Quadruped Bionic Spider Robot Tutorial

Perface

Our Company

ACEBOTT STEM Education Tech Co.,Ltd

Founded in China's Silicon Valley in 2013, ACEBOTT is a STEM education solution leader We have a team of 150 individuals, including members from research and development, sales, and logistics. Our goal is to provide high-quality STEM education products and services to our customers. We are working together with STEM education experts and our business partners to produce successful STE products together Our self-own factory also provides CEM services for our clients including Logo customization on product packaging and PCB.

Our Tutorial

This course and quadruped robot learning kit is designed for 8+ children and teenagers to learn more about ESP8266 controller board and quadruped robot knowledge, and electronic hardware. If you like to learn quadruped robot knowledge, this kit could provide you the knowledge and steps to build your own quadruped robot.

Through this kit, you can:

1. Learn how to effectively use the ESP8266 controller board, including downloading code, understanding its features, and coding in the Arduino IDE.
2. Gain a solid foundation in the basics of the C languages, as ESP8266 utilizes the simplified C/C++ programming languages to control circuits and sensors.
3. Explore the working principle of the servo module and understand the collaborative work of multiple servos in the quadruped robot project.
4. Enhance your maker skills by building your own quadruped robot using the ACEBOTT kit, following step-by-step tutorials.
5. In the quadruped robot project, realized the basic functions such as front and back, left and right movement control, left and right rotation control, combined action control, App control , etc.

6. Develop a comprehensive understanding of quadruped robot concept to prepare for more advanced learning in the future.

In summary, the ACEBOTT quadruped robot is an ESP8266-based learning kit specifically designed for beginners. With this kit, users can gain a comprehensive understanding of how controller boards and servos function within the context of a quadruped robot. By following the provided tutorials, individuals of all ages can acquire valuable knowledge about quadruped robot and successfully build their own quadruped robot projects.

Customer service

ACEBOTT is a dynamic and fast-growing STEM education technology company that strives to offer excellent products and quality services that meet your expectations. We value your feedback and encourage you to drop us a line at support@acebott.com with any comments or suggestions you may have.

Our experienced engineers are dedicated to promptly addressing any problems or questions you may have about our products. We guarantee a response within 24 hours during business days.

Follow Us

Scan the QR codes to Follow Us for troubleshooting & the latest news.
We have a very large community that is very helpful for troubleshooting and we also have a support team at the ready to answer any questions.

ACEBOTT FB Group QR Code

YouTube QR Code

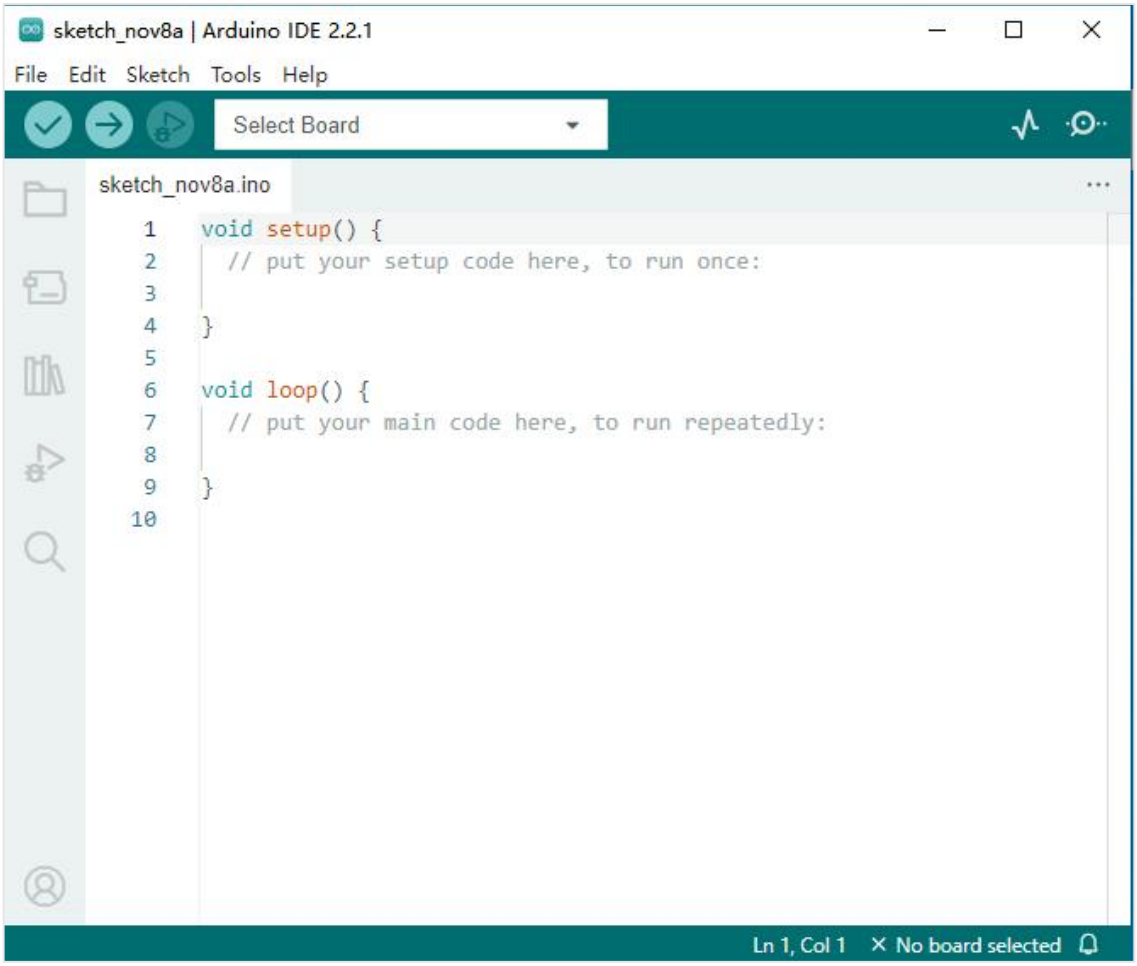
Contents

Lesson 1 Software installation and hardware understanding

Ⅰ.Software installation

In the project of quadruped bionic spider, we will mainly use Arduino IDE as programming software, which is an open source programming software and can be compatible with a variety of controller boards, such as Arduino, ESP32, ESP8266, STM32 and so on.

With Arduino IDE, you just write the program code in the IDE and upload it to the controller board and the program will tell the controller board what needs to be done.



1. Install Arduino IDE

First, open the official website where the Arduino IDE is downloaded:

https://www.arduino.cc/en/Main/Software

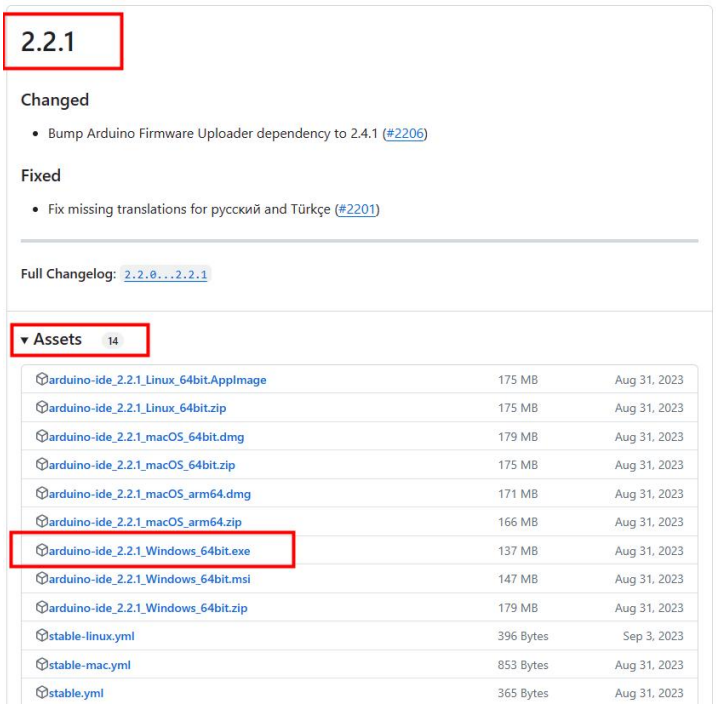
According to the user's computer system, choose the corresponding installation path.

(1) Installation method under Windows system

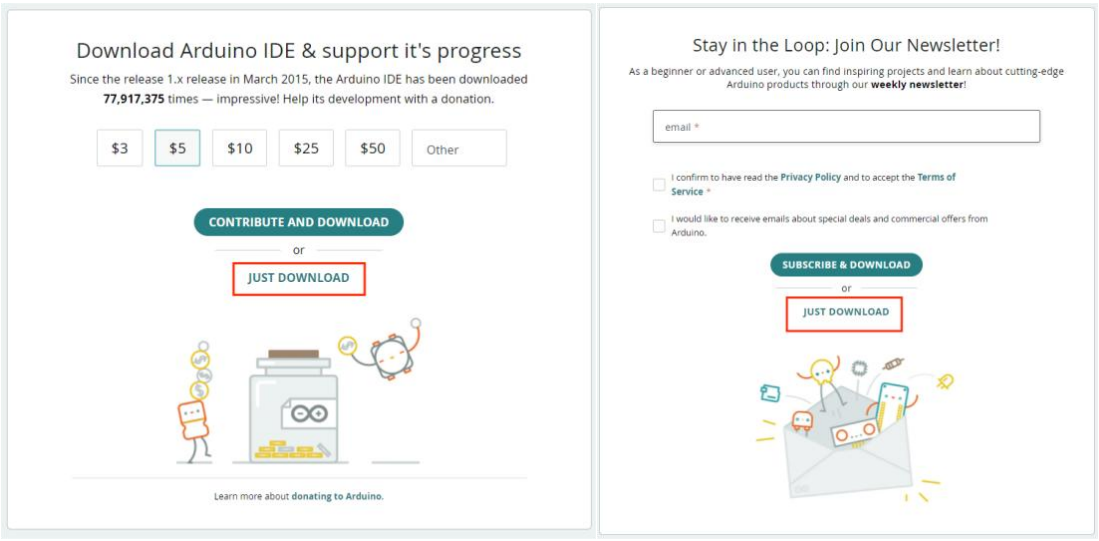
①Click the mouse at the position shown in the figure.**(Attention: It is recommended to install Arduino IDE version 2.2.1 because there may be compatibility issues with newer versions)**

Download link for Arduino IDE version 2.2.1:

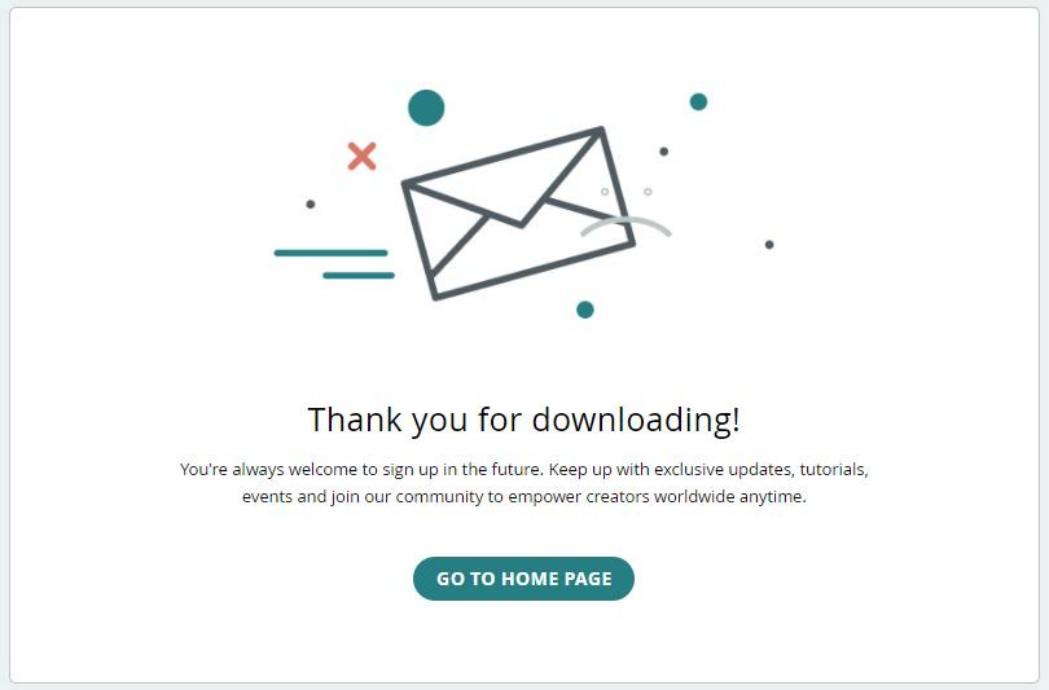
https://github.com/arduino/arduino-ide/releases



②Select JUSTDOWNLOAD.

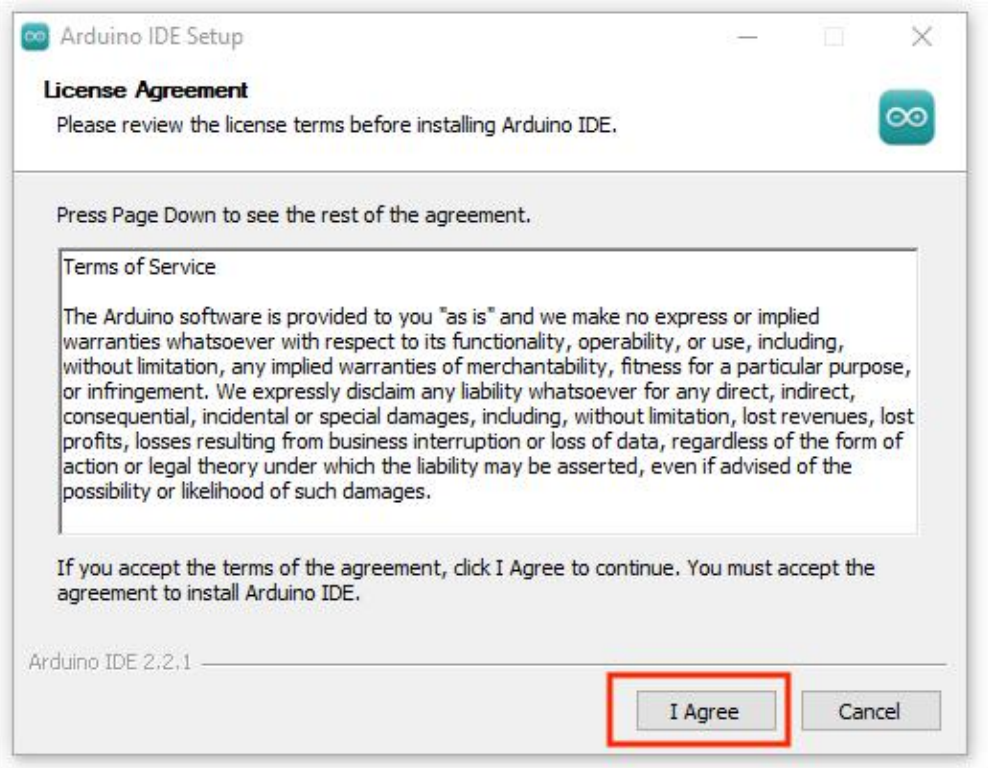


③When the following screen appears, the Arduino IDE is downloading.

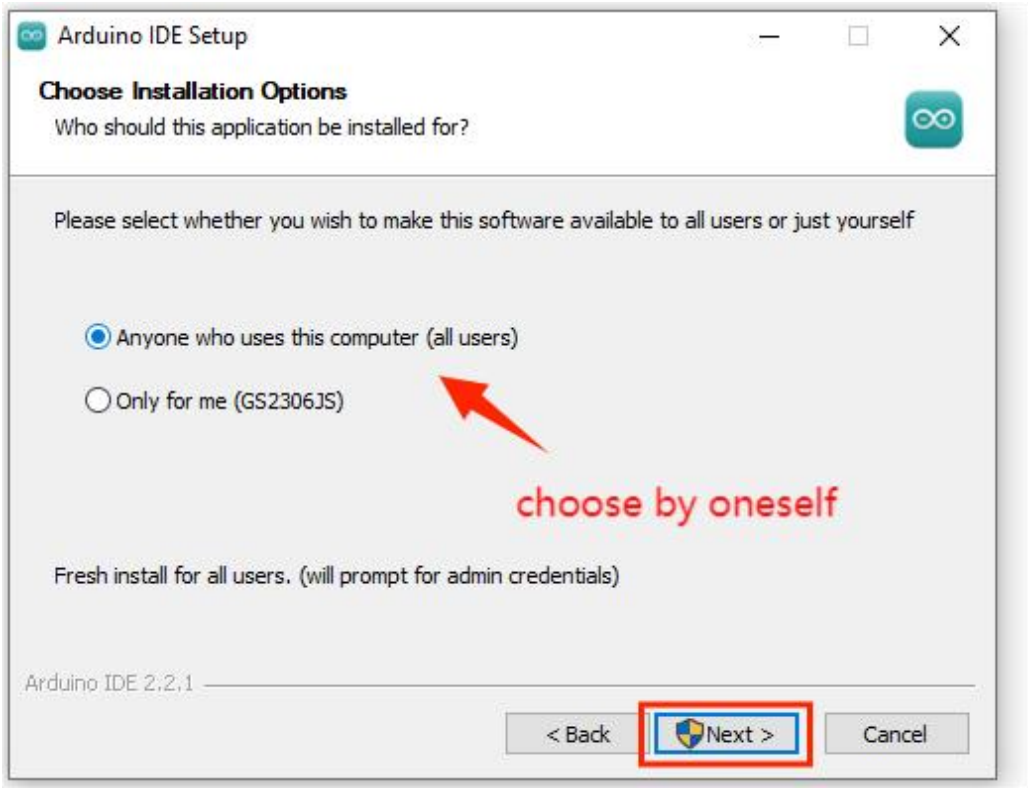


④When the download is complete, the icon file will appear. Click Install software.

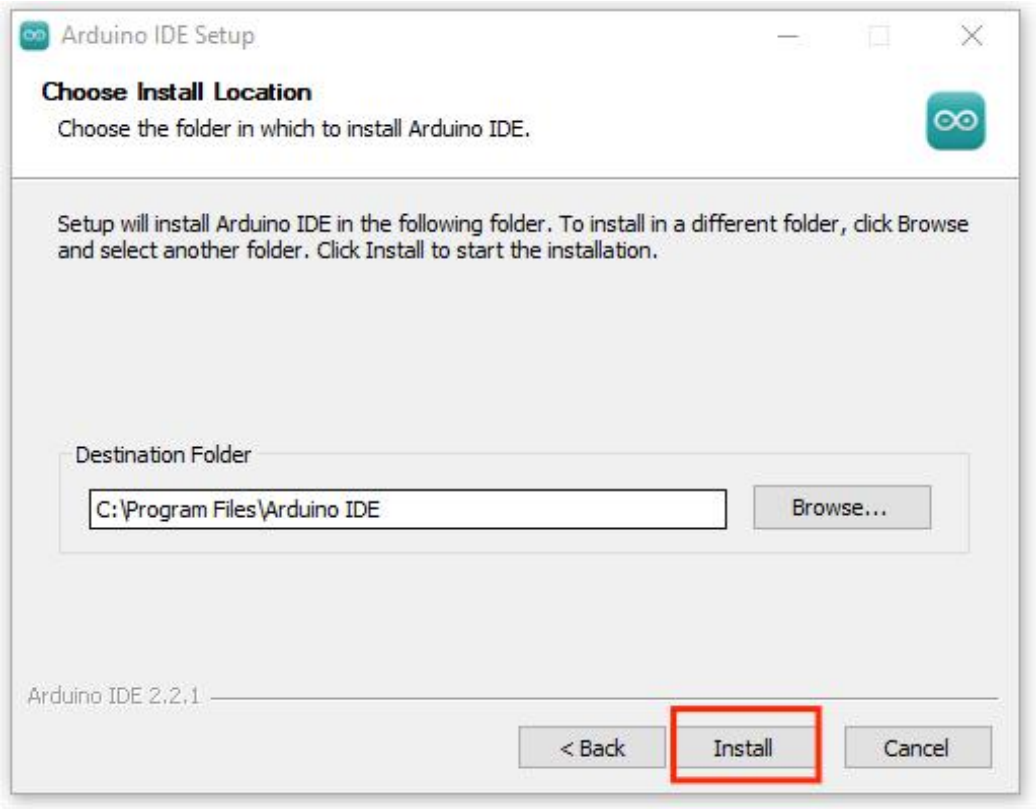
⑤After installation, the following screen appears and select "I Agree".



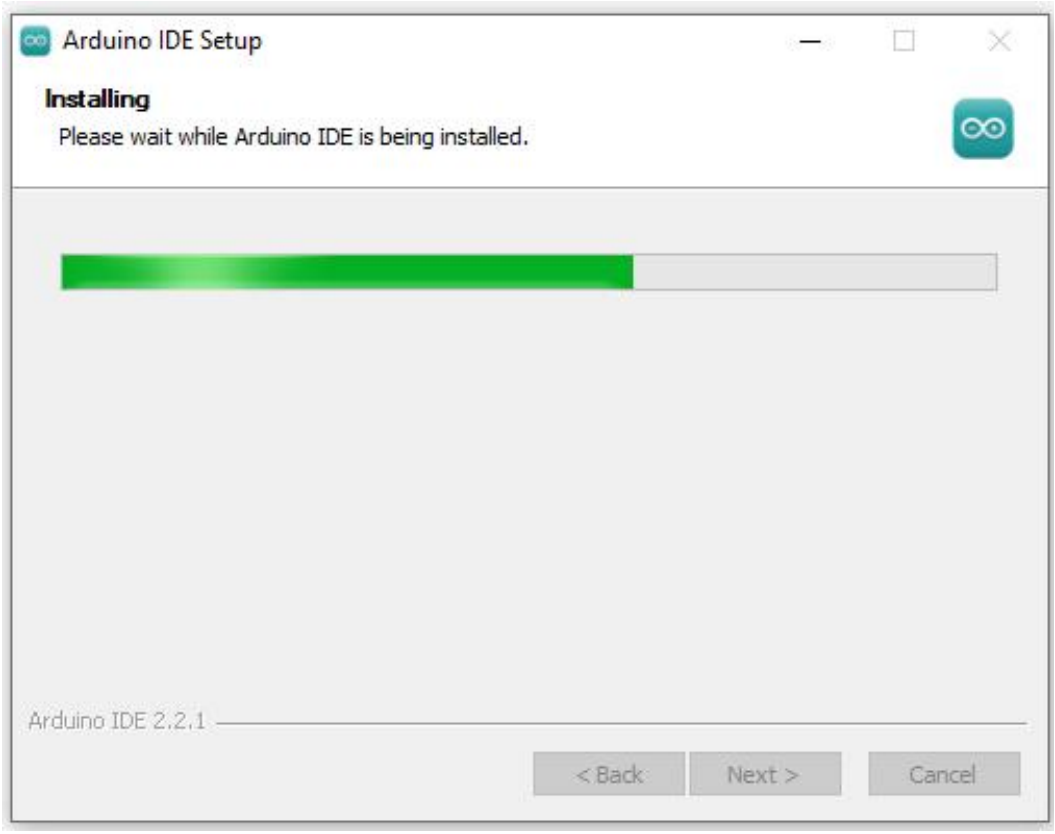
⑥After selecting "I Agree", the following screen will appear and select "Next".



