

电池相关知识以及软件调试SOP

笔记本: charaina

创建时间: 2014/2/21 11:32

更新时间: 2014/2/21 11:39

作者: 奉光跃

电池的常识

电池定义

电池 (Battery) 指盛有电解质溶液和[金属电极](#)以产生电流的杯、槽或其他[容器](#)或复合容器的部分空间, 能将化学能转化成电能的装置, 具有正负极之分。

电池的性能参数

电池主要性能包括电动势、[额定容量](#)、额定电压、开路[电压](#)、内阻、充放电[速率](#)、[阻抗](#)、寿命和[自放电率](#)。电动势取决于电极材料的化学性质, 与电池的大小无关。电池储存的能量有限, 电池所能输出的总电荷量为电池的容量, 通常用[安培](#)作单位, 它也是电池的一个重要[参数](#)。

额定容量

在设计规定的条件 (如温度、放电率、[终止电压](#)等) 下, 电池应能放出的最低容量, 单位为安培小时, 以符号C表示。容量受放电率的影响较大, 所以常在字母C的右下角以阿拉伯数字标明放电率, 如C20=50, 表明在20小时率下的容量为50安·小时。电池的[理论容量](#)可根据电池反应式中电极活性物质的用量和按[法拉第定律](#)计算的活性物质的[电化学](#)当量精确求出。由于电池中可能发生的副反应以及设计时的特殊需要, 电池的[实际容量](#)往往低于理论容量。

额定电压

电池在常温下的典型[工作电压](#), 又称标称电压。它是选用不同种类电池时的参考。电池的实际工作电压随不同使用条件等于正、负电极的平衡电极电势之差。它只与电极活性物质的种类有关, 而与活性物质的数量无关。电池电压本质上是[直流电压](#), 但在某些特殊条件下, 电极反应所引起的金属晶体或某些成相膜的相变会造成电压的微小波动, 这种现象称为[噪声](#)。波动的幅度很小但[频率](#)范围很宽, 故可与[电路](#)中自激噪声相区别。

开路电压

电池在开路[状态](#)下的端电压称为开路电压。电池的开路电压等于电池在断路时 (即没有电流通过两极时) 电池的正极电极电势与负极的电极电势之差。电池的开路电压用V开表示, 即 $V_{\text{开}} = \Phi_{+} - \Phi_{-}$, 其中 Φ_{+} 、 Φ_{-} 分别为电池的正负极[电极电位](#)。电池的开路电压, 一般均小于它的电动势。这是因为电池的两极在电解液溶液中所建立的电极电位, 通常并非平衡电极电位, 而是稳定电极电位。一般可近似认为电池的开路电压就是电池的电动势。

内阻

电池的内阻是指电流通过电池内部时受到的阻力。它包括欧姆内阻和极化内阻, 极化内阻又包括[电化学极化](#)内阻和浓差极化内阻。由于内阻的存在, 电池的工作电压总是小于电池的电动势或开路电压。电池的内阻不是常数, 在充放电过程中随时间不断变化 (逐渐变大), 这是因为活性物质的组成, 电解液的浓度和温度都在不断的改变。欧姆内阻遵守[欧姆定律](#), 极化内阻随电流密度增加而增大, 但不是线性关系。常随电流密度增大而增加。

内阻是决定电池性能的一个重要指标，它直接影响电池的工作电压，工作电流，输出的能量和功率，对于电池来说，其内阻越小越好。

阻抗

电池内具有很大的电极-电解质界面面积，故可将电池等效为一大电容与小电阻、电感的串联回路。但实际情况复杂得多，尤其是电池的阻抗随时间和直流电平而变化，所测得的阻抗只对具体的测量状态有效。

充放速率

有时率和倍率两种表示法。时率是以充放电时间表示的充放电速率，数值上等于电池的额定容量(安·小时)除以规定的充放电电流(安)所得的小时数。倍率是充放电速率的另一种表示法，其数值为时率的倒数。原电池的放电速率是以经某一固定电阻放电到终止电压的时间来表示。放电速率对电池性能的影响较大。

寿命

储存寿命指从电池制成到开始使用之间允许存放的最长时间，以年为单位。包括储存期和使用期在内的总期限称电池的有效期。储存电池的寿命有干储存寿命和湿储存寿命之分。循环寿命是蓄电池在满足规定条件下所能达到的最大充放电循环次数。在规定循环寿命时必须同时规定充放电循环试验的制度，包括充放电速率、放电深度和环境温度范围等。

