

SC20 Android Compiling Instructions

LTE Module Series

Rev. SC20_Android_Compiling_Instructions_V1.1

Date: 2018-06-20

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://quectel.com/support/sales.htm>

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

<http://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. 2018. All rights reserved.

About the Document

History

Revision	Date	Author	Description
1.0	2016-08-29	Barnett WANG	Initial
1.1	2018-06-20	Barnett WANG	<ol style="list-style-type: none">1. Updated entire Android compiling instructions in Chapter 2.2. Added FAQs (Chapter 4).

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. Entire Android Compiling Instructions.....	6
3 Compiling Different Parts of Android	9
4 FAQs.....	11

Figure Index

FIGURE 1: JDK VERSION CHANGED	7
FIGURE 2: BIN FILES GENERATED	8

1 Introduction

This document mainly provides android compiling instructions on Ubuntu for Quectel SC20 module. It includes details such as the 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.

Ubuntu: Ubuntu 64 bit 14.04.5 LTS

CPU: Intel® Core™ i7-4790 CPU @ 3.60GHz

Memory: 8G

Hard Disk: 500G

NOTE

It is not recommended to compile Android under other Ubuntu versions or any virtual machine such as VirtualBox. When compiling Android 7, the system swap partition should be created with a minimum space of 12G.

2.2. Entire Android Compiling Instructions

1. Use apt-get command to install software packages as below:

```
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 lib32readline-gplv2-dev lib32z1-dev libgl1-mesa-dev g++-multilib mingw32 tofrodos python-markdown libxml2-utils xsltproc
```

2. Use apt-get command to install JDK:

If the Android project is Android 5 or Android 6, **openjdk7** or **openjdk8** can be used. But if the Android project is Android 7, then **openjdk8** must be used.

Install **openjdk7** command:

```
sudo apt-get install openjdk-7-jdk
```

Install **openjdk8** command:

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

Reconfigure the **openjdk** command like below:

```
update-alternatives --config java
update-alternatives --config javac
update-alternatives --config javap
update-alternatives --config javah
update-alternatives --config javadoc
```

```
barnett@barnett-OptiPlex-5040:~/work/SC20_Android7.1_R06_r00049.1$ update-alternatives --config java
There are 3 choices for the alternative java (providing /usr/bin/java).

  Selection    Path
-----
  0            /usr/lib/jvm/java-7-openjdk-amd64/jre/bin/java    1071    auto mode
  1            /usr/java/jdk1.6.0_45/bin/java                    300     manual mode
  2            /usr/lib/jvm/java-7-openjdk-amd64/jre/bin/java    1071    manual mode
* 3            /usr/lib/jvm/java-8-openjdk-amd64/jre/bin/java    1069    manual mode

Press enter to keep the current choice[*], or type selection number: 3
barnett@barnett-OptiPlex-5040:~/work/SC20_Android7.1_R06_r00049.1$ update-alternatives --config javac
There are 3 choices for the alternative javac (providing /usr/bin/javac).

  Selection    Path
-----
  0            /usr/lib/jvm/java-7-openjdk-amd64/bin/javac      1071    auto mode
  1            /usr/java/jdk1.6.0_45/bin/javac                  300     manual mode
  2            /usr/lib/jvm/java-7-openjdk-amd64/bin/javac      1071    manual mode
* 3            /usr/lib/jvm/java-8-openjdk-amd64/bin/javac      1069    manual mode

Press enter to keep the current choice[*], or type selection number:
barnett@barnett-OptiPlex-5040:~/work/SC20_Android7.1_R06_r00049.1$ update-alternatives --config javap
There are 3 choices for the alternative javap (providing /usr/bin/javap).

  Selection    Path
-----
  0            /usr/lib/jvm/java-7-openjdk-amd64/bin/javap      1071    auto mode
  1            /usr/java/jdk1.6.0_45/bin/javap                  300     manual mode
  2            /usr/lib/jvm/java-7-openjdk-amd64/bin/javap      1071    manual mode
* 3            /usr/lib/jvm/java-8-openjdk-amd64/bin/javap      1069    manual mode

Press enter to keep the current choice[*], or type selection number:
barnett@barnett-OptiPlex-5040:~/work/SC20_Android7.1_R06_r00049.1$ update-alternatives --config javah
There are 3 choices for the alternative javah (providing /usr/bin/javah).

  Selection    Path
-----
  0            /usr/lib/jvm/java-7-openjdk-amd64/bin/javah      1071    auto mode
  1            /usr/java/jdk1.6.0_45/bin/javah                  300     manual mode
  2            /usr/lib/jvm/java-7-openjdk-amd64/bin/javah      1071    manual mode
* 3            /usr/lib/jvm/java-8-openjdk-amd64/bin/javah      1069    manual mode

Press enter to keep the current choice[*], or type selection number:
barnett@barnett-OptiPlex-5040:~/work/SC20_Android7.1_R06_r00049.1$ update-alternatives --config javadoc
There are 3 choices for the alternative javadoc (providing /usr/bin/javadoc).
```

Figure 1: JDK Version Changed

3. Run the following command in the project directory:

```
source build/envsetup.sh  
lunch msm8909-userdebug  
make -jn ("n" means the thread numbers of CPU)
```

4. After compiling, many BIN files will be generated in the following directory:
~/work/LINUX/android/out/target/product/msm8909



```
barnett@barnett-OptiPlex-5040:~/work/msm8909_1.2.3-10210/out/target/product/msm8909$ ls  
2k_nand_images      data                OTA_Target_Files   root  
android-info.txt   dt.img             ota_temp           secimage.log  
boot.img           emmc_appsboot.mbn  ota.zip            signed  
cache              fake_packages      package_backup.zip symbols  
cache.img          filesmap           persist            system  
ChinaMobile.ota.zip gen                persist.img        system.img  
ChinaTelecom.ota.zip installed-files.txt previous_build_config.mk  
ChinaUnicom.ota.zip kernel            ramdisk.img        userdata.img  
clean_steps.mk     msm8909-ota-eng.barnett.zip ramdisk-recovery.img  
CmccPower.ota.zip  obj               recovery           recovery  
CTA.ota.zip        OTA_Binary_Packs  recovery.img
```

Figure 2: 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/msm8909/
```

Target File:

```
emmc_appsboot.mbn
```

2. Compile kernel:

Input Command:

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

Target Folder:

```
work/LINUX/android/out/target/product/msm8909
```

Target File:

```
boot.img
```

3. Compile system:

Input Command:

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

Target Folder:

```
work/LINUX/android/out/target/product/msm8909
```

Target File:

```
system.img
```

4. Compile userdata:

Input Command:

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

Target Folder:

work/LINUX/android/out/target/product/msm8909

Target File:

userdata.img

5. Compile recovery:

Input Command:

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

Target Folder:

work/LINUX/android/out/target/product/msm8909

Target File:

recovery.img

4 FAQs

Q: When the Jack server cannot boot, how to restart it?

A: Please restart the server via the command below:

```
./prebuilts/sdk/tools/jack-admin kill-server  
./prebuilts/sdk/tools/jack-admin start-server
```

Q: How to reconfigure the memory?

A: Please reconfigure the memory with the following command:

```
export JACK_SERVER_VM_ARGUMENTS="-Dfile.encoding=UTF-8 -XX:+TieredCompilation  
-Xmx4096m"
```

Q: Which Android code needs Jack server?

A: Android 7 compiling needs Jack server.