GOSUNCN 高新兴物联
Welink your smart

DRIVER

高新兴物联模组
Android系统RIL适配参考

版本：V1.3
日期：2018-09-12

insight WeLink™

Website: www.gosuncnwelink.com
E-mail: welink@gosuncn.com

修订记录

版本	日期	说明
V1.0	2016-12-16	发布初始版本
V1.1	2018-04-29	更新文档模板
V1.2	2018-09-05	完善文档，提高文档易读性
V1.3	2018-09-12	新增第五章问题处理章节 详细无法连接网络时的问题原因分析

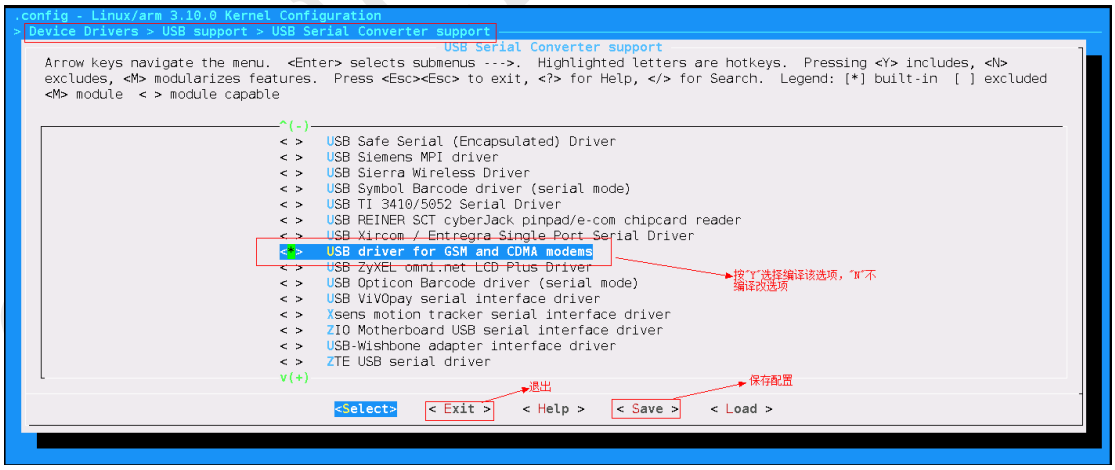
目录

1 RIL 适配

请按照本文档的指导对 Android 系统进行修改，否则 RIL 可能会无法正常运行。

1. **在内核中添加 USB 串口驱动和 USB 网卡驱动**，可以选择将其直接编入内核，或者编译为模块待内核启动时加载，总之，要确保 Linux 内核启动完成后，这两个驱动是运行在内核当中的。
   通常，配置内核是通过指令 make menuconfig，进入 android 系统的 kernel 目录下，执行该指令 make menuconfig 后：
   - 添加 USB 串口驱动：
     device drivers-->
       usb support-->
         usb serial converter support-->
           USB driver for GSM and CDMA modems

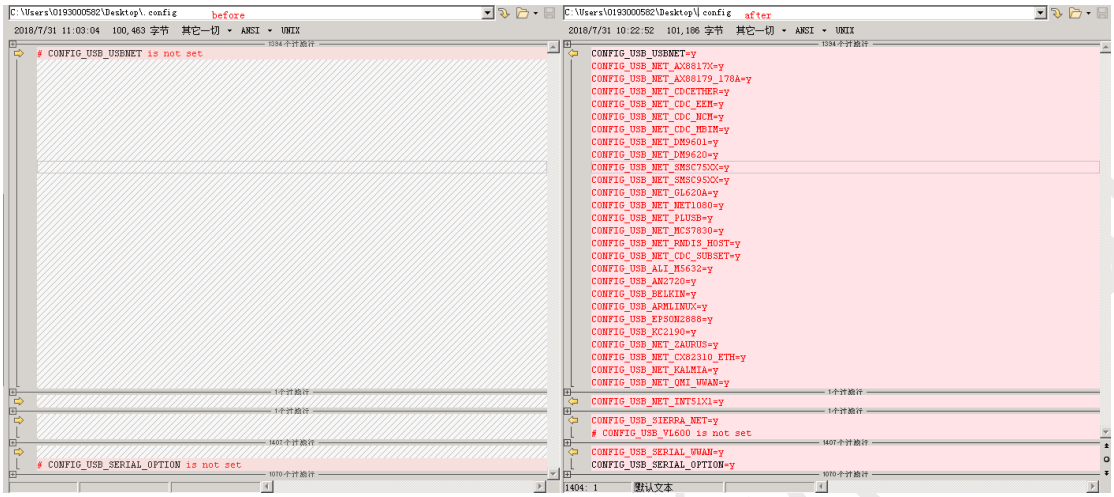
   - 添加 USB 网卡驱动
     devices drivers-->
       Network device support-->
         usb Network Adapters-->
           Mulil-purpose USB Networking Framework



