

**L8 Family 系统驱动集成及应用指导**

文档版本：V2.0.1

更新日期：2015.12.23

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 | 2015-10-17 | 初始版本 |
| V2.0.0 | 2015-10-23 | 1. 优化文档名称及标题 2. 优化文档结构及内容 3. 优化文档附图 |
| V2.0.1 | 2015-12-23 | 1. 2.7.2节增加CID2说明 2. 删除2.7.2节NCM上网DHCP相关内容 |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

适用型号

|  |  |  |
| --- | --- | --- |
| 序号 | 产品型号 | 说明 |
| 1 | L810-GL-00 |  |
| 2 | L810-GL-01 |  |
| 3 | L830-GL-00 |  |
| 4 | L830-GL-01 |  |
| 5 | L830-EA-00 |  |
| 6 | L831-EA-00 |  |
| 7 | L831-EA-01 |  |

**目录**

[1 简介 5](#_Toc433809222)

[1.1 目的 5](#_Toc433809223)

[1.2 范围 6](#_Toc433809224)

[2 Linux系统使用说明 7](#_Toc433809225)

[2.1 Linux 内核设备驱动架构 7](#_Toc433809226)

[2.2 Linux NCM驱动集成 8](#_Toc433809227)

[2.2.1 NCM驱动代码移植 8](#_Toc433809228)

[2.2.1.1 linux 2.6.38及以上的内核集成 8](#_Toc433809229)

[2.2.1.2 linux2.6.32至38内核集成 8](#_Toc433809230)

[2.2.1.3 linux2.6.26内核集成 8](#_Toc433809231)

[2.2.1.4 linux2.6.22内核集成 9](#_Toc433809232)

[2.2.1.5 低于linux2.6.22的内核 10](#_Toc433809233)

[2.2.2 NCM驱动配置 11](#_Toc433809234)

[2.3 Linux ACM驱动集成 12](#_Toc433809235)

[2.3.1 ACM驱动代码移植 12](#_Toc433809236)

[2.3.2 详细配置步骤 13](#_Toc433809237)

[2.4 NCM/ACM驱动配置确认 14](#_Toc433809238)

[2.5 端口形态说明 15](#_Toc433809239)

[2.6 端口测试 15](#_Toc433809240)

[2.6.1 命令行测试 15](#_Toc433809241)

[2.6.2 程序测试 16](#_Toc433809242)

[2.7 NCM上网AT发送 18](#_Toc433809243)

[2.7.1 查询信号、 SIM卡及注网状态 18](#_Toc433809244)

[2.7.2 拨号AT发送 19](#_Toc433809245)

[2.7.3 USBCDC与ACM端口对应列表 19](#_Toc433809246)

[2.7.4 IP及DNS地址查询 20](#_Toc433809247)

[2.8 NCM上网网络配置 20](#_Toc433809248)

[2.8.1 DNS地址配置 20](#_Toc433809249)

[2.8.2 IP地址及路由配置 20](#_Toc433809250)

[2.9 PPP拨号上网连接 21](#_Toc433809251)

[2.10 PPP拨号上网脚本说明 22](#_Toc433809252)

[3 Android系统使用说明 25](#_Toc433809253)

[3.1 Android内核驱动移植配置及加载 25](#_Toc433809254)

[3.2 系统集成及调试 25](#_Toc433809255)

[3.2.1 系统与模块端口的通讯 25](#_Toc433809256)

[3.2.2 RIL集成 25](#_Toc433809257)

[3.2.3 adb 工具安装 26](#_Toc433809258)

[3.2.4 RIL库更换 26](#_Toc433809259)

[3.2.4.1 RIL加载判断 28](#_Toc433809260)

[3.2.4.2 RIL版本号检查 28](#_Toc433809261)

[3.2.5 调试信号的显示、电话程序、短信等功能 28](#_Toc433809262)

[3.2.5.1 模块IMEI号查询 28](#_Toc433809263)

[3.2.5.2 信号强度查询 29](#_Toc433809264)

[3.2.5.3 语音及短信检测 29](#_Toc433809265)

[3.2.5.4 数据业务开启 30](#_Toc433809266)

[3.2.5.5 网络优先 30](#_Toc433809267)

[3.2.5.6 调试音频通道切换及音量调节 31](#_Toc433809268)

[4 Win8.1/Win10系统使用说明 32](#_Toc433809269)

[4.1 简介 32](#_Toc433809270)

[4.2 应用说明 32](#_Toc433809271)

[4.2.1 MBIM驱动加载 32](#_Toc433809272)

[4.2.2 应用步骤 32](#_Toc433809273)

[4.2.2.1 数据业务 32](#_Toc433809274)

[4.2.2.2 APN设置 33](#_Toc433809275)

[4.2.2.3 SIM PIN设置 34](#_Toc433809276)

# 简介

## 目的

* 本文主要是针对L8系列4G模块设备基于Android/Linux/Win8.1/Win10 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

