Lesson 4 Control PWM Servo Speed

1. Working Principle

A single 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 source code of program is 5. MaxArm Hardware Basics Learning/Arduino Development/Game Programs/Control PWM Servo Speed/PWMServo_speed/PWMServo_speed.ino

```
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.
    delay(200);               // The delay of 200ms
    int t[2]= {500, 2000};
    for(int i=0; i<2; i++){ // Run one round at different times, the longer the time, the slower the speed.
      SetPWMServo(1,500,t[i]); // Set the pulse width of ID1 PWM servo as 500
      delay(200);               // The delay of 200ms
      SetPWMServo(1,2500,t[i]);  // Set the pulse width of ID1 PWM servo as 2500
      delay(200);               // The delay of 200ms
      SetPWMServo(1,500,t[i]); // Set the pulse width of ID1 PWM servo as 500
      delay(200);               // The delay of 200ms
    }
    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.

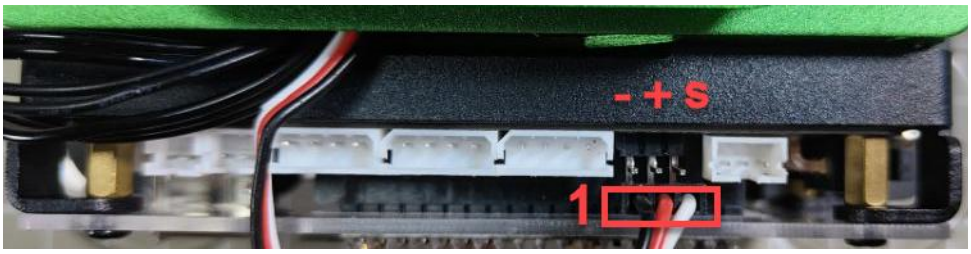
The second parameter "500" is the rotation position which is converted by pulse width data (pulse width=11.1×angle+500, the formula just for your information). Therefore, the parameter 500 corresponds to 0° rotation angle.

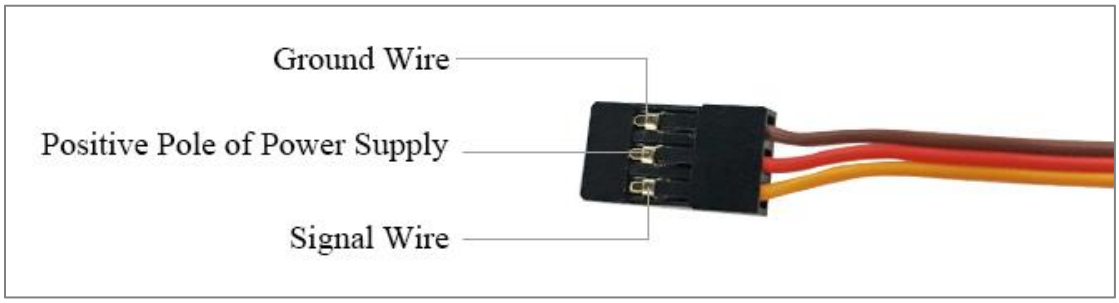
The third parameter "t" is the rotation time (unit:ms). The parameter for the first rotation is 500, i.e, 500ms. The second round of rotation is 2000, i.e. 2000ms.

