www.neoway.com

Neoway 有方

Android RIL 驱动使用说明

Version 1.1

有 物 联　方 智 能

GET CONNECTED GET SMART

版权声明

商标声明

Neoway 有方是深圳市有方科技股份有限公司所有商标。

本文档中出现的其他商标，由商标所有者所有。

说明

本文档适用产品为 ***N710、N720 模块。***

本文档的使用对象为系统工程师，开发工程师及测试工程师。

由于产品版本升级或其它原因，本文档内容会在不预先通知的情况下进行必要的更新。

除非另有约定，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

深圳市有方科技股份有限公司为客户提供全方位的技术支持，任何垂询请直接联系您的客户经理或发送邮件至以下邮箱：

Sales@neoway.com

Support@neoway.com

公司网址：http://www.neoway.com

修订记录		
版本号	生效年月	更改内容
V1.0	2017-04	初始版本
V1.1	2017-08	增加 GPS、增加短信功能、增加 Android6.0 相关的、完善库

目　　录

关于本文档

本文档用于指导客户如何把有方 RIL 库驱动集成到基于 Android OS 的设备中，如何修改配置文件进行 RIL 库驱动的调试完成 PPP 拨号或者 NIDS 拨号。

1 RIL 驱动介绍

1.1 产品

型号	是否支持
N720 系列	YES
N710 系列	YES
WM620	YES

1.2 VID 和 PID 表

有方 4G LTE 模块有多种型号，可以支持 NDIS 和 PPP 拨号，对应的 VID 和 PID 表如下：

型号	VID	PID
N720	0x2949	0x8241
N710	0x05c6	0x8080
WM620	0x05c6	0x9002

N710 的 USB 端口的配置如下：

Default USB composition: RNDIS + MODEM + TTY + Diag

N720 的 USB 端口的配置如下：

Default USB composition: RNDIS + MODEM + TTY + TTY + Mass storage + ADB + Diag

1.3 功能

功能	是否支持
数据业务	YES
短信	YES
GPS	YES

1.4 版本

版本	是否支持
Android 3.x	YES
Android 4.x	YES
Android 5.x	YES
Android 6.x	YES

2 USB 串口驱动配置

2.1 添加 VID/PID

以添加 N710 为例（不同的模块对应不同的 VID/PID），打开内核源码文件

ption.c(drivers/usb/serial/option.c)

