

# Rockchip RK3328 软件开发指南

---

发布版本：1.3

作者邮箱：[markhuang@rock-chips.com](mailto:markhuang@rock-chips.com)、[steven.chan@rock-chips.com](mailto:steven.chan@rock-chips.com)

日期：2019-03-29

文件密级：公开资料

Copyright 2017 @Fuzhou Rockchip Electronics Co., Ltd.

---

## 前言

## 概述

本文档作为Rockchip RK3328软件开发指南，旨在帮助软件开发工程师、技术支持工程师更快上手RK3328的开发及调试。

## 产品版本

芯片名称	内核版本	Android版本
v1.0	linux 4.4	android 9.0

## 读者对象

软件工程师，硬件工程师，技术支持工程师

## 修订记录

日期	版本	作者	审核	修改说明
2018-11-18	v1.0	黄建财	陈伟、张小朱	初始版本
2017-12-20	v1.1	黄建财	陈伟	1. 添加system as root 新特性分区对比介绍 2. 添加U-boot optee rpmb配置说明; 3. 添加Widewine L1 功能配置说明.
2019-01-09	v1.2	黄建财	陈伟	使用markdown风格修订格式
2019-03-29	v1.3	黄建财	陈伟	1.添加常用配置宏说明 2.更新WideWine功能配置说明 3.添加 显示参数的调整和保存 4.添加系统调试- Lastlog开启 5.添加系统调试-串口调试 6.添加系统调试-Log自动保存系统abc.

## Rockchip RK3328 软件开发指南

### 1 支持列表

- 1.1 DDR支持列表
- 1.2 EMMC支持列表
  - 1.2.1 高性能EMMC颗粒的选取
- 1.3 WiFi/BT支持列表
- 1.4 SDK软件包适用硬件列表
- 1.5 多媒体编解码支持列表

### 2 文档/工具索引

- 2.1 文档索引
- 2.2 工具索引

### 3 SDK编译/烧写

- 3.1 SDK获取
- 3.2 SDK编译配置
  - 3.2.1 分区大小配置
  - 3.2.2 Android Pie 新增特性配置说明
  - 3.2.3 jack-server配置
  - 3.2.4 全自动编译脚本
- 1.3 量产烧写

### 4 U-Boot开发

- 4.1 Rockchip U-Boot nextdev简介
- 4.2 平台配置
- 4.3 固件生成
- 4.4 U-Boot编译
- 4.5 U-Boot logo相关的配置
  - 4.5.1 U-Boot logo开关配置
  - 4.5.2 U-Boot Logo图片更换
- 4.6 U-Boot OPTEE RPMB配置

### 5 内核开发常见配置

- 5.1 DTS介绍
  - 5.1.1 DTS说明
  - 5.1.2 新增一个产品DTS
- 5.2 WiFi&BT的配置
- 5.3 GPIO对应关系注意
- 5.4 ARM、GPU、DDR频率修改
- 5.5 温控配置
- 5.6 PWM IR配置
- 5.7 DDR频率修改说明
- 5.8 RK3328 DDR模板时序配置

## 6 Android开发常见配置

- 6.1 Android编译配置
  - 6.1.1 lunch选项说明
  - 6.1.2 添加一个新的产品
- 6.2常用配置宏说明
- 6.3 预置APK
- 6.4 开/关机动画
- 6.5 Parameter说明
- 6.6 新增分区配置
- 6.7 显示框架配置
- 6.8 OTA升级
  - 6.8.1 OTA 介绍
  - 6.8.2 生成完整包
  - 6.8.3 生成差异包
- 6.9 预制Demo
- 6.10 开机视频
- 6.11 低内存机器内存优化配置
- 6.12 DRM Widevine Level 1配置
- 6.13 媒体中心
- 6.14 TWRP recovery
- 6.15 Magisk Support
- 6.16 安全启动方案
- 6.17 Microsoft PlayReady
- 6.18 动态加载 UiMode
- 6.19显示参数的调整和保存

## 7 系统调试

- 7.1 ADB工具
  - 7.1.1 概述
  - 7.1.2 USB adb使用说明
  - 7.1.3 网络adb使用要求
  - 7.1.4 SDK网络adb端口配置
  - 7.1.5 网络adb使用
  - 7.1.6 手动修改网络adb端口号
  - 7.1.7 ADB常用命令详解
- 7.2 Logcat工具
  - 7.2.1 Logcat命令使用
  - 7.2.2 常用的日志过滤方式
  - 7.2.3 查看上次log
- 7.3 Procrank工具
  - 7.3.1 使用procrank
  - 7.3.2 检索指定内容信息
  - 7.3.3 跟踪进程内存状态

- 7.4 Dumpsys工具
  - 7.4.1 使用Dumpsys
- 7.5 音视频问题调试工具及文档
- 7.5 串口调试
  - 7.5.1 串口配置
  - 7.5.2 FIQ模式
- 7.7 Last log开启
- 7.8 Log自动保存系统 abc
  - 7.8.1 使用方法：

8 常用工具说明

- 8.1 StressTest
- 8.2 PCBA测试工具
- 8.3 DDR测试工具
- 8.4 Android开发工具
  - 8.4.1 下载镜像
  - 8.4.2 升级固件
  - 8.4.3 高级功能
- 8.5 update.img打包
- 8.6 固件签名工具
- 8.7 序列号/Mac/厂商信息烧写-WNpctool工具
  - 8.7.1 序列号获取
  - 8.7.2 WNpctool写入步骤
  - 8.7.3 WNpctool读取步骤
- 8.8 OemTool打包工具
  - 8.8.1 Oem打包工具步骤
- 8.9 量产工具使用
  - 8.9.1 工具下载步骤
- 8.10 Box厂测工具

# 1 支持列表

## 1.1 DDR支持列表

RK3328 DDR支持DDR3、DDR4、LPDDR3。

表 1-1 RK3328 DRAM Support Type

Chip	DRAM Support Type
RK3328	DDR3/DDR4/LPDDR3

RK3328 DDR颗粒支持程度列表，详见RKDocs\common\Platform support lists目录下《RK DDR Support List Ver2.31》，下表中所标示的DDR支持程度表，只建议选用√、T/A标示的颗粒。

表 1-2 RK3328 DDR Support Symbol

Symbol	Description
√	Fully Tested and Mass production
T/A	Fully Tested and Applicable
N/A	Not Applicable

## 1.2 EMMC支持列表

RK3328支持eMMC 4.51，SDIO3.0，支持HS200模式，详见RKDocs\common\Platform support lists目录下《RKeMMCSupportList Ver1.38\_2018\_01\_22》，下表中所标示的EMMC支持程度表，只建议选用√、T/A标示的颗粒。

表 1-3 RK3328 EMMC Support Symbol

Symbol	Description
√	Fully Tested , Applicable and Mass Production
T/A	Fully Tested , Applicable and Ready for Mass Production
D/A	Datasheet Applicable,Need Sample to Test
N/A	Not Applicable

### 1.2.1 高性能EMMC颗粒的选取

为了提高系统性能，需要选取高性能的EMMC颗粒。请在挑选EMMC颗粒前，参照Rockchip提供支持列表中的型号，重点关注下厂商Datasheet中performance一章节。

参照厂商大小以及EMMC颗粒读写的速率进行筛选。建议选取顺序读速率>200MB/s、顺序写速率>40MB/s。

如有选型上的疑问，也可直接联系Rockchip Fae窗口。

6.1.5 Performance

[Table 23] Performance

Density	Partition Type	Performance	
		Read(MB/s)	Write (MB/s)
16GB	General	285	40
32GB		310	70
64GB		310	140
128GB		310	140
16GB	Enhanced	295	80
32GB		320	150
64GB		320	245
128GB		320	245

图1-1 EMMC Performance示例

1.3 WiFi/BT支持列表

RK3328 Android8.1的Kernel版本为Linux4.4，WiFi/BT支持列表，详见RKDocs\common\Platform support lists 目录下《Rockchip\_WiFi\_Situation\_20170214》，文档列表中为目前RK3328上大量测试过的Wifi/Bt芯片列表，建议按照列表上的型号进行选型。如果有其他WiFi/BT芯片调试，需要WiFi/BT芯片原厂提供Linux4.4版本的内核驱动程序。

如果疑问和建议可以与Rockchip Fae窗口联系。

1.4 SDK软件包适用硬件列表

本SDK是基于谷歌Android9.0 64bit系统，适配瑞芯微RK3328芯片的软件包，适用于RK3328 Box Evb开发板、RK3328 Box样机板及基于其上所有的开发产品。

若是基于RK3328 Box Evb开发板开发，内核配置可参考rk3328-evb-android-avb.dts进行改动。

另SDK中附带了RK3328 Box Evb开发板及RK3328 Box 样机板的硬件使用说明。

表1-4 RK3328 硬件说明列表

硬件板	对应文档说明
Evb开发板	RKDocs\rk3328\RK3328 BOX EVB用户使用指南_V1.0_20170223
Box样机板	RKDocs\rk3328\RK3328 Box样机硬件设计指南_V1.0_20170223

1.5 多媒体编解码支持列表

RK3328多媒体规格:

- 支持4K VP9 and 4K 10bits H265/H264 视频解码，高达60fps;
- 1080P 多格式视频解码 (VC-1, MPEG-1/2/4, VP8)。支持JPEG解码;
- 1080P H.264/H.265格式视频编码。支持JPEG编码;
- 支持HDR10,HLG HDR ,支持SDR和HDR之间的转换。

RK3328具体的编解码支持列表，详见RKDocs\rk3328目录下《RK3328 Multimedia Codec Benchmark v1.0》。

## 2 文档/工具索引

### 2.1 文档索引

随RK3328 Box SDK发布的文档旨在帮助开发者快速上手开发及调试，文档中涉及的内容并不能涵盖所有的开发知识和问题。文档列表也正在不断更新，如有文档上的疑问及需求，请联系我们的Fae窗口。

RK3328 SDK的RKDocs目录结构如下所示。

RKDocs/

├── android

| ├── Android8.0\_OEM内容预置功能说明\_V1.0\_20171122.pdf

| ├── Android8.0定制开关机动画（铃声）说明V1.0\_20170923.pdf

| ├── Android8.0性能模式使用说明V1.0\_20170923.pdf

| ├── Android8.0恢复出厂设置保护功能说明V1.0\_20170923.pdf

| ├── Android8.0预安装应用功能说明文档V1.0\_20171109.pdf

| ├── Android8.0验证启动功能说明V1.0\_20171109.pdf

| ├── Android 增加一个分区配置指南 V1.00.pdf

| ├── bt

| | └── ROCKCHIP\_ANDROID\_8.1\_BT配置说明\_V1.0\_20180103.pdf

| ├── project.config

| ├── RK\_PCBA\_Camera移植说明v1.0.pdf

| ├── Rockchip Android 8.1 BOX显示框架配置说明文档 V1.0-20180210.pdf

| ├── Rockchip Box媒体中心使用说明-v1.0.1-20170216.pdf

| ├── ROCKCHIP\_PCBA测试工具开发指南\_V1.2\_20180509.pdf

| ├── Rockchip Recovery用户操作指南V1.03.pdf

| └── wifi

| ├── RealTek wifi 驱动移植说明\_V1.1.pdf

| └── ROCKCHIP\_ANDROID\_9.0\_WIFI配置说明\_V1.4.pdf

├── common

| └── camera

- | | |— Camera目录文档说明.txt
- | | |— CIF\_ISP10\_Driver\_User\_Manual\_V1.0\_20171124.pdf
- | | |— CIF\_ISP11\_Driver\_User\_Manual\_V1.0.pdf
- | | |— readme\_En.txt
- | | |— RK312x\_Camera\_User\_Manual\_v1.4(适用3288&3368).pdf
- | | |— RK\_ISP10\_Camera\_User\_Manual\_v2.2.pdf
- | | |— RKISPV1\_Camera\_Module\_AVL\_v1.7.pdf
- | | |— RKISPV1\_Camera\_常见问题解决方法V1.0.pdf
- | | |— RKISPV1\_Camera\_驱动调试方法V1.0.pdf
- | | |— Rockchip\_Camera\_AVL\_v2.0\_Package\_20180515.7z
- | | |— Rockchip SOFIA 3G-R\_PMB8018(x3\_C3230RK)\_Camera\_Module\_AVL\_v1.6\_20160226.pdf
- | |— DDR
- | | |— DDR开发指南.pdf
- | | |— DDR问题排查手册.pdf
- | |— debug
- | | |— perf使用说明.pdf
- | | |— RK3399-LOG-EXPLANATION.pdf
- | | |— streamline使用说明.pdf
- | | |— systrace使用说明.pdf
- | |— display
- | | |— rockchip\_drm\_integration\_helper-zh.pdf
- | | |— Rockchip\_DRM\_Panel\_Porting\_Guide\_V1.5\_20180830.pdf
- | | |— Rockchip 基于DRM框架的HDMI开发指南 v1.1-20180322.pdf
- | | |— 基于DRM的Android显示使用指南\_V1.0\_20180129.pdf
- | |— driver
- | | |— RK817\_RK809\_Codec 开发指南\_V1.0\_20180228.pdf
- | | |— RK语音通话3A算法集成说明及参数调试说明文档\_V3.0.pdf
- | | |— Rockchip Audio 开发指南 V1.1-20170215-linux4.4.pdf
- | | |— Rockchip CPU-Freq开发指南 V1.0.1-20170213.pdf
- | | |— Rockchip-Developer-Guide-linux4.4-PCIe.pdf
- | | |— Rockchip-Developer-Guide-linux4.4-SDMMC-SDIO-eMMC.pdf
- | | |— Rockchip-Developer-Guide-linux4.4-USB.pdf



