Linux USB 3G\_4G模块配置使用教程

**作者：向仔州**

**3G模块移植……………………………………………………………………………………………………..2**

**现在我们来切换3G模块模式………………………………………………………………………….8**

**APN添加方法………………………………………………………………………………………………….14**

**用AT指令集查看SIM卡信息和信号强度……………………………………………………...16**

**LinuxUSB 4G模块使用…………………………………………………………………………………….17**

**4G模块调试过程在ARM9平台下编译pppd软件………………………………………...20**

**3G和4G ping百度域名问题…………………………………………………………………………..24**

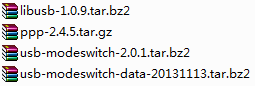
**Scsi驱动4G模块代码…………………………………………………………………………………….25**

**3G模块移植**

CPU：IMX6平台

操作系统：linux-3.14.28

1. 先要准备以下文件



将文件拷贝进ubuntu系统

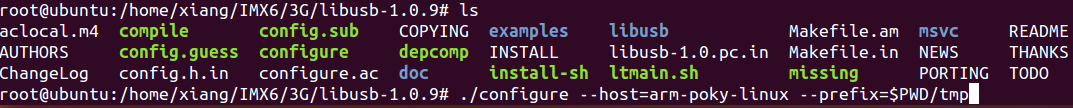


1. 先安装libusb软件

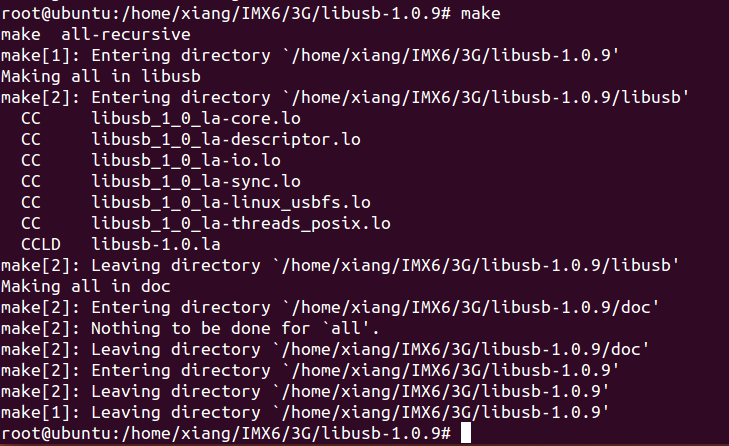
解压libusb

进入libusb-1.0.9目录

编译libusb



配置./configure

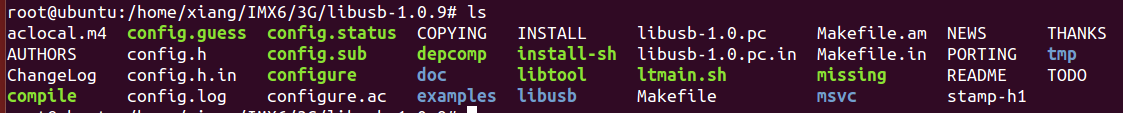


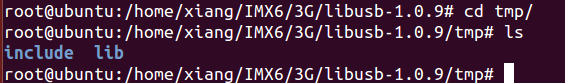
make



然后到libusb /tmp目录下看一下

这就是编译之后得到的tmp目录





把头文件和lib库文件复制进交叉编译工具链

因为我的CC编译器用的是下面这个路径

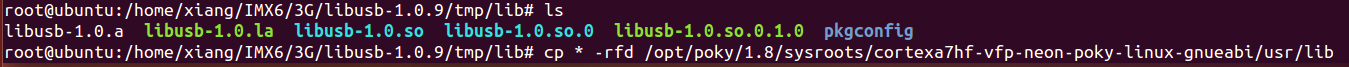
CC="arm-poky-linux-gnueabi-gcc -march=armv7-a -mthumb-interwork -mfloat-abi=hard -mfpu=neon -mtune=cortex-a9 --sysroot=/opt/poky/1.8/sysroots/cortexa7hf-vfp-neon-poky-linux-gnueabi"



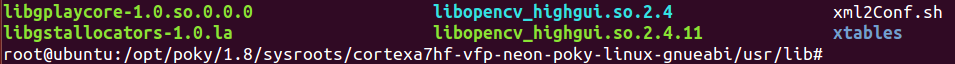
我要把libusb.h复制到交叉工具链这个路径



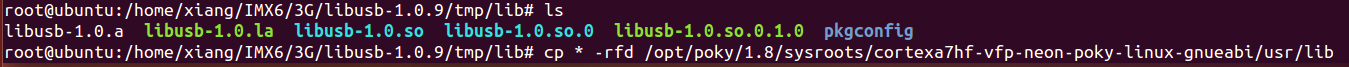
我还要把lib库复制进交叉工具链里面



cp \* -rfd 这个d就是说如果你是链接文件，就按照链接文件属性拷贝



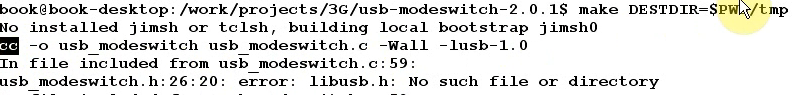
然后在将usblib库再拷贝一份到开发板文件系统的lib里面





其实拷贝一份usblib到开发板文件系统不就行了，为什么还要向X86的交叉编译工具链里面拷贝呢？ 那是因为我们下面编译usb\_modeswitch软件的时候要用到libusb里面的库。

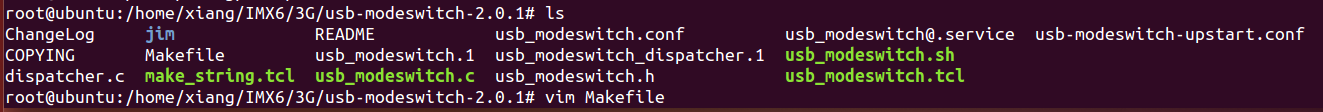
如果不拷贝一份include和lib的libusb库到交叉编译工具链里面，在编译usb\_modeswitch会报错



