



## Android 定制化修改 使用指南

文档版本 05

发布日期 2016-03-02

**版权所有 © 深圳市海思半导体有限公司 2016。保留一切权利。**

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

## **商标声明**



**HISILICON**、海思和其他海思商标均为深圳市海思半导体有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

## **注意**

您购买的产品、服务或特性等应受海思公司商业合同和条款的约束，本文档中描述的全部或部分产品、服务或特性可能不在您的购买或使用范围之内。除非合同另有约定，海思公司对本文档内容不做任何明示或默示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

## **深圳市海思半导体有限公司**

地址：深圳市龙岗区坂田华为基地华为总部

邮编：518129

网址：<http://www.hisilicon.com>

客户服务邮箱：[support@hisilicon.com](mailto:support@hisilicon.com)



# 前 言

## 概述

本文档主要介绍 Android 解决方案一些功能的使用、调试方法，通过实例介绍各模块的如何使用，调试及注意事项。

## 产品版本

与本文档相对应的产品版本如下。

产品名称	产品版本
Hi3798C	V1XX
Hi3798C	V2XX
Hi3796C	V1XX
Hi3798M	V1XX
Hi3796M	V1XX
HiSTBAndroid	V600R001
HiSTBAndroid	V600R002

## 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



## 作者信息

章节号	章节名称	作者信息
全文	全文	W00185899

## 修订记录

修订记录累积了每次文档更新的说明。最新版本的文档包含以前所有文档版本的更新内容。

修订日期	版本	修订说明
2014-08-11	00B01	第 1 次临时发布。
2014-09-18	00B02	新增 1.6 章节“开机扫描 APK 优化”和 4.9 章节“设置中默认参数配置”；修改文档名称，将“Android 解决方案新增特性 使用指南”改为“Android 定制化修改 使用指南”。
2014-10-31	01	第一次正式发布，评审后全面修改。
2015-03-05	02	修改 1.1、1.2 章节 PRODUCT_TARGET 的相关描述。
2015-06-25	03	新增 1.2 和 1.3 章节。
2016-01-22	04	新增 1.7 章节。
2016-03-02	05	新增支持 HiSTBAndroidV600R002。



# 目 录

前 言.....	iii
1 各产品形态定制化使用指南 .....	1-1
1.1 OTT 版本编译开关 .....	1-1
1.2 电信/联通版本编译开关 .....	1-2
1.2.1 版本编译 .....	1-2
1.2.2 网络设置功能—>增加 IPV6 设置界面安全 L2 编译配置 .....	1-3
1.2.3 ServiceSettings 编译 .....	1-3
1.2.4 Android 系统参数修改 .....	1-4
1.2.5 HiRMService 支持第三方 DHCP 待机服务的修改 .....	1-4
1.2.6 软件安装接口（PackageInstaller）屏蔽方法 .....	1-4
1.2.7 recovery 升级脚本配置 .....	1-5
1.2.8 待机功能切换开关 .....	1-5
1.2.9 电信适配层视频播放比例选择开关 .....	1-5
1.3 移动视频基地编译开关 .....	1-6
1.3.1 版本编译 .....	1-6
1.3.2 非华数客户端适配说明 .....	1-6
1.4 媒体扫描环境变量 .....	1-7
1.5 默认存储设备环境变量 .....	1-7
1.6 开机扫描 APK 优化 .....	1-8
1.7 BGA 和 QFP 统一升级包在线升级功能 .....	1-9
1.7.1 版本编译 .....	1-9
1.7.2 现网升级说明 .....	1-11
2 系统更新 .....	2-1
2.1 生成 OTA 升级包 .....	2-1
2.2 什么情况下会使用到变分区升级 .....	2-1
2.3 更改 reset 按键的 GPIO 接口 .....	2-2
2.4 Recovery 过程闪灯提示 .....	2-3
2.5 在线升级 .....	2-4
2.5.1 在线强制升级 .....	2-4
2.5.2 在线非强制升级 .....	2-6



2.6 在线更新开机 Logo、fastplay、开机动画 .....	2-10
2.6.1 更新开机 LOGO .....	2-10
2.6.2 更新 Fastplay .....	2-11
2.6.3 更新开机动画.....	2-12
<b>3 生产及调试工具.....</b>	<b>3-1</b>
3.1 厂测工具.....	3-1
3.2 指标测试工具.....	3-3
3.2.1 HDMITest .....	3-3
3.2.2 CVBSTest .....	3-4
3.2.3 SataAndUsbTest.....	3-5
3.3 远程调试工具.....	3-6
3.3.1 远程调试工具收件人配置.....	3-6
3.3.2 远程调试工具使用.....	3-7
<b>4 其他.....</b>	<b>4-1</b>
4.1 电源指示灯 .....	4-1
4.1.1 配置单色电源指示灯 .....	4-1
4.1.2 配置双色电源指示灯 .....	4-1
4.2 开机向导.....	4-2
4.3 如何支持一般的 USB 手柄 .....	4-3
4.4 多屏统一开关.....	4-3
解决办法 .....	4-4
4.5 如何修改开机向导中的二维码下载地址 .....	4-4
4.6 设置中默认参数配置.....	4-5



## 插图目录

图 1-1 开启选择默认存储设备功能.....	1-9
图 1-2 BGA 和 QFP 统一升级包在线升级流程.....	1-10
图 2-1 系统更新列表信息 1.....	2-5
图 2-2 系统更新列表信息 2.....	2-6
图 2-3 打开【HisiliconSetting】.....	2-7
图 2-4 进入【System】.....	2-7
图 2-5 进入【Upgrade Online】.....	2-8
图 2-6 弹出提示框.....	2-8
图 2-7 开始进行下载.....	2-9
图 2-8 下载完毕.....	2-9
图 2-9 Logo 更新列表信息.....	2-11
图 2-10 Fastplay 更新列表信息.....	2-12
图 2-11 BootAnimation 更新列表信息.....	2-13
图 3-1 厂测工具界面.....	3-2
图 3-2 HDMI 认证指定的码流文件.....	3-4
图 3-3 弹出测试项列表.....	3-4
图 3-4 选择码流和制式.....	3-5
图 3-5 测试项界面.....	3-6
图 3-6 mail.properties 界面.....	3-7
图 3-7 错误报告界面.....	3-7
图 4-1 配置单色电源指示灯.....	4-1
图 4-2 通用设置界面.....	4-4
图 4-3 修改字符串.....	4-4
图 4-4 修改代码.....	4-5
图 4-5 XX 应用停止运行.....	4-7



图 4-6 XX 进程停止运行 .....	4-7
-----------------------	-----

# 1 各产品形态定制化使用指南

HiSTBAndroidV600R001C00SPC060 及以后版本可以通过配置开关的方式实现同一份源码同时支持 OTT、电信、联通、移动视频基地及移动分省模式的招标和测试。客户只需要更改对应的配置开关即可完成切换。

## 1.1 OTT 版本编译开关

PRODUCT\_TARGET=demo 编译开关开启后主要配置如下内容：

- 删除系统默认 Launcher **【Launcher2】**——保证系统中只存在一个 Launcher
- 删除系统原生 Development 工具 **【Development】**——移除开发过程中使用的工具类 APK
- 删除 OTA(Over-the-Air Technology)升级 Demo **【HiOTAUpgrade】**——移除开发过程中使用的工具类 APK
- 增加开机向导 **【HiRunInstructions】【HiInstructions】**——第一次开机时介绍盒子的使用方法
- 删除文件管理器 Demo **【HiFileManager】**——保留媒体中心，删除功能相同的应用
- 删除厂测工具 Demo **【HiFactoryTest】**——移除开发过程中使用的工具类 APK
- 删除双网卡设置 Demo **【HiDualNetDemo】**——移除开发过程中使用的工具类 APK
- 删除待机模式选择 Demo **【HiSuspendMode】**——移除开发过程中使用的工具类 APK
- 删除 Debug 工具集 **【HiDebugKit】**——移除开发过程中使用的工具类 APK
- 删除转场动画 Demo **【HiAniApp】**——移除开发过程中使用的工具类 APK
- 增加错误日志反馈功能 **【HiErrorReport】**——用于用户反馈使用过程中发现的 BUG
- 删除媒体录制 Demo **【HiMediaRecorderTest】**——移除开发过程中使用的工具类 APK
- 删除老化测试 Demo **【HiAgingTest】**——移除开发过程中使用的工具类 APK
- 删除录音 Demo **【SpeechRecorder】**——移除开发过程中使用的工具类 APK





可以通过修改 device/hisilicon/xxxx/customer.mk 文件中的如下代码，来控制是否开启 OTT 编译开关：

```
PRODUCT_TARGET := demo
```

- 当 PRODUCT\_TARGET := demo 时，上述功能开启；
- 当 SUPPORT\_OTT\_DEMO 为其他值时，上述功能关闭。



说明

PRODUCT\_TARGET 默认值为 ott。

## 1.2 电信/联通版本编译开关

目前各个地区的电信运营商规范还存在不一致的情况，导致了某个特性无法同时满足多个运营商要求的情况，容易引起用户使用版本混乱。

为了客户更加方便的使用海思版本，本文将一些冲突的特性做了说明，用户可以方便的根据不同局点要求来配置特性。

### 1.2.1 版本编译

从 SPC060 版本开始可以通过新增编译控制宏和产品目标属性来区分不同领域的特性：

- 编译宏名称为 PRODUCT\_TARGET，该宏的定义在如下目录的文件中：  
device/hisilicon/Hi37xxxx/customer.mk
- 根据不同的产品属性取值分别为 shcmcc、telecom、unicom、ott、dvb 等，分别代表移动视频基地、电信、联通、消费者、广电等领域，默认值取 ott。

在 customer.mk 中，根据 PRODUCT\_TARGET 宏的取值自动对 ro.product.target 赋值，ro.product.target 取值须和 PRODUCT\_TARGET 值保持一致。

版本中的各模块会根据 ro.product.target 属性值走不同的行为分支。原来的 ro.iptv.enable 属性删除，不再使用。



注意

编译版本时，注意先根据产品目标的领域手动修改 customer.mk 中 PRODUCT\_TARGET 的值，然后再编译。

开启此属性后，默认开启以下功能：

- 本地视频播放—>支持声道，支持遥控器操作，需要使用时按“菜单”键进行选择。
- 本地视频播放—>字幕文件支持自动监测和手动选择，支持遥控器操作，需要使用时按“菜单”键进行选择。
- 图片播放功能—>支持图片文件自动播放的时间间隔设置，支持遥控器操作，在幻灯片播放时按“菜单”键进行选择。



- 图片播放功能—>支持图片显示效果的设置，至少包含缩放，上升，擦除，下降，展开等效果，支持遥控器操作，在幻灯片播放时按“菜单”键进行选择。
- 图片播放功能—>背景音乐（自动循环播放图片同级目录下的音乐文件），支持遥控器操作，在幻灯片播放时按“菜单”键进行选择。
- 文件管理功能—>文件删除，支持遥控器操作，需要使用时按“菜单”键进行选择。
- 文件管理功能—>文件排序，支持遥控器操作，需要使用时按“菜单”键进行选择。
- 文件管理功能—>文件查询（拼音首字母查询），支持遥控器操作，需要使用时按“菜单”键进行选择。
- 音乐播放功能—>支持图片播放功能，用户可配置选择图片列表
- 图片播放功能—>支持背景音乐播放，用户可配置选择背景音乐列表，支持遥控器操作，需要使用时按“菜单”键进行选择。

