

EC2x&AG35-QuecOpen

快速入门

LTE 系列

版本: EC2x&AG35-QuecOpen_快速入门_V1.4

日期: 2018-03-16

状态: 临时文件

上海移远通信技术股份有限公司始终以为客户提供最及时、最全面的服务为宗旨。如需任何帮助，请随时联系我司上海总部，联系方式如下：

上海移远通信技术股份有限公司

上海市徐汇区虹梅路 1801 号宏业大厦 7 楼 邮编：200233

电话：+86 21 51086236 邮箱：info@quectel.com

或联系我司当地办事处，详情请登录：

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

如需技术支持或反馈我司技术文档中的问题，可随时登陆如下网址：

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

或发送邮件至：support@quectel.com

前言

上海移远通信技术股份有限公司提供该文档内容用以支持其客户的产品设计。客户须按照文档中提供的规范、参数来设计其产品。由于客户操作不当而造成的人身伤害或财产损失，本公司不承担任何责任。在未声明前，上海移远通信技术股份有限公司有权对该文档进行更新。

版权申明

本文档版权属于上海移远通信技术股份有限公司，任何人未经我司允许而复制转载该文档将承担法律责任。

版权所有 ©上海移远通信技术股份有限公司 2018，保留一切权利。

Copyright © Quectel Wireless Solutions Co., Ltd. 2018.

文档历史

修订记录

版本	日期	作者	变更表述
1.0	2017-11-25	钱润生/ 高飞虎	初始版本
1.1	2018-02-01	钱润生	增加 FAQ 章节
1.2	2018-02-10	钱润生	增加 APP 开机自启动
1.3	2018-02-23	高飞虎	1. 增加内核配置文件的选择和修改方法 2. 增加 usrdata.ubi 的制作方法
1.4	2018-03-16	钱润生	1. 更新文档名称 2. 增加第 1 章引言和第 2 章文档阅读指导 3. 增加 Debug 固件版本编译方法（5.7 章节）

目录

文档历史	1
目录	2
表格索引	4
图片索引	5
1 引言	6
2 文档阅读指导.....	7
2.1. QuecOpen 产品简介	7
2.2. 硬件开发	7
2.3. 软件开发	8
3 QuecOpen 软件开发流程介绍.....	9
3.1. 对开发者的要求	9
3.2. QuecOpen 开发过程	9
4 开发环境准备.....	10
4.1. 工具安装	10
4.2. ADB 安装.....	10
4.2.1. 安装 ADB 驱动.....	10
4.2.2. 添加模块 USB VID.....	10
4.2.3. 枚举设备	11
4.3. 固件更新	11
4.4. 文件下载	11
4.4.1. 基于 ADB	11
4.4.2. 基于串口	11
4.5. SDK 安装.....	13
4.5.1. 解压	13
4.5.2. 文档内容介绍.....	13
4.5.3. 执行	14
4.5.4. 检验	15
5 Linux 开发与调试.....	17
5.1. Linux APP 开发	17
5.1.1. Helloworld.....	17
5.1.2. 单路拨号	18
5.1.3. 高级应用开发.....	18
5.1.4. 添加自启动 APP	18
5.2. Bootloader 开发.....	19
5.3. Kernel 开发.....	19
5.4. 文件系统制作.....	21
5.5. usrdata.ubi 制作	21
5.6. 一键化编译	22
5.7. Debug 固件版本编译	22

6	模块开机检查.....	23
7	FAQ.....	24

Quectel
Confidential

表格索引

表 1: QUECOPEN 项目硬件设计参考文档	7
表 2: SDK 压缩包中文档内容介绍	13

图片索引

图 1：使用 SECURECRT 工具进行文件传输 12

图 2：基于 ZMODEM 上传文件步骤详解..... 13

图 3：文件上传成功 13

1 引言

本文档主要介绍有关移远通信 EC2x&AG35-QuecOpen 模块的软件开发流程、开发环境、Linux 开发与调试的基本方法和步骤、以及模块开机检查的方法。

该文档主要适用于移远通信的如下模块：

- EC2x-QuecOpen
本文档中，EC2x 包含 EC25/EC21/EC20 R2.1/EC20 R2.0；
EC2x-QuecOpen 包含 EC25/EC21/EC20 R2.1/EC20 R2.0-QuecOpen
- AG35-QuecOpen