- | | | — Rockchip-Developer-Guide-MCU.pdf
- | | | — Rockchip-Developer-Guide-SPI.pdf
- | | | — Rockchip-Developer-Guide-UART.pdf
- | | | — Rockchip DEVFreq 开发指南 V1.0-20160701.pdf
- | | | — Rockchip gmac模块 开发指南V1.0-20170221.pdf
- | | | — Rockchip I2C 开发指南 V1.0-20160629.pdf
- | | | — Rockchip IO-Domain 开发指南 V1.0-20160630.pdf
- | | | — Rockchip Pin-Ctrl 开发指南 V1.0-20160725.pdf
- | | | — Rockchip pwm ir开发指南V1.00.pdf
- | | | — Rockchip pwm背光 开发指南-20170220.pdf
- | | | — Rockchip RK805开发指南 V1.0-20170217.pdf
- | | | — Rockchip RK816开发指南 V1.pdf
- | | | — Rockchip RK818\_6电量计 开发指南 V2.0-20170525mo.pdf
- | | | — Rockchip RK818电量计 开发指南 V1.0-20160725.pdf
- | | | — Rockchip\_Sensors开发指南V1.0\_20180605.pdf
- | | | — Rockchip Thermal 开发指南 V1.0.1-20170428.pdf
- | | | — Rockchip Vendor Storage Application Note.pdf
- | | | — Rockchip 以太网 开发指南 V2.3.1-20160708.pdf
- | | | — Rockchip 休眠唤醒 开发指南 V0.1-20160729.pdf
- | | | — Rockchip 时钟子模块 开发指南 V1.1-20170210.pdf
- | | | — Rockchip 电源 独立DCDC 开发指南 V1.0-20170519.pdf
- | | — hdmi-in
- | | | — HDMI\_IN开发指南V1.0\_20180726.pdf
- | | — mobile-net
- | | | — 3G数据卡USB切换文件制作说明\_v1.2.pdf
- | | | — ROCKCHIP\_3G\_DONGLE配置说明V1.0.pdf
- | | — Platform support lists
- | | | — RK3128 BOX Hardware Design Guide V10-201410.pdf
- | | | — RKeMMCSupportList Ver1.38\_2018\_01\_22.pdf
- | | | — RKISPV11\_Camera\_Module\_AVL\_v1.5.pdf
- | | | — RKISPV1\_Camera\_Module\_AVL\_v1.7.pdf
- | | | — RKNandFlashSupportList Ver2.72\_2016\_08\_30.pdf

- | | |— RK DDR Support List Ver2.31.pdf
- | | |— Rockchip Kodi支持程度列表\_V2.0\_20170715.pdf
- | | |— Rockchip\_WiFi\_Situation\_20180611.pdf
- | |— RKTools manuals
- | | |— Android开发工具手册.pdf
- | | |— REPO镜像服务器搭建和管理\_V2.2\_20131231.pdf
- | | |— RKUpgrade\_Dll\_UserManual.pdf
- | | |— RK平台apache\_tomcat\_ota服务器搭建说明.rar
- | | |— rk平台量产升级指导文档V1.1.pdf
- | | |— RockChip Box厂测工具V2.0.rar
- | | |— Rockchip Box厂测工具操作说明V2.0.pdf
- | | |— Rockchip Keybox Burning Guide V1.2-20180315.pdf
- | | |— Rockchip Parameter File Format Ver1.3.pdf
- | | |— Rockchip 量产烧录 指南 V1.1-20170214.pdf
- | | |— WNPctool写号工具简要使用说明\_V1.1.2.pdf
- | | |— 压力测试Stresstest文档 forVR\_ver3.0.pdf
- | | |— 瑞芯微KeyWrite使用指南\_V1.3\_20180508.pdf
- | | |— 量产工具升级及相关问题处理.pdf
- | |— security
- | | |— Rockchip-Secure-Boot-Application-Note-V1.9.pdf
- | | |— Rockchip\_TEE安全SDK开发手册\_V1.1\_20170516.pdf
- | |— u-boot
- | | |— Rockchip-Developer-Guide-Trust.pdf
- | | |— Rockchip-Developer-Guide-UBoot-nextdev.pdf
- | | |— Rockchip U-Boot 开发指南 V3.8-20170214.pdf
- | |— usb
- | | |— RK USB Compliance Test Note V1.2.1.pdf
- | | |— Rockchip-Developer-Guide-linux4.4-USB.pdf
- | | |— Rockchip-USB-Performance-Analysis-Guide.pdf
- | | |— Rockchip-USB-SQ-Test-Guide.pdf
- |— rk3328
- |— RK3328 BOX EVB硬件使用指南V1.0\_20170223.pdf

- └─ RK3328 Box样机硬件设计指南\_V1.0\_20170223.pdf
- └─ RK3328 Multimedia Codec Benchmark v1.0.pdf
- └─ kernel-4.4 rk3328 ddr模板时序配置说明.pdf
- └─ Rockchip RK3328 Android9.0-Box软件开发指南 V1.02-20190130.pdf

## 2.2 工具索引

随RK3328 Box SDK发布的工具，用于开发调试阶段及量产阶段。工具版本会随SDK更新不断更新，如有工具上的疑问及需求，请联系我们的Fae窗口。

RK3328 SDK中在RKTools目录下附带了linux（Linux操作系统环境下使用工具）、windows（Windows操作系统环境下使用工具）。

表2-1 工具索引表格

工具名称	工具说明	工具路径
AndroidTool	分立升级固件及整个update升级固件工具	RKTools\windows\AndroidTool_Release_v2.59
FactoryTool	量产升级工具	RKTools\windows\FactoryTool_v1.63
SecureBootTool	固件签名工具	RKTools\windows\SecureBootTool_v1.85_foruser
efuseTool	efuse烧写工具	RKTools\windows\efuse_v1.37
WNpctool	写号工具	RKTools\windows\WNpctool_Setup_V1.1.9_180118
SD_Firmware_Tool	SD卡镜像制作	RKTools\windows\SD_Firmware_Tool_v1.46
SpilimageTools	烧录器升级工具	RKTools\windows\SpilimageTools_v1.36
DriverAssitant	驱动安装工具	RKTools\windows\DriverAssitant_v4.5
OemTool	新增分区镜像制作工具	RKTools\windows\OemTool_v1.3
Rockchip平台DDR测试工具	DDR测试工具	RKTools\windows\Rockchip平台DDR测试工具_V1.35
Rockchip Box厂测工具	厂测工具	RKTools\windows\Rockchip Box厂测工具V2.0-M-20170327
Linux_Pack_Firmware	Linux打包工具	RKTools\linux\Linux_Pack_Firmware
Linux_SecureBoot	Linux签名工具	RKTools\linux\Linux_SecureBoot
Linux_Upgrade_Tool	Linux烧写工具	RKTools\linux\Linux_Upgrade_Tool

## 3 SDK编译/烧写

## 3.1 SDK获取

SDK通过瑞芯微代码服务器对外发布。客户向瑞芯微技术窗口申请SDK，需同步提供SSH公钥进行服务器认证授权，获得授权后即可同步代码。关于瑞芯微代码服务器SSH公钥授权，请参考《RK3328\_ANDROID9.0-BOX-SDK\_V1.00\_20181206发布说明.pdf》，该文档与SDK一同发布。

## 3.2 SDK编译配置

### 3.2.1 分区大小配置

SDK默认分区表一般定义在对应产品目录下的parameter.txt文件中（例如：rk3328\_box 对应 device/rockchip/rk3328/rk3328\_box/parameter.txt）。

BOARD\_SYSTEMIMAGE\_PARTITION\_SIZE等系统分区宏配置会自动读取分区表中分区大小赋值。

若需要调整分区大小，修改对应分区表定义文件parameter.txt即可。具体参见[6.4章节parameter说明](#)。

**注意：Android Pie 要求新增部分分区及说明如下：**

dtb分区：用来存放device tree blob的镜像，预留，默认不烧；

dtbo分区：用来存放device tree blob overlay 的镜像，默认烧对应产品目录下的预置镜像dtbo.img；

vbmeta分区：用来存放Android 验证启动（AVB）模式下编译自动生成的各分区校验数据，由于目前sdk已升级到支持vboot2.0，默认烧写时必须烧vbmeta.img。

### 3.2.2 Android Pie 新增特性配置说明

默认Android9.0 要求的新增特性配置修改在sdk源码device/rockchip/common/BoardConfig.mk中：

#### 1.要求默认使用64bit binder驱动

```
#binder protocol(8)

TARGET_USES_64_BIT_BINDER := true
```

#### 2.默认cmdline参数配置在产品配置中修改，不在parameter中了，如常见的selinux权限配置等。

```

ifneq ((filter true, (BOARD_AVB_ENABLE)), )
BOARD_KERNEL_CMDLINE := console=ttyFIQ0 androidboot.baseband=N/A androidboot.selinux=permissive
androidboot.wificountrycode=US androidboot.hardware=rk30board androidboot.console=ttyFIQ0
firmware_class.path=/vendor/etc/firmware init=/init skip_initramfs rootwait ro init=/init
else
#Config the cmdline for boot or recovery
BOARD_KERNEL_CMDLINE := console=ttyFIQ0 androidboot.baseband=N/A androidboot.selinux=permissive
androidboot.wificountrycode=US androidboot.verifymode=enforcing androidboot.hardware=rk30board
androidboot.console=ttyFIQ0 firmware_class.path=/vendor/etc/firmware init=/init skip_initramfs
rootwait ro init=/init root=PARTUUID=af01642c-9b84-11e8-9b2a-234eb5e198a0
endif

```

3.要求默认开启system as root功能，主要变动为将原ramdisk.img整合到system.img中，启动时system作为rootfs挂载，如下图为分区镜像差异。

```
BOARD_BUILD_SYSTEM_ROOT_IMAGE := true
```

1548836787728

详细介绍参见谷歌开发者网站system as root说明。

4.要求默认开启VNDK检测及属性兼容性配置，具体参见谷歌开发者网站中VNDK说明。

```

# Enable VNDK Check for Android P (MUST in P)
BOARD_VNDK_VERSION := current
PRODUCT_COMPATIBLE_PROPERTY_OVERRIDE := true

```

5.是否配置AVB(android verified boot)功能，具体参见谷歌开发者网站中AVB功能说明。

目前sdk中ATV产品默认要求开启，BOX产品默认关闭。

```

# Enable android verified boot 2.0
BOARD_AVB_ENABLE ?= false

```

6.要求userdebug模式下也必须开启DEXPREOPT优化，提升开机速度和应用第一次启动速度。

**注意：**开启DEXPREOPT优化对开发阶段debug时push apk和jar文件操作有影响，直接push无效。

若需要临时版本可以关闭此开关，参考下面修改配置：

**副作用：**由于art中会校验镜像，关闭后每次开机时间会很长。

```

WITH_DEXPRESOPT ?= true
diff --git a/BoardConfig.mk b/BoardConfig.mk
index 8543697..cb81367 100644
--- a/BoardConfig.mk
+++ b/BoardConfig.mk

```

```

@@ -167,18 +167,6 @@ TARGET_PROVIDES_INIT_RC ?= false
//MAX-SIZE=512M, for generate out/.../system.img
BOARD_FLASH_BLOCK_SIZE := 131072
+WITH_DEXPREEPT_BOOT_IMG_AND_SYSTEM_SERVER_ONLY ?= true
+
+## Enable dex-preoptimization to speed up first boot sequence
+ifeq ($(HOST_OS),linux)
+  ifeq ($(TARGET_BUILD_VARIANT),user)
+    ifeq ($(WITH_DEXPREEPT),)
+      WITH_DEXPREEPT ?= true
+    endif
+  else
+    WITH_DEXPREEPT ?= false
+  endif
+endif
ART_USE_HSPACE_COMPACT ?= true

