

The logo for Fibocom, featuring the company name in blue and Chinese characters in black. The background consists of a world map composed of small grey dots.

Fibocom 广和通

完 美 无 线 体 验

MTC

RIL 适配指南_Android

V1.9

免责声明

您使用本文档过程中所采取的任何行为均由您自行承担风险，本公司在任何情况下均不对任何性质的损害或损失负责。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。

本文档可能包含第三方信息、产品、服务、数据或内容（统称“第三方内容”）。本公司不控制且不对第三方内容承担任何责任，包括但不限于准确性、兼容性、可靠性、可用性、合法性、适当性、性能、不侵权、更新状态等，除非本文档另有明确说明。在本文档中提及或引用任何第三方内容不代表本公司对第三方内容的认可或保证。用户若需要第三方许可，须通过合法途径获取第三方许可，除非本文档另有明确说明。

版权声明

版权所有 ©2026 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。

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

目录

适用型号	3
修订记录	4
1 引言	5
2 RIL 驱动概述	6
2.1 支持的功能	6
2.2 支持的 Android 版本	6
3 RIL 集成配置	8
3.1 启动 ril-daemon	8
3.1.1 Android8 以前版本	8
3.1.2 Android8 及以后版本	9
3.1.3 参数说明	9
3.2 修改 RILD	9
3.3 修改 RILC	11
3.3.1 更换 libril.so 库	11
3.3.1.1 如何查看 libril.so 库的版本?	11
3.3.2 修改 RIL_VERSION.....	12
3.3.2.1 如何查看当前 RIL_VERSION 的值.....	13
3.3.3 manifest.xml	13
3.3.3.1 radio1.1.....	13
3.3.3.2 radio1.4.....	14
3.3.3.3 radio1.6.....	14
3.3.3.4 如何查看 manifest 的版本.....	17
3.3.4 确认三个 radio 版本相同	18
3.4 修改 config_mobile_data_capable 值.....	19
3.5 集成拨号驱动	19
3.5.1 集成 QMI_WWAN 驱动	19
3.6 加载 RIL 库文件.....	20
3.7 选择数据拨号方式	23
4 常见问题	24
4.1 如何抓取 RIL 日志.....	24
4.2 无法上网的排查方法	24

4.2.1 检查基本信息	24
4.2.2 检查网卡名是否正确(ECM)	25
4.2.3 检查路由是否正确(ECM)	25
4.2.3.1 问题现象	25
4.2.3.2 解决方法	27
4.2.4 检查 ip-up 和 ip-down 文件(PPP)	27
4.2.4.1 ip-up 文件	27
4.2.4.2 ip-down 文件	27
4.2.4.3 修改权限	28
4.2.5 抓 LOG 给广和通分析	28
4.3 无法拨通电话	28
4.4 无法发送短信	28
4.5 如何设置 APN?	28
4.5.1 UI 界面设置 APN	29
4.5.2 修改 apns-conf.xml 文件	29
4.6 如何修改 ECM 拨号使用的网卡名称?	30
4.7 ght_ril_config.ini 文件	30
4.7.1 使用说明	30
4.7.2 空三角时请打开 SIGNAL_POLL 选项	31
4.7.2.1 问题现象	31
4.7.2.2 解决方法	32
4.7.3 NL668 使用切卡功能时请打开 CGLA_SLEEP 选项	33
4.8 RIL LOG 中显示 “Permission denied” 解决方法	33
4.8.1 问题现象	33
4.8.2 解决方法	34
4.9 如何设置自定义属性	35
4.9.1 属性设置方法	35
4.9.2 ril.fibocom.dialmode	36
4.9.3 ril.fibocom.NetifName	37
4.9.4 ril.fibocom.usbmode	37
附录 A 拨号方式和 USBMODE	39
附录 B 常用术语和缩写	40
附录 C RIL Log 中 TAG 标签说明	41

适用型号

序号	适用型号	说明
1	L61x&LC61x&LG61x&MC61x&MG61x 系列	L610、LC610N、LG610、MC610、MC615、MC617、MC619
2	MC11x&MG110 系列	MC116、MG110
3	NL668&NL668T 系列	NL668、NL668T
4	L71x 系列	L716 和 L718
5	MC919 系列	MC919
6	MC66x&MG66x 系列	MC660、MC661、MC665、MC667、MC669、MG661、MG662
7	FG132 系列	FG132
8	MA510 系列	MA510

修订记录

V1.9 (2026-03-11)	1. 新增 CGLA_SLEEP 配置章节，详见 4.7.3 章节
V1.8 (2024-09-11)	1. 新增 MA510 适用型号 2. 新增关于 radio1.4 和 radio1.6 的内容，详见 3.3 章节
V1.7 (2024-06-11)	1. 增加 FG132 适用型号 2. 增加 QMI_WWAN 拨号
V1.6 (2024-01-16)	1. 修改格式错误 2. 删除 LEx 系列适用型号
V1.5 (2023-09-26)	1. 优化“适用型号”的内容 2. 优化部分截图
V1.4 (2023-08-23)	1. 增加 MG661 适配型号 2. 增加安卓 12 的适配
V1.3 (2023-06-07)	1. 添加 MC619 适用型号 2. 增加对 ght_ril_config.ini 配置文件的说明
V1.2 (2023-05-10)	添加 LE1x 适用型号
V1.1 (2023-04-26)	添加 MG110 适用型号
V1.0 (2023-04-15)	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的目标设备，以及如何修改启动 RIL 服务的配置文件。

2 RIL 驱动概述

2.1 支持的功能

表 1.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_Android_RIL 支持以下 Android 版本，见下表：

表 2. 支持的 Android 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
Android 12.x	YES
Android 13.x	YES
Android 14.x	YES



建议在安卓 11 或更高的版本上使用 5G 模组，否则 5G 信号值，图标等信息可能显示异常。

3 RIL 集成配置

本章介绍 RIL 库集成到客户 Android 系统的操作，包括配置启动脚本，修改 Android 系统 RILD 执行权限和加载 RIL 库文件等。

3.1 启动 ril-daemon

ril-daemon 进程负责寻找并启动 RIL 库，因此，确保 ril-daemon 被启动，并且用“-l”参数指定 RIL 库的路径。具体启动方法如下。

3.1.1 Android8 以前版本

Android8（不含）以前的版本，需要修改 <Android_Dir>/device/xx/xx/init.rc 文件来启动 ril-daemon。

- 32 位设备

在 init.rc 文件中，增加红色字体的内容用于指定 RIL 库的路径为“/system/lib/libreference-ril.so”：

```
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
```

- 64 位设备

在 init.rc 文件中，增加红色字体的内容用于指定 RIL 库的路径为“/system/lib64/libreference-ril.so”：

```
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
```



不同设备的 init.rc 路径不同，例如/device/rockchip/rk3399/init.rk3399.rc 等。

3.1.2 Android8 及以后版本



Android8 及以后版本，RIL 库不能放在/system 目录下，否则 RIL 的上层 RILD 无法访问到它。

Android8 及以后的版本，需要修改 `<Android_Dir>/hardware/ril/rild/rild.rc` 文件来启动 ril-daemon。

- 32 位设备

在 rild.rc 文件中，增加红色字体的内容用于指定 RIL 库的路径为“/vendor/lib/libreference-ril.so”：

```
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
```

- 64 位设备

在 rild.rc 文件中，增加红色字体的内容用于指定 RIL 库的路径为“/vendor/lib64/libreference-ril.so”：

```
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
```

3.1.3 参数说明

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

- -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：QMI_WWAN 拨号。

3.2 修改 RILD

ril-daemon 是进程的名称，它实际运行的是可执行文件 RILD，因此我们对 RILD 也要做一些适配性修改。

1. 因为 RILD 程序执行过程中需要 root 权限，所以要在 rild.c 文件（`<$android dir>/hardware/ril/rild/rild.c`）的 main 函数中，去掉 switchuser()，如下图所示：



部分安卓版本中没有 switchUser 函数（如 Android10），可直接执行下一步骤。

```
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

2. 为了确保 RILD 参与编译，检查<Android_Dir>/device/xx/device.mk 文件中是否有如下内容，如果没有则加入，如下图所示：

