



Perfect Wireless Experience
完美无线体验

L7 Family Linux/Android系统驱动集成及应用指导

文档版本：V1.0.2

更新日期：2016.06.20





版权声明

版权所有©2015 深圳市广和通无线股份有限公司。保留一切权利。

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。

注意

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，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。

商标申明



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。

版本记录

文档版本	更新日期	说明
V1.0.0	2016-04-25	初始版本
V1.0.1	2016-06-01	添加 ECM 及 Android 集成内容
V1.0.2	2016-06-20	1. linux 下使用 ECM 拨号需要先发送 AT+GTNETCARDTYPE=1 2. 更新目录及文档名称

适用型号

序号	产品型号	说明
1	L710	

目录

1 简介.....	6
1.1 目的.....	6
1.2 范围.....	6
2 Linux 系统使用说明.....	7
2.1 Linux 内核设备驱动架构.....	7
2.2 Linux USB 串口驱动集成.....	8
2.2.1 USB 串口驱动代码移植.....	8
2.2.2 详细配置步骤.....	9
2.3 USB ECM 驱动配置.....	11
2.3.1 ECM 配置确认.....	11
2.3.2 具体配置操作.....	11
2.4 USB 串口及 ECM 驱动配置确认.....	12
2.5 端口形态说明.....	13
2.6 端口测试.....	13
2.6.1 命令行测试.....	13
2.6.2 程序测试.....	13
2.7 ECM 数据上网 AT 指令.....	17
2.8 PPP 拨号上网连接.....	19
2.9 PPP 拨号上网脚本说明.....	20
3 Android 系统使用说明.....	23
3.1 Android 内核驱动移植配置及加载.....	23
3.2 系统集成及调试.....	23
3.2.1 系统与模块端口的通讯.....	23
3.2.2 RIL 集成.....	23
3.2.3 adb 工具安装.....	24
3.2.4 RIL 库更换.....	25
3.2.4.1 RIL 加载判断.....	26
3.2.4.2 RIL 版本号检查.....	27
3.2.5 调试信号的显示、电话程序、短信等功能.....	27
3.2.5.1 模块 IMEI 号查询.....	27
3.2.5.2 信号强度查询.....	27
3.2.5.3 语音及短信检测.....	28

3.2.5.4 数据业务开启.....	29
3.2.5.5 网络优先.....	29
3.2.5.6 调试音频通道切换及音量调节.....	29

1 简介

1.1 目的

- 本文主要是针对 L7 系列 4G 模块设备基于 Linux/Android 系统的驱动集成开发活动进行相关的指导说明。本文档主要面向基于上述系统的产品开发商的驱动开发人员

1.2 范围

本文适用于：

- Android4.2-4.4
- Linux2.6.22 以上版本

2 Linux 系统使用说明

2.1 Linux 内核设备驱动架构

L7 系列 4G 模块和 Linux 系统主要通过 USB 接口进行数据通信。Linux 系统的 Linux 内核需要根据 4G 模块设备上报的 USB 设备接口加载 USB 驱动，USB 驱动正确加载后，模块才能正常工作。

