



Fibocom 广和通

完 美 无 线 体 验

NL668T

RIL 适配指南_Android

V1.0

免责声明

客户须按照文档中提供的信息来设计其产品。因未能遵守有关操作、设计规范或安全规则而造成的损害，本公司不承担任何责任。由于产品版本升级或其他原因，本公司保留随时修改本文档中任何信息的权利，无需提前通知且不承担任何责任。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。

版权声明

版权所有 ©2023 深圳市广和通无线股份有限公司。本公司保留一切权利。

除非本公司特别授权，文档的接收方须对所接收的文档和信息保密，不得将其用于除本项目的实施与开展以外的任何其他目的。非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。对于任何违反保密义务、未经授权使用或以其他非法形式恶意使用所述文档和信息的违法侵权行为，本公司有权追究法律责任。

商标声明

 为深圳市广和通无线股份有限公司的注册商标，由所有人拥有。

本文档中出现的其他商标、产品名称、服务名称以及公司名称，由其各自的所有人拥有。

联系方式

网站：<https://www.fibocom.com>

地址：深圳市南山区西丽街道西丽社区打石一路深圳国际创新谷六栋 A 座 10-14 层

电话：0755-26733555

安全须知

在没有适当设备认证的情况下，不要在不建议使用无线电的区域操作无线通信产品。这些区域包括可能产生无线电干扰的环境，如易燃易爆环境、医疗设备、飞机或可能受到任何形式无线电干扰的任何其他设备。

任何车辆的驾驶员或操作员在控制车辆时不得操作无线通信产品。这样做会降低驾驶员或驾驶员对车辆的控制和操作，带来安全风险。

无线通信设备并不保障在任何情况下都能有效连接，例如在(U)SIM 卡无效或设备欠费时。在紧急情况下，请在开机状态下使用紧急呼叫功能，同时确保设备位于信号强度足够的区域。

目录

修订记录.....	3
1 引言	4
2 Android RIL 驱动概述.....	5
2.1 RIL 驱动支持的功能.....	5
2.2 支持的 Android 版本	5
3 RIL 集成配置	7
3.1 配置 Linux 内核串口驱动.....	7
3.2 配置 Android 启动脚本	7
3.2.1 对于 Android8 以前版本.....	7
3.2.1.1 32 位设备	7
3.2.1.2 64 位设备	7
3.2.2 对于 Android8 及以后版本.....	8
3.2.2.1 32 位设备	8
3.2.2.2 64 位设备	8
3.2.3 参数说明	8
3.3 修改 RILD 权限	8
3.4 加载 RIL 库文件.....	9
4 RIL 调试方法	11
4.1 抓取 RIL 日志.....	11
4.2 RIL Log 中 TAG 标签说明	11
5 FAQ.....	12
5.1 RIL 没有正常工作.....	12
5.2 拨号没有成功如何处理?	12
5.3 如何获取 ECM 的 IP、DNS 等信息?	13
5.4 如何使用 PPP 拨号方式?	14
5.4.1 切换 PPP 拨号方式	14
5.4.2 PPP 拨号没有成功如何处理?	14
5.4.3 如何获取 PPP 拨号后的 IP、DNS 等信息?	16
5.4.4 设备不支持 PPP 拨号的解决方法	17
5.5 电话打不通	19
5.6 无法发送短信	20

5.7 如何设置 APN?	20
5.7.1 手动添加 APN.....	20
5.7.2 配置 apns-conf.xml 文件	20
6 自定义系统属性	22
6.1 属性设置方法	22
6.2 ril.fibocom.dialmode	24
6.2.1 简介	24
6.2.2 取值范围	24
6.2.3 示例	25
6.3 ril.fibocom.NetifName	25
6.3.1 作用	25
6.3.2 取值范围	25
6.3.3 示例	26
6.4 ril.fibocom.cid	26
6.4.1 作用	26
6.4.2 取值范围	26
6.4.3 示例	27
6.5 ril.fibocom.cgdcont0.....	27
6.5.1 作用	27
6.5.2 取值范围	27
6.5.3 示例	27
6.6 ril.fibocom.usbmode	28
6.6.1 作用	28
6.6.2 取值范围	28
6.6.3 示例	28
7 附录 A 参考文件	29

修订记录

V1.0 (2023-01-29) 初始版本

1 引言

本文主要介绍如何将 RIL（无线接口层）驱动程序集成到客户设备以及如何修改相关配置文件。

2 Android RIL 驱动概述

2.1 RIL 驱动支持的功能

表 1. RIL 支持的功能

功能	是否支持
SMS	YES
VOICE CALL	YES
DATA SERVICE	YES
SIM TOOL KIT	NO

2.2 支持的 Android 版本

目前，Fibocom RIL driver 支持以下的 Android 版本，见下表：

表 2. RIL 支持 android 系统版本

功能	是否支持
Android 2.x	NO
Android 3.x	NO
Android 4.x	YES
Android 5.x	YES
Android 6.x	YES
Android 7.x	YES
Android 8.x	YES
Android 9.x	YES
Android 10.x	YES

Android 11.x

YES

3 RIL 集成配置

本章节介绍 RIL 库集成到客户 Android 系统的操作步骤，包括配置内核驱动、修改 Android 系统 RILD 执行权限和启动脚本、加载 RIL 库文件等。

3.1 配置 Linux 内核串口驱动

集成模块驱动时，请参考《Fibocom_NL668T_驱动安装指南_Linux》文档。

3.2 配置 Android 启动脚本

3.2.1 对于 Android8 以前版本

3.2.1.1 32 位设备

配置 init.rc 文件，添加 ril-daemon 进程：

```
service ril-daemon /system/bin/rild -l /system/lib/libreference-ril.so -- -w 1
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
```

