密级状态： 绝密(　　) 秘密(　　) 内部资料(　　) 公开(　√)
Security Class: Most Confidential() Confidential() Internal() Public(√)

Rockchip_RK3229_Developer_Guide_Android 9.0_Box_CN&EN

（Technical Department, R & D Dept. I）

文件状态： File Status: [ ] 草稿 [ ] Draft [ ] 正在修改 [ ] Modifying [√] 正式发布 [√] Released	**文件标识：** File Identification:	RK-KF-YF-514
	当前版本： Current Version:	2.0.0
	作　者： Author:	XYP
	完成日期： Completion Date:	2019-07-19
	审　核： Checked By:	CW, HuangJC, ZXZ
	审核日期： Audit Date:	2019-07-19

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。

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。

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。

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

Warranty Disclaimer

This document is provided according to "current situation" and Fuzhou Rockchip Electronics Co., Ltd. (“the company”, the same below) is not responsible for providing any express or implied statement or warranty of accuracy, reliability, completeness, marketability, specific purpose or non-infringement of any statement, information and content of this document. This document is intended as a guide only.

Due to product version upgrades or other reasons, this document may be updated or modified from time to time without notice.

Brand Statement

Rockchip, “瑞芯微”, “瑞芯”, and other Rockchip trademarks are trademarks of Fuzhou Rockchip electronics Co., Ltd., and are owned by Fuzhou Rockchip electronics Co., Ltd.

All other trademarks or registered trademarks mentioned in this document are owned by their respective owners.

Copyright © 2019 Fuzhou Rockchip Electronics Co., Ltd.

Beyond reasonable use range, any unit or individual shall not extract or copy part or all of the content of this document, and shall not spread in any form without the written permission.

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

前言 Preface

概述 Overview

文档作为 Rockchip RK3229 软件开发指南，旨在帮助软件开发工程师和技术支持工程师更快上手 RK3229 的开发及调试。

This document mainly describes Rockchip RK3229 software development guide, aiming to help software engineers and technical support engineers familiar with RK3229 development and debugging quickly.

产品版本 Product Version

芯片名称 Chip Name	内核版本 Kernel Version	Android 版本 Android Version
RK3229	Linux4.4	Android9.0.0

读者对象 Readers

本文档（本指南）主要适用于以下工程师：

This document(or guide) is mainly applicable to below engineers:

- 技术支持工程师 Technical Support Engineers
- 软件开发工程师 Software Engineers

修订记录 Modification Records

日期 Date	版本 Version	作者 Author	审核 Checked By	修改说明 Remarks
2019-01-25	V1.00	XYP	CW，HuangJC	创建初始发布版本 Initial release
2019-07-19	V2.00	XYP	CW，HuangJC	更新文档、工具索引 Update document and tool index

Copyright © 2019 Fuzhou Rockchip Electronics Co., Ltd.

目录 Contents

Copyright © 2019 Fuzhou Rockchip Electronics Co., Ltd.

插图目录 Figure Contents

Copyright © 2019 Fuzhou Rockchip Electronics Co., Ltd.

表格目录 Table Contents

1 支持列表 Support List

1.1 DDR 支持列表 DDR Support List

RK3229 DDR 支持 DDR3、DDR3L、LPDDR2、LPDDR3。

RK3229 DDR supports DDR3, DDR3L, LPDDR2, LPDDR3.

Table 1-1 RK3229 DRAM Support Type

Chip	DRAM Support Type
RK3229	DDR3/DDR3L/LPDDR2/LPDDR3

RK3229 DDR 颗粒支持程度列表，详见 RKDocs\common\Platform support lists 目录下《RK DDR Support List Ver2.38.pdf》，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。

RK3229 DDR type support list, see “RK DDR Support List Ver2.38.pdf” for detail information under the directory of “RKDocs\common\Platform support lists”. We recommend you only use DDR type marked with the symbol √ and T/A as shown in below table.

Table 1-2 RK3229 DDR Support Symbol

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

1.2 NAND 支持列表 NAND Support List

RK3229 Nand Flash 支持程度列表，详见 RKDocs\common\Platform support lists 目录下《RKNandFlashSupportList Ver2.73_20180615.pdf》，下表中所标示的 NAND 支持程度表，建议选用√、 T/A 标示的颗粒。

RK3229 Nand Flash support list, see “RKNandFlashSupportList Ver2.73_20180615.pdf” for detail information under the directory of “RKDocs\common\Platform support lists”. We recommend you only use NAND type marked with the symbol √ and T/A as shown in below table.

如有选型上的疑问，也可直接联系 Rockchip FAE 窗口。

If you have any question about type selection, you can also contact with Rockchip FAE members directly.

Table 1-3 RK3229 NAND 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

Symbol	Description
N/A	Not Applicable

1.3 EMMC 支持列表 EMMC Support List

RK3229 支持 eMMC 4.51，SDIO3.0，支持 HS200 模式，详见 RKDocs\common\Platform support lists 目录下《RKeMMCSupportList Ver1.43_2019_03_15.pdf》，下表中所标示的 EMMC 支持程度表，只建议选用 √、T/A 标示的颗粒。

RK3229 supports eMMC4.51, SDIO3.0, and HS200 mode, see “RKeMMCSupportList Ver1.43_2019_03_15.pdf “ for detail information under the directory of “RKDocs\common\Platform support lists”. We recommend you only use EMMC type marked with the symbol √ and T/A as shown in below table.

Table 1-4 RK3229 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

1.3.1 高性能 EMMC 颗粒的选取 High Performance EMMC Module Type Selection

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

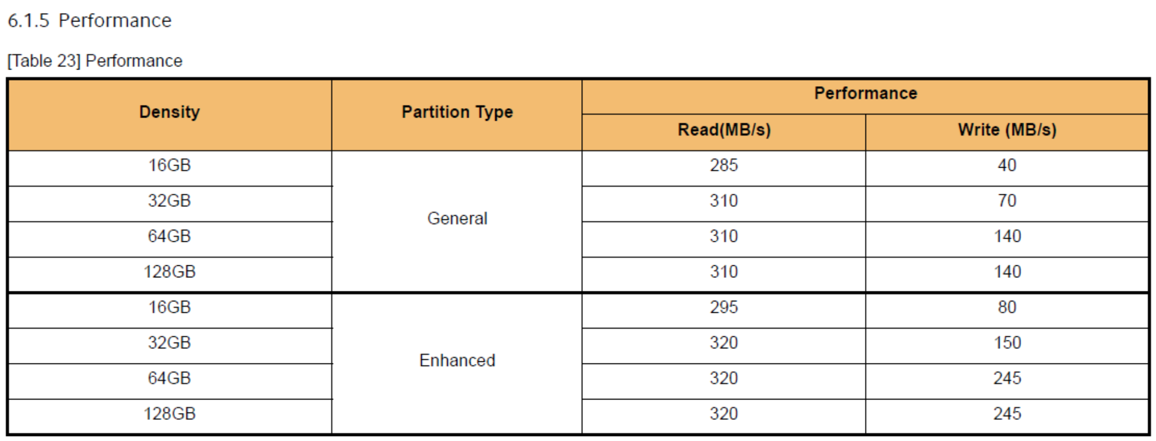
You need to choose high performance EMMC module type to improve system performance. Before you pick EMMC module type, refer to support list offered by Rockchip, and pay close attention to performance chapter in manufacturer’s datasheet.

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

Selection refer to manufacturer’s company size and EMMC’s read-write rate. Higher than 200MB/s sequence read rate, and higher than 40MB/s sequence write rate is recommended.

如有选型上的疑问，也可直接联系 Rockchip Fae 窗口。

If you have any question about type selection, you can also contact with Rockchip FAE members directly.



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

Figure 1-1 EMMC Performance Example

1.4 WiFi/BT 支持列表 WiFi/BT Support List

RK3229 Android9.0 的 Kernel 版本为 Linux4.4，WiFi/BT 支持列表，详见 RKDocs\common\Platform support lists 目录下《Rockchip_Introduction_WiFi_Situation_CN

.pdf》，文档列表中为目前 RK3229 上大量测试过的 Wifi/Bt 芯片列表，建议按照列表上的型号进行选型。如果有其他 WiFi/BT 芯片调试，需要 WiFI/BT 芯片原厂提供 Linux4.4 版本的内核驱动程序。

RK3229 Android9.0's kernel version is Linux4.4, WiFi/BT Support List, please refer to "Rockchip_Introduction_WiFi_Situation_CN.pdf" under the directory of "RKDocs\common\Platform support lists" for detail information. The WiFi/BT chips listed in the document is currently already tested a lot in RK3229. Recommend to choose the WiFi/BT chips in the table. If you want to debug other WiFi/BT chips, first need to communicate with WiFi/BT vendor whether they can provide the driver program which can work on Linux4.4 stably and technical support during debugging.

如果疑问和建议可以与 Rockchip Fae 窗口联系。

If you have any question about type selection, you can also contact with Rockchip FAE members directly.

1.5 SDK 软件包适用硬件列表 SDK Software Package Applicable Hardware List

本 SDK 是基于谷歌 Android9.0 32bit 系统，适配瑞芯微 RK3229 芯片的软件包，适用于 RK3229 Box Evb 开发板及基于其上所有的开发产品。

This SDK software package is based on Google Android9.0 32bit system, which is adaptive to Rockchip RK3229 chip. It is suitable for RK3229 Box EVB development board and all development products based on it.

若是基于 RK3229 Box Evb 开发板开发，内核配置可参考 rk3229-evb-android-avb.dts 进行改动。

If using RK3229 Box EVB board to develop, you can refer to "rk3229-evb-android-avb.dts" to configure kernel.

另 SDK 中附带了 RK3229 Box Evb 开发板的硬件使用说明。

Besides, along with SDK release, the hardware usage instruction of RK3229 Box EVB development board is also released.

Table 1-5 RK3229 Hardware Description List

硬件板 **Hardware Board**	**对应文档说明** **Relevant Document Description**
Evb 开发板 EVB Development Board	RKDocs\rk322x\ Rockchip_RK322X_Hardware_Design_Guide_V1.0_CN.pdf

1.6 多媒体编解码支持列表 Multimedia CODEC Support List

RK3229 多媒体规格，详见表 1-6:

RK3229 Multimedia specification，see details in Table 1-6:

Table 1-6 RK3229 Multimedia CODEC Support List

多媒体支持 **Multimedia Support**	支持 4K VP9 and 4K 8bits H264/H265 视频解码，高达 60fps。 Support 4K VP9 and 4K 8bits H264/H265 vedio decoding, up to 60fps.
	1080P 多格式视频解码 (VC-1, MPEG-1/2/4, VP8)。支持 JPEG 解码。 1080P multi format video decoding (VC-1, MPEG-1/2/4, VP8). Support JPEG decoding.
	1080P H.264 格式视频编码。支持 JPEG 编码。 1080P H.264 format video encoding. Support JPEG encoding.

RK3229 具体的编解码支持列表，详见 RKDocs\rk322x 目录下《Rockchip_RK3229_Multimedia_Codec_Benchmark_EN.pdf》

RK3229 detailed encoding and decoding support list, refer to “Rockchip_RK3229_Multimedia_Codec_Benchmark_EN.pdf” under the directory of “RKDocs\rk322x” for detailed information.

2 文档/工具索引 Document/Tool Index

2.1 文档索引 Document Index

随 RK3229 Box SDK 发布的文档旨在帮助开发者快速上手开发及调试，文档中涉及的内容并不能涵盖所有的开发知识和问题。文档列表也正在不断更新，如有文档上的疑问及需求，请联系我们的 Fae 窗口。

RK3229 Box 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.

RK3229 SDK 的 RKDocs 目录结构如下所示。

RK3229 SDK’s RKDocs directory structure is shown below:

```
RKDocs/
├── android
│   ├── bt
│   │   ├── Rockchip_Introduction_Android8.1_BT_Configuration_CN.pdf
│   │   └── Rockchip_Introduction_Android9.0_BT_Configuration_CN.pdf
│   ├── project.config
│   ├── Rockchip_Developer_Guide_Android_New_Partition_Configuration_CN.pdf
│   ├── Rockchip_Developer_Guide_PCBA_Test_Tool_CN&EN.pdf
│   ├── Rockchip_Developer_Guide_PCBA_Test_Tool_CN.pdf
│   ├── Rockchip_Introduction_Android8.0_Factory_Reset_Protection_CN.pdf
│   ├──
Rockchip_Introduction_Android8.0_Power_On_Off_Animation_and_Tone_Customization_CN.pdf
│   ├── Rockchip_Introduction_Android8.1_BOX_Display_Framework_Configuration_CN.pdf
│   ├── Rockchip_Introduction_Android9.0_AVB_Howto_CN.pdf
│   ├── Rockchip_Introduction_Android9.0_Safety_Boot_Solution_CN.pdf
│   ├── Rockchip_Introduction_Android9.0_System_New_Feature_CN&EN.pdf
│   ├── Rockchip_Introduction_Android_AB_System_Upgrading_CN.pdf
│   ├── Rockchip_Introduction_Android_AB_System_Upgrading_EN.pdf
│   ├── Rockchip_Introduction_Android_Application_Preinstallation_CN.pdf
│   ├── Rockchip_Introduction_Android_Performance_Mode_CN.pdf
│   ├── Rockchip_Introduction_Android_Verify_Boot_CN.pdf
│   ├── Rockchip_Introduction_Android_Widevine_Project_Start_Preparation_CN.pdf
│   ├── Rockchip_Introduction_Box_Media_Application_CN.pdf
│   ├── Rockchip_Introduction_PCBA_Camera_Porting_CN.pdf
│   ├── Rockchip_User_Guide_Magisk_Installation_EN.pdf
│   ├── Rockchip_User_Guide_Recovery_CN&EN.pdf
│   └── wifi
│       ├── Rockchip_Introduction_Android9.0_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.p
df
│   │   ├── 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_v2.0.pdf
│   │   │   ├── camera_hal3_user_manual_v2.1.pdf
│   │   │   ├── RKCIF_Driver_User_Manual_v1.0.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 Ver2.38.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.pdf
│   │   │   ├── 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_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
└── rk322x
    ├──Rockchip_RK3229_Android9.0_Box_SDK_Release_V1.0.0_20190125
_CN&EN.pdf
    ├── Rockchip_RK3229_Developer_Guide_Android9.0_Box_CN&EN.pdf
    ├── Rockchip_RK3229_Multimedia_Codec_Benchmark_EN.pdf
    └── Rockchip_RK322X_Hardware_Design_Guide_V1.0_CN.pdf
```

2.2 工具索引 Tool Index

随 RK3229 Box SDK 发布的工具，用于开发调试阶段及量产阶段。工具版本会随 SDK 更新不断更新，如有工具上的疑问及需求，请联系我们的 Fae 窗口。

Tools released with RK3229 Box SDK, is used in development debugging and MP stage. The tool version may update along with SDK update. Please contact our FAE if you have any question or requirement about the tool.

