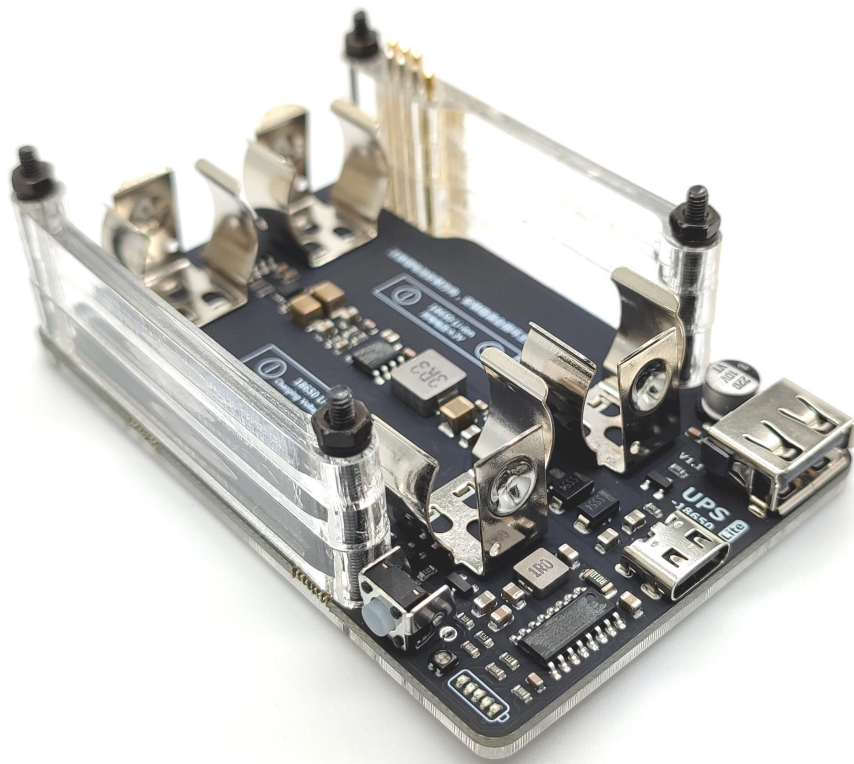# Instructions for UPS-18650-Lite

for Raspberry Pi 4B/3B+/3B   -- by XiaoJ

1. Introduction

UPS-18650-Lite is a UPS power supply specially designed for Raspberry Pi model B series boards (namely 4B/3B+/3B, etc., hereinafter referred to as pi). It uses two standard flat-head 18650 lithium batteries for power supply. It support over-charge, over-discharge, over-current, short-circuit protection, support external power plug-in detection, support charging and discharging, when the external power is plugged in, the pi is powered by the external power source, when the external power source is unplugged, the pi seamlessly switches to the lithium battery power supply . UPS-18650-Lite is connected to the pi through 7 pogopins. The power supply and battery power measurement functions of the pi are all done through pogopins. The pi does not need to plug any power cords, just plug the external power cord into the UPS type-C port ,it can complete power supply and charging by the UPS. In addition, UPS-18650-Lite integrates the professional fuel gauge chip CW2015, which can be used to measure UPS battery voltage and battery remaining power.
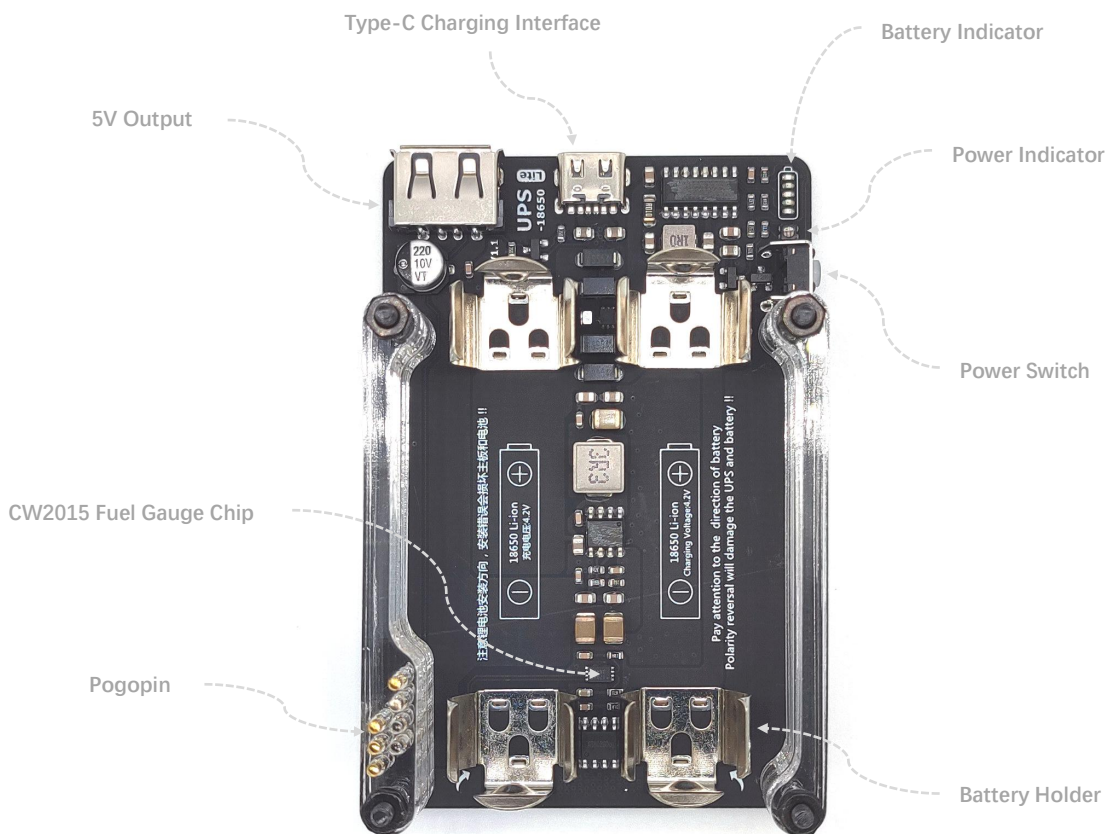
Parameter :

