Rockchip Android 14.0 Media Streaming SDK Developer Guide

ID: RK-KF-YF-F11

Release Version: V1.1.0

Release Date: 2025-09-01

Security Level: □Top-Secret □Secret □Internal ■Public

DISCLAIMER

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

Trademark Statement

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

Rockchip Electronics Co., Ltd.

No.18 Building, A District, No.89, software Boulevard Fuzhou, Fujian,PRC

Website: www.rock-chips.com

Customer service Tel: +86-4007-700-590

Customer service Fax: +86-591-83951833

Customer service e-Mail: fae@rock-chips.com

Preface

Overview

This document introduces the development of the Rockchip Android 14 Media Streaming SDK, which is primarily suitable for developing streaming media application products such as set-top boxes, projectors, and large-screen devices.

Intended Audience

This document (this guide) is mainly intended for:

Technical support engineers

Software development engineers

Revision History

Version	Author	Date	Change Description
V1.0.0	Mark Huang	2025-03-06	Release RKR1 SDK supporting RK3562/RK3528/RK3518
V1.1.0	Mark Huang	2025-09-01	Release RKR3 SDK supporting RK3676/RK3562/RK3528/RK3518 Support AI Lab Function

If there is any question about the document, please email to: huangjc@rock-chips.com

Contents

1. Rockchip Android 14.0 MS SDK Chipset support

Chipset platform	Support or not	SDK version
RK3562	Support	RKR3
RK3528/RK3518	Support	RKR3
RK3576	Support	RKR3

2. Rockchip Android 14.0 MS SDK code download and compile

2.1 Code download

2.1.1 Download address

```
repo init --repo-url https://gerrit.rock-chips.com:8443/repo-release/tools/repo -
u https://gerrit.rock-chips.com:8443/ms/Android_U/manifests -b master -m
Android14_ms.xml
```

2.1.2 Download server mirroring

```
repo init --repo-url https://gerrit.rock-chips.com:8443/repo-release/tools/repo -
u https://gerrit.rock-chips.com:8443/ms/Android_U/manifests -b master -m
Android14_ms.xml --mirror
```

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

```
git clone https://gerrit.rock-chips.com:8443/repo-release/tools/repo
```

Generally, Rockchip FAE contact will provide the initial compressed package of the corresponding version SDK in order to help customers acquire SDK source code quickly. Take `Rockchip_Android14.0_MS_SDK_RELEASE.tar.gz.*` as an example, you can sync the source code through the following command after getting the initial package:

```
mkdir Rockchip_Android14.0_MS_SDK_RELEASE
cat Rockchip_Android14.0_MS_SDK_RELEASE.tar.gz* | tar -zx -C
Rockchip_Android14.0_MS_SDK_RELEASE
cd Rockchip_Android14.0_MS_SDK_RELEASE
.repo/repo/repo sync -l
.repo/repo/repo sync -c
```

2.2 Set up your own repo code server

2.2.1 Environment

You can install openssh-server for remote login, git for project management, and keychain for public key and private key management tools.

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

2.2.2 Set up gitolite

2.2.2.1 Server-side operation

Take server address: 10.10.10.206 as an example for description.

1. create git account:

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

2. Log in to the server as a 'git' account;
3. Make sure that '~/.ssh/authorized_keys' is empty or non-existent;
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. Create bin directory in git user directory;

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

7. Please execute following commands to install gitolite, and the installation method is different for different versions. Please refer to the documentation in source code:

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

8. Set the administrator.

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

2.2.2.2 Client-side operation

1. Clone gitolite management warehouse of the server;

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

2. Add user's public key to the gitolite directory;

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

3. Add an administrator user.

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

2.2.3 Set up repo mirror

2.2.3.1 Server-side operation

1. Log in to the server as a 'git' account;
2. Download the repo tool in the root directory;

```
git clone https://gerrit.rock-chips.com:8443/repo-release/tools/repo
```

3. Create a new RK_Android14_mirror directory;

```
mkdir RK_Android14_mirror
```

4. Enter the RK_Android14_mirror directory;

```
cd RK_Android14_mirror
```

5. Download RK Android14 MS SDK mirror;

```
repo init --repo-url https://gerrit.rock-chips.com:8443/repo-release/tools/repo -
u https://gerrit.rock-chips.com:8443/ms/Android_U/manifests -b master -m
Android14_ms.xml --mirror
```

6. Create warehouse group permissions.

```
.repo/repo/repo list -n > android_u.conf
sed -i 's/^/@android_u = RK_Android14_mirror\/&/g' android_u.conf
```

2.2.3.2 Local client-side operation

1. Copy android_r.conf on the server-side to ·gitolite-admin/conf/· on the client-side;
2. Add group permissions.

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

```
vi conf/gitolite.conf
include "android_u.conf"
```

3. Create your own new manifests warehouse.

```
vi conf/android_u.conf
@android_u = Android_T/manifests_xxx
```

2.2.4 Other client-side operation

1. Download manifests_xxx warehouse on the client-side;

Download manifests_xxx.git warehouse on other client-side

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

2. Download original manifests warehouse on the client-side;

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

3. Submit manifest.xml file to manifest_xxx warehouse created newly;

Copy the files below original manifests to the manifests_xxx

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

check copy files

```
git status

 Android14_ms.xml
 default.xml
 include/rk3528_repository.xml
 include/rk_checkout_from_aosp.xml
 include/rk_modules_repository.xml
 remote.xml
 remove_u.xml
 remove_unused.xml
```

Local commits

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

Push to the remote branch

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

4. Create your own code download link.

Download the repo tool in the root directory

```
git clone https://gerrit.rock-chips.com:8443/repo-release/tools/repo
```

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

```
mkdir Android14
cd Android14
~/repo/repo init -u ssh://git@10.10.10.206/Android_U/manifests_xxx.git -m
Android14_ms.xml
```

Thereinto:
`//10.10.10.206` which is your server address

You can complete your own repo server set-up with above steps, and you can share your code server links with colleagues to work together.

2.3 Code management

After setting up the code server with above steps, most of the code warehouses use the default branches of RK. If some warehouses need to modify their own codes, you can refer to the following steps for operation.

2.3.1 Switch your own code branches

1. Enter the code warehouse that needs to be modified, and we take the kernel directory as an example to illustrate;

```
cd kernel-6.1
```

2. Switch a local branch;

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

3. Push xxx_branch to remote server;

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

Thereinto, `rk29` is remote, which can be completed automatically by tab key directly

4. Enter .repo/manifests directory and modify the branch which is appointed by manifest;

Enter .repo/manifests directory, and you can find the manifest location corresponding to the kernel warehouse by grep kernel

```
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-14.0-ms-rkr1" />
   <project path="hardware/rockchip/libhwjpeg" name="rk/hardware/rk29/libhwjpeg" remote="rk" revision="refs/tags/android-14.0-ms-rkr1"/>
   <project path="u-boot" name="rk/u-boot" remote="rk" revision="refs/tags/android-14.0-ms-rkr1"/>

-  <project path="kernel" name="rk/kernel" remote="rk29" revision="refs/tags/android-14.0-ms-rkr1"/>

+  <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-14.0-ms-rkr1"/>
   <project path="bootable/recovery/rkutility" name="platform/bootable/recovery/rk_utility" remote="rk" revision="refs/tags/android-14.0-ms-rkr1"/>
```

5. Submit the modified manifest to the remote branch.

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

After submitting manifests warehouse, other colleagues can synchronize the kernel codes of your own branches.

2.3.2 Code modification submittal

After switching branches according to the steps above, you can commit your modification on your branches and push them directly to the xxx_branch.

2.3.3 Synchronize RK codes

1. It's required to synchronize RK codes on the server-side;

```
cd RK_Android14_mirror
```

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

2. The manifests that RK modifies are combined by client-side;

- Download the original manifests warehouse of RK;

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

The manifests (RK original) and the manifests_xxx (yourselves) are compared with the contrast tools to combine the different parts that RK modifies to your own warehouses (mainly modify the tag, adding and removing the warehouse, etc)

- After comparing and confirming, the modification will be pushed to the manifests_xxx.

You can also confirm which warehouses are modified in this step, and in the next step you will combine the modified warehouses.

3. The directories switched branches by yourselves need to push the merge that RK modifies to your own branches manually.

Take kernel as an example:

- Check the pointed remote branches at present

```
wlq@wlq:~/home1/test2/kernel-6.1$ git branch -av
* android-11.0-mid-rkr7   0bde59fad73a ARM: configs: rockchip_defconfig enable
ION_CMA_HEAP
  xxx_branch              0bde59fad73a ARM: configs: rockchip_defconfig enable
ION_CMA_HEAP
  remotes/m/master        -> rk29/xxx_branch
  remotes/rk29/xxx_branch 0bde59fad73a ARM: configs: rockchip_defconfig enable
ION_CMA_HEAP
```

You can find that the branch pointed at present is: `remotes/m/master        -> rk29/xxx_branch`

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

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

- Check latest TAG published currently by RK

```
wlq@wlq:~/home1/test2/kernel$ git tag | grep rkr
android-10.0-mid-rkr1
android-10.0-mid-rkr10
android-10.0-mid-rkr11
android-10.0-mid-rkr13
android-10.0-mid-rkr2
android-10.0-mid-rkr3
android-10.0-mid-rkr4
android-10.0-mid-rkr5
android-10.0-mid-rkr6
android-10.0-mid-rkr7
android-10.0-mid-rkr8
android-10.0-mid-rkr9
android-11.0-ebook-rkr1
android-11.0-ebook-rkr2
android-11.0-ebook-rkr3
android-11.0-ebook-rkr4
android-11.0-ebook-rkr5
```

```
android-11.0-ebook-rkr6
android-11.0-mid-rkr1
android-11.0-mid-rkr2
android-11.0-mid-rkr3
android-11.0-mid-rkr4
android-11.0-mid-rkr4.1
android-11.0-mid-rkr5
android-11.0-mid-rkr6
android-11.0-mid-rkr7
android-11.0-mid-rkr7-prev
android-11.0-mid-rkr8
android-14.0-ms-rkr1
```

You can find the latest tag of Android14 currently is `android-14.0-ms-rkr1`

- combine `android-14.0-ms-rkr1` to the local branch

```
git merge android-14.0-ms-rkr1
```

Check if there is a conflict. If there is a conflict, resolve the conflict firstly. You can execute the next step when there is no conflict.

- push the codes which have been combined to the remote branch

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

- The other directories swiched can be combined and submitted in this way

2.4 kernel Code path description

Android14 supports version 6.1 of the kernel, kernel source code in the project kernel-6.1 directory,

2.5 Code compiling

2.5.1 Lunch item Description

lunch item	chipest adapted	other description
rk3518_box_32-user	RK3518	Suitable for Android 14 Box-type products, compatible with RK3518 development board hardware. Low-memory optimization is enabled by default, supporting hardware with 1GB or more of memory. Compiled as user version for production use. Android system supports 32-bit only.
rk3518_box_32-userdebug	RK3518	Suitable for Android 14 Box-type products, compatible with RK3518 development board hardware. Low-memory optimization is enabled by default, supporting hardware with 1GB or more of memory. Compiled as userdebug version for development and debugging. Android system supports 32-bit only.
rk3528_box_32-user	RK3528	Suitable for Android 14 Box-type products, compatible with RK3528 development board hardware. Low-memory optimization is enabled by default, recommended for hardware with 1.5GB or more of memory. Compiled as user version for production use. Android system supports 32-bit only.
rk3528_box_32-userdebug	RK3528	Suitable for Android 14 Box-type products, compatible with RK3528 development board hardware. Low-memory optimization is enabled by default, recommended for hardware with 1.5GB or more of memory. Compiled as userdebug version for development and debugging. Android system supports 32-bit only.
rk3528_box-user	RK3528	Suitable for Android 14 Box-type products, compatible with RK3528 development board hardware. Recommended for hardware with 2GB or more of memory. Compiled as user version for production use. Android system supports 64-bit only.
rk3528_box-userdebug	RK3528	Suitable for Android 14 Box-type products, compatible with RK3528 development board hardware. Recommended for hardware with 2GB or more of memory. Compiled as userdebug version for development and debugging. Android system supports 64-bit only.
rk3562_projector_32-user	RK3562	Suitable for Android 14 projector or Box-type products, compatible with RK3562 development board hardware. Low-memory optimization is enabled by default. Compiled as user version for production use. Android system supports 32-bit only.
rk3562_projector_32-userdebug	RK3562	Suitable for Android 14 projector or Box-type products, compatible with RK3562 development board hardware. Low-memory optimization is enabled by default. Compiled as userdebug version for development and debugging. Android system supports 32-bit only.
rk3562_projector-user	RK3562	Suitable for Android 14 projector or Box-type products, compatible with RK3562 development board hardware. Compiled as userdebug version for production use. Android system supports 64-bit only.

lunch item	chipest adapted	other description
rk3562_projector-userdebug	RK3562	Suitable for Android 14 projector or Box-type products, compatible with RK3562 development board hardware. Compiled as user version for development and debugging. Android system supports 64-bit only.
rk3576_projector-user	RK3576	Suitable for Android 14 projector or Box-type products, compatible with RK3576 development board hardware. Supports AI applications. Compiled as user version for production use. Android system supports 64-bit only.
rk3576_projector-userdebug	RK3576	Suitable for Android 14 projector or Box-type products, compatible with RK3576 development board hardware. Supports AI applications. Compiled as userdebug version for development and debugging. Android system supports 64-bit only.

2.5.2 One key compiling command

```
./build.sh -UCKAup
( 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 = huild kernel dts name
       -V = build version
       -J = build jobs
   ------------you can use according to the requirement, no need to record
uboot/kernel compiling commands------------------
 )

============================================================
Please remember to set the environment variable before using the one key
compiling command, and select the platform to be compiled, for example:
source build/envsetup.sh
lunch rk3528_box-userdebug
============================================================
```

2.5.3 Compiling command summary

Soc	type	The reference model	Android	one key compiling	kernel compiling	uboot compiling
RK3518	Development board	rk3518-evb1-ddr4-v10	build/envsetup.sh;lunch rk3518_box_32-userdebug	./build.sh -AUCKu	./build.sh -K	./build.sh -U
RK3528	Development board	rk3528-evb1-ddr4-v10	build/envsetup.sh;lunch rk3528_box-userdebug	./build.sh -AUCKu	./build.sh -K	./build.sh -U
RK3562	Development board	rk3562-evb2-ddr4-v10	build/envsetup.sh;lunch rk3562_projector-userdebug	./build.sh -AUCKu	./build.sh -K	./build.sh -U
RK3576	Development board	rk3576-evb1-v10	build/envsetup.sh;lunch rk3576_projector-userdebug	./build.sh -AUCKu	./build.sh -K	./build.sh -U

2.5.4 GKI

Select to disable GKI by default in the SDK of RK3562 platform, and unuse AB function by default. If you need to close GKI, you can modify as followed:

```
wlq@sys2206:~/b0_A14_bringup/device/rockchip/rk3562$ git diff
diff --git a/rk3562_u/BoardConfig.mk b/rk3562_u/BoardConfig.mk
index dc9cc50..a6657dd 100644
--- a/rk3562_u/BoardConfig.mk
+++ b/rk3562_u/BoardConfig.mk
@@ -16,7 +16,7 @@
 BUILD_WITH_GO_OPT := true
 PRODUCT_KERNEL_DTS := rk3562-rk817-tablet-v10
 CAMERA_SUPPORT_AUTOFOCUS := true
-BOARD_BUILD_GKI := true
+BOARD_BUILD_GKI := false
 include device/rockchip/rk3562/BoardConfig.mk

 DEVICE_IS_64BIT_ONLY := true
```

After BOARD_BUILD_GKI := false, the AB function will close automatically.
Please refer to the document about GKI details:
RKDocs/android/《Rockchip_Developer_Guide_Android14_GKI_CN》

2.5.5 Other compiling instruction

2.5.5.1 Android14.0 cannot directly flash kernel.img and resource.img

The following compilation is only applicable for non-GKI. For GKI, please refer to the document RKDocs/android/《Rockchip_Developer_Guide_Android14_GKI_CN》.
The kernel.img and resource.img of Android 14.0 are included in the boot.img. To compile the kernel, use the build.sh -AK command. After compilation, burn the boot.img under rockdev. This process involves recompiling Android, so compilation time will be relatively long. It is recommended to use the following method to compile the kernel separately.

2.5.5.2 Compile Kernel Separately to Generate boot.img

Principle of compilation: Replace the compiled kernel.img and resource.img in the kernel directory with the old boot.img.
Taking the RK3528prototype as an example, replace the corresponding boot.img and DTS during compilation. Here, BOOT_IMG=../rockdev/Image-rk3528_box/boot.img specifies the path of the old boot.img. The command is as follows:

Export clang to the environment

```
cd kernel-6.1
export PATH=../prebuilts/clang/host/linux-x86/clang-r487747c/bin:$PATH
```

```
#rk3528 64BIT:
alias msk='make CROSS_COMPILE=aarch64-linux-gnu- LLVM=1 LLVM_IAS=1'
msk ARCH=arm64 rockchip_defconfig android-14.config && msk ARCH=arm64
BOOT_IMG=../rockdev/Image-rk3528_box/boot.img rk3528-evb1-ddr4-v10.img -j32

#rk3528 32BIT:
alias msk='make CROSS_COMPILE=arm-linux-gnu- LLVM=1 LLVM_IAS=1'
msk ARCH=arm rockchip_defconfig android-14.config && msk ARCH=arm
BOOT_IMG=../rockdev/Image-rk3528_box_32/boot.img rk3528-evb1-ddr4-v10.img -j32
```

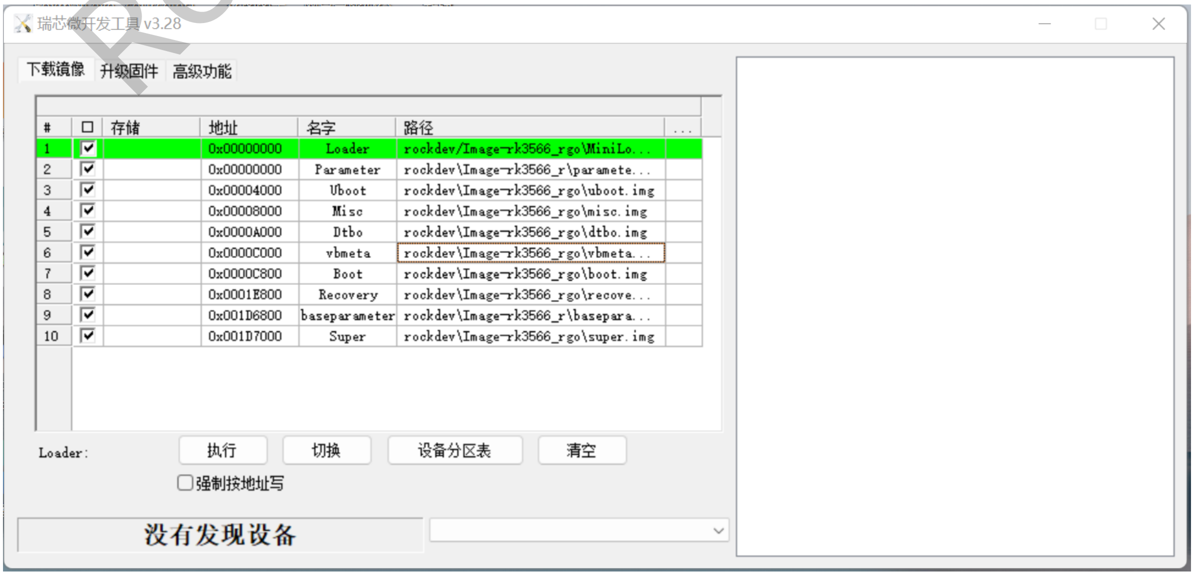
You can flash boot.img under the catalogue of kernel-6.1 directly to boot position of machine after compiling, and please load the partition table (parameter.txt) when flashing, for fear of flashing to the wrong place.

