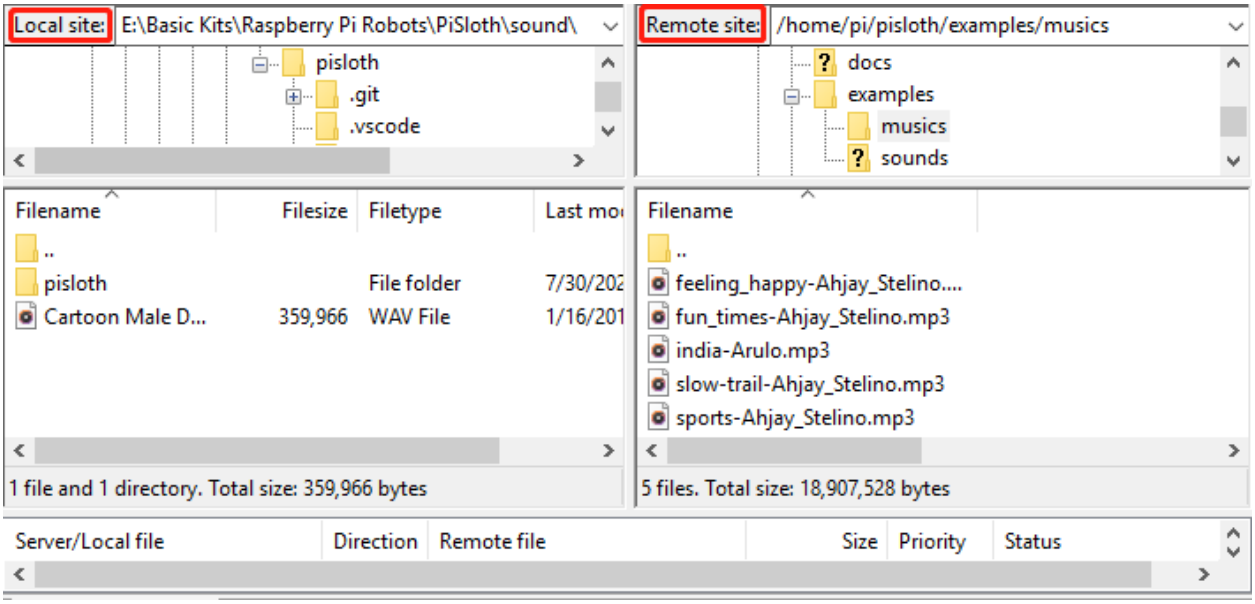


Note: Quick Connect is a good way to test your login information. If you want to create a permanent entry, you can select **File-> Copy Current Connection to Site Manager** after a successful Quick Connect, enter the name and click **OK**. Next time you will be able to connect by selecting the previously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



Note: Hello, welcome to the SunFounder Raspberry Pi & Arduino & ESP32 Enthusiasts Community on Facebook! Dive deeper into Raspberry Pi, Arduino, and ESP32 with fellow enthusiasts.

Why Join?

- **Expert Support**: Solve post-sale issues and technical challenges with help from our community and team.
- **Learn & Share**: Exchange tips and tutorials to enhance your skills.
- **Exclusive Previews**: Get early access to new product announcements and sneak peeks.
- **Special Discounts**: Enjoy exclusive discounts on our newest products.
- **Festive Promotions and Giveaways**: Take part in giveaways and holiday promotions.

Ready to explore and create with us? Click [] and join today!

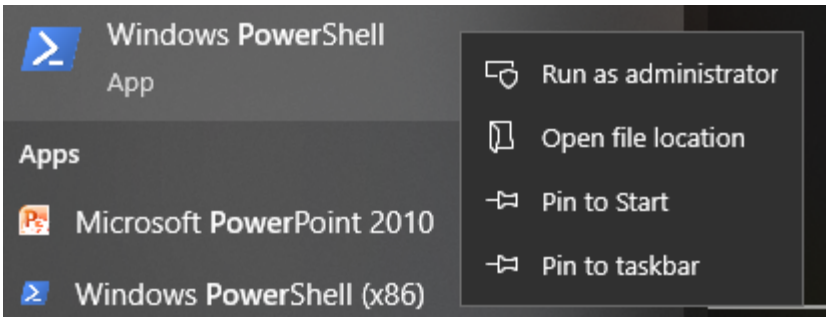
5.2 Install OpenSSH via Powershell

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.

