密级状态： 绝密(　　) 秘密(　　) 内部资料(　　) 公开(　√)

Security Class: Top-Secret (　　) Secret (　　) Internal (　　) Public (√)

RK3399_Android7.1_软件开发指南

RK3399_Android7.1_Software_Development_Guide

（第二系统产品部）

(Technical Department, R & D Dept. II）

文件状态： **Status:** [　] 草稿 [　] Draft [　] 正在修改 [　] Modifying [√] 正式发布 [√] Released	**文件标识：** **File No.:**	RK-KF-YF-224
	当前版本： **Current Version：**	V2.14
	作　　者： **Author:**	郝小伟、刘益星、张文平 Hao Xiaowei, Liu Yixing, Zhang Wenping
	完成日期： **Finish Date:**	2020-03-31
	审　　核： **Auditor:**	陈海燕、黄祖芳、邓训金 Chen Haiyan, Huang Zufang, Deng Xunjin
	审核日期： **Finish Date:**	2020-03-31

福州瑞芯微电子股份有限公司

Fuzhou Rockchip Electronics Co., Ltd.

免责声明

本文档按“现状”提供，福州瑞芯微电子股份有限公司（“本公司”，下同）不对本文档的任何陈述、信息和内容的准确性、可靠性、完整性、适销性、特定目的性和非侵权性提供任何明示或暗示的声明或保证。本文档仅作为使用指导的参考。

由于产品版本升级或其他原因，本文档将可能在未经任何通知的情况下，不定期进行更新或修改。

Disclaimer

THIS DOCUMENT IS PROVIDED “AS IS”. FUZHOU ROCKCHIP ELECTRONICS CO., LTD.(“ROCKCHIP”)DOES NOT PROVIDE ANY WARRANTY OF ANY KIND, EXPRESSED, IMPLIED OR OTHERWISE, WITH RESPECT TO THE ACCURACY, RELIABILITY, COMPLETENESS, MERCHANTABILITY, FITNESS FOR ANY PARTICULAR PURPOSE OR NON-INFRINGEMENT OF ANY REPRESENTATION, INFORMATION AND CONTENT IN THIS DOCUMENT. THIS DOCUMENT IS FOR REFERENCE ONLY. THIS DOCUMENT MAY BE UPDATED OR CHANGED WITHOUT ANY NOTICE AT ANY TIME DUE TO THE UPGRADES OF THE PRODUCT OR ANY OTHER REASONS.

商标声明

“Rockchip”、“瑞芯微”、“瑞芯”均为本公司的注册商标，归本公司所有。

本文档可能提及的其他所有注册商标或商标，由其各自拥有者所有。

Trademark Statement

“Rockchip”, “瑞芯微”, “瑞芯” shall be Rockchip’s registered trademarks and owned by Rockchip. All the other trademarks or registered trademarks mentioned in this document shall be owned by their respective owners.

版权所有 © 2019 福州瑞芯微电子股份有限公司

超越合理使用范畴，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。

Copyright © 2019 Fuzhou Rockchip Electronics Co., Ltd.

Beyond the scope of fair use, neither any entity nor individual shall extract, copy, or distribute this document in any form in whole or in part without the written approval of Rockchip.

福州瑞芯微电子股份有限公司
Fuzhou Rockchip Electronics Co., Ltd.
地址： 福建省福州市铜盘路软件园 A 区 18 号
网址： www.rock-chips.com
客户服务电话： +86-4007-700-590
客户服务传真： +86-591-83951833
客户服务邮箱： fae@rock-chips.com

Fuzhou Rockchip Electronics Co., Ltd.
Address: No. 18 Building, A District, No.89, software Boulevard Fuzhou, Fujian, PRC
Website: www.rock-chips.com
Customer service Tel.: +86-4007-700-590
Customer service Fax: +86-591-83951833
Customer service e-mail: fae@rock-chips.com

版 本 历 史 Revision History

版本号 Version no.	作者 Author	修改日期 Revision Date	修改说明 Revision description	备注 Remark
V1.00	郝小伟 Hao Xiaowei	2017.01.16	正式发布 Initial version release	
V2.00	刘益星 Liu Yixing	2018.03.12	修改适配最新代码 Modify to fit with the latest code 增加双屏异显、异触说明 Add the instruction of dual-screen different display and different touch 增加 HDMI IN 功能说明 Add the instruction of HDMI IN function 增加系统调试章节 Add the chapter of system debugging 常用工具详细说明 Add the instruction of Commonly used tools	
V2.01	张文平 Zhang Wenping	2018-4-18	增加 lpddr4 配置、深度学习、多路 camera、音视频多路编解码、audio 3A 算法等章节 Add the chapters such as lpddr4 configuration, deep learning, multiple cameras, multi-channel encoding/decoding of audio and video, audio 3A and so on	
V2.02	张文平 Zhang Wenping	2018-5-14	根据代码的修改，更新 lpddr4 的配置说明 Update lpddr4 configuration instruction according to the code modification	
V2.03	张文平 Zhang Wenping	2018-7-5	添加新的功能支持，包括手写优化、log 系统、梯形校正、SSD、widevine 等 Add new functions including hand-written optimization, log system, trapezoid correction, SSD, widevine and so on	

V2.04	张文平 Zhang Wenping	2018-8-20	1. 新增 next-dev 分支 u-boot 支持，详情请参考 4.7 节 Add uboot support of next-dev branch, referring to section 4.7 2. 新增双屏异触功能说明 Add the instruction of dual-screen different touch function	
V2.05	张文平 Zhang Wenping	2018-9-27	新增显示参数的调整和保存功能，详情请参考 6.20 节。 Add the function to adjust and save the display parameter, referring to section 6.20	
V2.06	张文平 Zhang Wenping	2018-11-7	1. 新增多屏拼接功能说明，详见 6.7。 Add the instruction of multi-screen splicing function, referring to section 6.7 2. 更新显示参数的调整和保存说明文档，详见 6.21。 Update the instruction document of display parameter adjustment and save, referring to section 6.21 3. 新增自动化测试和 debug 脚本说明，详见 6.22。 Add the instruction of automatic testing and debug script, referring to section 6.22 4. 新增文件系统切换为 EXT4 功能说明，详见 6.23。 Add the function instruction of switching file system to EXT4, referring to section 6.23 5. 新增 DATA 分区加密功能使能和禁用说明，详见 6.24。 Add the instruction to enable and disable DATA partition encryption function, referring to	

			section 6.24 6. 新增 RK3399 性能说明文档，详见 6.25。 Add RK3399 performance instruction document, referring to section 6.25	
V2.07	张文平 Zhang Wenping	2019-1-29	1. 更正行业 sdk 下载说明。 Update industry SDK download instruction 2. 更新 8.4 节 DDR 测试工具说明。 Update DDR testing tool instruction in section 8.4 3. 更新性能开发文档名称 Update the performance development document name 4. 增加 rk3399K 芯片支持，请参考 6.26 节。 Add RK3399K chipset support, referring to section 6.26	
V2.08	张文平/魏建兴 Zhang Wenping/Wei Jianxing	2019-3-29	1. 根据最新代码，更新主副屏旋转功能文档。 Update the main and secondary screens rotation function document according to the latest code 2. 新增常见问题章节，描述客户常见软硬件问题，请参考第 9 章。 Add FAQ chapter which decribes the common hardware and software issues reported by customers, referring to Chapter 9 3. 新增禁用串口打印功能说明，详见 6.27 节。 Add the instruction to disable the serial port print function, referring to section 6.27 4. 更新 6.23 节，加入 data 分区切	

			换为 EXT4 的一些问题说明。 Update section 6.23, add some issue instruction when data partition is switched to EXT4 5. 新增 dp 音频配置说明，请参考 6.28 节进行配置。 Add the instruction of dp audio configuration, referring to section 6.28 to configure	
V2.09	张文平 Zhang Wenping	2019-5-27	1. 更新 lpddr4 章节，请仔细阅读本节（尤其是 5.8.3 节） Update lpddr4 chapter, please read this chapter carefully (especially section 5.8.3) 2. 新增 mipi 转 lvds 芯片 tc358775 支持说明，详见 6.29 节。 Add the mipi to lvds chipset tc358775 support instruction, referring to section 6.29 3. 新增 V-BY-ONE 支持说明，详见 6.30 节。 Add V-BY-ONE support instruction, referring to section 6.30 4. 新增 DDR 带宽导致屏闪问题解决办法，详见 9.8 节 Add the solution of the flicker issue caused by ddr bandwidth, referring to section 9.8 5. 双屏异显添加基于 RK 方案的异触功能支持，详见 6.6 节。 Add different touch function support based on RK platform for dual-screen different display, referring to section 6.6	
V2.10	张文平 Zhang Wenping 董正勇 Dong	2019-7-16	1. 新增双屏异显副屏鼠标支持，详见 6.6.5。 Add usb mouse display on the secondary screen for dual-screen	

	Zhengyong		different display, referring to section 6.6.5. 2. DP 音频和 lpddr4 变频同时使能会有冲突，更新 DP 音频章节描述该问题，详见 6.28 节。 There are conflicts if enable dp audio and lpddr4 frequency scalling, referring to section 6.28. 3. 添加音频 dma 通道不够导致的声卡注册失败问题，详见 9.9 节。 Add the solution of the sound card register failed issue caused by lack of dma channel, referring to section 9.9 4. 删除 9.6 节“USB3.0 外设待机唤醒后重新枚举问题” Remove previous section 9.6 “USB3.0 peripherals re-enumerate issue after wake up from standby”	
V2.11	张文平 Zhang Wenping	2019-9-24	1. 新增 SDMMC 高速模式以及 SDIO3.0 配置说明章节，详见 9.9 节。 Add description of sdmmc high-speed mode and sdio3.0 mode, referring to section 9.9. 2. Camera 配置文件错误导致无法开机问题解决方法说明，详见 9.10 节。 Add description of hang on android logo because of camera configuration file error, referring to section 9.10. 3. 更新文档中的参考文档索引路径。 Update the reference documents file path.	
V2.12	张文平	2019-12-13	1. 更新 LPDDR4 配置说明，详见	

	Zhang Wenping		5.8 节。 Update LPDDR4 configure description, referring to section 5.8. 2. 新增 ddr 内存空间优化章节，详见 6.31 节。 Add decription of ddr memory space optimization, referring to 6.31. 3. 新增 PCBA 测试 UI 显示慢问题的解决方案，详见 9.11 节。 Add the solution of slow to display during PCBA test, referring to 9.11. 4. 新增 HDMI-IN 第三方 camera 应用兼容问题说明，详见 6.8.2 节。 Add description of HDMI-IN compatible issue with third party camera application, referring to section 6.8.2.	
V2.13	张文平 Zhang Wenping	2020-1-20	1. 新增 ISP 导致的开机卡死问题的解决方案，详见 9.12 节。 Add the solution of boot-up crash issue caused by ISP, referring to 9.12.	
V2.14	张文平 Zhang Wenping	2020-3-30	1. 修改和新增自动化测试和 debug 脚本章节内容，详见 6.22 节。 Modify and add the description of auto-test and debugging scripts, referring to section 6.22. 2. 新增 u-boot 设置 io-domain 章节，详见 9.1.2.2 节。 Add io-domain config in u-boot, referring to section 9.1.2.2. 3. 新增 efuse 标志传递到内核功能说明，详见 9.13 节。 Add the descrition of passing	

			efuse tag to kernel, referring to section 9.13.	

目 录 Contents

前 言 Preface

概述 Overview

本文档主要介绍 Rockchip RK3399 Android7.1 软件开发指南，旨在帮助软件开发工程师更快上手 RK3399 的开发及调试。

This document mainly describes Rockchip RK3399 Android7.1 software development guide aiming to help software engineers familiar with RK3399 development and debugging quickly.

产品版本 Product version

芯片名称 Chipset name	内核版本 kernel version	Android 版本 Android version
RK3399	Linux4.4	Android7.1.1

读者对象 Object

本文档（本指南）主要适用于以下工程师：

This document (guide) is mainly suitable for below engineers:

技术支持工程师

Field application engineers

软件开发工程师

Software development engineers

1 支持列表 Support list

1.1 DDR 支持列表 DDR support list

RK3399 DDR 目前选型列表支持双通道 DDR3、DDR3L、LPDDR3、LPDDR4。

RK3399 DDR current AVL supports dual channel DDR3, DDR3L, LPDDR3, LPDDR4.

Table 1-1 RK3399 DRAM Support Type

Chip	DRAM Support Type
RK3399	DDR3/DDR3L/LPDDR3/LPDDR4

RK3399 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。

RK3399 DDR component support level refers to 《RK DDR Support List Ver2.31》 in the directory of RKDocs\common\Platform support lists. Only recommend to use the components marked with the symbol √ and T/A as shown in below table.

Table 1-2 RK3399 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 EMMC support list

RK3399 支持 eMMC 5.1，SDIO3.0，可运行 HS200,HS400 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 DDR 支持程度表，只建议选用√、T/A 标示的颗粒。

RK3399 supports eMMC 5.1, SDIO3.0, and can run HS200, HS400 mode. For more details, please refer to 《RKeMMCSupportList Ver1.38_2018_01_22》 in the directory of RKDocs\common\Platform support lists. Only recommend to use the components marked with the symbol √ and T/A as shown in below table.

Table 1-3 RK3399 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 High performance eMMC component selection

为了提高系统性能，选取高性能的 EMMC 颗粒也是需要的。请在挑选 EMMC 颗粒前，参照我们的支持列表的型号，对应的研究下厂商提供的 Datasheet，重点关注下厂商标注的 performance 一章节。

It is necessary to select high performance EMMC component to improve system performance. Before selecting EMMC component, please refer to our AVL support list, study the corresponding datasheet from vendors, and especially pay attention to the performance chapter.

参照厂商大小、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。

Refer to the vendor and read/write rate to do the sorting. Recommend to choose the component with the sequential reading rate >200Mb/s and sequential writing rate >40Mb/s.

如有选型上的疑问，也可直接联系我们的 Fae 窗口。

Contact with our FAE if you have any questions about the component selection.

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

Picture 1-1 EMMC Performance example

1.3 WiFi/BT 支持列表 Wi-Fi/BT support list

RK3399 内核运行 Linux4.4，WiFi/BT 支持列表，详见 RKDocs\common\Platform support lists 目录下《Rockchip_WiFi_Situation_20171215.pdf》，下表中所标示为目前 RK3399 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，可先与 WiFi/BT 芯片原厂沟通，是否有可以稳定在 Linux4.4 运行的驱动程序，并能提供调试帮助。

RK3399 kernel is Linux4.4. For the Wi-Fi/BT support list, please refer to 《Rockchip_WiFi_Situation_20171215.pdf》 in the directory of RKDocs\common\Platform support lists. Below table shows the Wi-Fi/BT chipset list currently already verified in RK3399. Recommend to choose the components in the table. If want to debug other Wi-Fi/BT chipset, first need to communicate with Wi-Fi/BT vendor whether they can provide the driver program which can work on Linux4.4 stably and technical support during debugging.

另外后续我们会不断更新支持列表，如果疑问和建议可以与我们的 Fae 窗口联系。

Besides, we may keep upgrading the support list in future. You can contact with our FAE if there is any question or suggestion.

RK3399 Wi-Fi Situation

WiFi Chip	IFACE	IEEE 802.11 Standard	2.4GHz Band	5.0GHz Band	BT	GPS	NFC	11AC	SDIO3.0	MIMO	BT4.0	BT4.2	Android7.1
AP6330	SDIO	IEEE 802.11A/B/G/N	✓	✓	✓	×	×	×	×	×	✓	×	✓
AP6255	SDIO	IEEE 802.11A/B/G/N/AC	✓	✓	✓	×	×	✓	✓	×	✓	✓	✓
AP6354	SDIO	IEEE 802.11A/B/G/N/AC	✓	✓	✓	×	×	✓	✓	✓	✓	×	✓

1. √：支持 ×：不支持 注：空的表示没调过
2. 该列表仅适用kernel4.4

图 1-2 RK3399 目前大量测试的 Wifi/Bt 支持列表

Picture 1-2 RK3399 currently verified Wi-Fi/BT support list

1.4 SDK 软件包适用硬件列表 SDK software package applicable hardware list

本 SDK 是基于谷歌 Android7.1 64bit 系统，适配瑞芯微 RK3399 芯片的软件包，适用于 laptop 产品形态、Tablet 产品形态、sapphire（蓝宝石）& excavator（挖掘机）开发板、及其他基于 RK3399 平台开发的产品。

This SDK is compatible with RK3399 chipset software package based on Google Android7.1 64bit system. It is suitable for laptop product, Tablet product, sapphire & excavator development board, and other products developed based on RK3399 platform.

使用的是 sapphire（蓝宝石） & excavator（挖掘机）开发板的，kernel 配置可直接使用 rk3399-sapphire-excavator-edp.dts 进行配置。

If using sapphire & excavator development board, you can use rk3399-sapphire-excavator-edp.dts to config kernel directly.

参考《RK3399_VR&Tablet_V10_20160620》硬件设计（可以通过 FAE 窗口获取）的 TABLET 样机，kernel 配置可以参考：

For TABLET device designed according to《RK3399_VR&Tablet_V10_20160620》 (which can be acquired from FAE contact), you can configure kernel referring to below:

TABLET:rk3399-mid-818-android.dts

另外随 SDK 发布，附带了 Box 样机板，sapphire（蓝宝石） & excavator（挖掘机）开发板的硬件使用说明。

Besides, along with SDK release, the hardware usage instruction of Box demo board, sapphire & excavator development board is also released.

1.5 多媒体编解码支持列表 Multimedia encoder/decoder support list

RK3399 多媒体方面支持强大，支持 4K VP9 and 4K 10bits H265/H264 视频解码，高达 60fps，1080P 多格式视频解码 (WMV, MPEG-1/2/4, VP8)，1080P 视频编码，支持 H.264，VP8 格式，视频后期处理器：反交错、去噪、边缘/细节/色彩优化。

RK3399 has powerful multimedia which supports 4K VP9 and 4K 10bits H265/H264 video decoder up to 60fps, 1080P multi format video decoder (MWV, MPEG-1/2/4, VP8), 1080P video encoder, H.264, VP8 format, video post processor: de-interleaving, de-noising, edge/detail/color optimization.

具体的编解码支持列表，详见 RKDocs\rk3399 目录下《RK3399 Multimedia Codec Benchmark v1.0》。

For detailed encoder/decoder support list, refer to 《RK3399 Multimedia Codec Benchmark v1.0》 in the directory of RKDocs\rk3399.

2 文档/工具索引 Document/tool index

2.1 文档索引 Document index

RK3399 SDK 发布文档旨在帮助开发者快速上手开发及调试，文档中涉及的并不能涵盖所有的知识和问题。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。

RK3399 SDK release documents aim at helping developers familiar with development and debugging quickly. The documents may not cover all the knowledge and issues and the document list is also being updated continuously. Please contact our FAE if you have any question or requirement about the documents.

RK3399 SDK 中在 RKDocs 目录下附带了三大块的文档，分别为：android（android 相关开发文档），rk3399(3399 相关发布文档)，common（公共开发文档）；common 目录细分为内核驱动开发文档、uboot 开发文档、模块开发文档、Platform support lists（支持列表）、RKTools manuals（工具使用文档）等。

RK3399 SDK includes three kinds of documents in RKDocs directory, android(android related development documents), rk3399(RK3399 related release documents), and common(common development documents). Common directory consists of kernel driver development documents, uboot development document, module development documents, Platform support lists (support list), RKTools manuals (tool usage document) etc.

```
├── android
│   ├── project.config
│   ├── Rockchip_Developer_Guide_Android_New_Partition_Configuration_CN.pdf
│   ├── Rockchip_Developer_Guide_Crash_Issue_Quick_Analysis_CN.pdf
│   ├── Rockchip_Developer_Guide_PCBA_Test_Tool_CN&EN.pdf
│   ├── Rockchip_Introduction_Android_Widevine_Project_Start_Preparation_CN.pdf
│   ├── Rockchip_Introduction_Display_Parameter_Storage_and_Configuration_CN.pdf
│   ├── Rockchip_Introduction_High_Reliable_OTA_Usage_CN&EN.pdf
│   ├── Rockchip_User_Guide_Recovery_CN&EN.pdf
│   └── wifi
│       ├── Rockchip_Introduction_Android7.1_WIFI_Configuration_CN.pdf
│       └── Rockchip_Introduction_RealTek_WIFI_Driver_Porting_CN.pdf
├── common
│   ├── Audio
│   │   ├── Rockchip_Developer_Guide_Audio_Call_3A_Algorithm_Integration_and_Parameter_
Debugging_CN.pdf
│   │   ├── Rockchip_Developer_Guide_Linux4.4_Audio_CN.pdf
│   │   └── Rockchip_Developer_Guide_RK817_RK809_Codec_CN.pdf
│   ├── camera
│   │   ├── HAL1
│   │   │   ├── Camera_Document_Directory.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(3288&3368).pdf
    │   │   │   ├── RK_ISP10_Camera_User_Manual_v2.2.pdf
    │   │   │   ├── RKISPV1_Camera_Module_AVL_v1.7.pdf
    │   │   │   ├── Rockchip_Camera_AVL_v2.0_Package_20180515.7z
    │   │   │   ├── Rockchip_Introduction_RKISPV1_Camera_Driver_Debugging_Method_CN.pdf
    │   │   │   ├── Rockchip_Introduction_RKISPV1_Camera_FAQ_CN.pdf
    │   │   │   └── Rockchip SOFIA 3G-R_PMB8018(x3_C3230RK)_Camera_Module_AVL_v1.6_
20160226.pdf
    │   │   ├── HAL3
    │   │   │   ├── camera_engine_rkisp_user_manual_v2.2.pdf
    │   │   │   ├── camera_hal3_user_manual_v2.2.pdf
    │   │   │   ├── RKCIF_Driver_User_Manual_v1.0.pdf
    │   │   │   ├── RKISP1_IQ_Parameters_User_Guide_v1.2_20190821.pdf
    │   │   │   └── RKISP_Driver_User_Manual_v1.2.pdf
    │   │   └── README.txt
    │   ├── CRU
    │   │   └── Rockchip-Clock-Developer-Guide-RTOS-CN.pdf
    │   ├── DDR
    │   │   ├── Rockchip-Developer-Guide-DDR-CN.pdf
    │   │   ├── Rockchip-Developer-Guide-DDR-EN.pdf
    │   │   ├── Rockchip-Developer-Guide-DDR-Problem-Solution-CN.pdf
    │   │   ├── Rockchip-Developer-Guide-DDR-Problem-Solution-EN.pdf
    │   │   └── Rockchip-Developer-Guide-DDR-Verification-Process-CN.pdf
    │   ├── debug
    │   │   ├── RK3399-LOG-EXPLANATION.pdf
    │   │   ├── Rockchip_Quick_Start_Linux_Perf.pdf
    │   │   ├── Rockchip_Quick_Start_Linux_Streamline.pdf
    │   │   └── Rockchip_Quick_Start_Linux_Systrace.pdf
    │   ├── display
    │   │   ├── Rockchip_Developer_Guide_DRM_Panel_Porting_CN.pdf
    │   │   ├── Rockchip_Developer_Guide_Dual_Display_Rotation_Direction_Debugging_CN.pdf
    │   │   ├── Rockchip_Developer_Guide_HDMI_Based_on_DRM_Framework_CN.pdf
    │   │   ├── Rockchip_Introduction_Baseparameter_Storage_Format_CN.pdf
    │   │   ├── Rockchip_Introduction_DRM_Integration_Helper_CN.pdf
    │   │   └── Rockchip_User_Guide_Android_Display_Based_on_DRM_CN.pdf
    │   ├── DVFS
    │   │   ├── Rockchip-Developer-Guide-Linux4.4-CPUFreq-CN.pdf
    │   │   └── Rockchip-Developer-Guide-Linux4.4-Devfreq.pdf
    │   ├── GMAC
    │   │   └── Rockchip_Developer_Guide_Ethernet_CN.pdf
    │   ├── hdmi-in
    │   │   └── Rockchip_Developer_Guide_HDMI_IN_CN.pdf
```

```
    ├── I2C
    │   └── Rockchip-Developer-Guide-Linux-I2C.pdf
    ├── IO-Domain
    │   └── Rockchip-Developer-Guide-Linux-IO-DOMAIN-CN.pdf
    ├── Leds
    │   └── Rockchip_Introduction_Leds_GPIO_Configuration_for_Linux4.4_CN.pdf
    ├── MCU
    │   ├── Rockchip-Developer-Guide-linux4.4-MCU.pdf
    │   └── Rockchip-Developer-Guide-MCU-EN.pdf
    ├── MMC
    │   └── Rockchip-Developer-Guide-linux4.4-SDMMC-SDIO-eMMC.pdf
    ├── mobile-net
    │   ├── Rockchip_Introduction_3G_Data_Card_USB_File_Conversion_CN.pdf
    │   └── Rockchip_Introduction_3G_Dongle_Configuration_CN.pdf
    ├── other
    │   ├── RK3399-CPUINFO.pdf
    │   ├── RK3399-LOG-EXPLANATION.pdf
    │   └── Rockchip_Introduction_Browser_FAQ_CN.pdf
    ├── PCie
    │   └── Rockchip-Developer-Guide-linux4.4-PCIe.pdf
    ├── PIN-Ctrl
    │   └── Rockchip-Developer-Guide-Linux-Pin-Ctrl-CN.pdf
    ├── Platform support lists
    │   ├── RK3128 BOX  Hardware Design Guide V10-201410.pdf
    │   ├── RK_DDR_Support_List_Ver2.39.pdf
    │   ├── RKeMMCSupportList Ver1.43_2019_03_15.pdf
    │   ├── RKNandFlashSupportList Ver2.73_20180615.pdf
    │   ├── Rockchip_Camera_AVL_v2.0_Package.7z
    │   ├── Rockchip_Introduction_WiFi_Situation_CN.pdf
    │   └── Rockchip_Kodi_Support_List_CN.pdf
    ├── PMIC
    │   ├── Archive.zip
    │   ├── Rockchip-Developer-Guide-Power-Discrete-DCDC-Linux4.4.pdf
    │   ├── Rockchip-Developer-Guide-RK805.pdf
    │   ├── Rockchip_Developer_Guide_RK817_RK809_Fuel_Gauge_CN.pdf
    │   ├── Rockchip-Developer-Guide-RK818_6-Fuel-Gauge.pdf
    │   └── Rockchip-RK818-RK816-FG-Log-Description-linux4.4.pdf
    ├── power
    │   └── Rockchip_Developer_Guide_Sleep_and_Resume_CN.pdf
    ├── PWM
    │   ├── Rockchip-Developer-Guide-Linux-PWM-CN.pdf
    │   └── Rockchip_Developer_Guide_PWM_IR_CN.pdf
    ├── RKTools manuals
    │   ├── RKDevInfoWriteTool_User_Guide_V1.0.3.pdf
```

```
    │   │   ├── RKIQTool_User_Manual_v1.5-CH.pdf
    │   │   ├── RKIQTool_User_Manual_v1.5-EN.pdf
    │   │   ├── RK_Platform_apache_tomcat_ota_Server_Setup_Introduction.rar
    │   │   ├── Rockchip_Box_Factory_Test_Tool_V2.0.rar
    │   │   ├── Rockchip_Introduction_Image_Upgrading_Failure_Analysis_CN.pdf
    │   │   ├── Rockchip_Introduction_MP_Tool_Upgrading_and_Related_Issues_Debugging_CN.pd
f
    │   │   ├── Rockchip_Introduction_Parameter_File_Format_CN.pdf
    │   │   ├── Rockchip_Introduction_REPO_Mirror_Server_Build_and_Management_CN.pdf
    │   │   ├── Rockchip_Introduction_Stresstest_for_VR_CN.pdf
    │   │   ├── Rockchip_Introduction_WNpctool_Write_Tool_CN.pdf
    │   │   ├── Rockchip_User_Guide_Box_Factory_Test_Tool_CN.pdf
    │   │   ├── Rockchip_User_Guide_Keybox_Burning_EN.pdf
    │   │   ├── Rockchip_User_Guide_KeyWrite_CN.pdf
    │   │   ├── Rockchip_User_Guide_MP_Flashing_CN.pdf
    │   │   ├── Rockchip_User_Guide_RKDevInfoWriteTool_CN.pdf
    │   │   ├── Rockchip_User_Guide_RKDevInfoWriteTool_EN.pdf
    │   │   ├── Rockchip_User_Guide_RK_Platform_MP_Upgrading_CN.pdf
    │   │   ├── Rockchip_User_Manual_Android_Development_Tool_CN.pdf
    │   │   ├── Rockchip_User_Manual_RKIQTool_CN.pdf
    │   │   ├── Rockchip_User_Manual_RKIQTool_EN.pdf
    │   │   └── Rockchip_User_Manual_RKUpgrade_Dll_CN.pdf
    │   ├── security
    │   │   ├── Efuse process explain .pdf
    │   │   ├── RK3399_Efuse_Operation_Instructions_V1.00_20190214_EN.pdf
    │   │   ├── Rockchip_Developer_Guide_Secure_Boot_V1.1_20190603_CN.pdf
    │   │   ├── Rockchip_Developer_Guide_TEE_Secure_SDK_CN.pdf
    │   │   ├── Rockchip_RK3399_Introduction_Efuse_Operation_EN.pdf
    │   │   ├── Rockchip-Secure-Boot-Application-Note-V1.9.pdf
    │   │   └── Rockchip Vendor Storage Application Note.pdf
    │   ├── Sensors
    │   │   └── Rockchip_Developer_Guide_Sensors_CN.pdf
    │   ├── SPI
    │   │   └── Rockchip-Developer-Guide-linux4.4-SPI.pdf
    │   ├── Thermal
    │   │   ├── Rockchip-Developer-Guide-Linux4.4-Thermal-CN.pdf
    │   │   └── Rockchip-Developer-Guide-Linux4.4-Thermal-EN.pdf
    │   ├── TRUST
    │   │   ├── Rockchip-Developer-Guide-RK3308-System-Suspend.pdf
    │   │   └── Rockchip-Developer-Guide-Trust.pdf
    │   ├── UART
    │   │   ├── Rockchip-Developer-Guide-linux4.4-UART.pdf
    │   │   └── Rockchip-Developer-Guide-RT-Thread-UART.pdf
    │   ├── u-boot
```

```
│   │   ├── Rockchip-Developer-Guide-Linux-AB-System.pdf
│   │   ├── Rockchip-Developer-Guide-Trust.pdf
│   │   ├── Rockchip-Developer-Guide-Uboot-mmc-device-driver-analysis.pdf
│   │   └── Rockchip-Developer-Guide-UBoot-nextdev-CN.pdf
│   └── usb
│       ├── putty20190213_162833_1.log
│       ├── Rockchip-Developer-Guide-Linux4.4-RK3399-USB-DTS-CN.pdf
│       ├── Rockchip-Developer-Guide-Linux4.4-USB-CN.pdf
│       ├── Rockchip-Developer-Guide-Linux4.4-USB-FFS-Test-Demo-CN.pdf
│       ├── Rockchip-Developer-Guide-Linux4.4-USB-Gadget-UAC-CN.pdf
│       ├── Rockchip-Developer-Guide-USB-Initialization-Log-Analysis-CN.pdf
│       ├── Rockchip-Developer-Guide-USB-Performance-Analysis-CN.pdf
│       ├── Rockchip-Developer-Guide-USB-PHY-CN.pdf
│       └── Rockchip-Developer-Guide-USB-SQ-Test-CN.pdf
└── rk3399
    ├── patches
    │   ├── DoubleCamera-Example.rar
    │   ├── Dual_Touch_Patch_V1.0-20170724.zip
    │   ├── dual_wifi_patch_for_android7.1_and_android8.1.rar
    │   ├── Handwriting_Optimization_v1.0.rar
    │   ├── keystone_patch_for_rk3399_Industry_android7.1_v0.4_20171225.tar.gz
    │   ├── RK3399_SSD_Android_V1.1_20190307.rar
    │   ├── tochiba_tc358775_19.08.19.rar
    │   └── widevine_Patch.rar
    ├── Rockchip_RK3399_Developer_Guide_Android7.1_Software_CN&EN.pdf
    ├── Rockchip_RK3399_Development_Guide_Android7.1_CaffeOnACL_CN.pdf
    ├── Rockchip_RK3399_Instruction_Dual_Display_Audio_CN.pdf
    ├── Rockchip_RK3399_Instruction_SDK_Performance_CN&EN.pdf
    ├── Rockchip_RK3399_Introduction_Android7.1_MPI_Demo_CN.pdf
    ├── Rockchip_RK3399_Introduction_Multimedia_Codec_Benchmark_EN.pdf
    ├── Rockchip_RK3399_Introduction_Performance_Optimization_CN.pdf
    ├── Rockchip_RK3399_Introduction_Power_Consumption_Optimization_CN.pdf
    ├── Rockchip_RK3399_Introduction_Power_Consumption_Optimization_EN.pdf
    ├── Rockchip_RK3399_Introduction_SDK_Multimedia_Performance_EN.pdf
    ├── Rockchip_RK3399_Introduction_SSD_Android&Linux_CN.pdf
    └── Rockchip_RK3399_Introduction_USB_DTS_Configuration_CN.pdf
```

2.2 工具索引 Tool index

RK3399 SDK 发布的工具，用于开发调试阶段及量产阶段使用。工具可能随 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。

RK3399 SDK released tool is used in development debugging stage and MP stage. The tool may upgrade along with new SDK. Please contact with our FAE if there is any question or requirement about

the tool.

RK3399 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）。

RK3399 SDK contains linux (tool used in Linux operation system environment) and windows (tool used in Windows operation system environment) in RKTools directory.

```
RKTools
├── linux
│   ├── Linux_Pack_Firmware（Linux 固件打包工具）
│   ├── Linux_SecureBoot（Linux 固件签名工具）
│   ├── Linux_Upgrade_Tool（Linux 开发工具）
└── windows
    ├── AndroidTool（开发工具）
    │   ├── AndroidTool_Release_v2.38
    │   └── rockdev（固件打包工具）
    ├── DriverAssitant_v4.5（驱动安装助手）
    ├── Efuse_Tool_V1.36（Efuse 烧写工具）
    ├── FactoryTool-v1.42e.rar（工厂量产工具）
    ├── FWFactoryTool-5.3.rar（固件工厂工具）
    ├── OemTool_v1.2.rar（Demo 镜像制作工具）
    ├── SD_Firmware_Tool._v1.46.zip（SD 卡升级固件制作工具）
    ├── SecureBootTool_v1.83_foruser.rar（固件签名工具）
    ├── SpiImageTools_v1.36.zip
    └── UpgradeDllTool_v1.35.zip（厂商信息烧写工具—待更新版本）
```

3 SDK 编译/烧写 SDK compiling/flashing

3.1 SDK 获取 Acquire SDK

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《RK3399_ANDROID7.1-TABLET-SDK_V1.00 发布说明.pdf》。

SDK is released through Rockchip code server. Customers apply SDK from Rockchip FAE co ntact, and will be able to sync code after obtaining the server certificate authorization with SSH p ublic key. For more details about Rockchip code server SSH public key authorization, please refer to 《RK3399_ANDROID7.1-TABLET-SDK_V1.00_Release _Instruction.pdf》.