RK3229 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）。

RK3229 SDK contains linux (tools used in Linux operation system) and windows (tools used in Windows operation system) under RKTools directory.

Table 2-1 Tool Index Table

工具名称 **Tool Name**	工具说明 **Tool Description**	工具路径 **Tool Path**
AndroidTool	分立升级固件及整个 update 升级固件工具 seperated upgrade firmware and whole upgrade firmware tool	RKTools\windows\AndroidTool_Release_v2.65
FactoryTool	量产升级工具 MP upgrade tool	RKTools\windows\FactoryTool_v1.64
SecureBootTool	固件签名工具 firmware signature tool	RKTools\windows\SecureBootTool_v1.88
efuseTool	efuse 烧写工具 efuse flashing tool	RKTools\windows\efuse_v1.37
WNpctool	写号工具 a tool that writing SN, MAC and so on to the machine	RKTools\windows\WNpctool_Setup_V1.3.0_180813
SD_Firmware_Tool	SD 卡镜像制作 SD card mirror generation tool	RKTools\windows\SD_Firmware_Tool._v1.46
SpiImageTools	烧录器升级工具 userd for programmer upgrade tool	RKTools\windows\SpiImageTools_v1.36
DriverAssitant	驱动安装工具 driver installing tool	RKTools\windows\DriverAssitant_v4.5
OemTool	新增分区镜像制作工具 new partition mirror generating tool	RKTools\windows\OemTool_v1.3
Rockchip 平台 DDR 测试工具	DDR 测试工具 ddr test tool	RKTools\windows\Rockchip 平台 DDR 测试工具_V1.35
Rockchip Box 厂测工具	厂测工具 product test tool	RKTools\windows\Rockchip Box 厂测工具 V3.0-P-20181113
KeyWrite_v1.61	Widevine KEY 烧写工具 widevine key flashing tool	RKTools\windows\KeyWrite_v1.64.zip
Linux_Pack_Firmware	Linux 打包工具 linux firmware packing tool	RKTools\linux\Linux_Pack_Firmware
Linux_SecureBoot	Linux 签名工具 linux firmware signature Tool	RKTools\linux\Linux_SecureBoot

工具名称 **Tool Name**	工具说明 **Tool Description**	工具路径 **Tool Path**
Linux_Upgrade_Tool	Linux 烧写工具 linux firmware flashing Tool	RKTools\linux\Linux_Upgrade_Tool

3 SDK 编译/烧写 SDK Compiling/Burning

3.1 SDK 获取 SDK Acquirement

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《Rockchip_RK3229_Android9.0_Box_SDK_Release_V1.0.0_20190125_CN.pdf》，该文档与 SDK 一同发布。

SDK is released through Rockchip code server. Customers apply for SDK from Rockchip FAE contacter, and will be able to sync code after obtaining the server certificate authorization with SSH public key provided. For more details about Rockchip code server SSH public key authorization, please refer to《Rockchip_RK3229_Android9.0_Box_SDK_Release_V1.0.0_20190125_CN&EN.pdf》. This document is released along with SDK.

3.2 SDK 编译配置 SDK Compiling Configuration

3.2.1 分区大小配置 Partition Size Configuration

Sdk 默认分区表一般定义在对应产品目录下的 parameter.txt 文件中（例如：rk3229_box 对应 device/rockchip/rk322x/rk322x_box/parameter.txt）.

Generally, SDK default partiton table is defined in parameter.txt under corresponding product directory(e.g. rk3229_box matches “device/rockchip/rk322x/rk322x_box/parameter.txt”).

BOARD_SYSTEMIMAGE_PARTITION_SIZE 等系统分区宏配置会自动读取分区表中分区大小赋值。

System partiton macro configuration like BOARD_SYSTEMIMAGE_PARTITION_SIZE will automatically read the corresponding partiton size in partiton table for assignment..

若需要调整分区大小，修改对应分区表定义文件 parameter.txt 即可。具体参见 Parameter 说明 Parameter Instruction。

If you want to change partition size, you can edit corresponding partition table’s definition file parameter.txt. For detail information, refer to Parameter 说明 Parameter Instruction

注意：Android Pie 要求新增部分分区及说明如下：

Note: Android Pie requires adding some new partitions, descripbed as below:

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

dtb partition: used to store image of device tree blob, reserved, not burned by default.

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

dtbo partition: used to store image of device tree blob overlay, burn preset image under corresponding product directory by default.

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

vbmeta partition: used to store each partition’s validate data which is automatically generated by compiling in AVB(Android verified

```
boot) mode. At present, as SDK has been upgraded to support vboot2.0,
defaultly, vbmeta.img must be burned during program burning.
```

3.2.2 Android Pie 新增特性配置说明 Android Pie New Feature Configuration Instruction

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

By default, Android9.0 required new feature configuration modification is in the file of SDK source code "device/rockchip/common/BoardConfig.mk".

1. 要求默认使用 64bit binder 驱动

   Ask for using 64bit binder driver by default.

```
    #binder protocol(8)
    TARGET_USES_64_BIT_BINDER := true
```

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

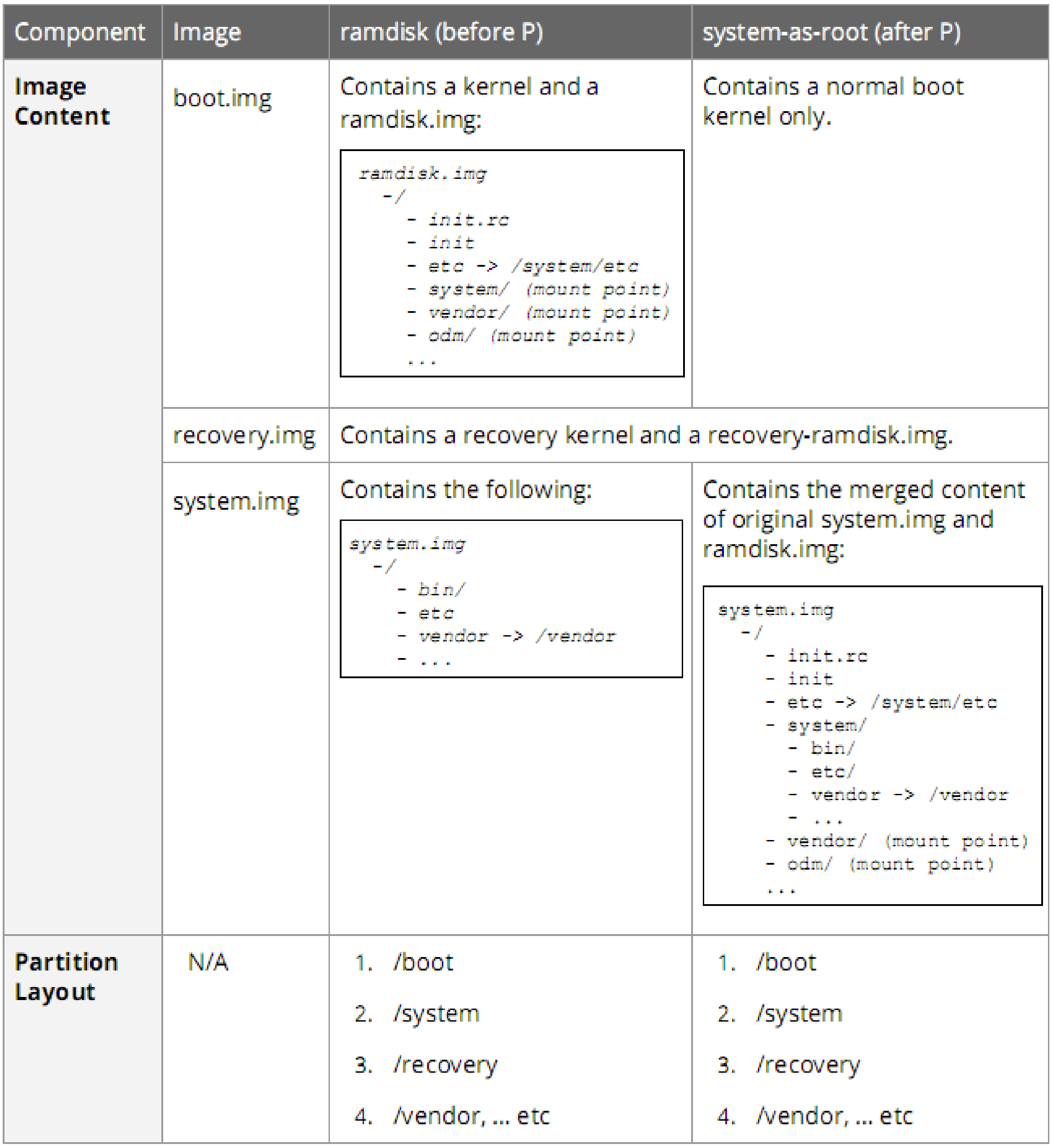
   Cmdline parameter configuration is modified in product configuration by default, not in parameter, e.g. the common selinux permission configuration and so on.

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

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

   Ask for opening system as root function by default, the main change is integrating ramdisk.img into system.img, mounting system as rootfs at system startup, the below figure shows partition image difference.

```
    BOARD_BUILD_SYSTEM_ROOT_IMAGE := true
```



Component	Image	ramdisk (before P)	system-as-root (after P)
Image Content	boot.img	Contains a kernel and a ramdisk.img: ramdisk.img -/ - init.rc - init - etc -> /system/etc - system/ (mount point) - vendor/ (mount point) - odm/ (mount point) ...	Contains a normal boot kernel only.
	recovery.img	Contains a recovery kernel and a recovery-ramdisk.img.	
	system.img	Contains the following: system.img -/ - bin/ - etc - vendor -> /vendor - ...	Contains the merged content of original system.img and ramdisk.img: system.img -/ - init.rc - init - etc -> /system/etc - system/ - bin/ - etc/ - vendor -> /vendor - ... - vendor/ (mount point) - odm/ (mount point) ...
Partition Layout	N/A	1. /boot 2. /system 3. /recovery 4. /vendor, ... etc	1. /boot 2. /system 3. /recovery 4. /vendor, ... etc

详细介绍参见谷歌开发者网站 system as root 说明。

For detailed introduction, refer to Google developer website’s instruction of “system as root”.

4. 要求默认开启 VNDK 检测及属性兼容性配置，具体参见谷歌开发者网站中 VNDK 说明。
   Ask for opening VNDK detection and property compatibility configuration defaultly. For detailed introduction, refer to Google developer website’s instruction of “VNDK”.

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

5. 是否配置 AVB(android verified boot)功能，具体参见谷歌开发者网站中 AVB 功能说明。目前 sdk 中 ATV 产品默认要求开启，BOX 产品默认关闭。
   Whether need to configure AVB(android verified boot) function, please refer to Google developer website’s insturction of “AVB function”. Currently, SDK’s ATV product requires AVB function to be enalbed by default, and BOX product requireds AVB function to be disabled by default.

```
# Enable android verified boot 2.0
BOARD_AVB_ENABLE ?= false
```

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

   In userdebug mode, it’s required to enable DEXPREOPT optimization, to improve system boot speed and application first time start up speed.

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

Note: enabling DEXPREOPT optimization will influence operation of pushing apk and jar file during developing debug period, direct push can be invalid.

若需要临时版本可以关闭此开关，参考下面修改配置:

If temporary version with this function disabled is required, refer to below modification to configure:

副作用：由于 art 中会校验镜像，关闭后每次开机时间会很长。

Side effect: As image is checked in art, if you disable this function, everytime system’s boot time will be long.

```
    WITH_DEXPREOPT ?= true
    diff --git a/BoardConfig.mk b/BoardConfig.mk
    index 8543697..cb81367 100644
    --- a/BoardConfig.mk
    +++ b/BoardConfig.mk
    @@ -167,18 +167,6 @@ TARGET_PROVIDES_INIT_RC ?= false
     //MAX-SIZE=512M, for generate out/.../system.img
     BOARD_FLASH_BLOCK_SIZE := 131072

    +WITH_DEXPREOPT_BOOT_IMG_AND_SYSTEM_SERVER_ONLY ?= true
    +
    +# Enable dex-preoptimization to speed up first boot sequence
    +ifeq ($(HOST_OS),linux)
    +  ifeq ($(TARGET_BUILD_VARIANT),user)
    +    ifeq ($(WITH_DEXPREOPT),)
    +      WITH_DEXPREOPT ?= true
    +    endif
    +  else
    +    WITH_DEXPREOPT ?= false
    +  endif
    +endif

     ART_USE_HSPACE_COMPACT ?= true
```

3.2.3 jack-server 配置 Jack-server Configuration

Android9.0 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：

Android9.0 uses jack-server as java compiler, below similar wrong information may appear during compilation:

```
    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。

This situation is mainly due to jack-server compiler’s limit itself, the same network port can not 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 Android9.0 时，需要配置各自使用不同的网络端口号。

That is, in the process of collaborative development on the server, multiple users compile Android9.0 at the same time, they need to configure each to use a different network port number.

jack-server 的两个配置文件，决定了它所使用的端口号：

jack-server's two configuration files, determine the port number the user uses.

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

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

These two configuration files need to configure two port numbers, that is server-side port number and client-side port number, two files' port number must match.

```
jack.server.service.port=8074
jack.server.admin.port=8075
及
and
SERVER_PORT_SERVICE=8074
SERVER_PORT_ADMIN=8075
```

配置步骤如下：

Configuration steps are below:

1) 确保两个配置文件存在，并且权限设置为 0600：

Make sure these two configuration files are existed, and their permission is set to be 0600:

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

2) 若两个配置文件不存在，请参照以下文本新建这两个配置文件。

If these two configuration files are not existed, please new these two files following below text.

config.properties 文件示例如下（端口号需按实际修改）：

e.g. config.properties file(port number requires to be modified according to the actual fact)

```
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）：

e.g. .jack-settings file(port number requires to be modified according to the actual fact)

```
# 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：

Modify port numbers, please modify service.port and admin.port, two files' port numbers must match.

```
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。

Recompile Android, see whether there is wrong information, if so, please change to other port number untill compiling is passed.

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 you have tried for five times modifying port number, and compiling is always failed, you can run command “jack-admin dump-report”, extract generated compressed file, and then analyse log record. If below log appears, you can reinstall libcurl:

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

3.2.4 全自动编译脚本 Script for Full-automatic Compiling

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。

In order to improve the compiling efficiency and lower down the possible mistake operation of manual compiling, this SDK integrates the script for full-automatic compiling, 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 script for full-automatic compiling is saved in:

```
device/rockchip/rk322x/build_box.sh
```

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

When repo sync, it will be copyed to the project root directory through copy option in the manifest

```
    <project path="device/rockchip/rk322x"
name="rk/device/rockchip/rk322x" remote="rk"
revision="rk33/mid/8.0/develop">
        <copyfile src="buildspec_box.mk" dest="buildspec.mk"/>
        <copyfile src="build_box.sh" dest="build_box.sh"/>
    </project>
```

3） 修改 build.sh 脚本中的特定变量以编出对应产品固件。

Edit specific variables in build.sh to build coresponding product firmware.

```
KERNEL_DTS=rk322x-evb-android-avb
```

变量请按实际项目情况，对应修改：

Please change the variable value according to actual project situation.