```
PRODUCT_PACKAGES += rild
```

```
605 PRODUCT_PROPERTY_OVERRIDES += \
606     rild.libpath=/system/lib/libril-rk29-dataonly.so
607 endif
608 endif
609
610 ifeq ($(strip $(BOARD_HAS_RK_4G_MODEM)),true)
611 PRODUCT_PACKAGES += \
612     rild \
613     librk-ril \
614     dhcpcd
615
```

图 2 RILD 编译



不同设备的 device.mk 路径不同，例如/device/xx/xx/device.mk 等。

3.3 修改 RILC

从安卓 8 开始引入了 android.hardware.radio 服务，并且版本在不断更新，例如 radio1.1、radio1.4 和 radio1.6 等，下面是广和通建议选择的 radio 版本。

对 Fibocom_Android_RIL_1.0.0.12 及以前的版本，建议一律使用 **radio1.1**。

对 Fibocom_Android_RIL_1.0.0.13 及以后的 RIL 版本，建议使用的 radio 版本如下：

表 3. 不同安卓版本的建议 radio 版本

Android	radio 版本
4-7	NA(直接跳到 修改 config_mobile_data_capable 值 章节)
8-9	1.1
10-14	1.4

为了使用对应的 radio 版本，请做如下修改。

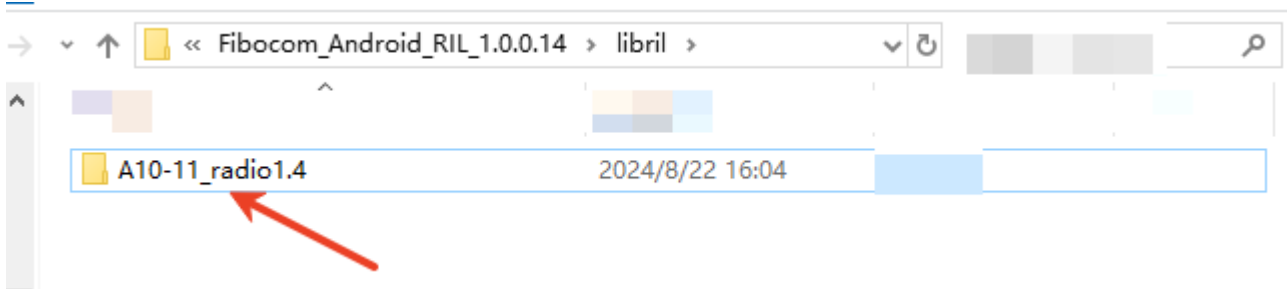


- 本章节剩余内容仅适用于 **Fibocom_Android_RIL_1.0.0.13 及以后**的 RIL 版本。
- 上表中的 Android 和 radio 版本非强制对应，例如安卓 11 也能使用 radio1.1，只要保证下文的 libril.so, RIL_VERSION 和 manifest.xml 三者对应即可。

3.3.1 更换 libril.so 库

广和通默认提供 radio1.4 对应的 libril.so 库，在版本文件夹中路径如下。

该 so 库放到安卓设备的/vendor/lib(64)/libril.so 路径下生效。



radio1.1 一般是安卓默认自带的版本，无需特殊提供；如需 radio1.6 请联系广和通技术支持获取。

3.3.1.1 如何查看 libril.so 库的版本？

对安卓 8 及以上的安卓版本，在源代码的<Android>/hardware/ril/libril/ril_service.cpp 文件或安卓 LOG 中搜索关键字“registerService: starting”，如果结果里有“**V1_1**”的关键字，说明是 **radio1.1** 的库；如果是“**V1_4**”，说明是 **radio1.4** 的库，以此类推：

```

10109     get_RILC_version();
10110
10111     radioService[i] = new RadioImpl_1_4;
10112     radioService[i]->mSlotId = i;
10113     RLOGD("registerService: starting android::hardware::radio::V1_4::IRadio %s for slot %d",
10114           serviceNames[i], i);
10115     android::status_t status = radioService[i]->registerAsService(serviceNames[i]);
10116
10117     RLOGD("registerService: OemHook is enabled = %s", kOemHookEnabled ? "true" : "false");
10118     if (kOemHookEnabled) {
10119         oemHookService[i] = new OemHookImpl;
10120         oemHookService[i]->mSlotId = i;
10121         status = oemHookService[i]->registerAsService(serviceNames[i]);
10122     }
10123
10124     ret = pthread_rwlock_unlock(radioServiceRwlockPtr);
~/Android/android11-rk3568/hardware/ril/libril/ril_service.cpp

```

```

all0730_ECM_A10_FG132_1.0.0.13_radio1.4_OK.txt
1  ----- beginning of kernel
07-30-16:51:33.545 I/ ( 1688): *****Enter get_RILC_version*****
行 7193: 07-30 16:51:33.545 D/RILC ( 1688): FIBOCOM RILC version: Fibocom_Android_RILC_1.4_V1.0.0.0
行 7194: 07-30 16:51:33.546 D/RILC ( 1688): registerService: starting android::hardware::radio::V1_4::IRadio slot1 for slot 0
行 7195: 07-30 16:51:33.546 D/RILC ( 1688): registerService: OemHook is enabled = true
行 7205: 07-30 16:51:33.549 D/RILC ( 1688): RILHIDL called registerService
行 7207: 07-30 16:51:33.552 I/RILC ( 1688): setResponseFunctions
行 7211: 07-30 16:51:33.553 D/RILC ( 1688): calling UNSOLICITED responseFunction for index 34
行 7216: 07-30 16:51:33.558 D/RILC ( 1688): failConnectedInd
行 7217: 07-30 16:51:33.558 D/RILC ( 1688): calling UNSOLICITED responseFunction for index 0
行 7218: 07-30 16:51:33.558 D/RILC ( 1688): radioStateChangedInd: radioState 1
行 7222: 07-30 16:51:33.558 I/RILC ( 1688): RIL Daemon version: Fibocom_Android_RIL_V10X.06.V1.0.0.12_001
行 7226: 07-30 16:51:33.559 I/RILC ( 1688): setRadioPower: serial 704312868 on 1
行 7230: 07-30 16:51:33.564 D/RILC

```

3.3.2 修改 RIL_VERSION

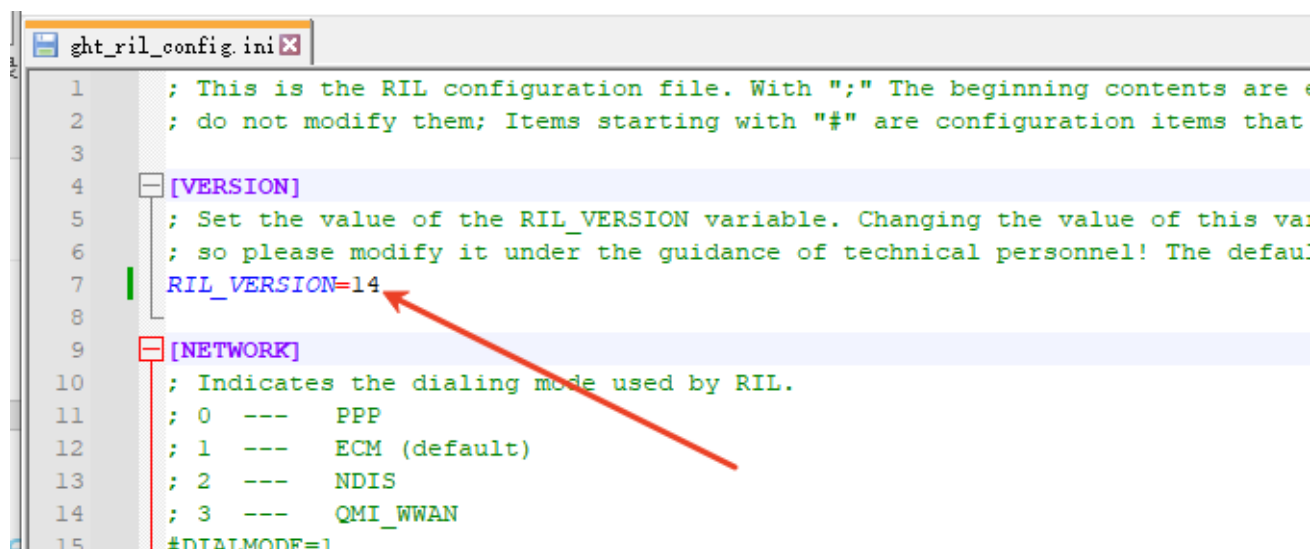
在 [gh_t ril_config.ini 文件](#) 中将 RIL_VERSION 改成下表对应的值。



不建议使用下表没有的 radio 版本。

表 4. 不同 radio 版本对应的 RIL_VERSION

radio 版本	RIL_VERSION
radio1.1	12
radio1.4	14
radio1.6	16



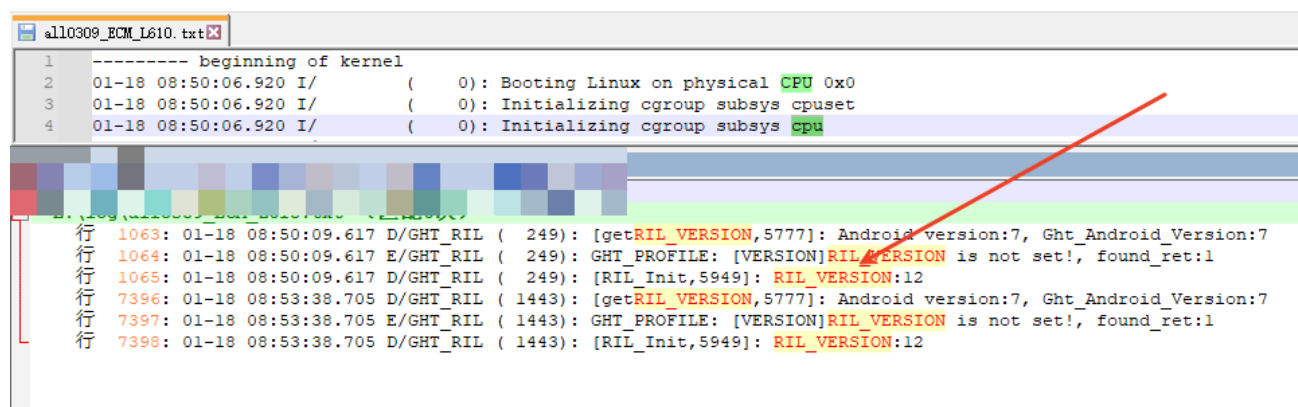
```

1 ; This is the RIL configuration file. With ";" The beginning contents are
2 ; do not modify them; Items starting with "#" are configuration items that
3
4 [VERSION]
5 ; Set the value of the RIL_VERSION variable. Changing the value of this var
6 ; so please modify it under the guidance of technical personnel! The default
7 RIL_VERSION=14
8
9 [NETWORK]
10 ; Indicates the dialing mode used by RIL.
11 ; 0 --- PPP
12 ; 1 --- ECM (default)
13 ; 2 --- NDIS
14 ; 3 --- QMI_WWAN
15 #DTATMODE=1

```

3.3.2.1 如何查看当前 RIL_VERSION 的值

抓取开机的 RIL LOG 后搜索“RIL_VERSION”，即可看到对应的值，然后根据“表 4 不同 radio 版本对应的 RIL_VERSION”即可找到对应的 radio 版本。



