

Rockchip

WIFI/BT 开发指南

发布版本:6.0

日期:2019.05

免责声明

本文档按“现状”提供，福州瑞芯微电子股份有限公司（“本公司”，下同）不对本文档的任何陈述、信息和内容的准确性、可靠性、完整性、适销性、特定目的性和非侵权性提供任何明示或暗示的声明或保证。本文档仅作为使用指导的参考。

由于产品版本升级或其他原因，本文档将可能在未经任何通知的情况下，不定期进行更新或修改。

商标声明

“Rockchip”、“瑞芯微”、“瑞芯”均为本公司的注册商标，归本公司所有。

本文档可能提及的其他所有注册商标或商标，由其各自所有者所有。

版权所有 © 2018 福州瑞芯微电子股份有限公司

超越合理使用范畴，非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

福州瑞芯微电子股份有限公司

Fuzhou Rockchip Electronics Co., Ltd.

地址：福建省福州市铜盘路软件园 A 区 18 号

网址：www.rock-chips.com

客户服务电话：+86-591-83991906

客户服务传真：+86-591-83951833

客户服务邮箱：www.rock-chips.com

前言

概述

本文档主要介绍基于 Rockchip 平台的 WIFI、BT 的内核配置、相关功能的开发等等；

产品版本

芯片名称	内核版本
RK3308/3326/3288/3399	4.4

读者对象

本文档（本指南）主要适用于以下工程师：

- 技术支持工程师
- 软件开发工程师

修订记录

日期	版本	作者	修改说明
2018/05/02	0.01	XY	初始版本
2018/05/16	1.0	XY	正式版本
2019/05/01	6.0	XY	完善 bringup 遇到问题的排查方法； 增加 WIFIBT 接口应用开发； 增加 WIFIBT 模组移植； 增加编译规则的说明； 增加硬件测试指南；

目录

目录

前言	1-1
1 WIFIBT 配置	1-1
1.1 DTS 配置	1-1
1.2 内核配置	1-3
1.3 Buildroot 配置.....	1-4
1.4 WIFIBT 的文件及其编译说明	1-4
2 WIFIBT 功能测试	2-5
2.1 WiFi 测试	2-5
2.2 BT 测试	2-7
2.3 WIFIBT MAC 地址.....	2-9
3 WiFi 的无线唤醒 (WoWLAN)	3-9
4 WiFi 的 monitor 模式.....	4-10
4.1 AP6xxx/海华芯片.....	4-10
4.2 Realtek 芯片.....	4-10
5 WIFIBT 硬件指标测试.....	5-10
5.1 测试项目	5-10
5.2 模组.....	5-10
5.3 COB	5-10
5.4 天线和晶振及 32.768K	5-10
5.5 平台测试工具和方法	5-11
6 WIFIBT 问题排查	6-11
6.1 Wlan0 设备无法识别	6-11
6.2 WiFi 无法连接路由器或断线连接不稳定.....	6-12
6.3 WiFi 其他问题.....	6-12
6.4 BT 排查	6-12
6.5 RTL8723DS 特别注意点.....	6-12
7 WIFIBT 应用开发.....	7-13
7.1 WIFI 开发	7-13
7.2 蓝牙开发	7-13
7.3 配网开发 (BLE/SOFTAP/AIRKISS)	7-13
8 WIFIBT 模组移植.....	8-13

前言

此文档主要介绍 RK linux 平台下 WiFi/BT 的开发， 作为该文档的补充：RTL 系列芯片参见:\docs\Linux reference documents: RK3308_RTL8723DS_WIFI_BT_说明文档_V1.20.pdf

1 WIFIBT 配置

1.1 DTS 配置

WIFI 硬件管脚的配置主要有以下几点：

切记一定要对照原理图进行配置，且确保使用的 **dts/dtsi** 里面包含以下节点！

WIFI_REG_ON: WiFi 的电源 PIN 脚

sdio_pwrseq: sdio-pwrseq {

compatible = "mmc-pwrseq-simple";

pinctrl-names = "default";

pinctrl-0 = <&wifi_enable_h>;

reset-gpios = <&gpio0 RK_PA2 GPIO_ACTIVE_LOW>; //有个注意要点是：这里的电平状态恰

好跟使能状态相反，比如 REG_ON 高有效，则这里为 LOW；如果 REG_ON 低有效，则填 HIGH

};

&pinctrl {

sdio-pwrseq {

wifi_enable_h: wifi-enable-h {

rockchip,pins =

<0 RK_PA2 RK_FUNC_GPIO &pcfg_pull_none>; // 对应上面的 WIFI_REG_ON

};

};

};

&sdio {

bus-width = <4>;

... ..

status = "okay";

};

WIFI_WAKE_HOST: WIFI 唤醒主控的 PIN 脚

wireless-wlan {

compatible = "wlan-platdata";

rockchip,grf = <&grf>;

wifi_chip_type = "ap6255"; //海华/正基模组可以不用修改此名称，realtek 需要按实际填写

WIFI,host_wake_irq = <&gpio0 RK_PA0 GPIO_ACTIVE_HIGH>; // WIFI_WAKE_HOST

GPIO_ACTIVE_HIGH 特别注意：确认下这个 wifi pin 脚跟主控的连接关系，直连的话就是 HIGH，如果中间加了一个反向管就要改成低电平 LOW 触发

status = "okay";

