Preface

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 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, the robot arm is easy to use and full of fun. 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 and find the tutorial under **LEARN**.

Free Support

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

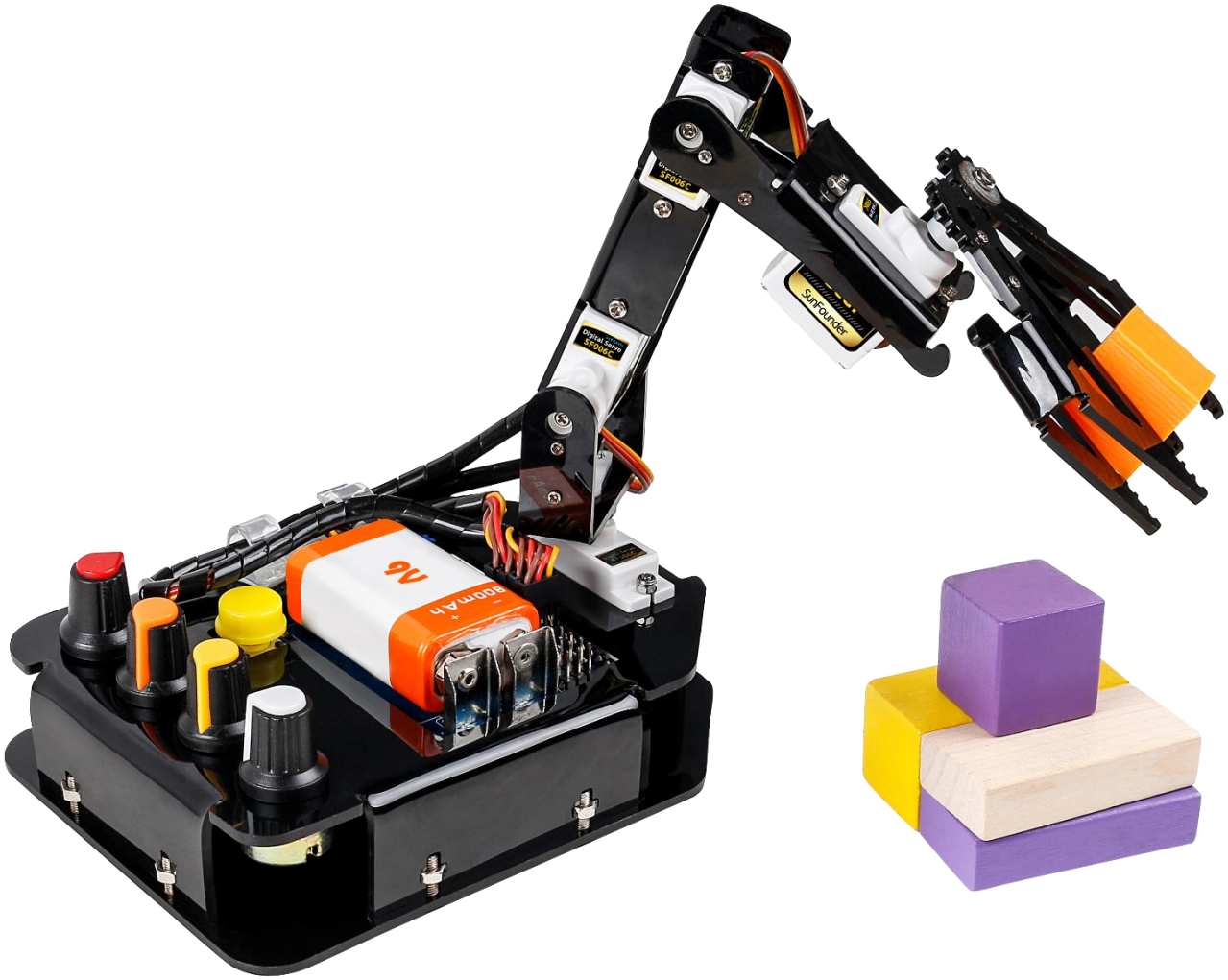
For NON-TECH questions like order and shipment issues, please **send an email to service@sunfounder.com**. You're also welcomed to share your projects on FORUM.

Contents

Introduction

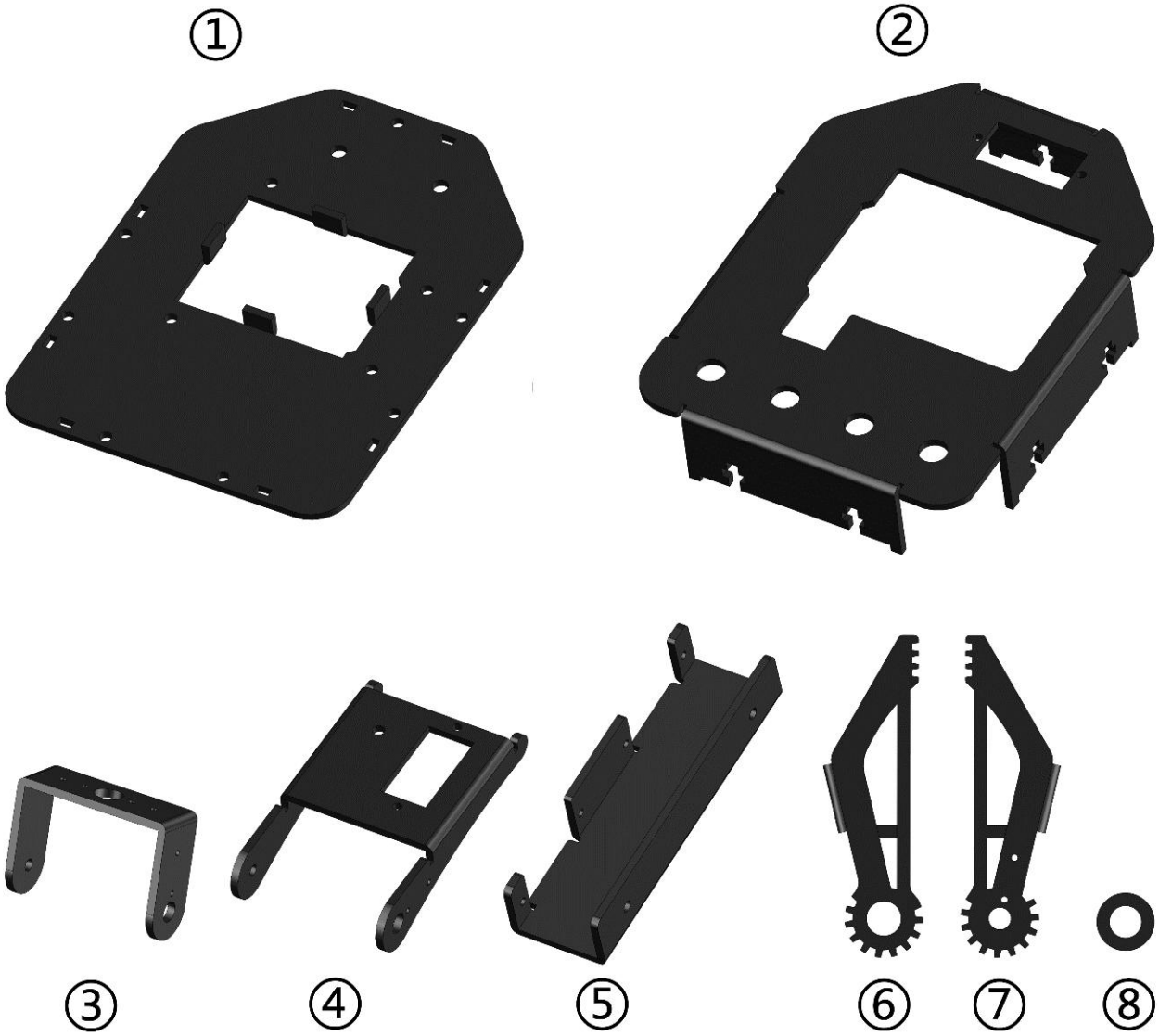
The Rollarm Kit is an interesting and useful learning tool for Arduino and robot hobbyists. With the structural plate and code based on Arduino, it enables users to learn programming from easy to difficult, control the mechanical arm freely and perform various fun operations!

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 our website. 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!

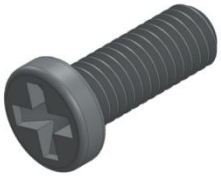
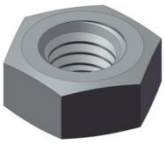
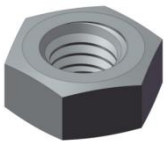


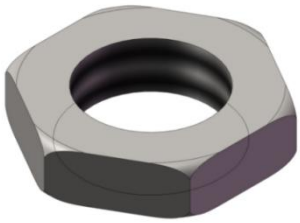
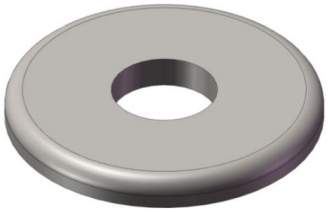
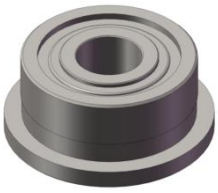
Components List

Structural Plate

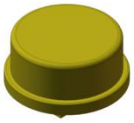
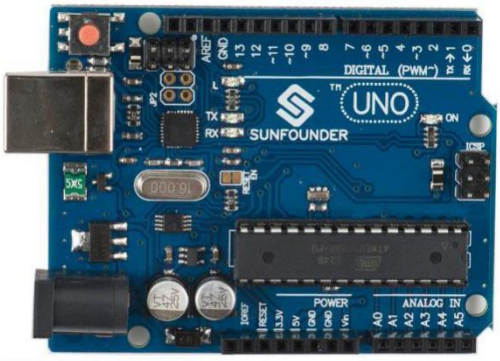


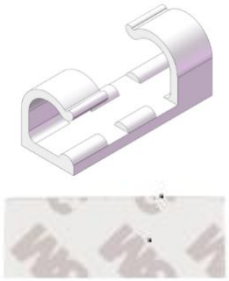
Mechanical Fasteners

Name	Parts	Qty.
M1.5*5 Self-tapping Screw		10
M2*8 Screw		10
M2 Self-locking Nut		10
M3*6 Screw		4
M3*10 Screw		10
M2.5*12 Screw		6
M2.5 Nut		6
M3 Nut		12

M3 Self-locking Nut		4
M7 Thin Nut		4
3*10*1Washer		1
M3*12 Aluminum Tube		1
φ3*φ8*2.5 Band Edge Bearing		1

Electronic Components

Name	Parts	Qty.
SF006C Servo		4
Potentiometer Button		4
Button		1
SunFounder UNO Board		1
Expansion Board		1

USB Type-B Cable		1
3M Non-skid pad		4
Cable Spiral Wrap		2
Cable Clip		2

Tools

Phillips Screw Driver		1
Cross Socket Wrench		1

Self-Provided Components

9V battery		1

How to control

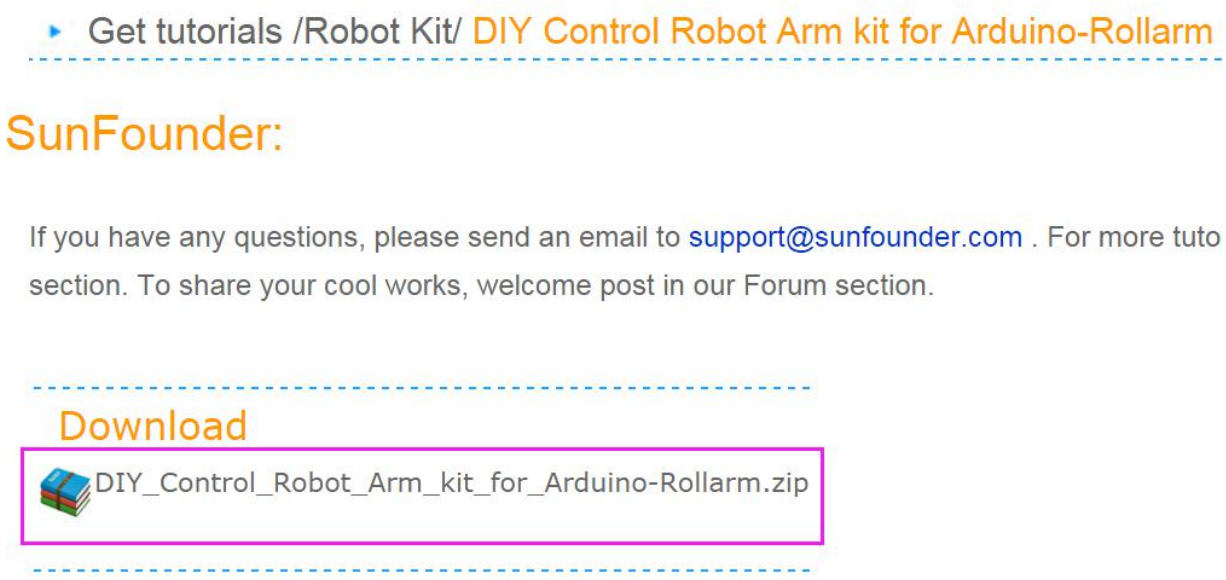
1. Install Arduino IDE
2. Assemble the Rollarm (Remember to **configure the servos** before assembly. Pay attention to the detailed operations)
3. Control with handle: **download** the Rollarm package and run the program (Indispensable step before operating the Rollarm)
4. Control with Labview: **download**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

Let's get started. Enjoy!

Getting Started with Software

Downloading Codes

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



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.

