

Lesson 3 Control a Single PWM Servo

1. Working Principle

A single PWM servo can be controlled by sending pulse signal so that you can modify 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 a Single PWM Servo/Single_PWMServo/Single_PWMServo.ino

```
Single_PWMServo $ ESP32PWMServo.cpp ESP32PWMServo.h Servo.cpp Servo.h
1 #include "ESP32PWMServo.h" // Import library file
2
3 // Control a single servo
4
5 void setup() {
6     PWMServo_init(); // Initialize PWM servo library
7     Serial.begin(9600); // Set the baud rate
8     Serial.println("start..."); // The serial port prints "start..."
9     delay(200); // The delay of 200ms
10
11 }
12
13 bool start_en = true;
14 void loop() {
15     // put your main code here, to run repeatedly:
16     if(start_en){
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         SetPWMServo(1,2500,2000); // Set the pulse width of ID1 PWM servo as 2500 and the running time as 2000ms.
20         delay(200); // The delay of 200ms
21         SetPWMServo(1,500,2000); // Set the pulse width of ID1 PWM servo as 500 and the running time as 2000ms.
22         start_en = false;
23     }
24     else{
25         delay(500); // The delay of 500ms
26     }
27 }
```

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

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

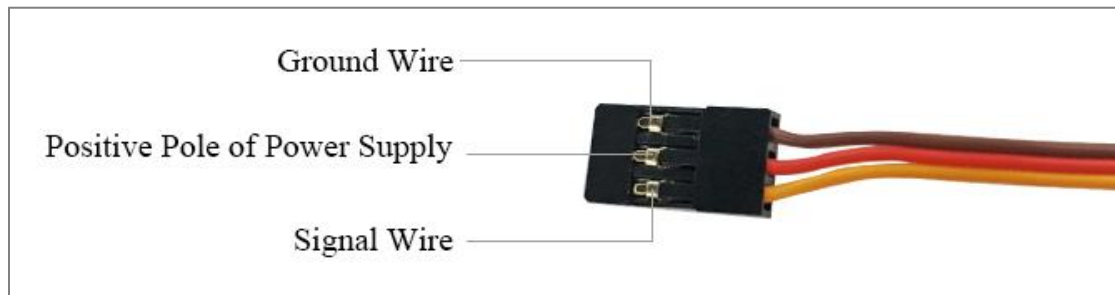
The second parameter "500" is the rotation position which is converted by pulse width data (pulse width= $11.1 \times \text{angle} + 500$, the formula just for your information). Therefore, the parameter 500 corresponds to 0° rotation angle.

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 suction nozzle is controlled by LFD-01M servo). The wiring method is as follow:



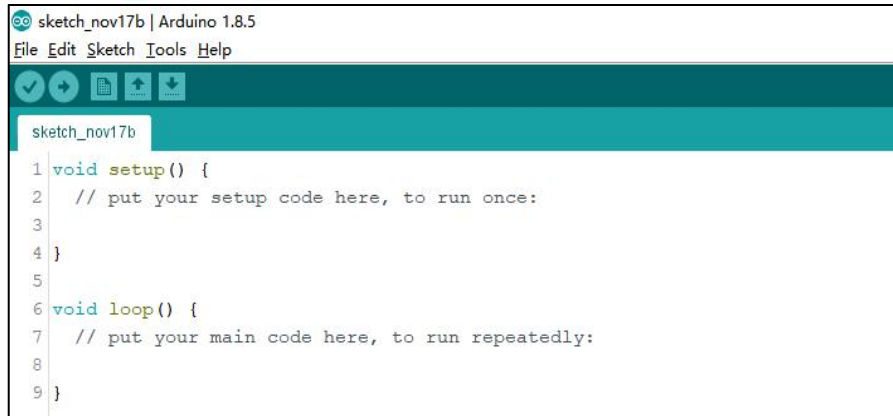
Note: Please note the direction of servo cable, otherwise servo may burn out (S pin is signal terminal).

2.2 Software

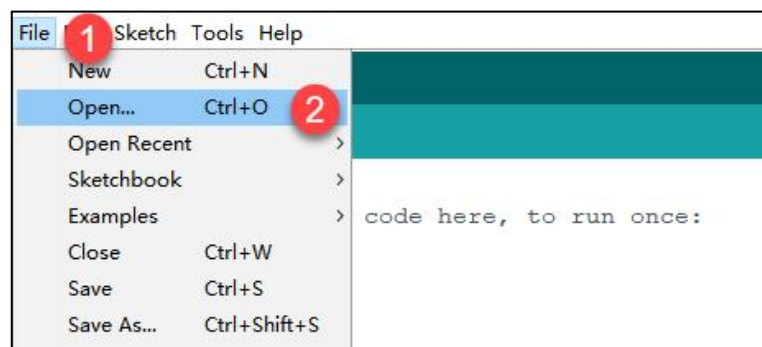
Please connect MaxArm to the Arduino editor according to the tutorial in folder “4. MaxArm Underlying Program Learning/Python Development/Lesson 1 Set Development Environment”.

