

# T40xp 双 sensor 开发板 wifi 测试

## 目录

|                                 |   |
|---------------------------------|---|
| 1 内核修改及配置.....                  | 1 |
| 1.1 支持 wifi 模块.....             | 1 |
| 1.2 支持 AP 模式 iptables 内核配置..... | 2 |
| 2 wifi 驱动编译.....                | 2 |
| 3 wifi 测试.....                  | 2 |
| 3.1 wifi 工具使用 .....             | 2 |
| 3.2 STA 模式测试 .....              | 3 |
| 3.3 AP 模式测试 .....               | 4 |
| 4 iptables 使用 .....             | 6 |
| 4.1 iptables 内核配置 .....         | 6 |
| 4.2 iptables 交叉编译 .....         | 6 |
| 4.3 注意事项.....                   | 7 |

说明：wifi 测试可参照 SDK 中《T 系列芯片 WiFi 移植指南.pdf》(路径：T40\SDK\ISVP-T40-1.1.1-20220120\software\doc\BSP\ T 系列芯片 WiFi 移植指南.pdf)

## 1 内核修改及配置

### 1.1 支持 wifi 模块

make menuconfig

(1) 检查 MSC 功能内核配置(默认支持):

Device Drivers --->

<\*> MMC/SD/SDIO card support --->

<\*> Ingenic(XBurst2) MMC/SD Card Controller(MSC) supp

ort

(2)

设备树开启 MSC1

文件：arch/mips/boot/dts/ingenic/shark.dts

```

&msc1 {
    status = "ok";
    pinctrl-names = "default";
    /*mmc-hs200-1 8v;*/
    cap-mmc-highspeed;
    max-frequency = <50000000>;
    bus-width = <4>;
    voltage-ranges = <1800 3300>;
    non-removable;

    ingenic,sdio_clk=<1>;

    /* special property */
    ingenic,wp-gpios = <0>;
    ingenic,rst-gpios = <0>;
    pinctrl-0 = <&msc1_pb>;
};

```

- ① 将 status 配置为 okay，使能 msc1 的功能
- ② 添加 ingenic,sdio\_clk = <1>; 这个属性
- ③ T40 上有两组 GPIO 可以配制成 msc1 功能，分别为 PB 和 PC，根据具体的硬件设计将 pinctrl-0 配置为 msc1\_pb 或 msc1\_pc。

## 1.2 支持 AP 模式 iptables 内核配置

iptables 内核配置

make menuconfig

Networking support -->

Networking options -->

[\*] Network packet filtering framework(Netfilter) -->

Core Netfilter Configuration --> //里面的所有选项全选

IP:Netfilter Configuration --> //里面的所有选项全选

## 2 wifi 驱动编译

下载: <http://xstrive.tpddns.cn:5000/share.cgi?ssid=ce803f25ba3d4cf395819548c8a1bb10>

执行 make clean;make

## 3 wifi 测试

### 3.1 wifi 工具使用

(1)下载路径: T40\SDK\ISVP-T40-1.1.1-20220120\software\sdk\Ingenic-SDK-T40-1.1.1-20220120\resource\tools\_t40\bin\uclibc\wifi

| « 项目管理 > T40 > SDK > ISVP-T40-1.1.1-20220120 > software > sdk > Ingenic-SDK-T40-1.1.1-20220120 > resource > tools_t40 > bin > uclibc > wifi |                  |     |        |  |
|---------------------------------------------------------------------------------------------------------------------------------------------|------------------|-----|--------|--|
| 名称                                                                                                                                          | 修改日期             | 类型  | 大小     |  |
| lib                                                                                                                                         | 2022-01-20 22:27 | 文件夹 |        |  |
| hostapd                                                                                                                                     | 2022-01-20 22:27 | 文件  | 654 KB |  |
| iwconfig                                                                                                                                    | 2022-01-20 22:27 | 文件  | 20 KB  |  |
| iwgetid                                                                                                                                     | 2022-01-20 22:27 | 文件  | 9 KB   |  |
| iwlist                                                                                                                                      | 2022-01-20 22:27 | 文件  | 24 KB  |  |
| iwpriv                                                                                                                                      | 2022-01-20 22:27 | 文件  | 10 KB  |  |
| wpa_cli                                                                                                                                     | 2022-01-20 22:27 | 文件  | 80 KB  |  |
| wpa_supplicant                                                                                                                              | 2022-01-20 22:27 | 文件  | 714 KB |  |