Charging current: Max **1A@5V**

Output current: Instant Discharge **3.5A@5.1V**, Continuous Discharge **2.5A@5.1V**

(PS: Please use good quality batteries and power adapters to ensure that the UPS works normally)
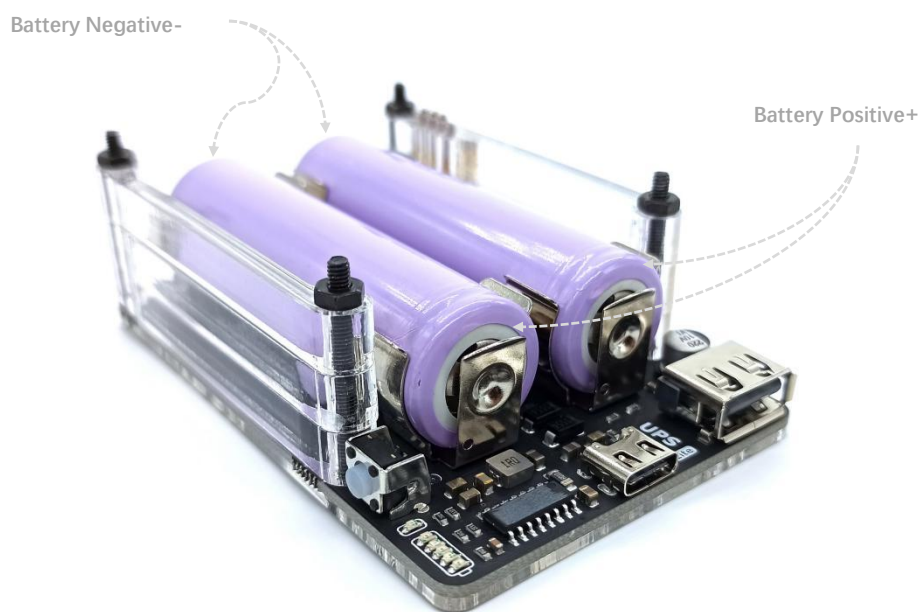
Battery measurement: Percentage of battery SOC, an error ±2%,

The measurement errors of battery voltages ± 3mV

(PS: The measurement error varies with the individual battery and the ambient temperature)

2. Install

a. Please install two 18650 lithium batteries in the direction of the battery logo on the board. Pay attention to the battery direction when installing. Incorrect installation will burn the board and the battery! ! ! (PS: The two lithium batteries must be standard flat-head 18650 batteries of the same type with similar internal resistance and capacity. Two lithium batteries with different parameters cannot be mixed, and lithium batteries with protective plates cannot be used)

Battery Negative-

Battery Positive+

PS: How to identify battery positive and negative poles

Battery Positive+

The side with the concave ring is the positive pole of the battery, and the other side is the negative pole

b. Align the four fixing holes of the pi with the four screws of the UPS, and then align the pin header of the pi with the pogopin, then lock the nuts!

