



投屏器

开发指南

文档版本 00B01

发布日期 2019-08-14

版权所有 © 上海海思技术有限公司2019。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海思技术有限公司的商标。

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。

上海海思技术有限公司

地址： 深圳市龙岗区坂田华为总部办公楼 邮编：518129

网址： <http://www.hisilicon.com/cn/>

客户服务邮箱： support@hisilicon.com



前言

概述

本文档介绍投屏器软件开发、生产测试所依赖的流程、接口等。

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
Hi3716M芯片	V430

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
- STBM的软件开发工程师

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
2019-08-14	00B01	第1次临时版本发布。



目录

前言.....	i
1 概述.....	1
1.1 投屏器介绍.....	1
1.2 软件架构.....	2
1.3 重要概念.....	3
2 SDK 使用.....	4
2.1 SDK 介绍.....	4
2.2 SDK 编译.....	4
3 重点开发流程.....	6
3.1 蓝牙配网流程.....	6
3.1.1 流程.....	6
3.1.2 API 介绍.....	7
3.1.3 示例.....	7
3.2 Wifi 接口使用.....	7
3.2.1 配网模式（STA 和 P2P 共存）.....	7
3.2.2 连接和断开网络.....	8
3.2.3 启动和关闭酒店模式.....	8
3.2.4 关闭 WiFi.....	10
3.3 Miracast.....	10
3.4 DLNA 和 Airplay.....	10
3.5 UI 开发.....	10
3.6 升级方案.....	10
3.6.1 方案介绍.....	10
3.6.2 升级软件版本生成.....	11
3.6.2.1 SDK 修改.....	11
3.6.2.2 编译镜像.....	11
3.6.2.3 烧写镜像.....	12
3.6.3 制作升级包.....	12
3.6.3.1 打包升级文件.....	12
3.6.3.2 签名升级包.....	13
3.6.4 接口使用说明.....	13
3.6.5 升级方式.....	14



4 工厂测试.....	16
4.1 方案介绍.....	16
4.1.1 方案介绍.....	16
4.1.2 准备工作.....	16
4.1.3 检测流程.....	16
4.1.3.1 模式检测.....	16
4.1.3.2 WiFi 检测.....	16
4.1.3.3 Bluetooth 检测.....	17
4.1.3.4 音视频播放.....	17
4.1.3.5 Key/LED 检测.....	18
4.1.3.6 SN 烧写.....	18
4.2 参考代码.....	18
5 FAQ.....	19
5.1 常见问题.....	19
5.1.1 渐进式 JPEG 图片无法解析.....	19
5.1.2 苹果 Airplay 播放卡顿.....	19
5.1.3 图片大小超过 8MB 的图片解码失败.....	20
5.1.4 手机拍摄的视频投屏播放失败.....	20
5.1.5 手机拍摄的视频投屏播放卡顿.....	20
5.2 升级校验修改.....	20
5.3 接口使用说明.....	20



1 概述

1.1 投屏器介绍

投屏器是一种可以接受手机、Pad等智能终端投屏，并将所屏内容解码后在电视上输出的设备。当手机跟投屏器连接同一个局域网时，可以轻松地把手机上的照片、视频、音频投射到投屏器，最后在大屏上输出，获得更佳的音视频体验，与家人分享你的愉快瞬间。

图 1-1 投屏器的使用场景



投屏的源设备可以是手机，也可以是PC、笔记本、Pad等设备。

投屏器可以支持投屏和同屏两种场景：

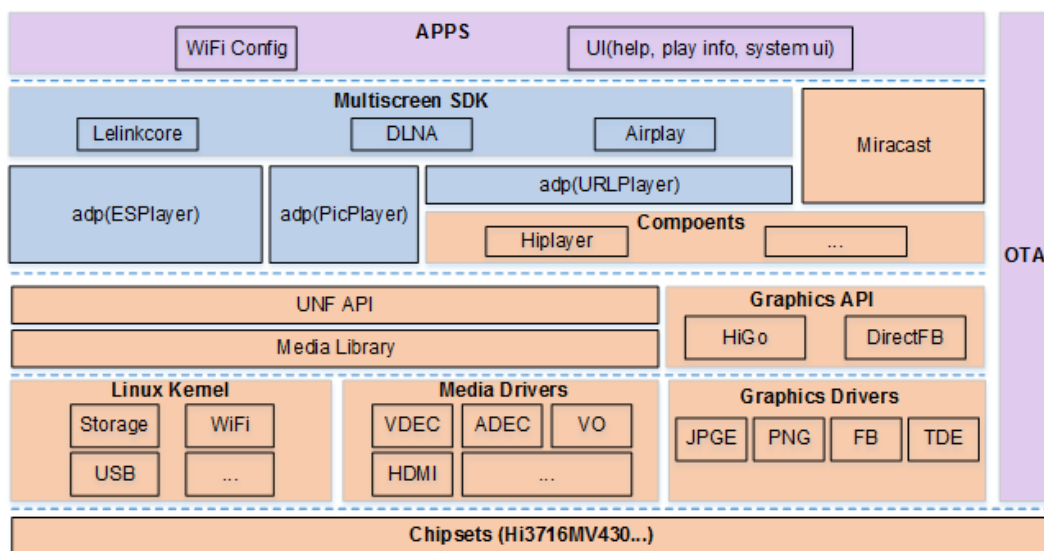
- 投屏
这种场景需要先将投屏器接入使用环境内（家庭、办公场所、酒店等）的WiFi网络，源设备通过DLNA等协议将音频、视频、图片的链接发送到投屏器，投屏器通过自身的WiFi网络去下载播放，最终在电视上输出。
- 同屏
这种场景由源设备将自身的屏幕及音频输出编码成标准的音视频流传送到投屏器，由投屏器来解码并在电视输出，获取大屏输出的体验。
不同的同屏协议有不同的实现方式，取决于最终产品支持的协议。典型方式是源设备和投屏器通过基于P2P直连的Miracast协议来实现。