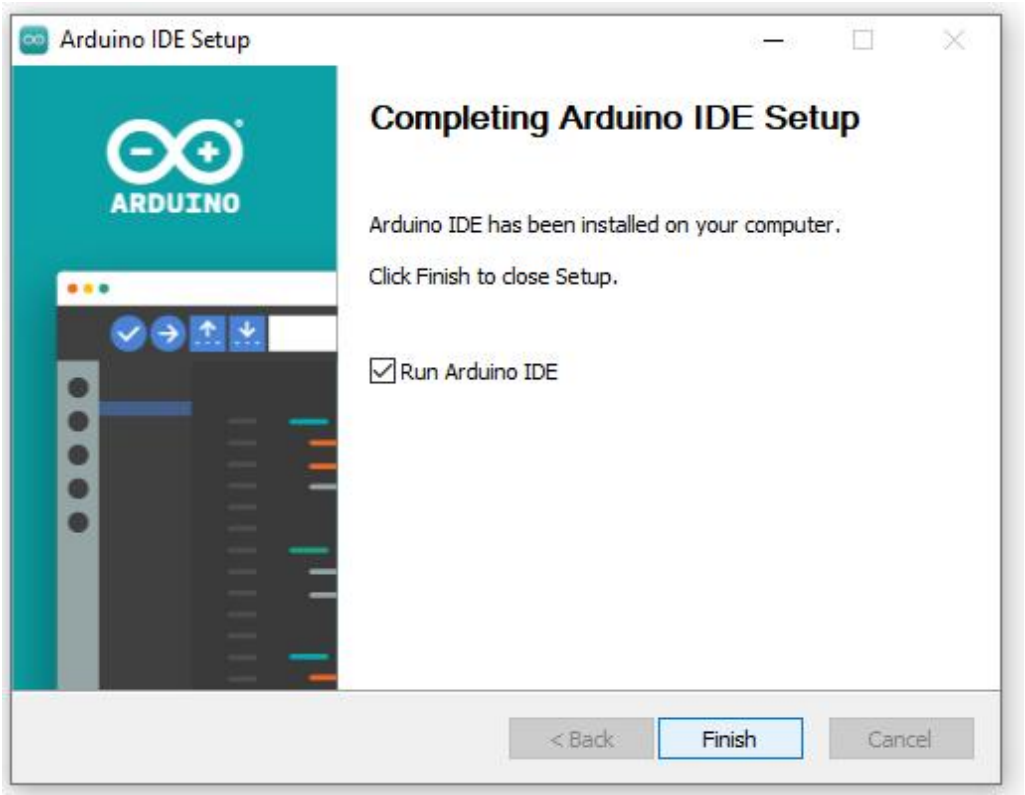
⑦Select "Next" and the following screen will appear. Select "Install".



⑧Arduino IDE software installed.

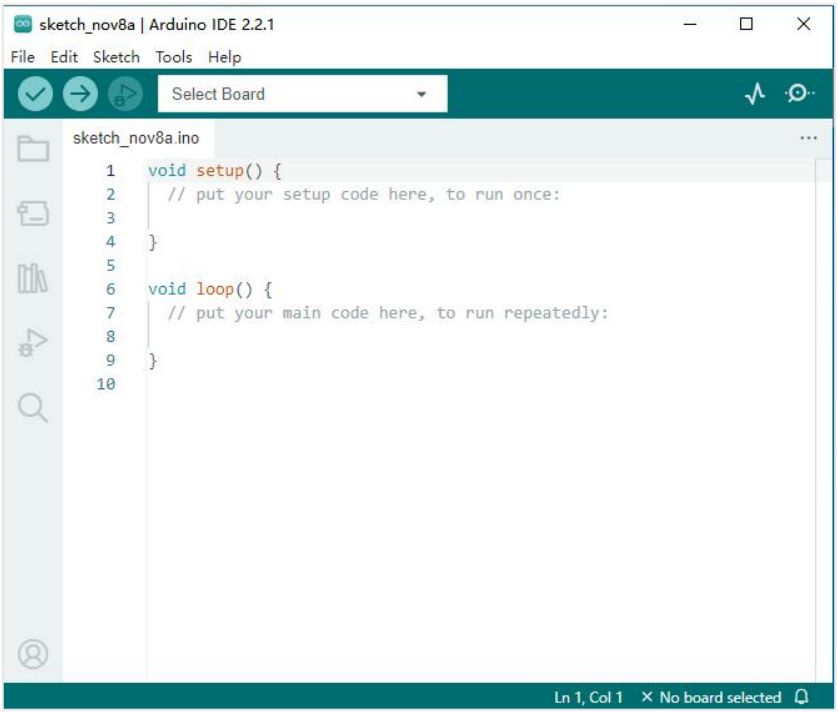


⑨When the installation is complete, click Finish.



⑩After installation, a shortcut icon of Arduino IDE will appear on your desktop.

⑪After opening, the following software interface will appear.

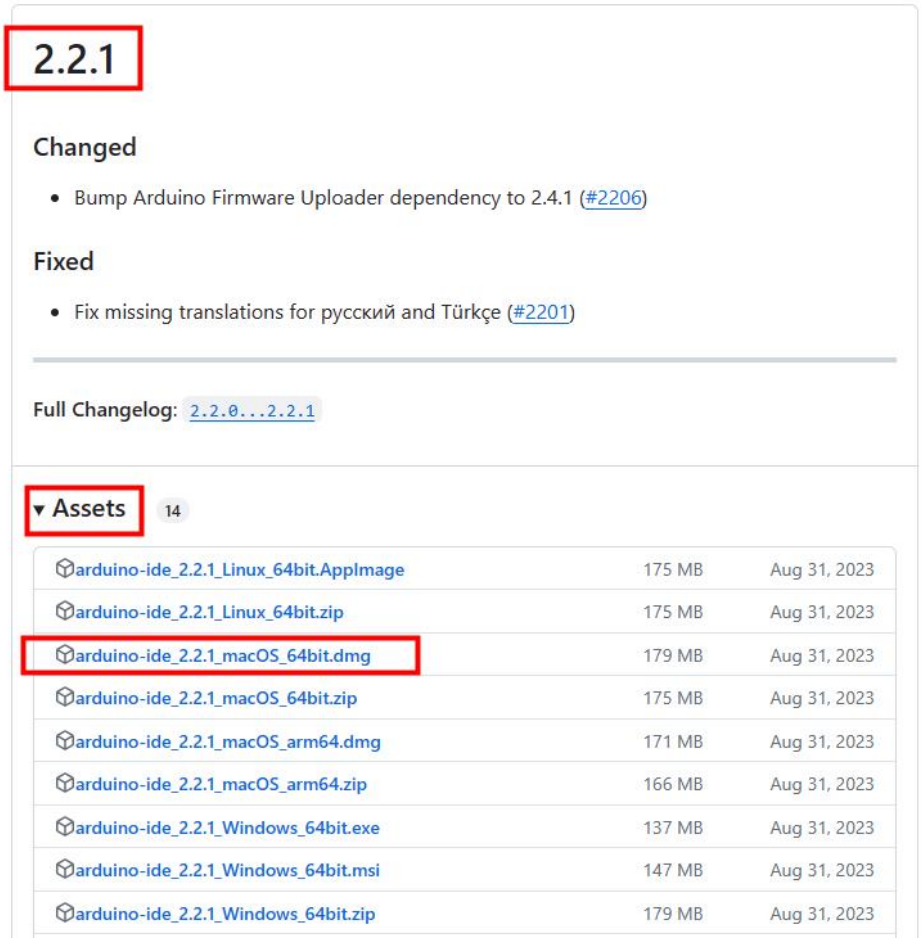


(2) Installation on Mac OS

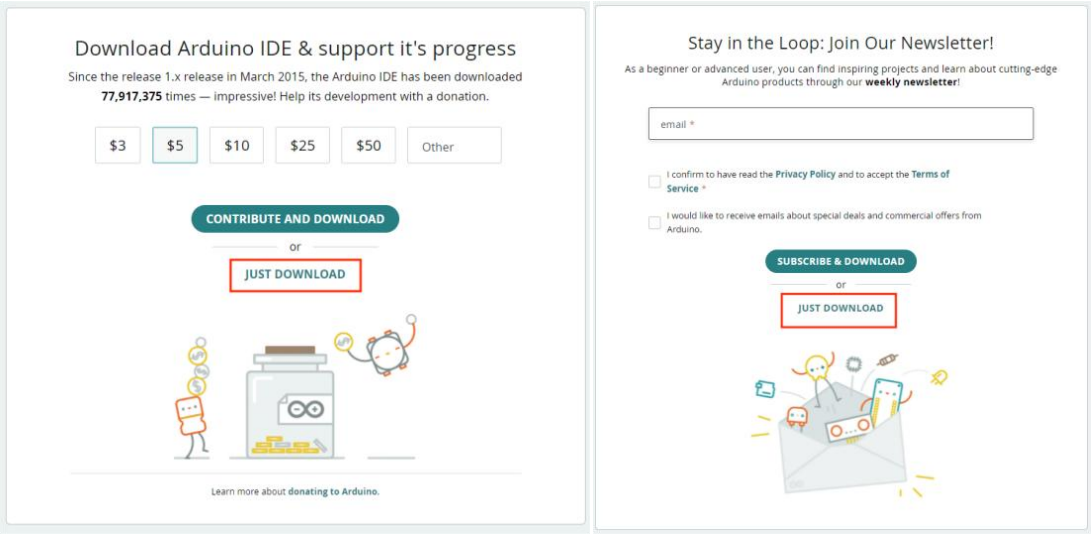
①Click the mouse at the position shown.**(Attention: It is recommended to install Arduino IDE version 2.2.1 because there may be compatibility issues with newer versions)**

Download link for Arduino IDE version 2.2.1：

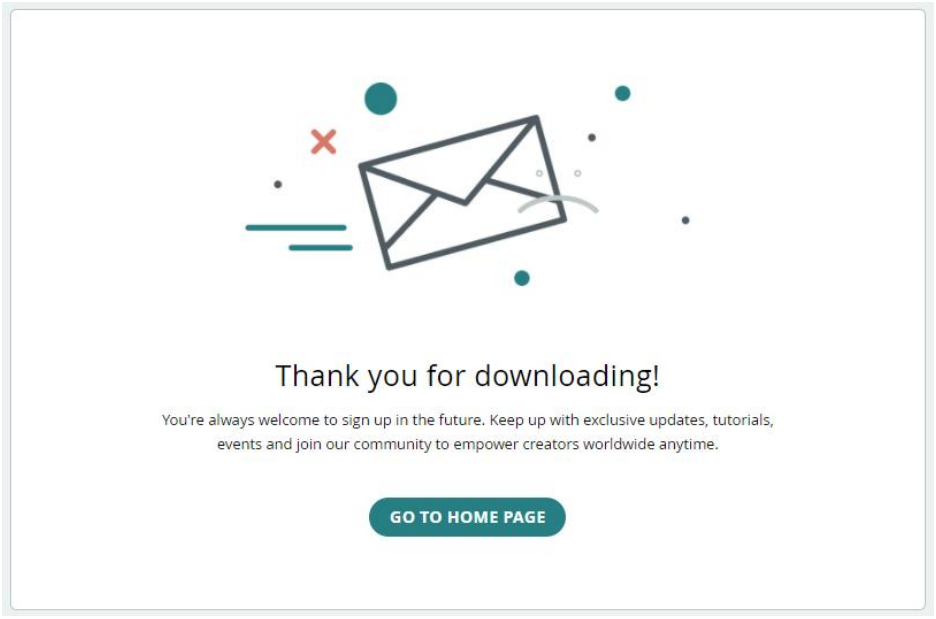
https://github.com/arduino/arduino-ide/releases



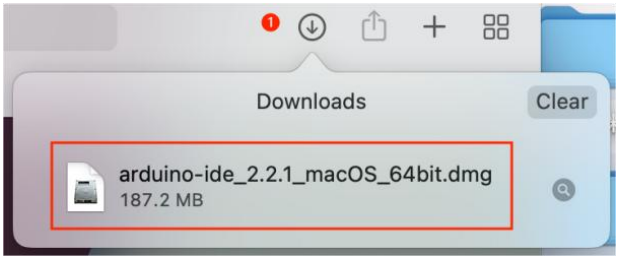
②Select JUSTDOWNLOAD.



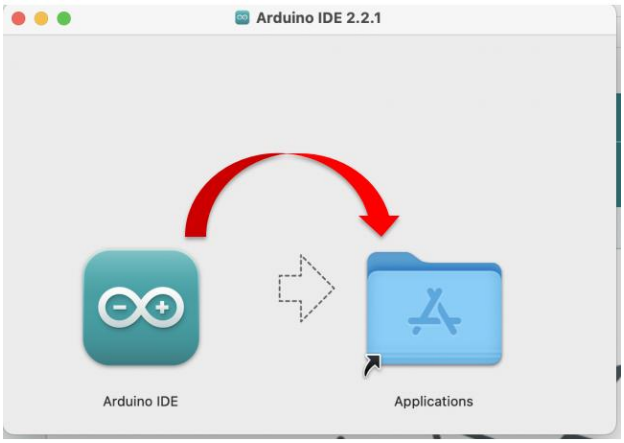
③When the following screen appears, the Arduino IDE is downloading.



④When the download is complete click the download icon in your browser and find the Arduino IDE installer.

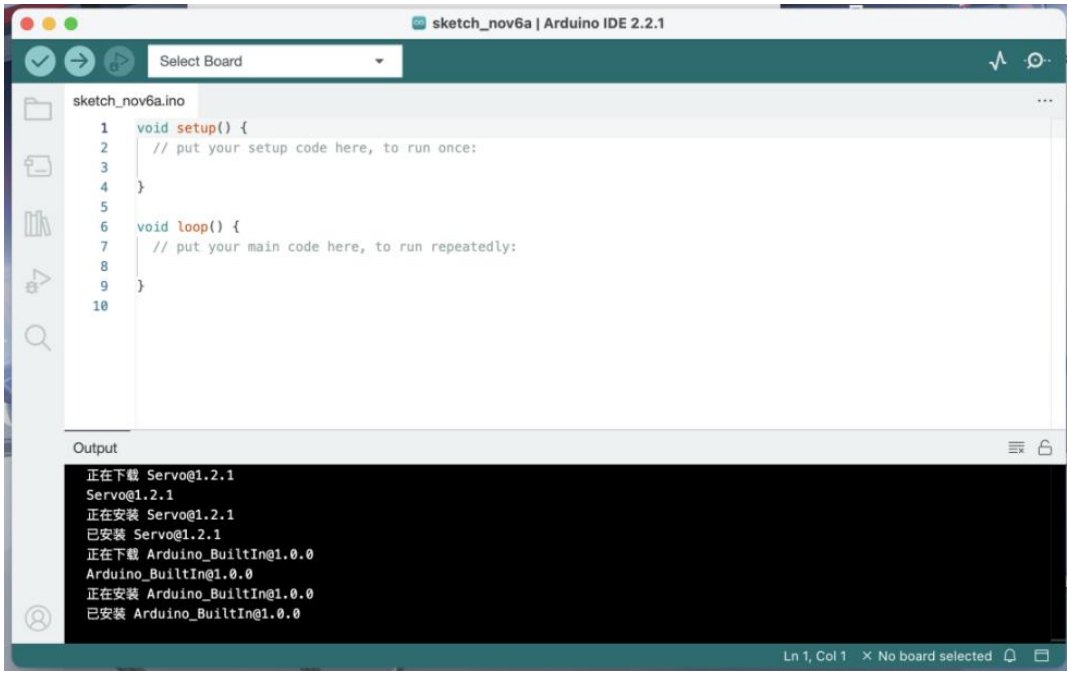


⑤Click on the install package to appear the installation screen, just select the Arduino IDE icon, move to the Applications to install the program.



⑥In the workbench, find the Arduino IDE and open it.

⑦After opening, you can see the following software interface.



(3) Install the serial driver (skip it if installed)

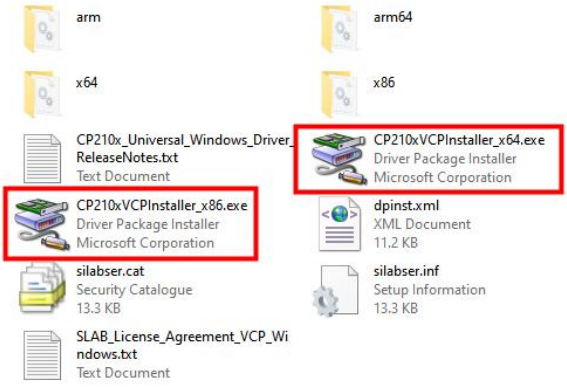
Serial port is a computer communication interface, usually used for the computer and other devices (such as modems, sensors, printers, microcontrollers, etc.) for data transmission, USB serial port is our most commonly used communication interface.

Generally, after installing the Arduino IDE software for the first time, it is necessary to install the serial port driver to transmit the edited program to the controller board.

The USB to serial port chip of ESP8266 control board is CP210x, so you need to install the driver of the chip. The installation of the driver can be downloaded on the Internet or found in the resource package named "3. Driver install "folder, then follow the steps below to install it.

①Windows system driver installation mode

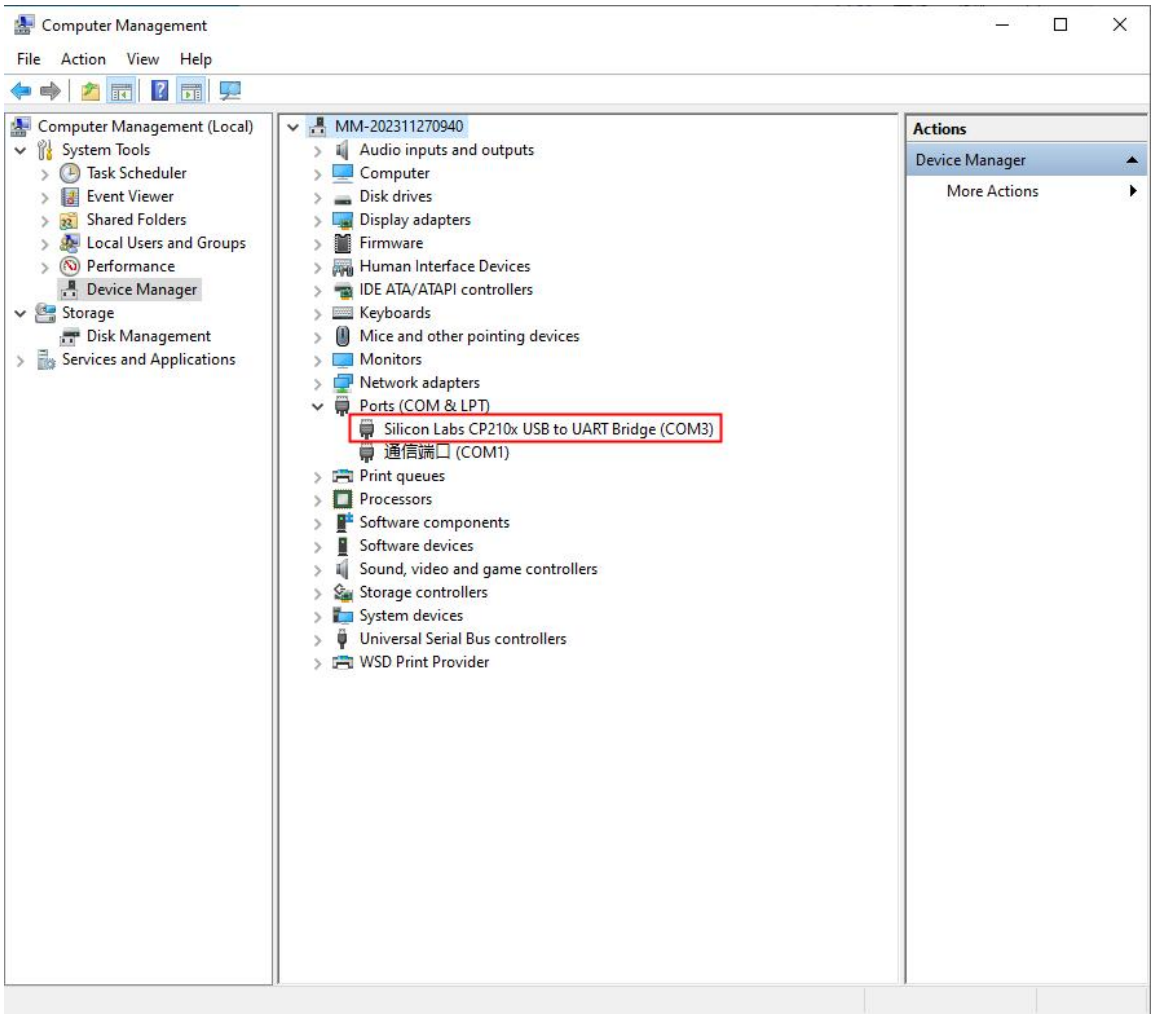
Open the "Windows system" folder, and then open the "CP210x_Universal_Windows_Driver" folder, you will see two exe files, please install according to the configuration of your Windows computer, if your computer system type is 64 bit, Please double-click "CP210xVCPInstaller_x64.exe" with the mouse; If your computer system type is 32 bit, please double-click "CP210xVCPInstaller_x86.exe" with the mouse.



After double-clicking the corresponding exe file, an installation window will pop up. Click "Next".

Until the last window appears, the installation is complete.Finally, click "Finish".

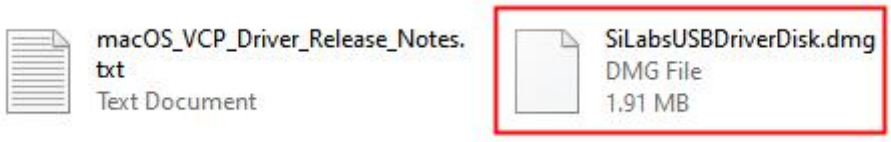
To confirm whether the installation is successful, insert one end of the USB cable into the ESP8266 controller board and the other end into the USB interface on the computer. Right click "My Computer "->" Properties "-> click" Device Manager ", after connecting to the control panel, the following picture shows that the installation is successful.



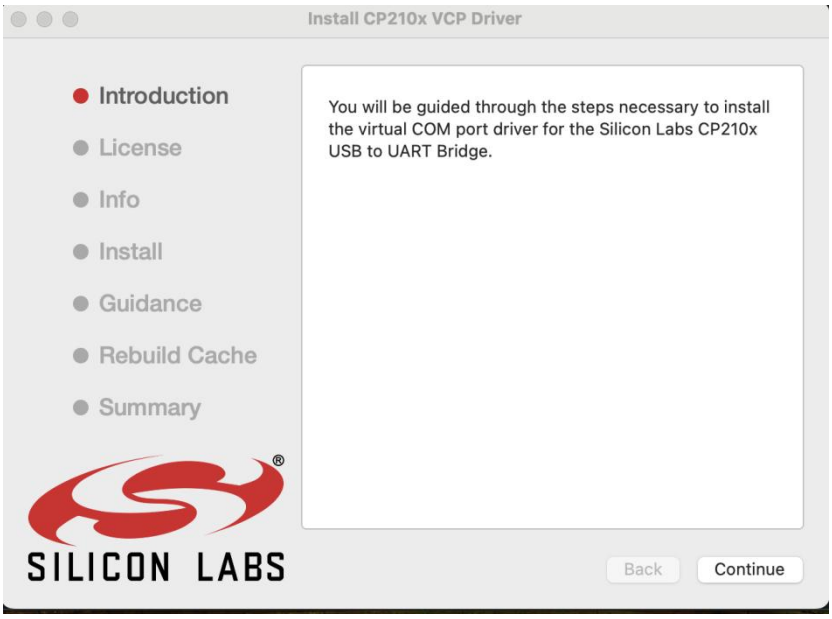
②Mac system driver installation mode

A. Install the serial port driver

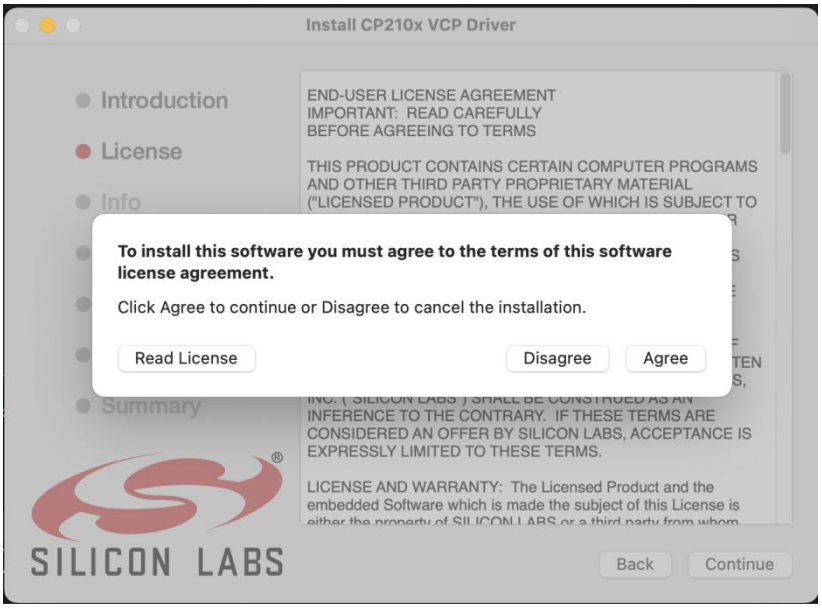
Open "Mac System" folder, then open "Mac_OSX_VCP_Driver" folder, and then double-click "SiLabsUSBDriverDisk.dmg" installation file, according to the following installation window prompts to install.



