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

RK1808 Linux SDK Release Note

（Technical Department, Dept.III）

Status： [ ]draft [ ] Modifying [√] Released	**Document ID：**	RK-FB-CS-007
	Version：	1.1.0
	Author：	HL
	Date：	2019-08-19
	Reviewer：	ZYY
	Date：	2019-08-19

福州瑞芯微电子股份有限公司
Fuzhou Rockchip Electronics Co.,Ltd.

Revision History

Version No.	Author	Revision Date	Revision Description	Remark
V1.0.0	HL	2018-12-27	Initial Release	
V1.1.0	HL	2019-8-8	Update firmware download address and NPU document path	

Table of content

Warranty Disclaimer

This document is provided by “status" and Fuzhou Rockchip Electronics Co., Ltd ("our company ", the same below) does not provide any express or implied statement or warranty of any accuracy, reliability, integrity, merchantability, specific purpose and non-infringement of any statement, information and content of this document. This document is intended as a guide only for use.

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

Trademarks

Rockchip and “瑞芯微”、“瑞芯” are trademarks of Fuzhou Rockchip Electronics Co., Ltd. and are exclusively owned by Fuzhou Rockchip Electronics Co., Ltd.

References to other companies and their products use trademarks owned by the respective companies and are for reference purpose only.

Copyright © 2018 Fuzhou Rockchip Electronics Co., Ltd.

Fuzhou Rockchip Electronics Co., Ltd
Address: No.18 Building, A District, No.89 Software Boulevard, FuZhou, FuJian, PRC
Website: www.rock-chips.com
Customer service Tel.： +86-4007-700-590
Customer service Fax： +86-591-83951833
Customer service e-Mail: fae@rock-chips.com

Chapter 1 Overview

This SDK is based on Buildroot-2018.02 with its kernel based on kernel 4.4 and contains system source code, drivers, tools and application packages for Linux system development. The SDK also includes NPU development related tools, including rknn_demo (InceptionV2 SSD), rknn-toolkit and related development documents, adapted to RK1808 chip platform and is suitable for RK1808 EVB board and its product development.

Chapter 2 Main Functions

Functions	Module Names
Data Communication	Wi-Fi、SDCARD, Ethernet Card, USB
NPU related tools	rknn-toolkit，RKNPUTools

Chapter 3 How to Acquire SDK

SDK is released by Rockchip server or download from Github. Please refer to Chapter 5 SDK Compile Instruction to Build development environment.

How to get SDK from Rockchip server:
To get RK1808 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 Instruction.
RK1808_LINUX_SDK download address 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 rk1808_linux_release.xml
```

Repo is a script that google uses Python script to call git. It 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 customers quickly access SDK source code, Rockchip Technology Window usually provides corresponding version of SDK initial compression package. In this way, developers can obtain SDK source code from decompression the initial compression package, which is the same as the one downloaded by repo.
Take rk1808_linux_v1.1.0_20190808.tgz as an example. After copying initialization package, you can get source code by the following command:

```
mkdir rk1808
tar xvf rk1808_linux_v1.1.0_20190808.tgz -C rk1808
cd rk1808
.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.

How to get SDK from Github:
Download repo tool:

```
git clone https://github.com/rockchip-linux/repo.git
```

To get source code：

```
mkdir rk1808_linux
cd rk1808_linux
../repo/repo init --repo-url=https://github.com/rockchip-
linux/repo -u https://github.com/rockchip-linux/manifests  -b
master -m rk1808_linux_release.xml
../repo/repo sync
```

Chapter 4 RK1808 Linux Project Directory Introduction

There are buildroot、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 submission should be done in the respective directory.
1) buildroot: customize root file system;
2) app: store application app, mainly some test applications;
3) external: related libraries, including audio, video and so on;
4) kernel: kernel source code.
5) device/rockchip/rk1808: Store some scripts and prepared files for compiling and packaging firmware.
6) docs: store project help files.
7) prebuilts: store cross-compilation toolchain.
8) rkbin: store firmware and tools.
9) rockdev: store compiled output firmware
10) tools: store some common tools.
11) u-boot: uboot code

Chapter 5 SDK Compiling Instruction

Ubuntu 16.04 system:
Please install software packages with below commands to setup Buildroot compiling 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
```

Ubuntu 17.04 system:
In addition to the above, the following dependencies is needed:

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

5.1 Uboot Compiling

Enter project u-boot directory and execute make.sh to get rk1808_loader_v1.03.104.bin trust.img uboot.img:
RK1808 evb board：