Android 默认编译为 rk322x_box-userdebug 模式，也可在脚本中对应修改，可改为 rk322x_box-user 及其它配置：

Default building variable is rk322x_box-userdebug, it can also be modified in the script. You can edit it as rk322x_box-user or other configuration:

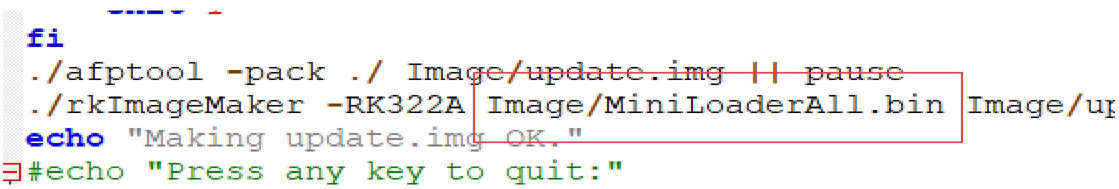
```
lunch rk322x_box-user
```

4） 指定 update.img 打包用的 loader：

Specify a loader for update.img packing

如 RKTools\linux\Linux_Pack_Firmware\rockdev\mkupdate.sh 脚本所示：

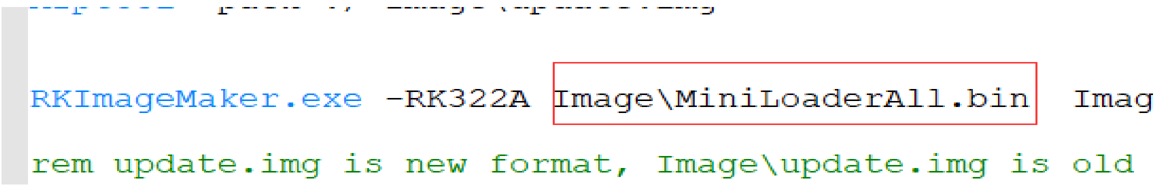
As shown of script “RKTools\linux\Linux_Pack_Firmware\rockdev\mkupdate.sh”



```
fi
./afptool -pack ./ Image/update.img || pause
./rkImageMaker -RK322A Image/MiniLoaderAll.bin Image/up
echo "Making update.img OK."
#echo "Press any key to quit:"
```

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

Windows’ packing script(RKTools\windows\AndroidTool\rockdev\mkupdate.bat) is similar, as shown below:



```
RKImageMaker.exe -RK322A Image\MiniLoaderAll.bin  Imag

rem update.img is new format, Image\update.img is old
```

update.img 打包用的 loader 被命名为 MiniLoaderAll.bin，由于 SDK 更新兼容 Loader，所以在此通过 u-boot 目录编译生成 rk322x_loader_v1.07.254.bin（拷贝时会重命名为 MiniLoaderAll.bin），需要指定脚本中 loader 文件名。

Loader for update.img packing is named “MiniLoaderAll.bin”, as SDK update is compatible to Loader, so here under the u-boot directory, build to generate rk322x_loader_v1.07.254.bin(it will be renamed as MiniLoaderAll.bin when copying), you need to specify loader file name in the script.

5） 执行自动编译脚本：

Run automatic compiling script:

```
source build_box.sh
```

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

This script will configure JDK enviroment variable automatically, build u-boot, kernel, Android and then generate images, at last pack images as update.img.

6） 脚本生成内容:

Files generated by script:

脚本会将编译生成的固件拷贝至：

IMAGE/RK3229-EVB-ANDROID-AVB_9_*****_RELEASE_TEST/IMAGES 目录下，具体路径以实际生成为准。每次编译都会新建目录保存，自动备份调试开发过程的固件版本，并存放固件版本的各类信息。

The script will copy compiled images to the directory of “IMAGE/RK3229-EVB-ANDROID-AVB_9_*****_RELEASE_TEST/IMAGES”, the fact path is subject to actual generation. Each compiling will create new directory to save images, backup image versions during debugging and developing and store all kinds of information of image versions.

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” under this directory can be directly used by Android Tool and Factory Tool for downloading and updating.

3.3 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》 under 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-Developer-Guide-UBoot-nextdev-CN.pdf》。

This chapter introduces U-Boot basic concepts and compiling attentions briefly, it helps customers understand Rockchip platform’s U-Boot framework. The details of U-Boot development can refer to “Rockchip-Developer-Guide-UBoot-nextdev-CN.pdf” under the directory of “RKDocs\common\u-boot”.

4.1 Rockchip U-Boot nextdev 简介 Rockchip U-Boot Nextdev Brief Introduction

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

next-dev is the development version that Rockchip cut out from U-Boot official version v2017.09. Currently the platform has supported RK’s all major selling chips.

目前支持的功能主要有：

Currently, major supported functions are as below:

- 支持 RK Android 平台的固件启动；Support RK Android platform’s firmware boot.
- 支持最新 Android AOSP(如 GVA)固件启动；Support latest Android AOSP(e.g. GVA)’s firmware boot.
- 支持 Linux Distro 固件启动；Support Linux Distro firmware boot.
- 支持 Rockchip miniloader 和 SPL/TPL 两种 pre-loader 引导；Support Rockchip MiniLoader and SPL/TPL two kinds of pre-loader guide.
- 支持 LVDS、EDP、MIPI、HDMI 等显示设备；Support display devices like LVDS, EDP, MIPI, HDMI and so on.
- 支持 Emmc、Nand Flash、SPI Nand flash、SPI NOR flash、SD 卡、U 盘等存储设备启动；Support booting from storage devices like eMMC, NAND Flash, SPI NAND Flash, SPI NOR Flash, SD card, U disk and so on.
- 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 boot logo display, charging animation display, low power management, power management.
- 支持 I2C、PMIC、CHARGE、GUAGE、USB、GPIO、PWM、GMAC、EMMC、NAND、中断等驱动；Support drivers for I2C, PMIC, CHARGE, GUAGE, USB, GPIO, PWM, GMAC, eMMC, NAND, Interrupt and so on.
- 支持 RockUSB 和 Google Fastboot 两种 USB gadget 烧写 EMMC；Support burning eMMC with RockUSB and Google Fastboot two kinds of USB gadget.
- 支持 Mass storage, ethernet, HID 等 USB 设备；Support USB devices like Mass storage, ethernet, HID and so on.
- 支持使用 kernel 的 dtb；Support dtb using kernel.

- 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。

U-Boot's doc directory provides rich README documents, which introduce U-Boot each function module's concepts, design idea, implementation method and so on. We suggest readers using these documents well to improve development efficiency.

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：

Platform configuration file is located in configs folder under U-Boot root directory, among which Rockchip relevant files are named starting with RK.

```
rk3288_defconfig
rk3126_defconfig
rk3128x_defconfig
rk322x_defconfig
rk3288_defconfig
rk3326_defconfig
rk322x_defconfig
rk3399_defconfig
```

RK3229 Box 开发调试选用的是 rk322x_defconfig 配置。

RK3229 Box development and debug, choose rk322x_defconfig configuration.

4.3 固件生成 Firmware Generation

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

Rockchip platform supports MiniLoader, firmware supports all kinds of storage devices. Corresponding Loader firmware is generated according to different platform's configuration. Meanwhile, after importing Arm Trusted Firmware, trust.image is generated.

以 RK322x 编译生成的镜像为例：

Take image generated by RK322X compiling as an example:

```
rk322x_loader_v1.07.254.bin
uboot.img
trust.img
```

其中 254 是发布的版本号，rockchip 定义 U-Boot loader 的版本，其中 254 是根据存储版本定义的，客户务必不要修改这个版本。

Here, 254 is a released version number, Rockchip defines U-Boot loader's version, among which 254 is defined according to storage version, customers should be sure not to modify this version.

uboot.img 是 U-Boot 作为二级 loader 的打包。

uboot.img is the package of U-Boot as a second level loader.

trust.img 是 trust firmware 的打包镜像。

trust.img is the package mirror of trust firmware.

4.4 U-Boot 编译 U-Boot Compiling

RK322x Box SDK 编译使用的是如下配置：

Rk322x Box SDK compilation uses below configuration:

```
./make.sh rk322x
```

编译完，会生成 trust.img、rk322x_loader_v1.07.254.bin、uboot.img 三个文件。

After compilation, trust.img, rk322x_loader_v1.07.254.bin, uboot.img three files are generated.

4.5 U-Boot Logo 相关的配置 U-Boot Logo Related Configuration

4.5.1 U-Boot Logo 开关配置 U-Boot Logo Swich Configuration

Sdk 默认开启 U-Boot logo 功能，以达到更快显示开机 logo 的目的，见 kernel/arch/arm/boot/dts/rk3229-evb-android.dtsi 中如下配置：

Defaultly, SDK enables U-Boot logo function, to reach faster boot logo display purpose. See below configuration in the file of “kernel/ arch/arm/boot/dts/rk3229-evb-android.dtsi”.

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

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

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

If you want to disable this function, please edit above dts file and set status = “disabled”.

4.5.2 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，重新编译内核即可。

The two pictures shown by U-Boot logo are under the kernel root directory, named logo.bmp and logo_kernel.bmp. If you want to change them, just replace them with the same named bmp picture files, and then recompile kernel.

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

Note: You don’t need to replace both two pictures, one picture is also allowed, if you only have one logo picture, you can just replace logo.bmp.

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

Note: Currently, boot logo only supports bmp format picture with size less than 8M, bit among 8bit, 16bit, 24bit and 32 bit.

4.6 U-Boot OPTEE RPMB 配置 U-Boot OPTEE RPMB Configuration

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

RPMB（Replay Protected Memory Block）Partition is a partition with security property in eMMC.

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

When the device flash uses eMMC, at present U-Boot security part will configure OPTEE to use RPMB partion by default, it is mainly used to store security relevant keys and AVB(android verified boot) relevant values.

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

Note:if eMMC material’s RPMB partition of customer device is programmed(For example, not RK keys), then it will lead starting up’s normal security code guide process to be abnormal and result in a boot failure. At this time, you need to switch to a non-RPMB approach.

可通过下面修改切换成非 rpmb 配置：

You can switch to a non-RPMB configuration according to below modification:

```
diff --git a/configs/rk322x_defconfig b/configs/rk322x_defconfig
index 6c769f9af3..81301f49cf 100644
--- a/configs/rk322x_defconfig
+++ b/configs/rk322x_defconfig
@@ -121,3 +121,4 @@ CONFIG_RK_AVB_LIBAVB_USER=y
 CONFIG_OPTEE_CLIENT=y
 CONFIG_OPTEE_V1=y
 CONFIG_TEST_ROCKCHIP=y
+CONFIG_OPTEE_ALWAYS_USE_SECURITY_PARTITION=y
```

5 内核开发常见配置 Kernel Development Common Configuration

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。RK3229 kernel 版本是 4.4，config 配置文件统一为 arch/arm/configs/ rockchip_defconfig 。RK3229 的串口波特率为 1500000，调试时请保证设置准确。

This chapter simply introduces modification of kernel's common configuration(mainly about DTS configuration), to help customers make some simple changes quickly and easily. RK3229 kernel version is 4.4 and config files are unified as arch/arm64/configs/ rockchip_defconfig. RK3229's serial port baud rate is 1500000. Please make sure the setting is accurate when you are debugging.

5.1 DTS 介绍 DTS Introduction

5.1.1 DTS 说明 DTS Introduction

RK3229 的 dts 文件在 kernel/arch/arm/boot/dts/下，如 RK3229 evb 评估板的 dts 文件为 rk3229-evb-android-avb.dts。产品的 dts 里需根据具体的产品需求配置 CPU、GPU、DDR 的频率和电压表；配置 io、wifi、bt、温控、电配置等等。

RK3229 dts file is under the dirictory of "kernel/arch/arm/boot/dts". For example RK3229 EVB board's dts file is named "rk3229-evb-android-avb.dts". Product's DTS file need to configure CPU, GPU, DDR frequency and voltage table according to the specific product requirement. Configure IO, WiFi, BT, thermal control, system power configuration etc.

请各位开发者尽量以 SDK 发布的示例产品 dts 文件做参考，进行后期的开发。

Developers please try to refer to the example product dts file in released SDK, and do further development.

5.1.2 新增一个产品 DTS Create a New Product DTS

RK3229 的产品 dts 文件需放在 kernel/arch/arm/boot/dts/下，

RK3229 product dts file needs to be located under the directory of "kernel/arch/arm/boot/dts/":

1、 以 rk3229-evb-android-avb.dts 为参照，拷贝一份 dts 文件命名为 rk3229-product.dts。
Take "rk3229-evb-android-avb.dts" as a reference, copy it and rename as rk3229-product.dts.

2、 修改 arch/arm/boot/dts/Makefile 文件，添加对应 dtb 申明
Edit file "arch/arm/boot/dts/Makefile", add corresponding dtb declaration.

```
      +rk3229-product.dtb
```

3、 修改编译脚本或编译命令。
Modify build script or build command.

4、 重新编译内核。
Recompile kernel.

5.2 WiFi&BT 的配置 WiFi&BT Configuration

RK3229 Android 9.0 平台上 WiFi、BT 可做到自动兼容，按照 RK 提供的编译 Android9.0 编译

步骤，生成固件后，默认就可以支持相应的 WiFi 模块,并且一套固件可以支持多个 WiFi 模块。目前 rk3229 android 9.0 平台 wifi、bt 模块 android 和 kernel 无需做任何配置。

On RK3229 Android9.0 Platform, WiFi and BT can be automatically compatible. According to Android9.0 compiling steps provided by RK, after firmware generated, corresponding WiFi modules are supported by default. Moreover, a suit of firmware can support multiple WiFi modules. Currently, on RK3229 Android9.0 paltform, no Android and kernel configures are required for WiFi and BT modules.

5.3 GPIO 对应关系注意 GPIO Corresponding Relationship Notification

关于原理图上的 gpio 跟 dts 里面的 gpio 的对应关系，这边有个需要注意的地方：例如 GPIO4_C0，那么对应的 dts 里面应该是“gpio4 16”。GPIO 分为 4 个端口 PORTA（0-7）、PORTB（8-15）、PORTC（16-23）、PORTD（24-31），每个 PORT 有 8 个 PIN，以此计算可得 C0 是 16，C1 口是 17，以次类推。

As for the GPIO corresponding relationship between schematic and DTS, there is something you need to pay attention to. For example GPIO4_c0, the corresponding GPIO value in dts should be “gpio4 16”. As GPIO are divided into 4 ports, PORTA(0-7), PORTB(8-15), PORTC(16-23), PORTD(24-31), every port has 8 PINS, according to these value, C0 is 16, C1 is 17, and so on.