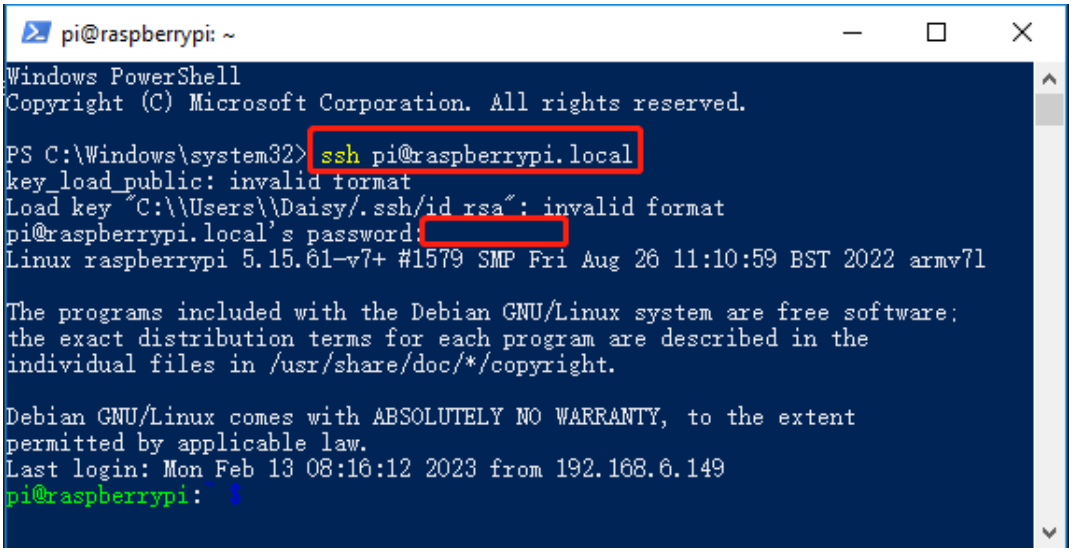
```
Get-WindowsCapability -Online | Where-Object Name -like 'OpenSSH*'
```

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.



Note: Hello, welcome to the SunFounder Raspberry Pi & Arduino & ESP32 Enthusiasts Community on Facebook! Dive deeper into Raspberry Pi, Arduino, and ESP32 with fellow enthusiasts.

Why Join?

- **Expert Support**: Solve post-sale issues and technical challenges with help from our community and team.
- **Learn & Share**: Exchange tips and tutorials to enhance your skills.
- **Exclusive Previews**: Get early access to new product announcements and sneak peeks.
- **Special Discounts**: Enjoy exclusive discounts on our newest products.
- **Festive Promotions and Giveaways**: Take part in giveaways and holiday promotions.

Ready to explore and create with us? Click [] and join today!

CHAPTER

SIX

HARDWARE

When you are writing code, you may need to know how each module works or the role of each pin, then please see this chapter.

In this chapter you will find a description of each module's function, technical parameters and working principle.

Note: Hello, welcome to the SunFounder Raspberry Pi & Arduino & ESP32 Enthusiasts Community on Facebook! Dive deeper into Raspberry Pi, Arduino, and ESP32 with fellow enthusiasts.

Why Join?

- **Expert Support**: Solve post-sale issues and technical challenges with help from our community and team.
- **Learn & Share**: Exchange tips and tutorials to enhance your skills.
- **Exclusive Previews**: Get early access to new product announcements and sneak peeks.
- **Special Discounts**: Enjoy exclusive discounts on our newest products.
- **Festive Promotions and Giveaways**: Take part in giveaways and holiday promotions.

Ready to explore and create with us? Click [] and join today!