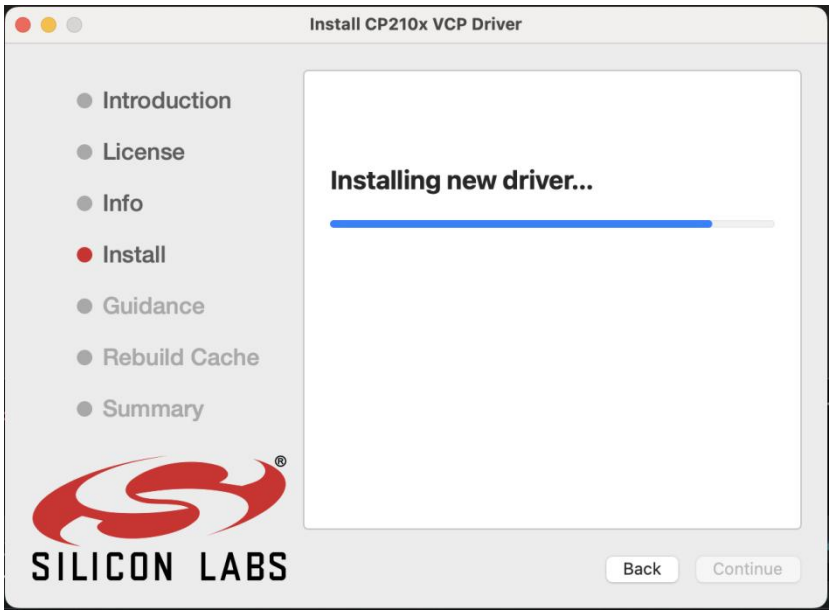
Click "Continue"



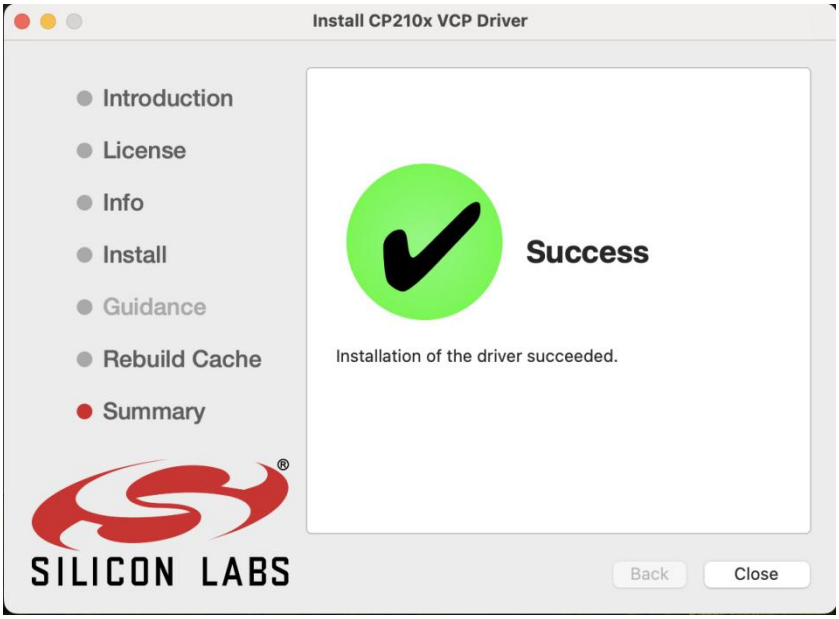
Click "Agree"



Your Mac may be prompted for a password for installation, so enter your login password.



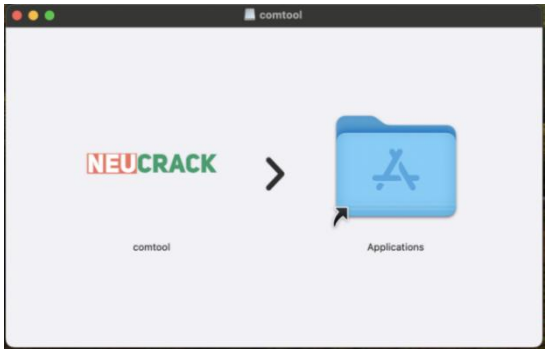
This prompt indicates that the driver has been installed.



B. Install serial debugging tools

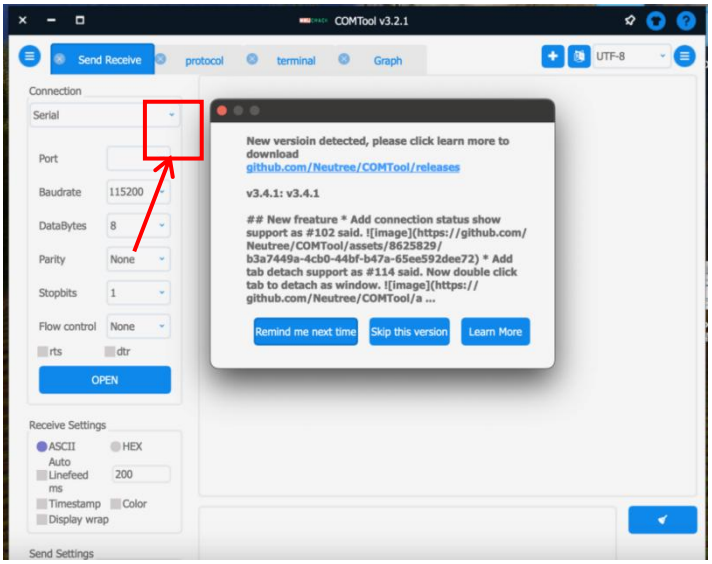
After the driver is installed, you need to use the serial debugging tool to check whether the serial port appears. You need to open the "Mac System" folder, and then double-click the "comtool_macos_v3.2.1.dmg" serial debugging tool installation file to install.

Drag the application into Applications.

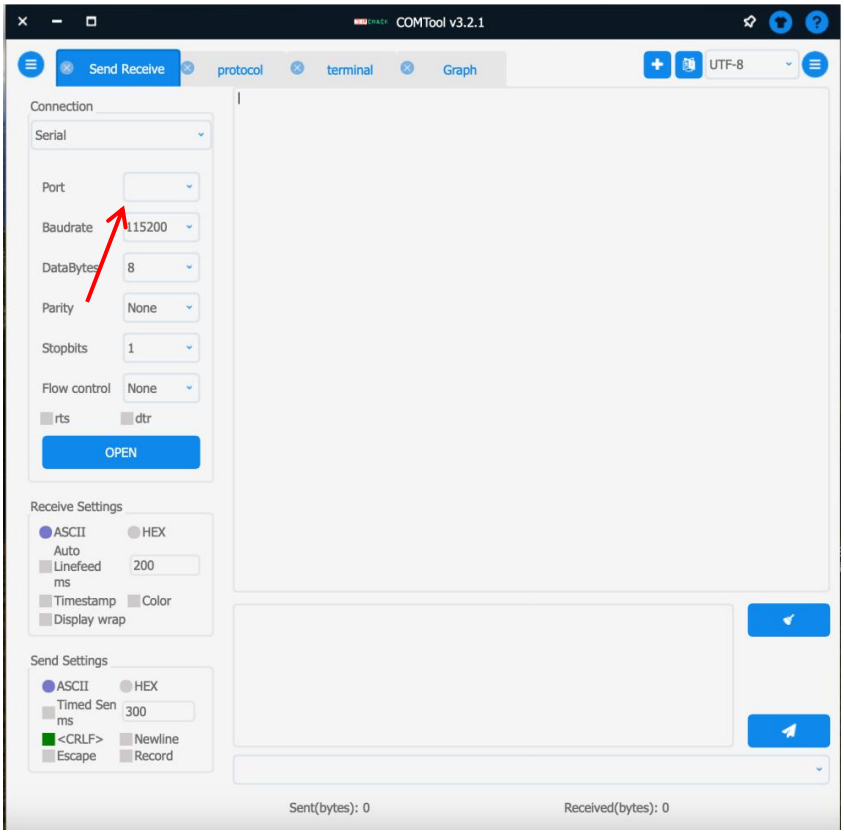


Double-click "comtool" in the launcher.

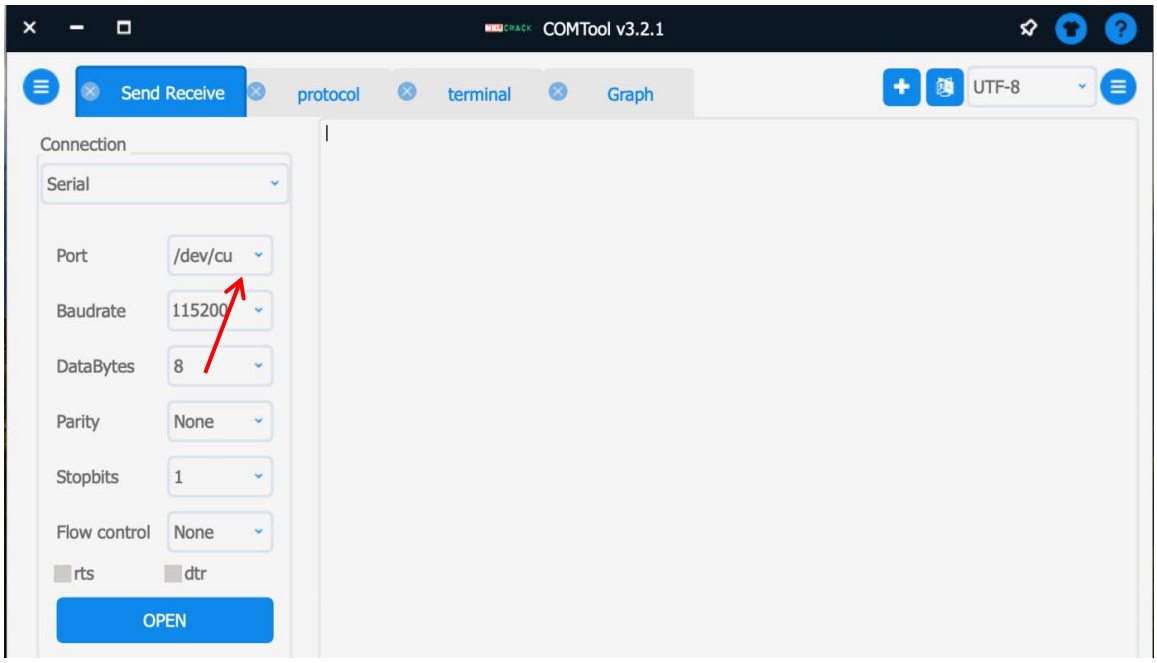
After the interface of the serial port application pops up, close the prompt window first.



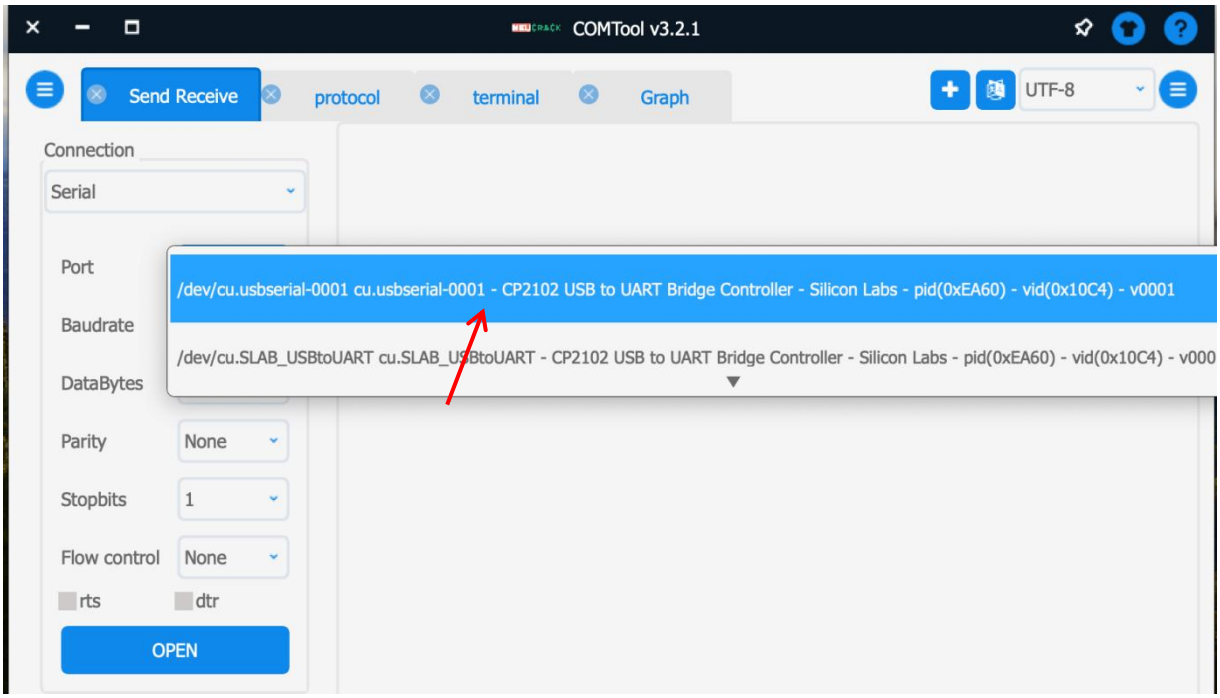
There is no serial port information at the beginning of the software interface. Please insert one end of the USB cable into the ESP8266 control board and the other end into the USB interface on the computer.



Then when you see the serial Port information in the Port interface bar, it means that the driver was installed successfully.



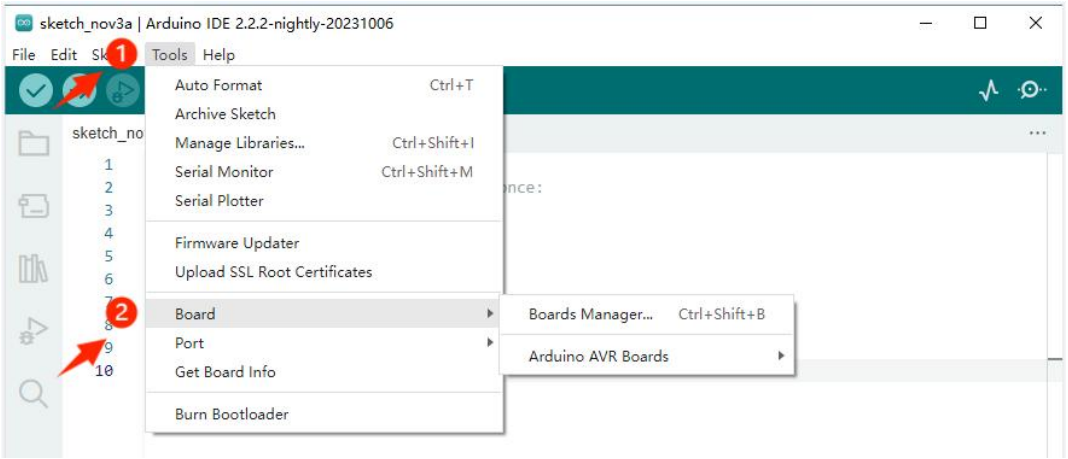
When clicking the drop-down triangle symbol, the specific serial port information of CP2102 will be displayed.



2. Install ESP8266 library

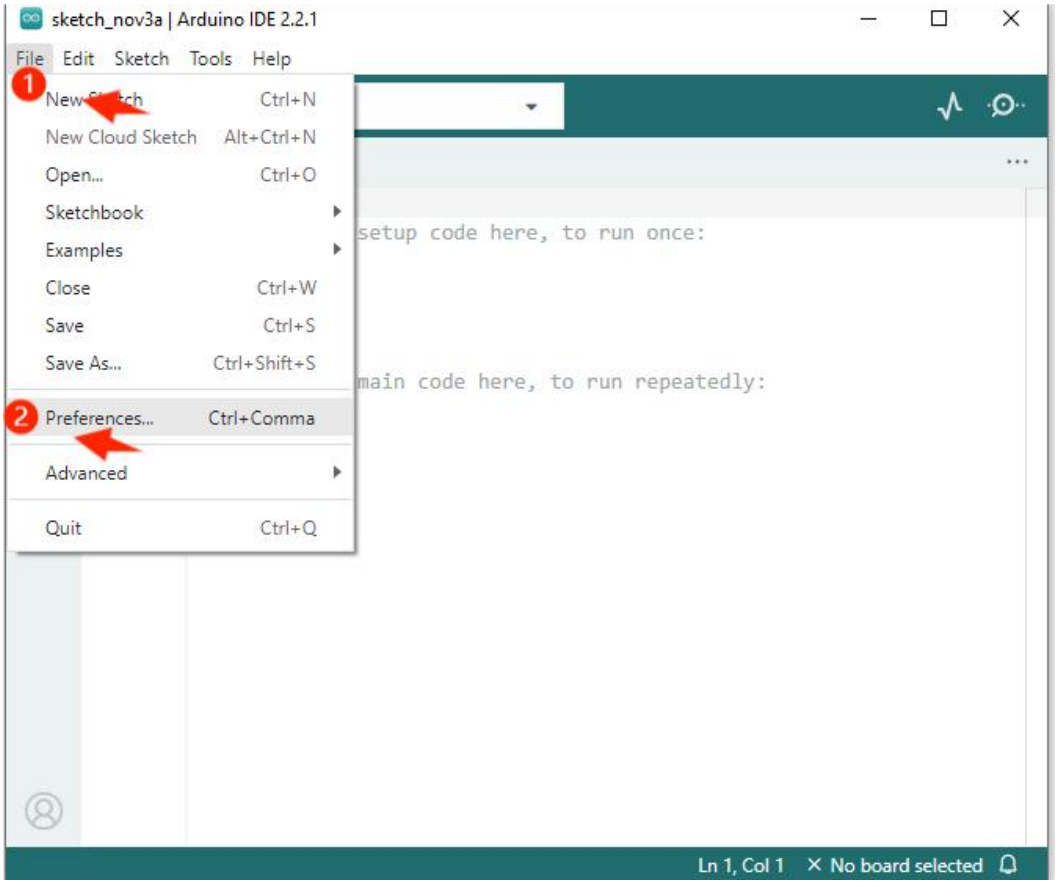
Because the controller board used by the robot is ESP8266, it is also necessary to add the library of ESP8266 to program the ESP8266 controller board on the Arduino IDE.

When you open the Arduino IDE, select Tools>Board, you will find that the Arduino IDE only has Arduino AVR Boards and no ESP8266.

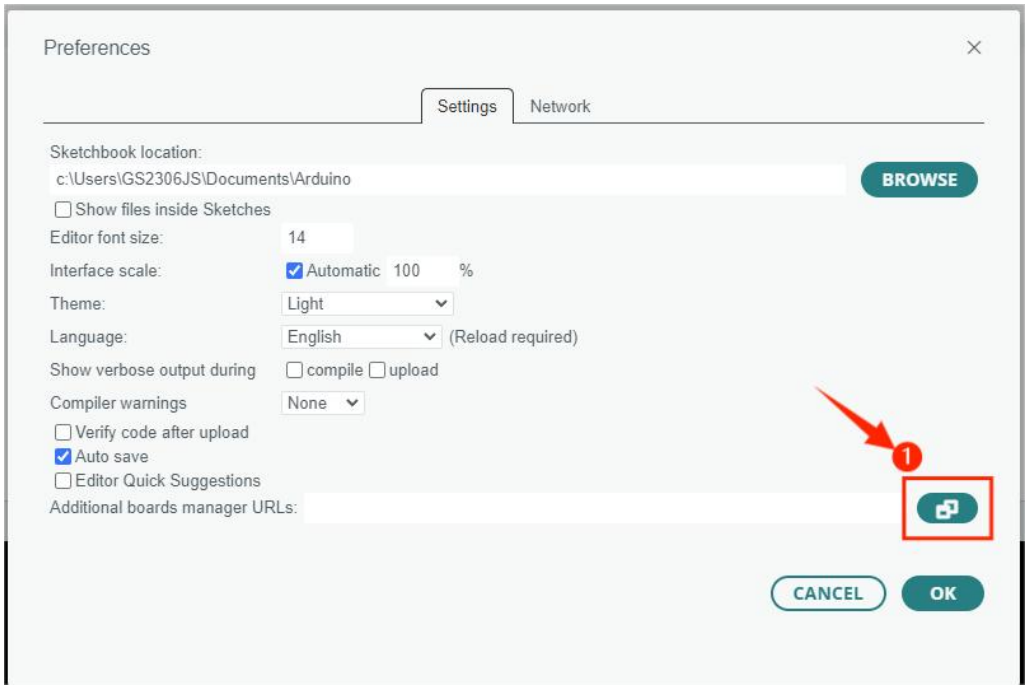


Next we need to install the ESP8266 library. Follow these steps:

①Go to File>Preferences.



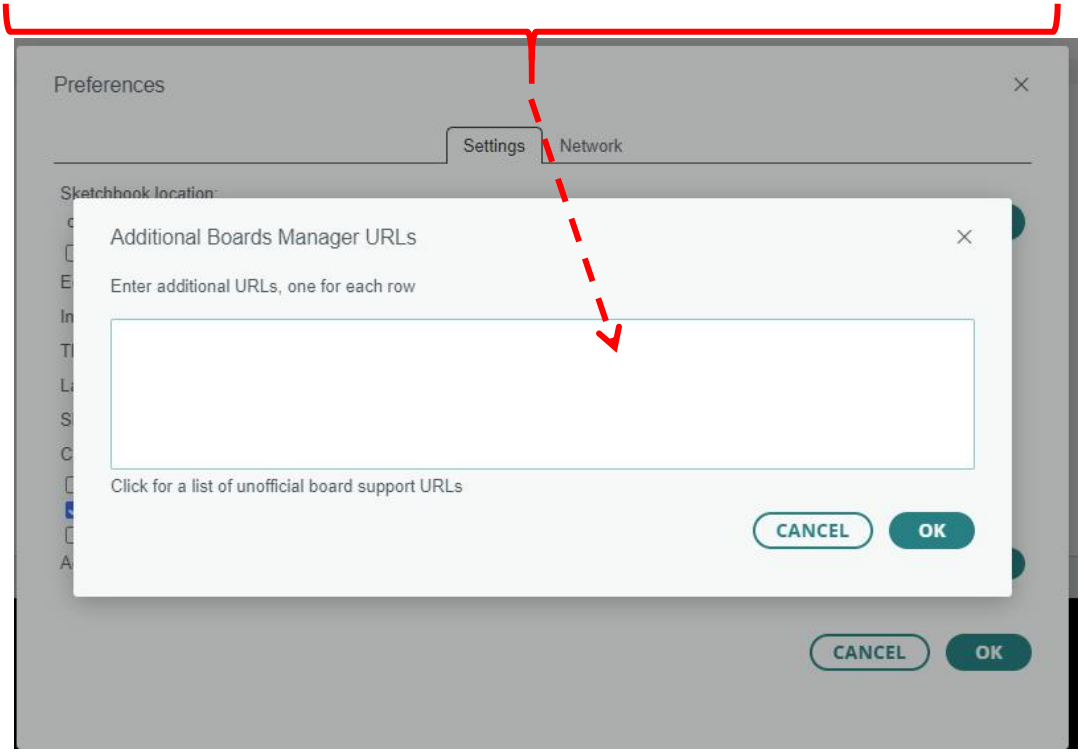
②Add controller board management address URL.



