

# RockChip Devicelo WIFI Interface Documentation

发布版本：1.0

作者：Jacky.Ge

日期：2019.3.29

文件密级：公开资料

### 概述

该文档旨在介绍RockChip Devicelo库中WiFi相关接口。

### 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	Jacky.Ge	初始版本

### RockChip Devicelo WIFI Interface Documentation

- 1、概述
- 2、接口说明
- 3、使用示例

## 1、概述

该代码模块集成在libDevicelo.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

```
typedef enum {
    RK_WIFI_State_IDLE = 0,
    RK_WIFI_State_CONNECTING,
    RK_WIFI_State_CONNECTFAILED,
    RK_WIFI_State_CONNECTFAILED_WRONG_KEY,
    RK_WIFI_State_CONNECTED,
    RK_WIFI_State_DISCONNECTED
} 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

```
typedef enum {
    NONE = 0,
    WPA,
    WEP
} RK_WIFI_CONNECTION_Encryp_e;
```

- `RK_WIFI_INFO_Connection_s`

WIFI状态信息，参考wpa\_cli -iwlan0 status

```
typedef struct {
    int id;
    char bssid[20];
    char ssid[64];
    int freq;
    char mode[20];
    char wpa_state[20];
    char ip_address[20];
    char mac_address[20];
} 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

```
#include <stdio.h>
#include <string.h>
#include <DeviceIo/Rk_wifi.h>

int _RK_wifi_state_callback(RK_WIFI_RUNNING_State_e state)
{
    printf("_RK_wifi_state_callback state:%d\n", state);
    return 0;
}
```

```
int main(int argc, char **argv)
{
    // 注册WIFI状态回调
    RK_wifi_register_callback(_RK_wifi_state_callback);

    // 设置hostname后获取打印
    char hostname[16];
    RK_wifi_set_hostname("RKWIFI");
    memset(hostname, 0, sizeof(hostname));
    RK_wifi_get_hostname(hostname, sizeof(hostname));
    printf("hostname:%s\n", hostname);

    // 获取MAC地址并打印
    char mac[32];
    memset(mac, 0, sizeof(mac));
    RK_wifi_get_mac(mac);
    printf("mac:%s\n", mac);

    // 如果有配置过WIFI, enable wifi自动连接到配置的WIFI
    // 否则连接到指定WIFI
    if (RK_wifi_has_config()) {
        RK_wifi_enable(1);
    } else {
        RK_wifi_enable(1);
        RK_wifi_connect("TP-LINK_C734BC", "12345678");
    }

    for (;;)
    // 断开WIFI并关闭WIFI模块
    RK_wifi_enable(0);

    return 0;
}
```