

RK818/RK816电量计 开发指南

发布版本：2.0

作者邮箱：chenjh@rock-chips.com

日期：2018.05

文档密级：公开资料

前言

概述

本文档主要介绍Rockchip的RK818/RK816子模块：电量计。介绍相关概念功能、DTS配置和一些常见问题的分析定位。

产品版本

芯片名称	内核版本
RK818	3.10、4.4
RK816	3.10

读者对象

本文档（本指南）主要适用于以下工程师：技术支持工程师 软件开发工程师

修订记录

日期	版本	作者	修改说明
2016.07.25	V1.0	陈健 洪	初稿，rk818在内核4.4的使用
2017.05.25	V2.0	陈健 洪	增加rk816、rk818在内核3.10的使用；修正充电电压/电流的说明错误
2018.05.31	V2.1	陈健 洪	调整文档格式和排版错误、无主要内容更新

RK818/RK816电量计 开发指南

- 1 概述RK818/RK816电量计
- 2 电量计原理
- 3 重要概念
- 4 驱动和menuconfig
 - 4.1 电量计驱动的功能
 - 4.2 内核4.4
 - 4.3 内核3.10
- 5 DTS配置
 - 5.1 内核4.4

5.2 内核 3.10
6 开始电量计相关的开发
6.1 准备
6.2 问题处理
7 电池校正
7.1 电池校正的原理
7.2 电池校正的方式
7.3 何时需要校正
8 常见问题分析定位

1 概述RK818/RK816电量计

RK818/RK816 是一款高性能 PMIC，集成了多路大电流DCDC，多个LDO，1个线性开关，1个USB 5V及 boost输出，还有开关充电，智能功率路径管理，库仑计，RTC 及可调上电时序等功能。其中“开关充电、智能功率路径管理、电量计（库仑计）”是本文档所要涉及的功能。

1. 充电管理：包括输入限流，涓流充电，恒流/恒压充电，充电终止，充电超时安全保护等功能。
2. 智能功率路径管理：可对输出电压进行调节以向系统负载提供所需要的功率，同时可以对电池进行充电。当进入输入限流状态时，输入功率会优先提供给系统负载，而剩余的功率才会提供给电池充电用。另外，在系统负载所需功率超过限定的输入功率，或者电源输入被断开时，智能功率路径管理功能会自动开启电池与系统负载间的开关，从而使电池可以同时向系统负载提供额外功率。
3. 电量计（库仑计）：通过采用自有专利技术的算法，该电量计可以根据不同电池的充放电特性曲线，精确地测量电池电量，并把电池电量信息通过I2C接口提供给系统主芯片。同时有对过度放电电池的小电流充电，电池温度检测，充电安全定时器，和芯片热保护等功能。

2 电量计原理

1. 三个基本原则：
 - a) 电池的开路电压与电池电量的百分比（OCV - SOC）曲线主要取决于电池制作的材料和工艺，并受温度、老化等的影响很小，即电池生产出来后SOC - OCV 的曲线基本不变。
 - b) 电池在工作期间受到电池极化等影响，难以从电池端口电压推算出电池的 OCV 电压值，因此只有在电池未极化时（如长时间关机后或长时间小电流工作时）才能得到可用的 OCV电压，并通过 OCV 推算得到 SOC。
 - c) 库仑计能测量实际流入或者流出电池的电量，若已知电池总容量，就能很容易的得到SOC 值，但库仑计的累计误差大，而且电池的总容量会受温度和老化等因素等影响，因此库仑计的方法只能保证在短时间内具有较好的精度，并且需要定期更新电池总容量。
 - d) 目前性能良好的电量计，都是建立在以上 OCV 估算与库仑计计算基础上得到实时的电量计剩余容量状态。
2. 库仑计法

在电池的正极和负极串接一个电流检查电阻，当有电流流经电阻时就会产生 V_{Δ} ，通过检测 V_{Δ} 就可以计算出流过电池的电流。因此可以精确的跟踪电池的电量变化，精度可以达到1%，另外通过配合电池电压和温度，就可以极大的减少电池老化等因素对测量结果的影响。

