

SIM7100 Linux NDIS 使用手册

一 SIM7100 USB 相关描述

SIM7100 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		
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 Net	NDIS wwan interface
6	USB adb	Android add debug port

在 linux 系统中 要防止 USB serial 驱动也把 Net interface 枚举占用。

二 USB 串口使用

1 LINUX kernel USB Serial 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

2 增加 SIM7100 VID/PID 并且调过 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（3.2 以上）

```
#define SIMCOM_SIM7100_VID      0x1E0E
#define SIMCOM_SIM7100_PID      0x9001

//for SIM7100 modem for NDIS
```

```
static const struct option_blacklist_info simcom_sim7100_blacklist = {
    .reserved = BIT(5),
};
```

在 option_ids 列表中增加

... ..

```
//for SIM7100 modem for NDIS
{ USB_DEVICE(SIMCOM_SIM7100_VID, SIMCOM_SIM7100_PID),
  .driver_info = (kernel_ulong_t)& simcom_sim7100_blacklist t
},
... ..
```

如果是较低的内核版本，

```
#define SIMCOM_SIM7100_VID          0x1E0E
#define SIMCOM_SIM7100_PID          0x9001
```

在 option_ids 列表中增加

```
{ USB_DEVICE(SIMCOM_SIM7100_VID, SIMCOM_SIM7100_PID)}, /*SIM7100 */
```

并且在 option_probe 里过滤掉 interface 5

```
/* sim7100 */
if (serial->dev->descriptor.idVendor == SIMCOM_SIM7100_VID &&
    serial->dev->descriptor.idProduct == SIMCOM_SIM7100_PID &&
    serial->interface->cur_altsetting->desc.bInterfaceNumber == 5 )
    return -ENODEV;
```

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 detected
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）。

三 USB NDIS NET 使用

1 内核支持

Linux 从 3.4.1 开始已经把 QMI WWAN 驱动包含到源码里。
因此，如果内核版本高于或者等于 3.4.1，只需将这三个配置项打开就可。

```
CONFIG_USB_WDM=y
CONFIG_USBNET=y
CONFIG_USB_NET_QMI_WWAN=y
```

并且在 qmi_wwan.c 里增加 SIM7100 的 VID/PID

```
{QMI_GOBI_DEVICE(0x1e0e, 0x9001)}, /* SIM7100 Modem Device */
```

如果内核版本低于 3.4.1,

```
CONFIG_USB_WDM=y
CONFIG_USBNET=y
```

我们会提供两个文件 cdc-wdm.c 和 qmi_wwan.c。

cdc-wdm.c 放在 drivers/usb/class 目录下，如果原来内核已经有该文件，请用新的替代。

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

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

```
qmi_wwan 1-1:1.5: cdc-wdm0: USB WDM device
qmi_wwan 1-1:1.5: wwan0: register 'qmi_wwan' at usb-rt3xxx-1, Qualcomm Gobi
wwan/QMI device, d6:d8:6c:10:b0:0e
```

2 使用 ifconfig 查看网卡信息，默认处于 down 状态。

```
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)

2 SIM7100 拨号指令

SIM7100 网卡拨号是使用一个 AT 命令来操作的。

AT\$QCRMCALL	
AT\$QCRMCALL =<Action>, <Instance> [,<IP Type> [,<Tech Pref > [,<umts profile number> [,<cdma profile number > [,<APN>]]]]]	Command triggers an RmNet call based on <Action> parameter which is typically a start of an RmNet Call or stop of a RmNet call. < Action > 0 – Stop 1 – Start <Instance> 1 to RMNET_ NUM_LAPTOP_ INSTANCES <IP Type> 1 – Ipv4 2 – Ipv6 3 – Ipv4v6 <Tech Pref> 1 – 3GPP2 2 – 3GPP <umts_profile> 1 to 16
AT\$QCRMCALL?	Read the current RmNet type.
Example	AT\$QCRMCALL=1,1,1,2,1,, OK

一般用法

AT\$QCRMCALL=1,1 拨号上网，
AT\$QCRMCALL=0,1 去激活。

3 利用 DHCP 给 Linux 网卡获取 IP

Modem 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)
```