

# Linux Network Configuration Introduction

---

ID: RK-KF-YF-378

Release Version: V1.0.2

Release Date: 2022-1-14

Security Level: ☐Top-Secret ☐Secret ☐Internal ☒Public

## DISCLAIMER

THIS DOCUMENT IS PROVIDED "AS IS". ROCKCHIP ELECTRONICS CO., LTD. ("ROCKCHIP") DOES NOT PROVIDE ANY WARRANTY OF ANY KIND, EXPRESSED, IMPLIED OR OTHERWISE, WITH RESPECT TO THE ACCURACY, RELIABILITY, COMPLETENESS, MERCHANTABILITY, FITNESS FOR ANY PARTICULAR PURPOSE OR NON-INFRINGEMENT OF ANY REPRESENTATION, INFORMATION AND CONTENT IN THIS DOCUMENT. THIS DOCUMENT IS FOR REFERENCE ONLY. THIS DOCUMENT MAY BE UPDATED OR CHANGED WITHOUT ANY NOTICE AT ANY TIME DUE TO THE UPGRADES OF THE PRODUCT OR ANY OTHER REASONS.

## Trademark Statement

"Rockchip", "瑞芯微", "瑞芯" shall be Rockchip's registered trademarks and owned by Rockchip. All the other trademarks or registered trademarks mentioned in this document shall be owned by their respective owners.

**All rights reserved. ©2022. Rockchip Electronics Co., Ltd.**

Beyond the scope of fair use, neither any entity nor individual shall extract, copy, or distribute this document in any form in whole or in part without the written approval of Rockchip.

Rockchip Electronics Co., Ltd.

No.18 Building, A District, No.89, software Boulevard Fuzhou, Fujian, PRC

Website: [www.rock-chips.com](http://www.rock-chips.com)

Customer service Tel: +86-4007-700-590

Customer service Fax: +86-591-83951833

Customer service e-Mail: [fae@rock-chips.com](mailto:fae@rock-chips.com)

## Preface

## Overview

This document mainly introduces the ways to config network on Rockchip Linux platform.

## Product Version

Chipset	Kernel Version
RK3308/RK3326/RK3288/RK3399/RK1808/RV1108	4.4

## Intended Audience

This document (this guide) is mainly intended for:

Technical support engineers

Software development engineers

## Revision History

Version	Author	Date	Change Description
V1.0.0	CTF/XY	2019-06-16	Initial version
V1.0.1	Ruby Zhang	2020-08-13	Update the format and company name
V1.0.2	Ruby Zhang	2022-01-14	Update contents of Chapter 1.1

## Contents

### Linux Network Configuration Introduction

1. Wi-Fi/BT Configuration
  - 1.1 Kernel Configuration
  - 1.2 Buildroot Configuration
  - 1.3 Compilation Introduction
2. Configuration Network by Command Line
3. Network Configuration by Mobile Phone
  - 3.1 BLE Network Configuration
    - 3.1.1 Introduction
    - 3.1.2 Interface Description
    - 3.1.3 Sample Program
    - 3.1.4 APP
    - 3.1.5 Network Configuration Steps
  - 3.2 AirKiss Network Configuration
    - 3.2.1 Introduction
    - 3.2.2 Kernel Modification
    - 3.2.3 Interface Description
    - 3.2.4 Sample Program
    - 3.2.5 WeChat Network Configuration
    - 3.2.6 Operation Example
  - 3.3 SoftAP Network Configuration
    - 3.3.1 Introduction
    - 3.3.2 APP
    - 3.3.3 Buildroot Configuration
    - 3.3.4 Interface Description
    - 3.3.5 Sample Program
    - 3.3.6 Network Configuration Steps
  - 3.4 Softap Web UI Configure Network
    - 3.4.1 Introduction
    - 3.4.2 Code Directory
    - 3.4.3 Buildroot Configuration
    - 3.4.4 Configure Network

# 1. Wi-Fi/BT Configuration

## 1.1 Kernel Configuration

Please refer to the the Second chapter "WIFI/BT Configuration" in the Rockchip\_Developer\_Guide\_Linux\_WIFI\_BT\_EN.pdf document under the /docs/Linux Wifibt directory.

