



目录

编译程序	2
3.1、网卡测试	2
3.1.1 设置静态 IP	3
3.1.2 设置动态 IP	5
3.2、SD 卡	5
3.3、使用 u 盘测试 usb_host	6
3.4、LED	7
3.5、按键 KEY	7
3.6、WATCHDOG	7
3.7、UART 测试程序	8
3.8、I2C	9
3.9、BACKLIGHT 背光控制	10
3.10、GPIO 测试	10
3.11、CAN	11
3.12、RTC	12
3.13、SSH	12
3.14、FTP 获取文件	13
3.15、屏幕重新校准	13
3.16、音频设备	14
3.17、SPI 测试程序	14
3.18、4G 模块	15
3.19、WiFi 模块	18
3.19.1 Wifi 热点发送功能	18
3.19.2 Wifi 连接热点功能	19



编译程序

以下测试方法中所用到的测试程序均提供源码，在 <http://www.embedsky.com/> 网站中的维基教程中可以下载源码，下面是编译源码的步骤（注：编译前需先搭建好编译环境）：

```
#tar xjf test_demo.tar.bz2 -C /
```

```
#cd /opt/EmbedSky/test_demo
```

```
#!/build.sh
```

编译完成之后可以在 `/opt/EmbedSky/test_demo/out` 目录下看到可执行程序，将其拷贝到板卡中直接运行即可达到下面测试同样的效果。

`build.sh` 脚本中默认使用 `arm-linux-gnueabi-hf-gcc` 交叉编译器，如需使用其他交叉编译器，可以修改 `build.sh` 文件，将其中的“`export CROSS_COMPILE=arm-linux-gnueabi-hf-`”改为所用的即可。

3.1、网卡测试

IMX6Q 具有千兆网络接口，测试前先关闭 PC 防火墙，并将 PC 与开发板直接使用网线连接。网口 0 插上网线后会自动分配 ip，板卡串口调试终端使用命令 `ifconfig` 查看 `eth0` 网卡是否有自动分配 ip，如果没有自动分配 ip 需要手动为网卡 0 设置 ip，输入以下命令：`ifconfig eth0 192.168.1.73`（注意要与 pc 机的 ip 同一网段），如下所示：

查看板卡网络信息：

```
root@Embedsky-Board:~# ifconfig
eth0      Link encap:Ethernet  Hwaddr be:d9:95:18:c6:54
          inet addr:192.168.1.237  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::bcd9:95ff:fe18:c654/64  Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:7776 errors:0 dropped:0 overruns:0 frame:0
          TX packets:417 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:598062 (598.0 KB)  TX bytes:39795 (39.7 KB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128  Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:202 errors:0 dropped:0 overruns:0 frame:0
          TX packets:202 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:17182 (17.1 KB)  TX bytes:17182 (17.1 KB)
```

设置板卡 ip:

```
#ifconfig eth0 192.168.1.73
```



设置网关:

```
#route add default gw 192.168.1.2
```

更改 ip 后, 可以 ping 一下网关和百度, 查看网络是否正常,

按 Ctrl+C 则会终止 ping 测试。

```
root@EmbedSky-Board:~# ifconfig eth0 192.168.1.73
root@EmbedSky-Board:~# route add default gw 192.168.1.2
root@EmbedSky-Board:~# ping 192.168.1.2
PING 192.168.1.2 (192.168.1.2) 56(84) bytes of data.
64 bytes from 192.168.1.2: icmp_seq=1 ttl=64 time=0.232 ms
64 bytes from 192.168.1.2: icmp_seq=2 ttl=64 time=0.204 ms
^C
--- 192.168.1.2 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 999ms
rtt min/avg/max/mdev = 0.204/0.218/0.232/0.014 ms
root@EmbedSky-Board:~# ping www.baidu.com
PING www.a.shifen.com (14.215.177.38) 56(84) bytes of data.
64 bytes from 14.215.177.38: icmp_seq=1 ttl=56 time=7.01 ms
64 bytes from 14.215.177.38: icmp_seq=2 ttl=56 time=6.80 ms
^C
--- www.a.shifen.com ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1001ms
rtt min/avg/max/mdev = 6.800/6.908/7.016/0.108 ms
```

注: 如果可以 ping 通网关而 ping 不通百度, 说明网关和 DNS 未设置正确,

使用以下命令设置:

设置网关:

```
#route add default gw 192.168.1.2
```

设置 DNS 的命令:

```
#echo "nameserver 223.5.5.5" > /etc/resolv.conf
```

(注: 根据实际情况填写, 223.5.5.5 为阿里的 DNS 服务器)

3.1.1 设置静态 IP

首先确保在 root 用户下操作。

(1) 修改配置文件/etc/network/interfaces

输入命令: # vi /etc/network/interfaces

```
auto eth0
iface eth0 inet static
address 192.168.1.73
netmask 255.255.255.0
gateway 192.168.1.2
~
```

开发板默认设置 eth0、eth1 网口为动态分配 ip, 如图所示:



```
# wired or wireless interfaces
auto eth0
iface eth0 inet dhcp
iface eth1 inet dhcp
```

这里将 eth0 改为静态获取, 添加以下内容:

```
auto eth0                #设置自动启动 eth0 网口

iface eth0 inet static    #配置静态 IP

address 192.168.1.73     #IP 地址

netmask 255.255.255.0    #子网掩码

gateway 192.168.1.2      #默认网关
```

注意要把 “iface eth0 inet dhcp” 删掉或注释掉, 修改后如图所示:

```
# wired or wireless interfaces
#auto eth0
#iface eth0 inet dhcp
iface eth1 inet dhcp

auto eth0
iface eth0 inet static
address 192.168.1.73
netmask 255.255.255.0
gateway 192.168.1.2
```

(2) 修改 DNS 信息, 输入命令:

```
#echo "nameserver 223.5.5.5" > /etc/resolv.conf
```

(注: 根据实际情况填写, 223.5.5.5 为阿里的 DNS 服务器)

(3) 重启网络, 使配置生效

```
#/etc/init.d/networking restart
```

```
root@Embedsky:/etc/network# /etc/init.d/networking restart
Running /etc/init.d/networking restart is deprecated because it may not enable again some
interfaces
Reconfiguring network interfaces...
fec 2188000.ethernet eth0: Freescale FEC PHY driver [Atheros 8035 ethernet] (mii_bus:phy_a
ddr=2188000.ethernet:00, irq=-1)
done.
root@Embedsky:/etc/network# fec 2188000.ethernet eth0: Link is up - 1Gbps/Full - flow cont
rol rx/tx
```

(4) 验证配置结果:



```
root@Embedsky:~# df
Filesystem      1K-blocks    Used Available Use% Mounted on
/dev/root        5951492  795580   5139528  14% /
devtmpfs         833752      4      833748   1% /dev
tmpfs            1030564     256    1030308   1% /run
tmpfs            1030564     136    1030428   1% /var/volatile
/dev/mmcblk3p1   91942       8786    83156   10% /boot
/dev/mmcblk3p3  1133528     1792   1072492   1% /recovery
/dev/mmcblk1p1  15473096   4256   15468840   1% /run/media/mmcblk1p1
```

使用命令进入挂载目录: `cd /run/media/mmcblk1p1/`, 使用命令进行写操作创建一个 `tianqian.txt` 文档:
`touch tianqian.txt`, 使用 `ls` 命令查看目录内容, 创建成功 SD 卡测试成功。如图:

```
root@Embedsky:~# cd /run/media/mmcblk1p1
root@Embedsky:/run/media/mmcblk1p1# ls
ceshi ceshi.rar test

root@Embedsky:/run/media/mmcblk1p1# touch tianqian.txt
root@Embedsky:/run/media/mmcblk1p1# ls
ceshi ceshi.rar test tianqian.txt
```

3.3、使用 u 盘测试 `usb_host`

将 U 盘格式化成 `fat` 格式, 插入 u 盘查看调试串口信息, 会打印 `usb` 的 u 盘挂载相关信息, 如图:

```
root@Embedsky:/# usb 1-1.2: new high-speed USB device number 3 using ci_hdrc
usb 1-1.2: New USB device found, idVendor=0bda, idProduct=0150
usb 1-1.2: New USB device strings: Mfr=1, Product=2, SerialNumber=3
usb 1-1.2: Product: USB2.0-CRW
usb 1-1.2: Manufacturer: Generic
usb 1-1.2: SerialNumber: 20120926571200000
usb-storage 1-1.2:1.0: USB Mass Storage device detected
scsi host1: usb-storage 1-1.2:1.0
scsi 1:0:0:0: Direct-Access          Generic- Multi-Card          1.00 PQ: 0 ANSI: 4
sd 1:0:0:0: [sda] 31116288 512-byte logical blocks: (15.9 GB/14.8 GiB)
sd 1:0:0:0: [sda] Write Protect is off
sd 1:0:0:0: [sda] write cache: disabled, read cache: enabled, doesn't support DPO or FUA
sda: sda1
sd 1:0:0:0: [sda] Attached SCSI removable disk
```

