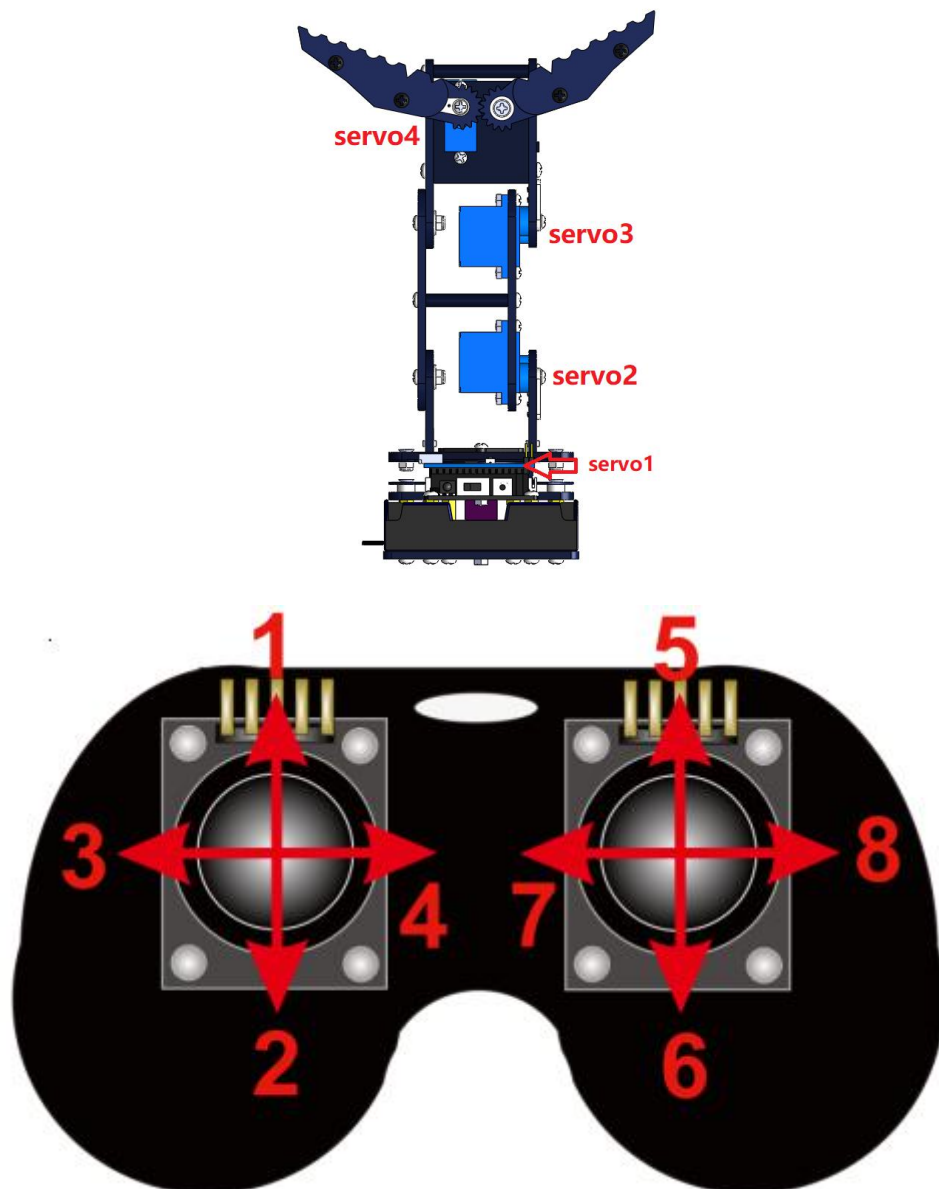


Lesson 8 Joystick control robotic arm

8.1 Manipulate the robotic arm

Description of the joystick handle:



1, 2: Control servo2 forward and reverse rotation.

3, 4: Control servo0 to rotate forward and backward.

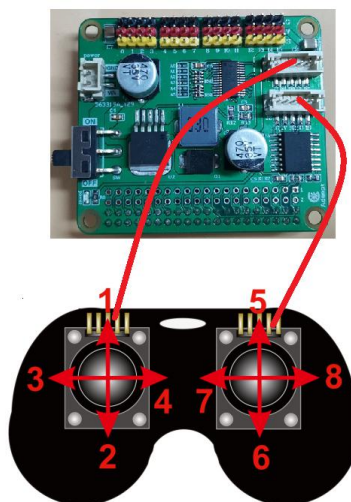
5, 6: Control servo3 to rotate forward and backward.

