SC60 Android Compiling Instructions

Smart LTE Module Series

Rev. SC60_Android_Compiling_Instructions_V1.0

Date: 2017-10-12

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

About the Document

History

Revision	Date	Author	Description
1.0	2017-10-12	Hank HAN	Initial

Contents

Figure Index

1 Introduction

This document mainly provides android compiling instructions on Ubuntu for Quectel SC60 module. It includes details such as the android compiling environment, how to compile the entire android software, how to compile Android 7.1 for multi-users 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 SSDs
Ubuntu: Ubuntu 64bit 14.04.5 LTS

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 command below 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

Then 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 Linux/android directory. Then, in android directory, run command below.**

```
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 directory of *~/work/LINUX/android/out/target/product/msm8953_64*.**

```
hank@smart1-build:~/SC60_30_36_Android7.1_R02/out/target/product/msm8953_64$ ls
android-info.txt       installed-files.json  oem4.img                  recovery.img
boot.img               installed-files.txt   OTA_Binary_Packs          root
build_fingerprint.txt  integrity             OTA_Target_Files          secimage.log
cache                  kernel                ota.zip                   signed
cache.img              mdtp.img              persist                   signed_encrypted
clean_steps.mk         module-info.json      persist.img               symbols
data                   obj                   previous_build_config.mk  system
dex_bootjars           obj_arm               ramdisk.img               system.img
emmc_appsboot.mbn      oem                   ramdisk-recovery.img      userdata.img
fake_packages          oem2.img              recovery
gen                    oem3.img              recovery.id
hank@smart1-build:~/SC60_30_36_Android7.1_R02/out/target/product/msm8953_64$
```

Figure 3: BIN Files Generated

2.3. How to Compile Android 7.1 for Multi-users

Compiling Android 7.1 on one Ubuntu for multi-users simultaneously is not supported by default. But if you have a requirement for this function, the following steps can be performed.

1. **After the user has complied Android 7.1, the file *.jack-settings* and the folder *.jack-server* would be created in directory ~/.**

2. **Modify the port numbers in files *.jack-settings and .jack-server/config.properties*.**

Use the following command to open the file *.jack-settings* and the content of the file is as following figure. Then modify the port numbers of “SERVER_PORT_SERVICE” and “SERVER_PORT_ADMIN”.

```
vim ~/.jack-settings
```

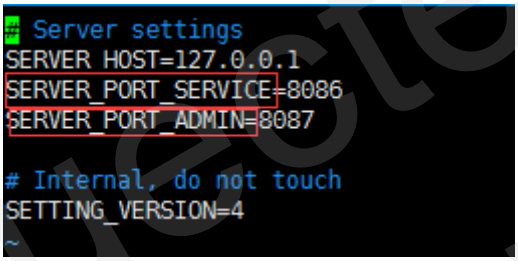

Figure 4: Content of .jack-settings

Similarly, use the following command to open the file *.jack-server/config.properties* and the content of the file is as following figure. Then modify the port numbers of “jack.server.service.port” and “jack.server.admin.port”.

```
vim ~/.jack-server/config.properties
```

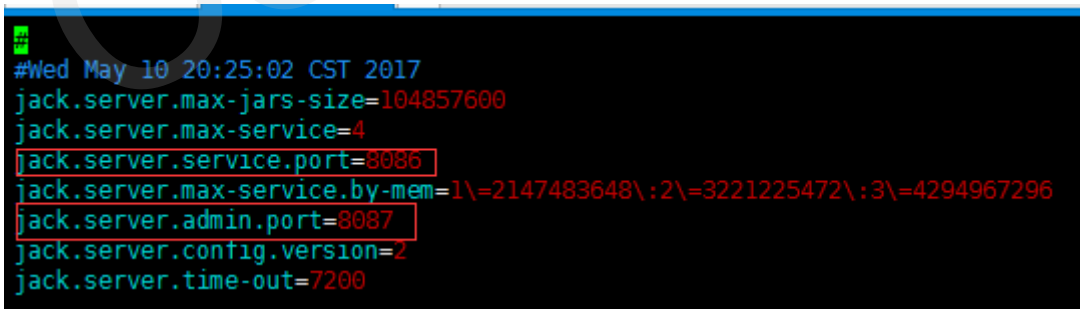

Figure 5: Content of .jack-server/config.properties

Please note that the modification of the four port numbers should meet the following criteria.

1) The range of the port numbers is 0~65536. Port numbers 0~1024 are not recommended because they are the commonly used numbers for users and are easily occupied by the server.
2) The port numbers of “SERVER_PORT_SERVICE” and “jack.server.service.port” should be the same and should be different from the default value 8076. The port numbers of “SERVER_PORT_ADMIN” and “jack.server.admin.port” should be the same and should be different from the default value 8077.
3) The port numbers that each user uses should be different.

3. Reboot Jack.

Enter into the directory of your Android 7.1 code, and then use the following two commands to reboot Jack (Java Android Compiler Kit).

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

3 Compiling Different Parts of Android

1. Compile aboot:

Input Command:
<make aboot –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>