

Rockchip RK3288 Linux SDK

发布说明

发布版本: **2.1**

日期: **2018.09**

免责声明

本文档按“现状”提供，福州瑞芯微电子股份有限公司（“本公司”，下同）不对本文档的任何陈述、信息和内容的准确性、可靠性、完整性、适销性、特定目的性和非侵权性提供任何明示或暗示的声明或保证。本文档仅作为使用指导的参考。

由于产品版本升级或其他原因，本文档将可能在未经任何通知的情况下，不定期进行更新或修改。

商标声明

“Rockchip”、“瑞芯微”、“瑞芯”均为本公司的注册商标，归本公司所有。

本文档可能提及的其他所有注册商标或商标，由其各自所有者所有。

版权所有 © 2018 福州瑞芯微电子股份有限公司

超越合理使用范畴，非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

福州瑞芯微电子股份有限公司

Fuzhou Rockchip Electronics Co., Ltd.

地址：福建省福州市铜盘路软件园 A 区 18 号

网址：www.rock-chips.com

客户服务电话：+86-591-83991906

客户服务传真：+86-591-83951833

客户服务邮箱：service@rock-chips.com

前言

概述

本 SDK 是基于 Linux 系统，内核基于 kernel 4.4，适用于 RK3288 EVB(兼容 RK3288 W version)及基于其开发的 Linux 产品。本 SDK 支持 VPU 硬解码、GPU 3D、Wayland 显示、QT 等功能。具体功能调试和接口说明，请阅读工程目录 docs 下的文档。

读者对象

本文档（本指南）主要适用于以下工程师：

- 技术支持工程师
- 软件开发工程师

修订记录

日期	版本	作者	修改说明
2018-09-26	V2.1	Nickey Yang	更新 SDK 编译和烧写方法
2018-06-20	V2.0	Nickey Yang	更新 SDK 规范和编译方式
2018-04-26	V1.3	Nickey Yang	增加 buildroot 编译说明
2018-04-23	V1.1	Nickey Yang	重命名和修订格式
2018-04-16	V1.0	Nickey Yang	初始版本

目录

1	SDK 获取说明	1-1
1.1	从瑞芯微代码服务器获取	1-1
1.2	从 github 开源网站获取	1-1
2	SDK 编译说明	2-1
2.1	BoardConfig 编译机制	2-1
2.2	Uboot 编译	2-2
2.3	Kernel 编译	2-2
2.4	Recovery 编译	2-2
2.5	buildroot 编译	2-3
2.6	全自动编译	2-3
2.7	Debian 系统编译	2-3
3	刷机说明	3-1
3.1	Windows 刷机说明	3-1
3.2	Linux 刷机说明	3-2
3.3	系统分区说明	3-3
4	Secure CRT 参数设置	4-1
5	RK3288 Linux 工程目录介绍	5-1
6	RK3288 SDK 固件	6-1
7	SSH 公钥操作说明	7-1
7.1	SSH 公钥生成	7-1
7.2	使用 key-chain 管理密钥	7-1
7.3	多台机器使用相同 SSH 公钥	7-2
7.4	一台机器切换不同 SSH 公钥	7-2
7.5	密钥权限管理	7-4
7.6	repo 权限申请说明	7-4

插图目录

图 3-1 RK_EVB_RK3288_LPDDR3P232SD6V10 开发板 3-1

图 3-2 烧写工具 AndroidTool 3-2

图 4-1 Secure CRT 参数设置 4-1

1 SDK 获取说明

本 SDK 可以通过瑞芯微代码服务器或者 github 开源网站获取。

1.1 从瑞芯微代码服务器获取

获取 RK3288 Linux 的软件包，需要有帐户能访问 Rockchip 提供的源代码仓库。客户向瑞芯微技术窗口申请 SDK，会同步提供 SSH 公钥进行服务器认证授权，获得授权后即可同步代码。关于瑞芯微代码服务器 SSH 公钥授权，请参考第 7 节 SSH 公钥操作说明。

RK3288 Linux SDK 的代码和相关文档被划分为了若干 git 仓库分别进行版本管理，开发者可以使用 repo 对这些 git 仓库进行统一的下载、提交、切换分支等操作。其下载地址如下：

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

RK3288 Linux SDK 下载命令如下：

```
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  
rk3288_linux_release.xml
```

为方便客户快速获取 SDK 源码，瑞芯微技术窗口通常会提供对应版本的 SDK 初始压缩包，开发者可以通过这种方式，获得 SDK 代码的初始压缩包，该压缩包解压得到的源码与通过 repo 下载的源码是一致的。以 rk3288_linux_sdk_v2.1_20180620.tgz 为例，拷贝到该初始化包后，通过如下命令可检出源码：

```
mkdir rk3288  
tar xvf rk3288_linux_sdk_v2.1_20180620.tgz -C rk3288  
cd rk3288  
.repo/repo/repo sync -l  
.repo/repo/repo sync -c
```

