



Prolin 电池管理系统说明

V 1.1.4

百富计算机技术(深圳)有限公司

PAX Computer Technology (Shenzhen) Co.,Ltd.

版权声明

Copyright © 2000-2020 富计算机技术（深圳）有限公司保留所有权利。未经百富计算机技术（深圳）有限公司书面许可，不得重新制作或散发本文档的任何内容。尽管百富计算机技术（深圳）有限公司努力确保本文档内容的准确性，但仍可能包含错误或遗漏。本文档所包含的信息如有更改，恕不另行通知。本文中的事例和样例程序仅用作说明演示目的，可能并不适用于您的要求，在您的软件投入商业使用前，请您仔细验证事例或样例程序的适用性。

修改记录

| 日期 | 版本号 | 备注 | 作者 |
|------------|--------|--|-----|
| 2018-12-08 | V1.0.0 | 初稿。 | 张梦莹 |
| 2018-12-19 | V1.0.1 | 修改充电参数。 | 张梦莹 |
| 2019-03-19 | V1.1.0 | 废弃上一版本文档，在《S920 BMS 设计说明(V1.1.3)》基础上删除了“续航能力实测数据”，“编程指南汇总”和“常见问题”章节。 | 张梦莹 |
| 2019-04-01 | V1.1.1 | 1. 更新“适用平台”小节； 2. 更新“自动识别工作模式”小节。 | 张梦莹 |
| 2019-04-22 | V1.1.2 | 更新“异常检测”和“供电记录”章节。 | 张梦莹 |
| 2019-08-16 | V1.1.3 | 更新“充放电参数”，“主动设置工作模式”，“自动识别工作模式”，“异常定义”和“捕获异常”章节。 | 张梦莹 |
| 2020-03-23 | V1.1.4 | 在“电量显示”章节新增示例代码。 | 张梦莹 |

目录

| | | |
|-----|------------------|----|
| 1 | 概述 | 1 |
| 1.1 | 背景 | 1 |
| 1.2 | 适用平台 | 2 |
| 1.3 | BMS 框架图 | 2 |
| 2 | 工作模式定义 | 3 |
| 2.1 | 模式定义 | 3 |
| 2.2 | 充放电参数 | 3 |
| 3 | 工作模式的识别及迁移 | 5 |
| 3.1 | 主动设置工作模式 | 5 |
| 3.2 | 自动识别工作模式 | 5 |
| 4 | 电量检测及显示 | 9 |
| 4.1 | 获取电池电量 | 9 |
| 4.2 | 电量显示 | 10 |
| 5 | Boot 逻辑 | 12 |
| 5.1 | 为什么要上电开机 | 12 |
| 5.2 | 开关机逻辑说明 | 12 |
| 6 | 异常检测 | 13 |
| 6.1 | 异常定义 | 13 |
| 6.2 | 捕获异常 | 14 |
| 6.3 | 异常保护 | 16 |
| 7 | 供电记录 | 17 |

图表目录

| | |
|---------------------|---|
| 图 1.1 BMS 框架图 | 2 |
| 图 3.1 模式迁移流程图 | 7 |
| 图 3.2 写数据库操作 | 8 |
| 表格 2.1 区间定义 | 3 |

1 概述

1.1 背景

S920 是一款配置了锂电池的终端，原来定位为手持终端，但一些 PN 也配置了充电底座。S920+底座的配置颇为常见。在配置底座的使用场景中，终端大概率是比较长时间是放置在底座上以便随时进行充电。

S920 配套的锂电池为一节 18650 锂电池，容量为 2600mAh。因此设计了相应的充放电电路，在 V09 开始更是配备了库仑计。早期，S920 一直是通过硬件电路来控制电池的充放电。

及至后来，我们发现硬件电路控制充放电存在某些不足，也不能根据实际的质量要求、场景要求提供灵活而由安全的控制。因此，我们从 2.4.132 版本开始，逐步在 Prolin 中加入了对电池的回充电压、满充电压等的控制，这些相关的设计我们称之为 BMS(Battery Management System)。

