**U9300 GobiNet驱动及AT命令拨号**

**使**

**用**

**手**

**册**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 编制 | 版本 | 审核 | 日期 | 说明 |
| Wangzm | V1.0 |  | 2016.11.18 | 初始版本 |
| 陈雷 | V1.1 |  | 2016.12.16 | 适配kernel 3.0.x;3.10.x |
| 陈雷 | V1.2 |  | 2017.2.10 | 1.调整qmi消息顺序  2.增加适配kernel |
| 张晓成 | V2.0 |  | 2017.04.27 | 1，解决大小端问题；  2，解决多模块环境下第二个及后模块拨号不成功的问题；  3，解决多模块mac地址都一样的问题；  4，兼容8300驱动；  5，增加对电源管理的控制；  6，解决客户多次ifconfig up/down 系统奔溃的问题。  7，解决客户多次加载注销驱动出现系统奔溃的问题； |
| 张晓成 | V2.0 |  | 2017.05.05 | 1,更新log规则  2,更新qcqmi创建规则 |
| 张晓成 | V2.0 |  | 2017.05.08 | 1,支持9507E新PID/VID |
| 张晓成  王战盟 | V2.1 |  | 2017.08.02  2017.08.17 | 1,通过增加驱动参数，支持多模块环境和CM自动拨号，默认是支持多模块。  2,修复针对一些特定内核版本中的bug  3，支持增加新的PID  4，增加对更多内核的适配，主流内核都已经支持  5，更新用户说明手册 |
| 张晓成 | V3.0 |  | 20180124 | 1. 优化驱动加载时间。 2. 更新用户说明手册。 3. 注：2017年2月8日后的版本，快速拨号返回1,V4. |
| 张晓成 | V3.2 |  | 20181116 | 1，解决概率拨号失败问题。 |

**目 录**

[0.手册说明 4](#_Toc10082)

[1. USB端口信息 4](#_Toc30766)

[2. 驱动文件说明 4](#_Toc7133)

[3. USB驱动添加系统组件 5](#_Toc5960)

[3.1 USB串口驱动添加 5](#_Toc14222)

[3.2 增加具体设备驱动 5](#_Toc3418)

[3.3 USB串口驱动过滤NDIS接口 5](#_Toc23948)

[3.4 USB串口驱动加载方法 6](#_Toc22503)

[4. Gobinet驱动加载 7](#_Toc18668)

[4.1驱动添加系统组件 7](#_Toc13006)

[4.2代码架构 7](#_Toc26313)

[4.3驱动编译 8](#_Toc15522)

[4.4驱动加载 9](#_Toc20904)

[4.5嵌入式系统适配说明 10](#_Toc19218)

[5. 拨号指令 10](#_Toc210)

[5.1 设置APN 10](#_Toc22350)

[5.2 拨号命令 10](#_Toc5012)

[5.3 挂断命令 11](#_Toc27334)

# 0.手册说明

本文档介绍了U9300系列模块在Linux系统下的Gobinet驱动以及AT命令拨号的使用方法。

对于U9300C/W Linux系统下的AT拨号，需要Driver 的支持，龙尚提供的Driver为 Gobinet驱动。本文介绍了基于GobiNet驱动的AT$QCRMCALL拨号的使用方法。

注：目前针对客户端拨号程序longsung-CM 和 AT命令快速拨号两种不同的拨号方式，Gobinet驱动是相同，需要根据客户需求使用相应的拨号方式。

# 1. USB端口信息

U8300C/W、U9300C/W 模块 USB 会枚举出 6 个逻辑端口，如表所示：

|  |  |  |  |
| --- | --- | --- | --- |
|  | U8300C/W | U9300/U9507C/W | U9507E |
| Vendor ID | 0x1c9e | 0x1c9e | 0x2df3 |
| Product ID | 0x9b05 | 0x9b3c | 0x9b3e |
| Interface 0 | Debug | ADB | Modem |
| Interface 1 | Modem | Modem | AT |
| Interface 2 | AT | AT | Pipe |
| Interface 3 | Pipe | Pipe | NDIS |
| Interface 4 | NDIS | NDIS | ADB |
| Interface 5 | ADB |  |  |

