

MDM9x07&MDM9628

OpenLinux 软件开发使用指南

About the Document

本文档适用于 MDM9628 和 MDM9X07 平台

History

| Revision | Date | Author | Description |
|----------|------------|-----------------|--------------------|
| 1.0 | 2017-11-25 | Running Gale | Initial |
| 1.1 | 2018-02-01 | Running | Add fax chapter |
| 1.2 | 2018-02-10 | Running | Add auto start app |

目录

| | |
|---|----|
| MDM9x07&MDM9628 OpenLinux 软件开发使用指南 | 0 |
| About the Document..... | 1 |
| 1. 开发流程介绍..... | 3 |
| 1.1 对开发者的要求..... | 3 |
| 1.2 QuecOpen 开发过程..... | 3 |
| 2. 开发环境准备..... | 4 |
| 2.1 工具安装..... | 4 |
| 2.2 ADB 安装..... | 4 |
| 2.3 固件更新..... | 4 |
| 2.5 文件下载..... | 4 |
| 2.5.1 基于 ADB..... | 5 |
| 2.5.2 基于串口..... | 5 |
| 2.6 SDK 安装..... | 6 |
| 3. Linux 开发与调试..... | 8 |
| 3.1 Linux APP 开发..... | 8 |
| 3.1.1 Helloworld..... | 8 |
| 3.1.2 单路拨号..... | 9 |
| 3.1.3 高级应用开发..... | 9 |
| 3.1.4 添加自启动 APP..... | 9 |
| 3.2 Bootloader 开发..... | 10 |
| 3.3 Kernel 开发..... | 10 |
| 3.4 文件系统制作..... | 11 |
| 3.5 一键化编译..... | 11 |
| 4. 模块开机检查..... | 13 |
| FAQ..... | 14 |

1. 开发流程介绍

1.1 对开发者的要求

- (1) 熟悉标准 GNU/Linux 应用开发，以及常见 Linux 系统命令
- (2) 掌握一些驱动、网络协议基本知识。
- (3) 了解一些 AT 命令知识，可参照 `quectel at` 命令手册。

1.2 QuecOpen 开发过程

- (1) 准备好 Ubuntu1404 或者 1604 系统，内存 4GB 以上，cpu 4 核心以上。如果用虚拟机，分配给虚拟机内存不低于 4GB
- (2) 按照第 2 章安装开发工具、驱动和 SDK
- (3) 按照 3.1 写一个简单的 APP，熟悉 QuecOpen SDK 的开发流程
- (4) 按照 3.4 节重新导入客户 APP 到根文件系统里，并重新生成文件系统 img
- (5) 高级开发可参照其他相关文档。

2. 开发环境准备

2.1 工具安装

Ubuntu 的 USB 驱动安装和烧录工具安装请参照《[Quectel_WCDMA<E_Linux_USB_Driver_User_Guide](#)》

2.2 ADB 安装

(1) 安装 ADB 驱动

```
sudo apt-get update
```

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

如果上面失败，请执行：

```
sudo add-apt-repository ppa:nilarimogard/webupd8
```

```
sudo apt-get update
```

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

安装成功之后：

```
ol@ql-Ubuntu:~$ adb
Android Debug Bridge version 1.0.32
```

(2) 添加模块 USB VID

查询设备 VID

```
lsusb
```

修改配置文件

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

(3) 枚举设备

```
sudo adb kill-server
```

```
sudo adb devices
```

2.3 固件更新

拿到最新固件和 SDK 后，需要对设备进行更新，一是对 2.1 工具安装的验证，而是对设备的升级做验证。固件更新请参照《[KBA_QuecOpen_软件升级下载方式介绍](#)》

2.5 文件下载

这里所描述的单独一个普通文件或者 APP 导入到模块 Linux 文件文件系统中

2.5.1 基于 ADB

命令基本格式:

