

Table of Contents

1. Introduction to arduino eArm	3
1.1 Product List	3
1.2 Parameters of the eArm:.....	5
1.3 Introduction to ESP32 Control Board:.....	6
2. Assemble the eArm	7
Before You Begin: Important Notes	7
Step 1 Assemble the Robot Base	8
Step 2 Assemble Servo A on Robot Base	9
Step 3 Assemble the Turntable	10
Step 4 Assemble the Lower arm	11
Step 5 Fix the lower arm.....	12
Step 6 Assemble the Middle Arm	13
Step 7 Assemble the Upper Arm.....	14
Step 8 Assemble the Servo B on lower arm.....	15
Step 9 Assemble the Servo C on Middle Arm	16
Step 10 Assemble the Servo D on Upper Arm	17
Step 11 Assemble the Left Clip Plate of The Clamp	18
Step 12 Assemble the Right Clip Plate of The Clamp	18
Step 13 Install the Left Clip Plate of The Clamp	19
Step 14 Install the Right Clip Plate of the Clamp	20
Step 15 Assemble the Clamp and Upper Arm	21
Step 16 Assemble the linkage between Servo B and Servo C	22
Step 17 Assemble the Mounting Structure for Servo A	23
Step 18 Assemble the Mounting Structure for Servo D	24
Step 19 Assemble the Joystick	25
Step 20 Install the Esp32 Board	26
Step 21 Connect the Joystick	27
Step 22 Connect the Servos to the ESP32 Board	28
Step 23 Initialize the Servo Angle (important!)	29
Step 24 Fix the Robot Arm to Servo A of the Base	30
Step 25 Assemble the Middle Arm	31
Step 26 Fix the Clip Plates to Servo D	32
Step 27 Assembly completed.....	33
3. Using the Robot Arm	34
3.1 Degrees of Freedom	34
3.2 Control the eArm with a Joystick	35
3.3 Control the eArm with Web App	36
4. Programming Learning	38
Before You Begin: Important Notes	38
4.1 Install Arduino IDE	38
4.2 Install ESP32 Development Environment for Arduino IDE	40

4.3 Install the Library File	43
4.4 Install CH340 USB driver	44
Install the Driver on Windows system	45
Install the Driver on Linux system	45
Install the Driver on MAC system	46
4.5 Code Examples	47
4.5.1 Onboard buzzer	47
4.5.2 To Drive a Servo	50
4.5.3 Read the Value of the Joystick	52
4.5.4 Read the Value of the Web APP	53
4.5.5 Using the Robot Arm	55
5. Factory Reset Function	56
5.1 Firmware	56
5.2 Burning Tool	58
5.3 How to Burn Firmware	58
6. QA	59
6.1 Unable to recognize USB serial ports	59
6.2 Robotic arm doesn't work	60
6.3 Other problems	60
7. Contact Us	60

1

Introduction to arduino eArm

The eArm robot is an ESP32-powered robotic arm, it offers a hands-on way to explore robotics, coding, and electronics in an engaging platform.

Features:

① Quick Assembly & Ready-to-Use

- The robotic arm is designed for effortless assembly—you can build it and start experimenting within 1-2 hours.

② Pre-Programmed for Instant Operation

- The control board comes pre-flashed with firmware, eliminating the need for initial code uploads. Simply power it on and begin using it right away.

③ Integrated Power Management

- Features an onboard battery holder with a power indicator. The robot can be charged directly via USB, ensuring uninterrupted operation.

④ Precision Control with Joystick

- The included ergonomic joystick delivers stable control, offering intuitive and responsive manipulation of the robotic arm.

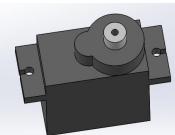
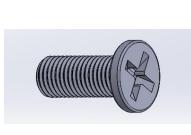
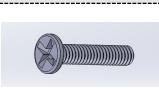
⑤ Web-Based Control

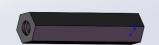
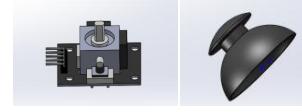
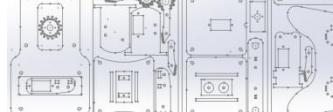
- Supports cross-platform control via a dedicated Web APP, accessible from any device with a browser—no additional installations required.

⑥ Expandable Learning with Arduino

- Includes step-by-step Arduino programming tutorials, helping users advance from basic operations to custom robotics development.

1.1 Product List

Control Board 1PCS	Servo 4PCS	M4x10mm Screw 20PCS
		
M3x14mm Screw 2PCS	M3x10mm Screw 18PCS	M2x10mm Screw 8PCS
		

M1.4x5mm Screw 6PCS	M4 Nut 4PCS	M3 Self-locking Nut 8PCS
		
M2 Nut 8PCS	M3x7mm Nylon Column 4PCS	M3x28mm Nylon Column 4PCS
		
M3x40mm Nylon Column 2PCS	M4x20mm Nylon Column 8PCS	M3x6mm Nylon Screw 8PCS
		
Flange Bearing 4PCS	Turntable 1PCS	5P Dupont Wire 2PCS
		
Joystick 2PCS	USB Cable 1PCS	A Set of Acrylic Sheet
		
M3 Screwdriver 1PCS	M1.5 Screwdriver 1PCS	Wrench 1PCS
		



Note: You need to buy a 18650 lithium battery yourself!

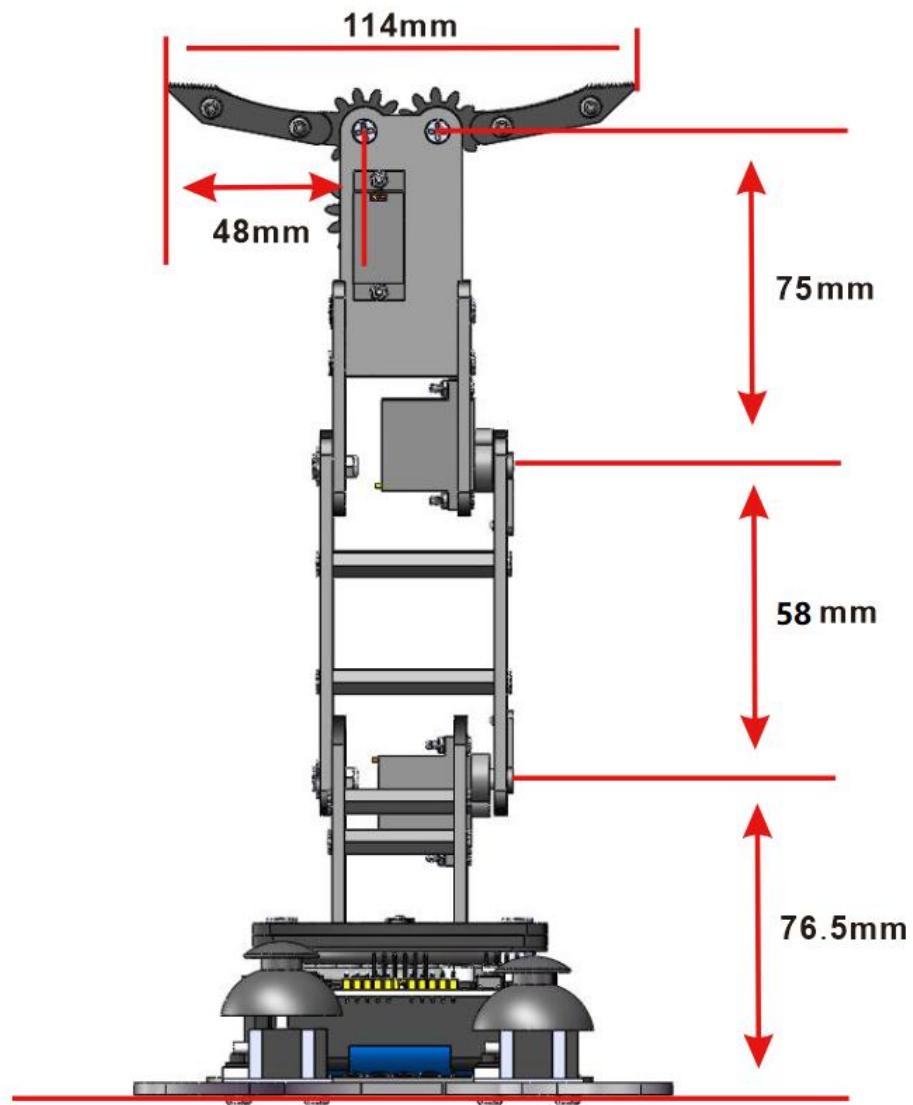
Parameters of 18650 lithium battery



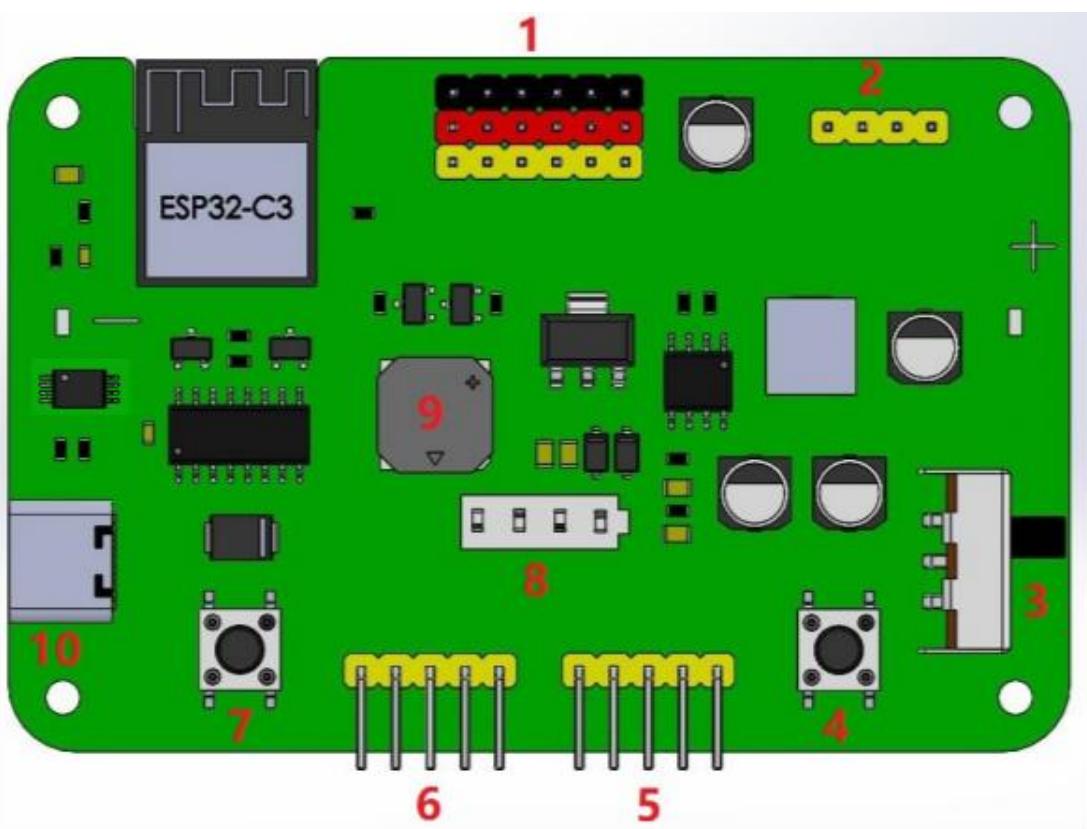
- Model: 18650 lithium battery
- Top type: Both button top and flat top are acceptable
- Capacity: >2000mAh
- Maximum charging voltage: 4.2V
- Nominal voltage: 3.7V
- End-off voltage: 2.75V
- Minimum charging current: >2A
- Minimum discharge current: >4A

1.2 Parameters of the eArm:

Control Board	eArm-ESP32-C3-MINI-1
Programming language	Arduino
Charge/discharge	5V/2A
Servo type	MG90S 180°
Remote control method	Joystick or Web APP
Assembly time required	1-2 hours
Product size	15X9.5X26MM
Packaging size	18.5X12.5X4.3MM
Applicable age	experienced players of any age or beginner aged 12+
Battery required	a 18650 battery(Not included)



1.3 Introduction to ESP32 Control Board:

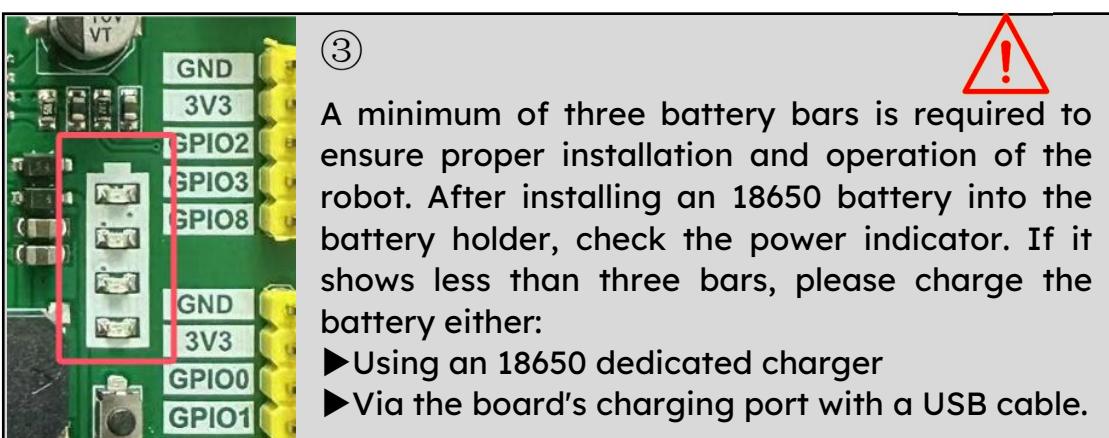
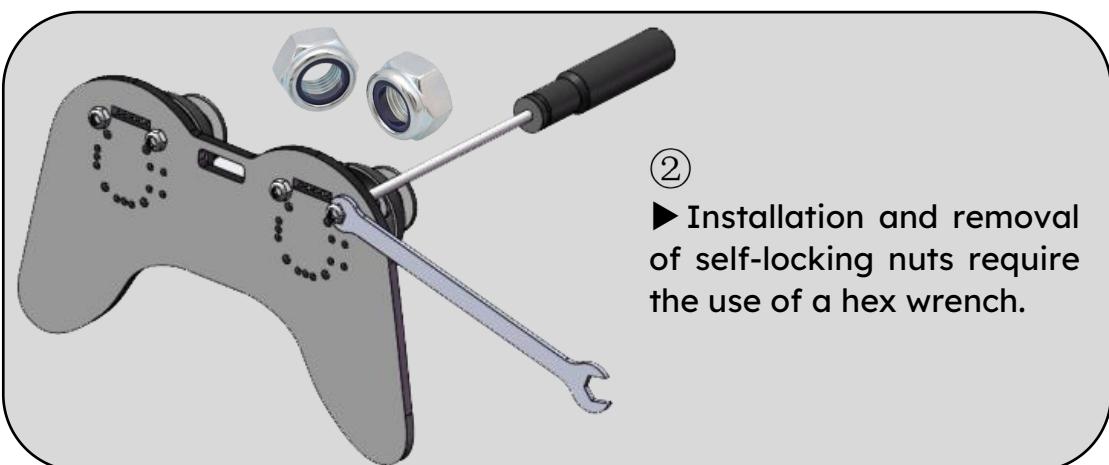


- 1: IO port, 5V/2A driving capability.
- 2: IO port and serial port expansion port.
- 3: Power switch.
- 4: Battery level display button(When the power switch is off, pressing it briefly can display the battery level).
- 5: IO port, 3.3V/500mA driving capability.
- 6: IO port, 3.3V/500mA driving capability.
- 7: Reset button.
- 8: Battery level indicator LED.
- 9: Buzzer.
- 10: USB Type-c Port for charging or uploading code

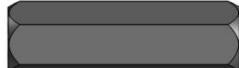
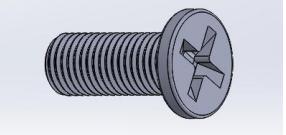
2

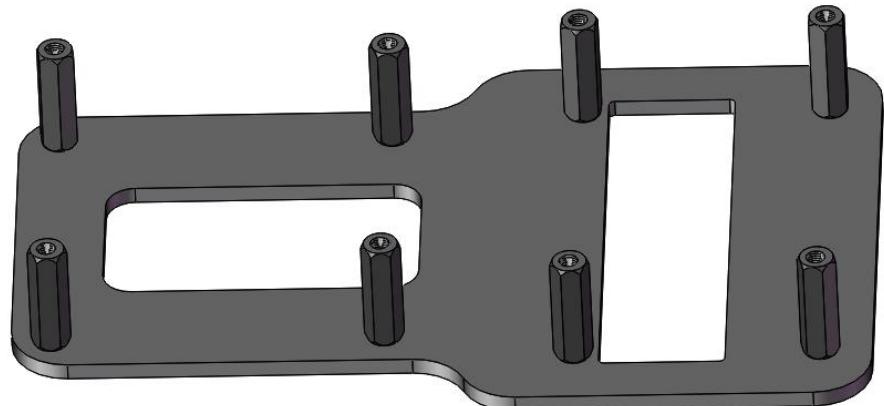
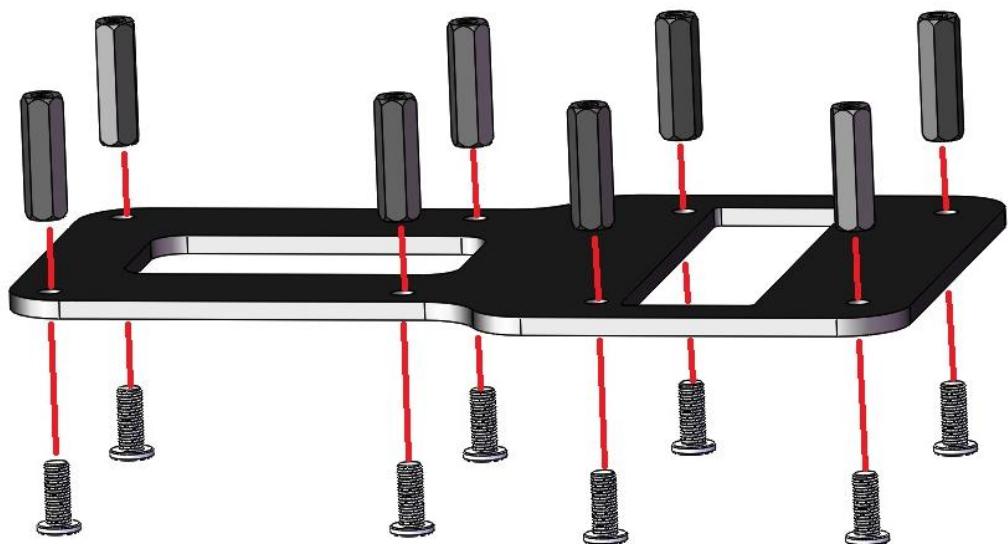
Assemble the eArm

Before You Begin: Important Notes

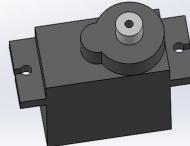
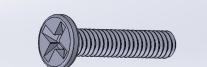
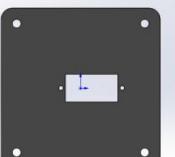


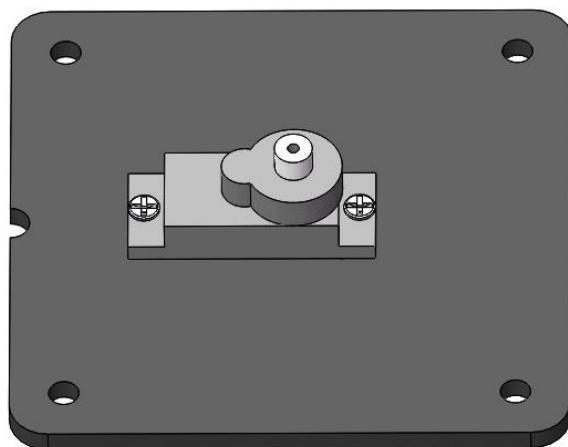
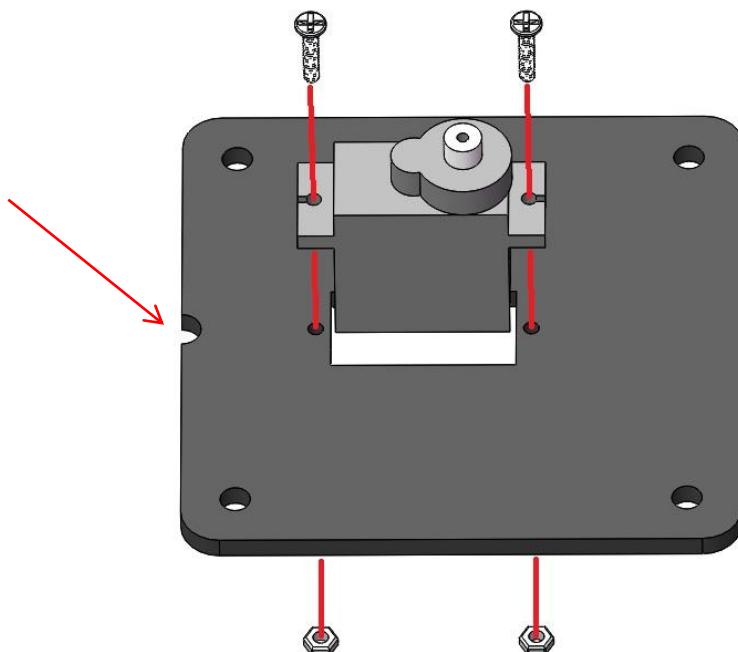
Step 1 Assemble the Robot Base

Acrylic Sheet 1PCS	M4X20MM Nylon Column 8PCS	M4X10MM Screw 8PCS
		

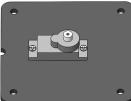
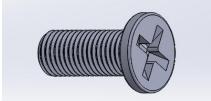


