

# **FIBOCOM\_4GFamily\_Linux 驱动程序 &PPP 拨号应用设计说明**

Version: V1.0.1

更新日期: 2015-03-30


---

# 版权声明

版权所有©深圳市广和通无线股份有限公司 2014。保留一切权利。

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

# 商标申明

 为深圳市广和通无线股份有限公司的注册商标，由所有人拥有。

# 注意

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

# 版本记录

Version	Date	Remarks
V1.0.0	2015-03-24	Initial Release
V1.0.1	2015-03-30	去除 ppp 拨号相关内容，添加 ncm 相关配置

## 目录

1 添加 CDC ACM 所需内核驱动配置 .....	4
1.1 修改内核编译配置(kernel 根目录下的.config 文件中), 确保下面的配置项已经被选定: .....	4
1.2 详细操作 .....	4
2 添加 CDC NCM 所需要内核驱动配置 .....	9
2.1 修改内核编译配置(kernel 根目录下的.config 文件中), 确保下面的配置项已经被选定: .....	9
2.2 详细操作 .....	9
3 如何确认 NCM/ACM 驱动已经配置入系统 .....	12
4 拨号 AT 发送 .....	13
5 网络配置 .....	15

# 1 添加 CDC ACM 所需内核驱动配置

## 1.1 修改内核编译配置(kernel 根目录下的.config 文件中), 确保下面的配置项已经被选定:

– PPP 拨号的相关配置项:

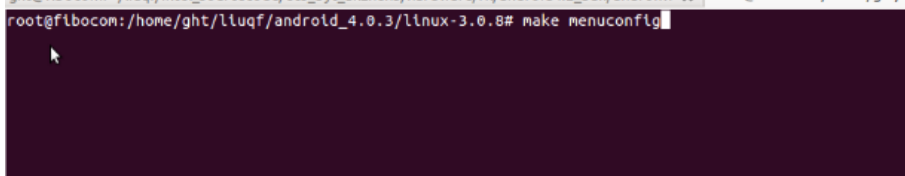
```
CONFIG_PPP=y  
CONFIG_PPP_MULTILINK=y  
CONFIG_PPP_FILTER=y  
CONFIG_PPP_ASYNC=y  
CONFIG_PPP_SYNC_TTY=y  
CONFIG_PPP_DEFLATE=y  
CONFIG_PPP_BSDCOMP=y
```

– USB ACM 相关配置项:

```
CONFIG_USB_ANNOUNCE_NEW_DEVICES=y (此选项存在的情况建议配置一下, 没有请忽略)  
CONFIG_USB_ACM=y
```

## 1.2 详细操作

打开 Terminal 工具, 进入 kernel 目录(假定为: /home/ght/liuqf/android-4.0.3/linux-3.0.8/), 然后执行 make <configuration>命令, 在本文中, 假定使用标准 make menuconfig)



按照下列图指引完成配置

– PPP 拨号的相关配置项:

## Linux Kernel Configuration

Arrow keys navigate the menu. <Enter> selects submenus --->. Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </> for Search. Legend: [\*] built-in [ ] excluded <M> module < > module capable

```
General setup --->
[*] Enable loadable module support --->
[*] Enable the block layer --->
System Type --->
Bus support --->
Kernel Features --->
Boot options --->
CPU Power Management --->
Floating point emulation --->
Userspace binary formats --->
Power management options --->
[*] Networking support --->
  Device Drivers --->
  File systems --->
  Kernel hacking --->
  Security options --->
  < > Cryptographic API --->
  Library routines --->
---
Load an Alternate Configuration File
```

v(+)

<Select> < Exit > < Help >

## Device Drivers

Arrow keys navigate the menu. <Enter> selects submenus --->. Highlighted letters are hotkeys. Pressing <Y> includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </> for Search. Legend: [\*] built-in [ ] excluded <M> module < > module capable

```
^(-)
< > Parallel port support --->
[ ] Block devices --->
[*] Misc devices --->
< > ATA/ATAPI/MFM/RLL support --->
  SCSI device support --->
  < > Serial ATA (prod) and Parallel ATA (experimental) drivers --->
  [ ] Multiple devices driver support (RAID and LVM) --->
  [ ] Fusion MPT device support --->
  IEEE 1394 (FireWire) support --->
< > I2O device support --->
[*] Network device support --->
  [ ] ISDN support --->
  < > Telephony support --->
  Input device support --->
  Character devices --->
<*> I2C support --->
[*] SPI support --->
  PPS support --->
  -*- GPIO Support --->
  < > Dallas's 1-wire support --->
```

v(+)

