

title: Rockchip_Developer_Guide_Android11_SDK_V0.0.2_CN description:

Android11Development Guide published: true

date: 2021-07-13T06:21:56.250Z

tags: sdk

Classification status: Top secret () Secret () Internal () Public (✓)

Rockchip Android 11.0 SDKDevelopment Guide

File status:	
[]draft	
[✓]Officially released	
[]under revision	
File ID:	RK-KF-YF-279
current version:	V1.1.5
author:	Wu Liangqing
Completion Date:	2021-08-31
Review:	Chen Haiyan
Review date:	2021-08-31

Version Number	do By	Revise date	Modify the description	Prepare Note
V0.0.1	Wu good clear	2020-12-25	releaseRK3566/RK3568 AlphaVersion	
V0.0.2	Bian gold morning	2021-01-06	releasePX30/RK3326 BetaVersion	
V1.0.0	Wu good clear	2021-01-29	IncreaseRK3566/RK3568 EVBBoard compilation method	
V1.1.0	Wu good clear	2021-02-23	releaseRK3399 AlphaVersion	
V1.1.1	Wu good clear	2021-03-09	Modify and compile separately kernel instruction of	
V1.1.2	Wu good clear	2021-05-12	supportRK3288WChip platform	
V1.1.3	Wu good clear	2021-05-23	Add FAQ description	
V1.1.4	Wu good clear	2021-07-12	supportRK3566 BOXProduct form, supportRK3328 BOXProduct form, increase repoServer setup and FAQ instructions	
V1.1.5	Wu good clear	2021-08-31	Add FAQ description	

Feedback on documentation issues:wlg@rock-chips.com

Disclaimer

This document is provided "as is" and Rockchip Microelectronics Co., Ltd. ("the Company", the same below) does not assume any responsibility for the accuracy, reliability, completeness, merchantability, or fitness for a particular purpose of any statements, information and content of this document, and non-infringement provided without any representation or warranty, express or implied. This document is only used as a reference for usage guidance.

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

Trademark Statement

"Rockchip", "Rockchip" and "Rockchip" are registered trademarks of our company and are owned by our company. All other registered trademarks or trademarks that may be mentioned in this document are the property of their respective owners.

Beyond the scope of fair use, no unit or individual may excerpt or copy part or all of the contents of this document without the written permission of our company, and shall not disseminate it in any form.

Rockchip Microelectronics Co., Ltd.

Rockchip Electronics Co., Ltd.

address: Tongpan Road Software Park, Fuzhou City, Fujian ProvinceA

URL: district18Number www.rock-chips.com

Customer service phone number: +86-4007-700-590

Customer Service Fax: +86-591-83951833 Customer

service email:fae@rock-chips.com

Rockchip Android 11.0 SDKSupport chip

Chip platform	Whether to support	SDKVersion
RK3566	support	RKR1
RK3568	support	RKR1
PX30/RK3326	support	RKR1
RK3399	support	RKR5
RK3288W	support	RKR7
RK366-BOX	support	RKR9
RK3328	support	RKR9

Rockchip Android 11.0 SDKDevelopment Guide

Disclaimer

Trademark Statement

all rights reserved© 2020Rockchip Microelectronics Co.,

Ltd. Rockchip Android 11.0 SDKSupport chip Rockchip

Android 11.0 SDKCode download and compile

Code download

[download link](#)

[Server image download](#)

Build your ownreopcode server

environment

[gitolitebuild](#)

Server side operation

Client operations

repoMirror building

Server side operation

Client operations

Client operations

Code management

[Switch your own code branch](#)

Code modification submission

[SynchronizeRKcode](#)

code compilation

One-click compilation command

Summary of compilation commands for each platform

Other compilation instructions

[Android11.0Cannot be programmed directlykernel.imgandresource.img Compile](#)

[separatelykernelgenerateboot.img](#)

Firmware programming
Firmware burning tool
Firmware instructions
Firmware instructions

fastbootFlash dynamic partition
useDTBOFunction
Revisefstabdocument
Reviseparameter.txt
AndroidCommon configuration
New productlunch
Kernel dtstillustrate
New productdts

Documentation description
Peripheral support list
Androiddocument
Android_SELinux(Sepolicy)Development Guide Android
11System optimization guidance
Wi-Fidocument
3G/4GModule documentation
Kerneldocument
DDRRelated documents
AudioModule documentation
CRUModule documentation
GMACModule documentation
PCieModule documentation
I2CModule documentation
PIN-CtrlGPIOModule documentation
SPIModule documentation
SensorModule documentation
IO-DomainModule documentation
LedsModule documentation
ThermalTemperature control module documentation
PMICPower Management Module Documentation
MCUModule documentation
Power Consumption and Hibernation Module Documentation
UARTModule documentation
DVFS CPU/GPU/DDRFrequency conversion related
documents EMMC/SDMMC/SDIOModule documentation
PWMModule documentation
USBModule documentation
HDMI-INFFunctional documentation
Security module documentation
ubootIntroduction document
TrustIntroduction document
Cameradocument
Camera IQ ToolDocumentation tool
documentation
PCBADevelopment and usage documentation
Display Driver Debugging Guide
HDMIDebugging Guide
Image displayDRM Hardware Composer(HWC) Problem analysis and troubleshooting DRM
Show development guide
RGAAnalysis and troubleshooting of related issues
Analysis of Frequently Asked Questions about Graphical Display Framework

Tool usage

StressTest

Module related
Not module related

PCBAtest tools

DeviceTest

USBdrive

Develop programming tools

WindowsVersion

LinuxVersion

SDUpgrade startup creation tools

Number writing tool

DDRWelding test tools

efuseProgramming tool

efuse/otpSignature tool

Factory produced firmware programming tools

Firmware modification tool

userdataPartition data preset tool

Camera IQ Tool

system debugging

ADBtool

Overview

USB adbInstructions for use

networkadbRequirements

SDKnetworkadbPort configuration

networkadbuse

Manually modify the networkadbThe port number

ADBDetailed explanation of commonly used commands

Logcattool

LogcatCommand usage

Commonly used log filtering methods

Procranktool

useprocrank

Retrieve specified content information

Track process memory status

Dumpsystool

useDumpsys

Last logturn on

FIQmodel

logAutomatic collection

common problem

currentkernelandu-bootVersion? How to get the currentSDKcorresponding

RK releaseHow to confirm the local versionSDKalready fully updatedRK

PublishedSDKstate ubootandkernelstagelogoPicture replacement RK3566/

RK3568 NAND FLASHConfigure shutdown charging and low battery pre-

charging

BoxMachine standby and fake shutdown function

UbootStage charging picture packaging and replacement

RM310 4GConfiguration

WIFIHibernation policy configuration

Recoveryrotating configuration

Android Surfacerotate

replaceAOSPPartial source coderemote

userdataarea file system toEXT4

Modify the power on and off animation and power on and off ringtones

APPSet performance mode

GPURelated troubleshooting methods

OTPanfuseillustrate

How to determine the device in the codeOTP/EFUSEWhether the switch has been

programmedselinux

"Popup pops up"AndroidThere is a problem with the system" warning

How to open the Ethernet setting item in settings
aboutAVBandsecurity bootOperation IO
Command cannot be used
SNcommand rules
RK3288compiled newspaperLZ4mistake
RK7Previous version updated to RK7 and above version (or other machines with batteries that cannot be turned on after upgrading) reboot loader
Problems such as commands not working
RK356X IO-Domain GPIOVoltage configuration confirmation, GPIOIf the voltage is not configured correctly, it will cause the chip to GPIO
burn out RK356X kernelCompile popupIO-DomainConfirmation dialog RK356x PCIEModule related issues

PCIE3.0port if no external clock is provided to the chip butdtsIf this interface is enabled, it will cause the boot to freeze. rk356x
pcie2x1controller and sata2Controllers cannot be turned on at the same time Android SambaFunction

NFSstart up
ReviseDDRfrequency
Multiple screens, different displays and different touches
Strange sound on multiple screens

appendixA Compilation and development environment setup Compiling and development environment setup

Initializing a Build Environment

Choosing a Branch

Setting up a Linux build environment

Installing the JDK

Configuring USB Access

appendixB SSHPublic key operation instructions SSH public key operation instruction

appendixB-1 SSHPublic key generation SSH public key generation
appendixB-2 usekey-chainManage keys Use key-chain to manage the key Multiple machines use the same ssh
appendixB-3 public key Multiple devices use the same ssh public key One machine switches differently ssh
appendixB-4 public key Switch different ssh public keys on one device Key permission management Key
appendixB-5 authority management
appendixB-6 GitPermission application instructions Git authority application instruction

Rockchip Android 11.0 SDKCode download and compile

Code download

download link

```
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_R/manifests.git -m Android11.xml
```

Server image download

```
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_R/manifests.git -m Android11.xml -- mirror
```

Note, repoyes google use Python Scripted call git A script, mainly used to download and manage Android The software warehouse of the project, its download address is as follows:

```
git clone ssh:// git@www.rockchip.com.cn :2222/repo-release/tools/repo
```

To facilitate customers to quickly obtain SDK source code, Rockchip's technical window usually provides the corresponding version SDK initial compressed package. by

ROCKCHIP_ANDROID11.0_SDK_RELEASE.tar.gz.*

For example, after copying the initialization package, you can check it out through the following command

Source code:

```
mkdir ROCKCHIP_ANDROID11.0_SDK_RELEASE  
cat ROCKCHIP_ANDROID11.0_SDK_RELEASE.tar.gz* | tar -zx -C  
ROCKCHIP_ANDROID11.0_SDK_RELEASE  
cd ROCKCHIP_ANDROID11.0_SDK_RELEASE .repo/  
repo/repo sync -l  
.repo/repo/repo sync -c
```

Build your own repocode server

environment

Install `openssh-server` for remote login, `git` for management projects, `keychain` for public and private key management tools

```
sudo apt-get install openssh-server git keychain
```

gitolitebuild

Server side operation

(With server address: 10.10.10.206 Take an example to illustrate)

1.creategitAccount:

```
sudo adduser --system --shell /bin/bash --group git sudo passwd git
```

2.by "git" Account login server

3.Make sure "~/.ssh/authorized_keys" Empty or does not exist

4.Copy the server administrator's public key to "~/YourName.pub"

5.download gitolite Source code

```
git clone https://github.com/sitaramc/gitolite.git
```

6.exist git Created in user directory bin Table of contents

```
mkdir -p ~/bin
```

7.Execute the following command to install gitolite, different versions have different installation methods, please refer to the documentation in the source code:

```
gitolite/install -to ~/bin
```

8.Set up administrator

```
~/bin/gitolite setup -pk YourName.pub
```

Client operations

1.clone server gitolite Manage warehouse:

```
git clone ssh:// git@10.10.10.206 /gitolite-admin.git
```

2.Add user public key to gitoliteUnder contents

```
cp username.pub keydir/username.pub
```

3.Add admin user

```
viconf/gitolite.conf  
@admin = admin1 admin2 admin3  
repo gitolite-admin  
RW+      =@admin
```

repoMirror building

Server side operation

1.usegitAccount login server

2.Download in root directory repotool

```
gitclonessh:// git@www.rockchip.com.cn :2222/repo-release/tools/repo
```

3.NewRK_Android11_mirrorTable of contents

```
mkdir RK_Android11_mirror
```

4.EnterRK_Android11_mirrorTable of contents

```
cd RK_Android11_mirror
```

5.downloadRK Android11 SDKMirror

```
~/repo/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_R/manifests.git-mAndroid11.xml-- mirror
```

6.Create warehouse group permissions

```
. repo/repo/repo list -n > android_r.conf  
sed -i 's/^@android_r = RK_Android11_mirror\/&/g' android_r.conf
```

Client operations

1.Convert the server-side android_r.conf Copied to client.gitolite-admin/conf/.Down

2.Add group permissions

```
vi conf/android_r.conf  
@usergroup = user1 user2 user3 repo  
    @android_r  
R      = @usergroup  
RW+   = @admin
```

```
viconf/gitolite.conf  
include "android_r.conf"
```

5.Create your ownmanifestsstorehouse

```
vi conf/android_r.conf  
@android_r = Android_R/manifests_xxx
```

Client operations

1.Download on clientmanifests_xxxThe warehouse is downloaded on other
client computersmanifests_xxx.gitstorehouse

```
git clone ssh:// git@10.10.10.206 /Android_R/manifests_xxx.git
```

2.Download the original on the clientmanifestsstorehouse

```
git clone ssh:// git@10.10.10.206 /Android_R/manifests.git
```

3.submitmanifest.xmlfile to newmanifest_xxxThe original will be in the
warehousemanifestsThe following files were copied tomanifests_xxxInside

```
cd manifests_xxx  
cp -rf manifests/*.xml manifests_xxx/
```

View copied files

```
git status  
  
Android11.xml  
Android11_Express.xml  
default.xml  
include/partner_gms_express.xml include/  
partner_modules_express.xml include/  
rk3288_repository.xml include/  
rk326_repository.xml include/  
rk3399_repository.xml include/  
rk356x_repository.xml include/  
rk_checkout_from_aosp.xml include/  
rk_modules_repository.xml remote.xml  
  
remove_r.xml
```

local submission

```
git add -A  
git commit -m "init xxx"
```

pushto remote branch

```
git push origin master:master
```

7.Create your own code download link

Download in the root directoryrepotool

```
gitclonesh:// git@www.rockchip.com.cn :2222/repo-release/tools/repo
```

After following the above steps, your code download link is as follows

```
mkdir Android11  
cd Android11  
~/repo/repo init -u ssh:// git@10.10.10.206 /Android_R/manifests_xxx.git -m Android11.xml
```

in:

//10.10.10.206 is your server address

Through the above steps, you can complete your ownrepoAfter the server is set up, you can share your code server link with colleagues to work together.

Code management

After setting up the code server through the above steps, most of the code repositories useRKThe default branch. If there is a warehouse that needs to modify its own code, you can refer to the following steps.

Switch your own code branch

1.Enter the code repository that needs to be modified tokernelTake the directory as an example to illustrate

```
cd kernel
```

2.Switch a local branch

```
git checkout remotes/m/master -b xxx_branch
```

3. push xxx_branchBranch to remote server

```
git push rk29 xxx_branch:xxx_branch
```

in rk29 yesremotecan be directlytabKey auto-completion

4.Enter.repo/manifestsDirectory modificationmanifestThe branch specified in

Enter.repo/manifestsDirectory passedgrep kernelcan be foundkernelWarehouse correspondingmanifests position

```
cd .repo/manifests
```

```

--- a/include/rk_modules_repository.xml b/
+++ include/rk_modules_repository.xml
@@ -10,7 +10,7 @@
    <project path="hardware/rockchip/libgraphicpolicy" name="rk/
hardware/rk29/libgraphicpolicy" remote="rk" revision="refs/tags/
android-11.0-mid-rkr8" />
    <project path="hardware/rockchip/libhwjpeg" name="rk/hardware/rk29/libhwjpeg" revision="refs/
remote="rk" tags/android-11.0-mid-rkr8"/>
    <project path="u-boot" name="rk/u-boot" remote="rk"
revision="refs/tags/android-11.0-mid-rkr8"/>
- <project path="kernel" name="rk/kernel" remote="rk29" revision="refs/
tags/android-11.0-mid-rkr8"/>
+ <project path="kernel" name="rk/kernel" remote="rk29" revision="xxx_branch"/>

    <project path="bootable/recovery/rkupdate" name="platform/
bootable/recovery/rk_update" remote="rk" revision="refs/tags/
android-11.0-mid-rkr8"/>
    <project path="bootable/recovery/rkutility"
name="platform/bootable/recovery/rk_utility" revision="refs/remote="rk"
tags/android-11.0-mid-rkr8"/>

```

5. Submit changes manifest to remote branch

```

git add include/rk_modules_repository.xml
git commit -m "change kernel branch on xxx_branch" git push
origin default:master

```

submit manifests After repository, other colleagues can synchronize to your own branches.kernel code.

Code modification submission

After switching branches according to the above steps, you can submit your modifications on your own branch. Submit directly `pusharrive xxx_branch` above the branch.

Synchronize RK code

1. Synchronize RK The code needs to be done on the server side sync operate

```
cd RK_Android11_mirror
```

```
.repo/repo/repo sync-c
```

2. client merge RK right manifests Modifications

- download RK original manifests storehouse

```
git clone //10.10.10.206/wlq/test/manifests.git
```

Compare using the compare tool `manifests(RK original)` and `manifests_xxx(own)`, will RK The modified differences are merged into their own warehouse (main modification tag, add and delete warehouses, etc.).

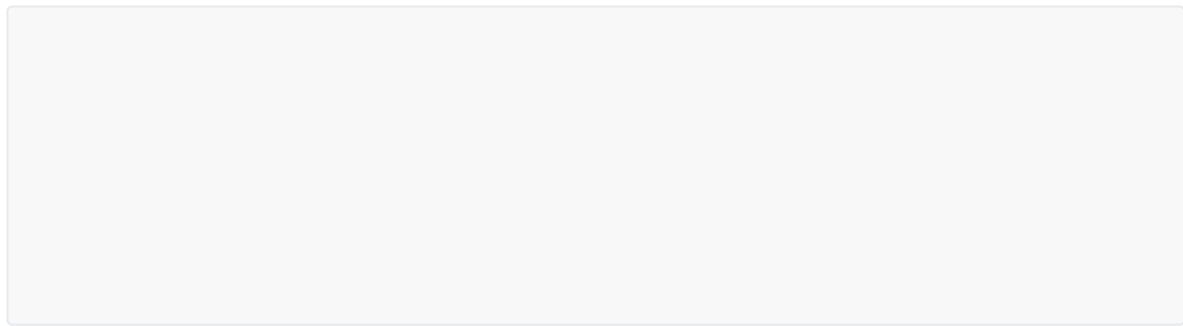
- Will be modified after comparison and confirmation `pusharrive manifests_xxx superior`.