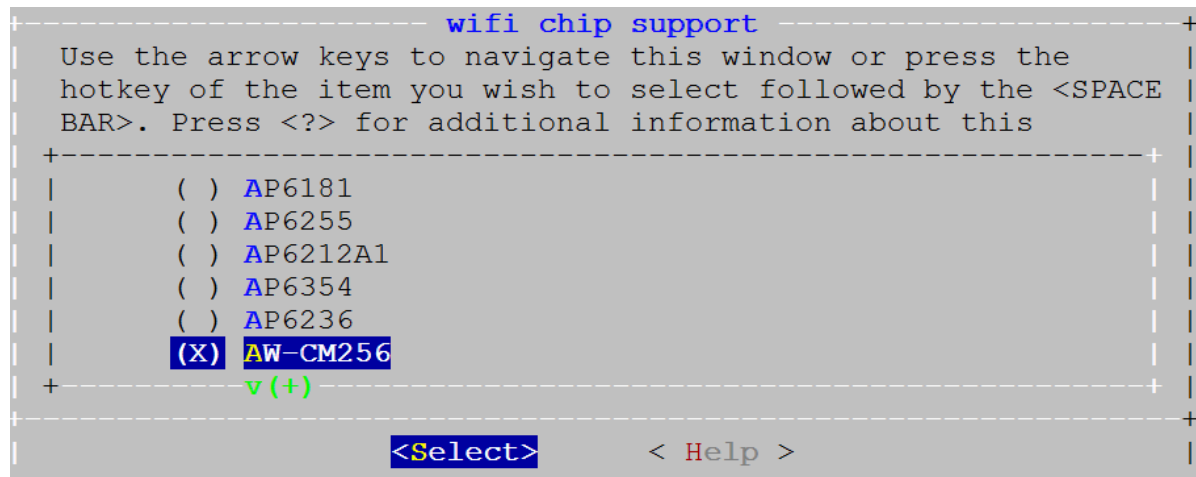
## 1.2 Buildroot Configuration

Execute in the root directory: `make menuconfig`.

### 1. Wi-Fi configuration:

Please select the corresponding configuration according to the actual Wi-Fi used, and it must be consistent with the kernel configuration.

```
Symbol: BR2_PACKAGE_RKWIFIBT [=y]
Type   : boolean
Prompt: rkwifi
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]
```



### 2. Bluetooth configuration

Realtek modules are recommended to use BlueZ protocol, and Ampak/AzureWave modules are recommended to use BSA protocol. Choose one of the following configurations according to the module type:

- For Realtek modules, please choose: bluez-utils 5.x, if BlueZ is used, bluez-alsa and readline are needed to be turned on at the same time

```

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

```

```

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

```

```

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

```

- For Ampak modules, please choose: broadcom(ampak) bsa server and app

Enter wifi/bt chip support(XXX)---> select the actual chip model, which must be consistent with the rkwifibt configuration.

- For AzureWave modules, please choose: broadcom(cypress) bsa server and app

Enter wifi/bt chip support(XXX)---> select the actual chip model, which must be consistent with the rkwifibt configuration.



3. Exit the configuration dialog box and save the configuration with `make savedefconfig`

## 1.3 Compilation Introduction

1. Compile rkwifibt and execute in the root directory:

```
make rkwifibt-dirclean && make rkwifibt-rebuild
```

2. Compile the Bluetooth module, choose one of the following three compilation options according to the module type

- Realtek module compilation:

```
make bluez5_utils-rebuild  
make bluez-alsa-rebuild
```

- Ampak module compilation:

```
make broadcom_bsa-rebuild
```

- AzureWave module compilation:

```
make cypress_bsa-rebuild
```

3. Compile deviceio, execute in the root directory:

```
make deviceio-dirclean && make deviceio-rebuild
```

4. Package the firmware and execute it in the root directory:

```
./mkfirmware.sh #Or ./build.sh, global compilation, will package the firmware  
automatically
```

## 2. Configuration Network by Command Line

1. First make sure that the Wi-Fi service process is enabled, and enter the serial port: `ps | grep wpa_supplicant`

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

2. If it does not start, please start it manually:

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

3. Modify the /data/cfg/wpa\_supplicant.conf file and add the following configuration items:

```
network={
    ssid="WiFi-AP" // Wi-Fi name
    psk="12345678" // Wi-Fi password
    key_mgmt=WPA-PSK // Encryption method(Optional), if it is not filled in, it
can be identified automatically
    #key_mgmt=NONE // No encryption
}
```

4. Re-read the above configuration: `wpa_cli reconfigure`

5. Rconnect: `wpa_cli reconnect`

## 3. Network Configuration by Mobile Phone

---

### 3.1 BLE Network Configuration

#### 3.1.1 Introduction

BLE network configuration supports both BlueZ BLE and BSA BLE. Please refer to the first chapter "WIFI/BT Configuration" of this document for details. And BLE network configuration has been integrated into deviceio, and the interface is located in RkBle.h.