## 范围

本文适用于：

* Win8.1/Win10
* Android 4.0及以上版本
* Linux2.6.22以上版本

# Linux系统使用说明

## Linux 内核设备驱动架构

L8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：

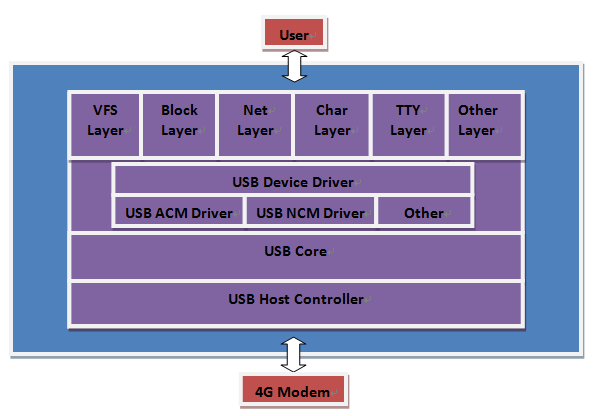


图2-1驱动架构

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 ACM驱动模块及USB NCM 驱动模块。

ACM驱动：USB ACM驱动，支持如modem 端口，AT 端口等；该驱动模块



的代码（cdc-acm.c）已经内置于Linux 内核的源码。

NCM 驱动：USB标准的NCM 网络设备驱动，主要用于传输网络数据。

## Linux NCM驱动集成

### NCM驱动代码移植

#### linux 2.6.38及以上的内核集成

针对linux2.6.38及以上内核版本，请直接按照[2.2.2](#_NCM驱动配置)章节添加配置

#### linux2.6.32至38内核集成

请参照下述步骤进行集成：

1. 登陆<https://www.kernel.org/pub/linux/kernel/v2.6/>网址，下载2.6.38以上的内核源码
2. 解压下载的内核源码，将以下文件拷贝至对应的目录

drivers/net/usb/cdc\_ncm.c

include/linux/usb/cdc.h

include/linux/atomic.h

1. 修改自身源码drivers/net/usb/目录的Makefile及Kconfig文件

将obj-$(CONFIG\_USB\_NET\_CDC\_NCM) += cdc\_ncm.o添加至Makefile的末尾。

将下述语句添加至Kconfig文件末尾。

config USB\_NET\_CDC\_NCM

tristate "CDC NCM support"

depends on USB\_USBNET

default y

1. 修改include/linux/usbnet.h

将#define FLAG\_MULTI\_PACKET 0x2000 以及 int (\*manage\_power)(struct usbnet \*, int);添加至文件中即可，添加位置可参照2.6.38的usbnet.h.

#### linux2.6.26内核集成

请参照下述步骤进行集成：

1. 登陆<https://www.kernel.org/pub/linux/kernel/v2.6/>网址，下载2.6.38的内核源码
2. 解压下载的内核源码，将以下文件拷贝至对应的目录

drivers/net/usb/cdc\_ncm.c

include/linux/usb/cdc.h

include/linux/atomic.h

1. 修改drivers/net/usb/usbnet.c

删除usbnet\_start\_xmit函数的static声明：

int usbnet\_start\_xmit (struct sk\_buff \*skb, struct net\_device \*net)

并添加下述语句：

EXPORT\_SYMBOL\_GPL(usbnet\_start\_xmit);

1. 修改自身源码drivers/net/usb/目录的Makefile及Kconfig文件

将obj-$(CONFIG\_USB\_NET\_CDC\_NCM) += cdc\_ncm.o添加至Makefile的末尾。

将下述语句添加至Kconfig文件末尾。

config USB\_NET\_CDC\_NCM

tristate "CDC NCM support"

depends on USB\_USBNET

default y

1. 修改include/linux/usbnet.h

将#define FLAG\_MULTI\_PACKET 0x1000 以及 int (\*manage\_power)(struct usbnet \*, int);添加至文件中即可，添加位置可参照2.6.38的usbnet.h.

再添加下述语句：

extern int usbnet\_start\_xmit(struct sk\_buff \*skb,

struct net\_device \*net);

#### linux2.6.22内核集成

请参照下述步骤进行集成：

1. 登陆<https://www.kernel.org/pub/linux/kernel/v2.6/>网址，下载2.6.38及2.6.26的内核源码
2. 解压下载的2.6.26内核源码，将以下文件拷贝到对应的目录

drivers/net/usb/usbnet.c

include/linux/usb/usbnet.h

并按照下述步骤修改usbnet.c

* 1. 搜索关键词”print\_mac”,并注释掉包含printf\_mac的打印语句
  2. 注释DECLARE\_MAC\_BUF(mac);函数调用
  3. 删除usbnet\_start\_xmit函数的static声明：

int usbnet\_start\_xmit (struct sk\_buff \*skb, struct net\_device \*net)

并添加下述语句：

EXPORT\_SYMBOL\_GPL(usbnet\_start\_xmit);

