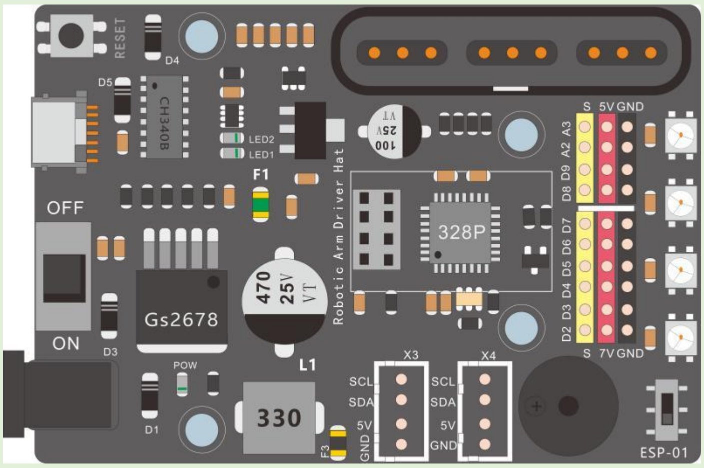
Lesson 4 Test and adjust the angle of the Servo

Table

1. What do you need to prepare

Components	Quantity	Picture
USB cable	1	
MG996 Servo(20KG)	5	
MG996 Servo(25KG)	1	
Robotic Arm Drive Hat	1	 RESET D4 D5 CH340B LED2 LED1 F1 OFF ON D3 Gs2678 470 25V VT 100 25V VT Robotic Arm Driver Hat 328P POW L1 D1 330 F3 X3 X4 SCL SDA 5V GND S 5V GND A3 A2 D9 D8 D7 D6 D5 D4 D3 D2 S 7V GND ESP-01

2. Introduction of the servo

2.1 What's the Servo

Servo motors are commonly referred to as servos, which are small devices with output shafts. When we send a control signal to the server, the output shaft can be turned to a specific position. As long as the control signal remains constant, the servo mechanism will maintain the angle position of the shaft unchanged. If the control signal changes, the position of the output shaft will also change accordingly.

A servo is a type of position servo driver, mainly composed of a casing, circuit board, coreless motor, gears, and position detector. Its working principle is that the receiver or microcontroller sends a signal to the servo, and there is a reference circuit inside it that generates a reference signal with a period of 20ms and a width of 1.5ms. The obtained DC bias voltage is compared with the voltage of the potentiometer to obtain a voltage difference output. The direction of rotation is determined by the IC on the circuit board, and then the non core motor is driven to start rotating. Power is transmitted to the swing arm through the reduction gear, and a signal is sent back by the position detector to determine whether the positioning has been reached. Suitable for control systems that require constant angle changes and can be maintained. When the motor speed is constant, the potentiometer is driven to rotate by a cascaded reduction gear, causing a voltage difference of 0 and the motor to stop rotating. The general rotation angle range of a servo is from 0 degrees to 180 degrees.

In daily life, servos are often used for remote control of airplanes, cars, robots, and other fields. Servo motors are very useful in the field of robotics. Because servo motors have built-in control circuits, although their size is small, their output force is large enough. A standard servo like the Futaba S-148 can provide a torque of 0.3 N/m, which is already powerful enough compared to its external size. Meanwhile, the energy consumed by the servo is directly proportional to the mechanical load. Therefore, a lightweight servo system will not consume too much energy.

2.2 Physical picture

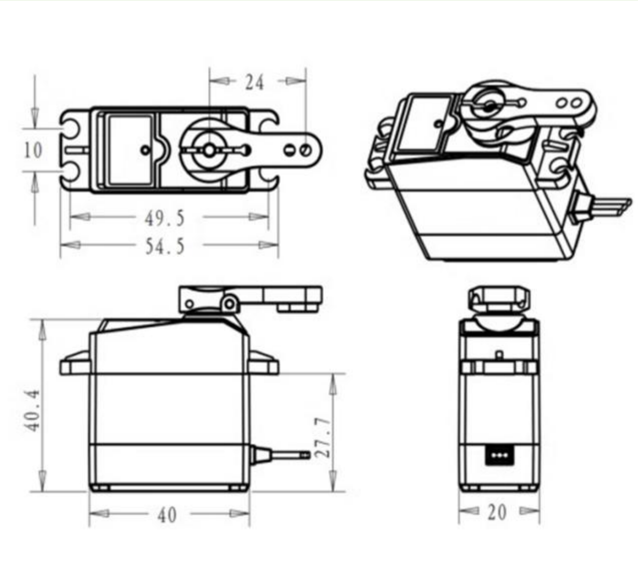
MG996 Servo(20KG)

MG996 Servo(25KG)

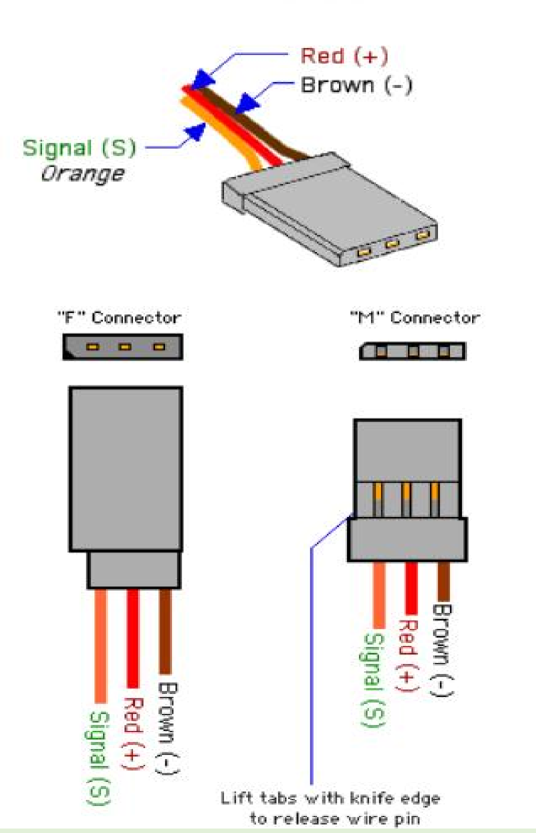
2.3 The Drawings & the wire and plug

The dimensions of MG996 Servo (25KG) and MG996 Servo (20KG) are the same, and the wires and connectors are also the same.

The Drawings:



The wire and plug:



2.4 Parameters

MG996 Servo(20KG)

Products Name:TD-8620MG Digital servo		Products No.	
1. Apply Environmental Conditon			
No.	Item	Specifcation	
1.1	Storage Temperature Range	-30°C～80°C	
1.2	Operating Temperature Range	-25°C～70°C	
2. Standard Test Environment			
No.	Item	Specifcation	
2.1	Temperature Range	-25°C～70°C	
2.2	Humidity Range	65%±10%	
3. Mechanical Specification			
No.	Item	Specifcation	
3.1	Size	40.0*20.0*40.4mm	
3.2	Weight	67g±5%	
3.3	Gear type	5 Metal	
3.4	Limit angle	360°	
3.5	Bearing	2BB	
3.6	Horn gear spline	25T diameter:5.9mm	
3.7	Horn type	Plastic, POM	
3.8	Case	Engineering plastics(PBT)+Aluminum alloy	
3.9	Connector wire	300mm±5mm	
3.10	Motor	Carbon brush motor	
3.11	Splash water resistance	No	
4. Electrical Specification			
No.	Operat voltage	6.0V	8.4V
4.1	Idle current	200mA	220mA
4.2	No load speed	0.19sec/60°	0.14sec/60°
4.3	Runnig current	200mA	220mA
4.4	Peak stall torque	24.2kg.cm	30.5kg.cm
4.5	Stall current	2200mA±10%	2900mA±10%
4.6	Working voltage range	6.0-8.4V	
5.Control Specification			
No.	Item	Specifcation	
5.1	Command signal	Pulse width modification	
5.2	Amplifier type	Digital controller	
5.3	Pulse width range	500～2500usec	
5.4	Neutral position	1500usec	
5.5	Running degree	180°±3°(when 500～2500usec)	

5.6	Dead band width	5 usec	
5.7	Rotating direction	Counterclockwise (when 1000～2000usec)	

MG996 Servo(25KG)

