

Rockchip RK2108 RT-Thread SDK Release Note

ID: RK-FB-YF-356

Release Version: V1.0.0

Release Date: 2020-08-12

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.

All rights reserved. ©2020. Rockchip Electronics Co., Ltd.

Beyond the scope of fair use, neither any entity nor individual shall extract, copy, or distribute this document in any form in whole or in part without the written approval of Rockchip.

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 RK2108 RT-Thread SDK release notes, aiming to help engineers get started with RK2108 RT-Thread 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	Kernel Version
RK2108	RT-Thread v3.1.x

Revision History

Version	Author	Date	Revision History
V0.0.1	Chris Zhong	2019-09-05	Initial version
V0.1.0	Chris Zhong	2020-03-30	Modify the document directory
V1.0.0	Chris Zhong	2020-08-12	Official release version

Contents

Rockchip RK2108 RT-Thread SDK Release Note

1. Overview
2. Main Functions
3. How to Get the SDK
 - 3.1 SDK Download Command
 - 3.2 SDK Code Compression Package
 - 3.3 To Get the SDK Version
 - 3.4 SDK Code Update
4. RK2108 RT-Thread Project Directory Introduction
5. SDK Building Instructions
 - 5.1 Set up the Building Environment
 - 5.2 Basic Building and Packaging Commands
6. SSH Public Key Operation Introduction
 - 6.1 Multi-device Use the Same SSH Public Key
 - 6.2 Switch Different SSH Public Keys on the Same Device
 - 6.3 Key Authority Management
 - 6.4 Reference Documents

1. Overview

This SDK is based on RT-Thread v3.1.3, which contains system source code, drivers, tools, and application software packages used for RT-Thread system development, and it also contains development documents and tool usage documents. Adapting to RK2108 chip platform, it is suitable for RK2108 EVB development board and all products developed based on RK2108 platform.

2. Main Functions

Functions	Module Name
Data Communication	Wi-Fi, BT, USB
Audio Interfaces	Analog MIC, Digital MIC(PDM, I2S), Audio PWM, VAD
Display Interfaces	MCU panel、SPI panel
Application Demo	Audio control module, voice recorder

3. How to Get the SDK

Rockchip SDKs are released by Rockchip server. Please refer to Chapter 5 [SDK Building Introduction](#) to build a development environment.

To get RK2108 RT-Thread SDK 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(e-Mail: fae@rock-chips.com). About Rockchip server SSH public key authorization, please refer to Chapter 6 [SSH Public Key Operation Introduction](#).

3.1 SDK Download Command

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:

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

RK2108 RT-Thread SDK download command:

```
1 | repo init --repo-url ssh://git@www.rockchip.com.cn/repo/rk/tools/repo -u  
ssh://git@www.rockchip.com.cn/rtos/rt-thread/rk/platform/release/manifests -b  
master -m rk2108_release.xml
```

After the code repository is initialized, you can use the following command to synchronize the code:

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

3.2 SDK Code Compression Package

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 RK2108_RT-Thead_SDK_Release_V1.0.0_20200812.tar.gz as an example. After getting an initialization package, you can get the source code by running the following command:

```
1 tar xzvf RK2108_RT-Thead_SDK_Release_V1.0.0_20200812.tar.gz
2 cd RK2108_RT-Thead_SDK_Release_V1.0.0_20200812
3 .repo/repo/repo sync -l
4 .repo/repo/repo sync
```

3.3 To Get the SDK Version

Please get the SDK release version through project xml file by the following command:

```
1 cd .repo/manifests
2 git log rk2108_release.xml
```

Or check the current SDK version through RKDocs/RK2108_RT-Thread_Release_Note.txt.

3.4 SDK Code Update

```
1 | .repo/repo/repo sync
```

4. RK2108 RT-Thread Project Directory Introduction

The following is the main SDK directory:

```

1 | -- applications                # Rockchip application demo source code
2 | └─ AUTHORS
3 | -- bsp                        # chip related source code
4 |   └─ rockchip
5 |     └─ common
6 |       └─ drivers              # Universal driver of rockchip OS adaptation
layer
7 |         └─ hal                # Rockchip HAL (hardware Abstraction Layer)
implementation
8 |           └─ tests            # Rockchip driver test code

