

密级状态：绝密() 秘密() 内部资料() 公开(☒)

RICOH619 电量计的相关问题说明

(技术研发部，手机组)

文件状态： [] 草稿 [<input checked="" type="checkbox"/>] 正式发布 [] 正在修改	文件标识：	RICOH619 电量计的相关问题说明
	当前版本：	1.0
	作 者：	张晴
	完成日期：	2014-9-19

版 本 历 史

版本号	作者	修改日期	修改说明
1.0	张晴	2014-9-19	初稿

目录

主要问题说明:	4
一、 充电不能充满.....	4
1.1 充电界面或者关机界面下无法充满.....	4
1.2 进入系统后无法充满.....	5
二、 低电量下电池耗电过快.....	6
三、 99%-100%充电时间过久	6
四、 DEBUG 相关方法	7
五、 注意.....	7

主要问题说明:

一、 充电不能充满

1.1 充电界面或者关机界面下无法充满

如果是在充电界面下或者关机界面下面无法充满，问题在 UBOOT 中，需要核对 UBOOT 中设置：

Drivers/power/fuel_gauge/fg_ricoh619.c 中

- (1) 满充电压是否设置？（4.35V 或者 4.2V 电池）

函数 ricoh619_get_voltage () 中：

```
val=i2c_reg_read(pmic->hw.i2c.addr, BATSET2_REG);
if(vol > 4100&&(val!=0x44)){
    i2c_reg_write(pmic->hw.i2c.addr, CHGCTL1_REG,
        i2c_reg_read(pmic->hw.i2c.addr, CHGCTL1_REG) | 0x08);
    i2c_reg_write(pmic->hw.i2c.addr, BATSET2_REG, 0x44); /* VFCHG 4.35v) */
    i2c_reg_write(pmic->hw.i2c.addr, CHGCTL1_REG,
        i2c_reg_read(pmic->hw.i2c.addr, CHGCTL1_REG) & 0xf7);
}
```

满充状态标志判断 (get_capacity ()):

```
chgstate = i2c_reg_read(PMU_I2C_ADDRESS, CHGSTATE_REG);
if ((chgstate & 0x3f) == 0x04){
    printf("%s chg complete\n", __func__);
    cap = 100;
}
```

- (2) 充电电流是否正常？（主要 USB 或者 ADP 检测）

主要是核对充电器，是单口充电还是双口充电，支持的充电类型：

ADP、USB 口都支持；

ADP 口充电；

USB 口充电（USB 接到 619 的 ADP 输入口）：打开 SUPPORT_USB_CONNECT_TO_ADP；

设置电流函数：ricoh619_charger_setting ()，可以增加打印确认下。

- (3) 充电时间是否设置？

RICOH619 充电只有 5 小时，超过 5 小时就会关闭充电，而目前客户使用的电池越来越大，

5 小时有可能会不满足要求，所以我们需要将 timeout 功能关闭。

```
/******DISABLE_CHARGER_TIMER******/
val = i2c_reg_read(pmic->hw.i2c.addr, CHGSTATE_REG);
if ((val > 4000) && (val & 0xc0)) {
    //printf("*****DISABLE_CHARGER_TIMER*****\n");
    val = i2c_reg_read(pmic->hw.i2c.addr, TIMSET_REG);
    val2 = val & 0x03;
    if (val2 == 0x02){
        i2c_reg_write(pmic->hw.i2c.addr, TIMSET_REG, val | 0x03);
    } else {
        i2c_reg_write(pmic->hw.i2c.addr, TIMSET_REG, val & 0xfe);
        i2c_reg_write(pmic->hw.i2c.addr, TIMSET_REG, val | 0x02);
    }
}
}
/*******/
```

1.2 进入系统后无法充满

核对代码就是 kernel 中 ricoh619 驱动的配置

在 arch/arm/boot/dts/ricoh619.dtsi 中

- (1) 满充电压是否设置? (4.35V 或者 4.2V 电池)

```
battery {
    compatible = "ricoh,ricoh619-battery";
    ricoh,alarm-vol-mv = <3300>;
    ricoh,multiple = <100>;
    ricoh,monitor-time = <1>;

    ricoh,ch-vfchg = <0xff>;/* VFCHG    = 0 - 4 (4.05v, 4.10v, 4.15v, 4.20v, 4.35v) */
    ricoh,ch-vrchg = <0xff>;/* VRCHG    = 0 - 4 (3.85v, 3.90v, 3.95v, 4.00v, 4.10v) */
    ricoh,ch-vbatovset = <0xff>;/* VBATOVSSET = 0 or 1 (0 : 4.38v(up)/3.95v(down) 1: 4.53v(up)/4.10v(down)) */
```

