密级状态：绝密(　　)　秘密(　　)　内部(　)　公开(√)

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

RK3399Pro_Android9.0_软件开发指南

RK3399Pro_Android9.0_Software_Developer_Guide

（技术部，第二系统产品部）

(Technical Department, R & D Dept. II)

文件状态： Status: [　] 正在修改 [　]Modifying [√] 正式发布 [√]Released	当前版本： Current Version:	V1.0.1
	作　　者： Author:	卞金晨 Bian Jinchen
	完成日期： Finish Date:	2019-08-06
	审　　核： Auditor:	吴良清 Wu Liangqing
	完成日期： Finish Date:	2019-08-06

福州瑞芯微电子股份有限公司

Fuzhou Rockchip Electronics Co., Ltd

版 本 历 史 Revision History

版本号 Version no.	作者 Author	修改日期 Revision Date	修改说明 Revision description	备注 Remark
V1.0.0	卞金晨 Bian Jinchen	2019.08.06	发布初始版本 Initial version release	
V1.0.1	周为新 Weixin Zhou	2019.08.14	添加 npu pice 通信版本说明 Add npu pice communication version description	

目 录 Content

前 言 Preface

概述 Overview

本文档主要介绍 Rockchip RK3399Pro Android9.0 软件开发指南，旨在帮助软件开发工程师更快上手 RK3399Pro 的开发及调试。

This document mainly describes Rockchip RK3399Pro Android9.0 software development guide aiming to help software engineers familiar with RK3399Pro development and debugging quickly.

产品版本 Product version

芯片名 Chipset name	内核版本 kernel version	Android 版本 Android version
RK3399Pro	Linux4.4	Android9.0.0

读者对象 Object

本文档（本指南）主要适用于以下工程师：

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

技术支持工程师

Field application engineers

软件开发工程师

Software development engineers

1 支持列表 Support list

1.1 DDR 支持列表 DDR support list

RK3399Pro DDR 目前选型列表支持双通道 DDR3、DDR3L、LPDDR3、LPDDR4。

RK3399Pro DDR current AVL supports dual channel DDR3, DDR3L, LPDDR3, LPDDR4.

Table 1-1 RK3399Pro DRAM Support Type

Chip	DRAM Support Type
RK3399Pro	DDR3/DDR3L/LPDDR3/LPDDR4

RK3399Pro DDR 颗粒支持程度列表，详见 RKDocs\common\Platform support lists 目录下《RK DDR Support List Ver2.39》，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。

RK3399Pro DDR component support level refers to 《RK DDR Support List Ver2.39》 in the directory of RKDocs\common\Platform support lists. Only recommend to use the components marked with the symbol √ and T/A as shown in below table.

Table 1-2 RK3399Pro DDR Support Symbol

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

1.2 EMMC 支持列表 EMMC support list

RK3399Pro 支持 eMMC 5.1，SDIO3.0，可运行 HS200,HS400 模式，详见 RKDocs\common\Platform support lists 目录下《RKeMMCSupportList Ver1.41》，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。

RK3399Pro supports eMMC 5.1, SDIO3.0, and can run HS200, HS400 mode. For more details, refer to 《RKeMMCSupportList Ver1.41》 in the directory of RKDocs\common\Platform support lists. Only recommend to use the components marked with the symbol √ and T/A as shown in below table.

Table 1-3 RK3399Pro EMMC Support Symbol

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

1.2.1 高性能 EMMC 颗粒的选取 High performance eMMC component selection

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

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

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

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

如有选型上的疑问，也可直接联系我们的 Fae 窗口。

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

6.1.5 Performance

[Table 23] Performance

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

图 1- 1 EMMC Performance 示例

Picture 1-1 EMMC Performance example

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

RK3399Pro 和 RK3399 WiFi/BT 支持列表可以共用，RK3399Pro 内核运行 Linux4.4，WiFi/BT 支持列表，详见 RKDocs\common\Platform support lists 目录下《Rockchip_WiFi_Situation_20180403.pdf》，下表中所标示为目前 RK3399 上大量测试过的 Wifi/Bt 芯片列表，建议按照列表上的型号进行选型。如果有其他 WiFi/BT 芯片调试，可先与 WiFi/BT 芯片原厂沟通，是否有可以稳定在 Linux4.4 运行的驱动程序，并能提供调试帮助。

RK3399Pro kernel is Linux4.4. RK3399Pro and RK3399 share the same Wi-Fi/BT support list《Rockchip_WiFi_Situation_20180403.pdf》in the directory of RKDocs\common\Platform support lists.

Below table shows the Wi-Fi/BT chipset list currently already verified in RK3399. Recommend to choose the component in the table. If want to debug other Wi-Fi/BT chipset, first need to communicate with Wi-Fi/BT vendor whether they can provide the driver program which can work on Linux4.4 stably and technical support during debugging.

另外后续我们会不断更新支持列表，如果疑问和建议可以与我们的 Fae 窗口联系(WIF/BT avl 可以和 RK3399 共用)。

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

RK3399 Wi-Fi Situation

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

1. √：支持 ×：不支持 注：空的表示没调过
2. 该列表仅适用kerne14.4

图 1-2 RK3399 目前大量测试的 Wifi/Bt 支持列表

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

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

本 SDK 是基于谷歌 Android9.0 64bit 系统，适配瑞芯微 RK3399Pro 芯片的软件包。

This SDK is compatible with RK3399Pro chipset software package based on Google Android9.0 64bit system.

如果使用瑞芯微提供的开发板，具体参考《3399Pro_Evb 板说明》，kernel 配置可直接使用 rk3399pro-evb-v11-avb.dts 进行配置。

If using Rockchip evb board, refer to 《3399Pro_Evb board instruction》 for details. You can use rk3399pro-evb-v11-avb.dts to config kernel directly.

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

RK3399Pro 多媒体方面支持强大，支持 4K VP9 and 4K 10bits H265/H264 视频解码，高达 60fps，1080P 多格式视频解码 (WMV, MPEG-1/2/4, VP8)，1080P 视频编码，支持 H.264，VP8 格式，视频后期处理器：反交错、去噪、边缘/细节/色彩优化。

RK3399Pro has powerful multimedia which supports 4K VP9 and 4K 10bits H265/H264 video

decoder up to 60fps, 1080P multi-format video decoder (MWV, MPEG-1/2/4, VP8), 1080P video encoder, H.264, VP8 format, video post processor: de-interleaving, de-noising, edge/detail/color optimization.

具体的编解码支持列表，详见 RKDocs\rk3399Pro 目录下《RK3399 Multimedia Codec Benchmark v1.0》。

For detailed encoder/decoder support list, refer to 《RK3399 Multimedia Codec Benchmark v1.0》 in the directory of RKDocs\rk3399Pro.

1.6 NPU 开发文档 NPU development document

NPU 的开发资料参考以下目录：

NPU development documents refer to the following directories:

/rknn-toolkit

/RKNPUTools

RKDocs/rk3399pro/Rockchip_RK3399Pro_Introduction_NPU_Boot_Up_CN.pdf

2 文档/工具索引 Document/tool index

2.1 文档索引 Document index

RK3399Pro SDK 发布文档旨在帮助开发者快速上手开发及调试，文档中涉及的并不能涵盖所有的知识和问题。文档列表也正在不断更新，如有文档上的疑问及需求，请联系我们的 Fae 窗口。

RK3399Pro 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.

RK3399Pro SDK 中在 RKDocs 目录下附带了三大块的文档，分别为：android（android 相关开发文档），rk3399Pro(3399Pro 相关发布文档)，common（公共开发文档）；common 目录细分为内核驱动开发文档、uboot 开发文档、模块开发文档、Platform support lists（支持列表）、RKTools manuals（工具使用文档）等。

RK3399Pro SDK includes three kinds of documents in RKDocs directory, android(android related development documents), rk3399Pro(RK3399Pro related release documents), and common(common development documents). Common directory consists of kernel driver development documents, uboot development documents, module development documents, Platform support lists (support list), RKTools manuals (tool usage document) etc.

```
|-- android
|   |-- 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
|   |-- bt
|   |   |-- Rockchip_Introduction_Android8.1_BT_Configuration_CN.pdf
|   |   `-- Rockchip_Introduction_Android9.0_BT_Configuration_CN.pdf
|   |-- project.config
|   `-- 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_De
bugging_CN.pdf
|   |   |-- Rockchip_Developer_Guide_Linux4.4_Audio_CN.pdf
|   |   `-- Rockchip_Developer_Guide_RK817_RK809_Codec_CN.pdf
|   |-- 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
```

```
|   |-- DVFS
|   |   |-- Rockchip-Developer-Guide-Linux4.4-CPUFreq-CN.pdf
|   |   `-- Rockchip-Developer-Guide-Linux4.4-Devfreq.pdf
|   |-- GMAC
|   |   `-- Rockchip_Developer_Guide_Ethernet_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-MCU-EN.pdf
|   |   `-- Rockchip-Developer-Guide-linux4.4-MCU.pdf
|   |-- MMC
|   |   `-- Rockchip-Developer-Guide-linux4.4-SDMMC-SDIO-eMMC.pdf
|   |-- PCie
|   |   `-- Rockchip-Developer-Guide-linux4.4-PCIe.pdf
|   |-- PIN-Ctrl
|   |   `-- Rockchip-Developer-Guide-Linux-Pin-Ctrl-CN.pdf
|   |-- PMIC
|   |   |-- Archive.zip
|   |   |-- Rockchip-Developer-Guide-Power-Discrete-DCDC-Linux4.4.pdf
|   |   |-- Rockchip-Developer-Guide-RK805.pdf
|   |   |-- Rockchip-Developer-Guide-RK818_6-Fuel-Gauge.pdf
|   |   |-- Rockchip-RK818-RK816-FG-Log-Description-linux4.4.pdf
|   |   `-- Rockchip_Developer_Guide_RK817_RK809_Fuel_Gauge_CN.pdf
|   |-- PWM
|   |   |-- Rockchip-Developer-Guide-Linux-PWM-CN.pdf
|   |   `-- Rockchip_Developer_Guide_PWM_IR_CN.pdf
```

```
|   |-- Platform\ support\ lists
|   |   |-- RK\ DDR\ Support\ List\ Ver2.38.pdf
|   |   |-- RK3128\ BOX\ \ Hardware\ Design\ Guide\ V10-201410.pdf
|   |   |-- RKNandFlashSupportList\ Ver2.73_20180615.pdf
|   |   |-- RKeMMCSupportList\ Ver1.43_2019_03_15.pdf
|   |   |-- Rockchip_Camera_AVL_v2.0_Package.7z
|   |   |-- Rockchip_Introduction_WiFi_Situation_CN.pdf
|   |   `-- Rockchip_Kodi_Support_List_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_KeyWrite_CN.pdf
|   |   |-- Rockchip_User_Guide_Keybox_Burning_EN.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
|   |-- SPI
|   |   `-- Rockchip-Developer-Guide-linux4.4-SPI.pdf
|   |-- Sensors
|   |   `-- Rockchip_Developer_Guide_Sensors_CN.pdf
|   |-- TRUST
|   |   |-- Rockchip-Developer-Guide-RK3308-System-Suspend.pdf
|   |   `-- Rockchip-Developer-Guide-Trust.pdf
|   |-- Thermal
|   |   |-- Rockchip-Developer-Guide-Linux4.4-Thermal-CN.pdf
|   |   `-- Rockchip-Developer-Guide-Linux4.4-Thermal-EN.pdf
|   |-- UART
|   |   |-- Rockchip-Developer-Guide-RT-Thread-UART.pdf
|   |   `-- Rockchip-Developer-Guide-linux4.4-UART.pdf
|   |-- camera
|   |   |-- HAL1
|   |   |   |-- CIF_ISP10_Driver_User_Manual_V1.0_20171124.pdf
|   |   |   |-- CIF_ISP11_Driver_User_Manual_V1.0.pdf
|   |   |   |-- Camera_Document_Directory.txt
|   |   |   |-- RK312x_Camera_User_Manual_v1.4(3288&3368).pdf
|   |   |   |-- RKISPV1_Camera_Module_AVL_v1.7.pdf
|   |   |   |-- RK_ISP10_Camera_User_Manual_v2.2.pdf
|   |   |   |-- Rockchip\ SOFIA\ 3G-R_PMB8018(x3_C3230RK)_Camera_Module_AVL_v1.6_2
0160226.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
|   |   |   `-- readme_En.txt
|   |   |-- HAL3
|   |   |   |-- RKCIF_Driver_User_Manual_v1.0.pdf
```

```
|   |   |   |-- RKISP_Driver_User_Manual_v1.2.pdf
|   |   |   |-- camera_engine_rkisp_user_manual_v2.0.pdf
|   |   |   `-- camera_hal3_user_manual_v2.1.pdf
|   |   `-- README.txt
|   |-- 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
|   |-- hdmi-in
|   |   `-- Rockchip_Developer_Guide_HDMI_IN_CN.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
|   |-- power
|   |   `-- Rockchip_Developer_Guide_Sleep_and_Resume_CN.pdf
|   |-- security
|   |   |-- Efuse\ process\ explain\ .pdf
|   |   |-- RK3399_Efuse_Operation_Instructions_V1.00_20190214_EN.pdf
```

```
|   |   |-- Rockchip\ Vendor\ Storage\ Application\ Note.pdf
|   |   |-- Rockchip-Secure-Boot-Application-Note-V1.9.pdf
|   |   |-- Rockchip_Developer_Guide_TEE_Secure_SDK_CN.pdf
|   |   `-- Rockchip_RK3399_Introduction_Efuse_Operation_EN.pdf
|   |-- u-boot
|   |   |-- Rockchip-Developer-Guide-Linux-AB-System.pdf
|   |   |-- Rockchip-Developer-Guide-Trust.pdf
|   |   |-- Rockchip-Developer-Guide-UBoot-nextdev-CN.pdf
|   |   `-- Rockchip-Developer-Guide-Uboot-mmc-device-driver-analysis.pdf
|   `-- usb
|       |-- 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-PHY-CN.pdf
|       |-- Rockchip-Developer-Guide-USB-Performance-Analysis-CN.pdf
|       |-- Rockchip-Developer-Guide-USB-SQ-Test-CN.pdf
|       `-- putty20190213_162833_1.log
`-- rk3399pro
    |-- Rockchip_RK3399Pro_Android8.1_SDK_Release_V1.01_201801215_EN.pdf
    |-- Rockchip_RK3399Pro_Developer_Guide_Android8.1_Software_CN.pdf
    |-- Rockchip_RK3399Pro_Developer_Guide_Android8.1_Software_EN.pdf
    |-- Rockchip_RK3399Pro_Introduction_NPU_Boot_Up_CN.pdf
    |-- Rockchip_RK3399Pro_User_Guide_Hardware_CN.pdf
    |-- Rockchip_RK3399Pro_User_Guide_Hardware_EN.pdf
    |-- Rockchip_RK3399_Introduction_Multimedia_Codec_Benchmark_EN.pdf
    |-- Rockchip_RK3399_Introduction_SDK_Multimedia_Performance_EN.pdf
    `-- Rockchip_RK3399_Introduction_USB_DTS_Configuration_CN.pdf
```