};

wireless-bluetooth {

compatible = "bluetooth-platdata";

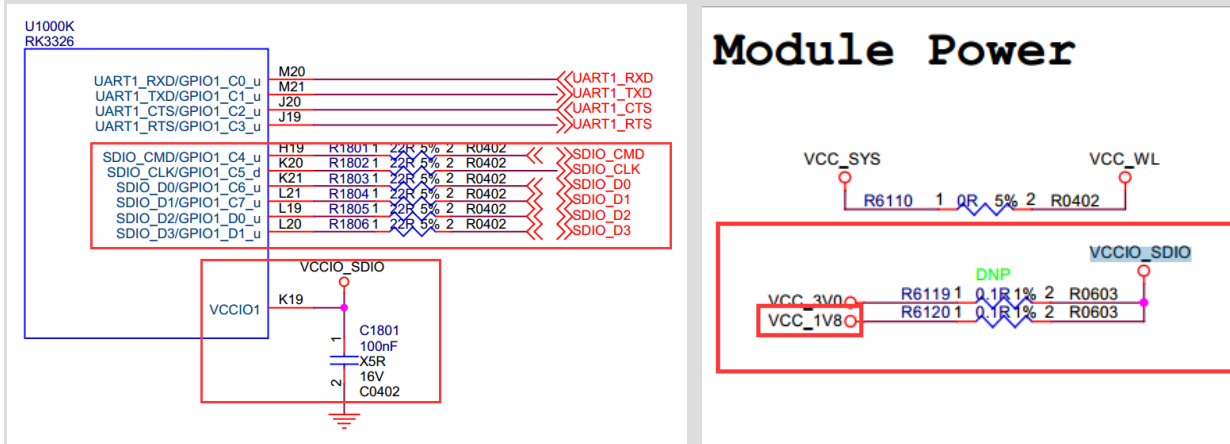
uart_rts_gpios = <&gpio4 RK_PA7 GPIO_ACTIVE_LOW>;

```

pinctrl-names = "default", "rts_gpio";
pinctrl-0 = <&uart4_rts>;
pinctrl-1 = <&uart4_rts_gpio>;
BT,power_gpio    = <&gpio4 RK_PB3 GPIO_ACTIVE_HIGH>; // BT_REG_ON
BT,wake_host_irq = <&gpio4 RK_PB4 GPIO_ACTIVE_HIGH>; // BT_WAKE_HOST
status = "okay";
};

```

IO 电源域的配置:



查看原理图，找到 wifi 对应的 sdio 接口部分，图中有标注 vccio_x，比如这个是 vccio1，则给 vccio1 供电的是 vccio_sdio，查找 vccio_sdio 连接的网路是 3.3v 还是 1.8v，可以看到上图的 vccio_sdio 是 vcc_1v8 供电的，则对应 dts/dtsi 配置如下：

```

&io_domains {
    vccio1-supply = <&vccio_sdio>; //vccio1 引用 vccio_sdio
    vccio2-supply = <&vccio_sd>;
    ...
};

vccio_sdio: vcc_1v8: vcc-1v8 { //vccio_sdio 引用 vcc_1v8,
    compatible = "regulator-fixed";
    regulator-name = "vcc_1v8";
    regulator-always-on;
    regulator-boot-on;
    regulator-min-microvolt = <1800000>; //vcc_1v8 供电 1.8v
    regulator-max-microvolt = <1800000>;
    vin-supply = <&vcc_io>;
};

```

以上配置要一一对应，如果硬件是 3.3v，按照对应关系进行修改

32.768K

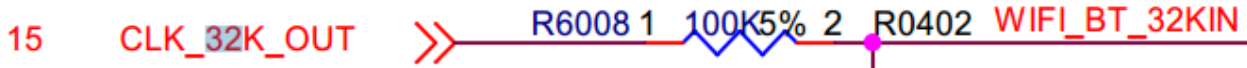
如果 WiFi 模组需要外部供这个频率，则有两种情况：

- 1、如果有带 RK8XX 型号的 PMU，则直接接到相关管脚；
- 2、如果没有，则需要 CPU 去供，dts 需要添加如下配置：

```

GPIO0_C3/RTC_CLK | Y2 |>> CLK_32K_OUT 23

```



```
&pinctrl {
    pinctrl-names = "default";
    pinctrl-0 = <&rtc_32k>;
}
```

1.2 内核配置

根据实际 **WiFi** 选择对应配置

```
CONFIG_WL_ROCKCHIP:
Enable compatible wifi drivers for Rockchip platform.
Symbol: WL_ROCKCHIP [=y]
Type : boolean
Prompt: Rockchip wireless LAN support
Location:
-> Device Drivers
-> Network device support (NETDEVICES [=y])
-> Wireless LAN (WLAN [=y])
Defined at drivers/net/wireless/rockchip_wlan/kconfig:2
Depends on: NETDEVICES [=y] && WLAN [=y]
Selects: WIRELESS_EXT [=y] && WEXT_PRIV [=y] && CFG80211 [=y] && MAC80211 [=y]
```

```
--- Rockchip wireless LAN support
[ ] build wifi ko modules
[*] wifi load driver when kernel bootup
< > ap6xxx wireless sdio cards support
< * > Cypress wireless sdio cards support
[ ] Realtek wireless Device Driver Support ----
< > Realtek 8723B SDIO or SPI WiFi
< > Realtek 8723C SDIO or SPI WiFi
< > Realtek 8723D SDIO or SPI WiFi
< > Marvell 88W8977 SDIO WiFi
```