Make sure the pin header is aligned with the pogopin

M2.5 *4
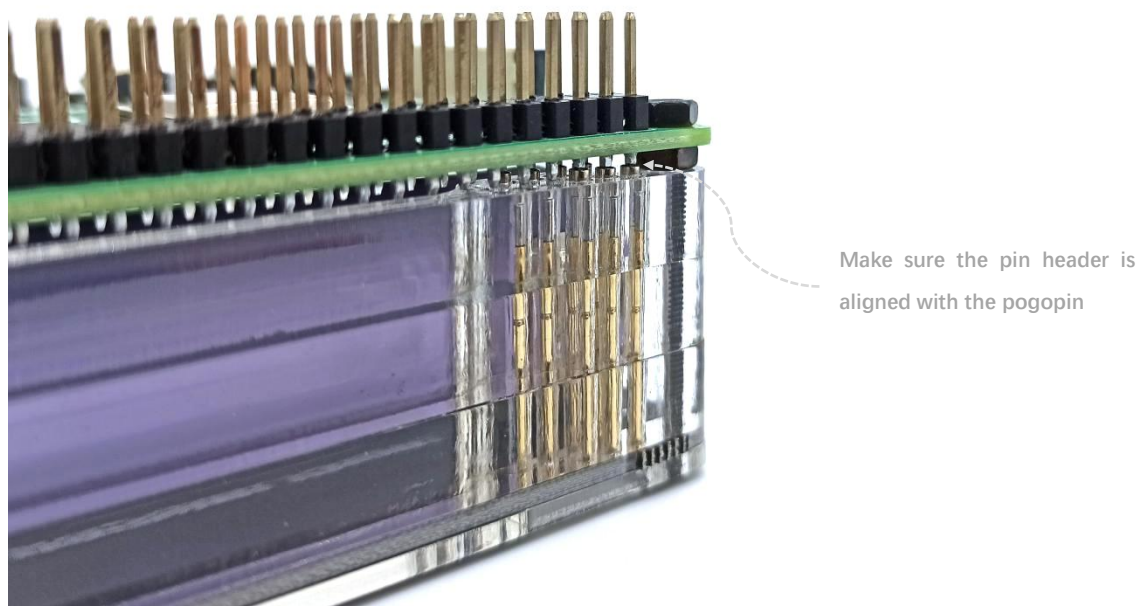
c. To remove and replace the battery, please use tweezers to insert it in the direction indicated by the arrow on the board and then pry up the battery


Insert here, pry up the battery
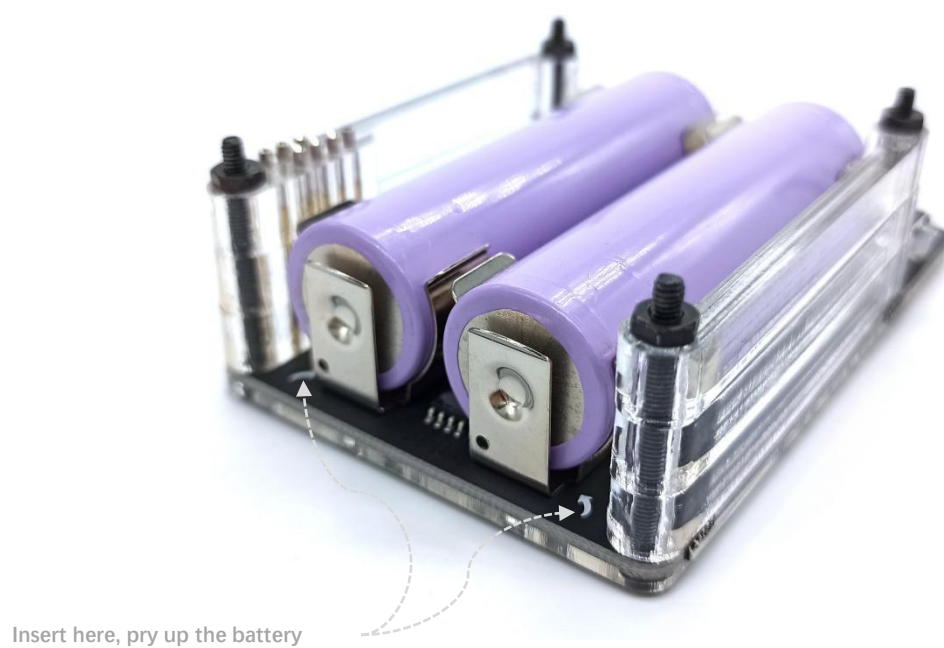
3.Function and using