1.2 软件架构

海思的投屏器解决方案支持客户基于我们的标准SDK做产品开发。产品的整体软件架构如图1-2所示。其中，

- 橙色部分属于SDK，由海思交付；
- 蓝色部分属于多屏SDK，可以第三方或OEM实现；
- 紫色部分属于UI和应用部分，可以第三方或OEM实现。

图 1-2 投屏器软件架构图



海思SDK中包含的功能有：

- Linux Kernel: 标准的Linux内核；
- Media Drivers: 媒体相关驱动，包括音视频解码器（ADEC\VDEC\AVPLAY）、音视频输出设备驱动（Sound/VO/Display/HDMI）等；
- Graphics Drivers: 包括图形处理、加速相关驱动（JPEG\PNG解码器、2D加速引擎TDE、显存管理FB等）；
- Graphics API: HiGo组件可以实现图层管理、线、矩形、文本处理、图片解码等基本的图形接口；另外也提供标准DirectFB组件，可以根据需要选择使用；
- UNF API: 海思音视频驱动、部分外设驱动对用户态提供统一接口，统一命名为HI_UNF_XXX；
- Components部分包含URL播放器HiPlayer，以及Miracast组件；
由第三方或OEM实现的部分有：
- Multiscreen SDK: 根据需要提供支持的协议，典型如DLNA等；
- Application: 主要包括UI、网络管理、播放界面、事件处理等部分。



1.3 重要概念

- DLNA

Digital Living Network Alliance数字生活网络联盟，由索尼，英特尔，微软等公司发起，适用于音频，视频，图片推送，可将多媒体文件通过无线方式投射到投屏器上。

- Miracast

由WiFi联盟于2012年所制定的，是以WiFi直联为基础的无线显示标准，支持将手机屏幕内容传送到投屏器的镜像投屏功能。



2 SDK 使用

2.1 SDK 介绍

发布包里面一级文件：

configs	编译配置文件
out	编译输出目录
sample	sample
scripts	kconfig
source	主要源代码
third_party	第三方代码
tools	工具

Hi3716M V430投屏器相关的配置文件有：

- configs/hi3716mv430/hi3716mv43dmd_hi3716mv430_embedded_dongle_cfg.mak
用于编译投屏器的参考boot/kernel/rootfs以及驱动；
- configs/hi3716mv430/
hi3716mv43dmd_hi3716mv430_embedded_dongle_loader_cfg.mak
用于编译投屏器的参考loader。

2.2 SDK 编译



以下编译方式请在发布包的一级目录下操作。

请注意拿到的SDK可能是基础SDK或是补丁版本，补丁版本需要打到相对应的基础SDK上，具体请参考使用版本的版本描述文件。

编译前需要拷贝一个配置文件到一级目录下作为cfg.mak，方式如下：

```
source env.sh
cp configs/hi3716mv430/
hi3716mv43dmd hi3716mv430 embedded dongle xx.mak ./cfg.mak
make build -j
```