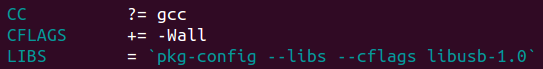
1. 编译usb\_modeswitch



进入usb-modeswitch-2.0.1目录

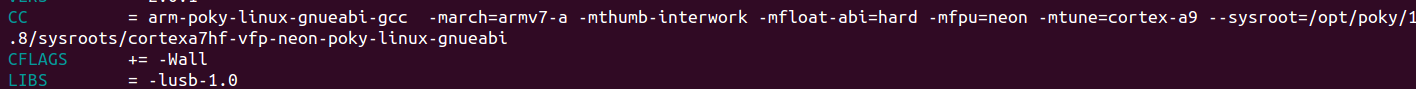


修改Makefile

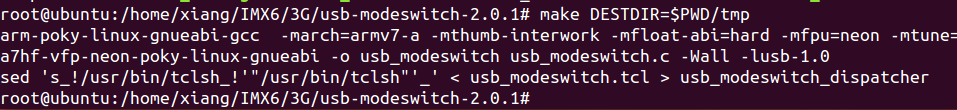


修改CC指定的交叉编译工具

修改LIBS，指定我们前面编译的libusb库



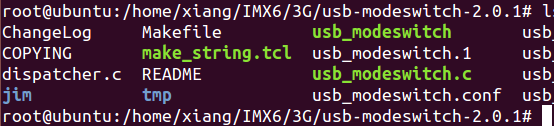
保存Makefile



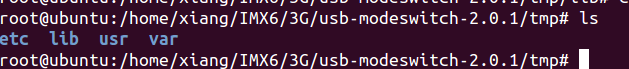
编译成功



Make install 会在目录下出现tmp



这就是编译之后得到的tmp目录





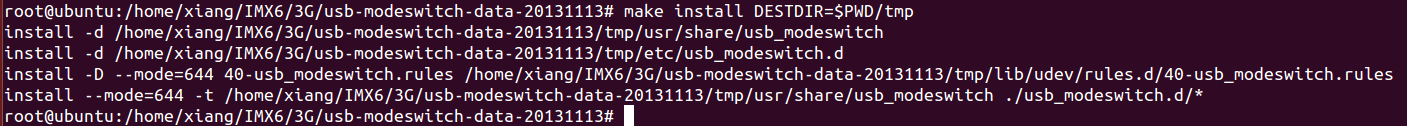
将这四个目录拷贝到开发板相应的文件系统下面去

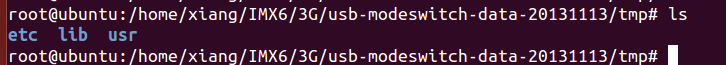
1. 编译usb\_modeswitch-data



编译







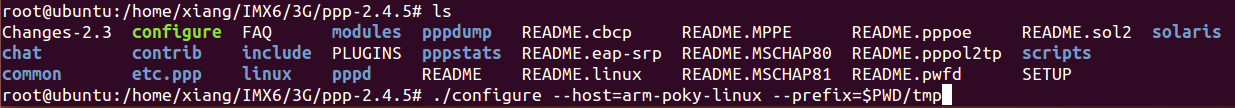
将usb\_modeswitch\_data里面的文件全部拷贝到开发板的文件系统里面。

5.编译pppd

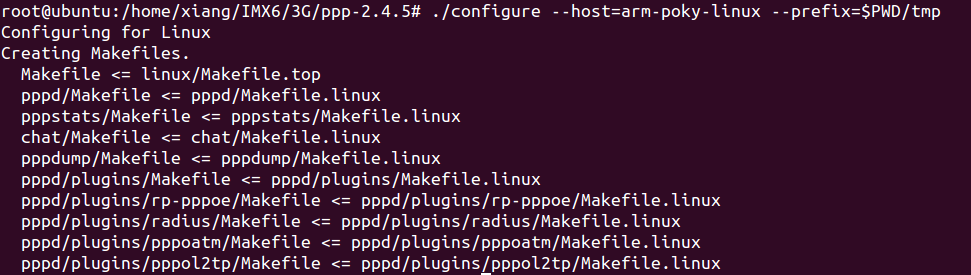


进入目录

错误的pppd编译方法，正确的编译方法在下面



配置configure

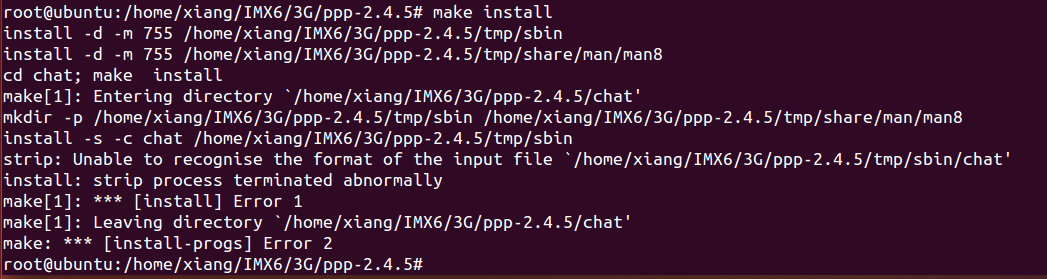


因为makefile我们没有看到CC所以我们在Make 外面指定GCC

配置之后是这样

然后我们编译





编译完成后在make install的时候发生错误。



根据这条错误问题，我们知道要用arm-linux-strip这个命令来消除符号。但是我觉得这样太麻烦了。还要一种方法就是去文件里面删除 -s。

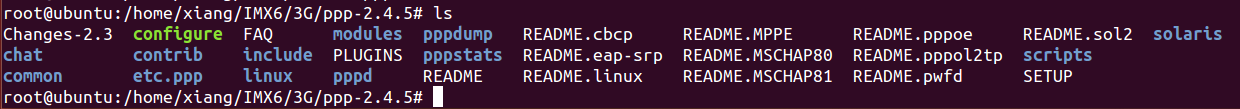


有这麽多-s 我们一个一个去删除吧。

正确的pppd编译方法



进入目录





就是要把这些Makefile.linux里面的-s全部去掉，不去掉会出现make的时候strip报错

**strip: Unable to recognise the format of the input file XXXX文件或者.so**

有些交叉编译器只需要取消下面6个-s就可以了，IMX6的交叉编译器就是，但是ARM9的不是

修改以下6个文件路径的Makefile.linux

chat/Makefile.linux:28: $(INSTALL) -s -c chat $(BINDIR)

pppd/plugins/radius/Makefile.linux:39: $(INSTALL) -s -c -m 755 radius.so $(LIBDIR)

