

---

# **SunFounder DIY\_Control\_Robot\_Arm\_kit\_for\_Arduino- Rollarm**

**[www.sunfounder.com](http://www.sunfounder.com)**

**Oct 21, 2021**



# CONTENTS

<b>1</b>	<b>About SunFounder</b>	<b>1</b>
<b>2</b>	<b>About This Kit</b>	<b>3</b>
<b>3</b>	<b>Free Support</b>	<b>5</b>
3.1	Introduction . . . . .	5
3.2	Components List . . . . .	7
3.2.1	Structural Plate . . . . .	7
3.2.2	Electronic Components . . . . .	9
3.2.3	Tools . . . . .	10
3.2.4	Self-Provided Components . . . . .	10
3.3	How to control . . . . .	10
3.4	Getting Started with Software . . . . .	10
3.4.1	Downloading Codes . . . . .	10
3.4.2	Install Arduino IDE . . . . .	11
3.5	Assembly . . . . .	17
3.5.1	Install the base of Rollarm . . . . .	17
3.5.2	Install Accessories . . . . .	22
3.5.3	Calibrate the steering gear angle . . . . .	23
3.5.4	Install Servo Rocker Arm . . . . .	26
3.5.5	Install Button . . . . .	38
3.5.6	Install Battery . . . . .	41
3.6	Control the Rollarm . . . . .	42
3.6.1	Manual Control . . . . .	43
3.6.2	PC Control . . . . .	48
3.7	FAQ . . . . .	56
3.7.1	About the Assembly . . . . .	56
3.7.2	About the Arduino code control . . . . .	57
3.7.3	About the Labview software control . . . . .	59
3.8	Thank You . . . . .	61
<b>4</b>	<b>Copyright Notice</b>	<b>63</b>



## **ABOUT SUNFOUNDER**

SunFounder is a technology company focused on Raspberry Pi and Arduino open source community development. Committed to the promotion of open source culture, we strive to bring the fun of electronics making to people all around the world and enable everyone to be a maker. Our products include learning kits, development boards, robots, sensor modules and development tools. In addition to high quality products, SunFounder also offers video tutorials to help you build your own project. If you have interest in open source or making something cool, welcome to join us! Visit [www.sunfounder.com](http://www.sunfounder.com) for more!



## ABOUT THIS KIT

The Rollarm Kit for Arduino is designed for mass hobbyists to learn robot arm control. With the open source MCU Arduino UNO and a servo expansion board, you can control its four axes by the 4 potentiometer on the handle, as well as make them move by operating on your computer. In addition, it can memorize the movements it's made and repeat again and again, making it a great tool for repeated tasks. In this book, you can learn the basics of how a mechanical arm works and how to make one piece by piece. For more information, please go to our website [www.sunfounder.com](http://www.sunfounder.com) and find the tutorial under Learn.



## FREE SUPPORT

If you have any **TECHNICAL** questions, add a topic under Community -> Forum section on our website and we'll reply as soon as possible.

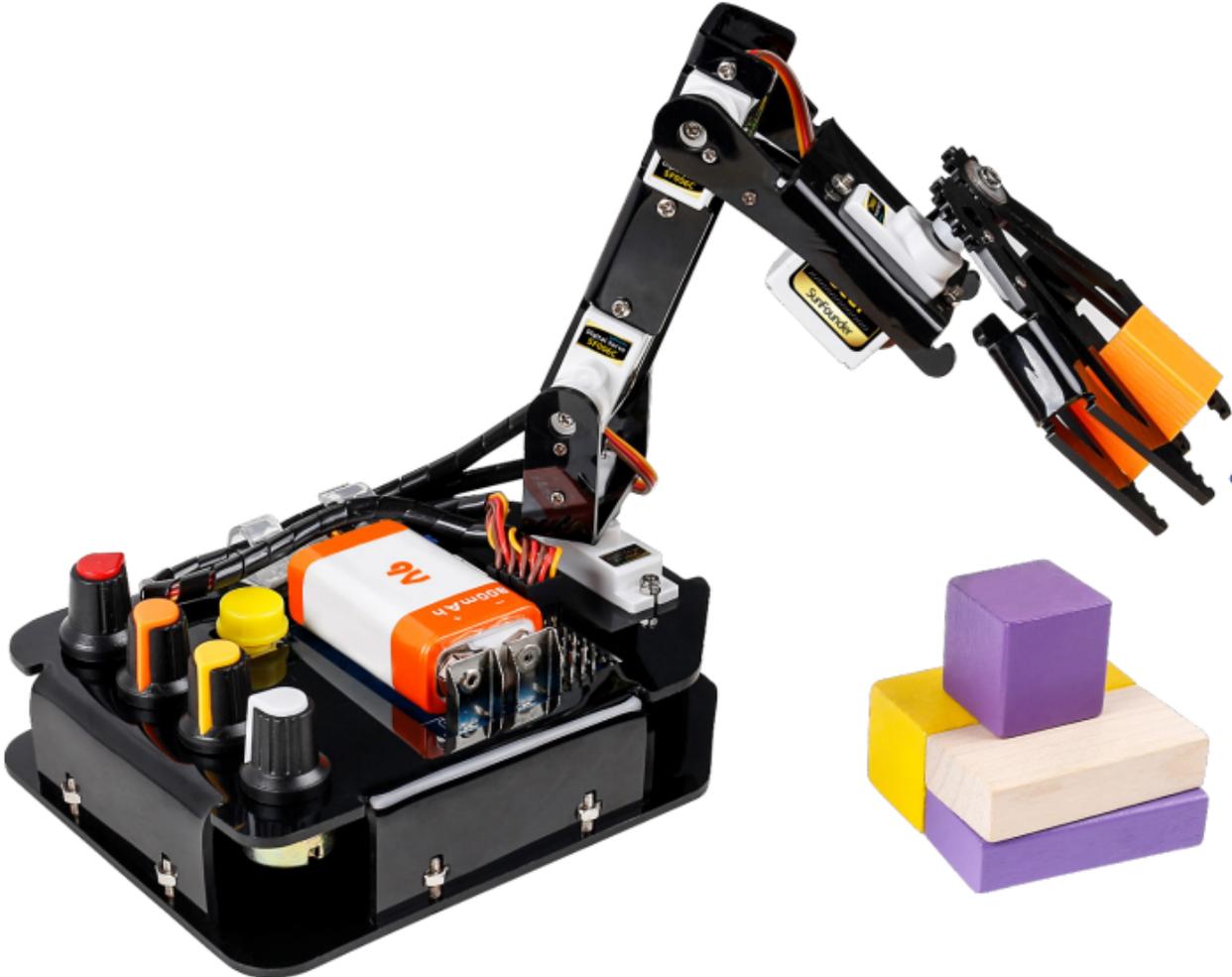
For **NON-TECH** questions like order and shipment issues, please send an email to [service@sunfounder.com](mailto:service@sunfounder.com). You're also welcomed to share your projects on Forum.

### Content

## 3.1 Introduction

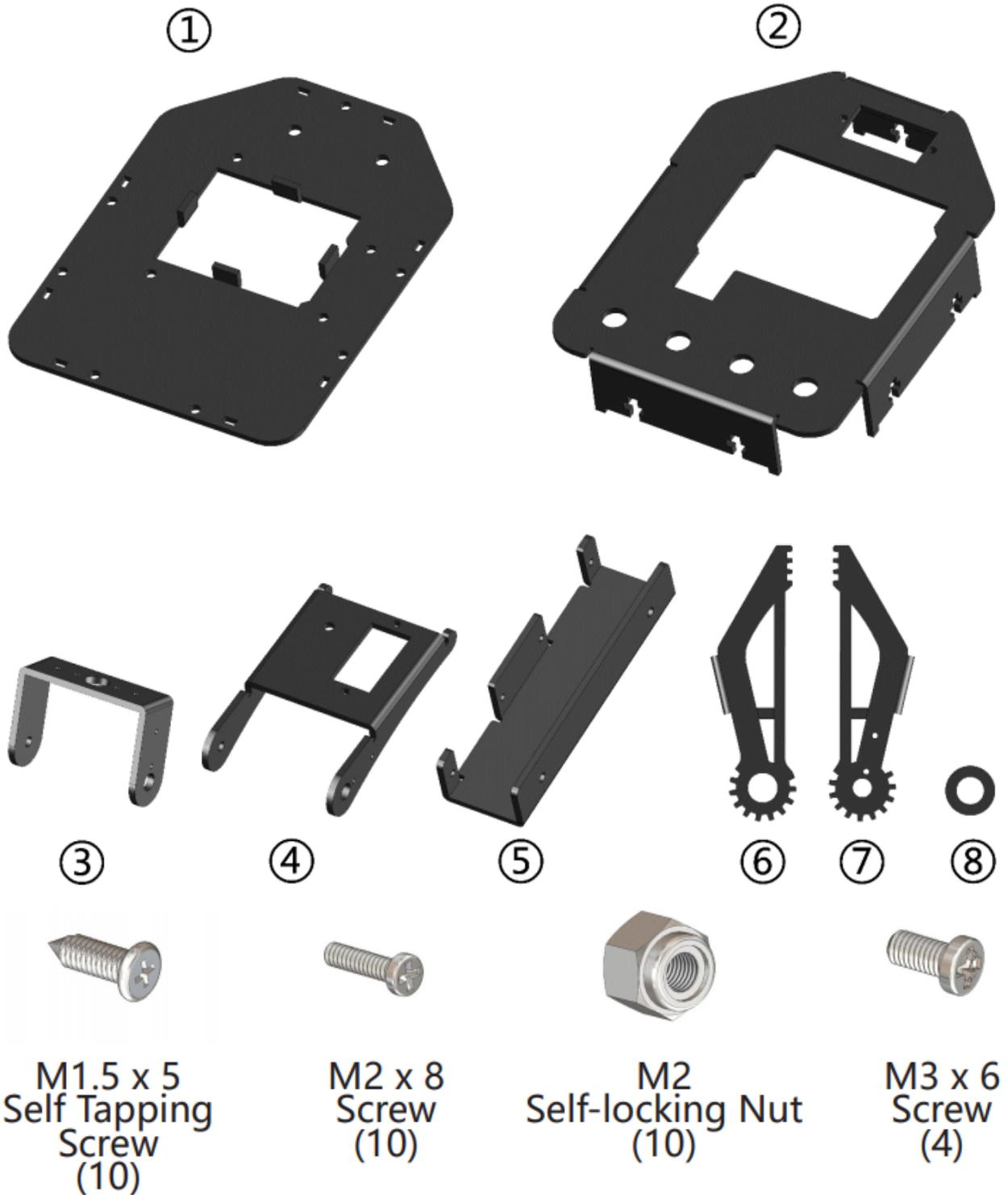
The Rollarm Kit is an interesting and useful learning tool for Arduino and robot hobbyists. With the structural plate and code based on Arduino, it enables users to learn programming from easy to difficult, control the mechanical arm freely and perform various fun operations!

This fun mechanical arm consists of 4 axes, each controlled by a servo. Powered by a 9V battery, the control systems is composed of SunFounder Uno board, servo extension board with remote control. The kit includes all necessary components like structural plate, circuit boards, and connector parts. For your better learning, installation and debugging video tutorials are provided on youtube. Also you can download the user manual which elaborates on the installation procedures and program explanation. With these resources, you can quickly and effortlessly make creative projects with the Rollarm. Now let's go to get the fun!



## 3.2 Components List

### 3.2.1 Structural Plate





M3 x 10  
Screw  
(10)



M2.5 x 12  
Screw  
(6)



M2.5  
Nut  
(6)



M3  
Nut  
(12)



M3  
Self-locking Nut  
(4)



M7  
Thin Nut  
(4)



3\*10\*1  
Washer  
(1)

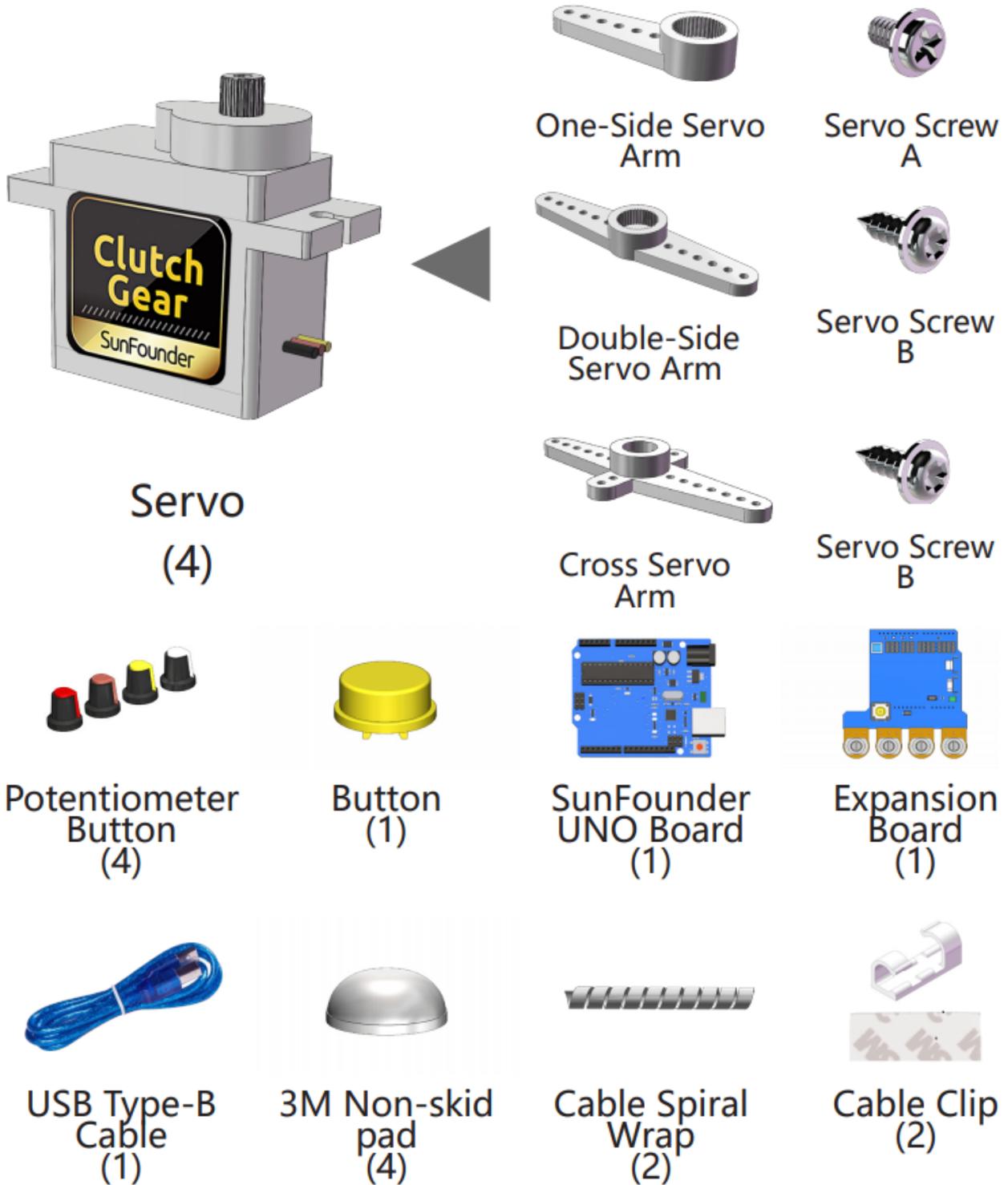


M3\*12  
Aluminum Tube  
(1)



φ3\*φ8\*2.5  
Band Edge  
Bearing  
(1)

### 3.2.2 Electronic Components



### 3.2.3 Tools



M2.5  
Screwdriver  
(1)



M3  
Screwdriver  
(1)



Tweezers  
(1)



Spanner  
(1)

### 3.2.4 Self-Provided Components



9V  
Battery  
(1)

## 3.3 How to control

1. Install Arduino IDE
2. Assemble the Rollarm (Remember to **configure the servos** before assembly. Pay attention to the detailed operations)
3. Control with handle: **download** the Rollarm package and run the program (Indispensable step before operating the Rollarm)
4. Control with Labview: **download** the Labview to your PC and install for control (download is a must-do before subsequent operating) Let's get started. Enjoy!

## 3.4 Getting Started with Software

### 3.4.1 Downloading Codes

Go to **Learn->Robot Kit->DIY Control Robot Arm kit for Arduino-Rollarm** and download the package

► [Get tutorials /Robot Kit/ DIY Control Robot Arm kit for Arduino-Rollarm](#)

SunFounder:

If you have any questions, please send an email to [support@sunfounder.com](mailto:support@sunfounder.com).

For more tutorials, back to the Learn section. To share your cool works, welcome post in our Forum section.



### 3.4.2 Install Arduino IDE

Arduino is an open source platform that applies simple software and hardware. You can get it in a short even when you know little of it. It provides an integrated development environment (IDE) for code editing and compiling, compatible with multiple control boards. So you can just download the Arduino IDE, upload the sketches (i.e. the code files) to the board, and then you can see experimental phenomena. For more information, refer to <http://www.arduino.cc>.

The code in this kit is written based on Arduino, so you need to install the IDE first. Skip it if you have done this.

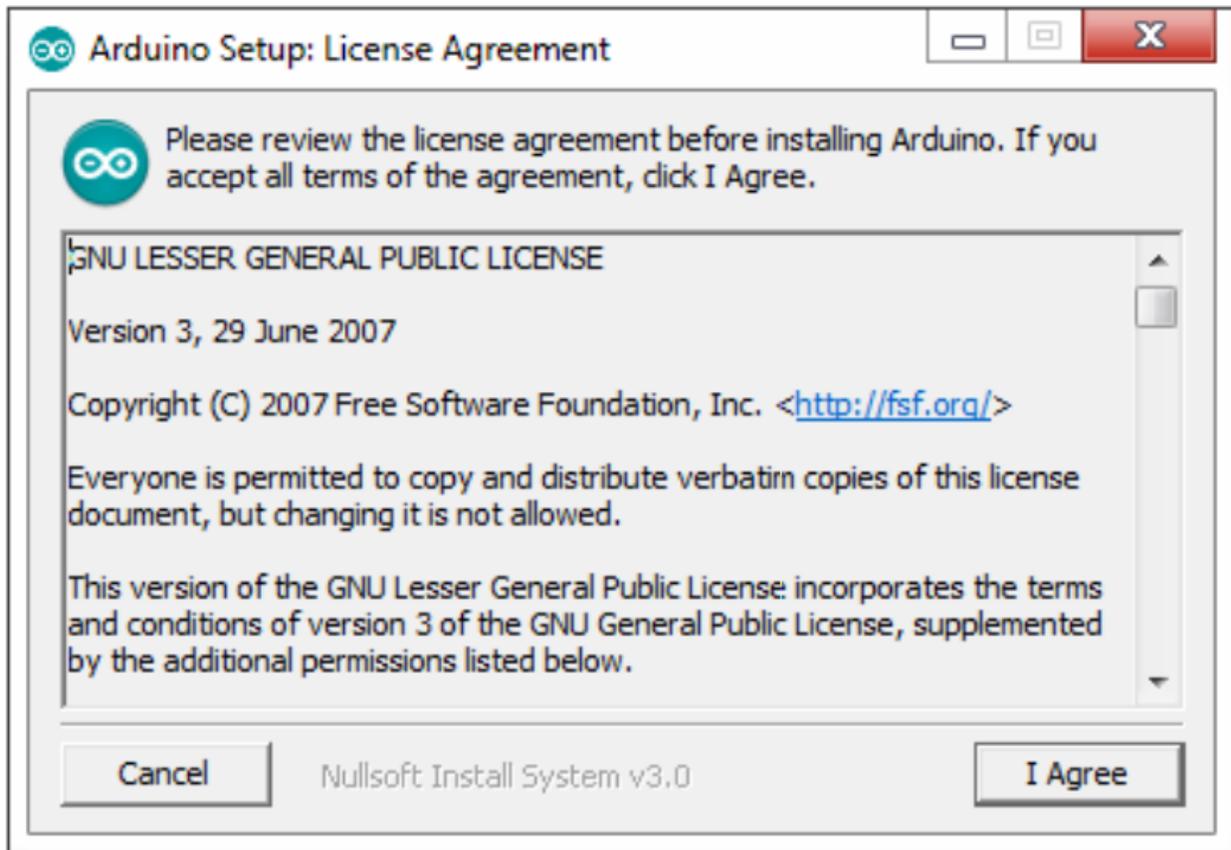
**Step 1:** Go to the [arduino.cc](http://arduino.cc) website and click Download. On the page, check the software list on the right side under Download the Arduino Software.



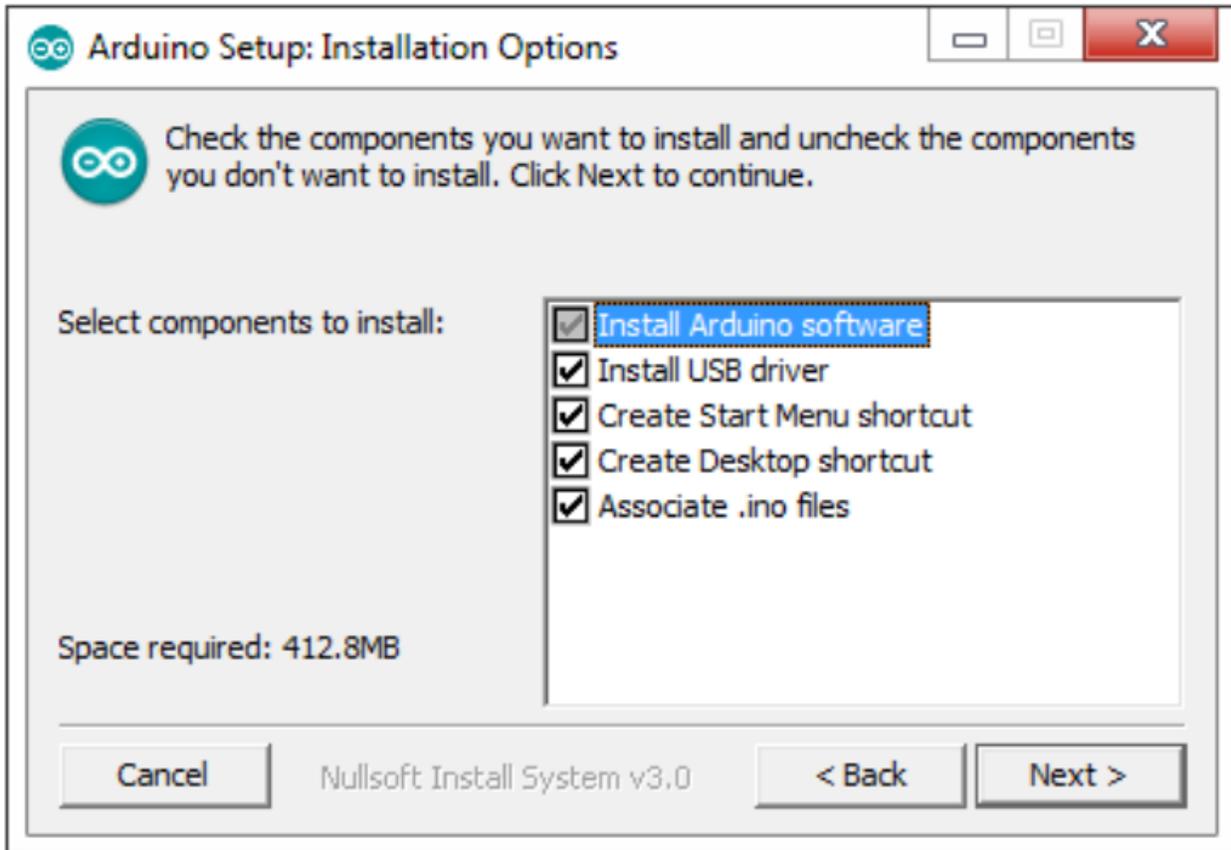
Find the one that suits your operation system and click to download. There are two versions of Arduino for Windows:

Installer or ZIP file. You're recommended to download the former.

**Step 2:** Double click the exe. file and the following window will show up. Click I Agree. The following interface will show up.