```

1 ----- beginning of kernel
2 01-18 08:50:06.920 I/ ( 0): Booting Linux on physical CPU 0x0
3 01-18 08:50:06.920 I/ ( 0): Initializing cgroup subsys cpuset
4 01-18 08:50:06.920 I/ ( 0): Initializing cgroup subsys cpu
...
行 1063: 01-18 08:50:09.617 D/GHT_RIL ( 249): [getRIL_VERSION,5777]: Android version:7, Ght_Android_Version:7
行 1064: 01-18 08:50:09.617 E/GHT_RIL ( 249): GHT_PROFILE: [VERSION]RIL_VERSION is not set!, found_ret:1
行 1065: 01-18 08:50:09.617 D/GHT_RIL ( 249): [RIL_Init,5949]: RIL_VERSION:12
行 7396: 01-18 08:53:38.705 D/GHT_RIL ( 1443): [getRIL_VERSION,5777]: Android version:7, Ght_Android_Version:7
行 7397: 01-18 08:53:38.705 E/GHT_RIL ( 1443): GHT_PROFILE: [VERSION]RIL_VERSION is not set!, found_ret:1
行 7398: 01-18 08:53:38.705 D/GHT_RIL ( 1443): [RIL_Init,5949]: RIL_VERSION:12

```

3.3.3 manifest.xml

对安卓 8 及以后的版本，需要在<Android>/device/xx/xx/manifest.xml 文件中加入 radio 服务相关内容，来保证安卓上层与 ril-daemon 能正常通信。



不同设备的 manifest.xml 路径不同，例如/device/rockchip/rk3399/manifest.xml 等。

需要添加的内容如下（仅作参考）。

3.3.3.1 radio1.1

```

<hal format="hidl">
  <name>android.hardware.radio</name>

```

```

    <transport>hwbinder</transport>
    <fqname>@1.1::IRadio/slot1</fqname>
    <fqname>@1.1::IRadio/slot2</fqname>
    <fqname>@1.2::ISap/slot1</fqname>
</hal>
<hal format="hidl">
    <name>android.hardware.radio.deprecated</name>
    <transport>hwbinder</transport>
    <version>1.0</version>
    <interface>
        <name>IOemHook</name>
        <instance>slot1</instance>
    </interface>
    <fqname>@1.0::IOemHook/slot1</fqname>
</hal>

```

3.3.3.2 radio1.4

```

<hal format="hidl">
    <name>android.hardware.radio</name>
    <transport>hwbinder</transport>
    <fqname>@1.4::IRadio/slot1</fqname>
    <fqname>@1.4::IRadio/slot2</fqname>
    <fqname>@1.2::ISap/slot1</fqname>
    <fqname>@1.2::ISap/slot2</fqname>
</hal>
<hal format="hidl">
    <name>android.hardware.radio.config</name>
    <transport>hwbinder</transport>
    <version>1.1</version>
    <interface>
        <name>IRadioConfig</name>
        <instance>default</instance>
    </interface>
</hal>

```

3.3.3.3 radio1.6

安卓 13 和 14 建议添加的内容略有区别，分别如下：

3.3.3.3.1 安卓 13

```
<hal format="aidl">
    <name>android.hardware.radio.config</name>
    <fqname>IRadioConfig/default</fqname>
</hal>
<hal format="aidl">
    <name>android.hardware.radio.data</name>
    <fqname>IRadioData/slot1</fqname>
</hal>
<hal format="aidl">
    <name>android.hardware.radio.messaging</name>
    <fqname>IRadioMessaging/slot1</fqname>
</hal>
<hal format="aidl">
    <name>android.hardware.radio.modem</name>
    <fqname>IRadioModem/slot1</fqname>
</hal>
<hal format="aidl">
    <name>android.hardware.radio.network</name>
    <fqname>IRadioNetwork/slot1</fqname>
</hal>
<hal format="aidl">
    <name>android.hardware.radio.sim</name>
    <fqname>IRadioSim/slot1</fqname>
</hal>
<hal format="aidl">
    <name>android.hardware.radio.voice</name>
    <fqname>IRadioVoice/slot1</fqname>
</hal>
```

3.3.3.3.2 安卓 14

```
<hal format="aidl">
    <name>android.hardware.radio.config</name>
    <version>2</version>
    <fqname>IRadioConfig/default</fqname>
</hal>
```

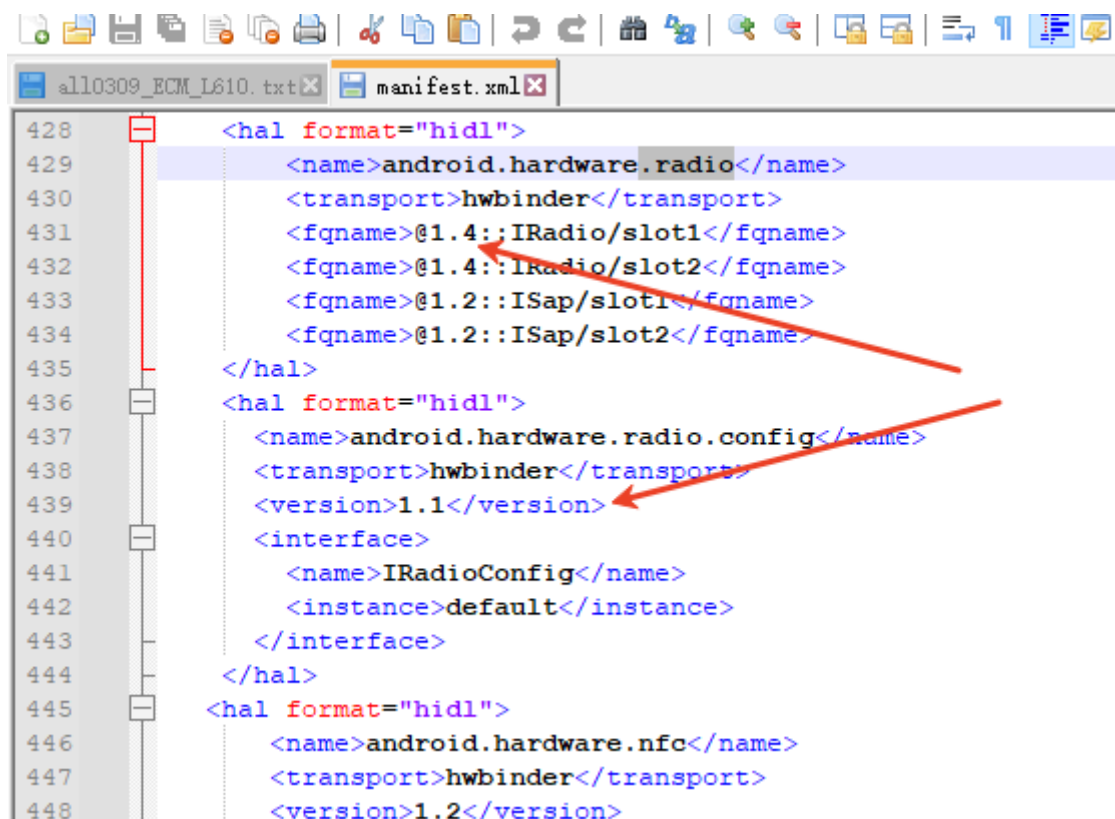
```
<hal format="aidl">
    <name>android.hardware.radio.data</name>
    <version>2</version>
    <fqname>IRadioData/slot1</fqname>
</hal>
<hal format="aidl">
    <name>android.hardware.radio.ims</name>
    <fqname>IRadioIms/slot1</fqname>
</hal>
<hal format="aidl">
    <name>android.hardware.radio.ims.media</name>
    <fqname>IImsMedia/default</fqname>
</hal>
<hal format="aidl">
    <name>android.hardware.radio.messaging</name>
    <version>2</version>
    <fqname>IRadioMessaging/slot1</fqname>
</hal>
<hal format="aidl">
    <name>android.hardware.radio.modem</name>
    <version>2</version>
    <fqname>IRadioModem/slot1</fqname>
</hal>
<hal format="aidl">
    <name>android.hardware.radio.network</name>
    <version>2</version>
    <fqname>IRadioNetwork/slot1</fqname>
</hal>
<hal format="aidl">
    <name>android.hardware.radio.sim</name>
    <version>2</version>
    <fqname>IRadioSim/slot1</fqname>
</hal>
<hal format="aidl">
    <name>android.hardware.radio.sap</name>
    <fqname>ISap/slot1</fqname>
</hal>
<hal format="aidl">
```