Linux 系统的 Linux 内核驱动架构，如图 2-1 所示：

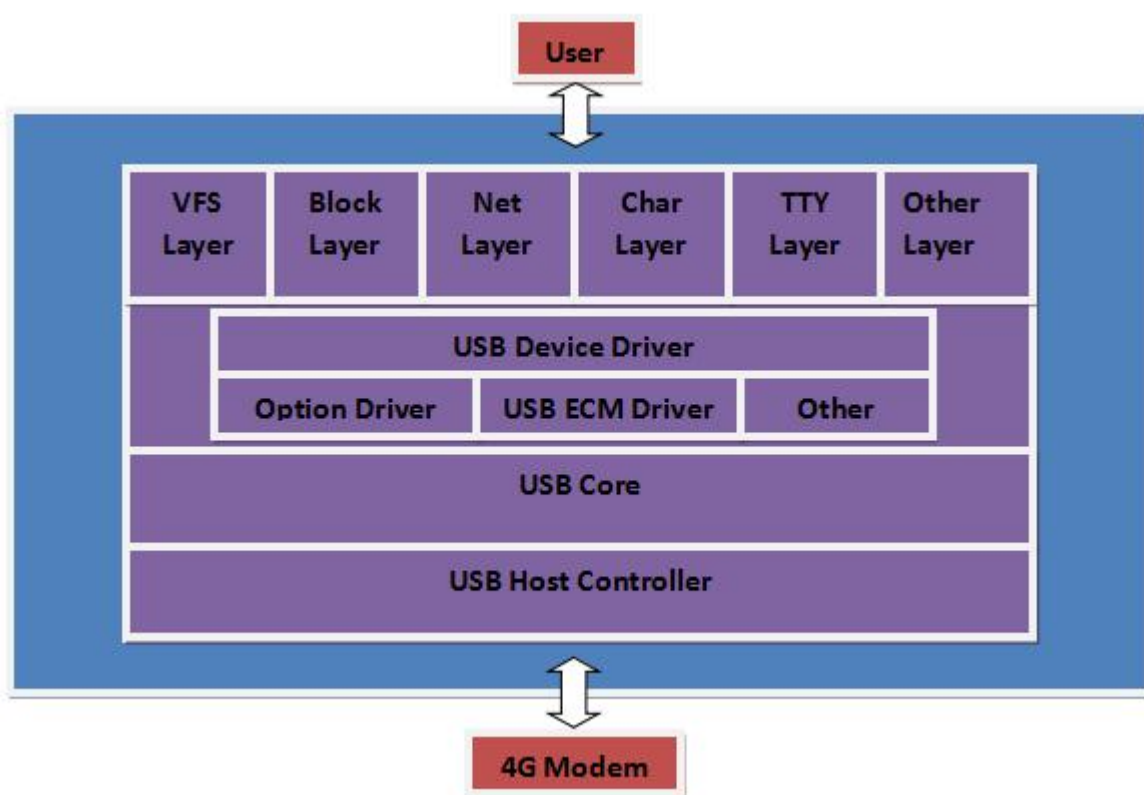


图 2-1 驱动架构

如图 2-1 所示，在 Linux 系统的 USB 驱动架构中与 4G 设备相关的驱动模块是

USB Option 驱动模块



Option 驱动即 USB 转串口驱动，支持如 modem 端口，AT 端口等；该驱动模块的代码（option.c）已经内置于 Linux 内核的源码。

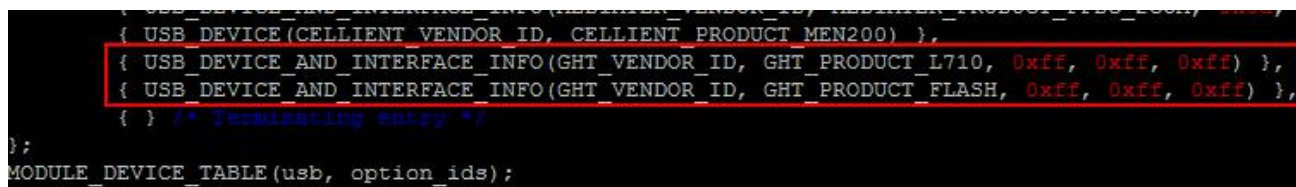


Ecm 驱动，主要用于传输网络数据。

2.2 Linux USB 串口驱动集成

2.2.1 USB 串口驱动代码移植

1. 驱动代码修改 在 drivers/usb/serial/option.c 文件的 `static const struct usb_device_id option_ids[]` 数组内图 2-2 添加红框所示代码：



```

{
    { USB_DEVICE_AND_INTERFACE_INFO(GHT_VENDOR_ID, GHT_PRODUCT_L710, 0xff, 0xff, 0xff) },
    { USB_DEVICE_AND_INTERFACE_INFO(GHT_VENDOR_ID, GHT_PRODUCT_FLASH, 0xff, 0xff, 0xff) },
    { } /* Terminating entry */
};
MODULE_DEVICE_TABLE(usb, option_ids);
    
```

图 2-2 option_ids 代码

图 2-2 代码如下：

```

{ USB_DEVICE_AND_INTERFACE_INFO(GHT_VENDOR_ID, GHT_PRODUCT_L710, 0xff,
0xff, 0xff) }, /* GHT WCDMA products */
    
```

```

{ USB_DEVICE_AND_INTERFACE_INFO(GHT_VENDOR_ID, GHT_PRODUCT_FLASH, 0xff,
0xff, 0xff) }, /* GHT FLASH products */
    
```

上述宏 `GHT_VENDOR_ID` , `GHT_PRODUCT_L710` , `GHT_PRODUCT_FLASH` 定义 , 如下红框所示：



```

#define CELLIENT_PRODUCT_MEN200 0x9005

/* GHT PRODUCTS */
#define GHT_VENDOR_ID 0x2cb7
#define GHT_PRODUCT_L710 0x0001
#define GHT_PRODUCT_FLASH 0x0256

/* some devices interfaces need special handling due to a number of reasons */
    
```

图 2-3 宏

图 2-3 代码如下：

```
/* GHT PRODUCTS */
```

```
#define GHT_VENDOR_ID 0x2cb7
```

```
#define GHT_PRODUCT_L710 0x0001
```

```
#define GHT_PRODUCT_FLASH 0x0256
```


2. 修改内核编译配置(kernel 根目录下的.config 文件中), 确保以下配置项已被选定 :

1) PPP 拨号的相关配置项 :

```
CONFIG_PPP=y
CONFIG_PPP_MULTILINK=y
CONFIG_PPP_FILTER=y
CONFIG_PPP_ASYNC=y
CONFIG_PPP_SYNC_TTY=y
CONFIG_PPP_DEFLATE=y
CONFIG_PPP_BSDCOMP=y
```

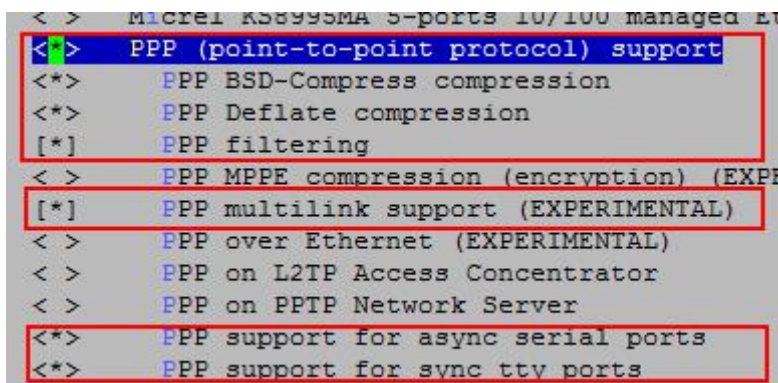
2) USB 串口相关配置项 :

```
CONFIG_USB_SERIAL=y
CONFIG_USB_ANNOUNCE_NEW_DEVICES=y (若此选项存在, 建议配置, 若不存在, 请忽略)
CONFIG_USB_SERIAL_OPTION=y
```

2.2.2详细配置步骤

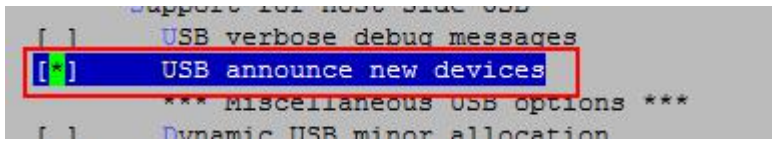
1. 打开 Terminal 工具, 进入 kernel 目录 (假定为 “/home/ght /linux-3.0.8/”), 执行 make <configuration> 命令 (假定使用标准 make menuconfig) 。
2. 按下述指引完成 PPP 拨号相关配置 :

