RK818/RK816 电量计 开发指南

文件标识：RK-KF-YF-18

发布版本：V2.2.0

日期：2021-06-08

文件密级：□绝密 □秘密 □内部资料 ■公开

瑞芯微电子股份有限公司

Rockchip Electronics Co., Ltd.

地址： 福建省福州市铜盘路软件园A区18号

网址： www.rock-chips.com

客户服务电话： +86-4007-700-590

客户服务传真： +86-591-83951833

客户服务邮箱： fae@rock-chips.com

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、RK816	Linux3.10、Linux4.4、Linux4.19

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

版本号	作者	修改日期	修改说明
V1.0.0	陈健洪	2016-07-25	初始版本
V2.0.0	陈健洪	2017-05-25	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
V2.1.0	陈健洪	2018-05-31	调整文档格式和排版错误、无主要内容更新
V2.2.0	陈健洪	2021-06-08	增加4.19内核支持

目录

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：

（1）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。

（2）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。

（3）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。

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_delta，通过检测 V_delta 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：

  ```
  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，可以把松弛模式近似理解为一种开路状态。

- `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 3.10

rk818 驱动和宏配置：

```
CONFIG_BATTERY_RK818
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
```

rk816 驱动和宏配置：

```
CONFIG_BATTERY_RK816
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
```

4.3 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）：

```
CONFIG_BATTERY_RK818
CONFIG_CHARGER_RK818
drivers/power/rk818_battery.c            // 负责处理电量显示
drivers/power/rk818_charger.c            // 负责处理充电器检测、充电电压、电流设置
```

rk816 驱动和宏配置：

```
CONFIG_BATTERY_RK816
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
```

4.4 内核 4.19

rk818 驱动和宏配置：

```
CONFIG_BATTERY_RK818
CONFIG_CHARGER_RK818
drivers/power/supply/rk818_battery.c    // 负责处理电量显示
drivers/power/supply/rk818_charger.c    // 负责处理充电器检测、充电电压、电流设置
```

rk816 驱动和宏配置：

```
CONFIG_BATTERY_RK816
drivers/power/supply/rk816_battery.c    // 负责处理电量显示 + 充电器检测、充电电压、电流设置
```

5. DTS 配置

5.1 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。

```
battery {
    compatible = "rk818-battery"  // 如果是rk816，则改成"rk816-battery"
    ocv_table = <3400 3599 3671 3701 3728 3746 3762
                 3772 3781 3792 38163836 3866 3910
                 3942 39714002 4050 4088 4132 4183>;
    ntc_table = <43662 41676 39793 38005 36308 34696 33164
                 31709 30326 29011 27760 26570 25438 24361
                 23335 22358 21427 20540 19695 18890 18121
                 17389 16690 16022  14778 14197 13642 13113
                 12606 12122 11659 11216 10793 10388 10000
                 9629 9273 8933 8607 8295>;
    ntc_degree_from = <1 10>;
    design_capacity = <4000>;
    design_qmax = <4100>;
    bat_res = <120>;
    max_input_current = <2000>;
    max_chrg_current = <1800>;
    max_chrg_voltage = <4200>;
    sleep_enter_current = <300>;
    sleep_exit_current = <300>;
    sleep_filter_current = <100>;   // rk818不需要这个属性
    power_off_thresd = <3400>;
    zero_algorithm_vol = <3850>;
    energy_mode = <0>;
    fb_temperature = <105>;
    max_soc_offset = <60>;
```

```
    monitor_sec = <5>;
    virtual_power = <0>;
    power_dc2otg = <1>;
    dc_det_gpio = <&gpio0 GPIO_C1 GPIO_ACTIVE_LOW>;
    dc_det_adc = <1>;    // rk818不需要这个属性，rk816才可能需要
};
```

5.2 内核 4.4

RK818 DTS配置

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、temperature_chrg_table、dc_det_gpio是可选部分，其余是必选部分。