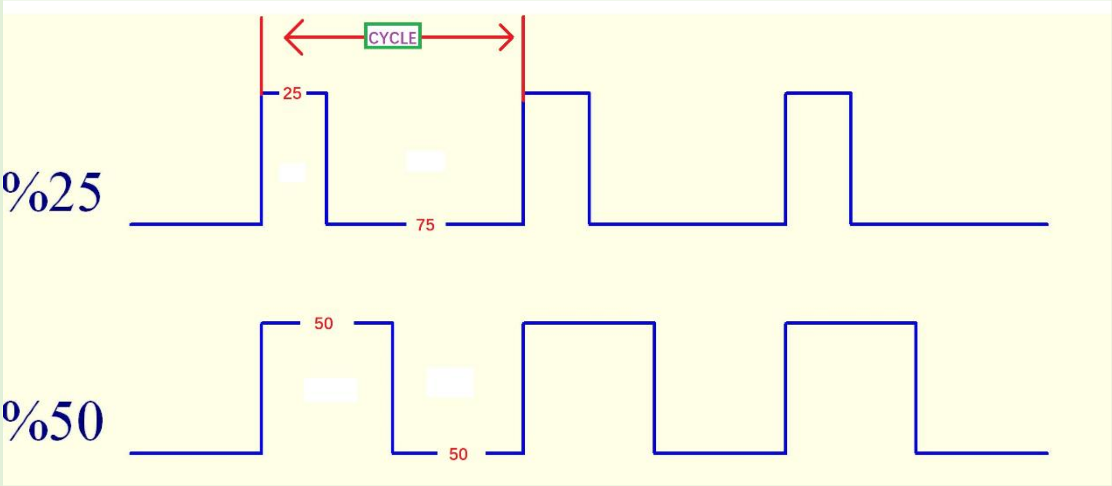
Products Name:TD-8625MG Digital servo		Products No.	
1. Apply Environmental Conditon			
No.	Item	Specifcation	
1.1	Storage Temperature Range	-30°C～80°C	
1.2	Operating Temperature Range	-25°C～70°C	
2. Standard Test Environment			
No.	Item	Specifcation	
2.1	Temperature Range	-25°C～70°C	
2.2	Humidity Range	65%±10%	
3. Mechanical Specification			
No.	Item	Specifcation	
3.1	Size	40.0*20.0*40.4mm	
3.2	Weight	67g±5%	
3.3	Gear type	5 Metal	
3.4	Limit angle	360°	
3.5	Bearing	2BB	
3.6	Horn gear spline	25T diameter:5.9mm	
3.7	Horn type	Plastic, POM	
3.8	Case	Engineering plastics(PBT)+Aluminum alloy	
3.9	Connector wire	300mm±5mm	
3.10	Motor	Carbon brush motor	
3.11	Splash water resistance	No	
4. Electrical Specification			
No.	Operat voltage	6.0V	8.4V
4.1	Idle current	200mA	220mA
4.2	No load speed	0.19sec/60°	0.14sec/60°
4.3	Runnig current	200mA	220mA
4.4	Peak stall torque	25.2kg.cm	31.5kg.cm
4.5	Stall current	2200mA±10%	2900mA±10%
4.6	Working voltage range	6.0-8.4V	
5.Control Specification			
No.	Item	Specifcation	
5.1	Command signal	Pulse width modification	
5.2	Amplifier type	Digital controller	
5.3	Pulse width range	500～2500usec	
5.4	Neutral position	1500usec	

5.5	Running degree	180°±3°(when 500～2500usec)	
5.6	Dead band width	5 usec	
5.7	Rotating direction	Counterclockwise (when 1000～2000usec)	

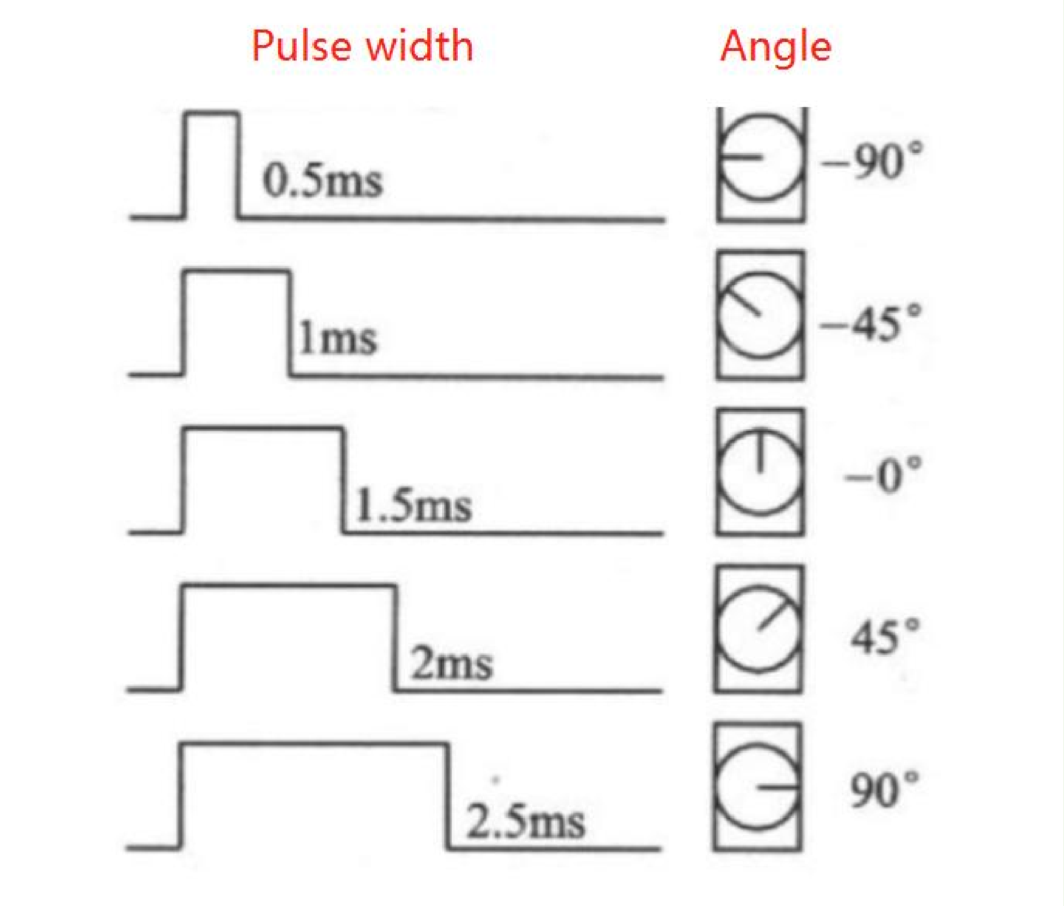
2.5 Working Principle

The control signal of the servo is a PWM signal with a period of 20ms, in which the pulse width is from 0.5ms-2.5ms, and the corresponding position of the servo is 0-180 degrees, which changes linearly.Provide it with a certain pulse width, and its output shaft will remaster at a corresponding angle, no matter how the external torque changes, until a new pulse signal of different width is provided to it, it will change the output angle to the new corresponding position. There is a reference voltage inside the servo, which generates a reference signal with a period of 20ms and a width of 1.5ms. There is a comparator that compares the applied signal with the reference signal to determine the direction and size, thereby generating the rotation signal of the motor. The internal control circuit board of the servo receives the control signal from the signal line, and controls the rotation of the motor. The motor drives a series of gear sets, which are driven to the output steering wheel after deceleration. The output shaft of the servo is connected to the position feedback potentiometer. When the steering wheel rotates, it drives the position feedback potentiometer. The potentiometer will output a voltage signal to the control circuit board for position feedback. The control circuit board determines the rotation direction and speed of the motor according to the position of the output shaft, so that the output shaft stops when it reaches the target.

