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. Hardware Basic Learning/ Python Development/Program Files/Control a Single PWM Servo /main.py

```python
import time
from PWMServo import PWMServo

# Conrol a single PWM servo

pwm = PWMServo()
pwm.work_with_time()

if __name__ == '__main__':
    pwm.run(1, 500, 1000) # ID1 PWM servo is set as 500 pulse width. The running time is 1000ms (PWM servo
    cannot read the current position, so the runing time can not be controlled at the firs time)
    time.sleep_ms(2000) # The delay of 2000ms

    pwm.run(1, 2500, 2000) # The ID1 PWM servo is set as 2500 width pulse and the running time is set as 2000ms.
    time.sleep_ms(2000)

    pwm.run(1, 500, 2000) # The ID1 PWM servo is set as 500 width pulse and the running time is set as 2000ms.
    time.sleep_ms(2000)
```

PWM servo mainly uses run() 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×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

Connect a single PWM servo to PWM servo port on MaxArm controller. Take connecting LFD-01M servo (5V) to No.1 port as example (The servo used to control the suction nozzle is LFD-01M, do not need to assemble by yourself). The wiring method is as follow:


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

3. Operation Steps

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

Port:COM18 Board:esp32,connect successfully!
>>>

2) After connecting, change the path of Workspace to "5. Hardware Basic Learning/Python Development" and select "Program Files".
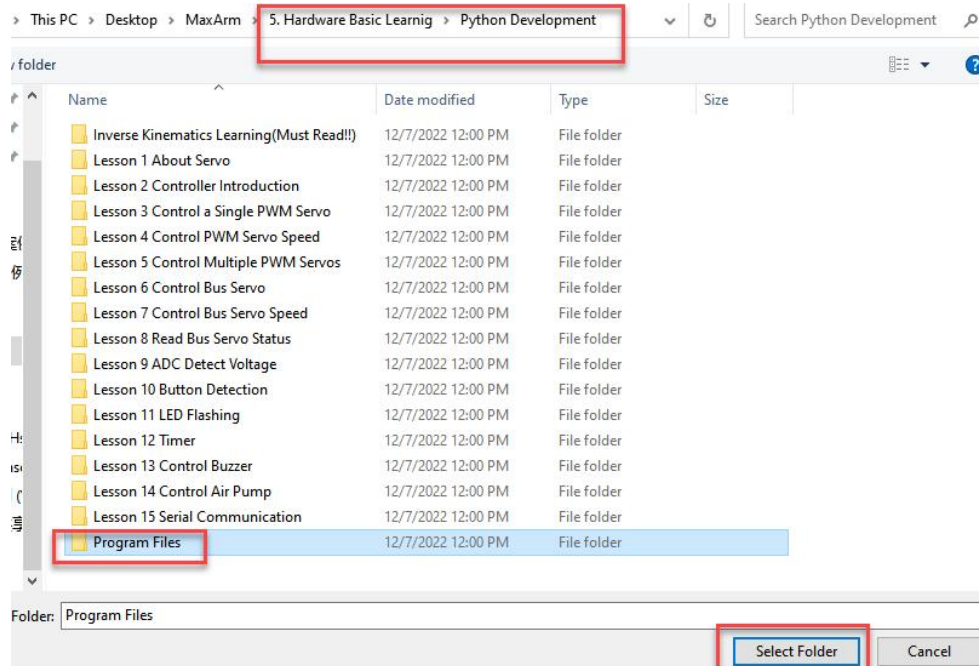
3) Double click folder "Control a Single PWM Servo", and then double click "main.py" to open program.
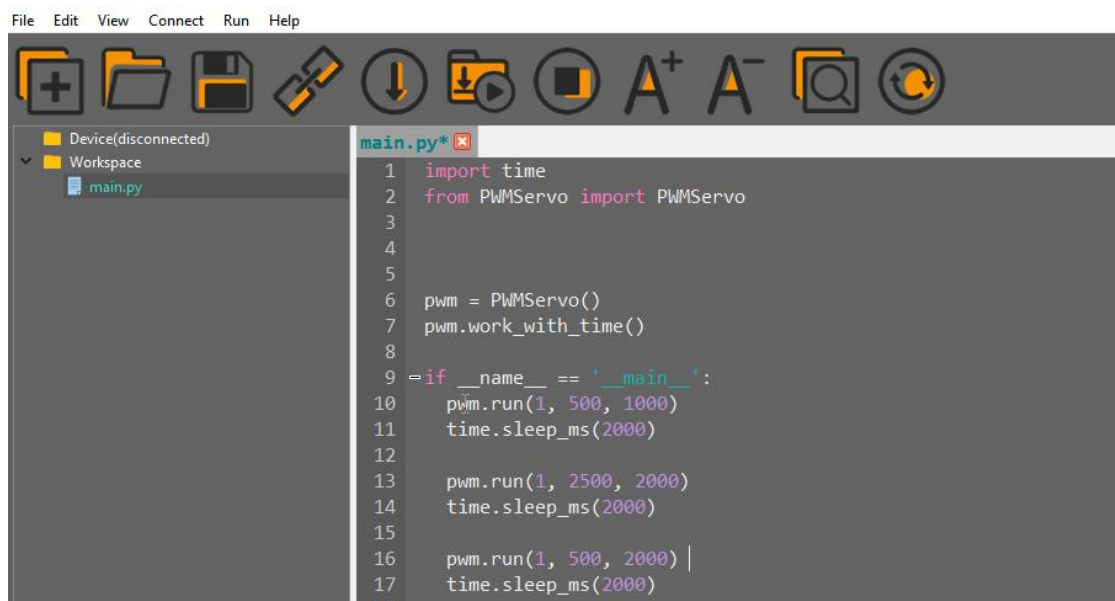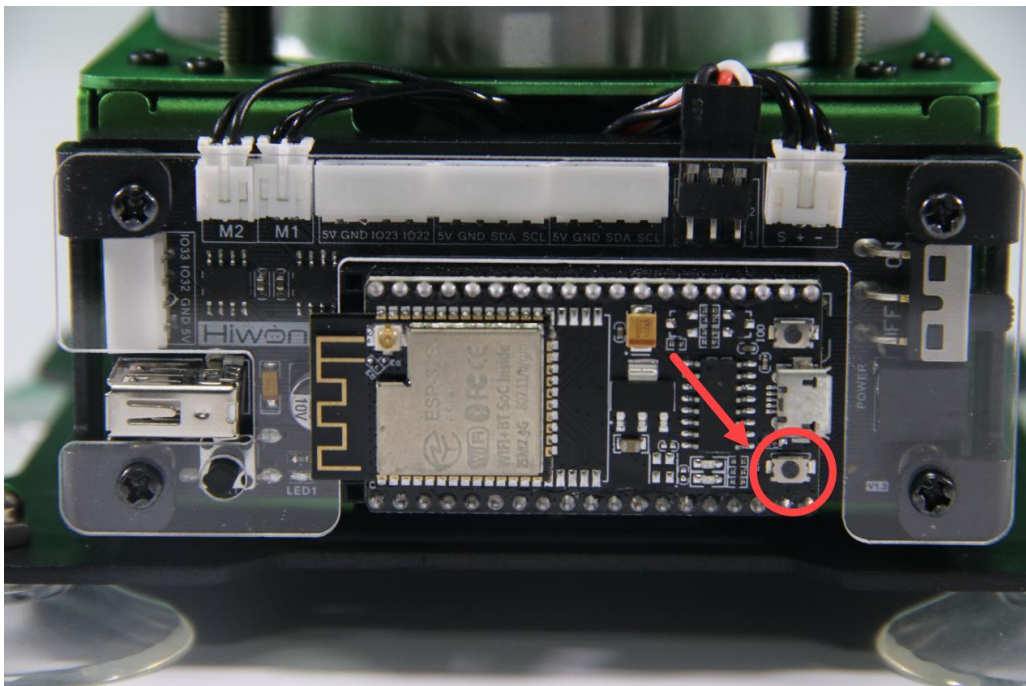
4) Click on the download icon to download program to ESP32 controller.