3. 电流信号调理

采用20或10毫欧电阻取样电流，通过恒流源提供一个偏置，将负电流信号提高到正值，再经运放放大到参考电压Vref范围，并通过ADC模块转换为数字量。

4. 电压采集

通过分压电路将电池电压分压(分压比0.5)到Vref范围内，在经ADC模块转换成数字输出。

5. 平均电流采集

数字部分对修正过的电流值与多位数的累加器每秒进行256次的叠加。一秒结束后将累加器的值除以256得到平均电流的值。

6. 库仑计更新

库仑计是在平均电流更新的时候，自动累加一次。

3 重要概念

- **ocv电压** 开路电压。PMIC在上电时序过程中采集，因为此时的负载还非常非常小，近似于开路状态，所以此时的电压是准确的。用途：当满足关机至少30分钟时，我们认为电池的极化基本消除，此时获取的OCV电压真实有效，因此会用该电压去查询ocv_table得到一个新的电量去更新库仑计的值，进行一次库仑计的校正。
- **ocv_table** 每款电池都有自己的电池特性曲线，根据ocv特定电压对应特定电量的原则，我们将0%~100%的电量细分成21个点，步进5%电量，由此得到一张/组“电压<-->电量”的表格。这个表的用途就是第一次接电池开机、长时间关机后再开机、长时间休眠后校正库仑计的依据所在。例如：

```
1 ocv_table = <3400 3599 3671 3701 3728 3746 3762 .....  
  4088 4132 4183>;
```

对应关系：3400mv: 0%、3599mv: 5%、3671mv: 10%、.... 4183mv: 100%;

- **最大输入电流和最大充电电流** 软件上配置可从适配器获取的最大电流，称之为“最大输入电流”。例如5V/2A的适配器，我们一般软件配置最大输入电流为2A（也可以设置为1.8A...）。RK818/816有智能电源路径管理功能，即来自适配器的电优先供应给系统使用，剩余的再给电池充电，软件上配置的允许向电池充电的最大剩余电流值称之为“最大充电电流”。
- **发生输入限流** 可以简单理解为，当给电池和系统的供电电流需求超过最大输入电流时，发生这种“不够用”的情况时就称之为“发生了输入限流”。或者还可以理解为：当无法在电池所需条件下以最大充电电流状态给电池充电时，即发生了输入限流。该功能主要是用来作为充电截止的三个条件之一（另外两个为充电截止电压和截止电流）。
- **松弛模式** 在极低负载情况下（目前只针对于二级待机），如果系统的负载电流持续超过一定时间（软件可配）都小于某个阈值，则电量计模块进入松弛模式。

- **松弛电压** 在松弛模式下电量计每隔8分钟会采集一组电压，我们称之为松弛电压。用途：二级待机的负载很小，我们近似地认为松弛电压近似于开路电压，因此驱动处理上，在系统从二级待机唤醒且满足一定条件时会用它查询ocv_table表进行库仑计的校正。
- **finish充电截止信号** 当发生电池充电截止时，寄存器会产生一个状态信号，称之为finish的信号。用途：软件只有获取到该信号才认为是真正的、硬件上的充电截止，然后会进入相对应的算法流程调整显示的电量。
- **芯片热保护** PMIC的自我保护机制，实际是一个反馈机制：当芯片温度大于设置的阈值时，会发生输入电流被逐渐减小的过程，以此降低PMIC工作负荷，降低芯片热量。这个反馈过程由硬件自动完成，软件无法参与，也没有严格的温度和电流大小的对应比值，极端恶劣条件下，甚至停止充电。
- **充电截止条件** 当充电电流达到截止电流，电压达到截止电压，且没有发生输入限流的情况下产生充电截止信号，不再继续充电，我们认为此时电池已经充满。（这里需要输入限流的条件来判读，否则无法分清是真的充电电流变小了，还是因为此时的系统负载比较大导致给电池的充电电流小）

4 驱动和menuconfig

4.1 电量计驱动的功能

1. 电池电量的统计和显示；
2. 充电电流、电压的设置（根据电池、充电器类型），支持单口/双口充电；
3. OTG设备的5V供电；
4. 电池温度侦测。

4.2 内核4.4

rk818 驱动和宏配置（功能太过庞大，因此拆分成2个驱动）：

```
1 drivers/power/rk818_battery.c           // 负责处理电量显示
2 drivers/power/rk818_charger.c           // 负责处理充电器检测、充电电压、电流设置
```

```
1 CONFIG_BATTERY_RK818
2 CONFIG_CHARGER_RK818
```

rk816 驱动和宏配置：

```
1 drivers/power/rk816_battery.c           // 负责处理电量显示 + 充电器检测、充电电压、电流设置
```

```
1 CONFIG_BATTERY_RK816
```

