Rockchip RK2118 RT-Thread SDK Release Note

ID: RK-FB-YF-592

Release Version: V0.1.0

Release Date: 2024-03-15

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

The document presents Rockchip RK2118 RT-Thread SDK release notes, aiming to help engineers get started with RK2118 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
RK2118	RT-Thread v4.1.x

Revision History

Version	Author	Date	Revision History
V0.0.1	Roger Hu	2024-03-15	Initial version
V0.1.0	Roger Hu	2024-04-15	Beta version

Contents

Overview

This SDK is based on RT-Thread v4.1.1, 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 RK2118 chip platform, it is suitable for RK2118 EVB development board and all products developed based on RK2118 platform.

Main Functions

Functions	Module Name
Data Communication	Wi-Fi, BT, USB, Ethernet
Audio Interfaces	SAI(PCM, I2S, TDM), SPDIF TX/RX, PDM
Display Interfaces	MCU panel, SPI panel, LED Segmenet display
Application Demo	Soundbar, Partbox, Car audio, Pickup, etc.

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 RK2118 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 contact(e-Mail: fae@rock-chips.com). About Rockchip server SSH public key authorization, please refer to Chapter 6 SSH Public Key Operation Introduction.

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:

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

RK2118 RT-Thread SDK download command:

```
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 rk2118.xml
```

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

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

SDK Code Compression Package

For quick access to SDK source code, Rockchip Technical Contact 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 RK2118_RT-Thead_SDK_Alpha_V0.0.1_20240315.tar.gz as an example. After getting an initialization package, you can get the source code by running the following command:

```
tar zxvf RK2118_RT-Thead_SDK_Alpha_V0.0.1_20240315.tar.gz
cd RK2118_RT-Thead_SDK_Alpha_V0.0.1_20240315
.repo/repo/repo sync -l
.repo/repo/repo sync
```

To Get the SDK Version

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

```
cd .repo/manifests
git log rk2118.xml
```

SDK Code Update

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

SDK Directory Introduction

The following is the main SDK directory:

```
├── applications           # Rockchip application demo source
├── AUTHORS
├── bsp                    # Chip related source code
│   ├── rockchip
│   │   ├── common
│   │   │   ├── drivers    # Universal driver of Rockchip OS adaptation layer
│   │   │   ├── hal        # Rockchip HAL (Hardware abstraction layer)
implementation
│   │   │   └── tests      # Rockchip driver test code
│   │   ├── rk2118         # RK2118 main directory
│   │   │   ├── board      # Board level configuration
│   │   │   ├── build      # Build main directory and store the intermediate
files
│   │   │   ├── drivers    # RK2118 private driver directory
│   │   │   ├── Image      # Stores firmware
│   │   └── tools          # Rockchip commonly used tools
├── ChangeLog.md
├── components             # Various components of the system, including file
system, shell and framework layer and other drivers
│   ├── hifi4
│   │   ├── dsp            # DSP code
│   │   ├── rtt            # DSP related code running on mcu
│   │   ├── shared         # mcu/dsp common code
│   │   └── tools          # dsp firmware generation tool
├── documentation          # RT-Thread official documentation
├── examples               # RT-Thread example program and test code
├── include                # RT-Thread official header file directory
├── Kconfig
├── libcpu
├── LICENSE
├── README.md
├── README_zh.md
├── RKDocs                # Rockchip documents
├── src                   # RT-Thread kernel source code
├── third_party           # Directory of third-party code added by Rockchip
└── tools                 # RT-Thread official tool directory, including
menuconfig and building scripts
```

SDK Building Instructions

Set up the Building Environment

The recommended build environment for this SDK is a 64-bit version of Ubuntu 20.04 or Ubuntu 18.04. It has not been tested on other Linux distributions, so it is recommended to install the same distribution as RK developers.

The compilation tools used are SCons + GCC, as officially recommended by RT-Thread. SCons is an open-source build system written in Python, and the GCC cross-compiler is provided by ARM officially. You can directly use the following commands to install all the required tools:

```
sudo apt-get install gcc-arm-embedded scons clang-format astyle libncurses5-dev build-essential python-configparser
```

Download the compiler from the ARM official website, and specify the path to the toolchain via environment variables, as detailed below:

```
wget https://developer.arm.com/-/media/Files/downloads/gnu/13.2.rel1/binrel/arm-gnu-toolchain-13.2.rel1-x86_64-arm-none-eabi.tar.xz
tar xvf arm-gnu-toolchain-13.2.rel1-x86_64-arm-none-eabi.tar.xz
export RTT_EXEC_PATH=/path/to/toolchain/arm-gnu-toolchain-13.2.Rel1-x86_64-arm-none-eabi/bin
```

