WH-G405tf Linux 系统 PC 侧驱动编译与安装说明

文档版本：V1.0.0

目录

1、背景 3
2、基本步骤 3
2.1 下载内核文件并提取驱动文件 3
2.2 修改虚拟串口驱动代码 3
2.3 修改网卡驱动代码 5
2.4 编译脚本 5
2.5 驱动加载 7
2.6 AT 指令拨号上网 8
3、系统差异 10
3.1 系统差异 10
3.2 Fedora 系统 kernel-devel 安装 10
3.3 嵌入式 Linux 平台 10
4、联系方式 11
5、免责声明 11
6、更新历史 11

1、背景

对于 usb 虚拟串口和 NDIS 网卡模式，linux 平台的桌面操作系统在某些情况下需要自己编译驱动文件，比如 pid 的新增、修改，接口号顺序的变化。本文档所提到的 linux 操作系统桌面版指的是内核版本 3.11 以上的主流操作系统。所有代码编译也是基于内核版本 3.11 以上。

2、基本步骤

2.1 下载内核文件并提取驱动文件

STEP1：在使用的系统上，用 uname–a 命令查看系统的内核版本号，并从 kernel.org 上面下载对应的内核版本。查看版本号的例子如下：

图 2-1 Linux 系统版本号查看

由上图的例子看见，所查看到的系统内核版本号为 4.13.0-32-generic，对于内核版本号，只需要关注到 4.13.0 即可，后面的-32-generic 不用关注，查看 linux 版本遵照此规则。

从 ***https://www.kernel.org/***网站找到对应的内核源码包，然后下载。

STEP2：从下载的内核版本中提取虚拟串口驱动文件 option.c 和必须的头文件 usb-wwan.h，网卡驱动文件 qmi_wwan.c。文件的路径如下：

l***inux_src\drivers\usb\serial\option.c***

linux_src\drivers\usb\serial\usb-wwan.h

linux_src \drivers\net\usb\qmi_wwan.c

2.2 修改虚拟串口驱动代码

STEP1：在 linux_src\drivers\usb\serial\option.c 文件中添加驱动 idProduct(即 pid)。

在静态结构体数组 static const struct usb_device_id option_ids[]中增加需要配置的 pid 信息。在这个数组中找到包含“ZTE_VENDOR_ID”的最后一行，在其后添加如下代码，其中斜体加黑的是需要增加的内容(因 G405tf 的 PID 为 0x0579, 所以实际上只需要加红色字体那一行即可)：

{ USB_VENDOR_AND_INTERFACE_INFO(ZTE_VENDOR_ID, 0xff, 0x02, 0x01) },

{ USB_VENDOR_AND_INTERFACE_INFO(ZTE_VENDOR_ID, 0xff, 0x02, 0x05) },

{ USB_VENDOR_AND_INTERFACE_INFO(ZTE_VENDOR_ID, 0xff, 0x86, 0x10) },

{ USB_DEVICE_AND_INTERFACE_INFO(ZTE_VENDOR_ID, 0x0579, 0xff, 0xff, 0xff) },

{ USB_DEVICE_AND_INTERFACE_INFO(ZTE_VENDOR_ID, 0x0543, 0xff, 0xff, 0xff),

.driver_info = (kernel_ulong_t)&net_intf1_blacklist},

{ USB_DEVICE_AND_INTERFACE_INFO(ZTE_VENDOR_ID, 0533, 0xff, 0xff, 0xff),

.driver_info = (kernel_ulong_t)&net_intf14_blacklist },

STEP2：网卡接口的剔除

ZXIC 的模块有 NDIS 网卡和 ECM 网卡两种网卡模式。

对于 ECM 网卡设备，可以直接参照 0x0579 这个 pid 的配置方式，拷贝一行在其下面粘贴，然后修改为设备的实际 pid 即可。具体所用的网卡模式和 pid 信息，可以从设备说明获取。

```
[redacted]-ubuntu:~$ lsusb
Bus 001 Device 003: ID 19d2:0579 ZTE WCDMA Technologies MSM
Bus 001 Device 002: ID 80ee:0021 VirtualBox USB Tablet
Bus 001 Device 001: ID 1d6b:0001 Linux Foundation 1.1 root hub
```