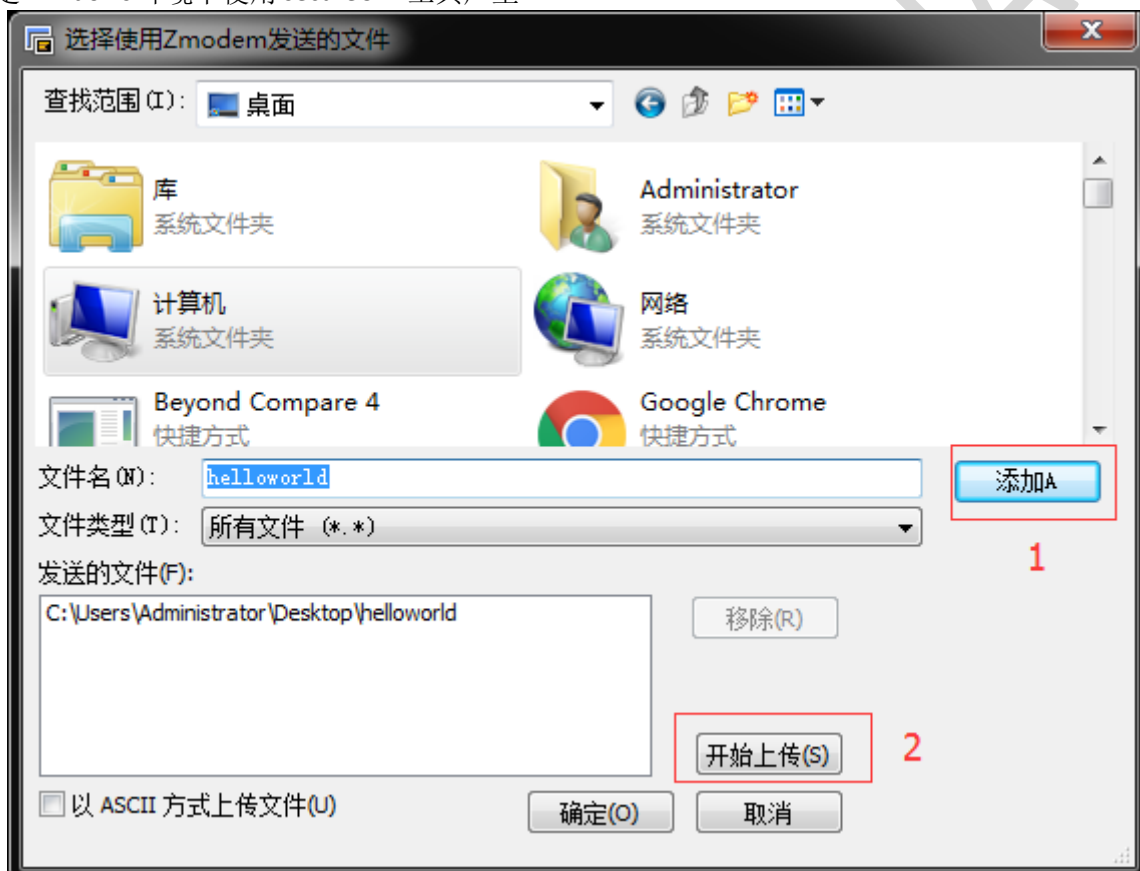
```
sudo adb push <local path> <module path>
```