1. 解压下载的2.6.38内核源码，将以下文件拷贝至对应的目录

drivers/net/usb/cdc\_ncm.c

include/linux/usb/cdc.h

include/linux/atomic.h

1. 修改自身源码drivers/net/usb/目录的Makefile及Kconfig文件

将obj-$(CONFIG\_USB\_NET\_CDC\_NCM) += cdc\_ncm.o添加至Makefile的末尾。

将下述语句添加至Kconfig文件末尾。

config USB\_NET\_CDC\_NCM

tristate "CDC NCM support"

depends on USB\_USBNET

default y

1. 修改include/linux/usbnet.h

将#define FLAG\_MULTI\_PACKET 0x1000 以及 int (\*manage\_power)(struct usbnet \*, int);添加至文件中即可，添加位置可参照2.6.38的usbnet.h.

再添加下述语句：

extern int usbnet\_start\_xmit(struct sk\_buff \*skb,

struct net\_device \*net);

#### 低于linux2.6.22的内核

针对低于2.6.22的Linux内核版本，可参照[2.2.1.4](#_linux2.6.22内核集成)节进行的集成，但集成时会存在兼容性问题，可联系广和通工程师一起联调。

### NCM驱动配置

**步骤1：**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\_USBNET=y

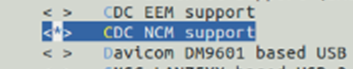
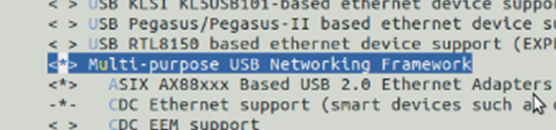
CONFIG\_NETDEVICES=y

CONFIG\_USB\_NET\_CDC\_NCM=y

**步骤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 NCM support选项:



1. 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。
2. 完成配置后，运行make命令，编译修改后的内核。

## Linux ACM驱动集成

### ACM驱动代码移植

1. 驱动代码修改：在drivers/usb/class/cdc-acm.c文件的static const struct usb\_device\_id acm\_ids[ ] 数组内图2-2添加红框所示代码：

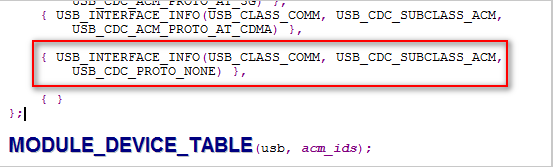


图2-2 acm\_ids代码

图2-2代码如下：

{ USB\_INTERFACE\_INFO(USB\_CLASS\_COMM, USB\_CDC\_SUBCLASS\_ACM,

USB\_CDC\_PROTO\_NONE) },

1. 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 ：
2. 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

1. USB ACM相关配置项：

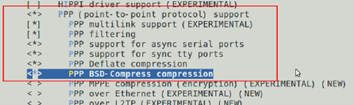
CONFIG\_USB\_ANNOUNCE\_NEW\_DEVICES=y (若此选项存在，建议配置，若不存在，请忽略)

CONFIG\_USB\_ACM=y

### 详细配置步骤

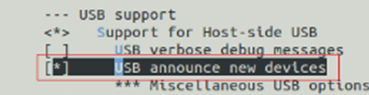
1. 打开Terminal工具，进入kernel目录（假定为“/home/ght /linux-3.0.8/”)，执行make <configuration>命令（假定使用标准make menuconfig）。
2. 按下述指引完成PPP拨号相关配置：

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



1. 按下述指引完成 ACM驱动相关配置**：**

进入“**Device Drivers**“→“**USB support**” 菜单后选择 USB announce new devices及USB Modem(CDC ACM) support选项：



1. 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>”并退出出。
2. 完成配置后，运行make命令，编译修改后的内核。

## NCM/ACM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，查找图2-3红框信息即说明系统已经成功配置NCM驱动。

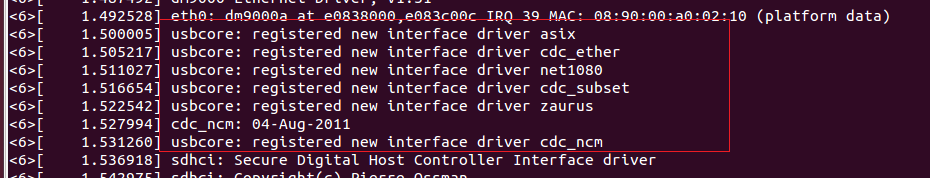


图2-3 NCM配置

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NCM驱动已经成功加载，并且生成usb0、 usb1、 usb2等NCM网络设备节点，在linux-3.10及以上内核， NCM网络设备节点将改变成 wwan0、 wwan1、 wwan2等。在驱动加载时可以确认端口节点，如图2-4红色方框所示，如果为usb, 那么NCM网络设备节点为usb0-usbx, 如果为wwan,那么NCM网络设备节点为wwan0 – wwanx。