3.1 Charging and Power Insertion Detection

It is recommended to use a power adapter with a power of 5V3A and above to charge the UPS-18650-Lite. Because when the lithium battery is low, the external power adapter not only needs to supply power to the pi, but also needs to provide part of the current to charge the lithium battery. During the charging , the battery indicator will flash one after another. When the lithium battery is fully charged, the 5 green battery indicators will all light up, indicating that the battery is fully charged. The battery capacity indicated by the 5 battery indicators is 20%, 40%, 60%, 80%, and 100%.

UPS-18650-Lite has an external power adapter insertion detection function. When the external power supply is plugged in, the blue indicator light is on. You can judge whether the external power supply is inserted by the high and low levels of the GPIO port. When the power supply is plugged in, the io4 (BCM mode) of the pi will detect a low level. When unplugging ,the pi will detect a high level, to enable the detection function, the two pads on the back of the UPS must be shorted, and the GPIO is set to the input state. If the pad is not shorted, the state of the GPIO detection is unstable. See the following figure for details.

Type-C Charging Interface

The two short pads
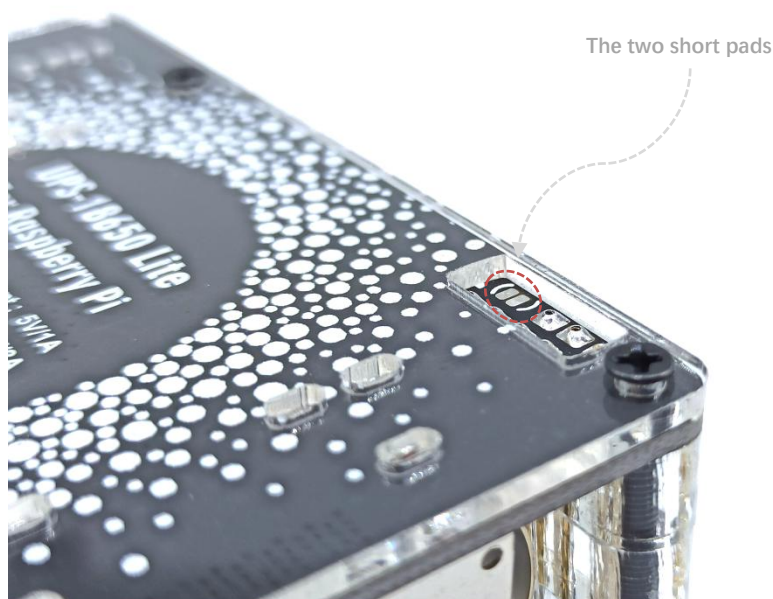
3.2 Power Output Operation

When the external power adapter is plug in, press and hold the power switch for more than 3 seconds, the red power indicator light will be on, indicatingthat the 5V voltage output is turned on. Press and hold again for more than 3 seconds, the red power indicator light is off, indicating that the 5V voltage output is turned off.

When the external power adapter is unplug, press and hold the power switch for more than 3 seconds, the red power indicator light will be on, indicatingthat the 5V voltage output is turned on. Press and hold again for more than 3 seconds, the red power indicator light is off, indicating that the 5V voltage output is turned off. You can also turn off the 5V voltage output by quick double click the power switch. Short click the power switch once, the power indicator shows the battery power, and it will go out later.

The output power of the booster circuit of UPS-18650-Lite is affected by factors such as battery voltage, battery discharge current, external power supply power, and power cord voltage drop. Therefore, it is recommended to use a high-quality 18650 battery with a large discharge current for power supply. In addition, in order to improve the circuit boost efficiency, it is recommended that the power line choose a line with a small line voltage drop, so as to minimize the power consumption caused by the line voltage drop.