4.3 内核3.10

rk818 驱动和宏配置:

```
1 drivers/power/rk818_battery.c // 负责处理电量显示 + 充电器检测、充电电压、电流设置
```

```
1 CONFIG_BATTERY_RK818
```

rk816 驱动和宏配置:

```
1 drivers/power/rk816_battery.c // 负责处理电量显示 + 充电器检测、充电电压、电流设置
```

```
1 CONFIG_BATTERY_RK816
```

5 DTS配置

5.1 内核4.4

DTS的配置包括两个部分:

1. battery部分: 必选。

一个完整的battery节点信息如下所示, 该节点放在RK818节点之内, RK818的battery和charger子设备驱动都会用到battery节点内的信息。其中ntc_table、ntc_degree_from、dc_det_gpio是可选部分, 其余是必选部分。

```
1 battery {
2     compatible = "rk818-battery"
3     ocv_table = <3400 3599 3671 3701 3728 3746 3762
4                 3772 3781 3792 38163836 3866 3910
5                 3942 39714002 4050 4088 4132 4183>;
6     ntc_table = <43662 41676 39793 38005 36308 34696
7                 33164
8                 31709 30326 29011 27760 26570 25438
9                 24361
10                23335 22358 21427 20540 19695 18890
11                18121
12                17389 16690 16022 14778 14197 13642
13                13113
14                12606 12122 11659 11216 10793 10388
15                10000
16                9629 9273 8933 8607 8295>;
17     ntc_degree_from = <1 10>;
18     design_capacity = <4000>;
19     design_qmax = <4100>;
20     bat_res = <120>;
21     max_input_current = <2000>;
22 }
```

```

17     max_chrg_current = <1800>;
18     max_chrg_voltage = <4200>;
19     sleep_enter_current = <300>;
20     sleep_exit_current = <300>;
21     power_off_thresd = <3400>;
22     zero_algorithm_vol = <3850>;
23     energy_mode = <0>;
24     fb_temperature = <105>;
25     sample_res = <10>;
26     max_soc_offset = <60>;
27     monitor_sec = <5>;
28     virtual_power = <0>;
29     power_dc2otg = <1>;
30     dc_det_gpio = <&gpio0 GPIO_C1 GPIO_ACTIVE_LOW>;
31 };

```

2. charger部分：可选。

如果不支持typec口充电，无需这部分的配置。对于支持typec充电口的机器，请在rk818的根节点下面加入引用“extcon= <&fusbn>”节点(其中，n=0,1..，具体引用请参考实际硬件情况)，如下图。因为rk818_charger.c需要根据该引用去注册typec的通知链，以此获取typec的charger类型检测信息。

```

1  rk818: pmic@1c {
2      compatible= "rockchip,rk818";
3      status= "okay";
4      reg= <0x1c>;
5      clock-output-names= "xin32k", "wifibt_32kin";
6      interrupt-parent= <&gpio1>;
7      interrupts = <21 IRQ_TYPE_LEVEL_LOW>;
8      pinctrl-names= "default";
9      pinctrl-0= <&pmic_int_1>;
10     rockchip,system-power-controller;
11     rk818,support_dc_chg= <1>;/*1: dc chg; 0:usb
    chg*/
12     wakeup-source;
13     extcon = <&fusbn>;      // 重要!!!
14     #clock-cells= <1>;
15
16     battery {
17         .....
18         .....
19     };
20 };