```
cd u-boot
./make.sh rk1808
```

The compiled file is in u-boot directory:

```
u-boot/
├── rk1808_loader_v1.01.101.bin
```



```
├── trust.img
└── uboot.img
```

5.2 Kernel Compiling Steps

Enter project root directory and execute the following command to automatically compile and package kernel:
RK1808 evb board：

```
cd kernel
make rk1808_linux_defconfig
make rk1808-evb-v10.img -j12
```

After compiling, the boot.img is generated in kernel directory, including image and dtb of kernel.

5.3 Recovery Compiling Steps

Enter project root directory and execute the following command to automatically complete compiling and packaging of Recovery.
RK1808 evb board：

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

The recovery.img is generated in Buildroot directory /output/rockchip_rk1808_recovery/images after compiling.

5.4 Rootfs System Compiling

Enter project root directory and execute the following commands to automatically complete compiling and packaging of Rootfs.
RK1808 evb board：

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

After compiling, the rootfs.ext4 is generated in Buildroot directory /output/rockchip_rk1808/images.

Note：
If you need to compile a single module or a third-party application, you need to configure cross-compilation environment.
Cross-compilation tool is located in buildroot/output/rockchip_rk1808/host/usr directory. You need to set bin/ directory of tools and aarch64-rockchip-linux-gnueabihf/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
```

Enter the command to view:

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

When the following log printed, configuration is successful:

```
aarch64-linux-gcc.br_real (Buildroot 2018.02-rc3-05646-g17bb6ab)
6.4.0
```

5.5 Fully Automatic Compiling

Compile various parts of Kernel/Uboot/Recovery/Rootfs above, enter root directory of project directory and execute the following commands to automatically complete all compilation:

```
./build.sh
```

Detailed parameter usage, you can help to search, such as
Rk1808$ ./build.sh --help
Can't found build config, please check again

```
Rk1808$ ./build.sh --help
Can't found build config, please check again
```



```
====USAGE: build.sh modules====
uboot       -build uboot
kernel      -build kernel
rootfs      -build default rootfs, currently build buildroot as
default
buildroot   -build buildroot rootfs
yocto       -build yocto rootfs, currently build ros as default
ros         -build ros rootfs
debian      -build debian rootfs
pcba        -build pcba
all         -build uboot, kernel, rootfs, recovery image
default     -build all modules
```

5.6 Firmware Packaging Steps

After compiling 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: `./mkfirmware.sh`

Chapter 6 Upgrade Instruction

6.1 Windows Upgrade Instruction

SDK provides windows upgrade tools (tools need V2.61 or later version) which are located in project root directory:

```
tools/
├── windows/AndroidTool
```

As shown below, after compiling the corresponding firmware, device needs to enter MASKROM mode for update. After connecting usb cable, long press the button "MSROM" and press reset button "RST" and release "Maskrom" button about 2 seconds later, device will enter MASKROM Mode. Then you should load the paths of the corresponding images and click "Run" to start update. Partition offset and update files of MASKROM Mode are shown as follows (Note: Window PC needs to run tools as an administrator to execute):