#define QUALCOMM_VENDOR_ID 0x05C6

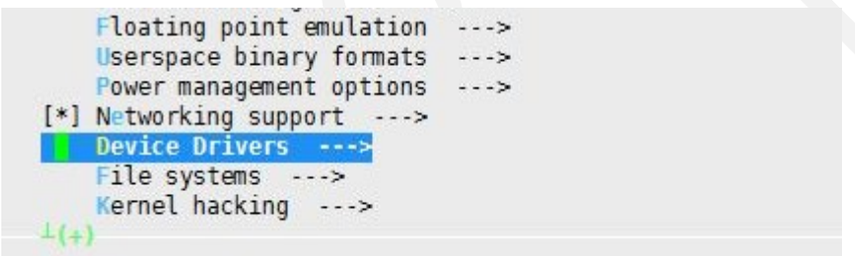
在 option_ids 数组添加：

USB_DEVICE(QUALCOMM_VENDOR_ID,0x8080)}

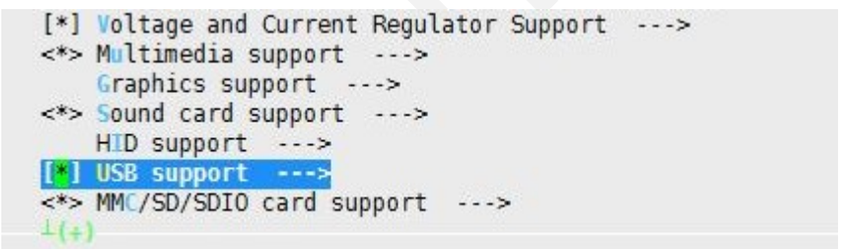
2.2 Menuconfig 图形界面中配置串口驱动

在 kernel 目录里执行 make menuconfig，弹出内核配置图形界面进行如下选择：

步骤 1：进入到“Device Drivers”选项中；



步骤 2：选中“USB support”选项；



步骤 3：选中 “USB Gadget Support”选项；

步骤 4：选中 “USB Gadget Drivers (Serial Gadget (with CDC ACM and CDC OBEX support))”选项；

```
--- USB Gadget Support
[ ]   Debugging information files (DEVELOPMENT)
[ ]   Debugging information files in debugfs (DEVELOPMENT)
(2)   Maximum VBUS Power usage (2-500 mA)
(2)   Number of storage pipeline buffers
<*>   USB Peripheral Controller (Inventra HDRC USB Peripheral (TI, ADI, ...))  --->
<*>   USB Gadget Drivers (Serial Gadget (with CDC ACM and CDC OBEX support))  --->
```

步骤 5：返回上一级目录，选中“USB Serial Converter support”选项。

```
      *** USB port drivers ***
<*>   USB Serial Converter support  --->
      *** USB Miscellaneous drivers ***
< >   EMI 6|2m USB Audio interface support
< >   EMI 2|6 USB Audio interface support
< >   ADU devices from Ontrak Control Systems
< >   USB 7-Segment LED Display
< >   USB Diamond Rio500 support
< >   USB Lego Infrared Tower support
```

2.3 测试

按上述步骤编译重刷后，对串口配置成功之后，查看设置节点是否挂出/dev/ttyUSB*生成。如果已经生成，上层应用就可以通过这些设备和模块交互了（发送 AT 命令等）。

```
root@sugar-adtv:/dev # ls ttyUSB*
ls ttyUSB*
ttyUSB0
ttyUSB1
ttyUSB2
ttyUSB3
ttyUSB4
```

N710 如果出现 3 个 ttyUSB 口：ttyUSB0 是 AT 口，ttyUSB1 是 modem 口；如果出现 5 个 ttyUSB 口：ttyUSB2 是 AT 口，ttyUSB3 是 modem 口。

N720 如果出现 6 个 ttyUSB 口：ttyUSB3 是 AT 口，ttyUSB1 是 modem 口；如果出现 5 个 ttyUSB 口：ttyUSB2 是 AT 口，ttyUSB0 是 modem 口。

测试 AT 指令，若提示需要权限，则需要先修改 ttyUSB 的权限。

```
root@sugar-adtv:/dev # cat ttyUSB2
cat ttyUSB2


+CGMR: N710_C147CNBZ_V011

OK



+CGMR: N710_C147CNBZ_V011

OK
```

```
root@sugar-adtv:/dev # echo -e "AT+CGMR\r\n">ttyUSB2
echo -e "AT+CGMR\r\n">ttyUSB2
root@sugar-adtv:/dev # echo -e "AT+CGMR\r\n">ttyUSB2
echo -e "AT+CGMR\r\n">ttyUSB2
root@sugar-adtv:/dev # ^C
C:\Users\Leatricw>
```