## 1.2.2 网络设置功能—>增加 IPV6 设置界面安全 L2 编译配置

电信运营商出于运营安全角度的考虑，对终端设备提出了安全启动、安全升级、系统安全等要求。海思将这些功能统称为安全 Level 2,并集成到了版本中。

版本中安全 L2 功能默认关闭，功能配置总开关位于如下文件中：

device/hisilicon/Hi37xxMV100/customer.mk

HISILICON\_SECURITY\_L2 的初始值为 false，即：

```
HISILICON_SECURITY_L2 := false
```

如果需要打开安全 L2 功能，可以将 false 改为 true 即可。

System 业务保护功能是安全 L2 功能的一部分，为了客户集成开发方便，该功能设置了一个独立的控制开关，默认关闭，如果需要打开，则需要修改

device/hisilicon/bigfish/sdk/configs/hi379XXv100 下的对应的  
hi3798mdmo\_hi3798mv100\_android\_cfg.mak 文件中的配置项。

修改：

hi3798mdmo\_hi3798mv100\_android\_cfg.mak 中的  
CFG\_HI\_ANDROID\_SECURITY\_L2\_SYSTEM\_CHECK 配置项为：

```
CFG_HI_ANDROID_SECURITY_L2_SYSTEM_CHECK=y
```

然后再执行编译。

## 1.2.3 ServiceSettings 编译

ServiceSetting 是卓影中间件 AIDL 适配接口，为不影响非电信卓影中间件版本，默认版本不编译；如果电信运营商指定的中间件为卓影则需要编译，需修改

device/hisilicon/bigfish/framework/Andriod.mk 文件，添加 ServiceSettings 模块名称至 driver\_modules 中。如下新增一行代码：

```
driver_modules += ServiceSettings
```



## 1.2.4 Android 系统参数修改

很多运营商对产品的系统参数有要求，比如 `android.os.Build.BOARD`、`android.os.Build.BRAND` 等，`android` 系统本来就支持这些属性，将它们集中定义在 `frameworks/base/core/java/android/os/Build.java` 文件中，客户可以直接在代码中引用。

## 1.2.5 HiRMService 支持第三方 DHCP 待机服务的修改

运营商要求支持 DHCP+功能，此功能涉及与电信头端的身份认证，通常由第三方中间件实现。

电信规范中要求系统待机前需要主动将网络断开，并且保证在待机恢复后，可以正常认证开机。海思的待机方案中，待机前会断开网络，但是在 DHCP+功能使用了第三方中间件提供的 DHCP 客户端的情况下，待机前断开的操作需要第三方自行处理。

因此，这种情况下，待机过程需要将断开网络的操作去掉。具体方法如下，以卓影中间件为例：

HRMservice 需要作如下修改：

- 将卓影 apk 加入白名单，防止被 kill。

```
static final String PACKAGE_SUYINGAPK= "com.android.smart.terminal.iptv"  
static final String  
PACKAGE_SUYINGAIDL="com.android.smart.terminal.iptv.aidl"
```

- 删除待机前关闭网络和唤醒后恢复网络的相关操作。
- 确保卓影 apk 中有主动调用 `dhcp release` 的操作。

版本中 `device/hisilicon/Hi37xxx/customer.mk` 中定义了控制开关，默认为 `false`，海思方案生效，改为 `true` 则卓影方案生效。

```
PRODUCT_PROPERTY_OVERRIDES += \  
  
ro.thirdparty.dhcp=false
```



### 注意

务必注意卓影的 APK 包名是否正确，要保持同步修改。另外，系统中只是针对卓影中间件的场景做了处理，如果适配的是其他中间件的 APK，则需要参考系统代码重新处理。

## 1.2.6 软件安装接口（PackageInstaller）屏蔽方法

电信运营商要求屏蔽 `android` 原生 `PackageInstaller`.APK 软件安装接口，来屏蔽对 APK 的安装，可以按如下操作：

- 修改方法：在系统编译中不编译 `/packages/apps/PackageInstaller` 目录。
- 修改文件：`build/target/product/core_minimal.mk` 去掉编译 `PackageInstaller`。



## 1.2.7 recovery 升级脚本配置

电信运营商要求升级包放入 cache 分区，并且升级过程中不删除 cache 分区，所以需要修改升级脚本，脚本为 bootable/recovery/etc/META-INF/com/google/android/updater-script-emmc，将其中的如下三行代码删掉：

```
ui_print("format cache.....");  
unmount("/cache");  
format("ext4", "EMMC", "/dev/block/platform/hi_mci.1/by-name/cache", "0",  
"/cache");
```

## 1.2.8 待机功能切换开关

不同的电信运营商对待机唤醒的要求是不一样的，例如广东电信、上海电信的要求是浅待机，其他电信的要求则是真待机。海思版本中添加了控制开关可以对待机功能进行选择。在 device/hisilicon/bigfish/build/customer.mk 中定义了变量：

```
PRODUCT_PROPERTY_OVERRIDES += \  
persist.suspend.mode=deep_restart  
smart_suspend表示浅待机  
deep_launcher表示真待机回Launcher方案  
deep_restart待机重启（默认值）  
deep_resume原生待机方案。
```

## 1.2.9 电信适配层视频播放比例选择开关

电信运营商对视频默认输出比例有不同的要求，通过在 device/hisilicon/bigfish/Hi37XXXX/customer.mk 文件中修改 persist.sys.video.cvr 的值来设置视频的输比例：

- 1 表示 4:3 加黑边
- 0 表示 16:9 全屏拉伸

默认形式为 4:3 加黑边。

注意修改正确的产品形态的属性值，如下只需要修改 telecom 和 unicom 控制的属性值。

```
#set video output rate for telecom and unicom, default 4:3  
ifeq ($(strip $(PRODUCT_TARGET)),telecom)  
PRODUCT_PROPERTY_OVERRIDES += \  
    persist.sys.video.cvr= 1  
else ifeq ($(strip $(PRODUCT_TARGET)),unicom)  
PRODUCT_PROPERTY_OVERRIDES += \  
    persist.sys.video.cvr= 1  
else  
PRODUCT_PROPERTY_OVERRIDES += \  
    persist.sys.video.cvr= 0
```



```
endif
```

## 1.3 移动视频基地编译开关

移动市场通过基地模式引导与规范部分省市的 IPTV 业务，海思提供基础交付版本，满足视频基地的测试需求，协助客户通过视频基地测试，保障客户正常出货。

随着移动视频基地规范版本的演进与修订，移动各种定制化需求应运而生，如：卡拉 ok、视频通话等，本文档在基础版本使用指南的基础上，详细描述了移动定制化需求，为客户在定制化需求上的使用与二次开发提供帮助。

### 1.3.1 版本编译

该版本通过新增编译控制宏和产品目标属性来区分不同领域的特性：

- 编译宏名称为 `PRODUCT_TARGET`，定义在在 `device/hisilicon/Hi37xxxx/customer.mk` 中。
- 根据不同的产品属性取值分别为 `shcmcc`、`telecom`、`unicom`、`ott`、`dvb` 等，默认值取 `ott`。其中，视频基地版本宏的属性值为 `shcmcc`。

为满足视频播放画面与图形层同步调整，需要打开宏开关。具体为在 `device/hisilicon/bigfish/sdk/configs/hi3798mv100` 路径下的 `hi3798mdmo_hi3798mv100_android_cfg.mk` 文件中（具体路径与文件根据芯片类型与单板类型会有差异）将：

```
CFG_HI_DISP_VIRT_OFFSET_EFFECT_WHEN_FULLSCREEN is not set
```

修改为：

```
CFG_HI_DISP_VIRT_OFFSET_EFFECT_WHEN_FULLSCREEN=y
```

视频基地的编译步骤为：

1. 手动修改 `customer.mk` 中 `PRODUCT_TARGET` 的值。
2. 修改内核配置文件，设置 `CFG_HI_DISP_VIRT_OFFSET_EFFECT_WHEN_FULLSCREEN` 属性为 `y`。

### 1.3.2 非华数客户端适配说明

视频基地版本中，系统通过属性值 `client.apk.name` 来区分华数客户端与其他客户端，主要原因为媒体部分逻辑处理中，华数客户端与其他客户端有差异，`client.apk.name` 默认值为 `wasu.app`，表示系统默认适配华数客户端。客户若需要适配其他客户端，需要在编译版本之前，修改 `client.apk.name` 的值为一个非“`wasu.app`”的字符串，修改方法为：

修改文件 `device.mk`（路径：`device/hisilicon/Hi37xxxx/`），将

`client.apk.name= wasu.app` 修改为：`client.apk.name= nowasu.app`（示例）

机顶盒串口也可直接修改此属性，修改方法为在串口下：



1. 修改系统为可写权限：`mount -o remount /system/ /system/`
2. Vi 打开编辑属性文件：`busybox vi system/build.prop`
3. 修改 `client.apk.name`，修改方法同上
4. wq (wq!) 保存或强制保存修改，重启系统。

## 1.4 媒体扫描环境变量

`ro.mediaScanner.enable` 环境变量用于标示当前版本是否开启 Android 媒体扫描功能。

可以通过修改 `device/hisilicon/Hi37xxxx/device.mk` 文件中的如下代码，来控制是否开启 Android 媒体扫描功能：

```
# MediaScanner
PRODUCT_PROPERTY_OVERRIDES += \
ro.mediaScanner.enable=false
```

- 当 `ro.mediaScanner.enable=true` 时，上述功能开启；
- 当 `ro.mediaScanner.enable=false` 时，上述功能关闭。