BMS 的设计基于锂电池特性，同时，BMS 需要达到两个目的：

- 减少锂电池的充放电循环数量
- 避免锂电池长期处于浮充状态

后续的篇幅将从多个维度对 BMS 进行说明，帮助读者更好的理解 BMS 技术方案。

1.2 适用平台

目前该方案适用于 Prolin-2.4、Prolin-phoenix-2.5、Prolin-pelican-2.7 和 Prolin-peng-2.8 平台，分别从 Prolin-2.4.138、Prolin-phoenix-2.5.29、Prolin-pelican-2.7.41、Prolin-peng-2.8.2 以上的版本开始支持，机型包括 S920（带电量计）、D190、D195、QR68、QR65 和 Q92。

1.3 BMS 框架图

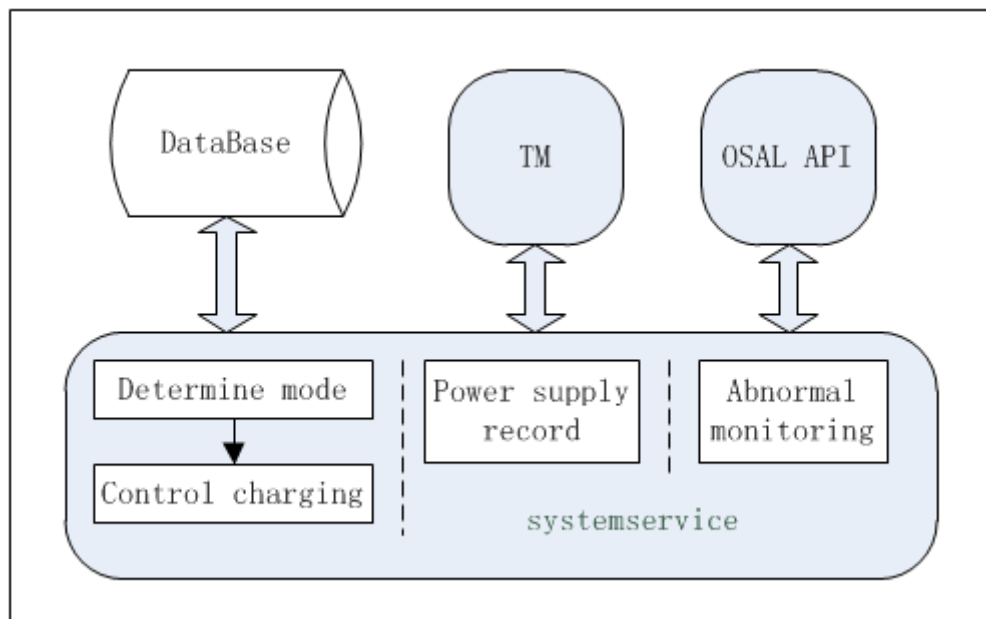


图 1.1 BMS 框架图

2 工作模式定义

我们从电池的续航能力以及电池的安全两个方面定义了 2 个终端工作模式，并设计了如何识别这些模式以及模式之间如何迁移。

2.1 模式定义

- 桌面模式（Counter-topMode）

表示终端以使用外部电源供电为主，外部电源包括充电底座、电源适配器和无线充电

- 移动模式（Mobile Mode）

表示终端以电池供电为主，使用时一般不插外电

我们对两种工作模式设计了不同的充放电参数，详见下文。

2.2 充放电参数

表格 2.1 区间定义

| | 满充电量 | 回充电量 |
|------|------|------|
| 移动模式 | 100% | 85% |
| 桌面模式 | 80% | 65% |

上述表格中,满充电量表示电池充电到某个程度后即停止充电。由于电池有自放电,以及部分耗电较多的无线模块可能存在峰值高的电流,电源适配器不足以满足无线模块峰值电流,所以仍需从电池取电,这样即使保持电源适配器供电,电池在满电被断开充电后,其电量仍会逐渐减少,当减少到一定程度后再开启电池充电,这个重复充电阈值成为回充电量。

无线模块的峰值电流出现在模块向基站注册网络时。举个例子,终端在开始时,峰值电流可能达到 5V2A,后续正常工作只需要 5V600mA。因为一些客观因素,电源适配器一般只提供 5V1A 供电能力。在特定应用场景中,当电源适配器不足以满足终端耗电时,终端则会从电池取电。当电池适配器供电超过终端耗电时,则会给电池充电。