6.1 Robot HAT

is a multifunctional expansion board that allows Raspberry Pi to be quickly turned into a robot. An MCU is on board to extend the PWM output and ADC input for the Raspberry Pi, as well as a motor driver chip, I2S audio module and mono speaker. As well as the GPIOs that lead out of the Raspberry Pi itself.

It also comes with a Speaker, which can be used to play background music, sound effects and implement TTS functions to make your project more interesting.

Accepts 7-12V power input with 2 battery indicators, 1 charge indicator and 1 power indicator. The board also has a user available LED and a button for you to quickly test some effects.

For detailed instructions, please refer to: .

Note: Hello, welcome to the SunFounder Raspberry Pi & Arduino & ESP32 Enthusiasts Community on Facebook! Dive deeper into Raspberry Pi, Arduino, and ESP32 with fellow enthusiasts.

Why Join?

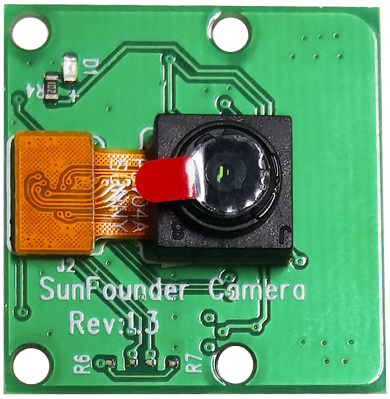
- **Expert Support**: Solve post-sale issues and technical challenges with help from our community and team.
- **Learn & Share**: Exchange tips and tutorials to enhance your skills.

- **Exclusive Previews**: Get early access to new product announcements and sneak peeks.
- **Special Discounts**: Enjoy exclusive discounts on our newest products.
- **Festive Promotions and Giveaways**: Take part in giveaways and holiday promotions.

Ready to explore and create with us? Click [] and join today!

6.2 Camera Module

Description



This is a 5MP Raspberry Pi camera module with OV5647 sensor. It's plug and play, connect the included ribbon cable to the CSI (Camera Serial Interface) port on your Raspberry Pi and you're ready to go.

The board is small, about 25mm x 23mm x 9mm, and weighs 3g, making it ideal for mobile or other size and weight-critical applications. The camera module has a native resolution of 5 megapixels and has an on-board fixed focus lens that captures still images at 2592 x 1944 pixels, and also supports 1080p30, 720p60 and 640x480p90 video.

Note: The module is only capable of capturing pictures and videos, not sound.