Pulse width modulation (PWM) refers to the use of the digital output of a microprocessor to control an analog circuit, and is a method of digitally encoding the level of an analog signal.
PWM (Pulse Width Modulation): Pulse Width Modulation
Pulse: square wave, frequency (freq)
Width: the width of the high level, the duty cycle (duty)
Cycle: CYCLE
Duty cycle: the proportion of high level (100&%)
Duty cycle static diagram:



The relationship between the output angle of MG996 Servo and the pulse width of the input signal



CYCLE=20ms

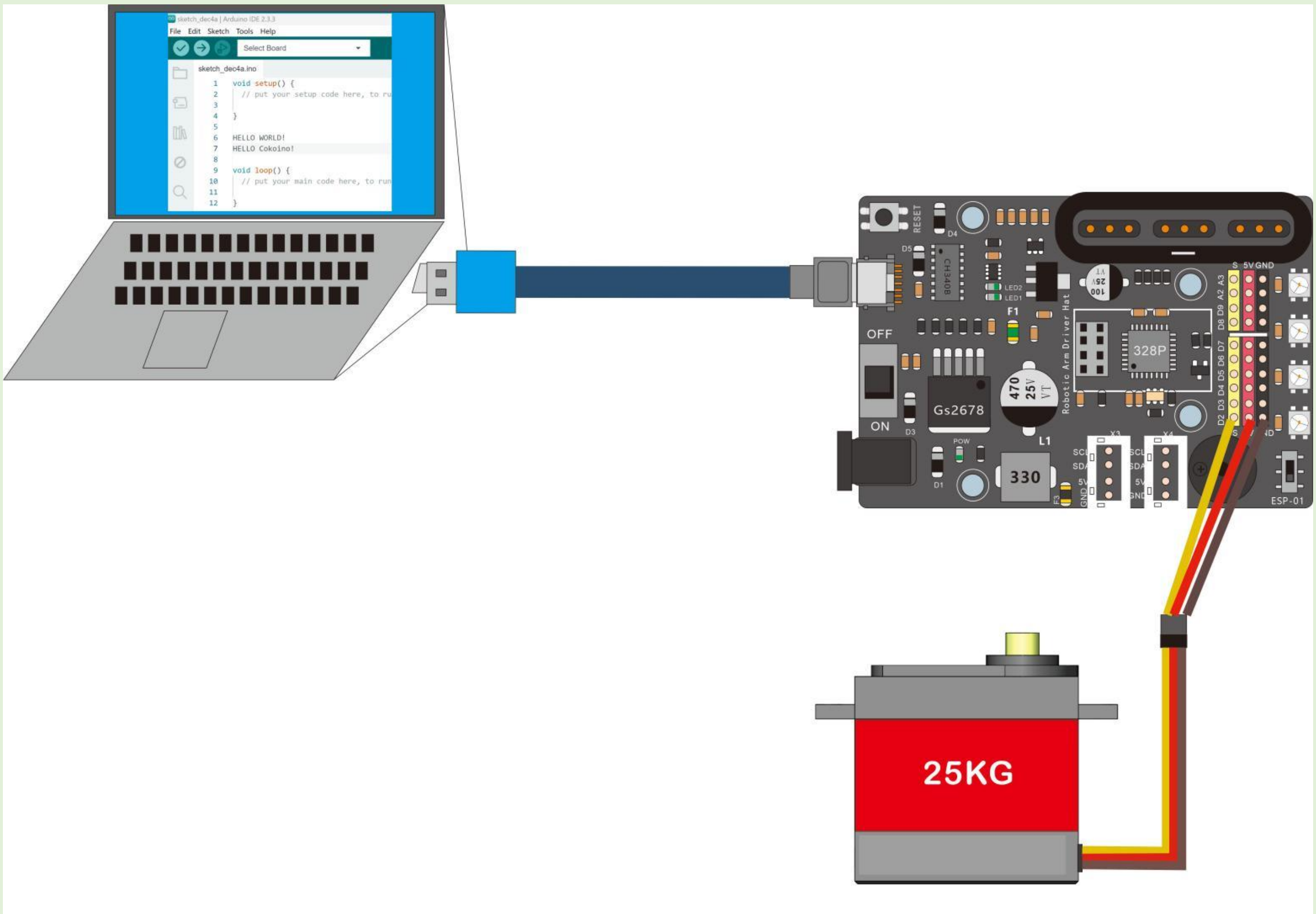
Duty_16=65532*Pulse Width/CYCLE

The relationship between the corresponding pulse width and duty_u16:

Pulse Width(ms)	Duty_16
0.5	1638
1	3276
1.5	4914
2	6552
2.5	8192

3. Hardware Connection Circuit

Taking MG996 Servo (25KG) as an example, the signal pin of the tail plug of the servo wire is connected to the D2 pin of the control board. When the servo is not loaded, it can be powered by the computer with a USB cable.Of course, you can also plug in an adapter to power the control board.



```
void setup() {
  // put your setup code here, to ru

}

HELLO WORLD!
HELLO Cokoino!

void loop() {
  // put your main code here, to run

}
```

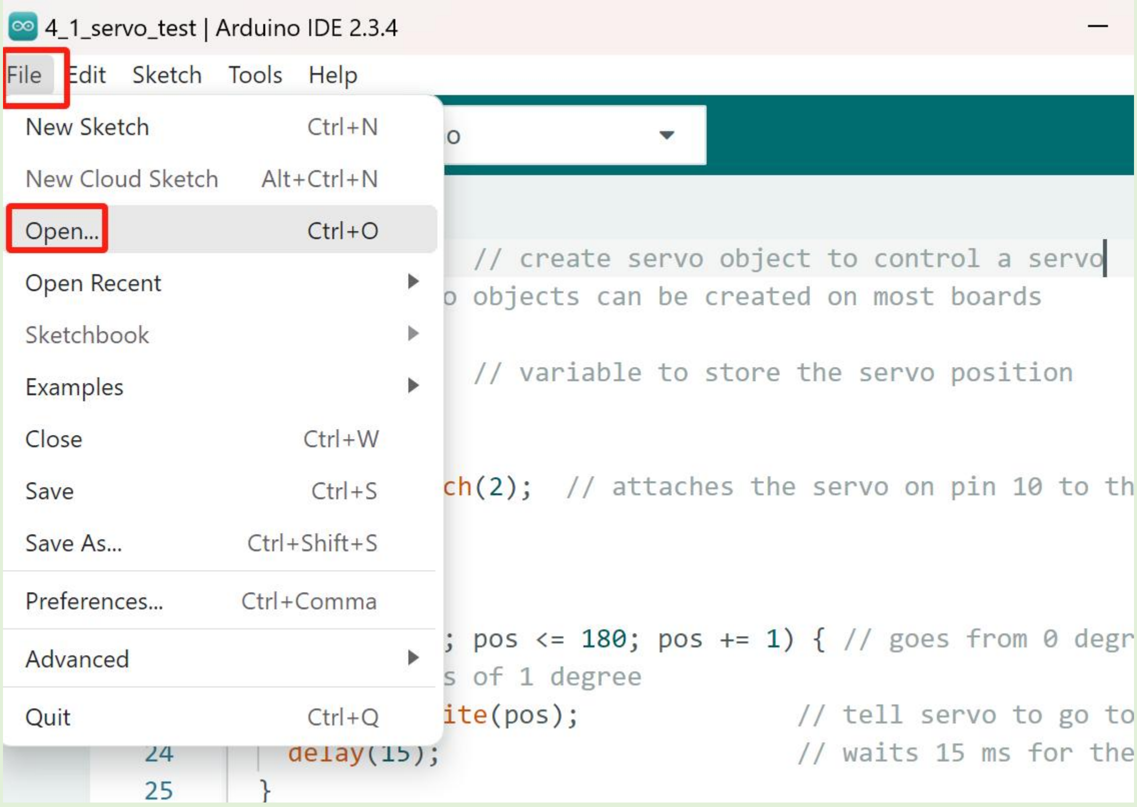
Wiring between the servo and the control board	
Connector of the Servo	Connector of the control board
+	7V
-	GND
Signal	D2

4. Upload the code and test

The code used in this lesson is placed in this folder：“ E:\CKK0017-main\Tutorial\sketches”