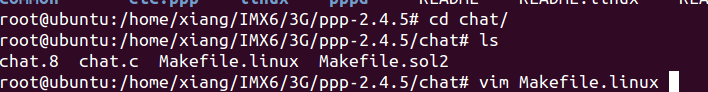
pppd/plugins/rp-pppoe/Makefile.linux:46: $(INSTALL) -s -c -m 4550 rp-pppoe.so $(LIBDIR)

pppd/Makefile.linux:102:EXTRAINSTALL = $(INSTALL) -s -c -m 555 srp-entry $(BINDIR)/srp-entry

pppdump/Makefile.linux:20: $(INSTALL) -s -c pppdump $(BINDIR)

pppstats/Makefile.linux:25: $(INSTALL) -s -c pppstats $(BINDIR)

我们先去修改这个chat/Makefile.linux: 文件





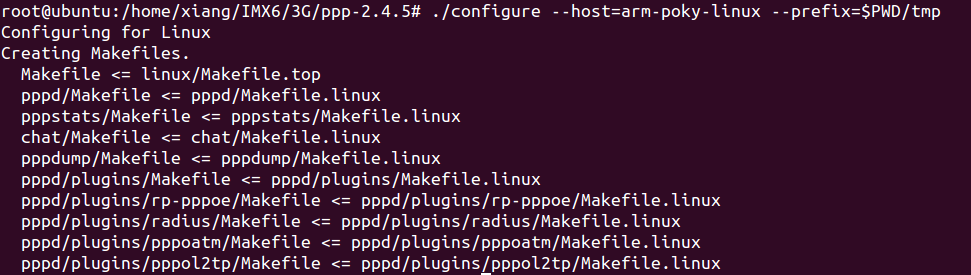
删除-s

修改后就是下面这样



然后保存退出

其余5个文件也是这样修改，删除所有-s的位置。



然后配置文件



编译make

然后make install 安装文件就在tmp目录下了

将这些文件拷贝进开发板文件系统。

3G用到的库文件移植就算完成了

**现在我们来切换3G模块模式**

将3G模块切换成ttyUSB功能有两种方法。

第一种：有些模块厂商已经在3G模块内部自动安装的模块切换协议。

我们就以M7281为例



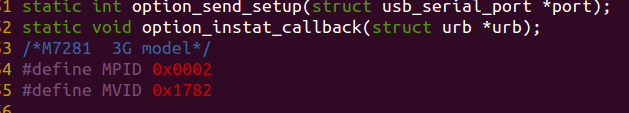
模块插入USB后会连接进USB设备，我们lsusb命令来查看模块的PID,VID。