后续章节将介绍如何为终端选择合适的工作模式。

3 工作模式的识别及迁移

3.1 主动设置工作模式

应用程序可以通过注册表项“persist.sys.battery.type”主动设定工作模式，重启后生效

- persist.sys.battery.type = 0: 表示终端自动识别工作模式;
- persist.sys.battery.type = 1: 设定工作模式为移动模式;
- persist.sys.battery.type = 2: 设定工作模式为桌面模式。

如果客户能够明确终端应用场景，可以设置该注册表项为“1”或“2”，这样系统不再改变应用的设定模式。例如“POS 终端+售卖机”的场景，由于这种场景下 POS 终端会始终插着外电，可设置其工作模式为桌面模式。而移动办公的 POS 终端在充满电后，主要依靠电池供电，可设置为移动模式。

相反，如果应用场景不明确，可以设定该值为 0，由 Prolin 根据特定的算法自动识别工作模式或者自动改变工作模式。

接下来我们详细说明提及的自动识别算法。

3.2 自动识别工作模式

自动识别工作模式是指根据终端用户的使用习惯，比如统计近一个星期的供电类型记录，预测接下来的应用场景，切换相应的电池充电模式。

1. 终端开机后默认为 Mobile Mode。
2. 电源管理服务线程 A 负责使用数据库记录一个星期内的供电类型和时间，每隔 10s 秒记录一次。为了避免长时间开机引起数据库过大，终端每 24 小时会清理数据，删除 7 天前的数据。支持通过以下两种方式清空该数据库文件，都是重启后生效：
 - 选择 TM->“System Config”->“Reset BMS DB”菜单，按确认键清空；
 - 应用设置注册表项"persist.sys.bms.db"的值为"1"。
3. 电源管理服务线程 B 动态切换电池工作模式，每隔 30 分钟查询一次当前时间 +1~+(1+n)h 时间段的历史数据。不同机型，n 的取值可能不同，主要取决于终端续航时间，如 S920 的 n 取值为 3 小时，D190、D195、QR68、Q92 的 n 取值为 5 小时。当样本数据至少有 n 小时的数据量，且外电供电样本大于电池供电样本时，切换到 Counter-top Mode；否则，切换到 Mobile Mode。
4. 切换到 Counter-top Mode 或者 Mobile Mode 后，根据终端对应的两种模式表格，设置满充电量和回充电量。