```
<name>android.hardware.radio.voice</name>
<version>2</version>
<fqname>IRadioVoice/slot1</fqname>
</hal>
```

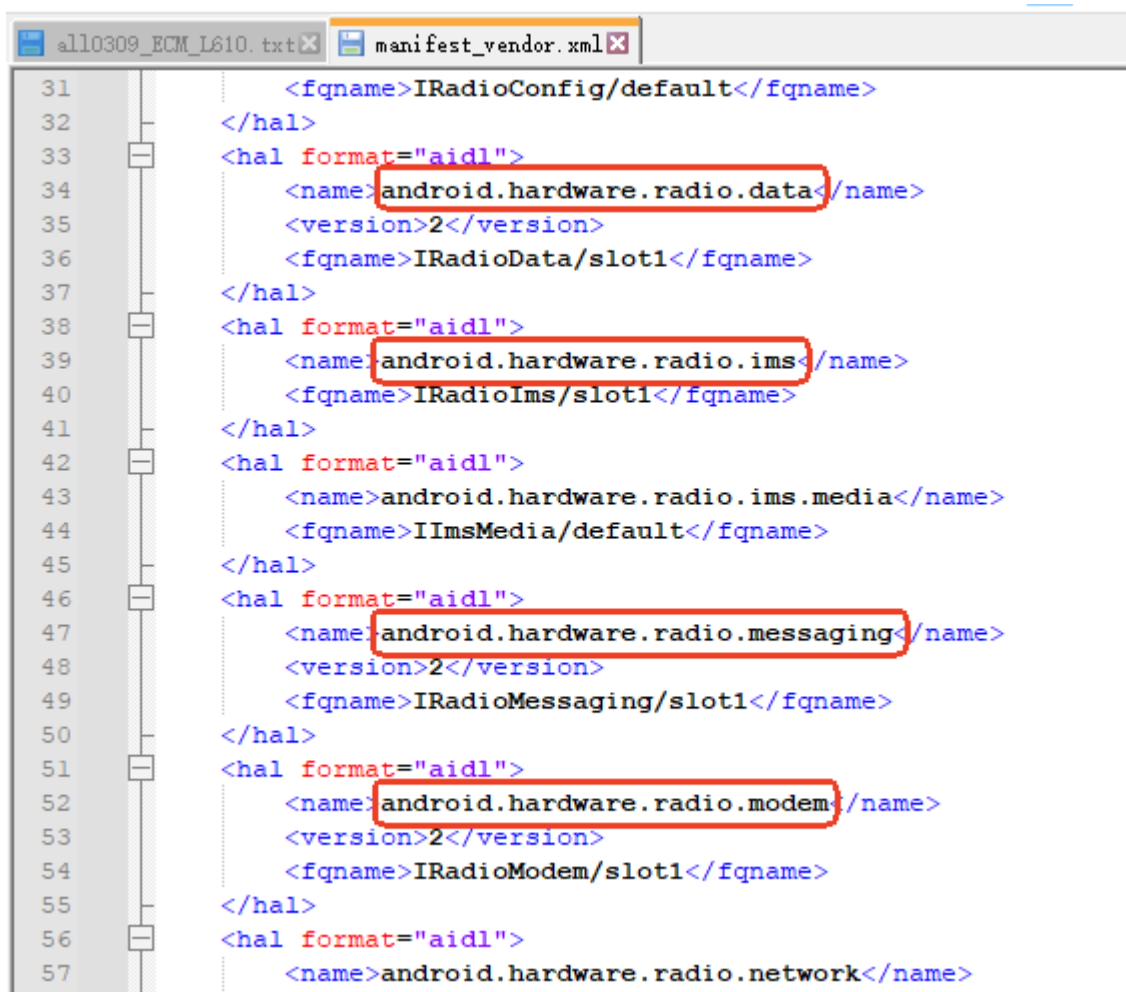
3.3.3.4 如何查看 manifest 的版本

在安卓设备中，可通过查看/vendor/etc/vintf/manifest.xml 文件查找相关服务。在该文件中搜索对应的服务名称，并查看 <fqname> 标签中标识的版本号，其中最高的版本数字即为该设备所支持的最大版本。例如下图的 android.hardware.radio，其最高支持版本为 radio1.4。

也可以直接查看<version>标签，例如下面的 android.hardware.radio.config 最高支持 1.1 版本。



radio1.6 版本的 android.hardware.radio 比较特殊，被分成了 android.hardware.radio.data、android.hardware.radio.ims 等。

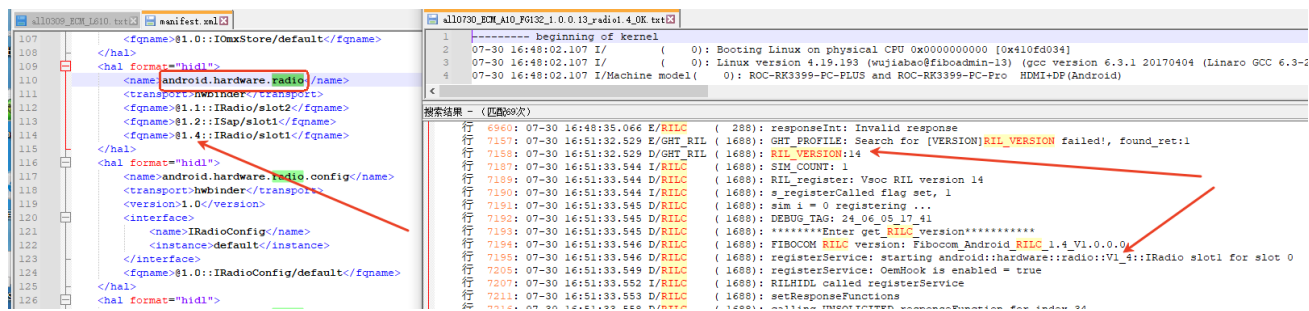


3.3.4 确认三个 radio 版本相同

按照上述步骤配置完 RILC 后，要确保 libril.so, RIL_VERSION 和 manifest.xml 三个对应的 radio 版本相同，否则 RILC 无法正常运行。

如果检查结果有异常，请返回前面对应章节修改。

下图是 radio1.4 的正常示例，可以看到三个值都是对应 radio1.4，符合预期。

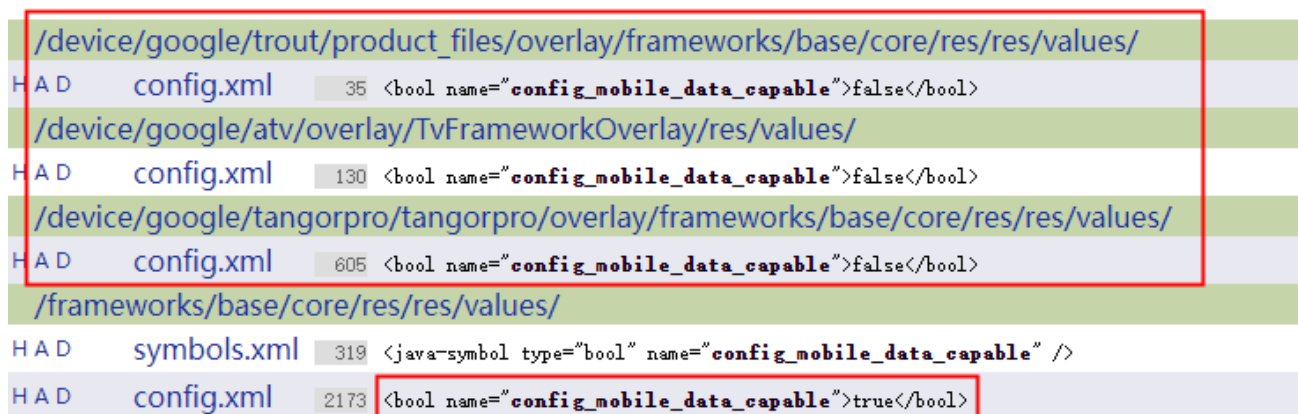


3.4 修改 config_mobile_data_capable 值

在 <Android>/frameworks/base/core/res/res/values/config.xml 或 <Android>/device/./overlay/frameworks/base/core/res/res/values/config.xml 文件中检查 config_mobile_data_capable 变量的值，如果是 false，就改成 true：



重点检查 device 目录下的 config_mobile_data_capable 值，因为该目录下的值优先级更高，会覆盖 framework 目录的值。



3.5 集成拨号驱动

3.5.1 集成 QMI_WWAN 驱动



- 非高通平台模组请跳过该章节。
- 对高通平台模组（如 FG132、NL668 或 NL668T 等），广和通优先推荐 ECM 拨号，本节内容非必需，可以跳过。

使用 QMI_WWAN 拨号前，需要在安卓源码中集成 QMI_WWAN 驱动。具体过程请参考《Fibocom_MTC_拨号指南_Linux》文档的“QMI_WWAN 拨号”章节。

3.6 加载 RIL 库文件

1. 完成前面的步骤后就可以编译安卓镜像了（编译方法各不相同，根据实际情况而定）。编译完后将镜像下载到安卓设备中。
2. 如果前面集成了 QMI_WWAN 驱动，需要通过如下步骤确认集成是否成功（未集成 QMI_WWAN 驱动请跳过该步骤）。
 - a. 参考[拨号方式和 USBMODE](#) 和模组型号找到 QMI_WWAN 对应的 USBMODE，然后将模组的 USBMODE 改成对应值。下图是在安卓设备上，将 FG132 模组的 USBMODE 改成 QMI_WWAN 所需的 36。

```
[10.111.ecc11st]: [112,911]
rk3568_firefly_aioj:/ # stop ril-daemon
rk3568_firefly_aioj:/ # cat /dev/ttyUSB1 &
[1] 2211
rk3568_firefly_aioj:/ # echo -en "AT+GTUSBMODE=36\r" > /dev/ttyUSB1
rk3568_firefly_aioj:/ #

OK

rk3568_firefly_aioj:/ # |
```

- b. 连接模组与安卓，如果 QMI_WWAN 驱动已集成成功，预期可以识别出如下的“wwan0”网卡。如果未识别出来，请仔细检查[集成 QMI_WWAN 驱动](#)过程，或联系广和通技术支持人员。

```
D: (wa)cmdet @ ashell
rk3568_firefly_aioj:/ # ifconfig wwan0
wwan0      Link encap:Ethernet  HWaddr 26:64:27:a8:d4:e2  Driver qmi_wwan_f
           NOARP  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:1000
           RX bytes:0 TX bytes:0
```

3. 然后用 `getprop|grep ril` 命令查询，`ril-daemon` 的状态应是“running”，如下图：

```
console:/system/lib # getprop | grep ril
[init.svc.vendor.ril-daemon]: [running]
[init.svc_debug_pid.vendor.ril-daemon]: [502]
[rild.libpath]: [/system/lib/libreference-ril.so]
[ro.boottime.vendor.ril-daemon]: [9581217087]
console:/system/lib # getprop | grep ril
[init.svc.vendor.ril-daemon]: [running]
[init.svc_debug_pid.vendor.ril-daemon]: [502]
[rild.libpath]: [/system/lib/libreference-ril.so]
[ro.boottime.vendor.ril-daemon]: [9581217087]
```

4. 将 RIL 库 push 到安卓设备中运行。

a. 运行如下命令打开安卓设备文件的操作权限：

