



## 目录

编译程序.....	2
SD 卡测试.....	3
U 盘测试.....	4
USB 摄像头测试.....	5
LED 测试.....	6
按键测试.....	7
看门狗测试.....	8
UART 测试.....	9
I2C 测试.....	10
背光测试.....	11
GPIO 测试.....	12
CAN 测试.....	13
RTC 测试.....	14
SSH 测试.....	15
FTP 测试.....	16
屏幕校准测试.....	17
显示测试.....	18
音频播放测试.....	19
SPI 测试.....	20
4G 模块测试.....	21
WiFi 模块测试.....	23
Wifi 热点发送功能.....	23
Wifi 连接热点功能.....	24



## 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。



## SD 卡测试

SD 卡格式需要设置为 FAT，板子才会识别并自动挂载。插入 SD 卡时，串口会打印识别和 SD 卡挂载信息，使用命令 `df` 查看 SD 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     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
```



## U 盘测试

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
```



## USB 摄像头测试

目前测试程序只支持 uvc 格式的 usb 摄像头，将 usb 接口接入板卡中，在控制台中输入：

```
#/test_file/uvc_test /dev/videoX
```

其中/dev/videoX 为摄像头的设备节点，根据实际情况设置，执行完成之后可以在显示屏上看到 usb 摄像头采集到的图像。

EmbedSky

天嵌科技



## LED 测试

### 1.查看 led 灯设备

```
#ls /sys/class/leds/
```

查看到的目录除 mmcX::开头的外，其余均是 led 灯设备

### 2.控制 led 灯

控制 LED 只是设置一下高低电平即可。系统下所有的 led 灯操作方法是一样的，只是路径不一样。

控制灯亮：

```
#echo 1 > /sys/class/leds/xxx/brightness
```

控制灯灭：

```
#echo 0 > /sys/class/leds/xxx/brightness
```

其中 xxx 为步骤 1 中查看到的 led 设备。



## 按键测试

在命令行中执行：

```
#!/test_file/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
```



## 看门狗测试

执行命令：

```
#/test_file/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，中断程序后系统则会自动重启。

EmbedSky

天嵌科技



## UART 测试

在 <http://wiki.armbbs.net/tqwiki/public/docs/> 链接中找到所使用的板卡，然后进入”硬件开发指南->pin 脚功能”目录寻找 uart 的管脚和设备节点。

### (1) 测试串口收发测试

使用串口 2 发送数据：

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

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：

```
#/test_file/uart_test /dev/ttySAC1 -b 115200 -r -l
```

下图是对/dev/ttySAC1 进行自收发的运行截图：

```
root@Embedsky:~# ./test_file/uart_test /dev/ttySAC1 -b 115200 -asc -rw "123456" -l
Device [/dev/ttySAC1]
format ascii
read write w:123456
loop true
boudrate = 115200
interval : 1000 ms
/dev/ttySAC1 write :123456

/dev/ttySAC1 read=6 : 123456
/dev/ttySAC1 write :123456

/dev/ttySAC1 read=6 : 123456
/dev/ttySAC1 write :123456
```

注：如需一直接收或发送可以增加“-l”参数。



## I2C 测试

在 <http://wiki.armbbs.net/tqwiki/public/docs/> 链接中找到所使用的板卡，然后进入”硬件开发指南->pin 脚功能”目录寻找 i2c 的管脚和设备节点，例如笔者将其与测试模块的 EPROM 芯片相连接。

在命令行输入：

```
#/test_file/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=====
```



## 背光测试

### 1.查看背光设备

```
#ls /sys/class/backlight/
```

查看到的目录背光设备，每个目录表示一个设备。

### 2.查看可设置的最亮值

```
#cat /sys/class/backlight/xxx/max_brightness
```

其中 xxx 为步骤 1 中查看到的设备。

### 3.查看当前背光亮度值：

```
#cat /sys/class/backlight/xxx/brightness
```

其中 xxx 为步骤 1 中查看到的设备。

### 4.设置背光亮度值：

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

其中 xxx 为步骤 1 中查看到的设备，N 为背光亮度值，最大值为步骤 2 获取到的值。



## 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:

```
#/test_file/gpio_test 104 1
```

将 gpio id 为 104 的管脚设置为输出低电平:

```
#/test_file/gpio_test 104 0
```

(3) 读取 GPIO 输入电平:

读取 gpio id 为 104 的管脚输入电平状态:

```
#/test_file/gpio_test 104
```

测试结果如下所示:

```
root@Embedsky:~# /test_file/gpio_test 104 1
pin 104 set 1
root@Embedsky:~# /test_file/gpio_test 104 0
pin 104 set 0
root@Embedsky:~# /test_file/gpio_test 104
pin 104 read 0
```



## CAN 测试

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 &
```

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

```
#cansend can0 111#1122334455667788
```



## RTC 测试

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
```

EmbedSky

天嵌科技



## SSH 测试

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#
```



## FTP 测试

按照 WIKI 教程中的环境搭建章节搭建好 ftp 服务器

获取 pc 机服务器上的文件，执行以下命令：

```
#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
```



## 屏幕校准测试

运行校准程序：

```
#ts_calibrate
```

运行完之后可以在屏幕中看到校准准心，点击完 5 次准心后完成校准

运行校准测试程序：

```
#ts_test
```

运行完之后可以点击屏幕，屏幕上的准心会跟随触摸走动

Embedsky

天嵌科技



## 显示测试

运行测试 fb\_test 可在显示屏上显示一张 bmp 图, 使用方法如下:

```
#/test_file/fb_test /test_file/logo.bmp
```

执行完后可以看到文件/test\_file/logo.bmp 在屏幕上显示。

EmbedSky

天嵌科技



## 音频播放测试

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'
```



## SPI 测试

在 <http://wiki.armbbs.net/tqwiki/public/docs/> 链接中找到所使用的板卡，然后进入”硬件开发指南->pin 脚功能”目录寻找 spi 的管脚和设备节点。

测试命令为：

```
#/test_file/spi_test xxx
```

其中 xxx 通过 WIKI “pin 脚功能”章节查看到的设备节点。

例如：

```
#/test_file/spi_test /dev/spidev1.0
```

测试结果如下所示：

```
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。



## 4G 模块测试



板卡配套的 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 ConfRej id=0x1 <compress VJ 0f 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。



## WiFi 模块测试

### 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_hdc
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_ndevice_init(wlan0) if1 mac_addr=48:46:c1:08:59:7f
RTL871X: rtw_ndevice_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：

```
#/test_file/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
```

## 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：

```
#vi /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"