<Select> < Exit > < Help >

```
Network device support
Arrow keys navigate the menu. <Enter> selects submenus --->. Highlighted letters are hotkeys. Pressing
<Y> includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </> for
Search. Legend: [*] built-in [ ] excluded <M> module <> module capable

^(-)
*** Enable WiMAX (Networking options) to see the WiMAX drivers ***
USB Network Adapters --->
[ ] Wan interfaces support --->
<> FDDI driver support
[ ] HIPPI driver support (EXPERIMENTAL)
<*> 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) (NEW)
<> PPP over Ethernet (EXPERIMENTAL) (NEW)
<> PPP over L2TP (EXPERIMENTAL) (NEW)
<> PPP on L2TP Access Concentrator (NEW)
<> PPP on PPTP Network Server (NEW)
<> SLIP (serial line) support
[ ] Fibre Channel driver support
<> Network console logging support (EXPERIMENTAL)

<Select> < Exit > < Help >
```

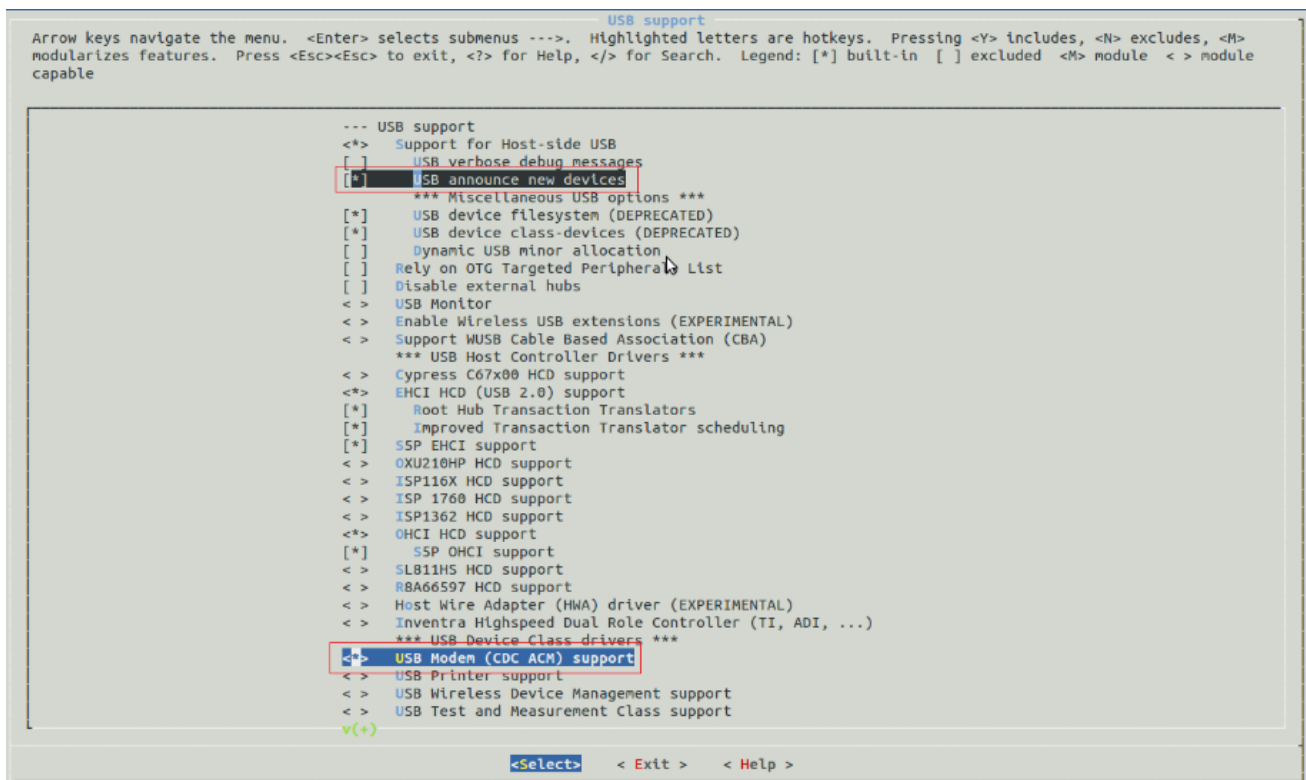