```
adb root
```

```
adb remount
```

b. 根据不同的系统，将 RIL 库 push 到不同的路径。具体如下：

Android 8 以前，32 位系统：

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

Android 8 以前，64 位系统：

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

Android 8 及以上，32 位系统：

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

Android 8 及以上，64 位系统：

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

c. 重启安卓设备，用 `getprop|grep ril` 查询，结果中有“`ril.fibocom.version`”属性，说明 RIL 库运行正常，如下图所示：

```
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 57.935/60.009/62.084/2.088 ms
rk3399_roc_pc_plus:/ $ getprop|grep ril
[gsm.version.ril-impl]: [Fibocom_RIL_V10X.06.V1.0.7_001]
[init.svc.ril-daemon]: [running]
[persist.ril.current_usbmode]: [17]
[ril.currentapntype]: [default]
[ril.fibocom.version]: [Fibocom_RIL_V10X.06.V1.0.7_001]
[ril.function.dataonly]: [1]
[rild.libargs]: [-d]
[rild.libpath]: [/vendor/lib64/libreference-ril.so]
[ro.boot.noril]: [true]
[ro.boottime.ril-daemon]: [3637641642]
[ro.ril.ecclist]: [112,911]
rk3399_roc_pc_plus:/ $ su
:/ # find -name rild 2>/dev/null
./vendor/bin/hw/rild
1|:/ #
```

至此 RIL 库已集成成功，接下来选择拨号方式。

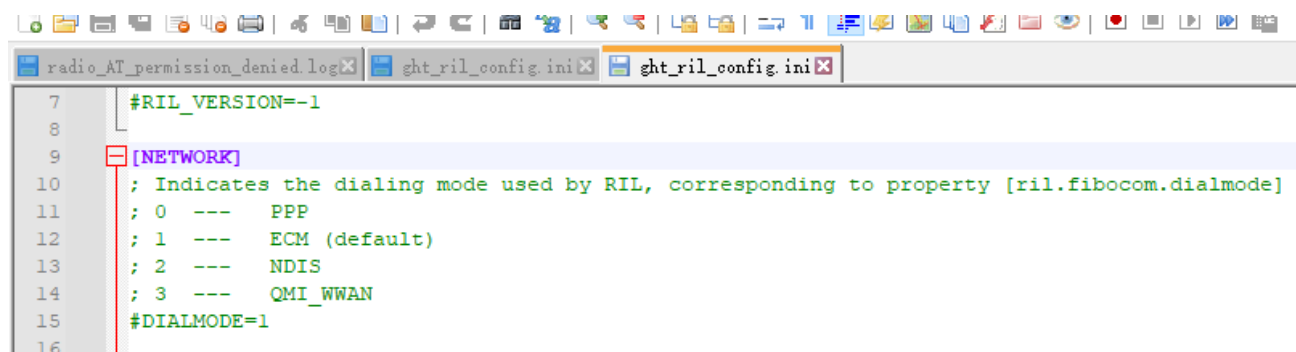
3.7 选择数据拨号方式

目前可选的拨号方式有 ECM、PPP 和 QMI_WWAN 三种拨号方式，广和通 RIL 库默认使用 ECM 拨号，所以使用 ECM 拨号可以直接跳过本章。如果客户希望切换成其它拨号方式请参考如下步骤。

1. 对不同 RIL 版本，修改拨号方式的方法不同，具体如下：

- Fibocom_Android_RIL_1.0.6 及以后

如果是 Fibocom_Android_RIL_1.0.6 及以后的 RIL 版本，建议修改 `ght_ril_config.ini` 文件里的 `DIALMODE` 值，其中 NDIS 拨号（值为 2）不建议使用。



- Fibocom_Android_RIL_1.0.6 以前

Fibocom_Android_RIL_1.0.6 版本以前的 RIL 不支持 `ght_ril_config.ini` 文件，所以需要设置如下属性值：

```
ril.fibocom.dialmode=<dialmode> //0 : PPP
                                //1 : ECM
                                //3 : QMI_WWAN
```

具体的属性设置方法请参考[属性设置方法](#)章节。

2. 然后连接模组与安卓设备并插入 SIM 卡，然后重启安卓设备并等待 30s 左右：

- 如果设备能上网，或者 ping 命令能正常返回，则说明拨号成功，如下图所示：

```
~C
--- ins-r23tsuuf.ias.tencent-cloud.net ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 65.139/76.375/87.611/11.236 ms
rk3399_roc_pc_plus:/ $ ping www.qq.com
PING ins-r23tsuuf.ias.tencent-cloud.net (183.194.238.19) 56(84) bytes of data.
64 bytes from (183.194.238.19): icmp_seq=1 ttl=246 time=70.1 ms
64 bytes from (183.194.238.19): icmp_seq=2 ttl=246 time=86.3 ms
64 bytes from (183.194.238.19): icmp_seq=3 ttl=246 time=86.8 ms
64 bytes from (183.194.238.19): icmp_seq=4 ttl=246 time=83.7 ms
64 bytes from (183.194.238.19): icmp_seq=5 ttl=246 time=60.8 ms
64 bytes from (183.194.238.19): icmp_seq=6 ttl=246 time=62.3 ms
~C
--- ins-r23tsuuf.ias.tencent-cloud.net ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5008ms
rtt min/avg/max/mdev = 60.827/75.054/86.890/11.042 ms
rk3399_roc_pc_plus:/ $
```

- 如果不能上网，请参考[无法上网](#)章节排查。

4 常见问题

4.1 如何抓取 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
```

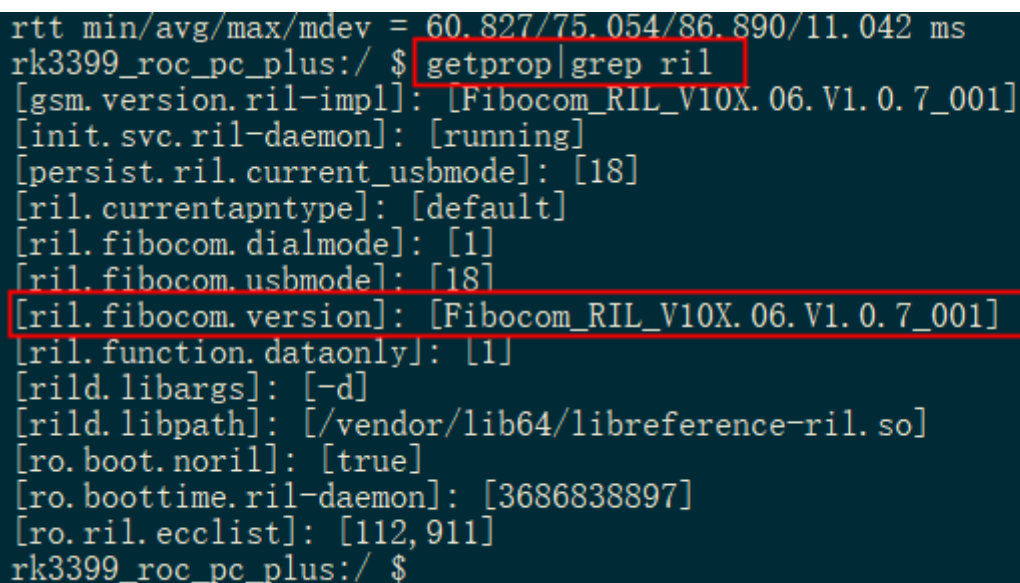
抓到的日志将被保存到“radio.txt”文件中。

4.2 无法上网的排查方法

如果安卓设备无法上网，可以按照如下步骤排查问题：

4.2.1 检查基本信息

1. 先用 adb shell 命令进入安卓设备，然后用 getprop|grep ril 命令查看是否有“ril.fibocom.version”属性，如下图所示：



```
rtt min/avg/max/mdev = 60.827/75.054/86.890/11.042 ms
rk3399_roc_pc_plus:/ $ getprop|grep ril
[gsm.version.ril-impl]: [Fibocom_RIL_V10X.06.V1.0.7_001]
[init.svc.ril-daemon]: [running]
[persist.ril.current_usbmode]: [18]
[ril.currentapntype]: [default]
[ril.fibocom.dialmode]: [1]
[ril.fibocom.usbmode]: [18]
[ril.fibocom.version]: [Fibocom_RIL_V10X.06.V1.0.7_001]
[ril.function.dataonly]: [1]
[rild.libargs]: [-d]
[rild.libpath]: [/vendor/lib64/libreference-ril.so]
[ro.boot.noril]: [true]
[ro.boottime.ril-daemon]: [3686838897]
[ro.ril.ecclist]: [112,911]
rk3399_roc_pc_plus:/ $
```

- a. 如果没有该属性，说明 RIL 库没有被正常加载。请先参考 [RIL 集成配置](#) 章节配置 RIL 库。
- b. 如果 RIL 库运行正常，请使用 ls /dev/ttyUSB* 命令确认模组是否识别出了端口：