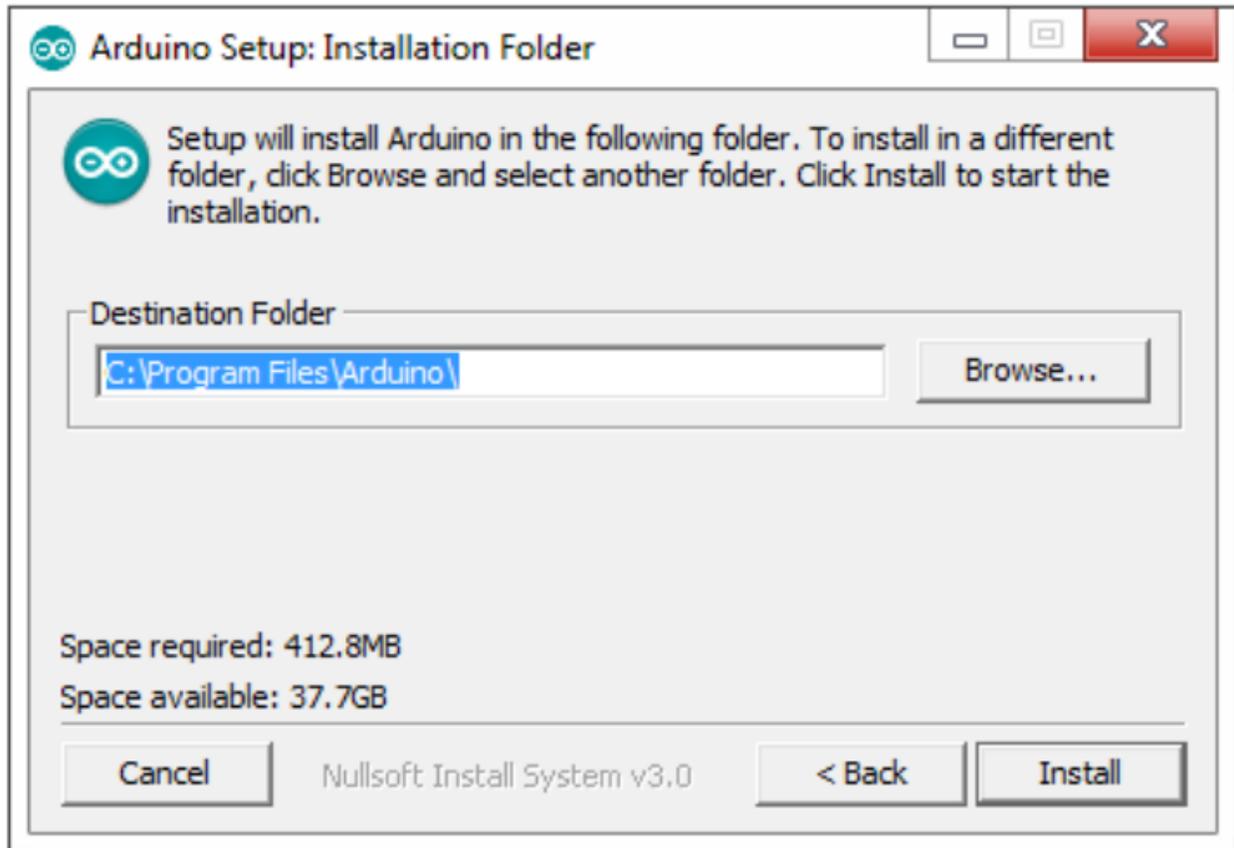


Choose Next.



Click **Browse** to choose the installation path or enter a directory at the Destination Folder.

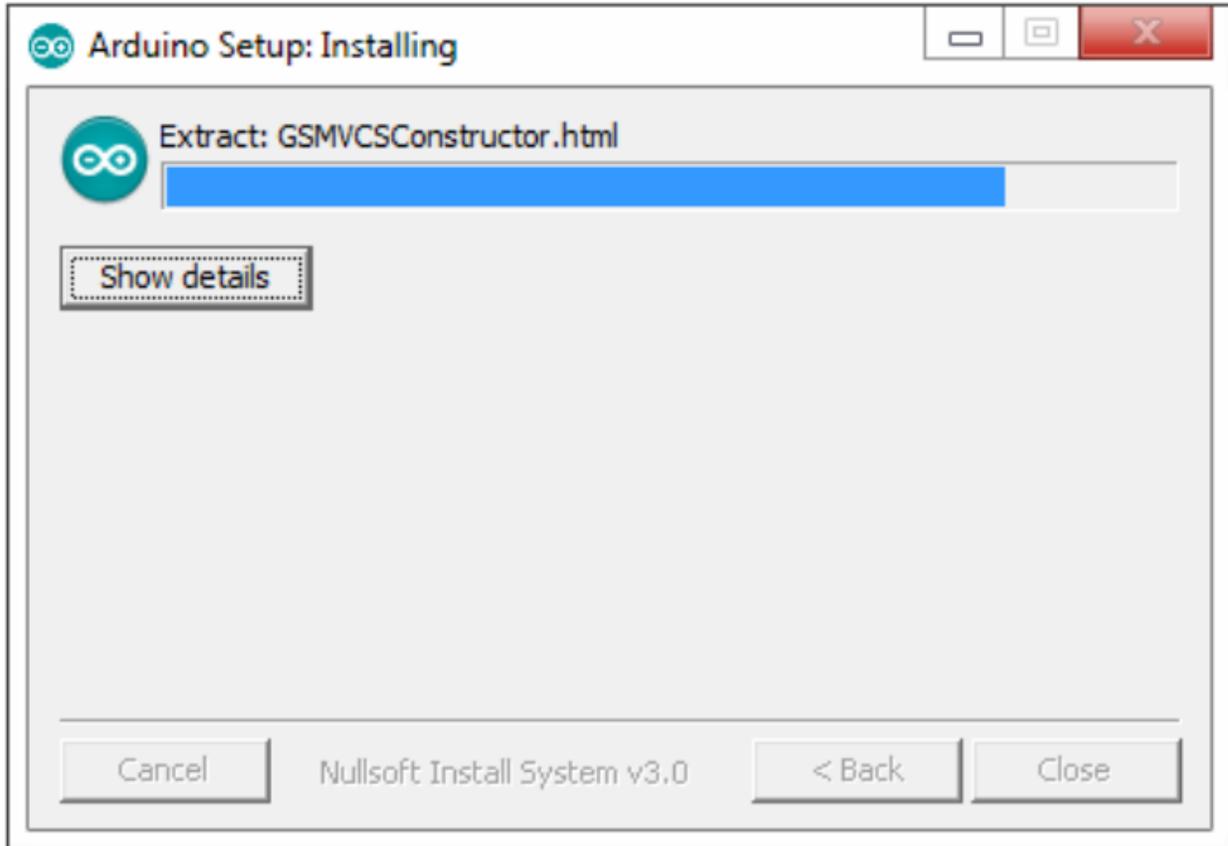
Click **Install**.



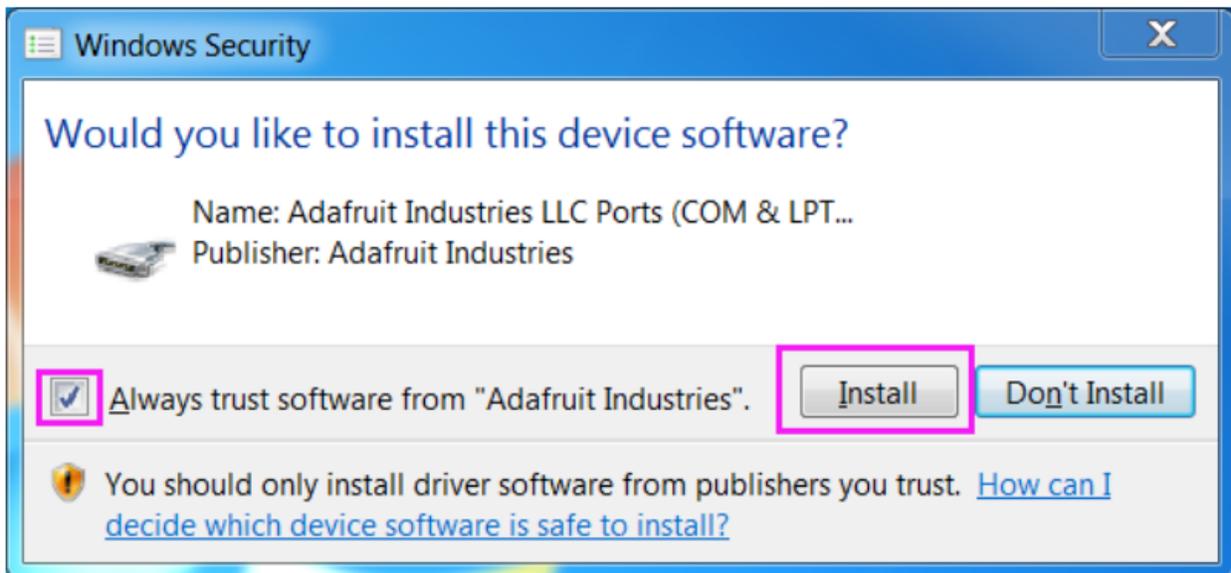
The following interface will show up.

**Note:** After the installing progress bar goes to the end, the Close button may be enabled for some PC. Just click it to complete the installation.

---



Then a prompt appears. Select Always trust software for “Adafruit Industries” and click Install.



Select “Always trust software for Arduino srl” and click **Install**.



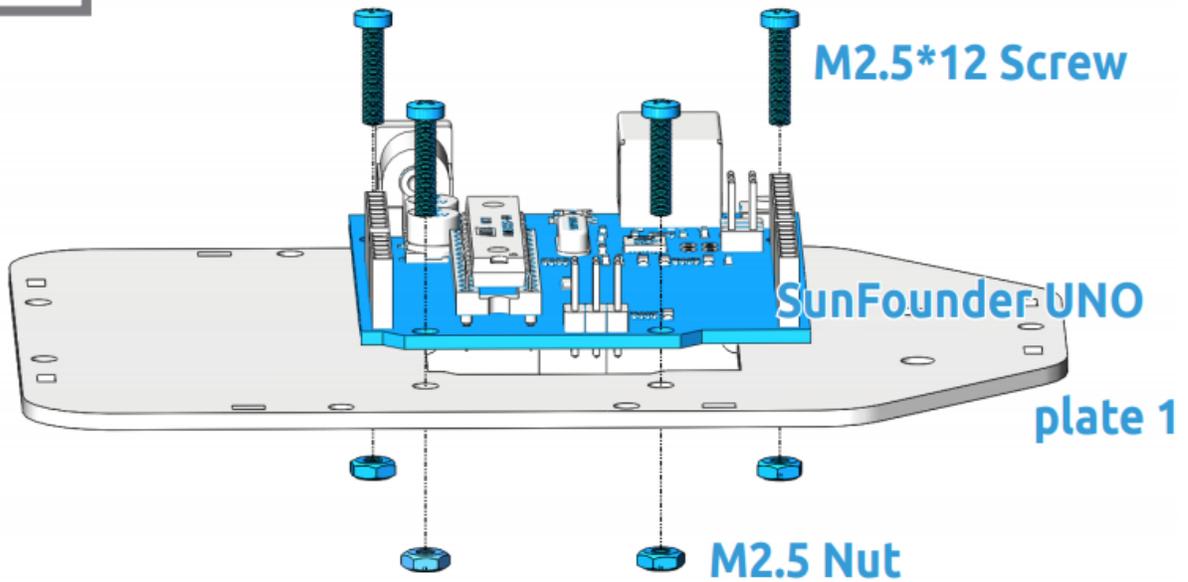
After the installation is done, click **Close**. Then an Arduino icon will appear on the desktop:



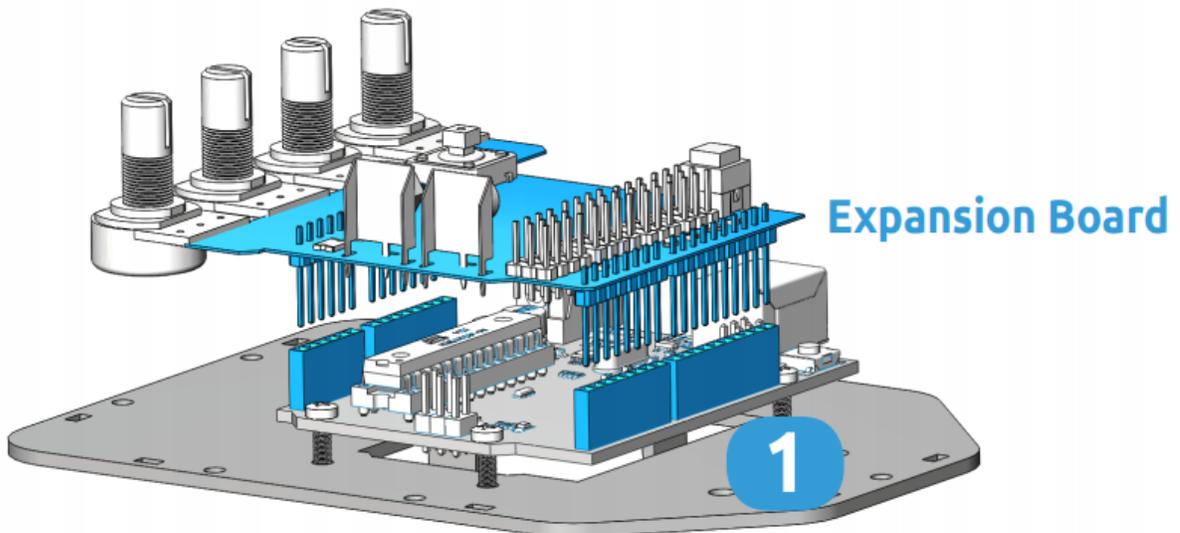
## 3.5 Assembly

### 3.5.1 Install the base of Rollarm

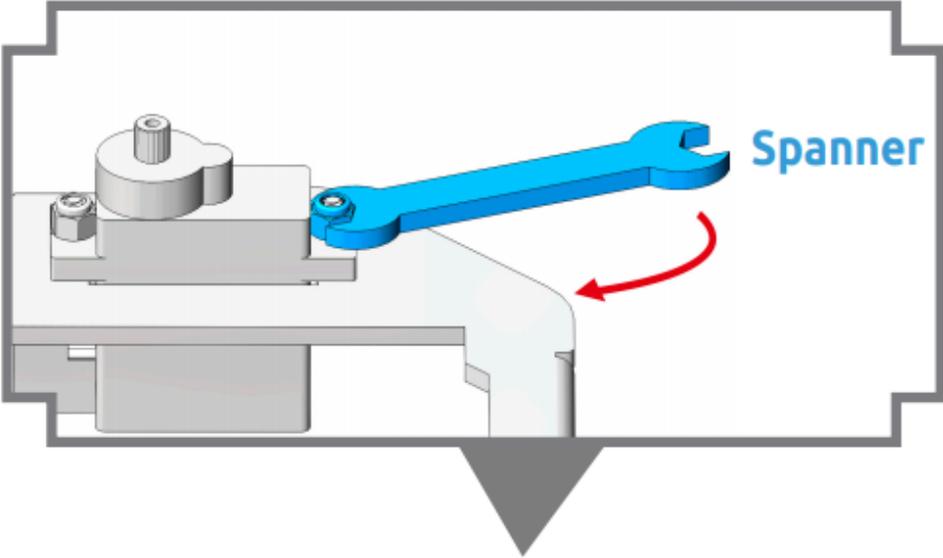
#### 1 Complete UNO Board + plate1

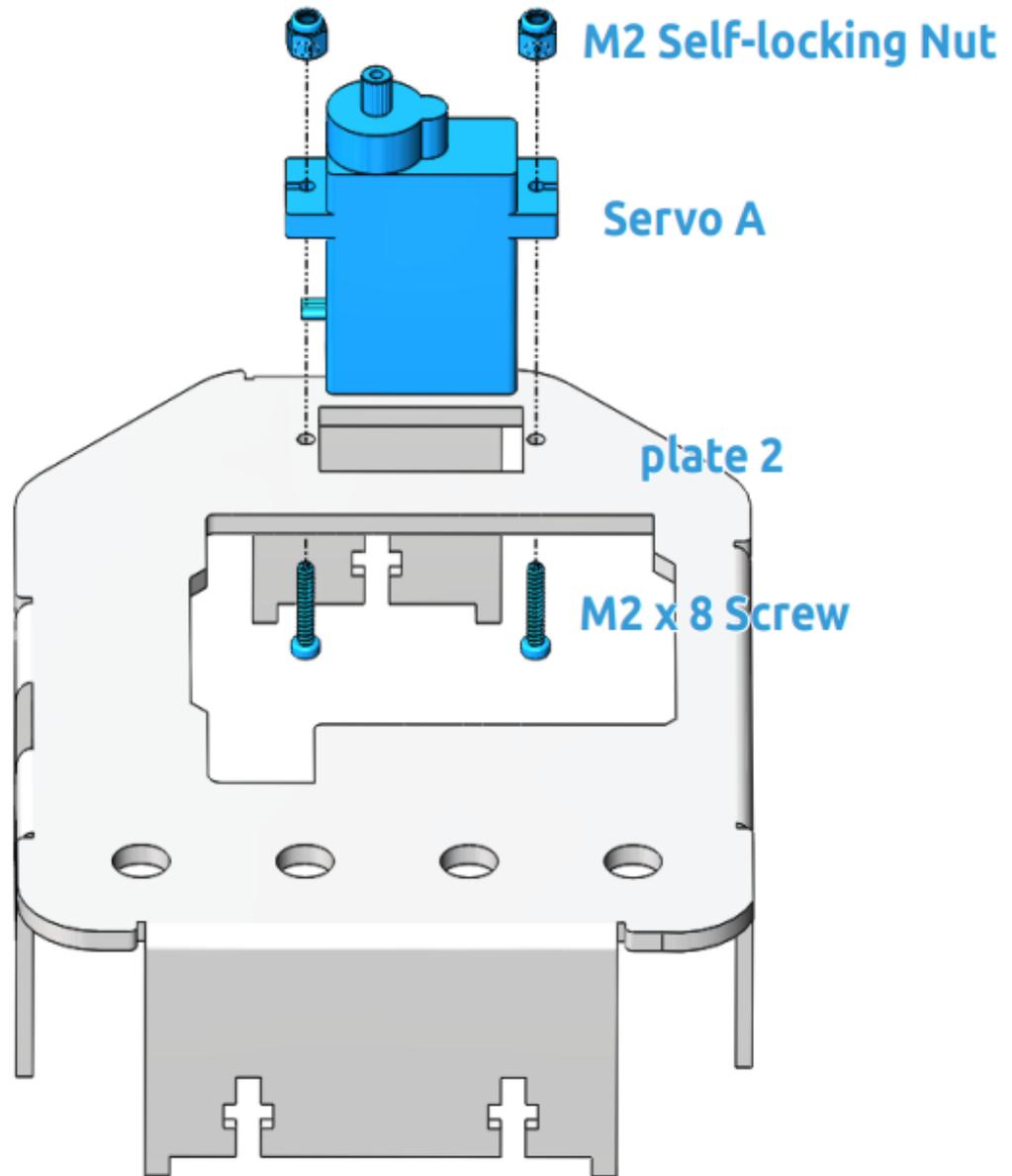


#### 2 Complete Expansion Board



**3** Complete Servo A + plate 2



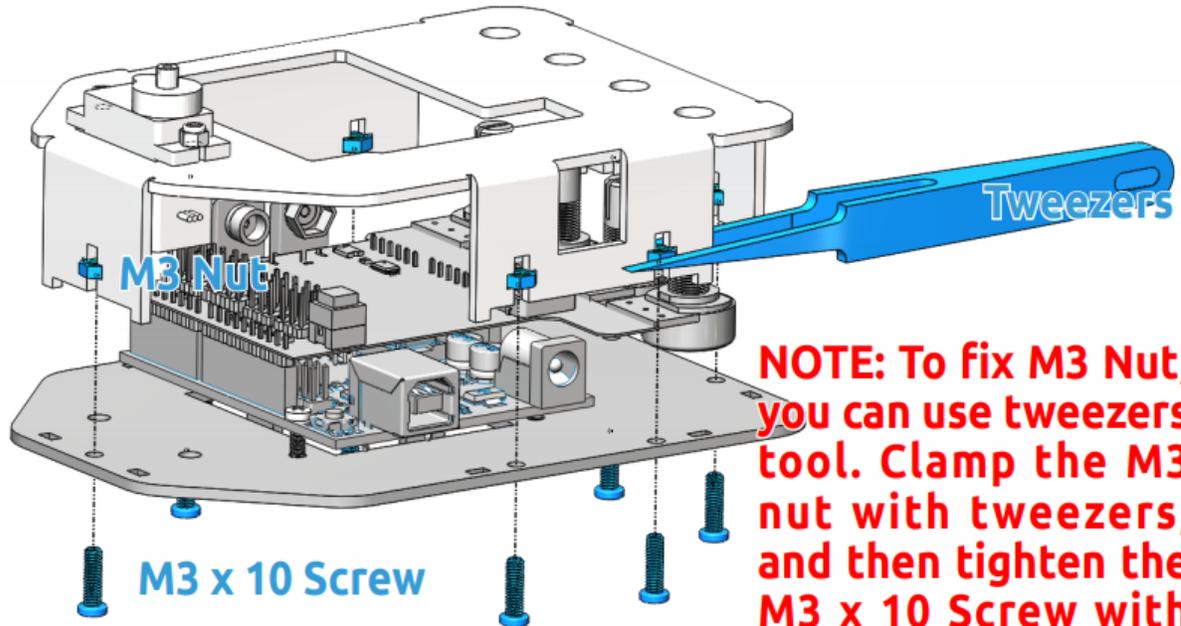


---

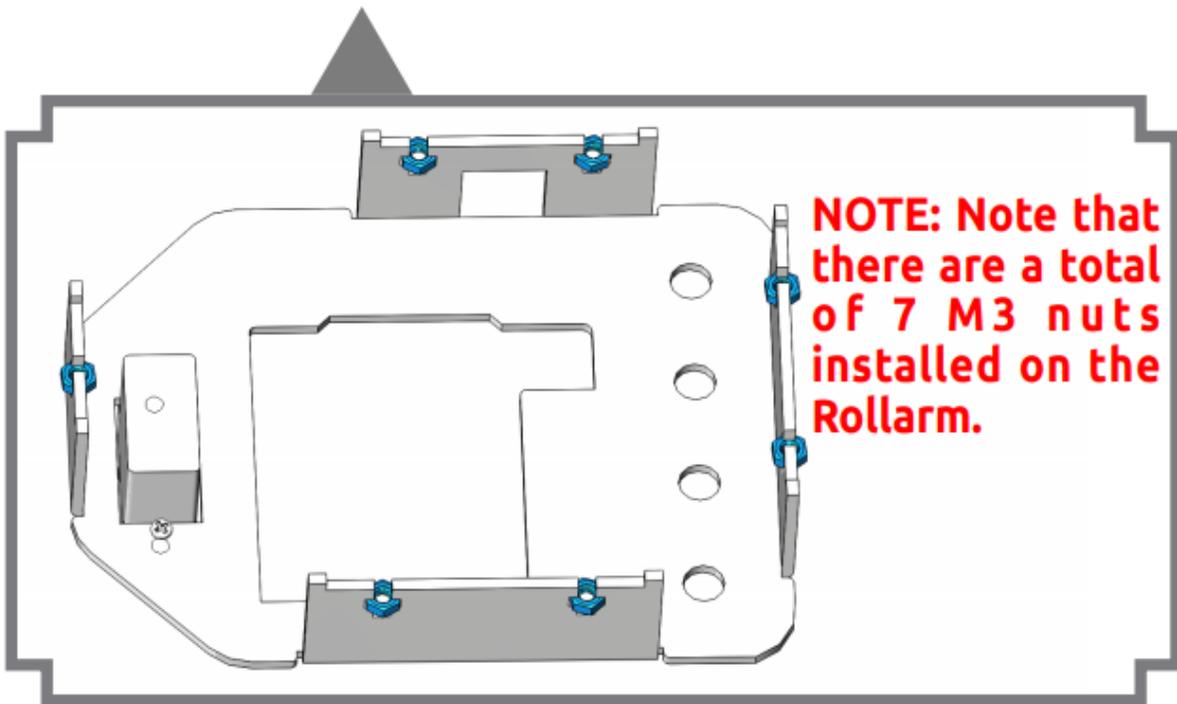
**Note:** You can turn the wrench clockwise to tighten the selflocking nut.

