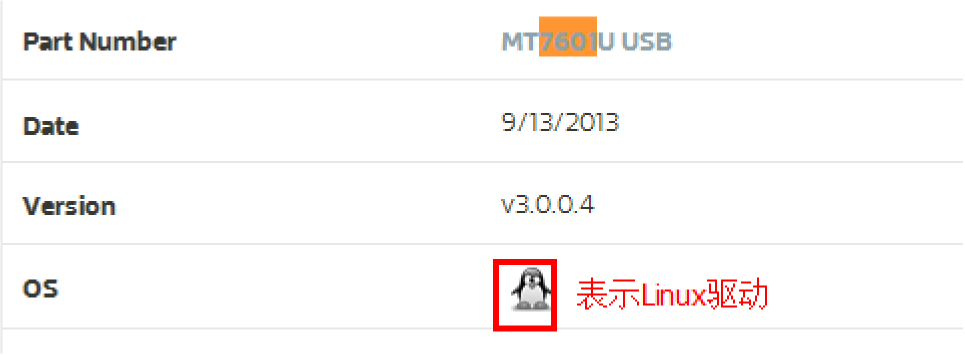
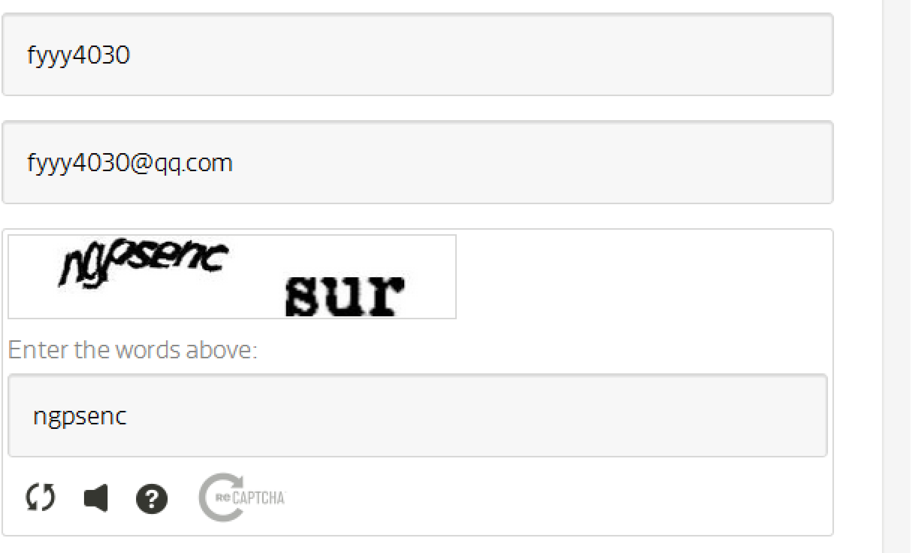
360 WIFI 2 随身 WIFI 驱动移植

1． 下载无线网卡驱动 ：http://www.mtk.com.tw/en/downloads/



Part Number	MT7601U USB
Date	9/13/2013
Version	v3.0.0.4
OS	表示Linux驱动

点击 MT7601U USB 会进入 http://www.mtk.com.tw/en/downloads/mt7601u-usb/ 页面，在页面下方填写上你的用户名和邮箱，以及验证码，提交。



fyyy4030

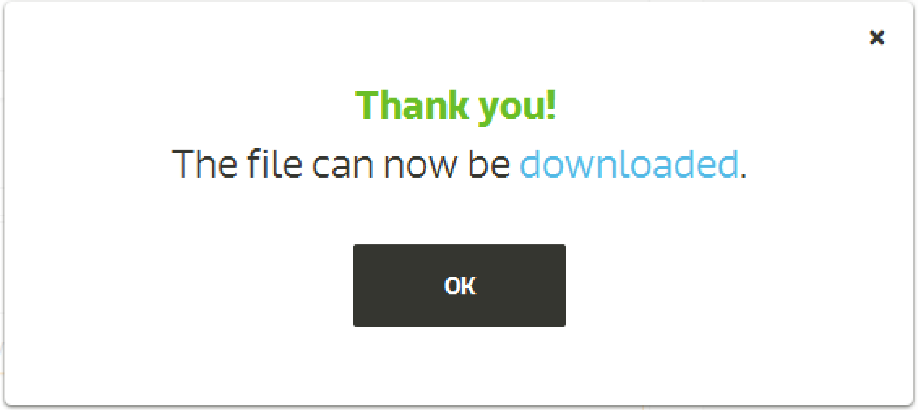
fyyy4030@qq.com

ngpsenc sur

Enter the words above:

ngpsenc

输入完成后，点击提交，会自动下载文件，同时出现以下窗口，如果没有下载文件 则可以手动点上面的“downloaded ”



Thank you!

The file can now be downloaded.

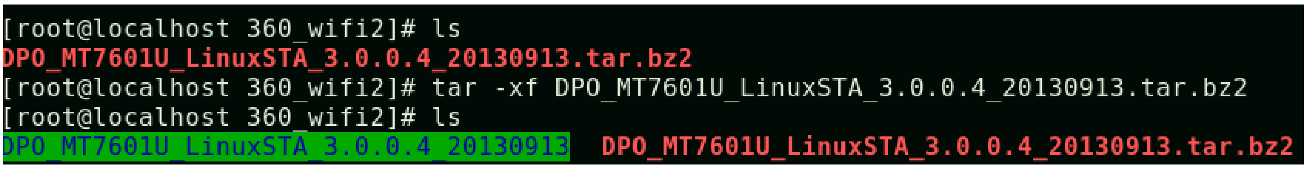
打开邮箱就可以下载到了

DPO_MT7601U_LinuxSTA_3.0.0.4_20130913.tar.bz2

http://s3.amazonaws.com/mtk.cfs/Downloads/linux/DPO_MT7601U_LinuxSTA_3.0.0.4_20130913.tar.bz2

2. 把驱动复制到 Linux 系统中解压

```
[root@localhost 360_wifi2]# ls
DPO_MT7601U_LinuxSTA_3.0.0.4_20130913.tar.bz2
[root@localhost 360_wifi2]# tar -xf DPO_MT7601U_LinuxSTA_3.0.0.4_20130913.tar.bz2
[root@localhost 360_wifi2]# ls
DPO_MT7601U_LinuxSTA_3.0.0.4_20130913    DPO_MT7601U_LinuxSTA_3.0.0.4_20130913.tar.bz2
[root@localhost 360_wifi2]# cd DPO_MT7601U_LinuxSTA_3.0.0.4_20130913
[root@localhost DPO_MT7601U_LinuxSTA_3.0.0.4_20130913]#
```



```
[root@localhost 360_wifi2]# ls
DPO_MT7601U_LinuxSTA_3.0.0.4_20130913.tar.bz2
[root@localhost 360_wifi2]# tar -xf DPO_MT7601U_LinuxSTA_3.0.0.4_20130913.tar.bz2
[root@localhost 360_wifi2]# ls
DPO_MT7601U_LinuxSTA_3.0.0.4_20130913  DPO_MT7601U_LinuxSTA_3.0.0.4_20130913.tar.bz2
```

3. 添加 ID 支持：

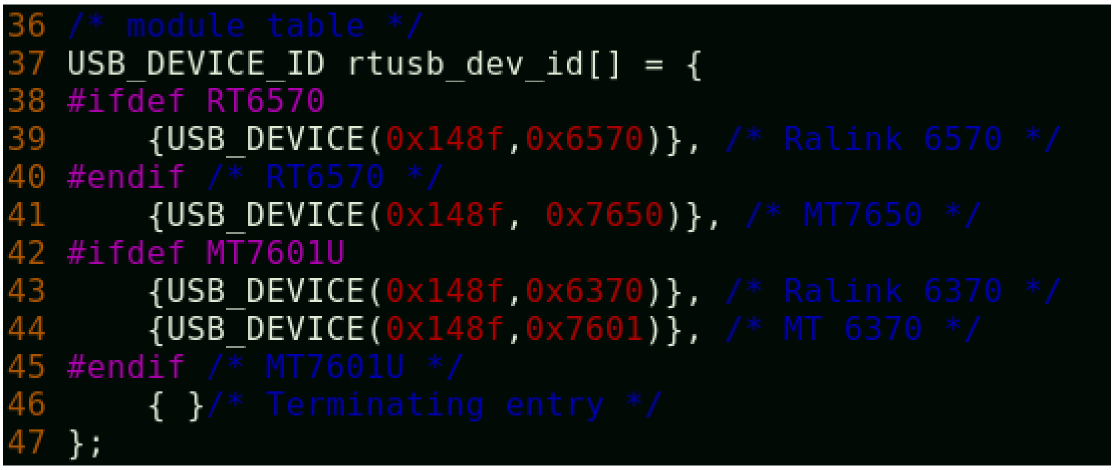
修改 common/rtusb_dev_id.c 文件，

```
[root@localhost DPO_MT7601U_LinuxSTA_3.0.0.4_20130913]#vim common/rtusb_dev_id.c +/7601
```

```
[root@localhost DPO_MT7601U_LinuxSTA_3.0.0.4_20130913]# vim common/rtusb_dev_id.c +/7601
```

找到以下数组定义

```
/* module table */
USB_DEVICE_ID rtusb_dev_id[] = {
#ifdef RT6570
        {USB_DEVICE(0x148f,0x6570)}, /* Ralink 6570 */
#endif /* RT6570 */
        {USB_DEVICE(0x148f, 0x7650)}, /* MT7650 */
#ifdef MT7601U
        {USB_DEVICE(0x148f,0x6370)}, /* Ralink 6370 */
        {USB_DEVICE(0x148f,0x7601)}, /* MT 6370 */
#endif /* MT7601U */
        { }/* Terminating entry */
};
```



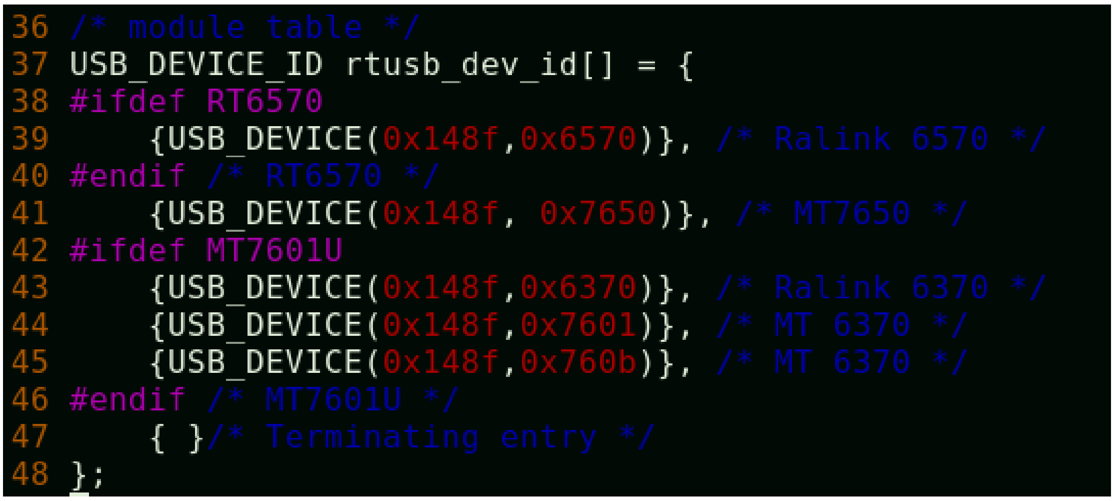
```
/* module table */
USB_DEVICE_ID rtusb_dev_id[] = {
#ifdef RT6570
    {USB_DEVICE(0x148f,0x6570)}, /* Ralink 6570 */
#endif /* RT6570 */
    {USB_DEVICE(0x148f, 0x7650)}, /* MT7650 */
#ifdef MT7601U
    {USB_DEVICE(0x148f,0x6370)}, /* Ralink 6370 */
    {USB_DEVICE(0x148f,0x7601)}, /* MT 6370 */
#endif /* MT7601U */
    { }/* Terminating entry */
};
```