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

1.2 从 github 开源网站获取

下载 repo 工具：

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

建立并进入 rk3288_linux 目录：

```
mkdir rk3288_linux && cd rk3288_linux
```

初始化 repo 仓库：

```
../repo/repo init --repo-url=https://github.com/rockchip-linux/repo -u  
https://github.com/rockchip-linux/manifests -b master -m rk3288_linux_release.xml
```

同步下载整个工程：

```
../repo/repo sync
```

注意事项：

若使用 https 下载有问题，可替换上述初始化 repo 仓库命令中的 https 为 http，来使用 http 的下载。

2 SDK 编译说明

本 SDK 开发环境是在 Ubuntu 上开发测试的。我们推荐使用 Ubuntu 16.04 的系统进行编译。其他的 Linux 版本可能需要对软件包做相应调整。除了系统要求外，还有其他软硬件方面的要求。

硬件要求：64 位系统，硬盘空间大于 40G。如果您进行多个构建，将需要更大的硬盘空间。

软件要求：

Ubuntu 16.04 系统编译所需的软件包

```
sudo apt-get install git gcc-arm-linux-gnueabi u-boot-tools device-tree-compiler mtools parted libudev-dev libusb-1.0-0-dev python-linaro-image-tools linaro-image-tools libssl-dev autotools-dev libsigsegv2 m4 libdrm-dev curl sed make binutils build-essential gcc g++ bash patch gzip bzip2 perl tar cpio python unzip rsync file bc wget libncurses5 libglib2.0-dev openssh-client
```

Ubuntu 17.04 系统编译所需的软件包

```
sudo apt-get install git gcc-arm-linux-gnueabi u-boot-tools device-tree-compiler mtools parted libudev-dev libusb-1.0-0-dev python-linaro-image-tools linaro-image-tools libssl-dev autotools-dev libsigsegv2 m4 libdrm-dev curl sed make binutils build-essential gcc g++ bash patch gzip bzip2 perl tar cpio python unzip rsync file bc wget libncurses5 libglib2.0-dev openssh-client lib32gcc-7-dev g++-7 libstdc++-7-dev
```

Ubuntu 18.04 系统编译所需的软件包

```
sudo apt-get install git gcc-arm-linux-gnueabi u-boot-tools device-tree-compiler mtools parted libudev-dev libusb-1.0-0-dev lib32gcc-7-dev python-linaro-image-tools autotools-dev linaro-image-tools libstdc++-7-dev autoconf 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 libglib2.0-dev libgtk2.0-dev libglade2-dev cvs mercurial openssh-client w3m dlatex texinfo graphviz python-matplotlib libssl-dev e2fsprogs fakeroot devscripts lib2c-dev pv libncurses5-dev
```

若编译遇到报错，请视报错信息，安装对应的软件包。

2.1 BoardConfig 编译机制

现有的 sdk 引入了 BoardConfig 编译机制，搭配 build.sh 脚本来完成自动化的编译和固件生成。

BoardConfig 中部分环境变量的介绍如下：

```
# Uboot defconfig
export RK_UBOOT_DEFCONFIG=fennec-rk3288
# Kernel dts
export RK_KERNEL_DTS=rk3288-evb-rk808-linux
# parameter for GPT table
export RK_PARAMETER=parameter-buildroot.txt
# rootfs image path
export RK_ROOTFS_IMG
```

默认的 BoardConfig 配置对应的是 rk3288-evb-rk808 开发板，上层使用 buildroot 系统。对于不同的板子，需要使用不同的 BoardConfig 配置文件：

rk3288-evb-act8846：请先在工程目录根目录下执行 `./build.sh BoardConfig_8846.mk`

rk3288-firefly：请先在工程目录根目录下执行 `./build.sh BoardConfig_firefly.mk`

如果需要使用 debian 系统，请先在工程目录根目录下应用对应的 debian config，比如 rk3288-evb-rk808 就是 `./build.sh BoardConfig_debian.mk`

BoardConfig 位于在 **device/rockchip/rk3288** 下，客户定制化的板子请根据自己的情况先行添加并应用对应的 **BoardConfig** 文件。

2.2 Uboot 编译

进入工程目录根目录执行以下命令完成 uboot 的编译及打包：

- rk3288-evb-rk808 开发板：`cd u-boot && ./make.sh fennec-rk3288` 或 `./build.sh u-boot`
- rk3288-evb-act8846 开发板：`cd u-boot && ./make.sh evb-rk3288`
- rk3288-firefly 开发板：`cd u-boot && ./make.sh firefly-rk3288`