```

```

 9 |   |   |   | rk2108      # RK2108 main directory
10 |   |   |   |   | board  # Board level configuration
11 |   |   |   |   | build  # Build main directory and store the intermediate
    |   |   |   |   | files
12 |   |   |   |   |   | build.sh # RK2108 build script
13 |   |   |   |   |   | drivers  # RK2108 Private driver directory
14 |   |   |   |   |   | dsp_fw  # Stores dsp firmware
15 |   |   |   |   |   | Image   # Stores firmware
16 |   |   |   |   |   | tests   # RK2108 private test code
17 |   |   |   |   |   | start   # Start relataded soure code
18 |   |   |   |   |   | tools   # Rockchip commonly used tools
19 |   |   |   |   | ChangeLog.md
20 |   |   |   |   | |-- components      # various components of the system,including
    |   |   |   |   | file system, shell and framework layer and other drivers
21 |   |   |   |   |   | hifi3
22 |   |   |   |   |   | rkdsp          # DSP project directory, please refer to the
    |   |   |   |   |   | document "Rockchip_Developer_Guide_RTOS_DSP_CN.pdf" for details
23 |   |   |   |   |   | |-- documentation      # RT-Thread Official documents
24 |   |   |   |   |   | |-- examples          # RT-Thread example program and test code
25 |   |   |   |   |   | |-- include           # RT-Thread official header file directory
26 |   |   |   |   |   |   | Kconfig
27 |   |   |   |   |   |   | |-- libcpu
28 |   |   |   |   |   |   |   | LICENSE
29 |   |   |   |   |   |   |   | README.md
30 |   |   |   |   |   |   |   | README_zh.md
31 |   |   |   |   |   |   |   | |-- RKDocs      # Rockchip documents
32 |   |   |   |   |   |   |   | |-- src         # RT-Thread kernel source code
33 |   |   |   |   |   |   |   | |-- third_party # Directory of third-party code added by
    |   |   |   |   |   |   |   | Rockchip
34 |   |   |   |   |   |   |   | |-- tools      # RT-Thread official tool directory, including
    |   |   |   |   |   |   |   | menuconfig and building scripts

```

5. SDK Building Instructions

5.1 Set up the Building Environment

It is recommended to take 64-bit Ubuntu 16.04 or Ubuntu 18.04 system as an building environment, for other Linux systems have not been tested yet, it is recommended to install the release version consistent with Rockchip developers.

The building tool are SCons + GCC officially recommended by RT-Thread. SCons is an open source build system written in Python language. And GCC cross building tool is officially provided by ARM. You can directly install all the required tools by the following commands:

```

1 | sudo add-apt-repository ppa:team-gcc-arm-embedded/ppa
2 | sudo apt-get update
3 | sudo apt-get install gcc-arm-embedded scons clang-format astyle libncurses5-
    | dev build-essential python-configparser

```

If the toolchain can not be installed, you can also download them from ARM official website and specify their path through environment variables as follows:

```

1 | wget https://developer.arm.com/-/media/Files/downloads/gnu-rm/7-2018q2/gcc-
   | arm-none-eabi-7-2018-q2-update-linux.tar.bz2
2 | tar xvf gcc-arm-none-eabi-7-2018-q2-update-linux.tar.bz2
3 | export RTT_EXEC_PATH=/path/to/toolchain/gcc-arm-none-eabi-7-2018-q2-
   | update/bin

```

5.2 Basic Building and Packaging Commands

The building command is as follows:

```

1 | cd RK2108_RT-Thead_SDK_Beta_V1.0.0_20200812
2 | cd bsp/rockchip/rk2108
3 | cp board/the name of the development board /defconfig .config
4 | scons --menuconfig //Modify the switch of building modules, after exiting,
   | the rtconfig.h file will be generated.
5 | ./build.sh

```

The generated firmware is located in:

```

1 | Image/Firmware.img

```

For more detailed buildings, debugs, and flashes instructions about RK2108 RT-Thread SDK, please refer to the following document:

/RKDocs/manuals/Rockchip_RK2108_Quick_Start_RT-Thread_CN.pdf

6. 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.

6.1 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.

```
1 | Agent admitted failure to sign using the key
```