模式迁移可以使用如下的流程图表示（K=n，PA 表示外电供电计数，PB 表示电池供电计数）：

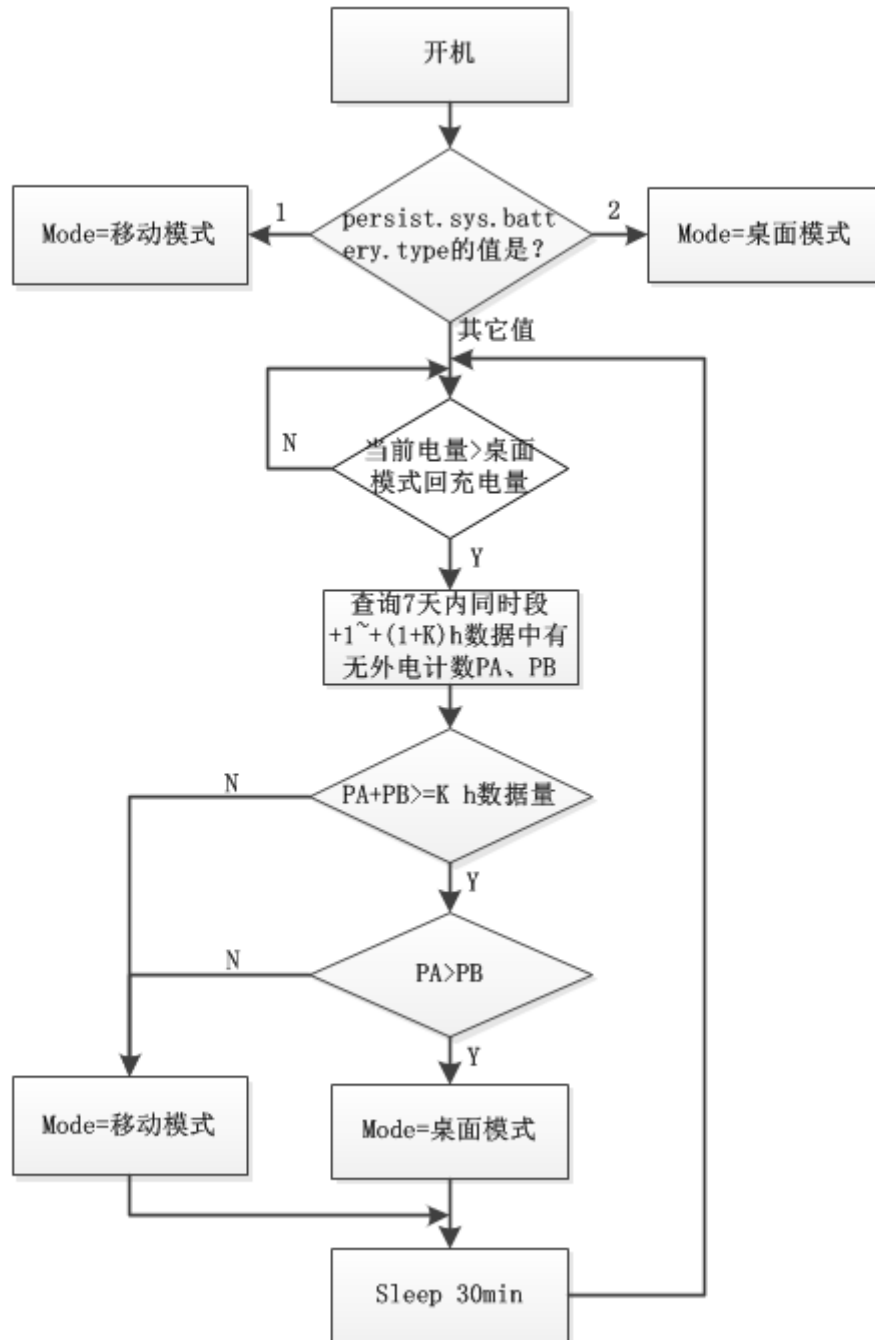


图 3.1 模式迁移流程图

写数据库的相关操作可用下图表示：

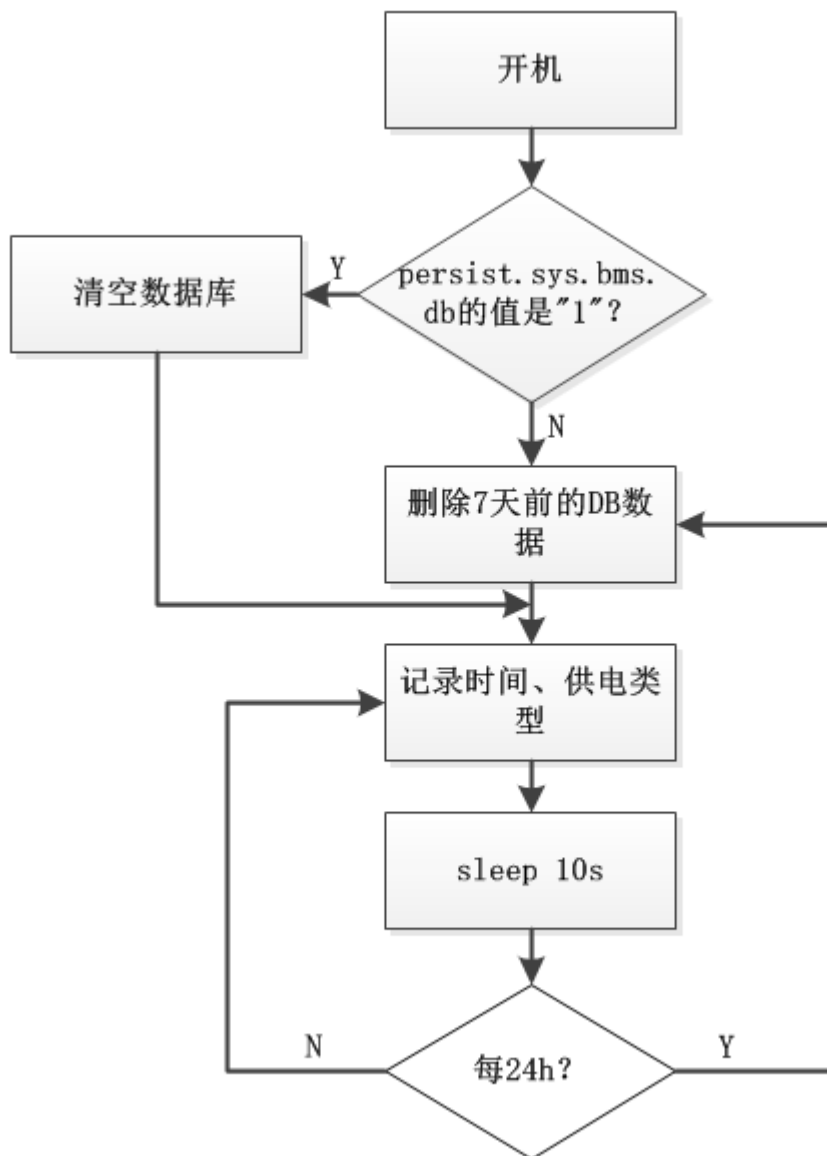


图 3.2 写数据库操作

4电量检测及显示

4.1 获取电池电量

Prolin 提供了 OsCheckBattery()这个 API 供应用程序获取电池的电量值，该 API 的说明及用法如下：

| | | |
|----|---------------------------------|---|
| 原型 | intOsCheckBattery(void); | |
| 功能 | 检测电池电量。 | |
| 参数 | 无 | |
| 返回 | BATTERY_LEVEL_0 | 电量 0~5%，电源电量低，需立即充电。建议不要进行交易、打印、无线通讯等操作。当电池电量过低，系统会自动关机 |
| | BATTERY_LEVEL_1 | 电量 5%~15% |
| | BATTERY_LEVEL_2 | 电量 15%~40% |
| | BATTERY_LEVEL_3 | 电量 40%~70% |
| | BATTERY_LEVEL_4 | 电量 70%~100% |
| | BATTERY_LEVEL_CHARGE | 电池正在充电 |
| | BATTERY_LEVEL_COMPLETE | 电池充满，外电供电 |
| | BATTERY_LEVEL_ABSENT | 电池不存在，外电供电 |
| | ERR_SYS_NOT_SUPPORT | 系统不支持检测电量， |

| | |
|----|--|
| | S800/S300/P ^x 7/P ^x 5 返回该值。 |
| 用法 | <ol style="list-style-type: none"> 1. 外部电源供电时，只能获取电池是否充满，但无法检测电池电量，只有在内置电池供电时，才能检测电量； 2. RF 寻卡、无线模块附着网络和打印过程中，建议不要调用此函数，函数的返回剩余电量会出现抖动； 3. 当返回值为 BATTERY_LEVEL_0 时，无线模块、打印机和 RF 模块可能无法正常工作，此时需要对电池进行充电。 |