在上面的数据中添加一项:

{USB_DEVICE(0x148f,0x760b)}, /* 360 Wifi */

添加后如下：

```
/* module table */
USB_DEVICE_ID rtusb_dev_id[] = {
#ifdef RT6570
        {USB_DEVICE(0x148f,0x6570)}, /* Ralink 6570 */
#endif /* RT6570 */
        {USB_DEVICE(0x148f, 0x7650)}, /* MT7650 */
#ifdef MT7601U
        {USB_DEVICE(0x148f,0x6370)}, /* Ralink 6370 */
        {USB_DEVICE(0x148f,0x7601)}, /* MT 6370 */
        {USB_DEVICE(0x148f,0x760b)}, /* MT 6370 */
#endif /* MT7601U */
        { }/* Terminating entry */
};
```



```
/* module table */
USB_DEVICE_ID rtusb_dev_id[] = {
#ifdef RT6570
    {USB_DEVICE(0x148f,0x6570)}, /* Ralink 6570 */
#endif /* RT6570 */
    {USB_DEVICE(0x148f, 0x7650)}, /* MT7650 */
#ifdef MT7601U
    {USB_DEVICE(0x148f,0x6370)}, /* Ralink 6370 */
    {USB_DEVICE(0x148f,0x7601)}, /* MT 6370 */
    {USB_DEVICE(0x148f,0x760b)}, /* MT 6370 */
#endif /* MT7601U */
    { }/* Terminating entry */
};
```

（参考的 http://www.freemindworld.com/blog/2013/131010_360_wifi_in_linux.shtml）

4. 修改头文件，去除调试信息　　include/os/rt_linux.h

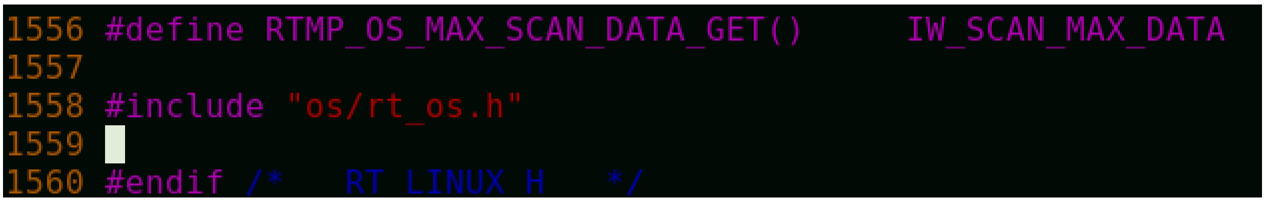
这一步其实可以不做，不过经笔者移植发现，不做这一步，WIFI 驱动安装上后，终端会不断输出调度信息，让我们无法使用终端输入命令。

跳转到这个文件的 1559 行，如下：

1558 #include "os/rt_os.h"

1559

1560 #endif /* __RT_LINUX_H__ */



```
1556 #define RTMP_OS_MAX_SCAN_DATA_GET()      IW_SCAN_MAX_DATA
1557
1558 #include "os/rt_os.h"
1559
1560 #endif /*  __RT_LINUX_H__  */
```

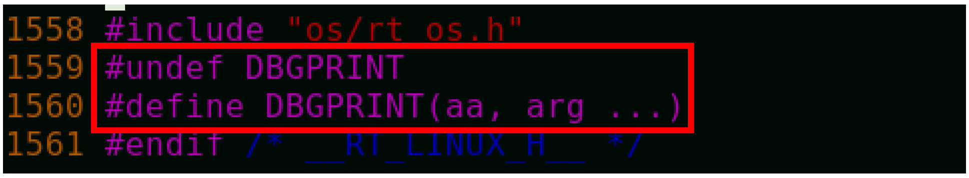
添加两行代码：

1558 #include "os/rt_os.h"

1559 #undef DBGPRINT

1560 #define DBGPRINT(aa, arg ...)

1561 #endif /* __RT_LINUX_H__ */

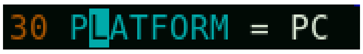


```
1558 #include "os/rt_os.h"
1559 #undef DBGPRINT
1560 #define DBGPRINT(aa, arg ...)
1561 #endif /*  __RT_LINUX_H__  */
```

5. 修改 Makefile,使它适合自己的开发板内核

1） 找到

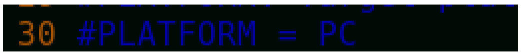
PLATFORM = PC



```
30 PLATFORM = PC
```

注释这行,即在前面添加#,如下：

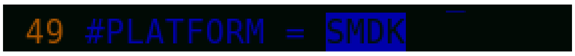
#PLATFORM = PC



```
30 #PLATFORM = PC
```

2） 找到

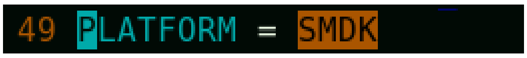
#PLATFORM = SMDK



```
49 #PLATFORM = SMDK
```

开启这行，即这行前面的号去除。如下：

PLATFORM = SMDK



```
49 PLATFORM = SMDK
```

3） 修改内核源码路径及交叉编译器。

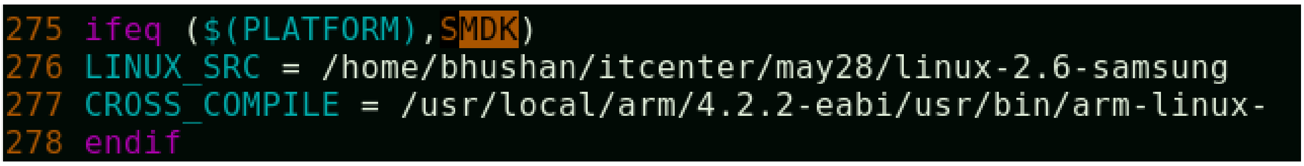
找到以下：（前面的是行号）

275 ifeq ($(PLATFORM),SMDK)

276 LINUX_SRC = /home/bhushan/itcenter/may28/linux-2.6-samsung

277 CROSS_COMPILE = /usr/local/arm/4.2.2-eabi/usr/bin/arm-linux-

278 endif



```
275 ifeq ($(PLATFORM),SMDK)
276 LINUX_SRC = /home/bhushan/itcenter/may28/linux-2.6-samsung
277 CROSS_COMPILE = /usr/local/arm/4.2.2-eabi/usr/bin/arm-linux-
278 endif
```

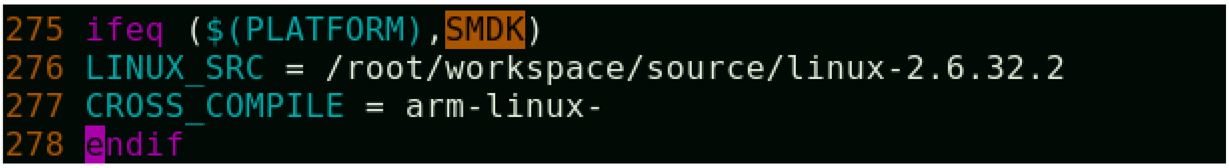
修改为自己的内核源码路径以及编译器路径，我的源码路径是：root/workspace/source/linux-2.6.32.2 ，编译器是 arm-linux-gcc，所以修改成以下：

275 ifeq ($(PLATFORM),SMDK)

276 LINUX_SRC = /root/workspace/source/linux-2.6.32.2

277 CROSS_COMPILE = arm-linux-

278 endif



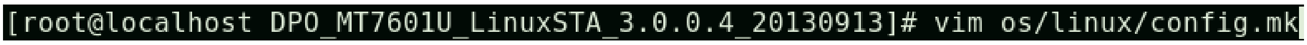
```
275 ifeq ($(PLATFORM),SMDK)
276 LINUX_SRC = /root/workspace/source/linux-2.6.32.2
277 CROSS_COMPILE = arm-linux-
278 endif
```

6. 按照 README_STA_usb 中的说明 make 和 make install。把 os/linux/mt7601Usta.ko 复制到开发板根文件系统中

arm-linux-strip -S os/linux/mt7601Usta.ko

1） 修改 WIFI 配置，打开 os/linux/config.mk，

[root@localhost DPO_MT7601U_LinuxSTA_3.0.0.4_20130913]# vim os/linux/config.mk

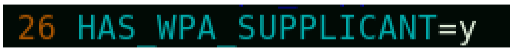


```
[root@localhost DPO_MT7601U_LinuxSTA_3.0.0.4_20130913]# vim os/linux/config.mk
```

找到以下选项并且进行配置。

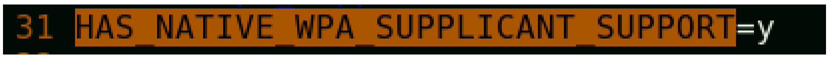
源码默认地址/etc/Wireless/RT2870STA
include/os/rt_linux.h

26 HAS_WPA_SUPPLICANT=y



```
26 HAS_WPA_SUPPLICANT=y
```

31 HAS_NATIVE_WPA_SUPPLICANT_SUPPORT=y

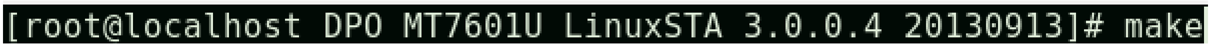


```
31 HAS_NATIVE_WPA_SUPPLICANT_SUPPORT=y
```

2） 编译 ：

直接输入 make 命令即可。

[root@localhost DPO_MT7601U_LinuxSTA_3.0.0.4_20130913]# make

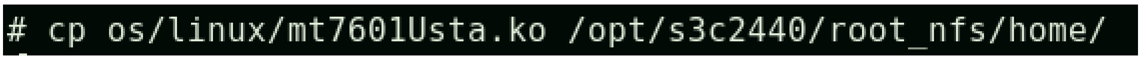


```
[root@localhost DPO_MT7601U_LinuxSTA_3.0.0.4_20130913]# make
```

3） 复制得到的驱动文件到开发板根文件系统中

本人使用 NFS 文件系统，路径是/opt/s3c2440/root_nfs/home/

cp os/linux/mt7601Usta.ko /opt/s3c2440/root_nfs/home/

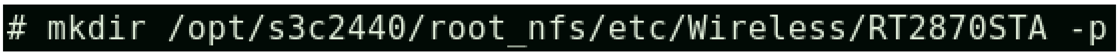


```
# cp os/linux/mt7601Usta.ko /opt/s3c2440/root_nfs/home/
```

7. 复制配置文件到开发板根文件系统

1）先创建目录

mkdir /opt/s3c2440/root_nfs/etc/Wireless/RT2870STA -p



```
# mkdir /opt/s3c2440/root_nfs/etc/Wireless/RT2870STA -p
```

2）复制 RT2870STA 到刚刚创建的目录中

#cp RT2870STA.dat /opt/s3c2440/root_nfs/etc/Wireless/RT2870STA/

```
# cp RT2870STA.dat /opt/s3c2440/root nfs/etc/Wireless/RT2870STA/
```

