

blonwiner@outlook.com

BLOONWINER Arduino ARM Board



blonwiner@outlook.com

About BLONWINER

BLONWINER is a technical service team of open source software and hardware. Dedicated to applying the Internet and the latest industrial technology in open source area, we strive to provide best hardware support and software service for general makers and electronic enthusiasts around the world. We aim to create infinite possibilities with sharing. No matter what field you are in, we can lead you into the electronic world and bring your ideas into reality.

Technical Support: blonwiner@outlook.com

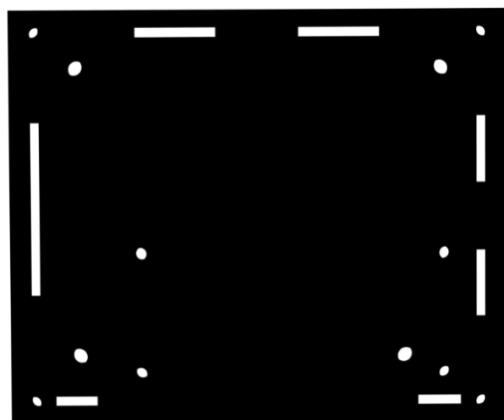
Contents

Components List.....	1
1.1. Acrylic Plates	1
1.2. Machinery Parts	4
1.3. Electronic Parts	5
1.4 Tool	7
1.5 We do not provide 18650 batteries, you need to purchase them yourself.	7
Introduction of Robotic Arm.....	8
Introduction of BLONWINER Arduino ARM Board.....	9
Introduction of power module	11
The power module's switch button can be controlled using the handle of Cross-head Screwdriver	13
Building the Arduino Development Environment.....	14
1. Arduino development language	14
2.Arduino program structure	14
3. The construction of the Arduino development environment.....	14
4. Introduction of Arduino software interface	18
5.Connecting the BLONWINER Arduino Arm Board and the computer.....	23
6.The solution for situation that Arduino IDE cannot be opened	27
7、 Download Processing	29
8. Configuring the "libraries" folder of the Arduino IDE	35
Lesson 0 The servo initialization code of robotic arm assembly	40
Lesson 1 How to Read the Data of the Potentiometer	41
1.1 Components used in this course	41
1.2 Introduction of Potentiometer	41
1.3 Wiring diagram (Circuit diagram)	42
1.4 Reading the value of the potentiometer and converting it into an angle	43
Lesson 2 Controlling the Servo	50
2.1 Components used in this course	50
2.2 The introduction of the Servo	50
2.3Wiring diagram (Circuit diagram)	53
2.4 How to control Servo	53
Lesson 3 Displaying Text on the OLED Screen	58
3.1 Components used in this course	58
3.2 Introduction of OLED Screen	58
3.3 Wiring diagram (Circuit diagram)	59
3.4 How to display text on the OLED screen	60
Lesson 4 Saving Data with EEPROM	64
4.1 Components used in this course	64
4.2 About EEPROM	64
4.3 Wiring diagram (Circuit diagram)	64
4.4 How to use EEPROM to save data	65

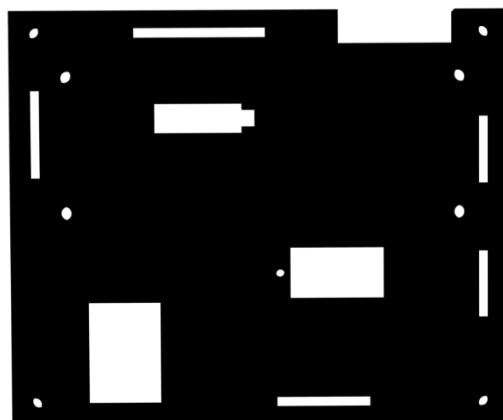
How to Assemble the Robotic Arm	74
Functions of the robotic arm	110
Lesson 5 Potentiometer controls Robotic arm	110
5.1 Components used in this course	110
Lesson 6 Processing controls servo	118
6.1 Components used in this course	119
Lesson 7 IR Remote Controller	129
7.1 Components used in this course	129
Lesson 8 Control Robot by IR Remote	137
8.1 Components used in this course	137

Components List

1.1. Acrylic Plates



A.01/1PCS



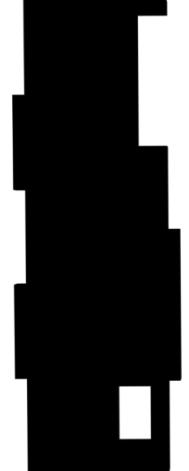
A.02/1PCS



A.03/1PCS



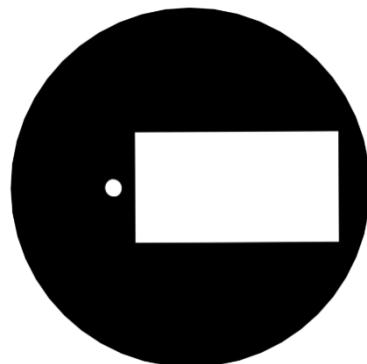
A.04/1PCS



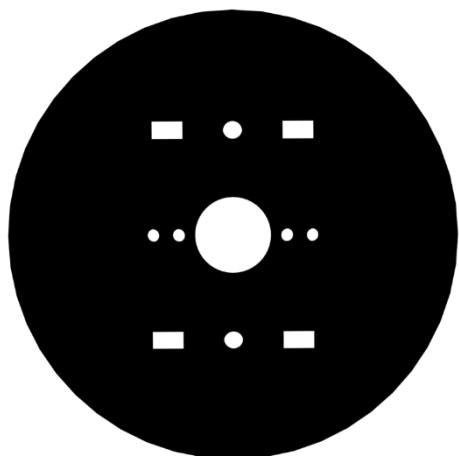
A.05/1PCS



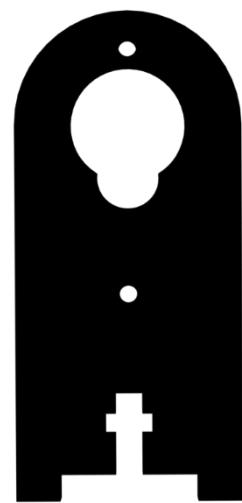
A.06/1PCS



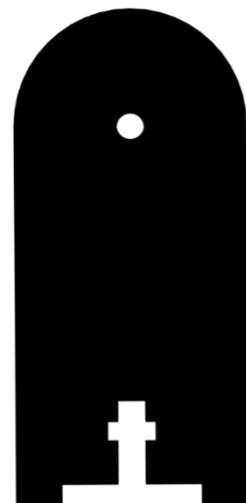
B.01/1PCS



B.02/1PCS



C.01/1PCS



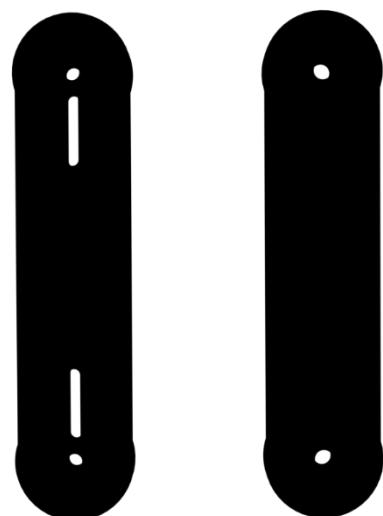
C.02/1PCS



C.03/1PCS



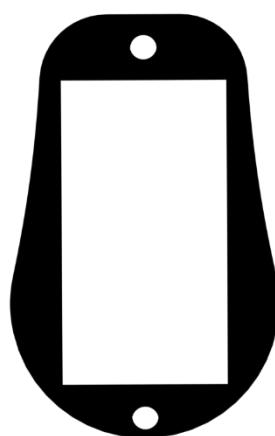
C.04/1PCS



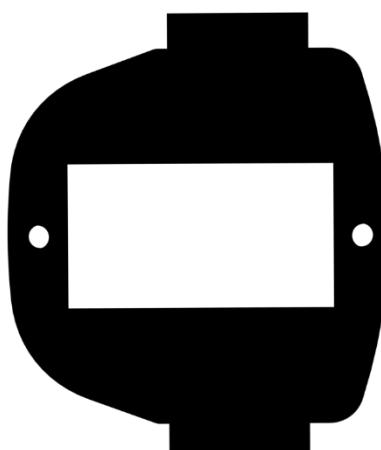
C.05/1PCS



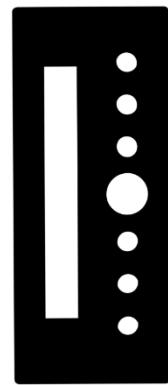
C.06/1PCS



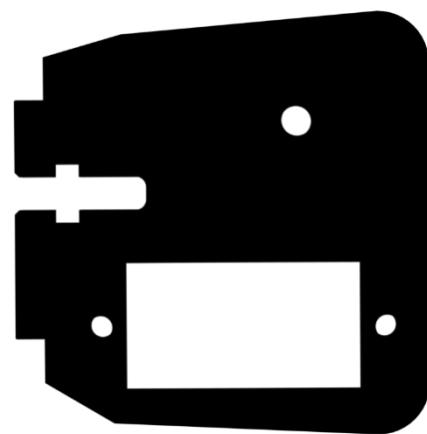
C.07/6PCS



C.08/1PCS



C.09/1PCS



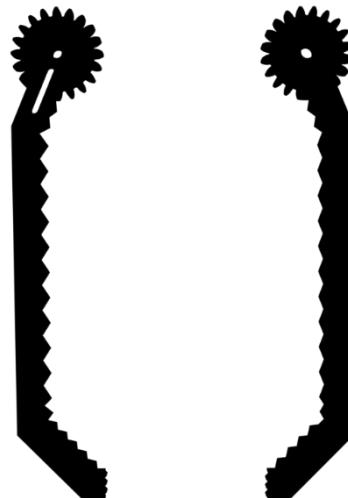
C.10/1PCS



D.01/1PCS



D.02/1PCS



D.03/1PCS



D.04/1PCS

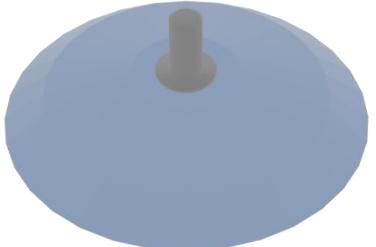
The acrylic plates are fragile, so please be careful when assembling them in case of breaking.

The acrylic plate is covered with a layer of protective film. You need to remove it first.

Some holes in the acrylic may have residues, so you need to clean them before the use.

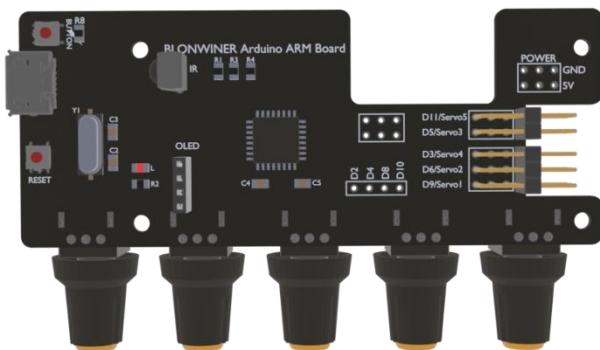
1.2. Machinery Parts

M2.5 Nylon Nut 4PCS 	M2.5*4+6 Copper Standoff 4PCS 	M2.5*8 Screw 7PCS 	M2 Nut 9PCS 	M2*16 Screw 9PCS 
M3 Lock Nut 3PCS 	M3 Nut 3PCS 	M3*8 Nylon Standoff 1PCS 	M3*8 Screw 18PCS 	M3*12 Screw 5PCS 
M3*14 Nylon Standoff 4PCS 	M3*18 Screw 1PCS 	M3*30 Nylon Standoff 5PCS 		

Round Suction Cup M4 Nut(Round Suction Cup Nut) 4PCS Round Suction Cup Screws 4PCS  	51108 Bearing 
--	---

1.3. Electronic Parts

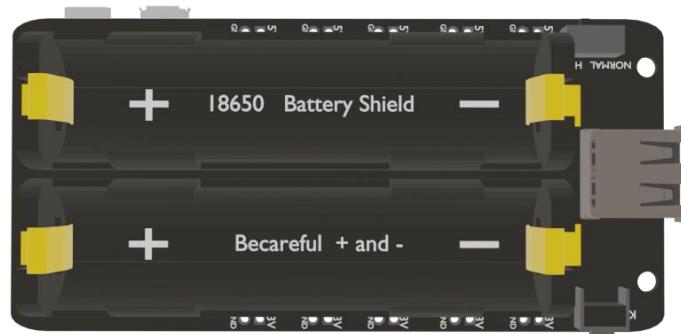
BLOONWINER Arduino ARM Board 1PCS



OLED module 1PCS

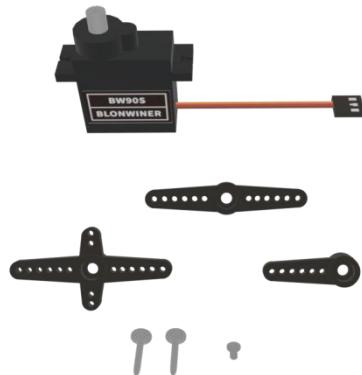


Power Module 1PCS



blonwiner@outlook.com

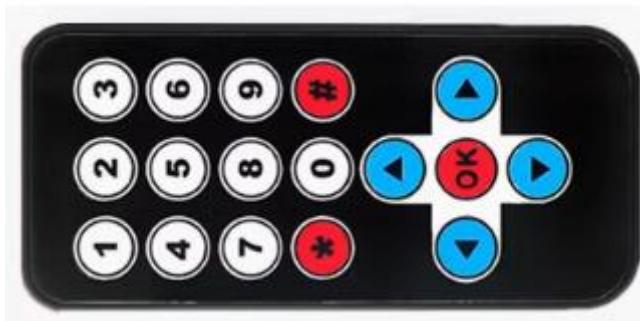
Servo 5PCS



Servo Extension Cable 1PCS



Remote Controller 1pcs



USB Cable 1PCS



1.4 Tool

Cross Socket Wrench 1PCS



Cross-head Screwdriver 1PCS



Winding Pipe 1PCS



1.5 We do not provide 18650 batteries, you need to purchase them yourself.

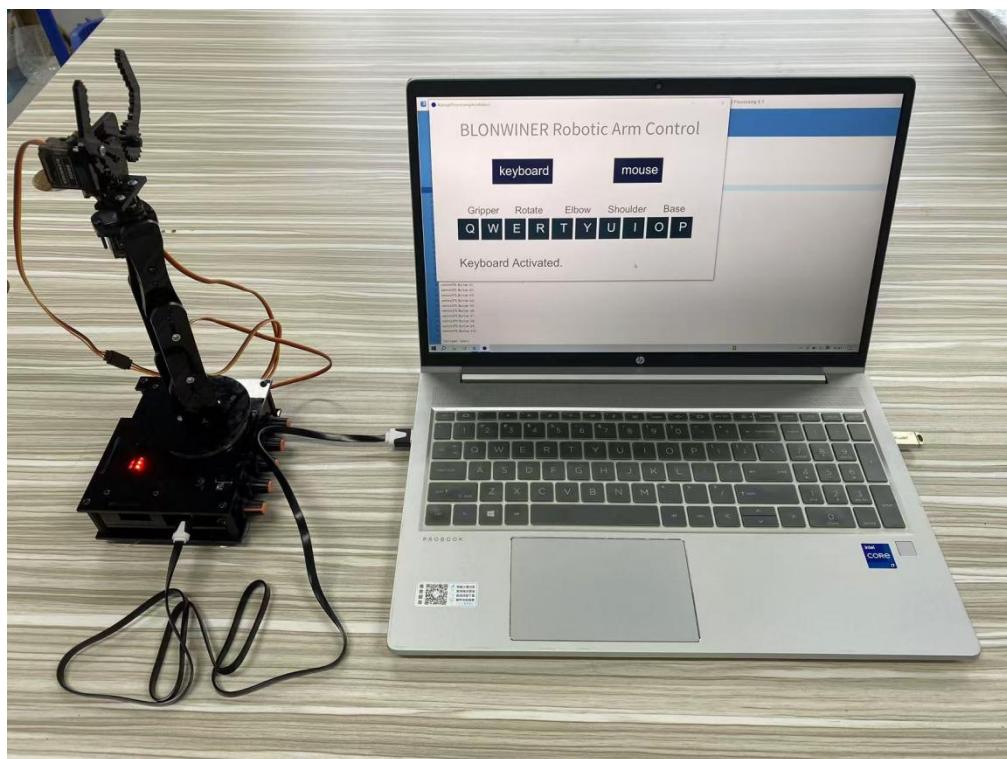
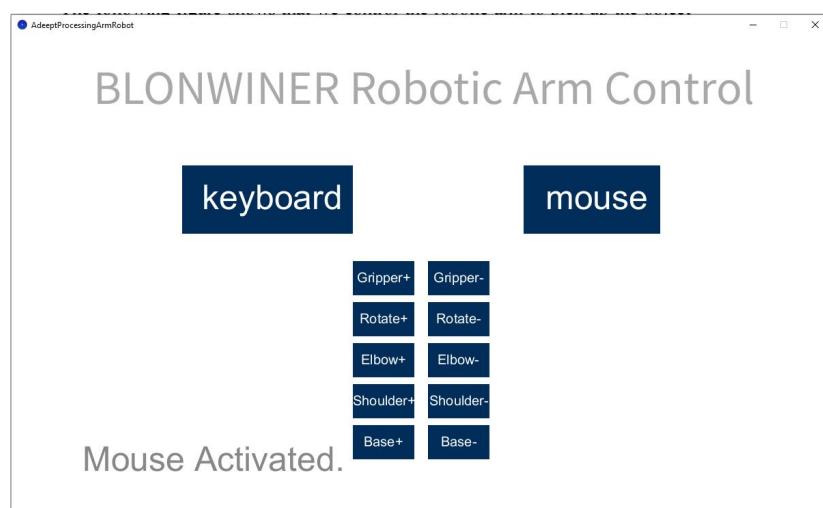
18650 Battery 2PCS



Introduction of Robotic Arm

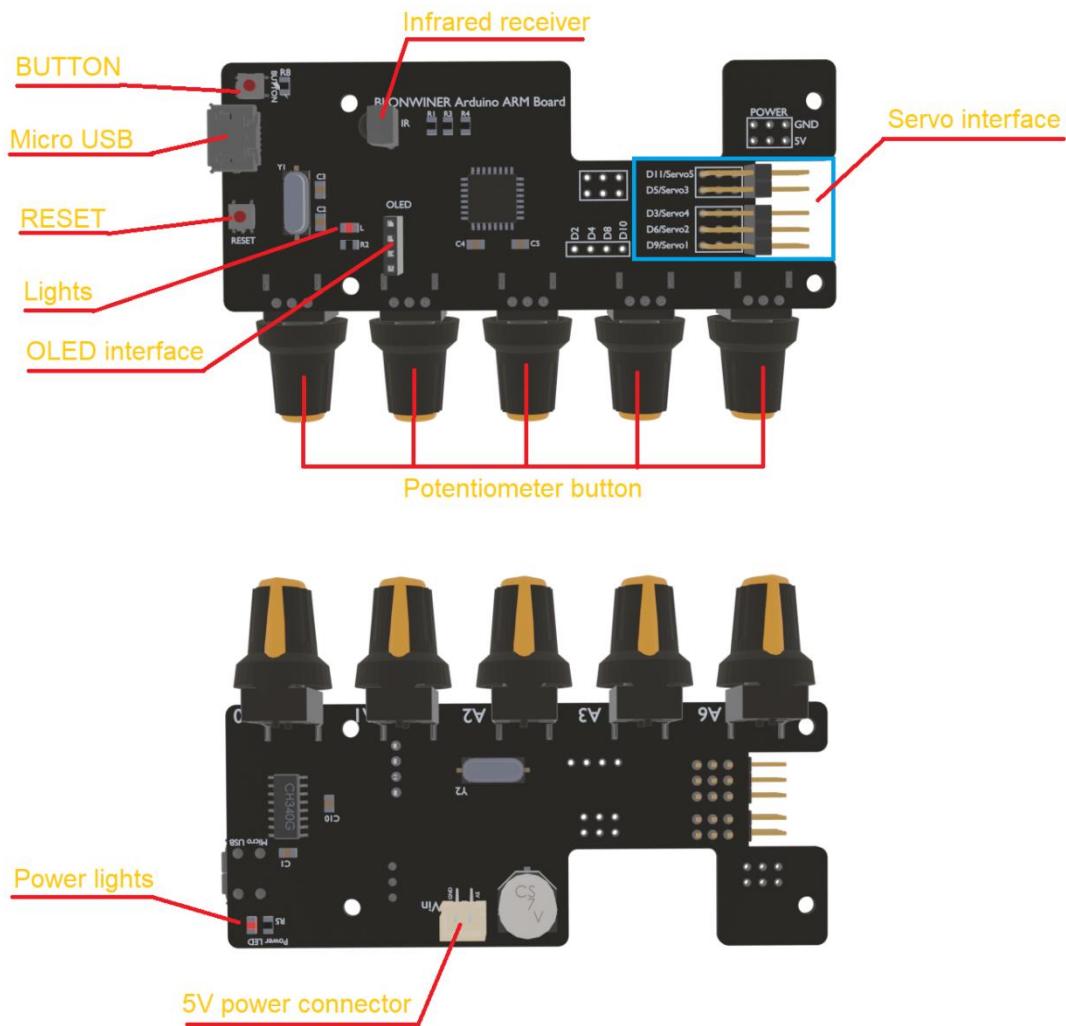
Nowadays, under the progress of science and technology, the biggest difference between a robotic arm and a human arm lies inflexibility and strength. That is, the Biggest advantage of the robotic arm is that normally it can repeat the same motion without feeling tired. Today BLONWINER recommends a robotic learning kit to learn how to assemble a robotic arm and learn how to write the code to control the robotic arm to perform the specific motions. We provide a completed using method for learning Arduino and Processing write PC software and send motion commands to the robotic arm with Processing; write the motion of the servo of the robotic arm with Arduino.

The following figure shows that we control the robotic arm to pick up the object through the keyboard with serial communication.



Introduction of BLONWINER Arduino ARM Board

The BLONWINER Arm Drive Board development board is the main component of the robotic arm. Similar to the Arduino UNO development board, it is also an easy-to-use open source electronic prototyping platform, including the hardware part and the software part(Arduino IDE). The BLONWINER Arduino ARM Board is mainly composed of a micro-controller(MCU), a universal input/output interface, etc. You can understand it as a micro-computer motherboard. We will introduce the BLONWINER Arm Board development board in detail.



Power LED:

Power LED is used to indicate the power status of the system. The LED is on, indicating that the system is powered on and ready to run; the LED is off, indicating that the system is not powered on.

Servo interface:

It is the pin interface of Servo.

5V power connector:

It is the pin interface for external power supply. Use 5V DC external power supply to power the BLONWINER Arduino ARM board.

RESET:

Restarting the BLONWINER Arduino ARM board.

Micro USB:

It is used to connect the Micro USB interface of the computer to realize the serial communication, uploading program and serial monitoring between the BLONWINER Arduino ARM board and the computer.

Potentiometer button:

Potentiometer button has five buttons: A0, A1, A2, A3, and A6. By rotating these buttons, you can control the movement of the robotic arm.

In the following courses, we will combine the application of various components to further learn the practical application of the BLONWINER Arduino ARM board.

BUTTON:

BUTTON connects to the D4 data port of Arduino UNO.

Light:

The signal light is connected to the Arduino UNO D13 data port

OLED interface:

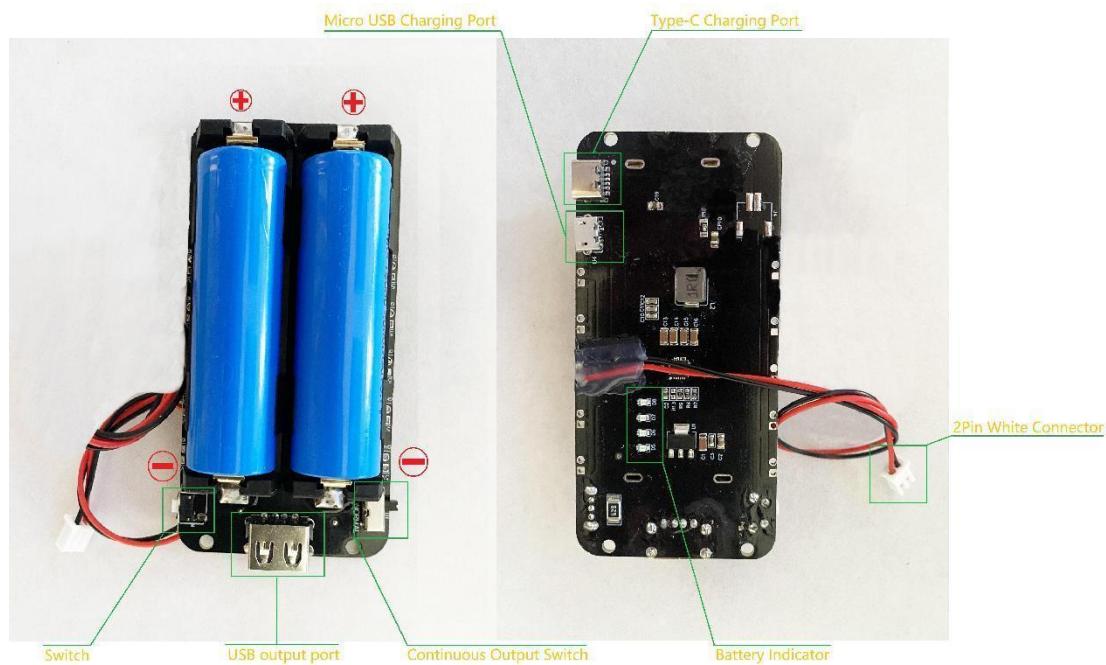
This interface is connected to the interface of the OLED module.

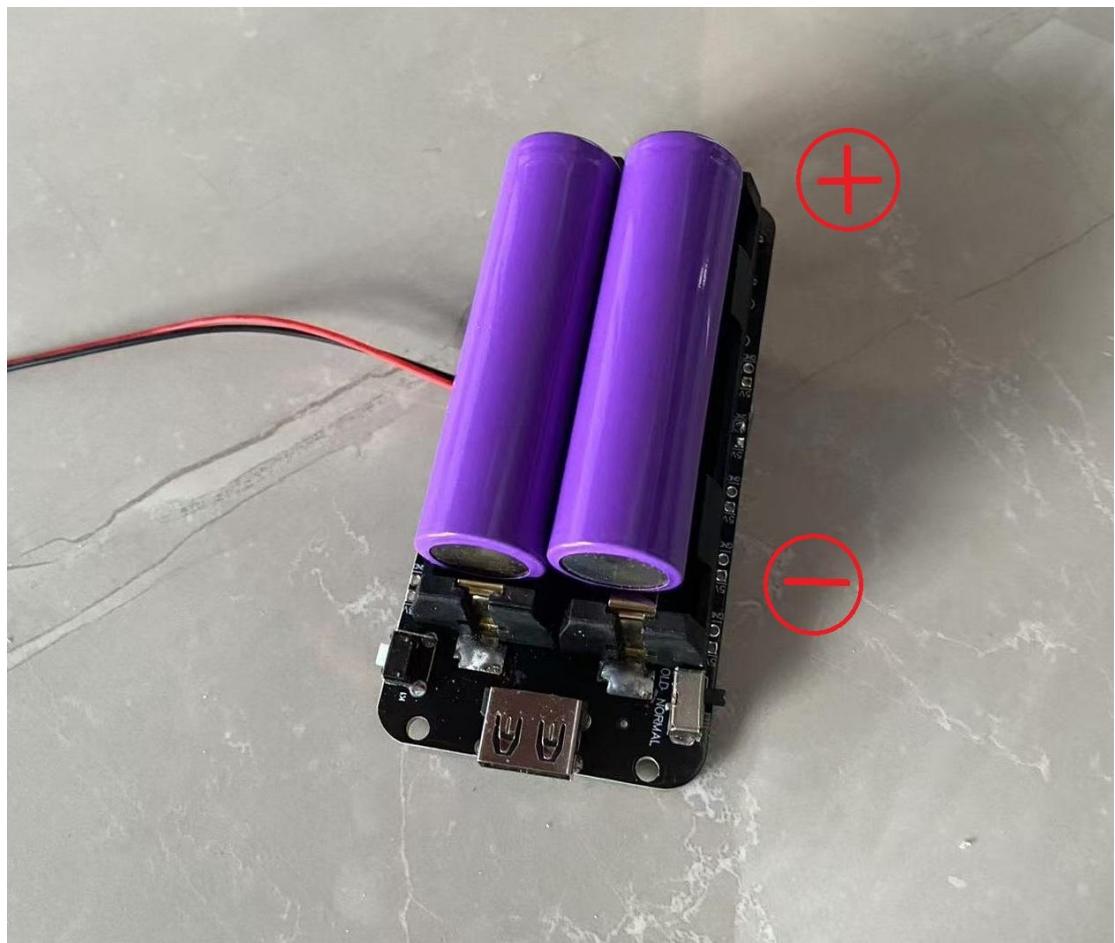
Infrared receiver:

This sensor is used to receive signals from the remote control.

Introduction of power module

When you get the robot product, you will see a board with its name: Power Module(18650 Battery Shield), which is an important part of the robot. There are many interfaces on the Power Module. Note that the positive and negative poles of the battery cannot be installed incorrectly, otherwise the power module will be burned out.(Please refer to the assembly video)





Switch Power module switch:

Short press once: Turn on the power module

Short press twice/(Press and hold for two seconds): Turn off the power module

USB Output Port

USB output port: max 2A current

Continuous Output Switch:

● The switch is turned to the HOLD position to control the 2Pin white connector and USB interface to continuously output power, and the power module will not automatically turn off the power supply.

● The switch is turned to the NORMAL position, and the 2Pin white connector and USB interface are controlled to automatically output power, and the power module will automatically turn off the power.

Micro USB Charging Port

You can charge two 18650 batteries through this port.

Type-C Charging Port

You can charge two 18650 batteries through this port.

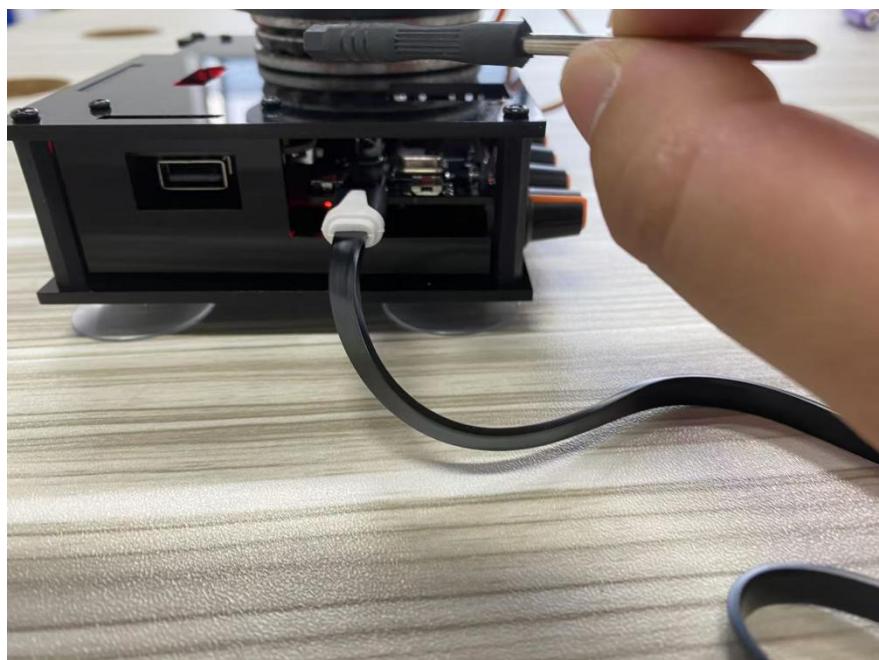
2Pin White Connector

The output voltage is 5V, and the maximum current is 2A. Connect the black connector of 5Vin(2in1 RPI Board)

Battery Indicator

Displays the battery level and shows that the battery is charging.

The power module's switch button can be controlled using the handle of Cross-head Screwdriver .



Building the Arduino Development Environment

1. Arduino development language

Arduino uses C/C++ to write programs, so before learning Arduino, you need to master the C/C++ language. Although C++ is compatible with the C language, these are two different languages. C is a process-oriented programming language, and C++ is an object-oriented programming language. The early Arduino core library was written in C language. Later, object-oriented ideas were introduced. At present, the latest Arduino core library is written in C and C++.

Generally speaking, the Arduino language refers to a collection of various Application Programming Interfaces (APIs) provided by the Arduino core library files. These APIs are formed by secondary packaging of the lower-level microcontroller support library. For example, the core library of Arduino using AVR microcontroller is the secondary packaging of AVR-Libc (GCC-based AVR support library).

In the traditional development method, multiple registers need to be configured to achieve the corresponding functions. In Arduino, the complicated registers are encapsulated into simple APIs, which can be intuitively controlled, enhancing the readability of the program and improving the development efficiency.

2.Arduino program structure

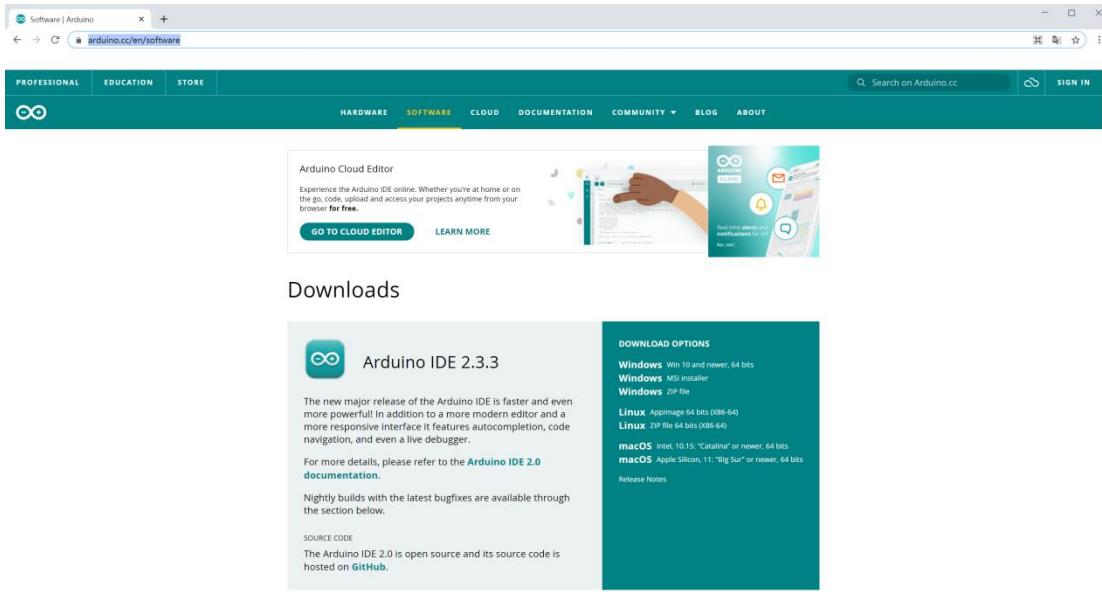
The Arduino program structure is different from the traditional C/C++ program structure-there is no main() function in the Arduino program. In fact, it is not that there is no main() function in the Arduino program, but that the definition of the main() function is hidden in the core library file of the Arduino. In the development of Arduino, the main function is not directly operated, but the two functions of setup() and loop() are used instead.

3. The construction of the Arduino development environment

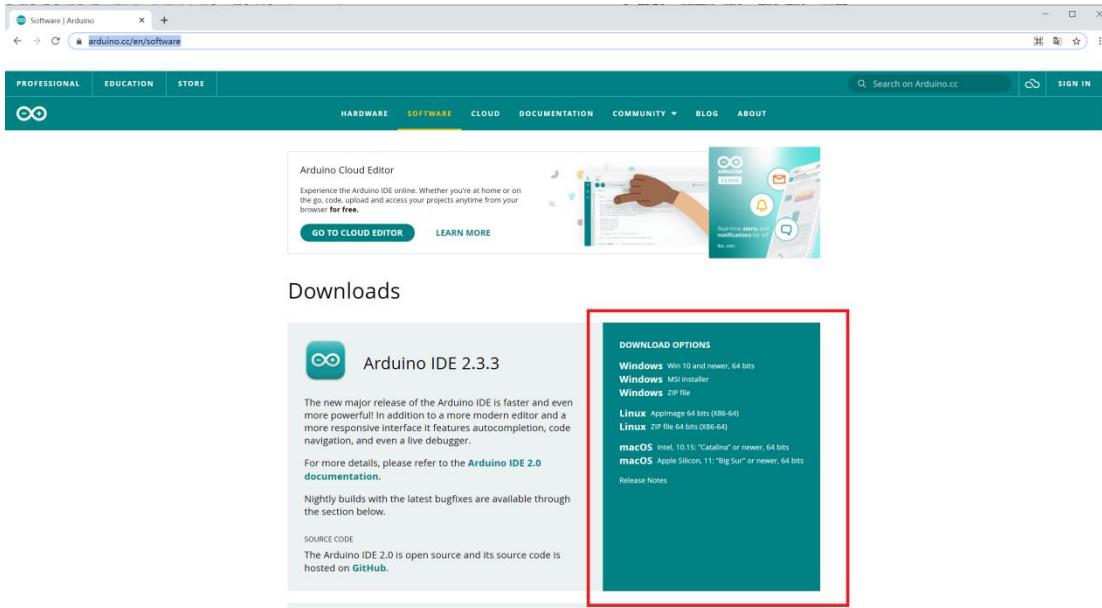
The IDE of the Arduino development environment can be downloaded from the official website. The download address of the Arduino IDE is:

<https://www.arduino.cc/en/software>

After successfully opening the interface as shown below.



2. After jumping to the following interface, slide the mouse to the middle to find the part marked in the red circle. You can find that the official website provides us with installation files for Windows, Mac OS X, and Linux systems.



3. We click the installation package of Windows ZIP file for non admin install. After the interface jumps, we select JUST DOWNLOAD. And then start the download. The download status will be displayed in the lower left of Google Chrome. Then we wait for the download to complete.

Downloads

Arduino IDE 2.3.3



The new major release of the Arduino IDE is faster and even more powerful! In addition to a more modern editor and a more responsive interface it features autocompletion, code navigation, and even a live debugger.

For more details, please refer to the [Arduino IDE 2.0 documentation](#).

Nightly builds with the latest bugfixes are available through the section below.

SOURCE CODE
The Arduino IDE 2.0 is open source and its source code is hosted on [GitHub](#).

DOWNLOAD OPTIONS

Windows Win 10 and newer, 64 bits
Windows MSI installer
Windows ZIP file (selected)

Linux AppImage 64 bits (X86-64)
Linux ZIP file 64 bits (X86-64)

macOS Intel, 10.15: "Catalina" or newer, 64 bits
macOS Apple Silicon, 11: "Big Sur" or newer, 64 bits

[Release Notes](#)

Download Arduino IDE & support its progress

Since the 1.x release in March 2015, the Arduino IDE has been downloaded **87,381,265** times — impressive! Help its development with a donation.

\$3 \$5 \$10 \$25 \$50 Other

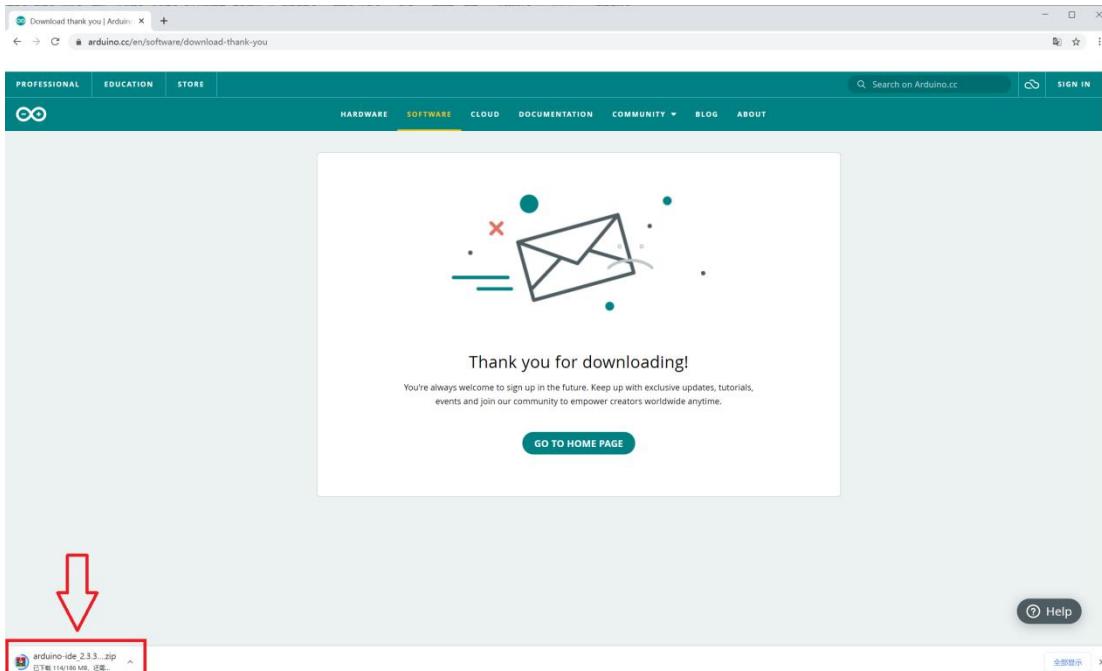
CONTRIBUTE AND DOWNLOAD

or

JUST DOWNLOAD (selected)



Learn more about [donating to Arduino](#).



4. After the download is complete, open the folder. There are downloaded compressed installation files:

arduino-ide_2.3.3_Windows_64bit.zip

Win 10 Ent x64 (C:) > DownLoad >			
名称	修改日期	类型	大小
arduino-ide_2.3.3_Windows_64bit.zip	2024/10/8 10:52	WinRAR ZIP 压缩文件	190,546 KB

5. Double-click to open the file and unzip it.

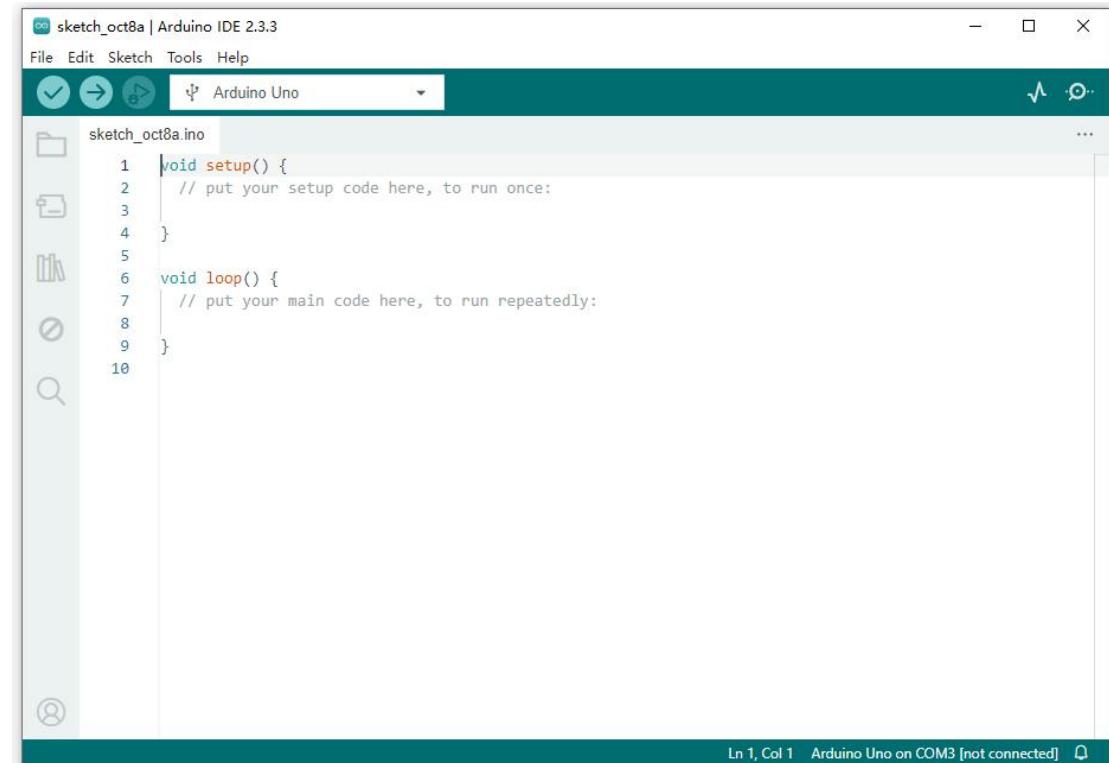
Win 10 Ent x64 (C:) > DownLoad >			
名称	修改日期	类型	大小
locales	2024/10/8 11:08	文件夹	
resources	2024/10/8 11:08	文件夹	
Arduino IDE.exe	2024/9/25 11:43	应用程序	168,604 KB
<input checked="" type="checkbox"/> arduino-ide_2.3.3_Windows_64bit.zip	2024/10/8 10:52	WinRAR ZIP 压缩文件	190,546 KB
chrome_100_percent.pak	2024/9/25 11:41	PAK 文件	133 KB
chrome_200_percent.pak	2024/9/25 11:41	PAK 文件	191 KB
d3dcompiler_47.dll	2024/9/25 11:41	应用程序扩展	4,802 KB
ffmpeg.dll	2024/9/25 11:41	应用程序扩展	2,820 KB
icudtl.dat	2024/9/25 11:41	DAT 文件	10,467 KB
libEGL.dll	2024/9/25 11:41	应用程序扩展	478 KB
libGLESv2.dll	2024/9/25 11:41	应用程序扩展	7,436 KB
LICENSE	2024/9/25 11:41	文本文档	2 KB
LICENSES	2024/9/25 11:41	Microsoft Edge HT...	9,011 KB
resources	2024/9/25 11:41	PAK 文件	5,356 KB

A progress dialog box is overlaid on the file list, showing the extraction of 'arduino-ide_2.3.3_Windows_64bit.zip' to 'C:\DownLoad\arduino-ide_2.3.3_Windows_64bit'. The progress is at 96%. The dialog includes buttons for '后台(B)', '暂停(P)', '取消', '模式(M)...', and '帮助'.

6. Double-click arduino.exe to open the software.

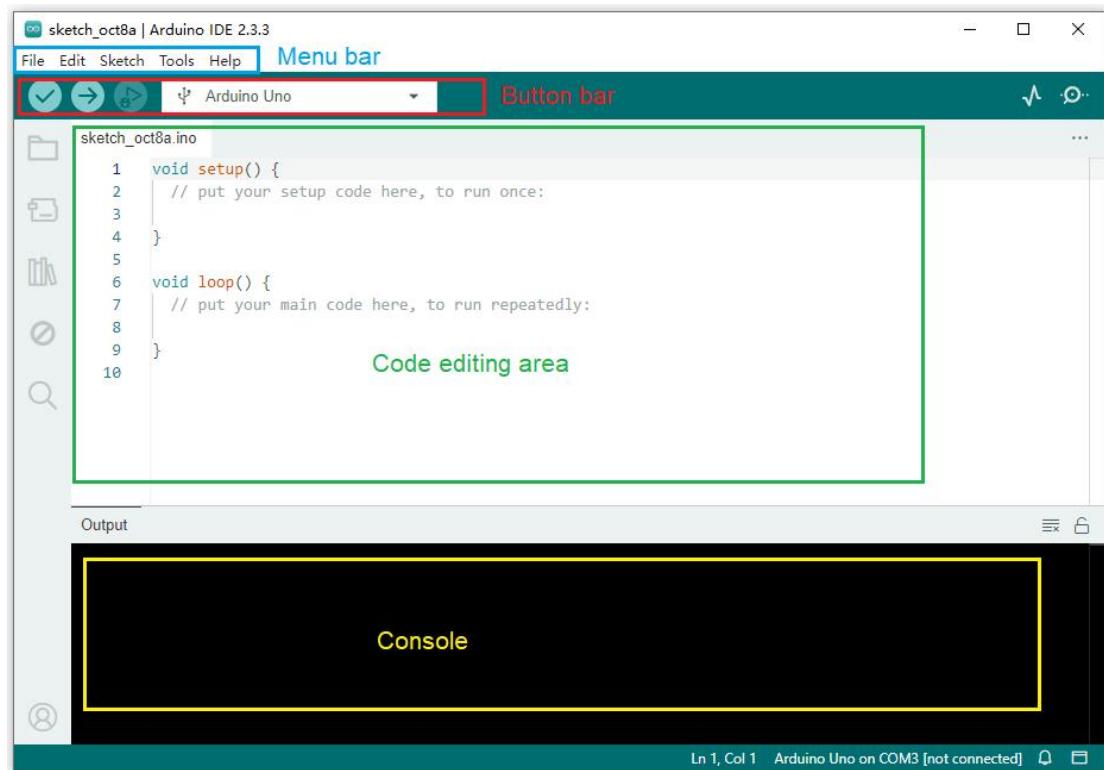
Win 10 Ent x64 (C:) > DownLoad			
名称	修改日期	类型	大小
locales	2024/9/25 11:41	文件夹	
resources	2024/9/25 11:41	文件夹	
Arduino IDE.exe	2024/9/25 11:43	应用程序	168,604 KB
arduino-ide_2.3.3_Windows_64bit.zip	2024/10/8 10:52	WinRAR ZIP 压缩文件	190,546 KB
chrome_100_percent.pak	2024/9/25 11:41	PAK 文件	133 KB
chrome_200_percent.pak	2024/9/25 11:41	PAK 文件	191 KB
d3dcompiler_47.dll	2024/9/25 11:41	应用程序扩展	4,802 KB
ffmpeg.dll	2024/9/25 11:41	应用程序扩展	2,820 KB
icudtl.dat	2024/9/25 11:41	DAT 文件	10,467 KB
libEGL.dll	2024/9/25 11:41	应用程序扩展	478 KB
libGLESv2.dll	2024/9/25 11:41	应用程序扩展	7,436 KB
LICENSE.electron.txt	2024/9/25 11:41	文本文档	2 KB
LICENSES.chromium.html	2024/9/25 11:41	Microsoft Edge HT...	9,011 KB
resources.pak	2024/9/25 11:41	PAK 文件	5,356 KB
snapshot_blob.bin	2024/9/25 11:41	BIN 文件	262 KB
v8_context_snapshot.bin	2024/9/25 11:41	BIN 文件	612 KB
vk_swiftshader.dll	2024/9/25 11:41	应用程序扩展	5,059 KB
vk_swiftshader_icd.json	2024/9/25 11:41	JSON 文件	1 KB
vulkan-1.dll	2024/9/25 11:41	应用程序扩展	932 KB

7. The interface will show as follows after the Arduino software is opened, indicating that our software has been downloaded and installed successfully.