```

5.2 内核 3.10

RK816和RK818的节点信息基本一致，只有个别属性存在区别，如下以RK818为例进行说明。

一个完整的battery节点信息如下所示，该节点放在RK818节点之内。其中ntc_table、ntc_degree_from、dc_det_gpio、dc_det_adc是可选部分，其余是必选部分。

```
1 battery {
2     compatible = "rk818-battery" // 如果是rk816, 则改
    成"rk816-battery"
3     ocv_table = <3400 3599 3671 3701 3728 3746 3762
4                 3772 3781 3792 38163836 3866 3910
5                 3942 39714002 4050 4088 4132 4183>;
6     ntc_table = <43662 41676 39793 38005 36308 34696
    33164
7                 31709 30326 29011 27760 26570 25438
    24361
8                 23335 22358 21427 20540 19695 18890
    18121
9                 17389 16690 16022 14778 14197 13642
    13113
10                12606 12122 11659 11216 10793 10388
    10000
11                9629 9273 8933 8607 8295>;
12     ntc_degree_from = <1 10>;
13     design_capacity = <4000>;
14     design_qmax = <4100>;
15     bat_res = <120>;
16     max_input_current = <2000>;
17     max_chrg_current = <1800>;
18     max_chrg_voltage = <4200>;
19     sleep_enter_current = <300>;
20     sleep_exit_current = <300>;
21     sleep_filter_current = <100>; // rk818不需要这个属
    性
22     power_off_thresd = <3400>;
23     zero_algorithm_vol = <3850>;
24     energy_mode = <0>;
25     fb_temperature = <105>;
26     max_soc_offset = <60>;
27     monitor_sec = <5>;
28     virtual_power = <0>;
29     power_dc2otg = <1>;
30     dc_det_gpio = <&gpio0 GPIO_C1 GPIO_ACTIVE_LOW>;
31     dc_det_adc = <1>; // rk818不需要这个属性, rk816才可
    能需要
32 };
```

参数说明:

- **ocv_table** 开路电压-电量表。即“电压对应电量”，一共21个电压值，分别对应0% -->100%, 电压值之间的电量步进为5%。该数据表可以由电池原厂提供，也可以由RK深圳分公司进行测量，或者RK提供的测量工具得到，具体请咨询深圳分公司相关工程师。

- **ntc_table** 电池ntc表，单位：欧姆。如果需要进行电池温度检测，请填写对应的ntc值。一个值代表一个温度，相邻值之间温度步进1摄氏度，从左到右依次增大。如果不需要检测电池温度，请去除该属性字段。上述示例表示-10~30摄氏度对应的ntc值。
- **ntc_degree_from** ntc_table[0]对应的温度，即起始温度值。所以在配置了ntc_table的前提下再添加该属性。因为DTS不能传递负数，所以ntc_degree_from由两个字段组成，第一个表示正负符号：1：负数，0：正数；第二个字段表示温度大小。如 ntc_degree_from = <1, 10>表示 -10摄氏度。
- **design_capacity** 实际电池容量。经实际测量后确定的实际可用容量。例如标称4000mah，但是实测只有3850mah，则该值请填写3850。
- **design_qmax** 最大容量值，主要用途是作为软件处理的纠错条件之一。目前请该值请填写标称容量的1.1倍数值：即标称容量*1.1。
- **bat_res** 电池内阻。主要在放电算法中会用到，非常重要！该值在测量ocv_table时一起获取，所以请注意这个参数的测量，切勿遗漏。
- **max_input_current**

最大输入电流。目前有如下档位（单位：mA）：

```
1 RK818: <450, 80, 850, 1000, 1250, 1500, 1750, 2000, 2250, 2500, 2750, 3000>
2 RK816: <450, 80, 850, 1000, 1250, 1500, 1750, 2000>
```

注意，第2个档位是80，不是800！使用中一般不去设置80ma的档位。

- **max_chrg_current**

最大充电电流。目前有如下档位（单位：mA）：

```
1 RK818: <1000, 1200, 1400, 1600, 1800, 2000, 2250, 2400, 2600, 2800, 3000>
2 RK816: <1000, 1200, 1400, 1600, 1800, 2000, 2250, 2400>
```

- **max_chrg_voltage**

最大充电电压，即电池满充的截止电压。目前有如下档位（单位：mV）：

```
1 RK818: <4050, 4100, 4150, 4200, 4250, 4300, 4350>
2 RK816: <4050, 4100, 4150, 4200, 4250, 4300, 4350>
```

- **sleep_enter_current**

进入松弛模式的条件之一。目前填写300，不做改动。

- **sleep_exit_current**

退出松弛模式的条件之一。目前填写300，不做改动。

- **sleep_filter_current**

过滤无效的松弛电流。目前填写100，不做改动。

- `power_off_thresd` 请仔细阅读和理解

期待的系统关机电压，单位：mV。特别注意：该值指的是**VSYS**的瞬时电压，而不是**vbat**端的电压（但是电量计采集的是**vbat**端的电压）！

原理说明：**Vbat**端的电压需要经过一个阻值大约50毫欧的**mos**管后（除此外，其实另外还有PCB走线带来的阻抗）才转换为**VSYS**供给系统，所以把**VSYS**作为关机点依据才是正确的。由此我们可知：相同的**vbat**端电压，当前的负载电流越大，则**vsys**端电压就越低；反之，相同**vsys**下，当前负载电流越大，对应的**vbat**电压也就越高。