In this step, you can also confirm which warehouses you have modified. In the next step, you will merge the modified warehouses.

3.Directories that have their own branches need to be manually editedRKModificationsmergeto your own branch

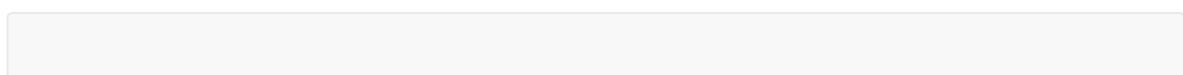
tokernelFor example:

- View the remote branch currently pointed to

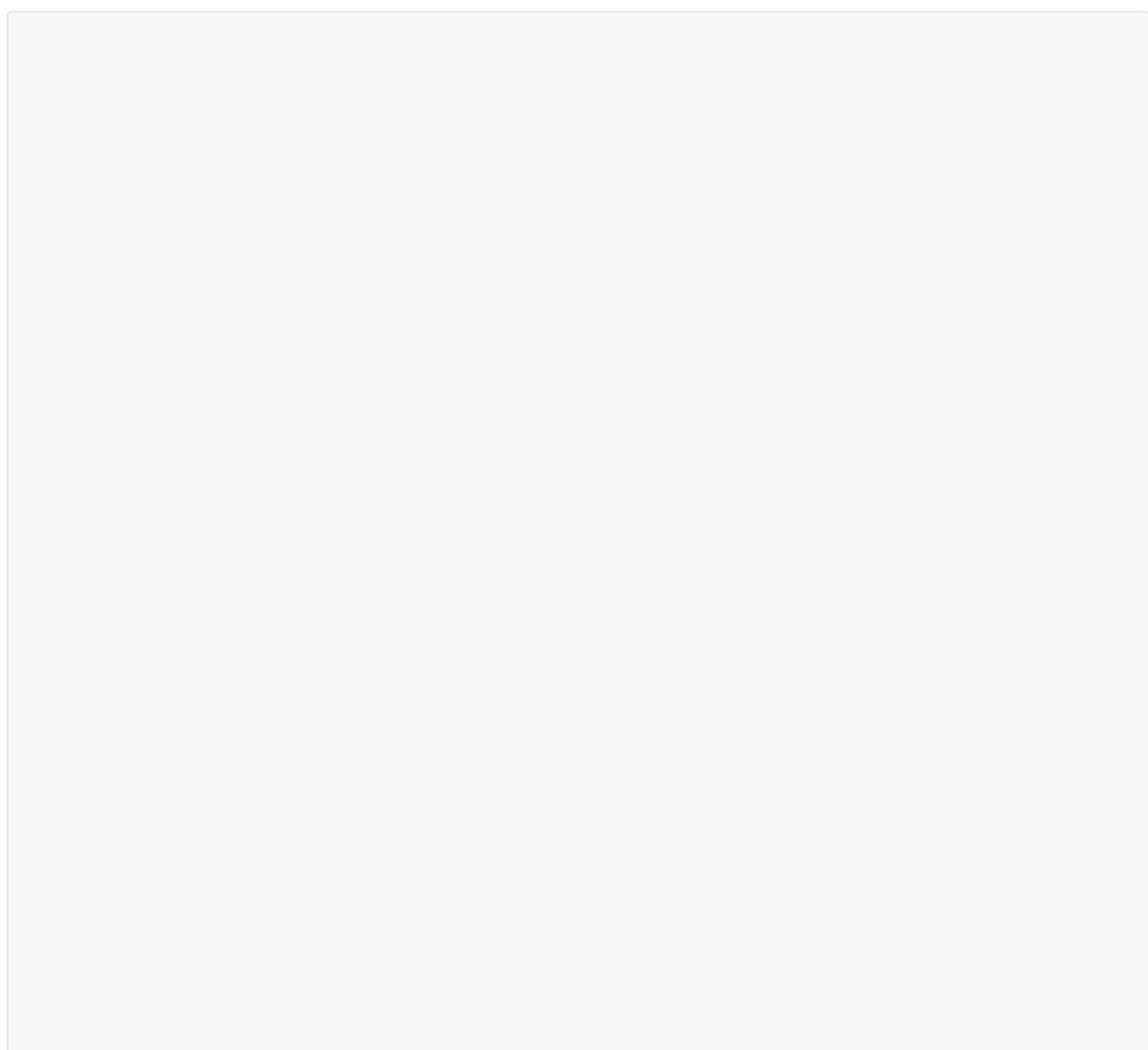


You can see that it is currently pointing to: `remotes/m/master` → `rk29/xxx_branch`

- Create a local branch (cut from your own remote branch)



- Confirm currentRKPublished



-

```
git merge android-11.0-mid-rkr8
```

Check whether there is a conflict. If there is a conflict, resolve the conflict first. If there is no conflict, proceed to the next step.

- pushMerge the completed code to the remote branch

```
git push rk29 local_xxx_branch:xxx_branch
```

- All other split directories can be merged and submitted in this way.

code compilation

One-click compilation command

```
./build.sh-UKAup  
(WHERE: -U = build uboot  
      - C = build kernel with Clang  
      - K = build kernel  
      - A = build android  
      - p = will build packaging in IMAGE  
      - o = build OTA package  
      - u = build update.img  
      - v = build android with 'user' or 'userdebug'  
      - d = build kernel dts name  
      - V = build version  
      - J = build jobs  
----- Everyone can use it as needed, no need to record ituboot/kernelCompile command-----  
)
```

```
===== Please note that before  
using the one-click compilation command, you need to set environment variables and select the  
platform you need to compile. For example: source build/envsetup.sh lunch rk3566_rgo-userdebug  
=====
```

Summary of compilation commands for each platform

Soc	type	model	Android	One-click compilation	kernelcompile	ubootcompile translate
RK3566	flat	prototype	build/envsetup.sh;lunch rk3566_rgo-userdebug	./build.sh -AUCu	make ARCH=arm64 rockchip_defconfig android-11.config;make ARCH=arm64rk3566-rk817-tablet.img-j24	./make.sh rk3566
RK3568	develop plate	EVB1- DDR4-V10	build/envsetup.sh;lunch rk3568_r-userdebug	./build.sh -AUCu	make ARCH=arm64 rockchip_defconfig rk356x_evb.config android-11.config;make ARCH=arm64rk3568-evb1-ddr4-v10.img-j24	./make.sh rk3568
RK3568	develop plate	EVB2- LPDDR4X- V10	build/envsetup.sh;lunch rk3568_r-userdebug	./build.sh -AUCu	make ARCH=arm64 rockchip_defconfig rk356x_evb.config android-11.config;make ARCH=arm64rk3568-evb2-lp4x-v10.img-j24	./make.sh rk3568
RK3568	develop plate	EVB4- LPDDR3- V10	build/envsetup.sh;lunch rk3568_r-userdebug	./build.sh -AUCu	make ARCH=arm64 rockchip_defconfig rk356x_evb.config android-11.config;make ARCH=arm64rk3568-evb4-lp3-v10.img-j24	./make.sh rk3568
RK3568	develop plate	EVB5- DDR4-V10	build/envsetup.sh;lunch rk3568_r-userdebug	./build.sh -AUCu	make ARCH=arm64 rockchip_defconfig rk356x_evb.config android-11.config;make ARCH=arm64rk3568-evb5-ddr4-v10.img-j24	./make.sh rk3568
RK3568	develop plate	EVB6- DDR3-V10	build/envsetup.sh;lunch rk3568_r-userdebug	./build.sh -AUCu	make ARCH=arm64 rockchip_defconfig rk356x_evb.config android-11.config;make ARCH=arm64rk3568-evb6-ddr3-v10.img-j24	./make.sh rk3568
RK3568	develop plate	EVB7- DDR4-V10	build/envsetup.sh;lunch rk3568_r-userdebug	./build.sh -AUCu	make ARCH=arm64 rockchip_defconfig rk356x_evb.config android-11.config;make ARCH=arm64rk3568-evb7-ddr4-v10.img-j24	./make.sh rk3568
RK3566	develop plate	EVB1- DDR4-V10	build/envsetup.sh;lunch rk3566_r-userdebug	./build.sh -AUCu -d rk3566-evb2-lp4x-v10	make ARCH=arm64 rockchip_defconfig rk356x_evb.config android-11.config;make ARCH=arm64rk3566-evb1-ddr4-v10.img-j24	./make.sh rk3566
RK3566	develop plate	EVB2- LP4X-V10	build/envsetup.sh;lunch rk3566_r-userdebug	./build.sh -AUCu -d rk3566-evb2-lp4x-v10	make ARCH=arm64 rockchip_defconfig rk356x_evb.config android-11.config;make ARCH=arm64rk3566-evb2-lp4x-v10.img-j24	./make.sh rk3566

Soc	type	model	Android	One-click compilation	kernelcompile	ubootcompile translate
RK3566	develop plate	EVB3- DDR3-V10	build/envsetup.sh;lunch rk3566_r-userdebug	./build.sh - AUCu -d rk3566- evb2-lp4x- v10	make ARCH=arm64 rockchip_defconfig rk356x_evb.config android-11.config;make ARCH=arm64rk3566- evb3-ddr3-v10.img-j24	./make.sh rk3566
RK3566	develop plate	EVB5- LPDDR4X- V10	build/envsetup.sh;lunch rk3566_r-userdebug	./build.sh - AUCu -d rk3566- evb2-lp4x- v10	make ARCH=arm64 rockchip_defconfig rk356x_evb.config android-11.config;make ARCH=arm64rk3566- evb5-lp4x-v10.img-j24	./make.sh rk3566
RK3566	BOX- DEMO plate	rk3566- box- demo-v10	build/envsetup.sh;lunch rk356x_box-userdebug	./build.sh - AUCu -d rk3566- box-demo- v10	make ARCH=arm64 rockchip_defconfig rk356x_evb.config android-11.config;make ARCH=arm64rk3566- box-demo-v10.img-j24	./make.sh rk3566
RK3326	flat	863prototype	build/envsetup.sh;lunch rk3326_rgo-userdebug	./build.sh - AUCu	make ARCH=arm64 rockchip_defconfig android-11- go.config;make ARCH=arm64rk3326- 863-lp3-v10-rksip1.img- j24	./make.sh rk3326
PX30	develop plate	px30-evb- ddr3-v10- avb	build/envsetup.sh;lunch PX30_Android11- userdebug	./build.sh - AUCu	make ARCH=arm64 rockchip_defconfig android-11.config;make ARCH=arm64 px30-evb- ddr3-v10-avb.img-j24	./make.sh px30
RK3399	dig <small>Turn on the machine Hair board</small>	rk3399- sapphire- excavator- edp-avb	build/envsetup.sh;lunch rk3399_Android11- userdebug	./build.sh - AUCu -d rk3399- sapphire- excavator- edp-avb	make ARCH=arm64 rockchip_defconfig android-11.config;make ARCH=arm64rk3399- sapphire-excavator- edpavb.img-j24	./make.sh rk3399
RK3399	INDOK <small>business opened Hair board</small>	rk3399- evb-ind- lpddr4- android- avb	build/envsetup.sh;lunch rk3399_Android11- userdebug	./build.sh - AUCu	make ARCH=arm64 rockchip_defconfig android-11.config;make ARCH=arm64rk3399- evb-ind-lpddr4- androidavb.img-j24	./make.sh rk3399
RK3288	develop plate	rk3288- evb- android- rk808- edp-avb	build/envsetup.sh;lunch rk3288_Android11- userdebug	./build.sh - AUCu	make ARCH=arm rockchip_defconfig android-11.config;make ARCH=arm rk3288-evb- android-rk808-edp- avb.img-j24	./make.sh rk3288
RK3328	boxopen <small>Hair board</small>	rk3328- box- liantong- avb	build/envsetup.sh;lunch rk3328_box-userdebug	./build.sh - AUCu	make ARCH=arm rockchip_defconfig android-11.config;make ARCH=arm rk3328-box- liantong-avb.img-j24	./make.sh rk3328

Other compilation instructions

Android11.0 Cannot be programmed directly kernel.img and resource.img

Android11.0ofkernel.imgandresource.imgincludedinboot.imgin,needtousebuild.sh-Kcommandtocompilekernel.Burnaftercompilationrockdevbelowboot.img.Youcanalsocompileitseparatelyusingthefollowingmethodkernel.

Compile separately kernel generate boot.img

Principle of compilation: inkernelDirectory will be compiled and generated with **kernel.img** and **source.img** Replace with old one **boot.img** middle.

RK3566 Take the prototype as an example, replace the corresponding boot.img and dts:

BOOT_IMG=..\\rockdev\\Image-rk3566_r\\boot.img What is specified here is the old one boot.img path, command such as

Down:

```
cd kernel  
make ARCH=arm64 rockchip_defconfig android-11.config  
make ARCH=arm64 BOOT_IMG=..\\rockdev\\Image-rk3566_r\\boot.img rk3566-rk817-tablet.img  
-j24
```

Can be programmed directly after compilation kernel under the directory boot.img (**Notice: 32bit The platform is zboot.img, like 3126c/rk3288w**) to the machine boot location, when programming **Please load the partition table first (parameter.txt)**, to avoid incorrect programming location.

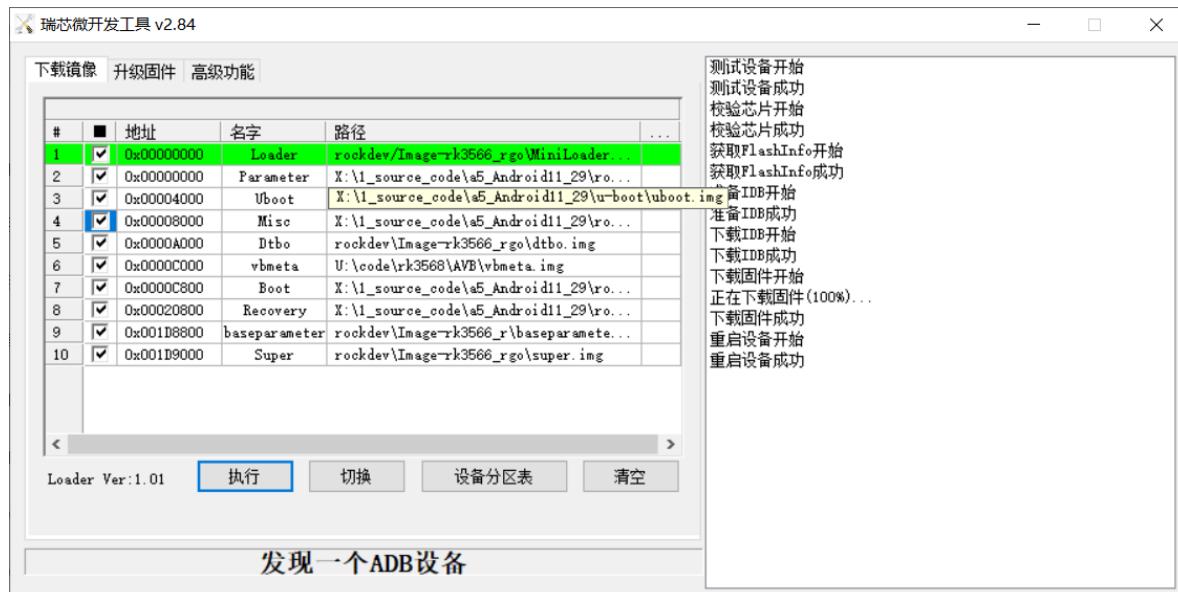
Firmware programming

Firmware burning tool

Android11ofUSBdriveDriverAssitantNeed to update to V5.1.1 version, you can refer to the Tools chapter below to update.

Windows Burning tool: (the tool is updated all the time, please update it in time)

RKTools/windows/AndroidTool/AndroidTool_Release_v2.84



RKTools/linux/Linux_Upgrade_Tool/Linux_Upgrade_Tool_v1.56

Detailed instructions are provided in the tool description section below.