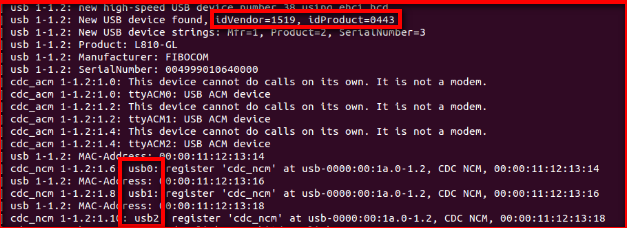
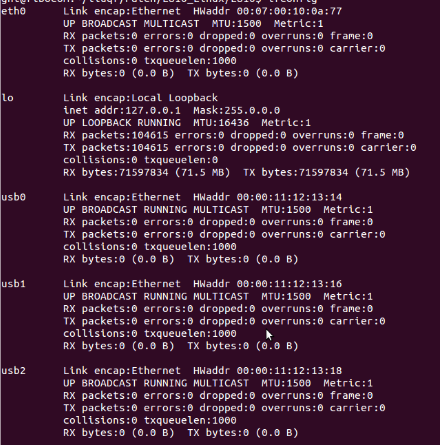


图2-4驱动加载

执行netcfg命令可以查询到有usb0、usb1、usb2等NCM网络设备节点。



执行 ls –al /dev/ttyACM\*命令可以查询到ttyACM0、ttyACM1、ttyACM2。

## 端口形态说明

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 端口名称 | 端口形态 | 备注 |
| 1 | ttyACM0 | 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 | ttyACM1 | Trace Port | 用于抓取模块调试信息 |
| 3 | ttyACM2 | At Port | 用于AT通讯，即收发AT命令 |
| 4 | usb0(wwan0)usb1(wwan1)  usb2(wwan2)usb3(wwan3) | NCM Port | USB 的标准NCM 网络设备驱动，主要用于传输网络数据。  映射的节点一般为3个，请以实际映射节点数为准。 |

## 端口测试

### 命令行测试

1. 打开terminal终端
2. 执行echo -e "ate0\r\n" > /dev/ttyACM2 **（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/ttyACM2 查询软件版本
4. 执行 cat /dev/ttyACM2 读取查询返回
5. 执行echo -e "at+cgdcont=1,\"ip\",\"3gnet\"\r\n" > /dev/ttyACM2 设置APN
6. 执行 cat /dev/ttyACM2读取设置返回结果

**若AT命令中包含双引号须加转义字符 “\”，格式参照步骤5。**



### 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/ttyACM2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/ttyACM2"

#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\_lflag &= ~(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。



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。



## NCM上网AT发送

### 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；

### 拨号AT发送

在信号、SIM卡状态以及运营商注网都正常的情况下，需要发送拨号AT指令（AT端口请选择ttyACM2)。

若模块注册上3G、2G网络请从第3步骤开始执行，若注册上4G网络，请按顺序执行

1） AT+CGDCONT? 查询返回值的第4个参数的IP地址是否为空，如果为空请按顺序执行，若存在IP地址则直接跳至第5步骤。

2） 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

3） AT+XDNS=1,1使用DNS地址查询命令

4） AT+MGAUTH=1,1,”cmnet”,”1234” 设置用户名密码，参数2为鉴权类型，1表示PAP，2表示CHAP，0表示NONE，参数3、4分别为用户名及密码。

5） AT+CGACT=1,1 激活PDP上下文

6） AT+XDATACHANNEL=1,1, "/USBCDC/2", "/USBHS/NCM/0",0 配置数据通道，参数3为AT通道，参数4为数据通道

7） AT+CGDATA="M-RAW\_IP",1 启动数据状态。至此，拨号AT发送完毕。

AT+CGACT命令的返回需要较长时间，用户名密码在不需要设置的情况下请跳过第4步骤。上述标红的参数为CID，默认是使用CID1，若需要使用CID2,请依次将标红部分参数修改为2即可，依此类推。



### USBCDC与ACM端口对应列表

|  |  |  |  |
| --- | --- | --- | --- |
| 序号 | 端口 | 描述符 | 备注 |
| 1 | ttyACM0 | /USBCDC/0 | 此表为[2.7.2](#_拨号AT发送)节第6步骤中XDATACHANNEL的参数3、参数4与驱动设备节点对应列表，若选用ttyACM0下发[2.7.2](#_拨号AT发送)节所有AT，则第6步骤中参数3需修改为/USBCDC/0 |
| 2 | ttyACM1 | /USBCDC/1 |
| 3 | ttyACM2 | /USBCDC/2 |
| 4 | usb0(wwan0) | /USBHS/NCM/0 |
| 5 | usb1(wwan1) | /USBHS/NCM/1 |
| 6 | usb2(wwan2) | /USBHS/NCM/2 |

### IP及DNS地址查询

AT+CGPADDR=1参数2将返回ipv4地址，如果返回0.0.0.0表示步骤2拨号失败，需要重新进行步骤2.