8. 安装驱动模块

insmod /home/mt7601Usta.ko

成功后有以下信息提示：

rtusb init rt2870 --->

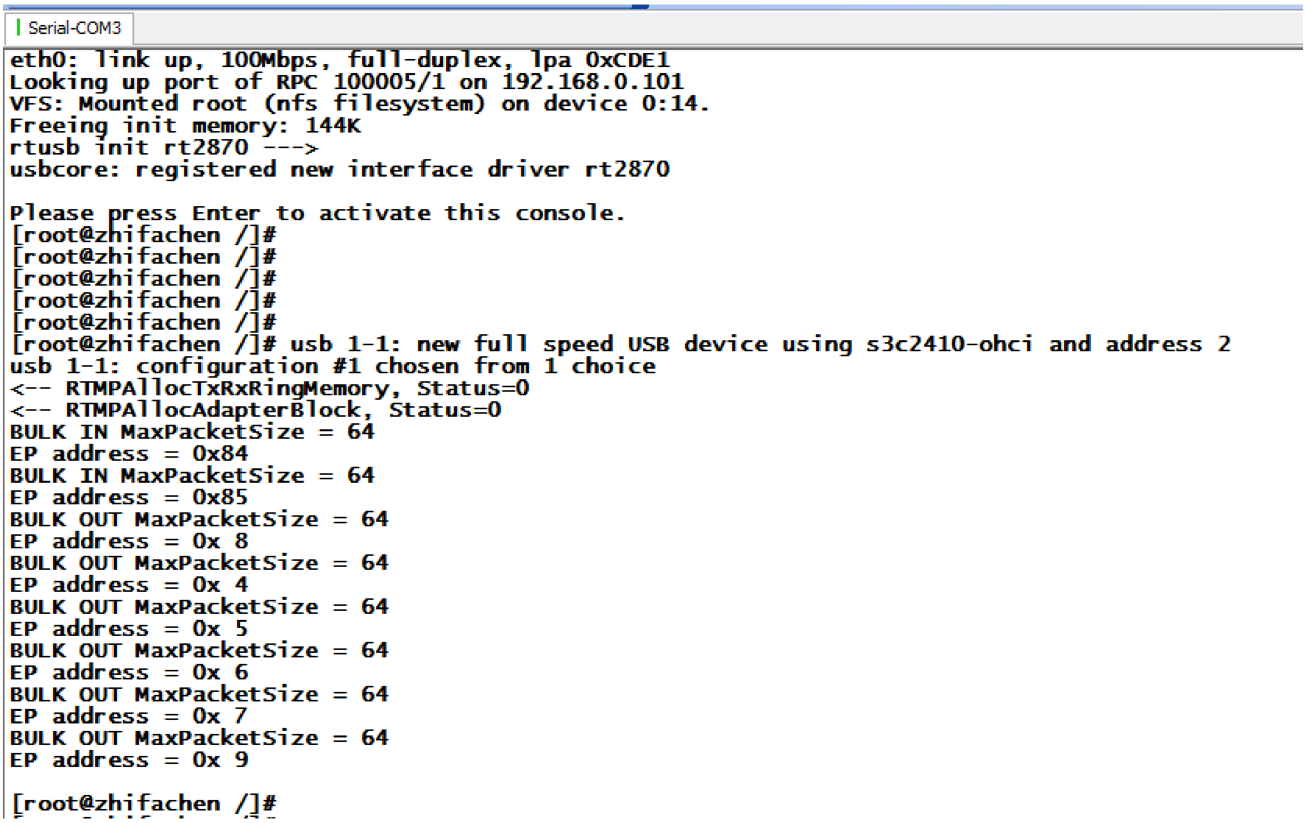
usbcore: registered new interface driver rt2870

9. 插上 USB WIFI,如果成功，会出现以下提示：

```
usb 1-1: new full speed USB device using s3c2410-ohci and address 2
usb 1-1: configuration #1 chosen from 1 choice
<-- RTMPAllocTxRxRingMemory, Status=0
<-- RTMPAllocAdapterBlock, Status=0
BULK IN MaxPacketSize = 64
EP address = 0x84
BULK IN MaxPacketSize = 64
EP address = 0x85
BULK OUT MaxPacketSize = 64
EP address = 0x 8
BULK OUT MaxPacketSize = 64
EP address = 0x 4
BULK OUT MaxPacketSize = 64
EP address = 0x 5
BULK OUT MaxPacketSize = 64
EP address = 0x 6
BULK OUT MaxPacketSize = 64
EP address = 0x 7
BULK OUT MaxPacketSize = 64
EP address = 0x 9
```

提示错误：

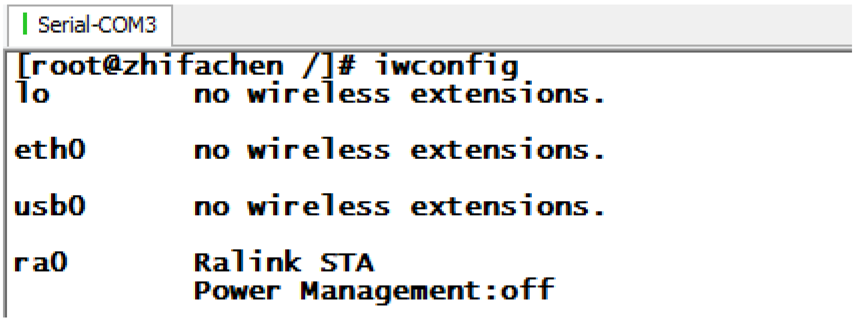
```
[   53.975000] ERROR!!! Failed to allocate memory - TxRxRing
[   53.975000] <-- RTMPAllocAdapterBlock, Status=3
[   53.975000] rt2870: probe of 1-2.3.3:1.0 failed with error -1
coherent_pool=2M
```



```
eth0: link up, 100Mbps, full-duplex, lpa 0xCDE1
Looking up port of RPC 100005/1 on 192.168.0.101
VFS: Mounted root (nfs filesystem) on device 0:14.
Freeing init memory: 144K
rtusb init rt2870 --->
usbcore: registered new interface driver rt2870

Please press Enter to activate this console.
[root@zhifachen /]#
[root@zhifachen /]#
[root@zhifachen /]#
[root@zhifachen /]#
[root@zhifachen /]#
[root@zhifachen /]# usb 1-1: new full speed USB device using s3c2410-ohci and address 2
usb 1-1: configuration #1 chosen from 1 choice
<-- RTMPAllocTxRxRingMemory, Status=0
<-- RTMPAllocAdapterBlock, Status=0
BULK IN MaxPacketSize = 64
EP address = 0x84
BULK IN MaxPacketSize = 64
EP address = 0x85
BULK OUT MaxPacketSize = 64
EP address = 0x 8
BULK OUT MaxPacketSize = 64
EP address = 0x 4
BULK OUT MaxPacketSize = 64
EP address = 0x 5
BULK OUT MaxPacketSize = 64
EP address = 0x 6
BULK OUT MaxPacketSize = 64
EP address = 0x 7
BULK OUT MaxPacketSize = 64
EP address = 0x 9

[root@zhifachen /]#
```

可以使用 iwconfig 命令查看是否安装了 wifi 设备。

```
[root@zhifachen /]# iwconfig
lo          no wireless extensions.

eth0        no wireless extensions.

usb0        no wireless extensions.

ra0         Ralink STA
            Power Management:off
```



```
[root@zhifachen /]# iwconfig
lo          no wireless extensions.

eth0        no wireless extensions.

usb0        no wireless extensions.

ra0         Ralink STA
            Power Management:off
```

10. 在开发板根文件系统/etc/目录创建 wpa_supplicant.conf 文件 ， 输入以下内容：

```
# WPA-PSK/TKIP
ctrl_interface=/var/run/wap_supplicant
```

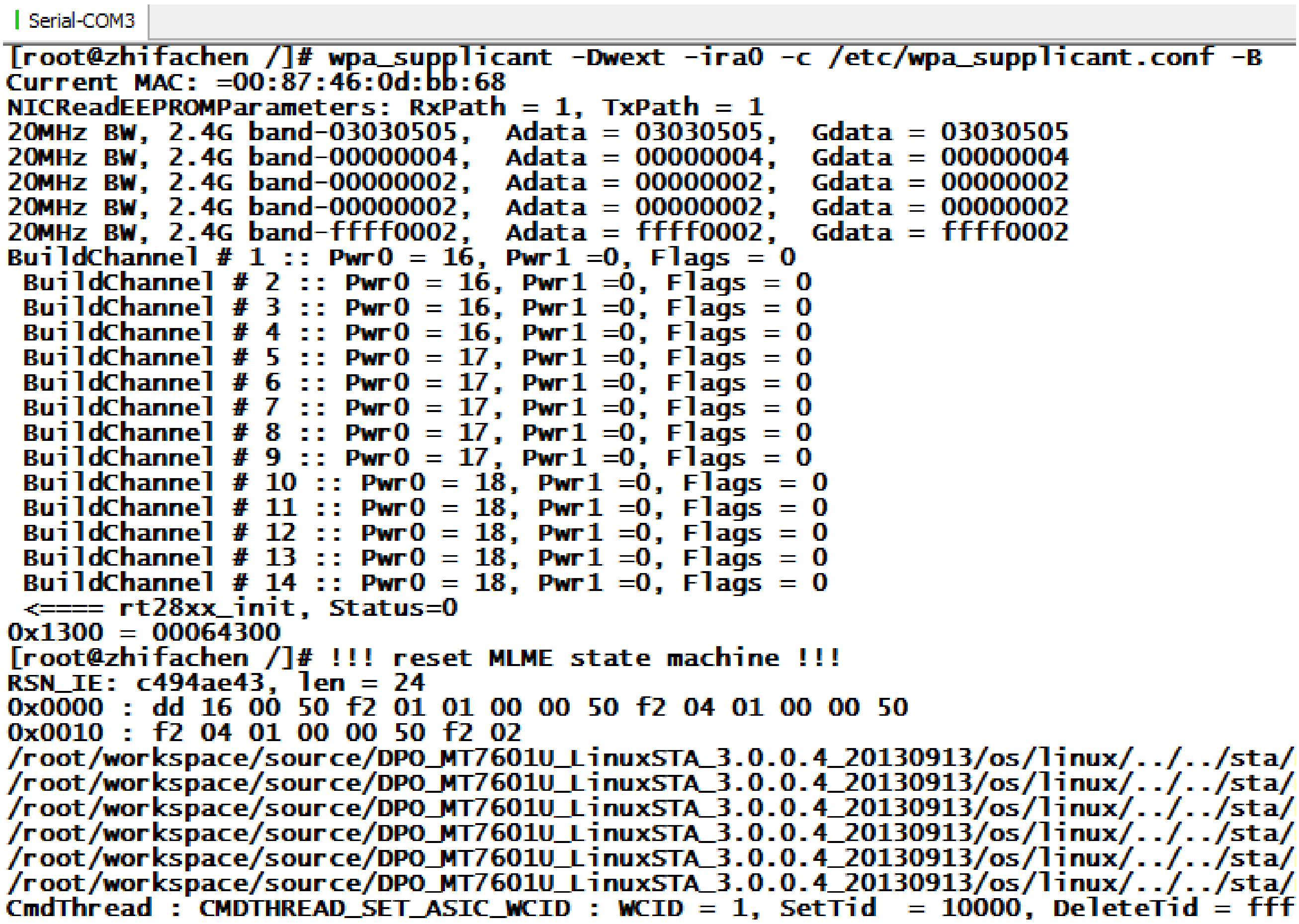
```
ap_scan=1
update_config=1
network={
        ssid="fyyy"      #WIFI 网络名，根据自己的修改
        key_mgmt=WPA-PSK
        proto=WPA
        pairwise=TKIP CCMP
        group=TKIP CCMP
#        priority=2
        psk="5C-63-BF-C7-D7-96"      #WIFI 密码，根据自己的修改

}
```