#### 3.1.2 Interface Description

Please refer to the the second chapter "BLE Interface Introduction (RkBle.h)" in the "Rockchip\_Developer\_Guide\_Rk3308\_DeviceIo\_Bluetooth\_EN.pdf" document in the /docs/Develop reference documents/DeviceIo directory.

#### 3.1.3 Sample Program

The sample program is located in: `external/deviceio/test/rk_ble_app.c`.

### 3.1.4 APP

APP is located in: /external/app/RockHome.apk

APP source code is located in: /external/app/src/RockHome

This APP is only used as a development demo on the mobile phone side. We have adapted Honor 8, Remi 6, Xiaomi 6, OnePlus 6, OPPO A5 models, iPhone 6s (plus), Samsung S6, VIVO X9 and other mobile phones. Other models of mobile phones have not been tested, and there may be app compatibility risk.

### 3.1.5 Network Configuration Steps

The network configuration steps are explained by taking BSA BLE network configuration as an example, and all logs on the board side are logs of BSA configuration network. The BlueZ operation steps are the same, but the log on the board side is different.

1. Make sure that the Wi-Fi service process is started, and enter from the serial port: `ps | grep wpa_supplicant`

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

2. If it does not start, please start it manually:

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

3. On the board side, command line execute " `deviceio_test wificonfig` ", enter 1 and press Enter to start BLE network configuration

```
# 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.
```

4. The name of the "set BLE broadcast device" must be prefixed with **RockChip**, otherwise the APK cannot search the device:



```

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

```

5. Open the APK on the mobile phone

Click CONTINUE -> START SCAN to scan for BLE devices named with RockChip as the prefix:



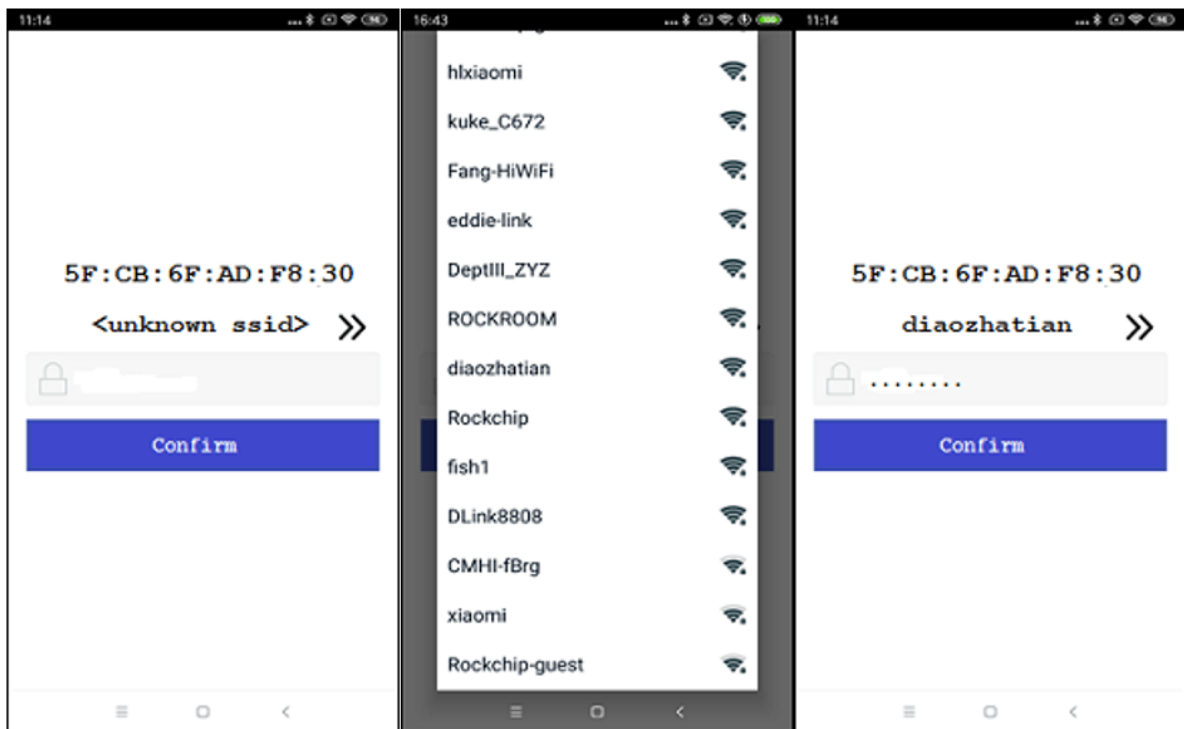
6. Click on the BLE device you want to connect to and start to connect to the device. If the device is connected successfully, the log on the board side will be showed as follows

```

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