# 2. 驱动文件说明

龙尚提供的驱动与客户端清单：

1） GobiNet驱动文件；

2） usbnet修改文件，龙尚的驱动需要对usbnet有特定修改； (有些内核需要修改该文件(通常小于linux3.4的内核都需要修改，仅供参考，不做验证凭证)，具体需要在实施中验证不同的客户系统版本)

当U9300模块正常工作并且Gobinet驱动成功加载后，会在系统中创建两个文件：

1） usb\* （根据内核版本会有差异，可能会是 eth\*等）

网络设备，\*表示设备顺序号，该值会根据系统中usb设备的不同而不同，例如usb0。该设备是用做传输网络数据。

2） qcqmi\*

该设备是应用层 与 usbnet网卡通信的通道，用来获取网卡设备信息。

# 3. USB驱动添加系统组件

## 3.1 USB串口驱动添加

普通的数据卡(modem)使用USB接口，对上层的功能接口是串口，在Linux系统中通常使用USB转串口的驱动。

驱动添加需要配置Linux内核，配置方法如下：

cd kernel

make menuconfig

device drivers->usb support->usb serial converter support

选中如下组件：

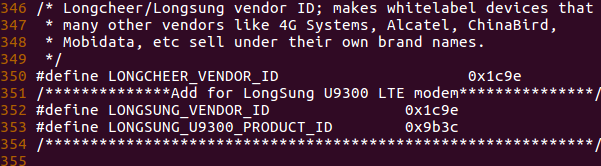
USB driver for GSM and CDMA modems

选中后保存配置。

## 3.2 增加具体设备驱动

打开内核源码文件option.c(路径一般为drivers/usb/serial/option.c)；

在源码中找到option\_ids数组，在数组中添加Longsung 9300产品的VID(0x1c9e)和PID(0x9b3c)；新增海外产品VID(0x2df3)和PID(0x9b3e)，添加方法一样。不同产品的VID/PID会有不同，具体细节请联系FAE。



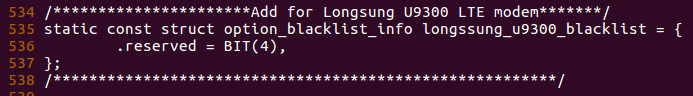
## 3.3 USB串口驱动过滤NDIS接口

由于USB串口跟netdev都属于非标准CDC设备，需要防止NDIS口被USB串口驱动加载而导致无法正常加载NDIS口驱动。有三种方式可以解决：

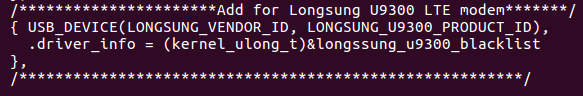
特别注意：新增海外产品PID/VID(0x2df3)和PID(0x9b3e)，NDIS口是3 即 interface 3。过滤方法一样。只是更改端口号即可。

方法1>比较新的kernel版本(3.8以上)，在option.c中的opiton\_ids中添加blacklist，驱动在加载时会自动跳过blacklist指定的interface；

设置interface 4不加载otpion驱动：



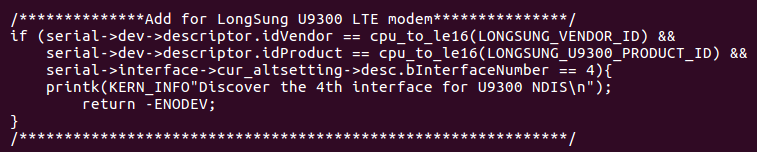
添加blacklist到option\_ids数组中：



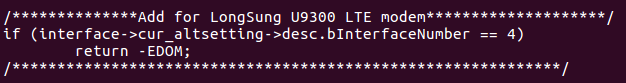
方法2>（推荐）对于之前的内核，不支持在option\_ids数组中设置过blacklist，要先增加U8300的PID和VID：



