



# A76xx Series Open SDK WIFI SCAN\_应用指导

LTE 模组

芯讯通无线科技(上海)有限公司  
上海市长宁区临虹路289号3号楼芯讯通总部大楼  
电话: 86-21-31575100  
技术支持邮箱: support@simcom.com  
官网: www.simcom.com

名称:	A76xx Series Open SDK_WIFI SCAN_应用指导
版本:	V1.00
类别:	应用文档
状态:	已发布

## 版权声明

本手册包含芯讯通无线科技(上海)有限公司(简称:芯讯通)的技术信息。除非经芯讯通书面许可,任何单位和个人不得擅自摘抄、复制本手册内容的部分或全部,并不得以任何形式传播,违反者将被追究法律责任。对技术信息涉及的专利、实用新型或者外观设计等知识产权,芯讯通保留一切权利。芯讯通有权在不通知的情况下随时更新本手册的具体内容。

本手册版权属于芯讯通,任何人未经我公司书面同意进行复制、引用或者修改本手册都将承担法律责任。

### 芯讯通无线科技(上海)有限公司

上海市长宁区临虹路289号3号楼芯讯通总部大楼

电话: 86-21-31575100

邮箱: [simcom@simcom.com](mailto:simcom@simcom.com)

官网: [www.simcom.com](http://www.simcom.com)

了解更多资料, 请点击以下链接:

<http://cn.simcom.com/download/list-230-cn.html>

技术支持, 请点击以下链接:

<http://cn.simcom.com/ask/index-cn.html> 或发送邮件至 [support@simcom.com](mailto:support@simcom.com)

版权所有 © 芯讯通无线科技(上海)有限公司 2023, 保留一切权利。

## Version History

Version	Date	Owner	What is new
V1.00	2022-11-17		第一版

SIMCom  
Confidential

## About this Document

本文档适用于 A1803S open 系列、A1603 open 系列、A1606 open 系列。

SIMCom  
Confidential

# 目录

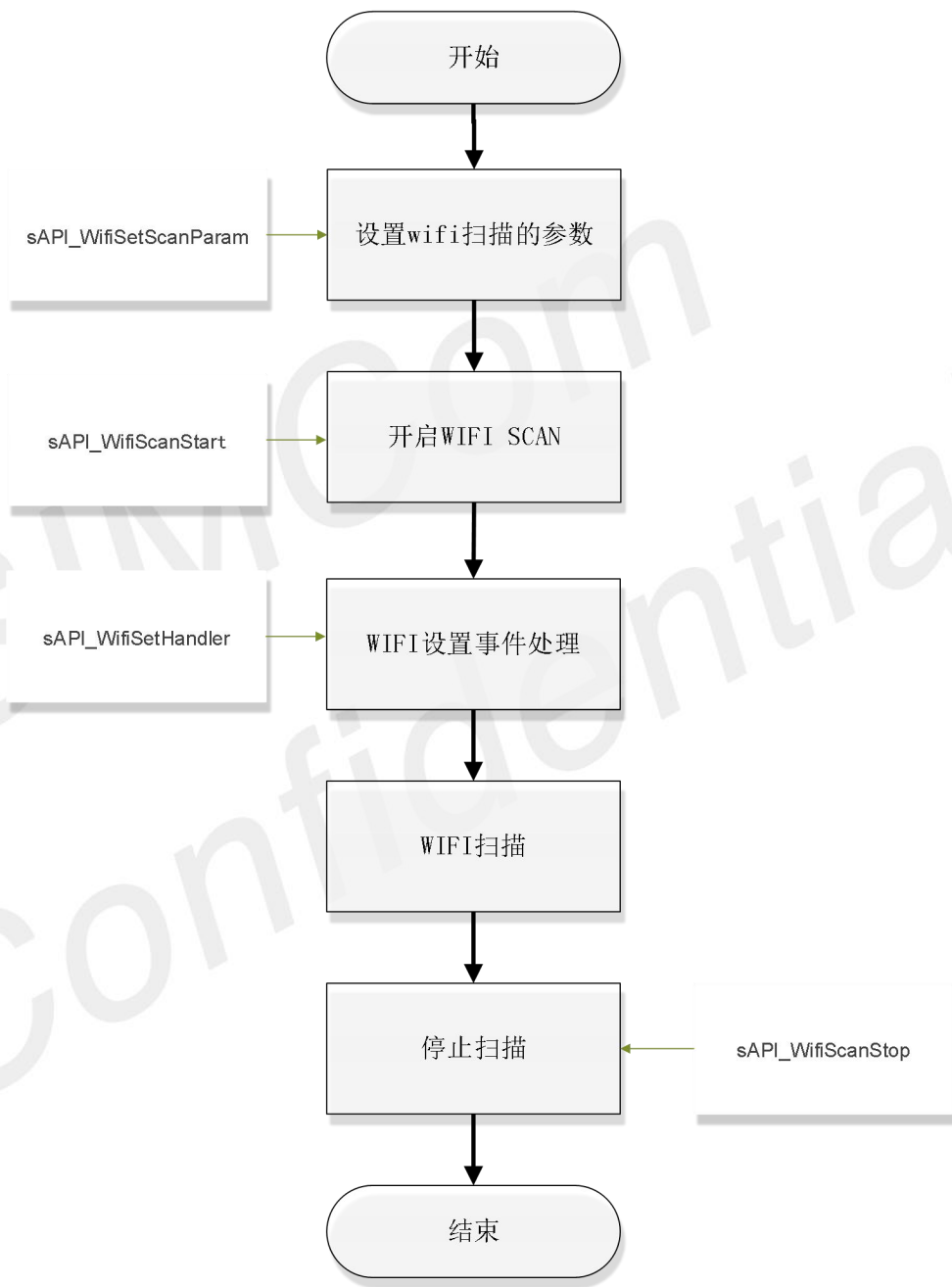
版权声明.....	2
Version History .....	3
About this Document .....	4
目录 .....	5
缩略语 .....	6
1WIFI SCAN 使用流程 .....	7
2API 介绍 .....	8
2.1 sAPI_WifiScanStart .....	8
2.2 sAPI_WifiScanStop .....	8
2.3 sAPI_WifiSetHandler .....	8
2.4 sAPI_WifiSetScanParam .....	9
3 变量定义 .....	10
3.1 SC_WIFI_EVENT_T .....	10
3.2 SC_WIFI_SCANPARAM_S .....	10
4Example .....	11
4.1 sAPI_WifiScanStart .....	11
4.2 sAPI_WifiScanStop .....	11
4.3 sAPI_WifiSetHandler .....	11
4.4 sAPI_WifiSetScanParam .....	12
5Demo .....	13
5.1 参考 demo .....	13
5.2 Demo 源码 .....	15