* 注：①如果您的内核结构与上面不一致，可能需要在其它的路径下面选择，总之，只要确保源文件中的 option.c 及其相关的部分（USB 串口驱动），cdc_ether.c 及其相关部分（USB 网卡驱动）参与编译即可。

②如果您的内核不支持 make menuconfig 命令配置，可以直接修改 kernel 目录下

的 .config 文件，手动添加配置项。



2. 在内核中添加 PPP 组件

通常，配置内核是通过指令 make menuconfig，执行该指令后：

devices drivers-->

Network device support-->

ppp support-->

ppp filtering

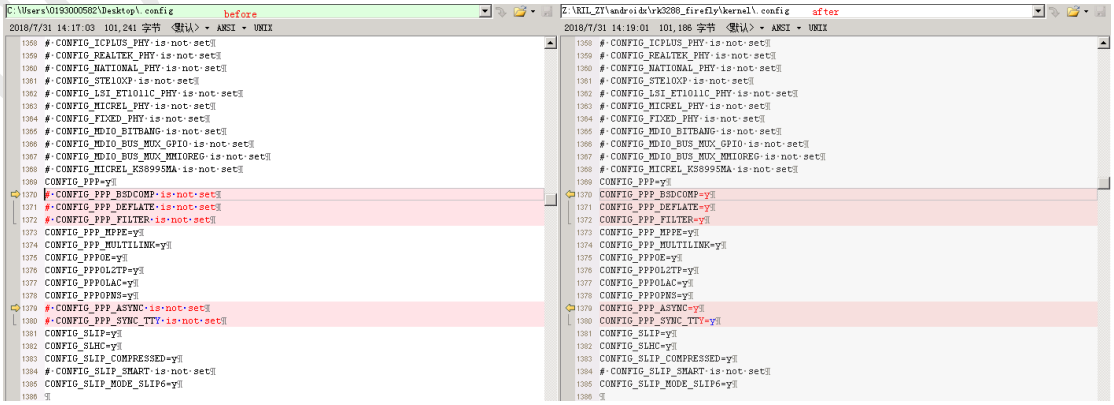
ppp support for async serial ports

ppp support for sync tty ports

ppp deflate compression

ppp BSD-compress compression

* 注： 如果您的内核不支持 make menuconfig 命令配置，可以直接修改 kernel 目录下的 .config 文件，手动添加配置项。

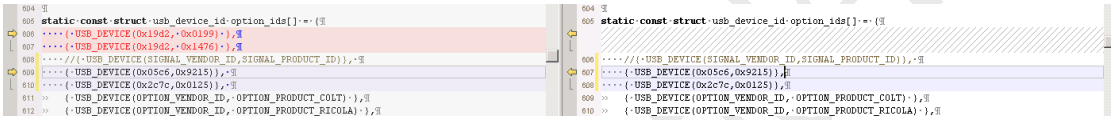


3. **在内核驱动源文件中添加 ZTE 模块相关的信息：**USB 网卡驱动可以自动识别到 GOSUNCN 模块，因此，其对应的 cdc_ether.c 文件中不需要添加任何内容。但是 USB 串口驱动不能

自动识别，必须要添加 GOSUNCN 模块的设备信息到源文件 option.c 中。
文件路径：/kernel/drivers/usb/serial/option.c

- 添加 USB 端口的 VID 和 PID 信息，见下面蓝色部分。这里 0x0199 为 ME3860 模块和 ME3760_V2 模块的 PID，0x1476 为 ME3620 和 ME3630 模块的 PID，如果您使用的是其它模块，将其中的 PID 值更换为相应的值即可。

```
static const struct usb_device_id option_ids[] = {
    { USB_DEVICE(0x19d2, 0x0199) },
    { USB_DEVICE(0x19d2, 0x1476) },
    ……
}
```