我们M7281的VID是1782，PID是0002，这个不要搞反咯，我就是搞反了搞了几天。

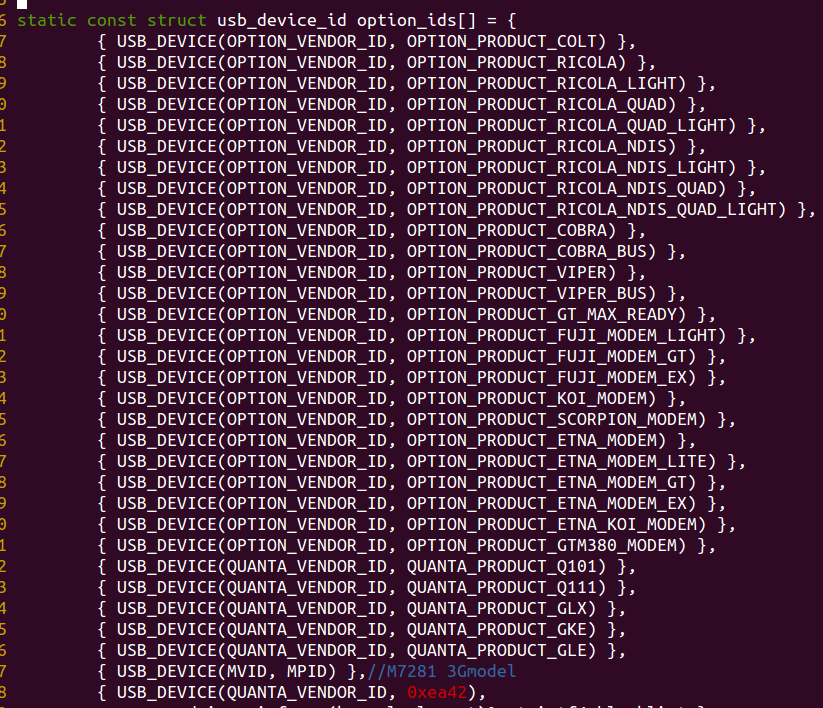
进入内核usb转串口目录

修改option文件

下面是修改具体步骤



在option.c文件里面加上PID,VID的宏，自己根据模块型号来定义。

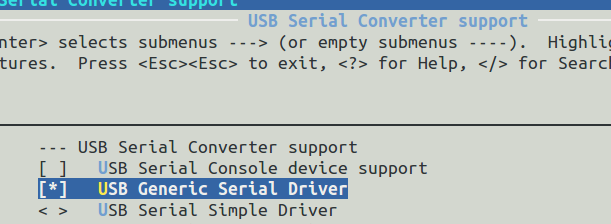


在option\_ids[]数组里面加一项自己模块的VID,PID宏。

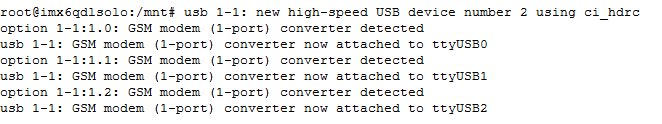




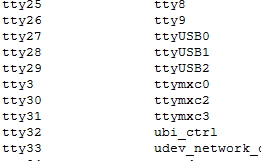


全部编译进内核，然后开发板上电。上电完成后启动3G模块就可以看到下面打印了。



这就是模块成功转换成了ttyUSB模式

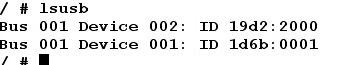
你在/dev设备节点下面也能看到。

第二种：有些3G厂商模块内部不带自动切换模式功能，那么我们要用usb\_modeswitch来切换

所以我们前面编译的usb\_modeswitch就是用在这里的。

假如我们用的是中兴的3G模块，型号为MF637，那么我们建立一个MF637.cfg文件

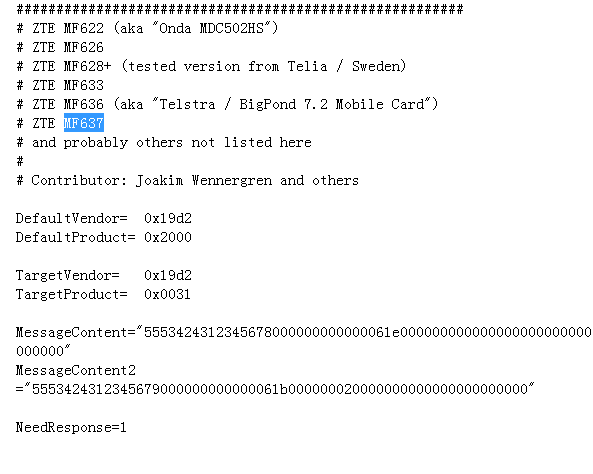
先插入MF637 USB 3G模块



全部复制粘贴进MF637.cfg文件

搜索到了VID,PID VID=19d2 ，PID=2000

然后我们拿到厂家给出的模式切换文件，寻找对应的PID,VID



找到对应的PID,VID后将上面所有的字符，复制粘贴进我刚才建立的MF637.cfg文件。

然后将MF637.cfg文件拷贝进开发板的文件系统/etc目录下。

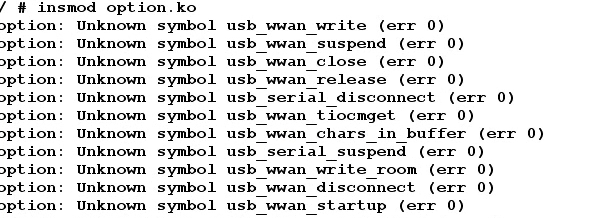
执行usb\_modeswitch

现在还没有出现ttyUSB，那是因为你还没有装驱动程序。



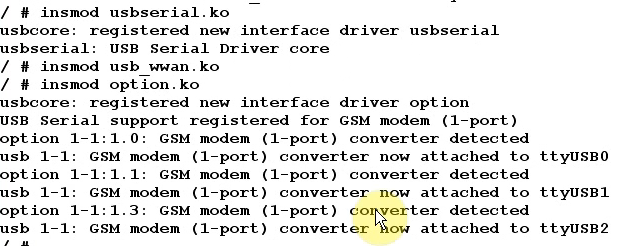
你可以用lsusb来查看，你会发现VID,PID已经是19d2:0031了，而不是之前的19D2：2000

接下来安装驱动程序：



安装option.ko发现找不到调用函数，这个问题是安装驱动程序顺序不对。

先安装usbserial.ko



最后是option

再安装wwan

你看ttyUSB已经出现了，证明模块以及正常转换了，下面我们来pppd拨号。

给开发板文件系统添加ppp文件

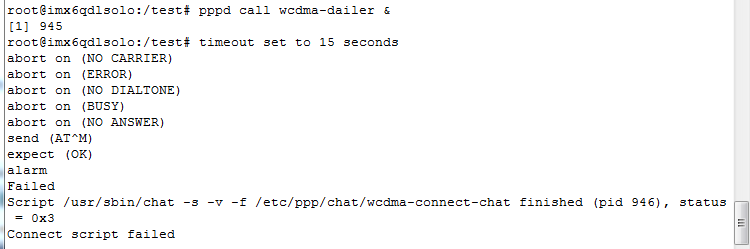


将该ppp文件放在文件系统/etc目录下



重复上面的3G模块启动动作，控制模块启动开关和复位开关的GPIO

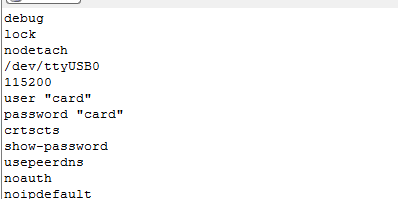
执行 pppd call wcdma-dailer & //这是联通sim卡的启动脚本 移动的又不一样哦！！



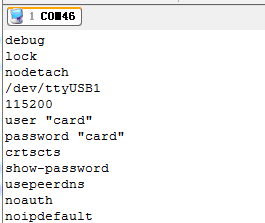
发现出现连接脚本错误



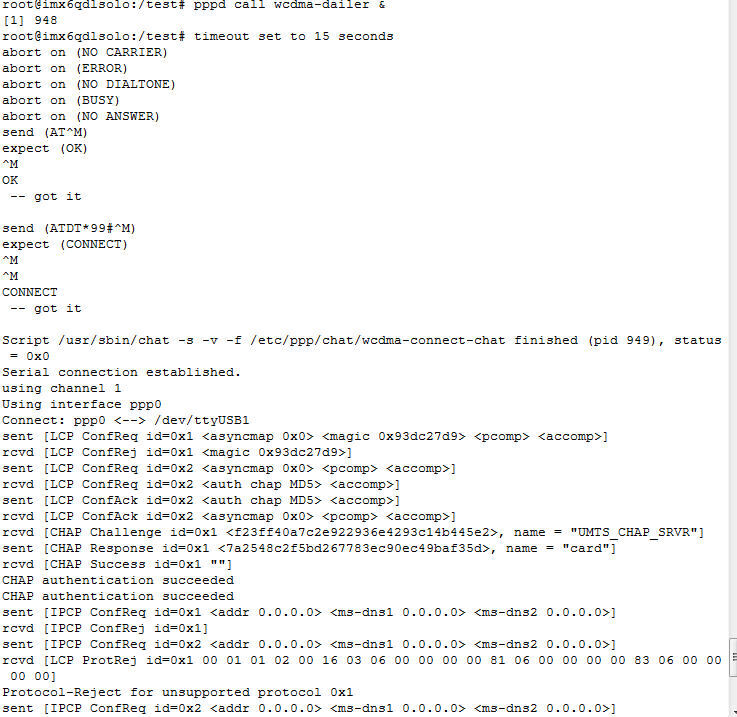
去开发板文件系统/etc/ppp/peer目录下查看wcdma-dailer文件