对于 NDIS 网卡设备，ndis 网卡接口是要剔除掉的，否则会被当做串口设备。如上面的代码中 pid 为 0x0533,0x0543 的两行，使用 driver_info 这个结构体定义来剔除对应的接口信息，然后用已经定义好的结构体 net_intf***X***_blacklist。其中 intf 后红色的 X 用于指明对应的接口号，该类结构体定义位于 linux_src\drivers\usb\serial\option.c 文件中的结构体 option_blacklist_info 变量定义之后。

比如一般接口 1 用来做 NDIS 网卡，那么就应像如下方式：

.driver_info = (kernel_ulong_t)&net_intf1_blacklist

net_intf1_blacklist 的定义如下，该结构体已有定义可以直接使用：

static const struct option_blacklist_info net_intf1_blacklist = {

.reserved = BIT(1),

};

如果使用其他接口，那么先查看是否已有定义，如果没有就自己新增一个。

STEP3：双网卡中网卡接口的剔除

对于双网卡来说，需要剔除掉两个接口。这种情况下，如果已有的网卡剔除结构体定义中没有，那么需要自己新增定义 ***net_intf14_blacklist***(本例中接口 1 和 4 是 ndis 网卡)，如下：

static const struct option_blacklist_info net_intf14_blacklist = {

.reserved = BIT(1) | BIT(4),

};

同理，如果是其他的接口，将对应接口值修改。

然后在 option_ids 中照如下方式添加：

{ USB_DEVICE_AND_INTERFACE_INFO(ZTE_VENDOR_ID, 0533, 0xff, 0xff, 0xff),

.driver_info = (kernel_ulong_t)&net_intf14_blacklist },

STEP4：adb 端口的剔除

某些模块设备含有 adb 调试端口，也需要剔除不适配串口驱动，剔除方法同网卡接口剔除方法，不再重复。

2.3 修改网卡驱动代码

在 linux_src \drivers\net\usb\qmi_wwan.c 中修改两处，在结构体数组 static const struct usb_device_id products[]中照如下添加，在{QMI_FIXED_INTF(0x19d2, 0x2002, 4)}这一行后面，增加针对 pid：0x0533,0x0543 的支持：

注：pid：0x0533,0x0543 是当前 ZMP 版本配置的 NDIS 网卡的 pid。

{QMI_FIXED_INTF(0x19d2, 0x2002, 4)}, /* ZTE (Vodafone) K3765-Z */

{QMI_FIXED_INTF(0x19d2, 0x0533, 1)},

{QMI_FIXED_INTF(0x19d2,0x0533, 4)},

{QMI_FIXED_INTF(0x19d2, 0x0543, 1)},

2.4 编译脚本

由于仅是对系统驱动模块进行增加修改，所以不用重新编译内核，仅把对应模块编译成.ko 文件即可。所以编译脚本不使用内核源码进行编译，而是直接在所用的系统上通过 kernel-devel 进行编译。kernel-devel 包含了用于内核开发环境所必须的内核头文件和 Makefile。编译使用的 Makefile 如下：

这个 Makefile 文件仅需要修改要编译的模块名称即可，也就是下面斜体代码中的倒数第二句 ***your_module.o***，将 ***your_module*** 修改为编译的模块名字。编译完成后直接生成.ko 文件。

注意：如果编译的代码是取自内核文件，那么必须保持其模块名与内核版本命名一致，否则会导致加载匹配错误。

 # To build modules outside of the kernel tree, we run "make"

 # in the kernel source tree; the Makefile these then includes this

 # Makefile once again.

 # This conditional selects whether we are being included from the

 # kernel Makefile or not.

ifeq ($(KERNELRELEASE),)

 # Assume the source tree is where the running kernel was built

 # You should set KERNELDIR in the environment if it's elsewhere

KERNELDIR ?= /lib/modules/$(shell uname -r)/build

#KERNELDIR ?= /lib/modules/2.6.28-11-generic/build

 # The current directory is passed to sub-makes as argument

PWD := $(shell pwd)

modules:

$(MAKE) -C $(KERNELDIR) M=$(PWD) modules

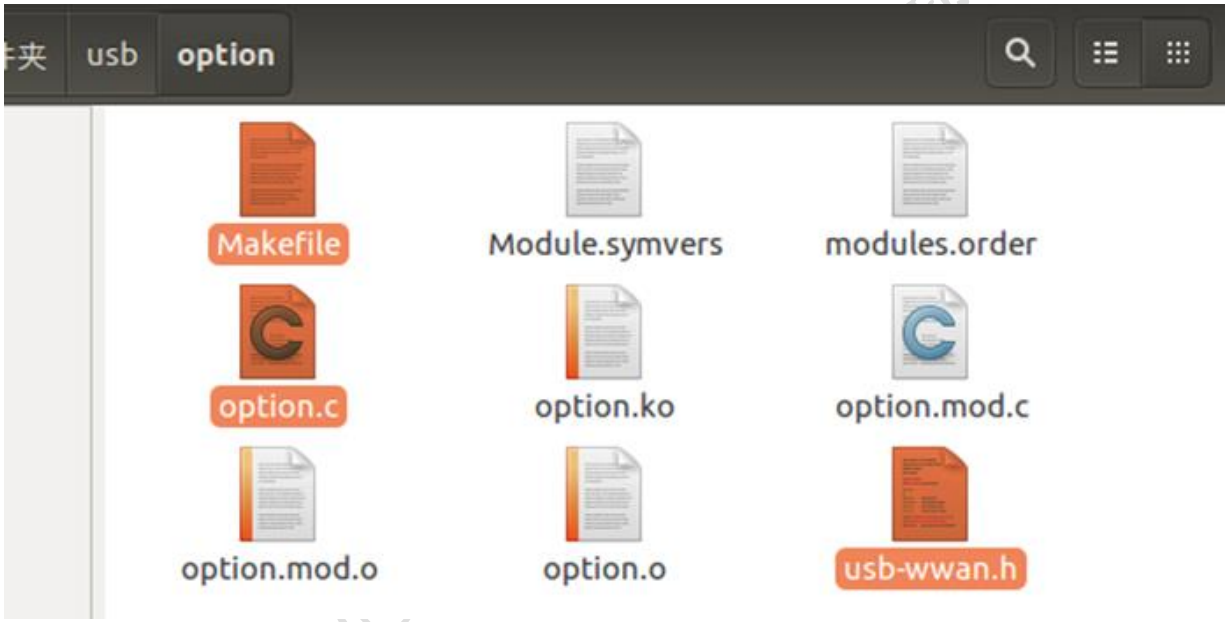
modules_install:

```
$(MAKE) -C $(KERNELDIR) M=$(PWD) modules_install
clean:
rm -rf *.o *~ core .depend .*.cmd *.ko *.mod.c .tmp_versions
.PHONY: modules modules_install clean
else
# called from kernel build system: just declare what our modules are
obj-m :=your_module.o
endif
```

2.4.1 option 编译的内容

option 仅需要三个文件 Makefile，option.c，usb-wwan.h。如下：

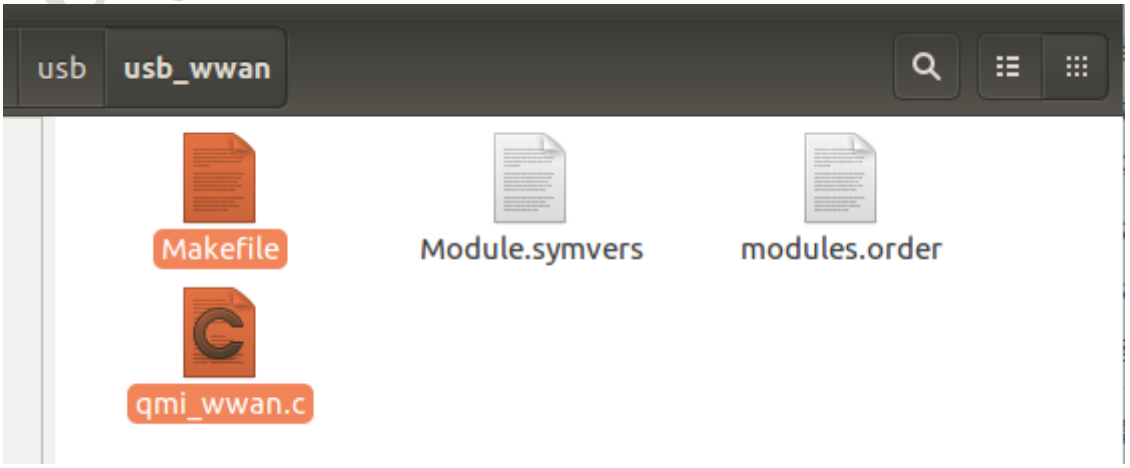
图 2-2 option 编译内容



2.4.2 qmi_wwan 编译的内容

qmi_wwan 编译需要两个文件：Makefile，qmi_wwan.c。

图 2-3 qmi_wwan 编译内容



2.5 驱动加载

2.5.1 文件依赖

编译完成的.ko 文件通过 insmod 命令进行加载。在终端进入到编译目录所在，然后输入如下命令：

insmod ./option.ko
./是指 option.ko 所在的当前目录。

直接加载一般都会遇到文件依赖问题，提示如下：
insmod: cannot insert 'option.ko': Unknown symbol in module