编译后会在 u-boot 目录下生成下面三个文件：

u-boot/

└─ rk3288_loader_v1.06.236.bin

└─ trust.img

└─ uboot.img

2.3 Kernel 编译

进入工程目录根目录执行以下命令完成 kernel 的编译及打包：

- rk3288-evb-rk808 开发板：
`cd kernel`
`make ARCH=arm rockchip_linux_defconfig`
`make ARCH=arm rk3288-evb-rk808-linux.img -j12`
或
`./build.sh kernel`
- rk3288-evb-act8846 开发板：
`cd kernel`
`make ARCH=arm rockchip_linux_defconfig`
`make ARCH=arm rk3288-evb-act8846.img -j12`
- rk3288-firefly 开发板：
`cd kernel`
`make ARCH=arm rockchip_linux_defconfig`
`make ARCH=arm rk3288-firefly.img -j12`

编译后会在 kernel 目录下生成下面 zboot.img，这个 zboot.img 包含编译 kernel 生成的 zImage、编译 dts 生成的 dtb、logo.bmp 及 logo_kernel.bmp。

2.4 Recovery 编译

进入工程目录根目录执行以下命令完成 recovery 的编译及打包：


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

编译后会在 buildroot/output/rockchip_rk3288_recovery/images 目录下生成 recovery.img。

注意事项:

recovery.img 需要打包 kernel 生成的固件，所以请在编译 recovery 之前先完成 kernel 的编译。

2.5 buildroot 编译

进入工程目录根目录执行以下命令完成 buildroot 及 app 的编译及打包:

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

编译后会在 buildroot/output/rockchip_rk3288/images 目录下生成 rootfs.ext4。

注意事项:

若需要编译单个模块或者第三方应用，需要对交叉编译环境进行配置。请查阅 sdk 目录 /doc/rockchip linux 软件开发指南第 10 节: Buildroot 开发有更详细的介绍。

2.6 全自动编译

进入工程目录根目录执行以下命令完成上述 uboot/kernel/recovery/buildroot 等各部分的编译以及固件的打包:

```
./build.sh
```

2.7 Debian 系统编译

先进入 rootfs/目录

```
cd rootfs/
```

2.7.1 Build 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
```

```
ARCH=armhf ./mk-base-debian.sh
```

编译后会在 rootfs 目录下生成: linaro-stretch-alip-xxxxx-1.tar.gz(xxxxxx 表示生成时间戳)

2.7.2 Build rk-debian rootfs

```
RELEASE=stretch ARCH=armhf ./mk-rootfs.sh
```

2.7.3 Creat the ext4 image

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

编译后会在 rootfs 目录下生成: linaro-rootfs.img。

2.7.4 Package debian firmware

进入工程目录根目录执行以下命令完成 uboot/kernel/recovery/debian 等各部分的固件打包到 rockdev 目录下:

```
./mkfirmware.sh
```

3 刷机说明

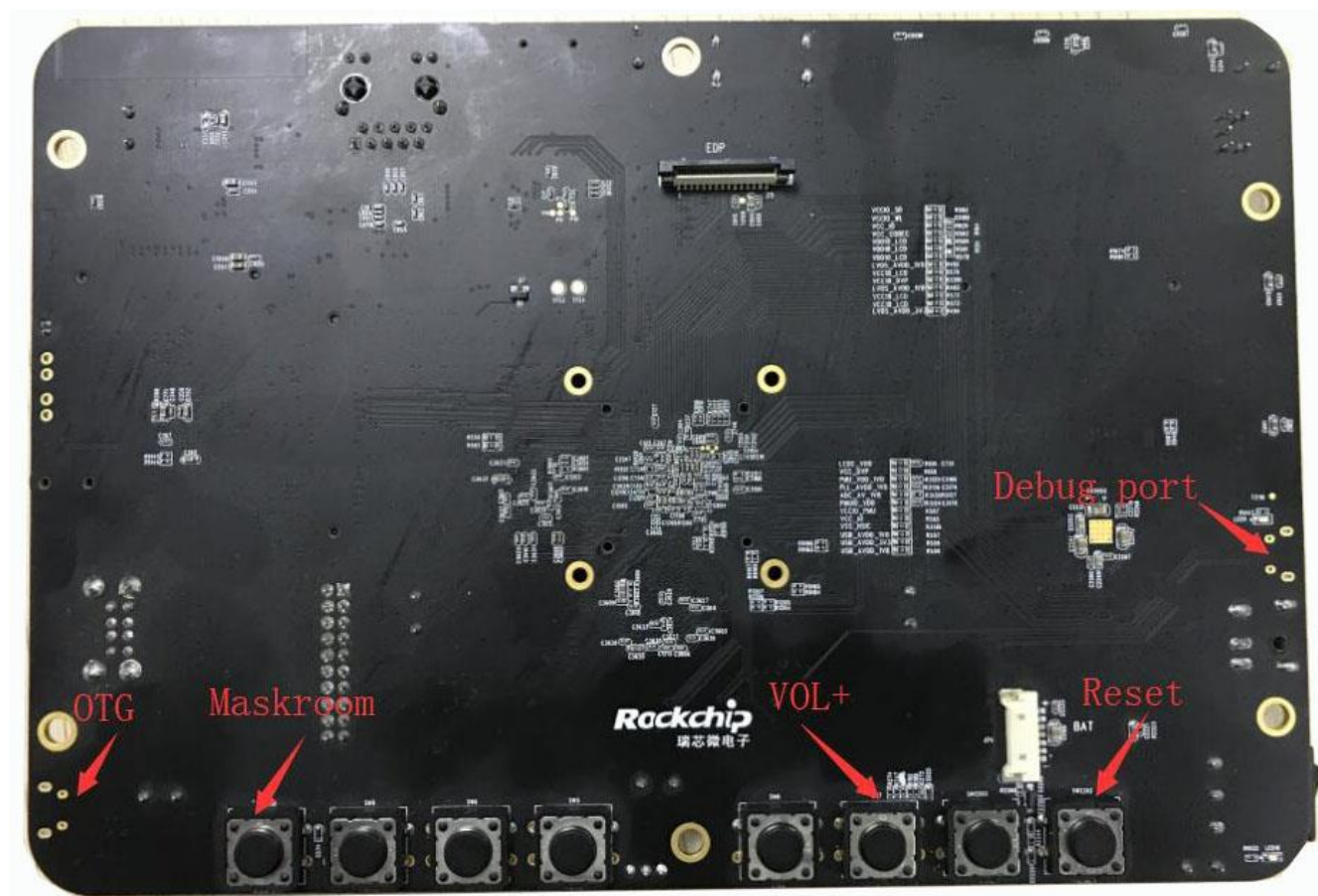


图 3-1 RK_EVB_RK3288_LPDDR3P232SD6V10 开发板

3.1 Windows 刷机说明

本 SDK 支持 Windows 烧写(工具版本需 V2.55 或以上), 工具位于 tools/windows/AndroidTool。

如上图, 编译生成相应的固件后, 设备烧写需要进入 MASKROM 烧写模式, 连接好 usb 下载线后, 按住按键“MASKROM”不放并按下复位键“Reset”后松手, 就能进入 MASKROM 模式, 加载编译生成固件的相应路径后, 点击“执行”进行烧写。也可以按“vol+/recovery”按键不放并按下复位键“Reset”后松手进入 LOADER 模式进行烧写, 下面是 MASKROM 模式的分区偏移及烧写文件。



图 3-2 烧写工具 AndroidTool

注意事项:

AndroidTool 可能需要使用管理员权限运行

烧写前需安装最新驱动，详见：tools/windows/DriverAssitant_v4.6.zip

3.2 Linux 刷机说明

本 SDK 支持 Linux 下的烧写，工具位于 tools/linux 目录下(Linux_Upgrade_Tool 版本需 V1.33 或以上)。

在工程目录根目录，机器在 MASKROM / LOADER ROCKUSB 状态运行如下命令进行完整的固件升级：

```
./rkflash.sh
```

同时，系统也支持单独分区的烧写，方便开发的时候 debug，

在 LOADER ROCKUSB 模式：

```
更新 loader 分区： ./rkflash.sh loader
```

```
更新 parameter 分区： ./rkflash.sh parameter
```

```
更新 uboot 分区： ./rkflash.sh uboot
```

```
更新 trust 分区： ./rkflash.sh trust
```

```
更新 kernel 分区： ./rkflash.sh boot
```

```
更新 recovery 分区： ./rkflash.sh recovery
```

```
更新 oem 分区： ./rkflash.sh oem
```

```
更新 misc 分区： ./rkflash.sh misc
```

更新 userdata 分区: `./rkflash.sh userdata`

更新 rootfs 分区: `./rkflash.sh rootfs`

在 MASKROM 模式:

需要先执行 `./rkflash.sh loader` 然后使用上述命令对指定的分区进行烧写。

擦除板子上的固件:

```
sudo ./tools/linux/Linux_Upgrade_Tool/Linux_Upgrade_Tool/upgrade_tool ef  
rockdev/MiniLoaderAll.bin
```

3.3 系统分区说明

默认分区说明:

uboot 分区: 烧写 uboot 编译出来的 uboot.img

trust 分区: 烧写 uboot 编译出来的 trust.img

misc 分区: 烧写 misc.img, 给 recovery 使用

boot 分区: 烧写 kernel 编译出来的 zboot.img

recovery 分区: 烧写 recovery.img

backup 分区: 预留,暂时没用。后续作为 recovery 的 backup 使用

oem 分区: 给厂家使用,存放厂家的 app 或数据。只读。

rootfs 分区: 存放 buildroot 或者 debian 编出来的 rootfs.img

userdata 分区: 存放 app 临时生成的文件或给最终用户使用。可读写, 挂载在 /userdata 目录下

新增分区配置:

请参考 \RKDocs\RKTools manuals\Android 增加一个分区配置指南 V1.00.pdf

4 Secure CRT 参数设置

Windows 下可以利用 Secure CRT 软件打印调试信息，需要对串口参数进行设置，具体设置细节如下图：

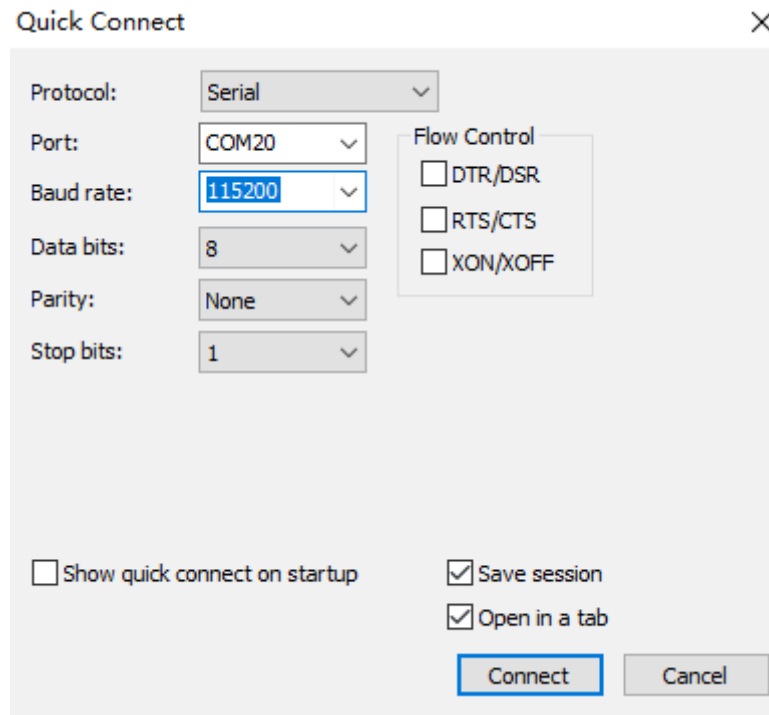


图 4-1 Secure CRT 参数设置

5 RK3288 Linux 工程目录介绍

工程目录下有 buildroot、recovery、app、kernel、u-boot、device、docs、external 等目录。每个目录或其子目录会对应一个 git 工程，提交需要在各自的目录下进行。

- 1)app: 上层应用，包含 QLauncher 等应用程序
- 2)buildroot: buildroot 根文件系统
- 3)device/rockchip/rk3288: 编译和打包固件的脚本和预备文件
- 4)docs: 工程开发参考文档
- 5)external: 相关库，包括音频、视频、网络等
- 6)kernel: 内核代码
- 7)prebuilts: 交叉编译工具链
- 8)recovery: recovery 工程文件
- 9)rkbin: 编译 uboot 时需要的固件和工具
- 10)rockdev: 打包生成的固件目录
- 11)rootfs: Debian 根文件系统
- 12)tools: Linux/Windows 下打包、烧录工具
- 13)u-boot: uboot 代码

6 RK3288 SDK 固件

RK3288 Linux 的固件可以通过下列 ftp 服务器地址获取:

ftp://ftp.rock-chips.com

user: linux_rk3288

psw: hZx9WgsuE6

链接地址包含如下内容:

readme.txt : 固件的使用说明和更新说明

rk3288-evb-rk808 开发板: rk3288-evb-rk808-V2.1_20180926.tar

rk3288-evb-act8846 开发板: rk3288-evb-act8846-V2.1_20180926.tar

rk3288-firefly 开发板: rk3288-firefly-V2.1_20180926.tar

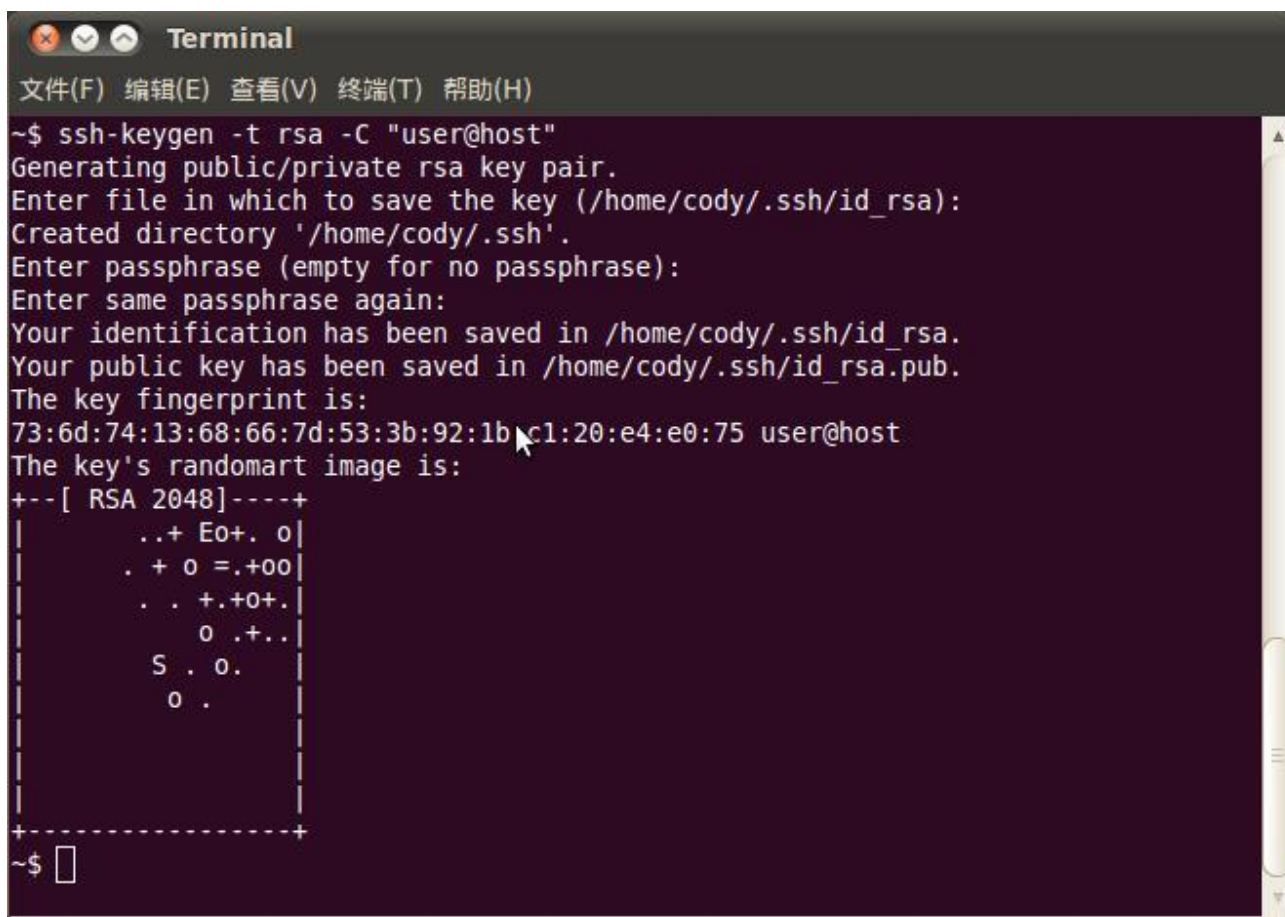
7 SSH 公钥操作说明

7.1 SSH 公钥生成

使用如下命令生成公钥：

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

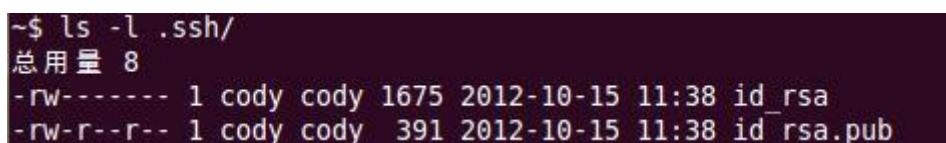
请将 user@host 替换成您的邮箱地址



