Rockchip RK312X Linux SDK Release Note

ID: RK-FB-YF-376

Release Version: V1.2.0

Release Date: 2020-08-06

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

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

The document presents Rockchip RK312X Linux SDK release notes, aiming to help engineers get started with RK312X Linux SDK development and debugging faster.

Intended Audience

This document (this guide) is mainly intended for:

Technical support engineers

Software development engineers

Chipset and System Support

Chipset	Buildroot	Debian 9	Debian 10	Yocto
RK312X	Y	Y	N	N

Revision History

Date	Version	Author	Revision History
2019-04-29	BETA V0.1	Owen Chen	Initial version
2019-07-19	V1.0.0	Hans Yang	An official release version

| 2020-08-06 | V1.2.0 | Hans Yang | Rewrite the document with Markdown
Release V1.2.0 Version
1) add Debian compile introduction
2) add RK3126C/RK3128 config switch introduction |

Contents

1. Overview

This SDK is based on Buildroot 2018.02-rc3, Debian 9, base on kernel 4.4 and U-boot v2017.09. It is suitable for RK312X EVB development boards and all other Linux products developed based on it. This SDK supports VPU hardware decoding, GPU 3D, Wayland/X11 display, Qt and other function. For detailed functions debugging and interface introductions, please refer to the documents under the project's docs/ directory.

2. Main Functions

Functions	**Module Name**
Data Communication	Wi-Fi, Ethernet Card, USB, SDCARD
Applications	Multimedia playback, settings, browser, file management

3. How to Get the SDK

SDK is released by Rockchip server. Please refer to Chapter 7 SDK Building Introduciton to build a development environment.

To get RK312X Linux software package, customers need an account to access the source code repository provided by Rockchip. In order to be able to obtain code synchronization, please provide SSH public key for server authentication and authorization when apply for SDK from Rockchip technical window. About Rockchip server SSH public key authorization, please refer to Chapter 9 SSH Public Key Operation Introduction.

The command for downloading RK312X_Linux_SDK is as follows:

```
repo init --repo-url ssh://git@www.rockchip.com.cn/repo/rk/tools/repo -u
ssh://git@www.rockchip.com.cn/linux/rk/platform/manifests -b linux -m
rk312x_linux_release.xml
```

Repo, a tool built on Python script by Google to help manage git repositories, is mainly used to download and manage software repository of projects. The download address is as follows:

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

For quick access to SDK source code, Rockchip Technical Window usually provides corresponding version of SDK initial compression package. In this way, developers can get SDK source code through decompressing the initial compression package, which is the same as the one downloaded by repo. Take rk312x_linux_sdk_release_v1.2.0_20200806.tgz as an example. After getting an initialization package, you can get the source code by running the following command:

```
mkdir rk312x
tar xvf rk312x_linux_sdk_release_v1.2.0_20200806.tgz -C rk312x
cd rk312x
.repo/repo/repo sync -l
.repo/repo/repo sync
```

Developers can update via `.repo/repo/repo sync` command according to update instructions that are regularly released by FAE window.

4. Software Development Guide

4.1 Development Guide

The Kernel version of RK312X Linux SDK Kernel is Linux4.4, Rootfs is Buidlroot(2018.02-rc3) or Debian9 respectively. To help engineers quick start of SDK development and debugging, “Rockchip_Developer_Guide_Linux_Software_EN” is released with the SDK. It can be obtained in the docs/ directory and will be continuously updated.

4.2 Software Update History

Software release version upgrade can be checked through project xml file by the following command:

```
.repo/manifests$ ls -l -h rk312x_linux_release.xml
```

Software release version updated information can be found through the project text file by the following command:

```
.repo/manifests$ cat rk312x_linux_v1.00/RK312X_Linux_SDK_Release_Note.txt
```

Or refer to the project directory:

```
<SDK>/docs/Socs/RK312X/RK312X_Linux_SDK_Release_Note.txt
```

5. Hardware Development Guide

Please refer to user guides in the project directory for hardware development:

RK3126C hardware development guide:

```
<SDK>/docs/Socs/RK312X/Rockchip_RK3126C_Hardware_Design_Guide_V1.0_CN.pdf
```

RK3128 hardware development guide:

```
<SDK>/docs/Socs/RK312X/Rockchip_RK3128_Hardware_Design_Guide_V0.1_CN.pdf
```

6. SDK Project Directory Introduction

There are buildroot, debian, recovery, app, kernel, u-boot, device, docs, external and other directories in the project directory. Each directory or its sub-directories will correspond to a git project, and the commit should be done in the respective directory.

