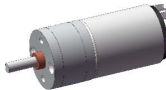


# Lesson 10 Control the DC Motor to Work

## 10.1 Overview

In this lesson, we'll learn to control a DC motor with a Raspberry Pi and Adeept Robot HAT V3.2. We'll cover components, principles, wiring, run a Python program for forward - backward rotation, and analyze the code. By the end, you can control the motor's direction and speed.

## 10.2 Required Components

Components	Quantity	Picture
Raspberry Pi	1	
Adeept Robot HAT V3.2	1	
DC Motor	1	

## 10.3 Principle Introduction

### DC Motor Working Principle

A DC motor is an electrical machine that converts electrical energy into mechanical energy. It operates based on the principle of electromagnetic induction. When an electric current passes

through the coil of the DC motor placed in a magnetic field, a force is exerted on the coil according to Fleming's left - hand rule. This force causes the coil to rotate.

### Speed Control Principle

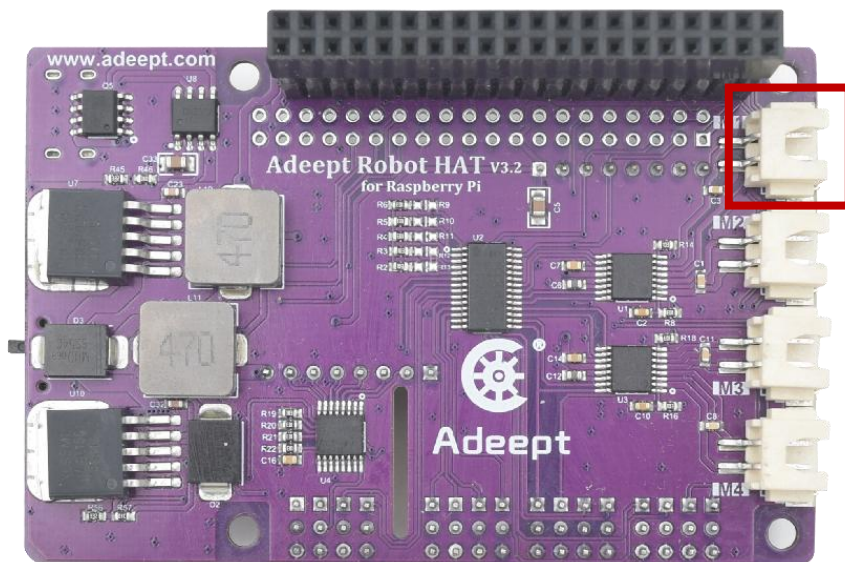
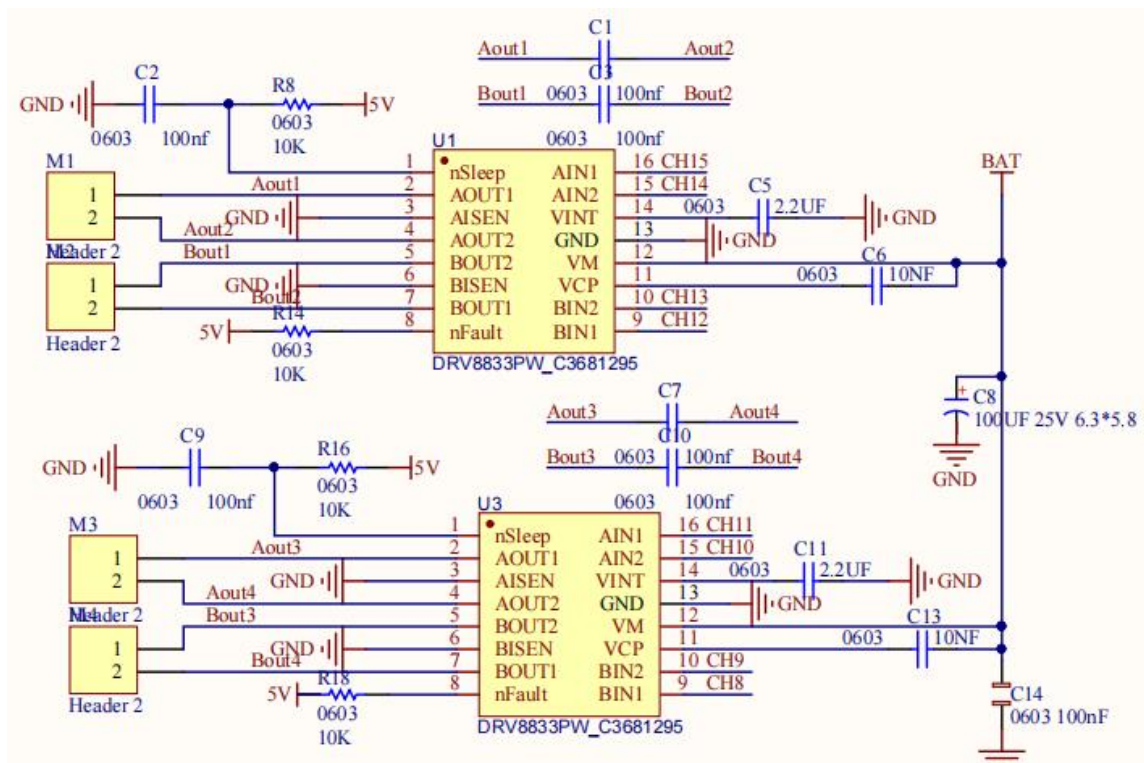
The speed of a DC motor can be controlled in several ways. In this project, we use Pulse Width Modulation (PWM). PWM is a technique for controlling the average value of a voltage (or current) by rapidly switching it on and off. The ratio of the time the signal is on (high) to the total period of the signal is called the duty cycle.