2 文档阅读指导

2.1. QuecOpen 产品简介

欲了解有关 EC2x&AG35-QuecOpen 产品的技术架构和软硬件资源，请分别参考如下文档：

- *Quectel_EC2x-QuecOpen_技术与资源综述*
- *Quectel_AG35-QuecOpen_技术与资源综述*

2.2. 硬件开发

1) 功能引脚选择

请参阅如下功能需求和引脚资源定义文档：

- *Quectel_EC2x-QuecOpen_GPIO_Assignment_Speadsheet*
- *Quectel_AG35-QuecOpen_GPIO_Assignment_Speadsheet*

2) SCH 和 PCB 设计

表 1: QuecOpen 项目硬件设计参考文档

	参考文档	
硬件设计指导	<i>Quectel_EC2x-QuecOpen_Hardware_Design</i>	<i>Quectel_AG35-QuecOpen_Hardware_Design</i>
模块原理图设计	<i>EC2x-TE-A_SCH</i> <i>AG35-TE-A_SCH</i>	
EVB 板原理图	<i>LTE-OPEN-EVB_SCH</i>	

3) 参考如上步骤进行设计的过程中，若有任何疑问请联系移远通信技术支持人员。

2.3. 软件开发

请先直接阅读完本篇文档后，再阅读 *软件开发* 文件夹下面对应的功能文档

3 QuecOpen 软件开发流程介绍

3.1. 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，可参阅如下 AT 命令手册：
 - Quectel_AG35_AT_Commands_Manual
 - Quectel_EC2x_AT_Commands_Manual

3.2. QuecOpen 开发过程

- 1) 准备好 Ubuntu 1404 或者 1604 的 64 位系统，内存 4GB 以上，CPU 4 核心以上；如果用虚拟机，则分配给虚拟机的内存应不低于 4GB；
- 2) 参阅 **第4章** 安装开发工具、驱动和 SDK；
- 3) 参阅 **5.1 章节** 写一个简单的 APP，熟悉 QuecOpen SDK 的开发流程；
- 4) 参阅 **5.4 章节** 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) 高级开发可参阅其他相关文档。

4 开发环境准备

4.1. 工具安装

Ubuntu 的 USB 驱动安装和烧录工具安装请参阅如下文档:
[Quectel_WCDMA<E_Linux_USB_Driver_User_Guide](#)

4.2. ADB 安装

4.2.1. 安装 ADB 驱动

运行如下命令安装 ADB 驱动:

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

如果上面方法失败, 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, 显示如下:

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

4.2.2. 添加模块 USB VID

查询设备 VID:

```
lsusb
```

修改配置文件:

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

4.2.3. 枚举设备

运行如下命令枚举设备：

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

4.3. 固件更新

拿到最新固件和 SDK 后，需要对设备进行更新：一方面对所安装的工具进行验证；另一方面对设备的升级做验证。

固件更新方法，请参阅 *Quectel_EC2x&AG35-QuecOpen_软件升级下载方式介绍*。

4.4. 文件下载

本章节主要介绍如何将单独一个普通文件或者 APP 导入到模块 Linux 文件系统里。

4.4.1. 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
```

4.4.2. 基于串口

如下主要介绍在 Windows 环境下，如何使用 SecureCRT 工具下载文件。

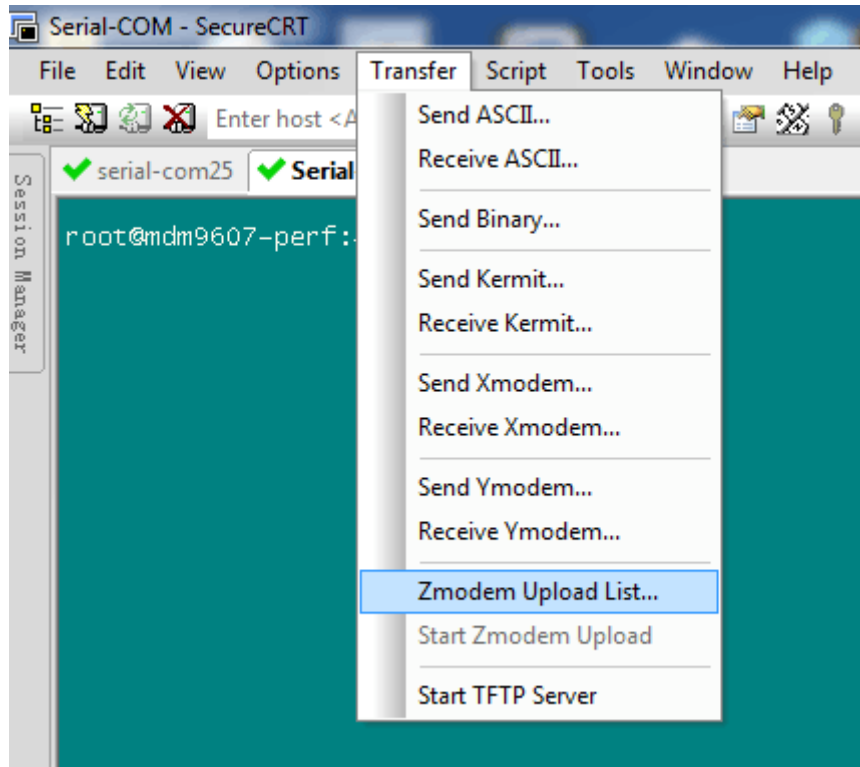


图 1：使用 SecureCRT 工具进行文件传输

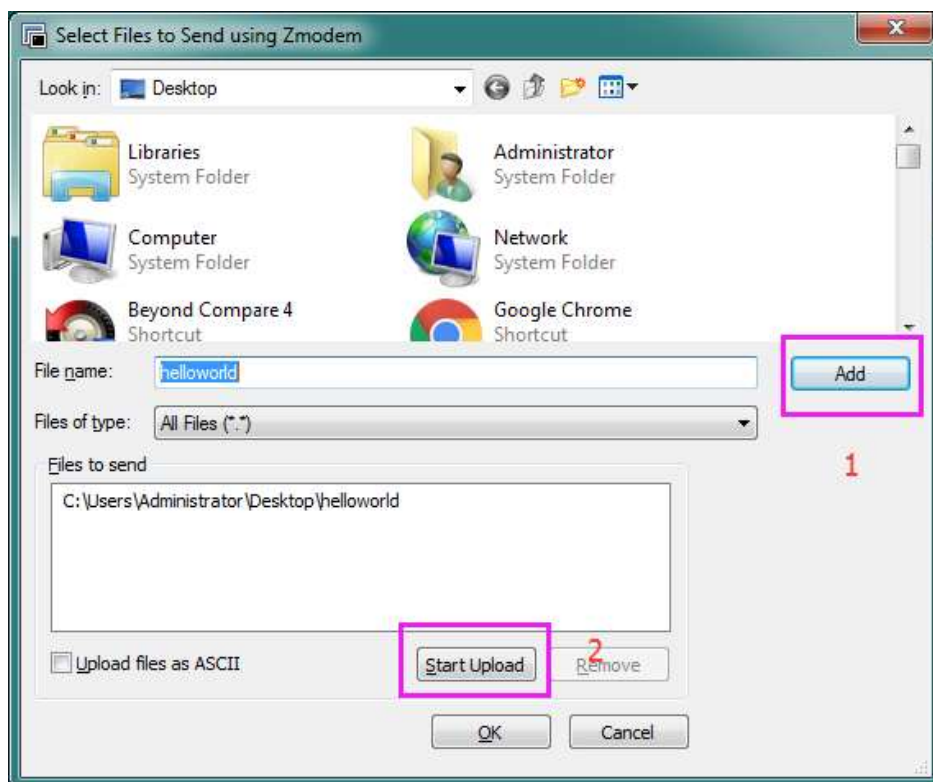


图 2：基于 Zmodem 上传文件步骤详解

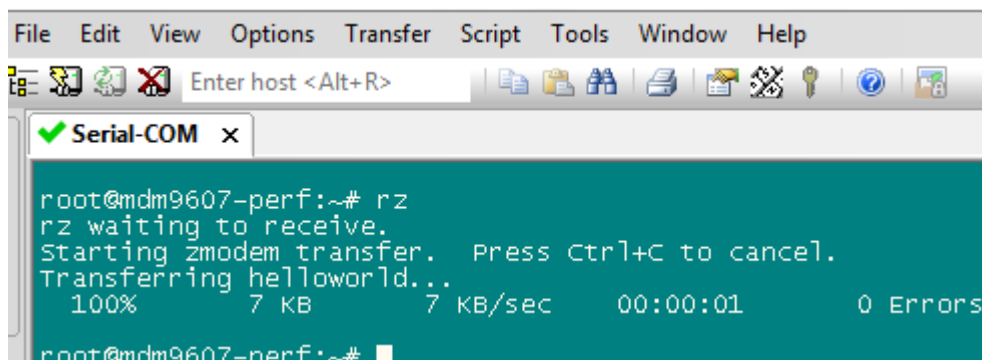


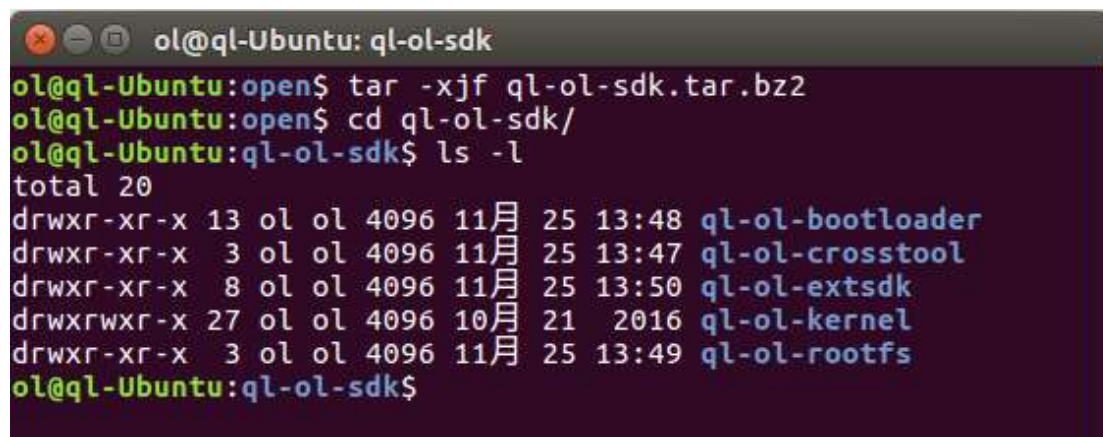
图 3：文件上传成功

4.5. SDK 安装

4.5.1. 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
```