`ro.mediaScanner.enable` 默认值为 `false`。

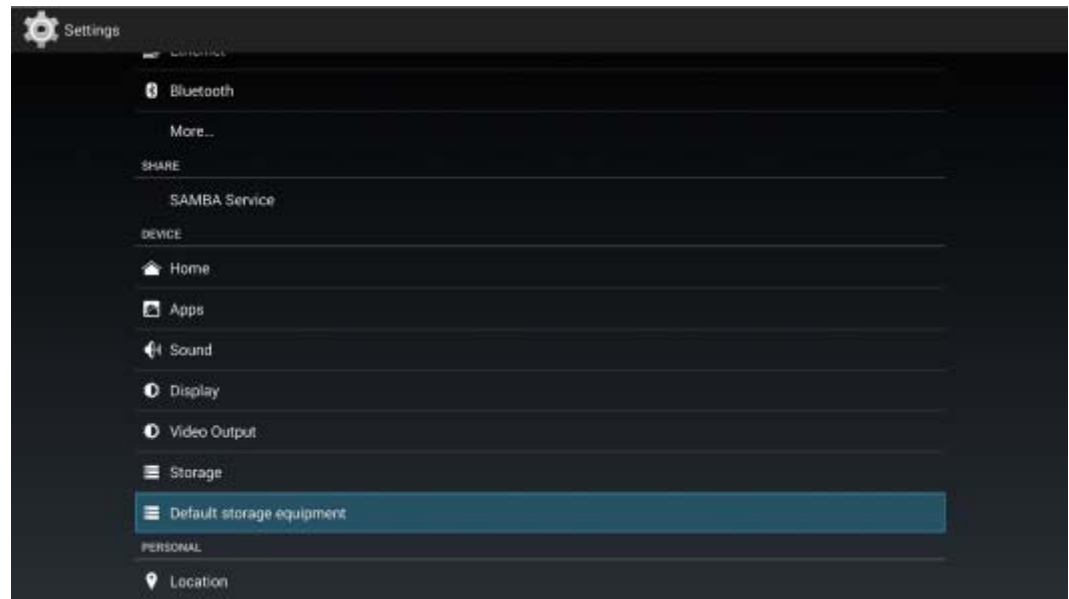
## 1.5 默认存储设备环境变量

`ro.defaultStorage.enable` 环境变量用于标示当前版本是否开启选择默认存储设备功能。

当启动此功能后，Android 系统原生【设置】中将增加【指定默认存储设备】选项。此项功能支持将任意外接 U 盘、移动硬盘等存储设备作为 Android 系统默认存储设备，支持将已安装的 APK 移动到上述指定的设备中，各类应用在内置 SD 卡中创建的相关文件也将存储在上述指定的设备中。如图 1-1 所示。



图1-1 开启选择默认存储设备功能



可以通过修改 device/hisilicon/Hi37xxxx/device.mk 文件中的如下代码，来控制是否开启选择默认存储设备功能：

```
# DEFAULT STORAGE
PRODUCT_PROPERTY_OVERRIDES += \
ro.defaultStorage.enable=false
```

- 当 ro.defaultStorage.enable=true 时，上述功能开启；
- 当 ro.defaultStorage.enable=false 时，上述功能关闭。

说明

ro.defaultStorage.enable 默认值为 false。

## 1.6 开机扫描 APK 优化

ro.laterscan.enable 环境变量用于标示当前版本是否开启开机扫描 APK 优化。

当启动此功能后，Android 系统启动时将不对/data/app 目录下的应用进行扫描，启动完成后再进行扫描。这样可以达到无论安装多少第三方应用，启动速度不变的目的。

可以通过修改 device/hisilicon/Hi37xxxx/device.mk 文件中的如下代码，来控制是否开启开机扫描 APK 优化功能：

```
PRODUCT_PROPERTY_OVERRIDES += \
ro.laterscan.enable=true
```

- 当 ro.laterscan.enable=true 时，上述功能开启；
- 当 ro.laterscan.enable=false 时，上述功能关闭。

ro.laterscan.enable 默认值为 true。





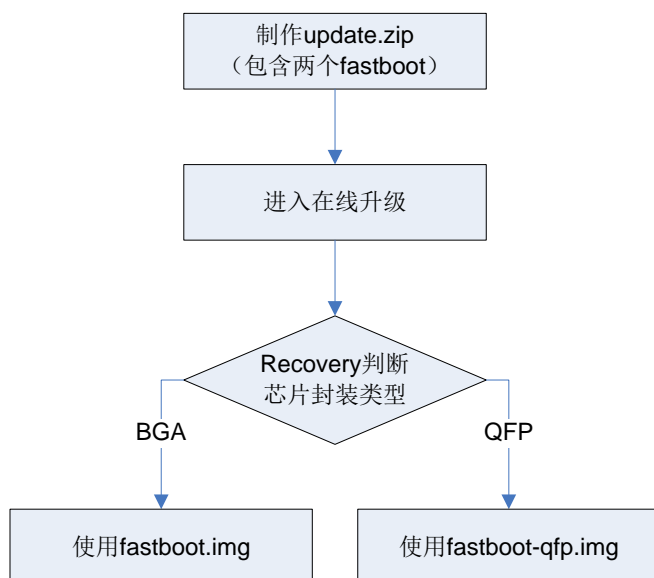
## 1.7 BGA 和 QFP 统一升级包在线升级功能

该特性支持 BGA 和 QFP 单板用同一个 update.zip 升级包。在配置文件中分别给 BGA 和 QFP 指定单板表格，编译时会同时生成两个 boot，一个 BGA 的 fastboot 和一个 QFP 的 fastboot。升级时，recovery 会依据芯片封装类型自动选择 update.zip 中相应的 fastboot。

主要涉及 3 个方面，

- 编译生成 update 包时，update.zip 中包含 2 个 fastboot（fastboot.img 用于 BGA，fastboot-qfp.img 用于 QFP）。
- 修改 update.zip 升级脚本，通过 getprop 区分 fastboot。
- Recovery 区分芯片封装类型，升级对应的 fastboot。

图1-2 BGA 和 QFP 统一升级包在线升级流程



### 注意

该功能目前只适用于 Hi3798MV100 芯片，只支持 EMMC flash 类型。

### 1.7.1 版本编译

新增编译控制宏 SUPPORT\_UNIFIED\_UPDATE，控制统一升级包升级功能是否使能：

- 编译控制宏 SUPPORT\_UNIFIED\_UPDATE 定义在 device/hisilicon/Hi3798MV100/customer.mk 中。
- 当 SUPPORT\_UNIFIED\_UPDATE := true 时，上述功能开启。
  - 非安全版本，out/target/product/Hi3798MV100/Emmc 下编译生成 2 个 fastboot：





- fastboot.bin 用于 BGA
- fastboot-qfp.bin 用于 QFP
- update.zip 中包含 2 个 fastboot:
  - fastboot.img 用于 BGA
  - fastboot-qfp.img 用于 QFP。
- 二级安全方案版本，out/target/product/Hi3798MV100/Security\_L2/MAINTAIN 下编译生成 2 个签名的 fastboot:
  - fastboot.bin 用于 BGA
  - fastboot-qfp.bin 用于 QFP
 update.zip 中包含 2 个 fastboot:
  - fastboot.img 用于 BGA
  - fastboot-qfp.img 用于 QFP
 out/target/product/Hi3798MV100/Security\_L2/PRODUCTION 下编译生成 2 个未签名的 fastboot:
  - fastboot.bin 用于 BGA
  - fastboot-qfp.bin 用于 QFP
 update.zip 与 MAINTAIN 下的相同。
- 当 SUPPORT\_UNIFIED\_UPDATE := false 时，上述功能关闭，编译生成 1 个 fastboot；默认为 false。

新增 QFP EMMC fastboot 配置选项 EMMC\_BOOT\_CFG\_NAME\_2、EMMC\_BOOT\_REG\_NAME\_2:

- EMMC\_BOOT\_CFG\_NAME\_2、EMMC\_BOOT\_REG\_NAME\_2 定义在 device/hisilicon/Hi3798MV100/BoardConfig.mk 中。
- 当 SUPPORT\_UNIFIED\_UPDATE := true 时，此 2 项编译宏有效。
  - EMMC\_BOOT\_CFG\_NAME、EMMC\_BOOT\_REG\_NAME 用于指定 BGA 的 fastboot 配置文件，对于海思公版，可以指定 1a、1b、1c、1d。
  - EMMC\_BOOT\_CFG\_NAME\_2、EMMC\_BOOT\_REG\_NAME\_2 用于指定 QFP 的 fastboot 配置文件，对于海思公版，可以指定 1f、1g。

例如:

```
EMMC_BOOT_CFG_NAME :=
hi3798mdmola_hi3798mv100_ddr3_1gbyte_16bitx2_4layers_emmc.cfg
EMMC_BOOT_REG_NAME :=
hi3798mdmola_hi3798mv100_ddr3_1gbyte_16bitx2_4layers_emmc.reg
EMMC_BOOT_CFG_NAME_2 :=
hi3798mdmolf_hi3798mv100_ddr3_1gbyte_16bitx2_2layers_emmc.cfg
EMMC_BOOT_REG_NAME_2 :=
hi3798mdmolf_hi3798mv100_ddr3_1gbyte_16bitx2_2layers_emmc.reg
```

- EMMC fastboot 配置文件位于 device/hisilicon/bigfish/sdk/source/boot/sysreg 目录。



## 1.7.2 现网升级说明

如果现网盒子所用软件版本不是“`SUPPORT_UNIFIED_UPDATE := true`”情况下编译产生的，而后续在线升级时希望使用 BGA 和 QFP 统一升级包，那么请按照如下步骤操作：

步骤 1 当 `SUPPORT_UNIFIED_UPDATE := true` 时编译 `recovery.img`，制作只升级 `recovery.img` 的 `update.zip`，在线升级。

于是，盒子的 `recovery` 具有依据芯片封装类型自动选择对应 `fastboot` 的功能。

步骤 2 使用 BGA 和 QFP 统一升级包进行在线升级。

----结束



# 2 系统更新

## 2.1 生成 OTA 升级包

### 问题描述

如何生成两个版本之间的 OTA 升级包？

### 解决办法

假设 A 版本的升级包为 update\_A.zip，B 版本的升级包为 update\_B.zip，那么如果要从 A 升级到 B 版本，则 A 与 B 之前的 OTA 升级包制作如下：

```
./device/hisilicon/bigfish/upgrade/ota/ota_from_target_files -i  
update_A.zip update_B.zip ota_A_to_B.zip
```

因此 ota\_A\_to\_B.zip 即为 A 与 B 之前的 OTA 包。

注意事项如下：

- OTA 包更适合于小版本之间，不适合两个大版本之间，因为 OTA 的强项是升级包体积小
- 制作两个版本之前的 OTA 包，只支持两个版本都是 eng 版本，或者 user 版本，而不能支持一个版本为 eng,另外一个版本为 user

## 2.2 什么情况下会使用到变分区升级

### 问题描述

当需要更新用户系统且分区信息有变化时。如何进行变分区升级呢。

### 解决办法

由于论述较多，请参考：《Android 解决方案 使用指南》的“4.2.5 变分区升级”章节。



## 2.3 更改 reset 按键的 GPIO 接口

### 问题描述

按下 reset 按键后上电，系统会强制进入升级模式，在此模式下，系统检查外接 U 盘根目录下是否存在 update.zip 文件，如果存在，用此文件进行升级，如果不存在，提示升级失败。如果 reset 按键的 GPIO 口变更，如何修改呢？