自放电率

电池在存放过程中电容量自行损失的速率。用单位储存时间内自放电损失的容量占储存前容量的百分数表示。

锂电池

以锂为负极的电池。它是60年代以后发展起来的新型高能量电池。按所用电解质不同分为：①高温熔融盐锂电池；②有机电解质锂电池；③无机非水电解质锂电池；④固体锂电池

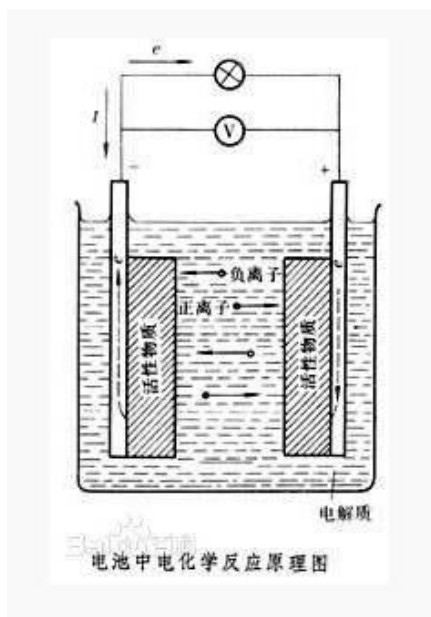
电解质锂电池；⑤锂水电池。锂电池的优点是单体电池电压高，比能量大，储存寿命长(可达10年)，高低温性能好，可在-40~150℃使用。缺点是价格昂贵，安全性不高。另外电压滞后和安全问题尚待改善。大力发展动力电池和新的正极材料的出现，特别是磷酸亚铁锂材料的发展，对锂电发展有很大帮助

1、锂电池的结构。

锂电池的结构是十分简单的，如图所示。至于实际的电池，结构要复杂一些，比如18650这种圆筒形的电池里面图中的各层以薄片形式存在，卷起而成圆筒形电池。另外还要有排气的安全阀，防止锂离子通过负极逃逸等机构。

2、锂电池的工作原理。

刚刚制造完毕的锂电池，负极表面的介质层是刚刚形成的，过程是当电解质和负极材料比如石墨接触后，锂离子进入石墨的片层结构，而有机大分子则留在石墨表面堆积而成介质膜，属于混合物，里面有很多分子碎片以极其弱的键混合在一起，微观上无定形。这个现象是自然界的恩赐，有了它，锂电池才可以工作，否则有机大分子在充电的时候直接进入石墨将把碳的片层结构破坏，电镜下可以看到石墨的每一层象被撬开一样。这就是锂电池刚刚生产完毕的第一次化学反应。之后随着充电和放电，这层介质膜慢慢稳定下来。充电的时候外加电场将电解质中的锂离子通过介质膜推入石墨中储存，放电的时候石墨中的锂离子通过该层介质膜释放出来。锂离子通过这层介质的细节很复杂，至今学术界的观点也不一致，但是你可以想象它是一个过滤网。这层介质很重要，如果它的性质改变了，将会导致或者冲不进电，或者放不出来，总之电池的充放电表现在很大程度上受制于这层介质。由于锂电池的充放电过程十分简单，所以理论上锂电池不存在记忆效应。锂电池一般经过500次完全充放电循环还应该70~80%的电量，而我们使用中的电池往往达不到这种表现，原因很多。