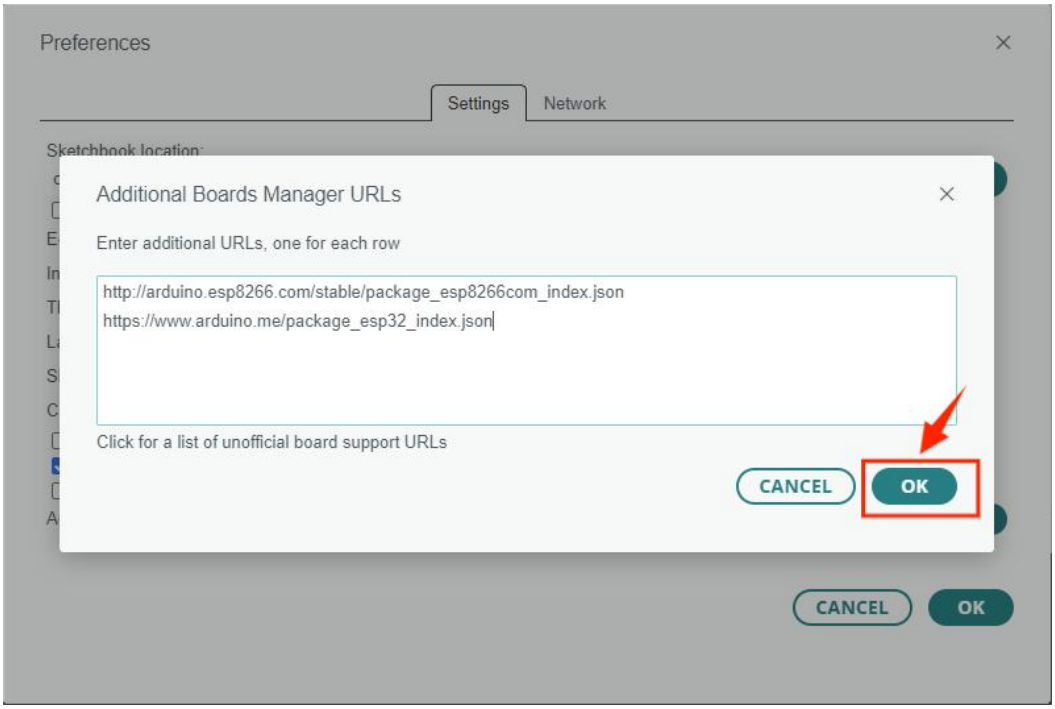
③Copy the URL in the lower text box and add it to the "Additional Boards Manager URLs".

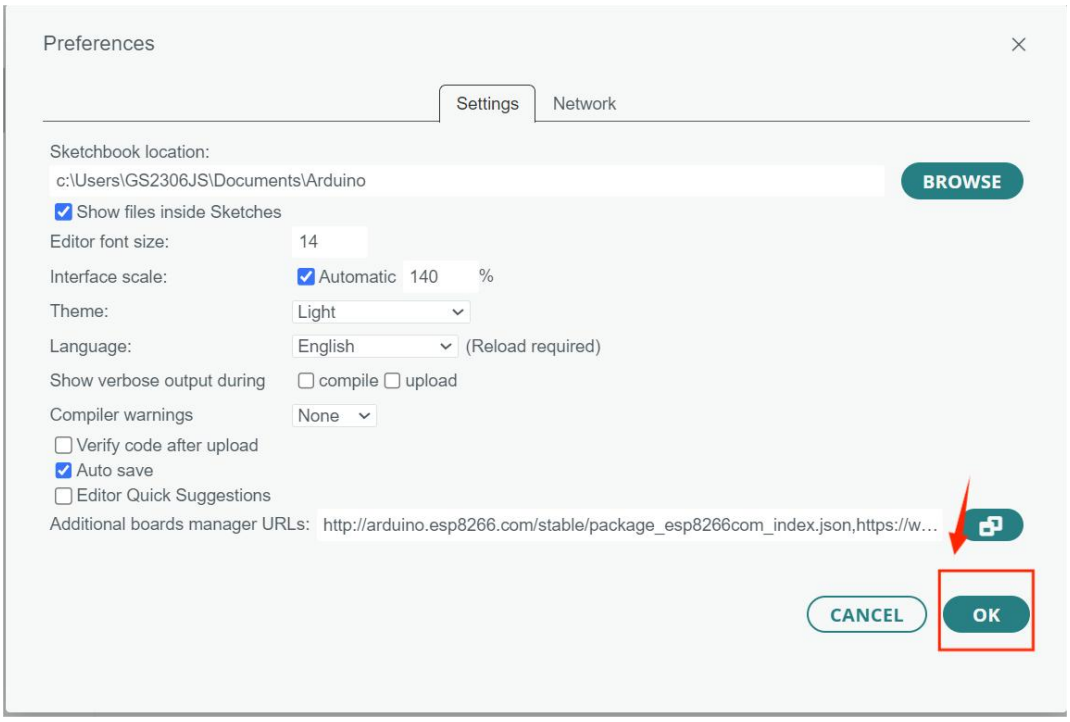
http://arduino.esp8266.com/stable/package_esp8266com_index.json

https://www.arduino.me/package_esp32_index.json

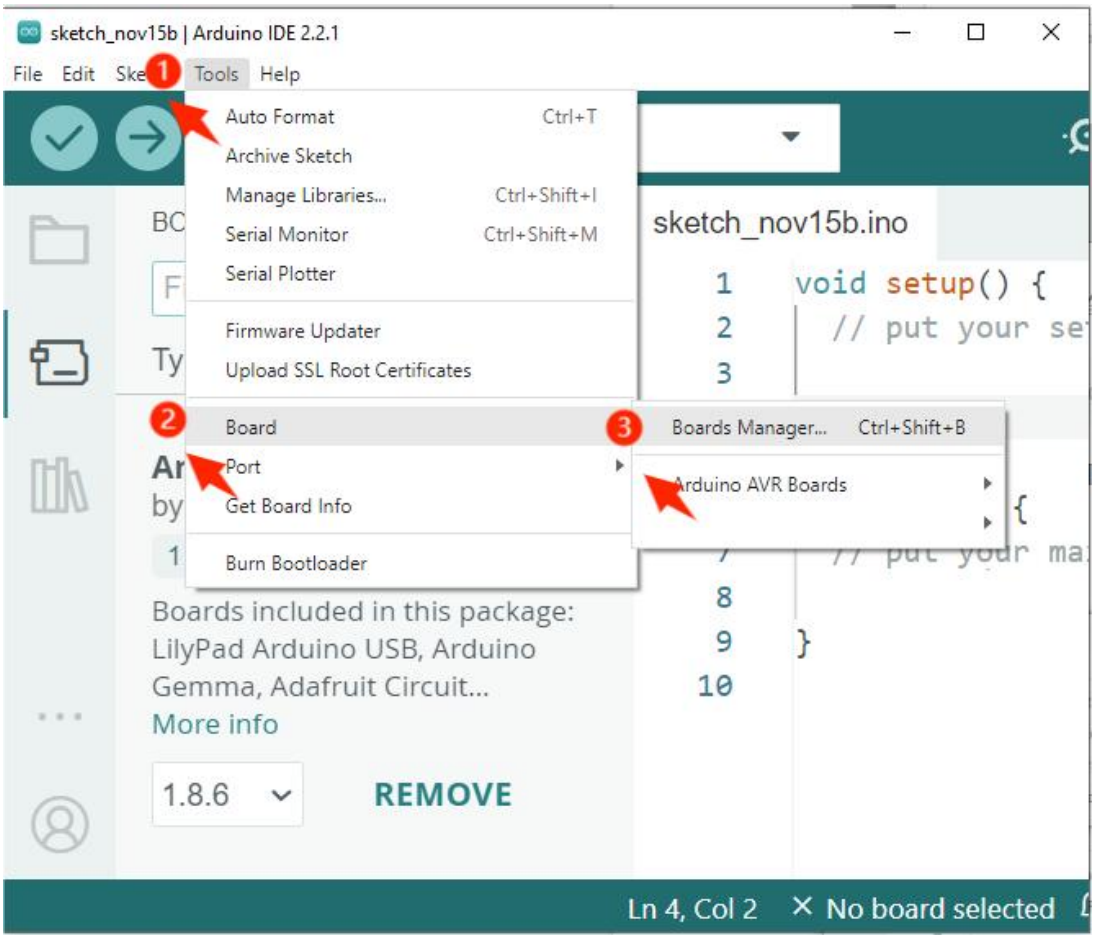


④After adding the URL, click OK.



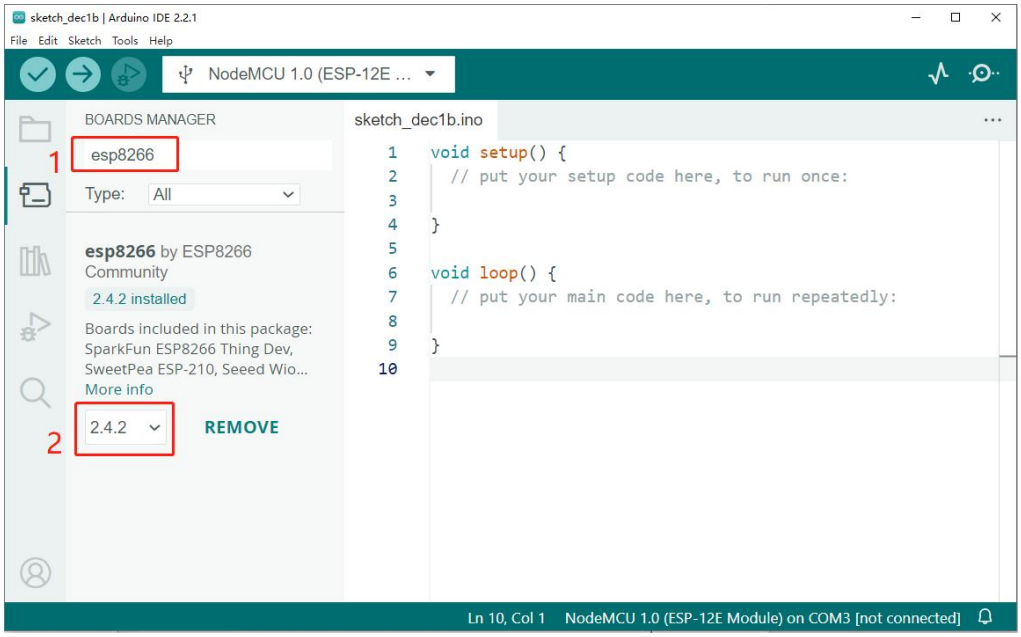


⑤Go to Tools>Board>Boards Manager... .

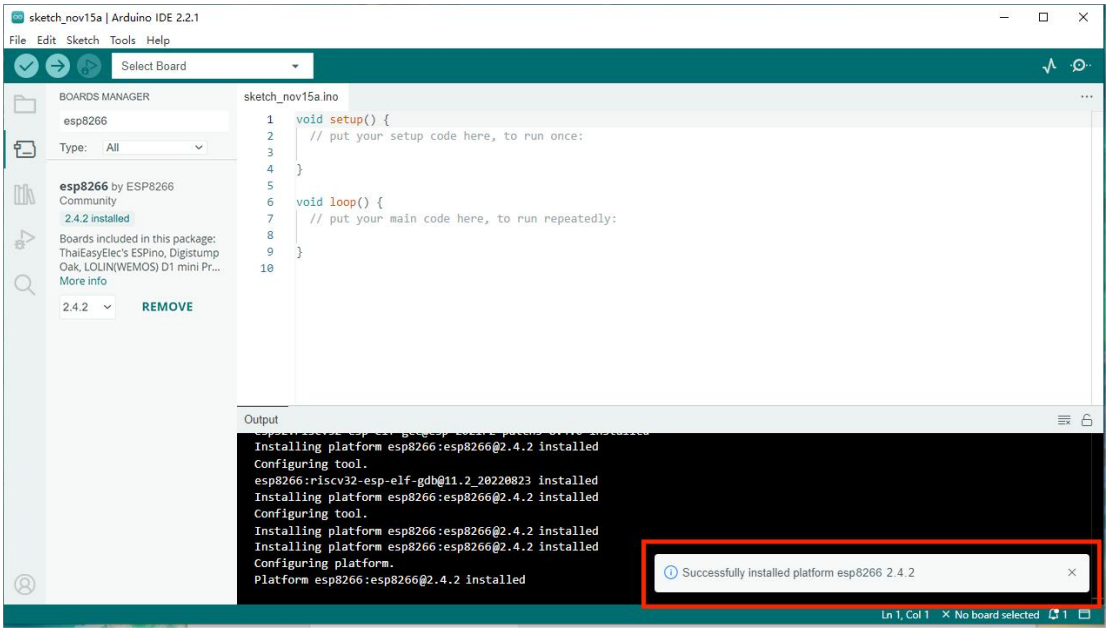


⑥Search for "esp8266" in the BOARDS MANAGER search bar and **install only version 2.4.2, do not install other versions.(Because other versions**

of the steering gear library may be different, resulting in abnormal operation.)



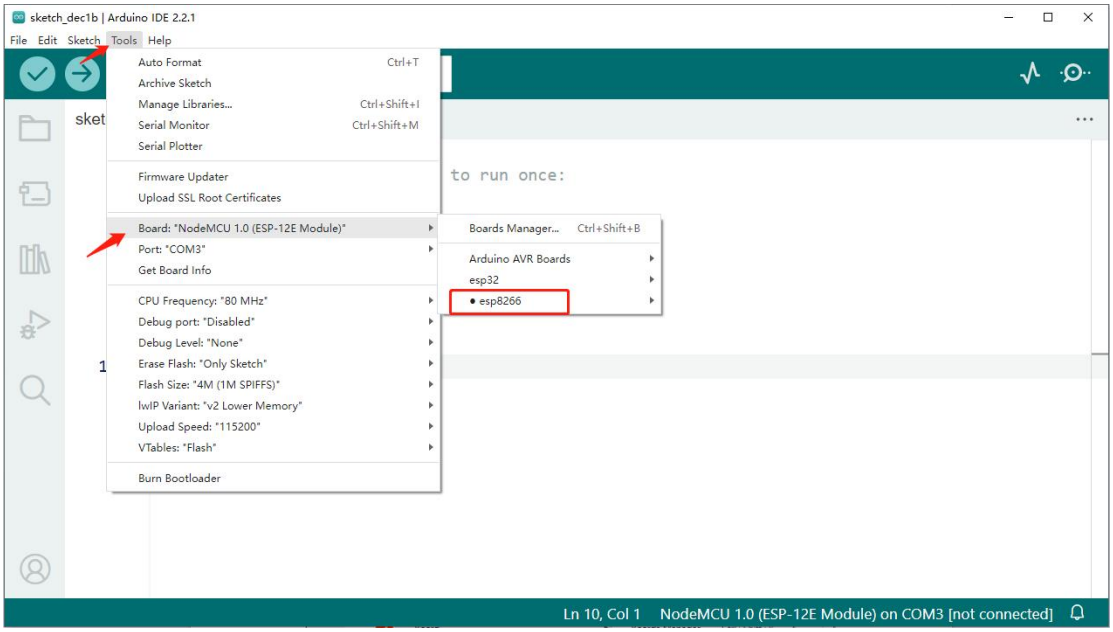
⑦Please wait until the installation is complete and turn off the Arduino IDE.



Attention:

Because the package is published on github, the network speed is affected, if the installation is not successful, try a few times.

⑧Re-open the Arduino IDE and select Tools>Board, you will find that the esp8266 board appears.

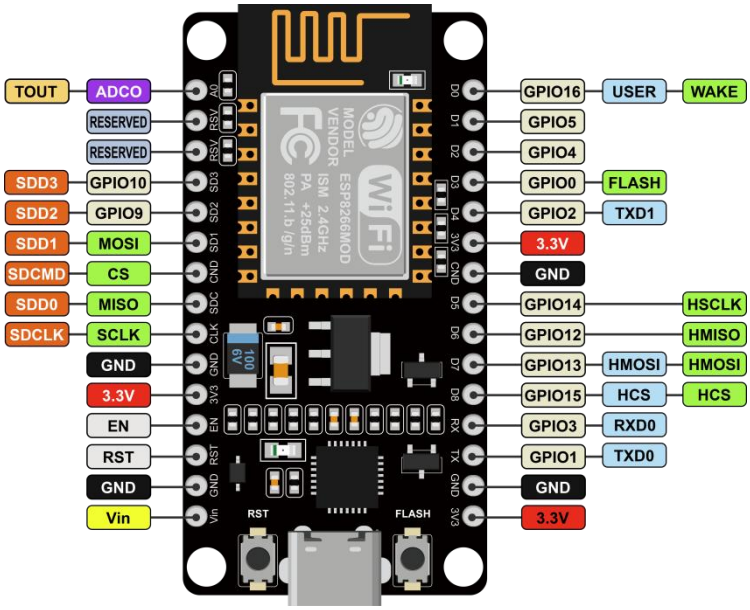


Ⅱ.Hardware knowledge

1.Recognize ESP8266

ESP8266 is a low-power, high-performance WiFi module. It integrates WiFi function and TCP/IP protocol stack, and can communicate with the main controller through the serial port. It is a module suitable for mobile devices, wearable electronics, and iot applications.

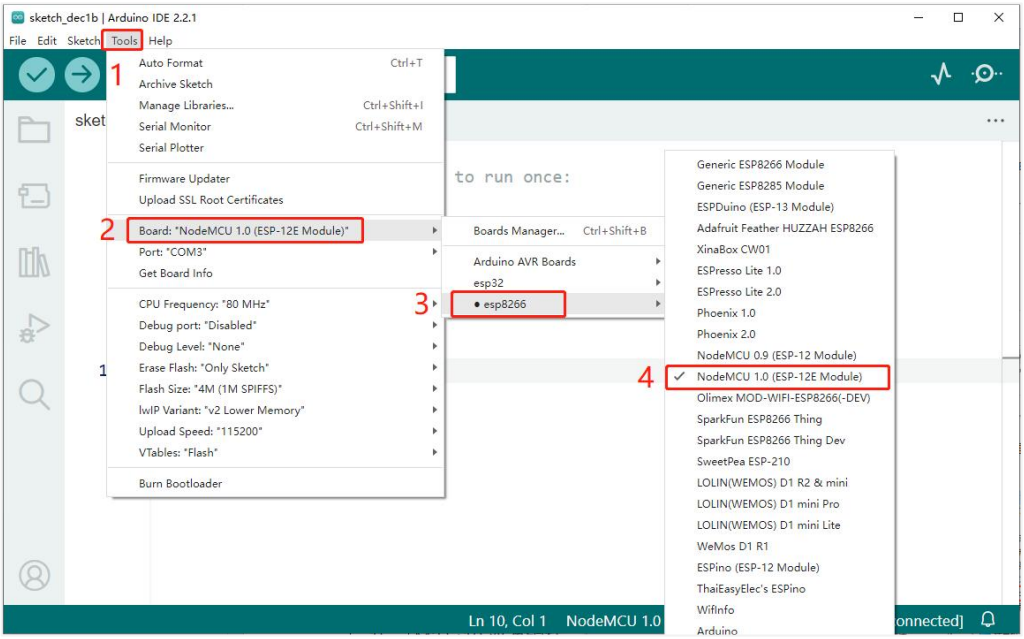
ESP8266 controller board includes: ESP-01, ESP-12E, NodeMCU and so on.



2.Test the development environment

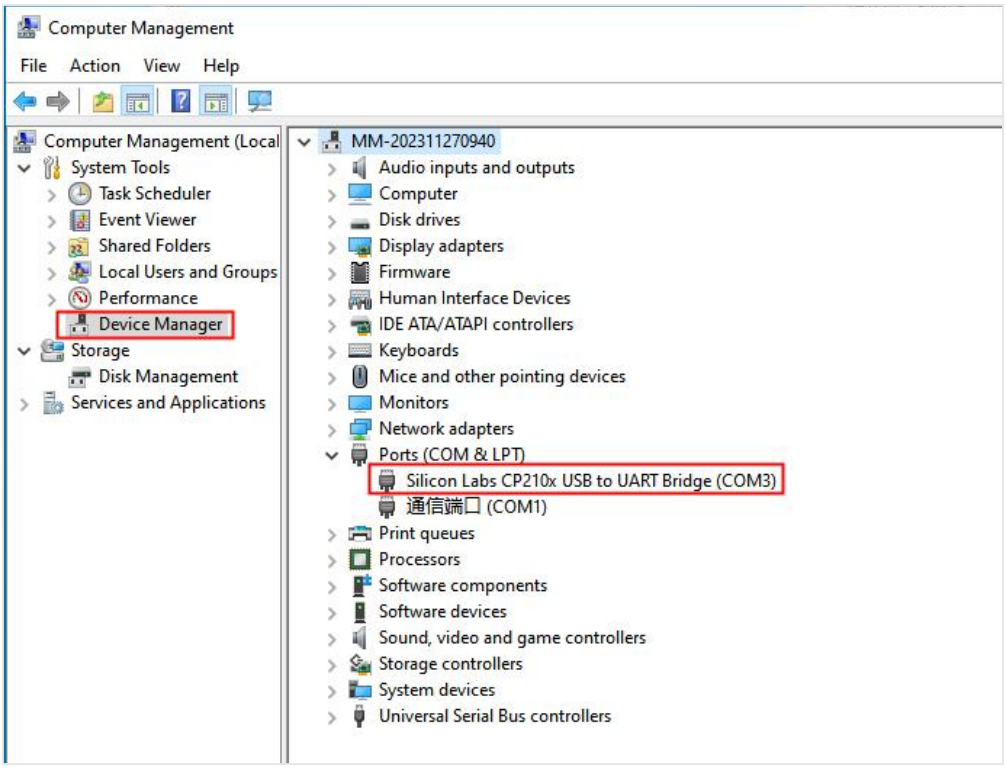
After installing Arduino IDE and ESP8266 controller board extension library, you can test whether the development environment is successfully built through a simple program, please follow the steps below:

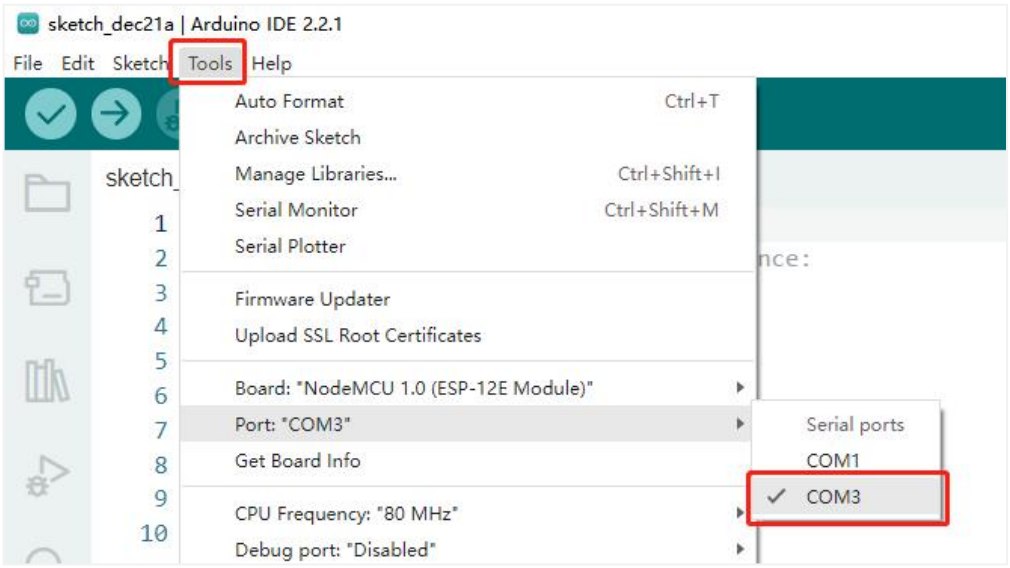
①Connect the controller board to the computer > Open Arduino IDE> Click Tools > Select ESP8266> Select NodeMCU1.0(ESP-12E Model).



