EC2X&AG35-QuecOpen

Quick Start

LTE Module Series

Rev. EC2X&AG35-QuecOpen_Quick_Start_V1.4

Date: 2018-03-16

Status: Temporary

www.quectel.com

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	2017-11-25	Running QIAN/ Gale GAO	Initial
1.1	2018-02-01	Running QIAN	Add FAQ Chapter
1.2	2018-02-10	Running QIAN	Add auto start app
1.3	2018-02-23	Gale GAO	1. Add the selection and modification of kernel configuration file 2. Add the production of usrdata.ubi
1.4	2018-03-16	Running QIAN	1. Change this document name 2. Add Chapter 1-2 3. Add the production of debug version

Contents

Table Index

Figure Index

1 Introduction

This document mainly introduces the software development process, the development environment, the basic methods and steps of Linux development and debugging, as well as the method of module startup inspection, which is about the EC2x&AG35-QuecOpen module of Quectel.

This document mainly applies to the following modules of Quectel:

- EC2x-QuecOpen

In this document, EC2x includes EC25/EC21/EC20 R2.1/EC20 R2.0
EC2x-QuecOpen, includes EC25/EC21/EC20 R2.1/EC20 R2.0-QuecOpen

- AG35-QuecOpen

2 Document Reading Instruction

2.1. QuecOpen Product Introduction

To understand the technical architecture and hardware and software resources of the product, please refer to following documents:

- ***Quectel_EC2x-QuecOpen_ Technology_And_Resources_Overview***
- ***Quectel_AG35-QuecOpen_ technology_And_Resources_Overview***

2.2. Hardware Development

(1) Functional PIN Selection

Please refer to following documents about functional requirements and pin resource definitions:

- Quectel_EC2x-QuecOpen_GPIO_Assignment_Speadsheet
- Quectel_AG35-QuecOpen_GPIO_Assignment_Speadsheet

(2) SCH and PCB Design

Table 1：QuecOpen Project Hardware Design Reference Document

	Reference Documents
Hardware Design Guide	*Quectel_EC2x-QuecOpen_Hardware_Design* *Quectel_AG35-QuecOpen_Hardware_Design*
Module Schematic Diagram Design	*EC2x-TE-A_SCH* *AG35-TE-A_SCH*
EVB Board Schematic	*LTE-OPEN-EVB_SCH*

(3) During the design process referring to the above steps, if you have any doubts, please contact Quectel technical support personnel.

2.3. Software development

After reading this document directly, read the corresponding functional documents under the software development folder.

3 Introduction of QuecOpen Software Development

3.1. Requirements for Developers

(1) Familiar with standard GNU/Linux application development, and common Linux system commands;
(2) Grasp the basic knowledge of some driving and network protocols.
(3) Understand some AT command knowledge and refer to the Quectel at command manual：
- Quectel_AG35_AT_Commands_Manual
- Quectel_EC2x_AT_Commands_Manual

3.2. QuecOpen Development Process

(1) Ubuntu1404 or 1604 system, 4GB memory or above, 4 core CPU or above. If using virtual machine, the memory is allocated to the virtual machine not less than 4GB.
(2) Install development tools, drivers and SDK as the **Chapter 4** listed.
(3) Get familiar with the development process of QuecOpen SDK by writing a simple APP according **Chapter 5.1**
(4) Reimport the customer APP into the root file system according to the **Chapter 5.4** and regenerate the file system image;
(5) Advance development can refer to other relevant documents

4 Development Environment Preparation

4.1. Install Required Tools

Ubuntu USB driver installation and burning tool installation please refer to below document:
Quectel_WCDMA&LTE_Linux_USB_Driver_User_Guide

4.2. Install ADB

4.2.1. Install ADB Driver

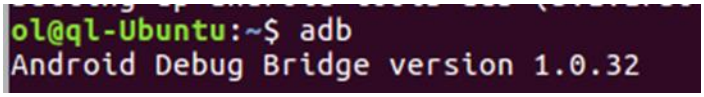
Run the following command to install the ADB driver:

```
sudo apt-get update
sudo apt-get install android-tools-adb
```

If above command fails, please try following ones:

```
sudo add-apt-repository ppa:nilarimogard/webupd8
sudo apt-get update
sudo apt-get install android-tools-adb
```

After the installation is successful, the display is as follows:



4.2.2. Add Module USB VID

Query device VID：

```
lsusb
```

Modify the configuration file：

```
sudo vi .android/adb_usb.ini
```

4.2.3. Enumerate Equipment

Run the following command to enumerate equipment:

```
sudo adb kill-server
sudo adb devices
```

4.3. Firmware Update

When we get the latest firmware and SDK, we need update the device first, by this way, we can verify the installation of the 2.1 tools, as well as the upgrade of the device.