For a DC motor, a higher duty cycle means the motor is supplied with power for a longer time within a period, resulting in a higher average voltage across the motor terminals. According to the motor's characteristics, a higher voltage leads to a higher rotational speed. Conversely, a lower duty cycle reduces the average voltage, and the motor rotates more slowly.

In the code, we use the map function to map the desired speed value (ranging from 0 - 100 in our case) to a value between 0 - 1.0, which is used to set the throttle value of the DC motor object. The throttle value represents the duty cycle of the PWM signal sent to the motor, thus controlling its speed.

## 10.4 Wiring Diagram

When the DC Motor module is in use, it needs to be connected to the M1 interface on the Adept Robot HAT V3.2 expansion board.



## 10.5 Demonstration

1. **Remotely log:** Remotely log in to the Raspberry Pi terminal.
2. **Navigate to the Program Folder:** Enter the following command in the terminal and press Enter to access the folder where the program is located:

```
cd Adeept_PiCar-Pro/Examples/04_Motor/
```

```
pi@raspberrypi:~ $ cd Adeept_PiCar-Pro/Examples/04_Motor/  
pi@raspberrypi:~/Adeept_PiCar-Pro/Examples/04_Motor $
```

3. **View Directory Contents:** Type "ls" in the terminal and press Enter. This will display all the files in the current directory, ensuring that the "**MotorCtrl.py**" file is present:

```
ls
```

```
pi@raspberrypi:~/Adeept_PiCar-Pro/Examples/04_Motor $ ls  
MotorCtrl.py
```

4. **Run the Program:** Enter the command below and press Enter to start the Motor control program:

```
sudo python3 MotorCtrl.py
```

```
pi@raspberrypi:~/Adeept_PiCar-Pro/Examples/04_Motor $ sudo python3 MotorCtrl.py  
Forward  
Backward
```

5. **Observation and Termination:** Once the program runs successfully, you'll observe the DC motor rotating forward and backward. The program first sets the motor to rotate forward at a speed of 50 (scaled to the appropriate duty - cycle value) for 2 seconds, then rotates it backward at the same speed for 2 seconds. This process repeats 10 times. To stop the running program, simply press the "**Ctrl + C**" shortcut on the keyboard. This action will stop the motor and release the PCA9685 resources.