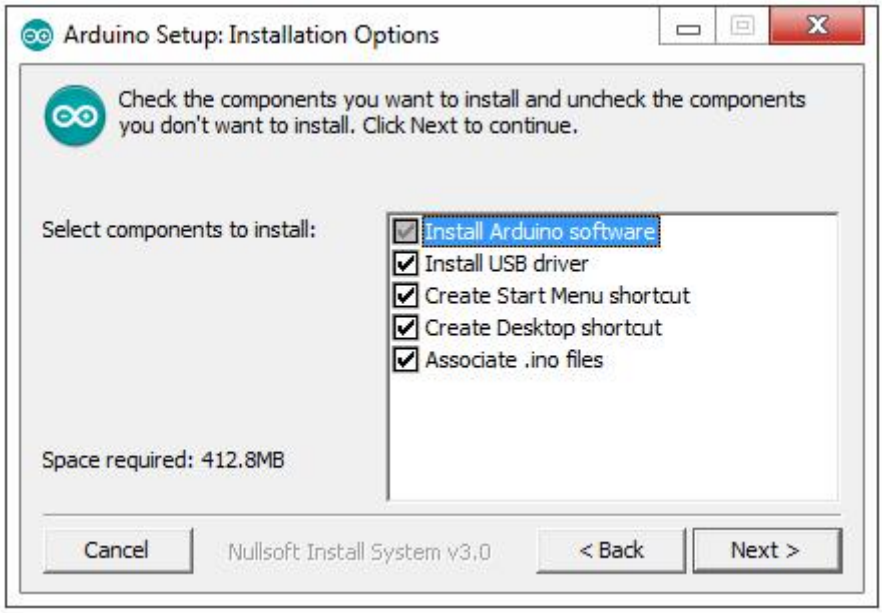
The code in this kit is written based on Arduino, so you need to install the IDE first. Skip it if you have done this.

Step 1: Go to the arduino.cc 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.

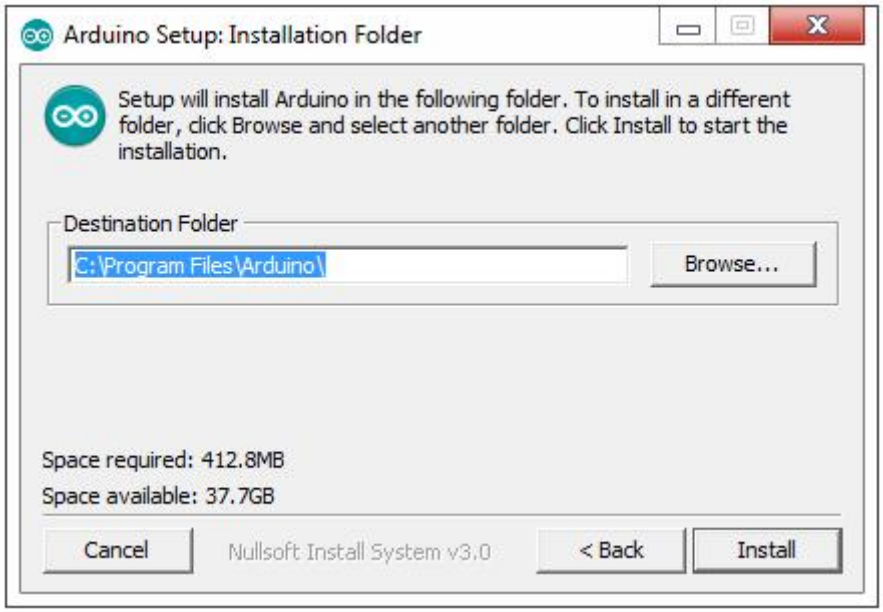
Step 2: Double click the exe. file and the following window will show up. Click I Agree. The following interface will show up.

Choose **Next.**



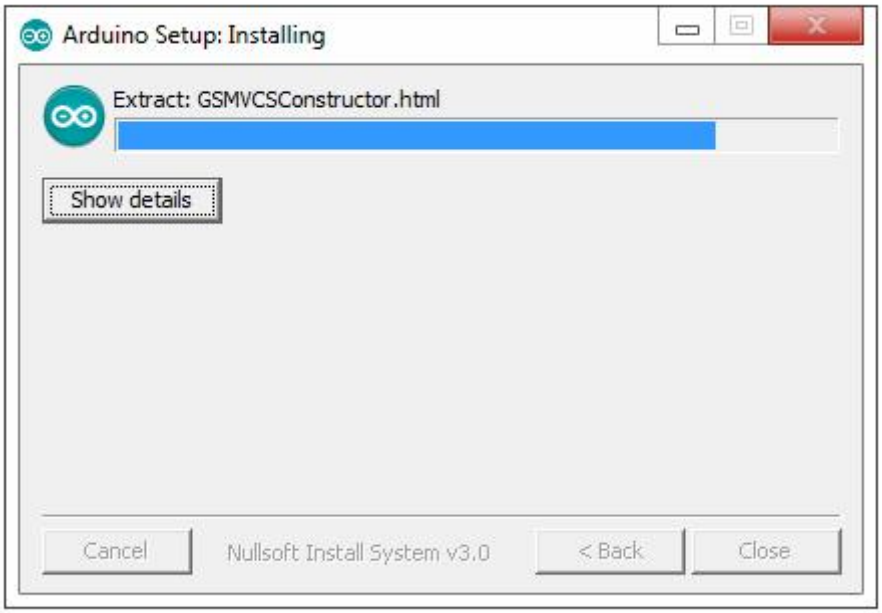
Click **Browse** to choose the installation path or enter a directory at the Destination Folder.

Click **Install.**

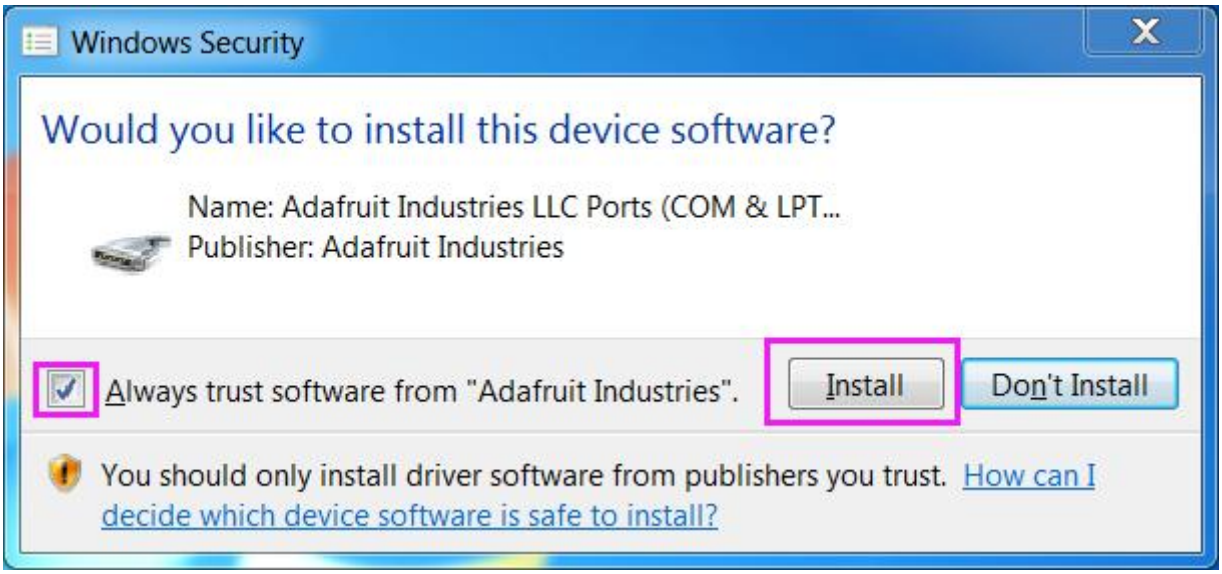


The following interface will show up.

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**.

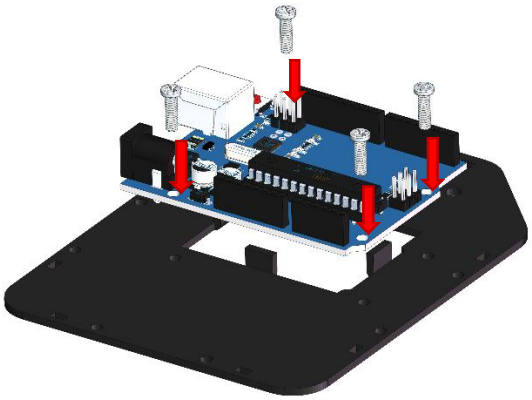
After the installation is done, click **Close**. Then an Arduino icon will appear on the desktop:

Assembly

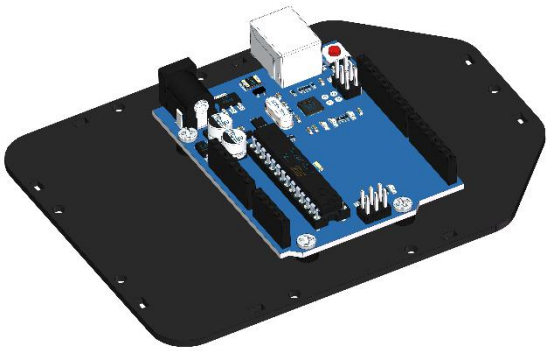
Base Bottom + Circuit Board

1) Assemble the UNO board to the plate ①with M2.5 x 12 screws and M2.5 nuts.

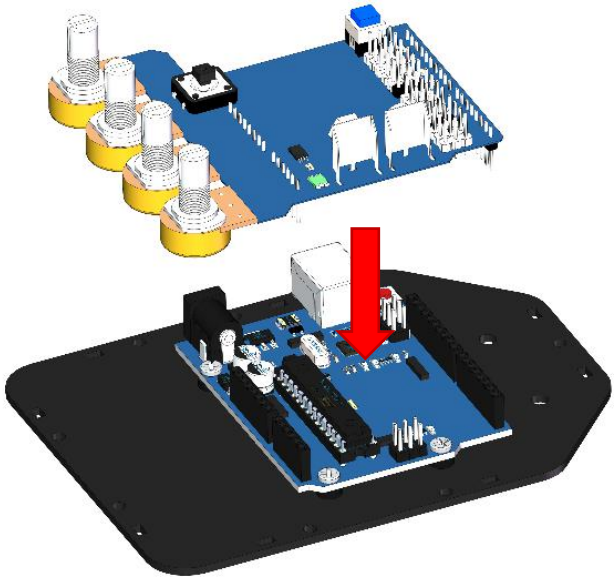
Note: The screws should not be too tight, to avoid damage to the board.



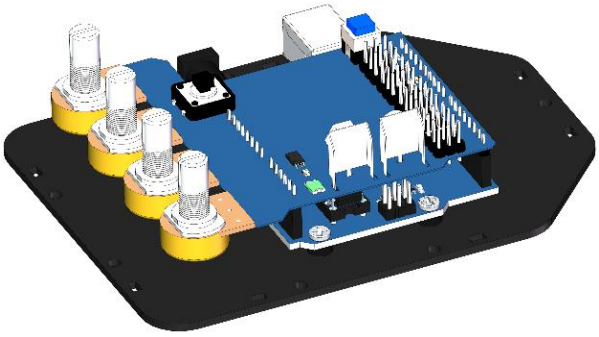
2) The effect is as follows:



3) Align the pin headers of the expansion board with the sockets of the UNO board.

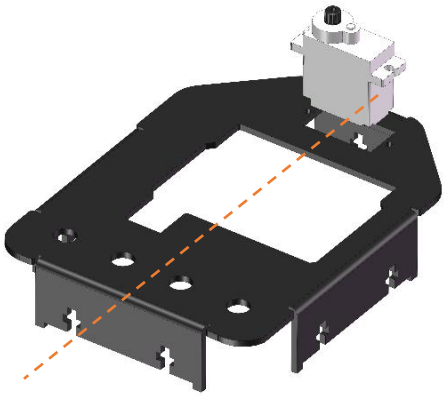


4) Insert the pin headers into the sockets.

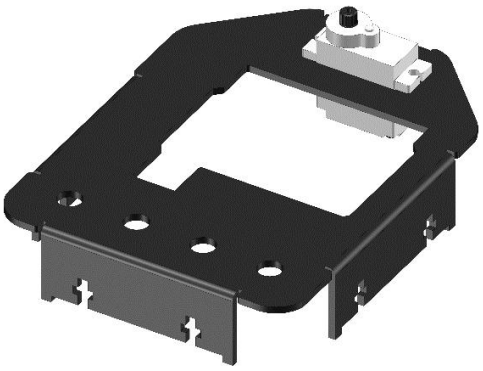


Base Upper Plate + Servo

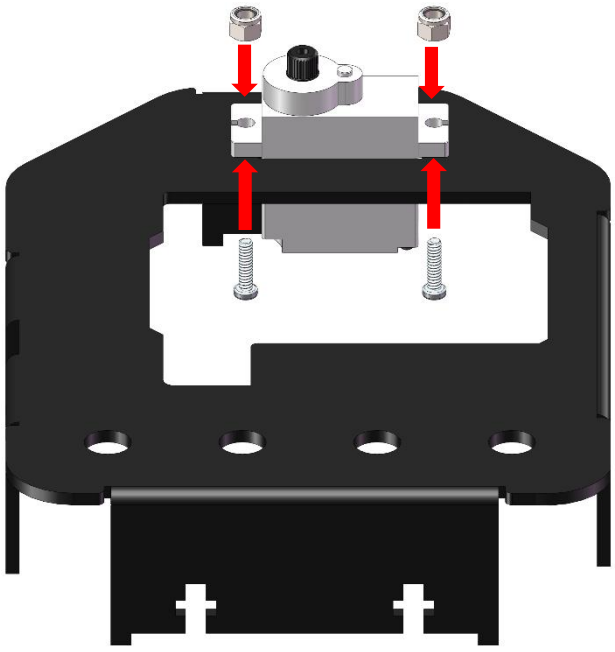
1) Align the servo with the slot of the **plate ②** with the rotating axis close to the middle line of the **plate ②**.



