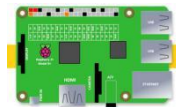
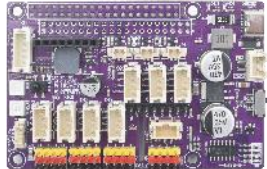



## Lesson 10 How to Control DC Motor

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

### 10.1 Components used in this course

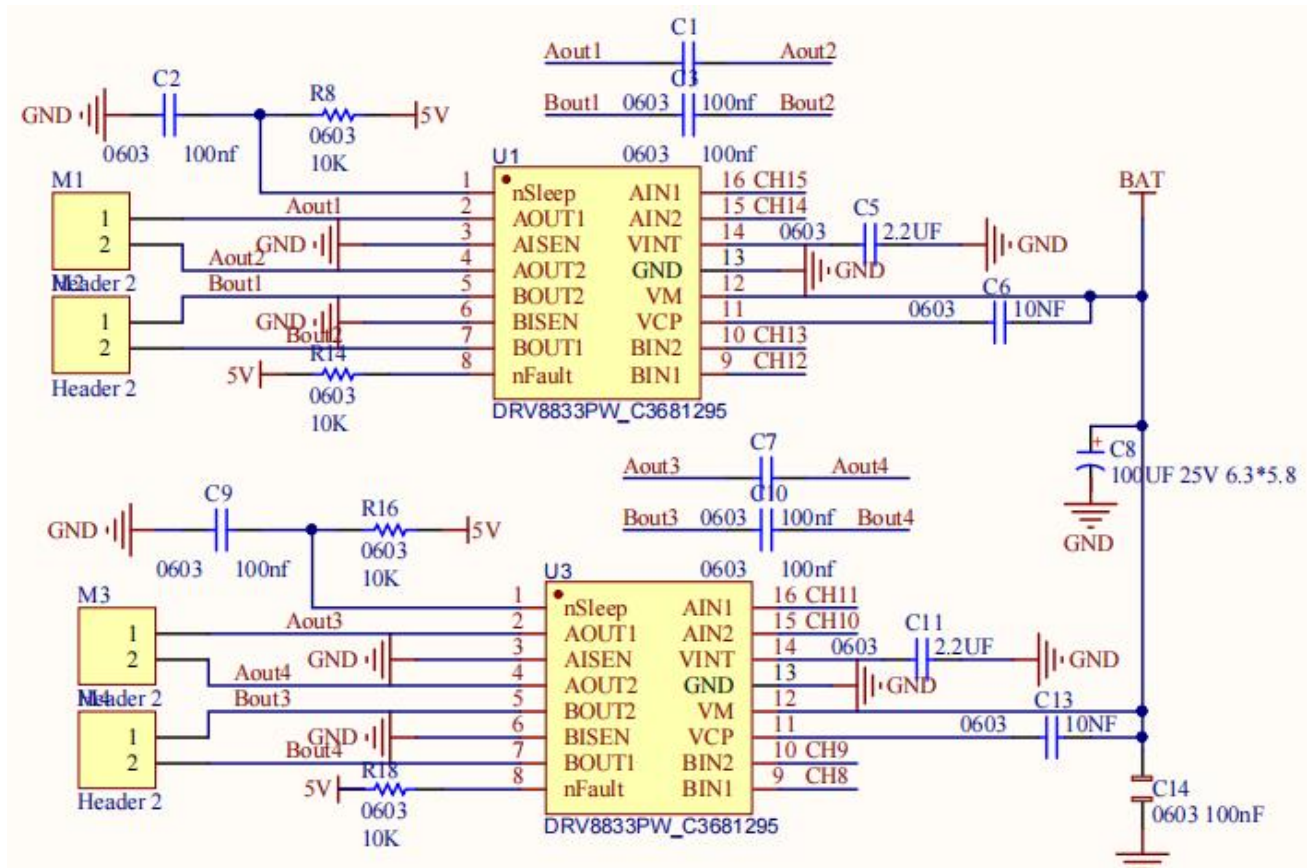
Components	Quantity	Picture
Raspberry Pi	1	
Adeept Robot HAT V3.2	1	
DC Motor	1	