3 USB NDIS 内核配置

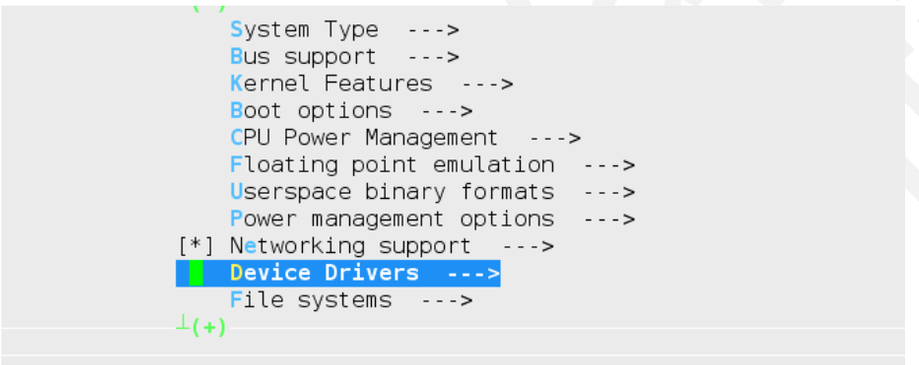
如果是 PPP 拨号方式，直接跳过这一步，进入 5 RIL 集成。

USB NDIS 内核配置也有两种方法，一种在 deconfig 中配置，一种在 menuconfig 图形界面中配置，以下我们以 menuconfig 图形界面配置为例。

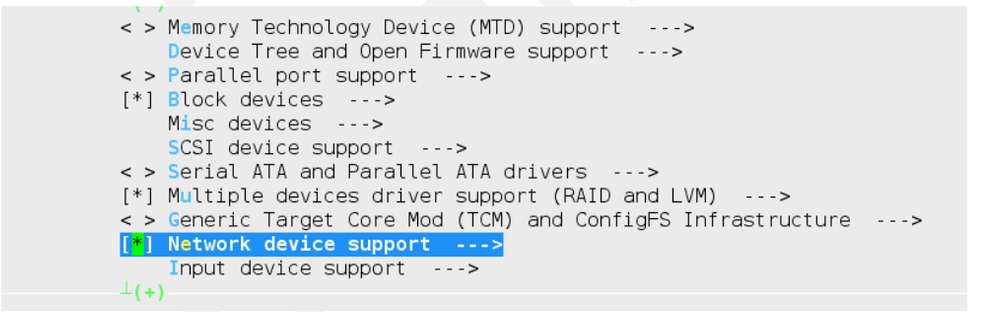
3.1 menuconfig 图形界面配置 NDIS

进入 kernel 目录，执行 make menuconfig 弹出内核配置图形界面进行如下选择：

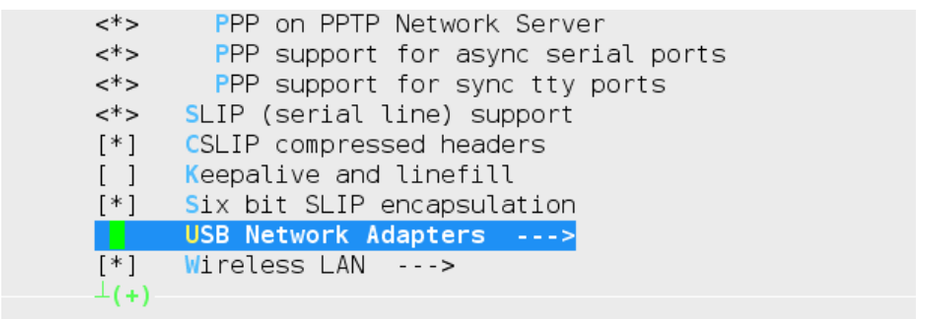
步骤 1：选择 Device Drivers；



步骤 2：选择 Network device support；



步骤 3：选择 USB Network Adapters；



步骤 4：选择 Multi-purpose USB Networking Framework；

```
    <*> USB CATC NetMate-based Ethernet device support
    <*> USB KLSI KL5USB101-based ethernet device support
    <*> USB Pegasus/Pegasus-II based ethernet device support
    <*> USB RTL8150 based ethernet device support
    <*> Realtek RTL8152 Based USB 2.0 Ethernet Adapters
    <*> Multi-purpose USB Networking Framework
    <*>   ASIX AX88xxx Based USB 2.0 Ethernet Adapters
    <*>   ASIX AX88179/178A USB 3.0/2.0 to Gigabit Ethernet
    -*-   CDC Ethernet support (smart devices such as cable modems)
    <*>   CDC EEM support
    -*-   CDC NCM support
    ⊥(+)
```