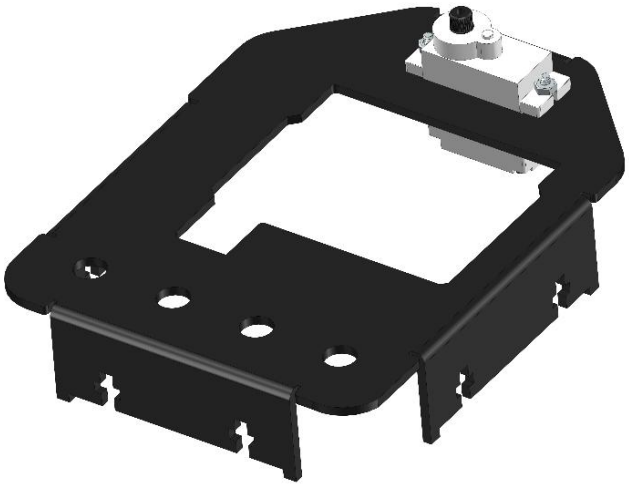
2) Insert the servo into the slot.



3) Place two M2 self-locking nuts above the hole of the servo and insert two M2 x 8 screws on the other side.

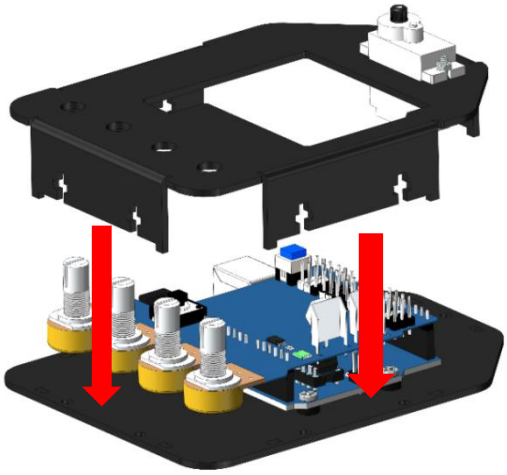


4) Tighten with cross socket wrench and screwdriver as follows.

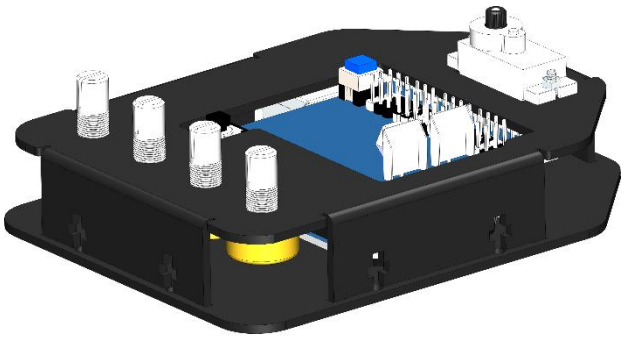


Base Fixing Plate + Base Upper Plate

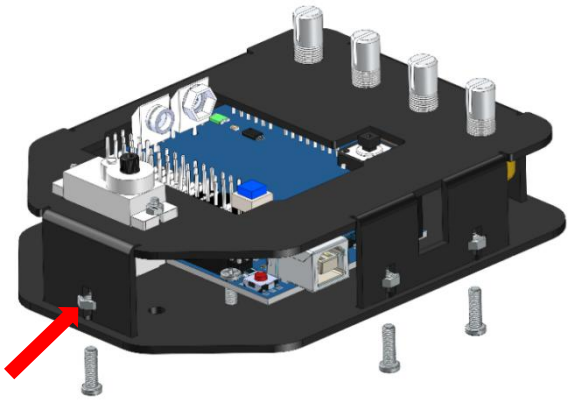
1) Align the hole of the **plate ①** with the bulges of the **plate ②.**



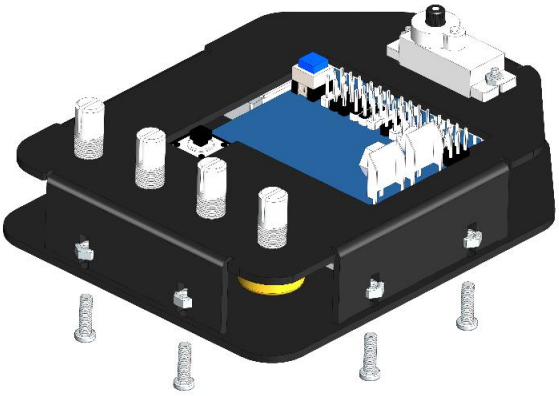
2) Insert the bulges into the holes.



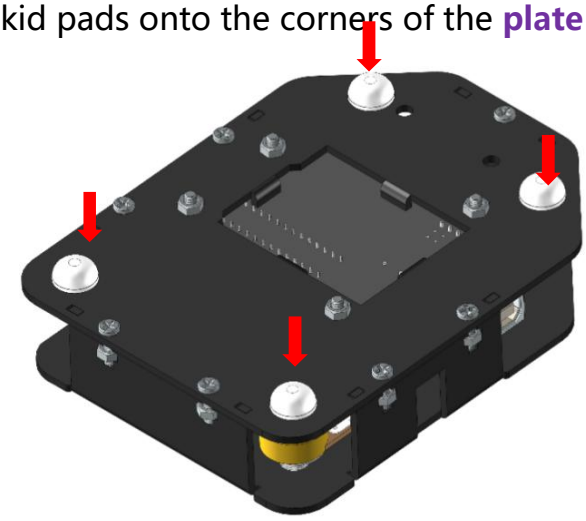
3) Place the M3 nuts in the hole in the **plate ②**. Insert M3 x 10 screws through the plate into the nut and fasten them with the screw driver.



4) Fix the other screws in the same way.

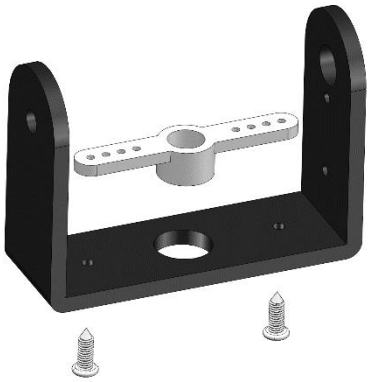


5) Sticking four non-skid pads onto the corners of the **plate ①.**

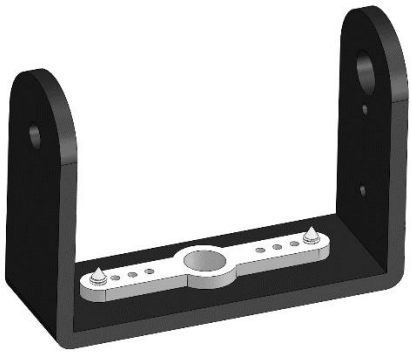


Installing Servo Rocker Arm

1) Align the 2-arm rocker arm with the hole of the **plate** ③. Insert two M1.5*5 self-tapping screws into the hole of the plate through the rocker arm, fasten them with the screwdriver.



2) The effect is as follows:

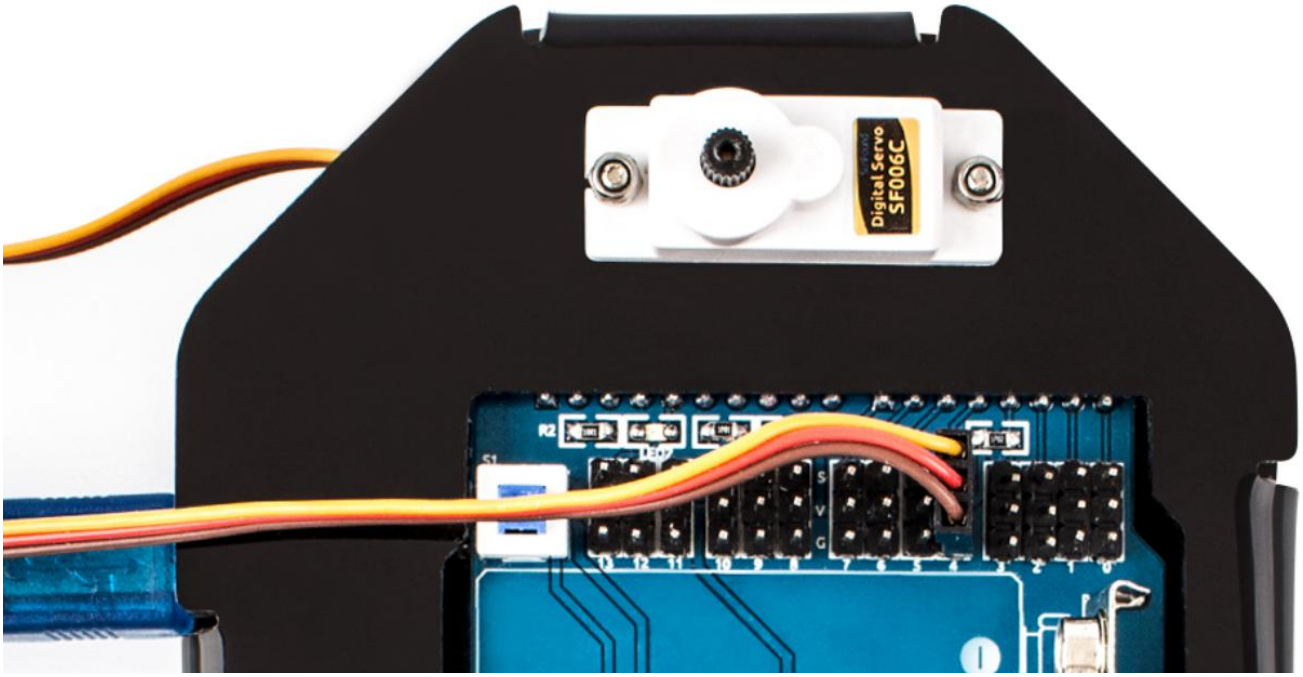


3) Similarly, fix a 1-arm rocker arm on the **plate** ⑦.

Base + Joint 1 Connecting Plate

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!

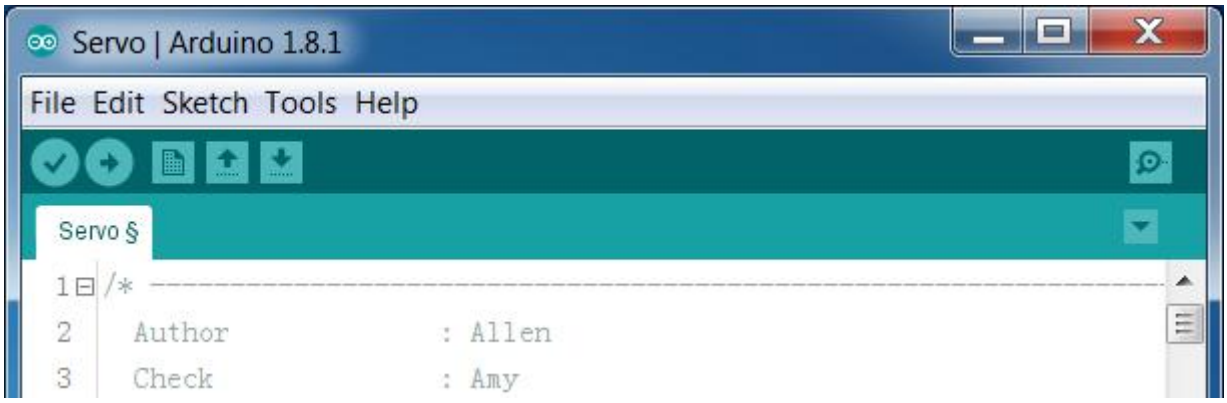
Step 1: Insert the servo wires into D4, connect the Servo Control Board to the PC via the USB cable, and the PC will automatically install the driver. The COM port connected will appear.



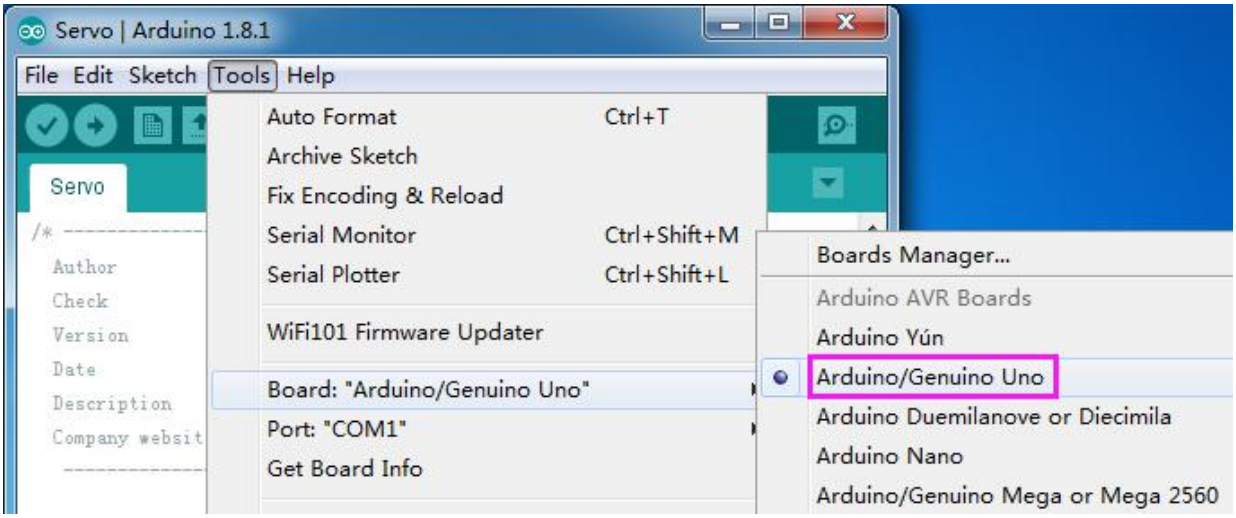
Connect the yellow, red, and brown wire to S, V, and G port as shown above.

Step 2: Connect the Rollarm to your computer with the Type-B cable: the driver will be installed automatically at that time, then you can see COMxx in device manager.