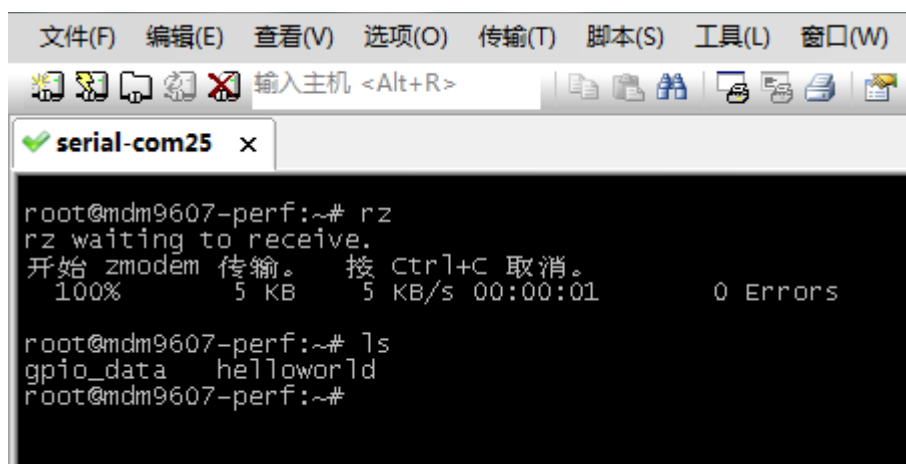
例如

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

2.5.2 基于串口

这里是 Windows 环境下使用 secureCRT 工具产生





```

文件(F) 编辑(E) 查看(V) 选项(O) 传输(T) 脚本(S) 工具(L) 窗口(W)
输入主机 <Alt+R>
serial-com25 x
root@mdm9607-perf:~# rz
rz waiting to receive.
开始 zmodem 传输。 按 Ctrl+C 取消。
100%      5 KB      5 KB/s 00:00:01      0 Errors
root@mdm9607-perf:~# ls
gpio_data  helloworld
root@mdm9607-perf:~#

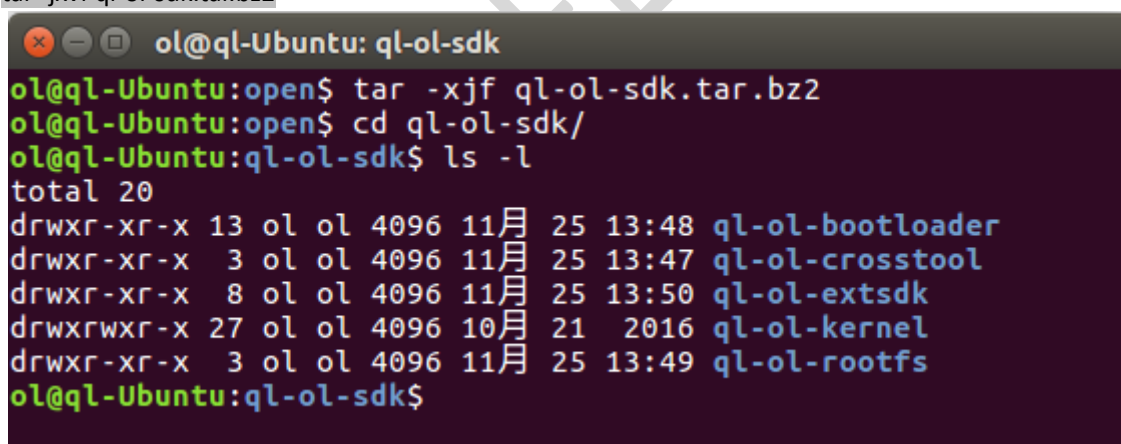
```

2.6 SDK 安装

(1) 解压

SDK 压缩包的解压必须在 **ubuntu 普通用户** 环境下解压。

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



```

ol@ql-Ubuntu: ql-ol-sdk
ol@ql-Ubuntu:open$ tar -xjf ql-ol-sdk.tar.bz2
ol@ql-Ubuntu:open$ cd ql-ol-sdk/
ol@ql-Ubuntu:ql-ol-sdk$ ls -l
total 20
drwxr-xr-x 13 ol ol 4096 11月 25 13:48 ql-ol-bootloader
drwxr-xr-x  3 ol ol 4096 11月 25 13:47 ql-ol-crosstool
drwxr-xr-x  8 ol ol 4096 11月 25 13:50 ql-ol-extsdk
drwxrwxr-x 27 ol ol 4096 10月 21 2016 ql-ol-kernel
drwxr-xr-x  3 ol ol 4096 11月 25 13:49 ql-ol-rootfs
ol@ql-Ubuntu:ql-ol-sdk$

```

(2) 文档内容介绍

| 目录 | 说明 |
|------------------|---------------------------------|
| ql-ol-crosstool | 交叉工具链 |
| ql-ol-bootloader | 高通 bootloader 源码（根据客户定制权限才开放） |
| ql-ol-kernel | Linux 内核源码（根据客户定制权限才开放） |
| ql-ol-rootfs | 平台运行时的根文件系统 |
| ql-ol-extsdk | 包含了一些 API, example 以及 tools 工具包 |