The continuous output current of UPS-18650-Lite is 2.5A, and the instantaneous maximum output current is 3.5A. In the case of continuous high current output for a long time, please take heat dissipation measures and use it in a ventilated place as much as possible to avoid high temperature causing thermal shutdown protection and affect battery life.
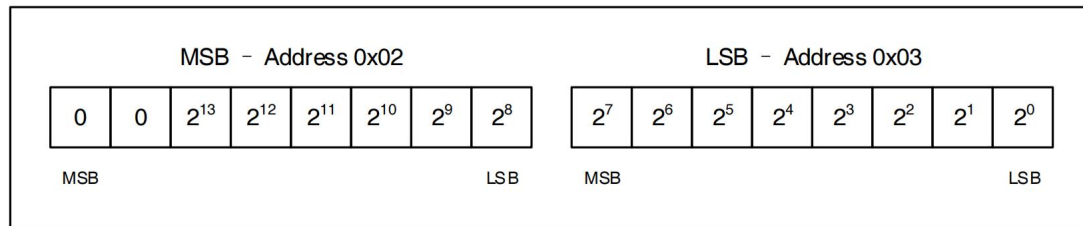

Power Switch
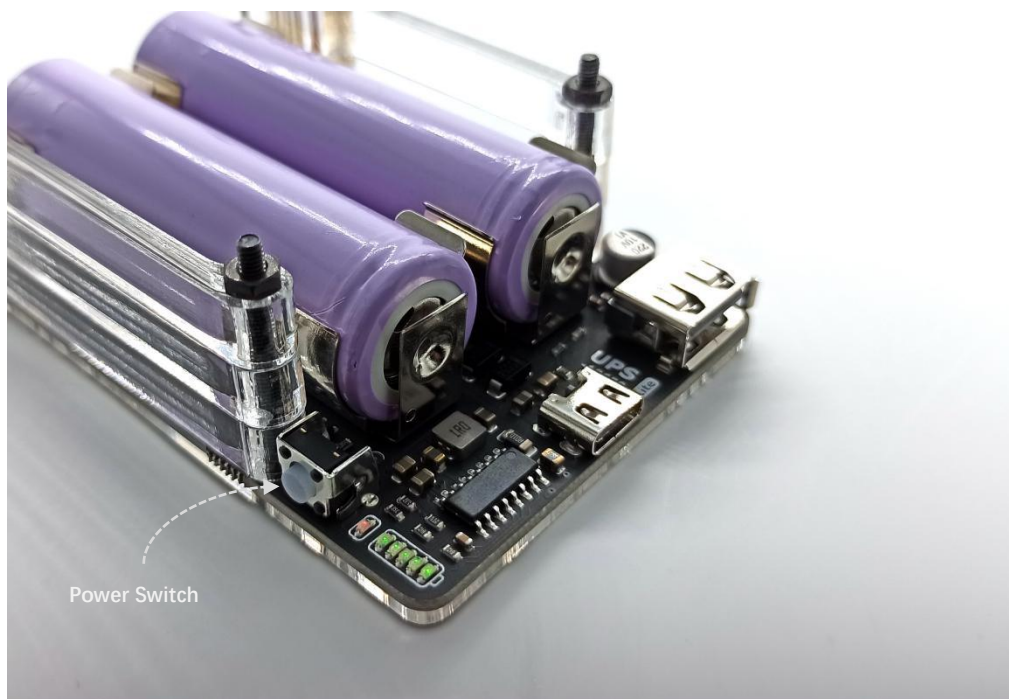
3.3 Battery Power Measurement

Create a script program named UPS_18650_Lite_V1.2_CW2015.py with the nano editor. The code is as follows (see appendix for detailed code). This script uses Python's smbus library to perform i2c read operations on the CW2015. The CW2015 device address is 0x62, the register VCELL is the 14bit battery voltage ADC measurement value, the address is 02h-03h, and the ADC measurement accuracy unit is 305uV. The register SOC is a 16-bit battery capacity percentage reading, The address is 04h-05h. The upper 8-bit unit of the SOC is 1% of the battery capacity, and the lower 8-bit unit is 1/256%. It provides the reading of the decimal point of the battery capacity percentage. Save the UPS_18650_Lite_V1.2_CW2015.py script program to a directory you know (such as the following to the /home/pi/ directory), then run the program in Python, you can see that the program will output the current battery voltage value　and the percentage of battery capacity every two seconds. In addition, due to the calculation method of the CW2015 battery capacity, when the battery capacity reading is 1%, the battery voltage reading is about 3.5V. When the voltage of the lithium battery is 3.5V, the corresponding battery capacity is already very low, and excessive discharge will damage the battery. Therefore, when the user subsequently writes a low-power automatic shutdown program, it is recommended to automatically turn off the pi when the battery capacity is 1%. When running, when the battery voltage drops to 3V, the UPS-Lite protection circuit will automatically stop supplying power. See below for specific steps
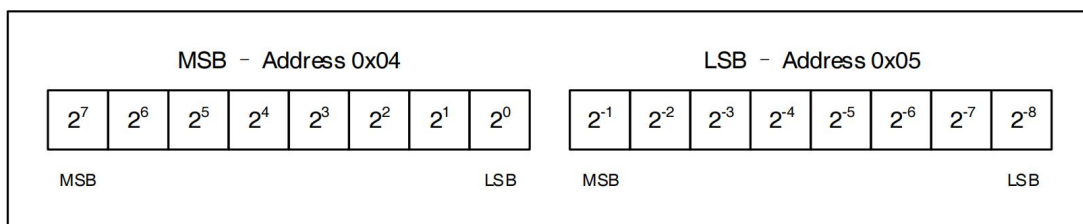
CW2015 register address and Features