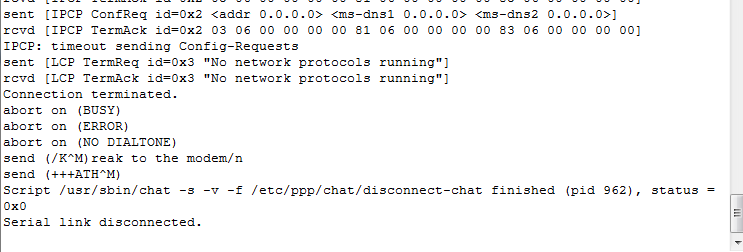
我们映射了三个ttyUSB。但是有个USB是执行AT指令集的，还有个USB是执行3G模块ppp协议的，所以我们要试试到底是哪个USB口



我改成ttyUSB1试试，然后保存wcdma-dailer文件

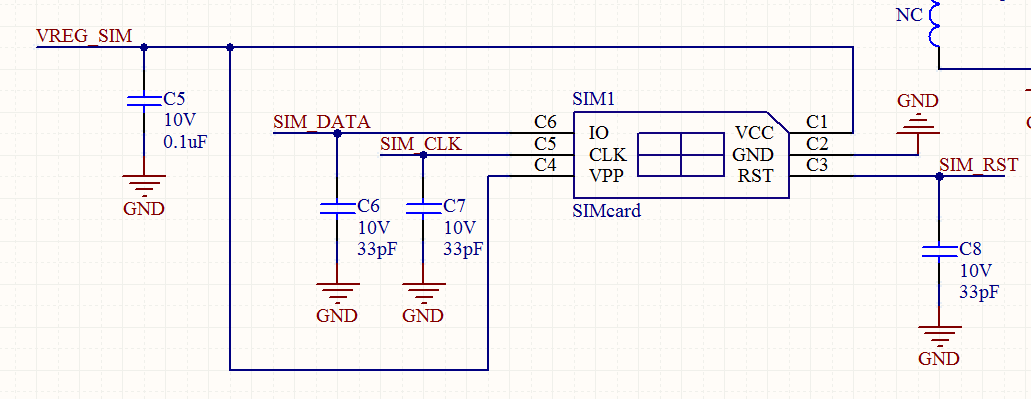


再次执行 pppd call wcdma-dailer & 可以进行网络连接了，证明联网功能已经实现，但是还有个问题



连着连着出现这个，这是因为SIM卡供电可能存在问题.

