ML302 Android5.0.2 驱动说明文档

重要声明

版权声明

商标声明

中移物联网有限公司和中移物联网有限公司的产品是中移物联网有限公司专有。在提及其他公司及其产品时将使用各自公司所拥有的商标，这种使用的目的仅限于引用。

不作保证声明

中移物联网有限公司不对此文档中的任何内容作任何明示或暗示的陈述或保证，而且不对特定目的的适销性及适用性或者任何间接、特殊或连带的损失承担任何责任。

保密声明

本文档（包括任何附件）包含的信息是保密信息。接收人了解其获得的本文档是保密的，除用于规定的目的外不得用于任何目的，也不得将本文档泄露给任何第三方。

1.配置 Android 侧 USB 驱动

（1）ML302 使用 USB 接口，对上层的功能接口是串口，在 Linux 系统中使用 USB 转串口的驱动。

驱动加载需要首先配置 Linux 内核，配置方法如下：

在 Android 源码的 linux 目录下输入命令：Make menuconfig 。

在出现的框图中找到 Device drivers->usb support->usb serial converter support

选中如下组件的支持：

USB driver for GSM and CDMA modems

选择后保存设置。

（2）

A、添加 RNDIS 内核支持

```
Device Drivers --->
    [*] Network device support --->
        USB Network Adapters --->
            <*> Multi-purpose USB Networking Framework
            <*> Host for RNDIS and ActiveSync devices
```

B、在 init.rc 中添加 ECM 开关

```
setprop ril.cmiot.dialmode ecm
```

（3）打开内核源码文件 option.c(路径 drivers/usb/serial/option.c)，找到 option_ids 数组，添加如下内容：

```
{ USB_DEVICE_AND_INTERFACE_INFO(0x1782, 0x4d10, 0xff, 0, 0) },
{ USB_DEVICE_AND_INTERFACE_INFO(0x1782, 0x4d11, 0xff, 0, 0) },
```

编译好内核烧写后，插上 USB 线连接模组，使用 ls /dev 查看设备，出现 ttyUSB0 到 ttyUSB3 或 ttyUSB7 表示 USB 枚举成功。

2. RIL 驱动安装与调试

（ 1 ），在 Android 源码目录下找 init.rc 文件，输入 find -name init.rc

然后在 init.rc 中，对 service ril-daemon 按照下列所示进行修改（**2019 年 11 月之前的版本为 ttyUSB3**）

service ril-daemon /system/bin/rild -l /system/lib/libref-ril-ml302.so -- -d /dev/**ttyUSB7**

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

注意：ttyUSB0 是与模组通信的 AT 口，先确认完成步骤 1 后出现的是否仅出现了 ttyUSB0-ttyUSB。

对 service pppd_gprs 按照下列所示进行修改 (没有的话就直接添加)

service pppd_gprs /system/bin/ppp/init.gprs-pppd

socket rild-ppp stream 660 root radio

user root

group radio cache inet misc

disabled

oneshot

on property:net.gprs.enable=1

start pppd_gprs

on property:net.gprs.enable=0

stop pppd_gprs

on property:ril.reset.rild=1

stop ril-daemon

start ril-daemon

setprop ril.reset.rild 0

然后保存修改后退出。

（ 2 ），可以选用以下两种方法中的一种安装 libreference-ril.so 库：

第一种方法，找到原来的 libreference-ril.so 的目录位置，输入命令 find -name libreference-ril.so 进入该目录中，将给的 libreference-ril.so 放入该目录下替换掉原来 libreference-ril.so ，最后重新对系统进行编译。

第二种方法，完成了（1）中的修改后，先编译下载，启动开发板。然后用 adb 登录进去，进入/system/lib 目录下，rm libreference-ril.so 删除掉以前的 libreference-ril.so 库，然后在使用 adb push 的命令将驱动包里的 libreference-ril.so 放入到/system/lib 目录下，最后重新启动开发板

（ 3 ），对于版本较高的 Android 系统，进行第（1）步的操作后在命令行中可能会出现 SELinux domain defined; please fix! 的警告，如果出现请参考以下网址进行修改(https://blog.csdn.net/qq_22508501/article/details/78208975)，可通过内核编译配置禁用 SElinux 功能。

4.对于其它 Android 系统的源码替换

(1)，如果要利用源码自己生成库，在做完步骤 3 的（1）步时，先跳过步骤 3 中（2），（3）步。然后在 Android 源码中进入 /hardware/ril 的目录中，对自己原来的 reference-ril 文件夹做个备份然后删除，将驱动包里的 reference-ril 文件夹拷贝入/hardware/ril 的目录下 。然后对系统进行编译，下载后输入 logcat -b radio 查看相关的 log。看看 RIL 的框架是否正常运行。

注意：该 ril 源码在 Android5.0.2 进行的编译，在其它不同版本的 Android 的系统的进行编译可能会编译报错，但应该都是和系统不匹配的错误，可以根据报错的提示进行匹配性的修改调整。