注意事项:

对应 **buildin** 方式 (默认推荐):

```
[ ] build wifi ko modules
[*] wifi load driver when kernel bootup
```

a、只能选择一个型号，realtek 模组和 ap6xxx 模组不能同时选择为 y，且 realtek 的系列模组也只能选择其中一个；

b、ap6xxx 和 cypress 也是互斥的，只能选择一个，且如果选择 ap6xxx，cypress 的配置自动消失，去掉 ap 配置，cypress 自动出现；

对于 **ko** 方式:

```
[*] build wifi ko modules
[ ] wifi load driver when kernel bootup
```

且没有 buildin 的限制，可以选择多个 wifi；

BT 配置

AP 和海华系列内核无需配置，对于 realtek 的需要如下修改：(realtek 使用自己的 hciuart 驱动，源码目录为 external\rkwifibt\realtek\bluetooth_uart_driver，使用 ko 方式加载)

去掉默认内核默认 CONFIG_BT_HCIUART 驱动

```
Symbol: BT_HCIUART [=n]
Type : tristate
Prompt: HCI UART driver
Location:
  -> Networking support (NET [=y])
    -> Bluetooth subsystem support (BT [=y])
(1)   -> Bluetooth device drivers
Defined at drivers/bluetooth/kconfig:72
Depends on: NET [=y] && BT [=y] && TTY [=y]
```

```
< > HCI SDIO driver
< > HCI UART driver
< > HCI VHCI (virtual HCI device) driver
< > Marvell Bluetooth driver support
```

1.3 Buildroot 配置

根据实际 WiFi 选择对应配置，要跟**内核配置一致**：

```
There is no help available for this option.
Prompt: wifi chip support
Location:
  -> Target packages
    -> rockchip BSP packages (BR2_PACKAGE_ROCKCHIP [=y])
      -> rkwifi (BR2_PACKAGE_RKWIFIBT [=y])
Defined at package/rockchip/rkwifibt/config.in:5
Depends on: BR2_PACKAGE_ROCKCHIP [=y] && BR2_PACKAGE_RKWIFIBT [=y]
Selected by: BR2_PACKAGE_ROCKCHIP [=y] && BR2_PACKAGE_RKWIFIBT [=y] && m
```

```
wifi chip support
Use the arrow keys to navigate this window or press the
hotkey of the item you wish to select followed by the <SPACE
BAR>. Press <?> for additional information about this

( ) AP6255
( ) AP6212A1
( ) AW-CM256
( ) AW-NAB197
(X) RTL8723DS
( ) RTL8189FS

<Select> < Help >
```

对于有带蓝牙的模组，要配置对应的 **tty** 编号，对应硬件的 **uart** 口：

```
--- rkwifi
wifi chip support (AP6255) --->
(ttyS4) bt uart
```

1.4 WiFiBT 的文件及其编译说明

涉及的文件：

RK 平台适配的 WiFi 驱动对应的文件目录：kernel/drivers/net/wireless/rockchip_wlan/

RK 平台适配的 BT 驱动及蓝牙 firmware 文件目录对应: external/rkwifibt/
AP 模组: external/rkwifibt/firmware/broadcom/
Realtek 模组: external/rkwifibt/realtek/
对应的编译规则: buildroot/package/rockchip/rkwifibt/rkwifibt.mk (Config.in)

编译说明:

1 请仔细阅读这 **rkwifibt.mk**、**Config.in** 文件, 这两个文件主要完成对应的 WiFi 模组的 firmware 的拷贝、对应模组蓝牙驱动以及可执行文件的编译拷贝, 所以**开发人员必须熟悉编译规则, 这对于后面的调试非常重要**;

2 对于 kernel wifi 配置的修改, make menuconfig 选择完成后, 一定要修改对应的 defconfig 文件, 比如我使用的是: kernel/arch/arm/configs/rockchip_xxxx_defconfig, 要把对应的修改更新到这个文件。否则使用 **./build.sh kernel** 脚本编译时会被原来覆盖掉, 导致修改未生效;

3 对于 buildroot 配置的修改, make menuconfig 选择完成后, 在根目录执行 make savedefconfig 进行保存。否则也会出现使用 **./build.sh** 脚本编译时会被原来覆盖掉, 导致修改未生效; 修改完且保存后, 执行:

```
make rkwifibt-dirclean //清除掉之前的
make rkwifibt-rebuild //重新编译
./build.sh //重新打包生成固件
```

2 WiFiBT 功能测试

2.1 WiFi 测试

◆ 首先执行 **ifconfig**, 确保出现 **wlan0** 节点:

```
/ # ifconfig wlan0
wlan0      Link encap:Ethernet  HWaddr F0:85:C1:0F:9C:02
           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)
```

如果没有出现 **wlan0** 节点, 请先检查 **dts** 部分是否配置正确, 如果正确则请**参考第 6 章节进行排查**。

◆ 然后查看 WiFi 的服务进程启动: **ps** 看下是否有 **wpa_supplicant** 进程, **如果没启动**可以手动运行:

```
wpa_supplicant -B -i wlan0 -c /data/cfg/wpa_supplicant.conf
```

