

SC600Y&SC600T

Android Compiling

Instructions

Smart LTE Module Series

Rev. SC600&SC600T_Android_Compiling_Instructions_V1.0

Date: 2019-04-29

Status: Released

Our aim is to provide customers with timely and comprehensive service. For any assistance, please contact our company headquarters:

Quectel Wireless Solutions Co., Ltd.

7th Floor, Hongye Building, No.1801 Hongmei Road, Xuhui District, Shanghai 200233, China

Tel: +86 21 5108 6236

Email: info@quectel.com

Or our local office. For more information, please visit:

<http://www.quectel.com/support/sales.htm>

For technical support, or to report documentation errors, please visit:

<http://www.quectel.com/support/technical.htm>

Or email to: support@quectel.com

GENERAL NOTES

QUECTEL OFFERS THE INFORMATION AS A SERVICE TO ITS CUSTOMERS. THE INFORMATION PROVIDED IS BASED UPON CUSTOMERS' REQUIREMENTS. QUECTEL MAKES EVERY EFFORT TO ENSURE THE QUALITY OF THE INFORMATION IT MAKES AVAILABLE. QUECTEL DOES NOT MAKE ANY WARRANTY AS TO THE INFORMATION CONTAINED HEREIN, AND DOES NOT ACCEPT ANY LIABILITY FOR ANY INJURY, LOSS OR DAMAGE OF ANY KIND INCURRED BY USE OF OR RELIANCE UPON THE INFORMATION. ALL INFORMATION SUPPLIED HEREIN IS SUBJECT TO CHANGE WITHOUT PRIOR NOTICE.

COPYRIGHT

THE INFORMATION CONTAINED HERE IS PROPRIETARY TECHNICAL INFORMATION OF QUECTEL WIRELESS SOLUTIONS CO., LTD. TRANSMITTING, REPRODUCTION, DISSEMINATION AND EDITING OF THIS DOCUMENT AS WELL AS UTILIZATION OF THE CONTENT ARE FORBIDDEN WITHOUT PERMISSION. OFFENDERS WILL BE HELD LIABLE FOR PAYMENT OF DAMAGES. ALL RIGHTS ARE RESERVED IN THE EVENT OF A PATENT GRANT OR REGISTRATION OF A UTILITY MODEL OR DESIGN.

Copyright © Quectel Wireless Solutions Co., Ltd. 2019. All rights reserved.

About the Document

History

Revision	Date	Author	Description
1.0	2019-04-29	Hank HAN	Initial

Contents

About the Document	2
Contents	3
Figure Index	4
1 Introduction	5
2 Compiling Android on Ubuntu	6
2.1. Compiling Environment	6
2.2. How to Compile the Entire Android Software	6
3 Compiling Different Parts of Android	9

Figure Index

FIGURE 1: CHOOSE JDK8.....	7
FIGURE 2: JDK VERSION INFORMATION	7
FIGURE 3: BIN FILES GENERATED	8

1 Introduction

This document mainly provides android compiling instructions on Ubuntu for Quectel SC600Y&SC600T modules. It includes android compiling environment, how to compile the entire android software and how to compile different parts of Android on Ubuntu.

2 Compiling Android on Ubuntu

2.1. Compiling Environment

The following is an example of the android compiling environment.

CPU: Intel(R) Core(TM) i7-4790 CPU @ 3.60GHz

Memory: 8G

Hard Disk: 500G SSD

Ubuntu: Ubuntu 64bit 14.04.5 LTS

NOTES

1. The compiling environment should not be virtual machine.
2. It is recommended to use Ubuntu 64bit 14.04.5 LTS, for the compiling work on Ubuntu 64bit 14.04.5 LTS has been verified by Quectel. If other Ubuntu versions are employed, Quectel can't guarantee the compiling performances.