3.2.1.2 64 位设备

配置 init.rc 文件，添加 ril-daemon 进程：

```
service ril-daemon /system/bin/rild -l /system/lib64/libreference-ril.so -- -w 1
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
```

3.2.2 对于 Android8 及以后版本

3.2.2.1 32 位设备

配置 rild.rc 文件，添加 ril-daemon 进程：

```
service ril-daemon /vendor/bin/hw/rild -l /vendor/lib/libreference-ril.so -- -w 1
    class main
    user root
    group radio cache inet misc audio log readproc wakelock
    capabilities BLOCK_SUSPEND NET_ADMIN NET_RAW
```

3.2.2.2 64 位设备

配置 rild.rc 文件，添加 ril-daemon 进程：

```
service ril-daemon /vendor/bin/hw/rild -l /vendor/lib64/libreference-ril.so -- -w 1
    class main
    user root
    group radio cache inet misc audio log readproc wakelock
    capabilities BLOCK_SUSPEND NET_ADMIN NET_RAW
```

3.2.3 参数说明

init.rc(或 rild.rc)脚本里，-l 和 -w 选项的作用如下：

- -l 指定 RIL 库加载路径
- -w 指定拨号方式，0 ： ppp 拨号，1 ： ECM 拨号

3.3 修改 RILD 权限

RILD 程序执行过程中需要 root 权限，可在 rild.c (<\$android dir> /hardware/ril/rild/rild.c) 中的 main 函数中，去掉 switchuser 函数的调用来达到此目的，如下图 1 所示

```
224     arg_device[sizeof(arg_device)-1] = 0;
225     arg_overrides[1] = "-d";
226     arg_overrides[2] = arg_device;
227     done = 1;
228
229     } while (0);
230
231     if (done) {
232         argv = arg_overrides;
233         argc = 3;
234         i = 1;
235         hasLibArgs = 1;
236         rilLibPath = REFERENCE_RIL_PATH;
237
238         |   RLOGD("overriding with %s %s", arg_overrides[1], arg_overrides[2]);
239     }
240 }
241 OpenLib:
242 #endif
243 //switchUser();
244
245 dlHandle = dlopen(rilLibPath, RTLD_NOW);
246
247 if (dlHandle == NULL) {
248     RLOGE("dlopen failed: %s", dlerror());
249     exit(-1);
250 }
251
```

图 1. RILD 中 main 函数修改



部分安卓版本中没有 switchUser 函数，如 Android10，可忽略该节内容。

3.4 加载 RIL 库文件

```
adb root
```

```
adb remount
```

如果是 Android 8 以前，32 位系统

```
adb push <local-dir>/libreference-ril.so /system/lib
```

如果是 Android 8 以前，64 位系统

```
adb push <local-dir>/libreference-ril.so /system/lib64
```

如果是 Android 8 及以上，32 位系统

```
adb push <local-dir>/libreference-ril.so /vendor/lib
```

如果是 Android 8 及以上，64 位系统

```
adb push <local-dir>/libreference-ril.so /vendor/lib64
```

然后重启安卓设备，预期此时设备能自动拨号成功。

4 RIL 调试方法

4.1 抓取 RIL 日志

在 Windows 操作系统的 cmd 窗口，输入如下命令：

```
adb logcat -b all -v time > radio.txt
```

4.2 RIL Log 中 TAG 标签说明

RIL log 中各个进程日志都在同一个文件中，可以通过 tag 标签来区分 log 来自于哪个文件，TAG 标签与文件的对应关系见下表。

表 3. RIL log 中 TAG 标签说明

TAG	文件
GHT-RIL	hardware/ril/reference-ril/reference-ril.c
GHT-AT	hardware/ril/reference-ril/atchannel.c
RILC	hardware/ril/libril/ril.cpp
RILD	hardware/ril/rild/rild.c
RILJ	frameworks/opt/telephony/src/java/com/android/internal/telephony/RIL.java

5 FAQ

5.1 RIL 没有正常工作

有很多原因导致 Android 设备无法启动 RIL，如果出现这类现象请按照以下流程进程排查：

1. RIL 库没有被正确的加载

进入 Android shell 终端，然后检查 init.rc 或 rild.rc 中是否有 [3.2](#) 节的内容。如果没有，按照 3.2 节的步骤添加对应内容，以便加载正确的库文件。

2. RILD 是否正在运行

进入 Android shell 终端：输入 “getprop | grep init.svc.ril-daemon”；检测 RIL 的运行状态，期待的结果：
[init.svc.ril-daemon]: [running]

3. USB 串口是否被枚举

进入 Android shell 终端：输入 ls /dev/ttyUSB* -l 以检测 usb 串口是否被枚举。如果没有枚举端口，RIL 也无法正常工作。

5.2 拨号没有成功如何处理？

1. 先用 adb shell 命令进入安卓设备，然后用如下命令确认是否识别出了端口。如果没有，说明上位机没有识别到模组，请检查硬件电路：

```
root@android:/ # ls /dev/ttyUSB*
```

 ① 若设备正常挂载，将会有如下内容返回：

```
ls /dev/ttyUSB*
```

```
/dev/ttyUSB0
```

```
/dev/ttyUSB1
```

```
/dev/ttyUSB2
```

```
/dev/ttyUSB3
```

2. 检查使用 dmesg | grep cdc 命令获取的网卡名称和 RADIO log 中“configuring xxx”的网卡名称是否相同，如下图。如果二者不同，在/system/build.prop 设置属性

```
ril.fibocom.NetifName=xx
```