2.6 Image flashing

2.6.1 Image flashing tool

Android14 requires to update the USB driver DriverAssitant to V5.1.1 version. You can refer to the tool instruction chapter to do the upgrade.
Windows flashing tool:

```
RKTools/windows/AndroidTool/RKDevTool_v3.30_for_window.zip
```



#	□	存储	地址	名字	路径	...
1	☑		0x00000000	Loader	rockdev/Image-rk3566_rgo\MiniLo...	
2	☑		0x00000000	Parameter	rockdev\Image-rk3566_r\paramete...	
3	☑		0x00004000	Uboot	rockdev\Image-rk3566_rgo\uboot.img	
4	☑		0x00008000	Misc	rockdev\Image-rk3566_rgo\misc.img	
5	☑		0x0000A000	Dtbo	rockdev\Image-rk3566_rgo\dtbo.img	
6	☑		0x0000C000	vbmeta	rockdev\Image-rk3566_rgo\vbmeta...	
7	☑		0x0000C800	Boot	rockdev\Image-rk3566_rgo\boot.img	
8	☑		0x0001E800	Recovery	rockdev\Image-rk3566_rgo\recove...	
9	☑		0x001D6800	baseparameter	rockdev\Image-rk3566_r\basepara...	
10	☑		0x001D7000	Super	rockdev\Image-rk3566_rgo\super.img	

```
RKTools/linux/Linux_Upgrade_Tool/Linux_Upgrade_Tool_v2.30
```

There are more details in the tool instruction chapter.

2.6.2 Image instruction

After complete compiling, it will generate the following files:

```
rockdev/Image-rk3528_box/
├── baseparameter.img
├── 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
```

Just flash the following files:

```
rockdev/Image-rk3528_box/
├── boot.img
├── dtbo.img
├── MiniLoaderAll.bin
├── misc.img
├── parameter.txt
├── recovery.img
├── super.img
├── uboot.img
└── vbmeta.img
```

or you can directly flash `update.img`

2.6.3 Image instruction

Image	Instruction
boot.img	including ramdis, kernel, dtb
boot-debug.img	the difference between boot.img and boot-debug.img is that user image can flash this boot.img to do root operation
dtbo.img	Device Tree Overlays refer to dtbo chapter instruction later
config.cfg	the configuration file of the flash tool, you can directly load the options required to be flashed for the flash tool
MiniLoaderAll.bin	including first level loader
misc.img	including recovery-wipe boot symbol information, after flashing it will enter recovery
parameter.txt	including partition information
pcba_small_misc.img	including pcba boot symbol information, after flashing it will enter the simple pcba mode
pcba_whole_misc.img	including pcba boot symbol information, after flashing it will enter the complete pcba mode
recovery.img	including recovery-ramdis, kernel, dtb
super.img	including the contents of odm, vendor, system partitions
trust.img	including BL31, BL32 which are not generated for RK3566/RK3568, no need to flash
uboot.img	including uboot image
vbmeta.img	including avb verification information, used for AVB verification
update.img	including the above img files to be flashed, can be used for the tool to directly flash the whole image package

2.7 Generic Kernel Image (GKI)

All the Android 14 products certificating GTVS and EDLA are forced to use GKI for kernel, please refer to the document for GKI configuration and compiling:

RKDocs/android/Rockchip_Developer_Guide_Android14_GKI_CN.pdf

2.8 Use fastboot to flash dynamic partition

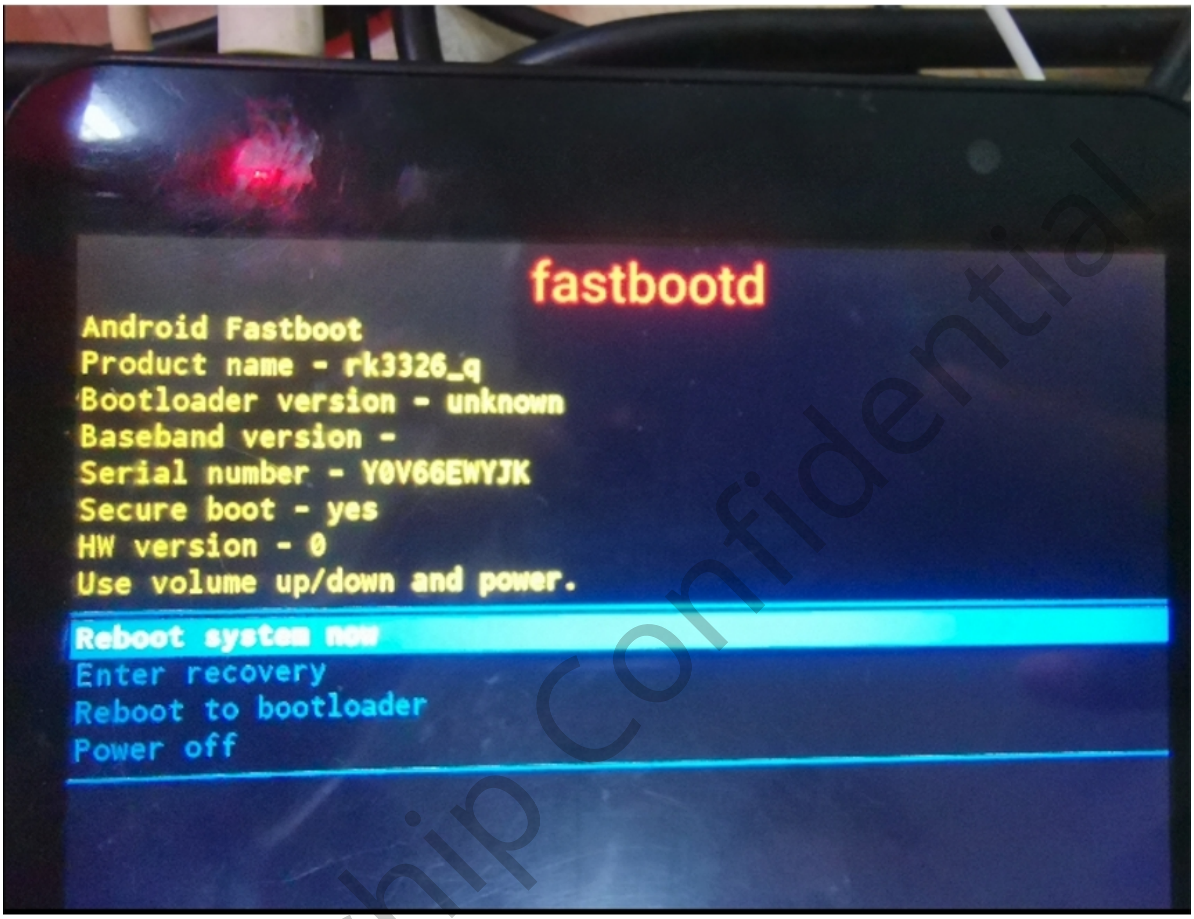
The new device with R supports dynamic partition, and already removessystem/vendor/odm/product/system_ext partitions. Please flash super.img. Use `fastbootd` can flash system/vendor/odm alone. The version of adb and fastboot should be the latest. SDK provides the compiled tool package:

```
RKTools/linux/Linux_adb_fastboot (Linux_x86 version)
RKTools/windows/adb_fastboot (Windows_x86 version)
```

- Use the command to flash dynamic partition:

```
adb reboot fastboot
fastboot flash vendor vendor.img
fastboot flash system system.img
fastboot flash odm odm.img
```

Note: After entering fastbootd mode, relative information of the device will be displayed on the screen, as shown below:



```
fastbootd
Android Fastboot
Product name - rk3326_q
Bootloader version - unknown
Baseband version -
Serial number - Y0V66EWYJK
Secure boot - yes
HW version - 0
Use volume up/down and power.
Reboot system now
Enter recovery
Reboot to bootloader
Power off
```

Note: please enter bootloader when using fastboot in the non-dynamic partition:

```
adb reboot bootloader
```

The way to flash GSI:

- After the device is unlocked, enter fastbootd, only need to flash system.img of GSI and misc.img of the image, and after flashing it will enter recovery to do factory reset. Attach the complete flashing process as below:

1. Reboot to bootloader, lock->unlock the device:

```
adb reboot bootloader
fastboot oem at-unlock-vboot ## for the customers already flashing avb public
key, please refer to the corresponding document to unlock.
```

2. Reset to factory setting, reboot to fastbootd:

```
fastboot flash misc misc.img
fastboot reboot fastboot  ## now it will enter fastbootd
```

3. Start to flash GSI

```
fastboot delete-logical-partition product  ## (optional) for the device with
small partition space, you can execute this command to delete product partition
first and then flash GSI
fastboot flash system system.img
fastboot reboot  ## after flashing successfully, reboot the device
```

- Use DSU(Dynamic System Updates) to flash GSI, and current Rockchip platform already supports DSU by default. As this function requires large memory, it is not recommended to use on the device with 1G DDR or less. For the instruction and usage of DSU, please refer to Android official website:

https://source.android.com/devices/tech/ota/dynamic-system-updates

- Note 1: when testing VTS, need to flash the compiled boot-debug.img to boot partition.
- Note 2: when testing CTS-ON-GSI, no need to flash boot-debug.img.
- Note 3: please use GSI image ended with -signed released by Google for testing.

3. DTBO function

Android 10.0 and above versions support Device Tree Overlays function, which requires to flash dtbo.img during development, and is compatible with multiple products.
The modification method:

1. Find (or specify) the template file:

```
get_build_var PRODUCT_DTBO_TEMPLATE
```

For example:

```
PRODUCT_DTBO_TEMPLATE := $(LOCAL_PATH)/dt-
overlay.in(device/rockchip/rk388/rk3588_t/dt-overlay.in)
```

2. Add or modify the required node:

For example:

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

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

&firmware_android {
    vbmeta {
        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: There must be alias in the dts when using dtbo, otherwise it cannot overlay successfully

4. Modify fstab file

1. Fine (or specify) the template file:

```
get_build_var PRODUCT_FSTAB_TEMPLATE
```

For example:

```
PRODUCT_FSTAB_TEMPLATE := device/rockchip/common/scripts/fstab_tools/fstab.in
```

2. Modify: add partition mounting, modify swap_zram parameter, modify data partition format and so on

5. Modify parameter.txt

Android 14 adds the tool that can generate parameter.txt, and support to compile parameter.txt according to the configuration parameters. If there is no configuration template file, it will find and add the modified parameter.txt file.

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. Partition size configuration(example as below):

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

3. Not to use parameter generation tool:

Just add a parameter.txt file to your device directory:
For example:

```
device/rockchip/rk3326/rk3326_u/parameter.txt
```

4. Only use the tool to generate parameter.txt(example as below):

```
parameter_tools --input
device/rockchip/common/scripts/parameter_tools/parameter.in --firmware-version
14.0 --machine-model rk3326 --manufacturer rockchip --machine rk3326_u --
partition-list
uboot_a:4096K,trust_a:4M,misc:4M,dtbo_a:4M,vbmeta_a:4M,boot_a:33554432,backup:300
M,security:4M,cache:300M,metadata:4096,frp:512K,super:2G --output
parameter_new.txt
```

Note: If need to do the big version upgrade through OTA, please directly use the previous version's parameter.txt

5. Add a new partition

Take baseparameter adding for example:

- Give a definition in the BoardConfig.mk of the product:BOARD_WITH_SPECIAL_PARTITIONS

like: BOARD_WITH_SPECIAL_PARTITIONS := baseparameter:1M,logo:16M

```
device/rockchip/rk356x/rk3566_u/BoardConfig.mk
+++ b/BoardConfig.mk
@@ -494,4 +494,11 @@ ifeq ($(strip $(BOARD_TWRP_ENABLE)), true)
+BOARD_WITH_SPECIAL_PARTITIONS := baseparameter:1M
```

- Add BOARD_WITH_SPECIAL_PARTITIONS to RebuildParameter.mk

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

6. Android common configuration

6.1 Create product lunch

Take RK3562 platform as example to create a new rk3562_new_u product. The steps are as below:

1. Modify device/rockchip/rk3562/AndroidProducts.mk to add rk3562_new_u lunch

```
--- a/AndroidProducts.mk
+++ b/AndroidProducts.mk
@@ -17,10 +17,14 @@
 PRODUCT_MAKEFILES := \
         $(LOCAL_DIR)/rk3562_u/rk3562_u.mk \
+        $(LOCAL_DIR)/rk3562_new_u/rk3562_new_u.mk \

 COMMON_LUNCH_CHOICES := \
     rk3562_u-userdebug \
     rk3562_u-user \
+    rk3562_new_u-userdebug \
+    rk3562_new_u-user \
```

2. Create rk3562_new_u directory under device/rockchip/rk3562

Create referring to the existing rk3562_u product directory in device/rockchip/rk3562. You can directly copy rk3562_u to rk3562_new_u, and then replace all the `rk3562_u` under rk3562_new_u directory with `rk3562_new_u`

7. Kernel dts instruction

7.1 Create new product dts

You can select the corresponding dts according to the configuration in the following table as reference to create new product dts.

Soc	PMIC	Type	Model	DTS
RK3588	Dual PMIC: RK806 * 2	development board	RK3588 EVB1	rk3588-evb1-lp4-v10
RK3588	Single PMIC: RK806+RK860	development board	RK3588 Hardware reference design corresponding software configuration	rk3588-evb4-lp7-v10
RK3588S	Dual PMIC: RK806 * 2	development board	RK3588S EVB1	rk3588s-evb1-lp4x-v10
RK3588S	Single PMIC: RK806 +RK860	tablet	Refer to the hardware design diagram for the RK3588S tablet	rk3588s-tablet-v11
RK3588S	Dual PMIC: RK806*2	tablet	devices	rk3588s-tablet-v10
RK3588	Single PMIC: RK860	box evb	RK3588_NVR_DEMO1_LP4X	rk3588-nvr-demo-v10-android
RK3566	RK817	tablet	devices	rk3566-rk817-tablet
RK3566	RK809	development board	RK3566 EVB2	rk3566-evb2-lp4x-v10
RK3566	dcdc	development board	RK3566 BOX DEMO	rk3566-box-demo-v10
RK3568	RK809	development board	RK3568 EVB1	rk3568-evb1-ddr4-v10
RK3562	RK817	development board	RK3562 EVB1	rk3562-evb1-lp4x-v10
RK3562	RK809	development board	RK3562 EVB2	rk3562-evb2-ddr4-v10
RK3562	RK817	tablet	devices	rk3562-rk817-tablet-v10
RK3528	dcdc	BOX	RK3528 EVB1	rk3528-evb1-ddr4-v10
RK3518	dcdc	BOX	RK3518 EVB1	rk3518-evb1-ddr4-v10

8. Patch release

Redmine system will release some important patches from time to time, the link is as follows:

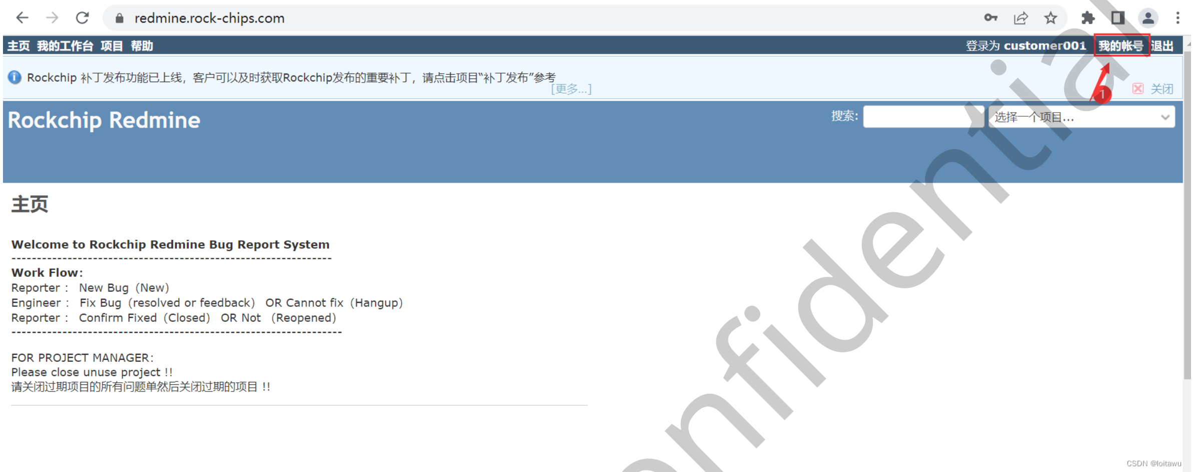
```
https://redmine.rock-chips.com/projects/rockchip_patch/issues
```

You can subscribe to receive real-time email notification of patch release in the following ways:

Step 1: Log in to redmine system

Log in to redmine system with redmine account registered with Rockchip.

Step 2: access my account



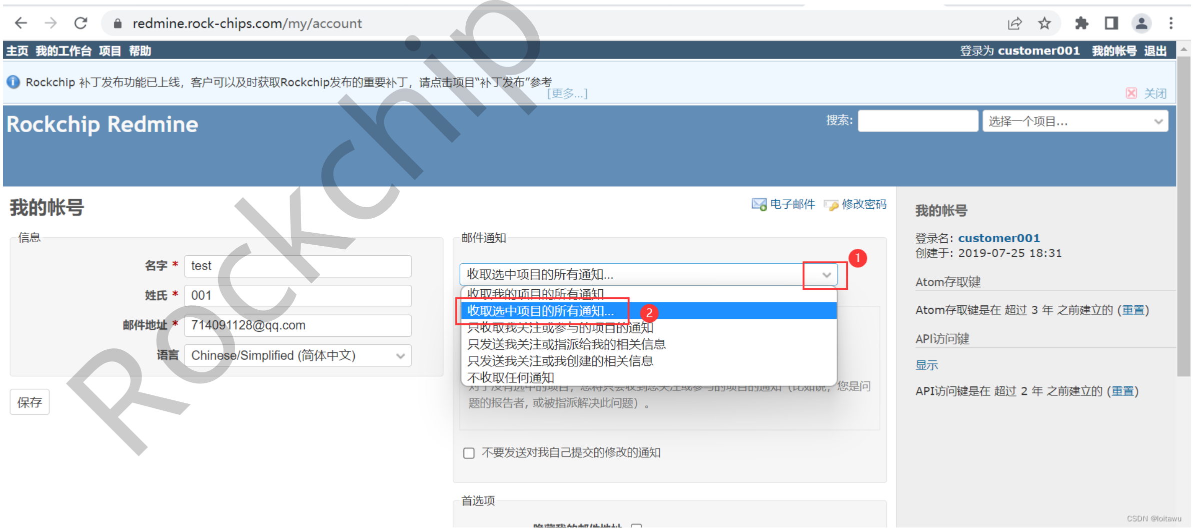
Rockchip 补丁发布功能已上线，客户可以及时获取Rockchip发布的重要补丁，请点击项目“补丁发布”参考 [更多...]

主页

```
Welcome to Rockchip Redmine Bug Report System
------------------------------------------------------------
Work Flow:
Reporter :  New Bug (New)
Engineer :  Fix Bug (resolved or feedback)  OR Cannot fix (Hangup)
Reporter :  Confirm Fixed (Closed)  OR Not  (Reopened)
--------------------------------------------------------------

FOR PROJECT MANAGER:
Please close unuse project !!
请关闭过期项目的所有问题单然后关闭过期的项目 !!
```

Step 3: Select an email notification type

In the following figure, select the mail notification drop-down to receive all notifications for the selected items



redmine.rock-chips.com/my/account
主页 我的工作台 项目 帮助
登录为 customer001 我的帐号 退出
关闭
Rockchip Redmine
搜索:
选择一个项目...
电子邮件
修改密码
1
2
保存
CSDN @loitawu

Rockchip 补丁发布功能已上线，客户可以及时获取Rockchip发布的重要补丁，请点击项目“补丁发布”参考 [更多...]

我的帐号

信息

名字 * test

姓氏 * 001

邮件地址 * 714091128@qq.com

语言 Chinese/Simplified (简体中文)

邮件通知

收取选中项目的所有通知...

