Lesson 5 Control Multiple PWM Servos

1. Working Principle

PWM servo can be controlled by sending pulse signal so that you can change servo port, rotation angle and rotation time in program to control servo.

The path to the source code of the program is 5. MaxArm Hardware Basic Learning/Arduino Development/Game Programs/Control Multiple PWM Servos/Multi_PWMServo\Multi_PWMServo.ino

```
void setup() {
  PWMServo_init();    // Initialize PWM servo library
  Serial.begin(9600); // Set the baud rate
  Serial.println("start...");  // The serial port prints "start..."
  delay(200);                  // The delay of 200ms

}

bool start_en = true;
void loop() {
  // put your main code here, to run repeatedly:
  if(start_en){
    SetPWMServo(1,500,2000);  // Set the pulse width of ID1 PWM servo as 500 and the running time as 2000ms.
    SetPWMServo(2,500,2000);  // Set the pulse width of ID2 PWM servo as 500 and the running time as 2000ms.
    delay(200);               // The delay of 200ms
    SetPWMServo(1,2500,2000); // Set the pulse width of ID1 PWM servo as 2500 and the running time as 2000ms.
    SetPWMServo(2,2500,2000); // Set the pulse width of ID2 PWM servo as 2500 and the running time as 2000ms.
    delay(200);               // The delay of 200ms
    SetPWMServo(1,500,2000);  // Set the pulse width of ID1 PWM servo as 500 and the running time as 2000ms.
    SetPWMServo(2,500,2000);  // Set the pulse width of ID2 PWM servo as 500 and the running time as 2000ms.
    start_en = false;
  }
  else{
    delay(500); // The delay of 500ms
  }
}
```

PWM servo mainly calls SetPWMServo() function in PWMServo library. Take the code “SetPWMServo(1,500,1000)” as example.

The first parameter “1” is the port number of PWM servo. Here is No.1 port.

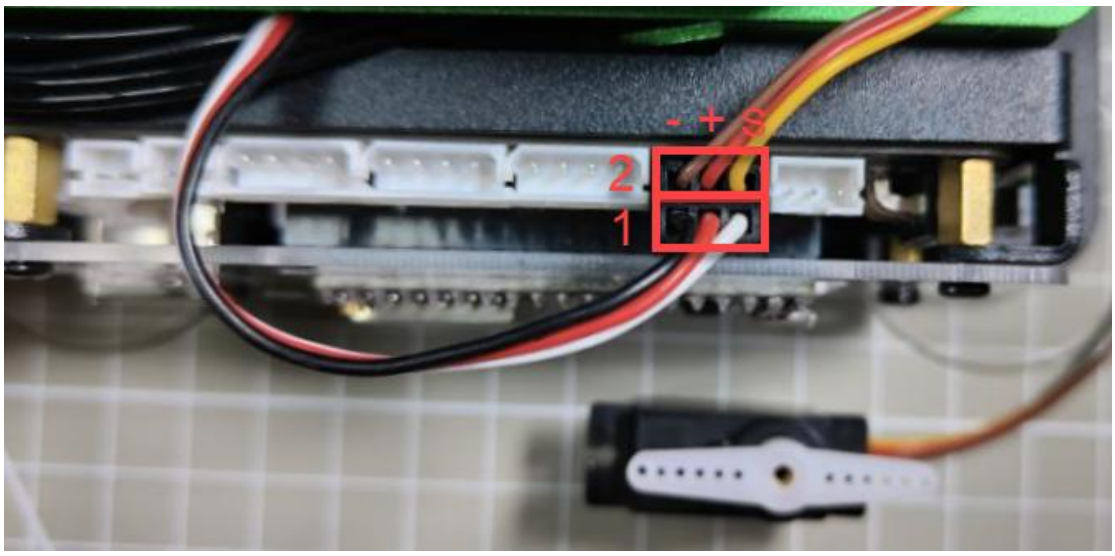
The second parameter “500” is the rotation position which is converted by pulse width data (pulse width=11.1×angle+500, the formula just for your information). Therefore, the parameter 500 corresponds to 0° rotation angle.