“xx”是 dmesg | grep cdc 查到的结果，如下图中的“eth1”：

```

17219 11-29 16:56:26.498 300 300 D DHCP : ip 10.83.105.230 gw 10.83.105.1 prefixLength 24
17220 11-29 16:56:26.498 300 300 D DHCP : dns1: 211.137.130.2
17221 11-29 16:56:26.498 300 300 D DHCP : server 192.168.1.1, lease 30840 seconds
17222 11-29 16:56:26.498 300 300 D DHCP : configuring eth1
17223 11-29 16:56:26.533 0 0 E selinux : avc: denied { set } for property=net.eth1 dns1 pid=300 uid=0 gid=10
17224 11-29 16:56:26.528 300 300 D GHT_RIL : fibo_bring_up_interface_do_dhcp ret = 0, retry = 1
17225 11-29 16:56:26.529 300 300 D GHT_RIL : fibo_get_ip: 10.83.105.230, if_name:eth1, strlen(ifr.ifr_name):4
17226 11-29 16:56:26.529 300 300 D GHT_RIL : fibo_get_ip: get hwaddr[0]:f0
17227 11-29 16:56:26.529 300 300 D GHT_RIL : fibo_get_ip: get hwaddr[1]:4b

[ril.fibocom.version]: [Fibocom_RIL_V10X.06.V1.0.6_004]
[ril.function.dataonly]: [1]
[rild.libargs]: [-d]
[rild.libpath]: [/vendor/lib64/libquectel-ril.so]
[ro.boot.noril]: [true]
[ro.boottime.ril-daemon]: [3885591876]
[ro.ril.ecclist]: [112,911]
rk3399_roc_pc_plus:/ $ su
:/ # dmesg | grep cdc
[ 0.872489] usbcore: registered new interface driver cdc_ether
[ 0.872517] usbcore: registered new interface driver cdc_eem
[ 0.872777] usbcore: registered new interface driver cdc_subset
[ 0.873021] usbcore: registered new interface driver cdc_ncm
[ 0.873080] usbcore: registered new interface driver cdc_mbim
[ 0.950998] usbcore: registered new interface driver cdc_acm
[ 0.951015] cdc_acm: USB Abstract Control Model driver for USB modems and ISDN adapters
[ 0.951107] usbcore: registered new interface driver cdc_wdm
[ 1704.445966] cdc_ether 5-1:1.0 eth1: register 'cdc_ether' at usb-fe380000.usb-1, CDC Ethernet De
vice, f0:4b:b3:b9:eb:e5
:/ #

```

3. 检查安卓是否设置了 APN。APN 可以通过安卓 UI 界面设置，一般流程如下：

Settings -> Network & Internet -> More -> Mobile network -> Advanced -> Access Point Names (详见 [5.7](#) 节)

4. 然后重启设备，如果还是无法上网，就通过抓取 RIL log，确认模块注册网络、信号强度、端口、APN 和 DNS 等情况。

5.3 如何获取 ECM 的 IP、DNS 等信息？

ECM 拨号成功后，local-ip, primary-dns, second-dns 等信息会被写入到 Android OS 属性项 net.xx.local-ip, net.xx.dns1, net.xx.dns2 中（“xx”是 ECM 使用的网卡名称，每个设备的网卡名称不同，如 eth1、usb0 等）。

以网卡名“eth1”为例，这几个属性可通过如下操作获取：

1. 执行 Android OS 命令：

```
getprop net.eth1 //获取 eth1 网卡的所有信息
```

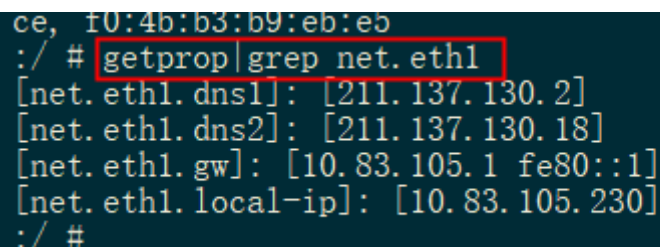
2. 如下属性分别对应 IP、网关、PDNS 和 SDNS 等信息：

```
net.eth1.local-ip //IP 地址
```



```
net.eth1.gw          //网关(gateway)
net.eth1.dns1        //pdns
net.eth1.dns2        //sdns
```

示例图如下：



```
ce, f0:4b:b3:b9:eb:e5
:/ # getprop|grep net.eth1
[net.eth1.dns1]: [211.137.130.2]
[net.eth1.dns2]: [211.137.130.18]
[net.eth1.gw]: [10.83.105.1 fe80::1]
[net.eth1.local-ip]: [10.83.105.230]
:/ #
```

如果 ECM 拨号使用的是其它网卡(如 usb0、eth2 等)，将上述属性中的“eth1”改成其它网卡名称即可。例如 net.usb0.local-ip、net.usb0.dns1 等。

5.4 如何使用 PPP 拨号方式？

5.4.1 切换 PPP 拨号方式

可以按照如下步骤切换 PPP 拨号：

1. 拨号前，请确保已参考第 3 章成功集成 RIL 库。
2. 在/system/build.prop 文件中设置如下属性值

```
ril.fibocom.dialmode=0
ril.fibocom.usbmode=17
```

，具体方法请参考 6.1 节。

3. 重启安卓设备。

5.4.2 PPP 拨号没有成功如何处理？

可以按照如下步骤排查：

1. 使用 getprop|grep ril 命令，检查 dialmode 和 usbmode 是否符合预期（dialmode 预期是 0，usbmode 预期是 17）。

```
D:\wu\cmdr $ ashell
rk3399_firefly:/ $ getprop|grep ril
[init.svc.ril-daemon]: [running]
[persist.ril.current_usbmode]: [10]
[ril.currentapntype]: [default]
[ril.datachannel]: [/dev/ttyUSB0]
[ril.fibocom.NetifName]: [eth1]
[ril.fibocom.dialmode]: [0]
[ril.fibocom.usbmode]: [17]
[ril.fibocom.version]: [Fibocom_RIL_V7X.06.V1.0.6_001]
[ril.function.dataonly]: [1]
[rild.libargs]: [-d /dev/ttyACM0]
[rild.libpath]: [/system/lib64/libreference-ril.so]
[ro.boot.noril]: [true]
[ro.ril.ecclist]: [112,911]
rk3399_firefly:/ $ |
```