---

# 4 Complete plate 1 + plate 2

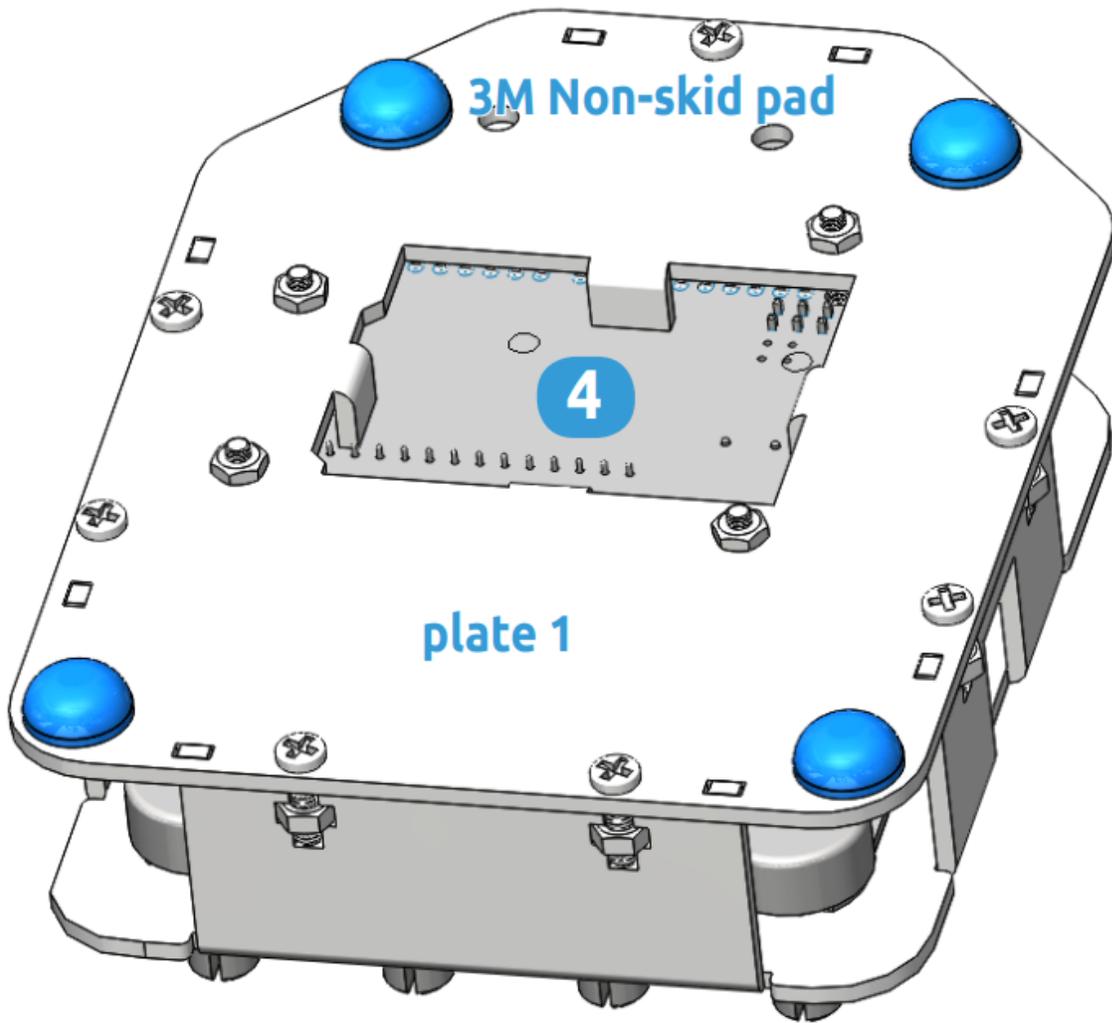


**NOTE:** To fix M3 Nut, you can use tweezers tool. Clamp the M3 nut with tweezers, and then tighten the M3 x 10 Screw with an M3 screwdriver.



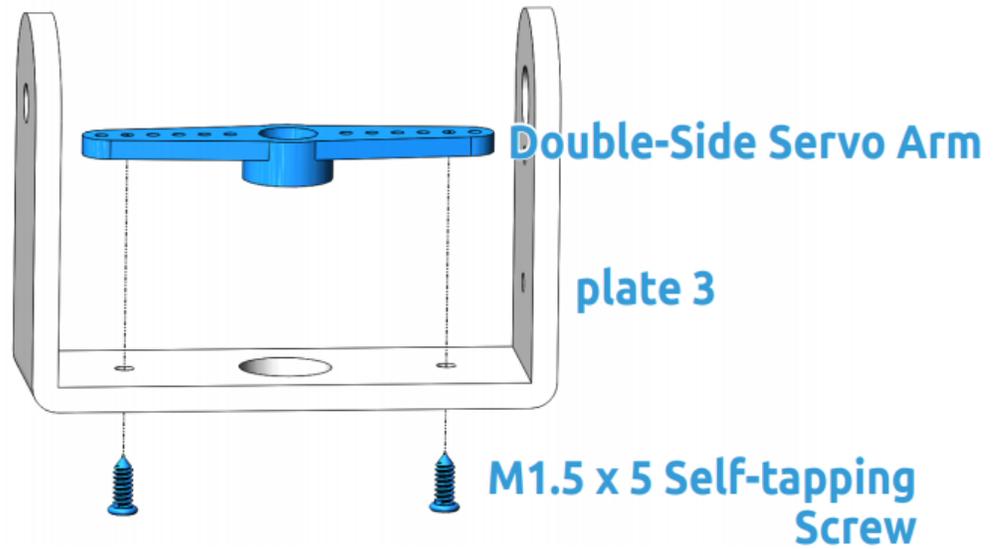
**NOTE:** Note that there are a total of 7 M3 nuts installed on the Rollarm.

**5** Complete 3M Non-skid pad

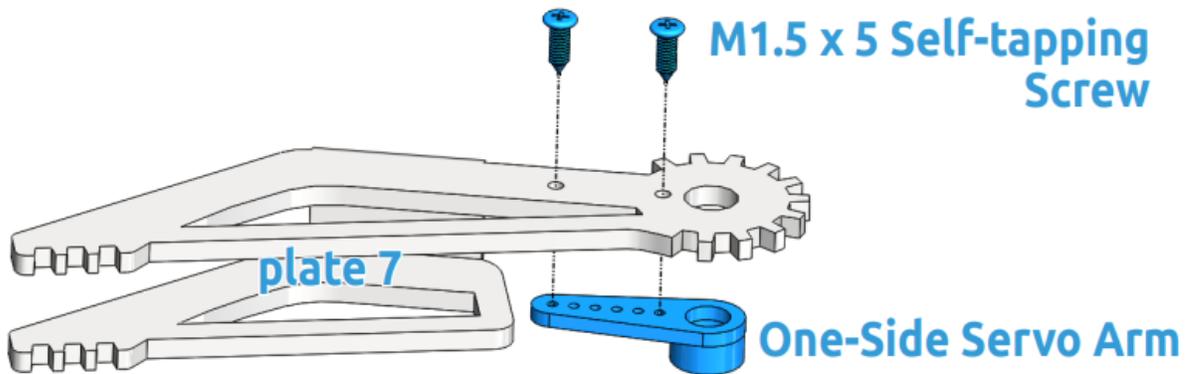


### 3.5.2 Install Accessories

#### 1 Build accessories A



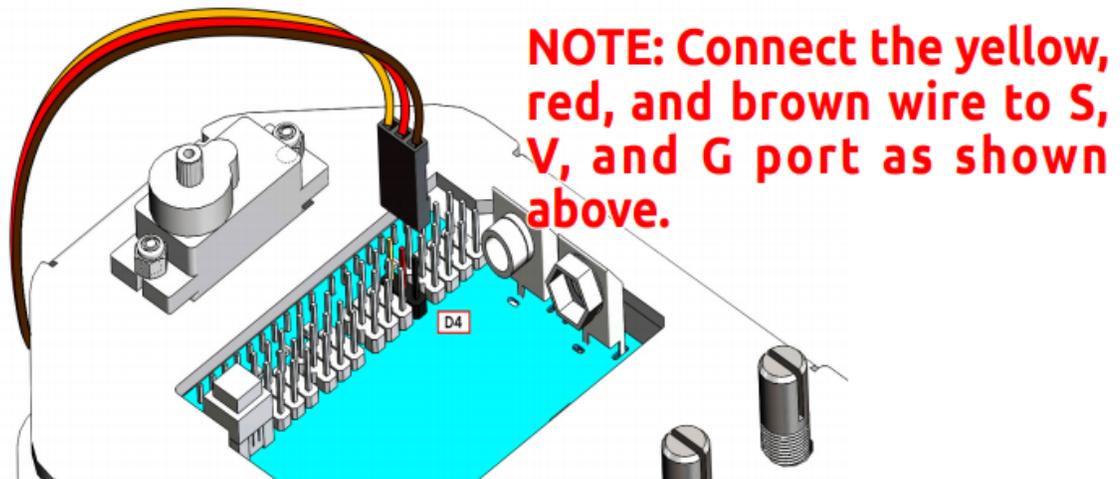
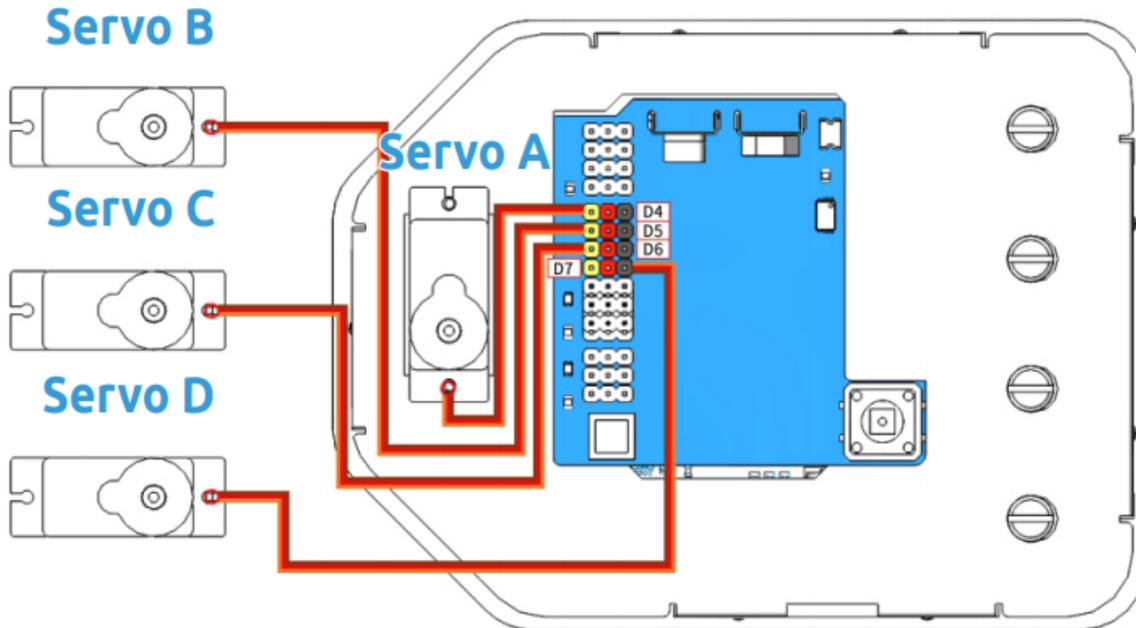
#### 2 Build accessories B



### 3.5.3 Calibrate the steering gear angle

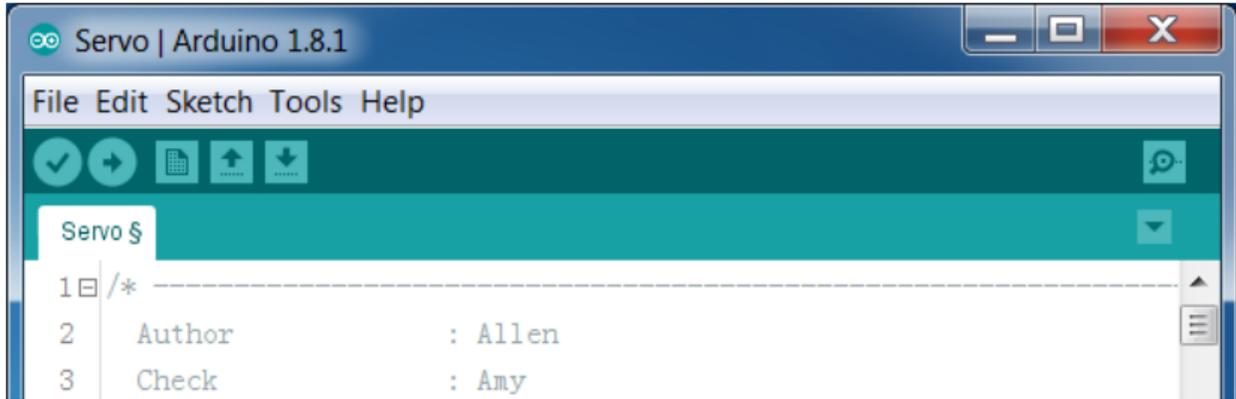
**Warning:** Before installing the rocker arms for each servo, you need to adjust the servo. If you skip this step and finish assembling and power the product, the servo may appear to block or be damaged, you may even need to disassemble the robot from this step to reassemble it!

**Step 1:** Prepare four white servos, and then insert the Servo A wires into D4, Insert the Servo B wires into D5, Insert the Servo C wires into D6, Insert the Servo D wires into D7.

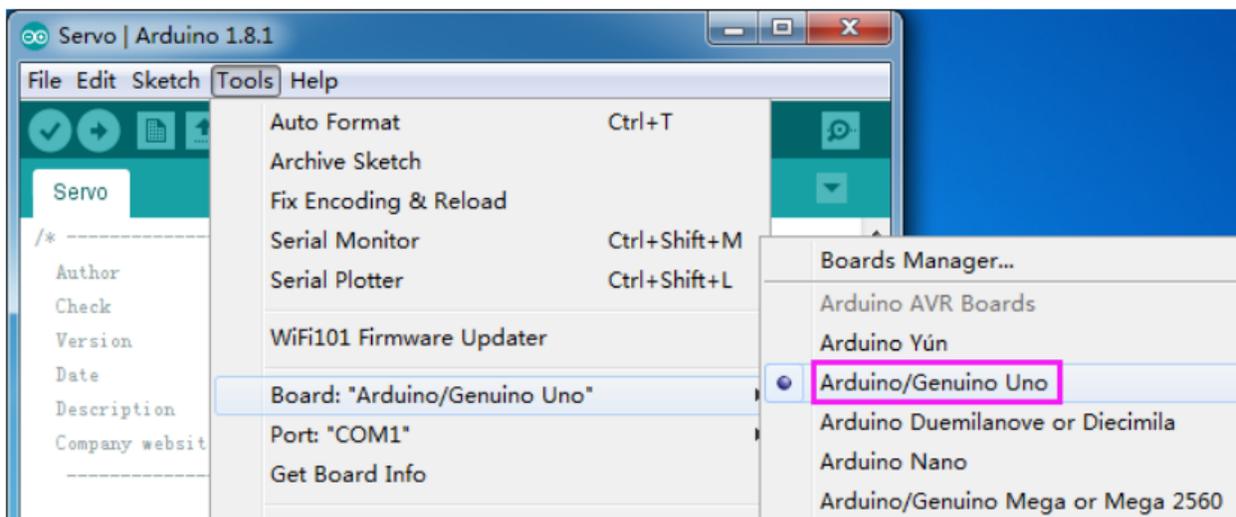


**Step 2:** connect the Servo Control Board to the PC via the USB Type-B Cable, and the PC will automatically install the driver.

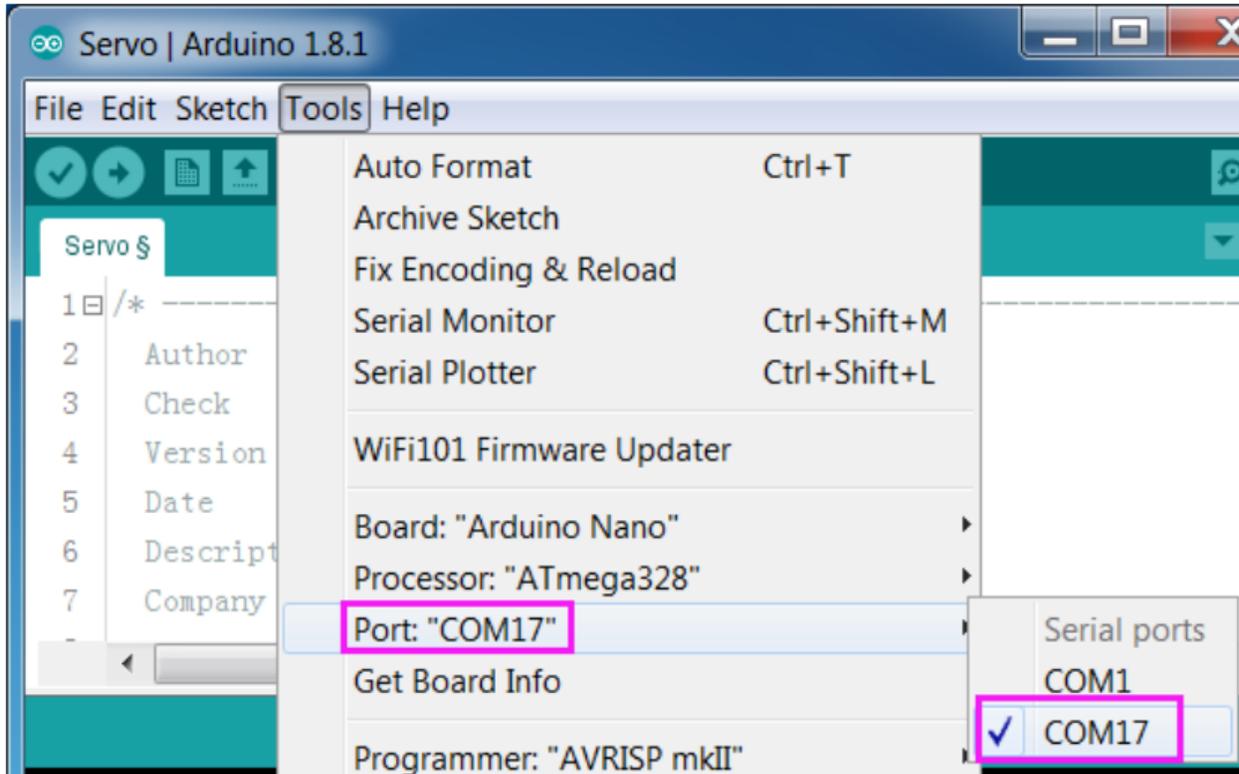
**Step 3:** Go to the folder DIY Control Robot Arm kit for Arduino-Rollarm/Arduino Code/Servo and open the file Servo.ino.



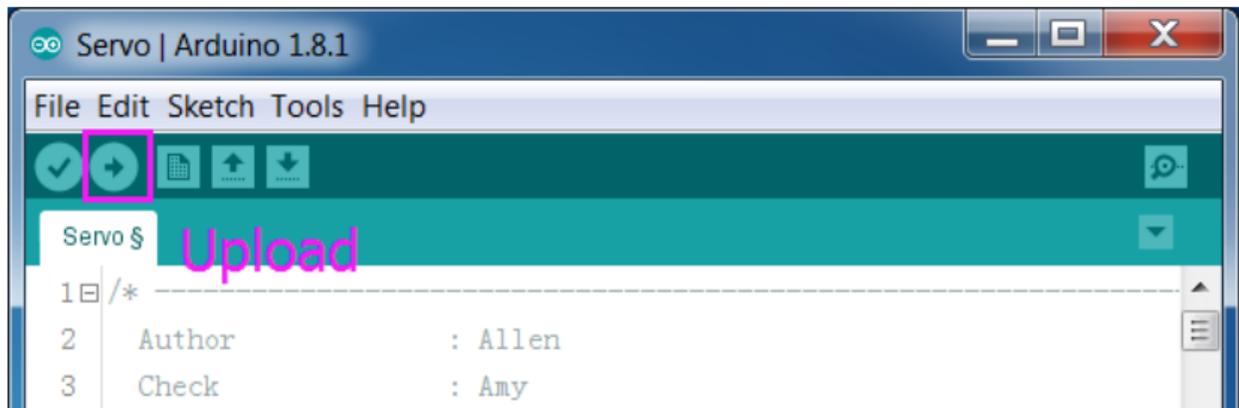
**Step 4:** Select the **Board**.



and **Port**



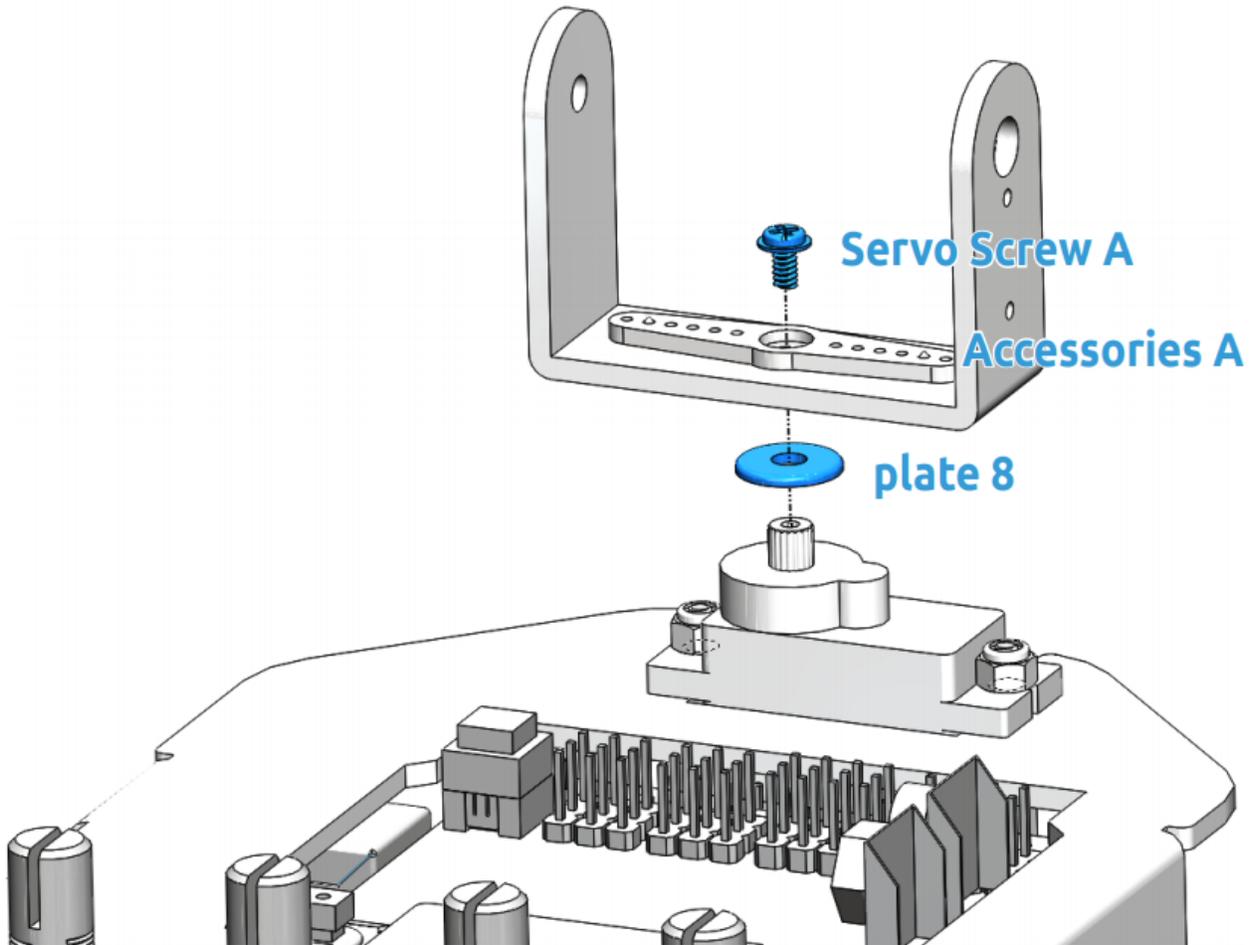
**Step 5:** Click Upload.



