

## 近期处理问题总结

**Author:**李东汉 **Email:**ldh@roch-chips.com

### 目录

◆ 天启：休眠状态下闹铃时间到唤不醒.....	1
(一) 客户问题描述.....	1
(二) 分析：.....	1
(三) 总结.....	1
◆ 金百盛：日历 APK 提醒有时有声音有时没声音.....	1
(一) 客户问题描述.....	1
(二) 分析.....	1
(三) 总结.....	2
◆ 京华：只有两个按键，修改进入 recovery 和正常进行操作.....	3
(一) 客户问题描述.....	3
(二) 分析.....	3
(三) 总结.....	4
◆ 10.1 寸高清屏（1366X768），对比同是 10.1 寸 IPS 屏(1280X800)，体验效果及反应速度要差.....	4
(二) 分析.....	4
(三) 总结.....	5
◆ 秉祥:在设备锁屏界面，不弹出“打开存储设备”提示界面.....	5
(一) 客户问题描述.....	5
(二) 分析.....	5
(三) 总结.....	6

## ◆ 天启：休眠状态下闹铃时间到唤不醒

### (一) 客户问题描述

2928 机器，RTC 型号：HYM8563。设定好闹钟时间后，休眠，时间到时机器没有被自动唤醒。客户怀疑是 RTC 的中断没有来导致

### (二) 分析：

先让客户协助做以下验证：

- 1>.确认 RTC 中断的上拉电阻焊上
- 2>.在中断处理函数处增加打印 LOG
- 3>.中断 IO 口是否有复用

客户确认说均没有问题，机器中断还是不正常。硬件分析后，直接将中断口拉线接地模拟中断也没有作用，这时只有一种情况，就是该中断 GPIO 口有问题，但客户说单独控制该 GPIO 口时高低电平都没有问题。换了另外一个 GPIO 做中断输入口，验证没有问题。

后面让客户再查软件，发现原本的 GPIO 口确实有复用了，被设置成了输出口。

### (三) 总结

客户描述得很玄乎的问题其实大都是其基本的分析步骤没有做好，导致问题前后线索接不上，变成看似无解的问题。

客户问题到公司后其基本的分析步骤最好再认真验证上一遍。

## ◆ 金百盛：日历 APK 提醒有时有声音有时没声音

### (一) 客户问题描述

“日历”应用程序，添加“新建活动”设置声音和振动的提醒功能，时间一到只会在状态栏“无声音和无振动”的提示，有时则是“有声音和有振动”的提示，无规律性。并且 RK 所有平台的 4.0,4.1 系统都存在此问题

### (二) 分析

我先随便借了台 3066/4.1 的机器进行验证，大概验证有 1 个多钟头，并没有发现什么问题。让客户协助确认两个事：

- 1).该机器是否有震动马达
- 2).问题重现的具体步骤

第一个问题没回复，第二个问题回了比较详细的步骤，我让测试组按照该步骤进行测试后发现如下规律：

	详细步骤	概率
情况一	如果当前已过时间在设置的活动总时间的 25%以内，会出现提示音 1. 当前时间为 8:00 至 8:15 内任意时间 2. 新建活动，活动时间从 8:00 到 9:00 3. 完成，过一会通知栏弹出活动信息，有提示音	100%
情况二	如果当前已过时间超过设置的活动总时间的 25%，则不会出现提示音 1. 当前时间为 8:15 至 9:00 内任意时间 2. 新建活动，活动时间从 8:00 至 9:00 3. 完成，过一会通知栏弹出活动信息，没有提示音	100%

测试组提供的这个规律相当重要，至少说明了客户反映的“时有时无”的无规律性是不成立的。

比较正常有声音的提醒和无声音的提醒后发现，正常的 Log 每次都会有这个打印信息而不正常无声音的提醒则没有：

```
D/AlertService(20664): Posting individual alarm notification, eventId:36, notificationId:1, quiet, high-priority
D/AlertService(20664): No low priority events, canceling the digest notification.
```