2. 如果 dialmode 和 usbmode 正确，可以尝试修改/system/etc/ppp/目录下的 ip-up,ip-down 文件。

ip-up 文件修改如下：

```
#!/system/bin/sh
/system/bin/setprop "net.interfaces.defaultroute" "ppp0"
/system/bin/setprop "net.ppp0.dns1" "$DNS1"
/system/bin/setprop "net.ppp0.dns2" "$DNS2"
/system/bin/setprop "net.ppp0.local-ip" "$IPLOCAL"
/system/bin/setprop "net.ppp0.remote-ip" "$IPREMOTE"
exit 0
```

ip-down 文件修改如下：

```
#!/system/bin/sh
case $1 in
ppp1)
echo 0 > /proc/sys/net/ipv4/ip_forward;
esac
rm /etc/ppp/ppp*.pid
# Use interface name if linkname is not available
NAME=${LINKNAME:-"$1"}
/system/bin/setprop "net.$NAME.local-ip" ""
```

```
/system/bin/setprop "net.$NAME.remote-ip" ""  
/system/bin/setprop "net.interfaces.defaultroute" ""  
/system/bin/setprop "net.ppp0.dns1" ""  
/system/bin/setprop "net.ppp0.dns2" ""  
/system/bin/setprop "net.ppp0.local-ip" ""  
/system/bin/setprop "net.ppp0.remote-ip" ""
```



注意：ip-up,ip-down 文件需要具有执行权限 555。

3. 检查安卓是否设置了 APN。APN 可以通过安卓 UI 界面设置，一般流程如下：

Settings -> Network& Internet -> More -> Mobile network -> Advanced -> Access Point Names（详见 [5.7](#) 节）

4. 然后重启设备，如果还是无法拨号，请抓取 RIL LOG 并联系我司相关人员。

5.4.3 如何获取 PPP 拨号后的 IP、DNS 等信息？

ppp 拨号成功后，IP 地址，primary dns，second dns 被写入到属性 net.ppp0.local-ip, net.ppp0.dns1, net.ppp0.dns2 中，执行如下操作可获取：

1. 执行 Android OS 命令：

```
getprop net.ppp0
```

2. 如下属性分别对应 IP、网关、PDNS 和 SDNS 等信息：

net.ppp0.local-ip	//IP 地址
net.ppp0.gw	//网关(gateway)
net.ppp0.dns1	//pdns
net.ppp0.dns2	//sdns

示例图如下：

```
[ro.ril.ecclist]: [112,911]  
rk3399_roc_pc_plus:/ $ getprop|grep net.ppp0  
[net.ppp0.dns1]: [211.137.130.2]  
[net.ppp0.dns2]: [211.137.130.18]  
[net.ppp0.gw]: [192.168.0.1]  
[net.ppp0.local-ip]: [10.35.46.214]  
[net.ppp0.mask]: []  
[net.ppp0.remote-ip]: [192.168.0.1]  
rk3399_roc_pc_plus:/ $ |
```

5.4.4 设备不支持 PPP 拨号的解决方法

使用如下步骤来编译、安装 PPP：

1. 加入内核支持

在<android_dir>/kernel 目录下，输入 “make menuconfig” 命令开始编译内核，然后选择如下组件：

Device Drivers --- >

Network device support --- >

<*> PPP(point-to-point protocol) support

[*] PPP multilink support (EXPERIMENTAL)

<*> PPP support for async serial ports

<*> PPP support for sync tty ports

<*> PPP Deflate compression

<*> PPP BSD-Compress compression

具体步骤如下图所示：

```

Boot options --->
Userspace binary formats --->
Power management options --->
CPU Power Management --->
[*] Networking support --->
[*] Device Drivers --->
    Firmware Drivers --->
[ ] ACPI (Advanced Configuration and Power Interface) Support ----
File systems --->
[ ] Virtualization ----
    Kernel hacking --->
    Security options --->
-*. Cryptographic API --->
    Library routines --->

```

```

-*. Device Tree and Open Firmware support --->
< > Parallel port support ----
[*] Block devices --->
<*> NVM Express block device
    Misc devices --->
        SCSI device support --->
<*> Serial ATA and Parallel ATA drivers (libata) --->
[*] Multiple devices driver support (RAID and LVM) --->
< > Generic Target Core Mod (TCM) and ConfigFS Infrastructure ----
[ ] Fusion MPT device support ----
    IEEE 1394 (FireWire) support --->
[*] Network device support --->
[ ] Open-Channel SSD target support ----
    Input device support --->
        Character devices --->
        I2C support --->
[*] SPI support --->
< > SPMI support ----
< > HSI support ----
    PPS support --->
    PTP clock support --->

```

```

[ ] HIPPI driver support
-*. PHY Device support and infrastructure --->
< > Micrel KS8995MA 5-ports 10/100 managed Ethernet switch
<*> PPP (point-to-point protocol) support
<*> PPP BSD-Compress compression
<*> PPP Deflate compression
[*] PPP filtering
<*> PPP MPPE compression (encryption)
[*] PPP multilink support
<*> PPP over Ethernet
<*> PPP over L2TP
<*> PPP on L2TP Access Concentrator
<*> PPP on PPTP Network Server
<*> PPP support for async serial ports
<*> PPP support for sync tty ports
<*> SLIP (serial line) support
[*] CSLIP compressed headers
[ ] Keepalive and linefill
[*] Six bit SLIP encapsulation
<*> USB Network Adapters --->
[*] Wireless LAN --->