```

### 3.2.3 jack-server配置

Android8.1以后系统使用jack-server作为java代码编译器，在编译过程中可能会遇到以下类似的错误：

```

Jack server already installed in "/home/yhx/.jack-server"
Communication error with Jack server (1), try 'jack-diagnose' or see Jack server log
Communication error with Jack server 1. Try 'jack-diagnose'
Communication error with Jack server 1. Try 'jack-diagnose'

```

这种情况主要是由于jack-server本身编译器限制，同一个网络端口号不能多个用户同时使用。

也就是在服务器上协同开发过程中，多用户同时编译Android8.1/9.0时，需要配置各自使用不同的网络端口号。

jack-server的两个配置文件，决定了它所使用的端口号：

```

~/ .jack-server/config.properties
~/ .jack-settings

```

这两个配置文件需要配置两个端口号，分别为服务端端口号，及客户端端口号，两个配置文件中的端口号要匹配。

```

jack.server.service.port=8074
jack.server.admin.port=8075

```

及

```

SERVER_PORT_SERVICE=8074
SERVER_PORT_ADMIN=8075

```

配置步骤如下：

1) 确保两个配置文件存在，并且权限设置为0600：

```
chmod 0600 ~/.jack-server/config.properties
chmod 0600 ~/.jack-settings
```

2) 若两个配置文件不存在，请参照以下文本新建这两个配置文件。

config.properties文件示例如下（端口号需按实际修改）：

```
jack.server.max-jars-size=104857600
jack.server.max-service=4
jack.server.service.port=8074
jack.server.max-service.by-mem=1\=2147483648\:2\=3221225472\:3\=4294967296
jack.server.admin.port=8075
jack.server.config.version=2
jack.server.time-out=7200
```

.jack-settings文件示例如下（端口号需按实际修改）：

```
#Server settings
SERVER_HOST=127.0.0.1
SERVER_PORT_SERVICE=8074
SERVER_PORT_ADMIN=8075
#Internal, do not touch
SETTING_VERSION=4
```

3) 修改端口号，请更改service port及admin port为其他端口号，两个配置文件里的端口号需要匹配。示例如下：

```
jack.server.service.port=8023
jack.server.admin.port=8024

SERVER_PORT_SERVICE=8023
SERVER_PORT_ADMIN=8024
```

4) 重新编译Android，看是否会报错，若依然报错，请尝试更改其他端口号，直至编译通过。

5) 若更改5次编译依然无法通过，可以执行jack-admin dump-report命令，解压命令生成的压缩包，分析log日志，若出现以下log，可以重新安装下libcurl：

```
$ JACK_EXTRA_CURL_OPTIONS=-v jack-admin list server
* Protocol https not supported or disabled in libcurl
* Closing connection -1
Communication error with Jack server 1. Try 'jack-diagnose'
```

### 3.2.4 全自动编译脚本

为了提高编译的效率，降低人工编译可能出现的误操作，该SDK中集成了全自动化编译脚本，方便固件编译、备份。

1) 该全自动化编译脚本原始文件存放于：

```
device/rockchip/rk3328/build_box.sh
```

2) 在repo sync的时候，通过manifest中的copy选项拷贝至工程根目录下：

```
<project path="device/rockchip/rk3328" name="rk/device/rockchip/rk3328" remote="rk"
revision="rk33/mid/8.0/develop">

  <copyfile src="buildspec_box.mk" dest="buildspec.mk"/>

  <copyfile src="build_box.sh" dest="build_box.sh"/>

</project>
```

3) 修改build.sh脚本中的特定变量以编出对应产品固件。

```
KERNEL_DTS=rk3328-evb-android-avb
```

变量请按实际项目情况，对应修改：

KERNEL\_DTS变量指定编译kernel的产品板级配置，如使用RK3328 Box 样机配置，则可将改变量改为：rk3328-box。

Android默认编译为rk3328\_box-userdebug模式，也可在脚本中对应修改，可改为rk3328\_box-user及其它配置：

```
lunch rk3328_box-user
```

4) 指定update.img打包用的loader：

如RKTools\linux\Linux\_Pack\_Firmware\rockdev\mkupdate.sh脚本所示：

img

Windows打包脚本（RKTools\windows\AndroidTool\rockdev\mkupdate.bat）也是类似，如下所示：

img

update.img打包用的loader被命名位MiniLoaderAll.bin，由于SDK更新兼容Loader，所以在此通过u-boot目录编译生成rk3328\_loader\_v1.06.238.bin（拷贝时会重命名为MiniLoaderAll.bin），需要指定脚本中loader文件名。

5) 执行自动编译脚本：

```
source build_box.sh
```



该脚本会自动配置JDK环境变量，编译u-boot，编译kernel，编译Android，继而生成固件，并打包成update.img。

## 6) 脚本生成内容:

脚本会将编译生成的固件拷贝至：

`IMAGE/RK3328-EVB-ANDROID-AVB_9_*****_RELEASE_TEST/IMAGES` 目录下，具体路径以实际生成为准。每次编译都会新建目录保存，自动备份调试开发过程的固件版本，并存放固件版本的各类信息。

该目录下的update.img可直接用于Android开发工具及工厂烧写工具下载更新。

## 1.3 量产烧写

量产上考虑到生产效率及工厂工位安排，量产烧写说明详见RKDocs\common\RKTools manuals目录下《Rockchip 量产烧录 指南 V1.1-20170214》。

在量产过程中如涉及到工具上的问题，可以联系我们的Fae窗口。

# 4 U-Boot开发

本节简单介绍U-Boot基本概念和编译的注意事项，帮助客户了解RK平台U-Boot框架，具体U-Boot开发细节可参考RKDocs\common\u-boot目录下《Rockchip-Developer-Guide-UBoot-nextdev.pdf》。

## 4.1 Rockchip U-Boot nextdev简介

next-dev是Rockchip从U-Boot官方的v2017.09正式版本中切出来进行开发的版本。目前在该平台上已经支持RK所有主流在售芯片。

**目前支持的功能主要有：**

- 支持RK Android平台的固件启动；
- 支持最新Android AOSP(如GVA)固件启动；
- 支持Linux Distro固件启动；
- 支持Rockchip miniloader和SPL/TPL两种pre-loader引导；
- 支持LVDS、EDP、MIPI、HDMI等显示设备；
- 支持Emmc、Nand Flash、SPI Nand flash、SPI NOR flash、SD卡、U盘等存储设备启动；
- 支持FAT、EXT2、EXT4文件系统；
- 支持GPT、RK parameter分区格式；
- 支持开机logo显示、充电动画显示，低电管理、电源管理；
- 支持I2C、PMIC、CHARGE、GUAGE、USB、GPIO、PWM、GMAC、EMMC、NAND、中断等驱动；
- 支持RockUSB 和 Google Fastboot两种USB gadget烧写EMMC；
- 支持Mass storage, ethernet, HID等USB设备；
- 支持使用kernel的dtb；
- 支持dtbo功能；

U-Boot的doc目录下提供了很丰富的README文档，它们向开发者介绍了U-Boot里各个功能模块的概念、设计理念、实现方法等，建议读者好好利用这些文档提高开发效率。

## 4.2 平台配置

平台配置文件位于U-Boot根目录下的configs文件夹下，其中Rockchip相关的以RK开头：

如RK3328H Box开发调试选用的是rk3328配置。

## 4.3 固件生成

Rockchip平台支持MiniLoader，固件支持所有的存储设备，根据不同的平台配置生成相应的Loader固件。同时引入Arm Trusted Firmware后会生成trust image。

以RK3328编译生成的镜像为例：

其中249是发布的版本号，rockchip定义U-Boot loader的版本，其中249是根据存储版本定义的，客户务必不要修改这个版本。

uboot.img是U-Boot作为二级loader的打包。

trust.img是trust fireware的打包镜像。

## 4.4 U-Boot编译

RK3128H Box SDK编译使用的是如下配置：

```
./make.sh rk3328
```

编译完，会生成trust.img、rk3328\_loader\_v1.13.249.bin、uboot.img三个文件。

## 4.5 U-Boot logo相关的配置

### 4.5.1 U-Boot logo开关配置

Sdk默认开启U-Boot logo功能，以达到更快显示开机logo的目的，见kernel/arch/arm64/boot/dts/rockchip/rk3328-android.dtsi 中如下配置：

```
&display_subsystem {
    logo-memory-region = <&drm_logo>;
    secure-memory-region = <&secure_memory>;
    status = "okay";
    route {
        route_hdmi: route-hdmi {
            status = "okay";
            logo,uboot = "logo.bmp";
            logo,kernel = "logo_kernel.bmp";
            logo,mode = "center";
            charge_logo,mode = "center";
            connect = <&vop_out_hdmi>;
        };

        route_tve: route-tve {
            status = "okay";
            logo,uboot = "logo.bmp";
            logo,kernel = "logo_kernel.bmp";
            logo,mode = "center";
            charge_logo,mode = "center";
        };
    };
};
```

```
connect = <&vop_out_tve>;  
};  
};  
};
```

如果需要关闭该功能，请将上述的dts文件中改为status = “disabled”。

## 4.5.2 U-Boot Logo图片更换

U-boot logo显示的两张图片是kernel根目录下的logo.bmp和logo\_kernel.bmp，如果需要更换，用同名的bmp替换掉，重新编译内核即可。

附：logo替换不一定要两张图片，可以只要一张，如果只有开发者手上只有一张logo图片，就保留logo.bmp这一张即可。

附：开机Logo图片大小目前只支持到8M以内大小的bmp格式图片，支持8、16、24、32位的bmp。

## 4.6 U-Boot OPTEE RPMB配置

RPMB ( Replay Protected Memory Block ) Partition 是 eMMC 中的一个具有安全特性的分区。

当机器硬件flash采用EMMC的情况，目前U-boot安全部分默认配置optee 使用rpmb，其主要存储安全相关的key，和AVB(android verified boot)相关的值。

**注意：**若客户机器硬件EMMC物料本身rpmb区域已被编程过（如写过非rk密钥）等非新料的情况，会导致开机正常安全代码引导流程异常，导致无法开机，则需要切换为非rpmb方式处理。

可通过下面修改切换成非rpmb配置：

```
diff --git a/configs/rk3328_defconfig b/configs/rk3328_defconfig  
index 0a3d1a3..5ee8c8f 100644  
--- a/configs/rk3328_defconfig  
+++ b/configs/rk3328_defconfig  
@@ -130,4 +130,5 @@ CONFIG_AVB_LIBAVB_USER=y  
CONFIG_RK_AVB_LIBAVB_USER=y  
CONFIG_OPTEE_CLIENT=y  
CONFIG_OPTEE_V1=y  
+CONFIG_OPTEE_ALWAYS_USE_SECURITY_PARTITION=y  
CONFIG_TEST_ROCKCHIP=y
```

# 5 内核开发常见配置

本节简单介绍内核一些常见配置的修改，主要是dts的配置，帮助客户更快更方便的进行一些简单的修改。RK3328 kernel版本是4.4，config配置文件统一为arch/arm64/configs/ rockchip\_defconfig 。RK3328的串口波特率为1500000，调试时请保证设置准确。

## 5.1 DTS介绍

### 5.1.1 DTS说明

RK3328的dts文件在kernel/arch/arm64/boot/dts/rockchip/下，如RK3328 evb评估板的dts文件为rk3328-evb-android-avb.dts。产品的dts里需根据具体的产品需求配置CPU、GPU、DDR的频率和电压表；配置io、wifi、bt、温控、电配置等等。

请各位开发者尽量以SDK发布的示例产品dts文件做参考，进行后期的开发。

### 5.1.2 新增一个产品DTS

RK3328的产品dts文件需放在kernel/arch/arm64/boot/dts/rockchip下，

1、以rk3328-evb-android.dts为参照，拷贝一份dts文件命名为rk3328-product.dts。

2、修改arch/arm64/boot/dts/Makefile文件，添加对应dtb申明

```
+rk3328-product.dtb
```

3、修改编译脚本或编译命令。

4、重新编译内核。

## 5.2 WiFi&BT的配置

RK3328 Android 9.0平台上WiFi、BT可做到自动兼容，按照RK提供的编译Android8.1编译步骤，生成固件后，默认就可以支持相应的WiFi模块,并且一套固件可以支持多个WiFi模块。目前rk3328 android 8.1平台wifi、bt 模块 android 和 kernel无需做任何配置。

## 5.3 GPIO对应关系注意

关于原理图上的gpio跟dts里面的gpio的对应关系，这边有个需要注意的地方：例如GPIO4\_C0，那么对应的dts里面应该是“gpio4 16”。GPIO 分为4个端口PORTA ( 0-7 )、PORTB ( 8-15 )、PORTC ( 16-23 )、PORTD ( 24-31 )，每个PORT有8个PIN，以此计算可得C0是16，C1口是17，以次类推。

