# Rockchip Linux Network Config Documentation

发布版本：1.0

作者：CTF

日期：2019.5.13

文件密级：公开资料

**概述**

该文档旨在介绍Rockchip Linux各种配网方式。

**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

**对应DeviceIo库版本**

V1.2.1以上，不包含V1.2.1

**修订记录**

| 日期 | 版本 | 作者 | 修改说明 |
| --- | --- | --- | --- |
| 2019-4-29 | V1.0 | CTF | 初始版本 |
| 2019-5-13 | V1.0.1 | CTF | 修正手机配网各流程说明 |

# 1、WIFI/BT配置

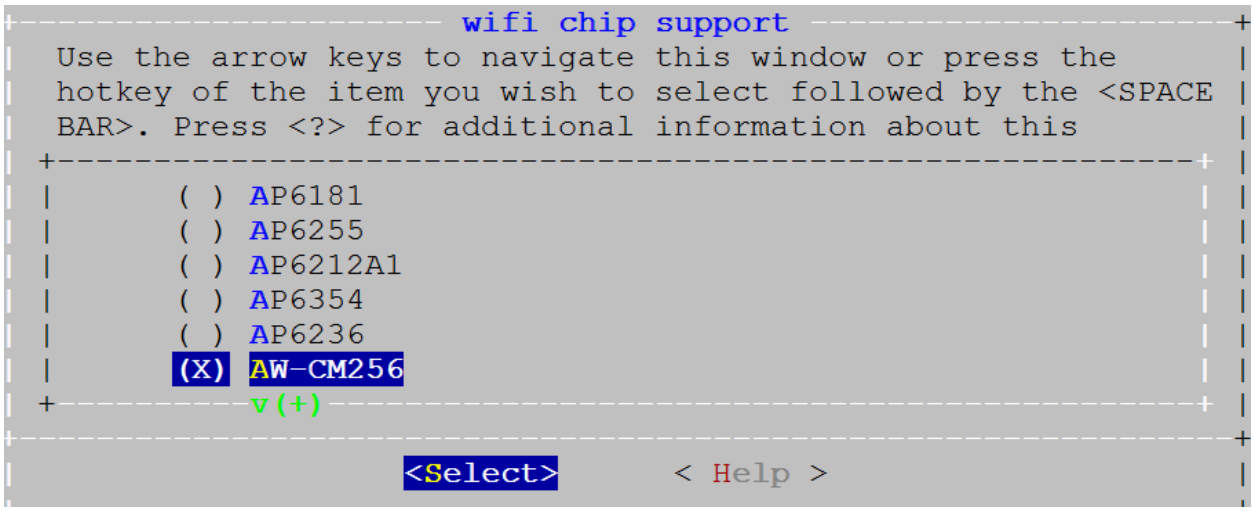
## 1.1 kernel配置

- 请参考 /docs/Linux reference documents 目录下的 Rockchip Linux WIFI BT 开发指南 V6.0.pdf 文档，第一章节'WIFI/BT 配置'

## 1.2 buildroot配置

- 根目录下执行`make menuconfig`
- rkwifibt配置，根据实际WiFi选择对应配置，必须跟kernel配置一致

```
Symbol: BR2_PACKAGE_RKWIFIBT [=y]
Type  : boolean
Prompt: rkwifibt
  Location:
    -> Target packages
(1)   -> rockchip BSP packages (BR2_PACKAGE_ROCKCHIP [=y])
  Defined at package/rockchip/rkwifibt/Config.in:1
  Depends on: BR2_PACKAGE_ROCKCHIP [=y]
```




- 蓝牙配置
  - realtek模组建议使用bluez 协议，正基/海华模组建议使用bsa 协议。
  - 以下配置，根据模组类型三选一
    - realtek模组选择：`bluez-utils 5.x`，使用bluez需要同时开启`bluez-alsa` `readline`

```
Symbol: BR2_PACKAGE_BLUEZ5_UTILS [=y]
Type  : boolean
Prompt: bluez-utils 5.x
  Location:
    -> Target packages
(2)   -> Networking applications
  Defined at package/bluez5_utils/Config.in:1
  Depends on: BR2_USE_WCHAR [=y] && BR2_TOOLCHAIN_HAS_THREADS [=y] && BR2_U
  Selects: BR2_PACKAGE_DBUS [=y] && BR2_PACKAGE_LIBGLIB2 [=y]
  Selected by: BR2_PACKAGE_BLUEZ_ALSA [=y] && !BR2_STATIC_LIBS [=n] && !BR2
```

```
    [ ] bcusdk
    [ ] bind
    [ ] bluez-tools
    [ ] bluez-utils
    -*- bluez-utils 5.x
    [ ]   build OBEX support
    [*]   build CLI client
    [ ]     install GATT tool
    [ ]   build experimental plugins
    [ ]   build sixaxis plugin
    [ ]   build tests
    [ ] bmon
```

```
Symbol: BR2_PACKAGE_BLUEZ_ALSA [=y]
Type  : boolean
Prompt: bluez-alsa
  Location:
    -> Target packages
(9)   -> Audio and video applications
  Defined at package/rockchip/bluez-alsa/Config.in:1
  Depends on: !BR2_STATIC_LIBS [=n] && !BR2_PACKAGE_BLUEZ_UTILS [=n] && BR2
  Selects: BR2_PACKAGE_ALSA_LIB [=y] && BR2_PACKAGE_BLUEZ5_UTILS [=y] && BR
```

```
    [*] alsa-utils  --->
    [*] alsa-plugins  ----
    [ ] atest
    [ ] aumix
    [ ] bellagio
    [*] bluez-alsa
    [*]   hcitop
    [ ] dvblast
    [ ] dvdauthor
    [ ] dvdrw-tools
    [ ] espeak
    -*- faad2
```

```
Symbol: BR2_PACKAGE_READLINE [=y]
Type  : boolean
Prompt: readline
  Location:
    -> Target packages
      -> Libraries
(7)     -> Text and terminal handling
  Defined at package/readline/Config.in:1
  Selects: BR2_PACKAGE_NCURSES [=y]
  Selected by: BR2_PACKAGE_BLE_WIFICONFIG [=n] && BR2_PACKAGE_ROCKCHIP [=y]
```