```



1. 按“ENTER”键进入对应目录，按“y”选择对应选项，按“n”取消选择对应选项。
2. “PPP(point-to-point) support”下之前所提的选项必选，此例中为全部选上，然后编译进内核。

完成这些步骤后，进行内核的编译。将生成的内核映像文件下载到板子上。内核启动后，会在 /dev 目录下生成 ppp 设备节点。如：

```
rk3399pro:/dev # ls ppp -l
crw-rw---- 1 radio vpn 108,  0 2020-11-29 13:23 ppp
rk3399pro:/dev #
```

2. 下载 PPP

使用如下命令得到 PPP 相关文件：

```
wget -c https://ftp.samba.org/pub/ppp/ppp-2.4.5.tar.gz
```

然后使用解压命令解压获取到的文件，生成目录 ppp-2.4.5：

```
tar -zxvf ppp-2.4.5.tar.gz
```

进入目录 ppp-2.4.5 后,执行以下命令进行交叉编译：

```
# ./configure
# make CC=arm-linux-gcc
```



若执行“make CC=arm-linux-gcc”时提示 “arm-linux-gcc：命令未找到” 错误，可能是因为 Linux 上没有安装 arm-linux-gcc 工具或该工具配置错误，客户可上网寻找解决办法或询问我司相关人员。

将 ppp-2.4.5 目录下 pppd , chat , pppdump , pppstats 下生成的可执行程序 pppd , chat, pppdump, pppstats 拷贝到开发板/usr/sbin 目录下。至此，发起 PPP 拨号前的准备工作完成。

5.5 电话打不通

如果电话打不通，确认 SIM 卡是否支持语音功能，如果支持需要向 Fibocom 技术支持人员确认模块在当前网络制式下是否支持电话功能。

5.6 无法发送短信

如果无法发送短信，确认 SIM 卡是否支持短信功能，如果支持需要向 Fibocom 技术支持人员确认模块在当前网络制式下是否支持短信功能。

5.7 如何设置 APN?

在 APN 为空的情况下，需要设置 APN 才能拨号，有两种方式。

5.7.1 手动添加 APN

请执行如下操作(仅适用于调试):

- 将模块安装在 Android 开发板上
- 在开发板上插入 SIM 卡
- 打开电源，等待开机。
- 设置拨号用到的 APN：
 - 设置->无线和网线（更多）->移动网络->接入点名称（APN）
 - 按菜单键选择“新建 APN”，输入 Name，APN，MCC，MNC，按菜单键保存。

如果运营商有提供特殊 APN 需要鉴权用户名、密码时，请使用运营商提供的 APN、鉴权用户名、鉴权密码。

- 选用设置的 APN 拨号，可以使用以下命令确认 RIL 是否拨号成功：

1. 检查网卡是否有 IP。执行 Android OS 命令：

```
ifconfig
```

2. 查看拨号成功后的 local ip，执行 Android OS 命令：

```
getprop | grep local-ip
```

5.7.2 配置 apns-conf.xml 文件

Android 设备路径：/system/etc/apns-conf.xml

修改 apns-conf.xml 文件时，根据 SIM 卡的实际信息进行添加，添加的主要信息有：

apn,mcc,mnc,user,password,type,protocol 等。

以下是列举的某个 APN 的信息添加：

```
<apn carrier="China Mobile"
    apn="cmnet"
    mcc="460"
    mnc="04"
    user=""
    server=""
    password=""
    proxy=""
    port=""
    mmsproxy=""
    mmsport=""
    mmsc=""
    type="default,net,supl"
    preferred="true"
    localized_name="APN_NAME_CMNET"
    protocol="IPV4V6"
    roaming_protocol="IPV4V6"
/>
```

更新 APN 配置列表后，由于 Android 设备启动很多次，telephony.db 已经初始化完成，因此需要先删除数据库文件，然后再次启动，才会加载新的配置文件。使用如下命令删除数据库：

```
rm /data/user_de/0/com.android.providers.telephony/databases/telephony.db
```


6 自定义系统属性

RIL 支持用户修改自定义系统属性，客户可以通过修改这些属性的值来自定义某些配置，修改这些属性的方法和支持的属性如下。

6.1 属性设置方法

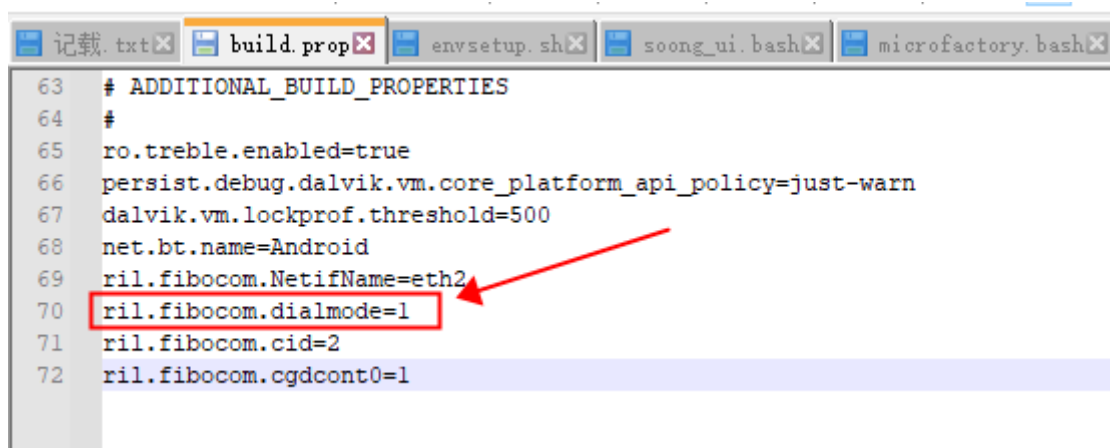
在/system/build.prop 文件里增加属性和它们的值，就可以永久地保存这些属性，在设备重启后依然保留。具体步骤如下：