RK的平台不建议**vsys**端的电压低于3.4v，这样容易导致VCC_IO（3.3v）等DCDC/LDO的供电不稳定。

- `zero_algorithm_vol`

进入电压+库仑计放电模式的电压值，单位：mV。低于该值，进入电压+库仑计结合的软件放电算法。建议：4.2v电池设置为3850mv，4.3v及其以上的电池设置为3950mv。

- `energy_mode`

有些客户会很关心曲线的平滑度，有些则更关心电池是否可以完全放电放完，二者难以平衡。所以预留了这个属性来选择，当该值为1时表示尽可能采取将电池电量放完的方式，为0时表示尽量考虑曲线平滑的合理性（比如设置3.4v关机，有时可能3.5v会关机），驱动上已经尽量均衡曲线平滑度和关机电压点，所以建议设置为0，如果测试后发现关机电压点实在不能满足需求，可以设置为1进行尝试或者直接联系驱动维护者进行优化。

- `fb_temperature`

芯片热保护温度阈值，目前有四个温度档位（单位：摄氏度）：

```
1 < 85, 95, 105, 115>
```

目前VR上选择115，其余都选择105。如果设置为0即关闭温控反馈功能，该值一般只用于排除问题时设置（见“1.5 常见问题分析定位”）。正常使用时绝不允许关闭温控反馈功能。

- `sample_res`

电池端附近的采样电阻大小，单位：毫欧。库仑计是通过该电阻来获知当前系统的电流大小，请根据实际硬件贴的电阻大小填写。目前电阻的大小只支持10mR或者20mR。

说明：只有4.4内核上的rk818电量计驱动才支持。3.10内核都不支持，默认20mR。

- `max_soc_offset`

开机校正时允许的最大电量误差。如果关机至少30分钟，则开机时会进行一次ocv表的电量查询，并且对比关机前的电量，如果偏差超过max_soc_offset，即进行强制校正，把电量设置为ocv表对应的真实值。例如：当前显示电量是20%，但是根据ocv电压推算的实际电量为80%，则此时显示的电量直接显示为80%。一般在发生死机后会出现这种电量偏差极大的情况，这个值的大小依客户的可接受程度，由客户自己进行设置，不建议这个值小于60。

- `monitor_sec`

轮询时间（秒）。电量计驱动是需要不停地轮询才能正常工作，期间需要进行不少I2C读写操作，但是考虑到不同平台上I2C的健壮程度不同，所以预留该配置选项。目前建议5~10s比较合适，设置为5s是最佳选择。

- `virtual_power`

测试模式。有时候在拷机过程中不希望因为电量、充电电流等原因导致系统供电不足导致系统关机。设置该值为1，即放开充电电流限制，系统输入电流始终为max_input_current来满足供电。此时驱动始终上报给android，当前为充电状态，电量66%。

- `power_dc2otg`

是否支持otg设备从dc获取5v供电。对于支持双口充电的机器，硬件电路上支持插入dc的时候，otg设备的5v供电直接由dc供给，不需要额外再由RK818提供5v输出。支持设置为1，不支持或者没有dc口的请设置0。

- `dc_det_gpio`

指定dc脚对应的gpio。如果没有该项功能，请去掉该属性。

- `dc_det_adc`

是否支持使用saradc检测dc脚，1：支持，0：不支持。一般而言，这个属性和“dc_det_gpio”肯定是二选一的情况。

6 开始电量计相关的开发

6.1 准备

1. 测量电池的ocv曲线和电池内阻：每款电池都有电压-电量特性曲线（ocv_table）和内阻，RK深圳公司可以进行这方面的测量。
2. 填写DTS参数，参考1.5章节进行正确的配置。
3. 对电池进行校正，参考1.7章节。
4. 开始正常使用。

