**cortex-A8上3G模块的支持**

**作者：陈老师,**[华清远见嵌入式学院](http://www.embedu.org/)讲师。

|  开发主机          VMware ubuntu10.10    |  
        |  移植内核             linux-2.6.22.6            |  
        |   3G模块                 华为 E220           |  
        |    SIM卡                      联通3G                     |   
        |    pppd                       ppp-2.4.4             
        |         |  
        |\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_|

内核配置：

增加内核支持USB 转Serial 驱动

-> Device Drivers                                                               
        x       -> USB support                                                                
        x         -> USB Serial Converter support                                             
        x           -> USB Serial Converter support (USB\_SERIAL [=y])     
                           [\*]USB Generic Serial Driver  
                                      <\*>USB driver for GSM and CDMA modems

增加内核支持PPP协议驱动  
        -> Device Drivers                                                               
        x       -> Network device support                                         
        x         -> Network device support (NETDEVICES [=y])     
                     <\*>   PPP (point-to-point protocol) support                         
                       [\*]     PPP multilink support (EXPERIMENTAL)             
                           [\*]     PPP filtering                                                      
                            <\*>     PPP support for async serial ports                             
                            <\*>     PPP support for sync tty ports                                    
                            <\*>     PPP Deflate compression                                           
                            <\*>     PPP BSD-Compress compression                                         
                            <\*>     PPP MPPE compression (encryption) (EXPERIMENTAL)                     
                            <\*>     PPP over Ethernet (EXPERIMENTAL)

pppd：

下载地址：http://www.linuxfromscratch.org/blfs/view/6.3/basicnet/ppp.html  
        解压  ppp-2.4.4.tar.gz  
        执行脚本产生Makefile等文件：./configure

若为android系统，则需要改变编译方式，需要静态编译；（因为 android c库使用的是Bionic库，不与GNU C库，ucLibc，或任何已知的Linux C库相兼容,所以编译时需要静态编译）

chat:   chat.o  
                $(CC) -o chat chat.o  -static  
        chat.o: chat.c  
                $(CC) -c $(CFLAGS) -o chat.o chat.c  -static

用开发板的交叉编译工具链编译: make CC=arm-unknown-linux-gnueabi-gcc

将ppp-2.4.4/chat/chat      ppp-2.4.4/pppd/pppd  复制到开发板文件系统/usr/sbin

在到开发板文件系统/var/ 下建立run目录

开始编写脚本:

在开发板文件系统里建立/etc/ppp/peers文件夹，下面所创建出的脚本都存放在该路径

1、创建拨号选项脚本： vi  wcdma

1 noauth  
         2 debug  
         3 connect '/usr/sbin/chat -v -f /etc/ppp/peers/wcdma-chat-connect'  
         4 /dev/ttyUSB0  
         5 115200    
         6 defaultroute  
         7 noipdefault  
         8 novj  
         9 novjccomp  
        10 noccp  
        11 ipcp-accept-local  
        12 ipcp-accept-remote  
        13 local  
        15 dump  
        16 nodetach  
        17 nocrtscts  
        18 #usepeerdns

2、创建脚本： vi wcdma-chat-connect

1 "" AT  
        2 OK ATDT  \*99#    
        3 CONNECT

测试：

启动开发板，进入文件系统，插上3G模块（MG3732 ）

提示：

usb 1-1: new full speed USB device using s3c2410-ohci and address 4  
        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  
        libusual: modprobe for usb-storage succeeded, but module is not present

我们使用的是ttyUSB0,虽然有module is not present，但是不影响ping;