GPIO的使用请参考RKDocs\common\driver目录下《Rockchip Pin-Ctrl 开发指南 V1.0-20160725.pdf》

## 5.4 ARM、GPU、DDR频率修改

DVFS ( Dynamic Voltage and Frequency Scaling ) 动态电压频率调节，是一种实时的电压和频率调节技术。目前4.4内核中支持DVFS的模块有CPU、GPU、DDR。CPU使用cpufreq框架，GPU和DDR使用devfreq框架。

CPUFreq是内核开发者定义的一套支持动态调整CPU频率和电压的的框架模型。它能有效的降低CPU的功耗，同时兼顾CPU的性能。

CPUFreq通过不同的变频策略，选择一个合适的频率供CPU使用，目前的内核版本提供了以下几种策略：

- interactive：根据CPU负载动态调频调压；
- conservative：保守策略，逐级调整频率和电压；
- ondemand：根据CPU负载动态调频调压，比interactive策略反应慢；
- userspace：用户自己设置电压和频率，系统不会自动调整；
- powersave：功耗优先，始终将频率设置在最低值；
- performance：性能优先，始终将频率设置为最高值。

详细的模块功能及配置，请参考RKDocs\common\driver目录下《Rockchip CPU-Freq开发指南 V1.0.1-20170213.pdf》

DEVFreq 是内核开发者定义的一套支持动态调整设备频率和电压的的框架模型。它能有效的降低该设备的功耗，同时兼顾其性能。目前我们的平台，有 GPU 和 DDR 在使用 DEVFreq。DEVFreq 通过不同的变频策略，选择一个合适的频率供设备使用，目前的内核版本提供了以下几种策略：

- Simple Ondemand：根据负载动态调频调压；
- Userspace：用户自己设置电压和频率，系统不会自动调整；
- Powersave：功耗优先，始终将频率设置在最低值；
- Performance：性能优先，始终将频率设置为最高值；
- Dmc Ondemand：我司实现的ddr变频策略，支持负载和场景变频；

GPU默认使用的是Simple Ondemand负载变频，DDR默认使用DMC Ondemand的变频策略是RK自己实现的。

## 5.5 温控配置

---

在 Linux 内核中，定义一套温控框架 linux Generic Thermal Sysfs Drivers，它可以通过不同的策略控制系统的温度，目前常用的有以下几种策略：

- power\_allocator：引入 PID（比例-积分-微分）控制，根据当前温度，动态给各模块分配 power，并将 power 转换为频率，从而达到根据温度限制频率的效果。
- step\_wise：根据当前温度，逐级限制频率。
- userspace：不限制频率

详细的模块功能及配置，请参考RKDocs\common\driver目录下《Rockchip Thermal 开发指南 V1.0.1-20170428》

## 5.6 PWM IR配置

---

红外遥控的发射电路是采用红外发光二极管来发出经过调制的红外光波；红外接收电路由红外接收二极管、三极管或硅光电池组成，它们将红外发射器发射的红外光转换为相应的电信号，再送后置放大器。鉴于家用电器的品种多样化和用户的使用特点，生产厂家对进行了严格的规范编码，这些编码各不相同，从而形成不同的编码方式，统一称为红外遥控器编码传输协议。目前RK 平台只支持NEC 编码的红外协议。

RK3328平台详细的遥控器适配，键值添加，红外按键定义，及遥控器功能相关调试内容请参考RKDocs\common\driver目录下《Rockchip pwm ir开发指南V1.00》。

## 5.7 DDR频率修改说明

---

请参考RKDocs\common\DDR目录下《DDR开发指南》如何修改ddr频率章节。

## 5.8 RK3328 DDR模板时序配置

---

请参考RKDocs\rk3328目录下《kernel-4.4 rk3328 ddr模板时序配置说明》文档说明。

---

# 6 Android开发常见配置

---

本节简单介绍Android开发中一些常见配置的修改，RK3328平台搭载的是最新的Android9.0系统。

## 6.1 Android编译配置

### 6.1.1 lunch选项说明

```
rk3328_box-userdebug : rk3328平台box产品userdebug ( 64位 )
```

```
rk3328_box-user : rk3328平台box产品user ( 64位 )
```

User版本开启selinux权限校验，。开发过程中涉及到apk及jar的更新，log打印调试相对麻烦很多。

建议开发调试阶段默认选择userdebug编译。

### 6.1.2 添加一个新的产品

各开发厂商可能有同款芯片不同产品开发的需求，一套SDK需同时编译生成多款产品固件。

RK3328平台支持Box类型各种产品形态，当需要添加一个新的产品时，可以基于已有的rk3328\_box来建立，如下以建立一个新的平板产品为例进行说明，具体步骤为：

1 ) 产品命令规则：

Box产品名中需带有“box”字样；

请务必遵守以上规则，否则系统会异常。

2 ) 新增文件夹device/rockchip/rk3328/rk3328\_box\_000，基于rk3328\_box.mk创建rk3328\_box\_000.mk，将rk3328\_box目录下的所有文件拷贝至rk3328\_box\_000目录下。

```
cd device/rockchip/rk3328
mkdir rk3328_box_000
cp rk3328_box.mk ./ rk3328_box_000.mk
cp rk3328_box/* rk3328_box_000/
```

3 ) 在device/rockchip/rk3328/AndroidProducts.mk中添加：

```
PRODUCT_MAKEFILES := \
    $(LOCAL_DIR)/rk3328.mk \
    $(LOCAL_DIR)/rk3328_box.mk \
    $(LOCAL_DIR)/rk3328_box_000.mk \
```

4 ) 在vendorsetup.sh中添加产品对应的lunch选项：

```
add_lunch_combo rk3328_box-eng
add_lunch_combo rk3328_box-userdebug
add_lunch_combo rk3328_box-user
add_lunch_combo rk3328_box_000-userdebug
add_lunch_combo rk3328_box_000-user
```

5 ) 修改rk3328\_box\_000.mk及rk3328\_box\_000目录下的新产品所需要修改的配置。

6 ) 修改编译脚本或编译命令，重新lunch产品名称进行新产品编译。

## 6.2常用配置宏说明

宏配置	功能说明
BUILD_WITH_GOOGLE_MARKET	若为 true 则集成 GMS 包，false 不集成
BUILD_BOX_WITH_GOOGLE_MARKET	若为 true 集成 BOX 的 GMS 包，false 不集成
BOARD_WIDEVINE_OEMCRYPTO_LEVEL	集成 Widevine 插件库，1位level1，3为level3
BOARD_CAMERA_SUPPORT_EXT	若为 true支持usb camera组件，fale不集成
BOARD_ALLOW_ROOTSERVICE	若为 true支持root服务（support su方案），false不支持root
BUILD_WITH_GTVS	若为 true则集成GTVS（android tv）包，false不集成
BUILD_WITH_GOOGLE_FRP	使能恢复出厂设置保护 FRP 功能
BUILD_WITH_MICROSOFT_PLAYREADY	使能playready插件功能
BOARD_USE_SPARSE_SYSTEM_IMAGE	使能system镜像压缩功能
BOOTIMG_SUPPORT_MAGISK	使能magisk框架支持
BUILD_WITH_GO_OPT	使能低内存机器优化配置
BOOT_VIDEO_ENABLE	使能开机视频功能

## 6.3 预置APK

Android上的应用预安装功能，主要是指配置产品时，根据厂商要求，将事先准备好的第三方应用预置进Android系统。

**预安装的APK应用需要得到对应厂商授权，若因为开发者及客户厂商私自预安装未授权应用进而需要承担法律责任的，RK概不负责。**

预安装分为可卸载预安装和不可卸载预安装，本文主要阐述的是可卸载预安装的功能。配置步骤如下：

- 1) 若是希望可卸载预安装，新增文件夹device/rockchip/rk3328/rk3328\_box/preinstall\_del；若是不可卸载原装，新增文件夹device/rockchip/rk3328/rk3328\_box/preinstall。
- 2) 拷贝需要预制的第三方应用到上述文件夹，注意apk文件名尽量使用英文，避免空格。
- 3) 编译结束后会将预制的文件拷贝至system固件中。烧录后，系统会自动安装这些应用到data/app目录。

4) 需要注意的是，在preinstall目录中的应用，即使用户在使用过程中将其卸载，但在恢复出厂设置后，应用又会自动安装。如果希望恢复出厂设置后不再恢复预安装应用，可以将上述文件夹名字改为preinstall\_del\_forever即可实现。

## 6.4 开/关机动画

---

需要在产品的device/rockchip /common/BoardConfig.mk中配置BOOT\_SHUTDOWN\_ANIMATION\_RINGING := true，并且准备如下相应资源文件，编译结束后对应的资源文件会拷贝到相应的out目录下。

将开机动画 复制到 device/rockchip/common/bootanimation.zip (源码路径)

将关机动画 复制到 device/rockchip/common/shutdownanimation.zip (源码路径)

## 6.5 Parameter说明

---

请参考device/rockchip/rk3328/rk3328\_box目录下parameter.txt文件来相应修改配置，关于parameter中各个参数、分区情况细节，请参考RKDocs\common\RKTools manuals目录下的《Rockchip Parameter File Format Ver1.3.pdf》文档。

## 6.6 新增分区配置

---

请参考RKDocs\android\Android 增加一个分区配置指南 V1.00.pdf

## 6.7 显示框架配置

---

请参考RKDocs\android\Rockchip Android 8.1 BOX显示框架配置说明文档 V1.0-20180210.pdf

## 6.8 OTA升级

---

### 6.8.1 OTA 介绍

OTA ( over the air ) 升级是Android 系统提供的标准软件升级方式。它功能强大，提供了完全升级（完整包）、增量升级模式（差异包），可以通过本地升级，也可以通过网络升级。

详细的OTA升级及Recovery模块功能及配置，请参考RKDocs\android目录下《Rockchip Recovery用户操作指南 V1.03》。

### 6.8.2 生成完整包

完整包所包含内容：boot.img uboot.img vbmeta.img 及system、vendor、oem的升级patch.

发布一个固件正确的顺序：

- 1、make -j4
- 2、make otapackage -j4
- 3、./mkimage.sh

在out/target/product/rkxxxx/目录下会生成ota完整包rkxxxx-ota-eng.root.zip，改成update.zip即可拷贝到T卡或者内置的flash进行升级。



## 6.8.3 生成差异包

OTA差异包只有差异内容，包大小比较小，主要用于OTA在线升级，也可T卡本地升级。OTA差异包制作需要特殊的编译进行手动制作。

1、首先发布v1版本的固件，生成v1版本的完整包

2、保存

out/target/product/rkxxxx/obj/PACKAGING/target\_files\_intermediates/rk3188-target\_files-eng.root.zip 为 rkxxxx-target\_files-v1.zip，作为v1版本的基础素材包。

3、修改kernel代码或者android 代码，发布v2版本固件，生成v2版本完整包

4、保存

out/target/product/rkxxxx/obj/PACKAGING/target\_files\_intermediates/rk3188-target\_files-eng.root.zip 为 rkxxxx-target\_files-v2.zip，作为v2版本的基础素材包。

5、生成v1-v2的差异升级包：

```
./build/tools/releasetools/ota_from_target_files --block -v -i rkxxxx-target_files-v1.zip -p  
out/host/linux-x86 -k build/target/product/security/testkey rkxxxx-target_files-v2.zip  
out/target/product/rk3328/rkxxxx-v1-v2.zip
```

说明: 生成差异包命令格式:

```
ota_from_target_files  
--block  
-v -i 用于比较的前一个target file  
-p host主机编译环境  
-k 打包密钥  
用于比较的后一个target file  
最后生成的ota差异包。
```

## 6.9 预制Demo

在开发及样机准备中，多数开发者及厂商有需要集成测试音视频资源、图片资源等，本SDK也附带了预置Demo资源的功能，详情见8.8节OemTool打包工具使用。

## 6.10 开机视频

需要在产品的device/rockchip/common/BoardConfig.mk中配置BOOT\_VIDEO\_ENABLE ?= true，并且准备如下相应开机视频文件bootanimation.ts（默认代码中识别此视频后缀，其它格式可直接修改其文件名及后缀为bootanimation.ts即可，不需要转格式），编译结束后对应的资源文件会拷贝到相应的out目录下。

将开机视频复制到device/rockchip/common/bootvideo/bootanimation.ts(源码路径)

注意：开启开机视频功能后，默认已配置为将视频播完，通过属性persist.sys.bootvideo.showtime可控制播放时间：

-1：代表没设置时长，按照开机自然阶段时间展示；

-2：代表要将视频播完才能进入launcher；

配置其它大于0的数字表示具体要播放的，超过120秒按120秒播放。

## 6.11 低内存机器内存优化配置

需要在产品的device/rockchip/rk3328/rk3328\_box/BoardConfig.mk中添加配置

```
BUILD_WITH_GO_OPT := true
```