4. Introduction of Arduino software interface

The following figure is the interface introduction of Arduino software



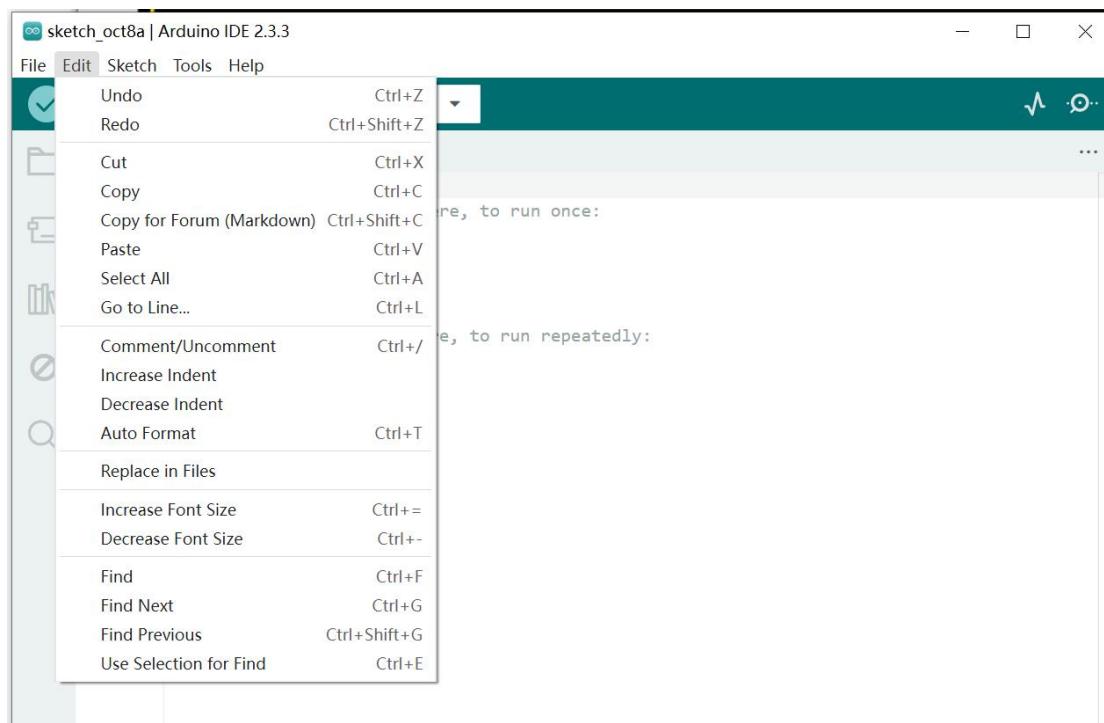
(1)Menu bar

Menu bar contains File, Edit, Sketch, Tools and Help.

(1) "File" can operate new file, open file, save file, close file, save, etc. For the Examples, you can check the official sample program.



(2) "Edit" has the functions for the program code of editing, copying and pasting, commenting, indenting, searching, etc.

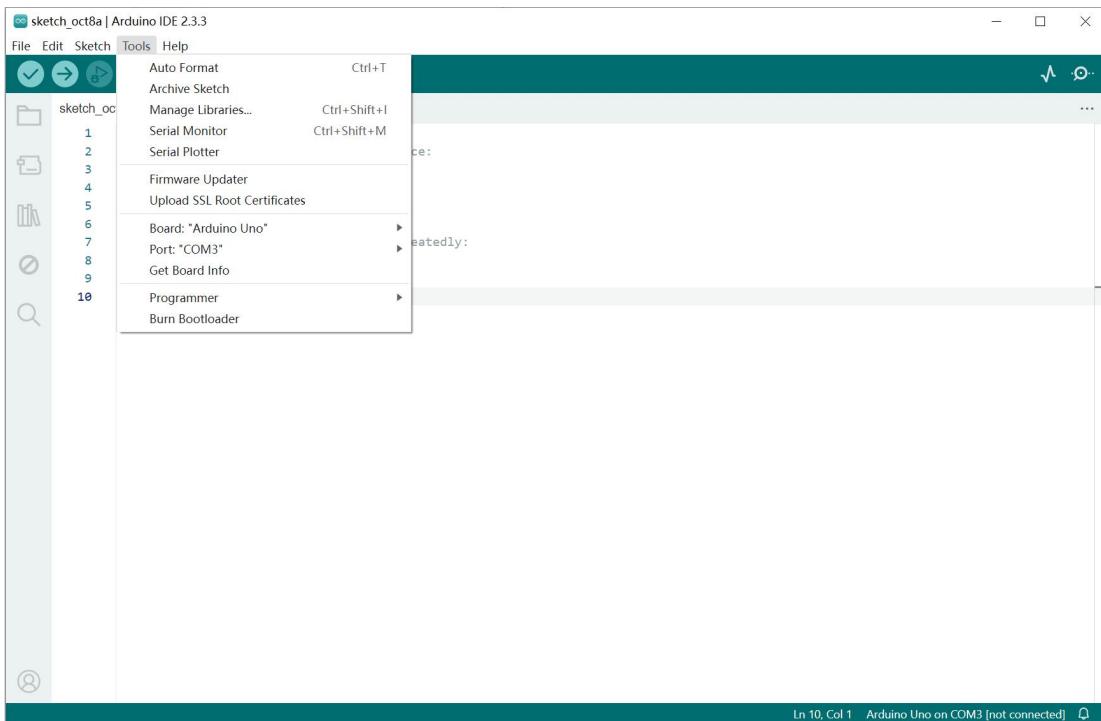


(3) Sketch can perform Verify/Compile, Upload and other operations on the written project.

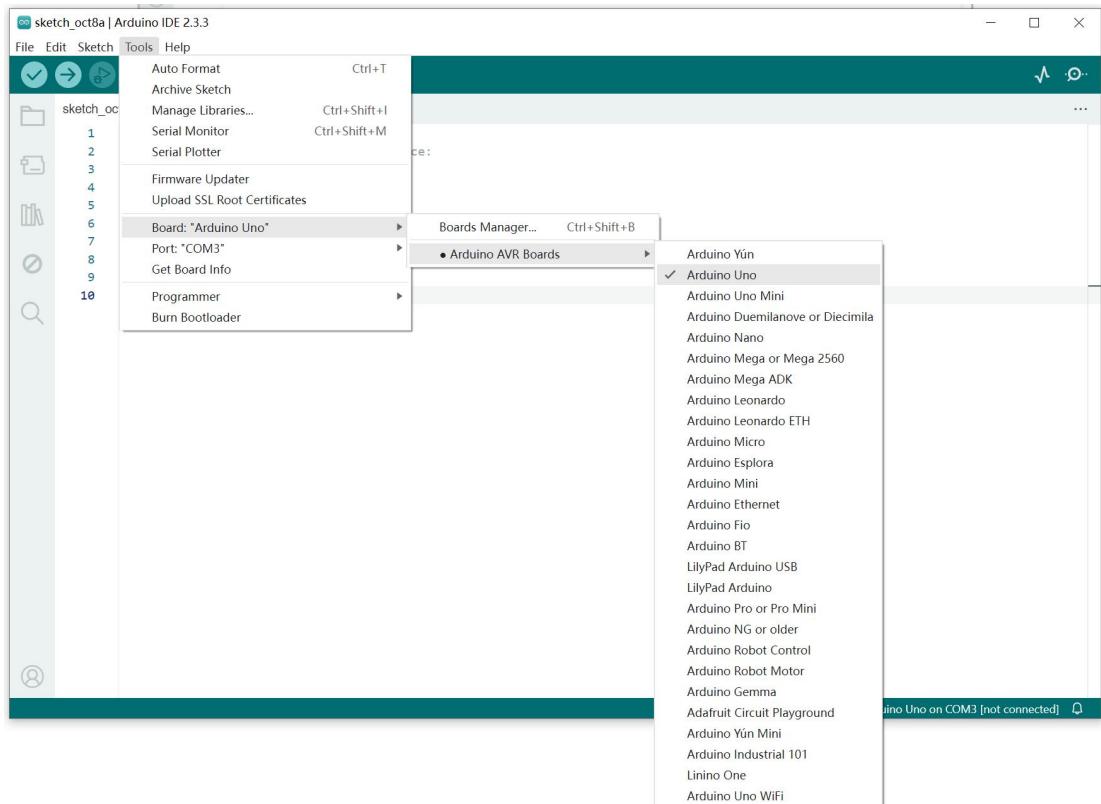


The Include Library can load the library. After selecting the library file in the list, the relevant header files are automatically added in the code editing area.

(4) Board and Port are often used in "Tools".

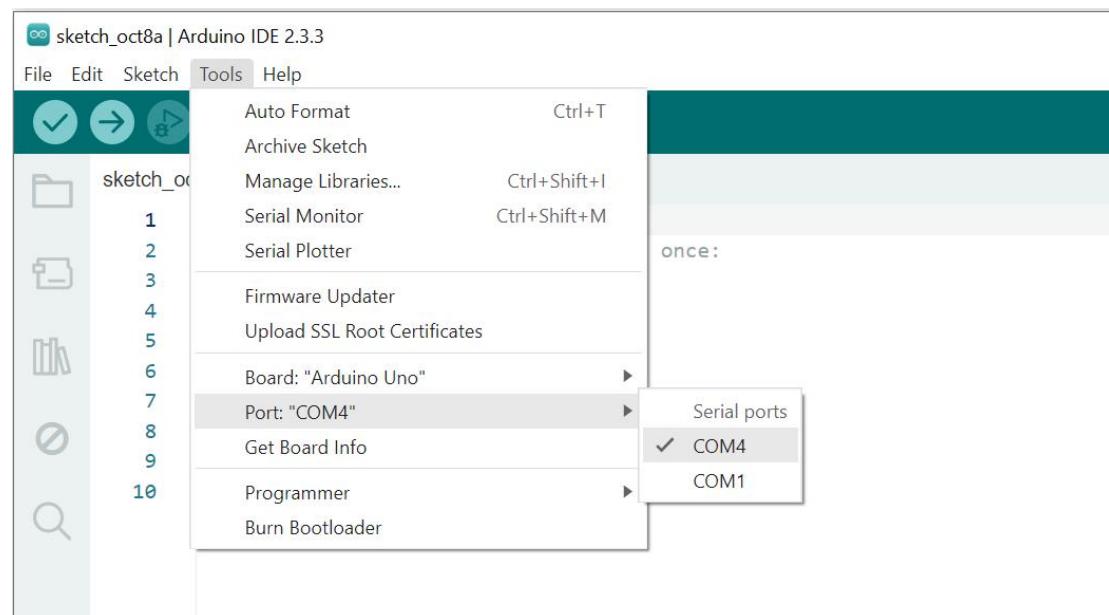
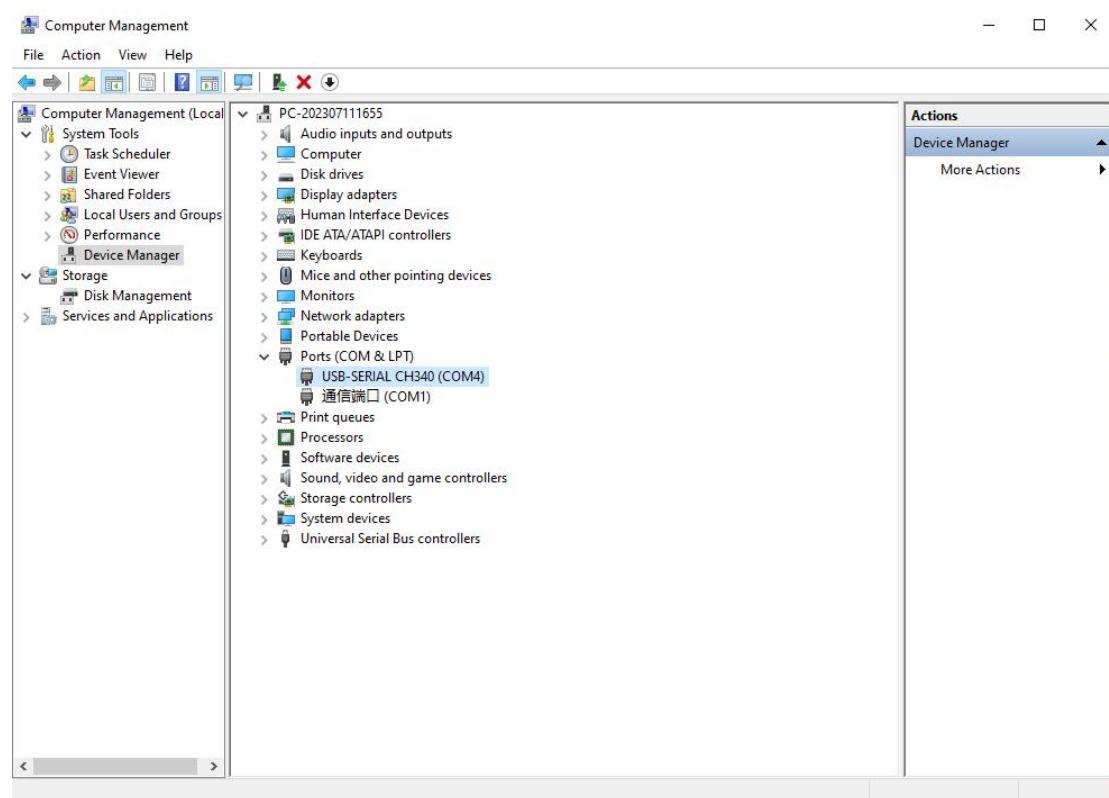


Board can choose different development boards. Our course uses Arduino Uno development board, so we need to choose Arduino Uno. The list contains many Arduino development board models. We choose the corresponding ones according to the model.



Port can set the port used by Arduino IDE to download the program, that is, the port number of the development board connected to the computer. The port display of each

computer is different. When we use the Arduino Uno to connect to the computer, it displays the COM3 port number.(It is determined that the COM3 port is connected to UNO through the task manager.)



(2) Button bar

Button bar includes functions of Verifying,Uploading,Building New,Opening and Saving.

(1) Verify :

Checking and compilation. This button is used to check the correctness of your "syntax" or code. If your code has any syntax errors or undefined variables, an error message will appear at the bottom of the IDE screen. At the same time, the line of error code will be marked with a red background color for easy modification. But if it is correct, you will see the message that the compilation is complete.

(2) Upload  :

Download the program code to the Arduino development board. It is better to click Verify first, and then click Upload.

(3)Code editing area

The code editing area is where to write program code and code comments.

(4)Console

The debug window will output information showing various compilation and debugging results. For example, if your code is written incorrectly, you will be prompted about what went wrong.

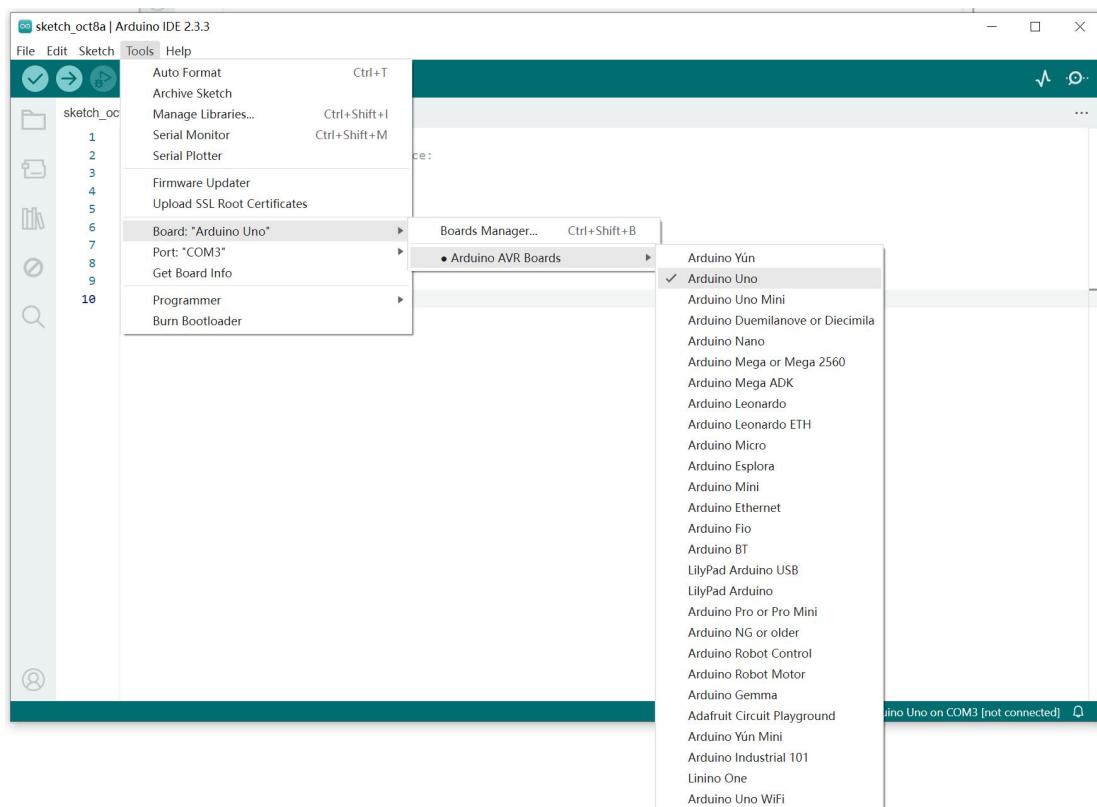
5.Connecting the BLONWINER Arduino Arm Board and the computer

(1)Connecting the BLONWINER Arduino Arm Board and the computer

You need to use USB Cable to connect the BLONWINER Arduino Arm Board to the computer. As shown below:

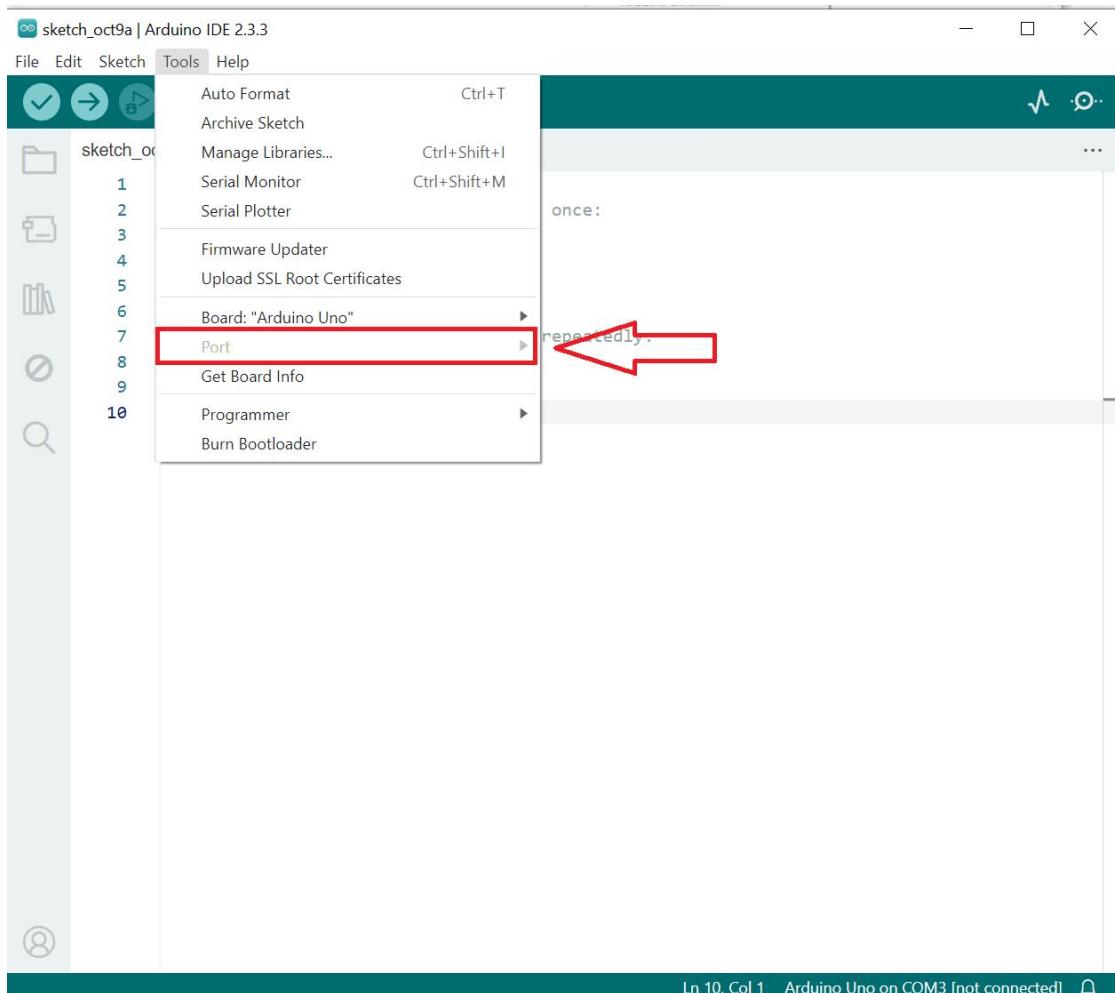
(2) Select the Arduino Uno development board in Tools

Open Arduino IDE under Tools—>Board. Select Arduino UNO in the list.

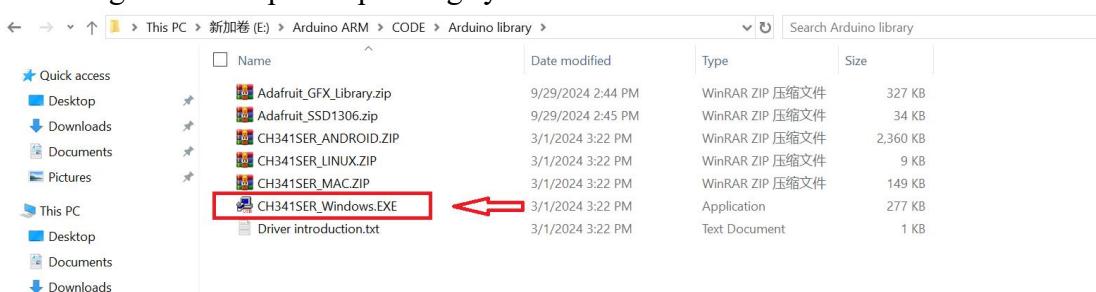


(3)Install CH341SER driver

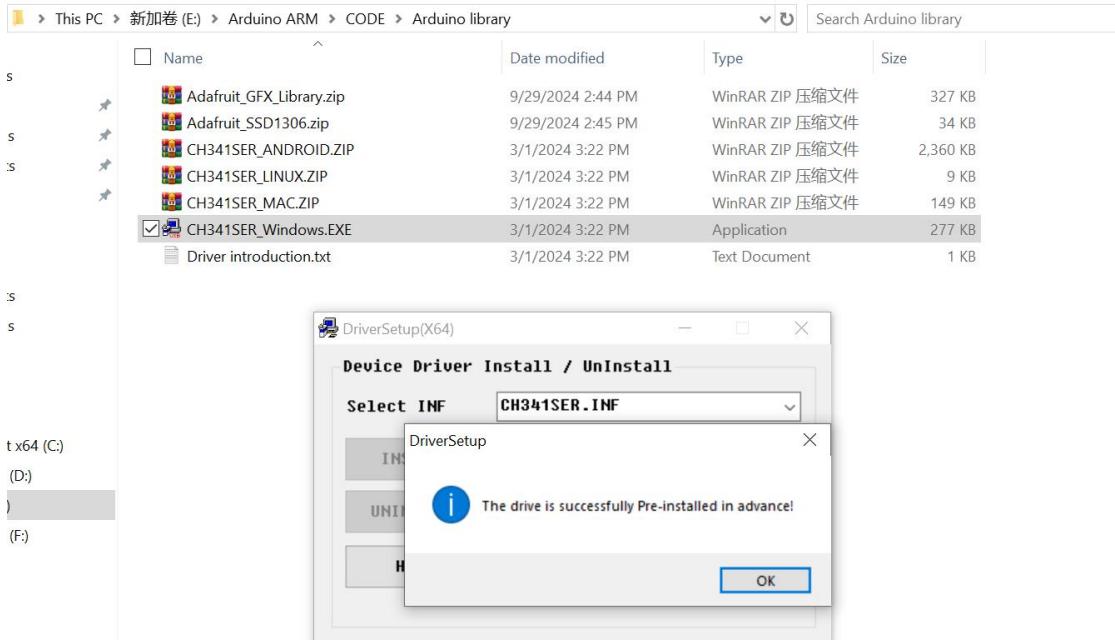
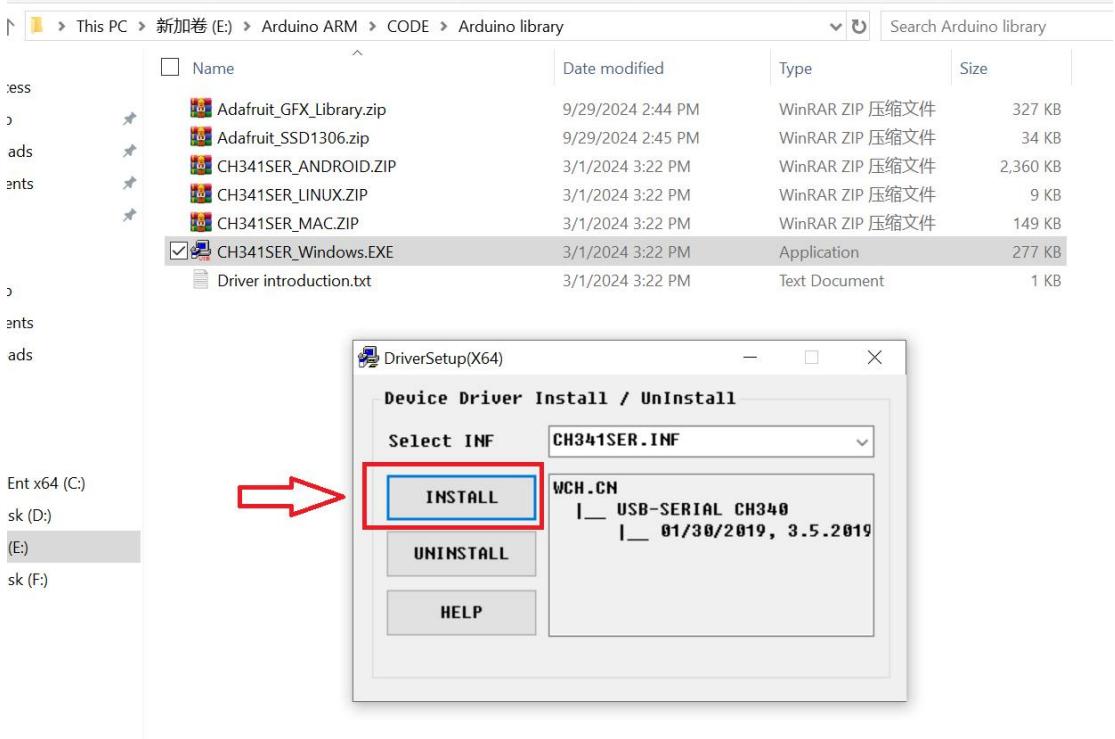
1. Open the Arduino IDE, in the Port on the Tools toolbar, you will see that the serial port cannot be accessed, which means that you have not installed the serial port driver.



2. You need to find the user folder provided by BLONWINER: BLONWINERRoboticArmforArduino, find the Arduino library folder, and open the BLONWINER driver folder. If you are using a Windows system, you can directly double-click to open CH341SER_Windows.EXE, install corresponding driver according to the computer operating system.

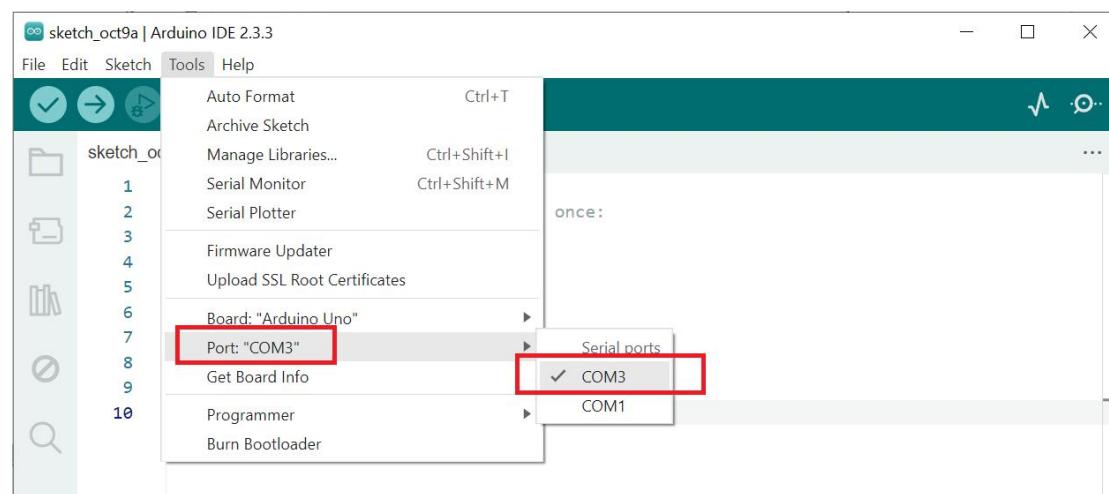
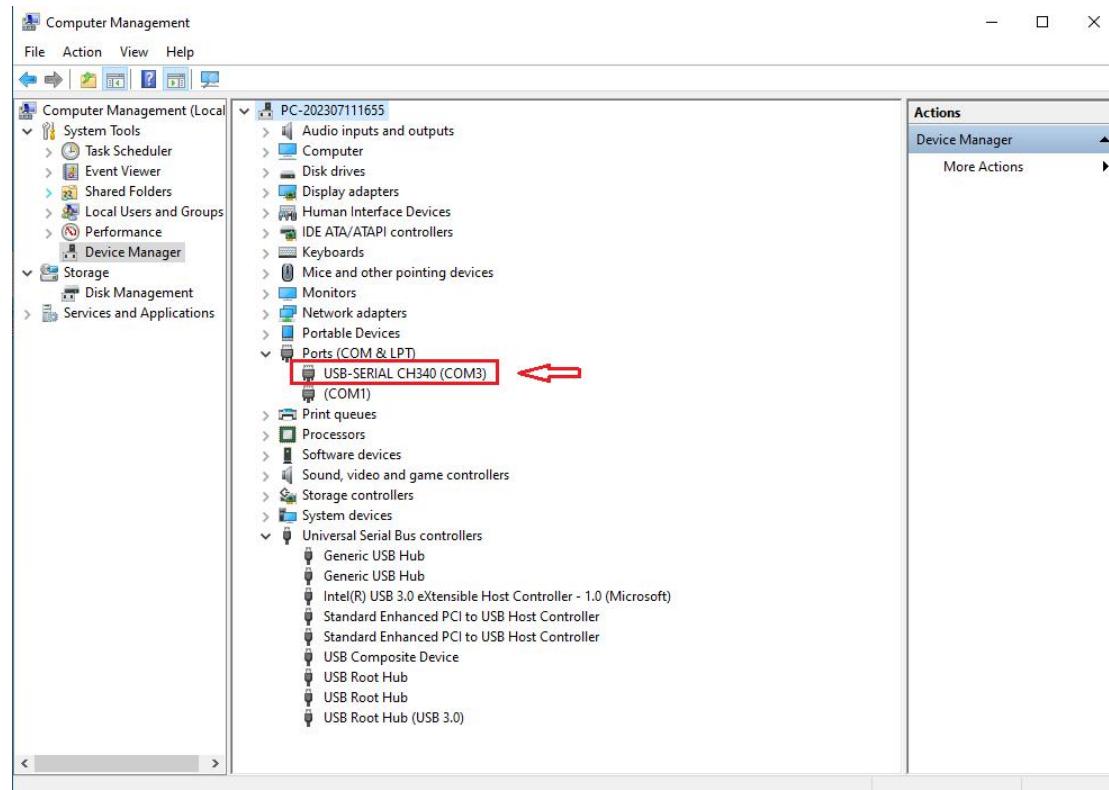


3. Click INSTALL. Wait for the installation to succeed. And click OK



4. Now you will find the Arduino serial port is accessible (different computer configuration has different serial port). It means that the Arduino UNO development board has been successfully connected to the computer. You will need to pay attention to this connection step in the following course.

blonwiner@outlook.com



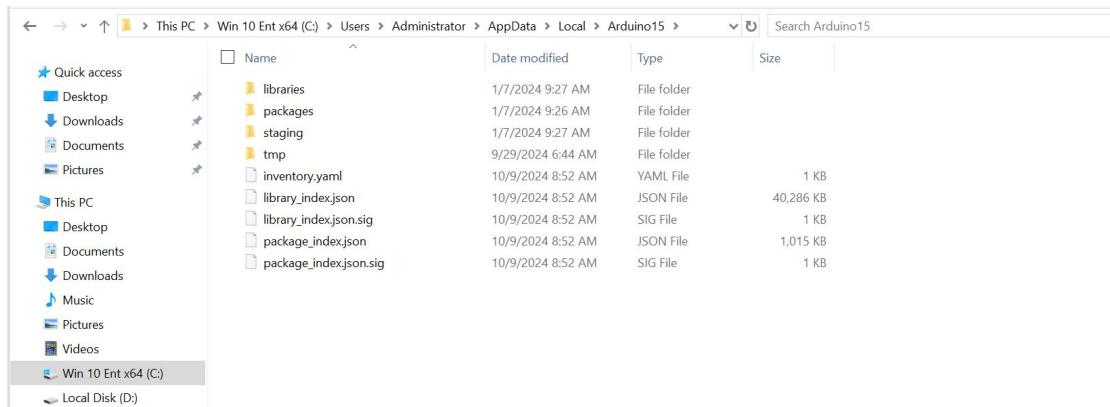
6. The solution for situation that Arduino IDE cannot be opened

When opening the Arduino IDE, you will suddenly encounter a situation that it cannot be opened.

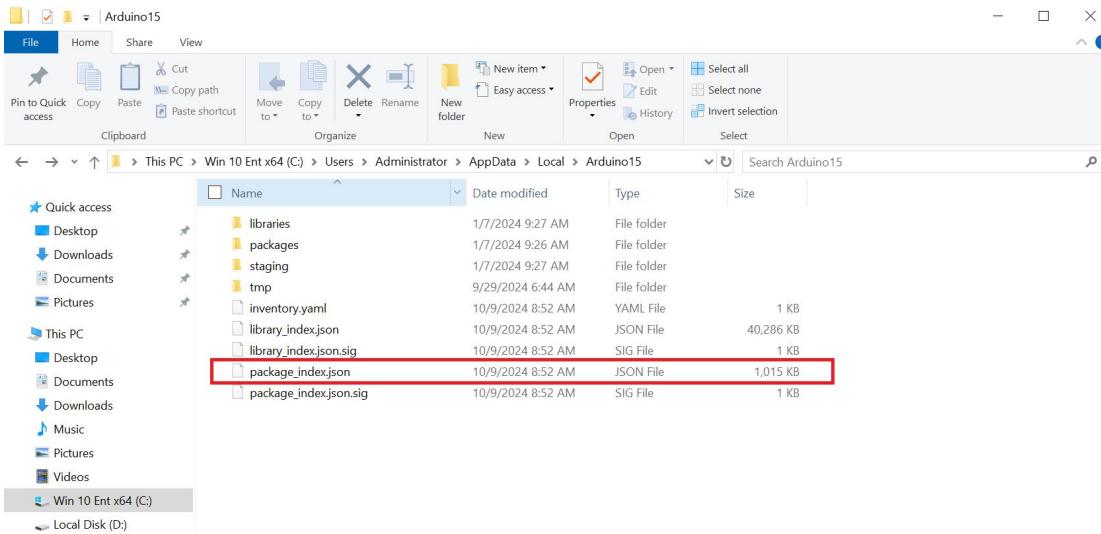


【Solution】

You need to find the Arduino15 folder in the C:\Users\Administrator\AppData\Local\Arduino15 directory of the C drive. As shown below:



You need to delete the package_index.json file, and then reopen the Arduino IDE.



7、Download Processing

Processing is a revolutionary and forward-looking new computer language. Its concept is to introduce programming languages in the environment of electronic art and introduce the concept of electronic art to programmers. It is an extension of the Java language and supports many existing Java language architectures. It is not only much simpler in syntax, but has many intimate and user-friendly designs. Processing can be used on Windows, MAC OS X, MAC OS 9, Linux and other operating systems.

The latest version is Processing 3. The work done in Processing can be used on the personal computer side or exported to the Internet in the form of Java Applets.

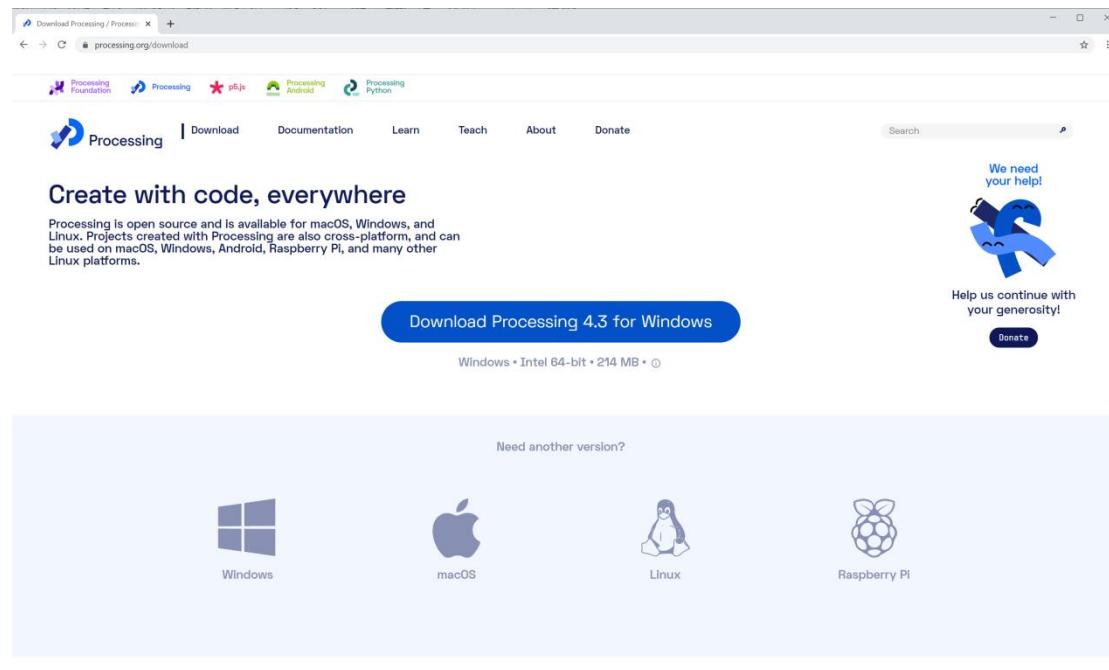
How to download Processing?

1. Enter this URL with Google Chrome:<https://processing.org/>

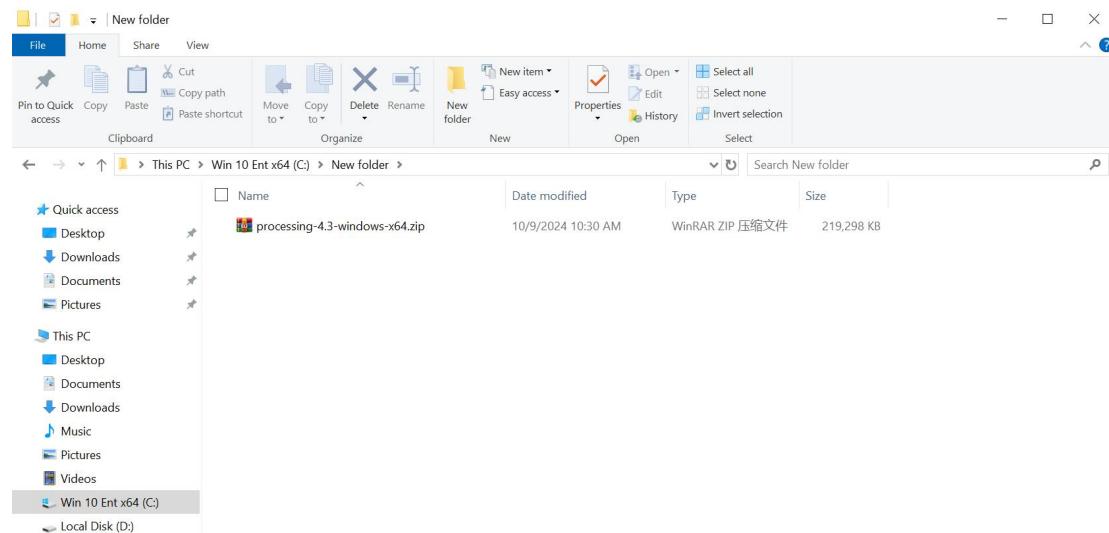
A screenshot of the Processing.org website. The header includes links for 'Processing Foundation', 'Processing', 'p5.js', 'Android', and 'Processing Python'. Below the header are navigation links for 'Download', 'Documentation', 'Learn', 'Teach', 'About', and 'Donate'. A search bar is on the right. The main content features a large blue 'P' logo on a grid background. Below the logo are sections for 'Welcome to Processing!', 'Examples' (with thumbnail images), and 'Open Editor'. At the bottom left are 'Download', 'Reference', and 'Donate' buttons.

2. Click Download Processing, as shown below:

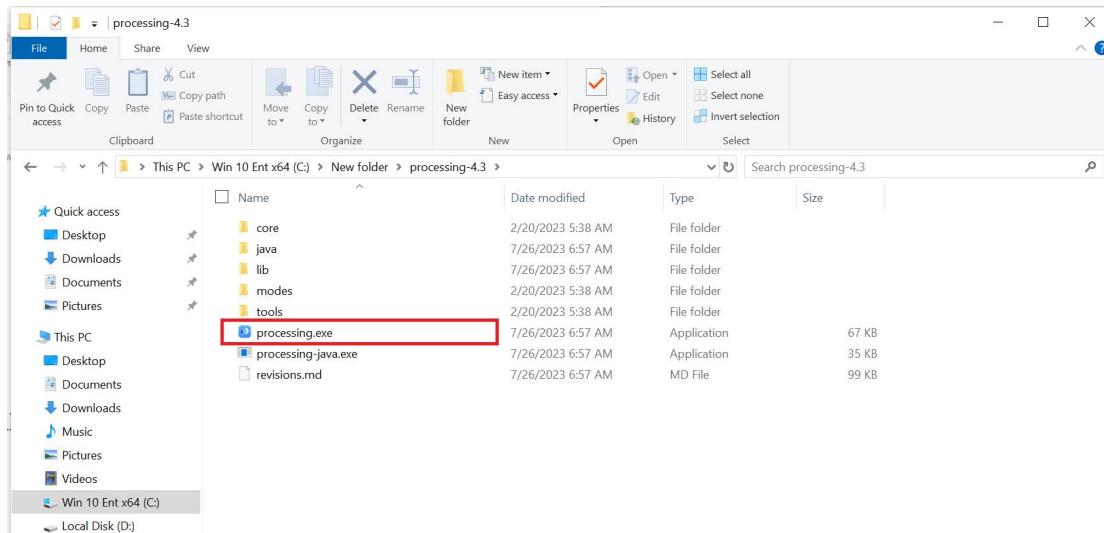
blonwiner@outlook.com



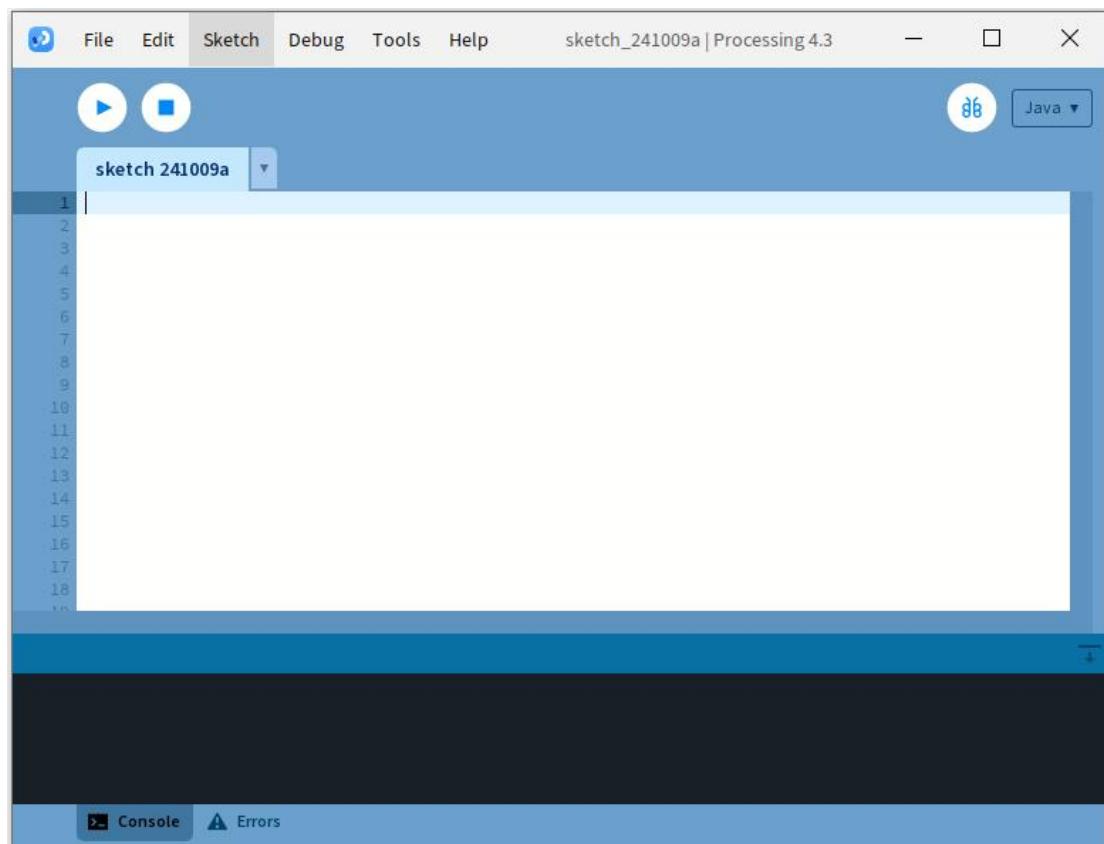
3. When finish downloading, you will get a compressed file "processing-4.3-windows-x64.zip".



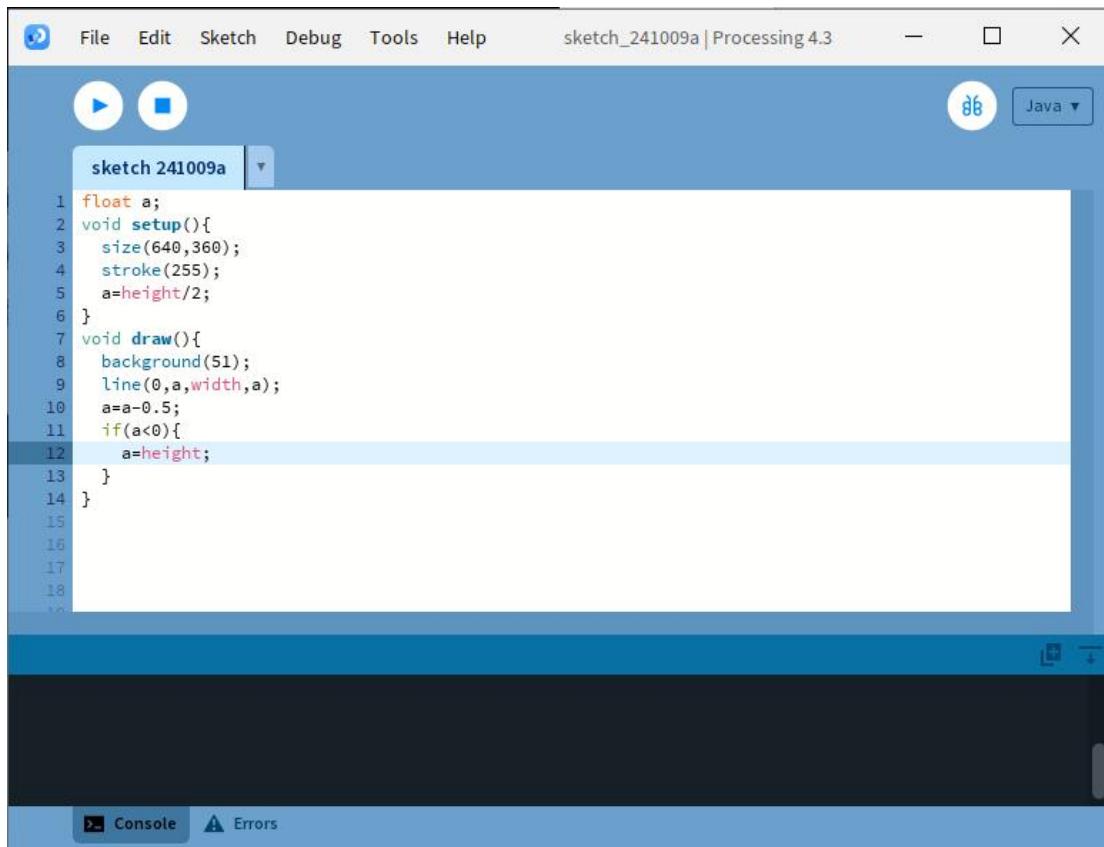
4. After extracting this file, you can get the following file, just click to run processing, it can be run directly without installation.