如果不是驱动修改引起的问题，一般就是所依赖的驱动没有加载导致的。需要查看所编译的驱动依赖哪些驱动文件。用如下命令：
modinfo ./option.ko

会给出一大段内容，最主要的下给出的内容的结尾，是驱动依赖：
depends: usb_wwan,usbserial

用 lsmod 命令查看驱动加载情况：
lsmod | grep usb

如果驱动已经加载，那么返回内容中会包含这两个驱动名字。如果没有返回内容或者返回内容中没有这两个，说明驱动未加载。

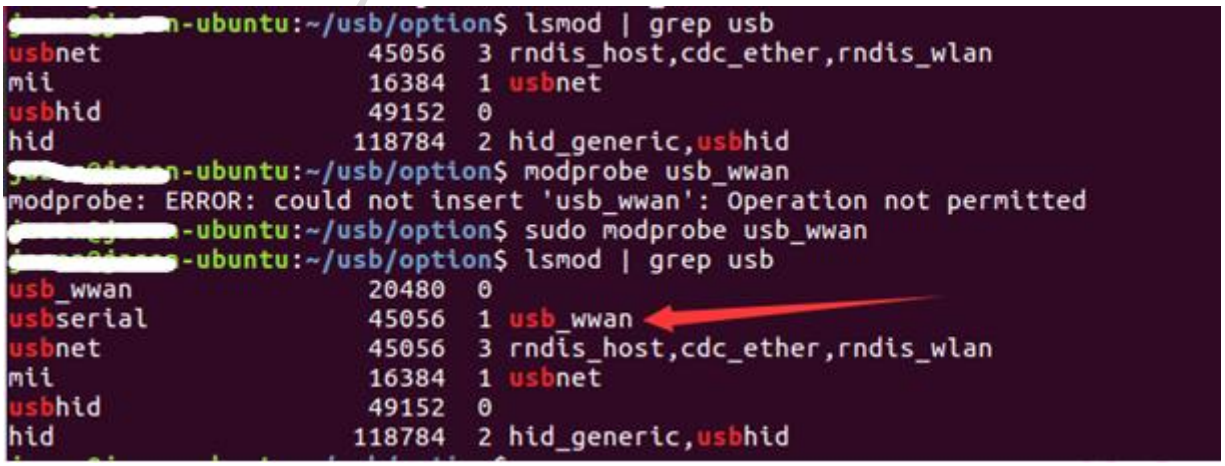
2.5.2 加载依赖的驱动文件

依赖的驱动加载用 modprobe 命令：

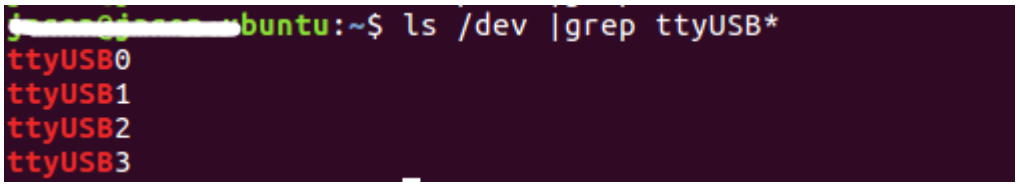
modprobe usb_wwan
该命令会自动加载 usb_wwan.ko 驱动文件以及它所依赖的驱动文件。

然后再用 lsmod 命令查看加载结果：
lsmod | grep usb

图 2-4 用 lsmod 命令查看加载结果



正常加载驱动以后，使用 ls /dev |grep ttyUSB*命令，可查询已经生成的设备：



2.6 AT 指令拨号上网

安装 minicom 串口助手
apt-get install minicom
配置 minicom
使用的串口号 minicom –s,
波特率 115200,8 位数据位，无校验，1 位停止位，无流控
依次发送 AT 指令