REGISTER NAME	ADDRESS	DESCRIPTION	READ/WRITE	DEFAULT VALUE
VERSION	0x00	Returns IC version, software version	R	0x73
VCELL	0x02-0x03	Reports 14-bit A/D measurement of battery voltage	R	0x00
SOC	0x04-0x05	Reports 16-bit SOC result calculated	R	0x00
RRT_Alert	0x06-0x07	13 bits remaining run time and low SOC alert bit	W/R	0x00
CONFIG	0x08	Configure register, alert threshold set	W/R	0x18
MODE	0x0A	Special command for IC state	W/R	0xC0

VCELL register

MSB – Address 0x02									LSB – Address 0x03							
0	0	2^{13}	2^{12}	2^{11}	2^{10}	2^9	2^8		2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0

MSB   LSB   MSB   LSB

SOC register

MSB – Address 0x04									LSB – Address 0x05							
2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0		2^{-1}	2^{-2}	2^{-3}	2^{-4}	2^{-5}	2^{-6}	2^{-7}	2^{-8}

MSB   LSB   MSB   LSB

a. Open pi configuration tool raspi-config, enable I2C interface

The ARM I2C interface is enabled

Enable I2C interface

<Ok>
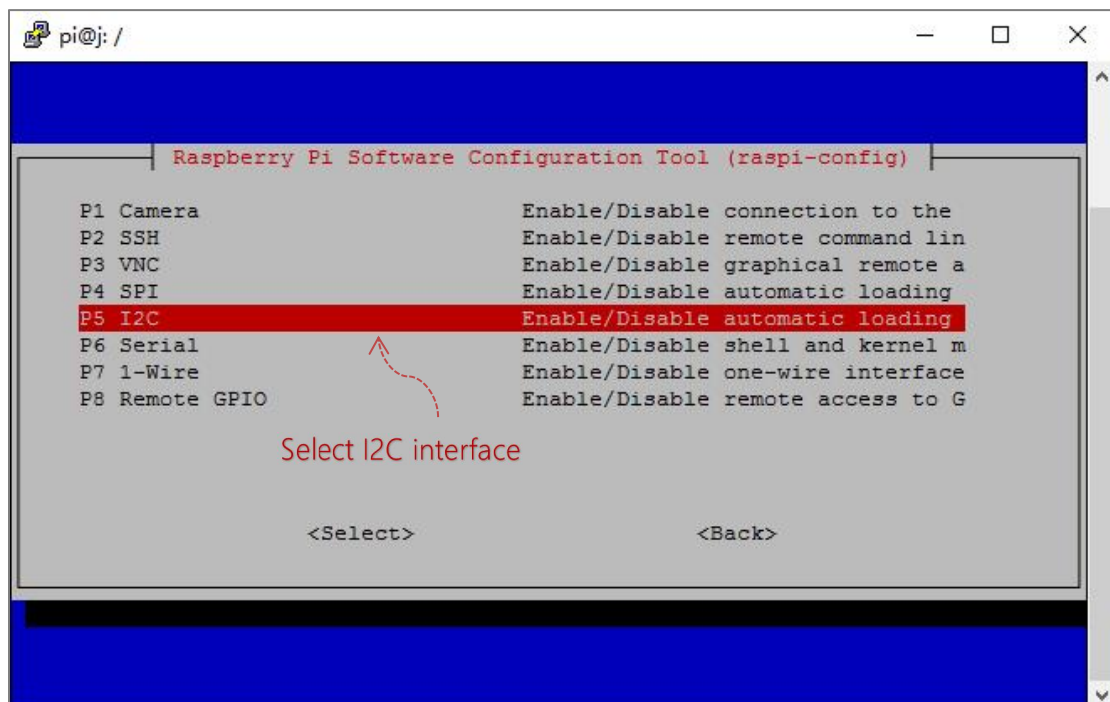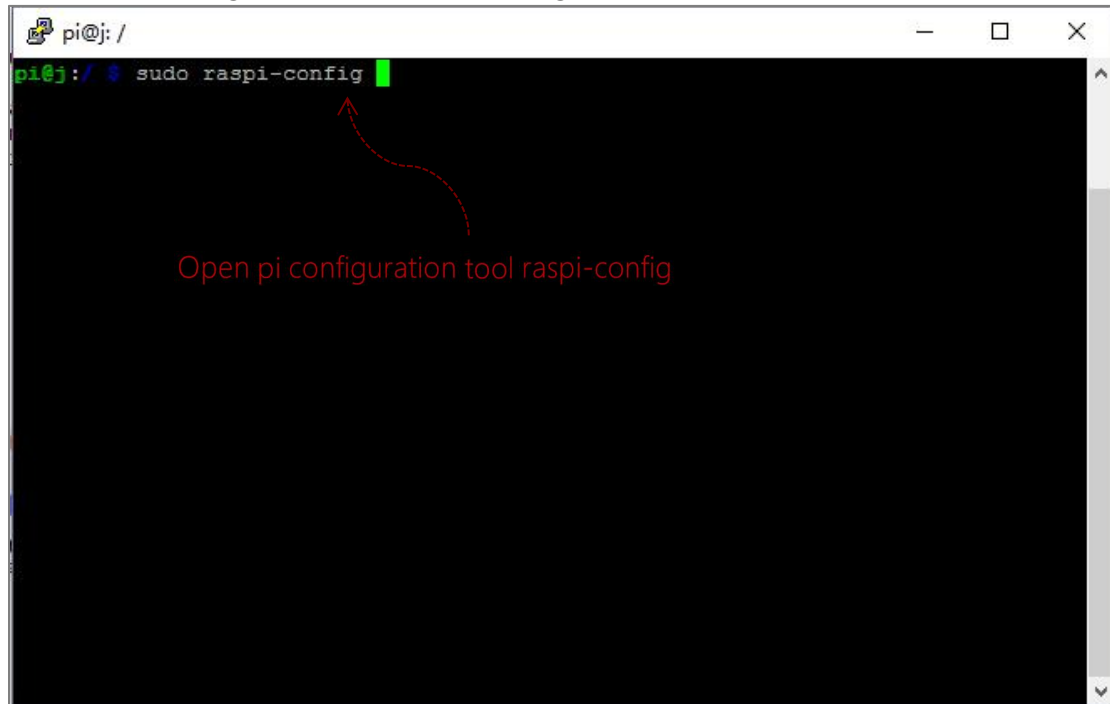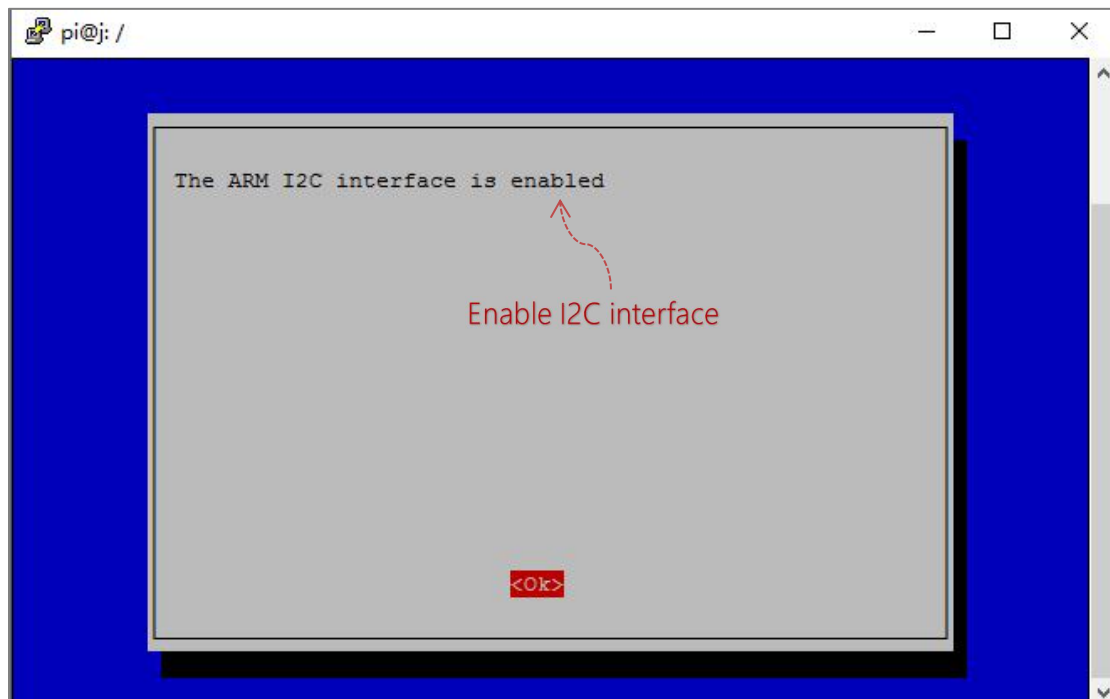
b. Install i2c-tools and python-smbus, after the installation is complete. reboot pi


```
pi@j:/ $ sudo raspi-config
pi@j:/ $ sudo apt-get install i2c-tools python-smbus
Reading package lists... Done
Building dependency tree
Reading state information... Done
i2c-tools is already the newest version.
python-smbus is already the newest version.
0 upgraded, 0 newly installed, 0 to remove and 57 not upgraded.
pi@j:/ $ sudo reboot
```