注意: **wpa_supplicant.conf** 文件请根据实际平台的存放位置进行修改

◆ 扫描周边的 ap

```
wpa_cli -i wlan0 -p /var/run/wpa_supplicant scan
wpa_cli -i wlan0 -p /var/run/wpa_supplicant scan_results
```

```

/ #
/ # wpa_cli -i wlan0 -p /var/run/wpa_supplicant scan_results
bssid / frequency / signal level / flags / ssid
dc:ef:09:a7:77:53      2437      -30      [WPA2-PSK-CCMP] [ESS]      fish1
10:be:f5:1d:a3:74      2447      -34      [WPA-PSK-CCMP+TKIP] [WPA2-PSK-CCMP+TKIP] [ESS]      DLink8808
d4:ee:07:5b:81:80      2432      -35      [WPA-PSK-CCMP] [WPA2-PSK-CCMP] [ESS]      Fang-HiWiFi
76:7d:24:51:39:d0      2422      -35      [WPA-PSK-CCMP+TKIP] [WPA2-PSK-CCMP+TKIP] [ESS]      @PHICOMM_CE
2c:b2:1a:3a:7f:d6      2412      -42      [WPA-PSK-CCMP+TKIP] [WPA2-PSK-CCMP+TKIP] [ESS]      RK_0101
74:05:a5:29:5f:cc      2412      -42      [WPA-PSK-CCMP] [WPA2-PSK-CCMP] [ESS]      TP-LINK_5FJK
24:69:68:98:aa:42      2437      -42      [WPA-PSK-CCMP] [WPA2-PSK-CCMP] [ESS]      ZainAP
d4:ee:07:1c:2d:18      2427      -43      [WPA-PSK-CCMP] [WPA2-PSK-CCMP] [ESS]      ROCKROOM
9c:21:6a:c8:6f:7c      2462      -43      [WPA-PSK-CCMP] [WPA2-PSK-CCMP] [ESS]      TP-LINK_HKH

```

注意：要看下扫描到的热点的个数是否匹配你周围的路由器个数，可以跟你手机 WiFi 扫到对比下（如果你的模组不支持 5G，只对比 2.4G 的个数）；还有就是看下离你最近的路由器的信号强度，如果路由器离你很近，但信号强度却非常弱（正常情况下：-30 到 -55，偏弱：-55 到 -70，非常差 -70 到 -90），这时就要查下你的 WiFi 模组是否有接天线，模组的 RF 指标是否合格等等（参考第 5 章节 WiFiBT 的硬件测试）。

◆ WiFi 连接路由器，两种方法

a) 方法一：

修改如下文件：

```

/ # vi /data/cfg/wpa_supplicant.conf
ctrl_interface=/var/run/wpa_supplicant //注意这个接口配置，如果有修改的话对应 wpa_cli 命令-p 参数要相应进行修改，wpa_cli -i wlan0 -p <ctrl_interface> xxx
ap_scan=1
update_config=1 //这个配置使 wpa_cli 命令配置的热点保存到 conf 文件里面（wpa_cli save_config）
#添加如下配置项
network={
    ssid="WiFi-AP"           // WiFi 名字
    psk="12345678"           // WiFi 密码
    key_mgmt=WPA-PSK         // 填加密方式
    # key_mgmt=NONE          // 如果 wifi 不加密
}

```

让 wpa_supplicant 进程读取上述配置，命令如下：

```
wpa_cli -i wlan0 -p /var/run/wpa_supplicant reconfigure
```

发起连接：

```
wpa_cli -i wlan0 -p /var/run/wpa_supplicant reconnect
```

b) 方法二：

加密：

```

wpa_cli -i wlan0 -p /var/run/wpa_supplicant remove_network 0
wpa_cli -i wlan0 -p /var/run/wpa_supplicant ap_scan 1
wpa_cli -i wlan0 -p /var/run/wpa_supplicant add_network
wpa_cli -i wlan0 -p /var/run/wpa_supplicant set_network 0 ssid "dlink"
wpa_cli -i wlan0 -p /var/run/wpa_supplicant set_network 0 key_mgmt WPA-PSK
wpa_cli -i wlan0 -p /var/run/wpa_supplicant set_network 0 psk "12345678"
wpa_cli -i wlan0 -p /var/run/wpa_supplicant select_network 0
wpa_cli -i wlan0 -p /var/run/wpa_supplicant save_config //保存上述配置到 conf 文件

```

不加密：

```
wpa_cli -i wlan0 -p /var/run/wpa_supplicant remove_network 0
wpa_cli -i wlan0 -p /var/run/wpa_supplicant ap_scan 1
wpa_cli -i wlan0 -p /var/run/wpa_supplicant add_network
wpa_cli -i wlan0 -p /var/run/wpa_supplicant set_network 0 ssid "dlink"
wpa_cli -i wlan0 -p /var/run/wpa_supplicant set_network 0 key_mgmt NONE
wpa_cli -i wlan0 -p /var/run/wpa_supplicant select_network 0
wpa_cli -i wlan0 -p /var/run/wpa_supplicant save_config
```