- 收取我的项目的所有通知
- 收取选中项目的所有通知...
- 只收取我关注或参与的项目的通知
- 只发送我关注或指派给我的相关信息
- 只发送我关注或我创建的相关信息
- 不收取任何通知

对于没有选中的项目，您将只会收到您关注或参与的项目的通知（比如说，您是问题的报告者，或被指派解决此问题）。

□ 不要发送对我自己提交的修改的通知

首选项

隐藏我的邮件地址 □

我的帐号

登录名: customer001
创建于: 2019-07-25 18:31

Atom存取键

Atom存取键是在 超过 3 年 之前建立的 (重置)

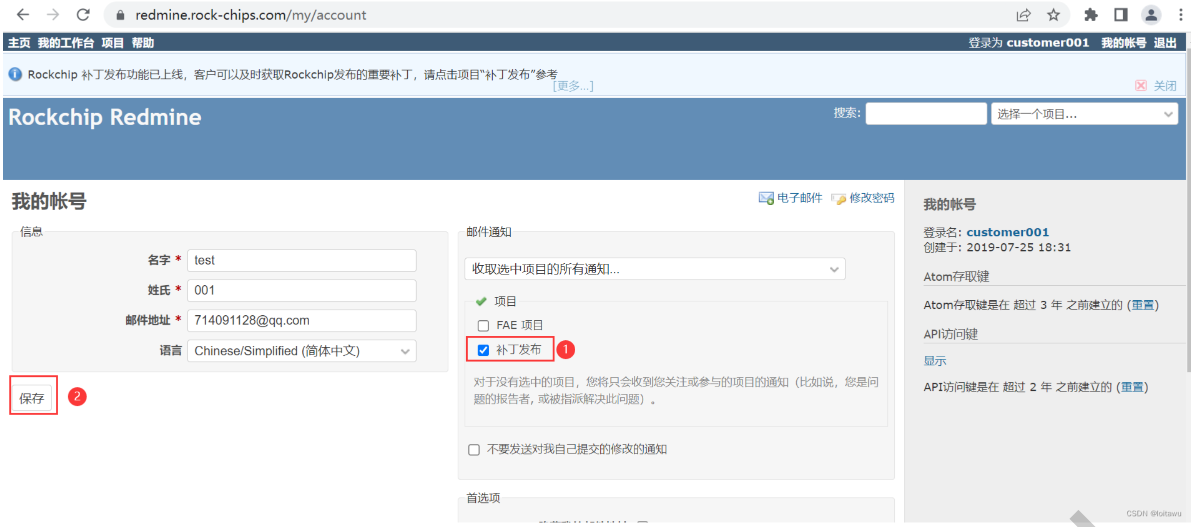
API访问键

显示

API访问键是在 超过 2 年 之前建立的 (重置)

Step 4: Select the project

Check the patch release item as shown below and click Save



redmine.rock-chips.com/my/account

主页 我的工作台 项目 帮助 登录为 customer001 我的帐号 退出

Rockchip 补丁发布功能已上线，客户可以及时获取Rockchip发布的重要补丁，请点击项目"补丁发布"参考 [更多...] 关闭

Rockchip Redmine 搜索: 选择一个项目...

我的帐号 电子邮件 修改密码

信息

名字 * test

姓氏 * 001

邮件地址 * 714091128@qq.com

语言 Chinese/Simplified (简体中文)

保存 2

邮件通知

收取选中项目的所有通知...

项目

□ FAE 项目

☑ 补丁发布 1

对于没有选中的项目，您将只会收到您关注或参与的项目的通知（比如说，您是问题的报告者，或被指派解决此问题）。

□ 不要发送对我自己提交的修改的通知

首选项

我的帐号

登录名: customer001
创建于: 2019-07-25 18:31

Atom存取键

Atom存取键是在 超过 3 年 之前建立的 (重置)

API访问键

显示

API访问键是在 超过 2 年 之前建立的 (重置)

CSDN @loitawu

Done:

After a successful subscription, Rockchip can receive email notifications when patches are released through the email address registered on redmine.

9. Document instruction

9.1 Peripheral support list

DDR/EMMC/NAND FLASH/WIFI/3G/CAMERA support lists keep updating in redmine, through the following link:

```
https://redmine.rockchip.com.cn/projects/fae/documents
```

9.2 Camera IQ Tool Document

```
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
```

9.3 rknn-toolkit2 developing SDK and document

```
hardware/rockchip/rknn-toolkit2/doc/
```

9.4 RKDocs Instruction

```
RKDocs/
├── android
│   ├── Android11 异显开发说明.zip
│   ├── audio
│   │   ├── Rockchip_Developer_Guide_Android_3A_Module_CN.pdf
│   │   ├── Rockchip_Developer_Guide_Android_EQ_DRC_CN.pdf
│   │   ├── Rockchip_Developer_Guide_Android_Multi_Audio_CN.pdf
│   │   └── Rockchip_Developer_Guide_Automotive_Audio_CN.pdf
│   ├── bt
│   │   └── Rockchip_Introduction_Android9.0_BT_Configuration_CN.pdf
│   ├── patches
│   ├── Rockchip_Android_Remote_key_Provisioning_Guide.pdf
│   ├── Rockchip_Developer_Guide_AB_System_OTA_from_Android12_to_Android14_CN.pdf
│   ├── Rockchip_Developer_Guide_AB_System_OTA_from_Android13_to_Android14_CN.pdf
│   ├── Rockchip_Developer_Guide_Android11_Optimization_CN.pdf
│   ├── Rockchip_Developer_Guide_Android14_GKI_CN.pdf
│   ├── Rockchip_Developer_Guide_Android14_GKI_EN.pdf
│   ├── Rockchip_Developer_Guide_Android15_SDK_CN.pdf
│   ├── Rockchip_Developer_Guide_Android15_SDK_EN.pdf
│   ├── Rockchip_Developer_Guide_Android_AB_System_Upgrading_CN.pdf
│   ├── Rockchip_Developer_Guide_Android_Automotive_CN.pdf
│   ├── Rockchip_Developer_Guide_Android_Recovery_CN.pdf
│   ├── Rockchip_Developer_Guide_Android_SELinux(Sepolicy)_CN.pdf
│   ├── Rockchip_Developer_Guide_OTA_from_Android12_to_Android14_CN.pdf
│   ├── Rockchip_Developer_Guide_OTA_from_Android13_to_Android14_CN.pdf
│   ├── Rockchip_Developer_Guide_PCBA_Test_Tool_V1.3_CN&EN.pdf
│   ├── Rockchip_Firmware_Upgrade_Failed_Analyze_Method_CN.pdf
│   ├── Rockchip_Introduction_Android_Application_Preinstallation_CN&EN.pdf
│   ├── Rockchip_Introduction_Android_Boot_Video_CN.pdf
│   ├── Rockchip_Introduction_Android_BOX_Display_Framework_Configuration_CN.pdf
│   ├── Rockchip_Introduction_Android_Factory_Reset_Protection_CN&EN.pdf
│   ├── Rockchip_Introduction_Android_Log_System.pdf
│   ├── Rockchip_Introduction_Android_Performance_Mode_CN&EN.pdf
│   ├──
Rockchip_Introduction_Android_Power_On_Off_Animation_and_Tone_Customization_CN&EN
.pdf
│   ├── Rockchip_Introduction_Android_Samba_CN.pdf
│   ├── Rockchip_Introduction_Android_Widevine_Project_Start_Preparation_CN.pdf
│   ├── Rockchip_Introduction_Box_Media_Application_CN&EN.pdf
│   ├── Rockchip-Parameter-File-Format-Version1.4-CN.pdf
│   ├── Rockchip_User_Guide_Android_Device_ID_Attestation_Flash_CN.pdf
│   ├── Rockchip_User_Guide_Android_GMS_Configuration_CN.pdf
│   ├── Rockchip_User_Guide_Android_GMS_Configuration_EN.pdf
│   ├── Rockchip_User_Guide_Box_FactoryTestTool_V3.0_CN.pdf
│   ├── Rockchip_User_Guide_Dr.G_CN&EN.pdf
│   ├── Rockchip_User_Guide_Magisk_Installation_EN.pdf
│   ├── video
│   │   ├── Rockchip_Android_Introduction_Libhwjpeg_Interface_CN.pdf
│   │   ├── Rockchip_Android_Introduction_MediaCodec_Vendor_Parameters_CN.pdf
│   │   ├── Rockchip_Android_Multimedia_FAQ_CN.pdf
│   │   ├── Rockchip_Android_Multimedia_FAQ_EN.md
│   │   └── Rockchip_Android_Widevine_L1_Guide.pdf
│   └── wifi
│       ├── Rockchip_Introduction_REALTEK_WIFI_Driver_Porting_CN&EN.pdf
│       └── Rockchip_Introduction_WIFI_Configuration_CN&EN.pdf
└── common
    ├── Audio
    │   ├──
Rockchip_Developer_Guide_Audio_Call_3A_Algorithm_Integration_and_Parameter_Debugg
ing_CN.pdf
    │   ├── Rockchip_Developer_Guide_Audio_CN.pdf
    │   ├── Rockchip_Developer_Guide_RK817_RK809_Codec_CN.pdf
    │   └── Rockchip_User_Guide_Android14_Audio_Function_CN.pdf
    ├── camera
    │   ├── common
    │   │   └── Camera_External_FAQ_v1.0 .pdf
    │   ├── HAL1
    │   │   ├── README_CN.txt
    │   │   ├── README_EN.txt
    │   │   ├── RK312x_Camera_User_Manual_v1.4(3288&3368).pdf
    │   │   ├── RK_ISP10_Camera_User_Manual_v2.3.pdf
    │   │   ├── RKISPV1_Camera_Module_AVL_v1.7.pdf
    │   │   ├── Rockchip_Camera_AVL_v2.0_Package_20180515.7z
    │   │   ├──
Rockchip_Introduction_RKISPV1_Camera_Driver_Debugging_Method_CN.pdf
    │   │   ├── Rockchip_Introduction_RKISPV1_Camera_FAQ_CN.pdf
    │   │   ├── Rockchip SOFIA 3G-
R_PMB8018(x3_C3230RK)_Camera_Module_AVL_v1.6_20160226.pdf
    │   │   └── Rockchip_Trouble_Shooting_Android_CameraHAL1_CN_EN.pdf
    │   ├── HAL3
    │   │   ├── camera_engine_rkisp_user_manual_v2.2.pdf
    │   │   ├── camera_hal3_user_manual_v2.3.pdf
    │   │   ├── README_CN.txt
    │   │   ├── RKCIF_Driver_User_Manual_v1.0.pdf
    │   │   ├── RKISP1_IQ_Parameters_User_Guide_v1.2.pdf
    │   │   ├── RKISP_Driver_User_Manual_v1.3.pdf
    │   │   ├── Rockchip_Camera_Module_OTP_Calibration_Guide
    │   │   │   └── ISP30
    │   │   │       ├── CN
    │   │   │       │   └── Rockchip Camera Module OTP Calibration Guide_CN.pdf
    │   │   │       └── EN
    │   │   │           └── Rockchip Camera Module OTP Calibration Guide_EN.pdf
    │   │   ├── Rockchip_Color_Optimization_Guide_ISP
    │   │   │   ├── ISP21
    │   │   │   │   └── CN
    │   │   │   │       └── Rockchip_Color_Optimization_Guide_ISP21_CN_v2.0.1.pdf
    │   │   │   ├── ISP30
    │   │   │   │   ├── CN
    │   │   │   │   │   └── Rockchip_Color_Optimization_Guide_ISP30_CN_v3.0.2.pdf
    │   │   │   │   └── EN
    │   │   │   │       └── Rockchip_Color_Optimization_Guide_ISP30_EN_v3.0.2.pdf
    │   │   │   └── ISP32-lite
    │   │   │       └── CN
    │   │   │           └──
Rockchip_Color_Optimization_Guide_ISP32_Lite_CN_v3.1.0.pdf
    │   │   ├── Rockchip_Development_Guide_3A_ISP
    │   │   │   └── ISP30
    │   │   │       ├── CN
    │   │   │       │   └── Rockchip_Development_Guide_3A_ISP30_v1.1.0.pdf
    │   │   │       └── EN
    │   │   │           └── Rockchip_Development_Guide_3A_ISP30_EN_v1.1.0.pdf
    │   │   ├── Rockchip_Development_Guide_ISP
    │   │   │   ├── ISP21
    │   │   │   │   └── CN
    │   │   │   │       └── Rockchip_Development_Guide_ISP21_CN_v2.1.0.pdf
    │   │   │   ├── ISP30
    │   │   │   │   ├── CN
    │   │   │   │   │   └── Rockchip_Development_Guide_ISP30_CN_v1.2.5.pdf
    │   │   │   │   └── EN
    │   │   │   │       └── Rockchip_Development_Guide_ISP30_EN_v1.2.5.pdf
    │   │   │   └── ISP32-lite
    │   │   │       └── CN
    │   │   │           └── Rockchip_Development_Guide_ISP32_Lite_CN_v1.0.0.pdf
    │   │   ├── Rockchip_Driver_Guide_VI
    │   │   │   ├── CN
    │   │   │   │   └── Rockchip_Driver_Guide_VI_CN_v1.1.5.pdf
    │   │   │   └── EN
    │   │   │       └── Rockchip_Driver_Guide_VI_EN_v1.1.5.pdf
    │   │   ├── Rockchip_IQ_Tools_Guide_ISP
    │   │   │   ├── ISP21
    │   │   │   │   └── Rockchip_IQ_Tools_Guide_ISP2x_CN_v2.0.3.pdf
    │   │   │   ├── ISP30
    │   │   │   │   └── Rockchip_IQ_Tools_Guide_ISP21_ISP30_CN_v2.0.4.pdf
    │   │   │   └── ISP32-lite
    │   │   │       └── CN
    │   │   │           └── Rockchip_IQ_Tools_Guide_v2.0.7_CN.pdf
    │   │   ├── Rockchip_Trouble_Shooting_CameraHAL3_CN_EN.pdf
    │   │   ├── Rockchip_Tuning_Guide_ISP
    │   │   │   ├── ISP21
    │   │   │   │   └── CN
    │   │   │   │       └── Rockchip_Tuning_Guide_ISP21_CN_v2.1.0.pdf
    │   │   │   ├── ISP30
    │   │   │   │   ├── CN
    │   │   │   │   │   └── Rockchip_Tuning_Guide_ISP30_CN_v1.2.0.pdf
    │   │   │   │   └── EN
    │   │   │   │       └── Rockchip_Tuning_Guide_ISP30_EN_v1.2.0.pdf
    │   │   │   └── ISP32-lite
    │   │   │       └── CN
    │   │   │           └── Rockchip_Tuning_Guide_ISP32-lite_CN_v1.0.0.pdf
    │   │   └── USB_UVC_Integrated_Cameras.pdf
    │   ├── README.txt
    │   └── vehicle
    │       ├── Rockchip_Android_Fast_Reverse_Image_System_Developer_Guide_CN.pdf
    │       └── Rockchip_Developer_Guide_Automotive_Camera_CN.pdf
    ├── Can
    │   ├── Rockchip_Developer_Guide_Can_CN.pdf
    │   └── Rockchip_Developer_Guide_CAN_FD_CN.pdf
    ├── CLK
    │   ├── Rockchip_Developer_Guide_Clock_CN.pdf
    │   ├── Rockchip_Developer_Guide_Gpio_Output_Clocks_CN.pdf
    │   ├── Rockchip_Developer_Guide_Pll_Ssmod_Clock_CN.pdf
    │   └── Rockchip_RK3399_Developer_Guide_Clock_CN.pdf
    ├── CRYPTO
    │   ├── Rockchip_Developer_Guide_Crypto_HWRNG_CN.pdf
    │   └── Rockchip_Developer_Guide_Crypto_HWRNG_EN.pdf
    ├── DDR
    │   ├── DDR_bandwidth_statistics_tool
    │   │   ├── rk-msch-probe-for-user-32bit
    │   │   ├── rk-msch-probe-for-user-64bit
    │   │   └── Rockchip_Introduction_DDR_Bandwidth_Tool_CN.pdf
    │   ├── Rockchip_Developer_Guide_DDR_CN.pdf
    │   └── Rockchip_Developer_Guide_DDR_EN.pdf
    ├── debug
    │   ├── Rockchip_Developer_Guide_DS5_CN.pdf
    │   └── Rockchip_User_Guide_J-Link_CN.pdf
    ├── display
    │   ├── Rockchip_Android13_SDK_Developer_Guide_Multi_Screen_Display_CN.pdf
```

```
    │   ├── Rockchip_BT656_TX_AND_BT1120_TX_Developer_Guide_CN.pdf
    │   ├── Rockchip_Developer_Guide_Android_Car_MultiDisplay_CN.pdf
    │   ├── Rockchip_Developer_Guide_Baseparameter_Format_Define_And_Use_CN.pdf
    │   ├── Rockchip_Developer_Guide_DisplayPort_CN.pdf
    │   ├── Rockchip_Developer_Guide_DRM_Direct_Show_CN.pdf
    │   ├── Rockchip_Developer_Guide_DRM_Display_Driver_CN.pdf
    │   ├── Rockchip_Developer_Guide_DRM_Panel_Porting_CN.pdf
    │   ├──
Rockchip_Developer_Guide_Dual_Display_Rotation_Direction_Debugging_CN.pdf
    │   ├── Rockchip_Developer_Guide_eDP_CN.pdf
    │   ├── Rockchip_Developer_Guide_HDMI_Based_on_DRM_Framework_CN&EN.pdf
    │   ├── Rockchip_Developer_Guide_HDMI-CEC_CN.pdf
    │   ├── Rockchip_Developer_Guide_HDMI_CN.pdf
    │   ├── Rockchip_Developer_Guide_LVDS_CN.pdf
    │   ├── Rockchip_Developer_Guide_MIPI_DSI2_CN.pdf
    │   ├── Rockchip_Developer_Guide_RGB_MCU_CN.pdf
    │   ├── Rockchip_Developer_Guide_RK628_For_All_Porting_CN.pdf
    │   ├── Rockchip_Developer_Guide_RK628_For_All_Porting_EN.pdf
    │   ├── Rockchip_Developer_Guide_RK628_MCU_CN.pdf
    │   ├── Rockchip_Developer_Guide_RkDisplayOutputManager_CN.pdf
    │   ├── Rockchip_Developer_Guide_RkDisplayOutputManager_EN.pdf
    │   ├── Rockchip_DisplayPort_SI_Test_Guide_CN.pdf
    │   ├── Rockchip_DRM_RK628_Porting_Guide_CN.pdf
    │   ├── Rockchip FAQ DRM Hardware Composer V1.00-20181213.pdf
    │   ├── Rockchip_Introduction_DisplayAdjust_APK_CN.pdf
    │   ├── Rockchip_RK3399_Developer_Guide_DisplayPort_CN.pdf
    │   ├── Rockchip_RK3588_Developer_Guide_DisplayPort_CN.pdf
    │   ├── Rockchip_RK3588_Developer_Guide_HDCP_CN.pdf
    │   ├── Rockchip_RK3588_Developer_Guide_MIPI_DSI2_CN.pdf
    │   ├── Rockchip_RK3588_Developer_Guide_Vsync_Adjust_CN.pdf
    │   ├── Rockchip_RK3588_User_Guide_DP_CN.pdf
    │   ├── Rockchip_RK3588_User_Guide_eDP_CN.pdf
    ├── DVFS
    │   ├── Rockchip_Developer_Guide_CPUFreq_CN.pdf
    │   └── Rockchip_Developer_Guide_CPUFreq_EN.pdf
    ├── Ebook
    │   ├── Rockchip_RK3566_Introduction_EBOOK_Display_Mode_CN.pdf
    │   └── Rockchip_RK3566_Introduction_EBOOK_Sleep_Mode_CN.pdf
    ├── GMAC
    │   ├── Rockchip_Developer_Guide_Ethernet_CN.pdf
    │   ├── Rockchip_Developer_Guide_Linux_GMAC_CN.pdf
    │   ├── Rockchip_Developer_Guide_Linux_GMAC_DPDK_CN.pdf
    │   ├── Rockchip_Developer_Guide_Linux_GMAC_Mode_Configuration_CN.pdf
    │   ├── Rockchip_Developer_Guide_Linux_GMAC_RGMII_Delayline_CN.pdf
    │   ├── Rockchip_Developer_Guide_Linux_GMAC_RGMII_Delayline_EN.pdf
    │   └── Rockchip_Developer_Guide_Linux_MAC_TO_MAC_CN.pdf
    ├── hdmi-in
    │   ├── apk
    │   │   ├── HdmiInDemo_based_on_CameraHal1_2020.06.11_v1.2.tar.gz
    │   │   └── rkCamera2_based_on_CameraHal3_V1.3.tar.gz
    │   ├── Rockchip_Developer_Guide_Android12+_HDMI_IN_Bridge_CN.pdf
    │   ├── Rockchip_Developer_Guide_HDMI_CEC_CN.pdf
    │   ├── Rockchip_Developer_Guide_HDMI_IN_Based_On_CameraHal3_CN.pdf
    │   ├── Rockchip_Developer_Guide_HDMI_RX_CN.pdf
    │   └── Rockchip_RK3588_HDMI_To_CSI_Dual_Mipi_Developer_Guide.pdf
    ├── I2C
    │   ├── Rockchip_Developer_Guide_I2C_CN.pdf
    │   └── Rockchip_Developer_Guide_I2C_EN.pdf
    ├── IO-Domain
    │   ├── Rockchip_Developer_Guide_Linux_IO_DOMAIN_CN.pdf
    │   ├── Rockchip_PX30_Introduction_IO_Power_Domains_Configuration.pdf
    │   ├── Rockchip_RK3288_Introduction_IO_Power_Domains_Configuration.pdf
    │   ├── Rockchip_RK3326_Introduction_IO_Power_Domains_Configuration.pdf
    │   ├── Rockchip_RK3399_Introduction_IO_Power_Domains_Configuration.pdf
    │   ├── Rockchip_RK3399Pro_Introduction_IO_Power_Domains_Configuration.pdf
    │   └── Rockchip_RK356X_Introduction_IO_Power_Domains_Configuration.pdf
    ├── IOMMU
    │   ├── Rockchip_Developer_Guide_Linux_IOMMU_CN.pdf
    │   └── Rockchip_Developer_Guide_Linux_IOMMU_EN.pdf
    ├── MCU
    │   ├── Rockchip_RK3399_Developer_Guide_MCU_CN.pdf
    │   └── Rockchip_RK3399_Developer_Guide_MCU_EN.pdf
    ├── Memory
    │   ├── Rockchip_Developer_Guide_Linux_CMA_CN.pdf
    │   ├── Rockchip_Developer_Guide_Linux_DMABUF_CN.pdf
    │   └── Rockchip_Developer_Guide_Linux_Meminfo_CN.pdf
    ├── MMC
    │   └── Rockchip_Developer_Guide_SDMMC_SDIO_eMMC_CN.pdf
    ├── MPP
    │   ├── Rockchip_Developer_Guide_MPP_CN.pdf
    │   └── Rockchip_Developer_Guide_MPP_EN.pdf
    ├── NVM
    │   ├── Rockchip_Application_Notes_Storage_CN.pdf
    │   ├── Rockchip_Application_Notes_Storage_EN.pdf
    │   ├── Rockchip_Developer_FAQ_Storage_CN.pdf
    │   ├── Rockchip_Developer_Guide_Dual_Storage_CN.pdf
    │   ├── Rockchip_Developer_Guide_OTP_CN.pdf
    │   ├── Rockchip_Developer_Guide_OTP_EN.pdf
    │   ├── Rockchip_Developer_Guide_SATA_CN.pdf
    │   ├── Rockchip_Developer_Guide_UFS_CN.pdf
    │   ├── Rockchip_Introduction_Partition_CN.pdf
    │   ├── Rockchip_Introduction_Partition_EN.pdf
    │   ├── Rockchip_RK3568_Reference_SATA_Performance_CN.pdf
    │   ├── Rockchip_RK356X_Developer_Guide_SATA_CN.pdf
    │   └── Rockchip_Trouble_Shooting_Firmware_Upgrade_Issue_CN.pdf
    ├── PCie
    │   ├── Rockchip-Developer-Guide-linux4.4-PCIe.pdf
    │   ├── Rockchip_Developer_Guide_PCIe_CN.pdf
    │   ├── Rockchip_Developer_Guide_PCIE_EP_Standard_Card_CN.pdf
    │   ├── Rockchip_Developer_Guide_PCIe_Performance_CN.pdf
    │   ├── Rockchip_PCIe_Virtualization_Developer_Guide_CN.pdf
    │   └── Rockchip_RK3399_Developer_Guide_PCIe_CN.pdf
    ├── perf
    │   ├── Rockchip_Developer_FAQ_FileSystem_CN.pdf
    │   ├──
Rockchip_Developer_Guide_Linux_RealTime_Performance_Test_Report_CN.pdf
    │   ├──
Rockchip_Developer_Guide_Linux_RealTime_Performance_Test_Report_EN.pdf
    │   ├── Rockchip_Optimize_Tutorial_Linux_IO_CN.pdf
    │   ├── Rockchip_Quick_Start_Linux_Perf_CN.pdf
    │   ├── Rockchip_Quick_Start_Linux_Performance_Analyse_CN.pdf
    │   └── Rockchip_Quick_Start_Linux_Systrace_CN.pdf
    ├── PIN-Ctrl
    │   ├── Rockchip_Developer_Guide_Linux_Pinctrl_CN.pdf
    │   └── Rockchip_Developer_Guide_Linux_Pinctrl_EN.pdf
    ├── PMIC
    │   ├── Rockchip_Developer_Guide_FreeRTOS_PMIC_CHARGER_POWERKEY_CN.pdf
    │   ├── Rockchip_Developer_Guide_Power_Discrete_DCDC_EN.pdf
    │   ├── Rockchip_RK805_Developer_Guide_CN.pdf
    │   ├── Rockchip_RK806_Developer_Guide_CN.pdf
    │   ├── Rockchip_RK808_Developer_Guide_CN.pdf
    │   ├── Rockchip_RK809_Developer_Guide_CN.pdf
    │   ├── Rockchip_RK816_Developer_Guide_CN.pdf
    │   ├── Rockchip_RK817_Developer_Guide_CN.pdf
    │   ├── Rockchip_RK818_Developer_Guide_CN.pdf
    │   ├── Rockchip_RK818_RK816_Developer_Guide_Fuel_Gauge_CN.pdf
    │   └── Rockchip_RK818_RK816_Introduction_Fuel_Gauge_Log_CN.pdf
    ├── power
    │   ├── Rockchip_Developer_Guide_Power_Analysis_CN.pdf
    │   ├── Rockchip_Developer_Guide_Power_Analysis_EN.pdf
    │   └── Rockchip_Developer_Guide_Sleep_and_Resume_CN.pdf
    ├── PWM
    │   ├── Rockchip_Developer_Guide_Linux_PWM_CN.pdf
    │   └── Rockchip_Developer_Guide_Linux_PWM_EN.pdf
    ├── RGA
    │   ├── Rockchip_Developer_Guide_RGA_CN.pdf
    │   ├── Rockchip_Developer_Guide_RGA_EN.pdf
    │   ├── Rockchip_FAQ_RGA_CN.pdf
    │   └── Rockchip_FAQ_RGA_EN.pdf
    ├── RK628
    │   ├── Rockchip_Developer_Guide_RK628F_H_HDMI_IN_CN.pdf
    │   ├── Rockchip_Developer_Guide_RK628_For_All_Porting_CN.pdf
    │   └── Rockchip_RK628D_Application_Notes_CN.pdf
    ├── RKTools manuals
    │   ├── RKIQTool_User_Manual_v1.5-CH.pdf
    │   ├── RKIQTool_User_Manual_v1.5-EN.pdf
    │   ├── RK_Platform_apache_tomcat_ota_Server_Setup_Introduction.rar
    │   ├── Rockchip_Box_Factory_Test_Tool_V2.0.rar
    │   ├──
Rockchip_Introduction_MP_Tool_Upgrading_and_Related_Issues_Debugging_CN.pdf
    │   ├── Rockchip_Introduction_REPO_Mirror_Server_Build_and_Management_CN.pdf
    │   ├── Rockchip_Introduction_WNpctool_Write_Tool_CN.pdf
    │   ├── Rockchip_User_Guide_Box_Factory_Test_Tool_CN.pdf
    │   ├── Rockchip_User_Guide_Keybox_Burning_EN.pdf
    │   ├── Rockchip_User_Guide_KeyWrite_CN.pdf
    │   ├── Rockchip_User_Guide_MP_Flashing_v1.2_CN.pdf
    │   ├── Rockchip_User_Guide_Production_For_Firmware_Download_CN.pdf
    │   ├── Rockchip_User_Guide_RKDevInfoWriteTool_CN.pdf
    │   ├── Rockchip_User_Guide_RKDevInfoWriteTool_EN.pdf
    │   ├── Rockchip_User_Guide_RK_Platform_MP_Upgrading_CN.pdf
    │   ├── Rockchip_User_Manual_Android_Development_Tool_CN.pdf
    │   ├── Rockchip_User_Manual_RKIQTool_CN.pdf
    │   ├── Rockchip_User_Manual_RKIQTool_EN.pdf
    │   ├── Rockchip_User_Manual_Upgrade_DLL_CN.pdf
    │   └── SecureBootTool_UserManual.pdf
    ├── SARADC
    │   ├── Rockchip_Developer_Guide_Linux_SARADC_CN.pdf
    │   └── Rockchip_Developer_Guide_Linux_SARADC_EN.pdf
    ├── security
    │   ├── patch
    │   │   └── u-boot
    │   │       └── 0001-avb-add-embedded-key.patch
    │   ├── RK3399_Efuse_Operation_Instructions_V1.00_EN.pdf
    │   ├── RK356X_SecurityBoot_And_AVB_instructions_CN.pdf
    │   ├── RK356X_SecurityBoot_And_AVB_instructions_EN.pdf
    │   ├── RK3588_SecurityBoot_And_AVB_instructions_CN.pdf
    │   ├── RK3588_SecurityBoot_And_AVB_instructions_EN.pdf
    │   ├── Rockchip_Developer_Guide_Anti_Copy_Board_CN.pdf
    │   ├── Rockchip_Developer_Guide_Crypto_HWRNG_CN.pdf
    │   ├── Rockchip_Developer_Guide_Crypto_HWRNG_EN.pdf
    │   ├── Rockchip_Developer_Guide_OTP_CN.pdf
    │   ├── Rockchip_Developer_Guide_OTP_EN.pdf
    │   ├── Rockchip_Developer_Guide_Secure_Boot_Application_Note_EN.pdf
    │   ├── Rockchip_Developer_Guide_Secure_Boot_for_UBoot_Next_Dev_CN.pdf
    │   ├── Rockchip_Developer_Guide_Secure_Boot_for_UBoot_Next_Dev_EN.pdf
    │   ├── Rockchip_Developer_Guide_TEE_SDK_CN.pdf
    │   ├── Rockchip_Developer_Guide_TEE_SDK_EN.pdf
    │   ├── Rockchip_Introduction_Android_Root_CN.pdf
    │   ├── Rockchip_RK3399_User_Guide_SecurityBoot_And_AVB_CN.pdf
    │   └── Rockchip Vendor Storage Application Note.pdf
    ├── Sensors
    │   └── Rockchip_Developer_Guide_Sensors_CN.pdf
    ├── SPI
    │   ├── Rockchip_Developer_Guide_Linux_SPI_CN.pdf
    │   └── Rockchip_Developer_Guide_Linux_SPI_EN.pdf
    ├── Thermal
    │   ├── Rockchip_Developer_Guide_Thermal_CN.pdf
    │   └── Rockchip_Developer_Guide_Thermal_EN.pdf
    ├── TRUST
    │   ├── Rockchip_Developer_Guide_Trust_CN.pdf
    │   ├── Rockchip_Developer_Guide_Trust_EN.pdf
    │   ├── Rockchip_RK3308_Developer_Guide_System_Suspend_CN.pdf
    │   ├── Rockchip_RK3308_Developer_Guide_System_Suspend_EN.pdf
    │   ├── Rockchip_RK3399_Developer_Guide_System_Suspend_CN.pdf
    │   ├── Rockchip_RK356X_Developer_Guide_System_Suspend_CN.pdf
    │   ├── Rockchip_RK3576_Developer_Guide_System_Suspend_CN.pdf
    │   └── Rockchip_RK3588_Developer_Guide_System_Suspend_CN.pdf
    ├── Tutorial
    │   ├── RK3399-CPUINFO.pdf
    │   ├── RK3399-LOG-EXPLANATION.pdf
    │   ├── Rockchip_Developer_FAQ_Browser_CN.pdf
    │   ├── Rockchip_Developer_FAQ_FileSystem_CN.pdf
    │   └── Rockchip_Trouble_Shooting_Firmware_Upgrade_Issue_CN.pdf
    ├── UART
    │   ├── Rockchip-Developer-Guide-RT-Thread-UART.pdf
    │   ├── Rockchip_Developer_Guide_UART_CN.pdf
    │   ├── Rockchip_Developer_Guide_UART_EN.pdf
    │   └── Rockchip_Developer_Guide_UART_FAQ_CN.pdf
    ├── u-boot
    │   ├── Rockchip_Developer_Guide_Linux_AB_System_CN.pdf
    │   ├── Rockchip_Developer_Guide_UBoot_MMC_Device_Analysis_CN.pdf
    │   ├── Rockchip_Developer_Guide_UBoot_Nextdev_CN.pdf
    │   └── Rockchip_Developer_Guide_UBoot_Nextdev_EN.pdf
    ├── UFS
    │   └── Rockchip_Developer_Guide_UFS_CN.pdf
    ├── usb
    │   ├── Rockchip_Developer_Guide_Linux_USB_Initialization_Log_Analysis_CN.pdf
    │   ├── Rockchip_Developer_Guide_Linux_USB_Performance_Analysis_CN.pdf
    │   ├── Rockchip_Developer_Guide_USB2_Compliance_Test_CN.pdf
    │   ├── Rockchip_Developer_Guide_USB_CN.pdf
    │   ├── Rockchip_Developer_Guide_USB_EN.pdf
    │   ├── Rockchip_Developer_Guide_USB_FFS_Test_Demo_CN.pdf
```

```
    │   ├── Rockchip_Developer_Guide_USB_Gadget_UAC_CN.pdf
    │   ├── Rockchip_Developer_Guide_USB_SQ_Test_CN.pdf
    │   ├── Rockchip_Introduction_USB_SQ_Tool_CN.pdf
    │   ├── Rockchip_RK3399_Developer_Guide_USB_CN.pdf
    │   ├── Rockchip_RK356x_Developer_Guide_USB_CN.pdf
    │   ├── Rockchip_RK3576_Developer_Guide_USB_CN.pdf
    │   ├── Rockchip_RK3588_Developer_Guide_USB_CN.pdf
    │   ├── Rockchip_Trouble_Shooting_Linux4.19_USB_Gadget_UVC_CN.pdf
    │   └── Rockchip_Trouble_Shooting_Linux_USB_Host_UVC_CN.pdf
    └── watchdog
        ├── Rockchip_Developer_Guide_Linux_WDT_CN.pdf
        └── Rockchip_Developer_Guide_Linux_WDT_EN.pdf
```