2.2. How to Compile the Entire Android Software

1. Use “apt-get” command to install software packages.

```
sudo apt-get install git-core gnupg flex bison gperf build-essential zip curl zlib1g-dev libc6-dev  
lib32ncurses5-dev x11proto-core-dev libx11-dev lib32z-dev libgl1-mesa-dev g++-multilib mingw32  
tofrodos python-markdown libxml2-utils xsltproc
```

2. Use “apt-get” commands to install JDK8.

```
sudo add-apt-repository ppa:openjdk-r/ppa  
sudo apt-get update  
sudo apt-get install openjdk-8-jdk
```

If you have installed JDK7 on Ubuntu before, you can use the following command to choose JDK8.

```
sudo update-alternatives --config java
```

```
hank@smart1-build:~$ sudo update-alternatives --config java
There are 2 choices for the alternative java (providing /usr/bin/java).

  Selection    Path                                                    Priority    Status
  -----
  0            /usr/lib/jvm/java-7-openjdk-amd64/jre/bin/java         1071      auto mode
  * 1          /usr/lib/jvm/java-7-openjdk-amd64/jre/bin/java         1071      manual mode
  2            /usr/lib/jvm/java-8-openjdk-amd64/jre/bin/java         1069      manual mode

Press enter to keep the current choice[*], or type selection number: 2
```

Figure 1: Choose JDK8

Use the following command to check whether JDK8 is chosen successfully. If yes, the information of JDK version will be shown as the following figure.

```
java -version
```

```
hank@smart1-build:~$ java -version
openjdk version "1.8.0_111"
OpenJDK Runtime Environment (build 1.8.0_111-8u111-b14-3~14.04.1-b14)
OpenJDK 64-Bit Server VM (build 25.111-b14, mixed mode)
hank@smart1-build:~$
```

Figure 2: JDK Version Information

3. Copy Android code to the Linux/Android directory. Then, in android directory, run the following command.

```
source build/envsetup.sh
lunch msm8953_64-userdebug
make -jn ("n" means the thread numbers of CPU)
```


4. After compiling, it will generate many BIN files in the directory of `~/work/LINUX/android/out/target/product/msm8953_64`.

```
hank@smart1-build:~/SC600_Android9_R05_r000120/out/target/product/msm8953_64$ ls
android-info.txt      fake_packages          obj                    root
boot.img              gen                    obj_arm               secimage.log
build_fingerprint.txt installed-files.json    persist               signed
build_thumbprint.txt installed-files.txt     persist.img           signed_encrypted
cache                 installed-files-vendor.json prebuilt_dtbo.img    symbols
cache.img             installed-files-vendor.txt previous_build_config.mk system
clean_steps.mk        integrity              product_copy_files_ignored.txt system.img
data                  kernel                 ramdisk.img           userdata.img
dex_bootjars          mdtp.img               ramdisk-recovery.img vbmeta.img
dlkm                  metadata               recovery               vendor
dtbo.img              metadata.img            recovery.id            vendor.img
emmc_appsboot.mbn     module-info.json       recovery.img
hank@smart1-build:~/SC600_Android9_R05_r000120/out/target/product/msm8953_64$
```

Figure 3: BIN Files Generated

3 Compiling Different Parts of Android

1. Compile about:

Input Command:

```
<make about -jn>
```

Target Folder:

```
<work/LINUX/android/out/target/product/msm8953_64>
```

Target File:

```
<emmc_appsboot.mbn>
```

2. Compile kernel:

Input Command:

```
<make bootimage -jn>
```

Target Folder:

```
<work/LINUX/android/out/target/product/msm8953_64>
```

Target File:

```
<boot.img>
```

3. Compile system:

Input Command:

```
<make systemimage -jn>
```

Target Folder:

```
<work/LINUX/android/out/target/product/msm8953_64>
```

Target File:

```
<system.img>
```

4. Compile userdata:

Input Command:

```
<make userdataimage -jn>
```

Target Folder:

<work/LINUX/android/out/target/product/msm8953_64>

Target File:

<userdata.img>

5. Compile recovery:

Input Command:

```
<make recoveryimage -jn>
```

Target Folder:

<work/LINUX/android/out/target/product/msm8953_64>

Target File:

<recovery.img>

6. Compile vendor:

Input Command:

```
<make vendorimage -jn>
```

Target Folder:

<work/LINUX/android/out/target/product/msm8953_64>

Target File:

<vendor.img>