步骤 5：选择 Host for RNDIS and ActiveSync devices。

```
  <*>   GeneSys GL620USB-A based cables
  <*>   NetChip 1080 based cables (Laplink, ...)
  <*>   Prolific PL-2301/2302/25A1 based cables
  <*>   MosChip MCS7830 based Ethernet adapters
  <*>   Host for RNDIS and ActiveSync devices
  <*>   Simple USB Network Links (CDC Ethernet subset)
  [*]     ALi M5632 based 'USB 2.0 Data Link' cables
  [*]     AnchorChips 2720 based cables (Xircom PGUNET, ...)
  [*]     eTEK based host-to-host cables (Advance, Belkin, ...)
  [*]     Embedded ARM Linux links (iPaq, ...)
  [*]     Epson 2888 based firmware (DEVELOPMENT)
  ⊥(+)
```

3.2 测试

重新编译代码，烧录软件到机器，机器重新上电开机，插上模块。上电开机后，用 netcfg 指令查询，多了 usb0 设备。

```
netcfg
lo       UP                                   127.0.0.1/8   0x00000049 00:00:00:
00:00:00
sit0     DOWN                                   0.0.0.0/0   0x00000080 00:00:00:
00:00:00
usb0     DOWN                                   0.0.0.0/0   0x00001002 da:47:17:
e5:fd:a2
eth0     UP                                     0.0.0.0/0   0x00001003 ae:d0:bc:
3a:09:36
```

4 RIL 集成

解压 Neoway_Android_Ril.tar.gz

libreference-ril.so(ppp 拨号使用)

libreference-ril_ndis.so（ndis 拨号使用，使用时注意库名修改）

gps.default.so（GPS 使用）

Neoway_Android_Ril_Driver_User_Guide_V1.1

4.1 RIL 简单介绍

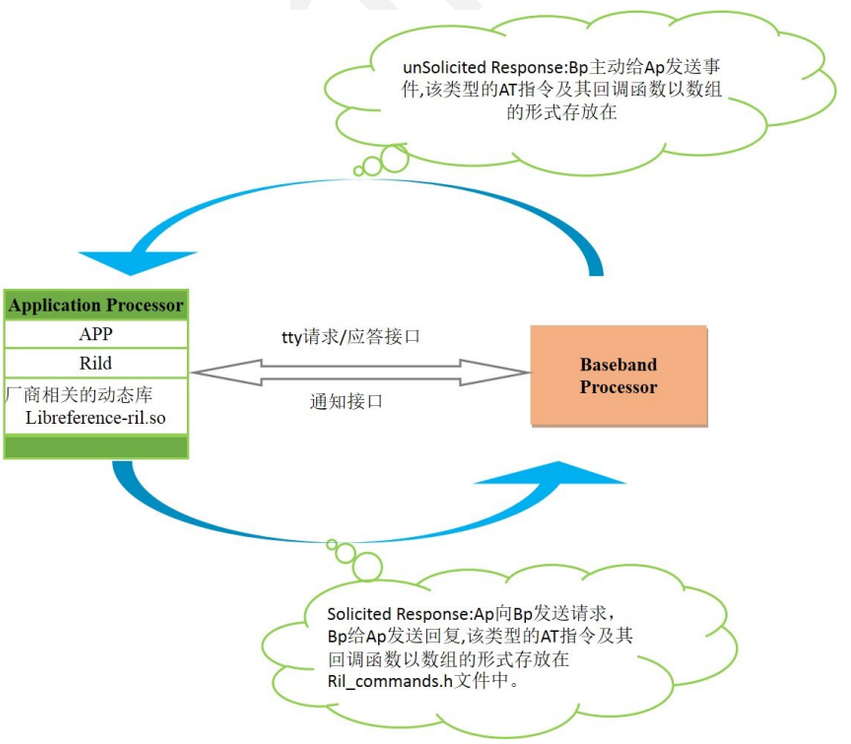
Android RIL（Radio Interface Layer）提供了 Telephony 服务和 Radio 硬件之间的抽象层。RIL 负责数据的可靠传输、AT 命令的发送以及 response 的解析。