RK3399Pro SDK 发布的工具，用于开发调试阶段及量产阶段使用。工具可能随 SDK 更新不断

更新，如有工具上的疑问及需求，请联系我们的 Fae 窗口。

RK3399Pro SDK released tool is used in development debugging stage and MP stage. The tool may upgrade along with new SDK. Please contact with our FAE if there is any question or requirement about the tool.

RK3399Pro 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）。

RK3399Pro SDK contains linux (tool used in Linux operation system environment) and windows (tool used in Windows operation system environment) in RKTools directory.

```
|-- linux
|    |-- Linux_AttestationKeyboxPack_Tool.rar
|    |-- Linux_Pack_Firmware
|    |    |-- Linux_rockdev.zip
|    |    `-- rockdev
|    |          |-- afptool
|    |          |-- mkupdate.sh
|    |          |-- package-file
|    |          |-- readme.txt
|    |          |-- rkImageMaker
|    |          `-- unpack.sh
|    |-- Linux_SecureBoot
|    |    `-- SecureBootConsole_v1.90.rar
|    |-- Linux_TA_Sign_Tool.rar
|    |-- Linux_Upgrade_Tool
|    |    `-- Linux_Upgrade_Tool_v1.43.zip
|    `-- Linux_Upgrade_Tool_v1.43.zip
`-- windows
|-- AndroidTool
|    |-- AndroidTool_Release_v2.65.zip
|    `-- rockdev
|          |-- AFPTool.exe
```

```
|           |-- RKImageMaker.exe
|           |-- backupimage
|           |    |-- backup.img
|           |    `-- package-file
|           |-- baseparamer.img
|           |-- mkupdate.bat
|           |-- package-file
|           |-- recover-script
|           `-- update-script
|-- BatteryArrayTestTool
|    |-- ADC_BatteryArray_TestTool_V2.3.pdf
|    `-- BatteryArray_V2.4.apk
|-- Demo_Image_DownloadTools.zip
|-- DriverAssitant_v4.5.zip
|-- FWFactoryTool_V5.52.rar
|-- FactoryTool_1.66.zip
|-- KeyBoxWrite_v1.53.rar
|-- OemTool_v1.3.rar
|-- RKDevInfoWriteTool_Setup_V1.0.3.rar
|-- RKImageMaker_v1.62.zip
|-- Rockchip_Box_FactoryTestTool_V2.0-M-20170327.zip
|-- Rockchip_Platform_DDR_Test_Tool_V1.38_Release_Annoucement_CN.7z
|-- Rockchip_Platform_DDR_Test_Tool_V1.38_Release_Annoucement_EN.7z
|-- SDDiskTool_v1.57.zip
|-- SecureBootTool_v1.94.zip
|-- SpiImageTools_v1.41.zip
|-- UpgradeDllTool_v1.35.zip
|-- Windows_TA_Sign_Tool.rar
|-- efuse_v1.37.rar
|-- parameter_adjustment_tool.xlsx
```

```
`-- rk312x-pcba-tools.rar
```

3 SDK 编译/烧写 SDK compiling/flashing

3.1 SDK 获取 How to get SDK

SDK 通过瑞芯微代码服务器对外发布。客户向瑞芯微技术窗口申请 SDK，需同步提供 SSH 公钥进行服务器认证授权，获得授权后即可同步代码。

SDK is released through Rockchip code server. Customers apply SDK from Rockchip FAE contact, and will be able to sync code after obtaining the server certificate authorization with SSH public key.

3.1.1 SDK 下载链接 SDK download link

RK3399Pro_ANDROID9.0_SDK 下载地址如下：

RK3399Pro_ANDROID9.0_SDK download address is as below:

repo init --repo-url=ssh://git@www.rockchip.com.cn:2222/repo-release/tools/repo.git -u ssh://git@www.rockchip.com.cn:2222/Android_pie_stable/platform/rk3399pro/manifests.git -m Rk3399pro_Android_Pie_release.xml

3.1.2 repo

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

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

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

3.1.3 SDK 代码压缩包 SDK code compressed package

为方便客户快速获取 SDK 源码，瑞芯微技术窗口通常会提供对应版本的 SDK 初始压缩包，开发者可以通过这种方式，获得 SDK 代码的初始压缩包，该压缩包解压得到的源码，与通过 repo 下载的源码是一致的。以 Rk3399Pro_Android9.0_SDK_Release_V1.0_20190806.tar.gz 为例，拷贝到该初始化包后，通过如下命令可检出源码：

Rockchip FAE contact usually will provide the initial compressed package of the corresponding version SDK in order to help customers acquire SDK source code quickly. Developer can acquire the SDK code initial compressed package in this way and unzip it to get the source code. It is the same as the source code downloaded through repo. Take Rk3399Pro_Android9.0_SDK_Release_V1.0_20190806.tar.gz as an example, you can sync the source code through below command after copy the initial package:

```
mkdir rk3399Pro
tar zxvf Rk3399Pro_Android9.0_SDK_Release_V1.0_20190806.tar.gz -C rk3399Pro
cd rk3399Pro
.repo/repo/repo sync -l
.repo/repo/repo sync
```

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

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

3.1.4 补充说明 Supplementary description

Android9.0 SDK 已不再支持 UMS 功能，平台设备皆使用合并分区；

Android9.0 SDK doesn't support UMS function any more. All the devices use combined partition.

Android9.0 SDK 已支持全盘加密功能；

Android9.0 SDK already supports full disk encryption function.

Android9.0 SDK 已支持 Verified boot 2.0 (avb)的功能。

Android9.0 SDK already supports Verified boot 2.0 (avb) function.

有关其他 Android 9.0 中新增加的说明和修改，请参照

RKDocs/android/Rockchip_Introduction_Android9.0_System_New_Feature_CN&EN.pdf

For other related new descriptions and changes of Android9.0, please refer to RKDocs/android/Rockchip_Introduction_Android9.0_System_New_Feature_CN&EN.pdf.

3.2 SDK 编译 SDK compiling

3.2.1 JDK 安装 JDK installation

Android9.0 系统编译依赖于 JAVA 8。编译之前需安装 OpenJDK。

Android9.0 system compiling is dependent on JAVA 8. Need to install OpenJDK before compiling.

安装命令如下：

Install command is as below:

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

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

Configure JAVA environment variable, for example, if the install path is /usr/lib/jvm/java-8-ope

njdk-amd64, it is able to execute below command to configure environment variable at the termination.

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

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

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

可直接执行以下命令，配置 JDK：

Directly execute below command to configure JDK:

```
source javaenv.sh
```

3.2.2 编译模式 Compiling mode

SDK 默认以 userdebug 模式编译。

SDK default compiling mode is userdebug.