1. 使用 adb pull 命令将/system/build.prop 文件从设备中 pull 出来：

adb root	❶ 需要进入 root 模式
adb remount	❷ 将 system 分区重新挂载成可读写模式
adb pull /system/build.prop <LOCAL_PATH>	❸ 将 build.prop 文件 pull 出来

```
Administrator@GHT-20200103NIB D:\wu_Desktop\Tools\cmdr
> adb root
Administrator@GHT-20200103NIB D:\wu_Desktop\Tools\cmdr
> adb remount
remount succeeded
Administrator@GHT-20200103NIB D:\wu_Desktop\Tools\cmdr
> adb pull /system/build.prop d:/log/build.prop
/system/build.prop: 1 file pulled. 0.6 MB/s (2557 bytes in 0.004s)
Administrator@GHT-20200103NIB D:\wu_Desktop\Tools\cmdr
>
```

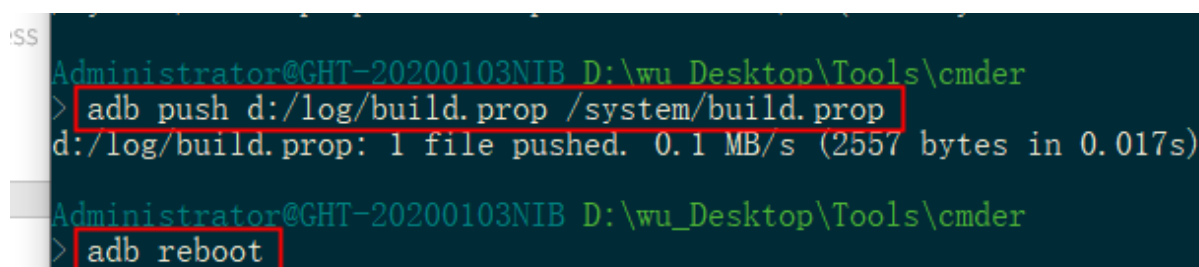
2. 在 build.prop 文件末尾新增属性(以"ril.fibocom.dialmode"为例)，然后保存修改：



```
63 # ADDITIONAL_BUILD_PROPERTIES
64 #
65 ro.treble.enabled=true
66 persist.debug.dalvik.vm.core_platform_api_policy=just-warn
67 dalvik.vm.lockprof.threshold=500
68 net.bt.name=Android
69 ril.fibocom.NetifName=eth2
70 ril.fibocom.dialmode=1
71 ril.fibocom.cid=2
72 ril.fibocom.cgdcont0=1
```

3. 将 build.prop 文件 push 进设备里的原位置，并重启设备：

```
adb push d:/log/build.prop /system/build.prop    ❶ 将 build.prop 文件重新 push 进设备里
adb reboot                                       ❷ 重启设备
```



```
Administrator@GHT-20200103NIB D:\wu\Desktop\Tools\cmdr
> adb push d:/log/build.prop /system/build.prop
d:/log/build.prop: 1 file pushed. 0.1 MB/s (2557 bytes in 0.017s)
Administrator@GHT-20200103NIB D:\wu\Desktop\Tools\cmdr
> adb reboot
```

4. 使用 adb shell 命令进入安卓设备，然后用 getprop|grep ril 命令即可看到我们设置的属性。如下：

```
adb shell    ❶ 进入安卓设备
getprop | grep ril  ❷ 查找与“ril”相关的属性
```

```
Administrator@GHT-20200103NIB D:\wu_Desktop\Tools\cmdex
> adb shell
rk3568_firefly_aioj:/ $ getprop|grep ril
[gsm.version.ril-impl]: []
[init.svc.ril-daemon]: [running]
[init.svc_debug_pid.ril-daemon]: [383]
[ril.currentapntype]: [default]
[ril.datachannel]: [/dev/ttyUSB6]
[ril.fibocom.NetifName]: [eth2]
[ril.fibocom.cgdcont0]: [1]
[ril.fibocom.cid]: [2]
[ril.fibocom.dialmode]: [1]
[ril.fibocom.version]: [L61x&MC61x_RIL_V11X.06.V1.0.3_RESET]
[ril.function.dataonly]: [1]
[rild.libargs]: [-d]
[rild.libpath]: [/vendor/lib64/libreference-ril.so]
[ro.boot.noril]: [false]
[ro.boottime.ril-daemon]: [6395227170]
[ro.ril.ecclist]: [112,911]
rk3568_firefly_aioj:/ $
```

我们自己设置的属性

6.2 ril.fibocom.dialmode

6.2.1 简介

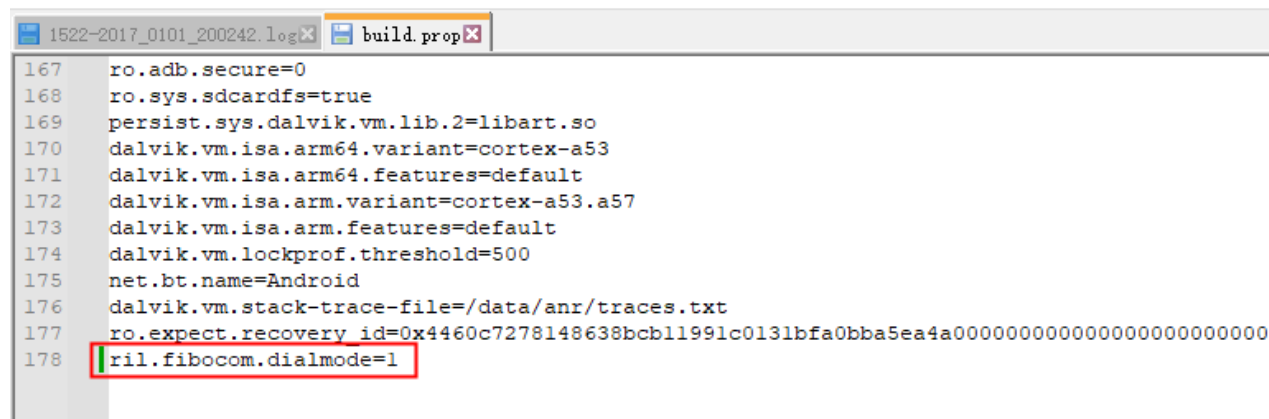
该属性用于指定 RIL 使用的拨号方式是 ECM、PPP 还是 RNDIS 拨号。