根据这个线索，在代码前后增加一些打印信息，首先弄明白为什么无声音的提醒没有进入这里。再结合代码，问题终于能水落石出：

阅读代码，发现是由于事件优先级的划分导致，

一般将提醒的优先级分为高优先级、中优先级和低优先级。

优先级的划分依据如下代码：

```
if (beginTimeAdjustedForAllDay > highPriorityCutoff) {
// High priority = future events or events that just started
highPriorityEvents.add(newInfo);
} else if (allDay && tz != null && DateUtils.isToday(beginTimeAdjustedForAllDay)) {
// Medium priority = in progress all day events
mediumPriorityEvents.add(newInfo);
} else {
lowPriorityEvents.add(newInfo);
}
```

即未来或当前发生的事为最高优先级，全天提醒为中优先级，其他为低优先级。

高 优先级会有声音震动等提示，中优先级会有声音震动等提示，而低优先级的提醒事件不会有。详细见 AlertService.java 的方法 public static boolean generateAlerts(Context context, NotificationMgr nm,

SharedPreferences prefs, Cursor alertCursor, final long currentTime, final int maxNotifications)

### （三）总结

最后得出结论，该问题是正常现象，也是符合常理的，为了更进一步验证，拿来谷歌的亲儿子 Nexus7 验证，也是同样的现象。

其实每个问题都有规律性，随机 BUG 也有必现的规律，看我们会不会找。

## ◆ 京华：只有两个按键，修改进入 **recovery** 和正常进行操作

### （一）客户问题描述

2928 机器，只有两个按键，一个是 HOME 键，一个是 POWER 键。  
进入 recovery 界面后没法操作

### （二）分析

发了胡阳辉前段时间写的 recovery 文档《RK292X\_recover 模式 1.doc》给客户自己修改。除了反映编译不到的问题，客户也很快根据这个文档修改进入了 recovery 界面。

剩下的问题其实也很简单，之前是音量加减键作为 recovery 菜单的上下选择键，我让客户改为 HOME 键作为菜单的移动，没有加减就做成一个键循环。后面改了差不多一天没改好，说菜单循环有问题，看了他的修改如下：

#### 1) .按键定义修改

```
device\rockchip\rk2928sdk\recovery\recovery_ui.cpp
int HandleMenuKey(int key_code, int visible) {
    if (visible) {
        switch (key_code) {
            case KEY_DOWN:
            case KEY_VOLUMEDOWN:
            case KEY_HOME: //仅仅增加这个定义
                return kHighlightDown;

            case KEY_UP:
            case KEY_VOLUMEUP:
                return kHighlightUp;

            case KEY_ENTER:
            case KEY_POWER:
                return kInvokeItem;
        }
    }

    return kNoAction;
}
```

#### 2) .菜单单按键循环修改

```
bootable\recovery\recovery.cpp
get_menu_selection(const char* const * headers, const char* const * items,
                    int menu_only, int initial_selection, Device* device)
```

```

.....
switch (action) {
.....

case Device::kHighlightDown:

    ++selected;

selected = ui->SelectMenu(selected);
LOGI("selected=%d\n",selected);
break;

case Device::kHighlightDown:

if(selected == initial_selection+8)
{
    Selected = initial_selection;
}
Else
{
    ++selected;
}

selected = ui->SelectMenu(selected);
LOGI("selected=%d\n",selected);
break;

```

正确的修改:

```

case Device::kHighlightDown:
    selected =(selected+1)%8;
    LOGI("selected = %d,initial_selection=%d\n",selected,initial_selection);
    selected = ui->SelectMenu(selected);
    break;

```

### （三）总结

程序语言基本的算法一定要掌握

## ◆ 10.1 寸高清屏（1366X768），对比同是 10.1 寸 IPS 屏(1280X800)，体验效果及反应速度要差

### （一）客户问题描述

目前正在调试的 M1009 项目，使用 10.1 寸 1366X768 高清显示屏，与 MX01B 项目，使用 10.1 寸 1280x800 IPS 显示屏，做了对比测试，前者 M1009 比后者 MX01，在系统反应速度及系统图像界面的左右滑动效果要差，尤其是界面的左右滑动，M1009 可以明显看到滑动较缓慢，并且图标闪烁着出来