11.　创建网卡控制接口：mkdir /var/run/wap_supplicant –p

12.　启动网卡

```
[root@zhifachen /]# wpa_supplicant -Dwext -ira0 -c /etc/wpa_supplicant.conf -B
Current MAC: =00:87:46:0d:bb:68
NICReadEEPROMParameters: RxPath = 1, TxPath = 1
20MHz BW, 2.4G band-03030505,   Adata = 03030505,   Gdata = 03030505
20MHz BW, 2.4G band-00000004,   Adata = 00000004,   Gdata = 00000004
20MHz BW, 2.4G band-00000002,   Adata = 00000002,   Gdata = 00000002
20MHz BW, 2.4G band-00000002,   Adata = 00000002,   Gdata = 00000002
20MHz BW, 2.4G band-ffff0002,   Adata = ffff0002,   Gdata = ffff0002
BuildChannel # 1 :: Pwr0 = 16, Pwr1 =0, Flags = 0
 BuildChannel # 2 :: Pwr0 = 16, Pwr1 =0, Flags = 0
 BuildChannel # 3 :: Pwr0 = 16, Pwr1 =0, Flags = 0
 BuildChannel # 4 :: Pwr0 = 16, Pwr1 =0, Flags = 0
 BuildChannel # 5 :: Pwr0 = 17, Pwr1 =0, Flags = 0
 BuildChannel # 6 :: Pwr0 = 17, Pwr1 =0, Flags = 0
 BuildChannel # 7 :: Pwr0 = 17, Pwr1 =0, Flags = 0
 BuildChannel # 8 :: Pwr0 = 17, Pwr1 =0, Flags = 0
 BuildChannel # 9 :: Pwr0 = 17, Pwr1 =0, Flags = 0
 BuildChannel # 10 :: Pwr0 = 18, Pwr1 =0, Flags = 0
 BuildChannel # 11 :: Pwr0 = 18, Pwr1 =0, Flags = 0
 BuildChannel # 12 :: Pwr0 = 18, Pwr1 =0, Flags = 0
 BuildChannel # 13 :: Pwr0 = 18, Pwr1 =0, Flags = 0
 BuildChannel # 14 :: Pwr0 = 18, Pwr1 =0, Flags = 0
 <==== rt28xx_init, Status=0
0x1300 = 00064300
[root@zhifachen /]# !!! reset MLME state machine !!!
RSN_IE: c494ae43, len = 24
0x0000 : dd 16 00 50 f2 01 01 00 00 50 f2 04 01 00 00 50
```

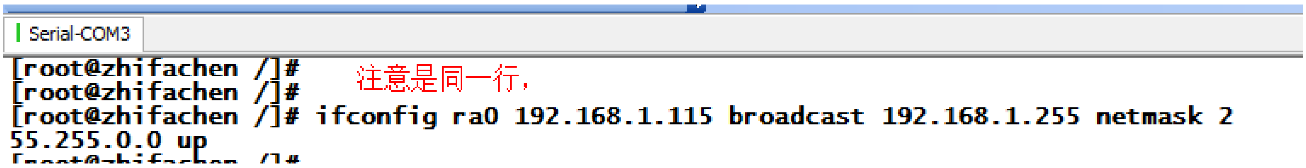
```
0x0010 : f2 04 01 00 00 50 f2 02
/root/workspace/source/DPO_MT7601U_LinuxSTA_3.0.0.4_20130913/os/linux/../../sta/rtmp_data.c:540 assert pRxWI->RxWIWirelessCliID == BSSID_WCIDfailed
/root/workspace/source/DPO_MT7601U_LinuxSTA_3.0.0.4_20130913/os/linux/../../sta/rtmp_data.c:540 assert pRxWI->RxWIWirelessCliID == BSSID_WCIDfailed
/root/workspace/source/DPO_MT7601U_LinuxSTA_3.0.0.4_20130913/os/linux/../../sta/rtmp_data.c:540 assert pRxWI->RxWIWirelessCliID == BSSID_WCIDfailed
/root/workspace/source/DPO_MT7601U_LinuxSTA_3.0.0.4_20130913/os/linux/../../sta/rtmp_data.c:540 assert pRxWI->RxWIWirelessCliID == BSSID_WCIDfailed
/root/workspace/source/DPO_MT7601U_LinuxSTA_3.0.0.4_20130913/os/linux/../../sta/rtmp_data.c:540 assert pRxWI->RxWIWirelessCliID == BSSID_WCIDfailed
/root/workspace/source/DPO_MT7601U_LinuxSTA_3.0.0.4_20130913/os/linux/../../sta/rtmp_data.c:540 assert pRxWI->RxWIWirelessCliID == BSSID_WCIDfailed
CmdThread : CMDTHREAD_SET_ASIC_WCID : WCID = 1, SetTid  = 10000, DeleteTid = ffffffff.
1-MACValue= 7a75f04,
2-MACValue= 118d5,
        Key = bf:d8:b7:00:27:f4:de:a7:34:6e:0a:28:e7:e2:6f:b5
        Rx MIC Key = 00:00:00:00:00:00:00:00
        Tx MIC Key = 00:00:00:00:00:00:00:00

[root@zhifachen /]#
```



```
[root@zhifachen /]# wpa_supplicant -Dwext -ira0 -c /etc/wpa_supplicant.conf -B
Current MAC: =00:87:46:0d:bb:68
NICReadEEPROMParameters: RxPath = 1, TxPath = 1
20MHz BW, 2.4G band-03030505,  Adata = 03030505,  Gdata = 03030505
20MHz BW, 2.4G band-00000004,  Adata = 00000004,  Gdata = 00000004
20MHz BW, 2.4G band-00000002,  Adata = 00000002,  Gdata = 00000002
20MHz BW, 2.4G band-00000002,  Adata = 00000002,  Gdata = 00000002
20MHz BW, 2.4G band-ffff0002,  Adata = ffff0002,  Gdata = ffff0002
BuildChannel # 1 :: Pwr0 = 16, Pwr1 =0, Flags = 0
 BuildChannel # 2 :: Pwr0 = 16, Pwr1 =0, Flags = 0
 BuildChannel # 3 :: Pwr0 = 16, Pwr1 =0, Flags = 0
 BuildChannel # 4 :: Pwr0 = 16, Pwr1 =0, Flags = 0
 BuildChannel # 5 :: Pwr0 = 17, Pwr1 =0, Flags = 0
 BuildChannel # 6 :: Pwr0 = 17, Pwr1 =0, Flags = 0
 BuildChannel # 7 :: Pwr0 = 17, Pwr1 =0, Flags = 0
 BuildChannel # 8 :: Pwr0 = 17, Pwr1 =0, Flags = 0
 BuildChannel # 9 :: Pwr0 = 17, Pwr1 =0, Flags = 0
 BuildChannel # 10 :: Pwr0 = 18, Pwr1 =0, Flags = 0
 BuildChannel # 11 :: Pwr0 = 18, Pwr1 =0, Flags = 0
 BuildChannel # 12 :: Pwr0 = 18, Pwr1 =0, Flags = 0
 BuildChannel # 13 :: Pwr0 = 18, Pwr1 =0, Flags = 0
 BuildChannel # 14 :: Pwr0 = 18, Pwr1 =0, Flags = 0
 <==== rt28xx_init, Status=0
0x1300 = 00064300
[root@zhifachen /]# !!! reset MLME state machine !!!
RSN_IE: c494ae43, len = 24
0x0000 : dd 16 00 50 f2 01 01 00 00 50 f2 04 01 00 00 50
0x0010 : f2 04 01 00 00 50 f2 02
/root/workspace/source/DPO_MT7601U_LinuxSTA_3.0.0.4_20130913/os/linux/../../sta/
/root/workspace/source/DPO_MT7601U_LinuxSTA_3.0.0.4_20130913/os/linux/../../sta/
/root/workspace/source/DPO_MT7601U_LinuxSTA_3.0.0.4_20130913/os/linux/../../sta/
/root/workspace/source/DPO_MT7601U_LinuxSTA_3.0.0.4_20130913/os/linux/../../sta/
/root/workspace/source/DPO_MT7601U_LinuxSTA_3.0.0.4_20130913/os/linux/../../sta/
/root/workspace/source/DPO_MT7601U_LinuxSTA_3.0.0.4_20130913/os/linux/../../sta/
CmdThread : CMDTHREAD_SET_ASIC_WCID : WCID = 1, SetTid  = 10000, DeleteTid = fff
```

13. 配置网卡 IP:

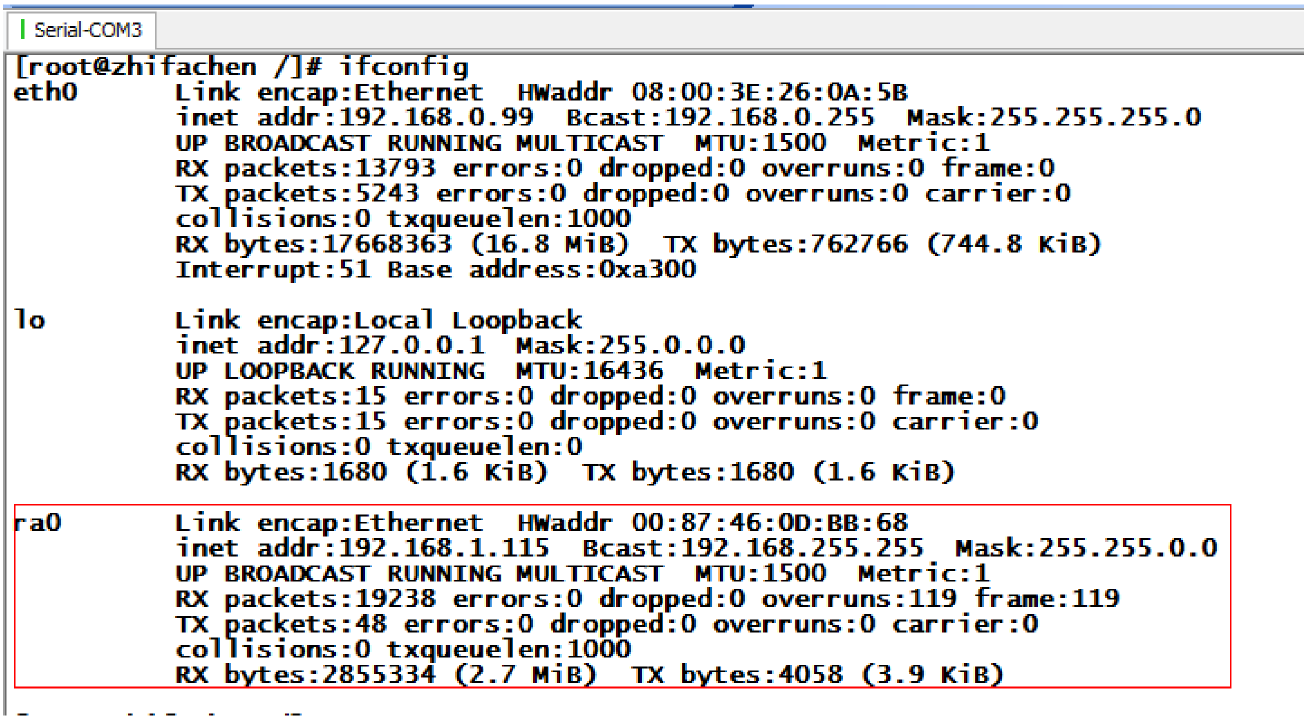
ifconfig ra0 192.168.1.115 broadcast 192.168.1.255 netmask 255.255.0.0 up



```
[root@zhifachen /]#
[root@zhifachen /]#
[root@zhifachen /]# ifconfig ra0 192.168.1.115 broadcast 192.168.1.255 netmask 2
55.255.0.0 up
[root@zhifachen /]#
```

注意是同一行，

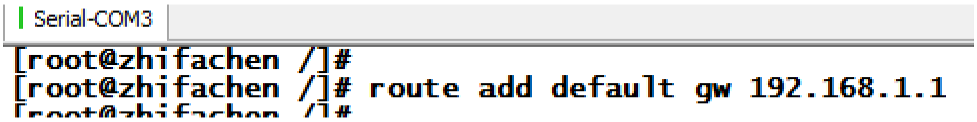
查看是否配置成功



```
[root@zhifachen /]# ifconfig
eth0      Link encap:Ethernet  HWaddr 08:00:3E:26:0A:5B
          inet addr:192.168.0.99  Bcast:192.168.0.255  Mask:255.255.255.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:13793 errors:0 dropped:0 overruns:0 frame:0
          TX packets:5243 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:17668363 (16.8 MiB)  TX bytes:762766 (744.8 KiB)
          Interrupt:51 Base address:0xa300

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:15 errors:0 dropped:0 overruns:0 frame:0
          TX packets:15 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:1680 (1.6 KiB)  TX bytes:1680 (1.6 KiB)

ra0       Link encap:Ethernet  HWaddr 00:87:46:0D:BB:68
          inet addr:192.168.1.115  Bcast:192.168.255.255  Mask:255.255.0.0
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:19238 errors:0 dropped:0 overruns:119 frame:119
          TX packets:48 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:2855334 (2.7 MiB)  TX bytes:4058 (3.9 KiB)
```