4.1 Double-click the Arduino IDE shortcut on the desktop to open it

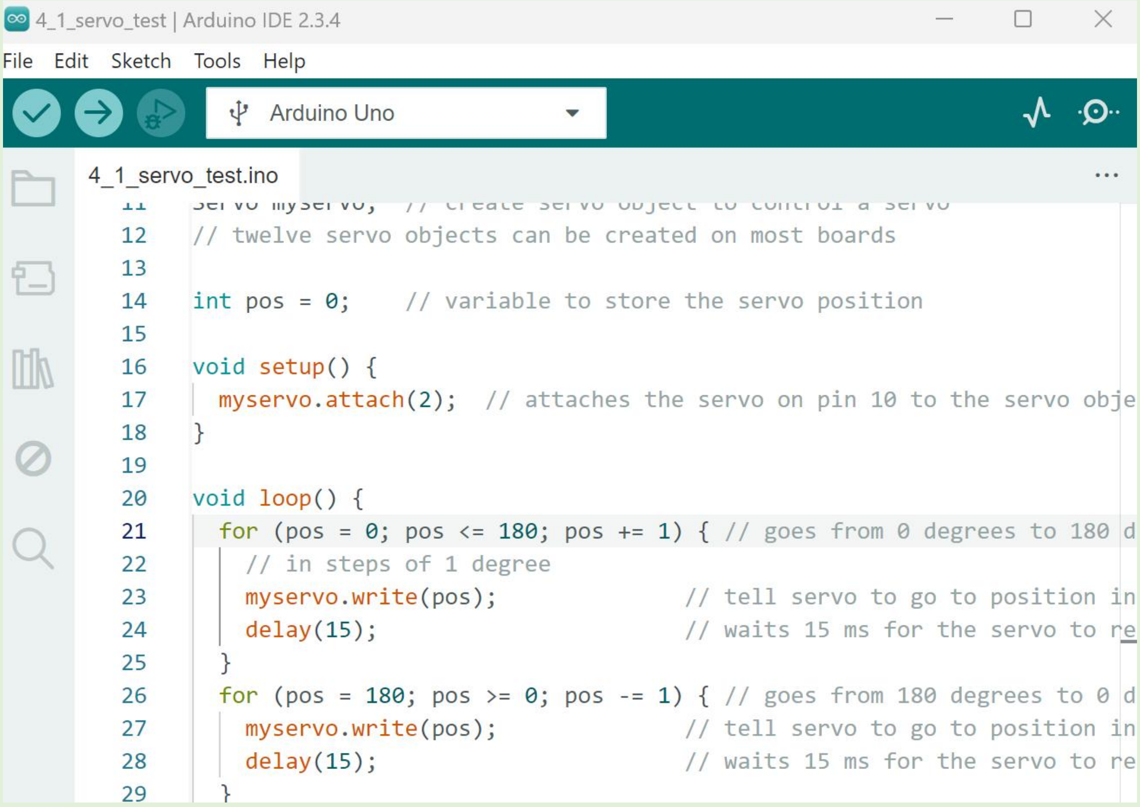
4.2 Click "File" --- "open"



```
 // create servo object to control a servo
o objects can be created on most boards

 // variable to store the servo position

ch(2);  // attaches the servo on pin 10 to th

; pos <= 180; pos += 1) { // goes from 0 degr
s of 1 degree
ite(pos);             // tell servo to go to
  delay(15);           // waits 15 ms for the
}
```

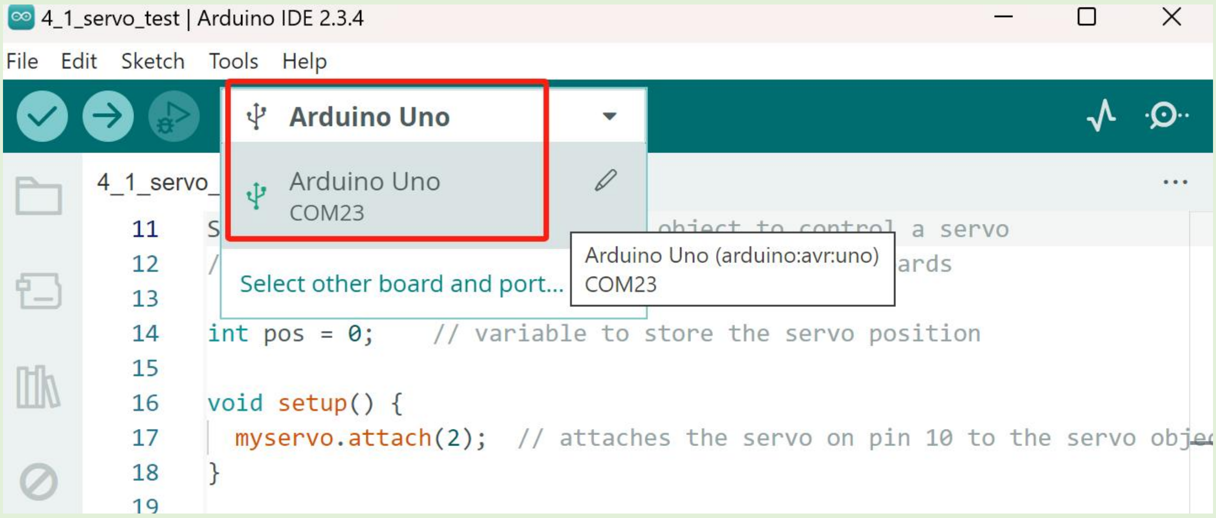
4.3 Select the code in the folder named 4_1_servo_test：

E:\CKK0017-main\Tutorial\sketches\4_1_servo_test. Click "open"


4_1_servo_test | Arduino IDE 2.3.4
File Edit Sketch Tools Help
Arduino Uno
4_1_servo_test.ino

```
Servo myservo;  // create servo object to control a servo
// twelve servo objects can be created on most boards

int pos = 0;    // variable to store the servo position

void setup() {
  myservo.attach(2);  // attaches the servo on pin 10 to the servo obje
}

void loop() {
  for (pos = 0; pos <= 180; pos += 1) { // goes from 0 degrees to 180 d
    // in steps of 1 degree
    myservo.write(pos);              // tell servo to go to position in
    delay(15);                       // waits 15 ms for the servo to re
  }
  for (pos = 180; pos >= 0; pos -= 1) { // goes from 180 degrees to 0 d
    myservo.write(pos);              // tell servo to go to position in
    delay(15);                       // waits 15 ms for the servo to re
  }
```

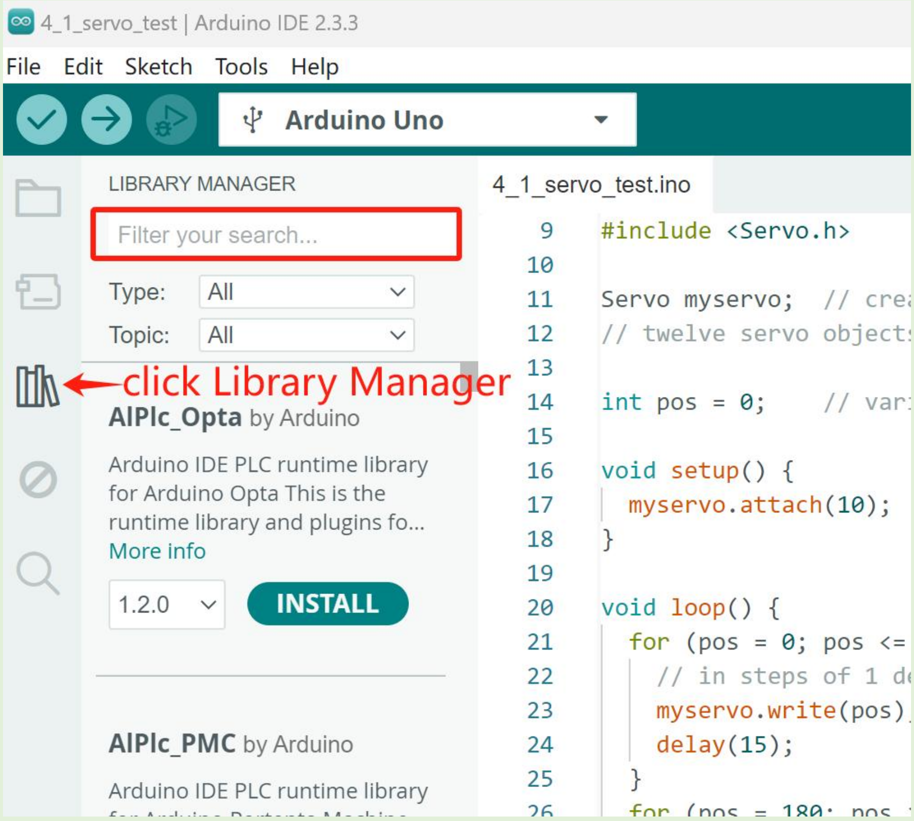
4.4 Select the board "Arduino UNO"and Port "COM23" (COM port is commonly known as an input output port for a device normally PC which enables communication between Arduino and PC. You can check your arduino com number in device manager, the com port of our arduino board is recognized as COM23 in this tutorial)


4_1_servo_test | Arduino IDE 2.3.4
File Edit Sketch Tools Help
Arduino Uno
Arduino Uno
COM23
Select other board and port...
Arduino Uno (arduino:avr:uno)
COM23
4_1_servo_

```
Servo myservo;  // create servo object to control a servo
// twelve servo objects can be created on most boards

int pos = 0;    // variable to store the servo position

void setup() {
  myservo.attach(2);  // attaches the servo on pin 10 to the servo obje
}
```

4.5 Install library Servo.h

Method 1: Directly click "Library Manager" on the left side of the Arduino IDE interface, the interface is as follows