(3) 执行

```
cd ql-ol-sdk
```

```
source ql-ol-croostool/ql-ol-croostool-env-init
```

```
ol@ql-Ubuntu: ql-ol-sdk
```

```
ol@ql-Ubuntu:ql-ol-sdk$ source ql-ol-croostool/ql-ol-croostool-env-init
```

```
QUECTEL_PROJECT_NAME      =EC20CE_FA
QUECTEL_PROJECT_REV       =EC20CEFAR05U1
QUECTEL_FEATURE_OPENLINUX =OL
```

```
ol@ql-Ubuntu: ql-ol-sdk
```

```
ol@ql-Ubuntu:ql-ol-sdk$ arm-oe-linux-gnueabi-gcc -v
```

```
Using built-in specs.
```

```
COLLECT_GCC=arm-oe-linux-gnueabi-gcc
```

```
COLLECT_LTO_WRAPPER=/home/ol/ol-sdk/open/ql-ol-sdk/ql-ol-croostool/sysroots/x86_64-oesdk-linux/usr/bin/arm-oe-linux-gnueabi/../../../../libexec/arm-oe-linux-gnueabi/gcc/arm-oe-linux-gnueabi/4.9.2/lto-wrapper
```

```
Target: arm-oe-linux-gnueabi
```

```
Configured with: /home/ol/ws/ol-ql/MDM9x07/OpenLinux/MCU_R05_update01/apps_proc/oe-core/build/tmp-glibc/work-shared/gcc-4.9.2-r0/gcc-4.9.2/configure --build=x86_64-linux --host=x86_64-oesdk-linux --target=arm-oe-linux-gnueabi --prefix=/usr/local/oe-core-x86_64/sysroots/x86_64-oesdk-linux/usr --exec_prefix=/usr/local/oe-core-x86_64/sysroots/x86_64-oesdk-linux/usr --bindir=/usr/local/oe-core-x86_64/sysroots/x86_64-oesdk-linux/usr/bin/arm-oe-linux-gnueabi --sbindir=/usr/local/oe-core-x86_64/sysroots/x86_64-oesdk-linux/usr/bin/arm-oe-linux-gnueabi --libexecdir=/usr/local/oe-core-x86_64/sysroots/x86_64-oesdk-linux/usr/share --sysconfdir=/usr/local/oe-core-x86_64/sysroots/x86_64-oesdk-linux/etc --sharedstatedir=/usr/local/oe-core-x86_64/sysroots/x86_64-oesdk-linux/com --localstatedir=/usr/local/oe-core-x86_64/sysroots/x86_64-oesdk-linux/var --libdir=/usr/local/oe-core-x86_64/sysroots/x86_64-oesdk-linux/usr/lib/arm-oe-linux-gnueabi --includedir=/usr/local/oe-core-x86_64/sysroots/x86_64-oesdk-linux/usr/include --oldincludedir=/usr/local/oe-core-x86_64/sysroots/x86_64-oesdk-linux/usr/include --infodir=/usr/local/oe-core-x86_64/sysroots/x86_64-oesdk-linux/usr/share/info --mandir=/usr/local/oe-core-x86_64/sysroots/x86_64-oesdk-linux/usr/share/man --disable-silent-rules --disable-dependency-tracking --with-libtool-sysroot=/home/ol/ws/ol-ql/MDM9x07/OpenLinux/MCU_R05_update01/apps_proc/oe-core/build/tmp-glibc/sysroots/x86_64-nativesdk-oesdk-linux --with-gnu-ld --enable-shared --enable-languages=c,c++ --enable-threads=posix --enable-multi-lib --enable-c99 --enable-long-long --enable-symvers=gnu --enable-libstdc++-pch --program-prefix=arm-oe-linux-gnueabi- --without-local-prefix --enable-target-optspace --enable-lto --enable-libssp --disable-bootstrap --disable-libmudflap --with-system-zlib --with-linker-hash-style=gnu --enable-linker-build-id --with-ppl=no --with-cloog=no --enable-checking=release --enable-headers=c_global --with-gxx-include-dir=/usr/local/oe-core-x86_64/sysroots/x86_64-oesdk-linux/usr/armv7a-vfp-neon-oe-linux-gnueabi/usr/include/c++/4.9.2 --with-build-time-tools=/home/ol/ws/ol-ql/MDM9x07/OpenLinux/MCU_R05_update01/apps_proc/oe-core/build/tmp-glibc/sysroots/x86_64-linux/usr/arm-oe-linux-gnueabi/bin --with-sysroot=/usr/local/oe-core-x86_64/sysroots/x86_64-oesdk-linux/usr/armv7a-vfp-neon-oe-linux-gnueabi --with-build-sysroot=/home/ol/ws/ol-ql/MDM9x07/OpenLinux/MCU_R05_update01/apps_proc/oe-core/build/tmp-glibc/sysroots/mdm9607-perf --enable-poison-system-directories --with-mpfr=/home/ol/ws/ol-ql/MDM9x07/OpenLinux/MCU_R05_update01/apps_proc/oe-core/build/tmp-glibc/sysroots/x86_64-nativesdk-oesdk-linux --with-mpc=/home/ol/ws/ol-ql/MDM9x07/OpenLinux/MCU_R05_update01/apps_proc/oe-core/build/tmp-glibc/sysroots/x86_64-nativesdk-oesdk-linux --enable-nls
Thread model: posix
gcc version 4.9.2 (GCC)
```

```
ol@ql-Ubuntu:ql-ol-sdk$
```

