



A31(s) android4.4.2 wifi 模块 移植说明文档

V1.4

2014-04-24

Revision History

Version	Date	Changes compared to previous issue
V1.0	2013-12-20	a31(s) android4.4 wifi porting initial version
V1.1	2014-01-14	add solutions to common problems
V1.2	2014-02-21	add support for ap6476 module
V1.3	2014-03-12	add support for rtl8723bs module
V1.4	2014-04-24	change for a31(s) box android4.4.2

CONFIDENTIAL



目录

1. 前言.....	5
1.1. 编写目的.....	5
1.2. 适用范围.....	5
1.3. 相关人员.....	5
2. 模块介绍.....	6
3. ap6181.....	7
3.1. 内核配置.....	7
3.2. 引脚配置.....	7
3.3. 平台配置.....	9
3.3.1. BoardConfig.mk.....	9
3.3.2. init.sun6i.rc.....	9
3.3.3. fiber-Fir.mk.....	11
3.3.4. config.xml.....	11
3.4. ap6181 模组移植相关文件.....	12
4. ap6210.....	13
4.1. 内核配置.....	13
4.1.1. wifi 部分.....	13
4.1.2. 蓝牙部分.....	13
4.2. 引脚配置.....	14
4.2.1. wifi 部分.....	14
4.2.2. 蓝牙部分.....	15
4.3. 平台配置.....	16
4.3.1. BoardConfig.mk.....	16
4.3.2. init.sun6i.rc.....	17
4.3.3. fiber-k1.mk.....	19
4.3.4. config.xml.....	19
4.4. ap6210 模组移植相关文件.....	20
5. rtl8188eu.....	21
5.1. 内核配置.....	21
5.2. 引脚配置.....	21
5.3. 平台配置.....	22
5.3.1. BoardConfig.mk.....	22
5.3.2. init.sun6i.rc.....	23
5.3.3. fiber-a31st.mk.....	24
5.3.4. config.xml.....	24
5.4. rtl8188eu 模组移植相关文件.....	25
6. ap6476.....	26
6.1. 内核配置.....	26
6.1.1. wifi 部分.....	26
6.1.2. 蓝牙部分.....	26



6.2. 引脚配置.....	27
6.2.1. wifi 部分.....	27
6.2.2. 蓝牙部分.....	29
6.3. 平台配置.....	29
6.3.1. BoardConfig.mk.....	29
6.3.2. init.sun6i.rc.....	30
6.3.3. fiber-a31st.mk.....	32
6.3.4. config.xml.....	33
6.3.5. 波特率修改方法.....	34
6.3.6. 蓝牙 PCM 支持.....	35
6.4. 代码修改.....	36
6.5. ap6476 模组移植相关文件.....	37
7. rtl8723bs.....	38
7.1. 内核配置.....	38
7.1.1. wifi 部分.....	38
7.1.2. 蓝牙部分.....	38
7.2. 引脚配置.....	38
7.2.1. wifi 部分.....	38
7.2.2. 蓝牙部分.....	40
7.3. 平台配置.....	41
7.3.1. BoardConfig.mk.....	41
7.3.2. init.sun6i.rc.....	42
7.3.3. fiber-evb.mk.....	43
7.3.4. config.xml.....	44
7.4. rtl8723bs 模组移植相关文件.....	45
8. F&Q.....	46
8.1. 如何编译.....	46
8.2. 编译报错.....	46
8.3. 修改不起效.....	47
8.4. 常见问题解决.....	47
8.5. 蓝牙注意事项.....	48
9. Declaration.....	50

1. 前言

1.1. 编写目的

本文档编写目的是指导相关人员进行 android4.4.2 平台上 wifi 模块移植，只需按照文档中的步骤修改配置，即可以将 SDK 所支持的 wifi 模组使用起来。

1.2. 适用范围

本文档适用于 A31(s) android4.4.2 平台移植 wifi 模块使用。

1.3. 相关人员

本文档适合需要使用或移植 wifi 模块的人员查阅。

CONFIDENTIAL

2. 模块介绍

目前 a31(s) android4.4.2 平台上已支持 wifi 模组有 5 款，本文档将以每款 wifi 模组的参考方案为例进行说明。

每款 wifi 模组均带有 wifi 功能，部分模组带蓝牙、GPS 和 FM 功能；wifi 的全功能包括 station、softap 和 wifi direct，此三种功能每款 wifi 模组均支持；wifi 模组中 wifi 数据传输接口分 USB 接口和 SDIO 接口两种类型，蓝牙数据传输接口分 USB 接口和 UART 接口两种类型，GPS 数据传输接口一般为 UART 接口。

表 1 android4.4.2 平台 wifi 模组列表

模组	功能（协议）	接口	wifi 功能			参考方案
			Station	Softap	wifi direct	
ap6181	802.11 b/g/n	SDIO	✓	✓	✓	
ap6210	802.11 b/g/n Bluetooth 4.0	SDIO/PCM/UART	✓	✓	✓	
rtl8188eu	802.11 b/g/n	USB	✓	✓	✓	
ap6476	802.11 b/g/n Bluetooth 4.0 GPS、FM	SDIO/PCM/UART	✓	✓	✓	
rtl8723bs	802.11 b/g/n Bluetooth 4.0	SDIO/PCM/UART	✓	✓	✓	

内核默认会编译所有的 wifi 驱动，在确定使用哪一款 wifi 的情况下可把其余的去除编译，这样可减少最终生成的固件大小，减少烧写固件的时间。配置 wifi 驱动编译选项方法如下：

```

去到 linux-3.3 目录下，输入 make ARCH=arm menuconfig
然后选择 Device Drivers --->
    [*] Network device support --->
        [*] Wireless LAN --->
            <M> Broadcom 4329/30 wireless cards support
            ... ..
            <M> Realtek 8192C USB WiFi for SW
            <M> Realtek 8188E USB WiFi
            <M> Realtek 8189E SDIO WiFi
            <M> Realtek 8723A SDIO WiFi
            <M> Realtek 8723A USB WiFi
    
```

内核默认会编译蓝牙，如果实际平台没有蓝牙功能可将蓝牙去除编译，减少生成的内核大小，修改方法如下：

```
去到 linux-3.3 目录下，输入 make ARCH=arm menuconfig
然后选择[*] Networking support --->
      <> Bluetooth subsystem support --->
      .....
      <> RF switch subsystem support --->
```

CONFIDENTIAL

3. ap6181

模块组成: wifi

wifi 功能: wifi station、softap、wifi direct

接口类型: SDIO

3.1. 内核配置

ap6181 模组所使用的 wifi 驱动为 bcmhdh, 代码路径为 \linux-3.3\drivers\net\wireless\bcmhdh\, 编译内核前确认已将其选择上, 并编译成模块, 查看方法如下:

去到 linux-3.3 目录下, 输入 make ARCH=arm menuconfig

然后选择 Device Drivers --->

Network device support --->

Wireless LAN --->

Broadcom 4329/30 wireless cards support

ap6181 模组支持 wifi 唤醒主控功能, 该功能通过 GPIO 中断实现, 中断触发方式需根据硬件原理图做相应配置, 若 WL-WAKE-HOST 脚与主控相连之间接有三极管, 则采用低电平触发, 配置如下:

去到 linux-3.3 目录下, 输入 make ARCH=arm menuconfig

然后选择 Device Drivers --->

Network device support --->

Wireless LAN --->

Low level trigger for OOB interrupt

若 WL-WAKE-HOST 脚与主控相连之间没有接三极管, 则采用高电平触发, 配置如下:

去到 linux-3.3 目录下, 输入 make ARCH=arm menuconfig

然后选择 Device Drivers --->

Network device support --->

Wireless LAN --->

Low level trigger for OOB interrupt

3.2. 引脚配置

ap6181 引脚在 \tools\pack_brandy\chips\sun6i\configs\android\xx-platform-xx\sys_config.fex 中定义, 需根据硬件原理图做相应配置, 需要配置的选项有:

```
;------  
;wifi configuration  
;wifi_sdc_id --- 0- SDC0, 1- SDC1, 2- SDC2, 3- SDC3  
;wifi_usbc_id --- 0- USB0, 1- USB1, 2- USB2  
;wifi_usbc_type -- 1- EHCI(speed 2.0), 2- OHCI(speed 1.0)
```




```
;wifi_mod_sel --- 0- none, 1- ap6181, 2- ap6210(wifi+bt),  
;  
; 3 - rtl8723as(wifi+bt), 4- rtl8189es(SM89E00),  
;  
; 5 - rtl8192cu, 6 - rtl8188eu, 7 - mt6620,  
;  
; 8 - rtl8723au, 9 - ap6330(wifi+bt+fm)  
;-----  
[wifi_para]  
wifi_used = 1  
wifi_sdc_id = 1  
wifi_usbc_id = 1  
wifi_usbc_type = 1  
wifi_mod_sel = 1  
wifi_power = "axp22_dldo1"  
  
; 1 - ap6181 sdio wifi gpio config  
ap6xxx_wl_regon = port:PL05<1><default><default><0>  
ap6xxx_wl_host_wake = port:PL06<0><default><default><0>  
ap6xxx_gpio_power = "axp22_dldo2"  
ap6xxx_clk_power = "axp22_dldo4"
```

说明:

- 1、“;”符号起注释作用;
- 2、“wifi_used”宏赋值为1表示使用wifi,为0表示不使用;
- 3、“wifi_sdc_id”宏表示SDIO接口wifi使用哪个MMC控制器;
- 4、“wifi_mod_sel”宏表示选用哪一款wifi模组;
- 5、“wifi_power”宏表示给模组供电的是axp的axp22_ald01,根据原理图配置;
- 6、“ap6xxx_wl_regon”宏表示wifi使能脚,根据原理图配置;
- 7.“ap6xxx_wl_host_wake”宏表示wifi唤醒主控引脚,根据原理图配置;
- 8.“ap6xxx_gpio_power”宏表示给PG该组gpio供电(主要给SDIO接口使用)的是axp的axp22_dldo2,根据原理图配置,如果不需要则注释掉该行;
- 9.“ap6xxx_clk_power”宏表示模组32k时钟源来自主控时,为时钟输出供电的是axp的axp22_dldo4,根据原理图配置,如果不需要则注释掉该行;

```
[wakeup_src_para]  
cpu_en = 0  
cpu_freq = 48  
;(cpu:apb:ahb)  
pll_ratio = 0x111  
dram_selfresh_en = 1  
dram_freq = 36  
;wakeup_src0 = port:PL5<2><1><default><default>  
wakeup_wifi = port:PL06<2><default><default><0>
```

说明:

- 1、“wakeup_wifi”子键是为wifi唤醒主体眠主控而添加;

2、“wakeup_wifi”所指定的引脚必须与[wifi_para]主键下 ap6181 中的“ap6xxx_wl_host_wake”指定的引脚保持一致；

3.3. 平台配置

3.3.1. BoardConfig.mk

BoardConfig.mk 文件决定 android 加载哪一款 wifi 模组，要配置成使用 ap6181 模组需要把 BoardConfig.mk 文件的相关代码修改成如下。

```
# broadcom wifi support
BOARD_WIFI_VENDOR := broadcom
ifeq ($(BOARD_WIFI_VENDOR), broadcom)
    BOARD_WPA_SUPPLICANT_DRIVER := NL80211
    WPA_SUPPLICANT_VERSION      := VER_0_8_X
    BOARD_WPA_SUPPLICANT_PRIVATE_LIB := lib_driver_cmd_bcmdhd
    BOARD_HOSTAPD_DRIVER        := NL80211
    BOARD_HOSTAPD_PRIVATE_LIB   := lib_driver_cmd_bcmdhd
    BOARD_WLAN_DEVICE           := bcmdhd
    WIFI_DRIVER_FW_PATH_PARAM   := "/sys/module/bcmdhd/parameters/firmware_path"

    SW_BOARD_USR_WIFI := AP6181

    include hardware/broadcom/wlan/bcmdhd/firmware/firmware-bcm.mk
endif
```