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

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

3.2.3 RK3399Pro Evb 编译 RK3399Pro EVB compilation

uboot 编译:

uboot compiling:

```
cd u-boot
./make.sh rk3399pro
```

kernel 编译:

kernel compiling:

```
如果客户有拿到我们的 evb_v10 的硬件板子（绿色）kernel 编译方法如下：
If for evb_v10 board(green), kernel compiling method is as below:
cd kernel
make ARCH=arm64 rockchip_defconfig -j8
make ARCH=arm64 rk3399pro-evb-v10.img -j12
```

如果客户有拿到我们的 evb_v11 的硬件板子（黑色）kernel 编译方法如下：

If for evb_v11 board(black), kernel compiling method is as below:

```
cd kernel
make ARCH=arm64 rockchip_defconfig -j8
make ARCH=arm64 rk3399pro-evb-v11.img -j12
```

android 编译:

android compiling:

```
source build/envsetup.sh
lunch rk3399pro-userdebug
make -j12
./mkimage.sh
```

3.2.4 固件生成步骤 Image build steps

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

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

```
rockdev/Image-xxx/
├── boot.img
├── dtbo.img
├── kernel.img
├── MiniLoaderAll.bin
├── misc.img
├── oem.img
├── parameter.txt
├── pcba_small_misc.img
├── pcba_whole_misc.img
├── recovery.img
├── resource.img
├── system.img
```

```
├── trust.img
├── uboot.img
├── vbmeta.img
└── vendor.img
```

注意：Android9.0 移除了 kernel.img/resource.img，这里的镜像仅供调试，原来的 kernel/resource 打包到了 boot 和 recovery 中；另外，增加了 vbmeta.img 和 dtbo.img，固件烧写的时候必须烧写这两个 img，否则系统无法开机。具体修改细节请参照文档：RKDocs/android/Rockchip_Introduction_Android9.0_System_New_Feature_CN&EN.pdf

Note: kernel.img and resource.img are removed in Android 9.0 and these images are for debugging only. Now the kernel and resource are packaged into boot and recovery. In addition, Android 9.0 adds two new images: vbmeta and dtbo. These two images must be flashed during image flashing, otherwise the system will fail to boot up. For more details, please refer to the document: RKDocs/android/Rockchip_Introduction_Android9.0_System_New_Feature_CN&EN.pdf.

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

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

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

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

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

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

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

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

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

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

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

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

配置步骤如下：

Configure steps are as below:

1) 确保两个配置文件存在，并且权限设置为 0600：

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

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

2) 若两个配置文件不存在，请参照以下文本新建这两个配置文件。

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

config.properties 文件示例如下（端口号需按实际修改）：

config.properties file example is as below (the port number should be modified as needed):

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

```
jack.server.time-out=7200
```

.jack-settings 文件示例如下（端口号需按实际修改）：

.jack-settings file example is as below (the port number should be modified as needed):

```
# Server settings
SERVER_HOST=127.0.0.1
SERVER_PORT_SERVICE=8074
SERVER_PORT_ADMIN=8075


# Internal, do not touch
SETTING_VERSION=4
```

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

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

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


SERVER_PORT_SERVICE=8023
SERVER_PORT_ADMIN=8024
```

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

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

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

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

```
$ JACK_EXTRA_CURL_OPTIONS=-v jack-admin list server
* Protocol https not supported or disabled in libcurl
* Closing connection -1
```

```
Communication error with Jack server 1. Try 'jack-diagnose'
```

3.2.6 全自动编译脚本 Fully automatic compiling script

如前几节所述，编译可大致分为 u-boot、kernel、android 三大部分进行编译，为了提高编译的效率，降低人工编译可能出现的误操作，该 SDK 中集成了全自动化编译脚本，方便固件编译、备份。

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

1) 该全自动化编译脚本原始文件存放于：

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

```
device/rockchip/RK3399Pro/build.sh
```

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

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

<project path="device/rockchip/rk3399Pro" name="rk/device/rockchip/rk3399Pro" remote="rk" revision="rk33/mid/8.1/develop">

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

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

</project>

3) 修改 build.sh 脚本中的特定变量以编出对应产品固件。

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

```
KERNEL_DTS=rk3399pro-evb-v10
```

变量请按实际项目情况，对应修改：

Modify the variable according to the actual project situation:

KERNEL_DTS 变量指定编译 kernel 的产品板极配置；

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

Android 编译需要指定对应的 lunch 选项，请在执行 build.sh 之前执行 lunch 操作，确保使用了正确的 lunch 选项，例如：

Android compiling needs to specify the corresponding lunch option, please execute lunch operation before executing build.sh to make sure to use the correct lunch option. For exemple:

```
lunch rk3399Pro-userdebug
```

4) 执行自动编译脚本：

Execute automatic compiling script:

```
source build.sh
```

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

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

5) 脚本生成内容:

The contents generated by the script:

脚本会将编译生成的固件拷贝至：

The script will copy the compiled images to:

IMAGE/RK3399Pro *****_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。建议在每次大版本编译的时候 ，使用这个编译脚本生成固件，里面包含了很多的版本信息，便于追查问题的时候定位代码的状态。

the directory of IMAGE/RK3399Pro *****_RELEASE_TEST/IMAGES which path is subject to the actual generation. Each compiling will create new directory and save, automatically backup images version during debugging, and keep all the information of images version. Recommend to use this compiling script to generate images for every big version compilation. It includes much version information which is convenient to locate code status while debugging issues.

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

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

3.3 固件烧写 Image flashing

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

Flashing instruction refers to 《Android 开发工具手册.pdf》 in the directory of RKDocs\common\RKTools manuals. SDK provides flashing tools as shown in below picture. After compiling to generate corresponding images, enter flashing mode, it is able to flash images. For the devices

with existing images, you can select to re-flash images, or format the device, erase idb, and then flash the images.

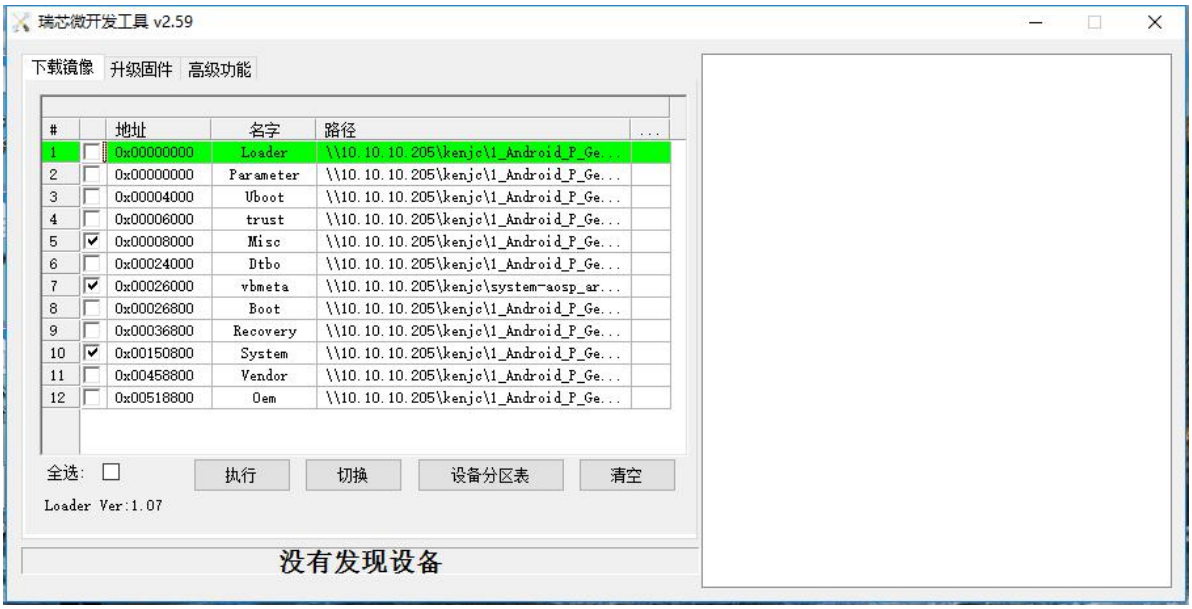

图 3- 1 Android 开发工具烧写界面

Picture 3-1 Android development tool flashing interface

注：Note:

1) 烧写前，需安装最新的 USB 驱动，驱动详见：

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

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

2) Android9.0 移除了 kernel.img/resource.img，相对的，增加了 vbmeta.img 和 dtbo.img，固件烧写的时候必须烧写这两个 img，否则系统无法开机。

Android9.0 removes kernel.img/resource.img, and accordingly adds vbmeta.img and dtbo.img. The two img must be flashed when flashing images, otherwise the system will fail to boot up.

3.4 量产烧写 MP flashing

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

Considering the production efficiency and factory work station arrangement during MP, the flashing instruction refers to 《Rockchip 量产烧录指南 V1.1-20170214.pdf》 in the directory of RKDocs\ common\RKTools manuals.

在量产过程中如涉及到工具上的问题，可以联系我们的 Fae 窗口。

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

4 U-Boot 开发 U-Boot development

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

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

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

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

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

- 支持芯片：rk3288、rk3036、rk312x、rk3368、rk312x、rk3366、rk3399 等；

  Support chip: RK3288, RK3036, RK312X, RK3368, RK312X, RK3366, RK3399, etc.

- 支持 Android 平台的固件启动；

  Support image boot up on Android platform.

- 支持 ROCKUSB 和 Google Fastboot 两种方式烧写；

  Support ROCKUSB and Google Fastboot two kinds of flashing method.

- 支持 secure boot 固件签名加密保护机制；

  Support secure boot image sign encryption protection mechanism.

- 支持 LVDS、EDP、MIPI、HDMI、CVBS 等显示设备；

  Support LVDS, EDP, MIPI, HDMI, CVBS and other display devices.

- 支持 SDCard、Emmc、Nand Flash、U 盘等存储设备；

  Support SDCard, Emmc, Nand Flash, U disk and other memory devices.

- 支持开机 logo 显示、充电动画显示，低电管理、电源管理；

  Support boot up logo display, charging animation display, low battery management, power management.

- 支持 I2C、SPI、PMIC、CHARGE、GUAGE、USB、GPIO、PWM、DMA、GMAC、EMMC、NAND 中断等驱动；

Support I2C, SPI, PMIC, CHARGE, GAUGE, USB, GPIO, PWM, DMA, GMAC, EMMC, NAND interrupt and other drivers.

4.2 平台配置 Platform configuration

平台配置文件位于 U-Boot 根目录下的 configs 文件夹下，其中 Rockchip 相关的以 RK 开头，并根据产品形态分为 MID 和 BOX 两种配置：

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

```
rk3288_defconfig
rk3126_defconfig
rk3128_defconfig
rk3368_defconfig
rk3399Pro_defconfig


rk3288_box_defconfig
rk3128_box_defconfig
rk3036_box_defconfig
rk3368_box_defconfig
rk322x_box_defconfig
rk3399_box_defconfig
```

4.3 固件生成 Images generation

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

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

4.3.1 一级 Loader 模式 Level 1 Loader mode

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

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

```
rk3399pro_loader_v1.22.115.bin
```

其中 V1.22.115 是发布的版本号。

V1.22.115 is the released version number.

4.3.2 二级 Loader 模式 Level 2 Loader mode

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

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

以 rk3399pro 编译生成的镜像为例：

Take the mirror generated by rk3399pro compilation as an example:

```
rk3399pro_loader_v1.22.115.bin
uboot.img
trust.img
```

其中 v1.22.115 是发布的版本号，rockchip 定义 U-Boot loader 的版本，其中 1.22.115 是根据存储版本定义的，客户务必不要修改这个版本。

V1.22.115 is the released version number. U-Boot loader version is defined by rockchip. 1.22.115 is defined according to the memory version and should not be changed by customers.

uboot.img 是 U-Boot 作为二级 loader 的打包。

uboot.img is the package when U-Boot as level 2 loader.

trust.img 是 U-Boot 作为二级 loader 的打包。

trust.img is the package when U-Boot as level 2 loader.

RK3036、RK3126、RK3128、RK322x、RK3368、RK3366、RK3399、RK3399Pro 等采用二级 loader 模式。

RK3036, RK3126, RK3128, RK322x, RK3368, RK3366, RK3399, RK3399Pro etc. use level 2 loader mode.

4.4 U-Boot 编译 U-Boot compiling

RK3399Pro SDK 编译使用的是如下配置：

RK3399Pro SDK compiling uses below configuration:

./make.sh rk3399pro

编译完，会生成 trust.img、rk3399pro_loader_v1.22.115.bin、uboot.img 三个文件。

After compiling, it will generate three files: trust.img, rk3399pro_loader_v1.22.115.bin and uboot.img.

目前编译出来的 rk3399pro_loader_v1.22.115.bin DDR 为定频 800MHz 版本。

Currently compiled rk3399pro_loader_v1.22.115.bin DDR frequency is fixed as 800MHz.

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

4.5.1 充电图片打包 U-boot charging logo package

充电图片需要打包进 resource.img 才能被充电驱动读取并且显示。编译内核时默认不会打包充电图片，所以需要另外单独把这些图片打包进 resource.img。

Charging images can be read and displayed by the charging driver only after they are packaged into resource.img. Charging images are not packaged by default when compiling the kernel, so you need to package them separately into resource.img.

打包命令：

The package command:

./scripts/pack_resource.sh <input resource.img>

这个命令默认会把./tools/images/目录里的图片作为充电图片打包进 resource.img，新的 resource.img 会生成在 UBoot 根目录下，请把该 resource.img 拷贝回 kernel 目录，并在 Android SDK 根目录执行打包脚本，烧写的时候请烧写打包后产生的新的 boot.img 和 recovery.img。

By default, this command will pack the images in ./tools/images/ directory into resource.img as charging images. New resource.img will be generated in the root directory of UBoot. Please copy the resource.img back to kernel directory and execute the package script in the root directory of Android SDK. When flashing, please flash the new boot.img and recovery.img generated by package.

如下是打包时的提示信息：

Below is the package information:

./scripts/pack_resource.sh /home/guest/3399/kernel/resource.img

Pack ./tools/images/ & /home/guest/3399/kernel/resource.img to resource.img ...

Unpacking old image(/home/guest/3399/kernel/resource.img):

rk-kernel.dtb logo.bmp logo_kernel.bmp

Pack to resource.img successed!

Packed resources:

rk-kernel.dtb battery_1.bmp battery_2.bmp battery_3.bmp battery_4.bmp battery_5.bmp

battery_fail.bmp logo.bmp logo_kernel.bmp battery_0.bmp

resource.img is packed ready

4.5.2 DTS 使能充电 Enalbe charging logo display in dts

默认代码已经使能了该驱动，通过在 dts 里增加并且使能 charge-animation 节点即可使能充电动画的功能。

The default code already enables the driver by adding and enabling charge-animation node in dts.

charge-animation {

compatible = "rockchip,uboot-charge";

status = "okay";

rockchip,uboot-charge-on = <0>; // 是否在 U-Boot 进行充电 Enable U-Boot charging or not

rockchip,android-charge-on = <1>; // 是否在 Android 进行充电 Enable Android charging or not

rockchip,uboot-exit-charge-level = <5>; // U-Boot 充电时，允许开机的最低电量 The minimum power level allowed to U-Boot charge

rockchip,uboot-exit-charge-voltage = <3650>; // U-Boot 充电时，允许开机的最低电压 The minimum voltage allowed to exit U-Boot charge

rockchip,screen-on-voltage = <3400>; // U-Boot 充电时，允许点亮屏幕的最低电压 The minimum voltage allowed to display

rockchip,uboot-low-power-voltage = <3350>; // U-Boot 无条件强制进入充电模式的最低电压 The minimum voltage forcedly entering U-Boot charging mode

rockchip,system-suspend = <1>; // 灭屏时进入 trust 进行低功耗待机 Enter trust for suspend mode with low power consumption

rockchip,auto-off-screen-interval = <20>; // 亮屏超时后自动灭屏，单位秒。(如果没有这个属性，则默认 15s) Automatically turn off the screen after the set time, the unit is second (default is 15s if not defined)

rockchip,auto-wakeup-interval = <10>; // 休眠自动唤醒时间，单位秒。(如果值为 0 或没有这个属性，则禁止休眠自动唤醒) Auto wakeup interval after suspend, the unit is second (if the value is 0 or not defined, auto wakeup is disabled).

rockchip,auto-wakeup-screen-invert = <1>; // 休眠自动唤醒的时候，是否让屏幕产生亮/灭效果}; Whether to display the screen or not when auto wakeup from suspend.

自动休眠唤醒功能的作用：

The purpose of auto suspend/resume:

1. 考虑到有些电量计（比如 adc）需要定时更新软件算法，否则会造成电量统计不准，因此不能让 cpu一直处于休眠状态；

Some fuel gauges (such as adc) need to periodically update software algorithm to calibrate the battery, so cpu cannot be always in suspend mode.

2. 方便进行休眠唤醒的压力测试；

Convenient for suspend/resume stress test.

4.5.3 低功耗休眠 Low power standby

进入充电流程后可通过短按 power 实现系统亮灭屏。非低电状态下，长按 power 可退出充电流程进行开机。

Short press “Power” button can turn on/off the screen during charging. If the device is not in low battery state, long press “Power” button can power up.

4.5.4 更换充电图片 Replace the charging picture

1. 更换./tools/images/目录下的图片，图片采用 8bit 或 24bit bmp 格式。使用命令“ls |sort”确认图片排列顺序是低电量到高电量，在使用 pack_resource.sh 脚本打包时，所有图片会按照这个顺序被打包进 resource；

Replace the pictures in the ./tools/images/ directory. The pictures are using 8bit or 24bit bmp format. Use the command "ls |sort" to confirm that the image is sorted from low to high. When packaged with the pack_resource.sh script, all the images will be packaged into the resource in this order.

2. 修改./drivers/power/charge_animation.c 里的图片和电量关系信息：

Modify the picture and battery relationship information in ./drivers/power/charge_animation.c.

name：图片的名字；

name: the name of the picture.

soc：图片对应的电量；

soc: the power corresponding to the picture.

period：图片刷新时间（单位：ms）；

period: the refresh time of the picture (unit: ms).

注意：最后一张图片一定要是 failed 的图片，且“soc=-1”不可改变。

**Note: **The last picture must be failed picture, and "soc=-1" cannot be changed.

3. 执行 pack_resource.sh 打包命令获取新的 resource.img 即可；

Execute the pack_resource.sh package command to acquire the new resource.img.

5 Kernel 开发 Kernel development

本节简单介绍内核一些常见配置的修改，主要是 dts 的配置，帮助客户更快更方便的进行一些简单的修改。RK3399Pro kernel 版本是 4.4，config 配置文件统一为 arch/arm64/configs/ rockchip_defconfig，RK3399Pro 的串口波特率为 1500000，调试时请保证设置准确。

This chapter simply introduces some kernel common configurations changes, mainly for dts configuration, to help customers to do some simple changes easier and more convenient. RK3399Pro kernel version is 4.4 and config files are unified as arch/arm64/configs/ rockchip_defconfig. RK3399Pro serial port baud rate is 1500000. Please make sure the setting correct for debugging.

5.1 DTS 介绍 DTS introduction

5.1.1 DTS 说明 DTS introduction

RK3399Pro 的 dts 文件在 kernel/arch/arm64/boot/dts/rockchip/下，其中 rk3399pro.dtsi 是核心配置文件定义了平台相关的内容；RK3399-android.dtsi 是产品级配置文件定义了一些外围设备；具体的产品 dts 需要 include 这两个文件，而在新的 Android9.0 平台，dts 中又需要声明启动介质，因此可以新增加一个 dts 声明启动介质及启动参数，而将以前调试好的 dts 改为 dtsi，新增加一个-avb 的 dts 并 include 刚刚的 dtsi 文件，如 RK3399Pro evb 的 dts 文件为 rk3399pro-evb-v10-avb.dts，具体产品的配置则为 rk3399pro-evb-v10.dtsi。产品的 dtsi 里面根据具体的产品需求配置 CPU、GPU、DDR 的频

率和电压表；配置 io、屏、wifi、bt、sensor、温控、背光、电池、系统供电配置等等。

RK3399Pro dts file is in kernel/arch/arm64/boot/dts/rockchip/. Rk3399pro.dtsi is the core configuration file which defines the platform related contents. RK3399-android.dtsi is the product level configuration file which defines some peripheral devices. The specific product dts needs to include these two files. But on new Android9.0 platform, dts also needs to declare boot device, so you can add a dts to declare “boot device” and “boot args”, change the previous dts to dtsi, and newly add a dts with –avb and include the new dtsi file, e.g. RK3399Pro evb dts file is rk3399pro-evb-v10-avb.dts and the specific configuration is rk3399pro-evb-v10.dtsi. Configure CPU, GPU, DDR frequency and voltage table, configure io, panel, wifi, bt, sensor, thermal control, backlight, battery, system power configuration etc. in the product dtsi according to the detailed product requirement.

硬件版本与软件配置说明 Hardware version and software configuration instructions

RK official development board	Dts	Npu communication mode
RK_EVB_RK3399PRO_XXX_V10	rk3399pro-evb-v10-avb.dts	Usb2.0+Usb3.0
RK_EVB_RK3399PRO_XXX_V11\V12	rk3399pro-evb-v11-avb.dts	Usb2.0+Usb3.0
RK_EVB_RK3399PRO_XXX_V13	rk3399pro-evb-v13-multi-cam-avb.dts	Usb2.0+PCIE

注意：

RK_EVB_RK3399PRO_XXX_V13，camera 默认启动 N4 AHD camera，无法识别 usb camera

RK_EVB_RK3399PRO_XXX_V13, camera starts N4 AHD camera by default, usb camera cannot be recognized

若不使用 AHD camera 需要做一下修改：

If you do not use AHD camera, you need to make some modifications:

+++ b/arch/arm64/boot/dts/rockchip/rk3399pro-evb-v13-multi-cam-avb.dts

&usbacm_video_control {

- status = "okay";

+ status = "disabled";

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

Rk3399Pro 的产品 dts 文件需放在 kernel/arch/arm64/boot/dts/rockchip/下。

RK3399Pro product dts file should be put in kernel/arch/arm64/boot/dts/rockchip/.

1、 以 rk3399pro-evb-v10.dtsi 为参照，拷贝一份 dts 文件命名为 rk3399Pro-product.dtsi。

Take rk3399pro-evb-v10.dtsi as reference, copy a dts file and name it as rk3399Pro-product.dtsi.

2、 以 rk3399pro-evb-v10-avb.dts 为参照，拷贝一份 dts 文件命名为 rk3399Pro-product-avb.dts, 修改其中的 include 部分为新增的 rk3399Pro-product.dtsi

Take rk3399pro-evb-v10-avb.dts as reference, copy a dts file, name it as rk3399Pro-product-avb.dts, and replace the include part with newly added rk3399Pro-product.dtsi.

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

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

```
+rk3399Pro-product.dtb
```

4、 修改编译脚本或编译命令。

Modify the compiling script or command.

5、 重新编译内核。

Re-compile kernel.

5.2 WiFi 配置 Wi-Fi configuration

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

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

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

wifi_chip_type = " ap6354";

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

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

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

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

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

WIFI,host_wake_irq = <&gpio0 3 GPIO_ACTIVE_HIGH>;

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

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

Broadcom wifi AP6xxx 系统会使用此中断脚作为 WiFi 数据中断脚，此中断脚有异常将会导致 WiFi 无法正常工作。其它 WiFi，例如 RTL8723BS，在机器进入休眠时，如果有 WiFi 数据到来时此中断用来唤醒机器。此中断脚有异常并不会造成 WiFi 无法正常工作。

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

5.3 BT 配置 BT configuration