14. 配置默网关

route add default gw 192.168.1.1



```
[root@zhifachen /]#
[root@zhifachen /]# route add default gw 192.168.1.1
[root@zhifachen /]#
```

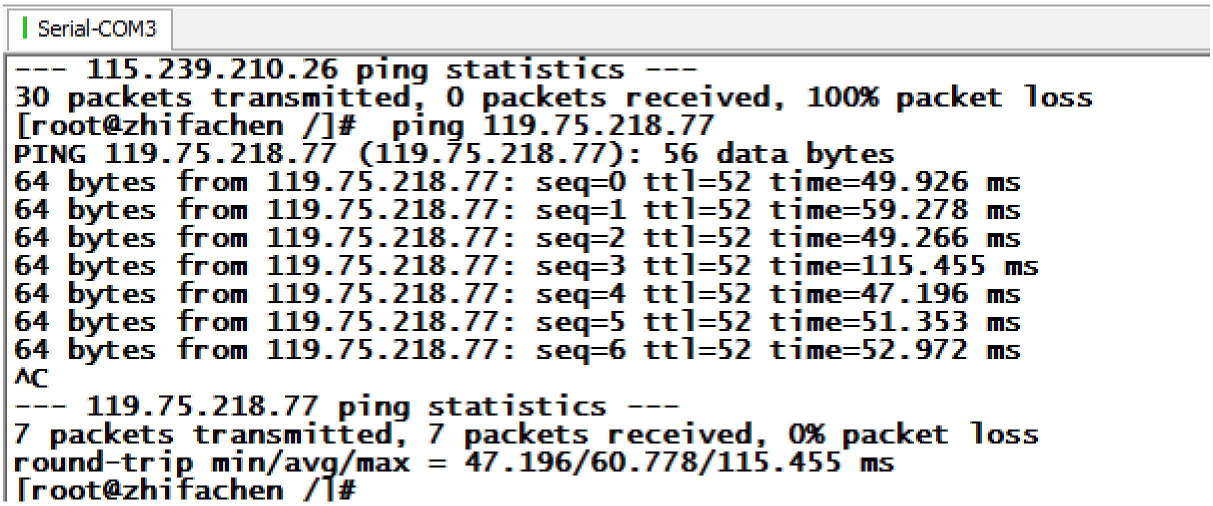
15. ping 外网测试，

www.baidu.con 的 IP 是 119.75.218.77，使用 ping 命令测试是否联通: ping 119.75.218.77

```
[root@zhifachen /]#   ping 119.75.218.77
PING 119.75.218.77 (119.75.218.77): 56 data bytes
64 bytes from 119.75.218.77: seq=0 ttl=52 time=49.926 ms
64 bytes from 119.75.218.77: seq=1 ttl=52 time=59.278 ms
64 bytes from 119.75.218.77: seq=2 ttl=52 time=49.266 ms
64 bytes from 119.75.218.77: seq=3 ttl=52 time=115.455 ms
64 bytes from 119.75.218.77: seq=4 ttl=52 time=47.196 ms
```

64 bytes from 119.75.218.77: seq=5 ttl=52 time=51.353 ms
64 bytes from 119.75.218.77: seq=6 ttl=52 time=52.972 ms
^C
--- 119.75.218.77 ping statistics ---
7 packets transmitted, 7 packets received, 0% packet loss
round-trip min/avg/max = 47.196/60.778/115.455 ms
[root@zhifachen /]#

这样说明连通了。



```
--- 115.239.210.26 ping statistics ---
30 packets transmitted, 0 packets received, 100% packet loss
[root@zhifachen /]#  ping 119.75.218.77
PING 119.75.218.77 (119.75.218.77): 56 data bytes
64 bytes from 119.75.218.77: seq=0 ttl=52 time=49.926 ms
64 bytes from 119.75.218.77: seq=1 ttl=52 time=59.278 ms
64 bytes from 119.75.218.77: seq=2 ttl=52 time=49.266 ms
64 bytes from 119.75.218.77: seq=3 ttl=52 time=115.455 ms
64 bytes from 119.75.218.77: seq=4 ttl=52 time=47.196 ms
64 bytes from 119.75.218.77: seq=5 ttl=52 time=51.353 ms
64 bytes from 119.75.218.77: seq=6 ttl=52 time=52.972 ms
^C
--- 119.75.218.77 ping statistics ---
7 packets transmitted, 7 packets received, 0% packet loss
round-trip min/avg/max = 47.196/60.778/115.455 ms
[root@zhifachen /]#
```

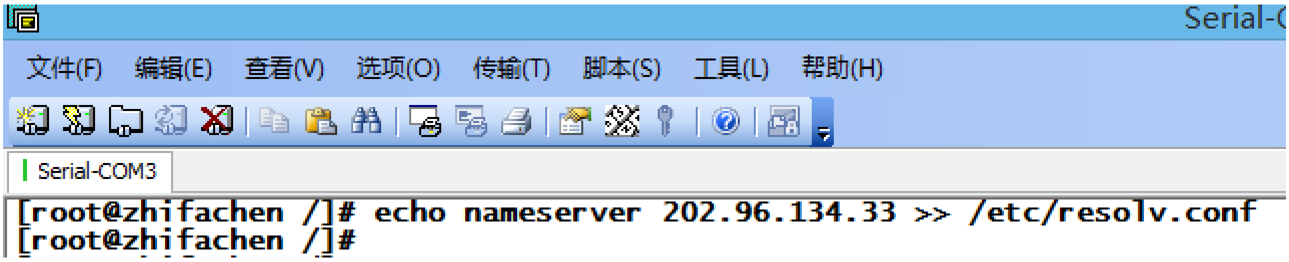
16. 实现 ping 域名功能。

在/etc/目录下创建一个 resolv.conf 文件，输入 DNS 地址，如下：

nameserver 202.96.134.33

也可以直接在串口终端使用命令创建并且写入内容。如下：

[root@zhifachen /]# echo nameserver 202.96.134.33 >> /etc/resolv.conf

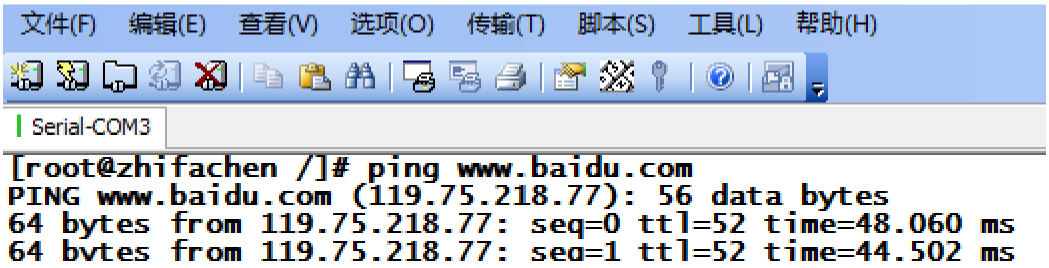


```
[root@zhifachen /]# echo nameserver 202.96.134.33 >> /etc/resolv.conf
[root@zhifachen /]#
```

说明：

深圳：DNS

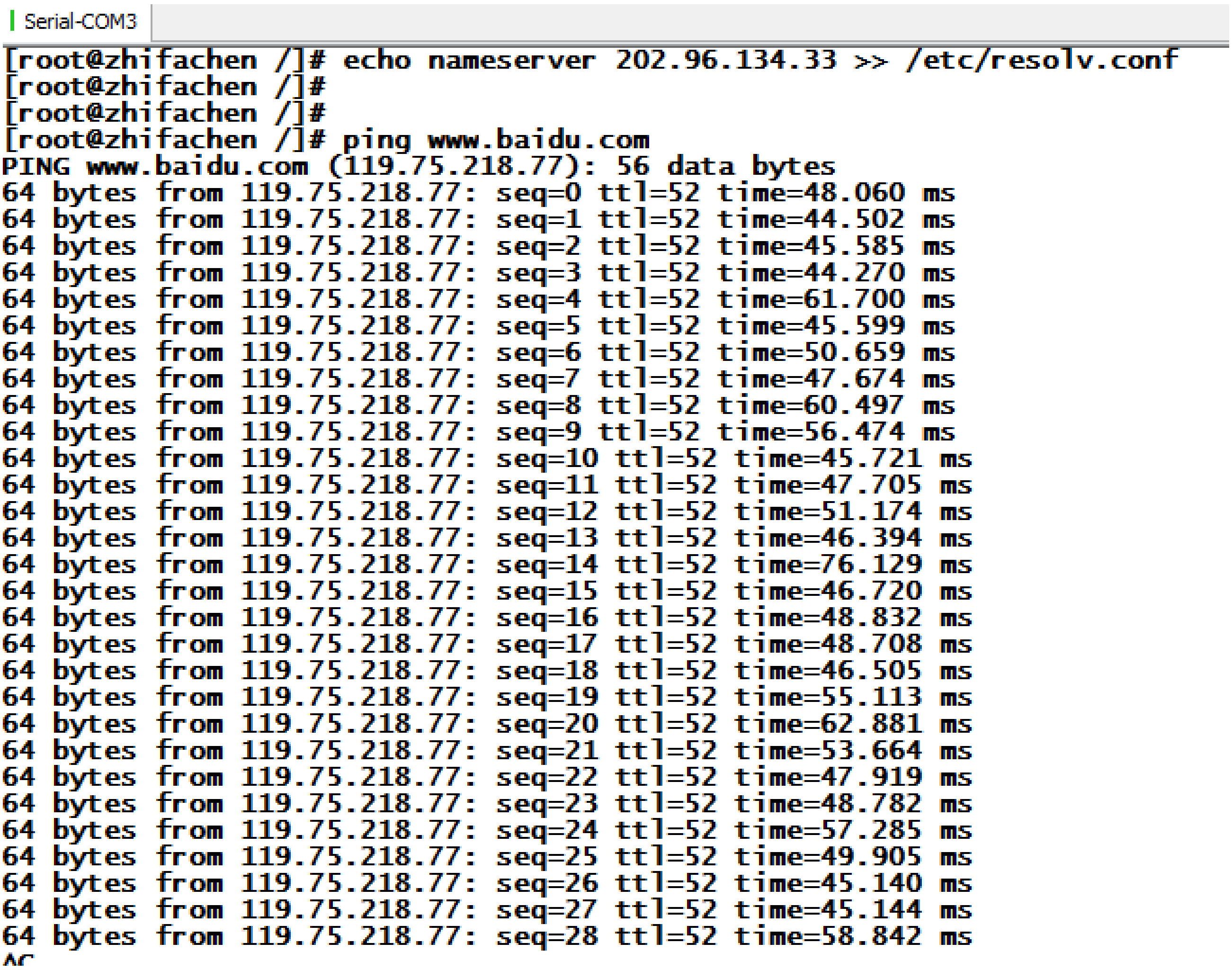
202.96.134.33



```
[root@zhifachen /]# ping www.baidu.com
PING www.baidu.com (119.75.218.77): 56 data bytes
64 bytes from 119.75.218.77: seq=0 ttl=52 time=48.060 ms
64 bytes from 119.75.218.77: seq=1 ttl=52 time=44.502 ms
```