3. Program Download

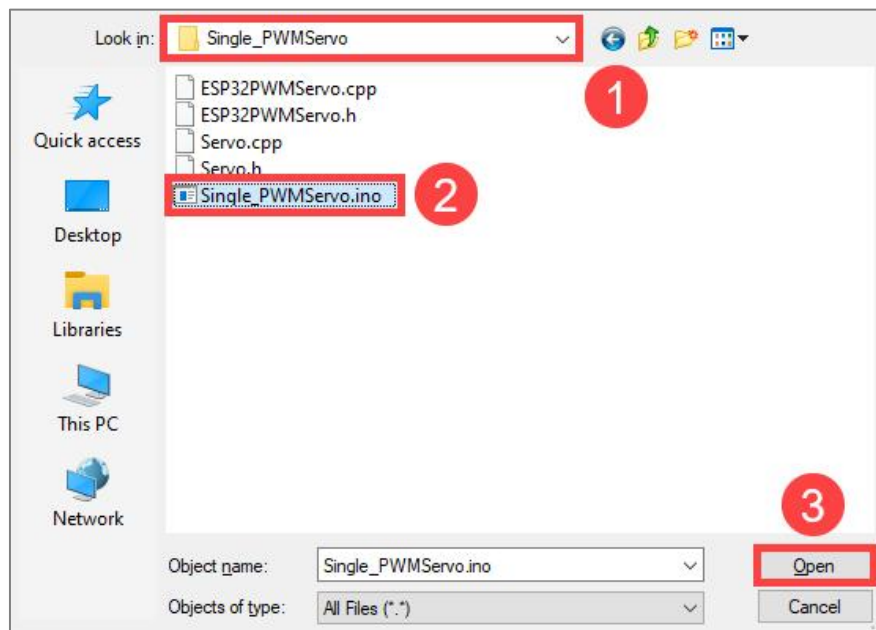
- 1) Double click on  icon to open Arduino IDE.



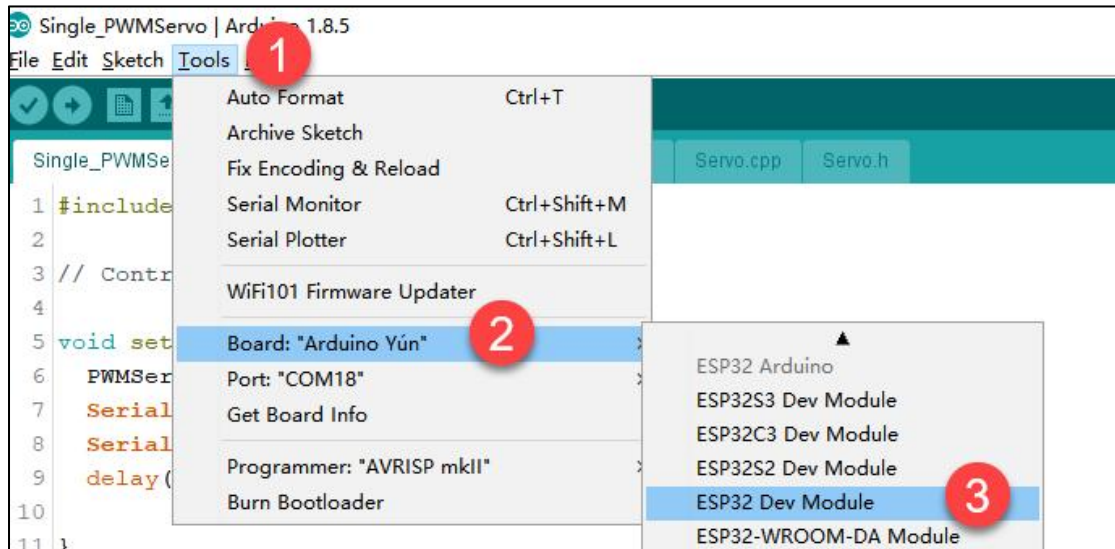
2) Click "File->Open" in turn.



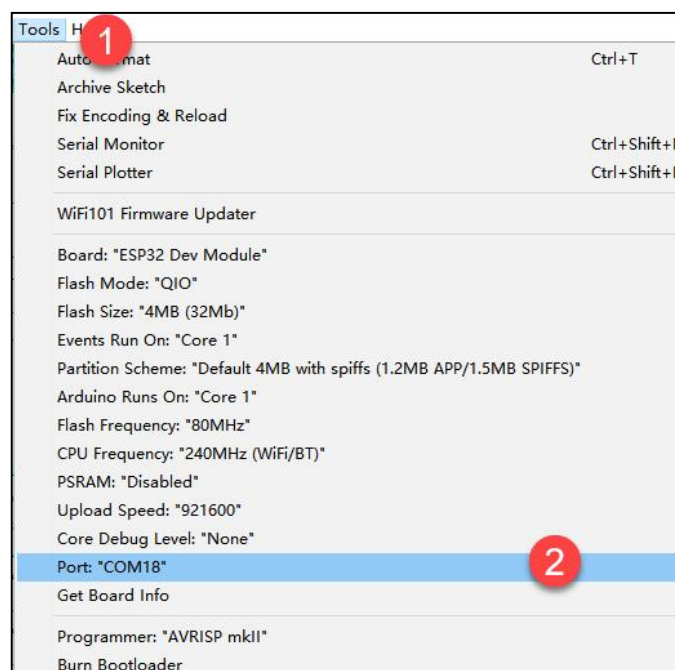
3) Select the program "Single_PWM servo.ino" in the folder "5.MaxArm Hardware Basic Learning/Arduino Development/Game Programs/Control a Single PWM servo/Single_PWM servo".