Firmware instructions

After complete compilation, the following file will be generated: (starting with RK3566 For example, here lunch yes rk3566_rgo-userdebug)

```
rockdev\\Image-rk3566_rgo/  
|---boot-debug.img  
|---boot.img
```

```
|── config.cfg  
|── dtbo.img  
|── MiniLoaderAll.bin  
|── misc.img  
|── parameter.txt  
|── pcba_small_misc.img  
|── pcba_whole_misc.img  
|── recovery.img  
|── resource.img  
|── super.img  
|── uboot.img  
|── update.img  
└── vbmeta.img
```

The tool can be used to program the following files: (RK3566/RK3568No need to flashtrust.img)

```
rockdev/Image-rk3566_rgo/  
├── boot.img  
├── dtbo.img  
├── MiniLoaderAll.bin  
├── misc.img  
├── parameter.txt  
├── recovery.img  
├── super.img  
├── uboot.img  
└── vbmeta.img
```

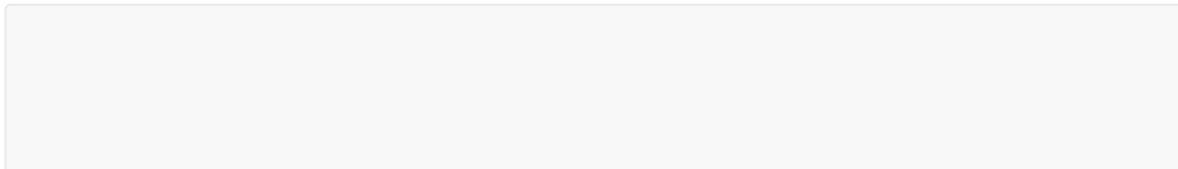
You can also directly program [update.img](#)

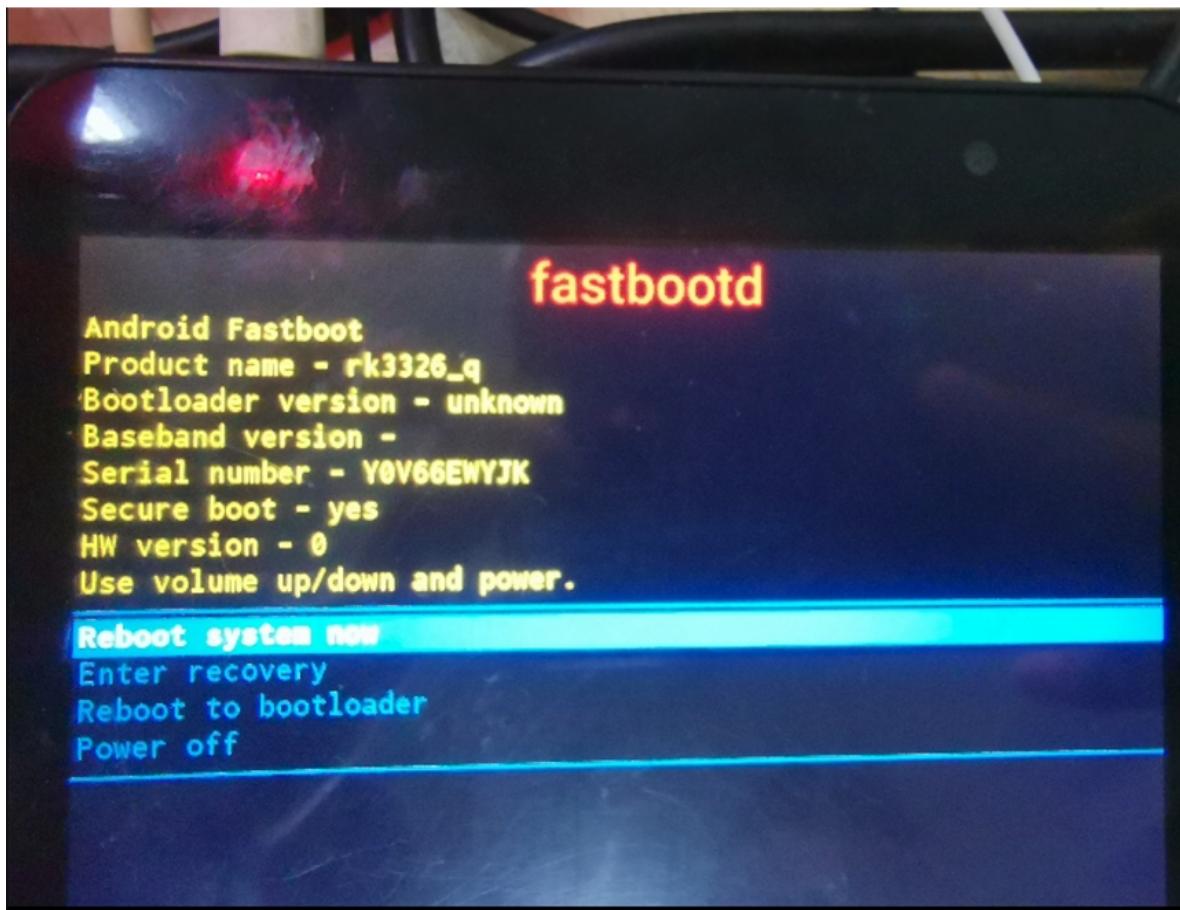
[Firmware instructions](#)

firmware	illustrate
boot.img	Includeramdis,kernel,dtb
boot-debug.img	andboot.imgThe difference is userThe firmware can be programmed with thisboot.imgconductrootPermission operation
dtbo.img	Device Tree OverlaysRefer to the followingdtboChapter Description
config.cfg	The configuration file of the programming tool can be directly imported into the programming tool to display the options that need to be programmed.
MiniLoaderAll.bin	Contains one levelloader
misc.img	Includerecovery-wipeThe boot identification information will be carried out after programming.recovery
parameter.txt	Contains partition information
pcba_small_misc.img	IncludepcbaBoot identification information, after programming, you will enter the simple versionpcbamodel
pcba_whole_misc.img	IncludepcbaThe boot identification information will enter the full version after programming.pcbamodel
recovery.img	Includerecovery-ramdis,kernel,dtb
super.img	Includeodm,product,vendor,system,system_extPartition content
trust.img	IncludeBL31,BL32 RK3566/RK3568This firmware is not generated and does not need to be programmed.
uboot.img	Includeubootfirmware
vbmota.img	IncludeavbVerification information forAVBcheck
update.img	Including the above that need to be programmedimgfile, which can be used by tools to directly program the entire firmware package.

fastbootFlash dynamic partition

RThe new device supports dynamic partitioning, which has been removed.system/vendor/odm/product/system_extPartition, please burn super.img, separately programmedsystem/vendor/odmWait (can be inoutFind the corresponding belowimgfile) can be used
fastbootd ,Require readband fastbootThe versions are all





Note: Non-dynamic partition usefastboot, please enterbootloader:

```
adb reboot bootloader
```

FlashGSIMethods:

- After confirming that the machine is unlocked, enterfastbootd, just need to programGSImiddleSystem.img and firmwareMisc.img, you will enter after programmingrecoveryPerform a factory reset. The entire programming process is attached below:

1.Reboot tobootloader, not unlocked -> machine unlocked:

```
adb reboot bootloader
```

fastboot oem at-unlock-vboot ##For burningavbFor customers with public keys, please refer to the corresponding documents to unlock.

2.Restore factory settings and reboot tofastbootd:

```
fastboot flash misc misc.img fastboot  
reboot fastboot ##
```

You will now enterfastbootd

3.Start programmingGSI

```
fastboot delete-logical-partition product ## (Optional) For devices with limited partition space, you can execute this command to  
delete them first.productBurn after partitioningGSI fastboot flash system system.img fastboot reboot ##After programming is  
successful, restart
```

- Note: You can also useDSU (Dynamic System Updates)FlashGSI,at presentRockchipThe platform already supports it by default DSU. Since this function consumes a lot of memory, it is not recommended1G DDRand following equipment use, related toDSUdescription and use

Use, please refer to Android Official website:

<https://source.android.com/devices/tech/ota/dynamic-system-updates> Note1: VTS When testing,

- you need to burn and compile the compiledboot-debug.img arrive boot partition; Note2: CTS-ON-GSI There is no need to burn it during testing boot-debug.img;
- Note3: Please use when testing Google Officially released with -signed final GSI mirror;

useDTBOFunction

Android 10.0 and above support Device Tree Overlay Function, the development process is reflected in the need to program dtbo.img, used for compatibility between multiple products, etc.

Modification method:

1. Find (or specify) the template file:

```
get_build_var PRODUCT_DTBO_TEMPLATE
```

For example:

```
PRODUCT_DTBO_TEMPLATE := $(LOCAL_PATH)/dtoverlay.in(device/rockchip/rk356x/rk3566_r/dt-overlay.in)
```

2. Add or modify the required nodes:

For example:

```
/dts-v1/;
/plugin/;

&chosen {
    bootargs_ext = "androidboot.boot_devices=${_boot_device}";
};

&firmware_android {
    vbmota {
        status = "disabled";
    };
    fstab {
        status = "disabled";
    };
};

&reboot_mode {
    mode-bootloader = <0x5242C309>;
    mode-charge= <0x5242C30B>;
    mode-fastboot = <0x5242C303>;
    mode-loader = <0x5242C301>;
    mode-normal = <0x5242C300>;
    mode-recovery = <0x5242C303>;
};
```

Note: Used tbo Be sure to dt exist in alias, otherwise it will not succeed overlay

Revisefstabdocument

1. Find (or specify) the template file:

```
get_build_var PRODUCT_FSTAB_TEMPLATE
```

For example:

```
PRODUCT_FSTAB_TEMPLATE := device/rockchip/rk356x/rk3566_eink/fstab_eink.in
```

2.Modification: Add partition mounting and modificationswap_zramParameters, modificationdataPartition format, etc.

Reviseparameter.txt

Android 11Added buildparameter.txtTools that support compiling according to configuration parametersparameter.txt. If there is no configuration template file, it will look for the modifiedparameter.txtdocument.

1.Find (or specify) the template file:

```
get_build_var PRODUCT_PARAMETER_TEMPLATE
```

For example:

```
PRODUCT_PARAMETER_TEMPLATE := device/rockchip/common/scripts/
parameter_tools/parameter.in
```

2.Modify configuration partition size (for example):

```
BOARD_SUPER_PARTITION_SIZE := 2688548864
BOARD_DTBOIMG_PARTITION_SIZE := xxxx
BOARD_BOOTIMAGE_PARTITION_SIZE := xxxxx
BOARD_CACHEIMAGE_PARTITION_SIZE := xxxx
```

3.Do not useparameterGenerating tools:

add oneparameter.txtfile to yourdeviceJust go to the directory:

For example:device/rockchip/rk3326/rk3326_q/parameter.txt

4.Generate only using toolsp parameter.txt(For example):

```
parameter_tools --input device/rockchip/common/scripts/parameter_tools/parameter.in --firmware-
version 11.0 --machine-model rk3326 --manufacturer rockchip --machine rk3326_r -- partition-list
```

```
uboot_a:4096K,trust_a:4M,misc:4M,dtbo_a:4M,vbmeta_a:4M,boot_a:33554432,backup:30
0M,security:4M,cache:300M,metadata:4096,frp:512K,super:2G -- output parameter_new.txt
```

Note: If you need a larger versionOTATO upgrade, please use the previous version directly.parameter.txt

5.Add a new partition

Create newbaseparameterTake partition as an example to illustrate:

- in productBoardConfig.mkDefinition:BOARD_WITH_SPECIAL_PARTITIONS like:
BOARD_WITH_SPECIAL_PARTITIONS := baseparameter:1M,logo:16M

```
device/rockchip/rk356x/rk3566_r/BoardConfig.mk
+++ b/BoardConfig.mk
@@ -494,4 +494,11 @@ ifeq ($(strip $(BOARD_TWRP_ENABLE)), true)
+ BOARD_WITH_SPECIAL_PARTITIONS := baseparameter:1M
```

- existRebuildParameter.mkAdd inBOARD_WITH_SPECIAL_PARTITIONS

```
device/rockchip/common/build/rockchip/RebuildParameter.mk
+ ifneq ($(strip $(BOARD_WITH_SPECIAL_PARTITIONS)), )
+ partition_list := $(partition_list)$(BOARD_WITH_SPECIAL_PARTITIONS)
+ endif
```

AndroidCommon configuration

New productlunch

byRK356xNew platformrk3568_rTaking the product as an example, the steps are as follows:

1)Revisedevice/rockchip/rk356x/AndroidProducts.mkIncreaserk3568_roflunch

```
--- a/AndroidProducts.mk
+++ b/AndroidProducts.mk
@@ -17,10 +17,14 @@
PRODUCT_MAKEFILES := \
    $(LOCAL_DIR)/rk3566_rgo/rk3566_rgo.mk $      \
    $(LOCAL_DIR)/rk3566_r/rk3566_r.mk $      \
+    $(LOCAL_DIR)/rk3568_r/rk3568_r.mk      \
COMMON_LUNCH_CHOICES := \
    rk3566_rgo-userdebug      \
    rk3566_rgo-user \
    rk3566_r-userdebug      \
    rk3566_r-user \
+    rk3568_r-userdebug      \
+    rk3568_r-user \
```

2)existdevice/rockchip/rk356xCreate a new directoryrk3568_rTable of contents

refer todevice/rockchip/rk356xdownload existingrk3566_rTo create a new product catalog, you can directly copy it firstrk3566_rforsk3568_r,

Thenrk3568_rAll in the directory rk3566_r characters changed tork3568_r

Kernel dtsillustrate

New productdts

Product newdtsYou can select the corresponding one according to the configuration in the table belowdtsReference.

Soc	PMIC	DDR	Development board type	model	DTS
RK3566	RK817	LPDDR4	flat	prototype	rk3566-rk817-tablet
RK3566	RK809	LPDDR4	Development board	RK3566 EVB2	rk3566-evb2-lp4x-v10
RK3566	separate	DDR4	Development board	RK3566 BOX DEMO	rk3566-box-demo-v10
RK3568	RK809	DDR4	Development board	RK3568 EVB1	rk3568-evb1-ddr4-v10
RK3326	RK817	DDR3	Development board	evb	rk3326-evb-lp3-v10-avb
PX30	RK809	DDR3	Development board	evb	px30-evb-ddr3-v10-avb
RK3326	RK817	DDR3	flat	prototype	rk3326-863-lp3-v10-rkisp1
RK3326	RK809	DDR3	artificial intelligence Able to speak	EVB	rk3326-evb-ai-va-v12
RK3399	RK808	LPDDR3	Development board	excavator	rk3399-sapphire-excavator-edp-avb
RK3399	RK809	LPDDR4	Development board	INDDevelopment board	rk3399-evb-ind-lpddr4-android-avb
RK3399	RK808	LPDDR4	flat	BQ25703pair battery saver	rk3399-tve1030g-avb
RK3399	RK818	LPDDR3	flat	edpScreen	rk3399-mid-818-android
RK3288W	RK808	LPDDR3	Development board	edpScreen	rk3288-evb-android-rk808-edp-avb
RK3328	separate	DDR3	Development board	RK3328develop plate	rk3328-box-liantong-avb

Documentation description

Peripheral support list

DDR/EMMC/NAND FLASH/WIFI/3G/CAMERAThe support list is updated in real time atredmineon, the link is as follows:

<https://redmine.rockchip.com.cn/projects/fae/documents>

Android document

RKDocs\android

Android_SELinux(Sepolicy)Development Guide

RKDocs/android/Rockchip_Developer_Guide_Android_SELinux(Sepolicy)_CN.pdf

Android 11System optimization guidance

Including boot speed,AppStartup speed, performance, memory optimization and commonly used analysis tools

RKDocs\android\Rockchip_Developer_Guide_Android11_Optimization_CN.pdf

Wi-Fi document

RKDocs/android/wifi/
└── Rockchip_Introduction_Android10.0_WIFI_Configuration_CN&EN.pdf
└── Rockchip_Introduction_REALTEK_WIFI_Driver_Porting_CN&EN.pdf

3G/4GModule documentation

RKDocs/common/mobile-net/
└── Rockchip_Introduction_3G_Data_Card_USB_File_Conversion_CN.pdf
└── Rockchip_Introduction_3G_Dongle_Configuration_CN.pdf
└── Rockchip_Introduction_4G_Module_Configuration_CN&EN.pdf

Kerneldocument

RKDocs\common

DDRRelated documents

RKDocs/common/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

AudioModule documentation

RKDocs/common/Audio/
└── Rockchip_Developer_Guide_Audio_Call_3A_Algorithm_Integration_and_Parameter_Debug_ging_CN.pdf
└── Rockchip_Developer_Guide_Linux4.4_Audio_CN.pdf
└── Rockchip_Developer_Guide_RK817_RK809_Codec_CN.pdf

CRUModule documentation

RKDocs/common/CRU/
└── Rockchip_Developer_Guide_Linux3.10_Clock_CN.pdf
└── Rockchip_RK3399_Developer_Guide_Linux4.4_Clock_CN.pdf

GMACModule documentation

```
RKDocs/common/GMAC/  
└── Rockchip_Developer_Guide_Ethernet_CN.pdf
```

PCieModule documentation

```
RKDocs/common/PCie/  
└── Rockchip-Developer-Guide-linux4.4-PCIe.pdf
```

I2CModule documentation

```
RKDocs/common/I2C/  
└── Rockchip_Developer_Guide_I2C_CN.pdf
```

PIN-CtrlGPIOModule documentation

```
RKDocs/common/PIN-Ctrl/  
└── Rockchip-Developer-Guide-Linux-Pin-Ctrl-CN.pdf
```

SPIModule documentation

```
RKDocs/common/SPI/  
└── Rockchip-Developer-Guide-linux4.4-SPI.pdf
```

SensorModule documentation

```
RKDocs/common/Sensors/  
└── Rockchip_Developer_Guide_Sensors_CN.pdf
```

IO-DomainModule documentation

```
RKDocs/common/IO-Domain/  
└── Rockchip_Developer_Guide_Linux_IO_DOMAIN_CN.pdf
```