About how to update firmware please refer to introduction of ***KBA_QuecOpen_Download_Guide***

4.4. Download One File to the Module

This chapter introduces how to download one common file or one App to the Linux file system.

4.4.1. Using ADB

Command basic format：

```
sudo adb push <local path> <module path>
```

for example:

```
adb push ~/ql-ol-sdk/ql-ol-extsdk/example/helloWorld/hellolworld /usrdata
```

4.4.2. Using Serial Port

The following figures are generated by secureCRT in Windows when choose Zmodem to send file.

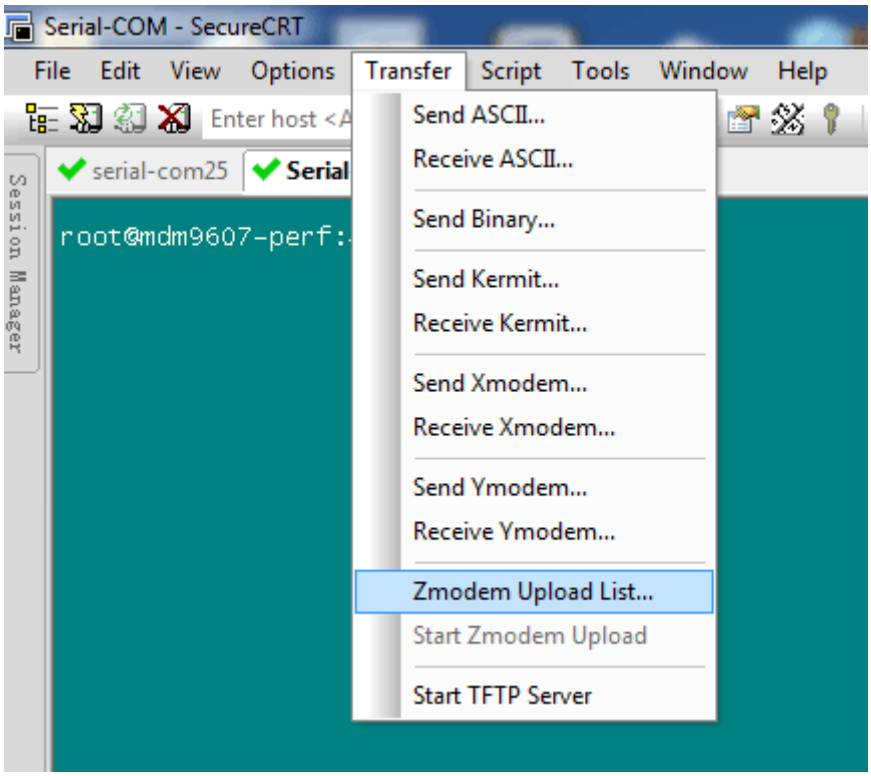


Figure 1：Use SecureCRT Tool for File Transfer

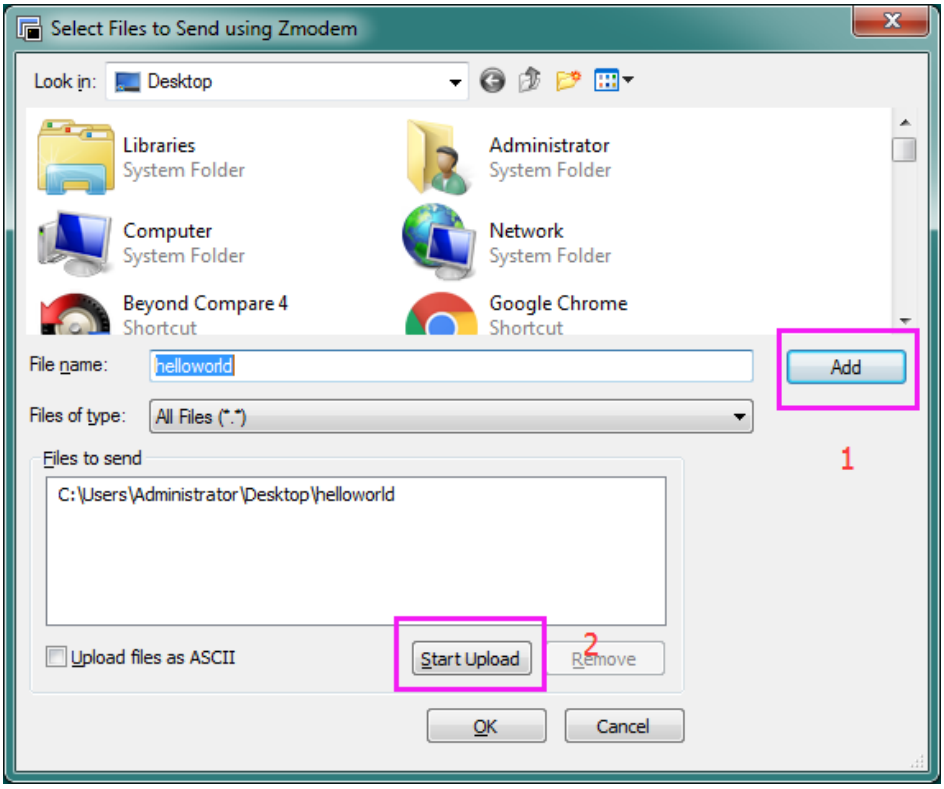


Figure 2：Steps for uploading files based on Zmodem

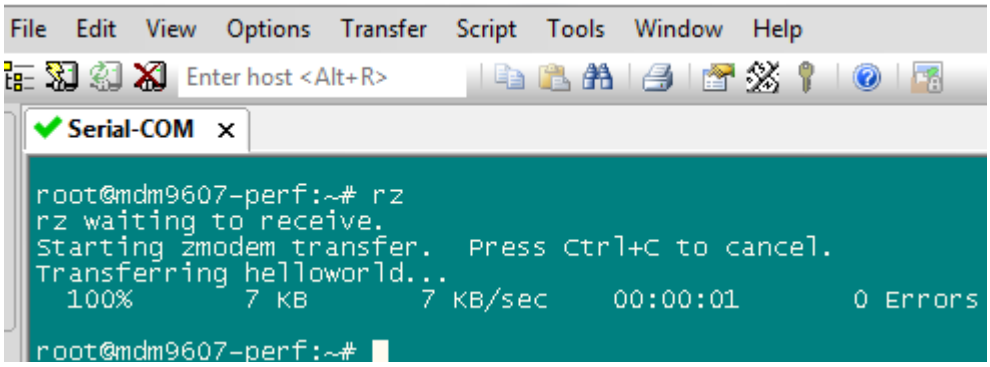


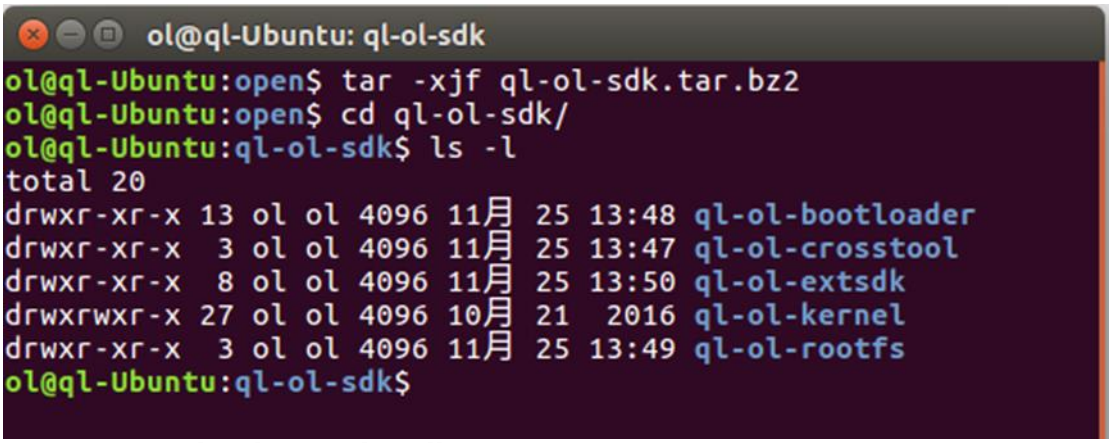
Figure 3：File uploaded successfully