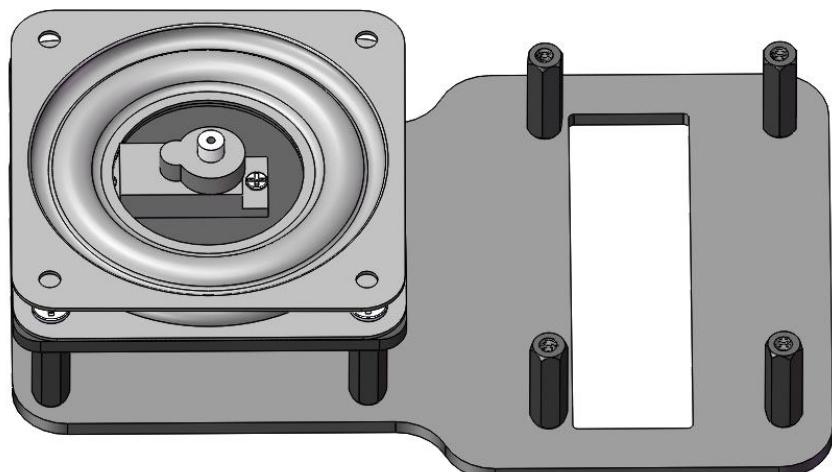
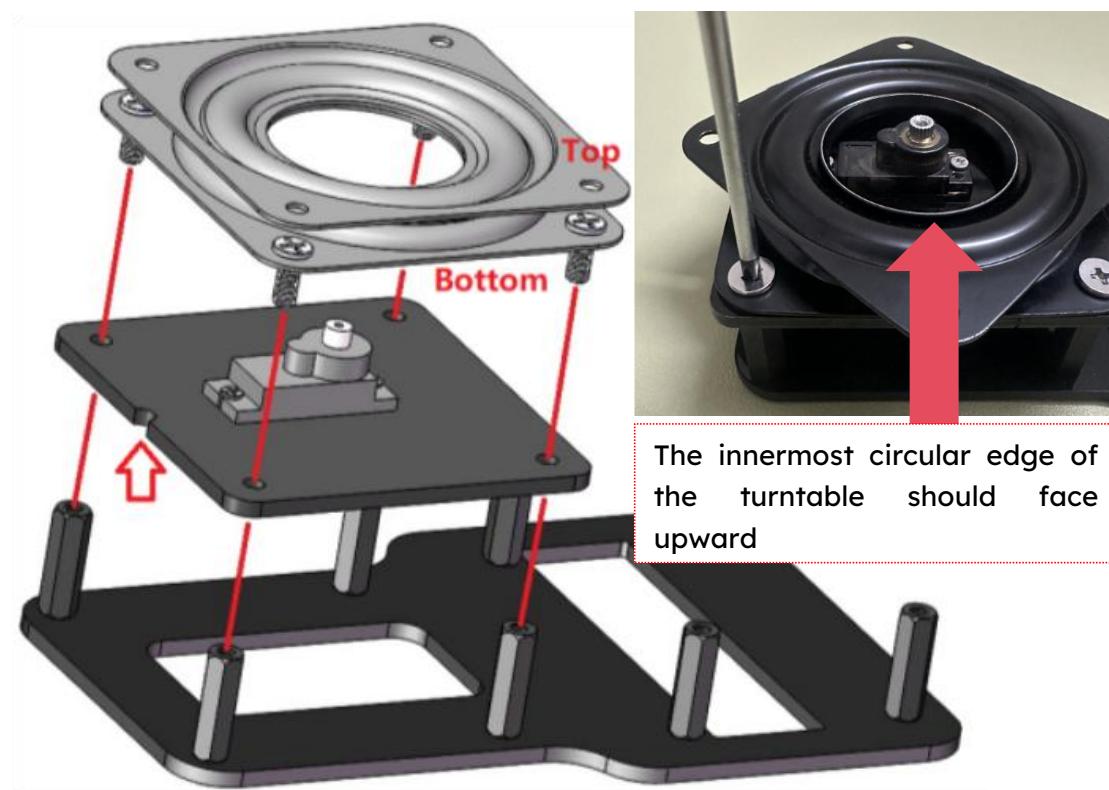
Step 2 Assemble Servo A on Robot Base

Servo 1PCS	M2x10MM Screw 2PCS	M2 Nut 2PCS	Acrylic Sheet 1PCS
			



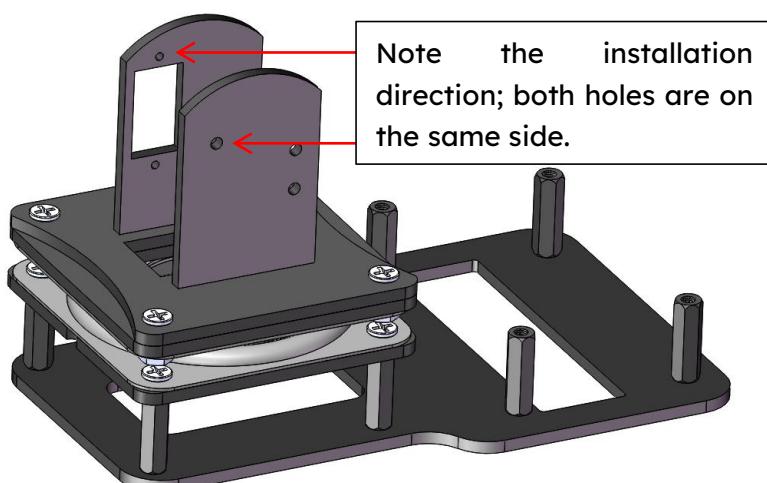
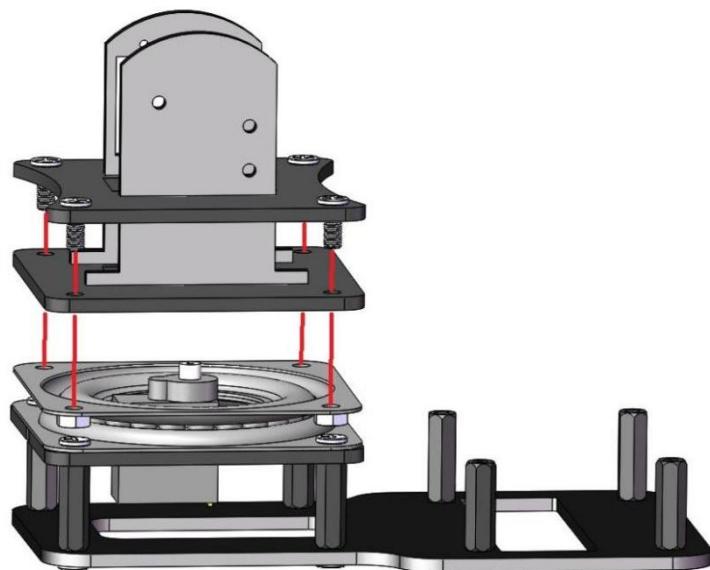
Step 3 Assemble the Turntable

Step 1 Structure	Step 2 Structure	Turntable 1PCS	M4X10MM Screw 4PCS
			

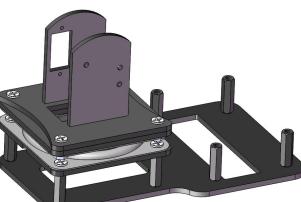
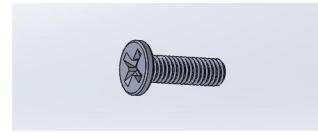


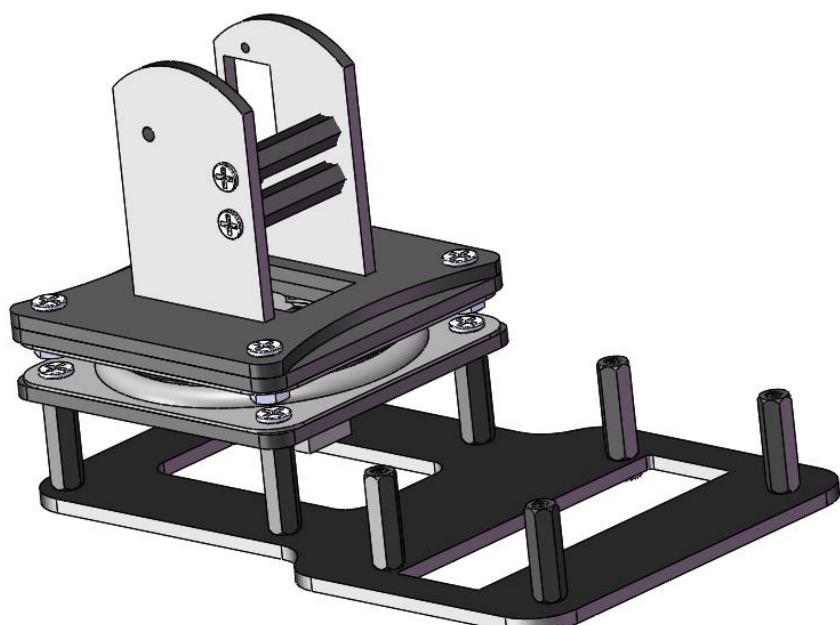
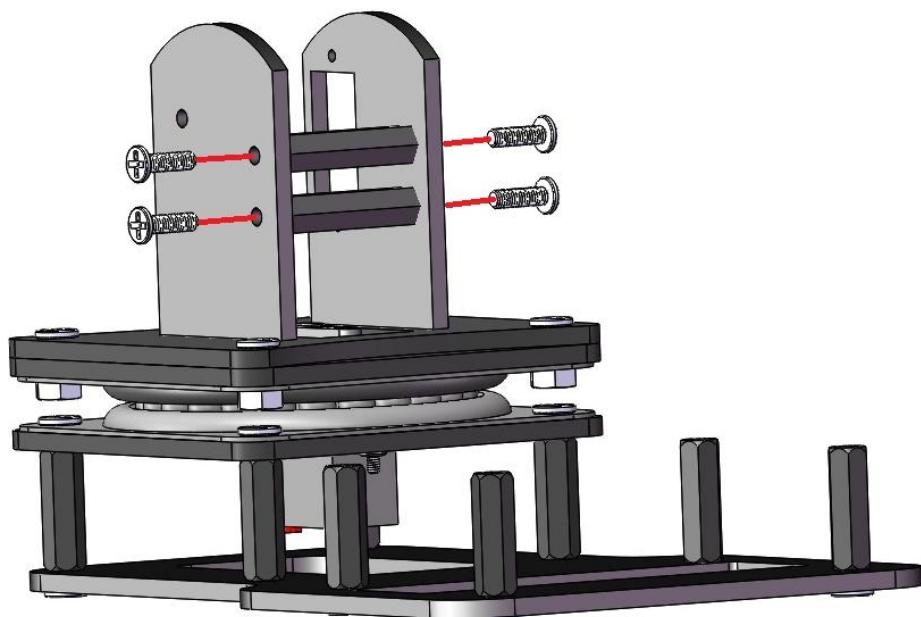
Step 4 Assemble the Lower arm

Step 3 Structure	Acrylic Sheet 1PCS	Acrylic Sheet 1PCS	
Acrylic Sheet 1PCS	Acrylic Sheet 1PCS	M4X10MM Screw 4PCS	M4 Nut 4PCS



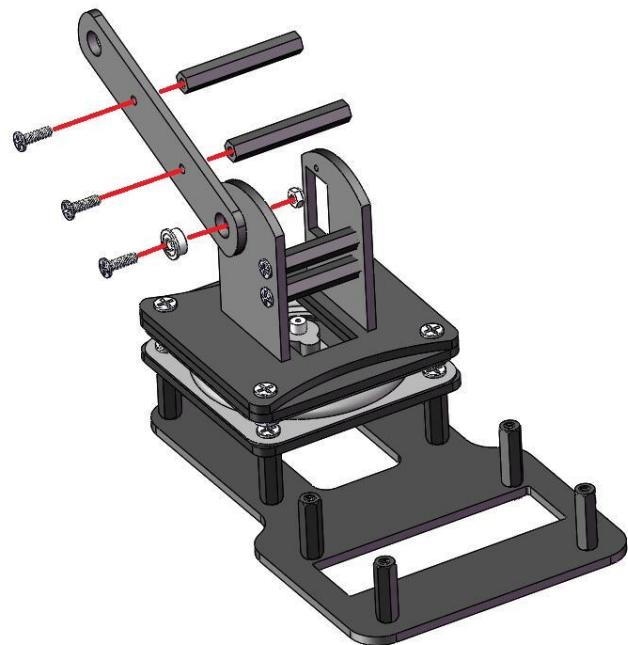
Step 5 Fix the lower arm

Step 4 Structure	M3x28MM Nylon Column 2PCS	M3X10MM Screw 4PCS
		



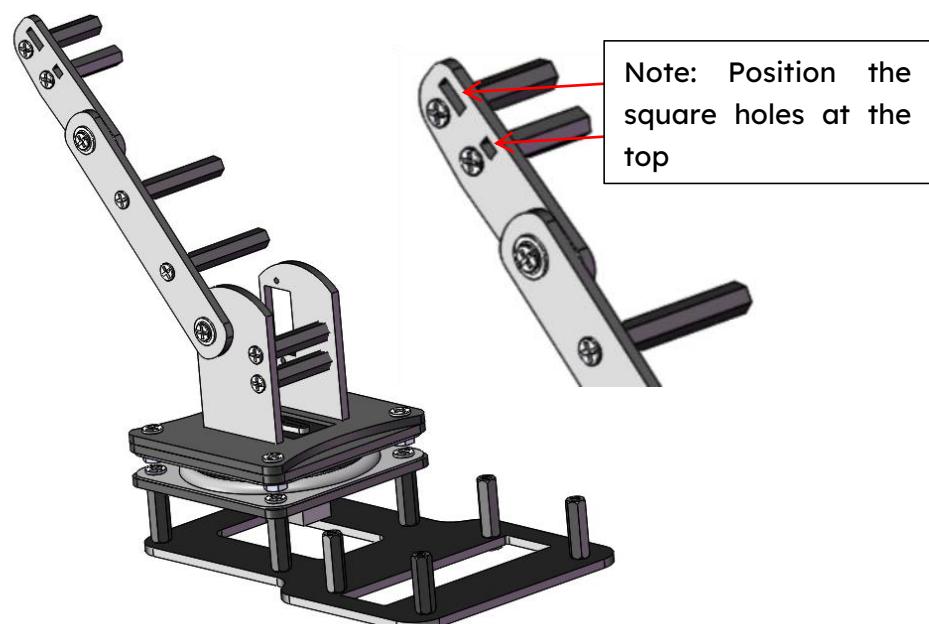
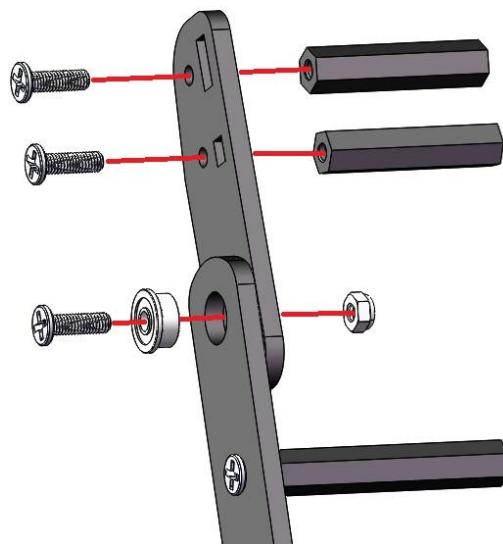
Step 6 Assemble the Middle Arm

Step 5 Structure	Acrylic Sheet 1PCS	Flange Bearing 1PCS
M3x40MM Nylon Column 2PCS	M3X10MM Screw 3PCS	M3 Self-locking Nut 1PCS

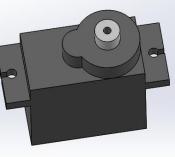


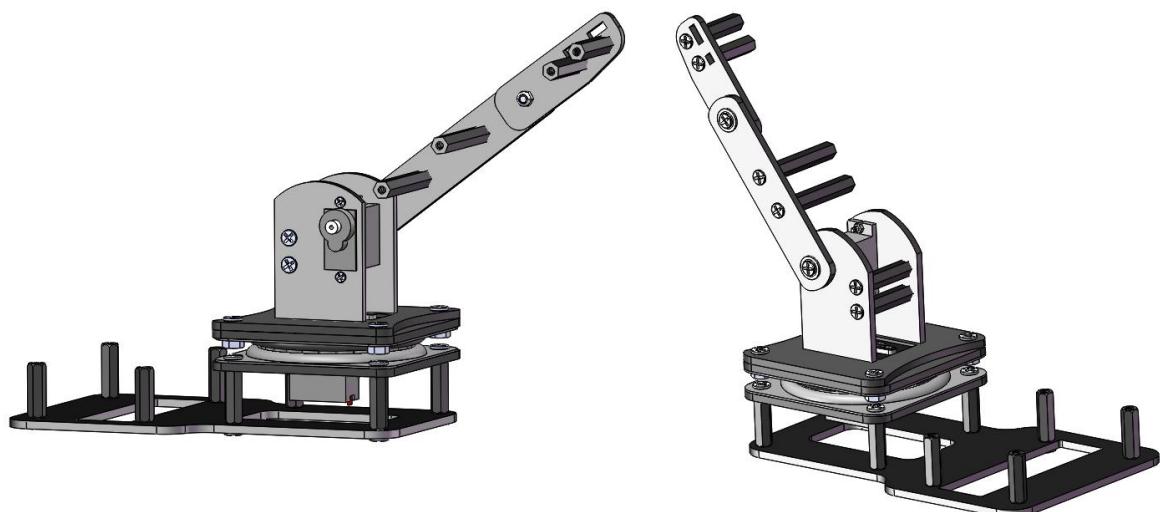
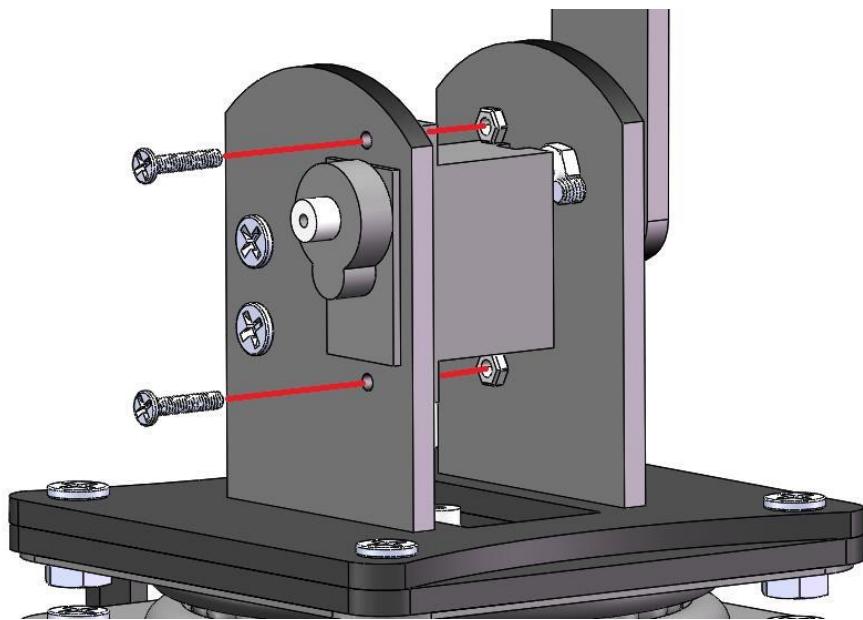
Step 7 Assemble the Upper Arm

Step 6 Structure	Acrylic Sheet 1PCS	Flange Bearing 1PCS
M3x28MM Nylon Column 2PCS	M3X10MM Screw 3PCS	M3 Self-locking Nut 1PCS

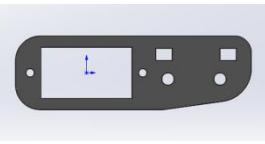
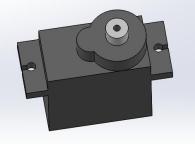
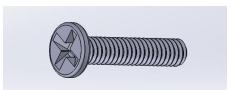


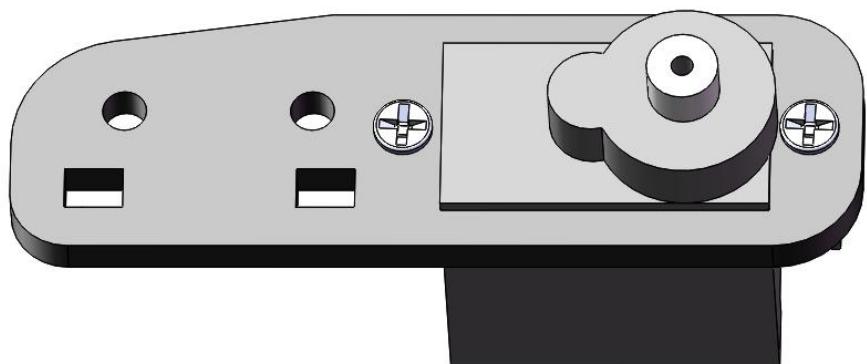
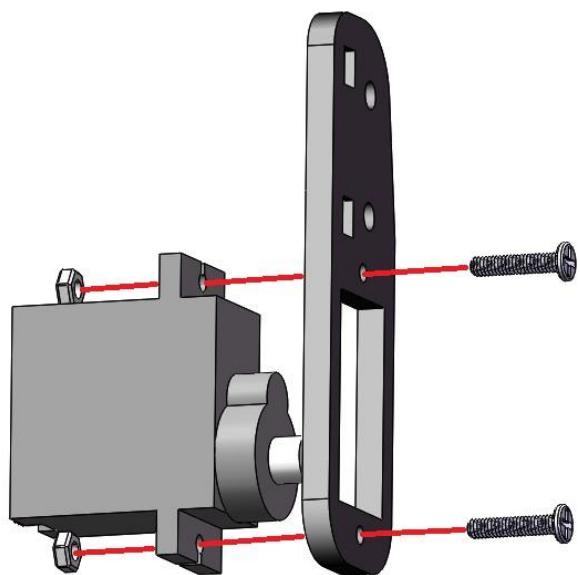
Step 8 Assemble the Servo B on lower arm

Step 7 Structure	Servo 1PCS	M2x10MM Screw 2PCS	M2 Nut 2PCS
			

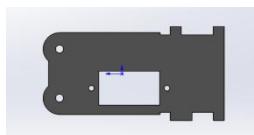
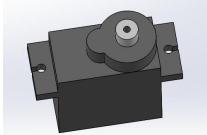
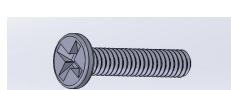


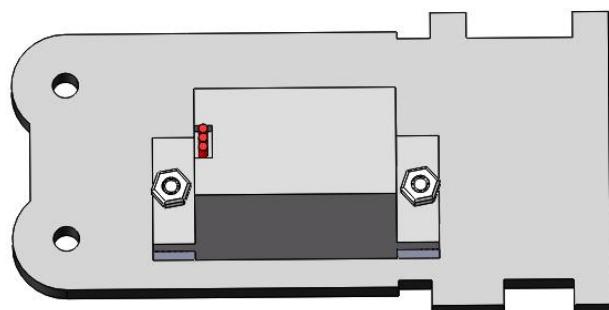
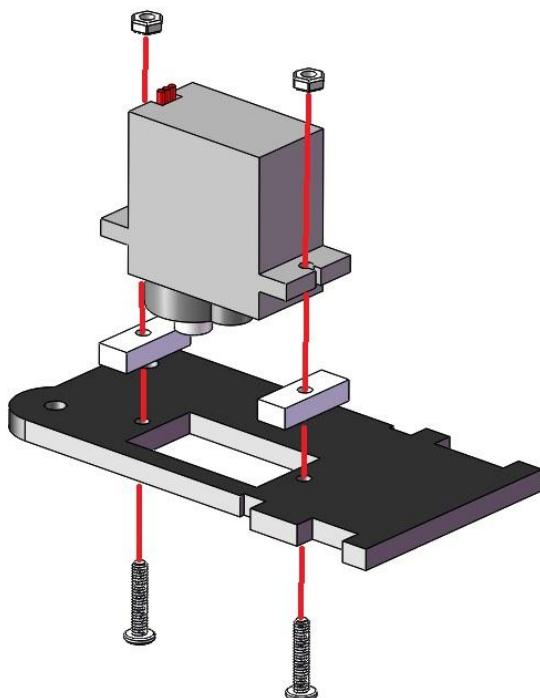
Step 9 Assemble the Servo C on Middle Arm