### 解决办法

#### 步骤 1 修改 GPIO 口

打开 `device/hisilicon/bigfish/sdk/source/boot/product/android/recovery.c` 中的 static int `check_gpio_recovery(void)` 函数，根据不同版本，修改相应的寄存器值，具体寄存器值，可咨询硬件相关人员或查阅相应芯片手册（如下红色部分）。

```
static int check_gpio_recovery(void)
{
    int t;
    int count = 2;
    long long chipid;
    chipid = get_chipid();
    printf("get chipid =%llx\n", chipid);
    switch(chipid){
        case _HI3716C_V200:
            printf("get chipType (HI3716C_V200)\n");
#ifdef HI_OTTGPI0_SUPPORT
            /* get gpio5_1 buttom */
            while((((REG_VAL(0xF80043FC) & 0x02)>>1) == 0) && (count > 0)){
                udelay(1000*1000);
                count--;
            }
#endif
            break;
#ifdef defined(CHIP_TYPE_hi3798cv100) || defined(CHIP_TYPE_hi3798cv100_a)
        case _HI3798CV100:
        case _HI3798CV100A:
            printf("get chipType (HI3798C_V100)\n");
            // get GPIO_STB1 buttom
            REG_VAL(0xF8008038) = 0x1 ; // 设置gpio方向寄存器
            while((((REG_VAL(0xF80093FC) & 0x1) == 0) && (count > 0)){
                udelay(1000*1000);
                count--;
            }
            break;
#endif
    }
}
```



```
    }  
    printf("count=%x\n", count);  
    if(count > 0){  
        return HI_FAILURE;  
    }else{  
        return HI_SUCCESS;  
    }  
}
```

## 步骤 2 添加新版本类型

在以上函数 switch 后，添加自己的版本号，`get_chipid()`函数会获取出来，仿照其他版本，添加代码。寄存器参考芯片手册的外设 GPIO 章节。

----结束

Reset 按键主要功能：

- 如果 U 盘根目录下未找到升级包 `update.zip` 文件，或者未插入 U 盘，按键进入升级模式后，会执行恢复出厂设置。（擦除 `cache`、`data`）
- 如果 U 盘根目录下找到升级包 `update.zip` 文件，进入升级版本模式。

## 2.4 Recovery 过程闪灯提示

### 问题描述

如何在升级中添加闪灯流程，从而表明系统正处于升级过程中。

### 解决办法

在升级脚本中：最开始添加 `open_led()`，结尾处添加 `close_led()`。

- `open_led()`：开始闪灯
- `close_led()`：关闭闪灯

参考升级脚本：

```
vim bootable/recovery/etc/META-INF/com/google/android/updater-script-emmc
```

`open_led()`、`close_led()`最终函数实现请参考：`vim bootable/recovery/updater/gpio.c` 中的 `flashled()`。



## 2.5 在线升级

### 2.5.1 在线强制升级

#### 2.5.1.1 设置更新列表访问地址

##### 问题描述

系统启动后，后台服务会自动访问服务器上的系统升级包更新列表，若有更新的升级包，则自动下载并进行更新。如何修改指定更新列表的访问地址呢？

##### 解决办法

修改

device/hisilicon/bigfish/packages/apps/HiRMService/src/com/hisilicon/android/hiRMService/SystemUpgrade.java 文件中的如下代码：

```
private final String INTRA_NET_PATH = "no";
```

当 INTRA\_NET\_PATH = "no" 时，表示无升级包信息访问地址，不进行更新；

将 no 替换成实际地址，例：

```
INTRA_NET_PATH="http://10.67.212.148/upgrade.xml")，
```

则系统会访问 <http://10.67.212.148/upgrade.xml>，并解析更新信息。

INTRA\_NET\_PATH 默认值为 no。

#### 2.5.1.2 配置更新信息

##### 问题描述

系统会访问并解析服务器上的 XML 文件，那么如何在 XML 文件中标识更新信息？

##### 解决办法

创建 XML 文件，其中【CompleteUpgrade】标签是完整升级包的更新信息，包含的子标签有：

- 【MinVersion】：最低固件版本
- 【MaxVersion】：最高固件版本
- 【TargetVersion】：升级后的固件版本
- 【Description】：升级描述
- 【FilePath】：更新包下载路径
- 【ReleaseTime】：发布时间
- 【FileSize】：升级包的大小
- 【ChangePartition】：是否变分区升级
- 【CompulsoryUpgrade】：是否强制升级



通过比较【TargetVersion】中的时间信息（例 20140517.203300），判断是否更新。如图 2-1、图 2-2 所示。

图2-1 系统更新列表信息 1

```
1 <?xml version="1.0" encoding="UTF-8" ?>
2 <HiSTBAndroidUpgrade>
3   <!--Complete upgrade-->
4   <CompleteUpgrade>
5     <!--Support the minimum version-->
6     <MinVersion>
7       Hi3716CV200-eng 4.2.2 JDQ39 eng.wangyang.20130101.203322 test-keys
8     </MinVersion>
9     <!--Support the highest version-->
10    <MaxVersion>
11      Hi3716CV200-eng 4.4.2 JDQ39 eng.wangyang.20140409.203300 test-keys
12    </MaxVersion>
13    <!--The target version-->
14    <TargetVersion>
15      Hi3716CV200-eng 4.4.2 JDQ39 eng.wangyang.20140409.203448 test-keys
16    </TargetVersion>
17    <!--Upgrade Description-->
18    <Description>
19      Solve the following problem:
20      1.Upgrade to Android4.4 version
21      2.Support online upgrade
22    </Description>
23    <!--Upgrade download url-->
24    <FilePath>
25      http://203.195.181.80/e2372aadad7c2b1a/update0411.zip
26    </FilePath>
27    <!--Release time-->
28    <ReleaseTime>
29      2014-04-11
30    </ReleaseTime>
31    <!--Upgrade file size-->
32    <FileSize>
33      270.4MB
34    </FileSize>
35    <!--Whether to change partitions-->
36    <ChangePartition>
37      false
38    </ChangePartition>
39    <!--Whether to forced to upgrade-->
40    <CompulsoryUpgrade>
41      true
42    </CompulsoryUpgrade>
43  </CompleteUpgrade>
44 </HiSTBAndroidUpgrade>
45
46
```



图2-2 系统更新列表信息 2

```
1 <?xml version="1.0" encoding="UTF-8" ?>
2 <HiSTBAndroidUpgrade>
3   <!--OTA upgrade-->
4   <OTAUpgrade>
5     <!--Support system version-->
6     <CurrentVersion>
7       Hi3716CV200-eng 4.4.2 KOT49H eng.wangyang.20140517.185642 test-keys
8     </CurrentVersion>
9     <!--The target version-->
10    <TargetVersion>
11      Hi3716CV200-eng 4.4.2 KOT49H eng.wangyang.20140529.085406 test-keys
12    </TargetVersion>
13    <!--Upgrade Description-->
14    <Description>
15      Solve the following problem:
16      1.Upgrade to Android4.4 version
17      2.Support online upgrade
18    </Description>
19    <!--Upgrade download url-->
20    <FilePath>
21      http://203.195.181.80/e2372aad7c2b1a/update0517-0529-ota.zip
22    </FilePath>
23    <!--Release time-->
24    <ReleaseTime>
25      2014-05-29
26    </ReleaseTime>
27    <!--Upgrade file size-->
28    <FileSize>
29      37.1MB
30    </FileSize>
31    <!--Whether to forced to upgrade-->
32    <CompulsoryUpgrade>
33      true
34    </CompulsoryUpgrade>
35  </OTAUpgrade>
36 </HiSTBAndroidUpgrade>
```

## 2.5.2 在线非强制升级

### 问题描述

如何进行在线非强制升级？

### 解决办法

- 步骤 1 设置更新列表访问地址时，修改文件  
device/hisilicon/bigfish/packages/apps/HiSetting/src/com/android/hisiliconsetting/upgrade/OnlineUpgradeActivity.java。修改内容请参考“2.5.1.1 设置更新列表访问地址”。
- 步骤 2 配置更新信息请参考“2.5.1.2 配置更新信息”。
- 步骤 3 打开【HisiliconSetting】，如图 2-3 所示。