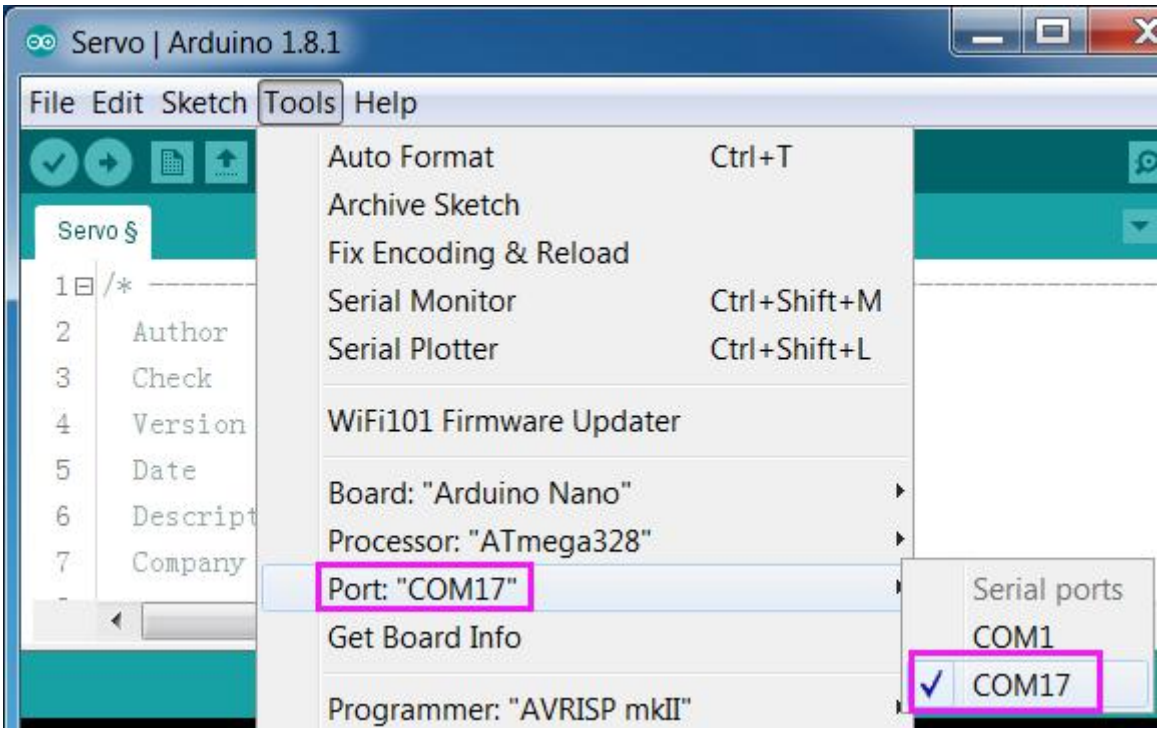
Step 3: Go to the folder **DIY Control Robot Arm kit for *Arduino-Rollarm/Arduino Code/Servo*** and open the file Servo.ino.



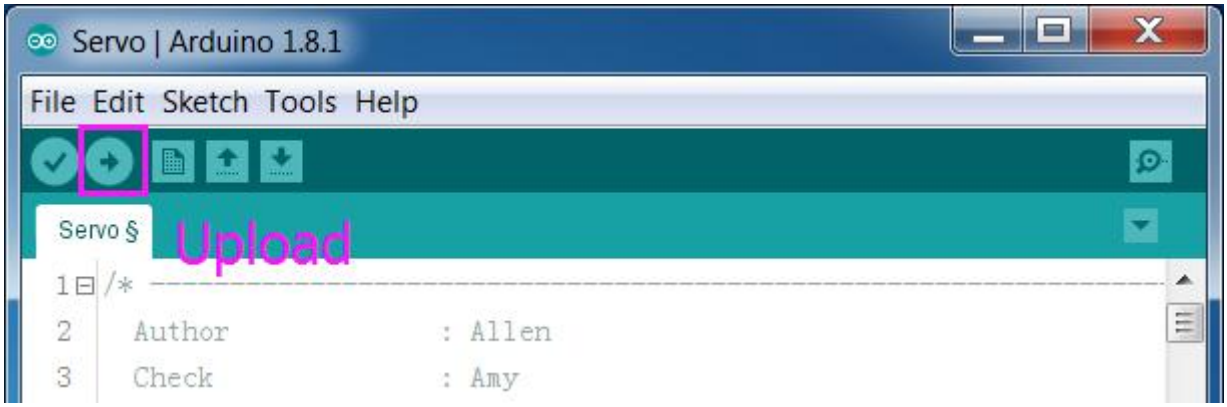
Step 4: Select the **Board.**



And **Port**

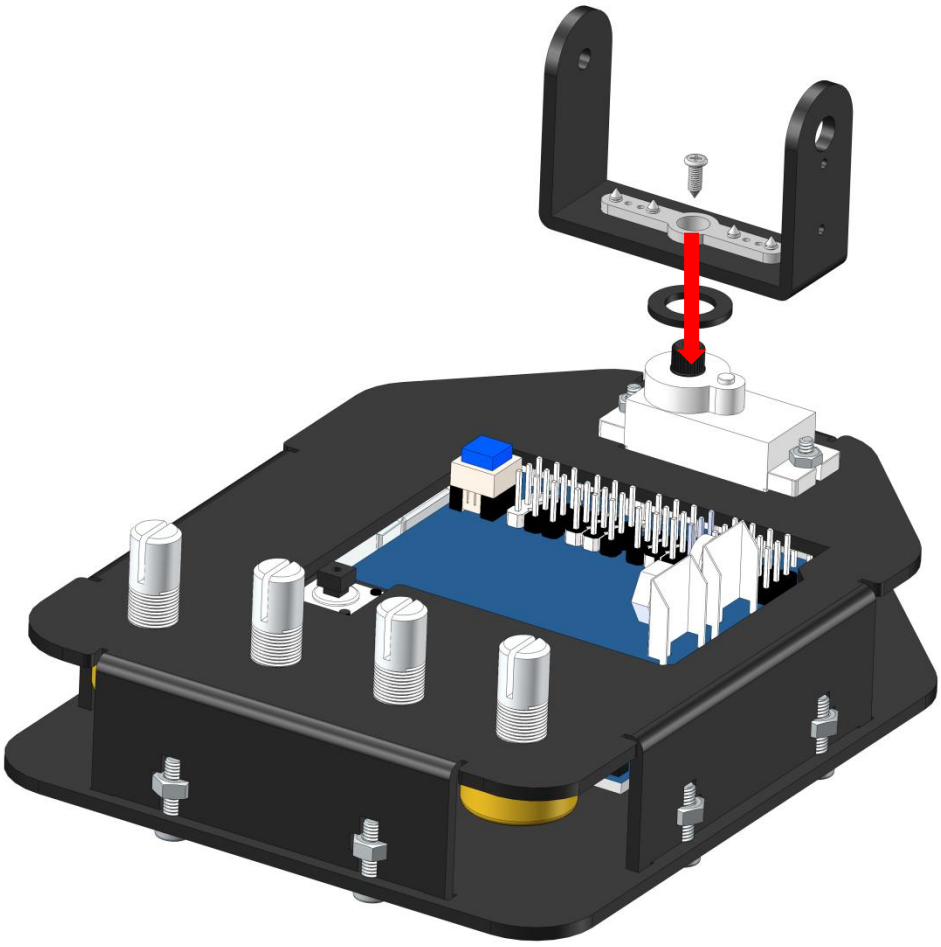


Step 5: Click **Upload**.



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.

1) Align the edge of the **plate** ③ with the edge of the **plate** ②. Fix the **plate** ③ and ⑧ and servo with the **matching smallest screws** of the servo.

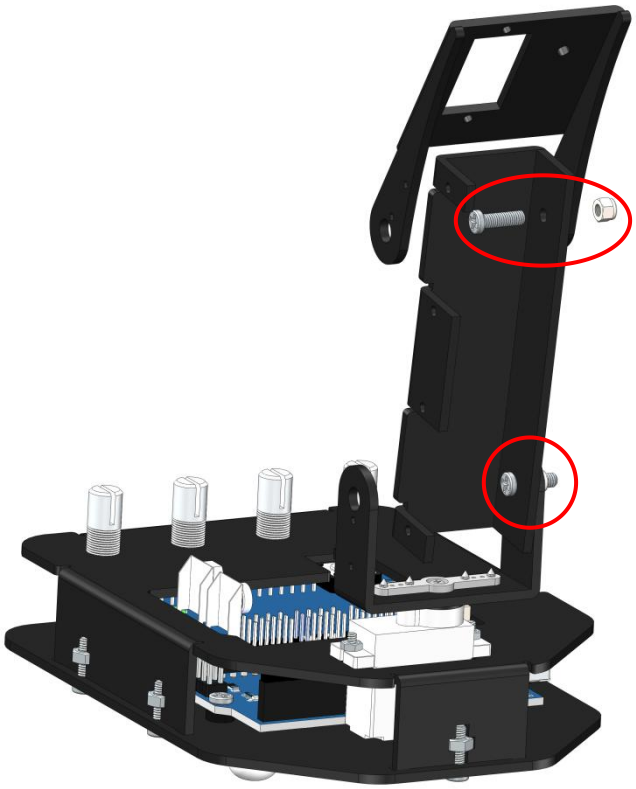


Joint 1 + Joint 2 + Joint 3

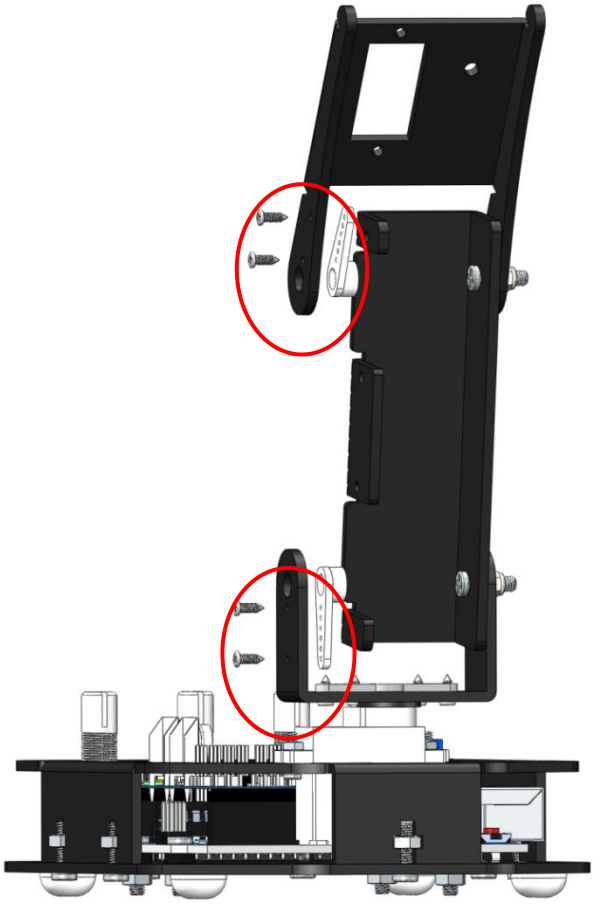
1) Fix the **plate** ③, **plate** ④ and **plate** ⑤ with an M3 x 10 screw and an M3 self-locking nut.

Note:

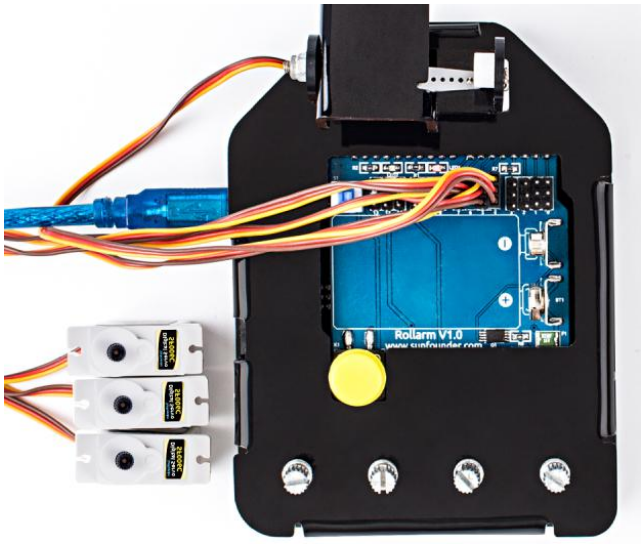
- Fix the self-locking nut with a cross socket wrench, and then tighten with a screwdriver.
- Can't tighten too much, need to allow **plate** ④ and **plate** ⑤ to move freely.



2) Fix two 1-arm rocker arm respectively to the **plate** ③ and ④ with M1.5 x 5 self-tapping screws.

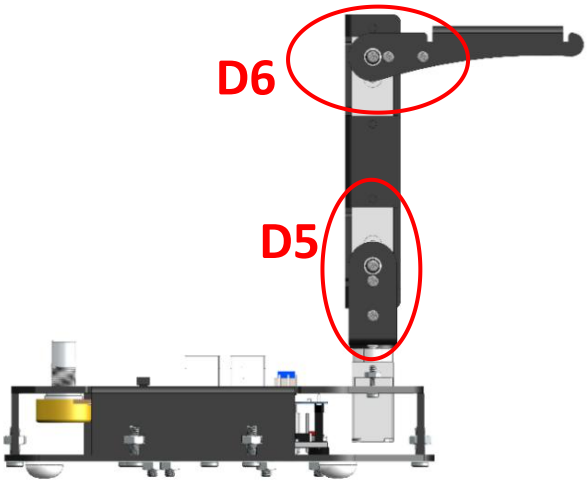


3) **Keep the USB cable connecting** and connect the 3 more servos to D5, D6, and D7 of the expansion board. At this time, the 3 servos are adjusted to the corresponding angle.