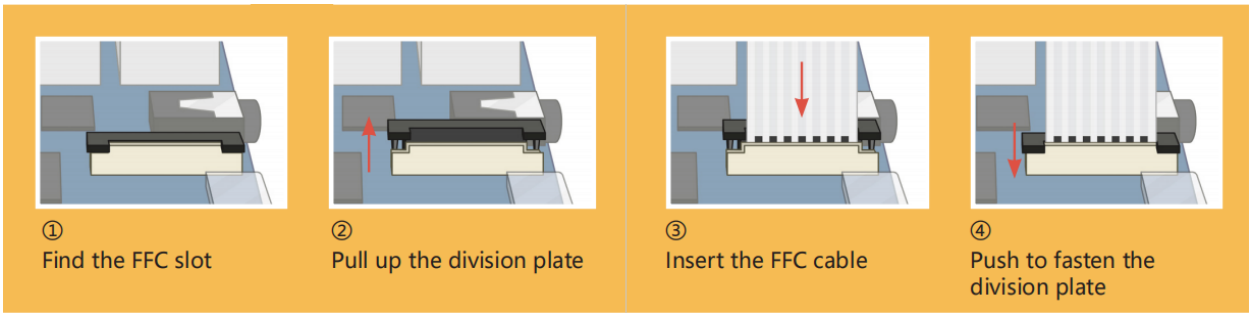
Specification

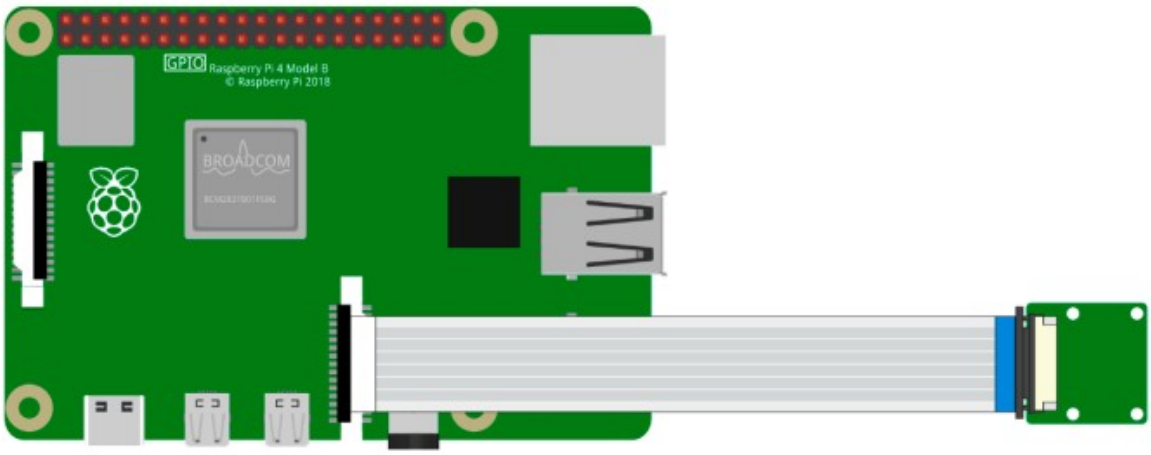
- **Static Images Resolution**: 2592×1944
- **Supported Video Resolution**: 1080p/30 fps, 720p/ 60fps and 640 x480p 60/90 video recording
- **Aperture (F)**: 1.8
- **Visual Angle**: 65 degree
- **Dimension**: 24mmx23.5mmx8mm
- **Weight**: 3g
- **Interface**: CSI connector
- **Supported OS**: Raspberry Pi OS(latest version recommended)

Assemble the Camera Module

On the camera module or Raspberry Pi, you will find a flat plastic connector. Carefully pull out the black fixing switch until the fixing switch is partially pulled out. Insert the FFC cable into the plastic connector in the direction shown and push the fixing switch back into place.

If the FFC wire is installed correctly, it will be straight and will not pull out when you gently pull on it. If not, reinstall it again.





Warning: Do not install the camera with the power on, it may damage your camera.

Note: Hello, welcome to the SunFounder Raspberry Pi & Arduino & ESP32 Enthusiasts Community on Facebook! Dive deeper into Raspberry Pi, Arduino, and ESP32 with fellow enthusiasts.

Why Join?

- **Expert Support**: Solve post-sale issues and technical challenges with help from our community and team.
- **Learn & Share**: Exchange tips and tutorials to enhance your skills.
- **Exclusive Previews**: Get early access to new product announcements and sneak peeks.
- **Special Discounts**: Enjoy exclusive discounts on our newest products.
- **Festive Promotions and Giveaways**: Take part in giveaways and holiday promotions.

Ready to explore and create with us? Click [] and join today!

6.3 Ultrasonic Module

- **TRIG**: Trigger Pulse Input
- **ECHO**: Echo Pulse Output
- **GND**: Ground
- **VCC**: 5V Supply

This is the HC-SR04 ultrasonic distance sensor, providing non-contact measurement from 2 cm to 400 cm with a range accuracy of up to 3 mm. Included on the module is an ultrasonic transmitter, a receiver and a control circuit.

You only need to connect 4 pins: VCC (power), Trig (trigger), Echo (receive) and GND (ground) to make it easy to use for your measurement projects.