install software

Reboot pi

c. Run　sudo i2cdetect -l　to see which i2c bus the current pi is using.

d. Run　sudo i2cdetect -y 1　to view the devices mounted on the i2c bus of the current pi

e. Run the script　UPS_18650_Lite_V1.2_CW2015.py


```
pi@raspberrypi:~ $ python UPS_18650_Lite.py        ← Run the script
Initialize the MAX17040 ......
++++++++++++++++++++
Voltage: 0.00V
Battery:    0%
Battery LOW
Power Adapter Plug In                    Run the script
++++++++++++++++++++
++++++++++++++++++++
Voltage: 4.16V
Battery:   96%          ←  ───
Power Adapter Plug In
++++++++++++++++++++
++++++++++++++++++++
Voltage: 4.16V
Battery:   96%
Power Adapter Plug In        Print out the battery voltage value, battery
++++++++++++++++++++
++++++++++++++++++++
Voltage: 4.16V
Battery:   96%            capacity percentage, external power supply
Power Adapter Plug In
++++++++++++++++++++
```

appendix:

Script code of UPS_18650_Lite_V1.2_CW2015.py :

```python
#!/usr/bin/env python
import struct
import smbus
import sys
import time
import RPi.GPIO as GPIO


CW2015_ADDRESS  = 0X62
CW2015_REG_MODE = 0X0A



def readVoltage(bus):
     "This function returns as float the voltage from the Raspi UPS Hat
via the provided SMBus object"
     read = bus.read_word_data(CW2015_ADDRESS, 0X02)
     swapped = struct.unpack("<H", struct.pack(">H", read))[0]
     voltage = swapped * 0.305 /1000
     return voltage



def readCapacity(bus):
     "This function returns as a float the remaining capacity of the
battery connected to the Raspi UPS Hat via the provided SMBus object"
     read = bus.read_word_data(CW2015_ADDRESS, 0X04)
     swapped = struct.unpack("<H", struct.pack(">H", read))[0]
     capacity = swapped/256
     return capacity



def QuickStart(bus):
     "This function wake up the CW2015 and make a quick-start fuel-gauge
calculations "
     bus.write_word_data(CW2015_ADDRESS, CW2015_REG_MODE, 0x30)




#-----------------------------------------------------------------



GPIO.setmode(GPIO.BCM)
GPIO.setwarnings(False)
```

```python
GPIO.setup(4,GPIO.IN)  # GPIO4 is used to detect whether an external power
supply is inserted

bus = smbus.SMBus(1)  # 0 = /dev/i2c-0 (port I2C0), 1 = /dev/i2c-1 (port
I2C1)


QuickStart(bus)

print ("  ")
print ("Initialize the CW2015 ......")



while True:

 print ("++++++++++++++++++++")
 print ("Voltage:%5.2fV" % readVoltage(bus))
 print ("Battery:%5i%%" % readCapacity(bus))

 if readCapacity(bus) == 100:
      print ("Battery FULL")
 if readCapacity(bus) < 5:
      print ("Battery LOW")



 #The following is the power plug detection judgment program of V1.2 version.
 GPIO is low when power is plugged in
 if (GPIO.input(4) == GPIO.LOW):
      print ("Power Adapter Plug In")
 if (GPIO.input(4) == GPIO.HIGH):
      print ("Power Adapter Unplug")
 #------------------------------------


 print ("++++++++++++++++++++")
 time.sleep(2)
```

If you have other questions, please contact me: 416386001@qq.com