3、锂电池的优点。

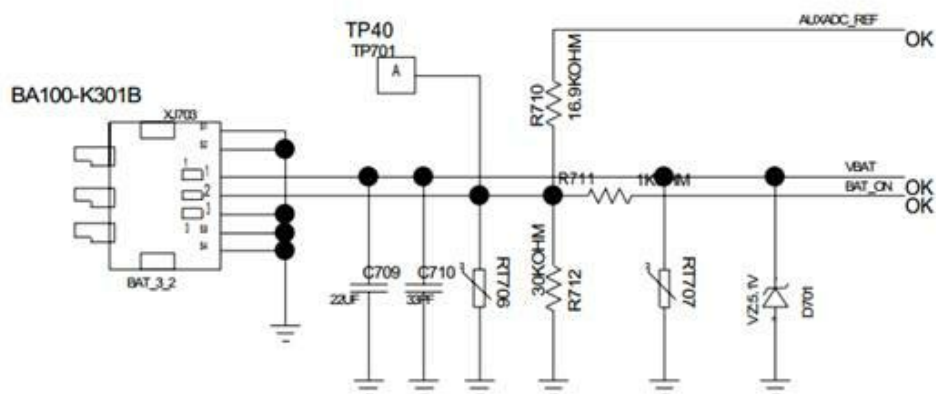
结论，高体积[能量密度](#)、高重量[能量密度](#)、高电压

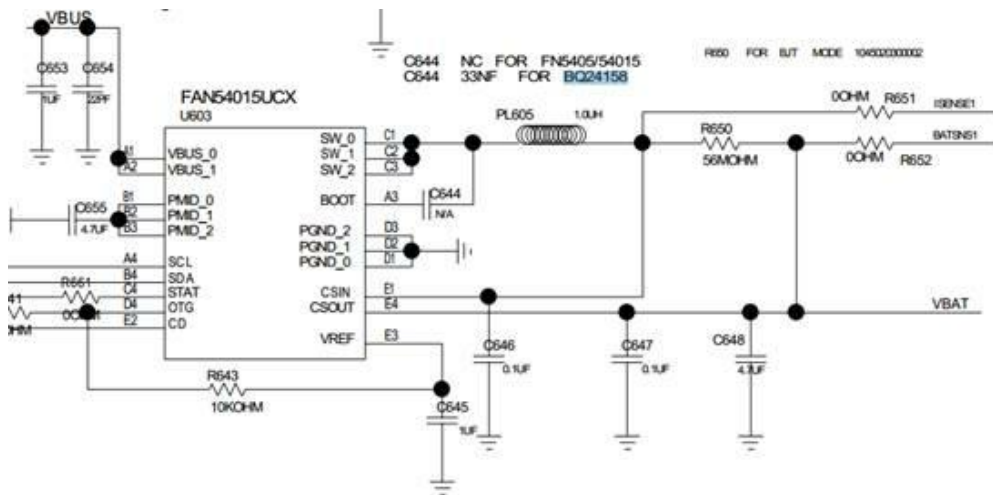
电池的优势

利用电池作为能量来源，可以得到具有稳定电压，稳定电流，长时间稳定供电，受外界影响很小的电流，并且电池结构简单，携带方便，充放电操作简便易行，不受外界气候。

MTK平台电池相关知识

硬件相关





软件配置相关

MT6572/MT6582/MT6592平台的电池电量建议使用SW fuel gauge计算方法。

在cust_battery_meter.h文件里，把SOC_BY_SW_FG宏打开。如图所示：



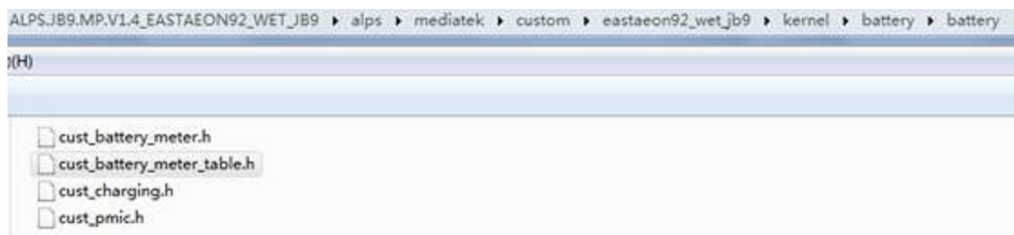
为了使电池电量检测准确，一般要检查下面这个方面。

一、电池检测的硬件电路是否异常。

电池的容量是随着温度而变化的，所Fuel gauge的算法是跟温度相关的。电路里要检查相关的硬件电路是否有正常工作。咨询硬件主板工程师，主板这块的电阻分压阻值是按照10K还是47来设计的。咨询电池厂或是看电池规格书，得到电池的NTC电阻是10K还是47K。然后可以得出主板电路与所要求的电池的NTC电阻值是否匹配的结论。若是得出了相关的结论，当然软件代码里，这时就要设定相关的阻值了。