LedsModule documentation

```
RKDocs/common/Leds/  
└── Rockchip_Introduction_Leds_GPIO_Configuration_for_Linux4.4_CN.pdf
```

ThermalTemperature control module documentation

```
RKDocs/common/Thermal/  
├── Rockchip-Developer-Guide-Linux4.4-Thermal-CN.pdf  
└── Rockchip-Developer-Guide-Linux4.4-Thermal-EN.pdf
```

RKDocs/common/PMIC/
└── Archive.zip
└── Rockchip_Developer_Guide_Power_Discrete_DCDC_EN.pdf Rockchip-
 └── Developer-Guide-Power-Discrete-DCDC-Linux4.4.pdf Rockchip-Developer-
 └── Guide-RK805.pdf
 └── Rockchip_Developer_Guide_RK817_RK809_Fuel_Gauge_CN.pdf
 └── Rockchip_RK805_Developer_Guide_CN.pdf
 └── Rockchip_RK818_RK816_Introduction_Fuel_Gauge_Log_CN.pdf

MCU Module documentation

RKDocs/common/MCU/
└── Rockchip_Developer_Guide MCU EN.pdf

Power Consumption and Hibernation Module Documentation

RKDocs/common/power/
└── Rockchip_Developer_Guide_Power_Analysis_EN.pdf
└── Rockchip_Developer_Guide_Sleep_and_Resume_CN.pdf

UART Module documentation

RKDocs/common/UART/
└── Rockchip-Developer-Guide-linux4.4-UART.pdf Rockchip-
 └── Developer-Guide-RT-Thread-UART.pdf

DVFS CPU/GPU/DDR Frequency conversion related documents

RKDocs/common/DVFS/
└── Rockchip_Developer_Guide_CPUFreq_CN.pdf
└── Rockchip_Developer_Guide_CPUFreq_EN.pdf
└── Rockchip_Developer_Guide_Devfreq_CN.pdf
└── Rockchip_Developer_Guide_Linux4.4_CPUFreq_CN.pdf
└── Rockchip_Developer_Guide_Linux4.4_Devfreq_CN.pdf

EMMC/SDMMC/SDIO Module documentation

RKDocs/common/MMC
└── Rockchip-Developer-Guide-linux4.4-SDMMC-SDIO-eMMC.pdf

PWM Module documentation

RKDocs/common/PWM/
└── Rockchip-Developer-Guide-Linux-PWM-CN.pdf
└── Rockchip_Developer_Guide_PWM_IR_CN.pdf

USB Module documentation

RKDocs/common/usb/
 putty20190213_162833_1.log Rockchip-Developer-Guide-Linux4.4-RK3399-USB-DTS-CN.pdf Rockchip-Developer-Guide-Linux4.4-USB-CN.pdf Rockchip-Developer-Guide-Linux4.4-USB-FFS -Test-Demo-CN.pdf Rockchip-Developer-Guide-Linux4.4-USB-Gadget-UAC-CN.pdf Rockchip-Developer-Guide-USB-Initialization-Log-Analysis-CN.pdf Rockchip-Developer-Guide- USB-Performance-Analysis-CN.pdf Rockchip-Developer-Guide-USB-PHY-CN.pdf

 Rockchip-Developer-Guide-USB-SQ-Test-CN.pdf

HDMI-INFOunctional documentation

RKDocs/common/hdmi-in/
 Rockchip_Developer_Guide_HDMI_IN_CN.pdf

Security module documentation

RKDocs/common/security/ |—— Efuse
process explain .pdf |——
 RK3399_Efuse_Operation_Instructions_V1.00_20190214_EN.pdf
 Rockchip_Developer_Guide_Secure_Boot_V1.1_20190603_CN.pdf
 Rockchip_Developer_Guide_TEE_Secure_SDK_CN.pdf
 Rockchip_RK3399_Introduction_Efuse_Operation_EN.pdf Rockchip-Secure-
 Boot 2.0.pdf
 Rockchip-Secure-Boot-Application-Note-V1.9.pdf
 Rockchip Vendor Storage Application Note.pdf

ubootIntroduction document

RKDocs\common\u-boot\Rockchip-Developer-Guide-UBoot-nextdev-CN.pdf

TrustIntroduction document

RKDocs/common/TRUST/
 Rockchip_Developer_Guide_Trust_CN.pdf
 Rockchip_Developer_Guide_Trust_EN.pdf

Camerasdocument

RKDocs\common\camera\HAL3\

Camera IQ Tooldocument

external/camera_engine_rkaiq/rkisp2x_tuner/doc/ |——
 Rockchip_Color_Optimization_Guide_ISP2x_CN_v2.0.0.pdf
 Rockchip_IQ_Tools_Guide_ISP2x_CN_v2.0.0.pdf
 Rockchip_Tuning_Guide_ISP21_CN_v2.0.0.pdf

Tool documentation

RKDocs\common\RKTools manuals

PCBA Development and usage documentation

RKDocs\android\Rockchip_Developer_Guide_PCBA_Test_Tool_CN&EN.pdf

Display Driver Debugging Guide

RKDocs\common\display\Rockchip_Developer_Guide_DRM_Panel_Porting_CN.pdf

HDMI Debugging Guide

RKDocs\common\display\Rockchip_Developer_Guide_HDMI_Based_on_DRM_Framework_CN.pdf

Image display DRM Hardware Composer(HWC) Problem analysis and troubleshooting

RKDocs\common\display\Rockchip FAQ DRM Hardware Composer V1.00-20181213.pdf

DRMShow development guide

RKDocs\common\display\Rockchip DRM Display Driver Development Guide V1.0.pdf

RGA Analysis and troubleshooting of related issues

RKDocs\common\display\Rockchip_RGA_FAQ.pdf

Analysis of Frequently Asked Questions about Graphical Display Framework

includeframeworks, GPU.Gralloc, GUI, HWComposer, HWUI, RGA

RKDocs\common\display\Rockchip_Trouble_Shooting_Graphics

Tool usage

StressTest

Use on deviceStresstestTool to perform stress testing on various functions of the equipment under test to ensure the stability of the entire system operation.SDK By opening the Calculator app, enter **"83991906=** Password, can be activatedStressTestApplication, perform stress testing of various functions.

StresstestThe content tested by the testing tool mainly includes:

Module related

- CameraStress testing: includedCameraOpen close, CameraTake photos and CameraSwitch.
- BluetoothStress testing: includedBluetoothOpen close.
- Wi-Fi Stress testing: includedWi-FiOpen close, (pingTest welliperfTesting to be added).

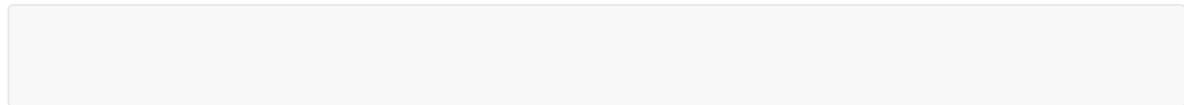
Not module related

- Airplane mode switch test
- Sleep wake-up copy machine test
- Video copy test
- Restart the copy machine test
- Restore factory settings and copy machine test
- ArmFrequency conversion test
- GPUFrequency conversion test
- DDRFrequency conversion test

PCBAtest tools

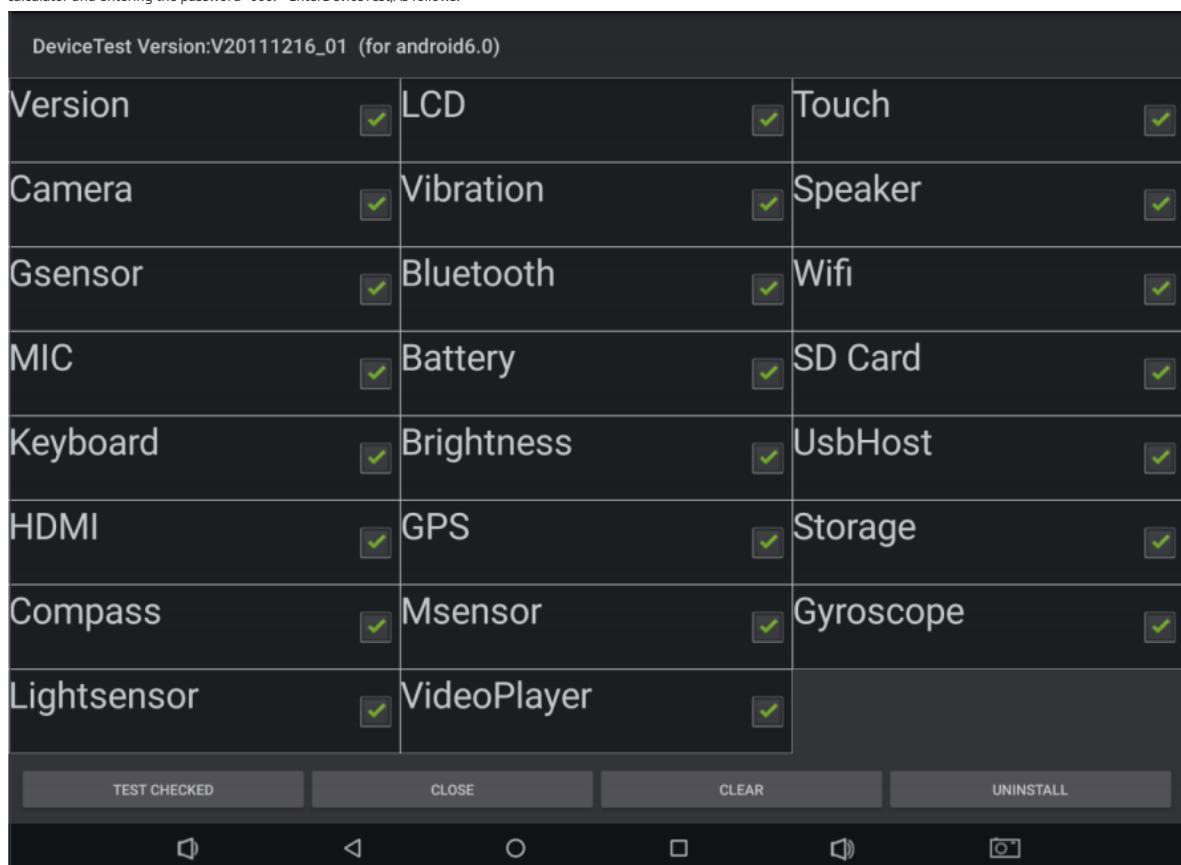
PCBAtesting tools are used to help quickly identify the quality of product functions during mass production and improve production efficiency.

Currently includes screens (LCD),wireless(Wi-Fi),Bluetooth(bluetooth),DDR/EMMCstorage,SDCard(sdcard),USB HOST, button (KEY), speaker headphones (Codec)Test items. These test items include automatic test items and manual test items, wireless network,DDR/EMMC, Ethernet is an automatic test item, press the button, SDCard,USB HOST,Codec, is a manual test project. specificPCBAFor functional configuration and usage instructions, please refer to:



DeviceTest

DeviceTestIt is used for factory complete machine testing, mainly testing whether the peripheral components are normal after the complete machine is installed.SDKBy opening the calculator and entering the password "000.=EnterDeviceTest,As follows:



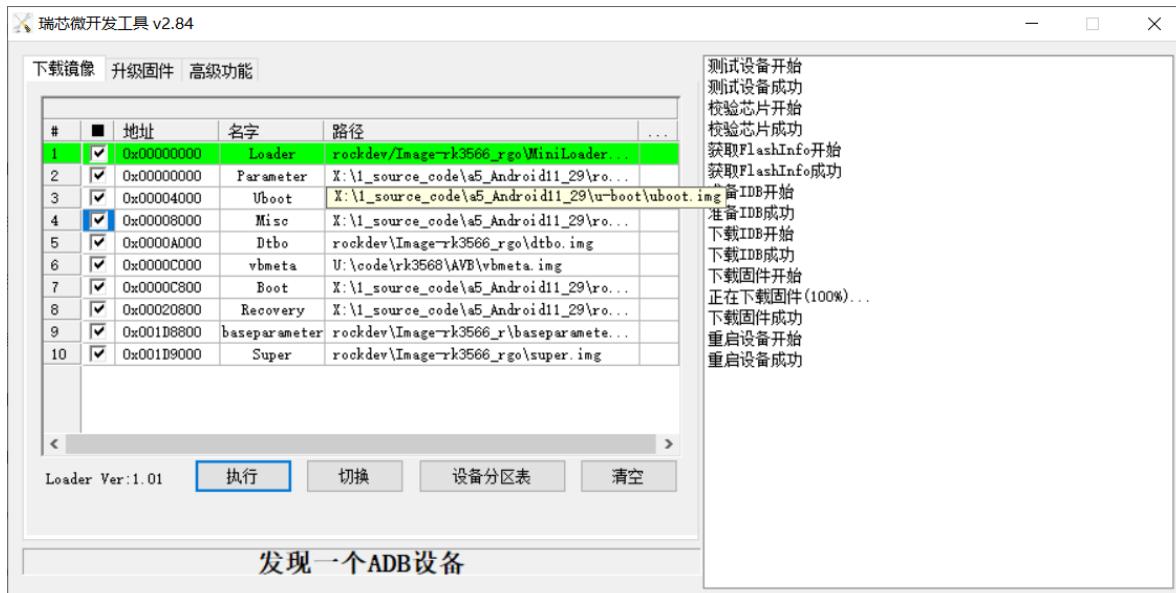
In the production line, you can test the corresponding peripherals based on this interface. Click "TEST CHECKED"Test the tested items one by one. If the test is successful, clickpass, failed clickfailed,

RKTools\windows\DriverAssitant_v5.1.1.zip

Develop programming tools

WindowsVersion

RKTools/windows/AndroidTool/AndroidTool_Release_v2.84.zip, the tool version will be updated all the time, please update it in time.



LinuxVersion

RKTools/linux/Linux_Upgrade_Tool/Linux_Upgrade_Tool_v1.56.zip

```
Linux_Upgrade_Tool_v1.56$ sudo ./upgrade_tool -h Program
Data in /home/wlq/.config/upgrade_tool

----- Tool Usage -----
Help:          H
Quit:          Q
Version:       V
Clear Screen: CS

----- Upgrade Command -----
ChooseDevice:   CD
ListDevice:     LD
SwitchDevice:   SD
UpgradeFirmware: UF <Firmware> [-noreset]
UpgradeLoader:  UL <Loader> [-noreset]
DownloadImage:  DI <-p|-b|-k|-s|-r|-m|-u|-t|-re> image>
DownloadBoot:   DB <Loader>
EraseFlash:    EF <Loader|firmware> [DirectLBA]
PartitionList:  PL
WriteSN:        SN <serial number>
ReadSN:         RSN

----- Professional Command -----
TestDevice:    TD
ResetDevice:   RD [subcode]
ResetPipe:     RP [pipe]
ReadCapability: RCB
```

ReadFlashID:	RID
ReadFlashInfo:	RFI
ReadChipInfo:	RCI
ReadSector:	RS <BeginSec> <SectorLen> [-decode] [File] <BeginSec>
WriteSector:	WS <File>
ReadLBA:	RL <BeginSec> <SectorLen> [File]
WriteLBA:	wL <BeginSec> <File>
EraseLBA:	EL <BeginSec> <EraseCount>
EraseBlock:	EB <CS> <BeginBlock> <BlockLen> [--Force]

SDUpgrade startup creation tools

used to makeSDcard upgrade, SDcard activation, SDCardPCBAtest

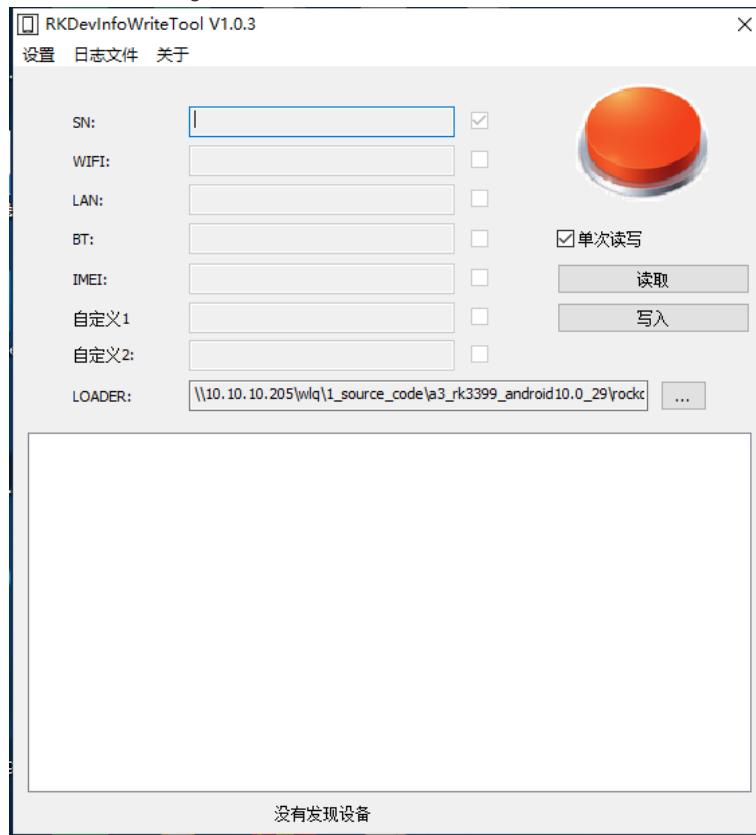
RKTools\windows\SDDiskTool_v1.59.zip

Number writing tool

RKTools\windows\RKDeviceInfoWriteTool_Setup_V1.0.3.rar Unzip

RKDeviceInfoWriteTool_Setup_V1.0.3.rar After installation, open the software

with administrator rights



For tool instructions, please refer to:

RKDocs\common\RKTools manuals\RKDeviceInfoWriteTool_User_Guide_V1.0.3.pdf

DDRWelding test tools

for testing DDR Hardware connections, troubleshooting hardware problems such as virtual soldering

RKTools\windows\Rockchip_Platform_DDR_Test_Tool_V1.38_Release_Annoucement_CN.7z
RKTools\windows\Rockchip_Platform_DDR_Test_Tool_V1.38_Release_Annoucement_EN.7z

efuseProgramming tool

used for efuse programming, suitable for RK3288W/RK3368/RK3399 platform

RKTools\windows\efuse_v1.37.rar

efuse/otpSignature tool

for firmware efuse/otp sign

RKTools\windows\SecureBootTool_v1.94.zip

Factory produced firmware programming tools

Used for factory batch programming of firmware

RKTools\windows\FactoryTool_1.66.zip

Firmware modification tool

for modification update.imgfirmware

RKTools\windows\FWFactoryTool_V5.52.rar

userdataPartition data preset tool

used to make userdataTool for partitioning provisioning data packages

RKTools\windows\OemTool_v1.3.rar

Camera IQ Tool

for debugging ISPImage effects

external/camera_engine_rkaiq/rkisp2x_tuner

system debugging

ADBtool

Overview

ADB(Android Debug Bridge) yes Android SDK tool in , you can use this tool to operate and manage Android simulator or real Androidequipment. The main functions are:

- running equipment shell(Command Line)
- Manage port mappings for an emulator or device

- Upload/download files between computer and device
 - will localapkInstall the software to the emulator orAndroidequipment
- ADBIt is a "client-server" program, where the client mainly refers toPC, the server side isAndroidThe physical machine or virtual machine of the device. according toPCconnectAndroidThe equipment is different,ADBCan be divided into two
- categories: NetworkADB: The host is connected to theSTBequipment USBADB: Host passesUSBline connected toSTB
 - equipment

USB adbInstructions for use

USB adbUse subject to the following restrictions:

- Only supportsUSB OTGmouth
- Simultaneous use of multiple clients (such ascmdwindow,eclipseetc.) Only supports the host
- to connect to one device, and does not support connecting to multiple devices.

The connection steps are as follows:

1,AndroidThe device is already runningAndroidSystem, Settings->Developer Options->Connected to Computer to open,USBThe debugging switch is turned on.

2,PCThe host only passesUSBwire to machineUSBport, and then the computer uses the following command to communicate withAndroidThe device is connected.

adb shell

3, test whether the connection is successful, run"adb devices"command, if the serial number of the machine is displayed, the connection is successful.

networkadbRequirements

adbEarlier versions could only passUSBTo debug the device, start fromadb v1.0.25To start, an increase in thepasstcp/ipdebugAndroidDevice functionality.

If you need to use theInternetadbTo debug the equipment, the following conditions must be met: 1,

the device must first have a network port, or passWiFiConnect Network.

2, equipment and R&D machines (PCmachine) has been connected to the LAN, and the device has a LANIPaddress. 3, to ensure that R&D machines and equipment can interact with each otherpingMakes sense. 4, the R&D machine has been installedadb.

5,make sureAndroidin deviceadbprocess(adbbackground process) is already running.adbdThe process will listen on the port5555to carry out adb Connection debugging.

SDKnetworkadbPort configuration

SDKNetwork is not enabled by defaultadb, needs to be turned on manually in developer options. Setting-System-Advanced-Developer options-Open net adb

networkadbuse

This section assumes that the device'sipfor192.168.1.5, this will be used belowipEstablishadbConnect and debug the device.

1,firstAndroidThe device needs to be started first, if so, you can confirmadbWhether to start (PScommand to view). 2 ,existPCmachinemecmd, enter:

adb connect 192.168.1.5:5555

If the connection is successful, relevant prompts will be displayed. If it fails, you can firstkill-servercommand and retry the connection.

adb kill-server

3. If the connection has been established, in the R&D machine, you can enter adb relevant commands to debug. For example, entering adb shell will pass TCP/IP to the device. And USB Debugging is the same.

4. After debugging is completed, enter the following command on the R&D machine to disconnect:

```
adb disconnect 192.168.1.5:5555
```

Manually modify the network adb port number

If the SDK is not joined, the adb port number configuration, or want to modify it, the port number can be modified as follows: 1, first pass normally USB Connect to the target machine, in windows cmd execute under adb shell Enter. 2, set up adb listening port:

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

3, pass PS command search adb of pid

4, restart adb

```
# kill -9 $pid
```

After restarting, automatically restart after process adb. After restarting, I found that the setting of service.adb.tcp.port, it will automatically change to listen for network requests.

ADB Detailed explanation of commonly used commands

(1) Check the equipment status

View the connections to your computer Android Device or emulator:

```
adb devices
```

The returned result is the connection to the development machine. The serial number of the device or IP and port number (Port), state. (2) Install apk

will specify apk install the file onto the device:

```
adb install <apkFile path>
```

Examples are as follows:

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

Reinstall the app:

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

(3) Uninstall apk

Complete uninstall:

```
adb uninstall <package>
```

Examples are as follows:

```
adb uninstall com.wishtv
```

(4) User removes apk document:

Examples are as follows:

Example description: Remove "system/app"under the directory "WishTV.apk" document. (5) into the device and emulatorshell Enter the device or emulatorshell environment:

(6) Upload files from computer to device

usepushThe command can upload any file or folder on the local computer to the device. The local path generally refers to the local computer; the remote path generally refers toadbConnected single board devices.

adb push <Local path> <Remote path> Examples
are as follows:

Example description: Change the local "WishTV.apk"File upload toAndroidsystematic"system/app"Under contents. (7)

Download files from device to computer

pullThe command can download files or folders on the device to the local computer.

Examples are as follows:

Example description: putAndroidsystem"system/app"Download files or folders in the directory to local "F:\\"Under contents. (8)

)CheckbugReport

To view all error message reports generated by the system, you can runadb bugreportTo achieve this, the instruction willAndroidsystematic dumpsys,dumpstateandlogcatThe information is displayed. (9) View the system information of the device

existadb shellClick below for specific commands to view device system information.

Logcattool

AndroidThe log system provides the function of recording and viewing system debugging information. Logs are recorded from the buffers of various software and some systems. The buffers can beLogcatto view and use.LogcatIt is used for debugging programs

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

Examples are as follows:

```
adb shell  
logcat
```

Commonly used log filtering methods

Several ways to control log output:

- Control log output priority

Examples are as follows:

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

Example description: The display priority is warning or higher log information.

- Control log labels and output priority

Examples are as follows:

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

Example description: Supports all log information except those labeled "ActivityManager" and the priority is "Info". The above, labeled "MyApp" and the priority is ""Debug". Above.

- Only output logs with specific tags

Examples are as follows:

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

or

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

Example description: Only output tags WishTV log.

- Only output logs with specified priority and label

Examples are as follows:

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

Example description: Only the output priority is I, labeled WishTV log.

Procranktool

ProcrankyesAndroidA built-in debugging tool that runs on the device sideshellenvironment, it is used to output the memory snapshot of the process to facilitate effective observation of the memory usage of the process.

Includes the following memory information:

- VSS:Virtual Set SizeVirtual memory size consumed (including memory occupied by shared libraries) RSS:Resident
- Set SizeActual physical memory size used (including memory occupied by shared libraries)
- PSS:Proportional Set SizeActual physical memory size used (proportional allocation of memory occupied by shared libraries) USS:Unique Set
- SizeThe size of the physical memory occupied by the process alone (excluding the memory occupied by shared libraries) Note:
 - USSThe size represents the memory size that is only used by this process. It will be completely recycled after the process is killed; VSS/RSSContains the memory used by shared libraries and has no reference value for viewing the memory status of a single process; PSSIt is the occupation of the shared memory area by a single process after the shared memory is divided according to proportion.

useprocrank

implementprocrankYou need to let the terminal get it beforerootPermissions su

Command format:

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

Description of common instructions:

- v:according toVSSsort
- r:according toRSSsort
- p:according toPSSsort
- u:according toUSSsort
- R: Convert to ascending [descending] sorting
- w: only showworking setStatistical count of
- W:Resetworking setStatistical count of
- h:help

Example:

Output memory snapshot:

```
procrank
```

according toVSSOutput memory snapshots in descending order:

```
procrank -v
```

defaultprocrankThe output is viaPSSSort.

Retrieve specified content information

To view the memory usage status of a specified process, the command format is as follows:

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

incmdlineIndicates the name of the application to be found,PIDIndicates the application process to be found. outputsystemUIProcess memory usage status:

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

or:

```
procrank | grep 3396
```

Track process memory status

By tracking the memory usage status, we can analyze whether there is a memory leak scenario in the process. Use scripting to continuously output memory snapshots of the process, and compare the USsegment, you can learn whether there is a memory leak in this process. Example: Output process name com.android.systemui The application memory usage status to check whether there is any leakage: 1, writing script test.sh

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

2, pass adbOnce the tool is connected to the device, run this script: ./test.sh

Dumpsystool

Dumpsystool is an Android debugging tool that comes with the system and runs on the device sideshellEnvironment, provides status information function of services running in the system. A running service is an Android binderThe server process in the mechanism. dumpsysConditions for output printing:

- 1, can only print what has been loaded into ServiceManager services in;
- 2, if in the server code dumpIf the function is not implemented, no information is output.

use Dumpsys

- CheckDumpsyshelp

Function: output dumpsys help information.

```
dumpsys -help
```

- CheckDumpsysContains service list

Function: output dumpsys All printable service information, developers can focus on the name of the service that needs to be debugged.

```
dumpsys -l
```

- Output information about the specified service

Function: Output the specified service dump information.

Format: dumpsys [servicename] Example: Output ServiceSurfaceFlinger

Information, executable command:

```
dumpsys SurfaceFlinger
```

- Output information about specified services and processes

Function: Output the specified application process information for the specified service.

Format: dumpsys [servicename] [application name]

Example: Output service name meminfo, the process name is com.android.systemui Memory information, execute the command:

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

Note: Service names are case-sensitive and the full service name must be entered.

Last logturn on

- existdtsAdd the following two nodes to the 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>;  
};
```

- View in machine last log 130 |
root@rk3399:/sys/fs/pstore dmesg - #ls
ramoops-0 Last kernel panic saved laterlog. Last
pmsg-ramoops-0 user space log, androidoflog.
ftrace-ramoops-0 Print within a certain time period function trace.
console-ramoops-0 last_log Last started kernel log, but only saves the priority than the default log level High
log.

- Instructions:

```
cat dmesg-ramoops-0  
cat console-ramoops-0  
logcat -L(pmsg-ramoops-0) pass logcat Take it out and parse it pull out by logcat and parse cat ftrace-  
ramoops-0
```

FIQmodel

When the device crashes or gets stuck, you can input it through the serial port.fiq Use the command to view the status of the system. The specific command is as follows:

```
127|console:/ $fiq  
debug> help  
FIQ Debugger commands:  
pc PC status  
regs Register dump  
allregs Extended Register dump  
bt Stack trace  
reboot [<c>] Reboot with command <c> Hard  
reset [<c>] reset with command <c> Interrupt  
irqs status  
kmsg Kernel log  
version Kernel version  
sleep Allow sleep while in FIQ Disable  
nosleep sleep while in FIQ Switch terminal  
console to console Current CPU  
cpu
```

```
cpu <number>     Switch to CPU<number>
PS                  Process list
sysrq              sysrq options
sysrq <param> Execute sysrq with <param>
```

logAutomatic collection

- Collected content

```
android:android    log
kernel:kernel      log
```

- Open method
- turn onDeveloper options
- Setting-System-Advanced-Developer options-Android bug collector
- logsave route

```
data/vendor/logs/
```

common problem

current kernel and u-boot Version?

```
Android 11.0 corresponding kernel The version is: develop-4.19, u-boot The branches are next-dev branch
```

How to get the current SDK corresponding RK release Version

Rockchip Android 11.0 SDK includes AOSP original code and RK. There are two parts of the modified code, among which RK is contained in

.repo/manifests/include under the directory xml/middle, AOSP Default repository
exists .repo/manifests/default.xml.

Version confirmation:

- RK Modify part

```
vim .repo/manifests/include/rk_checkout_from_aosp.xml
<project groups="pdk" name="platform/build" path="build/make" remote="rk" revision="refs/
tags/android-11.0-mid-rkr1">
```

illustrate RK The version is android-11.0-mid-rkr1

- AOSP part

```
vim .repo/manifests/default.xml
<default revision="refs/tags/android-10.0.0_r14"...>
```

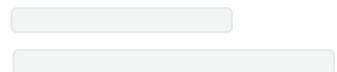
illustrate AOSP The version is android-10.0.0_r14 When version information needs to be provided, just provide the above two version information. A single warehouse can be obtained directly through the following command tag information

RKThe version is based onandroid-11.0-mid-rkrxxThe format is incremental, so the current

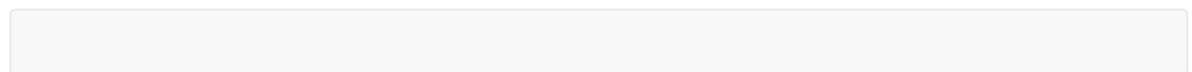
•

•

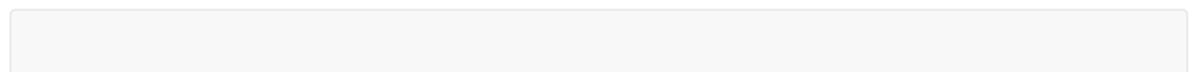
•



•



•



```

cd device/rockchip/rk356x
diff --git a/rk3566_rgo/BoardConfig.mk b/rk3566_rgo/BoardConfig.mk index
    be9020a..306a72d 100644 a/
--- rk3566_rgo/BoardConfig.mk b/
+++ rk3566_rgo/BoardConfig.mk
@@ -15,7 +15,7 @@
#
include device/rockchip/rk356x/BoardConfig.mk
BUILD_WITH_GO_OPT := true
-PRODUCT_UBOOT_CONFIG := rk3566
+PRODUCT_UBOOT_CONFIG := rk3566-nand
PRODUCT_KERNEL_DTS := rk3566-rk817-tablet
CAMERA_SUPPORT_AUTOFOCUS := true

```

Shutdown charging and low battery pre-charging

Shutdown charging and low battery pre-charging can be done in dts configuration, as follows:

```

charge-animation {
    compatible = "rockchip,uboot-charge";
    rockchip,uboot-charge-on = <1>;
    rockchip,android-charge-on rockchip,uboot-
    low-power-voltage rockchip,screen-on-voltage<3400>;
    <3500>; status = "okay";

};

```

in:

rockchip,uboot-charge-on:ubootShut down and charge, with androidShut down and charge mutually exclusive rockchip,android-charge-on:androidShut down and charge, with ubootShut down and charge mutually exclusive rockchip,uboot-low-power-voltage: Configure low-power precharge to startup voltage, which can be configured according to actual needs. rockchip,screen-on-voltage: Configure the voltage for low-power precharging to bright screen, which can be configured according to actual needs.

BoxMachine standby and fake shutdown function

BOXThe latest version already supports low-power true standby function by default, short press power key to verify functionality;

If you need a false shutdown (wake up and restart) function, please note that this function must PD_PMU of pwm can be supported, for example:

```

&pwm3 {
    status = "okay";

    compatible = "rockchip,remotectl-pwm";
    remote_pwm_id = <3>;
    handle_cpu_id = <1>;
    remote_support_psci = <0>;
    pinctrl-names = "default"; pinctrl-0
    =
    <&pwm3_pins>.....
}


```

```

diff --git a/arch/arm64/boot/dts/rockchip/rk3566-box.dtsi b/arch/arm64/
boot/dts/rockchip/rk3566-box.dtsi index
    7818060..58ffe6b 100644 a/arch/arm64/boot/dts/rockchip/
--- rk3566-box.dtsi b/arch/arm64/boot/dts/rockchip/rk3566-
+++ box.dtsi

```

```

@@ -309,6 +309,7 @@
    | RKPM_SLP_32K_PVTM
    )
>;
+    rockchip,virtual-poweroff      = <1>;
    rockchip,wakeup-config        = <
(0
| RKPM_PWM0_WKUP_EN
.....
>

```

UbootStage charging picture packaging and replacement

The charging image path can directly replace the file with the same name, and the format requirements are the same as the original file.