说明：

- 1、“#”符号起注释作用；
- 2、“BOARD_WIFI_VENDOR := broadcom”指明使用 broadcom 系列 wifi 模组；
- 3、“SW_BOARD_USR_WIFI := AP6181”指明 wifi 使用 AP6181（注意大写）；

3.3.2. init.sun6i.rc

init.sun6i.rc 是资源和服务配置相关的文件，使用 ap6181 wifi 模组需要作如下修改：

```
# broadcom wifi service
# 1 broadcom wifi sta service
service wpa_supplicant /system/bin/wpa_supplicant \
    -iwlan0 -Dnl80211 -c/data/misc/wifi/wpa_supplicant.conf \
    -I/system/etc/wifi/wpa_supplicant_overlay.conf \
    -O/data/misc/wifi/sockets \
    -e/data/misc/wifi/entropy.bin -g@android:wpa_wlan0
# we will start as root and wpa_supplicant will switch to user wifi
# after setting up the capabilities required for WEXT
# user wifi
```



```
# group wifi inet keystore
class main
socket wpa_wlan0 dgram 660 wifi wifi
disabled
oneshot

# 2 broadcom wifi sta p2p concurrent service
service p2p_suppllicant /system/bin/wpa_suppllicant \
-iwlan0 -Dnl80211 -c/data/misc/wifi/wpa_suppllicant.conf \
-I/system/etc/wifi/wpa_suppllicant_overlay.conf \
-O/data/misc/wifi/sockets -N \
-ip2p0 -Dnl80211 -c/data/misc/wifi/p2p_suppllicant.conf \
-I/system/etc/wifi/p2p_suppllicant_overlay.conf \
-puse_p2p_group_interface=1 -e/data/misc/wifi/entropy.bin \
-g@android:wpa_wlan0
# we will start as root and wpa_suppllicant will switch to user wifi
# after setting up the capabilities required for WEXT
# user wifi
# group wifi inet keystore
class main
socket wpa_wlan0 dgram 660 wifi wifi
disabled
oneshot
```

说明:

1、init.sun6i.rc 文件指定开启 wpa_suppllicant 和 p2p_suppllicant 两个服务，并设置 suppllicant 的启动参数；

```
#insmod boardcom wifi driver
insmod /system/vendor/modules/bcmdhd.ko
```

说明:

1、加载 broadcom wifi 驱动 bcmdhd.ko；

3.3.3. fiber-Fir.mk

fiber-Fir.mk 文件会在编译时拷贝指定文件到特定目录下，使用 ap6181 需要做如下修改：

```
# wifi features
PRODUCT_COPY_FILES += \
    frameworks/native/data/etc/android.hardware.wifi.xml:system/etc/permissions/android.hardware.wifi.xml \
    frameworks/native/data/etc/android.hardware.wifi.direct.xml:system/etc/permissions/android.hardware.wifi.direct.xml \
```

说明:

- 1、将配置文件 android.hardware.wifi.xml 拷贝到指定目录, 以使设置界面显示 wifi;
- 2、将配置文件 android.hardware.wifi.direct.xml 拷贝到指定目录, 以使设置界面显示 wifi direct;

3.3.4. config.xml

ap6181 支持 softap 功能, 在界面上显示 softap, 需在 android\device\softwinner\xx-platform-xx\overlay\frameworks\base\core\res\res\values\config.xml 做如下配置:

```
<!-- List of regexps describing the interface (if any) that represent tetherable
      Wifi interfaces.  If the device doesn't want to support tethering over Wifi this
      should be empty.  An example would be "softap.*" -->
<string-array translatable="false" name="config_tether_wifi_regexs">
<item>"wlan0"</item>
</string-array>
```

ap6181 支持 miracast 功能, 在界面上显示 miracast, 需在 android\device\softwinner\xx-platform-xx\overlay\frameworks\base\core\res\res\values\config.xml 做如下配置:

```
<!-- Whether WiFi display is supported by this device.
      There are many prerequisites for this feature to work correctly.
      Here are a few of them:
      * The WiFi radio must support WiFi P2P.
      * The WiFi radio must support concurrent connections to the WiFi display and
        to an access point.
      * The Audio Flinger audio_policy.conf file must specify a rule for the "r_submix"
        remote submix module.  This module is used to record and stream system
        audio output to the WiFi display encoder in the media server.
      * The remote submix module "audio.r_submix.default" must be installed on the device.
      * The device must be provisioned with HDCP keys (for protected content).
-->
<bool name="config_enableWifiDisplay">true</bool>
```

3.4. ap6181 模组移植相关文件

以下文件是与 ap6181 模组移植相关的, 无需再对这些文件作修改, 只需了解即可。

一、ap6181 驱动代码

linux-3.3\drivers\net\wireless\bcmdhd

二、电源及 GPIO 控制

lichee\linux-3.3\arch\arm\mach-sun6i\rf\wifi_pm.c

lichee\linux-3.3\arch\arm\mach-sun6i\rf\wifi_pm_ap6xxx.c

wifi_pm_ap6xxx.c 是 ap6181 模组电源和 GPIO 控制的实现文件, 需要把 ap6181 GPIO 控制实现函数

接口添加到 wifi_pm.c。

三、wifi firmware 和 nvram

\android\hardware\broadcom\wlan\bcmdhd\firmware\ap6181\

\android\hardware\broadcom\wlan\bcmdhd\firmware\firmware-bcm.mk

firmware-bcm.mk 通过 Boardconfig.mk 中指定的 SW_BOARD_USR_WIFI := AP6181，将 ap6181 目录下的 firmware 和 nvram 拷贝到指定目录下。

CONFIDENTIAL

4. ap6210

模块组成: wifi、蓝牙

wifi 功能: wifi station、softap、wifi direct

接口类型: SDIO、UART

4.1. 内核配置

4.1.1. wifi 部分

ap6210 模组所使用的 wifi 驱动为 bcmhdh, 代码路径为 \linux-3.3\drivers\net\wireless\bcmhdh\, 编译内核前确认已将其选择上, 并编译成模块, 查看方法如下:

```
去到 linux-3.3 目录下, 输入 make ARCH=arm menuconfig
然后选择 Device Drivers --->
    Network device support --->
        Wireless LAN --->
            <M> Broadcom 4329/30 wireless cards support
```

ap6210 模组支持 wifi 唤醒主控功能, 该功能通过 GPIO 中断实现, 中断触发方式需根据硬件原理图做相应配置, 若 WL-WAKE-HOST 脚与主控相连之间接有三极管, 则采用低电平触发, 配置如下:

```
去到 linux-3.3 目录下, 输入 make ARCH=arm menuconfig
然后选择 Device Drivers --->
    Network device support --->
        Wireless LAN --->
            [*] Low level trigger for OOB interrupt
```

若 WL-WAKE-HOST 脚与主控相连之间没有接三极管, 则采用高电平触发, 配置如下:

```
去到 linux-3.3 目录下, 输入 make ARCH=arm menuconfig
然后选择 Device Drivers --->
    Network device support --->
        Wireless LAN --->
            [ ] Low level trigger for OOB interrupt
```

4.1.2. 蓝牙部分

ap6210 模组支持蓝牙唤醒主控功能, 内核默认没有开启此功能, 开启该功能需配置如下:

```
去到 linux-3.3 目录下, 输入 make ARCH=arm menuconfig
然后选择[*] Networking support --->
    <*> Bluetooth subsystem support --->
        Bluetooth device drivers --->
```

<*> Bluetooth Low Power Manager Support

与 wifi 唤醒主控一样，蓝牙唤醒主控功能也是通过 GPIO 中断实现中断触发方式需根据硬件原理图做相应配置，若 BT-WAKE-HOST 脚与主控相连之间接有三极管，则采用低电平触发，配置如下：

```
去到 linux-3.3 目录下，输入 make ARCH=arm menuconfig
然后选择[*] Networking support --->
    <*> Bluetooth subsystem support --->
        Bluetooth device drivers --->
            <*> An inverter between bt hostwake pin and cpu
```

若 BT-WAKE-HOST 脚与主控相连之间没有接三极管，则采用高电平触发，配置如下：

```
去到 linux-3.3 目录下，输入 make ARCH=arm menuconfig
然后选择[*] Networking support --->
    <*> Bluetooth subsystem support --->
        Bluetooth device drivers --->
            <> An inverter between bt hostwake pin and cpu
```

4.2. 引脚配置

ap6210 引脚在\tools\pack_brandy\chips\sun6i\configs\android\xx-platform-xx\sys_config.fex 中定义，需根据硬件原理图做相应配置，需要配置的选项有：

4.2.1. wifi 部分

```
-----
;wifi configuration
;wifi_sdc_id --- 0- SDC0, 1- SDC1, 2- SDC2, 3- SDC3
;wifi_usbc_id --- 0- USB0, 1- USB1, 2- USB2
;wifi_usbc_type -- 1- EHCI(speed 2.0), 2- OHCI(speed 1.0)
;wifi_mod_sel --- 0- none, 1- ap6181, 2- ap6210(wifi+bt),
; 3 - rtl8723as(wifi+bt), 4- rtl8189es(SM89E00),
; 5 - rtl8192cu, 6 - rtl8188eu, 7 - mt6620
; 8 - rtl8723au, 9 - ap6330(wifi+bt+fm)
;-----
[wifi_para]
wifi_used = 1
wifi_sdc_id = 1
wifi_usbc_id = 1
wifi_usbc_type = 1
wifi_mod_sel = 2
wifi_power = "xpc22_dldo1"
```

```

; 2 - ap6210 sdio wifi gpio config
ap6xxx_wl_regon      = port:PL05<1><default><default><0>
ap6xxx_wl_host_wake = port:PL06<0><default><default><0>
ap6xxx_bt_regon     = port:PL08<1><default><default><0>
ap6xxx_bt_wake      = port:PG17<1><default><default><0>
ap6xxx_bt_host_wake = port:PL07<0><default><default><0>
ap6xxx_gpio_power   = "axp22_dldo2"
ap6xxx_clk_power    = "axp22_dldo4"

```

说明:

- 1、“;”符号起注释作用;
- 2、“wifi_used”宏赋值为1表示使用wifi,为0表示不使用;
- 3、“wifi_sdc_id”宏表示SDIO接口wifi使用哪个MMC控制器;
- 4、“wifi_mod_sel”宏表示选用哪一款wifi模组;
- 5、“wifi_power”宏表示给模组供电的是axp的axp22_aldol,根据原理图配置;
- 6、“ap6xxx_wl_regon”宏表示wifi使能脚,“ap6xxx_bt_regon”表示蓝牙使能脚,根据原理图配置;
- 7、“ap6xxx_wl_host_wake”宏表示wifi唤醒主控引脚,“ap6xxx_bt_host_wake”宏表示蓝牙唤醒主控引脚,根据原理图配置;
- 8、“ap6xxx_bt_wake”宏表示主控唤醒引脚蓝牙,根据原理图配置;
- 9.“ap6xxx_gpio_power”宏表示给PG该组gpio供电(主要给SDIO接口使用)的是axp的axp22_dldo2,根据原理图配置, **如果不需要则注释掉该行;**
- 10.“ap6xxx_clk_power”宏表示模组32k时钟源来自主控时,为时钟输出供电的是axp的axp22_dldo4,根据原理图配置, **如果不需要则注释掉该行;**

```

[wakeup_src_para]
cpu_en      = 0
cpu_freq    = 48
;(cpu:apb:ahb)
pll_ratio   = 0x111
dram_selfresh_en = 1
dram_freq   = 36
;wakeup_src0 = port:PL5<2><1><default><default>
wakeup_wifi = port:PL06<2><default><default><0>
wakeup_src_bt = port:PL07<2><default><default><0>