代码里此值，是与NTC电阻温度表里相关联的。

在文件cust_battery_meter_table.h，打开10K或是47K。示例是打开了10K电阻。



```
// =====  
// define  
// =====  
#define BAT_NTC_10 1  
#define BAT_NTC_47 0  
  
#if (BAT_NTC_10 == 1)  
#define RBAT_PULL_UP_R 16900  
#define RBAT_PULL_DOWN_R 27000  
#endif
```

检查电流采样电阻BATSENSE是否足够的接近电源管理IC，若电流采样电阻BATSENSE离电源管理IC太远，也会导致计算出来的电量不够准确。这要与硬件工程师沟通。

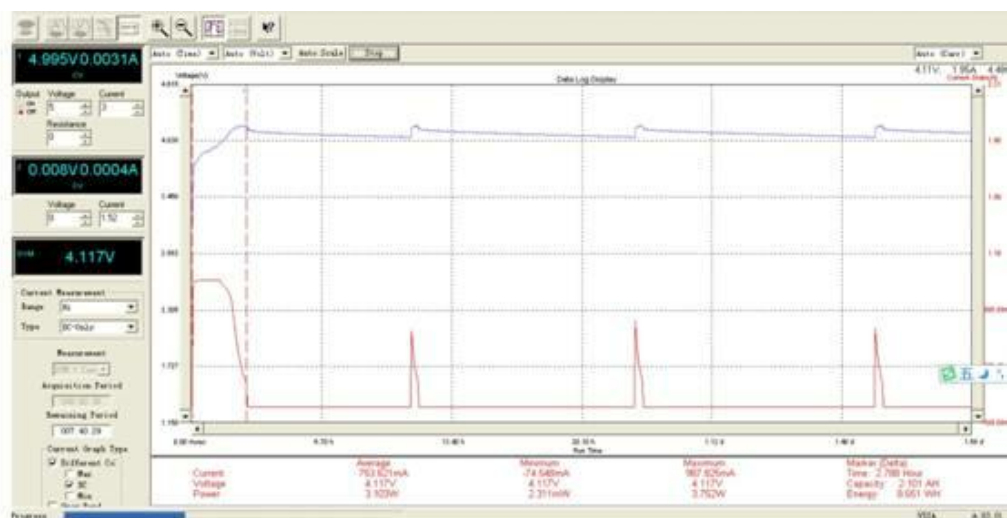
二、电池的充放电曲线是否异常。

充放电曲线直接影响到电池电量的显示，所以充放电曲线至关重要，一个项目开始时，一定要关注充放电曲线。检查电池充放电曲线是否能正常截止，我司的代码里设置的截止电流为146mA。