AIPlc_Opta by Arduino

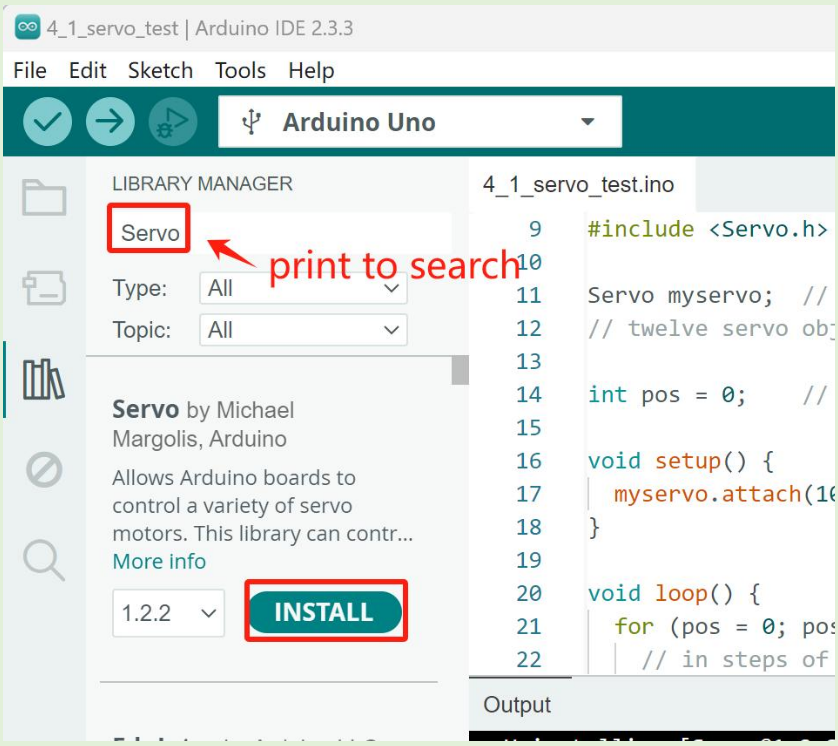
Arduino IDE PLC runtime library for Arduino Opta This is the runtime library and plugins fo...

AIPlc_PMC by Arduino

Arduino IDE PLC runtime library

```
#include <Servo.h>

Servo myservo;  // cre
// twelve servo object

int pos = 0;    // var

void setup() {
  myservo.attach(10);
}

void loop() {
  for (pos = 0; pos <=
    // in steps of 1 de
    myservo.write(pos)
    delay(15);
  }
```

Search for Servo in the red search box, Library Manager will quickly match related library files

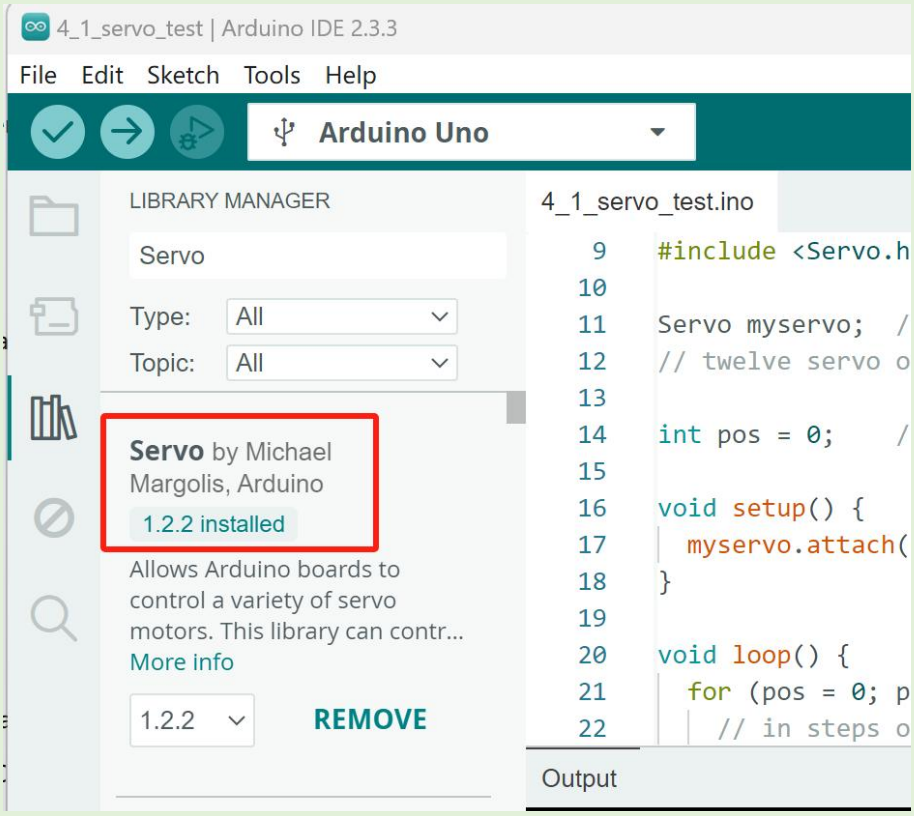


Servo by Michael Margolis, Arduino

Allows Arduino boards to control a variety of servo motors. This library can contr...

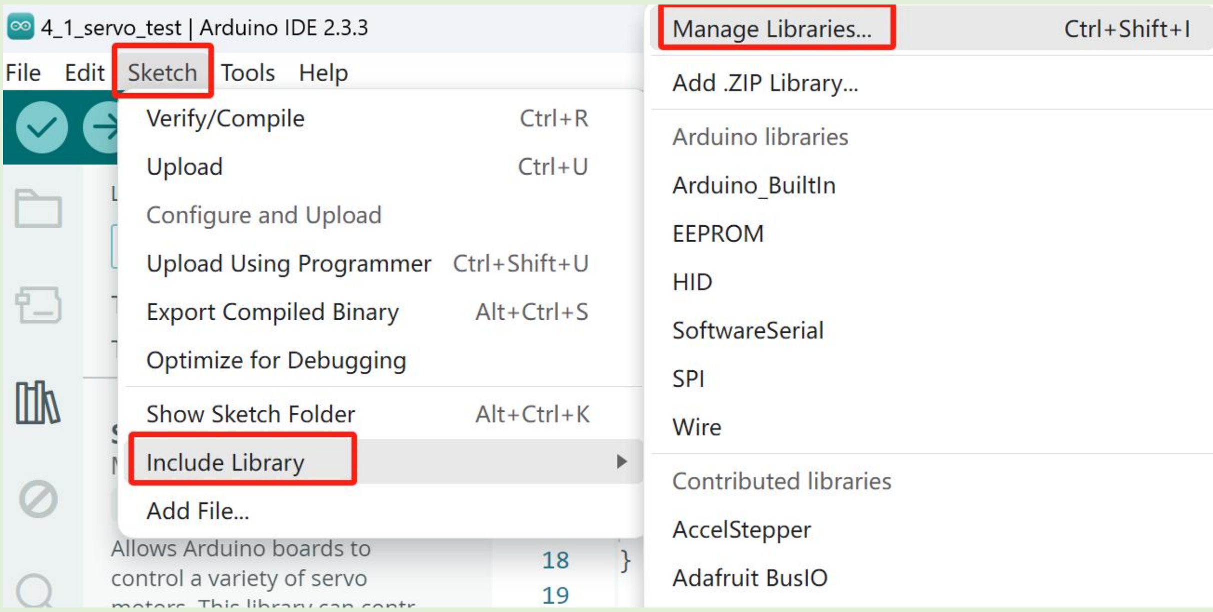
```
#include <Servo.h>

Servo myservo;  //
// twelve servo obj

int pos = 0;    //

void setup() {
  myservo.attach(1
}

void loop() {
  for (pos = 0; pos
    // in steps of
```

Click "INSTALL" to install the library file of Servo. After the installation is complete, it will display "installed"