在probe函数内判断当前interface号进行过滤(海外产品过滤3口)；



方法3>（不推荐）对于使用usb-serial.ko驱动的用户，需要在usb-serial.c文件中的usb\_serial\_probe()函数开始增加如下判断来过滤NDIS接口：



## 3.3.1 USB串口驱动过滤adb接口

强烈建议：模块U9300/U9507C/W，如果客户平台不使用adb口，即ttyUSB0, 在option驱动中将该口过滤掉。过滤方法如下：

if(serial->dev->descriptor.idVendor == 0x1c9e &&

serial->dev->descriptor.idProduct == 0x9b3c &&

serial->interface->cur\_altsetting->desc.bInterfaceNumber == 0){

printk("GZZ Discover the 0th interface for 9300 ndis\n");

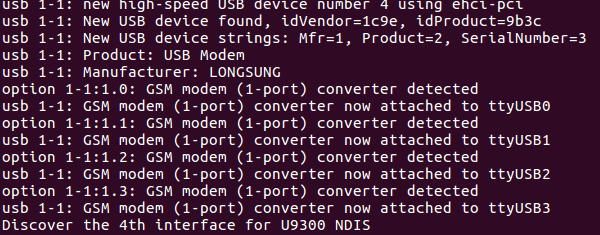
return -ENODEV;

}

## 3.4 USB串口驱动加载方法

本文以Ubuntu系统中的option驱动为例。加载USB串口驱动：sudo modprobe option

使用dmesg命令查看系统log，如下图所示，除了NDIS口，其它端口都加载上USB串口驱动。



# 4. Gobinet驱动加载

## 4.1驱动添加系统组件

Gobinet驱动需要系统的usbnet驱动支持，因此需要配置Linux内核，配置方法如下：

cd kernel

make menuconfig

device drivers->Network device support->usb Network Adapters

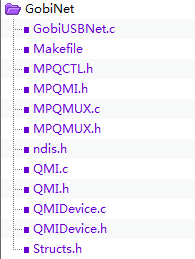
选中如下组件：

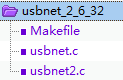
Multi-purpose USB Networking Framework

选中后保存配置，重新编译内核。

## 4.2代码架构

驱动以源代码的形式提供，由用户在自己的系统编译，下面是代码结构：





龙尚的驱动需要对usbnet有特定修改，其中usbnet.c是修改前的文件，usbne2.c是修改后的文件，用户可以将修改点增加在系统原有的文件中，或者用龙尚的代码编译成新的usbnet.ko。

不同内核可能会存在少许差异，需要根据具体情况进行修改。可以参考我们在参考代码进行修改。

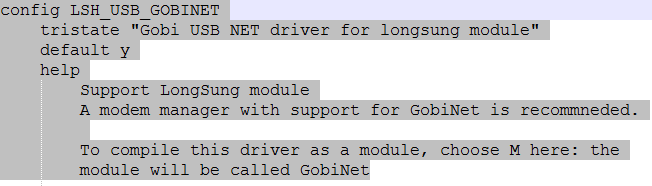
## 4.3驱动编译

用户可以单独编译，也可以将代码放入内核中一起编译。

1） 与内核一起编译

将代码文件复制到用户自己的kernel/drivers/net/usb目录下，编到内核中即可，可以参考下面方法：

1>在系统的kernel/drivers/net/usb/Kconfig中增加以下内容：



2>在kernel/drivers/net/usb/Makefile中增加以下内容：



* 之后每次编译内核都会自动编译Gobinet驱动。

2） 以module形式单独编译成ko文件

* 修改Makefile中KDIR的值为kernel的编译路径；
* 在GobiNet驱动目录下执行make命令，编译成功就可以在当前目录下生成GobiNet.ko驱动；
* 如果用make install命令，就会把编译出来的GobiNet.ko同时安装到系统的module目录中，并且分析依赖关系，将所依赖的其它相关驱动自动链接。这种方式需要修改makefile文件中的OUTPUTDIR的值为相应系统的目录，编译成功后就会把GobiNet.ko安装到系统中。