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

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
```

如果没有识别出这些端口，说明模组没有被正常挂载。先根据[适用型号](#)章节确定模组是否适配过 RIL，然后需要检查模组供电是否正常、模组和安卓设备的连接是否正常等。

2. 如果模组正常识别出了端口，请确认 SIM 卡是否正常，没有欠费、注销等现象，并且天线连接是否正常。

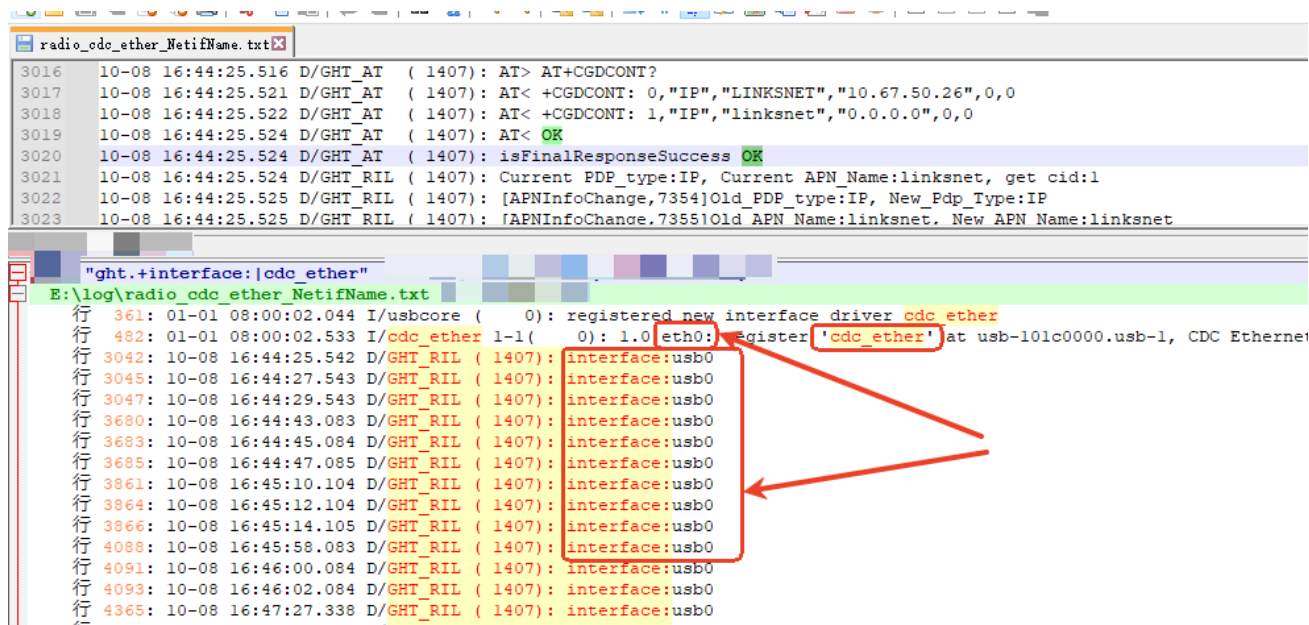
4.2.2 检查网卡名是否正确(ECM)

如果当前使用 ECM 拨号，并且是 Fibocom_Android_RIL_1.0.7 及以前的 RIL 版本，那么需要检查 ECM 网卡的名称是否符合预期，具体方法如下：

1. 开机用 `logcat -b all` 命令开始抓 RIL LOG，抓 2-3 分钟左右。

2. 在 RIL LOG 中搜索正则表达式“`glt.+interface:|cdc_ether`”，搜索结果里二者对应的网卡名称必须相同。

下图是二者不同的异常示例：



此时需要添加如下属性：

```
ril.fibocom.NetifName=<cdc_name> //<cdc_name>是"cdc_ether"搜出的网卡名，上图是"eth0"
```

添加属性方法请参考[属性设置方法](#)章节。

4.2.3 检查路由是否正确(ECM)

4.2.3.1 问题现象

如果是 ECM 拨号，并且安卓 UI 界面出现了 4G 图标但是无法上网，那么问题原因可能是模组网络和安卓以太网产生冲突，导致路由设置失败。

该现象一般在高版本安卓（如安卓 13、14 等）比较容易出现，可以通过如下现象确认：

1. 用“ifconfig”查出的安卓上位机识别出的网卡名称是“ethx”，有 IP 地址但是 PING 不通：

```
D:\wu\cmdr $ adb shell
android:/ $ ifconfig
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:1000
            RX bytes:0 TX bytes:0

eth1        Link encap:Ethernet  HWaddr f0:4b:b3:b9:eb:e5  Driver cdc_ether
            inet addr:10.233.43.209 Bcast:10.233.43.255  Mask:255.255.255.0
            inet6 addr: fe80::ed38:8bfb:3de:17b1/64 Scope: Link
            UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
            RX packets:28 errors:0 dropped:0 overruns:0 frame:0
            TX packets:48 errors:0 dropped:0 overruns:0 carrier:0
            collisions:0 txqueuelen:1000
            RX bytes:7071 TX bytes:6013

android:/ $ ping 8.8.8.8
connect: Network is unreachable
2|android:/ $ |
```

2. 用“netstat -r”查询的路由结果里没有“default”关键字，但是用“getprop | grep net”查到“net.xx.gw”属性有网关地址：

```
2|android:/ $ netstat -r
Kernel IP routing table
Destination      Gateway          Genmask         Flags         MSS Window  irtt  Iface
10.233.43.0      *                255.255.255.0   U              0 0        0     eth1
android:/ $ getprop | grep net
[gsm.network.type]: [LTE]
[init.svc.netd]: [running]
[init.svc.debug_pid.netd]: [451]
[net.bt.name]: [Android]
[net.eth1.dns1]: [211.137.130.2]
[net.eth1.dns2]: [211.137.130.18]
[net.eth1.gw]: [10.233.43.1]
[net.eth1.local-ip]: [10.233.43.209]
[net.tcp_def_init_rwnd]: [60]
[persist.netd.stable_secret]: [60e2:ba12:5ab0:4674:211b:9a0b:9b8d:97bb]
[ro.boottime.netd]: [9813101712]
[ro.rk.ethernet_settings]: [true]
```

3. 并且用如下“ip route”命令设置默认网关成功后，“netstat -r”结果里可以看到“default”关键字，并且预期可以 ping 通网络。

```
su
```

```
ip route add default via <gw> dev <netif>  //<gw>是“net.xx.gw”属性里的 IPV4 网关地址
                                             //<netif>是当前网卡名称，如上图的“eth1”
```

```
[persist.netd.stable_secret]: [60e2:ba12:5ab0:4674:211b:9a0b:9b8d:97bb]
[ro.boottime.netd]: [9813101712]
[ro.rk.ethernet_settings]: [true]
[ro.telephony.default_network]: [9]
android:/ $ su
android:/ # ip route add default via 10.233.43.1 dev eth1
android:/ # netstat -r
Kernel IP routing table
Destination        Gateway            Genmask           Flags   MSS Window  irtt Iface
default            10.233.43.1       0.0.0.0           UG      0 0      0 eth1
10.233.43.0        *                 255.255.255.0     U       0 0      0 eth1
android:/ # ping 8.8.8.8
PING 8.8.8.8 (8.8.8.8) 56(84) bytes of data.
64 bytes from 8.8.8.8: icmp_seq=1 ttl=51 time=142 ms
64 bytes from 8.8.8.8: icmp_seq=2 ttl=51 time=77.3 ms
^C
--- 8.8.8.8 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 77.379/109.926/142.474/32.549 ms
android:/ #
```

4.2.3.2 解决方法

如果上述现象全部符合，请按照[如何修改 ECM 拨号使用的网卡名称？](#)章节将模组识别出的网卡名称改成非"ethx"(例如"lte"等)，避免和以太网冲突。

4.2.4 检查 ip-up 和 ip-down 文件(PPP)

如果是 PPP 拨号，请按照如下内容编写 ip-up、ip-down 文件，并将它们放在/system/etc/ppp 目录下。

4.2.4.1 ip-up 文件

```
#!/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
```

4.2.4.2 ip-down 文件

```
#!/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" ""
```

4.2.4.3 修改权限

使用如下命令将 ip-up 和 ip-down 的执行权限改成 555:

```
chmod 555 /system/etc/ppp/ip-up
chmod 555 /system/etc/ppp/ip-down
```

4.2.5 抓 LOG 给广和通分析

如果按上面步骤排查完后如果还是无法上网，请抓取 LOG 并联系广和通技术支持（抓 LOG 方法请参考[如何抓取 RIL 日志](#)章节）。

4.3 无法拨通电话

如果无法拨通电话，确认 SIM 卡是否支持语音功能，并且向广和通技术支持确认模块在当前网络制式下是否支持电话功能。

4.4 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，并且向广和通技术支持确认模块在当前网络制式下是否支持短信功能。

4.5 如何设置 APN?

APN 的设置有两种方式，分别是在安卓 UI 上设置和修改安卓设备里的/system/etc/apns-conf.xml 文件。

4.5.1 UI 界面设置 APN

请执行如下操作（仅适用于调试）：

1. 将模组安装在 Android 开发板上
2. 在开发板上插入 SIM 卡
3. 打开电源，等待开机
4. 在安卓的 UI 界面上，单击“设置->无线和网线（更多）->移动网络->接入点名称（APN）”，选择指定 APN 进行修改，也可以单击右上角菜单键选择“新建 APN”来添加 APN。输入 Name，APN，MCC，MNC 等，单击右上角菜单键保存。

4.5.2 修改 apns-conf.xml 文件

修改 apns-conf.xml 文件时，根据 SIM 卡的实际信息进行修改或添加，添加的主要信息有：apn，mcc，mnc，user，password，type，protocol 等，如下所示：

```
<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 配置后，由于 Android 设备多次启动，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。删除命令如下：

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



telephony.db 的路径在不同设备上可能略有不同，按实际情况而定。

4.6 如何修改 ECM 拨号使用的网卡名称？

如果修改 ECM 使用的网卡名称，可以修改安卓源码的 kernel/drivers/net/usb/usbnet.c 文件，通过 VID/PID 来匹配网卡名。如下（VID/PID 根据实际情况修改）：

```
if ((dev->driver_info->flags & FLAG_ETHER) != 0 &&
    ((dev->driver_info->flags & FLAG_POINTTOPOINT) == 0 ||
    (net->dev_addr[0] & 0x02) == 0)) {
    #if 1
    if ((le16_to_cpu(
        xdev->descriptor.idVendor) == 0x19d2) &&
        (le16_to_cpu(
            xdev->descriptor.idProduct) == 0x0199))
        strcpy(net->name, "lte%d");
    else if ((le16_to_cpu(
        xdev->descriptor.idVendor) == 0x2CB7) &&
        (le16_to_cpu(
            xdev->descriptor.idProduct) == 0x0001))
        strcpy(net->name, "lte%d");
    else
    #endif
        strcpy(net->name, "eth%d");
}
```