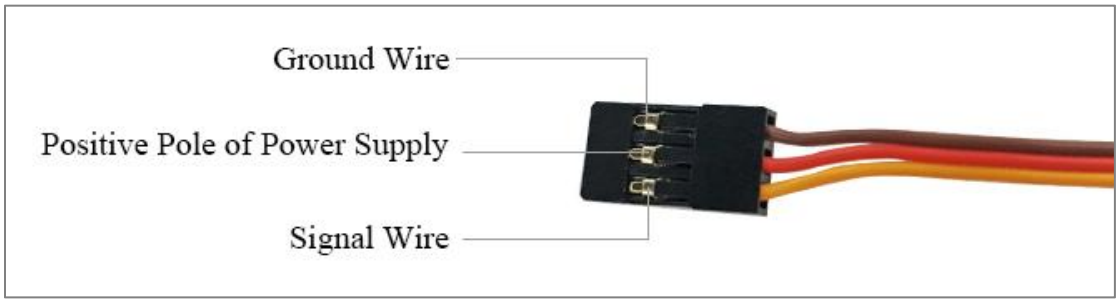
The third parameter “1000” is the rotation time (unit: ms). The parameter here is 1000, i.e, 1000ms.

2. Preparation

2.1 Hardware

Connect a single PWM servo to PWM servo port on MaxArm controller. Take connecting LFD-01 servo (5V) to No.1 port as example. The wiring method is as follow:





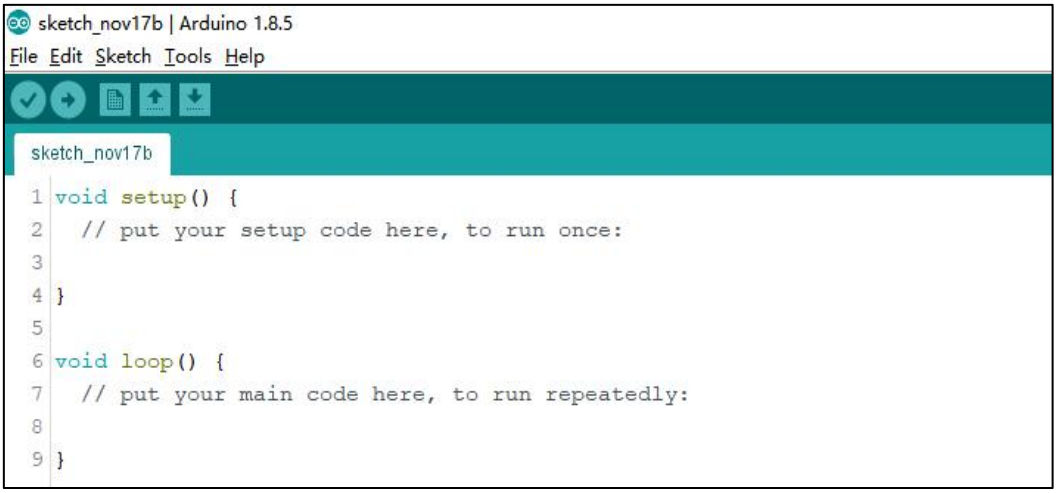
Note: Please note the direction of servo cable, otherwise servo may burn out (S pin is signal terminal).

2.2 Software

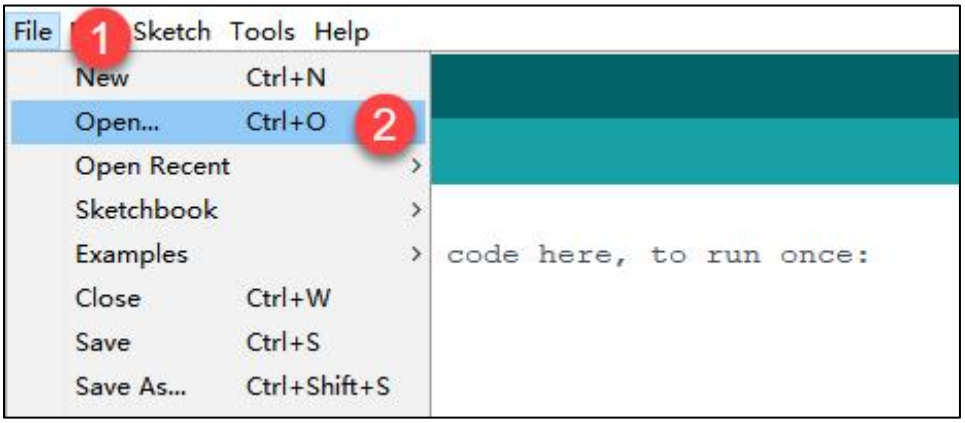
Please connect MaxArm to the Arduino editor according to the tutorial in folder “4. MaxArm Underlying Program Learning/Python Development/Lesson 1 Set Development Environment”.