经过检查发现SIM卡有个引脚必须悬空，不能接电源。



VPP引脚要悬空，否则SIM卡会进入保护模式，这样就无法获取联通的DNS地址。

还有记住要给联通卡加APN，这个在AT指令集文件里面加

APN添加方法

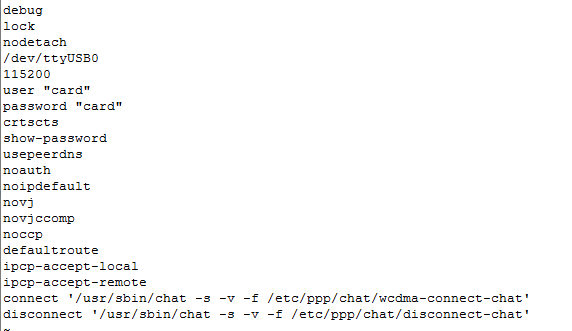


这个pppd联网命令是去执行/etc/ppp/目录下面的wcdma-dailer文件

那我们去查看wcdma-dailer 文件

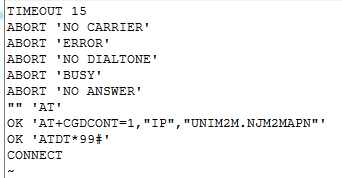


在/etc/ppp/perrs目录下



我们发现connect连接的是/etc/ppp/chat/这个路径的wcdma-connect-char文件

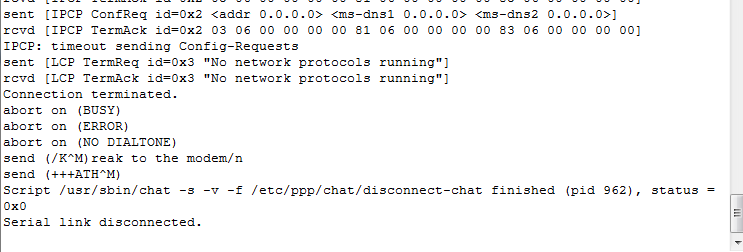
我们去查看这个文件



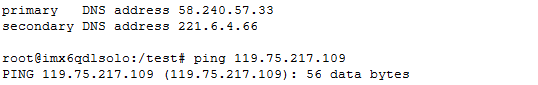
这是联通卡的APN值

在wcdma-connect-char文件里面加入APN。我这里加入的是联通的APN，因为我用的是联通的上网卡。注意格式不要写错。

经过以上测试之后还是出现



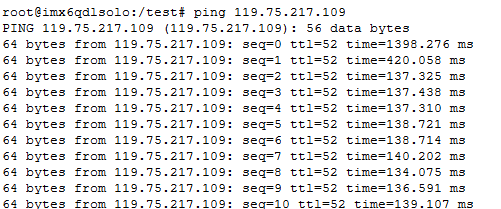
或者出现



卡在这里

这是因为天线没有匹配好造成的，你看DNS都已经获取了。

然后要在硬件上匹配天线。



你看这样我ping百度的IP地址就没有问题的。

还有个问题就是ping域名



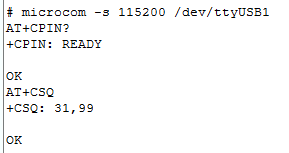
上面能ping IP地址，但是无法ping英文域名

**用AT指令集查看SIM卡信息和信号强度**

microcom –s 115200 /dev/ttyUSB1

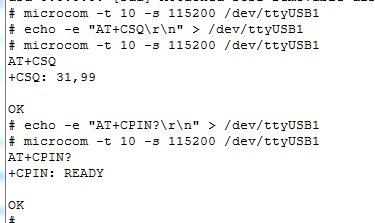
-s 指定波特率 指定3G/4G模块AT串口设备节点

如果这样写，串口就会卡在界面上使用

就只能这样使用，无法退出microrom程序

microcom –t 10 –s 115200 /dev/ttyUSB1

加上-t 就是microcom程序退出时间



READY表示SIM卡插入4G模块

这样你可以先用echo把AT指令集发给4G模块，然后用microcom 加上-t查看

AT指令集返回信息解析

AT+CSQ

<rssi>GSM制式：0-31，99；TD制式：100-199；LTE制式：100-199GSM制式的映射关系：



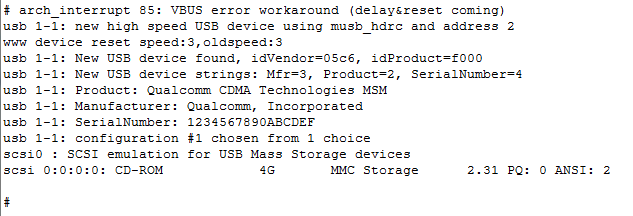
+CSQ:31,99正常信号强度 +CSQ:99,99无信号

**LinuxUSB 4G模块使用**

我用的marvell 88MP1802的4G模块

我在driver/…option.c里面增加了该模块的VID/PID

插入USB4G模块



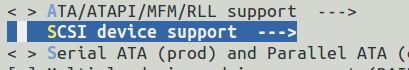
显示的是CD-ROM，证明不是普通的4G模块那样在option.c加VID/PID