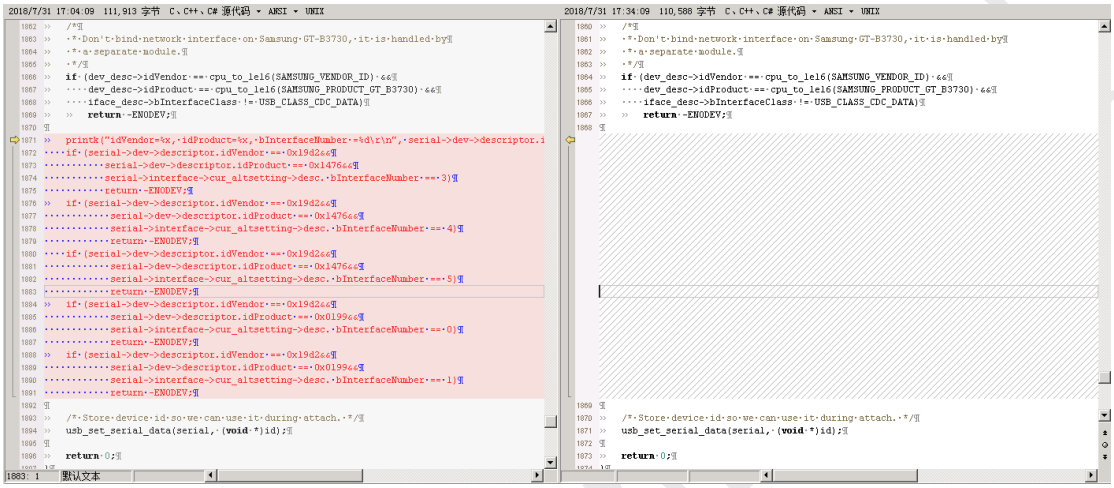


- 添加黑名单信息，上面添加模块信息时只添加了设备的 VID 和 PID，没有附加任何额外的端口信息，这样会导致设备的网卡也被加载成为 USB 串口，下面提供的是一种类似于黑名单的方式，在 option_probe 函数中，将网卡对应的端口加入黑名单，防止 USB 网卡被加载成为 USB 串口。
  对于 ME3860 和 ME3760_V2，其网卡对应的端口为 0 和 1，对于 ME3620 和 ME3630，其网卡对应的端口为 3 和 4。请将以下代码添加到 option_probe 函数中

```
printk("idVendor=%x, idProduct=%x, bInterfaceNumber =%d\r\n",
        serial->dev->descriptor.idVendor,
        serial->dev->descriptor.idProduct,
        serial->interface->cur_altsetting->desc. bInterfaceNumber);

if (serial->dev->descriptor.idVendor == 0x19d2 &&
        serial->dev->descriptor.idProduct == 0x1476 &&
        serial->interface->cur_altsetting->desc. bInterfaceNumber == 3)
        return -ENODEV;
if (serial->dev->descriptor.idVendor == 0x19d2 &&
            serial->dev->descriptor.idProduct == 0x1476 &&
            serial->interface->cur_altsetting->desc. bInterfaceNumber == 4)
            return -ENODEV;
if (serial->dev->descriptor.idVendor == 0x19d2 &&
            serial->dev->descriptor.idProduct == 0x1476 &&
            serial->interface->cur_altsetting->desc. bInterfaceNumber == 5)
            return -ENODEV;
if (serial->dev->descriptor.idVendor == 0x19d2 &&
            serial->dev->descriptor.idProduct == 0x0199 &&
            serial->interface->cur_altsetting->desc. bInterfaceNumber == 0)
            return -ENODEV;
```

```
if (serial->dev->descriptor.idVendor == 0x19d2 &&
        serial->dev->descriptor.idProduct == 0x0199 &&
        serial->interface->cur_altsetting->desc. bInterfaceNumber == 1)
        return -ENODEV;
```



* 注：第一行的 printk 是为了方便调试而打印的，虽无实际效果，最好能带上。下面的几个 if 语句分别判断了需要加入黑名单的端口号，如果您使用的是除 ME3860，ME3760_V2 和 ME3620 之外的模块，上面 if 语句中的判断条件也要做相应修改。

4. 在 Android 系统的 init.rc 中添加服务

文件路径：一般情况下，路径在../system/core/rootdir/init.rc，注意：有些平台厂商维护了自己的 init.rc，因此修改该目录下的 init.rc 可能不会生效。厂商维护的 init.rc 所在路径可能位于 device/xxx/yyy 或者 vendor/xxx/yyy 目录下。其中 xxx 表示平台提供厂商， yyy 表示具体平台型号，如 freescale 的 imx_51bbg 开发平台中，init.rc位于目录\device\fsl\imx51_bbg 下。

- ril-daemon 服务 （添加前注释掉原来的 ril-daemon 服务）

```
service ril-daemon /system/bin/rild -l /system/lib/libreference-ril-gosuncn.so
    socket rild stream 660 root radio
    socket rild-debug stream 660 radio system
    user root
    group radio cache inet misc audio log
```