- app: store application apps like qcamera/qfm/qplayer/qseting and other applications.
- buildroot: root file system based on Buildroot (2018.02-rc3).
- debian: root file system based on Debian 9.
- device/rockchip: store board-level configuration for each chip and some scripts and prepared files for building and packaging firmware.
- docs: stores development guides, platform support lists, tool usage, Linux development guides, and so on.
- IMAGE: stores building time, XML, patch and firmware directory for each building.
- external: stores some third-party libraries, including audio, video, network, recovery and so on.
- kernel: stores kernel4.4 development code.
- prebuilts: stores cross-compilation toolchain.
- rkbin: stores Rockchip Binary and tools.
- rockdev: stores building output firmware.
- tools: stores some commonly used tools under Linux and Windows system.
- u-boot: store U-Boot code developed based on v2017.09 version.

7. SDK Building Introduction

Ubuntu 16.04 system: Please install software packages with below commands to setup Buildroot building environment:

```
sudo apt-get install repo git-core gitk git-gui gcc-arm-linux-gnueabihf u-boot-tools
device-tree-compiler gcc-aarch64-linux-gnu mtools parted libudev-dev libusb-1.0-0-dev
python-linaro-image-tools linaro-image-tools autoconf autotools-dev libsigsegv2 m4 intltool
libdrm-dev curl sed make binutils build-essential gcc g++ bash patch gzip bzip2 perl tar
cpio python unzip rsync file bc wget libncurses5 libqt4-dev libglib2.0-dev libgtk2.0-dev
libglade2-dev cvs git mercurial rsync openssh-client subversion asciidoc w3m dblatex
graphviz python-matplotlib libc6:i386 libssl-dev texinfo liblz4-tool genext2fs expect
patchelf xutils-dev
```

Please install software packages with below commands to setup Debian building environment:

```
sudo apt-get install repo git-core gitk git-gui gcc-arm-linux-gnueabihf u-boot-tools
device-tree-compiler gcc-aarch64-linux-gnu mtools parted libudev-dev libusb-1.0-0-dev
python-linaro-image-tools linaro-image-tools gcc-arm-linux-gnueabihf libssl-dev gcc-
aarch64-linux-gnu g+conf autotools-dev libsigsegv2 m4 intltool libdrm-dev curl sed make
binutils build-essential gcc g++ bash patch gzip bzip2 perl tar cpio python unzip rsync
file bc wget libncurses5 libqt4-dev libglib2.0-dev libgtk2.0-dev libglade2-dev cvs git
mercurial rsync openssh-client subversion asciidoc w3m dblatex graphviz python-matplotlib
libc6:i386 libssl-dev texinfo liblz4-tool genext2fs xutils-dev
```

Ubuntu 17.04 or later version system: in addition to the above, the following dependencies is needed:

```
sudo apt-get install lib32gcc-7-dev g++-7 libstdc++-7-dev
```

It is recommended to use Ubuntu 18.04 system or higher version for development. If you encounter an error during building, you can install the corresponding software packages according to the error message.

7.1 Choose Develop Platform

RK312X Linux SDK support RK3126C and RK3128 platform，The board-level configuration file is located in the device/rockchip/ directory:

Platform	BoardConfig.mk	Rootfs	U-Boot Config	Kernel Config	Kernel DTS
RK3126C	rk3126c/BoardConfig.mk	Buildroot	rk3126_defconfig	rockchip_linux_defconfig	rk3126-linux.dts
RK3128	rk3128/BoardConfig.mk	Buildroot	rk3128_defconfig	rockchip_linux_defconfig	rk3128-fireprime.dts
RK3128	rk3128/BoardConfig_debian.mk	Debian	rk3128_defconfig	rockchip_linux_defconfig	rk3128-fireprime.dts

Mainly including Uboot config, Kernel Config, DTS, Buildroot Config. Customers can configure the switch based on the actual situation of the project.