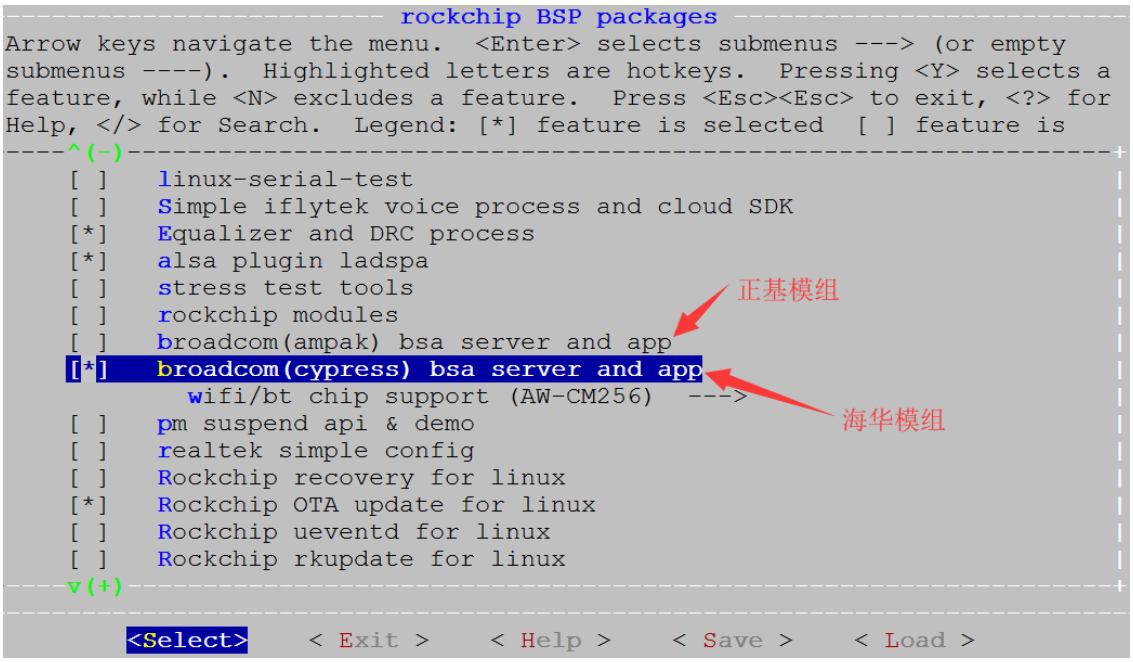
```
    -*-   UTF-8/16/32 support in pcre
    -*-   Unicode properties support in pcre
    [ ] pcre2
    -*- popt
    -*- readline
    [ ] slang
    [ ] tclap
    [ ] ustr
```

- 正基模组选择： `broadcom(ampak) bsa server and app`

  进入 `wifi/bt chip support(XXX)--->` 选择实际的芯片型号，必须跟rkwifibt配置一致

- 海华模组选择 ： `broadcom(cypress) bsa server and app`

  进入 `wifi/bt chip support(XXX)--->` 选择实际的芯片型号，必须跟rkwifibt配置一致




- 退出配置框，make savedefconfig保存配置

## 1.3 编译说明

- 根目录下执行： `make rkwifibt-dirclean && make rkwifibt-rebuild`
- 以下编译选项，根据模组类型三选一
  - realtek模组编译： `make bluez5_utils-rebuild`

    `make bluez-alsa-rebuild`
  - 正基模组编译： `make broadcom_bsa-rebuild`
  - 海华模组编译： `make cypress_bsa-rebuild`
- 根目录下执行： `make deviceio-dirclean && make deviceio-rebuild`
- 根目录下执行： `./build.sh`

## 2、命令行配网

- 首先确保WiFi的服务进程启动，串口输入：`ps | grep wpa_supplicant`

```
# ps | grep wpa_supplicant
  532 root       3380 S    wpa_supplicant -B -i wlan0 -c /data/cfg/wpa_supplica
  618 root       1836 R    grep wpa_supplicant
```