6.2.2 取值范围

表 4. dialmode 取值范围

ril.fibocom.dialmode	拨号方式
0	PPP
1(默认)	ECM
2	RNDIS

6.2.3 示例



```
167 ro.adb.secure=0
168 ro.sys.sdcardfs=true
169 persist.sys.dalvik.vm.lib.2=libart.so
170 dalvik.vm.isa.arm64.variant=cortex-a53
171 dalvik.vm.isa.arm64.features=default
172 dalvik.vm.isa.arm.variant=cortex-a53.a57
173 dalvik.vm.isa.arm.features=default
174 dalvik.vm.lockprof.threshold=500
175 net.bt.name=Android
176 dalvik.vm.stack-trace-file=/data/anr/traces.txt
177 ro.expect.recovery_id=0x4460c7278148638bcb11991c0131bfa0bba5ea4a00000000000000000000000000000000
178 ril.fibocom.dialmode=1
```

6.3 ril.fibocom.NetifName

6.3.1 作用

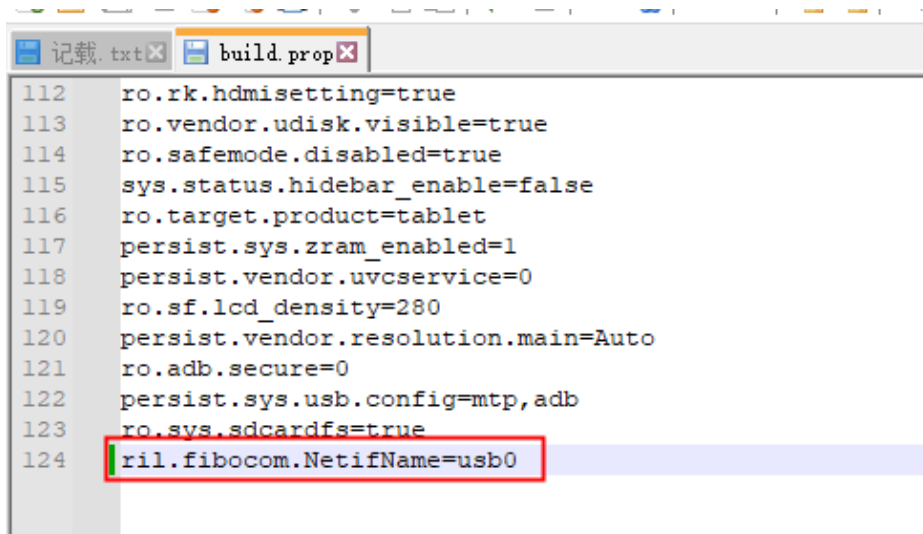
自定义 RIL 拨号使用的网卡(常用于 ECM 拨号)。

RIL 发起 ECM 拨号时, 选择网卡的顺序是: NetifName 指定->RIL 预设(如 eth1)->usb0。只有前两种方法都没有找到网卡时, 才会使用默认网卡“usb0”。

6.3.2 取值范围

常用的取值为 usb0、eth0 和 eth1 等, 默认值是 usb0。

6.3.3 示例



```
112 ro.rk.hdmisetting=true
113 ro.vendor.udisk.visible=true
114 ro.safemode.disabled=true
115 sys.status.hidebar_enable=false
116 ro.target.product=tablet
117 persist.sys.zram_enabled=1
118 persist.vendor.uvcservice=0
119 ro.sf.lcd_density=280
120 persist.vendor.resolution.main=Auto
121 ro.adb.secure=0
122 persist.sys.usb.config=mtp,adb
123 ro.sys.sdcardfs=true
124 ril.fibocom.NetifName=usb0
```

6.4 ril.fibocom.cid

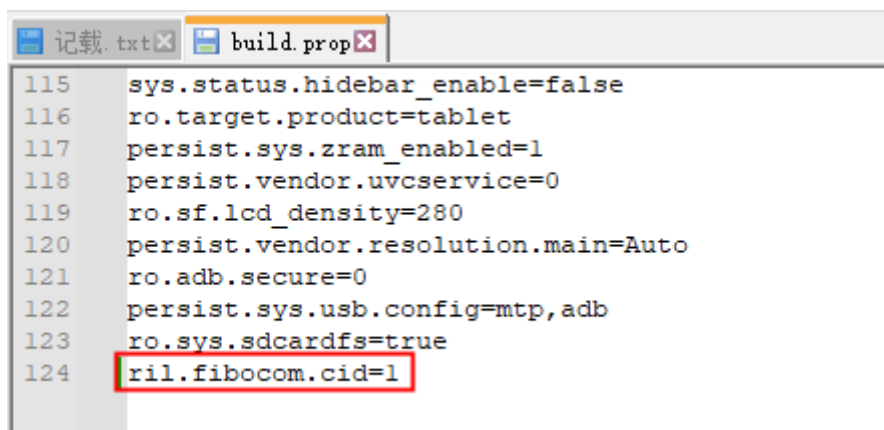
6.4.1 作用

自定义模块使用哪一路的 cid 进行拨号。

6.4.2 取值范围

该属性的取值范围是[1,7]，默认值是 1。

6.4.3 示例



```
115 sys.status.hidebar_enable=false
116 ro.target.product=tablet
117 persist.sys.zram_enabled=1
118 persist.vendor.uvcservice=0
119 ro.sf.lcd_density=280
120 persist.vendor.resolution.main=Auto
121 ro.adb.secure=0
122 persist.sys.usb.config=mtp,adb
123 ro.sys.sdcardfs=true
124 ril.fibocom.cid=1
```