1. Please use the following command to build Buildroot system on RK3126C platform:

```
$./build.sh device/rockchip/rk3126c/BoardConfig.mk
```

2. Please use the following command to build Buildroot system on RK3128 platform:

```
$./build.sh device/rockchip/rk3128/BoardConfig.mk
```

3. Please use the following command to build Debian system on RK3128 platform:

```
$./build.sh device/rockchip/rk3128/BoardConfig_debian.mk
```

7.2 U-boot Building

Enter project u-boot directory and run `make.sh` to get rk312x_loader_v2.12.256.bin trust.img and uboot.img.

RK3126C EVB:

```
./make.sh rk3126
```

RK3128 EVB:

```
./make.sh rk3128
```

The built files are in u-boot directory:

```
u-boot/
├── rk312x_loader_v2.12.256.bin
├── trust.img
└── uboot.img
```

7.3 Kernel Building Steps

Enter project root directory and run the following command to automatically build and package kernel:

RK3126C EVB:

```
cd kernel
make ARCH=arm rockchip_linux_defconfig
make ARCH=arm rk3126-linux.img -j12
```

RK3128 EVB:

```
cd kernel
make ARCH=arm rockchip_linux_defconfig
make ARCH=arm rk3128-fireprime.img -j12
```

The boot.img includes image and DTB of kernel will be generated after building in the kernel directory.

7.4 Recovery Building Steps

Enter project root directory and execute the following command to automatically complete building and packaging of Recovery

```
./build.sh recovery
```

The recovery.img will be generated in Buildroot directory “output/rockchip_rk312x_recovery/images” after building.

Please pay attention to that recovery.img contains kernel.img, so every time the kernel changes, recovery needs to be repackaged and generated. For example:

```
SDK$source envsetup.sh rockchip_rk312x
SDK$make recovery-rebuild
SDK$./build.sh recovery
```

7.5 Buildroot Building

7.5.1 Buildroot Rootfs Building

Enter project root directory and execute the following commands to automatically complete building and packaging of Rootfs.

```
 ./build.sh rootfs
```

Note:

If you need to build a single module or a third-party application, you need to configure the cross-compilation environment. Cross-compilation tool is located in “buildroot/output/rockchip_rk312x/host/usr” directory. You need to set the “bin/” directory of tools and “aarch64-buildroot-linux-gnu/bin/” directory to environment variables, and execute auto-configuration environment variable script in the top-level directory (only valid for current console):

```
source envsetup.sh rockchip_rk312x
```

Enter the command to view:

```
aarch64-linux-gcc --version
```

When the following log is printed, configuration is successful:

```
arm-buildroot-linux-gnueabihf-gcc.br_real (Buildroot 2018.02-rc3-01112-g829f85a-dirty)6.5.0
```

7.5.2 Build Modules in Buildroot

Take the frequently used building commands for qplayer module for example:

- Set the environment

```
SDK$source envsetup.sh rockchip_rk312x
```

- Build qplayer

```
SDK$make qplayer
```

- Rebuild qplayer

```
SDK$make qplayer-rebuild
```

- Delete qplayer

```
SDK$make qplayer-dirclean
or
SDK$rm -rf /buildroot/output/rockchip_rk312x/build/qlayer-1.0
```

7.6 Debian 9 Building

```
 ./build.sh debian
```

Or enter debian/ directory:

```
cd debian/
```

Please refer to the readme.md in the directory for later building and Debian firmware generation.

(1) Building base Debian system

```
sudo apt-get install binfmt-support qemu-user-static live-build
sudo dpkg -i ubuntu-build-service/packages/*
sudo apt-get install -f
```

Build 32 bit Debian:

```
RELEASE=stretch TARGET=desktop ARCH=armhf ./mk-base-debian.sh
```

After building, linaro-stretch-alip-xxxxx-1.tar.gz (xxxxx is timestamp generated) will be generated in "debian/":

FAQ:

- If you encounter the following problem during above building:

```
noexec or nodev issue /usr/share/debootstrap/functions: line 1450:
..../rootfs/ubuntu-build-service/stretch-desktop-armhf/chroot/test-dev-null: Permission
denied E: Cannot install into target
'/home/foxluo/work3/rockchip/rk_linux/rk312x_linux/rootfs/ubuntu-build-service/stretch-
desktop-armhf/chroot' mounted with noexec or nodev
```

Solution：

```
mount -o remount,exec,dev xxx (xxx is the project directory), and then rebuild
```

In addition, if there are other building issues, please check firstly that the building system is not ext2/ext4.