5.The interface is as follows after the Processing runs



6.Let's write a simple code that implements the following functions "Change the variable to create a moving line. When the line moves out of the window edge, the variable becomes 0 and the line goes back to the bottom of the screen

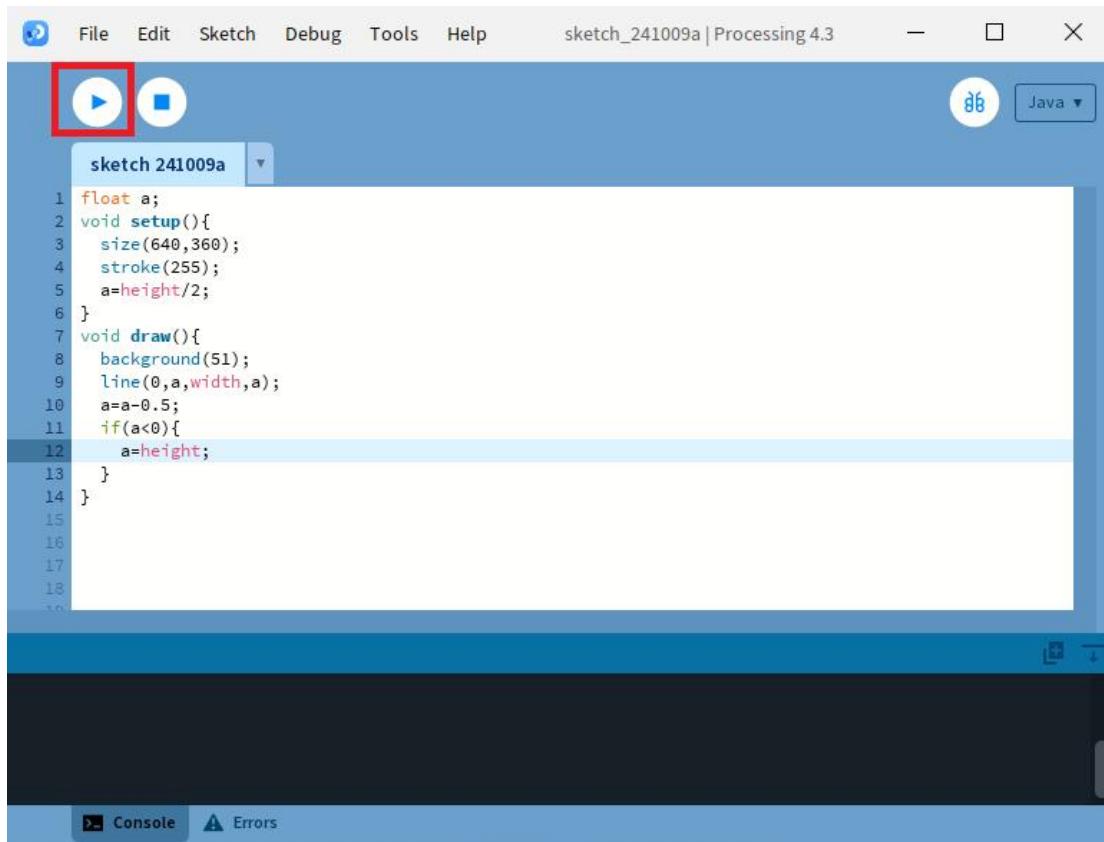


The screenshot shows the Processing 4.3 IDE interface. The menu bar includes File, Edit, Sketch, Debug, Tools, and Help. The title bar says "sketch_241009a | Processing 4.3". Below the menu is a toolbar with a play button (highlighted with a red box) and a stop button. The main area displays the following code:

```
1 float a;
2 void setup(){
3     size(640,360);
4     stroke(255);
5     a=height/2;
6 }
7 void draw(){
8     background(51);
9     line(0,a,width,a);
10    a=a-0.5;
11    if(a<0){
12        a=height;
13    }
14 }
```

The code defines a variable `a` and performs a vertical oscillation from the top to the bottom of the window. The `setup()` function sets the canvas size to 640x360 and initializes `a` to half the height. The `draw()` function repeatedly draws a horizontal line at position `a` and then moves `a` downwards by 0.5 units. If `a` reaches below zero, it is reset to the top height.

7. Click “Run”.

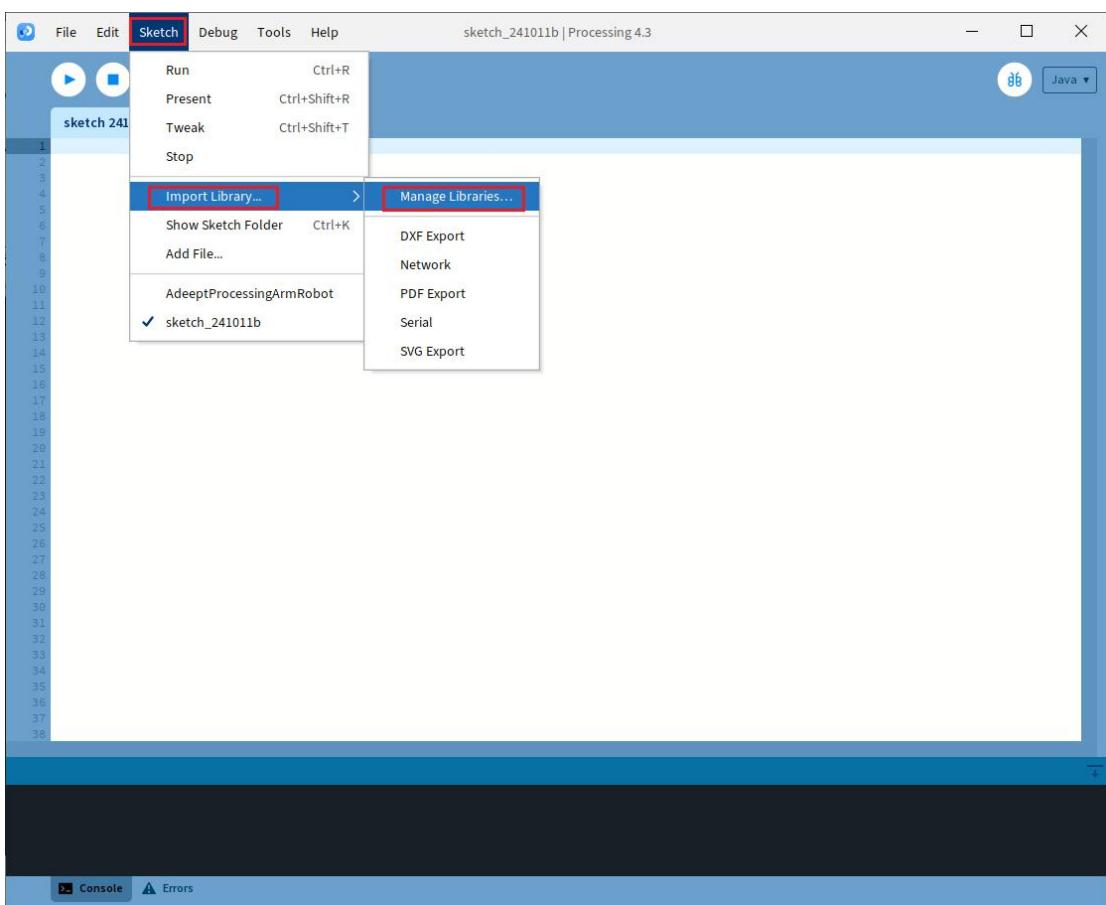


The screenshot shows the Processing 4.3 IDE interface, similar to the previous one but with a red box highlighting the play button in the toolbar. The code for sketch 241009a is identical to the previous screenshot. The interface is dark-themed.

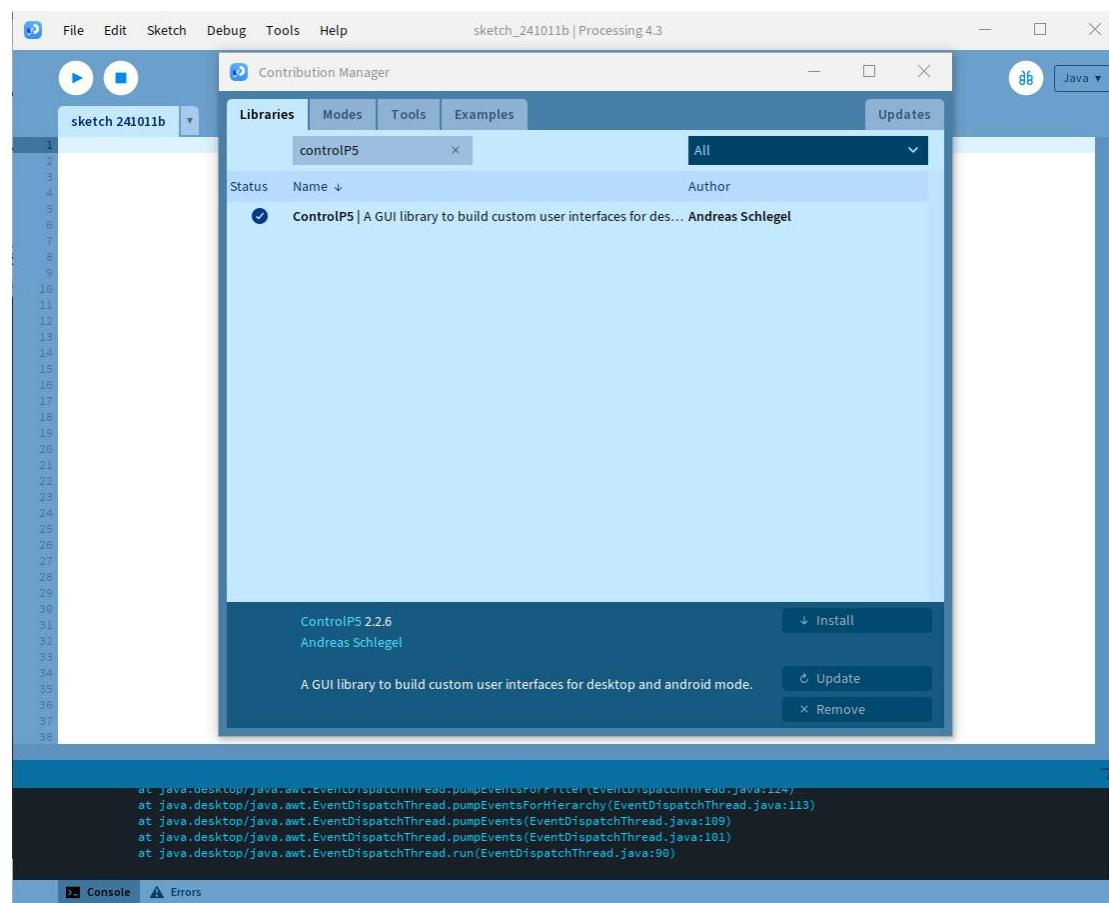
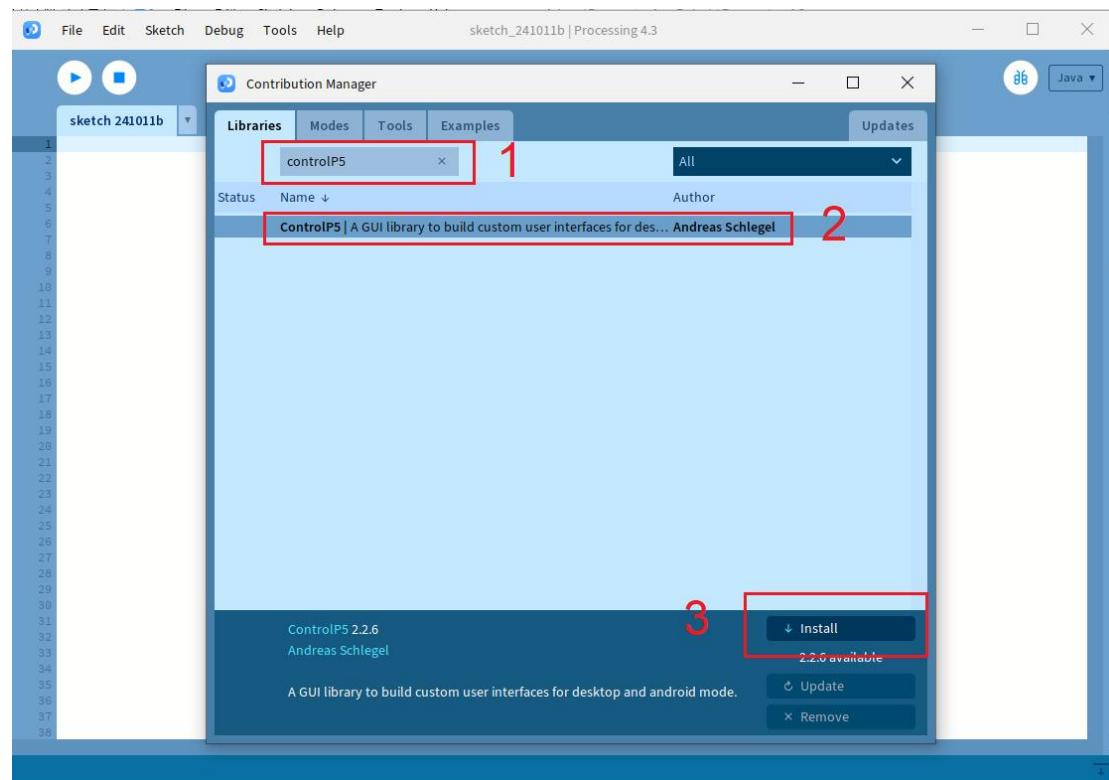
8. Running effect is as follow.



Install the controlP5 library,



Enter 'controlP5' here, select 'controlP5', and click 'Install'.



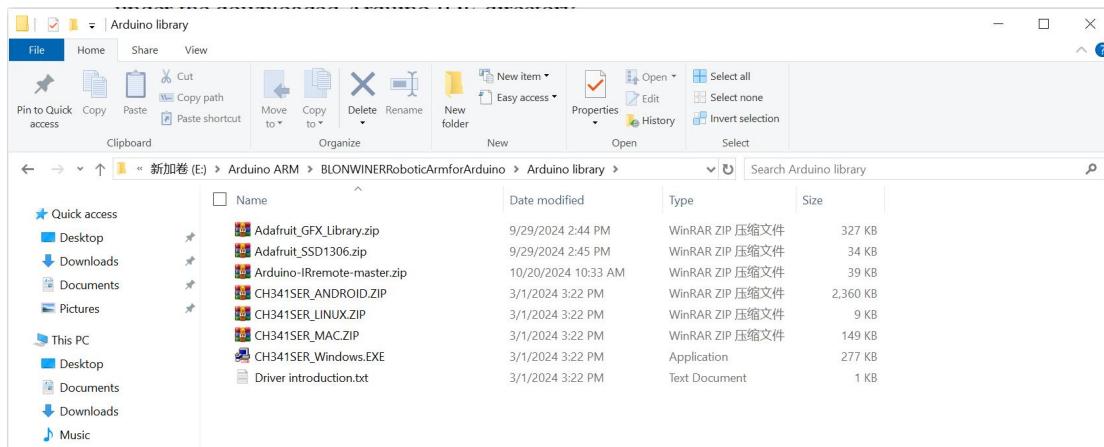
We need to upload a piece of code to the BLONWINER Arduino ARM Board before starting to assemble the robotic arm. Find out "Lesson 0 The servo initialization code of robotic arm assembly" in the documentation we provided and upload the code from the file to BLONWINER Arduino ARM Board.

8. Configuring the "libraries" folder of the Arduino IDE

Before using BLONWINER Robotic Arm, you need to configure the "libraries" folder under the downloaded Arduino IDE directory.

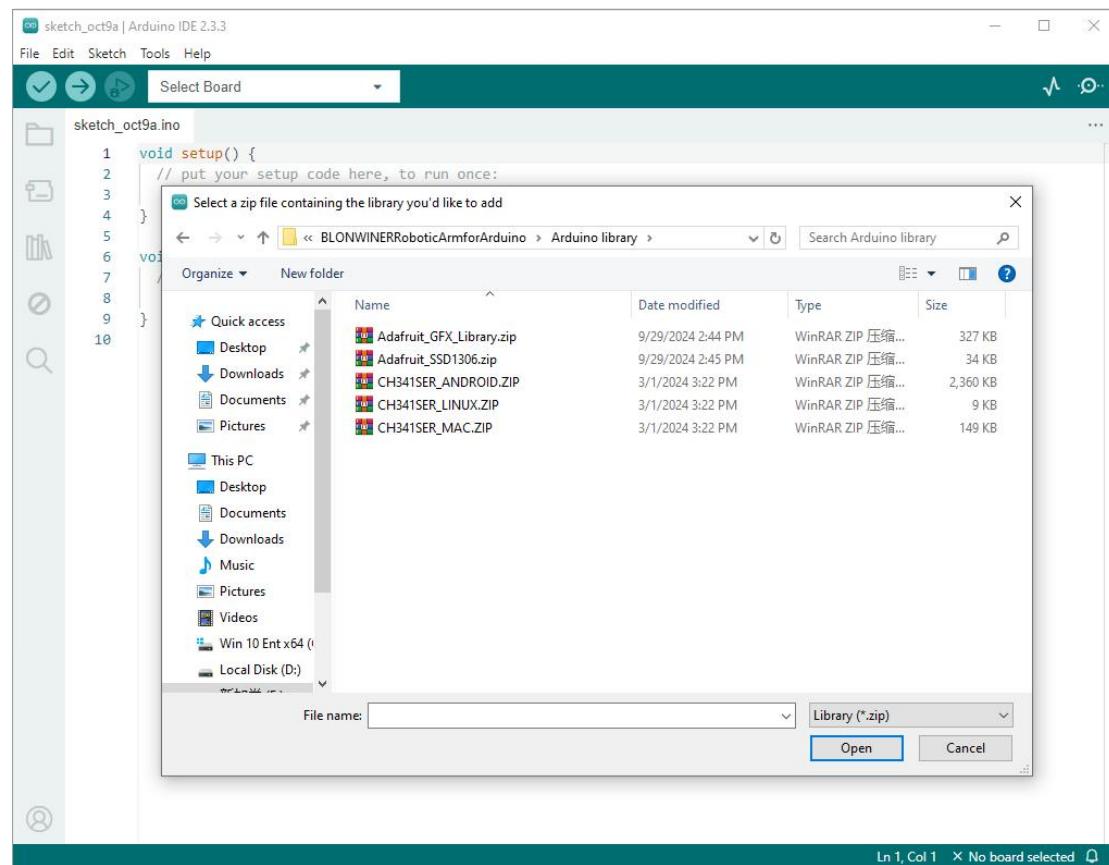
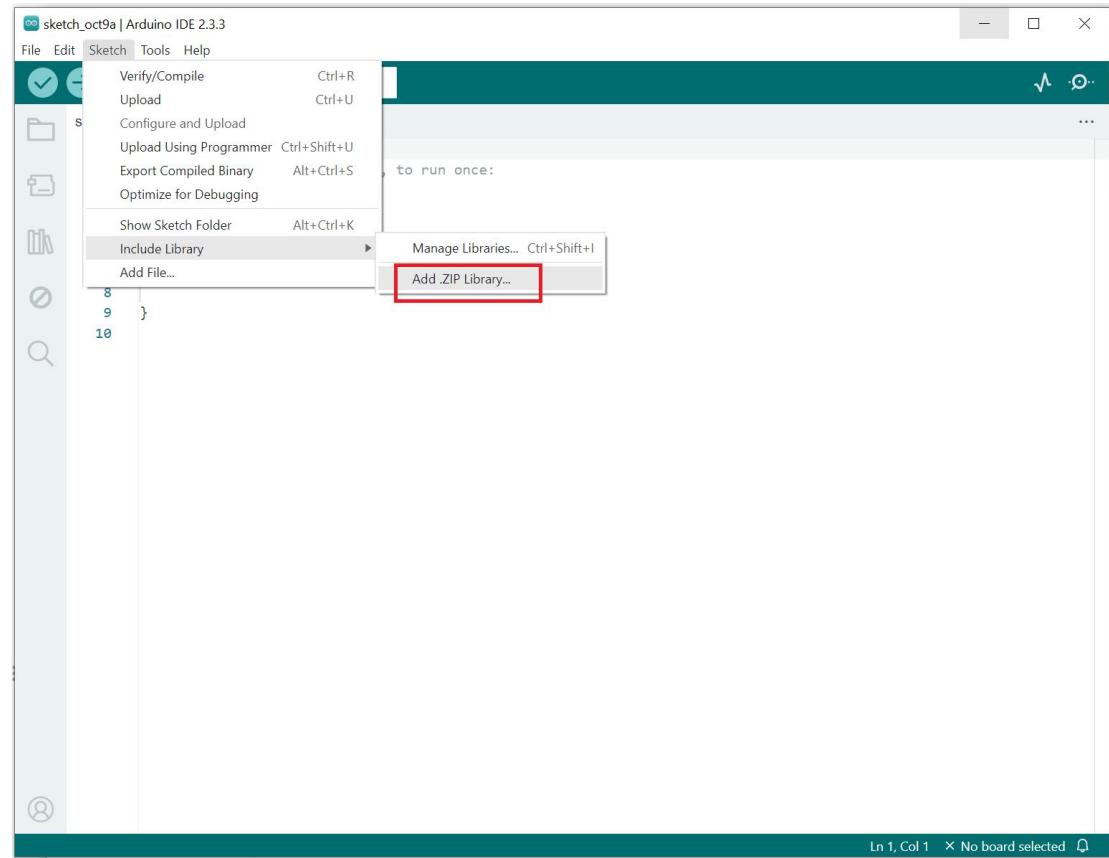
First, you need to find the user folder provided by BLONWINER:

BLONWINERRoboticArmforArduino, and find the "Arduino library" folder. Open the "Arduino library" folder, as shown below:



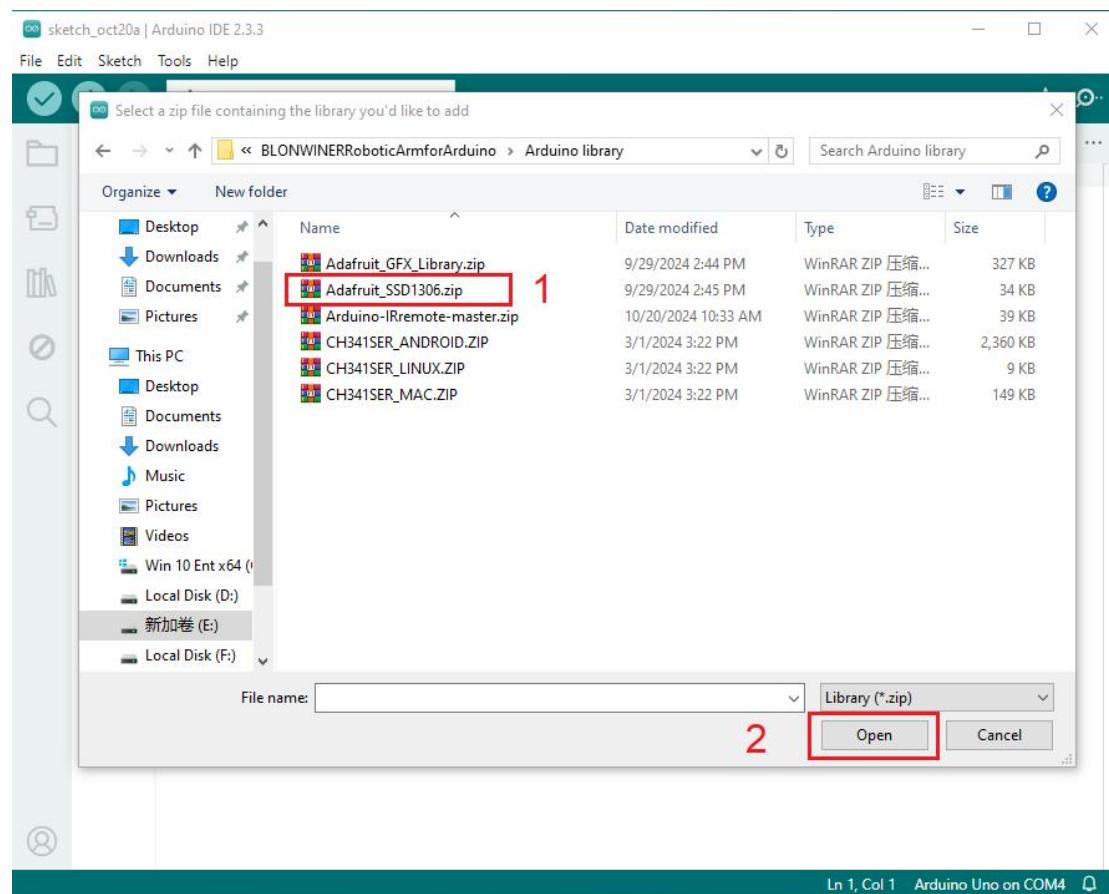
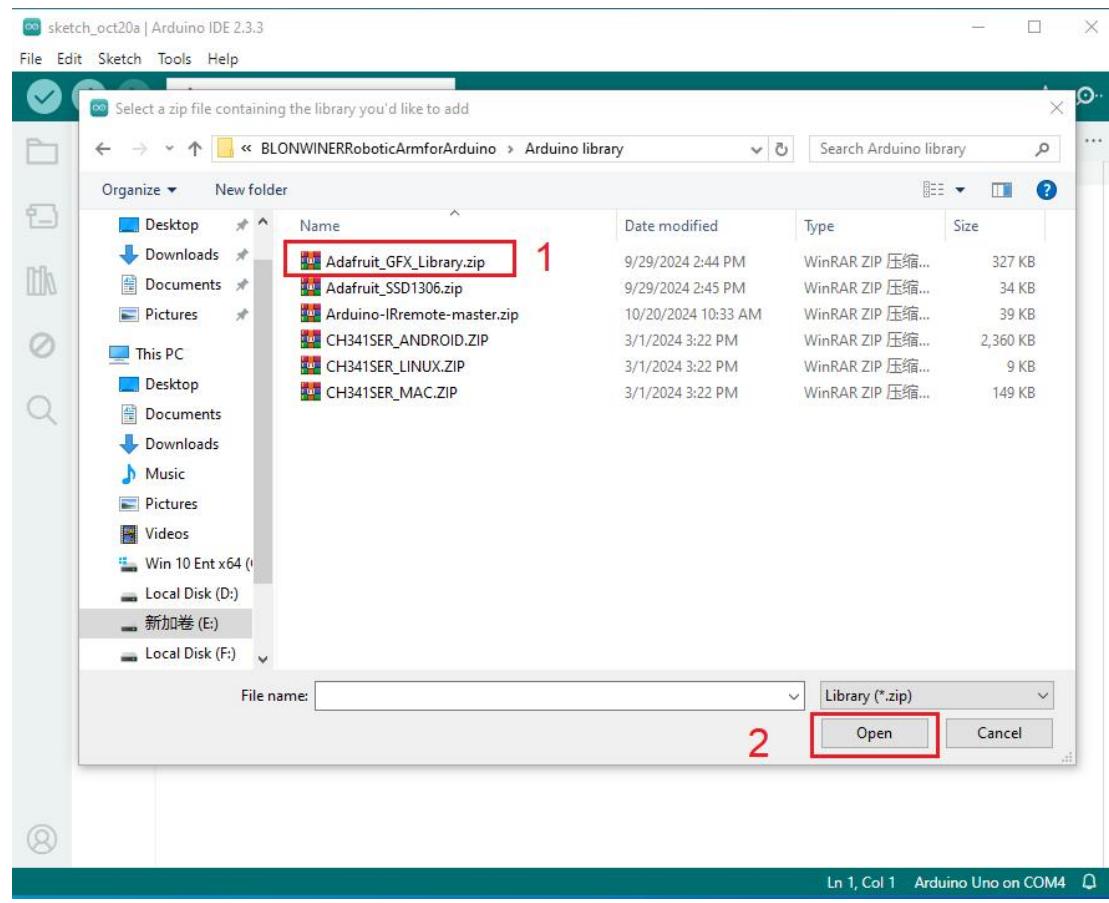
Open the arduino IDE software and select 'Add .ZIP Library...', as shown in the figure below

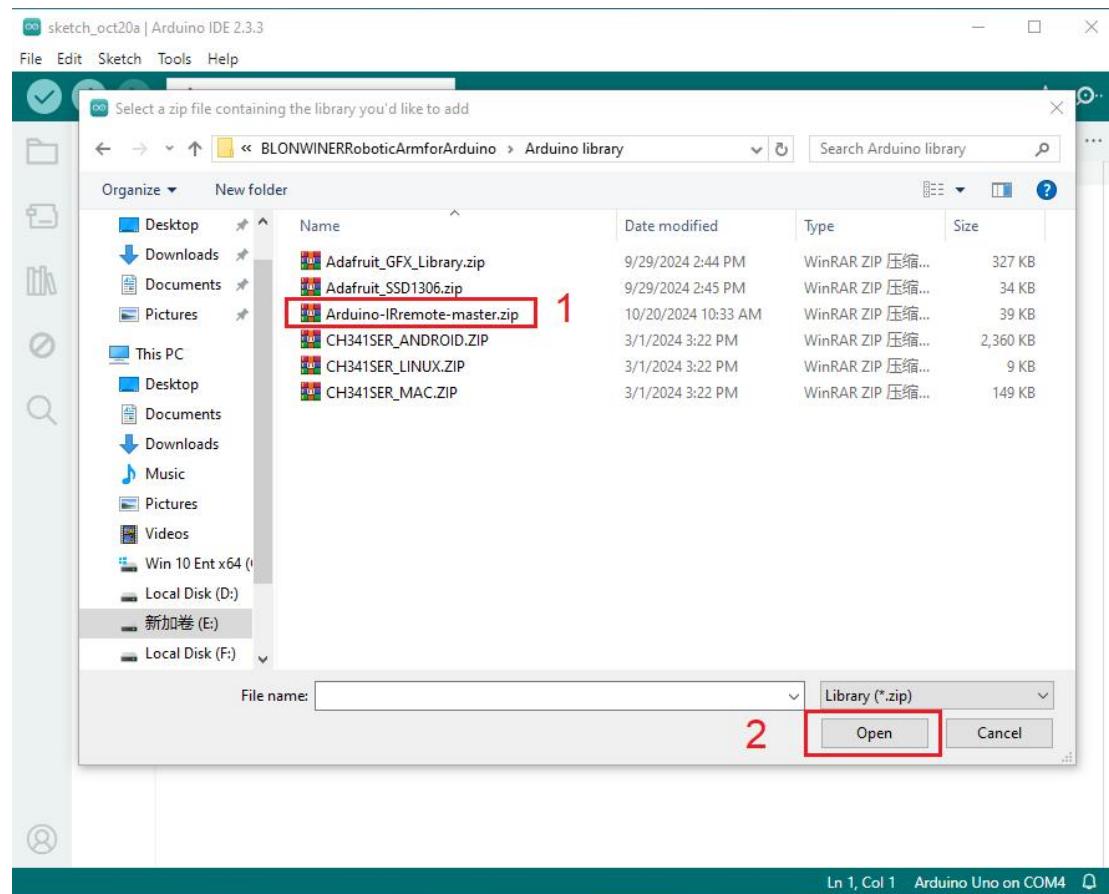
blonwiner@outlook.com



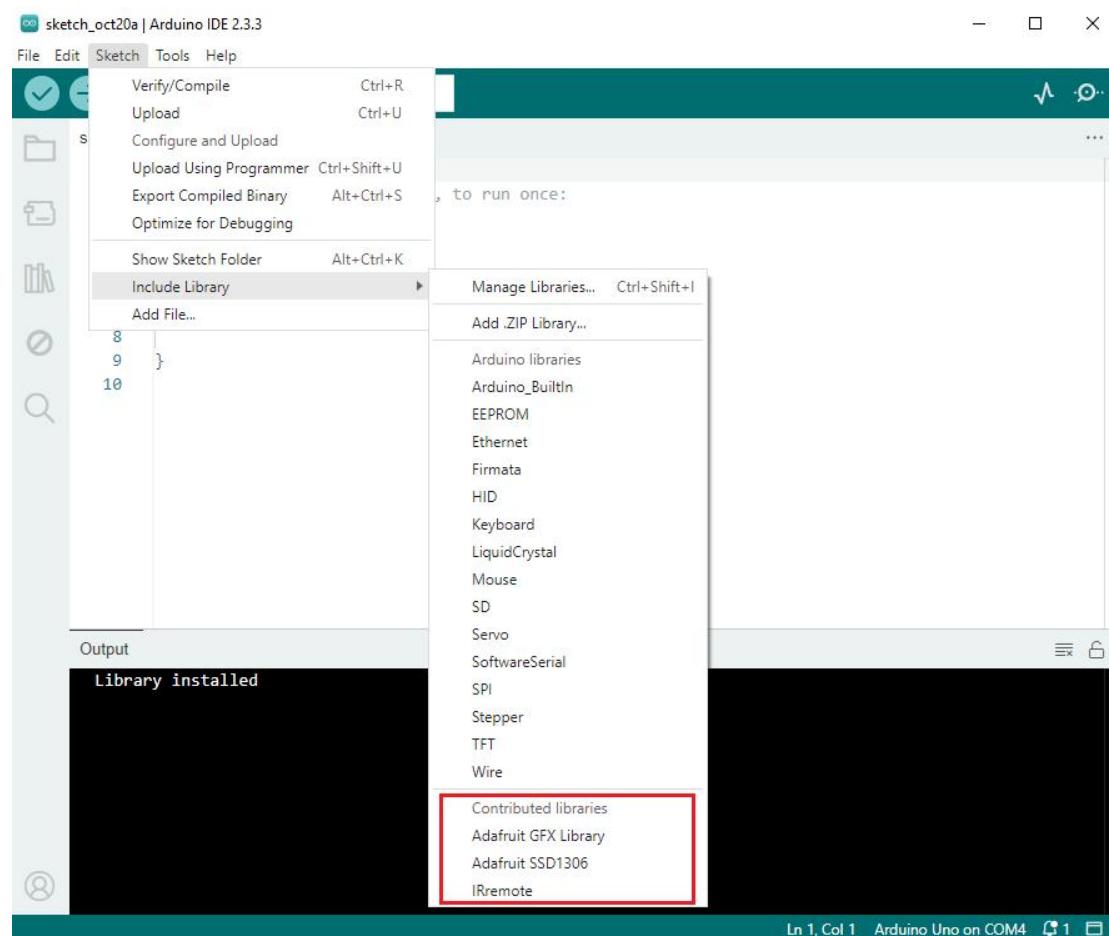
Add the two library files 'Adafruit_GFX_Library.zip' and 'Adafruit_SSD1306.zip' one by one.

blonwiner@outlook.com



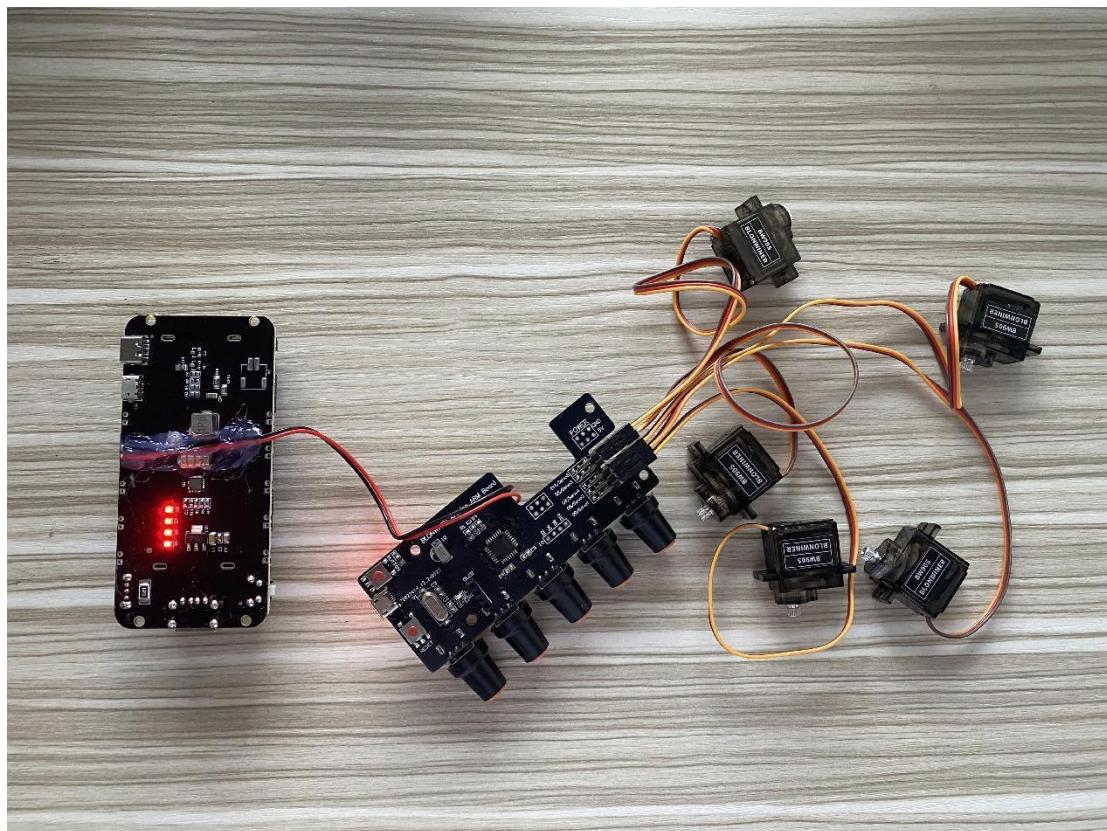


After successful installation, you can see that the Arduino IDE will appear these two library files.



Lesson 0 The servo initialization code of robotic arm assembly

Download the program to the BLONWINER Arduino ARM Board. The procedure is to control the servo to turn to the 90° position. Then refer to the assembly section at the back to start assembling the robotic arm.



Lesson 1 How to Read the Data of the Potentiometer

In this lesson, we will learn how to read the data of the potentiometer and convert the data into an angle.

1.1 Components used in this course

Components	Quantity	Picture
BLONWNER Arduino Arm Board	1	 A photograph of the BLONWNER Arduino Arm Board. It is a black rectangular board with various electronic components, including a central microcontroller chip, several pins, and some surface-mount components. It has a red LED and a small blue component labeled 'OLED'.
Micro USB Cable	1	 A photograph of a standard black Micro USB cable, coiled and shown against a white background.

1.2 Introduction of Potentiometer

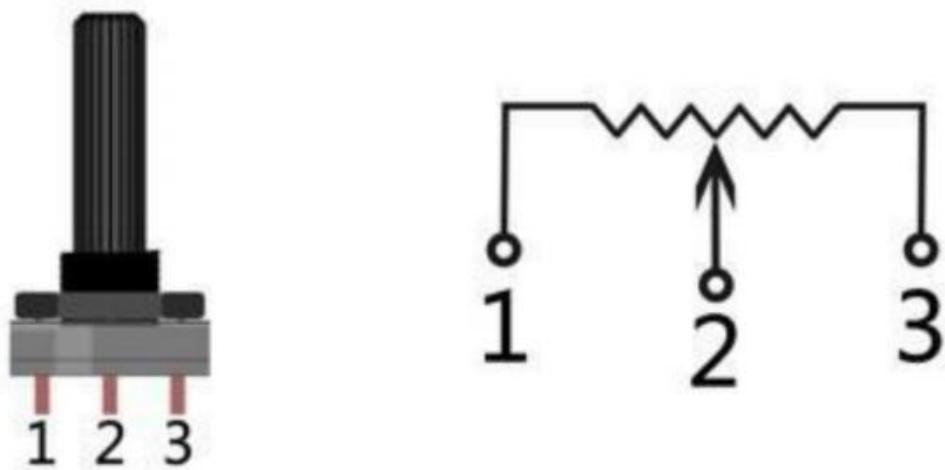
(1) Potentiometer

The potentiometer is a resistance element with three terminals and the resistance value can be adjusted according to a certain change law, which is equivalent to a variable resistor. Because its role in the circuit is to obtain a certain relationship with the input voltage (external voltage) to output Voltage, so called potentiometer. Potentiometers can be divided into rotary potentiometers, push-pull potentiometers, straight slide potentiometers, etc. according to the adjustment method. Our course experiment uses a rotary potentiometer. Its three pins are showed as below:

The rotary potentiometer is an adjustable resistance element. It is composed of a resistor and a rotating system. When a voltage is applied between the two fixed contacts of the resistive body, the position of the contact on the resistive body is changed by the rotating system, and a voltage that has a certain relationship with the position of the moving contact can be achieved between the moving contact and the fixed contact. Potentiometer can be used to adjust the voltage and current.

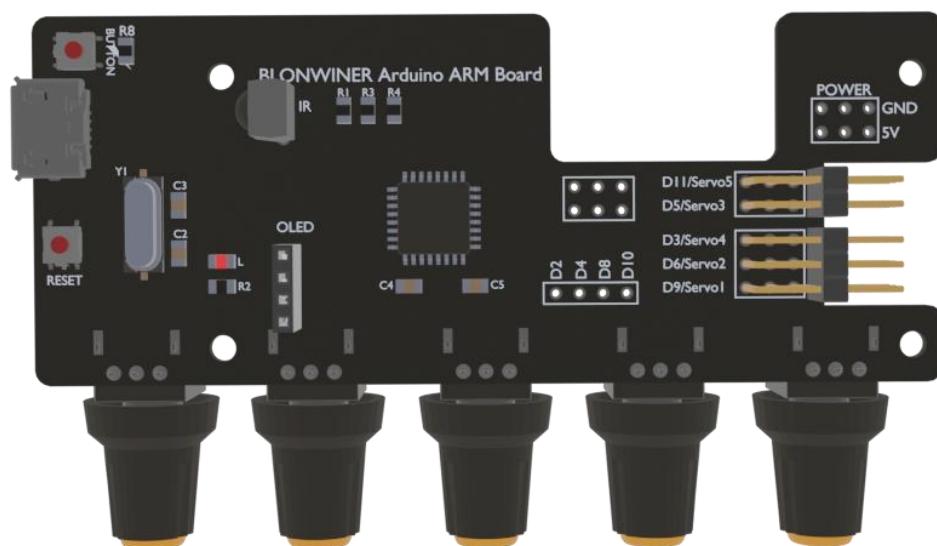
Our course uses a rotary potentiometer. Its structure is as shown in the figure below. By rotating the knob, the position of pin 2 is changed, thereby changing the resistance value from pin 2 to both ends. In the experiment. Connect pin 1 and pin 3 to the GND

and 5V of the development board respectively. And then read the voltage divided by the pin 2 of the potentiometer through the analog input pin A0. The range is between 0V and 5V. The analog input function of Arduino has 10-bit precision, that is, it can convert the voltage signal of 0 to 5V into an integer form of 0 to 1024.

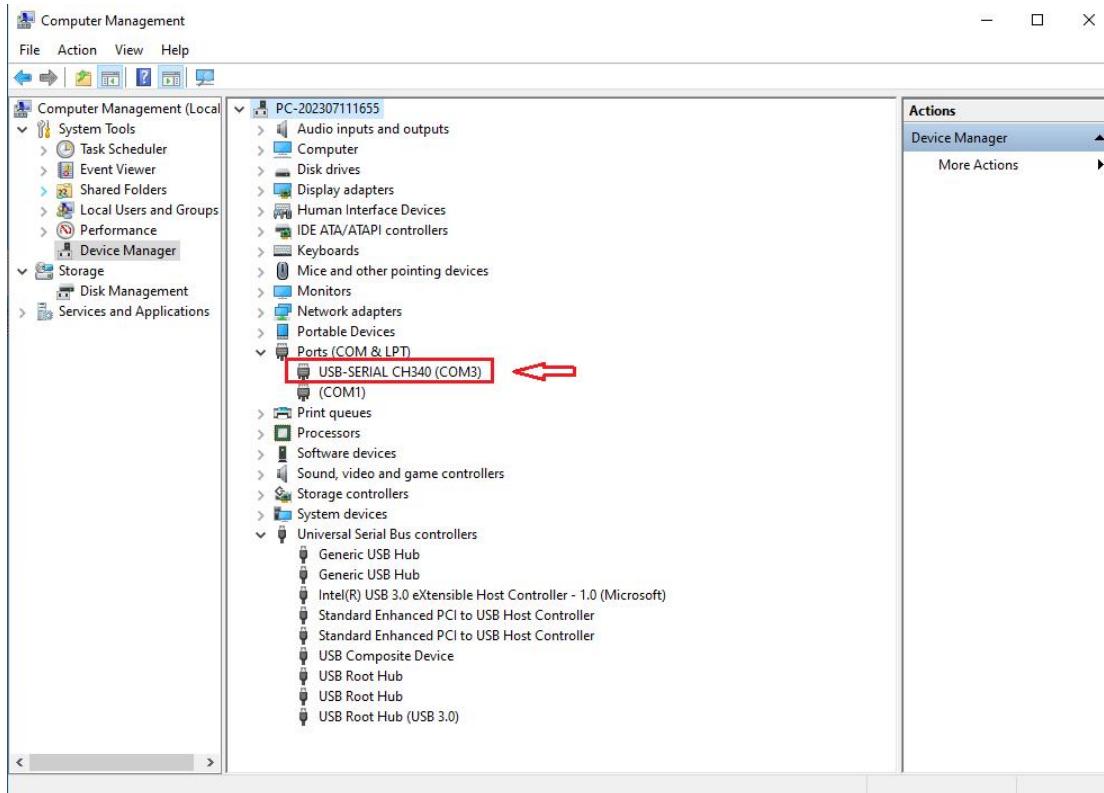


1.3 Wiring diagram (Circuit diagram)

Figure as below:



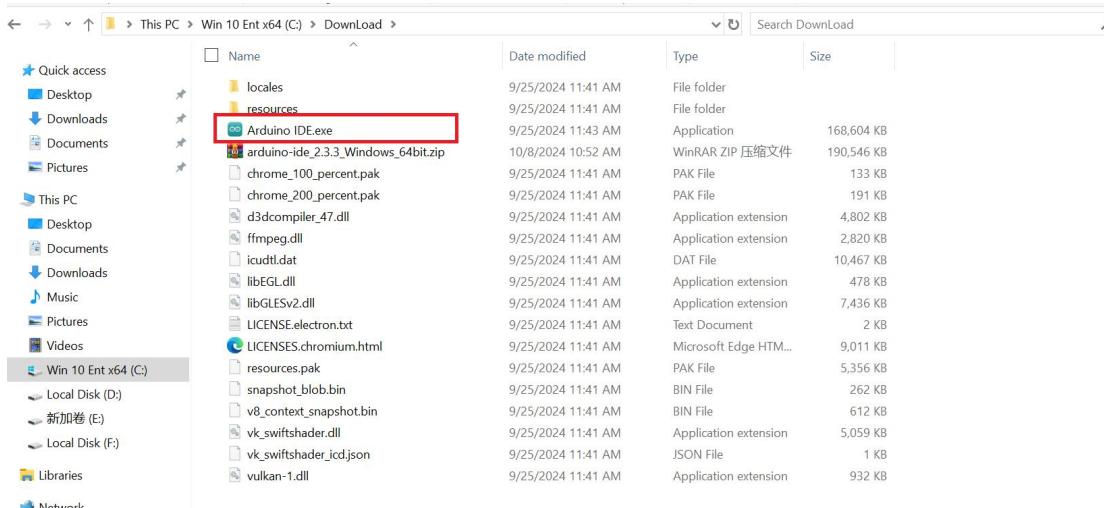
BLOWINER Arduino Arm Board is connected to a computer. Once you have successfully connected to your computer, Computer Management will display this.