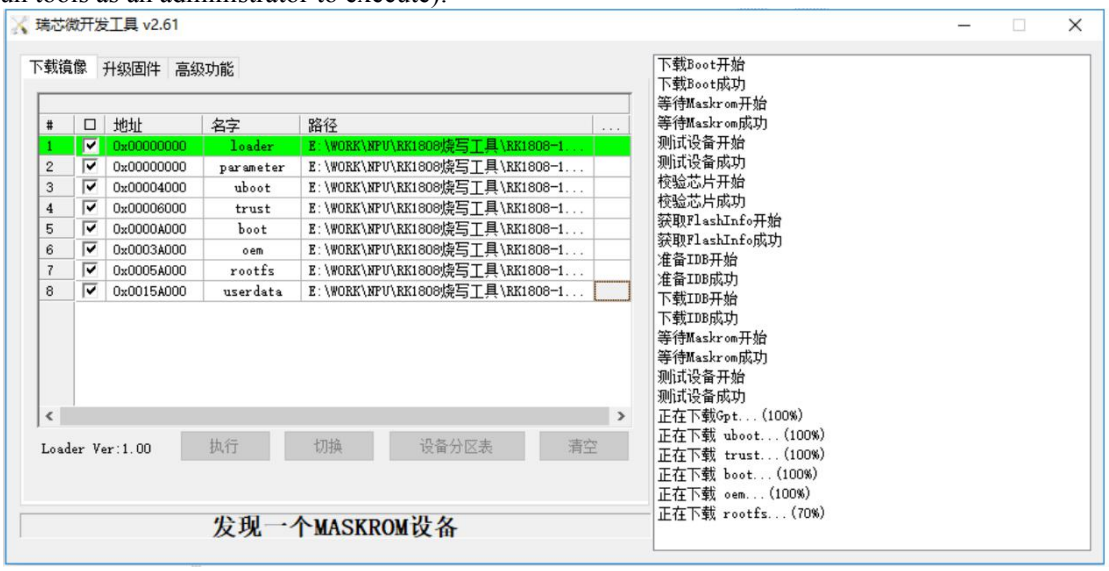


Figure 2 Upgrade tool **AndroidTool.exe**

Note：Before upgrade, need to install the latest USB driver, which is in under the below directory:

```
tools/windows/DriverAssitant_v4.7
```

6.2 Linux Upgrade Instruction

The Linux upgrade tool (Linux_Upgrade_Tool needs v1.38 or later versions) is located in tools/linux directory. Please make sure your board is connected tomaskrom/loader rockusb, if the compiled firmware is in rockdev directory, update 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 -r              rockdev/recovery.img
sudo ./upgrade_tool di -oem         rockdev/oem.img
sudo ./upgrade_tool di -rootfs       rockdev/rootfs.img
sudo ./upgrade_tool di -userdata   rockdev/userdata.img
sudo ./upgrade_tool  rd
```

Or in root directory, machine run the following command to upgrade in maskrom state:

```
./rkflash.sh
```

6.3 System Partition Introduction

Default partition description (below is RK1808 evb partition reference):

Number	Start (sector)	End (sector)	Size	Code	Name
1	16384	24575	4096K	0700	uboot
2	24576	32767	4096K	0700	trust
3	32768	40959	4096K	0700	misc
4	40960	106495	32.0M	0700	boot
5	106496	172031	32.0M	0700	recovery
6	172032	237567	32.0M	0700	backup
7	237568	368639	64.0M	0700	oem
8	368640	3514367	1536M	0700	rootfs
9	3514368	30535646	12.8G	0700	userdata

Uboot partition: update uboot.img Compiled by uboot.
trust partition: update trust.img Compiled by uboot.
misc partition: update misc.img. Use for recovery.
boot partition: update boot.img Compiled by kernel.