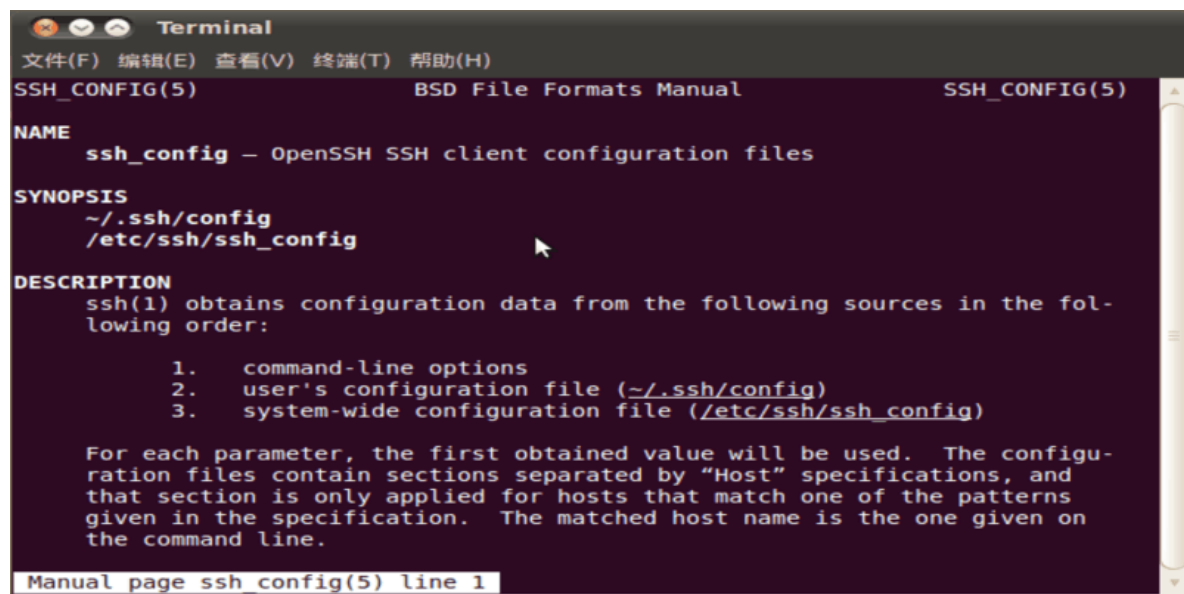
Please enter the following command in console to solve:

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

6.2 Switch Different SSH Public Keys on the Same Device

You can configure SSH according to the `ssh_config` documentation.

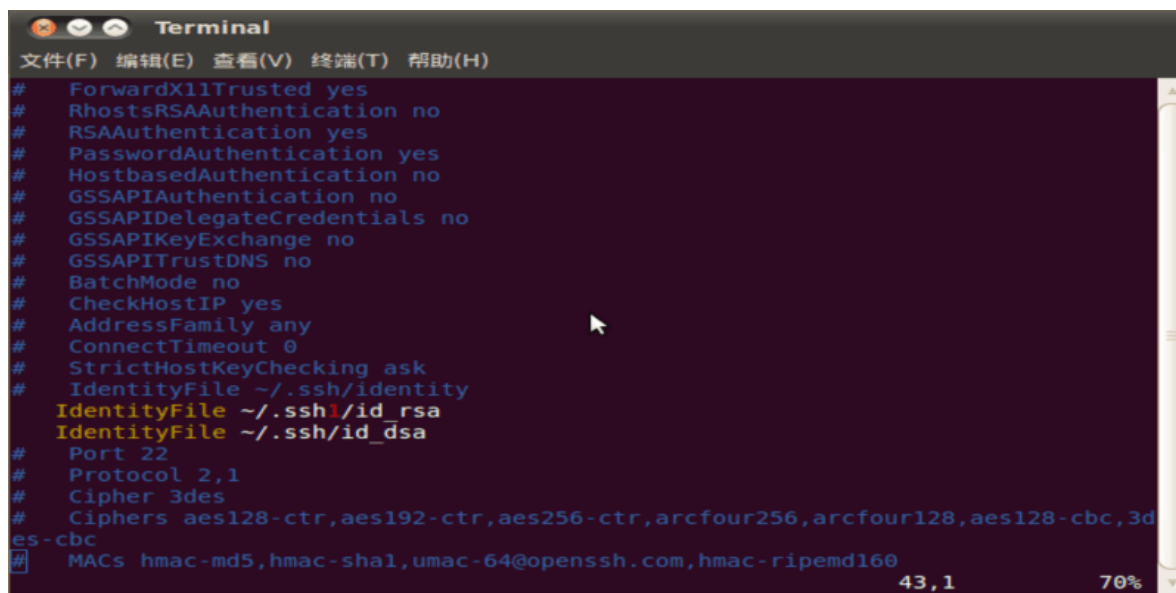
```
1 | ~$ man ssh_config
```



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

```
1 | ~$ cp /etc/ssh/ssh_config ~/.ssh/config
2 | ~$ 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.

A screenshot of a macOS Terminal window titled "Terminal". The menu bar at the top shows "文件(F)", "编辑(E)", "查看(V)", "终端(T)", and "帮助(H)". The terminal displays the output of the 'ssh -v' command, showing various configuration options and their values. The output is as follows:

```
# 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
```

The status bar at the bottom right shows "43,1" and "70%".

6.3 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.

6.4 Reference Documents

For more details, please refer to the document :

[/RKDocs/Others/Rockchip_User_Guide_SDK_Application_And_Synchronization_CN.pdf](#)