## 10.6 Code

Complete code refer to [MotorCtrl.py](#)

```
01 #!/usr/bin/env/python3  
02 # File name : MotorCtrl.py
```

```
03 # Website      : www.Adeept.com
04 # Author       : Adeept
05 # Date        : 2025/03/6
06 import time
07 from board import SCL, SDA
08 import busio
09 from adafruit_pca9685 import PCA9685
10 from adafruit_motor import motor
11
12 MOTOR_M1_IN1 = 15      #Define the positive pole of M1
13 MOTOR_M1_IN2 = 14      #Define the negative pole of M1
14 MOTOR_M2_IN1 = 12      #Define the positive pole of M2
15 MOTOR_M2_IN2 = 13      #Define the negative pole of M2
16 MOTOR_M3_IN1 = 11      #Define the positive pole of M3
17 MOTOR_M3_IN2 = 10      #Define the negative pole of M3
18 MOTOR_M4_IN1 = 8       #Define the positive pole of M4
19 MOTOR_M4_IN2 = 9       #Define the negative pole of M4
20
21
22 def map(x,in_min,in_max,out_min,out_max):
23     return (x - in_min)/(in_max - in_min) *(out_max - out_min) +out_min
24
25
26 i2c = busio.I2C(SCL, SDA)
27 pwm_motor = PCA9685(i2c, address=0x5f)
28 pwm_motor.frequency = 50
29
30 motor1 = motor.DCMotor(pwm_motor.channels[MOTOR_M1_IN1],pwm_motor.channels[MOTOR_M1_IN2] )
31 motor1.decay_mode = (motor.SLOW_DECAY)
32 motor2 = motor.DCMotor(pwm_motor.channels[MOTOR_M2_IN1],pwm_motor.channels[MOTOR_M2_IN2] )
33 motor2.decay_mode = (motor.SLOW_DECAY)
34 motor3 = motor.DCMotor(pwm_motor.channels[MOTOR_M3_IN1],pwm_motor.channels[MOTOR_M3_IN2] )
35 motor3.decay_mode = (motor.SLOW_DECAY)
36 motor4 = motor.DCMotor(pwm_motor.channels[MOTOR_M4_IN1],pwm_motor.channels[MOTOR_M4_IN2] )
37 motor4.decay_mode = (motor.SLOW_DECAY)
38
39 def Motor(channel,direction,motor_speed):
40     if motor_speed > 100:
41         motor_speed = 100
42     elif motor_speed < 0:
43         motor_speed = 0
44     speed = map(motor_speed, 0, 100, 0, 1.0)
45     if direction == -1:
46         speed = -speed
47
48
49     if channel == 1:
50         motor1.throttle = speed
51     elif channel == 2:
52         motor2.throttle = speed
53     elif channel == 3:
54         motor3.throttle = speed
55     elif channel == 4:
56         motor4.throttle = speed
57
58
```

```
59 def motorStop():#Motor stops
60     motor1.throttle = 0
61     motor2.throttle = 0
62     motor3.throttle = 0
63     motor4.throttle = 0
64
65 def destroy():
66     motorStop()
67     pwm_motor.deinit()
68
69
70 if __name__ == '__main__':
71     try:
72
73         for i in range(10):
74             speed_set = 50
75             Motor(1, 1, speed_set)
76             print("Forward")
77             time.sleep(2)
78             Motor(1, -1 ,speed_set)
79             print("Backward")
80             time.sleep(2)
81         destroy()
82     except KeyboardInterrupt:
83         destroy()
84
85
86
87
88
```

## Code explanation

### Initialization Stage:

Connect the PCA9685 module (address 0x5F) via the I2C protocol. Set the PWM frequency to 50Hz and initialize 4 DC motors: M1 - M4 (bind them to GPIO pins respectively and configure them in slow decay mode).

### Loop Control Process:

The main program loops 10 times:

Stage 1:Motor M1 rotates forward at 50% speed → Run for 2 seconds.

Stage 2:Motor M1 rotates backward at 50% speed → Run for 2 seconds.

Safety Mechanism:

motorStop(): Immediately stop all motors.

destroy(): Release the PCA9685 resources.

Press **Ctrl+C** to trigger a safe exit.