```
battery {
    compatible = "rk818-battery"
    ocv_table = <3400 3599 3671 3701 3728 3746 3762
                 3772 3781 3792 38163836 3866 3910
                 3942 39714002 4050 4088 4132 4183>;
    ntc_table = <43662 41676 39793 38005 36308 34696 33164
                  31709 30326 29011 27760 26570 25438 24361
                  23335 22358 21427 20540 19695 18890 18121
                  17389 16690 16022  14778 14197 13642 13113
                  12606 12122 11659 11216 10793 10388 10000
                  9629 9273 8933 8607 8295>;
    ntc_degree_from = <1 10>;
    temperature_chrg_table = <0 10 1000 1>, <0 20 1200 2>, <1 30 1400 5>, <0 40
700 3>;
    design_capacity = <4000>;
    design_qmax = <4100>;
    bat_res = <120>;
    max_input_current = <2000>;
    max_chrg_current = <1800>;
    max_chrg_voltage = <4200>;
    sleep_enter_current = <300>;
    sleep_exit_current = <300>;
    power_off_thresd = <3400>;
    zero_algorithm_vol = <3850>;
    energy_mode = <0>;
    fb_temperature = <105>;
    sample_res = <10>;
    max_soc_offset = <60>;
    monitor_sec = <5>;
    virtual_power = <0>;
    power_dc2otg = <1>;
    dc_det_gpio = <&gpio0 GPIO_C1 GPIO_ACTIVE_LOW>;
};
```

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。

```
rk818: pmic@1c {
    compatible= "rockchip,rk818";
    status= "okay";
    reg= <0x1c>;
    clock-output-names= "xin32k", "wifibt_32kin";
    interrupt-parent= <&gpio1>;
    interrupts = <21IRQ_TYPE_LEVEL_LOW>;
    pinctrl-names= "default";
    pinctrl-0= <&pmic_int_l>;
    rockchip,system-power-controller;
    rk818,support_dc_chg= <1>;/*1: dc chg; 0:usb chg*/
    wakeup-source;
    extcon = <&fusb0>;      // 重要！！！
    #clock-cells= <1>;

    battery {
            ...........
            ...........
    };
};
```

RK816 DTS配置

请参考上述rk818的DTS配置。差异点：rk816仅需要battery节点，不需要charger相关的配置。

5.3 内核 4.19

请参考内核4.4配置。

5.4 DTS参数说明

下列所有参数说明适用于所有内核版本，但各版本支持哪些参数，请参考上述各版本内核的DTS配置。

- `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，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`和`ntc_degree_from_v2`

  ntc_table的起始温度，即ntc_table[0]对应的温度值，单位：摄氏度。配置ntc_table时需要一起配置该属性。

  （1）3.10和4.4内核：使用`ntc_degree_from`

该属性由两个字段组成，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。

例如：ntc_degree_from = <1, 10>表示 -10 度，ntc_degree_from = <0, 10>表示 10 度。

（2）4.19内核：使用`ntc_degree_from_v2`

该属性由一个字段组成。

例如：ntc_degree_from_v2 = <(-10)>表示 -10 度，ntc_degree_from_v2 = <10>表示 10 度。

- `temperature_chrg_table`和`temperature_chrg_table_v2`

  温度充电表，可填写多组。作用：根据电池温度范围设置充电电流，不在范围内则默认配置为`max_input_current`和`max_chrg_current`。如果不需要本功能，请移除该属性。

  提醒： 因为该功能是后期增加的，建议用户务必检查当前驱动里是否有解析该属性，判断是否支持该功能。

  （1）4.4内核：使用`temperature_chrg_table`，只有rk818支持。

  格式：`temperature_chrg_table = <参数1  参数2  参数3  参数4>, ...`

  - 参数1：温度正负值，1：负数，0：正数
  - 参数2：温度值，与参数1配合使用
  - 参数3：充电电流
  - 参数4：温度偏差值，当参数3为正数时往大偏，为负数时往小偏

  范例：

  ```
  temperature_chrg_table = <0 10 1000 2>, <1 30 1400 5>;
  ```

  <0 10 1000 2> 含义：[10, 12]之间，设置充电电流1000mA

  <1 30 1400 5> 含义：[-35, -30]之间，设置充电电流1400mA

  （2）4.19内核：使用`temperature_chrg_table_v2`，rk818/rk816都支持。

  格式：`temperature_chrg_table_v2 = <参数1  参数2  参数3>, ...`

  - 参数1：温度下限
  - 参数2：温度上限
  - 参数3：充电电流

  范例：

  ```
  temperature_chrg_table_v2 = <(-40) 10 850>, <22 33 1000>, <40 55 1400>;
  ```

  <(-40) 10 850> 含义：[-40, 10]之间，设置充电电流850mA

  <22 33 1000> 含义：[22, 33]之间，设置充电电流1000mA

  <40 55 1400> 含义：[40, 55]之间，设置充电电流1400mA

- `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，单位：欧姆。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）：

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

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