3.1.1 SDK 下载链接 SDK download link

RK3399_ANDROID7.1-Industry-SDK 下载地址如下：

RK3399_ANDROID7.1-Industry-SDK download address is as below:

```
repo init --repo-url=ssh://git@www.rockchip.com.cn:2222/repo-release/tools/repo.git -u ssh://git@w
ww.rockchip.com.cn:2222/rk3399-n-all/manifests.git -m rk3399_all_release.xml
```

repo 是 google 用 Python 脚本写的调用 git 的一个脚本，主要是用来下载、管理 Android 项目的软件仓库，其下载地址如下：

repo is a script invoking git developed by Google using Python script, and mainly used to download, manage Android project software lib. The download address is as below:

```
git clone ssh://git@www.rockchip.com.cn/repo/rk/tools/repo
```

3.1.2 repo

repo 是 google 用 Python 脚本写的调用 git 的一个脚本，主要是用来下载、管理 Android 项目的软件仓库，其下载地址如下：

repo is a script invoking git developed by Google using Python script, and mainly used to download, manage Android project software lib. The download address is as below:

```
git clone ssh://git@www.rockchip.com.cn/repo/rk/tools/repo
```

3.1.3 SDK 代码压缩包 SDK code compressed package

为方便客户快速获取 SDK 源码，瑞芯微技术窗口通常会提供对应版本的 SDK 初始压缩包，开发者可以通过这种方式，获得 SDK 代码的初始压缩包，该压缩包解压得到的源码，与通过 repo 下载的源码是一致的。以 Rk3399_Android7.1-Tablet-SDK.tar.gz 为例，拷贝到该初始化包后，通过如下命令可检出源码：

Rockchip FAE contact usually will provide the initial compressed package of the corresponding version SDK in order to help customers acquire SDK source code quickly. Developer can acquire the initial compressed package of SDK code in this way and unzip it to get the source code. It is the same as the

source code downloaded through repo. Take Rk3399_Android7.1-Tablet-SDK.tar.gz as an example, you can sync the source code through below command after copy the initial package:

```
mkdir rk3399
tar zxvf rk3399_android7.1_Industry_v1.0.tar.gz -C rk3399
cd rk3399
.repo/repo/repo sync -l
.repo/repo/repo sync
```

后续开发者可根据 Fae 窗口定期发布的更新说明，通过".repo/repo/repo sync"命令同步更新。

Developers can execute the command ".repo/repo/repo sync" to sync the new code according to the update notice released by FAE contact periodically in future.

3.2 SDK 编译 SDK compiling

3.2.1 JDK 安装 JDK installation

Android7.1 系统编译依赖于 JAVA 8。编译之前需安装 OpenJDK。

Android7.1 system compiling is dependent on JAVA 8. Need to install OpenJDK before compiling.

安装命令如下：

Install command is as below:

```
sudo apt-get install openjdk-8-jdk
```

配置 JAVA 环境变量，例如，安装路径为/usr/lib/jvm/java-8-openjdk-amd64，可在终端执行如下命令配置环境变量：

Configure JAVA environment variable, for example, if the install path is /usr/lib/jvm/java-8-openjdk-amd64, it is able to execute below command to configure environment variable at the termination:

```
export JAVA_HOME=/usr/lib/jvm/java-8-openjdk-amd64
export PATH=$JAVA_HOME/bin:$PATH
export CLASSPATH=.:$JAVA_HOME/lib:$JAVA_HOME/lib/tools.jar
```

SDK 带有 Open JDK8 的配置脚本，在工程根目录下，命名为 javaenv.sh。

SDK contains Open JDK8 configuration script named javaenv.sh in project root directory.

可直接执行以下命令，配置 JDK：

Directly execute below command to configure JDK:

```
source javaenv.sh
```

3.2.2 编译模式 Compilation mode

SDK 默认以 userdebug 模式编译。

SDK default compiling mode is userdebug.

使用 adb 时，**需要先执行 adb root ，adb disable-verity 关闭 system 分区的 verity 特性，重启后再执行 adb root, adb remount，进而进行 push 操作来 debug**。

While using adb, **first need to execute adb root, adb disable-verity to close the verity feature of the system partition, then execute adb root, adb remount after reboot, and then execute push operation to debug.**

3.2.3 挖掘机编译 Excavator compilation

uboot 编译:

uboot compiling:

```
cd  u-boot
make rk3399_defconfig
make ARCHV=aarch64
```

kernel 编译:

kernel compiling:

```
cd  kernel
make ARCH=arm64 rockchip_defconfig -j8
make ARCH=arm64 rk3399-sapphire-excavator-edp.img -j12
```

android 编译:

Android compiling:

```
source build/envsetup.sh
lunch rk3399_all-userdebug
make -j12
./mkimage.sh
```

3.2.4 固件生成步骤 Image build steps

执行./mkimage.sh 后，在 rockdev/Image-xxx/目录生成完整的固件包(xxx 是具体 lunch 的产品名)

The complete images package will be generated in rockdev/Image-xxx/ (xxx is the specific name of lunched product) directory after executing ./mkimage.sh.

```
rockdev/Image-xxx/
├── boot.img
├── kernel.img
├── misc.img
├── parameter.txt
├── recovery.img
├── resource.img
├── RK3399MiniLoaderAll.bin
├── system.img
├── trust.img
└── uboot.img
```

3.2.5 jack-server 配置 jack-server configuration

Android7.1 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：

Android7.1 system uses jack-server as java code compiler and may meet below errors during compiling:

```
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。

In this case it is mainly limited by jack-server compiler itself, one network port number cannot be used by multiple users at the same time.

也就是在服务器上协同开发过程中，多用户同时编译 Android7.1 时，需要配置各自使用不同的网络端口号。

That means multiple users need to configure different network port numbers separately during co-development in the server while they compile Android7.1 at the same time.

jack-server 的两个配置文件(yhx 为对应用户的用户名)，决定了它所使用的端口号：

The two configuration files (yhx corresponds to the user name) of jack-server determine its port number:

```
/home/yhx/.jack-server/config.properties
/home/yhx/.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。

The two configuration files need to configure two port numbers. One is server port number and the other is client port number. The port numbers in the two configuration files should match.

```
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：

Configure steps are as below:

1) 确保两个配置文件存在，并且权限设置为 0600：

   Confirm the two configuration files existing and set the authority as 0600:

```
chmod 0600 /home/yhx/.jack-server/config.properties
chmod 0600 /home/yhx/.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。

   If the two configuration files are not existing, please refer to below to create the two configuration files.

   config.properties 文件示例如下（端口号需按实际修改）：

   config.properties file example is as below (port number needs to be changed according to the actual):

```
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）：

   .jack-settings file example is as below (port number needs to be changed according to the actual):

```
# 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：

Change port number, please change service port and admin port as other port numbers and the port numbers in the two configuration files need to match. Example is as below:

```
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。

Re-compile Android, if error still occurs, try to modify other port number until compile successfully.

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：

If compile still cannot pass over 5-time modification, execute the command jack-admin dump-report, unzip the generated compressed package, and analyze the log. If there is below log, re-install 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’
```

3.2.6 全自动编译脚本 Fully automatic compiling script

如前几节所述，编译可大致分为 u-boot、kernel、android 三大部分进行编译，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。

As described above, the compilation mainly contains three parts compiling u-boot, kernel and android. In order to improve the compiling efficiency and lower down the possible mistake operation of manual compiling, this SDK integrates the fully automatic compiling script which is convenient for image compiling and backup.

1) 该全自动化编译脚本原始文件存放于：

The original file of the fully automatic compiling script is put in:

```
device/rockchip/RK3399/build-rk3399-all.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：

When repo sync, copy it to the project root directory through manifest:

```
<project path="device/rockchip/rk3399" name="rk/device/rockchip/rk3399" remote="rk" revision="rk32/mid/7.0/develop">
        <copyfile src="buildspec.mk" dest="buildspec.mk"/>
        <copyfile src="build-rk3399-all.sh" dest="build-rk3399-all.sh"/>
```

```
</project>
```

3) 修改 build-rk3399-all.sh 脚本中的特定变量以编出对应产品固件。

Modify the specific variable in build-rk3399-all.sh script to build out the corresponding product images.

```
KERNEL_DTS=rk3399-sapphire-excavator-edp
```

变量请按实际项目情况，对应修改：

Modify the variable according to the actual project situation:

KERNEL_DTS 变量指定编译 kernel 的产品板极配置；

KERNEL_DTS variable specifies the product board level configuration for kernel compiling.

Android 使用 build-rk3399-all.sh 编译时，请先进行下面两步操作：

When Android uses build-rk3399-all.sh to compile, please first execute below two operations:

```
source build/envsetup.sh
lunch rk3399_all-userdebug
```

也可在脚本中对应修改，可改为 rk3399_all-user 及其它配置：

Or you can modify to rk3399_all-user and other configuration accordingly in the script:

```
lunch rk3399_all-userdebug
```

4) 执行自动编译脚本：

Execute automatic compiling script:

```
source build.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。

The script will automatically configure JDK environment variable, compile u-boot, compile kernel, compile Android, then generate images and version information, and package them to be update.img

5) 脚本生成内容:

The script generated contents:

脚本会将编译生成的固件拷贝至：

The script will copy the compiled images to:

IMAGE/RK3399 *****_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。

the directory of IMAGE/RK3399 *****_RELEASE_TEST/IMAGES which path is subject to the actual generation. Each compiling will create new directory and save, automatically backup images version during debugging, and keep all the information of images version. Recommend to use this compiling script to generate images for every big version compilation. It includes much version information which is convenient to locate code status while debugging issues.

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。

update.img in the directory can be directly used to download and update Android development tool and factory flashing tool.

3.3 固件烧写 Image flashing

刷机说明详见 RKDocs\common\RKTools manuals 目录下《Rockchip_User_Manual_Android_Development_Tool_CN.pdf》。SDK 提供烧写工具，如下图所示。编译生成相应的固件后，进入烧写模式，即可进行刷机。对于已烧过其它固件的机器，可以选择重新烧录固件，或是选择低格设备，擦除 idb，然后进行刷机。

Flashing instruction refers to 《Android 开发工具手册.pdf》 in the directory of RKDocs\common\RKTools manuals. SDK provides flashing tools as shown in below picture. After compiling to generate corresponding images, enter flashing mode, it is able to flash images. For the devices with existing images, you can select to re-flash images, or format the device, erase idb, and then flash the images.

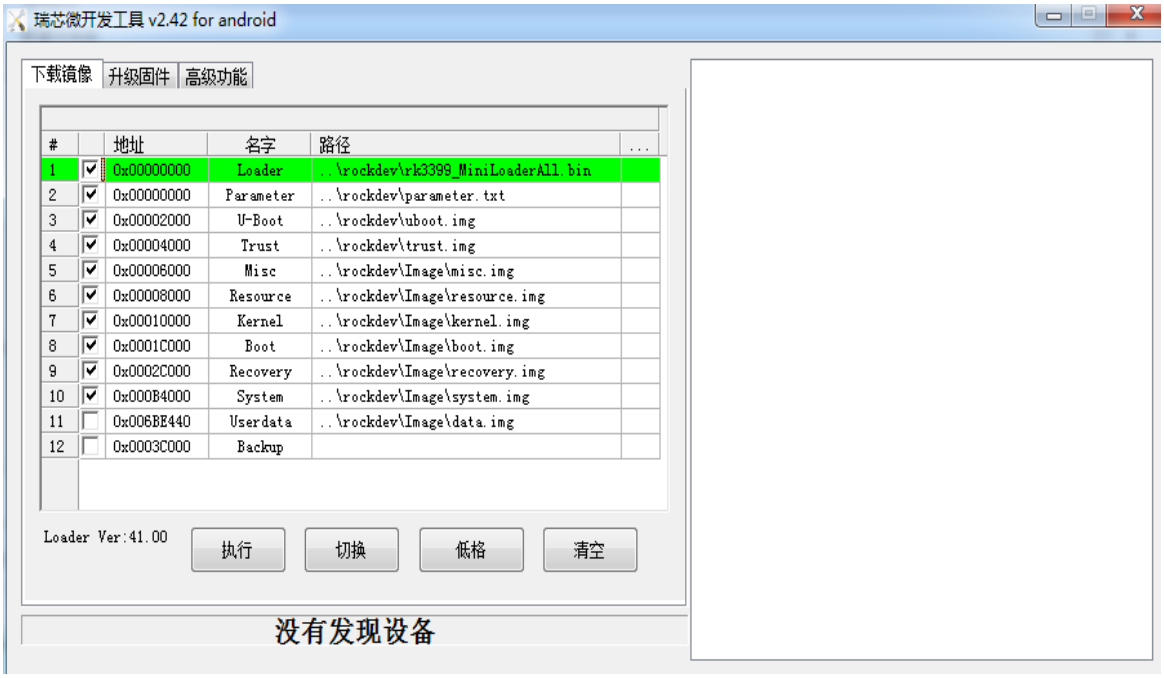


图 3-1Android 开发工具烧写界面

Picture 3-1 Android development tool flashing interface

注：烧写前，需安装最新的的 USB 驱动，驱动详见：

Note: Need to install the latest USB driver before flashing. The driver refers to:

```
RKTools/windows/
├── DriverAssitant_v4.5
```

3.4 量产烧写 MP flashing

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_User_Guide_MP_Flashing_CN.pdf》。

Considering the production efficiency and factory work station arrangement during MP, the flashing instruction refers to 《Rockchip_User_Guide_MP_Flashing_CN.pdf》 in the directory of RKDocs\ common\RKTools manuals.

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。

Please contact with our FAE if you have any tool related issues during production.

4 U-Boot 开发 U-Boot development

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 U-Boot 开发指南

V3.8-20170214.pdf》。

This chapter simply introduces U-Boot basic concept and compilation notices to help customers understand RK platforms U-Boot framework. For U-Boot development details, you can refer to 《Rockchip U-Boot 开发指南 V3.8-20170214.pdf》 in the directory of RKDocs\common\u-boot.

4.1 Rockchip U-Boot 简介 Rockchip U-Boot brief introduction

Rockchip U-Boot 是基于开源的 UBoot 2014.10 正式版进行开发的，主要支持：

Rockchip U-Boot is developed based on open source UBoot 2014.10 official version, and mainly supports:

- 支持芯片：rk3288、rk3036、rk312x、rk3368、rk322x、rk3366、rk3399 等；
- Support chipsets: rk3288, rk3036, rk312x, rk3368, rk322x, rk3366, rk3399 etc.
- 支持 Android 平台的固件启动；
- Support Android platform image bootup.
- 支持 ROCKUSB 和 Google Fastboot 两种方式烧写；
- Support ROCKUSB and Google Fastboot two methods of flasing.
- 支持 secure boot 固件签名加密保护机制；
- Support the encryption protection mechanism using secure boot to sign image.
- 支持 LVDS、EDP、MIPI、HDMI、CVBS 等显示设备；
- Support to display devices with LVDS, EDP, MIPI, HDMI, CVBS etc.
- 支持 SDCard、Emmc、Nand Flash、U 盘等存储设备；
- Support the memory devices such as SDCard, Emmc, Nand Flash, U disk etc.
- 支持开机 logo 显示、充电动画显示，低电管理、电源管理；
- Support boot up logo display, charging animation display, low battery management and power management.
- 支持 I2C、SPI、PMIC、CHARGE、GUAGE、USB、GPIO、PWM、DMA、GMAC、EMMC、NAND 中断等驱动；
- Support the drivers such as I2C, SPI, PMIC, CHARGE, GAUGE, USB, GPIO, PWM, DMA, GMAC, EMMC, NAND interrupt and so on.

4.2 平台配置 Platform configuration

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，并根据产品形态分为 MID 和 BOX 两种配置：

The platform configuration file is in the configs folder under U-Boot root directory. Rockchip related files begin with RK and can be divided into MID and BOX configuration according to the product types.

```
rk3288_defconfig
rk3126_defconfig
rk3128_defconfig
rk3368_defconfig
rk3399_defconfig

rk3288_box_defconfig
```

```
rk3128_box_defconfig
rk3036_box_defconfig
rk3368_box_defconfig
rk322x_box_defconfig
rk3399_box_defconfig
```

RK3399 Laptop/Tablet 选用的是 rk3399_defconfig 配置。

RK3399 Laptop/Tablet uses rk3399_defconfig configuration.

4.3 固件生成 Images generation

Rockchip 平台 Loader 分为一级模式和二级模式，根据不同的平台配置生成相应的 Loader 固件。通过宏 CONFIG_SECOND_LEVEL_BOOTLOADER 定义二级 Loader 模式。

Rockchip platform Loader mode is divided into level 1 and level 2. Generate the corresponding Loader image according to different platform configuration. Define level 2 Loader mode through macro CONFIG_SECOND_LEVEL_BOOTLOADER.

4.3.1 一级 Loader 模式 Level 1 Loader mode

U-BOOT 作为一级 Loader 模式，那么仅支持 EMMC 存储设备，编译完成后生成的镜像：

If U-Boot is as level 1 Loader mode, the image only supports EMMC memory device. The generated mirror after compiling:

```
rk3399_loader_v1.09.110.bin
```

其中 V1.09.110 是发布的版本号。

V1.09.110 is the released version number.

4.3.2 二级 Loader 模式 Level 2 Loader mode

U-Boot 作为二级 Loader 模式，那么固件支持所有的存储设备，该模式下，需要 MiniLoader 支持，通过宏 CONFIG_MERGER_MINILOADER 进行配置生成。同时引入 Arm Trusted Firmware 后会生成 trust image，这个通过宏 CONFIG_MERGER_TRUSTIMAGE 进行配置生成。

If U-Boot is as level 2 Loader mode, the image supports all the memory devices. In this mode, need MiniLoader support, through macro CONFIG_MERGER_MINILOADER to configure to generate. At the same time, it will generate trust image through macro CONFIG_MERGER_TRUSTIMAGE configuration after introducing Arm Trusted Firmware.

以 rk3399 编译生成的镜像为例：

Take the mirror generated by rk3399 compilation as an example:

```
rk3399_loader_v1.09.110.bin
uboot.img
trust.img
```

其中 V1.09.110 是发布的版本号，rockchip 定义 U-Boot loader 的版本，其中 1.09.110 是根据存储版本定义的，客户务必不要修改这个版本。

V1.09.110 is the released version number. U-Boot loader version is defined by rockchip. V1.09.110 is defined according to the memory version and should not be changed by customers.

uboot.img 是 U-Boot 作为二级 loader 的打包。

uboot.img is the package when U-Boot as level 2 loader.
trust.img 是 U-Boot 作为二级 loader 的打包。
trust.img is the package when U-Boot as level 2 loader.
RK3036、RK3126、RK3128、RK322x、RK3368、RK3366、RK3399 等采用二级 loader 模式。
RK3036, RK3126, RK3128, RK322x, RK3368, RK3366, RK3399 etc. use level 2 loader mode.

4.4 U-Boot 编译 U-Boot compilation

RK3399 SDK 编译使用的是如下配置：
RK3399 SDK compilation uses below configuration:

```
make rk3399_defconfig
make ARCHV=aarch64
```

编译完，会生成 trust.img、rk3399_loader_v1.09.110.bin、uboot.img 三个文件。
After compilation, it will generate trust.img, rk3399_loader_v1.09.110.bin, uboot.img the three files.
目前编译出来的 rk3399_loader_v1.09.110.bin DDR 为定频 800MHz 版本。
Currently compiled rk3399_loader_v1.09.110.bin DDR frequency is fixed as 800MHz.

4.5 U-Boot 充电相关配置 U-Boot charging related configuration

4.5.1 低电预充 Low battery pre-charging

u-boot 支持低电预充，需要在 uboot/include/configs/rk33plat.h 中打开如下开关，默认该功能是关闭的。

u-boot supports low battery pre-charging, need to enable the below switch in uboot/include/configs/rk33plat.h as it is disabled by default.

```
#define CONFIG_UBOOT_CHARGE
#define CONFIG_SCREEN_ON_VOL_THRESD 3400//3.4v
#define CONFIG_SYSTEM_ON_VOL_THRESD 3500//3.5v
```

其中 CONFIG_SCREEN_ON_VOL_THRESD 是系统点亮屏幕的电压门限，低于这个电压，禁止系统亮屏。CONFIG_SYSTEM_ON_VOL_THRESD 是系统正常启动的电压门限，低于这个电压，禁止 uboot 启动内核。这两个电压可以根据具体的产品设计灵活调整。

CONFIG_SCREEN_ON_VOL_THRESD is the voltage threshold for the system to light up the screen. If the voltage is lower than this value, it will prohibit the system to light up the screen. CONFIG_SYSTEM_ON_VOL_THRESD is the voltage threshold for the system to boot up. If the voltage is lower than this value, it will prohibit uboot to boot up kernel. The two voltages can be flexibly adjusted according to the product design.

产品板级 dts 中如下节点进行充电模式开关配置，可以灵活配置使用 uboot 还是 Android 的关机充电模式：

Use below nodes in the product board level dts to configure the charging mode. You can configure flexibly to use uboot or Android power off charging mode:

```
uboot-charge {
        compatible = "rockchip,uboot-charge";
        rockchip,uboot-charge-on = <0>;
```

```
                rockchip,android-charge-on = <1>;
        };
```

rockchip,uboot-charge-on 开关 uboot 阶段的充电动画，rockchip,android-charge-on 开关 android 充电动画。

rockchip,uboot-charge-on turn on/off the uboot charging animation and rockchip,android-charge-on turn on/off the android charging animation.

4.5.2 u-boot 充电图标显示 u-boot charging logo display

如果选择使用 uboot 阶段的充电动画，即 rockchip,uboot-charge-on = <1>时，还需要将动画图片资源文件打包在 resource.img，充电图标打包可以参考文档 RKDocs\common\u-boot\Rockchip U-Boot 开发指南 V3.8-20170214.pdf 8.1.1 章节，具体打包方法如下：

If choose to use uboot charging animation, that is rockchip,uboot-charge-on = <1>, also need to package the animation picture resource files in resource.img. Package charging logo refers to the 8.1.1 chapter of the document RKDocs\common\u-boot\Rockchip U-Boot 开发指南 V3.8-20170214.pdf. The detailed package method is as below:

- 进到 u-boot 目录下；

  Enter u-boot directory.

  拷贝充电图片到 tools\resource_tool\resources\images，这里面默认有充电图片，替换可以参考这里面的图片格式进行替换；

  Copy the charging logo to tools\resource_tool\resources\images. Replace the default charging logo here referring to the existing picture format.

- 执行如下打包命令：

  Execute below package command:

```
sudo ./tools/resource_tool/pack_resource.sh tools/resource_tool/resources/ ../kernel/resource.img resource.img tools/resource_tool/resource_tool
```

执行后会将 tools/resource_tool/resources/目录下的动画图片资源打包在../kernel/resource.img，生成新的 resource.img 在 uboot 根目录。

After execution, it will package the animation picture resource in the directory of tools/resource_tool/resources/ into ../kernel/resource.img and generate new resource.img in uboot root directory.

4.6 U-Boot logo 相关的配置 U-Boot logo related configuration

4.6.1 U-Boot logo 开关配置 U-Boot logo switch configuration

Sdk 默认开启 U-Boot logo 功能，以达到更快显示开机 logo 的目的：

SDK enables U-Boot logo function by default in order to display boot-up logo quickly:

```
rockchip,uboot-logo-on = <1>;
```

如果需要关闭这个功能，请在 kernel 的 dts 中设置 rockchip,uboot-logo-on = <0>;即可。

If need to close this function, please set rockchip,uboot-logo-on = <0>; in kernel dts.

4.6.2 U-Boot logo 图片更换 U-Boot logo picture change

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 resource.img 即可。

The two pictures of U-Boot logo are logo.bmp and logo_kernel.bmp in kernel root directory. If need to change, use the bmp with the same name to replace and then re-compile resource.img.

附：不一定要两张图片，可以只要一张，如果只有一张就保留 logo.bmp 这一张即可。

P.S.: there may be one picture instead of two. If only one picture, just need to keep logo.bmp.

4.7 Next-dev 分支 U-Boot U-Boot of Next-dev branch

4.7.1 Next-dev 分支 u-boot 简介 u-boot of Next-dev branch brief introduction

前面几节介绍基于rkdevelop分支的U-BOOT，但是有些客户对于u-boot有一些特殊的需求，例如u盘读写、有线网卡等功能，这些功能在rkdevelop上不支持，因此我们在u-boot代码上新开发了一个分支，称为next-dev，下面简单介绍下next-dev分支的u-boot相关功能。

The previous chapters introduce the U-BOOT of rkdevelop branch, but some customers have some special requirements on u-boot, such as u disk read/write, wired LAN etc. functions. These functions are not supported on rkdevelop, so we create a new branch named next-dev in u-boot code. Below we will introduce briefly u-boot related functions of next-dev branch.

next-dev是从U-Boot官方的v2017.09正式版本中切出来进行开发的版本。随着upstream的U-Boot功能越来越完善，以及我们实际产品上对U-Boot的需求更加多样，因此我们进行了nextdev开发。next-dev作为rkdevelop的升级版本，可以支持更多的功能，并且同U-Boot官方版本会每隔一段时间进行同步。目前在该平台上已经支持RK所有主流在售芯片。

Next-dev version is developed separately from U-Boot official version v2017.09. As upstreamed U-Boot functions are more and more complete, and our actual products have more requirements on U-Boot, we developed nextdev. Next-dev as the updated version of rkdevelop can support more functions, and will sync with U-Boot official version periodically. Currently this platform already supports all RK mainstream chipsets under selling.

目前支持的功能主要有：

Currently it mainly supports the functions:

- 支持RK Android平台的固件启动；
- Support RK Android platforms image bootup.
- 支持最新Android AOSP(如GVA)固件启动；
- Support the latest Android AOSP (such as GVA) image bootup.
- 支持Linux Distro固件启动；
- Support Linux Distro image bootup.
- 支持Rockchip miniloader和SPL/TPL两种pre-loader引导；
- Support Rockchip miniloader and SPL/TPL two kinds of pre-loader loading.
- 支持LVDS、EDP、MIPI、HDMI等显示设备；
- Support to display devices with LVDS, EDP, MIPI, HDMI etc.
- 支持Emmc、Nand Flash、SPI Nand flash、SPI NOR flash、SD卡、 U盘等存储设备启动；
- Support to bootup with the memory devices such as eMMC, Nand Flash, SPI Nand flash, SPI NOR flash, SD Card, U disk etc.

- 支持FAT、EXT2、EXT4文件系统；
- Support FAT, EXT2, EXT4 file system.
- 支持GPT、RK parameter分区格式；
- Support GPT, RK parameter partition format.
- 支持开机logo显示、充电动画显示，低电管理、电源管理；
- Support bootup logo display, charging animation display, low battery management, power management.
- 支持I2C、PMIC、CHARGE、GUAGE、USB、GPIO、PWM、GMAC、EMMC、NAND、中断等驱动；
- Support the drivers such as I2C, PMIC, CHARGE, GUAGE, USB, GPIO, PWM, GMAC, EMMC, NAND, interrupt etc.
- 支持RockUSB 和 Google Fastboot两种USB gadget烧写EMMC；
- Support RockUSB and Google Fastboot two USB gadget to flash EMMC.
- 支持Mass storage, ethernet, HID等USB设备；
- Support USB devices such as Mass storage, Ethernet, HID etc.
- 支持使用kernel的dtb；
- Support to use kernel dtb.
- 支持dtbo功能；
- Support dtbo function.

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。另外，我们提供了一份说明文档描述next-dev在开发中容易碰到的问题以及配置，请参考RKDocs/common/ u-boot/ Rockchip-Developer-Guide-UBoot-nextdev-CN.pdf。

The doc directory of U-Boot provides many README documents, which introduce for developers the concept, design philosophy and implementation method etc. of the function modules in U-Boot. Recommend users make good use of these documents to improve the development efficiency. Besides, we provide a document describing the common issues and configuration during next-dev development. Please refer to RKDocs/common/ u-boot/ Rockchip-Developer-Guide-UBoot-nextdev-CN.pdf.

4.7.2 Next-dev 分支 u-boot 开发 u-boot development of Next-dev branch

“RK3399_ANDROID7.1-Industry-SDK_V1.0”该 SDK 支持 next-dev 分支的 u-boot，代码路径和原来一样，但是客户默认同步下来是 rkdevelop 分支的 u-boot，如果需要切到 next-dev 分支，请确保 sdk 已经更新到 v1.4 版本以上，可按如下方法确认当前 sdk 版本：

“RK3399_ANDROID7.1-Industry-SDK_V1.0” this SDK supports u-boot of next-dev branch, the code path is same as before, but the default u-boot synced by customers is rkdevelop branch, if need to switch to next-dev branch, please ensure that sdk is already updated to v1.4 version or above, you can use below method to check current sdk version:

```
ls .repo/manifests/ -l
total 4
drwxrwxr-x 2 zwp zwp 4096 Aug 20 17:35 rk3399_all_release
lrwxrwxrwx 1 zwp zwp      47 Aug 20 17:36 rk3399_all_release.xml ->
rk3399_all_release/rk3399_all_v1.4_20180820.xml
```

并且按照如下方法切到 next-dev 分支：

And switch to next-dev branch according to below method:

```
cd u-boot
git fetch rk
git checkout –f rk/next-dev-release –b next-dev-release
```

编译方法为：

The compiling method is:

```
./make.sh rk3399
```

最终固件的生成还是用 android 目录下的 mkimage.sh 来生成，另外，假如客户使用 build-rk3399-all.sh 脚本来编译，请记得修改脚本里面 u-boot 对应的编译命令。

The final image is generated by mkimage.sh in android directory. Besides, if customers use build-rk3399-all.sh script to compile, please remember to modify the compiling command corresponding to u-boot in the script.

5 Kernel 开发 Kernel development

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。RK3399 kernel 版本是 4.4，config 配置文件统一为 arch/arm64/configs/ rockchip_defconfig，RK3399 的串口波特率为 1500000，调试时请保证设置准确。

This chapter simply introduces some kernel common configurations changes, mainly for dts configuration, to help customers to do some simple changes easier and more convenient. RK3399 kernel version is 4.4 and config files are unified as arch/arm64/configs/ rockchip_defconfig. RK3399 serial port baud rate is 1500000. Please make sure the setting correct for debugging.

5.1 DTS 介绍 DTS introduction

5.1.1 DTS 说明 DTS introduction

RK3399 的 dts 文件在 kernel/arch/arm64/boot/dts/rockchip/下，其中 rk3399.dtsi 是核心配置文件定义了平台相关的内容；RK3399-android.dtsi 是产品级配置文件定义了一些外围设备；具体的产品 dts 需要 include 这两个文件，如 Tablet 产品的 dts 文件 rk3399-mid-818-android.dts。产品的 dts 里面根据具体的产品需求配置 CPU、GPU、DDR 的频率和电压表；配置 io、屏、wifi、bt、sensor、温控、背光、电池、系统供电配置等等。

RK3399 dts file is in kernel/arch/arm64/boot/dts/rockchip/. Rk3399.dtsi is the core configuration file which defines the platform related contents. RK3399-android.dtsi is the product level configuration file which defines some peripheral devices. The product dts needs to include these two files, e.g. Tablet product dts file rk3399-mid-818-android.dts. Configure CPU, GPU, DDR frequency and voltage table in product dts according to the specific product requirement. Configure io, panel, wifi, bt, sensor, thermal control, backlight, battery, system power configuration etc.

挖掘机采用 rk3399-sapphire-excavator-edp.dts 这个文件。

Excavator uses the file rk3399-sapphire-excavator-edp.dts.

5.1.2 新增一个产品 DTS Create a new product DTS

Rk3399 的产品 dts 文件需放在 kernel/arch/arm64/boot/dts/rockchip/下。

RK3399 product dts file should be put in kernel/arch/arm64/boot/dts/rockchip/.

1、以 rk3399-sapphire-excavator-edp.dts 或 rk3399-mid-818-android.dts 为参照，拷贝一份 dts 文件命名为 rk3399-product.dts。

Take rk3399-sapphire-excavator-edp.dts or rk3399-mid-818-android.dts as reference, copy a dts file and name it as rk3399-product.dts.

2、修改 arch/arm64/boot/dts/rockchip/Makefile 文件，添加对应 dtb 声明：

Modify arch/arm64/boot/dts/rockchip/Makefile file, add the corresponding dtb statement:

```
+rk3399-product.dtb
```

3、修改编译脚本或编译命令。

Modify the compiling script or command.

4、重新编译内核。

Re-compile kernel.

5.2 USB 配置 USB configuration

RK3399 Type-c 模块需要外挂一个 fusb302 逻辑检测芯片来识别接入设备类型及 USB 的正反插。Fusb302 的软件驱动在 dts 里面的配置如下：

RK3399 Type-c module needs to add an external fusb302 logic detection chipset to identify the connected device type and USB plug and reverse. The configuration of fusb302 software driver in dts is as below:

```
fusb0: fusb30x@22 {
                compatible = "fairchild,fusb302";
                reg = <0x22>;
                pinctrl-names = "default";
                pinctrl-0 = <&fusb0_int>;
                int-n-gpios = <&gpio1 1 GPIO_ACTIVE_HIGH>;
                status = "okay";
        };
```

中断脚配置：

The configuration of interrupt pin:

```
&pinctrl {
fusb30x {
        fusb0_int: fusb0-int {
            rockchip,pins = <1 2 RK_FUNC_GPIO &pcfg_pull_up>;
        };
    };
}
```

在 fusb302 及 usb phy 检测区分接入 type-c 口的是哪一类设备（充电器、USB、OTG、DP 等）之后，通知系统，所以相关联的模块代码需要注册 fusb302 的 extcon notifier 来接收，需要在模块 dts 配置加入 extcon = <&fusb0>。

After fusb302 and usb phy identify the device type (charger, USB, OTG, DP etc.) connected to the type-c, it will inform the system, so related module code needs to register extcon notifier of fusb302 to receive, need to add extcon = <&fusb0> in the dts configuration of module.

如 rk818 dts 节点中加入 extcon = <&fusb0>，通过 fusb302 及 usb phy 检测区分充电器、USB、

OTG 的拔插后，rk818 模块决定相关的充电电流配置及 OTG 的开关。

For example, add extcon = <&fusb0> in rk818 dts node, after fusb302 and usb phy detect and identify the plug/unplug of charger, USB, OTG, rk818 module decides relative charging current configuration and OTG switch.

目前 sdk 参考 dts 中默认 enable 了 fusb302 的配置，如果产品未使用 type-c 接口、未使用 fusb302，产品 dts 中请 disabled 节点 tcphy0 及 fusb0，并将 USB 相关联的模块 dts 中 extcon = <&fusb0>改为 extcon = <&u2phy0>。

Currently fusb302 configuration is enabled by default in sdk reference dts, if product doesn't use type-c interface and fusb302, please disable the node tcphy0 and fusb0 in product dts, and change extcon = <&fusb0> to extcon = <&u2phy0> in the dts of USB related module.