注意:模块默认为 ZNCARD=0，也就是在 Windows 平台上使用。当在 Linux 平台上使用时，先设置 ZNCARD=1。否则无法在 linux 上无法正常上网。切换模式后必须重启模块才可以，重新上电或发送 AT+Z

```
jason@jason-ubuntu: ~
AT+CFUN?
+CFUN: 0

OK
AT+CFUN=1
+CREG: 2

+CGREG: 2

OK

+CEREG: 2

+CGEV: ME PDN ACT 1

+CREG: 4

+ZLTENOCELL

+CEN1: 1,460

+CEREG: 1

^MODE: 17,10
at+csq
+CSQ: 135,99,17

OK
AT+CGACT?
+CME ERROR: 8010
AT+CGACT=1,1
+ZGIPDNS: 1,1,"IP","10.38.36.231","0.0.0.0","218.4.4.4","218.2.2.2"

OK
AT+ZGACT=1,1
+ZCONSTAT: 1,1

OK
```

指令执行完后，电脑可自动获取到 IP，此时可正常上网了，如下截图，IP 地址为：10.38.36.231。

```
enx00a0c6000000 Link encap:以太网  硬件地址 00:a0:c6:00:00:00
          inet 地址:10.38.36.231  广播:10.38.36.239  掩码:255.255.255.240
          inet6 地址: fe80::5435:a928:9896:f3b4/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  跃点数:1
          接收数据包:21 错误:0 丢弃:0 过载:0 帧数:0
          发送数据包:40 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:2899 (2.8 KB)  发送字节:6379 (6.3 KB)
```

```
[redacted]ubuntu:/etc/chatscripts$ ping www.baidu.com
PING www.a.shifen.com (180.97.33.108) 56(84) bytes of data.
64 bytes from 180.97.33.108: icmp_seq=1 ttl=53 time=25.9 ms
64 bytes from 180.97.33.108: icmp_seq=2 ttl=53 time=37.4 ms
64 bytes from 180.97.33.108: icmp_seq=3 ttl=53 time=35.0 ms
64 bytes from 180.97.33.108: icmp_seq=4 ttl=53 time=41.1 ms
64 bytes from 180.97.33.108: icmp_seq=5 ttl=53 time=31.0 ms
64 bytes from 180.97.33.108: icmp_seq=6 ttl=53 time=40.9 ms
^C
--- www.a.shifen.com ping statistics ---
6 packets transmitted, 6 received, 0% packet loss, time 5007ms
rtt min/avg/max/mdev = 25.987/35.293/41.198/5.426 ms
[redacted]ubuntu:/etc/chatscripts$
```

3、系统差异

3.1 系统差异

ubuntu 和 fedora 的系统差异见如下表格：

表 3-1 ubuntu 和 fedora 的系统差异

系统	环境	option 驱动依赖	Qmi_wwan 驱动依赖
Ubuntu	已安装 gcc,kernel-devel	usb_wwan, usbserial	usbnet, cdc-wdm
Fedora	需要自己安装 gcc,kernel-devel	usb_wwan	usbnet, cdc-wdm

3.2 Fedora 系统 kernel-devel 安装

fedora 系统默认不安装 kernel-devel，需要自己安装。如果通过 yum install(Fedora22 使用 dnf 替代 yum)方式安装，会发现所安装的 kernel-devel 与版本的内核版本不匹配问题。解决方法是自己从 linux 开源镜像站搜索与内核版本匹配的 kernel-devel 的 rpm 包，下载后自己手动安装。

如下几个版本的 kernel-devel 下载路径：

Fedora20：

http://mirrors.sohu.com/fedora/releases/20/Everything/x86_64/os/Packages/k/kernel-3.11.10-301.fc20.x86_64.rpm

Fedora21：

http://mirrors.163.com/fedora/releases/21/Everything/x86_64/os/Packages/k/kernel-3.17.4-301.fc21.x86_64.rpm

Fedora22：

http://mirrors.163.com/fedora/releases/22/Everything/x86_64/os/Packages/k/kernel-4.0.4-301.fc22.x86_64.rpm

3.3 嵌入式 Linux 平台

其他嵌入式 Linux 平台的驱动编译方法不用参考 2.4 节和 2.5 节的编译加载方法。根据自身需要，直接按照 2.2 节和 2.3 节所述修改方法修改后，随版本编译即可。

4、联系方式

公 司：上海稳恒电子科技有限公司

地 址：上海市闵行区秀文路 898 号西子国际五号楼 611 室

网 址：**www.mokuai.cn**

邮 箱： sales@mokuai.cn

电 话：021-52960996 或者 021-52960879

使命：做芯片到产品的桥梁

愿景：全球有影响力的模块公司

价值观：信任 专注 创新

产品观：稳定的基础上追求高性价比

5、免责声明

6、更新历史

2018-06-05　版本　1.0.0　创立