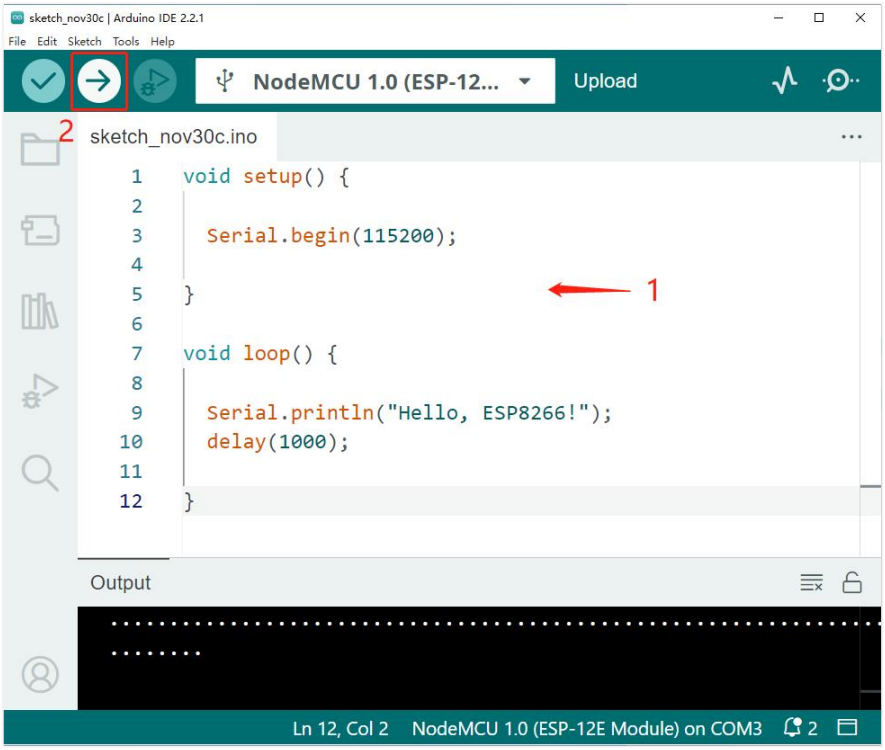
②Select the serial port (you can check the serial port number in the computer equipment manager, and then see whether the following port appears the serial port

number, because each controller board has a different COM number, please select according to the actual COM number displayed).





③Open "1.1 Hello_esp8266.ino" in English\Arduino\2.Arduino program\Lesson 1\1.1hello_esp32, connect ESP8266 controller board and computer with USB cable, select the correct controller board and port, Burn the code to the ESP8266 controller board, select 115200 baud rate, you can see the serial port monitor constantly output "Hello, ESP8266!" .



Ⅲ.Servo drive

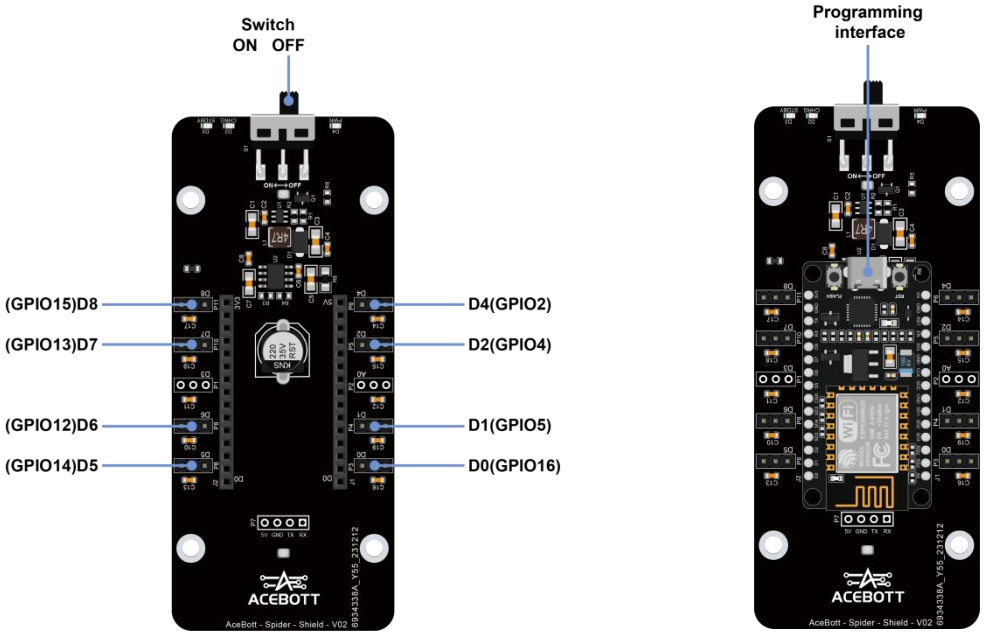
1.Know the servo extension board

Servo expansion board is mainly used to connect eight servos, and under the control of ESP8266 controller board, drive the servo to complete different actions. The following points should be Attentiond when using the servo expansion board:

- Connect the controller board: insert the pin of the ESP8266 controller board into the bus interface of the servo expansion board. Pay attention to the USB

port of the controller board and the switch of the expansion board on the same side.

- Charging mode: After connecting the battery and turning the switch to the OFF block, connect the USB charging cable and expansion board.
- Use mode: put the switch to the ON block.



2.Drive servo program

(1) Pin description of servo extension board

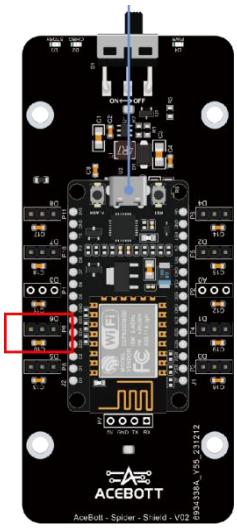
NO.	Expand plate pin number	GPIO pin number for ESP8266
1	D0	GPIO16
2	D1	GPIO5
3	D2	GPIO4
4	D4	GPIO2
5	D5	GPIO14
6	D6	GPIO12
7	D7	GPIO13
8	D8	GPIO15

(2) Servo pin definition

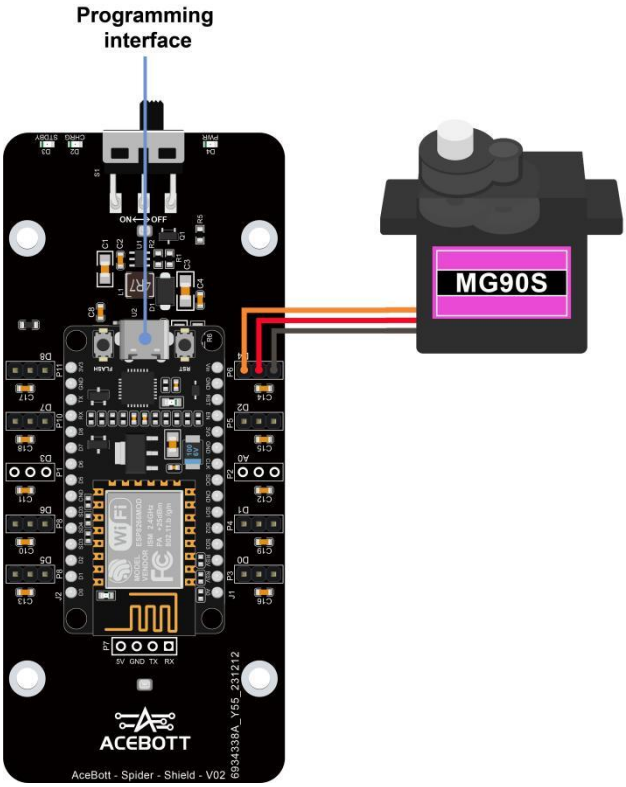
①Usually the servo has three control lines: power line, ground line and signal line.

②Servo pin definition: brown line -- GND, red line -- 5V, orange line -- signal.

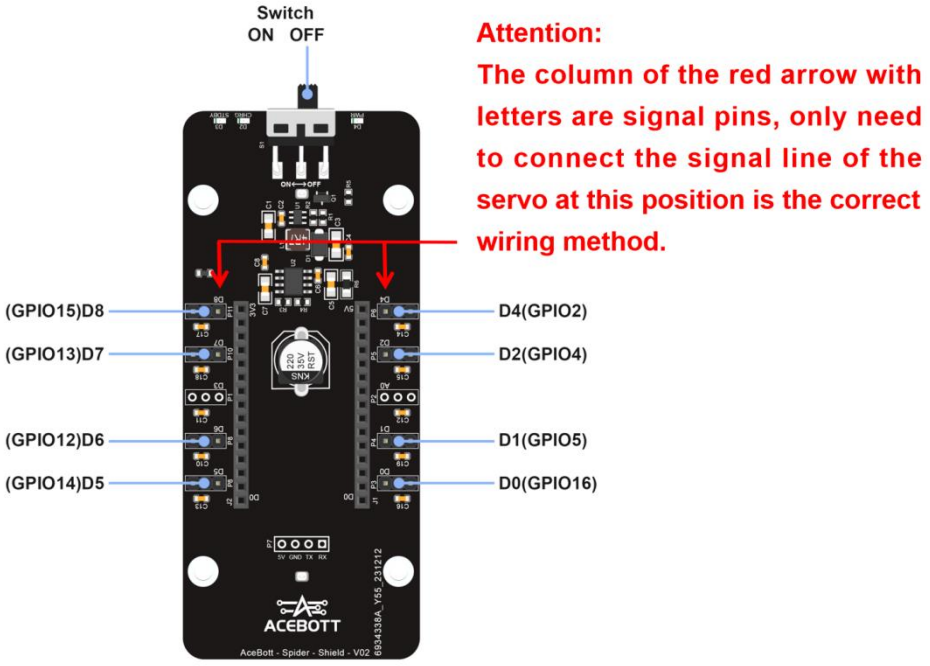
③Connect the servo and the expansion board, and the connection relationship is as follows.

Servo	Expansion board	Figure
Brown line	-	
Red line	+	
Orange line	D6(GPIO12)	

④Servo wiring diagram.



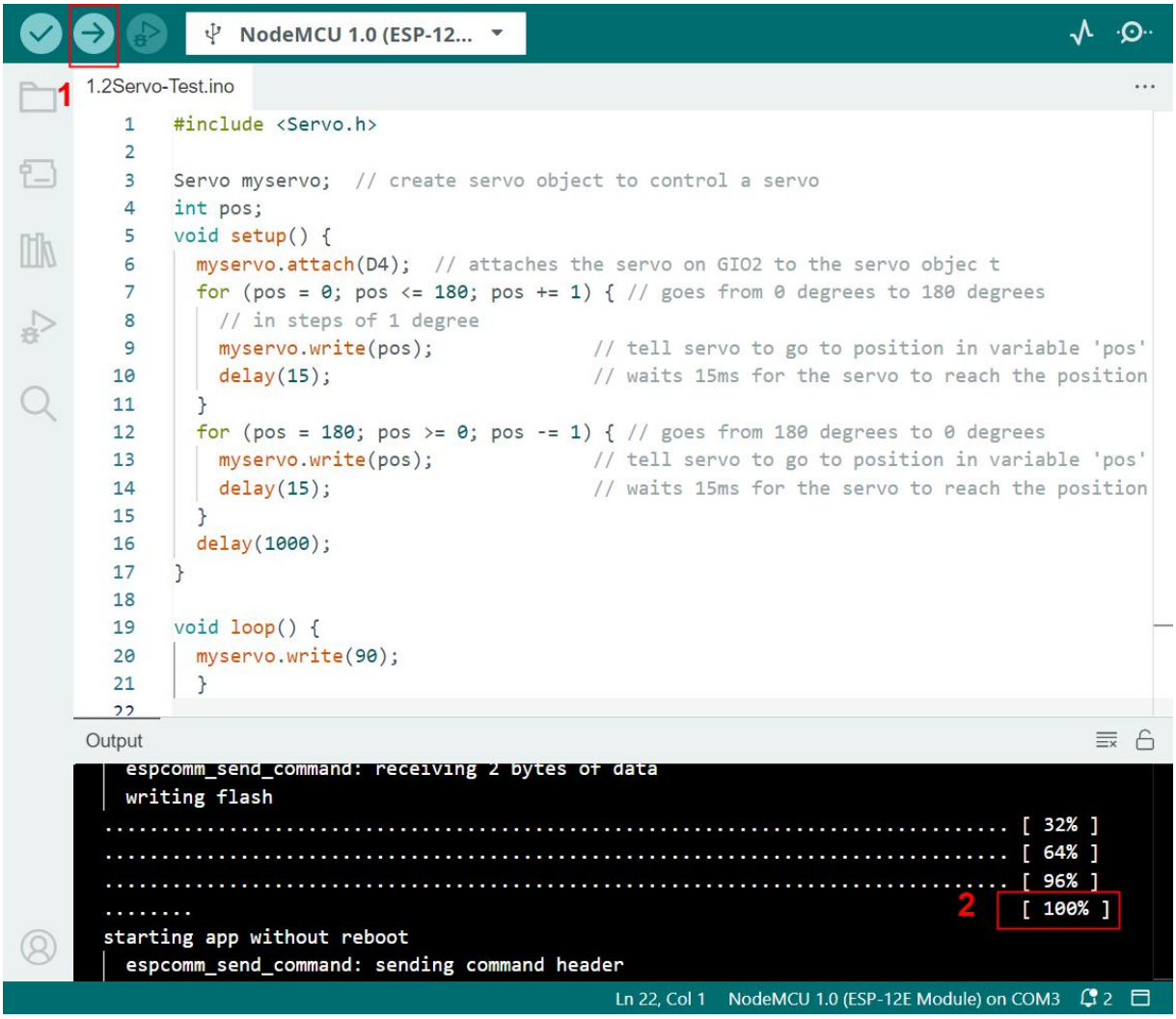
⑤The columns with letters on the expansion board are signal pins, as shown in the following figure.



(3) Servo drive steps

①Open "1.2 Servo-Test.ino" in English\Arduino\2.Arduino program\Lesson 1\1.2Servo-Test, connect ESP8266 controller board and computer with USB cable, select the correct controller board, processor and port, Burn the code to the ESP8266 controller board.

Before uploading, turn the toggle switch of the servo expansion board to OFF gear; After uploading, put the toggle switch of the servo extension board to the ON gear, at the same time, unplug the USB cable.



②If the steering gear is normal, it should rotate from 0 degrees to 180 degrees, then from 180 degrees to 0 degrees, and finally to 90 degrees.

③Take apart the 8 servos provided in the kit and test them respectively according to the same operation to ensure that each servos is normal and in good condition.

3.Upload servo zero program

Before assembling the quadruped robot, in order to install its structure smoothly, we need to upload the zero program in advance.

Open "1.3Zero_0.ino" in English\Arduino\2.Arduino program\Lesson 1\1.3zero_0, connect ESP8266 development board and computer with USB cable, select the correct development board, processor and port, and upload the code to the ESP8266 development board.

Lesson 2 Assembling the robot

Ⅰ.List of parts

Ⅱ.List of structural parts

Body x1

Body -Top x1

Coxa x8

Femur x4

Tibia x4

Ⅲ.Assembly steps

Attention: If you need to watch the assembly video, please click the link below and select the corresponding construction video to watch.

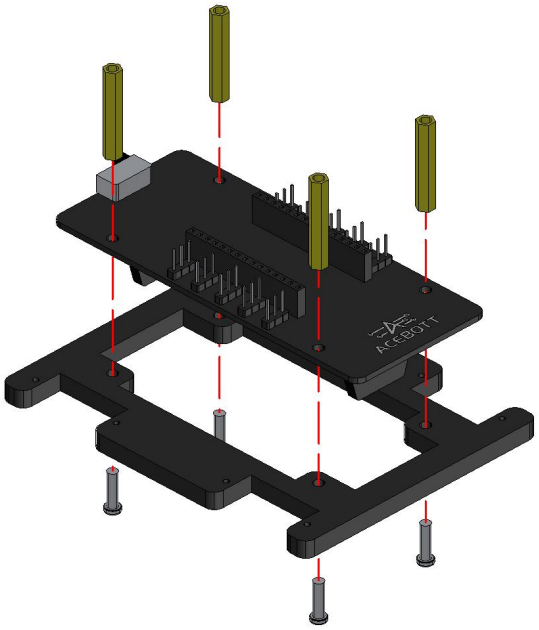
https://www.youtube.com/playlist?list=PLkW5fEtHNu6JIQVJpGalnKdkuk6Er4dIN

Or scan the QR code below.

1.Remove the protective paper attached to the acrylic structure

2.Install servo expansion board on the Body

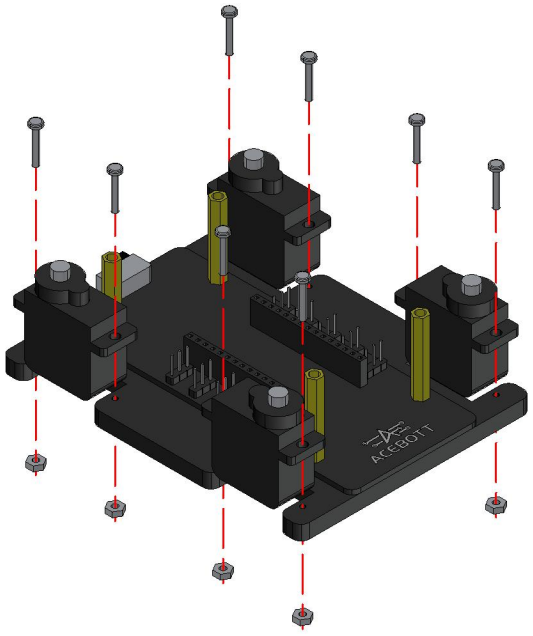
Parts List	
Name	Quantity
ESP32 Expansion board	1
Body Acrylic	1
M3*25MM Dual-Pass Copper Pillar	4
M3*10MM Round Head Screw	4



3.Install steering gear on the Body

Parts List	
Name	Quantity
Servo	4
M2*14MM Round Head Screw	8
M2 Nut	8

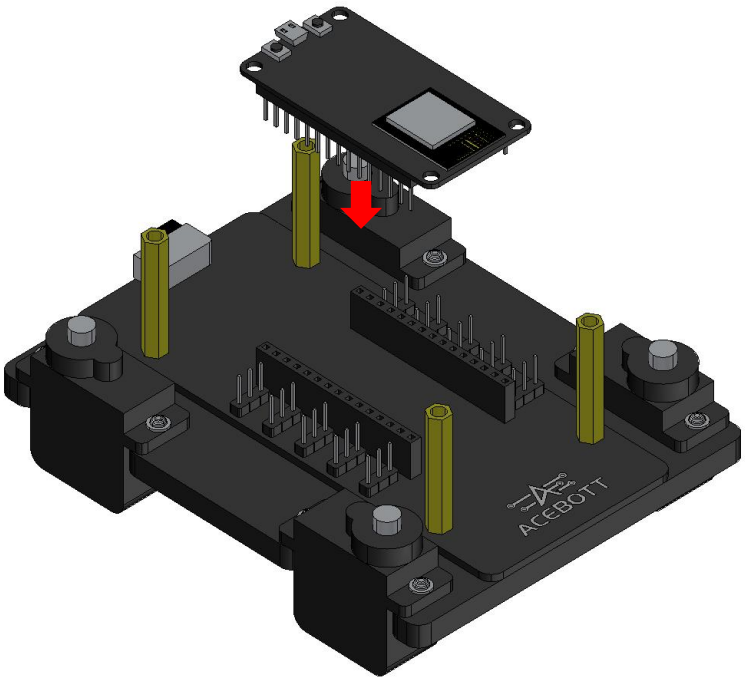
Attention: The direction of the servo axis should be closer to the outer edge of the long axis of the Body.



4.Install esp8266 controller board

Parts List	
Name	Quantity
esp8266 Controller Board	1

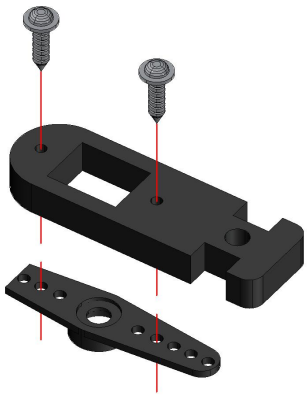
Attention: The USB port on the esp8266 Controller Board is on the same side as the switch on the steering gear expansion board.



5.Install steering rocker arm to Coxa

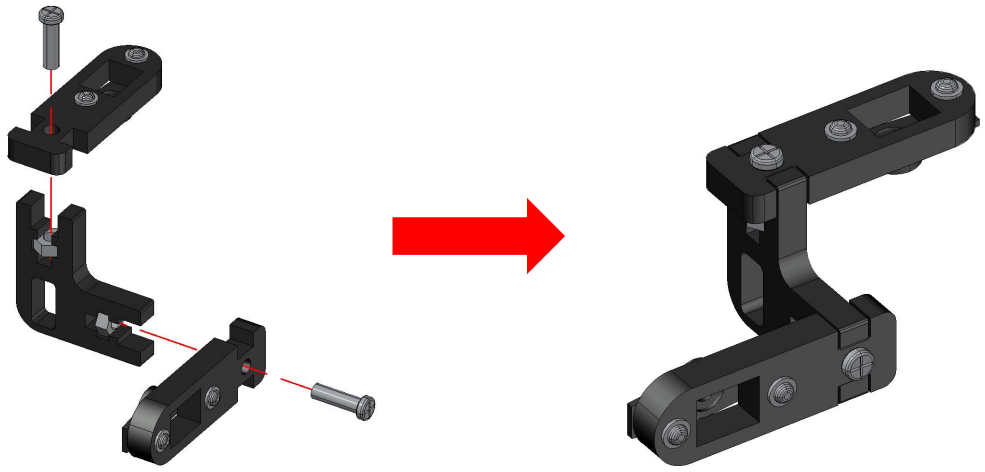
Parts List	
Name	Quantity
Coxa Structure	1
M2*8MM Round Head Tapping Screw	2
Single-axis Servo Horn	1

Attention:
(1) The M2*8MM Round Head Tapping Screw used in this kit are all in the steering gear package;
(2) A total of 8 structures are assembled.

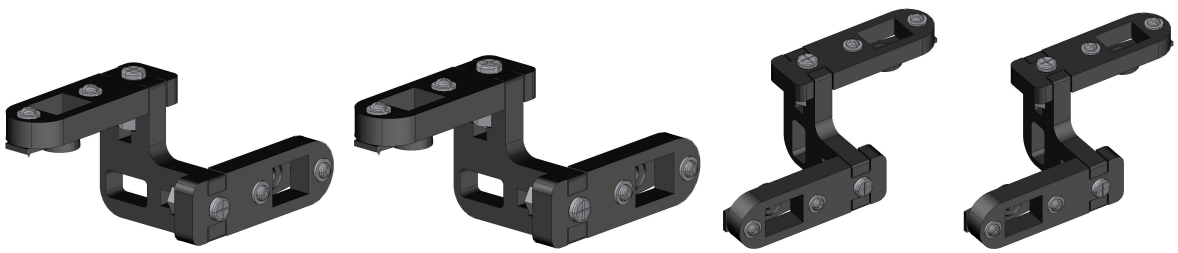


6.Install Coxa and Femur together

Parts List	
Name	Quantity
Femur Structure	1
M3*10MMRound Head Screw	2
M3 Nut	2



Attention: Similarly, install the remaining three, the effect after installation is shown below, pay attention to the direction, be sure to be consistent with the picture (2 to the left, 2 to the right).

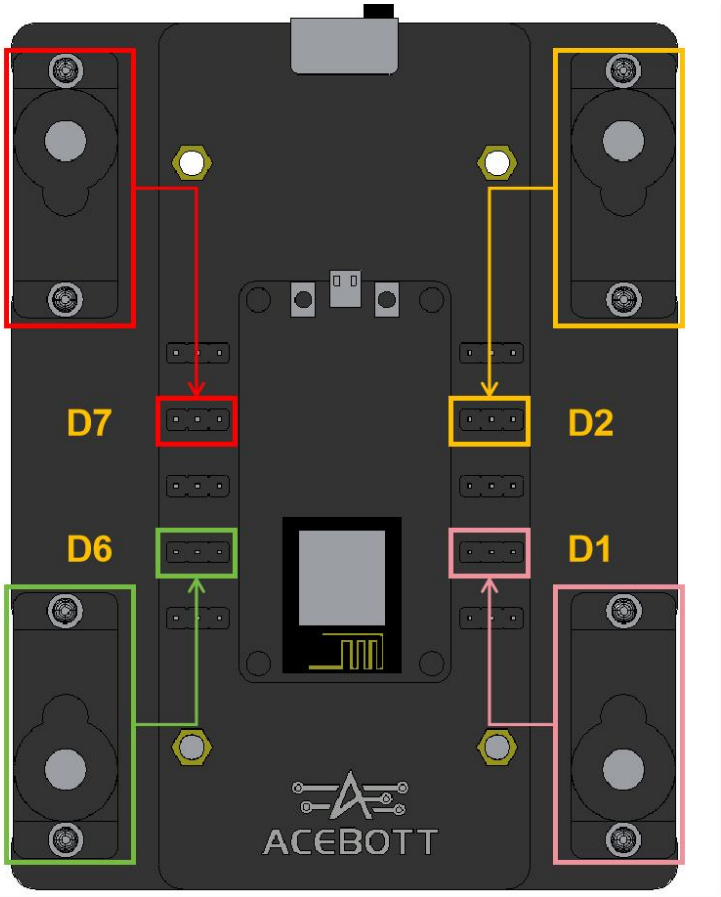


7.Connect the four servos on the Body to the servos expansion board

Insert the servo wire into the controller board as shown below (Attention the wiring sequence of the servo, the brown is GND, the red is VCC, and the orange is the signal wire in the servo wire, remember not to connect the reverse).