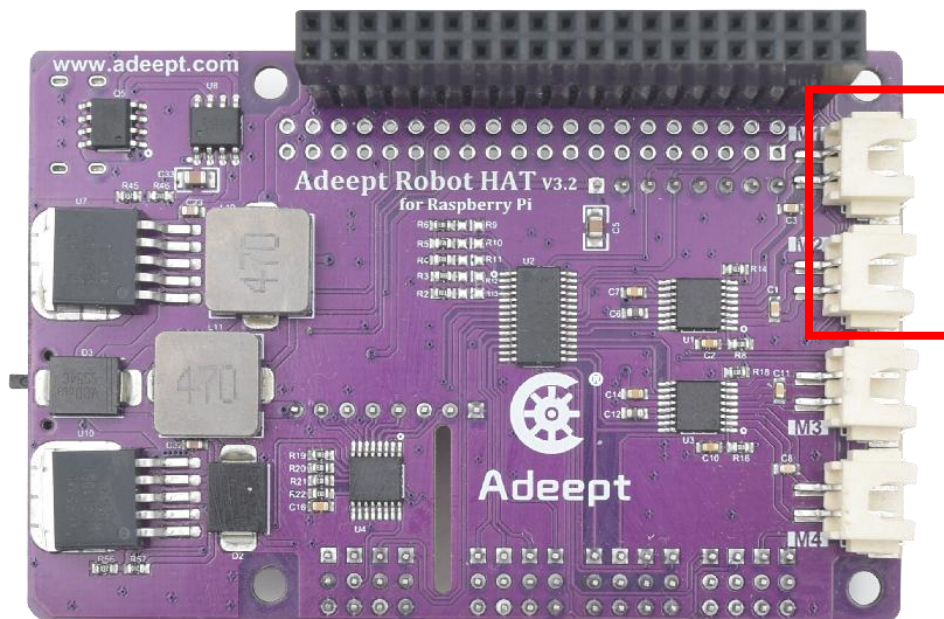
### 10.2 Introduction of DC Motor

RaspTank Metal 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.

## 10.3 Wiring Diagram (Circuit Diagram)

When the DC Motor module is in use, it needs to be connected to the M1/M2 interface on the Adept Robot HAT V3.2 expansion board.





## 10.4 How to Control DC 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. See the occupied I2C address.

**i2cdetect -y 1**

```
pi@raspberrypi:~ $
pi@raspberrypi:~ $ i2cdetect -y 1
    0  1  2  3  4  5  6  7  8  9  a  b  c  d  e  f
00:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
10:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
20:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
30:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
40:  --  --  --  --  --  --  --  48  --  --  --  --  --  --  --  --
50:  --  --  --  --  --  --  --  --  --  --  --  --  --  5f  --  --
60:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
70:  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --  --
pi@raspberrypi:~ $
```

3. Enter the command and press Enter to enter the folder where the program is located:

**cd Adeept\_RaspTank\_Metal/examples/**

```
pi@raspberrypi:~ $ cd Adeept_RaspTank_Metal/examples/
pi@raspberrypi:~/Adeept_RaspTank_Metal/examples $
```

4. View the contents of the current directory file:

**ls**

```
pi@raspberrypi:~/Adeept_RaspTank_Metal/examples $ ls
01_LED.py 02_Buzzer.py 03_Servo.py 04_Motor.py 05_WS2812.py 06_Ultrasonic.py 07_TrackingLine.py
```

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

**sudo python3 04\_Motor.py**

```
pi@raspberrypi:~/Adeept_RaspTank_Metal/examples $ sudo python3 04_Motor.py
Forward
Backward
```

6. After running the program successfully, You will observe that the motor will repeat the forward and reverse rotation ten times.