调用脚本拨号：  
        # pppd call wcdma &

终端会出现：  
       [root@wuqing/]#pppd options in effect:  
debug           # (from /etc/ppp/peers/wcdma)  
nodetach                # (from /etc/ppp/peers/wcdma)  
dump            # (from /etc/ppp/peers/wcdma)  
noauth          # (from /etc/ppp/peers/wcdma)  
/dev/ttyUSB0            # (from /etc/ppp/peers/wcdma)  
115200          # (from /etc/ppp/peers/wcdma)  
connect /usr/sbin/chat -v -f /etc/ppp/peers/wcdma-chat-connect          # (from                                            /etc/ppp/peers/wcdma)  
nocrtscts               # (from /etc/ppp/peers/wcdma)  
local           # (from /etc/ppp/peers/wcdma)  
novj            # (from /etc/ppp/peers/wcdma)  
novjccomp               # (from /etc/ppp/peers/wcdma)  
ipcp-accept-local               # (from /etc/ppp/peers/wcdma)  
ipcp-accept-remote              # (from /etc/ppp/peers/wcdma)  
noipdefault             # (from /etc/ppp/peers/wcdma)  
defaultroute            # (from /etc/ppp/peers/wcdma)  
noccp           # (from /etc/ppp/peers/wcdma)  
Serial connection established.  
using channel 3  
Using interface ppp0  
Connect: ppp0 <--> /dev/ttyUSB0  
sent [LCP ConfReq id=0x1 <asyncmap 0x0> <magic 0x9793836> <pcomp> <accomp>]  
rcvd [LCP ConfReq id=0x0 <asyncmap 0x0> <auth chap MD5> <magic 0xcd7ea3> <pcomp>                                              <accomp>]  
No auth is possible  
sent [LCP ConfRej id=0x0 <auth chap MD5>]  
rcvd [LCP ConfAck id=0x1 <asyncmap 0x0> <magic 0x9793836> <pcomp> <accomp>]  
rcvd [LCP ConfReq id=0x1 <asyncmap 0x0> <magic 0xcd7ea3> <pcomp> <accomp>]  
sent [LCP ConfAck id=0x1 <asyncmap 0x0> <magic 0xcd7ea3> <pcomp> <accomp>]  
sent [IPCP ConfReq id=0x1 <addr 0.0.0.0>]  
rcvd [LCP DiscReq id=0x2 magic=0xcd7ea3]  
rcvd [IPCP ConfNak id=0x1 <ms-dns1 10.11.12.13> <ms-dns3 10.11.12.14> <ms-wins 10.11.12.13> <ms-wins 10.11.12.14>]  
sent [IPCP ConfReq id=0x2 <addr 0.0.0.0>]  
rcvd [IPCP ConfNak id=0x2 <ms-dns1 10.11.12.13> <ms-dns3 10.11.12.14> <ms-wins 10.11.12.13> <ms-wins 10.11.12.14>]  
sent [IPCP ConfReq id=0x3 <addr 0.0.0.0>]  
rcvd [IPCP ConfNak id=0x3 <ms-dns1 10.11.12.13> <ms-dns3 10.11.12.14> <ms-wins 10.11.12.13> <ms-wins 10.11.12.14>]  
sent [IPCP ConfReq id=0x4 <addr 0.0.0.0>]  
rcvd [IPCP ConfReq id=0x0]  
sent [IPCP ConfNak id=0x0 <addr 0.0.0.0>]  
rcvd [IPCP ConfNak id=0x4 <addr 172.17.15.110>]  
sent [IPCP ConfReq id=0x5 <addr 172.17.15.110>]  
rcvd [IPCP ConfReq id=0x1]  
sent [IPCP ConfAck id=0x1]  
rcvd [IPCP ConfAck id=0x5 <addr 172.17.15.110>]  
Could not determine remote IP address: defaulting to 10.64.64.64  
local  IP address 172.17.15.110  
remote IP address 10.64.64.64

此时用ifconfig命令，会看到ppp0：    
       # ifconfig  
eth0      Link encap:Ethernet  HWaddr 00:00:3E:26:0A:00  
          inet addr:192.168.1.3  Bcast:192.168.1.255  Mask:255.255.255.0  
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1  
          RX packets:3402 errors:53 dropped:53 overruns:0 frame:0  
          TX packets:1742 errors:0 dropped:0 overruns:0 carrier:0  
          collisions:1498 txqueuelen:1000  
          RX bytes:4058678 (3.8 MiB)  TX bytes:663372 (647.8 KiB)  
          Interrupt:53 Base address:0x300

lo        Link encap:Local Loopback  
          inet addr:127.0.0.1  Mask:255.0.0.0  
          UP LOOPBACK RUNNING  MTU:16436  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:0  
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

ppp0      Link encap:Point-to-Point Protocol  
          inet addr:172.17.15.110  P-t-P:10.64.64.64  Mask:255.255.255.255  
          UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1500  Metric:1  
          RX packets:8 errors:0 dropped:0 overruns:0 frame:0  
          TX packets:8 errors:0 dropped:0 overruns:0 carrier:0  
          collisions:0 txqueuelen:3  
          RX bytes:168 (168.0 B)  TX bytes:104 (104.0 B)

看到ppp0 说明拨号成功，现在可以ping一下外网IP 测试

#ping 220.181.111.147 -I ppp0   （需要指明数据出口网卡，有可能默认从eth0出去）   
PING www.baidu.com (220.181.111.147): 56 data bytes  
64 bytes from 220.181.111.147: seq=0 ttl=49 time=397.414 ms  
64 bytes from 220.181.111.147: seq=1 ttl=49 time=345.459 ms  
64 bytes from 220.181.111.147: seq=2 ttl=49 time=340.552 ms  
64 bytes from 220.181.111.147: seq=3 ttl=49 time=355.576 ms  
64 bytes from 220.181.111.147: seq=4 ttl=49 time=330.651 ms  
64 bytes from 220.181.111.147: seq=5 ttl=49 time=355.569 ms  
64 bytes from 220.181.111.147: seq=6 ttl=49 time=330.554 ms  
64 bytes from 220.181.111.147: seq=7 ttl=49 time=335.571 ms  
64 bytes from 220.181.111.147: seq=8 ttl=49 time=360.553 ms