如果服务器较慢，可以不带-j参数（多线程编译）。

编译输出的库和镜像在out/hi3716mv430/目录下。

如果需要修改配置项，可以在一级目录下执行make menuconfig来重新配置。



如果需要重新编译，请执行：

```
make clean  
make build -j
```

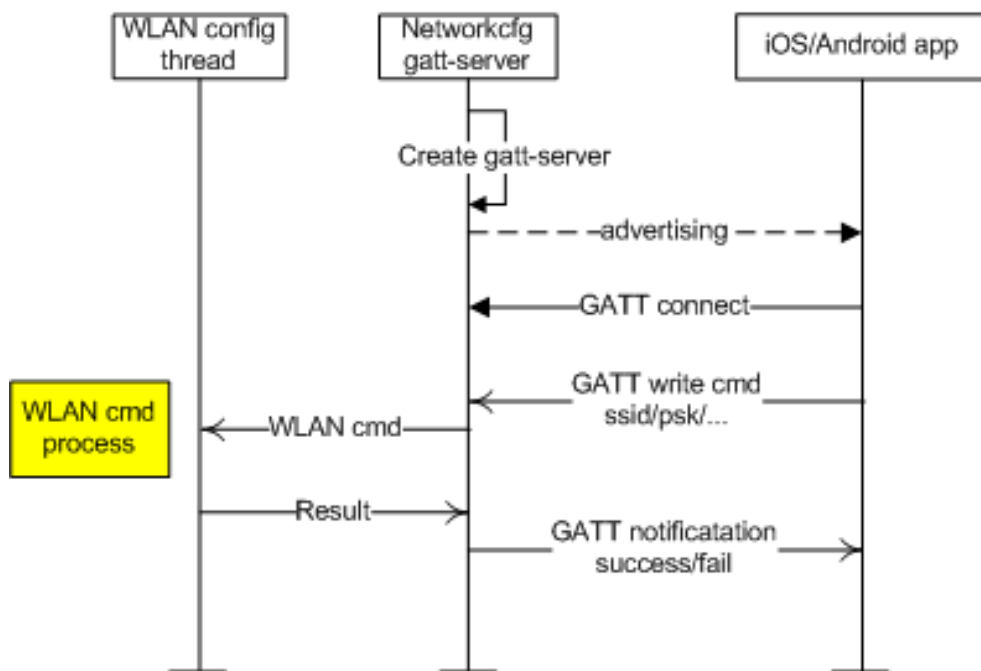
3 重点开发流程

3.1 蓝牙配网流程

3.1.1 流程

投屏器的蓝牙适配流程如图3-1所示。

图 3-1 蓝牙配网流程



步骤1 单板启动后，Networkcfg会创建 gatt-server，用户可在此配置设备名称，然后发起广播。

步骤2 手机上点击app，保证软件运行起来。

步骤3 进入配网界面，扫描附近的投屏器设备。



- 步骤4** 点击需要进行配网的投屏器，此时手机与投屏器会建立GATT连接，然后进入配网界面。
- 步骤5** 在配网界面配置好热点名称和输入正确的连接密码。
- 步骤6** 点击配网按键，app将会通过蓝牙GATT把ssid和psk等信息发送给投屏器，投屏器端的蓝牙收到配网命令后将连接命令发给WiFi进程进行连接网络。
- 步骤7** WiFi进程通过事件将连接状态发给投屏器应用，投屏器通过GATT将状态通知到手机app（如：66：成功 11：连接中 22：失败）。

----结束

3.1.2 API 介绍

参考原厂的SDK。

3.1.3 示例

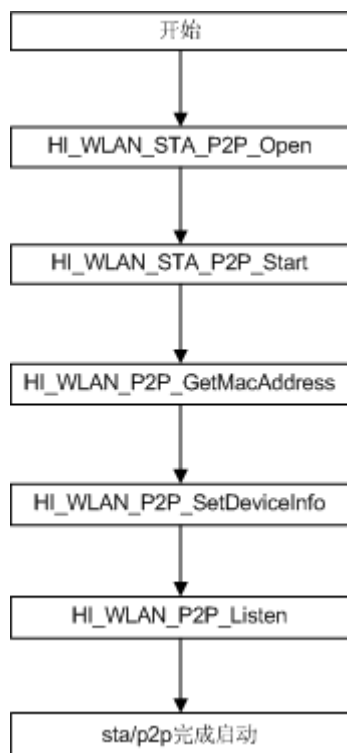
参考原厂的SDK。

3.2 Wifi 接口使用

3.2.1 配网模式（STA 和 P2P 共存）

配网模式打开WiFi的流程如下 [图3-2](#)所示。

图 3-2 打开 WiFi 流程图



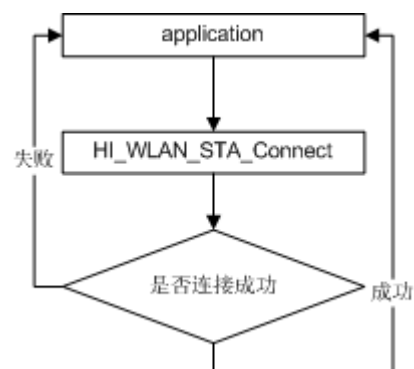
注意

sta(station)/p2p(peer-to-peer)完成启动后，可进行网络连接和p2p连接

3.2.2 连接和断开网络

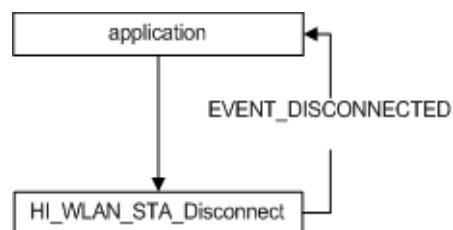
sta连接流程如下图3-3所示；断开网络流程如下图3-4所示。

图 3-3 连接网络流程

**注意**

连接成功后需要启动udhcpd获取IP.

图 3-4 断开网络流程



3.2.3 启动和关闭酒店模式

启动酒店模式如下图3-5所示；关闭酒店模式如下图3-6所示。

图 3-5 进入酒店模式流程

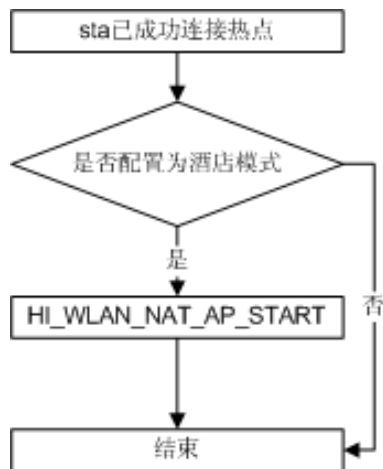
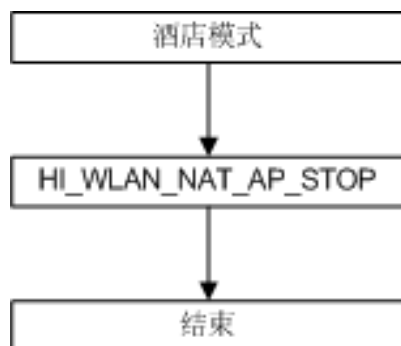