重新编译系统生效。

主要优化内容：

- 1.系统开启Android Go内存优化成果；
- 2.视频库播放内存占用及缓存大小优化；
- 3.lowmemorykiller水线及策略调整；

目前SDK默认未开启。

## 6.12 DRM Widevine Level 1配置

SDK 默认支持 Widevine Level 3功能，如果需要使用 widevine Level 1功能，请先参考文档：RKDocs/android/《Rockchip Android Widevine 项目启动准备指南 V2.0-20180622.pdf》进行相关准备工作，代码部分按下列配置即可。

按如下修改打开配置编译，注意需要烧写L1 key才能生效。

device/rockchip/rk3328\$ git diff .

```
diff --git a/rk3328_box/BoardConfig.mk b/rk3328_box/BoardConfig.mk
index 1425e95..3978741 100755
--- a/rk3328_box/BoardConfig.mk
+++ b/rk3328_box/BoardConfig.mk
@@ -95,7 +95,7 @@ BUILD_WITH_GTVS := false
 BUILD_WITH_GOOGLE_FRP := false

# for widevine drm
-BOARD_WIDEVINE_OEMCRYPTO_LEVEL := 3
+BOARD_WIDEVINE_OEMCRYPTO_LEVEL := 1

#for microsoft drm
BUILD_WITH_MICROSOFT_PLAYREADY :=true
```

**注意：**开启L1功能时，播放需要支持SVP的视频，目前需要在内核中将安全内存单独划分出来，如3328 样机板开启svp内存256M配置方法：

kernel\$ git diff

```
diff --git a/arch/arm64/boot/dts/rockchip/rk3328-box-liantong.dtsi
b/arch/arm64/boot/dts/rockchip/rk3328-box-liantong.dtsi
index ccaca5d..a695bf4 100644
--- a/arch/arm64/boot/dts/rockchip/rk3328-box-liantong.dtsi
+++ b/arch/arm64/boot/dts/rockchip/rk3328-box-liantong.dtsi
@@ -314,7 +314,7 @@
     * enable like this ( 256M ) :
     * reg = <0x0 0x20000000 0x0 0x10000000>;
     */
-    reg = <0x0 0x20000000 0x0 0x0>;
+    reg = <0x0 0x20000000 0x0 0x10000000>;
};
```

## 6.13 媒体中心

- **新增 4K 图片展示功能**，可在对应产品目录下 device.mk 配置persist.media.4k 属性控制默认状态。当前默认配置为 false。

persist.media.4k : true 媒体中心 4K 图片，默认显示原图大小

persist.media.4k : false 媒体中心 4K 图片，进行大小自适应调整显示

- **新增 heic 动态图片支持**，对 heic 动态图片可长按“center”键，显示动态效

## 6.14 TWRP recovery

Team Win Recovery Project (TWRP)是一个开放源码软件的定制恢复模式映像，供基于安卓的设备使用。它提供了一个支持鼠标操作的界面，允许用户向第三方安装固件和备份当前的系统。

可根据 <https://github.com/rockchip-software/TWRP> 编译对应的 recovery.img,并通过 RKTools\windows\AndroidToolReleasev2.65 工具烧写。

## 6.15 Magisk Support

Magisk 是一套用于定制 Android 的开源工具，支持高于 Android 5.0 (API 21)的设备。它涵盖了Android 定制的基本部分: root, boot scripts, SELinux patches, AVB2.0 / dm-verity / forceencrypt removals 等。

可根据 RKDocs/android/Rockchip\_Magisk\_installation\_guidelines.pdf 文档在 Rockchip 平台安装Magis

## 6.16 安全启动方案

Rockchip 安全启动方案基于 RK 芯片提供的硬件保护机制，对机顶盒的引导程序 loader, uboot, trust 镜像以及 Android 系统 ( boot(含 kernel), recovery、system、vendor、oem 等镜像 ) 提供可靠的安全保护。对于机顶盒产品可用于保护机顶盒系统安全，防止机顶盒被刷机或业务相关应用被篡改等。

可参考文档 RKDocs/android/Rockchip Android Pie 安全启动方案使用指南 V1.00.pdf

## 6.17 Microsoft PlayReady

PlayReady 为微软公司的新的 DRM 系统，它是 WMDRM(Windows Media DRM)的升级产品，可以为数字媒体提供内容保护支持。微软 PlayReady 官方网站为 <http://www.microsoft.com/playready/>，可以从该网站获取 PlayReady 产品、技术、文档、License 及支持等信息；PlayReady 官方测试网站为<http://test.playready.microsoft.com>，可以使用该网站的测试用例和码流对用户的 PlayReady 产品进行基本功能测试。Rockchip PlayReady 方案分为 SW 版本和 HW 版本，根据微软证书定义的 Security Level，我们通常也把 SW 版本称为 SL2000 版本，HW 版本称为 SL3000 版本。SW 版本能为内容和证书提供基本和必要的保护，对芯片和方案没有特别要求，适用于内容提供商没有特殊要求的场景。HW 版本使用 Rockchip 芯片的 Trustzone 硬件保护机制，对证书、加解密密钥以及解密后的码流提供更高级别的保护，HW 版本适用于内容提供商明确要求 Trustzone 特性或 Secure Video Path 的场景。

在 rockchip 平台使用 playready 功能，需先向微软申请 license，然后向 rockchip 申请相关补丁，详细可联系 FAE 咨询。

## 6.18 动态加载 UiMode

在 box 类型产品中，由于默认 UI 资源及布局使用的是 television 风格（即这里讲的 UiMode 配置），如果客户需要集成只兼容 phone/mid 风格的应用时，可能会由于 apk 本身兼容性问题引起无法使用的异常。现在我们提供了一种方式，通过白名单配置来动态的为 App 加载 UiMode（源码路径：

device/rockchip/common/uimode/uimode\_app.xml, 设备路径：vendor/etc/uimode\_app.xml）。

目前此方式仅支持 box 设备, 使用方法请参考 device/rockchip/common/uimode/ReadME.

为方便调试，客户可以在机器运行过程中通过进入 TVSettings-->app-->如 netflix 应用设置菜单，在最下面一项 uimode 中选择要测试的 uimode 进行调试。

## 6.19 显示参数的调整和保存

RK 平台提供一个专门的分区来保存显示参数，系统可以从该分区读取显示参数并且应用。该分区我们称为 baseparameter 分区，主要功能包括：

1. 提供系统和应用保存和调整 hdmi 或者 dp 等显示设备支持的分辨率、时序等信息位置；
2. 保持显示信息的独立和统一性，优化开机各阶段显示效果统一的体验。

SDK 默认已开启此分区功能，编译完成打包固件时会自动生成 baseparameter.img 镜像供烧写。

为了客户预置方便，SDK 默认提供常用的 720p 和 1080p 显示镜像供预置，具体见源码目录：

device/rockchip/common/baseparameter/

若需要改成其他预置镜像，一种是通过文件编辑器小端修改预置的baseparameter.img参数；一种是在机器中通过设置应用设置好需要预置的显示配置如分辨率，color等，通过机器中命令行工具saveBaseParameter命令进行配置和生成新镜像，saveBaseParameter命令使用方法如下：

```
rk3328_box:/ # saveBaseParameter
saveParameter: read and write baseparameter partition tool
Usage:
    -h          Help info
    -p          Print Baseparameter
    -t          output to target file (e: "/sdcard/baseparameter.img")
    -d          Choose Display to Setting (e: 0 or 1)
    -f          Framebuffer Resolution (e: 1920x1080@60)
    -D          Display Attach Devices (e: HDMI-A,TV)
    -c          Color (e: RGB-8bit or YCBCR444-10bit)
    -u          Is Enable Auto Resolution (2:auto resolution; 1:set one fixed resolution)
    -o          Overscan (e: overscan "100,100,100,100")
    -b          BCSH (e: "50,50,50,50")
    -R          Reset Baseparameter (1:only reset user setting baseparameter partition; 2:reset
baseparameter paratition include backup)

Example: saveBaseParameter -d 0 -f 1920x1080@60 -D "HDMI-A,TV" -c Auto -u 2 -o "100,100,100,100"
-b "50,50,50,50"

===== Rockchip All Rights Reserved =====
```

## 7 系统调试

本节重点介绍SDK开发过程中的一些调试工具和调试方法，并会不断补充完善，帮助开发者快速上手基础系统调试，并做出正确的分析。

### 7.1 ADB工具

#### 7.1.1 概述

ADB ( Android Debug Bridge ) 是 Android SDK里的一个工具，用这个工具可以操作管理 Android 模拟器或真实的 Android 设备。主要功能有：

- 运行设备的 shell ( 命令行 )
- 管理模拟器或设备的端口映射
- 计算机和设备之间上传/下载文件
- 将本地 apk软件安装至模拟器或 Android 设备

ADB 是一个“客户端 - 服务器端”程序，其中客户端主要是指PC，服务器端是Android 设备的实体机器或者虚拟机。根据PC连接Box机器的方式不同，ADB 可以分为两类：

- 网络 ADB：主机通过有线/无线网络（同一局域网）连接到STB设备
- USB ADB：主机通过 USB 线连接到STB设备

## 7.1.2 USB adb使用说明

USB adb 使用有以下限制：

- 只支持 USB OTG 口
- 不支持多个客户端同时使用（如 cmd 窗口，eclipse等）
- 只支持主机连接一个设备，不支持连接多个设备

连接步骤如下：

- 1、Box机器已经运行 Android 系统，设置->开发者选项->已连接到计算机 打开，usb调试开关打开。
- 2、PC主机只通过 USB 线连接到机器 USB otg 口，然后电脑通过如下命令与Box机器相连。

```
adb shell
```

- 3、测试是否连接成功，运“adb devices”命令，如果显示机器的序列号，表示连接成功。

## 7.1.3 网络adb使用要求

adb早期版本只能通过USB来对设备调试，从adb v1.0.25开始，增加了对通过tcp/ip调试Android设备的功能。

如果你需要使用网络adb来调试设备，必须要满足如下条件：

- 1、设备上面首先要有网口，或者通过WiFi连接网络。
- 2、设备和研发机（PC机）已经接入局域网，并且设备设有局域网的IP地址。
- 3、要确保研发机和设备能够相互ping得通。
- 4、研发机已经安装了adb。
- 5、确保Android设备中adbd进程（adb的后台进程）已经运行。adbd进程将会监听端口5555来进行adb连接调试。

## 7.1.4 SDK网络adb端口配置

SDK默认未开启网络adb，需要手动在开发者选项中打开。

## 7.1.5 网络adb使用

本节假设设备的ip为192.168.1.5，下文将会用这个ip建立adb连接，并调试设备。

- 1、首先Android设备需要先启动，如果可以话，可以确保一下adbd启动(ps命令查看)。
- 2、在PC机的cmd中，输入：

```
adb connect 192.168.1.5:5555
```

如果连接成功会进行相关的提示，如果失败的话，可以先kill-server命令，然后重试连接。

```
adb kill-server
```

- 3、如果连接已经建立，在研发机中，可以输入adb相关的命令进行调试了。比如adb shell，将会通过tcp/ip连接设备上。和USB调试是一样的。

- 4、调试完成之后，在研发机上面输入如下的命令断开连接：

```
adb disconnect 192.168.1.5:5555
```

## 7.1.6 手动修改网络adb端口号

若SDK未加入adb端口号配置，或是想修改adb端口号，可通过如下方式修改：

1、首先还是正常地通过USB连接目标机，在windows cmd下执行adb shell进入。

2、设置adb监听端口：

```
#setprop service.adb.tcp.port 5555
```

3、通过ps命令查找adbd的pid

4、重启adbd

```
#kill -9<pid>，这个pid就是上一步找到那个pid
```

杀死adbd之后，android的init进程后自动重启adbd。adbd重启后，发现设置了service.adb.tcp.port，就会自动改为监听网络请求。

## 7.1.7 ADB常用命令详解

### (1) 查看设备情况

查看连接到计算机的 Android 设备或者模拟器：

```
adb devices
```

返回的结果为连接至开发机的 Android 设备的序列号或是IP和端口号（Port）、状态。

### (2) 安装apk

将指定的 apk 文件安装到设备上：

```
adb install <apk文件路径>
```

示例如下：

```
adb install "F:\WishTV\WishTV.apk"
```

重新安装应用：

```
adb install -r <apk文件路径>
```

示例如下：

```
adb install -r "F:\WishTV\WishTV.apk"
```

### (3) 卸载apk

完全卸载：

```
adb uninstall <package>
```

示例如下：

```
adb uninstall com.wishtv
```

### (4) 使用 rm 移除 apk 文件：

```
adb shell rm <filepath>
```

示例如下：

```
adb shell  
rm "system/app/WishTV.apk"
```

示例说明：移除“system/app”目录下的“WishTV.apk”文件。

### （5）进入设备和模拟器的shell

进入设备或模拟器的 shell 环境：

```
adb shell
```

### （6）从电脑上传文件到设备

用 push 命令可以把本机电脑上的任意文件或者文件夹上传到设备。本地路径一般指本机电脑；远程路径一般指 adb 连接的单板设备。

```
adb push <本地路径> <远程路径>
```