(2) 将 wifi/lib 中的库复制的 root-uclibc-toolchain720/lib/

| T40 > SDK > ISVP-T40-1.1.1-20220120 > software > sdk > Ingenic-SDK-T40-1.1.1-20220120 > resource > tools_t40 > bin > uclibc > wifi > lib |                  |                    |          |  |
|------------------------------------------------------------------------------------------------------------------------------------------|------------------|--------------------|----------|--|
| 名称                                                                                                                                       | 修改日期             | 类型                 | 大小       |  |
|  libcrypto.so.1.0.0                                     | 2022-01-20 22:27 | 0 文件               | 1,859 KB |  |
|  libiw.so.29                                            | 2022-01-20 22:27 | 29 文件              | 29 KB    |  |
|  libnl.so.1                                             | 2022-01-20 22:27 | DxDesigner Desi... | 309 KB   |  |
|  libssl.so.1.0.0                                        | 2022-01-20 22:27 | 0 文件               | 383 KB   |  |

修改权限 `chmod 777 /wifi/*`

将 wifi/中其他可执行文件拷贝到 root-uclibc-toolchain720/bin/  
重新制作 rootfs.squashfs，然后烧写

## 3.2 STA 模式测试

(1) 应用程序及配置文件

`wpa_supplicant` // 负责完成 wifi 认证相关的登录、加密等工作

`wpa_supplicant.conf` // 配置想要连接的 wifi 热点的名称和密码

添加内容：

```
ctrl_interface=/var/run/wpa_supplicant
update_config=1
network={
    ssid="JZ_Guest"
    psk="#wwwingeniccn#"
}
```

(2) 使用到的动态库

`libcrypto.so.1.0.0` // 加密相关

`libiw.so.29` // iw 相关的命令会使用到该库，如 `iwlist`、`iwconfig`

`libnl.so.1` // `hostpad` 需要用到的库

`libssl.so.1.0.0` // `openssl` 需要使用到的库

将动态库放到系统相应的 `lib` 目录下面。

(3) 测试流程

①加载 `8821cs.ko`，加载成功之后会生成 `wlan0` 网卡

②启动网卡 `ifconfig wlan0 up`

③ 将上面的 4 个动态库文件放到 `/system/lib` 目录下

④ `./wpa_supplicant -i wlan0 -Dnl80211 -c wpa_supplicant.conf &`

```
[root@ingenic-uc1_1:samples]# ls
autorun.sh      udhcpd.conf      wpa_supplicant
hostapd.conf    wpa_cli          wpa_supplicant.conf
[root@ingenic-uc1_1:samples]# ./wpa_supplicant -i wlan0 -Dnl80211 -c wpa_supplicant.conf &
[root@ingenic-uc1_1:samples]# Successfully initialized wpa_supplicant
[ 5807.140053] start_addr=(0x8000), end_addr=(0x10000), buffer_size=(0x8000), smp_number_max=(4096)
wlan0: Trying to associate with 68:77:24:38:07:d2 (SSID='xstrive' freq=2437 MHz)
wlan0: ASSOCIATED with 68:77:24:38:07:d2
wlan0: CTRL-EVENT-SUBNET-STATUS-UPDATE status=0
wlan0: WPA: Key negotiation completed with 68:77:24:38:07:d2 [PTK=CCMP GTK=CCMP]
wlan0: CTRL-EVENT-CONNECTED - Connection to 68:77:24:38:07:d2 completed [id=0 id_str=]
```

⑤ `udhcpd -i wlan0` // 动态获取 IP 地址

```
[root@ingenic-uc1_1:samples]# udhcpc -i wlan0
udhcpc (v1.22.1) started
Sending discover...
Sending select for 192.168.0.95...
Lease of 192.168.0.95 obtained, lease time 1800
deleting routers
adding dns 192.168.0.1
[root@ingenic-uc1_1:samples]#
```