```

u-boot/tools/images/
├── battery_0.bmp
├── battery_1.bmp
├── battery_2.bmp
├── battery_3.bmp
├── battery_4.bmp
├── battery_5.bmp
└── battery_fail.bmp

```

if openubootCharging, but no charging picture is displayed. It may be that the picture is not packaged.resource.img, you can package it by following the command

```

cd u-boot
./scripts/pack_resource.sh ..//kernel/resource.img cp
resource.img ..//kernel/resource.img

```

After executing the above commandubootCharging pictures will be packaged to kernel directoryresource.img, at this time it is necessary to change the resource.imgPack to boot.imgin, can be in androidExecute in root directory./mkimage.sh, and then programrockdev/belowboot.imgThat's it.

RM310 4GConfiguration

4GFunctionSDKIt is turned off by default. To turn it on, do the following:

```

vim device/rockchip/common/BoardConfig.mk
# for rk 4g modem
- BOARD_HAS_RK_4G_MODEM ?= false
+ BOARD_HAS_RK_4G_MODEM ?= true

```

WIFI Hibernation policy configuration

wifiThe default sleep policy is to stay connected during hibernation. If you need to disconnect during hibernation, do the following:

```
--  
a/rk3566_rgo/overlay/frameworks/base/packages/SettingsProvider/res/values/default_ts.xml
```

```
+++  
b/rk3566_rgo/overlay/frameworks/base/packages/SettingsProvider/res/values/default_ts.xml
```

```
@@ -24,5 +24,5 @@
```

```
You can configure persist.wifi.sleep.delay.ms to delay closing wifi. The default is 15  
minutes, 0 means that the wifi is turned off immediately after the screen is off. -->
```

```
-     <integer name="def_wifi_sleep_policy">2</integer>  
+     <integer name="def_wifi_sleep_policy">0</integer>  
</resources>
```

Recoveryrotating configuration

supportRecoveryrotate0/90/180/270degree, the default is not to rotate (i.e. rotate0degree), the rotation configuration is described as follows:

```
vim device/rockchip/common/BoardConfig.mk  
#0: ROTATION_NONE    rotate0Spend  
#90: ROTATION_RIGHT   rotate90Spend  
#180: ROTATION_DOWN   rotate180Spend  
#270: ROTATION_LEFT   rotate270Spend  
#For Recovery Rotation  
TARGET_RECOVERY_DEFAULT_ROTATION ?=ROTATION_NONE
```

Android Surfacerotate

AndroidThe system displays rotation. You can modify the following configuration. The configuration parameters are0/90/180/270

```
# For Surface Flinger Rotation  
SF_PRIMARY_DISPLAY_ORIENTATION ?= 0
```

replaceAOSPPartial source coderemote

Customer downloadRKofreleaseThe code is slower and can beAOSPofremoteModify it to domestic mirror source, foreign customers can modify it to Googlemirror source. This improves download speed. The specific operation method is as follows:

implementrepo init(or unzipbasepackage), modify.repo/manifests/remote.xml, put theAOSPthis remote
offetchfrom

```
< remote name="aosp" fetch="./" review="https://10.10.10.29" />
```

Change to

Domestic customers: (Tsinghua University mirror source is used as an example in China, and it can be modified to other domestic mirror sources as needed)

```
< remote name="aosp" fetch="https://aosp.tuna.tsinghua.edu.cn" />;
```

Overseas customers: (GoogleMirror source)

```
<remote name="aosp" fetch="https://android.googlesource.com" />
```

userdataarea file system toEXT4

defaultdataThe file system of the partition isf2fs, it is recommended that products without batteries can bedataThe file system of the area is changed toext4, which can reduce the probability of data loss after abnormal power outage. The modification method is as follows:

byrk3566_rFor example:

```
device/rockchip/common$ git diff
diff --git a/scripts/fstab_tools/fstab.in b/scripts/fstab_tools/fstab.in index
      6e78b00..a658332 100755 a/scripts/
---  fstab_tools/fstab.in b/scripts/
+++  fstab_tools/fstab.in
@@ -20,6 +20,6 @@ ${_block_prefix}system_ext /system_ext $          ext4ro,barrier=1
{_flags},first_stage_
#For sdmmc
/devices/platform/${_sdmmc_device}/mmc_host*           auto   auto   defaults
voldmanaged=sdcard1:auto
# Full disk encryption has less effect on rk3326, so default to enable this.
-/dev/block/by-name/userdata /data f2fs
noatime,nosuid,nodev,discard,reserve_root=32768,resgid=1065
latemount,wait,check,fileencryption=aes-256-xts:aes-256- cts:v2
+inlinecrypt_optimized,quota,formattable,reservedsize=128M,checkpoint=fs
+ # /dev/block/by-name/userdata /data f2fs
noatime,nosuid,nodev,discard,reserve_root=32768,resgid=1065
latemount,wait,check,fileencryption=aes-256-xts:aes-256- cts:
v2+inlinecrypt_optimized,quota,formattable,reservedsize=128M,checkpoint=fs
# for ext4
- # /dev/block/by-name/userdata      /data      ext4
discard,noatime,nosuid,nodev,noauto_da_alloc,data=ordered,user_xattr,barrier=1
latemount,wait,formattable,check,fileencryption=software,quota,reservedsize=128M, checkpoint =block

+/dev/block/by-name/userdata      /data      ext4
discard,noatime,nosuid,nodev,noauto_da_alloc,data=ordered,user_xattr,barrier=1
latemount,wait,formattable,check,fileencryption=software,quota,reservedsize=128M, checkpoint= block
```

```
device/rockchip/rk356x$ git diff
diff --git a/rk3566_r/recovery.fstab b/rk3566_r/recovery.fstab index
      7532217..cf789ac 100755 a/
---  rk3566_r/recovery.fstab b/
+++  rk3566_r/recovery.fstab
@@ -7,7 +7,7 @@
/dev/block/by-name/odm           /odm      ext4
defaults             defaults
/dev/block/by-name/cache        /cache    ext4
defaults             defaults
/dev/block/by-name/metadata     /metadata ext4
defaults             defaults
-/dev/block/by-name/userdata    /data     f2fs
defaults             defaults
+ /dev/block/by-name/userdata   /data     ext4
defaults             defaults
/dev/block/by-name/cust         /cust    ext4
defaults             defaults
/dev/block/by-name/custom       /custom   ext4
defaults             defaults
/dev/block/by-name/radical_update defaults ext4
```

Modify the power on and off animation and power on and off ringtones

Reference documentation:

RKDocs\android\Rockchip_Introduction_Android_Power_On_Off_Animation_and_Tone_Customization_CN&EN.pdf

APPSet performance mode

device/rockchip/rk3xxx/The following configuration file:package_performance.xml, add the package name that requires performance mode to the node: (useaapt dump badging (file_path.apk)Get package name)

```
< app package="Package names" mode="Whether to enable acceleration, enable is1, closed as0"/>
```

For example, the reference for AnTuTu is as follows:

```
< app package="com.antutu.ABenchMark" mode="1"/>
< app package="com.antutu.benchmark.full" mode="1" />
< app package="com.antutu.benchmark.full" mode="1" />
```

The files are packaged into the firmware during compilation.

GPURelated troubleshooting methods

Please refer to the following documents to do preliminary troubleshooting.

RKDocs\android\Rockchip_User_Guide_Dr.G_CN&EN.pdf

OTPanfuseillustrate

OTPSupport chip

- RK3326
- PX30
- RK3566
- RK3568

EFUSESupport chip

- RK3288
- RK3368
- RK3399

firmware signature andotp/efuseBurning reference documentation

RKDocs\common\security\Rockchip-Secure-Boot-Application-Note-V1.9.pdf

How to determine the device in the codeOTP/EFUSEHas it been programmed?

OTP/EFUSEstatus will pass kernel of cmdlineto pass on, cmdlinemiddle fuse.programmed used to identify OTP/EFUSEstatus, as follows:

- "fuse.programmed=1": Software firmware package has been carried out secure-bootSignature, hardware device efuse/otphas been burned.

- "fuse.programmed=0": Software firmware package has been carried outsecure-bootSignature, hardware deviceefuse/otpNot burned.
- cmdlineNonefuse.programmed: Software firmware package is not carried outsecure-bootsign(Miniloadernot passed), orMiniloaderToo old to support passing.

switchselinux

Modify as follows, falseTo close, true to open

```
device/rockchip/common$  
--- a/BoardConfig.mk  
+++ b/BoardConfig.mk  
@@ -67,7 +67,7 @@ endif  
  
# Enable android verified boot 2.0  
BOARD_AVB_ENABLE ?= false  
- BOARD_SELINUX_ENFORCING ?= false  
+ BOARD_SELINUX_ENFORCING ?= true
```

"Popup pops up"AndroidThere is a problem with the system" warning

There are two reasons why the warning box appears:

- 1.Firmware mismatch, system/boot/vendorthreefingerprintInconsistent, not the same set of firmware.
- 2.The machine is enabled and supported!Odebugging functionconfig, when compiling, use the kernel compilation command mentioned earlier in the document to turn it off.
- 3.For needs useIOProjects with debugging functions can be directly merged regardless of the above two reasons.frameworks/basedown patchRemove pop-ups:

```
diff --git  
a/services/core/java/com/android/server/wm/ActivityTaskManagerService.java b/services/core/  
java/com/android/server/wm/ActivityTaskManagerService.java index  
595c340..d4e495a 100644 a/services/core/java/com/android/server/wm/  
--- ActivityTaskManagerService.java b/services/core/java/com/android/server/wm/  
+++ ActivityTaskManagerService.java  
@@ -6555,7 +6555,7 @@ public class ActivityTaskManagerService extends  
IActivityTaskManager.Stub {  
    } catch (RemoteException e) {}  
  
-        if (!Build.isBuildConsistent()) { if (0 && !  
+        Build.isBuildConsistent() {  
            Slog.e(TAG, "Build fingerprint is not consistent, warning  
user");  
            mUiHandler.post(() -> {  
                if (mShowDialogs) {
```

How to open the Ethernet setting item in settings

There is no Ethernet setting option in the system settings by default. If Ethernet is required in the project, it can be opened according to the following configuration:

```
-- a/BoardConfig.mk
+++ b/BoardConfig.mk
@@ -146,3 +146,6 @@ endif
ifeq ($(strip $(BOARD_USES_AB_IMAGE)), true)
DEVICE_MANIFEST_FILE :=
device/rockchip/$(TARGET_BOARD_PLATFORM)/manifest_ab.xml endif

+ # for ethernet
+ BOARD_HS_EETHERNET := true
```

aboutAVBandsecurity bootOperation

AVBandsecurity bootOperational reference documentation

RKDocs/common/security/RK356X_SecurityBoot_And_AVB_instructions_CN.pdf

IOCommand cannot be used

The command requires dependencies DEVMEM and DEVMEMIt is turned off by default, so it results in IOUnable to use by default, use if needed for debugging. The command can be modified as follows:

```
wlq@ubuntu:~/1_source_code/a3_rk3399_android10.0_29/kernel$ vim kernel/
configs/android-11.config
```

in the case of GO products need to be modified:

```
wlq@ubuntu:~/1_source_code/a3_rk3399_android10.0_29/kernel$ vim kernel/
configs/android-11-go.config
```

Delete the following line:

```
# CONFIG_DEVME is not set
```

```
wlq@ubuntu:~/1_source_code/a3_rk3399_android10.0_29/kernel$ vim configs/r/
android-4.19/android-base.config
```

Delete the following line:

```
# CONFIG_DEVME is not set
```

The above two need to be deleted at the same time

SNcommand rules

The number must start with a letter and the length 14 within bytes.

RK3288compiled newspaper LZ4mistake

RK3288compilekernelWhen reporting the following error:

```
SORTEX vmlinux
SYSMAP System.map
OBJCOPY arch/arm/boot/Image
Kernel: arch/arm/boot/Image is ready
SHIPPED arch/arm/boot/compressed/hyp-stub.S
SHIPPED arch/arm/boot/compressed/fdt_rw.c
SHIPPED arch/arm/boot/compressed/fdt.h
SHIPPED arch/arm/boot/compressed/libfdt.h
SHIPPED arch/arm/boot/compressed/libfdt_internal.h
SHIPPED arch/arm/boot/compressed/fdt_ro.c
SHIPPED arch/arm/boot/compressed/fdt_wip.c
SHIPPED arch/arm/boot/compressed/fdt.c
SHIPPED arch/arm/boot/compressed/libifuncs.S
SHIPPED arch/arm/boot/compressed/ashldis.S
SHIPPED arch/arm/boot/compressed/bswapsdi2.S
LDS arch/arm/boot/compressed/vmlinux.lds
AS arch/arm/boot/compressed/head.o
LZ4 arch/arm/boot/compressed/piggy_data
Incorrect parameters
Usage :
    lz4 [arg] [input] [output]
Arguments :
    -1 : Fast compression (default)
    -9 : High compression
    -d : decompression (default for .lz4 extension)
    -z : force compression
    -f : overwrite output without prompting
    -h/-H : display help/long help and exit
arch/arm/boot/compressed/Makefile:191: recipe for target 'arch/arm/boot/compressed/piggy_data' failed
make[2]: *** [arch/arm/boot/compressed/piggy_data] Error 1
arch/arm/boot/Makefile:71: recipe for target 'arch/arm/boot/compressed/vmlinux' failed
make[1]: *** [arch/arm/boot/compressed/vmlinux] Error 2
arch/arm/Makefile:351: recipe for target 'zImage' failed
make: *** [zImage] Error 2
```

problem causes:

The system comes with lz4. The version is too low, please 1.8.3 and above.

```
wlq@ubuntu:~$lz4 -v
*** LZ4 command line interface 64-bit v1.8.3, by Yann Collet ***
refusing to read from a console
```

Solution:

Direct copy android compiled lz4. Covering the system lz4

```
sudo cp out/host/linux-x86/bin/lz4 /usr/bin/lz4
```

RKR7 Previous version updated to RKR7 and above version (orota Machines with batteries that cannot be turned on after upgrading) reboot loader Problems such as commands not working

problem causes:

RKR7 Previous version (less than or equal to RKR6) There is a probability of being unable to shut down, RKR7 Version kernel. The following submissions address this issue:

```
commit 65a3eeeaa668ca570009caa1423f9780d44efebbf
Author: Wu Liangqing <wlq@rock-chips.com> Date: Fri Apr 23
17:11:25 2021 +0800

    arm64: dts: rockchip: add not-save-power-en to rk356x board

    fix reboot block as follows log:
        15.874382] binder: release 247:268 transaction 4234 in, still active
        [ 15.874418] binder: send failed reply for transaction 4234 to 395:455
        [ 15.959849] binder: undelivered TRANSACTION_ERROR: 29189
        [ 16.085993] binder: 147:147 transaction failed 29189/-22, size 100-0 line
3059
        [ 16.128154] android_work: sent uevent USB_STATE=DISCONNECTED
        [ 16.145570] logd.klogd: 24 output lines suppressed due to ratelimiting
```

```

[ 16.690141] cpu cpu0: min=816000, max=816000
[ 16.696558] rk808 0-0020: reboot: force RK817_RST_FUNC_REG ok!
[ 33.769778] vcc5v0_otg: disabling
[ 33.770099] vcc3v3_lcd0_n: disabling
[ 33.770424] vcc3v3_lcd1_n: disabling
[ 37.699768] rcu: INFO: rcu_preempt detected stalls on CPUs/tasks:
[ 37.700342] rcu: 3...0: (0 ticks this GP)
idle=b52/1/0x4000000000000000 softirq=4194/4194 fqs=2012
[ 37.701150] rcu: (detected by 0, t=6302 jiffies, g=3301, q=16)
[ 37.701684] Task dump for CPU 3:
[ 37.701981] init R running task 0 1 0
0x0400000a
[ 37.702609] Call trace:
[ 37.702851] __switch_to+0xe4/0x138
[ 37.703166] lock_timer_base+0x5c/0xa0
[ 37.703502] try_to_del_timer_sync+0x30/0x98
[ 37.703883] del_timer_sync+0x50/0x60
[ 37.704220] schedule_timeout+0x19c/0x478
[ 37.704582] clk_gate_endisable+0x2c/0xc8
[ 100.716421] rcu: INFO: rcu_preempt detected stalls on CPUs/tasks:
[ 100.716991] rcu: 3...0: (0 ticks this GP)
idle=b52/1/0x4000000000000000 softirq=4194/4194 fqs=7921
[ 100.717799] rcu: (detected by 0, t=25207 jiffies, g=3301, q=19)
[ 100.718334] Task dump for CPU 3:
[ 100.718632] init R running task 0 1 0
0x0400000a
[ 100.719260] Call trace:
[ 100.719500] __switch_to+0xe4/0x138
[ 100.719816] lock_timer_base+0x5c/0xa0
[ 100.720152] try_to_del_timer_sync+0x30/0x98
[ 100.720533] del_timer_sync+0x50/0x60
[ 100.720870] schedule_timeout+0x19c/0x478
[ 100.721231] clk_gate_endisable+0x2c/0xc8
[ 163.733075] rcu: INFO: rcu_preempt detected stalls on CPUs/tasks:
[ 163.733643] rcu: 3...0: (0 ticks this GP)
idle=b52/1/0x4000000000000000 softirq=4194/4194 fqs=13833
[ 163.734462] rcu: (detected by 0, t=44112 jiffies, g=3301, q=20)
[ 163.734997] Task dump for CPU 3:
[ 163.735294] init R running task 0 1 0
0x0400000a
[ 163.735921] Call trace:
[ 163.736160] __switch_to+0xe4/0x138
[ 163.736475] lock_timer_base+0x5c/0xa0
[ 163.736811] try_to_del_timer_sync+0x30/0x98
[ 163.737192] del_timer_sync+0x50/0x60
[ 163.737528] schedule_timeout+0x19c/0x478
[ 163.737889] clk_gate_endisable+0x2c/0xc8

```

Change-Id: I663b6b3e0b081ad17bf2845629b34e2ec9d2d76d Signed-off-by: Wu Liangqing <wlq@rock-chips.com >

Because the premise for the above submission to take effect is when the computer is turned on. Some of the register configurations are restored to the default state, but since these register configurations will retain the last value even if the power supply is uninterrupted, the above patch will be used in tools and tools. It is invalid when upgrading, that is, the first time after upgrading, reboot. The operation may still get stuck. The solution is: cut off the power once and use pmicreset again.

Machines with batteries can be connected to the serial port or ADB. Enter the following command to clear PMIC