```
Terminal
文件(F) 编辑(E) 查看(V) 终端(T) 帮助(H)

~$ 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:11:20:e4:e0:75 user@host
The key's randomart image is:
+--[ RSA 2048 ]-----+
|      ..+ Eo+. o|
|      . + 0 =.+00|
|      . . +.+0+.|
|      o .+..|
|      S . o.|
|      o .|
+-----+
~$
```

命令运行完成会在你的目录下生成 key 文件。



```
~$ 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
```

请妥善保存生成的私钥文件 id_rsa 和密码，并将 id_rsa.pub 发邮件给 SDK 发布服务器的管理员。

7.2 使用 key-chain 管理密钥

推荐您使用比较简易的工具 keychain 管理密钥。

具体使用方法如下：

1. 安装 keychain 软件包：

```
$sudo aptitude install keychain
```

2. 配置使用密钥：

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

增加下面这行：


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

其中, `id_rsa` 是私钥文件名称。

以上配置以后, 重新登录控制台, 会提示输入密码, 只需输入生成密钥时使用的密码即可, 若无密码可不输入。另外, 请尽量不要使用 `sudo` 或 `root` 用户, 除非您知道如何处理, 否则将导致权限以及密钥管理混乱。

7.3 多台机器使用相同 SSH 公钥

在不同机器使用, 可以将你的 `ssh` 私钥文件 `id_rsa` 拷贝到要使用的机器的“`~/.ssh/id_rsa`”即可。在使用错误的私钥会出现如下提示, 请注意替换成正确的私钥。

```
~/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: █
```

添加正确的私钥后, 就可以使用 `git` 克隆代码, 如下图。