进入 “Device Drivers ”→“ Network device support ”菜单后, 选择红框所有选项



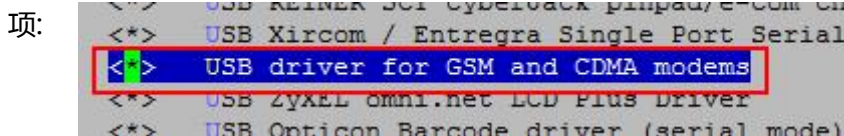
3. 按下述指引完成 announce new devices(若此选项存在, 建议配置, 若不存在, 请忽略)配置 :

进入 “Device Drivers ”→ “USB support” 菜单后选择 USB announce new devices 选项：



4. 按下述指引完成 USB 串口驱动配置:

进入 “Device Drivers ”→ “USB support” → “USB Serial Converter support” 选择红框所有选项:



5. 如上配置后，通过选择 “<Exit>” 逐级退出配置界面。最后在保存界面中选择 “<Yes>” 并退出。

6. 完成配置后，运行 make 命令，编译修改后的内核。

2.3 USB ECM 驱动配置

模块 ECM 端口使用的是标准的 Linux 驱动，所以不需要修改源码，仅需要 ECM 配置即可。

2.3.1 ECM 配置确认

修改内核编译配置(kernel 根目录下的.config 文件中)，确保以下配置项已被选定：

`CONFIG_USB_NET_CDCETHER=y`

`CONFIG_USB_USBNET=y`

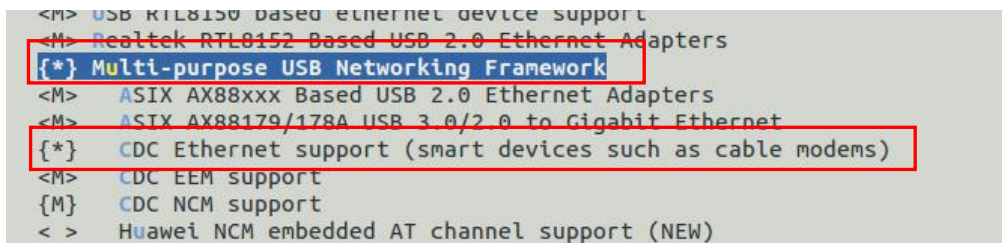
2.3.2 具体配置操作

1. 打开 Terminal 工具，进入 kernel 目录（假定为 “/home/ght/linux-3.0.8/”），执行 make <configuration>

命令（假定使用标准 make menuconfig）。

2. 按照下述指引完成 NCM 驱动配置:

进入 “Device Drivers ”→ “Network device support ”→ “ USB NetworkAdapters ”菜单后选择 Multi-purpose USB Networking Framework 及 CDC Ethernet support (smart devices such as cable modems)选项:



3. 如上配置后，通过选择 “<Exit> ”逐级退出配置界面。最后在保存界面中选择 “<Yes>” 并退出。

4. 完成配置后，运行 make 命令，编译修改后的内核。

2.4 USB 串口及 ECM 驱动配置确认

系统开机启动时，执行 `dmesg` 命令，查看内核打印信息，查找到下述信息即说明系统已经成功配置 USB 串口驱动。

usbcore: registered new interface driver option

usbserial: USB Serial support registered for GSM modem (1-port)

usbcore: registered new interface driver cdc_ether

系统启动后，4G 模块上电开机，执行 `dmesg` 命令，查看内核打印信息，查找图 2-4 红框信息说明 USB 串口驱动已经成功加载。

```
usb 2-3: new high-speed USB device number 11 using xhci_hcd
usb 2-3: New USB device found, idVendor=2cb7, idProduct=0001
usb 2-3: New USB device strings: Mfr=1, Product=2, SerialNumber=1
usb 2-3: Product: Fibocom-Modem
usb 2-3: Manufacturer: Fibocom
usb 2-3: SerialNumber: Fibocom
cdc_ether 2-3:1.0 eth1: register 'cdc_ether' at usb-0000:00:14.0-3,
option 2-3:1.2: GSM modem (1-port) converter detected
usb 2-3: GSM modem (1-port) converter now attached to ttyUSB0
option 2-3:1.3: GSM modem (1-port) converter detected
usb 2-3: GSM modem (1-port) converter now attached to ttyUSB1
option 2-3:1.4: GSM modem (1-port) converter detected
usb 2-3: GSM modem (1-port) converter now attached to ttyUSB2
option 2-3:1.5: GSM modem (1-port) converter detected
usb 2-3: GSM modem (1-port) converter now attached to ttyUSB3
option 2-3:1.6: GSM modem (1-port) converter detected
usb 2-3: GSM modem (1-port) converter now attached to ttyUSB4
```

图 2-4 驱动加载

执行 `ls -al /dev/ttyUSB*` 命令可以查询到 `ttyUSB0`、`ttyUSB1`、`ttyUSB2`、`ttyUSB3`、`ttyUSB4`

```
glt@fibocom:~$ ls /dev/ttyUSB*
/dev/ttyUSB0 /dev/ttyUSB1 /dev/ttyUSB2
glt@fibocom:~$
```

执行 `ifconfig` 命令查询到当前生成的 ECM 网口

```
eth2      Link encap:以太网  硬件地址 00:a0:c6:00:00:00
inet6 地址: fe80::2a0:c6ff:fe00:0/64 Scope:Link
UP BROADCAST MULTICAST  MTU:1500  跃点数:1
接收数据包:0  错误:0  丢弃:0  过载:0  帧数:0
发送数据包:0  错误:0  丢弃:0  过载:0  载波:0
碰撞:0  发送队列长度:1000
接收字节:0 (0.0 B)  发送字节:0 (0.0 B)
```