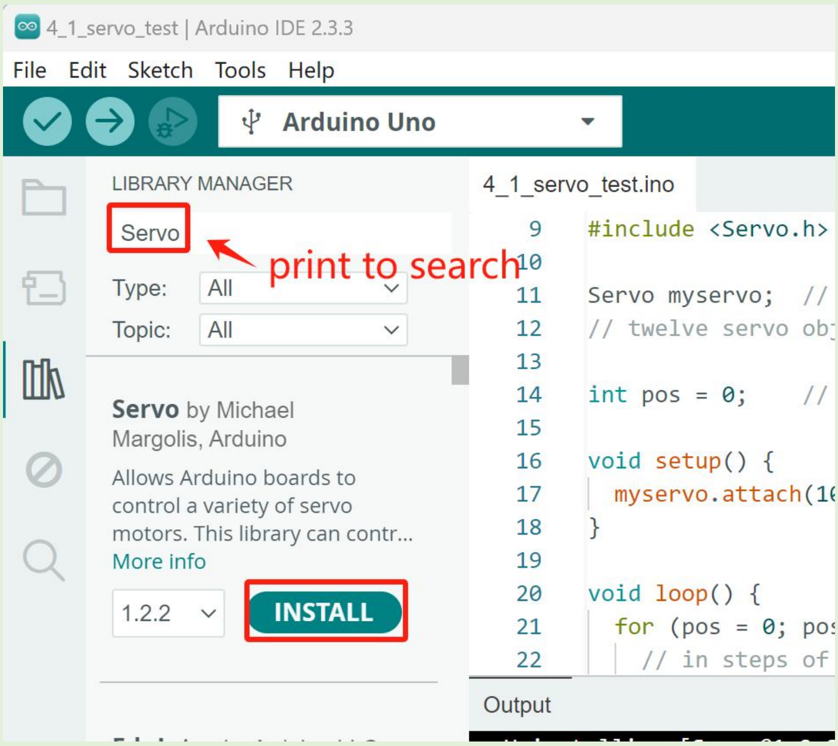

4_1_servo_test | Arduino IDE 2.3.3
File Edit Sketch Tools Help
Arduino Uno
LIBRARY MANAGER
Servo
Type: All
Topic: All
Servo by Michael Margolis, Arduino
1.2.2 installed
Allows Arduino boards to control a variety of servo motors. This library can contr...
More info
1.2.2
REMOVE
4_1_servo_test.ino
Output

```
#include <Servo.h

Servo myservo;  /
// twelve servo o

int pos = 0;    /

void setup() {
  myservo.attach(
}

void loop() {
  for (pos = 0; p
    // in steps o
```

Method 2: After opening the 4_1_servo_test code, click "Sketch" --- "Include Library" --- "Manage Libraries..." to open the library management interface.


4_1_servo_test | Arduino IDE 2.3.3
File Edit Sketch Tools Help
Verify/Compile Ctrl+R
Upload Ctrl+U
Configure and Upload
Upload Using Programmer Ctrl+Shift+U
Export Compiled Binary Alt+Ctrl+S
Optimize for Debugging
Show Sketch Folder Alt+Ctrl+K
Include Library
Add File...
Allows Arduino boards to control a variety of servo
Manage Libraries... Ctrl+Shift+I
Add .ZIP Library...
Arduino libraries
Arduino_BuiltIn
EEPROM
HID
SoftwareSerial
SPI
Wire
Contributed libraries
AccelStepper
Adafruit BusIO

Search for Servo in the red search box, Library Manager will quickly match related library files



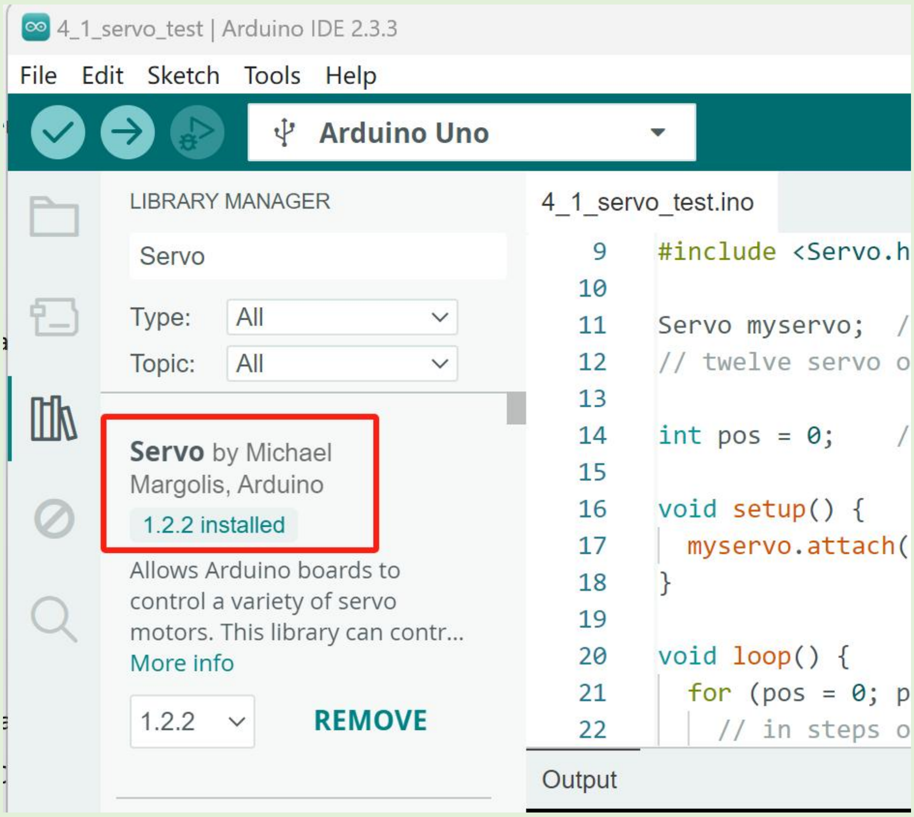
Servo

Servo by Michael Margolis, Arduino

Allows Arduino boards to control a variety of servo motors. This library can contr...
More info

```
#include <Servo.h>

Servo myservo;  //
// twelve servo obj

int pos = 0;    //

void setup() {
  myservo.attach(1
}

void loop() {
  for (pos = 0; pos
    // in steps of
```

Click "INSTALL" to install the library file of Servo. After the installation is complete, it will display "installed"



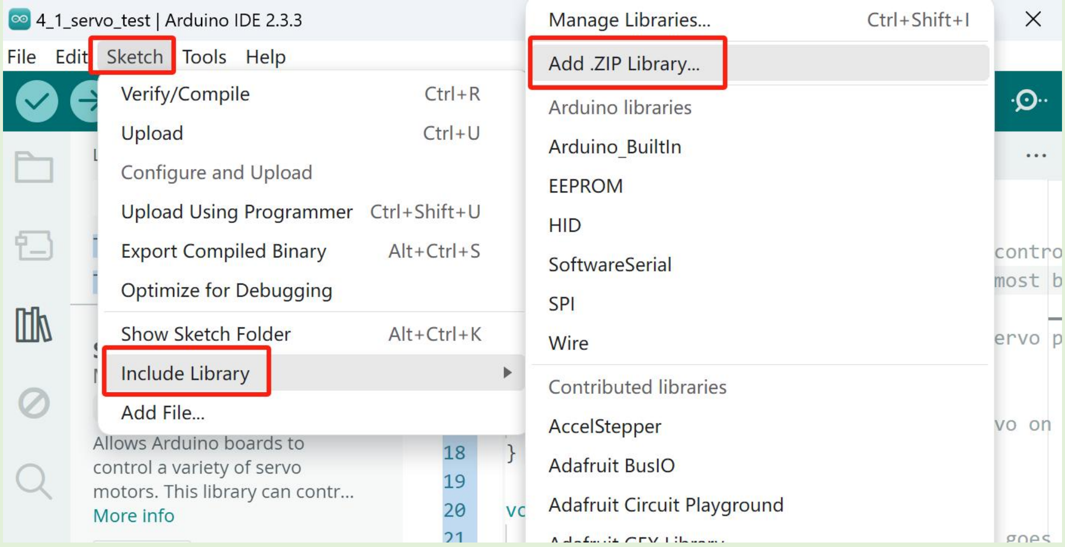
Servo

Servo by Michael Margolis, Arduino
1.2.2 installed

Allows Arduino boards to control a variety of servo motors. This library can contr...
More info