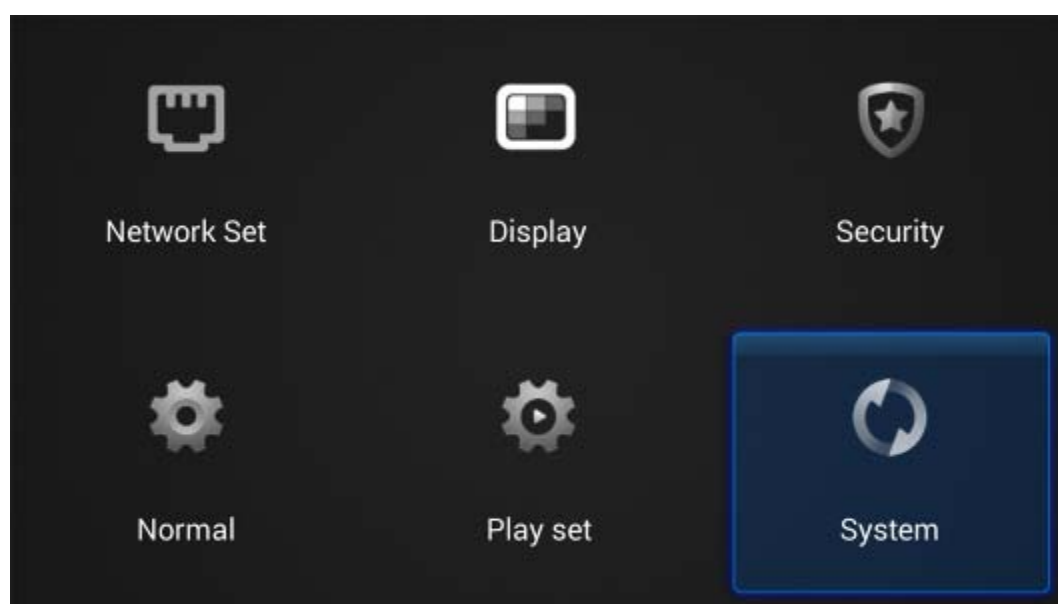


图2-3 打开【HisiliconSetting】



步骤 4 进入【System】，如图 2-4 所示。

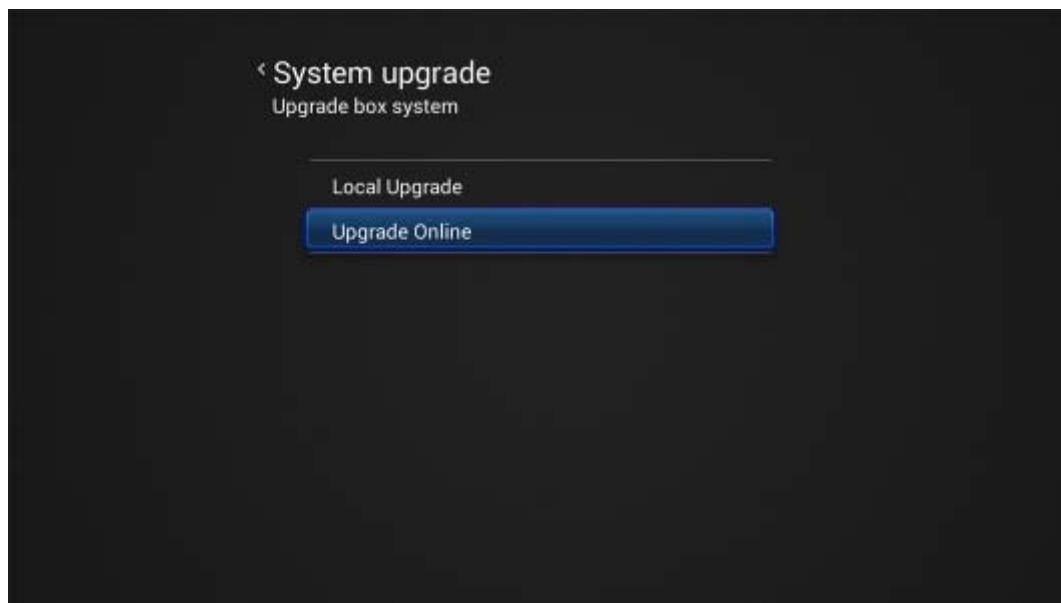
图2-4 进入【System】



步骤 5 进入【Upgrade Online】，如图 2-5 所示。

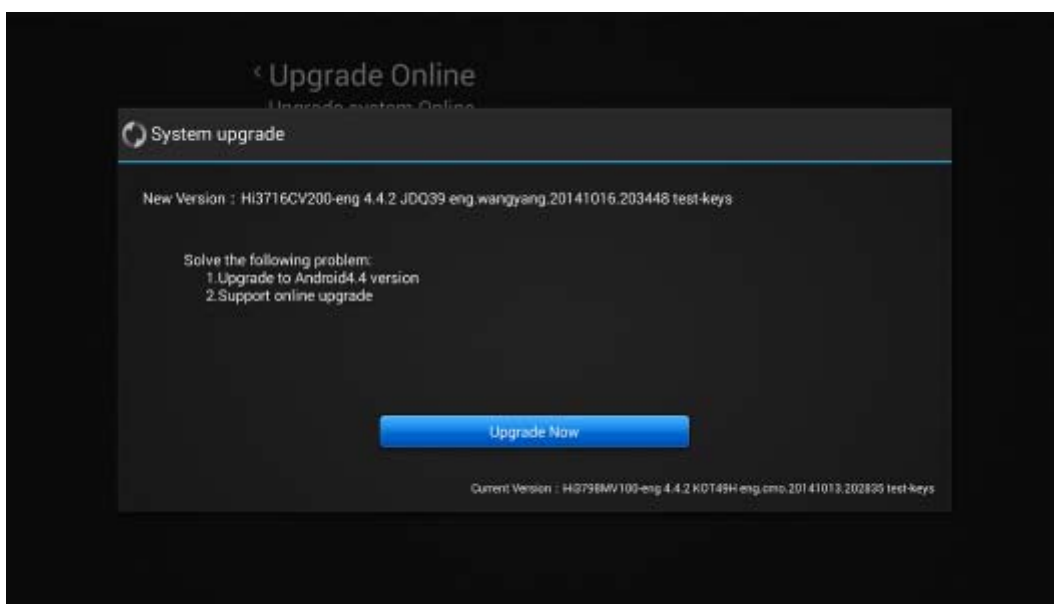


图2-5 进入【Upgrade Online】



步骤 6 如果有更新版本，则会弹出提示框，如图 2-6 所示。

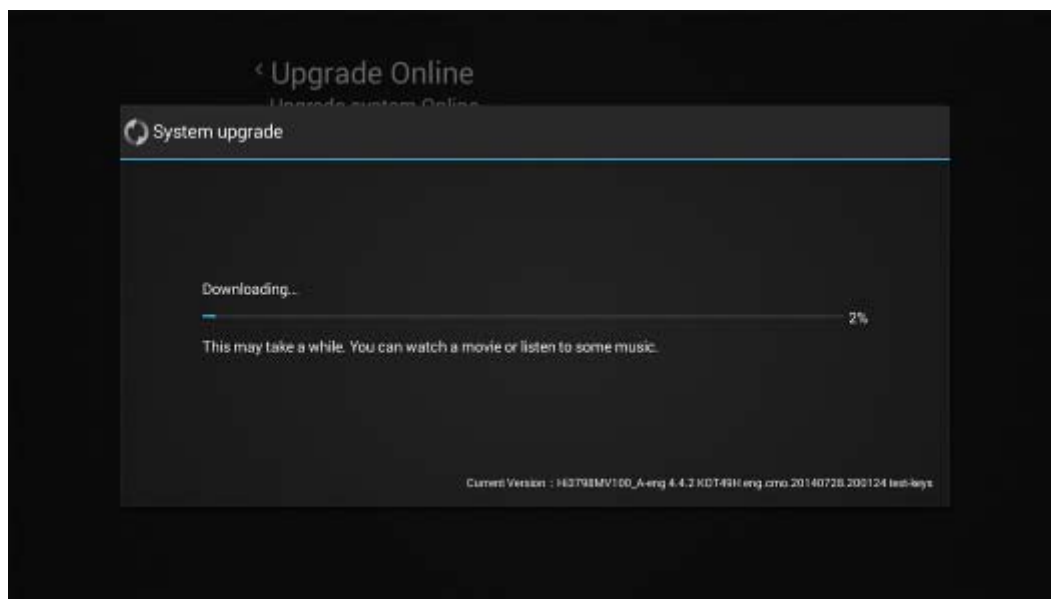
图2-6 弹出提示框



步骤 7 点击【Upgrade Now】，则开始进行下载，如图 2-7 所示。

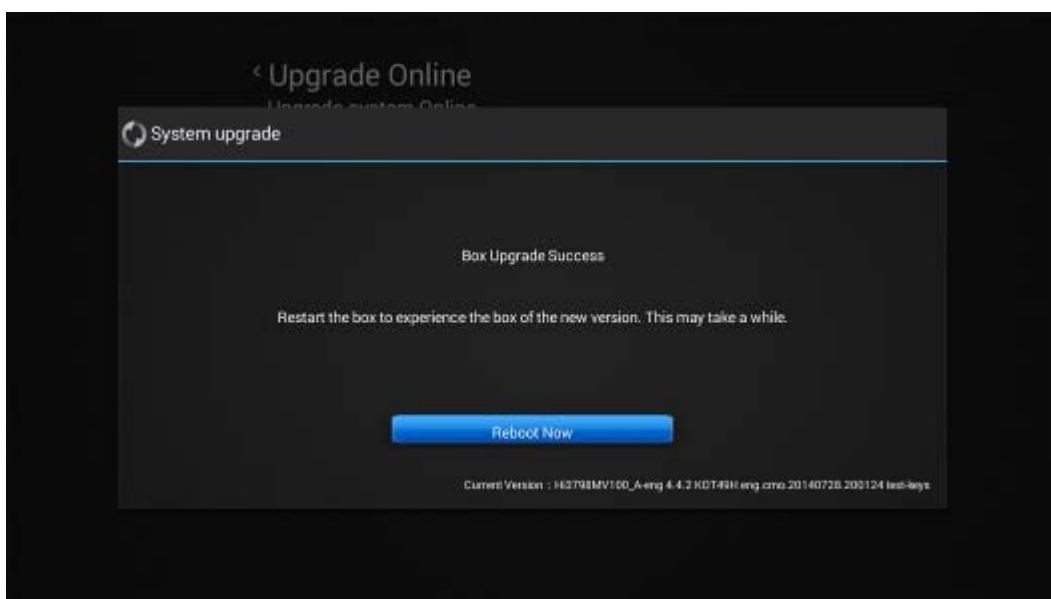


图2-7 开始进行下载



步骤 8 下载完毕，提示重启系统，如图 2-8 所示。

图2-8 下载完毕



步骤 9 点击【Reboot Now】，就开始重启系统并进行系统升级。

----结束



## 2.6 在线更新开机 Logo、fastplay、开机动画

### 2.6.1 更新开机 LOGO

#### 2.6.1.1 设置更新列表访问地址

##### 问题描述

系统启动后，后台服务会自动访问服务器上的更新列表，若有开机 LOGO 的更新信息，则自动下载并进行 LOGO 更新。如何指定更新列表的访问地址呢？

##### 解决办法

修改

device/hisilicon/bigfish/packages/apps/HiRMService/src/com/hisilicon/android/hiRMService/LogoUpgrade.java 文件中的如下代码：

```
private final String INTRA_NET_PATH = "no";
```

当 INTRA\_NET\_PATH = "no" 时，访问不到地址，则不进行更新；

将 no 替换成实际地址（例：INTRA\_NET\_PATH= “http://10.67.212.148/logo.xml”），则系统会访问 <http://10.67.212.148/logo.xml>，并解析更新信息。

INTRA\_NET\_PATH 默认值为 no。

#### 2.6.1.2 配置更新信息

##### 问题描述

系统会访问并解析服务器上的 XML 文件，如何在 XML 文件中标识更新信息？

##### 解决办法

创建 XML 文件，其中 **【LogoUpgrade】** 标签标识一个完整的更新信息，包含的子标签有：

- **【FilePath】**：更新包下载路径
- **【ReleaseTime】**：发布时间
- **【FileSize】**：更新包大小

通过比较发布时间来判断是否更新。如图 2-9 所示。



图2-9 Logo 更新列表信息

```
1 <?xml version="1.0" encoding="UTF-8" ?>
2 <HiSTBAndroidUpgrade>
3   <!--Logo Upgrade-->
4   <LogoUpgrade>
5     <!--Upgrade download url-->
6     <FilePath>
7       http://10.67.212.148/v5/wy/logo-wanglin.img
8     </FilePath>
9     <!--Release time-->
10    <ReleaseTime>
11      2014-04-01
12    </ReleaseTime>
13    <!--Upgrade file size-->
14    <FileSize>
15      168KB
16    </FileSize>
17  </LogoUpgrade>
18 </HiSTBAndroidUpgrade>
```

## 2.6.2 更新 Fastplay

### 2.6.2.1 设置更新列表访问地址

#### 问题描述

系统启动后，后台服务会自动访问服务器上的 fastplay 更新列表，若有更新信息，则自动下载并进行更新。如何指定更新列表的访问地址呢？

#### 解决办法

修改

device/hisilicon/bigfish/packages/apps/HiRMService/src/com/hisilicon/android/hiRMService/FastplayUpgrade.java 文件中的如下代码：

```
private final String INTRA_NET_PATH = "no";
```

当 INTRA\_NET\_PATH = "no" 时，访问不到地址，则不进行更新。

将 no 替换成实际地址（例：INTRA\_NET\_PATH = "http://10.67.212.148/fastplay.xml"），则系统会访问 <http://10.67.212.148/fastplay.xml>，并解析更新信息。

INTRA\_NET\_PATH 默认值为 no。

### 2.6.2.2 配置更新信息

#### 问题描述

系统会访问并解析服务器上的 XML 文件，如何在 XML 文件中标识更新信息？



## 解决办法

创建 XML 文件，其中【FastplayUpgrade】标签标识一个完整的更新信息，包含的子标签有：

- 【FilePath】：更新包下载路径
- 【ReleaseTime】：发布时间
- 【FileSize】：更新包大小

通过比较发布时间来判断是否更新。如图 2-10 所示。

图2-10 Fastplay 更新列表信息

```
1 <?xml version="1.0" encoding="UTF-8" ?>
2 <HiSTBAndroidUpgrade>
3   <!--Fastplay Upgrade-->
4   <FastplayUpgrade>
5     <!--Upgrade download url-->
6     <FilePath>
7       http://10.67.212.148/v5/hcb/fastplay_old.img
8     </FilePath>
9     <!--Release time-->
10    <ReleaseTime>
11      2014-04-01
12    </ReleaseTime>
13    <!--Upgrade file size-->
14    <FileSize>
15      32MB
16    </FileSize>
17  </FastplayUpgrade>
18 </HiSTBAndroidUpgrade>
19
```