2.5 端口形态说明

序号	端口名称	端口形态	备注
1	ttyUSB0	Modem Port	用于 PPP 数据业务 在非数据模式下也可以收发 AT 命令
2	ttyUSB1、ttyUSB3、ttyUSB4	Trace Port	用于抓取模块调试信息
3	ttyUSB2	At Port	用于 AT 通讯，即收发 AT 命令
4	eth0(1,2,3)	ECM 网口	用于 ECM 数据通讯

2.6 端口测试

2.6.1 命令行测试

1. 打开 terminal 终端
2. 执行 `echo -e "ate0\r\n" > /dev/ttyUSB2` （在使用前必须先发此命令，否则 cat 命令会有异常）
3. 执行 `echo -e "at+cgmr\r\n" > /dev/ttyUSB2` 查询软件版本
4. 执行 `cat /dev/ttyUSB2 &` 读取查询返回
5. 执行 `echo -e "at+cgdcont=1,\"ip\", \"3gnet\"\r\n" > /dev/ttyUSB2` 设置 APN



若 AT 命令中包含双引号须加转义字符 “\”，格式参照步骤 5。

```

root@fibocom:/# echo -e "ATE0\r\n" > /dev/ttyUSB2
root@fibocom:/# echo -e "AT+CGMR\r\n" > /dev/ttyUSB2
root@fibocom:/# cat /dev/ttyUSB2 &
[1] 14820
root@fibocom:/#
+CGMR: L710_V5C.0E.00.0C_T1
OK

root@fibocom:/# echo -e "at+cgdcont=1,\"ip\", \"3gnet\"\r\n" > /dev/ttyUSB2
root@fibocom:/#
OK

```

2.6.2 程序测试

下方的 C 程序可用于检测端口 AT 的收发。该程序打开 /dev/ttyUSB2 设备节点，调用 write 和 read 函数来发送 AT 命令并接收 AT 回复。

```
#include <stdio.h>

#include <string.h>

#include <unistd.h>

#include <fcntl.h>

#include <errno.h>

#include <termios.h>

#define ATPORT "/dev/ttyUSB2"

#define BUFSIZE 1000

#define BAUDRATE B115200

int open_port(char *port)
{
    struct termios options;

    int fd;

    fd = open(port, O_RDWR | O_NOCTTY | O_NDELAY);

    if (fd == -1) {

        printf("%s: Unable to open the port - \r\n",__func__);

    } else {

        fcntl(fd, F_SETFL, FNDELAY);

        tcgetattr( fd, &options );

        cfsetispeed( &options, BAUDRATE );

        cfsetospeed( &options, BAUDRATE );

        options.c_cflag |= ( CLOCAL | CREAD);

        options.c_cflag &= ~(CSIZE | PARENB | CSTOPB | CSIZE);

        options.c_cflag |= CS8;

        options.c_cflag &= ~CRTSCTS;

        options.c_iflag &= ~(ICANON | ECHO | ECHOE | ISIG);

        options.c_iflag &= ~(IXON | IXOFF | IXANY | ICRNL | INLCR | IGNCR);
```



```

options.c_oflag &= ~OPOST;

if ( tcsetattr( fd, TCSANOW, &options ) == -1 ) {

    printf ("Error with tcsetattr = %s\r\n", strerror ( errno ) );

} else {

    printf ( "Open port succeed\r\n");

}

}

return (fd);

}

int main()

{

    int fd = open_port(ATPORT);

    char at_cmd_ch[50]="AT+CGMR\r\n";

    char buf[BUFSIZE];

    memset(buf,0,BUFSIZE);

    printf("AtSend: %s\r\n", at_cmd_ch);

    write(fd, at_cmd_ch , strlen(at_cmd_ch));

    sleep(1);

    read( fd, buf, BUFSIZE );

    printf("AtRecevie: %s\r\n", buf);

    close(fd);

    return 0;

}

```

将上述代码保存于 TestPort.c 文本中，执行 `gcc -o TestPort TestPort.c` 命令编译出 TestPort 程序，执行编译后的程序即可查看返回结果。

```
root@fibocom:/home/ght# ./TestPort
Open port succeed
AtSend: AT+CGMR
AtRecevie:
+CGMR: L710_V5C.0E.00.0C_T1
OK
```

因为 4G 模块发出 “AT+CGMR” 命令后需要处理时间，所以需要在读之前加上至少 500ms 延迟。上述演示代码中 sleep(1) 仅起参考作用。



sleep(1) 表示延迟 1 秒。

2.7 ECM 数据上网 AT 指令

1. 查询信号、SIM 卡及注网状态

AT+CSQ 返回值参数 1 的范围是 0 – 31 或者 99，如果是 99 表示无信号，请检查天线情况。

AT+CSQ

+CSQ: 23,99

OK

AT+CPIN? 检查 SIM 卡状态，返回 READY，表示 SIM 卡可用，如果返回 SIM PIN，请用 AT+CPIN="correct PIN" 解 PIN。

AT+CPIN?

+CPIN: READY

OK

AT+CPIN?

+CPIN: SIM PIN

OK

AT+CPIN="<correct PIN>"

OK

AT+COPS? 查询运营商选择及注网情况，如果仅有一个参数返回的话，请检查天线，SIM 卡状态是否正常。

AT+COPS?