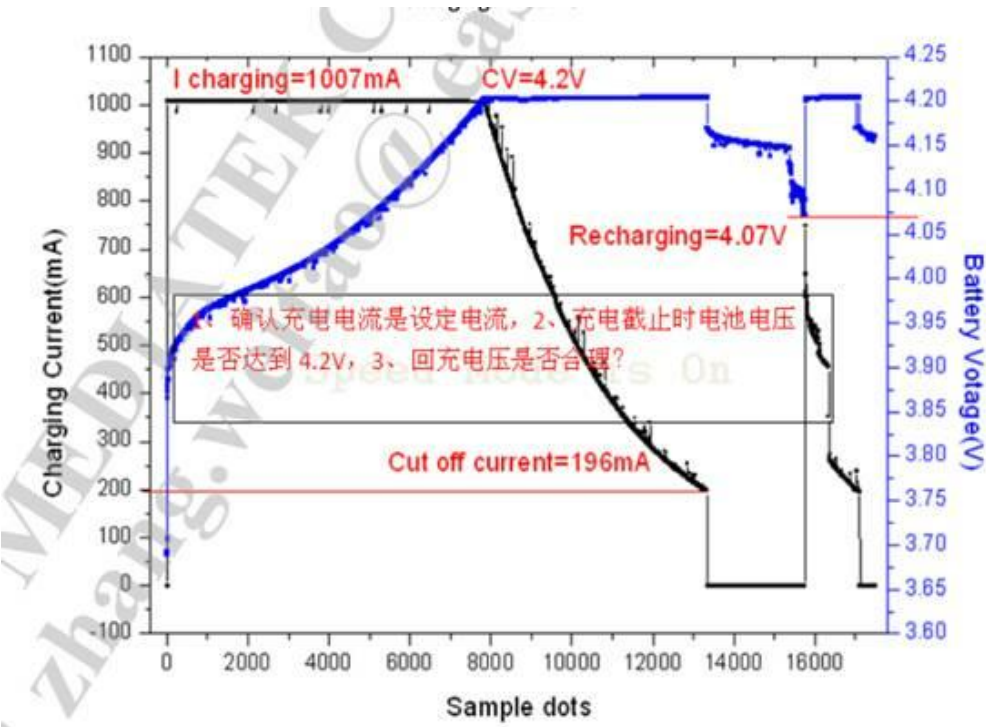
检查充放电曲线是否正常，需要带着下面的这个疑问仔细检查：

- 1、若为BQ24158外置充电芯片，检测CC MODE, CV MODE过程里，它的电压与电流是否有异常？
- 2、若是PMIC内置充电芯片，检测CC MODE, CV MODE, 涓流等过程里，它的电压与电流是否有异常？
- 3、CC MODE里的充电电流是否是代码里设定的电流值？
- 4、充电截止时，电压值是否接按电池标称的最高值？
- 5、二次充电过程启动的电压值是否在合理的范围内？
- 6、电池的容量与充电电流的时间积分是否在正常的范围内？

下图我司实测试正常的充放电曲线：



MTK为更好理解，有适当处理：



三、电池ZCV参数的正确导入。

ZCV电池参数表，它包括了50度，25度，0度，-10度四个温度里相关参数。

每个温度里OCV开路电压，VC负载电压，mAh累计放电容量，R(battery)电池内阻，DOD放电深度百分比，R (x1000) 电池内阻放大1000倍。

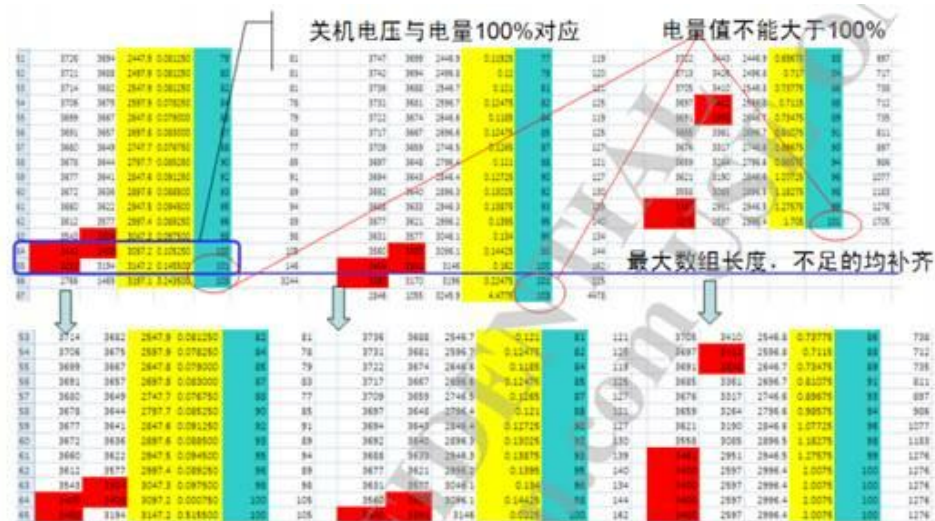
截图如下：

50度	25度	0度	-10度
OCV	OCV	OCV	OCV
VC	VC	VC	VC
mAh	mAh	mAh	mAh
R(battery)	R(battery)	R(battery)	R(battery)
DOD	DOD	DOD	DOD
R (x1000)	R (x1000)	R (x1000)	R (x1000)