```
wireless-bluetooth {
        compatible = "bluetooth-platdata";
        //wifi-bt-power-toggle;
        uart_rts_gpios = <&gpio2 19 GPIO_ACTIVE_LOW>; /* GPIO2_C3 */
        pinctrl-names = "default", "rts_gpio";
```

```
        pinctrl-0 = <&uart0_rts>;
        pinctrl-1 = <&uart0_gpios>;
        //BT,power_gpio    = <&gpio3 19 GPIO_ACTIVE_HIGH>; /* GPIOx_xx */
        BT,reset_gpio       = <&gpio0 9 GPIO_ACTIVE_HIGH>; /* GPIO0_B1 */
        BT,wake_gpio         = <&gpio2 26 GPIO_ACTIVE_HIGH>; /* GPIO2_D2 */
        BT,wake_host_irq = <&gpio0 4 GPIO_ACTIVE_HIGH>; /* GPIO0_A4 */
        status = "okay";
    };
```

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

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

BT,reset_gpio = <&gpio0 9 GPIO_ACTIVE_HIGH>;

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

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

BT,power_gpio = <&gpio3 19 GPIO_ACTIVE_HIGH>

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

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

BT,wake_gpio = <&gpio2 26 GPIO_ACTIVE_HIGH>;

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

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

BT,wake_host_irq = <&gpio0 4 GPIO_ACTIVE_HIGH>

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

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

默认 BT 使用 uart0 接口连接，uart0 的配置如下：

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

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

5.4 GPIO

RK3399Pro 提供 5 组 GPIO(GPIO0~GPIO4)共 122 个，所有的 GPIO 都可以用作中断，GPIO0/GPIO1 可以作为系统唤醒脚，所有 GPIO 都可以软件配置为上拉或者下拉，所有 GPIO 默认为输入，GPIO 的驱动能力软件可以配置。

RK3399Pro provides 5 groups GPIO(GPIO0~GPIO4) total 122pcs. All GPIO can be used as interrupt. GPIO0/GPIO1 can be used as system wakeup pin. All GPIO can be pulled up or down by software configuration. All GPIO default is input and the driver ability can be configured by software.

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

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

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

GPIO usage refers to 《Rockchip Pin-Ctrl 开发指南 V1.0-20160725.pdf》 in the directory of RKDocs\common\driver\.

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

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

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

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

CPUFreq is a set of framework model supporting dynamically adjusting CPU frequency and voltage

defined by kernel developers. It can effectively lower down CPU power consumption and balance CPU performance at the same time.

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

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

- interactive：根据 CPU 负载动态调频调压；

  interactive: dynamically adjust frequency and voltage according to CPU load.

- conservative：保守策略，逐级调整频率和电压；

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

- ondemand：根据 CPU 负载动态调频调压，比 interactive 策略反应慢；

  ondemand: dynamically adjust frequency and voltage according to CPU load, slower than interactive.

- userspace：用户自己设置电压和频率，系统不会自动调整；

  userspace: user to set voltage and frequency, system doesn't automatically adjust.

- powersave：功耗优先，始终将频率设置在最低值；

  powersave: power consumption first, always set the frequency to the lowest value.

- performance：性能优先，始终将频率设置为最高值；

  performance: performance first, always set the frequency to the max value.

详细的模块功能及配置，请参考 RKDocs/common/driver/目录下《Rockchip CPU-Freq 开发指南 V1.0.1-20170213.pdf》和《Rockchip DEVFreq 开发指南 V1.0-20160701.pdf》文档。

The detailed module function and configuration refer to 《Rockchip CPU-Freq 开发指南 V1.0.1-20170213.pdf》 and 《Rockchip DEVFreq 开发指南 V1.0-20160701.pdf》 in the directory of RKDocs/common/driver/.

A53/A72/GPU/DDR 分别有对应的调试接口，可以通过 ADB 命令进行操作，对应的接口目录如下：

A53/A72/GPU/DDR all have corresponding debugging interface which can be operated with ADB command. The corresponding interface contents are as below:

A53：/sys/devices/system/cpu/cpu0/cpufreq/

A72: /sys/devices/system/cpu/cpu4/cpufreq/

GPU: /sys/class/devfreq/ff9a0000.gpu/

DDR: /sys/class/devfreq/dmc/

这些目录下有如下类似节点：

These contents have below similar nodes:

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

以 GPU 为例进行定频操作，流程如下：

Take GPU as example to do the fixed frequency operation. The process is as below:

- 查看支持哪些频率

  Check the supported frequencies

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

- 切换变频策略

  Switch the frequency scaling strategy

  echo userspace > /sys/class/devfreq/ff9a0000.gpu/governor

- 定频

  Fix the frequency

  echo 400000000 > /sys/class/devfreq/ff9a0000.gpu/userspace/set_freq

- 设置完后，查看当前频率

  Check current frequency after setting

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

5.6 温控配置 Thermal control configuration

RK3399Pro 芯片的 ARM 核和 GPU 核分别带有温控传感器，可以实时监控 cpu 和 gpu 的温度，并通过算法来控制 cpu 和 gpu 的频率从而控制 cpu 和 gpu 的温度。每个产品的硬件设计和模具不同对应的散热情况也不同，可以通过 dts 中的如下配置进行适当的调整温控参数来适配产品：

RK3399Pro chipset ARM core and GPU core have separate thermal control sensors which can real-time monitor CPU and GPU temperature and then control CPU and GPU temperatures by controlling CPU and GPU frequency through algorithm. Each product's different hardware design and mold

correspond to different heat dissipation situation. The following configurations in dts can be used to adjust thermal control parameters to fit the product:

设置温控开启的温度：

Set the temperature to enable the thermal control:

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

设置温控上限温度：

Set the upper limit of thermal control temperature:

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

设置软件关机温度：

Set the software shutdown temperature:

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

配置硬件关机温度：

Configure the hardware shutdown temperature:

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

温控的具体说明可以参考 RKDocs\common\driver 目录下《Rockchip Thermal 开发指南 V1.0.1-20170428.pdf》。

The detailed thermal control instruction refers to 《Rockchip Thermal 开发指南 V1.0.1-20170428.pdf》 in the directory of RKDocs\common\driver.

5.7 LPDDR4 配置 LPDDR4 configuration

rk3399Pro 使用 lpddr4 的 dts 配置请参考文件：arch/arm64/boot/dts/rockchip/rk3399pro-evb-lp4-v11-avb.dts，将该文件中的下述 3 个节点拷贝到对应的产品 dts 中即可：

rk3399Pro lpddr4 dts configuration refers to the file: arch/arm64/boot/dts/rockchip/rk3399pro-evb-lp4-v11-avb.dts. Just need to copy below three nodes in the file to the corresponding product dts:

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

&dmc {
        status = "okay";
        center-supply = <&vdd_center>;//这里需要客户根据实际硬件电路来配置 here need customers to configure according to the actual hardware circuits
        system-status-freq = <
                /*system status          freq(KHz)*/
                SYS_STATUS_NORMAL        800000
                SYS_STATUS_REBOOT        400000
                SYS_STATUS_SUSPEND       400000
                SYS_STATUS_VIDEO_1080P   400000
                SYS_STATUS_VIDEO_4K      800000
                SYS_STATUS_VIDEO_4K_10B  800000
                SYS_STATUS_PERFORMANCE   800000
                SYS_STATUS_BOOST         400000
                SYS_STATUS_DUALVIEW      800000
                SYS_STATUS_ISP           800000
        >;
        auto-min-freq = <400000>;
        auto-freq-en = <0>;
};
&dmc_opp_table {
        compatible = "operating-points-v2";

        opp-200000000 {
                opp-hz = /bits/ 64 <200000000>;
                opp-microvolt = <825000>;
                status = "disabled";
```

```
            };
            opp-300000000 {
                    opp-hz = /bits/ 64 <300000000>;
                    opp-microvolt = <850000>;
                    status = "disabled";
            };
            opp-400000000 {
                    opp-hz = /bits/ 64 <400000000>;
                    opp-microvolt = <900000>;
            };
            opp-528000000 {
                    opp-hz = /bits/ 64 <528000000>;
                    opp-microvolt = <900000>;
                    status = "disabled";
            };
            opp-600000000 {
                    opp-hz = /bits/ 64 <600000000>;
                    opp-microvolt = <900000>;
                    status = "disabled";
            };
            opp-800000000 {
                    opp-hz = /bits/ 64 <800000000>;
                    opp-microvolt = <900000>;
            };
            opp-928000000 {
                    opp-hz = /bits/ 64 <928000000>;
                    opp-microvolt = <900000>;
                    status = "disabled";
            };
            opp-1056000000 {
                    opp-hz = /bits/ 64 <1056000000>;
                    opp-microvolt = <900000>;
                    status = "disabled";
            };
    };
```

这里需要注意的是，1）lpddr4 我们只支持 416M 和 856M 两档频率，其他频率被 disabled 掉了，所以如果客户要使用同一个 dts 来支持 lpddr4 和其他类型的 ddr，则其他类型的 ddr 也将只有 416M 和 856M 的频率，这个请务必注意；2）以上配置默认开启 DDR 变频功能。lpddr4 的变频功能对声卡的数量有所限制，说明如下：

Here we need to pay attention to that: 1) lpddr4 only supports 416MHz and 856MHz, other frequencies are disabled. If customers want to use the same dts to support lpddr4 and other ddr, other ddr will also only support 416MHz and 856MHz. Please pay attention to this. 2) Above configuration enables DDR frequency scaling function by default. Lpddr4 frequency scaling function has some limitation on the audio card number as described below:

如果 lpddr4 需要变频功能，则需要将音频 buffer 移到 sram 中，RK3399Pro 的 sram 空间有限，可用空间 128k，目前预分配给单个音频流的空间为 32k，所以系统支持的上限声卡数最多只能 2 个（32k * 2 * 2，每个声卡包含 playback 和 capture），更多的声卡无法创建成功，除非减小单个流的预分配大小，但这也相对的减小了底下支持的 buffer size max,如果用户层使用声卡想设置更大 buffer 时将受限。需注意，USB 声卡由于未使用 dma，所以不在限制范围内，也就是说，可以有 2 个声卡（包含 hdmi、spdif、i2s 等接口的声卡）加上多个 usb 声卡。因此，接下来分成两种情况描述：

If lpddr4 needs frequency scaling function, need to transfer audio buffer to sram. RK3399Pro sram space is limited, available space is 128k, currently pre-allocated space for single audio stream is 32k, so the system can support only 2 audio card at most(32k*2*2, each audio card includes playback and capture). More audio cards cannot be created successfully unless to decrease the single stream pre-allocated size. However, it also relatively decreases the buffer size max supported by bottom layer and it will be limited if user layer wants to set a larger buffer for audio card. Need to notice that, USB audio card is not subject to the limitation because it doesn't use dma. That means, you can use two audio cards (audio cards with hdmi, spdif, i2s etc. interfaces) and multiple USB audio cards. Therefore, the following description is divided into two cases:

5.7.1 需要 lpddr4 的变频 Need lpddr4 frequency scaling

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

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

1. dts 中添加 sram 节点

   Add sram node in dts

```
/* first 64k(0xff8c0000~0xff8d0000) for ddr and suspend */
iram: sram@ff8d0000 {
```

```
        compatible = "mmio-sram";
        reg = <0x0 0xff8d0000 0x0 0x20000>; /* 128k */
    };
```

2. 相对应的产品 dts 中引用 iram 节点。

   Invoke iram node in the corresponding product dts.