Features

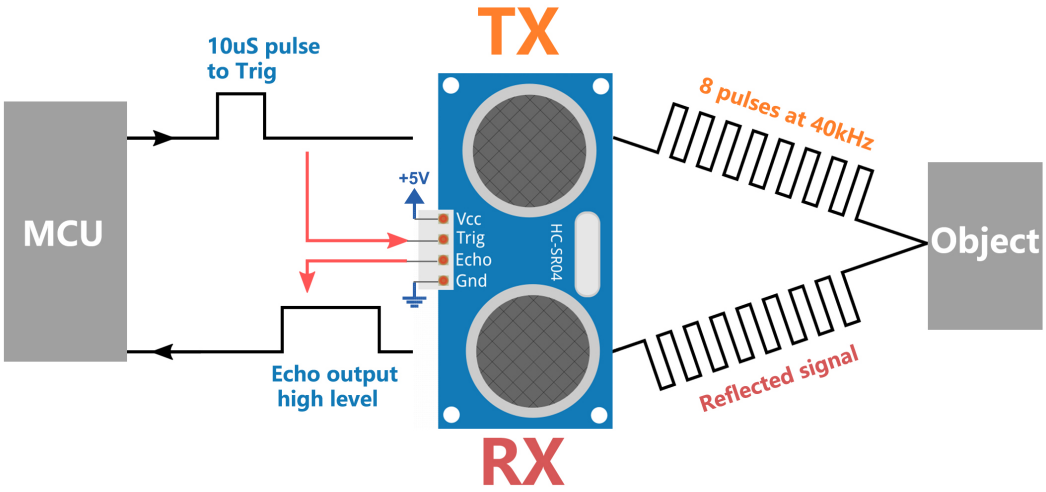
- Working Voltage: DC5V
- Working Current: 16mA
- Working Frequency: 40Hz
- Max Range: 500cm
- Min Range: 2cm
- Trigger Input Signal: 10uS TTL pulse
- Echo Output Signal: Input TTL lever signal and the range in proportion
- Connector: XH2.54-4P
- Dimension: 46x20.5x15 mm

Principle

The basic principles are as follows:

- Using IO trigger for at least 10us high level signal.
- The module sends an 8 cycle burst of ultrasound at 40 kHz and detects whether a pulse signal is received.

- Echo will output a high level if a signal is returned; the duration of the high level is the time from emission to return.
- Distance = (high level time x velocity of sound (340M/S)) / 2



Formula:

- us / 58 = centimeters distance
- us / 148 = inch distance
- distance = high level time x velocity (340M/S) / 2

Application Notes

- This module should not be connected under power up, if necessary, let the module's GND be connected first. Otherwise, it will affect the work of the module.
- The area of the object to be measured should be at least 0.5 square meters and as flat as possible. Otherwise, it will affect results.

Note: Hello, welcome to the SunFounder Raspberry Pi & Arduino & ESP32 Enthusiasts Community on Facebook! Dive deeper into Raspberry Pi, Arduino, and ESP32 with fellow enthusiasts.

Why Join?

- **Expert Support**: Solve post-sale issues and technical challenges with help from our community and team.
- **Learn & Share**: Exchange tips and tutorials to enhance your skills.
- **Exclusive Previews**: Get early access to new product announcements and sneak peeks.
- **Special Discounts**: Enjoy exclusive discounts on our newest products.
- **Festive Promotions and Giveaways**: Take part in giveaways and holiday promotions.

Ready to explore and create with us? Click [] and join today!

6.4 3-pin Battery

- **VCC**: Battery positive terminal, here there are two sets of VCC and GND is to increase the current and reduce the resistance.
- **Middle**: To balance the voltage between the two cells and thus protect the battery.
- **GND**: Negative battery terminal.

This is a custom battery pack made by SunFounder consisting of two 18650 batteries with a capacity of 2000mAh. The connector is XH2.54 3P, which can be charged directly after being inserted into the Robot HAT.

Features

- Composition: Li-ion
- Battery Capacity: 2000mAh, 14.8Wh
- Battery Weight: 90.8g
- Number of Cells: 2

- Connector: XH2.54 3P
- Over-discharge protection: 6.0V

6.4.1 5-pin Battery

- **VCC**: Battery positive terminal, here there are two sets of VCC and GND is to increase the current and reduce the resistance.
- **Middle**: To balance the voltage between the two cells and thus protect the battery.
- **GND**: Negative battery terminal.