6.2 问题处理

1. 使用过程中出现了异常，请马上打开调试信息抓取现场的log，然后分析；
2. 如果是觉得充电/放电曲线有问题，比如跳变、过快/过慢。那么请抓取完整（0100%或者100%0%，原则：尽量涵盖大的电量区间）的充电/放

电log;

3. 如果自己无法分析出原因，请抓取现场或者复现的log，提到redmine上。说明：抓取的log一定要打开电量计的debug信息！！一些常见问题可以参考第8章节。

7 电池校正

7.1 电池校正的原理

1. 机器关机后其实只是PMIC关闭了各路DCDC/LDO，但是本身没有完全下电，而是以极低的负载维持在上电状态。PMIC自身提供了一些空白的data寄存器可以用于存储电量计的数据，目前存储的信息有关机前的：电量、库仑计容量、电池满充容量（即design_capacity），每次开机进行电量计驱动初始化时，这些值并不受DTS的影响，而是读取关机前的数据，继续使用。
2. 当需要对上述3个信息做出校正时，就需要PMIC完全下电来清空这几个数据。当完全下电再上电的时候，电池的GGSTS_REG[4]（“第一次上电”）的状态位会被置位。所以就需要通过卸掉电池达到此目的。
3. 当再次重新接上电池后，驱动判断当前是“第一次上电”，则所有的相关数据都会重新从DTS获取和并计算相关的电池容量、电量等。这样，我们就得到了一次校正后的准确状态。
4. 对于第3点要注意，电池取下后，应该是确认极化基本消除后才重新接回去（可以静置电池，等待它极化消除），否则开机的ocv电压也不准确。比如：当大负载放电的时，将电池取下，这时候电池实际上还处于极化状态，电池电压会慢慢回升，如果这时候就马上再接回去，那么这个时候采集的ocv电压是不准的！

我们要保证：校正的时候电池处于极化基本消除的状态，这样第一次开机的ocv电压才准确，才能获得准确电量。

7.2 电池校正的方式

目前有两种方式可以对电池进行校正，2选1：

法一：硬件法：卸下电池10s左右，再重新接上；

法二：软件法：使用串口执行如下操作：

1. 查找bat节点路径：busybox find /sys/ -name bat，例如路径为"/sys/rk818/bat"；（如果是rk816, 路径为"/sys/rk816/bat"，下同）
2. 执行：echo m > /sys/rk818/bat;
3. 读回来确认：echo r > /sys/rk818/bat, 返回值的BIT(4)应该为1才对；
4. 然后正常关机，关机时间至少30分钟以上再开机（此时才能得到准确的ocv电压）。

补充：如果需要清除步骤2中的操作，执行：echo c > /sys/rk818/bat;

7.3 何时需要校正

1. 当DTS配置的电池容量有改变时；

2. 很明显电量已经不准（原因可能是机器死机、某些特别的非电量计压力测试等）；
3. 电量计专项拷机前校正一次，保证电池是在准确的情况下开始的测试，这样才有意义（只需要所有测试项的最开始校正一次即可，不用每个case测试前都校正）。

8 常见问题分析定位

1. 如何打开调试信息，抓取log？

法一：编译前把驱动第一行的static int dbg_enable = 0 改为1即可。

法二：如果固件没有打开dbg_enable, 运行是也可以串口输入如下命令进行开关： 打开： `echo 1 >`

`/sys/module/rk818_battery/parameters/dbg_level` 关闭： `echo 0 > /sys/module/rk818_battery/parameters/dbg_level`

如果是rk816, 则节点改

为： `/sys/module/rk816_battery/parameters/dbg_level`

2. 为什么插着适配器或者usb，关机后马上又重启，无法关机？ PMIC芯片设计之初就是定义为只要插着充电器，就不能关机。
3. 为什么开机后进行电池热拔插，寄存器GGSTS[4]（电池是否存在）指明的状态和实际的情况不一样？ PMIC不支持电池的热拔插检测，只在开机上电的时候做一次检测。
4. 为什么关机后的电压跟DTS配置的关机点不同，那么高的电压就关机了？ 关机电压以最后log打印的实时电压为准，而且这个关机电压是vsys电压（具体参考1.5.2章节）我们要保证的是实时电压不低于预设的关机点。并且关机后系统下电，锂电池极化慢慢消失，会有一个电压回升的过程，这是锂电池的特性。
5. 为什么在开关机、reboot、二级待机等拷机时，电量都不怎么变，电压却慢慢在下降？