```
~$ 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
```

添加 `ssh` 私钥可能出现如下提示错误。

Agent admitted failure to sign using the key

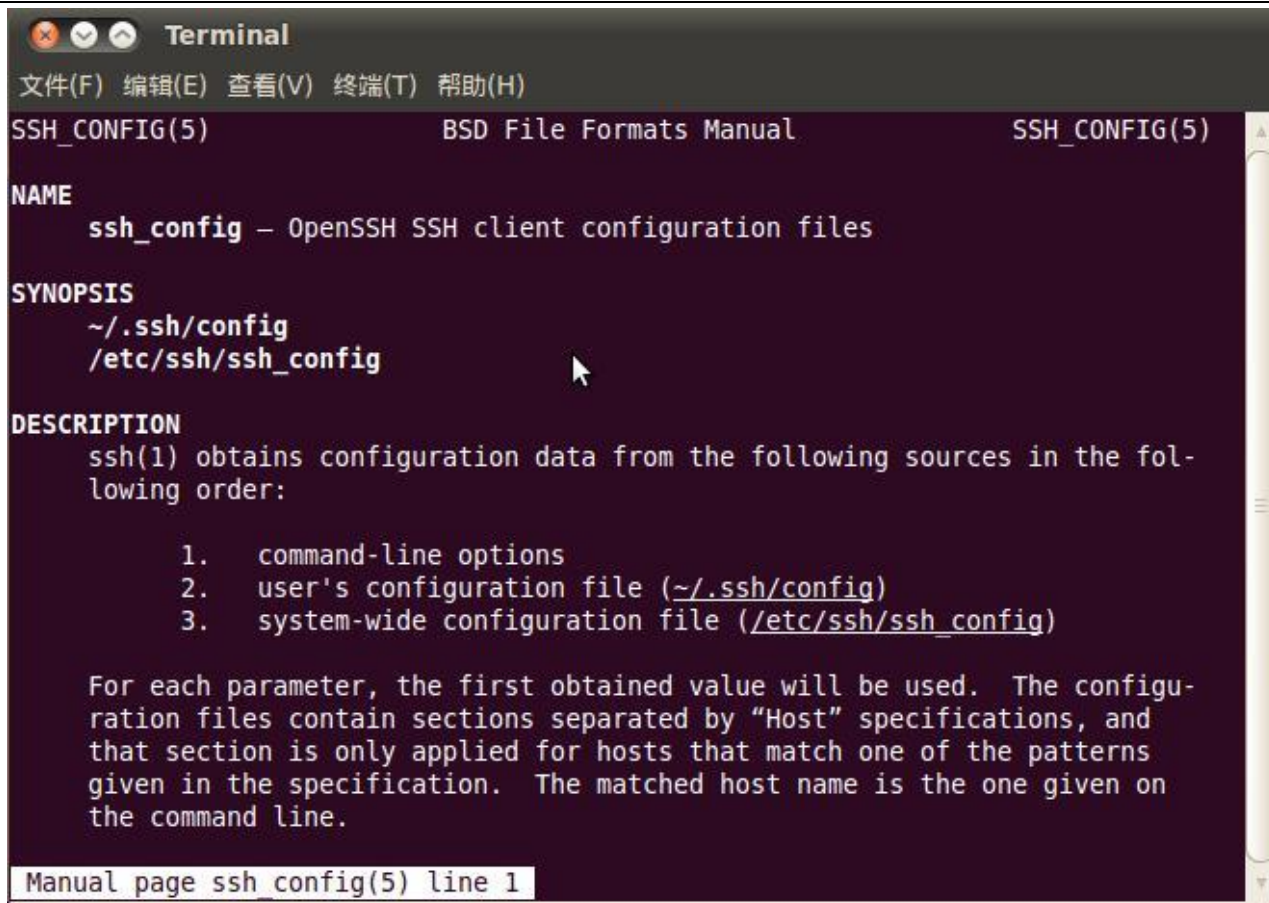
在 `console` 输入如下命令即可解决。

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

7.4 一台机器切换不同 SSH 公钥

可以参考 `ssh_config` 文档配置 `SSH`。

```
~$ man ssh_config
```



```
Terminal
文件(F) 编辑(E) 查看(V) 终端(T) 帮助(H)
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 following 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 configuration 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.