```

说明:

- 1、“wakeup_wifi”子键是为wifi唤醒主体睡眠主控而添加;
- 2、“wakeup_wifi”所指定的引脚必须与[wifi_para]主键下ap6210中的“ap6xxx_wl_host_wake”指定的引脚保持一致;

4.2.2. 蓝牙部分

```

;-----
;blue tooth

```




```
;bt_used      ---- blue tooth used (0- no used, 1- used)
;bt_uart_id   ---- uart index
;-----
[bt_para]
bt_used      = 1
bt_uart_id   = 2
bt_wakeup    =
bt_gpio      =
bt_rst       =
```

说明:

- 1、“bt_used”宏赋值为1表示使用蓝牙，为0表示不使用；
- 2、“bt_uart_id”宏表示蓝牙所使用的是哪个uart；

```
[wakeup_src_para]
cpu_en       = 0
cpu_freq     = 48
;(cpu:apb:ahb)
pll_ratio    = 0x111
dram_selfresh_en = 1
dram_freq    = 36
;wakeup_src0 = port:PL5<2><1><default><default>
wakeup_wifi  = port:PL06<2><default><default><0>
wakeup_src_bt = port:PL07<2><default><default><0>
```

说明:

- 1、“wakeup_src_bt”子键是为蓝牙唤醒主体睡眠主控而添加；
- 2、“wakeup_src_bt”所指定的引脚必须与[wifi_para]主键下ap6210中的“ap6xxx_bt_host_wake”指定的引脚保持一致；

4.3. 平台配置

4.3.1. BoardConfig.mk

BoardConfig.mk 文件决定 android 加载哪一款 wifi 模组，要配置成使用 ap6210 模组需要把 BoardConfig.mk 文件的相关代码修改成如下：

```
# bt default config
BOARD_BLUETOOTH_BDROID_BUILDCFG_INCLUDE_DIR :=
device/softwinner/fiber-k1/bluetooth

# 1. broadcom wifi support
BOARD_WIFI_VENDOR := broadcom
ifeq ($(BOARD_WIFI_VENDOR), broadcom)
    BOARD_WPA_SUPPLICANT_DRIVER := NL80211
```



```
WPA_SUPPLICANT_VERSION := VER_0_8_X
BOARD_WPA_SUPPLICANT_PRIVATE_LIB := lib_driver_cmd_bcmdhd
BOARD_HOSTAPD_DRIVER := NL80211
BOARD_HOSTAPD_PRIVATE_LIB := lib_driver_cmd_bcmdhd
BOARD_WLAN_DEVICE := bcmdhd
WIFI_DRIVER_FW_PATH_PARAM := "/sys/module/bcmdhd/parameters/firmware_path"

SW_BOARD_USR_WIFI := AP6210

include hardware/broadcom/wlan/bcmdhd/firmware/firmware-bcm.mk

endif

# 2. Bluetooth Configuration
# make sure BOARD_HAVE_BLUETOOTH is true for every bt vendor
BOARD_HAVE_BLUETOOTH := true
BOARD_HAVE_BLUETOOTH_BCM := true
SW_BOARD_HAVE_BLUETOOTH_NAME := ap6210
```

说明:

- 1、“#”符号起注释作用;
- 2、“BOARD_WIFI_VENDOR := broadcom”指明使用 broadcom 系列 wifi 模组;
- 3、“SW_BOARD_USR_WIFI := AP6210”指明 wifi 使用 AP6210 (注意大写);
- 4、“device/softwinner/fiber-k1/bluetooth”用于设置蓝牙设备名;
- 5、“BOARD_HAVE_BLUETOOTH := true”指明拥有蓝牙功能;
- 6、“BOARD_HAVE_BLUETOOTH_BCM := true”指明使用 broadcom 系列蓝牙;
- 7、“SW_BOARD_HAVE_BLUETOOTH_NAME := ap6210”指明使用 ap6210 (注意小写) 蓝牙;

4.3.2. init.sun6i.rc

init.sun6i.rc 是资源和服务配置相关的文件, 使用 ap6181 wifi 模组需要作如下修改:

```
# broadcom wifi service
# 1 broadcom wifi sta service
service wpa_supplicant /system/bin/wpa_supplicant \
    -iwlan0 -Dnl80211 -c/data/misc/wifi/wpa_supplicant.conf \
    -I/system/etc/wifi/wpa_supplicant_overlay.conf \
    -O/data/misc/wifi/sockets \
    -e/data/misc/wifi/entropy.bin -g@android:wpa_wlan0
# we will start as root and wpa_supplicant will switch to user wifi
# after setting up the capabilities required for WEXT
# user wifi
# group wifi inet keystore
class main
socket wpa_wlan0 dgram 660 wifi wifi
disabled
```



```
oneshot

# 2 broadcom wifi sta p2p concurrent service
service p2p_suppllicant /system/bin/wpa_suppllicant \
    -iwlan0 -Dnl80211 -c/data/misc/wifi/wpa_suppllicant.conf \
    -I/system/etc/wifi/wpa_suppllicant_overlay.conf \
    -O/data/misc/wifi/sockets -N \
    -ip2p0 -Dnl80211 -c/data/misc/wifi/p2p_suppllicant.conf \
    -I/system/etc/wifi/p2p_suppllicant_overlay.conf \
    -puse_p2p_group_interface=1 -e/data/misc/wifi/entropy.bin \
    -g@android:wpa_wlan0
# we will start as root and wpa_suppllicant will switch to user wifi
# after setting up the capabilities required for WEXT
# user wifi
# group wifi inet keystore
class main
socket wpa_wlan0 dgram 660 wifi wifi
disabled
oneshot
```

说明:

- 1、init.sun6i.rc 文件指定开启 wpa_suppllicant 和 p2p_suppllicant 两个服务, 并设置 suppllicant 的启动参数;

```
#insmod boardcom wifi driver
insmod /system/vendor/modules/bcmdhd.ko
```

说明:

- 1、加载 broadcom wifi 驱动 bcmdhd.ko;

```
# bluetooth
# UART device
    chmod 0660 /dev/ttyS2
    chown bluetooth net_bt_stack /dev/ttyS2
# power up/down interface
    chmod 0660 /sys/class/rfkill/rfkill0/state
    chmod 0660 /sys/class/rfkill/rfkill0/type
    chown bluetooth net_bt_stack /sys/class/rfkill/rfkill0/state
    chown bluetooth net_bt_stack /sys/class/rfkill/rfkill0/type
# bluetooth LPM
    chmod 0220 /proc/bluetooth/sleep/lpm
    chmod 0220 /proc/bluetooth/sleep/btwrite
    chown bluetooth net_bt_stack /proc/bluetooth/sleep/lpm
    chown bluetooth net_bt_stack /proc/bluetooth/sleep/btwrite
```

说明:

1、设置蓝牙属性，设置蓝牙相关文件节点权限和用户组等操作;

4.3.3. fiber-k1.mk

fiber-k1.mk 文件会在编译时拷贝指定文件到特定目录下，使用 ap6210 需要做如下修改:

```
# wifi & bt config file
PRODUCT_COPY_FILES += \
    frameworks/native/data/etc/android.hardware.wifi.xml:system/etc/permissions/android.hardware.wifi.xml \
    frameworks/native/data/etc/android.hardware.wifi.direct.xml:system/etc/permissions/android.hardware.wifi.direct.xml \
    frameworks/native/data/etc/android.hardware.bluetooth.xml:system/etc/permissions/android.hardware.bluetooth.xml \
    frameworks/native/data/etc/android.hardware.bluetooth_le.xml:system/etc/permissions/android.hardware.bluetooth_le.xml

# build Bluetooth.apk
PRODUCT_PACKAGES += \
    Bluetooth
```

说明:

- 1、将配置文件 android.hardware.wifi.xml 拷贝到指定目录，以使设置界面显示 wifi;
- 2、将配置文件 android.hardware.wifi.direct.xml 拷贝到指定目录，以使设置界面显示 wifi direct;
- 3、将配置文件 android.hardware.bluetooth.xml 拷贝到指定目录，以使设置界面显示蓝牙;
- 4、将配置文件 android.hardware.bluetooth_le.xml 拷贝到指定目录，以支持蓝牙低功耗设备;
- 5、“PRODUCT_PACKAGES += Bluetooth”指定去编译 Bluetooth.apk;

4.3.4. config.xml

ap6210 支持 softap 功能，在界面上显示 softap，需在 android\device\softwinner\xx-platform-xx\overlay\frameworks\base\core\res\res\values\config.xml 做如下配置:

```
<!-- List of regexpressions describing the interface (if any) that represent tetherable
Wifi interfaces. If the device doesn't want to support tethering over Wifi this
should be empty. An example would be "softap.*" -->
<string-array translatable="false" name="config_tether_wifi_regexp">
    <item>"wlan0"</item>
</string-array>
```

ap6210 支持 miracast 功能，在界面上显示 miracast，需在 android\device\softwinner\xx-platform-xx\overlay\frameworks\base\core\res\res\values\config.xml 做如下配置:

```
<!-- Whether WiFi display is supported by this device.
There are many prerequisites for this feature to work correctly.
Here are a few of them:
```

- * The WiFi radio must support WiFi P2P.
- * The WiFi radio must support concurrent connections to the WiFi display and to an access point.
- * The Audio Flinger audio_policy.conf file must specify a rule for the "r_submix" remote submix module. This module is used to record and stream system audio output to the WiFi display encoder in the media server.
- * The remote submix module "audio.r_submix.default" must be installed on the device.
- * The device must be provisioned with HDCP keys (for protected content).

-->

```
<bool name="config_enableWifiDisplay">true</bool>
```

ap6210 支持蓝牙共享网络功能，在界面上显示蓝牙共享网络，需在 android\device\softwinner\xx-platform-xx\overlay\frameworks\base\core\res\res\values\config.xml 做如下配置：

```
<!-- default: disable Bluetooth PAN feature -->  
<string-array translatable="false" name="config_tether_bluetooth_regexp">  
  <item>"bt-pan"</item>  
</string-array>
```

4.4. ap6210 模组移植相关文件

以下文件是与 ap6210 模组移植相关的，无需再对这些文件作修改，只需了解即可。

一、ap6210 驱动代码

\linux-3.3\drivers\net\wireless\bcmdhd

二、电源及 GPIO 控制

\lichee\linux-3.3\arch\arm\mach-sun6i\rf\wifi_pm.c

\lichee\linux-3.3\arch\arm\mach-sun6i\rf\wifi_pm_ap6xxx.c

\lichee\linux-3.3\arch\arm\mach-sun6i\rf\bt_pm.c

wifi_pm_ap6xxx.c 是 ap6210 模组电源和 GPIO 控制的实现文件，需要把 ap6210 GPIO 控制实现函数接口添加到 wifi_pm.c。

bt_pm.c 是蓝牙电源管理的实现文件，需要把 ap6210 bt 的电源控制添加到 rkill_set_power 函数。

三、wifi firmware 和 nvram

\android\hardware\broadcom\wlan\bcmdhd\firmware\ap6210\

\android\hardware\broadcom\wlan\bcmdhd\firmware\firmware-bcm.mk

firmware-bcm.mk 通过 Boardconfig.mk 中指定的 SW_BOARD_USR_WIFI := AP6210，将 ap6210 目录下的 firmware 和 nvram 拷贝到指定目录下。