  ```
  RK818: <4050, 4100, 4150, 4200, 4250, 4300, 4350>
  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vsys的后级 VCC_IO输出不稳定（一般是 3.3v）。如果某些平台的硬件设计上vsys的后级最高电压是3.0v，那么关机电压可以低于3.4v。

- `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）：

  ```
  < 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` 请仔细阅读和理解

  电池端的采样电阻阻值，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 20mR（默认支持）和10mR（需要驱动支持）。

  其中：4.4和4.19内核上的 rk818/rk816 电量计驱动都支持。3.10 内核的rk818/rk816包含如下提交才支持：

  ```
  fdc31b1 power: rk818-battery: support different ohm sample resistor
  0f23c4b power: rk816-battery: support different ohm sample resistor
  ```

  默认情况下一般都使用20mR采样电阻，采用10mR时会让`max_chrg_current`支持的范围x2：

  ```
  // rk816
  10mR: [2000, 2400, 2800, 3200, 3600, 4000, 4500, 4800]
  20mR: [1000, 1200, 1400, 1600, 1800, 2000, 2250, 2400]
  // rk818
  10mR: [2000, 2400, 2800, 3200, 3600, 4000, 4500, 4800, 5200, 5600, 6000]
  20mR: [1000, 1200, 1400, 1600, 1800, 2000, 2250, 2400, 2600, 2800, 3000]
  ```

  说明：

  - 充电电流范围x2后，`max_chrg_current`仍然按实际需求配置即可，如：4000mA充电就配置为4000。
  - 用10mR采样电阻可让充电电流的范围x2，但是电量计获取的电流、电量(mAh)误差也同样被x2；
  - 充电电流的翻倍只是理论值，有些值实际达是不到的。这涉及功率转换的问题，以5V/2A充电器为例：

  ```
  5V * 2A = U_bat * I_bat
  ```

  那么当U_bat取最小3.4v时，I_bat为最大值2.94A。这是在不考虑能量损失且所有电流都给电池的情况下才能达到，实际上适配器的电流还得优先分配给系统使用，剩余才给电池；另外也需要考虑电池自身的饱和状态。

- `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，单位：秒。电量计驱动是需要不停地进行轮询才能正常工作，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`

  假电池模式（测试模式）。有些产品形态上没有电池，单独使用5V/2A的适配器供电。但是因为电量计驱动统计的是电流和时间的积分，并不只是关注电压。所以虽然插着5V/2A适配器时系统输入电压一直是5V，但是显示的电量依然会慢慢下降到0%。

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关机或者供电。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，当前为充电状态且电量 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，只有rk816会用到。

5.5 注意事项

- 无法完全关闭电池充电

  rk818/rk816的CHRG_CTRL_REG1[7]：CHRG_EN用于使能/关闭电池充电。但这个硬件功能存在一点问题，无法正常使用。所以开发者在任何时候都不能配置为0，否则可能出现因为供电不足导致的稳定性问题。**workarund**的方法：

  RK818：把当前的输入电流配置为80mA档位，尽量让电池少充电，但是无法根治问题；

  RK816：把当前的输入电流配置为80mA档位；或者把BAT_CTRL_REG[6]：USB_SYS_EN配置为0，即直接关闭适配器端的输入！

- 不同版本内核使用不同的DTS配置：`ntc_degree_from`、`ntc_degree_from_v2`、`temperature_chrg_table`、`temperature_chrg_table_v2`

- 温度-充电电流的处理策略（仅适用于已支持`temperature_chrg_table_v2`的情况）：

  1. 如果配置了ntc_table，当电池温度高于上限值或者低于下限值时，最大输入电流被限制为80mA。因为无法关闭CHRG_EN禁止充电，只能以此做到尽量不给电池充电；
  2. 如果配置了temperature_chrg_table或者temperature_chrg_table_v2，根据不同温度范围配置不同充电电流，如果充电电流值小于PMIC支持的最低档位，则改为限制最大输入电流。如果不在温度限制范围内，则根据max_input_current、max_chrg_current、适配器类型配置。
  3. 用户对温度的需求不尽相同，可自行修改驱动。

6. 电量计开发

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，请参考上述章节。
3. 对电池进行校正，请参考上述章节。
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（0~100%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： 抓取前log 一定要打开电量计的 debug 信息！！一些常见问题请参考《常见问题分析定位》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（rk816上返回值的BIT(1)应该为 1）。
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，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。再不行的话直接关闭温控试（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、休眠较正时用到这个表。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。

（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。