+COPS: 0,0,"CHINA MOBILE",7

OK

返回值的参数 1 表示，注册模式，0 表示自动，1 表示手动；

返回值的参数 2 表示，显示格式，0 表示长字符串格式，1 表示短字符格式，2 表示字数；

返回值的参数 3 表示，按参数 2 来显示运营商名字，CHINA MOBILE 为中国移动；

返回值的参数 4 表示，注网情况，7 表示 LTE 网，2 表示 UMTS, 0 表示 GSM

2. 数据连接 AT 发送

1). AT+CGDCONT=1,"ip", "cmnet" 定义 PDP 上下文,参数 3 为 APN 类型,联通卡请设置为 3gnet

2). AT+GTRNDIS=1,<cid> 命令,模块自动进行 PDP 激活,并将获取的 IP 地址分配给客户端设备。(注:<cid>即发送+CGDCONT 命令设置 PDP 时选择的 cid)

3. IP 及 DNS 地址查询

1). AT+GTRNDIS? 命令可以查询当前数据连接状态，命令返回格式：

+GTRNDIS: <state>[,<cid>,<ip>,<pdns>,<sdns>]

参数说明：

<state>: 0, 去激活状态

1, 激活状态

<cid> : PDP 激活使用的 cid

<ip> : PDP 激活获取的 IP 地址 (该 IP 地址也是分配给客户端设备的 IP)

<pdns>: PDP 激活时获取的首选 DNS

<sdns>: PDP 激活时获取的备用 DNS

注：处于激活状态，即<state>值为 1 时，+GTRNDIS 命令查询结果才会包含<cid> , <ip> , <pdns> , <sdns>这几个参数。

4. 数据连接断开

1). AT+GTRNDIS=0,<cid>命令去激活；

2). AT+CGACT=0,<cid>命令去激活 PDP；

(注：<cid>即激活时选择的 cid)

注：

1) 步骤 3 执行成功后，执行 ifconfig 命令可查询 ECM 端口将获取到 IP 地址。

执行 ping www.baidu.com 可验证数据连接是否成功。

2)linux 下使用需要先发送 AT+GTNETCARDTYPE=1,之后重启模块，才能正常使用 ECM 功能。

+GTNETCARDTYPE 命令说明

1) 命令格式：AT+GTNETCARDTYPE=<type>

2) 参数说明：

<type>: 0, DIS

1, ECM under Linux

2, BIM

3, RNDIS under Windows

2.8 PPP 拨号上网连接

在一些无法支持 ECM 驱动的应用场景下，则需要选择使用 PPP 拨号上网。

ppp 拨号包含的脚本文件一共三个：chat-wcdma-connect、chat-wcdma-disconnect、wcdma，脚本内容如 2.10 章节所示。

1. 将上述三个脚本文件放置于/etc/ppp/peers/目录，使用 chmod 777 xxx 命令给文件赋上可读可写可执行权限。在命令行输入：

```
pppd call <拨号脚本>
```

例如拨号脚本名称为“wcdma”，则命令如下：

```
pppd call wcdma
```

2. 拨号上网成功后，执行 ifconfig 命令查询 IP 地址。

图 2-5 为 ppp 成功拨号上网后执行 ifconfig 命令的查询结果：

```
[root@wavelet peers]# ifconfig
eth0      Link encap:Ethernet  HWaddr 00:19:D1:75:1F:3A
          inet6 addr: fe80::219:d1ff:fe75:1f3a/64 Scope:Link
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:147400 errors:0 dropped:0 overruns:0 frame:0
          TX packets:29822 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:52099010 (49.6 MiB)  TX bytes:3672236 (3.5 MiB)
          Interrupt:21 Memory:dfde0000-dfe00000

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:16436  Metric:1
          RX packets:70 errors:0 dropped:0 overruns:0 frame:0
          TX packets:70 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:7024 (6.8 KiB)  TX bytes:7024 (6.8 KiB)

ppp0      Link encap:Point-to-Point Protocol
          inet addr: 172.20.19.220  P-t-P: 172.20.19.220  Mask:255.255.255.255
          UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1280  Metric:1
          RX packets:4 errors:0 dropped:0 overruns:0 frame:0
          TX packets:6 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:3
          RX bytes:58 (58.0 b)  TX bytes:108 (108.0 b)
```

图 2-5 ifconfig 查询



拨号前提

1. 模块插入有效 SIM 卡;
2. 模块上电运行;
3. 模块能注册网络。

2.9 PPP 拨号上网脚本说明

wcdma 脚本示例：

nodetach

lock

/dev/ttyUSB0

115200

crtcts

debug

#logfile /data/logfile

modem

hide-password

usepeerdns

noauth

noipdefault

novj

novjccomp

noccp

defaultroute

ipcp-accept-local

ipcp-accept-remote

connect 'chat -s -v -f /etc/ppp/peers/chat-wcdma-connect'

disconnect 'chat -s -v -f /etc/ppp/peers/chat-wcdma-disconnect'



/dev/ttyUSB0 指定拨号所用端口，如果需要使用 USB2 端口进行拨号，只需要把 ttyUSB0 修改成 ttyUSB2 即可。

chat-wcdma-connect 脚本示例：

" AT

OK "

ABORT 'NO CARRIER'

ABORT 'ERROR'

ABORT 'NO DIALTONE'

ABORT 'BUSY'

ABORT 'NO ANSWER'

" AT

OK ATZ

OK AT+GTRAT?

OK AT+CMEE=2

OK AT+CSQ

OK AT+CPIN?

OK AT+COPS?