5.注意事项

（1） ，在做第 2 步和第 3 步操作时，请确保 sim 卡的资费正常，并且给 ML302 插上 4G 天线。否则可能会对拨号上网产生误判。

（2），对第 4 步的调试主体思路是先跑通大的 RIL 框架，判断依据主要应用如下的指令 logcat -b radio 查看，

下图是底层执行拨号上网的的一个 log

```
I/use-Rlog/RLOG-RIL( 1657): processRequest() SETUP_DATA_CALL
I/use-Rlog/RLOG-RIL( 1657): my RadioState is 4
D/use-Rlog/RLOG-RIL( 1657): Identifying modem capabilities...
D/use-Rlog/RLOG-AT( 1657): AT(13)> AT^DIALMODE?
I/use-Rlog/RLOG-AT( 1657): AT(13)< ^DIALMODE:0,0
I/use-Rlog/RLOG-AT( 1657): AT(13)< OK
D/use-Rlog/RLOG-RIL( 1657): Modem mode: 0 [0=ppp,1=ndis,2=ppp&ndis
I/use-Rlog/RLOG-RIL( 1657): requesting data connection to APN 'cmnet'
D/use-Rlog/RLOG-RIL( 1657): Identifying modem capabilities...
D/use-Rlog/RLOG-AT( 1657): AT(13)> AT^DIALMODE?
I/use-Rlog/RLOG-AT( 1657): AT(13)< ^DIALMODE:0,0
I/use-Rlog/RLOG-AT( 1657): AT(13)< OK
D/use-Rlog/RLOG-RIL( 1657): Modem mode: 0 [0=ppp,1=ndis,2=ppp&ndis
D/use-Rlog/RLOG-AT( 1657): AT(13)> ATH
I/use-Rlog/RLOG-AT( 1657): AT(13)< OK
D/use-Rlog/RLOG-AT( 1657): AT(13)> AT+CGDCONT=1,"IP","cmnet"
I/use-Rlog/RLOG-AT( 1657): AT(13)< OK
D/use-Rlog/RLOG-AT( 1657): AT(13)> AT+CFUN=1
I/use-Rlog/RLOG-AT( 1657): AT(13)< OK
D/use-Rlog/RLOG-RIL( 1657): Trying to setup PPP connnection...
D/use-Rlog/RLOG-RIL( 1657): Waiting until net ifc ppp0 is up
D/use-Rlog/RLOG-AT( 1657): AT(13)> AT+CGCONTRDP=1
I/use-Rlog/RLOG-AT( 1657): AT(13)< +CGCONTRDP: 1,5,"cmnet","100.20.12.86.255.255.255.252","100.20.12.85","18
3.230.126.225","183.230.126.224",,,
I/use-Rlog/RLOG-AT( 1657): AT(13)< OK
I/use-Rlog/RLOG-RIL( 1657): Got ifname: ppp0, local-ip: 100.20.12.86, dns1: 183.230.126.225, dns2: 183.230.1
26.224, gw: 100.20.12.85
```

正常的话 ppp 拨号成功，底层能够获取到 ppp0 端口，ip 地址，网关和 dns 的信息。

同时，底层会将这些信息上报上去：

```
D/RILJ    ( 2597): [3714]< SETUP_DATA_CALL DataCallResponse: {version=6 status=0 retry=-1 cid=1 active=2 typ
e=PPP ifname=ppp0 mtu=0 addresses=[100.20.12.86] dnses=[183.230.126.225,183.230.126.224] gateways=[100.20.12
.85] pcscf=[]} [SUB0]
```

当系统闲置没有电话相关业务时，会周期性的检测 CSQ 信号强度

```
I/use-Rlog/RLOG-RIL( 1657): processRequest() SIGNAL_STRENGTH
I/use-Rlog/RLOG-RIL( 1657): my RadioState is 4
D/use-Rlog/RLOG-AT( 1657): AT(13)> AT+CSQ
I/use-Rlog/RLOG-AT( 1657):[  179.180397] binder: 2239:2589 transaction failed 29189, size 168-0
 AT(13)< +CSQ: 13,99
I/use-Rlog/RLOG-AT( 1657): AT(13)< OK
I/use-Rlog/RLOG-RIL( 1657): SignalStrength 13 BER: 99
D/RILC    ( 1657): RequestComplete, RIL_SOCKET_1
E/RILC    ( 1657): Send Response to RIL_SOCKET_1
V/RILJ    ( 2597): Read packet: 64 bytes
E/RILJ    ( 2597): processSolicited: 19
D/RILJ    ( 2597): [3718]< SIGNAL_STRENGTH SignalStrength: 13 99 0 0 0 0 0 0 0 0 0 gsm|lte [SUB0]
D/SubController( 2597): [getSubId]- size == 0, return dummy instead
D/DefaultPhoneNotifier( 2597): notifySignalStrength: mRegistry=com.android.internal.telephony.ITelephonyRegi
stry$Stub$Proxy@27e2db11 ss=SignalStrength: 13 99 -120 -160 -120 0 -1 0 2147483647 2147483647 0 0 gsm|lte se
nder=Handler (com.android.internal.telephony.gsm.GSMPhone) {35c63f9e}
```

获取 SIM 卡 ICCID

RIL 驱动里已将 SIM 卡 ICCID 号写入 ril.audio.iccid 属性，可通过读取该属性值获取 ICCID 号码

获取 SIM 卡 IMSI

RIL 驱动里已将 SIM 卡 IMEI 号写入 ril.audio.imsi 属性，可通过读取该属性值获取 IMEI 号码