- pppd_gprs 服务

service pppd_gprs /system/etc/init.gprs_pppd
user root
group radio cache inet misc log
disabled
oneshot

```
#service ril-daemon /system/bin/rild
#    class main
#    socket rild stream 660 root radio
#    socket rild-debug stream 660 radio system
#    user root
#    group radio cache inet misc audio log
service ril-daemon /system/bin/rild -l /system/lib/libreference-ril-gosuncn.so
    socket rild stream 660 root radio
    socket rild-debug stream 660 radio system
    user root
    group radio cache inet misc audio log

service pppd_gprs /system/etc/init.gprs_pppd
user root
    group radio cache inet misc log
    disabled
    oneshot
```

5. 打包相关文件到系统中

将下表中的几个文件打包到系统中，确保系统编译完成后，文件出现在对应的目录中。

如果 Android 系统不支持 netcfg 命令，需要将 netcfg 文件放到 Android 系统中的/system/bin 目录下。

文件名	打包后在 Android 系统中的目录
libreference-ril-gosuncn.so	/system/lib
netcfg	/system/bin
init.gprs_pppd	/system/etc
Ip-up-ppp0	/system/etc/ppp
ip-down-ppp0	/system/etc/ppp

ril库适配文档. zip

* 注：将文件导入 android 系统的命令：adb push D:\ libreference-ril.so /system/lib，如果系统提示“read-only file system”，先执行
adb root
adb remount
再 push 文件。

6. 配置 APN

如果 android 系统没有设置过 APN,，需要在 setting->wireles&networks->cellular networks->access point names 配置与 sim 卡匹配的 APN 信息，其中

电信：ctnet

联通：3gnet

移动：cmnet

7. 验证

成功：显示驻网信息，并且能连接互联网。

失败：执行 adb logcat -b radio -v time >D:11/log 打印 log 继续分析

2 可选配置

本节介绍了几种可以选择的配置，用户可以通过这些配置来控制或者获取 RIL 的行为。一般情况下，按照第一节的内容修改 Android 系统之后，RIL 已经可以正常运行，因此本节内容主要用于调试，如果没有特殊需要可以忽略，设置属性值的命令 setprop，查看属性值的命令 getprop。

1. **通过系统属性设置 RIL 连接网络时的拨号方式（ECM 拨号或者 PPP 拨号）**
   属性名：ril.dial.mode
   说明：该属性值为 0 代表使用 PPP 拨号，为其它值则使用 ECM 拨号
   默认值：1
2. **通过系统属性设置 PPP 拨号失败时的最大重连次数**
   属性名：ril.ppp.retry.times
   说明：该属性值为一个整数，必须在 1-5 之间，超出范围会被强制设置为 1
   默认值：1
3. **通过系统属性设置 RIL 是否主动给系统上报信号强度**
   属性名：ril.unsol.signal
   说明：该值为 0 代表 RIL 不主动上报信号强度，为其它值代表则主动上报信号强度
   默认值：1
4. **通过系统属性设置心跳指令的发送周期**
   属性名：ril.keepalive.circle
   说明：RIL 每隔一定时间会给模块发送心跳指令确保模块可用，如果周期为 0 则不发送
   默认值：9（单位为秒）
5. **通过系统属性设置 RIL 检查网络连接状态的周期**
   属性名：ril.check.network.circle
   说明：数据业务打开时，RIL 每隔一定时间会检查网络状态，如果周期为 0 则不检查
   默认值：60（单位为秒）

3 运行日志的抓取

RIL 调试过程中不可避免地会遇到很多问题，分析问题需要抓取相应的日志，下面是几种常用的日志的抓取方法

1. 抓取 Radio log
   adb logcat -b radio -v time

2. 抓取 Android 系统日志
   adb logcat –v time

3. 抓取内核日志
   adb shell dmesg
   或者
   adb shell cat /proc/kmsg

4. 抓取连网过程中 pppd 程序的日志
   adb logcat -s pppd
   * 这个是分析连网问题时使用的，如果使用的是 PPP 拨号，则抓取 pppd 日志，系统默认使用 ECM 拨号，可以通过系统属性 ril.dial.mode 修改，参见“可选配置”章节。