4.5. SDK Installation

4.5.1. Unzip SDK File

SDKl.zip unzipped process must be done under non-root Ubuntu environment.

```
tar -jxvf ql-ol-sdk.tar.bz2
```



4.5.2. Introduction of the Files

Table 2：Introduction of the files in SDK

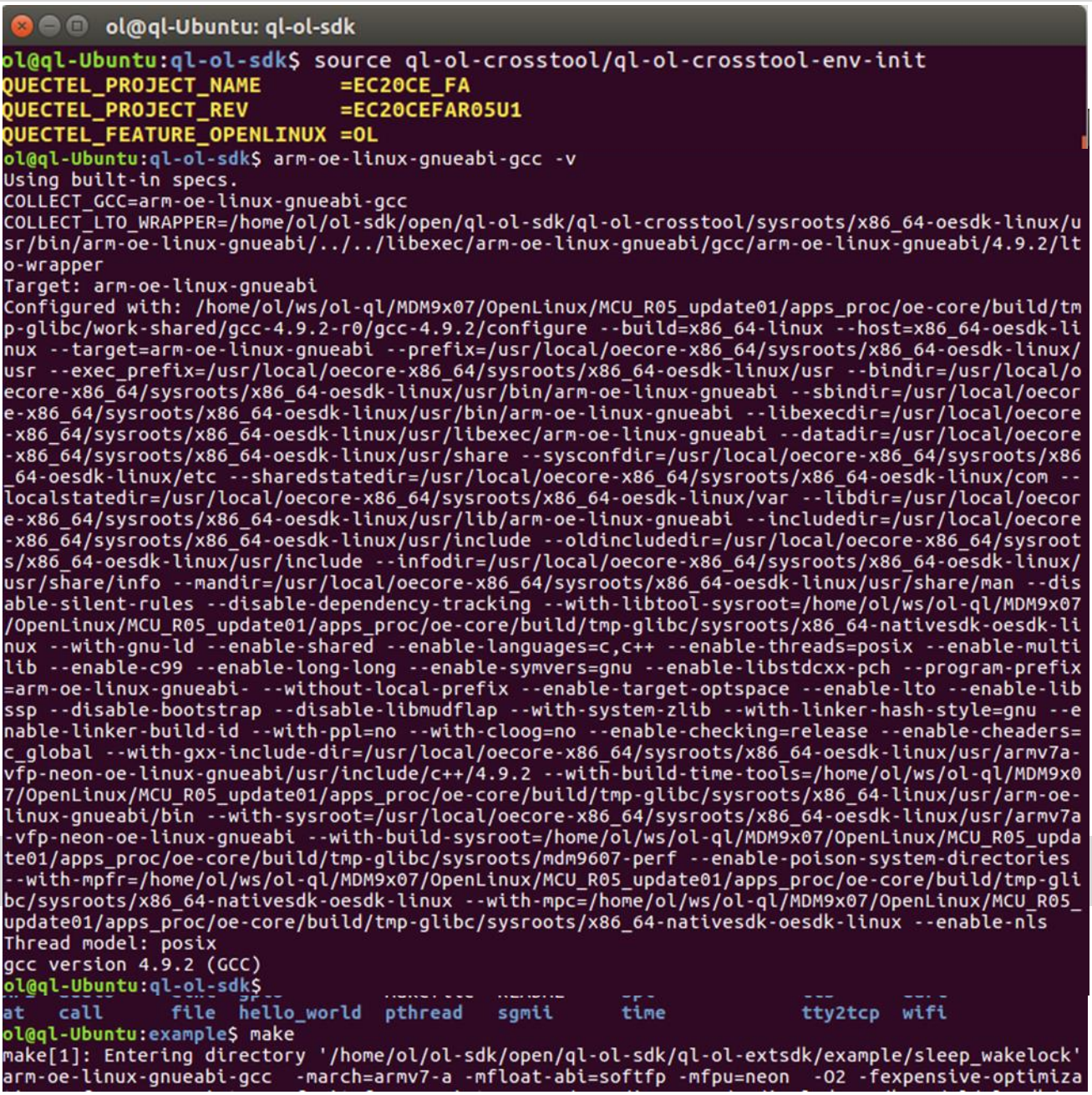
Directory	Content
ql-ol-crosstool	Cross tool chain
ql-ol-bootloader	QTI bootloader source code (Only available as customer specified)