测试结果：

```
[root@zhifachen /]# ping www.baidu.com
PING www.baidu.com (119.75.218.77): 56 data bytes
64 bytes from 119.75.218.77: seq=0 ttl=52 time=48.060 ms
64 bytes from 119.75.218.77: seq=1 ttl=52 time=44.502 ms
64 bytes from 119.75.218.77: seq=2 ttl=52 time=45.585 ms
64 bytes from 119.75.218.77: seq=3 ttl=52 time=44.270 ms
64 bytes from 119.75.218.77: seq=4 ttl=52 time=61.700 ms
64 bytes from 119.75.218.77: seq=5 ttl=52 time=45.599 ms
64 bytes from 119.75.218.77: seq=6 ttl=52 time=50.659 ms
64 bytes from 119.75.218.77: seq=7 ttl=52 time=47.674 ms
64 bytes from 119.75.218.77: seq=8 ttl=52 time=60.497 ms
64 bytes from 119.75.218.77: seq=9 ttl=52 time=56.474 ms
64 bytes from 119.75.218.77: seq=10 ttl=52 time=45.721 ms
64 bytes from 119.75.218.77: seq=11 ttl=52 time=47.705 ms
64 bytes from 119.75.218.77: seq=12 ttl=52 time=51.174 ms
64 bytes from 119.75.218.77: seq=13 ttl=52 time=46.394 ms
64 bytes from 119.75.218.77: seq=14 ttl=52 time=76.129 ms
64 bytes from 119.75.218.77: seq=15 ttl=52 time=46.720 ms
64 bytes from 119.75.218.77: seq=16 ttl=52 time=48.832 ms
64 bytes from 119.75.218.77: seq=17 ttl=52 time=48.708 ms
64 bytes from 119.75.218.77: seq=18 ttl=52 time=46.505 ms
64 bytes from 119.75.218.77: seq=19 ttl=52 time=55.113 ms
64 bytes from 119.75.218.77: seq=20 ttl=52 time=62.881 ms
64 bytes from 119.75.218.77: seq=21 ttl=52 time=53.664 ms
64 bytes from 119.75.218.77: seq=22 ttl=52 time=47.919 ms
64 bytes from 119.75.218.77: seq=23 ttl=52 time=48.782 ms
64 bytes from 119.75.218.77: seq=24 ttl=52 time=57.285 ms
64 bytes from 119.75.218.77: seq=25 ttl=52 time=49.905 ms
64 bytes from 119.75.218.77: seq=26 ttl=52 time=45.140 ms
64 bytes from 119.75.218.77: seq=27 ttl=52 time=45.144 ms
64 bytes from 119.75.218.77: seq=28 ttl=52 time=58.842 ms
^C
--- www.baidu.com ping statistics ---
29 packets transmitted, 29 packets received, 0% packet loss
round-trip min/avg/max = 44.270/51.295/76.129 ms
[root@zhifachen /]#
```



```
[root@zhifachen /]# echo nameserver 202.96.134.33 >> /etc/resolv.conf
[root@zhifachen /]#
[root@zhifachen /]#
[root@zhifachen /]# ping www.baidu.com
PING www.baidu.com (119.75.218.77): 56 data bytes
64 bytes from 119.75.218.77: seq=0 ttl=52 time=48.060 ms
64 bytes from 119.75.218.77: seq=1 ttl=52 time=44.502 ms
64 bytes from 119.75.218.77: seq=2 ttl=52 time=45.585 ms
64 bytes from 119.75.218.77: seq=3 ttl=52 time=44.270 ms
64 bytes from 119.75.218.77: seq=4 ttl=52 time=61.700 ms
64 bytes from 119.75.218.77: seq=5 ttl=52 time=45.599 ms
64 bytes from 119.75.218.77: seq=6 ttl=52 time=50.659 ms
64 bytes from 119.75.218.77: seq=7 ttl=52 time=47.674 ms
64 bytes from 119.75.218.77: seq=8 ttl=52 time=60.497 ms
64 bytes from 119.75.218.77: seq=9 ttl=52 time=56.474 ms
64 bytes from 119.75.218.77: seq=10 ttl=52 time=45.721 ms
64 bytes from 119.75.218.77: seq=11 ttl=52 time=47.705 ms
64 bytes from 119.75.218.77: seq=12 ttl=52 time=51.174 ms
64 bytes from 119.75.218.77: seq=13 ttl=52 time=46.394 ms
64 bytes from 119.75.218.77: seq=14 ttl=52 time=76.129 ms
64 bytes from 119.75.218.77: seq=15 ttl=52 time=46.720 ms
64 bytes from 119.75.218.77: seq=16 ttl=52 time=48.832 ms
64 bytes from 119.75.218.77: seq=17 ttl=52 time=48.708 ms
64 bytes from 119.75.218.77: seq=18 ttl=52 time=46.505 ms
64 bytes from 119.75.218.77: seq=19 ttl=52 time=55.113 ms
64 bytes from 119.75.218.77: seq=20 ttl=52 time=62.881 ms
64 bytes from 119.75.218.77: seq=21 ttl=52 time=53.664 ms
64 bytes from 119.75.218.77: seq=22 ttl=52 time=47.919 ms
64 bytes from 119.75.218.77: seq=23 ttl=52 time=48.782 ms
64 bytes from 119.75.218.77: seq=24 ttl=52 time=57.285 ms
64 bytes from 119.75.218.77: seq=25 ttl=52 time=49.905 ms
64 bytes from 119.75.218.77: seq=26 ttl=52 time=45.140 ms
64 bytes from 119.75.218.77: seq=27 ttl=52 time=45.144 ms
64 bytes from 119.75.218.77: seq=28 ttl=52 time=58.842 ms
^C
```

17. 可以把操作的步骤写在/etc/init.d/rcS 文件中。实现开机自己连接 WIFI。

内容如下：

```
#WIFI
#insmod /home/8192cu.ko
insmod /home/mt7601Usta.ko
mkdir /var/run/wap_supplicant -p
wpa_supplicant -Dwext -ira0 -c /etc/wpa_supplicant.conf -B
ifconfig ra0 192.168.1.115 broadcast 192.168.1.255 netmask 255.255.0.0 up
route add default gw 192.168.1.1
```

说明：

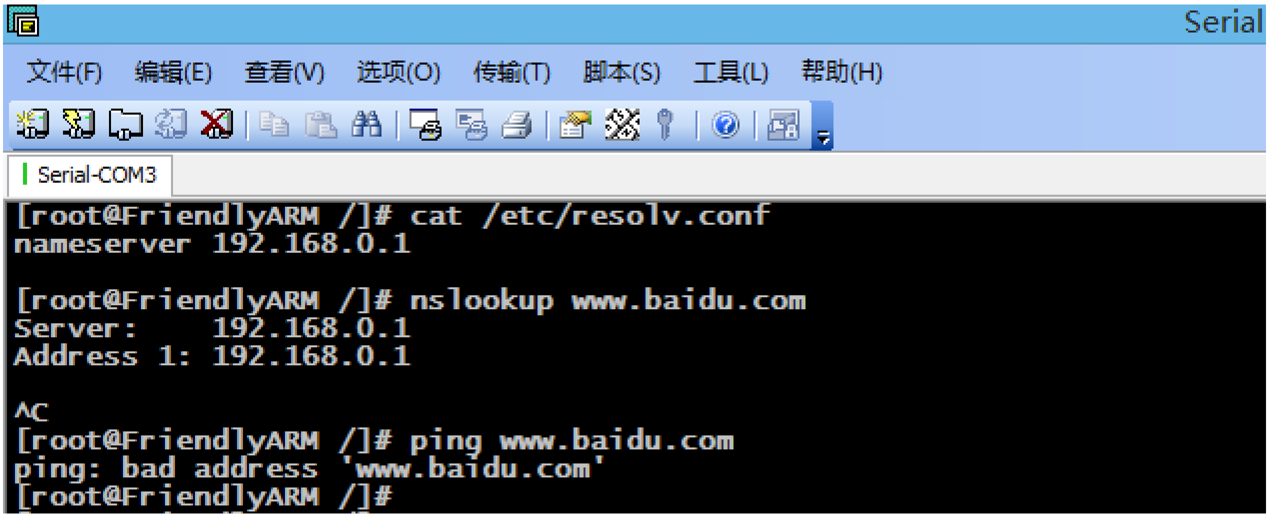
1．开发板中要想 ping 通外网 IP，需要设置默认网关，这个网关必须要和所连接的路由器设置的网关相同。

2．开发板要想 ping 通外网域名，则要设置 DNS 服务器，在 etc 目录下创建 resolv.conf 文件，其中写上 DNS 服务器 IP，一般直接写上所使用的网关即可，由路由器来解析，比如，开发板所连接的路由器设置的网关

是 192.168.1.1,则 resolv.conf 文件中写入 nameserver 192.168.1.1 即可以。也可以写公网的 DNS 服务器，比如写深圳电信 DNS,其中有一个是 202.96.134.33，则 resolv.conf 内容是 nameserver 202.96.134.33 。谷歌的 DNS 是 8.8.8.8。也可以同时写多个，则系统会逐个尝试，直到连接成功。

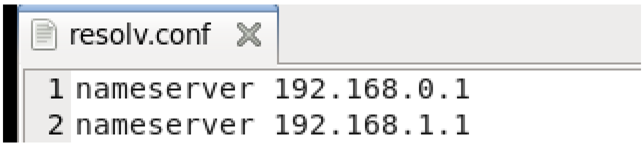
假设板子连接的是路由网关是 192.168.1.1，如果在 resolv.conf 文件中只写一个网关，并且这个网关不是 192.168.1.1 ,而是 192.168.0.1 则无法 ping 通 www.baidu.com， 如下：

```
[root@FriendlyARM /]# cat /etc/resolv.conf
nameserver 192.168.0.1

[root@FriendlyARM /]# nslookup www.baidu.com
Server:     192.168.0.1
Address 1: 192.168.0.1

^C
[root@FriendlyARM /]# ping www.baidu.com
ping: bad address 'www.baidu.com'
[root@FriendlyARM /]#
```



```
[root@FriendlyARM /]# cat /etc/resolv.conf
nameserver 192.168.0.1

[root@FriendlyARM /]# nslookup www.baidu.com
Server:    192.168.0.1
Address 1: 192.168.0.1

^C
[root@FriendlyARM /]# ping www.baidu.com
ping: bad address 'www.baidu.com'
[root@FriendlyARM /]#
```