使用命令 `df` 查看 U 盘挂载目录:

```
root@Embedsky:/# df
Filesystem      1K-blocks    Used Available Use% Mounted on
/dev/root        5951492  795580   5139528  14% /
devtmpfs         833752      4      833748   1% /dev
tmpfs            1030564     264    1030300   1% /run
tmpfs            1030564     136    1030428   1% /var/volatile
/dev/mmcblk3p1   91942       8786    83156   10% /boot
/dev/mmcblk3p3  1133528     1792   1072492   1% /recovery
192.168.1.74:/nfsroot 163975680 16371072 139252096 11% /mnt
/dev/sda1       15473096   895880  14577216   6% /run/media/sda1
```

使用命令进入挂载目录: `cd /run/media/sda1`, 使用命令进行写操作创建一个 `tq.txt` 文档: `touch tq.txt`,
使用 `ls` 命令查看目录内容, 创建成功 USB 测试成功。如图:



```
root@Embedsky:~/# cd /run/media/sda1
root@Embedsky:/run/media/sda1# touch tq.txt
root@Embedsky:/run/media/sda1# ls
Embedsky.ini
imx6q-sabresd_IMX6_CoreC_for_linux_v3.14.3.dtb
rootfs-qt5_IMX6_for_linux_v3.8.2.img
tq.txt
u-boot_IMX6_for_linux_v3.14.5.imx
zImage_IMX6_for_linux_v3.14.5
```

3.4、LED

控制 LED 只是设置一下高低电平即可。系统有 3 个 led 灯，具体操作是一样的，只有路径有不一样，led4 的操作目录是/sys/class/leds/LED4/，led5 的操作目录是/sys/class/leds/LED5/，led6 的操作目录是/sys/class/leds/LED6/。下面以 led4 为例，简单说明如何使用：

```
root@Embedsky:~/# cd /sys/class/leds/
root@Embedsky:/sys/class/leds# ls
led4 led5 led6 mmc1:: mmc2:: mmc3::
root@Embedsky:/sys/class/leds# echo 0 > /sys/class/leds/led4/brightness
root@Embedsky:/sys/class/leds# echo 1 > /sys/class/leds/led4/brightness
```

执行 ech 0 为熄灭，执行 ech 1 为点亮。

3.5、按键 KEY

在命令行中执行：./key_test

按下按键，终端打印按键按下，按键弹起，按回车键退出程序。

```
root@Embedsky:/opt/EmbedSky/test_demo/out# ./key_test
=====START TEST KEY=====
Press ENTER or type command to continue
key 102 Pressed
key 102 Released
key 116 Pressed
key 116 Released
key 1 Pressed
key 1 Released
key 158 Pressed
key 158 Released
```

3.6、WATCHDOG

执行命令：#./watchdog_test



程序会到/dev/watchdog 目录找看门狗设备节点，如果没找到，则会打印 watchdog no found，如果找到，则会打开看门狗设备，喂狗，重启系统。

3.7、UART 测试程序

imx6Q 板有 5 个串口，其中 com0 为调试串口(注：串口 0 只要接上串口线连接到 PC，打开 secureCRT 等串口终端软件，有信息输出即正常)，如下图所示：