示例如下：

```
adb push "F:\WishTV\WishTV.apk" "system/app"
```

示例说明：将本地“WishTV.apk”文件上传到 Android 系统的“system/app”目录下。

### （7）从设备下载文件到电脑

pull 命令可以把设备上的文件或者文件夹下载到本机电脑中。

```
adb pull <远程路径> <本地路径>
```

示例如下：

```
adb pull system/app/Contacts.apk F:\
```

示例说明：将 Android 系统“system/app”目录下的文件或文件夹下载到本地“F:\”目录下。

### （8）查看 bug 报告

需要查看系统生成的所有错误消息报告，可以运行 adb bugreport 指令来实现，该指令会将 Android 系统的 dumsys、dumpstate 与 logcat 信息都显示出来。

### （9）查看设备的系统信息

在 adb shell 下查看设备系统信息的具体命令。

```
adb shell getprop
```

## 7.2 Logcat 工具

Android 日志系统提供了记录和查看系统调试信息的功能。日志都是从各种软件和一些系统的缓冲区中记录下来的，缓冲区可以通过 Logcat 来查看和使用。Logcat 是调试程序用的最多的功能。该功能主要是通过打印日志来显示程序的运行情况。由于要打印的日志量非常大，需要对其进行过滤等操作。

### 7.2.1 Logcat 命令使用



用 logcat 命令来查看系统日志缓冲区的内容：

基本格式：

```
[adb] logcat [  
  ]
```

示例如下：

```
adb shell
```

```
logcat
```

## 7.2.2 常用的日志过滤方式

控制日志输出的几种方式：

- 控制日志输出优先级。

示例如下：

```
adb shell  
logcat *:W
```

示例说明：显示优先级为 warning 或更高的日志信息。

- 控制日志标签和输出优先级。

示例如下：

```
adb shell  
logcat ActivityManager:I MyApp:D *:S
```

示例说明：支持所有的日志信息，除了那些标签为“ActivityManager”和优先级为“Info”以上的、标签为“MyApp”和优先级为“Debug”以上的。

- 只输出特定标签的日志

示例如下：

```
adb shell  
logcat WishTV:* *:S
```

或者

```
adb shell  
logcat -s WishTV
```

示例说明：只输出标签为 WishTV 的日志。

- 只输出指定优先级和标签的日志

示例如下：

```
adb shell
logcat WishTV:I *:S
```

示例说明：只输出优先级为 I，标签为 WishTV 的日志。

### 7.2.3 查看上次log

可以加-L 参数来打印出上次系统复位前的 logcat 信息。若出现拷机异常或者异常掉电的情况，可通过该命令打印出上一次 Android 运行状态的日志。命令如下：

```
adb shell
logcat -L
```

## 7.3 Procrank工具

Procrank 是 Android 自带一款调试工具，运行在设备侧的 shell 环境下，用来输出进程的内存快照，便于有效的观察进程的内存占用情况。

包括如下内存信息：

- VSS：Virtual Set Size 虚拟耗用内存大小（包含共享库占用的内存）
- RSS：Resident Set Size 实际使用物理内存大小（包含共享库占用的内存）
- PSS：Proportional Set Size 实际使用的物理内存大小（比例分配共享库占用的内存）
- USS：Unique Set Size 进程独自占用的物理内存大小（不包含共享库占用的内存）

注意：

- USS 大小代表只属于本进程正在使用的内存大小，进程被杀死后会被完整回收；
- VSS/RSS 包含了共享库使用的内存，对查看单一进程内存状态没有参考价值；
- PSS 是按照比例将共享内存分割后，某单一进程对共享内存区的占用情况。

### 7.3.1 使用procrank

执行procrank，前需要先让终端获取到root权限

su

命令格式：

```
procrank [ -W ] [ -v | -r | -p | -u | -h ]
```

常用指令说明：

- v：按照 VSS 排序
- r：按照 RSS 排序
- p：按照 PSS 排序

- u：按照 USS 排序
- R：转换为递增[递减]方式排序
- w：只显示 working set的统计计数
- W：重置 working set 的统计计数
- h：帮助

示例：

- 输出内存快照：

```
procrank
```

- 按照 VSS 降序排列输出内存快照：

```
procrank -v
```

默认procrank输出是通过PSS排序。

## 7.3.2 检索指定内容信息

查看指定进程的内存占用状态，命令格式如下：

```
procrank | grep [cmdline | PID]
```

其中 cmdline 表示需要查找的应用程序名，PID 表示需要查找的应用进程。

输出 systemUI进程的内存占用状态:

```
procrank | grep "com.android.systemui"
```

或者：

```
procrank | grep 3396
```

## 7.3.3 跟踪进程内存状态

通过跟踪内存的占用状态，进而分析进程中是否存在内存泄露场景。使用编写脚本的方式，连续输出进程的内存快照，通过对比 USS 段，可以了解到此进程是否内存泄露。

示例：输出进程名为 com.android.systemui 的应用内存占用状态，查看是否有泄露：

1、编写脚本 test.sh

```
#!/bin/bash
while true;do
adb shell procrank | grep "com.android.systemui"
sleep 1
done
```

2、通过 adb 工具连接到设备后，运行此脚本：./test.sh。如图所示。

1548835622725

图7-1跟踪进程内存状态

## 7.4 Dumpsys工具

Dumpsys 工具是 Android系统中自带的一款调试工具，运行在设备侧的 shell 环境下，提供系统中正在运行的服务状态信息功能。正在运行的服务是指 Android binder机制中的服务端进程。

dumpsys 输出打印的条件：

- 1、只能打印已经加载到 ServiceManager中的服务；
- 2、如果服务端代码中的 dump 函数没有被实现，则没有信息输出。

### 7.4.1 使用Dumpsys

- 查看Dumpsys帮助

作用：输出dumpsys帮助信息。

```
dumpsys -help
```

- 查看Dumpsys包含服务列表

作用：输出dumpsys所有可打印服务信息，开发者可以关注需要调试服务的名称。

```
dumpsys -l
```

- 输出指定服务的信息

作用：输出指定的服务的 dump 信息。

格式：`dumpsys [servicename]`

示例：输出服务 SurfaceFlinger的信息，可执行命令：

```
dumpsys SurfaceFlinger
```

- 输出指定服务和应有进程的信息

作用：输出指定服务指定应用进程信息。

格式：`dumpsys [servicename][应用名]`

示例：输出服务名为 meminfo，进程名为 com.android.systemui 的内存信息，执行命令：

```
dumpsys meminfo com.android.systemui
```

注意：服务名称是大小写敏感的，并且必须输入完整服务名称。

## 7.5 音视频问题调试工具及文档

RKDocs\common\video目录下《Rockchip Videodebug\_V1.0\_20170109》为音视频问题调试工具包，其中也包含了详细的调试说明文档《Rockchip音视频框架调试文档V0.4》可帮助开发者对音视频问题做出基本的排查和分析。

## 7.5 串口调试

## 7.5.1 串口配置

调试过程中最方便的就是串口的输入输出，这里需要注意的是 RK3328 波特率设置为 1500000。RTS/CTS 不要勾选，否则串口无法输入。

## 7.5.2 FIQ模式

快速中断请求 ( Fast Interrupt Request , FIQ)在 ARM中，FIQ 模式是特权模式中的一种，同时也属于异常模式一类。

RK 平台上，在串口输入“fiq”，可以进入该模式。此时会有使用帮助跳出，可根据情况进行一些调试。经常在死机，或系统卡死的时候起作用。

## 7.7 Last log开启

在 dts 文件里面添加下面两个节点

```
ramoops_mem: ramoops_mem {
    reg = <0x0 0x110000 0x0 0xf0000>;
    reg-names = "ramoops_mem";
};

ramoops {
    compatible = "ramoops";
    record-size = <0x0 0x20000>;
    console-size = <0x0 0x80000>;
    ftrace-size = <0x0 0x00000>;
    pmsg-size = <0x0 0x50000>;
    memory-region = <&ramoops_mem>;
};
```

```
root@rk328_box:/sys/fs/pstore # ls
```

dmesg-ramoops-0 上次内核 panic 后保存的 log。

pmsg-ramoops-0 上次用户空间的 log，android 的 log。

ftrace-ramoops-0 打印某个时间段内的 function trace。

console-ramoops-0 last\_log 上次启动的 kernel log，但只保存了优先级比默认 log level 高的 log。

使用方法：

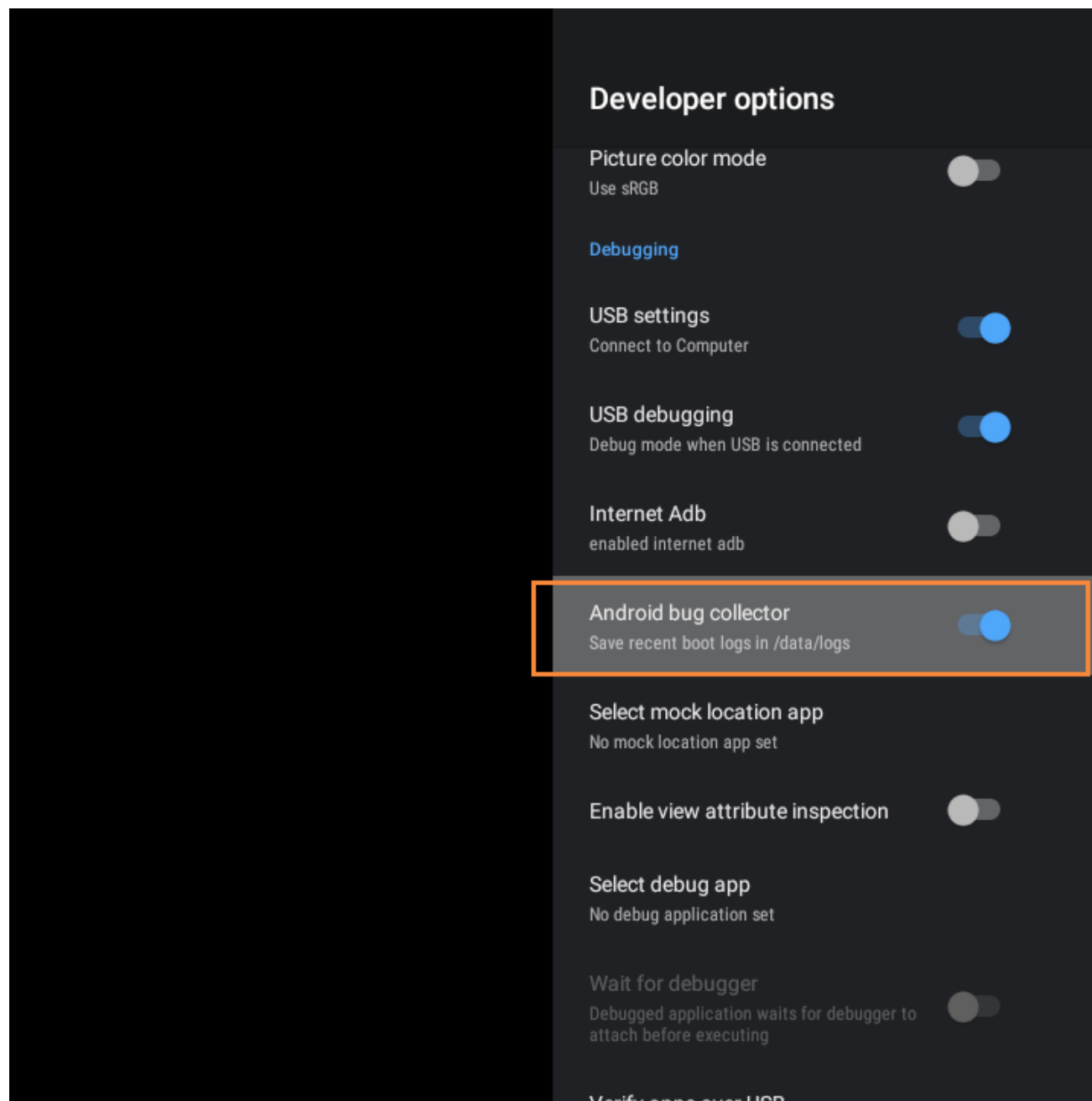
```
cat /sys/fs/pstore/dmesg-ramoops-0
cat /sys/fs/pstore/console-ramoops-0
logcat -L (pmsg-ramoops-0) 通过 logcat 取出来并解析
cat /sys/fs/pstore/ftrace-ramoops-0
```

