



SIM7500_SIM7600系列 _LINUX NDIS驱动调试文档

LTE 模组

芯讯通无线科技(上海)有限公司
上海市长宁区金钟路633号晨讯科技大楼B座6楼
电话：86-21-31575100
技术支持邮箱：support@simcom.com
官网：www.simcom.com

名称:	SIM7500_SIM7600系列_Linux NDIS驱动调试文档
版本:	2.00
类别:	应用文档
状态:	已发布

版权声明

本手册包含芯讯通无线科技（上海）有限公司（简称：芯讯通）的技术信息。除非经芯讯通书面许可，任何单位和个人不得擅自摘抄、复制本手册内容的部分或全部，并不得以任何形式传播，违反者将被追究法律责任。对技术信息涉及的专利、实用新型或者外观设计等知识产权，芯讯通保留一切权利。芯讯通有权在不通知的情况下随时更新本手册的具体内容。

本手册版权属于芯讯通，任何人未经我公司书面同意进行复制、引用或者修改本手册都将承担法律责任。

芯讯通无线科技(上海)有限公司

上海市长宁区金钟路 633 号晨讯科技大楼 B 座 6 楼

电话：86-21-31575100

邮箱：simcom@simcom.com

官网：www.simcom.com

了解更多资料，请点击以下链接：

<http://cn.simcom.com/download/list-230-cn.html>

技术支持，请点击以下链接：

<http://cn.simcom.com/ask/index-cn.html> 或发送邮件至 support@simcom.com

版权所有 © 芯讯通无线科技(上海)有限公司 2020，保留一切权利。

关于文档

版本历史

版本	日期	作者	备注
2.00	2020.8.6		更新文档格式

SIMCom
Confidential

目录

版权声明.....	2
关于文档.....	3
版本历史.....	3
目录.....	4
1 . 本文目的.....	5
2 .SIM7500_SIM7600 USB 相关描述.....	6
3 .USB 串口驱动使用.....	7
3.1 USB Serial 的内核配置支持.....	7
3.2 修改驱动代码增加 SIM7500_SIM7600 的 VID/PID 并且跳过 Interface 5.....	7
4 . USB NDIS NET 使用.....	9
4.1 内核配置支持.....	9
4.2 使用 ifconfig 查看网卡信息，默认处于 down 状态.....	10
4.3 SIM7500_SIM7600 拨号指令.....	10
4.4 利用 DHCP 给 Linux 网卡获取 IP.....	10

1. 本文目的

本文档介绍了如何在 Linux OS 中为 SIMCOM 模块生成 USB 驱动程序，以及如何加载使用 SIMCOM WWAN 驱动。

SIMCom
Confidential

2.SIM7500_SIM7600 USB 相关描述

SIM7500_SIM7600 系列模块的 USB VID 是 0x1E0E PID 是 0x9001。

作为 Slave USB 设备，配置如下表

Interface number	Driver	Description
0	USB Serial	Diagnostic Interface
1	USB Serial	GPS NMEA Interface
2	USB Serial	AT port Interface
3	USB Serial	Modem port Interface
4	USB Serial	USB Audio Interface
5	USB RMNet	wwan interface
6	adb	Android adb debug port

※ 特别注意

在 Linux 系统中要防止 USB Serial 驱动把 Net interface 枚举占用

3.USB 串口驱动使用

3.1 USB Serial 的内核配置支持

在 linux 内核配置中，请确保

```
CONFIG_USB_SERIAL=y  
CONFIG_USB_SERIAL_WWAN=y  
CONFIG_USB_SERIAL_OPTION=y
```

3.2 修改驱动代码增加 SIM7500_SIM7600 的 VID/PID 并且跳过 Interface 5

找到内核源码文件 option.c(一般情况下，路径在 drivers/usb/serial/option.c)

➤ 如果是较新的内核版本，一般可以跳过对于 option.c 驱动文件的修改

可以跳过 option.c 驱动文件修改的特征，option_ids 列表中包含下列语句块之一

```
1. { USB_DEVICE(ALINK_VENDOR_ID, SIMCOM_PRODUCT_SIM7100E),  
2.   .driver_info = RSVD(5) | RSVD(6) },  
3. { USB_DEVICE(ALINK_VENDOR_ID, SIMCOM_PRODUCT_SIM7100E),  
4.   .driver_info = (kernel_ulong_t)&simcom_sim7100e_blacklist },
```

在 option_ids 列表中增加

```
5. ... ..  
6. static const struct usb_device_id option_ids[] = {  
7.   //for SIM7600 modem for NDIS  
8.   #if 1 //Added by Simcom  
9.   { USB_DEVICE(0x1e0e, 0x9001) },  
10.  #endif  
11.  ... ..
```

并且在 `option_probe` 里过滤掉 interface 5

```
1. static int option_probe(struct usb_serial *serial,
2.                         const struct usb_device_id *id)
3. {
4.     .....
5.     /* sim7600 */
6.     #if 1 //Added by Simcom`
7.     if (serial->dev->descriptor.idVendor == cpu_to_le16(0x1E0E) &&
8.         serial->dev->descriptor.idProduct == cpu_to_le16(0x9001)
9.         && serial->interface->cur_altsetting->desc.bInterfaceNumber >=5)
10.        return -ENODEV;
11. #endif
12.     .....
13. }
```