AT+CGPADDR=1

+CGPADDR: 1,”10.47.67.169”

OK

AT+XDNS? 参数2跟参数3分别为首选及次选DNS地址。

AT+XDNS？

+XDNS:1,”221,179,38,7”,”120.196.165.7”

OK

## NCM上网网络配置

### DNS地址配置

红色字体为根据AT+XDNS?查询到的DNS地址，根据实际查询结果写入/etc/resolv.conf文件中，格式如下：

nameserver 221.179.38.7

nameserver 120.196.165.7

### IP地址及路由配置

根据AT+CGDCONT?查询到的IP地址，设置主机IP及路由，其中$1为IP地址，$2为网关地址（网关地址可以跟IP地址一样，亦可将IP地址的最后一个字节设置为1）。

ifconfig usb0 $1 netmask 255.255.255.255 -arp

ip r add $2 devusb0

ip r add 0.0.0.0/0 via $2 dev usb0

（ 若NCM网络设备节点名称为wwan，请将红色usb0替换成wwan0



设置完后请用 ping [www.baidu.com](http://www.baidu.com/) 查询主机网络是否能ping通。

## PPP拨号上网连接

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

ppp拔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 <拔号脚本>

例如拔号脚本名称为“wcdma”，则命令如下：

pppd call wcdma

1. 拔号上网成功后，执行ifconfig命令查询IP 地址。

图2-5为ppp 成功拔号上网后执行ifconfig命令的查询结果：

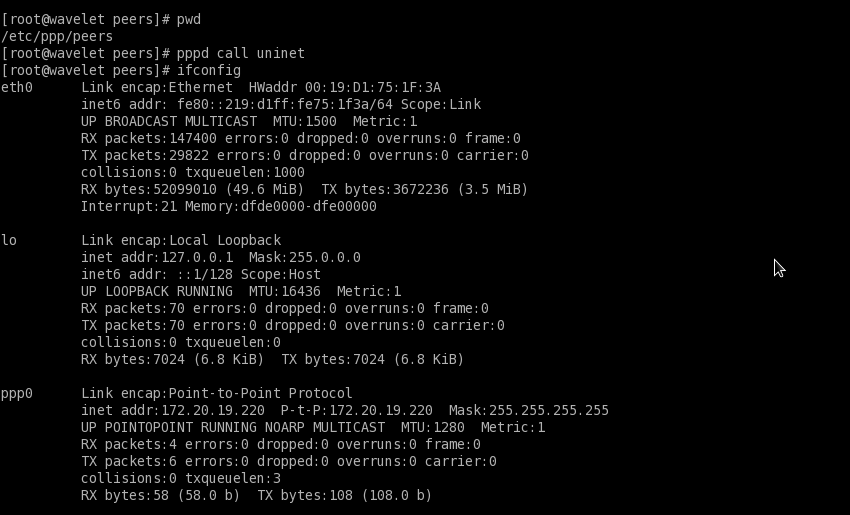


图2-5 ifconfig查询

注：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。

## PPP拨号上网脚本说明

wcdma脚本示例：

nodetach

lock

/dev/ttyACM0

115200

crtscts

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'

注：/dev/ttyACM0 指定拨号所用端口，如果需要使用ACM2端口进行拔号， 只需要把ttyACM0修改成ttyACM2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 ''

注：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"

# Android系统使用说明

## 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](#_Linux_NCM驱动集成)、[2.3](#_Linux_ACM驱动集成)、[2.4](#_NCM/ACM驱动配置确认)章节进行配置。

## 系统集成及调试

### 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](#_Android内核驱动移植配置及加载)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。