## 7.8 Log自动保存系统 abc

为了简化客户的操作，让没有 android 开发基础的人抓取出问题时的 log，我们新增一个 log 自动保存功能。

### 7.8.1 使用方法：

(1)在开发者选项中点击 Android bug collector 来开启此功能:



(2) 开启后log会开始保存在/data/vendor/logs目录下对应文件夹中（按生成日期命名），里面主要保存此次开机后系统的android和内核log：

android

kernel

process

---

## 8 常用工具说明

---

本节简单介绍SDK附带的一些开发及量产工具的使用说明，方便开发者了解熟悉RK平台工具的使用。详细的工具使用说明请见RKTools目录下各工具附带文档，及RKDocs\common\ RKTools manuals目录下工具文档。

### 8.1 StressTest

---

设备上使用Stresstest 工具，对待测设备的各项功能进行压力测试，确保各项整个系统运行的稳定性。SDK通过打开计算器应用，输入“83991906=”暗码，可启动StressTest应用，进行各功能压力测试。

Stresstest 测试工具测试的内容主要包括：

#### 模块相关

- Camera 压力测试：包括Camera 打开关闭，Camera 拍照以及Camera 切换。
- Bluetooth 压力测试： 包括Bluetooth 打开关闭。
- Wifi 压力测试： 包括Wifi 打开关闭，（ ping 测试以及iperf 测试待加入 ）。

#### 非模块相关

- 飞行模式开关测试。
- 休眠唤醒拷机测试。
- 视频拷机测试。
- 重启拷机测试
- 恢复出厂设置拷机测试。

### 8.2 PCBA测试工具

---

PCBA 测试工具用于帮助在量产的过程中快速地甄别产品功能的好坏，提高生产效率。目前包括屏幕（ LCD ）、无线（ wifi ）、蓝牙（ bluetooth ）、DDR/EMMC 存储、SD 卡（ sdcard ）、UST HOST、按键（ KEY ），喇叭耳机（ Codec ）测试项目。

这些测试项目包括自动测试项和手动测试项，无线网络、DDR/EMMC、以太网为自动测试项，按键、SD卡、USB HOST、Codec、为手动测试项目。

具体PCBA功能配置及使用说明，请参考RKDocs\android\ROCKCHIP\_PCBA测试工具开发指南\_V1.1\_20171222.pdf。

### 8.3 DDR测试工具

---

设备上使用DDR测试工具，对待测设备的DDR进行稳定性测试，确保DDR功能正常及稳定。RK3328 DDR测试工具还未发布，后续会随SDK更新。

### 8.4 Android开发工具

---

#### 8.4.1 下载镜像

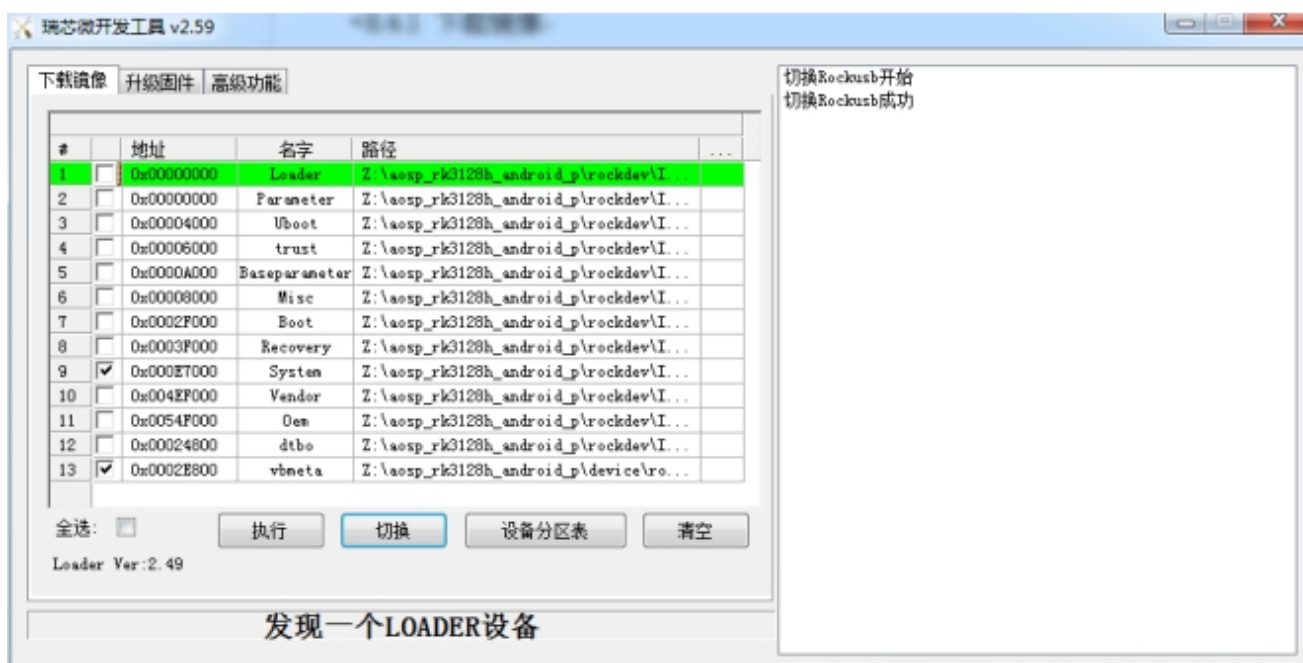


图8-1 Android开发工具下载镜像

- 1) 连接开发板进入下载模式（下载模式先按住开发板reset按键，再长按recover按键约3-4s时间进入loader模式）。
- 2) 打开工具点击下载镜像菜单，点击红色箭头对应列会跳出来一个文件选择框，可以选择对应分区的img本地地址，其他几项依次配置。
- 3) 配置完成后，点击执行就可以看到右边空白框进入下载提示。

其中“低格”按钮是用来擦除设备的，“清空”按钮是清空编辑框文本。

## 8.4.2 升级固件



图8-2 Android开发工具升级固件

- 1) 进行打包固件。



2) 点击固件选择刚打包好的update.img文件，并点击升级按钮进行下载。（注意设备必须在下载模式下）。

### 8.4.3 高级功能



图8-3 Android开发工具高级功能

Boot只能选择打包好的update.img文件或是loader的文件；

固件必须使用打包后的update.img;

解包功能可将update.img拆解为各部分镜像文件。

## 8.5 update.img打包

RK3328平台支持将各零散镜像文件，打包成一个完成的update.img形式，方便量产烧写及升级。具体打包步骤如下：

1) 打开AndroidTool工具目录底下的rockdev目录。编辑package-file。

按照package-file进行配置，package-file里面有一些img镜像放在Image目录底下的，如果没有该目录存在，则自己手工新建该Image目录，并将需要放到Image目录的镜像放进去即可。且注意配置时，镜像名字的准确。其中注意bootloader选项，应该根据自己生成的loader名称进行修改。

2) 编辑mkupdate.bat

```
1 Afptool -pack .\backupimage backupimage\backup.img
2 Afptool -pack ./ Image\update.img
3
4
5 RKImageMaker.exe -RK322H RK3328MiniLoaderAll.bin Image\update.img update.img -os_type:androidos
6
7 rem update.img is new format, Image\update.img is old format, so delete older format
8 del Image\update.img
9
10 pause
```

图8-4 update.img打包脚本

需要修改loader名称为实际存放的loader名称即可。

3) 点击mkupdate.bat运行即可，运行完会在该目录生成一个update.img。

## 8.6 固件签名工具

选择chip类型和加密类型，如果是RK3328则选择efuse。

点击“Generate Key Pairs”按钮，则会生成公私钥对，点击保存。

点击加载密钥，会连续跳出来两次选择密钥文件的界面，第一次为选择私钥文件，第二次为公钥选择文件。

点击“Sign Firmware”按钮，签名update.img文件。

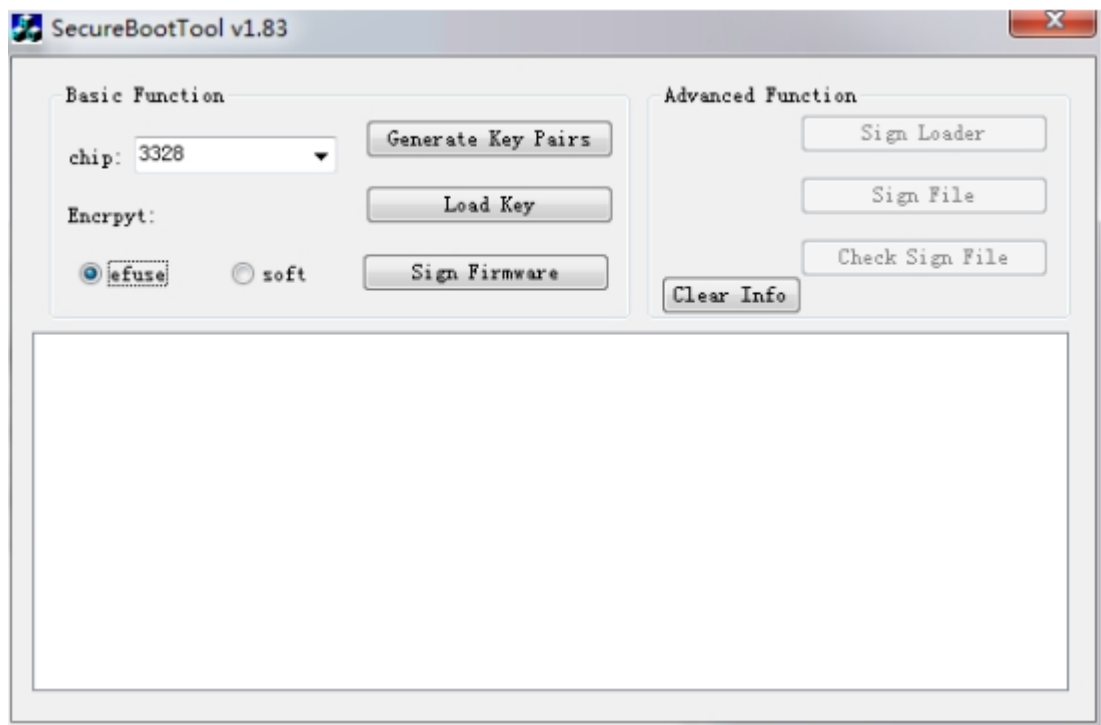


图8-5 固件签名工具

附键盘输入R+K+Ctrl+Alt键可打开右侧隐藏功能。

## 8.7 序列号/Mac/厂商信息烧写-WNpctool工具

在RK3328平台上，序列号/Mac/厂商信息烧写，都是使用WNpctool工具进行的。以下说明该工具基本的用法。

### 8.7.1 序列号获取

在RK3328平台上当未用工具烧写过序列号时，默认是读取WiFi Mac地址，并依此随机产生一个序列号的。若需要读取工具烧录的序列号值，需要手动修改对应的配置选项。

需修改/system/core/ drmservice/drmservice.c文件中：

```
#define SERIALNO_FROM_IDB 1 //if 1 read sn from idb3; if 0 generate sn auto
```

设为1后，默认会从vendor storage中读取工具写入的序列号。

### 8.7.2 WNpctool写入步骤

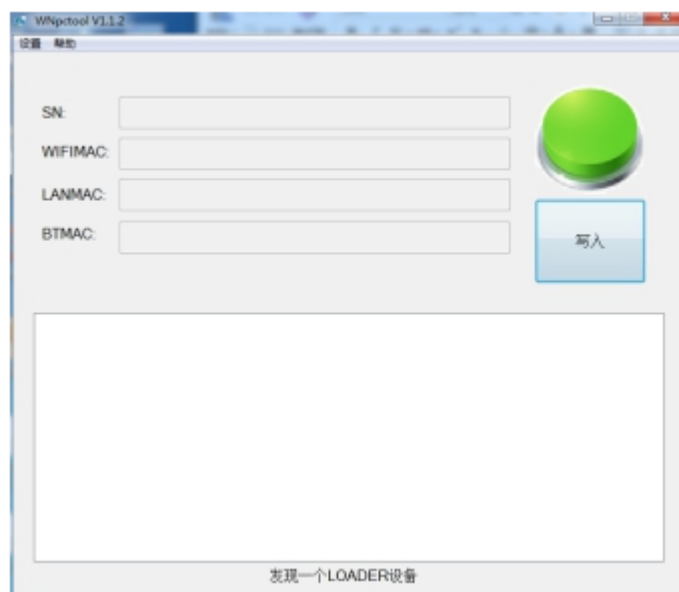


图8-6 WNpctool工具

- 1) 进入loader模式。
- 2) 点击设置按钮，会有一个下拉框按钮，点击“读取”按钮，用来切换是写入还是读取功能。切换到写入功能。
- 3) 点击模式，出现下列窗口，用来设置SN/WIFI/LAN/BT