Acrylic Sheet 1PCS	Servo 1PCS	M2x10MM Screw 2PCS	M2 Nut 2PCS
			

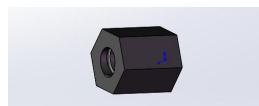


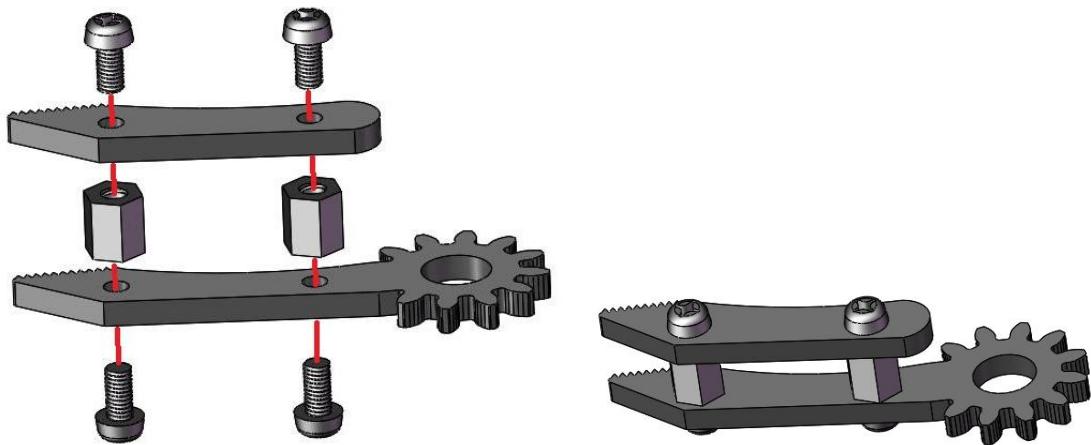
Step 10 Assemble the Servo D on Upper Arm

Acrylic Sheet 1PCS	Servo 1PCS	
		
Acrylic Sheet 2PCS	M2x10MM Screw 2PCS	M2 Nut 2PCS
		

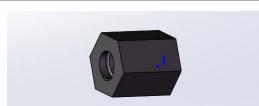


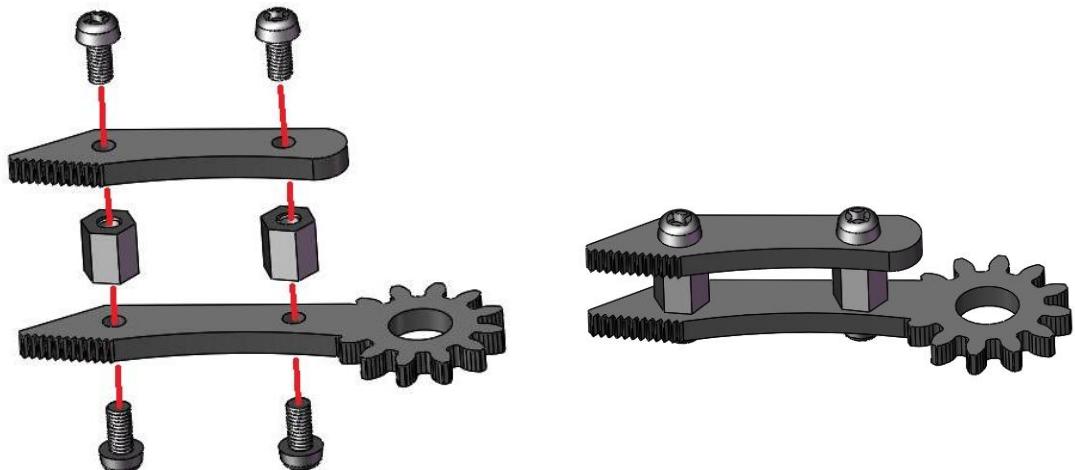
Step 11 Assemble the Left Clip Plate of The Clamp

Acrylic Sheet 1PCS	Acrylic Sheet 1PCS	M3x7MM Nylon Column 2PCS	M3x6MM Nylon Screw 4PCS
			



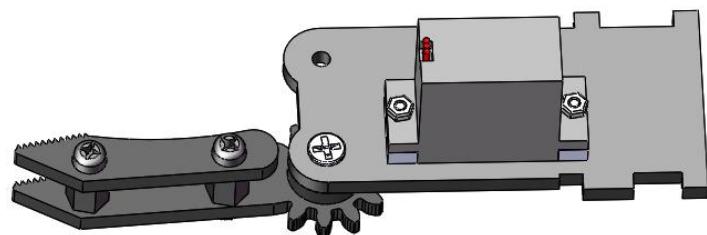
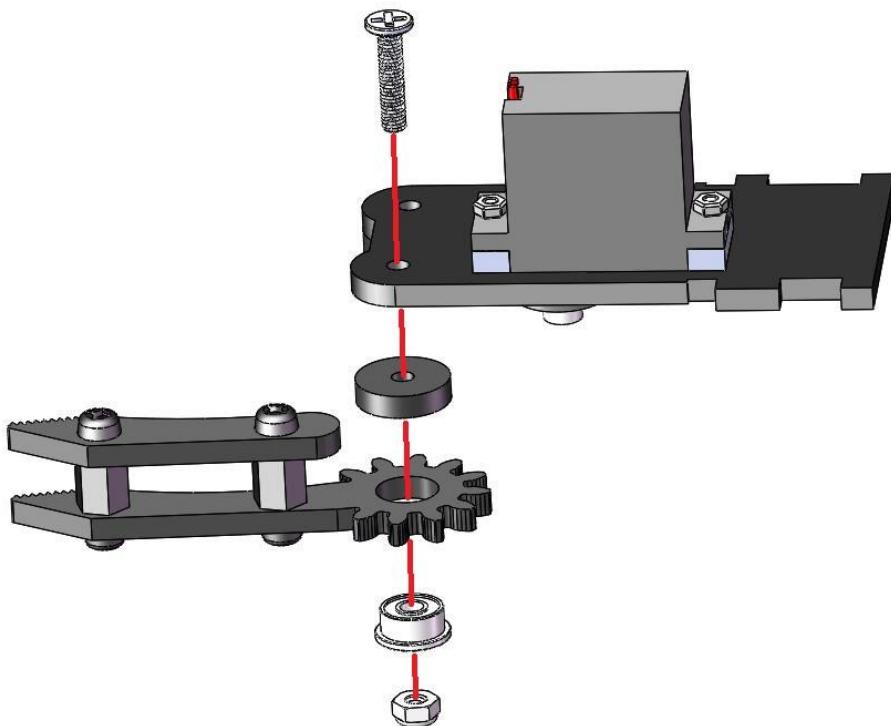
Step 12 Assemble the Right Clip Plate of The Clamp

Acrylic Sheet 1PCS	Acrylic Sheet 1PCS	M3x7MM Nylon Column 2PCS	M3x6MM Nylon Screw 4PCS
			



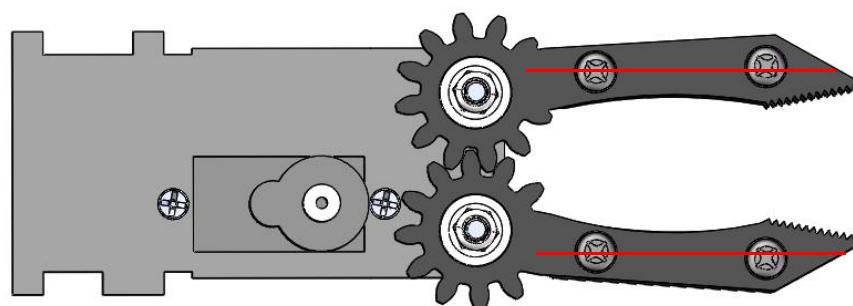
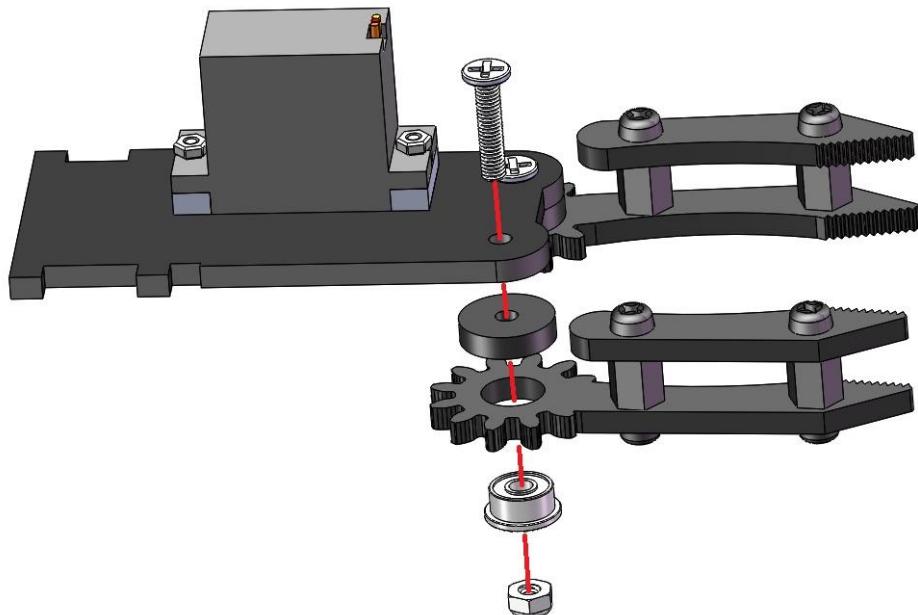
Step 13 Install the Left Clip Plate of The Clamp

Step 10 Structure	Step 11 Structure	Flange Bearing 1PCS
Acrylic 1PCS	M3X14MM Screw 1PCS	M3 Self-locking Nut 1PCS



Step 14 Install the Right Clip Plate of the Clamp

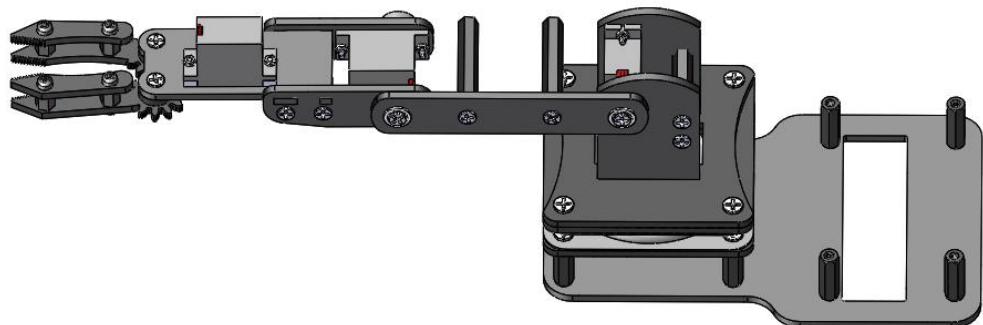
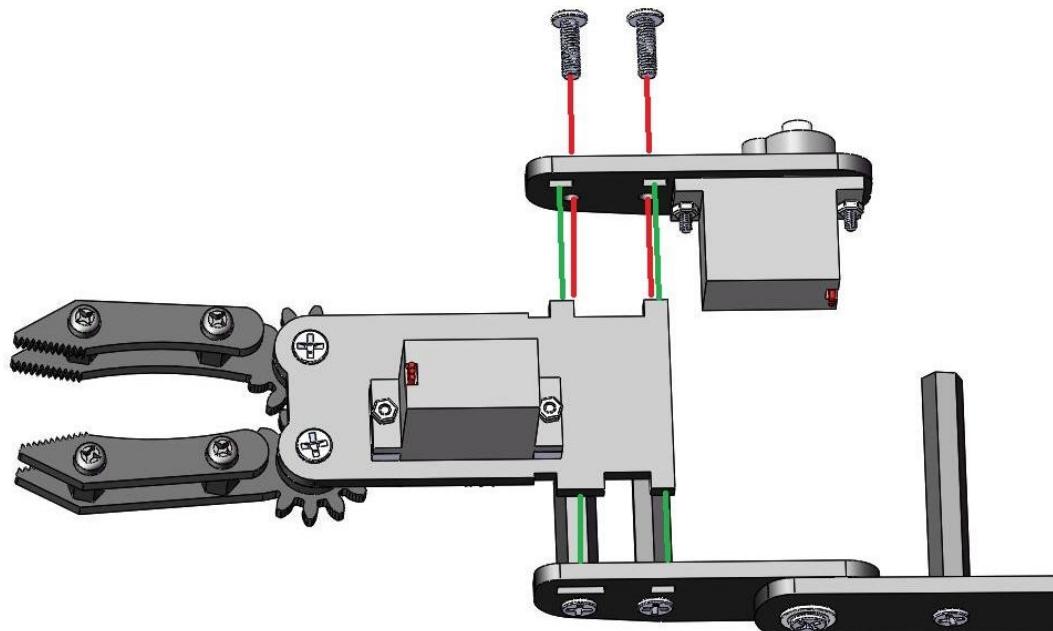
Step 13 Structure	Step 12 Structure	Flange Bearing 1PCS
		
Acrylic 1PCS	M3X14MM Screw 1PCS	M3 Self-locking Nut 1PCS
		



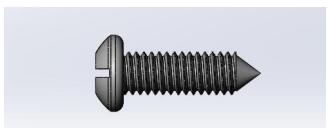
During installation, keep the screw threads horizontal

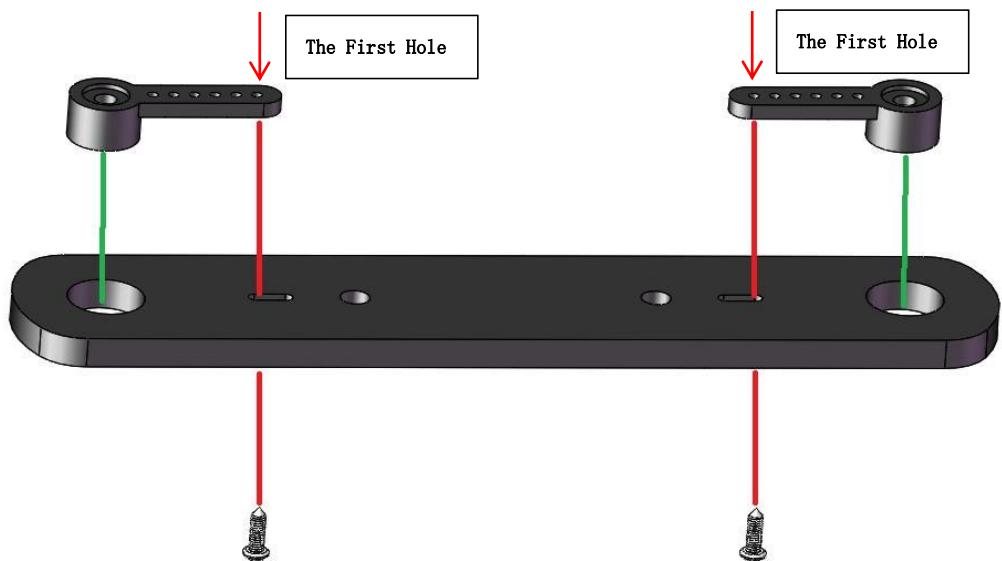
Step 15 Assemble the Clamp and Upper Arm

Step 8 Structure	Step 9 Structure
Step 14 Structure	M3X10MM Screw 2PCS

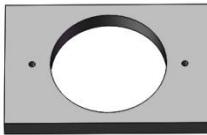


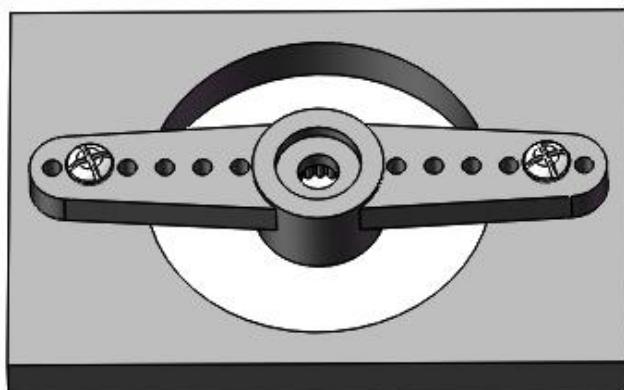
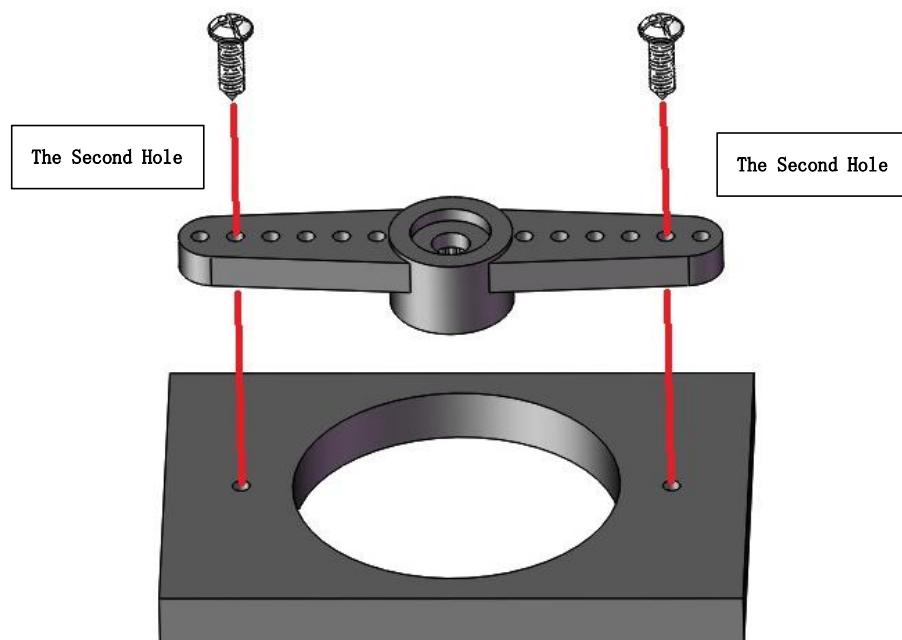
Step 16 Assemble the linkage between Servo B and Servo C

Acrylic Sheet 1PCS	Servo Arm 2PCS	M1.4X5 Self-locking Screw 2PCS
		

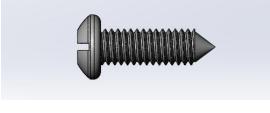


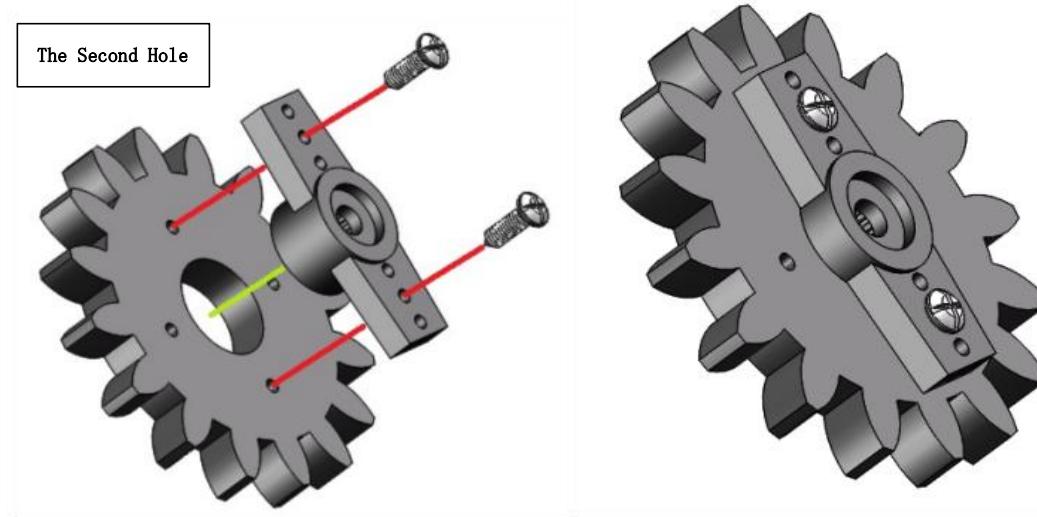
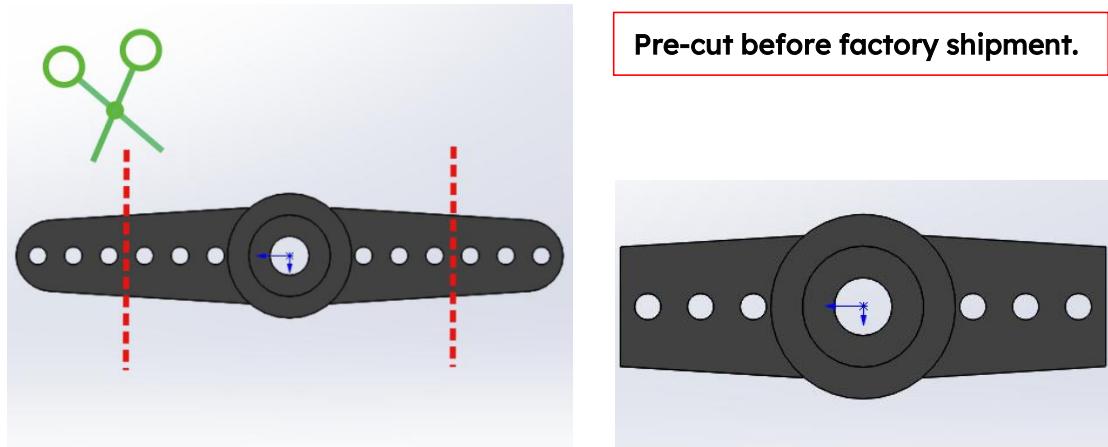
Step 17 Assemble the Mounting Structure for Servo A

Acrylic Sheet 1PCS	Servo Arm 1PCS	M1.4X5 Self-locking Screw 2PCS
		

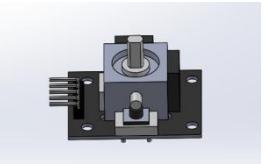


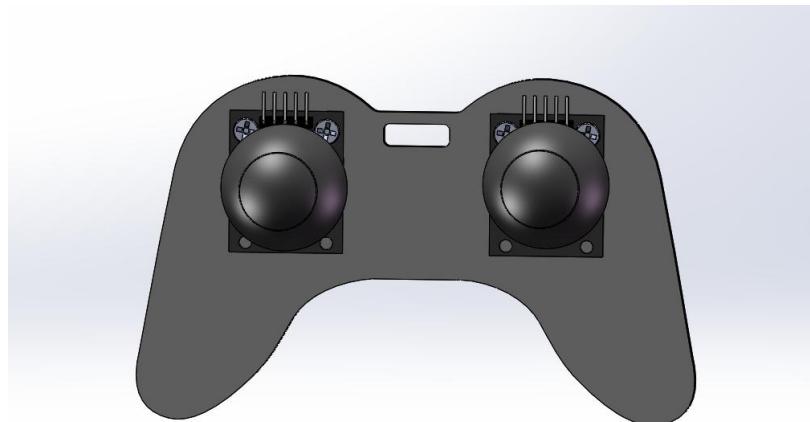
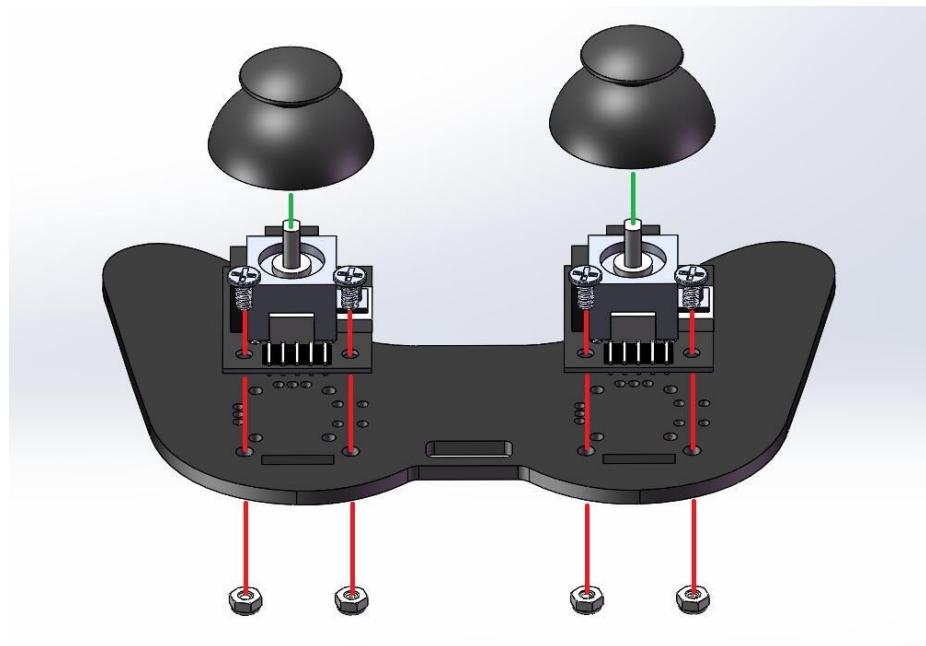
Step 18 Assemble the Mounting Structure for Servo D

