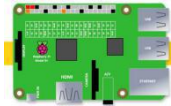
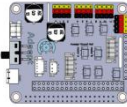



Lesson 5 How to Control a 180° Servo

In this lesson, we will learn how to control a 180-degree servo.

5.1 Components & Parts

Components	Quantity	Picture
Raspberry Pi	1	
Robot HAT	1	
180° Servo	1	

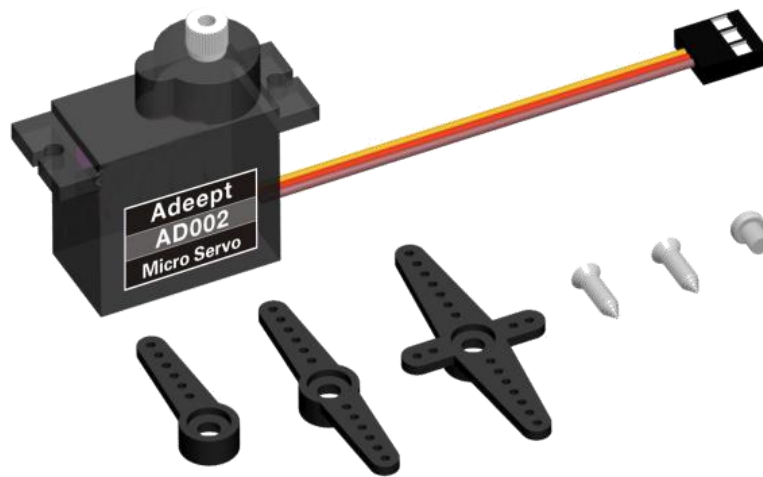
5.2 Introduction of a 180° Servo

What is a servo?

Servo is a position (angle) servo driver, which is suitable for control systems that require constant angle changes and can be maintained. It has been widely used in high-end remote control toys, such as airplane toys, submarine models, and remote control robots.

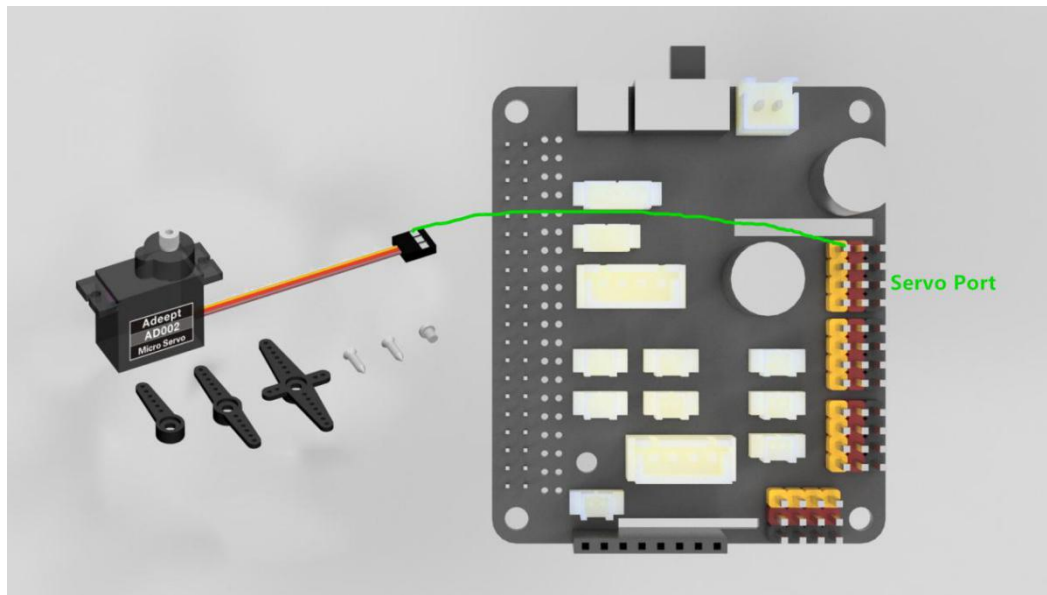
We use a 180° servo in this lesson, which can move between 0° and 180°. Since the 180° servo can be used to control the rotation angle of a certain mechanism via PWM signal, it is a more commonly used module in robot products.

On the Raspberry Pi driver board Robot HAT, there is a PCA9685 chip specially used to control the servo. The Raspberry Pi uses I2C to communicate with the PCA9685. The chip controls the servo by sending pulse signals from the Raspberry Pi which tell the servo where to move. Below is what a 180° servo looks like:



5.3 Schematic Diagram (Circuit Diagram)

The 180° servo module needs to connect to the servo interface on the Robot HAT. The yellow wire is connected to the yellow pin, the red wire to the red pin, and the brown wire to the black pin, as shown below (**connect to pin 0**):



5.3 Controlling a 180° Servo

Run the code

1. Remotely log in to the Raspberry Pi terminal.

```
Linux raspberrypi 4.19.118-v7l+ #1311 SMP Mon Apr 27 14:26:42 BST 2020 armv7l

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Sat Aug 29 08:17:49 2020 from 192.168.3.208

SSH is enabled and the default password for the 'pi' user has not been changed.
This is a security risk - please login as the 'pi' user and type 'passwd' to set
a new password.

pi@raspberrypi:~ $
```

2. Enter the command, press Enter to go to the folder where the program is located:

```
cd adeept_darkpaw/server/
```

```
pi@raspberrypi:~ $ cd adeept_darkpaw/server/
pi@raspberrypi:~/adeept_darkpaw/server $
```