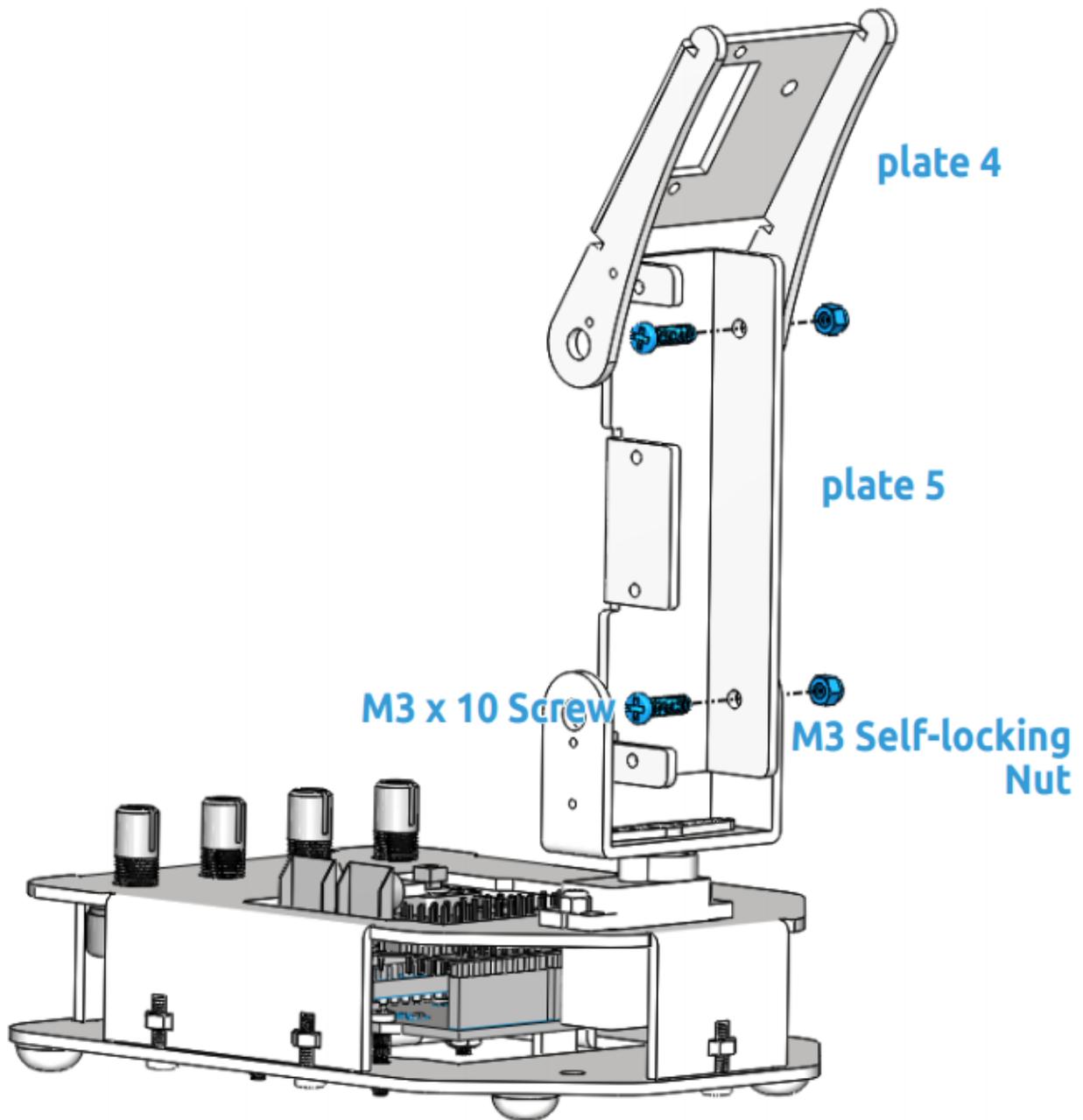
After the upload is completed successfully, keep the USB cable connecting. You may hear the sound of gear moving (or may not, if the servo shaft happens to be at 90 degrees at the beginning; but you GENTLY spin the rocker arm and you'll find it's unmovable). So now the servo is adjusted to 90 degrees.

### 3.5.4 Install Servo Rocker Arm

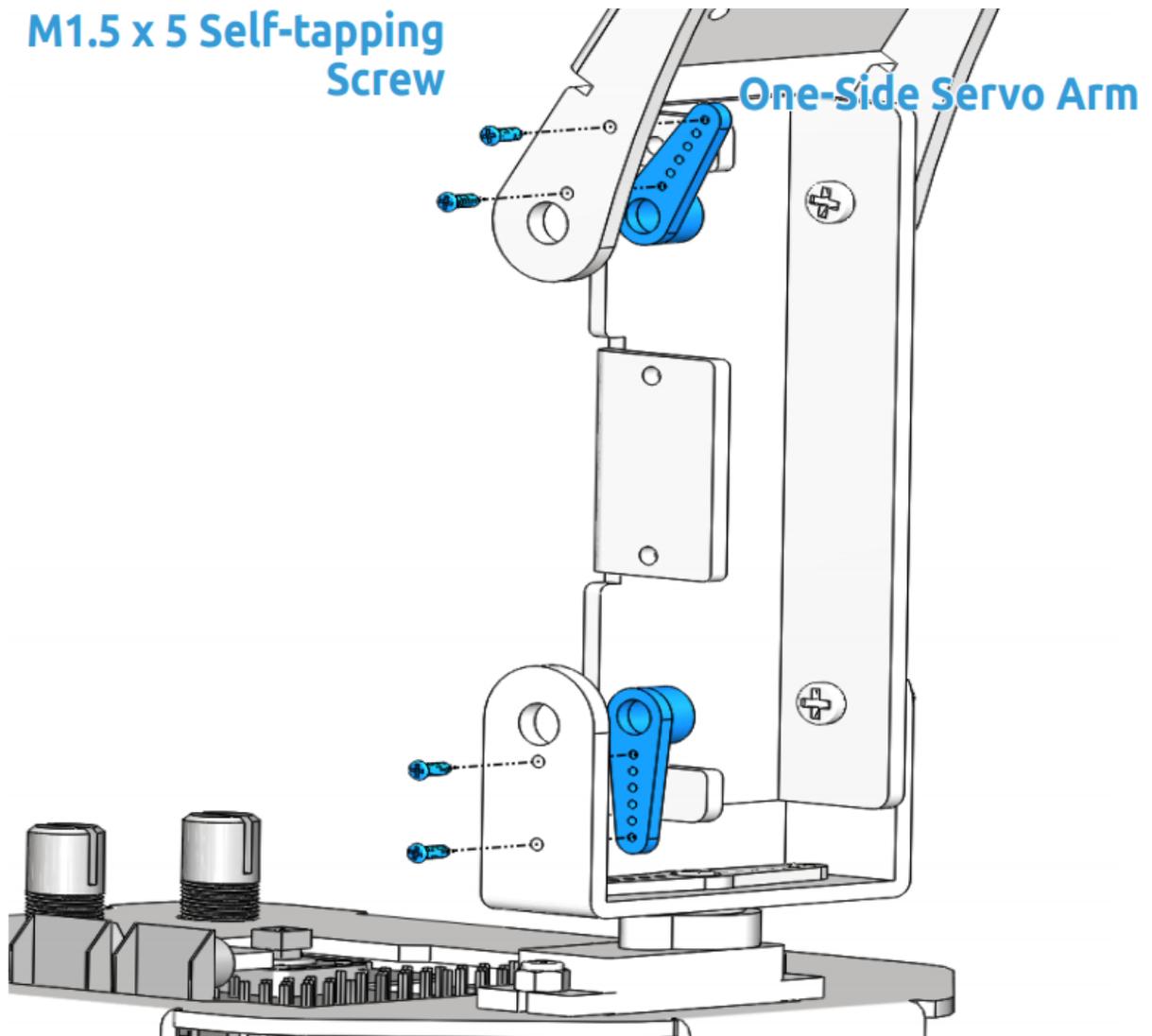
## 1 Complete Accessories A



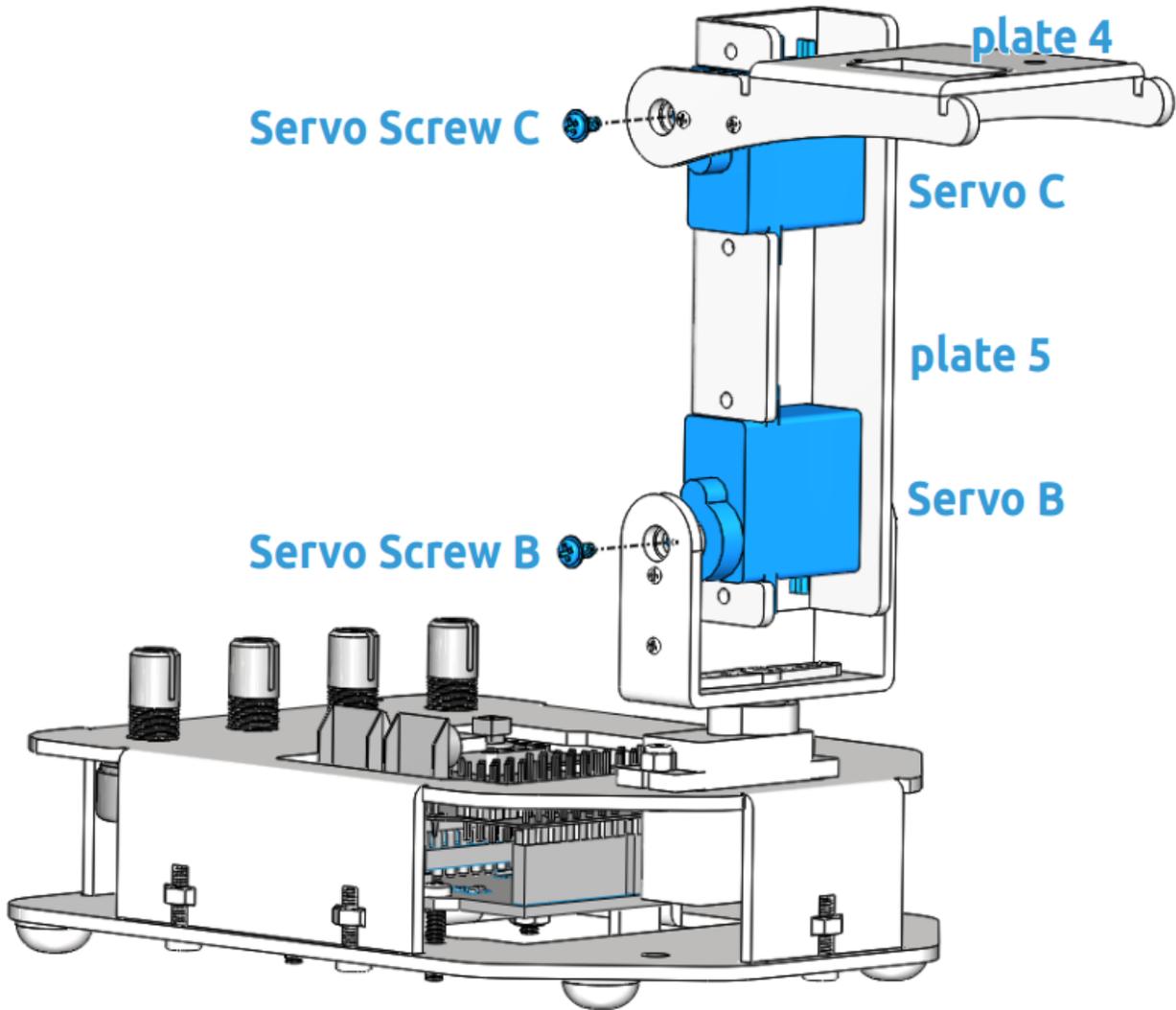
## 2 Complete plate 5 + plate 4



**3** Complete One-Side Servo Arm



**4** Complete Servo B and Servo C

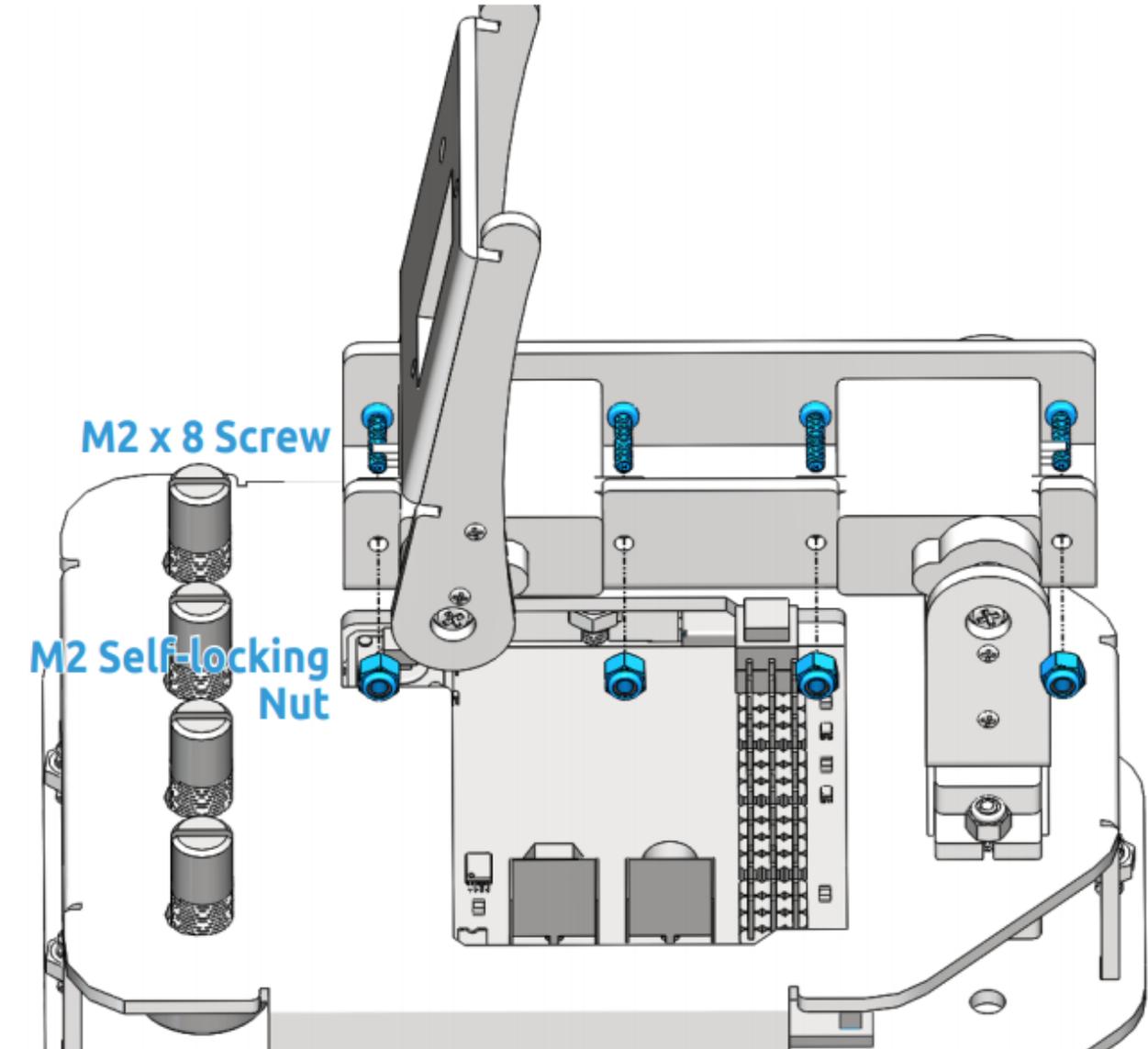


---

**Note:** In this step, please keep the control board power on, Rotate plate 4 to a 90 degree angle with plate 5.

---

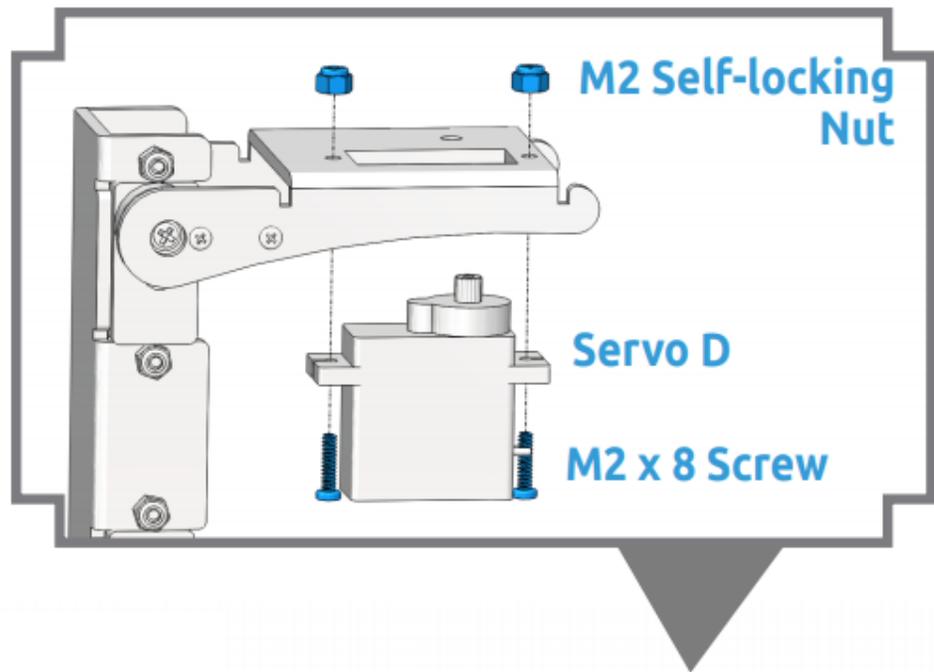
# 5 Fixed Servo B and Servo C



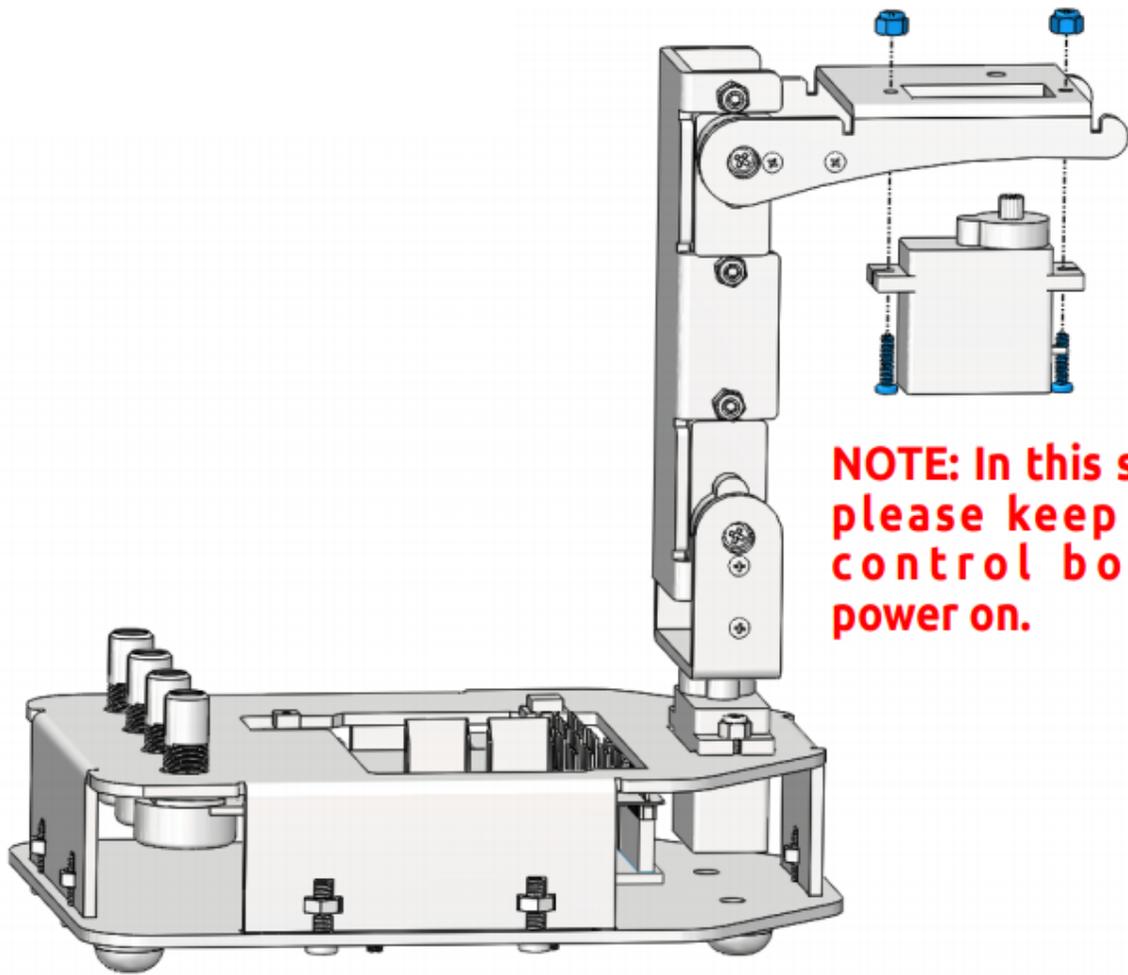
**Note:** In this step, please temporarily turn off the power of the control board, and then rotate the plate 5 to an appropriate angle to fix the servo. Similarly, use a spanner to tighten these M2 self-locking nuts.