Acrylic Sheet 1PCS	Servo Arm 1PCS	M1.4X5 Self-locking Screw 2PCS
		



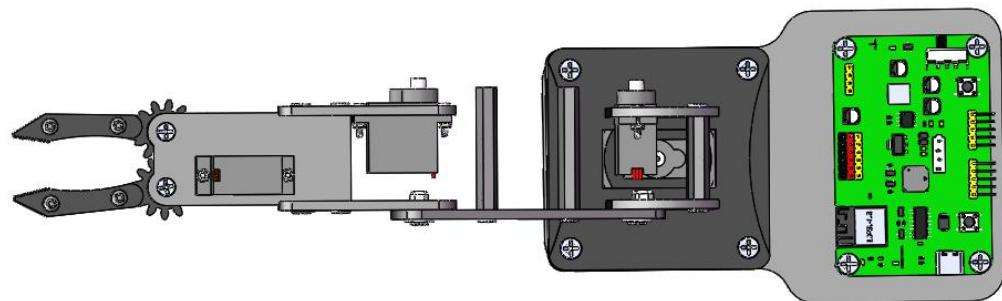
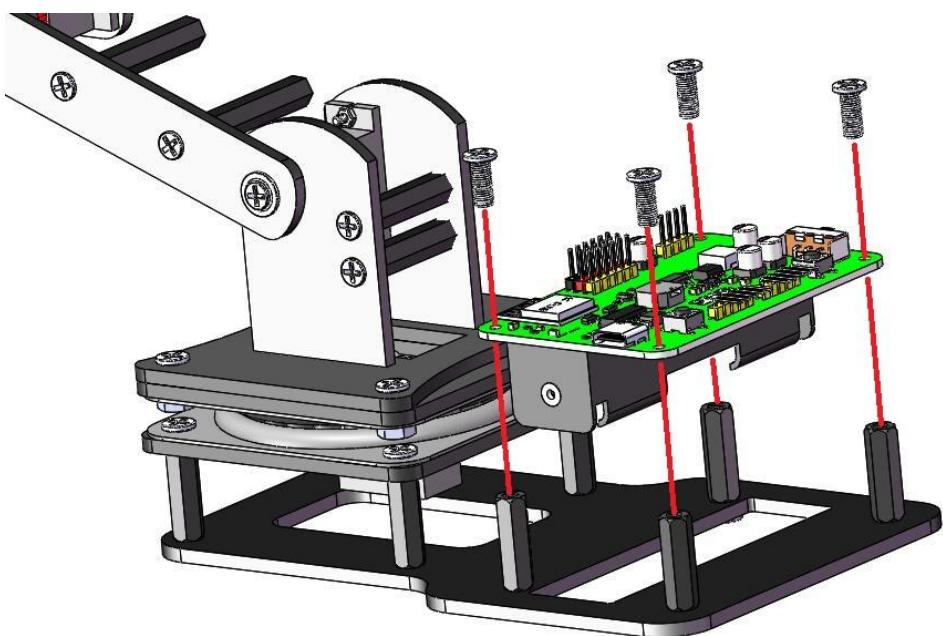
Step 19 Assemble the Joystick

Acrylic Sheet 1PCS	Joystick Module 2PCS	
		
Joystick Hat 2PCS	M3X10MM Screw 4PCS	M3 Self-locking Nut 4PCS
		



Step 20 Install the Esp32 Board

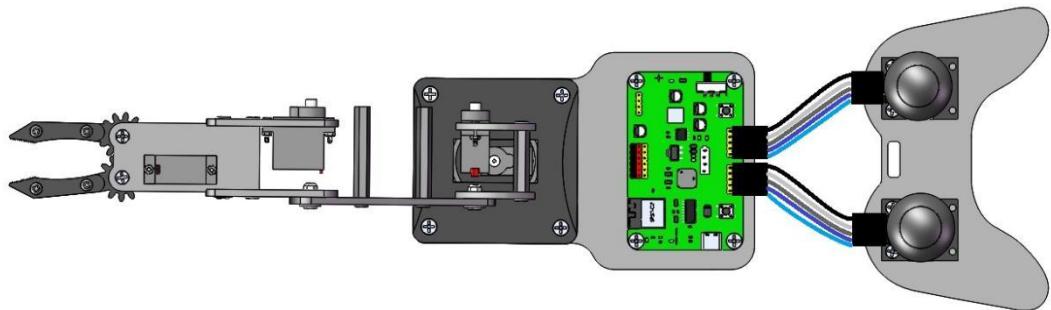
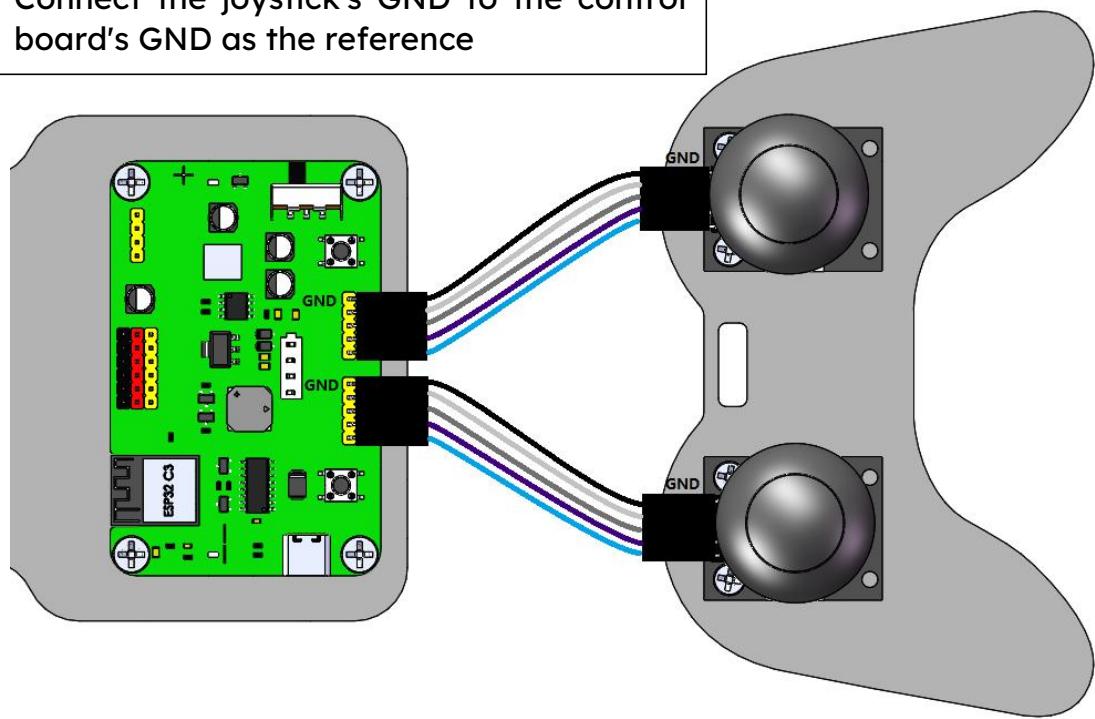
Step 15 Structure	ESP32 Control Board 1PCS	M4X10MM Screw 4PCS



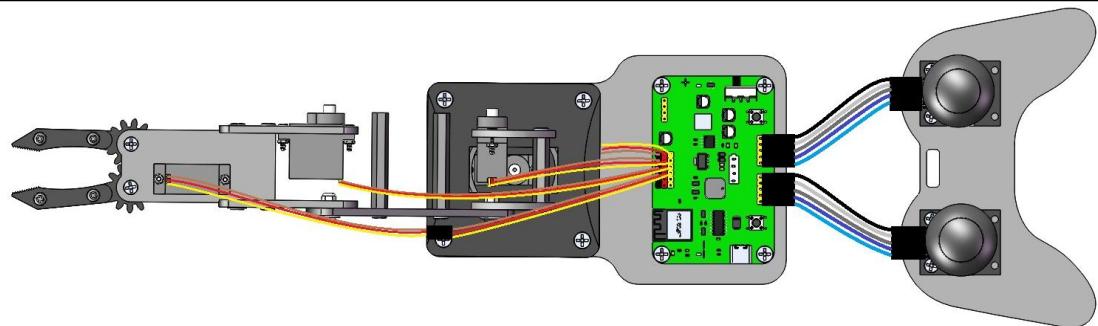
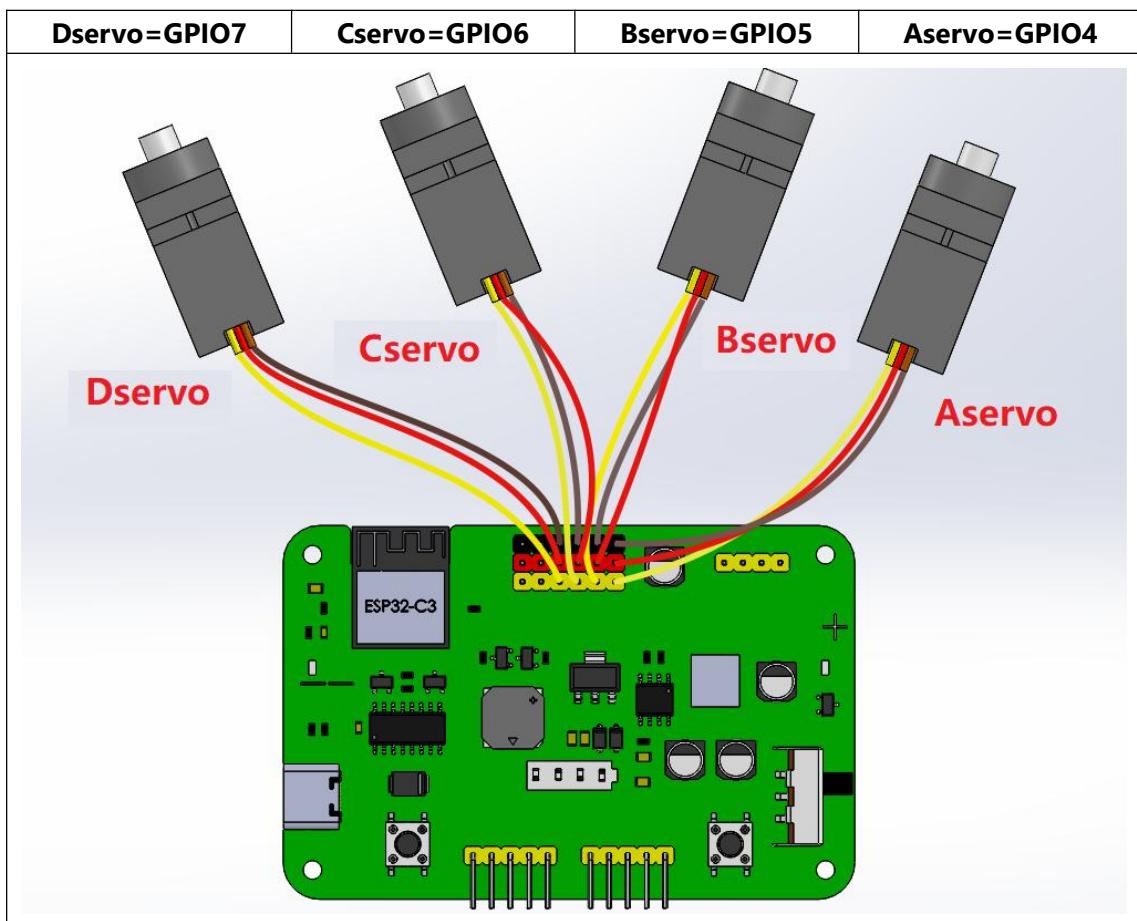
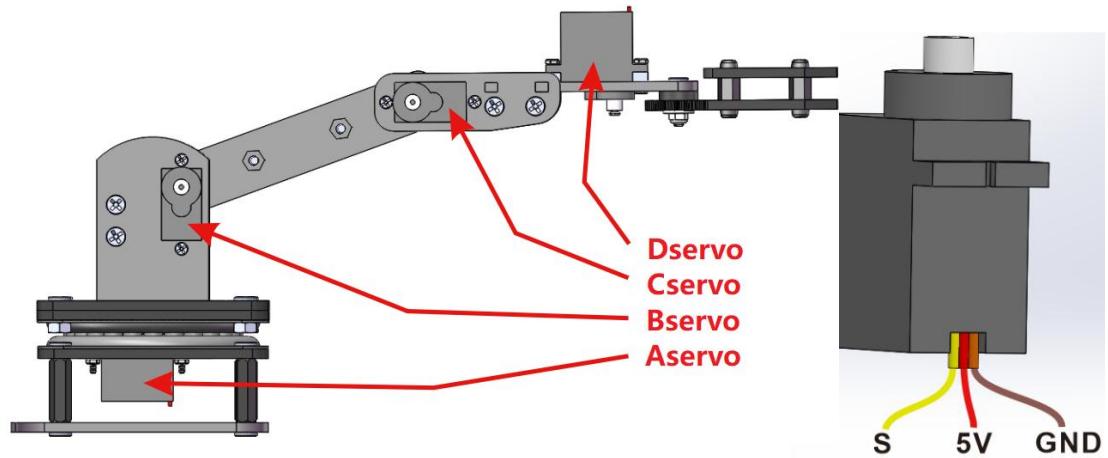
Step 21 Connect the Joystick

Part in Step 19	Part in Step 20
5P Dupont Wire 2PCS	

Connect the joystick's GND to the control board's GND as the reference

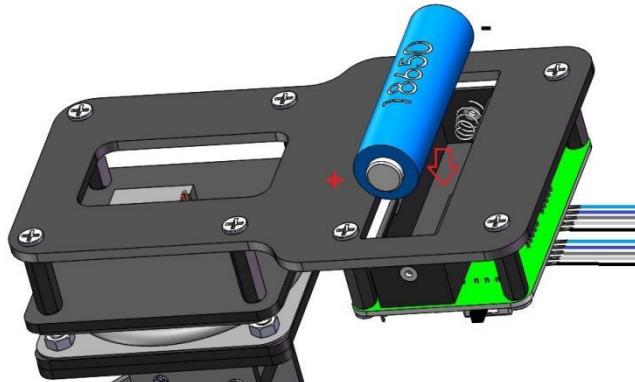


Step 22 Connect the Servos to the ESP32 Board

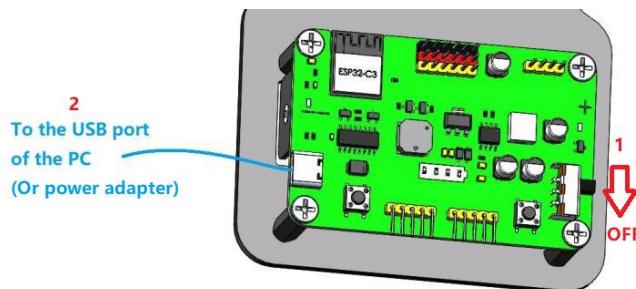


Step 23 Initialize the Servo Angle (important!)

Install a 18650 lithium battery (need to be purchased by yourself)

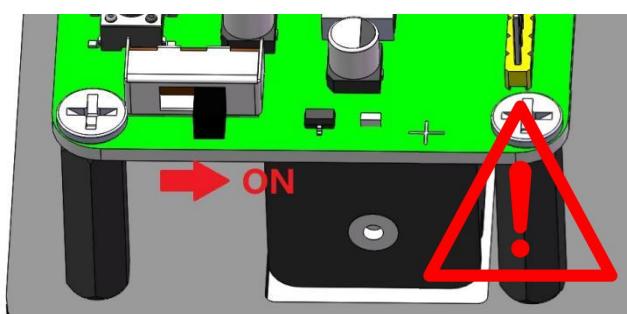


When installing the battery for the first time, please use the USB cable to charge the battery to activate the battery!



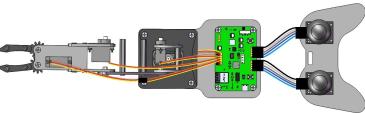
Very important: Turn on the power switch! ! !

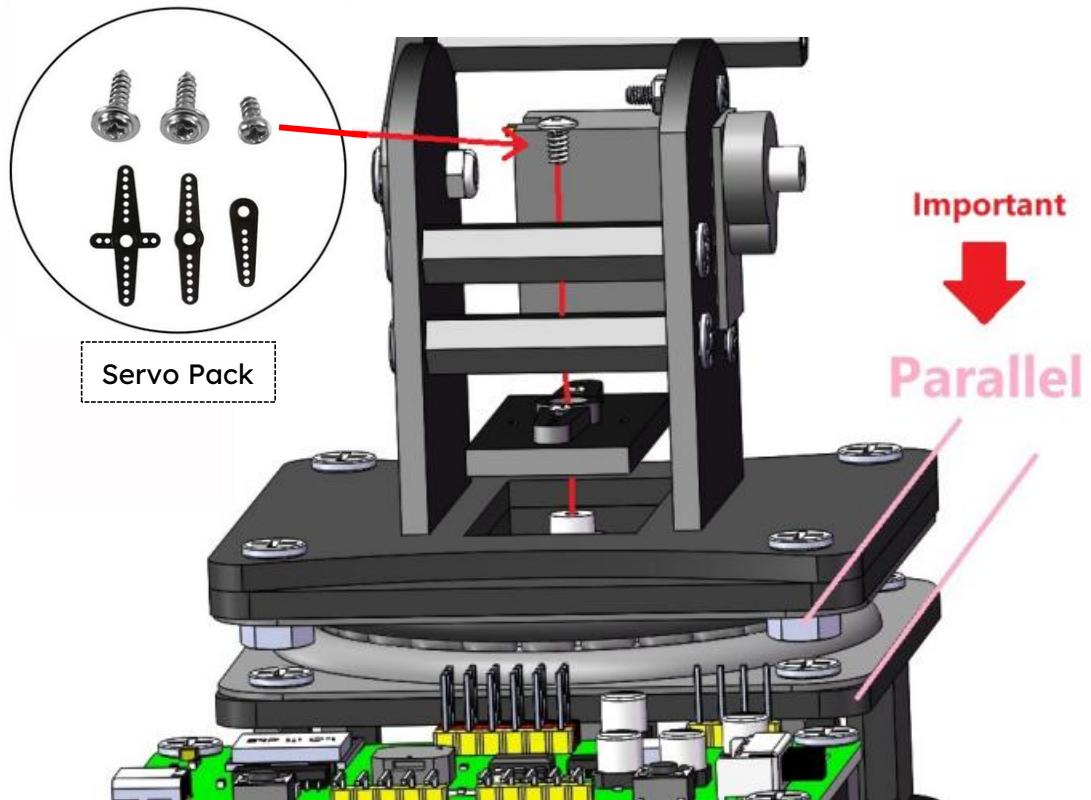
The **firmware** has been successfully programmed into the control board. When powered on, the board will automatically position all servo motors at the **90-degree** default position.



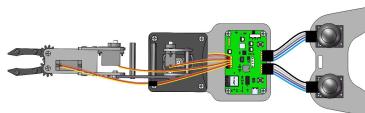
1. Make sure the battery is fully charged!
2. When you turn on the power switch, You will hear the servo motor shaft rotating.
3. In the following installation steps, please keep the power on to prevent the servo angle from being changed!