各个串口对应的设备节点如下所示：

169	UART1_TXD(SD3_DAT7)	串口1(调试串口)发送	对应设备节点/dev/ttySAC0	170	UART1_RXD(SD3_DAT6)	串口1(调试串口)发送	对应设备节点/dev/ttySAC0
171	UART1_CTS(EIM_D19)	串口1流控管脚		172	UART1_RTS(EIM_D20)	串口1流控管脚	
173	UART2_TXD(EIM_D26)	串口2发送	对应设备节点/dev/ttySAC1	174	UART2_RXD(EIM_D27)	串口2接收	对应设备节点/dev/ttySAC1
175	UART2_CTS(EIM_D28)	串口2流控管脚		176	UART2_RTS(EIM_D29)	串口2流控管脚	
177	UART3_TXD(EIM_D24)	串口3发送	对应设备节点/dev/ttySAC2	178	UART3_RTS(EIM_D25)	串口3接收	对应设备节点/dev/ttySAC2
179	UART4_TXD(KEY_COL0)	串口4发送	对应设备节点/dev/ttySAC3	180	UART4_RXD(KEY_ROW0)	串口4接收	对应设备节点/dev/ttySAC3
181	UART5_TXD(KEY_COL1)	串口5发送	对应设备节点/dev/ttySAC4	182	UART5_RXD(KEY_ROW1)	串口5接收	对应设备节点/dev/ttySAC4

(1) 测试串口收发测试

使用串口 2 发送数据：

```
#./uart_test /dev/ttySAC1 -b 115200 -w "123456"
```



```
"123456"dsky:/opt/EmbedSky/test_demo/out# ./uart_test /dev/ttySAC1 -b 115200 -w
```

```
=====
write only w:123456
boudrate = 115200
interval : 1000 ms
fd-open=3
=====END TEST UART=====
```

其中 “/dev/ttySAC1” 对应串口设备节点，“123456” 为发送的数据，可根据实际需要修改。

使用串口 2 接收数据：

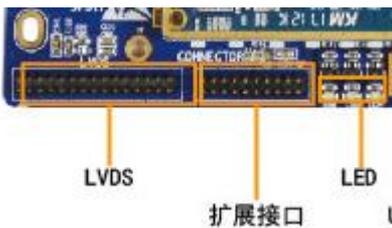
```
#./uart_test /dev/ttySAC1 -b 115200 -r
```

```
root@Embedsky:/opt/EmbedSky/test_demo/out# ./uart_test /dev/ttySAC1 -b 115200 -r
=====
read only
boudrate = 115200
interval : 1000 ms
fd-open=3
=====END TEST UART=====
```

注：如需一直接收或发送可以增加“-l”参数。

3.8、I2C

Imx6Q 板的 I2C 接口位于开发板左下侧（网口朝上放置）的扩展接口（丝印为 CONNECTOR）的第 9、10 引脚，将 6Q 板外接测试模块，例如笔者将其与测试模块的 EPROM 芯片相连接。



开发板的 I2C 设备节点如下所示：

97	I2C2_SCL(KEY_COL3)	IIC2时钟	连接声卡芯片，对应i2c节点为/dev/i2c-1	98	I2C2_SDA(KEY_ROW3)	IIC2数据	连接声卡芯片，对应i2c节点为/dev/i2c-1
99	I2C1_SCL(CSI0_DAT9)	IIC1时钟	连接电容电阻触摸芯片，对应i2c节点为/dev/i2c-0	100	I2C1_SDA(CSI0_DAT8)	IIC1数据	连接电容电阻触摸芯片，对应i2c节点为/dev/i2c-0

在命令行输入：`./i2c_test /dev/i2c-2`

其中/dev/i2c-2 为 I2C 节点设备，测试时根据所要测试的 I2C 选择相应 I2C 节点，测试前确认将 I2C



的 SDA 和 SCL 线连至 I2C 芯片。

测试结果如下所示:

```
root@Embedsky:/test_demo/out# ./i2c_test /dev/i2c-2
=====START TEST I2C=====
Please enter 8 integer data:
eg:11 22 33 44 55 66 77 88
12 22 32 42 52 62 72 82
Writing Data :
12 22 32 42 52 62 72 82
Read Data :
12 22 32 42 52 62 72 82
---> /dev/i2c-2 test ok !
=====END TEST I2C=====
```

3.9、BACKLIGHT 背光控制

设置 lcd 显示屏的背光:

```
#echo xxx >/sys/class/backlight/backlight_lcd/brightness
```

其中 xxx 为 0~94 的数值, 表示背光的亮度, 0 为最暗, 94 为最亮。

设置 lvds 屏幕背光:

```
#echo xxx > /sys/class/backlight/backlight_lvds/brightness
```

其中 xxx 为 0~94 的数值, 表示背光的亮度, 0 为最暗, 94 为最亮。

3.10、GPIO 测试

(1) GPIO ID 的获取: 引脚名称的格式为 GPIOx_y, 其编号等于 $(x-1)*32+y$,

例如: GPIO2_7: $(2-1)*32+7=39$, 则其引脚编号为 39.