### 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/ttyACM2

class main

socket rild stream 660 root radio

socket rild-debug stream 660 radio system

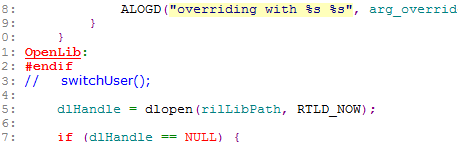
user root

group radio cache inet misc audio sdcard\_rw log

#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ttyACM2为4G模块的AT通讯口

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



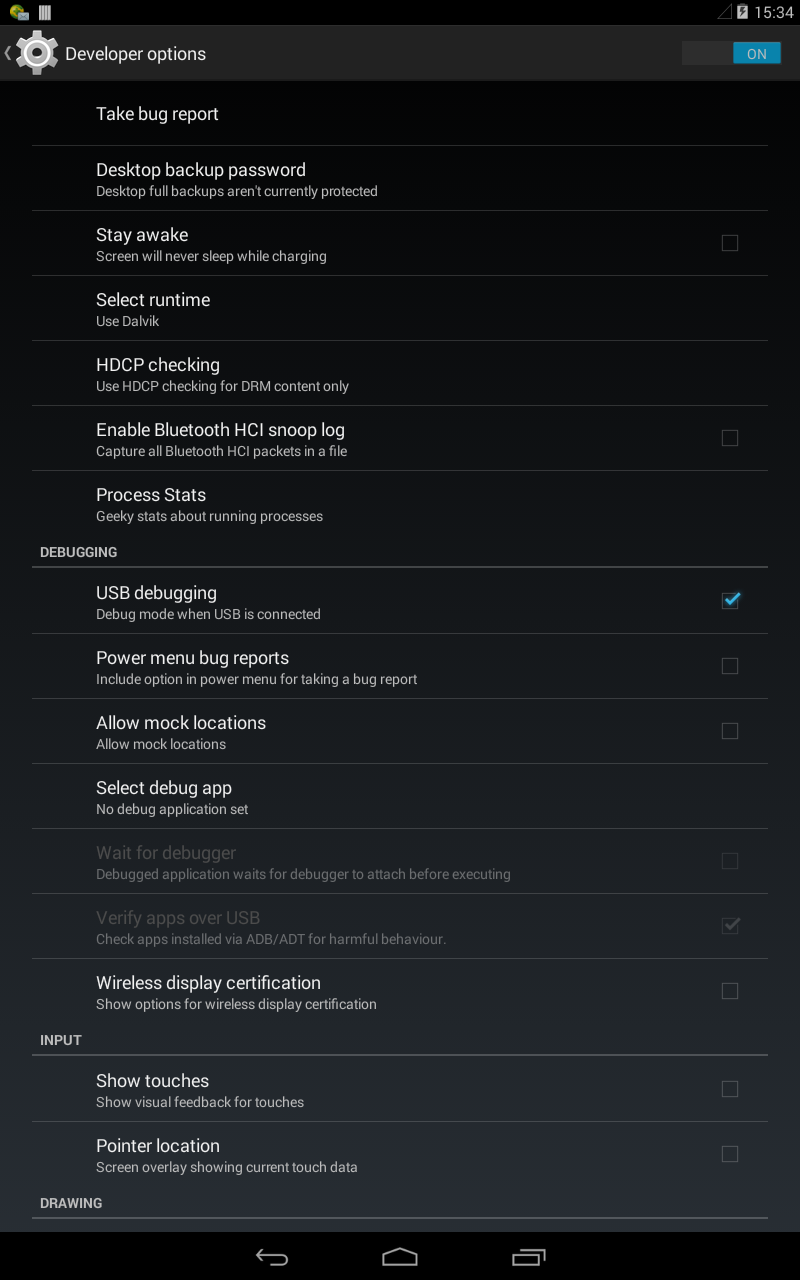
编译时把Fibocom提供的libreference-ril.so文件打包至系统镜像的/system/lib/目录，重新烧录新的系统镜像即可。

### adb 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。

### 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。

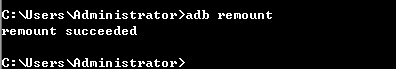
**1. adb devices --- 查询设备是否已经被adb工具识别**



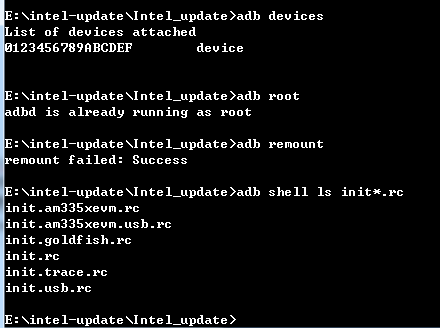
**2. adb root --- 将adb终端切换成root用户**



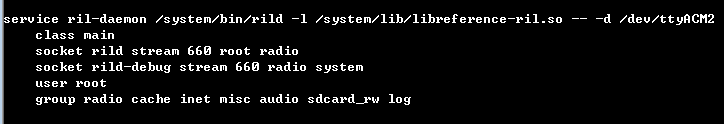
**3. adb remount --- 赋予 /system/目录可读写权限**



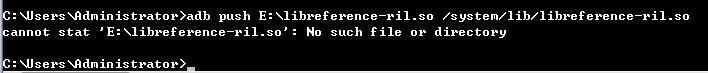
**4. adb shell ls init\*.rc --- 查询android系统有几个init.xx.rc文件**



**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)**



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



**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。**

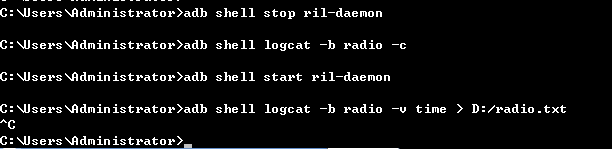
**7. adb shell stop ril-daemon --- 暂停ril服务**