- 如果没启动，请手动启动：

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

- 修改`/data/cfg/wpa_supplicant.conf`文件，添加配置项

```
network={
        ssid="WiFi-AP"      // WiFi名字
        psk="12345678"      // WiFi密码
        key_mgmt=WPA-PSK    // 选填加密方式，不填的话可以自动识别
        # key_mgmt=NONE     // 不加密
}
```

- 重新读取上述配置：`wpa_cli reconfigure`
- 重新连接：`wpa_cli reconnect`

# 3、手机配网

## 3.1 ble 配网

- 简介

  ble配网同时支持bluez ble配网和bsa ble配网，配置参照本文档的第一章节'WIFI/BT 配置'。并且ble配网已集成到deviceio，接口位于RkBle.h。

- 接口说明

  请参考/docs/Develop reference documents/DeviceIo目录下Rockchip_Developer_Guide_Rk3308_DeviceIo_Bluetooth_CN.pdf文档，第二章节'BLE接口介绍（RkBle.h）'。

- 示例程序

  示例程序的路径为：`external/deviceio/test/rk_ble_app.c`

- APP

  app路径：`/external/app/RockHome.apk`

  app源码路径：`/external/app/src/RockHome`

- 配网步骤

  - 该配网步骤以bsa ble配网为例进行说明，所有板端log均为bsa的配网log。bluez操作步骤相同，板端log不同。
  - 首先确保WiFi的服务进程启动，串口输入：`ps | grep wpa_supplicant`

    ```
    # ps | grep wpa_supplicant
      532 root      3380 S    wpa_supplicant -B -i wlan0 -c /data/cfg/wpa_supplica
      618 root      1836 R    grep wpa_supplicant
    ```

  - 如果没启动，请手动启动：

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

  - 板端命令行执行：`deviceio_test wificonfig`，输入1回车， 启动ble 配网

    ```
    # deviceio_test wificonfig
    version:V1.2.1
    #### Please Input Your Test Command Index ####
    01.  ble_wifi_config_start
    02.  ble_wifi_config_stop
    03.  airkiss_wifi_config_start
    04.  airkiss_wifi_config_stop
    05.  softap_wifi_config_start
    06.  softap_wifi_config_stop
    07.  voiceprint_wifi_config_start
    08.  voiceprint_wifi_config_stop
    Which would you like: 1
    ===== rk_ble_wifi_init =====
    hcd_file = /system/etc/firmware/BCM4345C0.hcd
    killall: bsa_server: no process killed
    bsa_server died.
    [   24.786822] [BT_RFKILL]: ENABLE UART_RTS
    [   24.889285] [BT_RFKILL]: DISABLE UART_RTS
    [   24.889432] [BT_RFKILL]: bt turn on power
    start broadcom bluetooth server bsa_sever
    |----- bluetooth bsa server is open ------|
    [   25.072066] dw-apb-uart ff0e0000.serial: got rx and tx dma channels
    DEBUG: check_bsa_server: wait bsa_server open.
    DEBUG: check_bsa_server: bsa_server has been opened.
    ```

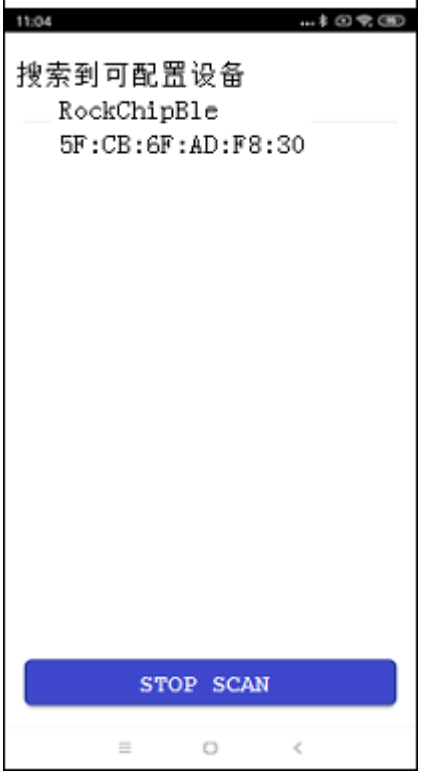
  - 设置的ble广播设备名必须以RockChip为前缀，否则apk无法检索到设备

```
DEBUG: app_ble_rk_server_open: app_ble_rk_server_open
[RK] ble status: RK_BLE_STATE_IDLE
INFO: app_ble_start: app_ble_start
BSA_trace 1029@ 01/01 09h:56m:09s:326ms: BSA_BleEnableInit
BSA_trace 1030@ 01/01 09h:56m:09s:326ms: BSA_BleEnable
DEBUG: app_ble_rk_server_set_device_name: app_ble_device_name: RockChipBle
INFO: app_ble_rk_server_gatt_server_init: wifi_introducer_gatt_server_init
BSA_trace 1031@ 01/01 09h:56m:09s:328ms: BSA_BleSeAppRegisterInit
BSA_trace 1032@ 01/01 09h:56m:09s:329ms: BSA_BleSeAppRegister
INFO: app_ble_rk_server_register: server_if:4
```