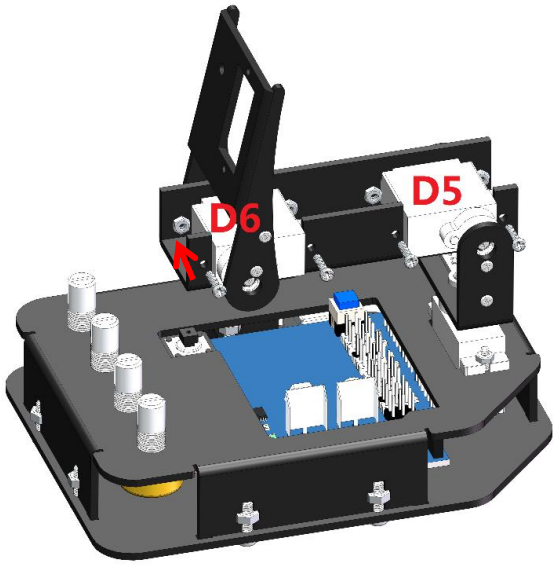
4) Insert the servo shaft into the rocker arm, fasten them with the **matching smallest screws.**

Note: In case the servos need to be re-calibrated later, please make sure that plate ③ and ⑤ are in the same line when fixing the servo of D5. When installing D6 servo, it is necessary to ensure the angle between plate ④ and ⑤ is 90°



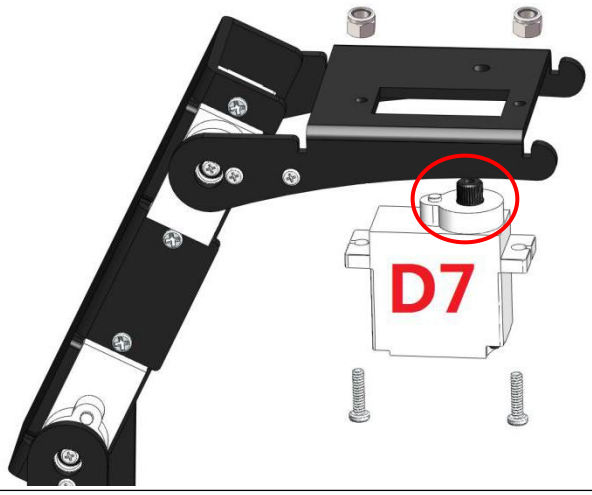
5) Fix the servos on the **plate** ⑤ with M2 x 8 screws and M2 self-locking nuts.

Note: Power off the control board in this step.



6) Fix the servo connected to the D7 to the **plate** ④ with the M2 x 8 screws and the M2 self-locking nuts.

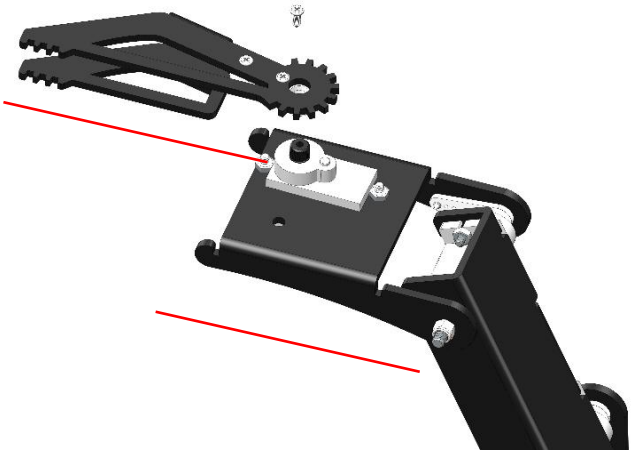
Note that the servo shaft is facing forward.



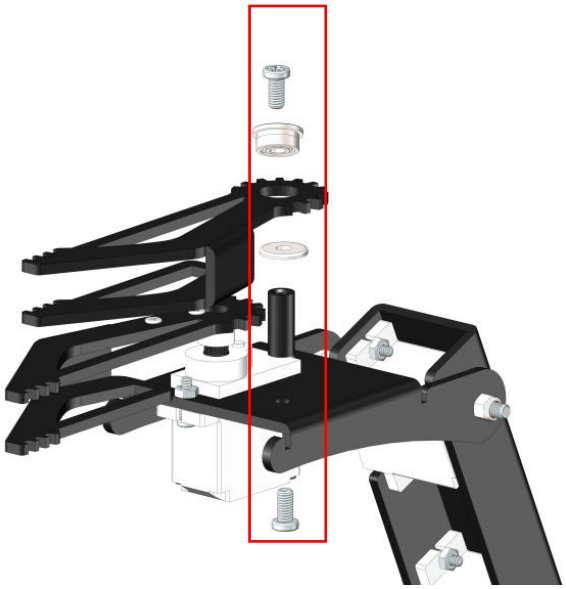
7) Insert the gripper into the servo arm, fasten them with the **matching smallest screws**.

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.



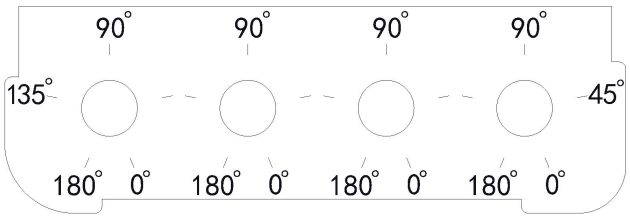
8) Fix the M3 x 12 aluminum column to the **plate** ④ with M3 x 6 screws, and insert the hole of 3*10*1 washers, **plate** ⑥ and band edge bearing by M3 x 6 screw, then fix on the aluminum column.



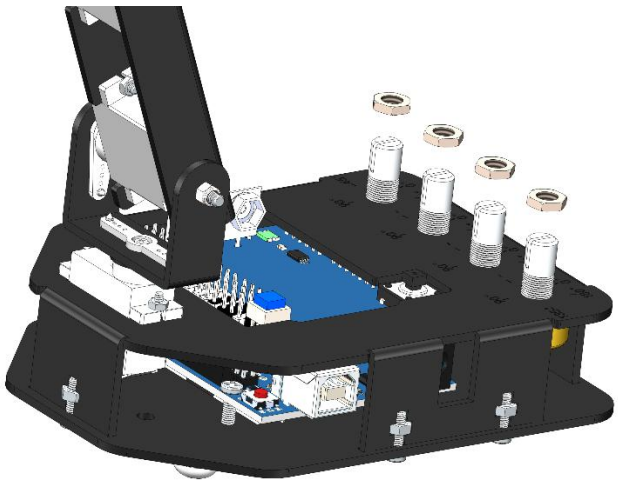
9) Make sure that the inner sides of the **plate** ⑥ and ⑦ are parallel and the gears meshed.

10) 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)

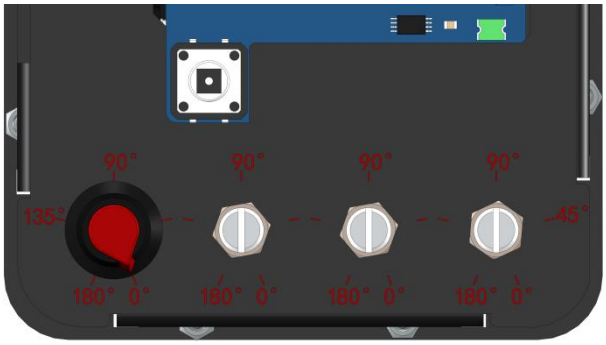


11) Fix four potentiometers and scale paper with four M7 thin nuts.



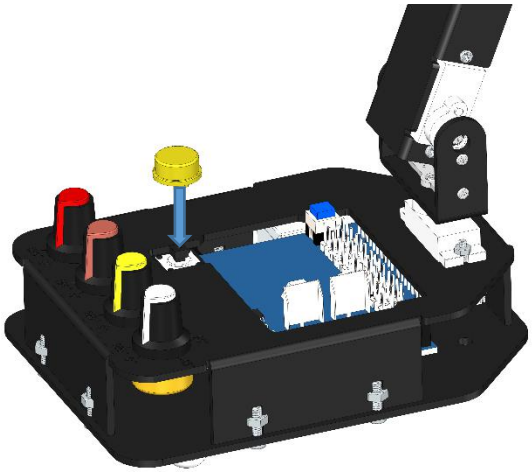
12) Rotate the potentiometer **clockwise** to the end, and put the button onto the potentiometer and press to tighten them.

Note: align the bulged part of button with the 0° position.

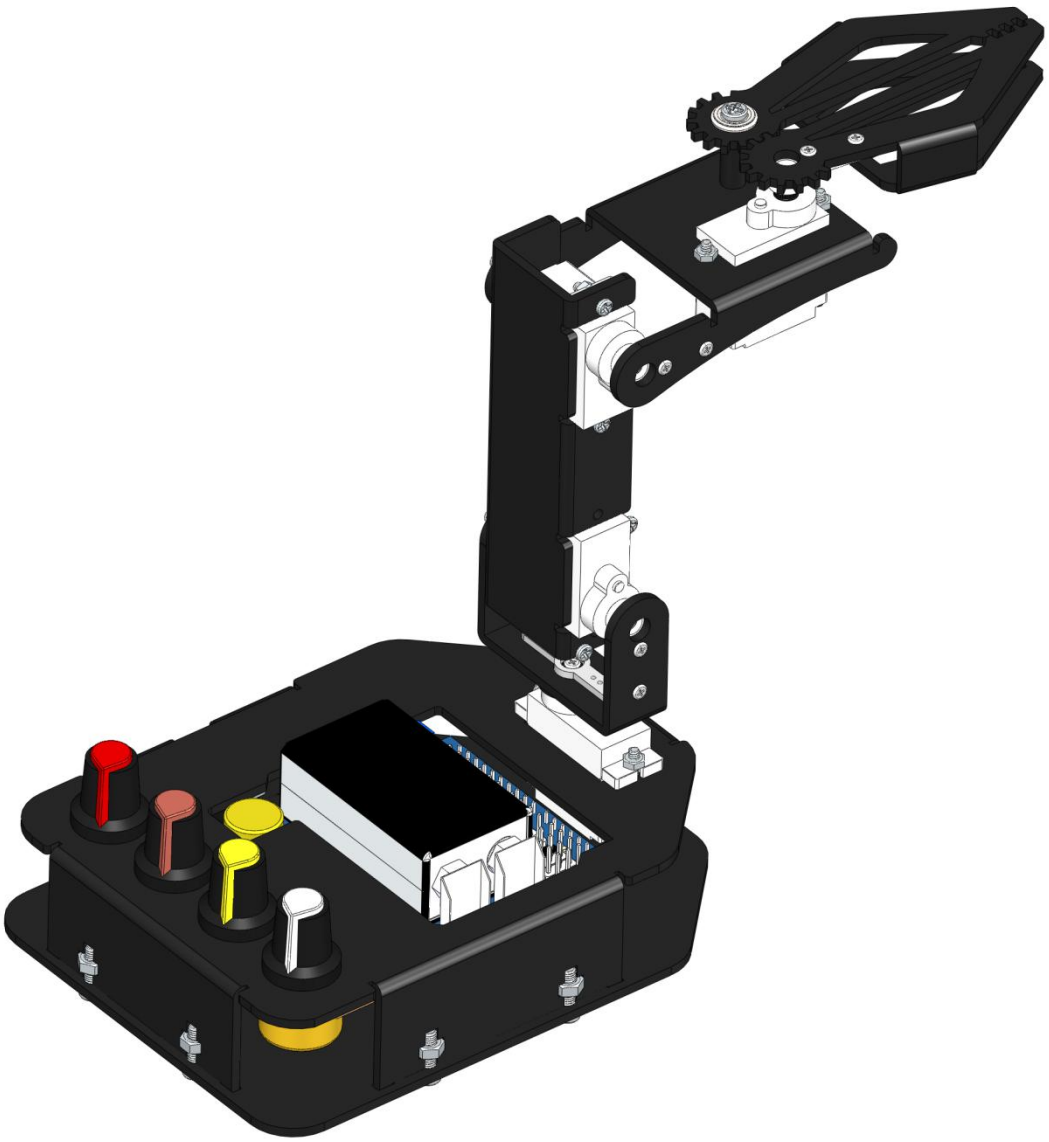


13) Install the other three pot buttons in the same way.

14) Install the button.



15) Install 9V battery on the expansion board.



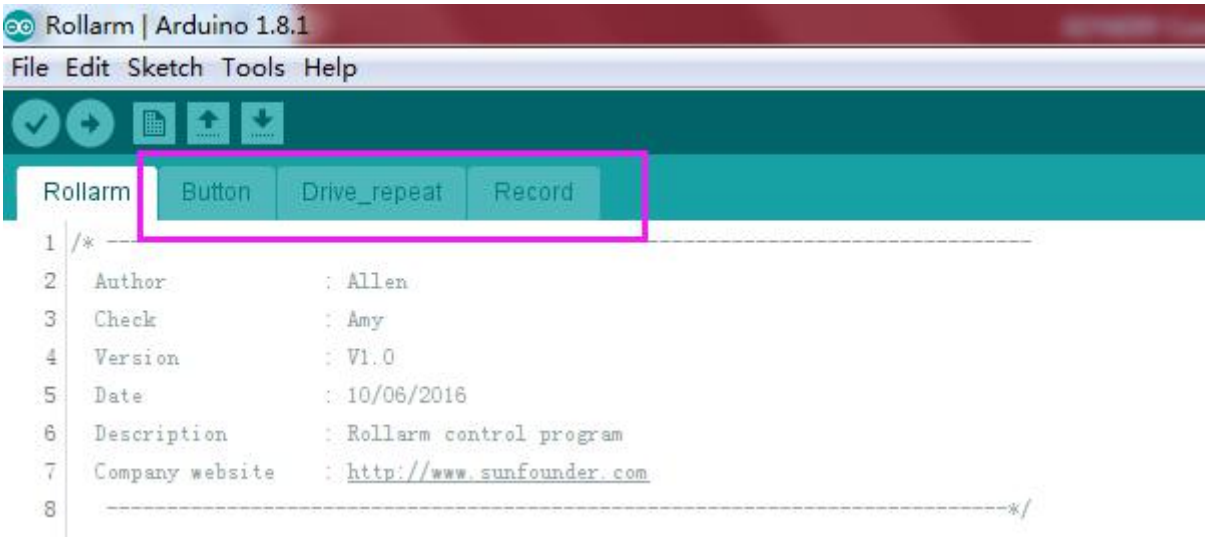
Control the Rollarm

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.