上图中的“lte%d”也可以改成其它名称，如“usb%d”等。

对 Fibocom_Android_RIL_1.0.7 及以前的 RIL 版本，修改完 usbnet.c 文件后，要同步修改 RIL 库使用的网卡名称，否则可能导致 RIL 拨号失败。

例如，假设修改后用“ifconfig -a”命令查出的网卡名称是“lte0”，则需要新增如下属性(新增方法参考[属性设置方法](#)章节)：

```
ril.fibocom.NetifName=lte0
```

Fibocom_Android_RIL_1.0.7 以后的 RIL 无需执行该操作。

4.7 ght_ril_config.ini 文件

4.7.1 使用说明

ght_ril_config.ini 文件用于配置 RIL 的一些选项，例如拨号方式、模组 USBMODE 等，它适用于 Fibocom_Android_RIL_1.0.6 及以后的 RIL 版本。该文件在 RIL 版本文件夹的路径如下。


```
ght@:~/work/Fibocom_Android_RIL_1.0.7/config$  
ght@:~/work/Fibocom_Android_RIL_1.0.7/config$  
ght@:~/work/Fibocom_Android_RIL_1.0.7/config$  
ght@:~/work/Fibocom_Android_RIL_1.0.7/config$ ls  
ght_ril_config.ini  
ght@:~/work/Fibocom_Android_RIL_1.0.7/config$  
ght@:~/work/Fibocom_Android_RIL_1.0.7/config$  
ght@:~/work/Fibocom_Android_RIL_1.0.7/config$  
ght@:~/work/Fibocom_Android_RIL_1.0.7/config$
```

该文件放在安卓设备的/system/etc 目录下生效(Fibocom_Android_RIL_1.0.0.13 及以后的 RIL 版本可以放在 /vendor/etc/ 或 /system/etc/ 路径, /vendor/etc/路径优先, 并且建议放在/vendor/etc/路径)。



请在广和通技术支持指导下使用该配置文件。

4.7.2 空三角时请打开 SIGNAL_POLL 选项

4.7.2.1 问题现象

在高版本的安卓设备(如安卓 13、14)上, 有时会出现数据连接正常, 但是信号图标显示为空三角的现象, 如下图。



4.7.2.2 解决方法

对 Fibocom_Android_RIL_1.0.0.9 及以后的 RIL 版本，将 [ghl_ril_config.ini 文件](#) 中 SIGNAL_POLL 的值改成 1 来解决该问题。

```
ghl_ril_config.ini
44 ; Control the delay time before DHCP, default to 0s.
45 #DHCP_DELAY=1
46
47 ; Enable/disable the signal polling mechanism. Check the signal every 10 seconds, and if there is a s
48 ; it to the upper level. The default value is 0.
49 SIGNAL_POLL=1
50
51 ; Set the period for signal polling. The range is 10 to 600, in seconds, with a default value of 10.
52 #SIGNAL_POLL_PERIOD=10
53
54 ; Enable or disable sending 'ip rule add from all lookup main pref 31999' commands.
55 ; 0 : Disalbe(Default)
56 ; 1 : Enable
57 #IP_RULE=0
58
59 ; Manually set the default route for IPV4. This feature is disabled by default.
60 ; 0 : Disable(Default)
61 ; 1 : Enable
62 #SET_V4_GW=0
63
64 ; Enable/disable the cops polling mechanism.
65 #COPS_POLL=0
66
67 ; Set the period for cops polling. The range is 2 to 600. in seconds. with a default value of 20.
```

修改后预期的正常现象如下图：



4.7.3 NL668 使用切卡功能时请打开 CGLA_SLEEP 选项

为确保 NL668 切卡功能正常，必须在操作前将 CGLA_SLEEP 选项设置为 3，否则可能导致切卡后读取到的 IMSI 信息仍为上一张 SIM 卡。

```
ght_ril_config.ini x
38 #CGLA_FLAG=011
39
40 ; Set the sleep time of CGLA after receiving the BF31 response to wa
41 ; from 0 to 10. The default value is 0.
42 CGLA_SLEEP=3
43
44 ; Control the delay time before DHCP, default to 0s.
45 #DHCP_DELAY=1
46
47 ; Enable/disable the signal polling mechanism. Check the signal ever
48 ; it to the upper level. The default value is 0.
49 #SIGNAL_POLL=0
```

4.8 RIL LOG 中显示“Permission denied”解决方法

4.8.1 问题现象

如果发现 RIL LOG 中无 AT 命令下发，并且有如下“Permission denied”的打印，说明 RIL 当前无权限操作 AT 口，无法收发 AT 命令。

```
radio_AT_permission_denied.log
208 10-30 06:42:36.985 D/GHT_RIL ( 630): ttyModem = [ttyUSB1]
209 10-30 06:42:36.985 D/GHT_RIL ( 630): ttyAT = [ttyUSB2]
210 10-30 06:42:36.985 D/GHT_RIL ( 630): getUsbPortInfo diagchannel :[/dev/ttyUSB0]
211 10-30 06:42:36.985 D/GHT_RIL ( 630): getUsbPortInfo atchannel :[/dev/ttyUSB2]
212 10-30 06:42:36.985 D/GHT_RIL ( 630): getUsbPortInfo datachannel :[/dev/ttyUSB1]
213 10-30 06:42:36.985 D/GHT_RIL ( 630): opening AT interface. retrying...:Permission denied
214 10-30 06:42:37.746 D/TelephonyRegistry( 688): listen oscl: mHasNotifySubscriptionInfoChangedOccurred==false no callback
215 10-30 06:42:37.749 D/TelephonyRegistry( 688): listen oscl: mHasNotifySubscriptionInfoChangedOccurred==false no callback
216 10-30 06:42:37.780 D/TelephonyRegistry( 688): listen oscl: mHasNotifySubscriptionInfoChangedOccurred==false no callback
217 10-30 06:42:37.825 D/TelephonyRegistry( 688): systemRunning register for intents
218 10-30 06:42:37.852 D/TelephonyRegistry( 688): listen oscl: mHasNotifySubscriptionInfoChangedOccurred==false no callback
219 10-30 06:42:37.985 D/GHT_RIL ( 630): open port fail, retry usb detect
220 10-30 06:42:37.986 D/GHT_RIL ( 630): find fibocom module idVendor:[1508] idProduct:[1001]
221 10-30 06:42:37.986 D/GHT_RIL ( 630): usb_device_info.usbdevice_path:[/sys/bus/usb/devices/l-1.4]
222 10-30 06:42:37.986 E/GHT_RIL ( 630): can not find usb serial option driver
223 10-30 06:42:37.986 D/GHT_RIL ( 630): Property ril.fibocom.ttyDiag not set, so try to get ttyname via USB.
224 10-30 06:42:37.986 D/GHT_RIL ( 630): find /sys/bus/usb/devices/l-1.4:1.0/ttyUSB0
225 10-30 06:42:37.986 D/GHT_RIL ( 630): Property ril.fibocom.ttyModem not set, so try to get ttyname via USB.
226 10-30 06:42:37.986 D/GHT_RIL ( 630): find /sys/bus/usb/devices/l-1.4:1.1/ttyUSB1
227 10-30 06:42:37.986 D/GHT_RIL ( 630): Property ril.fibocom.ttyAT not set, so try to get ttyname via USB.
```

4.8.2 解决方法

1. 检查 init.rc 或 rild.rc 中 RIL 服务(一般是"service ril-daemon"或"service vendor.ril-daemon") 里是否有"user radio"的关键字, 如果有, 改成"user root"即可。

```
index b7fe83e7b8d..da8f05f8c68 100644
--- a/hardware/ril/rild/rild.rc
+++ b/hardware/ril/rild/rild.rc
@@ -1,5 +1,5 @@
 service vendor.ril-daemon /vendor/bin/hw/rild -l /vendor/lib64/libreference-ril.so
 class main
- user radio
+ user root
 group radio cache inet misc audio log readproc wakelock
 capabilities BLOCK_SUSPEND NET_ADMIN NET_RAW
 publicusers@pos-code-server:~/lizes/Android/rk3576_android14.0$
```

init.rc或rild.rc 在安卓源码里的路径一般如下:

```
hardware/ril/rild/rild.rc
device/xx/xx/init.rc
```

4.9 如何设置自定义属性

广和通有很多自定义的系统属性，用来控制 RIL 的拨号方式、网卡名称等。客户可以修改这些属性的值来修改一些配置。

4.9.1 属性设置方法

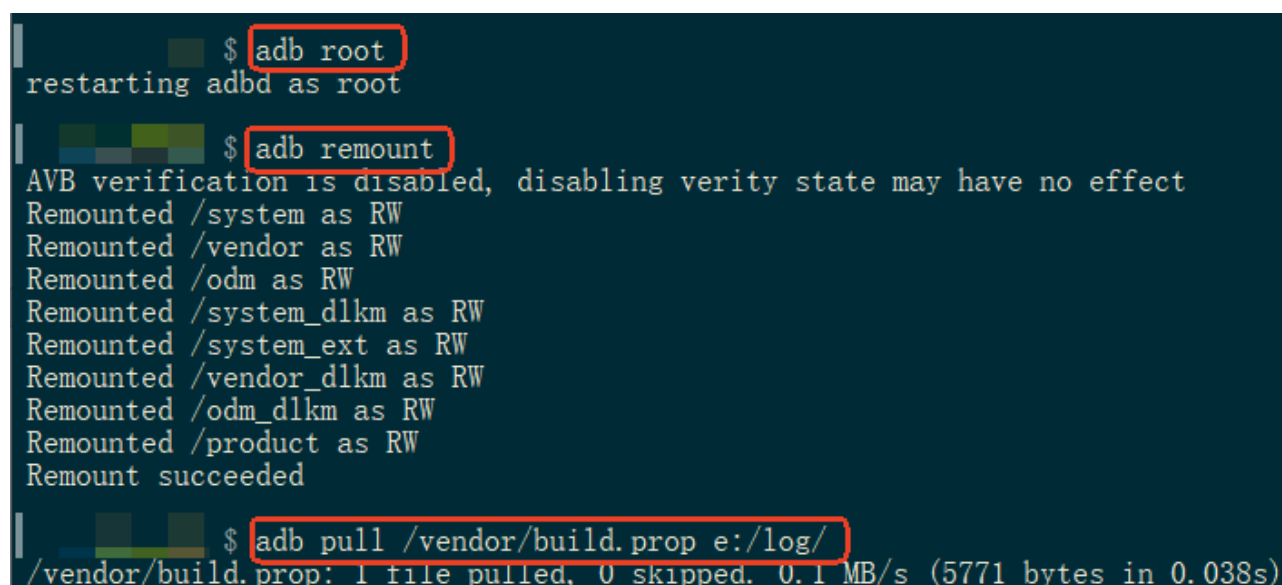
在属性文件里增加属性和对应的值，就可以永久地保存这些属性，属性文件的路径如下：

```
/system/build.prop    //❶安卓 7 及以前建议使用  
/vendor/build.prop    //❷安卓 8 及以后建议使用
```