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

GPIO usage refers to 《Rockchip-Developer-Guide-Linux-Pin-Ctrl-CN.pdf》 under the directory of “RKDocs\common\PIN-Ctrl”.

5.4 ARM、GPU、DDR 频率修改 ARM, GPU and DDR Frequency Modification

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

DVFS（Dynamic Voltage and Frequency Scaling）is a realtime voltage and frequency scaling technology. At present, modules support DVFS in kernel4.4 are CPU, GPU and DDR. CPU uses cpufreq structure, GPU and DDR use devfreq structure.

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

CPUFreq is a suit of framework model defined by kernel developer which supports dynamic CPU frequency and voltage scaling. It can reduce CPU power cosumption effectively, at the same time taking CPU performance into account.

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 chooses a proper frequency for CPU usage according to different frequency scaling strategy. Current kernel version provides below strategies:

- interactive：根据 CPU 负载动态调频调压；
  interactive：DVFS according to CPU load

- conservative：保守策略，逐级调整频率和电压；
  conservative：step by step adjust frequency and voltage
- ondemand：根据 CPU 负载动态调频调压，比 interactive 策略反应慢；
  ondemand：DVFS according to CPU load, slower than interactive strategy
- userspace：用户自己设置电压和频率，系统不会自动调整；
  userspace：Users set up voltage and frequency by themselves, system won’t adjust automatically
- powersave：功耗优先，始终将频率设置在最低值；
  powersave：power comsuption first, always set frequency to the lowest.
- performance：性能优先，始终将频率设置为最高值。
  performance：performance first, always set frequency to the highest.

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》

For detailed module function and configuration, please refer to “Rockchip-Developer-Guide-Linux4.4-CPUFreq-CN.pdf” under the directory of “RKDocs\common\DVFS”.

DEVFreq 是内核开发者定义的一套支持动态调整设备频率和电压的的框架模型。它能有效的降低该设备的功耗，同时兼顾其性能。目前我们的平台，有 GPU 和 DDR 在使用 DEVFreq。

DEVFreq is a suit of framework model defined by kernel developers which supports dynamic device frequency and voltage scaling. It can reduce the device’s power cosumption effectively, at the same time taking its performance into account. At the moment on our platform, there are GPU and DDR modules that use DEVFreq.

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

DEVFreq chooses a proper frequency for device usage according to different frequency scaling strategy. Current kernel version provides below strategies:

- Simple Ondemand ：根据负载动态调频调压；
  Simple Ondemand ：DVFS according to the load.
- Userspace：用户自己设置电压和频率，系统不会自动调整；
  Userspace：Users set up voltage and frequency by themselves, system won’t adjust automatically
- Powersave：功耗优先，始终将频率设置在最低值；
  Powersave：power comsuption first, always set frequency to the lowest.
- Performance：性能优先，始终将频率设置为最高值；
  Performance：performance first, always set frequency to the highest.
- Dmc Ondemand：我司实现的 ddr 变频策略，支持负载和场景变频；
  Dmc Ondemand：DDR frequency scaling strategy accomplished by RK, support frequency scaling according to the load and situation.

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

Defaultly GPU uses Simple Ondemand strategy, DDR uses DMC Ondemand strategy.

5.5 温控配置 Thermal Control Configuration

在 Linux 内核中，定义一套温控框架 linux Generic Thermal Sysfs Drivers，它可以通过不同的

策略控制系统的温度，目前常用的有以下几种策略：

In Linux kernel, it defines a thermal control framework called linux Generic Thermal Sysfs Drivers. It can control system’s temperature according to different strategy, At present, there are below common strategies:

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

  power_allocator：Bring in PID(Proportion-Integration-Differentiation) to control, it allocates power to each module dynamically according to the current temperature, and changes power to frequency, to achieve the effect that limiting frequency according to temperature.
- step_wise ：根据当前温度，逐级限制频率。

  step_wise：step by step limit frequency according to the current temperature
- userspace：不限制频率

  userspace：no limit to frequency

详细的模块功能及配置，请参考 RKDocs\common\Thermal 目录下《Rockchip-Developer-Guide-Linux4.4-Thermal-CN.pdf》

For detailed module function and configuration, please refer to “Rockchip-Developer-Guide-Linux4.4-Thermal-CN.pdf” under the directory of “RKDocs\common\Thermal”.

5.6 PWM IR 配置 PWM IR Configuration

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

Radiating circuit of infrared remote control emits modulated infrared light waves using infrared light emitting diode. Infrared receiving circuit is made of infrared receiving diode, audion, or silicon solar cell. They change infrared light radiated by infrared transmitter into corresponding electricity signal, then resend the signal to post amplifier. Considering household electrical appliances’ diversity and users’ handling characteristics, the manufacturers make some strict coding stardard. These codings are different and form different coding schemes, they are uniformly called “Infrared remote control coding transmission protocol”. At present, RK platform only supports infrared protocol of NEC coding.

RK3229 平台详细的遥控器适配，键值添加，红外按键定义，及遥控器功能相关调试内容请参考 RKDocs\common\PWM 目录下《Rockchip_Developer_Guide_PWM_IR_CN.pdf》。

RK3229 platform detailed remote control adaptation, key value addition, infrared button definition and remote control function related debugging, please refer to “Rockchip_Developer_Guide_PWM_IR_CN.pdf” under the directory of “RKDocs\common\PWM”.

5.7 DDR 频率修改说明 DDR Frequency Modification Description

请参考 RKDocs\common\DDR 目录下《Rockchip-Developer-Guide-DDR-CN.pdf》如何修改 ddr 频率章节。

Please refer to the chapter how to change DDR frequency, which is in the document “Rockchip-

Developer-Guide-DDR-CN.pdf" under the directory of "RKDocs\common\DDR".

6 Android 开发常见配置 Android Development Common Configuration

本节简单介绍 Android 开发中一些常见配置的修改，RK3229 平台搭载的是最新的 Android9.0 系统。

This chapter simply introduces common configuration modifications during Android development, RK3229 platform carries the latest Android9.0 system.

6.1 Android 编译配置 Android Build Configuration

6.1.1 lunch 选项说明 Lunch Option Description

rk322x_box-userdebug：rk3229 平台 box 产品 userdebug（32 位）

rk322x_box-userdebug：RK3229 platform box product userdebug(32 bit)

rk322x_box-user：rk3229 平台 box 产品 user（32 位）

rk322x_box-user：RK3229 platform box product user(32 bit)

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

User version enables selinux permission authentication. APK and JAR updates are involved during development process, log printing and debugging are troublesome.

建议开发调试阶段默认选择 userdebug 编译。

You are suggested to choose userdebug for compiling during development and debugging stage.

6.1.2 添加一个新的产品 Add a New Product

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

The developers may have the requirement that different products use a same chip to develop, a suit of SDK needs to be compiled to generate multiple product firmwares at the same time.

RK3229 平台支持 Box 类型各种产品形态，当需要添加一个新的产品时，可以基于已有的 rk3229_box 来建立，如下以建立一个新的平板产品为例进行说明，具体步骤为：

RK3229 platform supports variety of Box type products, when you need to add a new product, you can build it based on existing RK3229_box. Below take a new tablet product as an example to introduce. Here is the detailed steps:

1） 产品命名规则：
Product Naming Rule

Box 产品名中需带有“box”字样；请务必遵守以上规则，否则系统会异常。

The word “box” is required for Box product name. Please be sure to comply with above rule, otherwise system will be abnormal.

2） 新增文件夹 device/rockchip/rk322x/rk322x_box_000，基于 rk322x_box.mk 创建 rk322x_box_000.mk，将 rk322x_box 目录下的所有文件拷贝至 rk322x_box_000 目录下。

Create a new folder “device/rockchip/rk322x/rk322x_box_000”, then create rk322x_box_000.mk based on rk322x_box.mk, copy all files under rk322x_box directory to rk322x_box_000 directory.

```
   cd device/rockchip/rk322x
   mkdir rk322x_box _000
   cp rk322x_box.mk ./ rk322x_box _000.mk
   cp rk322x_box/* rk322x_ box _000/
```

3） 在 device/rockchip/rk322x/ AndroidProducts.mk 中添加：

Add below to the file “device/rockchip/rk322x/ AndroidProducts.mk”

```
   PRODUCT_MAKEFILES := \
       $(LOCAL_DIR)/rk322x.mk \
       $(LOCAL_DIR)/rk322x_box.mk \
       $(LOCAL_DIR)/rk322x_box_000.mk \
```

4） 在 vendorsetup.sh 中添加产品对应的 lunch 选项：

Add product corresponding lunch option to the file vendorsetup.sh:

```
   add_lunch_combo rk322x_box-eng
   add_lunch_combo rk322x_box-userdebug
   add_lunch_combo rk322x_box-user
   add_lunch_combo rk322x_box_000-userdebug
   add_lunch_combo rk322x_box_000-user
```

5） 修改 rk322x_box_000.mk 及 rk322x_box_000 目录下的新产品所需要修改的配置。

Modify the required configuration of rk322x_box_000.mk and the new products under rk322x_box_000 directory

6） 修改编译脚本或编译命令，重新 lunch 产品名称进行新产品编译。

Modify the compiling script or compiling command, re-lunch the product name to recompile the new product.

6.2 预置 APK Pre-installed APK

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

Application pre-install function on Android system mainly refers to install the pre-prepared third-party application into the Android system according to manufacturer’s requirements when configuring the product.

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

Preinstalled APK applications need to get authorization from relevant manufacturers. If developers and customer manufacturers preintall unauthorized applications in private and thus need to take legal responsibility, RK is not resonsible for it.

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：

Preinstallation function involves uninstalled preinstallation and non uninstalled preinstallation, this document mainly describes uninstalled preinstallation function. Below is the configuration steps:

1）若是希望可卸载预安装，新增文件夹 device/rockchip/rk322x/rk322x_box/preinstall_del；若是不可卸载原装，新增文件夹 device/rockchip/rk322x/rk322x_box/preinstall。

For uninstalled preinstallation, create folder “device/rockchip/rk322x/rk322x_box/preinstall_del”, for non uninstalled preinstallation, create folder “device/rockchip/rk322x/rk322x_box/preinstall”

2）拷贝需要预制的第三方应用到上述文件夹，注意 apk 文件名尽量使用英文，避免空格。

Copy third-party applications which need to be preinstalled to the above folder, pay attention that

APK file names use English as possible as you can, and avoid blank space.

3）编译结束后会将预制的文件拷贝至 system 固件中。烧录后，系统会自动安装这些应用到 data/app 目录。

After compiliation, preinstalled files will be copied to system image. After flashing, the system will install these applications to ‘data/app’ directory automatically

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

Attention: Applications under preinstall directory, even though that users uninstall them during usage, they will be installed automatically when the device performs factory reset. If you hope that the preinstalled applications won’t be reinstalled after factory reset, you can modify the above folder name as preinstall_del_forever

6.3 开/关机动画 Power on/off Animation

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

Neet to configure BOOT_SHUTDOWN_ANIMATION_RINGING := true in the file “device/rockchip/common/BoardConfig.mk”, and prepare below corresponding resource files, after compilation, corresponding resource files will be copied to corresponding out directory.

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

Copy power on animation file 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 power off animation file to “device/rockchip/common/shutdownanimation.zip” (source code path)

6.4 Parameter 说明 Parameter Instruction

请参考 device/rockchip/rk322x/rk322x_box 目录下 parameter.txt 文件来相应修改配置，关于 parameter 中各个参数、分区情况细节，请参考\RKDocs\common\RKTools manuals 目录下的《Rockchip_Introduction_Parameter_File_Format_CN.pdf》文档。

Please refer to parameter.txt file under directory of “device/rockchip/rk322x/rk322x_box” to do relevant configuration modification. For each parameter and partition situation detailed information of parameter file, please refer to the document “Rockchip_Introduction_Parameter_File_Format_CN.pdf” under the directory of “\RKDocs\common\RKTools manuals”.

6.5 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.

6.6 显示框架配置 Display Framework Configuration

请参考 RKDocs\android\Rockchip_Introduction_Android8.1_BOX_Display_Framework_Configuration_CN.pdf

Please refer to RKDocs\android\Rockchip_Introduction_Android8.1_BOX_Display_Framework_Configuration_CN.pdf

6.7 OTA 升级 OTA Upgrade

6.7.1 OTA 介绍 OTA Introduction

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。

OTA (over the air) upgrade is the standard software upgrade method provided by Android system. It's powerful. It provides complete upgrade (full package) and incremental upgrade mode (difference package). You can upgrade locally or over the network.

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》。

For the detailed OTA upgrade and Recovery module function and configuration, please refer to "Rockchip_User_Guide_Recovery_CN&EN.pdf" under the directory of "RKDocs\android".

6.7.2 生成完整包 Full Package Generated

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

Full package involves boot.img, uboot.img, vbmeta.img and upgrade patch of system, vendor and oem.

发布一个固件正确的顺序：

Correct steps to release a firmware:

1、make -j4

2、make otapackage -j4

3、./mkimage.sh

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

OTA full package rkxxxx-ota-eng.root.zip will be generated under directory of "out/target/product/rkxxxx/", rename it as update.zip and then copy it to T card or built-in flash to upgrade.

6.7.3 生成差异包 Difference Package Generated

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

OTA difference package only involves difference content, the package size is small, it is mainly used for OTA on line upgrade, also for T card upgrade locally. OTA difference package generating needs special compiling and operation manually.

1、 首先发布 v1 版本的固件，生成 v1 版本的完整包。

First release firmware of v1 version, generate full package of v1 version.

2、 保存 out/target/product/rkxxxx/obj/PACKAGING/target_files_intermediates/rk3188-target_files-eng.root.zip 为 rkxxxx-target_files-v1.zip，作为 v1 版本的基础素材包。

Save “out/target/product/rkxxxx/obj/PACKAGING/target_files_intermediates/rk3188-target_files-eng.root.zip” as “rkxxxx-target_files-v1.zip”, to be the basic material package of v1 version.

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

Modify kernel code or Android code, release v2 version firmware, generate full package of v2 version.

4、保存 out/target/product/rkxxxx/obj/PACKAGING/target_files_intermediates/rk3188-target_files-eng.root.zip 为 rkxxxx-target_files-v2.zip，作为 v2 版本的基础素材包。

Save “out/target/product/rkxxxx/obj/PACKAGING/target_files_intermediates/rk3188-target_files-eng.root.zip” as “rkxxxx-target_files-v2.zip”, to be the basic material package of v2 version.