- 手机端打开apk

  点击CONTINUE -> START SCAN，扫描以RockChip为前缀命名的ble设备

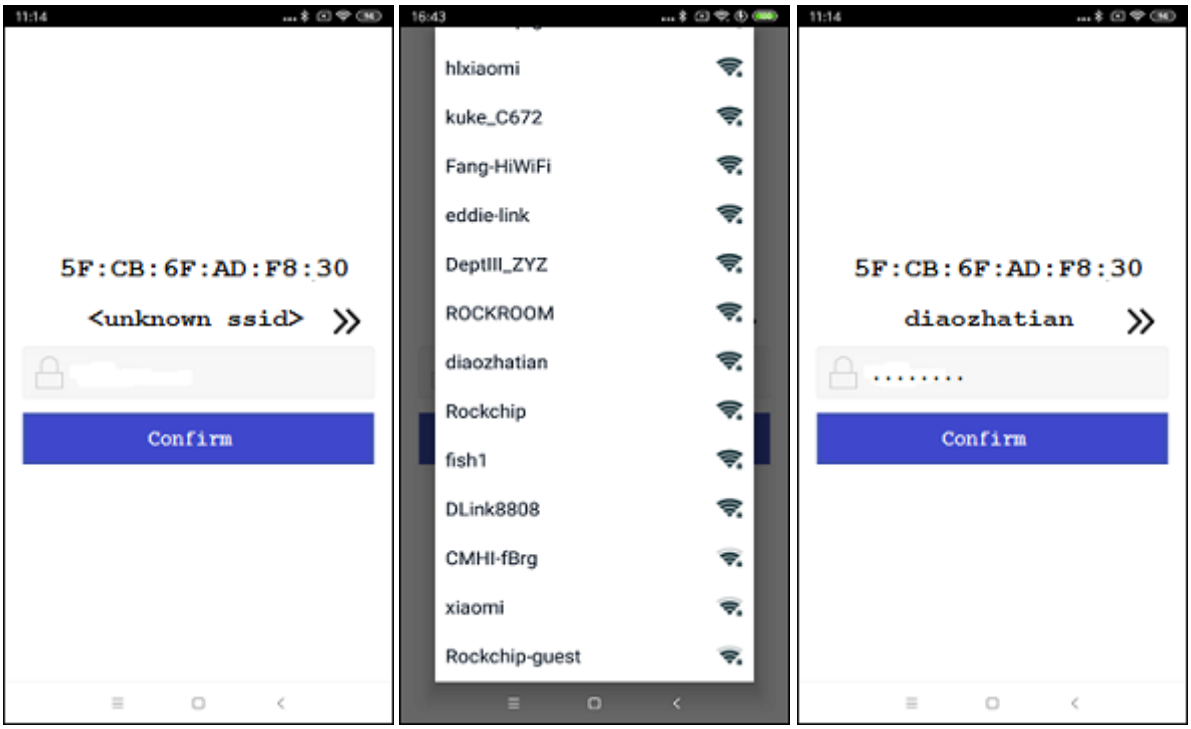




- 点击想要连接的ble设备，开始连接设备，设备连接成功，板端log如下

```
INFO: app_ble_rk_server_profile_cback: BSA_BLE_SE_OPEN_EVT status:0
INFO: app_ble_rk_server_profile_cback: app_ble_rk_server_conn_up conn_id:0x4
INFO: app_ble_rk_server_profile_cback: app_ble_rk_server_conn_up connected to [40:BD
:ED:F8:9A:1D]
DEBUG: app_dm_set_ble_visibility: Set BLE Visibility Discoverable:0 Connectable:0
BSA_trace 1049@ 01/01 09h:57m:56s:262ms: BSA_DmSetConfigInit
BSA_trace 1050@ 01/01 09h:57m:56s:263ms: BSA_DmSetConfig
[RK] ble status: RK_BLE_STATE_CONNECT
INFO: app_ble_rk_server_profile_cback: Stopping Advertisements
BSA_trace 1051@ 01/01 09h:57m:56s:267ms: bsa_sec_event_hdlr event:0
DEBUG: app_mgr_security_callback: event:0
DEBUG: app_mgr_security_callback: BSA_SEC_LINK_UP_EVT bd_addr: 40:bd:ed:f8:9a:1d
DEBUG: app_mgr_security_callback: ClassOfDevice:00:00:00 => Misc device
DEBUG: app_mgr_security_callback: LinkType: 2
DEBUG: bt_mgr_notify_callback: BT_LINK_UP_EVT
```