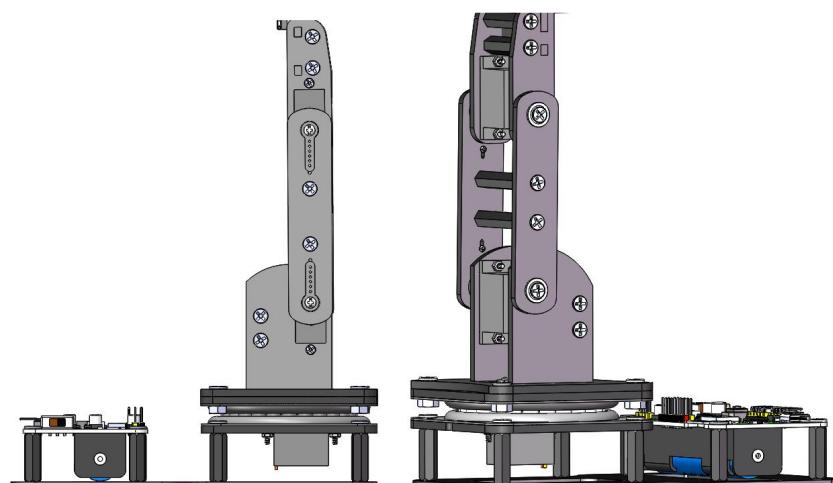
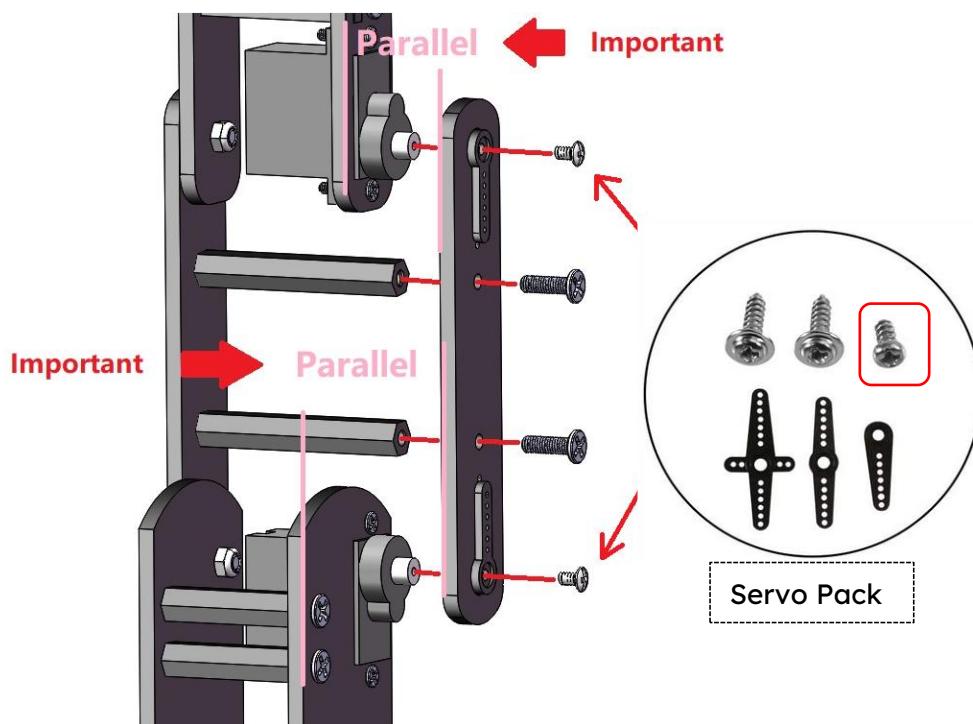
Step 24 Fix the Robot Arm to Servo A of the Base

Step 23 Structure	Step 17 Structure	M2.5X4mm Screw 1PCS
		



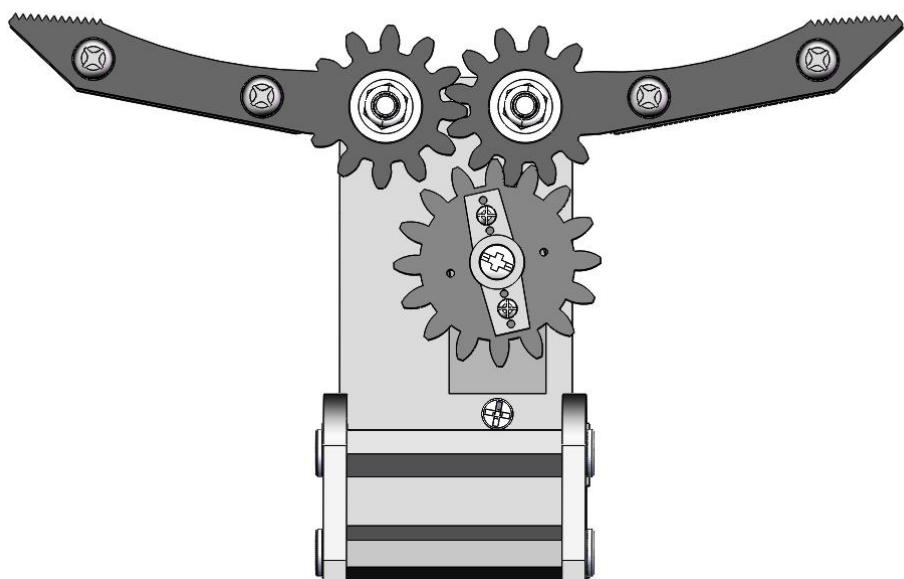
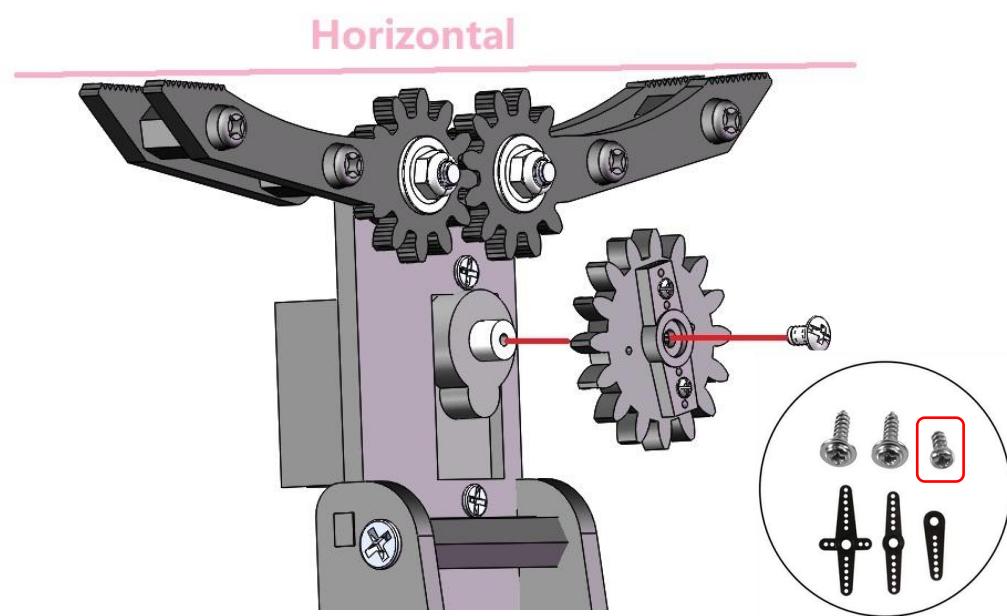
Step 25 Assemble the Middle Arm

Step 24 Structure	Step 16 Structure
	
M3X10MM Screw 2PCS	M2.5X4mm Screw 2PCS
	

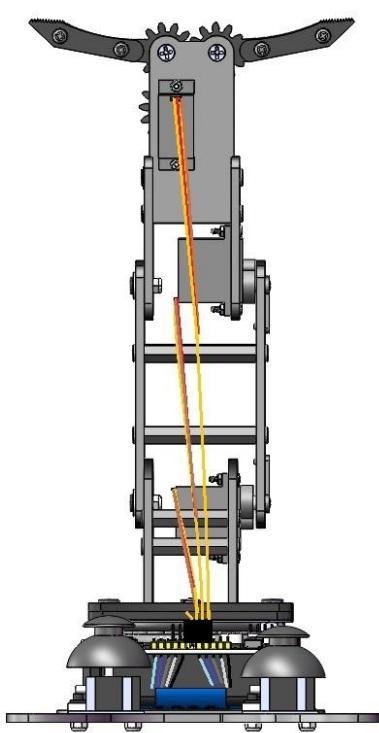
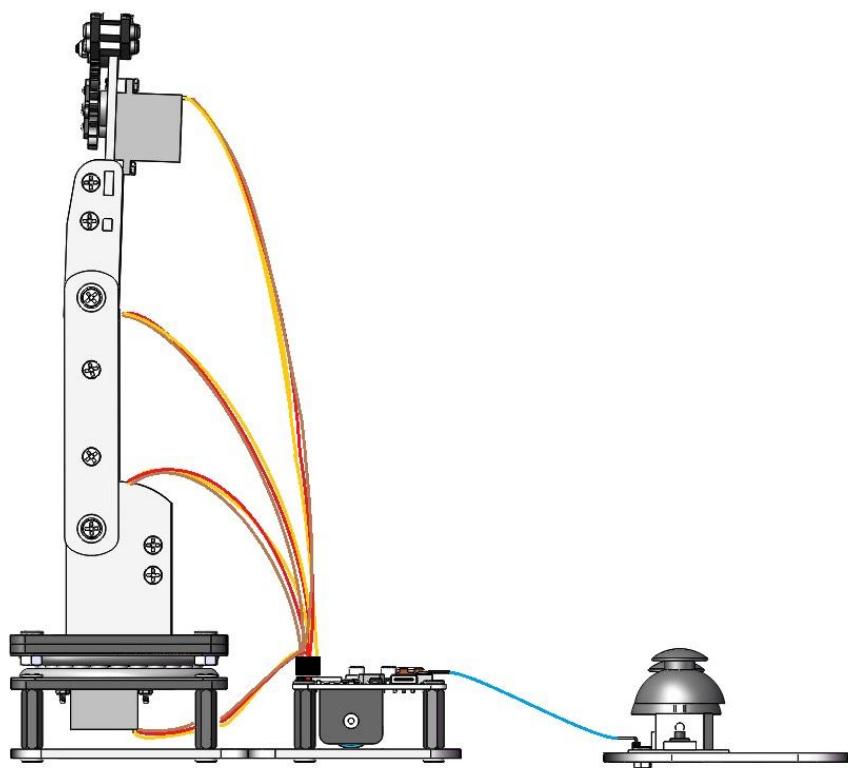


Step 26 Fix the Clip Plates to Servo D

Step 25 Structure	Step 18 Structure	M2.5X4mm Screw 1PCS

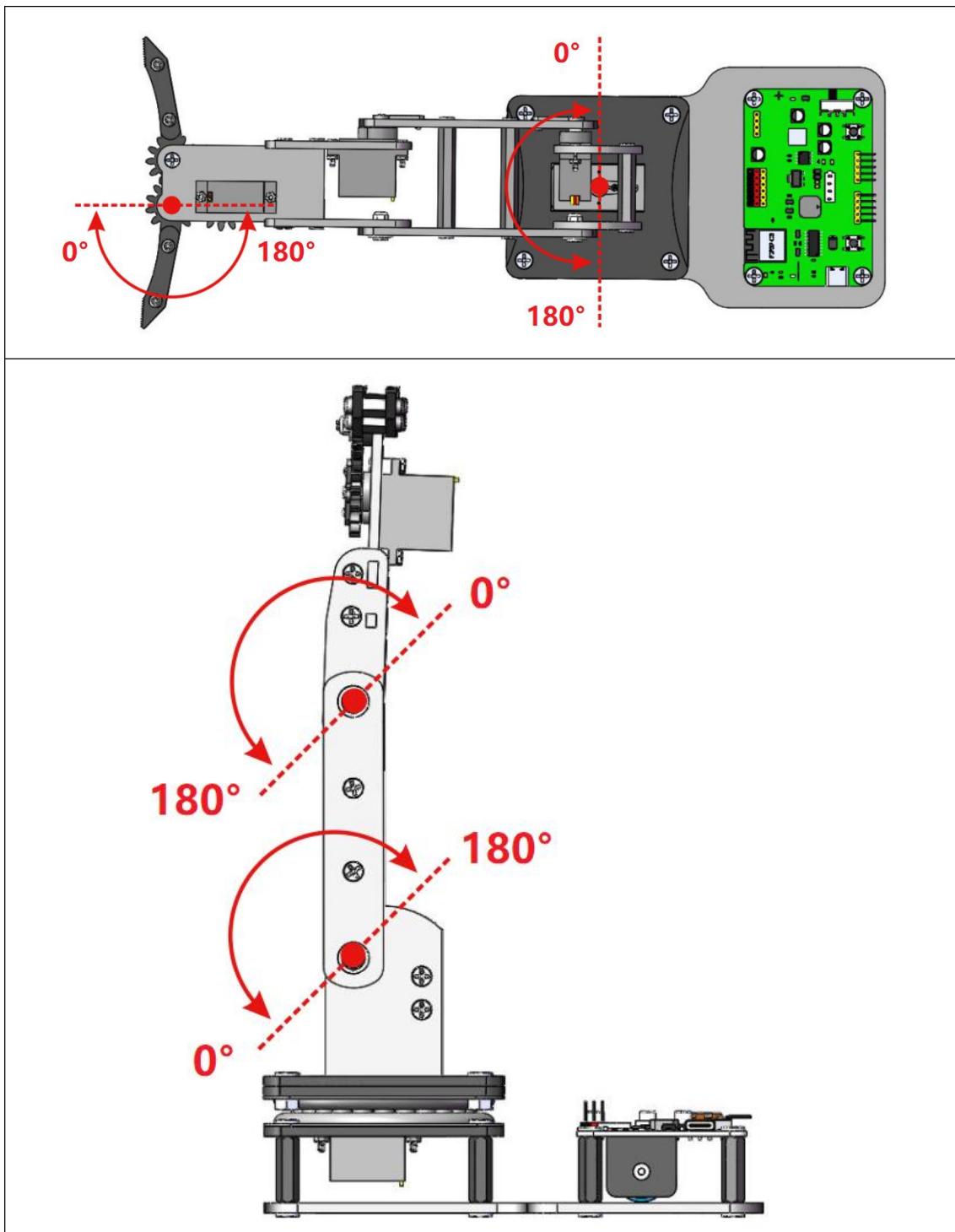


Step 27 Assembly completed



Using the Robot Arm

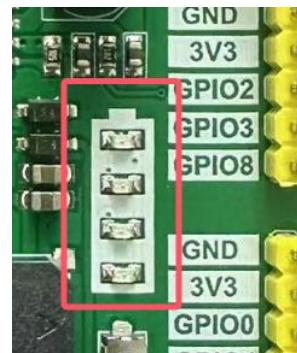
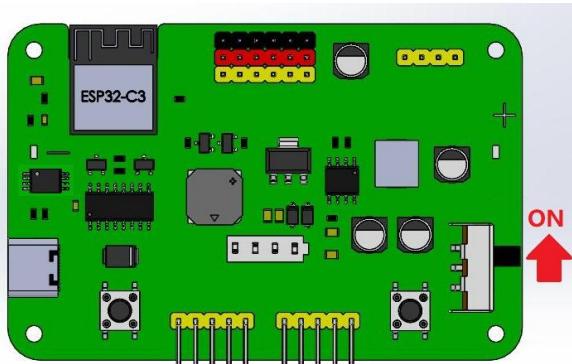
3.1 Degrees of Freedom



3.2 Control the eArm with a Joystick

Turn the eArm's power switch to the **ON** position.

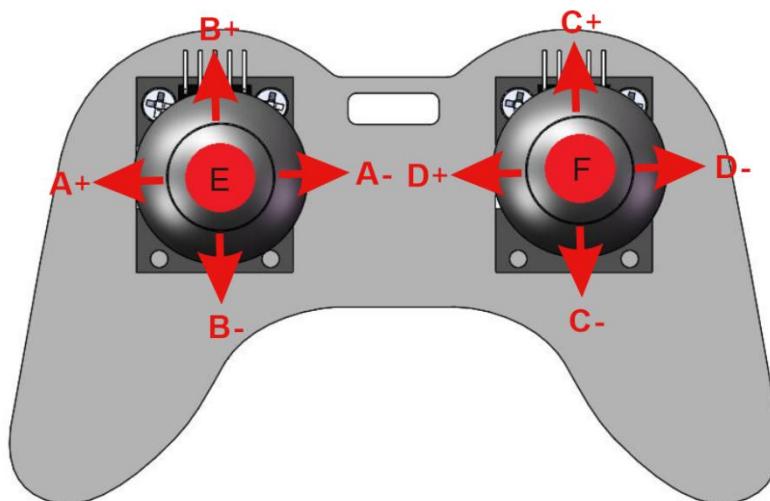
- 1: Make sure the 18650 lithium battery is installed.
- 2: The battery indicator shows 3 bars or more.



After powering on the robot, the default control mode is **Joystick Mode**. To switch between control modes, press the "**F**" button on the right joystick. Available modes include:

- ✓ ■ Joystick Mode (default)
- ✓ ■ Web App Mode

When you press "F", the buzzer will beep once, indicating the switch to **Web App Mode**. In this mode, joystick controls will be disabled.



A+: Rotate arm left (Servo A)

B-: Raise rear arm (Servo B)

C-: Raise the forearm (Servo C)

D+: Open the gripper (Servo D)

A-: Rotate arm right (Servo A)

B+: Lower rear arm (Servo B)

C+: Lower the forearm (Servo C)

D-: Close the gripper (Servo D)

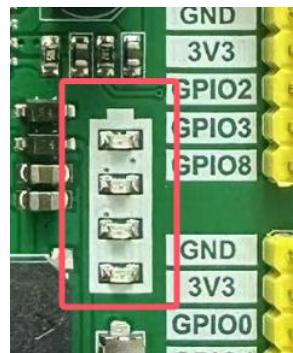
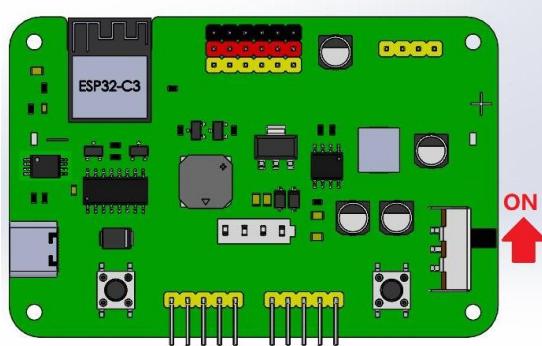
F: Toggle between Joystick and Web App control modes

E: Enable/disable the buzzer

3.3 Control the eArm with Web App

Turn the eArm's power switch to the **ON** position.

- 1: Make sure the 18650 lithium battery is installed.
- 2: The battery indicator shows 3 bars or more.

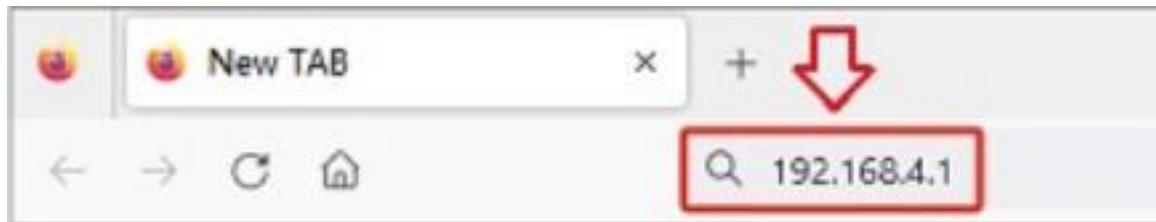


Turn on the phone's Wi-Fi and connect to the network named '**eArm**'

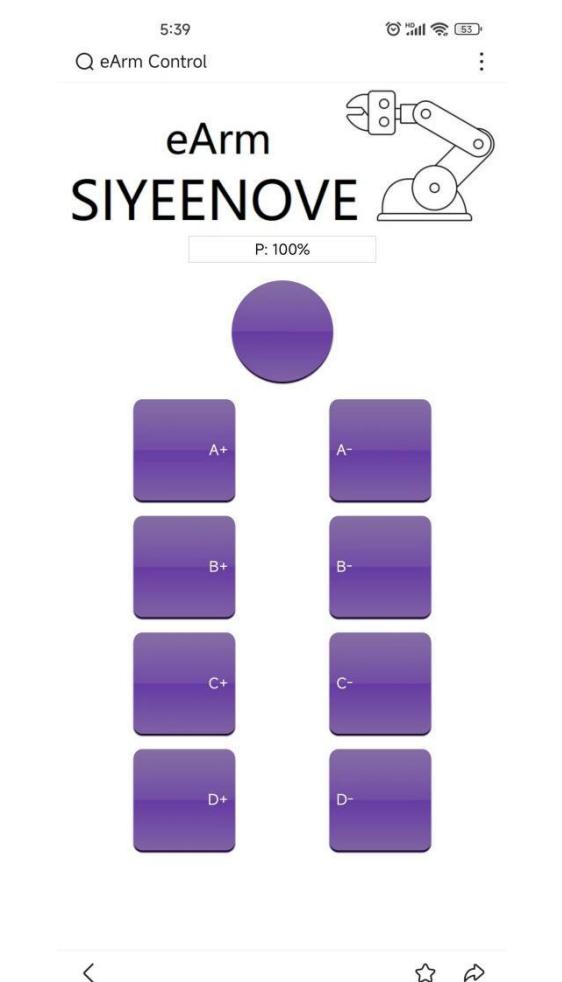
- Once connected to the Wi-Fi, you may see a '**Network connection failed**' message; **just disregard it**.
- Step 1: **Turn off mobile data** in your phone settings
- Step 2: This ensures stable connection to the Web App



1. Open the web browser on your phone
2. Type **192.168.4.1** into the address bar
3. Press Enter to connect



After successful connection, the control interface will appear in your browser - you can now operate the eArm!



The screenshot shows a mobile browser interface for 'eArm Control'. At the top, it displays the time '5:39' and signal strength. Below the header, there's a logo featuring a stylized robot arm and the text 'eArm SIYEENOVE' with a progress bar at 'P: 100%'. The main area contains a 4x2 grid of buttons. A large purple circle is positioned above the first column of buttons. The buttons are labeled as follows:

Button	Mechanical Movement
A+	Robot arm turns left
A-	Robot arm turns right
B-	Rear arm raises
B+	Rear arm lowers
C-	Front arm raises
C+	Front arm lowers
D+	Gripper opens
D-	Gripper closes

At the bottom of the screen, there are navigation icons: a left arrow, a star, and a right arrow.

4

Programming Learning

Before You Begin: Important Notes

The ESP32 control board comes preloaded with firmware that enables robotic arm operation as demonstrated in the previous section. This functionality will remain available until you upload new code.

Important notes about code uploading:

- 1: The ESP32 board always executes the most newly uploaded program
- 2: Each new code upload overwrites the previous program

This section will guide you through programming the robot using code examples ranging from basic to advanced. The final example contains our factory-preloaded firmware - the same code that enables the arm functionality shown in the previous section.

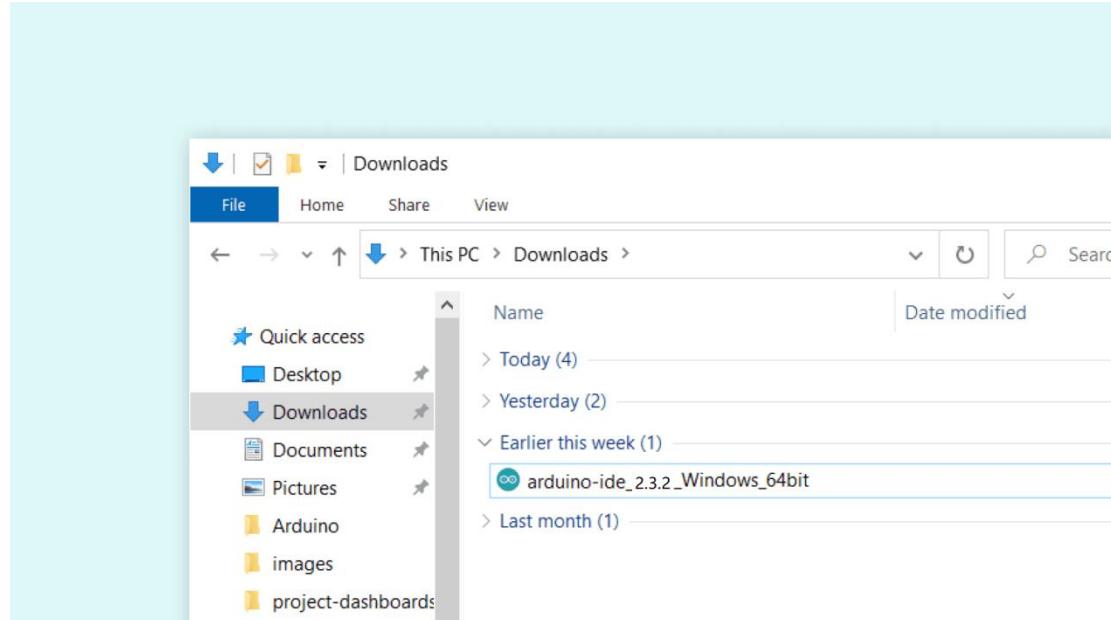
4.1 Install Arduino IDE

The latest version of Arduino IDE can be downloaded from the Arduino official website: <https://www.arduino.cc/en/software>

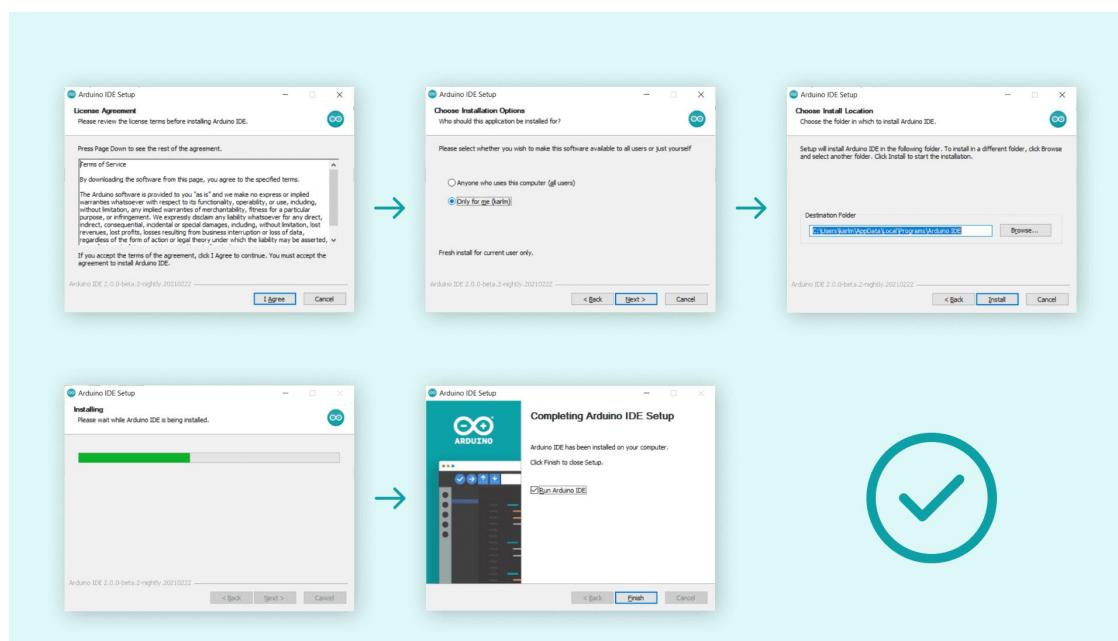
In the following we will demonstrate the installation of **Windows Win 10 and newer, 64 bits** version IDE on PC.