## 2.6.3 更新开机动画

### 2.6.3.1 设置更新列表访问地址

#### 问题描述

系统启动后，后台服务会自动访问服务器上的开机动画更新列表，若有更新信息，则自动下载并进行更新。如何指定更新列表的访问地址呢？

#### 解决办法

修改

device/hisilicon/bigfish/packages/apps/HiRMService/src/com/hisilicon/android/hiRMService/AnimationUpgrade.java 文件中的如下代码：

```
private final String INTRA_NET_PATH = "no";
```

当 INTRA\_NET\_PATH = "no" 时，访问不到地址，则不进行更新。



将 no 替换成实际地址（例：INTRA\_NET\_PATH=“http://10.67.212.148/bootanimation.xml”），则系统会访问 http://10.67.212.148/bootanimation.xml，并解析更新信息。

INTRA\_NET\_PATH 默认值为 no。

### 2.6.3.2 配置更新信息

#### 问题描述

系统会访问并解析服务器上的 XML 文件，如何在 XML 文件中标识更新信息？

#### 解决办法

创建 XML 文件，其中【BootAnimationUpgrade】标签标识一个完整的更新信息，包含的子标签有：

- 【FilePath】：更新包下载路径
- 【ReleaseTime】：发布时间
- 【FileSize】：更新包大小

通过比较发布时间来判断是否更新。如图 2-11 所示。

图2-11 BootAnimation 更新列表信息

```
1 <?xml version="1.0" encoding="UTF-8" ?>
2 <HiSTBAndroidUpgrade>
3   <!--BootAnimation Upgrade-->
4   <BootAnimationUpgrade>
5     <!--Upgrade download url-->
6     <FilePath>
7       http://10.67.212.148/v5/wy/bootanimation-cts.zip
8     </FilePath>
9     <!--Release time-->
10    <ReleaseTime>
11      2014-04-26
12    </ReleaseTime>
13    <!--Upgrade file size-->
14    <FileSize>
15      8MB
16    </FileSize>
17  </BootAnimationUpgrade>
18 </HiSTBAndroidUpgrade>
19
```



# 3 生产及调试工具

## 3.1 厂测工具

### 问题描述

【厂测工具】是出厂测试的辅助工具，如何使用该工具？

### 解决方案

#### 【测试准备】

- 路由器  
开启 DHCP，开启 wifi，设置用户名 HiABC，密码 9988776655，加密方式 wpa2。
- 单板端
  - 插入网线，能通过 DHCP 获取到 IP 地址。
  - 所有 USB 口插入 U 盘，每个 U 盘根目录下分别存放 usb1.mpg，usb2.mpg，usb3.mpg，usb4.mpg。
  - SATA 口接入硬盘（可选），硬盘根目录下存放 sata.mpg。
  - SDCARD 口接入 SD 卡（可选），SD 卡根目录下存放 sdcard.mpg。

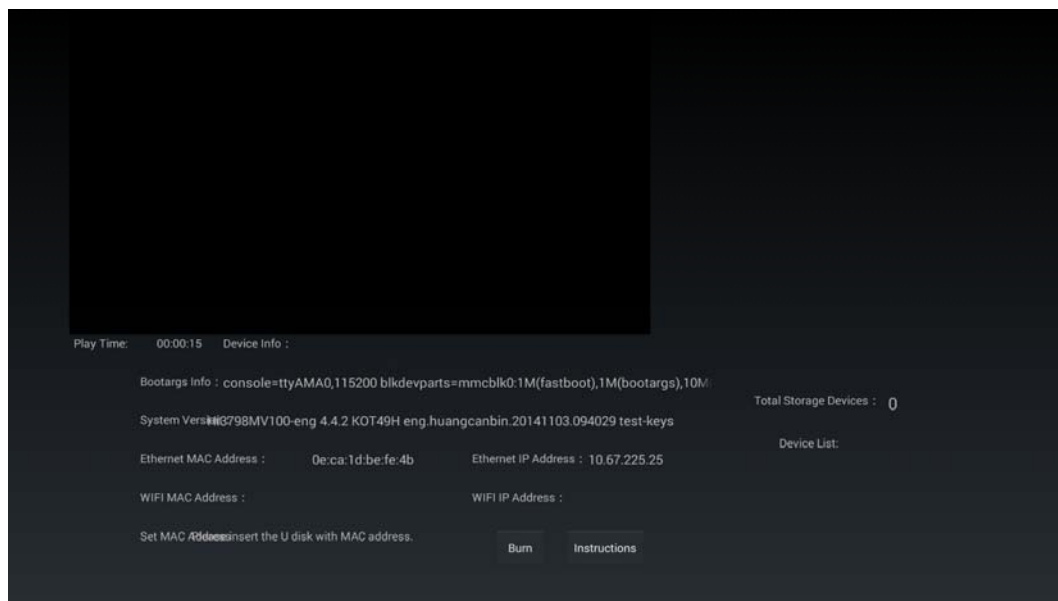
#### 【测试关注】

- 视频不断循环播放，无花屏、无闪烁。
- 有线 MAC 地址正常，有线 IP 地址正常。
- 无线 MAC 地址正常，无线 IP 地址正常。
- 已接入的设备数量正常，已接入的设备列表正常。





图3-1 厂测工具界面



【说明】厂测及老化工具是默认隐藏图标不做显示的，如果需要进入，可以在系统原生设置界面中依次按下遥控器的数字键 5918，就能启动厂测 APK 了。同理，如果想启动老化 APK，请参照如下代码自行修改。

代码路径 packages/apps/Settings/src/com/android/settings/Settings.java

```
private final String PACKAGE_NAME = "com.android.factorytest";
private final String ACTIVITY_NAME =
    "com.android.factorytest.factorytestActivity";
private int keyTemp = 0;
@Override
public boolean onKeyDown(int keyCode, KeyEvent event) {
    if (KeyEvent.KEYCODE_5 == keyCode) {
        keyTemp = 1;
    } else if (KeyEvent.KEYCODE_9 == keyCode) {
        if (keyTemp == 1)
            keyTemp++;
        else
            keyTemp = 0;
    } else if (KeyEvent.KEYCODE_1 == keyCode) {
        if (keyTemp == 2)
            keyTemp++;
        else
            keyTemp = 0;
    } else if (KeyEvent.KEYCODE_8 == keyCode) {
        if (keyTemp == 3) {
            try {
```



```
        Intent intent = new Intent();
        ComponentName name = new ComponentName(PACKAGE_NAME,
        ACTIVITY_NAME);
        intent.setComponent(name);
        intent.setAction(Intent.ACTION_VIEW);
        intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
        getApplication().startActivity(intent);
    } catch (Exception e) {
        e.printStackTrace();
    }
    } else
        keyTemp = 0;
    }
    return super.onKeyDown(keyCode, event);
}
```

## 3.2 指标测试工具

### 3.2.1 HDMITest

#### 问题描述

【HDMITest】是 HDMI 认证测试的辅助工具，如何使用该工具？

#### 解决方案








- 步骤 1 打开【HDMITest】
- 步骤 2 点击【...】按钮，选择码流文件
- 步骤 3 点击【开始测试】按钮，进入码流播放界面



说明

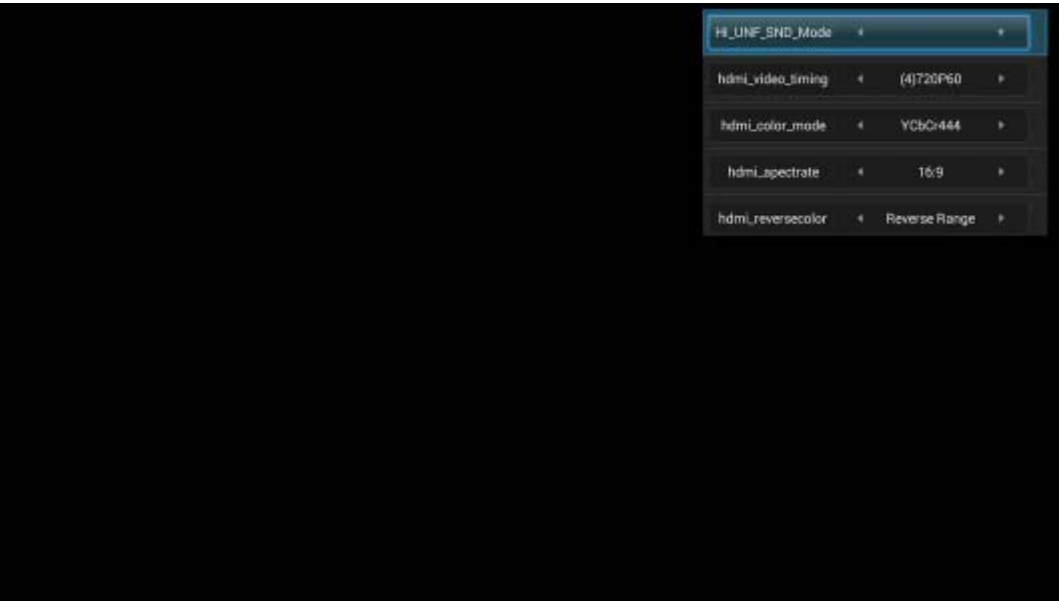
码流文件需要选择 HDMI 认证指定的码流，如图 3-2 所示。

图3-2 HDMI 认证指定的码流文件

	ccitt_mux_32K.ts	2010/12/27 15:07
	dada_192KHz_8ch_16bit.wav	2013/11/16 1:28
	Ghost.MLP	2010/10/8 15:44
	hd_dts_orchestra_long_lossless_dtshd...	2010/10/8 15:45
	Matrx625.Trp	2013/2/20 11:45
	PCM8_96Khz_pcm7.1.m2ts	2010/7/19 15:48
	serenity_english_5_1_1536.ddp	2010/10/8 15:44