```

7. When the device is connected successfully, enters the network configuration interface on the APK, click the >> button to get the Wi-Fi list, select the Wi-Fi you want to connect to, enter the password, and click Confirm to start the network configuration:



8. After the board receives the ssid and psk, it starts to connect to the network

```
[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"
```

9. If the connection is successful, the board will send a notification to the phone 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: 00009999-0000-1000-8000-00805F9B34FB
DEBUG: app_ble_rk_server_send_notification: uuid_string: 0000180A-0000-1000-8000-00805F9B34FB
DEBUG: app_ble_rk_server_send_notification: uuid_string: 00009999-0000-1000-8000-00805F9B34FB
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
```

10. After the APK receives the notification with successful network configuration, it disconnects the BLE connection and returns to the device search interface. The log on the board side is as follows

```

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

```

11. To start the network configuration again, you need to enter 2 to turn off the BLE network configuration; then enter 1 to restart BLE, and repeat the above network configuration process.

## 3.2 AirKiss Network Configuration

### 3.2.1 Introduction

At present, AirKiss only supports rtl8723ds, please refer to the first chapter of this document 'Wi-Fi/BT Configuration' for detailed configuration; for AP modules, please refer to the introduction in the external/wifiAutoSetup directory.

AirKiss has poor compatibility and it is not recommended to be used as the only one network configuration solution, and other network configuration solution need to be added. Please refer to "/docs/Develop reference documents/WIFIBT/RK platform RTL8723DS AIRKISS network configuration introduction.pdf" for detailed introduction.

At present, AirKiss network configuration has been integrated into deviceio, and the interface is located in Rk\_wifi.h.

### 3.2.2 Kernel Modification

Modify the /drivers/net/wireless/rockchip\_wlan/rtl8723ds/Makefile file:

```

-CONFIG_WIFI_MONITOR = n
+CONFIG_WIFI_MONITOR = y

```

### 3.2.3 Interface Description

Start AirKiss network configuration, return 0 if it succeeds, return -1 if it fails:

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

- ssid: the name of Wi-Fi sent from the mobile phone
- password: the password of the Wi-Fi sent by the mobile phone

Close AirKiss network configuration:

```
void RK_wifi_airkiss_stop()
```

### 3.2.4 Sample Program

The sample program is located in: external/deviceio/test/rk\_wifi\_test.c

This test case calls RK\_wifi\_airkiss\_start() to start AirKiss, obtain ssid and password, and start Wi-Fi network configuration. Main interface : void rk\_wifi\_airkiss\_start(void \*data), called in DeviceIOTest.cpp.

```
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
```

### 3.2.5 WeChat Network Configuration

You can use the mobile APP or scan the WeChat QR code to configure the network.

1. Mobile APP download address: <https://iot.weixin.qq.com/wiki/document-download.html>, enter the download center -> WiFi device -> AirKiss debugging tool, download AirKissDebugger.apk

#### WiFi设备

AirKiss技术简介 : 下载

AirKiss调试工具 : 下载

AirLink调试工具 : 下载

2. Scan the following QR code on WeChat. When configuring the network with the QR code, the phone must be connected to Wi-Fi first, otherwise it will prompt: failed to search for the device, please turn on the phone's Wi-Fi and try again



微信扫描二维码配置网络

### 3.2.6 Operation Example

1. Take the APP as an example to explain the operation on the mobile phone. Open AirKissDebugger.apk, enter the ssid and password, and the AESKey can leave empty. Click the send button, will prompt "AirKissDebugger: Bingo" when the network configuration is successful



2. Command line execute: `deviceio_test wificonfig` on the board side , enter 3 and press Enter to start airkiss network configuration

