

RK3308 WIFI Interface Introduction

ID: RK-KF-YF-338

Release Version: V1.0.1

Release Date: 2020-02-28

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

DISCLAIMER

THIS DOCUMENT IS PROVIDED "AS IS". FUZHOU 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. ©2019. Fuzhou 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.

Fuzhou Rockchip Electronics Co., Ltd.

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

Website: 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

Preface

Preface

This document mainly introduce the interfaces in the RK3308 DeviceIo library

Chipset

RK3308

Intended Audience

This document (this guide) is mainly intended for:

Technical support engineers Software development engineers

Revision History

Date	Version	Author	Revision History
2019-3-29	V1.0.0	Jacky Ge	Initial version
2020-02-28	V1.0.1	Ruby Zhang	Update the format and the name of document

Contents

RK3308 WIFI Interface Introduction

- Preface
 - Contents
 - 1. Overview
 - 2. Interface Introduction
 - 3. Application Demo
-

1. Overview

This code module is integrated in the libDeviceIo.so dynamic library and it is a WIFI operation interface based on WPA package.

2. Interface Introduction

- `RK_WIFI_RUNNING_State_e`

About several status definitions of WIFI :

```
1  typedef enum {
2      RK_WIFI_State_IDLE = 0,
3      RK_WIFI_State_CONNECTING,
4      RK_WIFI_State_CONNECTFAILED,
5      RK_WIFI_State_CONNECTFAILED_WRONG_KEY,
6      RK_WIFI_State_CONNECTED,
7      RK_WIFI_State_DISCONNECTED
8  } RK_WIFI_RUNNING_State_e;
```

- `RK_WIFI_CONNECTION_Encryp_e`

There are three WIFI encryption types: passwordless, WPA and WEP.

```
1  typedef enum {
2      NONE = 0,
3      WPA,
4      WEP
5  } RK_WIFI_CONNECTION_Encryp_e;
```

- `RK_WIFI_INFO_Connection_s`

Please refer to `wpa_cli -iwlan0 status` for WIFI status information.

```
1  typedef struct {
2      int id;
3      char bssid[20];
4      char ssid[64];
5      int freq;
6      char mode[20];
7      char wpa_state[20];
8      char ip_address[20];
9      char mac_address[20];
10 } RK_WIFI_INFO_Connection_s;
```

- `int RK_wifi_register_callback(RK_wifi_state_callback cb)`

To register the WIFI status callback interface, and callback when WIFI status changes.

- `int RK_wifi_ble_register_callback(RK_wifi_state_callback cb)`

The BLE WIFI callback interface, which is used for callback status during BLE network configuration.

- `int RK_wifi_running_getState(RK_WIFI_RUNNING_State_e* pState)`

To get current WIFI status and return 0 if successful.

- `int RK_wifi_running_getConnectionInfo(RK_WIFI_INFO_Connection_s* pInfo)`

To get current WIFI connection information.

- `int RK_wifi_enable_ap(const char* ssid, const char* psk, const char* ip)`

To enable softAp based on the value of ssid, psk and ip.

- `int RK_wifi_disable_ap()`

Close softAp.

- `int RK_wifi_scan(void)`

Please refer to `wpa_cli -iwlan0 scan` and execute WIFI scan operation .

- `char* RK_wifi_scan_r(void)`

Please refer to `wpa_cli -iwlan0 scan_r`, and get WIFI scan result, and return JSON.

- `char* RK_wifi_scan_r_sec(const unsigned int cols)`

Please refer to `RK_wifi_scan_r(void)`, and get specified column from WIFI scan result, and return JSON.

bssid / frequency / signal level / flags / ssid

To use 5 binary numbers to represent the above data in order from left to right. For example,

`RK_wifi_scan_r_sec (0x01)` is used to get bssid data, `RK_wifi_scan_r_sec (0x10)` is used to get ssid data, and `RK_wifi_scan_r_sec (0x1F)` is used to get all data.

- `int RK_wifi_connect(const char* ssid, const char* psk)`

Connect the specified hotspot by the default WPA encryption method.

- `int RK_wifi_connect1(const char* ssid, const char* psk, const RK_WIFI_CONNECTION_Encryp_e encryp, const int hide)`

Please see `RK_wifi_connect` interface for expanding encryption type, ssid hidden parameter.

- `int RK_wifi_disconnect_network(void)`

To disconnect WIFI connection.

- `int RK_wifi_set_hostname(const char* name)`

To configure hostname.

- `int RK_wifi_get_hostname(char* name, int len)`

To get hostname.

- `int RK_wifi_get_mac(char *wifi_mac)`

To get mac address.

- `int RK_wifi_has_config(void)`

To check whether the network been configured.

- `int RK_wifi_ping(void)`

To check whether the network is connected by ping.

3. Application Demo

```
1  #include <stdio.h>
2  #include <string.h>
3  #include <DeviceIo/Rk_wifi.h>
4
5  int _RK_wifi_state_callback(RK_WIFI_RUNNING_State_e state)
6  {
7      printf("_RK_wifi_state_callback state:%d\n", state);
8      return 0;
9  }
10
11 int main(int argc, char **argv)
12 {
13     // To register WIFI status callback
14     RK_wifi_register_callback(_RK_wifi_state_callback);
15
16     // To get printing after setting hostname
17     char hostname[16];
18     RK_wifi_set_hostname("RKWIFI");
19     memset(hostname, 0, sizeof(hostname));
20     RK_wifi_get_hostname(hostname, sizeof(hostname));
21     printf("hostname:%s\n", hostname);
22
23     // To get MAC address and print
24     char mac[32];
25     memset(mac, 0, sizeof(mac));
26     RK_wifi_get_mac(mac);
27     printf("mac:%s\n", mac);
28
29     // If you have already configured WIFI, enable the WIFI will
    automatically connect to the configured WIFI
30     // Otherwise connect to the specified WIFI
31     if (RK_wifi_has_config()) {
32         RK_wifi_enable(1);
33     } else {
34         RK_wifi_enable(1);
35         RK_wifi_connect("TP-LINK_C734BC", "12345678");
36     }
37
38     for (;;)
39     // Disconnect WIFI and turn off the WIFI module
40     RK_wifi_enable(0);
41
42     return 0;
43 }
```