The screenshot shows the Arduino IDE 2.3.2 download page. On the left, there's a logo and the text "Arduino IDE 2.3.2". Below it, a paragraph describes the features of the new release. A link to "Arduino IDE 2.0 documentation" is provided. At the bottom, there's a "SOURCE CODE" link and a note about GitHub. On the right, a "DOWNLOAD OPTIONS" section lists "Windows Win 10 and newer, 64 bits" (selected), "Windows MSI installer", "Windows ZIP file", "Linux AppImage 64 bits (X86-64)", "Linux ZIP file 64 bits (X86-64)", "macOS Intel, 10.15: "Catalina" or newer, 64 bits", and "macOS Apple Silicon, 11: "Big Sur" or newer, 64 bits". A "Release Notes" link is also present.

To install the Arduino IDE 2.x.x on a Windows computer, simply run the file downloaded from the software page.



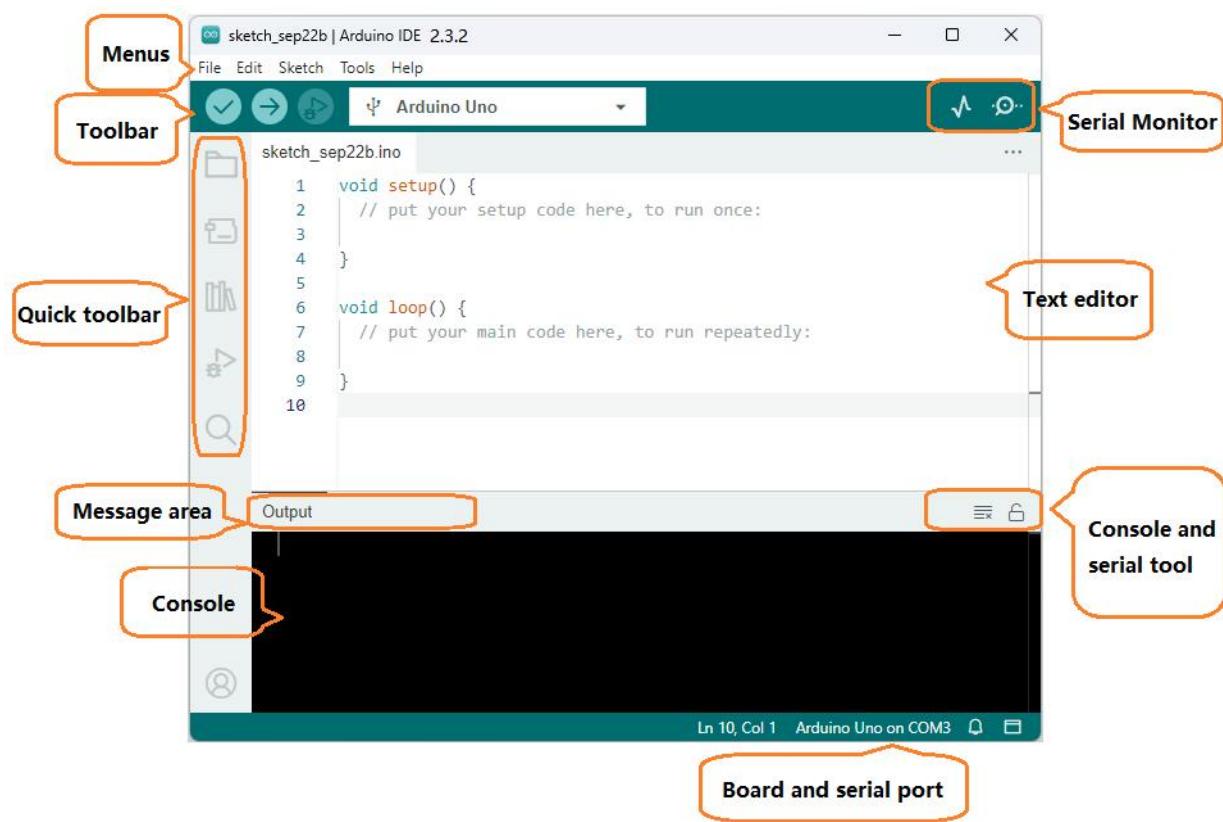
Follow the pop-up window prompts to complete the installation process.



Startup icon:

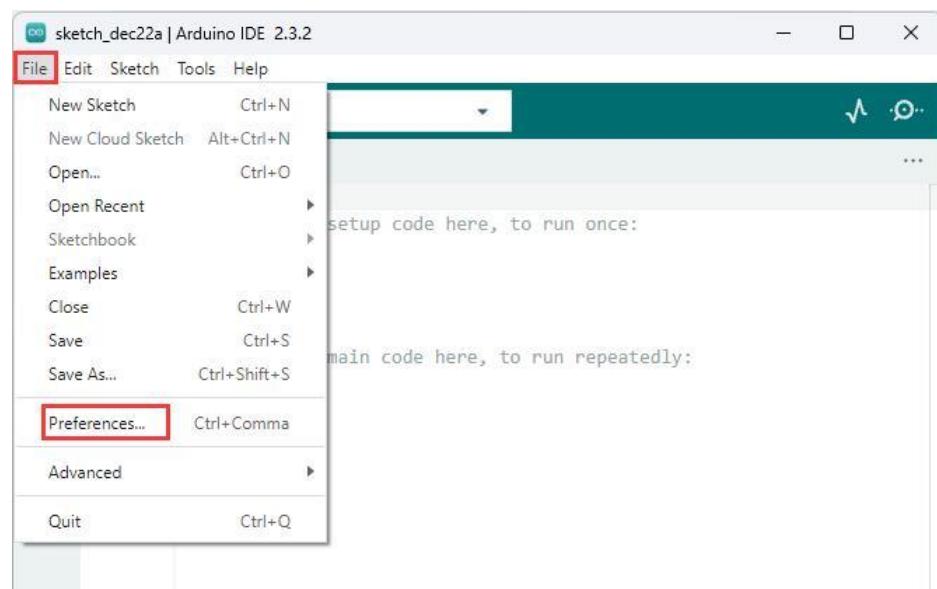


The homepage of Arduino IDE



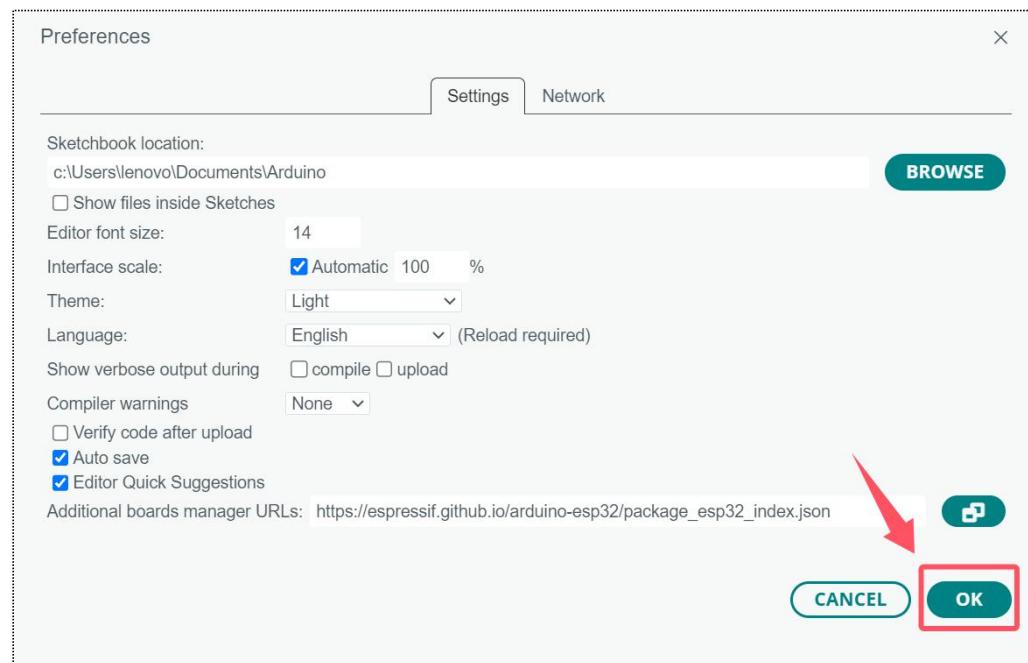
4.2 Install ESP32 Development Environment for Arduino IDE

Open the Arduino IDE, click “File” > “Preferences”, as shown below:

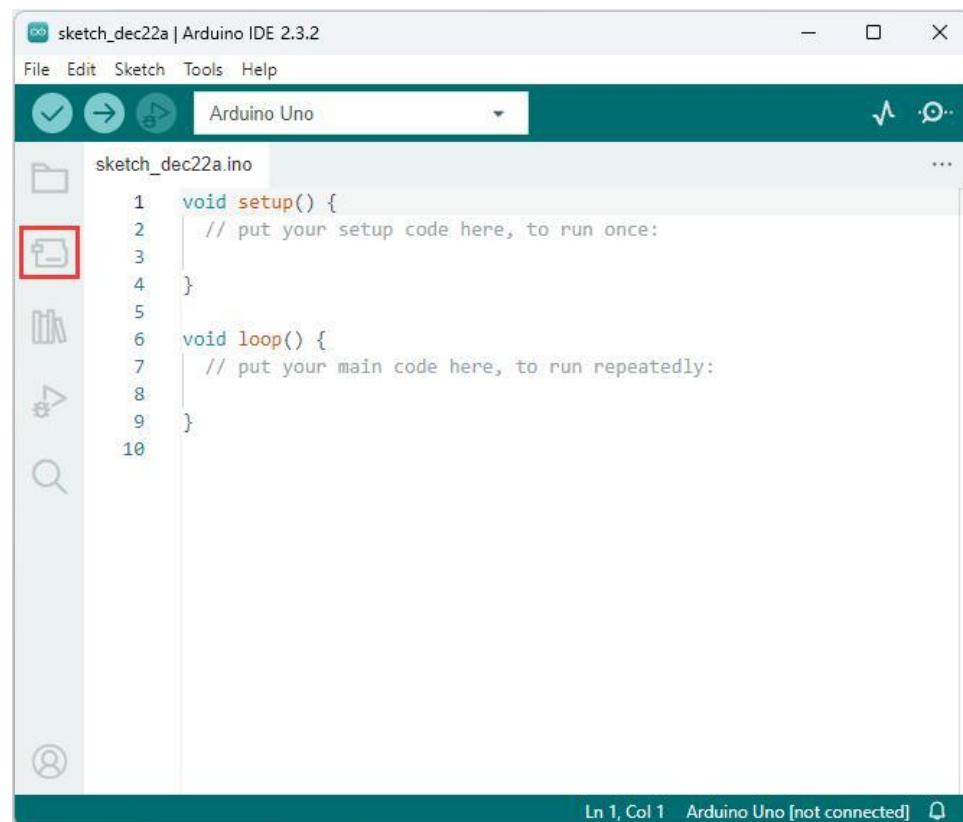


Scroll down the pop-up window and copy the link below into the **Additional boards manager URLs** text box. Then click “OK”.

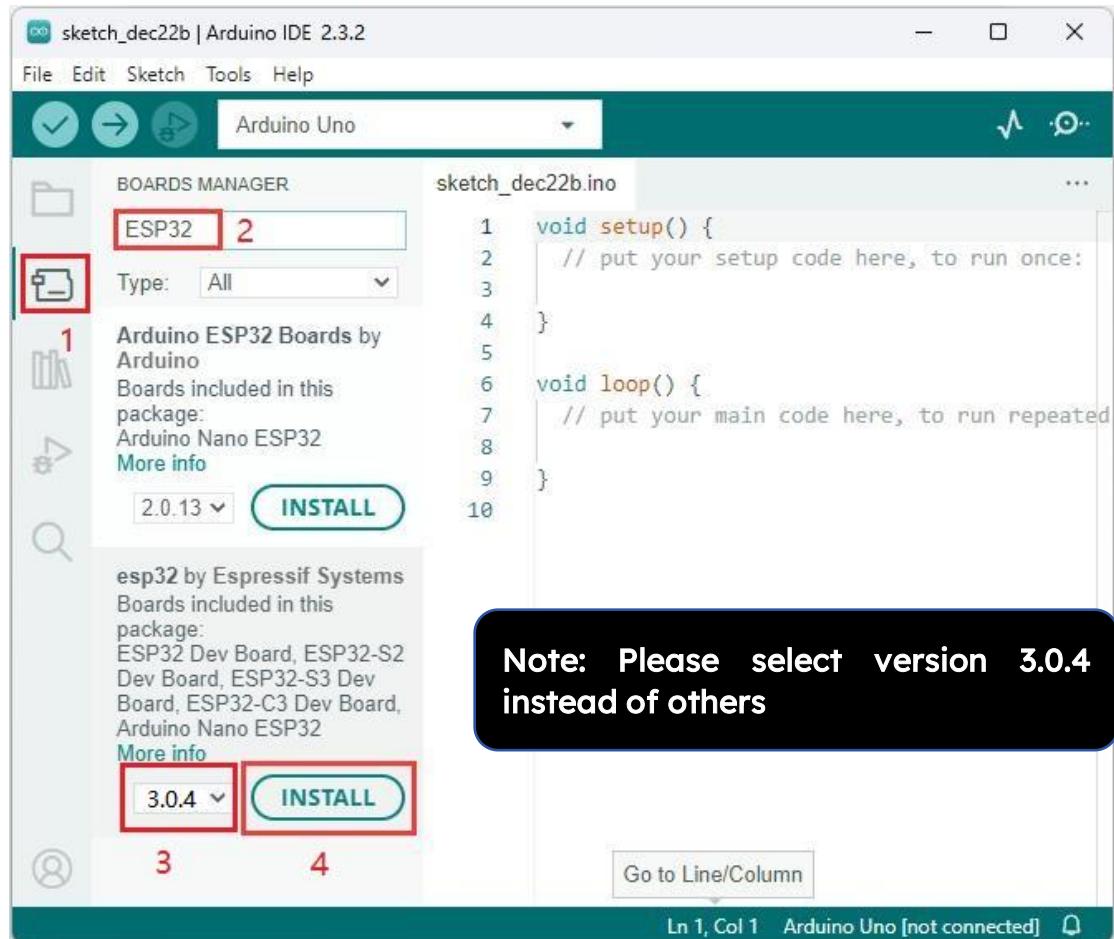
https://espressif.github.io/arduino-esp32/package_esp32_index.json



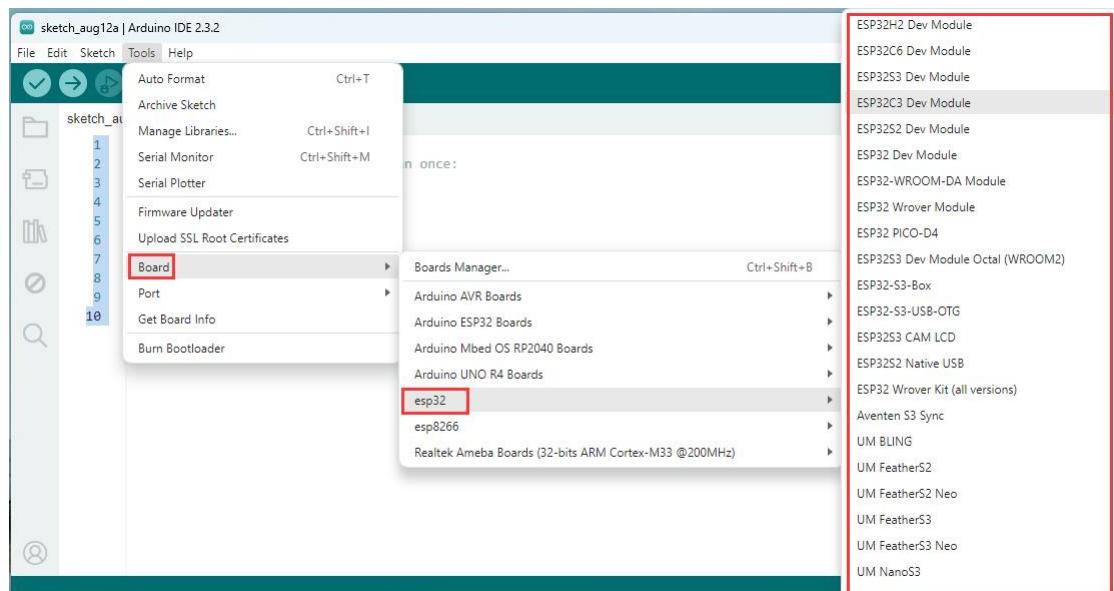
Click “Boards Manager”:



Search for **ESP32** in the Boards Manager, select **esp32 by Espressif Systems** in **3.0.4** version and click **INSTALL**



After successfully installing the ESP32 board, click the "Tools"->"Board"->"ESP32" menu, and select **ESP32C3 DVE Module**.



4.3 Install the Library File

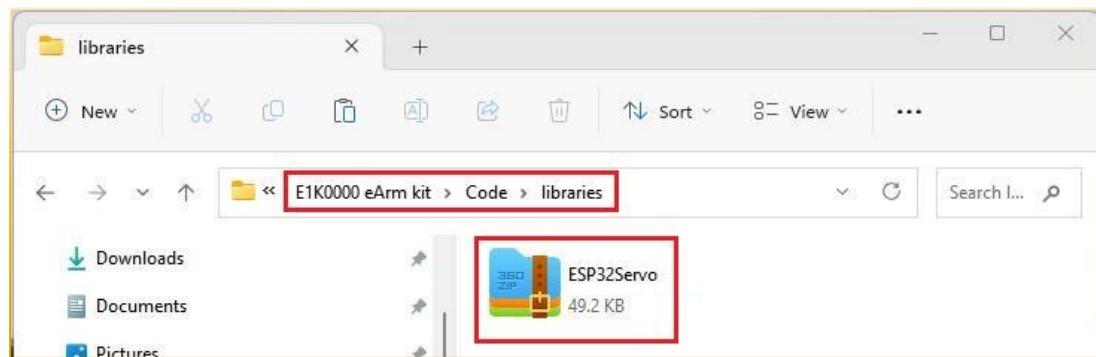
Method 1:

Open the library manager of Arduino IDE and type "ESP32Servo" in the search bar, select the library file named ESP32Servo and version 3.0.5 and click Install.

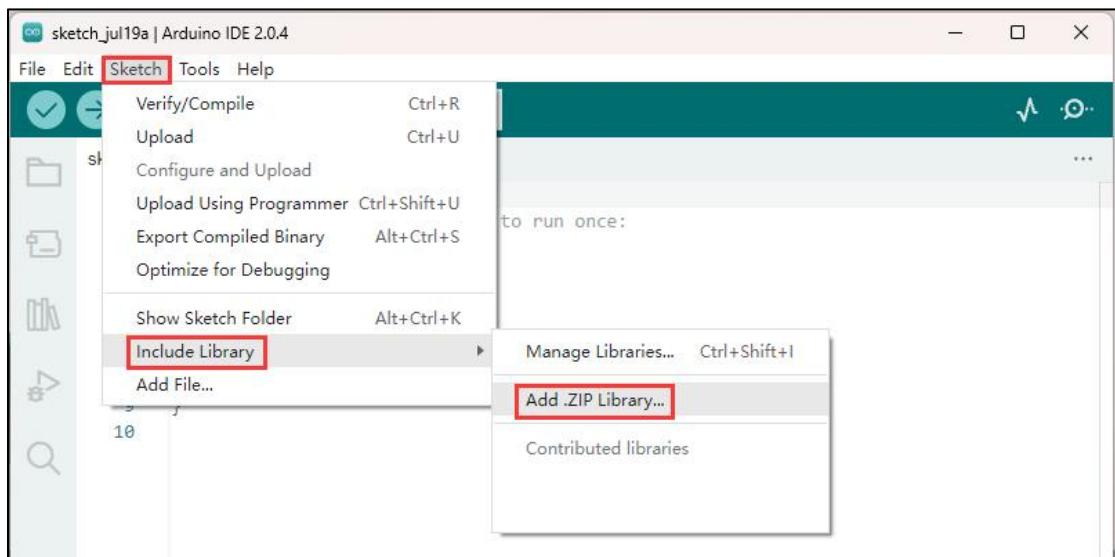


Method 2:

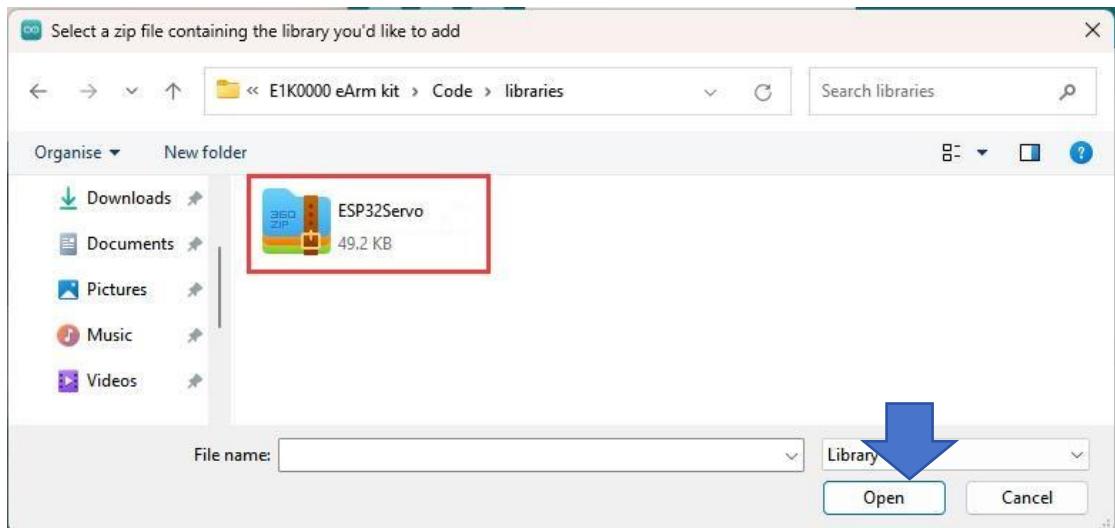
We provide a library zip file in the tutorial package, in the following folder:



Import the library zip file from the Arduino IDE as follows:

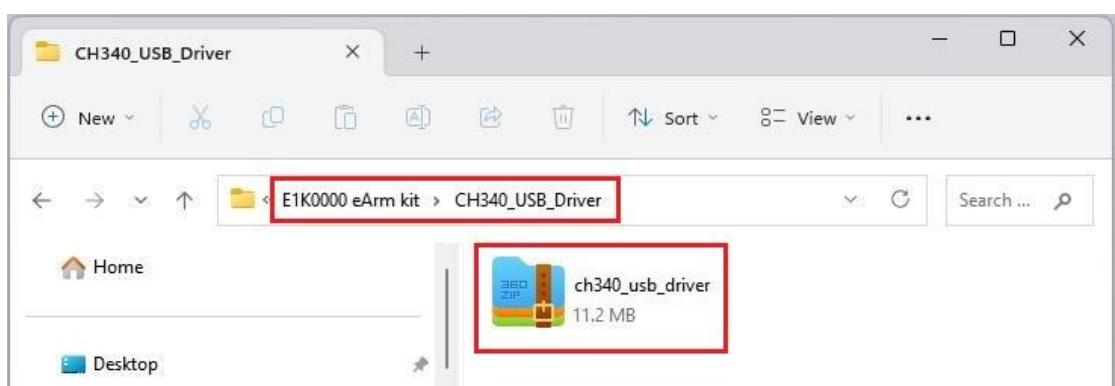


Click Open to install the library file.