5. rtl8188eu

模块组成: wifi

wifi 功能: wifi station、softap、wifi direct

接口类型: USB

5.1. 内核配置

rtl8188eu 模组所使用的 wifi 驱动为 8188eu，代码路径为 \linux-3.3\drivers\net\wireless\rtl8188eu\，编译内核前确认已将其选择上，并编译成模块，查看方法如下：

去到 linux-3.3 目录下，输入 make ARCH=arm menuconfig

然后选择 Device Drivers --->

Network device support --->

Wireless LAN --->

<M> Realtek 8188E USB WiFi

5.2. 引脚配置

ap6210 引脚在 \tools\pack_brandy\chips\sun6i\configs\android\xx-platform-xx\sys_config.fex 中定义，**需**要根据硬件原理图做相应配置，需要配置的选项有：

```
-----  
;wifi configuration  
;wifi_sdc_id    --- 0- SDC0, 1- SDC1, 2- SDC2, 3- SDC3  
;wifi_usbc_id  --- 0- USB0, 1- USB1, 2- USB2  
;wifi_usbc_type -- 1- EHCI(speed 2.0), 2- OHCI(speed 1.0)  
;wifi_mod_sel  --- 0- none, 1- ap6181, 2- ap6210(wifi+bt),  
;                3 - rtl8723as(wifi+bt), 4- rtl8189es(SM89E00),  
;                5 - rtl8192eu, 6 - rtl8188eu, 7 - mt6620,  
;                8 - rtl8723au, 9 - ap6330(wifi+bt+fm)  
;-----  
[wifi_para]  
wifi_used      = 1  
wifi_sdc_id    = 1  
wifi_usbc_id   = 1  
wifi_usbc_type = 1  
wifi_mod_sel   = 6  
wifi_power     = "axp22_dldo1"
```

说明：

- 1、“;”符号起注释作用；
- 2、“wifi_used”宏赋值为 1 表示使用 wifi，为 0 表示不使用；
- 3、“wifi_usbc_id”宏表示使用 usb wifi 使用哪个 usb 端口；

- 4、“wifi_usbc_type”宏表示使哪种速度的usb;
- 5、“wifi_mod_sel”宏表示选用哪一款wifi模组;
- 6、“wifi_power”宏表示给模组供电的是axp的axp22_aldol,根据原理图配置;
- 7、“usb_host_init_state”宏设成0是wifi模组的电源由wifi控制,设成1wifi模组的电源将由usb控制;

```
;-
;---      USB1 控制标志
;-
[usbc1]
usb_used          = 1
usb_port_type     = 1
usb_detect_type   = 0
usb_id_gpio       =
usb_det_vbus_gpio =
usb_drv_vbus_gpio =
usb_restrict_gpio =
usb_host_init_state = 0
usb_restric_flag  = 0
```

说明:

- 1、“usb_host_init_state”宏设成0表示usb控制器在系统启动后不上电,wifi模组供电由AXP控制,设成1表示usb控制器系统启动后即上电,wifi模组供电由AXP控制失效。

5.3. 平台配置

5.3.1. BoardConfig.mk

BoardConfig.mk 文件决定 android 加载哪一款 wifi 模组,要配置成使用 rtl8188eu 模组需要把 BoardConfig.mk 文件的相关代码修改成如下:

```
# 1. realtek wifi configuration
BOARD_WIFI_VENDOR := realtek
ifeq ($(BOARD_WIFI_VENDOR), realtek)
    WPA_SUPPLICANT_VERSION := VER_0_8_X
    BOARD_WPA_SUPPLICANT_DRIVER := NL80211
    BOARD_WPA_SUPPLICANT_PRIVATE_LIB := lib_driver_cmd_rtl
    BOARD_HOSTAPD_DRIVER := NL80211
    BOARD_HOSTAPD_PRIVATE_LIB := lib_driver_cmd_rtl

    SW_BOARD_USR_WIFI := rtl8188eu
    BOARD_WLAN_DEVICE := rtl8188eu
endif
```

说明:

- 1、“#”符号起注释作用;
- 2、“BOARD_WIFI_VENDOR := realtek”指明使用 realtek 系列 wifi 模组;
- 3、“SW_BOARD_USR_WIFI := rtl8188eu”、“BOARD_WLAN_DEVICE := rtl8188eu”宏指明使用 rtl8188eu 模组;

5.3.2. init.sun6i.rc

init.sun6i.rc 是资源和服务配置相关的文件，使用 rtl8188eu wifi 模组需要作如下修改:

```
# realtek wifi service
# 1. realtek wifi sta service
service wpa_supplicant /system/bin/wpa_supplicant \
    -iwlan0 -Dnl80211 -c/data/misc/wifi/wpa_supplicant.conf \
    -O/data/misc/wifi/sockets \
    -e/data/misc/wifi/entropy.bin -g@android:wpa_wlan0
# we will start as root and wpa_supplicant will switch to user wifi
# after setting up the capabilities required for WEXT
# user wifi
# group wifi inet keystore
class main
socket wpa_wlan0 dgram 660 wifi wifi
disabled
oneshot

# 1. realtek wifi sta p2p concurrent service
service p2p_supplicant /system/bin/wpa_supplicant \
    -ip2p0 -Dnl80211 -c/data/misc/wifi/p2p_supplicant.conf \
    -e/data/misc/wifi/entropy.bin -N \
    -iwlan0 -Dnl80211 -c/data/misc/wifi/wpa_supplicant.conf \
    -O/data/misc/wifi/sockets \
    -g@android:wpa_wlan0
class main
socket wpa_wlan0 dgram 660 wifi wifi
disabled
oneshot
```

说明:

- 1、init.sun6i.rc 文件指定开启 wpa_supplicant 和 p2p_supplicant 两个服务，并设置 supplicant 的启动参数;

5.3.3. fiber-a31st.mk

fiber-a31st.mk 文件会在编译时拷贝指定文件到特定目录下，使用 rtl8188eu 需要做如下修改。


```
# wifi features
PRODUCT_COPY_FILES += \
    frameworks/native/data/etc/android.hardware.wifi.xml:system/etc/permissions/android.hardware.wifi.xml \
    frameworks/native/data/etc/android.hardware.wifi.direct.xml:system/etc/permissions/android.hardware.wifi.direct.xml \
```

说明:

- 1、将配置文件 android.hardware.wifi.xml 拷贝到指定目录，以使设置界面显示 wifi;
- 2、将配置文件 android.hardware.wifi.direct.xml 拷贝到指定目录，以使设置界面显示 wifi direct;

5.3.4. config.xml

rtl8188eu 支持 softap 功能，在界面上显示 softap，需在 android\device\softwinner\xx-platform-xx\overlay\frameworks\base\core\res\res\values\config.xml 做如下配置:

```
<!-- List of regexpressions describing the interface (if any) that represent tetherable
      Wifi interfaces.  If the device doesn't want to support tethering over Wifi this
      should be empty.  An example would be "softap.*" -->
<string-array translatable="false" name="config_tether_wifi_regexs">
<item>"wlan0"</item>
</string-array>
```

rtl8188eu 支持 miracast 功能，在界面上显示 miracast，需在 android\device\softwinner\xx-platform-xx\overlay\frameworks\base\core\res\res\values\config.xml 做如下配置:

```
<!-- Whether WiFi display is supported by this device.
      There are many prerequisites for this feature to work correctly.
      Here are a few of them:
      * The WiFi radio must support WiFi P2P.
      * The WiFi radio must support concurrent connections to the WiFi display and
        to an access point.
      * The Audio Flinger audio_policy.conf file must specify a rule for the "r_submix"
        remote submix module.  This module is used to record and stream system
        audio output to the WiFi display encoder in the media server.
      * The remote submix module "audio.r_submix.default" must be installed on the device.
      * The device must be provisioned with HDCP keys (for protected content).
-->
<bool name="config_enableWifiDisplay">true</bool>
```

5.4. rtl8188eu 模组移植相关文件

以下文件是与 rtl8188eu 模组移植相关的，无需再对这些文件作修改，只需了解即可。

一、rtl8188eu 驱动代码

\linux-3.3\drivers\net\wireless\rtl8188eu

二、电源控制

\lichee\linux-3.3\arch\arm\mach-sun6i\rf\wifi_pm.c

\lichee\linux-3.3\arch\arm\mach-sun6i\rf\wifi_pm_rtl8188eu.c

wifi_pm_rtl8188eu.c 是 rtl8188eu 模组电源控制的实现文件，需要把 rtl8188eu 控制实现函数接口添加到 wifi_pm.c。

CONFIDENTIAL

6. ap6476

模块组成: wifi、蓝牙、FM、GPS

wifi 功能: wifi station、softap、wifi direct

接口类型: SDIO、UART

6.1. 内核配置

6.1.1. wifi 部分

ap6476 模组所使用的 wifi 驱动为 bcmdhd, 代码路径为 \linux-3.3\drivers\net\wireless\bcmdhd\, 编译内核前确认已将其选择上, 并编译成模块, 查看方法如下:

```
去到 linux-3.3 目录下, 输入 make ARCH=arm menuconfig
然后选择 Device Drivers --->
    Network device support --->
        Wireless LAN --->
            <M> Broadcom 4329/30 wireless cards support
```

ap6476 模组支持 wifi 唤醒主控功能, 该功能通过 GPIO 中断实现, 中断触发方式需根据硬件原理图做相应配置, 若 WL-WAKE-HOST 脚与主控相连之间接有三极管, 则采用低电平触发, 配置如下:

```
去到 linux-3.3 目录下, 输入 make ARCH=arm menuconfig
然后选择 Device Drivers --->
    Network device support --->
        Wireless LAN --->
            [*] Low level trigger for OOB interrupt
```

若 WL-WAKE-HOST 脚与主控相连之间没有接三极管, 则采用高电平触发, 配置如下:

```
去到 linux-3.3 目录下, 输入 make ARCH=arm menuconfig
然后选择 Device Drivers --->
    Network device support --->
        Wireless LAN --->
            [ ] Low level trigger for OOB interrupt
```

6.1.2. 蓝牙部分

ap6476 模组支持蓝牙唤醒主控功能, 内核默认没有开启此功能, 开启该功能需配置如下:

```
去到 linux-3.3 目录下, 输入 make ARCH=arm menuconfig
然后选择[*] Networking support --->
    <*> Bluetooth subsystem support --->
        Bluetooth device drivers --->
```

<*> Bluetooth Low Power Manager Support

与 wifi 唤醒主控一样，蓝牙唤醒主控功能也是通过 GPIO 中断实现中断触发方式需根据硬件原理图做相应配置，若 BT-WAKE-HOST 脚与主控相连之间接有三极管，则采用低电平触发，配置如下：

```
去到 linux-3.3 目录下，输入 make ARCH=arm menuconfig
然后选择[*] Networking support --->
    <*> Bluetooth subsystem support --->
        Bluetooth device drivers --->
            <*> An inverter between bt hostwake pin and cpu
```

若 BT-WAKE-HOST 脚与主控相连之间没有接三极管，则采用高电平触发，配置如下：

```
去到 linux-3.3 目录下，输入 make ARCH=arm menuconfig
然后选择[*] Networking support --->
    <*> Bluetooth subsystem support --->
        Bluetooth device drivers --->
            <> An inverter between bt hostwake pin and cpu
```

6.2. 引脚配置

ap6476 引脚在\tools\pack_brandy\chips\sun6i\configs\android\xx-platform-xx\sys_config.fex 中定义，需根据硬件原理图做相应配置，需要配置的选项有：