4.5.2. 文档内容介绍

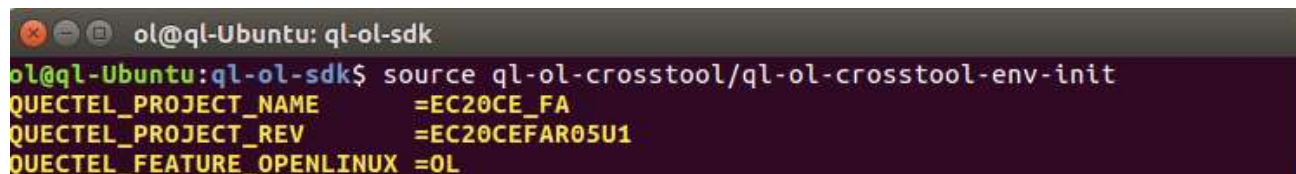
表 2：SDK 压缩包中文档内容介绍

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

4.5.3. 执行

执行如下命令以安装 SDK:

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



```
ol@ql-Ubuntu: ql-ol-sdk
ol@ql-Ubuntu:ql-ol-sdk$ source ql-ol-crosstool/ql-ol-crosstool-env-init
QUECTEL_PROJECT_NAME      =EC20CE_FA
QUECTEL_PROJECT_REV       =EC20CEFAR05U1
QUECTEL_FEATURE_OPENLINUX =0L
```



```

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-crosstool/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 --libexecdir=/usr/local/oe-core-x86_64/sysroots/x86_64-oesdk-linux/usr/libexec/arm-oe-linux-gnueabi --datadir=/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$

```

4.5.4. 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  sgmi       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$
```

5 Linux 开发与调试

所有开发之前，都必须先执行：

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

5.1. Linux APP 开发

QuecOpen Linux 标准 APP 开发与传统嵌入式 ARM-Linux 的开发，在流程上并无任何特殊之处；开发者只要有过基本的 Linux 应用开发经验即可。

如下将介绍从 Helloworld 创建到单路拨号上网，以引导开发者去体验 QuecOpen Linux 开发过程。

5.1.1. 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

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-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$
```

4) 下载与运行 APP

- 参照 4.4 章节下载
- 修改文件权限为可执行
- 运行 helloworld

5.1.2. 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:0 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:~#
```

5.1.3. 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。

5.1.4. 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"
```

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

5.2. 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 T00
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$
```

5.3. Kernel 开发

首先需要进行内核选项文件的指定和修改, 使用以下命令指定 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 0=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
choice value used outside its choice group
choice 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 >
```

可以按需要修改内核选项，若不修改则直接退出；此时会生成隐藏文件在 `ql-ol-kernel/build/.config`，此文件是 **make kernel** 过程中所依赖的内核配置文件，当用户确定此修改后的 `.config` 文件要保存提交代码仓库，则需要运行如下命令以保存之前的修改到 `ql-ol-kernel/arch/arm/configs/mdm9607-perf_defconfig`（若用户使用 `git` 维护代码，那么需要追踪此文件，而不要追踪 `.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 //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.d
make[1]: Entering directory `/home/gale/MDM9x07/SDK_FAG1127/ql-ol-sdk/ql-ol-kern
```

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
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 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
```

5.4. 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_ubi/ubinize
./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 image
```

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$
```

5.5. usrdata.ubi 制作

Flash 默认有一个 usr_data 分区，可以用来存放用户的文件和进行 DFOTA 升级使用。

make usrdata //生成 usrdata.ubi

```
ql-ol-sdk$ make usrdata
ql-ol-sdk ; chmod +x ./ql-ol-extsdk/tools/quectel_ubi/* ; ./ql-ol-extsdk/tools/quectel_ubi/ubinize -o usrdata.ubi -m 4096 -p 256KiB -s 4096 ql-ol-extsdk/tools/quectel_ubi/ubinize
target/
```

make usrdata/clean //清除上一条指令生成的文件

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

5.6. 一键化编译

QuecOpen SDK 支持一键化整体编译包括 aboot, kernel, kernel_module, rootfs, usrdata.ubi 在内的所有工程。

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。

5.7. Debug 固件版本编译

遇到问题需要打开内核调试 Log 的时候，需要编译 Debug 版本，编译方法如下：

1) 配置

make debug_kernel_menuconfig

2) 编译文件系统及内核

make debug_version

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

将生成的文件系统和内核镜像重新烧录即可。

6 模块开机检查

本章以 AT 命令来阐述模块开机后，用户如何判断模块是否处于正常工作状态，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(U)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(U)SIM 卡：AT+CPIN?
 - 检测信号强度：AT+CSQ
 - 检测模块注网：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流程：

```
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+qnwinf
+QNWINFO: "TDD LTE","46000","LTE BAND 40",38950

OK
atd1[REDACTED];
OK
```

7 FAQ

1. 为什么压缩包必须在普通用户环境下解压？

通过查询 **tar** 命令手册可以看到：

```
--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)
```