下面以“/vendor/build.prop”文件为例，介绍设置属性的具体步骤，安卓 7 及以前只需将“vendor”替换成“system”即可。

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

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



```
$ adb root  
restarting adbd as root  
  
$ adb remount  
AVB verification is disabled, disabling verity state may have no effect  
Remounted /system as RW  
Remounted /vendor as RW  
Remounted /odm as RW  
Remounted /system_dlkm as RW  
Remounted /system_ext as RW  
Remounted /vendor_dlkm as RW  
Remounted /odm_dlkm as RW  
Remounted /product as RW  
Remount succeeded  
  
$ adb pull /vendor/build.prop e:/log/  
/vendor/build.prop: 1 file pulled, 0 skipped. 0.1 MB/s (5771 bytes in 0.038s)
```

2. 在 build.prop 文件末尾加入自定义的属性(以“ril.fibocom.dialmode”为例)，然后保存修改：

```
build.prop
161 ro.sf.lcd_density=280
162 ro.wifi.sleep.power.down=true
163 persist.wifi.sleep.delay.ms=0
164 persist.bt.power.down=true
165 vendor.hwc.device.primary=HDMI-A
166 vendor.hwc.device.extend=DSI
167 persist.vendor.resolution.main=Auto
168 ril.fibocom.dialmode=1
169 # end of file
170
```

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 /vendor/build.prop //①将 build.prop 文件重新 push 进设备里
adb reboot //②重启设备
```

```
$ adb push e:/log/build.prop /vendor/build.prop
e:/log/build.prop: 1 file pushed, 0 skipped. 2.0 MB/s (5794 bytes in 0.003s)
```

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

```
adb shell //①进入安卓设备
getprop | grep ril //②查找与“ril”相关的属性
```

```
D:\wu\cmdr $
D:\wu\cmdr $
D:\wu\cmdr $ adb shell
rk3568_firefly_aioj:/ # getprop|grep ril
[init.svc.ril-daemon]: [running]
[init.svc_debug_pid.ril-daemon]: [11621]
[persist.ril.current_usbmode]: [32]
[ril.fibocom.dialmode]: [1]
[ril.fibocom.version]: [Fibocom_Android_RIL_V11X.06.V1.0.0.8]
[ril.function.dataonly]: [1]
[rild.libargs]: [-d]
[rild.libpath]: [/vendor/lib64/libquectel-ril.so]
[ro.boot.noril]: [false]
[ro.boottime.ril-daemon]: [6984883646]
[ro.ril.ecclist]: [112,911]
rk3568_firefly_aioj:/ #
```

我们自己设置的属性

4.9.2 ril.fibocom.dialmode

描述

该属性用于指定 RIL 使用的拨号方式是 ECM 还是 PPP 拨号。

取值范围

表 5. dialmode 取值范围

ril.fibocom.dialmode	拨号方式
0	PPP
1(默认)	ECM
3	QMI_WWAN

示例

```
build.prop
113 persist.nwd.tdgps.highspeed=1
114 drm.service.enabled=true
115 ro.rksdk.version=ANDROID13_RKR5
116 dalvik.vm.boot-dex2oat-threads=4
117 dalvik.vm.dex2oat-threads=4
118 ro.boot.noril=false
119 ro.telephony.default_network=9
120 ril.fibocom.dialmode=1
121 #ril.fibocom.usbmode=73
122 vendor.rild.libpath=/vendor/lib/libreference-ril.so
123 ro.rk.bt_enable=true
124 persist.bluetooth.btsnoopenable=false
125 persist.bluetooth.btsnoopath=/sdcard/btsnoop_hci.cfa
```

4.9.3 ril.fibocom.NetifName

描述

自定义 RIL 拨号使用的网卡。

RIL 发起拨号时，选择网卡的顺序是：NetifName 指定->默认网卡（如 usb0 或 ppp0）。

取值范围

常用的取值为 usb0、eth0 和 eth1 等。

示例

```
build.prop
116 dalvik.vm.boot-dex2oat-threads=4
117 dalvik.vm.dex2oat-threads=4
118 ro.boot.noril=false
119 ro.telephony.default_network=9
120 ril.fibocom.dialmode=1
121 #ril.fibocom.usbmode=73
122 ril.fibocom.NetifName=usb0
123 vendor.rild.libpath=/vendor/lib/libreference-ril.so
124 ro.rk.bt_enable=true
125 persist.bluetooth.btsnoopenable=false
126 persist.bluetooth.btsnoopath=/sdcard/btsnoop_hci.cfa
127 persist.bluetooth.btsnoopsize=0xffff
128 persist.bluetooth.rtkcoex=true
```

4.9.4 ril.fibocom.usbmode

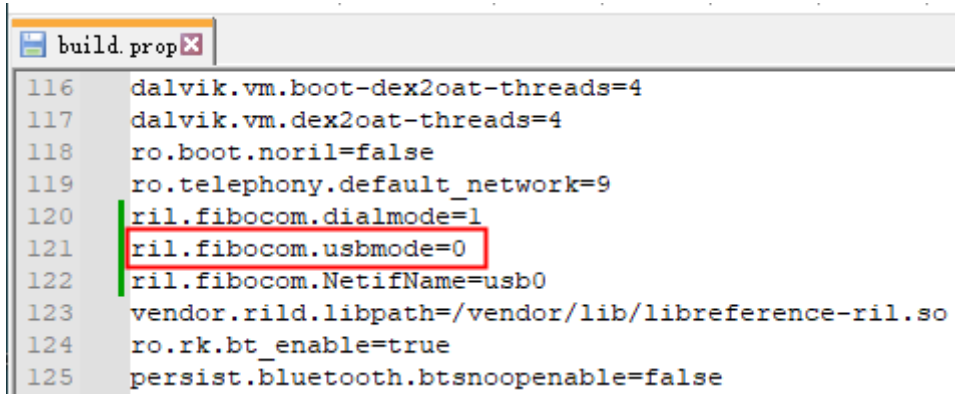
描述

该属性用于修改模块的 USBMODE 为指定值。

取值范围

该属性的取值范围是 0 和[17-75]，默认值是 0。不设置该属性或该属性值为 0 时，表示不修改模块的 USBMODE。

示例



```
build.prop
116 dalvik.vm.boot-dex2oat-threads=4
117 dalvik.vm.dex2oat-threads=4
118 ro.boot.noril=false
119 ro.telephony.default_network=9
120 ril.fibocom.dialmode=1
121 ril.fibocom.usbmode=0
122 ril.fibocom.NetifName=usb0
123 vendor.rild.libpath=/vendor/lib/libreference-ril.so
124 ro.rk.bt_enable=true
125 persist.bluetooth.btsnoopenable=false
```


附录 A 拨号方式和 USBMODE

不同的拨号方式（例如 PPP、ECM 或 QMI_WWAN）对应模组不同的 USBMODE。具体关系如下表：

表 6. 支持的模组型号

Module Types	VID;PID	USBMODE		
		PPP	ECM	QMI_WWAN
L610	1782;4D10 1782;4D11	31	32	NA
MC61x: MC615、MC619	2CB7;0A01 2CBC;0A02 2CB7;0A03	31	32	NA
NL668: NL668-CN、NL668-JP	1508;1001	18	18	NA
MG110	1508;1001	18	18	NA
NL668T: NL668T-GL	2CA3;4009	NA	31	30
L71x: L716、L718	2CB7;0001 2CB7;0579	10 或 11	10	NA
MC919: MC919-CN	2CB7;0C01 2CB7;0C02 2CB7;0C03	72	70	NA
MC66x: MC660/MC661/MC665/MC667/MC669	2CB7;0A0A 2CB7;0A0B 2CB7;0A0C	73	74	NA
MG66x: MG661				
FG132	2CB7;010B 2CB7;0112 2CB7;0113	NA	38	36
MA510	2CB7;0106 2CB7;010A	31 或 32	31 或 32	NA



"NA"表示不支持该拨号方式。

附录 B 常用术语和缩写

RIL 中用到的常用术语和缩写见下表：

表 7. 术语和缩写

缩写词	描述
GSM	Global System for Mobile Communications(全球移动通信系统)
VID	Vendor ID(供应商 ID)
PID	Product ID(产品识别码)
RIL	Radio Interface Layer(无线接口层)
RILD	Radio Interface Layer Daemon(无线接口层守护进程)
APN	Access Point Name(接入点名称)

附录 C RIL Log 中 TAG 标签说明

RIL Log 中各个进程产生的日志都保存在同一个 Log 文件中，可以通过 TAG 标签来区分 Log 来自哪个进程，TAG 标签与对应的进程关系如下表所示。

表 8. RIL Log 中 TAG 标签说明

TAG	进程
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