Attention:

The brown wire of the servo needs to be connected to the outermost pin. (The pin farthest from the center of the robot)



Attention:

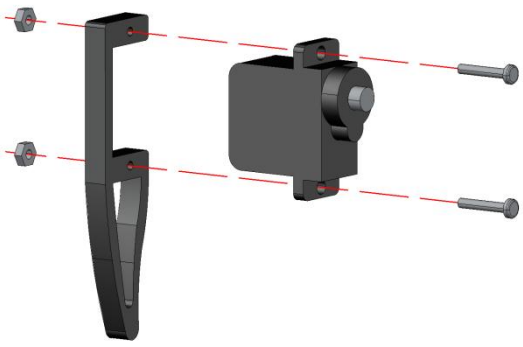
①This step requires installing a fully charged 18650 battery into the battery slot of the steering gear expansion plate, and then turning the toggle switch to ON gear. Because when installing the arms and claws of the quadruped robot, it needs to be powered on and installed.

② Important!!! Avoid damage to the servo, please don't turn the shaft of the servo when the power is on!!!

(3)Please make sure to strictly follow the wiring instructions when connecting the module to the expansion board. Incorrect wiring may cause a short circuit and damage the controller board.

8.Install four calves

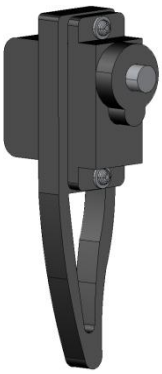
Parts List	
Name	Quantity
Tibia Structure	1
M2*14MMRound Head Screw	2
M2 Nut	2

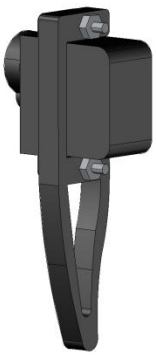


Attention:

(1)There are two Tibia installation as shown on the left, the other two Tibia installation as shown on the right.

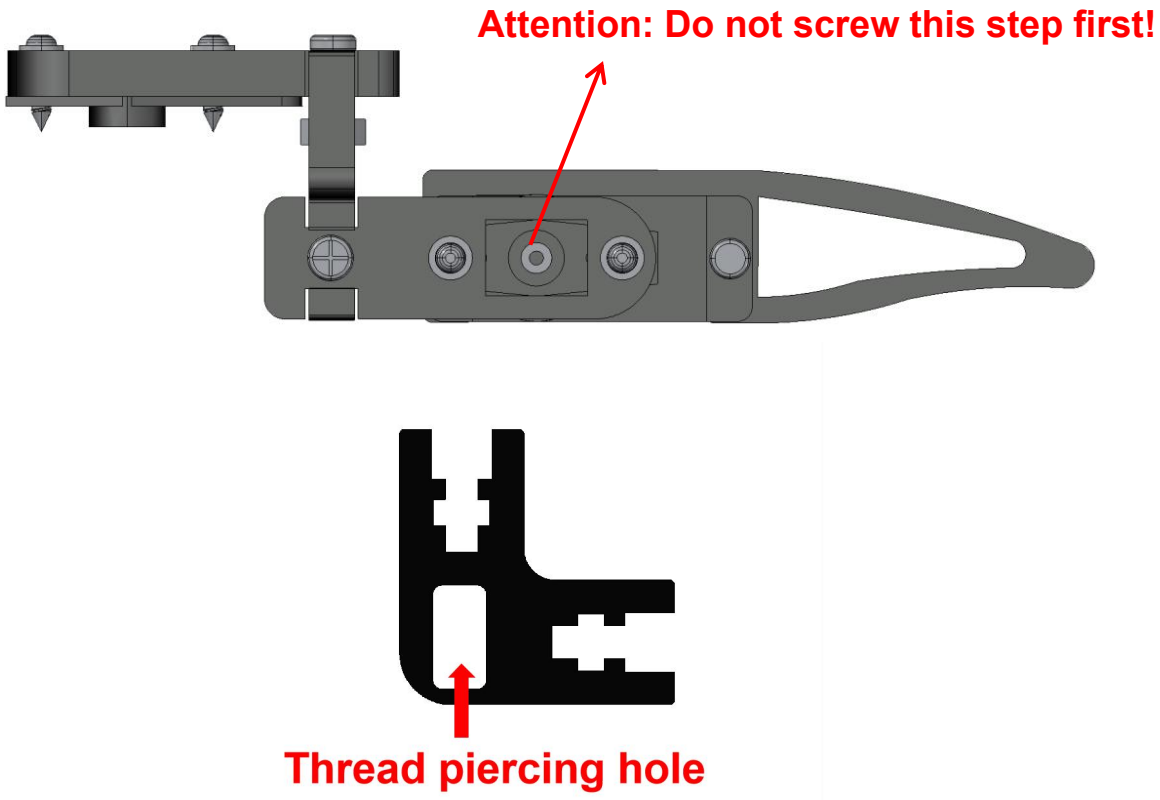
(2)After installation, you can see the following screenshot.





(3)Install the four paws and attach them to the calves, taking care to keep them as level as possible.

(4)Before fixing the four servos on the steering wheel, the servo line should be passed through the line hole, in addition, the steering wheel is not fixed here, and do not turn the servo during installation, as shown in the following figure.



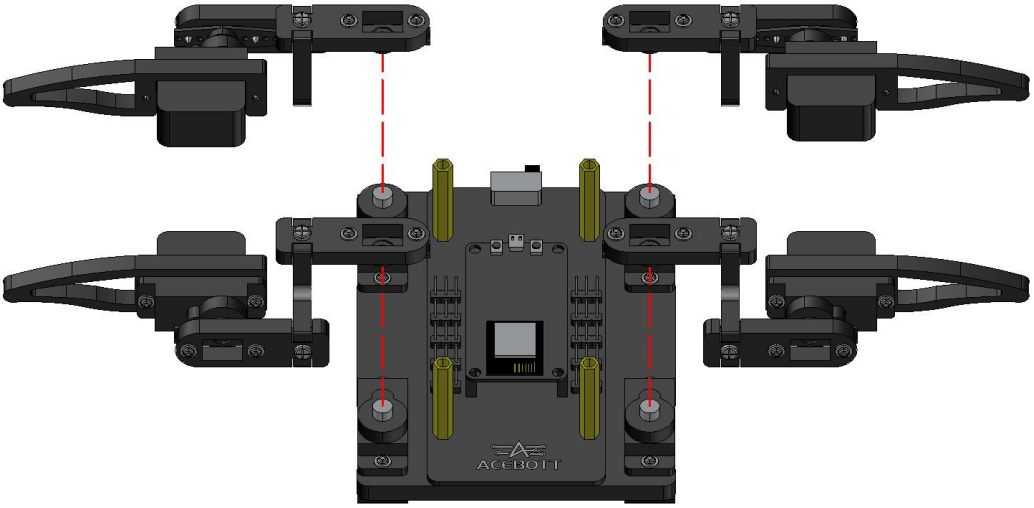
9.Connecting the four thighs

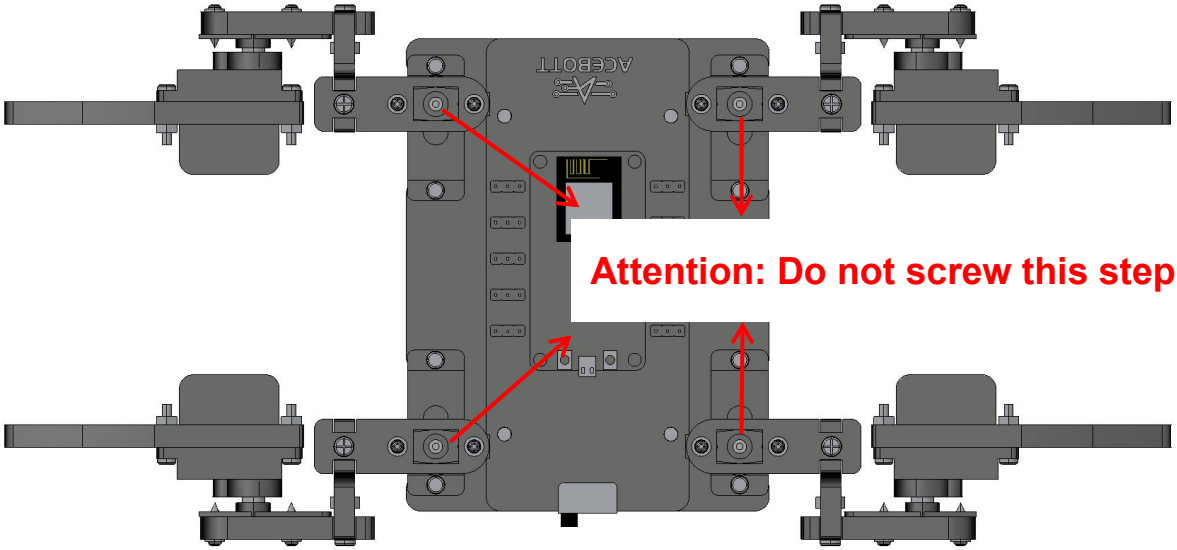
Attention:

(1)Install the steering wheel according to the Angle shown below, keeping it as vertical as possible.

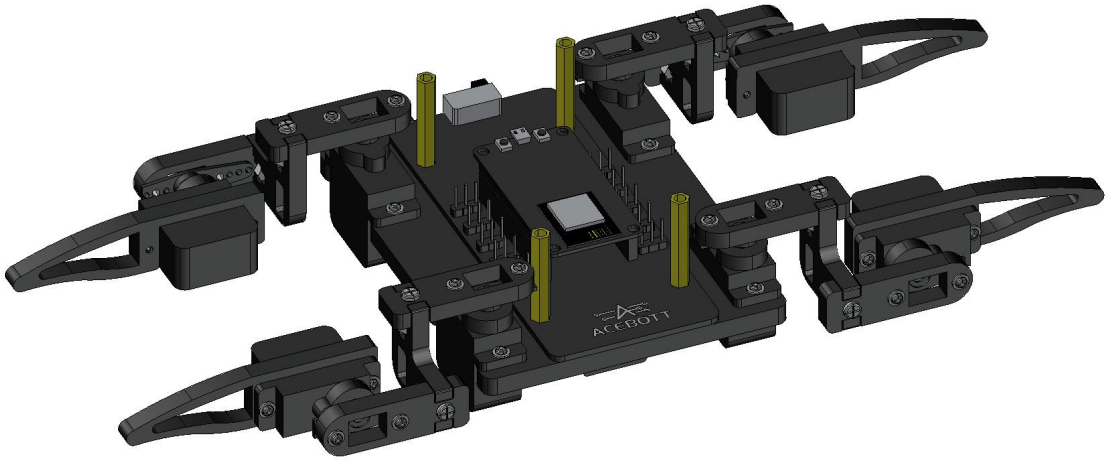
(2)Do not fix the steering wheel here.

(3)Important!!! Avoid damage to the servo, please don't turn the shaft of the servo when the power is on!!!





After installation, you can see the following screenshot.

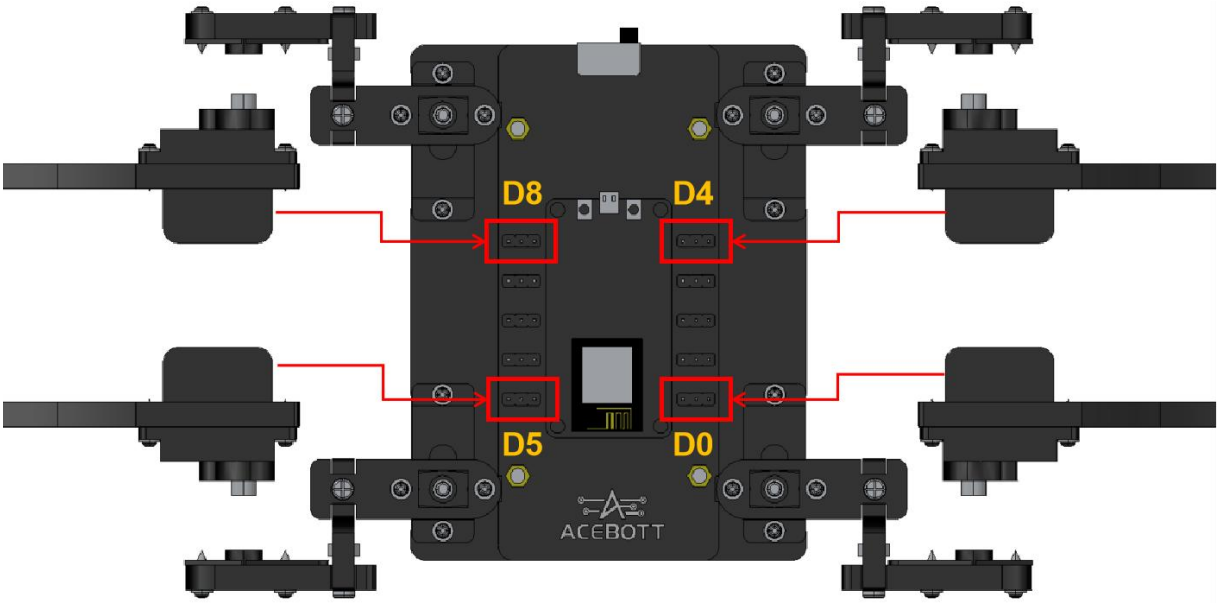


10.The servo is wired at the shank

Insert the servo wire into the controller board as shown below (Attention the wiring sequence of the servo, the brown is GND, the red is VCC, and theorange is the signal wire in the servo wire, remember not to connect the reverse).

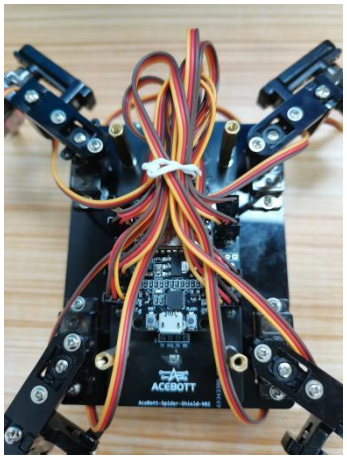
Attention:

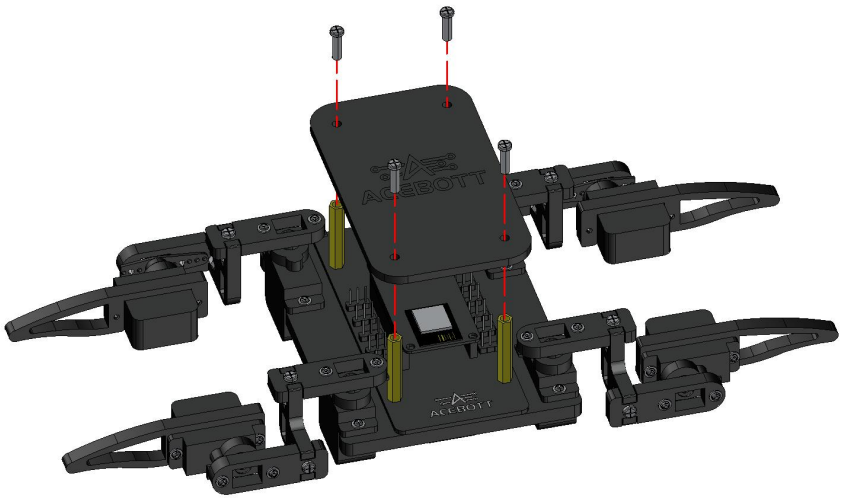
The brown wire of the servo needs to be connected to the outermost pin. (The pin farthest from the center of the robot)



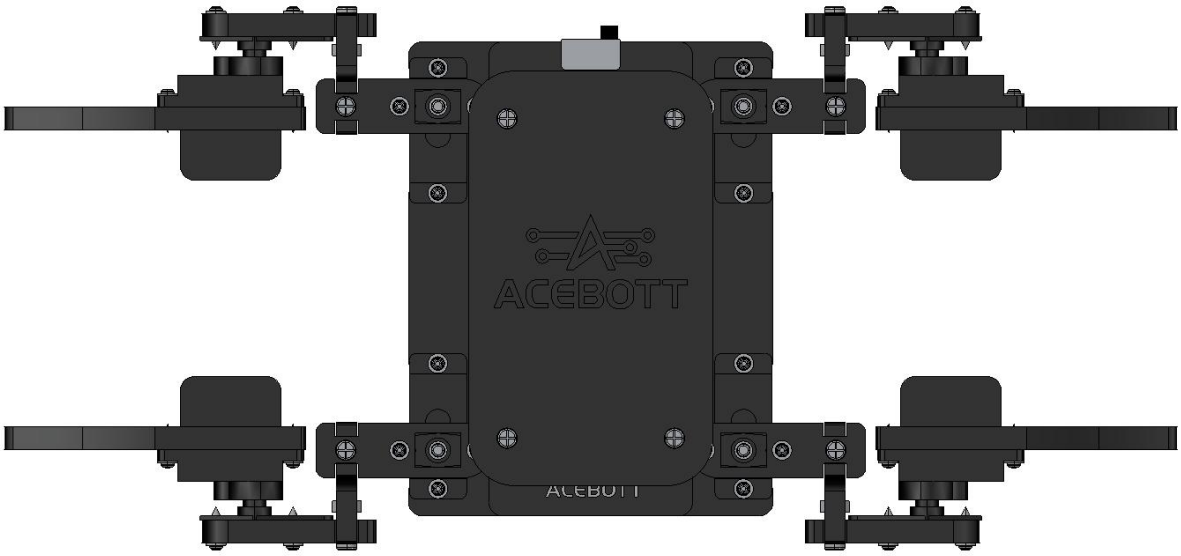
11.Install Body-Top

After finishing the wires, tie them with cable ties and install the Body-Top structural part with 4 M3*10 round head screws, as shown in the following picture.





12.The effect after installation



At this point, the four-legged bionic spider installation is complete!

Attention:

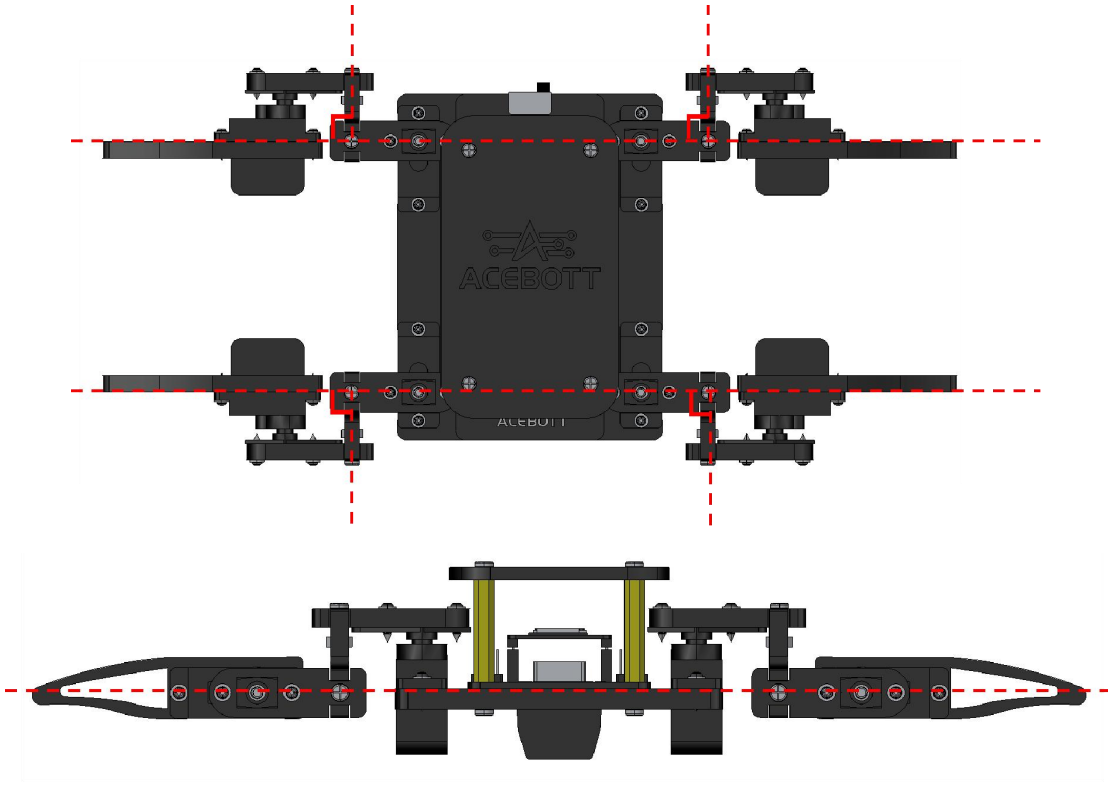
(1)Since the servo accessory kit comes in a standard packaging, the remaining parts such as the control disc are not required for use in this product;

(2)Please install one 18650 battery in the battery compartment of the spider robot, ensuring that the positive and negative poles are not reversed.

Lesson 3 Position Initialization

Ⅰ.Return to zero position

1.The zeroing position is shown in the figure



Before assembly, we have uploaded the zeroing program of the servo. During assembly, we have installed the robot as shown in the figure. This is the initial posture of the robot, that is, the zeroing position.

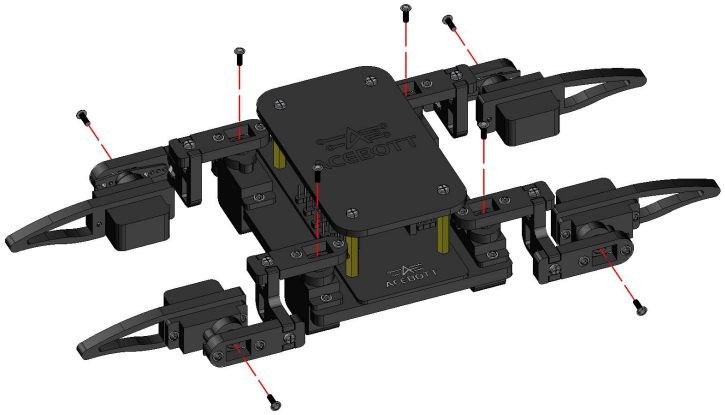
Turn on the battery switch, keep the quadruped robot in the power state, and then observe whether the posture of the quadruped robot remains at the zero position as shown in the figure. Because the last screw is not locked, if the posture of individual positions is very different, we can remove the structure with the difference and install it after corresponding to the posture again, but the axis of the servo can not be rotated during this process.

Attention:

Because the gear of the servo is clearance, not necessarily can be just installed to the most ideal state, there may be a slight deviation, this is a normal phenomenon.

2.Install steering wheel screws

After the final adjustment, eight servos are fixed with eight M2.5*4 round head screws.

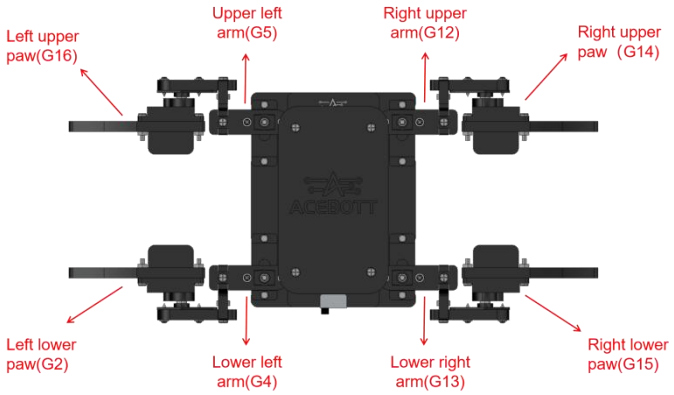


3.Corresponding position of servo pin

After installing the robot, each servo assumes different functions of the robot. If we want to control the Angle of each servo through the program and realize different functions, we must know the corresponding pin number of each servo.