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:

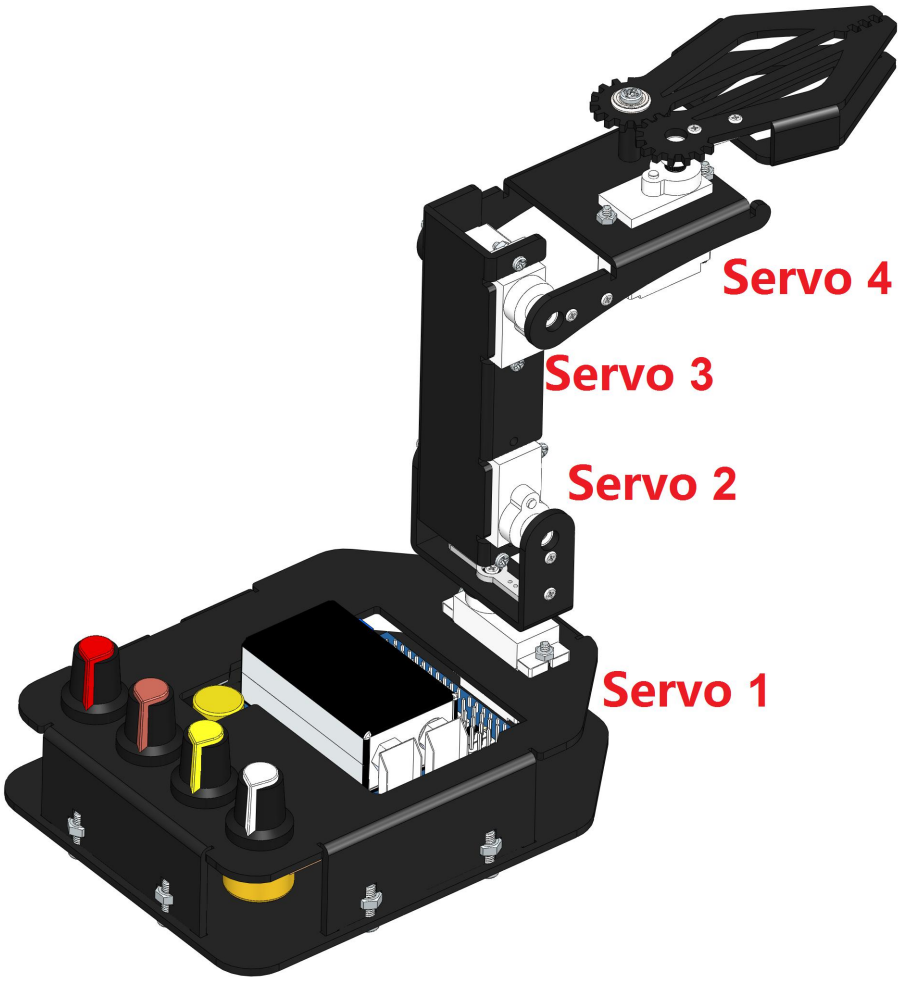


Step 2: Select the corresponding board and port, then click **Upload.**

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 4, the yellow to control Servo 3, the orange to control Servo 2, and the red one to control Servo 1.

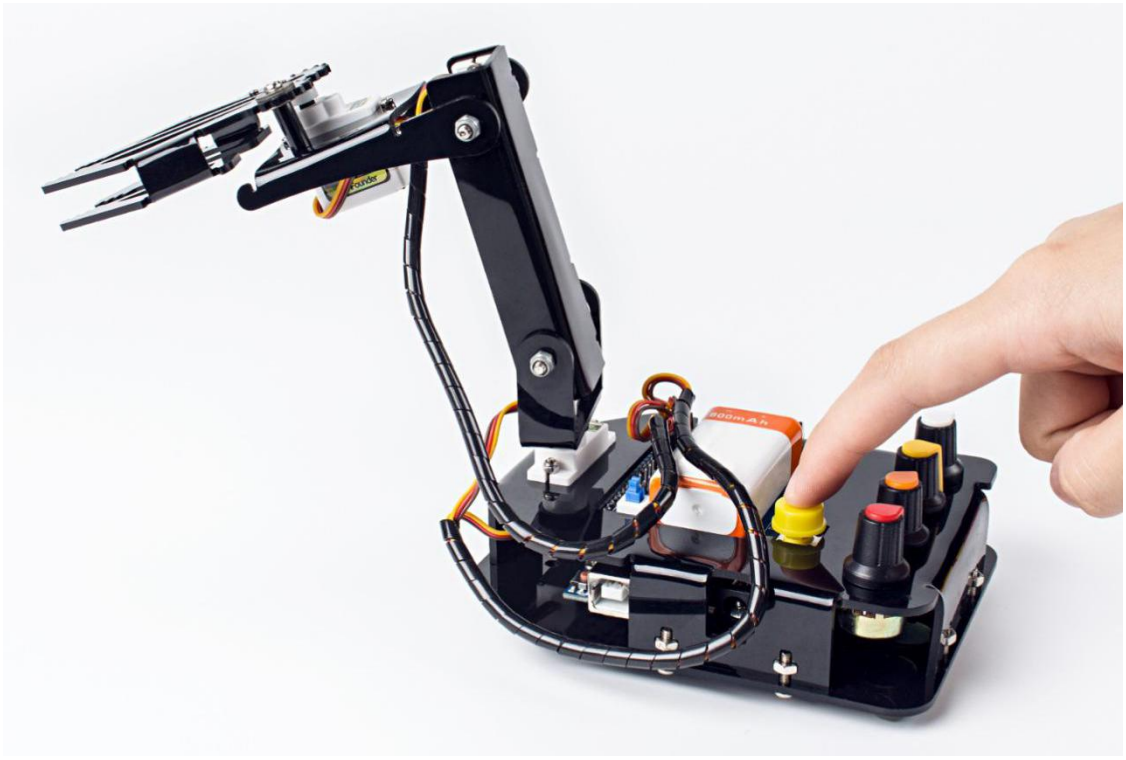


➢ Record behavior

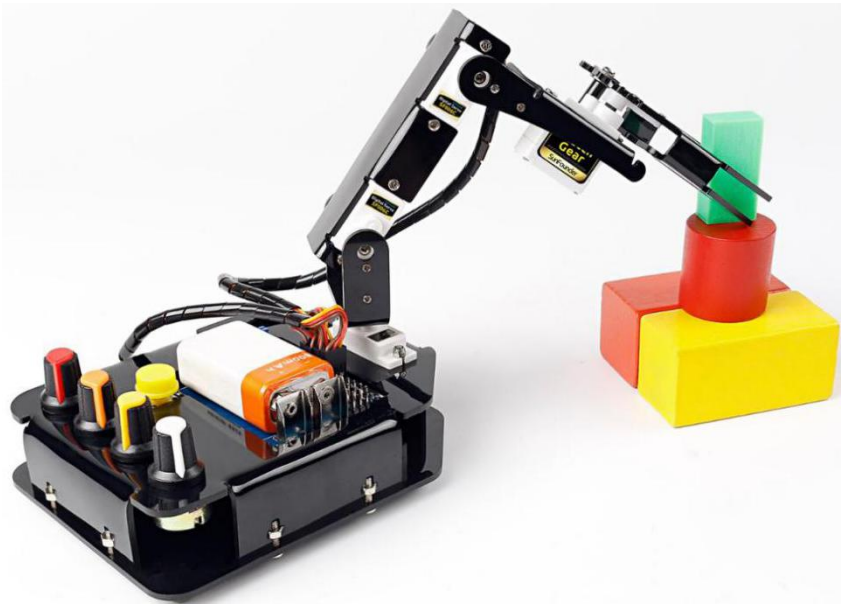
With the handle, the Rollarm can record its behaviors:

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 ot 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();
    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);
    }
  }
```

PC Control (Labview)

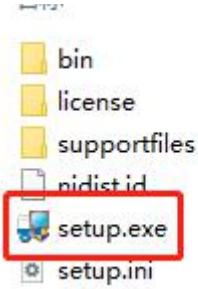
➢ Installing Labview Software

For this kit, we use the Labview software for control on PC. If you have other better options, welcome to share by post under **FORUM** on our website www.sunfounder.com.

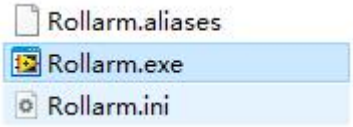
Download the Labview package in the link below:

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

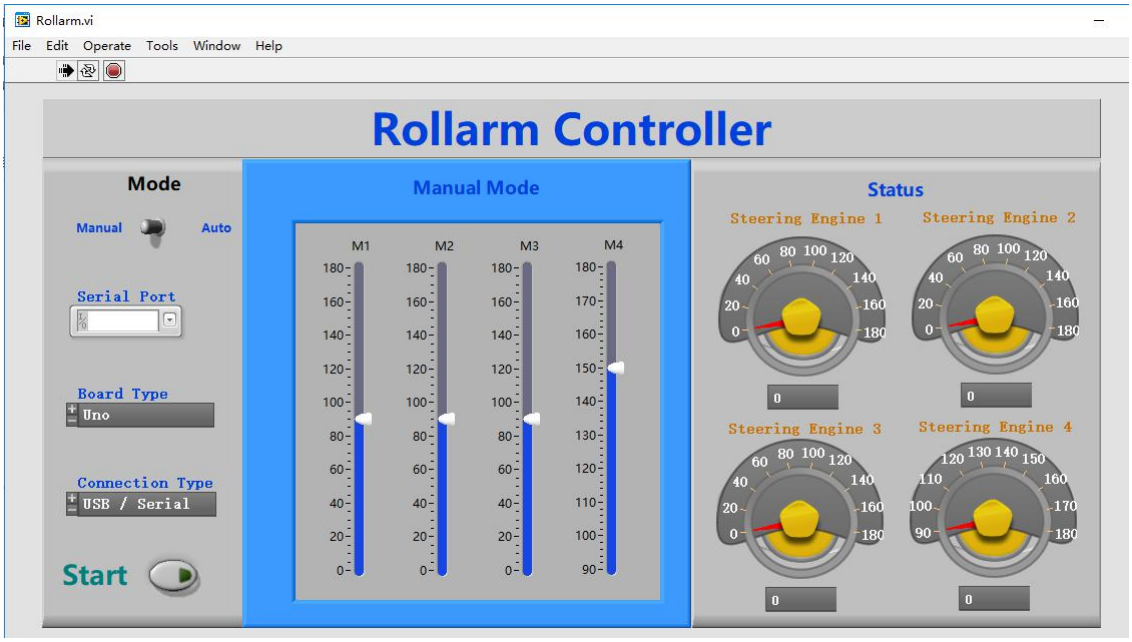
After downloading, open it and you will see the folder *Rollarm's Installer*. Then find the setup file as shown below:



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.

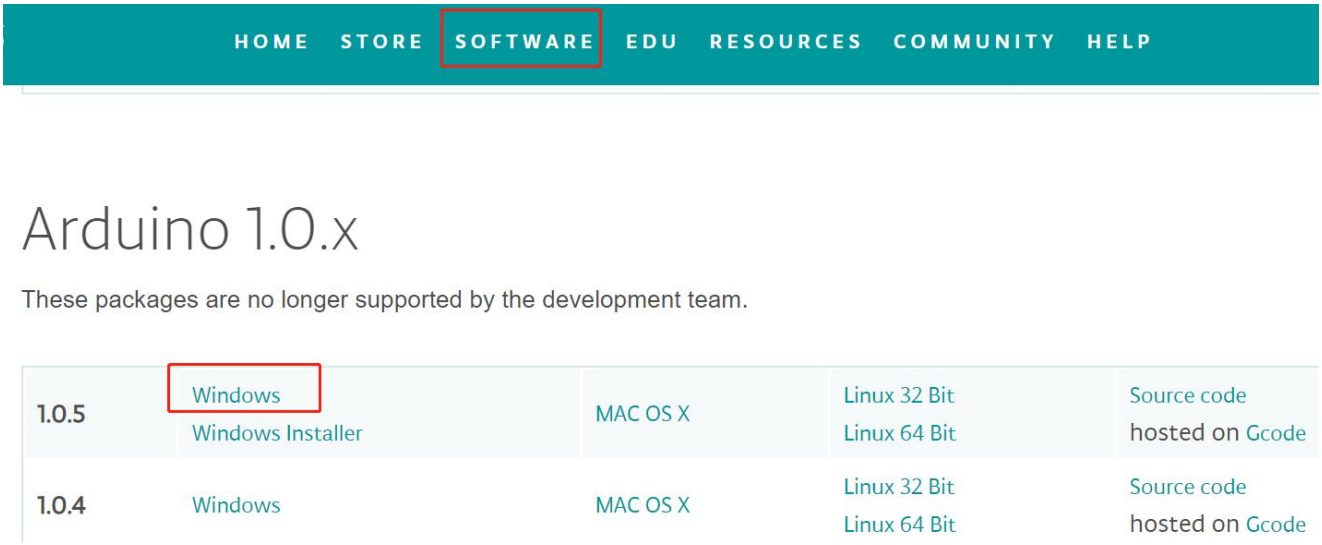


➢ Upload the Code

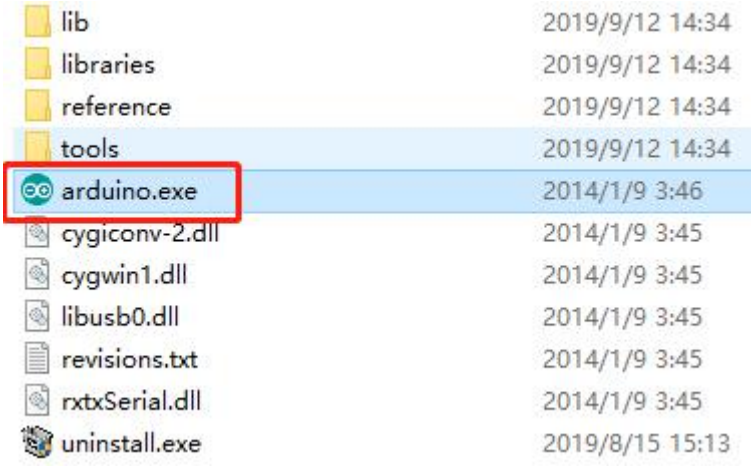
Before using the Rollarm Labview software, flash the control codes into the Rollarm robot, and the steps are as follows.

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.



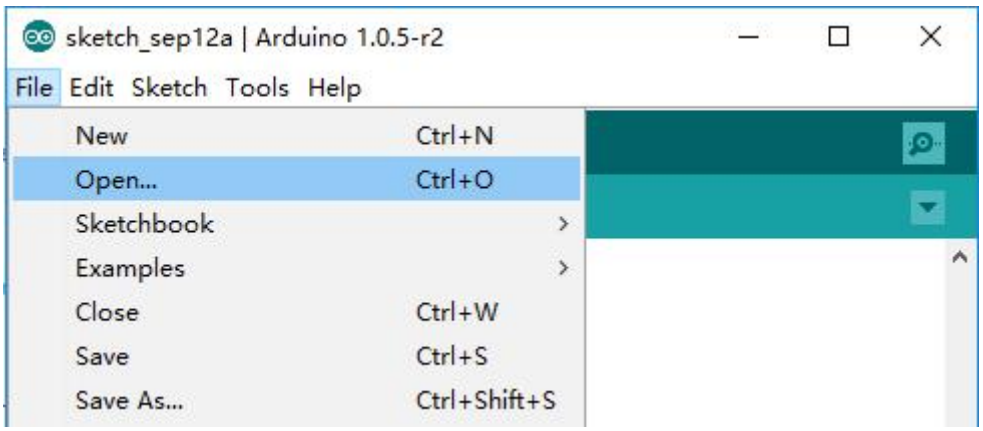
Step 2: After downloading and unzipping them, you need to double click the arduino.exe to open it.



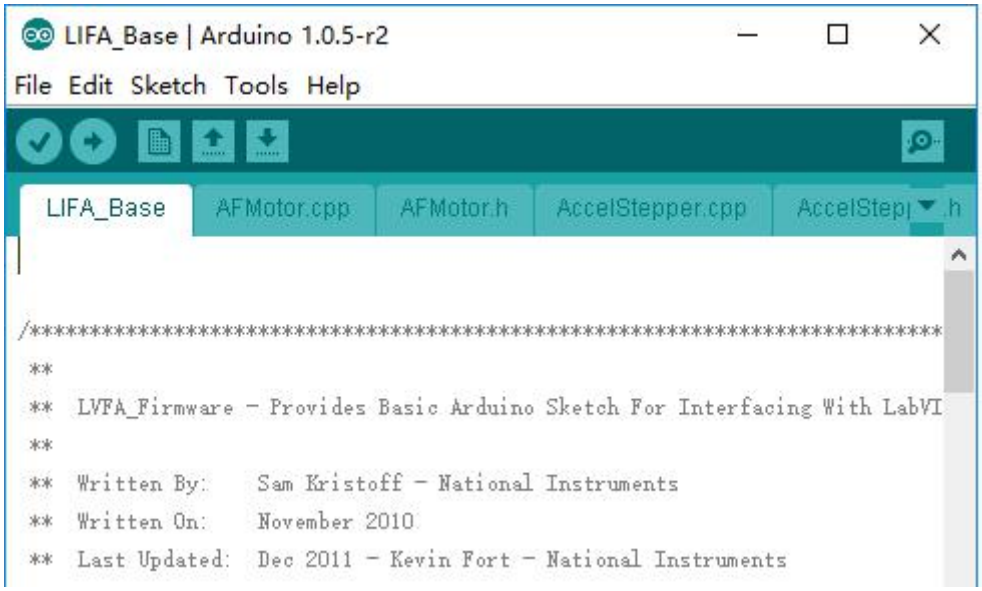
Step 3: Click File- >Open, and you can find the **LIFA_Base.ino on the** path

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.



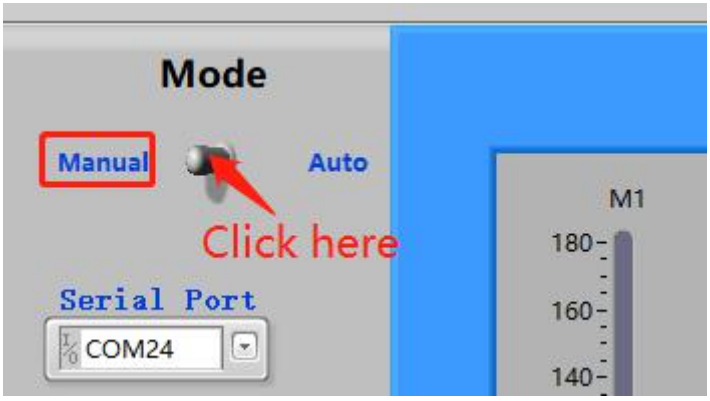
DO NOT unplug the USB cable at the moment.

➢ Using the Software

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

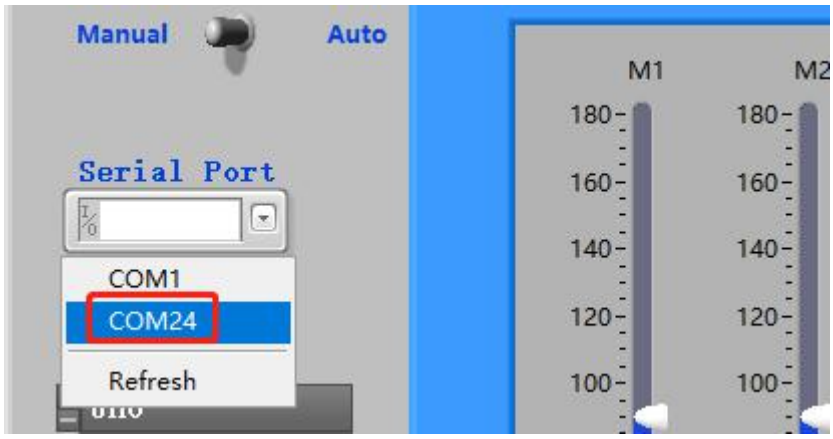
1) Manual Mode

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



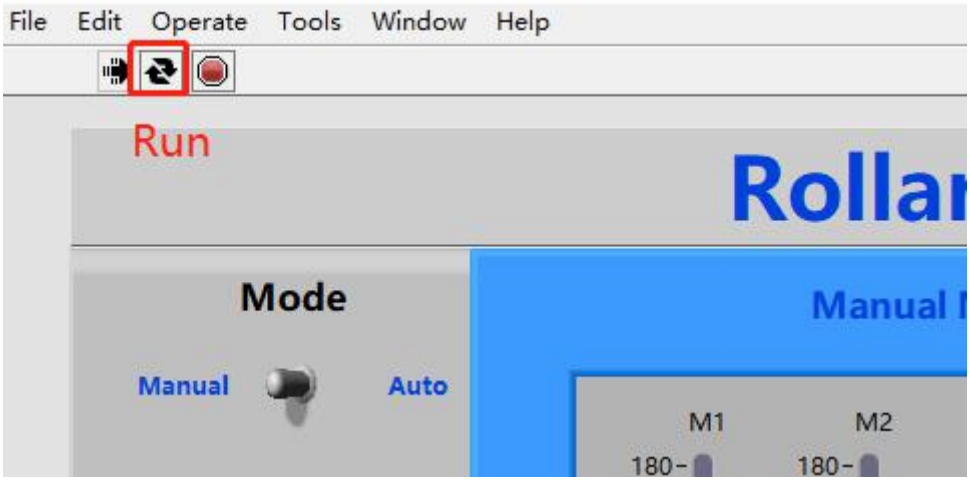
Step2: Click the inverted triangle icon for **Serial Port**, select the port according to you COM port. Here is COM24, which varies for different computers.

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

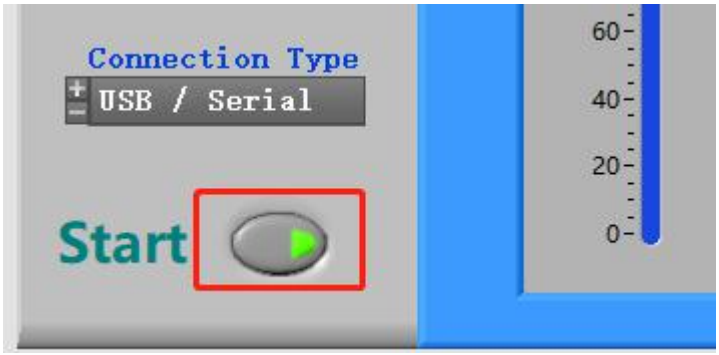


Step 3: Select the **Board Type (Uno)**and **Connection Type(USB/Serial)**.

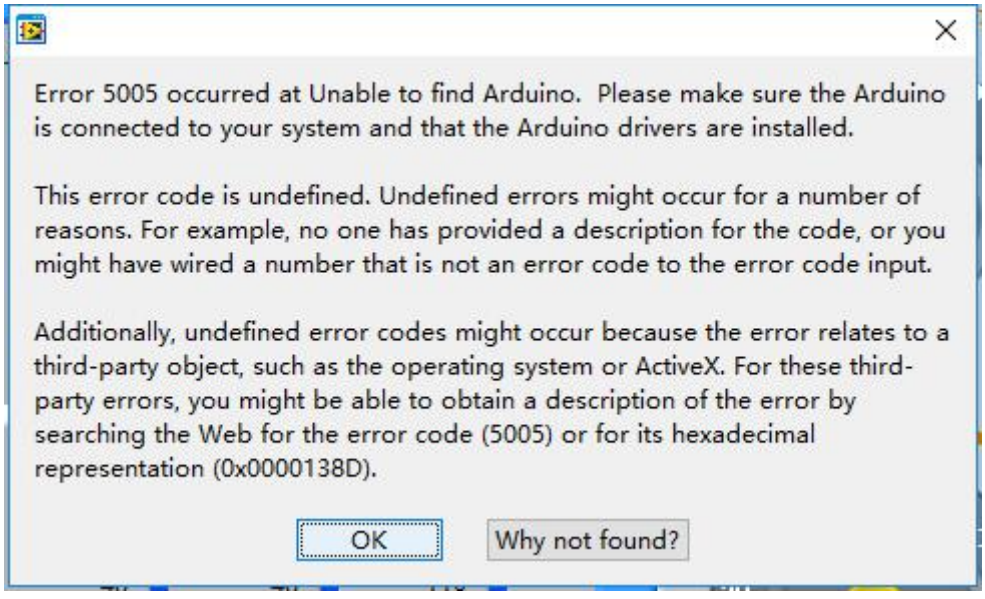