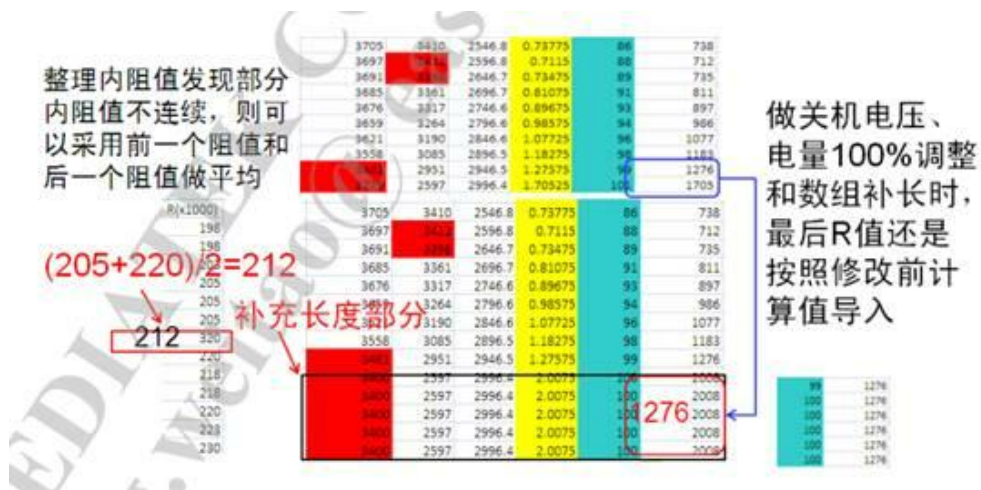
首先客户给过来的ZCV参数表，四个温度的长度是不一样的，我们可以选定参数最长参数表做为最终的参数长度，其他四组使用最后一个复制来补上，以保证四个温度的数据长度是一致的。

长度统一后，第一个开路电压对应的内阻值以第二个开路电压对应的内阻值来替代。检查四个温度里，关机时它的放电机电电压值为3.4V，低于3.4V的电压值，使用3.4V来替换。

检查四个温度里，放电深度百分比超过100的值，使用100来替换。



整理内阻时发现部分内阻不连续，可以采用前一个电池内阻值与后一个电池内阻值取平均来得到。



做关机电压3.4V，放电深度百分比调整，或是数组补长时，以最后一个内阻值来替换。

注意每个温度数组最后一个数据不要加上逗号。

[illegible]

为数组等长补充的数据

```
// Battery profile for actual temperature. (The size sh  
BATTERY_PROFILE_STRUC battery_profile_temperature[] =  
{  
(0, 0),  
(0, 0),  
(0, 0),  
(0, 0),  
(0, 0),  
(0, 0),  
(0, 0),
```

[illegible]

The image shows a C code snippet for defining current limits based on temperature. The code uses preprocessor directives to define values for different temperatures. A chart on the right illustrates the linear relationship between temperature and current, with data points for 50, 25, 0, and -25 degrees Celsius.

Code Snippet:

```

/* Qmax for battery */
#define Q_MAX_POS_50 1399
#define Q_MAX_POS_25 1499
#define Q_MAX_POS_0 1377
#define Q_MAX_NEG_10 1237

#define Q_MAX_POS_50_H_CURRENT 1377
#define Q_MAX_POS_25_H_CURRENT 1473
#define Q_MAX_POS_0_H_CURRENT 1247
#define Q_MAX_NEG_10_H_CURRENT 819

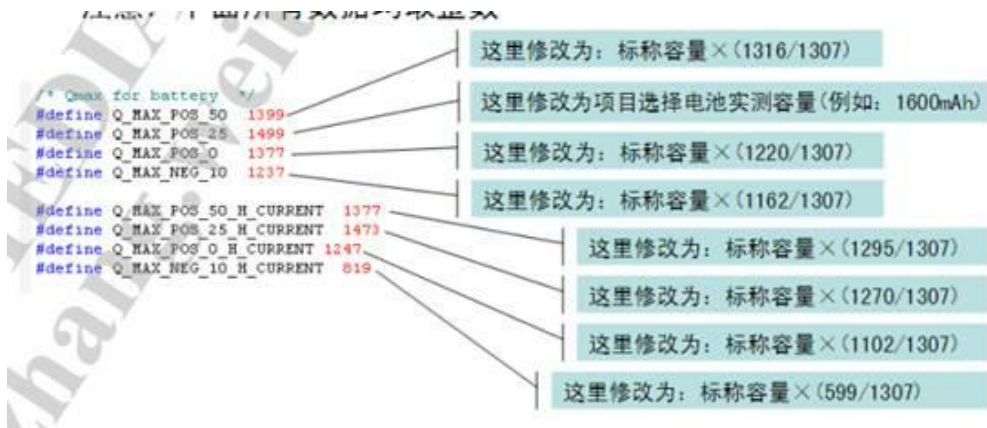
/* Discharge Percentage */
#define OAM_DS 1 // 1 : DS, 0: D2