OK AT+CGACT=0,1

OK AT+CGDCONT=1,\"IP\",\"cmnet\",,0,0

OK ATDT*99#

CONNECT "



AT+CGDCONT=1,\"IP\",\"cmnet\",,0,0 （ cmnet 为移动 APN , 联通 APN 为 3gnet ）

chat-wcdma-disconnect 脚本示例：

ABORT OK

ABORT BUSY

ABORT DELAYED

ABORT "NO ANSWER"

ABORT "NO CARRIER"

ABORT "NO DIALTONE"

ABORT VOICE

ABORT ERROR

ABORT RINGING

TIMEOUT 12

"" \K

"" \K

"" \K

"" +++ATH

"" +++ATH

"" +++ATH

"" ATZ

SAY "\nGoodbay\n"

3 Android 系统使用说明

3.1 Android 内核驱动移植配置及加载

Android 内核是基于 Linux 内核实现的，有关 Android 内核部分请参照 [2.2](#)、[2.3](#)、[2.4](#) 章节进行配置。

3.2 系统集成及调试

3.2.1 系统与模块端口的通讯

Android 系统的通信功能是通过 RIL 与模块进行 AT 命令的交互以实现的，系统应具备有与外设进行数据通讯的硬件接口，如 UART、USB 等，若系统使用的是 USB 口收发 AT 指令，则需在 Android 系统的内核中加载对应的 USB 驱动，详细加载方法请参照 [3.1](#) 章节。

RIL (Radio Interface Layer) 将用户所用到的各种功能抽象成对应的 request 和 unsolicited command，最终转换成 3GPP 标准的 AT 命令，交由 4G 模块处理，并将处理结果返回给 UI。

3.2.2 RIL 集成

1. 修改 init.rc 文件中 ril-daemon 服务：

```
#begin
```

```
service ril-daemon /system/bin/rild -l /system/lib/libreference-ril.so -- -d /dev/ttyUSB2
```

```
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
```

```
service dhcpcd_rndis /system/bin/sh /system/etc/init.dhcpcd-rndis
```

```
user root
```

```
group radio cache inet misc
```

```
disabled
```

oneshot

service rndis_down /system/bin/sh /system/etc/init.rndis-down

user root

group radio cache inet misc

disabled

#end

-d 后接的参数为实际的 usb 映射端口(AT 通讯口)。一般情况下 ttyUSB2 为 4G 模块的 AT 通讯口

2. 修改 android/hardware/ril/rild/rild.c 文件，注释 switchUser()函数调用。

```
#endif
// switchUser();

dlHandle = dlopen(rilLibPath, RTLD_NOW);
```

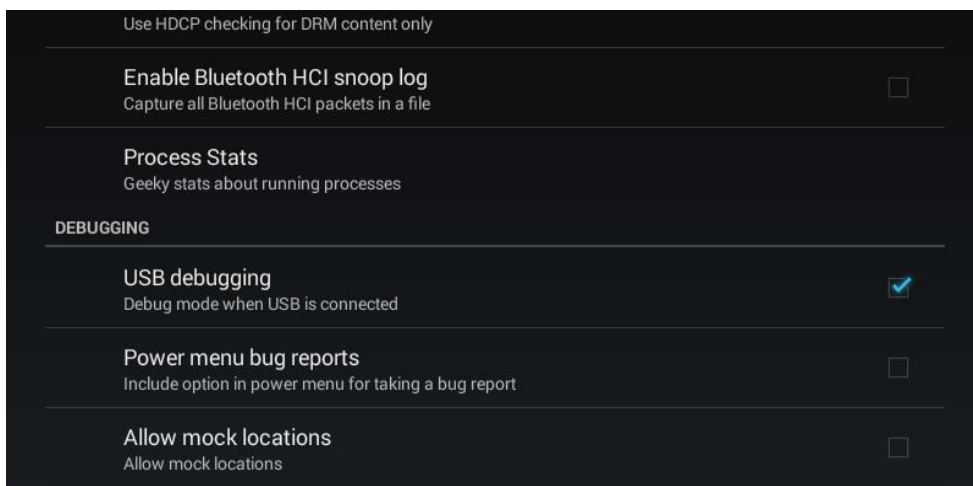
编译时把 Fibocom 提供的 libreference-ril.so 文件打包至系统镜像的/system/lib/目录，重新烧录新的系统镜像即可。

3.2.3adb 工具安装

查看 RIL 是否正常加载时，需要用 adb 工具辅助查看 RIL 运行 LOG.

1. adb 工具安装及使用说明

开机后，在 Android 系统中的设置里勾选上“开发人员选项” ->> “usb 调试” 选项，以使用 adb 调试功能。



安卓设备与 PC 连接，安装 adb 驱动（可通过常用的安卓手机助手工具进行安装）。成功安装后在设备管理中会发现一个 adb 的设备。



在 windows 系统中打开一个 cmd 窗口，进入到 adb 工具所在目录，输入 “adb shell” 命令即可进入 Android 设备的命令行终端进行各种调试工作。

3.2.4 RIL 库更换

在调试过程中经常需要更新 RIL 库，按顺序执行下述步骤即可更新 RIL 库。

1. adb devices --- 查询设备是否已经被 adb 工具识别

```
C:\Users\Administrator>adb devices
List of devices attached
0123456789ABCDEF      device
```

2. adb root --- 将 adb 终端切换到 root 用户

```
C:\Users\Administrator>adb root
adb is already running as root
```

3. adb remount --- 赋予 /system/目录可读写权限

```
C:\Users\Administrator>adb remount
remount succeeded
```

4. adb shell ls init*.rc --- 查询 android 系统有几个 init.xx.rc 文件

```
C:\Users\Administrator>adb shell ls init*.rc
init.am335xevm.rc
init.am335xevm.usb.rc
init.goldfish.rc
init.rc
init.trace.rc
init.usb.rc
```

5. adb shell cat init.xx.rc – 逐个查询 init.xx.rc 文件，直到查询到/system/bin/rild 关键词为止，通过 rild 关键词查找当前使用 ril 库的名称:libreference-ril.so (intel 平台 rild 关键词一般在 init.modem.rc 文件中)

```
service ril-daemon /system/bin/rild -l /system/lib/libreference-ril.so -- -d /dev/ttyACM2
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
```