3. Check files under the current directory:

```
ls
pi@raspberrypi:~/adeept_darkpaw/server $
pi@raspberrypi:~/adeept_darkpaw/server $ ls
app.py          functions.py    LED.py          raspi-config.py SpiderG.py
base_camera.py  info.py        move.py         robotLight.py   switch.py
camera_opencv.py instruction.txt mpu6050Test.py server.py        test.py
dist            Instruction.txt PID.py          serverTest.py   webServer.py
FPV.py          Kalman_filter.py __pycache__     servo.py
```

4. Enter the command and press Enter to run the program:

```
sudo python3 servo.py
```

```
pi@raspberrypi:~/adeept_darkpaw/server $
pi@raspberrypi:~/adeept_darkpaw/server $ sudo python3 servo.py
```

5. After running the program successfully, you can see the servo rotates regularly.

6. To terminate the program, you can press the shortcut "Ctrl + C".

5.4 Code

For the complete code, refer to [servo.py](#) or [RPIServo.py](#).

Control the servo to rotate to a certain angle:

```
1. import Adafruit_PCA9685 # Import the library used to communicate with PCA9685
2. import time
3.
4. pwm = Adafruit_PCA9685.PCA9685() # Instantiate the object used to control the PWM
5. pwm.set_pwm_freq(50) # Set the frequency of the PWM signal
```

```
6.  
7. while 1:    # Make the servo connected to the No. 3 servo port on the Robot HAT reciprocate  
8.     pwm.set_pwm(0, 0, 300)  
9.     time.sleep(1)  
10.    pwm.set_pwm(0, 0, 400)  
11.    time.sleep(1)
```

In the above code, `set_pwm_freq(50)` is used to set the PWM frequency to 50Hz. This setting depends on the model of the servo. Servos used on Adeept robot products are controlled by a 50Hz PWM signal. For other servos, the value depends on the servo specifications.

`pwm.set_pwm(0, 0, 300)` is to control the rotation of a servo to a certain position. To explain the three values:

"0" is the port number of the servo, corresponding to the number marked on the Robot HAT driver board (when connecting the servo, pay attention not to insert the ground wire, VCC and signal wire inversely – correct way is servo's brown wire to the black of HAT, red to red, and yellow to yellow).

"0" is the deviation value for controlling the servo's rotation, which is not used in the program here. This function is used to correct the deviation (for the cause of the error, refer to the precautions for structural assembly).

"300" is the value of the PWM duty cycle to be set. The rotation angle varies from servos. The PWM duty cycle range of the servo used in the robot is about 100 to 560, which corresponds to the range of rotating angle about 0° to 180°.

The code above does not control the rotation speed of the servo. To control the robot to slowly swing back and forth between two positions, you need to use the method of increasing or decreasing variables to control the servo.

Control the servo to move slowly

```
1. import Adafruit_PCA9685
```

2. **import** time

Import dependent libraries as shown below:

1. `pwm = Adafruit_PCA9685.PCA9685()`
2. `pwm.set_pwm_freq(50)`

`Set_pwm_freq(50)` is used to set the PWM frequency to 50Hz. The value depends on the servo model. Servos used here are controlled by a 50Hz PWM signal. For other servos, please set the value based on the servo's specifications.

1. **while** 1:
2. **for** i **in** range(0, 100):
3. `pwm.set_pwm(0, 0, (300+i))`
4. `time.sleep(0.05)`
5. **for** i **in** range(0,100):
6. `pwm.set_pwm(0, 0, (400-i))`
7. `time.sleep(0.05)`

`pwm.set_pwm(0, 0, 300+i)` is used to control the servo to rotate to a certain position. "0" is the port number of the servo, corresponding to the number marked on the Robot HAT driver board (when connecting the servo, pay attention not to insert the ground wire, VCC and signal wire inversely – correct way is servo's brown wire to the black of HAT, red to red, and yellow to yellow). "0" is the deviation value for controlling the servo's rotation. "300+i" is the value of the PWM duty cycle to be set. The rotation angle varies from servos. The PWM duty cycle range of the servo used in the robot is about 100 to 560, which corresponds to the range of rotating angle about 0° to 180°.

By the formula $(\text{duty ratio} - 100) / 2.55$, the angle corresponding to the current duty cycle can be calculated.

The code above can be used to control the servo to rotate slowly back and forth between 300 and 400, but it has great drawbacks. When the program is executed to the slow-motion part of the servo, it will be blocked, which will seriously affect the program. Therefore, a multi-threaded solution is provided in the robot product program to solve this problem.

Non-blocking control method

Find the *RPIservo.py* file in the *server* folder of the robot's code package, copy to the same folder as the program you want to run, and then you can run it in your program.

```
1. import RPIservo # Import a library that uses multiple threads to control the steering gear
2. import time
3.
4. sc = RPIservo.ServoCtrl() # Instantiate the object that controls the servo
5. sc.start() # Start this thread, when the servo does not move, the thread is suspended
6.
7. while 1:
8.     sc.singleServo(3, -1, 2)
9.     time.sleep(1)
10.    sc.stopWiggle()
11.
12.    sc.singleServo(3, 1, 2)
13.    time.sleep(1)
14.    sc.stopWiggle()
```

Use the above code to control the servo to reciprocate, except for *time.sleep()*, which will not block the running of the context program.

Call *singleServo()* to start and *stopWiggle()* to stop motion. *singleServo()* requires three parameter: the port numbers of the servo to be controlled (e.g., "3" means to control the No. 3 servo connected to Robot HAT), the direction of the servo (1 or -1), and the speed of the servo (the larger the value, the faster the movement).