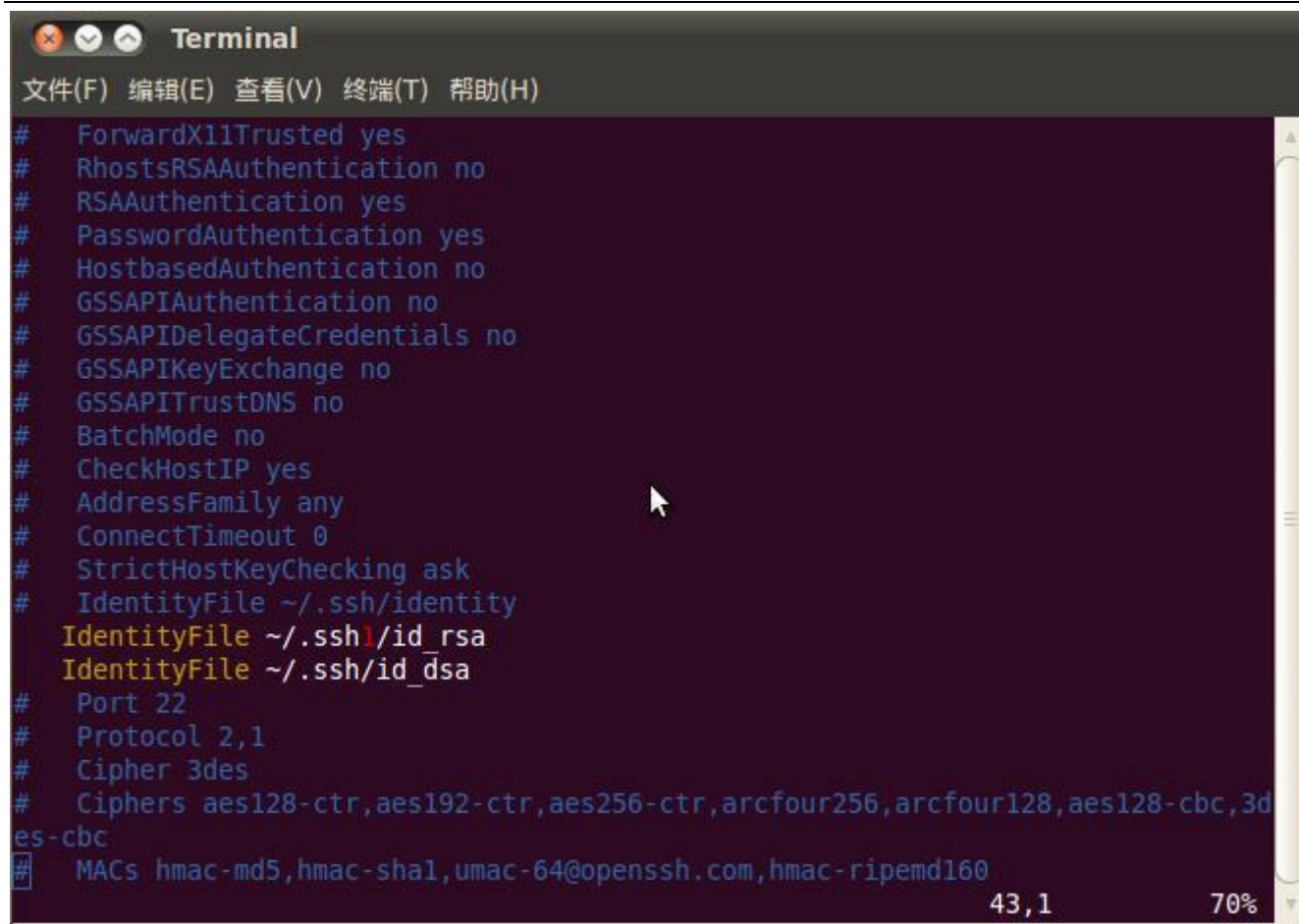
Manual page ssh_config(5) line 1
```

通过如下命令，配置当前用户的 SSH 配置。

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

```
~$ vi .ssh/config
```

如图，将 ssh 使用另一个目录的文件“~/.ssh1/id_rsa”作为认证私钥。通过这种方法，可以切换不同的的密钥。

A terminal window titled "Terminal" with a menu bar containing "文件(F)", "编辑(E)", "查看(V)", "终端(T)", and "帮助(H)". The terminal displays a list of SSH configuration options, each preceded by a hash symbol (#). The options are: 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 ~/.ssh/id_rsa, IdentityFile ~/.ssh/id_dsa, Port 22, Protocol 2,1, Cipher 3des, Ciphers aes128-ctr,aes192-ctr,aes256-ctr,arcfour256,arcfour128,aes128-cbc,3des-cbc, and MACs hmac-md5,hmac-sha1,umac-64@openssh.com,hmac-ripemd160. The terminal has a dark background and a light-colored cursor. The bottom right corner shows "43,1" and "70%".

```
Terminal
文件(F) 编辑(E) 查看(V) 终端(T) 帮助(H)
# 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 ~/.ssh/id_rsa
IdentityFile ~/.ssh/id_dsa
# Port 22
# Protocol 2,1
# Cipher 3des
# Ciphers aes128-ctr,aes192-ctr,aes256-ctr,arcfour256,arcfour128,aes128-cbc,3des-cbc
# MACs hmac-md5,hmac-sha1,umac-64@openssh.com,hmac-ripemd160
43,1 70%
```

7.5 密钥权限管理

本服务器可以实时监控某个 key 的下载次数、IP 等信息,如果发现异常将禁用相应的 key 的下载权限。请妥善保管私钥文件。并不要二次授权与第三方使用。

7.6 repo 权限申请说明

参考上述章节,生成公钥文件,发邮件至 fae@rock-chips.com。申请开通 SDK 代码下载权限。