3. Program Download

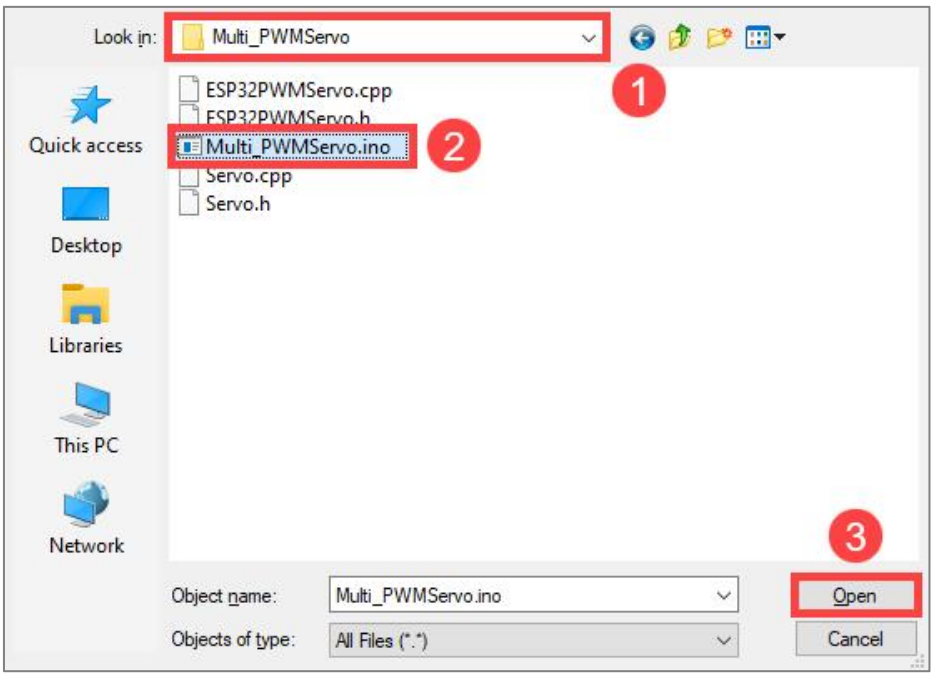
1) Double click on  icon to open Arduino IDE.



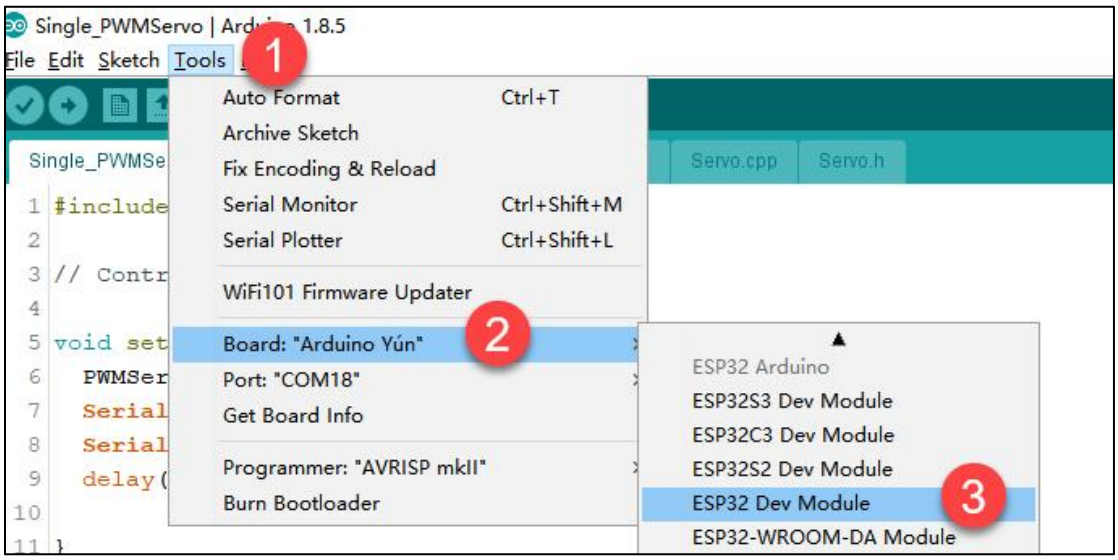
2) Click “File->Open” in turn.