recovery partition: update recovery.img.
backup partition: reserved, temporarily useless. Used for backup of recovery like android in future.
oem partition: used by manufacturer to store manufacturer's app or data. Read only. Replace the data partition of original speakers. Mounted in /oem directory
rootfs partition: store rootfs.img compiled by buildroot or debian, read only.
userdata partition: store files temporarily generated by app or for end users. Read and write, mounted in /userdata directory.

Chapter 7 RK1808 SDK Firmware

RK1808_Linux_V1.1.0_20190808 firmware can be downloaded from the following address:

https://rockchips-my.sharepoint.com/:f:/g/personal/lin_huang_rockchips_onmicrosoft_com/EtlfLj7zywNHnLpdZFbGJ8wBZ4-UVP8H12lWGfTBtHt8bw?e=wkWor7

Chapter 8 RK1808 NPU Related Development Tools

This SDK includes an rknn_demo (InceptionV2 SSD). For details, see docs/SoC platform relate/RK1808/Rockchip RKNN_DEMO Module Development Guide V0.2.pdf in project directory.
This SDK includes the rknn-toolkit which is in project directory external/rknn-toolkit. For detailed instructions, please refer to document docs/SoC platform relate/RK1808/RKNN-Toolkit User Guide_V1.1.0.pdf.
Other NPU reference documents, please see “docs/Develop reference documents/NPU/” for details.

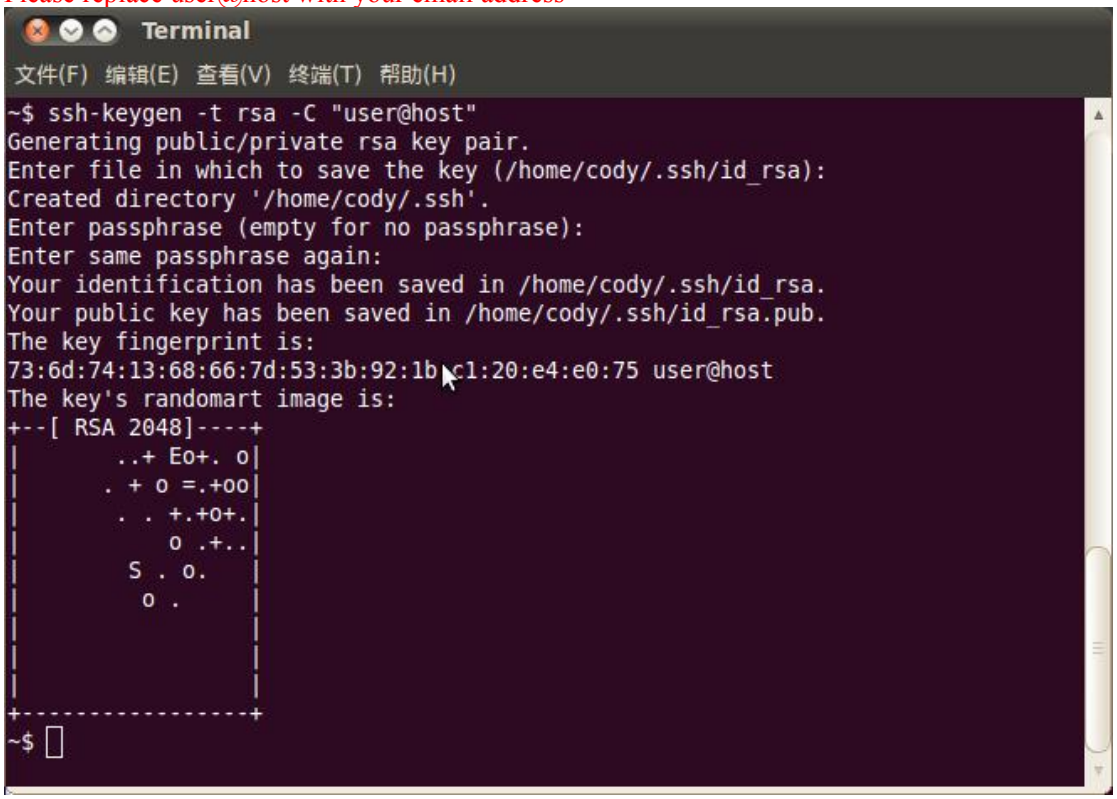
Chapter 9 SSH Public Key Operation Instruction

9.1 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



A public key file will be generated in your directory after running the command.