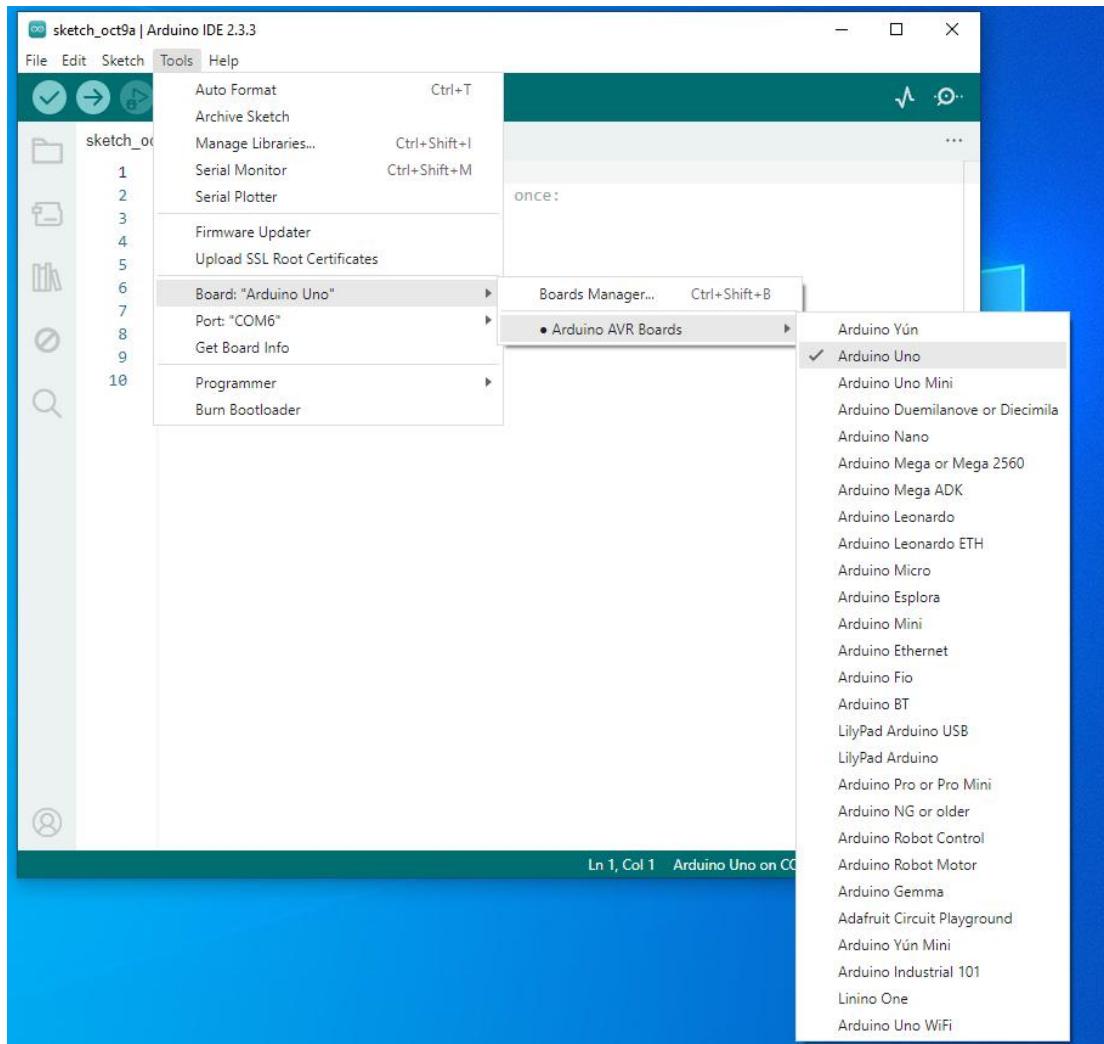
1.4 Reading the value of the potentiometer and converting it into an angle

1.4.1 Compile and run the code program of this course

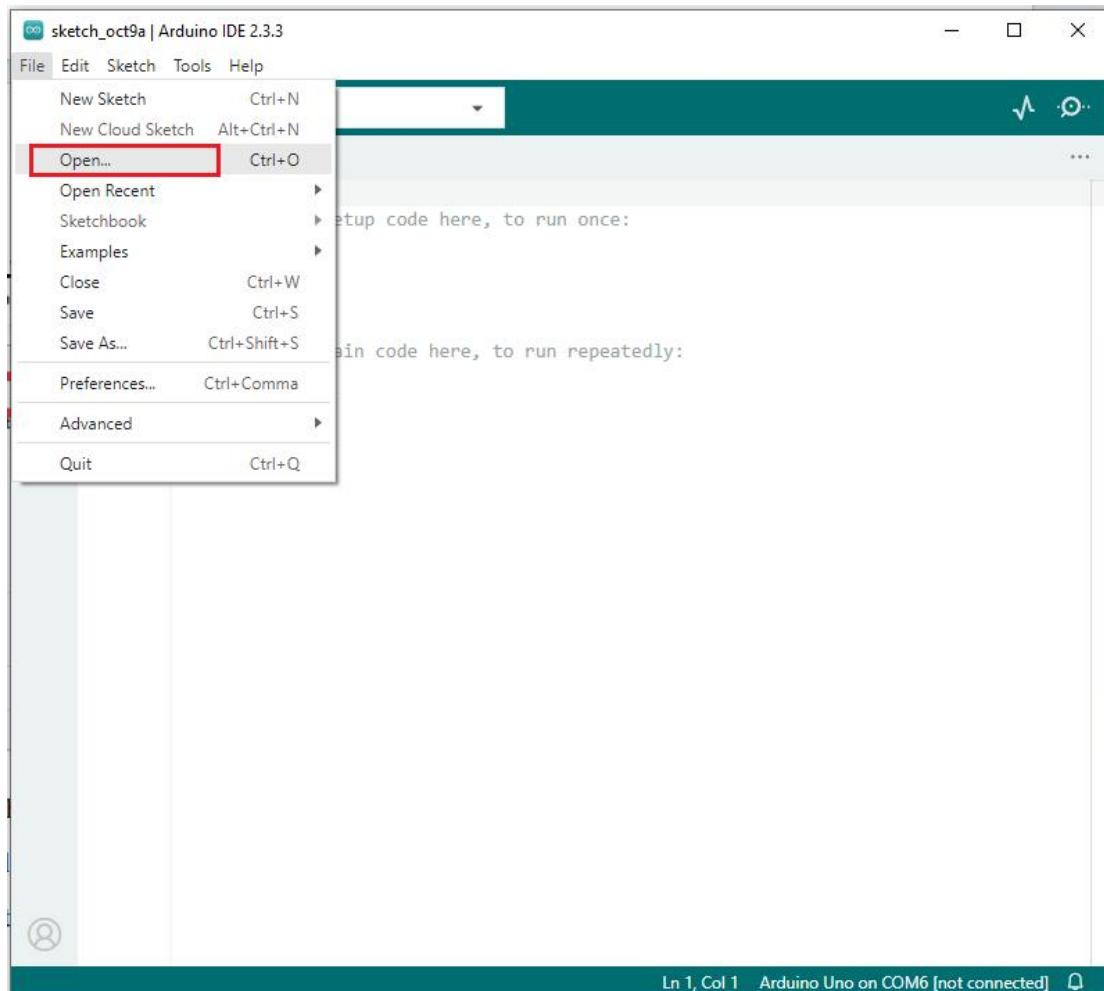
1. Open the Arduino IDE software, as shown below:



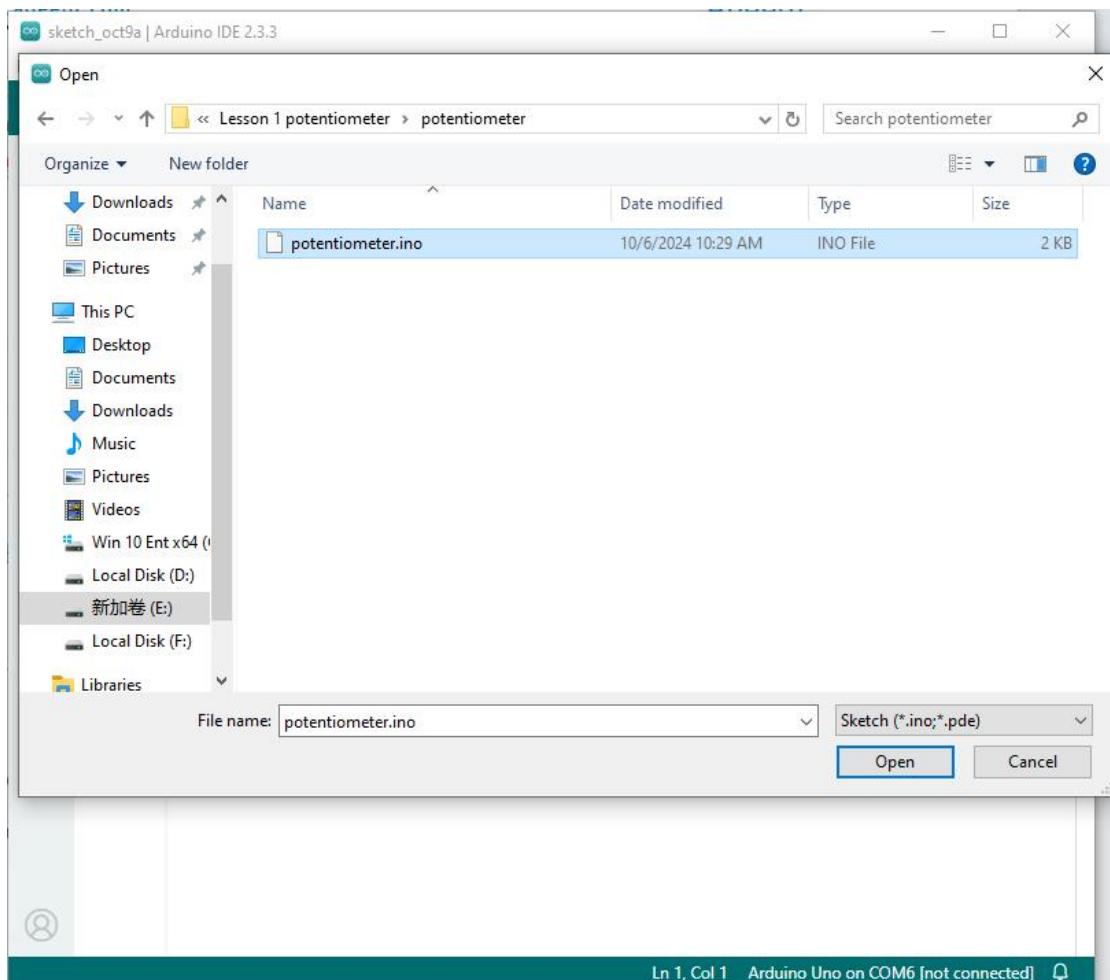
2. In the Tools toolbar, find Board and select Arduino Uno, as shown below:



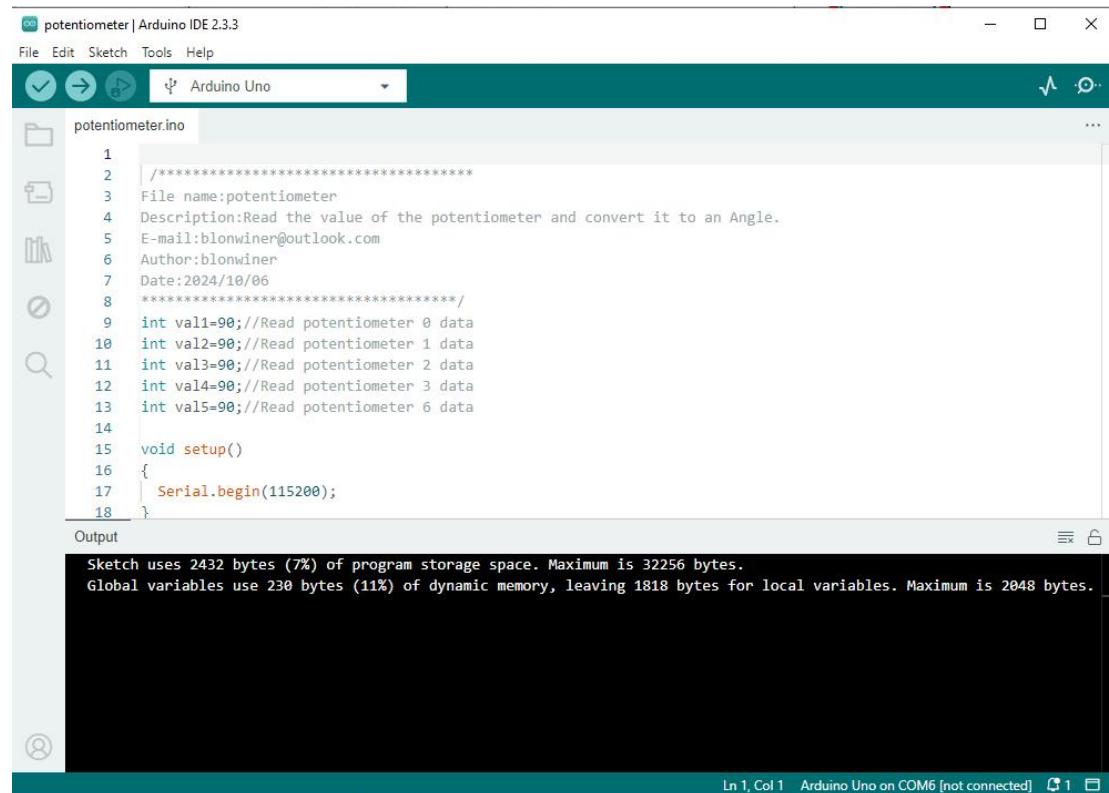
3.Click Open in the File drop-down menu:



4. Find the folder BLOWINERRoboticArmforArduino that we provide to the user. Enter the Lesson 1 potentiometer directory. Select potentiometer.ino. This file is the code program we need in this course. Then click Open.



5. After opening, click  to upload the code program to the Arduino UNO. If there is no error warning in the console below, it means that the Upload is successful.



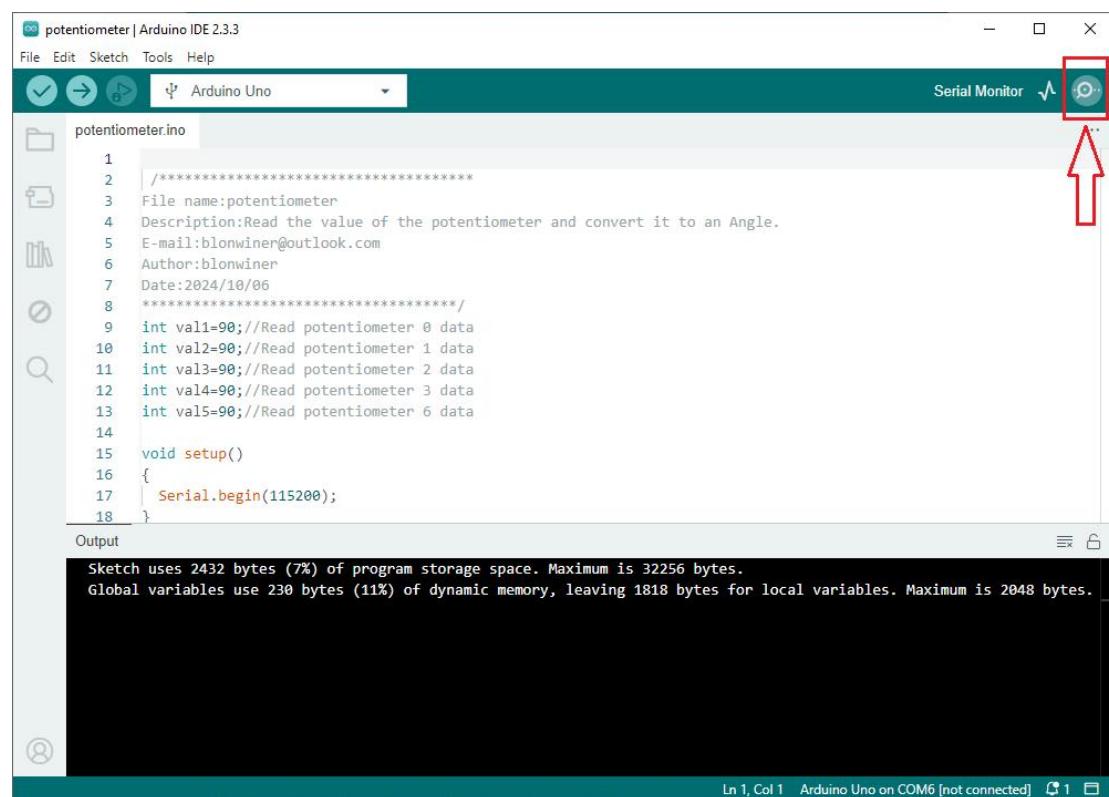
```
1 //*****
2 // File name:potentiometer
3 // Description:Read the value of the potentiometer and convert it to an Angle.
4 // E-mail:blonwiner@outlook.com
5 // Author:blonwiner
6 // Date:2024/10/06
7 //*****
8 int val1=90;//Read potentiometer 0 data
9 int val2=90;//Read potentiometer 1 data
10 int val3=90;//Read potentiometer 2 data
11 int val4=90;//Read potentiometer 3 data
12 int val5=90;//Read potentiometer 6 data
13
14 void setup()
15 {
16     Serial.begin(115200);
17 }
18 
```

Output

```
Sketch uses 2432 bytes (7%) of program storage space. Maximum is 32256 bytes.
Global variables use 230 bytes (11%) of dynamic memory, leaving 1818 bytes for local variables. Maximum is 2048 bytes.
```

Ln 1, Col 1 Arduino Uno on COM6 [not connected] 1

6. After successfully running the program, we need to observe the value of the potentiometer by opening the serial monitor and click  , as shown in the figure below:



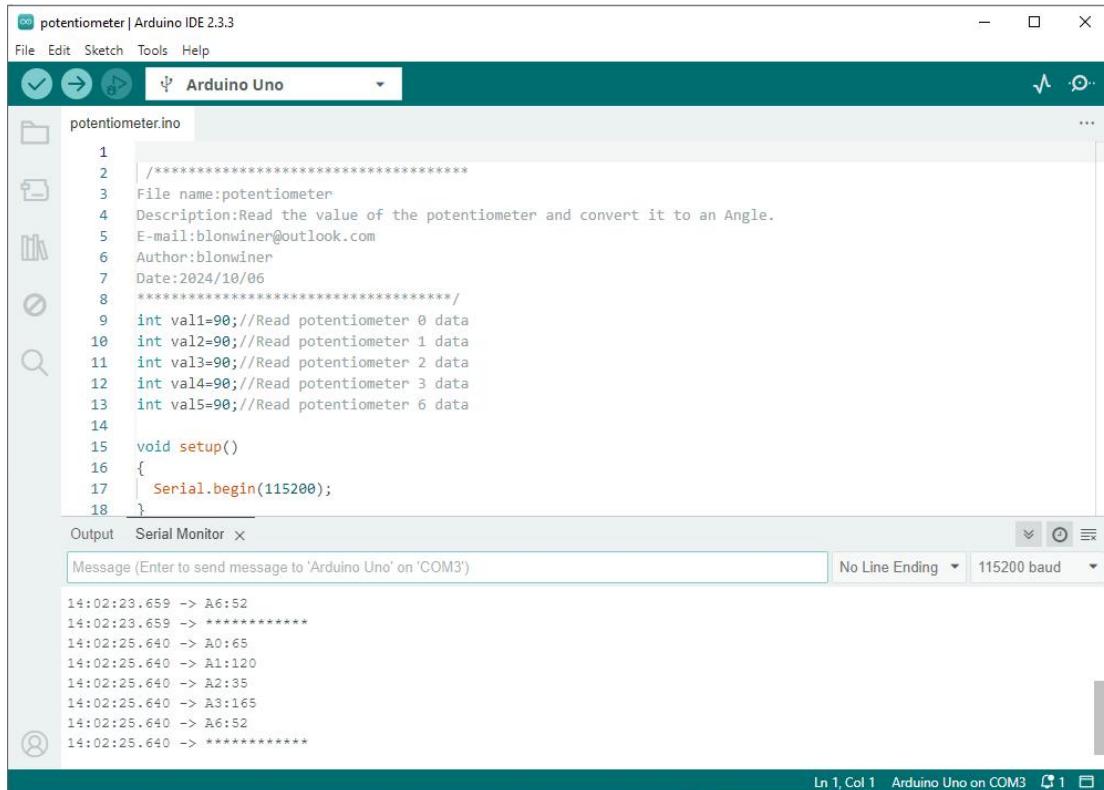
```
1 //*****
2 // File name:potentiometer
3 // Description:Read the value of the potentiometer and convert it to an Angle.
4 // E-mail:blonwiner@outlook.com
5 // Author:blonwiner
6 // Date:2024/10/06
7 //*****
8 int val1=90;//Read potentiometer 0 data
9 int val2=90;//Read potentiometer 1 data
10 int val3=90;//Read potentiometer 2 data
11 int val4=90;//Read potentiometer 3 data
12 int val5=90;//Read potentiometer 6 data
13
14 void setup()
15 {
16     Serial.begin(115200);
17 }
18 
```

Output

```
Sketch uses 2432 bytes (7%) of program storage space. Maximum is 32256 bytes.
Global variables use 230 bytes (11%) of dynamic memory, leaving 1818 bytes for local variables. Maximum is 2048 bytes.
```

Ln 1, Col 1 Arduino Uno on COM6 [not connected] 1

Then open the serial monitor, you need to modify the displayed bit rate and the bit rate set in the code to 115200, so that the display will not appear garbled. You can observe the data changes corresponding to each button by rotating the buttons of A0, A1, A2, A3, and A6. When the buttons of A0, A1, A2, and A3 are rotated, the data change range is from 0 to 180. The data becomes smaller when rotating clockwise, and the data becomes larger when rotating counterclockwise. When the A6 button is rotated, the data change range is 35~90. When it is rotated clockwise, the data becomes smaller, and when it is rotated counterclockwise, the data becomes larger.



The screenshot shows the Arduino IDE interface with the following details:

- Title Bar:** potentiometer | Arduino IDE 2.3.3
- File Menu:** File Edit Sketch Tools Help
- Sketch Selection:** Arduino Uno
- Code Area:** The code is named "potentiometer.ino". It includes comments for file name, description, author, and date. It defines five integer variables (val1-val5) for potentiometers A0-A4 and A6. The setup() function initializes the serial port at 115200 baud.
- Output Area:** The "Serial Monitor" tab is selected. The message input field contains "Message (Enter to send message to 'Arduino Uno' on 'COM3')". The baud rate is set to "115200 baud".
- Serial Monitor Output:** The output shows the following data:

```
14:02:23.659 -> A6:52
14:02:23.659 -> ****
14:02:25.640 -> A0:65
14:02:25.640 -> A1:120
14:02:25.640 -> A2:35
14:02:25.640 -> A3:165
14:02:25.640 -> A6:52
14:02:25.640 -> ****
```
- Status Bar:** Ln 1, Col 1 Arduino Uno on COM3

1.4.2 Learning the code program of this lesson

Initialize potentiometers A0, A1, A2, A3, A6.

```
9  int val1=90;//Read potentiometer 0 data
10 int val2=90;//Read potentiometer 1 data
11 int val3=90;//Read potentiometer 2 data
12 int val4=90;//Read potentiometer 3 data
13 int val5=90;//Read potentiometer 6 data
14
15 void setup()
16 {
17 | Serial.begin(115200);
18 }
```

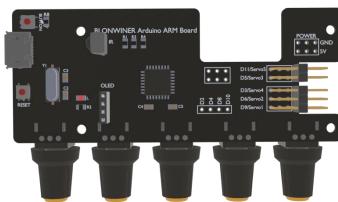
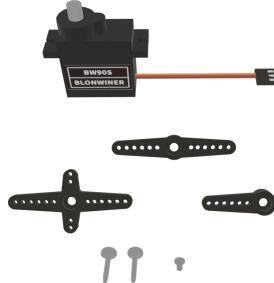
Convert the value of 1023 to 180 proportionally, and then print out the converted data to the serial monitor.

```
19 void loop()
20 {
21     val1 = map(analogRead(0), 0, 1023, 0, 180);
22     val2 = map(analogRead(1), 0, 1023, 0, 180);
23     val3 = map(analogRead(2), 0, 1023, 0, 180);
24     val4 = map(analogRead(3), 0, 1023, 0, 180);
25     val5 = map(analogRead(6), 0, 1023, 35, 90);
26     Serial.print("A0:");Serial.println(val1);
27     Serial.print("A1:");Serial.println(val2);
28     Serial.print("A2:");Serial.println(val3);
29     Serial.print("A3:");Serial.println(val4);
30     Serial.print("A6:");Serial.println(val5);
31     Serial.println("*****");
32     delay(2000);
33 }
```

Lesson 2 Controlling the Servo

In this lesson, we will learn how to control the Servo.

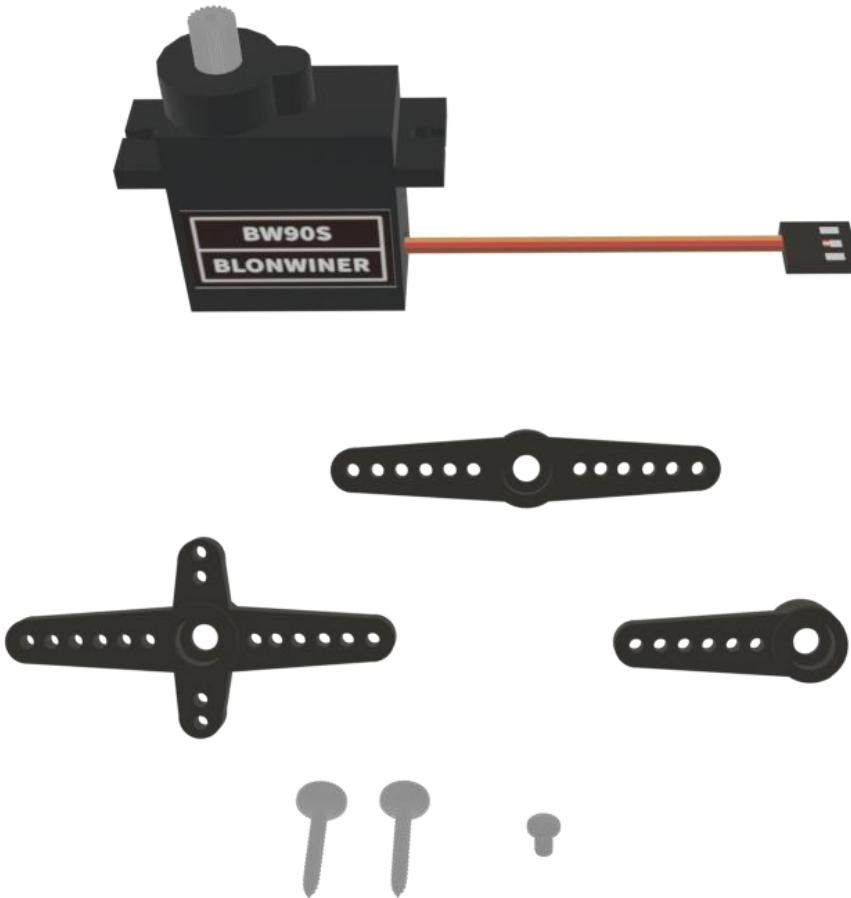
2.1 Components used in this course

Components	Quantity	Picture
BLONWINER Arduino Arm Board	1	 A black printed circuit board labeled "BLONWINER Arduino ARM Board". It features several components: a central microcontroller chip, a red LED, a green LED, a small potentiometer, and four black female headers labeled D1, D2, D3, and D4. The board has a white background with black text and symbols.
Micro USB Cable	1	 A black Micro USB cable with a white plastic protective套 on the left end.
Servo	1	 A collection of servo-related parts. At the top is a black servo motor with a red lead wire. Below it are two black servo horns. At the bottom are three metal servo mounting brackets.

2.2 The introduction of the Servo

2.2.1 Servo

Servo motor refers to the engine that controls mechanical component operation in the servo system. It is a kind of auxiliary motor indirect transmission device. The servo motor is a gear motor that can rotate only 180 degrees. It is controlled by sending pulses from the microcontroller. These pulses tell the server where to move. The servo motor system includes housing, circuit board, non-core motor, gearing and position detection. Servo motor is shown in the figure:

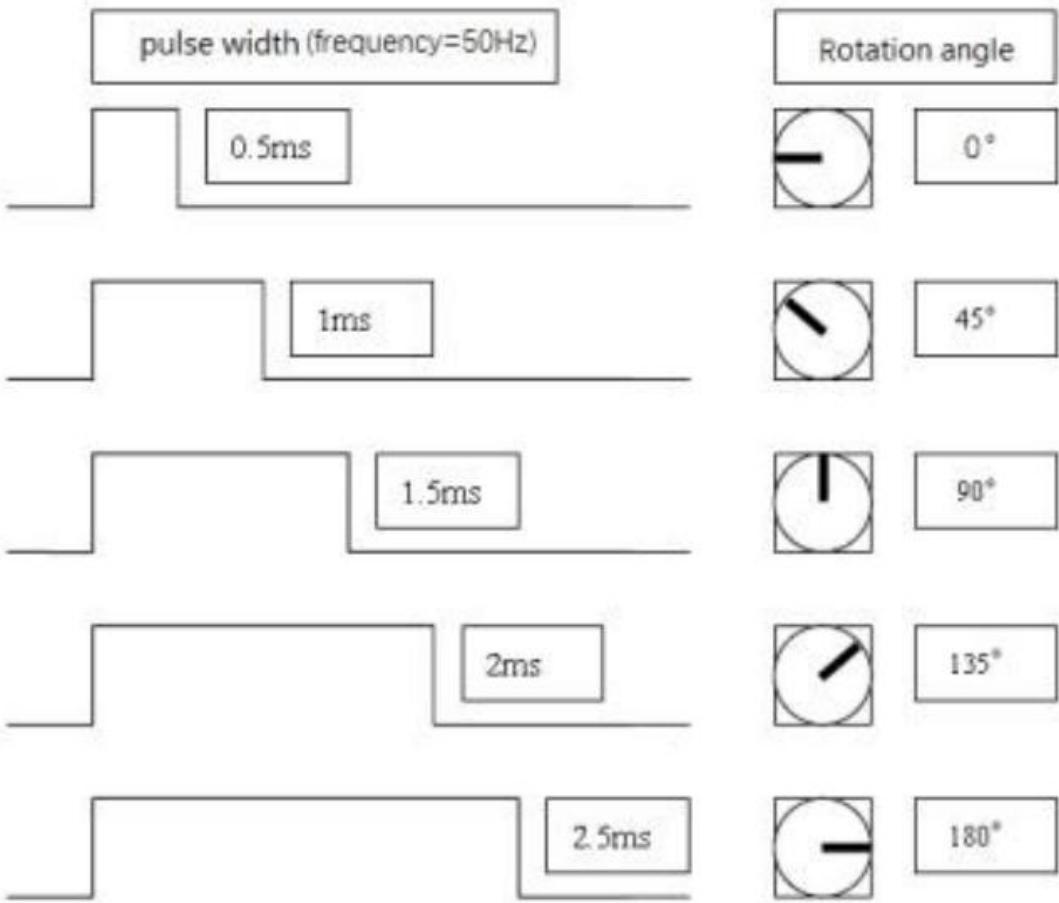


2.2.2 The working principle of the Servo

The servo mechanism is an automatic control system that enables the object's position, orientation, state and other output controlled quantities to follow arbitrary changes in the input target (or given value). The servo mainly depends on Pulsefor location. Basically, it can be understood that the servo motor receives an impulse and rotates the angle corresponding to the impulse to realize displacement. Because the servo motor itself has the function of sending out pulses, the servo motor rotates every time at an angle, and a corresponding number of pulses will be sent out. In this way, the pulses received by the servo motor form a response, or a closed loop. In this way, the system will know how many pulses are sent to the servo motor and how many pulses are received. In this way, it is possible to precisely control the rotation of the motor, thereby achieving precise positioning.

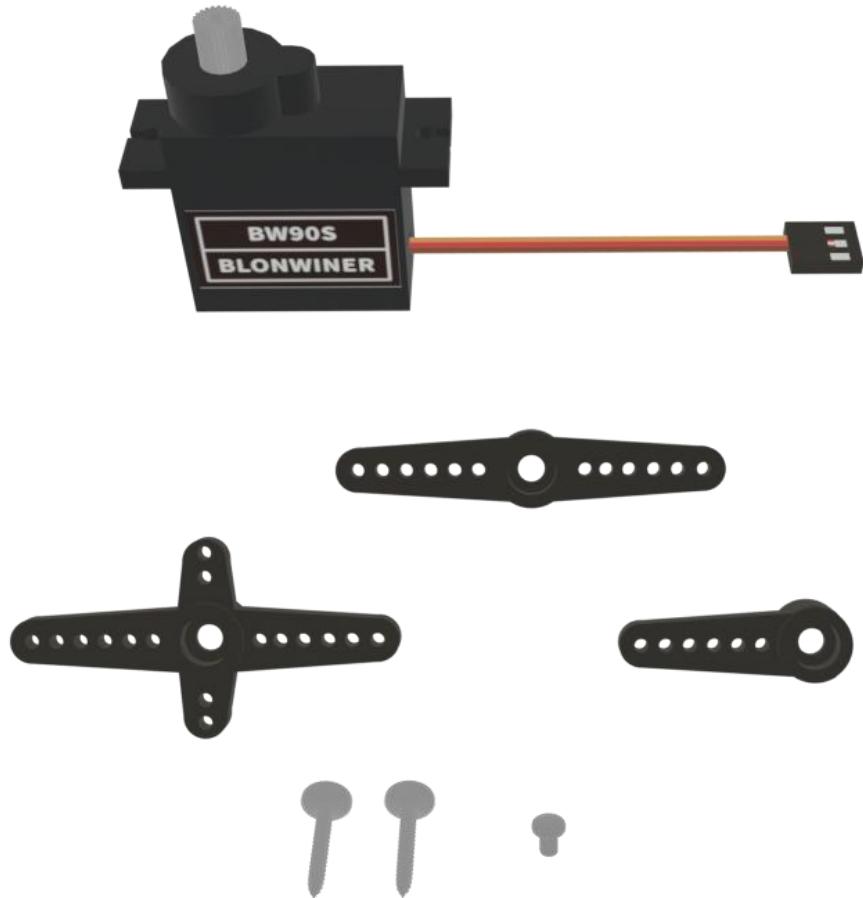
BLONWINER Arduino Arm Board sends a PWM signal to a servomotor, which is then processed by an IC on the circuit board to calculate the rotation direction of the drive motor, which is then transmitted through a reduction gear to the swing arm. At

the same time, the position detector returns a position signal to determine whether the set position has been reached or not.



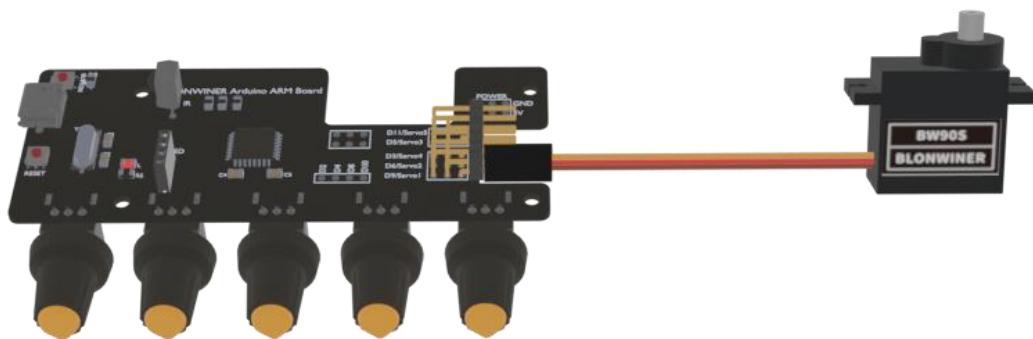
2.2.3 The principle of write () function

In the program, we use the write() function to control the rotation of the servo. For standard servos, the write() function will rotate the servo axis to the corresponding angular position. For the continuous rotation type of servo, the write() function can set the rotation speed of the servo (0 indicates that the servo rotates at full speed in one direction, 180 indicates that the servo rotates at full speed in another direction, and 90 indicates that the servo is stationary). The servo which is used this time is a standard servo.



2.3Wiring diagram (Circuit diagram)

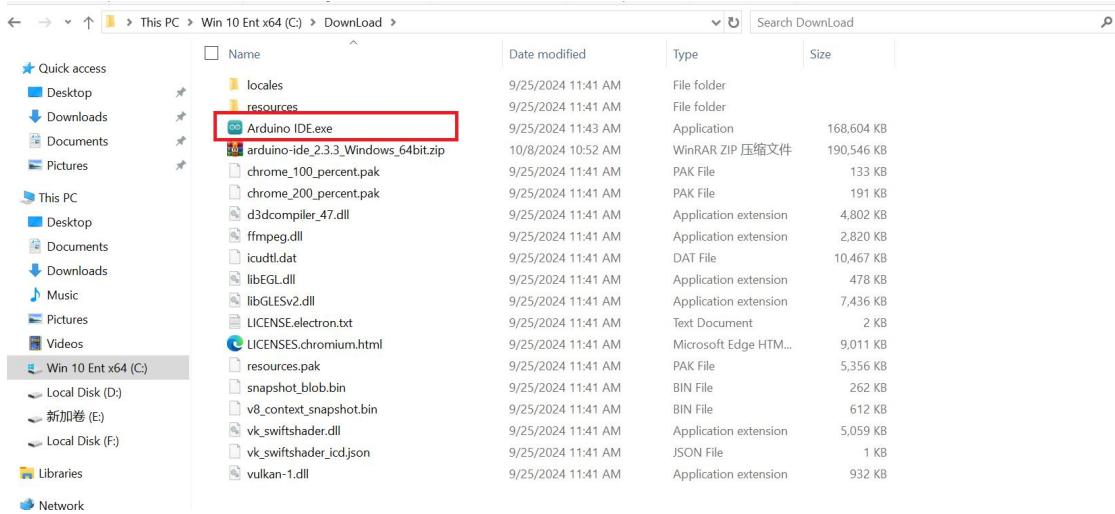
Connect Servo to the servo port on the BLONWINER Arduino Arm Board, as shown below:



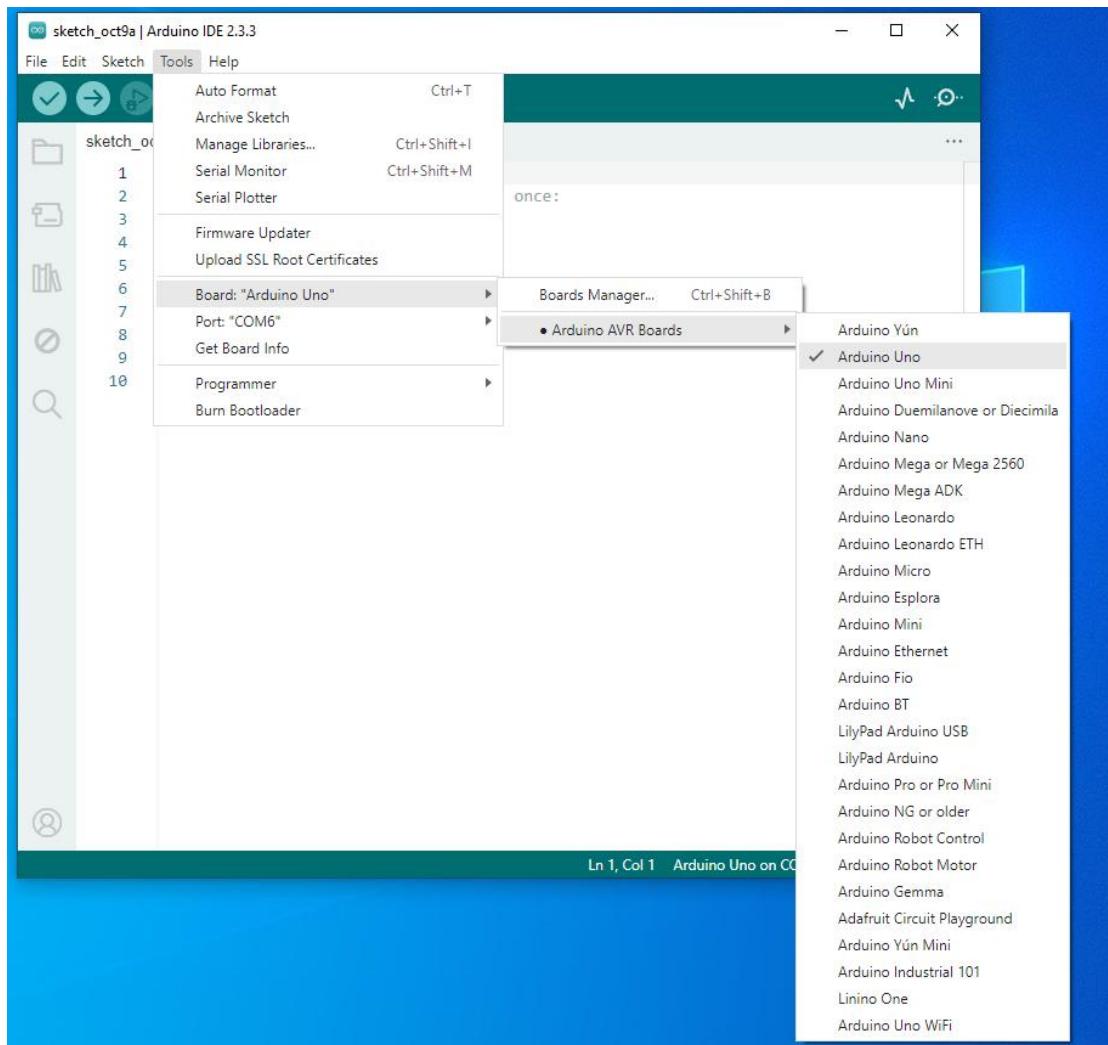
2.4 How to control Servo

2.4.1Compile and run the code program of this course

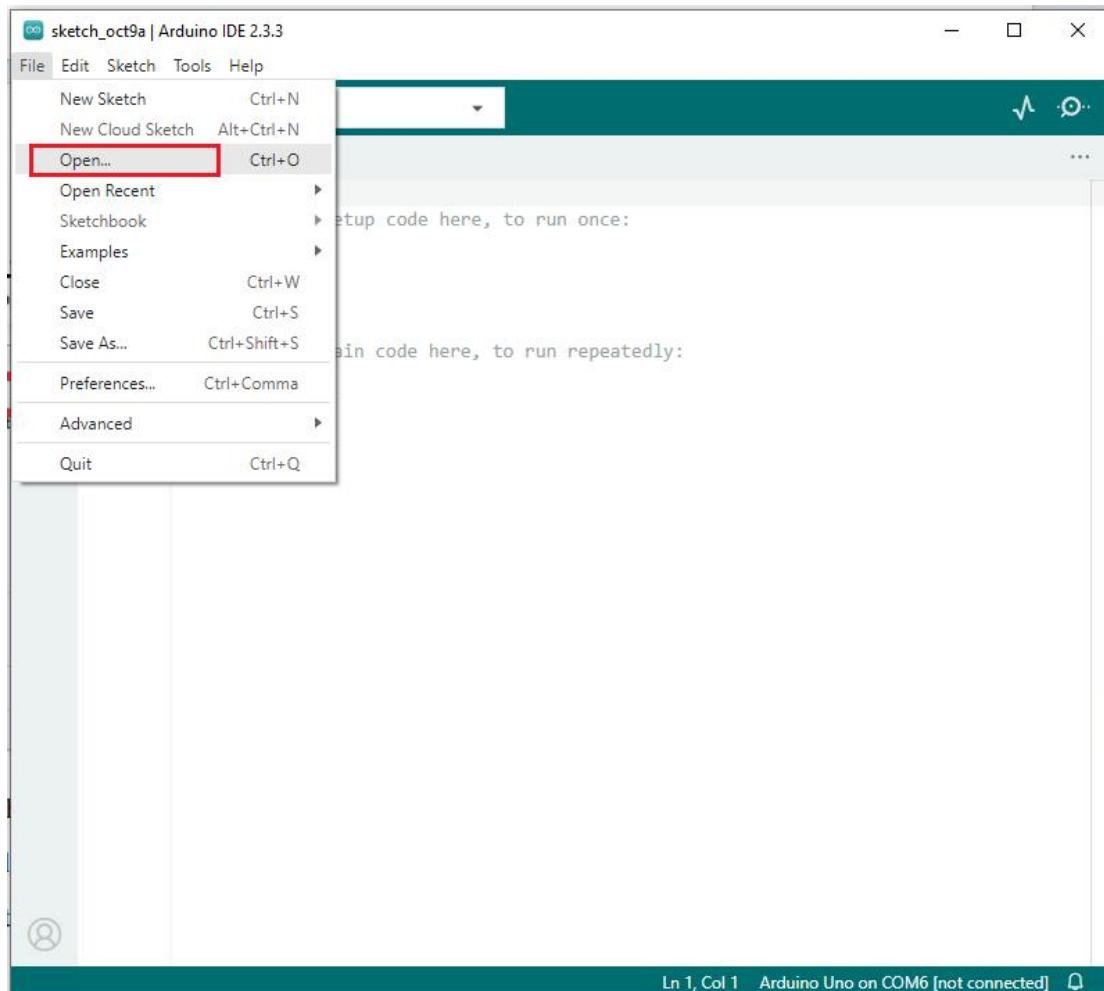
1. Open the Arduino IDE software, as shown below:



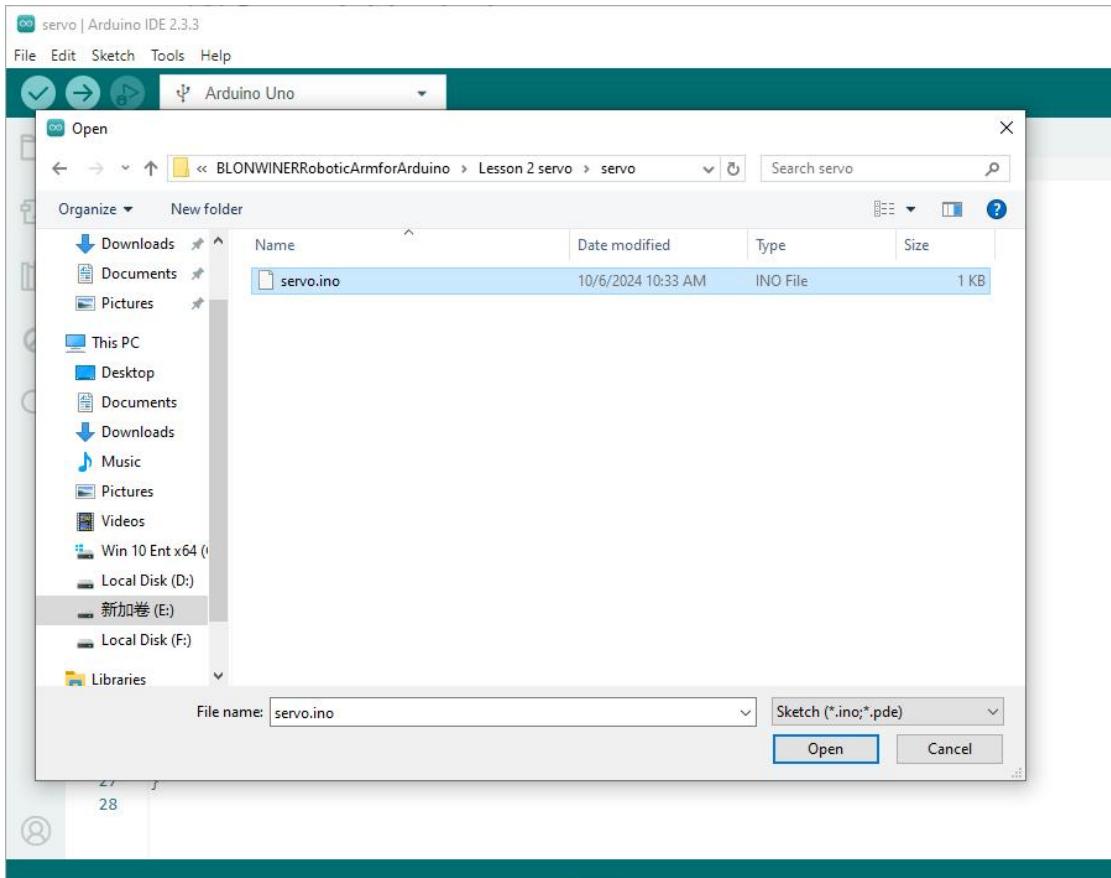
2. In the Tools toolbar, find Board and select Arduino Uno, as shown below:



3. Click Open in the File drop-down menu:



4. Find the folder BLOWINERRoboticArmforArduino that we provide to the user. Enter the Lesson 2 servo directory. Select servo.ino. This file is the code program we need in this course. Then click Open.



5. After opening, click  to upload the code program to the Arduino UNO. If there is no error warning in the console below, it means that the Upload is successful.

The screenshot shows the Arduino IDE with the code editor open. The file "servo.ino" is displayed, containing the following code:

```
1 //*****
2 // File name: servo.ino
3 // Description: The servo motor circulates to 180 degrees, 90 degrees
4 // | | | | | | | Degrees, 0 degrees,
5 // E-mail: blonwiner@outlook.com
6 // Author: blonwiner
7 // Date: 2024/10/06
8 *****/
9 #include <Servo.h>
10 Servo myservo;//create servo object to control a servo
11 void setup()
12 {
13     myservo.attach(9);//attaches the servo on pin 9 to servo object
14     myservo.write(0);//back to 0 degrees
15     delay(1000);//wait for a second
16 }
17 void loop()
18 {
```

The serial monitor at the bottom shows the following output:

```
Sketch uses 2060 bytes (6%) of program storage space. Maximum is 32256 bytes.
Global variables use 50 bytes (2%) of dynamic memory, leaving 1998 bytes for local variables. Maximum is 2048 bytes.
```

The status bar at the bottom right indicates "Ln 1, Col 1 - Arduino Uno on COM3 [not connected]".

6. After successfully running the program, you will observe the movement of the servo.

2.4.2 Learning the code program of this lesson

Create servo object to control a servo.

```
10 Servo myservo;//create servo object to control a servo
```

In the setup() function, attach the servo on pin 9 to servo object; back to 0 degrees; wait for a second.

```
11 void setup()
12 {
13     myservo.attach(9);//attaches the servo on pin 9 to servo object
14     myservo.write(0);//back to 0 degrees
15     delay(1000);//wait for a second
16 }
```

In the loop() function, respectively control Servo to turn to different angles.

```
17 void loop()
18 {
19     myservo.write(180);//goes to 15 degrees
20     delay(2000);//wait for a second
21
22     myservo.write(90);//goes to 30 degrees
23     delay(2000);//wait for a second.33
24
25     myservo.write(0);//goes to 45 degrees
26     delay(2000);//wait for a second.33
27 }
```

Lesson 3 Displaying Text on the OLED Screen

In this lesson, we will learn how to display text on the OLED screen.

3.1 Components used in this course

Components	Quantity	Picture
BLONWINER Arduino Arm Board	1	 A black printed circuit board (PCB) labeled "BLONWINER Arduino ARM Board". It features a central microcontroller chip, several component pads, and four black female header pins at the bottom for connection to an external breadboard or other components.
Micro USB Cable	1	 A standard black Micro USB cable with a white strain relief band around its middle section.
OLED screen	1	 An OLED screen module with a dark gray rectangular display panel. The word "BLONWINER" is displayed in white capital letters on the screen. Above the display, there are four small circular pads labeled "GND", "VDD", "SCK", and "SDA". Below the display, there is a small orange ribbon cable connector.

3.2 Introduction of OLED Screen

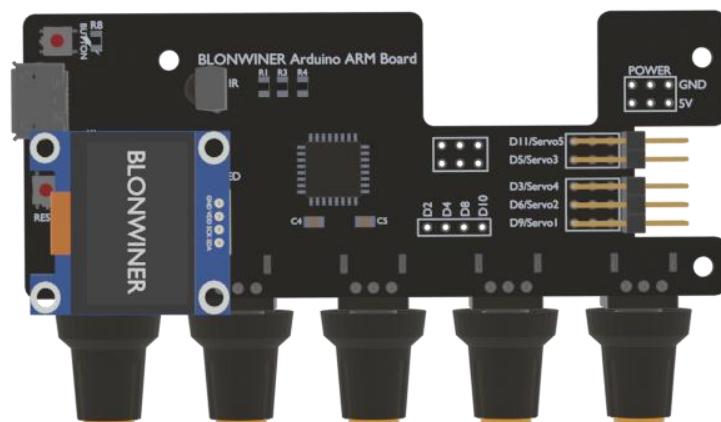
OLED (Organic Light-Emitting Diode), also known as organic electric laser display, organic light emitting semiconductor (Organic Electroluminescence Display, OLED). OLED is a kind of current-type organic light-emitting device, which produces light by the injection and recombination of carriers, and the luminous intensity is proportional to the injected current. The robot uses an OLED screen to display the expressions or some parameters of the robot. OLED Screen is a commonly used module on robot products. Due to the black non-luminous feature of OLED Screen, this type of screen has extremely high contrast. Even if the ambient light is strong, you can see the information on the OLED Screen clearly, and the power consumption is relatively low.