5.3 WiFi 配置 Wi-Fi configuration

```
wireless-wlan {
        compatible = "wlan-platdata";
        rockchip,grf = <&grf>;
        wifi_chip_type = "ap6354";
        sdio_vref = <1800>;
        WIFI,host_wake_irq = <&gpio0 3 GPIO_ACTIVE_HIGH>; /* GPIO0_a3 */
        status = "okay";
    };
```

上面部分内容是 WiFi 的 dts 配置内容，主要包括电源控制、中断等功能脚的配置。下面将对各个配置项（一般客户只需要修改下面红色标出部分参数）的功能进行详细描述：

The above is the content of Wi-Fi dts configuration, mainly including the configuration of function pins such as the power control, interrupt etc.. The configuration items (generally customers only need to modify the parameters marked in red) function will be explained as below:

wifi_chip_type = " ap6354";

用来确认 WiFi 芯片型号，实际使用什么型号的 WiFi 需要在这里指定：

Use to check Wi-Fi chipset. Need to specify the actually used Wi-Fi model here:

sdio_vref = <1800>; //1800mv or 3300mv

这个配置项配置 WiFi 模组的 IO 参考电压值，根据实际硬件设计中提供给 WiFi 模组参考电压输入的电压值来进行设定，参考电压设置错误会导致 WiFi 通信异常，引起 WiFi 打不开或者工作不稳定。

This item configures IO reference voltage value of Wi-Fi module, set the input voltage value according to the Wi-Fi module reference voltage provided by the actual hardware design. The reference voltage set improperly will cause Wi-Fi communication abnormal, and then lead to Wi-Fi fail to work or work unstably.

WIFI,host_wake_irq = <&gpio0 3 GPIO_ACTIVE_HIGH>;

这个配置项是 WiFi 中断脚的配置，某些 WiFi 模组没有这个脚可以不用配置直接将此配置项注释掉。使用 Broadcom 的 WiFi，比如 AP6xxx 以及 RK90x 等模组都需要正确配置这 GPIO。

This item configures Wi-Fi interrupt pin. If some Wi-Fi module doesn't have this pin, directly comment it out without configuration. Broadcom Wi-Fi such as AP6xxx and RK90x etc. modules all need to configure this GPIO correctly.

Broadcom wifi AP6xxx 系统会使用此中断脚作为 WiFi 数据中断脚，此中断脚有异常将会导致

WiFi 无法正常工作。其它 WiFi，例如 RTL8723BS，在机器进入休眠时，如果有 WiFi 数据到来时此中断用来唤醒机器。此中断脚有异常并不会造成 WiFi 无法正常工作。

For Boardcom wifi AP6xxx system uses this interrupt pin as Wi-Fi data interrupt pin and Wi-Fi cannot work normally if there is problem with the interrupt pin. For other Wi-Fi, such as RTL8723BS, when the device is in sleep mode, the interrupt is used to wake up the device if there is Wi-Fi data coming. So the problem of the interrupt pin will not cause that Wi-Fi cannot work normally.

5.4 BT 配置 BT configuration

```
wireless-bluetooth {
        compatible = "bluetooth-platdata";
        //wifi-bt-power-toggle;
        uart_rts_gpios = <&gpio2 19 GPIO_ACTIVE_LOW>; /* GPIO2_C3 */
        pinctrl-names = "default", "rts_gpio";
        pinctrl-0 = <&uart0_rts>;
        pinctrl-1 = <&uart0_gpios>;
        //BT,power_gpio    = <&gpio3 19 GPIO_ACTIVE_HIGH>; /* GPIOx_xx */
        BT,reset_gpio       = <&gpio0 9 GPIO_ACTIVE_HIGH>; /* GPIO0_B1 */
        BT,wake_gpio         = <&gpio2 26 GPIO_ACTIVE_HIGH>; /* GPIO2_D2 */
        BT,wake_host_irq = <&gpio0 4 GPIO_ACTIVE_HIGH>; /* GPIO0_A4 */
        status = "okay";
    };
```

以上是 BT 在 dts 里面的配置，下面对常见可能需要修改的部分进行简单的说明：

Above is the BT configuration in dts. Simply introduce some common parts that may need to be modified as below:

BT,reset_gpio = <&gpio0 9 GPIO_ACTIVE_HIGH>;

这个配置项是关于 BT 的 RESET 脚配置，这个脚不同的 BT 模组不一定都有，具体以实际原理图为准。

This configuration item is about BT RESET pin configuration. Not all BT modules have this pin. Refer to the actual schematic.

BT,power_gpio = <&gpio3 19 GPIO_ACTIVE_HIGH>

这个配置项是关于 BT 的电源控制 GPIO 配置，高电平有效，具体以实际原理图为准。

This configuration item is about BT power control GPIO configuration, high level active, refer to the actual schematic.

BT,wake_gpio = <&gpio2 26 GPIO_ACTIVE_HIGH>;

这个配置项是关于 BT 的 WAKE 脚配置， 对应原理图中的 BT_WAKE 管脚，高电平有效。

This configuration item is about BT WAKE pin configuration, corresponding to BT_WAKE pin in the schematic, high level active.

BT,wake_host_irq = <&gpio0 4 GPIO_ACTIVE_HIGH>

这个配置项是关于 BT 的中断脚配置， 对应原理图中的 BT_HOST_WAKE 管脚，高电平有效。

This configuration item is about BT interrupt configuration, corresponding to BT_HOST_WAKE pin in the schematic, high level active.

默认 BT 使用 uart0 接口连接，uart0 的配置如下：

BT uses uart0 interface to connect by default. Uart0 configuration is as below:

```
&uart0 {
    pinctrl-names = "default";
    pinctrl-0 = <&uart0_xfer &uart0_cts>;
    status = "okay";
};
```

5.5 GPIO

RK3399 提供 5 组 GPIO(GPIO0~GPIO4)共 122 个，所有的 GPIO 都可以用作中断，GPIO0/GPIO1 可以作为系统唤醒脚，所有 GPIO 都可以软件配置为上拉或者下拉，所有 GPIO 默认为输入，GPIO 的驱动能力软件可以配置。

RK3399 provides 5 groups GPIO(GPIO0~GPIO4) total 122pcs. All GPIO can be used as interrupt. GPIO0/GPIO1 can be used as system wakeup pin. All GPIO can be pulled up or down by software configuration. All GPIO by default is input and the driver ability can be configured by software.

关于原理图上的 gpio 跟 dts 里面的 gpio 的对应关系，例如 GPIO4c0，那么对应的 dts 里面应该是“gpio4 16”。因为 GPIO4A 有 8 个 pin，GPIO4B 也有 8 个 pin，以此计算可得 c0 口就是 16，c1 口就是 17，以此类推；

As for the gpio corresponding relationship between schematic and dts, such as GPIO4c0, the corresponding gpio in dts should be “gpio4 16”. As GPIO4A has 8 pins, GPIO4B also has 8 pins, inferring in this way, we can know c0 port is 16, c1 port is 17, and so on.

GPIO 的使用请参考 RKDocs\common\driver\ 目录下《Rockchip-Developer-Guide-Linux-Pin-Ctrl-CN.pdf》。

GPIO usage refers to 《Rockchip-Developer-Guide-Linux-Pin-Ctrl-CN.pdf》 in the directory of RKDocs\common\driver\.

5.6 ARM、GPU、DDR 频率修改 ARM, GPU, DDR frequency change

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。

DVFS (Dynamic Voltage and Frequency Scaling) is a real-time voltage and frequency adjusting technology. Currently in kernel 4.4 CPU, GPU, DDR modules support DVFS.

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

CPUFreq is a set of framework model supporting dynamically adjusting CPU frequency and voltage defined by kernel developers. It can effectively lower down CPU power consumption and balance CPU performance at the same time.

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：

CPUFreq selects a suitable frequency for CPU through different frequency scaling strategies. Current kernel version provides below strategies:

- interactive：根据 CPU 负载动态调频调压；
  interactive: dynamically adjust frequency and voltage according to CPU load.
- conservative：保守策略，逐级调整频率和电压；

conservative: conservative strategy, adjust frequency and voltage step by step.

- ondemand：根据 CPU 负载动态调频调压，比 interactive 策略反应慢；
  ondemand: dynamically adjust frequency and voltage according to CPU load, slower than interactive.
- userspace：用户自己设置电压和频率，系统不会自动调整；
  userspace: user to set voltage and frequency, system doesn't automatically adjust.
- powersave：功耗优先，始终将频率设置在最低值；
  powersave: power consumption first, always set the frequency to the lowest value.
- performance：性能优先，始终将频率设置为最高值；
  performance: performance first, always set the frequency to the max value.

详细的模块功能及配置，请参考 RKDocs/common/DVFS / 目录下《Rockchip-Developer-Guide-Linux4.4-CPUFreq-CN.pdf》和《Rockchip-Developer-Guide-Linux4.4-Devfreq.pdf》文档。

The detailed module function and configuration refer to《Rockchip-Developer-Guide-Linux4.4-CPUFreq-CN.pdf》and《Rockchip-Developer-Guide-Linux4.4-Devfreq.pdf》 in the directory of RKDocs/common/ DVFS/.

A53/A72/GPU/DDR 分别有对应的调试接口，可以通过 ADB 命令进行操作，对应的接口目录如下：
A53/A72/GPU/DDR all have corresponding debugging interface which can be operated with ADB command. The corresponding interface contents are as below:

A53：/sys/devices/system/cpu/cpu0/cpufreq/
A72: /sys/devices/system/cpu/cpu4/cpufreq/
GPU: /sys/class/devfreq/ff9a0000.gpu/
DDR: /sys/class/devfreq/dmc/

这些目录下有如下类似节点：
These contents have below similar nodes:

- available_frequencies：显示支持的频率 show the supported frequency
- available_governors：显示支持的变频策略 show the supported frequency scaling strategy
- cur_freq：显示当前频率 show current frequency
- Governor：显示当前的变频策略 show current frequency scaling strategy
- max_freq：显示当前最高能跑的频率 show current supported max frequency
- min_freq：显示当前最低能跑的频率 show current supported min frequency

以 GPU 为例进行定频操作，流程如下：
Take GPU as example to do the fixed frequency operation. The process is as below:

- 查看支持哪些频率
  Check the supported frequencies
  cat /sys/class/devfreq/ff9a0000.gpu/available_frequencies
- 切换变频策略
  Switch the frequency scaling strategy
  echo userspace > /sys/class/devfreq/ff9a0000.gpu/governor
- 定频
  Fix the frequency
  echo 400000000 > /sys/class/devfreq/ff9a0000.gpu/userspace/set_freq
- 设置完后，查看当前频率
  Check current frequency after setting

cat /sys/class/devfreq/ff9a0000.gpu/cur_freq

5.7 温控配置 Thermal control configuration

RK3399 芯片的 ARM 核和 GPU 核分别带有温控传感器，可以实时监控 cpu 和 gpu 的温度，并通过算法来控制 cpu 和 gpu 的频率从而控制 cpu 和 gpu 的温度。每个产品的硬件设计和模具不同对应的散热情况也不同，可以通过 dts 中的如下配置进行适当的调整温控参数来适配产品：

RK3399 chipset ARM core and GPU core have separate thermal control sensors which can real-time monitor CPU and GPU temperature and then control CPU and GPU temperatures by controlling CPU and GPU frequency through algorithm. Each product's different hardware design and mold correspond to different heat dissipation situation. The following configurations in dts can be used to adjust thermal control parameters to fit the product:

设置温控开启的温度：

Set the temperature to enable the thermal control:

```
&threshold {
    temperature = <85000>; /* millicelsius */
};
```

设置温控上限温度：

Set the upper limit of thermal control temperature:

```
&target {
    temperature = <100000>; /* millicelsius */
};
```

设置软件关机温度：

Set the software shutdown temperature:

```
&soc_crit {
    temperature = <105000>; /* millicelsius */
};
```

配置硬件关机温度：

Configure the hardware shutdown temperature:

```
&tsadc {
    rockchip,hw-tshut-mode = <1>; /* tshut mode 0:CRU 1:GPIO */
    rockchip,hw-tshut-polarity = <1>; /* tshut polarity 0:LOW 1:HIGH */
    rockchip,hw-tshut-temp = <110000>;
    status = "okay";
};
```

温控的具体说明可以参考 RKDocs\common\Thermal 目录下《Rockchip-Developer-Guide-Linux4.4-Thermal-CN.pdf》。

The detailed thermal control instruction refers to 《Rockchip-Developer-Guide-Linux4.4-Thermal-EN.pdf》 in the directory of RKDocs\common\Thermal.

5.8 LPDDR4 配置 LPDDR4 configuration

【注意】lpddr4 版本更新过程中，有些配置有变更，所以请一定要更新 uboot、rkbin（如果 uboot

是 nextdev 分支的则需要更新这个目录）、kernel 这三个仓库，并且按照对应 sdk 的 rk3399-evb-rev3-android-lp4.dts 这个 dts 进行配置：

【Note】 Some configurations may be changed along with lpddr4 version update, so please remember to update the three libs: uboot, rkbin （ this directory should be updated if uboot is using nextdev branch ） , kernel, and configure the dts according to rk3399-evb-rev3-android-lp4.dts of the corresponding sdk:

- 从行业 SDK V2.6 开始，lpddr4 的频点从原来的 416/856 两档改为 328/416/666/856 这 4 挡，默认关闭负载变频。【注】如果为了功耗考虑，需要开启负载变频，请将 dmc 节点中的 auto-freq-en 配置为 1，但是需要进行稳定性压测。

  Starting from industry SDK V2.6, the frequency of lpddr4 is changed from two levels 416/856 to four levels 328/416/666/856, and the frequency scaling with loading is enabled. If considering power issue, can enable ddr frequency scaling with loading, that is, auto-freq-en in dmc node configured as 1, but we strongly recommending to do the stability testing.

- 行业 sdk v2.2 版本到 v2.5 版本，lpddr4 频率为 416M 和 856M，对应的 dts 按照 rk3399-evb-rev3-android-lp4.dts 进行配置。

  From industry sdk v2.2 to v2.5, lpddr4 frequency supports 416M and 856M,please modify you devicetree according to rk3399-evb-rev3-android-lp4.dts.（modify the corresponding dts according to the following instruction）. The command to check the version is:

- 行业 sdk v2.1 及以前的版本，lpddr4 频率为 400M 和 800M。

  For industry sdk v2.1 and former versions, lpddr4 frequency supports 400M and 800M.

- 如果机器已经出货，需要对 loader 进行升级，请按照 5.8.3 节进行修改。

  If the devices are already shipped out, and need to upgrade the loader to new version , please modify following section 5.8.3.

- 查看 sdk 版本的命令为：

  The command to check the version is:

```
ls .repo/manifests -l
total 4
drwxrwxr-x 2 zwp zwp 4096 May 28 11:12 rk3399_all_release
lrwxrwxrwx 1 zwp zwp      47 May 28 11:13 rk3399_all_release.xml ->
rk3399_all_release/rk3399_all_v2.2_20190528.xml
```

rk3399 使用 lpddr4 的 dts 配置请参考文件：arch/arm64/boot/dts/rockchip/rk3399-evb-rev3-android-lp4.dts，将该文件中的下述 3 个节点拷贝到对应的产品 dts 中：

RK3399 lpddr4 dts configuration refers to the file: arch/arm64/boot/dts/rockchip/rk3399-evb-rev3-android-lp4.dts. Just need to copy below three nodes in the file to the corresponding product dts:

```
&dfi {
        status = "okay";
};

&dmc {
    status = "okay";
    center-supply = <&vdd_center>;
```

```
        system-status-freq = <
            /*system status                freq(KHz)*/
            SYS_STATUS_NORMAL             666000
            SYS_STATUS_REBOOT             666000
            SYS_STATUS_SUSPEND            328000
            SYS_STATUS_VIDEO_1080P        666000
            SYS_STATUS_VIDEO_4K           856000
            SYS_STATUS_VIDEO_4K_10B       856000
            SYS_STATUS_PERFORMANCE        856000
            SYS_STATUS_BOOST              856000
            SYS_STATUS_DUALVIEW           856000
            SYS_STATUS_ISP                856000
        >;
        vop-bw-dmc-freq = <
        /* min_bw(MB/s) max_bw(MB/s) freq(KHz) */
            0           762         416000
            763         3012        666000
            3013        99999       856000
        >;
        auto-min-freq = <328000>;
        auto-freq-en = <0>;
    };

    &dmc_opp_table {
        compatible = "operating-points-v2";

        opp-200000000 {
            opp-hz = /bits/ 64 <200000000>;
            opp-microvolt = <900000>;
            status = "disabled";
        };
        opp-300000000 {
            opp-hz = /bits/ 64 <300000000>;
            opp-microvolt = <900000>;
            status = "disabled";
        };
        opp-328000000 {
            opp-hz = /bits/ 64 <328000000>;
            opp-microvolt = <900000>;
        };
        opp-400000000 {
            opp-hz = /bits/ 64 <400000000>;
            opp-microvolt = <900000>;
            status = "disabled";
```

```
        };
        opp-416000000 {
            opp-hz = /bits/ 64 <416000000>;
            opp-microvolt = <900000>;
        };
        opp-528000000 {
            opp-hz = /bits/ 64 <528000000>;
            opp-microvolt = <900000>;
            status = "disabled";
        };
        opp-666000000 {
            opp-hz = /bits/ 64 <666000000>;
            opp-microvolt = <900000>;
        };
        opp-600000000 {
            opp-hz = /bits/ 64 <600000000>;
            opp-microvolt = <900000>;
            status = "disabled";
        };
        opp-800000000 {
            opp-hz = /bits/ 64 <800000000>;
            opp-microvolt = <900000>;
            status = "disabled";
        };
        opp-856000000 {
            opp-hz = /bits/ 64 <856000000>;
            opp-microvolt = <900000>;
        };
        opp-928000000 {
            opp-hz = /bits/ 64 <928000000>;
            opp-microvolt = <900000>;
            status = "disabled";
        };
        opp-1056000000 {
            opp-hz = /bits/ 64 <1056000000>;
            opp-microvolt = <900000>;
            status = "disabled";
        };
    };
```

另外，请确认下面这个配置项已经在内核使能（SDK v1.1 版本已经使能）：

Besides, please confirm below configuration item is enabled in kernel (SDK v1.1 already enabled):

```
CONFIG_SND_SOC_ROCKCHIP_FORCE_SRAM=y
```

这里需要注意的是：

Here we need to pay attention to that:

1）lpddr4 我们只支持 400M 和 800M 两档频率（从 v2.2 版本开始，频率变成了 416M 和 856M），其他频率被 disabled 掉了，所以如果客户要使用同一个 dts 来支持 lpddr4 和其他类型的 ddr，则其他类型的 ddr 也将只有 400M 和 800M 的频率（从 v2.2 版本开始为 416M 和 856M），这个请务必注意；

lpddr4 only supports 400MHz and 800MHz (starting from v2.2, the frequency is changed to be 416M and 856M), other frequencies are disabled. If customers want to use the same dts to support lpddr4 and other ddr, other ddr will also only support 400MHz and 800MHz (416M and 856M from v2.2). Please pay attention to this.

2）以上配置默认开启 ddr 变频， lpddr4 的变频功能对声卡数量有所限制，说明如下：

Above configuration enables DDR frequency scaling function by default. Lpddr4 frequency scaling function has some limitation on the audio card number as described below:

如果要支持 Lpddr4 变频，则需要将音频 buffer 移到 sram 中，RK3399 的 sram 空间有限，可用空间 128k，目前预分配给单个音频流的空间为 32k，所以系统支持的上限声卡数最多只能 2 个（32k * 2 * 2，每个声卡包含 playback 和 capture），更多的声卡无法创建成功，除非减小单个流的预分配大小，但这也相对的减小了底下支持的 buffer size max，如果用户层使用声卡想设置更大 buffer 时将受限。需注意，USB 声卡由于未使用 dma，所以不在限制范围内，也就是说，可以有 2 个声卡（包含 hdmi、spdif、i2s 等接口的声卡）加上多个 usb 声卡。因此，接下来分成两种情况描述：

If need to support lpddr4 frequency scaling function, need to transfer audio buffer to sram. RK3399 sram space is limited, available space is 128k, currently pre-allocated space for single audio stream is 32k, so the system can support only 2 audio card at most (32k*2*2, each audio card includes playback and capture). More audio cards cannot be created successfully unless to decrease the single stream pre-allocated size. However, it also relatively decreases the buffer size max supported by bottom layer and it will be limited if user layer wants to set a larger buffer for audio card. Need to notice that, USB audio card is not subject to the limitation because it doesn't use dma. That means, you can use two audio cards (audio cards with hdmi, spdif, i2s etc. interfaces) and multiple USB audio cards. Therefore, the following description is divided into two cases:

5.8.1 需要 lpddr4 的变频 Need lpddr4 frequency scaling

如果需要 lpddr4 变频，则需要将音频 buffer 移到 sram 中，此时系统最多只能支持 2 个声卡，请按照如下方法进行配置：

If need lpddr4 frequency scaling function, need to transfer audio buffer to sram, and now the system only support 2 audio cards at most. Please follow below steps to confirgure:

1. dts 中添加 sram 节点

   Add sram node in dts

```
/* first 64k(0xff8c0000~0xff8d0000) for ddr and suspend */
iram: sram@ff8d0000 {
    compatible = "mmio-sram";
    reg = <0x0 0xff8d0000 0x0 0x20000>; /* 128k */
};
```

2. 相对应的产品 dts 中引用 iram 节点。

   Invoke iram node in the corresponding product dts.

```
&dmac_bus {
    iram = <&iram>;
    rockchip,force-iram;
};
```

5.8.2 不需要 lpddr4 变频 Do not need lpddr4 frequency scaling

由于 lpddr4 变频有 2 个声卡的限制，因此如果需要 3 个以上声卡，需要关闭 lpddr4 的变频，即在对应产品的 dts 中将 dmc 节点 disable，如下所示：

If need 3 or more audio cards, need to disable lpddr4 frequency scaling function due to the 2 audio cards limitation. That is, to disable the dmc node in the corresponding product dts as shown below:

```
&dmc {
        status = "disabled";
        … …
};
```

另外，需要确保在内核中删除掉 5.8.1 节中描述的 2 个配置：

Besides, must delete the two kernel configurations described in 5.8.1 chapter.

1. 删除 dts 中的如下配置：

   Delete the following configuration in dts:

   ```
   /* first 64k(0xff8c0000~0xff8d0000) for ddr and suspend */
   iram: sram@ff8d0000 {
      compatible = "mmio-sram";
      reg = <0x0 0xff8d0000 0x0 0x20000>; /* 128k */
   };
   ```

2. 删除 dts 中的如下配置：

   Delete the following configuration in dts:

   ```
   &dmac_bus {
      iram = <&iram>;
      rockchip,force-iram;
   };
   ```

5.8.3 ota 升级 v2.2 版本 ddr 补丁 update v2.2 ddr patch through ota

如果客户的软件基于 V2.1 之前的版本（包含 v2.1），并且机器已经出货，但是又需要更新到 V2.2 版本的 DDR 补丁，需要修改 2 个部分：

If customer software is based on V2.1 or previous versions (including v2.1), and the devices are already shipped out, need to change two parts for updating the DDR patch of V2.2:

1. dts 中 dmc 节点的配置，将 lpddr4 的频点从 400/800 改为 416/856，生成的 resource.img 需要更新。

   The configuration of dmc node in dts, change lpddr4 frequency from 400/800 to 416/856, and need to update the generated resource.img.

2. u-boot 更新到最新代码，生成的 MiniLoaderAll.bin 和 trust.img 需要更新。

   Update u-boot to the latest code, and need to update the generated MiniLoaderAll.bin and trust.img.

Sdk 默认的 ota 升级只会升级 loader、parameter 以外的其余固件，如果需要升级 loader，则必须按照文档“RKDocs/android/Rockchip_User_Guide_Recovery_CN&EN.pdf”中的 3.1 节进行处理。【注意】由于行业 SDK 的编译环境的不同，操作方式和上面的文档略有不同：

SDK default ota only updates the images except loader and parameter, if need to update loader, you must operate per section 3.1 of the document “RKDocs/android/Rockchip_User_Guide_Recovery_CN&EN.pdf”. [Note] The operation may be slightly

different from the above document due to the different compiling environment of industry SDK.

a) 将需要升级的 loader，修改名字为 RKMiniLoaderAll.bin（固件默认生成的固件名为 MiniLoaderAll.bin）。
Change the name of loader to be updated to RKMiniLoaderAll.bin (originally generated image name is MiniLoaderAll.bin)

b) 将 RKMiniLoaderAll.bin 拷贝到 device/rockchip/rk3399/rk3399_all/ota/loader 目录下，然后生成 ota 包，该 ota 包即包含了需要升级的 miniloader。
Copy RKMiniLoaderAll.bin to the directory of device/rockchip/rk3399/rk3399_all/ota/loader, and then generate ota package which includes miniloader to be updated.

c) 如果客户有使能行业 sdk 中的高可靠 ota 功能（参考 6.20 节），并且需要从 v2.2 之前的版本(比如 v2.1)ota 升级到 v2.2 之后的版本，需要一些特殊的修改，请单独在 redmine 上联系本文档作者。
If customers enable the high reliable ota function in industry sdk (referring to section 6.20), and need to update from previous versions (such as v2.1) to v2.2 or later versions through ota, need to do some special changes. Please separately contact with the author of this document through redmine.

6 Android 常见配置 Android common configuration

6.1 Android 产品配置 Android product configuration

6.1.1 lunch 选项说明 lunch option description

rk3399_all-userdebug: //rk3399 平台平板产品 userdebug（64 位） rk3399 platform tablet product userdebug (64bit)

rk3399_all-user: //rk3399 平台平板产品 user（64 位） rk3399 platform tablet product user (64bit)

6.1.2 添加一个新的产品 Create a new product

rk3399 平台支持平板、Laptop、Box 等产品形态，当需要添加一个新的产品时，可以基于已有的 rk3399_all 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：

RK3399 platform supports tablet, Laptop, Box etc. products. You can create a new product based on existing rk3399_all. Below take a new tablet product as example to describe the detailed steps:

- 新增文件夹 device/rockchip/rk3399/rk3399_xxx，基于 rk3399_all.mk 创建 rk3399_xxx.mk，将 rk3399_all 目录下的所有文件拷贝至 rk3399_xxx 目录下。
  Create folder device/rockchip/rk3399/rk3399_xxx, based on rk3399_all.mk to create rk3399_xxx.mk, copy all the files in the directory of rk3399_all to the directory of rk3399_xxx.
  cd device/rockchip/rk3399
  mkdir rk3399_xxx
  cp rk3399_all.mk ./rk3399_xxx.mk

cp rk3399_all/* rk3399_xxx/

- 在 device/rockchip/rk3399/AndroidProducts.mk 中添加：

  Add in device/rockchip/rk3399/AndroidProducts.mk:

  PRODUCT_MAKEFILES := \
  $(LOCAL_DIR)/rk3399.mk \
  $(LOCAL_DIR)/rk3399_all.mk \
  $(LOCAL_DIR)/rk3399_xxx.mk

- 在 vendorsetup.sh 中添加产品对应的 lunch 选项：

  Add the corresponding lunch option of the product in vendorsetup.sh:

  add_lunch_combo rk3399_all-userdebug
  add_lunch_combo rk3399_all-user
  add_lunch_combo rk3399_xxx-userdebug
  add_lunch_combo rk3399_xxx-user

- 修改 rk3399_ xxx.mk 及 rk3399_xxx 目录下的新产品所需要修改的配置。

  Modify the configurations for the new product in the directory of rk3399_ xxx.mk and rk3399_xxx.

6.2 常用功能配置说明 Common function configuration instruction

6.2.1 常用配置宏说明 Common configuration macro instruction

宏配置 Macro configuration	功能说明 Function instruction
BUILD_WITH_GOOGLE_MARKET	若为 true 则集成 GMS 包，false 不集成 If true, integrate GMS package, false not to integrate
BUILD_WITH_GOOGLE_MARKET_ALL	若为 true 集成 full 的 GMS 包，false 集成 mini 的 GMS 包 If true, integrate full GMS package, false to integrate mini GMS package
BUILD_WITH_GOOGLE_FRP	使能恢复出厂设置保护 FRP 功能 Enable FRP factory reset protection function
BUILD_WITH_FORCEENCRYPT	使能默认全盘加密 Enable default full disk encryption
PRODUCT_SYSTEM_VERITY	使能 Verified boot Enable Verified boot
BUILD_WITH_GMS_CER	GMS 认证配置选项 GMS certificate configuration option
BUILD_WITH_WIDEVINE	集成 Widevine level3 插件库 Integrate Widevine level3 plug-in library
BOARD_NFC_SUPPORT	使能 NFC 功能 Enable NFC function
BOARD_SENSOR_ST	选用 ST 的 sensor 框架 Select ST sensor framework

BOARD_SENSOR_MPU	选用 MPU 的 sensor 框架 Select MPU sensor framework
BOARD_SENSOR_MPU_VR	选用 MPU_VR 的 sensor 框架 Select MPU_VR sensor framework
BOARD_GRAVITY_SENSOR_SUPPORT	使能 G-Sensor Enable G-Sensor
BOARD_COMPASS_SENSOR_SUPPORT	使能 Compass Enable Compass
BOARD_GYROSCOPE_SENSOR_SUPPORT	使能陀螺仪 Gyroscope Enable Gyroscope
BOARD_PROXIMITY_SENSOR_SUPPORT	使能距离感应器 Enable P-sensor
BOARD_LIGHT_SENSOR_SUPPORT	使能光感应器 Enable the light sensor
BOARD_PRESSURE_SENSOR_SUPPORT	使能压力感应器 Enable the pressure sensor
BOARD_TEMPERATURE_SENSOR_SUPPORT	使能温度传感器 Enable the temperature sensor
BOARD_ENABLE_3G_DONGLE	使能 3G Dongle 功能 Enable 3G Dongle function
TARGET_ROCKCHIP_PCBATEST	使能 PCBA 测试 Enable PCBA test
BOOT_SHUTDOWN_ANIMATION_RINGING	使能开关机动画+铃声 Enable power on/off animation and tones
BOARD_SYSTEMIMAGE_PARTITION_SIZE	System 分区最大容量 System partition maximum capacity
BOARD_USE_AUDIO_3A	是否使能 AUDIO 3A 算法 Whether to enable AUDIO 3A algorithm

6.2.2 预装 APK Pre-install 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制进 Android 系统。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：

Android apk pre-install function means to install the third application prepared in advance into the Android system when configuring the product according to customer requirements. Pre-install can be divided into uninstall installation and non-uninstall installation. This chapter mainly describes the function of uninstall pre-install. The configuration steps are as below:

- 新增文件夹 device/rockchip/rk3399/rk3399_all/preinstall_del，要确认是在 TARGET_DEVICE_DIR 定义的目录，get_build_var TARGET_DEVICE_DIR 可以看到。

  Confirm the new folder device/rockchip/rk3399/rk3399_all/preinstall_del is in the directory defined by TARGET_DEVICE_DIR, which can be seen by get_build_var TARGET_DEVICE_DIR.

- 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。

Copy the third apk to be pre-installed to above folder, pay attention to that the file name of apk should use English and avoid blank letter.

编译结束后会在 out/target/product/rk3399_all/system/目录，生成 preinstall_del 文件夹，文件夹内包含了预制的第三方应用。烧录后，系统会自动安装这些应用到 data/app 目录。因此他们是可卸载的。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。

After compilation, it will generate preinstall_del folder in the directory of out/target/product/rk3399_all/system/, and the folder includes the pre-installed third apk. After flashing, the system will automatically install these apks to the directory of data/app. So they are able to be uninstalled. Need to pay attention to that, the applications in the directory of preinstall will automatically install after factory reset even if users uninstall them during use. If want to uninstall the pre-install applications permanently, you can name the above folder as preinstall_del_forever.

6.2.3 开/关机动画及铃声 Power on/off animation and tones

需要在产品的 makefile 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。

Need to configure BOOT_SHUTDOWN_ANIMATION_RINGING := true in the makefile of the product, and prepare the corresponding resource files as below. After compilation, the corresponding resource files will be copied to the corresponding out directory.

将开机铃声 复制到 device/rockchip/common/startup.wav (源码路径)

Copy the bootup tone to device/rockchip/common/startup.wav (source code path)

将关机铃声 复制到 device/rockchip/common/startup.wav (源码路径)

Copy the shutdown tone to device/rockchip/common/startup.wav (source code path)

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

Copy the bootup animation to device/rockchip/common/bootanimation.zip (source code path)

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

Copy the shutdown animation to device/rockchip/common/shutdownanimation.zip (source code path)

6.3 Parameter 说明 Parameter instruction

rk3399 Android 7.1 平台有平板、Box、Laptop 等产品形态，不同的产品形态可能需要不同的 parameter 参数，请参考 device/rockchip/rk3399/下子目录 rk3399_all 中的来相应修改配置，关于 parameter 中各个参数、分区情况细节，请参考\RKDocs\RKTools manuals\ Rockchip Parameter File Format Ver1.3.pdf。

RK3399 Android7.1 platform supports tablet, Box, Laptop etc. product types and different product types may need different parameter. Please modify the configurations accordingly referring to the sub directory of rk3399_all under device/rockchip/rk3399/. For the parameter and partition details, please refer to \RKDocs\RKTools manuals\ Rockchip Parameter File Format Ver1.3.pdf.

6.4 新增分区配置 New partition configuration

请参考\RKDocs\android\《Rockchip_Developer_Guide_Android_New_Partition_Configuration_CN.pdf》。

Please refer to \RKDocs\android\《Rockchip_Developer_Guide_Android_New_Partition_Configuration_CN.pdf》.

6.5 OTA 升级 OTA upgrade

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_User_Guide_Recovery_CN&EN.pdf》。

OTA (over the air) upgrade is the standard software upgrade method provided by Android system. It provides complete upgrading (full package) and incremental upgrading mode (difference package). You can upgrade locally or over the network. For the detailed OTA upgrade and Recovery mode function and configuration, please refer to 《Rockchip_User_Guide_Recovery_CN&EN.pdf》 in the directory of RKDocs\android.

6.6 双屏异显/异触功能 Dual-screen different display/different touch function

6.6.1 功能描述 Function description

RockChip（以下简称 RK） SDK 平台上支持接双显示设备异显功能，包含 Android Presentation 和 RK Dualscreen 两种方案。

RockChip (called as RK hereinafter) SDK platforms support to connect two display devices to display different interfaces, including two solutions of Android Presentation and RK Dualscreen.

Android Presentation 是 Google 提供的双屏方案，实现了 View 级别的 VOP 派发，逻辑均在同一个 APP 上进行控制，请参考 google 官方文档进行开发。

Android Presentation is the dual screen solution provided by Google, which realizing VOP distribution at View level, and all the logics are controlled by the same one APP. Please refer to Google official document to do the development.

RK dualscreen 则是实现了 APP 级别的 VOP 派发，异显的两部分分别是不同的 APP。

RK dualscreen realized VOP distribution at APP level, and the two parts of dual display are controlled by different APP.