Step 4: There are **three** small icons at the top left. Click the middle on to run the software.



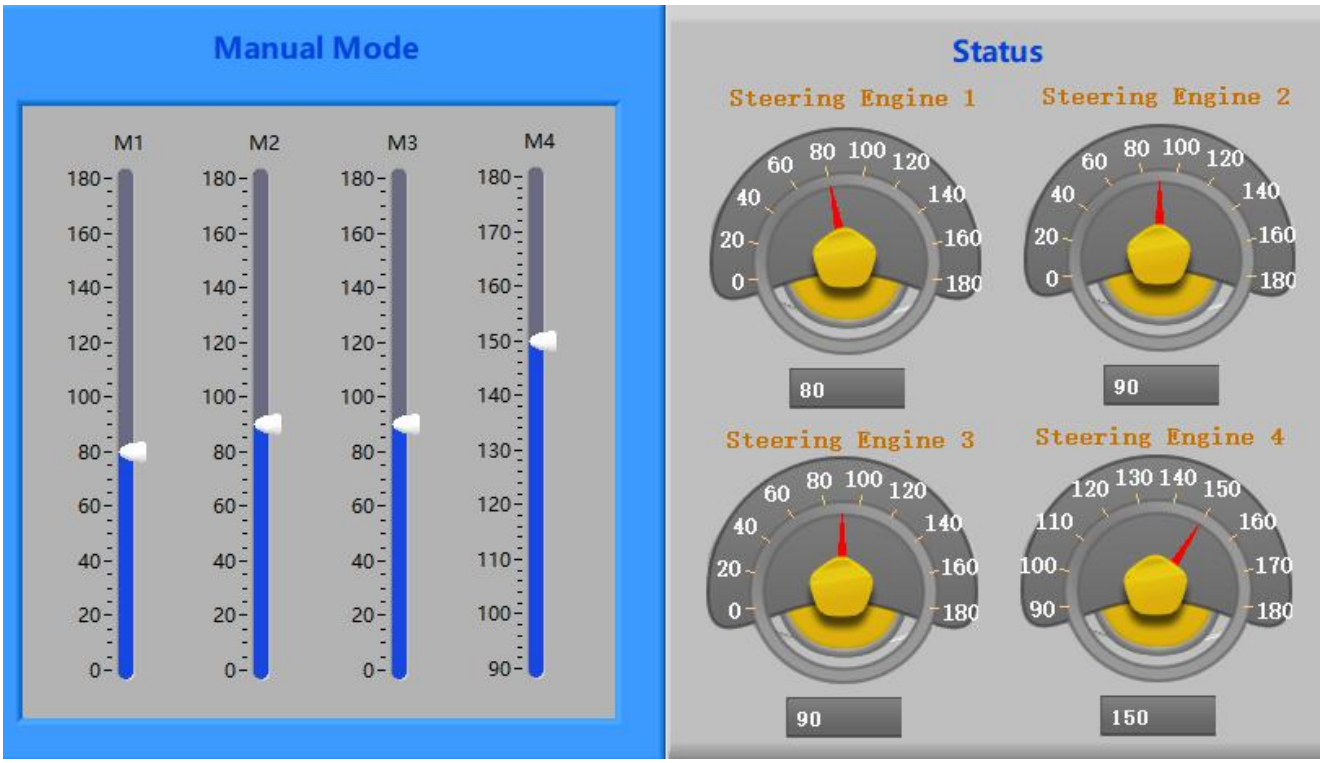
Step5: Click **Start**, and the button will change from dark to light **green.**

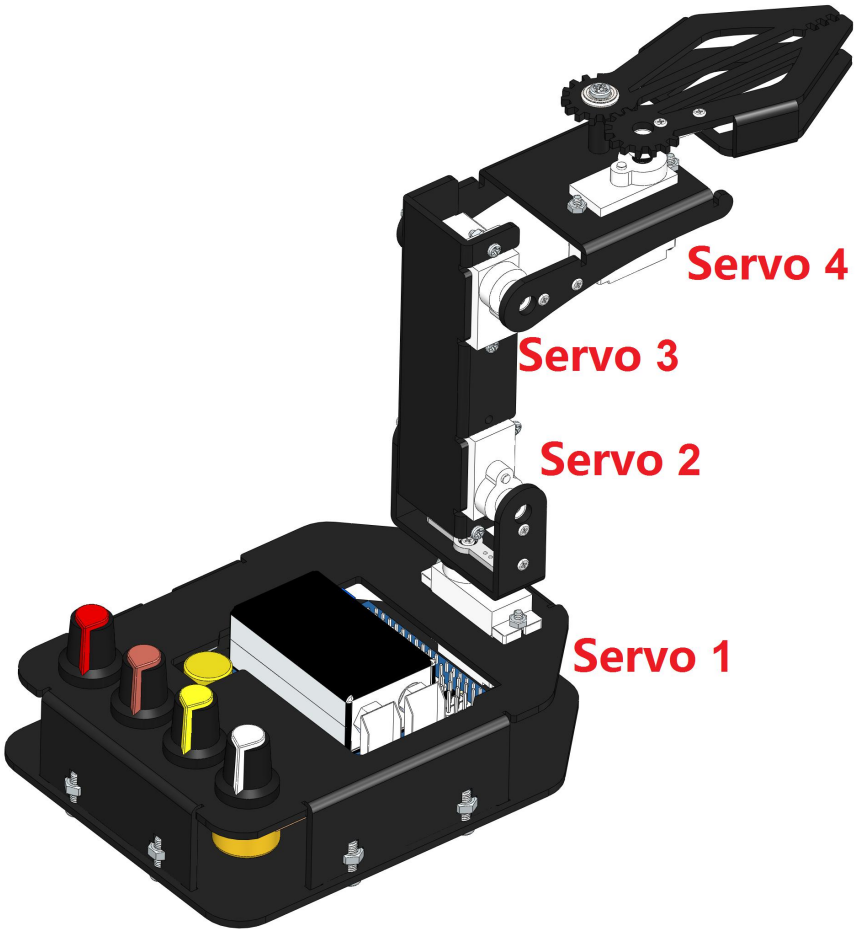


Note: Sometimes you can see the mistakes like 5005 or 5002 that are as results of the software recognition failure of Arduino control board. Now, please click OK. Then there is a quick quiver state in Rollarm robot to return to the setting position. After that, you can continue to do your next step.



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





2) 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.

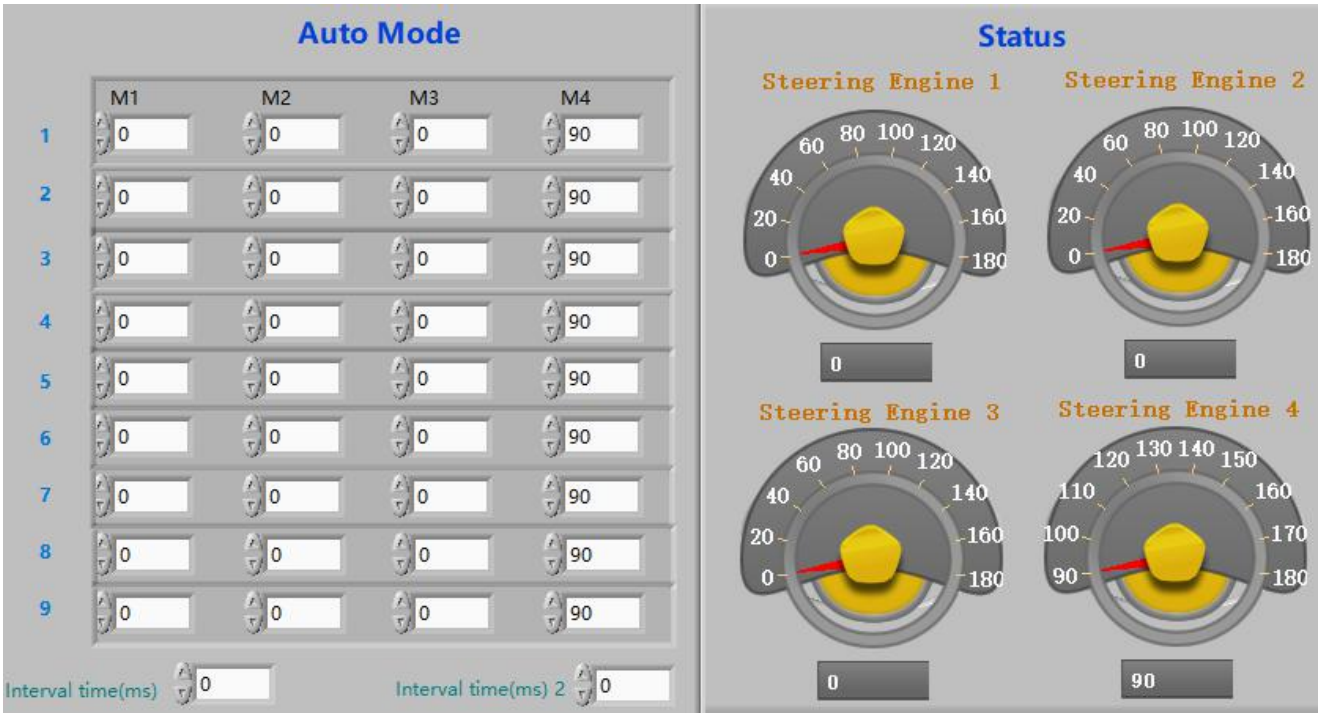
M1-M4: Servo1-Servo4.

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.



FAQ

1) About the Assembly:

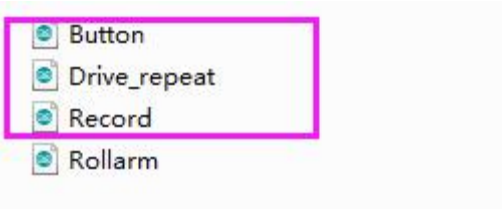
Q1: After assembly and program download, the Rollarm's four axes are in wired position, some maybe out of control. What should I do?

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

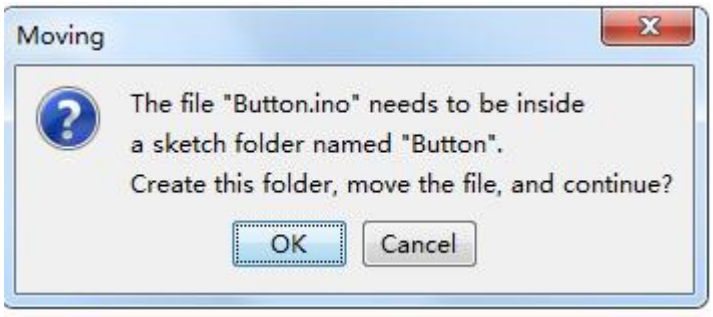
2) About the Arduino code control:

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?

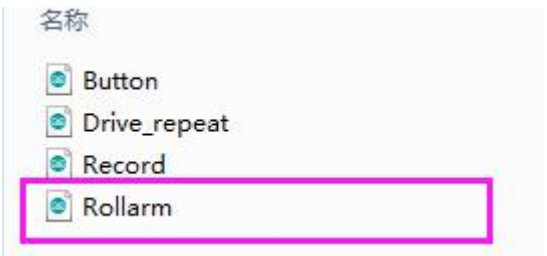
A: DO NOT open these subprograms under Arduino code\Rollarm separately:



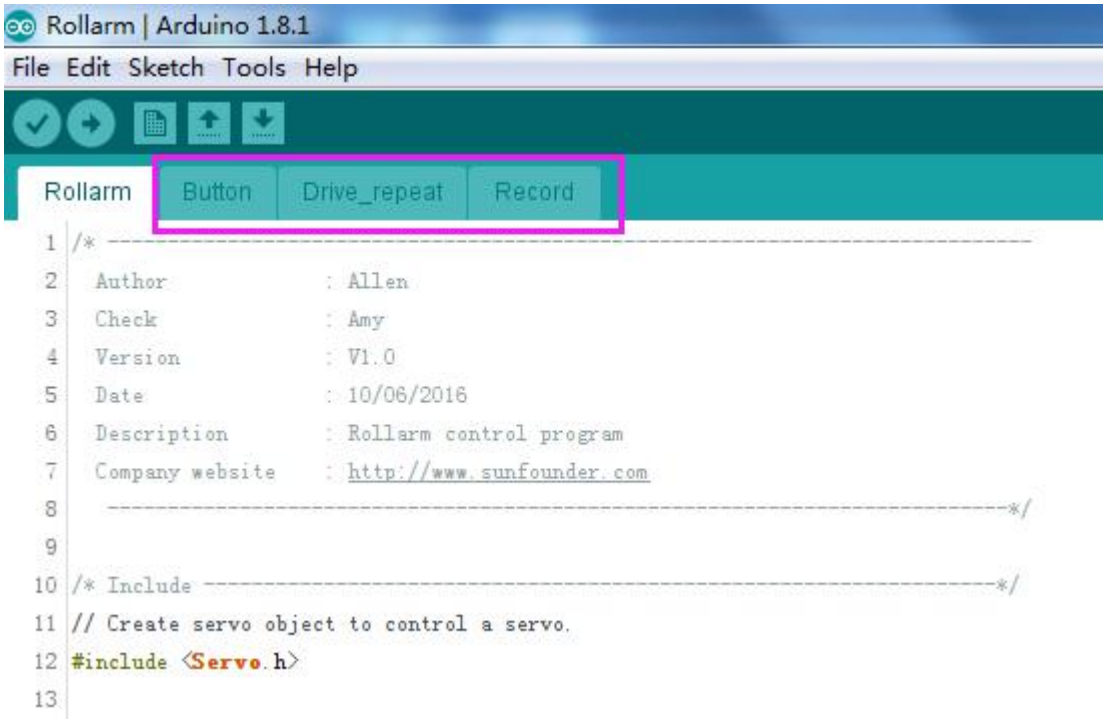
If you open the subprograms separately, a dialog box will pop up like this:



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:



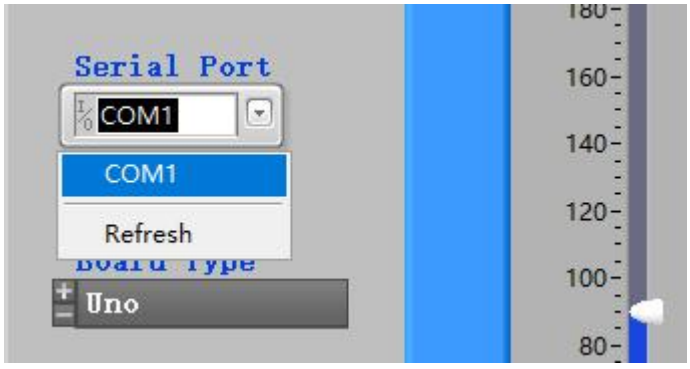
Then you can see the subprograms have been opened too:



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?

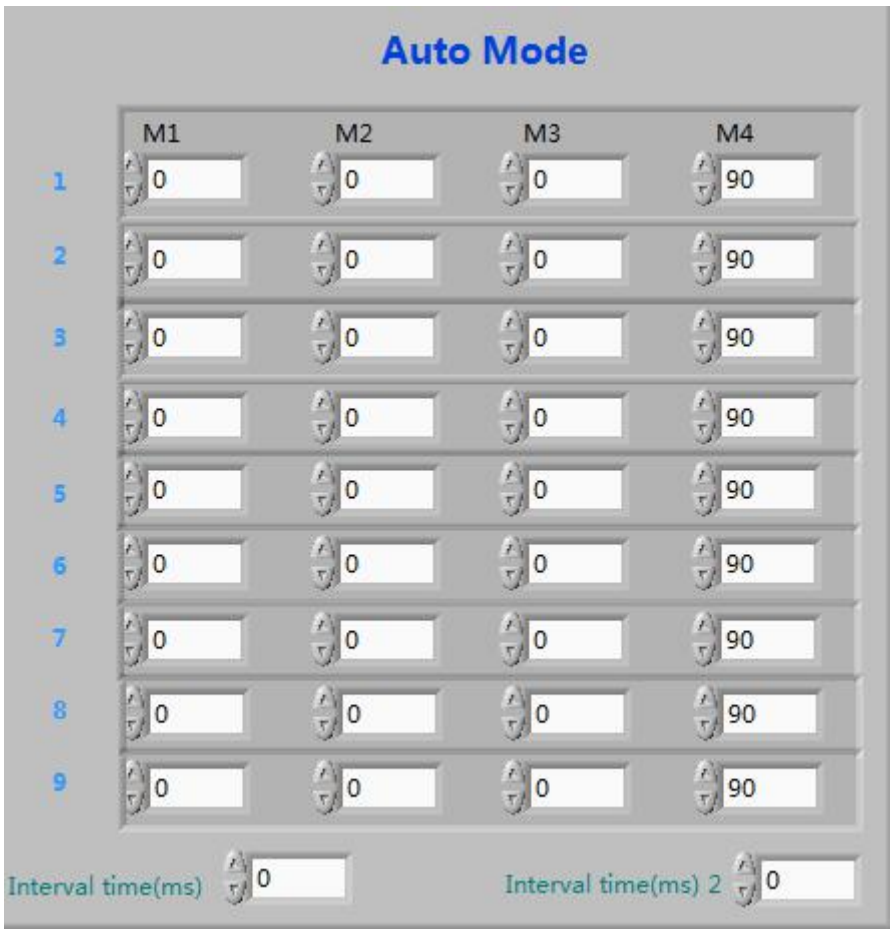
A: There may be something wrong with the Serial Port. For instance, the following condition may appear:



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

Q2: 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°, and the last one is in 90°. You need to fill the correct value of the rotating angle first, and click **Start** to run.



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.

➢ Copyright Notice