6

Complete Servo D

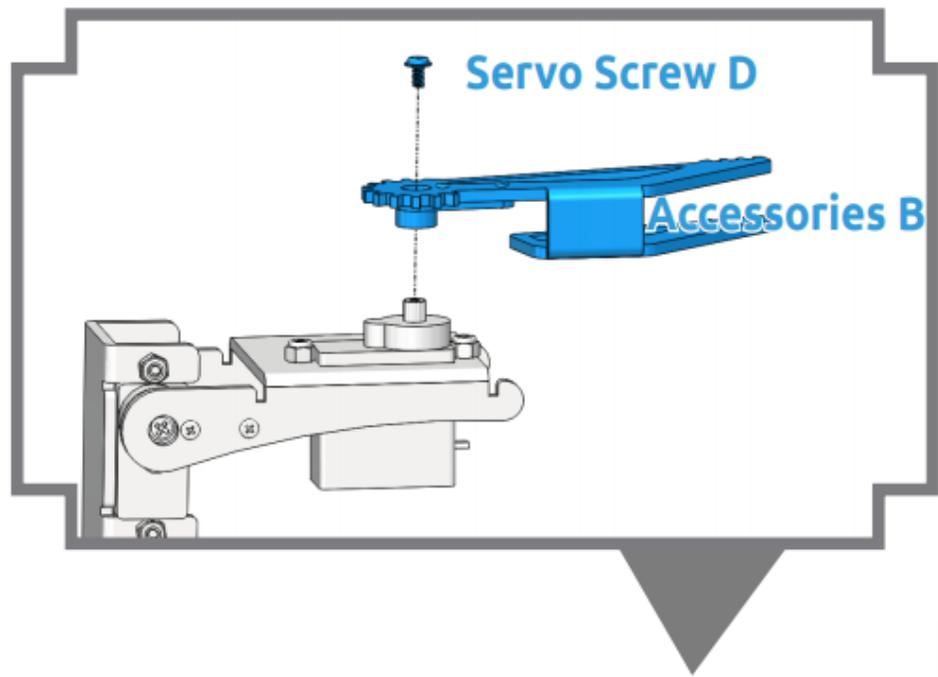


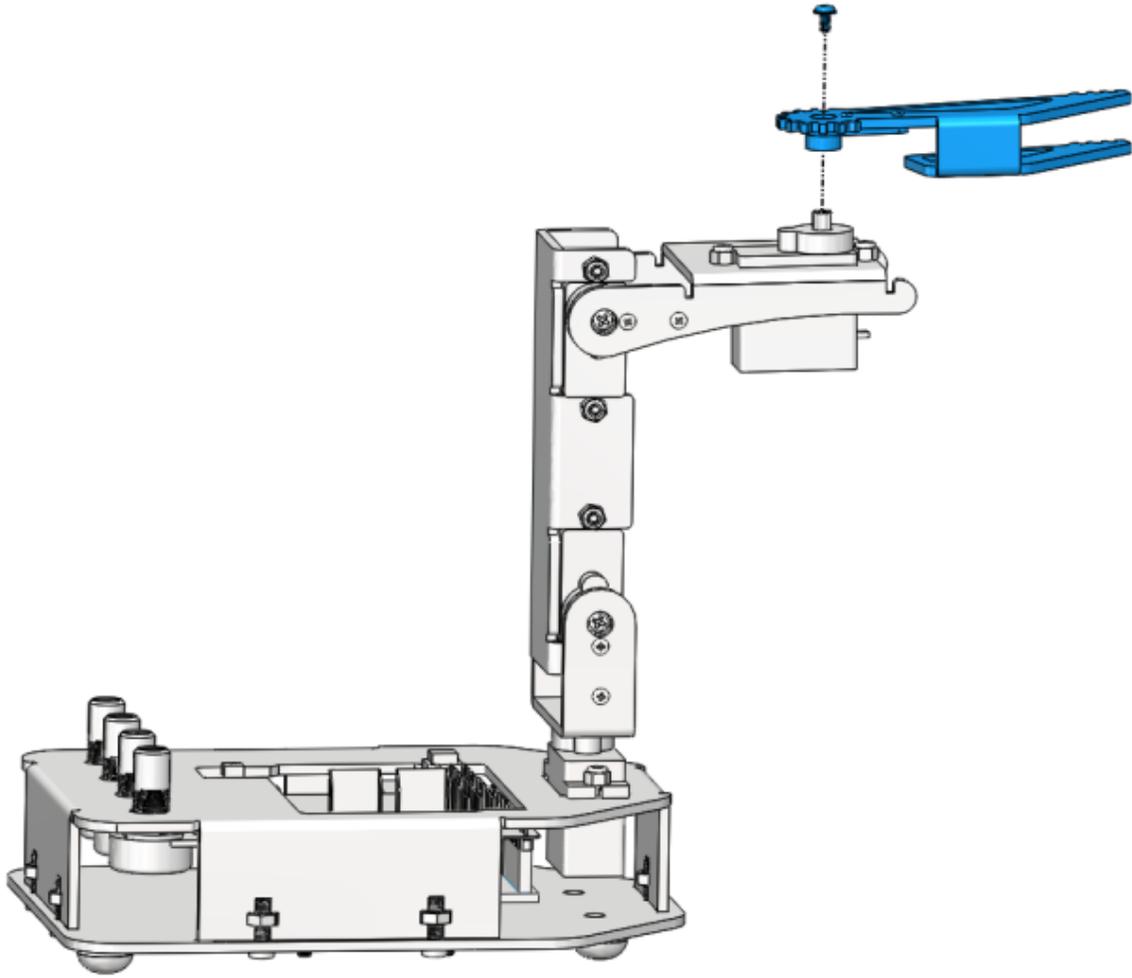
**Note:** Use a spanner to tighten these M2 self-locking nuts.



**NOTE: In this step, please keep the control board power on.**

**7** Complete Accessories B





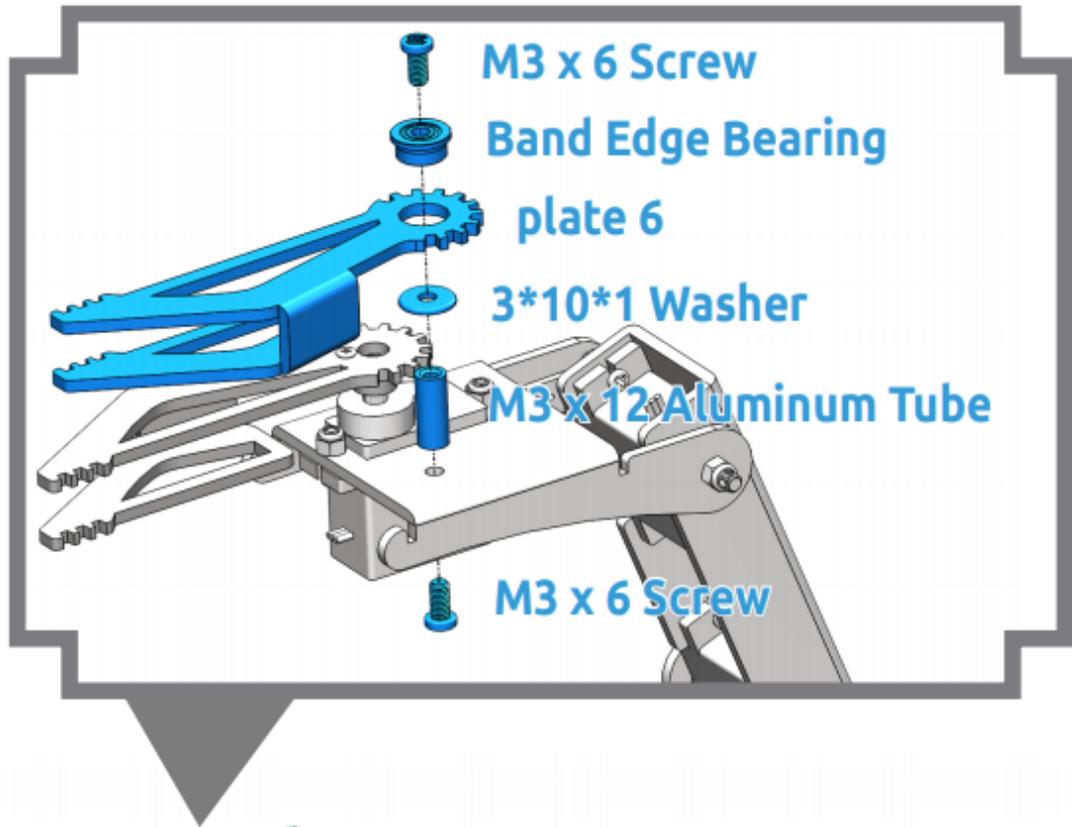
---

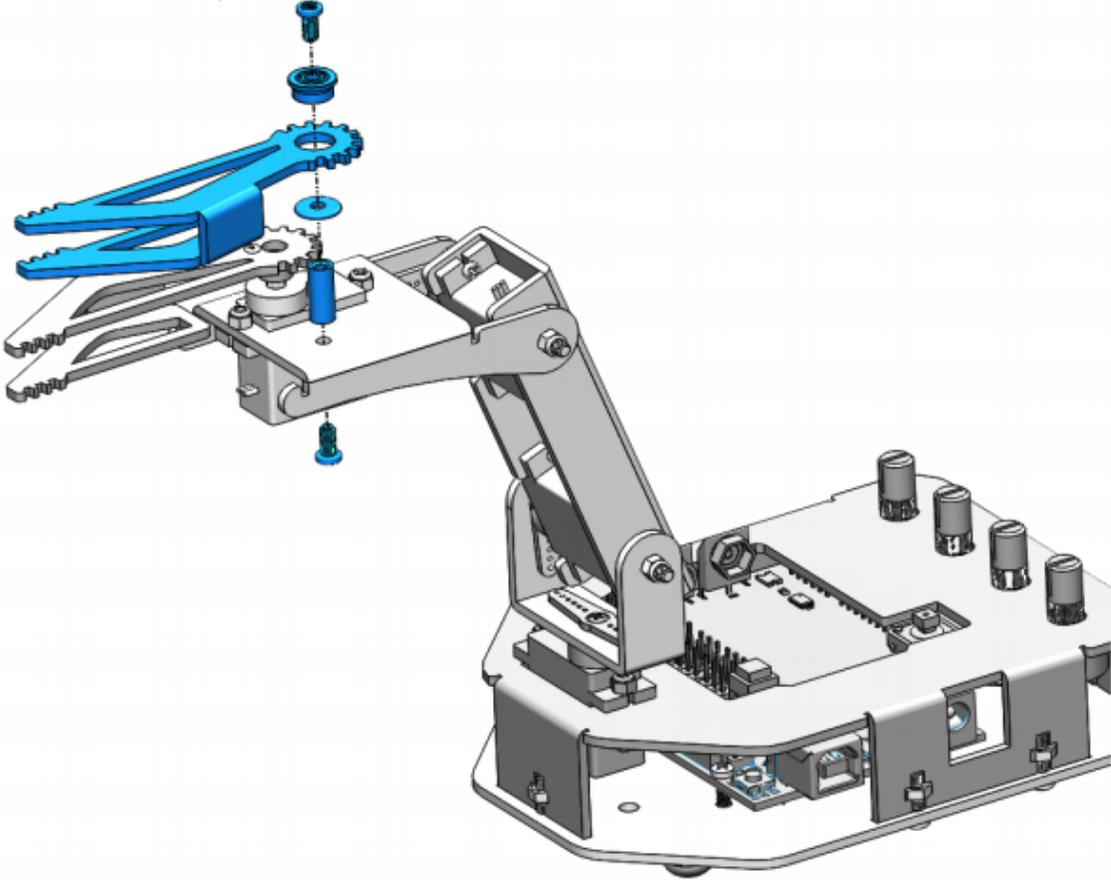
**Note:** Power on the control board again in this step. Keep the gripper edge of the plate 7 parallel to the side of the plate 4.

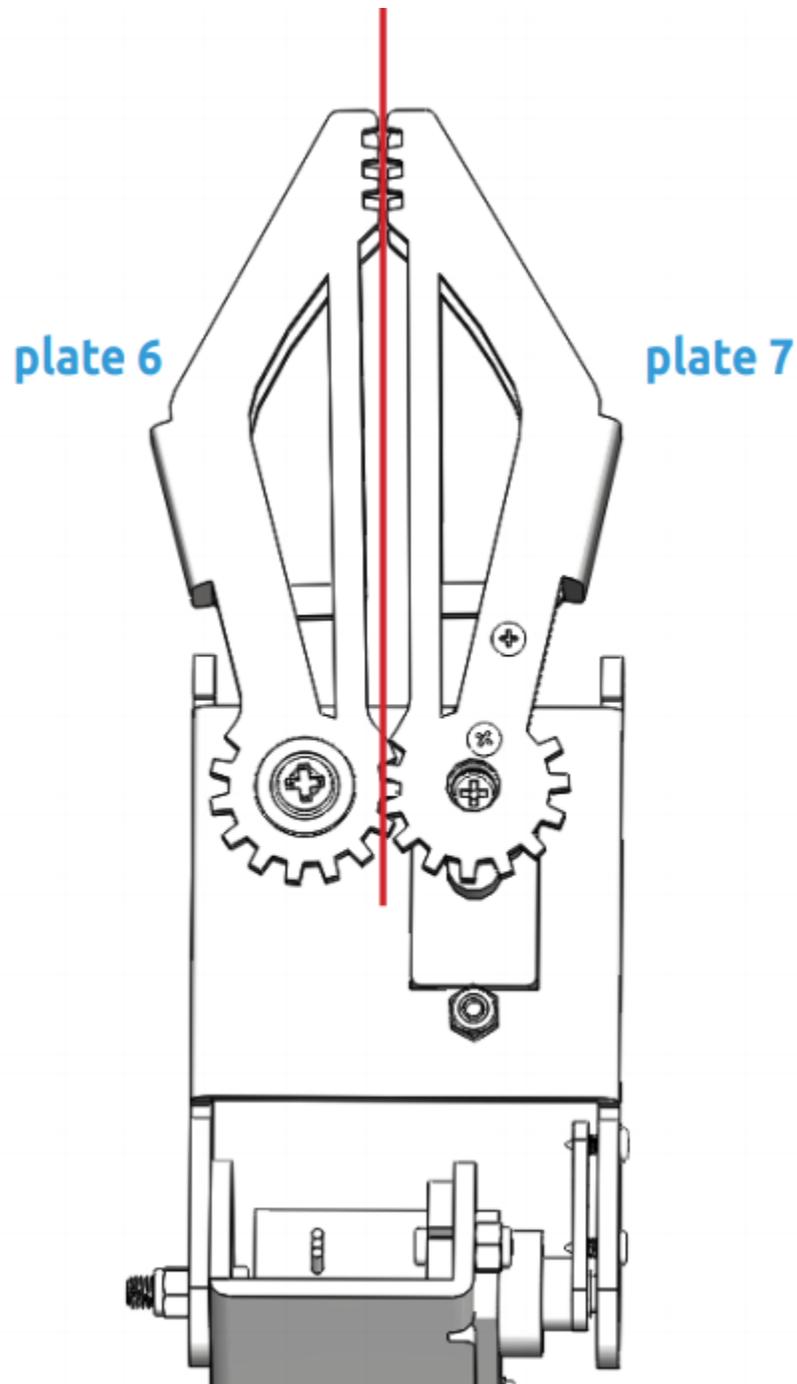
---

8

Complete plate 6







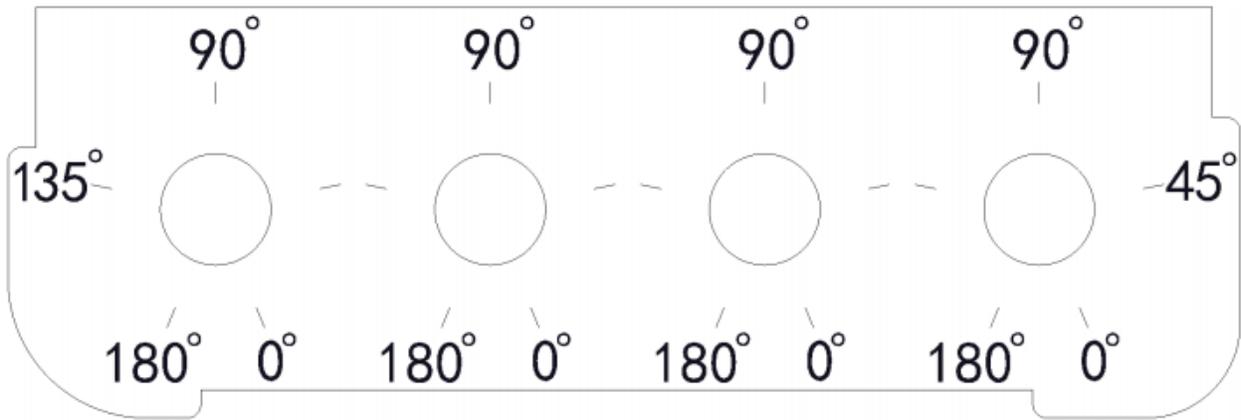
---

**Note:** Make sure that the inner sides of the plate 6 and plate 7 are parallel and the gears meshed.

---

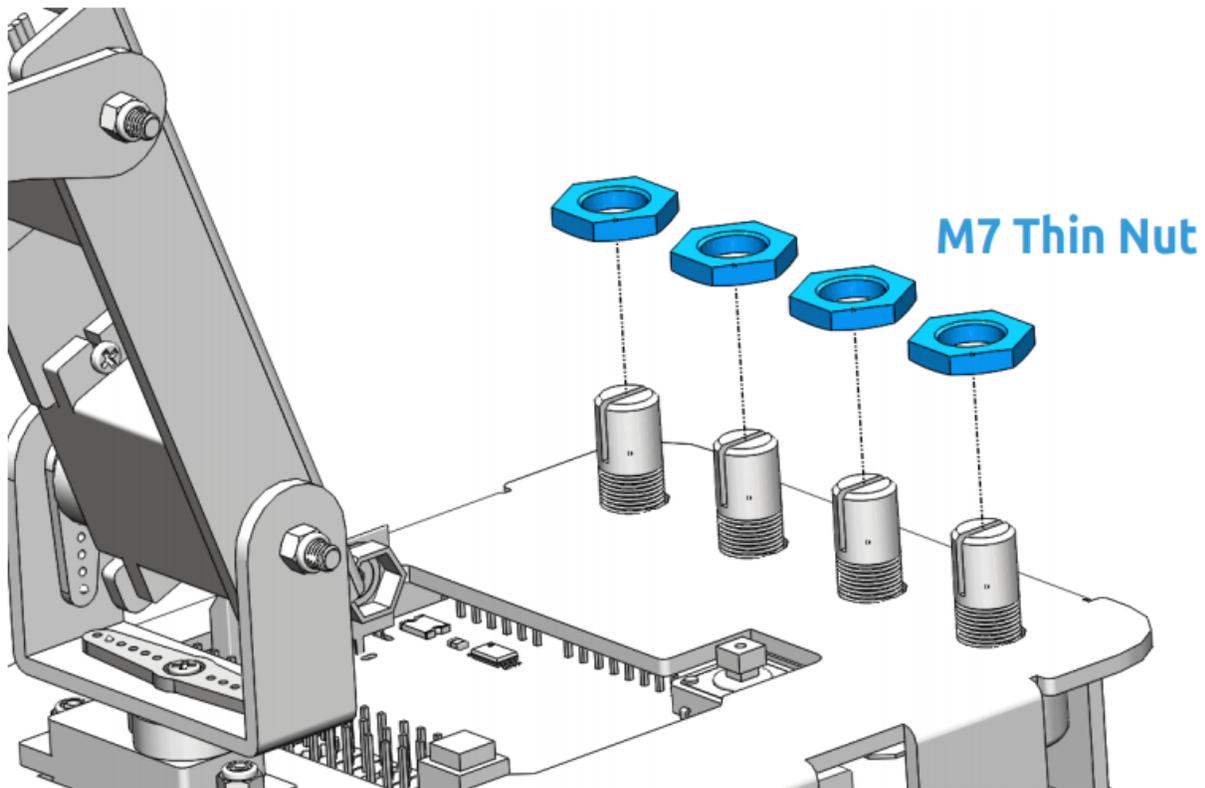
### 3.5.5 Install Button

Please print the **scale.dwg** file in a **1:1 ratio**. After the printed scale paper is cut, place the paper on plate 2.



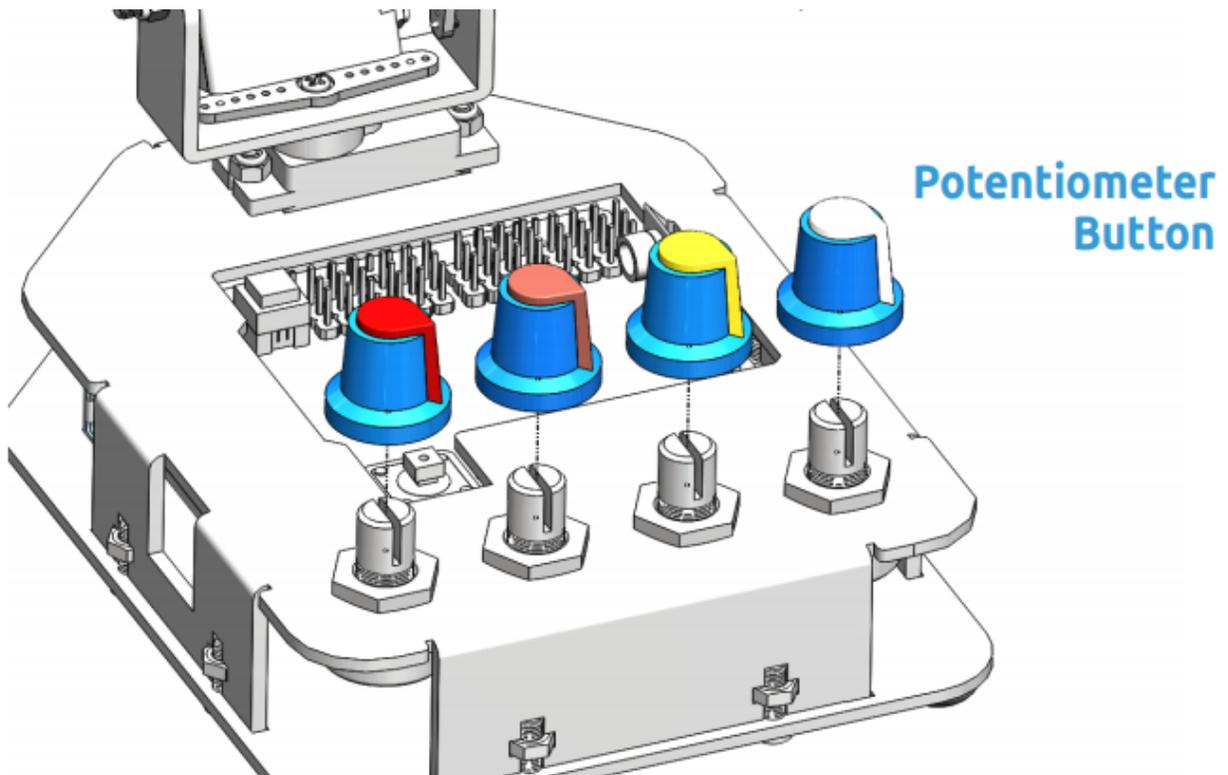
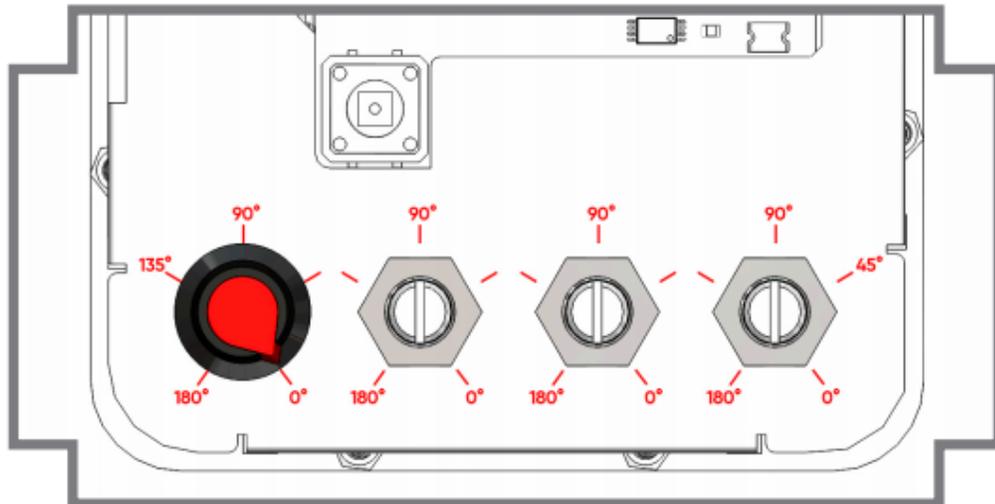
**Note:** It is recommended to use AutoCad software for printing.

## 1 Complete M7 Thin Nut



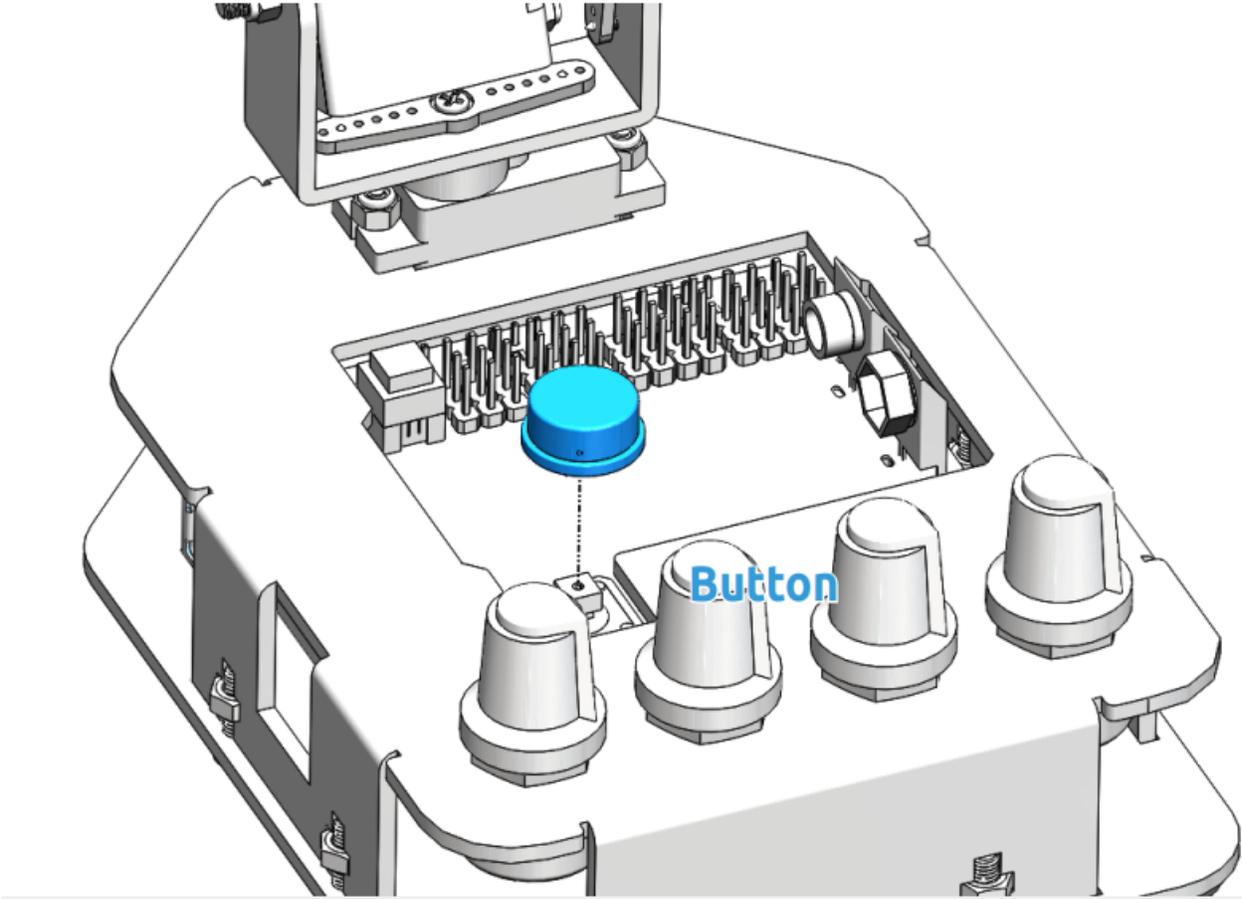
## 2

## Complete Potentiometer Button



**Note:** Align the bulged part of button with the 0° position. Rotate the potentiometer clockwise to the end, and put the button onto the potentiometer and press to tighten them.