Presentation 适用于对自身需求进行深入定制的方案，目前的双屏异触功能是基于此方案适配的；RK dualscreen 在满足深入定制方案下，支持快速集成多方 APP，进行功能整合，目前双屏异显双声功能是基于此方案适配，请参考文档：《RKDocs/rk3399/Rockchip_RK3399_Instruction_Dual_Display_Audio_CN.pdf》。客户请根据产品需求选择对应方案，下表为两个方案的简单说明。

Presentation is suitable for the solution with deeply customized self requirement. Current dual screen different touch fuction is fitted based on this solution. RK dualscreen supports the rapid integration of multiple APP and function integration besides the deep customization requirement. Current dual screen different display and dual sound function is fitted based on this solution. Please refer to the document《RKDocs/rk3399/ Rockchip_RK3399_Instruction_Dual_Display_Audio_CN.pdf》. Cutomers can select the corresponding solution according to the product requirement. Below table shows the simple instruction of these two solutions.

方案 Solution	双屏异触 Dual screen different touch	双屏异声 Dual screen different sound	控制逻辑 Control logic
RK dualscreen	支持（从 V2.2 版本开始默认支持，之前的版本需要打补丁） Suppot (support by default from V2.2, previous versions need to apply the patch)	支持 Support	两个独立的 app 显示在不同屏上 Two separate app displaying on different panels
Android Presentation	支持 Support	不支持 Not support	一个 app 控制不同 View 显示在不同屏上 One app controles different View to display on different panels

6.6.2 RK DUALSCREEN 配置方法 RK DUALSCREEN configuration method

代码默认已集成此功能，在系统 设置-显示-HDMI-双屏异显 中打开：

The code integrates the function by default, and you can open it in system setting-display-HDMI-Dual screen different display:

默认同时按音量“+”和音量“—”两个按键触发异显/同显模式切换；

Press volume “+” and volume “-“ at the same time will trigger the different/same display mode switch by default.

组合按键可通过设置属性 sys.dual_screen.keycodes 修改配置，如：

Group button can change the configuration by setting the property sys.dual_screen.keycodes, for example:

音量加减：sys.dual_screen.keycodes=24,25 其中 24,25 分别为按键的 android 键值。

Volume “+” “-“: sys.dual_screen.keycodes=24,25, among which 24, 25 are respectively the android key value of the buttons.

6.6.3 相关代码位置 Related code location

```
frameworks/base
frameworks/native
packages/apps/Settings
```

6.6.4 异触功能 Different touch function

双屏异触功能，请参考如下路径的补丁以及说明文档：

Please refer to the patch and instruction document below for dual screen different touch function:

RKDocs/rk3399/patches/Dual_Touch_Patch_V1.0-20170724.zip。

6.6.5 副屏鼠标支持 Support mouse display on secondary screen

鼠标坐标移动到屏幕边缘时，将自动切换显示到另外一个屏幕。需要 SDK 版本更新到 v2.4 版本。

When the mouse is moved to the edge of current screen, it will be automatically displayed in the center of the other screen. Need to update SDK to v2.4 to support this function.

6.7 多屏拼接功能 Multi-screen splicing function

6.7.1 功能描述 Function description

本 SDK 提供横向和纵向多屏拼接方案，在不超过 vop 尺寸的前提下，支持 vopb 和 vopl 之间任意比例拼接，方便客户根据产品需求进行选择。多屏拼接功能可以用于体感游戏显示设备、健身房、广告机等多显示场景。

This SDK provides horizontal and vertical multi-screen splicing solution, under the premise of not exceeding the vop size, supports any proportion splicing between vopb and vopl, which is convenient for customers to select according to product requirements. Multi-screen splicing function can be used in motion sensing game display device, gym, advertising machine and other multi-display scenarios.

6.7.2 相关代码位置 Related code location

```
device/rockchip/rk3399
hardware/rockchip/hwcomposer
hardware/rockchip/libgralloc
```

请确认代码已经更新到 v1.7 版本以上，可在.repo/repo/repo sync 后，按照如下方法确认当前版本：

Please make sure the code is already updated to v1.7 or higher version, after .repo/repo/repo

sync, confirm current version per below method:

```
$ ls .repo/manifests/rk3399_all_release.xml -l
lrwxrwxrwx 1 zwp zwp 47 Oct 22 11:52 .repo/manifests/rk3399_all_release.xml -> rk3399_all_release/rk3399_all_v1.7_20181016.xml
```

6.7.3 固定拼接配置方法 Fixed splicing configuration method

使用过程中客户不需要动态调整拼接比例和拼接方式，可以把拼接比例和拼接方式固定到系统固件里面。固定拼接配置：

Customers don't need to dynamically adjust the spicing proportion and method during use, so you can fix the spicing proportion and method to the system image. Fix the splicing configuration:

device/rockchip/rk3399/rk3399_all.mk 文件中打开拼接功能，make installclean 后重新编译固件。

Enable the splicing function in the file of device/rockchip/rk3399/rk3399_all.mk, and re-compile image after make installclean.

1. BOARD_MULTISCREEN_SPLICING := true 多屏拼接功能开关 multi-screen splicing function switch
2. BOARD_MULTISCREEN_SPLICING_MODE := 0 0 表示左右拼接，1 表示上下拼接 0 means horizontal splicing, 1 means vertical splicing
3. 多屏拼接相关属性配置 multi-screen splicing related properties configuration

```
#three 1080p screen splicing(1:1:1)
PRODUCT_PROPERTY_OVERRIDES += \
persist.sys.framebuffer.main=5760x1080@60 \
persist.sys.framebuffer.aux=5760x1080@60 \
persist.sys.dualModeRatioPri=2 \
persist.sys.dualModeRatioAux=1
```

1) persist.sys.framebuffer.main 是主屏 UI 布局的分辨率，persist.sys.framebuffer.aux 是副屏 UI 布局的分辨率。多屏拼接功能里这两个属性值必须一样，否则无法实现多屏拼接功能。例如 3 个屏 1920x1080 分辨率（左右拼接），为了每个屏显示效果好，推荐设为 5760x1080@60。

persist.sys.framebuffer.main is the UI resolution of the main screen, while persist.sys.framebuffer.aux is the UI resolution of the secondary screen. The two property values in multi-screen splicing function must be the same, otherwise it is not able to realize the multi-screen splicing function. For example, 3 panels with 1920x1080 resolution (horizontal splicing), in order to make each display have good effect, recomment to set the value as 5760x1080@60.

2) persist.sys.dualModeRatioPri 是主屏占的拼接比例，persist.sys.dualModeRatioAux 是副屏占的拼接比例。通常三个 1080p 的物理屏拼接，两个 mipi 屏在 vopb 上，hdmi 显示设备在 vopl 上。设置 mipi 为主屏，hdmi 设置为副屏，当配置 persist.sys.dualModeRatioPri=2 和 persist.sys.dualModeRatioAux=1（主屏：副屏=2:1）时，由于两个 mipi 屏是等分输出，所以三个屏的拼接比例为 1:1:1。

persist.sys.dualModeRatioPri is the splicing proportion occupied by the main screen, while persist.sys.dualModeRatioAux is the splicing proportion occupied by the secondary screen. Generally for three 1080p physical panels splicing, two mipi panles are on vopb, hdmi display device is on vopl. Set mipi as main screen, hdmi as secondary screen, when configure persist.sys.dualModeRatioPri=2 and persist.sys.dualModeRatioAux=1 (main screen : secondary screen=2:1), because two mipi panels output evenly , the splicing proportion of these three panels is 1:1:1.

注意：受 vop 的尺寸限制，主副屏 UI 布局尺寸不能超过 vop 的尺寸。

Note: UI size of the main and secondary screens cannot exceed vop size as it is limited by vop size.

	VOPB	VOPL
Max in/out	4096x2304/4096x2160	2048x1536/2048x1536

例如：主副屏的 UI 布局分辨率设置为 5760x1080@60，且主副屏比例设为 3:1，那么主屏的宽为 5760*3/4=4320，超过了 vopb 的最大宽 4096，因此主副屏不能设置为 3:1。

For example: UI resolution of the main and secondary screens is set as 5760x1080@60, and the proportion of the main and secondary screen is 3:1, then the width of the main screen is 5760*3/4=4320 which is exceeding the max width 4096 of vopb, therefore the proportion cannot be set as 3:1.

下面举个固定三等分屏左右拼接的例子，对应配置如下：

Here is an example of fixed trisection panel horizontal splicing, and the corresponding configuration is as below:

```
# multi-screen splicing
BOARD_MULTISCREEN_SPLICING := false
BOARD_MULTISCREEN_SPLICING_MODE := 0
#three 1080p screen splicing(1:1:1)
PRODUCT_PROPERTY_OVERRIDES += \
persist.sys.framebuffer.main=5760x1080@60 \
persist.sys.framebuffer.aux=5760x1080@60 \
persist.sys.dualModeRatioPri=2 \
persist.sys.dualModeRatioAux=1
```

6.7.4 APK 动态调整拼接配置方法 APK dynamically adjusting splicing configuration method

在 device/rockchip/rk3399/rk3399_all.mk 文件中打开拼接功能，然后 make installclean 后重新编译固件。

Enable the splicing function in the file of device/rockchip/rk3399/rk3399_all.mk, and re-compile image after make installclean.

1. BOARD_MULTISCREEN_SPLICING := true 拼接功能开关 splicing function switch
2. 动态设置拼接屏需要改动的 property 属性介绍：introduce the properties which should be modified for dynamically setting the panel splicing
   1) persist.sys.dualModeEnable 1 拼接使能开关 switch to enable splicing
   2) sys.hwc.compose_policy 0 关闭 hwc，用 gpu 合成 close hwc, use gpu to integrate
   3) persist.sys.dualModeTB 1 拼接模式选择 0 是左右拼接，1 是上下拼接 splicing mode selects 0 means horizontal splicing, while 1 means vertical splicing
   4) persist.sys.framebuffer.main 主屏 UI 布局的分辨率 UI resolution of the main screen
   5) persist.sys.framebuffer.aux 副屏 UI 布局的分辨率 UI resolution of the secondary screen
   6) persist.sys.dualModeRatioPri 2 主屏占总的比例 the proportion occupied by the main screen
   7) persist.sys.dualModeRatioAux 1 副屏占总的比例 the proportion occupied by the secondary screen

动态配置属性时，需要注意以下：

When dynamically configure the property, need to pay attention to below:

1. 拼接屏要求主副屏向上申请的 framebuffer size 一致，所以需要在切换分辨率的同时，设置 framebuffer size，具体命令如下：

Panel splicing requires framebuffer size applied by the main and secondary screens should be the same,

so need to set framebuffer size at the same time when switching the resolution. The specific command is as below:

setprop persist.sys.framebuffer.main 5760x1080@60

setprop persist.sys.framebuffer.aux 5760x1080@60

例如：

For example:

3 个 1080p 左右拼接，设置为：

Three 1080p horizontal splicing, set as:

setprop persist.sys.framebuffer.main 5760x1080@60

setprop persist.sys.framebuffer.aux 5760x1080@60

3 个 1080p 上下拼接，设置为：

Three 1080p vertical splicing, set as:

setprop persist.sys.framebuffer.main 1920x3240@60

setprop persist.sys.framebuffer.aux 1920x3240@60

2. 拼接比例

Splicing proportion

persist.sys.dualModeRatioPri=2

persist.sys.dualModeRatioAux=1

主屏：副屏 = 2:1

The main screen : the secondary screen = 2:1

注意：受 vop 的尺寸限制，显示尺寸不能超过 vop 的尺寸。

Note: the display size cannot exceed vop size due to vop size limitaion.

	VOPB	VOPL
Max in/out	4096x2304/4096x2160	2048x1536/2048x1536

例如：

For example:

三个 1080p 的物理屏，两个 mipi 屏在 VOPB 上，hdmi 显示设备在 VOPL 上，mipi 设置为主屏，hdmi 设置为副屏，假如主副屏比例为 3:1，那么主屏的宽为 5760*3/4=4320，超过了 4096。因此主副屏不能　设置为 3:1。

Three 1080p physical panels, two mipi panels are on VOPB, hdmi display device is on VOPL, set mipi as main screen, and hdmi as secondary screen. If the proportion of the main and secondary screen is 3:1, the width of the main screen is 5760*3/4=4320, which is exceeding 4096, so the proportion of the main and secondary cannot be set as 3:1.

属性设置完成后，需要重启 android，通过 stop;start 重启即可。

After the property setting is finished, need to restart android. Restart by stop;start.

6.8 HDMI IN 功能 HDMI IN function

6.8.1 功能描述 Function description

通过 HDMI 接口，将接入的设备画面显示到本机的显示画面中的功能方案，RK3399 平台底层已支持。

The function is to display the screen of the connected device through HDMI interface to local screen. Bottom layer of RK3399 platform already supports.

6.8.2 功能要求 Function requirement

1. 硬件上需要支持 HDMI IN 设计；
   Need to support HDMI IN from hardware design.

2. 需要安装 HDMI-IN 应用；
   Need to install HDMI-IN application.

   - HDMI IN apk 源码在：
     The source code of HDMI-IN apk is in:

```
RKDocs/rk3399/patches/HdmiInDemo_2019.09_v1.1.tar.gz
```

解压到 sdk 下，mm 命令可编译出 apk.

Unzip the source code to packages/app directory, and use mm command to compile apk.

   - HDMI IN 对第三方 camera 应用兼容性问题
     HDMI IN compatible issue with third party camera application

HDMI-IN 实现的原理为将 hdmi 输入信号通过转接芯片转换为 mipi 信号，输入主控并显示，软件上将这个 mipi 输入信号模拟为 camera 输入，这导致第三方 camera 应用以及专用的 hdmi-in apk 都能打开显示 HDMI IN。为了避免这种问题，我们在行业 SDK V2.6 版本上提交了相关补丁，默认只有 HDMI-IN demo apk 才能打开 HDMI-IN，其余 apk 不能打开，如果需打开其它特定 apk 能打开 HDMI-IN 需加上下面的补丁：

The implementation principle of HDMI-IN is to convert hdmi input signal to mipi signal through the conversion chip, transmit to SoC and then
display. This mipi input signal will be simulated as camera input in software, so third party camera application and specific hdmi-in apk can open and display HDMI IN. In order to avoid this kind of issue, we have submitted relative patches in industry SDK V2.6, only HDMI-IN demo apk can open HDMI-IN by default. If need to enable HDMI-IN for other specific apk, need to add the following patch:

```
frameworks/av$ git diff
diff --git a/services/camera/libcameraservice/CameraService.cpp
b/services/camera/libcameraservice/CameraService.cpp
index 6eb6063..548361a 100755
--- a/services/camera/libcameraservice/CameraService.cpp
+++ b/services/camera/libcameraservice/CameraService.cpp
@@ -576,7 +576,7 @@ Status CameraService::getNumberOfCameras(int32_t type, int32_t*
numCameras) {
        }
        property_get("sys.camera.callprocess", value, "");
        ALOGD("sys.camera.callprocess: %s", value);
-       if(strcmp(value, "teaonly.rk.droidipcam") && strcmp(value, "/system/bin/cameraserver"))
+       if(strcmp(value, "teaonly.rk.droidipcam")  && strcmp(value, "对应的 apk 包名") &&
strcmp(value, "/system/bin/cameraserver"))
            *numCameras = mNumberOfCameras - 1;
        return Status::ok();
```

```
}
```

【注】apk 中打开的 camera id 为最后一个 camera， 即 HDMI-IN 设备，如下所示：

[Note]The camera id opened in hdmi-in app is the last camera, which is hdmi-in device, such as below:

```
Camera.open(Camera.getNumberOfCameras（）- 1);
```

3. 需要根据实际的硬件连接，修改内核 dts 中 tc358749x 节点的 gpio 配置；
   Need to modify gpio configuration of tc358749x node in kernel dts according to the actual hardware connection.

4. 需要根据实际的硬件连接，修改 hardware/rockchip/camera/Config/cam_board_rk3399.xml 的配置。
   Need to modify the configuration of hardware/rockchip/camera/Config/cam_board_rk3399.xml according to the actual hardware connection.

6.8.3 相关代码位置 Related code location

```
Kernel/
hardware/rockchip/camera
```

6.9 主副屏旋转功能 Main and secondary screens rotation function

在双屏同显或者异显的场景下，由于屏幕的差别，例如主屏为物理横屏，副屏为物理竖屏，需要将主屏或者副屏进行旋转，为了更方便的进行配置，行业 sdk 在 v2.1 版本以后整理了相关的代码，请更新代码到 v2.1 版本，并且参考文档“RKDocs\common\display\Rockchip_Developer_Guide_Dual_Display_Rotation_Direction_Debugging_CN.pdf”进行调试。

In dual screen different or same display cases, due to the panel differences, such as main screen is physical landscape, secondary screen is physical portrait, need to rotate the main or secondary screen. In order to congifure conveniently, industry sdk updates the related code from v2.1. Please update the code to v2.1 and refer to the document “ RKDocs\common\display\Rockchip_Developer_Guide_Dual_Display_Rotation_Direction_Debugging_CN.pdf” for debugging.

6.10 深度学习 Deep learning

本 SDK 集成了深度学习框架 Caffe-on-ACL，源码路径为 external/caffe-on-acl/，具体使用方法请参考：RKDocs\rk3399\Rockchip_RK3399_Development_Guide_Android7.1_CaffeOnACL_CN.pdf。

This SDK integrates the deep learning framework Caffe-on-ACL, which source code path is external/caffe-on-acl/. For detailed usage, please refer to RKDocs\rk3399\Rockchip_RK3399_Development_Guide_Android7.1_CaffeOnACL_CN.pdf.

6.11 多路 camera 支持 Multi-camera support

SDK 支持同时接多路 camera，rk3399 最多支持两路 mipi 加多路 usb camera（产品设计请注意 usb 带宽限制），默认代码已经支持，app 样例源码请参考 RKDocs\rk3399\patchs\DoubleCamera-Example.rar。

SDK supports to connect multiple cameras at the same time. RK3399 supports at most two mipi and multiple usb cameras (please note the usb bandwidth limitation for product design). Which is already supported in default code. For app demo source code, please refer to RKDocs\rk3399\patchs\DoubleCamera-Example.rar.

6.12 音视频多路编解码 Multi-channel audio and video encoder/decoder

SDK 提供了整套接口以及 demo 源码，请参考文档：RKDocs\rk3399\Rockchip_RK3399_Introduction_Android7.1_MPI_Demo_CN.pdf 进行开发。

SDK provides a whole set of interfaces and demo source code. Please refer to the document "RKDocs\rk3399\Rockchip_RK3399_Introduction_Android7.1_MPI_Demo_CN.pdf" to do the development.

6.13 Audio 3A 算法 Audio 3A algorithm

本 SDK 源码已经集成了 AUDIO 3A 算法，默认该功能未启用，如果需要启用，请修改 device/rockchip/rk3399/BoardConfig.mk，将下面这个宏改为 true：

This SDK source code already integrates AUDIO 3A algorithm which is disabled by default. If need to enable it, please modify device/rockchip/rk3399/BoardConfig.mk. Change below macro as true:

```
diff --git a/BoardConfig.mk b/BoardConfig.mk
index 8f68933..8c420fc 100755
--- a/BoardConfig.mk
+++ b/BoardConfig.mk
@@ -42,7 +42,7 @@ TARGET_BOARD_PLATFORM_GPU := mali-t860
 BOARD_USE_DRM := true

 # for audio 3A algorithm
-BOARD_USE_AUDIO_3A := false
+BOARD_USE_AUDIO_3A := true
```

AUDIO 3A 对应的调试文档请参考 RKDocs/common/Rockchip_Developer_Guide_Audio_Call_3A_Algorithm_Integration_and_Parameter_Debugging_CN.pdf 。

For AUDIO 3A corresponding debugging document, please refer to RKDocs/common/Audio/Rockchip_Developer_Guide_Audio_Call_3A_Algorithm_Integration_and_Parameter_Debugging_CN.pdf.

6.14 双 wifi（station+ap） Dual wifi (station+ap)

SDK 提供范例实现供客户参考，补丁目录为：RKDocs\rk3399\patchs\dual_wifi_patch_for_android7.1_and_android8.1.rar。

SDK provides the implementation example as reference for customers. The patch directory is RKDocs\rk3399\patchs\ dual_wifi_patch_for_android7.1_and_android8.1.rar.

6.15 双以太网 Dual ethernet

SDK 支持双以太网卡，通过配置 device/rockchip/rk3399 目录下的 device.mk 文件，可以配置以太网的模式，有 3 种选择：nomal/multi/bridge，normal 表示单以太网，multi 表示普通的双以太网，bridge 表示双以太网的桥接方式。如下补丁，表示将以太网改为双网卡模式。

SDK supports dual Ethernet. You can configure Ethernet mode by configuring the device.mk file in the directory of device/rockchip/rk3399. There are three options: nomal/multi/bridge. Normal means single Ethernet, multi means normal dual Ethernet, and bridge means the bridging method of dual Ethernet. Below patch changes the Ethernet to dual Ethernet mode.

```
diff --git a/device.mk b/device.mk
index e37c3cf..f367a29 100755
--- a/device.mk
+++ b/device.mk
@@ -77,7 +77,7 @@ endif
 # normal for single ethernet, multi for two, bridge for lan bridge
 # You can change this value in system
 # It would work after reboot
-ADDITIONAL_DEFAULT_PROPERTIES += persist.net.ethernet.mode=normal
+ADDITIONAL_DEFAULT_PROPERTIES += persist.net.ethernet.mode=multi
```

6.16 手写优化 Handwritten optimization

手写优化相关的补丁和说明，请参考 RKDocs/rk3399/patches/ Handwriting_Optimization_v1.0.rar。

For handwritten optimization related patch and instruction, please refer to RKDocs/rk3399/patches/ Handwriting_Optimization_v1.0.rar.

6.17 梯形校正功能 Trapezoid correction function

SDK 支持通过 gpu 对输出进行梯形变换来达到投影设备梯形校正的效果，例如车载投影设备投影到前窗玻璃时，由于玻璃平面和投影设备间的角度以及投影面的不规则，会导致输出图像存在梯形畸变，该补丁可以让客户进行畸变的调校，具体的补丁和说明请参考：

SDK supports trapezoidal transformation of output by gpu to achieve the trapezoidal correction effect of the projection device. For example, when the projection device of the vehicle is projected onto the front glass, due to the angle between the glass plane and the projection device and the irregularity of the projection plane, there will be trapezoidal distortion in the output image, this patch allows customers to do

the distortion tuning. The specific patch and instruction refer to:

RKDocs/rk3399/patches/keystone_patch_for_rk3399_Industry_android7.1_v0.4_20171225.tar.gz。

6.18 基于深度学习的目标检测（SSD）方案 Object detection SSD solution based on deep learning

SDK 提供基于深度学习的目标检测（SSD）优化方案，可为 AI 人工智能行业提供准 Turnkey 解决方案，具体的使用方法请参考文档：RKDocs/rk3399/Rockchip_RK3399_Introduction_SSD_Android&Linux_CN.pdf。另外，demo 应用源码请参考 RKDocs/rk3399/patches/ RK3399_SSD_Android_V1.1_20190307.rar，该源码需要使用 Android Studio 进行编译。

SDK provides the optimized object detection (SSD) solution based on deep learning, which can provide Turnkey solution for AI industry. For the specific usage, please refer to the document "RKDocs/rk3399/ Rockchip_RK3399_Introduction_SSD_Android&Linux_CN.pdf". Besides, for demo source code, please refer to RKDocs/rk3399/patches/RK3399_SSD_Android_V1.0_20180613.rar, and the source code should use Android Studio to compile.

6.19 Widevine 补丁 Widevine patch

SDK 支持 Widevine 功能，如果需要支持 widevine 功能，请先参考文档：RKDocs/android/Rockchip_Introduction_Android_Widevine_Project_Start_Preparation_CN.pdf 进行相关准备工作，另外相关 patch 请参考 RKDocs/rk3399/patches/ widevine_Patch.rar。

SDK supports Widevine function. If need to support widevine function, please first refer to the document "RKDocs/android/Rockchip_Introduction_Android_Widevine_Project_Start_Preparation_CN.pdf" to do the preparation. Besides, for related patch, please refer to RKDocs/rk3399/patches/widevine_Patch.rar.

6.20 高可靠 OTA High reliable OTA

SDK 支持高可靠 OTA 升级方法，可以防止机器在升级过程中掉电等异常操作导致的机器再也无法启动问题，SDK 默认为不使能，如果要开启该功能请参考文档：RKDocs/android/Rockchip_Introduction_High_Reliable_OTA_Usage_CN&EN.pdf。

SDK supports high reliable OTA upgrade which can prevent the device from failing to bootup due to power down or other abnormal operations during upgrading. SDK disables this function by default. If need to enable it, please refer to the reference document "RKDocs/android/Rockchip_Introduction_High_Reliable_OTA_Usage_CN&EN.pdf".

6.21 显示参数的调整和保存 Adjust and save the display parameter

RK 平台提供一个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 baseparaemter 分区，有以下两种功能：

RK platforms provide a partition to save the display parameters. The system can read and use the

display parameters from this partition. We call this partition baseparaemter partition which has below two functions:

- 保存和调整 LCD 色温、对比度、保护度、色度等信息；

  Save and adjust LCD color temperature, contrast, protection degree, chrominance and other information.

- 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；

  Save and adjust the resolution, timing and other information supported by hdmi or dp and other display devices.

如果客户需要使用上面的功能，请按照下面的方法使能 BaseParameter 分区功能：

If customers need to use above functions, please enable the partition function of BaseParameter per below method:

```
device/rockchip/common$ git diff
diff --git a/BoardConfig.mk b/BoardConfig.mk
index afeb107..b371b97 100755
--- a/BoardConfig.mk
+++ b/BoardConfig.mk
@@ -56,7 +56,7 @@ TARGET_CPU_ABI2 ?=
 TARGET_CPU_SMP ?= true
 endif

-BOARD_BASEPARAMETER_SUPPORT ?= false
+BOARD_BASEPARAMETER_SUPPORT ?= true

 ifeq ($(strip $(BOARD_BASEPARAMETER_SUPPORT)), true)
     TARGET_RECOVERY_OVERSCAN_PERCENT := 2
```

系统会编译生成对应的 baseparameter.img 固件，但是默认的烧写工具中不包含这个分区，请根据文档:《RKDocs/android/ Rockchip_Developer_Guide_Android_New_Partition_Configuration_CN.pdf》增加一个分区。

The system will compile to generate the corresponding baseparameter.img, but default flashing tool doesn't include this partition, please add a partition referring to the document 《RKDocs/android/ Rockchip_Developer_Guide_Android_New_Partition_Configuration_CN.pdf》.

baseparameter 分区功能更为详细的介绍请参考文档：《RKDocs/android/ Rockchip_Introduction_Display_Parameter_Storage_and_Configuration_CN.pdf》。另外，有些客户有在运行时动态修改这些显示参数的需求，因此我们在框架层增加了一些接口以方便客户进行调用，具体也可以参考上述文档进行设计。

For more detailed partition function instruction of baseparameter, please refer to the document 《RKDocs/android/ Rockchip_Introduction_Display_Parameter_Storage_and_Configuration_CN.pdf》. Besides, some customers may have the requirement to dynamically modify these display parameters during running, so we add some interfaces in framework layer for customers to invoke conveniently. You can also refer to above document for the specific design.

6.22 自动化测试和 debug 脚本 Automatic test and debug script

针对自动化测试，我们提供了 StressTest apk 给客户，详情请见 8.1 节 StressTest，为了覆盖更多的测试场景，我们新增了自动化测试和 debug 脚本供客户使用，测试脚本位于如下代码目录中：

As for automatic test, we provide StressTest apk for customers, referring to section 8.1 StressTest. In order to cover more test cases, we add automatic test and debug script for customers. The test script is in the directory of below code:

```
device/rockchip/rk3399/rockchip_test
```

目录结构如下：

The directory structure is as below:

```
├── ddr
│   ├── ddr_test.sh
│   ├── libstlport.so
│   ├── memtester_32bit
│   ├── memtester_64bit
│   ├── memtester_test.sh
│   ├── stressapptest
│   └── stressapptest_test.sh
├── dvfs
│   └── dvfs_test.sh
├── rockchip_test.sh
└── system_monitor
    ├── gpio.sh
    ├── hardware_monitor.sh
    └── memory_monitor.sh
```

用户可以使用如下方式调用这些脚本：

Users can use below method to invoke these scripts:

```
sh /system/bin/rockchip_test.sh
*******************************************************
***                                                 ***
***          ********************                   ***
***         *ROCKCHIPS TEST TOOLS*                      ***
***          *                   *                  ***
***          ********************                   ***
***                                                 ***
*******************************************************
*******************************************************
ddr test :              1 (memtester & stressapptest)
dvfs_test:          2 (dvfs stresstest, including cpu/gpu/ddr)
memory_monitor:         3 (tools used to detect memory leak)
hardware_monitor:        4 (tools used to monitor cpu/gpu/ddr freq and temperature)
*******************************************************
please input your test moudle:
```

输入对应的选择，即会跑对应的脚本，详细解释如下：

Input the corresponding choice, and then it will execute the corresponding script. The detailed explanation is as below:

- Ddr test: ddr 压力测试工具 memtester 和 stressapptest，可用于测试 ddr 稳定性，既可以使用上面的 rockchip_test.sh 调用该脚本，也可以单独执行脚本：
  DDR test: DDR stress test tool memtester and stressapptest, can be used to test ddr stability. Please use rockchip_test.sh to call this script, or use the following command to execute this script alone:

```
sh /system/bin/stressapptest_test.sh &
sh /system/bin/memtester_test.sh &
```

- dvfs_test.sh：测试 cpu、gpu、ddr 变频测试，测试变频的稳定性。
  dvfs_test.sh: test CPU, GPU, DDR frequency scaling, and test the stability of frequency scaling.

```
sh /system/bin/dvfs_test.sh
**********stress dvfs test**********
this chip is    rk3399

Usage: /system/bin/dvfs_test.sh [OPTION]...
   -c cpu freqence conversion test
   -g gpu freqence conversion test
   -d ddr freqence conversion test
   -n delay n seconds after every operation
```

  按照上面的提示执行该 dvfs 脚本，该脚本可以和其他测试配合使用，例如捕鱼游戏或者高负载的一些应用。
  Please use the script as above, and you can also run this script together with other apps, such as FISH.apk or other high load applications.

- memory_monitor.sh： 该脚本用于检测内存泄露问题，会将内存使用情况打印出来，使用方法如下：
  memory_monitor.sh: this script is used to detect the memory leakage issue, it will print out the memory usage to shell. The method to use this script is as follows:

```
sh /system/bin/memory_monitor.sh -h
Usage: /system/bin/memory_monitor.sh [-i seconds] [-p "pid list"]...
   -i                  interval(second),default value is 30
   -p                   pid list

example: /system/bin/memory_monitor.sh -i 30 -p "233 513 679"
```

  由于打印比较多，建议单独运行该脚本，并且将 log 重定向到 data 分区中，比如:
  Because the log output is verbose, recommend to separately execute this script and redirect the output message to emmc in data partition as below:

```
sh /system/bin/memory_monitor.sh –i 30 –p "233 513 679" >/data/memory_usage.log
```

- hardware_monitor.sh：该脚本用于实时打印当前的 cpu、gpu、ddr 频率、负载以及温度情况。
  hardware_monitor.sh: this script is used to real-time print current frequency, load and temperature of CPU, GPU and DDR.
- gpio.sh：该脚本用于打印当前机器中的 io 状态，并且可以实时设置 gpio 的 iomux，使用方法为：
  gpio.sh: this script is used to print current iomux status of the device, and also can set the iomux of gpio at real-time. The method to use this script is as follows:

```
130|rk3399_all:/ # sh /data/gpio.sh
```

```
Usage: /data/gpio.sh [OPTION]...
    -r gpioxxx: such as '-r gpio0a0' means read all the functions gpio0a0 support and also tell the current function of gpio0a0
    -w gpioxxx -f funcs: such as '-w gpio4d0 -f pcie_clkreqn ' means set the iomux state of gpio4d0 to 'pcie_clkreqn'
    -L : list all the iomux state of gpios
```

6.23 DATA 分区文件系统切换为 EXT4 Change the file system of DATA partition to EXT4

从行业 SDK 的 V2.1 版本开始，data 分区默认从 F2FS 格式切换为 EXT4 格式文件系统，F2FS 相对于以前使用的 EXT4 而言，随机读写的效率会优于 EXT4。但是另外一方面，EXT4 使用的范围更广，经历的时间考验也更久，稳定性这块我们认为可能会优于 F2FS，所以对于行业客户而言，如果更注重文件系统的稳定性，建议使用 EXT4。

Starting from V2.1 of industry SDK, the file system of data partition is changed from F2FS format to EXT4 format by default. Comparing with previously used EXT4, F2FS random read/write efficiency is better than EXT4. But on the other hand, we think EXT4 stability is better than F2FS due to the wider usage and longer existence of EXT4. So for industry customers who focus more on the file system stability, we recommend to use EXT4.

6.23.1 注意事项 Notice

【注意】从 F2FS 文件系统切到 EXT4 需要格式化 data 分区，会导致 data 分区数据全部清空，因此如果机器已经量产，不建议切换文件系统。已量产的客户如果 sdk 更新到了 V2.1 版本，并且原先的文件系统为 F2FS，请回退 device/rockchip/rk3399 目录下的如下提交：

[Note] Switching from F2FS file system to EXT4 requires formatting data partition, which will cause the data partition to be cleaned up, so if the devices already MP, it is not recommended to switch the file system. If MP customers update SDK to V2.1, and the original file system is F2FS, please revert below commit in the directory of device/rockchip/rk3399:

```
commit 025a783298e3b74fa51c252a07678b867b4ba431
Author: Wenping Zhang <wenping.zhang@rock-chips.com>
Date:   Fri Mar 8 15:42:14 2019 +0800

    fstab: change the userdata partition to ext4 format.

    Change-Id: I15f6c33cc4e297cb48e86434ef1e31ff9ef52d5f
    Signed-off-by: Wenping Zhang wenping.zhang@rock-chips.com
```

如果客户能够接受 data 分区擦除，需使用完整固件烧写，不能够使用 ota 升级（因为 ota 升级默认不会重新格式化 data 分区）。或者可以考虑使用 sd 升级卡，将 sd 卡做成升级卡进行机器的升级，如果使用 sd 升级卡，需要作如下配置确保会重新格式化 data 分区：

If customers can accept to erase the data partition, need to use the complete image for flashing, and cannot use ota upgrade (because ota upgrade will not format data partition by default). Or you can consider to use the sd card to upgrade the device. If using the sd card to upgrade, need to configure as below to

ensure that the data partition will be formatted:

```
device/rockchip/common$ git diff
diff --git a/BoardConfig.mk b/BoardConfig.mk
index c7ac4d3..3db7bdc 100755
--- a/BoardConfig.mk
+++ b/BoardConfig.mk
@@ -366,7 +366,7 @@ BOARD_USB_ALLOW_DEFAULT_MTP ?= false
 BOARD_USE_FIX_WALLPAPER ?= false

 # SDBoot: Format data.
-RECOVERY_SDBOOT_FORMATE_DATA ?= false
+RECOVERY_SDBOOT_FORMATE_DATA ?= true

 HIGH_RELIABLE_RECOVERY_OTA := false
 BOARD_USES_FULL_RECOVERY_IMAGE := false
```

6.23.2 Recovery 擦除慢问题 Recovery erase slow issue

从 F2FS 切到 EXT4 以后，我们发现 recovery 擦除 data 分区会很慢，这个是因为 emmc 擦除的方式分为两种：

After switching from F2FS to EXT4, we found that it is very slow to erase data partition during recovery. This is because there are two ways to erase emmc:

- Secure Discard

这个擦除方式为物理擦除，会完全擦除 data 分区的数据，主要是为了确保数据的安全性，保证 data 分区不会被恶意导出。但是这种方式的缺点为耗时长，尤其是数据写满的情况下（比如从 F2FS 文件系统切换为 EXT4 文件系统时）。

This method is the physical erase, which will erase all the data in the data partition, mainly to ensure the data security and the data partition cannot be output maliciously. But the disadvantage of this method is that it will take a long time, especially when the data is full (e.g. when the file system is switched from F2FS to EXT4).