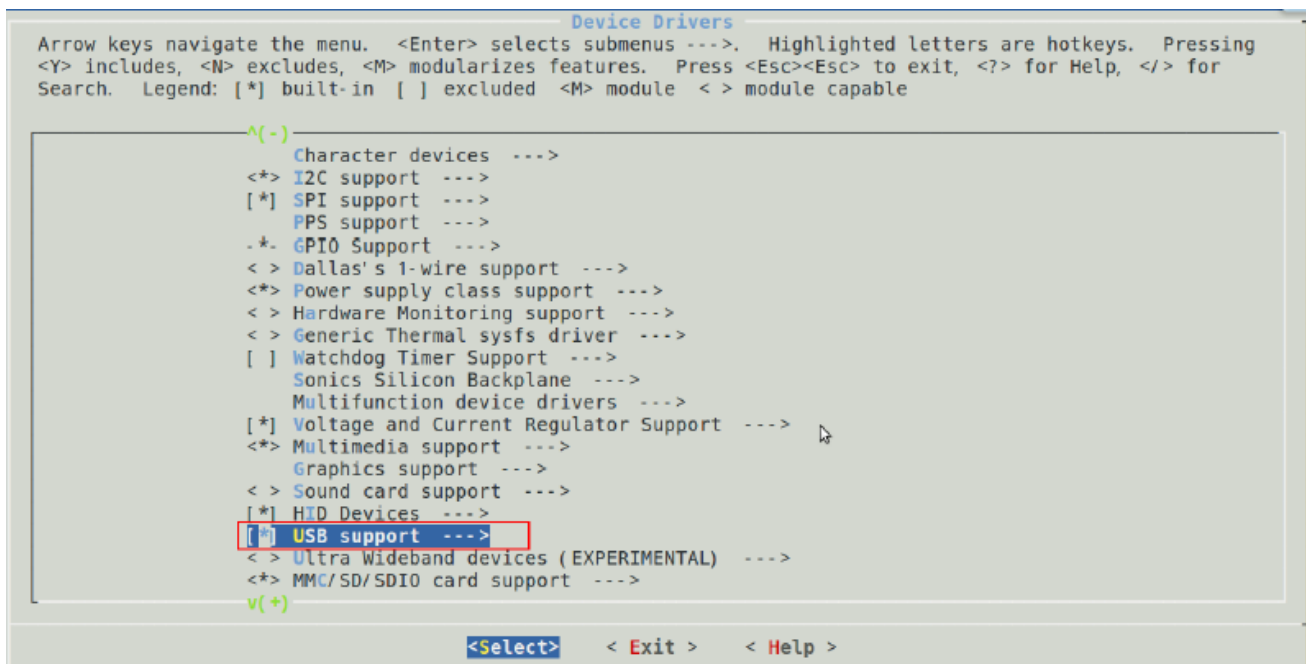
– USB ACM 相关配置项:

```
Linux Kernel Configuration
Arrow keys navigate the menu. <Enter> selects submenus --->. Highlighted letters are hotkeys. Pressing
<Y> includes, <N> excludes, <M> modularizes features. Press <Esc><Esc> to exit, <?> for Help, </> for
Search. Legend: [*] built-in [ ] excluded <M> module <> module capable

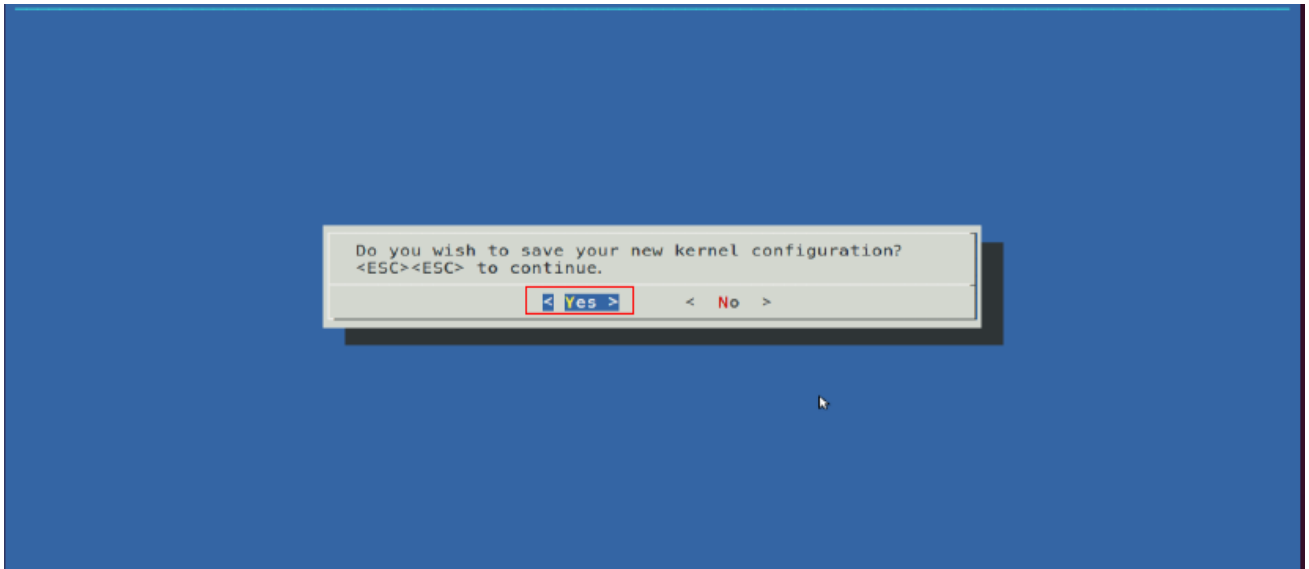
General setup --->
[*] Enable loadable module support --->
[*] Enable the block layer --->
System Type --->
Bus support --->
Kernel Features --->
Boot options --->
CPU Power Management --->
Floating point emulation --->
Userspace binary formats --->
Power management options --->
[*] Networking support --->
[*] Device Drivers --->
File systems --->
Kernel hacking --->
Security options --->
<> Cryptographic API --->
Library routines --->
---
Load an Alternate Configuration File

v(+)

<Select> < Exit > < Help >
```