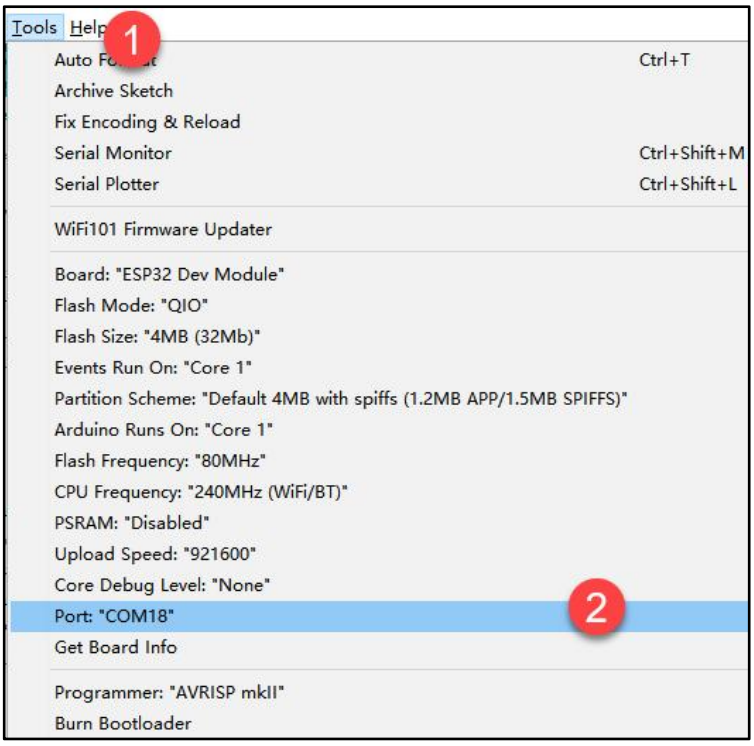
3) Select the program “Multi_PWMServo.ino” in the folder “5.MaxArm Hardware Basic Learning/Arduino Development/Game Programs/Control Multiple PWM Servos/Multi_PWMServo”.



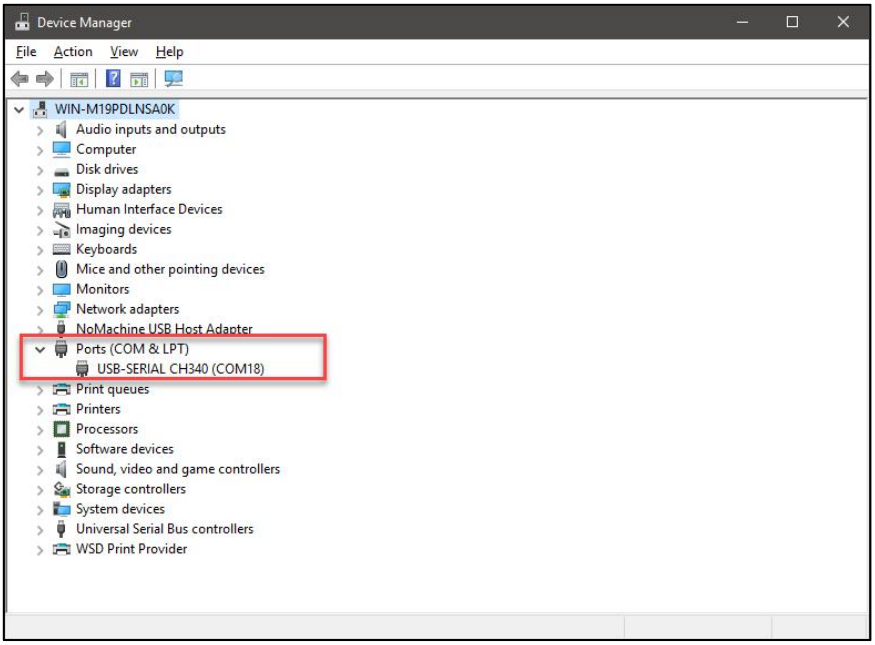
4) Check the board model. Click “Tools->Board” and select “ESP 32 Dev Module”. (If the model of development board has been configured when setting the development environment, you can skip this step.)



5) Select the corresponding port of ESP32 controller in “Tools->Port”. (Here take the port “COM5” as example. Please select the port based on your computer. If COM1 appears, please do not select because it is the system communication port but not the actual port of the development port.)