When using the OLED Screen, you need to connect it to the OLED interface on the BLONWINER Arduino Arm Board.



3.3 Wiring diagram (Circuit diagram)

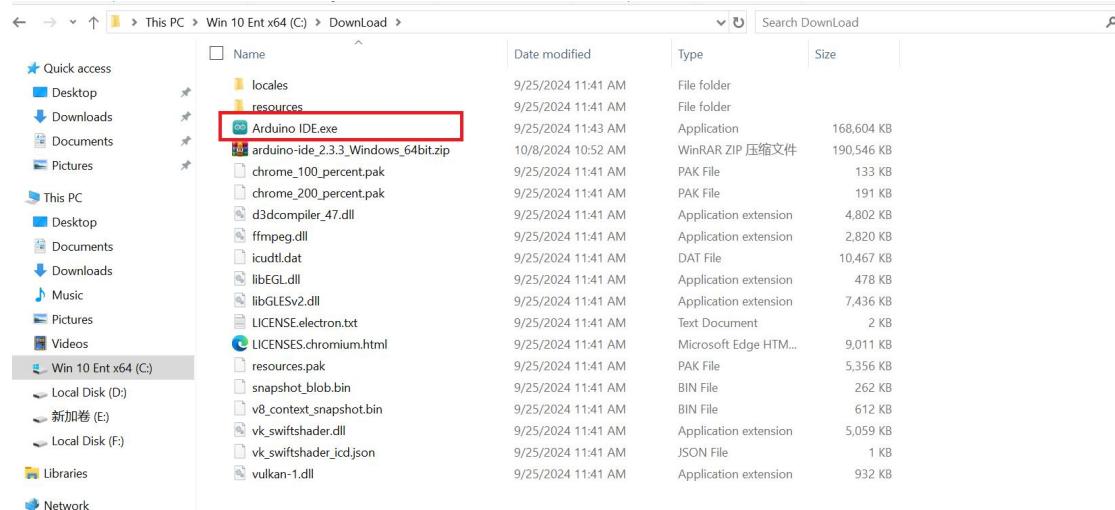
You need to connect it to the OLED interface on the BLONWINER Arduino Arm Board. As shown below:



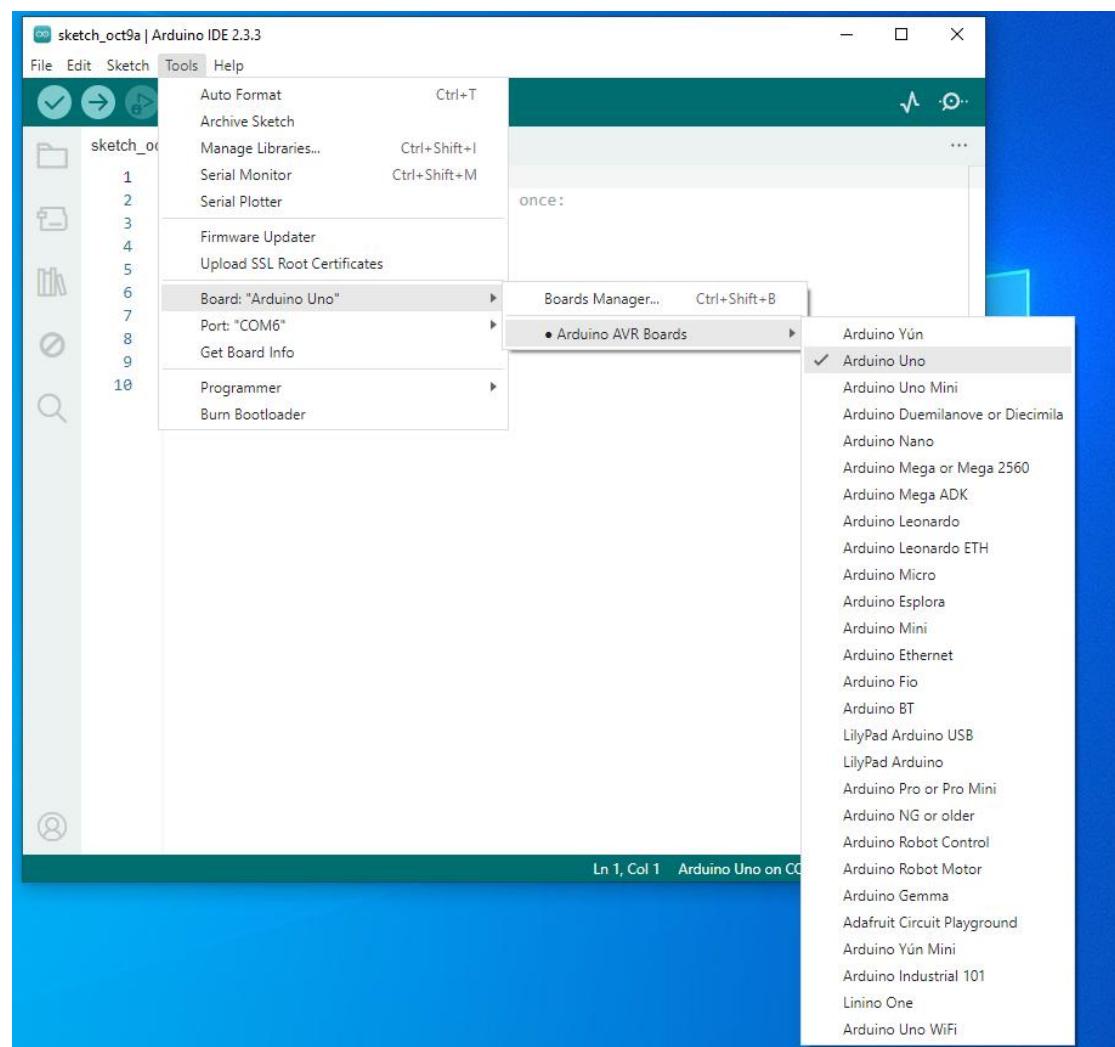
3.4 How to display text on the OLED screen

3.4.1 Compile and run the code program of this course

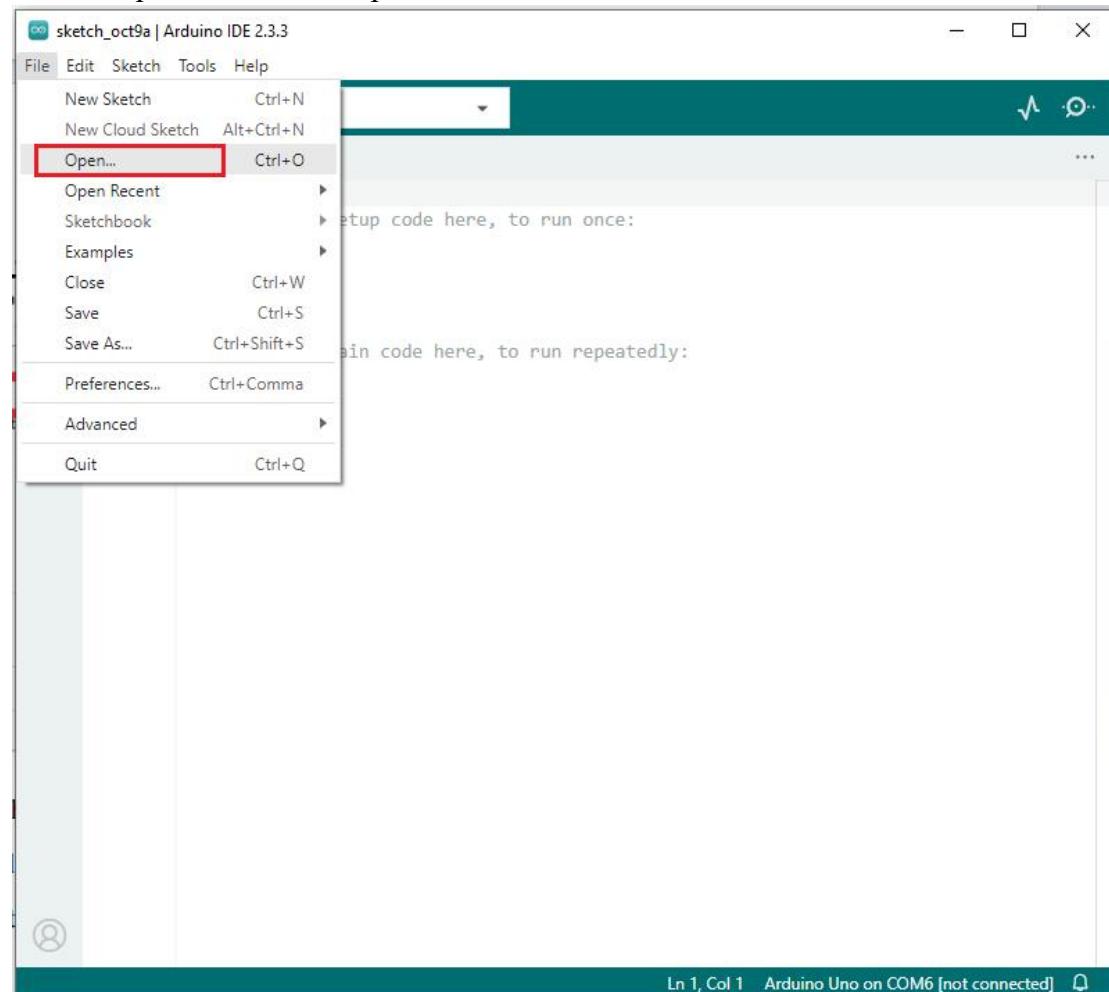
1. Open the Arduino IDE software, as shown below:



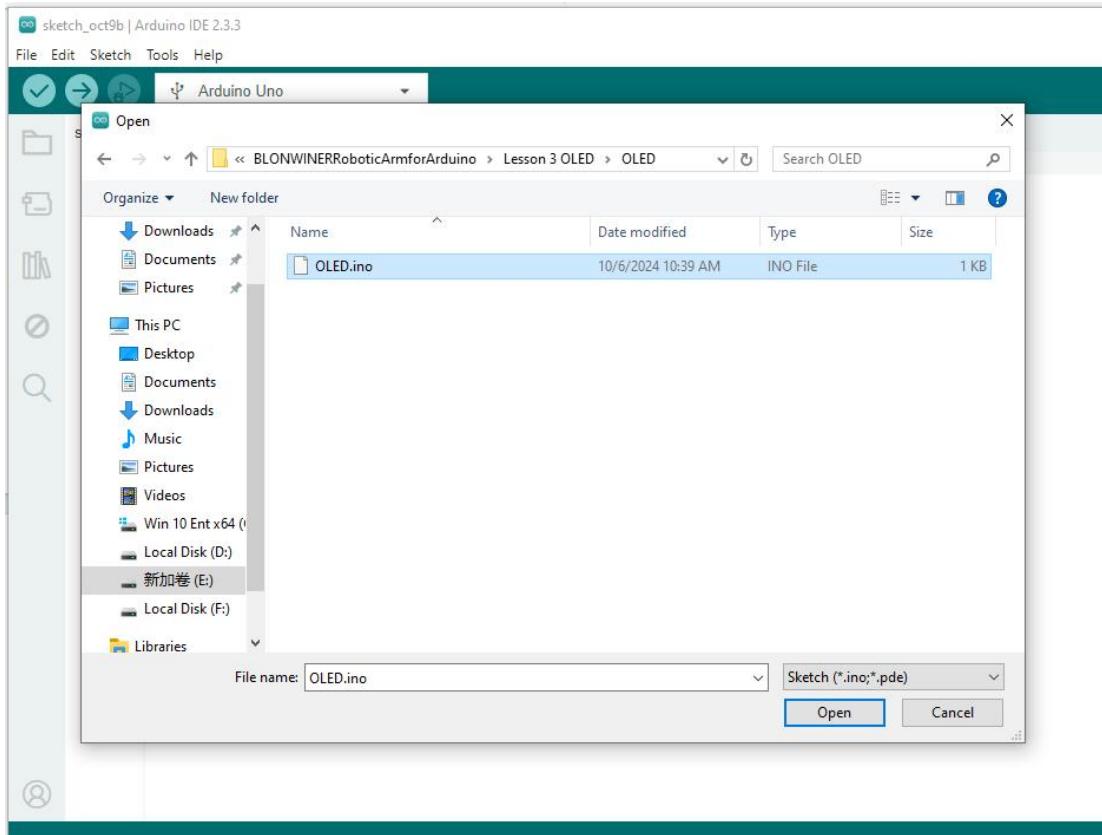
2. In the Tools toolbar, find Board and select Arduino Uno, as shown below:



3.Click Open in the File drop-down menu:



4.Find the folder BLOWINERRoboticArmforArduino that we provide to the user. Enter the Lesson 3 OLED directory. Select OLED.ino. This file is the code program we need in this course. Then click Open.



5. After opening, click to upload the code program to the Arduino UNO. If there is no error warning in the console below, it means that the Upload is successful.

The screenshot shows the Arduino IDE with the sketch 'OLED.ino' open. The code is as follows:

```
1 // ****
2 File name:OLED
3 Description:OLED Display images.
4 E-mail:blonwiner@outlook.com
5 Author:blonwiner
6 Date:2024/10/6
7 ****
8 #include <Wire.h>
9 #include <Adafruit_GFX.h>
10 #include <Adafruit_SSD1306.h>
11 #define OLED_RESET 4
12 Adafruit_SSD1306 display(128, 64, &Wire, OLED_RESET);
13
14
15 void setup() {
16     display.begin(SSD1306_SWITCHCAPVCC, 0x3C);
17     display.setTextColor(WHITE); // Sets the font display color
18     display.clearDisplay(); //cls
```

The 'Output' tab at the bottom shows the following message:

```
Sketch uses 13026 bytes (40%) of program storage space. Maximum is 32256 bytes.
Global variables use 350 bytes (17%) of dynamic memory, leaving 1698 bytes for local variables. Maximum is 2048 bytes.
```

At the very bottom, the status bar indicates: 'Ln 1, Col 1 Arduino Uno on COM3 [not connected] 4 1'.

6. After successfully running the program, you will observe that text will be displayed on the OLED screen.

3.4.2 Learning the code program of this lesson

After the above practical operation, you must be very curious to know how we use C language to program on the BLONWINER Arduino Arm Board to display text on the OLED screen. Below we will introduce how the main code program is implemented.

First, in the setup() function, set the display color of the font to white.

```
15 void setup() {  
16     display.begin(SSD1306_SWITCHCAPVCC, 0x3C);  
17     display.setTextColor(WHITE);//Sets the font display color  
18     display.clearDisplay();//cls  
19 }
```

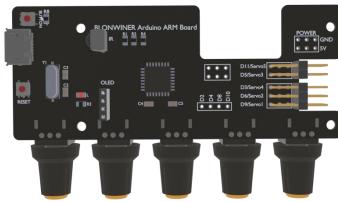
In the loop() function, set the display font size with setTextSize(1); setCursor(30,30) sets the position of the text displayed on the OLED screen, and print("Hello") prints out the text information that needs to be displayed.

```
21 void loop() {  
22     //Set the font size  
23     display.setTextSize(2);  
24     //Set the display location  
25     display.setCursor(30,30);  
26     //String displayed  
27     display.print("Hello");  
28     //Began to show  
29     display.display();  
30 }
```

Lesson 4 Saving Data with EEPROM

In this lesson, we will learn how to save data with EEPROM.

4.1 Components used in this course

Components	Quantity	Picture
BLONWINER Arduino Arm Board	1	 A black printed circuit board (PCB) labeled "BLONWINER Arduino ARM Board". It features a central microcontroller chip, several component pads, and four black cylindrical connectors at the bottom. On the right side, there are two black ribbon cables labeled "DServo1" and "DServo4" with multiple pins extending from them.
Micro USB Cable	1	 A black Micro USB cable coiled and shown against a white background. It has a standard Micro USB connector on one end and a standard USB-A connector on the other.

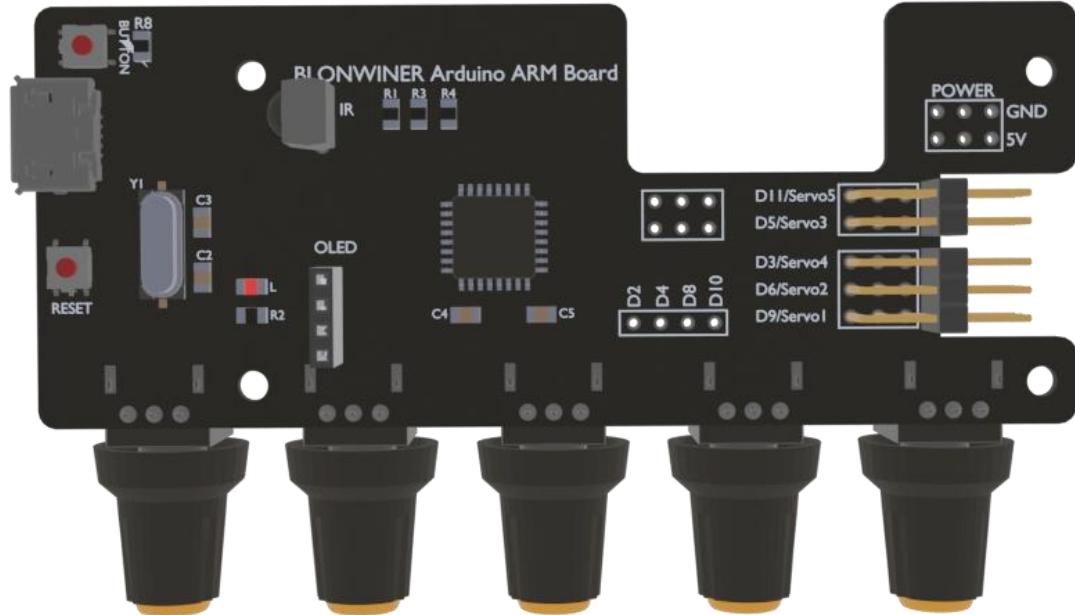
4.2 About EEPROM

EEPROM (Electrically Erasable Programmable Read Only Memory) refers to electrically erasable programmable read only memory. It is a memory chip that does not lose data after power failure. EEPROM can erase existing information on a computer or special equipment and reprogram it. It is generally used in plug and play. The BLONWINER Arduino Arm Board has its own EEPROM, and its memory size is 1K.

Arduino IDE comes with EEPROM usage method. The Arduino library has prepared EEPROM library for us. You can directly call EEPROM.h in the code when you are using the EEPROM library. And then use the write() and read() methods to operate the EEPROM.

4.3 Wiring diagram (Circuit diagram)

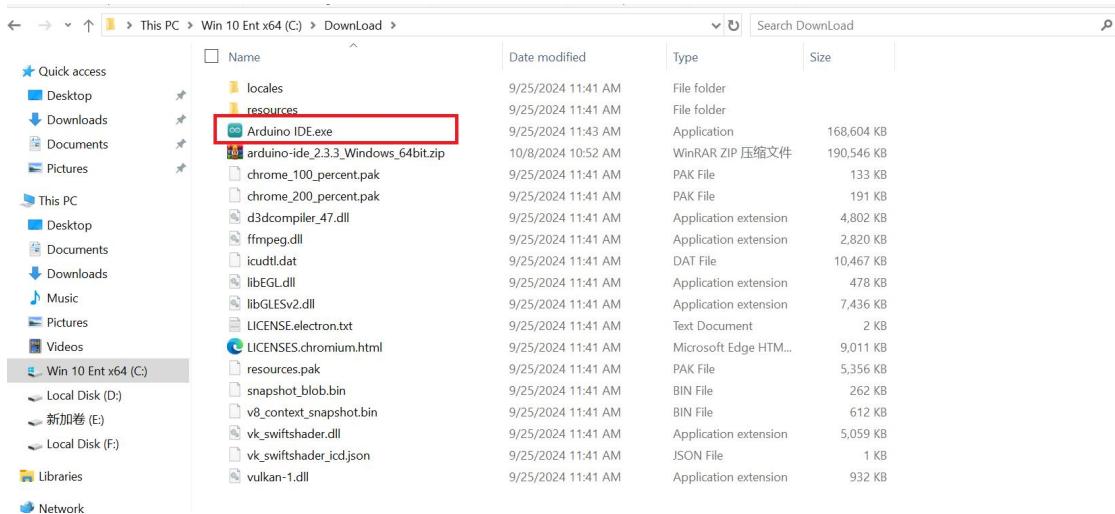
You need to connect it to the OLED interface on the BLONWINER Arduino Arm Board. As shown below:



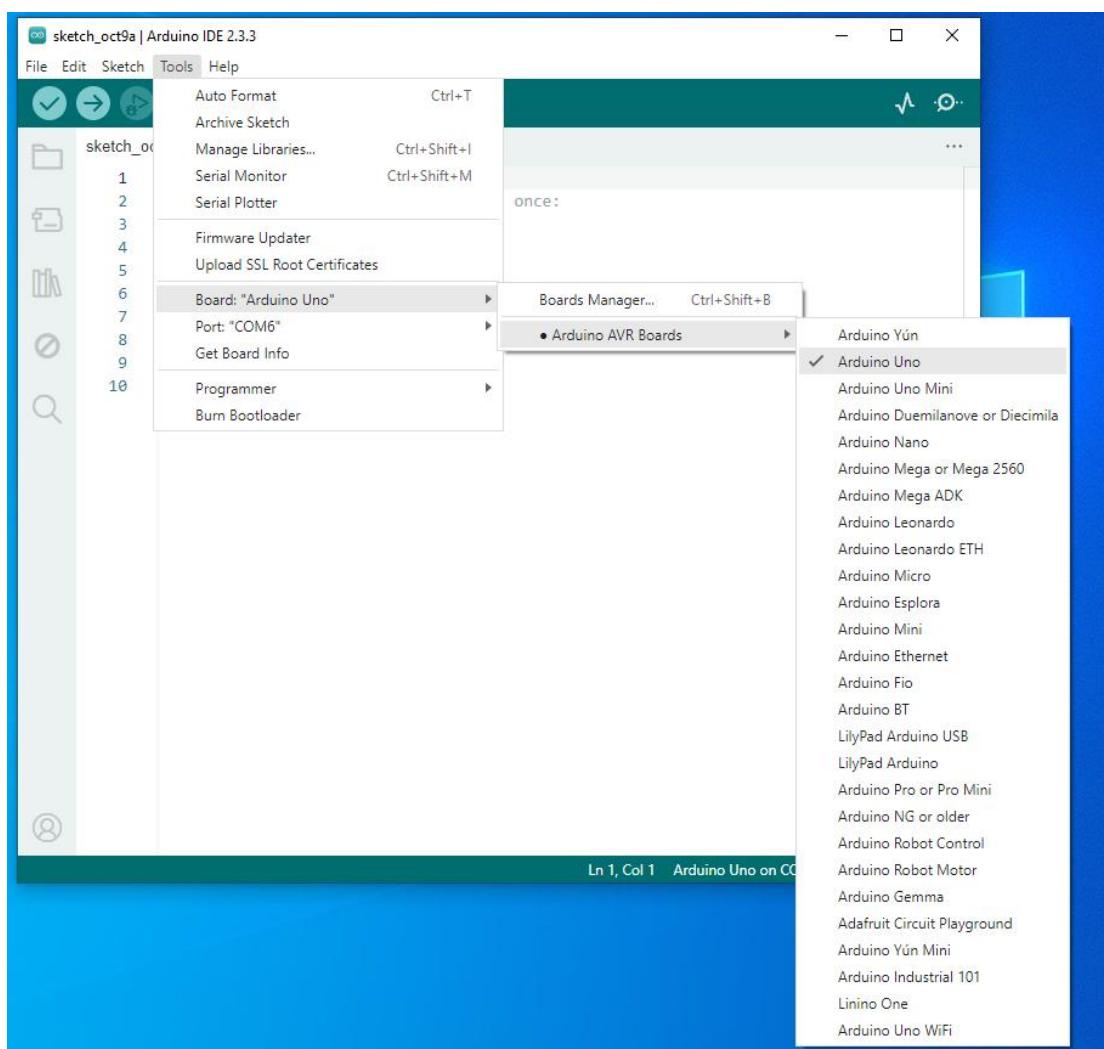
4.4 How to use EEPROM to save data

4.4.1 Compile and run the code program of this course

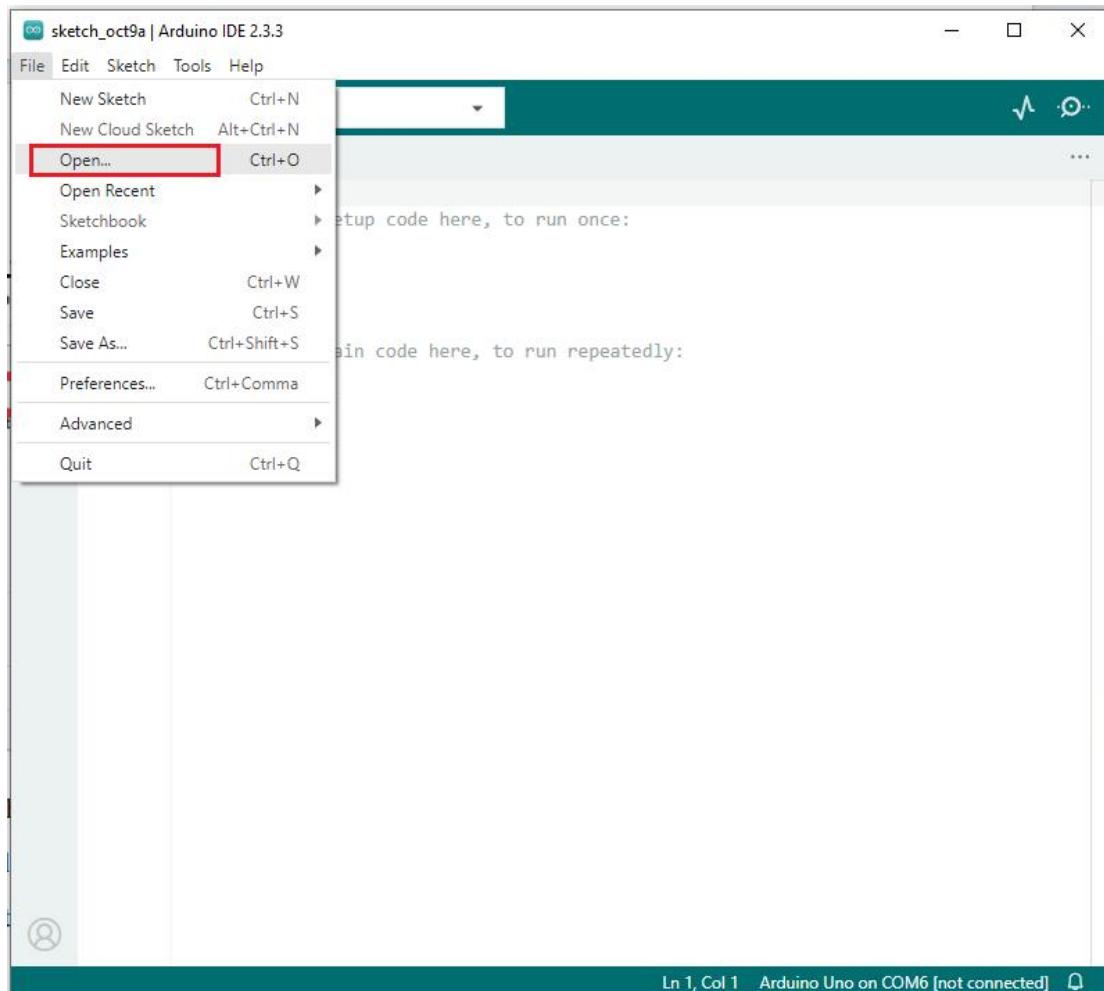
1. Open the Arduino IDE software, as shown below:



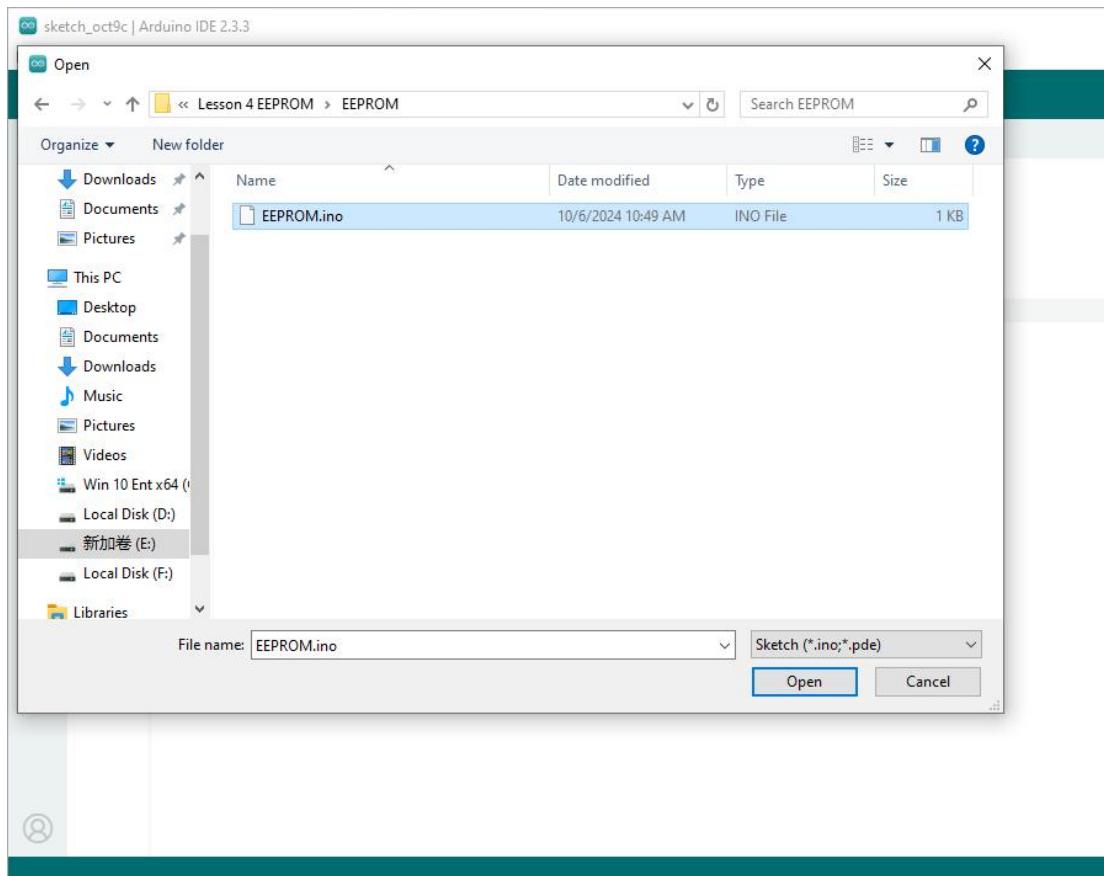
2. In the Tools toolbar, find Board and select Arduino Uno, as shown below:



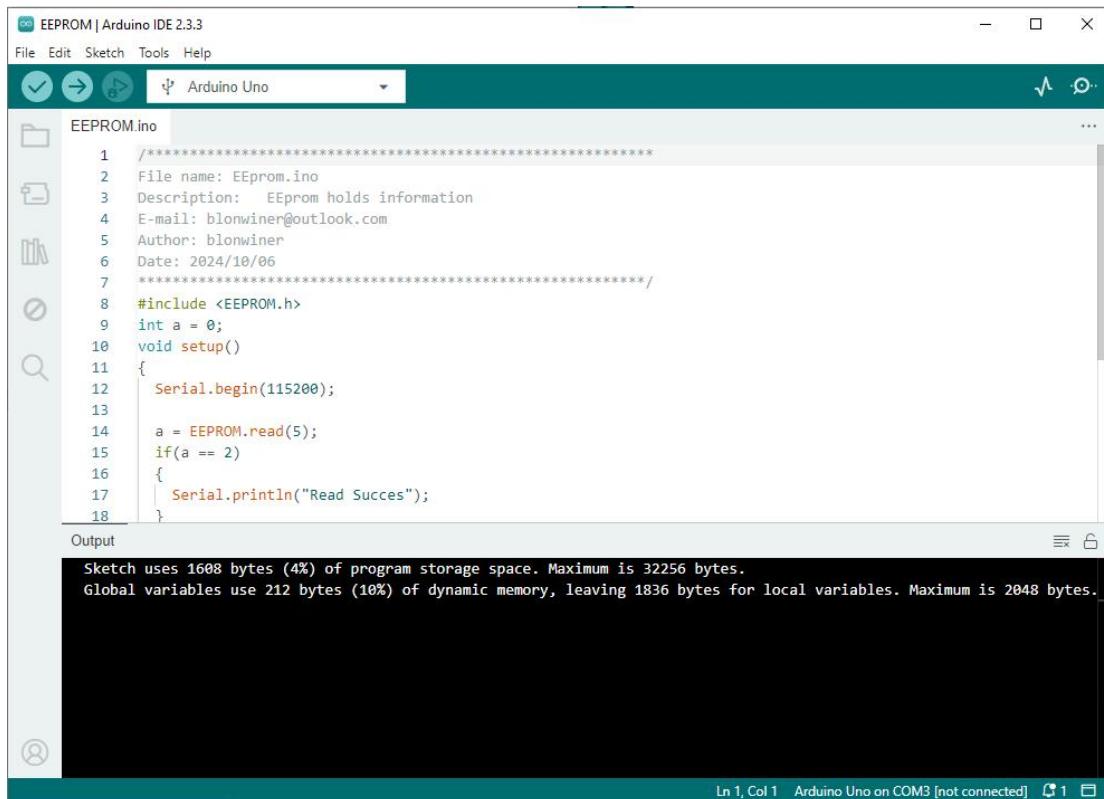
3. Click Open in the File drop-down menu:



4. Find the folder BLOWINERRoboticArmforArduino that we provide to the user. Enter the Lesson 4 EEPROM directory. Select EEPROM.ino. This file is the code program we need in this course. Then click Open.

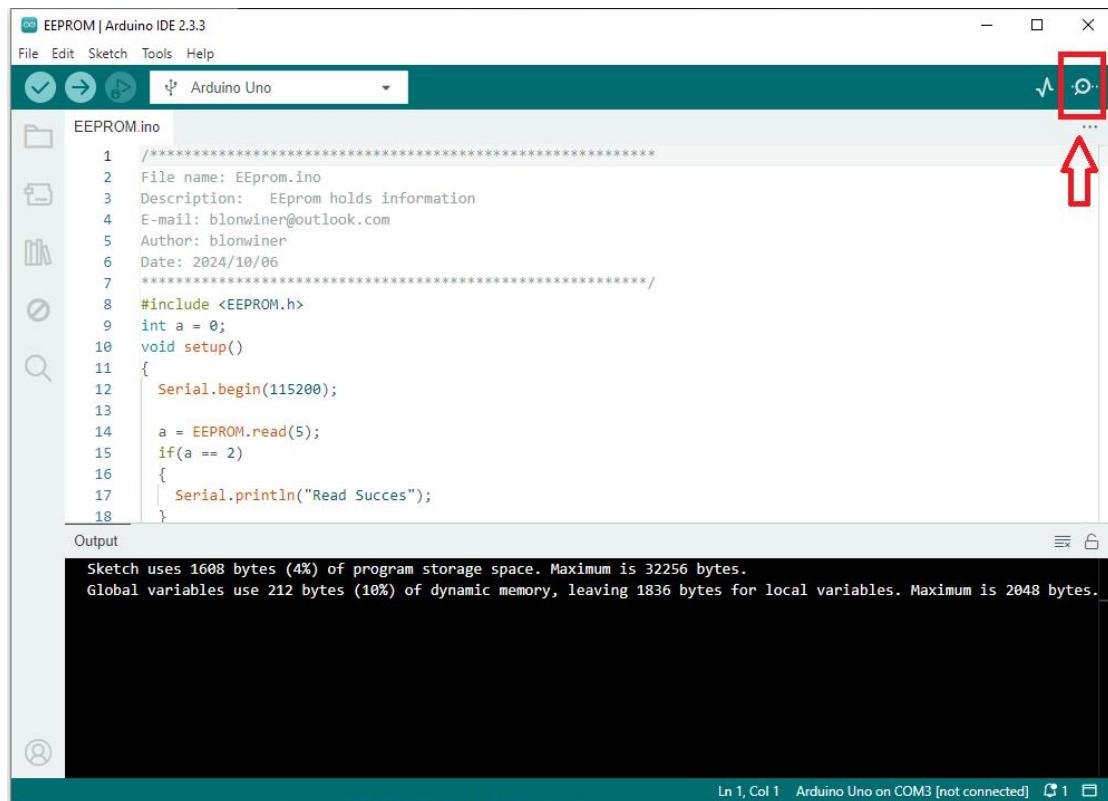


5. After opening, click to upload the code program to the Arduino UNO. If there is no error warning in the console below, it means that the Upload is successful.

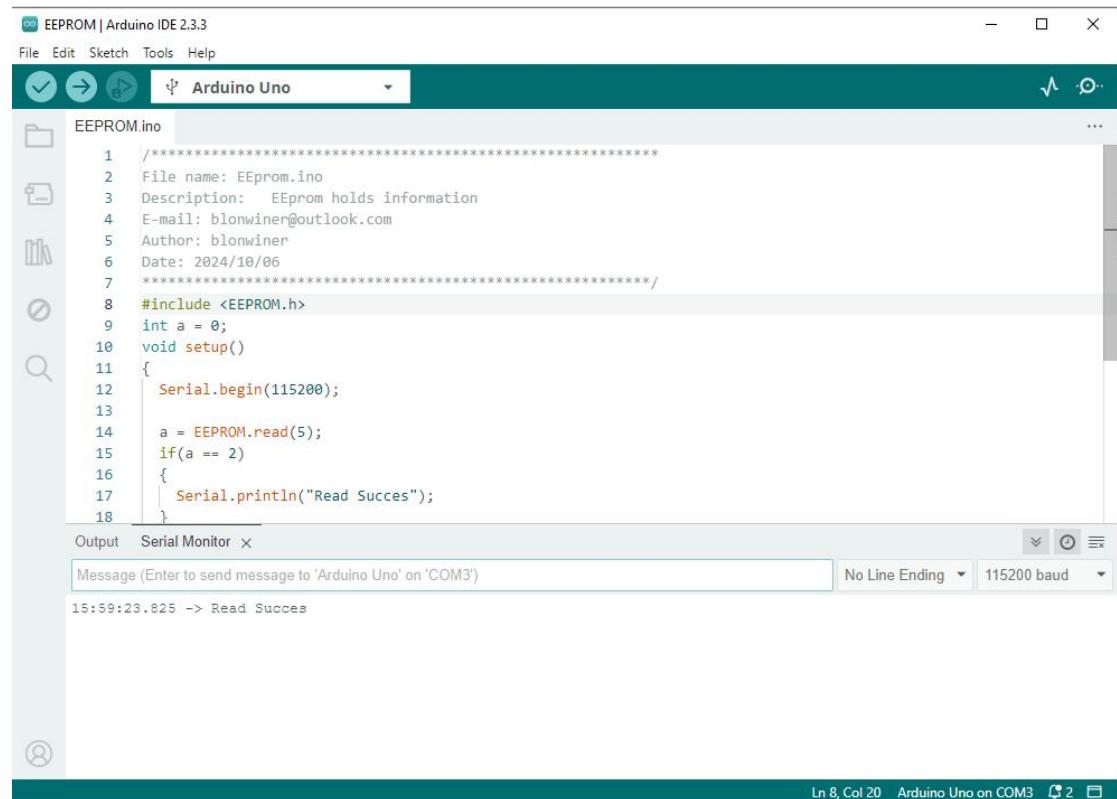


6. After successfully running the program, we need to observe the value of the

potentiometer by opening the serial monitor and click  , as shown in the figure below:



You will see the returned information in the serial monitor: Read Succes, indicating that the data has been saved successfully.



4.4.2 Learning the code program of this lesson

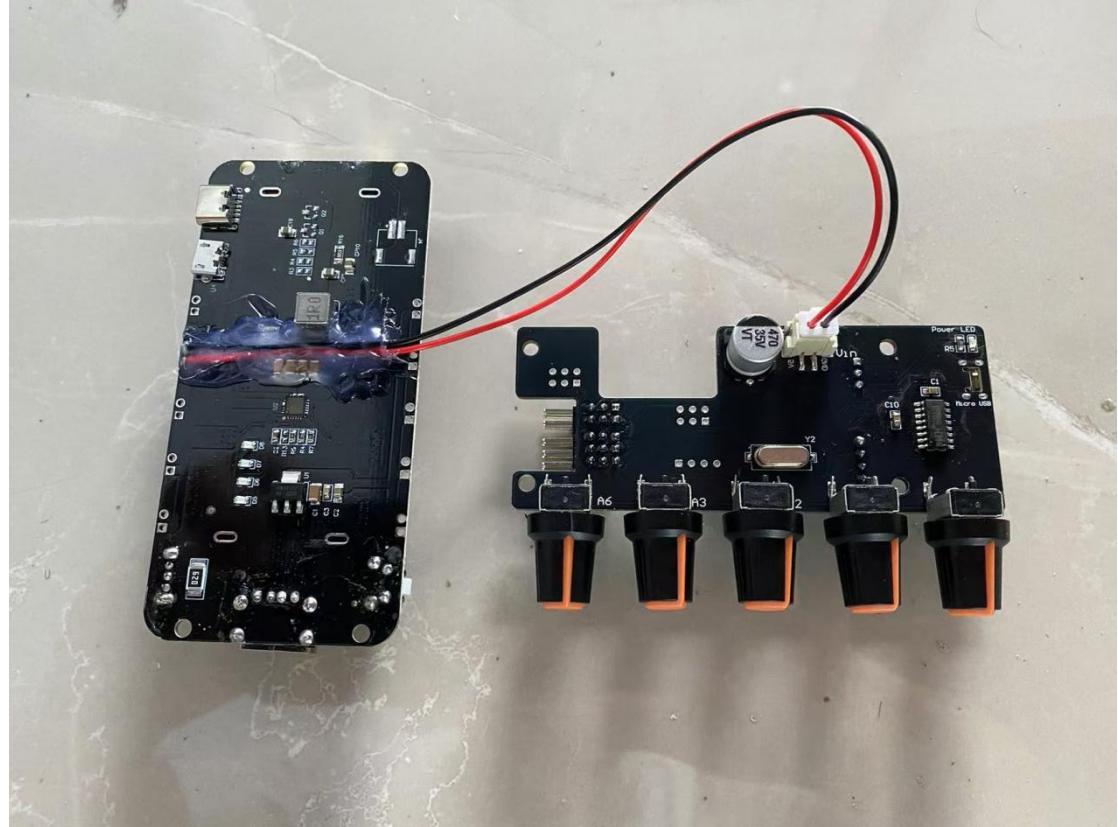
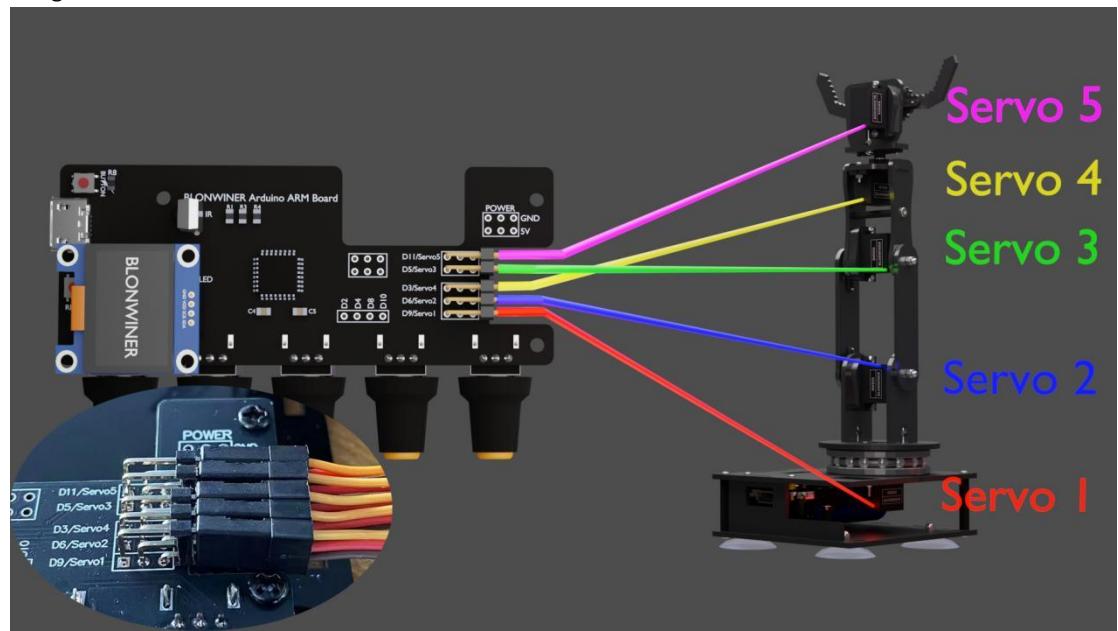
After the above practical operation, you must be very curious to know how we use C language to program on the BLONWINER Arduino Arm Board to save data with EEPROM. Below we will introduce how the main code program is implemented.