注：有些小厂的 emmc，Secure Discard 方式并未实现完全擦除，而是和 Discard 方式一样只是擦除索引文件。

Note: for some emmc from small vendor, Secure Discard doesn't realize the complete erase, but just erase the index file as Discard.

- Discard

这种擦除方式为逻辑擦除，类似于擦除索引文件，但不会擦除存储上的每个物理位。这种方式的优点是擦除速度快，缺点是在擦除后，由于只擦除了索引文件，data 分区数据理论上有被恶意导出的风险，建议使能 data 分区的加密功能来降低被恶意导出的风险（SDK 加密功能默认为开启的）。

This method is the logic erase, which is similar to ease the index file, but not erase each physical bit of the memory. The advantage of this method is that it is fast to erase, but the disadvantage is that after erase, data of the dada partition theoretically has the risk of being output maliciously because only index files are erased, recommend to enable the encryption function of data partition to lower down the risk of being output maliciously (SDK encryption function is enabled by default).

系统默认使用 Secure Discard 方式，考虑了上面的风险以后，如果客户仍然需要选择格式化更快

速的 Discard 方式，请打下面的补丁：

The system uses Secure Discard by default, after considering above risks, if customer still needs to choose the faster formatting Discard method, please apply below patch:

```
system/extras$ git diff
diff --git a/ext4_utils/wipe.c b/ext4_utils/wipe.c
index 5766632..0700797 100644
--- a/ext4_utils/wipe.c
+++ b/ext4_utils/wipe.c
@@ -44,7 +44,7 @@ int wipe_block_device(int fd, s64 len)

         range[0] = 0;
         range[1] = len;
-        ret = ioctl(fd, BLKSECDISCARD, &range);
+        ret = ioctl(fd, BLKDISCARD, &range);
         if (ret < 0) {
                 range[0] = 0;
                 range[1] = len;
```

6.23.3 F2FS 切换为 EXT4 方法 Method to switch from F2FS to EXT4

如果是 V2.1 以前的版本，data 分区默认还是 F2FS 文件系统，请更新到 V2.1 以后的版本，如果不想所有更新，请按照下述方法切换为 EXT4：

If it is V2.1 or previous version, data partition uses F2FS file system by default, please update to the version after V2.1. If do not want to update all, please switch to EXT4 per below method:

1. 首先确认是否有使用强制加密功能，即 BUILD_WITH_FORCEENCRYPT 这个宏是否有使能，SDK 默认为使能状态，这个宏会决定使用哪个 fstab 文件，详见下述代码：

   First confirm if the force encryption function is used or not, that is BUILD_WITH_FORCEENCRYPT macro is enabled or not. SDK enables it by default, and this macro will decide to use which fstab file referring to below code:

```
device/rockchip/rk3399$ vim device.mk
ifeq ($(BUILD_WITH_FORCEENCRYPT),true)
PRODUCT_COPY_FILES += \

$(LOCAL_PATH)/fstab.rk30board.bootmode.forceencrypt.unknown:root/fstab.rk30board.bootmode.unknown \

$(LOCAL_PATH)/fstab.rk30board.bootmode.forceencrypt.emmc:root/fstab.rk30board.bootmode.emmc \

$(LOCAL_PATH)/fstab.rk30board.bootmode.forceencrypt.nvme:root/fstab.rk30board.bootmode.nvme
else
PRODUCT_COPY_FILES += \
    $(LOCAL_PATH)/fstab.rk30board.bootmode.unknown:root/fstab.rk30board.bootmode.unknown \
    $(LOCAL_PATH)/fstab.rk30board.bootmode.emmc:root/fstab.rk30board.bootmode.emmc \
```

```
        $(LOCAL_PATH)/fstab.rk30board.bootmode.nvme:root/fstab.rk30board.bootmode.nvme
    endif
```

2. 需要根据存储的介质找到对应的 fstab，然后修改对应的 data 分区加载选项，下面以 emmc 为例：
   Need to find the corresponding fstab according to the memory medium, and then modify the corresponding loading option of data partition. Here take emmc as example:

```
    device/rockchip/rk3399$ git diff
    diff --git a/fstab.rk30board.bootmode.forceencrypt.emmc
b/fstab.rk30board.bootmode.forceencrypt.emmc
    index 9adbfb9..833c85e 100755
    --- a/fstab.rk30board.bootmode.forceencrypt.emmc
    +++ b/fstab.rk30board.bootmode.forceencrypt.emmc
    @@ -8,11 +8,11 @@
     #/dev/block/platform/fe330000.sdhci/by-name/system      /system      ext4
ro,noatime,nodiratime,noauto_da_alloc                                wait,check,verify
     /dev/block/platform/fe330000.sdhci/by-name/cache       /cache       ext4
noatime,nodiratime,nosuid,nodev,noauto_da_alloc,discard              wait,check
     /dev/block/platform/fe330000.sdhci/by-name/metadata    /metadata    ext4
noatime,nodiratime,nosuid,nodev,noauto_da_alloc,discard              wait,check
    -/dev/block/platform/fe330000.sdhci/by-name/userdata    /data        f2fs
noatime,nodiratime,nosuid,nodev,discard,inline_xattr
wait,check,notrim,forceencrypt=/metadata/key_file
    +#/dev/block/platform/fe330000.sdhci/by-name/userdata    /data        f2fs
noatime,nodiratime,nosuid,nodev,discard,inline_xattr
wait,check,notrim,forceencrypt=/metadata/key_file
     #data for f2fs nobarrier
     #/dev/block/platform/fe330000.sdhci/by-name/userdata    /data        f2fs
noatime,nodiratime,nosuid,nodev,discard,inline_xattr,nobarrier
wait,check,notrim,forceencrypt=/metadata/key_file
     #data for ext4
    -#/dev/block/platform/fe330000.sdhci/by-name/userdata    /data        ext4
noatime,nodiratime,nosuid,nodev,noauto_da_alloc,discard,errors=panic
wait,check,forceencrypt=/metadata/key_file
    +/dev/block/platform/fe330000.sdhci/by-name/userdata     /data        ext4
noatime,nodiratime,nosuid,nodev,noauto_da_alloc,discard,errors=panic
wait,check,forceencrypt=/metadata/key_file
     /dev/block/platform/fe330000.sdhci/by-name/misc        /misc        emmc
defaults        defaults
     # sdcard
     /devices/platform/fe320000.dwmmc/mmc_host*                auto    auto      defaults
voldmanaged=sdcard1:auto,encryptable=userdata
```

3. 修改 recovery 模式时 data 分区加载选项：
   Modify the loading option of data partition in recovery mode:

```
    device/rockchip/rk3399$ git diff
    diff --git a/recovery.emmc.fstab b/recovery.emmc.fstab
```

```
index d232dbd..ac58660 100755
--- a/recovery.emmc.fstab
+++ b/recovery.emmc.fstab
@@ -5,7 +5,7 @@
 /dev/block/platform/fe330000.sdhci/by-name/system                /system
ext4                defaults                defaults
 /dev/block/platform/fe330000.sdhci/by-name/cache                 /cache
ext4                defaults                defaults
 /dev/block/platform/fe330000.sdhci/by-name/metadata              /metadata           ext4
defaults                defaults
-/dev/block/platform/fe330000.sdhci/by-name/userdata              /data               f2fs
defaults                defaults
+/dev/block/platform/fe330000.sdhci/by-name/userdata              /data               ext4
defaults                defaults
 /dev/block/platform/fe330000.sdhci/by-name/cust                  /cust               ext4
defaults                defaults
 /dev/block/platform/fe330000.sdhci/by-name/custom                /custom
ext4                defaults                defaults
 /dev/block/platform/fe330000.sdhci/by-name/misc
```

6.24 Data 分区使能和禁用加密功能 Enable and disable encryption function of data partition

SDK 默认使能了 data 的加密功能，但是我们发现有些客户会有一些特殊的需求，比如在 android 很早的初始化阶段去访问 data 分区，但是此时 data 分区正处于加密流程，读取会出现失败问题。举例说明：

SDK enables data encryption function by default, but we find that some customers have some special requirements, such as accessing data partition at the very beginning of android initialization stage, but at the moment data partition is in the encryption process, it will fail to read. For example:

假如客户需要配置一个 persist 属性，需要重启后生效，并且这个属性在 surfaceflinger 服务启动的时候要去判断，此时会发现 surfaceflinger 服务读取该属性失败。

If customer needs to configure a persist property, which will be effective after reboot, and this property needs to judge when surfaceflinger service starts, at this moment surfaceflinger service will fail to read this property.

为了解决上述问题，并且不需要 data 分区的加密功能，可以将加密功能去掉，如下面补丁所示：

In order to resolve above issue, and don’t need data partition encryption function, you can remove the encryption function as shown in below patch:

```
device/rockchip/rk3399$ git diff
diff --git a/fstab.rk30board.bootmode.forceencrypt.emmc b/fstab.rk30board.bootmode.forceencrypt.
emmc
index 9adbfb9..c0afafc 100755
--- a/fstab.rk30board.bootmode.forceencrypt.emmc
+++ b/fstab.rk30board.bootmode.forceencrypt.emmc
```

```
@@ -8,7 +8,7 @@
#/dev/block/platform/fe330000.sdhci/by-name/system /system ext4 ro,noatime,nodiratime,noauto_da_alloc wait,check,verify
/dev/block/platform/fe330000.sdhci/by-name/cache /cache ext4 noatime,nodiratime,nosuid,nodev,noauto_da_alloc,discard wait,check
/dev/block/platform/fe330000.sdhci/by-name/metadata /metadata ext4 noatime,nodiratime,nosuid,nodev,noauto_da_alloc,discard wait,check
-/dev/block/platform/fe330000.sdhci/by-name/userdata /data f2fs noatime,nodiratime,nosuid,nodev,discard,inline_xattr wait,check,notrim,forceencrypt=/metadata/key_file
+/dev/block/platform/fe330000.sdhci/by-name/userdata /data f2fs noatime,nodiratime,nosuid,nodev,discard,inline_xattr wait,check,notrim,encryptable=/metadata/key_file
#data for f2fs nobarrier
#/dev/block/platform/fe330000.sdhci/by-name/userdata /data f2fs noatime,nodiratime,nosuid,nodev,discard,inline_xattr,nobarrier wait,check,notrim,forceencrypt=/metadata/key_file
#data for ext4
```

需要将 device/rockchip/rk3399 下面对应 fstab 文件中，将 data 分区的挂载参数从 forceencrypt 改为 encryptable。

Need to change the load parameter of data partition from forceencrypt to encryptable in the corresponding fstab file under device/rockchip/rk3399.

6.25 RK3399 性能优化方法 RK3399 performance optimization method

我们提供了一些方法对 RK3399 的性能进行优化，但是请注意，性能上的优化会引起功耗的增加以及发热量的增大，这块请客户自行考量项目的需求。性能优化的方法详见文档：

We provide some methods to optimize RK3399 performance. But please pay attention to that, performance optimization will cause the increase of power consumption and heating. Please customers consider according to the project requirement. The performance optimization method refers to the document:

```
RKDocs/rk3399/Rockchip_RK3399_Introduction_Performance_Optimization_CN.pdf
```

6.26 RK3399K 芯片支持 RK3399K chipset support

从 sdk 版本 v1.9（通过 ls .repo/manifests/ -l 确认版本）开始支持 RK3399K 芯片，但需要在最终使用的 dts 中手动加入下面的补丁，以 RK3399 SDK 为例：

SDK starts to support RK3399K from version v1.9 (confirm the version by ls .repo/manifests/ -l), but need to manually apply below patch in the final dts. Take RK3399 SDK as example:

```
git diff
diff --git a/arch/arm64/boot/dts/rockchip/rk3399-sapphire-excavator-edp.dts b/arch/arm64/boot/dts/rockchip/rk3399-sapphire-excavator-edp.dts
index 728b430..387b132 100644
```

```
--- a/arch/arm64/boot/dts/rockchip/rk3399-sapphire-excavator-edp.dts
+++ b/arch/arm64/boot/dts/rockchip/rk3399-sapphire-excavator-edp.dts
@@ -45,6 +45,7 @@
 #include "rk3399-excavator-sapphire.dtsi"
 #include "rk3399-android.dtsi"
 #include "rk3399-vop-clk-set.dtsi"
+#include "rk3399k-opp.dtsi"

 / {
        model = "Rockchip RK3399 Excavator Board edp (Android)";
```

6.27 禁用串口打印功能 Disable the serial print function

有些客户需要关闭串口打印功能，另外有些客户可能因为串口不够用需要将打印串口用作普通的串口使用，但是发现在调试过程中，经常出现由于配置不到位导致的系统无法开机问题。从 SDK V2.1 版本开始，加入了禁用串口打印功能的支持。需要注意如下几个配置：

Some customers need to close the serial print function, and some customers may need to use the print serial port as normal serial port because there are not enough serial ports, however, it was found that during debugging process, the system could not bootup due to improper configuration. Starting from SDK V2.1, it supports to disable the serial port print function. Need to pay attention to below configurations:

- 版本升级到 V2.1 以后，如果出现串口打印到一半停掉不再打印（系统未死机）的问题，请确认 dts 中 uart2 是否有 disable，因为新代码中 fiq 和普通串口只能二选一。如下所示：

After update the version to V2.1, if the serial port stops printing in the middle (the system doesn't crash), please confirm if uart2 in dts is diasbled or not, because you can only choose one between fiq and normal serial port in the new code. Shown as below:

```
--- a/arch/arm64/boot/dts/rockchip/rk3399-firefly-android.dts
+++ b/arch/arm64/boot/dts/rockchip/rk3399-firefly-android.dts
@@ -1050,7 +1050,7 @@
 };

 &uart2 {
-       status = "okay";
+       status = "disabled";
 };

 &usbdrd3_0 {
```

- 如果想禁用 uart2 的 fiq 调试功能，把 uart2 当成普通串口用，请按照如下方式修改：

If want to disable fiq debugging function of uart2, use uart2 as normal serial port, please modify per below method:

```
--- a/arch/arm64/boot/dts/rockchip/rk3399-firefly-android.dts
+++ b/arch/arm64/boot/dts/rockchip/rk3399-firefly-android.dts
@@ -1050,7 +1050,12 @@
 };
```

```
 &uart2 {
-           status = "disabled";
+           status = "okay";
+};
+
+&fiq_debugger {
+       rockchip,serial-id = <0xffffffff>;
+       status = "okay";
 };

 &usbdrd3_0 {
```

6.28 DP 音频功能 DP audio function

从行业 SDK V2.1 版本开始，为了解决一些 bug 以及提升 DP 音频的兼容性，DP 音频从 I2S 切到 SPDIF 通道，该修改涉及几个部分，请同步下面几个仓库的修改：

Starting from industry SDK V2.1, in order to resolve some bug and improve the compatibility of DP audio, DP audio is changed from I2S to SPDIF channel. The modification contains several parts. Please sync the modifications of below libs:

```
hardware/rockchip/audio
frameworks/base
kernel
```

【注意】如果使用 lpddr4，由于 lpddr4 的变频要求声卡的数量限制在 2 个以内（具体请参考本文档 5.8 节），而如果使能了 DP 的音频功能，则可能存在 3 个声卡，如下所示：

【Note】If use lpddr4 and need the ddr frequency scaling function, the system only support 2 audio card at most(reference to section 5.8). But if enable dp audio function, may exist 3 audio card, such as:

```
[       2.176483] ALSA device list:
[       2.176511]     #0: realtekrt5651codec_hdmiin
[       2.176520]     #1: rockchip,hdmi
[       2.176528]     #2: rockchip-cdndp-sound
```

所以请按照 5.8.2 节所述，将 lpddr4 的变频关闭。如果客户需要使能 lpddr4 变频，并且不需要 dp 音频，请直接 disable dp_sound 节点，如下所示：

So if you need dp audio function, please shutdown the lpddr4 frequency scaling fuction according to section 5.8.2. If you don’t need dp audio and need lpddr4 frequency scalling function, please disable the node dp_sound, such as:

```
+&dp_sound {
+           status = "okay";
+};
```

经过修改以后 HDMI 音频走 I2S 通道，DP 音频走 spdif 通道。上述修改中内核部分请同步到最新代码，并按照下面的说明确认当前的 dts 配置是否正确（**注意，由于客户参考的 dts 配置不同，请自行在对应的 dts 配置中添加对应的修改**）：

After modification, HDMI audio uses I2S channel, and DP audio uses spdif channel. In above modifications, please sync the kernel part to the latest code, and confirm if current dts configuration is correct or not according to below instruction (note, as the dts configuration referenced by customer is

different, please add the corresponding modification in the corresponding dts configuration by yourself):

- 删除下面的节点

  Delete below node

```
diff --git a/arch/arm64/boot/dts/rockchip/rk3399-android.dtsi
b/arch/arm64/boot/dts/rockchip/rk3399-android.dtsi
index 6028ac8..62a427b 100644
--- a/arch/arm64/boot/dts/rockchip/rk3399-android.dtsi
+++ b/arch/arm64/boot/dts/rockchip/rk3399-android.dtsi
@@ -255,13 +255,6 @@
                        rockchip,android-charge-on = <0>;
                };

-               hdmi_dp_sound: hdmi-dp-sound {
-                       status = "disabled";
-                       compatible = "rockchip,rk3399-hdmi-dp";
-                       rockchip,cpu = <&i2s2>;
-                       rockchip,codec = <&hdmi>, <&cdn_dp>;
-               };
-
```

- 添加 dp_sound 节点

  Add dp_sound node

```
diff --git a/arch/arm64/boot/dts/rockchip/rk3399-sapphire.dtsi
b/arch/arm64/boot/dts/rockchip/rk3399-sapphire.dtsi
index 8a3d7ea..a309c08 100644
--- a/arch/arm64/boot/dts/rockchip/rk3399-sapphire.dtsi
+++ b/arch/arm64/boot/dts/rockchip/rk3399-sapphire.dtsi
@@ -100,6 +100,13 @@
                        #sound-dai-cells = <0>;
                };

+               dp_sound: dp-sound {
+                       status = "disabled";
+                       compatible = "rockchip,cdndp-sound";
+                       rockchip,cpu = <&spdif>;
+                       rockchip,codec = <&cdn_dp 1>;
+               };
+
```

- 禁用以及使能一些节点

  Disable and enable some nodes

```
+&spdif {
+       status = "okay";
+};
+
+&dp_sound {
```

```
+		status = "okay";
+};
+
+/*
+ * if enable dp_sound, should disable spdif_sound and spdif_out
+ */
+&spdif_out {
+		status = "disabled";
+};
+
+&spdif_sound {
+		status = "disabled";
+};
+
+&hdmi_sound {
+		status = "okay";
+};
+
```

6.29 MIPI 转 LVDS 芯片 TC358775 支持 MIPI to LVDS chipset TC358775 support

从版本 v2.2 版本开始，加入了 TC358775 转接芯片支持，该芯片支持需要另外打补丁，补丁请参考 RKDocs/rk3399/patches/ tochiba_tc358775_19.08.19.rar

Starting from v2.2, it adds TC358775 conversion chipset support, and this chipset support requires to apply additional patch. Please refer to RKDocs/rk3399/patches/ tochiba_tc358775_19.08.19.rar for the patch.

6.30 V-BY-ONE 支持 V-BY-ONE support

SDK 默认可以通过 hdmi 转换芯片 EP9169 对 V-BY-ONE 进行支持，默认的代码只支持单分区刷新，从版本 v2.2 开始，加入双分区刷新功能，需要手动打上如下补丁：

SDK can support V-BY-ONE through hdmi conversion chipset EP9169 by default. The default code only supports single partition refresh. From v2.2, it adds dual partition refresh function which requires to manually apply below patch:

```
frameworks/native$ git diff
diff --git a/services/surfaceflinger/Android.mk b/services/surfaceflinger/Android.mk
index e88a863..5a15160 100755
--- a/services/surfaceflinger/Android.mk
+++ b/services/surfaceflinger/Android.mk
@@ -1,5 +1,5 @@
```

```
 LOCAL_PATH := $(call my-dir)
-VBYONE_PROCESS_ENABLE := false
+VBYONE_PROCESS_ENABLE := true
 include $(CLEAR_VARS)

 LOCAL_CLANG := true
```

6.31 优化 DDR 内存空间 Optimize the usage of ddr memory space

如果对于 ddr 内存空间有优化要求，请参考文档 RKDocs/rk3399/ Rockchip_RK3399_Introduction_Android7.1_SDK_Memory_Optimization_CN&EN.pdf。其中，文档中提到的 android 32bit 的补丁位于 RKDocs/rk3399/patches/ rk3399_7.1_32bit_patch.rar。

If you want to optimize the ddr memory usage, please refer to RKDocs/rk3399/ Rockchip_RK 3399_Introduction_Android7.1_SDK_Memory_Optimization_CN&EN.pdf. In addition, the android 32 bit patch mentioned in the document, can be found in path RKDocs/rk3399/patches/ rk3399_7.1_32 bit_patch.rar.

7 系统调试 System debug

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。

This chapter mainly introduces the debugging tools and methods used in SDK development an d will update and improve continually to help developers familiar with the basic system debugging quickly and analyze the issues correctly.

7.1 ADB 工具 ADB tool

7.1.1 概述 Overview

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：

ADB (Android Debug Bridge) is a tool in Android SDK which can be used to operate and manage Android simulator or the real Android device. The functions mainly include:

- 运行设备的 shell（命令行）
  Run the device shell (command line)
- 管理模拟器或设备的端口映射
  Manage the port mapping of the simulator or the device
- 计算机和设备之间上传/下载文件
  Upload/download files between the computer and the device
- 将本地 apk 软件安装至模拟器或 Android 设备
  Install the local apk to simulator or Android device

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 连接设备的方式不同，ADB 可以分为两类：

ADB is a “client – server” program. Usually the client is PC and the server is the actual Android device or simulator. The ADB can be divided into two categories according to the way PC connects to the device:

- 网络 ADB：主机通过有线/无线网络（同一局域网）连接到 STB 设备
  Network ADB: PC connects to STB device through cable/wireless network
- USB ADB：主机通过 USB 线连接到 STB 设备
  USB ADB: PC connects to STB device through USB cable

7.1.2 USB ADB 使用说明 USB ADB usage

USB ADB 使用有以下限制：

USB ADB usage has below limitations:

- 只支持 USB OTG 口
  Only support USB OTG port
- 不支持多个客户端同时使用（如 cmd 窗口，eclipse 等）
  Not support multiple clients at the same time (such as cmd window, eclipse etc.)
- 只支持主机连接一个设备，不支持连接多个设备
  Support host connects to only one device but multiple devices

连接步骤如下：

The connection steps are as below:

1、设备已经运行 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。

The device already running Android system, setting -> developer option -> connect to the computer, enable usb debugging switch.

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设备相连。

```
adb shell
```

PC connects to the device USB OTG port only through USB cable, and then the computer connects with the device through below command:

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。

Execute the command ”adb devices” to see if the connection is successful or not. If the device serial number shows up, the connection is successful.

7.1.3 网络 ADB 使用要求 Network ADB use requirement

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。

ADB early versions only support device debugging through USB, and the function of debugging Android devices through tcp/ip is added from adb v1.0.25.

如果你需要使用网络 ADB 来调试设备，必须要满足如下条件：

If you need to use network ADB to debug the device, must meet below conditions:

1、设备上面首先要有网口，或者通过 WiFi 连接网络。

The device must have network port, or connect the network through Wi-Fi.

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。

The device and PC are already in the local network and the device has IP address.

3、要确保研发机和设备能够相互 ping 得通。

Need to confirm the device and PC can ping each other.

4、研发机已经安装了 ADB。

PC already installs 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。

Confirm Android device adbd process (ADB backaground process) is already run. adbd process will monitor port 5555 to do ADB connection debugging.

7.1.4 SDK 网络 ADB 端口配置 SDK network ADB port configuration

SDK 默认未对网络 ADB 端口进行配置，需要手动修改打开配置。

SDK doesn't configure network ADB port by default. Need to manually modify to open the configuration.

修改 device/rockchip/rkxxxx/system.prop 文件，添加如下配置：

Modify device/rockchip/rkxxxx/device.mk file, and add below configuration:

```
service.adb.tcp.port=5555
```

7.1.5 网络 ADB 使用 Network ADB usage

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。

This chapter assumes the device IP is 192.168.1.5. This IP will be used for ADB connection and device debugging in the following context.

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)。

Firstly the Android device should boot up, if possible, confirm adbd is started (use ps command to check).

2、在 PC 机的 cmd 中，输入：

In PC cmd, input:

```
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。

If successful, it will prompt relative hints, if fail, you can execute kill-server command and then retry connection.

```
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 连接设备上面。和 USB 调试是一样的。

After connected, you can input ADB relative commands to debug in PC, such as adb shell, it will connect the device through TCP/IP which is the same as USB debugging.

4、调试完成之后，在研发机上面输入如下的命令断开连接：

After debugging, input below command to disconnect the connection in PC:

```
adb disconnect 192.168.1.5:5555
```

7.1.6 手动修改网络 ADB 端口号 Manually modify the network ADB port number

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：

If SDK doesn't add ADB port number configuration, or want to change ADB port number, you can

change through below method:

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。

Firstly also connect the device normally through USB, execute adb shell in windows cmd to enter.

2、设置 ADB 监听端口：

Set ADB monitor port:

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

3、通过 ps 命令查找 adbd 的 pid

Look up adbd pid using ps command.

4、重启 adbd

Reset 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。

After killing adbd, adbd will automatically restart after Android init process. After adbd restart, if service.adb.tcp.port is set, it will automatically change to monitor network request.

7.1.7 ADB 常用命令详解 ADB commonly used command elaberation

（1）查看设备情况

Check the device situation

查看连接到计算机的 Android 设备或者模拟器：

Check the Android device or simulator connected to computer:

```
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。

The return result is the serial number or IP and port number, status of the Android device connected to PC.

（2）安装 APK

Install APK

将指定的 APK 文件安装到设备上：

Install the specific APK file to the device:

```
adb install <apk 文件路径>
```

示例如下：

For example:

```
adb install “F:\WishTV\WishTV.apk”
```

重新安装应用：

Re-install application:

```
adb install –r <apk 文件路径>
```

示例如下：

For example:

```
adb install –r “F:\WishTV\WishTV.apk”
```

（3）卸载 APK

Uninstall APK

完全卸载：

Complete uninstall:

```
adb uninstall <package>
```

示例如下：

For example:

```
adb uninstall com.wishtv
```

（4）使用 rm 移除 APK 文件：

Use rm to remove APK file:

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

示例如下：

For example:

```
adb shell
rm “system/app/WishTV.apk”
```

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

Note: remove WishTV.apk file in the directory of system/app.

（5）进入设备和模拟器的 shell

Enter shell of the device and simulator

进入设备或模拟器的 shell 环境：

Enthe the shell environment of the device or simulator:

```
adb shell
```

（6）从电脑上传文件到设备

Upload file to the device from computer

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。

Use push command can upload any file or folder from computer to the device. Generally local path means the computer and remote path means the single board device connected with ADB.

```
adb push <本地路径 local path><远程路径 remote path>
```

示例如下：

For example:

```
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。

Note: upload local WishTV.apk file to the system/app directory of the Android system.

（7）从设备下载文件到电脑

Download file from the device to computer

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。

Use pull command can download file or folder from the device to local computer.

```
adb pull <远程路径 remote path><本地路径 local path>
```

示例如下：

For example:

```
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。

Note: download the file or folder from the system/app directory of Android system to local F:\ directory.

（8）查看 bug 报告

Check bug report

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 dumpsys、dumpstate 与 logcat 信息都显示出来。

Run adb bugreport command can check all the error message report generated by system. The command will show all dumpsys, dumpstate and logcat information of the Android system.

（9）查看设备的系统信息

Check the device system information

在 adb shell 下查看设备系统信息的具体命令。

The specific commands to check the device system information in adb shell.

```
adb shell getprop
```

7.2 Logcat 工具 Logcat tool

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。

Android logcat system provides the function to record and check the system debugging information. The logcats are all recorded from various softwares and some system buffer. The buffer can be checked and used through Logcat. Logcat is the most commonly used function for debugging program. The function shows the program running status mainly by printing logcat. Because the amount of logcat is very large, need to do filtering and other operations.

7.2.1 Logcat 命令使用 Logcat command usage

用 logcat 命令来查看系统日志缓冲区的内容：

Use logcat command to check the contents of the system logcat buffer:

基本格式：

The basic format:

```
[adb] logcat [<option>] [<filter-spec>]
```

示例如下：

For example:

```
adb shell
logcat
```

7.2.2 常用的日志过滤方式 The commonly used logcat filter method

控制日志输出的几种方式：

Several ways to control the logcat output:

- 控制日志输出优先级。

  Control the logcat output priority.

示例如下：

For example:

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

示例说明：显示优先级为 warning 或更高的日志信息。

Note: show the logcat information with priority of warning or higher.

- 控制日志标签和输出优先级。
  Control the logcat label and output priority.

示例如下：
For example:

```
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。

Note: support all the logcat information except those with label of ActivityManager and priority of Info above, label of MyApp and priority of Debug above.

- 只输出特定标签的日志
  Only output the logcat with the specific label

示例如下：
For example:

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

或者
or

```
adb shell
logcat –s WishTV
```

示例说明：只输出标签为 WishTV 的日志。
Note: only output the logcat with label of WishTV.

- 只输出指定优先级和标签的日志
  Only output the logcat with the specific priority and label

示例如下：
For example:

```
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。
Note: only output the logcat with priority of I and label of WishTV.

7.2.3 查看上次 log View last 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：

Add –L parameter can print out the logcat information before last system reset. If the stress test and power down abnormal occur, the command can be used to print out the logcat of last Android running status. The command is as below:

```
adb shell
logcat -L
```

7.3 Procrank 工具 Procrank tool

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。

Procrank is a debugging tool with Android, running in the shell environment of the device, used to output the memory snapshot of the process in order to effectively observe the memory usage status of the process.

包括如下内存信息：

Include below memory information:

- VSS：Virtual Set Size 虚拟耗用内存大小（包含共享库占用的内存）The memory size used by virtual (including the memory used by the shared lib)
- RSS：Resident Set Size 实际使用物理内存大小（包含共享库占用的内存）The actually used physical memory size (including the memory used by the shared lib)
- 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）The actually used physical memory size (allocate the memory used by the shared lib in proportion)
- 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）The physical memory used exclusively by the process (not including the memory used by the shared lib)

注意：Note:

- 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；
  USS size represents the memory size only used by the process, and it will be recovered completely after the process is killed.
- 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；
  VSS/RSS includes the memory used by the shared lib, so it is not helpful to check the memory status of the single process.
- 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。
  PSS is the shared memory status used by the specific single process after the shared memory is allocated in proportion.

7.3.1 使用 procrank Use procrank

执行 procrank 前需要先让终端获取到 root 权限

Make sure the terminal has the root authority before executing procrank

```
su
```

命令格式：

The command format:

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

常用指令说明 ：

The commonly used command instructions:

- -v：按照 VSS 排序 order by VSS
- -r：按照 RSS 排序 order by RSS
- -p：按照 PSS 排序 order by PSS
- -u：按照 USS 排序 order by USS
- -R：转换为递增[递减]方式排序 convert to order by increasing[decreasing] method
- -w：只显示 working set 的统计计数 only display the statistical count of working set
- -W：重置 working set 的统计计数 reset the statistical count of working set
- -h：帮助 help

示例：For example:

–输出内存快照：Output the memory snapshot:

```
procrank
```

–按照 VSS 降序排列输出内存快照：Output the memory snapshot in VSS decreasing order:

```
procrank –v
```

默认 procrank 输出是通过 PSS 排序。Procrank is output in PSS order by default.

7.3.2 检索指定内容信息 Search the specific content information

查看指定进程的内存占用状态，命令格式如下：

Use below command format to view the memory status of the specific process:

```
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。

cmdline means the target application name, PID means the target application process.

输出 systemUI 进程的内存占用状态:

Output the memory status used by systemUI process:

```
procrank | grep “com.android.systemui”
```

或者: Or:

```
procrank    | grep 3396
```

7.3.3 跟踪进程内存状态 Trace the process memory status

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。

Analyze if there is memory leakage in the process by tracing the memory usage status. Use the script to continuously output the process memory snapshot, and compare with USS segment to see if there is memory leakage in this process.

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：

For example: output the application memory usage of the process named com.android.systemui to see if there is leakage:

1、编写脚本 test.sh

Write the script 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。

After connect to the device by ADB tool, run the script ./test.sh as shown in below picture:

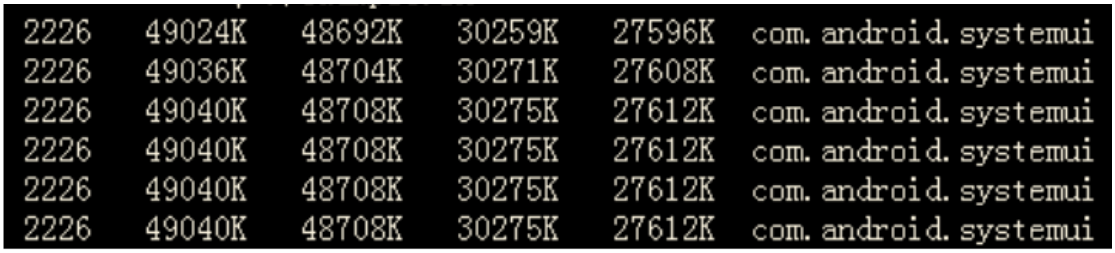


图 7-1 跟踪进程内存状态

Picture 7-1 Trace the process memory status

7.4 Dumpsys 工具 Dumpsys tool

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 tool is a debugging tool in Android system, running in the shell environment of the device, and provides the status information of the running service in the system. The running service means the service process in the Android binder mechanism.

dumpsys 输出打印的条件：

The conditions for dumpsys to output the print:

1、只能打印已经加载到 ServiceManager 中的服务；

Only print the services already loaded to ServiceManager.

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。

If the dump function in the service code is not implemented, there will be no information output.

7.4.1 使用 Dumpsys Use Dumpsys

- 查看 Dumpsys 帮助

  View Dumpsys help

作用：输出 dumpsys 帮助信息。

Function: output dympsys help information.

```
dumpsys   -help
```

- 查看 Dumpsys 包含服务列表

  View the service list of Dumpsys

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。

Function: output all the printable service information of dumpsys, developer can pay attention to the service names required for debugging.

```
dumpsys   -l
```

- 输出指定服务的信息

  Output the specific service information

作用：输出指定的服务的 dump 信息。

Function: output the specific service dump information.