(2) 控制 GPIO 输出电平:

将 gpio id 为 104 的管脚设置为输出高电平:

```
#!/gpio_test 104 1
```



将 gpio id 为 104 的管脚设置为输出低电平：

```
#!/gpio_test 104 0
```

测试结果如下所示：

```
root@Embedsky:/opt/EmbedSky/test_demo/out# ./gpio_test 104 1
Control GPIO output level:
gpio104, value = 1;
root@Embedsky:/opt/EmbedSky/test_demo/out# ./gpio_test 104 0
Control GPIO output level:
gpio104, value = 0;
```

(3) 读取 GPIO 输入电平：

读取 gpio id 为 104 的管脚输入电平状态：

```
#!/gpio_test 104
```

测试结果如下所示：

```
root@Embedsky:/opt/EmbedSky/test_demo/out# ./gpio_test 104
Read the GPIO input level:
gpio104, value = 0;
```

3.11、CAN

设置 can0（节点为 can0）的步骤如下：

(1) 设置波特率之前必须先关闭 can 口：

```
# ifconfig can0 down
```

(2) 设置波特率为 1000000：

```
# ip link set can0 type can bitrate 1000000 triple-sampling on
```

(3) 打开 can 口：

```
# ifconfig can0 up
```

设置其他节点（如 can1 等）只需将上面的 can0 替换掉即可。

接收：

```
# candump can0 & #监听 can0
```

发送 can 数据，其中 can id 为 111，数据为：1122334455667788 ：

```
# cansend can0 111#1122334455667788 #can0 发送数据
```



3.12、RTC

读取系统时间:

```
#data
```

设置系统时间:

```
#date -s "YYYY-MM-DD XX:XX:XX" (年月日 时分秒)
```

将系统时间写入到硬件 RTC 芯片中:

```
#hwclock -w
```

读取硬件 RTC 时间:

```
#hwclock -r
```

3.13、SSH

1.先在开发板中配置好网络,例如下面命令是将开发板的 ip 设置为 192.168.1.236:

```
#ifconfig eth0 192.168.1.236
```

2.然后在 PC 中执行 `ssh root@192.168.1.236` 可以登录开发板(前提是网络能够互相 ping 通)

3.第一次登录对方主机,系统一般会出现下面的提示:

```
root@ubuntu:/home/ljh/test_demo_20181204/out# ssh root@192.168.1.236
The authenticity of host '192.168.1.236 (192.168.1.236)' can't be established.
RSA key fingerprint is SHA256:2r76nxsVip1lKeTgE0hcJb7pN2HDE201ICxORXRJ06U.
Are you sure you want to continue connecting (yes/no)? yes
```

输入 yes,

```
Warning: Permanently added '192.168.1.236' (RSA) to the list of known hosts.
root@Embedsky:~# evdevtouch: Cannot open input device /dev/input/event1 (No such
file or directory)
Unable to query physical screen size, defaulting to 100 dpi.
To override, set QT_QPA_EGLFS_PHYSICAL_WIDTH and QT_QPA_EGLFS_PHYSICAL_HEIGHT (i
n millimeters).
```



按 ctrl+c, 即可看到登录成功:

```
root@Embedsky:/# ls
bin    etc    lib    mnt    recovery  sbin    tmp    var
boot  home  lost+found  opt    root    sys    unit_tests
dev    init  media  proc   run     test file  usr
```

退出 SSH 的时候输入“~”与“Ctrl-Z”组合。注意: 当输入“~”时不会立即在屏幕上看到, 只有当你按下 <Ctrl-Z> 并且按回车之后才一起显示。如下, 在远程主机中以此输入“~<Ctrl-Z>”:

```
root@Embedsky:/# ~^Z [suspend ssh]
[1]+  Stopped                  ssh root@192.168.1.236
root@ubuntu:/home/ljh/test_demo_20181204/out#
```

3.14、FTP 获取文件

按照环境搭建章节搭建好 ftp 服务器, 并开启 ftp 服务: #service vsftpd start

先在 PC 机上新建一个文件:

```
root@ubuntu:/opt/ftp# touch 111.txt
```

然后在开发板上执行以下命令可以获取电脑中的文件:

```
#ftpget -u ftpname -p 123 192.168.1.xxx filename
```