6. adb shell push xxx\libght-ril.so /system/lib/libreference-ril.so --- ril 库替换

```
C:\Users\Administrator>adb push E:\libreference-ril.so /system/lib/libreference-ril.so
```



push 的第一个参数为电脑本地目录的 ril 库路径，第二个参数为 android 系统使用的 ril 库路径，并且第二个参数的 ril 库名字必须与 5 步骤查询到的 ril 库名称一致，否则 ril 无法正常运行。

7. adb shell stop ril-daemon --- 暂停 ril 服务

8. adb shell logcat -b radio -c --- 清空 ril log

9. adb shell start ril-daemon --- 开启 ril 服务

10. adb shell logcat -b radio -v time > D:/radio.txt 抓取 ril log，确认版本号

用 ctrl+c 终止 log 抓取，打开 D 盘根目录的 radio.txt 文件，搜索 RIL Daemon version 关键词，即可查询 RIL 版本号，以确认 RIL 库是否更新成功。

```
C:\Users\Administrator>adb shell stop ril-daemon
C:\Users\Administrator>adb shell logcat -b radio -c
C:\Users\Administrator>adb shell start ril-daemon
C:\Users\Administrator>adb shell logcat -b radio -v time > D:/radio.txt
^C
C:\Users\Administrator>
```

3.2.4.1 RIL 加载判断

打开 adb shell，输入 logcat -b radio，通过输出的 log 可查看 AT 命令交互情况，下图为 RIL 初始化正常的 log 示例：

```
D/AT      < 1484>: AT> ATE0Q0U1
D/AT      < 1484>: AT< ATE0Q0U1
D/AT      < 1484>: AT< OK
D/AT      < 1484>: AT> ATE0Q0U1
D/AT      < 1484>: AT< OK
D/AT      < 1484>: AT> ATSD=0
D/AT      < 1484>: AT< OK
D/AT      < 1484>: AT> AT+CMEE=1
D/AT      < 1484>: AT< OK
D/AT      < 1484>: AT> AT+CREG=2
D/AT      < 1484>: AT< OK
D/AT      < 1484>: AT> AT+CGREG=1
D/AT      < 1484>: AT< OK
D/AT      < 1484>: AT> AT+CCWA=1
D/AT      < 1484>: AT< OK
D/AT      < 1484>: AT> AT+CMOD=0
```

3.2.4.2 RIL 版本号检查

打开 adb shell, 输入 logcat -b radio, 通过输出的 log 可查看 RIL 版本号 搜索关键词“RIL Daemon version”, 其中 RIL_V4x.00.02 为 RIL 版本号。

```
I/RILJ < 505>: Connected to 'rild' socket
I/RILC < 1264>: libril: new connection
I/RILC < 1264>: RIL Daemon version: RIL_V4x.00.02
D/RILJ < 505>: [UNSLI< UNSOL_RIL_CONNECTED <?>
D/RILJ < 505>: [0001]> RADIO_POWER off
D/RIL < 1264>: onRequest: RADIO_POWER
D/RIL < 1264>: mly1:onRequest: RADIO_POWER
D/RIL < 1264>: mly2:onRequest: RADIO_POWER
```

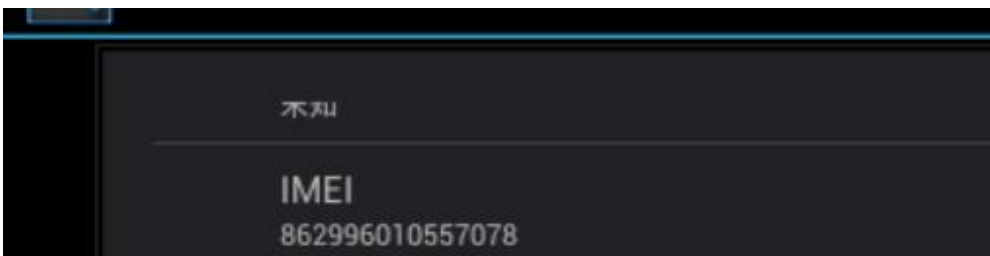
3.2.5 调试信号的显示、电话程序、短信等功能

3.2.5.1 模块 IMEI 号查询

在初始化完成后, 上层应用会查询模块的 IMEI 号, 下述为 RIL log 中 IMEI 号正常查询结果。

```
D/RIL < 1264>: onRequest: GET_IMEI
D/RIL < 1264>: mly1:onRequest: GET_IMEI
D/RIL < 1264>: mly2:onRequest: 38
D/RIL < 1264>: mly3:sState: 4
D/AT < 1264>: AT> AT+CGSN?
```

进入“设置”->“平板信息”->“状态信息”, 可查询 IMEI 号信息。若为“Unkown”, 表示 RIL 与模块交互异常。



3.2.5.2 信号强度查询

RIL 运行成功后, 上层会定时查询信号强度, 下述为 RIL log 中信号强度正常查询结果

```
D/RIL < 1505>: onRequest: SIGNAL_STRENGTH
D/RIL < 1505>: mly1:onRequest: SIGNAL_STRENGTH
D/RIL < 1505>: mly2:onRequest: 19
D/RIL < 1505>: mly3:sState: 2
D/AT < 1505>: AT> AT+CSQ
D/AT < 1505>: AT< +CSQ: 13,99
D/AT < 1505>: AT< OK
D/RILJ < 505>: [0097]< SIGNAL_STRENGTH SignalStrength: 13
```