10. Tool usage

10.1 StressTest

Use the Stresstest tool to do the stress test for the various functions on the target devices to make sure the whole system running stably. SDK can start StressTest application and perform stress test of various functions by entering `“83991906=”` code in the calculator.
The test items of Stresstest tool mainly include:

10.1.1 Module related

- Camera stress test: including Camera on/off, Camera taking photo and Camera switch.
- Bluetooth stress test: including Bluetooth on/off.
- Wi-Fi stress test: including Wi-Fi on/off, (plan to add ping test and iperf test).

10.1.2 Non module related

- Fly mode on/off test
- Suspend and resume stress test
- Video playing stress test
- Reboot stress test
- Recovery stress test
- ARM frequency scaling test
- GPU frequency scaling test
- DDR frequency scaling test

10.2 PCBA test tool

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

automatic test items, while key, SD card, USB Host, Codec are manual test items.
For the detailed PCBA function configuration and usage, please refer to:

```
RKDocs\android\Rockchip_Developer_Guide_PCBA_Test_Tool_CN&EN.pdf_V1.1_20171222.pd
f.
```

10.3 RKDeviceTest

RKDeviceTest is used for factory machine and basic aging tests, mainly testing whether the peripheral devices are normal after being installed into the whole machine and whether the basic stability of the whole machine is normal. For specific PCBA function configuration and instructions, please refer to:

```
RKDocs/android/Rockchip_User_Guide_Box_FactoryTestTool_V3.0_CN.pdf
```

In addition, if the customer needs to customize the tool, please contact the FAE window to apply for the corresponding source code.

10.4 USB driver

Rockchip USB driver install package includes ADB and image flashing driver

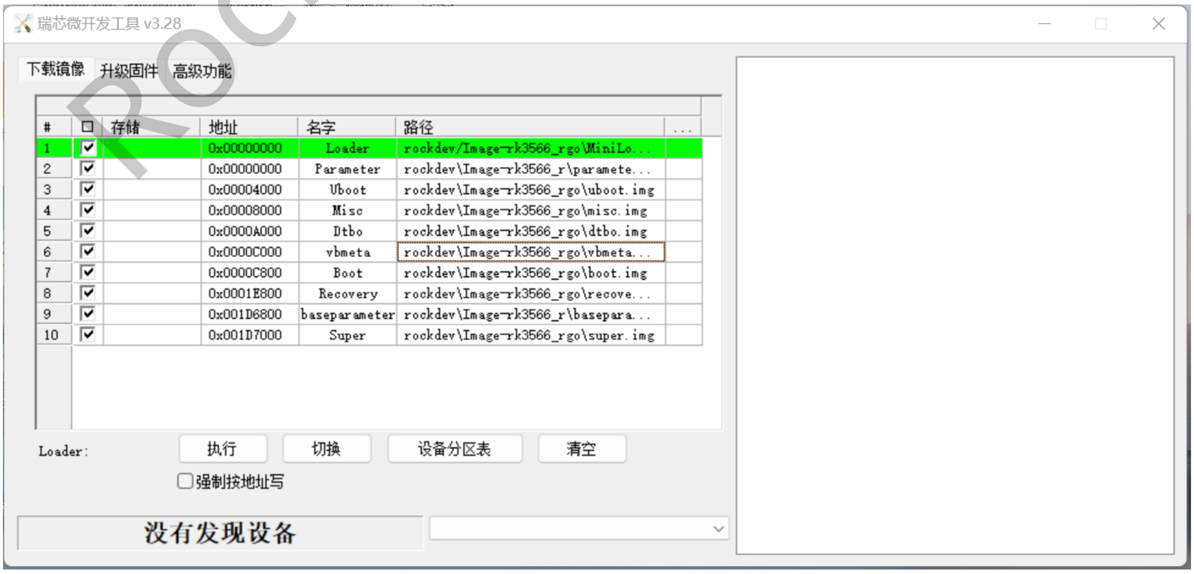
```
RKTools\windows\DriverAssitant_v5.1.1.zip
```

10.5 Development flashing tool

10.5.1 Windows version

RKTools/windows/AndroidTool/AndroidTool_Release_v3.28.zip

The tool version will update time by time, please synchronize and update in time.



#	□	存储	地址	名字	路径	...
1	☑		0x00000000	Loader	rockdev/Image-rk3566_rgo\MiniLo...	
2	☑		0x00000000	Parameter	rockdev\Image-rk3566_r\paramete...	
3	☑		0x00004000	Uboot	rockdev\Image-rk3566_rgo\uboot.img	
4	☑		0x00008000	Misc	rockdev\Image-rk3566_rgo\misc.img	
5	☑		0x0000A000	Dtbo	rockdev\Image-rk3566_rgo\dtbo.img	
6	☑		0x0000C000	vbmeta	rockdev\Image-rk3566_rgo\vbmeta...	
7	☑		0x0000C800	Boot	rockdev\Image-rk3566_rgo\boot.img	
8	☑		0x0001E800	Recovery	rockdev\Image-rk3566_rgo\recove...	
9	☑		0x001D6800	baseparameter	rockdev\Image-rk3566_r\basepara...	
10	☑		0x001D7000	Super	rockdev\Image-rk3566_rgo\super.img	

没有发现设备

10.5.2 Linux version

RKTools/linux/Linux_Upgrade_Tool/Linux_Upgrade_Tool_v2.30.zip

```
Linux_Upgrade_Tool_v2.26$ 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]
WriteSector:            WS  <BeginSec> <File>
ReadLBA:                RL  <BeginSec> <SectorLen> [File]
WriteLBA:               WL  <BeginSec> <File>
EraseLBA:               EL  <BeginSec> <EraseCount>
EraseBlock:             EB <CS> <BeginBlock> <BlokcLen> [--Force]
-------------------------------------------------------
```

10.6 Tool to implement SD upgrading and boot

It is used to implement SD card upgrading, SD card boot, SD card PCBA test.