图 3-6 退出酒店模式流程



注意：

酒店模式分配的 IP 范围是 192.168.49.20 ~ 192.168.49.254，若需要修改，请参照如下步骤更改：

步骤1 打开“source/component/wifi/interface/src/sta/wlan_hal.c”文件修改 wlan_creat_udhcpd_config 函数如下的内容：

```

asprintf(&wbuf, "start          192.168.49.20\n"
"end            192.168.49.254\n"
"interface      %s\n"
"opt    dns     %s\n"
"option subnet  255.255.255.0\n"
"opt    router   192.168.49.1\n"
"option domain  local\n"
"option lease   864000\n",
ifname, network_dns);
  
```

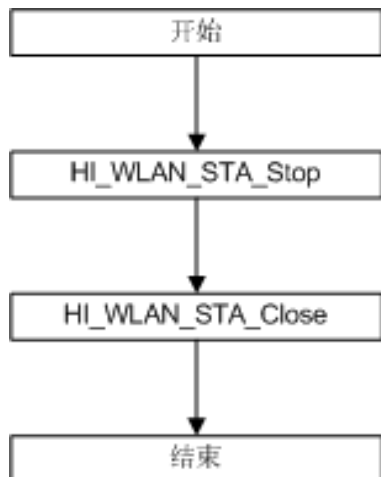
步骤2 打开“source/component/wifi/interface/src/sta/wlan_sm.c”文件修改 SOFTAP_IPADDR 字符串数组。

----结束

3.2.4 关闭 WiFi

关闭WiFi的流程如下图3-7所示。

图 3-7 关闭 WiFi 流程



注意

p2p不能单独关闭，可调用 HI_WLAN_P2P_StopListen 接口退出 listen 状态，退出 listen 后就不能被其他设备扫描到

3.3 Miracast

请参考《Miracast开发指南》。

3.4 DLNA 和 Airplay

自行开发或参考第三方文档。

3.5 UI 开发

参考《HMS开发指南》中Higo相关章节。

3.6 升级方案

默认提供双备份升级方案，如果需要其他方案可自行开发。

3.6.1 方案介绍

投屏器默认使用双备份升级方案，即关键分区(如Kernel，Rootfs)提供A、B分区，系统起来的时候会检测分区的版本号，开机自动使用新的版本号，如果升级失败，会以原来的镜像开机，保证投屏器可以正常开机。



3.6.2 升级软件版本生成

3.6.2.1 SDK 修改

投屏器默认支持升级校验，需要按照如下步骤生成秘钥对和修改基础SDK软件：

步骤1 使用CASignTool工具获取RSA秘钥对。

执行以下操作：

```
cd $(SDK_DIR)/tools/linux/utils/advca/CASignTool_Linux_STB/bin64
./CASignTool 8 3 .
```

此时当前目录会生成以下文件：

```
rsa_priv.txt rsa_pub.bin rsa_pub_crc.bin rsa_pub.h rsa_pub.txt
```

其中rsa_priv.txt和rsa_pub.txt分别是RSA私钥和公钥，需成对使用（用私钥签名数据用公钥校验数据）；注意保存好这对秘钥，后续升级包签名的时候需要使用。

步骤2 替换校验代码中的公钥数组

将生成的rsa_pub.h文件内容中的rsa_pub_N和rsa_pub_E数组内容替换到source/component/loader/ab_upgrade/source/verify/basic_ssd.c文件对应的s_rsa_pub_key_n和s_rsa_pub_key_e数组里面。