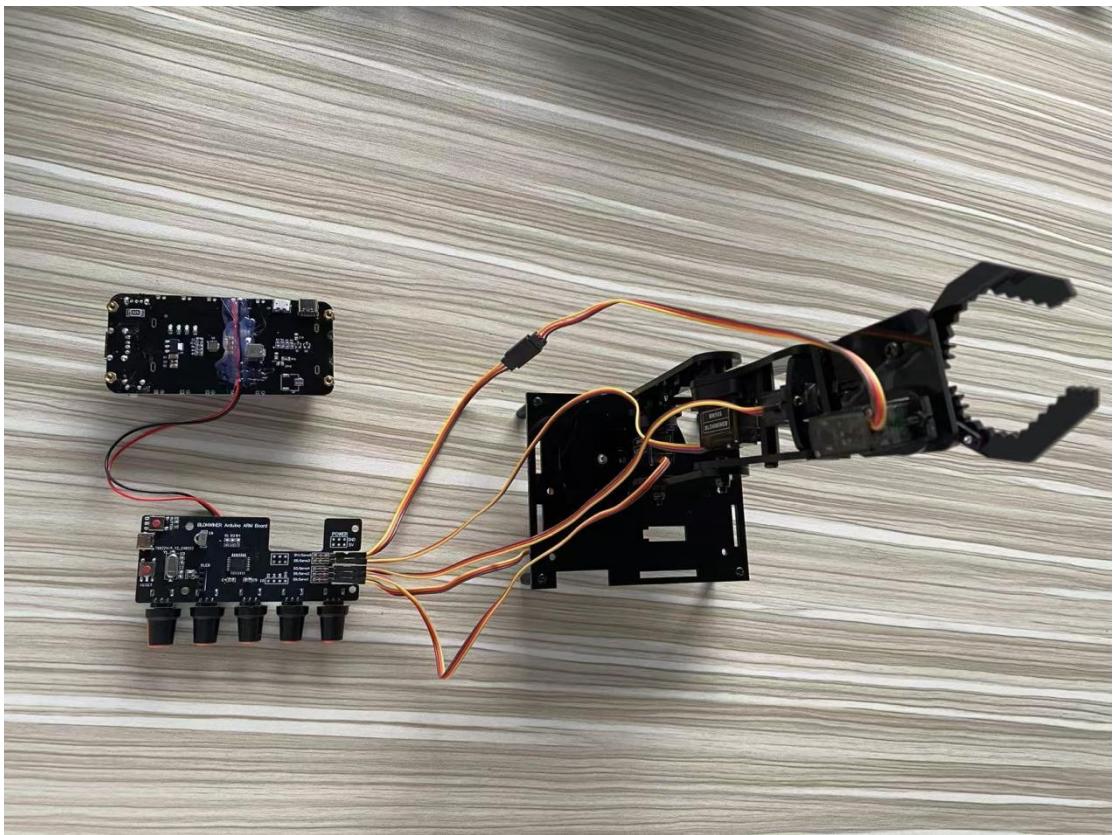
In the setup() function, first initialize the serial monitor, EEPROM.read(5) reads the data, and judges by if, if the read data is 2, then it is saved successfully.

```
10 void setup()
11 {
12     Serial.begin(115200);
13
14     a = EEPROM.read(5);
15     if(a == 2)
16     {
17         Serial.println("Read Succes");
18     }
19     else
20     {
21         Serial.println("Read Failed");
22         EEPROM.write(5,2);
23     }
24 }
25
26 }
```

blonwiner@outlook.com

Diagram of the robot circuit connection

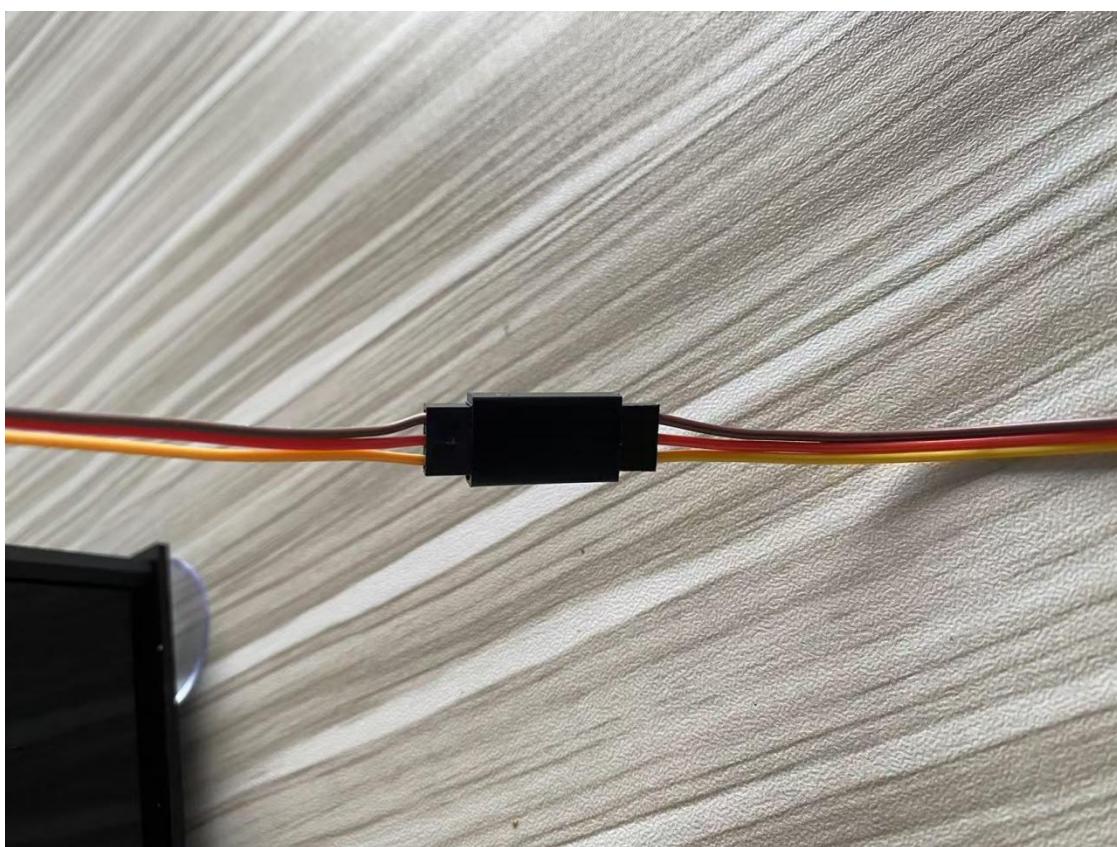
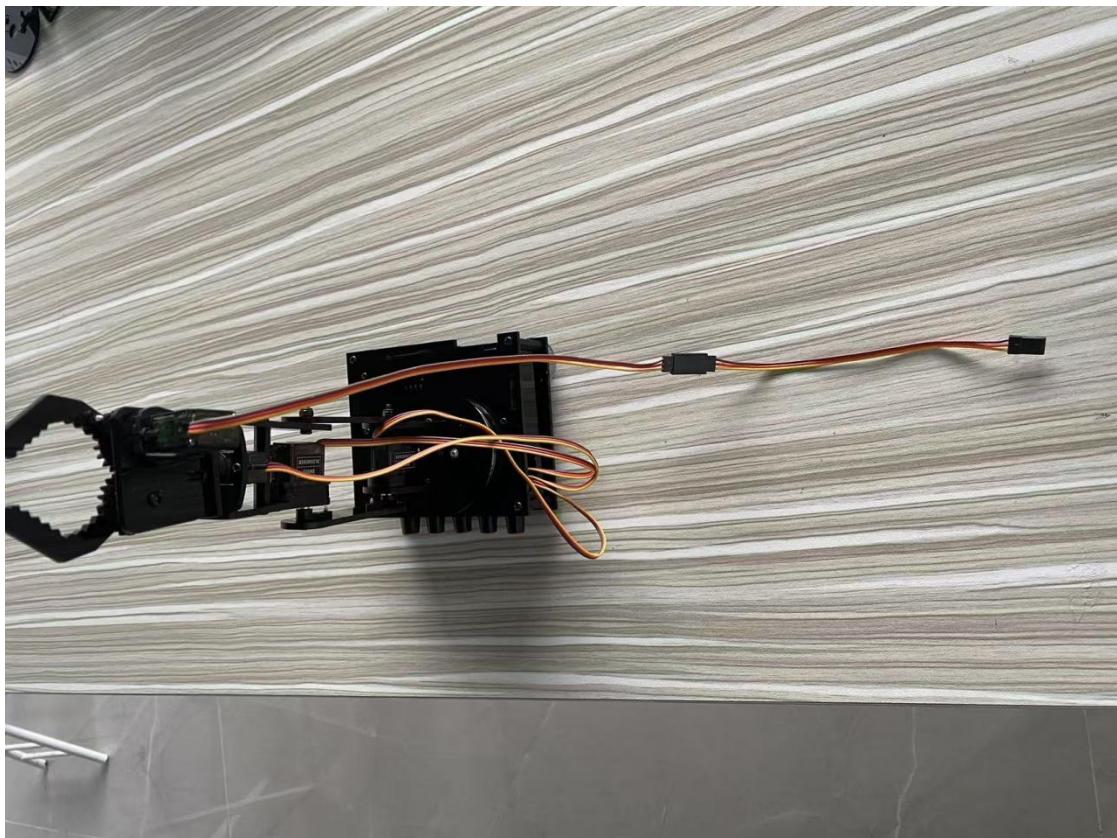




The servo extension cable is used on the servo 5.



blonwiner@outlook.com



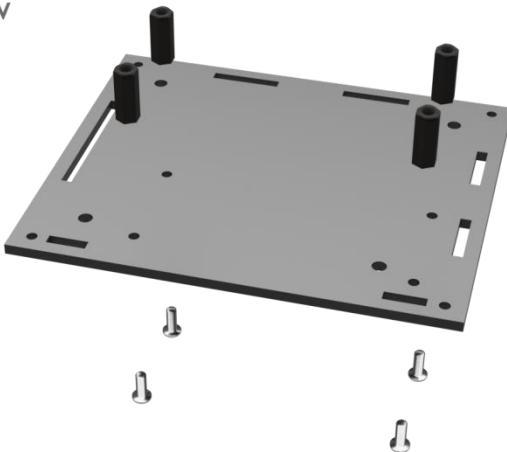
How to Assemble the Robotic Arm

Pedestal Assembly

1. Fix the four M3*14 nylon standoffs posts and four M3*8 screws on the A01.

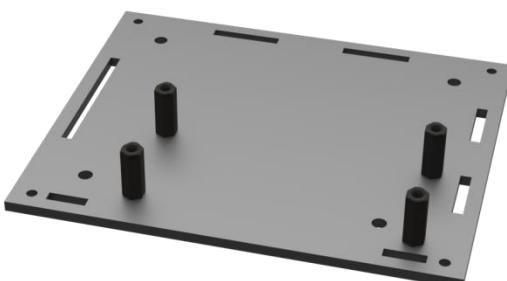
Assemble the following components

1PCS A.01 Acrylic plates
4PCS M3*14 Nylon Standoff
4PCS M3*8 Screw



Effect diagram after assembling

1PCS A.01 Acrylic plates
4PCS M3*14 Nylon Standoff
4PCS M3*8 Screw



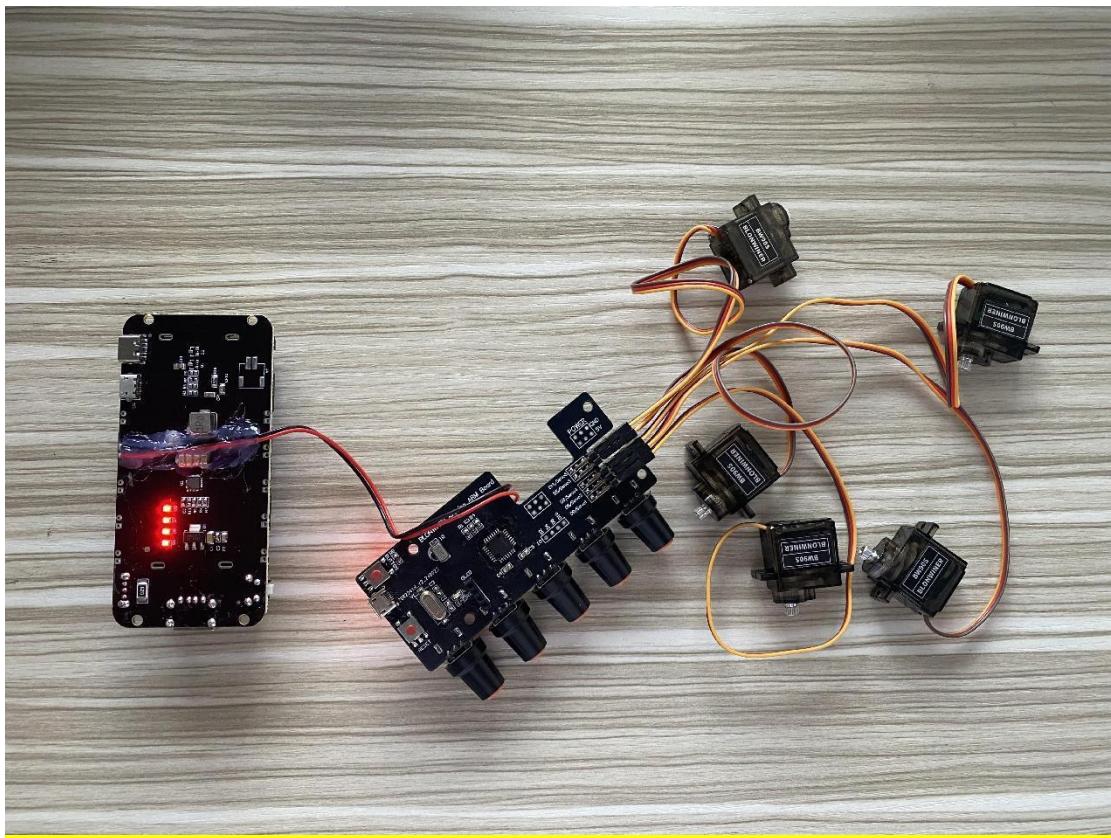
Note:

To assemble the robot, you need to download the code in Lesson 0

The servo initialization code of robotic arm assembly and let the

servos run to 90° .

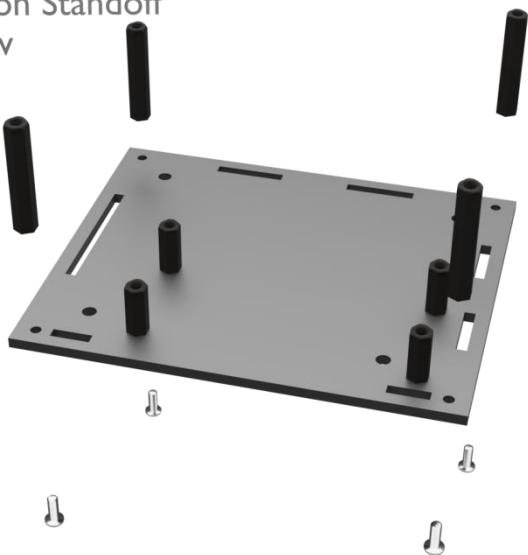
blonwiner@outlook.com



2.Fix the four M3*30 nylon standoffs posts and four M3*8 screws on the A01.

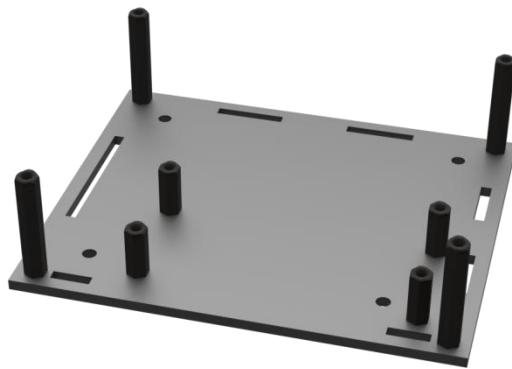
Assemble the following components

4PCS M3*30 Nylon Standoff
4PCS M3*8 Screw



Effect diagram after assembling

4PCS M3*30 Nylon Standoff
4PCS M3*8 Screw

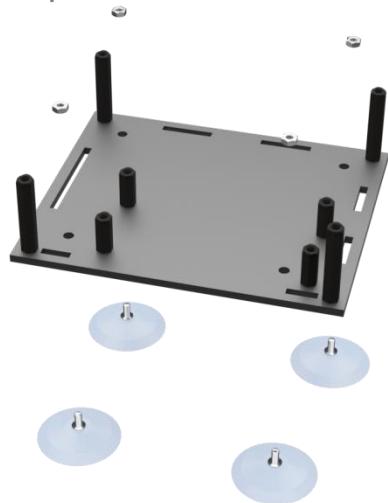


3.Fix four round suction cup on the four corners of A01.

Assemble the following components

4PCS Round Suction Cup Screws

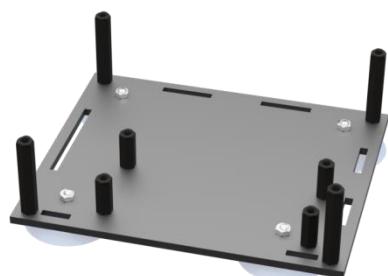
4PCS Round Suction Cup Nut



Effect diagram after assembling

4PCS Round Suction Cup Screws

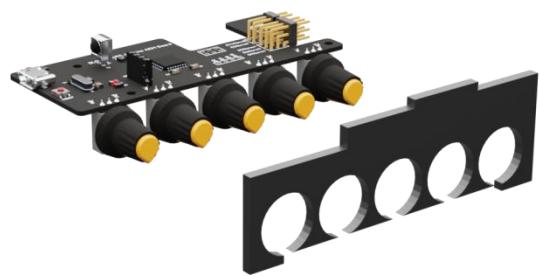
4PCS Round Suction Cup Nut



4.Fix BLONWINER Arduino ARM Board on A.06.

Assemble the following components

IPCS BLONWINER Arduino ARM Board
IPCS A.06 Acrylic plates



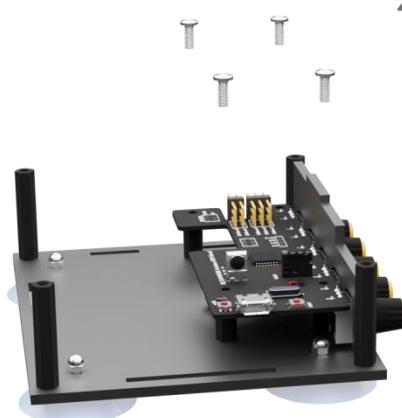
Effect diagram after assembling



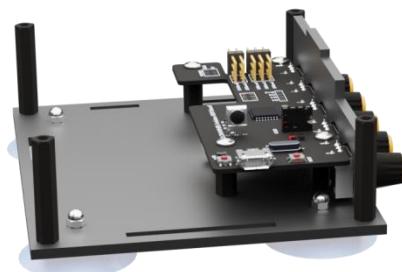
5.Fix BLONWINER Arduino ARM Board and A.06 on A.01.

Assemble the following components

4PCS M3*8 Screw



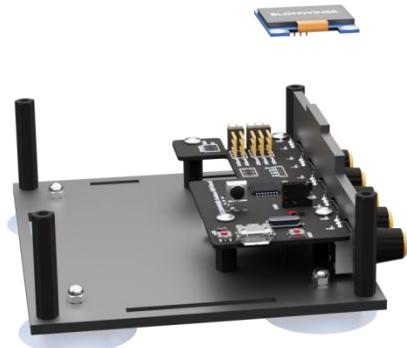
Effect diagram after assembling



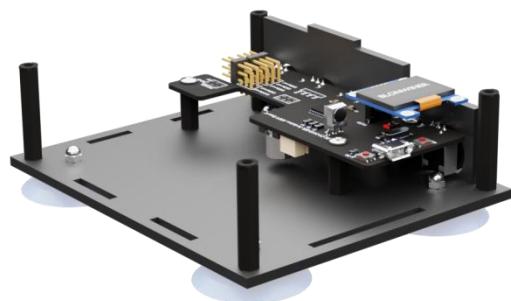
6.Fix OLED module on BLONWINER Arduino ARM Board.

Assemble the following components

IPCS OLED Module



Effect diagram after assembling



7.Fix four M2.5*4+6 copper standoff and four M2.5 nylon on power module.

Assemble the following components

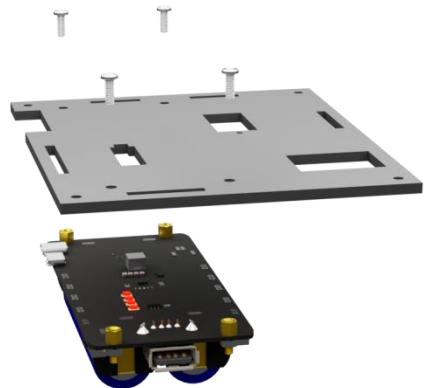


Effect diagram after assembling



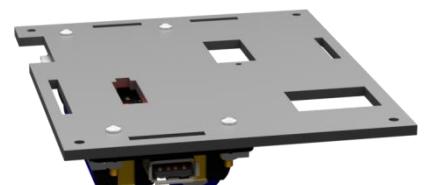
8.Fix four M2.5*8 screw and A.02 on power module.

Assemble the following components



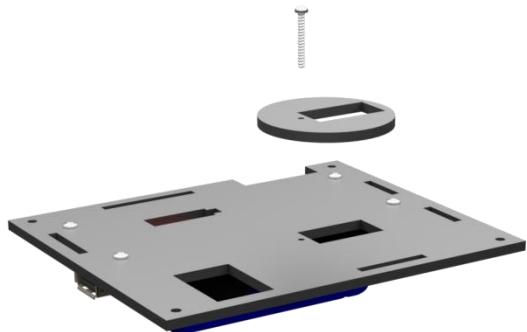
4PCS M2.5*8 Screw
1PCS Acrylic plate

Effect diagram after assembling



9.Fix M2*16 screw, M2 nut, B.01 and BLONWINER servo on A.02.

Assemble the following components



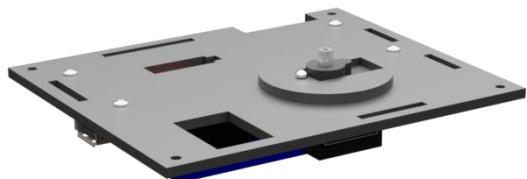
IPCS M2*16 Screw

IPCS M2 Nut

IPCS B.01 Acrylic plate

IPCS BLONWINER Servo

Effect diagram after assembling



10.Fix four M3*8 screw, A.03, A.04 and A.05 servo on A.02.

Assemble the following components



Effect diagram after assembling



11.Fix 51108 bearing on A.02.

Assemble the following components

IPCS 51108 Bearing



Effect diagram after assembling



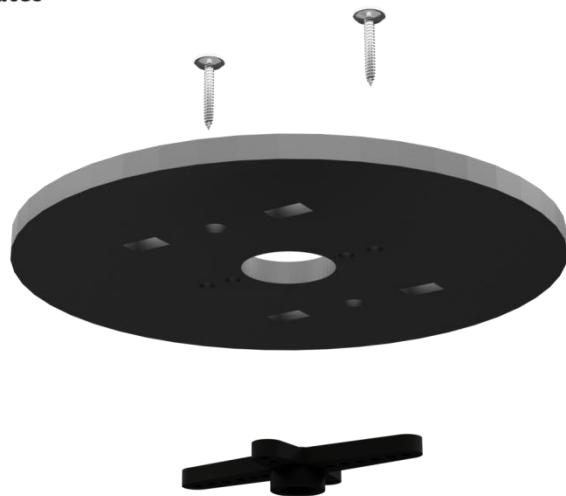
12.Fix two screw(fixing screw packaged with servo) and servo arm on B.02.

Assemble the following components

2PCS Fixing screw packaged with servo

1PCS B.02 Acrylic plates

1PCS Servo Arm



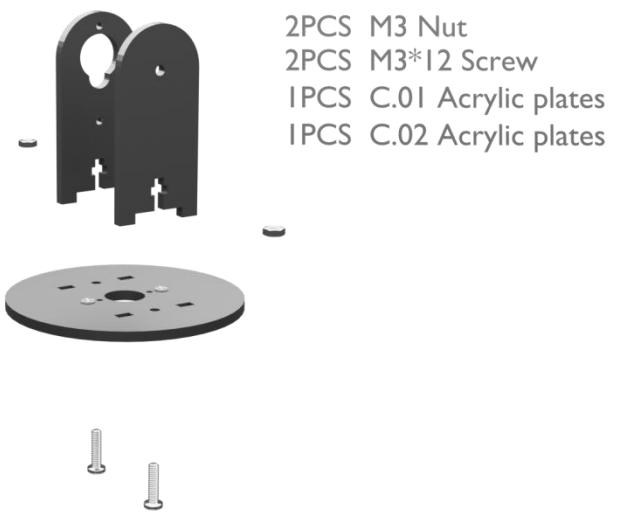
Effect diagram after assembling

Note: Do not tighten the screws(Self-tapping screw packaged with servo) here, leaving a large space between the servo rocker arm and B.02 acrylic plate.



13.Fix two M3*12 screw, two M3 nut, C.01 and C.02 on B.02.

Assemble the following components



Effect diagram after assembling



14. Fix the screw(fixing screw packaged with servo) on B.02.

Assemble the following components

IPCS Fixing screw packaged with servo



Effect diagram after assembling

Note: Please tighten this screw on the servo below.



Note: If the bearing(51108 Bearing) is very loose at this time, adjust these two screws to hold the bearing(51108 Bearing) in place.



15.Fix two M3*12 screw, two M3 lock nut, C.04 and C.06 on C.02.

Assemble the following components

1PCS C.06 Acrylic plates

1PCS C.04 Acrylic plates

2PCS M3 Lock Nut

2PCS M3*12 Screw



Effect diagram after assembling

Note:

The M3 Lock Nut cannot be tightened, otherwise it will cause the robot arm to fail to rotate and even burn the servo.



Note:

The M3 self-locking screw cannot be tightened, otherwise it will cause the mechanical arm to fail to rotate and even burn the servo.



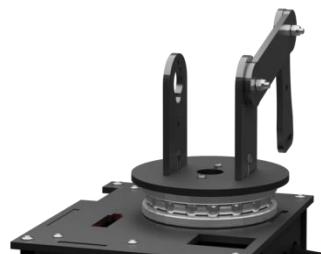
Note:

When the M3 Lock Nut is installed correctly, the C.04 acrylic sheet and C.06 acrylic sheet can be moved easily.



Note:

When the M3 Lock Nut is installed correctly, the C.04 acrylic sheet and C.06 acrylic sheet can be moved easily.



16.Fix M3*8 screw and M3*30 nylon standoff on C.04.

Assemble the following components

IPCS M3*30 Nylon Standoff
IPCS M3*8 Screw



Effect diagram after assembling

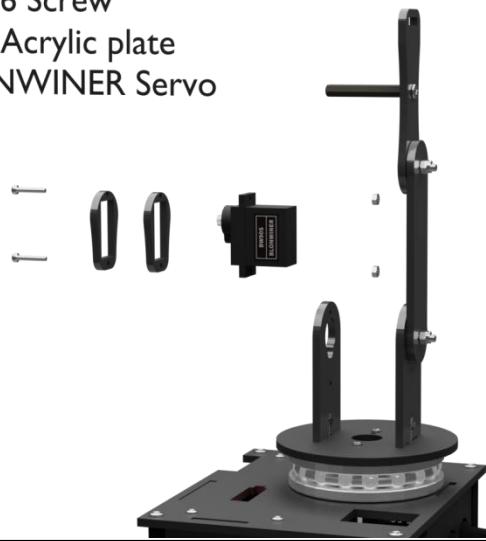
IPCS M3*30 Nylon Standoff
IPCS M3*8 Screw



17.Fix two M2*16 screw, two M2 nut, two C.07 and BLONWINER servo on C.01.

Assemble the following components

2PCS M2 Nut
2PCS M2*16 Screw
2PCS C.07 Acrylic plate
1PCS BLONWINER Servo



Effect diagram after assembling

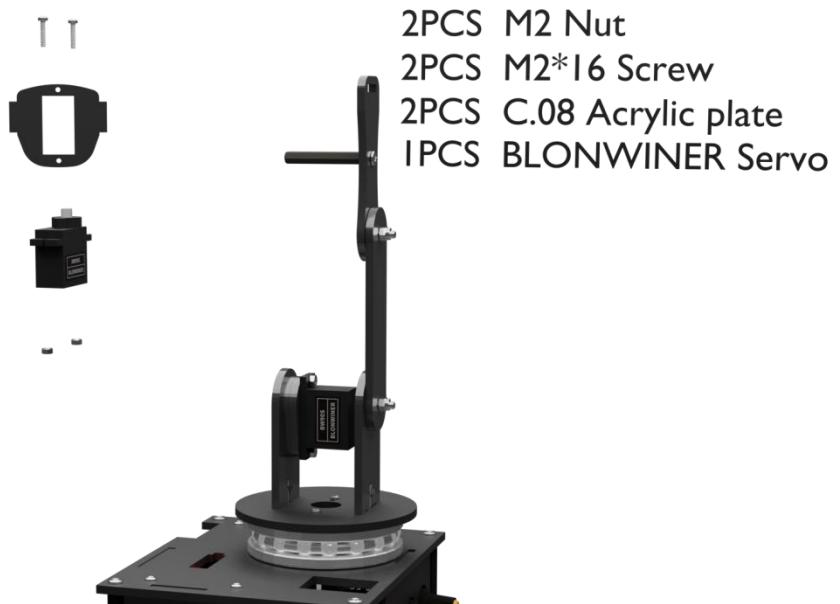


blonwiner@outlook.com



18.Fix two M2*16 screw, two M2 nut, two C.07 and BLONWINER servo on C.04.

Assemble the following components



Effect diagram after assembling

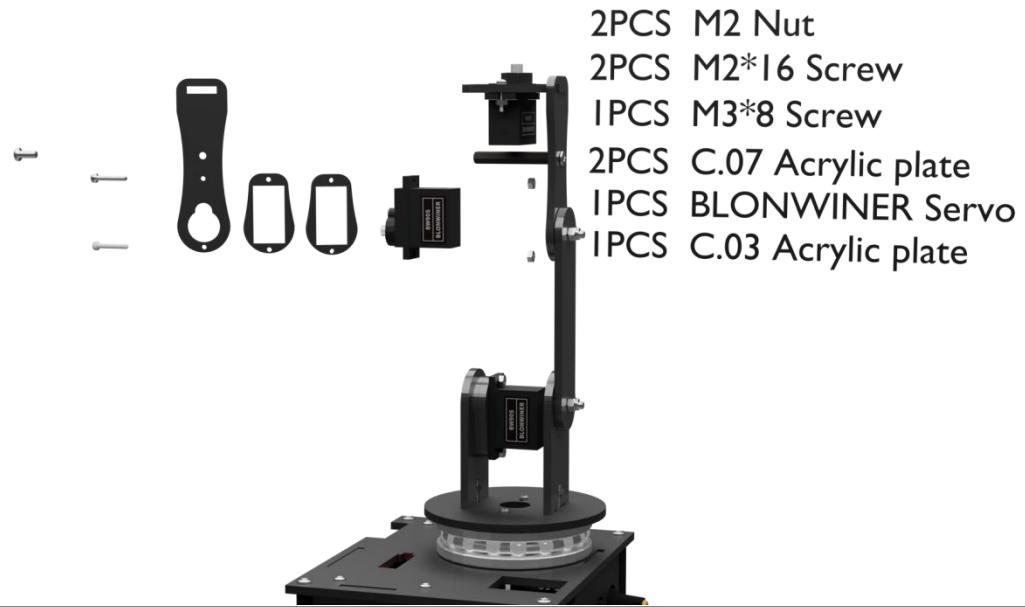


blonwiner@outlook.com



19.Fix M3*8 screw, two M2*16 screw, two M2 nut, two C.07 and BLONWINER servo on C.03.

Assemble the following components



Effect diagram after assembling



2PCS M2 Nut
2PCS M2*16 Screw
1PCS M3*8 Screw
2PCS C.07 Acrylic plate
1PCS BLONWINER Servo
1PCS C.03 Acrylic plate



20.Fix the screw(fixing screw packaged with servo), two M2.5*8 screw, C.05 and two servo arms on two BLONWINER servos.

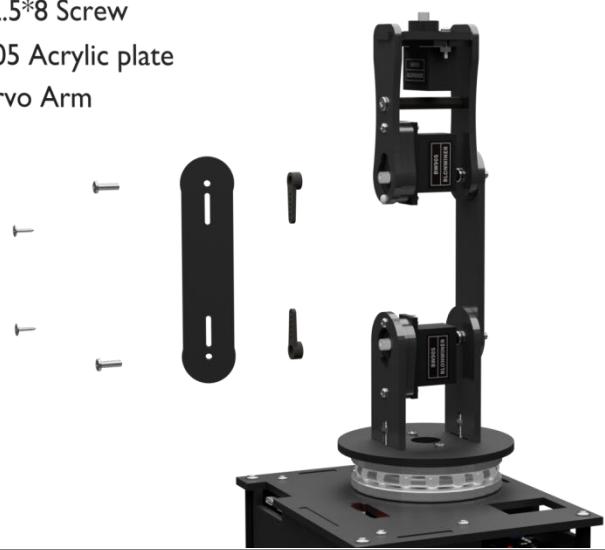
Assemble the following components

2PCS Self-tapping screw packaged with servo

2PCS M2.5*8 Screw

1PCS C.05 Acrylic plate

2PCS Servo Arm



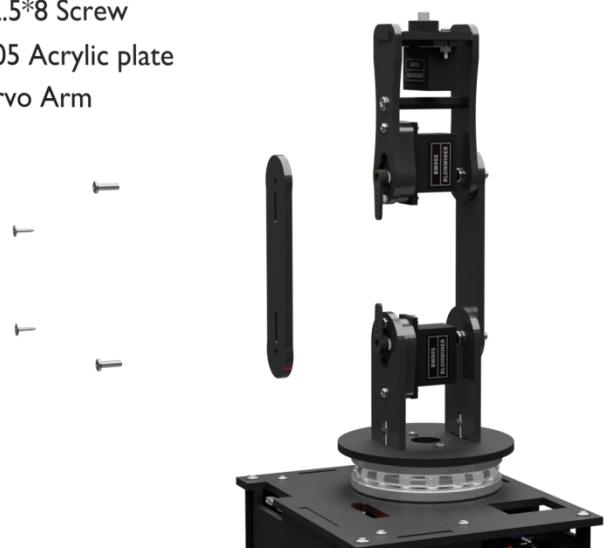
Effect diagram after assembling

2PCS Self-tapping screw packaged with servo

2PCS M2.5*8 Screw

1PCS C.05 Acrylic plate

2PCS Servo Arm



2PCS Self-tapping screw packaged with servo

2PCS M2.5*8 Screw

1PCS C.05 Acrylic plate

2PCS Servo Arm



2PCS Self-tapping screw packaged with servo

2PCS M2.5*8 Screw

1PCS C.05 Acrylic plate

2PCS Servo Arm



21.Fix two M2 nut, two M2*16 screw, BLONWINER servos and two C.07 on C.10.

Assemble the following components



2PCS M2 Nut
2PCS M2*16 Screw
1PCS C.10 Acrylic plates
2PCS C.07 Acrylic plates
1PCS BLONWINER Servo

Effect diagram after assembling



2PCS M2 Nut
2PCS M2*16 Screw
1PCS C.10 Acrylic plates
2PCS C.07 Acrylic plates
1PCS BLONWINER Servo



2PCS M2 Nut
2PCS M2*16 Screw
1PCS C.10 Acrylic plates
2PCS C.07 Acrylic plates
1PCS BLONWINER Servo

22.Fix M3 lock nut, M3*18 screw, M3*8 nylon standoff and D.02 on C.10.

Assemble the following components



IPCS M3*18 Screw
IPCS M3*8 Nylon Standoff
IPCS D.02 Acrylic plates
IPCS M3 Lock Nut

Effect diagram after assembling



IPCS M3*18 Screw
IPCS M3*8 Nylon Standoff
IPCS D.02 Acrylic plates
IPCS M3 Lock Nut



IPCS M3*18 Screw
IPCS M3*8 Nylon Standoff
IPCS D.02 Acrylic plates
IPCS M3 Lock Nut



Note:

The M3 Lock Nut cannot be tightened, otherwise it will cause the robot arm to fail to rotate and even burn the servo.

When the M3 Lock Nut is installed correctly, the D.02 acrylic plate can be moved easily.



Note:

The M3 Lock Nut cannot be tightened, otherwise it will cause the robot arm to fail to rotate and even burn the servo.

When the M3 Lock Nut is installed correctly, the D.02 acrylic plate can be moved easily.



Note:

The M3 Lock Nut cannot be tightened, otherwise it will cause the robot arm to fail to rotate and even burn the servo.

When the M3 Lock Nut is installed correctly, the D.02 acrylic plate can be moved easily.



Note:

The M3 Lock Nut cannot be tightened, otherwise it will cause the robot arm to fail to rotate and even burn the servo.

When the M3 Lock Nut is installed correctly, the D.02 acrylic plate can be moved easily.

23.Fix the screw(fixing screw packaged with servo), M2.5*8 screw, servo arm and D.01 on BLONWINER servo.

Assemble the following components



IPCS Servo Arms

IPCS D.01 Acrylic plates

IPCS M2.5*8 Screw

IPCS Self-tapping screw packaged with servo

Effect diagram after assembling



IPCS Servo Arms

IPCS D.01 Acrylic plates

IPCS M2.5*8 Screw

IPCS Self-tapping screw packaged with servo

blonwiner@outlook.com



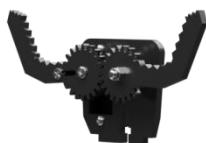
24.Fix the screw(fixing screw packaged with servo), and servo arm on C.09.

Assemble the following components

2PCS Self-tapping screw packaged with servo

1PCS C.09 Acrylic plates

1PCS Servo Arm



T T



Effect diagram after assembling



25.Fix M3*12 screw, M3 nut and C.09 on C.10.

Assemble the following components

IPCS M3*12 Screw

IPCS M3 Nut



Effect diagram after assembling

IPCS M3*12 Screw

IPCS M3 Nut



26.Fix the screw(fixing screw packaged with servo) and servo arm on BLONWINER servo.

Assemble the following components

IPCS Fixing screw packaged with servo



Effect diagram after assembling

IPCS Fixing screw packaged with servo



blonwiner@outlook.com

Rendering of the robot after it is assembled



Functions of the robotic arm

We provide the example to complete the function of carrying objects. You can give full play to your creativity to develop new functions. And learn with us.

Lesson 5 Potentiometer controls Robotic arm

Rotate the potentiometer on the driver board to control the robotic arm to clamp objects. Code PotentiometerControlArm.ino

Specific function descriptions.

- The potentiometer A0 on the driver board controls the movement of servo 1, range from 0 to 180 degrees.
- The potentiometer A1 on the driver board controls the movement of servo 2, range from 0 to 180 degrees.
- The potentiometer A2 on the driver board controls the movement of servo 3, range from 0 to 180 degrees.
- The potentiometer A3 on the driver board controls the movement of servo 4, range from 0 to 180 degrees.
- The potentiometer A6 on the driver board controls the movement of servo 5, range from 90 to 145 degrees.

The movement angle of servo 5 can be adjusted by modifying the value of dataServo5 in the program.

```
19 int dataServo5 = 90; // Servo 5 rotation range(dataServo5=0~180)
```

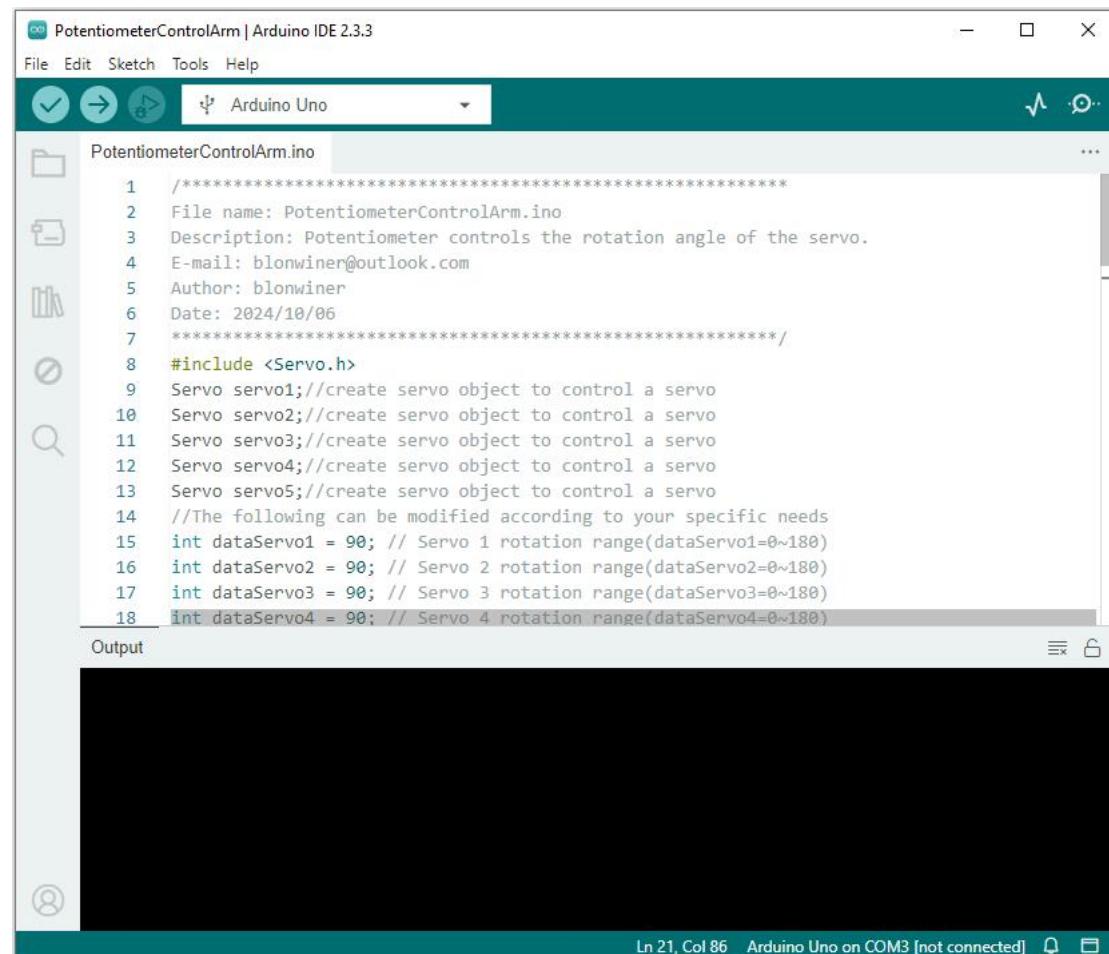
5.1 Components used in this course

Components	Quantity	Picture
BLONWINER Robot	1	

Micro USB Cable	1	
-----------------	---	---

Operating steps

Firstly open the code PotentiometerControlArm.ino we provide.

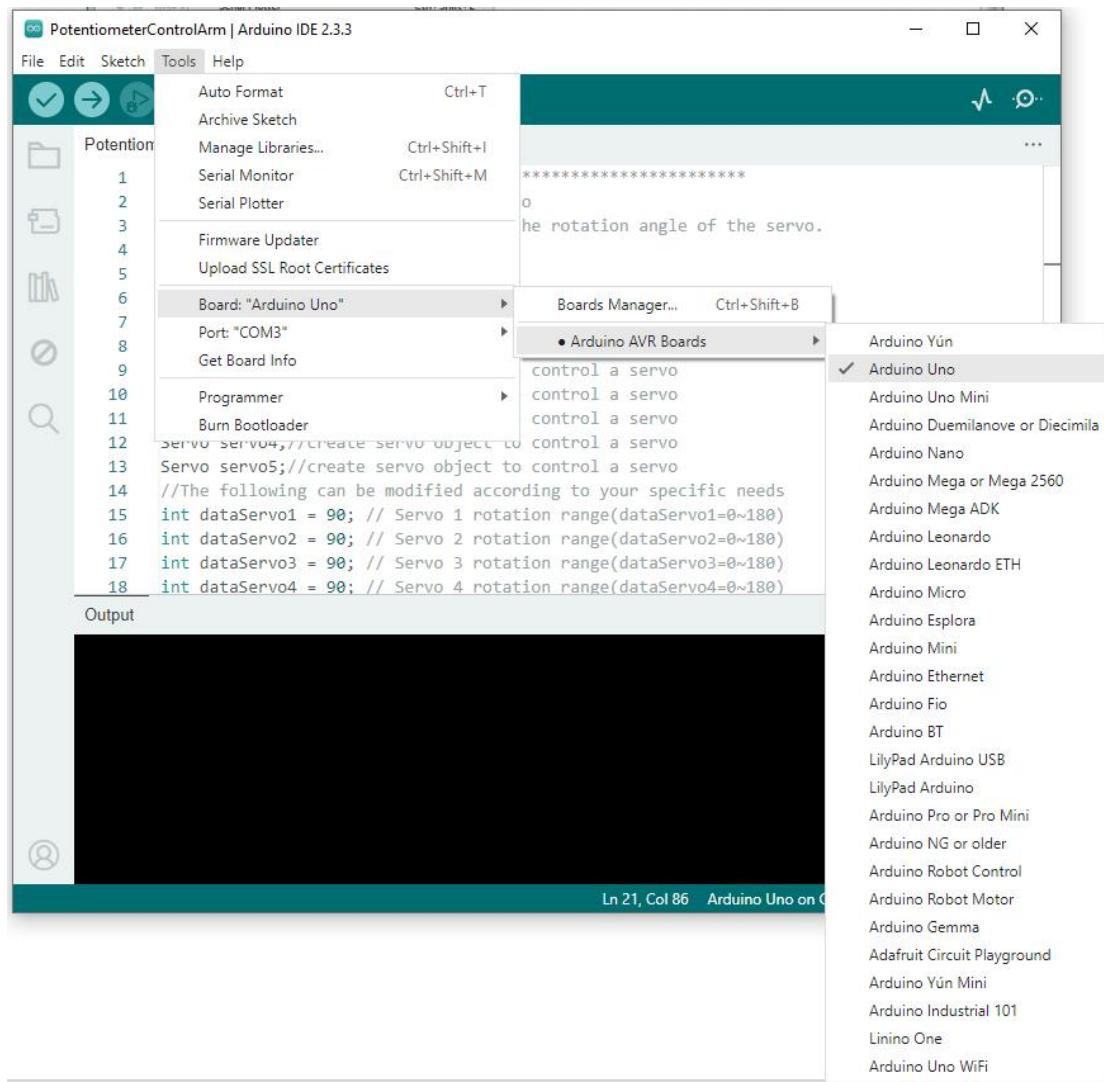


The screenshot shows the Arduino IDE interface with the following details:

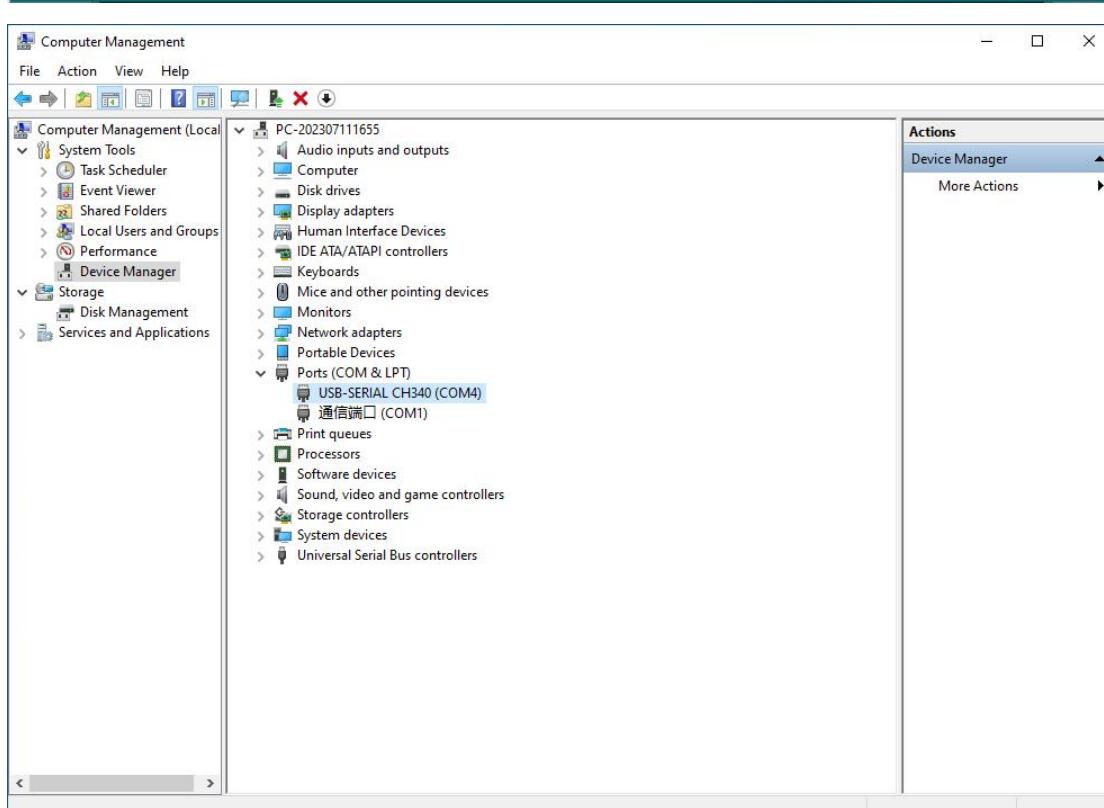
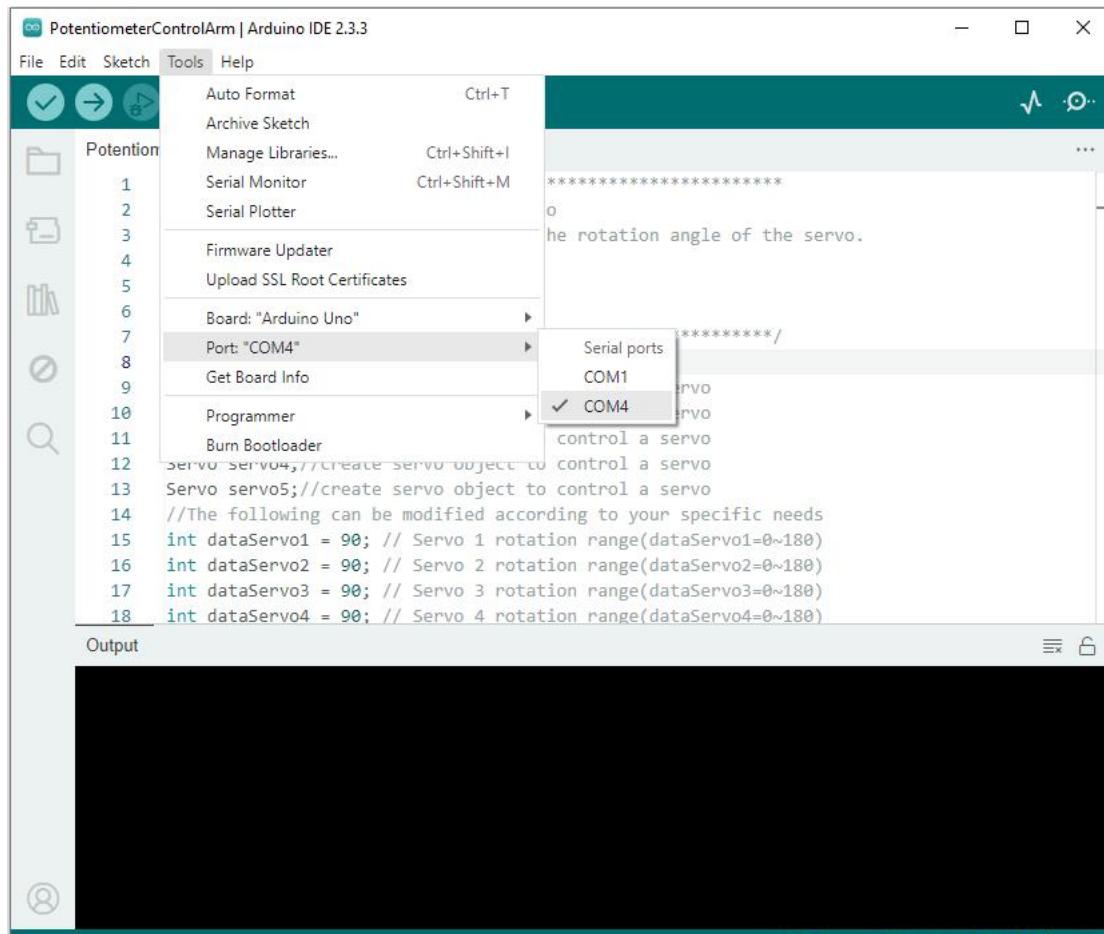
- Title Bar:** PotentiometerControlArm | Arduino IDE 2.3.3
- Menu Bar:** File Edit Sketch Tools Help
- Toolbar:** Includes icons for save, upload, and refresh.
- Sketch Area:** Displays the code for PotentiometerControlArm.ino. The code includes comments and declarations for five servos (servo1 to servo5) with rotation ranges from 0 to 180 degrees. It also includes a note about modifying the rotation range according to specific needs.
- Output Area:** Below the sketch area, it says "Ln 21, Col 86 Arduino Uno on COM3 [not connected]" followed by a bell icon and a refresh icon.

Then connect the robotic arm to the computer with the USB cable. (Note: Do not turn on the power supply to prevent damages of swinging arm. Also pay attention to this in the subsequent operation).