- 设备连接成功，apk进入配网界面，点击 >>按钮 获取wifi list，选择想要连接的wifi，输入密码，点击Confirm开始配网



- 板端接收到ssid和psk后，开始连接网络

```
[RK] ble_data.cmd: wifisetup, ble_data.start: 1, ble_data.end: 4
01-01 09:59:30.161   954   995 D [RK] wifi ssid is diaozhatian
01-01 09:59:30.162   954   995 D [RK] wifi psk is 7788123456
[RK] rk_config_wifi_thread
[RK] controlWifi connect ...
[RKWIFI] exec1: wpa_cli -iwlan0 disable_network all
[ 7170.184932] CFG80211-ERROR) wl_cfg80211_disconnect : Reason 3
[ 7170.191679] CFG80211-ERROR) wl_is_linkdown : Link down Reason : WLC_E_LINK
[ 7170.191800] link down if wlan0 may call cfg80211_disconnected. event : 16, reason
=2 from 64:09:80:0a:13:b0
[ 7170.216075] CFG80211-ERROR) wl_is_linkdown : Link down Reason : WLC_E_DEAUTH
[ 7170.219478] CFG80211-ERROR) wl_is_linkdown : Link down Reason : WLC_E_DEAUTH
[RKWIFI] exec1: wpa_cli -iwlan0 add_network
format_wifiinfo ssid: 6469616f7a68617469616e
[RKWIFI] exec1: wpa_cli -iwlan0 set_network 2 ssid 6469616f7a68617469616e
format_wifiinfo password: \7\7\8\8\1\2\3\4\5\6
[RKWIFI] exec1: wpa_cli -iwlan0 set_network 2 psk \"\7\7\8\8\1\2\3\4\5\6\"
01-01 09:59:31.301   954  3769 I RK_wifi_connect ssid:"diaozhatian" strlen(ssid):11;
 ori:"diaozhatian" strlen(ori):11; psk:"7788123456"
```

- 网络连接成功，板端发送通知给手机apk

```
wifi is connected.
OK
OK
[RK] rk_blewifi_state_callback state: 4
DEBUG: app_ble_rk_server_send_message: conn id : 0x4
INFO: app_ble_rk_server_send_message: Sending Notification
INFO: app_ble_rk_server_send_notification: app_ble_rk_server_send_notification
BSA_trace 1220@ 01/01 09h:59m:41s:219ms: BSA_BleSeSendIndInit
DEBUG: app_ble_rk_server_send_notification: uuid: 00009999-0000-1000-8000-00805F9B34
FB
DEBUG: app_ble_rk_server_send_notification: uuid_string: 0000180A-0000-1000-8000-008
05F9B34FB
DEBUG: app_ble_rk_server_send_notification: uuid_string: 00009999-0000-1000-8000-008
05F9B34FB
DEBUG: app_ble_rk_server_send_notification: attr_index_notify: 1
BSA_trace 1221@ 01/01 09h:59m:41s:222ms: send notification:
BSA_trace 1222@ 01/01 09h:59m:41s:223ms:     0000: 01
```

- apk端收到配网成功的通知后，断开ble连接，返回设备搜索界面，板端log如下

```
DEBUG: app_ble_rk_server_profile_cback: event = 23
INFO: app_ble_rk_server_profile_cback: BSA_BLE_SE_CLOSE_EVT status:19
INFO: app_ble_rk_server_profile_cback: conn_id:0x4
INFO: app_ble_rk_server_profile_cback: app_ble_rk_server_connection_down  conn_id:4
reason:19
DEBUG: app_dm_set_ble_adv_param: BDA:00:00:00:00:00:00
DEBUG: app_dm_set_ble_adv_param: adv_int_min:2056 adv_int_max:2056 inst_id:0
BSA_trace 224@ 01/01 08h:17m:48s:918ms: BSA_DmSetConfigInit
BSA_trace 225@ 01/01 08h:17m:48s:919ms: BSA_DmSetConfig
DEBUG: app_dm_set_ble_visibility: Set BLE Visibility Discoverable:1 Connectable:1
BSA_trace 226@ 01/01 08h:17m:48s:923ms: BSA_DmSetConfigInit
BSA_trace 227@ 01/01 08h:17m:48s:923ms: BSA_DmSetConfig
[RK] ble status: RK_BLE_STATE_DISCONNECT
BSA_trace 228@ 01/01 08h:17m:48s:928ms: bsa_sec_event_hdlr event:1
DEBUG: app_mgr_security_callback: event:1
DEBUG: app_mgr_security_callback: BSA_SEC_LINK_DOWN_EVT bd_addr: 51:59:51:a1:1d:03
DEBUG: app_mgr_security_callback: Reason: 19
DEBUG: app_mgr_security_callback: LinkType: 2
DEBUG: bt_mgr_notify_callback: BT_LINK_DOWN_EVT
```

- 再次启动配网，需要先输入2，关闭ble配网；再输入1重新启动ble，重复上述配网流程

## 3.2 airkiss 配网

- 简介

  目前airkiss配网只支持rtl8723ds，请参照本文档第一章节 'WIFI/BT 配置'进行相应配置；ap模组请参考external/wifiAutoSetup目录下的说明。

  airkiss兼容性很差，不建议客户使用，原因请参考 /docs/Develop reference documents/WIFIBT/RK平台_RTL8723DS_AIRKISS配网说明.pdf。目前airkiss配网已集成到deviceio中，接口位于Rk_wifi.h。