5. 抓取模块内部的运行日志
   ㈠ 抓取模块 AP 侧日志，有两种方法：
   ① 方法一：

   push android 侧的 ADB 插件到/system/bin 目录下，操作如下：

   adb remount　　//打开 android 系统读写权限

   adb push D:\Android_cmd\adb /system/bin　　//导入下载插件到/system/bin 目录下

   adb shell　　//进入 adb 环境下

   chmod 777 /system/bin/adb　　//给下载插件权限

   mkdir /data/ welinklogs　　//先 adb remount 然后在/data 目录下 mkdir /welinklogs

   echo –e "at+zadset=d" > /dev/ttyUSB1　　//切换出模块的 ADB 口

   adb logcat –v time > /data/welinklogs/ap.txt　　//准备好后就可以抓取 log

```
E:\customer\sis\platform-tools_adb>adb shell
sabresd_6dq:/ # adb logcat -v time > /data/welinklogs/zm5330s_ap.txt
^C
```

//退出 Android ADB 环境，执行 exit 再 pull 文件到本地

```
E:\customer\sis\platform-tools_adb>adb pull /data/welinklogs/zm5330s_ap.txt D:/y
fve
/data/welinklogs/zm5330s_ap.txt: 1 fil.... 12.8 MB/s (95599440 bytes in 7.150s)

E:\customer\sis\platform-tools_adb>
```

② 方法二：

push android 侧的 ADB 插件到/system/bin 目录下，操作如下：

adb remount //打开 android 系统读写权限

adb push D:\Android_cmd\adb /system/bin //导入下载插件到 /system/bin 目录下

adb shell //进入 adb 环境下

chmod 777 /system/bin/adb //给下载插件权限

mkdir /data/ welinklogs //先 adb remount 然后在/data 目录下 mkdir /welinklogs

echo –e " at+zlogcatd=on " > /dev/ttyUSB1 //下发 AT 指令打开 AP 侧 log 抓取，

echo –e " at+zrst " > /dev/ttyUSB1 //重启模块，zlogcatd 命令重启生效

echo –e " at+zlogcatd=off " > /dev/ttyUSB1 //在打开 AP 侧 log 抓取后，复现所需要的场景，复现完成后，请手动下发“at+zlogcatd=off”指令关闭 AP 侧 log 抓取

echo –e "at+zadset=d" > /dev/ttyUSB1 //切换出模块的 ADB 口

adb pull /data/logs / /data/welinklogs/ //导出 log 文件，先导出到 android 机器上，再导出到本地 windows 机器上

```
root@firefly:/data # mkdir welinklogs
mkdir welinklogs
root@firefly:/data # adb pull /data/logs/ /data/welinklogs
adb pull /data/logs/ /data/welinklogs
pull: building file list...
pull: /data/logs/logcat.log -> /data/welinklogs/logcat.log
1 file pulled. 0 files skipped.
828 KB/s (327880 bytes in 0.386s)
root@firefly:/data # exit
exit
```

㈡ 抓取模块侧 modem 日志

push android 侧的 bp_log & config 插件到/data/welinklogs 目录下，操作如下;

adb remount //打开 android 系统读写权限

adb push D:\Android_cmd\bp_log /data/welinklogs

adb push D:\Android_cmd\config /data/welinklogs //导入下载插件到/system/bin 目录下

adb shell //进入 adb 环境下

chmod 777 / data/welinklogs/bp_log

chmod 777 / data/welinklogs/config //给下载插件权限

在/ data/welinklogs/目录下执行 ./bp_log，抓取 QXDM 日志（提示没打开端口的话就多尝试几次）

将抓取的 QXDM 文件就全部导出来，包括.head，发给相关人员分析

```
sabresd_6dq:/ # cd /data/welinklogs/
sabresd_6dq:/data/welinklogs # ls
bp_log config zm5330s_ap.txt zm5330s_ap_1.txt zm5330s_kernel.txt
sabresd_6dq:/data/welinklogs # ./bp_log
diaglog version:LINUX_BP_LOGV1.00.03
detecting diag port......
no diag port!
sabresd_6dq:/data/welinklogs # ./bp_log
diaglog version:LINUX_BP_LOGV1.00.03
detecting diag port......
diag port name : /dev/ttyUSB0
sending diag setting command......
start to capture log
input 'q' to quit application.
Total read: 8001, Write to file: 4095
q
sending diag stop command......
Total read: 8803184, Write to file: 8803184
sabresd_6dq:/data/welinklogs # ls -l
total 243280
-rw-rw-rw- 1 root root    18913 2018-05-27 14:36 2018-05-27-14-36-04.zm
-rw-rw-rw- 1 root root  8803184 2018-05-27 14:36 2018-05-27-14-36-05.log
-rw-rw-rw- 1 root root        8 2018-05-27 14:36 2018-05-27-14-36-05.log.head
-rwxrwxrwx 1 root root    30420 2018-05-11 15:19 bp_log
-rwxrwxrwx 1 root root   144049 2018-03-20 17:02 config
-rw-rw-rw- 1 root root 95599440 2018-05-27 11:51 zm5330s_ap.txt
-rw-rw-rw- 1 root root 19880773 2018-05-27 14:12 zm5330s_ap_1.txt
-rw-rw-rw- 1 root root    65001 2018-05-27 09:56 zm5330s_kernel.txt
sabresd_6dq:/data/welinklogs #
```

