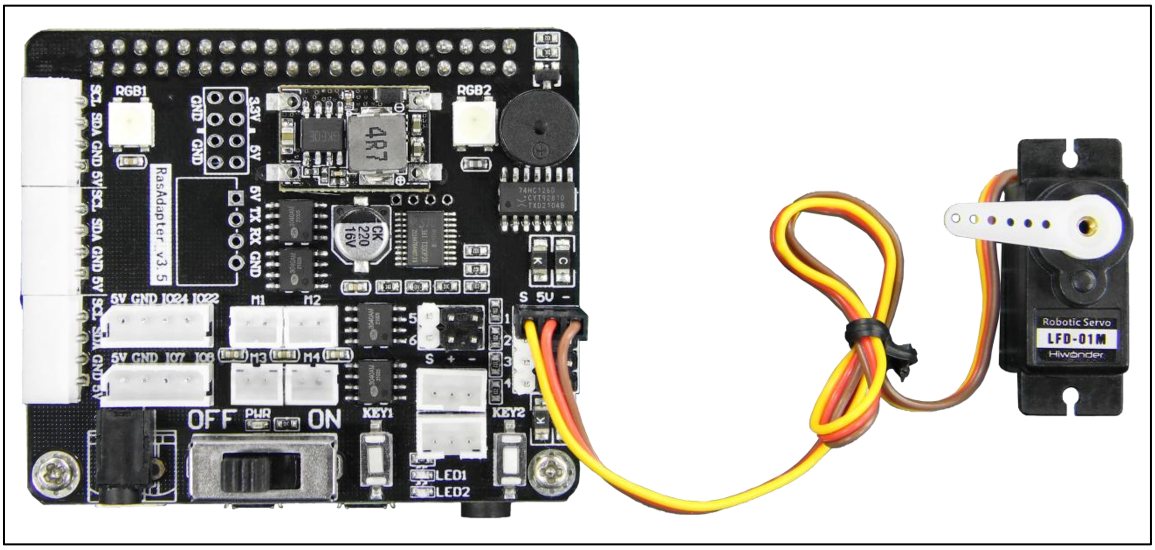
Lesson 4 PWM Servo Control

1. Getting Ready

Connect a PWM servo to No.1 servo port of Raspberry Pi expansion board. This lesson takes connecting LFD-01M servo (5V) as example.



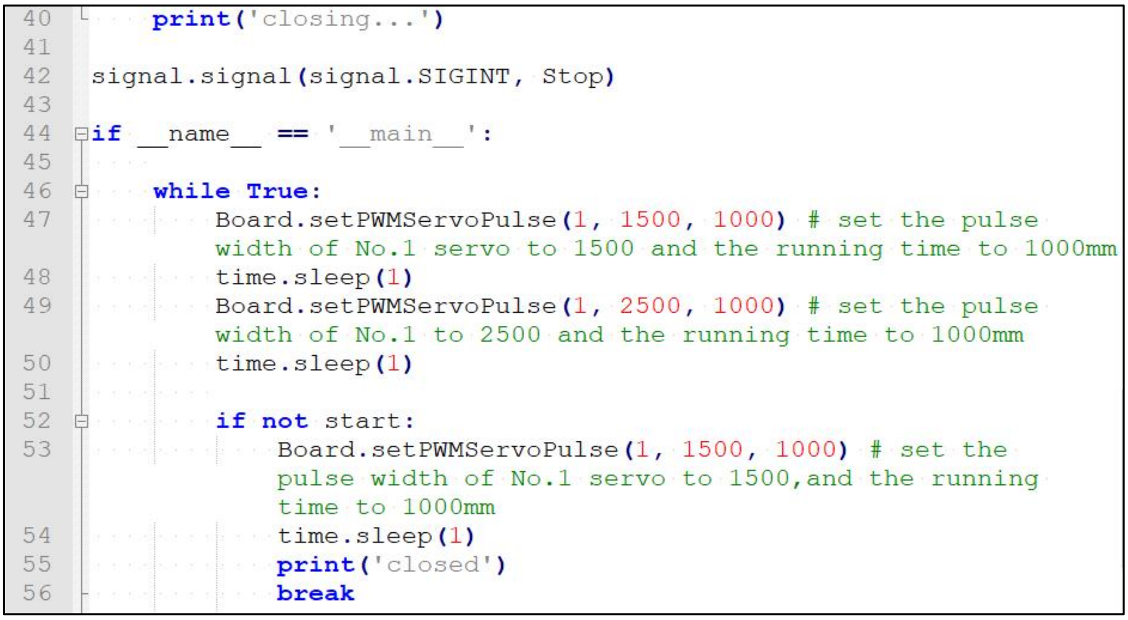
Note: When connecting servo, the yellow line is connected to “S”, red to “5V” and brown to “-”. Incorrect connection leads to servo burnout.