- kernel 修改

  修改 `/drivers/net/wireless/rockchip_wlan/rtl8723ds/Makefile` 文件

  ```
  -CONFIG_WIFI_MONITOR = n
  +CONFIG_WIFI_MONITOR = y
  ```

- 接口说明

  - 启动airkiss配网，成功返回0，失败返回-1

    `int RK_wifi_airkiss_start(char *ssid, char *password)`

    ssid：手机端发送的wifi名称

    password：手机端发送的wifi密码

  - 关闭airkiss配网

    `void RK_wifi_airkiss_stop()`

- 示例程序

  示例程序的路径为：`external/deviceio/test/rk_wifi_test.c`

  该测试用例调用`RK_wifi_airkiss_start()`启动airkiss，获取ssid和password并启动wifi配网。

  主要接口：`void rk_wifi_airkiss_start(void *data)`，在DeviceIOTest.cpp中调用。

```
void rk_wifi_airkiss_start(void *data)
{
    int err = 0;
    struct wifi_info info;
    pthread_t tid = 0;

    memset(&info, 0, sizeof(struct wifi_info));

    printf("===== %s =====\n", __func__);

    if(RK_wifi_airkiss_start(info.ssid, info.psk) < 0)
        return;

    err = pthread_create(&tid, NULL, rk_wifi_config_thread, &info);
    if (err) {
        printf("Error - pthread_create() return code: %d\n", err);
        return;
    }

    while (!wifi_state)
        sleep(1);
} « end rk_wifi_airkiss_start »
```

- 微信配网方式

  可以使用手机app 或者 扫描微信二维码的方式配置网络

  - 手机app

    下载地址：https://iot.weixin.qq.com/wiki/document-download.html ，进入下载中心 -> WiFi设备 -> airkiss 调试工具，下载AirKissDebugger.apk

  - 二维码

    微信扫描如下二维码，二维码配网时，手机必须先连接wifi，否则会提示：未能搜索设备，请开启手机wifi后重试

    微信扫描二维码配置网络

- 配网步骤

  - 首先确保WiFi的服务进程启动，串口输入：`ps | grep wpa_supplicant`

    ```
    # ps | grep wpa_supplicant
      532 root      3380 S    wpa_supplicant -B -i wlan0 -c /data/cfg/wpa_supplica
      618 root      1836 R    grep wpa_supplicant
    ```

  - 如果没启动，请手动启动：

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

- 手机端操作以app为例进行说明，打开AirKissDebugger.apk，输入ssid和password，AESKey为空、不输入。点击发送按钮，配网成功会弹窗提示“AirKissDebugger：Bingo”


- 板端命令行执行：`deviceio_test wificonfig`，输入3回车，启动airkiss 配网

```
# deviceio_test wificonfig
version:V1.2.1
#### Please Input Your Test Command Index ####
01.  ble_wifi_config_start
02.  ble_wifi_config_stop
03.  airkiss_wifi_config_start
04.  airkiss_wifi_config_stop
05.  softap_wifi_config_start
06.  softap_wifi_config_stop
07.  voiceprint_wifi_config_start
08.  voiceprint_wifi_config_stop
Which would you like: 3
===== rk_wifi_airkiss_start =====
```

- airkiss 启动成功

```
scan_ap_cnt: 42
use channel: 1 2 3 4 5 6 7 8 9 10 11 13
Start airkiss!
Airkiss init succeed!
```

- 成功接收ssid和password，并开始配网

```
AirKiss complete: ssid "diaozhatian", pwd "7788123456", random 0xa5
AIRKISS_STATUS_COMPLETE
airkiss_get_result() ok!
ssid = "diaozhatian", pwd = "7788123456", ssid_length = 11, "pwd_le
 = 0xa5
killall: wpa_supplicant: no process killed
```

- 配网成功

```
wpa_cli -iwlan0 status | grep wpa_state: wpa_state=COMPLETED

wpa_cli -iwlan0 status | grep ip_address: ip_address=192.168.31.164

Congratulation: wifi connected.
Selected interface 'wlan0'
OK
Selected interface 'wlan0'
OK
```

- 再次启动配网，需要先输入4，关闭airkiss配网；再输入3重新启动airkiss，重复上述配网流程

## 3.3 Softap 配网

- 简介

  首先，用SDK板的WiFi创建一个AP热点，在手机端连接该AP热点；其次，通过手机端apk获取SDK板的当前扫描到的热点列表，在手机端填入要连接AP的密码，apk会把AP的ssid和密码发到SDK板端；最后，SDK板端会根据收到的信息连接WiFi。