Or use the compiler included in the initial release package of the SDK: arm-gnu-toolchain-13.2.rel1-x86_64-arm-none-eabi.tar.xz, as detailed below:

```
tar -xvf arm-gnu-toolchain-13.2.rel1-x86_64-arm-none-eabi.tar.xz
export RTT_EXEC_PATH=/path/to/toolchain/arm-gnu-toolchain-13.2.Rel1-x86_64-arm-none-eabi/bin
```

Basic Building and Packaging Commands

The building command is as follows:

```
cd RK2118_RT-Thead_SDK_Alpha_V0.0.1_20240315
cd bsp/rockchip/rk2118
# Choose a board level such as adsp_demo
cp board/adsp_demo/defconfig .config
# Use .config to generate rtconfig.h
scons --useconfig=.config
# optional: Configure module switches according to your needs
scons --menuconfig
# Compile
scons -j8
# Package the firmware
./mkimage.sh board/adsp_demo/setting.ini
```

Or use the build script `build.sh` to compile, replacing the steps above:

```
# Note: Even if the configuration has not been modified, choose YES to save the new configuration when exiting menuconfig.
./build.sh adsp_demo
```

The generated firmware is located in:

```
Image/Firmware.img
```

For more detailed buildings, debugs, and flashes instructions about RK2118 RT-Thread SDK, please refer to the following document:

`Rockchip_RK2118_Quick_Start_RT-Thread_EN.pdf`

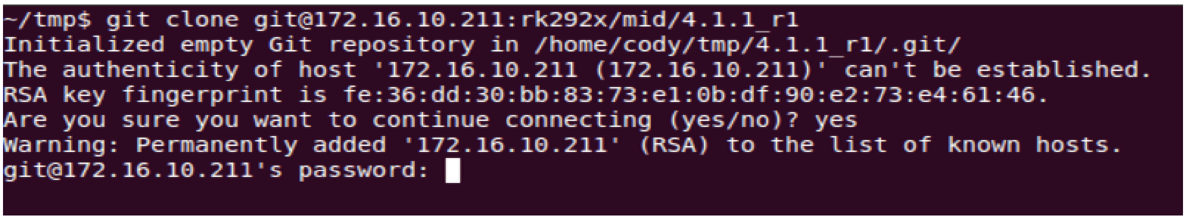
SSH Public Key Operation Introduction

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.

Multi-device Use the Same SSH Public Key

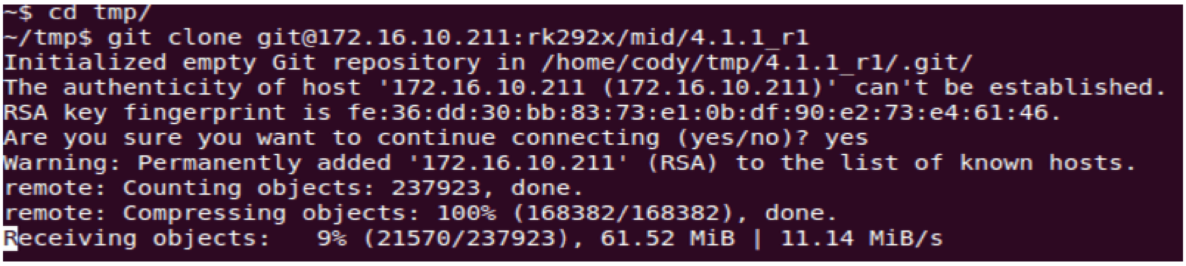
If the same SSH public key should be used in different devices, you can copy the SSH private key file id_rsa to "~/.ssh/id_rsa" of the device you want to use.