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:





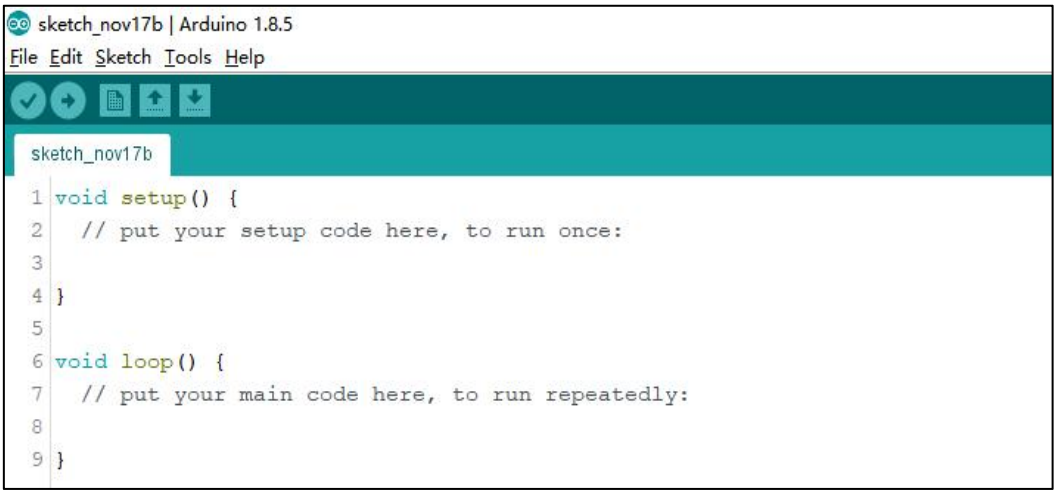
Note: Please note the direction of servo cable, otherwise servo may burn out (S pin is signal terminal).

2.2 Software

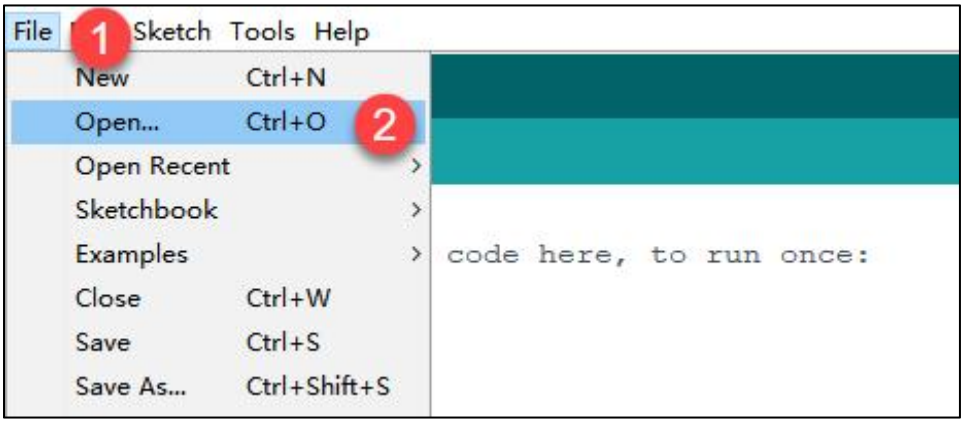
Please connect MaxArm to the Arduino editor according to the tutorial in folder “4. MaxArm Underlying Program Learning/Python Development/Lesson 1 Set Development Environment”.