步骤 4 遥控器按【菜单】键，在屏幕右上方弹出测试项列表，如图 3-3 所示。

图3-3 弹出测试项列表



步骤 5 遥控器【上】【下】是选择测试项，【左】【右】是选择测试值，【OK】是确认

步骤 6 通过专用仪器进行数据采集和分析即可。

----结束

### 3.2.2 CVBSTest

#### 问题描述

【CVBSTest】是 CVBS 测试的辅助工具，如何使用该工具？



## 解决方案

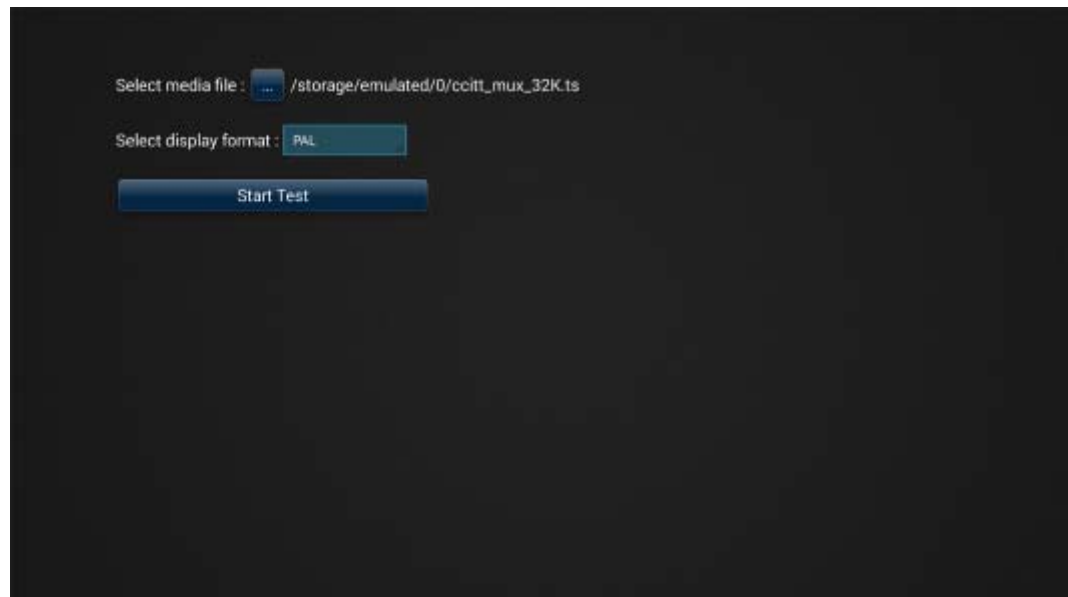
步骤 1 打开【CVBSTest】

步骤 2 点击【...】按钮，选择码流文件

注：码流请选择 CVBS 测试指定的码流，各厂家指定的码流都不一样。

步骤 3 点击下拉框，选择制式，如图 3-4 所示。

图3-4 选择码流和制式



步骤 4 点击【开始测试】按钮，进入码流播放界面。

步骤 5 通过专用仪器进行数据采集和分析即可。

----结束

## 3.2.3 SataAndUsbTest

### 问题描述

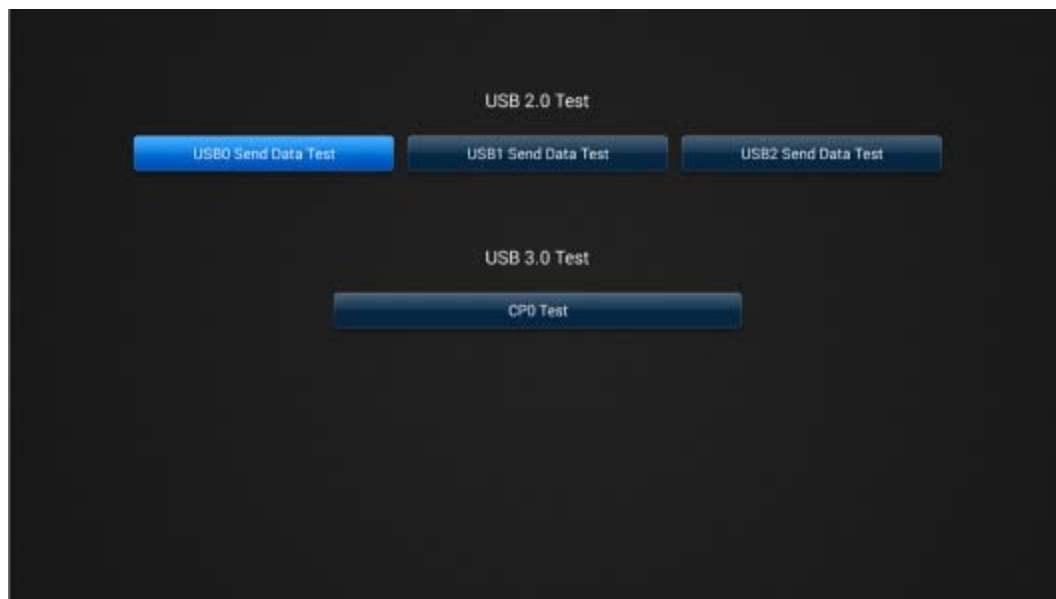
【SataAndUsbTest】是 SATA 和 USB 测试的辅助工具，如何使用该工具？

### 解决方案

步骤 1 打开【SataAndUsbTest】，显示测试项界面，如图 3-5 所示。



图3-5 测试项界面



步骤 2 根据测试需求点击对应的测试按钮。

步骤 3 通过专用仪器进行数据采集和分析即可。

----结束

## 3.3 远程调试工具

### 3.3.1 远程调试工具收件人配置

#### 问题描述

【远程调试工具】是可以抓取设备信息和 LOG，并给邮件给指定收件人的辅助工具，如何配置收件人呢？

#### 解决办法

修改 `/device/hisilicon/bigfish/development/apps/HiErrorReport/assets/mail.properties`

其中【to】标签标识收件人邮箱，并用分号隔开。【subject】标签标识邮件默认标题。【content】标签标识邮件默认内容。如图 3-6 所示。



图3-6 mail.properties 界面

```
#\u90AE\u7BB1\u914D\u7F6E
username = debuglog
pwd = 1qaz@WSX
from = debuglog@163.com
to = huawei.wangyang@hisilicon.com;wanting@hisilicon.com;jiangzhonglin@hisilicon.com;wangbaochang@hisilicon.com;
mail_protocol = mail.smtp.protocol
mail_auth = mail.smtp.auth
mail_host = mail.smtp.host
mail_port = mail.smtp.port
protocol = smtp
auth = true
host = smtp.163.com
port = 25
content = \u5B9D\u76D2\u51FA\u9519\u4E86,\u8BBE\u60C5\u8BF7\u770BLOG\u9644\u4EF6.
subject = \u5B9D\u76D2\u51FA\u9519Log\u62A5\u544A.
```

## 3.3.2 远程调试工具使用

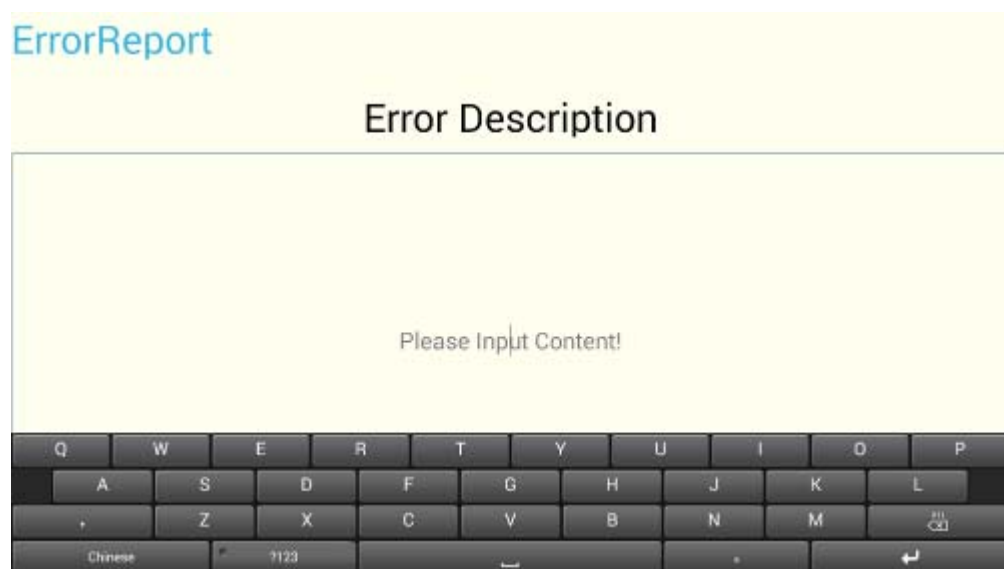
### 问题描述

【远程调试工具】是可以抓取设备信息和 LOG，如何使用该工具？

### 解决方案

步骤 1 打开【ErrorReport】，进入错误报告应用。如图 3-7 所示。

图3-7 错误报告界面



步骤 2 输入邮件内容，不输入则使用默认邮件内容。

步骤 3 点击“SendEmail”按钮发送邮件。

----结束



# 4 其他

## 4.1 电源指示灯

此处的电源指示灯指的是由 GPIO 驱动的 LED 灯。开机的时候显示蓝色或者绿色，待机的時候显示无色或者红色。一般可以分为单色灯和双色灯。单色指示灯可以通过 menuconfig 文件来配置；双色灯当前只支持手动修改代码。

### 4.1.1 配置单色电源指示灯

单色的 GPIO 驱动的 LED 灯可以在 SDK 的 `cfg.mak` 配置文件中配置。在 Board 菜单下的 Keyled Config 子菜单下，需要先选中 Gpio Light Support 选项，然后将 GPIO 的端口号设置到 Gpio Number 选项中，如图 4-1 (0x2f 表示 GPIO5\_7:  $5 * 8 + 7$ )：

图4-1 配置单色电源指示灯

```
Keyled Device Used (Ctl642 Keyled Select) --->
[*] Gpio Light Support
(0x2f) Gpio Number (NEW)
```

然后需要重新编译 boot 和 kernel 镜像文件。

### 4.1.2 配置双色电源指示灯

步骤 1 首先需要如 4.1.1 章节中一样，选中 Gpio Light Support 选项。

步骤 2 修改 SDK 目录下 `source\boot\fastboot\arch\arm\cpu\s5\start.S` 文件中的代码，可以根据不同的 GPIO 端口直接配置寄存器点亮绿灯，注意先配置 GPIO 的方向为输出，然后设置 GPIO 的输入电平；可以参考下面的代码：

```
#ifdef CONFIG_GPIO_LIGHT_SET
enable_green_light:
    ldr r1, =0xf8004000
    ldr r2, [r1, #0x400]
    orr r2, r2, #0x10
    str r2, [r1, #0x400]
```



```
ldr r2, [r1, #0x40]
orr    r2, r2, #0x10
str    r2, [r1, #0x40]

ldr r1, =0xf8009000
ldr    r2, [r1, #0x400]
orr    r2, r2, #0x80
str    r2, [r1, #0x400]
ldr    r2, [r1, #0x200]
and    r2, r2, #0x7f
str    r2, [r1, #0x200]
#endif
```

步骤 3 修改 SDK 目录下 source\msp\drv\pm\drv\_pmoc\_intf.c 文件 c51\_suspend 函数中的代码，在进入待机的时候点亮红灯：

```
#ifndef HI_GPIO_LIGHT_SUPPORT
    HI_REG_READ32(IO_ADDRESS(0xf8004400), ret);
    ret |= 0x10;
    HI_REG_WRITE32(IO_ADDRESS(0xf8004400), ret);
    HI_REG_READ32(IO_ADDRESS(0xf8004040), ret);
    ret &= 0xef;
    HI_REG_WRITE32(IO_ADDRESS(0xf8004040), ret);
    HI_REG_READ32(IO_ADDRESS(0xf8009400), ret);
    ret |= 0x80;
    HI_REG_WRITE32(IO_ADDRESS(0xf8009400), ret);
    HI_REG_READ32(IO_ADDRESS(0xf8009200), ret);
    ret |= 0x80;
    HI_REG_WRITE32(IO_ADDRESS(0xf8009200), ret);
#endif
```