----结束

图 3-8 替换校验代码中的公钥数组

```
00053: static hi_u8 s_rsa_pub_key_n[SSD_RSA_PUBLIC_KEY_N_LENGTH] =
00054: {
00055: 0xa8, 0x74, 0xb1, 0xdd, 0x4b, 0x55, 0xf9, 0x81, 0xd9, 0xbf, 0x0c, 0x1b, 0x0a, 0x0c, 0x95, 0x8f,
00056: 0x3b, 0x3e, 0x04, 0x3e, 0x8f, 0x70, 0x3e, 0xa9, 0x99, 0xae, 0x94, 0xd4, 0x8d, 0x72, 0xb5, 0x4e,
00057: 0xa7, 0xf5, 0xba, 0x99, 0xed, 0xd0, 0x0e, 0x8f, 0x82, 0x78, 0xd3, 0x96, 0x14, 0x11, 0xd9, 0xdd,
00058: 0xb5, 0xc2, 0x07, 0x5c, 0xc5, 0x49, 0xc4, 0x09, 0xf0, 0x55, 0xb8, 0x20, 0x56, 0x28, 0x9b, 0x7b,
00059: 0xda, 0x80, 0xae, 0xdf, 0xe0, 0xfa, 0x53, 0x49, 0x5e, 0x1f, 0xbb, 0xc6, 0x11, 0xb4, 0xf3, 0xb4,
00060: 0xb3, 0x16, 0x48, 0x77, 0x8b, 0x79, 0x67, 0xf6, 0x9e, 0x8f, 0x9f, 0x97, 0x7d, 0x6a, 0xac, 0xa4,
00061: 0x53, 0x45, 0xe8, 0xa2, 0xd6, 0xb2, 0x01, 0x55, 0xe4, 0x96, 0xf3, 0xb5, 0x17, 0x5b, 0xd8, 0xec,
00062: 0x39, 0xb9, 0x7b, 0xb9, 0x6b, 0x15, 0x54, 0xfb, 0x4c, 0x35, 0xb4, 0xfb, 0x52, 0xd9, 0x0f,
00063: 0x65, 0xa7, 0x35, 0x68, 0x7c, 0x54, 0x68, 0x73, 0x4d, 0xa8, 0xf8, 0xa1, 0x6d, 0x09, 0x42, 0x63,
00064: 0x31, 0xaa, 0x6f, 0x8f, 0xf7, 0xfd, 0xcb, 0x79, 0xa7, 0x02, 0xb7, 0x8c, 0xb2, 0x43, 0x37, 0x33,
00065: 0xd4, 0x4c, 0xf5, 0xd8, 0x98, 0xaf, 0xb2, 0x22, 0x2a, 0x37, 0x03, 0x79, 0xc0, 0x61, 0xaf, 0x67,
00066: 0x19, 0xae, 0xad, 0xb0, 0xf9, 0xf8, 0xa6, 0x3b, 0x2a, 0x97, 0x1a, 0xc5, 0x23, 0x11, 0x78, 0x22,
00067: 0x64, 0x98, 0x1b, 0x56, 0x84, 0x74, 0x9e, 0x14, 0x1b, 0xe1, 0xc2, 0x92, 0xf6, 0xd6, 0xd5, 0xac,
00068: 0x3e, 0x19, 0xca, 0x72, 0xea, 0x5d, 0xfa, 0x43, 0x94, 0x96, 0x84, 0x7e, 0xef, 0x08, 0x50, 0x84,
00069: 0x4a, 0xc8, 0xa4, 0xbd, 0xef, 0x35, 0x48, 0xb8, 0xfc, 0x0c, 0x31, 0xba, 0x0d, 0x2a, 0x48, 0xf8,
00070: 0xb1, 0x38, 0x34, 0xcf, 0x5f, 0x6d, 0xf9, 0xac, 0xb8, 0x40, 0x54, 0xac, 0x64, 0x70, 0x19, 0x97
00071: };
00072:
00073: static hi_u8 s_rsa_pub_key_e[SSD_RSA_PUBLIC_KEY_E_LENGTH] =
00074: {
00075: 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
00076: 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
00077: 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
00078: 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
00079: 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
00080: 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
00081: 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
00082: 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
00083: 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
00084: 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
00085: 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
00086: 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
00087: 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
00088: 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
00089: 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
00090: 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00, 0x00,
00091: };
00092:
```

注意

rsa_pub_N数组最后多了一个逗号，拷贝的时候需要将其删掉。

3.6.2.2 编译镜像

参考如下命令，使用loader配置文件编译出升级软件镜像。

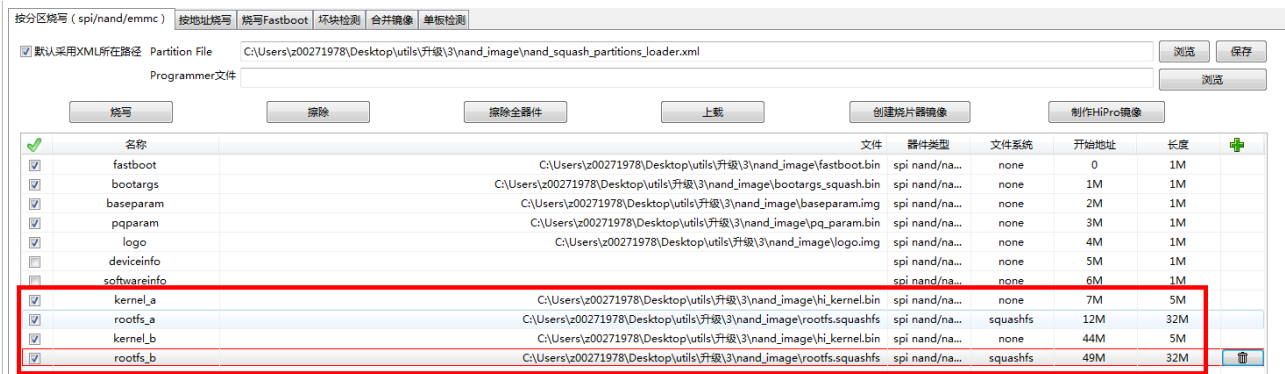


```
cd $(SDK DIR)
cp configs/hi3716mv430/
hi3716mv43dmdl hi3716mv430 embedded dongle loader cfg.mak cfg.mak
make clean; make build
```