### 3 Complete Button

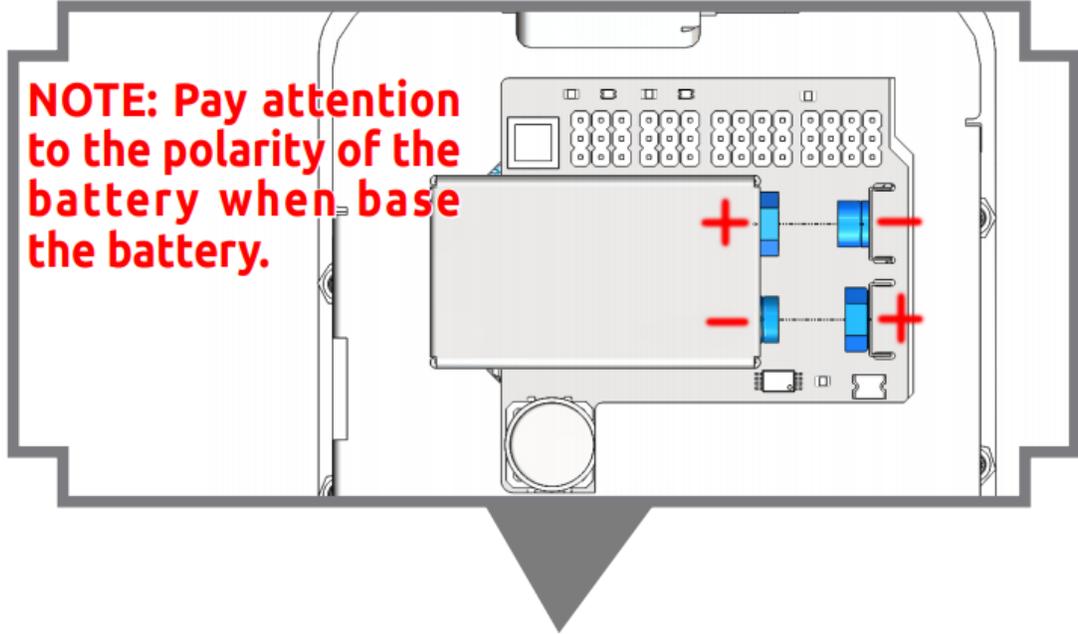


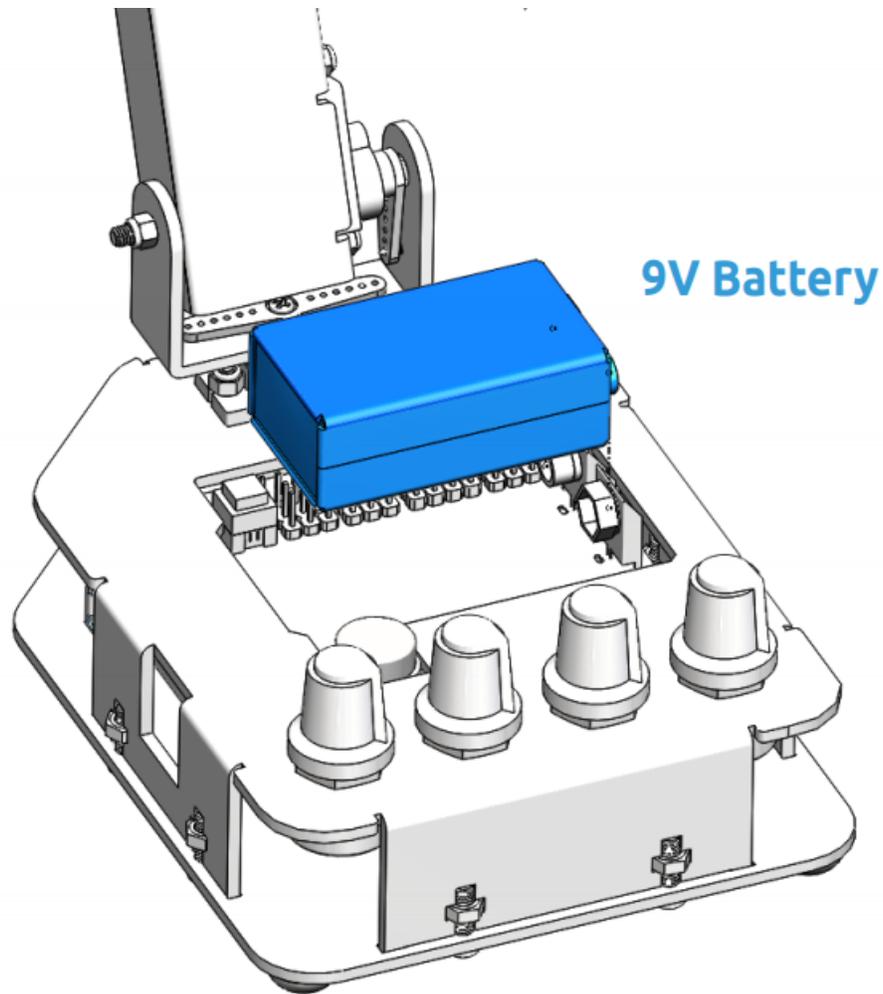
### 3.5.6 Install Battery

1

Complete 9V Battery

**NOTE: Pay attention to the polarity of the battery when base the battery.**



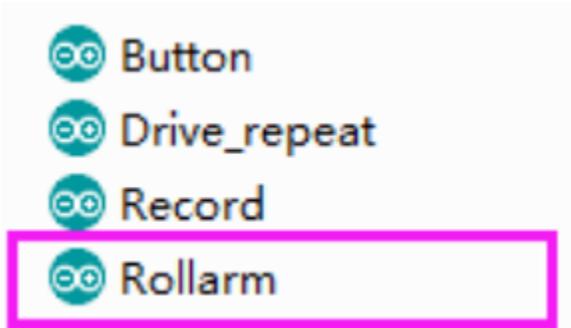


### 3.6 Control the Rollarm

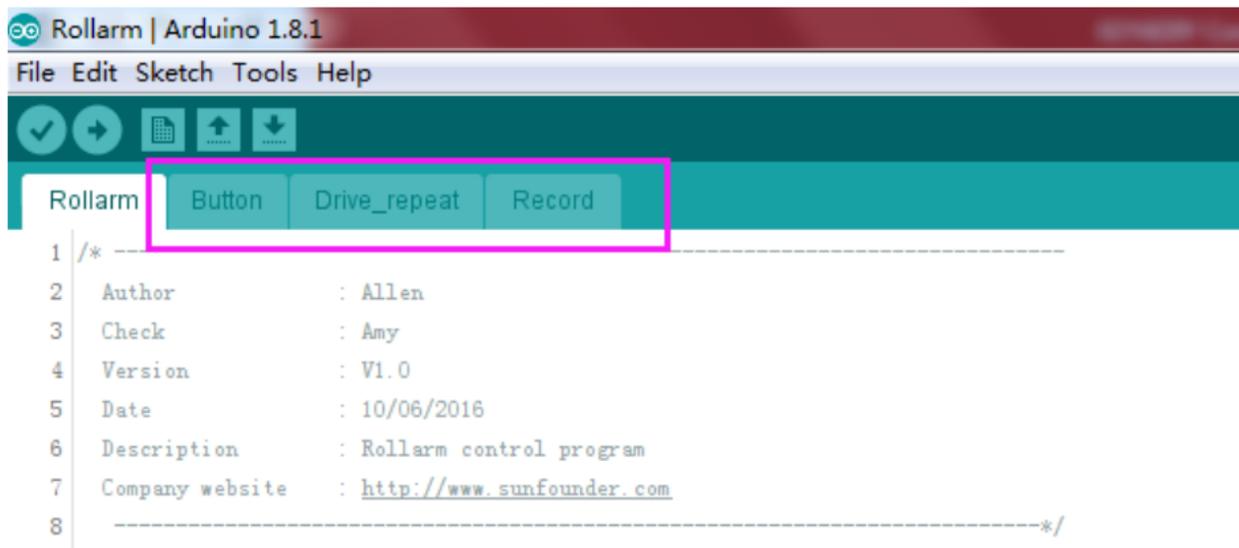
There are two ways to control the Rollarm: manual control (by handle), or PC control (by Labview). The detailed operations for two ways are as follows.

### 3.6.1 Manual Control

**Step 1:** Run the **Rollarm.ino** code under the path **DIY Control Robot Arm kit for Arduino-Rollarm\Arduino Code\Rollarm**. There are four code files in Rollarm, **Rollarm.ino** is the main program, when the others are subprograms.



When you open the main program, the subprograms will be opened automatically

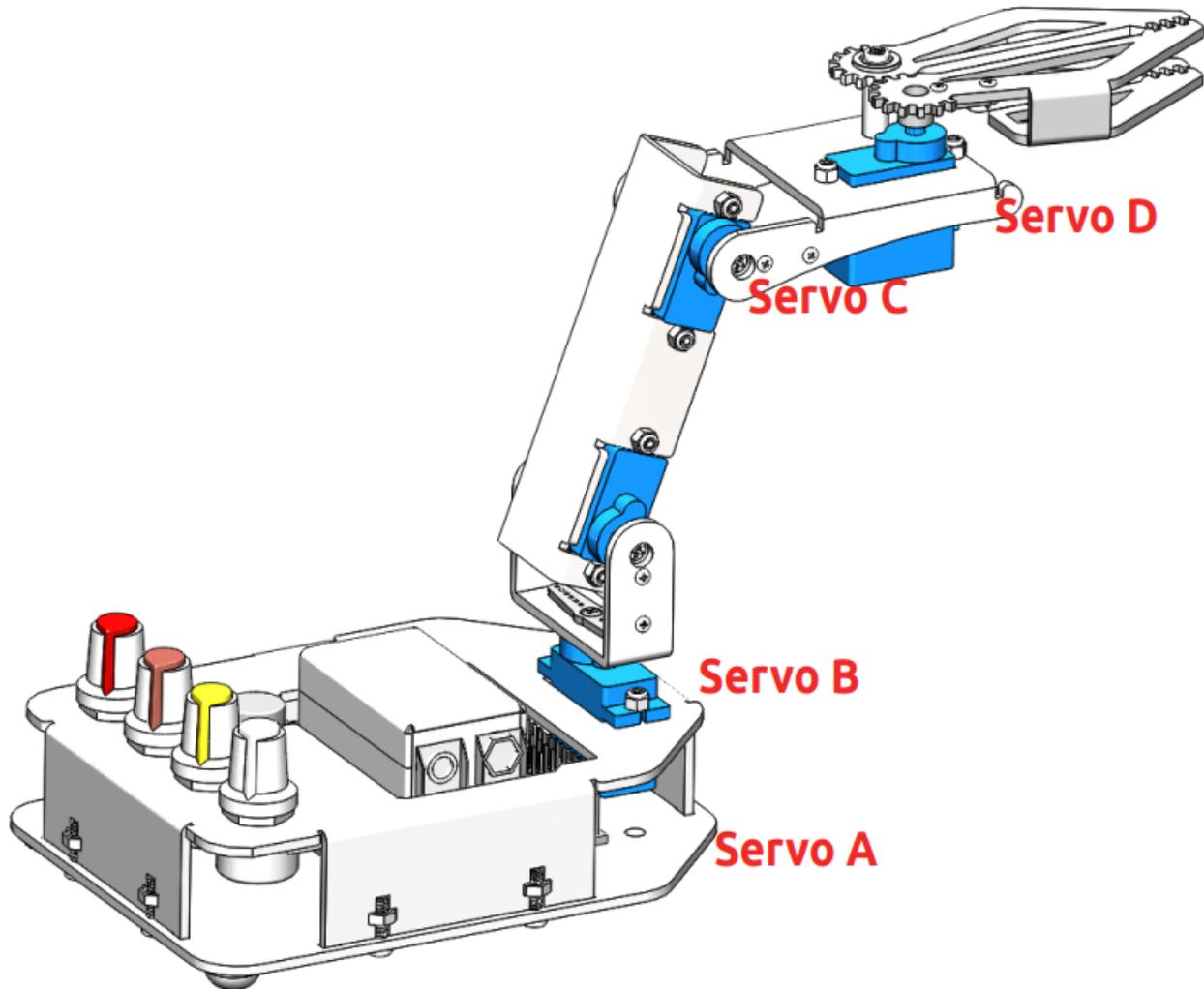


**Step 2:** Select the corresponding board and port, then click **Upload**.

**Step 3:** After the code upload, turn the power switch on, then we can try to control the Rollarm.

**Step 4:** Rotate the four potentiometer buttons to try the controlled servo and direction:

White potentiometer to control the Servo D, the yellow to control Servo C, the orange to control Servo B, and the red one to control Servo A.

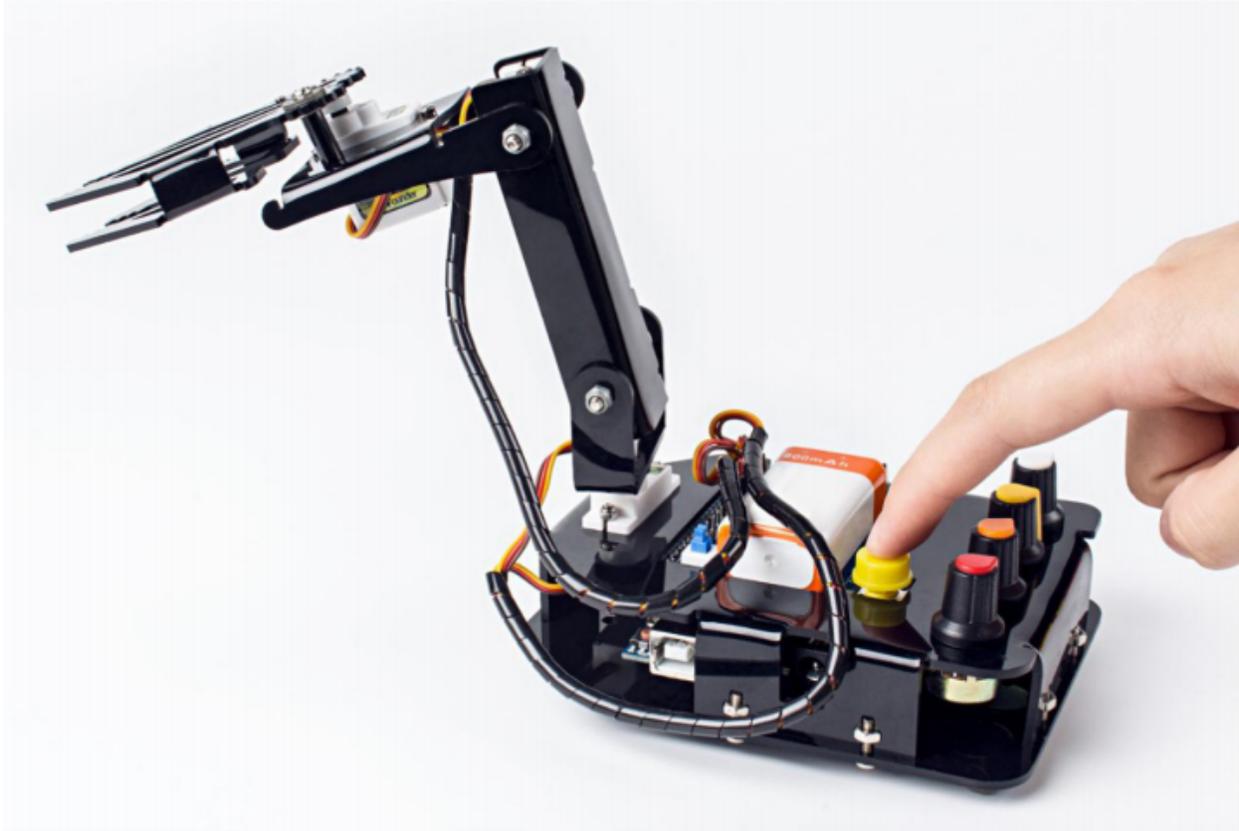


### Record behavior

With the handle, the Rollarm can record its behaviors:

**Rotate one potentiometer** to control one servo to the desired position, and **press the yellow button shortly** to let the control board record this step. Record the rest steps in this way.

When all the steps are done, **press the yellow button for a while (3s)**, it will repeat the recorded steps (Rollarm can record at most 100 steps because of the control board's memory limit.)



Thus we can make it automatically carry blocks continuously:

### Code Explanation

The program includes three parts: rotating the potentiometers to control the Rollarm, pressing the button slightly for less than one second to record Rollarm's behaviors and pressing the button for a relatively longer time to make Rollarm repeat the recorded steps.

There are four potentiometers to control the arms. The 4 servos from top to bottom are connected to port 4-7 respectively of the expansion board, and the 4 potentiometers control the ports accordingly. In other words, spin the white potentiometer to control the uppermost servo, the yellow to control the next servo below, the orange to control the next servo, and the red one to control the bottom servo.

Since the Rollarm has four servos acting as the moving joint, we need to include a header file for driving the servos and define them.

```
//Create servo object to control a servo.
#include <Servo.h>

Servo Servo_0;
Servo Servo_1;
Servo Servo_2;
Servo Servo_3;
```

After defining the function of driving the servos, we need to read the AD value of the potentiometers and convert it into the rotating angle of the servo since the servos are controlled by rotating the potentiometers.

```

//Read the values of the potentiometers.
void ReadPot ()
{
    SensVal[0] = 0;
    SensVal[1] = 0;
    SensVal[2] = 0;
    SensVal[3] = 0;

    SensVal[0] = analogRead(A0);
    SensVal[1] = analogRead(A1);
    SensVal[2] = analogRead(A2);
    SensVal[3] = analogRead(A3);
}

//The value of the potentiometer is matched to the angle value.
void Mapping0 ()
{
    SensVal[0] = map(SensVal[0], 0, 1023, 10, 170);
    SensVal[1] = map(SensVal[1], 0, 1023, 10, 170);
    SensVal[2] = map(SensVal[2], 0, 1023, 10, 170);
    SensVal[3] = map(SensVal[3], 0, 1023, 100, 180);
}

```

After compiling the program, we need to make Rollarm remember the steps, which is done through pressing the button.

```

//Calculate the time the button pressed
void Button()
{
    if (digitalRead(3) == 0)
    {
        delay(10);
        if (digitalRead(3) == 0)
        {
            KeyValue = 0;
            while (!digitalRead(3))
            {
                KeyValue++;
                delay(100);
            }
        }
    }
}

```

We can tell which part of the code the Rollarm is performing by reading the value upon pressing the button. When the value is larger than 10, it means Rollarm is repeating the steps. When it is between 0 and 10, it means Rollarm is remembering. And when it is 0, it means Rollarm is being controlled by the potentiometers. The specific program is as follows:

```

//Check the button.
static int Flag = 1;
Button();

//The time of pressing the button is not long then record the action.
if ((KeyValue < 10) && (KeyValue > 0))
{
    KeyValue = 0;
    Record();
}

```

(continues on next page)

(continued from previous page)

```

    Mapping1();
}
//Long press the button and open the auto mode,start repeating the action.
else if (KeyValue > 10)
{
    if (Flag == 1)
    {
        Flag = 0;
        Calculate();
    }
    Drive_init();
    delay(3000);
    for (int i = 1; i < Time; i++)
    {
        Drive_repeat(i);
        delay(500);
    }
}
//Did not press the button, open the manual mode.
else
{
    ReadPot();
    Mapping0();
}

```

Next, we are going to call the function to write the value of the servo rotating angle. However, it is not merely about writing the values directly; the difference between two adjacent rotating values will also be written into the servos. Here we take a servo program for example.

```

//The first axis.
if (Dif0[n] > 0)
{
    for (int j = Joint0[n - 1]; j <= Joint0[n]; j++)
    {
        Servo_0.write(j);
        delay(10);
    }
}
else
{
    for (int j = Joint0[n - 1]; j >= Joint0[n]; j--)
    {
        Servo_0.write(j);
        delay(10);
    }
}

```

### 3.6.2 PC Control

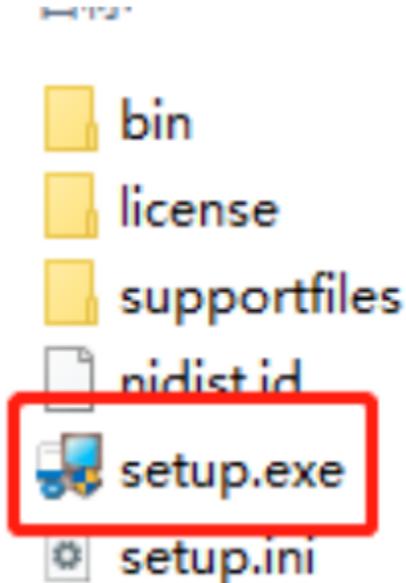
#### Install Labview Software

For this kit, we use the Labview software for control on PC. If you have other better options, welcome to share by post under **Community** -> **FORUM** on our website [www.sunfounder.com](http://www.sunfounder.com).