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

5.7.2 不需要 lpddr4 变频 Do not need lpddr4 frequency scaling

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

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

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

另外，需要确保在内核中删除掉 5.8.1 节中描述的 2 个配置：

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

1. 删除 dts 中的如下配置：

   Delete the following configuration in dts:

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

2. 删除 dts 中的如下配置：

   Delete the following configuration in dts:

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

5.8 SDCard 配置 SD Card configuration

Uart debug 与 sdcard 复用，默认配置是打开 debug，如果要使用 sdcard 需要如下配置：

Uart debug and sdcard are reused. The default configuration is to enable debug. If want to use sdcard, need to configure as below:

```
 &fiq_debugger {
+               status = "disabled";
                pinctrl-0 = <&uart2a_xfer>;
 };
&sdmmc {
                sd-uhs-sdr12;
                sd-uhs-sdr25;
                sd-uhs-sdr50;
                sd-uhs-sdr104;
+               status = "okay";
   };
```

6 Android 常见配置 Android common configuration

6.1 Android 产品配置 Android product configuration

6.1.1 lunch 选项说明 lunch option description

rk3399pro-userdebug: //rk3399Pro 平台产品 userdebug（64 位）rk3399Pro platform product userdebug (64bit)

rk3399pro-user: //rk3399Pro 平台产品 user（64 位）rk3399Pro platform product user (64bit)

rk3399pro_pcie-userdebug: //rk3399Pro 平台产品 userdebug（64 位）rk3399Pro platform

product userdebug (64bit)

rk3399pro_pcie-user: //rk3399Pro 平台产品 user（64 位）rk3399Pro platform product user (64bit)

注：pcie 表示 3399pro 与 npu 数据通信的方式，对应官方 evb v13.

Note: pcie means the mode of data communication between 3399pro and npu, corresponding to the official evb v13.

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

6.2.1 常用配置宏说明 Common configuration macro instruction

宏配置 Macro configuration	功能说明 Function description
BUILD_WITH_GOOGLE_MARKET	若为 true 则集成 GMS 包，false 不集成 If it is true, integrate GMS package, false not to integrate
BUILD_WITH_GOOGLE_MARKET_ALL	若为 true 集成 full 的 GMS 包，false 集成 mini 的 GMS 包 If it is true, integrate full GMS package, false to integrate mini GMS package
BUILD_WITH_GOOGLE_FRP	使能恢复出厂设置保护 FRP 功能 Enable FRP factory reset protection function
BOARD_AVB_ENABLE	使能 AVB (Android Verified Boot 2) Enable AVB (Android Verified Boot 2)
BOARD_USES_AB_IMAGE	使能 AB 升级功能 Enable AB upgrading function
BUILD_WITH_WIDEVINE	集成 Widevine level3 插件库 Integrate Widevine level3 plugin library
BOARD_NFC_SUPPORT	使能 NFC 功能 Enable NFC function
BOARD_SENSOR_ST	选用 ST 的 sensor 框架 Select ST sensor framework
BOARD_SENSOR_MPU	选用 MPU 的 sensor 框架 Select MPU sensor framework

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

6.2.2 预装 APK Pre-install APK

Android 上的应用预安装功能，主要是指配置产品时，根据厂商要求，将事先准备好的第三方应用预制进 Android 系统。预安装分为不可卸载安装、可永久卸载安装以及卸载后恢复出厂设置后自动恢复安装，详细配置和使用请参阅工程目录 RKDocs/android/下相关说明文档：

《Rockchip_Introduction_Android_Application_Preinstallation_CN.pdf》。

Android apk pre-install function means to install the third application prepared in advance into the Android system when configuring the product according to customer requirements. Pre-install can be divided into non-uninstall installation, permanent uninstall installation and automatic installation after factory reset. Please refer to below document in the project directory of RKDocs/android/ for the detailed configuration and usage: 《Rockchip_Introduction_Android_Application_Preinstallation_CN.pdf》.

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

定制 Android9.0 的开机铃声，关机铃声，开机动画，关机动画的详细方法请参阅工程目录 RKDocs/android/下的说明文档：《Rockchip_Introduction_Android8.0_Power_On_Off_Animation_and_Tone_Customization_CN.pdf》。

Android9.0 power on tones, power off tones, power on animation, and power off animation customizations refer to the document 《Rockchip_Introduction_Android8.0_Power_On_Off_Animation_and_Tone_Customization_CN.pdf》 in the project directory of RKDocs/android/.

6.3 Parameter 说明 Parameter instruction

rk3399Pro Android 9.0 平台有不同产品形态，不同的产品形态可能需要不同的 parameter 参数，关于 parameter 中各个参数、分区情况细节，请参考\RKDocs\common\RKTools manuals\ Rockchip Parameter File Format Ver1.3.pdf。

rk3399Pro Android9.0 platform supports various product types and different product types may need different parameter. For the parameter and partition details, please refer to \RKDocs\common\RKTools manuals\ Rockchip Parameter File Format Ver1.3.pdf.

6.4 新增分区配置 New partition configuration

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

Please refer to \RKDocs\android\《Android 增加一个分区配置指南 V1.00.pdf》.

6.5 OTA 升级 OTA upgrade

OTA （over the air）升级是 Android 系统提供的标准软件升级方式。它功能强大，提供了完全升级（完整包）、增量升级模式（差异包），可以通过本地升级，也可以通过网络升级。详细的 OTA 升级及 Recovery 模块功能及配置，请参考 RKDocs\android 目录下《Rockchip_User_Guide_Recovery_CN&EN.pdf》。

OTA (over the air) upgrade is the standard software upgrade method provided by Android system. It provides complete upgrading (full package) and incremental upgrading mode (difference package). You can

upgrade locally or over the network. For the detailed OTA upgrade and Recovery mode function and configuration, please refer to 《Rockchip_User_Guide_Recovery_CN&EN.pdf》 in the directory of RKDocs\android.

如果需要使用 AB 升级(也称 静默升级)，需要同时烧写两套固件，能够在开机时进行升级另外的一套固件，升级成功后系统重启后直接进入升级后的系统，如果要使用静默升级，请参考 RKDocs\android 目录下的：Rockchip_Introduction_Android_AB_System_Upgrading_CN.pdf

If need to use AB upgrading (also called silent upgrading), need to flash two set of images at the same time, which allows to upgrade the other set of image when power on. After upgrading, it will directly enter the new system after reboot. If want to use silent upgrading, please refer to Rockchip_Introduction_Android_AB_System_Upgrading_CN.pdf in the directory of RKDocs\android.

7 系统调试 System debug

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

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

7.1 ADB 工具 ADB tool

7.1.1 概述 Overview

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

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

- 运行设备的 shell（命令行）

  Run the device shell (command line)

- 管理模拟器或设备的端口映射

  Manage the port mapping of the simulator or the device

- 计算机和设备之间上传/下载文件

  Upload/download files between the computer and the device

- 将本地 apk 软件安装至模拟器或 Android 设备

Install the local apk to simulator or Android device

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

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

- 网络 ADB：主机通过有线/无线网络（同一局域网）连接到 STB 设备

  Network ADB: PC connects to STB device through cable/wireless network.

- USB ADB：主机通过 USB 线连接到 STB 设备

  USB ADB: PC connects to STB device through USB cable.

7.1.2 USB ADB 使用说明 USB ADB usage

USB ADB 使用有以下限制：

USB ADB usage has below limitations:

- 只支持 USB OTG 口

  Only support USB OTG port

- 不支持多个客户端同时使用（如 cmd 窗口，eclipse 等）

  Not support multiple clients at the same time (such as cmd window, eclipse etc.)

- 只支持主机连接一个设备，不支持连接多个设备

  Support host connects to only one device but multiple devices

连接步骤如下：

The connection steps are as below:

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

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

2、 PC 主机只通过 USB 线连接到机器 USB OTG 口，然后电脑通过如下命令与设备相连。

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

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

number shows up, the connection is successful.

7.1.3 网络 ADB 使用要求 Network ADB use requirement

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

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

如果你需要使用网络 ADB 来调试设备，必须要满足如下条件：

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

1、 设备上面首先要有网口，或者通过 WiFi 连接网络。

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

2、 设备和研发机（PC 机）已经接入局域网，并且设备设有局域网的 IP 地址。

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

3、 要确保研发机和设备能够相互 ping 得通。

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

4、 研发机已经安装了 ADB。

PC already installs ADB.

5、确保 Android 设备中 adbd 进程（ADB 的后台进程）已经运行。adbd 进程将会监听端口 5555 来进行 ADB 连接调试。

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

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

SDK 默认未对网络 ADB 端口进行配置，需要手动修改打开配置。

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

修改 device/rockchip/rkxxxx/device.mk 文件，在 PRODUCT_PROPERTY_OVERRIDES 后面追加如下配置：

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

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

7.1.5 网络 ADB 使用 Network ADB usage

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

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

1、 首先 Android 设备需要先启动，如果可以话，可以确保一下 adbd 启动(ps 命令查看)。

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

2、 在 PC 机的 cmd 中，输入：

In PC cmd, input:

```
adb connect 192.168.1.5:5555
```

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

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

```
adb kill-server
```

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

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

4、 调试完成之后，在研发机上面输入如下的命令断开连接：

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

```
adb disconnect 192.168.1.5:5555
```

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

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

If SDK doesn’t add ADB port number configuration, or want to change ADB port number, you can change through below method:

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

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

2、 设置 ADB 监听端口：

Set ADB monitor port:

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

3、 通过 ps 命令查找 adbd 的 pid

Look up adbd pid using ps command.

4、 重启 adbd

Reset adbd.

#kill -9<pid>，这个 pid 就是上一步找到那个 pid This pid is just the one found in last 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, adbd will automatically restart after Android init progress. After adbd restart, if service.adb.tcp.port is set, it will automatically change to monitor network request.

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

（1） 查看设备情况

Check the device situation

查看连接到计算机的 Android 设备或者模拟器：

Check the Android device or simulator connected to computer:

```
adb devices
```

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

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

（2） 安装 APK

Install APK

将指定的 APK 文件安装到设备上：

Install the specific APK file to the device:

```
adb install <apk 文件路径 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 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：

Complete uninstall:

```
adb uninstall <package>
```

示例如下：

For example:

```
adb uninstall com.wishtv
```

（4） 使用 rm 移除 APK 文件：

Use rm to remove APK file:

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

示例如下：

For example:

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

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

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

（5） 进入设备和模拟器的 shell

Enter shell of the device and simulator

进入设备或模拟器的 shell 环境：

Enthe the shell environment of the device or simulator:

```
adb shell
```

（6） 从电脑上传文件到设备

Upload file to the device from computer

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

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

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

示例如下：

For example:

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

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

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

（7） 从设备下载文件到电脑

Download file from the device to computer

pull 命令可以把设备上的文件或者文件夹下载到本机电脑中。

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

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

示例如下：

For example:

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

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

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

（8） 查看 bug 报告

Check bug report

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

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

（9） 查看设备的系统信息

Check the device system information

在 adb shell 下查看设备系统信息的具体命令。

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

```
adb shell getprop
```

7.2 Logcat 工具 Logcat tool

Android 日志系统提供了记录和查看系统调试信息的功能。日志都是从各种软件和一些系统的

缓冲区中记录下来的，缓冲区可以通过 Logcat 来查看和使用。Logcat 是调试程序用的最多的功能。该功能主要是通过打印日志来显示程序的运行情况。由于要打印的日志量非常大，需要对其进行过滤等操作。

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

7.2.1 Logcat 命令使用 Logcat command usage

用 logcat 命令来查看系统日志缓冲区的内容：

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

基本格式：

The basic format:

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

示例如下：

For example:

```
adb shell
logcat
```

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

控制日志输出的几种方式：

Several ways to control the logcat output:

- 控制日志输出优先级。

  Control the logcat output priority.

示例如下：

For example:

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

示例说明：显示优先级为 warning 或更高的日志信息。

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

- 控制日志标签和输出优先级。

Control the logcat label and output priority.

示例如下：

For example:

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

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

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

- 只输出特定标签的日志

  Only output the logcat with the specific label

示例如下：

For example:

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

或者

or

```
adb shell
logcat –s WishTV
```

示例说明：只输出标签为 WishTV 的日志。

Note: only output the logcat with label of WishTV.

- 只输出指定优先级和标签的日志

  Only output the logcat with the specific priority and label

示例如下：

For example:

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

示例说明：只输出优先级为 I，标签为 WishTV 的日志。

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

7.2.3 查看上次 log View last log

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

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

```
adb shell
logcat -L
```

7.3 Procrank 工具 Procrank tool

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

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

包括如下内存信息：

Include below memory information:

- VSS：Virtual Set Size 虚拟耗用内存大小（包含共享库占用的内存）The memory size used by virtual (including the memory used by the shared lib)
- RSS：Resident Set Size 实际使用物理内存大小（包含共享库占用的内存）The actually used physical memory size (including the memory used by the shared lib)
- PSS：Proportional Set Size 实际使用的物理内存大小（比例分配共享库占用的内存）The actually used physical memory size (allocate the memory used by the shared lib in proportion)

- USS：Unique Set Size 进程独自占用的物理内存大小（不包含共享库占用的内存）The physical memory used exclusively by the process (not including the memory used by the shared lib)

注意：Note:

- USS 大小代表只属于本进程正在使用的内存大小，进程被杀死后会被完整回收；
  USS size represents the memory size only used by the process, and it will be recovered completely after the process is killed.
- VSS/RSS 包含了共享库使用的内存，对查看单一进程内存状态没有参考价值；
  VSS/RSS includes the memory used by the shared lib, so it is not helpful to check the memory status of the single process.
- PSS 是按照比例将共享内存分割后，某单一进程对共享内存区的占用情况。
  PSS is the shared memory status used by the specific single process after the shared memory is allocated in proportion.

7.3.1 使用 procrank Use procrank

执行 procrank 前需要先让终端获取到 root 权限

Make sure the terminal has the root authority before executing procrank

```
su
```

命令格式：

The command format:

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

常用指令说明 ：

The commonly used command instructions:

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

示例：For example:

–输出内存快照：Output the memory snapshot:

```
procrank
```

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

```
procrank –v
```

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

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

查看指定进程的内存占用状态，命令格式如下：

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

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

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

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

输出 systemUI 进程的内存占用状态:

Output the memory status used by systemUI process:

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

或者：Or:

```
procrank | grep 3396
```

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

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

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

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

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

1、 编写脚本 test.sh

Write the script test.sh

```
#!/bin/bash
while true;do
```

```
adb shell procrank | grep “com.android.systemui”
sleep 1
done
```

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

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

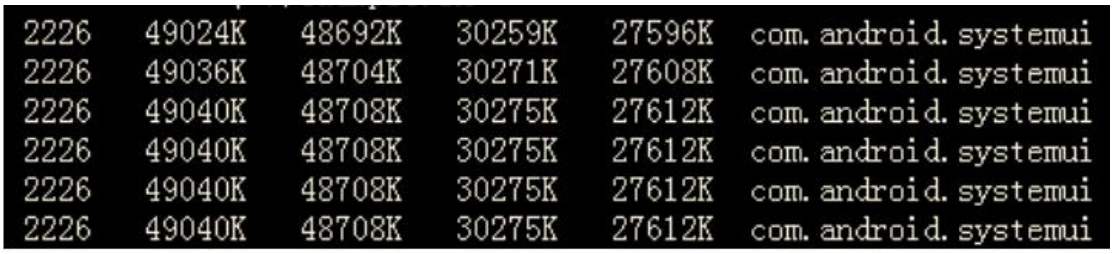

图 0-1 跟踪进程内存状态

Picture 7-1 Trace the process memory status

7.4 Dumpsys 工具 Dumpsys tool

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

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

dumpsys 输出打印的条件：

The conditions for dumpsys to output the print:

1、 只能打印已经加载到 ServiceManager 中的服务；

Only print the services already loaded to ServiceManager.

2、 如果服务端代码中的 dump 函数没有被实现，则没有信息输出。

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

7.4.1 使用 Dumpsys Use Dumpsys

- 查看 Dumpsys 帮助

  View Dumpsys help

作用：输出 dumpsys 帮助信息。

Function: output dympsys help information.

```
dumpsys -help
```

- 查看 Dumpsys 包含服务列表

View the service list of Dumpsys

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

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

```
dumpsys  -l
```

- 输出指定服务的信息

  Output the specific service information

作用：输出指定的服务的 dump 信息。

Function: output the specific service dump information.

格式：dumpsys [servicename]

Format: dumpsys [servicename]

示例：输出服务 SurfaceFlinger 的信息，可执行命令：

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