3.6.2.3 烧写镜像

进入如图3-9所示界面进行镜像烧写。

图 3-9 烧写镜像



注意

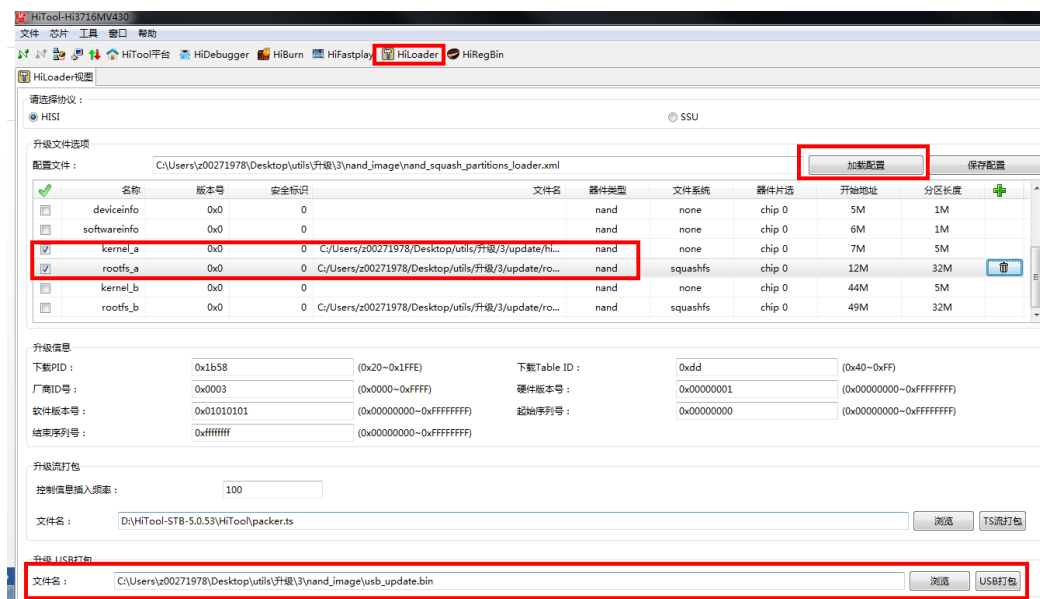
- 必须选择nand_squash_partitions_loader.xml。
- 第一次烧写固件，必须烧写kernel_b与rootfs_b分区，这个分区镜像内容使用kernel_a与rootfs_a分区同样镜像即可。
- fastboot分区暂不支持升级，其它分区均可升级，同时支持新增分区升级。

3.6.3 制作升级包

3.6.3.1 打包升级文件

打开HiTool→HiLoader工具，加载xml表格后选择需要升级的分区文件，点击USB打包生成升级文件usb_update.bin，如下图3-10所示。

图 3-10 打包升级文件



3.6.3.2 签名升级包

投屏器默认支持升级校验，需要使用3.6.2.1 SDK修改步骤一中生成的私钥来签名打包好的升级包：

将升级包usb_update.bin拷贝至服务器CASignTool工具同级目录下：

执行以下命令：

```
cd $(SDK_DIR)/tools/linux/utlis/advca/CASignTool_Linux_STB/bin64
./CASignTool 10 -K rsa_priv.txt -R usb_update.bin -O usb_update.bin.sig
./CASignTool 11 -K rsa_priv.txt -R usb_update.bin.sig -O
usb_update.bin.sig
```

此时工具将会把usb_update.bin用rsa_priv.txt签出一个签名升级包usb_update.bin.sig。

3.6.4 接口使用说明

- 示例代码：
\$(SDK_DIR)/source/component/loader/ab_upgrade/source/main.c
- 头文件：
\$(SDK_DIR)/source/component/loader/ab_upgrade/include/hi_unf_upgrade.h
- 动态库：
\$(SDK_DIR)/out/hi3716mv430/hi3716mv430_dmd1_dongle_loader/lib/share/libhi_loader.so
- 可执行文件：
\$(SDK_DIR)/out/hi3716mv430/hi3716mv430_dmd1_dongle_loader/obj/sample/loader/sample_abloader

如果客户需要，可以通过以下回调函数获取到进度信息：

```
hi_unf_upgrade_register(HI_UPGRADE_PROGRESS_DOWNLOAD, download_progress);
hi_unf_upgrade_register(HI_UPGRADE_PROGRESS_BURN, burn_progress);
```



U 盘方式升级

```
hi_unf_upgrade(1, "/mnt/usb_update.bin.sig", 64);
```

- 第一参数指定为1，表示使用U盘方式升级
- 第二个参数制定U盘中的升级包绝对路径
- 第三个参数为存储升级路径字符串buffer的长度扩展参数，数值必须大于0，小于128，如“64”

网络方式升级

```
hi_unf_upgrade(0, "http://192.168.100.8:8080/usb_update.bin.sig", 64);
```

- 第一参数指定为0，表示使用网络方式升级(HTTP协议)
- 第二个参数指定网络升级时用的URL地址
- 第三个参数为存储升级URL地址字符串buffer的长度，扩展参数，数值必须大于00，小于128，如“64”

注意

hi_unf_upgrade是阻塞接口，升级过程出错或者成功才能返回，通过返回值判断升级是否成功。

- 成功：HI_SUCCESS;
- 失败：HI_FAILURE;

3.6.5 升级方式

U 盘升级

将按3.6.3章节生成的usb_update.bin.sig拷贝到U盘，单板接上U盘，单板端执行以下步骤(sample_abloader在版本编译的时候会生成在obj/sample/loader目录下，可以事先获取到这个文件)：

```
1. mount /dev/sda /mnt
2. ./sample_abloader 1 /mnt/usb_update.bin.sig 64
```