4.2 电量显示

Prolin 使用/sys/class/power_supply/battery/capacity 节点表示真实电量值。在充放电参数章节我们给出了两个工作模式下的电量区间定义，桌面模式下，因为做了限制，所以电池真实电量并不能达到 100%。用户可以根据自身需求和使用场景需要，考虑在真实电量与显示电量之间做一个映射。

电量映射示例

```
#include <stdio.h>
#include <string.h>
#include <osal.h>
int get_capacity(void)
{
    int mode;
    int capacity;
    int fullcharge;
    int recharge;
    int ret;
    ret = OsCheckBMSMode(&mode, &capacity, &fullcharge, &recharge);
    if (ret == 0) {
        ret = capacity > recharge ? 100 : (capacity * 100 / recharge);
    }
    return ret;
}
int main(int argc, char *argv[])
{
    int display_capacity;
    display_capacity = get_capacity();
    if (display_capacity < 0) {
        /* Error handling */
    }
}
```

```
printf("capacity:%d\n", display_capacity);  
return 0;  
}
```


5 Boot 逻辑

5.1 为什么要上电开机

自从 S920 支持 BMS 方案后，Prolin 会根据自动识别或者用户应用程序设定的工作模式对电池的电量作出限制，由此，Prolin 必须处于运行状态，才能发挥上述功能。如果终端处于关机状态并且连接外部电源进行充电，充电的过程还是由硬件电路控制，达不到上述设计目的。

因此我们需要当终端连接外部电源的时候，终端能够处于开机状态或者自动开机的状态，需要对原来的 boot 进行升级，一旦接通外部电源，boot 立即启动，从而保证 Prolin 能够运行起来。