2. Working Principle

Let’s learn about how to realize this project:

Pulse signal is transmitted to control the rotation of servo. The rotation angle and rotation time of servo can be set with code parameter.

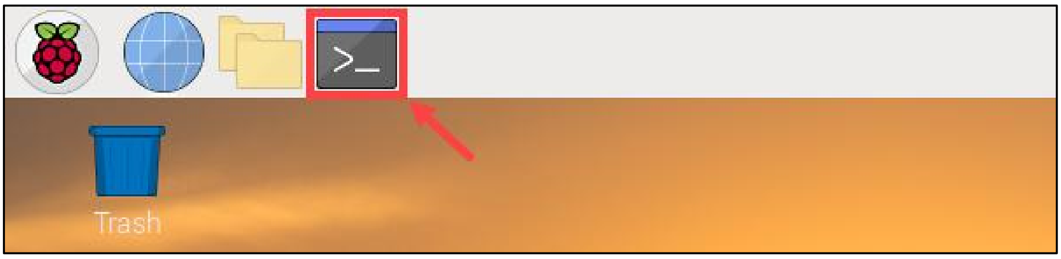
The source code of program is located in: **/home/pi/MasterPi/HiwonderSDK/PWMServoControlDemo.py**



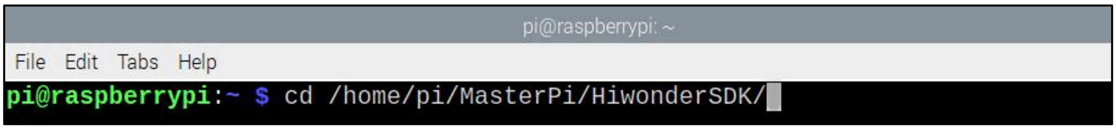
```
    print('closing...')

signal.signal(signal.SIGINT, Stop)

if __name__ == '__main__':

    while True:
        Board.setPWMServoPulse(1, 1500, 1000) # set the pulse width of No.1 servo to 1500 and the running time to 1000mm
        time.sleep(1)
        Board.setPWMServoPulse(1, 2500, 1000) # set the pulse width of No.1 to 2500 and the running time to 1000mm
        time.sleep(1)

        if not start:
            Board.setPWMServoPulse(1, 1500, 1000) # set the pulse width of No.1 servo to 1500,and the running time to 1000mm
            time.sleep(1)
            print('closed')
            break
```

3. Operation Steps

1) Click >_ or press “Ctrl+Alt+T” to open LX terminal.

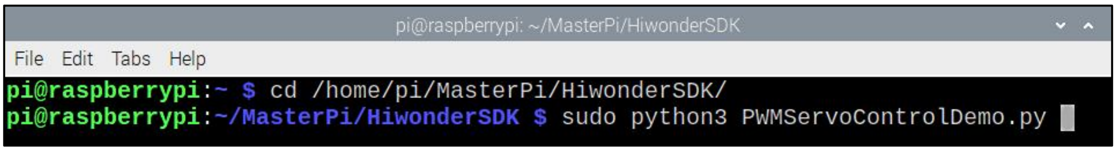


2) Enter “cd /home/pi/MasterPi/HiwonderSDK/” command in LX terminal and press “Enter” to come to the directory of game programs.



```
pi@raspberrypi:~ $ cd /home/pi/MasterPi/HiwonderSDK/
```

3) Then, enter “sudo python3 PWMServoControlDemo.py” command and press “Enter” to start game.



```
pi@raspberrypi:~ $ cd /home/pi/MasterPi/HiwonderSDK/
pi@raspberrypi:~/MasterPi/HiwonderSDK $ sudo python3 PWMServoControlDemo.py
```

4) If want to exit the program, you can press “Ctrl+C”.