4.1.1 AP 与 BP 通信

操作系统、用户界面和应用程序都在 **Appliction Processor(AP)**上执行，AP 一般采用 ARM 芯片的 CPU。而射频通讯控制软件，则运行在另一个分开的 CUP 上，这个 CPU 称为 **Baseband Processor(BP)**。

在 Android 系统中 rild 运行在 AP 上，AP 上的应用通过 rild 发送 AT 指令给 BP，BP 接收到信息后又通过 rild 传送给 AP。AP 与 BP 之间有两种通信方式：

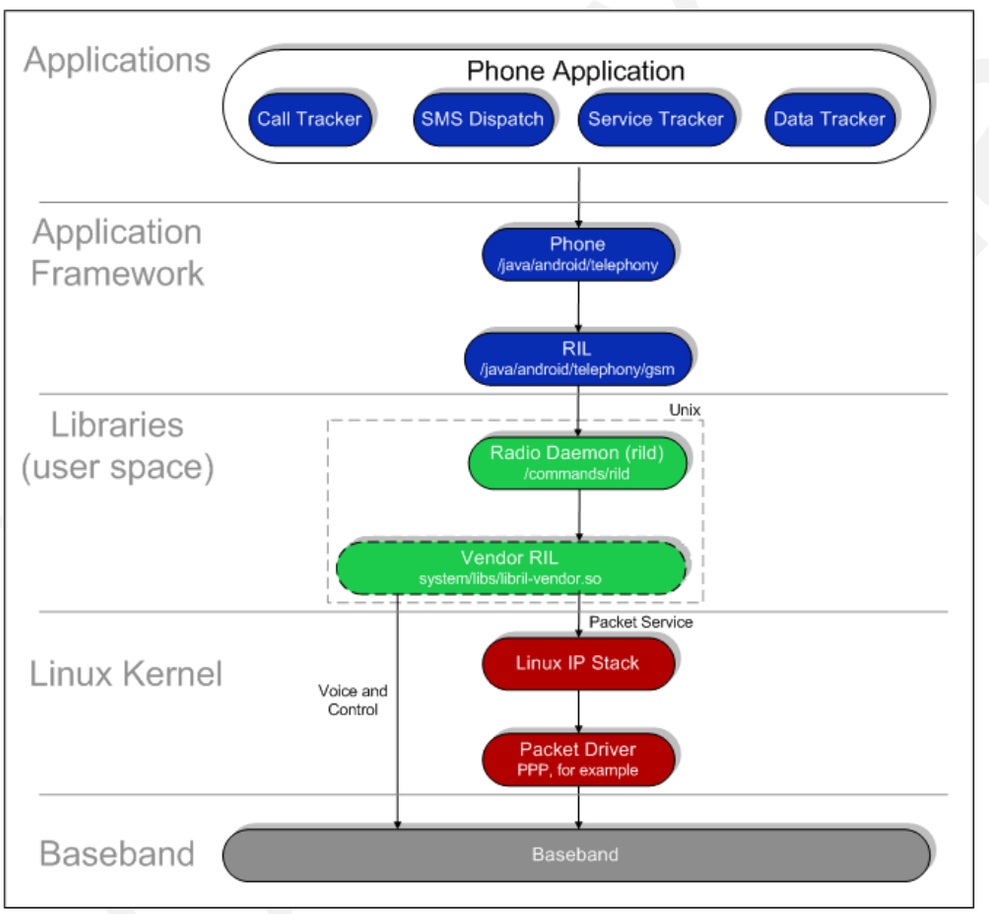
- Solicited Response:Ap 向 Bp 发送请求，Bp 给 Ap 发送回复
- unSolicited Response:Bp 主动给 Ap 发送事件