5、生成 v1-v2 的差异升级包：

Generate difference upgrade package of v1-v2:

```
   ./build/tools/releasetools/ota_from_target_files  --block -v  -i
rkxxxx-target_files-v1.zip  -p  out/host/linux-x86  -k
build/target/product/security/testkey  rkxxxx-target_files-v2.zip
out/target/product/rk322x/rkxxxx-v1-v2.zip
```

说明:生成差异包命令格式:

Instruction: command format of generating difference package:

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

ota_from_target_files
--block
-v -i “the prior target file used for comparing”
-p “Host computer compiling enviroment”
-k “package key”
“later target file used for comparing”
“OTA difference package generated last”

6.8 预制 Demo Precast Demo

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

During development and model machine preparation, most developers and manufacturers have the requirement to integrate audio and video resourse, picture resourse and so on. Precast demo resource function is attached with this SDK. For detailed information, please see charter 8.8 OemTool 打包工具.

6.9 开机视频 Boot Video

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

Need to configure BOOT_VIDEO_ENABLE ?= true in the file “device/rockchip/common/BoardConfig.mk” of product, also prepare below corresponding boot video file bootanimation.ts(Defaultly in the code, video files with .ts suffix are recognized. For other fomat files, developers can just rename the file and suffix as bootanimation.ts. There is no need to change format). After the compiling is completed, the corresponding resourse files will be copied to the corresponding out directory.

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

Copy boot video file to “device/rockchip/common/bootvideo/bootanimation.ts”(source code path)

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

Note:when the boot video function is enabled, defaultly the video is configured to be played completely, you can control playing time through attribute persist.sys.bootvideo.showtime.

-1：代表没设置时长，按照开机自然阶段时间展示；

-1: the value indicates that play time is not set up, displaying according to normal boot time.

-2：代表要将视频播完才能进入 launcher；

-2: the value indicates that boot video needs to be played completely, then the system can run into launcher.

配置其它大于 0 的数字表示具体要播放的，超过 120 秒按 120 秒播放。

Other value above 0 indicates the actual playing time, the maximum playing time is 120 seconds, above 120 seconds is regarded as 120 seconds too.

6.10 低内存机器内存优化配置 Low Memory Machine Memory Optimization Configuration

需要在产品的 device/rockchip/rk322x/rk322x_box/BoardConfig.mk 中添加配置

Need to add configuration in the product file “device/rockchip/rk322x/rk322x_box/BoardConfig.mk”.

```
BUILD_WITH_GO_OPT:=true
```

重新编译系统生效。

Recompile the system to make effect.

主要优化内容：

the main optimization content:

1.系统开启 Android Go 内存优化成果；

the system opens Andriod Go memory optimization result.

2.视频库播放内存占用及缓存大小优化;

Optimization for video library memory usage and buffer size.

3.lowmemorykiller 水线及策略调整;

Lowmemorykiller water line and strategy adjustment.

目前 SDK 默认未开启。

Currently this function is not enabled in the SDK defaultly.

6.11 DRM Widevine Level1 配置 DRM Widevine Level1 Configuration

按如下修改打开配置编译，注意需要烧写 L1 key 才能生效。

Open the configuration compilation according to the below modification, pay attention that it will make effect only after burning L1 key.

```
   device/rockchip/rk322x$ git diff .
   diff --git a/rk322x_box/BoardConfig.mk b/rk322x_box/BoardConfig.mk
   index 1425e95..3978741 100755
   --- a/rk322x_box/BoardConfig.mk
   +++ b/rk322x_box/BoardConfig.mk
   @@ -95,7 +95,7 @@ BUILD_WITH_GTVS := false
    BUILD_WITH_GOOGLE_FRP := false

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

    #for microsoft drm
    BUILD_WITH_MICROSOFT_PLAYREADY :=true
```

注意：开启 L1 功能时，播放需要支持 SVP 的视频，目前需要在内核中将安全内存单独划分出来，如 322x 样机板开启 svp 内存 256M 配置方法：

Note: If you play video suppoting SVP when L1 function is enabled, at present, you need to separate security memory alone in the kernel. For example, the configuration method of 322x model machine opening 256M SVP memory:

```
   kernel$ git diff
   diff --git a/arch/arm/boot/dts/rk3229-evb-android.dtsi
b/arch/arm/boot/dts/rk3229-evb-android.dtsi
   index 1f8d999cfb71..267f13b7e4fe 100644
   --- a/arch/arm/boot/dts/rk3229-evb-android.dtsi
   +++ b/arch/arm/boot/dts/rk3229-evb-android.dtsi
   @@ -265,7 +265,7 @@
            * enable like this:
            * reg = <0x80000000 0x10000000>;
            */
   -       reg = <0x80000000 0x0>;
   +       reg = <0x80000000 0x10000000>;
   };
```

6.12 媒体中心 Media Center

1.新增 4K 图片展示功能，可在对应产品目录下 device.mk 配置（例如：rk3229_box 对应 device/rockchip/rk322x/device.mk）.

Add 4K picture displaying function, it can be configured in the file device.mk under corresponding product directory(for example:rk3229_box matches device/rockchip/rk322x/device.mk).

persist.media.4k 属性控制默认状态。当前默认配置为 false。

persist.media.4k attribute is used to configure default state. Currently the default configuration value is false.

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

persist.media.4k: true indicates media center 4K picture, displaying original picture size in default

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

persist.media.4k: false indicates media center 4K picture, displaying picture according to size adaptive adjustment.

2.新增 heic 动态图片支持，对 heic 动态图片可长按“center”键，显示动态效果。

Add heic dynamic picture support, you can long press the “center” button to display dynamic effect for the heic dynamic picture

6.13 TWRP Recovery

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

Team Win Recovery Project (TWRP) is a customized recovery mode image, of which software is open source. It’s used for devices based on Android. It provides a kind of UI that suports mouse operation, allowing users to install firmwares and backup current system to the third party.

可根据 https://github.com/rockchip-software/TWRP 编译对应的 recovery.img, 并通过 RKTools\windows\AndroidTool_Release_v2.65 工具烧写。

You can build corresponding recovery.img according to https://github.com/rockchip-software/TWRP, and flash it through “RKTools\windows\AndroidTool_Release_v2.65” tool

6.14 Support Magisk

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

Magisk is a suit of open source tool for customizing Android, supporting devices whose system is above Android 5.0(API 21). It covers the basic parts of Android customization:root, boot scripts, SELinux patches, AVB2.0/dm-verity/forceencrypt removals and so on.

可根据 RKDocs/android/Rockchip_User_Guide_Magisk_Installation_EN.pdf 在 Rockchip 平台安装 Magisk。

You can install Magisk on Rockchip platform according to the document “RKDocs/android/Rockchip_User_Guide_Magisk_Installation_EN.pdf”.

6.15 安全启动方案 Security Boot Solution

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

Rockchip security boot solution is based on hardware protection mechanism provided by RK chip. It provides reliable security protection for Box's boot loader, uboot, trust image and Android system(boot(including kernel), recovery, system, vendor, oem images and so on). For Box products, it's used to protect Box system security, and prevent Box from being flashed or business related application from being tampered.

可参考文档 RKDocs/common/security/Rockchip-Secure-Boot-Application-Note-V1.9.pdf

You can refer to the document "RKDocs/common/security/Rockchip-Secure-Boot-Application-Note-V1.9.pdf"

6.16 Microsoft PlayReady

PlayReady 为微软公司的新的 DRM 系统，它是 WMDRM(Windows Media DRM)的升级产品，可以为数字媒体提供内容保护支持。微软 PlayReady 官方网站为 http://www.microsoft.com/playready/，可以从该网站获取 PlayReady 产品、技术、文档、License 及支持等信息；PlayReady 官方测试网站为 http://test.playready.microsoft.com，可以使用该网站的测试用例和码流对用户的 PlayReady 产品进行基本功能测试。

PlayReady is the new DRM system of Microsoft company, it is the upgrade product of WMDRM(Windows Media DRM). It can provide content protection support for digital media. Microsoft PlayReady official website is: http://www.microsoft.com/playready/, you can get PlayReady product, technology, document, License and support information through it. PlayReady officail test website is: http://test.playready.microsoft.com, you can do basic function test for the users' PlayReady products using this website's test cases and code stream.

Rockchip PlayReady 方案分为 SW 版本和 HW 版本，根据微软证书定义的 Security Level，我们通常也把 SW 版本称为 SL2000 版本，HW 版本称为 SL3000 版本。SW 版本能为内容和证书提供基本和必要的保护，对芯片和方案没有特别要求，适用于内容提供商没有特殊要求的场景。HW 版本使用 Rockchip 芯片的 Trustzone 硬件保护机制，对证书、加解密密钥以及解密后的码流提供更高级别的保护，HW 版本适用于内容提供商明确要求 Trustzone 特性或 Secure Video Path 的场景。

Rockchip PlayReady solution is divided into SW version and HW version, According to Security Level defined by Microsoft certificate, we usually call SW version as SL2000 version, HW version as SL3000 version. SW version can provide basic and necessary protection for content and certificate, it has no special requests for chip and solution, applies to the scenarios where content providers do not have special requirements. HW version uses Rockchip chip's Trustzone hardware protection mechanism, it provides higer level protection to certificate, encryption and decryption key and decoded code stream, HW version applies to the scenarios where content providers have clear requirements about Trustzone features or Secure Video Path.

在 rockkchip 平台使用 playready 功能，需先向微软申请 license，然后向 rockchip 申请相关补丁。

Using PlayReady function on Rockchip platform, you need to apply for license from Microsoft first, then apply for corresponding patches from Rockchip.

6.17 动态加载 UiMode Loading UIMode Dynamically

电视盒子系统中，有部分移动端 App 无法正常显示。因为全局的 UIMode 是 Configuration.UI_MODE_TYPE_TELEVISION，这部分 App 不支持 Tv 端的显示。现在我们提供了一种方式，通过白名单配置来动态的为 App 加载 UIMode（源码路径：device/rockchip/common/uimode/uimode_app.xml,设备路径：vendor/etc/uimode_app.xml）。

There are a part of mobile APPs which can’t display normally in tv box system. Since the global UIMode is Configuration.UI_MODE_TYPE_TELEVISION, these mobile apps are not suppoted to display on TV box. Now we provide a way to load UIMode for APPs dynamically through whitelist configuration(source code path: device/rockchip/common/uimode/uimode_app.xml, device path: vendor/etc/uimode_app.xml).

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

Currently, this way only supports box devices, usage method please refer to “device/rockchip/common/uimode/ReadME.md”.

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 during SDK development. The content will be updated and improved 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. Its 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 simulators or devices
- 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 apks to the simulator or the 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 Box 机器的方式不同，ADB 可以分为两类：

ADB is a “client side－server side” program. Usually the client side mainly refers PC and the server side refers Android actual devices or virtual machines. The ADB can be divided into two categories according to the way PC connects to the box devices:

- 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 the STB device through cable/wireless network(the same LAN)
- USB ADB：主机通过 USB 线连接到 STB 设备
  USB ADB: PC connects to the 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 using 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
  Only support that Host computer connects to one device, multiple devices are not supported.

连接步骤如下：

The connection steps are as below:

1、 Box 机器已经运行 Android 系统，设置->开发者选项->已连接到计算机 打开，usb 调试开关打开。

The box device already runs Android system. Choose setting -> developer option -> connect to the computer enabled, 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 Box 机器相连。

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

```
adb shell
```

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

Run the command “adb devices” to see whether the connection is successful. If the device serial number displays, it indicates that the connection is successful.

7.1.3 网络 adb 使用要求 Network ADB Usage Requirements

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, from adb v1.0.25, the function of debugging Android devices through tcp/ip is added.

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, it must meet below conditions:

1．设备上面首先要有网口，或者通过 WiFi 连接网络。

First, the device must have a network port, or connect to a Wi-Fi network.

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 connected to LAN and the device has a LAN IP address.

3．要确保研发机和设备能够相互 ping 得通。

Make sure that 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。

Make sure that Android device’s adbd process(adb’s backaground process) has 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 默认未开启网络 adb，需要手动在开发者选项中打开。

SDK doesn’t enalbe network ADB function by default. Need to manually enable it in the developer option.

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 that 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)。

The Android device should boot first, if possible, make sure that adbd has started up(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 relevant 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 connection, you can enter ADB relevant commands to debug in PC. For example, adb shell, it will connect to the device through TCP/IP. It is the same with 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, enter below command in PC to disconnect the connection:

```
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 hasn’t added ADB port number configuration, or you just want to change ADB port number, you can modify as follow:

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。

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

2．设置 adb 监听端口：

Set up 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

Restart adbd

```
#kill -9<pid>，这个 pid 就是上一步找到那个 pid
#kill -9<pid>, this pid is the one which is found in above step
```

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, Android init process will restart adbd automatically. After adbd is restarted, and it finds that service.adb.tcp.port is set, then it will automatically monitor network request instead.

7.1.7 ADB 常用命令详解 ADB Commonly Used Command Elaboration

（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 status of Android device or simulator connected to the 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 value is the serial number or IP and port number, status of the Android device.

（2）安装 apk Install APK

将指定的 apk 文件安装到设备上：

Install a specific APK file to the device:

```
adb install <apk 文件路径>
adb install <apk file path>
```

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>
adb install -r <apk file path>
```

示例如下：

For example:

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

（3）卸载 apk Uninstall APK

完全卸载：

Completely 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。

Example Description: remove WishTV.apk file under 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：

Enter 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 the local computer to the device. Generally speaking, local path indicates the local computer and remote path indicates the single board device connected with ADB.

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

Example Description: upload the local WishTV.apk file to the directory of “system/app” in Android system.

（7）从设备下载文件到电脑 Download File from the Device to the 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 the local computer.

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

Example Description: download the file or folder from the directory of “system/app” in 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’s System Information

在 adb shell 下查看设备系统信息的具体命令。

The specific command 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 log system provides the function to record and check the system debugging information. The logs are all recorded from various softwares and some system buffer zones. The buffer zone can be checked and used through Logcat function. Logcat is the most common used function for program debugging. The function displays the program running status mainly by printing logs. As the amount of logs to print 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 system log buffer zone:

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 Common Used Log Filtering Method

控制日志输出的几种方式：

Several ways to control log output:

- 控制日志输出优先级。

  Control the priority of log output

  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。

  Example description:display the log information with warning or higher priority

- 控制日志标签和输出优先级。

  Control the log 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。

Example description:support all log 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 logs with 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。

Example description:only output logs with WishTV label

- 只输出指定优先级和标签的日志

  Only output logs with 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。

Example description:only output logs with priority I and label WishTV.

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 along with Android, running in the shell environment of the device side. It is used to output 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：

Including 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）

  VSS：Virtual Set Size. Virtual consumed memory size(including memory occupied 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）

RSS： Resident Set Size. actual consumed physical memory size(including memory occupied 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）

  PSS：Proportional Set Size. actual consumed physical memory size(allocate the memory occupied 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）

  USS：Unique Set Size. physical memory size occupied by process alone(not including the memory occupied by the shared lib)

> 注意：
>
> Attention:
>
> ● 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 recycled 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 a 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 occupancy status of the shared memory area by a single process after the shared memory is divided in proportion.