如上操作选完所需选项后，通过选择<Exit>按钮，逐层退出各个配置界面。最后在保存配置界面中，选择<Yes>选项并退出。



完成配置后，即可运行 **make** 命令，开始编译修改后的内核。



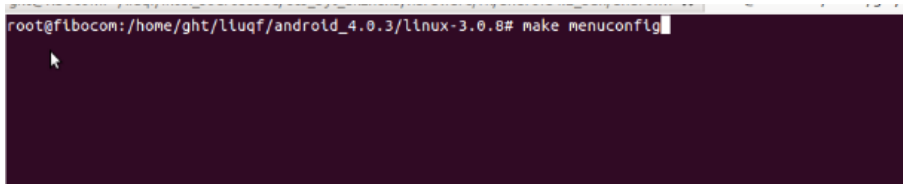
## 2 添加 CDC NCM 所需要内核驱动配置

### 2.1 修改内核编译配置(kernel 根目录下的.config 文件中), 确保下面的配置项已经被选定:

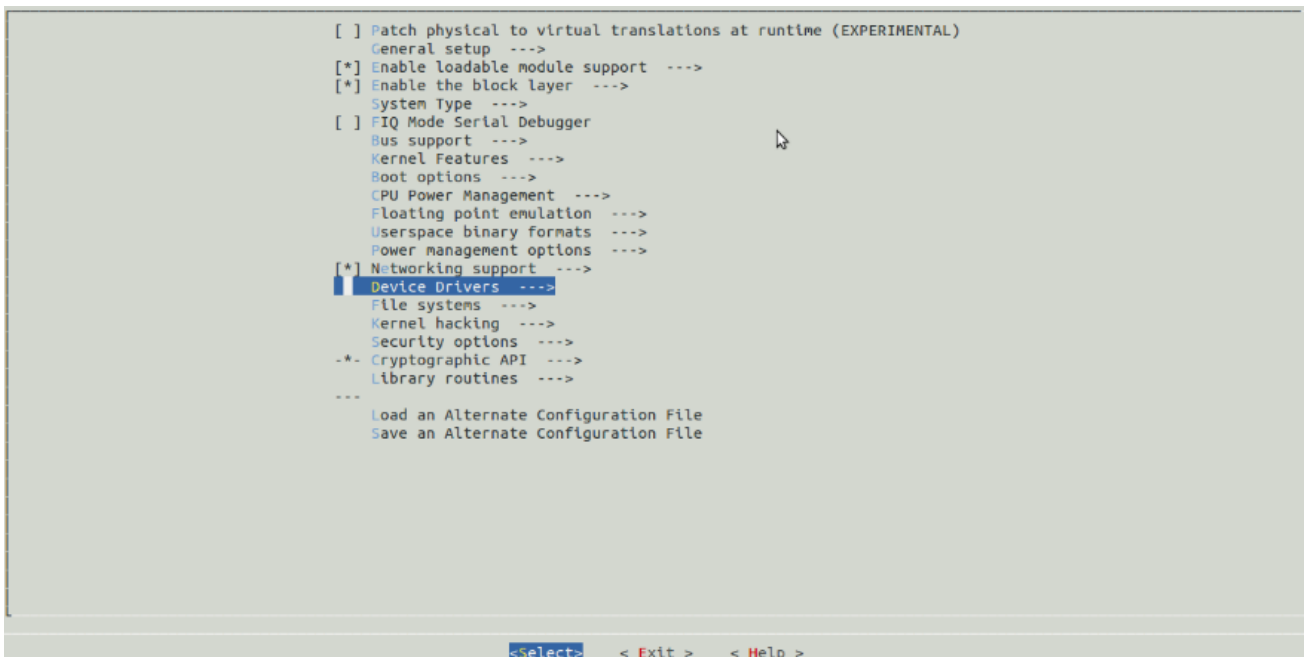
```
CONFIG_USB_USBNET=y  
CONFIG_NETDEVICES=y  
CONFIG_USB_NET_CDC_NCM=y
```

### 2.2 详细操作

打开 Terminal 工具, 进入 kernel 目录(假定为: /home/ght/liuqf/android-4.0.3/linux-3.0.8/), 然后执行 make <configuration>命令, 在本文中, 假定使用标准 make menuconfig)



CDC ECM 驱动配置项, 按照下列图指引完成配置:



```

    Generic Driver Options --->
< > Connector - unified userspace <-> kernel space linker --->
<*> Memory Technology Device (MTD) support --->
< > Parallel port support --->
[*] Block devices --->
[*] Misc devices --->
< > ATA/ATAPI/MFM/RLL support (DEPRECATED) --->
    SCSI device support --->
< > Serial ATA and Parallel ATA drivers --->
[*] Multiple devices driver support (RAID and LVM) --->
< > Generic Target Core Mod (TCM) and ConfigFS Infrastructure --->
[*] Network device support --->
[ ] ISDN support --->
< > Telephony support --->
    Input device support --->
    Character devices --->
<*> I2C support --->
[*] SPI support --->
    PPS support --->
    PTP clock support --->
-* GPIO Support --->
< > Dallas's 1-wire support --->