4.1.2 RIL 在 Android 体系中的位置

Android 的 RIL 分为上层下层两部分：

- 上层部分是基于 java 实现的，它位于 ANDROID 的 Java Framework 中。由两部分组成：
  - RIL 模块，主要用于与下层的 rild 进行通信；
  - Phone 模块，它是直接暴露电话功能接口给应用开发用户，供他们调用以进行电话功能的实现。
- 下层部分是基于 C/C++的实现，位于 Linux Kernel 之上，由两部分组成：
  - Rild，它负责 socket 与应用程序框架进行通信
  - Vendor RIL，负责向下，通过两种方式与 radio 进行通信：它们是直接与 radio 通信的 AT 指令通道和用于传输包数据的通道，数据通道用于手机上网的功能。



4.2 配置 RIL

根据拨号方式的不同，init,rc 所需要修改的有所不同：

步骤 1：修改 init.rc；

一般这个文件放在 android 源码/system/core/rootdir/init.rc 文件下面。可以用 adb shell 进入，修改该 init.rc 文件。

- **NDIS 拨号**

1. 指定库名

service ril-daemon /system/bin/rild –l libreference-ril.so

class main

socket rild stream 660 root radio

socket rild-debug stream 660 radio system

user root

group radio cache inet misc audio sdcard_rw

2. 把 libreference-ril.so 库预置到/system/lib 下，并修改相关文件的权限即可：

adb push libreference-ril.so /system/lib

adb shell chmod 777 /system/bin/ndc

adb shell chmod 777 /system/bin/netcfg

- **PPP 拨号**需要做如下步骤：

1. 在 init.rc 中添加 ril-daemon 服务

service ril-daemon /system/bin/rild -l /system/lib/libreference-ril.so

class main

socket rild stream 660 root radio

socket rild-debug stream 660 radio system

user root

group radio cache inet misc audio sdcard_rw log

如上所示，定义用于启动 RILD 的服务 ril-deamon，/system/bin/rild 就是该服务的执行文件，即 hardware/ril/rild.c 文件编译出来的可执行文件。参数如下：

-l：指定 AT 相关的动态链接库为 libreference-ril.so

init.rc 文件位置取决于各个平台的情况,例如:

system/core/rootdir/init.rc

device/fsl/imx6/init.rc

device/ti/am335xevm_sk/init.am335xevm.rc

device/generic/x86/init.rc

device/samsung/smdkv210/init.rc

2. libreference-ril.so 参数配置

- N710/N720/WM620 配置

  在有效的 system.prop 中指定路径：rild.libpath=/system/lib/libreference-ril.so

  rild.libargs=-d /dev/ttyUSB2

  ttyUSB2 可以是任何 ttyUSBX，X 为任何值，在 RIL 内部会自识别，只需要包含 ttyUSB 即可。

  路径例如：device\rockchip\rk3288\rk3288_box\system.prop

- 2G 模块配置

  在有效的 system.prop 中指定路径：rild.libpath=/system/lib/libreference-ril.so

  rild.libargs=-d /dev/ttySx -b 115200

ttySx 是客户设备中，链接模块的物理串口对应的设备字符

3. 修改服务 ril-daemon 用户权限，多数平台默认已经注释掉了

serivce ril-daemon require root privilege.

comment out the function switchUser() in the file hardware/ril/rild/rild.c. as below:

OpenLib:

#endif

//switchUser();