## 10.5 Code

```
1. #!/usr/bin/env/python3
2. import time
3. from board import SCL, SDA
4. import busio
5. #from adafruit_motor import servo,motor
```

```
6. #from adafruit_motor import motor
7. from adafruit_pca9685 import PCA9685
8. from adafruit_motor import motor

9. # motor_EN_A: Pin7 | motor_EN_B: Pin11
10. # motor_A: Pin8,Pin10 | motor_B: Pin13,Pin12

11. MOTOR_M1_IN1 = 15      #Define the positive pole of M1
12. MOTOR_M1_IN2 = 14      #Define the negative pole of M1
13. MOTOR_M2_IN1 = 12      #Define the positive pole of M2
14. MOTOR_M2_IN2 = 13      #Define the negative pole of M2
15. MOTOR_M3_IN1 = 11      #Define the positive pole of M3
16. MOTOR_M3_IN2 = 10      #Define the negative pole of M3
17. MOTOR_M4_IN1 = 8       #Define the positive pole of M4
18. MOTOR_M4_IN2 = 9       #Define the negative pole of M4

19. Dir_forward = 0
20. Dir_backward = 1

21. left_forward = 1
22. left_backward = 0

23. right_forward = 0
24. right_backward= 1

25. pwn_A = 0
26. pwm_B = 0
27.
28. def map(x,in_min,in_max,out_min,out_max):
29.     return (x - in_min)/(in_max - in_min) *(out_max - out_min) +out_min

30. #def setup():
31.     i2c = busio.I2C(SCL, SDA)
32.     #i2c = busio.I2C()
33.     # Create a simple PCA9685 class instance.
34.     # pwm_motor.channels[7].duty_cycle = 0xFFFF
35.     pwm_motor = PCA9685(i2c, address=0x5f) #default 0x40
36.     pwm_motor.frequency = 50
37.     # time.sleep(0.1)
38.     # pwm_motor.channels[7].duty_cycle = 0xFFFF

39. # motor11 = motor.DCMotor(pca.channels[5], pca.channels[6])
```

```
40. motor1 = motor.DCMotor(pwm_motor.channels[MOTOR_M1_IN1],pwm_motor.channels[MOTOR_M1_IN2] )
41. motor1.decay_mode = (motor.SLOW_DECAY)
42. motor2 = motor.DCMotor(pwm_motor.channels[MOTOR_M2_IN1],pwm_motor.channels[MOTOR_M2_IN2] )
43. motor2.decay_mode = (motor.SLOW_DECAY)
44. motor3 = motor.DCMotor(pwm_motor.channels[MOTOR_M3_IN1],pwm_motor.channels[MOTOR_M3_IN2] )
45. motor3.decay_mode = (motor.SLOW_DECAY)
46. motor4 = motor.DCMotor(pwm_motor.channels[MOTOR_M4_IN1],pwm_motor.channels[MOTOR_M4_IN2] )
47. motor4.decay_mode = (motor.SLOW_DECAY)
48. # motorStop()

49. def Motor(channel,direction,motor_speed):
50. if motor_speed > 100:
51. motor_speed = 100
52. elif motor_speed < 0:
53. motor_speed = 0
54. speed = map(motor_speed, 0, 100, 0, 1.0)
55. if direction == -1:
56. speed = -speed

57. if channel == 1:
58. motor1.throttle = speed
59. # print("1111")
60. elif channel == 2:
61. motor2.throttle = speed
62. # print("2222")
63. elif channel == 3:
64. motor3.throttle = speed
65. # print("3333")
66. elif channel == 4:
67. motor4.throttle = speed
68. # print("4444")

69. def motorStop():#Motor stops
70. motor1.throttle = 0
71. motor2.throttle = 0
72. motor3.throttle = 0
73. motor4.throttle = 0

74. def destroy():
75. motorStop()
76. pwm_motor.deinit()
```

```
77. if __name__ == '__main__':  
78. try:  
79.  
80. for i in range(10):  
81. speed_set = 50  
82. Motor(1, -1, speed_set)  
83. Motor(2, -1, speed_set)  
84. # Motor(3, -1, speed_set)  
85. # Motor(4, -1, speed_set)  
  
86. print("Forward")  
87. time.sleep(2)  
88. Motor(1, 1, speed_set)  
89. Motor(2, 1, speed_set)  
90. # Motor(3, 1, speed_set)  
91. # Motor(4, 1, speed_set)  
92. print("Backward")  
93. time.sleep(2)  
94. destroy()  
95. except KeyboardInterrupt:  
96. destroy()
```