5.2 开关机逻辑说明

- 当接入外电的时候，机器自动开机；
- 当有外电供电的情况下，调用关机 API 或者按关机键关机，终端不会关机而是重启；
- 终端需要在电池供电的情况下才能关机。

6 异常检测

为进一步增强终端安全性，我们同时加入了对充放电相关的异常检测机制，一旦发生异常发生，应用程序可以通过 Prolin 提供的接口捕获异常并作相应的处理。

接下来，我们先给出一些异常的定义，再说明如何捕获这些异常。

6.1 异常定义

- 充电 IC 异常
充电 IC 的驱动可以周期性的判断相关硬件管脚的电平特性，当连续 30 秒检测到存在电平特性异常时，Prolin 持续抛出充电 IC 异常信息，直至重启机器。
- 电压过高异常
电源管理进程每 2 秒读取当前电池电压，30 次取均值。每 2 分钟的异常检测读取该 1 分钟内的电压均值，超过保护值则抛出电压异常信息。
理论上，电池电压超过充电限制电压 4.25V 时则需要强制停止充电，但考虑到检测电路存在大约 3% 的误差，因此，上述保护值定义为 4.37V。
- 电池充电异常
S920 的充电芯片具有电池异常检测功能，驱动每 2 秒进行一次电池异常检测，当电池存在且电池 NTC 异常或者电池充电超时，则认为是异常现象，如果连续 60 秒保持在一定误差内，则抛出电池异常信息，后续不再重新检测，除非重启机器。
- 电池损坏
电池过度放电，电压低于 2.5V 或 2V 而导致的电池损坏。开机后，驱动会检测特定寄存器的标志，如果标志为复位，则持续上报异常，直至更换电池并重启机器。

- 充电周期超过限制
电池充电周期超过 1000 次，Prolin 持续抛出异常。
- 电池老化
电池健康状态 (SOH, State Of Health) 通过跟踪电池内部阻抗增量来表示老化程度，数值为 0-100%，数值越大越健康。当电池使用一段时间后，电池容量降低至原来的 60% 时，Prolin 将会抛出异常信息。

6.2 捕获异常

应用程序应该设定合适的机制，以便当 Prolin 抛出异常信息的时候，应用程序可以及时的捕获异常并作相应的处理。

由此，Prolin 在原来的 PM_MSG_T 结构中加入了异常消息域，新的结构体定义如下所示：

结构定义

```
typedef enum {  
    PM_MSG_NO_EVENT,          /* No event. */  
    PM_MSG_ENTER_SLEEP,       /* Device enters sleep mode. */  
    PM_MSG_EXIT_SLEEP,        /* Device exits sleep mode. */  
    PM_MSG_ENTER_SCREENSAVER, /* Device enters screensaver  
mode. */  
    PM_MSG_EXIT_SCREENSAVER, /* Device exits screensaver  
mode. */  
    PM_MSG_ENTER_POWEROFF,    /* Device starts to power off */  
    PM_MSG_POWER_ABNORMAL,    /* Device is powered abnormally */  
    PM_MSG_BATTERY_DAMAGE,    /* Device battery is damaged */  
} PM_MSG_T;
```

应用程序可以通过灵活使用 OsPmGetEvent() 以及 OsCheckPowerStatus() 来捕获异常并作相应的处理。以下是 OsCheckPowerStatus() 的说明以及示例代码：

| 原型 | intOsCheckPowerStatus(int *PowerStatus); | |
|----|---|---|
| 功能 | 检测供电状态是否异常。 | |
| 参数 | PowerStatus 【输出】 | 供电状态参数 Bit0: 电池电压状态, 0-正常, 1-异常; Bit1: 电池充电状态, 0-正常, 1-异常; Bit2: 充电 IC 状态, 0-正常, 1-异常; Bit3: 电池是否损坏, 0-否, 1-是; Bit4: 充电周期是否超过限制, 0-否, 1-是; |

| | | |
|----|---|---|
| | | Bit5: 电池健康状态 SOH, 0-健康, 1-老化严重; Bit6~Bit31 保留。 |
| 返回 | RET_OK | 成功 |
| | ERR_SYS_NOT_SUPPORT | 设备不支持 |
| | ERR_INVALID_PARAM | 非法参数 |
| 用法 | 1. 该功能仅支持部分带电池的终端, 如 S920、D190、D195; 2. 因机型和硬件差异, 系统不一定能检测上述列表的所有异常类型, 如 D190 V06 以上的版本只支持检测电池电压、电池充电等状态是否异常, 其他 D190 版本和 D195 仅支持检测电压异常; S920 V20 可以检测 Bit0-Bit5 的异常, 其余带电量计 S920 可以检测 Bit0-Bit2 的异常类型。 | |