(4) 检验

编译所有 example

```
cd ql-ol-extsdk/example
```

```
make
```

```
ol@ql-Ubuntu:ql-ol-sdk$ cd ql-ol-extsdk/example/
```

```
ol@ql-Ubuntu:example$ ls
```

```
adc  atc_pipe  data  gnss      i2c      qmi timer  sleep_wakelock  timer  tzone
API  audio     eint  gpio      Makefile  README  spi      tts      uart
at   call     file  hello_world  pthread  sgmiit  time     tty2tcp  wifi
```

```
ol@ql-Ubuntu:example$ make
```

```
make[1]: Entering directory '/home/ol/ol-sdk/open/ql-ol-sdk/ql-ol-extsdk/example/sleep_wakelock'
arm-oe-linux-gnueabi-gcc -march=armv7-a -mfloat-abi=softfp -mfpu=neon -O2 -fexpensive-optimiza
```

编译单独一个 example

```
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 -mfp=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 -mfp=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$
```

3. Linux 开发与调试

所有开发之前，都必须先执行 `source ql-ol-croostool/ql-ol-croostool-env-init`。
这里只是开始…

3.1 Linux APP 开发

QuecOpen Linux 标准 APP 开发与传统嵌入式 ARM-Linux 开发流程上并无任何特殊之处，对客户的要求只要有过基本的 Linux 应用开发经验即可。下面本文档从 helloworld 到单路拨号上网，引导客户去体验 QuecOpen Linux 开发过程。