3. Program Download

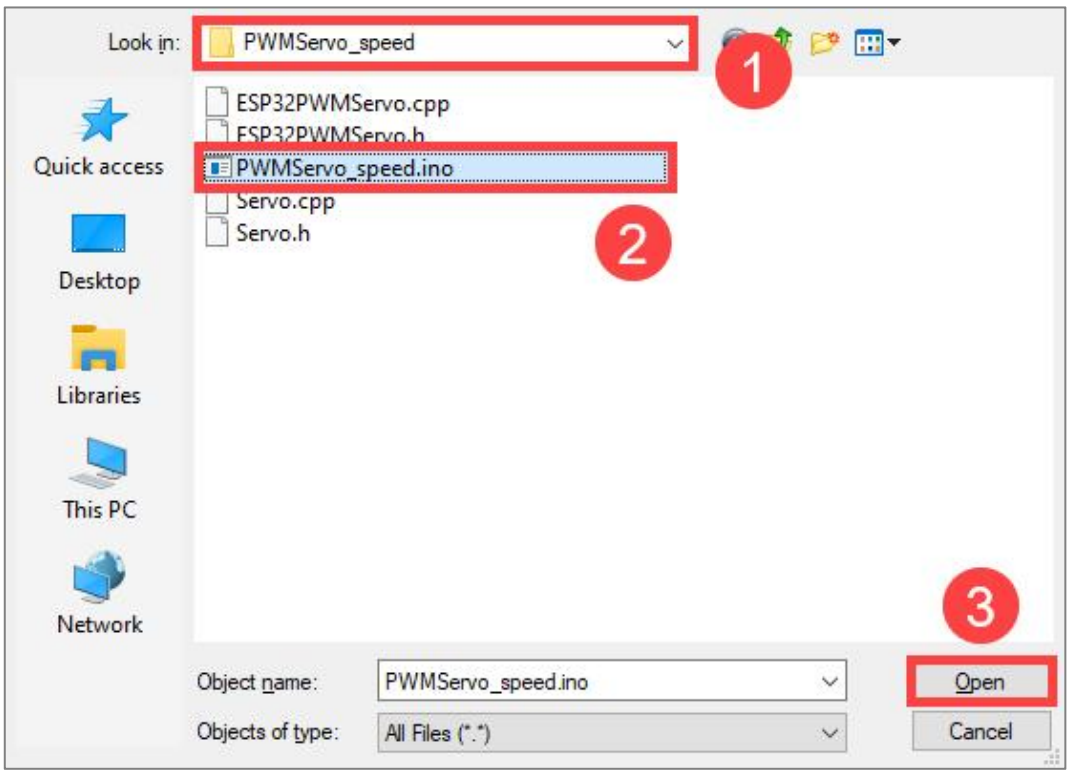
1) Double click on  icon to open Arduino IDE.



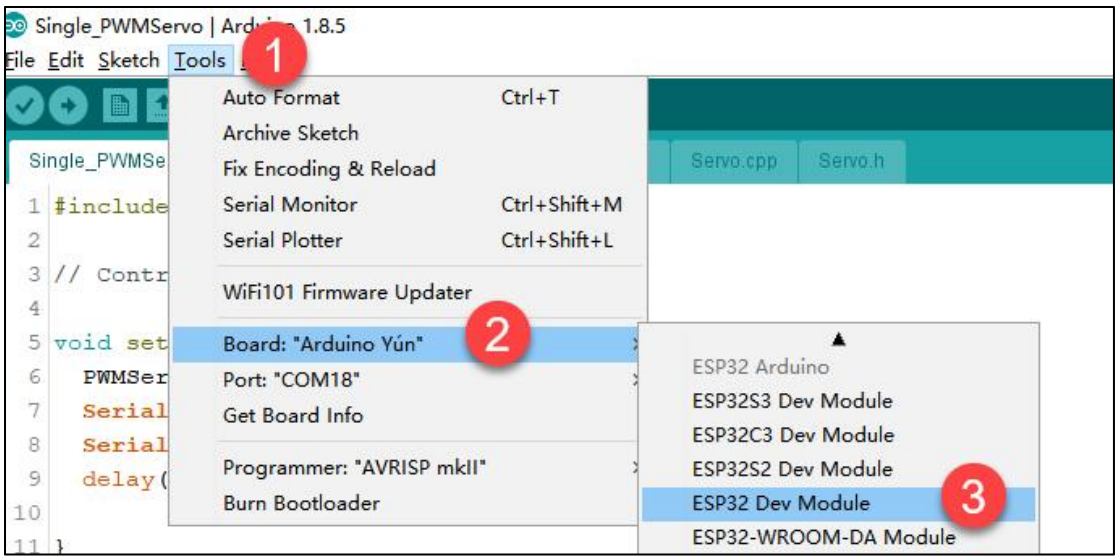
2) Click “File->Open” in turn.



3) Select the program “PWMServo_speed.ino” in the folder “5.MaxArm Hardware Basic Learning/Arduino Development/Game Programs/Control PWM Speed/\PWMServo_speed”.