出现 Bit3 电池损坏异常时, 系统会周期性广播 PM_MSG_BATTERY_DAMAGE 事件, 并停止充电, 电池电压控制在 3.6V 内。请务必更换电池。其余异常会广播 PM_MSG_POWER_ABNORMAL 事件。

示例代码

```
PM_MSG_T msg;
int status;
msg = OsPmGetEvent(0);
while (msg != PM_MSG_POWER_ABNORMAL)
{
    msg = OsPmGetEvent(200);
    usleep(100000);
}
OsCheckPowerStatus(&status);
if (status & (1 << 0)) {
    printf("Battery voltage is abnormal.\n");
}
if (status & (1 << 1)) {
    printf("Battery charging is abnormal.\n");
}
if (status & (1 << 2)) {
```

```
printf("Charging IC is abnormal.\n");
}
```

6.3 异常保护

一旦 Prolin 抛出“[异常定义](#)”章节中前四种异常中的任意一个，系统会停止充电。一些可能导致异常的因素和说明如下：

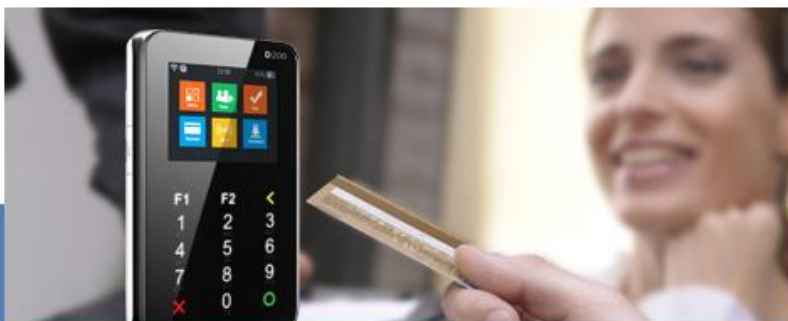
| 异常类型 | 可能导致因素 | 说明 |
|----------|----------------|--|
| 充电 IC 异常 | 本身损坏；误报 | 若重启机器后异常信息消失，则可能是误报，和充电 IC 引脚的状态变化延迟和消抖有关；否则可能是充电 IC 本身已经损坏，建议停止使用，返厂维修。 |
| 电压过高异常 | 充电 IC 或电量计损坏严重 | 建议停止使用终端，返厂维修。 |
| 电池充电异常 | 电池过放等 | 长时间充不满电或充不进电。建议更换新的电池，重启终端；低电量时及时充电。避免电池过放。 |
| 电池损坏 | 电池过度放电等 | 电池充电异常细化的一种，电池电压低于某个值导致不能开机。此时充电一段时间可开机，但由于内部已损坏，请务必更换电池，否则可能会发生电池事故。 |
| 充电周期超过限制 | - | 建议更换电池。 |
| 电池老化 | - | 建议更换电池。 |

7 供电记录

BMS 记录开机时间（标记 start）、插拔外电时间（分别简写 A、B）、关机时间（标记 shutdown）、异常信息（标记 excep），保存在/data/public/battery/battery_power.log。插拔外电的记录频率为 1 分钟。

S920 V20 开机时会检测电池 ID 的变化，并将其记录在 /data/public/battery/battery_info.log 中。用户可通过 U 盘在“TM->系统报告->电池报告菜单”中导出以上记录文件。

Prolin 电池管理系统说明



Your Payment Partner of Choice
www.pax.com.cn

深圳总部

深圳市高新区科技中二路软件园三号楼四层

电话：+86-755-88169630

传真：+86-755-86169634

邮编：518057

香港

香港湾仔港湾道30号新鸿基中心24楼2416室

电话：+852-25888800

传真：+852-28023300 / 22951800