```

Temperature Calibration Chart:

Temperature (°C)	Current (mA)
50	1629
25	1590
0	1519
-25	1404

若使用MTK默认的ZCV参数，下面的值这样计算：



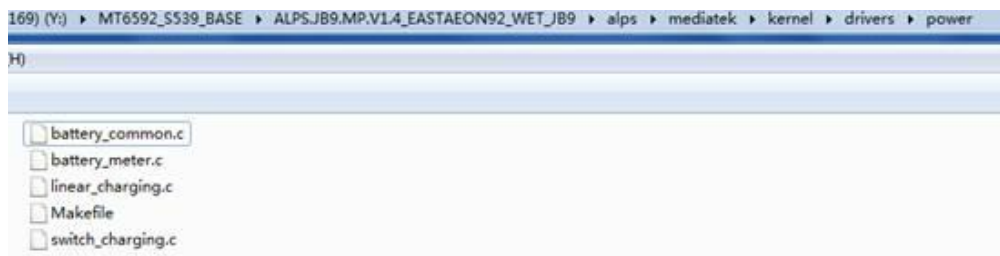
四、电池初始化电量确认。

确认开机时, 手机通过ADC读取的D0结果与ZCV数据表的OCV的数据对比, 检查是否差异很大。此值是初始化电量的重要依据, 直接影响后面的电量计量的准确度。若3.8V以上的误差都超过10%, 请确认测试方法, ADC精度以及BATSENSE的走线是否正常。

软件调试相关

一般按照上述的配置后, 若客户提供的ZCV电池参数准确, 电量显示的误差应该是在可接受的范围内。但是出现一些非正常的现象, 例如电量快速拉升, 电量虚高, 电池充不满等。这时需要调试代码, 查找那里出了问题。

电池充放电异常调试, 主要是查找充放电的流程是否出现异常。它的框架代码在battery_common.c, linear_charging.c, switch_charging.c这几个文件里。如图所示



PMIC的充放电流程在linear_charging.c里, BQ24158/FAN5405等充放电流程在switch_charging.c。

主体充放电框架代码为:

```

void mt_battery_charging_algorithm()
{
    switch(BMT_status.bat_charging_state)
    {
        case CHR_PRE :
            BAT_PreChargeModeAction();
            break;

        case CHR_CC :
            BAT_ConstantCurrentModeAction();
            break;

        case CHR_TOP_OFF :
            BAT_TopOffModeAction();
            break;

        case CHR_BATFULL:
            BAT_BatteryFullAction();
            break;

        case CHR_HOLD:
            BAT_BatteryHoldAction();
            break;

        case CHR_ERROR:
            BAT_BatteryStatusFailAction();
            break;
    }
}

```

示例

例如前段时间，神州项目电池充不满的例子。

主观上判断它主要是CC MODE出了问题，无法正常进入到BAT FULL MODE。这时可以加打印信息，看看BAT_ConstantCurrentModeAction发生了什么？

```

case CHR_CC :
    BAT_ConstantCurrentModeAction();
    break;

```

这样很容易找到从CC MODE进入到FULL MODE的判断的函数：

```

static kal_uint32 charging_full_check(void)
{
    kal_uint32 status;
    // yyf
    battery_charging_control(CHARGING_CMD_GET_CHARGING_STATUS,&status);
    if(KAL_TRUE == status )
    {
        if((BMT_status.UI_SOC != 100) || ( BMT_status.bat_vol < 4000 ))
        {
            status = KAL_FALSE;
            printk("%s %d\n",__func__,__LINE__);
        }
    }
    // end
    return status;
}

```

从函数上对应到充电的现象，就会发现问题出现BMT_status.UI_SOC != 100这条判断上了。

然后按如下修改即可验证修改是否有成效?

```
static kal_uint32 charging_full_check(void)
{
    kal_uint32 status;

    battery_charging_control(CHARGING_CMD_GET_CHARGING_STATUS,&status);
    if ( status == KAL_TRUE) {
        g_full_check_count++;
        if (g_full_check_count >= FULL_CHECK_TIMES) {
            return KAL_TRUE;
        }
        else
            return KAL_FALSE;
    } else {
        g_full_check_count = 0;
        return status;
    }
}
```

上述修改，经过反复压力测试，问题没有复现，说明修改是可以解决此类问题的。