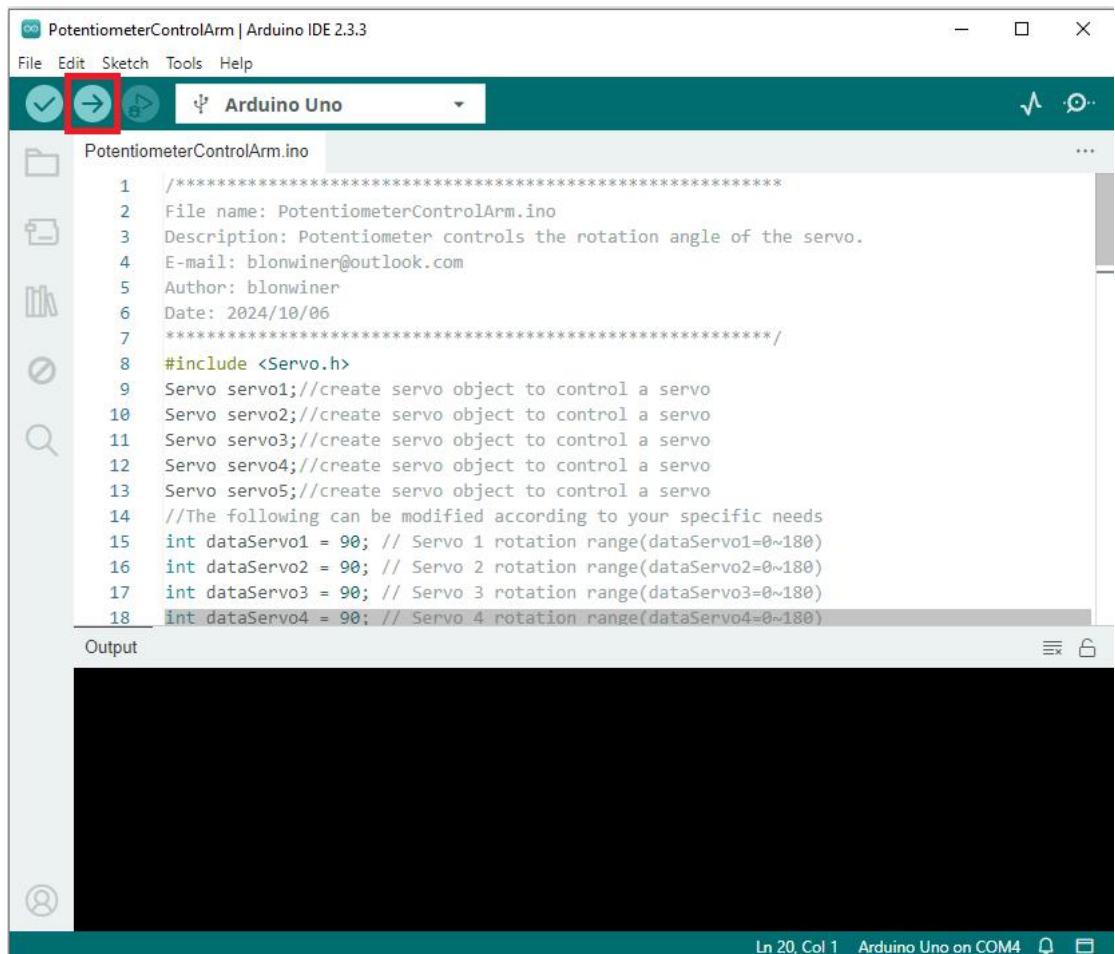
Motor software "Tools"->"Board"->"Arduino/Genuino Uno"



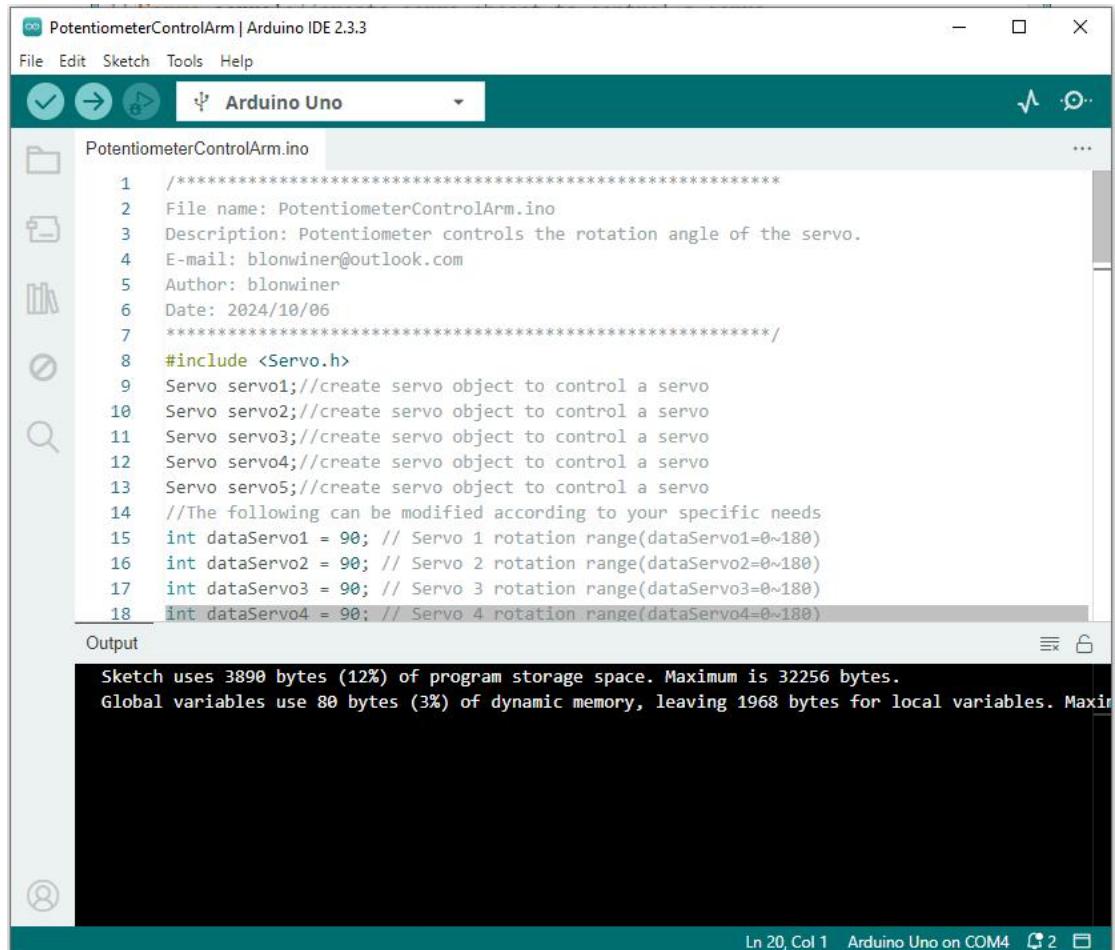
Next, click on "Tools" -> "Port:" -> "COM4" (note that the COM4 here may be recognized differently on different computers, it can be COM1, COM2 or COM3 and so on.)



Then click "Upload" to upload the program to the UNO board.



When the software prompts the following information, the code upload is complete.



The screenshot shows the Arduino IDE 2.3.3 interface with the following details:

- Title Bar:** PotentiometerControlArm | Arduino IDE 2.3.3
- Toolbar:** File, Edit, Sketch, Tools, Help
- Board Selection:** Arduino Uno
- Sketch Name:** PotentiometerControlArm.ino
- Code Content:**

```
1 //*****
2 // File name: PotentiometerControlArm.ino
3 // Description: Potentiometer controls the rotation angle of the servo.
4 // E-mail: blonwiner@outlook.com
5 // Author: blonwiner
6 // Date: 2024/10/06
7 *****/
8 #include <Servo.h>
9 Servo servo1;//create servo object to control a servo
10 Servo servo2;//create servo object to control a servo
11 Servo servo3;//create servo object to control a servo
12 Servo servo4;//create servo object to control a servo
13 Servo servo5;//create servo object to control a servo
14 //The following can be modified according to your specific needs
15 int dataServo1 = 90; // Servo 1 rotation range(dataServo1=0~180)
16 int dataServo2 = 90; // Servo 2 rotation range(dataServo2=0~180)
17 int dataServo3 = 90; // Servo 3 rotation range(dataServo3=0~180)
18 int dataServo4 = 90; // Servo 4 rotation range(dataServo4=0~180)
```
- Output Panel:**

```
Sketch uses 3890 bytes (12%) of program storage space. Maximum is 32256 bytes.
Global variables use 80 bytes (3%) of dynamic memory, leaving 1968 bytes for local variables. Maxin
```
- Status Bar:** Ln 20, Col 1 Arduino Uno on COM4 2

Next, unplug the USB cable connected to the robotic arm. **Note: Do not turn on the power of the arm after downloading the program.** Adjust the four potentiometers on the driver board to the center first, as shown below:



Gently support the robotic arm with your hand to prevent swinging arm. Turn on the power, and then rotate the four potentiometers on the driver board to control the arm to clamp and carry objects. The rotation angle of Servo5 is set in the code.

blonwiner@outlook.com



Lesson 6 Processing controls servo

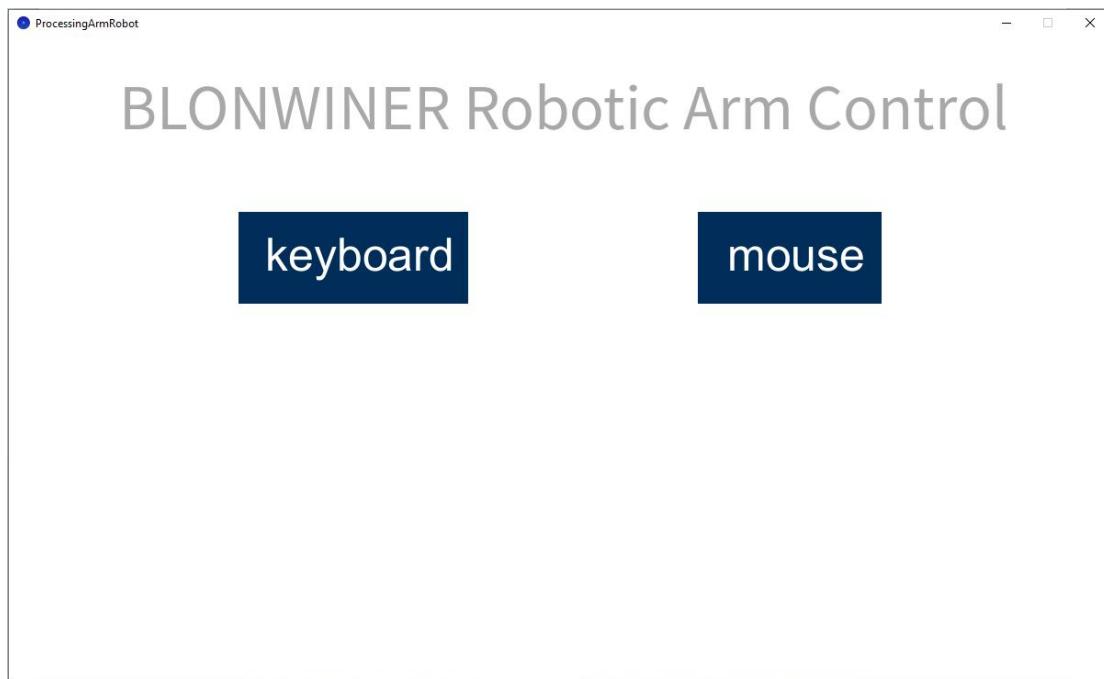
If the code hasn't been used in processing before, the library file controlP5 needs to be added.

You can refer to the previous manual for the installation process of Processing.

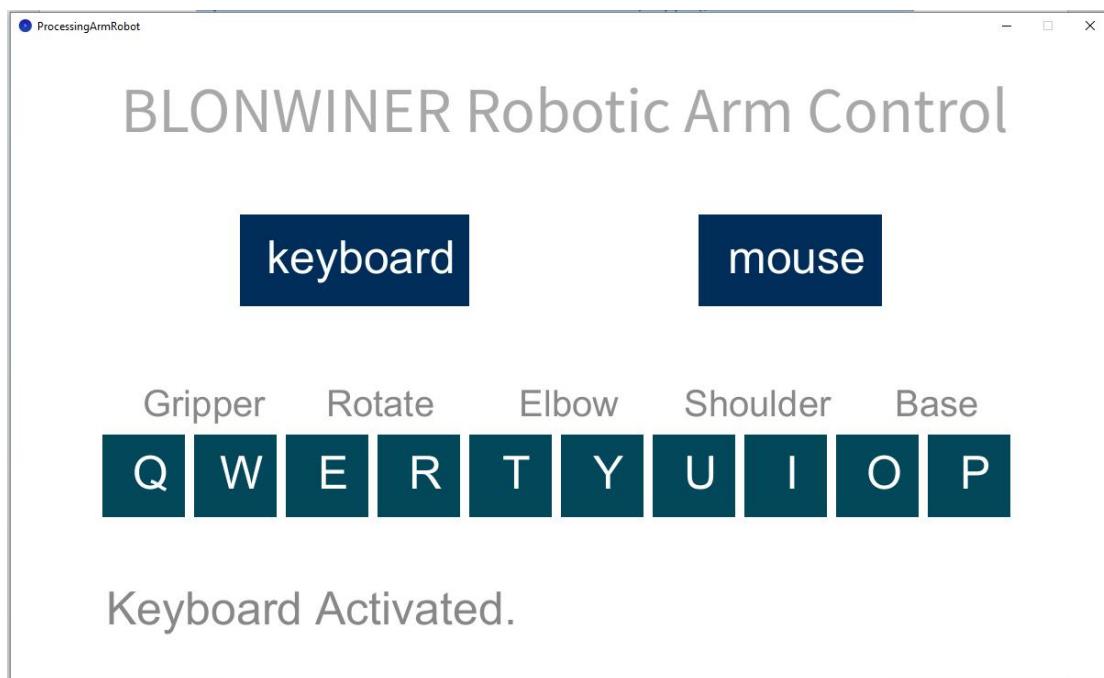
Specific function introduction:

Control the robotic arm to carry objects through Processing with serial communication.

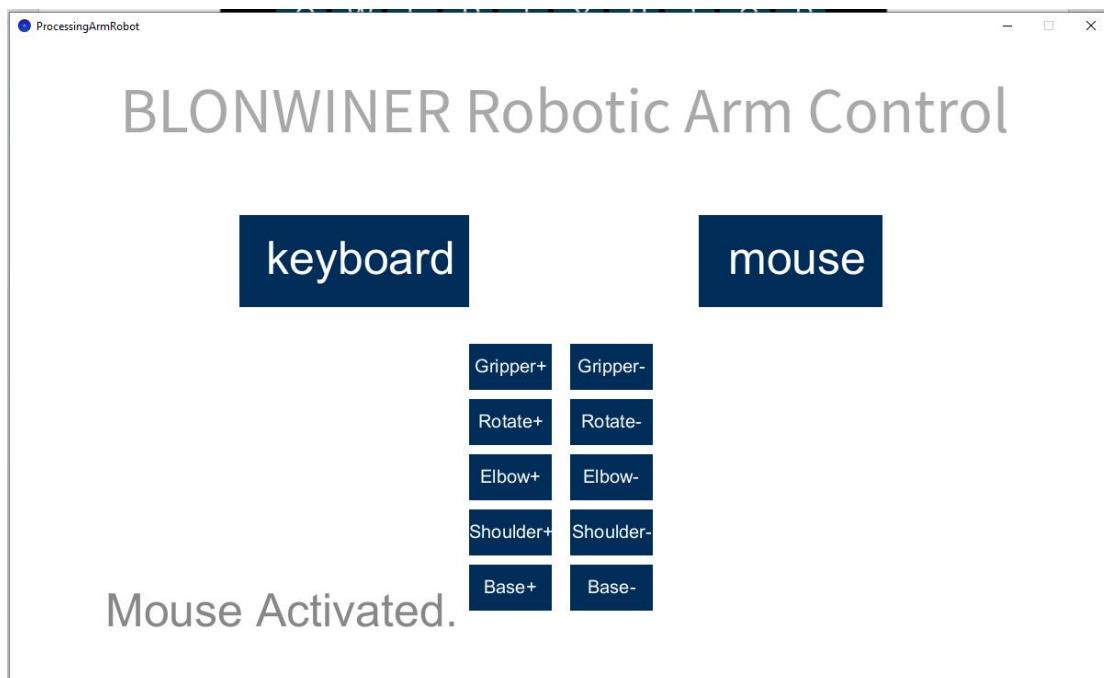
Processing interfaces are as follows.



Click "keyboard" the following interface will appear. Next, press the corresponding button on the keyboard to control the arm. "Q" and "W" control servo4 (Gripper), "E" and "R" control servo5 (Rotate), "T" and "Y" control servo3 (Elbow), "U" and "I" control servo2 (Shoulder), "O" and "P" control servo1 (Base).



Click "mouse" and the following interface will appear.



At this point, click the corresponding button, the robotic arm will make the corresponding movement. "Gripper+" and "Gripper-" control the servo4, "Rotate+" and "Rotate-" control the servo5, "Elbow+" and "Elbow-" control the servo3, "Shoulder+" and "Shoulder-" control the servo2, " Base+" and "Base-" control the servo1.

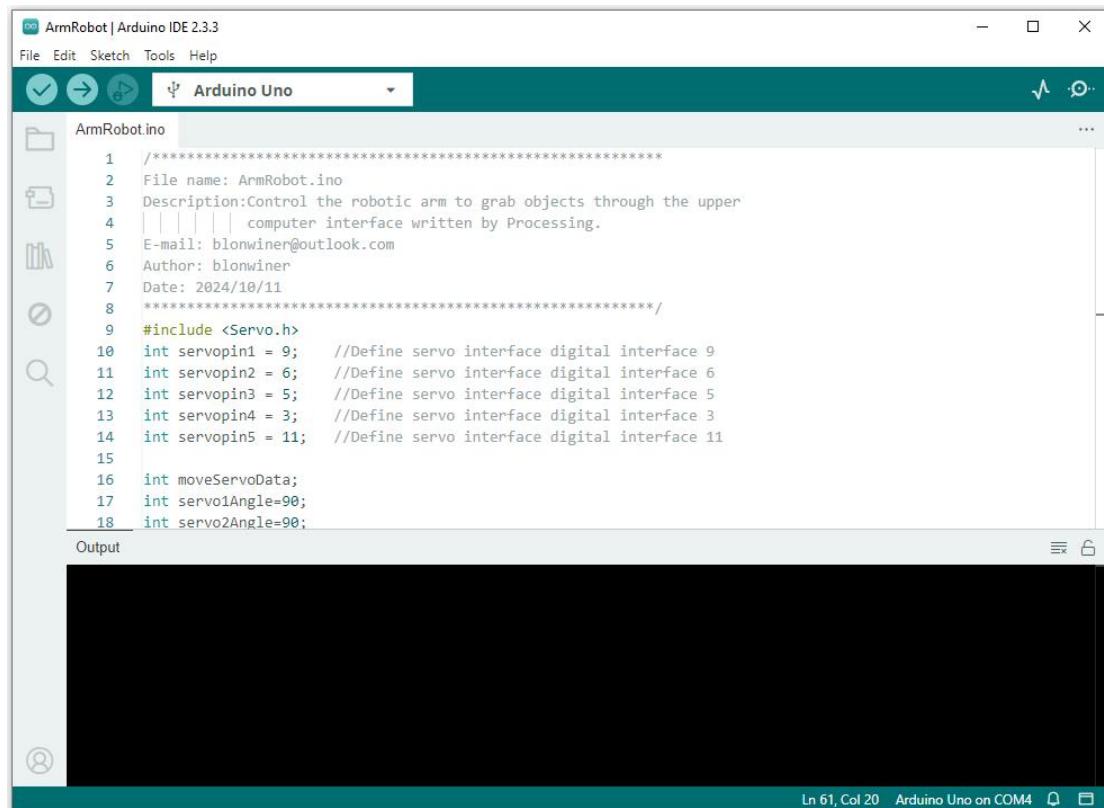
6.1 Components used in this course

Components	Quantity	Picture
BLOWINER Robot	1	

Micro USB Cable	1	
-----------------	---	---

Operating steps:

Open the file ArmRobot.ino, as shown below:



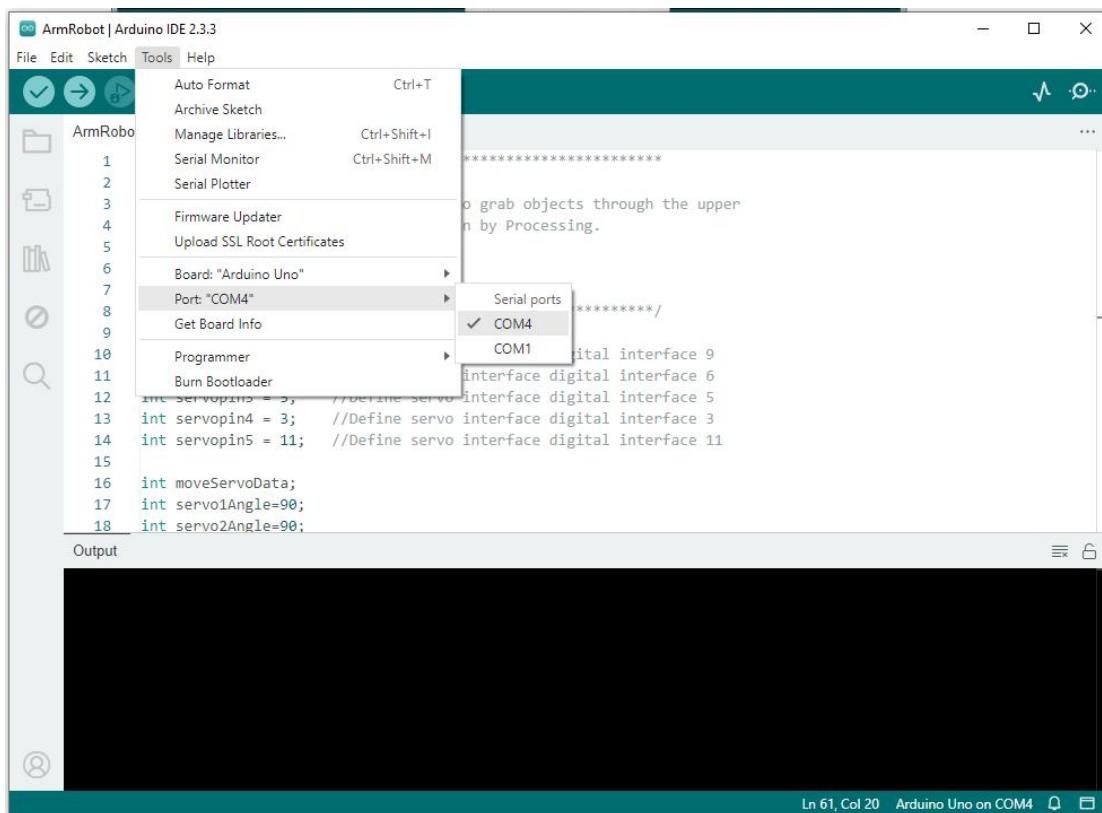
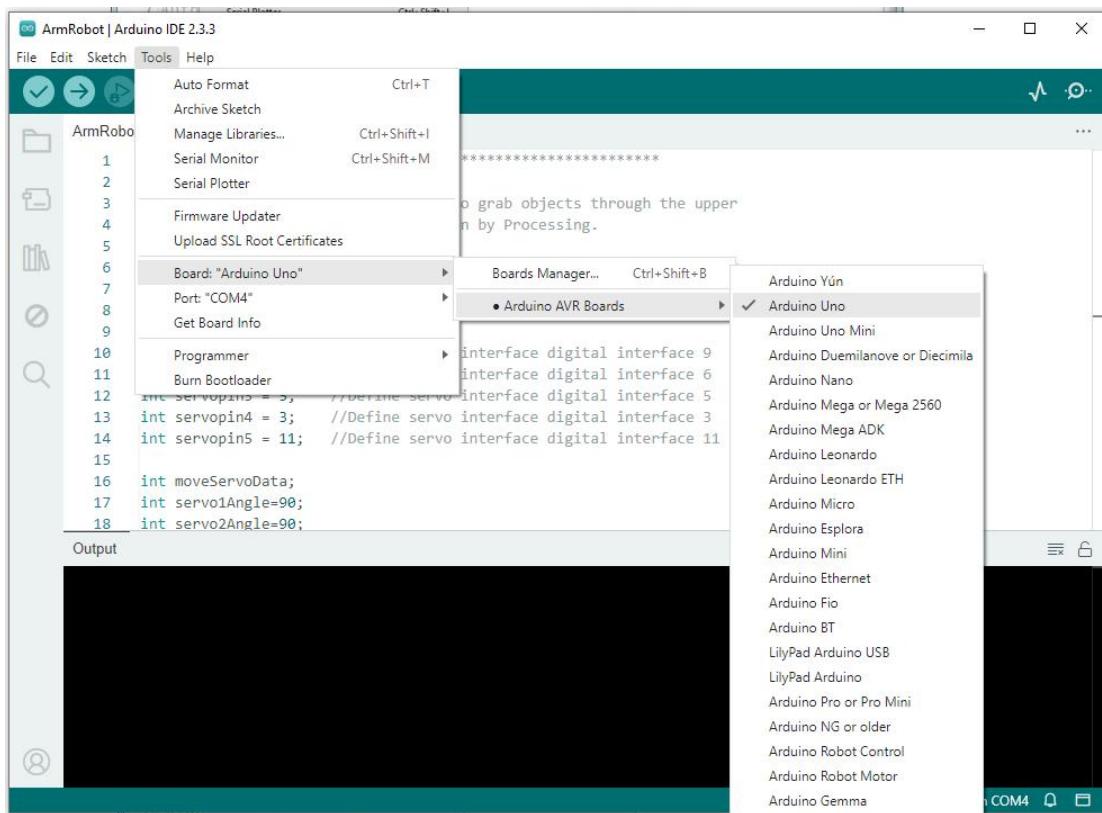
The screenshot shows the Arduino IDE interface with the following details:

- Title Bar:** ArmRobot | Arduino IDE 2.3.3
- Menu Bar:** File Edit Sketch Tools Help
- Board Selector:** Arduino Uno
- Sketch Area:** The code for `ArmRobot.ino` is displayed:

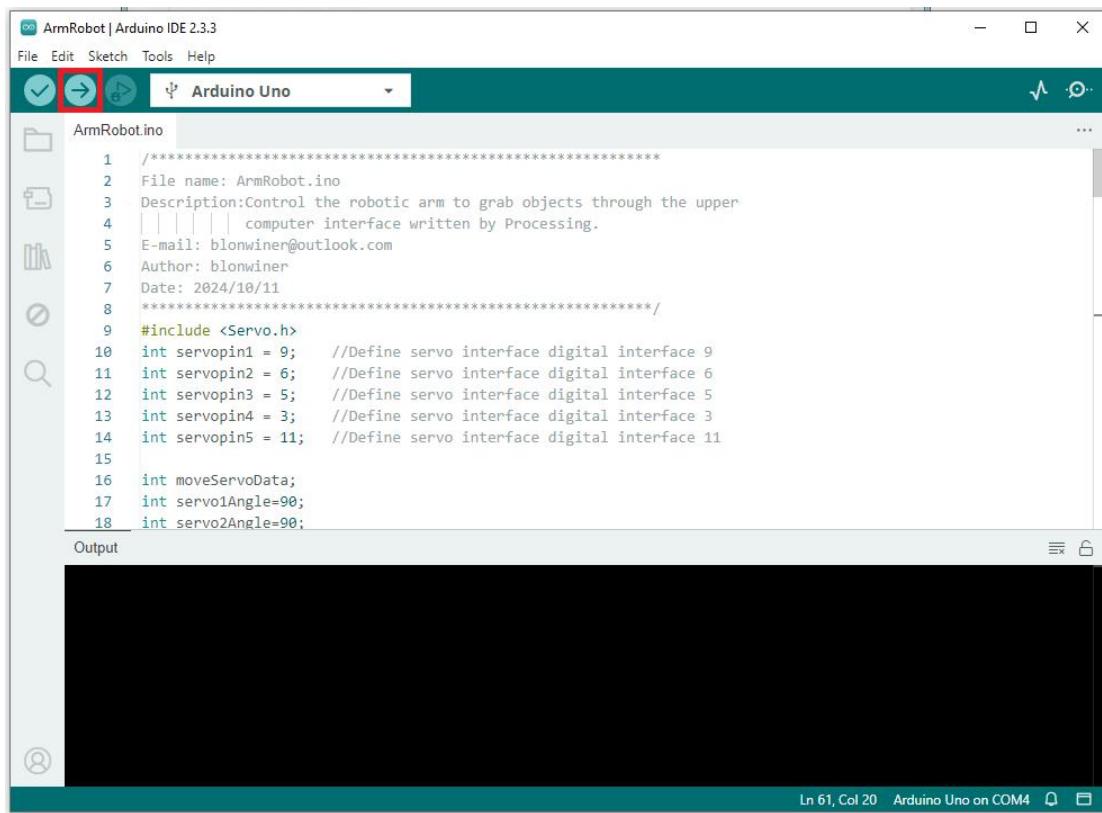
```
1  /*****
2  File name: ArmRobot.ino
3  Description:Control the robotic arm to grab objects through the upper
4  ||||| computer interface written by Processing.
5  E-mail: blonwiner@outlook.com
6  Author: blonwiner
7  Date: 2024/10/11
8 *****/
9 #include <Servo.h>
10 int servopin1 = 9; //Define servo interface digital interface 9
11 int servopin2 = 6; //Define servo interface digital interface 6
12 int servopin3 = 5; //Define servo interface digital interface 5
13 int servopin4 = 3; //Define servo interface digital interface 3
14 int servopin5 = 11; //Define servo interface digital interface 11
15
16 int moveServoData;
17 int servo1Angle=90;
18 int servo2Angle=90;
```
- Output Area:** A large black rectangular area representing the serial monitor.
- Status Bar:** Ln 61, Col 20 Arduino Uno on COM4

Next, connect the robotic arm to the computer with the USB cable. Select the development board model and port, as shown below:

blonwiner@outlook.com



Click "Upload" to upload the code to UNO, as shown below:



ArmRobot | Arduino IDE 2.3.3

File Edit Sketch Tools Help

Arduino Uno

ArmRobot.ino

```
1  /*****
2  File name: ArmRobot.ino
3  Description:Control the robotic arm to grab objects through the upper
4  [ ] computer interface written by Processing.
5  E-mail: blonwiner@outlook.com
6  Author: blonwiner
7  Date: 2024/10/11
8  *****/
9  #include <Servo.h>
10 int servopin1 = 9;      //Define servo interface digital interface 9
11 int servopin2 = 6;      //Define servo interface digital interface 6
12 int servopin3 = 5;      //Define servo interface digital interface 5
13 int servopin4 = 3;      //Define servo interface digital interface 3
14 int servopin5 = 11;     //Define servo interface digital interface 11
15
16 int moveServoData;
17 int servo1Angle=90;
18 int servo2Angle=90;
```

Output

Ln 61, Col 20 Arduino Uno on COM4

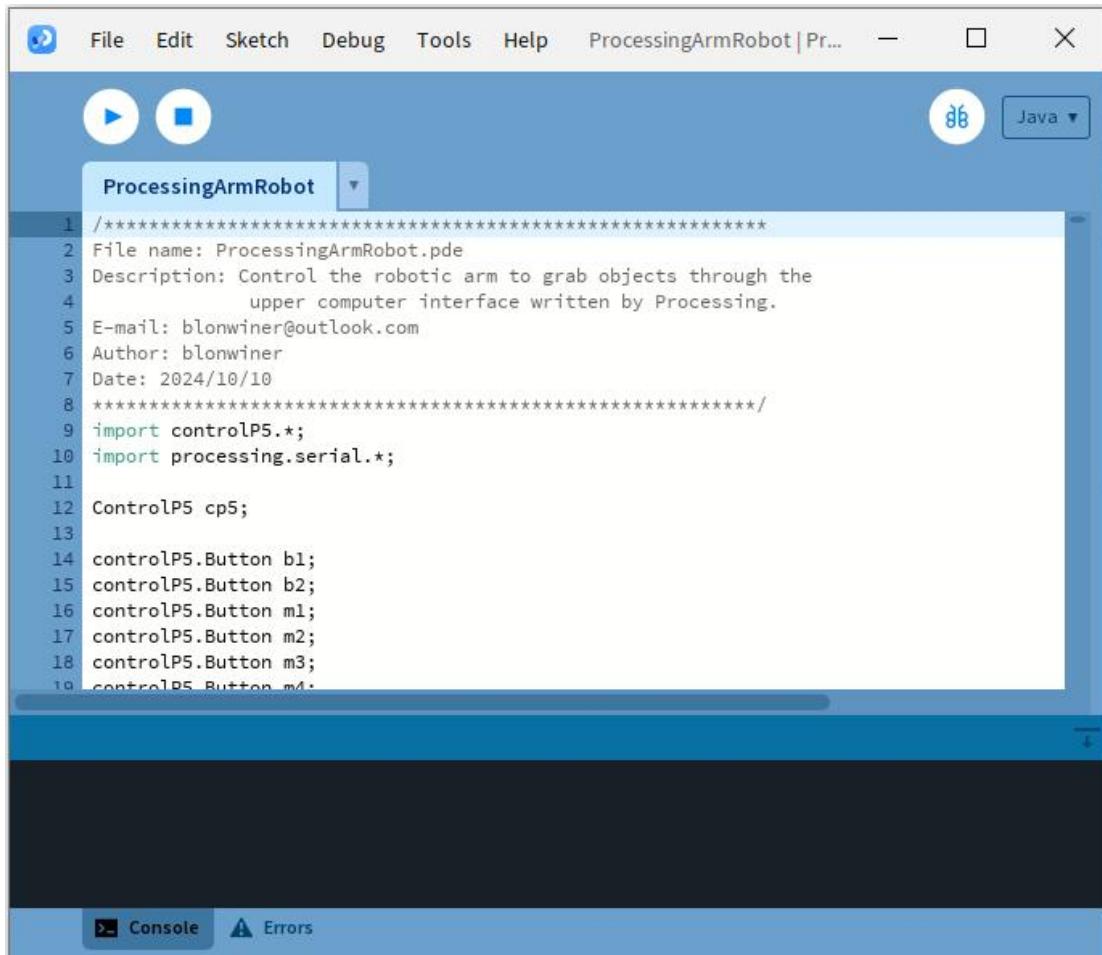
After downloading, close ArmRobot.ino.

Note: that the arm is still connected to the computer with the USB cable. Rotate the arm to the position as shown in the figure below (the initialization value is set in the program. If the arm is not in the position as shown below before powered on, it will swing and may cause damages when is energized), and then turn on the power.

blonwiner@outlook.com



Next open ProcessingArmRobot.pde



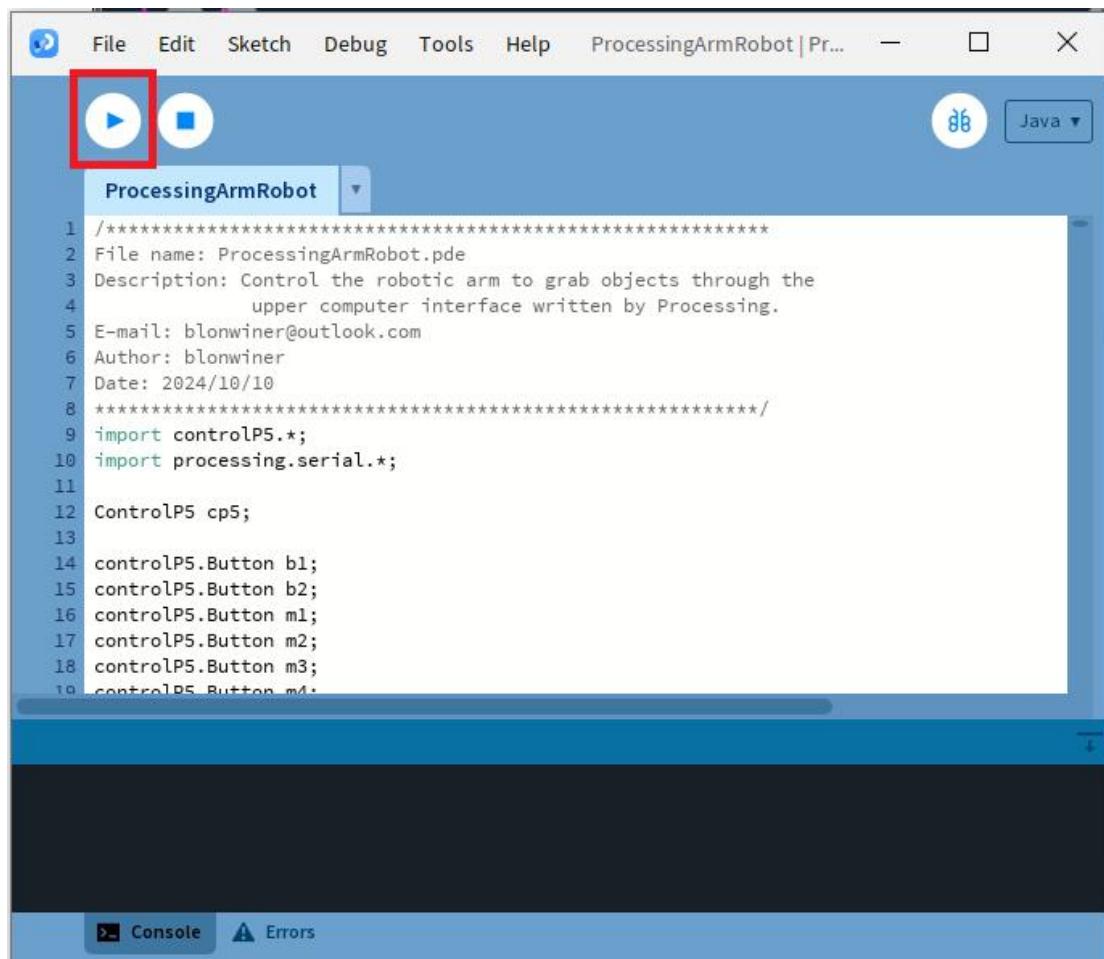
The screenshot shows the Processing IDE interface. The title bar reads "ProcessingArmRobot | Pr...". The menu bar includes File, Edit, Sketch, Debug, Tools, Help, and Java. Below the menu is a toolbar with a play button, a stop button, and a refresh icon. A dropdown menu is open over the play button. The main code editor window has a title "ProcessingArmRobot" and contains the following PDE code:

```
1 //*****  
2 File name: ProcessingArmRobot.pde  
3 Description: Control the robotic arm to grab objects through the  
4 upper computer interface written by Processing.  
5 E-mail: blonwiner@outlook.com  
6 Author: blonwiner  
7 Date: 2024/10/10  
8 *****  
9 import controlP5.*;  
10 import processing.serial.*;  
11  
12 ControlP5 cp5;  
13  
14 controlP5.Button b1;  
15 controlP5.Button b2;  
16 controlP5.Button m1;  
17 controlP5.Button m2;  
18 controlP5.Button m3;  
19 controlP5.Button m4;
```

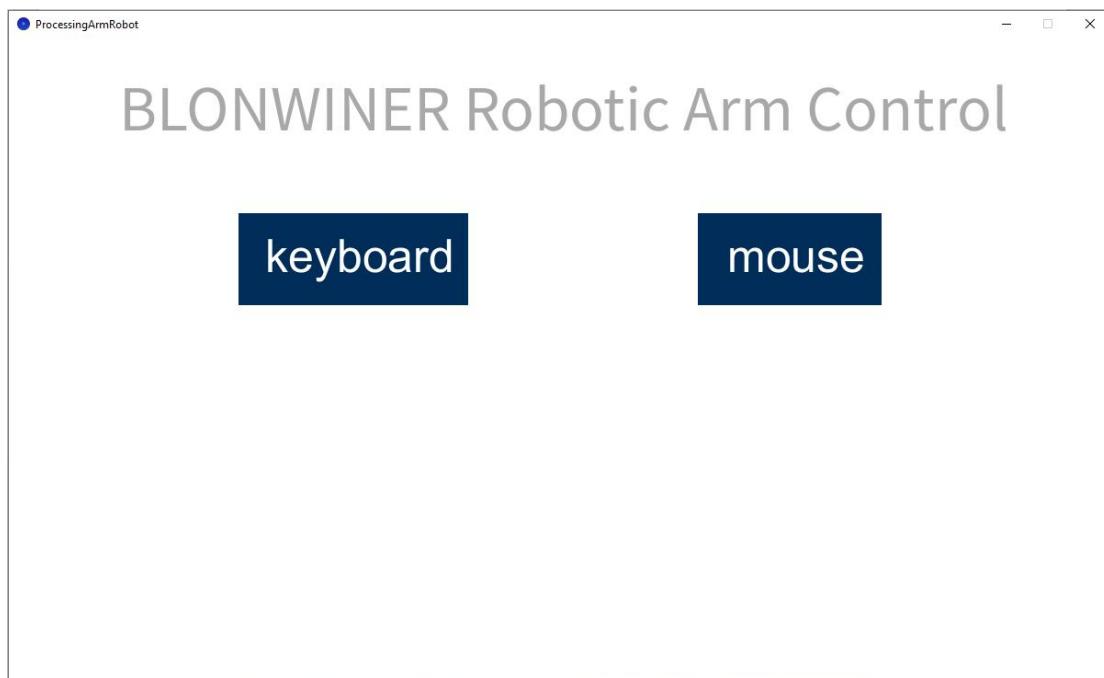
At the bottom of the IDE, there are tabs for "Console" and "Errors".

Click "Run" to run the code, as shown below:

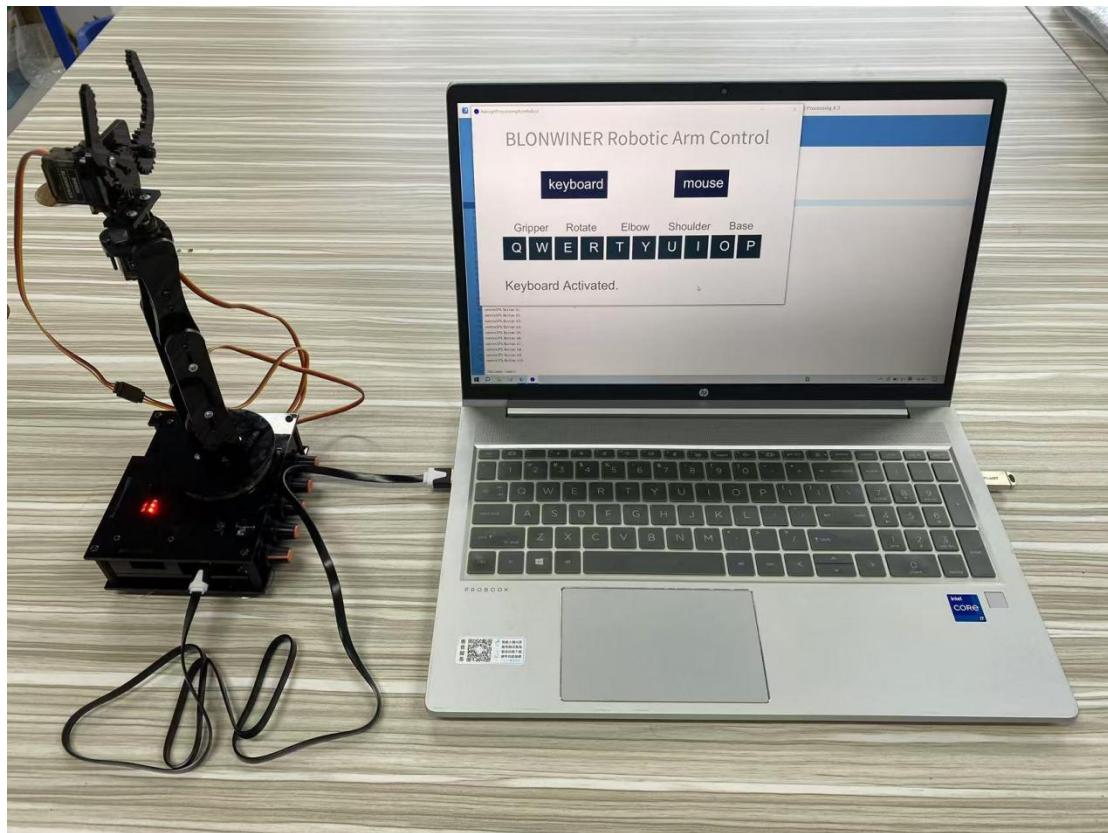
blonwiner@outlook.com



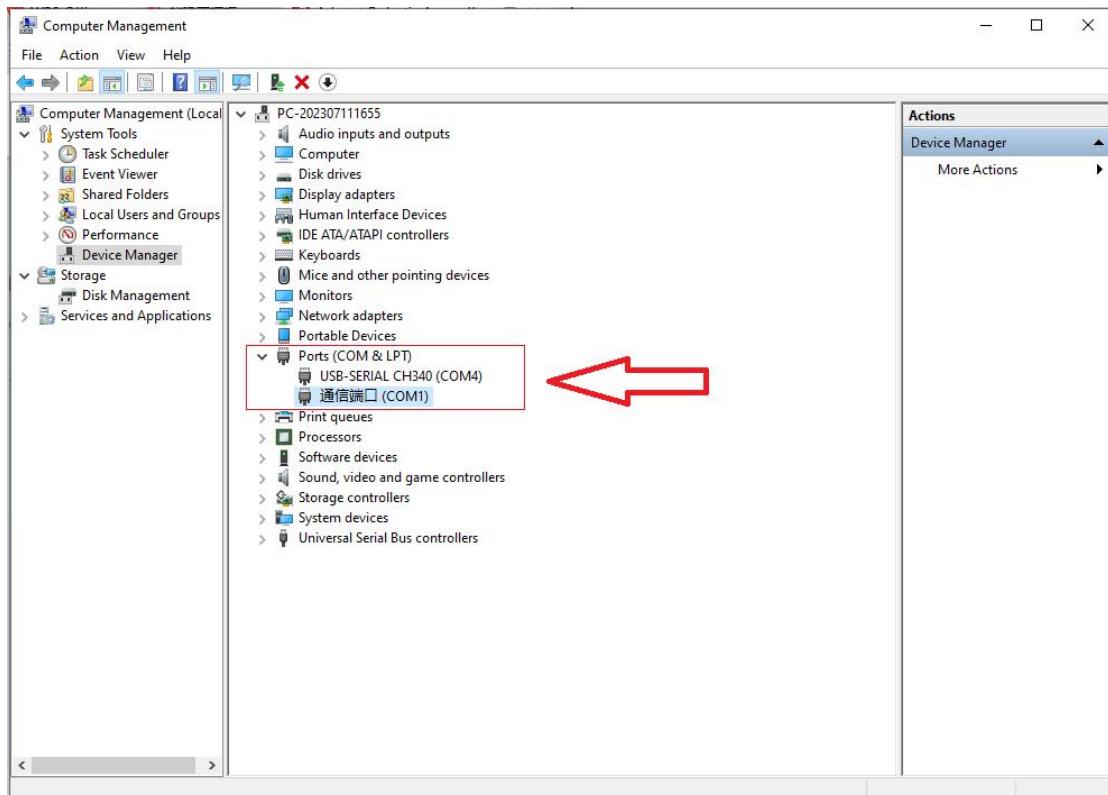
The interface of successful running is as below:



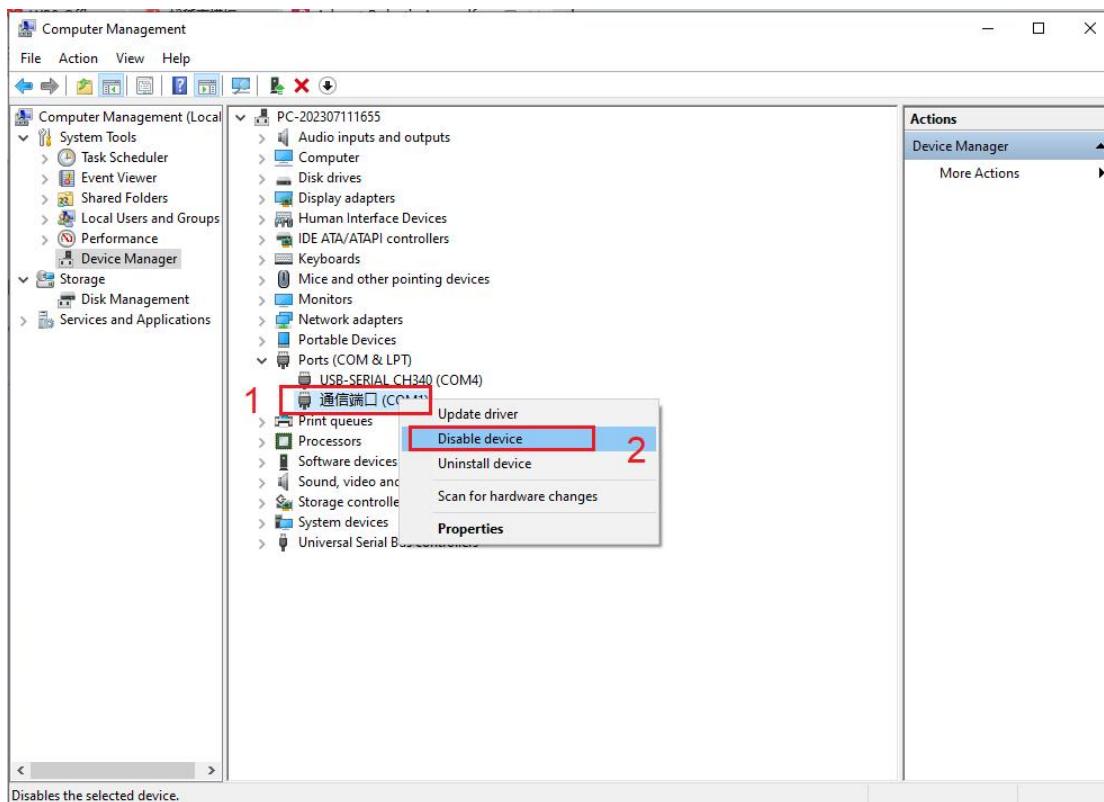
The control interface can then be used to control the robotic arm.