➤ 内核调试信息打印

如果驱动正确编译到内核，内核开机找到模块后，会打印如下信息

```
usb 1-1: new high speed USB device using rt3xxx-ehci and address 2
option 1-1:1.0: GSM modem (1-port) converter detected
usb 1-1: GSM modem (1-port) converter now attached to ttyUSB0
option 1-1:1.1: GSM modem (1-port) converter detectedSmart Machine Smart Decision
usb 1-1: GSM modem (1-port) converter now attached to ttyUSB1
option 1-1:1.2: GSM modem (1-port) converter detected
usb 1-1: GSM modem (1-port) converter now attached to ttyUSB2
option 1-1:1.3: GSM modem (1-port) converter detected
usb 1-1: GSM modem (1-port) converter now attached to ttyUSB3
option 1-1:1.4: GSM modem (1-port) converter detected
usb 1-1: GSM modem (1-port) converter now attached to ttyUSB4
```

`dev/ttyUSB0~4` 就会生成，上层应用就可以通过这些设备和模块交互了（发送 AT 命令等）

4. USB NDIS NET 使用

4.1 内核配置支持

首先需将这个配置项打开。

```
CONFIG_USB_USBNET=y
```

我们会提供一个文件 `sim7500_sim7600_wwan.c`，当然不限于这个文件名称，可以通过 Simcom 的技术支持获取。

`sim7500_sim7600_wwan.c` 放在 `drivers/net/usb` 目录下，并且修改 Makefile

```
obj-$(CONFIG_USB_USBNET) += usbnet.o sim7500_sim7600_wwan.o
```

驱动正确编译到内核，内核开机后连到模块会打印如下信息

```
sim7500_sim7600_wwan 1-1:1.5 wwan0: register 'sim7500_sim7600_wwan' at  
usb-0000:02:03.0-1, SIMCOM wwan/QMI device, 8a:de:f6:67:ce:1b
```

※ 特别注意

在 Linux 系统中要防止 `qmi_wwan.c` 驱动把 Net interface 枚举占用，`qmi_wwan.c` 位于 `[linux-src]/drivers/net/usb/` 下面。

```
{QMI_FIXED_INTF(0x1e0e, 0x9001, 5)},
```

该语句出现在 `qmi_wwan.c` 中，即枚举占用，可以移除所包含的语句，并使用我司提供的驱动

或者

```
{QMI_QUIRK_SET_DTR(0x1e0e, 0x9001, 5)}
```

该语句出现在 `qmi_wwan.c` 中，可以移除该语句并使用我司提供的驱动，也可以直接使用 `qmi_wwan.c` 针对于直接使用 `qmi_wwan.c` 这里还有个注意事项，就是 SIM7500/7600 模块是 Raw IP 模式的 需要设置驱动参数 `echo Y > /sys/class/net/wwan0/qmi/raw_ip`

4.2 使用 ifconfig 查看网卡信息，默认处于 down 状态

```
# ifconfig wwan0
```

```
wwan0  Link encap:Ethernet HWaddr D6:D8:6C:10:B0:0E  
        BROADCAST MULTICAST 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 (0.0 B) TX bytes:0 (0.0 B)
```

如果需要查询全部网卡信息，可以使用 **ifconfig -a**

4.3 SIM7500_SIM7600 拨号指令

SIM7500_SIM7600 网卡拨号是使用一个 AT 命令来操作的。

AT\$QCRMCALL=1,1 拨号上网（一般使用方法，具体 AT 描述见 NDIS 拨号文档）

AT\$QCRMCALL=0,1 挂断（一般使用方法，具体 AT 描述见 NDIS 拨号文档）

4.4 利用 DHCP 给 Linux 网卡获取 IP

模块拨号上网后，LINUX 平台一般需要 DHCP 客户程序来给 wwan 网卡获取 IP, DNS 等信息。

如果是 PC 端的 LINUX（比如 Ubuntu），只需要先让网卡 down 然后再 up 一下，PC 会自动用 netmanager 自带的 dhcp 获取 IP。

如果是嵌入式的 LINUX 系统，一般需要用 udhcpd 命令

```
udhcpd -i wwan0
```

再次查看网卡信息，会看到已经拿到 IP

```
wwan0  Link encap:Ethernet HWaddr D6:D8:6C:10:B0:0E  
        inetaddr:10.47.77.207 Bcast:10.47.77.223 Mask:255.255.255.224  
        inet6 addr: fe80::d4d8:6cff:fe10:b00e/64 Scope:Link  
        UP BROADCAST RUNNING MULTICAST MTU:1500 Metric:1  
        RX packets:2 errors:0 dropped:0 overruns:0 frame:0  
        TX packets:9 errors:0 dropped:0 overruns:0 carrier:0  
        collisions:0 txqueuelen:1000  
        RX bytes:612 (612.0 B) TX bytes:1642 (1.6 KiB)
```