其中 123 为用户 ftpname 的密码, 192.168.1.xxx 为 ftp 服务器的 ip, filename 为需要获取的文件名。

```
root@Embedsky:/opt# ftpget -u ftpname -p 123 192.168.1.74 111.txt
root@Embedsky:/opt# ls
111.txt  PDA
```

3.15、屏幕重新校准

使用命令删除校准文件: #rm /etc/pointercal

使用命令重启板子: reboot

启动后会自动进入校准程序重新校准



3.16、音频设备

系统的音频架构为 alsa，支持大部分的 alsa 测试方法，由于 alsa 架构比较复杂，这里只介绍最简单的测试方式：

1. 播放音乐

在命令行输入：`#aplay /usr/share/sounds/alsa/Front_Center.wav`

可以播放该 wav 文件。

```
enter.wavdsky:/opt/Embedsky/test_demo/out# aplay /usr/share/sounds/alsa/Front_Ce
Playing WAVE '/usr/share/sounds/alsa/Front_Center.wav' : Signed 16 bit Little Endian, Rate 48000 Hz, Mono
```

注意：aplay 不可以播放 mp3 文件。

2. 查看控制单元

amixer 工具主要完成控制部分，命令行执行：`amixer controls`

```
root@Embedsky:/# amixer controls
numid=8,iface=MIXER,name='Headphone Mux'
numid=6,iface=MIXER,name='Headphone Playback ZC Switch'
numid=5,iface=MIXER,name='Headphone Playback Volume'
numid=1,iface=MIXER,name='PCM Playback Volume'
numid=7,iface=MIXER,name='Mic Volume'
numid=3,iface=MIXER,name='Capture Attenuate Switch (-6dB)'
numid=9,iface=MIXER,name='Capture Mux'
numid=4,iface=MIXER,name='Capture ZC Switch'
numid=2,iface=MIXER,name='Capture Volume'
```

3. 设置耳机音量

`amixer cget numid=$id`

使用以上命令查看相关属性的设置的情况。

`amixer cset numid=$id $val`

使用以上命令进行相关属性的设置。

从上面第 2 步“查看控制单元”可知耳机音量控制单元为：

`numid=5,iface=MIXER,name='Headphone Playback Volume'`

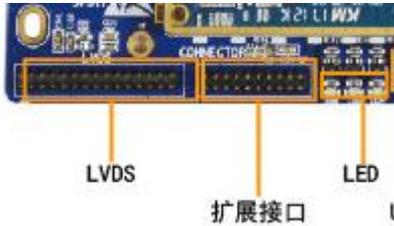
3.17、SPI 测试程序

Imx6Q 开发板只有 1 个 SPI 节点：`/dev/spidev1.0`，测试时短接 MISO 和 MOSI 引脚，进行自收发测试。

Imx6Q 板的 SPI 接口位于开发板左下侧（网口朝上放置）的扩展接口（丝印为 CONNECTOR），其中第



3、5 引脚分别为 MISO 和 MOSI:



测试命令为：`#!/spi_test`

```
root@Embedsky:/opt/Embedsky/test_demo/out# ./spi_test
SPI - Open succeed. Start Init SPI...
spi mode: 0
bits per word: 8
max speed: 12000 KHz (12 MHz)
```

```
FF FF FF FF FF FF
40 00 00 00 00 95
FF FF FF FF FF FF
FF FF FF FF FF FF
FF FF FF FF FF FF
DE AD BE EF BA AD
F0 0D
```