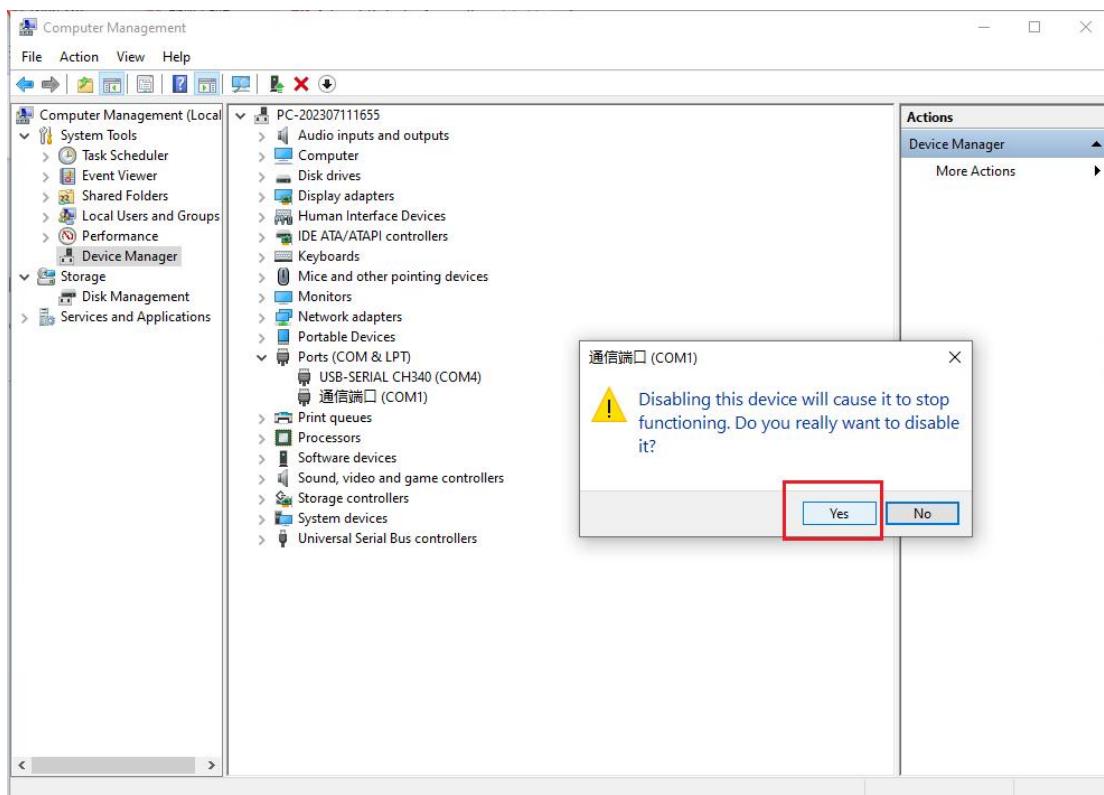
Note: When your computer's COM port has multiple displays(COM1 and COM4), as shown in the following figure:



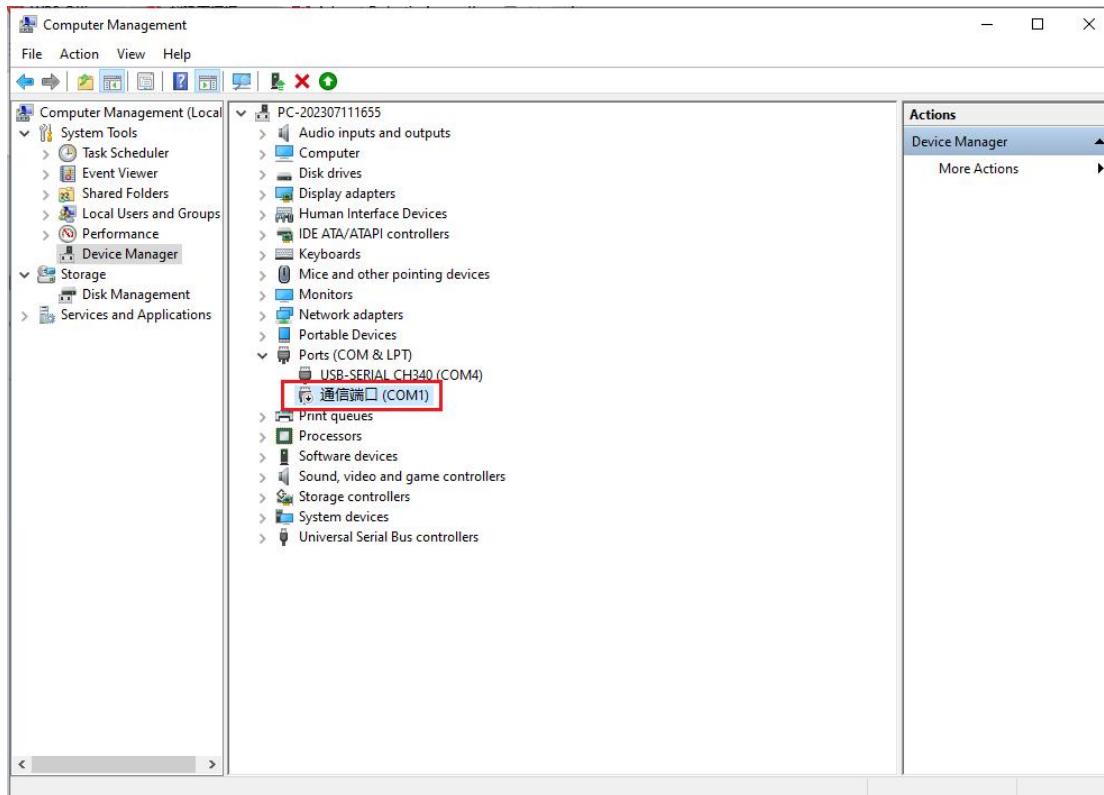
Running Processing fails. In this case, we need to select the COM port(COM1) that is not CH340 and right-click. Click on "Disable device"



Click on "Yes"



At this point, the port COM1 stops working.



At this time, only the USB-SERIAL CH340 (COM4) port is working, and the BLONWINER Arduino ARM Board can be successfully connected when the processing program is running.

Lesson 7 IR Remote Controller

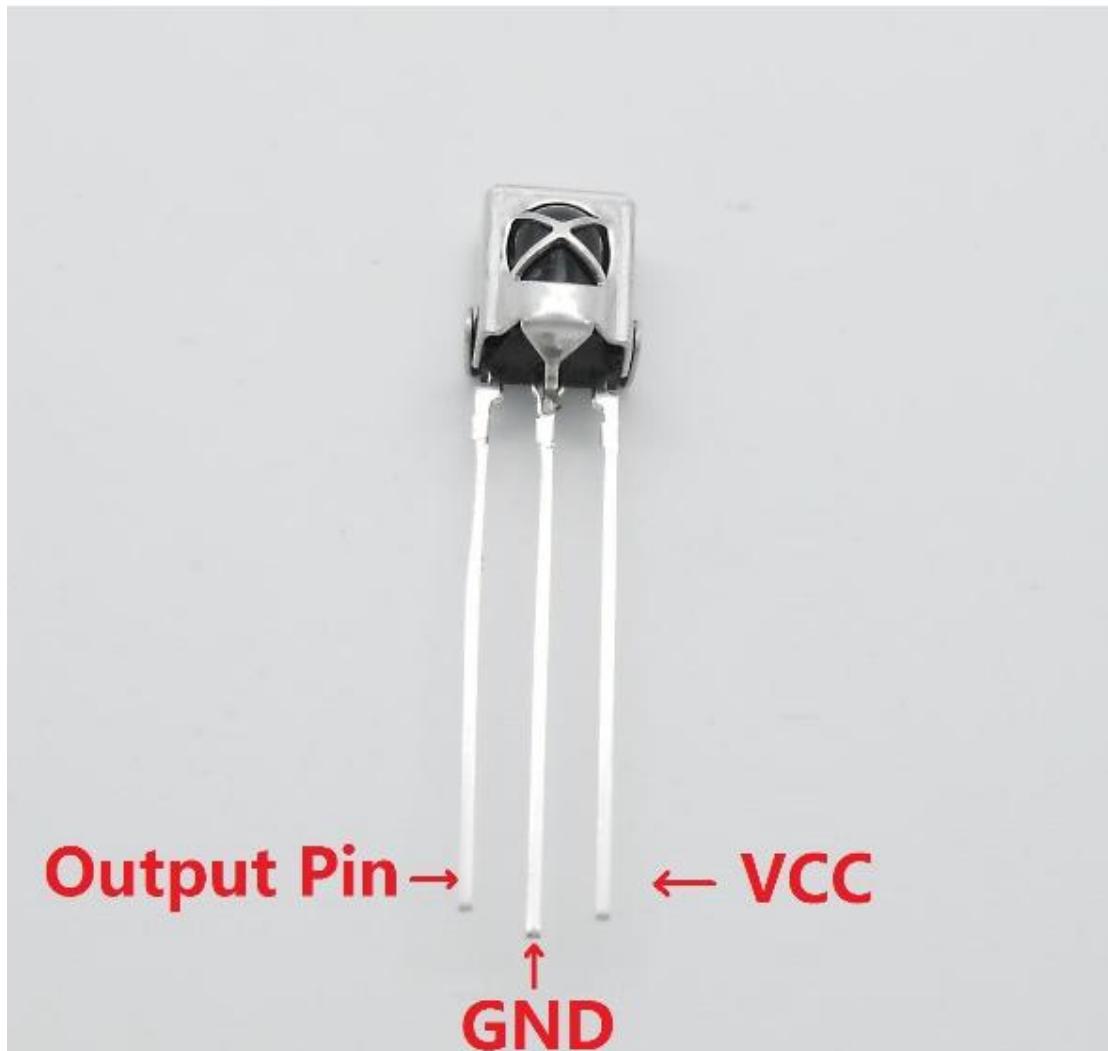
In this lesson, we will learn how to use an IR receiver to receive signals from a remote controller.

7.1 Components used in this course

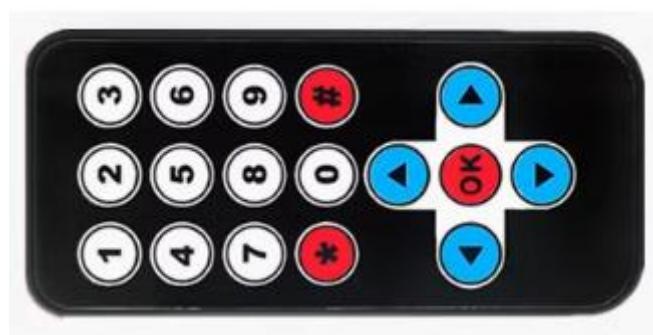
Components	Quantity	Picture
BLONWINER Robot	1	
Micro USB Cable	1	
Remote Controller	1	

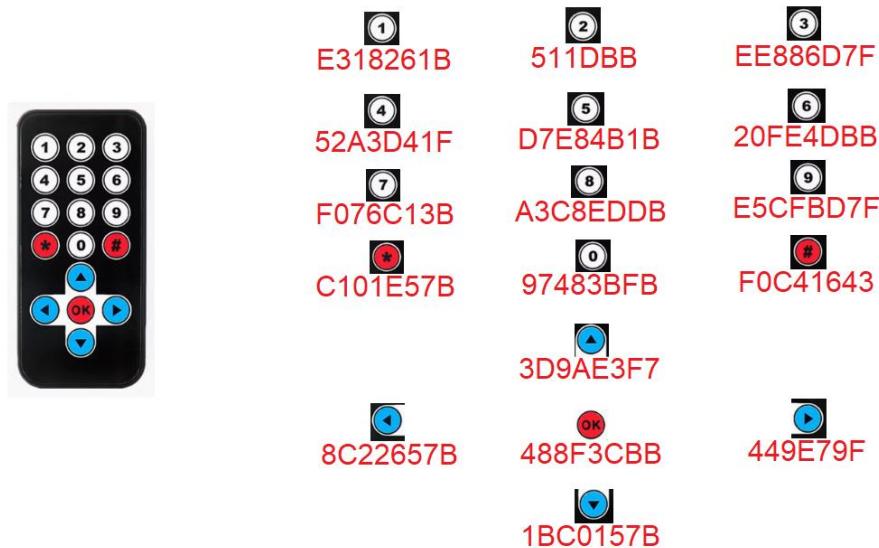
Principle

The IR receiver HX1838 can receive signals from an infrared (IR) remote controller. It has only three pins: signal, VCC and GND. So it is simple to connect with an Arduino board.



The following figure shows an IR remote controller:





In this experiment, we program the Arduino board to receive the infrared signals, and then send the received data to Serial Monitor. In the program, we use the Arduino-IRremote-master library (provided).

Operating steps

Firstly open the code IRRemoteController.ino we provide.

```

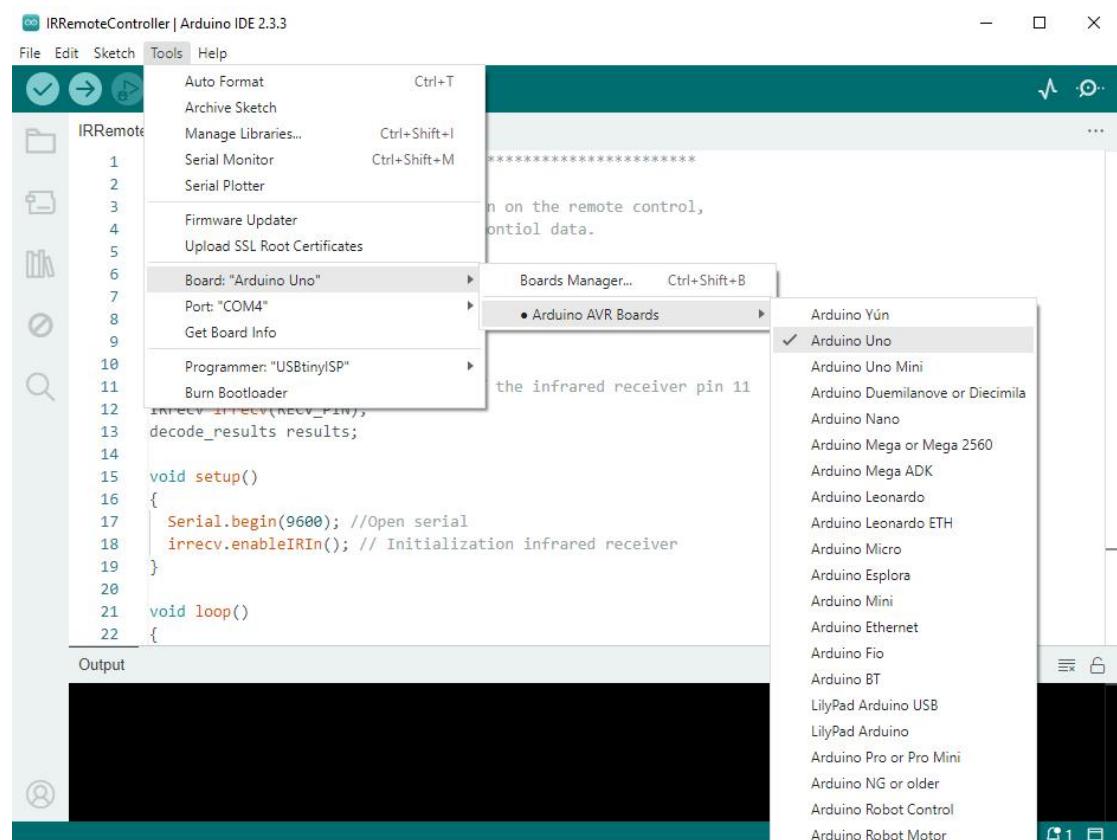
  IRRemoteController | Arduino IDE 2.3.3
  File Edit Sketch Tools Help
  ✓ → ↻ ⚡ Arduino Uno
  IRRemoteController.ino
  1  ****
  2  File name: IRRemoteController.ino
  3  Description: When you click the button on the remote control,
  4  | | | | | you can see the serial monitor data.
  5  E-mail: blonwiner@outlook.com
  6  Author: blonwiner
  7  Date: 2024/10/20
  8  ****
  9  #include <IRremote.h>
 10
 11 int RECV_PIN = 11;//The definition of the infrared receiver pin 11
 12 IRrecv irrecv(RECV_PIN);
 13 decode_results results;
 14
 15 void setup()
 16 {
 17   Serial.begin(9600); //Open serial
 18   irrecv.enableIRIn(); // Initialization infrared receiver
 19 }
 20
 21 void loop()
 22 {
  
```

Output

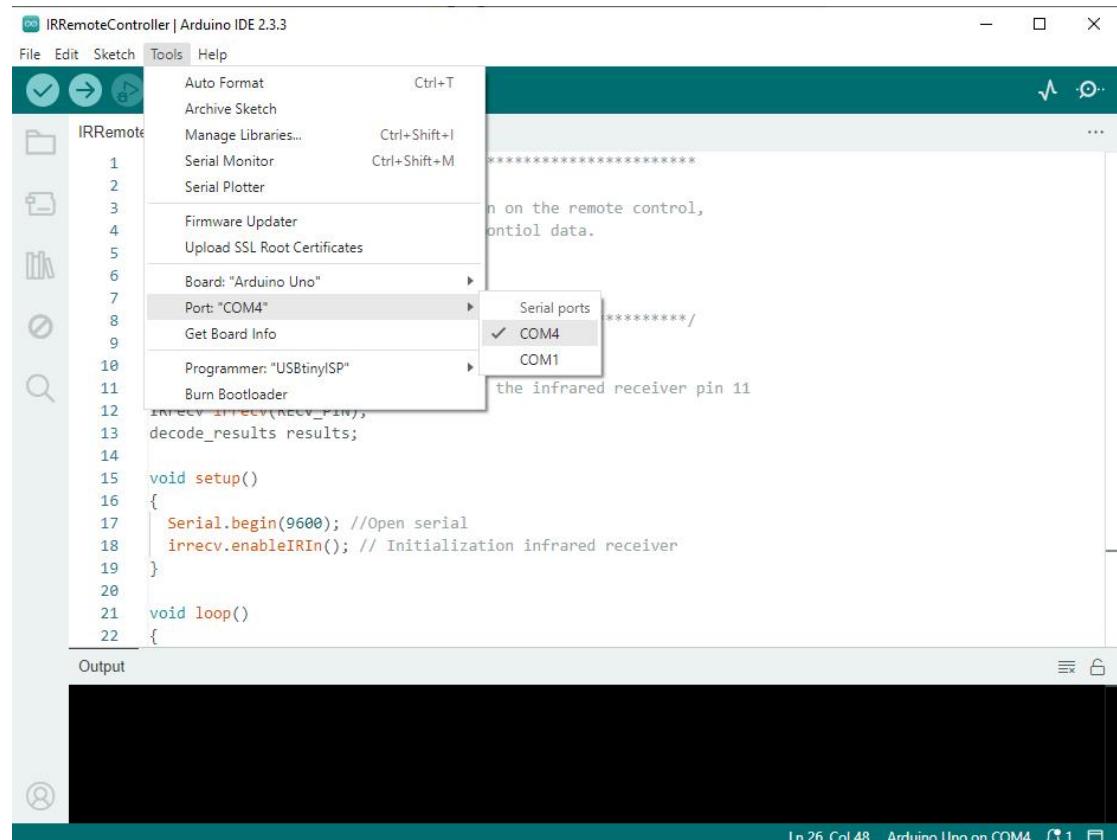
Ln 26, Col 48 Arduino Uno on COM4 1 1

Then connect the robotic arm to the computer with the USB cable. (Note: Do not turn on the power supply to prevent damages of swinging arm. Also pay attention to this in the subsequent operation).

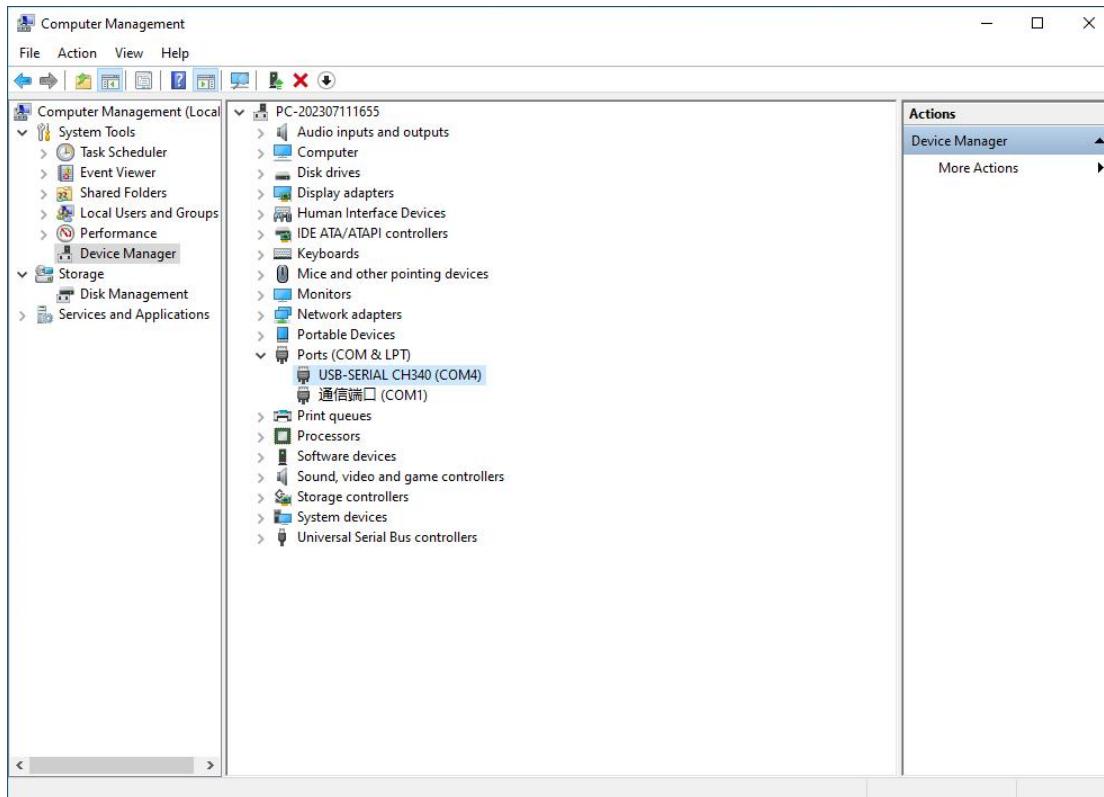
Motor software "Tools"->"Board"->"Arduino/Genuino Uno"



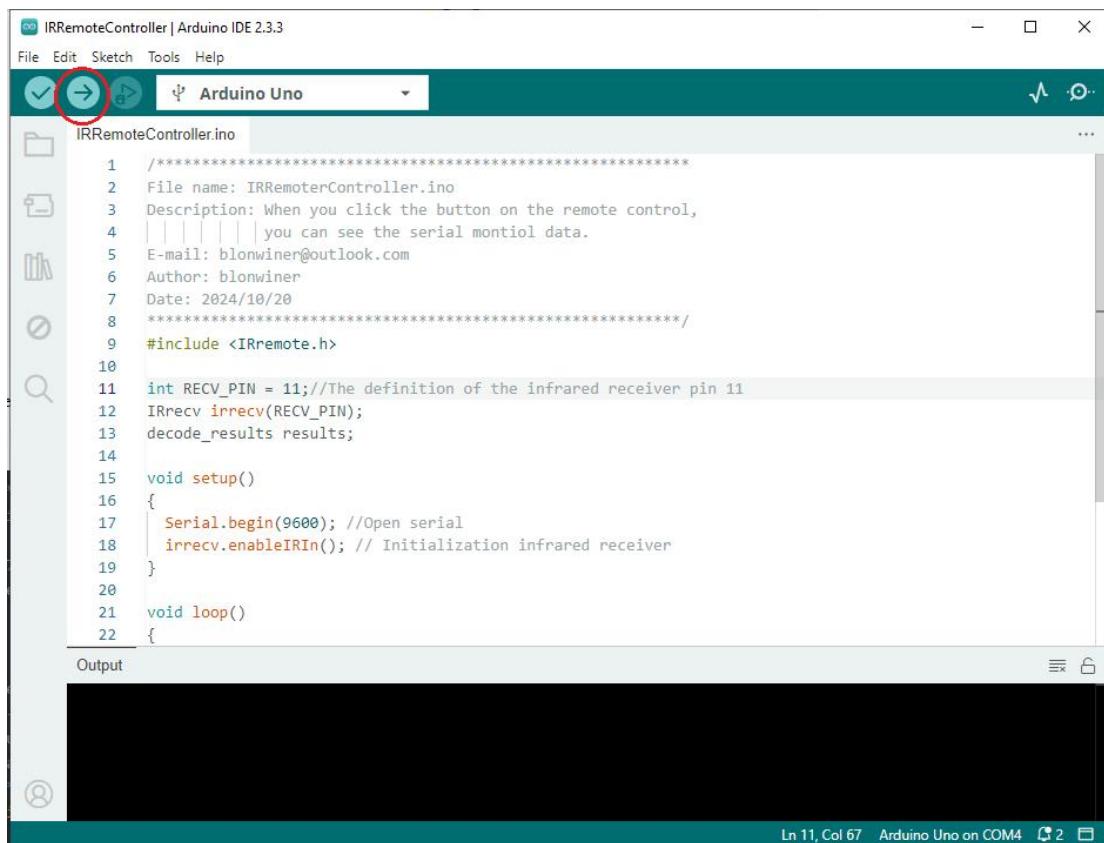
Next, click on "Tools" -> "Port:" -> "COM4" (note that the COM4 here may be recognized differently on different computers, it can be COM1, COM2 or COM3 and so on.)



blonwiner@outlook.com



Click "Upload" to upload the code to UNO, as shown below:



When the software prompts the following information, the code upload is complete.

The screenshot shows the Arduino IDE 2.3.3 interface. The title bar reads "IRRemoteController | Arduino IDE 2.3.3". The toolbar includes icons for file operations, a serial port selector set to "Arduino Uno", and a refresh button. The main area displays the code for "IRRemoteController.ino". The code is a C++ program that includes a header file, defines a receiver pin, initializes the infrared receiver, and sets up the serial connection. It also contains a setup() function that begins the serial port at 9600 bps and enables the infrared receiver, and a loop() function. The output window shows memory usage statistics: "Sketch uses 3192 bytes (9%) of program storage space. Maximum is 32256 bytes. Global variables use 408 bytes (19%) of dynamic memory, leaving 1640 bytes for local variables. Maximum is 2048 bytes". At the bottom, status information indicates "Ln 26, Col 48" and "Arduino Uno on COM4".

```
1  /*****
2  File name: IRRemoteController.ino
3  Description: When you click the button on the remote control,
4  | | | | | | you can see the serial montiol data.
5  E-mail: blonwiner@outlook.com
6  Author: blonwiner
7  Date: 2024/10/20
8  *****/
9  #include <IRremote.h>
10
11 int RECV_PIN = 11;//The definition of the infrared receiver pin 11
12 IRrecv irrecv(RECV_PIN);
13 decode_results results;
14
15 void setup()
16 {
17     Serial.begin(9600); //Open serial
18     irrecv.enableIRIn(); // Initialization infrared receiver
19 }
20
21 void loop()
22 {
```

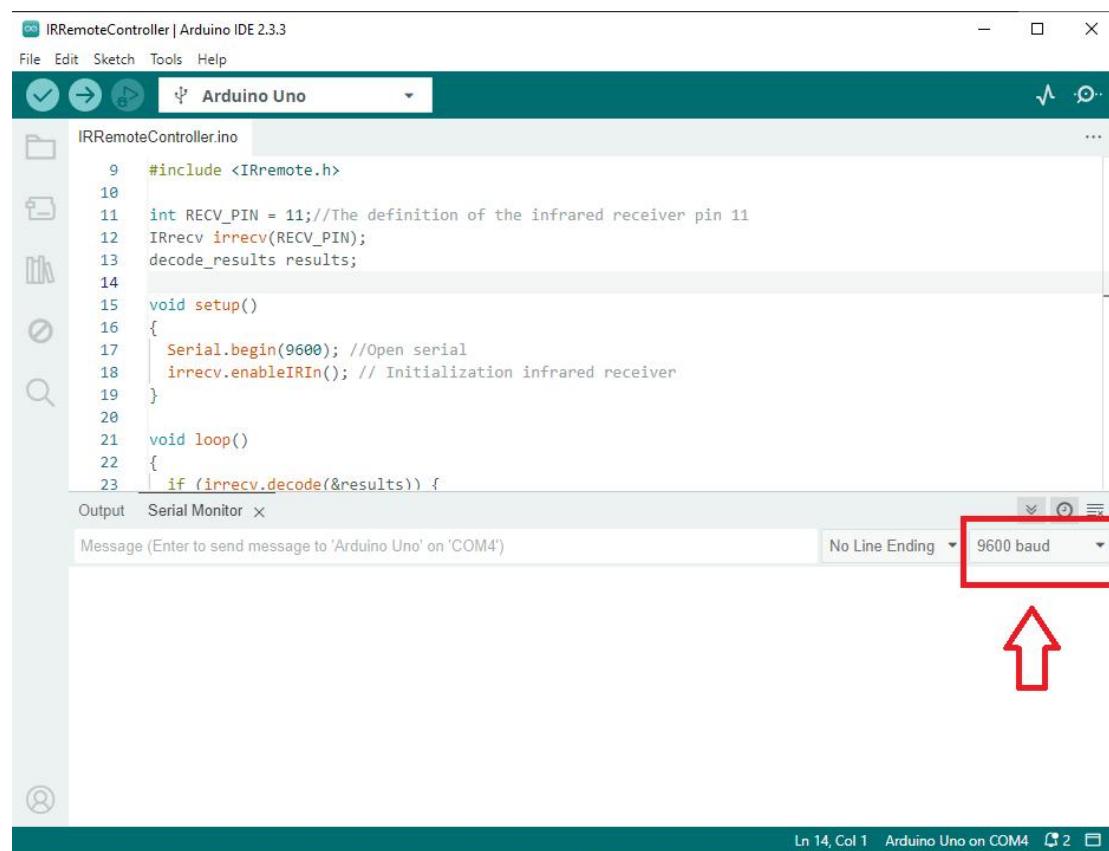
Output

```
Sketch uses 3192 bytes (9%) of program storage space. Maximum is 32256 bytes.
Global variables use 408 bytes (19%) of dynamic memory, leaving 1640 bytes for local variables. Maximum is 2048 bytes.
```

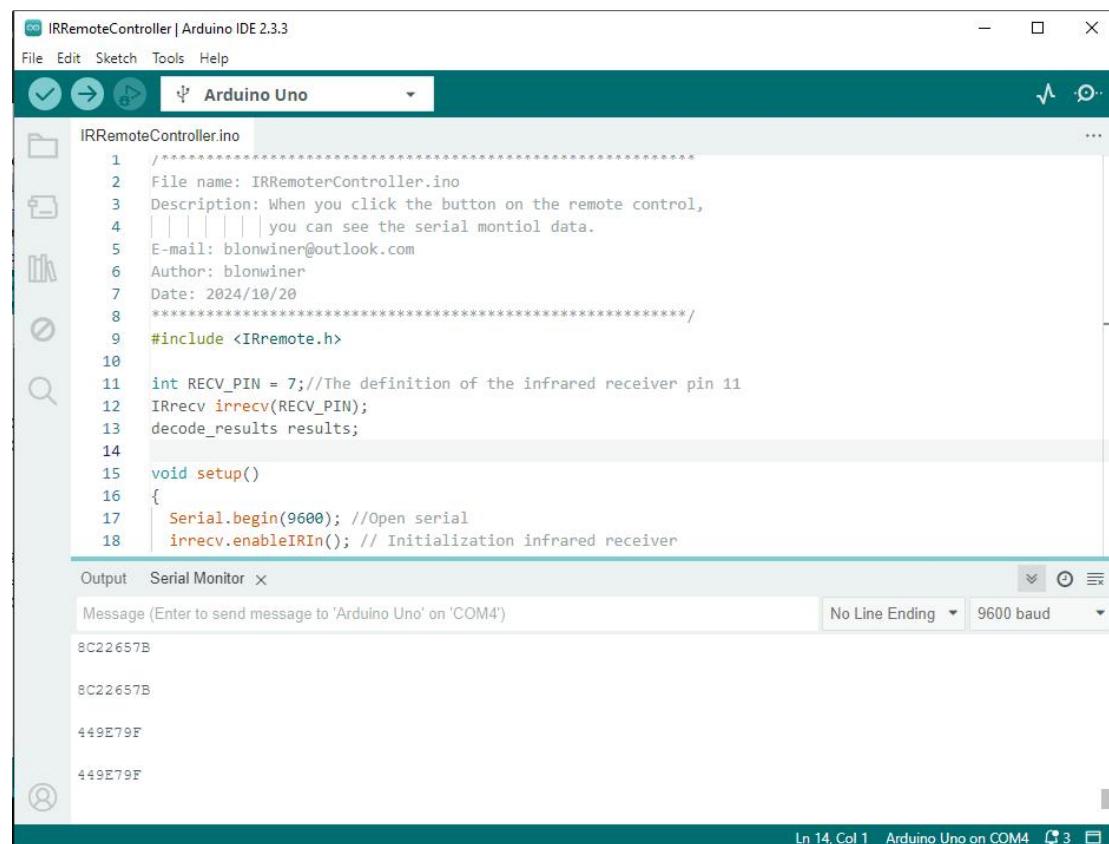
Ln 26, Col 48 Arduino Uno on COM4 2

Compile the program and upload to Arduino UNO board

Now, press a button on the remote controller, and you will see the button number displayed on Serial Monitor.



Now, press a button on the remote controller, and you will see the button number displayed on Serial Monitor.



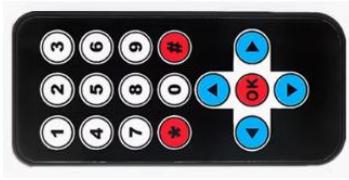
Note: The remote control needs to be facing the signal receiving port of the robotic arm. Other angles of the robot arm will not receive the remote control signal.



Lesson 8 Control Robot by IR Remote

In this lesson, the remote control controls the movement of the robotic arm.

8.1 Components used in this course

Components	Quantity	Picture
BLONWINER Robot	1	
Micro USB Cable	1	
Remote Controller	1	

Operating steps

Firstly open the code ControlRobotbyIRRemote.ino we provide.

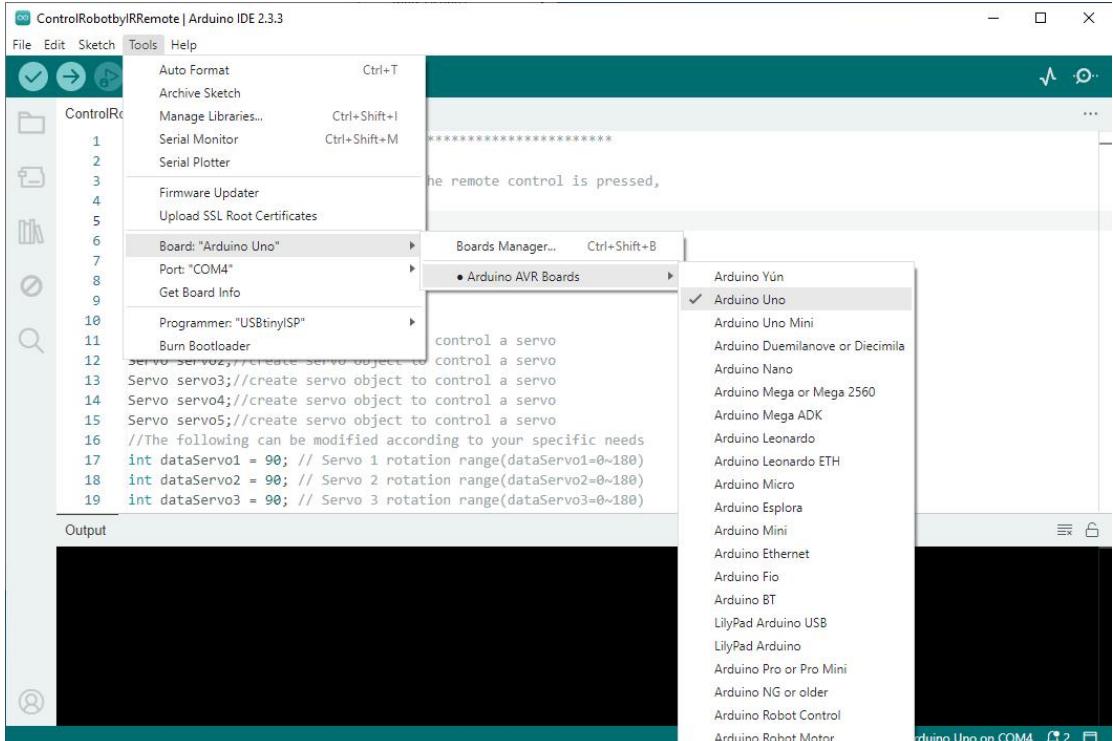
```

1 // ****
2 File name: ControlRobotbyIRRemote.ino
3 Description: Every time a button on the remote control is pressed,
4 | | | | | the robot turns once
5 E-mail: blonwiner@outlook.com
6 Author: blonwiner
7 Date: 2024/10/20
8 ****
9 #include <IRremote.h>
10 #include <Servo.h>
11 Servo servo1;//create servo object to control a servo
12 Servo servo2;//create servo object to control a servo
13 Servo servo3;//create servo object to control a servo
14 Servo servo4;//create servo object to control a servo
15 Servo servo5;//create servo object to control a servo
16 //The following can be modified according to your specific needs
17 int dataServo1 = 90; // Servo 1 rotation range(dataServo1=0~180)
18 int dataServo2 = 90; // Servo 2 rotation range(dataServo2=0~180)
19 int dataServo3 = 90; // Servo 3 rotation range(dataServo3=0~180)

```

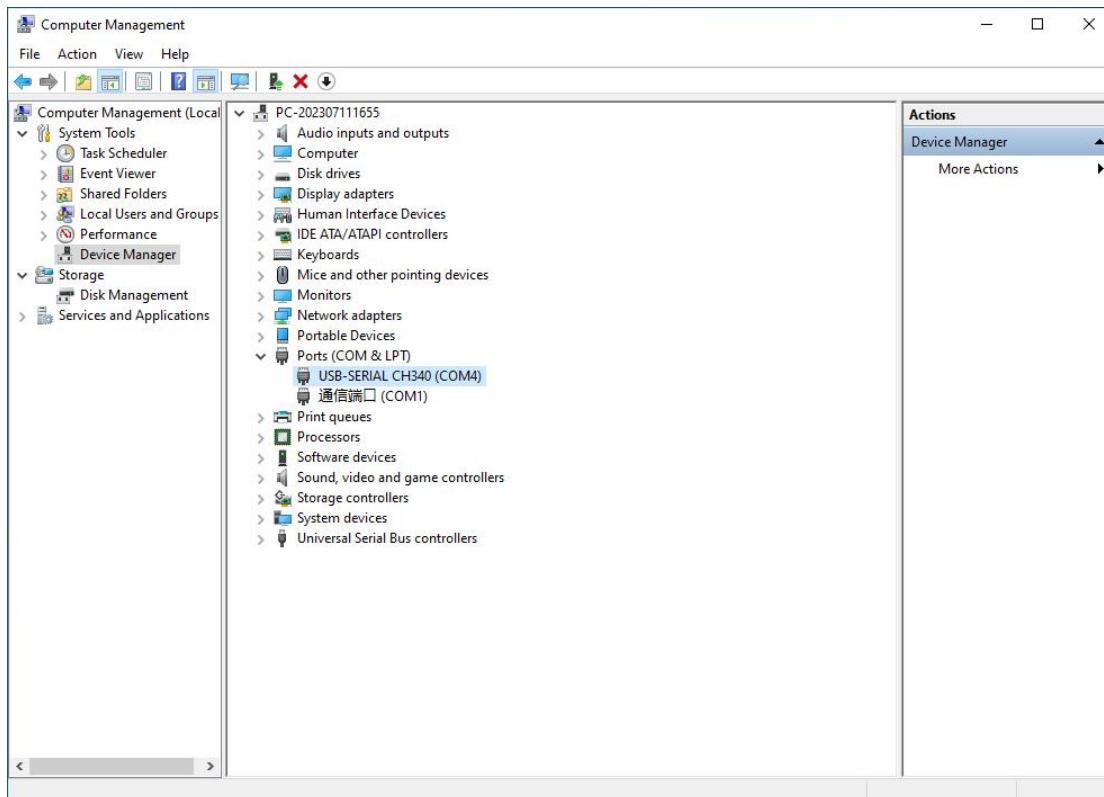
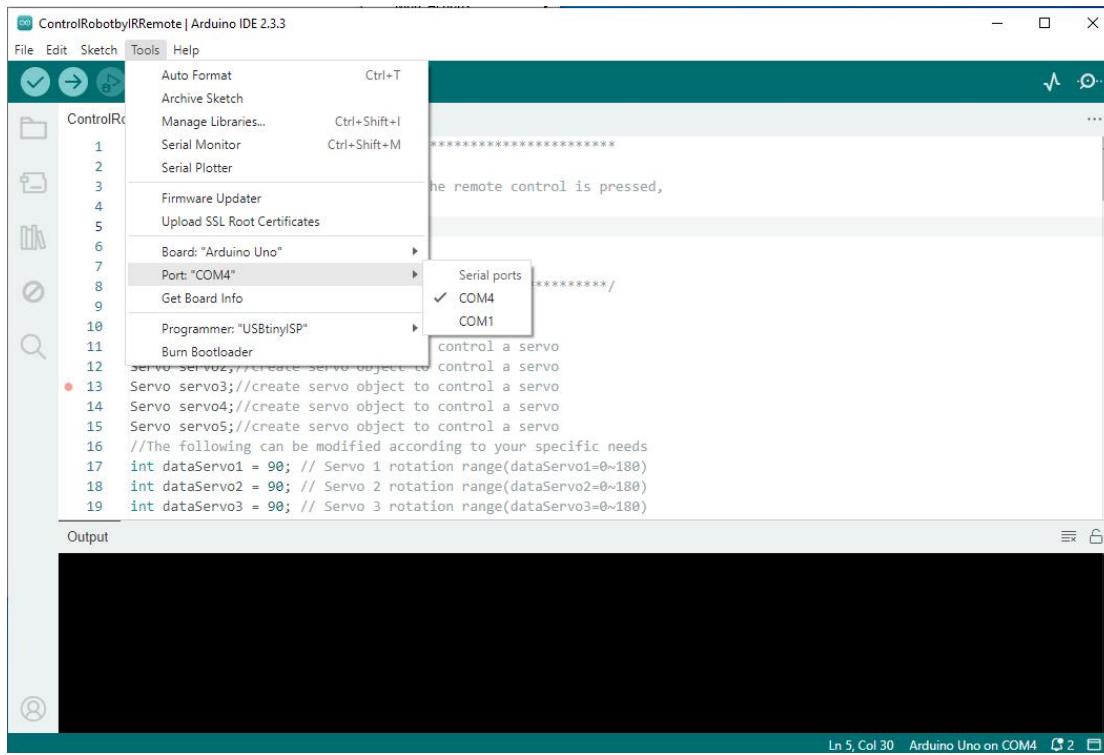
Then connect the robotic arm to the computer with the USB cable. (Note: Do not turn on the power supply to prevent damages of swinging arm. Also pay attention to this in the subsequent operation).

Motor software "Tools"->"Board"->"Arduino/Genuino Uno"



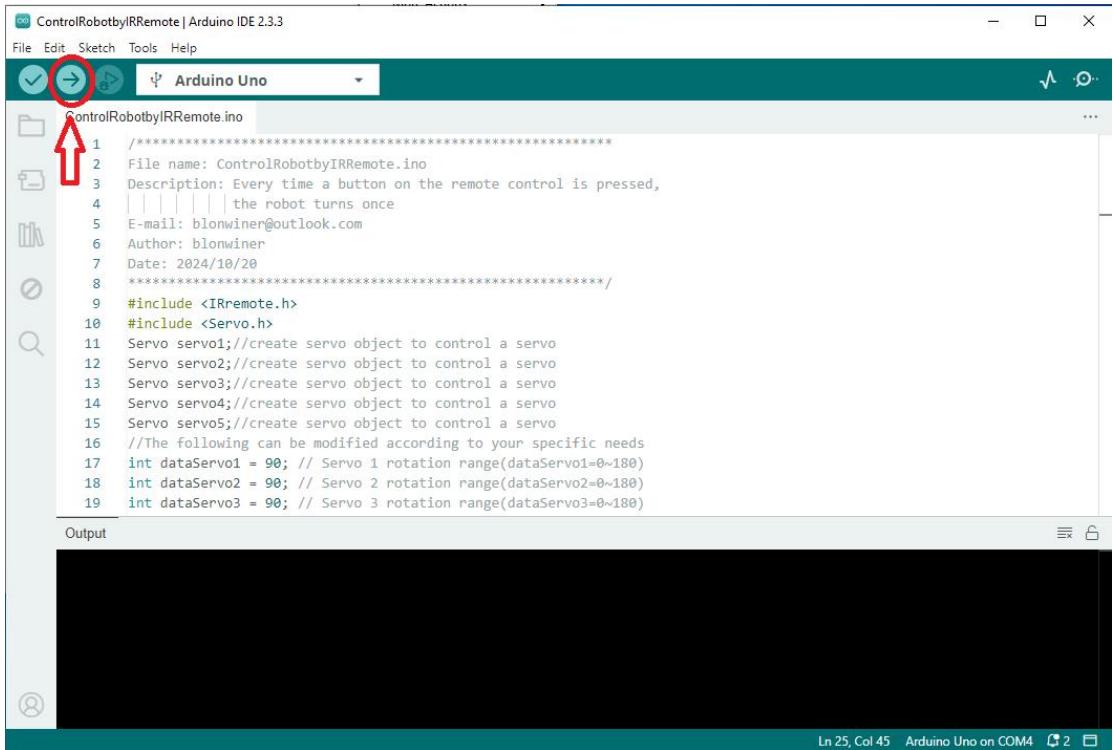
Next, click on "Tools" -> "Port:" -> "COM4" (note that the COM4 here may be recognized differently on different computers, it can be COM1, COM2 or COM3 and so on.)

blonwiner@outlook.com



Click "Upload" to upload the code to UNO, as shown below:

blonwiner@outlook.com



ControlRobotbyIRRemote | Arduino IDE 2.3.3

File Edit Sketch Tools Help

Arduino Uno

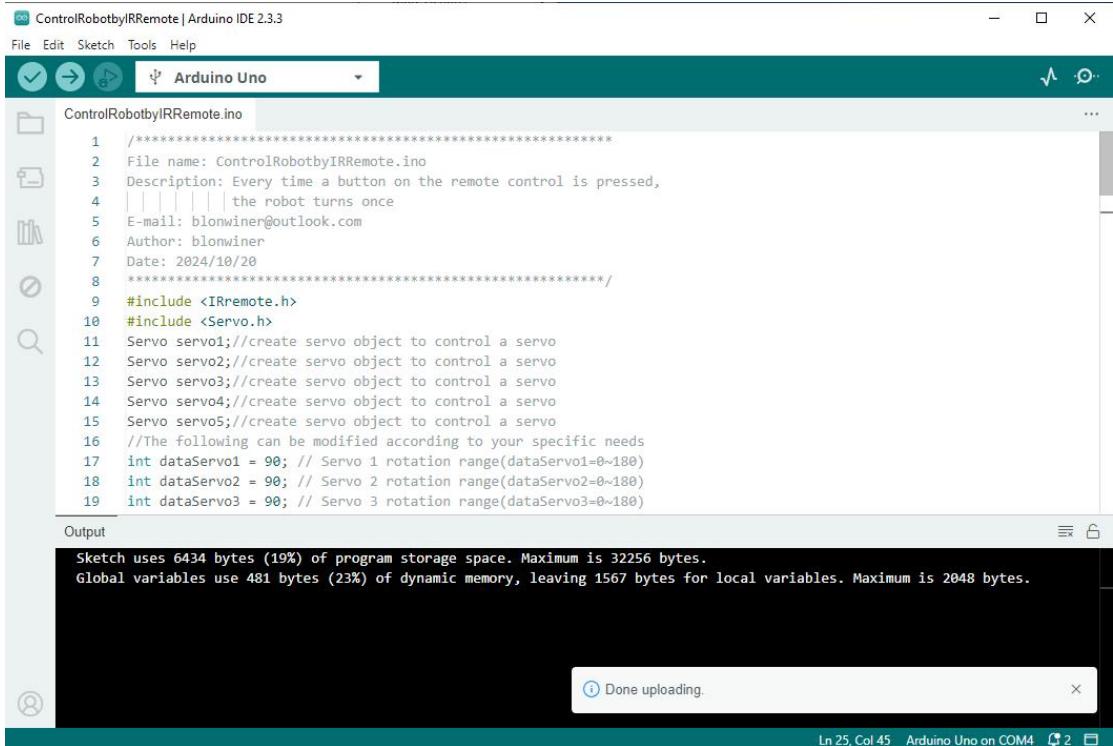
ControlRobotbyIRRemote.ino

```
1 //*****
2 File name: ControlRobotbyIRRemote.ino
3 Description: Every time a button on the remote control is pressed,
4 | | | | | the robot turns once
5 E-mail: blonwiner@outlook.com
6 Author: blonwiner
7 Date: 2024/10/20
8 *****/
9 #include <IRremote.h>
10 #include <Servo.h>
11 Servo servo1;//create servo object to control a servo
12 Servo servo2;//create servo object to control a servo
13 Servo servo3;//create servo object to control a servo
14 Servo servo4;//create servo object to control a servo
15 Servo servo5;//create servo object to control a servo
16 //The following can be modified according to your specific needs
17 int dataServo1 = 90; // Servo 1 rotation range(dataServo1=0~180)
18 int dataServo2 = 90; // Servo 2 rotation range(dataServo2=0~180)
19 int dataServo3 = 90; // Servo 3 rotation range(dataServo3=0~180)
```

Output

Ln 25, Col 45 Arduino Uno on COM4 2

When the software prompts the following information, the code upload is complete.



ControlRobotbyIRRemote | Arduino IDE 2.3.3

File Edit Sketch Tools Help

Arduino Uno

ControlRobotbyIRRemote.ino

```
1 //*****
2 File name: ControlRobotbyIRRemote.ino
3 Description: Every time a button on the remote control is pressed,
4 | | | | | the robot turns once
5 E-mail: blonwiner@outlook.com
6 Author: blonwiner
7 Date: 2024/10/20
8 *****/
9 #include <IRremote.h>
10 #include <Servo.h>
11 Servo servo1;//create servo object to control a servo
12 Servo servo2;//create servo object to control a servo
13 Servo servo3;//create servo object to control a servo
14 Servo servo4;//create servo object to control a servo
15 Servo servo5;//create servo object to control a servo
16 //The following can be modified according to your specific needs
17 int dataServo1 = 90; // Servo 1 rotation range(dataServo1=0~180)
18 int dataServo2 = 90; // Servo 2 rotation range(dataServo2=0~180)
19 int dataServo3 = 90; // Servo 3 rotation range(dataServo3=0~180)
```

Output

Sketch uses 6434 bytes (19%) of program storage space. Maximum is 32256 bytes.
Global variables use 481 bytes (23%) of dynamic memory, leaving 1567 bytes for local variables. Maximum is 2048 bytes.

Done uploading.

Ln 25, Col 45 Arduino Uno on COM4 2

Compile the program and upload to Arduino UNO board

Now, press a button on the remote controller, and the robot turns once.

Note: Each time you press the button, the robot will turn once. If you keep pressing the button, the robot will only turn once

Note: The remote control needs to be facing the signal receiving port of the robotic arm. Other angles of the robot arm will not receive the remote control signal.



blonwiner@outlook.com

If you encounter any problems in the process of learning the robot, please contact us by email.

blonwiner@outlook.com

We would be very happy if you could give me a 5 star review with us.