界面上的信号强度显示：



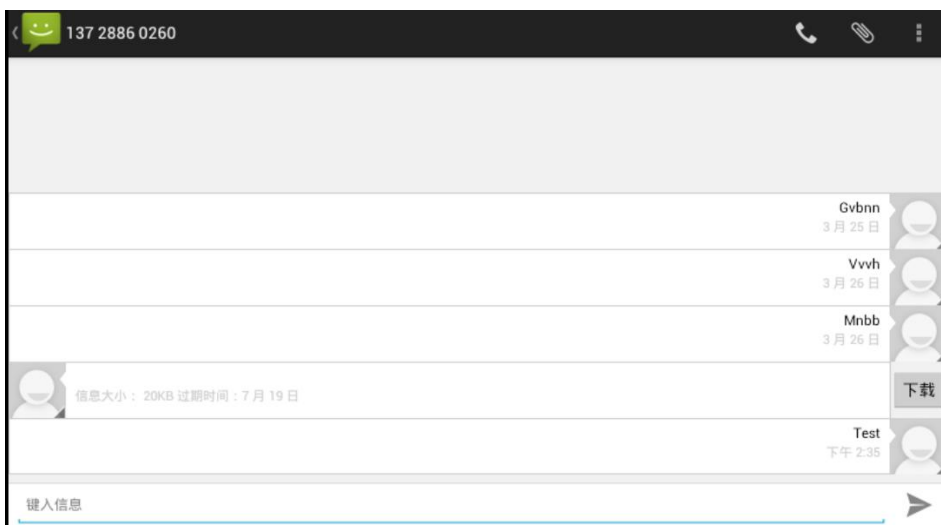
3.2.5.3语音及短信检测

上述的调试、测试正常后,可进行语音拨打、短信收发。在进行相应的操作时可以通过 adb 的 logcat -b radio 查看是否有相应的命令发送。

拨打 10086 , RIL 信息显示如下：

```
D/use-Rlog/RLOG-AT< 227>: AT> AT+CMUT=0
D/use-Rlog/RLOG-AT< 227>: AT< OK
D/use-Rlog/RLOG-RIL< 227>: onRequest: DIAL
D/use-Rlog/RLOG-AT< 227>: AT> ATD10086;
D/RILJ < 797>: [4989]< SET_MUTE
D/use-Rlog/RLOG-AT< 227>: AT< +STKCTRLIND: 0,0,, "10086",129
D/use-Rlog/RLOG-AT< 227>: AT< OK
D/use-Rlog/RLOG-AT< 227>: AT< +CLCC: 1,0,2,0,0, "10086",129
```

在短信程序中发送 “test” 给对方：

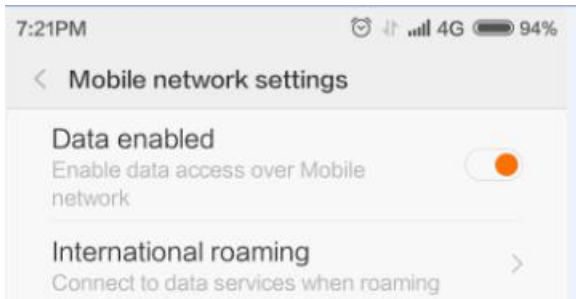


RIL 的相对应的 log：

```
D/RILJ < 2033>: [0058]< SEND_SMS
D/RIL < 1484>: onRequest: SEND_SMS
D/AT < 1484>: AT> AT+CMGS=17
D/AT < 1484>: AT< >
D/AT < 1484>: AT> 0001000b813127880662f00000004d4f29c0e^Z
D/AT < 1484>: AT< +CMGS: 24
D/AT < 1484>: AT< OK
D/RILJ < 2033>: [0058]< SEND_SMS { messageRef = 0, errorCode
11}
```

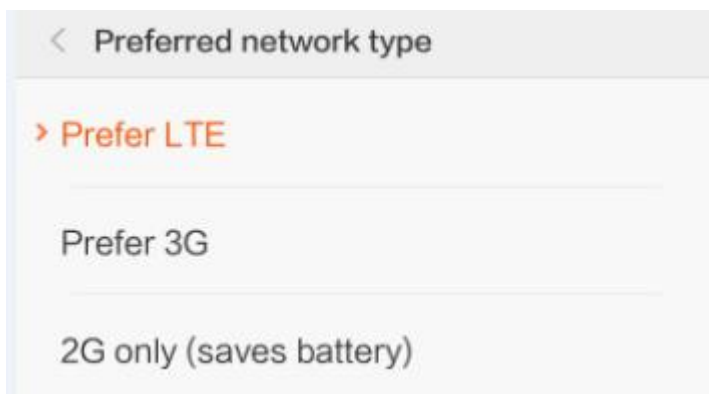
3.2.5.4数据业务开启

进入“设置”->“移动网络”->“启用数据网络”，点击启用后，平板信号格左上角将显示当前网络类型的图标：



3.2.5.5网络优先

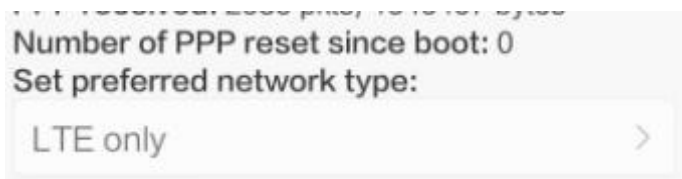
进入“设置”->“移动网络”->“网络优先”，点击对应的菜单，将切换至对应的网络



若没有 4G 优先的菜单，则需客户自行修改上层 UI 添加 4G 优先菜单。

没有 4G 优先菜单可以尝试进入工程模式进行网络切换：

打开 Phone 应用 -> 输入 `***#4636***` -> 选择“Tablet information”-> 下拉 Set preferred network type 菜单，选择 LTE only 后将切换至 4G 网络。此方法仅为测试使用，最终解决的方法仍需在上层 UI 添加 4G 优先菜单。



3.2.5.6调试音频通道切换及音量调节

Android 工程师结合相对应的平台，在电话服务层、HAL 层或驱动层做相应的添加处理，可以通过提供 AT



命令配合调试实现。