顺利的话执行 `ifconfig` 可以看到有获取到 IP 地址，也可以使用下面的命令：

```
/ #
/ # wpa_cli -i wlan0 -p /var/run/wpa_supplicant status
bssid=10:be:f5:1d:a3:74
freq=2447
ssid=DLink8808
id=0
mode=station
pairwise_cipher=CCMP
group_cipher=TKIP
key_mgmt=WPA2-PSK
wpa_state=COMPLETED
ip_address=192.168.100.142
address=8c:f7:10:49:3b:8a
/ #
```

如果没有获取到正确的 IP 地址，首先检查下 `dhcpcd` 获取 ip 地址的进程是否有启动，如果没有要先启动该进程，如果 `dhcpcd` 进程已经启动，但还是连不上请检查上一步骤的 `scan` 扫描以及信号强度是否正常，比如我连接的 DLink8808，`scan` 有扫描到该路由器且信号强度非常好。

```
/ #
/ # wpa_cli -i wlan0 -p /var/run/wpa_supplicant scan_results
bssid / frequency / signal level / flags / ssid
dc:ef:09:a7:77:53 2437 -30 [WPA2-PSK-CCMP][ESS] fish1
10:be:f5:1d:a3:74 2447 -34 [WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][ESS] DLink8808
d4:ee:07:5b:81:80 2432 -35 [WPA-PSK-CCMP][WPA2-PSK-CCMP][ESS] Fang-HiWifi
76:7d:24:51:39:d0 2422 -35 [WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][ESS] @PHICOMM_CE
2c:b2:1a:3a:7f:d6 2412 -42 [WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][ESS] RK_0101
74:05:a5:29:5f:cc 2412 -42 [WPA-PSK-CCMP][WPA2-PSK-CCMP][ESS] TP-LINK 5FJK
```

2.2 BT 测试

首先确保 `dtb` 要配置正确，以及对应的 `buildroot` 配置是否正确，请参考第 1 章节，这里对常用的两类 BT 模组进行说明，在配置正确情况下，系统会生成一个 `bt_pcba_test` 的脚本程序：

◆ REALTEK 模组

```
/ # cat usr/bin/bt_pcba_test
#!/bin/sh

echo 0 > /sys/class/rfkill/rfkill0/state //下电
sleep 2
echo 1 > /sys/class/rfkill/rfkill0/state //上电
sleep 2

insmod /usr/lib/modules/hci_uart.ko //realtek 模组需要加载特定驱动
rtk_hciattach -n -s 115200 /dev/ttyS4 rtk_h5 & //蓝色指的是蓝牙使用哪个 uart 口

hciconfig hci0 up
```

◆ AP6XXX/海华模组

```

/ # cat usr/bin/bt_pcba_test
#!/bin/sh

echo 0 > /sys/class/rfkill/rfkill0/state //下电
sleep 2
echo 1 > /sys/class/rfkill/rfkill0/state //上电
sleep 2

brcm_patchram_plus1 --bd_addr_rand --enable_hci --no2bytes
--use_baudrate_for_download --tosleep 200000 --baudrate 1500000 --patchram
/system/etc/firmware/bcm43438a1.hcd /dev/ttyS4 &
//上面有两个重要的参数:红色指的是 bt 对应型号 firmware 文件,蓝色指的是蓝牙使用哪个 uart 口,
一定要确保 firmware 文件存在且匹配对应的型号,且 tty 设备一定要对应。

hciconfig hci0 up

```

注意: `rtk_hciattach`、`hci_uart.ko`、`bcm43438a1.hcd` 等文件都是第 1 章节 **buildroot** 配置选择正确的 **WiFiBT** 模组的前提下才会生成,如果没有这些文件请检查上述配置。

执行该脚本后,执行:

```

hciconfig hci0 up
hciconfig -a

```

正常的情况下可以看到:

```

/ # hciconfig -a
hci0: Type: Primary Bus: UART
      BD Address: 2A:CB:74:E5:DF:92 ACL MTU: 1021:8 SCO MTU: 64:1
      UP RUNNING
      RX bytes:1224 acl:0 sco:0 events:60 errors:0
      TX bytes:796 acl:0 sco:0 commands:60 errors:0
      Features: 0xbf 0xfe 0xcf 0xfe 0xdb 0xff 0x7b 0x87
      Packet type: DM1 DM3 DM5 DH1 DH3 DH5 HV1 HV2 HV3
      Link policy: RSWITCH SNIFF
      Link mode: SLAVE ACCEPT
      Name: 'BCM43438A1 26MHz AP6212A1_CL1 BT4.0 OTP-BD-0058'
      Class: 0x000000
      Service Classes: Unspecified
      Device Class: Miscellaneous,
      HCI Version: 4.0 (0x6) Revision: 0xf9
      LMP Version: 4.0 (0x6) Subversion: 0x2209
      Manufacturer: Broadcom Corporation (15)

```

然后可以进行扫描周围的蓝牙设备:

```

/ # hcitool scan
Scanning ...
D0:C5:D3:92:D9:04 -A11-0308
2C:57:31:50:B3:09 E2
EC:D0:9F:B4:55:06 xing_mi6
5C:07:7A:CC:22:22 _AUDIO
18:F0:E4:E7:17:E2 小米手机123
AC:C1:EE:18:4C:D3 红米手机
B4:0B:44:E2:F7:0F n/a