⑥ ping www.baidu.com // 网络正常

```
[root@ingenic-uc1_1:samples]# udhcpc -i wlan0
udhcpc (v1.22.1) started
Sending discover...
Sending select for 192.168.0.95...
Lease of 192.168.0.95 obtained, lease time 1800
deleting routers
adding dns 192.168.0.1
[root@ingenic-uc1_1:samples]# ping www.baidu.com
PING www.baidu.com (14.215.177.38): 56 data bytes
64 bytes from 14.215.177.38: seq=0 ttl=54 time=24.553 ms
64 bytes from 14.215.177.38: seq=1 ttl=54 time=25.442 ms
64 bytes from 14.215.177.38: seq=2 ttl=54 time=33.756 ms
64 bytes from 14.215.177.38: seq=3 ttl=54 time=31.939 ms
■
```

### 3.3 AP 模式测试

AP 模式即无线接入点，是一个无线网络的创建者，是网络的中心节点。

(1) 应用程序及配置文件

hostapd // 将 wifi 网卡切换为 master 模式，模拟 AP 功能作为 AP 的认证服务器，负责控制管理 stations 的接入和认证

hostapd.conf // hostapd 的配置文件，生成的 wifi 热点的名称、密码、加密方式等

```
interface=wlan0
ctrl_interface=/var/run/hostapd
ctrl_interface_group=0
ssid=test_wifi
hw_mode=g
channel=1
beacon_int=100
driver=nl80211
macaddr_acl=0
auth_algs=1
ignore_broadcast_ssid=0
own_ip_addr=127.0.0.1
wpa=2
wpa_passphrase=12345678
rsn_pairwise=TKIP CCMP
```

udhcpd // dhcp 的 server，给 dhcp 的 client 分配 IP、网关等

udhcpd.conf // udhcpd 的配置文件，负责 IP、网关、DNS 等的分配  
添加内容：

```
start 192.168.3.201
end 192.168.3.254
interface wlan0
opt dns 192.168.3.200
```

```

option subnet 255.255.255.0
opt router 192.168.3.200
opt wins 114.114.114.114
option dns 114.114.114.114 # appened to above DNS servers for a total of 3
option domain local
option lease 864000

```

## (2) 使用到的动态库

```

libcrypto.so.1.0.0 //加密相关
libnl.so.1 // hostpad 需要用到的库
libssl.so.1.0.0 //openssl 需要使用到的库

```

## (3) 测试流程

- ① 加载 8821cs.ko, 加载成功之后会生成 wlan0 网卡
- ② 启动网卡并配置 IP 地址, ifconfig wlan0 192.168.3.200 up

```

[root@ingenic-uc1_1:samples]# ifconfig wlan0 192.168.0.200 up
[root@ingenic-uc1_1:samples]# ifconfig
lo        Link encap:Local Loopback
  inet addr:127.0.0.1  Mask:255.0.0.0
  UP LOOPBACK RUNNING  MTU:65536  Metric:1
  RX packets:7 errors:0 dropped:0 overruns:0 frame:0
  TX packets:7 errors:0 dropped:0 overruns:0 carrier:0
  collisions:0 txqueuelen:1
  RX bytes:560 (560.0 B)  TX bytes:560 (560.0 B)

wlan0     Link encap:Ethernet  HWaddr 54:EF:33:69:F2:32
  inet addr:192.168.0.200  Bcast:192.168.0.255  Mask:255.255.255.0
  UP BROADCAST MULTICAST  MTU:1500  Metric:1
  RX packets:6945 errors:0 dropped:2845 overruns:0 frame:0
  TX packets:29 errors:0 dropped:0 overruns:0 carrier:0
  collisions:0 txqueuelen:1000
  RX bytes:6044595 (5.7 MiB)  TX bytes:29190 (28.5 KiB)

[root@ingenic-uc1_1:samples]#

```

- ③ 将上面的 3 个动态库文件放到/system/lib 目录下

- ④ ./hostapd -B hostapd.conf // 创建热点

```

[root@ingenic-uc1_1:samples]# ./hostapd -B hostapd.conf
Configuration file: hostapd.conf
[ 7062.169760] start_addr=(0x8000), end_addr=(0x10000), buffer_size=(0x8000), smp_number_max=(4096)
Using interface wlan0 with hwaddr 54:ef:33:69:f2:32 and ssid "test_wifi"
wlan0: interface state UNINITIALIZED->ENABLED
wlan0: AP-ENABLED
[root@ingenic-uc1_1:samples]#