- Because building Base Debian requires to access to foreign websites, and when domestic networks access foreign websites, download failures often occur:

The live build is used in Debian9, you can configure like below to change the image source to domestic:

```
+++ b/ubuntu-build-service/stretch-desktop-arm64/configure
@@ -11,6 +11,11 @@ set -e
 echo "I: create configuration"
 export LB_BOOTSTRAP_INCLUDE="apt-transport-https gnupg"
 lb config \
+ --mirror-bootstrap "http://mirrors.163.com/debian" \
+ --mirror-chroot "http://mirrors.163.com/debian" \
+ --mirror-chroot-security "http://mirrors.163.com/debian-security" \
+ --mirror-binary "http://mirrors.163.com/debian" \
+ --mirror-binary-security "http://mirrors.163.com/debian-security" \
  --apt-indices false \
  --apt-recommends false \
  --apt-secure false \
```

If the package cannot be downloaded for other network reasons, there are pre-build packages shared on Baidu Cloud Disk, put it in the current directory, and do the next step.

(2) Building rk-debian rootfs

Build 32 bit Debian:

```
VERSION=debug ARCH=arm64 ./mk-rootfs-stretch.sh
```

(3) Creating the ext4 image(linaro-rootfs.img)

```
./mk-image.sh
```

The linaro-rootfs.img will be generated.

7.7 Full Automatic Building

After building various parts of Kernel/U-Boot/Rootfs above, enter root directory of project directory and execute the following commands to automatically complete all building:

```
$./build.sh all
```

Detailed parameters usage, you can use help to search, for example:

```
rk312x$ ./build.sh --help
Usage: build.sh [OPTIONS]
Available options:
BoardConfig*.mk    -switch to specified board config
uboot              -build uboot
spl                -build spl
kernel             -build kernel
modules            -build kernel modules
toolchain          -build toolchain
rootfs             -build default rootfs, currently build buildroot as default
buildroot          -build buildroot rootfs
```

```
ramboot             -build ramboot image
multi-npu_boot      -build boot image for multi-npu board
yocto               -build yocto rootfs
debian              -build debian9 stretch rootfs
distro              -build debian10 buster rootfs
pcba                -build pcba
recovery            -build recovery
all                 -build uboot, kernel, rootfs, recovery image
cleanall            -clean uboot, kernel, rootfs, recovery
firmware            -pack all the image we need to boot up system
updateimg           -pack update image
otapackage          -pack ab update otapackage image
save                -save images, patches, commands used to debug
allsave             -build all & firmware & updateimg & save

Default option is 'allsave'.
```

Board level configurations of each board need to be configured in /device/rockchip/rk3126c/Boardconfig.mk or /device/rockchip/rk3128/Boardconfig.mk.

Main configurations of RK3126C board are as follows:

```
# Target arch
export RK_ARCH=arm
# Uboot defconfig
export RK_UBOOT_DEFCONFIG=rk3126
# Trust ini config
export RK_TRUST_INI_CONFIG=RK3126TOS_LADDR.ini
# Uboot size
export RK_UBOOT_SIZE_CONFIG=1024\ 2
# Trust size
export RK_TRUST_SIZE_CONFIG=1024\ 2
# Kernel defconfig
export RK_KERNEL_DEFCONFIG=rockchip_linux_defconfig
# Kernel dts
export RK_KERNEL_DTS=rk3126-linux
# boot image type
export RK_BOOT_IMG=zboot.img
# kernel image path
export RK_KERNEL_IMG=kernel/arch/arm/boot/zImage
# parameter for GPT table
export RK_PARAMETER=parameter-buildroot.txt
# Buildroot config
export RK_CFG_BUILDROOT=rockchip_rk312x
# Recovery config
export RK_CFG_RECOVERY=rockchip_rk312x_recovery
```

7.8 Firmware Package

After building various parts of Kernel/Uboot/Recovery/Rootfs above, enter root directory of project directory and execute the following command to automatically complete all firmware packaged into rockdev directory:

Firmware Generation:

```
./mkfirmware.sh
```

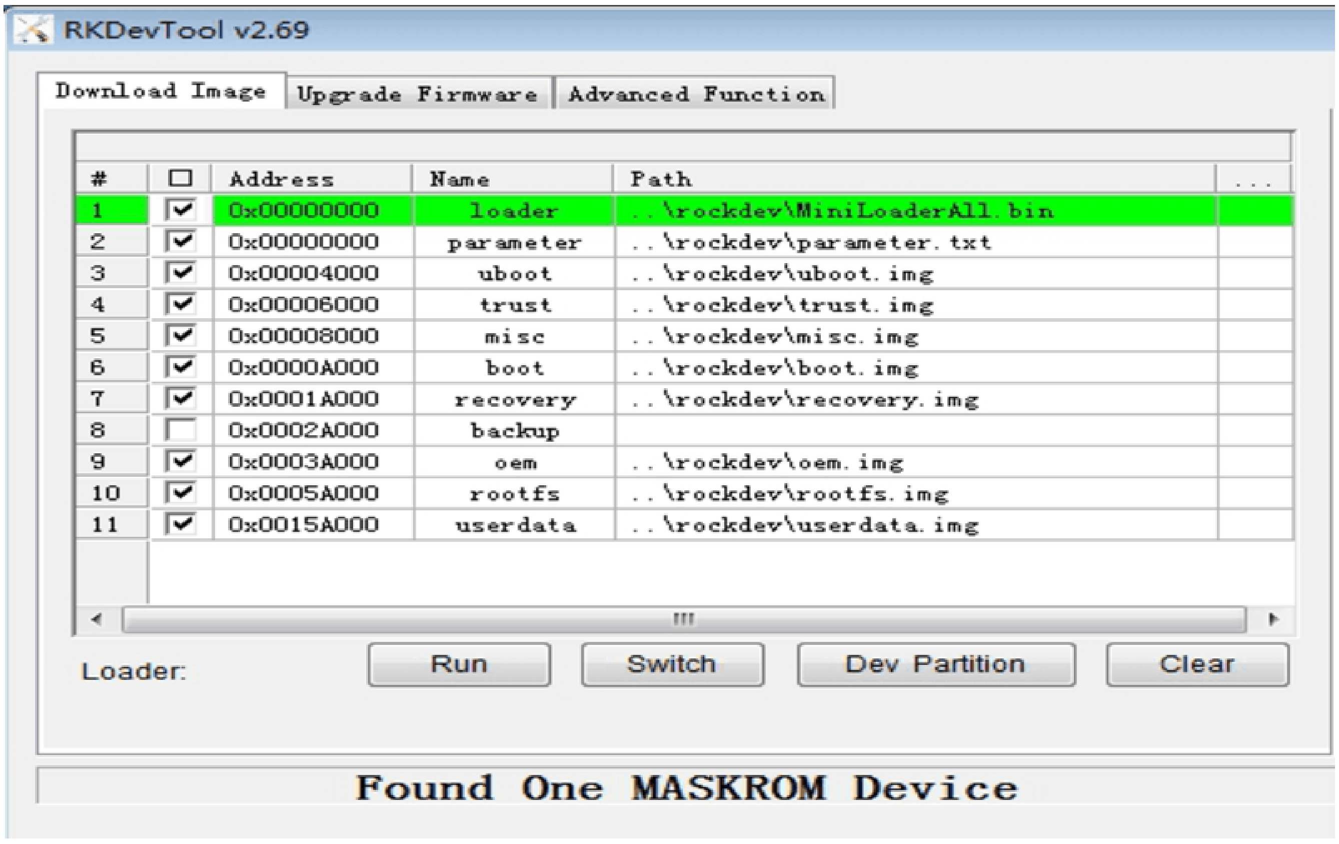
8. Upgrade Introduciton

8.1 Windows Upgrade Introduction

SDK provides windows upgrade tool (this tool should be V2.55 or later version) which is located in project root directory:

```
tools/
├── windows/RKDevTool
```

As shown below, after building and generating the corresponding firmware, device needs to enter MASKROM mode for upgrade. After connecting USB cable, long press the button “MASKROM” and press reset button “RST” at the same time and then release, device will enter MASKROM mode. Then you should load the paths of the corresponding images and click “Run” to start update. You can also press the “recovery” button and press reset button "RST" then release to enter loader mode to update. Partition offset and update files of MASKROM Mode are shown as follows (Note: you have to run the tool as an administrator in Windows PC):