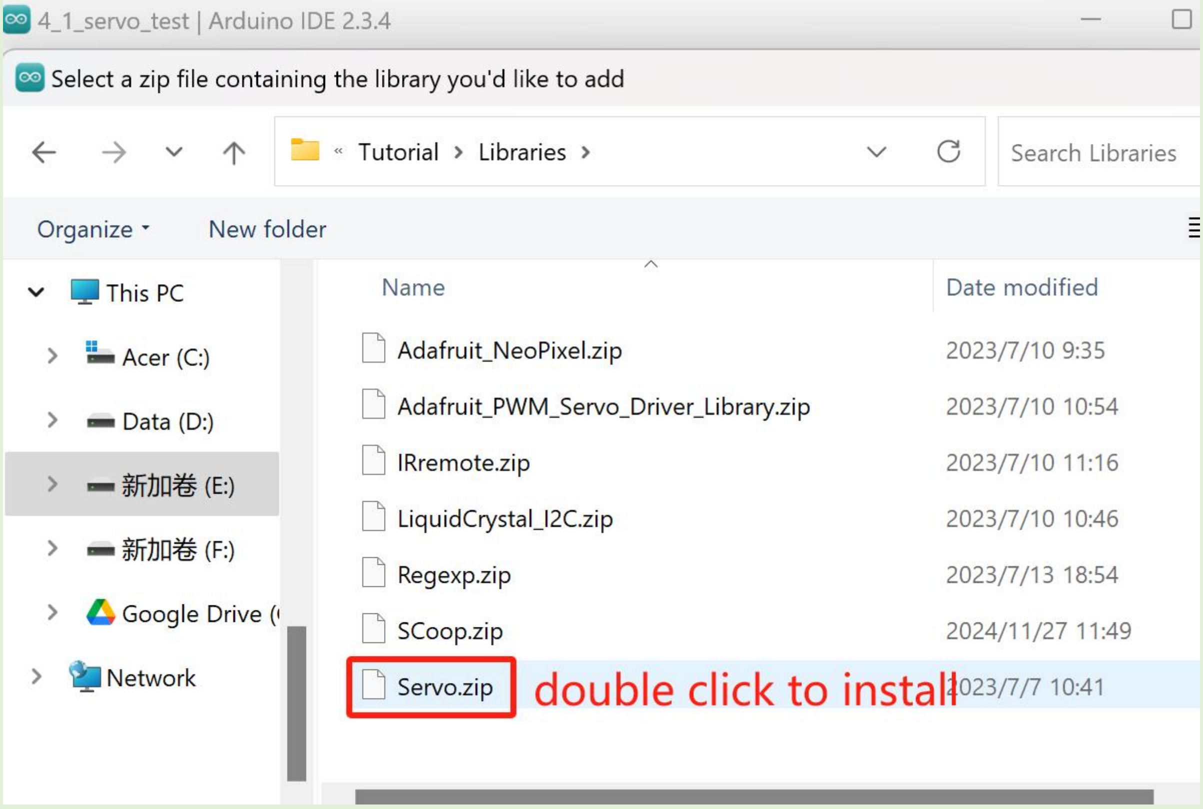
```
#include <Servo.h

Servo myservo;  /
// twelve servo o

int pos = 0;    /

void setup() {
  myservo.attach(
}

void loop() {
  for (pos = 0; p
    // in steps o
```

Method 3： After opening the 2_1_servo_test code, click "Sketch" ---“Include Library”---“Add.ZIP Library…”

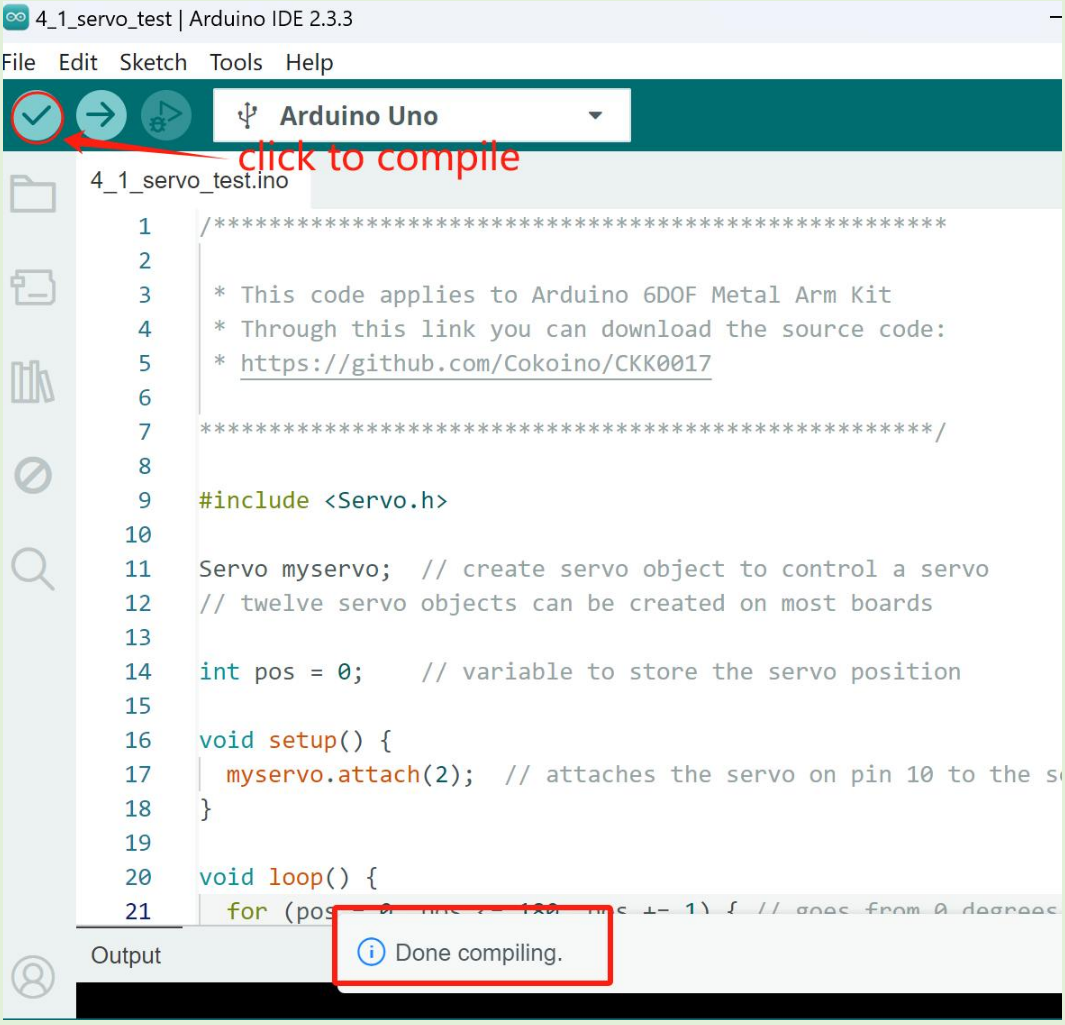

4_1_servo_test | Arduino IDE 2.3.3
File
Edit
Sketch
Tools
Help
Verify/Compile Ctrl+R
Upload Ctrl+U
Configure and Upload
Upload Using Programmer Ctrl+Shift+U
Export Compiled Binary Alt+Ctrl+S
Optimize for Debugging
Show Sketch Folder Alt+Ctrl+K
Include Library
Add File...
Manage Libraries... Ctrl+Shift+I
Add .ZIP Library...
Arduino libraries
Arduino_BuiltIn
EEPROM
HID
SoftwareSerial
SPI
Wire
Contributed libraries
AccelStepper
Adafruit BusIO
Adafruit Circuit Playground
Allows Arduino boards to control a variety of servo motors. This library can contr...
More info

Find the Servo.ZIP file in E:CKK0017-main\Tutorial\Libraries\Servo.zip"

Double click the Servo.ZIP file,then it will be installed into Arduino IDE


4_1_servo_test | Arduino IDE 2.3.4
Select a zip file containing the library you'd like to add
« Tutorial › Libraries ›
Search Libraries
Organize
New folder
This PC
Acer (C:)
Data (D:)
新加卷 (E:)
新加卷 (F:)
Google Drive (
Network
Name
Date modified
Adafruit_NeoPixel.zip 2023/7/10 9:35
Adafruit_PWM_Servo_Driver_Library.zip 2023/7/10 10:54
IRremote.zip 2023/7/10 11:16
LiquidCrystal_I2C.zip 2023/7/10 10:46
Regexp.zip 2023/7/13 18:54
SCoop.zip 2024/11/27 11:49
Servo.zip 2023/7/7 10:41
double click to install