格式：dumpsys [servicename]

Format: dumpsys [servicename]

示例：输出服务 SurfaceFlinger 的信息，可执行命令：

For example: execute below command can output the service information of SurfaceFlinger

```
dumpsys SurfaceFlinger
```

- 输出指定服务和应有进程的信息

  Output the specific service and application process information

作用：输出指定服务指定应用进程信息。

Function: output the specific service and application process information

格式：dumpsys [servicename] [应用名]

Format: dumpsys [servicename] [application name]

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：

For example: execute below command to output the memory information for the service named meminfo and process named com.android.systemui.

```
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。

Note: the service name is case sensitive and must input the full service name.

7.5 串口调试 Serial port debugging

7.5.1 串口配置 Serial port configuration

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 RK3399 波特率设置为 1500000。RTS/CTS 不要勾选，否则串口无法输入。

The serial input and output is the most convenient during debugging. Need to note that RK3399 baud rate is set as 1500000. No need to choose RTS/CTS, otherwise the serial port cannot be input.

7.5.2 FIQ 模式 FIQ mode

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。

FIQ (Fast interrupt request) in ARM is a kind of privilege modes and also one of the abnormal modes.

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。

In RK platforms, input fiq through serial port can enter this mode. At this moment the usage help will pop out and you can do some debugging according to the situation. Usually it is helpful when crash or system die happens.

7.6 音频 codec 问题调试工具及文档 Audio codec issue debugging tool and document

请参考 RKDocs\common\driver\ Rockchip_Developer_Guide_Linux4.4_Audio_CN.pdf。

Please refer to RKDocs\common\driver\ Rockchip_Developer_Guide_Linux4.4_Audio_CN.pdf.

7.7 Last log 开启 Last log enable

在 dts 文件里面添加下面两个节点

Add below two nodes in dts file:

```
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>;
};
```

- 130|root@rk3399:/sys/fs/pstore # ls
  dmesg-ramoops-0　上次内核 panic 后保存的 log。
  dmesg-ramoops-0　Log saved after last kernel panic
  pmsg-ramoops-0　上次用户空间的 log，android 的 log。
  pmsg-ramoops-0　Log of last user space, android log
  ftrace-ramoops-0　打印某个时间段内的 function trace。
  ftrace-ramoops-0　Print function trace during some period.
  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。
  console-ramoops-0　The kernel log for the last boot of last_log, but only save the log with higher priority than default log level
- 使用方法：Usage method:
  cat dmesg-ramoops-0
  cat console-ramoops-0
  logcat -L （pmsg-ramoops-0） 通过 logcat 取出来并解析 pull out by logcat and parse
  cat ftrace-ramoops-0

7.8 Log 自动保存系统 Log automatic saving system

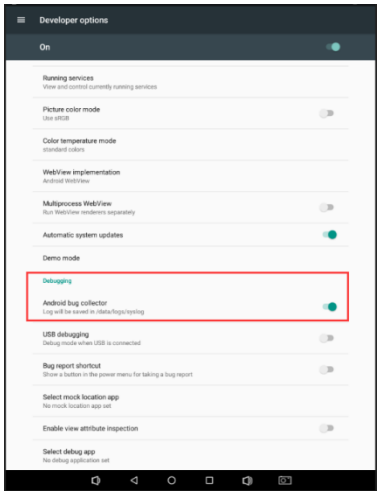
为了简化客户的操作，让没有 android 开发基础的人抓取出问题时的 log，我们新增一个 log 自动保存功能。

In order to simplify customer operation, and allow developers without android developing experience to capture log when issue occurs, we add a function of log automatic saving.

7.8.1 使用方法 Usage

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

Enable this function by clicking Android bug collector in developer option:



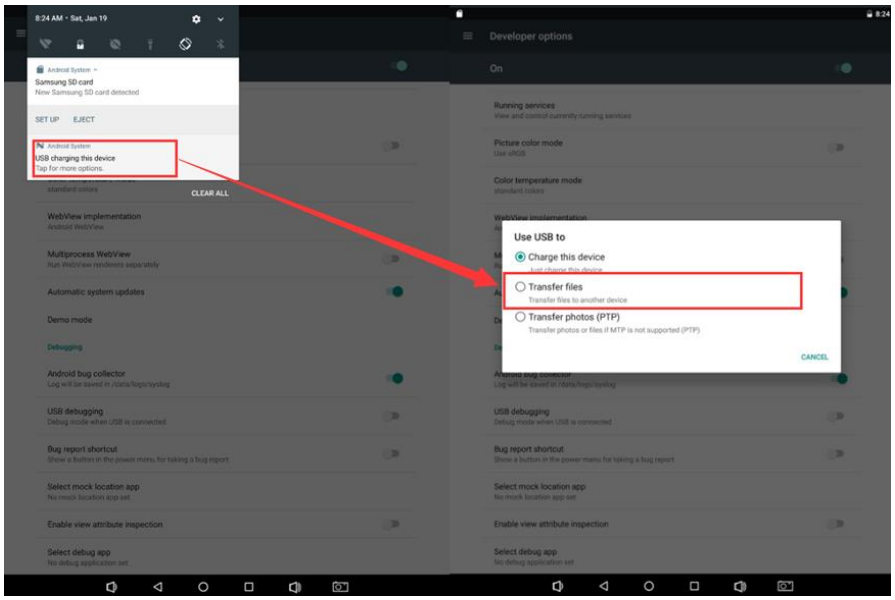
(2) 导出 flash 中的日志到 PC 的使用方法

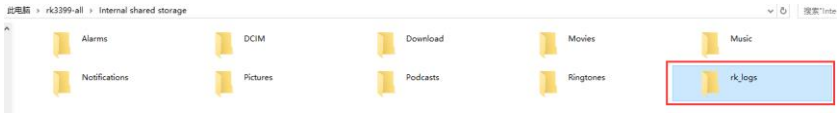
The method to output log in flash to PC

重新开机后连接电脑，可以看到/data/media/0 目录下有 rk_logs 文件夹，其中如果有 COPY-COMPLETE 文件表示拷贝完成，则可以直接通过 adb pull /data/media/0/rk_logs 导出日志，或者打开 usb 传输模式，直接拷贝到电脑上，见下图。

Connect to PC after reboot, you can see rk_logs folder in the directory of /data/media/0, and if there is COPY-COMPLETE it means the copy is completed. Then you can directly output the log through adb pull /data/media/0/rk_logs, or open usb transmission mode, directly copy to PC as shown below.

```
rk3399_all:/data/media/0/rk_logs # ls -al
total 9562
drwxrwxrwx 6 root root 3488 2013-01-20 09:19 .
drwxrwx--- 12 media_rw media_rw 3488 2013-01-20 09:19 ..
-rw-rw-rw- 1 root root 0 2013-01-20 09:19 COPY-COMPLETE
drwxrwxr-x 2 root root 3488 2013-01-20 09:19 anr
-rw-rw-rw- 1 root root 4867234 2013-01-20 09:19 bugreport.log
drwxrwx--- 16 root root 3488 2013-01-20 09:19 logs
drwxr-xr-x 2 root root 3488 2013-01-20 09:19 pstore
drwxr-xr-x 2 root root 3488 2013-01-20 09:19 tombstones
```





(3) 导出 Log 到 SD 卡的使用方法

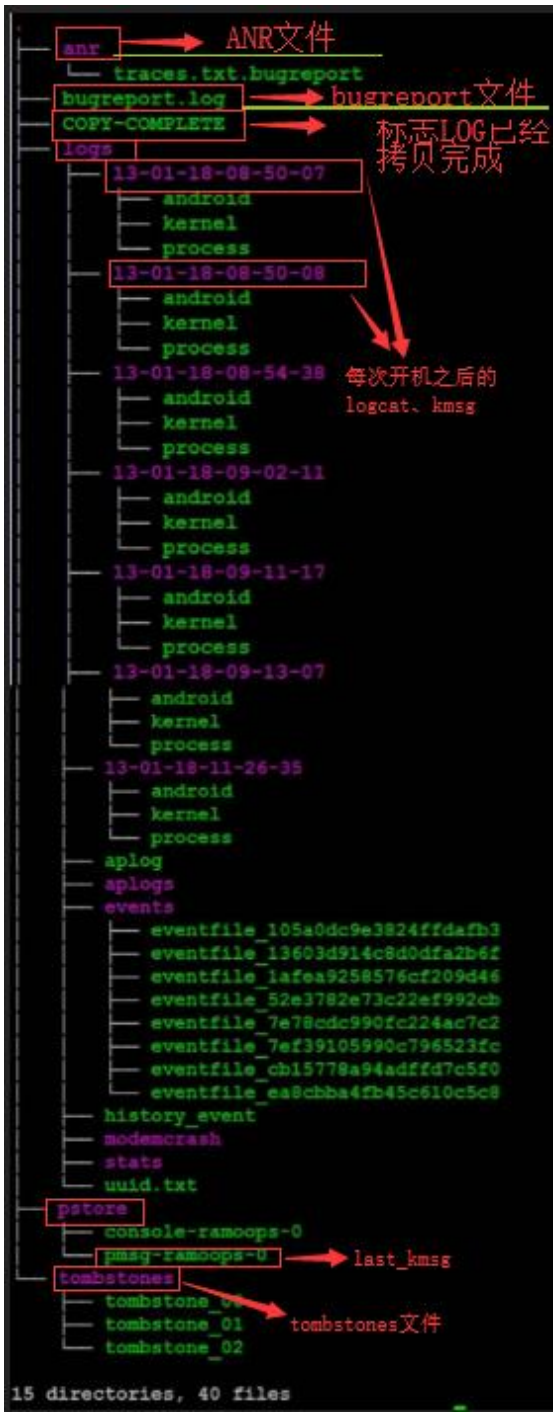
The method to output Log to SD card

sd 卡插入后，几秒后(生成 bugreport 需要一些时间)，后可以看到/storage/8527-18E3/ 目录(该目录每个产品可能各有不同)下有 rk_logs 目录有以下内容，并且有 COPY-COMPLETE 表示拷贝完成。

After sd card is inserted for several seconds (need some time to generate bugreport), you can see below contents in rk_logs directory under the directory of /storage/8527-18E3/ (this directory may be different for each product), and if there is COPY-COMPLETE it means the copy is completed.

```
rk3399_all:/storage/8527-18E3/rk_logs # ls -al
total 14744
drwxrwx--x 6 root sdcard_rw 4096 2013-01-20 09:27 .
drwxrwx--x 14 root sdcard_rw 4096 2013-01-20 09:27 ..
-rwxrwx--x 1 root sdcard_rw 0 2013-01-20 09:27 COPY-COMPLETE
drwxrwx--x 2 root sdcard_rw 4096 2013-01-20 09:27 anr
-rwxrwx--x 1 root sdcard_rw 7522502 2013-01-20 09:27 bugreport.log
drwxrwx--x 16 root sdcard_rw 4096 2013-01-20 09:27 logs
drwxrwx--x 2 root sdcard_rw 4096 2013-01-20 09:27 pstore
drwxrwx--x 2 root sdcard_rw 4096 2013-01-20 09:27 tombstones
```

7.8.2 对 log 的说明 log instruction



8 常用工具说明 Commonly used tool instruction

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\ RK Tools manuals 目录下工具文档。

This chapter simply describes some developing and MP tools usage along with SDK to help the developers familiar with RK platform tool usage. The detailed tool usage refers to the tool related documents in the directory of RKTools and RKDocs\ common\ RKTools manuals.

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。

Use the Stresstest tool to do the stress test for the vasious functions on the target devices to make sure the whole system running stably. SDK can start StressTest application and conduct stress test of various functions by opening the calculator and entering “83991906=” code.

Stresstest 测试工具测试的内容主要包括：

The test items of Stresstest tool mainly include:

模块相关

Module related

- Camera 压力测试：包括 Camera 打开关闭，Camera 拍照以及 Camera 切换。
  Camera stress test: including Camera on/off, Camera taking photo and Camera switch.
- Bluetooth 压力测试：包括 Bluetooth 打开关闭。
  Bluetooth stress test: including Bluetooth on/off.
- WiFi 压力测试：包括 WiFi 打开关闭，（ping 测试以及 iperf 测试待加入）。
  WiFi stress test: including WiFi on/off, (plan to add ping test and iperf test).

非模块相关

Non module related

- 飞行模式开关测试 fly mode on/off test
- 休眠唤醒拷机测试 sleep and resume stress test
- 视频拷机测试 video playing stress test
- 重启拷机测试 restart stress test
- 恢复出厂设置拷机测试 recovery stress test
- ARM 变频测试 ARM frequency scaling test
- GPU 变频测试 GPU frequency scaling test
- DDR 变频测试 DDR frequency scaling test

8.2 PCBA 测试工具 PCBA test tool

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。

PCBA test tool is used to help quickly identify good and bad product features during production to improve the production efficiency. Current test items include panel (LCD), wireless (Wi-Fi), Bluetooth, DDR/eMMC memory, SD card, USB HOST, key, speaker earphone (Codec).

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。

These test items include automatic test item and manual test item. Wireless network, DDR/eMMC, Ethernet are automatic test items, while key, SD card, USB Host, Codec are manual test items.

具体 PCBA 功能配置及使用说明，请参考：

For detailed PCBA function configuration and usage, pleaes refer to:

\RKDocs\common\RKTools manuals\ Rockchip_Developer_Guide_PCBA_Test_Tool_CN&EN.pdf

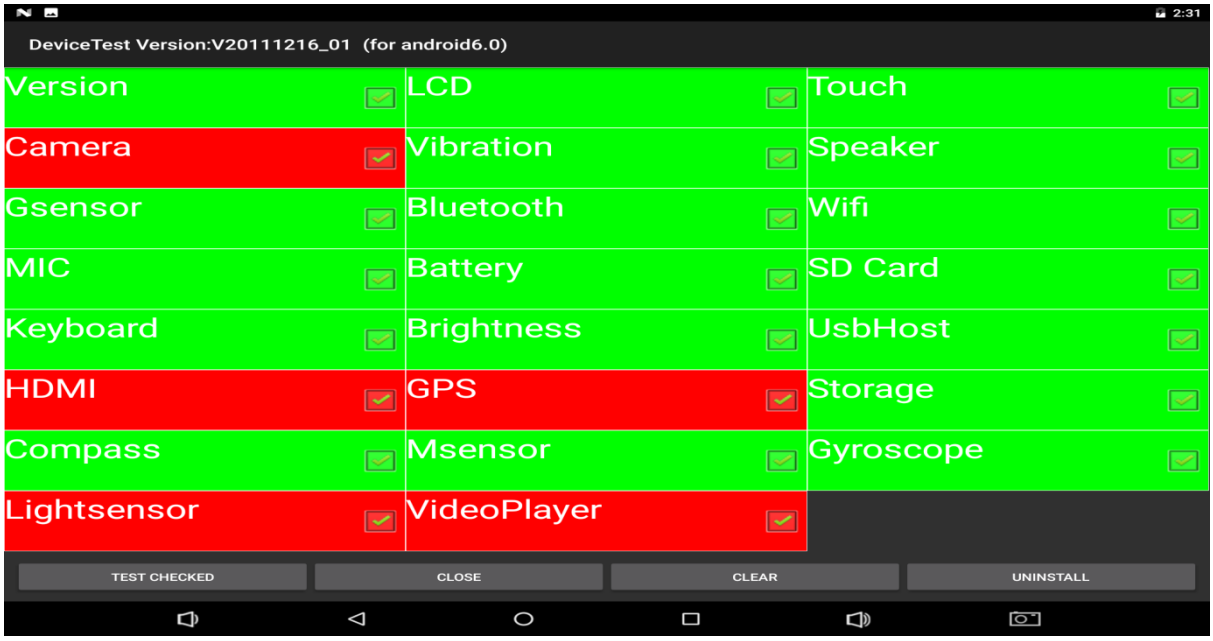
8.3 DeviceTest

DeviceTest 用于工厂整机测试，主要测试装成整机以后外围器件是否正常。SDK 通过打开计算器，输入暗码“000.=”进入 DeviceTest，如下所示：

DeviceTest is used for device testing in factory, mainly to test if the peripheral components of the device are normal or not. SDK can enter DeviceTest by opening the calculator and entering “000.=” code as shown below:

在产线可以根据这个界面进行对应外设的测试，测试时点击“TEST CHECKED”对所测项目逐项进行测试，测试如果成功点击 pass，失败点击 failed，最终结果会显示在界面上，如下图所示，红色为 failed 项，其余为通过项，工厂可根据测试结果进行相应的维修。另外，如果客户需要对该工具进行定制，请联系 FAE 窗口申请对应的源码。

In factory, you can select to do the corresponding test according to this interface. Click “TEST CHECKED” to test the items one by one, click pass if successful, otherwise click failed, the final results will display on the interface as shown below, failed items are marked with red, the others are passed, and the corresponding repaire can be arranged in factory based on the test result. Besides, if customers need to do the tool customization, please apply the corresponding source code by contacting with FAE.



8.4 DDR 测试工具 DDR test tool

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。DDR 稳定性测试工具请参考 6.22 节，另外我们还提供产线测试工具，具体请联系 FAE 获取。

Use DDR test tool to do the stability test on the target devices to make sure DDR function normal and stable. Please refer to section 6.22 for DDR stability test tool. Besides, we also provide factory test tool. Please acquire from FAE contact.

8.5 Android 开发工具 Android development tool

8.5.1 下载镜像 Download the mirror image

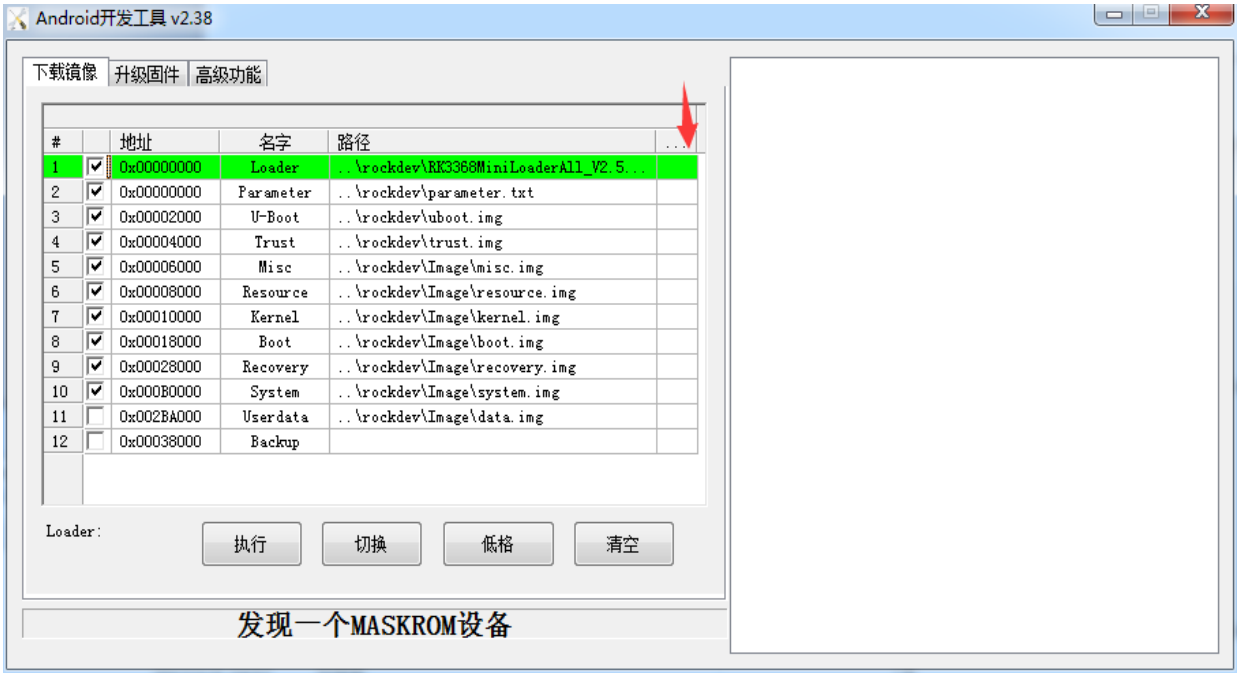


图 8-1 Android 开发工具下载镜像

Picture 8-1 Use Android development tool to download the mirror image

1）连接开发板进入下载模式。

Connect the development board to enter the download mode.

下载模式：先按住开发板 reset 按键，再长按 recovery 按键约 3-4s 时间进入。

Download mode: Firstly press reset key of the development board, and then long press recovery key around 3-4s to enter.

2）打开工具，点击“下载镜像”菜单。单击每一行末尾红色箭头所指处，会弹出文件选择框。选择对应分区的 img 文件路径。

Open the tool, and click “download mirror image” menu. Single click every line end as marked with red arrow, it will pop out file selection box and then choose the img file path of the corresponding partition.

3）依次设置所有 img 文件的路径。

Set all the img file paths successively.

4）配置完成后，点击“执行”。右侧信息框将显示相关信息。

After configuration, click “execute”. The right information box will display the relative information.

5）按钮说明

Button description

“低格”按钮：用于擦除设备

“低格” button: Used to erase the device

“清空”按钮：清空信息框

“清空” button: Used to clean up the information box

8.5.2 升级固件 Upgrade image

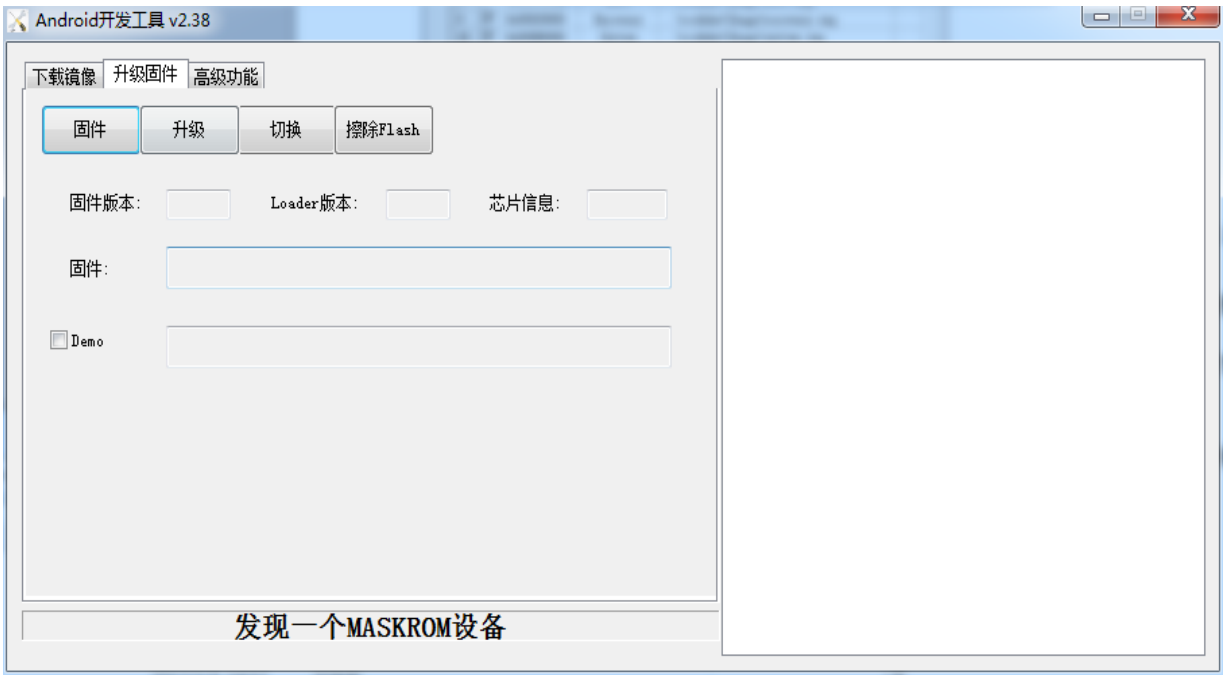


图 8-2 Android 开发工具升级固件

Picture 8-2 Use Android development tool to upgrade image

1) 准备目标固件。（可参考 update.img 打包）

Prepare the target image (refer to update.img package).

2) 确认设备已经进入下载模式。

Confirm the device is already in the download mode.

下载模式进入方法：先按住开发板 reset 按键，再长按 recovery 按键约 3-4s 时间进入。

The way to enter the download mode: Firstly press reset key of the development board, and then long press recovery key around 3-4s to enter.

3) 点击“固件”按钮，选择目标固件 update.img 文件。

Click “image” button, and choose the target image file update.img.

4) 点击“升级”按钮进行下载。右侧信息框将显示相关信息。

Click “upgrage” button to download. The right information box will display the relative information.

8.5.3 高级功能 Senior functions

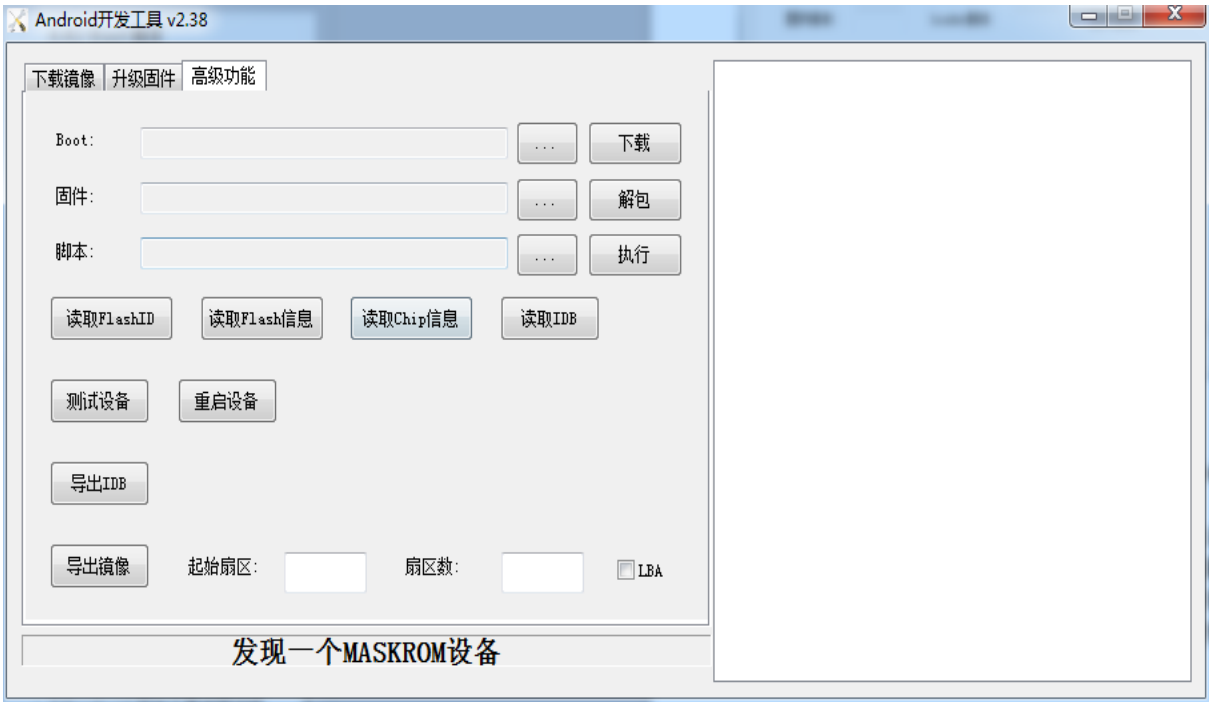


图 8-3 Android 开发工具高级功能

Picture 8-3 Android development tool senior functions

高级功能说明：

Senior functions description:

1) Boot 只能选择打包好的 update.img 文件或是 loader 文件。

   Boot can only select the packed update.img file or loader file.

2) 固件必须现使用打包后的 update.img。

   Image must use the packed update.img.

3) 解包功能可将 update.img 拆解为各部分镜像文件。

   The unpack function can unpack update.img into partial mirror files.

8.6 update.img 打包 update.img pack

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

This platform supports to pack the scattered mirror files into one complete update.img to benefit production flashing and upgrading. The detailed packing steps are as below:

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

   Open the rockdev directory under AndroidTool directory. Compile package-file.

2) 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。

Configure according to package-file, there are some img mirror put under the directory of Image in package-file. If the directory doesn't exist, you need to manually create the Image directory and put the needed mirror in the directory. Note that the mirror name must be correct during configuration and bootloader option should change the loader name according to the generated name yourself.

3) 编辑 mkupdate.bat。

Compile mkupdate.bat.

4) 修改 loader 名称为实际存放的 loader 名称。

Change loader name to be the one actually saved.

5) 点击 mkupdate.bat 运行，结束后会在该目录生成一个 update.img。

Click mkupdate.bat to run, and it will generate one update.img in the directory finally.

8.7 固件签名工具 Image signature tool

参考 RKTools\windows\SecureBootTool_v1.83_foruser.rar 中的《Rockchip Secure Boot Application Note》

Refer to 《Rockchip Secure Boot Application Note》 in RKTools\windows\SecureBootTool_v1.83_foruser.rar.

8.8 序列号/Mac/厂商信息烧写-WNpctool 工具 SN/Mac/Vendor information flashing-WNpctool tool

本平台使用 WNpctool 工具进行序列号/Mac/厂商信息的烧写。以下说明该工具的基本用法。

This platform uses WNpctool tool to flash SN/Mac/vendor information. The basic usage of the tool is described as below.

8.8.1 使用 WNpctool 写入 Use WNpctool to write

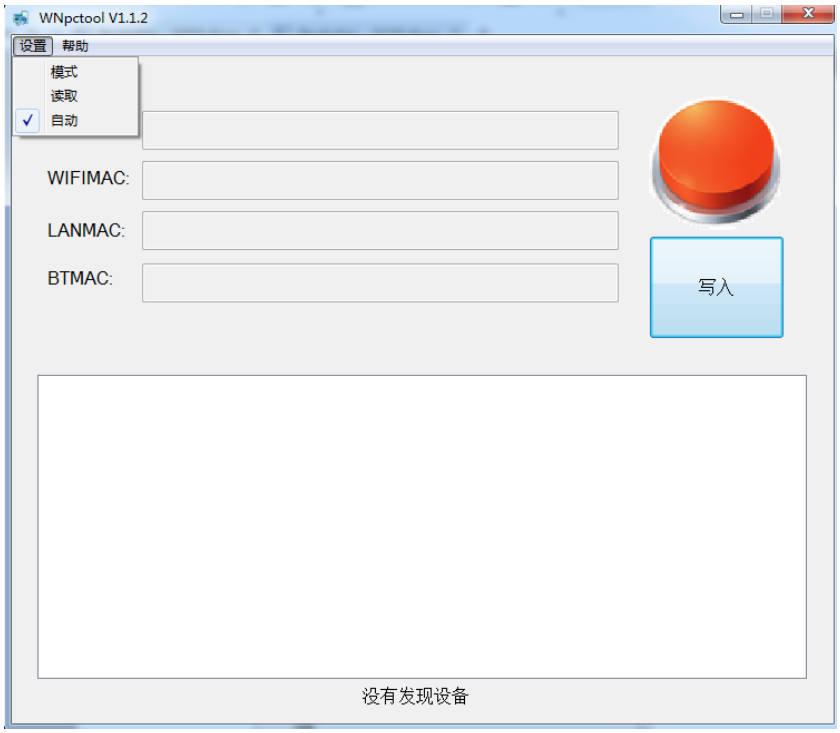


图 8-4WNpctool 工具

Picture 8-4 WNpctool tool

- 进入 loader 模式。

  Enter loader mode.

- 点击“设置”菜单，下拉框中取消勾选“读取”。

  Click “setting” menu, and deselect “Read” in the drop-down box.

（勾选“读取”进行读取，未勾选“读取”则切换到写入功能）

(Select “Read” means to read, and deselect “Read” will convert to write function)

- 点击“设置”菜单，点击“模式”，弹出“模式“窗口，用来设置 SN/WIFI/LAN/BT

  Click “setting” menu, click “mode”, pop out “mode” window to set SN/WIFI/LAN/BT

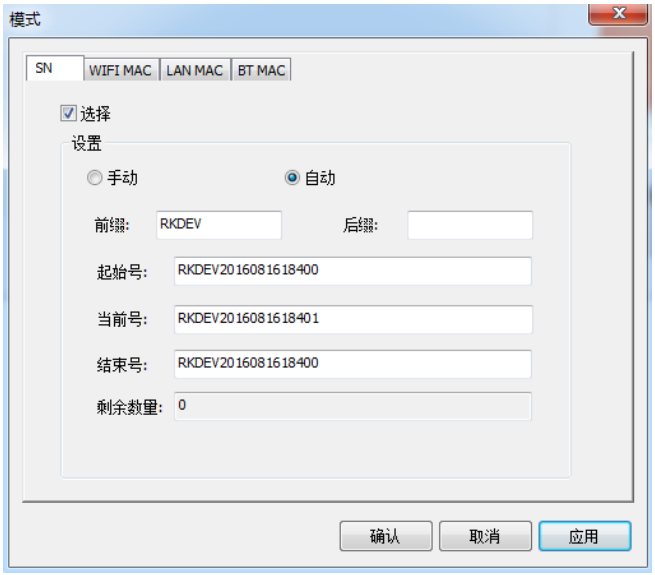


图 8-5WNpctool 工具模式设置

Picture 8-5 WNpctool tool mode setting

- 设置完成后，点击“应用”按钮，关闭模式设置窗口，返回主窗口。
  After setting, click “application” button, close mode setting window and back to the main window.
- 点击“写入”按钮即可。
  Click “Write” button.

8.8.2 使用 WNpctool 读取 Use WNpctool to read

1）进入 loader 模式。

Enter loader mode.

2）点击“设置”菜单，下拉框中勾选“读取”。

Click “setting” menu, and select “Read” in the drop-down box.

（勾选“读取”进行读取，未勾选“读取”则切换到写入功能）

(Select “Read” means to read, and deselect “Read” will convert to write function)

3）点击“读取”按钮即可。

Click “Read” button.

8.9 OemTool 打包工具 OemTool packing tool

8.9.1 Oem 打包工具步骤 Oem packing tool steps

下载分区默认 UserData 分区，可直接不填写。

The download partition is UserData partition by default, which can be directly left blank.

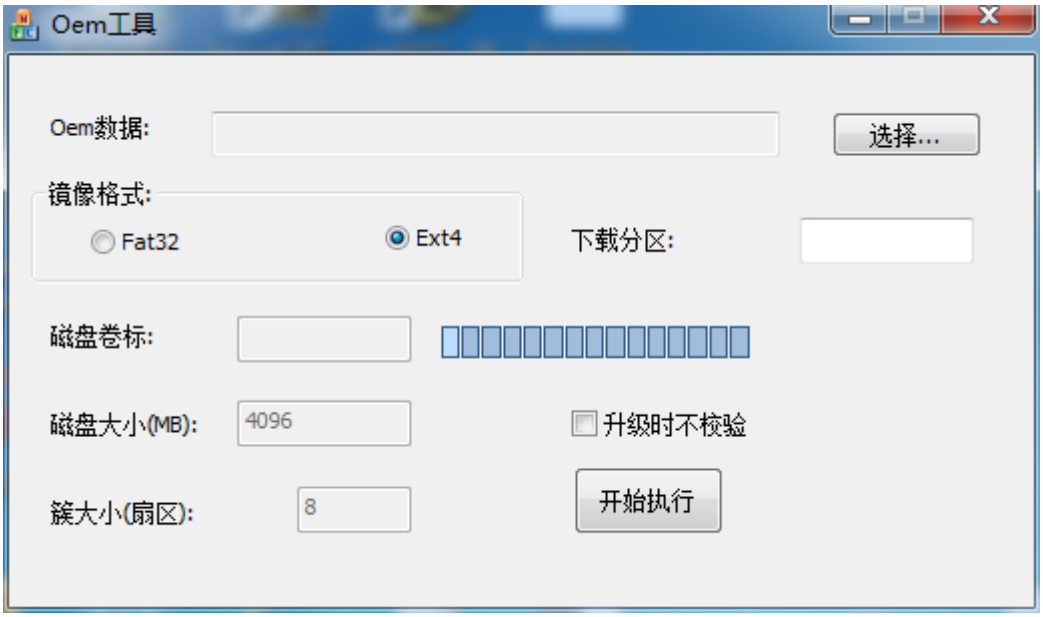


图 8-6 Oem 工具

Picture 8-6 Oem Tool

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，假设目录为/data/media/0,且 0 有一个文件为 sss.txt(如下图示)。则当升级完 demo 镜像的时候，会在设备的 data/media/0 目录下生成 sss.txt。

Click Choose button to select the data to be packed and the data must be directory. The default outermost directory is data directory, assuming that the directory is /data/media/0, and 0 has a file named sss.txt (as shown below). Then after demo mirror is updated, it will generate sss.txt in the data/media/0 directory of the device.

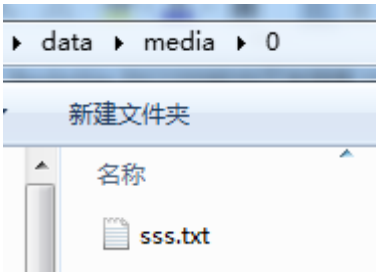


图 8-7 Oem 工具镜像制作文件夹路径要求

Picture 8-7 Oem tool image generation folder path requirement

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。

After select the file successfully, directly click Start to execute, and it will generate a mirror named OemImage.img in OEM tool directory. Just put the image on the FactoryTool to download.

8.10 量产工具使用 Production tool usage

8.10.1 工具下载步骤 Tool download steps

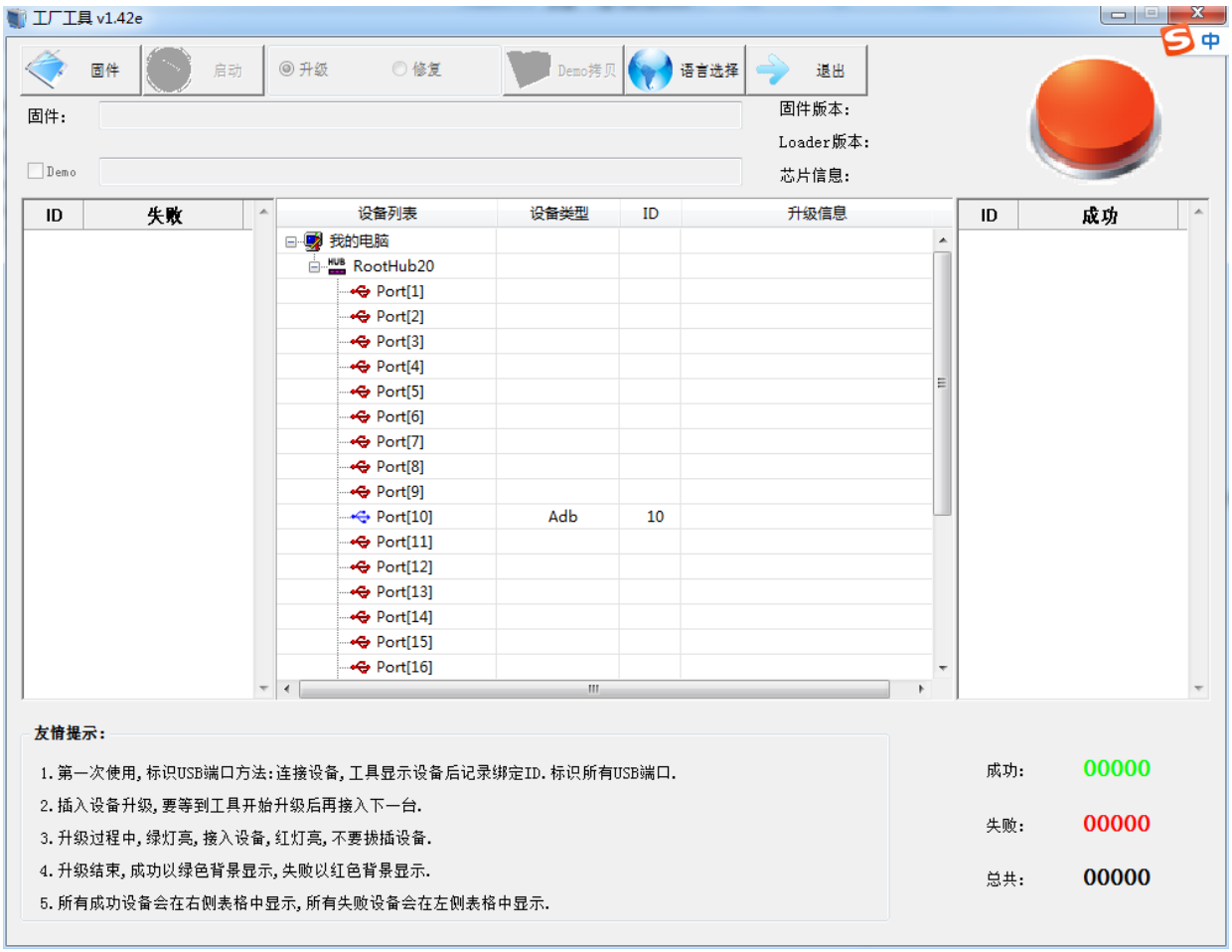


图 8-8 量产工具

Picture 8-8 Production tool

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。

Click image button, select the update.img packed by the package tool, and then wait for unpackage done.

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。

If need demo mirror, click Demo copy button, add the mirror packed by OEM tool, and single click Demo checkbox.

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。

Connect the device, make it enter loader or maskrom mode, and the tool will start to download automatically.

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。

It is able to connect multiple devices to do the flashing at the same time in order to improve the

factory flashing efficiency.

9 常见问题分析 FAQ

本章主要针对 RK3399 Android7.1 行业 SDK（RK3399_ANDROID7.1-Industry-SDK_V1.0_20180408）常见问题进行分析和解答，以方便客户更快的进行板级调试以及稳定性问题分析。

This chapter mainly describes the common issues analysis and solution for RK3399 Android7.1 industry SDK (RK3399_ANDROID7.1-Industry-SDK_V1.0_20180408), aiming to help customers to do the board level debugging and stability issue analyzing more quickly.

9.1 硬件设计对应软件修改 Software modifications corresponding to hardware design

本节主要介绍不同硬件设计对应的软件差异部分。

This section mainly describes the software differences corresponding to the different hardware designs.

9.1.1 如何选择 dts 配置 How to select dts configuration

客户根据不同参考设计，部分核心电路可能会有差异，比如参考平板的原理图以及挖掘机 SDK 或者 firefly 板子进行设计的电路，在软件配置上有些许的不同。首先客户在新机器进行板级调试前，先选择对应的 dts 配置，参考以下原则：

Some core circuits may be different as customers use different reference design, such as the circuits designed referring to tablet schematic and excavator SDK or firefly board, the software configurations will have some differences. Before customers do the board level debugging for the new device, first need to select the corresponding dts configuration referring to below principle:

- 参考平板以及挖掘机原理图设计的硬件请基于 rk3399-sapphire-excavator-edp.dts 进行配置，如果是 lpddr4 的机器，请在这个 dts 基础上加上 5.8 节描述的内容。

  If the hardware is designed referring to tablet and excavator schematic, please configure based on rk3399-sapphire-excavator-edp.dts. If the device uses lpddr4, please also add the contents described in section 5.8.

- 参考 firefly 板子原理图进行设计的硬件请基于 rk3399-firefly-android.dts 进行配置，如果是 lpddr4 的机器，请在这个 dts 基础上加上 5.8 节描述的内容。

  If the hardware is designed referring to firefly board schematic, please configure based on rk3399-firefly-android.dts. If the device uses lpddr4, please also add the contents described in section 5.8.

9.1.2 IO Domain 配置 IO Domain configuration

9.1.2.1 配置方法 Method to config

硬件上各个 IO domain 如下图所示：

The hardware IO domains are shown as below:

Part Port	Domain	Pin name in datasheet	I/O type
Part C	PMUIO1	pmuio1_gpio0ab	1.8V only
Part E	PMUIO2	pmu1830_gpio1abcd	1.8V(Default) 3.0V
Part I	APIO1	gmac_gpio3abc	3.3V only
Part L	APIO2	bt656_gpio2ab	1.8V(Default) 3.0V
Part G	APIO3	wifi/bt_gpio2cd	1.8V only
Part K	APIO4	gpio1830_gpio4cd	1.8V 3.0V(Default)
Part J	APIO5	audio_gpio3d_gpio4a	1.8V(Default) 3.0V
Part F	SDMMC0	sdmmc_gpio4b	1.8V 3.0V(Default)

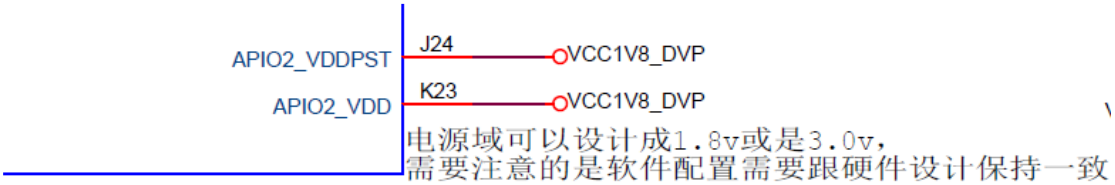
IO Domain 如果配置错误，可能出现一些莫名其妙的问题，在新机器调试前，先确认 io-domain，请参考文档“RKDocs/common/IO-Domain/Rockchip-Developer-Guide-Linux-IO-DOMAIN-CN.pdf”，并且结合板子电路图进行配置。下面简单描述下配置的方法：

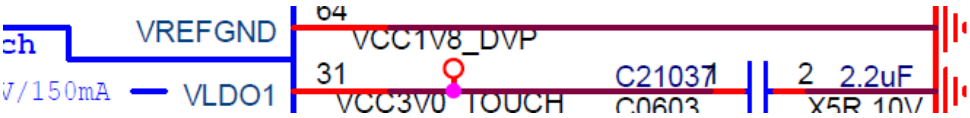
If IO Domain configuration is wrong, some curious issues may occur. Before debugging new device, first confirm io-domain, referring to the document “RKDocs/common/IO-Domain/Rockchip-Developer-Guide-Linux-IO-DOMAIN-CN.pdf”, and configure based on the circuit diagram of the board. Simply describe the configuration method as below:

- bt656-supply: The supply connected to APIO2_VDD.
- audio-supply: The supply connected to APIO5_VDD.
- sdmmc-supply: The supply connected to SDMMC0_VDD.
- gpio1830 The supply connected to APIO4_VDD.

上面列出了几个可配置的 io-domain 对应的芯片管脚名称，例如 bt656 对应的是 APIO2_VDD，所以需要查询电路图，确认 APIO2_VDD 连到了 PMU 的哪个 LDO 输出，举例说明：

Several chipset pin names corresponding to the configurable io-domain are listed as above, such as bt656 is corresponding to APIO2_VDD, so need to check the circuit diagram to confirm that APIO2_VDD is connected to which LDO output of PMU. For example:





上图中 APIO2_VDD 连到了 RK808 的 LDO1 上，RK808 的 LDO1 的定义如下：

APIO2_VDD is connected to LDO1 of RK808 as shown in above picture. LDO1 of RK808 is defined as below:

```
vcc1v8_dvp: LDO_REG1 {
  regulator-always-on;
  regulator-boot-on;
  regulator-min-microvolt = <1800000>;
  regulator-max-microvolt = <1800000>;
  regulator-name = "vcc1v8_dvp";
  regulator-state-mem {
    regulator-off-in-suspend;
  };
};
```

所以 bt656-supply 要定义为 vcc1v8_dvp，如下所示：

So bt656-supply should be defined as vcc1v8_dvp, shown as below:

```
&io_domains {
        status = "okay";

        bt656-supply = <&vcc1v8_dvp>;          /* bt656_gpio2ab_ms */
        audio-supply = <&vcca1v8_codec>;       /* audio_gpio3d4a_ms */
        sdmmc-supply = <&vcc_sd>;              /* sdmmc_gpio4b_ms */
        gpio1830-supply = <&vcc_3v0>;          /* gpio1833_gpio4cd_ms */
};
```

9.1.2.2 U-BOOT 配置 IO-DOMAIN configuration

SDK 默认使用 rkdevelop 分支 uboot，该 uboot 代码不支持读取 dts 的 io-domain 节点进行配置，这样会导致以下问题：

SDK uses rkdevelop branch uboot by default. This uboot code doesn't support to read io-domain node in dts for configuration, which will lead to the following problems:

- U-boot 下设置 gpio 口的 iomux 状态不对，或者某些 io 的电平不对，比如 pwm 的幅度不对。

The iomux status of gpio port configured under u-boot is incorrect, or some io level is incorrect, for example, the amplitufe of pwm is incorrect.

- 如果 u-boot 阶段未设置 io-domain，芯片会加载默认的配置，直到内核加载了 dts 正确的配置，这个持续时间大概会有 3~4s。如果某个 io-domain 连到 3.0v，但是默认的配置为 1.8v，则会出现电平不匹配的情况，有可能会烧坏 rk3399 的 io。

  If io-domain is not configured in u-boot stage, the chipset will load the default configuration until kernel finish loading the correct configuration from dts, which may take around 3~4s. If some io-domain is connected to 3.0V, but the default configuration is 1.8V, the level will mismatch, and it may burn io of RK3399.

在行业 sdk V2.9 版本更新了 rkdevelop 分支 u-boot 配置 io-domain 的修改，请首先按照 9.1.2.1 节的内容配置好 kernel 的 io-domain，确认 u-boot 和 kernel 有下述两个提交：

Industry SDK V2.9 version has updated the change of rkdevelop branch u-boot io-domain configuration. Please firstyly configure kernel io-domain according to section 9.1.2.1, and confirm there are the following two commits in u-boot and kernel:

```
u-boot：
commit 5a4e354a116b45b8ac2311aaa3464a7e88cd13d5
Author: Jianqun Xu <jay.xu@rock-chips.com>
Date:   Thu Mar 19 15:38:19 2020 +0800

    power: support rockchip io-domain set

    This patch should work with kernel dtb node property - "uboot-set"
    in io-domain node or pmu-io-domain node.

    Change-Id: I86fa43d234091310ee08978ec42f5895fb010a8c
    Signed-off-by: Jianqun Xu <jay.xu@rock-chips.com>
    Signed-off-by: Wenping Zhang <wenping.zhang@rock-chips.com>
Kernel：
commit f95ad65aabffa7450ff90a0dd4778cd5500de937
Author: Jianqun Xu <jay.xu@rock-chips.com>
Date:   Fri Mar 20 09:52:57 2020 +0800

    arm64: dts: rockchip: add 'uboot-set' in io-domain node for rk3399.

    Add 'uboot-set' for io-domain node, which used by u-boot board init
    to set io domain.

    Change-Id: Ia897e240c04d0b362551952278dadc8b4f2a8f2f
    Signed-off-by: Jianqun Xu <jay.xu@rock-chips.com>
    Signed-off-by: Wenping Zhang <wenping.zhang@rock-chips.com>
```

【注意】kernel 的提交只是修改了我司开发板的 dts，请自行将下述的修改放到贵司对应的板级 dts 配置中，并且根据原有的 dts 配置，指定 uboot-set 字段的值，比如 bt656-supply = <&vcc_3v0>，则表明 bt656 这个 io-domain 为 3v，需要在 uboot-set 字段中指定 RK3399_BT656_VDD_3V3。

[Note] kernel commit only modified dts for RK development board, please put the following change into your board level dts configuration by self, and specify the value of the field uboot-set according to original dts configuration, for example, for bt656-supply = <&vcc_3v0>, which means bt656 io-domain is 3V, need to specify RK3399_BT656_VDD_3V3 in the uboot-set field.

```
diff --git a/arch/arm64/boot/dts/rockchip/rk3399-sapphire.dtsi
b/arch/arm64/boot/dts/rockchip/rk3399-sapphire.dtsi
index a309c08d96d0..7d99b3acafbe 100644
--- a/arch/arm64/boot/dts/rockchip/rk3399-sapphire.dtsi
+++ b/arch/arm64/boot/dts/rockchip/rk3399-sapphire.dtsi
@@ -41,6 +41,7 @@
   */

 #include "dt-bindings/pwm/pwm.h"
+#include <dt-bindings/soc/rockchip-io-domain.h>
 #include "rk3399.dtsi"
 #include "rk3399-opp.dtsi"

@@ -518,6 +519,28 @@
 &io_domains {
         status = "okay";

+       /*
+        * If the uboot use rkdevelop branch, the io-domain is not set
+        * during uboot phase, it will cause the io voltage is not suit for
+        * the hardware,maybe will damage the io of rk3399, so we add
+        * uboot-set value to make sure the value is same with the kernel,
+        * ofcause you must make sure firstly the value here is suit for your
+        * hardware. The available value is defined in
+        * include/dt-bindings/soc/rockchip-io-domain.h, they are
+        * also list here for more convenient configuration:
+        * bt656-supply: RK3399_BT656_VDD_1V8 or RK3399_BT656_VDD_3V3
+        * audio-supply: RK3399_AUDIO_VDD_1V8 or RK3399_AUDIO_VDD_3V3
+        * gpio1830-supply: RK3399_GPIO1833_VDD_1V8 or RK3399_GPIO1833_VDD_3V3
+        * sdmmc-supply: don't need to configure because the voltage
+        *                                is set in source code during runtime.
+        * NOTE:
+        * 1. The value in uboot-set must be enclosed in "()", otherwise will
```

```
+             *         cause compile error.
+             * 2. The uboot-set configuration must be cooperated with submit
+             *         in u-boot.
+             */
+            uboot-set = <(RK3399_BT656_VDD_3V3 | RK3399_AUDIO_VDD_1V8 |
+                                        RK3399_GPIO1833_VDD_3V3)>;
            bt656-supply = <&vcc_3v0>;                    /* bt656_gpio2ab_ms */
            audio-supply = <&vcca1v8_codec>;              /* audio_gpio3d4a_ms */
            sdmmc-supply = <&vccio_sd>;                   /* sdmmc_gpio4b_ms */
@@ -538,6 +561,18 @@

 &pmu_io_domains {
            status = "okay";
+            /*
+             * If the uboot use rkdevelop branch, the pmu_io_domain is not set
+             * during uboot phase, it will cause the io voltage is not suit for
+             * the hardware,maybe will damage the io of rk3399, so we add
+             * uboot-set value to make sure the value is same with the kernel,
+             * ofcause you must make sure firstly the value here is suit for your
+             * hardware. The available value is defined in
+             * include/dt-bindings/soc/rockchip-io-domain.h, they are
+             * also list here for more convenient configuration:
+             * pmu1830-supply: RK3399_PMU1830_VDD_1V8 or RK3399_PMU1830_VDD_3V0
+             */
+            uboot-set = <RK3399_PMU1830_VDD_3V0>;
            pmu1830-supply = <&vcc_3v0>;
 };
```

9.1.3 Vdd_log 电压差异 Vdd_log voltage difference

VDD_LOG 电压采用的是 PMUIO2 电源域里面的 PWM（Pin M28）进行调压。PMUIO2 可配置为 1.8V（参考图）或 3.0V（挖掘机），所以 PWM 出来的逻辑电平也分 1.8V 和 3.0V 两种，对应的上拉电阻电源和调压电阻的参数也不一样。

VDD_LOG voltage uses PWM（Pin M28） of PMUIO2 power domain to the voltage regulation. PMUIO2 can be configured as 1.8V (reference circuit) or 3.0V (excavator), so the logic level from PWM is also divided into 1.8V and 3.0V, and the corresponding parameters of pull-up resistance power supply and voltage regulating resistance are also different.

VDD_LOG 电压我们要求在 0.90~0.95V 之间，不能太低，太低可能导致稳定性问题，也不建议太高，太高则功耗很高（超过 0.99v 可能会引入稳定性问题），由于电阻实际的阻值不是很精确，有可能电压并不是软件配置出的实际值，所以强烈建议在配置完以后量下实际的 vdd_Log 的值，

如果不对再微调下参数。另外上一段中描述的两种硬件设计中，RK808 对应的 io 电压也不一样，这块也需要软件进行修改，以下分别说明：

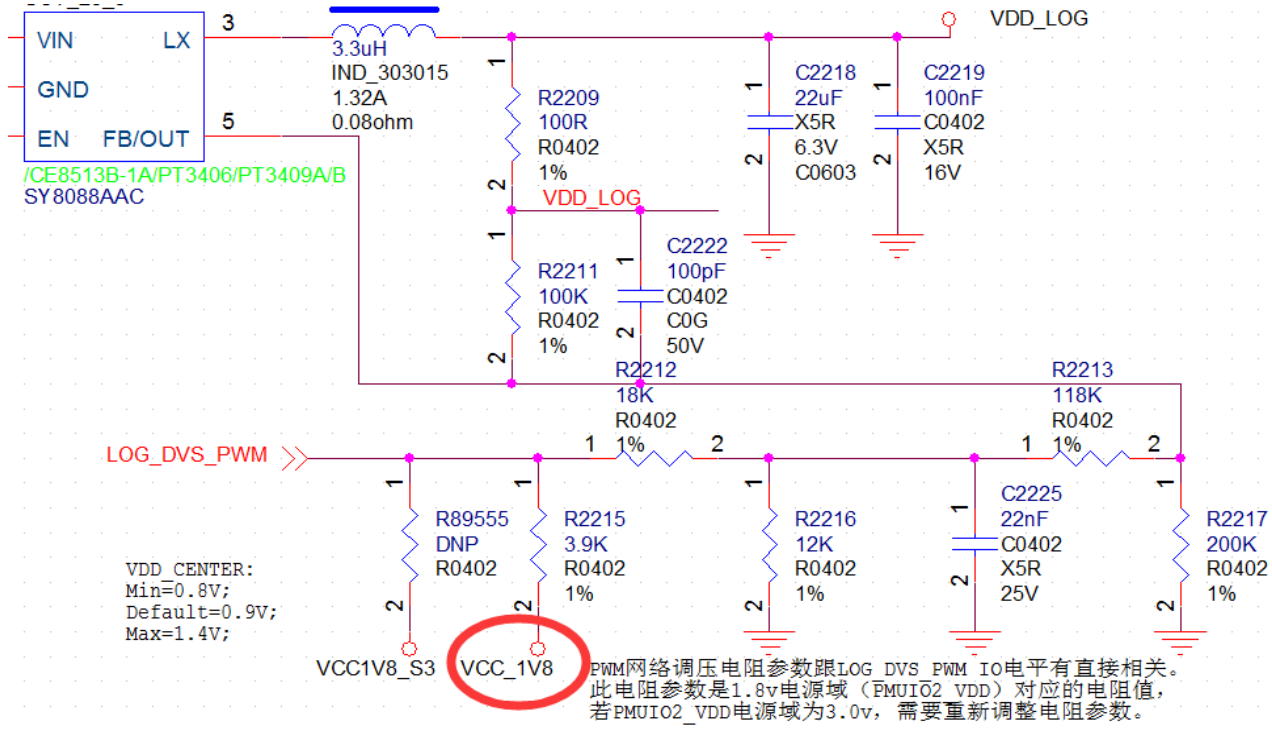
VDD_LOG voltage is required to be within 0.93~0.95V, too low may cause stability issue, while too high will cause the power consumption too high. As the actual resistance is not very accurate, maybe the voltage is not the actual value configured by the software, so it is strongly recommended to measure the actual value of vdd_Log after configuration, and fine tune the parameter if it is not correct. Besides, for the two hardware designs described in last paragraph, the corresponding io voltages of RK808 are also different, which also should be modified by software, separately introducing as below:

1. PMUIO 为 1.8V

PMUIO is 1.8V

PMUIO2 配置为 1.8V 的上拉电源和电阻参数，如下图所示：

The pull-up power and resistance parameter is shown as below when PMUIO2 is configured as 1.8V:



请参考如下配置：

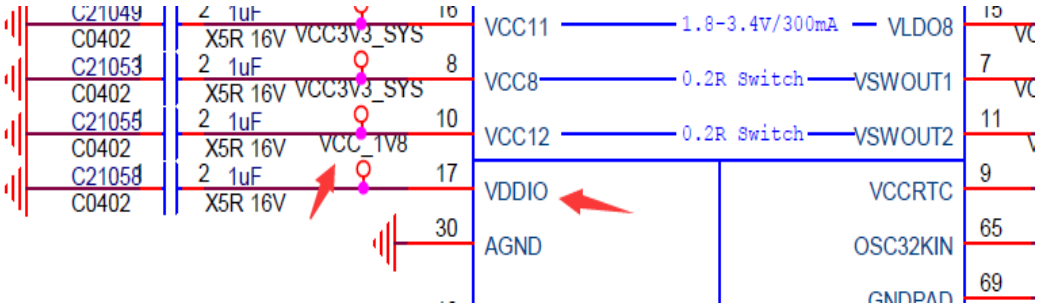
Please refer to the following configuration:

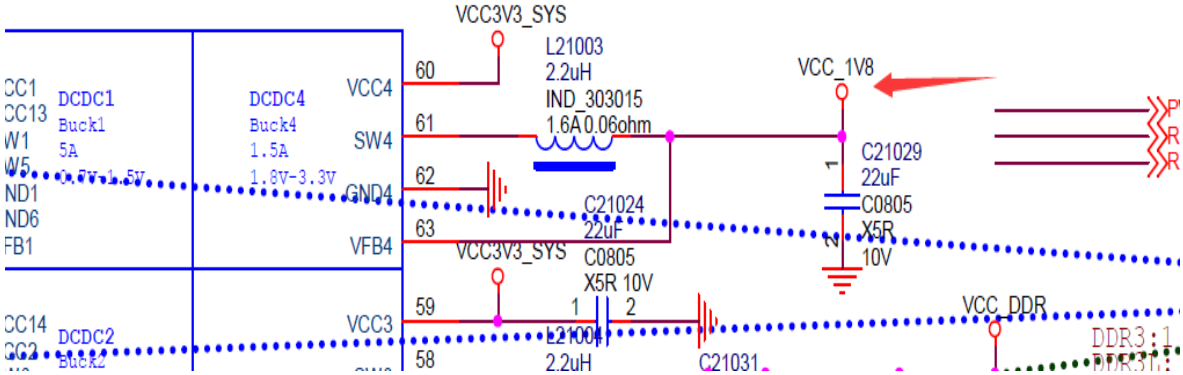
```
vdd_log: vdd-log {
        compatible = "pwm-regulator";
        pwms = <&pwm2 0 25000 1>;
        regulator-name = "vdd_log";
        regulator-min-microvolt = <800000>;
        regulator-max-microvolt = <1400000>;
        regulator-always-on;
        regulator-boot-on;
```

```
                /* for rockchip boot on */
                rockchip,pwm_id= <2>;
                rockchip,pwm_voltage = <900000>;
        };
```

RK808 的 VDDIO 要用 VCC_1V8 供电，如下图所示：

VDDIO of RK808 should use VCC_1V8 power supply, as shown below:





这里 RK808 的 VDDIO 使用 VCC_1V8 供电，而 VCC_1V8 连到了 RK808 的 DCDC4，所以 RK808 的 dts 节点里面配置：

Here VDDIO of RK808 uses VCC_1V8 power supply, but VCC_1V8 is connected to DCDC4 of RK808, so need to configure in the dts node of RK808:

```
vddio-supply = <&vcc_1v8>;
```

vcc_1v8 对应的电源为 DCDC4：

The corresponding power supply of vcc_1v8 is DCDC4:

```
                vcc_1v8: DCDC_REG4 {
                        regulator-always-on;
                        regulator-boot-on;
                        regulator-min-microvolt = <1800000>;
                        regulator-max-microvolt = <1800000>;
                        regulator-name = "vcc_1v8";
                        regulator-state-mem {
                                regulator-on-in-suspend;
                                regulator-suspend-microvolt = <1800000>;
```

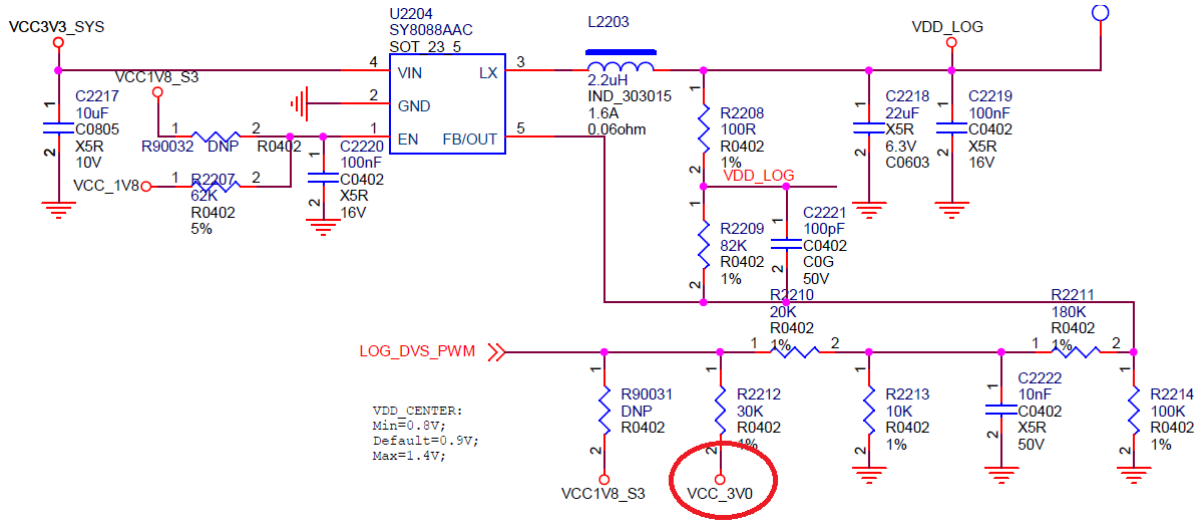
```
                                    };
                            };
```

2. PMUIO 为 3.3V

PMUIO is 3.3V

PMUIO2 配置为 3.0V 的上拉电源和电阻参数：

The pull-up power and resistance parameter when PMUIO2 is configured as 3.0V:



请参考如下配置：

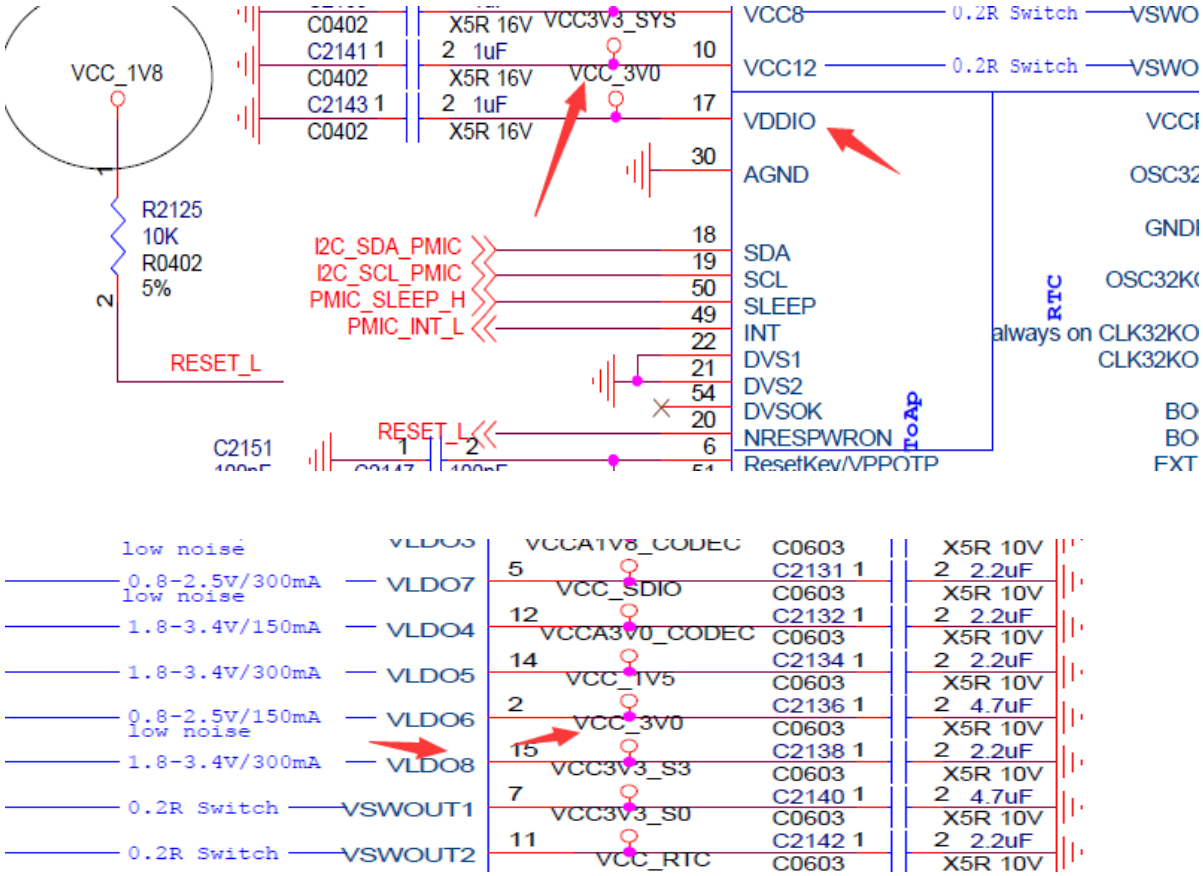
Please refer to the following configuration:

```
        vdd_log: vdd-log {
                compatible = "pwm-regulator";
                pwms = <&pwm2 0 25000 1>;
                regulator-name = "vdd_log";
                regulator-min-microvolt = <800000>;
                regulator-max-microvolt = <1400000>;
                regulator-always-on;
                regulator-boot-on;

                /* for rockchip boot on */
                rockchip,pwm_id= <2>;
                rockchip,pwm_voltage = <900000>;
        };
```

RK808 的 VDDIO 要用 VCC_3V0 供电：

VDDIO of RK808 should use VCC_3V0 power supply:



这里 RK808 的 VDDIO 使用 VCC_3V0 供电，而 VCC_3V0 连到了 RK808 的 LDO8，所以 RK808 的 dts 节点里面配置：

Here VDDIO of RK808 uses VCC_3V0 power supply, but VCC_3V0 is connected to LDO8 of RK808, so need to configure in the dts node of RK808:

```
vddio-supply = <&vcc_3v0>;
```

vcc_3v0 对应的电源为 LDO8：

The corresponding power supply of vcc_3v0 is LDO8:

```
vcc_3v0: LDO_REG8 {
        regulator-always-on;
        regulator-boot-on;
        regulator-min-microvolt = <3000000>;
        regulator-max-microvolt = <3000000>;
        regulator-name = "vcc_3v0";
        regulator-state-mem {
                regulator-on-in-suspend;
                regulator-suspend-microvolt = <3000000>;
        };
};
```

9.2 常见稳定性问题分析 Common stability issues analysis

本章主要介绍 RK3399 Android7.1 行业 SDK 中碰到的一些常见的稳定性问题以及解决思路和办法。

This chapter mainly describes some common stability issues and solutions for RK3399 Android7.1 industry SDK.

稳定性问题分成两类，一类是代码中确实存在软件 bug；一类是随机死机问题，下面分开描述。

Stability issues are divided into two categories, one is the software bug actually existing in the code, and the other is the random crash issue. Separately describe as below.

9.2.1 软件 BUG 导致死机或者卡死问题 Crash issue caused by software BUG

在某些特定场景下出现死机或者卡住问题，并且这类死机有着固定的死机 log 或者完全没有任何异常 log，这类问题需要分析具体的 log 确认问题点，下面主要介绍可能存在的类别并且需要抓取的 log。

If crash or hang issue occurs in some specific case, and this kind of crash has fixed crash log or doesn't have any abnormal log, need to analyze the specific log to locate the problem for this kind of issue. Below we mainly describe the possible category and the log needed.

1. 内核的死机

   Kernel crash

   如果内核的死机每次都是相同的 log，需要抓取 kernel log（串口 log 或者 last log），并且保留死机版本对应内核编译出来的 vmlinux（在 kernel 根目录下），然后提供给 RK 进行分析。

   If kernel crash with the same log every time, need to capture kernel log (serial port log or last log), and keep vmlinux (in kernel root directory) compiled by the corresponding kernel of the crash version, and then provide to RK for analysis.

2. 内核驱动死锁

   Kernel driver deadlock

   这种场景下内核的 log 有可能会有一些堆栈打印出来（接串口才能打印），另外还需要提供以下信息，在串口终端中输入“fiq”三个字符，会进入 fiq 模式，在该模式下连续输入几次“sysrq w”，从这些 log 可能可以看出一些异常，如果无法自行定位，请将打印得到的信息以及内核 log 一起发给 RK 进行确认。

   In this case, kernel log may print out some stack (need to connect the serial port to print), beside, also need to provide below information, input “fiq” at the serial terminal will enter fiq mode, and then continuously input “sysrq w” several times in this mode, there may be some abnormal seen from these logs. If you cannot locate by yourself, please send the print information and kernel log to RK for analysis.

3. Android 上层的死锁或者某些库崩溃问题

   Deadlock in Android upper layer or some lib crash issue

   该场景下需要一些 android 端的 log 来定位问题，出现此问题时，请抓取 aplog，aplog 默认开启，位于机器中的/data/logs/目录，请将该目录全部 pull 出来（同时抓取出问题时的 logcat）。另

外，一些 android 死锁问题会产生 ANR 信息，请获取/data/anr/traces.txt 文件进行分析，如果出现此类问题，并且无法自行定位，请将上述所有信息发送给 RK 进行分析和定位。更详细的调试方法请参考文档：RKDocs/android/ Rockchip_Developer_Guide_Crash_Issue_Quick_Analysis_CN.pdf

In this case, need some Android log to locate the problem. When issue occurs, please capture aplog, which is enabled by default, in /data/logs/ directory of the device. Pull out the whole directory (capture the logcat at the same time). Besides, some android deadlock issue may have ANR information, please capture /data/anr/traces.txt file to do analysis. If this kind of issue happenes, and you cannot locate by yourself, please send all the above information to RK to do analysis and location. For more detailed debugging method, please refer to the document " RKDocs/android/ Rockchip_Developer_Guide_Crash_Issue_Quick_Analysis_CN.pdf".

9.2.2 硬件相关导致的软件死机问题 Software crash issue caused by hardware

如果在调试过程中碰到一些莫名其妙的死机问题，并且这些死机问题出现时，log 中的堆栈是随机的，或者死机的点在内核很核心的函数，或者 Android 某些核心库出现非法指针，则需要关注以下几个方面：

If you encounter some inexplicable crash issues during debugging, and when these crash issues occur, the stack in the log is random, or the crash point is in the very core function of kernel, or there are illegal Pointers in some core libraries of Andoird, you need to pay attention to the following aspects:

1. DDR 相关 DDR related

RK3399 支持 DDR3、DDR3L、LPDDR3、LPDDR4 多种颗粒，在 DDR 选型上，请关注 RK FAE 窗口定期更新的 DDR 支持列表，根据里面的型号进行选型。如果存在一些特殊情况，确实需要使用某个颗粒，并且不在支持列表中，请联系 FAE 窗口协调新颗粒的调试事宜。如果有在支持列表，请确认更新最新的 loader 和 trust 是否能够解决问题，如果 u-boot 使用默认的分支，更新 u-boot 即可，如果有参考本文档 4.7 节切换 next-dev 分支的 u-boot，则更新 rkbin 即可更新 loader 和 trust。

RK3399 supports DDR3, DDR3L, LPDDR3 and LPDDR4 components. For DDR selection, please pay attention to DDR support list updated periodically through RK FAE window. Please select the model in the list. If in some special case, you need to use the component which is not in the support list, please contact with FAE window for the new component debugging coordination. If the component is in the supprt list, please confirm if the issue can be resolved or not after updating the latest loader and trust. If u-boot is using the default branch, only need to update u-boot. If you have switched to u-boot of next-dev branch referring to section 4.7 of this document, update rkbin will update loader and trust.

另外，确保内核有按照本文档的 5.8 节进行配置，目前 LPDDR4 只支持 400M、800M（V2.2 版本以后频率变更为 416M、856M）两个频点，因此需要注意 LPDDR4 与 DDR3/DDR3L/LPDDR3 的频点不同，LPDDR4 的频率表中不能包含 400M、800M 以外的频点，否则会导致系统异常。

Besides, ensure the kernel is configured according to section 5.8 of this document. Currently LPDDR4 only supports 400M and 800M two frequencies, so need to pay attention to that the frequency of LPDDR4 is differnent from DDR3/DDR3L/LPDDR3, the frequency table of LPDDR4 cannot include other frequencies except 400M and 800M, otherwise it will cause system abnormality.

如果软件上无法定位到原因，请将问题登记到 RK redmine 系统，并且发出 pcb 和原理图，申

请 RK 硬件进行 review，可能是信号或者电源上的异常导致。

If the root cause cannot be located from software, please raise issue in RK redmine system and attach the pcb and schematic, applying for RK hardware review, as the issue may be caused by signal or power abnormality.

2. IO domain 配置错误

IO domain configuration error

请参考 9.1.2 节进行配置。

Please configure referring to section 9.1.2.

3. 电源的异常

Power abnormality

如果出现随机性死机问题，并且上述章节中的内容均已确认无误，则建议量下各路供电是否有异常，请重点排查以下几个部分：

If random crash issue occurs, and all the contents in above chapters are confirmed correctly, it is recommended to measure the power supplies to see if there is abnormality. Please especially check the following aspects:

- CPU 频率及电压

  CPU frequency and voltage

1) 请勿随意修改 RK 提供的频率电压表；

   Please do not modify the frequency voltage table provided by RK.

2) 早期的行业 SDK 版本在低温情况下会出现概率性死机问题，如果发现版本比较旧，请更新到 V1.4 以后的版本（强烈建议代码更新到最新）；

   Previous industry SDK versions may occur probabilistic crash issue in low temperature environment, if the version is old, please update to V1.4 or higher version (strongly recommend to update the code to the latest version).

3) CPU 供电的纹波，如果纹波比较大，则需要将电压适当往上抬，但是需要注意，电压的增加是以 12.5mV 为单位，不能随意增减电压值，因为 PMIC 无法输出非 12.5mV 倍数的电压；

   If the ripple of CPU power supply is relatively large, the voltage should be appropriately increased. However, it is noted that the unit of the voltage increment is 12.5mV. The voltage cannot be arbitrarily increased or decreased, because PMIC cannot output the voltage not multiple of 12.5mV.

- Logic 电压（vdd_log）

  Logic voltage (vdd_log)

Logic 电压与 IO Domain 电压配置有关，强烈建议客户在板级调试的时候都量下 vdd_Log 的实际电压，确保这个电压在 0.9V 以上，建议定在 0.90~0.95v 左右，具体请参考 9.1.3 节进行配置和确认。在 sdk 版本 V2.7 以后，有解决一个 vdd_log 电压为 0.9~0.95 之间开关机会死机的问题，具体请参考 9.12 节。

Logic voltage is related with IO Domain voltage configuration. Strongly recommend customers to measure the actual voltage of vdd_Log during board level debugging to make sure the voltage is above 0.9V, recommend to set the voltage between 0.90~ 0.95v . Please refer to section 9.1.3 to do the configuration and confirmation. After sdk version V2.7, we resolved a boot-up crash issue with vdd_log voltage between 0.9~0.95v. For more details please refer to section 9.12.

- DDR 电压
  DDR voltage

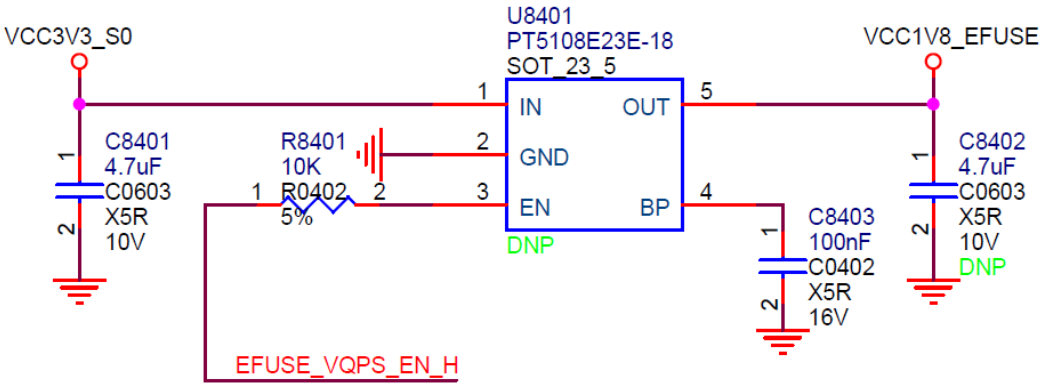
  确保 DDR 颗粒供电符合颗粒要求。

  Ensure that DDR power supply meets the requirement of the component.

9.3 Secure Boot 常见问题 Secure Boot common issues

烧写 efuse 以及对固件进行签名和固件烧写的说明请参考“RKDocs/rk3399/ RK3399_Efuse_Operation_Instructions_V1.00_20190214_EN.pdf”文档进行操作，需要注意的是，efuse 的烧写需要确保下图中的 VCC1V8_EFUSE 供电正常，该 LDO 的控制脚软件上是固定的（EFUSE_VQPS_EN_H），为 GPIO4_D3，不能改变。

For the instruction of effuse flashing and image signing and flashing, please refer to the document “RKDocs/rk3399/ RK3399_Efuse_Operation_Instructions_V1.00_20190214_EN.pdf”. It should be noted that, effuse flashing needs to ensure that the power supply of VCC1V8_EFUSE in below picture is normal. The control pin (EFUSE_VQPS_EN_H) of this LDO is fixed by software as GPIO4_D3 and cannot be changed.



9.4 Data 分区切换为 EXT4 后出现的问题 Issues occur after switching data partition to EXT4

DATA 分区切换为 EXT4 格式存在两个问题需要考虑：

There are two issues that should be considered for switching DATA partition to EXT4 format:

- 已量产客户不建议将 data 分区切为 EXT4 格式，因为切换分区格式会导致 data 分区格式化，并且 ota 升级存在问题。
  It is not recommended for MP customers to switch data partition to EXT4 format, because the partition format switch will cause the data partition to be formatted and there will be problem with ota upgrade.
- Recovery 擦除分区慢问题。
  It is slow to erase the partition during Recovery.
  以上具体说明请参考本文档 6.23 节内容进行修改。
  Please refer to section 6.23 of this document to do the modification.

9.5 更新代码后必现卡在 android 动画问题 100% stuck in android animation issue after updating the code

这种情况比较大的可能有两个：

There are probably two reasons for this case:

1. SDK Android 和 kernel 都有更新，但是客户只更新了 kernel 或者只更新了 android。
   Both SDK Android and kernel are updated, but customers only update kernel or android.
2. 客户的内核自己有新建一个 defconfig，导致 SDK 对于内核 defconfig 的更新没有同步到客户自己新建的 defconfig 上，需要客户把 arch/arm64/configs/rockchip_defconfig 上更新的内容同步到自己的 defconfig 上。
   Customers create a defconfig in their kernel, which causes kernel defconfig update of SDK is not synced to the defconfig created by customers, need customers to sync the update of arch/arm64/configs/rockchip_defconfig to their own defconfig.

上述的问题其实都是因为 GPU 库的上层和下层不匹配导致 android 无法启动，这个是目前碰到比较多的情况，如果不是上述所说问题，请根据 log 具体分析。

In fact, the above issues are all caused by the mismatch between the upper and lower layers of GPU lib, which leads to the failure of android bootup. This is the common issue encountered currently. If it is not the issue mentioned above, please make specific analysis according to log.

9.6 LPDDR4 开启负载变频 LPDDR4 enables load frequency scaling

负载变频可以较为显著的降低 lpddr4 的功耗，在 V2.5 版本 SDK 之前，请不要开启负载变频，可能会引起稳定性问题。建议 sdk 更新到 V2.6 版本以后再开启负载变频，并且开启负载变频后，测试下稳定性，负载变频配置方法如下述红字部分：

Load frequency scaling can obviously lower down the power consumption of lpddr4. Recommend to disable load frequency scaling fuction before V2.6, otherwise may cause stability issue. Please enable the load frequency scaling after update sdk version to V2.6 or above, and do the stability test after enable load frequency scaling.The configuration method of load frequency scaling is shown as below in red:

```
&dmc {
```

```
        status = "okay";
        center-supply = <&vdd_center>;
        system-status-freq = <
            /*system status              freq(KHz)*/
            SYS_STATUS_NORMAL           666000
            SYS_STATUS_REBOOT           666000
            SYS_STATUS_SUSPEND          328000
            SYS_STATUS_VIDEO_1080P      666000
            SYS_STATUS_VIDEO_4K         856000
            SYS_STATUS_VIDEO_4K_10B     856000
            SYS_STATUS_PERFORMANCE      856000
            SYS_STATUS_BOOST            856000
            SYS_STATUS_DUALVIEW         856000
            SYS_STATUS_ISP              856000
        >;
        vop-bw-dmc-freq = <
        /* min_bw(MB/s) max_bw(MB/s) freq(KHz) */
            0           762         416000
            763         3012        666000
            3013        99999       856000
        >;
        auto-min-freq = <328000>;
        auto-freq-en = <1>;
    };
```

LPDDR4 的负载变频在 sdk 端已经测试过绝大多数场景，但是由于行业客户的应用场景多样化，我们无法覆盖所有场景，请在开启负载变频的时候多测试下，确认是否有 ddr 带宽不足导致的闪屏问题。

SDK has tested most scenarios for load frequency scaling of LPDDR4, but because the industry customers' applications are diverse, we cannot cover all the scenarios. Please do more test after load frequency scaling is enabled to confirm whether there is any flicker issue caused by insufficient ddr bandwidth.

9.7 DDR 带宽导致屏闪问题 Flicker issue caused by DDR bandwidth

客户某些应用场景会出现屏闪的问题，看 log 会出现下面的打印：

Flicker issue may occur in some application of customer, and the log will have below print:

```
rockchip-vop ff900000.vop: [drm:vop_isr] ERROR POST_BUF_EMPTY irq err
```

这种一般出现在 4K 输出，或者播放 4k 视频的场景，根源为 ddr 带宽不够，针对这个请确认以下几点：

Generally it occurs in the scenario of 4K output, or 4K video playback, and the root cause is ddr bandwidth is insufficient. Please confirm the following points:

1. 由于这种场景下 ddr 带宽要求比较高，所以建议对应的场景变频的频点设置为最高频点，即改变 dmc 节点下面的对应场景变频的频点，【注】因为代码一直在更新，请根据 5.8 节说明按照我们参考的 lpddr4 频点进行配置：

   Because ddr bandwidth requirement is relatively high in this case, recommend to set the corresponding frequency of the scenario frequency scaling as the highest frequency, that is, to change the corresponding frequency of the scenario frequency scaling in dmc node:

```
system-status-freq = <
        /*system status          freq(KHz)*/
        SYS_STATUS_NORMAL        800000
        SYS_STATUS_REBOOT        528000
        SYS_STATUS_SUSPEND       200000
        SYS_STATUS_VIDEO_1080P   300000
        SYS_STATUS_VIDEO_4K      600000 //举例说明，如果发现播放 4k 视频的场景会出现闪屏，请将该对应频点改为 800000    For example, if flicker occurs while playing 4k video, please modify the corresponding frequency to 800000
        SYS_STATUS_VIDEO_4K_10B 800000
        SYS_STATUS_PERFORMANCE   800000
        SYS_STATUS_BOOST         400000
        SYS_STATUS_DUALVIEW      600000
        SYS_STATUS_ISP           600000
    >;
```

2. 如果上面的修改无效，并且代码开启了 ddr 负载变频，建议关闭负载变频（会导致功耗上升）：

   If the above modification is invalid, and the code enables ddr load frequency scaling, it is recommended to disable the load frequency scaling (will increase the power consumption):

```
&dmc {
    auto-freq-en = <0>;//默认为 1，改为 0   default value is 1, change to 0
    … …
};
```

3. 如果上面的修改仍然无效，请更新到行业 sdk v2.1 版本之后，并且确认 rk3399.dtsi 中有下面的参数 vop-pn-msch-readlatency：

   If the above modification is still invalid, please update to the version higher than industry sdk v2.1, and confirm there is the parameter vop-pn-msch-readlatency in rk3399.dtsi:

```
dmc: dmc {
        compatible = "rockchip,rk3399-dmc";
        devfreq-events = <&dfi>;
        interrupts = <GIC_SPI 1 IRQ_TYPE_LEVEL_HIGH 0>;
        clocks = <&cru SCLK_DDRCLK>;
        clock-names = "dmc_clk";
        ddr_timing = <&ddr_timing>;
        upthreshold = <40>;
        downdifferential = <20>;
        system-status-freq = <
                /*system status         freq(KHz)*/
                SYS_STATUS_NORMAL       800000
                SYS_STATUS_REBOOT       528000
                SYS_STATUS_SUSPEND      200000
                SYS_STATUS_VIDEO_1080P  300000
                SYS_STATUS_VIDEO_4K     600000
                SYS_STATUS_VIDEO_4K_10B 800000
                SYS_STATUS_PERFORMANCE  800000
                SYS_STATUS_BOOST        400000
                SYS_STATUS_DUALVIEW     600000
                SYS_STATUS_ISP          600000
        >;
        vop-pn-msch-readlatency = <
                /* plane_number  readlatency */
                0       0
                4       0x20
        >;
        auto-min-freq = <400000>;
        auto-freq-en = <1>;
```

vop-pn-msch-readlatency 这个参数指定了两列数据，第一列为 UI 的层数，第二列为 vop 获取 ddr 使用权限的延时，这个值越小，表示越容易获取访问权限，值为 0 表示默认值（默认值为 0x80），如果出现闪屏问题，建议将值改为 0x20，这个值不建议随意修改，并且这个值如果改动可能会影响性能。目前默认的值，只会在 UI 为 4 层的场景下，将 vop 的 ddr read lantency 改为 0x20，也就是在 4 层场景下提升 vop 对 ddr 访问的时效性，从而让 vop 更快的获取到数据。

The parameter vop-pn-msch-readlatency specifies two columns of data, the first column is the quantity of UI layer, and the second column is the latency for vop acquiring ddr usage authority, which indicates that it is easier to acquire the access authority if the value is smaller. 0 is the default value (default value is 0x80). If flicker issue occurs, recommend to change the value as 0x20. However, it is recommended not to change this value arbitrarily and the modification of this value may influence the performance. Currently the default value will only change ddr read latency of vop to 0x20 in the scenario of 4-layer UI, that is, improve the timeliness of vop access to ddr in 4-layer scenario, so as to make vop obtain data faster.

```
&dmc {
    vop-pn-msch-readlatency = <
            /* plane_number    readlatency */
            0           0 // readlatency 值为 0，表示默认值 0x80     readlatency value is 0, indicating the default value 0x80
            4           0x20
    >;
    …… 
};
```

如果上面默认的修改仍然无法解决闪屏问题，请尝试将 vop-pn-msch-readlatency 中的第一行的

值从 0 改为 0x20，也就是无论 UI 的层数是几层，都减小 vop 访问 ddr 的延时，如下所示：

If the above default modification still cannot resolve the flicker issue, please try to change the value in the first column of vop-pn-msch-readlatency from 0 to 0x20, that is, no matter how many layers UI has, decrease the latency of vop access to ddr, as shown below:

```
&dmc {
        vop-pn-msch-readlatency = <
                /* plane_number    readlatency */
                0               0x20
                4               0x20
        >;
        … …
};
```

【注意】最后一个修改，请确认所有的修改无效后再加该修改，因为这个修改可能会影响 cpu、gpu 访问 ddr 的时效性，进而可能影响性能（比如跑分会低）。

[Note]Please confirm all the previous modifications invalid before adding the last modification, because this modification may affect the timeliness of CPU and GPU accesses to ddr, and may further influence the performance (such as low scores).

9.8 音频 dma 通道不够导致声卡注册失败问题 Codec register failed issue due to lack of dma channel

如果在系统注册声卡时，出现下述的报错 log：

If sound card registered failed during system booting, and the output log is such as:

```
[    4.015736] rockchip-spdif ff870000.spdif: Missing dma channel for stream: 0
[    4.015871] rockchip-spdif ff870000.spdif: ASoC: pcm constructor failed: -22
[    4.015898] rockchip-cdndp-sound dp-sound: ASoC: can't create pcm DP PCM :-22
[    4.015923] rockchip-cdndp-sound dp-sound: ASoC: failed to instantiate card -22
[    4.016396] rockchip-cdndp-sound dp-sound: rockchip_sound_probe snd_soc_register_card fail -22
```

这个报错是因为 spdif 对应的 dma 控制器 dmac_bus 最多只能同时申请 6 个 channel：

This is because of the DMAC0 controller can only support 6 channels at the same time:

DMAC0 supports the following features:

- Supports Trustzone technology
- Supports 12 peripheral request
- Up to 64bits data size
- 6 channel at the same time
- Up to burst 16
- 12 interrupts output and 1 abort output
- Supports 32 MFIFO depth

Following table shows the DMAC0 request mapping scheme.

Table 13-1 DMAC0 Request Mapping Table

Req number	Source	Polarity
0	I2S0 tx	High level
1	I2S0 rx	High level
2	I2S1 tx	High level
3	I2S1 rx	High level
4	I2S2 tx	High level
5	I2S2 rx	High level
6	PWM	High level
7	SPDIF tx	High level
8	SPI5 tx	High level
9	SPI5 rx	High level
10	Reserved	
11	Reserved	

而代码中配置的 channel 数已经超过了 6 个，有如下两个解决方案：

But the registered dma channel is more than 6, there are 2 sollutions:

1. 请确认 I2S0/I2S1/I2S2/SPDIF/PWM/SPI5 这些外设是否有使用到 dma，并且已经使能，如果这些接口没有用到，请禁用对应的节点节省出 dma 通道，比如 I2S1 没有使用，则 diable 掉 I2S1 即可：

   Please check these peripherals like I2S0/I2S1/I2S2/SPDIF/PWM/SPI5 are enabled, and check they are really in use on your hareware, if not , disable the dts node to save the dma channel for sound card. For example, if i2s1 is not in use, disable it as below:

```
&i2s1 {
        status = “disabled”;
};
```

2. Hdmi 音频使用 i2s2，对 hdmi 音频来说只需要使用 dma 的 tx 通道，可以去掉 i2s2 的 dma 的 rx 通道，如下所示：

   Hdmi sound card use i2s2, but for hdmi sound card, it’s only use dma tx dma channel, and rx dma channel is useless for hdmi sound, so we can disable rx dma channel for i2s2, such as:

```
hdmi_sound: hdmi-sound {
 … …
  simple-audio-card,cpu {
    sound-dai = <&i2s2>;//hdmi_sound use i2s2
  };
  simple-audio-card,codec {
    sound-dai = <&hdmi>;
  };
 };
```

去掉 i2s2 的 dma rx 通道：

Disable dma rx channel for i2s2:

```
diff --git a/arch/arm64/boot/dts/rockchip/rk3399.dtsi b/arch/arm64/boot/dts/rockchip/rk3399.dtsi
```

```
index ab30e35..3dc8e34 100644
--- a/arch/arm64/boot/dts/rockchip/rk3399.dtsi
+++ b/arch/arm64/boot/dts/rockchip/rk3399.dtsi
@@ -1751,8 +1751,8 @@
                        compatible = "rockchip,rk3399-i2s", "rockchip,rk3066-i2s";
                        reg = <0x0 0xff8a0000 0x0 0x1000>;
                        interrupts = <GIC_SPI 41 IRQ_TYPE_LEVEL_HIGH 0>;
-                       dmas = <&dmac_bus 4>, <&dmac_bus 5>;
-                       dma-names = "tx", "rx";
+                       dmas = <&dmac_bus 4>;
+                       dma-names = "tx";
                        clock-names = "i2s_clk", "i2s_hclk";
                        clocks = <&cru SCLK_I2S2_8CH>, <&cru HCLK_I2S2_8CH>;
                        power-domains = <&power RK3399_PD_SDIOAUDIO>;
```

9.9 SDMMC 高速模式和 SDIO3.0 支持 SDMMC high-speed mode and SDIO3.0 mode support

Sdmmc 如果要支持高速 SD 卡，请在 dts 配置中加上如下配置：

If want to support sdmmc ultra high speed mode, please add below configs in dts:

```
+&sdmmc {
+           sd-uhs-sdr104;
+};
```

SDIO 如果要支持 sdio3.0 模式，请在 dts 中修改对应的时钟为 150M，如下所示：

If want to support sdio3.0 mode, please change the clock of sdio to 150M as below:

```
&sdio0 {
-           clock-frequency = <50000000>;
-           clock-freq-min-max = <200000 50000000>;
+           clock-frequency = <150000000>;
+           clock-freq-min-max = <200000 150000000>;
            supports-sdio;
            bus-width = <4>;
            disable-wp;
```

上述这些修改都需要首先确保板子信号能够满足高速模式要求，请自行测试兼容性和稳定性。

First of all, need to make sure the signal quality of hardware can meet the requirement of sdmmc ultra high-speed mode and sdio3.0 mode. Please test the stability and compatibility by youself.

9.10 media_profile.xml 文件损坏导致的机器无法开机问题

The device boot failure caused by media_profile.xml file damage

如果机器支持 Camera，则在开机过程中，会在 data 分区生成一个配置文件，文件路径为：/data/camera/media_profiles.xml，如果在生成该文件过程中，机器突然断电，则会导致该文件错误，在重启后，由于该文件错误，则会导致机器卡在 android logo。从行业 sdk 版本 V2.5 以后，修正了该 bug，如果解析发现该文件错误，则重新生成，避免服务无法开启。该补丁涉及三个目录，如果要单独更新该补丁，请更新如下提交：

If the device supports camera, during boot up, it will generate a configuration file in data partition with the file path: /data/camera/media_profiles.xml. If the power-down happens suddenly during the file generation, it will cause the file error. After restart, the device will be blocked on android logo due to the file error. Industry sdk already fixes this bug from version V2.5. If the file error is found during parsing, it will regenerate the file to prevent the service from being blocked. This patch involves three directories, and if you want to update this patch separately, please update the following commit:

hardware/rockchip/camera

```
commit ff93b2f3c7b771113fc1c013971598fc00f00358
Author: chenxiao <cx@rock-chips.com>
Date:   Mon Sep 9 17:14:20 2019 +0800

    Camera:
        usb camera hot-plug update.

    Change-Id: Ia5a0bcf60be43c7d76462505874fc896690d9aa7
    Signed-off-by: chenxiao cx@rock-chips.com
```

frameworks/base

```
commit 40a7012eb8f88fd96f3a4c8bbb237c4f1119c3fc
Author: chenxiao <cx@rock-chips.com>
Date:   Mon Sep 9 17:12:29 2019 +0800

    Camera:
        usb camera hot-plug update.

    Change-Id: I4f84c967a0c1f8cc4e19b80bf86ba085d087d49f
    Signed-off-by: chenxiao cx@rock-chips.com
```

frameworks/av

```
commit 2dac057d2a7617783ca7c215474df1572e67c1ef
Author: chenxiao <cx@rock-chips.com>
Date:   Mon Sep 9 17:13:36 2019 +0800
```

```
Camera:
    usb camera hot-plug update.

Change-Id: Ia126bd5a8681668f73f3d9eafd29c7adb910782d
Signed-off-by: chenxiao cx@rock-chips.com
```

9.11 PCBA 测试 UI 刷新慢问题 The issue of slow display during PCBA test

在行业 sdk V2.5 及之前的版本，pcba 测试显示刷新很慢，请更新到 v2.6 版本，或者在之前版本的基础上更新 external/rk-pcba-test 仓库下面的补丁：

For industry sdk v2.5 and former versions, pcba testing updates the UI very slowly. Please update the sdk version to v2.6, or update the below patch in external/rk-pcba-test directory:

```
external/rk-pcba-test$ git log
commit b02f8011450ce99fa13c27d88873f3a62d8f7e23
Author: lanshh <lsh@rock-chips.com>
Date:   Mon Apr 9 20:08:50 2018 +0800

    disabled GGL_BLEND in gr_fill
    optimize display speed

    Change-Id: I9ed4a754b200f745fae9c876a050f5754b341287
    Signed-off-by: lanshh <lsh@rock-chips.com>
```

9.12 【重要】ISP 导致的开机卡死问题 [Important] Boot-up crash issue caused by ISP

V2.7 版本解决了一个 isp 导致的开机卡死问题，问题现象为概率性的 reboot 卡死或者在 camera 和 hdmi-in 开关过程中死机，抬升 vdd_log 电压到 0.95v 以上可以降低该问题概率，V2.7 版本通过软件最终解决了该问题，更新后 vdd_log 电压不需要再抬高，考虑到客户布板因素，我们建议 vdd_log 在 0.9~0.95 之间，不要超过 0.99v，也不要低于 0.9v，否则可能引起稳定性问题，另外，电压太高也会影响芯片寿命。

V2.7 SDK fixes one boot-up crash issue caused by isp, which phenomenon is the device crashes during reboot or camera and hdmi-in on/off with probability. Increasing the voltage of vdd_log to 0.95v or higher can decrease the probability of this issue. V2.7 version fixes this issue completely through software, and no need to increase the voltage of vdd_log. Considering customers' layout, we recommend vdd_log should be 0.9~0.95, neigher higher than 0.99V, nor lower than 0.9V, otherwise it may lead to the stability issue, and too high voltage also will influence the chipset life.

该问题并不是只有在有使能 camera 或者 hdmi-in 的场景下才会出现，只要 dts 中 isp 节点有

打开就有问题。

This issue not only occurs in the scenario with camera or hdmi-in enabled, but also it may occur as long as the isp node is enabled in dts.

如果客户不想整体更新 sdk，请从最新的 kernel 上 cherry-pick 以下几个补丁：

If the customer doesn't want to update the whole sdk, please cherry-pick the following patches from the latest kernel:

```
03a2d9b camera: rockchip: camsys_drv: v0.0x30.2
8a22922 arm64: dts: rockchip: rk3399-android: add isp reset resource
e732998 camera: rockchip: camsys_drv: v0.0x30.1
921e6db camera: rockchip: camsys_drv: v0.0x30.0
25cc108 clk: rockchip: rk3399: Update the isp clocks
```

9.13 传递 EFUSE 标志到内核 Pass efuse tag to kernel

有些客户有固件 efuse 加密的需求，并且在产线需要确认某台机器是否加密成功（efuse 是否烧写成功），目前实现方式是在 loader 阶段读取 efuse 加密信息，并且通过 uboot 传递 command line 到 kernel，kernel 可以通过读取 command line 中的"fuse.programmed"字段，判断是否 efuse 烧写成功，"fuse.programmed=1"表示 efuse 已经烧写，"fuse.programmed=0"表示未烧写。本功能需要更新 u-boot 下面的下述几个提交：

Some customers have the requirement of image encryption with efuse and they need to confirm whether the device is encrypted successfully (whether efuse is flashed successfully or not) or not in the factory. Current implementation is to read efuse encryption information in loader stage and pass the tag in the command line to kernel through uboot. Kernel can judge whether effuse is flashed successfully or not by reading the field "fuse.programmed" in the command line. "fuse.programmed=1" means efuse is already flashed, while "fuse.programmed=0" means it is not flashed. This feature requires to update the following commits under u-boot:

```
commit 5fd2ec286e44bdfcf22174569d1f4e2158820dc7
Author: Joseph Chen <chenjh@rock-chips.com>
Date:   Fri Dec 6 09:32:02 2019 +0800

    rockchip: secureboot: add alias "fuse.programmed=..." for "Efuse_SecureBoot=..."

    - U-Boot nextdev branch uses this alias name, sync that solution;
    - Pass flag to kernel only when miniloader passes the fuse programmed magic;

    Optimise: (430a1cc rockchip: secureboot: pass efuse secure-boot flag to cmdline)

    Signed-off-by: Joseph Chen <chenjh@rock-chips.com>
    Change-Id: I61b73bd5d05cd55fc3412077d03f2f02671da3a8
```

```
commit 26f8c6866058334ef8ab99f9682d8d3d43810b0e
Author: Chen Fen <chenfen@rock-chips.com>
Date:   Thu Nov 28 14:13:37 2019 +0800

    rk3399: loader: update miniloader bin to v1.22.

    build from:
            c72d34 Pass efuse program flag to uboot.
    update feature:
            Pass efuse program flag to uboot.

    Signed-off-by: Chen Fen <chenfen@rock-chips.com>
    Change-Id: I412520ee6a1cabe078732dfeecebb9fda642836e
commit 430a1cc9be2718e531d7ec7a7ca39aeb122bed05
Author: Joseph Chen <chenjh@rock-chips.com>
Date:   Wed Oct 9 10:30:06 2019 +0800

    rockchip: secureboot: pass efuse secure-boot flag to cmdline

    Miniloader enables secure-boot once the fimware is signed even the
    efuse secure-boot flag is not enabled.

    Sometimes kernel needs to know whether the efuse secure-boot flag
    is enable or not.

    Change-Id: Iaf4e4aa237a47f3de9045d3ddae95a085feabea2
    Signed-off-by: Joseph Chen <chenjh@rock-chips.com>
```