<*> Power supply class support --->
<*> Hardware Monitoring support --->
< > Generic Thermal sysfs driver --->
[*] Watchdog Timer Support --->
    Sonics Silicon Backplane --->
    Broadcom specific AMBA --->
[*] Multifunction device drivers --->
[*] Voltage and Current Regulator Support --->
<*> Multimedia support --->
    Graphics support --->
v(+)

```

<select> < Exit > < Help >

```

--- Network device support
<*> Intermediate Functional Block support
< > Dummy net driver support
< > Bonding driver support
< > MAC-VLAN support (EXPERIMENTAL)
< > EQL (serial line load balancing) support
<*> Universal TUN/TAP device driver support
< > Virtual ethernet pair device
-* Generic Media Independent Interface device support
<*> PHY Device support and infrastructure --->
    Ethernet (10 or 100Mbit) --->
    [ ] Ethernet (1000 Mbit) --->
    [ ] Ethernet (10000 Mbit) --->
    [*] Wireless LAN --->
    *** Enable WiMAX (Networking options) to see the WiMAX drivers ***
    [*] USB Network Adapters --->
    [ ] Wan interfaces support --->
    *** CAIF transport drivers ***
<*> 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)
<*> PPP on L2TP Access Concentrator
<*> PPP on PPTP Network Server
< > SLIP (serial line) support
< > Network console logging support