----结束

## 4.2 开机向导

### 问题描述

开机向导相关的 APK 有：**【RunInstructions】**和**【HiInstructions】**。那么如何显示开机向导？

### 问题分析

当系统恢复出厂值后，第一次启动会运行**【RunInstructions】**，通过**【RunInstructions】**来启动**【HiInstructions】**，而此时，**【RunInstructions】**便设置为不可用，保证开机向导只运行一次。所以**【RunInstructions.apk】**的作用是启动**【HiInstructions】**，而





【HiInstructions】显示的才是真正的向导界面，这样做的目的是保证开机向导仅在第一次开机后会自动运行，并且在应用列表中显示图标，可以手动再次运行。

## 解决方案

- 自动运行：系统恢复出厂值后，第一次启动就会自动运行开机向导；
- 手动运行：系统启动后，在【Launcher】上找到【HiInstructions】并打开，就可以显示开机向导。

## 4.3 如何支持一般的 USB 手柄

### 问题描述

来了一款 USB 手柄，如果插上不能够使用，一般来说，是对应的驱动没有打开

### 解决方案

打开对应的驱动，以 Hi3798MV100 为例：

```
cd device/hisilicon/bigfish/sdk/source/kernel/linux-3.10.y/  
cp arch/arm/configs/hi3798cv100_android_defconfig .config  
make ARCH=arm CROSS_COMPILE=arm-hisiv200-linux- menuconfig  
Device Drivers ---> HID support ---> Special HID drivers --->选择手柄对应  
的驱动  
cp .config arch/arm/configs/hi3798cv100_android_defconfig  
make distclean
```

### 注意事项

如果你无法确定你手柄的驱动，可以尝试把 Special HID drivers—>所有的驱动打开，进行尝试。如果全部打开都无法使用，那么说明这个手柄驱动没有被内核集成，需要让厂商提供驱动，并集成到内核中。

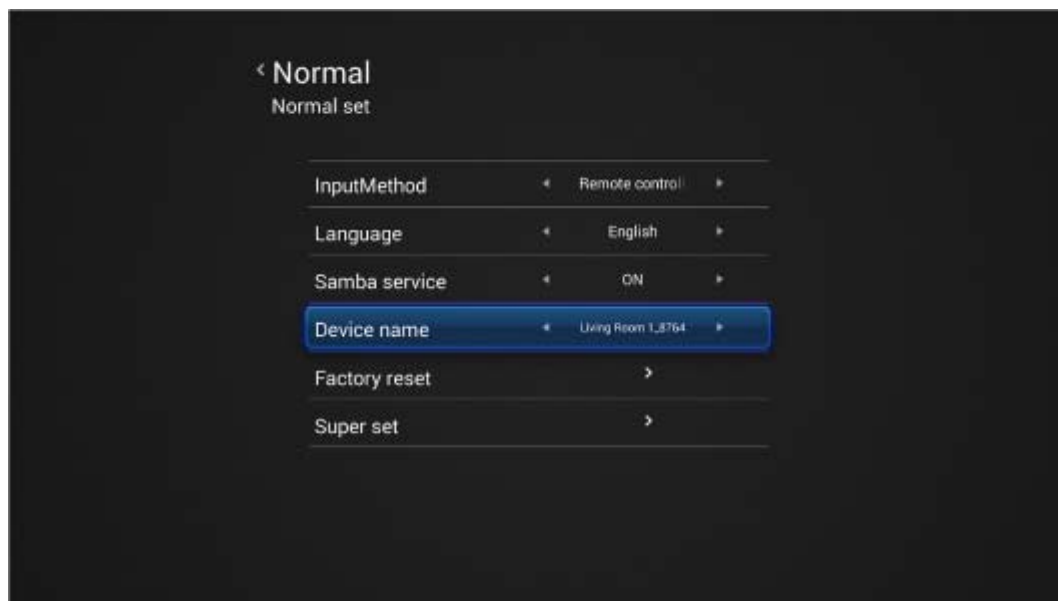
## 4.4 多屏统一开关

### 问题描述

如何设置 DLNA、Skyplay 的设备名称？



图4-2 通用设置界面



## 解决办法

修改 device/hisilicon/bigfish/packages/apps/HiSetting/res/values/ attrs.xml 中的 device\_name\_entries\_value 字符串数组的定义，当前只支持英文。如图 4-3 所示。

图4-3 修改字符串

```
82      <string-array name="device_name_entries_value">
83          <item>Living Room 1</item>
84          <item>Living Room 2</item>
85          <item>Bedroom 1</item>
86          <item>Bedroom 2</item>
```

## 4.5 如何修改开机向导中的二维码下载地址

### 问题描述

开机向导中包含了多屏客户端的二维码下载地址。如何修改该下载地址，生成新的二维码？

### 解决办法

可以通过修改 device/hisilicon/bigfish/packages/apps/HiInstructions/src/com/hisilicon/instructions/PageActivity3.java 的第 10 行，mURL 即 App 所在 URL。如图 4-4 所示。



图4-4 修改代码

```
8
9 public class PageActivity3 extends PageActivity {
10     private final static String mURL = "http://2.moonlightjewelbox.si
11     private final static int mQRImgWidth = 360;
12     private final static int mQRImgHeight = 288;
13     private boolean isSign;
14
```

## 4.6 设置中默认参数配置

- 有线网络设置相关默认值修改

- 有线开关默认值修改

修改文件 frameworks\base\packages\SettingsProvider\res\values\defaults.xml 中：

```
<bool name="def_ethernet_on">true</bool>的值，true默认有线设置开关  
打开，false默认关闭
```

- DHCP 与 Static 默认修改

无默认值。

- Wifi 开关默认值修改

修改文件 frameworks\base\packages\SettingsProvider\res\values\defaults.xml 中

```
<bool name="def_wifi_on">false</bool>的值，true默认wifi开关打开，false默  
认关闭
```

- Samba 默认开关修改

修改文件 packages/apps/Settings/src/com/android/settings/MyBroadCastReceiver.java  
中参数：

```
private static final String SAMBA_STATUS_DEF_VALUE="true"的值，“true”  
为默认第一次开机或回复出厂设置后samba默认打开，“false”为关闭。
```

- launcher 默认值

无默认值。

- volumes 默认值

修改文件 frameworks\base\media\java\android\media\AudioManager.java 中：

```
int[] DEFAULT_STREAM_VOLUME 中11, // STREAM_MUSIC的值，取值范围0-15默认值  
为开机后默认的音量大小。
```

- 第三方 APK 安装的允许开关

修改文件 frameworks\base\packages\SettingsProvider\res\values\defaults.xml 中：

```
<bool name="def_install_non_market_apps">false</bool>的值，true默认允许  
安装第三方应用，false不允许安装。
```

- 语言默认值开关



修改文件 build/target/product/ full\_base.mk 中:

```
PRODUCT_LOCALES := en_US的值。例如简体中文改为PRODUCT_LOCALES := zh_CN
```

- 分辨率相关默认值

UI size, Customer display format, Display area set 默认值在 base 里修改。

Optimal display format 修改文件

device/hisilicon/bigfish/frameworks/hidisplaymanager/hal/hi\_adp\_hdmi.c 中:

```
int getOptimalFormat()//0 is disable; 1 is enable
{
    char buffer[BUFLLEN];
    int value = 0;
    property_get("persist.sys.optimalfmt.enable", buffer, "1");
    value = atoi(buffer);
    ALOGE("getOptimalFormat, enable = %d", value);
    return value;
}
```

修改 property\_get("persist.sys.optimalfmt.enable", buffer, "1");中默认的返回值，系统第一次没有属性“persist.sys.optimalfmt.enable”值，返回默认值。1 默认打开自适应到最佳显示制式，0 默认关闭。

- 系统错误提示对话框优化
- 修改文件/frameworks/base/services/java/com/android/server/am/ActivityManagerService.java 中 Handler 的 SHOW\_ERROR\_MSG 消息处理:

```
if (mShowDialogs && !mSleeping && !mShuttingDown&&false) {
    Dialog d = new AppErrorDialog(mContext,
        ActivityManagerService.this, res, proc);
    d.show();
    proc.crashDialog = d;
} else {
    // The device is asleep, so just pretend that the
user
    // saw a crash dialog and hit "force quit".
    if (res != null) {
        res.set(0);
    }
}
```

将 if 条件中的&&false 去掉将显示系统中的“XX 应用停止运行”和“XX 进程停止运行”两种提示对话框，加上&&false 则不显示这两种提示对话框。如图 4-5、图 4-6 所示。



图4-5 XX 应用停止运行

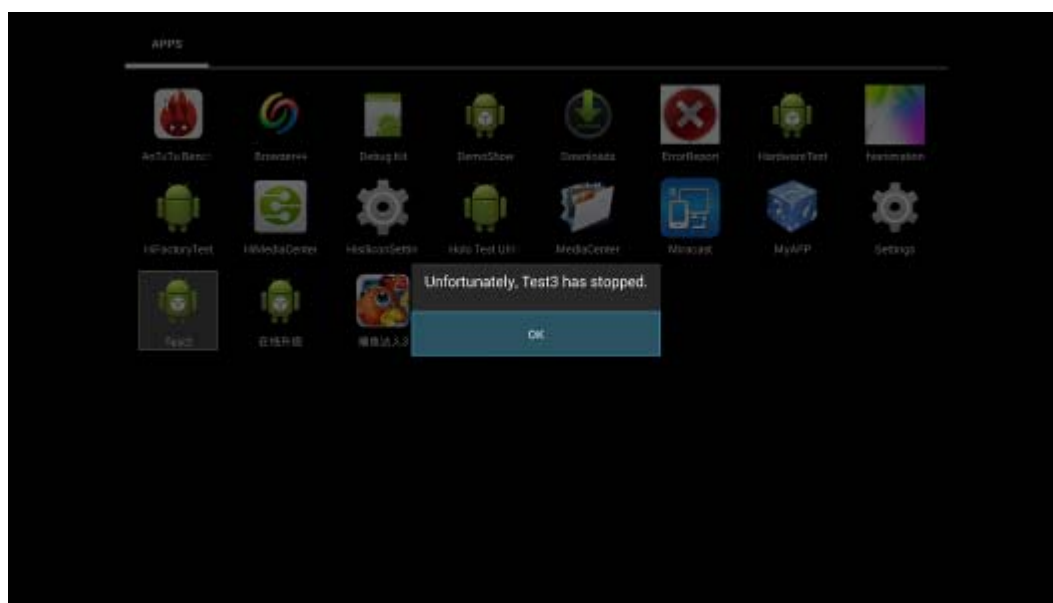


图4-6 XX 进程停止运行