【注：】如果没有短接，输出的数组为 0。

3.18、4G 模块

IMX6Q 板配套的 4G 模块是有方的 N720。

连接“有方 N720”4G 模块（包括 SIM 卡和天线），如图所示：



N720 模块使用以下命令：

```
#pppd call n720_init &
```

```
root@Embedsky:/test_file# cd /etc/ppp/peers/
root@Embedsky:/etc/ppp/peers# pppd call n720_init &

rcvd [IPCP ConfReq id=0x0]
sent [IPCP ConfNak id=0x0 <addr 0.0.0.0>]
rcvd [IPCP ConfReq id=0x1 <compress VJ Of 01>]
sent [IPCP ConfReq id=0x2 <addr 0.0.0.0> <ms-dns1 0.0.0.0> <ms-dns2 0.0.0.0>]
rcvd [IPCP ConfReq id=0x1]
sent [IPCP ConfAck id=0x1]
rcvd [IPCP ConfNak id=0x2 <addr 10.10.151.69> <ms-dns1 202.96.128.86> <ms-dns2 202.96.134.133>]
sent [IPCP ConfReq id=0x3 <addr 10.10.151.69> <ms-dns1 202.96.128.86> <ms-dns2 202.96.134.133>]
rcvd [IPCP ConfAck id=0x3 <addr 10.10.151.69> <ms-dns1 202.96.128.86> <ms-dns2 202.96.134.133>]
Could not determine remote IP address: defaulting to 10.64.64.64
local IP address 10.10.151.69
remote IP address 10.64.64.64
primary DNS address 202.96.128.86
secondary DNS address 202.96.134.133
Script /etc/ppp/ip-up started (pid 1070)
Script /etc/ppp/ip-up finished (pid 1070), status = 0x63
```

如看到如上类似的 LOG，表示拨号成功，这时候输入 ifconfig 可以看到生成了一个 ppp0 节点。



```
root@Embedsky:/test_file# ifconfig
eth0      Link encap:Ethernet  Hwaddr DA:18:F2:46:63:A3
          UP BROADCAST MULTICAST  MTU:1500  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:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:34 errors:0 dropped:0 overruns:0 frame:0
          TX packets:34 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:0
          RX bytes:2380 (2.3 KiB)  TX bytes:2380 (2.3 KiB)

ppp0      Link encap:Point-to-Point Protocol
          inet addr:10.10.151.69  P-t-P:10.64.64.64  Mask:255.255.255.255
          UP POINTOPOINT RUNNING NOARP MULTICAST  MTU:1500  Metric:1
          RX packets:5 errors:0 dropped:0 overruns:0 frame:0
          TX packets:6 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:3
          RX bytes:62 (62.0 B)  TX bytes:101 (101.0 B)
```

以上步骤成功之后, 只能 ping 通 ip, 无法 ping 通百度:

```
root@Embedsky:/test_file# ping 202.96.128.86
PING 202.96.128.86 (202.96.128.86): 56 data bytes
64 bytes from 202.96.128.86: seq=0 ttl=55 time=348.984 ms
64 bytes from 202.96.128.86: seq=1 ttl=55 time=21.020 ms
64 bytes from 202.96.128.86: seq=2 ttl=55 time=60.280 ms
^C
--- 202.96.128.86 ping statistics ---
3 packets transmitted, 3 packets received, 0% packet loss
round-trip min/avg/max = 21.020/143.428/348.984 ms
```

```
root@Embedsky:/test_file# ping www.baidu.com
ping: bad address 'www.baidu.com'
```

当拨号成功后, 会在/etc/ppp/目录下产生一个 resolv.conf 文件, 里面包含两个 DNS, 替换/etc 下的就可以了, 命令如下: **#cp /etc/ppp/resolv.conf /etc/resolv.conf**

```
root@Embedsky:/test_file# cp /etc/ppp/resolv.conf /etc/resolv.conf
root@Embedsky:/test_file# ping www.baidu.com
PING www.baidu.com (180.101.49.42): 56 data bytes
64 bytes from 180.101.49.42: seq=0 ttl=51 time=55.359 ms
64 bytes from 180.101.49.42: seq=1 ttl=51 time=91.227 ms
^C
--- www.baidu.com ping statistics ---
2 packets transmitted, 2 packets received, 0% packet loss
round-trip min/avg/max = 55.359/73.293/91.227 ms
```