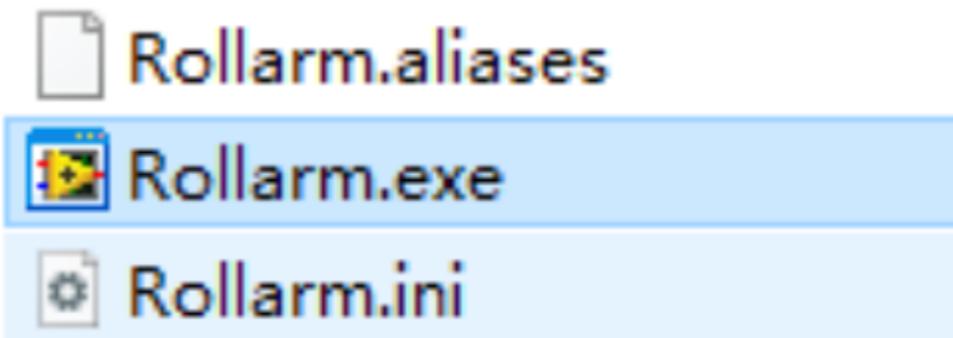
Download the Labview package in the link below:

<https://s3.amazonaws.com/sunfounder/Arduino/Labview.zip>

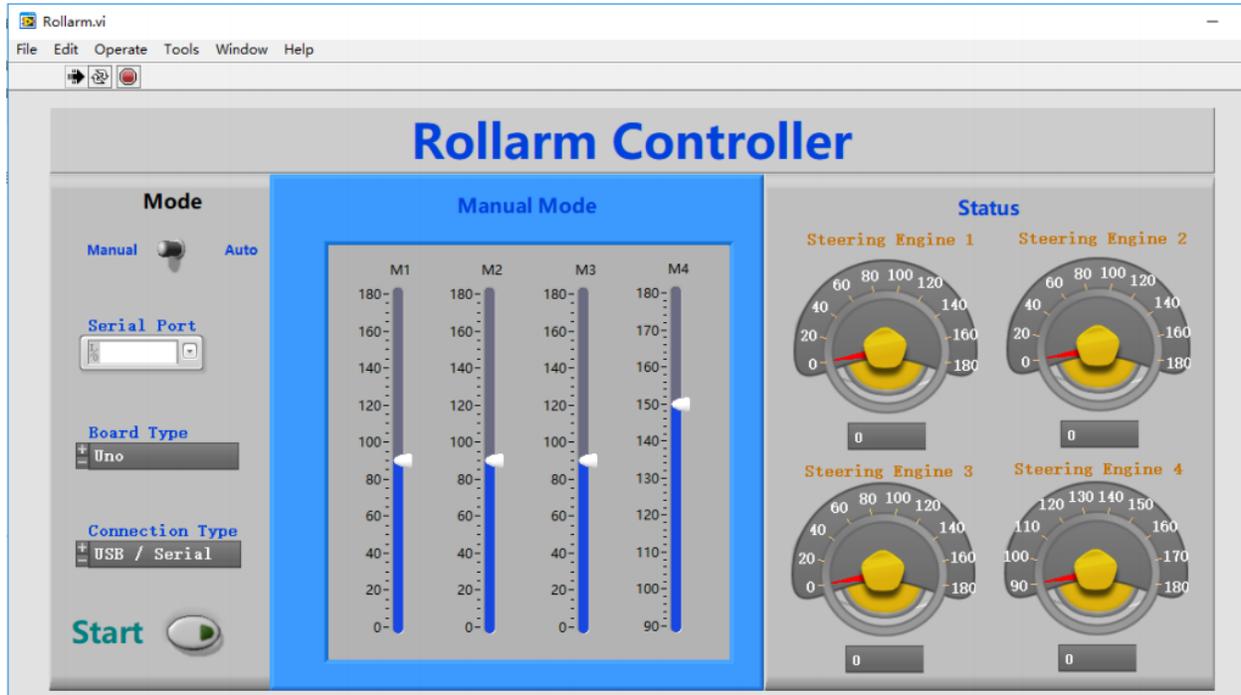
After downloading, open it and you will see the folder Rollarm's Installer. Then find the setup file as shown below:



After the installation is done, in Start Menu, find Rollarm to open the Rollarm software. Or, enter the installation directory we used just now, and double click Rollarm.exe to open it. The defaulted installation directory is: C:\Program Files (x86)\Rollarm.



The following interface will show up.



## Upload the Code

Before using the Rollarm Labview software, flash the control codes into the Rollarm robot, and the steps are as follows.

**Step 1:** In order to avoid the incompatibility, please download Arduino IDE 1.0.5 on Arduino official website:

<https://www.arduino.cc/en/Main/OldSoftwareReleases#previous>

You are suggested to download zip version because zip version does not need installation and you can use it directly.



## Arduino 1.0.x

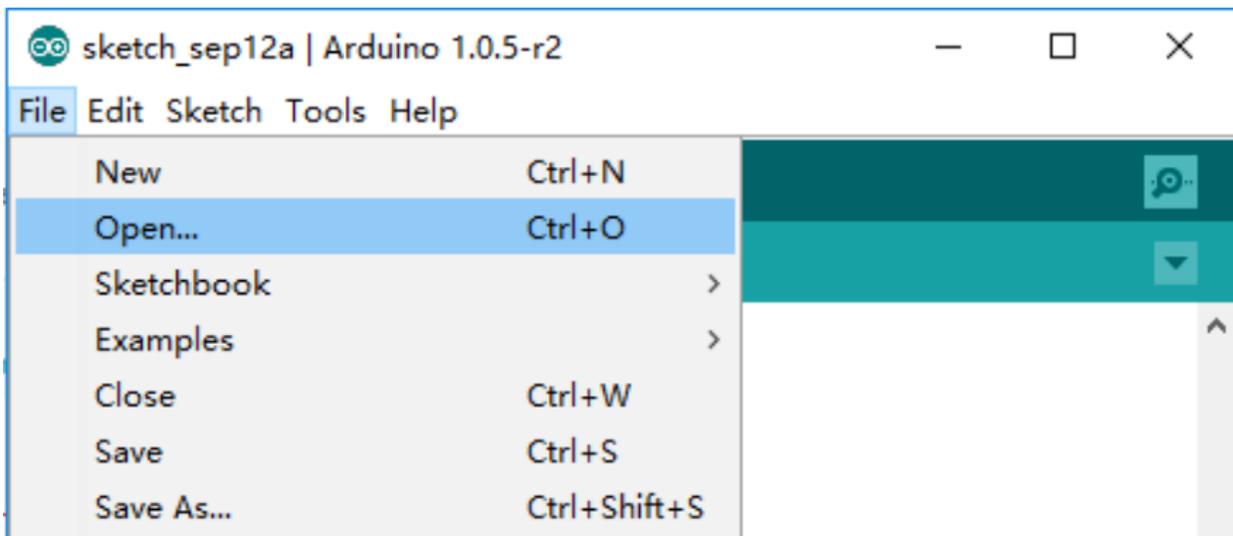
These packages are no longer supported by the development team.

1.0.5	Windows Windows Installer	MAC OS X	Linux 32 Bit Linux 64 Bit	Source code hosted on Gcode
1.0.4	Windows	MAC OS X	Linux 32 Bit Linux 64 Bit	Source code hosted on Gcode

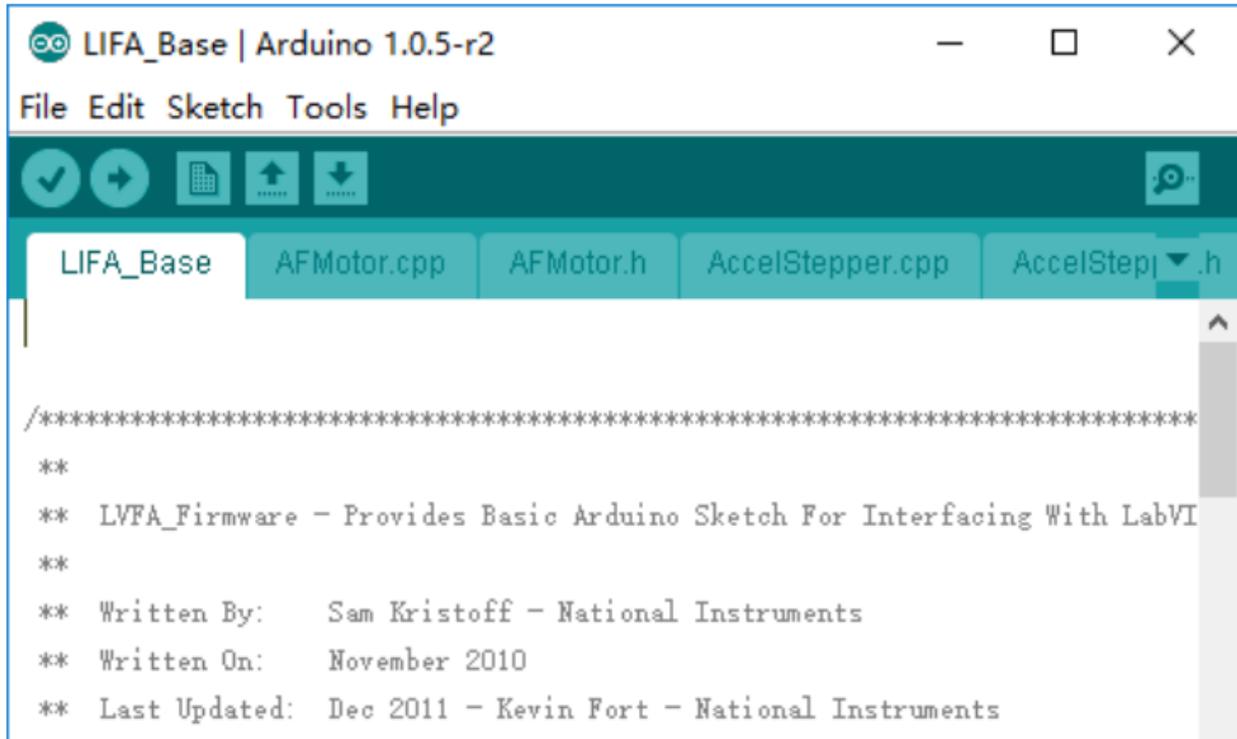
**Step 2:** After downloading and unzipping them, you need to double click the arduino.exe to open it.

lib	2019/9/12 14:34
libraries	2019/9/12 14:34
reference	2019/9/12 14:34
tools	2019/9/12 14:34
<b>arduino.exe</b>	2014/1/9 3:46
cygiconv-2.dll	2014/1/9 3:45
cygwin1.dll	2014/1/9 3:45
libusb0.dll	2014/1/9 3:45
revisions.txt	2014/1/9 3:45
rxtxSerial.dll	2014/1/9 3:45
uninstall.exe	2019/8/15 15:13

**Step 3:** Click File- >Open, and you can find the LIFA\_Base.ino on the path  
DIY\_Control\_Robot\_Arm\_kit\_for\_Arduino-Rollarm.1\Arduino Code\LIFA\_Base  
And click to open.



**Step 4:** Choose the proper Board and Port, and upload the codes to the control board.



```
/**  
** LVFA_Firmware - Provides Basic Arduino Sketch For Interfacing With LabVIEW  
**  
** Written By: Sam Kristoff - National Instruments  
** Written On: November 2010  
** Last Updated: Dec 2011 - Kevin Fort - National Instruments  
**
```

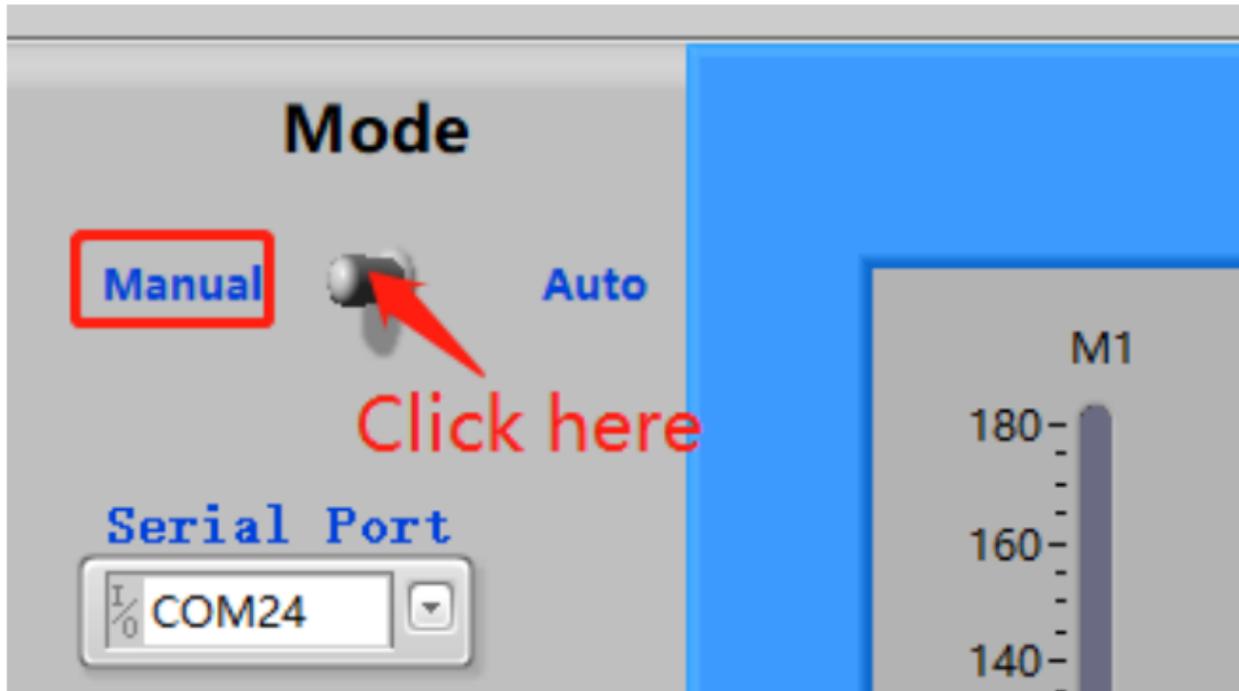
DO NOT unplug the USB cable at the moment.

### Using the Software

Back to the Rollarm Labview software, which includes two parts: **Manual** Mode and **Automatic** Mode.

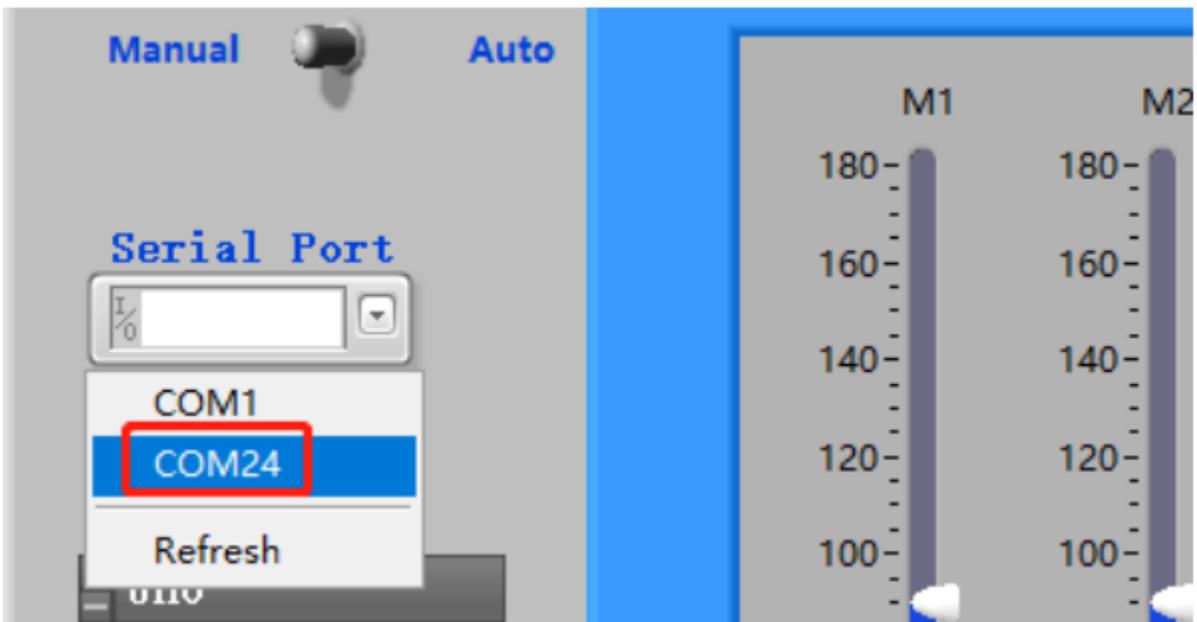
#### Manual Mode

**Step1:** See the interface of manual control below. After the Labview is installed and run, this mode is enabled by default.

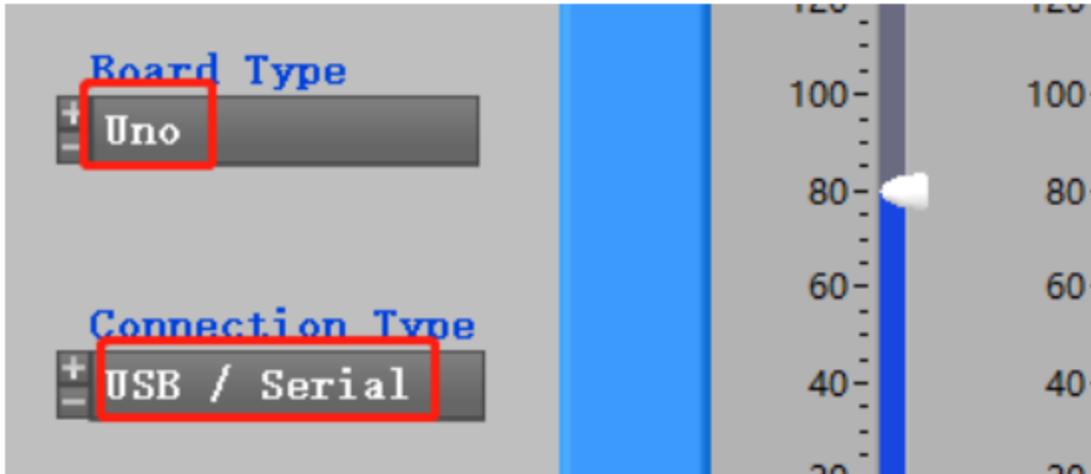


**Step2:** Click the inverted triangle icon for Serial Port, select the port according to you COM port. Here is COM24, which varies for different computers.

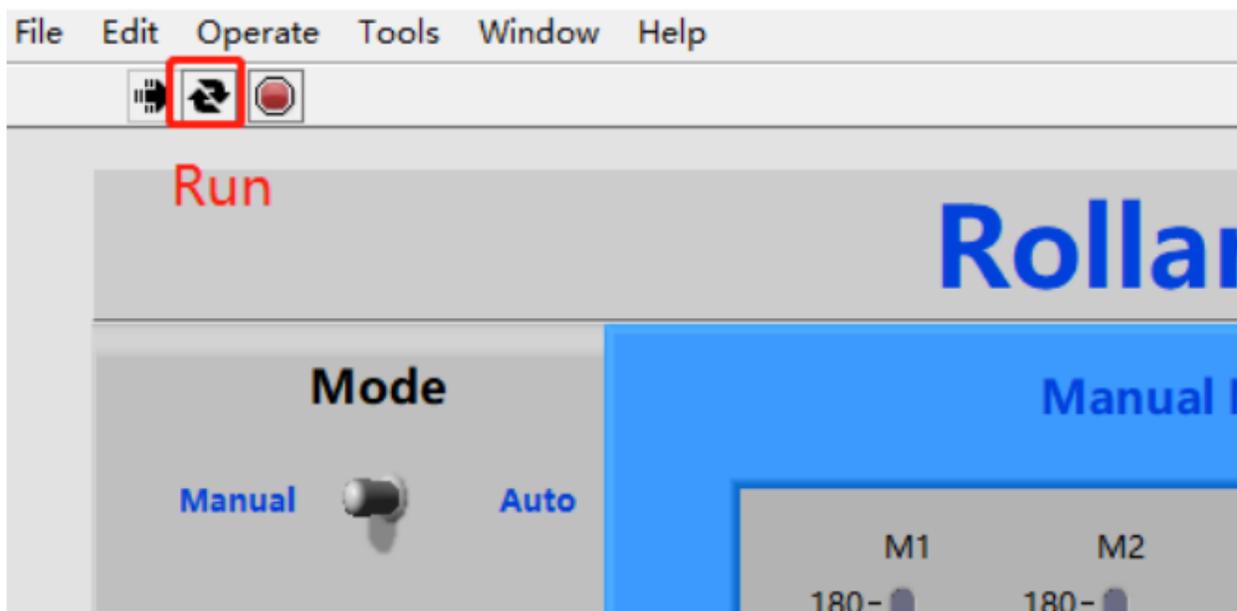
**Note:** If you can only see COM1, to solve the problem, just replug the USB cable. Then start from sketch upload again.



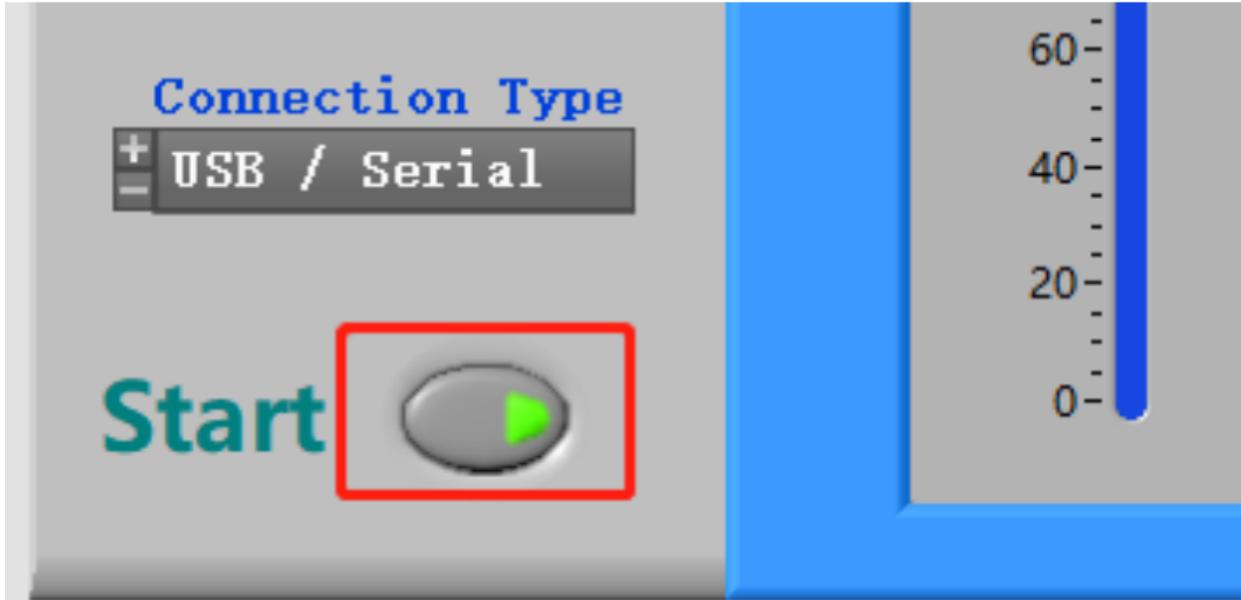
**Step 3:** Select the **Board Type (Uno)** and **Connection Type(USB/Serial)**.



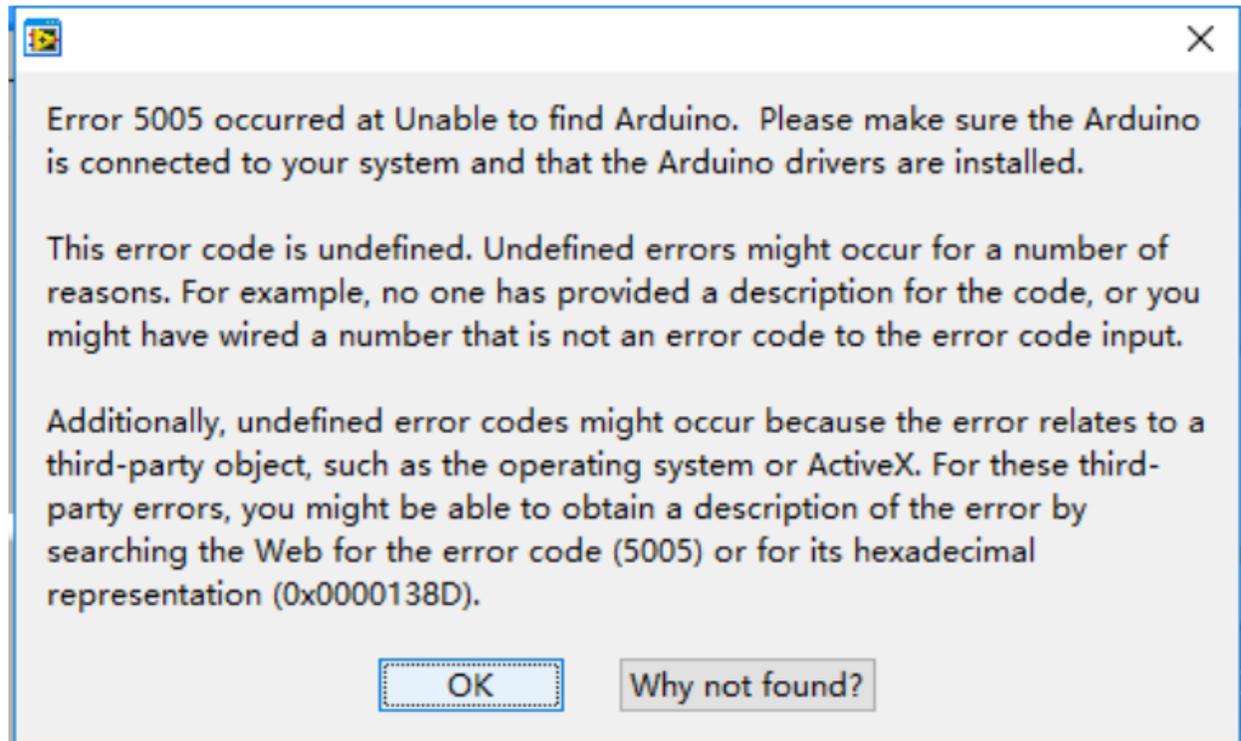
**Step 4:** There are **three** small icons at the top left. Click the middle one to run the software.