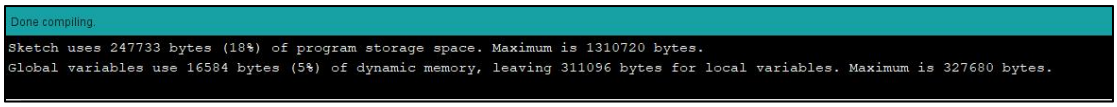


6) If you are not sure about the port number, please open the “This PC” and click “Properties->Device Manger” in turns to check the corresponding port number (the device is with CH340).

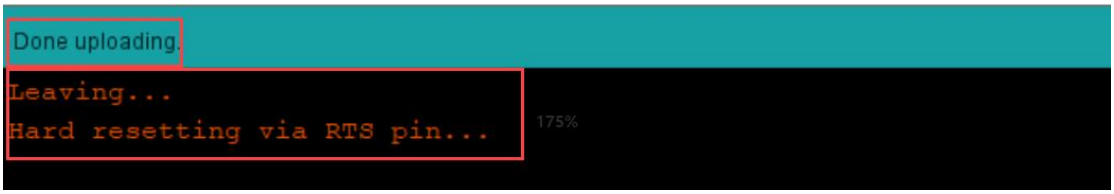


7) After selecting, confirm the board “ESP32 Dev Module” in the lower right corner and the port number “COM5” (it is an example here, please refer to the actual situation).

8) Then click on  icon to verify the program. If no error, the status area will display “Compiling->Compile complete” in turn. After compiling, the information such as the current used bytes, and occupied program storage space will be displayed.



9) After compiling, click on  icon to upload the program to the development board. The status area will display “Compiling->Uploading->Complete” in turn. After uploading, the status area will stop printing the uploading information.



3. Project Outcome

When running program, two LFD-01M servos will rotate from 0° to 180°, and then to 0°. After the servos stop rotating, exit program automatically.

4. Function Extension

The rotation position set in program is from 0° to 180°, and then to 0°. You can modify the rotation position by modifying the corresponding code. Here the second parameter of run() function of No.1 servo is changed from 500 to 2500, and the second parameter of the run() function of No.2 servo is changed from

2500 to 500. The specific operation steps are as follow:

Find the following program code:

```
17    SetPWMServo(1,500,2000);  // Set the pulse width of ID1 PWM servo as 500 and the running time as 2000ms.
18    SetPWMServo(2,500,2000);  // Set the pulse width of ID2 PWM servo as 500 and the running time as 2000ms.
19    delay(200);               // The delay of 200ms
20    SetPWMServo(1,2500,2000); // Set the pulse width of ID1 PWM servo as 2500 and the running time as 2000ms.
21    SetPWMServo(2,2500,2000); // Set the pulse width of ID2 PWM servo as 2500 and the running time as 2000ms.
22    delay(200);               // The delay of 200ms
23    SetPWMServo(1,500,2000);  // Set the pulse width of ID1 PWM servo as 500 and the running time as 2000ms.
24    SetPWMServo(2,500,2000);  // Set the pulse width of ID2 PWM servo as 500 and the running time as 2000ms.
25    start_en = false;
```

1) Change the second parameter of run() function of No.1 servo from 500 to 2500, and the second parameter of the run() function of No.2 servo from 2500 to 500, as shown in the image below:

```
16  if(start_en){
17    SetPWMServo(1,2500,2000);  // Set the pulse width of ID1 PWM servo as 2500 and the running time as 2000ms.
18    SetPWMServo(2,2500,2000);  // Set the pulse width of ID2 PWM servo as 2500 and the running time as 2000ms.
19    delay(200);                // The delay of 200ms
20    SetPWMServo(1,500,2000);  // Set the pulse width of ID1 PWM servo as 500 and the running time as 2000ms.
21    SetPWMServo(2,500,2000);  // Set the pulse width of ID2 PWM servo as 500 and the running time as 2000ms.
22    delay(200);                // The delay of 200ms
23    SetPWMServo(1,2500,2000);  // Set the pulse width of ID1 PWM servo as 2500 and the running time as 2000ms.
24    SetPWMServo(2,2500,2000);  // Set the pulse width of ID2 PWM servo as 2500 and the running time as 2000ms.
25    start_en = false;
```

2) After modifying, click on  icon to verify the program.

3) Click on  icon to upload the program to the development board, and then check the outcome.