### （二）分析

1>.测试机器的帧率 fps

#su

```
#setprop debug.sf.fps 1
# stop
# start
#logcat
Fps =29 左右，属于偏低
```

2>.试着修改 performance\_info.xml 中的  
<app package="launcher" mode = "2"/>将 2 改为 4，此为硬件加速应用程序和加速等级表  
但修改这里后也没有什么明显的变化

3>.请教闫孝军后其给出方案，在 LCD 驱动中将屏幕的宽从 1366 改成 1368  
测试后发现，帧率提升了一倍，客户接受

### （三）总结

因为 GPU 是 32Byte 对齐，每个像素点有 4 个字节，1366\*4 不是 32Byte 对齐，改成 1368\*4 后正好 32Byte 对齐。这样 GPU 的效率相当于提高了一倍。

由此得知，内存对齐非常重要：

如 NandFlash 强制要求要 4M 对齐，一是为了兼容性更好，二是为了提高数据的存取效率。

1366 和 1368 一样，每次 GPU 都要取 171 次数据，但是 1366 宽度的屏，每一行数据的读取，GPU 还得需要做额外的动作，即剔除掉多余的数据。而 1368 宽度的则不需要做这个额外的动作。效率高很多

比较详细的内存对齐说明请查阅：<http://blog.csdn.net/lgouc/article/details/8235471>

## ◆ 秉祥:在设备锁屏界面，不弹出“打开存储设备”提示界面

### （一）客户问题描述

有部分客户在软件需求中提到，设备在锁屏状态下，插入 USB 线，系统会自动弹出“打开存储设备”界面，点击打开，可以直接存储设备里面的文件，为了增加平板的安全性，我们的平台能不能做到在锁屏界面，插入 USB 线，不自动弹出“打开存储设备”界面，待解锁时再弹出。

### （二）分析

实现客户描述的功能只需要做两件事：

1>.找到插入 USB 时触发界面弹出的地方

2>.判断是否在锁屏界面的条件

第一个问题由于 USB 通知属于状态栏，所以其代码位置应该是在 frameworks\base\packages\SystemUI\ 里，更详细点是在 frameworks\base\packages\SystemUI\src\com\android\systemui\usb\StorageNotification.java 中，找到 POP\_UMS\_ACTIVITY\_ON\_CONNECT 及其使用的地方：

```
if (POP_UMS_ACTIVITY_ON_CONNECT && !adbOn && usbConnected )
{
    .....
}

final int notificationId = mUsbStorageNotification.icon;
if (visible) {
```

```

        notificationManager.notify(notificationId, mUsbStorageNotification);
    } else {
        notificationManager.cancel(notificationId);
    }
}

```

结合锁屏代码，其判断是否在锁屏界面的函数为：

```

public static final boolean isScreenLocked(Context c) {
    android.app.KeyguardManager mKeyguardManager = (KeyguardManager)
        c.getSystemService(c.KEYGUARD_SERVICE);
    Slog.d(TAG, "mKeyguardManager.inKeyguardRestrictedInputMode()="+mKeyguardManager.inKeyguardRestrictedInputMode());
    return mKeyguardManager.inKeyguardRestrictedInputMode();
}

```

返回真则为锁屏界面，否则其他界面。所以只要将上面的代码增加条件即可：

```

if (POP_UMS_ACTIVITY_ON_CONNECT && !adbOn && usbConnected &&
    (!isScreenLocked(mContext)))
{
    .....
}

final int notificationId = mUsbStorageNotification.icon;
if (visible && !isScreenLocked(mContext)) {
    notificationManager.notify(notificationId, mUsbStorageNotification);
} else {
    notificationManager.cancel(notificationId);
}

```

### （三）总结

增加，修改代码前一定要先读懂源码。