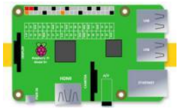
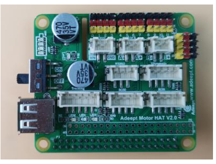
Lesson 6 How to Control DC Motor

In this lesson, we will learn how to control DC Motor.

6.1 Components used in this course

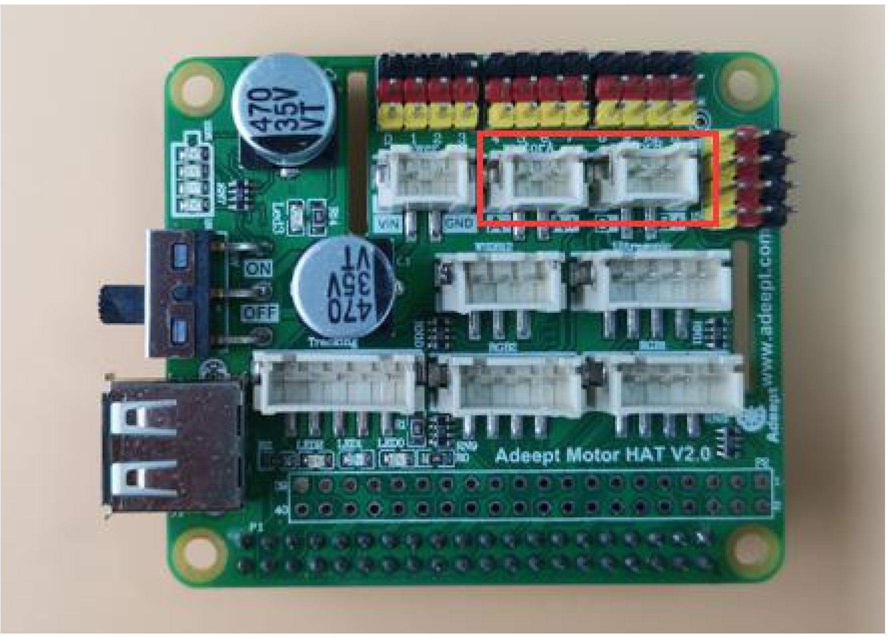
Components	Quantity	Picture
Raspberry Pi	1	
Robot HAT	1	
DC Motor	1	

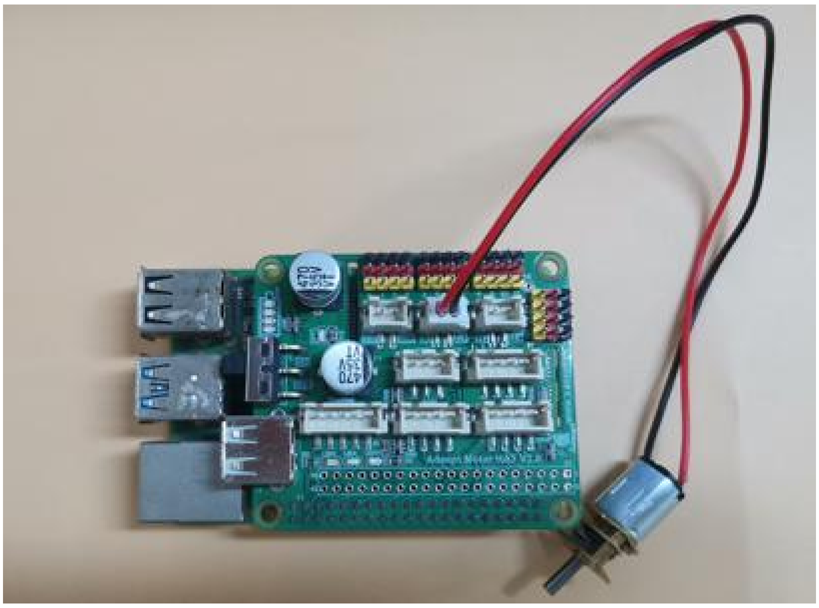
6.2 The introduction of DC Motor

RaspTankPro robot products use DC motor as a power device. DC motor is a device that converts DC electrical energy into mechanical energy. It is widely used to drive various equipment, such as electric fans, remote control cars, electric windows, etc.The DC motor is very suitable as the walking mechanism of the robot.

6.3 Wiring diagram (Circuit diagram)

When the DC Motor module is in use, it needs to be connected toOne of MotorA, MotorB interface on the Robot HAT drive board. as shown below:

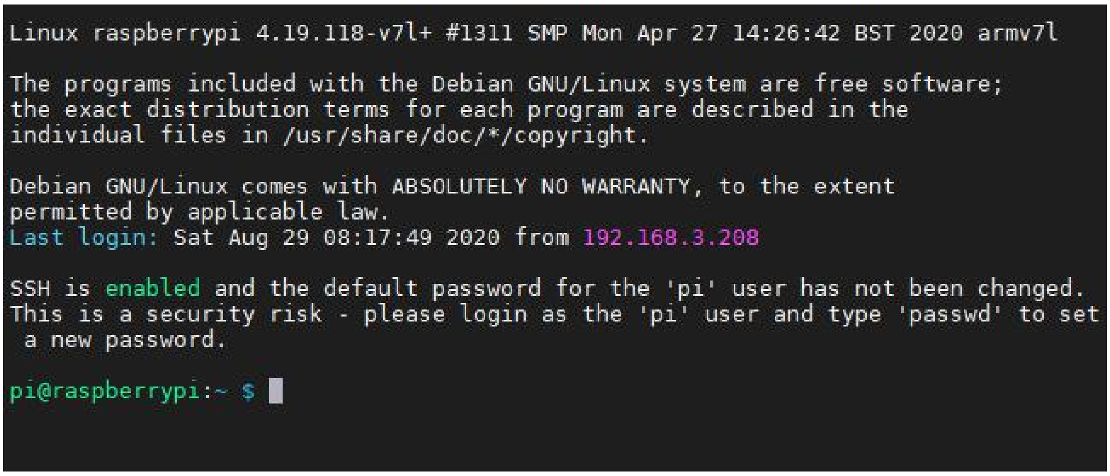




6.4 How to control Motor

Run the code

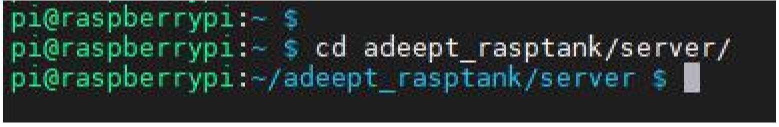
1. Remotely log in to the Raspberry Pi terminal.



```
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:

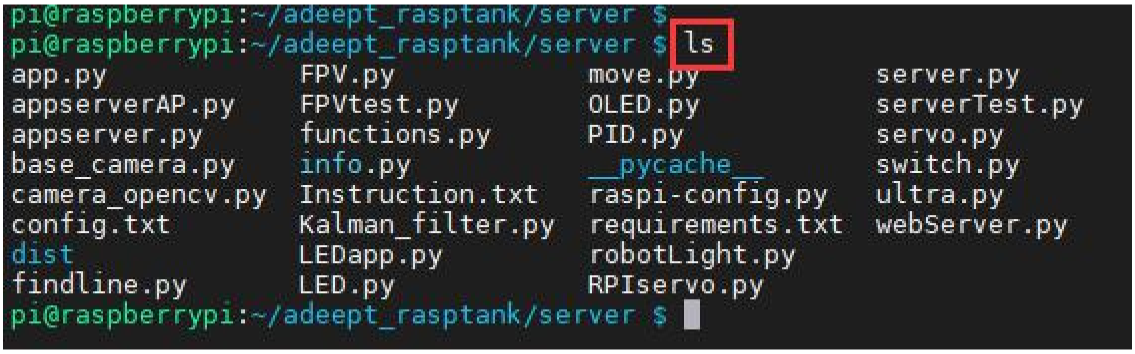
```
cd adeept_rasptank/server/
```



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

3. View the contents of the current directory file:

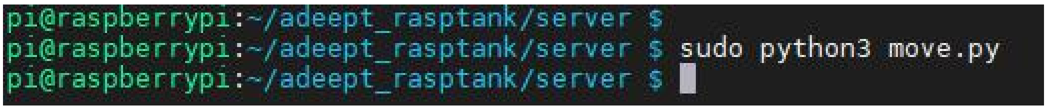
```
ls
```



```
pi@raspberrypi:~/adeept_rasptank/server $
pi@raspberrypi:~/adeept_rasptank/server $ ls
app.py             FPV.py            move.py           server.py
appserverAP.py     FPVtest.py        OLED.py           serverTest.py
appserver.py       functions.py      PID.py            servo.py
base_camera.py     info.py           __pycache__       switch.py
camera_opencv.py   Instruction.txt   raspi-config.py   ultra.py
config.txt         Kalman_filter.py  requirements.txt  webServer.py
dist               LEDapp.py         robotLight.py
findline.py        LED.py            RPIservo.py
pi@raspberrypi:~/adeept_rasptank/server $
```

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

```
sudo python3 move.py
```



```
pi@raspberrypi:~/adeept_rasptank/server $
pi@raspberrypi:~/adeept_rasptank/server $ sudo python3 move.py
pi@raspberrypi:~/adeept_rasptank/server $
```

5. After running the program successfully, you will observe that the Motor will rotate for about 1 second and then stop, and the program will also stop. If you need the motor to rotate again, you need to run the program again.

6.5 The main code program of this lesson

For the complete code, please refer to the file move.py.

```
import time
import RPi.GPIO as GPIO

Motor_A_EN    = 4
Motor_A_Pin1  = 26
Motor_A_Pin2  = 21
```

Import the dependent library, where Motor_EN = 4, Motor_Pin1 = 26, Motor_Pin2 = 21 are the parameters of the corresponding interface motorA.

```
def setup():#Motor initialization
    global pwm_A
    GPIO.setwarnings(False)
    GPIO.setmode(GPIO.BCM)
    GPIO.setup(Motor_A_EN, GPIO.OUT)
    GPIO.setup(Motor_A_Pin1, GPIO.OUT)
    GPIO.setup(Motor_A_Pin2, GPIO.OUT)

    motorStop()
    try:
        pwm_A = GPIO.PWM(Motor_A_EN, 1000)
    except:
        pass
```

Set the motor initialization function.

```
def destroy():
    motorStop()
    GPIO.cleanup()             # Release resource

```

Set the motor stop function.

```
def move(speed, direction, turn, radius=0.6):   # 0 < radius <= 1
    #speed = 100
    if direction == 'forward':
        if turn == 'right':
            motor_left(0, left_backward, int(speed*radius))
            motor_right(1, right_forward, speed)
        elif turn == 'left':
            motor_left(1, left_forward, speed)
            motor_right(0, right_backward, int(speed*radius))
        else:
            motor_left(1, left_forward, speed)
            motor_right(1, right_forward, speed)
    else:
        pass
```

Set the motor rotation function.

```
if __name__ == '__main__':
```

```
    try:
        speed_set = 60
        setup()
        move(speed_set, 'forward', 'no', 0.8)
        time.sleep(1.3)
        motorStop()
        destroy()
    except KeyboardInterrupt:
        destroy()
```

Instantiate the object.