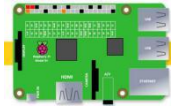
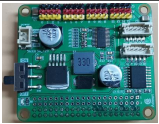



Lesson 5 How to Control the Joystick

In this lesson, we will learn how to control the Joystick.

5.1 Components used in this course

Components	Quantity	Picture
Raspberry Pi	1	
Arm HAT	1	
Joystick	1	

5.2 Introduction of the Joystick

This PS2 game dual axis joystick module is made of high quality metal PS2 joystick potentiometer, with (X, Y) 2 axis analog output, (Z) 1 button digital output.

5.3 Wiring diagram (Circuit diagram)

GND: Connect to the negative pole of the power supply.

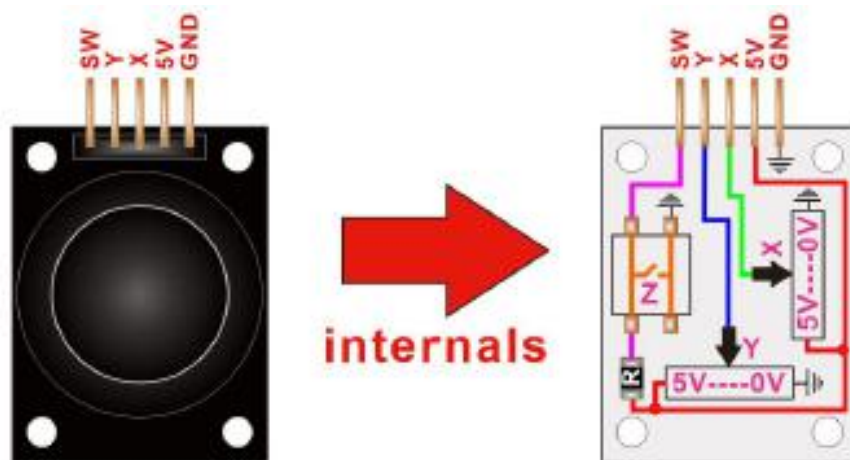
5V: connected to the positive pole of the power supply.

X: X axis analog value output.

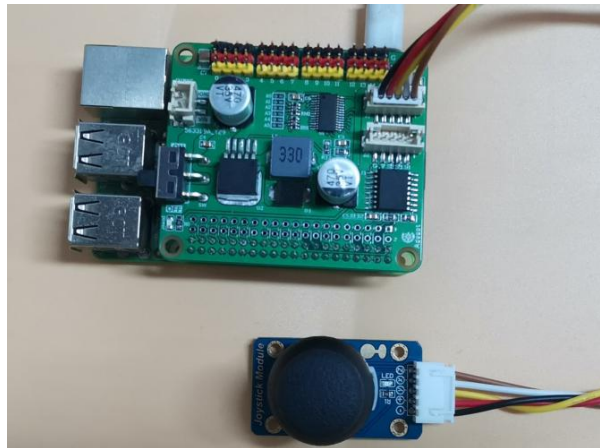
Y: Y-axis analog value output.

SW: Z-axis output (button).

The joystick is made up of two passive potentiometers (variable resistors) and a push button, it is made by mounting two potentiometers at a 90 degrees angle. The potentiometers are connected to a short stick centered by springs. This module produces an output of around 2.5V from X and Y when it is in resting position. Moving the joystick will cause the output to vary from 0v to 5V depending on its direction. This joystick also has a select button that is actuated when the joystick is pressed down.



Link the joystick to the interface on the upper side of the Arm HAT.



5.3 How to control the joystick

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 2_joystick.py
```

```
pi@raspberrypi:~/adeept_roboticarm $  
pi@raspberrypi:~/adeept_roboticarm $  
pi@raspberrypi:~/adeept_roboticarm $ sudo python3 2_joystick.py  
home  
up  
home  
down  
home  
█
```

5. After running the program successfully, Move the joystick and you will see the corresponding direction printed out.

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

5.4 The main code program

Complete code refer to [2_joystick.py](#) .

```
1. import RPi.GPIO as GPIO  
2. import PCF8591 as ADC  
3. import time  
4.  
5. btn = 12    # Define button pin  
6.  
7. def setup():  
8.     ADC.setup(0X48)
```

```
9.     GPIO.setmode(GPIO.BOARD)      # Numbers GPIOs by physical location
10.    GPIO.setup(btn, GPIO.IN, pull_up_down=GPIO.PUD_UP) # Setup button pin as input an pull
    it up
11.
12. def direction():    #get joystick result
13.     state = ['home', 'up', 'down', 'left', 'right', 'pressed']
14.     i = 0
15.
16.     if ADC.read(0) <= 5:
17.         i = 1        #up
18.     if ADC.read(0) >= 200:
19.         i = 2        #down
20.
21.     if ADC.read(1) <= 5:
22.         i = 3        #left
23.     if ADC.read(1) >= 200:
24.         i = 4        #right
25.
26.     return state[i]
27.
28. def loop():
29.     status = ''
30.     while True:
31.         tmp = direction()
32.         if tmp != None and tmp != status:
33.             print(tmp)
34.             status = tmp
35.             time.sleep(0.1)
36.
37. def destroy():
38.     GPIO.cleanup()              # Release resource
39.
40. if __name__ == '__main__':    # Program start from here
41.     setup()
42.     try:
43.         loop()
44.     except KeyboardInterrupt:    # When 'Ctrl+C' is pressed, the child program destroy()
        will be executed.
45.         destroy()
```