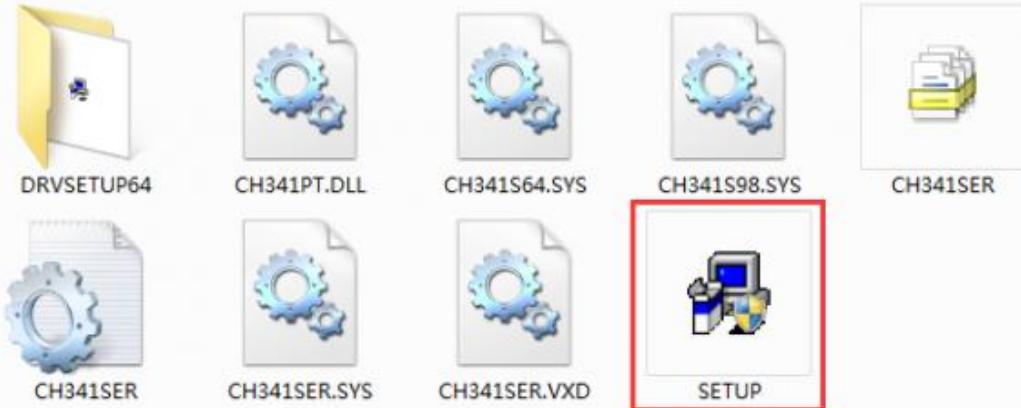
4.4 Install CH340 USB driver

Driver file is provided in our tutorial package:



Install the Driver on Windows system

Unzip the file we provided and double-click “**SETUP**” to run the installer:



If the driver is successfully installed, when the esp32 board is connected to the computer through the USB cable, under the path “**Computer**” -> “**Properties**” -> “**Device manager**”, you can find “**USB-SERIAL CH340 (COM3)**”, as shown below:

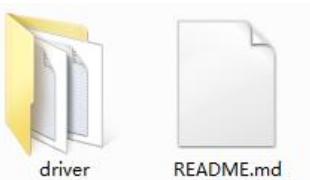


Note: Driver display symbol is “**USB-SERIAL CH340 (COMx)**”, “x” can be any number, it depends on what number your computer assigns to the ESP32 device.

Install the Driver on Linux system

Drivers are almost certainly built into your Linux kernel already and it will probably just work as soon as you plug it in. If not you can download the Linux CH340 Driver (but I’d recommend just upgrading your Linux install so that you get the “built in” one).

If you must install it by yourself, check the “**README.md**” file in the zip package.



Install the Driver on MAC system

Link to download the driver:

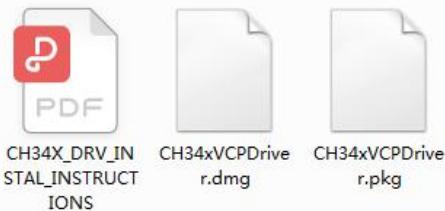
http://www.wch-ic.com/downloads/CH341SER_EXE.html

 download

relation files

file name	file content
CH341SER.ZIP	CH340/CH341 USB to serial port Windows driver, supports Windows XP/Vista/7/8/8.1/10/11/SERVER 2003/2008/2012/2016/2019/2022 -32/64bit, Microsoft WHQL Certified, supports USB to 3-line and 9-line serial port.
CH341SER_LINUX.ZIP	Linux driver for USB to serial port, supports CH340 and CH341, supports 32/64-bit operating systems.
CH341SER_MAC.ZIP	For CH340/CH341/CH342/CH343/CH344/CH347/CH9101/CH9102/CH9103/CH9104/CH9143, USB to serial port VCP vendor driver of macOS , supports OS X 10.9~10.15 , OS X 11 (Big Sur) and above , contains installation guide documents.
CH340DS1.PDF	CH340 datasheet, USB bus converter chip which converts USB to serial port, printer port, IrDA SIR etc. Integrated clock, supports various operating systems, chip information can be customized, this datasheet is about USB to serial port and USB to Infrared Adapter SIR.
CH341DS1.PDF	CH341 datasheet, USB bus converter chip with multiple communication interfaces, such as USB to serial port/parallel port/printer port/I2C interface etc. Drivers support for Windows/Linux/Android/Mac, etc. The datasheet is the description of USB to serial port and printer port.
CH341SER_ANDROID.ZIP	CH340/CH341/CH342/CH343/CH344/CH347/CH9101/CH9102/CH9103/CH9104/CH9143 USB to serial port Android driver-free application library and software, for Android OS 4.4 and above in USB Host mode, without loading Android kernel driver and without root access operation. Includes apk installer, lib library file (Java Driver), App Demo routines (USB to UART Demo project SDK).

Download the driver from the website and unzip the file to a local installation directory. You will get a PDF installation guide and two installation packages in different formats. For details, see the PDF installation guide.



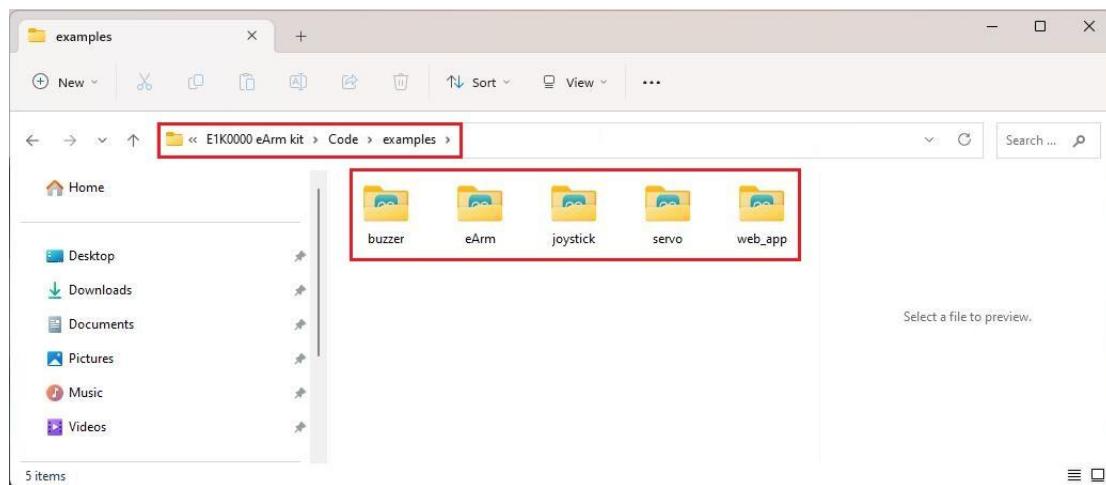
CH34X_DRV_INSTRUCTIONS

CH34xVCPDrive.r.dmg

CH34xVCPDrive.r.pkg

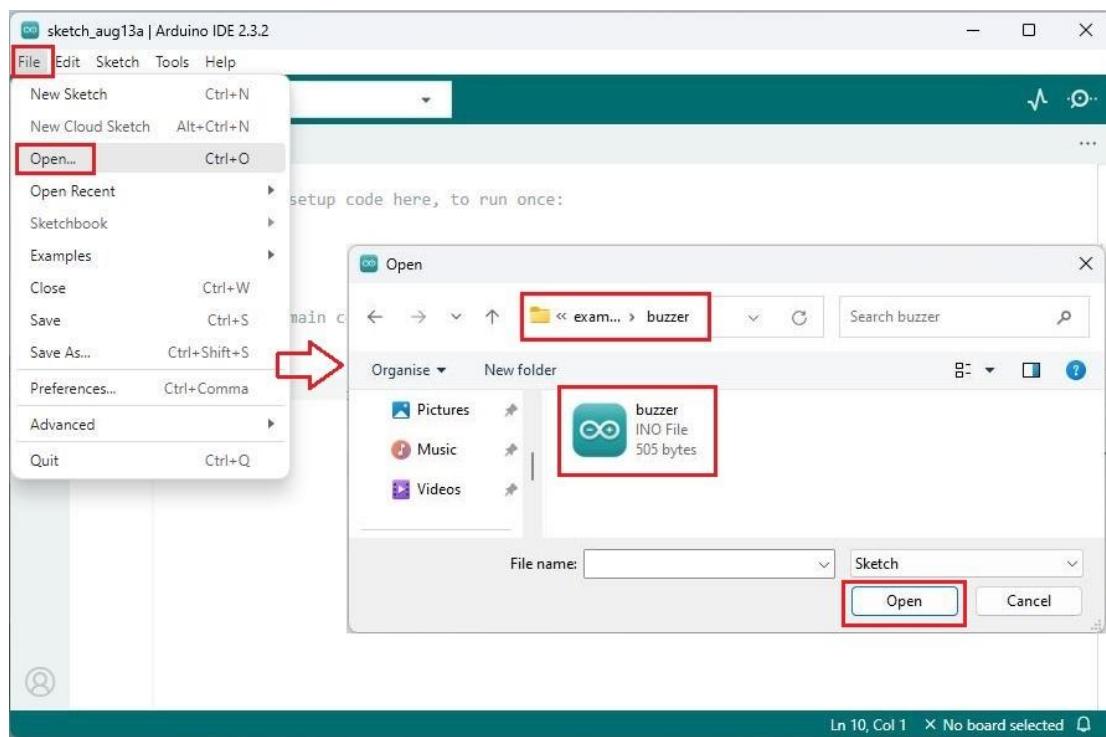
4.5 Code Examples

The sample codes are stored in the following folders in the tutorial package:

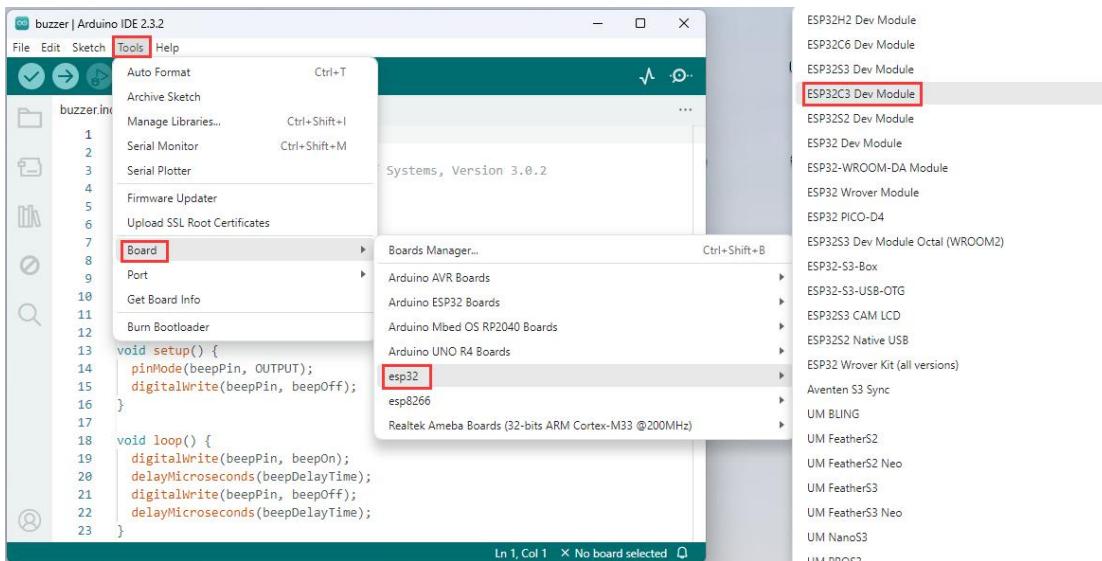


4.5.1 Onboard buzzer

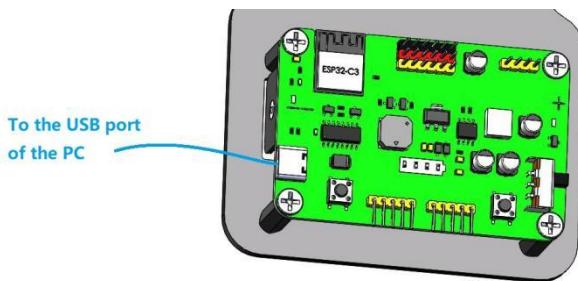
Open the "buzzer" example code in the Arduino IDE:



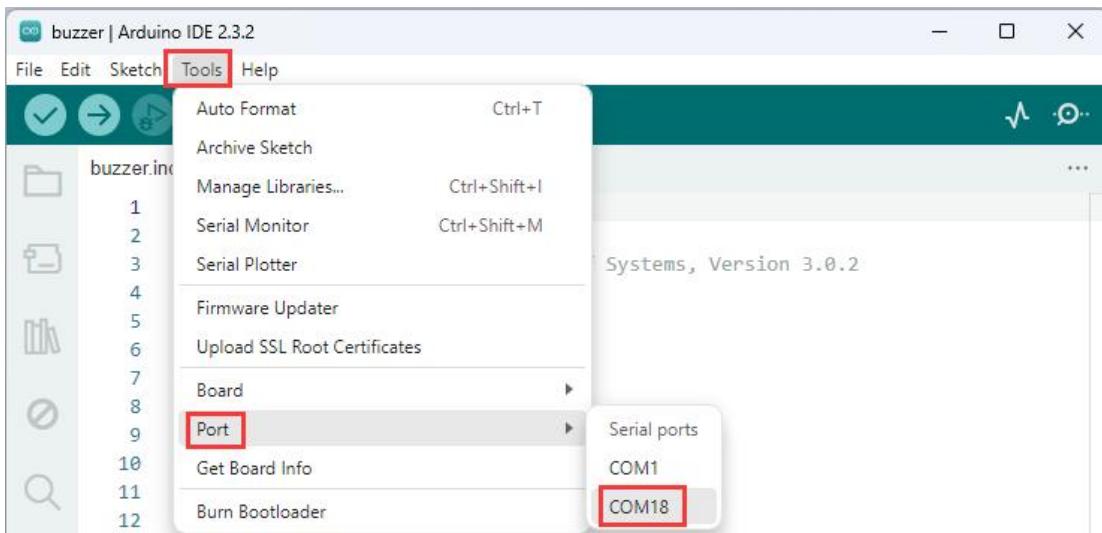
Select board type (ESP32C3 Dev Module) :



Use the USB cable to connect the PC and eArm, as shown below:



Select Port:



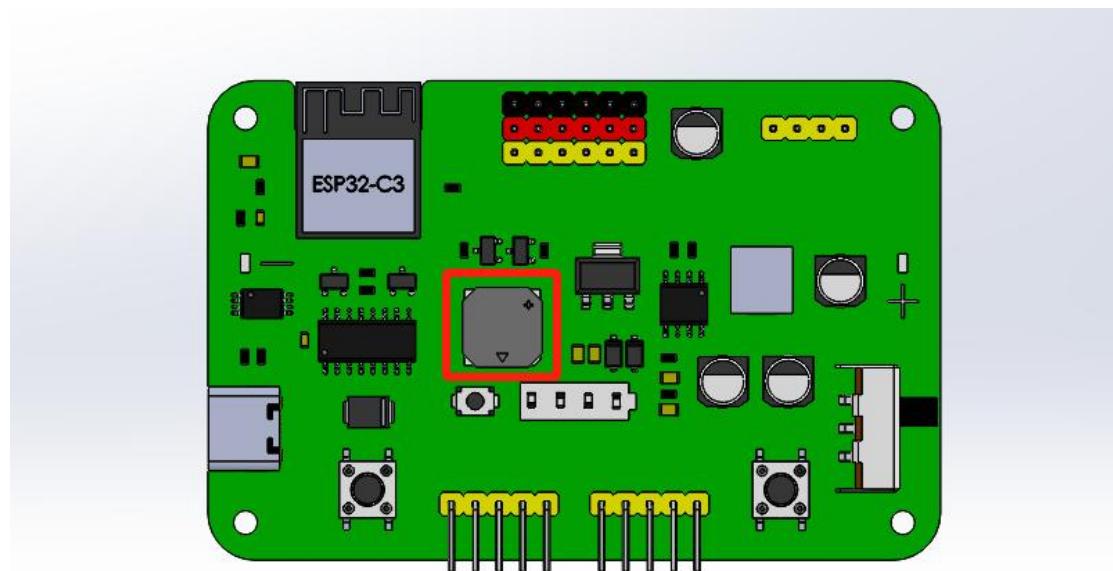
Note: The port in the above picture is COM18. Your computer may have a different numbered port. Please select it according to the port assigned by the device manager on your computer.

Click to upload the code:

The screenshot shows the Arduino IDE interface with the title "buzzer | Arduino IDE 2.3.2". The toolbar at the top has icons for file operations, sketch management, tools, and help. A red box highlights the "Upload" button (a right-pointing arrow) in the toolbar. The central workspace displays the code for "buzzer.ino". The code defines constants for beepOn (0), beepOff (1), beepFreq (500), and beepDelayTime (1000000). It also initializes a pin (beepPin = 9) and sets it to output mode. The "Output" tab at the bottom shows the terminal window with the following text:
Writing at 0x00050e0d... (100 %)
Wrote 274304 bytes (152477 compressed) at 0x00010000 in 4.4 seconds (effective 497.9 kb
Hash of data verified.

Leaving...
Hard resetting via RTS pin...
A red box highlights the last two lines of text in the terminal window.

After the code is uploaded successfully, the buzzer on the ESP32 board will keep making sounds:

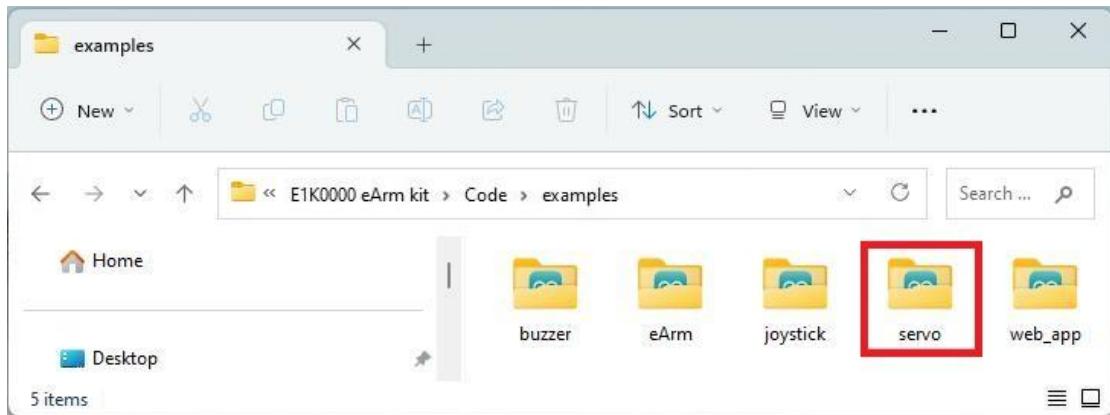


Code Explanation:

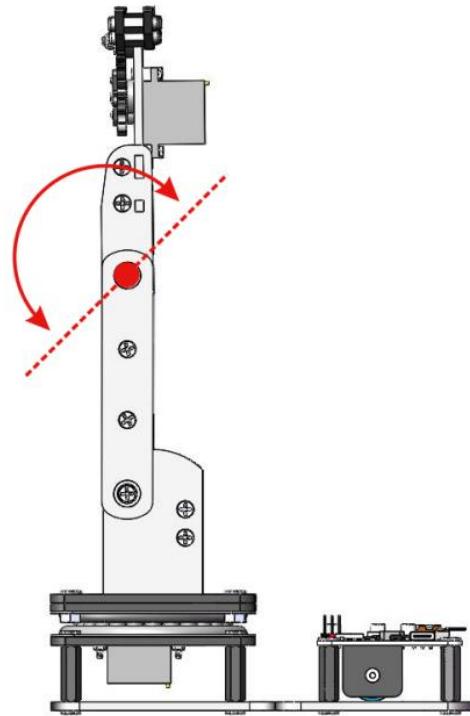
#define beepOn 0	Define a constant "beepOn" with a value of 0.
#define beepDelayTime 1000000 / beepFreq	Define a constant "beepDelayTime" with a value of: 1000000 / beepFreq.
char beepPin = 9;	Define a character variable "beepPin" with a value of 9, mapped to the control pin of the buzzer.
void setup() { ... }	Built-in function of Arduino IDE, only runs once when powered on.
pinMode(beepPin, OUTPUT);	Set pin 9 to output mode.
digitalWrite(beepPin, beepOff);	Pin 9 outputs a high level to turn off the buzzer.
void loop() { ... }	Built-in function of Arduino IDE, runs in a loop all the time when powered on.
delayMicroseconds(beepDelayTime);	Built-in function of Arduino IDE, delay in microseconds.