7.3.1 使用 procrank Using Procrank

执行 procrank，前需要先让终端获取到 root 权限

Make sure that the terminal gets root permission 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 ascending[descending] sort

- -w：只显示 working set 的统计计数 only display the statistical counting of working set

- -W：重置 working set 的统计计数 reset the statistical counting 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 by VSS descending sort:

```
procrank －v
```

默认 procrank 输出是通过 PSS 排序。

Defaultly, single Procrank command outputs the memory snapshot by PSS sort.

7.3.2 检索指定内容信息 Retrieve the Specified 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 check the memory occupancy status of a specified 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 indicates the target application name, PID indicates the target application process.

输出 systemUI 进程的内存占用状态:

Output the memory occupancy status of 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。

By Tracing the memory occupied status, you can analyze whether there is memory leakage in the process. By writing a script to output the process memory snapshot continuously, and then comparing USS segment , you can be aware of whether 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 memory occupancy status of an application whose process name is “com.android.systemui” and then see whether there is leakage.

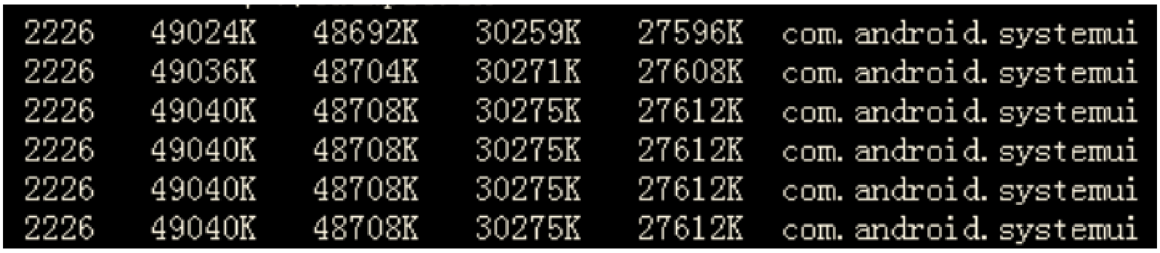
1．编写脚本 test.sh

Write a script named 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 PC is connected to the target device by ADB tool, run the script use command ”./test.sh”. The result is shown below.



```
2226   49024K   48692K   30259K   27596K   com.android.systemui
2226   49036K   48704K   30271K   27608K   com.android.systemui
2226   49040K   48708K   30275K   27612K   com.android.systemui
2226   49040K   48708K   30275K   27612K   com.android.systemui
2226   49040K   48708K   30275K   27612K   com.android.systemui
2226   49040K   48708K   30275K   27612K   com.android.systemui
```

Figure 7-1 跟踪进程内存状态 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 along with Android system. It runs in the shell environment of the device side, providing status information of a running service in the system. A running service indicates the server side process of Android binder mechanism.

dumpsys 输出打印的条件：

The conditions for dumpsys to output the printing:

1．只能打印已经加载到 ServiceManager 中的服务；

You can only print services that have been loaded to the 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 server-side code is not implemented, there will be none 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 dumpsys help information

```
dumpsys  -help
```

- 查看 Dumpsys 包含服务列表

  View the service list included by 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, developers can pay attention to the service names need to debug.

```
dumpsys  -l
```

- 输出指定服务的信息

  Output the specified service information

  作用：输出指定的服务的 dump 信息。

  Function: output the specified service’s 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
```

- 输出指定服务和应用进程的信息

  Output information of specified service and application process

  作用：输出指定服务指定应用进程信息。

  Function:Output information of specified service and application process

  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: Output memory information of service named meminfo and process named com.android.systemui. Execute below command:

```
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 you must input full service name.

8 常用工具说明 Common Tool Instructions

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

This chapter simply describes instructions of some developing and MP tools along with SDK, to help developers be familiar with RK platform tool usage. The detailed tool instructions please refer to tool related documents under 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 stress test for the vasious functions of the target devices to make sure that the whole system is running stably. SDK can start StressTest application and conduct stress test for the various functions by opening the calculator and entering “83991906=” secret code.

Stresstest 测试工具测试的内容主要包括：

The test items of the 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 (ping test and iperf test will be added in the future)

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/Wake stress test
- 视频拷机测试。Video 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）、USB HOST、按键（KEY），喇叭耳机（Codec）测试项目。

PCBA test tool is used to help quickly identify whether a product is good or not during production to improve the production efficiency. Current test items include panel(LCD), wireless(WiFi), 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 items and manual test items. 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 RKDocs\android\Rockchip_Developer_Guide_PCBA_Test_Tool_CN.pdf。

For detailed PCBA function configuration and usage, pleaes refer to: \RKDocs\android\Rockchip_Developer_Guide_PCBA_Test_Tool_CN.pdf

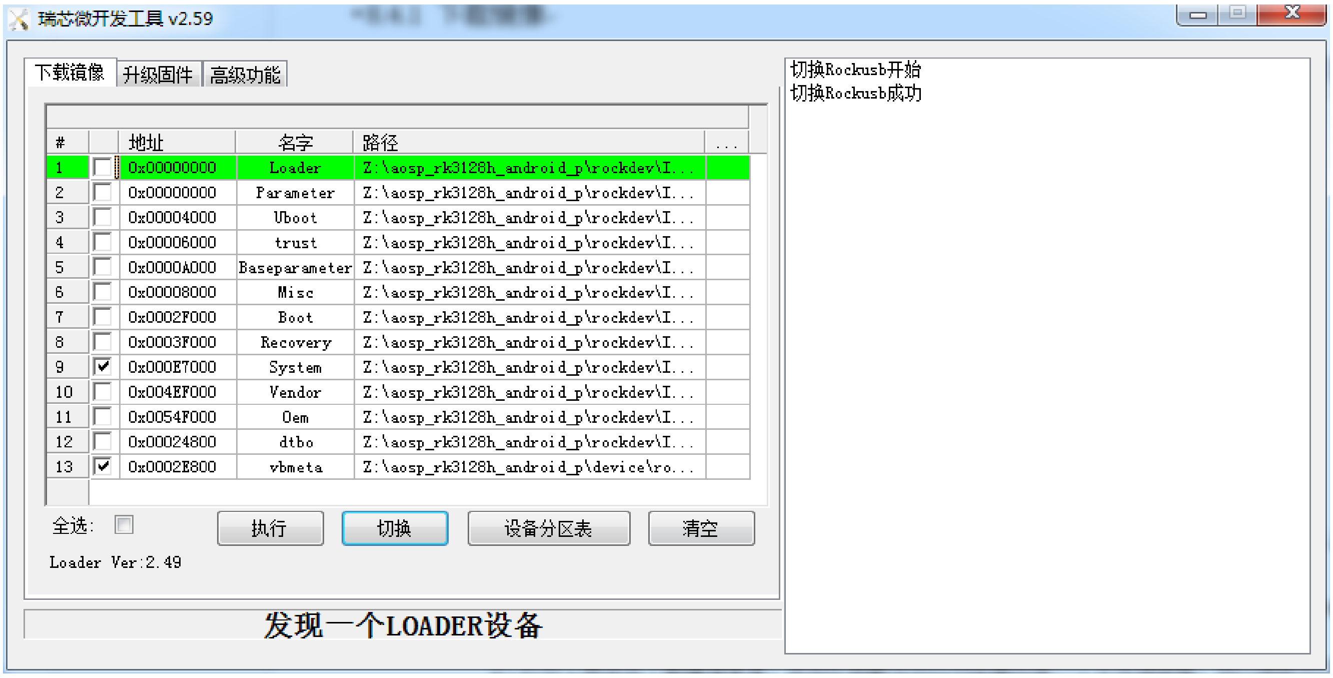
8.3 DDR 测试工具 DDR Test Tool

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。RK322x DDR 测试工具还未发布，后续会随 SDK 更新。

Use DDR test tool to do the stability test on the target devices to make sure DDR function normal and stable. Currently DDR test tool of this platform is not released yet, and it will be updated along with SDK later

8.4 Android 开发工具 Android Development Tool

8.4.1 下载镜像 Download the Mirror Image



瑞芯微开发工具 v2.59

#		地址	名字	路径	...
1	□	0x00000000	Loader	Z:\aosp_rk3128h_android_p\rockdev\I...	
2	□	0x00000000	Parameter	Z:\aosp_rk3128h_android_p\rockdev\I...	
3	□	0x00004000	Uboot	Z:\aosp_rk3128h_android_p\rockdev\I...	
4	□	0x00006000	trust	Z:\aosp_rk3128h_android_p\rockdev\I...	
5	□	0x0000A000	Baseparameter	Z:\aosp_rk3128h_android_p\rockdev\I...	
6	□	0x00008000	Misc	Z:\aosp_rk3128h_android_p\rockdev\I...	
7	□	0x0002F000	Boot	Z:\aosp_rk3128h_android_p\rockdev\I...	
8	□	0x0003F000	Recovery	Z:\aosp_rk3128h_android_p\rockdev\I...	
9	☑	0x000E7000	System	Z:\aosp_rk3128h_android_p\rockdev\I...	
10	□	0x004EF000	Vendor	Z:\aosp_rk3128h_android_p\rockdev\I...	
11	□	0x0054F000	Oem	Z:\aosp_rk3128h_android_p\rockdev\I...	
12	□	0x00024800	dtbo	Z:\aosp_rk3128h_android_p\rockdev\I...	
13	☑	0x0002E800	vbmeta	Z:\aosp_rk3128h_android_p\device\ro...	

全选: □

Loader Ver:2.49

发现一个LOADER设备

切换Rockusb开始
切换Rockusb成功

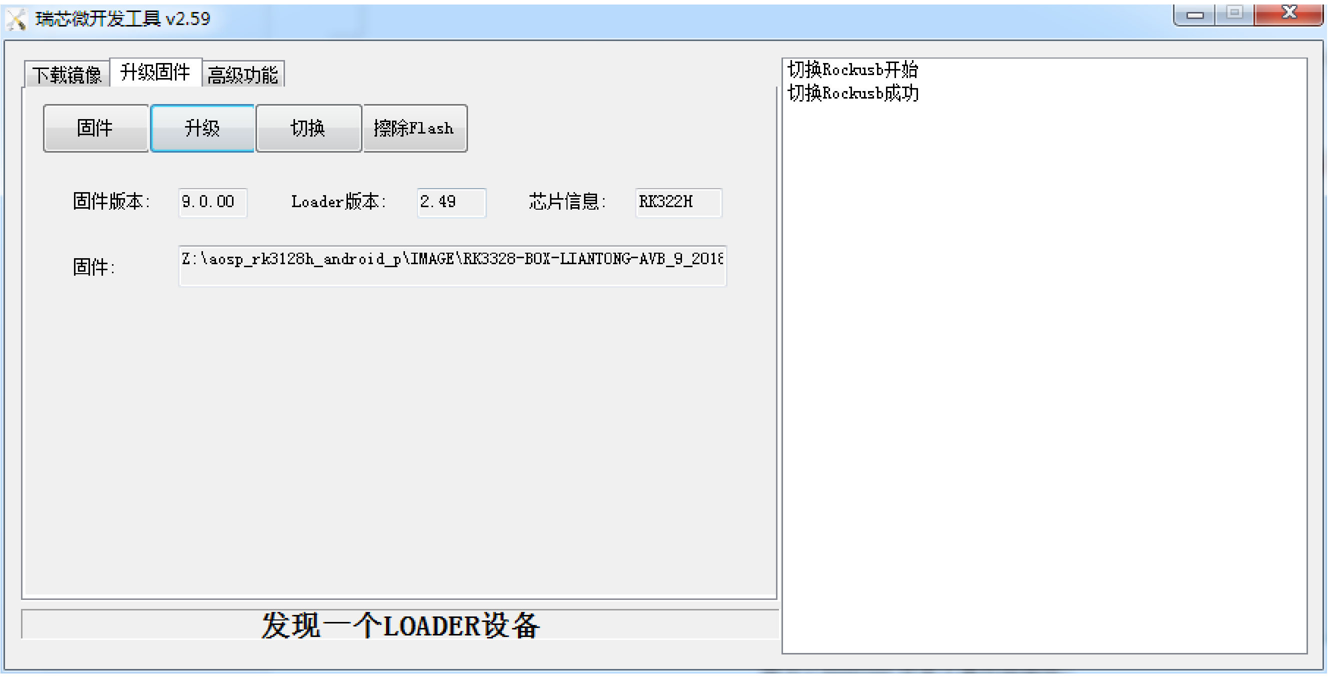
Figure 8-1 Android 开发工具下载镜像 Use Android Development Tool to Download the Mirror Image

1) 连接开发板进入下载模式（下载模式先按住开发板 reset 按键，再长按 recovery 按键约 3-4s 时间进入 loader 模式）。
   Connect PC to the development board, and make the development board enter the download mode.(Download mode: Firstly press reset key of the development board, and then long press recovery key about 3-4 seconds to enter loader mode.)
2) 打开工具点击下载镜像菜单，点击红色箭头对应列会跳出来一个文件选择框，可以选择对应分区的 img 本地地址，其他几项依次配置。
   Open the tool, and click “download image” menu. Single click the column with red arrow in every line end, it will pop out a file selection box and then choose the img file local path of the corresponding partition. Set all the other img file paths in turn.
3) 配置完成后，点击执行就可以看到右边空白框进入下载提示。
   After configuration, click “execute’. The right information box will display the related downloading prompt information.

其中 “低格”按钮是用来擦除设备的，“清空”按钮是清空编辑框文本。

Here “低格” button: Used to erase the device, “清空” button: Used to clean up the information box.