ql-ol-kernel	Linux kernel source code (Only available as customer specified)
ql-ol-rootfs	Root file system for platform runtime
ql-ol-extsdk	Include API，example and tools

4.5.3. Execute

Running the following command to install SDK：

```
cd ql-ol-sdk
source ql-ol-crosstool/ql-ol-crosstool-env-init
```




```
cd hello_world
```

make

```
ol@ql-Ubuntu:example$ cd hello_world/
ol@ql-Ubuntu:hello_world$ make clean
rm -rf helloworld *.o
ol@ql-Ubuntu:hello_world$ make
arm-oe-linux-gnueabi-gcc  -march=armv7-a -mfloat-abi=softfp -mfpu=neon  -O2 -fexpensive-optimiza
tions -frename-registers -fomit-frame-pointer -I./ -I/mdm9607/usr/include -I/home/ol/ol-sdk/open
/ql-ol-sdk/ql-ol-extsdk/example/hello_world/../../include  -c helloworld.c
arm-oe-linux-gnueabi-gcc  -march=armv7-a -mfloat-abi=softfp -mfpu=neon -L./ -L/home/ol/ol-sdk/op
en/ql-ol-sdk/ql-ol-extsdk/example/hello_world/../../lib -lrt   helloworld.o -o helloworld
ol@ql-Ubuntu:hello_world$
```

5 Linux Develop and Debug

You must execute “source ql-ol-crosstool/ql-ol-crosstool-env-init” first of all!
This is just the beginning…

5.1. Linux APP Development

QuecOpen Linux standard APP development is same as traditional embedded ARM-Linux development process, the requirements for customers, so long as there are basic Linux application development experience.

The next section introduces Helloworld's creation to single-path dialing and guides customers through the QuecOpen Linux development process.

5.1.1. Helloworld

(1) Create workspace

Here we create ws folder for example.

(2) Copy Demo

Copy ql-ol-sdk/ql-ol-extsdk/example/hello_world to ws directory

(3) Build

```
ol@ql-Ubuntu: hello_world
ol@ql-Ubuntu:open$ ls
ql-ol-sdk  ql-ol-sdk.tar.bz2
ol@ql-Ubuntu:open$ source ql-ol-sdk/ql-ol-crosstool/ql-ol-crosstool-env-init
QUECTEL_PROJECT_NAME      =EC20CE_FA
QUECTEL_PROJECT_REV       =EC20CEFAR05U1
QUECTEL_FEATURE_OPENLINUX =OL
ol@ql-Ubuntu:open$ mkdir ws
ol@ql-Ubuntu:open$ cd ws/
ol@ql-Ubuntu:ws$ cp -rf ../ql-ol-sdk/ql-ol-extsdk/example/hello_world ./
ol@ql-Ubuntu:ws$ ls
hello_world
ol@ql-Ubuntu:ws$ cd hello_world/
ol@ql-Ubuntu:hello_world$ make
arm-oe-linux-gnueabi-gcc  -march=armv7-a -mfloat-abi=softfp -mfpu=neon  -O2 -fex
pensive-optimizations -frename-registers -fomit-frame-pointer -I./ -I/mdm9607/us
r/include -I/home/ol/ol-sdk/open/ws/hello_world/../../include  -c helloworld.c
arm-oe-linux-gnueabi-gcc  -march=armv7-a -mfloat-abi=softfp -mfpu=neon -L./ -L/h
ome/ol/ol-sdk/open/ws/hello_world/../../lib -lrt   helloworld.o -o helloworld
ol@ql-Ubuntu:hello_world$
```