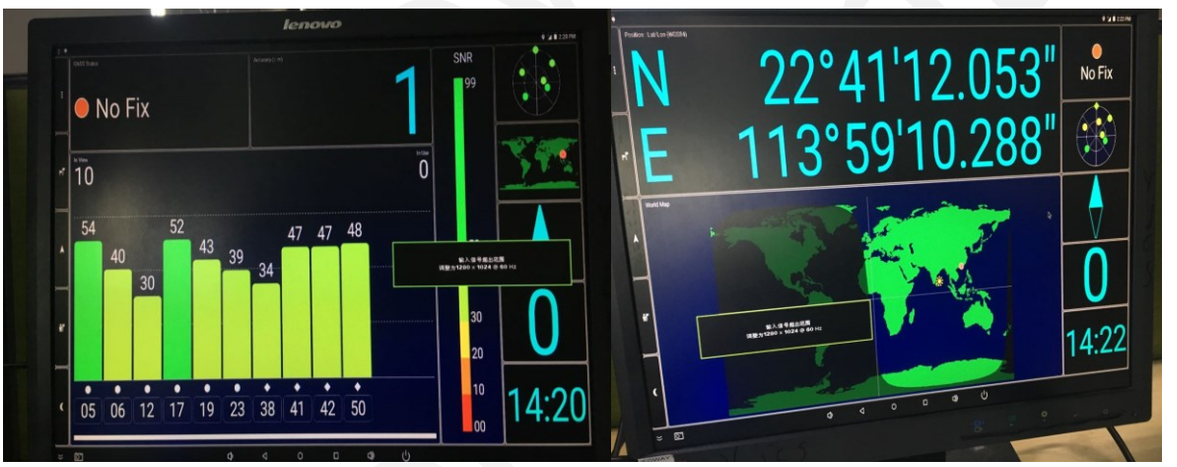
4. 把有方提供的 RIL 库（可能需要更名，与用户定制有关），push 到/system/lib 下，并赋可执行权限（android6.0 的机器将其 push 到/system/lib64）

5. 重启设备

- GPS 库使用

硬件要求：需要模块支持 GPS 功能，并且配备有源天线（现在只支持 GPS，不支持 AGPS）

移植步骤：GPS 库通过 adb push 到 system/lib/hw,保证该目录下无其他 GPS 库，有的话请删除其他的库，只保留有方提供的 gps.default.so（如果是 android6.0，需将 GPS 库防止再 system/lib64/hw 下），如果不起作用，需要将 gps 库放进用户源码编译。



5 FAQ

5.1 如何设置 APN

按上图 1~4 步操作：

1. 打开设置，找到 APN 设置地方；
2. 点击右上角新建 APN；
3. 根据不同运营商设置不同 APN；

   中国移动 2G、3G、4G：CMNET

   中国联通 2G、3G、4G：3GNET

   中国电信 2G、3G：CTNET；4G：CTLTE

4. 设置 MCC MNC。

   MMC MNC 一般是默认的，如果有不支持的 MMC MNC，可以参考如下：

   中国移动：46000、46002、46007

   中国联通：46001、46009

   中国电信：46003、46011、46099、45502、20404

5.2 抓取 log 的方法

1. 捉取 radio log：adb logcat -b radio -v time
2. 捉取 android system 层 log：adb logcat -v time
3. 如果设备与 PC 间是串口连接，进入串口 shell 模式后执行：

   Locat -b radio -v time

   Logcat -v time

   把打印出的 log 存为文件

4. 捉取 gps 的 log：adb logcat –b gps_nwy
5. 捉取完整 log：adb logcat -b main -b system -b radio -b events -v time

5.3 如何分析拨号失败

拨号失败的原因有很多，可以按照下面的思路在 log 里进行分析定位：

1. SIM 卡状态是否正常？

   AT+CPIN？返回值是否是 READY，显示 ERROR 或者 SIM PIN（需要输入 PIN 码），则需要检查硬件上的 SIM 卡槽是否识卡，有没有上电，或者检查 SIM 卡是否正常。

2. 是否注册上语音网络或者数据业务

   搜索“AT+CREG?”和“AT+CGREG”指令，如果是返回 0,1 或者 0,5，则能注册上语音网络或者数据网络，如果是返回其他值，则需要检查 SIM 卡是否欠费，使用的天线有没有问题。

3. PPP 拨号或 NDIS 拨号是否正常？

搜索“**PPP**”或者“**AT^NDISDUP=1,1**”看有没有进行 PPP 拨号或 NDIS 拨号，如果没有，就搜索“**No APN found for carrier**”字符串，出现这个字符串说明不支持这个 apn，需要在 **apns-conf.xml** 里添加进去；没有出现这个字符串就搜索“**setup_data_call**”字符串，出现这个字符串说明已经下发数据业务请求了，没有搜索到，请检查数据业务有没有打开。