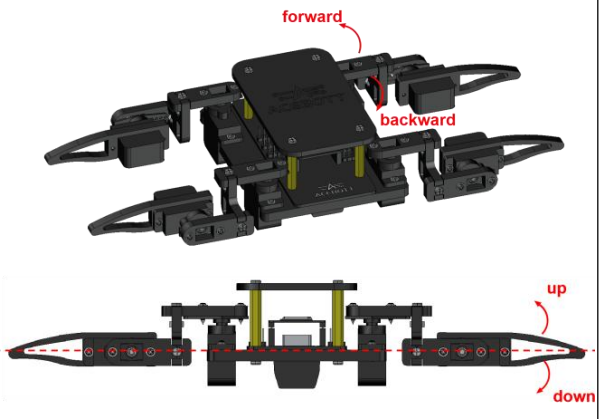
The robot uses a total of eight servos, four on the inside and four on the outside. For the sake of memory, we refer to the inner actuator as the robot arm and the outer actuator as the robot paw. At the same time, based on different servos are distributed in the position of up and down and left and right. So we distinguish the eight servo positions, which are divided into: right upper paw, right upper arm, right lower paw, right lower arm, left upper paw, left lower paw, and left lower arm.

The corresponding pin numbers are shown in the following table:

NO.	Pin number	Servo position	Figure
1	GPIO14	Right upper paw	
2	GPIO12	Right upper arm	
3	GPIO13	Lower right arm	
4	GPIO15	Right lower paw	
5	GPIO16	Left upper paw	
6	GPIO5	Upper left arm	
7	GPIO4	Lower left arm	
8	GPIO2	Left lower paw	

Ⅱ.The motion of the servo

In addition, the motion law of the servo at each position is shown in the following table:

NO.	Servo pin	Servo position	Law of motion	Figure
1	GPIO14	Right upper paw	The larger the Angle, the more the servo moves up	
2	GPIO12	Right upper arm	The larger the Angle, the more forward the servo moves	
3	GPIO13	Lower right arm	The larger the Angle, the more forward the servo moves	
4	GPIO15	Right lower paw	The larger the Angle, the more the servo moves down	
5	GPIO16	Left upper paw	The larger the Angle, the more the servo moves down	
6	GPIO5	Upper left arm	The larger the Angle, the more the servo moves backward	
7	GPIO4	Lower left arm	The larger the Angle, the more the servo moves backward	
8	GPIO2	Left lower paw	The larger the Angle, the more the servo moves up	

Lesson 4 The forward and backward movement of robots

The basic control actions of the robot include moving back and forth, left and right and rotating. As long as the use of these basic actions is mastered, other actions can be combined according to the basic actions.

Ⅰ.Standby mode program

Standby state is a state of preparation before the robot does other actions, it can quickly let the robot into other states.

Open "4.1standby.ino" in English\Arduino\2.Arduino program\Lesson 4\4.1standby, connect ESP8266 controller board and computer with USB cable, select the correct controller board and port, Burn the code to the ESP8266 controller board.

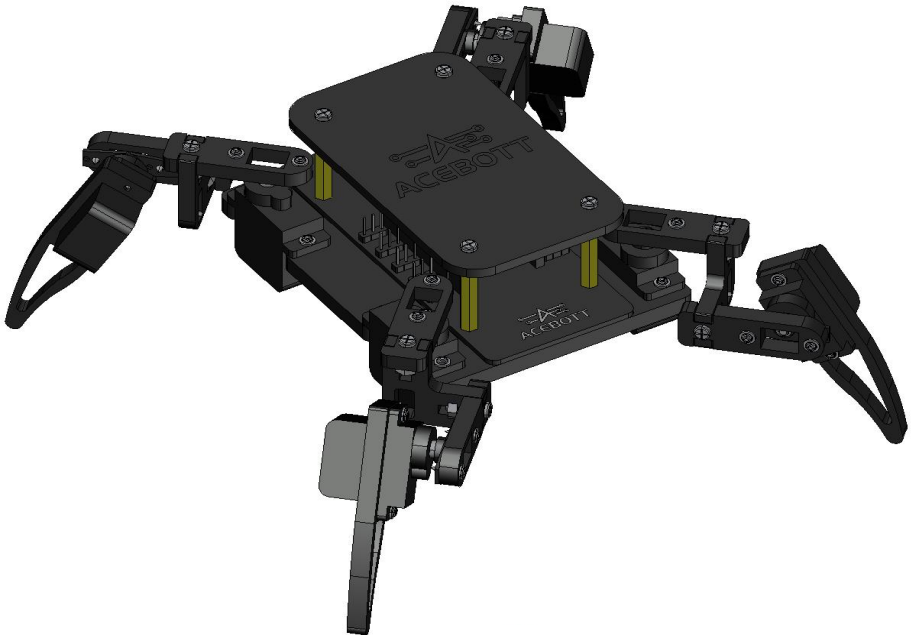
Before uploading, turn the toggle switch of the servo expansion board to OFF gear; After uploading, put the toggle switch of the servo extension board to the ON gear.

```
#include <Servo.h>
Servo servo_14;//Upper right [paw]
Servo servo_12;//Upper right [arm]
Servo servo_13;//Lower right [arm]
Servo servo_15;//Lower right [paw]
Servo servo_16;//Upper left [paw]
Servo servo_5;//Upper left [arm]
Servo servo_4;//Lower left [arm]
Servo servo_2;//Lower left [paw]

void setup(){
    servo_14.attach(14);
    servo_12.attach(12);
    servo_13.attach(13);
    servo_15.attach(15);
    servo_16.attach(16);
    servo_5.attach(5);
    servo_4.attach(4);
    servo_2.attach(2);
}
```

```
//Create the standby array
int array1[2][9] = {
//G14   G12   G13   G15   G16   G5   G4   G2   MS
  {90,  90,  90,  90,  90,  90,  90,  90,  500},//Center point position
  {70,  90,  90,  110, 110,  90,  90,  70,  500},//Standby mode
};
//A function that sets the standby state
void standby() {
      for (int i = 0; i <= 1; i = i + 1) {   //There are two sets of arrays that need to be traversed twice
          servo_14.write(array1[i][0]); //The servo executes row i and column 1
          servo_12.write(array1[i][1]); //The servo executes row i and column 2
          servo_13.write(array1[i][2]); //The servo executes row i and column 3
          servo_15.write(array1[i][3]); //The servo executes row i and column 4
          servo_16.write(array1[i][4]); //The servo executes row i and column 5
          servo_5.write(array1[i][5]); //The servo executes row i and column 6
          servo_4.write(array1[i][6]); //The servo executes row i and column 7
          servo_2.write(array1[i][7]); //The servo executes row i and column 8
          delay(array1[i][8]); //Row i, column 9, execution delay time
      }
}
//Run the standby function
void loop() {
    standby();
    delay(1000);
}
```

After uploading the program, we will find that the four arms of the robot are arranged in the shape of "X", and the four paws are also changed into the state of internal button to control the posture of the robot standing up. This is the standby state.



Ⅱ.Move forward program

Open "4.2forward.ino" in English\Arduino\2.Arduino program\Lesson 4\4.2forward, connect ESP8266 controller board and computer with USB cable, select the correct controller board and port, Burn the code to the ESP8266 controller board.

Before uploading, turn the toggle switch of the servo expansion board to OFF gear; After uploading, put the toggle switch of the servo extension board to the ON gear.

```
//Creates an array of action decompositions for moving forward
int array2[11][9] = { //There are 11 arrays with 9 columns
//G14  G12  G13  G15  G16  G5  G4  G2  MS （The latter is uniformly sorted according to this pin）
 {70, 90, 90, 110, 110, 90, 90, 70, 200},//Ready for standby
 {90, 90, 90, 110, 110, 90, 90, 90, 200},//Right upper paw and left lower paw are raised
 {90, 120, 90, 110, 110, 90, 60, 90, 200},//Right upper arm and left lower arm forward
 {70, 120, 90, 110, 110, 90, 60, 70, 200},//Right upper paw and left lower paw drop
 {70, 120, 90, 90, 90, 90, 60, 70, 200},//The left upper paw and right lower paw are raised
 {70, 90, 90, 90, 90, 90, 90, 70, 200},//Right upper arm and left lower arm back
 {70, 90, 120,90, 90, 60, 90, 70, 200},//Left upper arm and right lower arm forward
 {70, 90, 120,110, 110, 60, 90, 70, 200},//Left upper paw and right lower paw drop
 {90, 90, 120,110, 110, 60, 90, 90, 200},//Right upper paw and left lower paw are raised
 {90, 90, 90, 110, 110, 90, 90, 90, 200},//The upper left arm and lower right arm back
 {70, 90, 90, 110, 110, 90, 90, 70, 200},//Right upper paw and left lower paw drop
};
//Sets the forward function
void forward() {
    for (int i = 0; i <= 10; i = i + 1) {
    servo_14.write(array2[i][0]);
    servo_12.write(array2[i][1]);
    servo_13.write(array2[i][2]);
    servo_15.write(array2[i][3]);
    servo_16.write(array2[i][4]);
    servo_5.write(array2[i][5]);
    servo_4.write(array2[i][6]);
    servo_2.write(array2[i][7]);
    delay(array2[i][8]);
    }
}
// Run the forward function (to pause, put the swing switch on the servo extension board into OFF gear)
void loop() {
    forward();// The robot may not necessarily walk in a straight line due to errors
}
```

After uploading the program, it can be found that the arm paws on the diagonal side of the robot do the same action, and the two diagonal arm paws move forward in an alternate running action.

Ⅲ.Move backward program

Open "4.3back.ino" in English\Arduino\2.Arduino program\Lesson 4\4.3back, connect ESP8266 controller board and computer with USB cable, select the correct controller board and port, and burn the code to the ESP8266 controller board.

Before uploading, turn the toggle switch of the servo expansion board to OFF gear; After uploading, put the toggle switch of the servo extension board to the ON gear.

```
//Creates a backward-moving action decomposition array
int array3[11][9] = {
  {70, 90, 90, 110, 110, 90, 90, 70, 200},//Ready for standby
  {90, 90, 90, 110, 110, 90, 90, 90, 200},//Right upper paw and left lower paw are
raised
  {90, 60, 90, 110, 110, 90, 120,90, 200},//Right upper arm and left lower arm back
  {70, 60, 90, 110, 110, 90, 120,70, 200},//Right upper paw and left lower paw drop
  {70, 60, 90, 90, 90, 90, 120,70, 200},//The left upper paw and right lower paw are
raised
  {70, 90, 90, 90, 90, 90, 90, 70, 200},//Right upper arm and left lower arm forward
  {70, 90, 60, 90, 90, 120,90, 70, 200},//The upper left arm and lower right arm back
  {70, 90, 60,110, 110, 120,90, 70, 200},//Left upper paw and right lower paw drop
  {90, 90, 60,110, 110, 120,90, 90, 200},//Right upper paw and left lower paw are
raised
  {90, 90, 90, 110, 110, 90, 90, 90, 200},//Left upper arm and right lower arm forward
  {70, 90, 90, 110, 110, 90, 90, 70, 200},//Right upper paw and left lower paw drop
};
//Sets the backoff function
void back() {
       for (int i = 0; i <= 10; i = i + 1) {
           servo_14.write(array3[i][0]);
           servo_12.write(array3[i][1]);
           servo_13.write(array3[i][2]);
           servo_15.write(array3[i][3]);
           servo_16.write(array3[i][4]);
           servo_5.write(array3[i][5]);
           servo_4.write(array3[i][6]);
           servo_2.write(array3[i][7]);
           delay(array3[i][8]);
       }
}
//Running the backoff program
void loop() {
     back();//Calling the backoff function
}
```

After uploading the program, it can be found that the arm paws on the diagonal side of the robot do the same action, and the two diagonal arm paws move backward in an alternate running action.

Ⅳ.Extending tasks

You've already seen how the robot moves forward and backward in the tutorial, so let's test your learning on an extended task. I suggest you try it first and then follow the tutorial.

(1)Task Description:

The loop realizes that the robot moves forward three times, then moves backward two more times, and finally enters the standby state.

(2)Reference program:

Open "4.4expand.ino" in English\Arduino\2.Arduino program\Lesson 4\4.4expand, connect ESP8266 controller board and computer with USB cable, select the correct controller board and port, and burn the code to the ESP8266 controller board.

Before uploading, turn the toggle switch of the servo expansion board to OFF gear; After uploading, put the toggle switch of the servo extension board to the ON gear.

Attention:

The initialization and function definitions are the same as before.Only the execution part of the program is shown here.

```
void loop() {
    for (int i = 0; i <= 2; i = i + 1){
      forward();//Move forward three times
     }
    for (int i = 0; i <= 1; i = i + 1){
      back();//Move backwards twice
     }
    standby();//Standby mode
}
```

Lesson 5 Right and Left rotation of the robot

Now that you have learned the procedure for the robot to go forward and backward, in this lesson, you will continue to learn about the left and right rotation of the robot. The left and right rotation function is very important for the movement of the robot, and it allows the robot to go to any position on the plane.

If you want to change the rotation Angle, you need to modify the Angle of G12, G4, G13, G5. Due to the actual rotation error and friction, usually its actual rotation Angle is only half of the rotation Angle of the servo at the [arm] position. For example, from 90° to 135°, the rotation Angle is 45°, but the actual rotation Angle is only 22.5°. If you change it from 90° to 150°, the rotation will be 60 degrees, so the actual rotation will be 30 degrees, and so on. However, due to structural reasons, the rotation range of the robot servo is usually limited, so it is not recommended to change the Angle more than 60°.

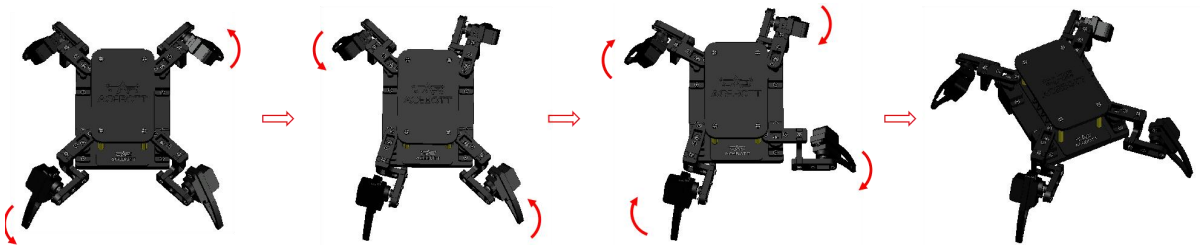
Ⅰ.Rotate left program

Open "5.1turn_left.ino" in English\Arduino\2.Arduino program\Lesson 5\5.1turn_left, connect ESP8266 controller board and computer with USB cable, select the correct controller board and port, Burn the code to the ESP8266 controller board.

Before uploading, turn the toggle switch of the servo expansion board to OFF gear; After uploading, put the toggle switch of the servo extension board to the ON gear.

```
//Create an array of turns to the left. By default, the robot rotates 22.5 degrees
int array6[8][9] = {
   //G14    G12    G13    G15    G16    G5    G4    G2    MS
      {70, 90, 90, 110, 110, 90, 90, 70, 200},//Ready for standby
      {90, 90, 90, 110, 110, 90, 90, 90, 200},//Right upper paw and left lower paw are raised
      {90, 135, 90, 110, 110, 90,135, 90, 200},//Right upper arm forward, left lower arm back
      {70, 135, 90, 110, 110, 90,135, 70, 200},//Right upper paw and left lower paw drop
      {70, 135, 90, 90, 90, 90, 135, 70, 200},//The left upper paw and right lower paw are raised
      {70, 135, 135, 90, 90, 135,135, 70, 200},//Put your upper left arm back and your lower right arm forward
      {70, 135, 135, 110, 110, 135,135, 70, 200},//Left upper paw and right lower paw drop
      {70, 90, 90, 110, 110, 90, 90, 70, 200},//Four arms rotation
};
//A function that sets the left turn
void turnleft() {
       for (int i = 0; i <= 7; i = i + 1) {
         servo_14.write(array6[i][0]);
         servo_12.write(array6[i][1]);
         servo_13.write(array6[i][2]);
         servo_15.write(array6[i][3]);
         servo_16.write(array6[i][4]);
         servo_5.write(array6[i][5]);
         servo_4.write(array6[i][6]);
         servo_2.write(array6[i][7]);
         delay(array6[i][8]);
       }
}
// Run the left turn (initialization and function definition are the same as before)
void loop() {
     turnleft();
}
```

After uploading the program, it can be found that the robot first rotates the four arms, and finally realizes the rotation action of the body to the left.



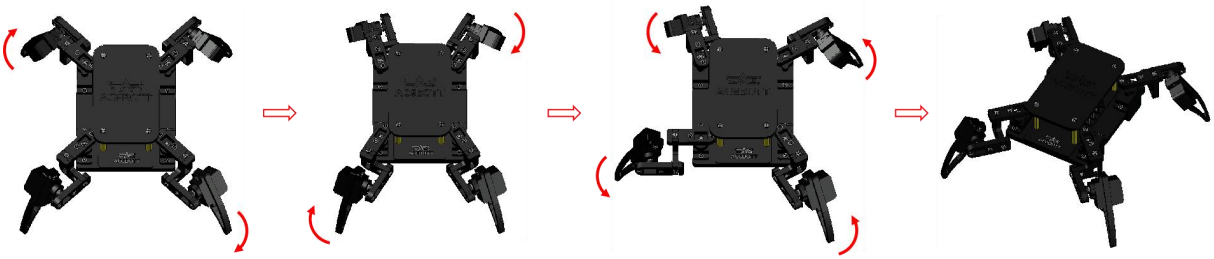
Ⅱ.Rotate right program

Open "5.2turn_right.ino" in English\Arduino\2.Arduino program\Lesson 5\5.2turn_right, connect ESP8266 controller board and computer with USB cable, select the correct controller board and port, Burn the code to the ESP8266 controller board.

Before uploading, turn the toggle switch of the servo expansion board to OFF gear; After uploading, put the toggle switch of the servo extension board to the ON gear.

```
//Create an array of actions for turning right
int array7[8][9] = {
        {70, 90, 90, 110, 110, 90, 90, 70, 200},//Ready for standby
        {70, 90, 90, 90, 90, 90, 90, 70, 200},//The left upper paw and right lower paw
are raised
        {70, 90, 45, 90, 90, 45, 90, 70, 200},//The upper left arm goes forward and the
lower right arm goes back
        {70, 90, 45, 110, 110, 45, 90, 70, 200},//Left upper paw and right lower paw
drop
        {90, 90, 45, 110, 110, 45, 90, 90, 200},//Right upper paw and left lower paw are
raised
        {90, 45, 45, 110, 110, 45, 45, 90, 200},//Right upper arm back, left lower arm
forward
        {70, 45, 45, 110, 110, 45, 45, 70, 200},//Right upper paw and left lower paw
drop
        {70, 90, 90, 110, 110, 90, 90, 70, 200},//Four arms rotation
};
void turnright() {//A function that sets the right turn
        for (int i = 0; i <= 7; i = i + 1) {
          servo_14.write(array7[i][0]);
          servo_12.write(array7[i][1]);
          servo_13.write(array7[i][2]);
          servo_15.write(array7[i][3]);
          servo_16.write(array7[i][4]);
          servo_5.write(array7[i][5]);
          servo_4.write(array7[i][6]);
          servo_2.write(array7[i][7]);
          delay(array7[i][8]);
        }
}
void loop() {//Run the right turn
      turnright();
}
```

After uploading the program, it can be found that the robot first performs the arm rotation, and finally realizes the body rotation action to the right.



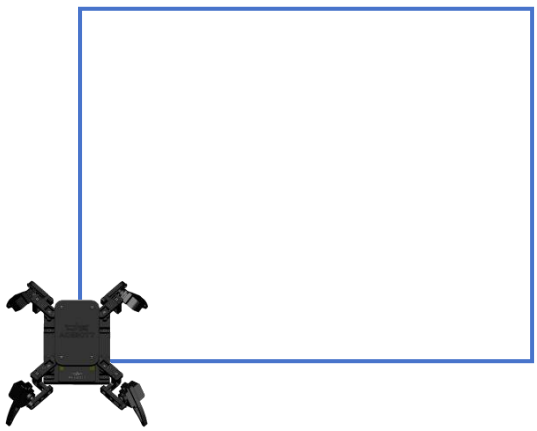
Ⅲ.Extending tasks

You've already seen how the robot moves left and right in the tutorial, so let's test your

learning on an extension task. I suggest you try it first and then follow the tutorial.

(1)Task Description:

Based on the content of the previous lessons, program the robot to walk a figure similar to a square.



(2) Reference program:

Open "5.3expand2.ino" in English\Arduino\2.Arduino program\Lesson 5\5.3expand2, connect ESP8266 controller board and computer with USB cable, select the correct controller board and port, and burn the code to the ESP8266 controller board.

Before uploading, turn the toggle switch of the servo expansion board to OFF gear; After uploading, put the toggle switch of the servo extension board to the ON gear.

Attention:

The parameters of the program should be adjusted according to the actual operation effect.

```
//Four edges need to repeat forward and rotation actions four times, preferentially using the traversal loop.
void loop() {
  for (int i = 0; i <= 3; i = i + 1){
     for (int i = 0; i <= 2; i = i + 1){
        forward(); //Three times forward
    }
     for (int i = 0; i <= 3; i = i + 1){
        turnright(); //One rotate it four times to the right, which is about 90 degrees
    }
  }
}
```

Lesson 6 Left and right movement of the robot

We've already seen how the robot can move backwards and forwards and rotate from side to side, but there are more interesting ways the robot can move, such as walking side-to-side. So follow this tutorial to learn how to make your robot move left and right.

Ⅰ.Move left program

Open “6.1left_move.ino” in English\Arduino\2.Arduino program\Lesson 6\6.1left_move, connect ESP8266 controller board and computer with USB cable, select the correct controller board and port, Burn the code to the ESP8266 controller board.

Before uploading, turn the toggle switch of the servo expansion board to OFF gear; After uploading, put the toggle switch of the servo extension board to the ON gear.

```
//Create an array of moves to the left
int array4[11][9] = {
    {70, 90, 90, 110, 110, 90, 90, 70, 200},//Ready for standby
    {70, 90, 90, 90, 90, 90, 90, 70, 200},//The left upper paw and right lower paw
are raised
    {70, 90, 60, 90, 90, 120,90, 70, 200},//The upper left arm and lower right arm
back
    {70, 90, 60, 110, 110, 120,90, 70, 200},//Left upper paw and right lower paw
drop
    {90, 90, 60, 110, 110, 120,90, 90, 200},//Right upper paw and left lower paw are
raised
    {90, 90, 90, 110, 110, 90, 90, 90, 200},//Left upper arm and right lower arm
forward
    {90, 120,90, 110, 110, 90, 60, 90, 200},//Right upper arm and left lower arm
forward
    {70, 120,90, 110, 110, 90, 60, 70, 200},//Right upper paw and left lower paw
drop
    {70, 120,90, 90, 90, 90, 60, 70, 200},//The left upper paw and right lower paw
are raised
    {70, 90, 90, 90, 90, 90, 90, 70, 200},//Right upper arm and left lower arm back
    {70, 90, 90, 110, 110, 90, 90, 70, 200},//Left upper paw and right lower paw
drop
};
//Sets the function to move to the left
void leftmove() {
      for (int i = 0; i <= 10; i = i + 1) {
        servo_14.write(array4[i][0]);
        servo_12.write(array4[i][1]);
        servo_13.write(array4[i][2]);
        servo_15.write(array4[i][3]);
        servo_16.write(array4[i][4]);
        servo_5.write(array4[i][5]);
        servo_4.write(array4[i][6]);
        servo_2.write(array4[i][7]);
        delay(array4[i][8]);
  }
}
// Run the program moved to the left
void loop() {
    leftmove();
}
```

After uploading the program, we can see that moving left is very similar to moving forward or backward, but in a different direction.