这时候再添加上一个 DNS，如下：

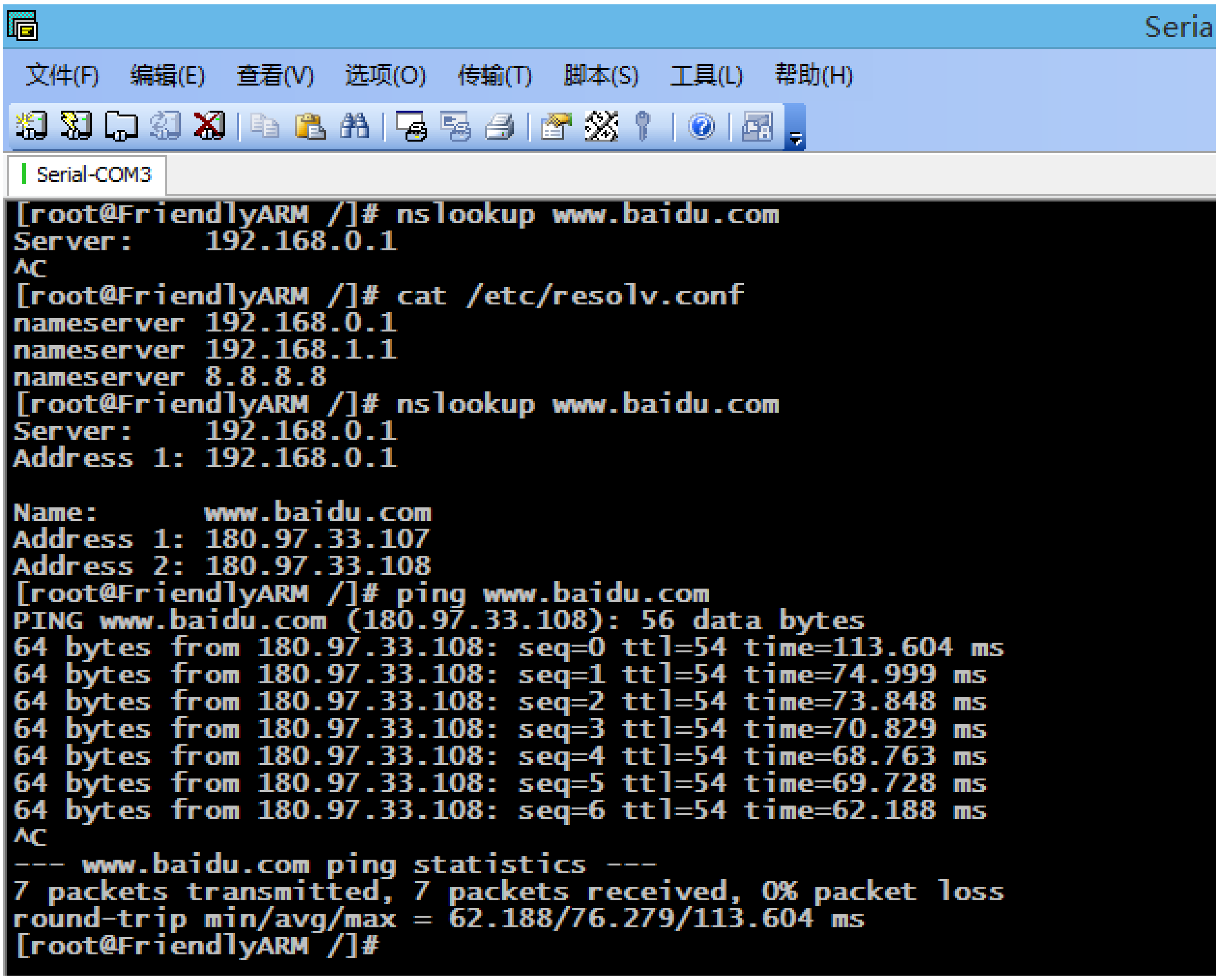


```
nameserver 192.168.0.1
nameserver 192.168.1.1
```

再执行 nslookup www.baidu.com 命令，现象如下：

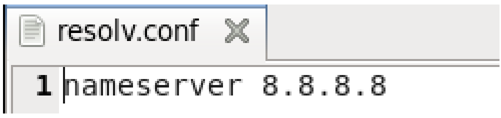
```
[root@FriendlyARM /]# nslookup www.baidu.com
Server:     192.168.0.1
^C
[root@FriendlyARM /]# cat /etc/resolv.conf
nameserver 192.168.0.1
nameserver 192.168.1.1
[root@FriendlyARM /]# nslookup www.baidu.com
Server:     192.168.0.1
```

```
Address 1: 192.168.0.1

Name:       www.baidu.com
Address 1: 180.97.33.107
Address 2: 180.97.33.108
[root@FriendlyARM /]# ping www.baidu.com
PING www.baidu.com (180.97.33.108): 56 data bytes
64 bytes from 180.97.33.108: seq=0 ttl=54 time=113.604 ms
64 bytes from 180.97.33.108: seq=1 ttl=54 time=74.999 ms
64 bytes from 180.97.33.108: seq=2 ttl=54 time=73.848 ms
64 bytes from 180.97.33.108: seq=3 ttl=54 time=70.829 ms
64 bytes from 180.97.33.108: seq=4 ttl=54 time=68.763 ms
64 bytes from 180.97.33.108: seq=5 ttl=54 time=69.728 ms
64 bytes from 180.97.33.108: seq=6 ttl=54 time=62.188 ms
^C
--- www.baidu.com ping statistics ---
7 packets transmitted, 7 packets received, 0% packet loss
round-trip min/avg/max = 62.188/76.279/113.604 ms
[root@FriendlyARM                                                  /]#
```



```
[root@FriendlyARM /]# nslookup www.baidu.com
Server:    192.168.0.1
^C
[root@FriendlyARM /]# cat /etc/resolv.conf
nameserver 192.168.0.1
nameserver 192.168.1.1
nameserver 8.8.8.8
[root@FriendlyARM /]# nslookup www.baidu.com
Server:    192.168.0.1
Address 1: 192.168.0.1

Name:      www.baidu.com
Address 1: 180.97.33.107
Address 2: 180.97.33.108
[root@FriendlyARM /]# ping www.baidu.com
PING www.baidu.com (180.97.33.108): 56 data bytes
64 bytes from 180.97.33.108: seq=0 ttl=54 time=113.604 ms
64 bytes from 180.97.33.108: seq=1 ttl=54 time=74.999 ms
64 bytes from 180.97.33.108: seq=2 ttl=54 time=73.848 ms
64 bytes from 180.97.33.108: seq=3 ttl=54 time=70.829 ms
64 bytes from 180.97.33.108: seq=4 ttl=54 time=68.763 ms
64 bytes from 180.97.33.108: seq=5 ttl=54 time=69.728 ms
64 bytes from 180.97.33.108: seq=6 ttl=54 time=62.188 ms
^C
--- www.baidu.com ping statistics ---
7 packets transmitted, 7 packets received, 0% packet loss
round-trip min/avg/max = 62.188/76.279/113.604 ms
[root@FriendlyARM /]#
```

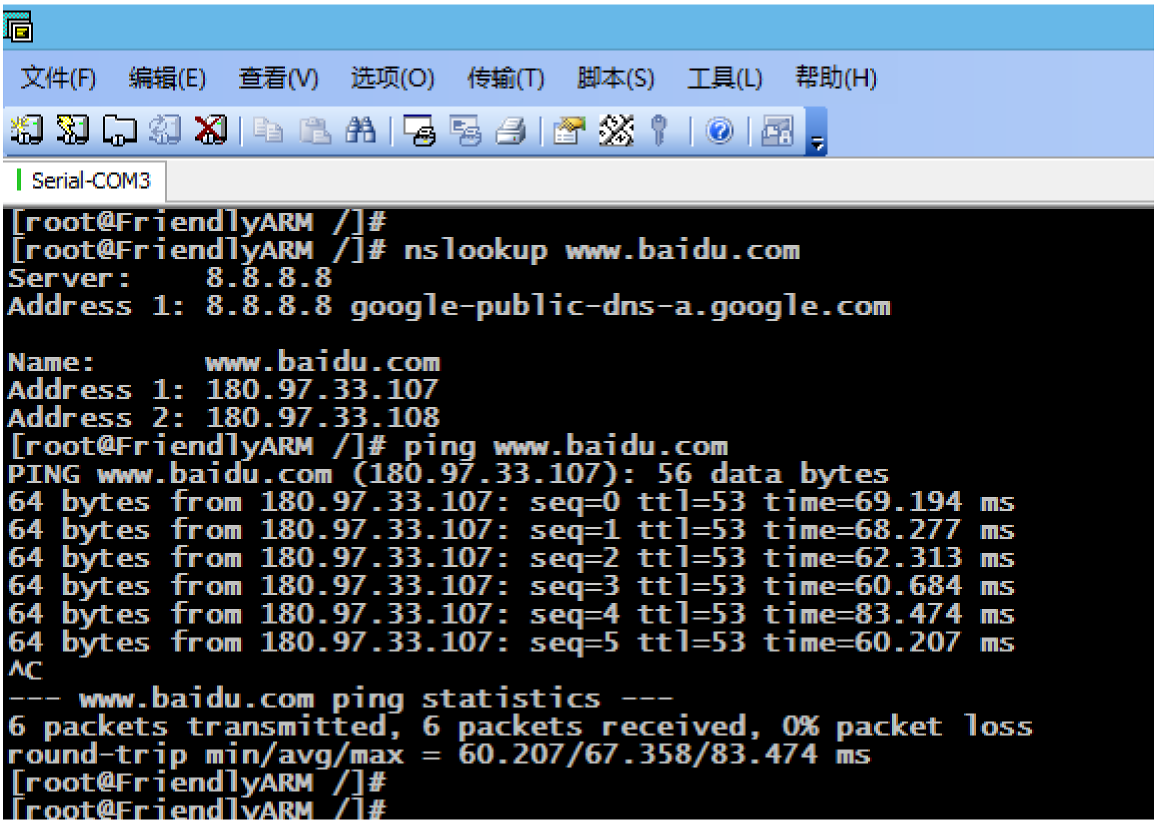
上面是测试本地网关，可以实现 DNS 解析，现在测试第三方 DNS 服务器情况。修改 resolv.conf 文件，如下：



```
nameserver 8.8.8.8
```

设置为谷歌的 DNS 服务器。

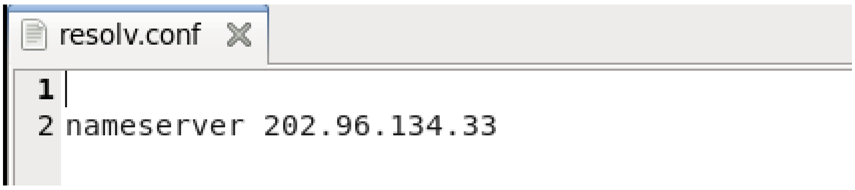
```
 [root@FriendlyARM /]# nslookup www.baidu.com
Server:    8.8.8.8
Address 1: 8.8.8.8 google-public-dns-a.google.com

Name:      www.baidu.com
Address 1: 180.97.33.107
Address 2: 180.97.33.108
[root@FriendlyARM /]#
```



```
[root@FriendlyARM /]#
[root@FriendlyARM /]# nslookup www.baidu.com
Server:    8.8.8.8
Address 1: 8.8.8.8 google-public-dns-a.google.com

Name:      www.baidu.com
Address 1: 180.97.33.107
Address 2: 180.97.33.108
[root@FriendlyARM /]# ping www.baidu.com
PING www.baidu.com (180.97.33.107): 56 data bytes
64 bytes from 180.97.33.107: seq=0 ttl=53 time=69.194 ms
64 bytes from 180.97.33.107: seq=1 ttl=53 time=68.277 ms
64 bytes from 180.97.33.107: seq=2 ttl=53 time=62.313 ms
64 bytes from 180.97.33.107: seq=3 ttl=53 time=60.684 ms
64 bytes from 180.97.33.107: seq=4 ttl=53 time=83.474 ms
64 bytes from 180.97.33.107: seq=5 ttl=53 time=60.207 ms
^C
--- www.baidu.com ping statistics ---
6 packets transmitted, 6 packets received, 0% packet loss
round-trip min/avg/max = 60.207/67.358/83.474 ms
[root@FriendlyARM /]#
[root@FriendlyARM /]#
```