6. 通过 AT 指令直接查询模块内部的运行状态

   指令执行过程如下：

   adb shell　　//进入 Linux 环境

   stop ril-daemon　　//停止 RIL

   cat /dev/ttyUSB1　　//监听 AT 口

   重新打开一个窗口，echo –e " AT 指令" > /dev/ttyUSB1　　//向 AT 口发送 AT 命令

4 Radio log 中简单问题的处理

本节介绍 radio log 中经常会出现的一些问题，内容正在完善中，仅供参考

1. **判断 RIL 库是否已经加载并运行**

   ******** Enter RIL_Init() ********

   RIL 库开始运行时会打印上述日志，从 Android 系统开机起抓取 radio log，如果不存在上述字样，则说明 RIL 库没有正确加载。需要检查 ril-daemon 服务是否添加并正确运行。

2. **判断无线通信模块端口是否正常**

   [init]Runtime 3G can't find supported modem　//未找到设备

   [init]Runtime 3G port found matched device with xxxxxxx　//找到设备

   如果 radio log 不停地打印前者，说明 RIL 无法在系统中找到无线通信模块，需要检查模

块是否正常上电，并且正常加载驱动。模块端口正常后，会打印后者。

3. 判断模块是否已经注册到运营商的网络

在 radio log 中搜索如下几个 AT 指令的返回结果：

AT+CPIN? //SIM 卡状态
AT+CSQ //信号强度
AT+ZPAS? //注册状态

关于指令结果的含义可以参考模块的 AT 指令集，下面仅给出例子。

AT+CPIN?
+CPIN: READY
OK
* 如果指令的返回结果中有 ERROR 字样，则说明 SIM 卡有问题

AT+CSQ
+CSQ: 31,0
OK
* 如果指令的返回结果中有 ERROR 字样，说明模块状态有问题
* 如果第一个数字为个位数或者为 99，则说明信号非常差

AT+ZPAS?
+ZPAS: "LTE","CS_PS"
OK
* 如果指令的返回结果中有 ERROR 字样，说明模块状态有问题
* 如果指令返回结果中有 no service 或者 limit service，则说明模块未注册上网络

4. 无法连接网络时的问题

如果在 Android 系统界面上没有显示网络连接成功的图标，首先确认：

(1) SIM 卡业务功能是否正常，确保 SIM 卡可以正常上网。
(2) 是否已经注册上运营商的网络（参加本节第 3 条）
(3) Android 系统中是否有对应的网络接入点（APN）
(4) 模块已经获取到 IP 地址，但是给系统 USB0 或者 eth0 分配 IP 地址时出错。
   a. 在 radio 日志中搜索 AT+ZECMCALL?的返回值，如下说明模块拨号成功
      AT< +ZECMCALL: IPV4, 10.76.145.181, 10.76.145.182, 120.80.80.80, 221.5.88.88
   b. 在 radio 日志中搜索 netcfg usb0 dhcp end ，查看如下的 status 值，如果为非零值，说明执行获取 IP 地址的 netcfg 命令失败。
      ===netcfg usb0 dhcp end : status = 32512===
   c. 出现 status 值为非 0 值时，需要手动在 Android 系统的 adb shell 中手动输入 netcfg usb0 dhcp 检查 netcfg 是否能在客户的 Android 系统上执行成功。
   d. 如果执行出错，则需要客户使用我们提供的 netcfg 源码重新编译 netcfg 文件，并替换当前使用的 netcfg 文件。

在上述几项没有问题的情况下，抓取日志分析：

(1) radio log 中搜索“SETUP_DATA_CALL ********”字样，这里会启动连接网络的流程，可以看到后续流程中出现了何种错误

(2) 抓取 pppd 或者 dhcpcd 日志（参见“运行日志抓取”章节中的第 4 条），其中可以看到拨号过程中发生了何种错误。如果其中内容只有几条，则说明 pppd 或者 dhcpcd 没有运行起来。