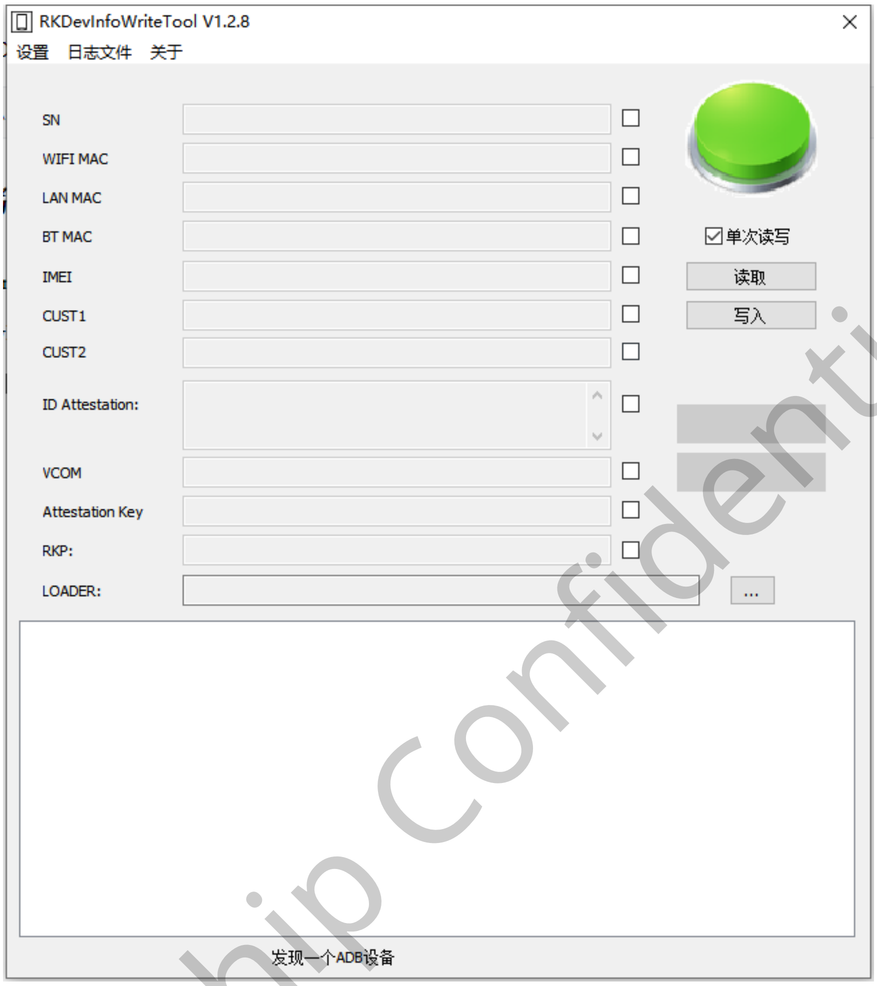
```
RKTools\windows\SDDiskTool_v1.74.zip
```

10.7 Write SN tool

RKTools\windows\RKDevInfoWriteTool-1.3.0.7z

Install after unzip RKDevInfoWriteTool-1.3.0.7z

Use admin ID to open the software



☑单次读写

发现一个ADB设备

For the tool instruction, please refer to:

```
RKDocs\common\RKTools manuals\Rockchip_User_Guide_RKDevInfoWriteTool_CN.pdf
```

10.8 DDR welding test tool

It is used to test DDR hardware conncection, troubleshooting hardware issues such as virtual welding.

```
RKTools\windows\Rockchip_Platform_DDR_Test_Tool_V1.39_Release_Annoucement.zip
```

10.9 efuse flashing tool

It is used to flash efuse, suitable for RK3288W/RK3368/RK3399 platforms.

```
RKTools\windows\efuse_v1.42.rar
```

10.10 efuse/otp sign tool

It is used to sign efuse/otp of image.

```
RKTools\windows\SecureBootTool_v1.99.zip
```

10.11 Factory production image flashing tool

It is uesed for batch image flashing in factory.

```
RKTools\windows\FactoryTool_v1.91.zip
```

10.12 userdata partition data prebuilt tool

It is the tool used to make userdata partition pre-built data package.

```
RKTools\windows\OemTool_v1.3.rar
```

10.13 Camera IQ Tool

It is used for debugging ISP image effects.

```
external/camera_engine_rkaiq/rkisp2x_tuner
```

11. System debugging

11.1 ADB tool

11.2 Overview

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

- Run the device shell (command line)
- Manage the port mapping of the simulator or the device
- Upload/download files between the computer and the device
- Install the local apk to simulator or Android device

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

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

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

11.2.1 USB adb usage

USB adb usage has the following limitations:

- Only support USB OTG port
- Not support multiple clients at the same time (such as cmd window, eclipse etc.)
- Support host connects to only one device but not multiple devices

The connection steps are as below:

1. The device already running Android system, setting -> developer option -> connect to the computer, enable usb debugging switch.
2. PC connects to the device USB otg port only through USB cable, and then the computer connects with Android device through below command:

```
adb shell
```

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

11.3 ADB commonly used command elaboration

1. Check the device situation

Check the Android device or simulator connected to computer:

```
adb devices
```

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

2. Install APK

Install the specific apk file to the device:

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

For example:

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

Re-install application:

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

3. Uninstall APK

Complete uninstall:

```
adb uninstall <package>
```

For example:

```
adb uninstall com.wishtv
```

4. Use rm to remove apk file:

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

For example:

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

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

5. Enter shell of the device and simulator

Enthe the shell environment of the device or simulator:

```
adb shell
```

6. Upload the file to the device from PC

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

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

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

7. Download the file from the device to PC

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

```
adb pull <remote path> <local path>
```

For example:

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

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

8. Check bug report

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

9. Check the device system information

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

```
adb shell getprop
```

11.4 Logcat tool

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

11.4.1 Logcat command usage

Use logcat command to check the contents of the system logcat buffer:
The basic format:

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

For example:

```
adb shell
logcat
```

11.5 The commonly used logcat filter method

Several ways to control the logcat output:

- Control the logcat output priority

For example:

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

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

- Control the logcat label and output priority

For example:

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

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

- Only output the logcat with the specific label

For example:

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

or

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

Note: only output the logcat with label of WishTV.

- Only output the logcat with the specific priority and label

For example:

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

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

11.6 Procrank tool

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

- VSS: Virtual Set Size The memory size used by virtual (including the memory used by the shared lib)
- RSS: Resident Set Size The actually used physical memory size (including the memory used by the shared lib)
- PSS: Proportional Set Size The actually used physical memory size (allocate the memory used by the shared lib in proportion)
- USS: Unique Set Size The physical memory used exclusively by the process (not including the memory used by the shared lib)

Note:

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

11.7 Use procrank

Make sure the terminal has the root authority before executing procrank

su

The command format:

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

The commonly used command instructions:

```
  -v: order by VSS
  -r: order by RSS
  -p: order by PSS
  -u: order by USS
  -R: convert to order by increasing[decreasing] method
  -w: only display the statistical count of working set
  -W: reset the statistical count of working set
  -h: help
```

For example:

Output the memory snapshot:

```
procrank
```

Output the memory snapshot in VSS decreasing order:

```
procrank –v
```

Procrank is output in PSS order by default.

11.8 Search the specific content information

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

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

cmdline means the target application name, PID means the target application process.
Output the memory status used by systemUI process:

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

or:

```
procrank  | grep 3396
```

11.9 Trace the process memory status

Analyze if there is memory leakage in the process by tracing the memory usage status. Use the script to continuously output the process memory snapshot, and compare with USS segment to see if there is memory leakage in this process.
For example: output the application memory usage of the process named com.android.systemui to see if there is leakage:

1. Write the script test.sh

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

2. After connect to the device by adb tool, run the script: ./test.sh

11.10 Dumpsys tool

Dumpsys tool is a debugging tool in Android system, running in the shell environment of the device, and provides the service status information running in the system. The running service means the service process in the Android binder mechanism.
The conditions for dumpsys to output the print:

1. Only print the services already loaded to ServiceManager.
2. If the dump function in the service code is not implemented, there will be no information output.

11.10.1 Use Dumpsys

- View Dumpsys help

Function: output dympsys help information.

```
dumpsys  -help
```

- View the service list of Dumpsys

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

```
dumpsys  -l
```

- Output the specific service information

Function: output the specific service dump information.
Format: dumpsys [servicename]
For example: execute below command can output the service information of SurfaceFlinger:

```
dumpsys SurfaceFlinger
```

- Output the specific service and application process information

Function: output the specific service and application process information.
Format: dumpsys [servicename] [application name]
For example: execute below command to output the memory information for the service named meminfo and process named com.android.systemui:

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

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

11.11 Last log enable

- Add the following two nodes in dts file

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

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

- Check last log in the device

```
130|root@rk3399:/sys/fs/pstore # ls
dmesg-ramoops-0   Log saved after last kernel panic
pmsg-ramoops-0    Log of last user space, android log
ftrace-ramoops-0    Print function trace during some period
console-ramoops-0  kernel log when last_log was enabled last time, but only save
the log with higher priority than default log level
```

- Usage:

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

11.12 FIQ mode

You can input fiq command through the serial port to check the system status when the device crashes or gets stuck. The specific command is as below:

```
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>
 reset [<c>]   Hard reset with command <c>
 irqs          Interupt status
 kmsg          Kernel log
 version       Kernel version
 sleep         Allow sleep while in FIQ
 nosleep        Disable sleep while in FIQ
 console        Switch terminal to console
 cpu            Current CPU
 cpu <number>   Switch to CPU<number>
 ps             Process list
 sysrq          sysrq options
 sysrq <param>  Execute sysrq with <param>
```

12. Common issues

12.1 What is current kernel version and u-boot version

The corresponding kernel version of Android14.0 is: 6.1, u-boot branch is next-dev branch

12.2 How to acquire the corresponding RK release version for current SDK

Rockchip Android14.0 Media Streaming SDK includes two parts: AOSP original code and RK modified code, where the RK modified repository is included in `.repo/manifests/include` In the xml below the directory, the default repository of AOSP is `.repo/manifests/default.xml` .
Version confirm:

- RK modification 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-14.0-ms-rkr1">
```

Means RK version is android-14.0-ms-rkr1

- AOSP part

```
vim .repo/manifests/default.xml
<default revision="u-tv-gsi-release"...>
```

Means OASP version is u-tv-gsi-release(Android 14 MR1 for TV)
Just provide the above two version information when needed.
You can directly acquire tag information through the following command for single lib:

```
kernel-6.1$ git tag
android-14.0-ms-rkr1
```

RK version is incremental with the format of android-14.0-ms-rkrxx, so current latest tag is android-14.0-ms-rkr1

12.3 How to confirm if local SDK is already updated to the latest SDK version released by RK

When RK SDK is released, the commit information corresponding to the version will be submited under the .repo/manifests/commit/ directory. Customers can confirm whether SDK is updated completely or not by comparing with the commit information. The specific operations are as follows:

- First confirm RK version of SDK according to the instruction of "How to acquire the corresponding RK release version for current SDK". Below take RKR1 version as example to introduce.
- Use the following command to save local commit information

```
.repo/repo/repo manifest -r -o release_manifest_rkr1_local.xml
```

- Comparing .repo/manifests/commit/commit_release_rkr1.xml with release_manifest_rkr1_local.xml can confirm whether SDK code is completely updated or not, while .repo/manifests/commit/commit_release_rkr1.xml is the commit information released along with RKR1 version.

12.4 Replace uboot and kernel logo picture

uboot and kernel logo are the first and second logo picture displayed during bootup, and they can be changed according to the product requirement.
uboot logo source file: `kernel-6.1/logo.bmp`
kernel logo source file: `kernel-6.1/logo_kernel.bmp`
If need to change the picture, just use the bmp with the same name to replace, and re-compile kernel. The compiled file is in boot.img.
Note: Logo picture size currently only supports to 8M with 8, 16, 24, 32bit bmp format.

12.5 How to modify Android system to support 64-bit only

The Android 14 products certifying GMS and EDLA are required to be configured to support 64-bit system only, not 32-bit one for reducing the memory usage. The detailed modification is as followed:(take rk3562_ugo as an example)

Modify corresponding configuration in the BoardConfig.mk of the product directory, as followed:

```
diff --git a/rk3562_ugo/BoardConfig.mk b/rk3562_ugo/BoardConfig.mk
index c06433e..dc9cc50 100644
--- a/rk3562_ugo/BoardConfig.mk
+++ b/rk3562_ugo/BoardConfig.mk
@@ -18,3 +18,11 @@ PRODUCT_KERNEL_DTS := rk3562-rk817-tablet-v10
 CAMERA_SUPPORT_AUTOFOCUS := true
 BOARD_BUILD_GKI := true
 include device/rockchip/rk3562/BoardConfig.mk
+
+DEVICE_IS_64BIT_ONLY := true
+
+TARGET_2ND_ARCH :=
+TARGET_2ND_ARCH_VARIANT :=
+TARGET_2ND_CPU_ABI :=
+TARGET_2ND_CPU_ABI2 :=
+TARGET_2ND_CPU_VARIANT :=
```

12.6 Power off charging and low battery precharging

Power off charging and low battery precharging can be configured in dts, as shown below:

```
        charge-animation {
                compatible = "rockchip,uboot-charge";
                rockchip,uboot-charge-on = <1>;
                rockchip,android-charge-on = <0>;
                rockchip,uboot-low-power-voltage = <3400>;
                rockchip,screen-on-voltage = <3500>;
                status = "okay";
        };
```

Note:
rockchip,uboot-charge-on: uboot power off charging is mutually exclusive with android power off charging
rockchip,android-charge-on: android power off charging is mutually exclusive with uboot power off charging
rockchip,uboot-low-power-voltage: configure the voltage for low battery precharging to boot, it can be configured according to the actual requirement
rockchip,screen-on-voltage: configure the voltage for low battery precharging to light the panel, it can be configured according to the actual requirement

12.7 Uboot charging logo package and replace

Charging logo path, you can directly replace with the file with the same name, and the format should be the same as 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 uboot charging is enabled, but there is no charging logo displaed, maybe it is because the picture is not packaged into resource.img. You can package per the following command:

```
cd u-boot
./scripts/pack_resource.sh ../kernel-6.1/resource.img
cp resource.img ../kernel/resource.img
```

After executing the above command, uboot charging logo will be packaged into resource.img in kernel directory. Now need to re-package resource.img into boot.img. You can execute ./mkiamge.sh in android root directory, and then flash boot.img under rockdev/.

12.8 HDMI IN configuration

hdmi in function is disabled in SDK by default. If need to enable, operate as below:

```
vim device/rockchip/rk3588/BoardConfig.mk
+BOARD_HDMI_IN_SUPPORT := true
```

12.9 RM310 4G configuration

4G function is disabled in SDK by default. If need to enable, operate as below:

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

12.10 WIFI Sleep Policy configuration

WIFI sleep policy is to keep sleep connecting, if you need to disconnect the sleep, please refer to the following:

```
---
a/rk3566_rgo/overlay/frameworks/base/packages/SettingsProvider/res/values/defaults.xml
+++
b/rk3566_rgo/overlay/frameworks/base/packages/SettingsProvider/res/values/defaults.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>
```

12.11 Recovery rotation configuration

Support Recovery rotation with 0/90/180/270 degree. Disabled by default (that is to rotate 0 degree) . The rotation configuration is described as below:

```
vim device/rockchip/common/BoardConfig.mk
#0:   ROTATION_NONE rotate 0 degree
#90:  ROTATION_RIGHT rotate 90 degrees
#180: ROTATION_DOWN rotate 180 degrees
#270: ROTATION_LEFT rotate 270 degrees
# For Recovery Rotation
TARGET_RECOVERY_DEFAULT_ROTATION ?= ROTATION_NONE
```

12.12 Android Surface rotation

For Android system display rotation, you can modify the following configuration with the parameters 0/90/180/270

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

12.13 Replace some remote of AOSP source code

The speed for customer to download RK release code is relatively slow. You can change the remote of AOSP to domestic mirror source, or Google mirror source for foreign customers, to improve the downloading speed. The detailed method is described as below:
After executing repo init (or unpacking base package), modify .repo/manifests/remote.xml. Change the remote fetch of AOSP from

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

to
for domestic customers: (here we take Tsinghua university mirror source as example. You can change to other domestic mirror source)

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

for foreign customers: (Google mirror source)

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

12.14 Data area read and write performance optimization

For devices with batteries, advised to add ‘fsync_mode=nobarrier’ to the data partition mounting parameter of fstab to improve storage read/write rates and performance. This parameter may cause data damage on devices without batteries. Therefore, it is not recommended to add this parameter to devices without batteries. Modified patches as follows:

```
cd device/rockchip/common

diff --git a/scripts/fstab_tools/fstab.in b/scripts/fstab_tools/fstab.in
index 2ec6c265..c890cc84 100755
--- a/scripts/fstab_tools/fstab.in
+++ b/scripts/fstab_tools/fstab.in
@@ -23,6 +23,6 @@ ${_block_prefix}odm     /odm      ext4 ro,barrier=1
${_flags},first_stage_mount
 # For sdmmc
 /devices/platform/${_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,keydirectory=/metadata/vold/metadata_encryption,quot
a,formattable,reservedsize=128M,checkpoint=fs
+/dev/block/by-name/userdata /data f2fs
noatime,nosuid,nodev,discard,reserve_root=32768,resgid=1065,fsync_mode=nobarrier
latemount,wait,check,fileencryption=aes-256-xts:aes-256-
cts:v2+inlinecrypt_optimized,keydirectory=/metadata/vold/metadata_encryption,quot
a,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
diff --git a/scripts/fstab_tools/fstab_go.in b/scripts/fstab_tools/fstab_go.in
index 582557f2..05c7653c 100755
--- a/scripts/fstab_tools/fstab_go.in
+++ b/scripts/fstab_tools/fstab_go.in
@@ -17,6 +17,6 @@ ${_block_prefix}odm     /odm      ext4 ro,barrier=1
${_flags},first_stage_mount
 # For sdmmc
 /devices/platform/${_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,keydirectory=/metadata/vold/metadata_encryption,quot
a,formattable,reservedsize=128M,checkpoint=fs
+/dev/block/by-name/userdata /data f2fs
noatime,nosuid,nodev,discard,reserve_root=32768,resgid=1065i,fsync_mode=nobarrier
latemount,wait,check,fileencryption=aes-256-xts:aes-256-
cts:v2+inlinecrypt_optimized,keydirectory=/metadata/vold/metadata_encryption,quot
a,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
```

12.15 Change userdata partition file system to EXT4

The default file system of data partition is f2fs. Recommend to change the file system of data partition to ext4 for the product without battery, as it can reduce the risk of data loss after abnormal power down. The modification method is as below:
Take rk3566_r as 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  ext4 ro,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      /radical_update      ext4
defaults                  defaults
```

12.16 Modify power on/off animation and tones

Reference document:

```
RKDocs\android\Rockchip_Introduction_Android_Power_On_Off_Animation_and_Tone_Cust
omization_CN&EN.pdf
```

12.17 APP performance mode setting

Configure the file: package_performance.xml in device/rockchip/rk3xxx/. Add the package names which need to use performance mode in the node: (use aapt dump badging (file_path.apk) to acquire the package name)

```
< app package="package name" mode="whether to enable the acceleration, 1 for
enable, 0 for disable"/>
```

Take antutu as example as below:

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

It will package the file into the image when compiling.

12.18 Debugging method of GPU related issues

You can do the initial debugging for the issues referring to the following documents.

```
RKDocs\android\Rockchip_User_Guide_Dr.G_CN&EN.pdf
```

12.19 OTP and efuse instruction

OTP support chipset

- RK3528
- RK3576
- RK3566
- RK3568
- RK3588

EFUSE support chipset

- RK3288
- RK3368
- RK3399

Refer to the document for image signing and otp/efuse flashing:

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

12.20 How to judge from the code whether OTP/EFUSE of the device is already flashed or not

The status of OTP/EFUSE will be transmitted through kernel cmdline and fuse.programmed in cmdline is used to mark the status of OTP/EFUSE. The details are as follows:

- "fuse.programmed=1": the software image package is already signed by secure-boot and efuse/otp of the hardware device is already flashed.
- "fuse.programmed=0": the software image package is already signed by secure-boot but efuse/otp of the hardware device is not flashed.
- there is no fuse.programmed in cmdline: the software image package is not signed by secure-boot (Miniloader doesn't tramsmit), or Miniloader is too old to support transmission.