#	☐	Address	Name	Path	...
1	✓	0x00000000	loader	..\rockdev\MiniLoaderAll.bin	
2	✓	0x00000000	parameter	..\rockdev\parameter.txt	
3	✓	0x00004000	uboot	..\rockdev\uboot.img	
4	✓	0x00006000	trust	..\rockdev\trust.img	
5	✓	0x00008000	misc	..\rockdev\misc.img	
6	✓	0x0000A000	boot	..\rockdev\boot.img	
7	✓	0x0001A000	recovery	..\rockdev\recovery.img	
8	☐	0x0002A000	backup		
9	✓	0x0003A000	oem	..\rockdev\oem.img	
10	✓	0x0005A000	rootfs	..\rockdev\rootfs.img	
11	✓	0x0015A000	userdata	..\rockdev\userdata.img	

Found One MASKROM Device

Note: before upgrade, please install the latest USB driver, which is in the below directory:

```
<SDK>/tools/windows/DriverAssitant_v4.8.zip
```

8.2 Linux Upgrade Introduction

The Linux upgrade tool (Linux_Upgrade_Tool should be v1.33 or later versions) is located in “tools/linux” directory. Please make sure your board is connected to MASKROM/loader rockusb, if the generated firmware is in rockdev directory, upgrade commands are as below:

```
sudo ./upgrade_tool ul rockdev/MiniLoaderAll.bin
sudo ./upgrade_tool di -p rockdev/parameter.txt
sudo ./upgrade_tool di -u rockdev/uboot.img
sudo ./upgrade_tool di -t rockdev/trust.img
sudo ./upgrade_tool di -misc rockdev/misc.img
sudo ./upgrade_tool di -b rockdev/boot.img
sudo ./upgrade_tool di -recovery rockdev/recovery.img
sudo ./upgrade_tool di -oem rockdev/oem.img
sudo ./upgrade_tool di -rootfs rocdev/rootfs.img
sudo ./upgrade_tool di -userdata rockdev/userdata.img
sudo ./upgrade_tool rd
```

Or upgrade the whole firmware after packaging:

```
sudo ./upgrade_tool uf rockdev/update.img
```

Or in root directory, run the following command on your device to upgrade in MASKROM state

```
./rkflash.sh
```

8.3 System Partition Introduction

Default partition introduction (below is RK3126C/RK3128 reference partition):

Number	Offset (sector)	Offset	Size	Name
1	0x2000	0x400000	2MB	uboot
2	0x3000	0x600000	2MB	trust
3	0x4000	0x800000	1MB	misc
4	0x4800	0x900000	8MB	boot
5	0x8800	0x1100000	14MB	recovery
7	0xF800	0x1F00000	20MB	oem
8	0x19800	0x3300000	65MB	rootfs
9	0x3A000	0x7400000	-	userdata

- uboot partition: for uboot.img built from uboot.

- trust partition: for trust.img built from uboot.
- misc partition: for misc.img built from recovery.
- boot partition: for boot.img built from kernel.
- recovery partition: for recovery.img built from recovery.
- oem partition: used by manufactor to store their APP or data, mounted in /oem directory
- rootfs partition: store rootfs.img built from buildroot or debian.
- userdata partition: store files temporarily generated by APP or for users, mounted in /userdata directory

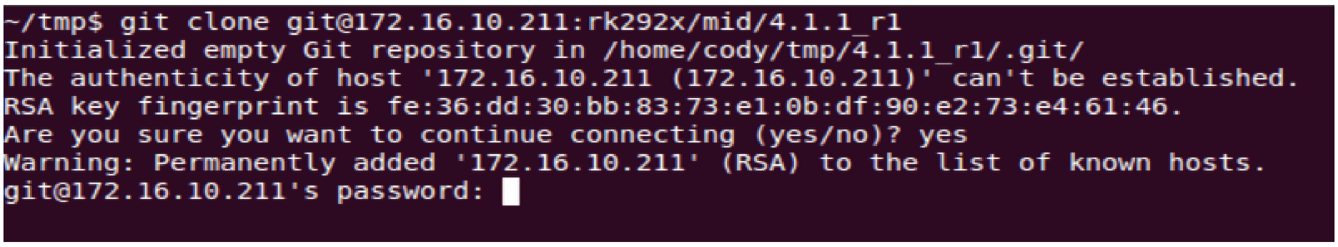
9. SSH Public Key Operation Introduction

Please follow the introduction in the “Rockchip SDK Application and Synchronization Guide” to generate an SSH public key and send the email to fae@rock-chips.com, applying for permission to download SDK code. This document will be released to customers during the process of applying for permission.