- Modify the file permissions to be executable
- Run helloworld

5.1.2. Single APN Data Call

(1) Copy Demo

Copy ql-ol-sdk/ql-ol-extsdk/example/data

(2) Compile

(3) Download and run

After dial successfully, can check via following command:

```
root@mdm9607-perf:~# ifconfig
bridge0   Link encap:Ethernet  HWaddr 5E:DF:73:B8:09:F2
          inet addr:192.168.225.1  Bcast:192.168.225.255  Mask:255.255.255.0
          inet6 addr: fe80::5cdf:73ff:feb5:6ef/64 Scope:Link
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:1 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 B)  TX bytes:76 (76.0 B)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

rmnet0    Link encap:UNSPEC  HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
          UP RUNNING  MTU:2000  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:5 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:388 (388.0 B)

rmnet_data0 Link encap:UNSPEC  HWaddr 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00
          inet addr:10.100.233.73  Mask:255.255.255.252
          inet6 addr: fe80::3e18:9f60:8eea:469b/64 Scope:Link
          UP RUNNING  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:5 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:388 (388.0 B)

root@mdm9607-perf:~#
```

older.

5.1.4. Add self - starting APP

Copy *ql-ol-extsdk/tools/quectel_ubi/QuecOpen_startapp* to *ql-ol-rootfs/etc/init.d/*, and modify the APP path corresponding to the variable AppProgram in the file.

```
# here user can indicate the user app path
AppProgram=/data/QuecOpen/helloworld

case "$1" in
        start)
                if ! [ -f $AppProgram ]
                then
                    echo "### QuecOpen Application doesn't exist ###"
                    exit 3
                fi
                echo -n ">>> Starting QuecOpen application: "
#                start-stop-daemon -S -b -a $AppProgram
                $AppProgram &
                echo "done"
```

```
cd ql-ol-rootfs/etc/rc5.d
ln -vsf ../init.d/QuecOpen_startapp S45QuecOpen_startapp
```

5.2. Bootloader Development

```
make aboot          // Build bootloader, and generate appsboot.mbn   in target/ folder of current path;
```

```
gale@eve-linux02:~/MDM9x07/SDK_FAG1127/ql-ol-sdk$ make aboot
cd /home/gale/MDM9x07/SDK_FAG1127/ql-ol-sdk/ql-ol-bootloader ; make -j 4 mdm9607 TOO
        cp build-mdm9607/appsboot.mbn /home/gale/MDM9x07/SDK_FAG1127/ql-ol-sdk/targe
make[1]: Entering directory `/home/gale/MDM9x07/SDK_FAG1127/ql-ol-sdk/ql-ol-bootloac
including app/aboot dev/keys dev/pmic/pm8x41 dev/vib lib/debug lib/heap lib/libc lib
```

```
make aboot/clean      // Clean up mid-file of last command
```

```
gale@eve-linux02:~/MDM9x07/SDK_FAG1127/ql-ol-sdk$ make aboot/clean
cd /home/gale/MDM9x07/SDK_FAG1127/ql-ol-sdk/ql-ol-bootloader ; rm -rf build-mdm9607
rm -rf target/appsboot.mbn
gale@eve-linux02:~/MDM9x07/SDK_FAG1127/ql-ol-sdk$
```

5.3. Kernel Development

First need to specify and modify the kernel options file, use following commands to specify ql-ol-kernel/arch/arm/configs/mdm9607-perf_defconfig:

```
make kernel_menuconfig
```

```
ql-ol-sdk$ make kernel_menuconfig
-sdk/ql-ol-kernel ; make ARCH=arm mdm9607-perf_defconfig menuconfig O=build ;
/MDM9x07/SDK_FAG0130/ql-ol-sdk/ql-ol-kernel'
/MDM9x07/SDK_FAG0130/ql-ol-sdk/ql-ol-kernel/build'

warning: defaults for choice values not supported
hoice value used outside its choice group
```

```
                    Linux/arm 3.18.20 Kernel Configuration
 <Enter> selects submenus ---> (or empty submenus ----).  Highlighted letters are hotkeys.  Pressin
for Help, </> for Search.  Legend: [*] built-in  [ ] excluded  <M> module  < > module capable

                    [ ] Provide early_ioremap() support for kernel initialization
                    [*] Patch physical to virtual translations at runtime
                        General setup  --->
                    [*] Enable loadable module support  --->
                    [*] Enable the block layer  --->
                        System Type  --->
                        Bus support  --->
                        Kernel Features  --->
                        Boot options  --->
                        CPU Power Management  --->
                        Floating point emulation  --->
                        Userspace binary formats  --->
                        Power management options  --->
                    [*] Networking support  --->
                        Device Drivers  --->
                        File systems  --->
                        Kernel hacking  --->
                        Security options  --->
                    -*- Cryptographic API  --->
                        Library routines  --->
                    [ ] Virtualization  ----
                        Quectel global configurations  --->

                        <Select>    < Exit >    < Help >    < Save >    < Load >
```