12.21 Enable/disable selinux

Refer to the following modification, false to disable, true to enable

```
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
```

12.22 Warning “There's an internal problem with your device.” pops up after boot up

There are two reasons to pop up the warning:

1. Image mismatch, the fingerprints of system/boot/vendor are not consistent.
2. The device is enabled with a configuration that supports IO debugging. This problem can be solved by using the previous compile kernel command in the documentation.
3. For projects with IO debugging, regardless of the above two reasons, please merge the following patches directly to eliminate the warning:

```
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) {
```

12.23 How to enable the setting options for Ethernet in Settings

There is no default option of Ethernet setting in the system Settings. If Ethernet is needed in the project, it can be turned on as follows:

```
--- 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_ETHERNET := true
```

12.24 About AVB and security boot

For AVB and security boot related instruction and configurations, you can refer to the document

```
RKDocs/common/security/RK356X_SecurityBoot_And_AVB_instructions_CN.pdf
```

12.25 Cannot use IO commands

IO commands rely on DEVMEM which is disabled by default, so it is not able to use IO by default. If need to use IO commands for debugging, you can modify as follows:

```
wlq@ubuntu:~/rk3562_Android14.0/$ vim mkcombinedroot/configs/android-14.config
```

For GO produts, need to modify:

```
wlq@ubuntu:~/rk3562_Android14.0$ vim mkcombinedroot/configs/android-14-go.config
```

delete the following line:

```
# CONFIG_DEVMEM is not set
```

If you want to compile Android, you also need to modify the following code

```
cd rk3562_Android14.0/kernel/configs
```

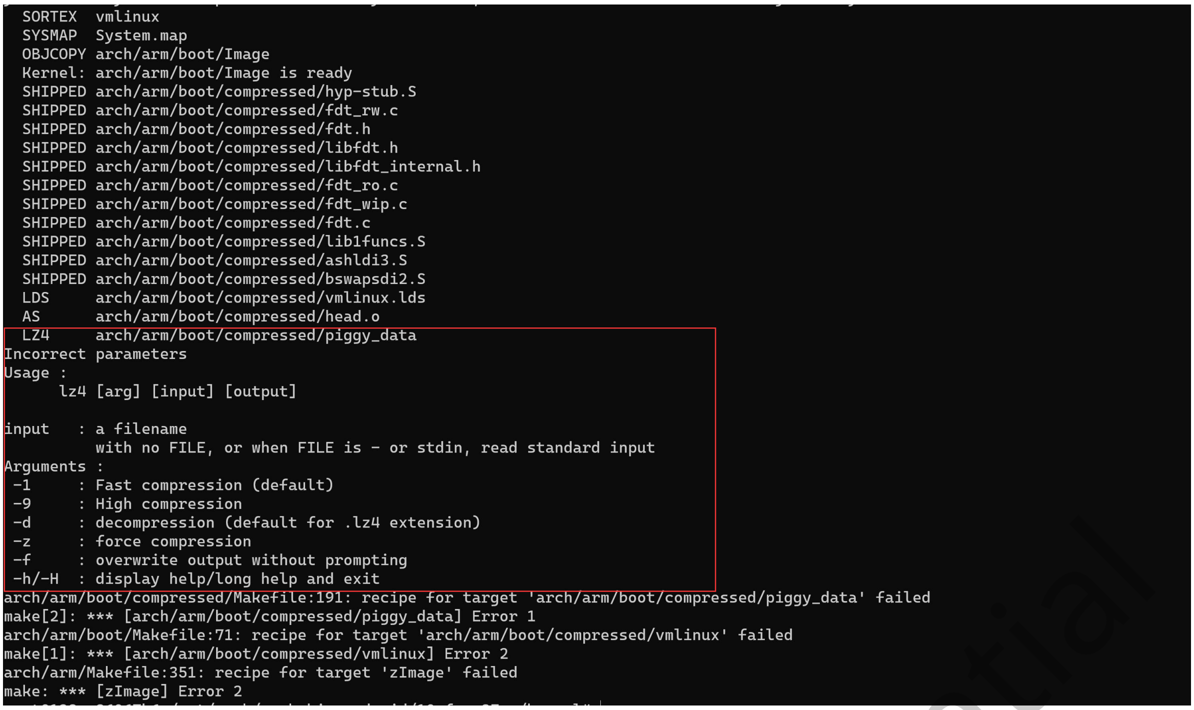
```
diff --git a/android-6.1/android-base.config b/android-6.1/android-base.config
index 5de76f0..6dcdf86 100644
--- a/android-6.1/android-base.config
+++ b/android-6.1/android-base.config
@@ -2,7 +2,6 @@
 # CONFIG_ANDROID_LOW_MEMORY_KILLER is not set
 # CONFIG_ANDROID_PARANOID_NETWORK is not set
 # CONFIG_BPFILTER is not set
-# CONFIG_DEVMEM is not set
 # CONFIG_FHANDLE is not set
 # CONFIG_FW_CACHE is not set
 # CONFIG_IP6_NF_NAT is not set
wlq@sys2206:~/rk3562_Android14.0/kernel/configs$ git diff
diff --git a/u/android-6.1/android-base.config b/u/android-6.1/android-
base.config
index 29b9e98..c1b21cf 100644
--- a/u/android-6.1/android-base.config
+++ b/u/android-6.1/android-base.config
@@ -2,7 +2,6 @@
 # CONFIG_ANDROID_LOW_MEMORY_KILLER is not set
 # CONFIG_ANDROID_PARANOID_NETWORK is not set
 # CONFIG_BPFILTER is not set
-# CONFIG_DEVMEM is not set
 # CONFIG_FHANDLE is not set
 # CONFIG_FW_CACHE is not set
 # CONFIG_IP6_NF_NAT is not set
```

12.26 The SN command rules

The SN must begin with a letter and be no more than 14 bytes.

12.27 Failer about LZ4 when Kernel compiling

kernel compiling fail log as:



```
  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/lib1funcs.S
  SHIPPED arch/arm/boot/compressed/ashldi3.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]

input   : a filename
          with no FILE, or when FILE is - or stdin, read standard input
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:
The LZ4 version of the system is too low, and the version 1.8.3 or above is required

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

solution:
copy the LZ4 compiled by Android to override the LZ4 of the system

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

12.28 Android Samba function

reference file

```
RKDocs/android/Rockchip_Introduction_Android_Samba_CN.pdf
```

12.29 NFS boot

Refer to the documents and patches:

```
RKDocs/android/patches/customized_functions/nfs_boot_patch_v1.1.0.zip
```

12.30 Update RK3528 DDR 4BIT Loader

```
diff --git a/RKBOOT/RK3528MINIALL.ini b/RKBOOT/RK3528MINIALL.ini
index a7e3779..6a952cf 100644
--- a/RKBOOT/RK3528MINIALL.ini
+++ b/RKBOOT/RK3528MINIALL.ini
@@ -14,7 +14,7 @@ Path1=bin/rk35/rk3528_usbplug_v1.03.bin
 NUM=2
 LOADER1=FlashData
 LOADER2=FlashBoot
-FlashData=bin/rk35/rk3528_ddr_1056MHz_v1.05.bin
+FlashData=bin/rk35/rk3528_ddr_1056MHz_4BIT_PCB_v1.05.bin
```

12.31 Multi-screen display and touch

Referenced document

```
RKDocs\android\patches\customized_functions/Android11异显开发说明.zip
```

12.32 Different screen different sound

Referenced document

```
RKDocs/android/patches/customized_functions/Dual_Audio_v1.0.zip
```

12.33 Boot video

The SDK supports native boot animation package playback function by default. The animation package is preset firmware /system/media/bootanimation.zip. To upgrade the animation package online, just download the animation package bootanimation.zip to the machine /data/local/bootanimation.zip, and restart to play the new animation package content.

Press up video preset: Configure BOOT_VIDEO_ENABLE ?= true in the product device/rockchip/common/BoardConfig.mk, and prepare the following corresponding boot video file bootanimation.ts (the default code recognizes this video suffix. Other formats can directly modify its file name and suffix to bootanimation.ts, and there is no need to convert the format). After compilation, the corresponding resource file will be copied to the corresponding out directory.

Copy the boot video to device/rockchip/common/bootvideo/bootanimation.ts (source path).

Note: After turning on the video function, the video will be played by default. The playback time can be controlled through the attribute persist.sys.bootvideo.showtime:

> -1: It means that the duration is not set, and it is displayed according to the natural stage of the startup;
>
> -2: It means that the video must be played before entering the launcher;

Configure other numbers greater than 0 to indicate the specific playback time, and play in 120 seconds after more than 120 seconds.

To upgrade the boot video online, just download the video material bootanimation.ts to the machine /data/local/bootanimation.ts and restart it.

12.34 Media Center

MediaCenter is a set of internal storage, SD card, USB drive, mobile hard disk, Samba storage, NFS storage services and other settings.

The Box TV app is installed in one scan discovery, file browsing, video browsing, music browsing, picture browsing, and APK. Should

For application adaptation to the RK series chip platform, please refer to "Rockchip_Introduction_Box_Media_Application_CN.pdf" for details of the function introduction.

12.35 UiMode Preconfiguration

In BOX type products, since the default UI resources and layout use the televison style (i.e. the UIMode configuration in this article), if customers need to integrate applications that are only compatible with phone/mid style, they may cause unusable exceptions due to compatibility issues of the apk itself. Now we provide a way to dynamically load UIMode for the App through whitelist configuration.

Source code path: device/rockchip/common/uimode/package_uimode_config.xml, device path: vendor/etc/uimode_app.xml.

Currently, this method only supports BOX devices. Please refer to device/rockchip/common/uimode/ReadME for usage.

To facilitate debugging, customers can select the uimode to be tested in the lowest uimode for debugging during the machine operation by entering the TVSettings-->app-->such as the netflix application settings menu.

12.36 Adjustment and saving of display parameters

The RK platform provides a special partition to save display parameters from which the system can read and apply the display parameters. We call this partition baseparaemter partition, and its main functions include:

```
1. Provide system and applications to save and adjust the display information
position such as resolution, frame rate, color depth and color space supported by
display devices such as hdmi, dp or cvbs;

2. Maintain the independence and unity of display information and optimize the
experience of unified display effects at each stage of startup;

3. Provide default display configuration backup, exceptions are used for
recovery, etc.
```

This partitioning function is enabled by default in the SDK. When the firmware is packaged, the baseparameter.img image will be automatically packaged for burning.

For an update display partition function, please refer to the document:

RKDocs/common/display/Rockchip_Developer_Guide_Baseparameter_Format_Define_And_Use_CN.pdf.

12.36.1 How to modify the default HDMI resolution

The SDK defaults to write the display partition mirror for the first configuration in adaptive mode, that is, it is automatically configured according to the optimal resolution of the TV feedback. If the project needs a fixed HDMI resolution configuration, it is necessary to modify the display partition mirror content in the corresponding platform product catalog, such as the corresponding path of the RK3528 display mirror:

device/rockchip/rk3528/etc/baseparameter_auto.img

Customized display partition mirrors can be generated through settings and commands in the burned machine. The specific methods are as follows:

1. Confirm that the SDK source code contains the following uboot and libparammeter submissions. If there is no need to update or patch support:

```
uboot:
commit 4cc4b116aad028770f51d3e5e2335632b889f276
Author: Damon Ding <damon.ding@rock-chips.com>
Date:   Thu Mar 28 15:06:52 2024 +0800

    video/drm: vop2: disable aclk pre auto gating in global init process

    This is a workaround for RK3528/RK3562/RK3576:

    The aclk pre auto gating function may disable the aclk
    in some unexpected cases, which detected by hardware
    automatically.

    For example, if the above function is enabled, the post
    scale function will be affected, resulting in abnormal
    display.

    Signed-off-by: Damon Ding <damon.ding@rock-chips.com>
    Change-Id: Ib0aa6828de57ea35140b42d6c9e21ee3512837b6
    (cherry picked from commit bb7ec3565f0348e393762342c0b2486b15eac1fb)

hardware/rockchip/libbaseparameter:
commit bce92033c35b5e90ce826d11185af78a9f6f332c
Author: huangjc <huangjc@rock-chips.com>
Date:   Fri Mar 22 11:29:31 2024 +0800

    fix box set_disp_info fail

    Change-Id: If983b1453229b2d93608de92dd02b9f38f780675
    Signed-off-by: huangjc <huangjc@rock-chips.com>

commit 774dee5f709be98e8f23d4e2e9cc403398dd7838
Author: huangjc <huangjc@rock-chips.com>
Date:   Fri Mar 22 09:15:45 2024 +0800

    api: update backup data for dump_baseparameter

    Change-Id: I36541da3731dba5d4abd178bb00e358af9b9dce7
    Signed-off-by: huangjc <huangjc@rock-chips.com>
```

2. Fry or update the firmware that has already contained the above submitted or patches to the machine. The machine is connected to HDMI. Only by displaying it in the settings can you select the required default HDMI resolution. After the switch is successful, perform the next step;
3. Run the command dump under the serial port or adb shell to display the partition image, such as the default 720p50 resolution img is generated below;
4. Extract the generated img and replace the SDK source code to display the partition image file by default;

```
device/rockchip/rk3528/etc/baseparameter_auto.img

rk3528_box_32:/ # saveBaseParameter -p -t /sdcard/baseparameter-720p60.img
print baseparameter
save to /sdcard/baseparameter-720p60.img (-t)
========== base parameter ==========
-connector type: 11 connector id: 0 offset: 104
        resolution: 1280x720@p60-1390-1430-1650-725-730-750-5 clk=74250
        corlor: format 4 depth 0
        feature:  0x3
        fbinfo: 0x0@60
        bcsh: 50 50 50 50
        overscan: 88 80 88 80
        gamma size:0
        3dlut size:0
-connector type: 11 connector id: 1 offset: 37040
        resolution: 1920x1080@p60-2008-2052-2200-1084-1089-1125-5 clk=148500
        corlor: format 0 depth 0
        feature:  0x0
        fbinfo: 0x0@60
        bcsh: 50 50 50 50
        overscan: 100 100 100 100
        gamma size:0
        3dlut size:0
-connector type: 13 connector id: 0 offset: 73976
        resolution: 0x0@p0-0-0-0-0-0-0-0 clk=0
        corlor: format 0 depth 0
        feature:  0x0
        fbinfo: 0x0@60
        bcsh: 50 50 50 50
        overscan: 100 100 100 100
        gamma size:0
        3dlut size:0

========== backup parameter ==========
-connector type: 11 connector id: 0 offset: 104
        resolution: 0x0@p0-0-0-0-0-0-0-0 clk=0
        corlor: format 0 depth 0
        feature:  0x1
        fbinfo: 0x0@60
        bcsh: 50 50 50 50
        overscan: 100 100 100 100
        gamma size:0
        3dlut size:0
-connector type: 11 connector id: 1 offset: 37040
        resolution: 1920x1080@p60-2008-2052-2200-1084-1089-1125-5 clk=148500
        corlor: format 0 depth 0
        feature:  0x0
        fbinfo: 0x0@60
        bcsh: 50 50 50 50
        overscan: 100 100 100 100
        gamma size:0
        3dlut size:0
-connector type: 13 connector id: 0 offset: 73976
        resolution: 0x0@p0-0-0-0-0-0-0-0 clk=0
        corlor: format 0 depth 0
        feature:  0x0
        fbinfo: 0x0@60
        bcsh: 50 50 50 50
        overscan: 100 100 100 100
        gamma size:0
        3dlut size:0
====================================
save to /sdcard/baseparameter-720p60.img successfully

E:\A-TV\rk3126\adb>adb pull /sdcard/baseparameter-720p60.img
```

12.36.2 How to modify the default HDMI AUTO to select the maximum resolution configuration

The SDK defaults to write the display partition mirror for the first configuration in adaptive mode, that is, it is automatically configured according to the optimal resolution of the TV feedback. If the project AUTO mode is changed to automatic maximum resolution, you can refer to the following patch to modify and verify:

```
1. hardware/rockchip/hwcomposer/drmhwc2:
diff --git a/drm/drmconnector.cpp b/drm/drmconnector.cpp
index 91cf31f..f3444ce 100644
--- a/drm/drmconnector.cpp
+++ b/drm/drmconnector.cpp
@@ -291,7 +291,7 @@ int DrmConnector::UpdateModes() {
         // Use only the first DRM_MODE_TYPE_PREFERRED mode found
         if (!preferred_mode_found && (new_modes.back().type() &
DRM_MODE_TYPE_PREFERRED)) {
             preferred_mode_id_ = new_modes.back().id();
-            preferred_mode_found = true;
+            preferred_mode_found = false;
         }
     }
     modes_.swap(new_modes);
@@ -573,13 +573,14 @@ int DrmConnector::GetBestDisplayMode(int display_id, int
update_base_timeline) {
         }
     }

-    for (const DrmMode& conn_mode : modes()) {
+  /*  for (const DrmMode& conn_mode : modes()) {
         if (conn_mode.type() & DRM_MODE_TYPE_PREFERRED) {
             set_best_mode(conn_mode);
             ALOGI_BEST_MODE_INFO(conn_mode);
             return 0;
         }
     }
+  */
```

```
     // 20230104
     for (const DrmMode& conn_mode : modes()) {

2. kernel-6.1:
diff --git a/drivers/gpu/drm/drm_modes.c b/drivers/gpu/drm/drm_modes.c
index 808345b17956..292d138dc848 100644
--- a/drivers/gpu/drm/drm_modes.c
+++ b/drivers/gpu/drm/drm_modes.c
@@ -1325,10 +1325,6 @@ static int drm_mode_compare(void *priv, struct list_head
*lh_a, struct list_head

    struct drm_display_mode *b = list_entry(lh_b, struct drm_display_mode, head);
    int diff;

-   diff = ((b->type & DRM_MODE_TYPE_PREFERRED) != 0) -
-       ((a->type & DRM_MODE_TYPE_PREFERRED) != 0);
-   if (diff)
-       return diff;

    diff = b->hdisplay * b->vdisplay - a->hdisplay * a->vdisplay;
    if (diff)
        return diff;