4.5.2 To Drive a Servo

Please refer to section 4.3.1 to upload the "servo" code to the esp32 control board:



After the code is uploaded successfully, the eArm's servo C will swing 0-180, 180-0 in a cycle:

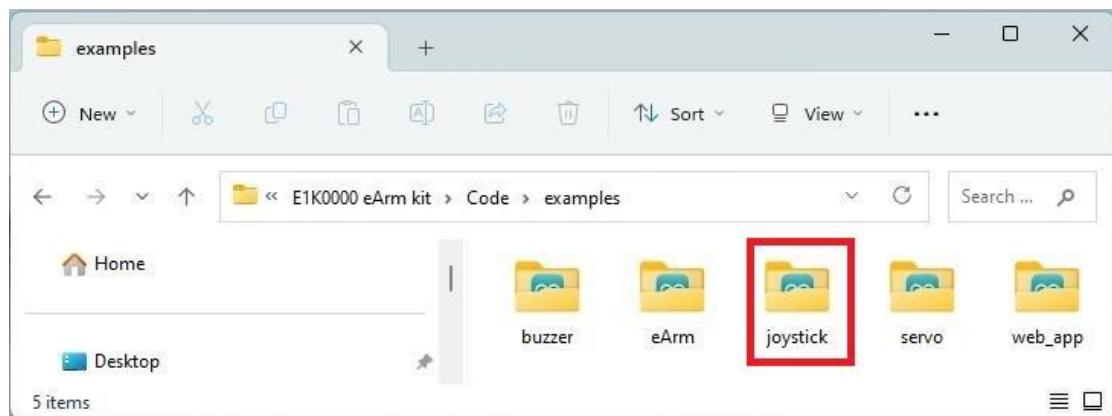


Code Explanation:

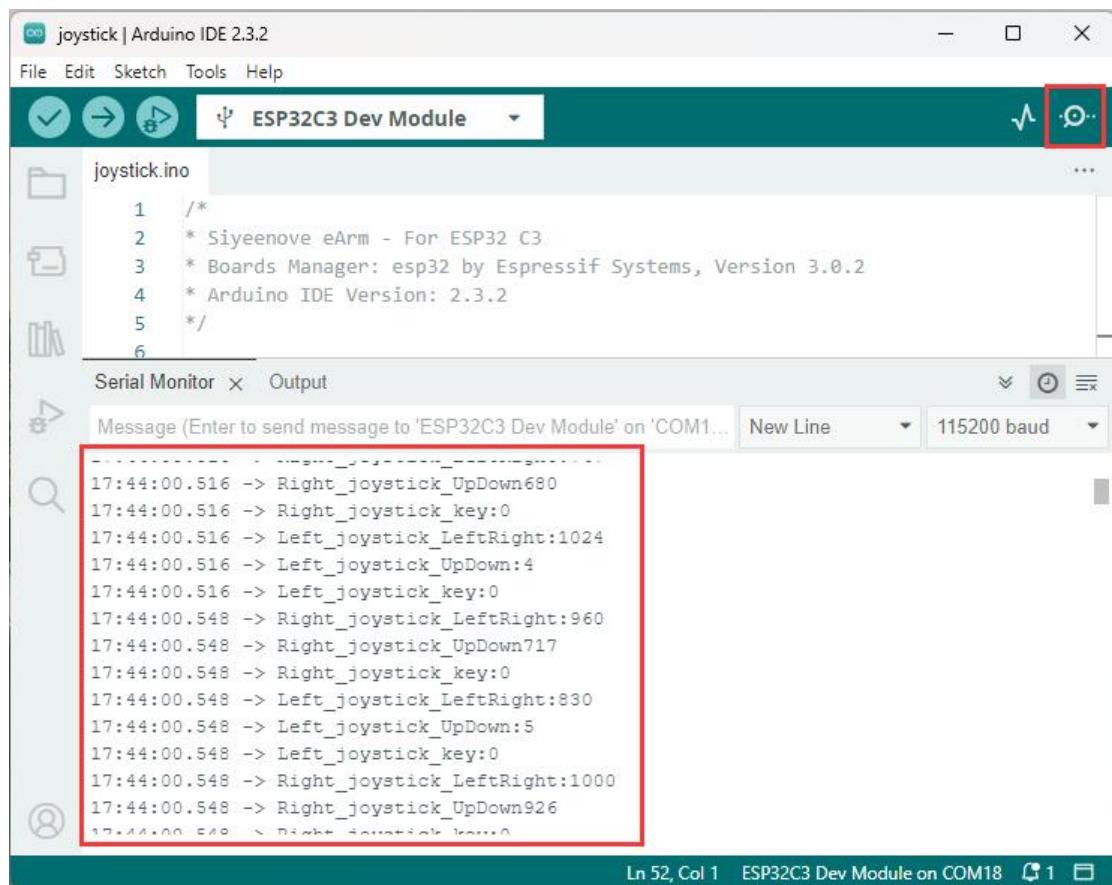
<code>#include <ESP32Servo.h></code>	Include an ESP32 servo driver library.
<code>Servo Aservo;</code>	Define a servo class to drive Servo1.
<code>ESP32PWM::allocateTimer(0);</code>	Allow the servo driver library to use timer 0.
<code>Aservo.setPeriodHertz(50);</code>	Set the pulse drive frequency of the driving servo Aservo to 50 Hz.
<code>Aservo.attach(AservoPin, 500, 2400);</code>	Set the driving servo Aservo to use the AservoPin pin to drive, the latest pulse width is 500 microseconds, and the maximum pulse width is 2400 microseconds.
<code>Aservo.write(aServoAngle);</code>	Set the driving servo Aservo to rotate to the angle aServoAngle..
<code>for(statement1;condition;statement2){ ... }</code>	For loop statement, each time statement1 is executed, it determines whether the condition is true. If so, the statements in the curly braces are executed, otherwise the loop is interrupted; Then statement2 is executed, and the condition is determined again, and the loop is repeated.

4.5.3 Read the Value of the Joystick

Please refer to section 4.3.1 to upload the "joystick" code to the esp32 control board:



Then open the serial port monitor of Arduino IDE, shake the handle left, right, up, down, or press the joystick down, and the serial port will print the joystick value:

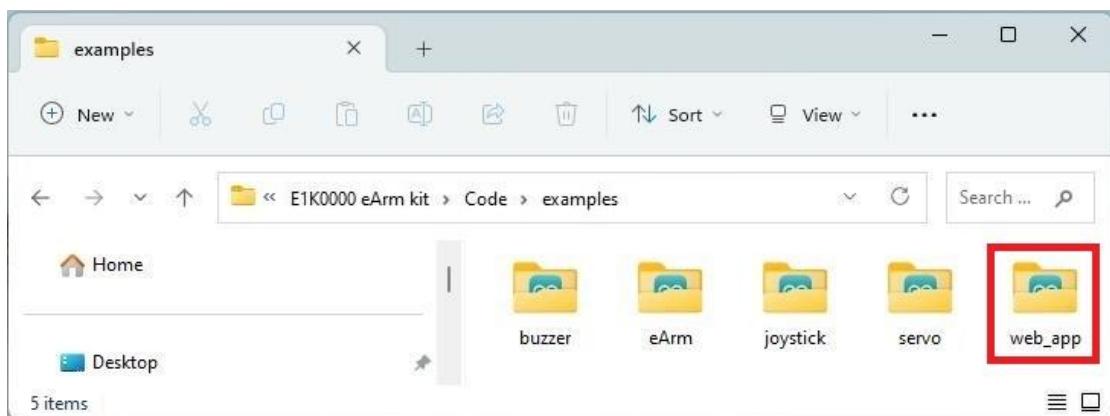


Code Explanation:

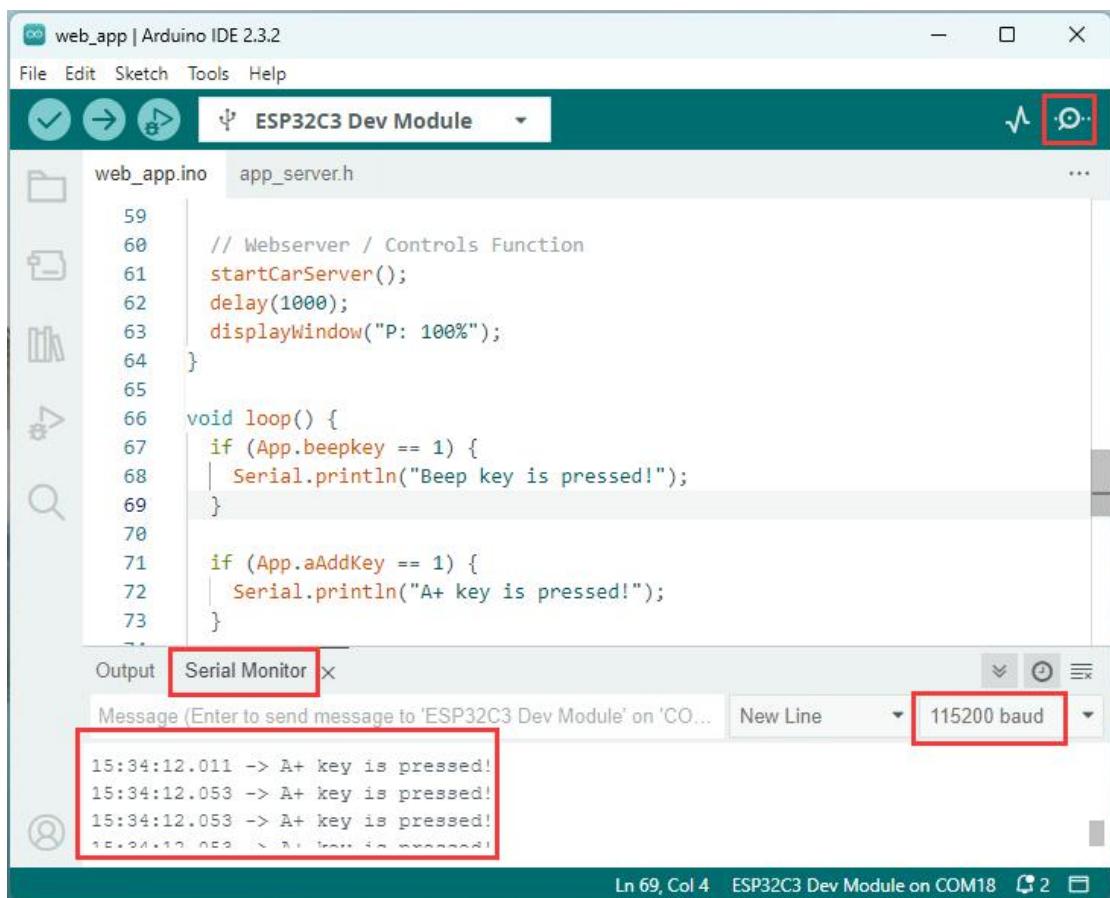
<code>Serial.begin(115200);</code>	Set the serial port baud rate to 115200
<code>pinMode(left_joystick_KeyPin, INPUT);</code>	Set the key pin of the left joystick to "left_joystick_KeyPin" and the mode to output.
<code>pinMode(right_joystick_KeyPin, INPUT_PULLUP);</code>	Set the key pin of the right joystick to "right_joystick_KeyPin", the mode to output, and with a pull-up resistor.
<code>Serial.print("Left_joystick_LeftRight:");</code>	The serial port prints a string without line breaks.
<code>Serial.println(analogRead(left_joystick_Lef tRightPin));</code>	The serial port prints analog values and wraps.
<code>analogRead(left_joystick_LeftRightPin);</code>	Read the value of the left joystick shaking left and right.
<code>delay(1000);</code>	Delay 1000 milliseconds.

4.5.4 Read the Value of the Web APP

Please refer to section 4.3.1 and upload the "web_app" code to the esp32 control board:



Refer to "[Section 3.3 Control the eArm with Web App](#)" to connect "eArm" to wifi and use the web app, then open the serial monitor of Arduino IDE and click the button of the web app. The serial monitor will print the string corresponding to the button:



Code Explanation:

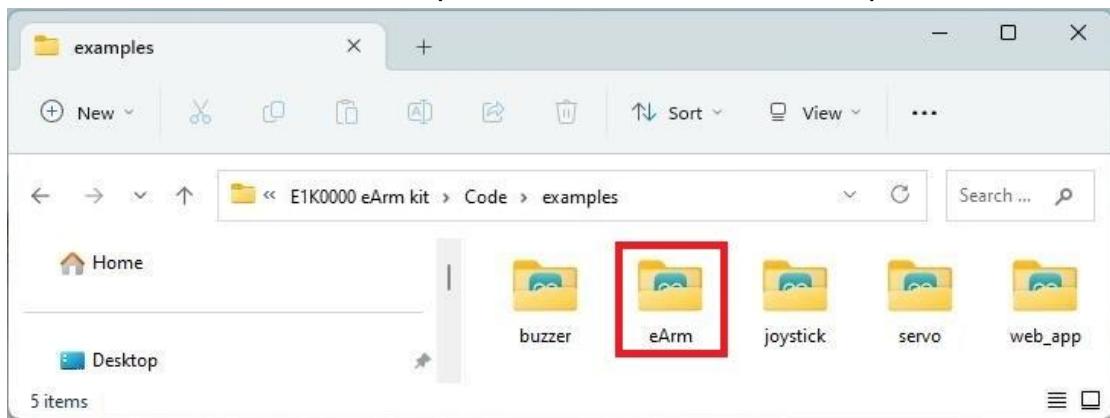
#include "app_server.h"	Include the "app_server.h" header file from the same folder.
bool ap = 1;	Define a Boolean variable "ap" and assign it a value of 1.
const char* ssid = "eArm";	Define a constant character pointer and assign a value to it.
int channel = 11;	Define an integer variable "channel" and assign it a value of 11.
WiFi.mode(WIFI_STA);	Set Wi-Fi to "station" mode.
WiFi.begin(ssid, password);	Set the Wi-Fi name and password, and start Wi-Fi.
while(condition) {...}	While conditional loop statement, if the condition is true, the loop executes the statements in the curly braces; otherwise, the loop is interrupted.
WiFi.status();	Read the wifi connection status.
WiFi.localIP();	Read the wifi IP address in "station" mode.

<code>WiFi.mode(WIFI_AP);</code>	Set the wifi to AP mode.
<code>WiFi.softAP(ssid, password, channel, hidden, maxconnection);</code>	In AP mode, set the wifi name, password, channel, whether to hide the name, and the maximum number of connections.
<code>WiFi.softAPIP();</code>	Read the IP address of the wifi in AP mode.
<code>startCarServer();</code>	Start the web server.
<code>displayWindow("P: 100%");</code>	Display the string on the front end of the web page.
<code>if(condition){ ... }</code>	If the condition is true, the statements in the curly braces are executed; otherwise, they are not executed.
<code>App.beepkey</code>	The state variable value of the buzzer button on the front end of the web page.

4.5.5 Using the Robot Arm

This sample code represents the complete source code of our factory firmware. By uploading it to the control board using the method below, you can fully restore the original factory functionality.

Please refer to section 4.3.1 to upload the "eArm" code to the esp32 control board:



After uploading the eArm code, you can now control this robotic arm as described in the previous Chapter 3: "Using the Robot Arm.

Code Explanation:

<code>TaskHandle_t TASK_HandleOne = NULL;</code>	Create a thread handle.
<code>xTaskCreate(...);</code>	Initialize the thread.
<code>void TASK_ONE(void* param) { ... }</code>	Thread function, which runs in parallel with the loop() function.

5

Factory Reset Function

5.1 Firmware

Providing the device's "factory reset program" directly to users in the form of a firmware file (e.g., .bin or .hex), allowing them to flash it into the device without setting up an Arduino development environment, thereby completing the restoration process.

1. Issues with the Traditional Approach

Typically, restoring a device via Arduino requires users to:

- ▶ Install the Arduino IDE or related development tools.
- ▶ Download the source code (.ino file), possibly requiring additional library setups and dependencies.
- ▶ Compile and upload it to the device.

This process has a high barrier for non-technical users and may fail due to environment-related issues.

2. Advantages of Providing Firmware Directly

Simplified User Process:

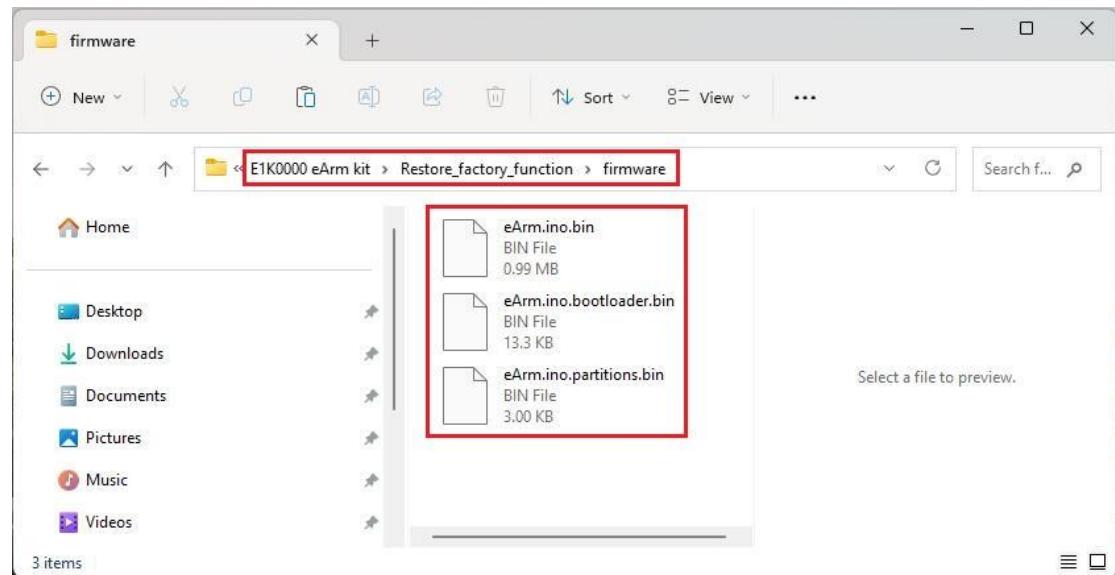
- ▶ Provide a pre-compiled firmware file (e.g., firmware.bin), allowing users to flash it with a simple tool (e.g., a serial flasher) in one step.
- ▶ No need to install Arduino IDE, resolve compilation errors, or manage library conflicts.

Use Cases:

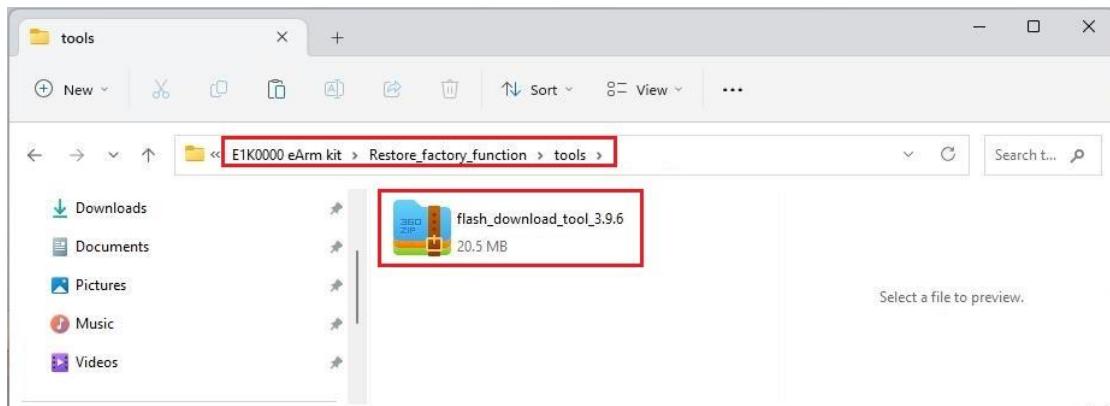
- ▶ Restoring a malfunctioning device to factory settings.
- ▶ Mass-flashing the same firmware to multiple devices (improves efficiency).

3. How to Implement This?

We provide factory firmware files for this product, which are saved in the following folders:



5.2 Burning Tool



You can also download the latest version of the burning tool from the official website:

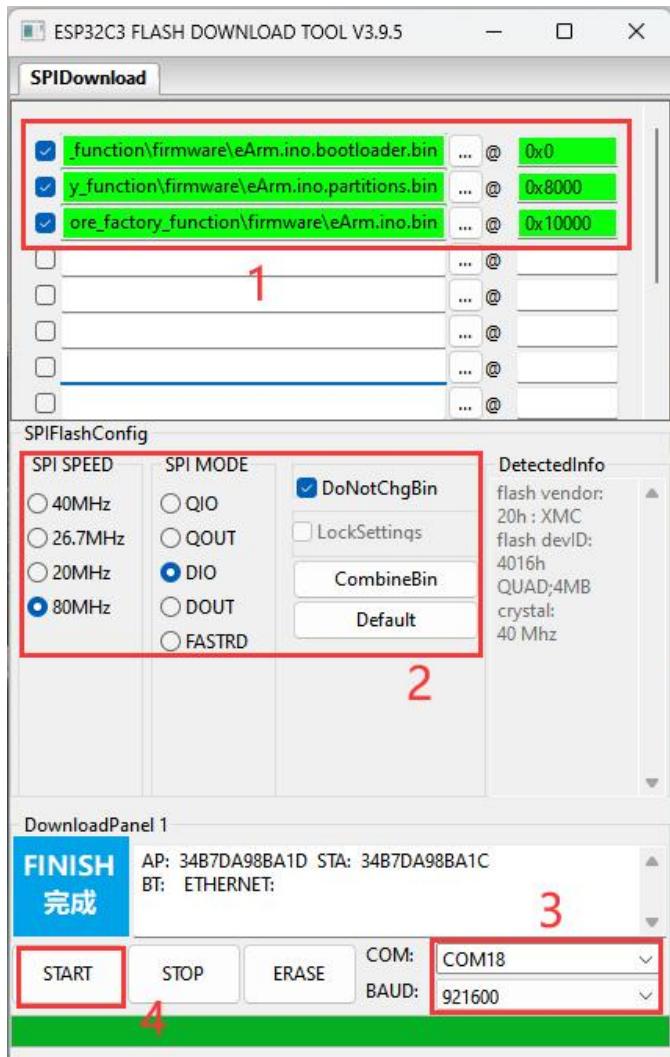
<https://www.espressif.com.cn/zh-hans/support/download/other-tools>

A screenshot of a web page titled 'Flash Download Tools'. At the top right are buttons for 'Expand all' and 'Download selected'. Below is a table with one row. The columns are 'Title', 'Platform', 'Version', 'Release Date', and 'Download'. The row shows 'Flash Download Tools' under 'Title', 'Windows PC' under 'Platform', 'V3.9.7' under 'Version', '2024.06.07' under 'Release Date', and a download button under 'Download'.

5.3 How to Burn Firmware

Use a USB cable to connect the PC and eArm, as shown below:	Start the burning tool:
A diagram of an eArm development board. A blue arrow points from the text 'To the USB port of the PC' to the USB connection point on the board.	A screenshot of the 'DOWN...' window of the Flash Download Tool. It has three dropdown menus: 'ChipType: ESP32-C3', 'WorkMode: Develop', and 'LoadMode: UART'. A red box highlights the 'ChipType' dropdown. Another red box highlights the 'OK' button at the bottom right.

Select the firmware we provided and fill in the burning address correctly:



You must have installed the CH340 driver and turn on the switch of the ESP32 board, otherwise the COM port will not be found!

6

QA

6.1 Unable to recognize USB serial ports

- 👉 Do you use a USB cable with data communication function?
- 👉 Does the USB cable connect well?
- 👉 Is the CH340 USB driver installed?

6.2 Robotic arm doesn't work

- 👉 Whether to turn on the power switch during installation to initialize the angle of the servo.
- 👉 Is the battery electricity sufficient?

6.3 Other problems

- 👉 Please check whether the assembly is correct?
- 👉 Please check whether the battery power is sufficient?
- 👉 Please check whether the battery used in accordance with the specifications?

7

Contact Us

If you encounter any technical issues or wish to share feedback with us, please feel free to contact us

 support@siyeeenove.com

 <https://siyeeenove.com>