The kernel options can be modified as needed, and will withdrawing directly if without modification, at this time will generate hidden files in ql-ol-kernel/build/.config, it is the kernel configuration file which making kernel process depended on. After this modification, if users confirm .config file need be saved and submitted to code library, the following commands are required to make previous modification saved in ql-ol-kernel/arch/arm/configs/mdm9607-perf_defconfig. (If users use git to maintain the code, need to keep track of this hidden file instead of tracing .config.)

cp ql-ol-kernel/build/.config ql-ol-kernel/arch/arm/configs/mdm9607-perf_defconfig

```
ql-ol-sdk$ cp ql-ol-kernel/build/.config ql-ol-kernel/arch/arm/configs/mdm9607-perf_defconfig
ql-ol-sdk$
```

make kernel // build and boot.img generated in the target / of current path

```
gale@eve-linux02:~/MDM9x07/SDK_FAG1127/ql-ol-sdk$ make kernel
cd /home/gale/MDM9x07/SDK_FAG1127/ql-ol-sdk/ql-ol-kernel ; make ARCH=arm mdm9607
        make ARCH=arm CC=arm-oe-linux-gnueabi-gcc LD=arm-oe-linux-gnueabi-ld.bfd
        cp build/arch/arm/boot/zImage build/arch/arm/boot/dts/qcom/mdm9607-mtp.d
make[1]: Entering directory `/home/gale/MDM9x07/SDK_FAG1127/ql-ol-sdk/ql-ol-kern
```

make kernel/clean //Clean up mid_file of last command;

```
gale@eve-linux02:~/MDM9x07/SDK_FAG1127/ql-ol-sdk$ make kernel/clean
cd /home/gale/MDM9x07/SDK_FAG1127/ql-ol-sdk/ql-ol-kernel ; make distclean || exit
make[1]: Entering directory `/home/gale/MDM9x07/SDK_FAG1127/ql-ol-sdk/ql-ol-kerne
```

make kernel_module // Compile kernel module (（Execute if the related kmod code is modified），automatically installed to the rootfs directory，remaking sysfs.ubi is required

```
gale@eve-linux02:~/MDM9x07/SDK_FAG1127/ql-ol-sdk$ make kernel_module
cd /home/gale/MDM9x07/SDK_FAG1127/ql-ol-sdk/ql-ol-kernel ; make modules ARCH=arm CC
        make ARCH=arm CROSS_COMPILE=arm-oe-linux-gnueabi- O=/home/gale/MDM9x07/SDK_
make[1]: Entering directory `/home/gale/MDM9x07/SDK_FAG1127/ql-ol-sdk/ql-ol-kernel'
make[2]: Entering directory `/home/gale/MDM9x07/SDK_FAG1127/ql-ol-sdk/ql-ol-kernel/
  CHK     include/config/kernel.release
```

5.4. Make System File

make rootfs //Generate *sysfs.ubi* in target/ of current path

```
gale@eve-linux02:~/MDM9x07/SDK_FAG1127/ql-ol-sdk$ make rootfs
cd /home/gale/MDM9x07/SDK_FAG1127/ql-ol-sdk ; chmod +x ./ql-ol-extsdk/tools/quectel_
        ./ql-ol-extsdk/tools/quectel_ubi/ubinize  -o mdm9607-perf-sysfs.ubi -m 4096
        mv mdm9607-perf-sysfs.ubifs mdm9607-perf-sysfs.ubi target/
ubinize: volume size was not specified in section "ubifs", assume minimum to fit ima
```

make rootfs/clean //Clean up mid-file of last command

```
gale@eve-linux02:~/MDM9x07/SDK_FAG1127/ql-ol-sdk$ make rootfs/clean
rm -rf target/*.ubi*
gale@eve-linux02:~/MDM9x07/SDK_FAG1127/ql-ol-sdk$
```