6.5 ril.fibocom.cgdcont0

6.5.1 作用

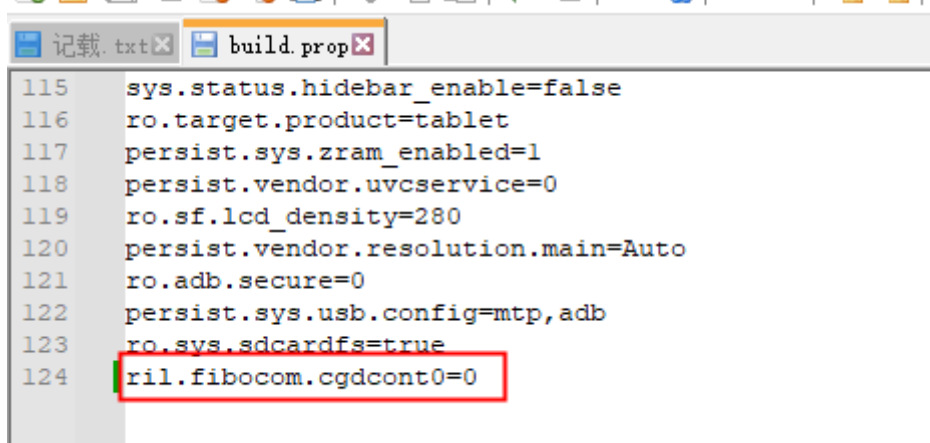
该属性用于自定义是否设置 cid0 的 APN。

6.5.2 取值范围

0: 不设置 cid0 的 APN(默认)

1: 设置 cid0 的 APN

6.5.3 示例



```
115 sys.status.hidebar_enable=false
116 ro.target.product=tablet
117 persist.sys.zram_enabled=1
118 persist.vendor.uvcservice=0
119 ro.sf.lcd_density=280
120 persist.vendor.resolution.main=Auto
121 ro.adb.secure=0
122 persist.sys.usb.config=mtp,adb
123 ro.sys.sdcardfs=true
124 ril.fibocom.cgdcont0=0
```

6.6 ril.fibocom.usbmode

6.6.1 作用

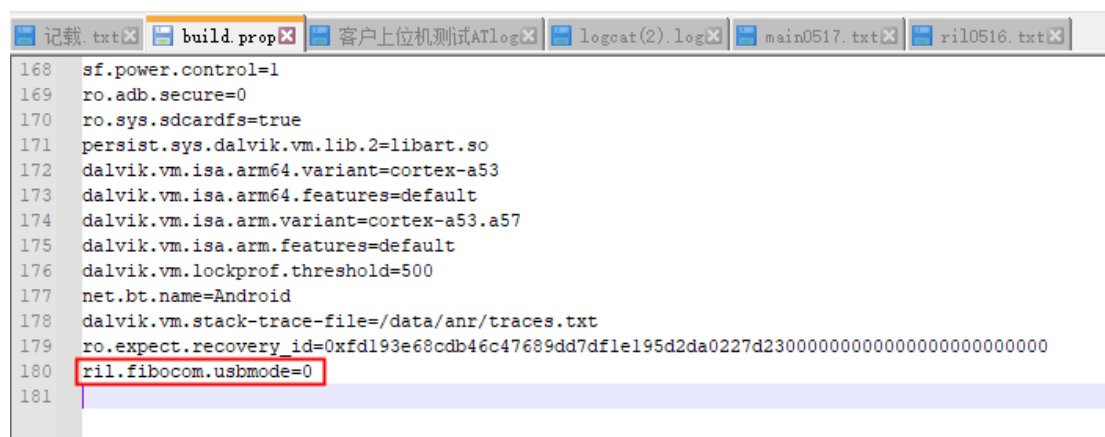
该属性用于修改模块的 USBMODE 为指定值。

6.6.2 取值范围

该属性的取值范围是 0 和[17-72]；不设置该属性或该属性值为 0 时，表示不修改模块的 USBMODE，此时不会有任何效果产生。

该属性的默认值是 0。

6.6.3 示例



```
168 sf.power.control=1
169 ro.adb.secure=0
170 ro.sys.sdcardfs=true
171 persist.sys.dalvik.vm.lib.2=libart.so
172 dalvik.vm.isa.arm64.variant=cortex-a53
173 dalvik.vm.isa.arm64.features=default
174 dalvik.vm.isa.arm.variant=cortex-a53.a57
175 dalvik.vm.isa.arm.features=default
176 dalvik.vm.lockprof.threshold=500
177 net.bt.name=Android
178 dalvik.vm.stack-trace-file=/data/anr/traces.txt
179 ro.expect.recovery_id=0x9fd193e68cdb46c47689dd7df1e195d2da0227d2300000000000000000000000000000000
180 ril.fibocom.usbmode=0
181
```

7 附录 A 参考文件

表 5. 术语和缩写

缩写词	描述
GSM	全球移动通信系统
VID	供应商 ID
PID	产品识别码
RIL	无线接口层
RILD	无线接口层守护进程
APN	接入点名称