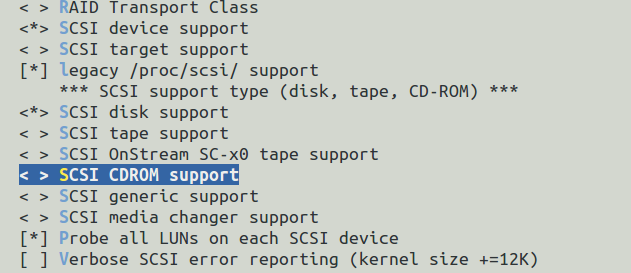
或者用usb-modeswitch就能切换到ttyUSB节点

这种设备要用SCSI命令发送CDB码去让CD-ROM转换成ttyUSB。

但是在/dev下面没有发现sr0，或者sg0之类的设备节点





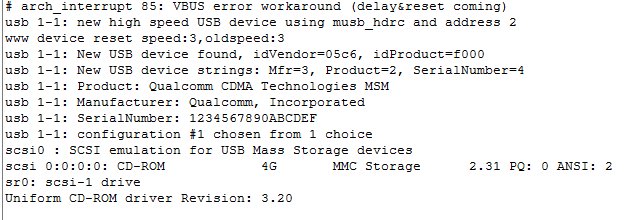


选择这个SCSI CDROM



设置完毕，编译内核，将内核烧录进开发板

再次插入USB4G模块



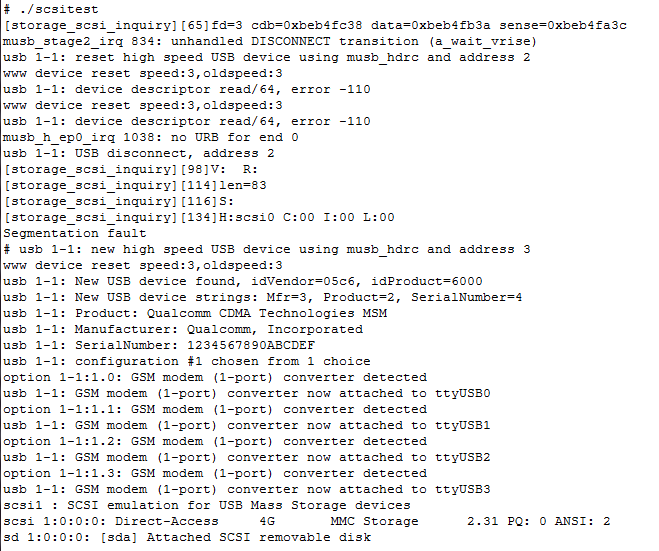
产生了sr0节点

在/dev目录下产生了sr0

这里要写一个scsi的发生命令程序，程序我写在了scsi驱动4G模块章节，在25页

////////////////////////////////////////////////////

执行写好的scsi发送命令程序



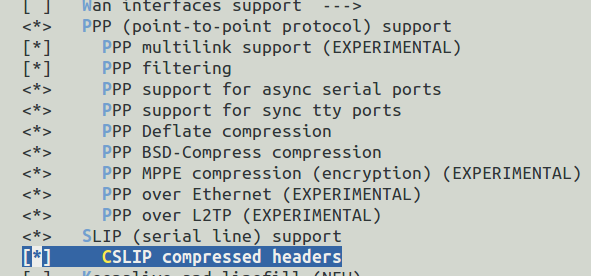
生成了ttyUSB设备节点

在设备节点下产生了ttyUSB，证明4G模块驱动成。

还要记得配置内核的USB转串口，和ppp协议







全部PPP，SLIP，CSLIP都选择上







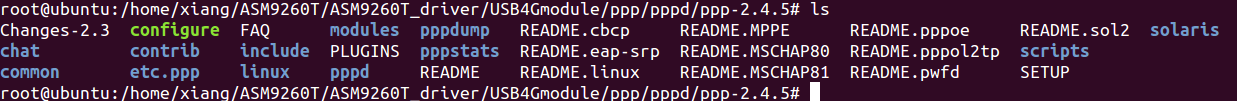
USB转串口全部选择上。

然后make编译成内核，烧录进开发板

**4G模块调试过程在ARM9平台下编译pppd软件**



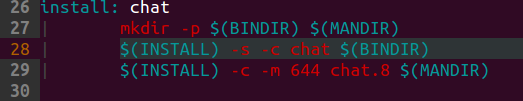
进入ppp-2.4.5目录



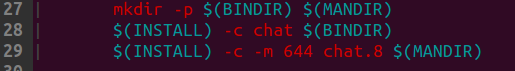
现在我们要将Makefile.linux文件的-s取消掉

ppp-2.4.5/chat/Makefile.linux





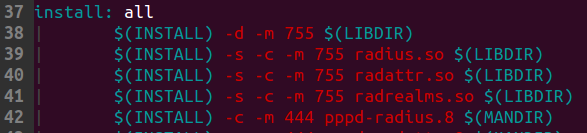
将-s取消掉



-s取消掉就成这样了

然后保存退出

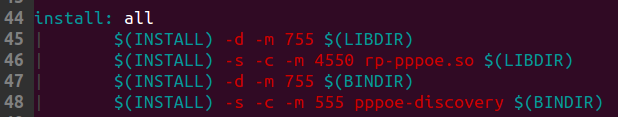
ppp-2.4.5/pppd/plugins/radius/Makefile.linux 将-s取消掉