  Softap配网已集成到deviceio中，接口位于Rk_softap.h。

- APP

  app路径：`/external/app/RockHome.apk`

  app源码路径：`/external/app/src/RockHome`

- buildroot配置

```
Type  : boolean
Prompt: softap mode to setup wifi
  Location:
    -> Target packages
(1)   -> rockchip BSP packages (BR2_PACKAGE_ROCKCHIP [=y])
  Defined at package/rockchip/softap/Config.in:1
  Depends on: BR2_PACKAGE_ROCKCHIP [=y]
  Selected by: BR2_PACKAGE_SOFTAPSERVER [=y] && BR2_PACKAGE_ROCKCHIP [=y]


Symbol: BR2_PACKAGE_SOFTAPSERVER [=y]
Type  : boolean
Prompt: socket server based on softap
  Location:
    -> Target packages
(2)   -> rockchip BSP packages (BR2_PACKAGE_ROCKCHIP [=y])
  Defined at package/rockchip/softapServer/Config.in:1
  Depends on: BR2_PACKAGE_ROCKCHIP [=y]
  Selects: BR2_PACKAGE_SOFTAP [=y]
```

```
Symbol: BR2_PACKAGE_IW [=y]
Type  : boolean
Prompt: iw
  Location:
    -> Target packages
(2)   -> Networking applications
  Defined at package/iw/Config.in:1
  Depends on: BR2_TOOLCHAIN_HAS_THREADS [=y]
  Selects: BR2_PACKAGE_LIBNL [=y]
```

- 接口说明
  - 启动softap配网：

    `RK_softap_start(char* name, RK_SOFTAP_SERVER_TYPE server_type)`

    name：wifi热点的名字，前缀必须为Rockchip-SoftAp

    server_type：网络协议类型，目前只支持TCP协议
  - 结束softap配网

    `int RK_softap_stop(void)`
  - 注册状态回调

    `RK_softap_register_callback(RK_SOFTAP_STATE_CALLBACK cb)`

    正在连接网络：`RK_SOFTAP_STATE_CONNECTTING`

    网络连接成功：`RK_SOFTAP_STATE_SUCCESS`

    网络连接失败：`RK_SOFTAP_STATE_FAIL`
- 示例程序

  示例程序的路径为：`external/deviceio/test/rk_wifi_test.c`

  主要接口：`void rk_wifi_softap_start(void *data)`，`rk_wifi_softap_stop(void *data)`，在DeviceIOTest.cpp中调用。
- 配网步骤
  - 首先确保WiFi的服务进程启动，串口输入：`ps | grep wpa_supplicant`

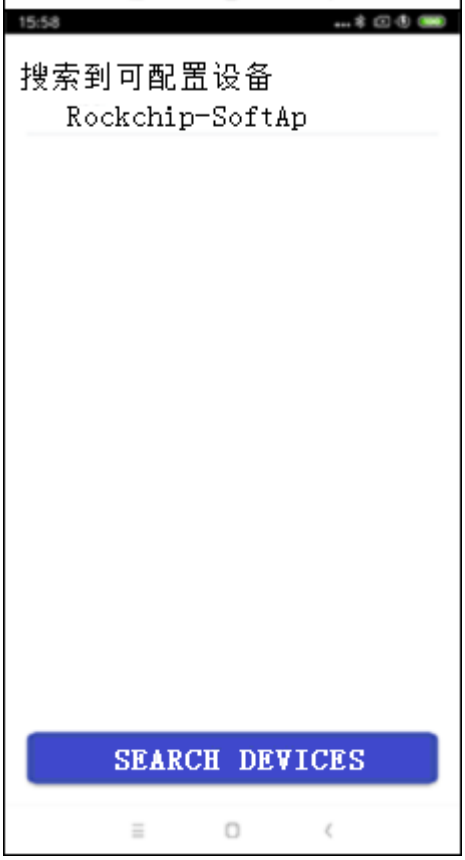
    ```
    # ps | grep wpa_supplicant
      532 root      3380 S    wpa_supplicant -B -i wlan0 -c /data/cfg/wpa_supplica
      618 root      1836 R    grep wpa_supplicant
    ```
  - 如果没启动，请手动启动：

    `wpa_supplicant -B -i wlan0 -c /data/cfg/wpa_supplicant.conf &`
  - 板端命令行执行`deviceio_test wificonfig`，输入5 回车，启动softap配网

    ```
    # deviceio_test wificonfig
    version:V1.2.1
    #### Please Input Your Test Command Index ####
    01.  ble_wifi_config_start
    02.  ble_wifi_config_stop
    03.  airkiss_wifi_config_start
    04.  airkiss_wifi_config_stop
    05.  softap_wifi_config_start
    06.  softap_wifi_config_stop
    07.  voiceprint_wifi_config_start
    08.  voiceprint_wifi_config_stop
    Which would you like: 5
    [ 4794.018629] Current WiFi chip is AP6255.
    Hostapd 143: wifi type: AP6255
    Hostapd 19: cmdline = killall dnsmasq
    Hostapd 19: cmdline = killall hostapd
    killall: hostapd: no process killed
    Hostapd 19: cmdline = ifconfig wlan1 down
    ifconfig: SIOCGIFFLAGS: No such device
    Hostapd 19: cmdline = rm -rf /userdata/bin/wlan1
    Hostapd 19: cmdline = iw dev wlan1 del
    command failed: No such device (-19)
    Hostapd 19: cmdline = ifconfig wlan0 up
    Hostapd 19: cmdline = iw phy0 interface add wlan1 type managed
    [ 4794.189854] CFG80211-ERROR) wl_cfg80211_event : ignore event 54, not interested
    [ 4794.191298] Register interface [wlan1]  MAC: 82:c5:f2:2e:e7:65
    [ 4794.191298]
    ```