```

<select> < Exit > < Help >

```

< > USB CATC NetMate-based Ethernet device support (EXPERIMENTAL)
< > USB KLSI KLSUSB101-based ethernet device support
< > USB Pegasus/Pegasus-II based ethernet device support
< > USB RTL8150 based ethernet device support (EXPERIMENTAL)
<+> Multi-purpose USB Networking Framework
<+> ASIX AX88xxx Based USB 2.0 Ethernet Adapters
-+> CDC Ethernet support (smart devices such as cable modems)
< > CDC EEM support
<+> CDC NCM support
< > Davicom DM9601 based USB 1.1 10/100 ethernet devices
< > SMSC LAN75XX based USB 2.0 gigabit ethernet devices
< > SMSC LAN95XX based USB 2.0 10/100 ethernet devices
< > GeneSys GL620USB-A based cables
<+> NetChip 1080 based cables (Laplink, ...)
< > Prolific PL-2301/2302/25A1 based cables
< > MosChip MCS7830 based Ethernet adapters
< > Host for RNDIS and ActiveSync devices (EXPERIMENTAL)
<+> Simple USB Network Links (CDC Ethernet subset)
[ ] ALi M5632 based 'USB 2.0 Data Link' cables
[ ] AnchorChips 2720 based cables (Xircom PGUNET, ...)
[*] eTEK based host-to-host cables (Advance, Belkin, ...)
[*] Embedded ARM Linux links (iPaq, ...)
[ ] Epson 2888 based firmware (DEVELOPMENT)
[ ] KT Technology KC2190 based cables (InstaNet)
<+> Sharp Zaurus (stock ROMs) and compatible
< > Conexant CX82310 USB ethernet port
< > Samsung Kalmia based LTE USB modem
< > Option USB High Speed Mobile Devices
< > Intellon PLC based usb adapter
< > CDC Phonet support
< > Apple iPhone USB Ethernet driver
< > USB-to-WWAN Driver for Sierra Wireless modems
v(+)

```

<Select> < Exit > < Help >

```

< > USB CATC NetMate-based Ethernet device support (EXPERIMENTAL)
< > USB KLSI KLSUSB101-based ethernet device support
< > USB Pegasus/Pegasus-II based ethernet device support
< > USB RTL8150 based ethernet device support (EXPERIMENTAL)
<+> Multi-purpose USB Networking Framework
<+> ASIX AX88xxx Based USB 2.0 Ethernet Adapters
-+> CDC Ethernet support (smart devices such as cable modems)
< > CDC EEM support
<+> CDC NCM support
< > Davicom DM9601 based USB 1.1 10/100 ethernet devices
< > SMSC LAN75XX based USB 2.0 gigabit ethernet devices
< > SMSC LAN95XX based USB 2.0 10/100 ethernet devices
< > GeneSys GL620USB-A based cables
<+> NetChip 1080 based cables (Laplink, ...)
< > Prolific PL-2301/2302/25A1 based cables
< > MosChip MCS7830 based Ethernet adapters
< > Host for RNDIS and ActiveSync devices (EXPERIMENTAL)
<+> Simple USB Network Links (CDC Ethernet subset)
[ ] ALi M5632 based 'USB 2.0 Data Link' cables
[ ] AnchorChips 2720 based cables (Xircom PGUNET, ...)
[*] eTEK based host-to-host cables (Advance, Belkin, ...)
[*] Embedded ARM Linux links (iPaq, ...)
[ ] Epson 2888 based firmware (DEVELOPMENT)
[ ] KT Technology KC2190 based cables (InstaNet)
<+> Sharp Zaurus (stock ROMs) and compatible
< > Conexant CX82310 USB ethernet port
< > Samsung Kalmia based LTE USB modem
< > Option USB High Speed Mobile Devices
< > Intellon PLC based usb adapter
< > CDC Phonet support
< > Apple iPhone USB Ethernet driver
< > USB-to-WWAN Driver for Sierra Wireless modems
v(+)

```

<Select> < Exit > < Help >

如上操作选完所需选项后，通过选择<Exit>按钮，逐层退出各个配置界面。最后在保存配置界面中，选择<Yes>并退出。

Do you wish to save your new configuration ? <ESC><ESC>  
to continue.