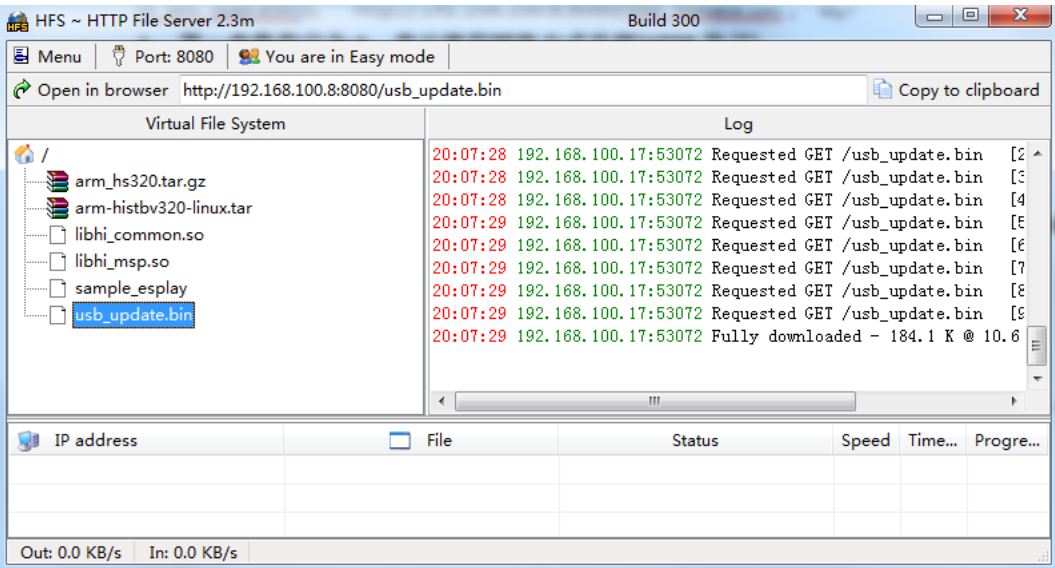
网络升级

测试时，需将投屏器与PC接入同一个局域网段，PC端通过hfs工具搭建一个HTTP服务器存放升级镜像，参考以下步骤：

步骤1 将usb_update.bin.sig加到hfs工具中；



图 3-11 升级文件加到 hfs 工具



步骤2 单板端执行命令： `./sample_loader 1 http://192.168.100.8:8080/usb_update.bin.sig 64`
----结束



4 工厂测试

4.1 方案介绍

4.1.1 方案介绍

SDK提供的参考厂测软件支持投屏器各输入输出接口的测试和SN的烧写。

集成了厂测软件的系统起来后会先检测deviceinfo分区中的SN，如果没有SN，则进入工厂测试程序，依次对WiFi、Bluetooth、音视频播放、Key/LED模块进行检测，检测成功后，通过USB扫描枪读取SN，厂测程序将SN烧录到单板Flash，全部成功则完成工厂测试。

如果需要调整厂测内容，可以在参考代码基础上自行修改。

4.1.2 准备工作

工厂需要提前准备路由器、蓝牙设备、电视机、USB接口的扫描枪、SN条码等。

路由器开启wifi、蓝牙设备把蓝牙功能打开，厂测软件会在测试过程中自动扫描wifi和蓝牙。

4.1.3 检测流程

4.1.3.1 模式检测

系统起来检测SN，如果没有SN，进入工厂检测，实现接口：

```
hi_s32 factory_detect_status_check();
```

返回值：

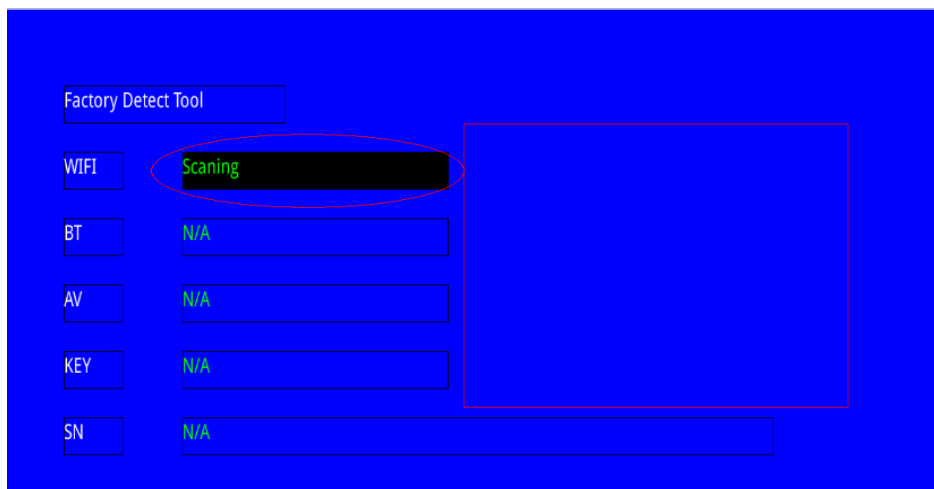
- -1: 获取信息失败
- 0: 已有SN，不需要工厂检测
- 1: 进入工厂检测

4.1.3.2 WiFi 检测

进入厂测程序后，程序会先扫描WiFi以测试WiFi功能是否正常，界面提示" scanning"。



图 4-1 WiFi 检测



- 如果搜索到WiFi设备，界面提示"PASS"，进入下一个功能检测；
- 如果未搜索到WiFi设备，界面提示"FAIL"，退出工厂检测；

4.1.3.3 Bluetooth 检测

进入Bluetooth检测后，程序会扫描以测试Bluetooth功能是否正常，界面提示"Scanning"。

- 如果搜索到Bluetooth设备，界面提示"PASS"，进入下一个功能检测；
- 如果未搜索到Bluetooth设备，界面提示"FAIL"，退出工厂检测；

4.1.3.4 音视频播放

进入播放功能测试后，程序会播放预置的节目。

文件路径：

```
#define FACTORY_PLAY_FILE_PATH    "/var/factory_test.avi"
```

- 如果播放失败，界面提示"FAIL"，退出工厂检测；
- 如果播放成功，界面提示"Press key to end play"