7, 8: Control servo4 forward and reverse rotation.

8.2 Wiring diagram (Circuit diagram)

Servo	Port
Servo1	Arm HAT Port 0
Servo2	Arm HAT Port 1
Servo3	Arm HAT Port 2
Servo4	Arm HAT Port 3

The left stick is connected to the upper interface of the Arm HAT, and the right stick is connected to the lower interface of the Arm HAT.



8.3 Run the code

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 4_arm.py
```

5. After running the program successfully, You can control the robotic arm through the joystick.

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

8.4 Memory mode

After running the [5_arm_memory.py](#) program, when you press the left joystick, the robotic arm starts to perform the memory function, and the Raspberry Pi command line prints "L-pressed" at this time. Operate the robotic arm as required, and the robotic arm will save the action. Press the left joystick again to end the memory function.

After pressing the right joystick, "R-pressed" will be printed on the Raspberry Pi command line. At this time, the robotic arm is in the memory playback state, and the robotic arm will execute the saved action. After pressing the right rocker button, the robotic arm will execute the saved action once. After the execution is completed, press the button again, and the robotic arm will repeat the action again.

8.5 The main code program

Complete code refer to [4_arm.py](#) .

```
1. import PCF8591 as ADC
2. import Adafruit_PCA9685
3. import time
4.
5. pwm = Adafruit_PCA9685.PCA9685()
6. pwm.set_pwm_freq(50)
7.
8. angle = [90, 90, 90, 90]
9. speed = 7 # servo rotation speed.
10. forward = 1
```

```
11. reverse = -1
12.
13. def setup():
14.     ADC.setup(0X48)
15.
16. def ctrl_range(raw, max_genout, min_genout):
17.     if raw > max_genout:
18.         raw_output = max_genout
19.     elif raw < min_genout:
20.         raw_output = min_genout
21.     else:
22.         raw_output = raw
23.     return int(raw_output)
24.
25. def rotation(ID, direction, speed):
26.     global angle
27.     if direction == 1:
28.         angle[ID] += speed
29.     else:
30.         angle[ID] -= speed
31.     if angle[ID] >= 500:
32.         angle[ID] = 500
33.     if angle[ID] <= 0:
34.         angle[ID] = 0
35.     pwm.set_pwm(ID, 0, angle[ID])
36.
37. def move_servo(value):
38.     if value == 1:           # servo 1
39.         rotation(0, forward, speed) # (servo_ID, direction, speed)
40.     elif value == -1:
41.         rotation(0, reverse, speed)
42.     elif value == 2:        # servo 2
43.         rotation(1, forward, speed)
44.     elif value == -2:
45.         rotation(1, reverse, speed)
46.     elif value == 3:        # servo 3
47.         rotation(2, forward, speed)
48.     elif value == -3:
49.         rotation(2, reverse, speed)
50.     elif value == 3:        # servo 4
51.         rotation(3, forward, speed)
52.     elif value == -3:
```

```
53.     rotation(3, reverse, speed)
54. else:
55.     pass
56.
57. def joystick(): #get joystick result
58.     # state = ['home', 'up', 'down', 'left', 'right']
59.     value = 0
60.     if ADC.read(0) <= 30:    # servo 1
61.         value = 1
62.     elif ADC.read(0) >= 210:
63.         value = -1
64.
65.     if ADC.read(1) <= 30:    # servo 2
66.         value = 2
67.     elif ADC.read(1) >= 210:
68.         value = -2
69.
70.     if ADC.read(2) <= 30:    # servo 3
71.         value = 3
72.     elif ADC.read(2) >= 210:
73.         value = -3
74.
75.     if ADC.read(3) <= 30:    # servo 4
76.         value = 4
77.     elif ADC.read(3) >= 210:
78.         value = -4
79.
80.     return value
81.
82. def loop():
83.     value = joystick()
84.     move_servo(value)
85.     time.sleep(0.1)
86.
87. def destroy():
88.     pass
89.
90. if __name__ == '__main__':
91.     try:
92.         while True:
93.             loop()
94.     except:
```

95.	<code>destroy()</code>
-----	------------------------