```
dumpsys SurfaceFlinger
```

- 输出指定服务和应用进程的信息

  Output the specific service and application process information

作用：输出指定服务指定应用进程信息。

Function: output the specific service and application process information

格式：dumpsys [servicename] [应用名]

Format: dumpsys [servicename] [application name]

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

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

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

注意：服务名称是大小写敏感的，并且必须输入完整服务名称。

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

7.5 串口调试 Serial port debugging

7.5.1 串口配置 Serial port configuration

调试过程中最方便的就是串口的输入输出，这里需要注意的是 RK3399 波特率设置为 1500000。RTS/CTS 不要勾选，否则串口无法输入。

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

7.5.2 FIQ 模式 FIQ mode

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

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

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

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

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

请参考 RKDocs\common\driver\ Rockchip Audio 开发指南 V1.1-20170215-linux4.4.pdf。

Please refer to RKDocs\common\driver\ Rockchip Audio 开发指南 V1.1-20170215-linux4.4.pdf.

7.7 Last log 开启 Enable Last log

在 dts 文件里面添加下面两个节点

Add the following two nodes in dts file:

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

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

- 130|root@rk3399:/sys/fs/pstore # ls

  dmesg-ramoops-0 上次内核 panic 后保存的 log。

  dmesg-ramoops-0 Log saved after last kernel panic

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

pmsg-ramoops-0 Log of last user space, android log

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

ftrace-ramoops-0 Print function trace during some period.

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

console-ramoops-0 The kernel log for the last boot of last_log, but only save the log with higher priority than default log level

- 使用方法：Usage method:

  cat dmesg-ramoops-0

  cat console-ramoops-0

  logcat -L （pmsg-ramoops-0） 通过 logcat 取出来并解析 pull out by logcat and parse

  cat ftrace-ramoops-0

8 常用工具说明 Commonly used tool instruction

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

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

8.1 StressTest

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

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

Stresstest 测试工具测试的内容主要包括：

The test items of Stresstest tool mainly include:

模块相关

Module related

- Camera 压力测试：包括 Camera 打开关闭，Camera 拍照以及 Camera 切换。

  Camera stress test: including Camera on/off, Camera taking photo and Camera switch.

- Bluetooth 压力测试：包括 Bluetooth 打开关闭。

  Bluetooth stress test: including Bluetooth on/off.

- WiFi 压力测试：包括 WiFi 打开关闭，（ping 测试以及 iperf 测试待加入）。

  WiFi stress test: including WiFi on/off, (plan to add ping test and iperf test).

非模块相关

Non module related

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

8.2 PCBA 测试工具 PCBA test tool

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

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

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

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

具体 PCBA 功能配置及使用说明，请参考：

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

\RKDocs\common\RKTools manuals\Rockchip PCBA 模块开发指南--20170210.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。本平台 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

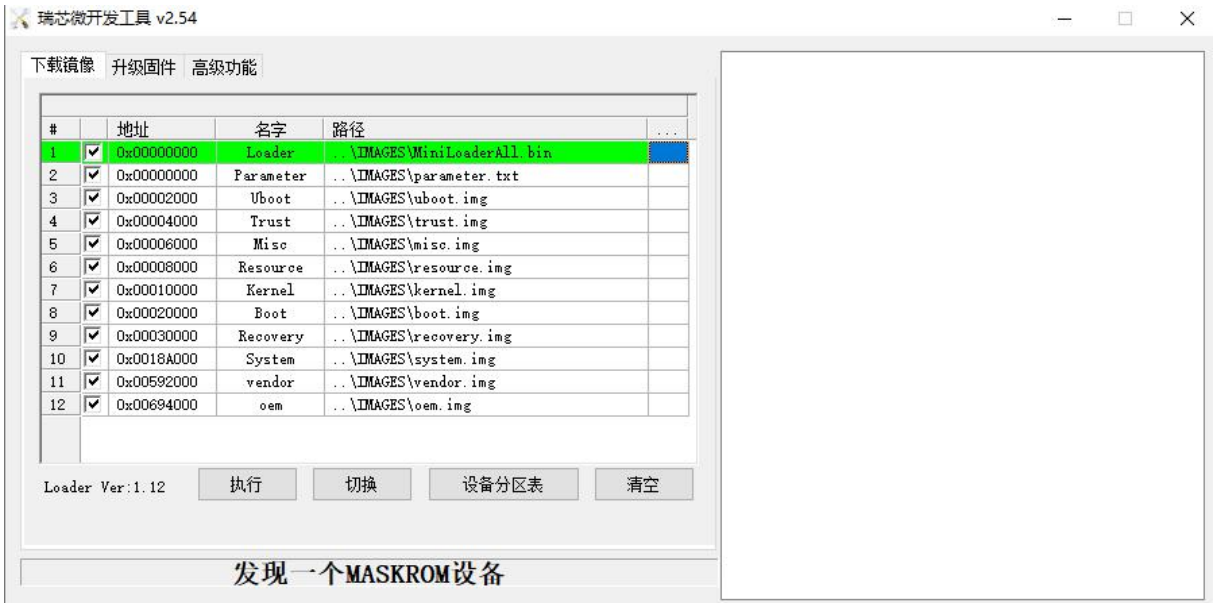

图 8-1 Android 开发工具下载镜像

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

1） 连接开发板进入下载模式。

Connect the development board to enter the download mode.

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

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

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

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

3） 依次设置所有 img 文件的路径。

Set all the img file paths successively.

4） 配置完成后，点击“执行”。右侧信息框将显示相关信息。

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

5） 按钮说明

Button description

“低格”按钮：用于擦除设备

“低格” button: Used to erase the device

“清空”按钮：清空信息框

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

8.4.2 升级固件 Upgrade image

图 8-2 Android 开发工具升级固件

Picture 8-2 Use Android development tool to upgrade image

1) 准备目标固件。（可参考 update.img 打包）

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

2) 确认设备已经进入下载模式。

Confirm the device is already in the download mode.

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

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

3) 点击“固件”按钮，选择目标固件 update.img 文件。

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

4) 点击“升级”按钮进行下载。右侧信息框将显示相关信息。

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

8.4.3 高级功能 Senior functions

图 8-3 Android 开发工具高级功能

Picture 8-3 Android development tool senior functions

高级功能说明：

Senior functions description:

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

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

2) 固件必须现使用打包后的 update.img。

Image must use the packed update.img.

3) 解包功能可将 update.img 拆解为各部分镜像文件。

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

8.5 update.img 打包 update.img pack

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

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

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

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

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

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

3) 编辑 mkupdate.bat。

Compile mkupdate.bat.

4) 修改 loader 名称为实际存放的 loader 名称。

Change loader name to be the one actually saved.

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

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

8.6 固件签名工具 Image signature tool

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

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

8.7 序列号/Mac/厂商信息烧写-RKDevInfoWriteTool 工具 SN/Mac/Vendor information flashing-RKDevInfoWriteTool

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

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

8.7.1 使用 RKDevInfoWriteTool 写入 Use RKDevInfoWriteTool to write

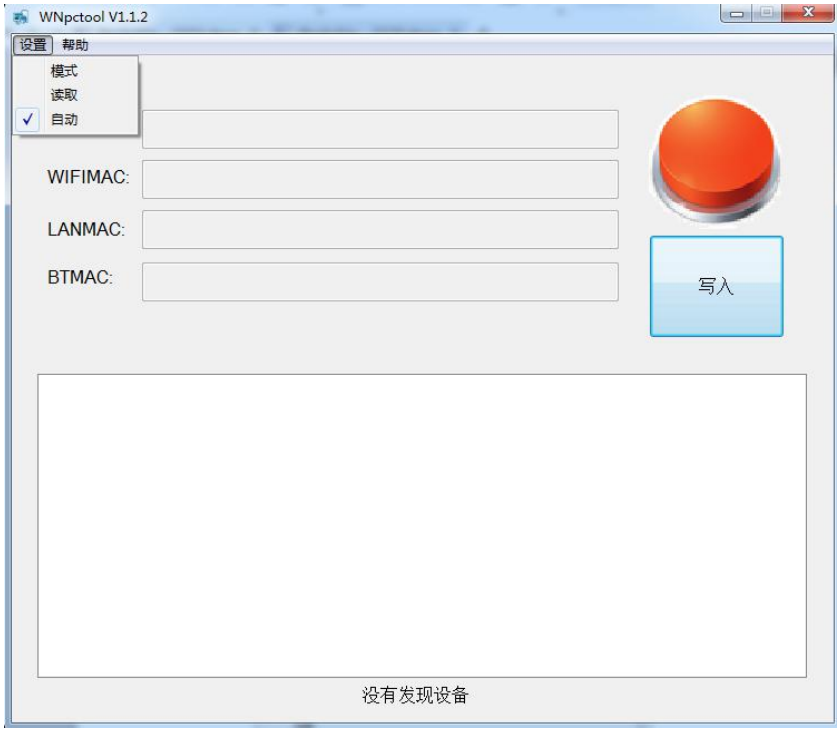

图 8-4 RKDevInfoWriteTool 工具

Picture 8-4 RKDevInfoWriteTool

1) 进入 loader 模式。

Enter loader mode.

2) 点击“设置”菜单，弹出设置窗口，用来设置 SN/WIFI/LAN/BT/IMEI。

Click “setting” menu will pop up the setting window used to set SN/WIFI/LAN/BT/IMEI.

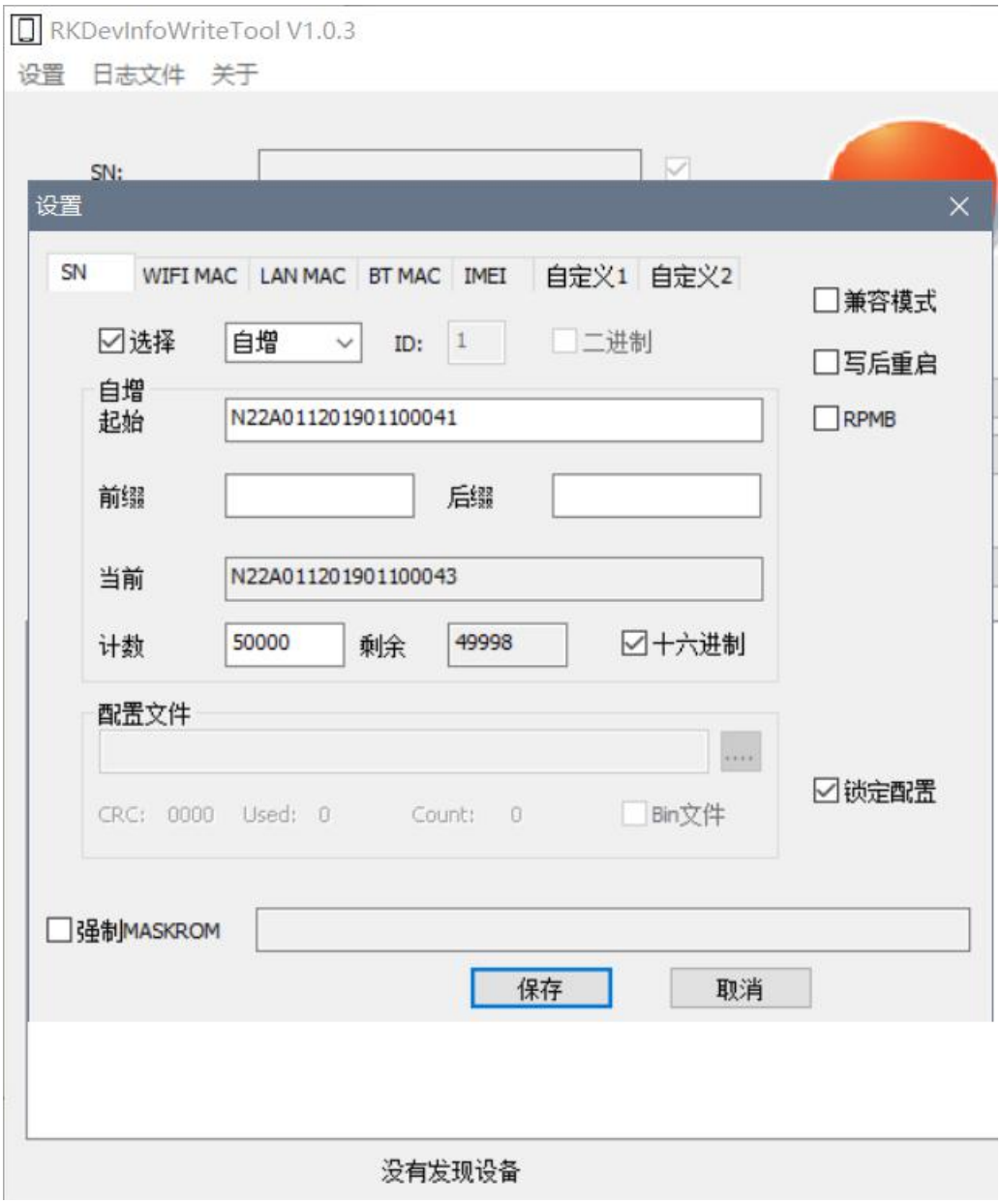

图 8-5 工具模式设置

Picture 8-5 Tool mode setting

3) 设置完成后，点击“保存”按钮，关闭模式设置窗口，返回主窗口。

After setting, click “Save” button, close mode setting window and back to the main window.

4) 点击“写入”按钮即可。

Click “Write” button.

8.7.2 使用 RKDevInfoWriteTool 读取 Use RKDevInfoWriteTool to read

1） 进入 loader 模式。

Enter loader mode.

2） 点击“读取”按钮即可。

Click “Read” button.

8.8 量产工具使用 Production tool usage

8.8.1 工具下载步骤 Tool download steps

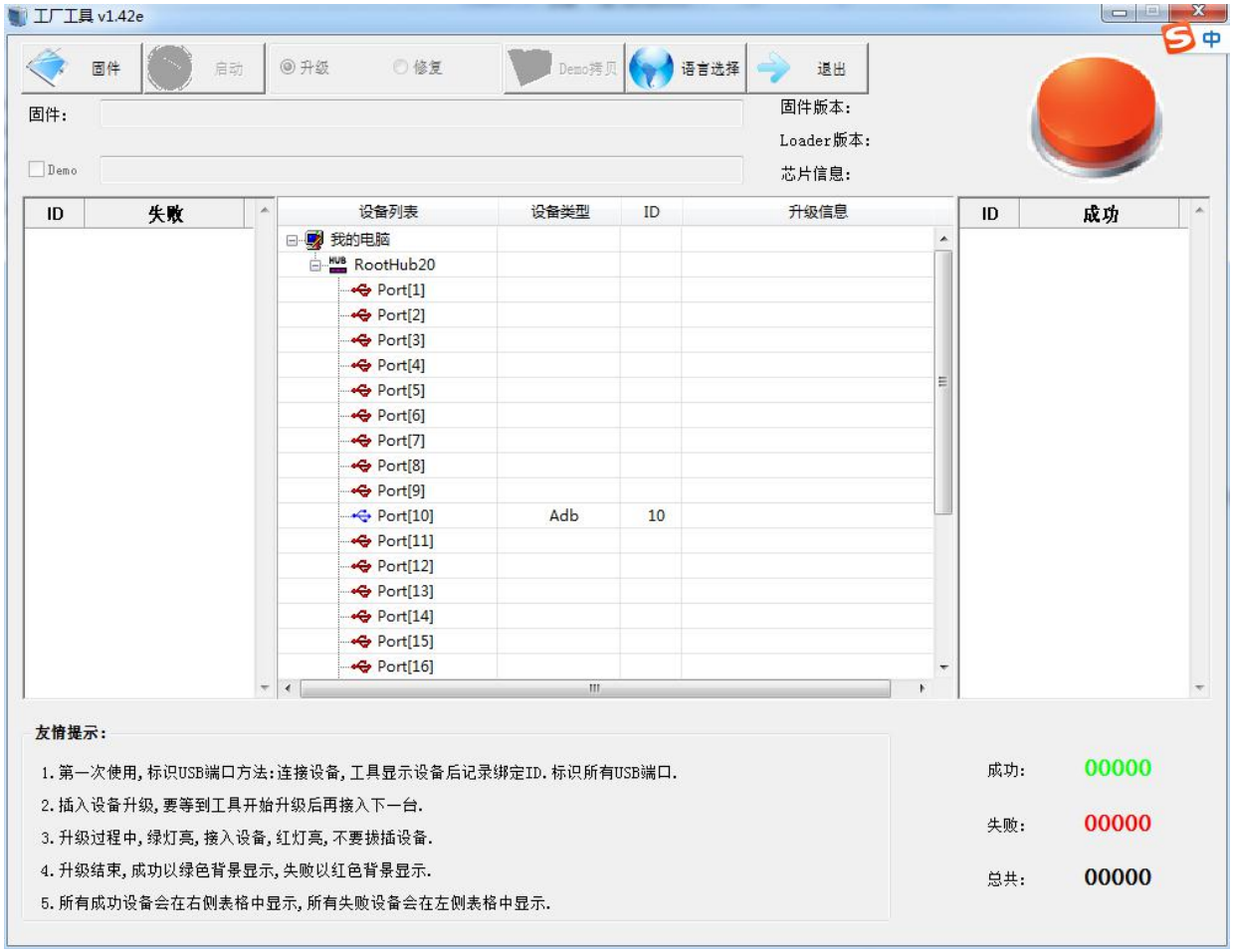

图 8-6 量产工具

Picture 8-8 Production tool

1） 点击固件按钮，选择打包工具打包后的 update.img，等待解包成功。

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

2） 连接设备，并让设备进入 loader 或者 maskrom 模式，工具会自动进行下载。

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

automatically.

3） 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.