```
i2cset -yf 0 0x20 0x99 0x0 b  
i2cset -yf 0 0x20 0xa4 0x0 b
```

RK356X IO-Domain GPIO Voltage configuration confirmation, GPIO If the voltage is not configured correctly, it will cause the chip to GPIO burn out

RK3566/RK3568 of IO-Domain GPIO The voltage is configured as 3.3V, it actually needs to be configured according to the hardware schematic diagram to RK3566/RK3568 EVB Take the board as an example: IO-Domain The default configuration of:

```
arch/arm64/boot/dts/rockchip/rk3568-evb.dtsi  
&pmu_io_domains {  
    status = "okay";  
    pmuio2-supply = <&vcc3v3_pmu>;  
    vccio1-supply = <&vccio_acodec>;  
    vccio3-supply = <&vccio_sd>;  
    vccio4-supply = <&vcc_3v3>;  
    vccio5-supply = <&vcc_3v3>;  
    vccio6-supply = <&vcc_3v3>;  
    vccio7-supply = <&vcc_3v3>;  
};
```

According to the actual hardware schematic diagram it should be changed to:

```
&pmu_io_domains {  
    status = "okay";  
    pmuio2-supply = <&vcc3v3_pmu>;  
    vccio1-supply = <&vccio_acodec>;  
    vccio3-supply = <&vccio_sd>;  
    vccio4-supply = <&vcc_1v8>;  
    vccio5-supply = <&vcc_3v3>;  
    vccio6-supply = <&vcc_1v8>;  
    vccio7-supply = <&vcc_3v3>;  
};
```

IO-Domain The configuration process is as follows:

Step 1: Obtain the hardware schematic and confirm the design of the hardware power supply

This article is based on RK_EVB1_RK3568_DDR4P216SD6_V10_20200911 EVB The board is introduced as an example. Hardware schematic diagram: RK_EVB1_RK3568_DDR4P216SD6_V10_20200911.pdf Power supply solution: analysis from the hardware schematic diagram, EVB plate RK_EVB1_RK3568_DDR4P216SD6_V10_20200911 It's a belt PMU (RK809-5) plan **Step 2: Find the corresponding kernel dts Configuration file**

From the first step, it can be seen that the EVB The hardware power supply design of the board is with PMU scheme, corresponding kernel dts The configuration file is located at: arch/arm64/boot/dts/rockchip/rk3568-evb.dtsi (The solution discussed in this article) **Step 3: Modify the kernel dts Power domain configuration node pmu_io_domains**

```

&pmu_io_domains {
    status = "okay";
    pmuio2-supply = <&vcc3v3_pmu>;
    vccio1-supply = <&vccio_acodec>;
    vccio3-supply = <&vccio_sd>;
    vccio4-supply = <&vcc_1v8>;
    vccio5-supply = <&vcc_3v3>;
    vccio6-supply = <&vcc_1v8>;
    vccio7-supply = <&vcc_3v3>;
};

}

```

vccio1-supply For example, first check the hardware schematic diagram to confirm vccio1 power domain (VCCIO1) configuration as shown in the figure.

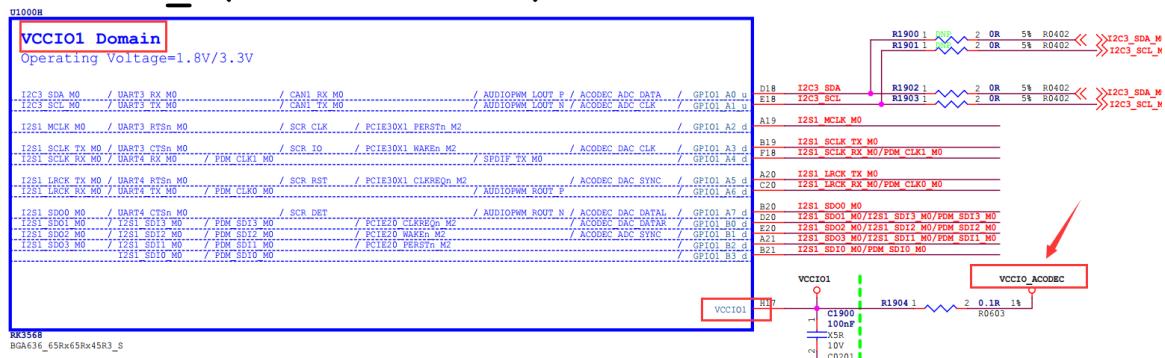
IO Power Domain Map

Updates must be Revision accordingly!

IO Domain	Pin Num	Support IO Voltage		Actual assigned IO Domain Voltage			Notes
		3.3V	1.8V	Supply Power Net Name	Power Source	Voltage	
PMUIO1	Pin Y20	✓	✗	VCC3V3_PMU	VCC3V3_PMU	3.3V	
PMUIO2	Pin W19	✓	✓	VCC3V3_PMU	VCC3V3_PMU	3.3V	
VCCIO1	Pin H17	✓	✓	VCCIO_ACODEC	VCCIO_ACODEC	3.3V	
VCCIO2	Pin H18	✓	✓	VCCIO_FLASH	VCC_1V8	1.8V	PIN "FLASH_VOL_SEL" must be logic High if VCCIO_FLASH=3.3V, FLASH_VOL_SEL must be logic low
VCCIO3	Pin L22	✓	✓	VCCIO_SD	VCCIO_SD	3.3V	
VCCIO4	Pin J21	✓	✓	VCCIO4	VCC_1V8	1.8V	
VCCIO5	Pin V10 Pin V11	✓	✓	VCCIO5	VCC_3V3	3.3V	
VCCIO6	Pin R9 Pin U9	✓	✓	VCCIO6	VCC_1V8	1.8V	
VCCIO7	Pin V12	✓	✓	VCCIO7	VCC_3V3	3.3V	

Search on the hardware schematic diagram **VCCIO1**, as follows

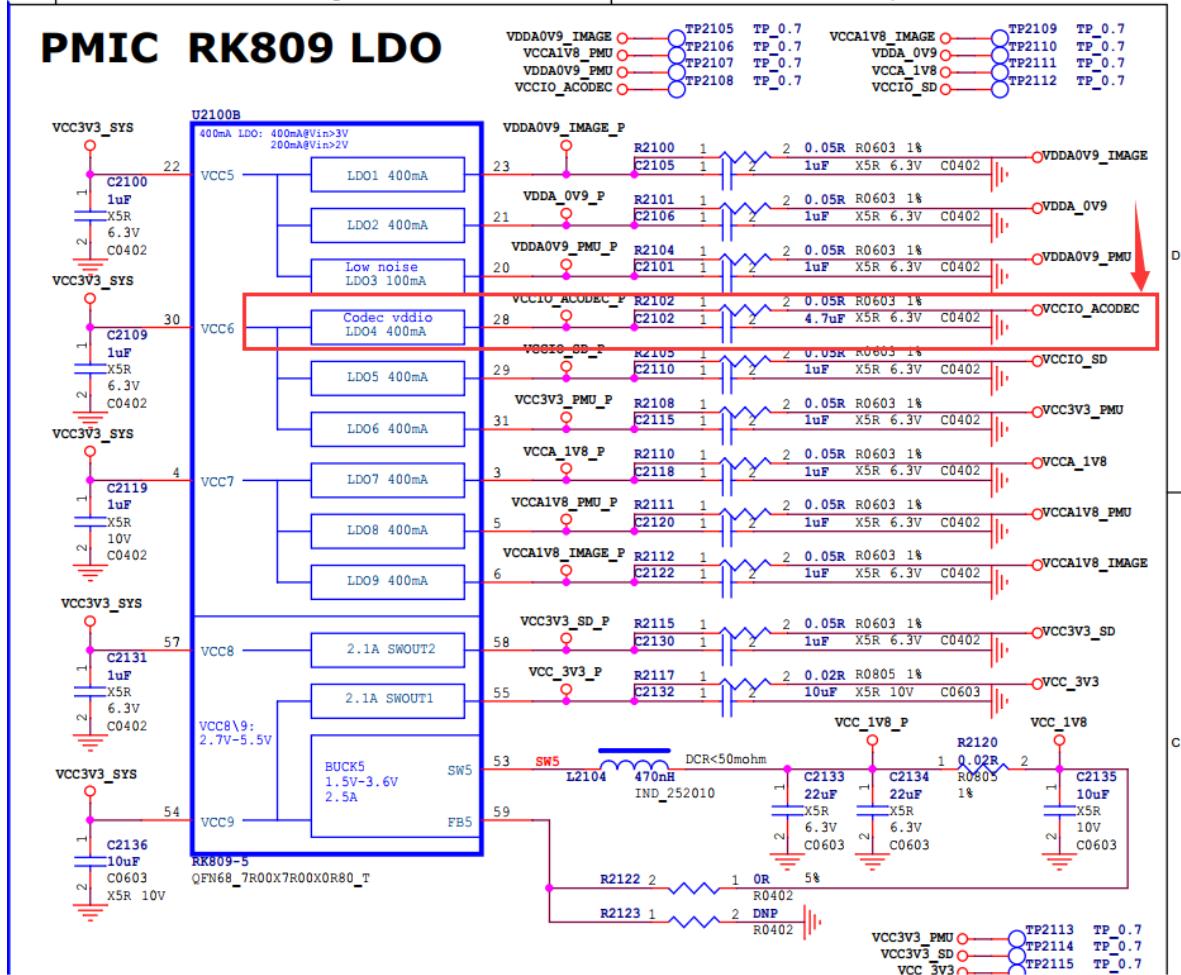
RK3568_H(VCCIO1 Domain)



Found from the picture above, the power supply is **vccio_acodec**.

Search on schematic **vccio_acodec**, can be found as shown below.

PMIC RK809 LDO



Found from the picture above **vccio_acodec** By RK809 of LDO4 powered by.

from software DTS found inside LDO_REG4(LDO4) configuration information, as follows

```

vccio_acodec: LDO_REG4 {
    regulator-always-on;
    regulator-boot-on;
    regulator-min-microvolt regulator-<3300000>;
    microvolt regulator-name =      = <3300000>;
    "vccio_acodec"; regulator-state-mem {

        regulator-off-in-suspend;
    };
}

```

Change the above **vccio_acodec** configured to **pmu_io_domains** in node **vccio1-supply** = <&vccio_acodec>; That's it become **vccio1** voltage configuration

```

&pmu_io_domains {
    status = "okay";
    pmuio2-supply = <&vcc3v3_pmu>;
    vccio1-supply = <&vccio_acodec>;
    vccio3-supply = <&vccio_sd>;
    vccio4-supply = <&vcc_1v8>;
    vccio5-supply = <&vcc_3v3>;
    vccio6-supply = <&vcc_1v8>;
    vccio7-supply = <&vcc_3v3>;
}

```

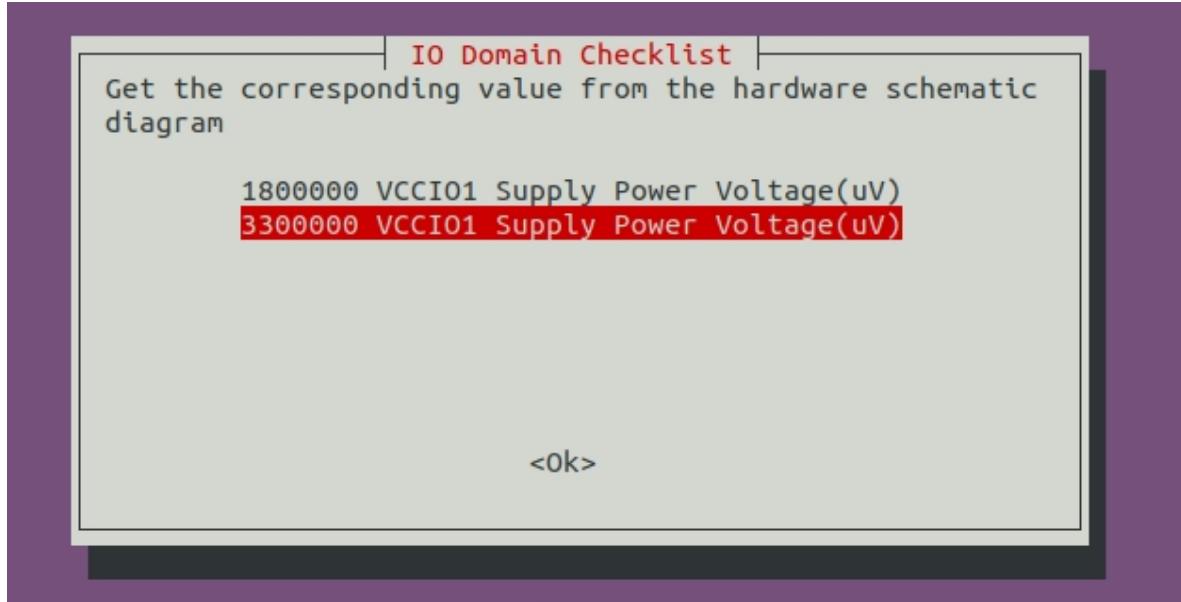
The other channels can be configured as above, among which vccio2The voltage is fixed by hardware and does not need to be configured.

- pmui02
- vccio1
- vccio3
- vccio5
- vccio6
- vccio7

GPIOThe voltage can be configured as above

RK356X kernelCompile popupIO-DomainConfirmation dialog

compilekernelThe following dialog box pops up:



The purpose of popping up this dialog box is to check the actual hardware schematic diagram and software dtsof GPIO whether the voltage matches needs to be selected based on the actual design voltage of the hardware schematic diagram (the value selected in the dialog box will not be saved to dtsmiddle.dts Need to be modified manually). If you are a software engineer, please check with your hardware engineer. This is very important. If GPIO Incorrect voltage configuration will cause the chip GPIO Burned out. When you confirm correctly GPIO After the voltage is set, this dialog box will no longer pop up (enter the value and dtsconfigured value), if dtsname or dtsinside io-domain If a change occurs, a pop-up will continue for confirmation.

RK356x PCIEModule related issues

PCIE3.0port if no external clock is provided to the chip but dtsIf this interface is enabled, it will cause the boot to freeze.

PCIE3.0Because an external clock is required, if dtsEnabledpcie3.0, but there is no external clock provided tork356x pcie3.0module (for example, if the external clock module is not attached, or the clock module works abnormally), it will cause the system to freeze. Problemslog As follows:

```
[    21.449927] rcu: INFO: rcu_preempt detected stalls on CPUs/tasks:  
[    21.449935] rcu: INFO: rcu_sched detected stalls on CPUs/tasks:  
[    21.449960] rcu:          1-...0: (1 GPs behind) idle=486/1/0x4000000000000000  
softirq=29/29      fqs=2057  
[21.449989] rcu:          1-...0: (87 ticks this GP) idle=486/1/0x4000000000000000  
softirq=7/29 fqs=2100  
[21.450003] rcu:          2-...0: (100 ticks this GP)  
idle=412/1/0x4000000000000000 [      softirq=7/52 fqs=2100  
[21.450010] rcu:          (detected by 0, t=6302 jiffies, g=-1183, q=1)  
[    21.450023] Task dump for CPU 1:  
[    21.450031]rk-pcie           R running task          0      53          2 0x0000002a
```