```
~$ 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 the generated private key file id_rsa and password properly, and email the public key id_rsa.pub to fae@rock-chips.com, and CC corresponding sales to open the SDK download permission.

9.2 Use key-chain to Manage Keys

It is recommended to use a simple tool keychain to manage keys.
The detail usage is as follows:
1. Install keychain package:

```
$sudo aptitude install keychain
```

2. Configure the key:

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

Add the following line:

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

id_rsa is private key file name among them.
After the above configuration, log in to console again and you will be prompted to enter password. Just enter password used to generate the key. If there is no password, you can skip it.
In addition, try not to use sudo or root users unless you know how to deal with them, which will lead to permission and key management confusion.

9.3 Multiple Machines Use the Same SSH Public Key

Use on different machines, you can copy ssh private key file id_rsa to "~/.ssh/id_rsa" of machines you want to use.
The following prompt will appear 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
```

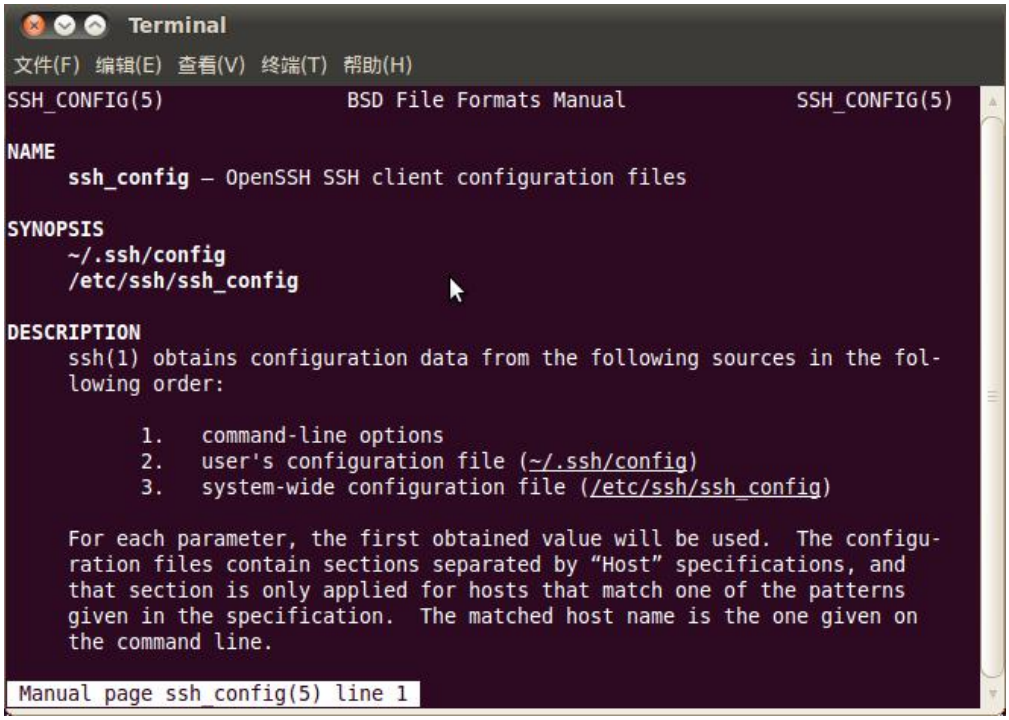
Enter the following command in console to solve

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

9.4 One Machine Switches Different SSH Public Keys

You can configure SSH by referring to ssh_config document.

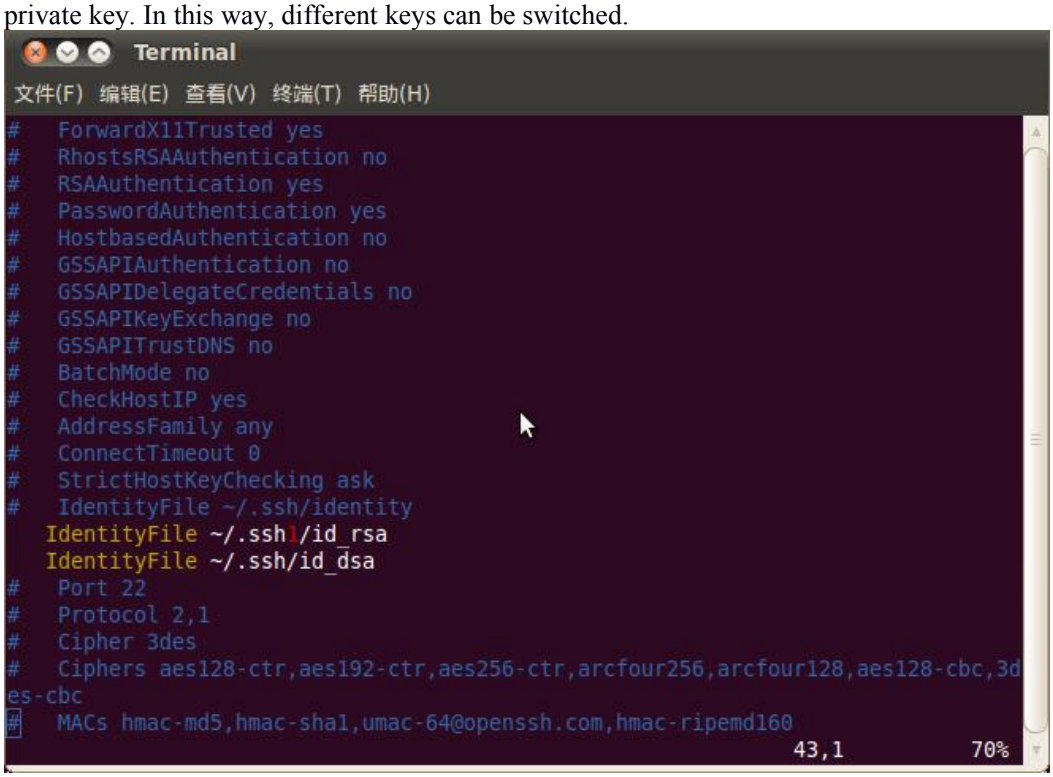
```
~$ man ssh_config
```



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.



9.5 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.

Please keep your private key file in a safe place. Do not grant second authorization to third parties.

9.6 Git Access Application Instruction

Please email the public key file created according to above chapter to fae@rock-chips.com, to apply for SDK code download permission.