```

2.3 WiFiBT MAC 地址

一般情况下 wifibt 的 mac 地址都是芯片内置的，如果需要自定义 mac 地址，需要使用 RK 专用工具写到 flash 自定义的 vendor 分区，AP 模组的需要注意参考 docs\Develop reference documents\WIFIBT 目录下的《RK 平台 AP WIFI 自定义 MAC 地址.pdf》文档。

对于自定义方式，例如：

```
读： vendor_storage -r "VENDOR_WIFI_MAC_ID"
    vendor_storage -r "VENDOR_BT_MAC_ID"
写： vendor_storage -w "VENDOR_WIFI_MAC_ID B4021192D25C"
    vendor_storage -w "VENDOR_BT_MAC_ID B4021192D25D"
```

3 WiFi 的无线唤醒（WoWLAN）

目前 WiFi 支持无线网络包唤醒系统，例如：音响设备正常连接 WiFi 并获取到正确的 IP 地址，则当设备休眠后，我们可以通过无线网络包唤醒系统，**唤醒规则：只要是发给这个设备 IP 地址的网络包，都会唤醒系统。**

AP6XXX/RTL 模组上层配置：修改 wpa_supplicant.conf 文件，添加如下配置：

```
wpa_supplicant.conf
ctrl_interface=/var/run/wpa_supplicant
update_config=1
ap_scan=1
+wowlan_triggers=any //确保修改生效
```

Realtek 模组请打开对应驱动的 Makefile 里面的如下配置：

```
/drivers/net/wireless/rockchip_wlan/rtl8xxx/Makefile
CONFIG_WOWLAN = y
CONFIG_GPIO_WAKEUP = y
```

注意：

1、确保配置 **WIFI_WAKE_HOST: WIFI** 唤醒主控的 **PIN** 脚

Dts 的 WiFi 配置：WIFI,host_wake_irq = <&gpio0 RK_PA0 GPIO_ACTIVE_HIGH>;
//WIFI_WAKE_HOST GPIO_ACTIVE_HIGH 特别注意：确认下这个 wifi pin 脚跟主控的连接关系，如果中间加了一个反向管就要改成低电平触发

2、休眠前请确保 **hostapd** 进程关掉，网络唤醒功能要求必须关掉 **hostapd** 进程；

软件参考 ping 源码即可；

测试方法：设备连上 WiFi 并正常获取到 IP 地址，（echo mem > sys/power/state）进休眠后，手机端下载一个 ping 软件（确保手机或者 PC 连接到同一局域网），然后去 ping 设备的 IP 地址，正常的话，可以看到设备会被唤醒。

问题排查：AP 和 RTL 芯片默认都是高电平触发，假设 WiFi_Wake_Host 脚和主控直连，则设备进入休眠后，pin 脚默认低电平，当有网络包唤醒时，这个脚用示波器可以测得高脉冲进来；所以当设备没有被唤醒时请用示波器测下这个 PIN 脚是否符合上述行为。

4 WiFi 的 monitor 模式

启用 WiFi 的 monitor 模式：

4.1 AP6xxx/海华芯片

dhd_priv SDK 自带该命令

设置信道：

```
dhd_priv channel 6 // channel numbers
```

开 monitor 模式：

```
dhd_priv monitor 1
```

关 monitor 模式：

```
dhd_priv monitor 0
```

4.2 Realtek 芯片

驱动 Makefile 需要打开：

CONFIG_WIFI_MONITOR = y

1. ifconfig wlan0 up

ifconfig p2p0 down

2. iwconfig wlan0 mode monitor /* support wext solution.*/

or iw dev wlan0 set type monitor /* support cfg80211 solution.*/

3. echo "<chan> 0 0" > /proc/net/<rtk_module>/wlan0/monitor /* <rtk_module> is the realtek wifi module name, such like rtl8812au, rtl8188eu ..etc */

4. tcpdump -i wlan0 -s 0 -w snf_pkts.pcap /*capture the sniffer packets and save it as a file "snf_pkts.pcap"

docs\Develop reference documents\WIFIB 目录下有抓包示例《WIFI MONITOR 抓包示例.txt》

5 Wi-Fi BT 硬件指标测试

5.1 测试项目

- 芯片部分：（传导测试方法）
发送：发射功率、EVM、频偏；
接收：接收灵敏度；
- 天线部分：增益、驻波比、方向性（天线厂）、天线与板子的阻抗匹配；

5.2 模组

使用模组的话，模组厂会把基本的指标测试校准一轮，然后对应的**校准数据**会内置到模组里面，基本问题不大，但是还是要去测试一轮，确保指标正常；

5.3 COB

相对于模组，COB 基本都是没有校准的，且外围电路、晶振都是需要自行设计布板的，请重点找拿货厂家确认；校准数据必须找他们协助进行测试校准，然后把校准数据写到芯片的 efuse 里面，以上工作都是可以找拿货的厂家协助；

5.4 天线和晶振及 32.768K

主要是去适配天线部分：必须去重点测试；

晶振部分：外置晶振：调频偏；内置晶振：不需要；

32.768K：要确认下你使用的模组或者芯片是否需要外部供给；

5.5 平台测试工具和方法

具体请参考 docs\Develop reference documents\WIFI/BT 目录下的文档

6 Wi-Fi BT 问题排查

6.1 Wlan0 设备无法识别

首先硬件测量 WIFI_REG_ON/VDDIO VBAT/SDIO_CLK/SDIO_CMD/SDIO_DATA0~SDIO_DATA3 的电压,详细描述如下:

- 确认 WIFI_CLK 的信号是否常，加载驱动时，SDIO_CLK 有时钟吐出来，识卡 400~100K 左右，稳定后会达到 DTS 设置的 CLK (50M)。(经常听客户反馈说量不到，这是因为当模组没有正常工作时，主控发了几条命令后没有得到回应，所以就没有继续发，所以必须仔细量下，最好用触发方式抓取波形)
- 确认上电后 37.4/24/26M 有正常起振，且符合对于模组的要求，有发现客户出现晶振正常但不符合模组的频率要求，比如模组要求 37.4M，但贴了一个 24M 的晶振(注意 realtek 模组内部产生不需要外部供);
- 确认 32.768K 方波信号是否正常，峰峰值有要求，必须是 $0.7 * VDDIO \sim VDDIO$ 这个范围内才行。否则会有问题 (注意 Realtek 模组内部产生不需要外部供);
- 确认 WL/WIFI_REG_ON WIFI 上电管脚在打开 WIFI 的时候是否正常被拉高，而且异常的时候是否电平有变化，VBAT 供电是否有波动；其中 DTS 部分的 WIFI_REG_ON (sdio-pwrseq) 的脚配置对应的解析代码操作参见: [drivers/mmc/core/pwrseq_simple.c](#)

解析 reset-gpio:

```
mmc_pwrseq_simple_alloc
```

上下电:

```
static struct mmc_pwrseq_ops mmc_pwrseq_simple_ops = {
    .pre_power_on = mmc_pwrseq_simple_pre_power_on, // 拉低 WiFi_REG_ON
    .post_power_on = mmc_pwrseq_simple_post_power_on, // 拉高 WiFi_REG_ON
    .power_off = mmc_pwrseq_simple_power_off, // 拉低 WiFi_REG_ON
    .free = mmc_pwrseq_simple_free,
};
```

通过示波器可以看到: 在进到内核后 WiFi 初始化时 WiFi_REG_ON 会被先拉低再拉高, 如果观测不到, 请在上面代码处加些 debug 确认;

- 确认晶体部分外围器件物料是否有漏焊或者焊错;
- 确认 sdio 的 4 根 data 走线是否有问题, 是否有干扰 (会导致跑不了高频);
- SDIO data 传输异常, 检查 sdio wifi 硬件使用的物料是否符合标准, 比如电容, 电阻有没有错接或者遗漏;
- 当出现通信出错时, 比如:

```
dwmmc_rockchip xxxx.dwmmc: All phases work, using default phase 0.
```

```
bcmsdh_sdmmc: Failed to Read word F1:@xxxx=ffffff, Err: 0xfffffac
```

有两种可能性:

- 请检查第一章的 IO 电源阈的设置, 供电和电源域设置不匹配;
- sdio 走线差, 跑不了高频, 可以降低频率来确认问题:

&sdio 节点的 dts 配置加上: max-frequency = <100000000>(SDIO3.0), max-frequency =

<20000000>(SDIO2.0), 如果可以正常打开请检查走线部分;

WiFi 正常识别的 log 中有如下类似打印:

mmcX: new ultra high speed SDR104 SDIO card at address 0001

或

mmcX: new high speed SDIO card at address 0001

6.2 WiFi 无法连接路由器或断线连接不稳定

1、请先确保如下两个进程是否有跑起来:

```
wpa_supplicant -B -i wlan0 -c /data/cfg/wpa_supplicant.conf
```

```
/sbin/dhpcd -f /etc/dhpcd.conf
```

2、wpa_cli scan 、wpa_scan_r 命令去扫描热点, 如果执行失败可以多次执行, 确认是否能扫描到 wifi: 正常的话会有如下信息: 在下面的信息中找下是否有 WiFi, 如果扫不到, 或者扫到的 wifi 跟你们手机或其他设备的数量差距很大, 或者信号强度很弱 (signal level), 请检测 WiFi 天线是否达标, 频偏是否符合要求等。

Selected interface 'wlan0'

bssid / frequency / signal level / flags / ssid

dc:ef:09:a7:77:52	5765	-33	[WPA2-PSK-CCMP][ESS]	NETGEAR75-5G
10:be:f5:1d:a3:76	5220	-53	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][ESS]	D-Link_DIR-880L_5GHz
d0:ee:07:1c:2d:18	5745	-54	[WPA-PSK-CCMP][WPA2-PSK-CCMP][ESS]	ROCKROOM_5G
a0:63:91:2e:16:fa	5765	-56	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][ESS]	hjk_5GEXT
30:fc:68:bb:09:bb	5745	-67	[WPA-PSK-CCMP][WPA2-PSK-CCMP][ESS]	TP-LINK_5G_09B9
64:09:80:0a:13:b1	5805	-59	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][ESS]	diaoizatian
74:7d:24:61:39:d0	5180	-48	[WPA-PSK-CCMP+TKIP][WPA2-PSK-CCMP+TKIP][ESS]	@PHICOMM_CE_5G