此时即可 ping 通百度。



3.19、WiFi 模块

3.19.1 Wifi 热点发送功能

本方法仅保证适用于 tq-8723-wifi 模块，插上模块，串口终端会打印以下 log：

```
root@Embedsky:/test_file# usb 1-1.2: new high-speed USB device number 4 using ci_hndrc
usb 1-1.2: New USB device found, idVendor=0bda, idProduct=b720
usb 1-1.2: New USB device strings: Mfr=1, Product=2, SerialNumber=3
usb 1-1.2: Product: 802.11n WLAN Adapter
usb 1-1.2: Manufacturer: Realtek
usb 1-1.2: SerialNumber: 00e04c000001
rtk_btusb: btusb_probe intf->cur_altsetting->desc.bInterfaceNumber=0
rtk_btusb: btusb_probe can_wakeup=1      flag2=0
rtk_btusb: patch_add
rtk_btusb: auto suspend is disabled
rtk_btusb: pid = 0xb720
rtk_btusb: set_bit(HCI_QUIRK_RESET_ON_CLOSE, &hdev->quirks);
rtk_btusb: btusb_open start pm_usage_cnt(0x1)
rtk_btusb: btusb_open hdev->promisc ==0
rtk_btusb: download_patch start
rtk_btusb: read_ver_rsp->lmp_subver = 0x8723
rtk_btusb: patch_entry->lmp_sub = 0x8723
rtk_btusb: get_firmware start
rtk_btusb: load_firmware start
rtk_btusb: lmp_version = 0x8723
rtk_btusb: config name is rtl8723bu_config
usb 1-1.2: Direct firmware load for rtl8723bu_config failed with error -2
usb 1-1.2: Falling back to user helper
RTL871X: hal_com_config_channel_plan chplan:0x20
RTL871X: rtw_ndev_init(wlan0) if1 mac_addr=48:46:c1:08:59:7f
RTL871X: rtw_ndev_init(wlan1) if2 mac_addr=4a:46:c1:08:59:7f
rtk_btusb: fw name is rtl8723b_fw
usb 1-1.2: Direct firmware load for rtl8723b_fw failed with error -2
usb 1-1.2: Falling back to user helper
rtk_btusb: get_firmware failed!
rtk_btusb: Rtk patch end -1
rtk_btusb: btusb_open failed pm_usage_cnt(0x0)
IPv6: ADDRCONF(NETDEV_UP): wlan0: link is not ready
RTL871X: set bssid:00:00:00:00:00:00
```

执行脚本：`#!/send_ap.sh`

```
root@Embedsky:~# cd /test_file/
root@Embedsky:/test_file# ./send_ap.sh
kill: not enough arguments
kill: not enough arguments
kill: not enough arguments
Configuration file: /etc/hostapd.conf
IPv6: ADDRCONF(NETDEV_UP): wlan0: link is not ready
Using interface wlan0 with hwaddr 48:46:c1:58:d5:14 and ssid "test"
random: cannot read from /dev/random: Resource temporarily unavailable
random: only 0/20 bytes of strong random data available from /dev/random
random: Not enough entropy pool available for secure operations
WPA: Not enough entropy in random pool for secure operations - update keys later when the first
station connects
RTL871X: assoc success
IPv6: ADDRCONF(NETDEV_CHANGE): wlan0: link becomes ready
wlan0: interface state UNINITIALIZED: set group key camid:1, addr:00:00:00:00:00:00, kid:1,
type:TKIP
ZED->ENABLED
wlan0: AP-ENABLED
root@Embedsky:/test_file# udhcpd (v1.23.2) started
```



【注：】修改账号密码方法：`#vi /etc/hostapd.conf`

（默认账号：test 密码：12345678）

账号对应项：ssid=test 密码对应项：wpa_passphrase=12345678

此时用手机查看 wifi 列表，可看到出现名称为“test”的 ssid，输入密码“12345678”，连接后可看到开发板串口打印出以下信息：

```
RTL871X: set pairwise key camid:4, addr:50:c8:e5:76:99:5d, kid:0, type:AES  
Sending OFFER of 192.168.0.20  
Sending OFFER of 192.168.0.20  
Sending ACK to 192.168.0.20
```

3.19.2 Wifi 连接热点功能

本方法仅保证适用于 tq-8723-wifi 模块，插上模块，在板卡串口终端上依次执行以下命令：

```
#ifconfig wlan0 up
```

```
#wpa_supplicant -i wlan0 -Dnl80211 -c /etc/wpa_supplicant.conf &
```

```
#udhcpc -i wlan0
```

出现以下打印即测试成功：

```
udhcpc (v1.23.2) started  
Sending discover...  
Sending discover...  
Sending discover...  
Sending discover...  
Sending discover...
```

【注：】修改 wifi 用户、密码：`#vim /etc/wpa_supplicant.conf`

（默认账号：CDMA-yalu 密码：1234567890）

账号对应项：ssid="CDMA-yalu"

密码对应项：psk="1234567890"