5) When the terminal prints the prompt, as shown in the image below, it means download completed.

```
>>>
Downloading....
main.py Download ok!
>>>
```

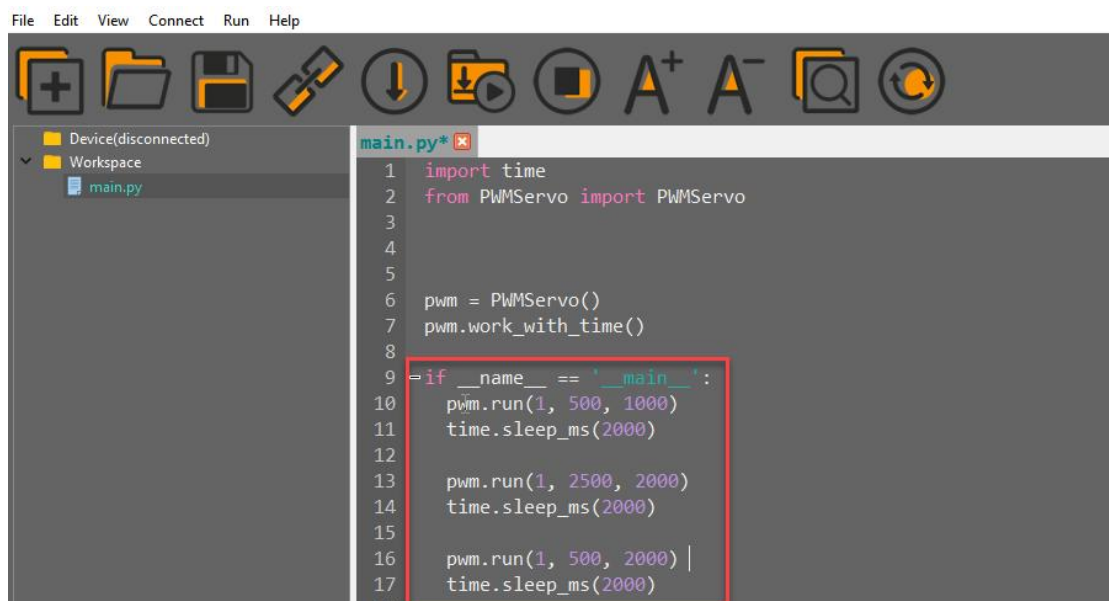6) After downloading, click on the reset icon or press the reset button on ESP32 controller to run program.

4. Project Outcome

When running program, LFD-01M servo will rotate from "90° to 0°, and then to 180°. After the servo stops rotating, exit program automatically.

5. Function Extension

The servo connection port set in program is No.1. If want to change connection port, please modify the corresponding code. Here the parameter of servo port is changed from No.1 to No.2. The specific operation steps are as follow:

Find the following program code:

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

```
 9 ⊟if __name__ == '__main__':
10     pwm.run(1, 500, 1000) # ID1 PWM servo is set as 500 pulse width.
11     time.sleep_ms(2000) # The delay of 2000ms
12
13     pwm.run(1, 2500, 2000) # The ID1 PWM servo is set as 2500 width
14     time.sleep_ms(2000)
15
16     pwm.run(1, 500, 2000) # The ID1 PWM servo is set as 500 width pu
17     time.sleep_ms(2000)
```

2) After modifying, click on icon to check grammar. In the mean time, the terminal will show the following prompt.

```
>>>
Syntax check completed,no errors
```

3) Click on icon and connect PWM servo to No.2 port.

4) Refer to "3.Operation steps 4-6", download and run the program to check the project outcome.