6.3 WiFi 其他问题

请提供 kernel dmesg 和 wpa_supplicant 的 log(方法: 在启动 wpa_supplicant 程序的地方加上 debug 选项, 如: wpa_supplicant -B -i wlan0 -c /data/cfg/wpa_supplicant.conf -f debug.txt //将 log 重定向到 debug.txt 文件里面)

6.4 BT 排查

主要有三点:

- dts 的里面 bt_reg_on 配置错误, 导致 bt 上电异常;
- buildroot 配置是否正确, 使用的 WiFi 型号是否和配置对应;
- uart 口配置是否跟硬件对应;

请参考第一章 BT DTS 配置和第二章节 BT 测试进行排查

6.5 RTL8723DS 特别注意点

对于 RTL8723DS COB 芯片 (realtek chip on board) 和模块 (欧智通/小瑞) PCM_OUT 管脚注意事项, 首先模块的 PIN25 管脚和 COB 的 PIN 22 管脚是 8723DS 内部复用管脚, 一个是作为 PCM_OUT 连接 RK3308 PCM_IN, 用作蓝牙 PCM 通话功能; 另外一个作为 WIFI IC LDO_SPS_SEL 功能(SPS 或者 LDO 模式选择), 当这个管脚为低电平表示 SPS 模式, 如果为高电平为 LDO 模式, 在跟 WIFI 模块上电的时候, LDO_SPS_SEL 会选择决定供给 WIFI IC 内部的 1.2V 是何种方式, 如果电平不对, 会导致 WIFI 无

法正常使用，比如无法扫描 SDIO 或者扫描不到 AP 问题。

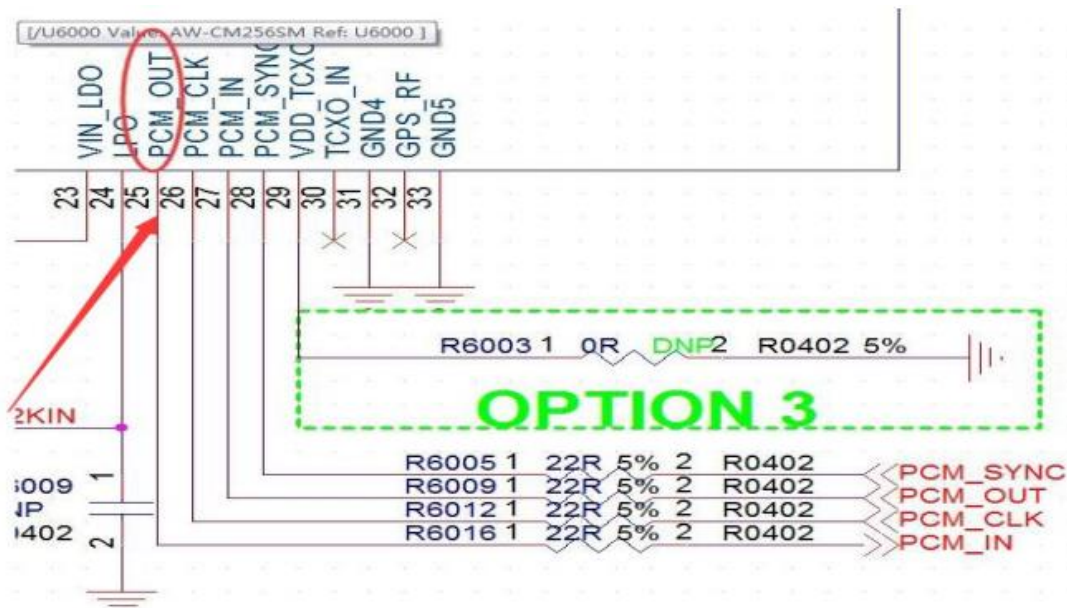
对于选择 COB 和 823DS 模块，PIN 22(realtek COB)或者 PIN25(欧智通/小瑞) 脚是连接到了 RK3308 内部下拉管脚(GPIO4_C0)



对于 COB 而言工作在 SPS 模式，低电平，但是对于模块，工作在 LDO 模式，连接到了这个 RK3308 内部下拉脚(GPIO4_C0)，就会出现半电平情况，导致 WIFI 异常，

解决办法：

使用模块的版型，硬件上将 WIFI 模块的 PIN25 脚接一个上拉电阻，直接拉高，工作在 LDO 模式，避免跟 RK3308 内部下拉脚（GPIO4_C0）对拉出现半电平，8723DS 模块 PCM_OUT PIN 如下图：



7 WIFIBT 应用开发

参考 docs\Develop reference documents\DeviceIo 目录下的文档：

7.1 WIFI 开发

Rockchip_Developer_Guide_DeviceIo_WIFI_CN.pdf

7.2 蓝牙开发

Rockchip_Developer_Guide_DeviceIo_Bluetooth_CN.pdf

7.3 配网开发（BLE/SOFTAP/AIRKISS）

Rockchip_Developer_Guide_Network_Config_CN.pdf

8 WIFIBT 模组移植

如果客户想用一些不在支持列表的其他模组，比如 AP/海华/REALTEK 的模组基本都可以支持；可以参考如下文档进行移植，docs\Develop reference documents\WIFIBT 目录下的：

《AP 模组移植到 RK 平台移植说明.txt》

《Realtek 移植到 RK 平台移植说明.txt》