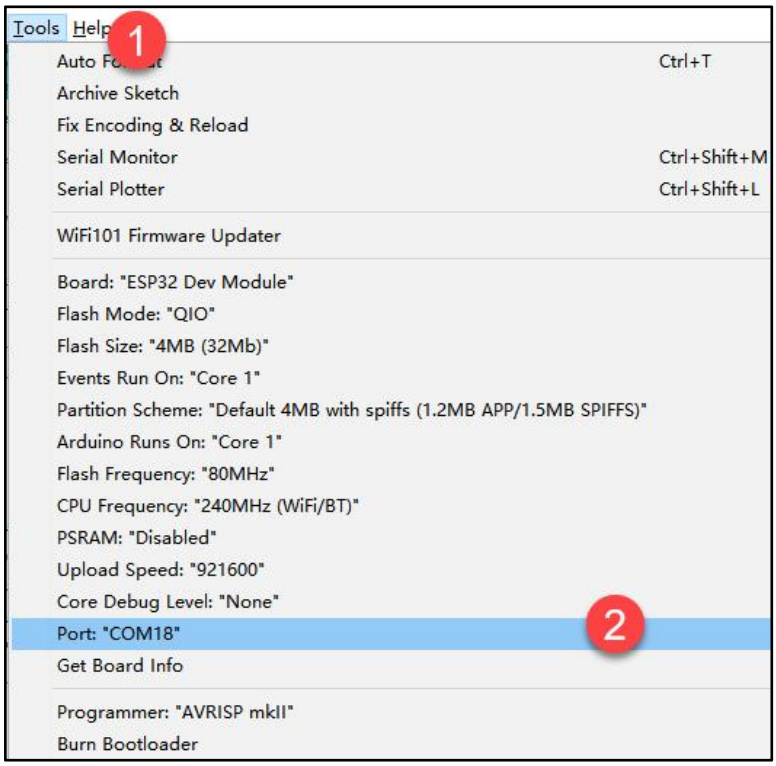


4) Check the board model. Click “Tools->Board” and select “ESP 32 Dev Module”. (If the model of development board has been configured when setting the development environment, you can skip this step.)

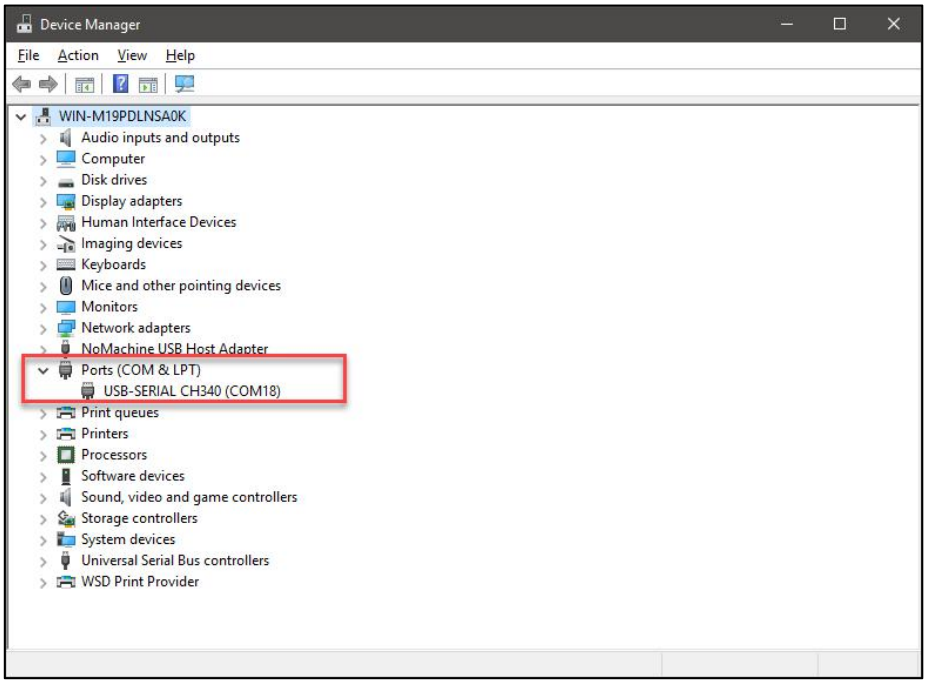


5) Select the corresponding port of ESP32 controller in “Tools->Port”. (Here take the port “COM5” as example. Please select the port based on your computer. If COM1 appears, please do not select because it is the system

communication port but not the actual port of the development port.)