6.2.1. wifi 部分

```
-----
;wifi configuration
;wifi_sdc_id --- 0- SDC0, 1- SDC1, 2- SDC2, 3- SDC3
;wifi_usbc_id --- 0- USB0, 1- USB1, 2- USB2
;wifi_usbc_type -- 1- EHCI(speed 2.0), 2- OHCI(speed 1.0)
;wifi_mod_sel --- 0- none, 1- ap6181, 2- ap6210(wifi+bt),
; 3 - rtl8723as(wifi+bt), 4- rtl8189es(SM89E00),
; 5 - rtl8192cu, 6 - rtl8188eu, 7 - mt6620
; 8 - rtl8723au, 9 - ap6330(wifi+bt+fm)
; 10 - ap6476(wifi+bt+fm+gps)
;
-----
[wifi_para]
wifi_used = 1
wifi_sdc_id = 1
wifi_usbc_id = 1
wifi_usbc_type = 1
wifi_mod_sel = 10
wifi_power = "axp22_dldo2"
```



```
; 10 - ap6476 sdio wifi + bt gpio config
ap6xxx_wl_regon      = port:PL05<1><default><default><0>
ap6xxx_wl_host_wake = port:PL06<0><default><default><0>
ap6xxx_bt_regon     = port:PL08<1><default><default><0>
ap6xxx_bt_wake      = port:PG17<1><default><default><0>
ap6xxx_bt_host_wake = port:PL07<0><default><default><0>
ap6xxx_gpio_power   = "axp22_dldo2"
ap6xxx_clk_power    = "axp22_dldo4"
```

说明:

- 1、“;”符号起注释作用;
- 2、“wifi_used”宏赋值为 1 表示使用 wifi, 为 0 表示不使用;
- 3、“wifi_sdc_id”宏表示 SDIO 接口 wifi 使用哪个 MMC 控制器;
- 4、“wifi_mod_sel”宏表示选用哪一款 wifi 模组;
- 5、“wifi_power”宏表示给模组供电的是 axp 的 axp22_aldo2, 根据原理图配置;
- 6、“ap6xxx_wl_regon”宏表示 wifi 使能脚, “ap6xxx_bt_regon”表示蓝牙使能脚, 根据原理图配置;
- 7.“ap6xxx_wl_host_wake”宏表示 wifi 唤醒主控引脚, “ap6xxx_bt_host_wake”宏表示蓝牙唤醒主控引脚, 根据原理图配置;
- 8.“ap6xxx_bt_wake”宏表示主控唤醒引脚蓝牙, 根据原理图配置;
- 9.“ap6xxx_gpio_power”宏表示给 PG 该组 gpio 供电(主要给 SDIO 接口使用)的是 axp 的 axp22_dldo2, 根据原理图配置, 如果不需要则注释掉该行; 如果 “wifi_power”引脚也是使用 axp22_dldo2, 则注释掉该行;
- 10.“ap6xxx_clk_power”宏表示模组 32k 时钟源来自主控时, 为时钟输出供电的是 axp 的 axp22_dldo4, 根据原理图配置, 如果不需要则注释掉该行; 如果 “wifi_power”引脚也是使用 axp22_dldo4, 则注释掉该行;

```
[wakeup_src_para]
cpu_en      = 0
cpu_freq    = 48
;(cpu:apb:ahb)
pll_ratio   = 0x111
dram_selfresh_en = 1
dram_freq   = 36
;wakeup_src0 = port:PL5<2><1><default><default>
wakeup_wifi = port:PL06<2><default><default><0>
wakeup_src_bt = port:PL07<2><default><default><0>
```

说明:

- 1、“wakeup_wifi”子键是为 wifi 唤醒主休眠主控而添加;
- 2、“wakeup_wifi”所指定的引脚必须与[wifi_para]主键下 ap6476 中的 “ap6xxx_wl_host_wake”指定的引脚保持一致;

6.2.2. 蓝牙部分

```

;-----
;blue tooth
;bt_used      ---- blue tooth used (0- no used, 1- used)
;bt_uart_id   ---- uart index
;-----

[bt_para]
bt_used      = 1
bt_uart_id   = 2
bt_wakeup    =
bt_gpio      =
bt_rst       =

```

说明:

- 1、“bt_used”宏赋值为1表示使用蓝牙，为0表示不使用；
- 2、“bt_uart_id”宏表示蓝牙所使用的是哪个uart；

```

[wakeup_src_para]
cpu_en       = 0
cpu_freq     = 48
;(cpu:apb:ahb)
pll_ratio    = 0x111
dram_selfresh_en = 1
dram_freq    = 36
;wakeup_src0 = port:PL5<2><1><default><default>
wakeup_wifi  = port:PL06<2><default><default><0>
wakeup_src_bt = port:PL07<2><default><default><0>

```

说明:

- 1、“wakeup_src_bt”子键是为蓝牙唤醒主休眠主控而添加；
- 2、“wakeup_src_bt”所指定的引脚必须与[wifi_para]主键下 ap6476 中的“ap6xxx_bt_host_wake”指定的引脚保持一致；

6.3. 平台配置

6.3.1. BoardConfig.mk

BoardConfig.mk 文件决定 android 加载哪一款 wifi 模组，要配置成使用 ap6476 模组需要把 BoardConfig.mk 文件的相关代码修改成如下：

```

# bt default config
BOARD_BLUETOOTH_BDROID_BUILDCFG_INCLUDE_DIR :=
device/softwinner/fiber-a31st/bluetooth

# 1. broadcom wifi support

```

```
BOARD_WIFI_VENDOR := broadcom
ifeq ($(BOARD_WIFI_VENDOR), broadcom)
    BOARD_WPA_SUPPLICANT_DRIVER := NL80211
    WPA_SUPPLICANT_VERSION := VER_0_8_X
    BOARD_WPA_SUPPLICANT_PRIVATE_LIB := lib_driver_cmd_bcmdhd
    BOARD_HOSTAPD_DRIVER := NL80211
    BOARD_HOSTAPD_PRIVATE_LIB := lib_driver_cmd_bcmdhd
    BOARD_WLAN_DEVICE := bcmdhd
    WIFI_DRIVER_FW_PATH_PARAM := "/sys/module/bcmdhd/parameters/firmware_path"

    SW_BOARD_USR_WIFI := AP6476
    include hardware/broadcom/wlan/bcmdhd/firmware/firmware-bcm.mk
endif

# 2. Bluetooth Configuration
# make sure BOARD_HAVE_BLUETOOTH is true for every bt vendor
BOARD_HAVE_BLUETOOTH := true
BOARD_HAVE_BLUETOOTH_BCM := true
SW_BOARD_HAVE_BLUETOOTH_NAME := ap6476

#Don't built FM application by default
BLUETOOTH_ENABLE_FM := true
```

说明:

- 1、“#”符号起注释作用;
- 2、“BOARD_WIFI_VENDOR := broadcom”指明使用 broadcom 系列 wifi 模组;
- 3、“SW_BOARD_USR_WIFI := AP6476”指明 wifi 使用 AP6476 (注意大写);
- 4、“device/softwinner/fiber-k1/bluetooth”用于设置蓝牙设备名;
- 5、“BOARD_HAVE_BLUETOOTH := true”指明拥有蓝牙功能;
- 6、“BOARD_HAVE_BLUETOOTH_BCM := true”指明使用 broadcom 系列蓝牙;
- 7、“SW_BOARD_HAVE_BLUETOOTH_NAME := ap6476”指明使用 ap6476 (注意小写) 蓝牙;
- 8、“BLUETOOTH_ENABLE_FM := true”指明编译 FM, 如果不需要 FM, 则改成 false 即可;

6.3.2. init.sun6i.rc

init.sun6i.rc 是资源和服务配置相关的文件, 使用 ap6476 wifi 模组需要作如下修改:

```
# broadcom wifi service
# 1 broadcom wifi sta service
service wpa_supplicant /system/bin/wpa_supplicant \
    -iwlan0 -Dnl80211 -c/data/misc/wifi/wpa_supplicant.conf \
    -I/system/etc/wifi/wpa_supplicant_overlay.conf \
    -O/data/misc/wifi/sockets \
    -e/data/misc/wifi/entropy.bin -g@android:wpa_wlan0
```



```
# we will start as root and wpa_supplicant will switch to user wifi
# after setting up the capabilities required for WEXT
# user wifi
# group wifi inet keystore
class main
socket wpa_wlan0 dgram 660 wifi wifi
disabled
oneshot

# 2 broadcom wifi sta p2p concurrent service
service p2p_supplicant /system/bin/wpa_supplicant \
-iwlan0 -Dnl80211 -c/data/misc/wifi/wpa_supplicant.conf \
-I/system/etc/wifi/wpa_supplicant_overlay.conf \
-O/data/misc/wifi/sockets -N \
-ip2p0 -Dnl80211 -c/data/misc/wifi/p2p_supplicant.conf \
-I/system/etc/wifi/p2p_supplicant_overlay.conf \
-puse_p2p_group_interface=1 -e/data/misc/wifi/entropy.bin \
-g@android:wpa_wlan0
# we will start as root and wpa_supplicant will switch to user wifi
# after setting up the capabilities required for WEXT
# user wifi
# group wifi inet keystore
class main
socket wpa_wlan0 dgram 660 wifi wifi
disabled
oneshot
```

说明:

2、init.sun6i.rc 文件指定开启 wpa_supplicant 和 p2p_supplicant 两个服务, 并设置 supplicant 的启动参数;

```
#insmod boardcom wifi driver
insmod /system/vendor/modules/bcmdhd.ko
```

说明:

1、加载 broadcom wifi 驱动 bcmdhd.ko;

```
# bluetooth
# UART device
chmod 0660 /dev/ttyS2
chown bluetooth net_bt_stack /dev/ttyS2
# power up/down interface
chmod 0660 /sys/class/rfkill/rfkill0/state
chmod 0660 /sys/class/rfkill/rfkill0/type
```



```
chown bluetooth net_bt_stack /sys/class/rfkill/rfkill0/state
chown bluetooth net_bt_stack /sys/class/rfkill/rfkill0/type
# bluetooth LPM
chmod 0220 /proc/bluetooth/sleep/lpm
chmod 0220 /proc/bluetooth/sleep/btwrite
chown bluetooth net_bt_stack /proc/bluetooth/sleep/lpm
chown bluetooth net_bt_stack /proc/bluetooth/sleep/btwrite
```

说明:

- 1、设置蓝牙属性，设置蓝牙相关文件节点权限和用户组等操作;

```
#GPS
chmod 777 /system/bin/glgps
chown root system /system/bin/glgps
chmod 777 /system/lib/hw/gps.default.so
chown root system /system/lib/hw/gps.default.so
mkdir /data/gps
chmod 770 /data/gps
chown system net_bt_stack /data/gps
```

说明:

- 1、修改 GPS 相关文件的权限;

```
#GPS Service
service gpsd /system/bin/glgps -c /system/etc/gps/gpsconfig.xml
    class main
    user root
    group net_bt_stack
```

说明:

- 1、指定开启 GPS 服务;

```
# talking standby and fm standby
chown radio system /sys/power/scene_lock
chmod 0664 /sys/power/scene_lock
chown radio system /sys/power/scene_unlock
chmod 0664 /sys/power/scene_unlock
chown radio system /sys/power/scene_state
chmod 0664 /sys/power/scene_state
```

说明:

- 2、设置与 fm standby 相关的文件节点的属性;

6.3.3. fiber-a31st.mk

fiber-a31st.mk 文件会在编译时拷贝指定文件到特定目录下，使用 ap6476 需要做如下修改:

```
# wifi & bt config file
PRODUCT_COPY_FILES += \
    frameworks/native/data/etc/android.hardware.wifi.xml:system/etc/permissions/android.hardware.wifi.xml \
    frameworks/native/data/etc/android.hardware.wifi.direct.xml:system/etc/permissions/android.hardware.wifi.direct.xml \
    frameworks/native/data/etc/android.hardware.bluetooth.xml:system/etc/permissions/android.hardware.bluetooth.xml \
    frameworks/native/data/etc/android.hardware.bluetooth_le.xml:system/etc/permissions/android.hardware.bluetooth_le.xml

# build Bluetooth.apk
PRODUCT_PACKAGES += \
    Bluetooth
```