5 常见问题及处理

Q1：使用 ME3630E1C 模块用联通卡测试，整机拨号正常可以获取 ip，但 ping 不通网络，且重启整机现象依旧存在，无法 ping 通网络。

问题分析：ping 不通网络，首先查看模块是否注册上网络，拨号是否成功，SIM 卡是否正常，客户反馈 SIM 卡未欠费可以打电话。

```
12-31 19:17:14.600 I/RIL      ( 2536): ******** 558, onRequest(27): SETUP_DATA_CALL ********
12-31 19:17:14.600 D/AT       ( 2536): AT> AT+ZPAS?
12-31 19:17:14.603 D/AT       ( 2536): AT< +ZPAS: "LTE","CS_PS","FDD"
12-31 19:17:14.605 D/RIL      ( 2536): the SIM card isn't China Telecom
12-31 19:17:14.622 D/AT       ( 2536): AT> AT+ZECMCALL?
12-31 19:17:14.627 D/AT       ( 2536): AT< +ZECMCALL: IPV4, , , ,
12-31 19:17:14.627 D/AT       ( 2536): AT< OK
12-31 19:17:14.627 D/AT       ( 2536): AT> AT+CGDCONT=1,"IPV4V6","3gnet",,0,0
12-31 19:17:14.653 D/AT       ( 2536): AT< OK
12-31 19:17:14.653 D/AT       ( 2536): AT> AT+ZPCONTEXT=1
12-31 19:17:14.657 D/AT       ( 2536): AT< OK
12-31 19:17:16.657 D/AT       ( 2536): AT> AT+ZECMCALL=1
12-31 19:17:19.838 D/AT       ( 2536): AT< +ZECMCALL: CONNECT
12-31 19:17:19.838 D/RIL      ( 2536): ******** Enter onUnsolicited() ********
12-31 19:17:19.838 E/RIL      ( 2536): [UNSOL] Got a response no one expected: +ZECMCALL: CONNECT
12-31 19:17:19.838 D/RIL      ( 2536): ******** Leave onUnsolicited() ********
12-31 19:17:19.838 D/AT       ( 2536): AT< OK
12-31 19:17:20.838 D/AT       ( 2536): AT> AT+ZECMCALL?
12-31 19:17:20.844 D/AT       ( 2536): AT< +ZECMCALL: IPV4, 10.64.223.169, 10.64.223.170, 120.80.80.80, 221.5.88.88
12-31 19:17:20.844 D/AT       ( 2536): AT< OK
12-31 19:17:20.844 W/RIL      ( 2536): [DATAjjh]p_response->p_intermediates->line is +ZECMCALL: IPV4, 10.64.223.169,
12-31 19:17:31.993 D/RIL      ( 2536): ++++++++++++ipadress = 10.64.223.169,len = 13
12-31 19:17:31.994 D/RIL      ( 2536): ++++++++++++dnses = 120.80.80.80,len = 12
12-31 19:17:31.994 D/RIL      ( 2536): ++++++++++++gateways = 10.64.223.170,len = 13
12-31 19:17:31.994 D/RIL      ( 2536): ------ipaddress-- ok
12-31 19:17:31.994 D/RIL      ( 2536): ------dnses-- ok
12-31 19:17:31.994 D/RIL      ( 2536): ------gateways-- ok
12-31 19:17:31.994 D/RIL      ( 2536): ALL IS OK
```

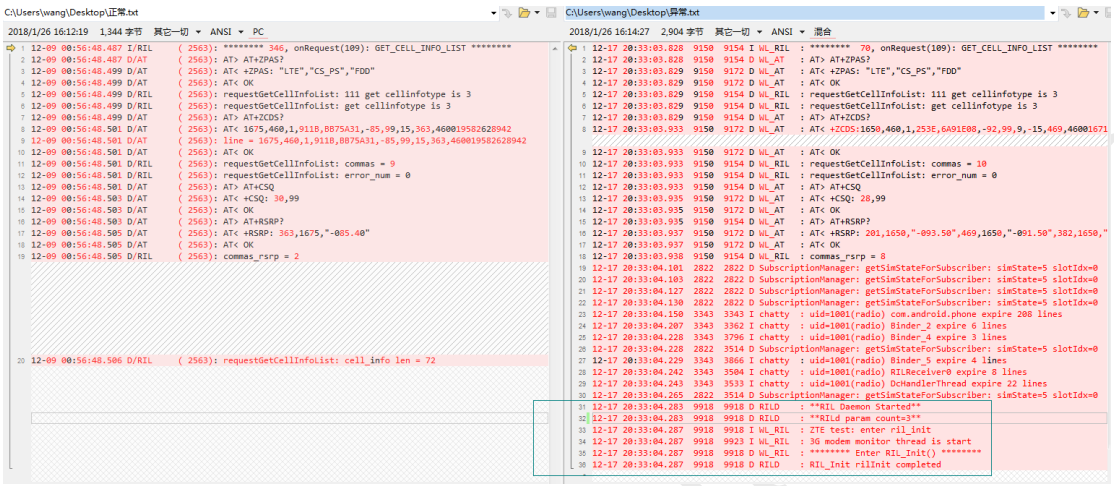
从 log 看模块注册到了 LTE 网络，ZECMCALL 拨号成功，获取到了 IP 地址，RIL 执行正常没有问题；这个时候需要确认下是模块的问题还是 RIL 的问题，连接模块的 debug 口，直接 ping ip 地址测试网络；客户反馈，通过模块的 debug 口也无法 ping 通网络。可以初步确定问题在模块端，但是从 log 打印的信息看，模块运行正常没有发现异常信息，排除模块自身的问题。进一步排查 SIM 卡的问题，包括 SIM 卡的类型，业务范围和业务类型等问题。