以下测试使用深圳 DNS 服务器解析。

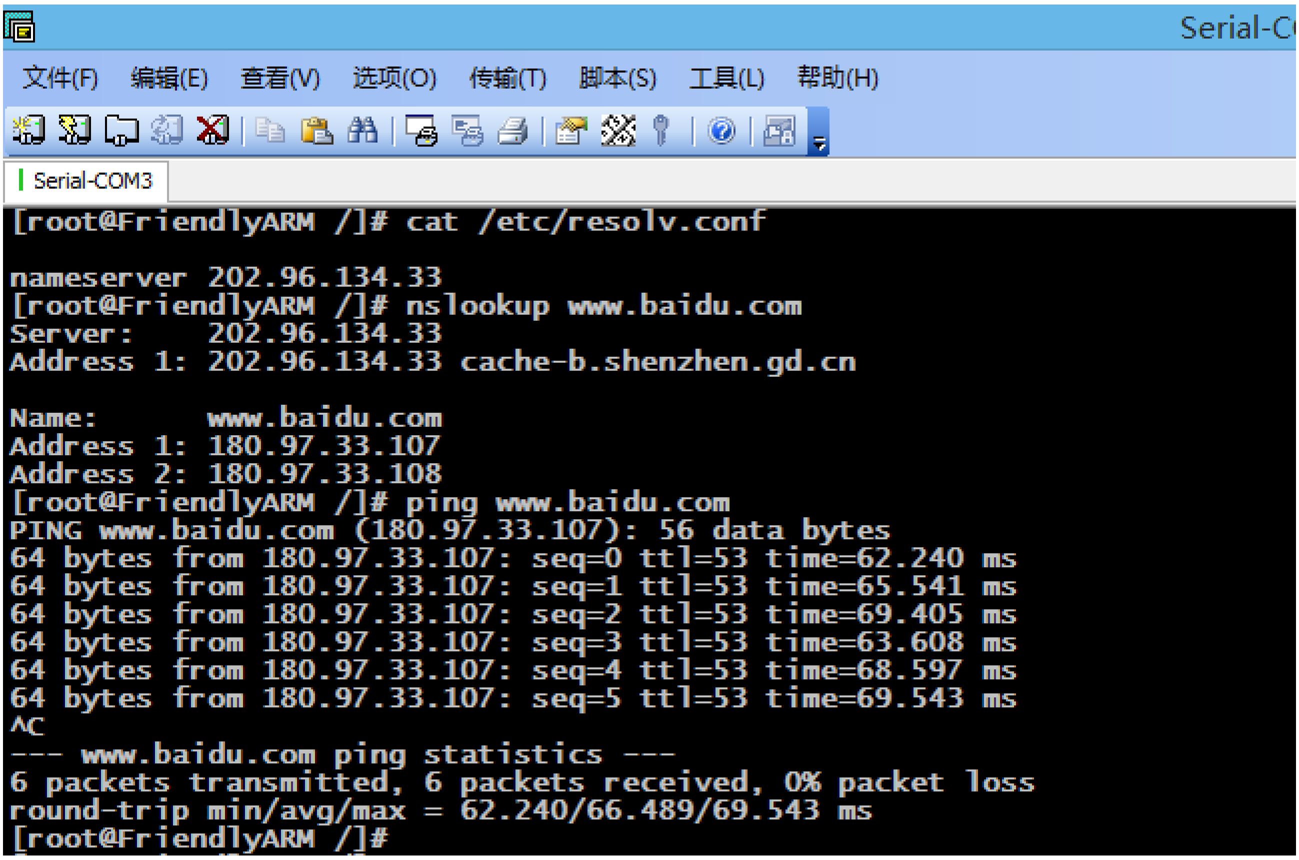
修改 resolv.conf 文件，



```

nameserver 202.96.134.33
```

```
[root@FriendlyARM /]# cat /etc/resolv.conf

nameserver 202.96.134.33
```

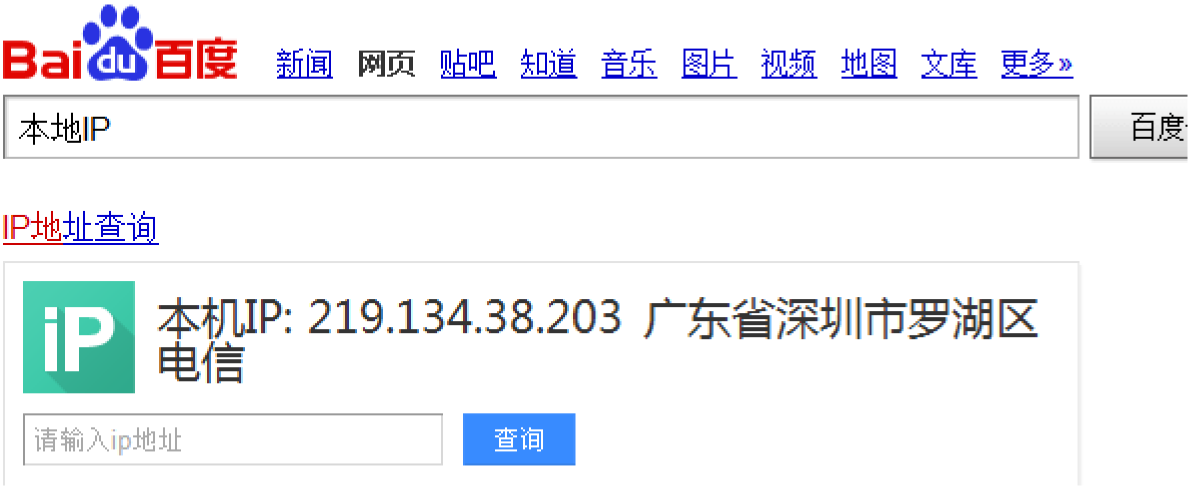
```
[root@FriendlyARM /]# nslookup www.baidu.com
Server:       202.96.134.33
Address 1: 202.96.134.33 cache-b.shenzhen.gd.cn

Name:         www.baidu.com
Address 1: 180.97.33.107
Address 2: 180.97.33.108
[root@FriendlyARM /]# ping www.baidu.com
PING www.baidu.com (180.97.33.107): 56 data bytes
64 bytes from 180.97.33.107: seq=0 ttl=53 time=62.240 ms
64 bytes from 180.97.33.107: seq=1 ttl=53 time=65.541 ms
64 bytes from 180.97.33.107: seq=2 ttl=53 time=69.405 ms
64 bytes from 180.97.33.107: seq=3 ttl=53 time=63.608 ms
64 bytes from 180.97.33.107: seq=4 ttl=53 time=68.597 ms
64 bytes from 180.97.33.107: seq=5 ttl=53 time=69.543 ms
^C
--- www.baidu.com ping statistics ---
6 packets transmitted, 6 packets received, 0% packet loss
round-trip min/avg/max = 62.240/66.489/69.543 ms
[root@FriendlyARM /]#
```



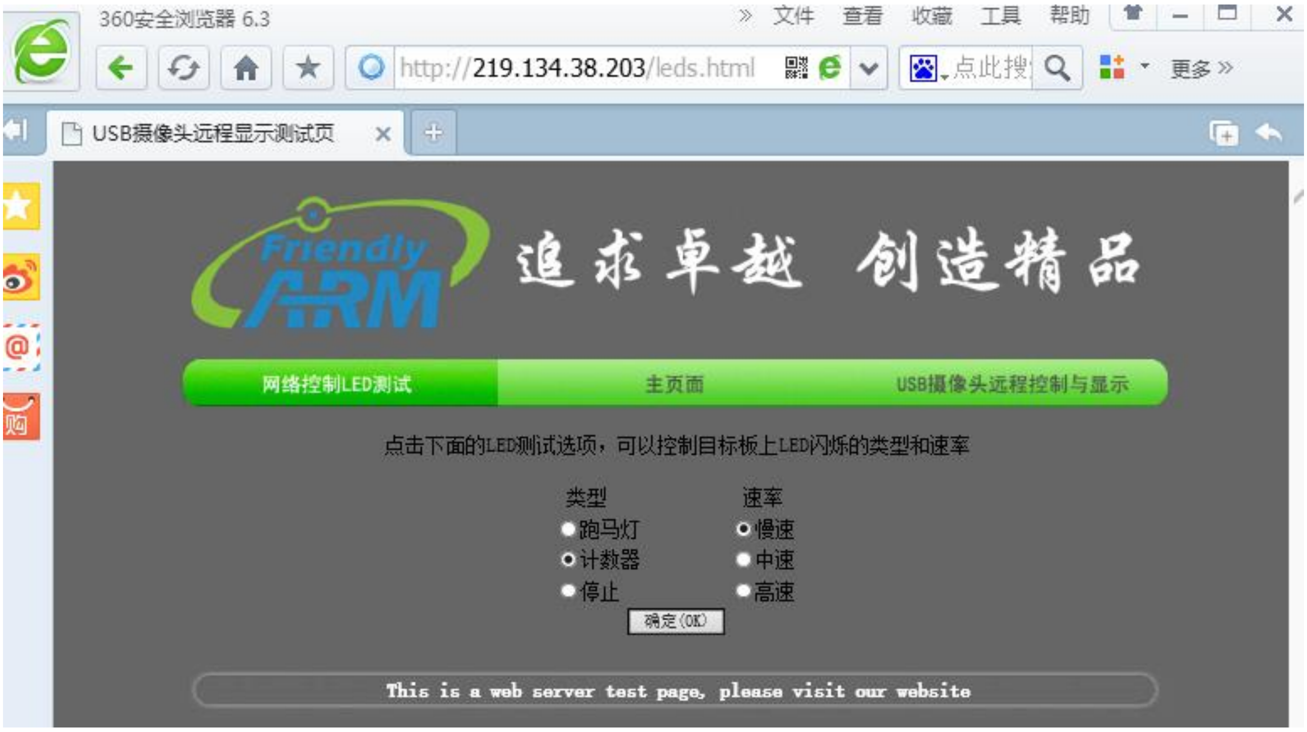
```
[root@FriendlyARM /]# cat /etc/resolv.conf

nameserver 202.96.134.33
[root@FriendlyARM /]# nslookup www.baidu.com
Server:    202.96.134.33
Address 1: 202.96.134.33 cache-b.shenzhen.gd.cn

Name:      www.baidu.com
Address 1: 180.97.33.107
Address 2: 180.97.33.108
[root@FriendlyARM /]# ping www.baidu.com
PING www.baidu.com (180.97.33.107): 56 data bytes
64 bytes from 180.97.33.107: seq=0 ttl=53 time=62.240 ms
64 bytes from 180.97.33.107: seq=1 ttl=53 time=65.541 ms
64 bytes from 180.97.33.107: seq=2 ttl=53 time=69.405 ms
64 bytes from 180.97.33.107: seq=3 ttl=53 time=63.608 ms
64 bytes from 180.97.33.107: seq=4 ttl=53 time=68.597 ms
64 bytes from 180.97.33.107: seq=5 ttl=53 time=69.543 ms
^C
--- www.baidu.com ping statistics ---
6 packets transmitted, 6 packets received, 0% packet loss
round-trip min/avg/max = 62.240/66.489/69.543 ms
[root@FriendlyARM /]#
```

目前存在问题：使用公网 IP,也就是路由外网 IP 可以访问开发板 ，但是访问都电脑必须也是连接同一个路

由器，否则不能访问。以下是设置 boa 端口是 80,如果不是 80，则要修改网址，比如设置为 8080，则下面网址为 http://219.134.38.203:8080/leds.html ,80 则不用加，加上也可以，因为 80 是 web 服务默认端口。



本机IP: 219.134.38.203 广东省深圳市罗湖区电信



http://219.134.38.203/leds.html

点击下面的LED测试选项，可以控制目标板上LED闪烁的类型和速率

类型	速率
○ 跑马灯	● 慢速
● 计数器	○ 中速
○ 停止	○ 高速

This is a web server test page, please visit our website

为解决外网不能访问的原因，和电信交涉后，得到答案是：为了建设和谐， 所以 封了网络的 HTTP，即 80、8080 端口。而以前一直使用的就是 8080 端口，认哉吧！ 运行不好，喝水都会呛着，吃下豆腐都会咽死！各位同志，如果你真遇到这个巧合，那么，恭喜你！以后喝水注意点。哈哈哈!!!!!

发现这个问题后，把 web 服务器端口修改为一个奇特的号 8765 ，竟然使用手机的 GPRS 网络能打开开发板上的网页，太厉害了！很多久前的心病终于去除了。

很简单的一个事情，在很久很久以前都已经懂了。但是，时政关注少了，边国家为共建和谐社会，而突然封杀比较好记的端口号都不知道。唉，看来搞技术还是不能两耳不闻窗外事，一心只想写程序，还得多关

注和谐政策才行！
http://219.134.38.203:8765/cgi-bin/led.cgi 这个是 boa 服务器的地址，前面的 IP 请百度“ 本机 IP 得到”

也可以使用虚拟域名，使用路由的 DNS 功能，
http://fyyy4030.oicp.net:8765/cgi-bin/led.cgi 这个网址也仅作为一个示例，大家根据自己的的实际情况修改。