5.5. Make usrdata.ubi

There is usr.data partition in Flash by defualy, can be used to store user files and DFOTA upgrade.

make usrdata //Generate *usrdata.ubi*

```
ql-ol-sdk$ make usrdata
-sdk ; chmod +x ./ql-ol-extsdk/tools/quectel_ubi/* ; ./ql-ol-extsdk/t

i/ubinize  -o usrdata.ubi -m 4096 -p 256KiB -s 4096 ql-ol-extsdk/tool
arget/
```

make usrdata/clean //Clean up mid-file of last command

```
ql-ol-sdk$ make usrdata/clean

ql-ol-sdk$
```

5.6. One key Compilation

QuecOpen SDK provides one key compilation to build all of aboot, kernel, kernel_module, rootfs.

make	// Build all and put the output to target/ folder.

```
gale@eve-linux02:~/MDM9x07/SDK_FAG1127/ql-ol-sdk$ ls target/
appsboot.mbn  mdm9607-perf-boot.img  mdm9607-perf-sysfs.ubi  mdm9607-perf-sysfs.ubifs
gale@eve-linux02:~/MDM9x07/SDK_FAG1127/ql-ol-sdk$
```

make clean	//Clean up mid-file of last command.

```
gale@eve-linux02:~/MDM9x07/SDK_FAG1127/ql-ol-sdk$ make clean
cd /home/gale/MDM9x07/SDK_FAG1127/ql-ol-sdk/ql-ol-bootloader ; rm -rf build-mdm9607
rm -rf target/appsboot.mbn
cd /home/gale/MDM9x07/SDK_FAG1127/ql-ol-sdk/ql-ol-kernel ; make distclean || exit
make[1]: Entering directory `/home/gale/MDM9x07/SDK_FAG1127/ql-ol-sdk/ql-ol-kernel'
```

Copy all the files in the target folder to the upgrade package and download them to the module.

5.7. Compile Debug Firmware Version

When encounter problems need to open the kernel debug Log, need compile Debug version, compilation method is as follows:

(1) Configuration

make debug_kernel_menuconfig

(2) Compile file system and kernel.

make debug_version

```
will@will-OptiPlex-790:/home/sdc/jackson/MCU_06
mdm9607-perf-boot.img  mdm9607-perf-sysfs.ubi
```

Re-burn the generated file system and the kernel image.

6 Module Startup Check

This chapter introduces how to check whether system run up successfully by using AT command. Specific steps are as follows:

(1) Connect PC with the module's MAIN UART port or USB AT port via the USB turn serial port cable;
(2) Insert (U)SIM card and connect antenna, then power on;
(3) Send following AT command via QCOM tool.
- Check AT port connected or not: **AT**
- Detect (U)SIM card: **AT+CPIN?**
- Check signal strength: **AT+CSQ**
- Detect module register network: **AT+CGREG?**
- Query Operator: **AT+COPS?**
- Query network standard: **AT+QNWINFO**
- Voice test: **ATDxxx**

The following picture shows the startup check when insert a general China Mobile SIM card.

```
at
OK
at+cpin?
+CPIN: READY

OK
at+csq
+CSQ: 15,99

OK
at+cgreg?
+CGREG: 0,1

OK
at+cops?
+COPS: 0,0,"CHINA MOBILE",7

OK
at+qnwinfo
+QNWINFO: "TDD LTE","46000","LTE BAND 40",38950

OK
atd1[redacted];
OK
```

7 FAQ

1. **Why the compression package must be decompressed in a ordinary users environment?**
   Query the tar command manual can see:

```
--same-owner
      try extracting files with the same ownership as exists in the ar-
      chive (default for superuser)

--no-same-owner
      extract files as yourself (default for ordinary users)
```