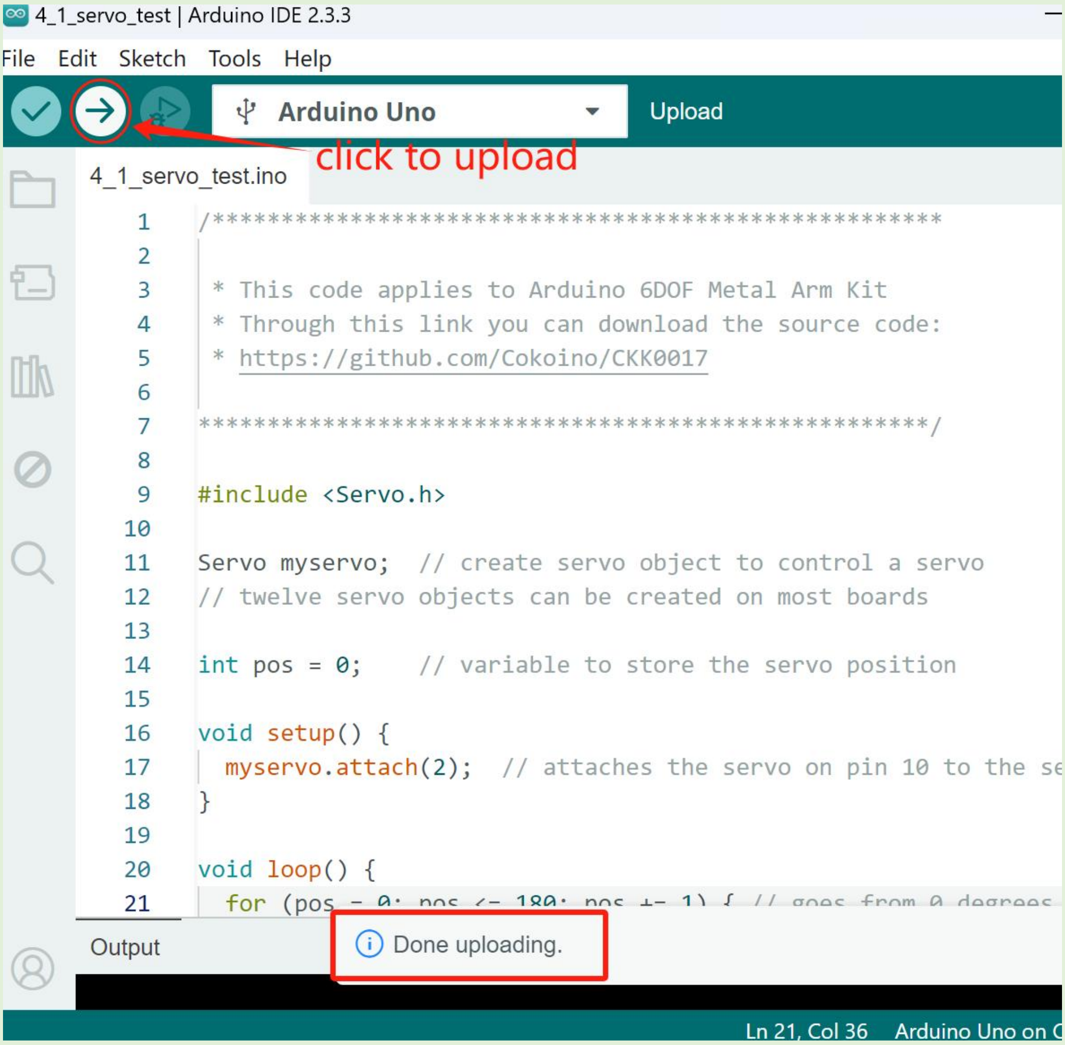
4.6 Click compile button ✓ , successfully compiled the code will display “Done compiling”


4_1_servo_test | Arduino IDE 2.3.3
File Edit Sketch Tools Help
Arduino Uno
click to compile
4_1_servo_test.ino
Output

```
/*****************************************************

 * This code applies to Arduino 6DOF Metal Arm Kit
 * Through this link you can download the source code:
 * https://github.com/Cokoino/CKK0017

*****************************************************/

#include <Servo.h>

Servo myservo;  // create servo object to control a servo
// twelve servo objects can be created on most boards

int pos = 0;    // variable to store the servo position

void setup() {
  myservo.attach(2);  // attaches the servo on pin 10 to the s
}

void loop() {
  for (pos = 0; pos <= 180; pos += 1) { // goes from 0 degrees
```

Done compiling.

4.7 Click upload button → , successfully uploading the code will display "Done uploading". When code is uploaded successfully, the program starts to run, the servo will cycle through 180 degrees.


4_1_servo_test | Arduino IDE 2.3.3
File Edit Sketch Tools Help
Arduino Uno
Upload
click to upload
4_1_servo_test.ino
Output
Ln 21, Col 36 Arduino Uno on C

```
/*****************************************************

 * This code applies to Arduino 6DOF Metal Arm Kit
 * Through this link you can download the source code:
 * https://github.com/Cokoino/CKK0017

*****************************************************/

#include <Servo.h>

Servo myservo;  // create servo object to control a servo
// twelve servo objects can be created on most boards

int pos = 0;    // variable to store the servo position

void setup() {
  myservo.attach(2);  // attaches the servo on pin 10 to the se
}

void loop() {
  for (pos = 0; pos <= 180; pos += 1) { // goes from 0 degrees
```

Done uploading.

4.8 Refer to the above steps and test the other 5 MG996 Servos (20KG) as well.

5. Code

Servo_test code:

```
#include <Servo.h>

Servo myservo;  // create servo object to control a servo
// twelve servo objects can be created on most boards
```

```
int pos = 0;    // variable to store the servo position

void setup() {
  myservo.attach(2);  // attaches the servo on pin 2 to the servo object
}

void loop() {
  for (pos = 0; pos <= 180; pos += 1) { // goes from 0 degrees to 180 degrees
    // in steps of 1 degree
    myservo.write(pos);              // tell servo to go to position in variable 'pos'
    delay(15);                       // waits 15 ms for the servo to reach the position
  }
  for (pos = 180; pos >= 0; pos -= 1) { // goes from 180 degrees to 0 degrees
    myservo.write(pos);              // tell servo to go to position in variable 'pos'
    delay(15);                       // waits 15 ms for the servo to reach the position
  }
}
```

6. Adjust all servo to 90° before assembly (Important)

Before assembling the servo to the robot arm you need to adjust them to 90° so that they can work well with the structure of the robot arm.Because the initialization state of the robotic arm has already been set in the underlying library where the code runs, all servos are in a 90 degree position. So we must ensure that the servo has been adjusted to 90 degrees during assembly, otherwise after being assembled onto the robotic arm, once powered on and initialized, the robotic arm cannot reach the ideal position state, and even mechanical collisions may occur.

The code to adjust the servo to 90° is placed in this folder:

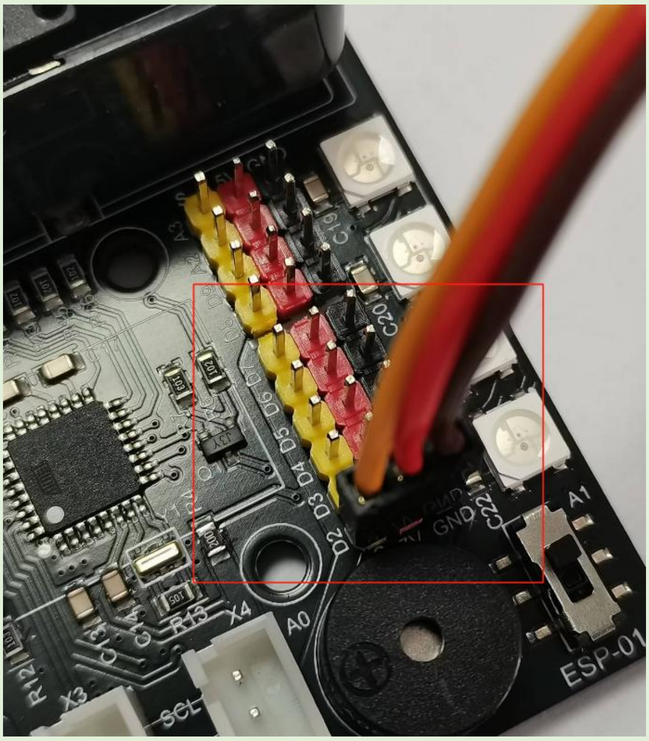
E:\CKK0017-main\Tutorial\sketches\4_2_servo_90_ADJ

Or you can also directly copy and paste the code from below to the arduino IDE

```
#include<Servo.h>
Servo myservo;  // Create a servo class

void setup() {
myservo.attach(2);  //Set the servo control pin as D2
delay(100);          //delay 100ms
}
/////////////////////////////////////////////////////////
void loop() {
 myservo.write(90);  //The servo is 90 degrees
 delay(1000);
 }
```

Upload the code to the Robotic Arm Driver Hat, and connect the adapter for power supply.

Insert the plug of the servo motor into D2 of the Robot Arm Driver Hat, paying attention to the signal pin corresponding to D2.



You will see the gears on the servo rotating and then stopping, indicating that the servo has been adjusted to 90 degrees. Remove the plug of the servo from the Robotic Arm Driver Hat board and use the same method to adjust the remaining servo one by one to 90 degrees.

7. Any questions and suggestions are welcome

Thank you for reading this document!

If you find any errors and omissions in the tutorial, or if you have any suggestions and questions, please feel free to contact us:

cokoino@outlook.com

We will do our best to make changes and publish revisions as soon as possible.

If you want to learn more about Arduino, Raspberry Pi, Smart Cars, Robotics and other interesting products in science and technology, please continue to visit our Amazon Store by search for " **LK COKOINO**" on Amazon. We will continue to launch fun, cost-effective, innovative and exciting products.

LK COKOINO