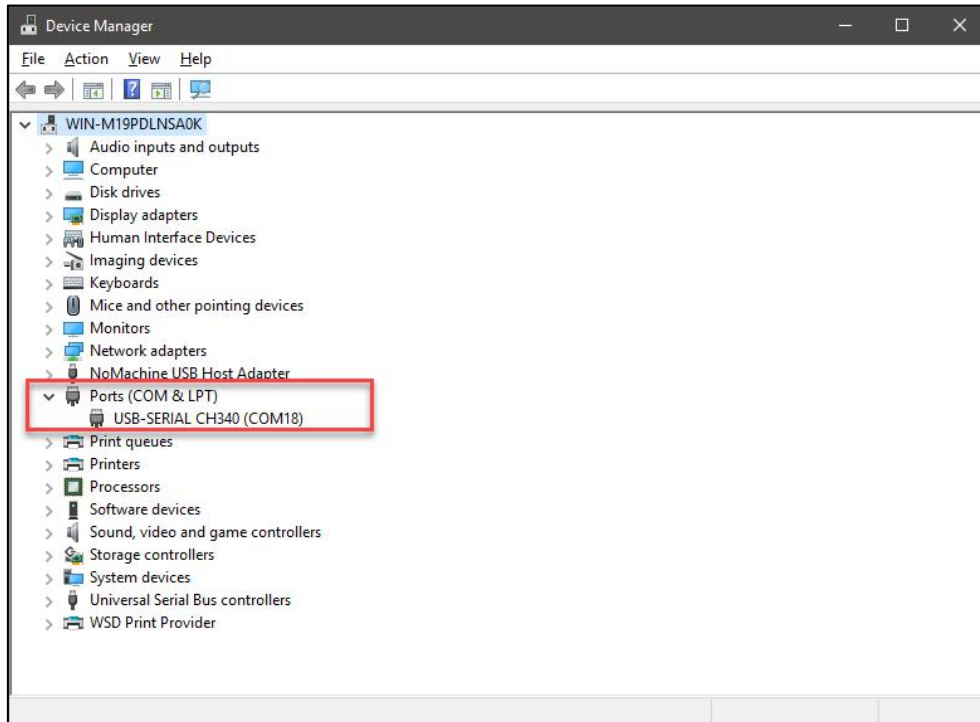
- 4) Check the board model. Click “Tools->Board” and select “ESP 32 Dev Module”. (If the model of development board has been configured when setting the development environment, you can skip this step.)



- 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).


ESP32 Dev Module, Disabled, Default 4MB with spiiffs (1.2MB APP/1.5MB SPIFFS), 240MHz (WiFi/BT), QIO, 80MHz, 4MB (32Mb), 921600, Core 1, Core 1, None on COM18

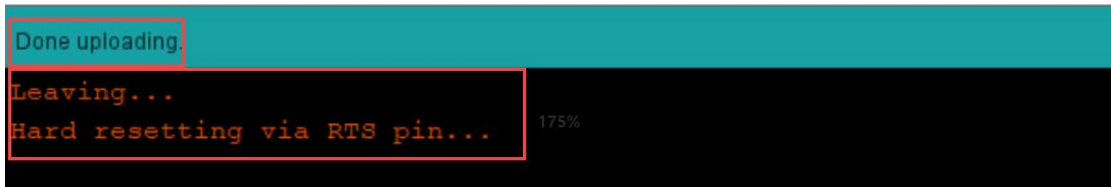
- 8) Then click on  icon to verify the program. If no error, the status area will display “Compiling->Compile complete” in turn. After compiling, the information such as the current used bytes, and occupied program storage space will be displayed.

Done compiling.

Sketch uses 247689 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.

- 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-01 servo will rotate from 90° to 0° , and then to 180° . After the program stops, exit the program automatically.

5. Function Extension


The servo port set in program is No.1 port. If want to change the port, you can modify the port through the code. This section takes changing No.1 port to No.2 port as example. The specific operation steps are as follow.

Find the following program:


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

- 1) Change the first parameter of SetPWMServo() function to 2, as shown in the image below:

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


- 2) After modifying, 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.

```
Done compiling.  
Sketch uses 247689 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.
```

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