3. u-boot:
diff --git a/common/edid.c b/common/edid.c
index d7d224cc58..6e1a0b9e23 100644
--- a/common/edid.c
+++ b/common/edid.c
@@ -6664,16 +6664,6 @@ void drm_mode_sort(struct hdmi_edid_data *edid_data)

        a = &edid_data->mode_buf[i];
        for (j = i + 1; j < edid_data->modes; j++) {
            b = &edid_data->mode_buf[j];
-           diff = ((b->type & DRM_MODE_TYPE_PREFERRED) != 0) -
-               ((a->type & DRM_MODE_TYPE_PREFERRED) != 0);
-           if (diff) {
-               if (diff > 0) {
-                   c = *a;
-                   *a = *b;
-                   *b = c;
-               }
-               continue;
-           }

            diff = b->hdisplay * b->vdisplay
                - a->hdisplay * a->vdisplay;
```

12.37 HDMI CEC

The SDK has supported most functions of the HDMI CEC protocol. The specific function menu is: Settings - Device Preferences - Input Port, which supports switch CEC function, CEC standby, and CEC wake-up function respectively. More supported functions are as follows:

cec feature	effect
set menu language	Change the menu language of tv, and change the box to the corresponding menu language
one touch play	Support control of tv startup and default switch to stb input source port
standby	tv or box standby, so that the other party can standby
wakeup	tv or box wake up to wake up the other party as well
devicee osd name transfer	Set-top box is powered off and restarted to check whether the input source device name displayed on the TV is defined by the box
remote control pass through	1. Control the power supply (standby), volume up and down, and mute functions of the TV through the set-top box remote control; the TV can be turned off, volume up and down, and mute 2. Use the TV to operate the power supply (standby), up and down, left and right, confirm, and return; the set-top box has corresponding reactions
system information	Enter dumpsys hdmi_control command to see system information print
power status	Enter dumpsys hdmi_control command to see the power status print
general protocol	Enter dumpsys hdmi_control command to see general protocol print

Related to hibernation wake-up, please refer to the following table for specific support for testing:

Serial number	Module	Test steps	Test results
1	CEC	1. Stb connects to CEC TV, tv wake-up state, stb wake-up state; 2. Operates tv standby, stb automatically standby	PASS
2	CEC	1. Stb connects to CEC TV, tv wake-up state, stb wake-up state; 2. Operates stb standby, tv automatically standby	PASS
3	CEC	1. Stb connects to CEC TV, tv standby state, stb standby state; 2. Operation stb wake-up, tv automatically wakes up	PASS
5	CEC	1. stb connects to CEC TV, tv wake-up state, stb wake-up state; 2. Operate the tv remote control up, down, left, right, confirm and other buttons, stb responds correctly	PASS
6	CEC	1. Stb connects to CEC TV, tv wake-up state, stb wake-up state; 2. Unplug the HDMI access port from HDMI 1, insert it into HDMI 2 port, and tv automatically switches to HDMI 2 port display	PASS

Android9 native HDMI CEC function provides the Settings database field mode to control some functions, and the main fields are described as follows:

Fields	Defaults	Function Description
Settings.Global.HDMI_CONTROL_ENABLED	1	System HDMI CEC service enable switch settings put global hdmi_control_enabled 1/0
Settings.Global .HDMI_CONTROL_AUTO_WAKEUP_ENABLED	1	HDMI CEC wakeup enable switch settings put hdmi_control_auto_wakeup_enabled 1/0
Settings.Global .HDMI_CONTROL_AUTO_DEVICE_OFF_ENABLED	0	HDMI CEC standby enable switch settings put global hdmi_control_auto_device_off_enabled1/0

In addition, HDMI CEC standby also supports functions such as setting timeout through attributes. The relevant attributes and functions are described as follows:

Properties	Function Description
persist.sys.CECStanbytimeout	Configure the HDMI CEC standby timeout time to control how long it takes to trigger the system standby after triggering the CEC standby message, unit milliseconds, such as 1 minute: 60 * 1000
persist.sys.HdmiStandby	Switch HDMI standby function, that is, control whether the standby function is triggered by unplugging HDMI. True is on, false is turned off
persist.sys.TvStanbytimeout	Configure the HDMI standby timeout time to control how long it takes to trigger the system standby operation after the HDMI is plugged out, unit seconds, such as 1 minute: 60

12.37.1 How to support HDMI-CEC wake-up box after shutdown

The default SDK supports the standby wake-up function of HDMI-CEC standard. Some projects need to shutdown when shutting down, and they also need to wake up the TV through the TV remote control and wake up the shutdown box function like infrared remote control. You can refer to the kernel below to modify it:

```
diff --git a/drivers/gpu/drm/rockchip/dw_hdmi-rockchip.c
b/drivers/gpu/drm/rockchip/dw_hdmi-rockchip.c
index b55dd36a5b09..02810caba8c2 100644
--- a/drivers/gpu/drm/rockchip/dw_hdmi-rockchip.c
+++ b/drivers/gpu/drm/rockchip/dw_hdmi-rockchip.c
@@ -2102,8 +2102,10 @@ static void dw_hdmi_rockchip_shutdown(struct
platform_device *pdev)

    if (hdmi->hpd_gpiod) {
        disable_irq(hdmi->hpd_irq);
+#if 0
        if (hdmi->hpd_wake_en)
            disable_irq_wake(hdmi->hpd_irq);
+#endif
    }
    dw_hdmi_suspend(&pdev->dev, hdmi->hdmi);
    pm_runtime_put_sync(&pdev->dev);
diff --git a/drivers/soc/rockchip/rockchip_pm_config.c
b/drivers/soc/rockchip/rockchip_pm_config.c
index 6c12ef87ccda..335464246b3c 100644
--- a/drivers/soc/rockchip/rockchip_pm_config.c
+++ b/drivers/soc/rockchip/rockchip_pm_config.c
```

```
@@ -76,7 +76,8 @@ static void rockchip_pm_virt_pwroff_prepare(void)
        pr_err("Disable nonboot cpus failed!\n");
        return;
    }

+   sip_smc_set_suspend_mode(7, 1, 135);

    sip_smc_set_suspend_mode(VIRTUAL_POWEROFF, 0, 1);
    sip_smc_virtual_poweroff();
```

12.38 AI Lab Function

The SDK supports advanced AI display, AI audio, AI real-time subtitles and other functions, with control of each sub-function switch set through the AI lab; refer to the documentation for details.

```
RKDocs/android/ailab/Rockchip_Developer_Guide_AILab_CN.pdf
```

12.39 WifiDisplay(Miracast)

The SDK supports WifiDisplay function and provides applications that implement Miracast sink function, namely the WFD screen projection function. The specific source code is vendor/rockchip/common/apps/WifiDisplay.

12.40 DLNA

The SDK supports multi-screen interactive DLNA function, mainly implementing DLNA-defined Digital Media Server (DMS),

There are three types of device: Digital Media Player (DMP) and Digital Media Renderer (DMR). The specific functions of each device type are as follows:

> 1. DMS:
>
> Digital media servers mainly realize the storage and sharing of media resources, and can provide stored media content to networked DMP (digital media player) and DMR (digital media renderer) access.
>
> 2. DMP:
>
> Digital media player mainly realizes finding and obtaining content sent by DMS or M-DMS, and provides playback, rendering, and broadcast control functions.
>
> 3. DMR:
>
> Digital Media Renderer, plays content played from DMS or M-DMS after being configured through other devices. The difference between DMR and DMP is that DMR needs to be controlled by DMC (Digital Media Controller) to obtain the content sent by DMS.

Our common screen projection to boxes generally use DMR function. By default, you need to open the DLNA application first, specifically in vendor/rockchip/common/apps/DLNA.

12.41 Add key value of infrared remote control

The infrared remote control function currently supports the remote control protocol in NEC encoding format, and other encoding methods are not yet supported.

The encoding format of infrared remote controls is usually available in two formats: NEC and RC5

Characteristics of NEC format:

- Use 38 kHz carrier frequency
- The boot code interval is 9 ms + 4.5 ms
- Use 16-bit customer code
- Use 8-bit data code and 8-bit inverse data code

Whether it is to adapt to a new infrared remote control or to add a new key value method for the remote control, it is mainly divided into 2 steps, as follows:

Step 1: Print new remote control new key value:

1. Turn on the debug switch for printing key values

You can enter it through adb or serial port:

```
echo 1 > sys/module/rockchip_pwm_remotectl/parameters/code_print
#Adb shell
logcat -b kernel
```

2. Press the remote control button to record the corresponding key value

For example, press the left arrow key of the remote control and print it as follows. The usercode (system code) of the remote control is 0xfe01, and the physical key value of the left arrow key is 0x63.

It should be noted that some standard tables of infrared physical key values use the last 8 bit data, which is complementary to the printed key value code. When testing and confirming, it needs to be converted. The following prints as mute keys. The physical key value reported by the driver is the first 8 bits: 0x63. ** Some standard tables indicate that the requirements are 0x9c (take the last 8 bits: 0xff-0x63) are also correct.

```
[ 592.280559] USERCODE=0xfe01

[ 592.307146] RMC_GETDATA=63
```

Repeat this until all key values on the remote control are recorded.

Step 2: Key-value mapping kl.

1. Low-level key-value mapping: Add the key value of each key recorded above and the usercode of the remote control to the &pwm3 {} key-value mapping structure table of the corresponding project dts file. RK3528 key mapping table is unified in kernel/arch/arm64/boot/dts/rockchip/rk-stb-ir-keymap.dtsi.

```
~/A14_MS/kernel-6.1$ vi arch/arm64/boot/dts/rockchip/rk-stb-ir-keymap.dtsi
...
/* for IPTV */
     ir_key4 {/*ir_keyX
                rockchip,usercode = <0x4db2>;
                rockchip,key_table =
                        <0x31   KEY_REPLY>,
                        <0x3a   KEY_BACK>,
                        <0x35   KEY_UP>,
                        <0x2d   KEY_DOWN>,
                        <0x66   KEY_LEFT>,
                        <0x3e   KEY_RIGHT>,
                        <0x7f   KEY_VOLUMEUP>,
                        <0xfe   KEY_VOLUMEDOWN>,
                        <0x23   KEY_POWER>,
                        <0x63   KEY_MUTE>,
                        <0x6d   KEY_1>,
                        <0x6c   KEY_2>,
                        <0x33   KEY_3>,
                        <0x71   KEY_4>,
                        <0x70   KEY_5>,
                        <0x37   KEY_6>,
                        <0x75   KEY_7>,
                        <0x74   KEY_8>,
                        <0x3b   KEY_9>,
                        <0x78   KEY_0>,
                        <0x73   KEY_PAGEDOWN>,
                        <0x22   KEY_PAGEUP>,
                        <0x72   KEY_SETUP>,
                        <0x7a   KEY_CHANNEL_UP>,
                        <0x79   KEY_CHANNEL_DN>,
                        <0x77   KEY_HOME_PAGE>,
                        <0x29   KEY_CH_CUT_BACK>,
                        <0x32   KEY_DIRECT_SEEDING>,
                        <0x6e   KEY_REVIEW>,
                        <0x7c   KEY_ON_DEMAND>,
                        <0x3c   KEY_INFO1>,
                        <0x67   KEY_SOUND1>,
                        <0x25   KEY_X1>,
                        <0x2f   KEY_X2>,
                        <0x7d   KEY_LOCAL>,
                        <0x6a   KEY_PLAYPAUSE>,
                        <0x0b   KEY_EQUAL>;
        };
```

2. System layer mapping kl: Refer to the mapping configuration added in the kernel to confirm that the kl matches. The kl file is in device/rockchip/rk3528/rk3528_box$ ffa90030_pwm.kl.

```
huangjc@171server:~/rk3528_9.0/workspace$ vi
device/rockchip/rk3528/rk3528_box/ffa90030_pwm.kl
#$_FOR_ROCKCHIP_RBOX_$
#$_rbox_$_modify_$_chenzhi_20120220: add for IR remote

#key xx,xx for kernel,  ENTER for
android:frameworks/native/include/input/InputEventLabels.h
key 28     ENTER
key 116    POWER
key 158    BACK
key 139    MENU
key 217    SEARCH
key 141    SETTINGS
key 164    MEDIA_PLAY_PAUSE
key 232    DPAD_CENTER
key 108    DPAD_DOWN
key 103    DPAD_UP
key 102   HOME
key 104   CHANNEL_UP
key 105   DPAD_LEFT
key 106   DPAD_RIGHT
key 109   CHANNEL_DOWN
key 115   VOLUME_UP
key 114   VOLUME_DOWN
key 110   INSERT
key 111   FORWARD_DEL
#key 143   NOTIFICATION
key 64          F6
key 65          F7
key 66          F8
key 67          F9
key 142   POWER
key 143   POWER
key 152   POWER
key 172   HOME
key 176   SETTINGS
key 177   PAGE_UP
key 178   PAGE_DOWN
key 92    PAGE_UP
key 93    PAGE_DOWN
key 500   MOBILE_M
key 14    DEL
key 113   VOLUME_MUTE
key 388   TV_KEYMOUSE_MODE_SWITCH
key 2     1
key 3     2
key 4     3
key 5     4
key 6     5
key 7     6
key 8     7
key 9     8
key 10    9
key 11    0
```

Note: If you need to add system keys, in addition to adding definitions or mappings in the kernel, you should also add corresponding definitions in the system frameworks. Otherwise, if you only add them in kl, the startup parsing of kl will fail to cause the remote control key to be invalid. For details, refer to the corresponding submission of frameworks/native/include/input/InputEventLabels.h, and the corresponding submission of frameworks/base/core/java/android/view/KeyEvent.java.

Introduction to the additional functions related to the button:

1. The infrared reception function method of the IO command switch is as follows. It can be used in a grill and other scenarios to shield infrared interference:

```
close:
io -4 -w 0xffa9003c 0x00006004
open:
io -4 -w 0xffa9003c 0x00006005
```

12.42 Infrared Remote Control

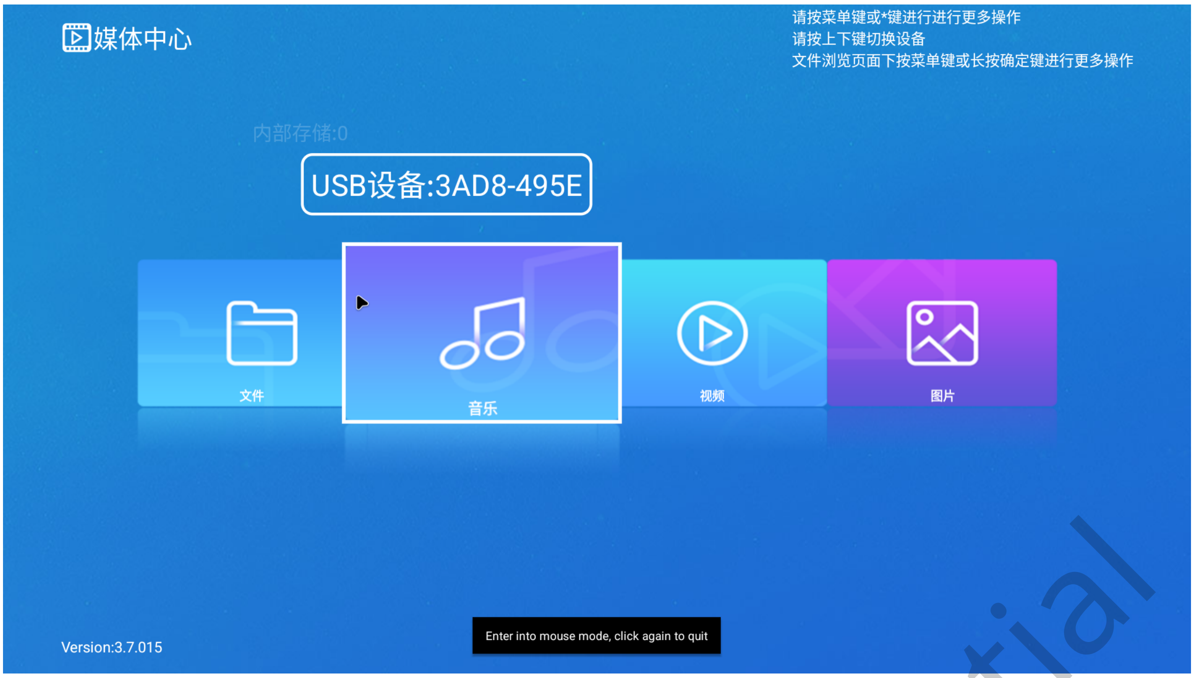
Virtual MouseThe virtual mouse function is mainly used in app interfaces that do not support remote control focus control. After enabling the virtual mouse function, the interface displays a virtual mouse icon, allowing users to directly use the directional keys and the confirmation key on the infrared remote control to move and click the virtual mouse icon, achieving a similar experience to using a physical mouse.

The SDK by default supports controlling the virtual mouse function with the specific key KEYCODE_PROFILE_SWITCH (288). If the remote control is not compatible, it can also be controlled via the property switch:

```
rk3576_projector:/ # setprop sys.KeyMouse.mKeyMouseState on
rk3576_projector:/ # setprop sys.KeyMouse.mKeyMouseState off
```

The following submissions correspond to the main functions; if the customer requires customization, they can refer to and modify them accordingly.

```
huangjc@171server:~/A14_MS/frameworks/base$ git show
5579c0a8c5bf9ec71c2092f2dadb94634ea9e9d4
commit 5579c0a8c5bf9ec71c2092f2dadb94634ea9e9d4
Author: Alex Wang <alex.wang@rock-chips.com>
Date:   Tue Oct 27 19:55:15 2020 +0800

    box:support for infrare simulate mouse.

    Change-Id: I48ffcbe3c85ca3f2b6713df3d79231403f090821
==================================================================
==================================================================
huangjc@171server:~/A14_MS/frameworks/native$ git show
4bdb2413f359db5ec16c4f5123b070090d274015
commit 4bdb2413f359db5ec16c4f5123b070090d274015 (tag: android-14.0-vehicle-rkr2)
Author: Alex Wang <alex.wang@rock-chips.com>
Date:   Tue Oct 27 19:52:12 2020 +0800

    box:support for infrare simulate mouse.

    Change-Id: I7224b873882de742b86b2dfda0dbaa298b34f2f7
```



请按菜单键或*键进行进行更多操作
请按上下键切换设备
文件浏览页面下按菜单键或长按确定键进行更多操作

Enter into mouse mode, click again to quit

12.43 How to configure RK3576 compilation for Box only

The default compilation configuration for RK3576, rk3576_projector, corresponds to the projector product configuration. Change it to the Box only configuration as per the following configuration and instructions:

```
huangjc@171server:~/A14_MS/device/rockchip/rk3576/rk3576_projector$ git diff
diff --git a/BoardConfig.mk b/BoardConfig.mk
index 7e55210..ab82c35 100644
--- a/BoardConfig.mk
+++ b/BoardConfig.mk
@@ -28,14 +28,14 @@ endif
 PRODUCT_KERNEL_DTS := rk3576-evb1-v10-projector
 PRODUCT_KERNEL_CONFIG += rk3576.config pcie_wifi.config
 BOARD_HS_ETHERNET := true
-BOARD_HDMI_IN_SUPPORT := true
+BOARD_HDMI_IN_SUPPORT := false

####support keystone
BOARD_SUPPORT_KEYSTONE := false
```

12.44 How RK3576 Supports Wake-Up from a Fake Shutdown via Infrared Remote Control

The RK3576 RKR3 version kernel update merges the new pinctlr framework and adjusts the pinctlr configuration for the RK806 driver. If you need to support waking up from a fake shutdown via the power button on the infrared remote control, the following modifications need to be added to the kernel:

```
diff --git a/drivers/regulator/rk806-regulator.c b/drivers/regulator/rk806-
regulator.c
index 2672bf91dd77..82979f63180f 100644
--- a/drivers/regulator/rk806-regulator.c
+++ b/drivers/regulator/rk806-regulator.c
@@ -1252,7 +1252,7 @@ static void rk806_regulator_shutdown(struct platform_device
*pdev)
                rk806_shutdown_requence_config(rk806);
                rk806_field_write(rk806, PWRCTRL1_FUN, PWRCTRL_NULL_FUN);
                rk806_field_write(rk806, PWRCTRL1_POL, POL_HIGH);
-               rk806_field_write(rk806, PWRCTRL1_FUN, PWRCTRL_POWOFF_FUN);
+               rk806_field_write(rk806, PWRCTRL1_FUN, PWRCTRL_SLP_FUN);
        }
 }
```

12.45 Show related problem cases and debug log collection

This section lists common set-top box display bug cases and log printing methods that need to be collected for reference, quickly judge the general problem points and provide effective analysis information to improve resolution efficiency:

Problem points	Main operation methods	Main phenomena
edid error	HDMI plug-in, switch-off restart	1) Partial edid is missing; 2) Inconsistent front and back;
Signal problems	Unplug HDMI, switch on and off	1) No signal will not be displayed; 2) Flash stripes/flash green screen;
Display exception	Switch resolution	1) Skim screen; 2) Various background colors (green, pink, red); 3) Picture tear; 4) UI color abnormality; 5) Focus error; 6) Local flashing of the picture; 7) Flashing stripes; 8) Not displayed;
Resolution switching problem	Resolution switching, setting high resolution, inserting a low resolution TV	1) Switching failed to maintain the last setting; 2) Not displayed; 3) The display does not match the setting value;
HDR problem	HDR film source + HDMI HDR mode	1) After unplugging HDMI, the HDR effect is unstable; 2) Exit playback without signal; 3) After unplugging HDMI, the screen is stuck; 4) Call out the player menu bar and move it; 5) Switch to SDR mode and not display; 6) The video screen display is abnormal (red, etc.);
Color Depth Issues	Switch Color Depth	UI Color Exception
clk problem	Switch resolution	1) Not displayed; 2) Display abnormalities such as splash screen, flower screen
vop problem	Unplug HDMI, power off and switch	1) Not displayed; 2) Playback stutter
summary layer problem	Unplug HDMI, play video to call out the menu bar, play video to rotate	1) Not displayed; 2) The video is stuck after the virtual menu key disappears; 3) No image display when the menu bar is called out; 4) No image display when the menu bar disappears; 5) The screen is splashed when the screen is rotated; 6) The video flashes green screen in the first frame;
hwc problem	Open settings and other applications	Splash screen in some locations
Overscan scan problem	Connection display	Incomplete display
Screen refresh problem	Switch resolution, multi-screen different display playback	1) Display half of the abnormality; 2) The different display playback is stuttering;
Sound output problem	Unplug, wake up standby, switch resolution, reset reset	1) No sound; 2) Sound output from the prototype;

Problem points	**Main operation methods**	**Main phenomena**
Settings	Resolution, screen zoom, etc.	1) Grayscale cannot be set; 2) If invalid and unresponsive after setting;
Problems with main and secondary screens	Power on and off, connection display	1) Display stretching; 2) Interface layout abnormality
VP problem	Two displays hanging on the same VP are connected at the same time, HDMI-0_8K+HDMI-1	System restart
Rotation Problem	Rotating Mockup	Display Direction 90°
Multi-screen problem	Multi-screen different display playback	1) Mouse sliding is stuck; 2) It is impossible to move on each screen; 3) Playback is stuck
Standby problem	Wake up and unplug HDMI	1) Display abnormality; 2) Not displayed; 3) The screen does not light up when the prototype side wakes up;
Stability issues	Video copying machine, HDMI switch copying machine	1) OOM causes the screen to be stuck or the Android restart; 2) The system crashes;

Problem points	Main operation methods	Main phenomena
Log information that needs to be crawled when there is a problem	adb shell cat /d/dri/0/summary > summary.log Query the hardware node of the VOP, that is, it is related to the screen display. If there is any display problem, this log is the most intuitive adb shell cat /d/clk/clk_summary > clk_summary.log Query the clock tree of the system, all hardware clocks will be printed out. If there is any problem with the clock, this log is also the most intuitive adb shell dumpsys SurfaceFlinger > sf.log adb shell vop2_dump.sh > vop2_dump.log adb shell dmesg > dmesg.log adb shell setprop vendor.hwc.log all ;logcat -c ;logcat > hwc.log	

13. APPENDIX A Compiling and development environment setup

13.1 Initializing a Build Environment

This section describes how to set up your local work environment to build the Android source files. You must use Linux or Mac OS; building under Windows is not currently supported.
For an overview of the entire code-review and code-update process, see Life of a Patch.
Note: All commands in this site are preceded by a dollar sign ($) to differentiate them from output or entries within files. You may use the Click to copy feature at the top right of each command box to copy all lines without the dollar signs or triple-click each line to copy it individually without the dollar sign.

13.2 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.

13.3 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.

13.4 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. The Ubuntu 15.04 OpenJDK 8 packages have been used successfully with Ubuntu 14.04. Newer package versions (e.g. those for 15.10, 16.04) were found not to work on 14.04 using the instructions below.

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 libc6-dev-i386 lib32ncurses5-dev x11proto-
core-dev libx11-dev lib32z-dev ccache libgl1-mesa-dev libxml2-utils xsltproc
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.

13.5 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

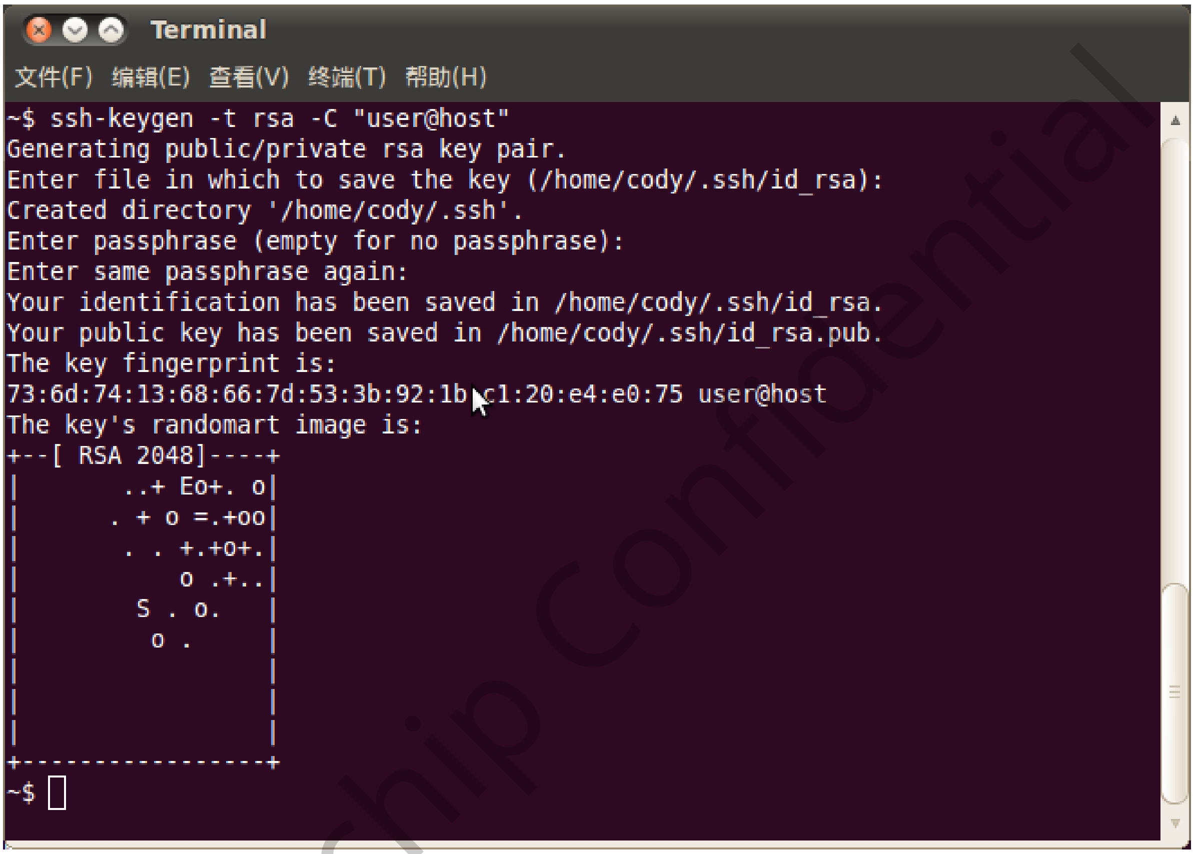
14. APPENDIX B SSH public key operation instruction

14.1 APPENDIX B-1 SSH public key generation

Use the following command to generate:

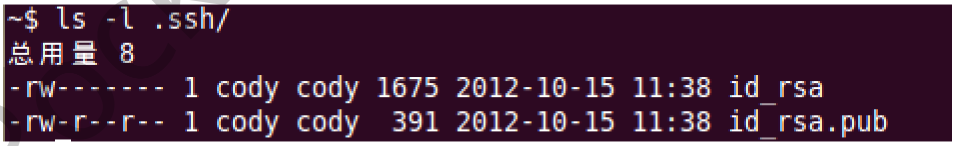
```
ssh-keygen -t rsa -C "user@host"
```

Please replace user@host with your email address.



```
~$ ssh-keygen -t rsa -C "user@host"
Generating public/private rsa key pair.
Enter file in which to save the key (/home/cody/.ssh/id_rsa):
Created directory '/home/cody/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/cody/.ssh/id_rsa.
Your public key has been saved in /home/cody/.ssh/id_rsa.pub.
The key fingerprint is:
73:6d:74:13:68:66:7d:53:3b:92:1b:c1:20:e4:e0:75 user@host
The key's randomart image is:
+--[ RSA 2048]----+
|       ..+ Eo+. o|
|      . + o =.+oo|
|       . . +.+O+.|
|           o .+..|
|        S . o.   |
|         o .     |
|                 |
|                 |
|                 |
+-----------------+
~$ 
```

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



```
~$ ls -l .ssh/
总用量 8
-rw------- 1 cody cody 1675 2012-10-15 11:38 id_rsa
-rw-r--r-- 1 cody cody  391 2012-10-15 11:38 id_rsa.pub
```

Please keep carefully the generated private key file id_rsa and password, and send id_rsa.pub to SDK release server admin through email.

14.2 APPENDIX B-2 Use key-chain to manage the key

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

1. Install keychain software package:

```
$sudo aptitude install keychain
```

2. Configure to use the key:

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

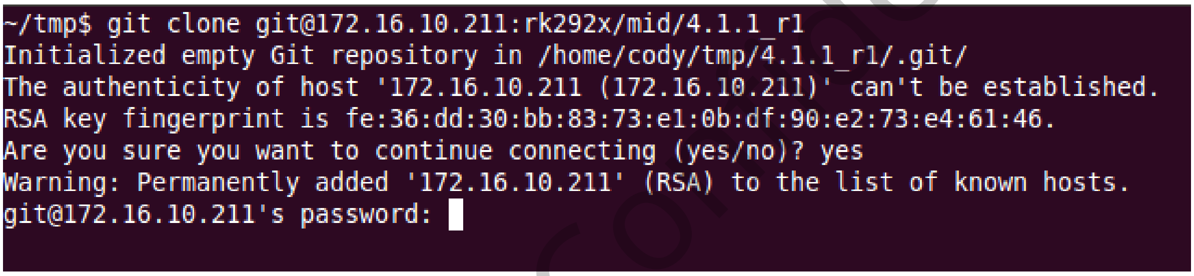
Add the following command:

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

Among which, id_rsa is the file name of the private key.
Log in 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.
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.

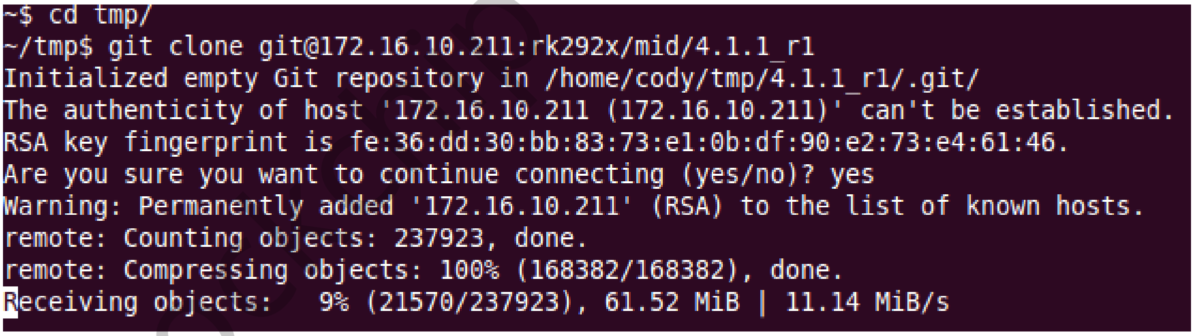
14.3 APPENDIX B-3 Multiple devices use the same ssh public key

In order to use on different devices, you can copy ssh private key file id_rsa to the target device "~/.ssh/id_rsa". Below hint will show up if using the wrong private key. Please replace with the correct private key.



```
~/tmp$ git clone git@172.16.10.211:rk292x/mid/4.1.1_r1
Initialized empty Git repository in /home/cody/tmp/4.1.1_r1/.git/
The authenticity of host '172.16.10.211 (172.16.10.211)' can't be established.
RSA key fingerprint is fe:36:dd:30:bb:83:73:e1:0b:df:90:e2:73:e4:61:46.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '172.16.10.211' (RSA) to the list of known hosts.
git@172.16.10.211's password:
```

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



```
~$ cd tmp/
~/tmp$ git clone git@172.16.10.211:rk292x/mid/4.1.1_r1
Initialized empty Git repository in /home/cody/tmp/4.1.1_r1/.git/
The authenticity of host '172.16.10.211 (172.16.10.211)' can't be established.
RSA key fingerprint is fe:36:dd:30:bb:83:73:e1:0b:df:90:e2:73:e4:61:46.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '172.16.10.211' (RSA) to the list of known hosts.
remote: Counting objects: 237923, done.
remote: Compressing objects: 100% (168382/168382), done.
Receiving objects:   9% (21570/237923), 61.52 MiB | 11.14 MiB/s
```

Below error may occur when adding ssh private key:

```
Agent admitted failture to sign using the key
```

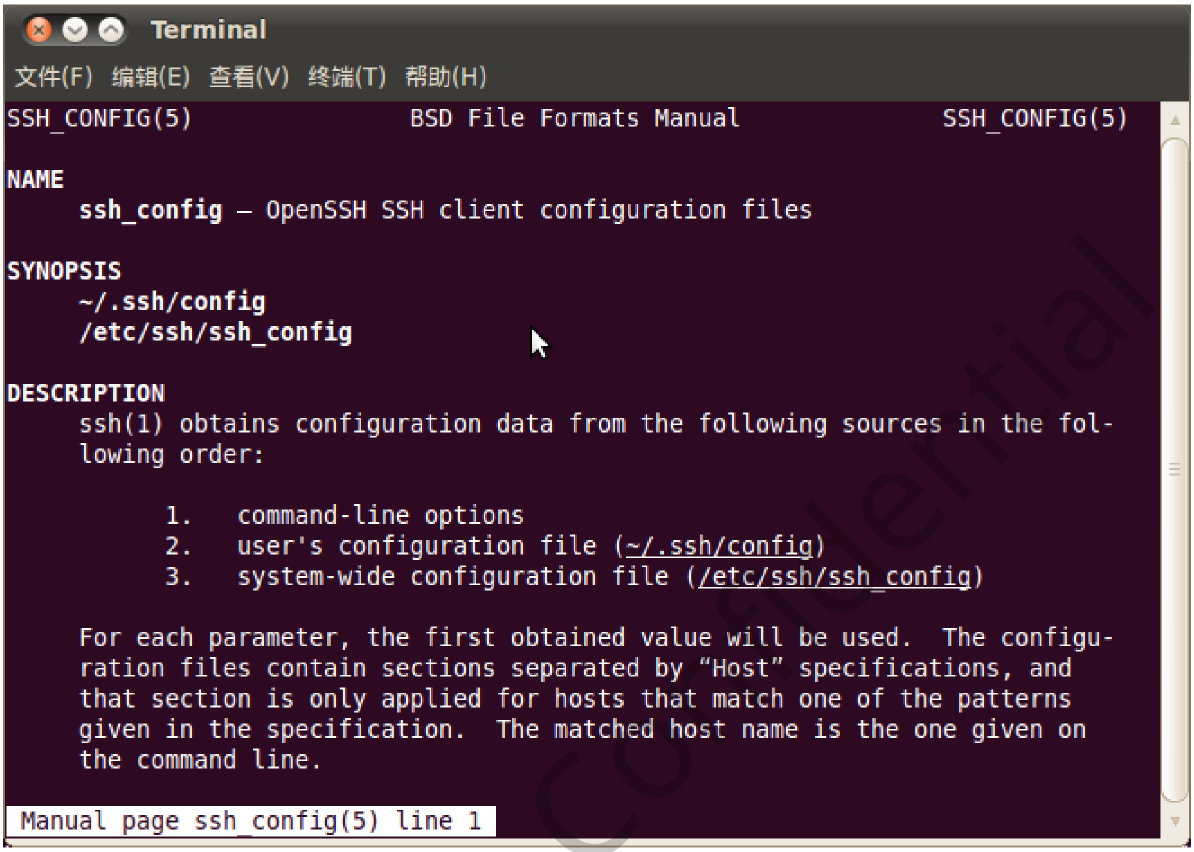
Input the following command at console can fix it.

```
ssh-add ~/.ssh/id_rsa
```

14.4 APPENDIX B-4 Switch different ssh public keys on one device

You can refer to ssh_config document to configure ssh.

```
~$ man ssh_config
```

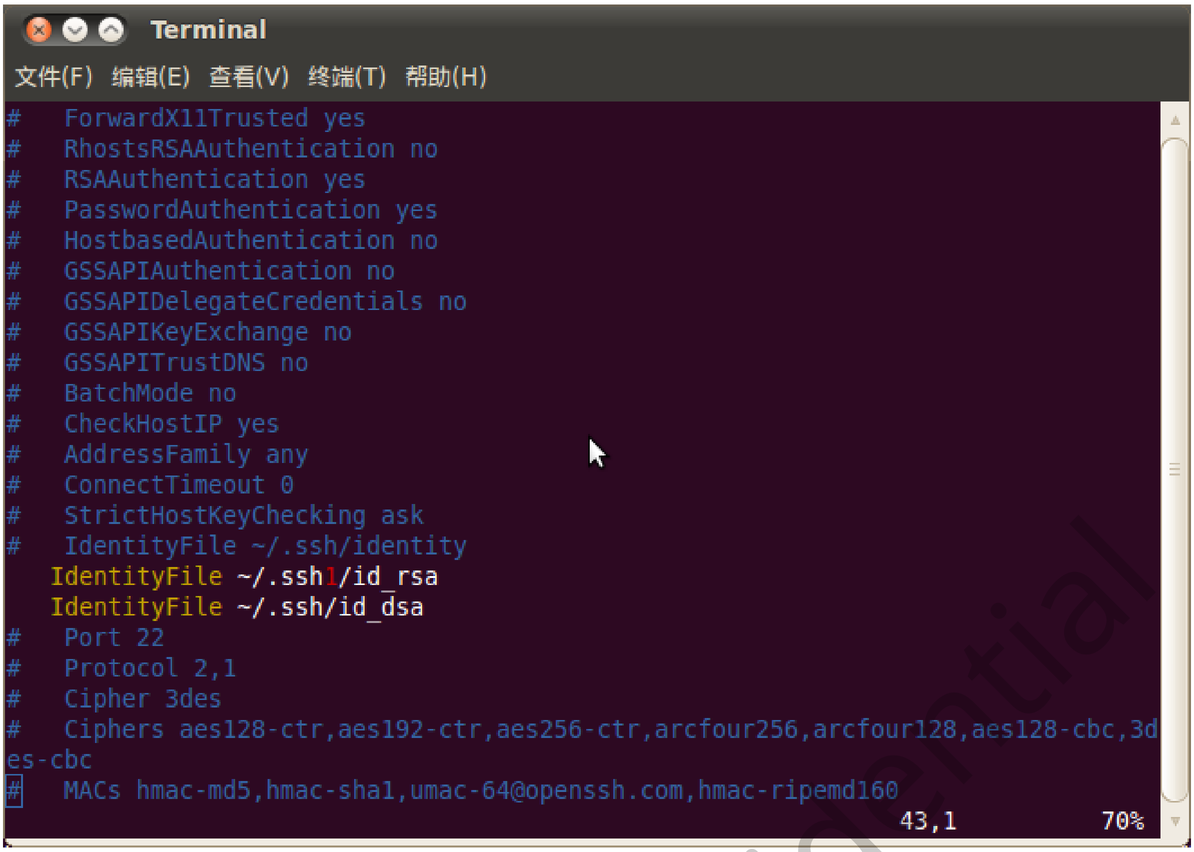


```
SSH_CONFIG(5)                BSD File Formats Manual               SSH_CONFIG(5)

NAME
     ssh_config — OpenSSH SSH client configuration files

SYNOPSIS
     ~/.ssh/config
     /etc/ssh/ssh_config

DESCRIPTION
     ssh(1) obtains configuration data from the following sources in the fol-
     lowing order:

           1.   command-line options
           2.   user's configuration file (~/.ssh/config)
           3.   system-wide configuration file (/etc/ssh/ssh_config)

     For each parameter, the first obtained value will be used.  The configu-
     ration files contain sections separated by “Host” specifications, and
     that section is only applied for hosts that match one of the patterns
     given in the specification.  The matched host name is the one given on
     the command line.

 Manual page ssh_config(5) line 1
```

Use the following commands to configure ssh for current user.

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

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



```
#   ForwardX11Trusted yes
#   RhostsRSAAuthentication no
#   RSAAuthentication yes
#   PasswordAuthentication yes
#   HostbasedAuthentication no
#   GSSAPIAuthentication no
#   GSSAPIDelegateCredentials no
#   GSSAPIKeyExchange no
#   GSSAPITrustDNS no
#   BatchMode no
#   CheckHostIP yes
#   AddressFamily any
#   ConnectTimeout 0
#   StrictHostKeyChecking ask
#   IdentityFile ~/.ssh/identity
   IdentityFile ~/.ssh1/id_rsa
   IdentityFile ~/.ssh/id_dsa
#   Port 22
#   Protocol 2,1
#   Cipher 3des
#   Ciphers aes128-ctr,aes192-ctr,aes256-ctr,arcfour256,arcfour128,aes128-cbc,3d
es-cbc
#   MACs hmac-md5,hmac-sha1,umac-64@openssh.com,hmac-ripemd160
                                                              43,1          70%
```

14.5 APPENDIX B-5 Key authority management

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

14.6 APPENDIX B-6 Git authority application instruction

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