4. Project Outcome

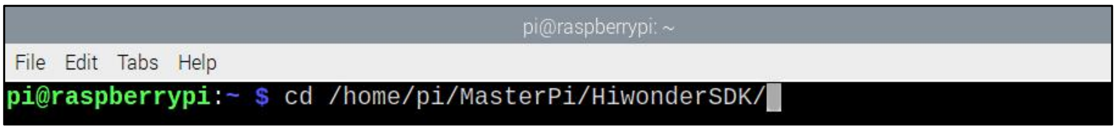
After starting the program, servo will rotate between 90° and 180° repeatedly.

5. Function Extension

5.1 Change servo port

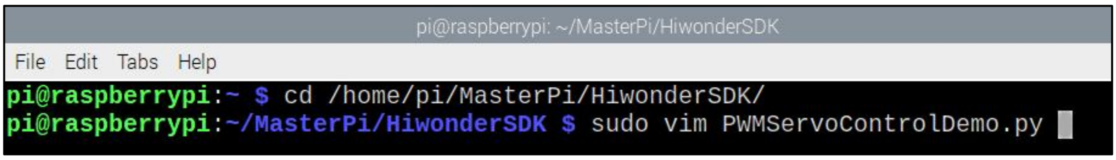
The program defaults to connect PWM servo to No.1 servo port. If want to change servo connection port, please refer to the following operation steps after connecting servo to the specified port.

1) Enter “cd /home/pi/MasterPi/HiwonderSDK/” command and press “Enter” to come to the directory of game programs.



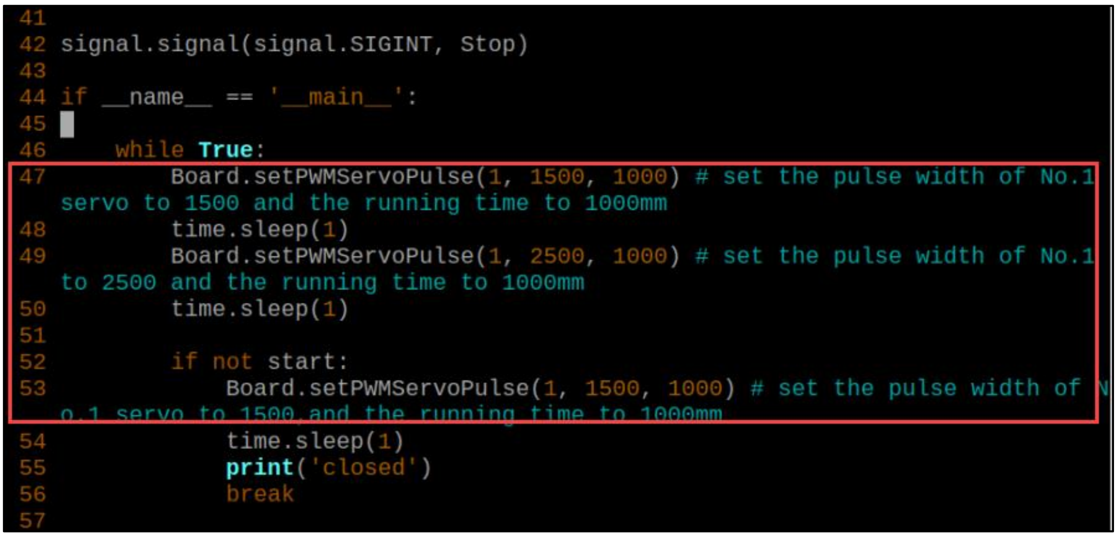
```
pi@raspberrypi:~ $ cd /home/pi/MasterPi/HiwonderSDK/
```

2) Then enter “sudo vim PWMServoControlDemo.py” command and press “Enter” to open program file.