说明:

- 1、将配置文件 android.hardware.wifi.xml 拷贝到指定目录，以使设置界面显示 wifi;
- 2、将配置文件 android.hardware.wifi.direct.xml 拷贝到指定目录，以使设置界面显示 wifi direct;
- 3、将配置文件 android.hardware.bluetooth.xml 拷贝到指定目录，以使设置界面显示蓝牙;
- 4、将配置文件 android.hardware.bluetooth_le.xml 拷贝到指定目录，以支持蓝牙低功耗设备;
- 5、“PRODUCT_PACKAGES += Bluetooth” 指定去编译 Bluetooth.apk;

```
#GPS(Enable follow copy procedure if you need GPS feature)
include device/softwinner/fiber-a31st/gps/gps.mk

PRODUCT_COPY_FILES += \
    frameworks/native/data/etc/android.hardware.location.gps.xml:system/etc/permissions/android.hardware.location.gps.xml
```

说明:

- 1、包含编译文件 gps.mk，用于将 GPS 库文件拷贝到指定目录;
- 2、将配置文件 android.hardware.location.gps.xml 拷贝到指定目录，以表示系统拥有 GPS 功能;

```
PRODUCT_PACKAGES += \
    com.broadcom.bt \
    com.broadcom.bt.xml \
    libbt-client-api
```

说明:

- 1、指定编译 broadcom 提供的蓝牙与 FM 包;

6.3.4. config.xml

ap6476 支持 softap 功能，在界面上显示 softap，需在 android\device\softwinner\xx-platform-xx

\overlay\frameworks\base\core\res\res\values\config.xml 做如下配置:

```
<!-- List of regexpressions describing the interface (if any) that represent tetherable
      Wifi interfaces.  If the device doesn't want to support tethering over Wifi this
      should be empty.  An example would be "softap.*" -->
<string-array translatable="false" name="config_tether_wifi_regexs">
  <item>"wlan0"</item>
</string-array>
```

ap6476 支持 miracast 功能, 在界面上显示 miracast, 需在 android\device\softwinner\xx-platform-xx\overlay\frameworks\base\core\res\res\values\config.xml 做如下配置:

```
<!-- Whether WiFi display is supported by this device.
      There are many prerequisites for this feature to work correctly.
      Here are a few of them:
      * The WiFi radio must support WiFi P2P.
      * The WiFi radio must support concurrent connections to the WiFi display and
        to an access point.
      * The Audio Flinger audio_policy.conf file must specify a rule for the "r_submix"
        remote submix module.  This module is used to record and stream system
        audio output to the WiFi display encoder in the media server.
      * The remote submix module "audio.r_submix.default" must be installed on the device.
      * The device must be provisioned with HDCP keys (for protected content).
-->
<bool name="config_enableWifiDisplay">true</bool>
```

ap6476 支持蓝牙共享网络功能, 在界面上显示蓝牙共享网络, 需在 android\device\softwinner\xx-platform-xx\overlay\frameworks\base\core\res\res\values\config.xml 做如下配置:

```
<!-- default: disable Bluetooth PAN feature -->
<string-array translatable="false" name="config_tether_bluetooth_regexs">
  <item>"bt-pan"</item>
</string-array>
```

6.3.5. 波特率修改方法

SDK 内核默认支持的最高串口波特率是 1.5M, 可通过修改 sys_config.fex 文件和 vnd_softwinner-ap6476.txt 文件, 把最高串口波特率改成 3M。

sys_config.fex 的修改是把 pll6 的值由 600 改成 480, 并在 pll10 下添加 apb2 子项, 赋值为 48:

```
[clock]
pll3          = 297
pll4          = 300
pll6         = 480
pll7          = 297
```



```
pll8          = 360  
pll9          = 297  
pll10         = 702  
apb2          = 48
```

vnd_softwinner-ap6476.txt 的修改是把 VendorBaudrate 的值由 1500000 改成 3000000, bt_vendor.conf 文件路径: android\hardware\broadcom\libbt\include\vnd_softwinner-ap6476.txt:

```
BLUETOOTH_UART_DEVICE_PORT = "/dev/ttyS2"  
FW_PATCHFILE_LOCATION = "/system/vendor/modules/"  
LPM_IDLE_TIMEOUT_MULTIPLE = 5  
UART_TARGET_BAUD_RATE = 3000000
```

6.3.6. 蓝牙 PCM 支持

ap6476 支持蓝牙语音功能, 在没有 3G 或 2G 情况下, 可使用蓝牙耳机进行 google talk、skype 等网络通话, 支持蓝牙 PCM 需做如下修改:

sys_config.fex 修改成如下, 有两处修改, 分别是:

- 把 pcm_used 设成 1, 代指使用该驱动;
- 把 pcm_sync_period 设成 16, 这是根据蓝牙模组端做匹配调整;

```
[pcm_para]  
pcm_used      = 1  
pcm_channel   = 2  
pcm_master    = 4  
pcm_select    = 1  
audio_format  = 4  
signal_inversion = 3  
over_sample_rate = 512  
sample_resolution = 16  
word_select_size = 32  
pcm_sync_period = 16  
msb_lsb_first = 0  
sign_extend   = 0  
slot_index    = 0  
slot_width    = 16  
frame_width   = 1  
tx_data_mode  = 0  
rx_data_mode  = 0  
pcm_mclk      =
```

sun6i-pcm.c (\lichee\linux-3.3\sound\soc\sun6i\pcm\sun6i-pcm.c)

- 修改 pcm 的 clk 为 128k, 与蓝牙模组端相匹配;

```
static int sun6i_pcm_set_clkdiv(struct snd_soc_dai *cpu_dai, int div_id, int div
    /*blk div caculate*/
    blk_div = mclk/(2*word_select_size);
    } else {
        mclk_div = 12;
        bclk_div = 16;
    }

    /*calculate MCLK Divide Ratio*/
```

[init.sun6i.rc](#) (\android\device\softwinner\fiber-xxx\init.sun6i.rc)

a. 加载 pcm 相关的驱动;

```
#insmod pcm
insmod /system/vendor/modules/sndpcm.ko
insmod /system/vendor/modules/sun6i-pcm.ko
insmod /system/vendor/modules/sun6i-pcmdma.ko
insmod /system/vendor/modules/sun6i-sndpcm.ko
```

[bt_vendor.conf](#) (android\hardware\broadcom\libbt\conf\softwinner\ap6476\bt_vendor.conf)

修改蓝牙 PCM 参数如下，三处修改:

- 蓝牙模组设成 pcm 通道;
- 蓝牙模组端设成从端;
- 蓝牙模组 clk 设成 128k;

```
AP6476_PCM2_Setup=xx:00:xx:00:F4:01:00:xx:0F:00:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx:xx
#There are total 25 configure, the first one is Test_option. 00:40:1F:00, is frequency settings.
#Just leave the field that you don't want to change as "xx"
#For more detail, please check BCM2076_PCM2_Setup_Command.pdf
```

[audio_hw.c](#) (android\device\softwinner\fiber-common\hardware\audio\audio_hw.c)

针对 BT SCO 进行输入输出通道切换;

[pcm.c](#) (android\external\tinyalsa\pcm.c)

配合 [audio_hw.c](#) 做修改;

6.4. 代码修改

由于目前的 SDK 中没有支持 ap6476，要支持 ap6476 需要添加和修改如下文件:

[lichee\linux-3.3\arch\arm\mach-sun6i\rf](#)

添加 ap6476 的电源管理;

[\lichee\linux-3.3\drivers\net\wireless\bcmdhd](#)

对比修改 broadcom 驱动，用于修复 bug;

android\external\bluetooth\bluedroid\

对比修改 bluedroid，增加 ap6476 蓝牙针对性管理；

android\frameworks\base\

对比修改 frameworks_base，修改包括蓝牙部分和 audio 部分；

android\hardware\broadcom\

对比修改和添加相关文件，添加 ap6476 的 firmware，以及蓝牙配置；

android\hardware\libhardware\

蓝牙与 FM 相关修改；

android\hardware\libhardware_legacy\

audio 相关修改，添加 FM audio 支持；

android\packages\apps\Bluetooth\

对比修改；

将 broadcom 文件夹添加到 android\目录下，该文件夹主要为 FM 代码；

将 gps 文件夹添加到 android\device\softwinner\fiber-a31st\目录下，该文件夹主要为 GPS 库文件；

6.5. ap6476 模组移植相关文件

以下文件是与 ap6476 模组移植相关的，无需再对这些文件作修改，只需了解即可。

一、ap6476 驱动代码

\linux-3.3\drivers\net\wireless\bcmdhd

二、电源及 GPIO 控制

\lichee\linux-3.3\arch\arm\mach-sun6i\rf\wifi_pm.c

\lichee\linux-3.3\arch\arm\mach-sun6i\rf\wifi_pm_ap6xxx.c

\lichee\linux-3.3\arch\arm\mach-sun6i\rf\bt_pm.c

wifi_pm_ap6xxx.c 是 ap6474 模组电源和 GPIO 控制的实现文件，需要把 ap6476 GPIO 控制实现函数接口添加到 wifi_pm.c。

bt_pm.c 是蓝牙电源管理的实现文件，需要把 ap6476 bt 的电源控制添加到 rkill_set_power 函数。

三、wifi firmware 和 nvram

\android\hardware\broadcom\wlan\bcmdhd\firmware\ap6476\

\android\hardware\broadcom\wlan\bcmdhd\firmware\firmware-bcm.mk

firmware-bcm.mk 通过 Boardconfig.mk 中指定的 SW_BOARD_USR_WIFI := AP6476，将 ap6476 目录下的 firmware 和 nvram 拷贝到指定目录下。

7. rtl8723bs

模块组成: wifi、蓝牙

wifi 功能: wifi station、softap、wifi direct

接口类型: SDIO、UART

7.1. 内核配置

7.1.1. wifi 部分

rtl8723bs 模组所使用的 wifi 驱动为 8723bs, 代码路径为\linux-3.3\drivers\net\wireless\rtl8723bs\, 编译内核前确认已将其选择上, 并编译成模块, 查看方法如下:

```
去到 linux-3.3 目录下, 输入 make ARCH=arm menuconfig  
然后选择 Device Drivers --->  
    Network device support --->  
        Wireless LAN --->  
            <M> Realtek 8723B SDIO WiFi
```

7.1.2. 蓝牙部分

rtl8723bs 模组支持蓝牙唤醒主控功能, 内核默认没有开启此功能, 开启该功能需配置如下:

```
去到 linux-3.3 目录下, 输入 make ARCH=arm menuconfig  
然后选择[*] Networking support --->  
    <*> Bluetooth subsystem support --->  
        Bluetooth device drivers --->  
            <*> Realtek Bluesleep driver support
```

7.2. 引脚配置

rtl8723bs 引脚在\tools\pack_brandy\chips\sun6i\configs\android\xx-platform-xx\sys_config.fex 中定义, 需要根据硬件原理图做相应配置, 需要配置的选项有:

7.2.1. wifi 部分

```
-----  
;wifi configuration  
;wifi_sdc_id    ---  0- SDC0, 1- SDC1, 2- SDC2, 3- SDC3  
;wifi_usbc_id  ---  0- USB0, 1- USB1, 2- USB2  
;wifi_usbc_type --  1- EHCI(speed 2.0), 2- OHCI(speed 1.0)  
;wifi_mod_sel  ---  0- none, 1- ap6181, 2- ap6210(wifi+bt),  
;                3- rtl8723as(wifi+bt), 4- rtl8189es(SM89E00),
```



```
;          5 - rtl8192cu, 6 - rtl8188eu, 7 - mt6620
;          8 - rtl8723au, 9 - ap6330(wifi+bt+fm)
;          10 - ap6476, 11 - rtl8723bs;
-----
[wifi_para]
wifi_used      = 1
wifi_sdc_id    = 1
wifi_usbc_id   = 1
wifi_usbc_type = 1
wifi_mod_sel   = 11
wifi_power     = "axp22_aldo1"

; 11 - rtl8723bs sdio wifi + bt
;rtl8723bs_power_ext1      =
;rtl8723bs_power_ext2      =
;rtl8723bs_chip_en        = port:PL11<1><default><default><0>
rtl8723bs_wl_regon        = port:PM00<1><default><default><0>
rtl8723bs_wl_host_wake    = port:PM01<0><default><default><0>
rtl8723bs_bt_regon        = port:PM03<1><default><default><0>
;rtl8723bs_bt_wake        = port:PG12<1><default><default><0>
rtl8723bs_bt_host_wake    = port:PM04<0><default><default><0>
;rtl8723bs_gpio_power     = "axp22_dldo2"
;rtl8723bs_clk_power      = "axp22_dldo4"
```

说明:

- 1、“;”符号起注释作用;
- 2、“wifi_used”宏赋值为1表示使用wifi,为0表示不使用;
- 3、“wifi_sdc_id”宏表示SDIO接口wifi使用哪个MMC控制器;
- 4、“wifi_mod_sel”宏表示选用哪一款wifi模组;
- 5、“wifi_power”宏表示给模组供电的是axp的axp22_aldo1,根据原理图配置;
- 6、“rtl8723bs_power_ext1”和“rtl8723bs_power_ext2”宏表示,对于部分模组,单靠“wifi_power”一路axp供电电流不够,需要使用两路甚至三路axp供电,如需使用打开注释并赋值axp名,不需要使用则将其注释;
- 7、“rtl8723bs_chip_en”宏表示芯片使能脚,如果模组不需要此引脚则将其注释,如需使用根据原理图配置;
- 8、“rtl8723bs_wl_regon”宏表示wifi使能脚,“rtl8723bs_bt_regon”表示蓝牙使能脚,根据原理图配置;
- 9、“rtl8723bs_wl_host_wake”宏表示wifi唤醒主控引脚,“rtl8723bs_bt_host_wake”宏表示蓝牙唤醒主控引脚,根据原理图配置;
- 10、“rtl8723bs_bt_wake”不需要使用,将其注释即可;
- 11、“rtl8723bs_gpio_power”和“rtl8723bs_clk_power”在模块32k晶振源由主控提供时使用,目前rtl8723bs没有这种设计方式,所以不需要使用将其注释即可;



```
[wakeup_src_para]
cpu_en           = 0
cpu_freq         = 48
;(cpu:apb:ahb)
pll_ratio        = 0x111
dram_selfresh_en = 1
dram_freq        = 36
;wakeup_src0     = port:PL5<2><1><default><default>
wakeup_src_wlan  = port:PM01<2><1><default><default>
wakeup_src_bt    = port:PM04<2><1><default><default>
```

说明:

- 1、“wakeup_wifi”子键是为wifi唤醒主休眠主控而添加;
- 2、“wakeup_wifi”所指定的引脚必须与[wifi_para]主键下rtl8723bs中的“rtl8723bs_wl_host_wake”指定的引脚保持一致;

7.2.2. 蓝牙部分

```
;------;
;blue tooth
;bt_used        ---- blue tooth used (0- no used, 1- used)
;bt_uart_id     ---- uart index
;------;
[bt_para]
bt_used         = 1
bt_uart_id      = 2
bt_wakeup       =
bt_gpio         =
bt_rst          =
```

说明:

- 1、“bt_used”宏赋值为1表示使用蓝牙，为0表示不使用;
- 2、“bt_uart_id”宏表示蓝牙所使用的是哪个uart;

```
[wakeup_src_para]
cpu_en           = 0
cpu_freq         = 48
;(cpu:apb:ahb)
pll_ratio        = 0x111
dram_selfresh_en = 1
dram_freq        = 36
;wakeup_src0     = port:PL5<2><1><default><default>
wakeup_src_wlan  = port:PM01<2><1><default><default>
wakeup_src_bt    = port:PM04<2><1><default><default>
```

说明:

- 1、“wakeup_src_bt”子键是为蓝牙唤醒主体睡眠主控而添加;
- 2、“wakeup_src_bt”所指定的引脚必须与[wifi_para]主键下 rtl8723bs 中的“rtl8723bs_bt_host_wake”指定的引脚保持一致;

7.3. 平台配置

7.3.1. BoardConfig.mk

BoardConfig.mk 文件决定 android 加载哪一款 wifi 模组，要配置成使用 rtl8723bs 模组需要把 BoardConfig.mk 文件的相关代码修改成如下:

```
# bt default config
BOARD_BLUETOOTH_BDROID_BUILDCFG_INCLUDE_DIR :=
device/softwinner/fiber-evb/bluetooth

# 1.1 realtek wifi configuration
BOARD_WIFI_VENDOR := realtek
ifeq ($(BOARD_WIFI_VENDOR), realtek)
    WPA_SUPPLICANT_VERSION := VER_0_8_X
    BOARD_WPA_SUPPLICANT_DRIVER := NL80211
    BOARD_WPA_SUPPLICANT_PRIVATE_LIB := lib_driver_cmd_rtl
    BOARD_HOSTAPD_DRIVER := NL80211
    BOARD_HOSTAPD_PRIVATE_LIB := lib_driver_cmd_rtl

    #SW_BOARD_USR_WIFI := rtl8723as
    #BOARD_WLAN_DEVICE := rtl8723as

    SW_BOARD_USR_WIFI := rtl8723bs
    BOARD_WLAN_DEVICE := rtl8723bs
endif

# 1.2 broadcom wifi support
#BOARD_WIFI_VENDOR := broadcom
ifeq ($(BOARD_WIFI_VENDOR), broadcom)
... ..
endif

# 2. Bluetooth Configuration
# make sure BOARD_HAVE_BLUETOOTH is true for every bt vendor
BOARD_HAVE_BLUETOOTH := true
#BOARD_HAVE_BLUETOOTH_BCM := true
```

```
SW_BOARD_HAVE_BLUETOOTH_NAME := rtl8723bs
```

```
BOARD_HAVE_BLUETOOTH_RTK := true
```

```
BLUETOOTH_HCI_USE_RTK_H5 := true
```

```
ifeq ($(BOARD_HAVE_BLUETOOTH_RTK), true)
include
device/softwinner/fiber-common/hardware/realtek/bluetooth/${SW_BOARD_HAVE_BLUETOOTH_
NAME)/firmware/rtlbtfw_cfg.mk
endif
```

说明:

- 1、“#”符号起注释作用;
- 2、“BOARD_WIFI_VENDOR := realtek”指明使用 realtek 系列 wifi 模组;
- 3、“#BOARD_WIFI_VENDOR := broadcom”指明不使用 broadcom 系统 wifi 模组;
- 4、“SW_BOARD_USR_WIFI := rtl8723bs”、“BOARD_WLAN_DEVICE := rtl8723bs”宏指明使用 rtl8723bs 模组;
- 5、“device/softwinner/fiber-evb/bluetooth”用于设置蓝牙设备名;
- 6、“BOARD_HAVE_BLUETOOTH := true”指明拥有蓝牙功能;
- 7、“BOARD_HAVE_BLUETOOTH_RTK := true”指明使用 realtek 系列蓝牙;
- 8、“SW_BOARD_HAVE_BLUETOOTH_NAME := rtl8723bs”指明使用 rtl8723bs (注意小写) 蓝牙;
- 9、“BLUETOOTH_HCI_USE_RTK_H5 := true”指明蓝牙使用 H5 协议;
- 10、“include device/xxx/rtlbtfw_cfg.mk”用于拷贝蓝牙 fw 和 config 文件;

7.3.2. init.sun6i.rc

init.sun6i.rc 是资源和服务配置相关的文件，使用 rtl8723bs wifi 模组需要作如下修改:

```
# 1. realtek wifi service
# 1.1 realtek wifi sta service
service wpa_supplicant /system/bin/wpa_supplicant \
    -iwlan0 -Dnl80211 -c/data/misc/wifi/wpa_supplicant.conf \
    -O/data/misc/wifi/sockets \
    -e/data/misc/wifi/entropy.bin -g@android:wpa_wlan0
# we will start as root and wpa_supplicant will switch to user wifi
# after setting up the capabilities required for WEXT
# user wifi
# group wifi inet keystore
class main
socket wpa_wlan0 dgram 660 wifi wifi
disabled
oneshot

# 1.2 realtek wifi sta p2p concurrent service
service p2p_supplicant /system/bin/wpa_supplicant \
```



```
-ip2p0 -Dnl80211 -c/data/misc/wifi/p2p_supplicant.conf \  
-e/data/misc/wifi/entropy.bin -N \  
-iwlan0 -Dnl80211 -c/data/misc/wifi/wpa_supplicant.conf \  
-O/data/misc/wifi/sockets \  
-g@android:wpa_wlan0  
class main  
socket wpa_wlan0 dgram 660 wifi wifi  
disabled  
oneshot
```

说明:

- 1、init.sun6i.rc 文件指定开启 wpa_supplicant 和 p2p_supplicant 两个服务，并设置 supplicant 的启动参数；
- 2、需要确定 broadcom 对应的 wpa_supplicant 和 p2p_supplicant 两个服务被注释掉；

```
# bluetooth  
# UART device  
    chmod 0660 /dev/ttyS2  
    chown bluetooth net_bt_stack /dev/ttyS2  
# power up/down interface  
    chmod 0660 /sys/class/rfkill/rfkill0/state  
    chmod 0660 /sys/class/rfkill/rfkill0/type  
    chown bluetooth net_bt_stack /sys/class/rfkill/rfkill0/state  
    chown bluetooth net_bt_stack /sys/class/rfkill/rfkill0/type  
# bluetooth LPM  
    chmod 0220 /proc/bluetooth/sleep/lpm  
    chmod 0220 /proc/bluetooth/sleep/btwrite  
    chown bluetooth net_bt_stack /proc/bluetooth/sleep/lpm  
    chown bluetooth net_bt_stack /proc/bluetooth/sleep/btwrite  
    write /proc/bluetooth/sleep/lpm 1
```

说明:

- 1、设置蓝牙属性，设置蓝牙相关文件节点权限和用户组等操作；

7.3.3. fiber-evb.mk

fiber-evb.mk 文件会在编译时拷贝指定文件到特定目录下，使用 rtl8723bs 需要做如下修改:

```
# wifi & bt config file  
PRODUCT_COPY_FILES += \  
    frameworks/native/data/etc/android.hardware.wifi.xml:system/etc/permissions/android.hardware.wifi.  
xml \  
    frameworks/native/data/etc/android.hardware.wifi.direct.xml:system/etc/permissions/android.hardware.  
e.wifi.direct.xml \  
    frameworks/native/data/etc/android.hardware.bluetooth.xml:system/etc/permissions/android.hardware
```



```
.bluetooth.xml \  
    frameworks/native/data/etc/android.hardware.bluetooth_le.xml:system/etc/permissions/android.  
hardware.bluetooth_le.xml  
  
# build Bluetooth.apk  
PRODUCT_PACKAGES += \  
    Bluetooth
```

说明:

- 1、将配置文件 android.hardware.wifi.xml 拷贝到指定目录，以使设置界面显示 wifi;
- 2、将配置文件 android.hardware.wifi.direct.xml 拷贝到指定目录，以使设置界面显示 wifi direct;
- 3、将配置文件 android.hardware.bluetooth.xml 拷贝到指定目录，以使设置界面显示蓝牙;
- 4、将配置文件 android.hardware.bluetooth_le.xml 拷贝到指定目录，以支持蓝牙低功耗设备;
- 5、“PRODUCT_PACKAGES += Bluetooth” 指定去编译 Bluetooth.apk;

```
PRODUCT_PROPERTY_OVERRIDES += \  
    ro.sf.lcd_density=160 \  
    ro.sf.showhdmisettings=5 \  
    persist.sys.ui.hw=true \  
    ro.product.8723b_bt.used=true \  
    persist.sys.usb.config=mass_storage,adb \  

```

说明:

- 1、设置系统属性 “ro.product.8723b_bt.used=true”，与代码配合使用;

7.3.4. config.xml

rtl8723bs 支持 softap 功能，在界面上显示 softap，需在 android\device\softwinner\xx-platform-xx\overlay\frameworks\base\core\res\res\values\config.xml 做如下配置:

```
<!-- List of regexps describing the interface (if any) that represent tetherable  
Wifi interfaces. If the device doesn't want to support tethering over Wifi this  
should be empty. An example would be "softap.*" -->  
<string-array translatable="false" name="config_tether_wifi_regexps">  
<item>"wlan0"</item>  
</string-array>
```

rtl8723bs 支持 miracast 功能，在界面上显示 miracast，需在 android\device\softwinner\xx-platform-xx\overlay\frameworks\base\core\res\res\values\config.xml 做如下配置:

```
<!-- Whether WiFi display is supported by this device.  
There are many prerequisites for this feature to work correctly.  
Here are a few of them:  
* The WiFi radio must support WiFi P2P.  
* The WiFi radio must support concurrent connections to the WiFi display and
```

to an access point.

- * The Audio Flinger audio_policy.conf file must specify a rule for the "r_submix" remote submix module. This module is used to record and stream system audio output to the WiFi display encoder in the media server.
- * The remote submix module "audio.r_submix.default" must be installed on the device.
- * The device must be provisioned with HDCP keys (for protected content).

-->

```
<bool name="config_enableWifiDisplay">true</bool>
```

rtl8723bs 支持蓝牙共享网络功能，在界面上显示蓝牙共享网络，需在 android\device\softwinner\xx-platform-xx\overlay\frameworks\base\core\res\res\values\config.xml 做如下配置：

```
<!-- default: disable Bluetooth PAN feature -->  
<string-array translatable="false" name="config_tether_bluetooth_regexs">  
  <item>"bt-pan"</item>  
</string-array>
```

7.4. rtl8723bs 模组移植相关文件

以下文件是与 rtl8723bs 模组移植相关的，无需再对这些文件作修改，只需了解即可。

一、rtl8723bs 驱动代码

linux-3.3\drivers\net\wireless\rtl8723bs\

二、电源及 GPIO 控制

lichee\linux-3.3\arch\arm\mach-sun6i\rf\wifi_pm.c

lichee\linux-3.3\arch\arm\mach-sun6i\rf\wifi_pm_rtl8723bs.c

lichee\linux-3.3\arch\arm\mach-sun6i\rf\bt_pm.c

wifi_pm_rtl8723bs.c 是 rtl8723bs 模组电源和 GPIO 控制的实现文件，需要把 rtl8723bs GPIO 控制实现函数接口添加到 wifi_pm.c。

bt_pm.c 是蓝牙电源管理的实现文件，需要把 rtl8723bs bt 的电源控制添加到 rkill_set_power 函数。

三、蓝牙 firmware

rtl8723bs 的蓝牙 firmware 和配置文件位于：

android\device\softwinner\fiber-common\hardware\realtek\bluetooth\

8. F&Q

8.1. 如何编译

一、如果只修改内核配置或者只修改内核驱动文件，可以在 `lichee` 目录下整体编译，将生成的驱动文件推进目标机器，然后重启目标机器即可，如修改 `bcmdhd` 驱动，则将 `bcmdhd.ko` 推进目标机器，方法如下：

```
adb push D:\bcmdhd.ko /system/vendor/modules/  
adb shell chmod 777 /system/vendor/modules/bcmdhd.ko
```

二、如果只修改了引脚配置，即只修改了 `sys_config.fex` 文件，则只需重新执行 `pack` 命令，将打包出来的固件重新烧写即可。

三、如果修改了平台配置，有两种编译方法可以使得修改生效，第一种是整体编译重烧固件，第二种是单独的模块编译然后把生成的文件 `push` 进相应的目录，详细操作如下。

整体编译

整体编译有两种方法可选

方法一：在 `android` 主目录下先“`make clean`”后再“`make -j8`”；

方法二：分别到 `android\hardware\libhardware_legacy\wifi` 和 `external\wpa_supplicant_8\wpa_supplicant` 目录下使用“`touch *`”命令更新目录下所有文件的时间戳，再把 `android` 主目录下使用“`make -j8`”命令编译；

单独模块编译

先分别到 `android\hardware\libhardware_legacy\wifi` 和 `external\wpa_supplicant_8\wpa_supplicant` 目录下使用“`touch *`”命令更新目录下所有文件的时间戳，然后在相应目录下使用“`mm`”命令进行编译，再把生的 `libhardware_legacy.so` 和 `wpa_supplicant` 推入目标机器目录，最后更改执行权限并重启。

若生成的 `libhardware_legacy.so` 和 `wpa_supplicant` 存放在电脑 D 盘下，则用如下命令把其推到目标机器目录下。

```
adb push D:\wap_supplicant /system/bin/wpa_supplicant  
adb push D:\libhardware_legacy.so /system/lib/libhardware_legacy.so  
adb shell chmod 777 /system/bin/ wpa_supplicant  
adb shell chmod 666 /system/lib/libhardware_legacy.so
```

注意：单独模块编译适合调试使用。

8.2. 编译报错

修改配置文件后，如果编译的时候报与 `wpa_supplicant_8` 相关的错误，可用以下两种方法解决。

方法一：将 out/target/product/fiber-方案名/obj/EXECUTABLES/wpa_supplicant_intermediates 下面的中间生成文件删除，再次编译。

方法二：使用方法一后编译还是报错，则需要先使用 make clean 命令清除中间文件和目标文件后，再使用 make -j8 命令进行整体编译。

8.3. 修改不起效

修改配置文件并进行整体编译重烧固件或单独模块编译把生成文件推进目标机器后发现指定的 wifi 模组不能工作，通过日志信息却看到了前一款 wifi 模组的相关信息。该问题是因为由于只修改配置文件，wifi 相关的编译文件的时间戳没改变，导致一部分文件没有重新编译，详细解决方法可参考“如何编译”。

8.4. 常见问题解决

1、broadcom wifi 打开失败

对于 broadcom wifi，如果遇到无法打开现象，需按如下步骤进行问题排查：

- ① 通过 adb shell lsmod 命令，检查 bcmhdhd 驱动是否已经加载上；
如果没有加载上，检查是否已经在 init.sun6i.rc 中添加如下代码：

```
#insmod boardcom wifi driver  
insmod /system/vendor/modules/bcmhdhd.ko
```

- ② 查看打开 wifi 过程中的 kernel 打印；
如果看到很多 mmc 相关的报错，需要核对 sys_config.fex 中的引脚配置和原理图中的引脚配置是否一致，测量供电引脚是否已经成功上电。
- ③ 如果上电正常，且没有 mmc 错误，但是有错误打印 dhd_bus_rxctl: rxcnt_timeout=1, rxlen=0；
检查 WL-WAKE-HOST 脚与主控相连之间是否连接三极管，正确配置：

```
Device Drivers --->  
    Network device support --->  
        Wireless LAN --->  
            [ ] Low level trigger for OOB interrupt
```

2、broadcom 蓝牙打开失败

对于 broadcom 蓝牙，如果遇到无法打开现象，需按如下步骤进行问题排查：

- ① 检查是否已经在 fiber-xxx.mk 中添加编译 Bluetooth.apk，添加如下代码：

```
# build Bluetooth.apk  
PRODUCT_PACKAGES += \  
    Bluetooth
```

- ② 需要核对 sys_config.fex 中的引脚配置和原理图中的引脚配置是否一致，测量供电引脚是否已经成功上电。
- ③ 检查模组有正确的 32k clk 时钟输入。

3、broadcom 蓝牙能打开，但是经常自动关闭

① 检查是否已经将蓝牙 lpm 功能支持已经选上，检查方法如下：

```
去到 linux-3.3 目录下，输入 make ARCH=arm menuconfig
然后选择[*] Networking support --->
    <*> Bluetooth subsystem support --->
        Bluetooth device drivers --->
            <*> Bluetooth Low Power Manager Support
```

同时根据 BT-WAKE-HOST 脚与主控相连之间是否接有三极管，正确配置：

```
Networking support --->
    <*> Bluetooth subsystem support --->
        Bluetooth device drivers --->
            <> An inverter between bt hostwake pin and cpu
```

② 检查是否在 init.sun6i.rc 中已经设置蓝牙属性、相关文件节点权限和用户组等：

```
# bluetooth
# UART device
    chmod 0660 /dev/ttyS2
    chown bluetooth net_bt_stack /dev/ttyS2
# power up/down interface
    chmod 0660 /sys/class/rfkill/rfkill0/state
    chmod 0660 /sys/class/rfkill/rfkill0/type
    chown bluetooth net_bt_stack /sys/class/rfkill/rfkill0/state
    chown bluetooth net_bt_stack /sys/class/rfkill/rfkill0/type
# bluetooth LPM
    chmod 0220 /proc/bluetooth/sleep/lpm
    chmod 0220 /proc/bluetooth/sleep/btwrite
    chown bluetooth net_bt_stack /proc/bluetooth/sleep/lpm
    chown bluetooth net_bt_stack /proc/bluetooth/sleep/btwrite
```

8.5. 蓝牙注意事项

若平台不支持蓝牙，需要确定以下三点以保证不影响 wifi 的正常使用。

- 1、内核中把蓝牙相关的编译去除（可参考概述中的方法）；
- 2、sys_config.fex 中 bt_used 项应该赋值为 0 以代码不使用蓝牙；
- 3、平台文件 init.xxx.rc 文件中以下的蓝牙属性设置是不存在或被注释起来的；

```
## bluetooth
## UART device
#   chmod 0660 /dev/ttyS2
#   chown bluetooth net_bt_stack /dev/ttyS2
## power up/down interface
#   chmod 0660 /sys/class/rfkill/rfkill0/state
#   chmod 0660 /sys/class/rfkill/rfkill0/type
#   chown bluetooth net_bt_stack /sys/class/rfkill/rfkill0/state
```



```
# chown bluetooth net_bt_stack /sys/class/rfkill/rfkill0/type
## bluetooth LPM
# chmod 0220 /proc/bluetooth/sleep/lpm
# chmod 0220 /proc/bluetooth/sleep/btwrite
# chown bluetooth net_bt_stack /proc/bluetooth/sleep/lpm
# chown bluetooth net_bt_stack /proc/bluetooth/sleep/btwrite
```

CONFIDENTIAL

9. Declaration

This document is the original work and copyrighted property of Allwinner Technology (“Allwinner”). Reproduction in whole or in part must obtain the written approval of Allwinner and give clear acknowledgement to the copyright owner.

The information furnished by Allwinner is believed to be accurate and reliable. Allwinner reserves the right to make changes in circuit design and/or specifications at any time without notice. Allwinner does not assume any responsibility and liability for its use. Nor for any infringements of patents or other rights of the third parties which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of Allwinner. This datasheet neither states nor implies warranty of any kind, including fitness for any particular application.

CONFIDENTIAL