```

- ⑤ udhcpd udhcpd.conf // 配置 dhcp 的 server

- ⑥ 使用手机连接前面生成的热点 test\_wifi, 然后 ping 192.168.3.201

```

[root@ingenic-uc1_1:samples]# ping 192.168.0.201
PING 192.168.0.201 (192.168.0.201): 56 data bytes
64 bytes from 192.168.0.201: seq=0 ttl=64 time=93.592 ms
64 bytes from 192.168.0.201: seq=1 ttl=64 time=186.867 ms
64 bytes from 192.168.0.201: seq=2 ttl=64 time=257.748 ms

--- 192.168.0.201 ping statistics ---

```



## 4 iptables 使用

### 4.1 iptables 内核配置

make menuconfig

Networking support -->

Networking options -->

[\*] Network packet filtering framework(Netfilter) -->

Core Netfilter Configuration --> //里面的所有选项全选

IP:Netfilter Configuration --> //里面的所有选项全选

No suitable bison/yacc found

### 4.2 iptables 交叉编译

(1)下载地址: <http://www.netfilter.org/projects/iptables/files/iptables-1.4.12.2.tar.bz2>

或: <http://xstrive.tpddns.cn:5000/share.cgi?ssid=a3ce5efcf52c42a78ab24d895079fb2c>

- tar jxf iptables-1.4.12.2.tar.bz2
- cd iptables-1.4.12.2
- ./configure --host=mips CC=mips-linux-uclibc-gnu-gcc --prefix=/home/xstrive/xuwen/project/t40/install\_iptables --enable-static --disable-shared
- Make
- Make install

在把在安装目录生成 iptables 复制到交叉编译平台的/usr/bin

备注: 如果下载的 iptables 版本为 iptables-1.6.0.tar.bz2,

./configure --prefix=/home/xstrive/xuwen/project/t40/ install\_iptables --disable-ipv6 --enable-devel **--disable-nftables** --host=mips CC=mips-linux-uclibc-gnu-gcc

(2) 打开包转发功能:

echo 1 > /proc/sys/net/ipv4/ip\_forward （打开 IP 转发功能）

(3) 启用 iptables 伪装 ip

./iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE (以 nat 方式转发 eth0 网卡，这句执行成功就可以上网了)



## 4.3 注意事项

(1) 报错 1:

```
*** Error: No suitable bison/yacc found. ***
Please install the 'bison' package.
```

解决办法: `sudo apt-get install bison`

(2) 报错 1:

```
*** Error: No suitable flex/lex found. ***
Please install the 'flex' package.
```

解决办法: `sudo apt-get install flex`

(3) 报错 3:

```
CC      libxt_pkttype.o
In file included from libxt_pkttype.c:10:0:
/opt/mips-gcc720-glibc226/mips-linux-gnu/libc/uclibc/usr/include/linux/if_packet.h:83:2:
error: unknown type name '__aligned_u64'
    __aligned_u64 tp_all;
    ^
/opt/mips-gcc720-glibc226/mips-linux-gnu/libc/uclibc/usr/include/linux/if_packet.h:84:2:
error: unknown type name '__aligned_u64'
    __aligned_u64 tp_huge;
    ^
/opt/mips-gcc720-glibc226/mips-linux-gnu/libc/uclibc/usr/include/linux/if_packet.h:85:2:
error: unknown type name '__aligned_u64'
    __aligned_u64 tp_failed;
    ^
/opt/mips-gcc720-glibc226/mips-linux-gnu/libc/uclibc/usr/include/linux/if_packet.h:205:2:
error: unknown type name '__aligned_u64'
    __aligned_u64 seq_num;
    ^
make[2]: *** [libxt_pkttype.o] Error 1
make[2]: Leaving directory '/home/xstrive/xuwen/project/t40/iptables/iptables-1.4.12.2/extensions'
make[1]: *** [all-recursive] Error 1
make[1]: Leaving directory '/home/xstrive/xuwen/project/t40/iptables/iptables-1.4.12.2'
make: *** [all] Error 2
xstrive@ubuntu:~/xuwen/project/t40/iptables/iptables-1.4.12.2$
```

执行:

```
sed -i '/if_packet/i#define __aligned_u64 __u64
__attribute__((aligned(8)))' extensions/libxt_pkttype.c
```