默认设置是支持 4.2V 电池, 如果是 4.35V 电池需要将上述参数修改为:

```
ricoh,ch-vfchg = <0x04>;/* VFCHG    = 0 - 4 (4.05v, 4.10v, 4.15v, 4.20v, 4.35v) */
ricoh,ch-vrchg = <0x04>;/* VRCHG    = 0 - 4 (3.85v, 3.90v, 3.95v, 4.00v, 4.10v) */
ricoh,ch-vbatovset = <0x01>;/* VBATOVSSET = 0 or 1 (0 : 4.38v(up)/3.95v(down) 1: 4.53v(up)/4.10v(down)) */
```

- (2) 充电电流是否正常? (主要 USB 或者 ADP 检测)

主要是核对充电器, 是单口充电还是双口充电, 支持的充电类型:

ADP、USB 口都支持;

ADP 口充电;

USB 口充电 (USB 接到 619 的 ADP 输入口): 打开 `//#define`

`SUPPORT_USB_CONNECT_TO_ADP;`

- (3) 截止充电电流

如果说充电电流已经为 0, 0xbd 寄存器的 bit4-0 是 4, 充电已经完成了, 但是电量没有显示 100%, 这个可以降低截止充电电流。

```
ricoh,chg-min-usb = <0x11>;/* ILM_USB = 0 - 0x10 (100mA - 3000mA) */
ricoh,chg-icchg = <0x03>;/* ICCHG = 0 - 3 (50mA 100mA 150mA 200mA) */
ricoh,fg-target-vsys = <3000>;/* This value is the target one to DSOC=0% */
```

二、 低电量下电池耗电过快

在 arch/arm/boot/dts/ricoh619.dtsi 中

```
ricoh,chg-icchg = <0x03>;/* ICCHG = 0 - 3 (50mA 100mA 150mA 200mA) */
ricoh,fg-target-vsys = <3000>;/* This value is the target one to DSOC=0% */
ricoh,fg-target-ibat = <1000>;/* This value is the target one to DSOC=0% */
ricoh,fg-noff-what = <0>;/* setting value of 0 per What */
```

可以微调 fg-target-vsys, 这个值是在低电的时候, 如果 VSYS 电压触发 值, 电量就会迅速逼近 0%, 因为如果 VSYS 电压持续被拉低会导致系统其他电源异常, 可能会出现花屏。(VSYS = VBST-I * (电池内阻+70 毫欧))

如果放电到自动关机后, 电池电压较高, 电池冗余电量较多, 可以在不修改电池曲线的情下降低 fg-target-vsys 电压。

如果低电的情况下出现闪屏或者系统异常, 自动关机后电池电压过低, 可以通过提高 fg-target-vsys 电压。

一般 fg-target-vsys 参考电压 3.0-3.2V。

三、 99%-100%充电时间过久

Ricoh619 的截止充电电流最大 200MA, 我们充电, 先是快速充电, 然后进入恒压充电, 恒压充电后, 充电电流会慢慢降低, 一直降低到 200MA, 才认为是充满。

但是目前客户电池容量比较大, 如果恒压充电截止电流 200MA, 充电时间就是会很久。所以我们算法, 在 99%后 15 分钟修正电量到 100%。(这最后的 1%几乎没有电量)

目前平台上的代码 99%-100%修正只有运行时会计时, 待机后是无法计时的, 如果需要待机也计时间去修正电量需要打上相应的补丁。

四、 **DEBUG** 相关方法

cat sys/kernel/debug/ricoh619

跟电量计相关的寄存器为:

0xb6 //adp 输入电流限制

0xb7 //usb 输入电流限制

0xb8 //充电电流设置

0xbb //满充电压 4.35V 电池: 0x44 ;4.2v 电池: 0x33

0xbd //充电状态, 如果是充电完成 bit4-0 :0x04

五、 **注意**

由于 ricoh619 电量计有自学习功能, 如果修改跟电量计相关 参数后, 必须将电池拔插一次, 否则内部还是会按照原来储存的学习数据。

修改电量计参数、电池曲线, 都需要拔插电池或者做一次完整的充放电后, 电量测试才可靠。