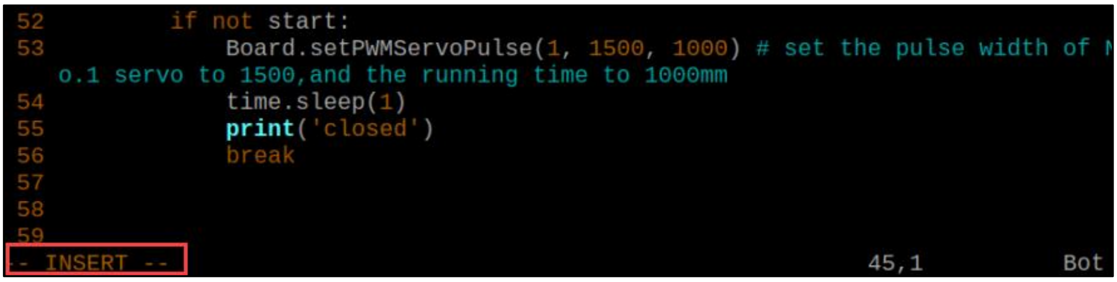
```
pi@raspberrypi:~ $ cd /home/pi/MasterPi/HiwonderSDK/
pi@raspberrypi:~/MasterPi/HiwonderSDK $ sudo vim PWMServoControlDemo.py
```

3) Find the code in the figure shown below:



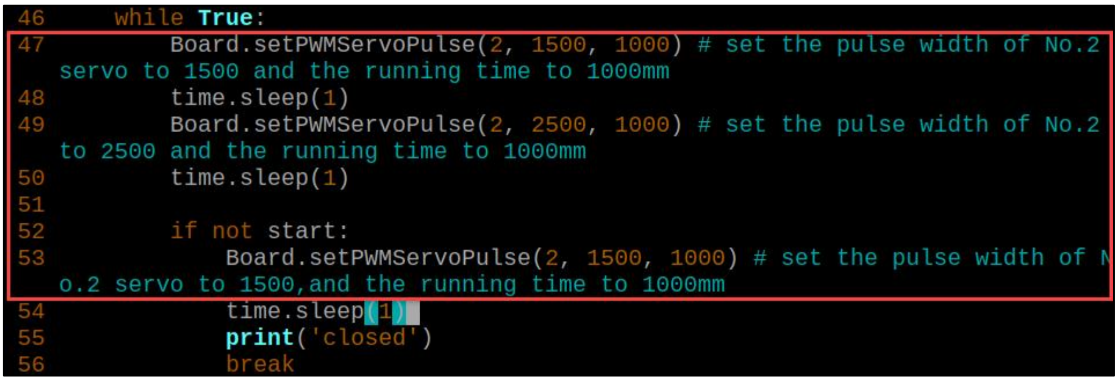
```
signal.signal(signal.SIGINT, Stop)

if __name__ == '__main__':

    while True:
        Board.setPWMServoPulse(1, 1500, 1000) # set the pulse width of No.1
servo to 1500 and the running time to 1000mm
        time.sleep(1)
        Board.setPWMServoPulse(1, 2500, 1000) # set the pulse width of No.1
to 2500 and the running time to 1000mm
        time.sleep(1)

        if not start:
            Board.setPWMServoPulse(1, 1500, 1000) # set the pulse width of N
o.1 servo to 1500,and the running time to 1000mm
            time.sleep(1)
            print('closed')
            break

```

4) Press “i” key. When “INSERT” word appears, which means it has entered the editing mode.



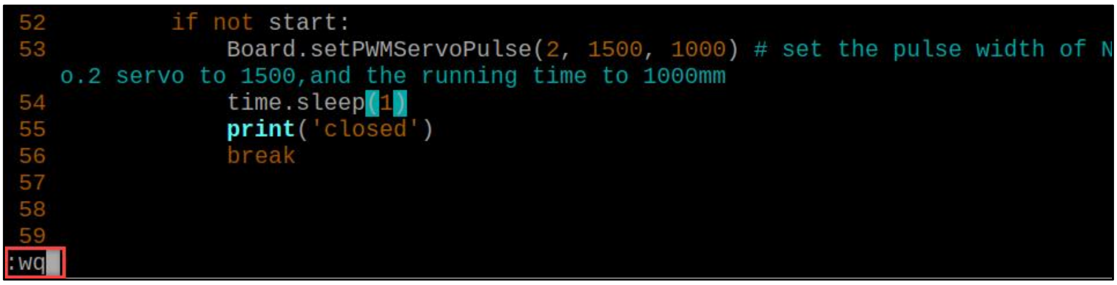
```
        if not start:
            Board.setPWMServoPulse(1, 1500, 1000) # set the pulse width of N
o.1 servo to 1500,and the running time to 1000mm
            time.sleep(1)
            print('closed')
            break



-- INSERT --                                                  45,1          Bot
```

