

RK3308 WIFI 接口介绍

文件标识: RK-KF-YF-338

发布版本: V1.0.1

日期: 2020-02-28

文件密级: ☐绝密 ☐秘密 ☐内部资料 ☒公开

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 © 2019 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: +86-4007-700-590

客户服务传真: +86-591-83951833

客户服务邮箱: fae@rock-chips.com

前言

概述

该文档旨在介绍RK3308 DeviceIo库中接口。

芯片名称

RK3308

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
2019-3-29	V1.0.0	Jacky.Ge	初始版本
2020-02-28	V1.0.1	Ruby.Zhang	调整文档格式，更新文档名称

目录

RK3308 WIFI 接口介绍

前言

目录

1、概述

2、接口说明

3、使用示例

1、概述

该代码模块集成在libDeviceIo.so动态库里面，基于wpa封装的WIFI操作接口。

2、接口说明

- `RK_WIFI_RUNNING_State_e`

WIFI的几种状态定义

```
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`

WIFI加密类型，包括无密码、WPA和WEP三种方式

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

- `RK_WIFI_INFO_Connection_s`

WIFI状态信息，参考wpa_cli -iwlan0 status

```
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)`

注册WIFI状态回调接口，在WIFI状态改变是回调

- `int RK_wifi_ble_register_callback(RK_wifi_state_callback cb)`

ble wifi回调接口，用于ble配网时回调状态

- `int RK_wifi_running_getState(RK_WIFI_RUNNING_State_e* pState)`

获取当前WIFI状态，成功返回0

- `int RK_wifi_running_getConnectionInfo(RK_WIFI_INFO_Connection_s* pInfo)`

获取当前WIFI连接信息

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

根据传入的ssid、psk和ip开启softAp

- `int RK_wifi_disable_ap()`

关闭softAp

- `int RK_wifi_scan(void)`

执行WIFI scan操作, 参见wpa_cli -iwlan0 scan

- `char* RK_wifi_scan_r(void)`

获取WIFI scan结果, 返回JSON。参见wpa_cli -iwlan0 scan_r

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

获取WIFI scan结果指定列, 返回JSON。参见RK_wifi_scan_r(void)

bssid / frequency / signal level / flags / ssid

使用5位二进制从左到右依次代表上述数据, 例如RK_wifi_scan_r_sec(0x01)获取bssid数据, RK_wifi_scan_r_sec(0x10) 获取ssid数据, RK_wifi_scan_r_sec(0x1F)获取所有数据

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

以默认WPA加密方式连接指定热点

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

参见RK_wifi_connect接口, 拓展加密类型, ssid隐藏性参数

- `int RK_wifi_disconnect_network(void)`

断开WIFI连接

- `int RK_wifi_set_hostname(const char* name)`

设置hostname

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

获取hostname

- `int RK_wifi_get_mac(char *wifi_mac)`

获取mac地址

- `int RK_wifi_has_config(void)`

网络是否配置过

- `int RK_wifi_ping(void)`

以ping的方式判断网络是否连接

3、使用示例

```
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     // 注册WIFI状态回调
14     RK_wifi_register_callback(_RK_wifi_state_callback);
15
16     // 设置hostname后获取打印
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     // 获取MAC地址并打印
24     char mac[32];
25     memset(mac, 0, sizeof(mac));
26     RK_wifi_get_mac(mac);
27     printf("mac:%s\n", mac);
28
29     // 如果有配置过WIFI, enable wifi自动连接到配置的WIFI
30     // 否则连接到指定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     // 断开WIFI并关闭WIFI模块
40     RK_wifi_enable(0);
41
42     return 0;
43 }
44
```