- 打开RockHome.apk，左侧滑选择第三个选项，进入softap配网方式，点击 SEARCH DEVICES，扫描以Rockchip-SoftAp为前缀命名的softap设备

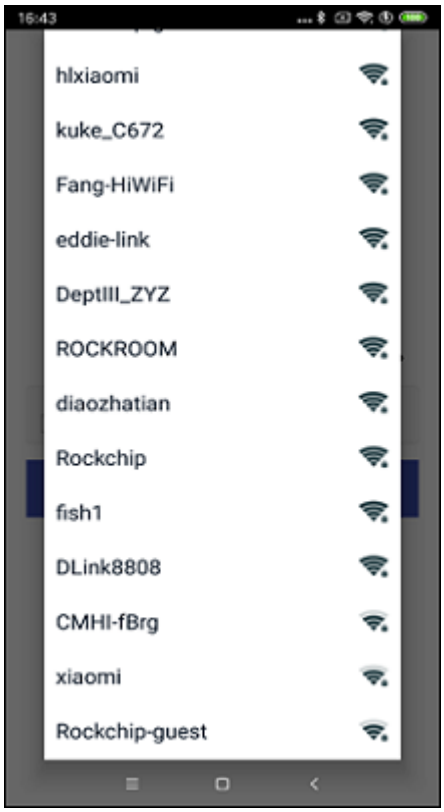




- 点击想要连接的softap设备，开始连接设备，设备连接成功，板端log如下

```
wlan1: STA 94:87:e0:34:e6:fd IEEE 802.11: associated
wlan1: AP-STA-CONNECTED 94:87:e0:34:e6:fd
[ 5955.601561] CFG80211-ERROR) wl_cfg80211_change_station : WLC_SCB_AUTHORIZE sta_fl
ags_mask not set
```

- 设备连接成功，apk进入配网界面，点击 >> 获取wifi list，选择想要连接的wifi，输入密码，点击Confirm开始配网


- 板子收到ssid和psk，开始连接网络

```
TcpServer recv buf:
POST /provision/wifiSetup HTTP/1.1
Content-Type: application/json
User-Agent: Dalvik/2.1.0 (Linux; U; Android 8.1.0; MI 6X MIUI/V10.2.2.0.ODCCNXM)
Host: 10.201.126.1:8443
Connection: Keep-Alive
Accept-Encoding: gzip
Content-Length: 41

{"ssid":"diaozhatian","pwd":"7788123456"}
do connect ssid:"diaozhatian", psk:"7788123456", isConnecting:0
RK_SOFTAP_STATE_CONNECTTING
```

- 网络连接成功

```
GET /provision/wifiState HTTP/1.1
Content-Type: application/json
User-Agent: Dalvik/2.1.0 (Linux; U; Android 8.1.0; MI 6X MIUI/V10.2.2.0.ODCCNXM)
Host: 10.201.126.1:8443
Connection: Keep-Alive
Accept-Encoding: gzip


[   64.288035] CFG80211-ERROR) wl_cfg80211_connect : Connecting with64:09:80:0a:13:b
0 ssid "diaozhatian", len (11) channel=4

[   64.613264] wl_bss_connect_done succeeded with 64:09:80:0a:13:b0
[   64.618258] CFG80211-ERROR) wl_cfg80211_determine_vsdb_mode : Same Channel concur
rency is enabled
[   64.696452] wl_bss_connect_done succeeded with 64:09:80:0a:13:b0
```

- 配网成功后，板端disableWifiAp，手机apk返回设备搜索界面，板端log如下

```
POST /provision/connectResult HTTP/1.1
Content-Type: application/json
User-Agent: Dalvik/2.1.0 (Linux; U; Android 8.1.0; MI 6X MIUI/V10.2.2.0.ODCCNXM)
Host: 10.201.126.1:8443
Connection: Keep-Alive
Accept-Encoding: gzip
Content-Length: 14

{"result":"1"}
RK_SOFTAP_STATE_SUCCESS
Hostapd 19: cmdline = killall hostapd
wlan1: interface state ENABLED->DISABLED
wlan1: AP-STA-DISCONNECTED 94:87:e0:34:e6:fd
[   67.201146] CFG80211-ERROR) wl_cfg80211_del_station : Disconnect STA : ff:ff:ff:f
f:ff:ff scb_val.val 3
[   67.201644] Current WiFi Hostapd 19: cmdline = killall dnschip is AP6255.
masq
[   67.205086] CFG80211-ERROR) wl_notify_connect_status_ap : event WLC_E_DEAUTH(5) s
tatus 0 reason 3
Hostapd 19: cmdline = ifconfig wlan1 down
wlan1: AP-DISABLED
nl80211: deinit ifname=wlan1 disabled_11b_rates=0
```

- 想要再次启动softap配网，需要先输入6，回车反初始化softap，再输入5重新初始化softap，重复上述配网步骤