5) Modify “(2, 1500, 1000)” in the parentheses of “Board.setPWMServoPulse (2, 1500, 1000)” to “(1, 1500, 1000)”, as the figure shown below:



```
    while True:
        Board.setPWMServoPulse(2, 1500, 1000) # set the pulse width of No.2
servo to 1500 and the running time to 1000mm
        time.sleep(1)
        Board.setPWMServoPulse(2, 2500, 1000) # set the pulse width of No.2
to 2500 and the running time to 1000mm
        time.sleep(1)

        if not start:
            Board.setPWMServoPulse(2, 1500, 1000) # set the pulse width of N
o.2 servo to 1500,and the running time to 1000mm
            time.sleep(1)
            print('closed')
            break
```

6) After modifying, press “Esc”. Then enter “:wq” and press “Enter” to save and exit.

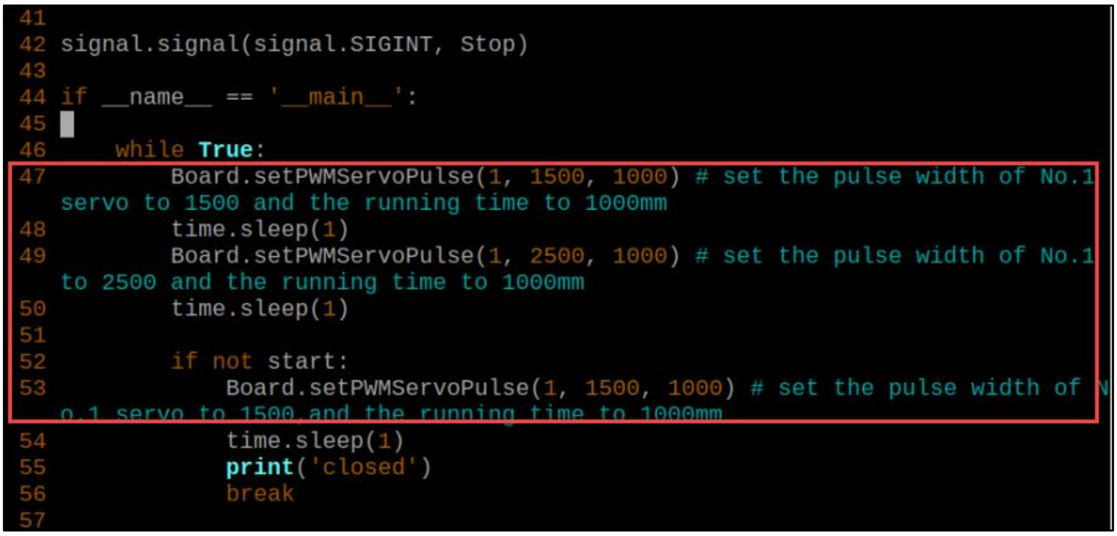


```
        if not start:
            Board.setPWMServoPulse(2, 1500, 1000) # set the pulse width of No.2 servo to 1500,and the running time to 1000mm
            time.sleep(1)
            print('closed')
            break



:wq
```