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

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

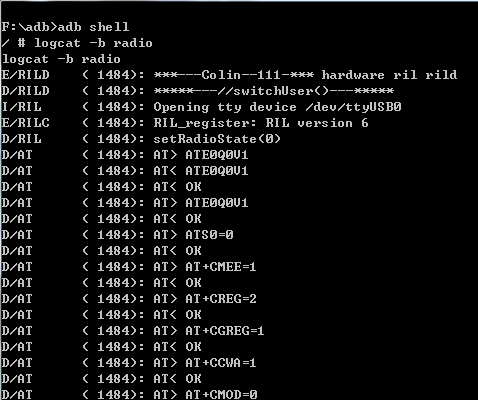
**10. adb shell logcat –b radio –v time > D:/radio.txt –抓取ril log，确认版本号**

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。



#### 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：



#### RIL版本号检查

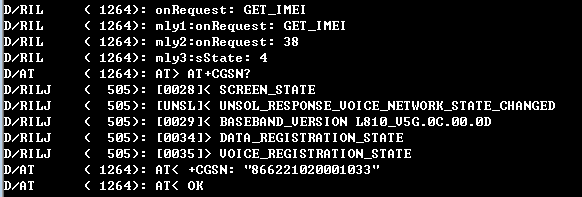
打开adb shell, 输入logcat –b radio, 通过输出的log可查看RIL版本号，搜索关键词“RIL Daemon version“ ，其中RIL\_V4x.00.02为RIL版本号。



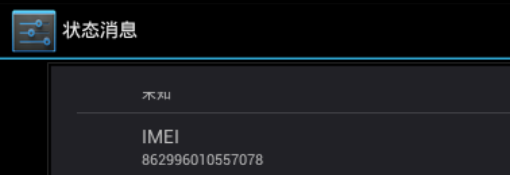
### 调试信号的显示、电话程序、短信等功能

#### 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。

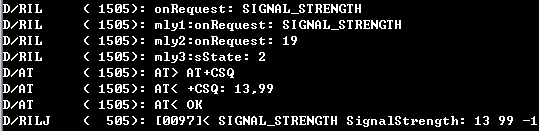


进入“设置”->“平板信息”->“状态信息”，可查询IMEI号信息。若为”Unkown”，表示RIL与模块交互异常。



#### 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



界面上的信号强度显示：



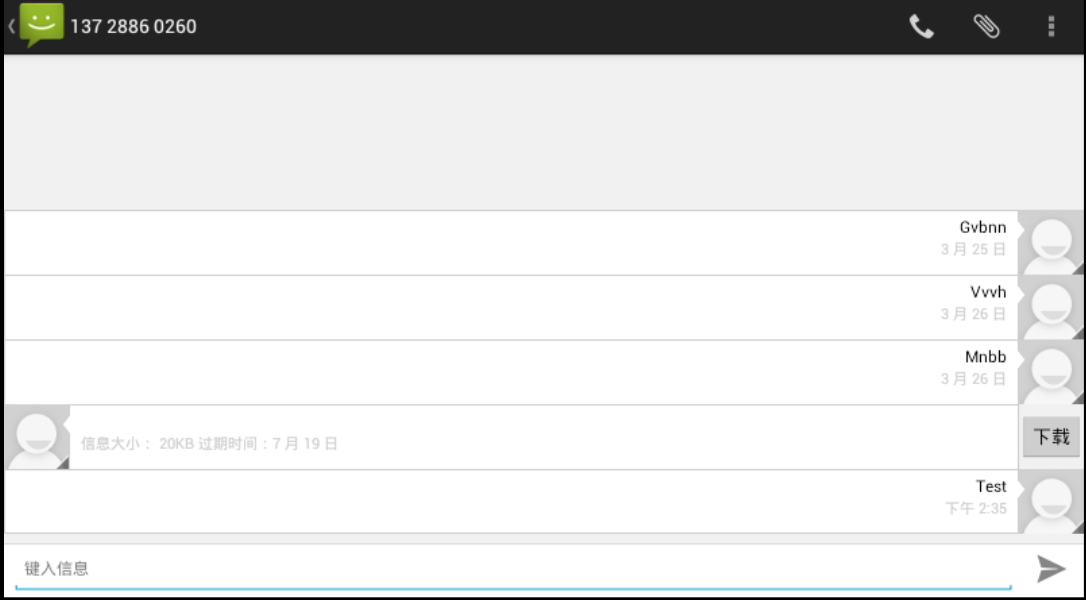
#### 语音及短信检测

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。

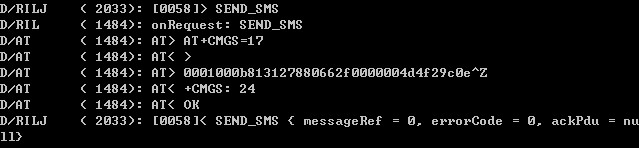
拨打10086，RIL信息显示如下：



**在短信程序中发送“test”给对方：**

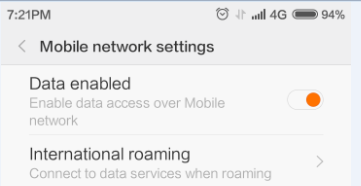


**RIL的相对应的log：**



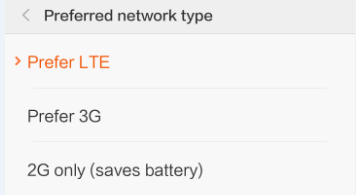
#### 数据业务开启

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：



#### 网络优先

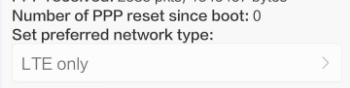
进入“设置”->“移动网络”->“网络优先”，点击对应的菜单，将切换至对应的网络