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
```

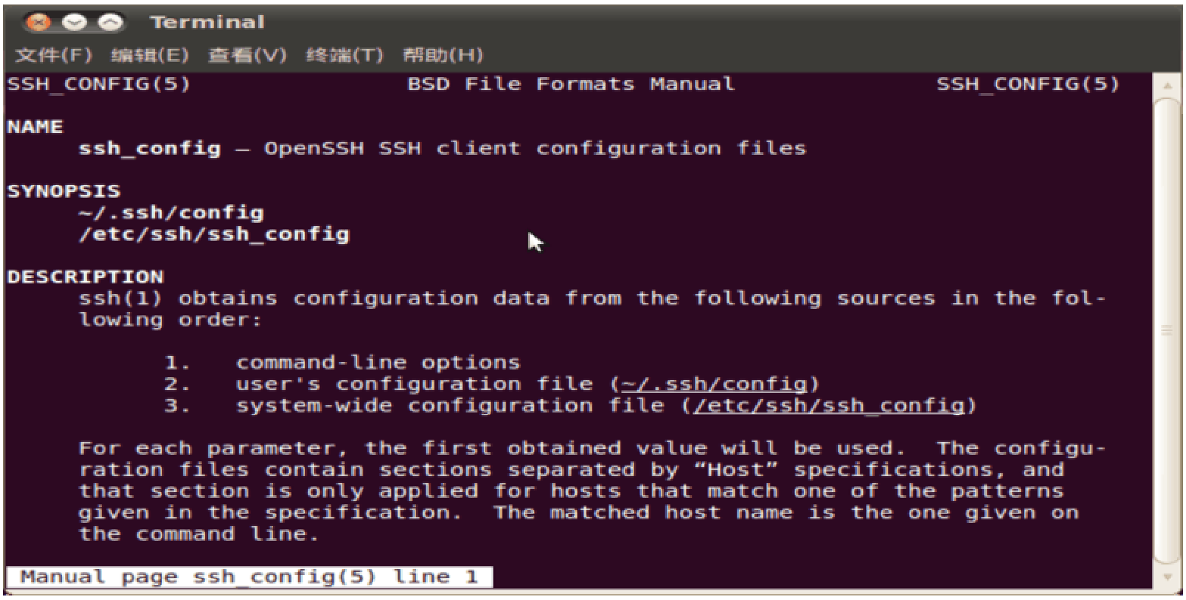
Please enter the following command in console to solve:

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

Switch Different SSH Public Keys on the Same Device

You can configure SSH according to the ssh_config documentation.

```
~$ man ssh_config
```

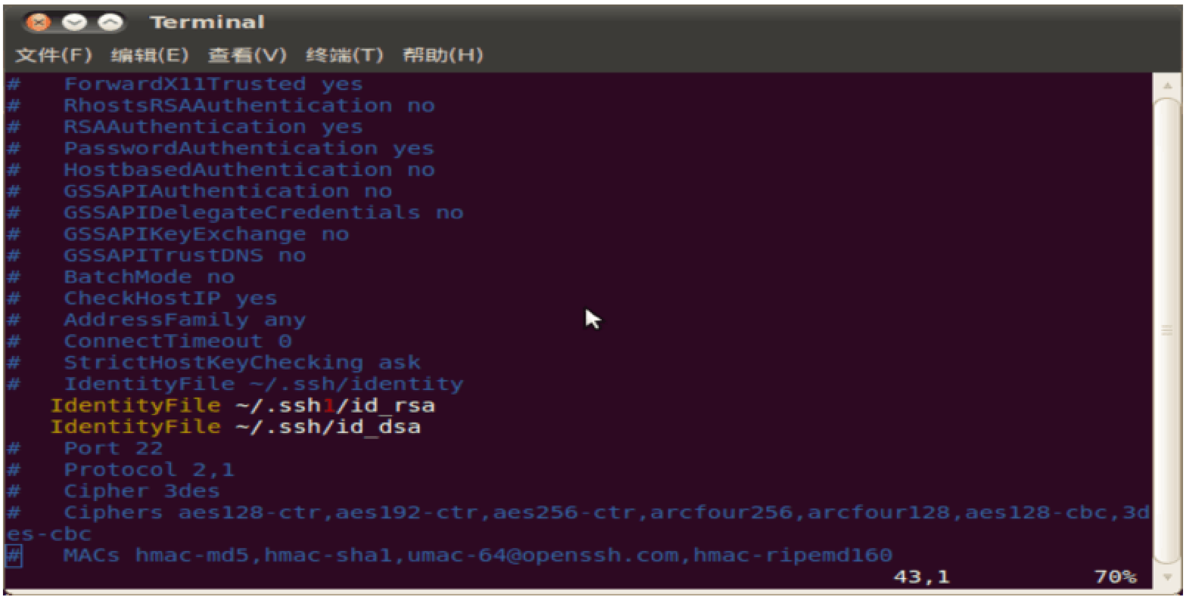


```
SSH_CONFIG(5)              BSD File Formats Manual              SSH_CONFIG(5)

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%
```

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.

Reference Documents

For more details, please refer to the document :

`<SDK>/RKDocs/Others/Rockchip_User_Guide_SDK_Application_And_Synchronization_EN.pdf`