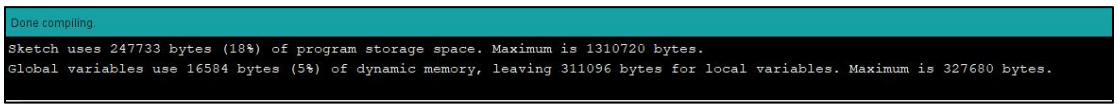


6) If you are not sure the port number, please open the “This PC” and click “Properties->Device Manger” in turns to check the corresponding port number (the device is with CH340).

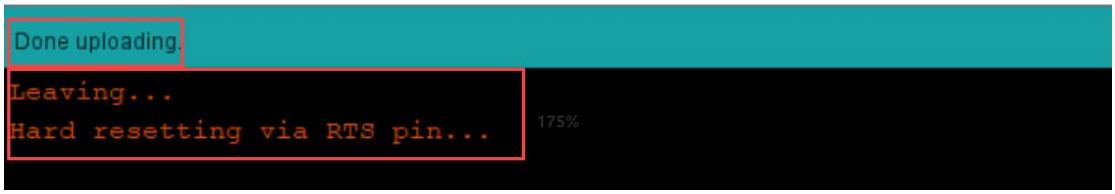


7) After selecting, confirm the board “ESP32 Dev Module” in the lower right corner and the port number “COM5” (it is an example here, please refer to the actual situation).

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.



4. Program Outcome

When running program, LFD-01M servo will rotate from “0° to 180°, and then to 0°. This process will repeat twice and the second rotation will be much faster than the first rotation. After the servo stops rotating, exit the program automatically.

5. Function Extension

1) The first rotation speed set in program is faster than the second one. If want to modify its rotation speed, please modify the corresponding code. Here the “t” parameter value is changed from (500,2000) to (2000,500). 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    delay(200);               // The delay of 200ms
19    int t[2]= {500, 2000};
20    for(int i=0; i<2; i++){ // Run one round at different times, the longer the time, the slower the speed.
21      SetPWMServo(1,500,t[i]); // Set the pulse width of ID1 PWM servo as 500
22      delay(200);               // The delay of 200ms
23      SetPWMServo(1,2500,t[i]);  // Set the pulse width of ID1 PWM servo as 2500
24      delay(200);               // The delay of 200ms
25      SetPWMServo(1,500,t[i]); // Set the pulse width of ID1 PWM servo as 500
26      delay(200);               // The delay of 200ms
```

2) Change the first parameter of “t” to 2000 and the second parameter to 500, as shown in the image below:

```
17    SetPWMServo(1,500,2000);  // Set the pulse width of ID1 PWM servo as 500 and the running time as 2000ms.
18    delay(200);               // The delay of 200ms
19    int t[2]= {2000, 500};
20    for(int i=0; i<2; i++){ // Run one round at different times, the longer the time, the slower the speed.
21      SetPWMServo(1,500,t[i]); // Set the pulse width of ID1 PWM servo as 500
22      delay(200);               // The delay of 200ms
23      SetPWMServo(1,2500,t[i]);  // Set the pulse width of ID1 PWM servo as 2500
24      delay(200);               // The delay of 200ms
25      SetPWMServo(1,500,t[i]); // Set the pulse width of ID1 PWM servo as 500
26      delay(200);               // The delay of 200ms
```

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

```
Done compiling.
Sketch uses 247733 bytes (18%) of program storage space. Maximum is 1310720 bytes.
Global variables use 16584 bytes (5%) of dynamic memory, leaving 311096 bytes for local variables. Maximum is 327680 bytes.
```

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