此时驱动无法正常轮询进入工作状态，所以不支持这类拷机测试。品质部和客户在做电量计测试时没有必要测试这3项。

6. 为什么用接着电池充电器，但是充电电流一直都很小？

a) 确认是否因为充电线的质量差，阻抗大，导致实际给vbus的电压远不足5v。可以外接稳压电源供电，适当提高电压，观察是否电流能增大；

b) 进入灭屏状态（一级待机），观察充电电流是否有增加，以此来确定是否和运行功耗有关；

c) 快充满的电池肯定充电电流很小，所以请注意电池电压；

d) 在高温、大负载情况下，有可能是PMIC温度升高触发了输入限流。所以先把反馈温度（fb_temperature）提高，观察是否有效。再不行直接关闭温控试（fb_temperature设置为0）。

7. 拔掉电池再开机后，电量变了是怎么回事？我们的需求是拆卸电池后电量不会跳变，是否能满足？ a) 拔掉电池后PMIC完全掉电，此时再开机只能ocv电压查询ocv_table反推电量，所以是正常的，是一次电池的重新校正； b) 拆卸后希望电池电量不跳变？几乎不可能，除非软件做规避：把关机前的电量写到文件里，上电后再去读。客户有需求的话，请客户自己增加这部分规避处理的代码。
8. PMIC有温度反馈功能来调节输入电流，那么如何知道PMIC此时的内部温度？没有办法知道，设计时没有留这个功能。

9. 为什么log上打印的电流这么离谱，正负符号颠倒或者电流大小和实际差那么多？

请确认选用的是20/10毫欧的采样电阻且电阻精度够高；其次请确认焊点焊接干净，采样电阻应该位于BAT-和GND之间。

10. 为什么明明还没有满电，比如才3.9v就报finish了？一般是因为电池质量不好，并且设置的最大充电电流过大，导致电池板发生了自我保护，因此导致PMIC误报finish状态。
11. 为什么finish状态下电流的值在正负飘动（值较小）？这个是满电后的电流零点误差，没有关系。
12. 为什么都上报finish了，显示的电量才90%多，没有100%？由于库仑计会有累积误差，而充电结束是由硬件完成，两者之间会有一定误差，无法那么准确把握finish上报的时机，所以出现这种情况是正常的。这种情况下软件会做一个处理，即慢慢让电量逼近100%，对于终端用户来说是察觉不到这回事的，使用上没有问题。
13. 为什么运行过程电量计的电量这么不准，和ocv_table的值相差这么多？概念混淆。ocv_table是开路不带负载情况状态下的电压-电量的比值，并且我们只是在开机校正、休眠校正时用到这个表。所以这样的对比毫无意义，原理上就说不通。
14. 是否支持更换不同的规格的电池？不支持。换了电池，那么ocv曲线、内阻、容量等参数都需要重新测试，填写。
15. PMIC判断电池充满的条件是什么？需要同时满足三个条件：电压达到截止电压，电流达到finish电流，且不发生输入限流。
16. 为什么电池图标始终显示50%正在充电？请把test_power驱动disable掉
17. 为什么电池图标始终显示66%正在充电？当前没有接入电池；或者DTS的virtual_power被配置成了1，请配置成0。
18. 为什么没有识别到充电器插拔？

a) 内核4.4: rk818_charger.c负责充电器的检测，usb口的充电器和otg设备插拔都依赖于usb的通知链，请注意串口log是否有打印“rk818-charger: recieve xxx notifier event: xxx”等串口信息，如果没有的话则是usb通知链没有注册成功（可能性小），或者USB驱动出现了问题。rk818_charger.c负责充电器的检测，usb口的充电器和otg设备插拔都依赖于usb的通知链，请注意串口log是否有打印“rk818-charger: recieve xxx notifier event: xxx”等串口信息，如果没有的话则是usb通知链没有注册成功（可能性小），或者USB驱动出现了问题。

b) 内核3.10:

rk818_battery.c负责充电器的检测（包括电池电量），usb口的充电器和otg设备插拔都依赖于usb的通知链，请注意串口Log是否有打印“rk818-bat: recieve xxx notifier event: xxx”等串口信息，如果没有的话则是usb通知链没有注册成功（可能性小），或者USB驱动出现了问题。