8.4.2 升级固件 Upgrade Image



固件版本: 9.0.00

Loader版本: 2.49

芯片信息: RK322H

固件: Z:\aosp_rk3128h_android_p\IMAGE\RK3328-BOX-LIANTONG-AVB_9_2018

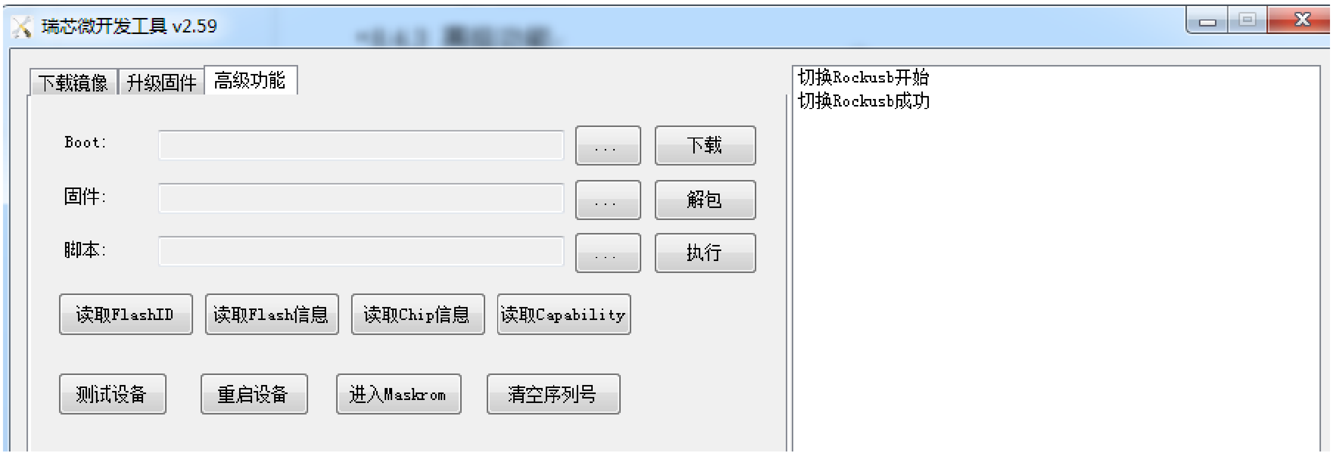
```
切换Rockusb开始
切换Rockusb成功
```

发现一个LOADER设备

Figure 8-2 Android 开发工具升级固件 Use Android Development Tool to Upgrade Image

1) 进行打包固件。
   Prepare the target image (refer to update.img package)
2) 点击固件选择刚打包好的 update.img 文件，并点击升级按钮进行下载。(注意设备必须在下载模式下）。
   Click ”固件” button, and choose the target image file update.img. Click “升级” button to download.(Note: You need to confirm that the device is already in the download mode.)

8.4.3 高级功能 Senior Functions



切换Rockusb开始
切换Rockusb成功

Figure 8-3 Android 开发工具高级功能 Android Development Tool Senior Functions

Boot 只能选择打包好的 update.img 文件或是 loader 的文件;

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

固件必须现使用打包后的 update.img;

固件: You must use the packed update.img.

解包功能可将 update.img 拆解为各部分镜像文件。

解包：The unpack function can unpack update.img into each part mirror files.

8.5 update.img 打包 Update.img Package

RK322x 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：

RK322x platform supports to pack the scattered mirror files into one complete update.img file to benefit production flashing and upgrade. 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。

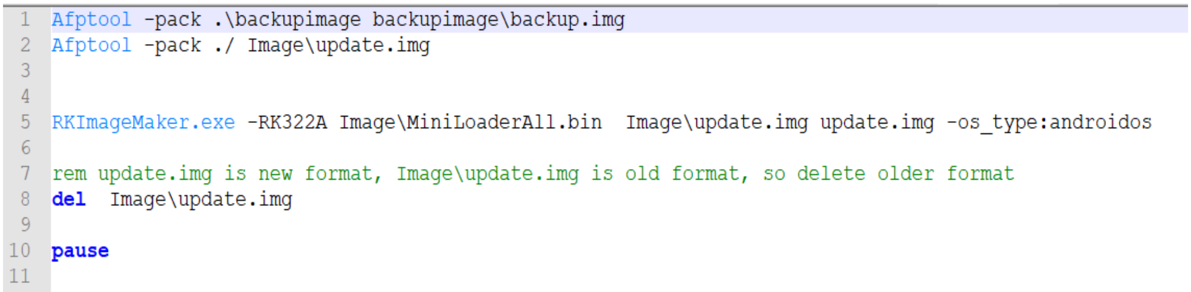
Open the rockdev directory under the AndroidTool directory. Edit package-file.

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

Configure according to package-file, there are some img mirrors 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 mirrors into the directory. Note that the mirror name must be correct during configuration. Pay attention to bootloader option, you should modify according to the loader name generated by yourself.

2) 编辑 mkupdate.bat

Edit mkupdate.bat



```
Afptool -pack .\backupimage backupimage\backup.img
Afptool -pack ./ Image\update.img


RKImageMaker.exe -RK322A Image\MiniLoaderAll.bin  Image\update.img update.img -os_type:androidos

rem update.img is new format, Image\update.img is old format, so delete older format
del  Image\update.img

pause

```

Figure 8-4 update.img 打包脚本 update.imp Packing Script

需要修改 loader 名称为实际存放的 loader 名称即可。

Need to modify loader name as the one actually saved.

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

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

8.6 固件签名工具 Image Signature Tool

选择 chip 类型和加密类型，如果是 RK3229 则选择 efuse。

Choose chip type and encryption type, For RK3229, choose efuse.

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

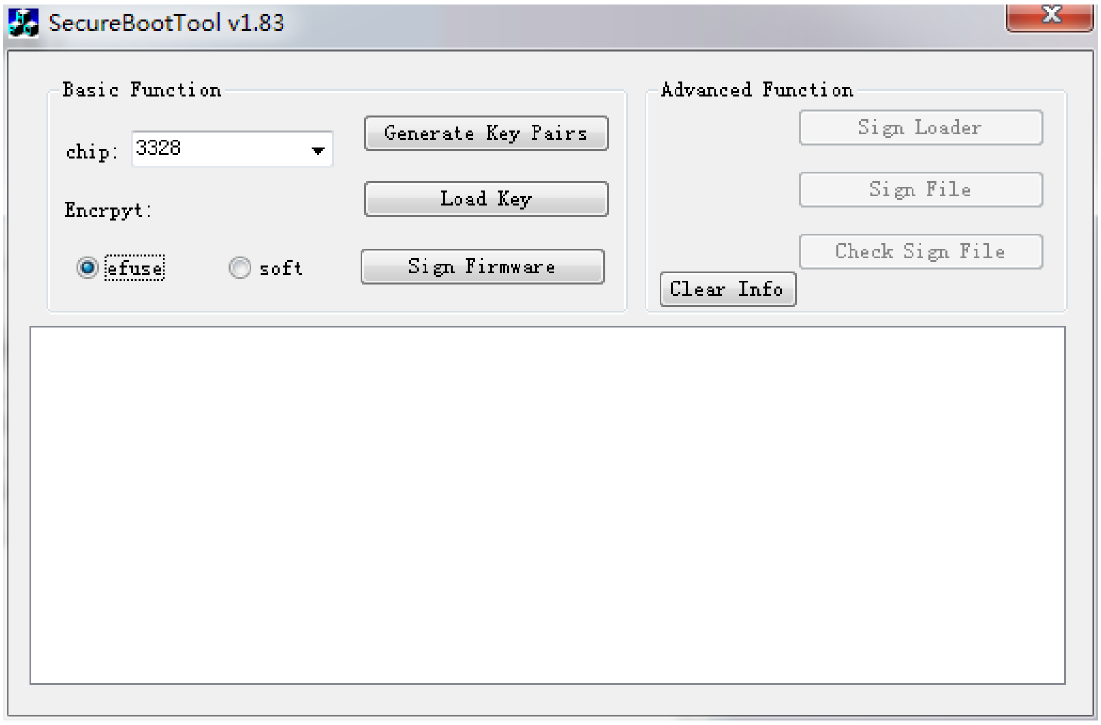
Click “Generate Key Pairs” button, then public and private key pairs are generated, click save to save them.

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

Click “Load Key”, the windows appears two times to choose key files. For the first time, choose private key file, for the second time, choose public key file.

点击”Sign Firmware”按钮，签名 update.img 文件。

Click “Sign Firmware” button, sign the update.img file.



chip: 3328

Encrpyt:

● efuse ○ soft

Figure 8-5 固件签名工具 Image Signature Tool

附键盘输入 R+K+Ctrl+Alt 键可打开右侧隐藏功能。

Note:entering “R+K+Ctrl+Alt” on the keyboard, the hidden function of the tool on the right side will appear.

8.7 序列号/Mac/厂商信息烧写-WNpctool 工具 SN/MAC/Vendor Information Flashing-WNpctool Tool

在 RK3229 平台上，序列号/Mac/厂商信息烧写，都是使用 WNpctool 工具进行的。以下说明该工具基本的用法。

RK3229 platform uses WNpctool tool to flash SN/MAC/vendor information. The basic usage of the tool is described as below.

8.7.1 序列号获取 Serial Number Obtainment

在 RK3229 平台上当未用工具烧写过序列号时，默认是读取 WiFi Mac 地址，并依此随机产生一个序列号的。若需要读取工具烧录的序列号值，需要手动修改对应的配置选项。

On RK3229 platform, when the target device hasn’t been burned serial number using tool, WiFi MAC address is read and a serial number is generated randomly in turn. If you need to read SN value that the tool burned, you need to modify corresponding configuration option manually.

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

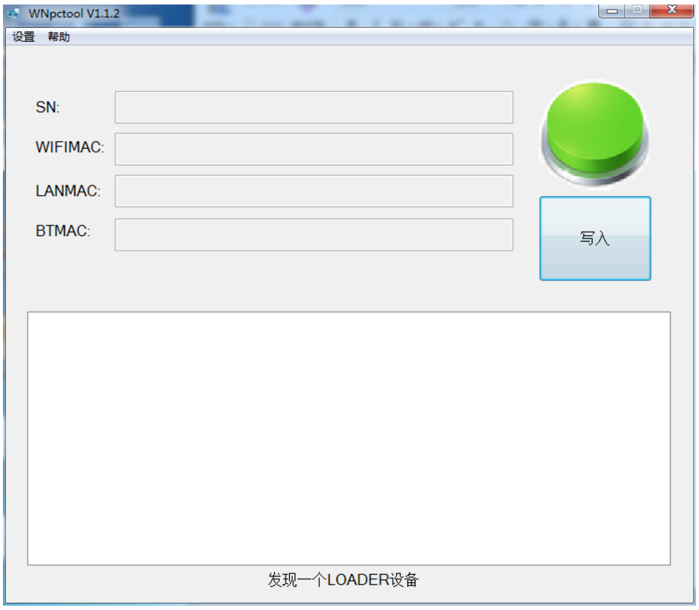
Edit below part in the file “/system/core/ drmservice/drmservice.c”:

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

设为 1 后，默认会从 vendor storage 中读取工具写入的序列号。

After SERIALNO_FROM_IDB is set to be 1, SN burned by tool will be read from vendor storage defaultly

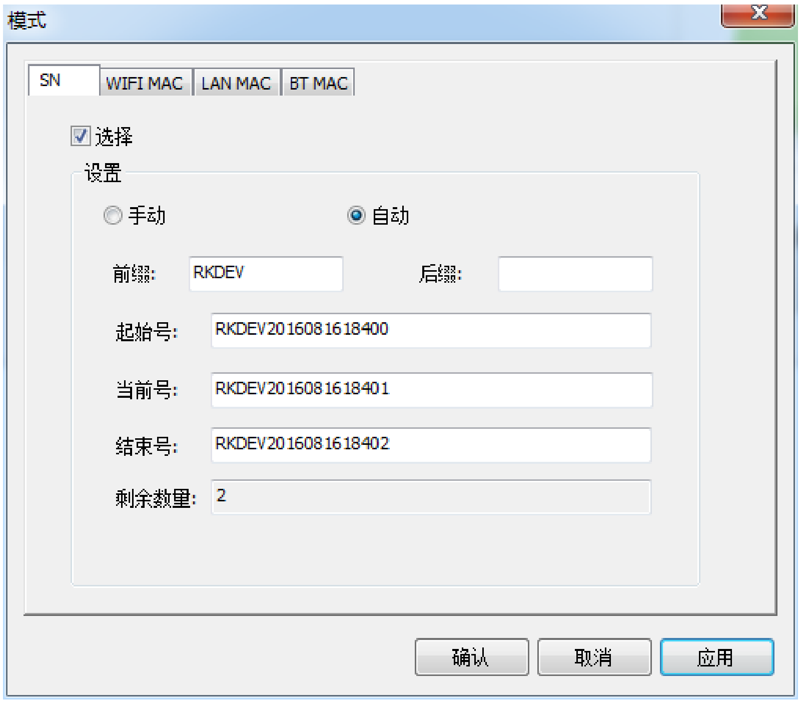
8.7.2 WNpctool 写入步骤 Wnpctool Writing Steps



发现一个LOADER设备

Figure 8-6 WNpctool 工具 Wnpctool Tool

1) 进入 loader 模式。
   Enter loader mode
2) 点击设置按钮，会有一个下拉框按钮，点击“读取”按钮，用来切换是写入还是读取功能。切换到写入功能。
   Click “设置” menu, there will be a drop down list box, deselect “读取” option so that it can switch to write function. This “读取” option is used to switch write and read function.
3) 点击模式，出现下列窗口，用来设置 SN/WIFI/LAN/BT
   Click “设置” menu, there will be a drop down list box, click “模式” option, below window appears, it’s used to set “SN/WIFI/LAN/BT”.



☑ 选择

设置

○ 手动 ● 自动

前缀: RKDEV

后缀:

起始号: RKDEV20160816 18400

当前号: RKDEV20160816 18401

结束号: RKDEV20160816 18402

剩余数量: 2

Figure 8-7 WNpctool 工具模式设置 WNpctool Tool Mode Setting

4) 设置完成后，点击应用按钮，关闭窗口，返回主窗口，点击写入按钮即可。
After setting, click “应用” button, close mode setting window and back to the main window, at last Click “写入” button.

8.7.3 WNpctool 读取步骤 Wnpctool Reading Steps

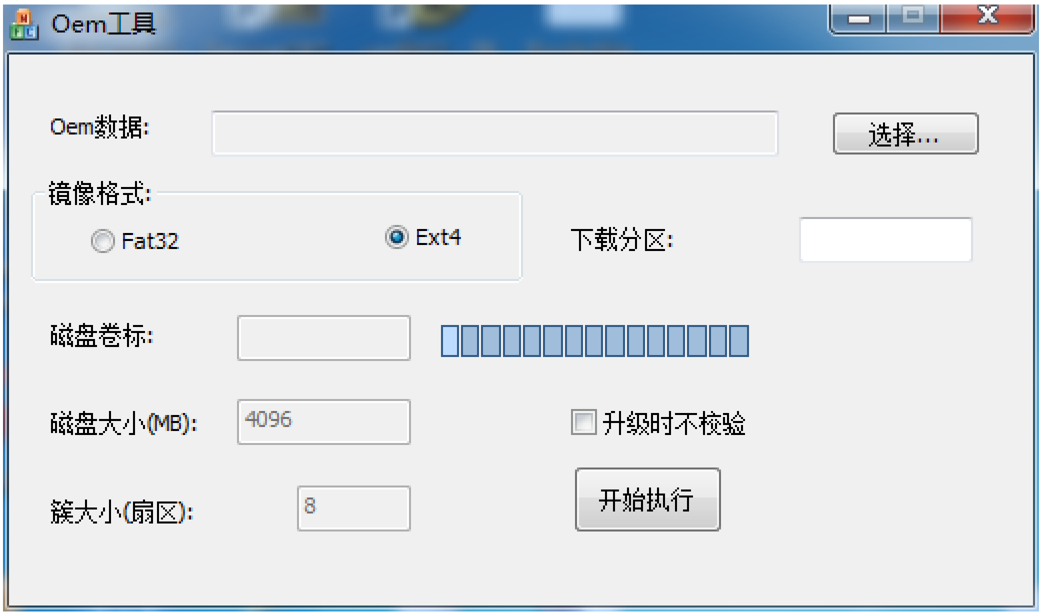
1) 进入 loader 模式。
Enter loader mode
2) 点击设置按钮，会有一个下拉框按钮，点击“读取”按钮，用来切换是写入还是读取功能。切换到读取功能。
Click “设置” menu, there will be a drop down list box, select “读取” option, so that it can switch to read function.
3) 点击“读取”即可。
At last click “读取” button.