补充：

当域名ping不通时，可能是默认的网关指向eth0的网关  
       虽然ping时指定了数据包的出口-I ppp0 但是使用的路由表却是eth0的路由表

(此为ping不通的情况)  
       # route  
       Kernel IP routing table  
       Destination     Gateway         Genmask         Flags Metric Ref    Use Iface  
       10.64.64.64     \*               255.255.255.255 UH    0      0        0 ppp0  
       192.168.7.0     \*               255.255.255.0   U     0      0        0 eth0  
       default         192.168.7.1     0.0.0.0         UG    0      0        0 eth0

(此为ping通时情况)  
       # route  
       Kernel IP routing table  
       Destination     Gateway         Genmask         Flags Metric Ref    Use Iface  
       10.64.64.64     \*               255.255.255.255 UH    0      0        0 ppp0  
       192.168.7.0     \*               255.255.255.0   U     0      0        0 eth0  
       default         \*               0.0.0.0         U     0      0        0 ppp0

补充2：

嵌入式Linux操作系统下，如果要进行PPP连接，就需要配置拨号要使用的连接脚本。这些脚本参数就决定了PPPD的行为，会对PPP连接产生很大的影响。和PPP连接关系密切的有两个脚本是wcdma-chat-connect'和wcdma 两个脚本。其中，wcdma-chat-connect脚本是用来进行AT呼叫和控制的脚本，而wcdma脚本就影响PPP连接的。

1.  wcdma-chat-connect脚本

一个简单的wcdma-chat-connect脚本有下面的结构：

""              AT  
        OK             ATDT  dialnumber  
        CONNECT      ""

一个wcdma-chat-connect脚本都是由字符串对来组成的。前面的字符串都是期望获取的串，紧跟的是发送的字符串。这个脚本每一项的具体含义是这样的：

1)   MODEM期望空字符串。这句话的意思直接理解就是MODEM不管收到什么字符串，先发出字符串AT；  
        2)   期望收到“OK”字符串，然后发送字符串“ATDT dialnumber”  
        3)   如果收到“CONNECT”，就不再发送，认为数据链路已建立连接。

这样的wcdma-chat-connect脚本是最简单的，如果需要进行超时控制，就可以加入如下字段：

TIMEOUT 10

如果要增加对特殊情况的处理，就加入下面字段：  
        ABORT           BUSY            
        ABORT           NO ANSWER   
        ABORT           RINGING

这三行语句的意思是：如果收到字符串“BUSY”、“NO ANSWER”、“RINGING”就退出执行。

所以在考虑到各种特殊情况下，配置一个PPP连接的wcdma-chat-connect脚本就可以像下面这样：

TIMEOUT         30  
        ABORT           BUSY            
       ABORT           NO ANSWER   
       ABORT           RINGING  
        ""                 AT  
        OK                ATDT dialnumber  
        CONNECT         ""

2. wcdma脚本

OPTIONS脚本的内容，为PPP连接指定了连接使用的设备、使用的控制字符传输速率、指定了硬件加速、溢出控制等。

例如下面的wcdma脚本：

ttyS0 ---- 指定连接使用的设备，例如：ttyS0、ttyS1等  
        57600 ---- 设置连接使用的控制字符传输速率，可以设置为57600、115200等  
        debug ---- 如果需要加入调试信息，就加入参数debug  
        logfile /var/ ppplog ---- 将连接过程中的信息输入到某个文件中  
        mtu 1500  
        -detach  
        noipdefault ---- 不使用默认IP就可以加入参数noipdefault  
        defaultroute  
        usepeerdns ---- 使用服务器端协商的DNS就可以设置参数usepeerdns  
        lcp-echo-failure 4 ---- 当连续4次没有收到发出的ＬＣＰ回声请求时，就认为服务器端已不再响应，就退出执行。这里的失败次数可以灵活来决定。  
        -ccp ---- 不使用压缩控制协议  
        -vj ---- 关掉式IP头压缩  
        -chap ---- 不使用chap鉴权  
        -mschap-v2 ---- 不使用mschap鉴权  
        user  
        hide-password  
        connect "/usr/bin/chat -v -t6 -f /var/ chat" ---- 制定了要使用的chat脚本的位置。加上参数-v 告诉 chat命令将其所有的输出／入拷贝到系统记录里（通常是 /var/log/messages）。-t 6 指定了执行chat该命令的时间为6s。chat脚本的位置可以位于/etc/目录下，也可以位于/var下，这个可以更加需要灵活设置。  
        persist --- 永久链接(自动重拨)  
        crtscts --- 告诉ppp使用modem的硬件流量控制  
        modem --- 使ppp使用DCD信号来判断连接是否正常，有无掉线现象  
        deflate --- 使pppd使用defalte压缩方式  
        idle --- 设置了一个时间限制，当在300秒的时间内没有数据传送，就断开连接  
        lock --- 则创建一个锁定文件，其他程序在发现存在这个文件后，就能得知相应的串口已经被使用。  
        demond --- 参数告诉pppd停留在后台，监视网络数据，一旦有要求就立即进行连网，超时后就断开连接，但pppd仍然停留在后台等待下次数据传送

其他的参数具体含义可以参照参考PPPD2.4.4 中options.c文件中注释。