```
# 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 =====
```

3. AirKiss started successfully

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

4. Received ssid and password successfully, and started to configure network

```
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
```

5. Network configuration is successful

```
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
```

6. To start the network configuration again, you need to enter 4 to close the AirKiss network configuration; then enter 3 to restart AirKiss and repeat the above network configuration process.

## 3.3 SoftAP Network Configuration

### 3.3.1 Introduction

First, use the Wi-Fi of the SDK board to create an AP hotspot and connect to the AP hotspot on the mobile phone; second, obtain the currently scanned hotspot list of the SDK board through the mobile phone APK, and fill in the password to connect to the AP on the mobile phone. The APK will send the AP's ssid and password to the SDK board; finally, the SDK board will connect to Wi-Fi according to the received information.

SoftAP configuration network has been integrated into deviceio, and the interface is located in Rk\_softap.h.

### 3.3.2 APP

APP is located in: /external/app/RockHome.apk

App source code is located in: /external/app/src/RockHome

### 3.3.3 Buildroot Configuration

```

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]

```

### 3.3.4 Interface Description

1. Start softap network configuration:

```
RK_softap_start(char* name, RK_SOFTAP_SERVER_TYPE server_type)
```

- name: the name of the Wi-Fi hotspot, the prefix must be Rockchip-SoftAp
- server\_type: network protocol type, currently, only supports TCP protocol

2. Stop softap network configuration

```
int RK_softap_stop(void)
```

3. Registration status callback

```
RK_softap_register_callback(RK_SOFTAP_STATE_CALLBACK cb)
```

- Connecting to the network: RK\_SOFTAP\_STATE\_CONNECTTING
- The network connection is successful: RK\_SOFTAP\_STATE\_SUCCESS
- Network connection failed: RK\_SOFTAP\_STATE\_FAIL

### 3.3.5 Sample Program

The path of the sample program is: external/deviceio/test/rk\_wifi\_test.c

Main interface:

```
void rk_wifi_softap_start(void *data)
rk_wifi_softap_stop(void *data)
```

Called in DeviceIOTest.cpp.

### 3.3.6 Network Configuration Steps

1. First, make sure that the Wi-Fi service process is started. Enter on the serial port: `ps | grep wpa_supplicant`, if it doesn't start, please start it manually:

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

2. Execute `deviceio_test wificonfig` in the command line of the board, enter 5 and press Enter to start SoftAP network configuration

```
# 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]
```

3. Open RockHome.apk, slide left to select the third option, enter the SoftAP network configuration mode, click SEARCH DEVICES, scan for SoftAP devices named with Rockchip-SoftAp as the prefix

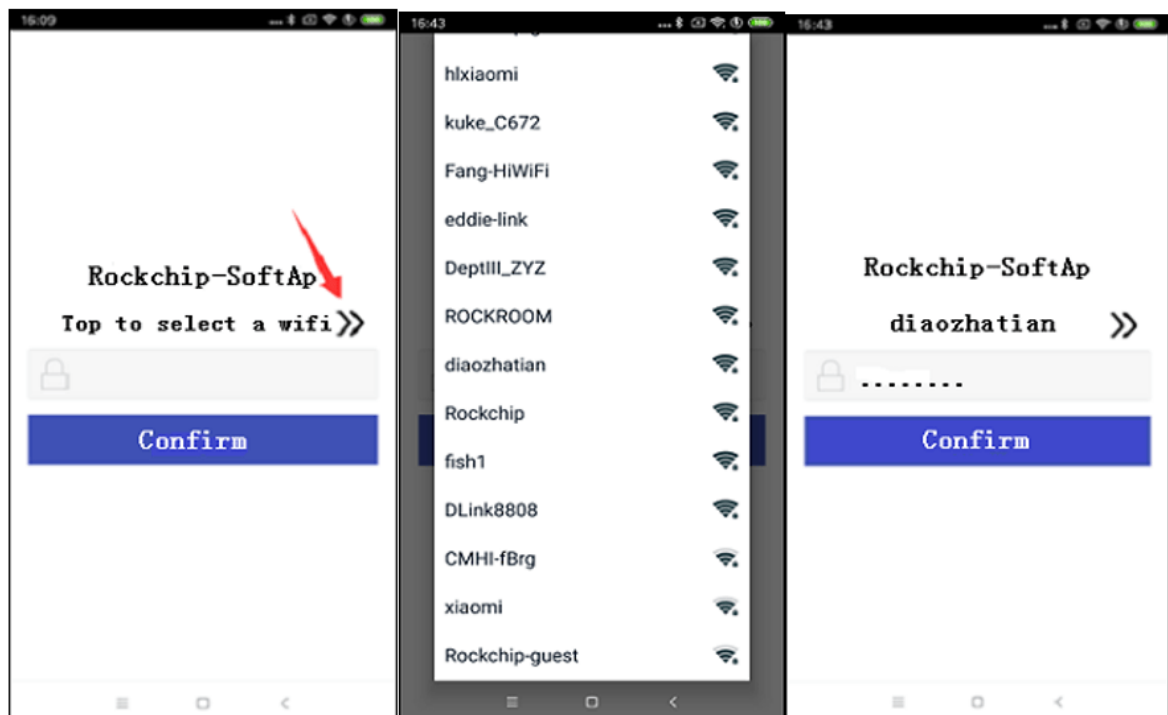




4. Click the SoftAP device you want to connect to, start to connect to the device, when the device is connected successfully, and there will be the following log on the board side:

```
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_flags_mask not set
```

5. When the device is successfully connected, enters the network configuration interface on the APK, click >> to get the Wi-Fi list, select the Wi-Fi you want to connect to, enter the password, and click Confirm to start the network configuration



6. The board receives the ssid and psk and starts to connect to the network

```
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_CONNECTING
```

7. The network connection is successful:

```
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 with 64:09:80:0a:13:b0
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 concurrency is enabled
[ 64.696452] wl_bss_connect_done succeeded with 64:09:80:0a:13:b0
```

8. After the network configuration is successful, disable WifiAp on the board side, the mobile phone APK returns to the device search interface, and the log on the board side is as follows:

```
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:ff:ff:ff scb_val.val 3
[ 67.201644] Current WiFi Hostapd 19: cmdline = killall dnsmasq
masq
[ 67.205086] CFG80211-ERROR) wl_notify_connect_status_ap : event WLC_E_DEAUTH(5) status 0 reason 3
Hostapd 19: cmdline = ifconfig wlan1 down
wlan1: AP-DISABLED
nl80211: deinit ifname=wlan1 disabled 11b_rates=0
```

9. If you want to start SoftAP network configuration again, you need to enter 6, press Enter to de-initialize SoftAP, and then enter 5 to re-initialize SoftAP, and repeat the above network configuration steps.

## 3.4 Softap Web UI Configure Network

### 3.4.1 Introduction

The rule of the Softap Web UI network configuration is the same as the above SoftAP network configuration, except that there is no need to install any APK on the mobile phone, connect to the hotspot, and then configure the network in the browser directly.

### 3.4.2 Code Directory

external/rk\_webui/ (contains source code, startup script)

buildroot/package/boa/

buildroot/package/rockchip/rk\_webui/ (including compilation script)

### 3.4.3 Buildroot Configuration

First, Buildroot selects `BR2_PACKAGE_RK_WEBUI = y`, then saves the configuration and recompiles `make rk_webui` to regenerate the new firmware.

```
There is no help available for this option.
Symbol: BR2_PACKAGE_RK_WEBUI [=y]
Type   : boolean
Prompt: Rockchip web ui
Location:
  -> Target packages
  -> rockchip BSP packages (BR2_PACKAGE_ROCKCHIP [=y])
Defined at package/rockchip/rk_webui/Config.in:1
Depends on: BR2_PACKAGE_ROCKCHIP [=y]
```

### 3.4.4 Configure Network

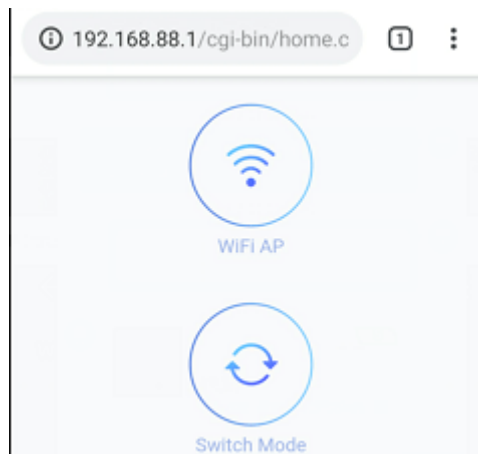
1. After startup regularly, execute `ps` to ensure that the following 4 processes are started:

```
394 root      3380 S      wpa_supplicant -B -i wlan0 -c /data/cfg/wpa_supplika
420 root      2004 S      dnsmasq -C /userdata/bin/dnsmasq.conf --interface=p2
422 root      3728 S      hostapd /userdata/bin/hostapd.conf
427 root      1532 S      boa
```

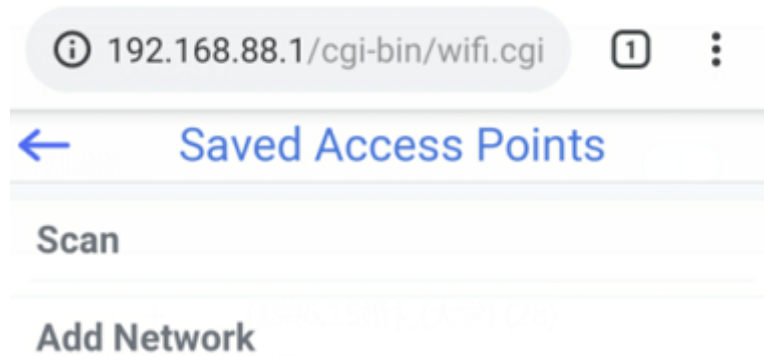
2. Open the setting interface of the phone and search for APs with `Rockchip_WebUI_` prefix, such as `Rockchip_WebUI_9604` (the last 4 number represents the last 4 number of the Wi-Fi MAC address of the device, which is convenient to distinguish), and click connect:



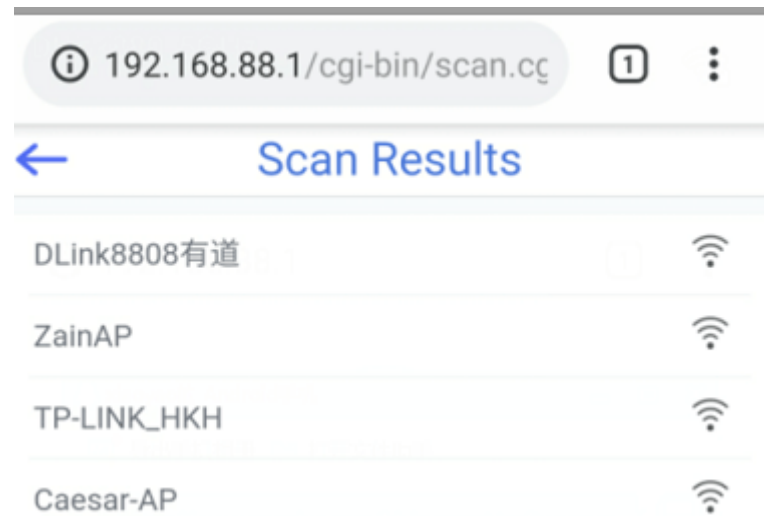
3. Open the mobile browser, enter: `192.168.88.1` (the browser will jump to `/cgi-bin/home.c` automatically), then press Enter, the following interface appears:



4. Click WiFi AP:



5. Click Scan:



6. Click the Wi-Fi you want to connect, then enter the password and click Connect (Note: Due to the hardware limitation of the Wi-Fi chip: When the current Wi-Fi connection such as TP-LINK\_HKH and its own hotspot Rockchip\_WebUI\_XXXX are not in the same channel, it will cause the phone disconnected from the hotspot, please reconnect to the hotspot to obtain the network configuration status)

192.168.88.1/cgi-bin/dialog.c

TP-LINK\_HKH

**BSSID:**  
9c:21:6a:c8:6f:7c

**Frequency:**  
2462Mhz

**Security:**  
[WPA-PSK-CCMP][WPA2-PSK-CCMP]  
[ESS]

Password:

ASCII ☒ HEX ☐

Connect

注意: 当连接目标WiFi和热点不在相同信道时会导致热点断开, 请重新在setting界面连接热点, 以获取配网状态。

7. Reconnect the phone to the hotspot, click refresh, you can see that it is connected (and supports forget and disconnect operations)

192.168.88.1/cgi-bin/wifi.cgi

Saved Access Points

Scan

TP-LINK_HKH	Connected	Forget	Disconnect
-------------	-----------	--------	------------

Add Network

注意: 当连接目标WiFi和热点不在相同信道时会导致热点断开, 请重新在setting界面连接热点, 以获取配网状态。