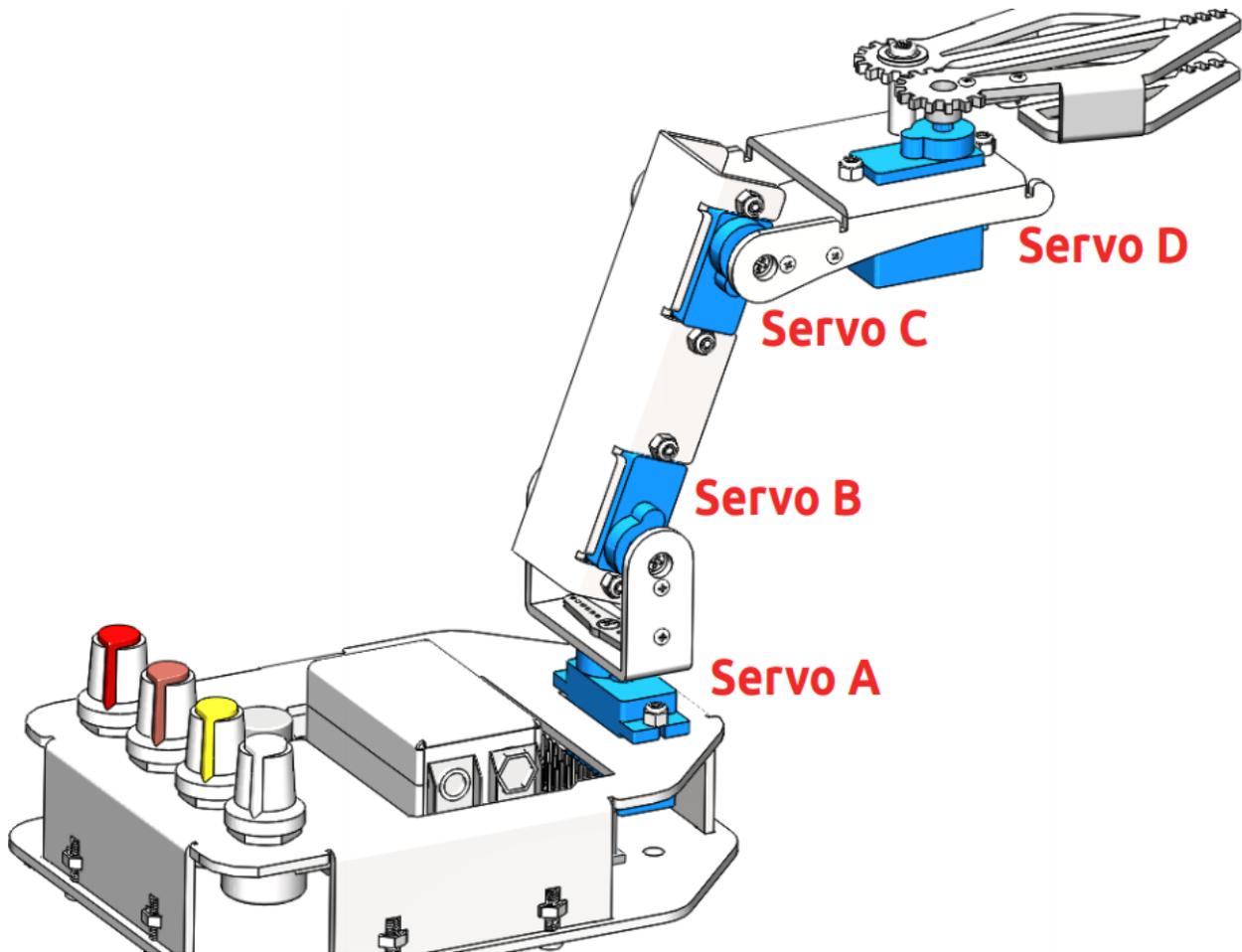
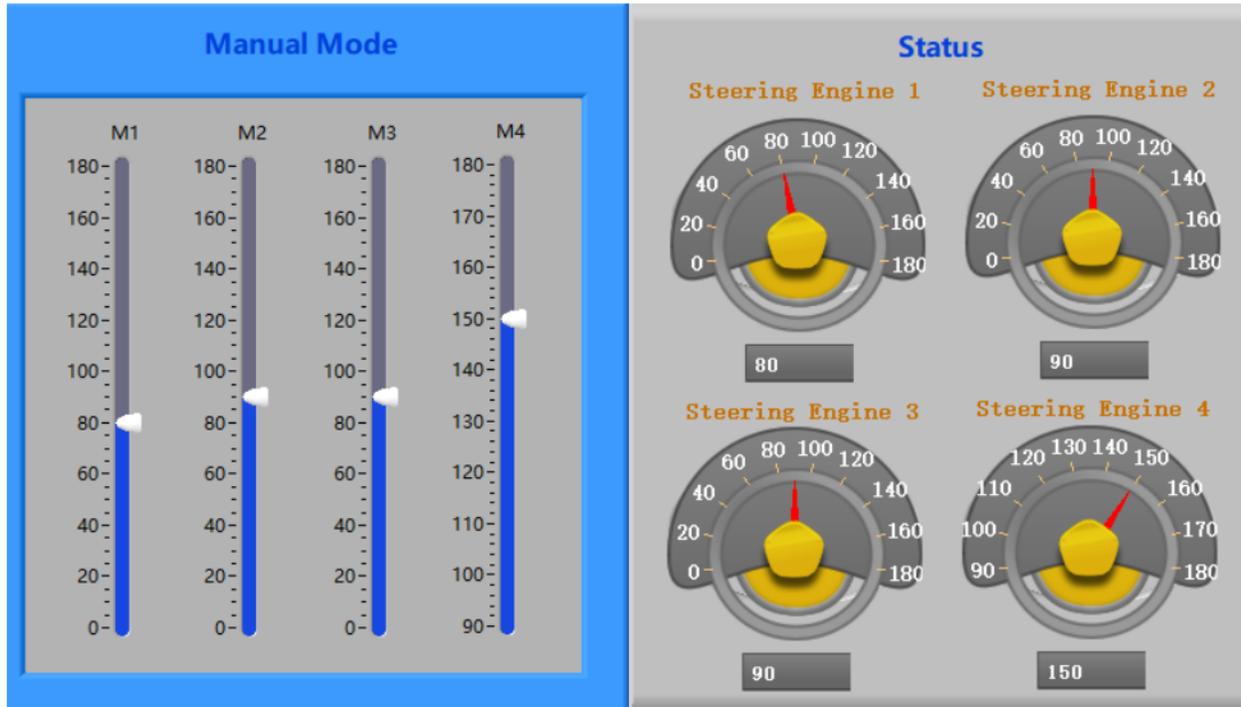
**Step5:** Click **Start**, and the button will change from dark to light **green**.



**Note:** Sometimes you can see the mistakes like 5005 or 5002 that are as results of the software recognition failure of Arduino control board. Now, please click OK. Then there is a quick quiver state in Rollarm robot to return to the setting position. After that, you can continue to do your next step.



**Step 6:** M1-M4 correspond to Servo1-Servo4, you can move the slider control the 4 servo on the Rollarm.



## Automatic Mode

You can also switch to **Auto Mode**. Fill the value of the rotating angle of the servos into the table under **Auto Mode** one by one. After filling the figures, click the **Start** button, then Rollarm will then perform as you just set.

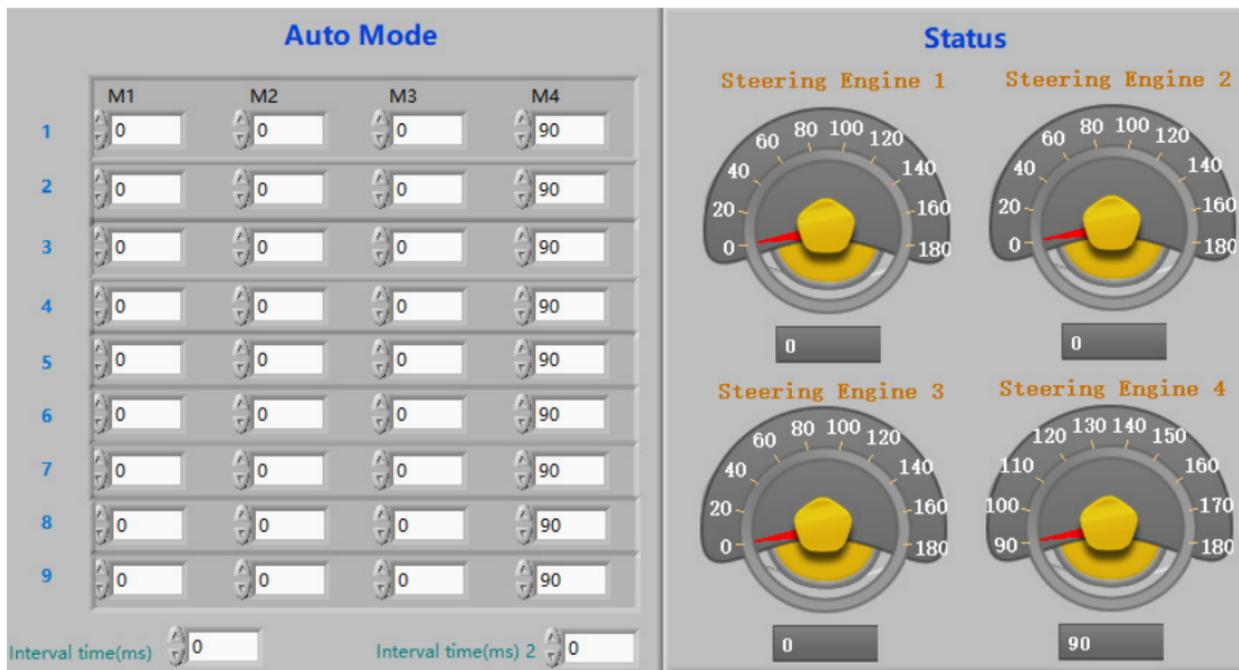
**M1-M4:** Servo1-Servo4.

**1-9:** 9 groups of rotating angle.

**Interval times(ms):** The interval times between two groups, such as interval times between M4 in row 1 and M1 in row 2.

**Interval times(ms) 2:** The interval time between two rotating angles within a group.

**Note:** the range of the data for Mode 4 is 90~180. Otherwise, it will be damaged due to stalling.



## 3.7 FAQ

### 3.7.1 About the Assembly

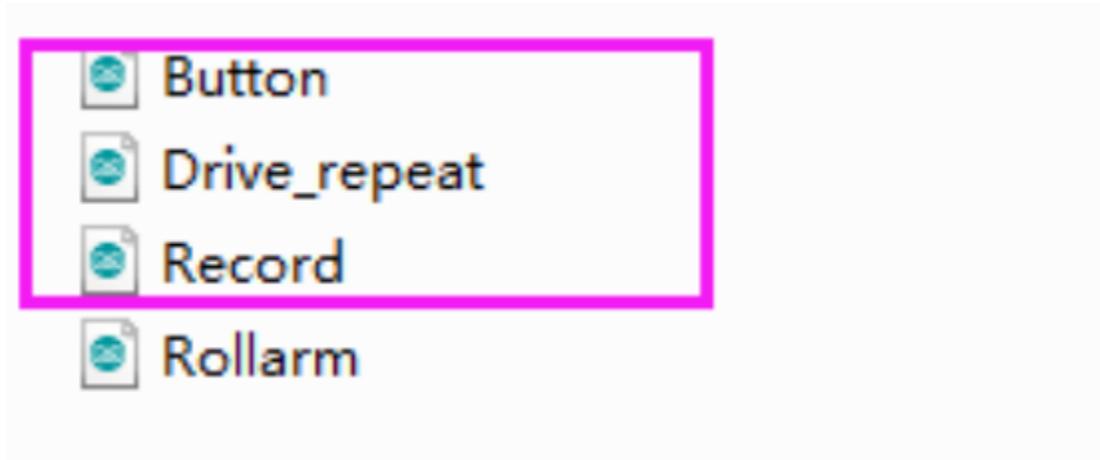
**Q1:** After assembly and program download, the Rollarm's four axes are in wired position, some maybe out of control. What should I do?

**A:** Remember to power on and calibrate each servo before assembly.

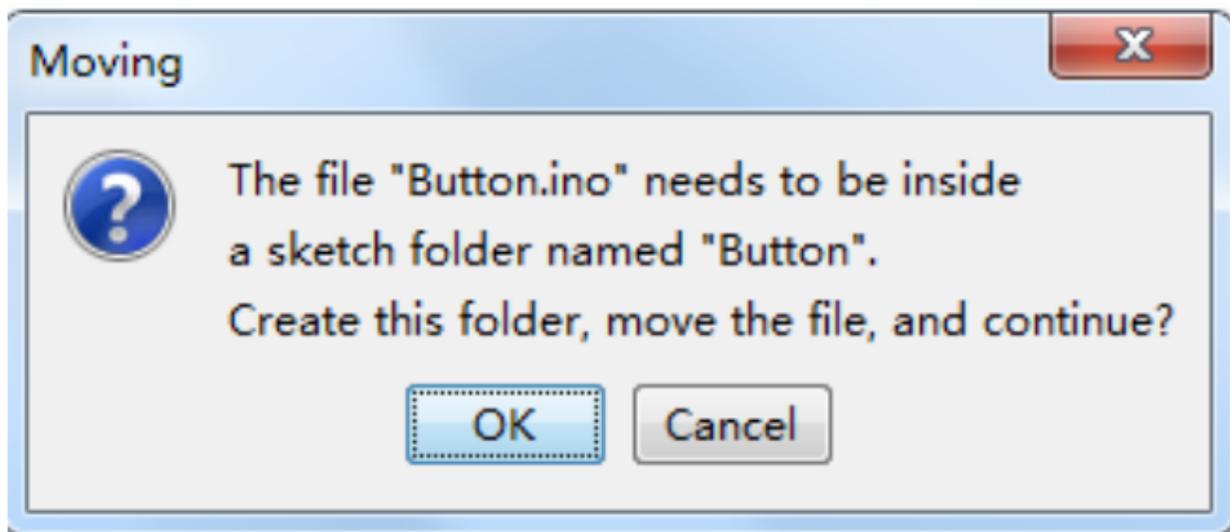
### 3.7.2 About the Arduino code control

**Q1:** When I open a program, it prompts me that a new folder should be created. After I click Yes and a new folder is created, the main program reports an error when I want to open the main program. What's going wrong?

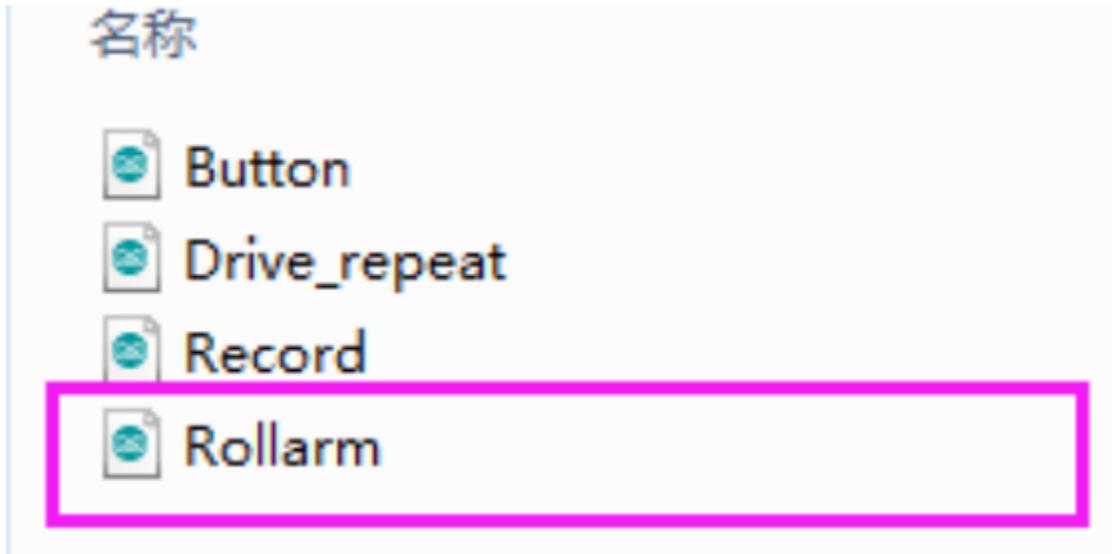
**A:** DO NOT open these subprograms under Arduino code\Rollarm separately:



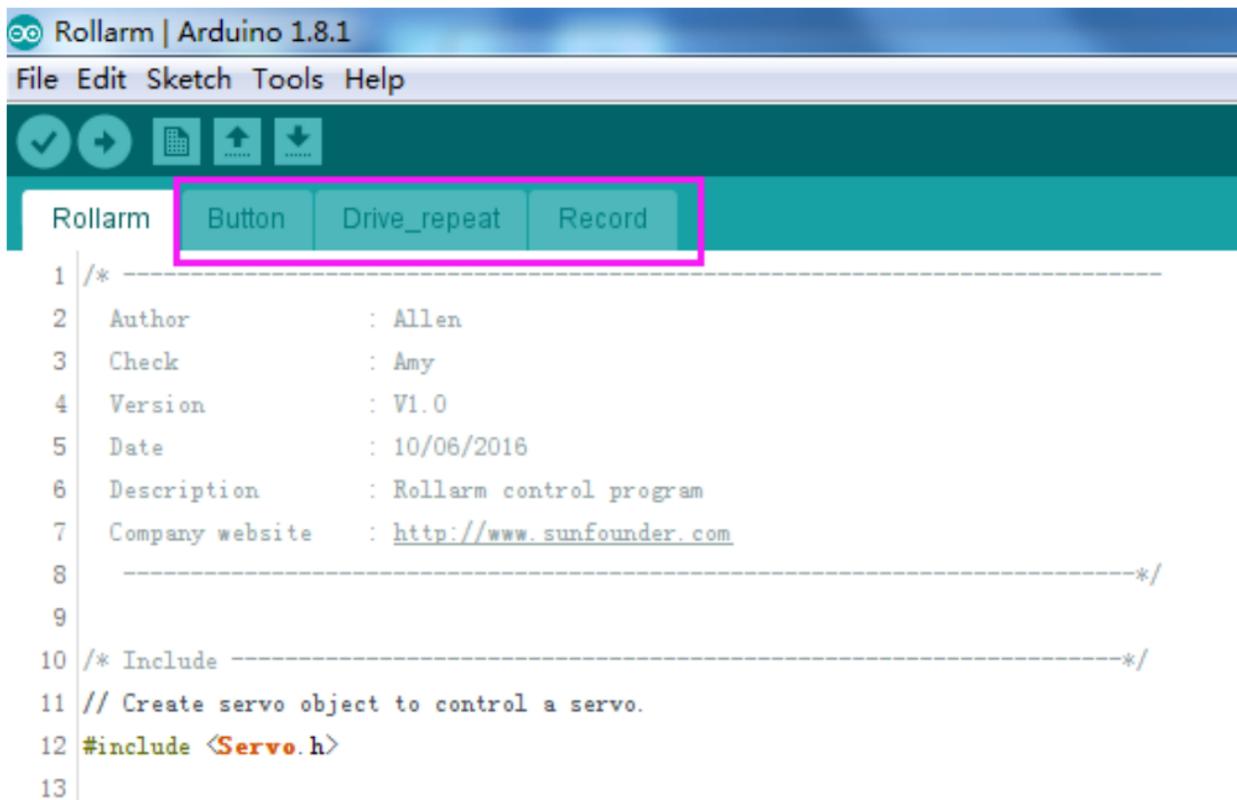
If you open the subprograms separately, a dialog box will pop up like this:



If a new folder has been created for the subprogram, please cut the subprogram file to the original directory Arduino code\Rollarm. Reopen the main program:



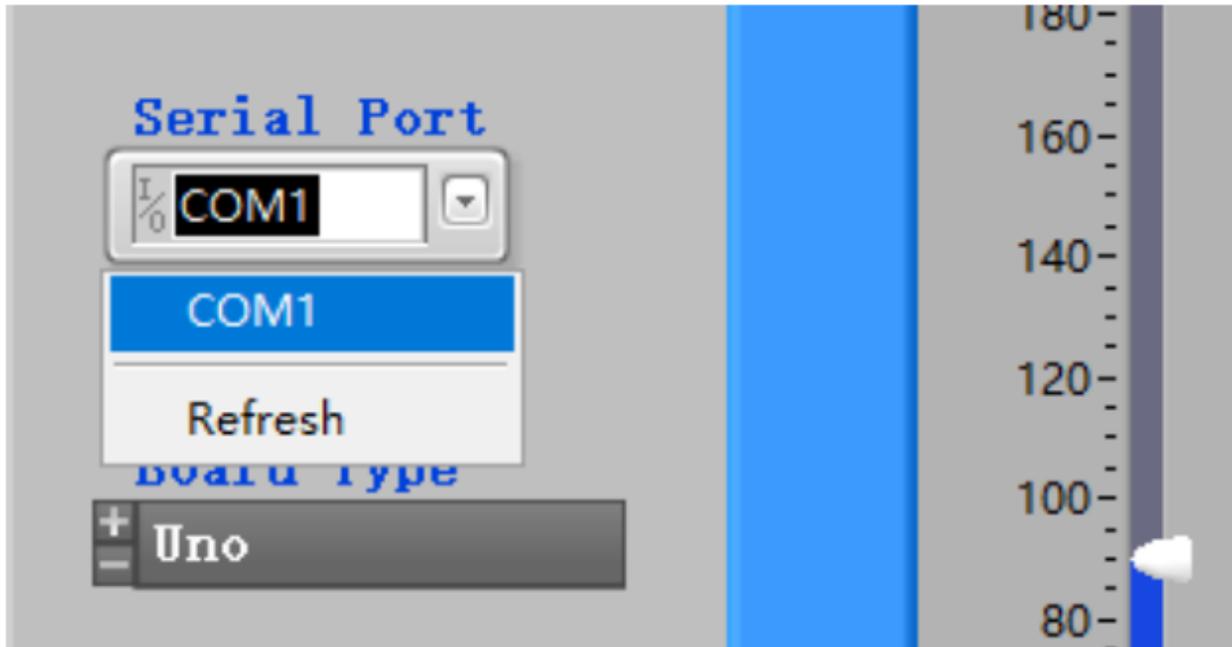
If a new folder has been created for the subprogram, please cut the subprogram file to the original directory Arduino code\Rollarm. Reopen the main program:



### 3.7.3 About the Labview software control

**Q1:** After powering on the Rollarm, why do the servos shake a little when there's no movement at all?

**A:** There may be something wrong with the Serial Port. For instance, the following condition may appear:



Turn off the Rollarm, power it on again, and reconnect the serial port to try.

**Q1:** The Rollarm is in a strange position when I click **Start** and it's in the automatic mode. Anything wrong?

**A:** Here no values is filled in the table yet. The first three axes are in  $0^\circ$ , and the last one is in  $90^\circ$ . You need to fill the correct value of the rotating angle first, and click **Start** to run.

## Auto Mode

	M1	M2	M3	M4
1	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="90"/>
2	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="90"/>
3	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="90"/>
4	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="90"/>
5	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="90"/>
6	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="90"/>
7	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="90"/>
8	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="90"/>
9	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="0"/>	<input type="text" value="90"/>

Interval time(ms)  Interval time(ms) 2

Before clicking **Start**, you need to fill in the rotating angle for each axis in different steps, and the interval time between steps. If you don't know the exact angle, you can shift to the manual mode and note down the angle values for each step, and then shift back to fill in. When all the steps above are done, you can click **Start** to let Rollarm perform the automatic control.

## 3.8 Thank You

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

### Particular Thanks

- Len Davisson
- Kalen Daniel
- Juan Delacosta

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

---

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

---



## **COPYRIGHT NOTICE**

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