8.8 OemTool 打包工具 OemTool Package Tool

8.8.1 Oem 打包工具步骤 Oem Package Tool Steps

RK3229 只支持 Ext4 镜像格式，故镜像格式选择 Ext4.下载分区默认 UserData 分区，可直接不填写。

RK3229 only supports Ext4 image format. So choose Ext4. Download partition uses UserData partition defaultly, it could be unfilled.


Oem工具

Oem数据:

选择...

镜像格式:

○ Fat32 ● Ext4

下载分区:

磁盘卷标:

磁盘大小(MB): 4096

□ 升级时不校验

簇大小(扇区): 8

开始执行

Figure 8-8 Oem 工具 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，假设你目录为/data/media/0,且 0 有一个文件为 sss.txt(如下图示)。则当你升级完 demo 镜像的时候，会在 RK3229 系统上的 data 目录下有目录 media/0,且在 0 目录下有文件 sss.txt 存在。

Click “选择” button to choose data need to be packaged, the data must be a directory. The most external part of the directory is data directory. Suppose your directory is “/data/media/0”, moreover there is a sss.txt file under 0 directory(shown as below). After you upgrade demo image, there will be a directory “media/0” under data directroy on RK3229 system, moreover there is a sss.txt file under 0 directory.

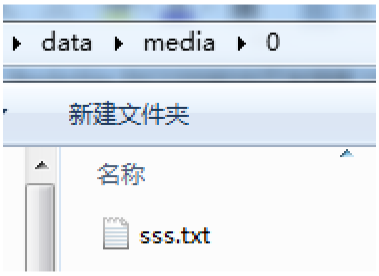


Figure 8-9 工具镜像制作文件夹路径要求 Oem Tool Image Make Folder Path Requirement

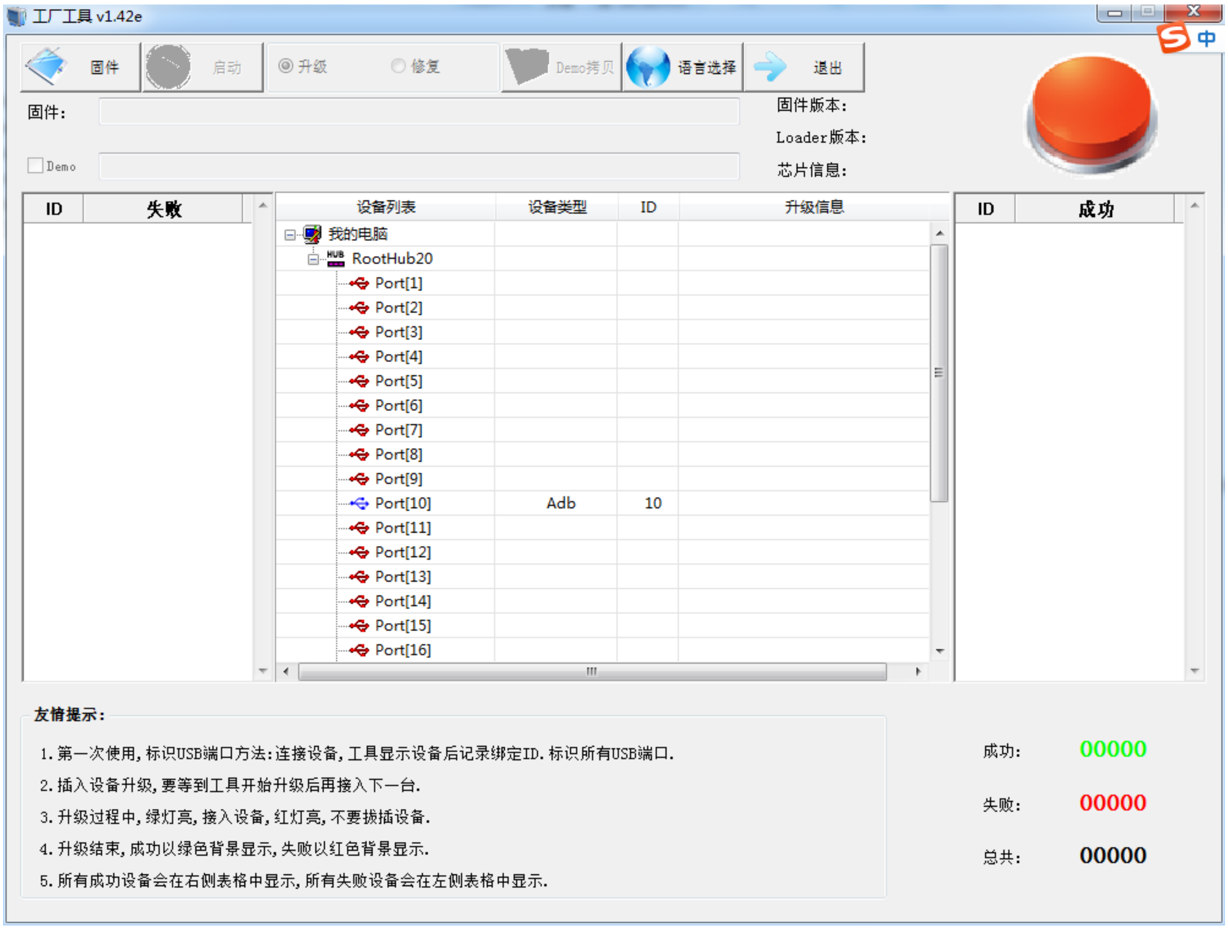
文件选择成功后，直接点击开始执行，会在 Oem 工具目录生成一个 OemImage.img 镜像。将镜像放在 FactoryTool 工具上下载即可。

After the file is selected, click “开始执行” to start run directly, an OemImage.img mirror is generated under Oem tool directory, place this mirror on FactoryTool to download.

8.9 量产工具使用 Factory Tool Usage

8.9.1 工具下载步骤 Tool Download Steps

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 “固件” menu, select the update.img packed by the package tool, and then wait for unpacking success.
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 you need demo mirror image, then click demo copy button, add mirror image 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 to the device, make it enter loader or maskrom mode, then 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.



设备列表	设备类型	ID	升级信息
我的电脑			
RootHub20			
Port[1]			
Port[2]			
Port[3]			
Port[4]			
Port[5]			
Port[6]			
Port[7]			
Port[8]			
Port[9]			
Port[10]	Adb	10	
Port[11]			
Port[12]			
Port[13]			
Port[14]			
Port[15]			
Port[16]			

友情提示：

1. 第一次使用，标识USB端口方法：连接设备，工具显示设备后记录绑定ID，标识所有USB端口.
2. 插入设备升级，要等到工具开始升级后再接入下一台.
3. 升级过程中，绿灯亮，接入设备，红灯亮，不要拔插设备.
4. 升级结束，成功以绿色背景显示，失败以红色背景显示.
5. 所有成功设备会在右侧表格中显示，所有失败设备会在左侧表格中显示.

成功: 00000

失败: 00000

总共: 00000

Figure 8-10 量产工具 Factory Tool

8.10 Box 厂测工具 Box Production Test Tool

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

This test tool is used to help test whether a device is good or not during MP process, and do long time running aging and stability test. Test tool only needs U disk or SDCard to guide start, it’s convenient and efficient, can help improve production efficiency.

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

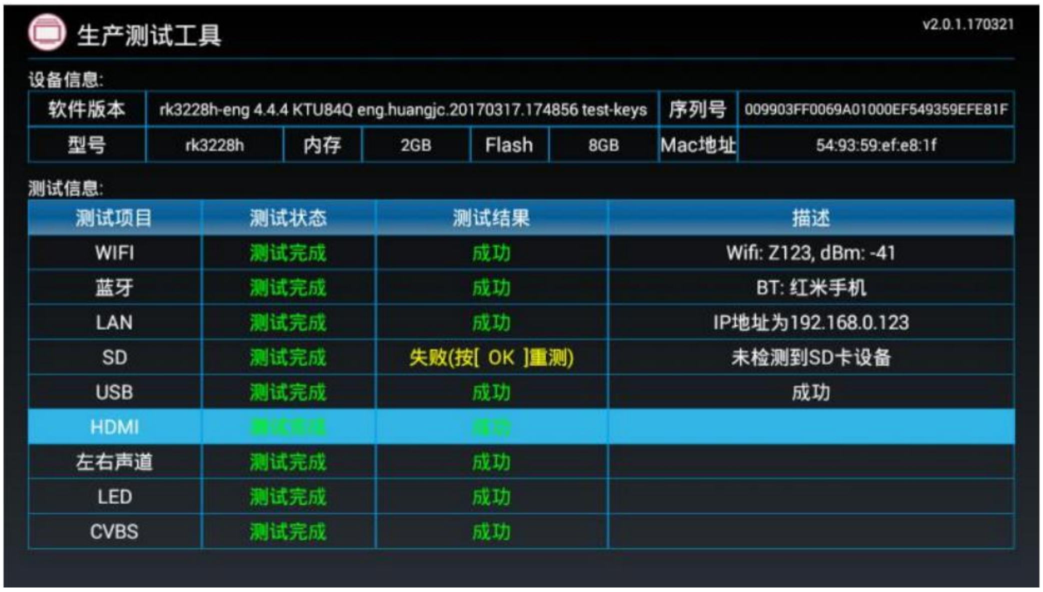
This test tool is applicable to PCBA or complete machine runing complete image, including function test and aging test. Function test mainly includes WiFi, BT, LAN, SD, USB, HDMI, left-right sound track, button, LED, CVBS and so on. Aging test includes CPU, VPU, GPU and Memory test.

SDK 默认编译已带有该测试工具，具体操作说明请参考 RKDocs\common\RKTools manuals\Rockchip_User_Guide_Box_Factory_Test_Tool_CN.pdf。

SDK default compiling carries with this test tool, detailed operating instruction please refer to “RKDocs\common\RKTools manuals\Rockchip_User_Guide_Box_Factory_Test_Tool_CN.pdf”.

配置文件参考请见 RKTools\windows\Rockchip Box 厂测工具 V3.0.rar。

Configuration file please refer to “RKTools\windows\Rockchip Box 厂测工具 V3.0.rar”



生产测试工具 v2.0.1.170321

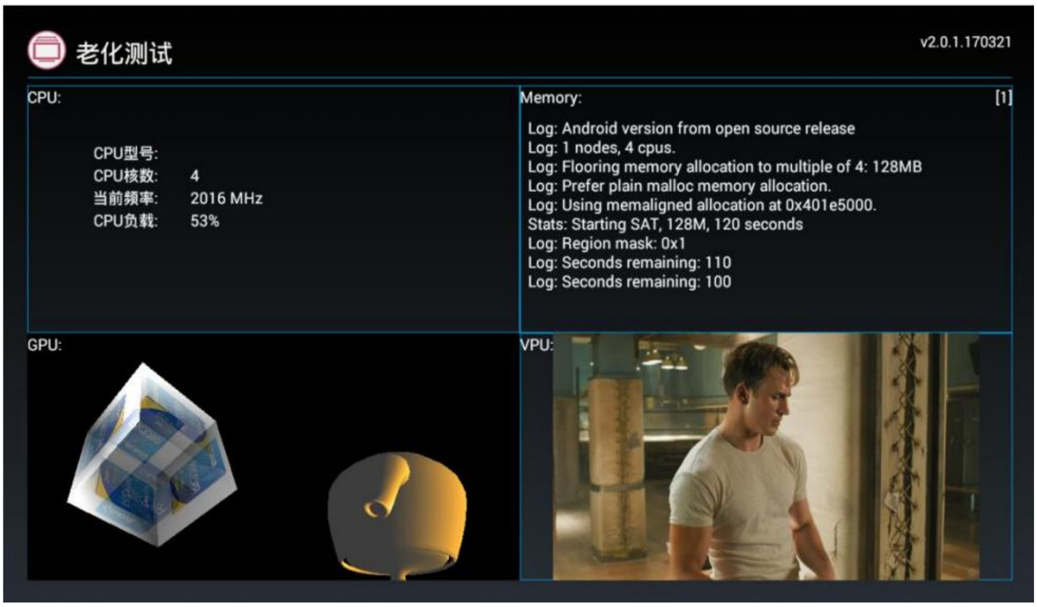
设备信息:

软件版本	rk3228h-eng 4.4.4 KTU84Q eng.huangjc.20170317.174856 test-keys					序列号	009903FF0069A01000EF549359EFE81F
型号	rk3228h	内存	2GB	Flash	8GB	Mac地址	54:93:59:ef:e8:1f

测试信息:

测试项目	测试状态	测试结果	描述
WIFI	测试完成	成功	Wifi: Z123, dBm: -41
蓝牙	测试完成	成功	BT: 红米手机
LAN	测试完成	成功	IP地址为192.168.0.123
SD	测试完成	失败(按[OK]重测)	未检测到SD卡设备
USB	测试完成	成功	成功
HDMI	测试完成	成功	
左右声道	测试完成	成功	
LED	测试完成	成功	
CVBS	测试完成	成功	

Figure 8-11 功能测试界面 Function Test Window



老化测试 v2.0.1.170321

CPU:

CPU型号:
CPU核数: 4
当前频率: 2016 MHz
CPU负载: 53%

Memory: [1]

```
Log: Android version from open source release
Log: 1 nodes, 4 cpus.
Log: Flooring memory allocation to multiple of 4: 128MB
Log: Prefer plain malloc memory allocation.
Log: Using memaligned allocation at 0x401e5000.
Stats: Starting SAT, 128M, 120 seconds
Log: Region mask: 0x1
Log: Seconds remaining: 110
Log: Seconds remaining: 100
```

Figure 8-12 老化测试界面 Aging Test Window

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

Note: As this tool is guided start when a peripheral device plugs in. If customer take manual installation after burning the machine, then the machine needs to be restarted for later test, otherwise because of system limitation, apk service will not start at once, it will cause the problem that test file is not able to be recognized.

Copyright 2019 @Fuzhou Rockchip Electronics Co., Ltd.