<Yes> < No >

### 3 如何确认 NCM/ACM 驱动已经配置入系统

开机启动时，执行 dmesg 命令，查看内核 LOG，发现红框信息即说明 NCM 驱动已经配置入系统

```
<6>[ 1.492528] eth0: dm9000a at e0838000,e083c00c IRQ 39 MAC: 08:90:00:a0:02:10 (platform data)
<6>[ 1.500005] usbcore: registered new interface driver asix
<6>[ 1.505217] usbcore: registered new interface driver cdc_ether
<6>[ 1.511027] usbcore: registered new interface driver net1080
<6>[ 1.516654] usbcore: registered new interface driver cdc_subset
<6>[ 1.522542] usbcore: registered new interface driver zaurus
<6>[ 1.527994] cdc_ncm: 04-Aug-2011
<6>[ 1.531260] usbcore: registered new interface driver cdc_ncm
<6>[ 1.536918] sdhci: Secure Digital Host Controller Interface driver
<6>[ 1.542075] sdhci: Copyright(c) Pierre Ossman
```

系统启动完全后，L810 模块上电开机，再执行 dmesg 命令，查看内核 LOG，发现红框信息说明 L810 NCM 驱动已经加载 OK，并且生成 usb0 usb1 usb2 usb3 等 NCM 网口。

```
<6>[71820.808950] usb 1-1.2: new high speed USB device number 8 using s5p-ehci
<6>[71820.928605] usb 1-1.2: New USB device found, idVendor=1519, idProduct=0443
<6>[71820.928669] usb 1-1.2: New USB device strings: Mfr=1, Product=2, SerialNumber=3
<6>[71820.928731] usb 1-1.2: Product: 3 CDC-ACM + 4 CDC-NCM
<6>[71820.928775] usb 1-1.2: Manufacturer: Comneon
<6>[71820.929750] usb 1-1.2: SerialNumber: 865204020007441
<3>[71820.960903] cdc_acm 1-1.2:1.0: This device cannot do calls on its own. It is not a modem.
<6>[71820.961505] cdc_acm 1-1.2:1.0: ttyACM0: USB ACM device
<3>[71820.967194] cdc_acm 1-1.2:1.2: This device cannot do calls on its own. It is not a modem.
<6>[71820.967753] cdc_acm 1-1.2:1.2: ttyACM1: USB ACM device
<3>[71820.972342] cdc_acm 1-1.2:1.4: This device cannot do calls on its own. It is not a modem.
<6>[71820.973884] cdc_acm 1-1.2:1.4: ttyACM2: USB ACM device
<6>[71820.994237] usb 1-1.2: MAC-Address: 0x00:0x00:0x11:0x12:0x13:0x14
<6>[71821.001998] cdc_ncm 1-1.2:1.6: usb0: register 'cdc_ncm' at usb-s5p-ehci-1.2, CDC NCM, 00:00:11:12:13:14
<6>[71821.020104] usb 1-1.2: MAC-Address: 0x00:0x00:0x11:0x12:0x13:0x16
<6>[71821.025785] cdc_ncm 1-1.2:1.8: usb1: register 'cdc_ncm' at usb-s5p-ehci-1.2, CDC NCM, 00:00:11:12:13:16
<6>[71821.040494] usb 1-1.2: MAC-Address: 0x00:0x00:0x11:0x12:0x13:0x18
<6>[71821.048300] cdc_ncm 1-1.2:1.10: usb2: register 'cdc_ncm' at usb-s5p-ehci-1.2, CDC NCM, 00:00:11:12:13:18
<6>[71821.066349] usb 1-1.2: MAC-Address: 0x00:0x00:0x11:0x12:0x13:0x1a
<6>[71821.072507] cdc_ncm 1-1.2:1.12: usb3: register 'cdc_ncm' at usb-s5p-ehci-1.2, CDC NCM, 00:00:11:12:13:1a
```

执行 netcfg 命令可以查询到有 usb0 usb1 usb2 usb3 等网口

```
/ # netcfg
lo UP 127.0.0.1/8 0x00000049 00:00:00:00:00:00
ifb0 DOWN 0.0.0.0/0 0x00000082 ca:98:28:4b:1d:a8
ifb1 DOWN 0.0.0.0/0 0x00000082 72:1a:49:e8:ec:14
eth0 UP 0.0.0.0/0 0x00001003 08:90:00:a0:02:10
sit0 DOWN 0.0.0.0/0 0x00000080 00:00:00:00:00:00
ip6tnl0 DOWN 0.0.0.0/0 0x00000080 00:00:00:00:00:00
usb0 DOWN 0.0.0.0/0 0x00001002 00:00:11:12:13:14
usb1 DOWN 0.0.0.0/0 0x00001002 00:00:11:12:13:16
usb2 DOWN 0.0.0.0/0 0x00001002 00:00:11:12:13:18
usb3 DOWN 0.0.0.0/0 0x00001002 00:00:11:12:13:1a
/ #
```