问题原因：后来经过问题排查，发现是由于客户的 SIM 卡套餐比较特殊，余额少于一块钱时，能打通电话但是无法上网。

Q2：基站定位问题使用新版本 ril 库测试获取到的基站信息为空，且会出现获取信息时 4G 图标消失几秒，然后再重新拨号，附件是相关 log。

问题分析：Android 系统获取基站信息时会下发”onRequest(109): GET_CELL_INFO_LIST”请求，

RIL 会下发 AT+ZCDS 和 AT+RSRP 两个 AT 命令查询。从 log 看模块在下发 AT+RSRP 后，处理返回结果时 RIL 重启了，正常情况下是不会重启的。可以确定是 RIL 的问题。



问题原因：该问题是由于 RIL 代码中打印了异常地址的信息，导致 ril 重启，所以 4G 图标消失几秒，然后重新拨号。

Q3：客户反馈 3G，2G 模式下获取不到基站信息，4G 模式下可以。

问题分析： Android 系统获取基站信息时会下发”onRequest(109):GET_CELL_INFO_LIST”请求，从 log 可以看到下发 AT+RSRP 后，命令返回为空，导致 RIL 处理异常，“rsrp_line_err_num=100”。该问题是由于模块在 3G,2G 模式下暂不支持用 RSRP 命令查询基站信息。

```
6732
6733  01-07 23:31:38.033  2595  2598 I WL_RIL  : ******** 171, onRequest(109): GET_CELL_INFO_LIST ********
6734  01-07 23:31:38.033  2595  2598 D WL_AT   : AT> AT+ZPAS?
6735  01-07 23:31:38.039  2595  2603 D WL_AT   : AT< +ZPAS: "UMTS","CS_PS"
6736  01-07 23:31:38.039  2595  2603 D WL_AT   : AT< OK
6737  01-07 23:31:38.039  2595  2598 D WL_RIL  : requestGetCellInfoList: 111 get cellinfotype is 4
6738  01-07 23:31:38.039  2595  2598 D WL_RIL  : requestGetCellInfoList: get cellinfotype is 4
6739  01-07 23:31:38.039  2595  2598 D WL_AT   : AT> AT+ZCDS?
6740  01-07 23:31:38.044  2595  2603 D WL_AT   : AT< +ZCDS:10663,460,1,A52A,D9FC0B3,199,99,0,0,460016717625051
6741  01-07 23:31:38.044  2595  2603 D WL_AT   : AT< OK
6742  01-07 23:31:38.044  2595  2598 D WL_RIL  : +ZCDS cellinfo_line = 10663,460,1,A52A,D9FC0B3,199,99,0,0,460016717625051
6743  01-07 23:31:38.044  2595  2598 D WL_RIL  : requestGetCellInfoList: commas = 9
6744  01-07 23:31:38.044  2595  2598 D WL_RIL  : requestGetCellInfoList: error_num = 0
6745  01-07 23:31:38.044  2595  2598 D WL_AT   : AT> AT+CSQ
6746  01-07 23:31:38.050  2595  2603 D WL_AT   : AT< +CSQ: 19,99
6747  01-07 23:31:38.050  2595  2603 D WL_AT   : AT< OK
6748  01-07 23:31:38.050  2595  2598 D WL_AT   : AT> AT+RSRP?
6749  01-07 23:31:38.054  2595  2603 D WL_AT   : AT< +RSRP:
6750  01-07 23:31:38.054  2595  2603 D WL_AT   : AT< OK
6751  01-07 23:31:38.054  2595  2598 D WL_RIL  : commas_rsrp = 0 rsrp_line=
6752  01-07 23:31:38.054  2595  2598 D WL_RIL  : rsrp_line err_num= 100
```

问题原因：修改代码逻辑，RSRP 返回为空时，继续上报+ZCDS 命令的返回结果。