3.1.1 Helloworld

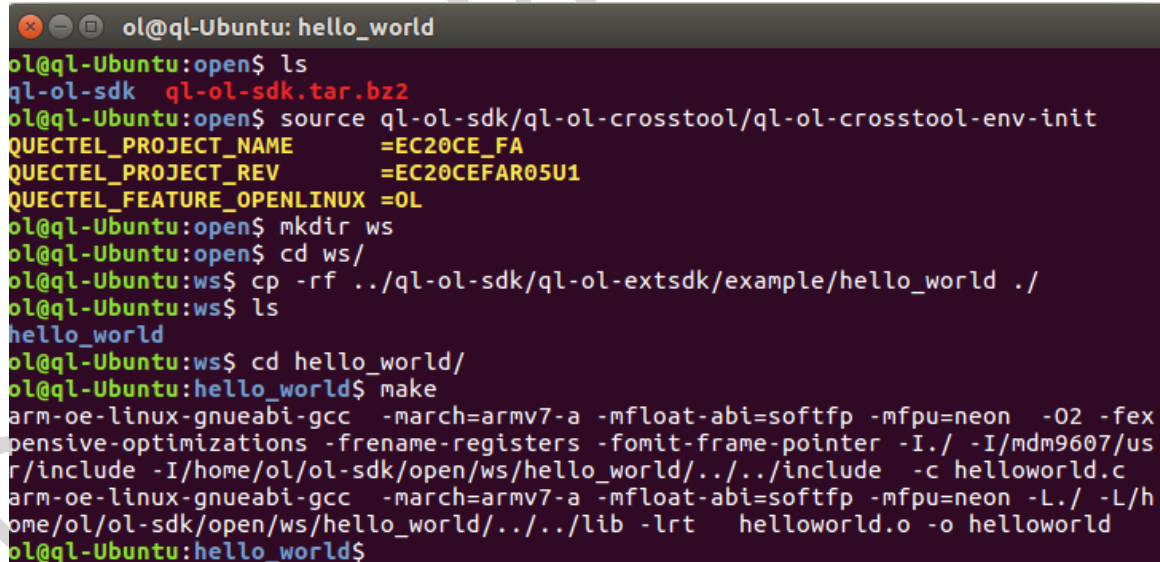
(1) 创建工作目录

这里比如创建 ws

(2) 复制 demo

复制 ql-ol-sdk/ql-ol-extsdk/example/hello_world 到 ws 目录下面

(3) 编译



```
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-croostool/ql-ol-croostool-env-init
QUECTEL_PROJECT_NAME      =EC20CE_FA
QUECTEL_PROJECT_REV       =EC20CEFAR05U1
QUECTEL_FEATURE_OPENLINUX =0L
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 -mcpu=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 -mcpu=neon -L./ -L/h
ome/ol/ol-sdk/open/ws/hello_world/../../lib -lrt helloworld.o -o helloworld
ol@ql-Ubuntu:hello_world$
```

(3) 下载与运行 APP

1) 下载参照 2.5 节下载

2) 修改文件权限为可执行

3) 运行 helloworld

3.1.2 单路拨号

- (1) 复制 demo
复制 ql-ol-sdk/ql-ol-extsdk/example/data
- (2) 编译
- (3) 下载与运行
拨号成功之后:

```

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:~#

```

3.1.3 高级应用开发

请参照我们应用开发文件下的指导文档和 SDK 里面的 example

3.1.4 添加自启动 APP

拷贝 ql-ol-extsdk/tools/quectel_ubi/QuecOpen_startapp 到 ql-ol-rootfs/etc/init.d/ 下，并修改文件内变量 AppProgram 对应的 app 路径

```
# 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"
    ;;
esac
```

cd ql-ol-rootfs/etc/rc5.d

ln -vsf ../init.d/QuecOpen_startapp S45QuecOpen_startapp

3.2 Bootloader 开发

make about //编译 bootloader, 并在当前路径 target/下生成 appsboot.mbn;

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

make about/clean //清除上一条指令生成的文件

```
gale@eve-linux02:~/MDM9x07/SDK_FAG1127/ql-ol-sdk$ make about/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$
```

3.3 Kernel 开发

make kernel //编译生成 boot.img 在当前路径 target/下

```
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.dts
make[1]: Entering directory `/home/gale/MDM9x07/SDK_FAG1127/ql-ol-sdk/ql-ol-kernel'
```

make kernel/clean //清除上一条指令生成的文件;

```
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 1
make[1]: Entering directory `/home/gale/MDM9x07/SDK_FAG1127/ql-ol-sdk/ql-ol-kernel'
```

make kernel_module //编译内核模块（修改了 kmod 相关代码才需要执行），并自动安装到 rootfs 目录下，需要重新制作 sysfs.ubi;

```
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 CROSS_COMPILE=arm-oe-linux-gnueabi- O=/home/gale/MDM9x07/SDK_FAG1127/ql-ol-sdk/ql-ol-kernel
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
```

3.4 文件系统制作

make rootfs //编译生成 sysfs.ubi，放在当前路径 target/下

```
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 //清除上一条指令生成的文件；

```
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$
```

3.5 一键化编译

QuecOpen SDK 提供了一键化整体 about, kernel, kernel_module, rootfs 在内的所以工程。

make //整体编译，放在当前路径 target/下

```
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 //删除上一条产生的文件

```
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'
```

将编译生成的文件放到版本包中进行烧录。

QUECTEL PRELIMINARY

4.模块开机检查

本章以 AT 命令来阐述一下模块开机后，用户如何判断模块是否处在基本的正常工作状态，具体步骤如下：

- (1) PC 通过 USB 转串口线连上模块的 MAIN UART 口或者 USB AT 口
- (2) 请插上 SIM 卡和天线，并上电
- (3) 通过 QCOM 工具，发送如下 AT 命令

检测串口：AT

检测 SIM 卡：AT+CPIN?

检测信号强度：AT+CSQ

检测模块注册网：AT+CGREG?

查询运营商：AT+COPS?

查询网络制式：AT+QNWINFO

语音测试：ATDxxx;

下图是插入一张普通中国移动卡的开机检查

```
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
```

FAQ

1. 为什么压缩包必须在普通用户环境下解压？

可以查询 tar 命令手册看到：

```
--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)
```