## 4.4驱动加载

1） 通过insmod命令单独加载Gobinet驱动：

sudo insmod GobiNet.ko

2） 或者通过modprobe命令加载Gobinet驱动：

modprobe GobiNet

会将驱动安装到系统的module目录中，并且分析依赖关系，将相关模块同时加载。如usbnet.ko等。

1. 在1)和2)的命令后面加 debug=1，可以将调试log全部打印出来：

sudo insmod GobiNet.ko debug=1

1. 如果是配合自动拨号软件CM程序，加载驱动时需要

添加参数mod2ls=1

sudo insmod GobiNet.ko mod2ls=1

如果客户要编译到内核，需要在GobiUSBNet.c 修改变量为int mod2ls = 1；

Eg: 下面以Ubuntu 10.04（kernel 2.6.32）系统为例说明以module形式把Gobinet驱动编译成ko的文件形式和加载方法：

1） sudo rmmod usbnet

2） 进入龙尚提供的usbnet目录下，依次执行 make install; modprobe usbnet2

3） 进入龙尚提供的GobiNet驱动目录，依次执行

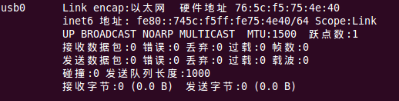
make install; modprobe GobiNet

4） 查询设备

ls /dev/qc\*，会查到/dev/qcqmi0等；



Ifconfig -a，会有usb0，（有的内核可能是eth0、eth1等）如下所示：



说明GobiNet驱动正常加载。

## 4.5嵌入式系统适配说明

龙尚的驱动需要对usbnet有特定修改，在文件usbnet.c中。usbnet2是在系统原始文件的基础上增加了修改点，是龙尚测试使用的，主要目的是防止干扰原系统的usbnet.ko，嵌入式系统中可以将修改点直接合入usbnet.c中。

# 5. 拨号指令

## 5.1 设置APN

APN必须被正确配置，可以使用AT+CGDCONT命令进行配置（AT命令的具体使用请参考龙尚的AT命令手册），并且注意公网卡和专网卡的APN区分。

## 5.2 拨号命令

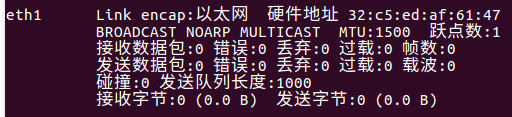
1）执行命令AT$QCRMCALL=1,1

若返回如下结果，则表明拨号成功：

$QCRMCALL:1，V4

OK

拨号成功后，用ifconfig -a可查到网卡信息（注意，不用-a选项时，查不到该信息）如图：



1. 获取ip地址

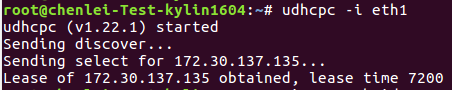
2.1在linux系统上需要执行udhcpc或dhclient等系统工具来获取IP地址。

例如 udhcpc -i eth1

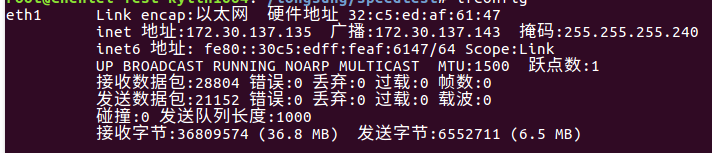
2.2 在android系统上需要执行android自带的工具来获取IP地址：

例如：netcfg eth1 up 设置eth1 up；

netcfg eth1 dhcp 获取eth1的IP地址；



用ifconfig命令查询，可以看到usb0（有的内核可能是eth0、eth1等）设备中已经有了ip地址，如下所示：



## 5.3 挂断命令

执行命令AT$QCRMCALL=0,1

若返回如下结果，则表明挂断成功：

OK

