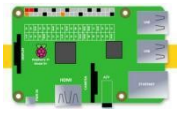
Lesson 16 How to measure the battery level

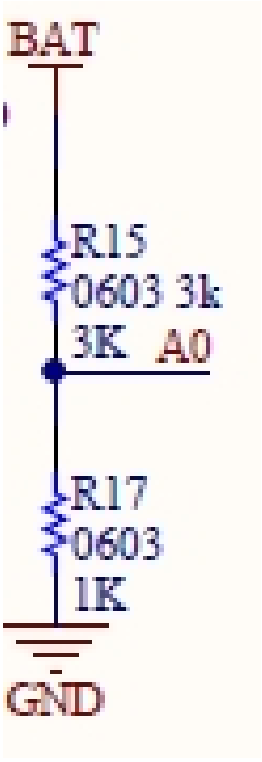
16.1 Components used in this Lesson

Components	Quantity	Picture
Raspberry Pi	1	
Adeept Robot HAT v3.1	1	

16.2 Changing Auto-run Program

The Raspberry Pi expansion board provides additional features and interfaces for the Raspberry Pi. In some projects, external power supply may be required, and understanding how to accurately measure battery voltage is crucial for ensuring stable system operation and effective battery management. This document will introduce how to use simple tools and methods to measure the battery voltage connected to a Raspberry Pi expansion board.

16.2 Wiring Diagram (Circuit Diagram)



16.3 How to measure electricity level

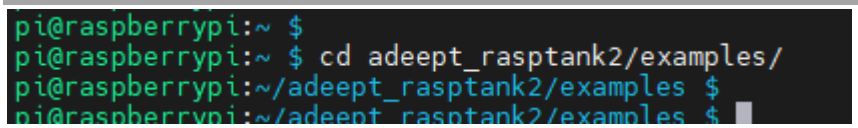
Run the code

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:~ $
```

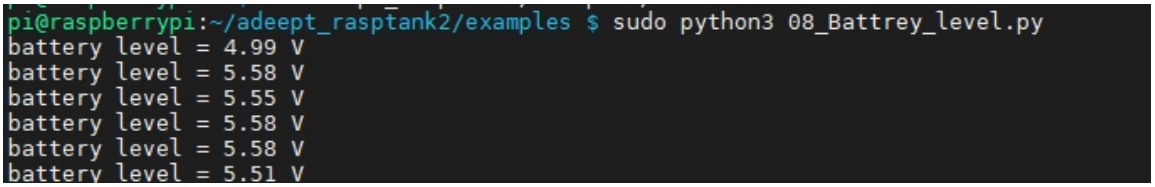
2. Enter the command and press Enter to enter the folder where the program is located:

```
cd adeept_rasptank2/examples/
```



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

```
sudo python3 08_Battrey_level.py
```



6. After successfully running the program, you will print the battery voltage on the terminal.

16.5 The main code program

For the complete code, please refer to the file 08_Battrey_level.py.

```
#!/usr/bin/env/python3
import time
import adafruit_ads7830.ads7830 as ADC
import busio
from adafruit_ads7830.analog_in import AnalogIn

i2c = board.I2C()

# Initialize ADS7830
adc = ADC.ADS7830(i2c,0x48)  #default is 0x48
chan1 = AnalogIn(adc, 1)
chan2 = AnalogIn(adc, 2)
chan3 = AnalogIn(adc, 3)
chan4 = AnalogIn(adc, 4)
chan5 = AnalogIn(adc, 5)
```

```
chan6 = AnalogIn(adc, 6)
Chan7 = AnalogIn(adc, 7)
chan0 = AnalogIn(adc, 0)
# motor_EN_A: Pin7  |  motor_EN_B: Pin11
# motor_A:  Pin8,Pin10    |  motor_B: Pin13,Pin12
if __name__ == "__main__":
    while True:
        print(f"battery level = {chan0.value/65535*8.4:.2f} V")
        time.sleep(0.5)
```