5.2 Modify rotation angle

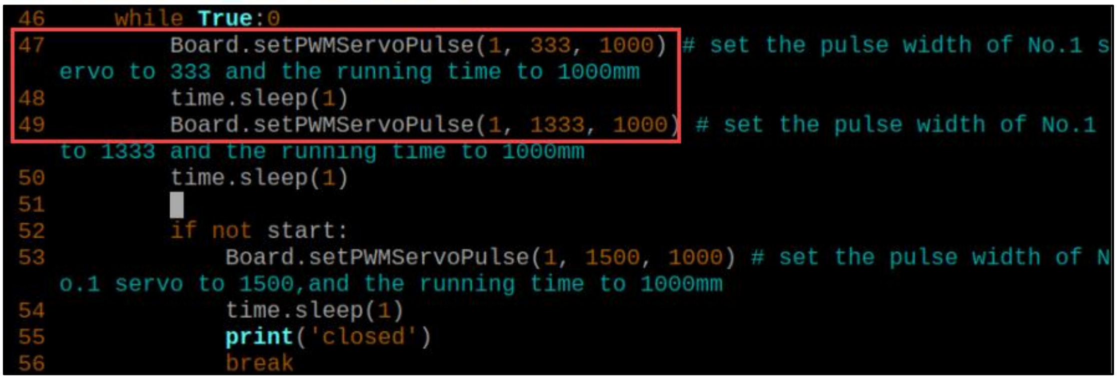
Servo defaults to rotate between 90° and 180° repeatedly. If want to customize servo rotation angle, for example, this example will modify the range of servo rotation angle to 30° -120° , please refer to the following operation steps to modify:

1) Refer to the operation steps in “5.1 Change servo port” to open program file and find the code shown in the figure below:



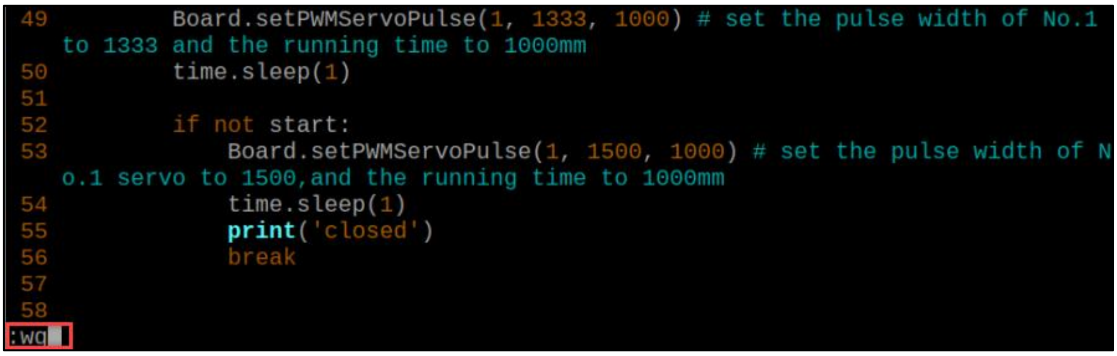
```

signal.signal(signal.SIGINT, Stop)

if __name__ == '__main__':

    while True:
        Board.setPWMServoPulse(1, 1500, 1000) # set the pulse width of No.1 servo to 1500 and the running time to 1000mm
        time.sleep(1)
        Board.setPWMServoPulse(1, 2500, 1000) # set the pulse width of No.1 to 2500 and the running time to 1000mm
        time.sleep(1)

        if not start:
            Board.setPWMServoPulse(1, 1500, 1000) # set the pulse width of No.1 servo to 1500,and the running time to 1000mm
            time.sleep(1)
            print('closed')
            break

```

2) The pulse width of servo rotation ranges from 500 to 2500, which is equivalent to 0° -180° . For example, 1000 pulse width equals 90° , which is 1 pulse width equals 0.09° , so the value setting range is between 333 and 1333.



```
    while True:0
        Board.setPWMServoPulse(1, 333, 1000) # set the pulse width of No.1 servo to 333 and the running time to 1000mm
        time.sleep(1)
        Board.setPWMServoPulse(1, 1333, 1000) # set the pulse width of No.1 to 1333 and the running time to 1000mm
        time.sleep(1)
        
        if not start:
            Board.setPWMServoPulse(1, 1500, 1000) # set the pulse width of No.1 servo to 1500,and the running time to 1000mm
            time.sleep(1)
            print('closed')
            break
```

3) After modifying, press “Esc”. Then enter “:wq” and press “Enter” to save and exit.



```
        Board.setPWMServoPulse(1, 1333, 1000) # set the pulse width of No.1 to 1333 and the running time to 1000mm
        time.sleep(1)

        if not start:
            Board.setPWMServoPulse(1, 1500, 1000) # set the pulse width of No.1 servo to 1500,and the running time to 1000mm
            time.sleep(1)
            print('closed')
            break


:wq
```