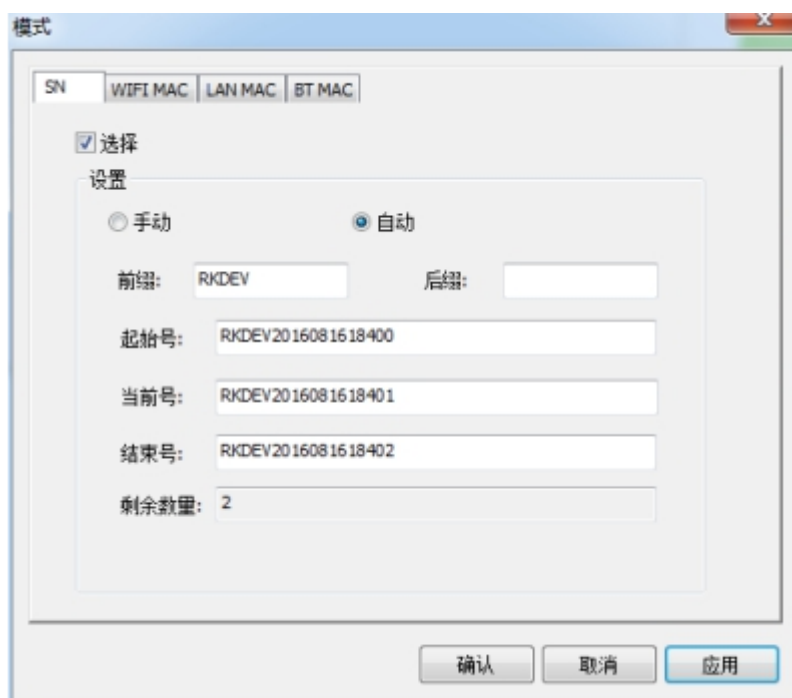


图8-7 WNpctool工具模式设置

- 4) 设置完成后，点击应用按钮，关闭窗口，返回主窗口，点击写入按钮即可。

### 8.7.3 WNpctool读取步骤

- 1) 进入loader模式。
- 2) 点击设置按钮，会有一个下拉框按钮，点击“读取”按钮，用来切换是写入还是读取功能。切换到读取功能。

3) 点击“读取”即可。

## 8.8 OemTool打包工具

### 8.8.1 Oem打包工具步骤

RK3328只支持Ext4镜像格式，故镜像格式选择Ext4.下载分区默认UserData分区，可直接不填写。

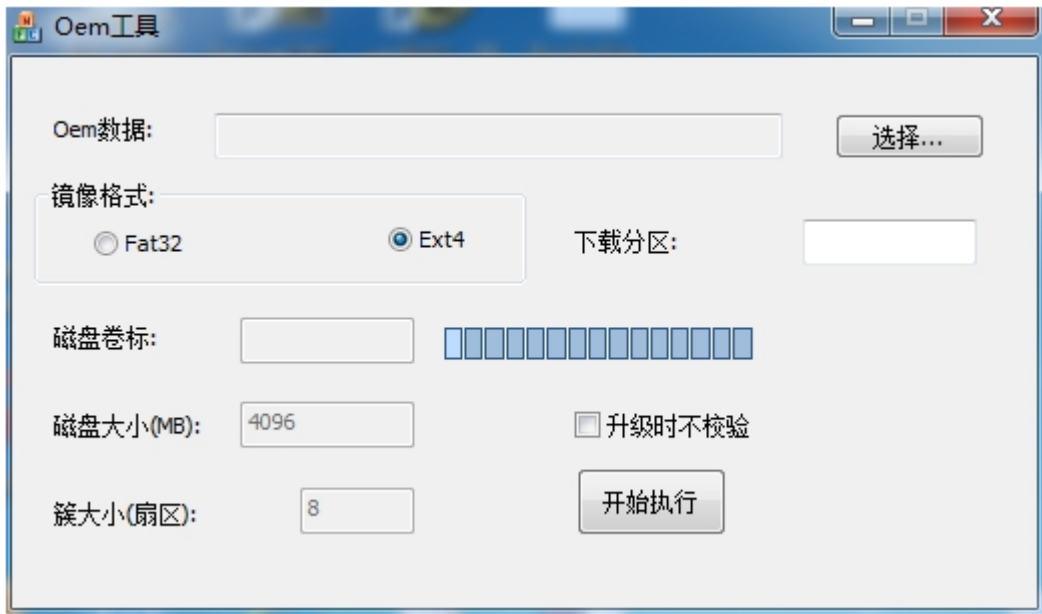


图8-8 Oem工具

点击选择按钮选择要打包的数据，数据必须是目录。目录最外围默认为data目录，假设你目录为/data/media/0,且0有一个文件为sss.txt(如下图示)。则当你升级完demo镜像的时候，会在RK3328系统上的data目录下有目录media/0,且在0目录下有文件sss.txt存在。

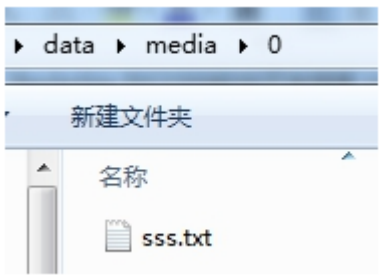


图8-9 Oem工具镜像制作文件夹路径要求

文件选择成功后，直接点击开始执行，会在Oem工具目录生成一个OemImage.img镜像。

将镜像放在FactoryTool工具上下载即可。

## 8.9 量产工具使用

### 8.9.1 工具下载步骤

- 1) 点击固件按钮，选择打包工具打包后的update.img，等待解包成功。
- 2) 如果需要demo镜像，则点击Demo拷贝按钮，添加由Oem工具打包的镜像，并单击Demo复选框。

- 3) 连接设备，并让设备进入loader或者maskrom模式，工具会自动进行下载。
- 4) 可同时连接多台设备，进行一拖多烧写，提高工厂烧写效率。

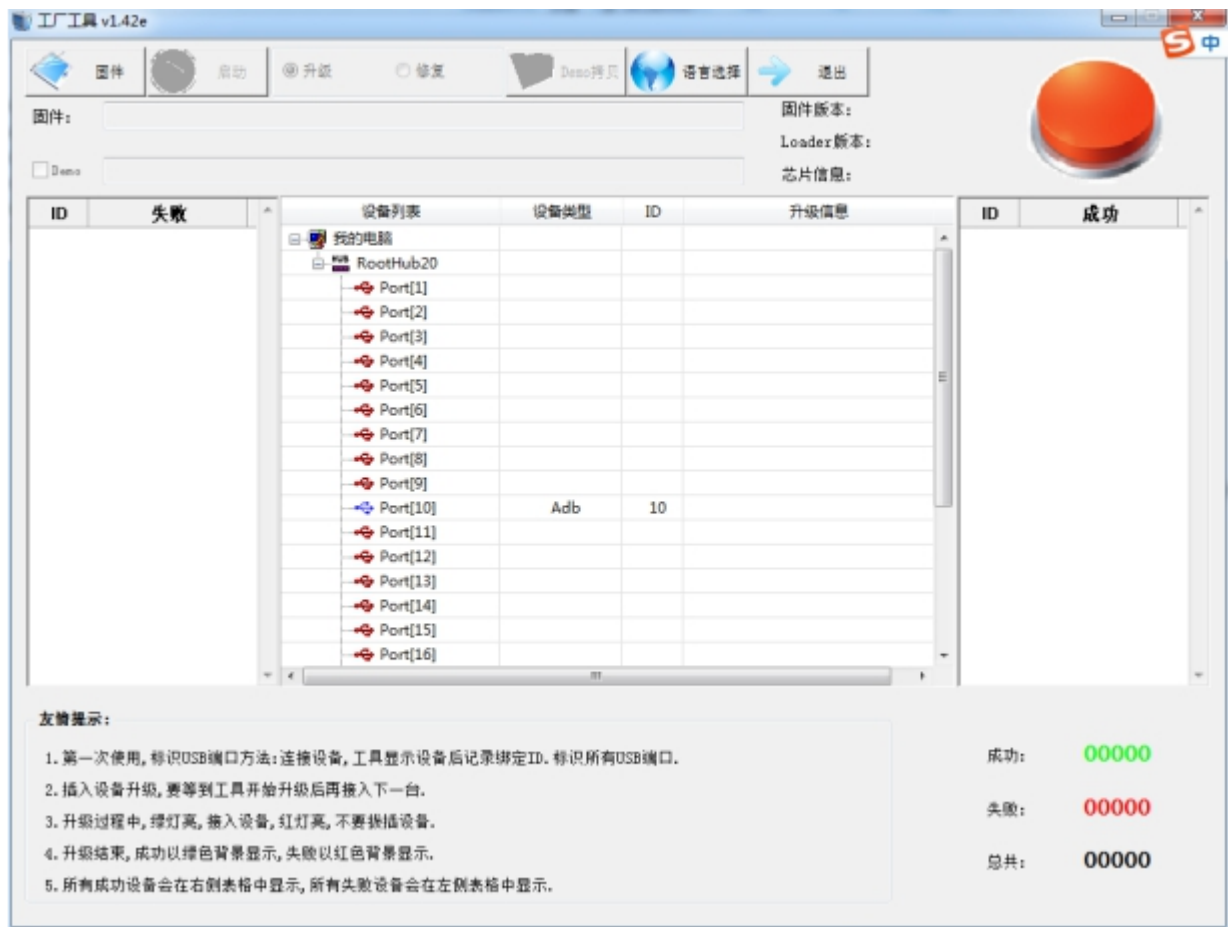


图8-10 量产工具

## 8.10 Box厂测工具

本测试工具用于帮助在量产过程中测试设备的好坏以及长时间运行的老化稳定性测试。测试工具只需用 U 盘或 SDCard 引导启动，方便快捷，提高生产效率。

本测试工具适用于运行完整固件的 PCBA 或整机测试，包含功能测试和老化测试。功能测试主要包含 WiFi、BT、LAN、SD、USB、HDMI、左右声道、按键、LED、CVBS 等。老化测试包含 CPU、VPU、GPU、Memory 的测试。

SDK默认编译已带有该测试工具，具体操作说明请参考RKDocs\common\RKTools manuals\Rockchip RK3328 Box厂测工具操作说明V2.0.pdf。

配置文件参考请见RKTools\windows\Rockchip Box厂测工具V2.0-M-20170327.zip。



图8-11 功能测试界面

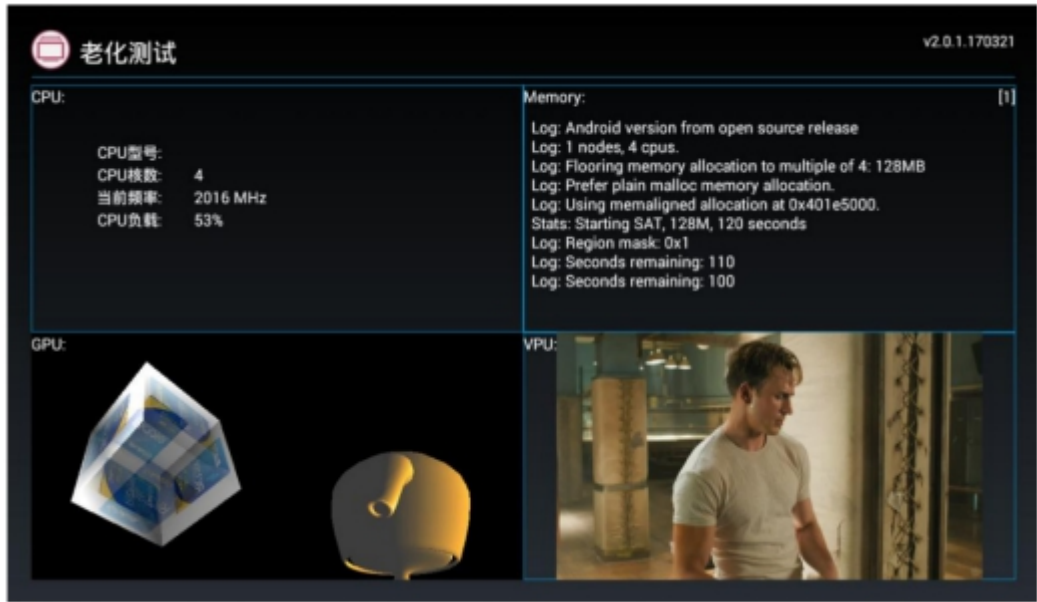


图8-12 老化测试界面

**注：**由于默认该工具插外设启动，如果客户是烧机后手动安装的方式，需要重启后再测试，不然由于系统本身限制，apk服务不会马上启动，引起无法识别测试文件问题。