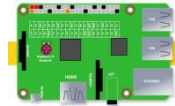




Lesson 4 How to Control 180° Servo

In this lesson, we will learn how to control 180° Servo.

4.1 Components used in this course

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

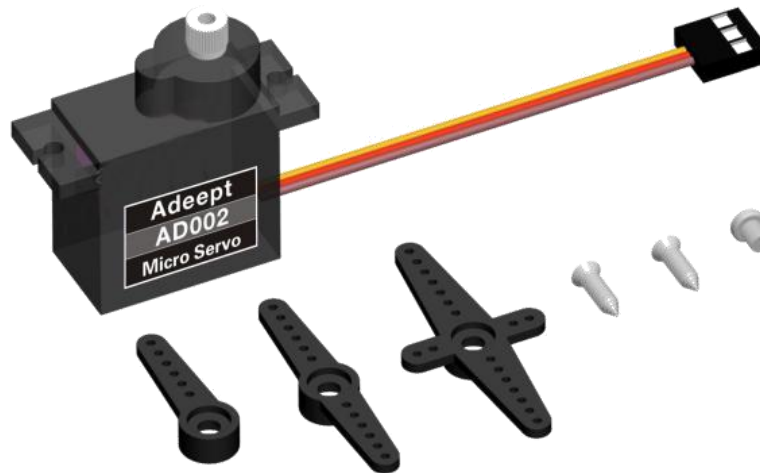
4.2 Introduction of 180° Servo

What is a servo?

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

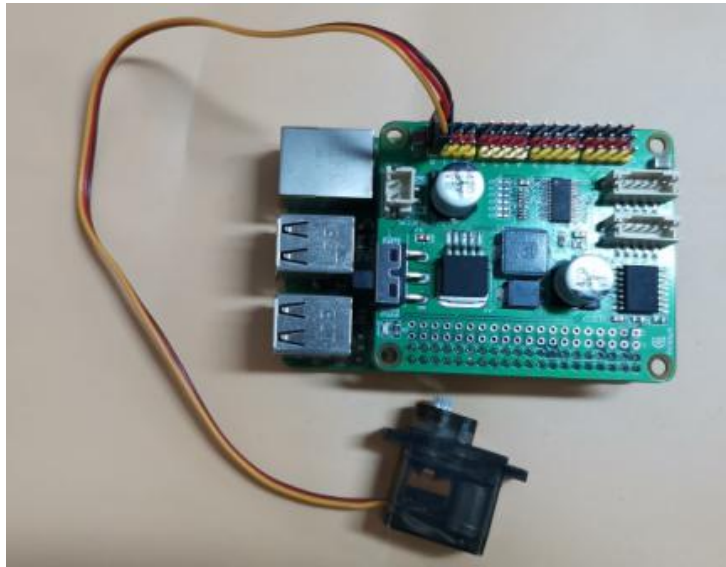
We use a 180° servo in this lesson, which can move between 0° and 180°. Since the 180° servo can use the PWM signal to control the rotation angle of a certain mechanism, it is a more commonly used module in robot products.

On the Raspberry Pi driver board Arm HAT, there is a PCA9685 chip specially used to control the servo. The Raspberry Pi uses I2C to communicate with the PCA9685. It controls the servo by sending pulse signals from the microcontroller. These pulses tell the servo mechanism of the servo where to move. The picture of the 180° servo is as follows:



4.3 Wiring diagram (Circuit diagram)

When the 180° Servo module is in use, it needs to be connected to the servo interface on the Arm HAT driver board. The yellow wire is connected to the yellow pin, the red wire is connected to the red pin, and the brown wire is connected to black pin, as shown below (connect to pin 0):



4.3 How to control 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 and press Enter to enter the folder where the program is located:

```
cd adeept_roboticarm/
```

```
pi@raspberrypi:~$ cd adeept_roboticarm/
pi@raspberrypi:~/adeept_roboticarm$
```

3. View the contents of the current directory file:

```
ls
```

```
pi@raspberrypi:~/adeept_roboticarm$
pi@raspberrypi:~/adeept_roboticarm$ ls
1_servo.py 2_joystick.py 3_initservo.py 4_arm.py PCF8591.py setup.py
pi@raspberrypi:~/adeept_roboticarm$
```

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

```
sudo python3 1_servo.py
```

```
pi@raspberrypi:~/adeept_roboticarm$
pi@raspberrypi:~/adeept_roboticarm$ sudo python3 1_servo.py
```

5. After running the program successfully, you will observe that the servo will rotate regularly.

6. When you want to terminate the running program, you can press the shortcut key "Ctrl + C" on the keyboard.

4.4 The main code program

Complete code refer to [1_servo.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 Arm HAT drive board reciprocate
8.     pwm.set_pwm(3, 0, 300)
```

```
9.    time.sleep(1)
10.   pwm.set_pwm(3, 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. The servo used by our robot products needs to be controlled by a 50Hz PWM signal. If you use For other servos, this value needs to be set by referring to the specific servo documentation.

`Pwm.set_pwm(3, 0, 300)` This method is used to control the rotation of a servo to a certain position, where 3 is the port number of the servo, which corresponds to the number marked on the Arm HAT driver board, but pay attention When the servo is connected to the drive board, do not insert the ground wire, VCC and signal wire in the reverse direction, brown to black, red to red, and yellow to yellow; 0 is the deviation value for controlling the rotation of the servo, which is not used in our program This function is used to correct the deviation (the cause of the error of the servo can be referred to the precautions for structural assembly); 300 is the value of the PWM duty cycle you want to set. Depending on the servo, this value represents a different servo angle. The PWM duty cycle range of the servo we use is about 100 to 560, which corresponds to a rotation range of about 0° to 180°.

Using the above code to control the servo does not control the rotation speed of the servo. If we want a servo to slowly swing back and forth between two positions, we need to use the method of increasing or decreasing variables to control the servo.

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