执行 ls /dev/ttyACM\* 命令可以查询到有 3 个通信端口 ttyACM0 ttyACM1 ttyACM2，即表示 ACM 驱动加载 OK

```
/ # ls /dev/ttyACM*
/dev/ttyACM0
/dev/ttyACM1
/dev/ttyACM2
```

## 4 拨号 AT 发送

拨号 AT 命令发送:

### 1. 查询信号 SIM 卡状态 及 注网状态

AT+CSQ, 参数 1 的范围是 0-31 或者 99, 如果是 99 表示无信号, 请检查天线情况, 参数 2 可以不关注

```
write(8): AT+CSQ
read(26): AT+CSQ
+CSQ: 5,2
OK
```

AT+CPIN?, 检查 SIM 卡状态, 返回 READY 状态, 表示 SIM 卡可用, 如果返回 SIM PIN 状态, 请用 AT+CPIN="correct PIN" 解 pin

```
read(31): at+cpin?
+CPIN: READY
OK
```

AT+COPS? 查询运营商选择及注网情况, 如果仅有一个参数返回的话, 请检查天线, SIM 卡状态是否正常。

```
read(46): AT+COPS?
+COPS: 0,0,"CHINA MOBILE",7
OK
```

参数 1 表示, 注册模式, 0 表示自动, 1 表示手动

参数 2 表示, 显示格式, 0 表示长字符串格式, 1 表示短字符串格式, 2 表示字数

参数 3 表示, 按参数 2 来显示运营商名字, CHINA MOBILE 为中国移动

参数 4 表示, 注网情况, 7 表示 LTE 网, 2 表示 UMTS, 0 表示 GSM

### 2. 拨号 AT 发送

在第 1 步骤检查完信号, SIM 卡状态及运营商注网情况都正常下, 需要进行拨号 AT 发送, 命令发送顺序如下:

AT+CGACT=0, 1, 先去激活 PDP 上下文, 以确保不重得激活 PDP 上下文

AT+CGDCONT=1,"ip","\cmnet", 定义 PDP 上下文, 参数 3 为 APN 类型, 联通卡请设置为 3gnet

AT+CGACT=1,1, 激活 PDP 上下文

AT+XDNS=1,1, 使用 DNS 地址查询命令

AT+XDATACHANNEL=1,1,"\USBCDC/2","\USBHS/NCM/0",0, 配置数据通道, 参数 4 为 AT 通道, 参数 5 为数据通道

AT+CGDATA="M-Raw\_IP",1, 输入数据状态，发命令发送成功后，模块会触发主机发送 DHCP 过程。至此拨号 AT 发送完毕，如果任意一步返回失败，请间隔 1 秒后再重发此 AT，如果重发 3 次依然不成功，请从第 2 节的第一条 AT 命令重新发送拨号 AT。

### 3. Ip 及 dns 地址查询

AT+CGDCONT?, 参数 4 将返回 ip 地址，如果返回 0.0.0.0 表示步骤 2 拨号失败，需要重新进行步骤 2.

```
read(92): AT+CGDCONT?  
+CGDCONT: 1,"IP","cmnet.mnc000.mcc460.gprs","10.39.144.88",0,0,0,0,0,0  
OK
```

AT+XDNS?, 参数 2 跟参数 3 分别为首选及次选 DNS 地址。

```
write(10): AT+XDNS?  
read(60): AT+XDNS?  
+XDNS: 1, "221.179.38.7", "120.196.165.7"  
OK
```

**注意：AT+CGACT 命令的返回有可能需要比较长的时间**

## 5 网络配置

根据 XDNS?查询到的 dns 地址，写入到/etc/resolv.conf 文件中

```
echo "nameserver 221.179.38.7" >> /etc/resolv.conf
```

```
echo "nameserver 120.196.165.7" >> /etc/resolv.conf
```

根据 CGDCONT?查询到的 ip 地址，设置主机 ip 及路由，其中\$1 为 ip 地址，\$2 为网关地址，**网关地址可以跟 ip 地址一样，也可以是 ip 地址的下一跳。**

```
ifconfig usb0 $1 netmask 255.255.255.255 -arp
```

```
ip r add $2 dev usb0
```

```
ip r add 0.0.0.0/0 via $2 dev usb0
```

设置完后请用 ping [www.baidu.com](http://www.baidu.com) 查询主机网络是否已经正确。