```
[ 21.450048] Call      trace:  
[ 21.450065] rcu:          2-...0: (15 ticks this GP) idle=412/1/0x4000000000000000  
softirq=37/52    fqs=2057  
[ 21.450071] rcu:          (detected by 3, t=6302 jiffies, g=-1147, q=720)  
[ 21.450084] Task dump for CPU 1:  
[ 21.450092]   rk-pcie           R running    task       0     53      2 0x0000002a  
[ 21.450125]   __switch_to+0xe4/0x138  
[ 21.450139]   kthread+0x12c/0x158  
[ 21.450150]   ret_from_fork+0x10/0x18  
[ 21.450156] Task dump for CPU 2:  
[ 21.450163]   kworker/u8:1      R running    task       0     32      2 0x0000002a  
[ 21.450178]   Call trace:  
[ 21.450199] Workqueue: events_unbound enable_ptr_key_workfn  
[ 21.450216]   __switch_to+0xe4/0x138  
[ 21.450224]   Call trace:  
[ 21.450239]   kthread+0x12c/0x158  
[ 21.450249]   ret_from_fork+0x10/0x18  
[ 21.450256] Task dump for CPU 2:  
[ 21.450270]   __switch_to+0xe4/0x138  
[ 21.450283]   bp_hardening_data+0x0/0x10 kworker/  
[ 21.450300]   u8:1 R running task       0     32      2 0x0000002a  
[ 21.450327] Workqueue: events_unbound enable_ptr_key_workfn  
[ 21.450340]   Call trace:  
[ 21.450356]   __switch_to+0xe4/0x138  
[ 21.450372]   bp_hardening_data+0x0/0x10  
[ 48.026601] watchdog: BUG: soft lockup - CPU#3 stuck for 22s! [swapper/0:1]  
[ 48.026638] Modules linked in:  
[ 48.026664] CPU: 3 PID: 1 Comm: swapper/0 Not tainted 4.19.172 #10  
[ 48.026676] Hardware name: Rockchip RK3568 EVB1 DDR4 V10 Board (DT)  
[ 48.026692] pstate: 80c00009 (Nzcv daif +PAN +UAO)  
[ 48.026716] pc: smp_call_function_many+0x314/0x350  
[ 48.026753] lr : smp_call_function_many+0x2d4/0x350  
.....  
[ 48.032654]   Call trace:  
[ 48.032678]   smp_call_function_many+0x314/0x350  
[ 48.032700]   smp_call_function+0x38/0x68  
[ 48.032729]   on_each_cpu+0x30/0x80  
[ 48.032759]   clock_was_set+0x1c/0x28  
[ 48.032787]   do_settimeofday64+0x130/0x180  
[ 48.032819]   rtc_hctosys+0x78/0x118  
[ 48.032846]   __rtc_register_device+0xd4/0x160  
[ 48.032874]   rk808_rtc_probe+0x174/0x2c0  
[ 48.032904]   platform_drv_probe+0x50/0xa8  
[ 48.032922]   really_probe+0x228/0x2a0  
[ 48.032940]   driver_probe_device+0x58/0x100  
[ 48.032969]   __device_attach_driver+0x90/0xc0  
[ 48.032995]   bus_for_each_drv+0x70/0xc8  
[ 48.033023]   __device_attach+0xec/0x148  
[ 48.033051]   device_initial_probe+0x10/0x18  
[ 48.033077]   bus_probe_device+0x94/0xa0  
[ 48.033103]   device_add+0x384/0x698  
[ 48.033130]   platform_device_add+0x108/0x248  
[ 48.033160]   mfd_add_device+0x2d8/0x358  
[ 48.033177]   mfd_add_devices+0xb0/0x170  
[ 48.033211]   devm_mfd_add_devices+0x78/0xe0  
[ 48.033238]   rk808_probe+0x724/0x780  
[ 48.033264]   i2c_device_probe+0x200/0x278  
[ 48.033296]   really_probe+0x228/0x2a0
```

```

[ 48.033323]     driver_probe_device+0x58/0x100
[ 48.033350]     __device_attach_driver+0x90/0xc0
[ 48.033377]     bus_for_each_drv+0x70/0xc8
[ 48.033404]     __device_attach+0xec/0x148
[ 48.033422]     device_initial_probe+0x10/0x18
[ 48.033438]     bus_probe_device+0x94/0xa0
[ 48.033463]     device_add+0x384/0x698
[ 48.033488]     device_register+0x1c/0x28
[ 48.033516]     i2c_new_device+0x1c0/0x398
[ 48.033546]     of_i2c_register_devices+0x134/0x160
[ 48.033573]     i2c_register_adapter+0x150/0x408
[ 48.033601]     __i2c_add_numbered_adapter+0x5c/0xa8
[ 48.033628]     i2c_add_adapter+0xa4/0xd8
[ 48.033655]     rk3x_i2c_probe+0x324/0x428
[ 48.033674]     platform_drv_probe+0x50/0xa8
[ 48.033708]     really_probe+0x228/0x2a0
[ 48.033736]     driver_probe_device+0x58/0x100
[ 48.033764]     device_driver_attach+0x6c/0x78
[ 48.033791]     __driver_attach+0xb0/0xf0
[ 48.033818]     bus_for_each_dev+0x68/0xc 8
[ 48.033844]     driver_attach+0x20/0x28
[ 48.033870]     bus_add_driver+0xf8/0x1f0
[ 48.033898]     driver_register+0x60/0x110
[ 48.033917]     __platform_driver_register+0x40/0x48
[ 48.033938]     rk3x_i2c_driver_init+0x18/0x20
[ 48.033966]     do_one_initcall+0x48/0x240
[ 48.033997]     kernel_init_freeable+0x210/0x37c
[ 48.034024]     kernel_init+0x10/0x108
[ 48.034052]     ret_from_fork+0x10/0x18

```

This appears log, if there is indeed an external clock module attached, check whether the module is working properly; if not, it means there is no need to use it.

Mouth, need to be indtasmiddleableThe following items:

```

&pcie3x1 {
    status = "disabled";
};

&pcie3x2 {
    status = "disabled";
};

```

rk356x pcie2x1controller andsata2Controllers cannot be turned on at the same time

because rk356x pcie2x1 controller andsata2The controllers are multiplexed and cannot be turned on at the same time, otherwise it will cause mutual interference and cause abnormalities.

Android SambaFunction

Reference documentation

RKDocs/android/Rockchip_Introduction_Android_Samba_CN.pdf

NFSstart up

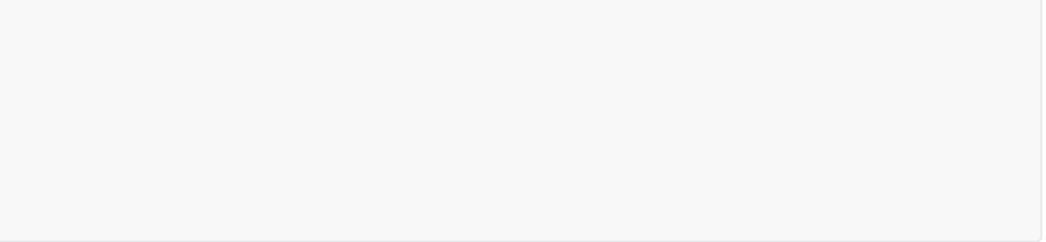
Reference documents and patches:

ReviseDDRfrequency

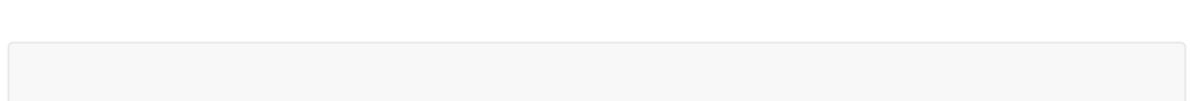
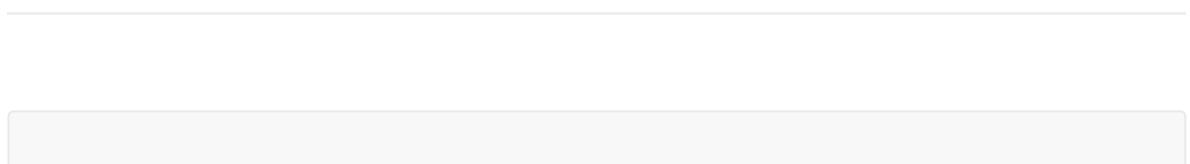
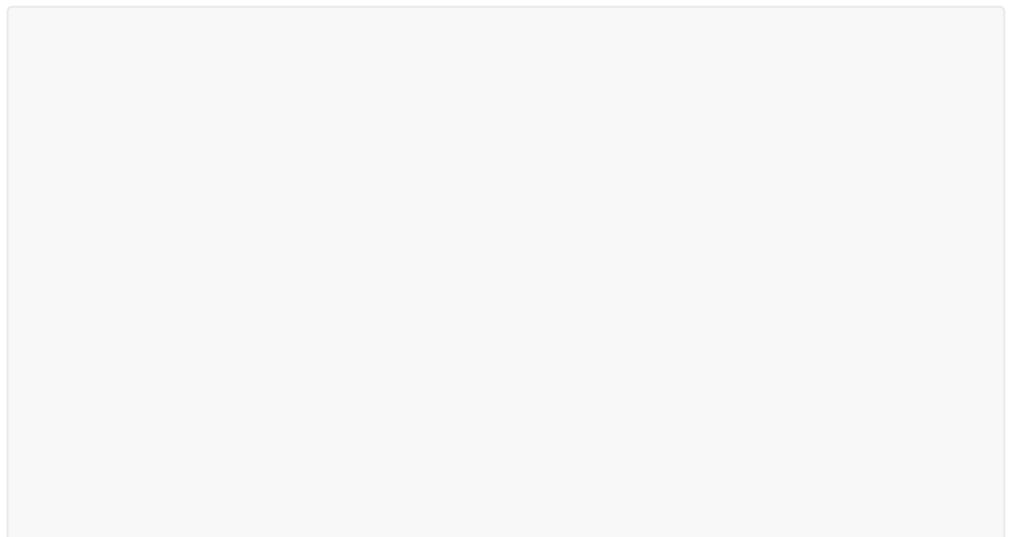
When the system is unstable, randomly stuck, or unable to boot, it may be due to DDRUnstable, you can reduce it firstDDRFrequency to troubleshoot problems. DDRThe frequency reduction method is as follows (withRK3568For example):

1.confirmRK3568Different frequencies supportedloaderWhat

2.ReviseRKBOOT/RK3568MINIALL.ini(This file name is different for different chips), assuming you want to modifyDDRFrequency of



Delete more than



Choosing a Branch

Some requirements for the build environment are determined by the version of the source code you plan to compile. For a full list of available branches, see Build Numbers. You can also choose to download and build the latest source code (called master), in which case you will simply omit the branch specification when you initialize the repository.

After you have selected a branch, follow the appropriate instructions below to set up your build environment.

Setting up a Linux build environment

These instructions apply to all branches, including master.

The Android build is routinely tested in house on recent versions of Ubuntu LTS (14.04) and Debian testing. Most other distributions should have the required build tools available.

For Gingerbread (2.3.x) and newer versions, including the master branch, a 64-bit environment is required. Older versions can be compiled on 32-bit systems.

Note: See Requirements for the complete list of hardware and software requirements, then follow the detailed instructions for Ubuntu and Mac OS below.

Installing the JDK

The master branch of Android in the Android Open Source Project (AOSP) comes with prebuilt versions of OpenJDK below `prebuilts/jdk/` so no additional installation is required.

Older versions of Android require a separate installation of the JDK. On Ubuntu, use OpenJDK. See JDK Requirements for precise versions and the sections below for instructions.

For Ubuntu >= 15.04

Run the following:

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

For Ubuntu LTS 14.04

There are no available supported OpenJDK 8 packages for Ubuntu 14.04.

1. Download the .deb packages for 64-bit architecture from old-releases.ubuntu.com:

```
openjdk-8-jre-headless_8u45-b14-1_amd64.deb with SHA256  
0f5aba8db39088283b51e00054813063173a4d8809f70033976f83e214ab56c0  
openjdk-8-jre_8u45-b14-1_amd64.deb with SHA256  
9ef76c4562d39432b69baf6c18f199707c5c56a5b4566847df908b7d74e15849  
openjdk-8-jdk_8u45-b14-1_amd64.deb with SHA256  
6e47215cf6205aa829e6a0a64985075bd29d1f428a4006a80c9db371c2fc3c4c
```

2. Optionally, confirm the checksums of the downloaded files against the SHA256 string listed with each package above. For example, with the `sha256sum` tool:

```
sha256sum {downloaded.deb file}
```

3. Install the packages:

```
sudo apt-get update
```

Run dpkg for each of the .deb files you downloaded. It may produce errors due to missing dependencies:

```
sudo dpkg -i {downloaded.deb file}
```

To fix missing dependencies:

```
sudo apt-get -f install
```

Update the default Java version - optional

Optionally, for the Ubuntu versions above update the default Java version by running:

```
sudo update-alternatives --config java
sudo update-alternatives --config javac
```

Note: If, during a build, you encounter version errors for Java, see Wrong Java version for likely causes and solutions.

Installing required packages (Ubuntu 14.04)

You will need a 64-bit version of Ubuntu. Ubuntu 14.04 is recommended.

```
sudo apt-get install git-core gnupg flex bison gperf build-essential zip curl zlib1g-dev gcc-multilib g++-multilib libgcrypt11-dev-i386 lib32ncurses5-dev x11proto-core-dev libx11-dev lib32z-dev ccache libgl1-mesa-dev libxml2-utils xscreensaver unzip python-pyelftools python3-pyelftools device-tree-compiler libfdt-dev libfdt1 libssl-dev liblz4-tool python-dev
```

Note: To use SELinux tools for policy analysis, also install the python-networkx package. Note: If you are using LDAP and want to run ART host tests, also install the libnss-sss:i386 package.

Configuring USB Access

Under GNU/linux systems (and specifically under Ubuntu systems), regular users can't directly access USB devices by default. The system needs to be configured to allow such access.

The recommended approach is to create a file /etc/udev/rules.d/51-android.rules (as the root user) and to copy the following lines in it. must be replaced by the actual username of the user who is authorized to access the phones over USB.

```
# adb protocol on passion (Rockchip products)
SUBSYSTEM=="usb", ATTR{idVendor}=="2207", ATTR{idProduct}=="0010", MODE="0600",
OWNER="<username>"
```

Those new rules take effect the next time a device is plugged in. It might therefore be necessary to unplug the device and plug it back into the computer.

This is known to work on both Ubuntu Hardy Heron (8.04.x LTS) and Lucid Lynx (10.04.x LTS). Other versions of Ubuntu or other variants of GNU/linux might require different configurations. References :
<http://source.android.com/source/initializing.html>

appendixB SSHPublic key operation instructionsSSH public key operation instruction

appendixB-1 SSHPublic key generationSSH public key generation

Use the following command to generate:

```
ssh-keygen -t rsa -C "user@host"
```

please change user@hostReplace with your email address.

Please replace user@host with your email address.

After the command is completed, it will be generated in your directory.keydocument.

It will generate the key file in your directory after the command is executed successfully.

Please keep the generated private key file properly id_rsa and password, and replace id_rsa.pub Send email to SDK Administrator of the publishing server. Please keep carefully the generated private key file id_rsa and password, and send id_rsa.pub to SDK release server admin through email.

appendixB-2usekey-chainManage keysUse key-chain to manage the key

It is recommended that you use simpler tools keychainManage keys.

Recommend you use the simple tool keychain to manage the key. The specific usage is as follows:

The detailed usage is as follows:

1. Install keychain software package:

Install keychain software package:

```
$sudo aptitude install keychain
```

2. Configure the usage key:

Configure to use the key:

```
$vim ~/.bashrc
```

Add the following line:

Add the following command:

```
eval keychain --eval ~/.ssh/id_rsa`
```

in, id_rsa is the name of the private key file.

Among which, id_rsa is the file name of the private key. After the above configuration, log in to the console again and you will be prompted to enter a password. You only need to enter the password used when generating the key. If there is no password, you do not need to enter it.

Log in to the console again after configuring as above, and it will prompt to input the password. Only need to input the password used for generating the key if there is one. Also, please try not to use sudo or root users, unless you know how to handle it, will lead to permission and key management confusion. Besides, please avoid using sudo or root user unless you know clearly how to deal with, otherwise it will cause the authority and key management problems.

appendixB-3Multiple machines use the same ssh public keyMultiple devices use the same ssh public key

When used on different machines, you can convert your ssh private key file id_rsa to "~/.ssh/id_rsa". That's it. In order to use on different devices, you can copy ssh private key file id_rsa to the target device "~/.ssh/id_rsa".

When using the wrong private key, the following prompt will appear. Please replace it with the correct private key.

Below hint will show up if using the wrong private key. Please replace with the correct private key.

Once you've added the correct private key, you're ready to use git clone the code as shown below.

After adding the correct private key, you can use git to clone code, shown as below picture:

Add to ssh. The private key may display the following error message.

Below error may occur when adding ssh private key:

Agent admitted failure to sign using the key

exist on console. Enter the following command to solve the problem.

Input the following command at console can fix it.

ssh-add ~/.ssh/id_rsa

appendixB-4 One machine switches differently ssh public keySwitch different ssh public keys on one device

Can refer to ssh_config document configuration.

You can refer to ssh_config document to configure ssh.

~\$ man ssh_config

Use the following command to configure the current user's ssh configuration.

Use the following commands to configure ssh for current user.

~\$ cp /etc/ssh/ssh_config ~/.ssh/config ~\$ vi .ssh/
config

As shown in the figure, put ssh config file from another directory "~/.ssh1/id_rsa" as the authentication private key. This way it is possible to switch between different keys.

As below picture, identify another directory ssh file "~/.ssh1/id_rsa" as certificate private key. In this way, you can switch different keys.

appendixB-5 Key permission management Key authority management

The server can monitor a certain number of downloads, IP and other information, if an abnormality is found, the corresponding key download permission. The server can real-time monitor for the specific key the download times, IP and other information. If any abnormal case is found, it will prohibit the download authority of the corresponding key.

Please keep the private key file properly. Do not use it with a third party for secondary authorization.

Please keep carefully the private key file. DO NOT re-authorize it to the third party.

appendixB-6 Git Permission application instructions Git authority application instruction

Refer to the above chapters to generate the public key file and send it via email to fae@rock-chips.com, apply for activationSDK Code download permission. Refer to above chapters, generate the public key file, and send email to fae@rock-chips.com applying for SDK code download authority.