Ⅱ.Move right program

Open "6.2right_move.ino" in English\Arduino\2.Arduino program\Lesson 6\6.2right_move, connect ESP8266 controller board and computer with USB cable, select the correct controller board and port, Burn the code to the ESP8266 controller board.

Before uploading, turn the toggle switch of the servo expansion board to OFF gear; After uploading, put the toggle switch of the servo extension board to the ON gear.

```
//Create an array of actions to move to the right
int array5[11][9] = {
    {70, 90, 90, 110, 110, 90, 90, 70, 200},//Ready for standby
    {90, 90, 90, 110, 110, 90, 90, 90, 200},//Right upper paw and left lower paw are
raised
    {90, 60, 90, 110, 110, 90, 120,90, 200},//Right upper arm and left lower arm
back
    {70, 60, 90, 110, 110, 90, 120,70, 200},//Right upper paw and left lower paw
drop
    {70, 60, 90, 90, 90, 90, 120,70, 200},//The left upper paw and right lower paw
are raised
    {70, 90, 90, 90, 90, 90, 90, 70, 200},//Right upper arm and left lower arm
forward
    {70, 90, 120,90, 90, 60, 90, 70, 200},//Left upper arm and right lower arm
forward
    {70, 90, 120,110, 110, 60, 90, 70, 200},//Left upper paw and right lower paw
drop
    {90, 90, 120,110, 110, 60, 90, 90, 200},//Right upper paw and left lower paw are
raised
    {90, 90, 90, 110, 110, 90, 90, 90, 200},//The upper left arm and lower right arm
back
    {70, 90, 90, 110, 110, 90, 90, 70, 200},//Right upper paw and left lower paw
drop
};
//Sets the function to move to the right
void rightmove() {
      for (int i = 0; i <= 10; i = i + 1) {
        servo_14.write(array5[i][0]);
        servo_12.write(array5[i][1]);
        servo_13.write(array5[i][2]);
        servo_15.write(array5[i][3]);
        servo_16.write(array5[i][4]);
        servo_5.write(array5[i][5]);
        servo_4.write(array5[i][6]);
        servo_2.write(array5[i][7]);
        delay(array5[i][8]);
      }
}
// Run the program moved to the right
void loop() {
    rightmove();
}
```

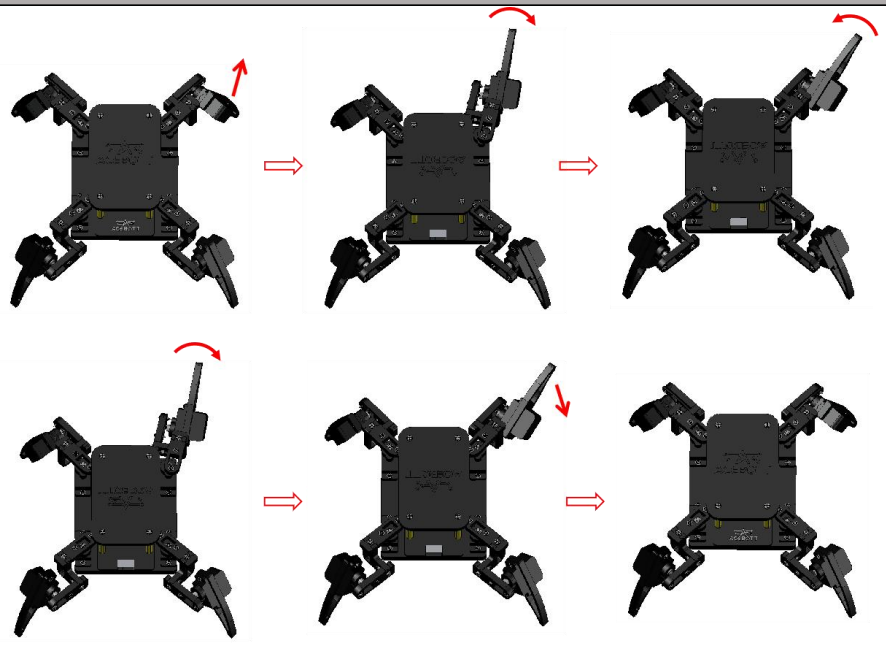
Ⅲ.Extending program

Moving left and right is relatively easy, so wouldn't it be more fun if the robot could also say hello to the audience after moving?

Open "6.3say_hello.ino" in English\Arduino\2.Arduino program\Lesson 6\6.3say_hello, connect ESP8266 controller board and computer with USB cable, select the correct controller board and port, Burn the code to the ESP8266 controller board.

Before uploading, turn the toggle switch of the servo expansion board to OFF gear; After uploading, put the toggle switch of the servo extension board to the ON gear.

```
//Create an array of hello actions
int array8[7][9] = {
      {70, 90, 135, 90, 90, 90, 90, 90, 400},//Left upper paw, right lower paw, left lower paw down
      {170, 90, 135, 90, 90, 90, 90, 90, 400},//Right upper paw raised
      {170, 130, 135, 90, 90, 90, 90, 90, 400},//Swing your right upper arm forward
      {170, 50, 135, 90, 90, 90, 90, 90, 400},//Swing your right upper arm back
      {170, 130, 135, 90, 90, 90, 90, 90, 400},//Swing your right upper arm forward
      {170, 90, 135, 90, 90, 90, 90, 90, 400},//Swing your right upper arm back
      {70, 90, 135, 90, 90, 90, 90, 90, 400},//Right upper arm down
};
void hello() {//Set the hello function
      for (int i = 0; i <= 6; i = i + 1) {
        servo_14.write(array8[i][0]);
        servo_12.write(array8[i][1]);
        servo_13.write(array8[i][2]);
        servo_15.write(array8[i][3]);
        servo_16.write(array8[i][4]);
        servo_5.write(array8[i][5]);
        servo_4.write(array8[i][6]);
        servo_2.write(array8[i][7]);
        delay(array8[i][8]);
  }
}
// Run the hello program
void loop() {
    hello();
}
```



Lesson 7 Robot dance program

We finally covered the basic movement, rotation and translation of the robot. I believe you already have some ideas that you can't wait to implement on the robot. This tutorial will teach you how to implement a few dance movements for the robot.

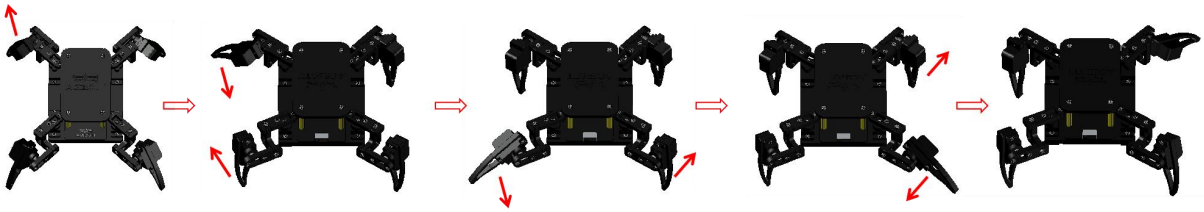
Ⅰ.Primary dance steps

Open "7.1dance1.ino" in English\Arduino\2.Arduino program\Lesson 7\7.1dance1, connect ESP8266 controller board and computer with USB cable, select the correct controller board and port, and burn the code to the ESP8266 controller board.

Before uploading, turn the toggle switch of the servo expansion board to OFF gear; After uploading, put the toggle switch of the servo extension board to the ON gear.

```
//Create an action array for the elementary step
int array9[10][9] = {
{90, 90, 90, 90, 90, 90, 90, 90, 400},//Lift all four paws
{50, 90, 90, 90, 90, 90, 90, 90, 400},//Right upper paw down
{90, 90, 90, 130, 90, 90, 90, 90, 400},//Right upper paw up, right lower paw down
{90, 90, 90, 90, 90, 90, 90, 50, 400},//Right lower paw up , left lower paw down
{90, 90, 90, 90, 130, 90, 90, 90, 400},//Left upper paw down, left lower paw up
{50, 90, 90, 90, 90, 90, 90, 90, 400},//Left upper paw up, right upper paw down
{90, 90, 90, 130, 90, 90, 90, 90, 400},//Right upper paw up, right lower paw down
{90, 90, 90, 90, 90, 90, 90, 50, 400},//Right lower paw up,left lower paw down
{90, 90, 90, 90, 130, 90, 90, 90, 400},//Left upper paw down, left lower paw up
{90, 90, 90, 90, 90, 90, 90, 90, 400},//Left upper paw up
};
//A function to set the elementary steps
void dance1() {
      for (int i = 0; i <= 9; i = i + 1) {
        servo_14.write(array9[i][0]);
        servo_12.write(array9[i][1]);
        servo_13.write(array9[i][2]);
        servo_15.write(array9[i][3]);
        servo_16.write(array9[i][4]);
        servo_5.write(array9[i][5]);
        servo_4.write(array9[i][6]);
        servo_2.write(array9[i][7]);
        delay(array9[i][8]);
      }
}
//Run the program for the elementary step
void loop() {
    dance1();
    delay(1000);
}
```

In the above elementary step, you can see that each paw of the robot in order to do the action of lifting and falling, relatively simple, next on this basis to add a little more difficulty.



Ⅱ.Intermediate dance steps

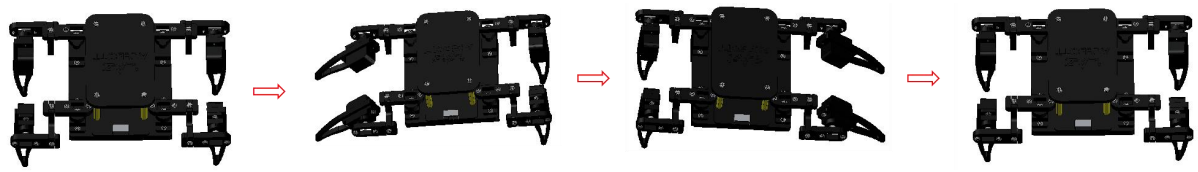
Open "7.2dance2.ino" in English\Arduino\2.Arduino program\Lesson 7\7.2dance2, connect ESP8266 controller board and computer with USB cable, select the correct controller board and port, and burn the code to the ESP8266 controller board.

Before uploading, turn the toggle switch of the servo expansion board to OFF gear; After uploading, put the toggle switch of the servo extension board to the ON gear.

```
//Create an action array for intermediate steps
int array10[9][9] = {
     {70, 45, 135, 110, 110, 135, 45, 70, 400},//The four arms are left and right together
     {115, 45, 135, 65, 110, 135, 45, 70, 400},//Right upper paw and right lower paw are raised
     {70, 45, 135, 110, 65, 135, 45, 115, 400},//Right upper and lower paws down,left upper and lower paws up
     {115, 45, 135, 65, 110, 135, 45, 70, 400},//Left upper and lower paws down,right upper and lower paws up
     {70, 45, 135, 110, 65, 135, 45, 115, 400},//Right upper and lower paws down,left upper and lower paws up
     {115, 45, 135, 65, 110, 135, 45, 70, 400},//Left upper and lower paws down,right upper and lower paws up
     {70, 45, 135, 110, 65, 135, 45, 115, 400},//Right upper and lower paws down,left upper and lower paws up
     {115, 45, 135, 65, 110, 135, 45, 70, 400},//Left upper and lower paws down,right upper and lower paws up
     {75, 45, 135, 105, 110, 135, 45, 70, 400},//Right upper paw and right lower paw down
};

//A function to set intermediate steps
void dance2() {
       for (int i = 0; i <= 8; i = i + 1) {
         servo_14.write(array10[i][0]);
         servo_12.write(array10[i][1]);
         servo_13.write(array10[i][2]);
         servo_15.write(array10[i][3]);
         servo_16.write(array10[i][4]);
         servo_5.write(array10[i][5]);
         servo_4.write(array10[i][6]);
         servo_2.write(array10[i][7]);
         delay(array10[i][8]);
       }
}
//Run the intermediate step program
void loop() {
     dance2();
     delay(1000);
}
```

How about intermediate steps? You can see the effect of the robot swinging from side to side continuously, the movement has been enriched, and then there are advanced dance steps.



Ⅲ.Advanced dance steps

Open "7.3_Dance3.ino" in English\Arduino\2.Arduino program\Lesson 7\7.3dance3, connect ESP8266 controller board and computer with USB cable, select the correct controller board and port, and burn the code to the ESP8266 controller board.

Before uploading, turn the toggle switch of the servo expansion board to OFF gear; After uploading, put the toggle switch of the servo extension board to the ON gear.

```
//Create an action array for advanced steps
int array11[10][9] = {
    {70, 45, 45, 110, 110, 135, 135, 70, 400},//All four arms back
    {90, 45, 45, 60, 90, 135, 135, 70, 400},//Right upper paw, left upper paw, right lower paw up
    {70, 45, 45, 110, 110, 135, 135, 70, 400},//Right upper paw, left upper paw and right lower paw down
    {90, 45, 45, 110, 90, 135, 135, 120, 400},//Right upper paw, left upper paw, left lower paw up
    {70, 45, 45, 110, 110, 135, 135, 70, 400},//Right upper paw, left upper paw and left lower paw down
    {90, 45, 45, 60, 90, 135, 135, 70, 400},//Right upper paw, left upper paw, right lower paw up
    {70, 45, 45, 110, 110, 135, 135, 70, 400},//Right upper paw, left upper paw and right lower paw down
    {90, 45, 45, 110, 90, 135, 135, 120, 400},//Right upper paw, left upper paw, left lower paw up
    {70, 45, 45, 110, 110, 135, 135, 70, 400},//Right upper paw, left upper paw and left lower paw down
    {70, 90, 90, 110, 110, 90, 90, 70, 400},//Standby mode
};

//A function to set advanced steps
void dance3() {
     for (int i = 0; i <= 9; i = i + 1) {
       servo_14.write(array11[i][0]);
       servo_12.write(array11[i][1]);
       servo_13.write(array11[i][2]);
       servo_15.write(array11[i][3]);
       servo_16.write(array11[i][4]);
       servo_5.write(array11[i][5]);
       servo_4.write(array11[i][6]);
       servo_2.write(array11[i][7]);
       delay(array11[i][8]);
     }
}
//Run the advanced step program
void loop() {
    dance3();
    delay(1000);
}
```

Advanced dance step can see the robot in a push-up position, the left lower arm and the right lower arm alternately fall and rise, show "hands and one leg" to support the body to rise and fall action, very cool.

Ⅳ.Extending tasks

Learning the above three kinds of dance steps, is not found that the robot is quite interesting, through different arrays can let the robot make a variety of flexible movements, I believe you have been eager to try. The next step is to improve your mastery by expanding your tasks.

(1)Task Description:

Application of serial communication to control the robot dance, such as serial input 1, the robot dance primary dance steps; Enter 2, jump intermediate dance steps; Enter 3 jump advanced dance steps.

(2)Refer to the program:

Open "7.4_Expand.ino" in English\Arduino\2.Arduino program\Lesson 7\7.4expand, connect ESP8266 development board and computer with USB cable, select the correct development board and port, and burn the code to ESP8266 development board.

Before uploading, turn the toggle switch of the servo expansion board to OFF gear; After uploading, put the toggle switch of the servo extension board to the ON gear.

①Set baud rate.

```
Serial.begin(115200); //The baud rate of esp8266 serial communication is 115200
```

②Create a new variable to receive serial port data.

```
String item; //Because serial data is in string format, the variables are also in string format
item = ""; //Initialize the variable as an empty string
```

③Program logic.

```
//Determine whether there is data in the serial port
if (Serial.available()) {
    item = Serial.readString(); //The variable is assigned to the data read by the serial port
    Serial.println(item); //The serial port prints out the input data

//According to the input data, the condition is judged and the corresponding function is executed
    if (item == "1") { //When 1 is input to the serial port, dance1 is executed
    dance1();
      }
    if (item == "2") { //When 2 is input to the serial port, dance2 is executed
       dance2();
      }
    if (item == "3") { //When 3 is input to the serial port, dance3 is executed
       dance3();
      }
}
```

④Attention that the serial port monitor has no terminator and the baud rate is set to 115200.

Lesson 8 Robot WiFi Control

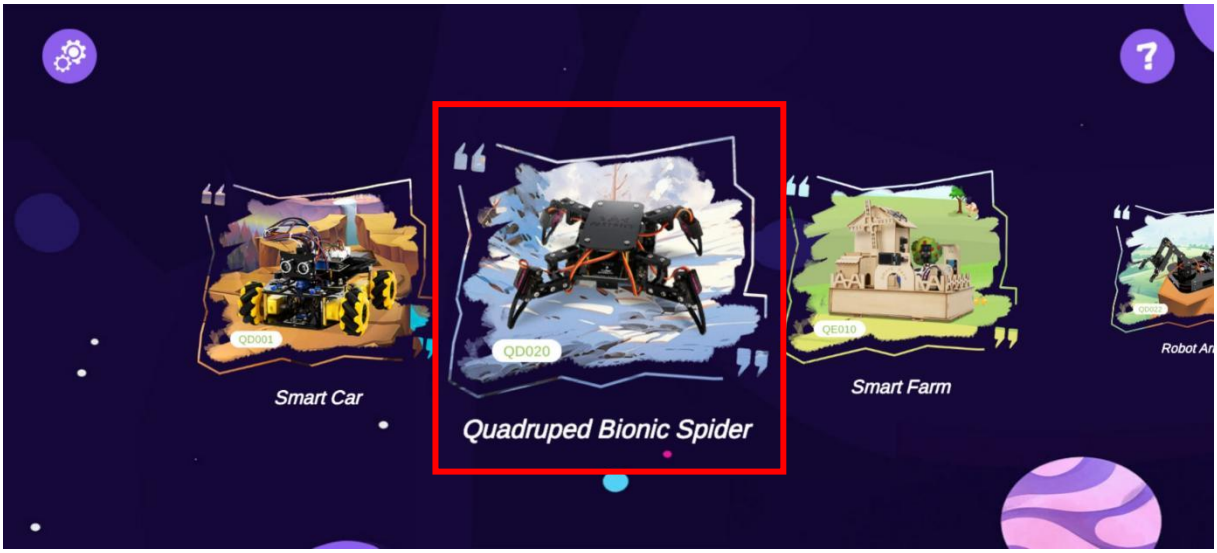
Ⅰ.APP download

(1)For an IOS device, search for ACEBOTT in the APP Store and download it. For Android phones, search the Google Play Store for ACEBOTT and download it. The icon is shown below.

Attention: 1. This tutorial is applicable to ACEBOTT APP version 2.0 and above. You can click the settings button in the upper left corner of the APP to view the software version number. Please make sure that the software version you are using meets the requirements; 2. If you need to update the ACEBOTT software version, you can refer to the method prompted in this tutorial to download the latest APP version.

(2)Open the APP and enter the screen interface.

(3)Enter the selection interface and select the quadruped robot.



（4）Enter the robot control interface (now can not be directly controlled, need to burn the program).

Ⅱ.Program download

1.Robot WiFi control program

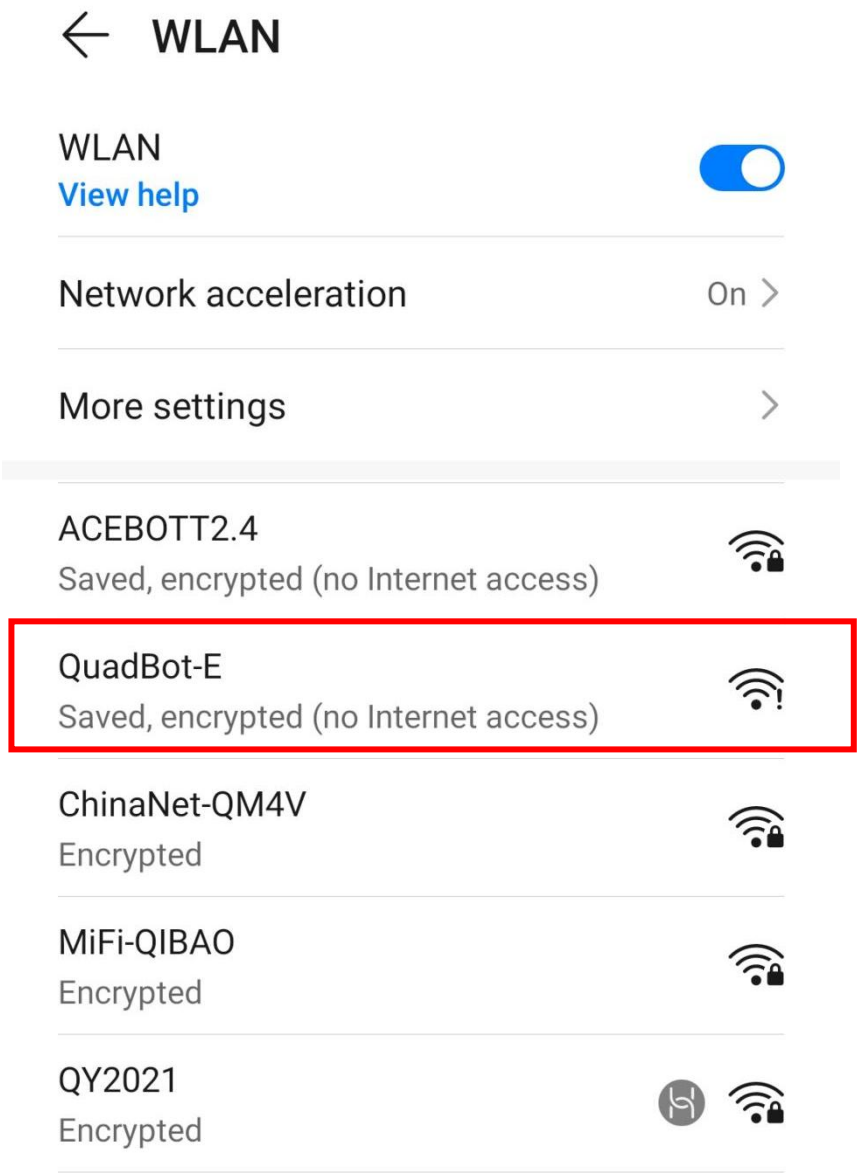
At present, it is not possible to control the robot directly with the APP, and the control program of WiFi needs to be burned to the robot to control it.

Open "8.1app_control.ino" in English\Arduino\2.Arduino program\Lesson 8\ 8.1app_control", connect ESP8266 controller board and computer with USB cable, Select the correct controller board and port and burn the code to the ESP8266 controller board.

Before uploading, turn the toggle switch of the servo expansion board to OFF gear; After uploading, put the toggle switch of the servo extension board to the ON gear.

2.Connect to WiFi

Mobile phone wireless network scan WIFI (turn off GPRS and other shared networks, make sure WIFI is the only network used) (specific operation in "Settings" → "WLAN" of the mobile phone), connect to the wifi hotspot named "QuadBot-E", the password is 12345678, as shown below.



Attention:

The name and password of the hotspot have been defined in the

program, but the user can customize and modify it. When we have multiple quadruped robots, we can distinguish each quadruped robot by different WiFi names.

```
const char* ssid = "QuadBot-E";//WiFi name
const char* password = "12345678";//WiFi password
```

3.Using APP controls

After connecting the WiFi, click the connection icon in the upper right corner of the APP to complete the connection.

Note: If you need to watch the APP operation video, please click the link below.

https://youtu.be/rzBv5HevS2M

After completing the aforementioned operations, return to the interface shown below, and then you can control the robot according to the button prompts.

The left panel is for the basic motion control of the robot: forward, backward, left move, right move, turn left, turn right.

The right panel is for the robot's action group control: standby, lying, sleep, greet, push-up, fighting, dancing, swing, and handsome.

In the upper right corner of the APP operation interface, there is a gyroscope control provided. After clicking this button, you can control the movement of the robot through the smartphone's gyroscope. If the phone does not have a built-in gyroscope, this feature can be ignored.


Quadruped Bionic Spider
Standby
Lying
Sleep
Hi
Greet
Push-up
Fighting
Dancing
Swing
Handsome

Follow Us

Scan the QR codes to Follow Us for troubleshooting & the latest news.

We have a very large community that is very helpful for troubleshooting and we also have a support team at the ready to answer any questions.

ACEBOTT FB Group QR Code

YouTube QR Code