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。



#### 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。

# Win8.1/Win10系统使用说明

4G模块支持Win8.1及Win10，在Win8.1与Win10上的应用类似，下文以Win10以例进行操作说明。

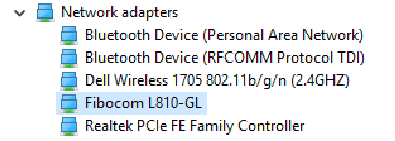
## 简介

4G模块在Win10系统上将映射出MBIM（Mobile Broadband Interface Model）端口，MBIM是伴随着移动宽带在notebook/Ultrabook, Tablet, Pad等移动设备广泛应用，由Intel, Microsoft等多家USB/IF 成员提出的一个新的接口标准。 具体标准可以在USB IF的官方网站下载。 它统一了移动宽带设备(USB 数据卡/上网卡，NGFF数据卡等) 和 PC端的接口标准。 **Modem厂商不用再提供驱动，目前Win8.1、Win10上已经内置支持。**

## 应用说明

### MBIM驱动加载

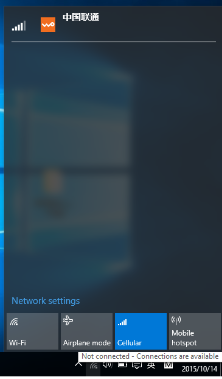
MBIM驱动是Win8.1及Win10系统自带，插入4G模块后将自动识别，在设备管理器的网络适配器层查看，显示Fibocom xxxx，即表示设备加载成功，示例如下：



### 应用步骤

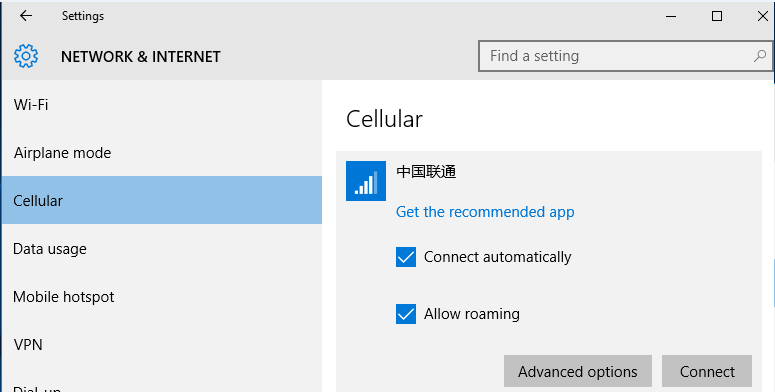
#### 数据业务

单击右下角信号图标显示所有的移动网络连接情况，单击



进入移动网络设置菜单选项，如图4-1所示：

图4-1 移动网络菜单

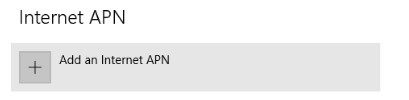


单击Connect按钮，连接数据，成功连接后将所示Connected，之后可浏览网页等。

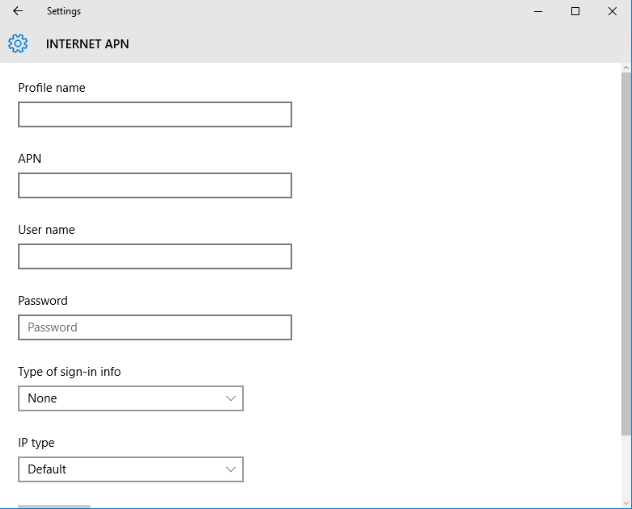


#### APN设置

单击：Advanced options选项，查找Internet APN选项，单击Add an Internet APN,



填写ProfileName及APN单击Save即可。根据实际情况填写UserName及Password。



#### SIM PIN设置

单击：Advanced options选项，查找Use SIM PIN选项，在弹出的菜单输入pin码，单击OK即可：