This is a custom battery pack made by SunFounder consisting of two 18650 batteries with a capacity of 2000mAh. The connector is PH2.0-5P, which can be charged directly after being inserted into the shield.

Features

- **Composition**: Li-ion
- **Battery Capacity**: 2000mAh, 14.8Wh

- **Battery Weight**: 90.8g
- **Number of Cells**: 2
- **Connector**: PH2.0-5P
- **Over-discharge protection**: 6.0V

Note: Hello, welcome to the SunFounder Raspberry Pi & Arduino & ESP32 Enthusiasts Community on Facebook! Dive deeper into Raspberry Pi, Arduino, and ESP32 with fellow enthusiasts.

Why Join?

- **Expert Support**: Solve post-sale issues and technical challenges with help from our community and team.
- **Learn & Share**: Exchange tips and tutorials to enhance your skills.
- **Exclusive Previews**: Get early access to new product announcements and sneak peeks.
- **Special Discounts**: Enjoy exclusive discounts on our newest products.
- **Festive Promotions and Giveaways**: Take part in giveaways and holiday promotions.

Ready to explore and create with us? Click [] and join today!

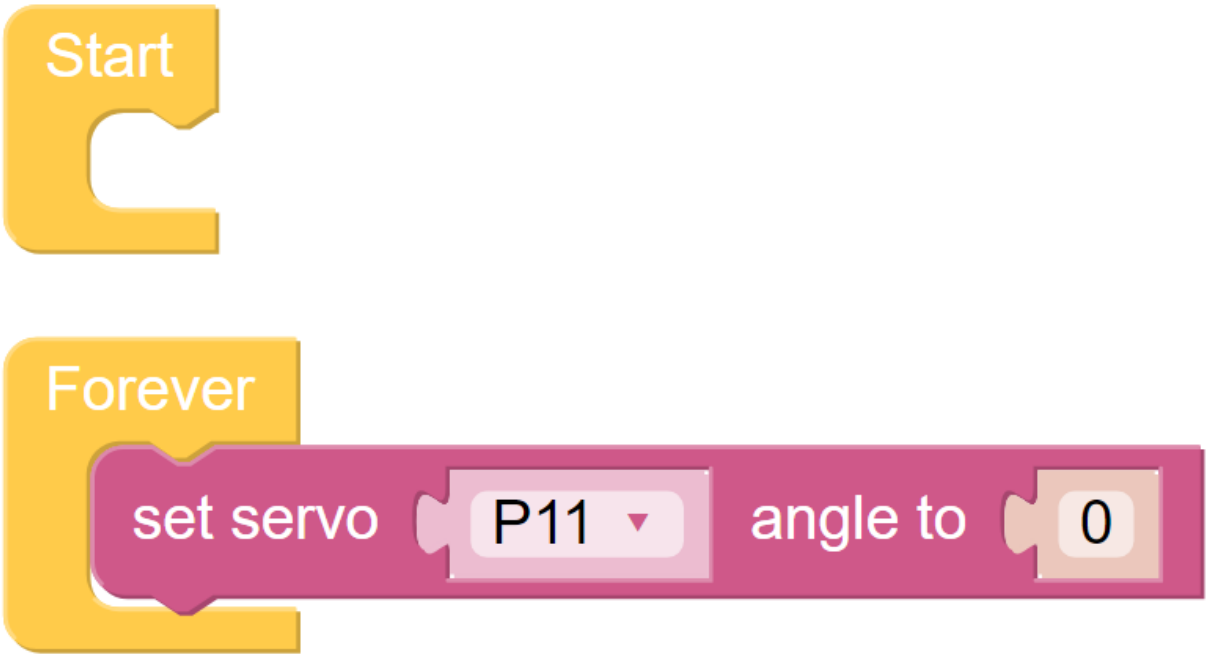
CHAPTER

SEVEN

FAQ

7.1 Q1: After installing Ezblock OS, the servo can't turn to 0°?

1) Check if the servo cable is properly connected and if the Robot HAT power is on.
2) Press Reset button.
3) If you have already run the program in Ezblock Studio, the custom program for P11 is no longer available. You can refer to the picture below to manually write a program in Ezblock Studio to set the servo angle to 0.