9.1 Multi-device Use the Same SSH Public Key

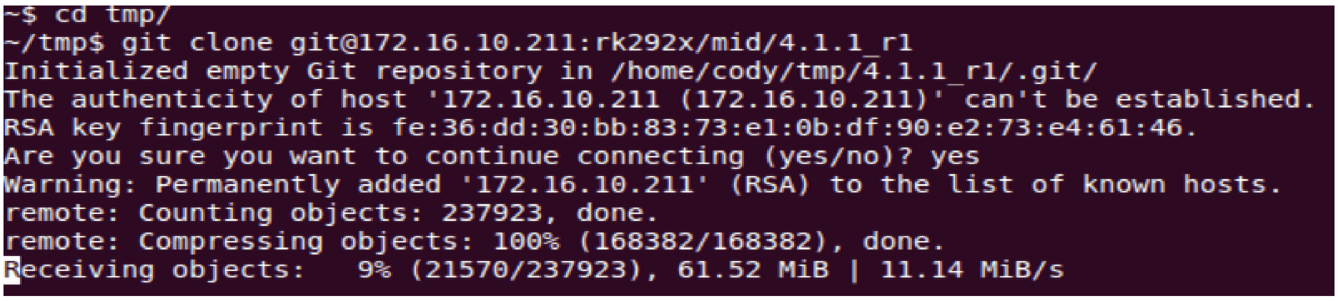
If the same SSH public key should be used in different devices, you can copy the SSH private key file id_rsa to “~/.ssh/id_rsa” of the device you want to use.

If the following prompt appears when using a wrong private key, please be careful to replace it 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, as shown below.



```
~$ 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
```

Adding SSH private key may result in the following error.

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

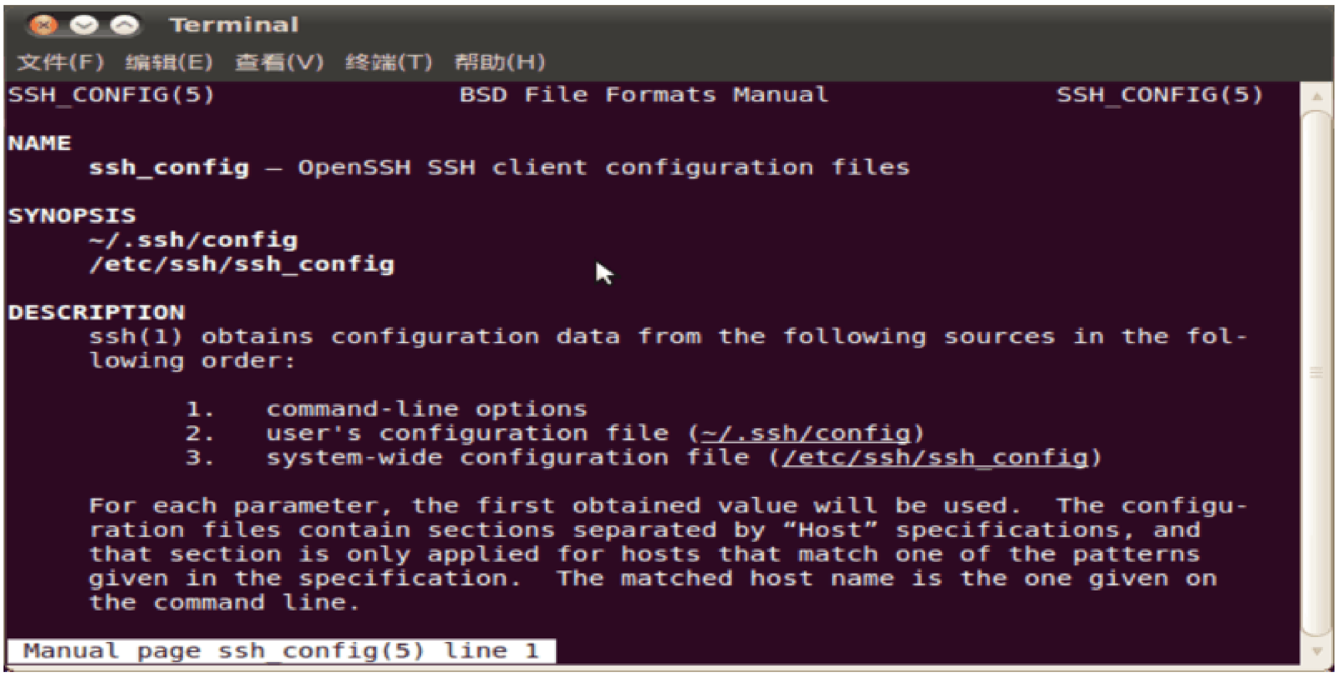
Enter the following command in console to solve:

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

9.2 Switch Different SSH Public Keys on the Same Device

You can configure SSH according to the ssh_config documentation.

```
~$ man ssh_config
```

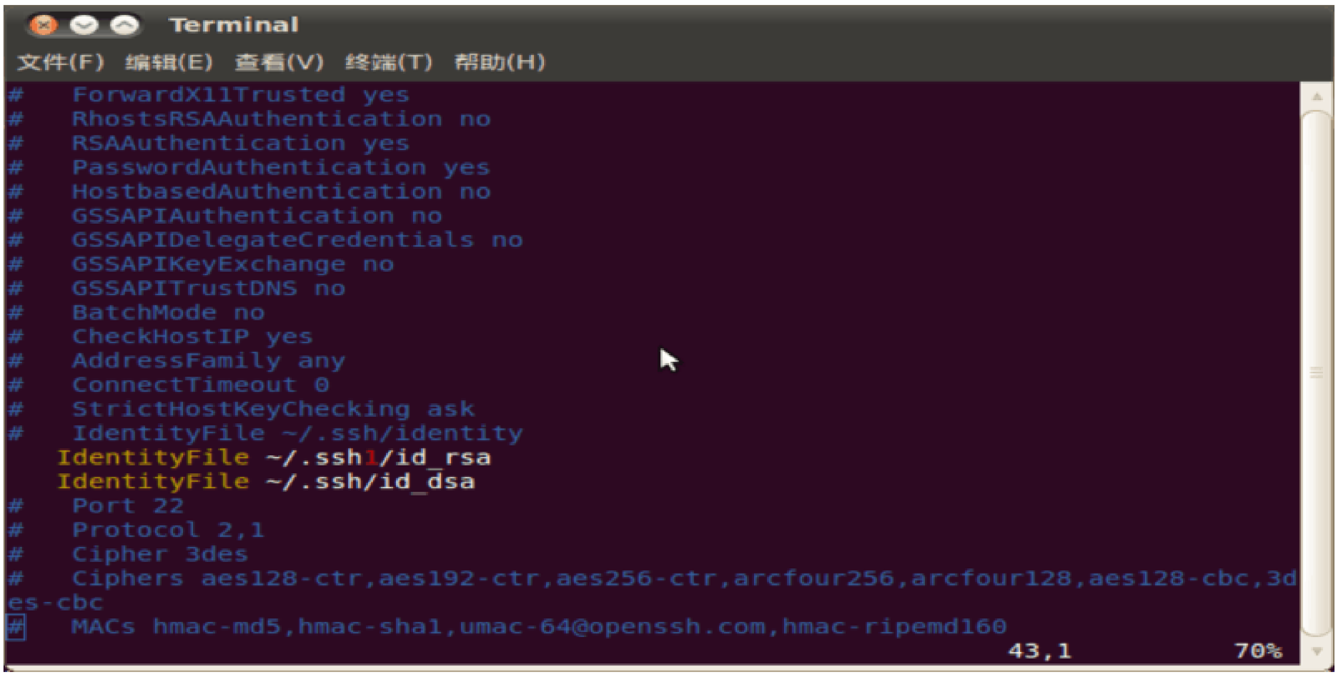


```
SSH_CONFIG(5)                BSD File Formats Manual               SSH_CONFIG(5)

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
```

Run the following command to configure SSH configuration of current user.

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

As shown in the figure, SSH uses the file “~/.ssh1/id_rsa” of another directory as an authentication private key. In this way, different keys can be switched.



```
#   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%
```

9.3 Key Authority Management

Server can monitor download times and IP information of a key in real time. If an abnormality is found, download permission of the corresponding key will be disabled.

Keep the private key file properly. Do not grant second authorization to third parties.

9.4 Reference Documents

For more details, please refer to document “sdk/docs/RKTools manuals/Rockchip SDK Kit Application GuideV1.6-201905.pdf”.