图 4-2 音视频播放





按下投屏器按键，会结束播放，界面提示"PASS"，进入下一个功能检测；

4.1.3.5 Key/LED 检测

Key/LED是和音视频播放一起检测的，播放过程中，会点亮LED；

通过触发按键key结束播放，同时熄灭LED；

音视频播放检测流程正常，Key/LED检测通过，界面提示"PASS"。

4.1.3.6 SN 烧写

功能测试完成后，程序会自动进入SN烧写环节。

步骤1 进入烧写模式，等待USB扫描枪连接并扫描输入SN，等待时间60秒钟；

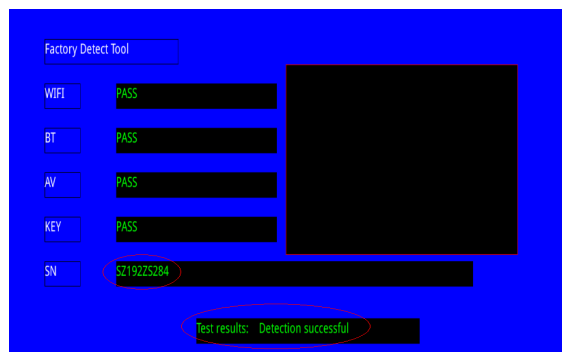
步骤2 输入SN之后，写到deviceinfo分区；

步骤3 如果烧写失败，界面提示"FAIL"，退出工厂检测；

步骤4 如果烧写成功，界面会显示已烧写的SN；工厂测试完成，如下所示。

----结束

图 4-3 SN 烧写



4.2 参考代码

路径：Code\sample\dongle_factory_detection



5 FAQ

5.1 常见问题

5.1.1 渐进式 JPEG 图片无法解析

问题描述

渐进式JPEG图片无法解析出来？

解决办法

渐进式JPEG图片解码需要耗费较大内存，需要128MB以上的版本才能支持。

5.1.2 苹果 Airplay 播放卡顿

问题描述

苹果Airplay播放卡顿？

解决办法

请确认是否由于P2P在后台抢占带宽导致，可以使用以下方法验证测试：

- 通过下述命令关闭P2P功能，同时观察卡顿现象是否得到改善：
`wpa_cli -ip2p0 -p /dev/wifi/wpa_supPLICANT P2P_STOP_FIND`
- 通过下述命令调节P2P开启时间，同时观察卡顿现象是否得到改善：
`wpa_cli -ip2p0 -p /dev/wifi/wpa_supPLICANT p2p_ext_listen 500 5000`

注意

参数中500和5000含义为在5000ms里面P2P功能开启500ms，其他时间关闭。



5.1.3 图片大小超过 8MB 的图片解码失败

问题描述

图片大小超过8M解析不出来？

解决办法

仅支持非渐进式JPEG图片，大于8M的JPEG图片采用流式解码方式，其它大于8M的图片类型均不支持，需要128MB以上的版本才能支持

5.1.4 手机拍摄的视频投屏播放失败

问题描述

手机拍摄的视频投屏失败，提示不支持的视频类型？

解决方法

在手机相机设置中选择FHD分辨率进行录制，高于FHD分辨率不支持。

5.1.5 手机拍摄的视频投屏播放卡顿

问题描述

手机拍摄的视频投屏播放不流畅，画面经常出现加载提示

解决方法

手机拍摄的视频码率很高，对网络带宽要求较高，2.4G热点情况下容易出现卡顿，要获得更好的观赏体验，请优先使用5G热点。

5.2 升级校验修改

5.3 接口使用说明

- 示例代码：
`$(SDK_DIR)/source/component/loader/ab_upgrade/source/main.c`
- 头文件：
`$(SDK_DIR)/source/component/loader/ab_upgrade/include/hi_unf_upgrade.h`
- 动态库：
`$(SDK_DIR)/out/hi3716mv430/hi3716mv430_dmd_dongle_loader/lib/share/libhi_loader.so`
- 可执行文件：
`$(SDK_DIR)/out/hi3716mv430/hi3716mv430_dmd_dongle_loader/rootbox/home/loader`

如果客户需要的话，通过以下回调函数获取到进度信息：



```
hi_unf_upgrade_register(HI_UPGRADE_PROGRESS_DOWNLOAD, download_progress);  
hi_unf_upgrade_register(HI_UPGRADE_PROGRESS_BURN, burn_progress);
```

U 盘方式升级

```
hi_unf_upgrade(1, "/mnt/usb_update.bin", 64);
```

- 第一参数指定为1，表示使用U盘方式升级
- 第二个参数制定U盘中的升级包绝对路径
- 第三个参数为存储升级路径字符串buffer的长度，数值必须大于0，小于128，如“64”

网络方式升级

```
hi_unf_upgrade(0, "http://192.168.100.8:8080/usb_update.bin", 64);
```

- 第一参数指定为0，表示使用网络方式升级(HTTP协议)
- 第二个参数指定网络升级时用的URL地址
- 第三个参数为存储升级URL地址字符串buffer的长度，，数值必须大于0，小于128，如“64”

注意

hi_unf_upgrade是阻塞接口，升级过程出错或者成功才能返回，通过返回值判断升级是否成功。

- 成功：HI_SUCCESS;
- 失败：HI_FAILURE;