7.2 Q2: When using VNC, I am prompted that the desktop cannot be displayed at the moment?

In Terminal, type `sudo raspi-config` to change the resolution.

7.3 Q3: Why does the servo sometimes return to the middle position for no reason?

When the servo is blocked by a structure or other object and cannot reach its intended position, the servo will enter the power-off protection mode in order to prevent the servo from being burned out by too much current.

After a period of power failure, if no PWM signal is given to the servo, the servo will automatically return to its original position.

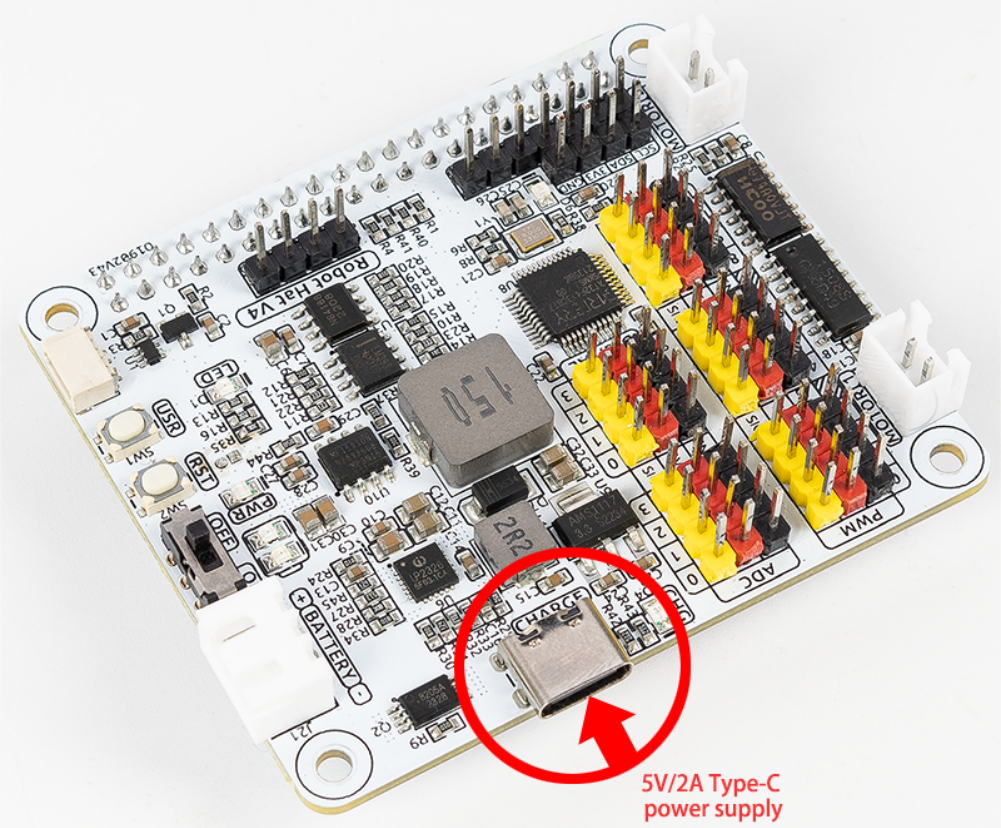
7.4 Q4: About the Robot HAT Detailed Tutorial?

You can find a comprehensive tutorial about the Robot HAT here, including information on its hardware and API.

-

7.5 Q5: About the Battery Charger?

To charge the battery, simply connect a 5V/2A Type-C power supply to the Robot Hat's power port. There's no need to turn on the Robot Hat's power switch during charging. You can also use the device while charging the battery.



During charging, the input power is boosted by the charging chip to charge the battery and simultaneously supply the DC-DC converter for external use, with a charging power of approximately 10W. If external power consumption remains high for an extended period, the battery may supplement the power supply, similar to using a phone while charging. However, be mindful of the battery's capacity to avoid completely depleting it during simultaneous charging and usage.

CHAPTER

EIGHT

COPYRIGHT NOTICE