取消掉-s

然后保存退出

ppp-2.4.5/pppd/plugins/rp-pppoe/Makefile.linux 将-s取消掉



取消掉-s

然后保存退出

ppp-2.4.5/pppd/Makefile.linux 将–s取消掉

取消掉-s

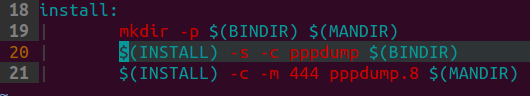




取消掉-s

然后保存退出

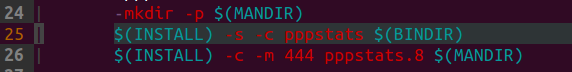
ppp-2.4.5/pppdump/Makefile.linux 将-s取消掉



取消掉-s

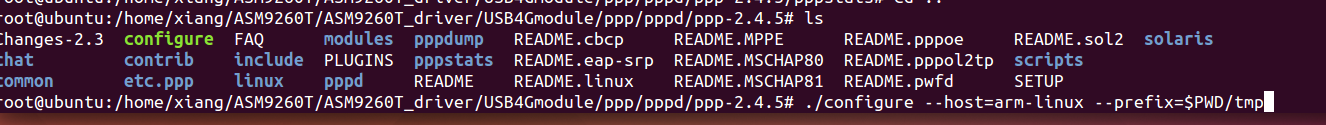
然后保存退出

ppp-2.4.5/pppstats/Makefile.linux 将-s取消掉

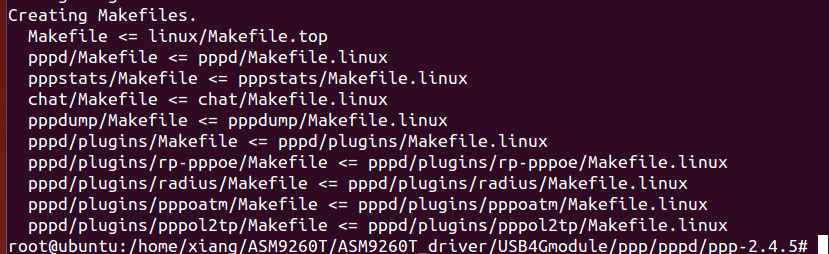


取消掉-s

保存并退出



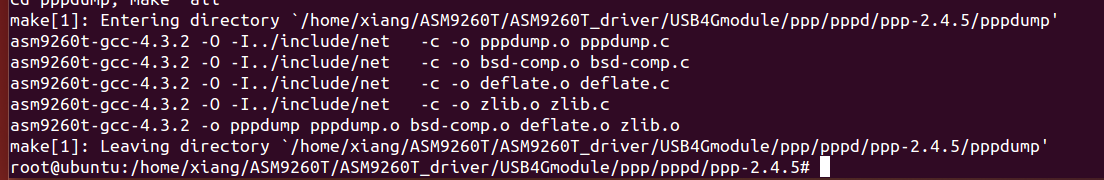
回到顶层目录配置configure，指定安装路径当前目录下tmp，回车



表示配置成功



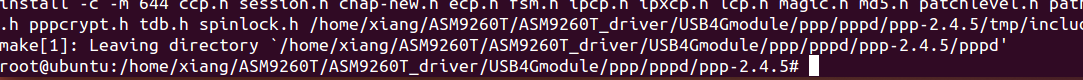
指定交叉编译器编译



编译成功就是这样。

然后在安装make install





安装完成



到tmp目录下

将lib库拷贝到开发板文件系统/usr/lib下，将sbin里面的软件拷贝到开发板文件系统/usr/sbin目录下。

确保开发板文件系统/usr/sbin下面有chat软件



然后在开发板文件系统/var下创建lock和run文件



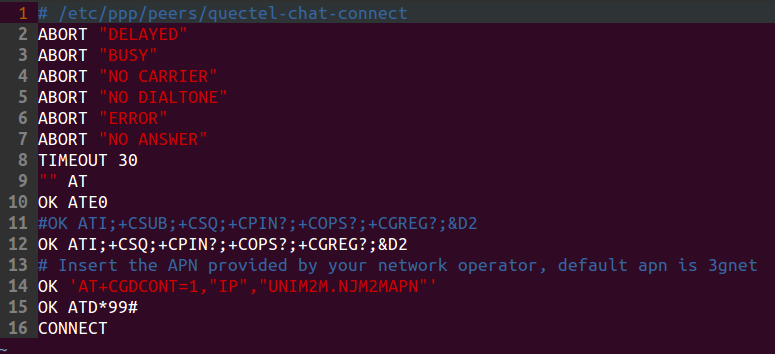
最后添加配置文件

配置文件分为chat和peers目录



chat目录下是这两个文件

wcdma-connect-chat脚本文件内容



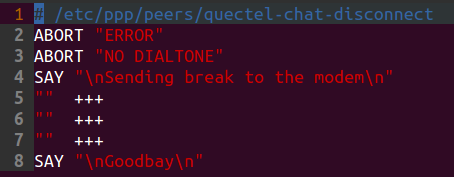
这里有个问题，这个APN是3G模块联通物联网卡的

要取消掉+CSUB



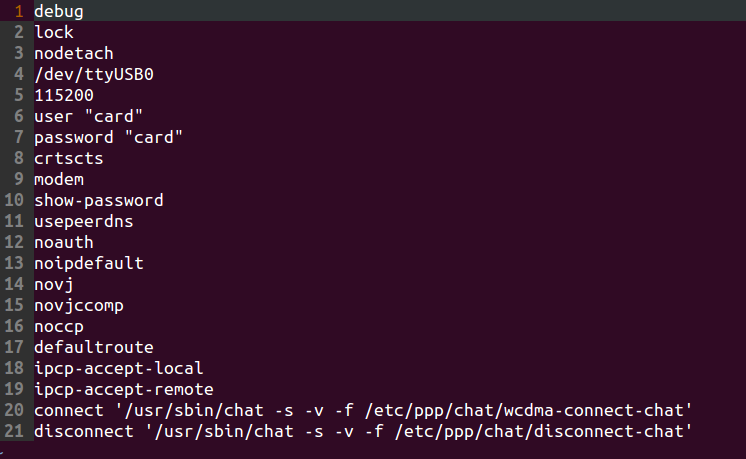
4G模块一定要改成CMNET。不然ping不通IP和域名

disconnect-chat脚本文件内容



然后我们再peers目录下创建拨号的执行文件

wcdma-dailer拨号脚本



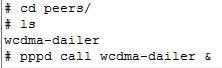
确定你的脚本放在文件系统什么路径下

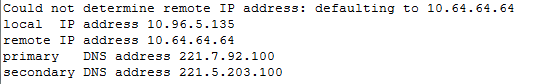
确定chat软件路径

根据4G模块厂家规格确定是ttyUSB几？为数据接口

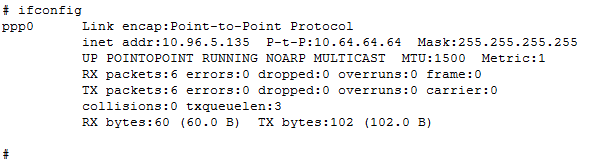
确定chat软件路径

然后将chat和peers目录拷贝到文件系统/etc/ppp目录下，烧写进开发板

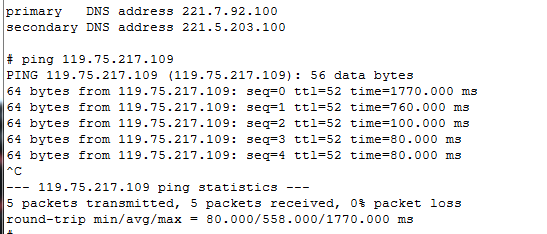
启动开发板执行拨号脚本



最后显示这个证明拨号成功



然后在ifconfig下可以看到ppp0被分配好了IP地址。



ping百度的ip地址没有问题

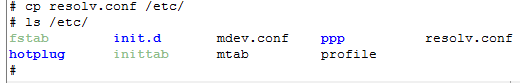


但是ping百度的域名就不行，这是因为你的DNS没有放在/etc目录下

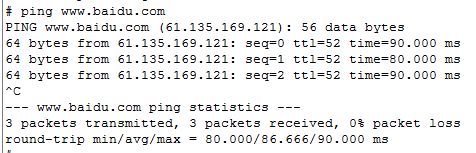
**3G和4G ping百度域名问题**

我们看到域名文件resolv.conf只在ppp目录下

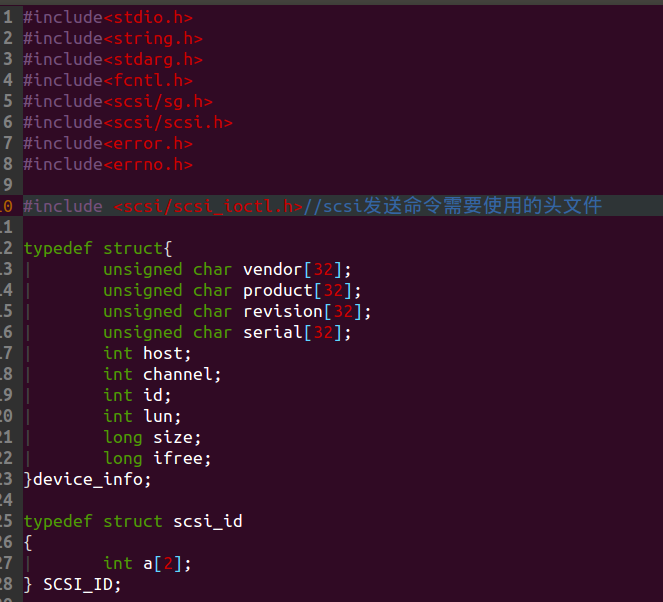




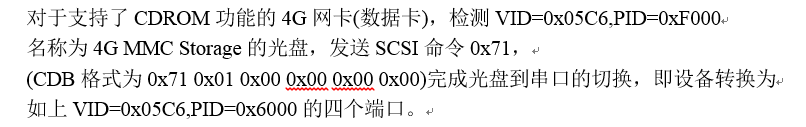
我们将resolv.conf文件拷贝到etc目录下

ping域名就可以了

**SCSI驱动4G模块代码**







就是这样编译执行