## 缩略语

RX	receive
PS	Packet Switch

SIMCom  
Confidential

## 1WIFI SCAN 使用流程



WIFI SCAN 工作流程图

wifi scan 功能，它不是全功能 wifi，只是 wifi RX 附近的 hotpot 扫描，主要是 wifi 定位。

在使用 wifi scan 之前，可使用 sAPI\_WifiSetScanParam 设置 wifi scan 的参数，也可使用默认参数直接执行扫描功能。

## 2API 介绍

WIFI SCAN 头文件名: simcom\_wifi.h

### 2.1 sAPI\_WifiScanStart

此应用程序界面是开始扫描 WIFI 网络。

接口:	void <b>sAPI_WifiScanStart</b> (void);
参数:	无
返回值:	无
NOTE:	

### 2.2 sAPI\_WifiScanStop

此应用程序界面是停止扫描 WIFI 网络。

接口:	void <b>sAPI_WifiScanStop</b> (void);
参数:	无
返回值:	无
NOTE:	

### 2.3 sAPI\_WifiSetHandler

这个应用程序接口是为 wifi 设置事件处理程序。

接口:	void <b>sAPI_WifiSetHandler</b> (void (*handler)(const SC_WIFI_EVENT_T *));
参数:	[in]事件句柄回调
返回值:	无
NOTE:	



## 2.4 sAPI\_WifiSetScanParam

该此应用程序界面是设置 wifi 扫描的参数。

接口:	void <b>sAPI_WifiSetScanParam</b> (const SC_WIFI_SCANPARAM_T *param);
参数:	参数:[in]wifi 扫描的参数 round_number:循环扫描数, 从 1 到 3, 默认是 3。 Max_bssid_number:最大扫描信息数, 从 4 到 10, 默认是 5。 timeout:扫描超时, 单位是秒, 默认是 25。 priority: 0 为 PS 优先, 1 为 wifi 优先, 默认值为 0。
返回值:	无
NOTE:	

## 3 变量定义

### 3.1 SC\_WIFI\_EVENT\_T

```
typedef struct
{
    unsigned char type;    //type: 0: SCAN_STOP; 1: SCAN_INFO
    union
    {
        SC_WIFI_INFO_T info;
    };
} SC_WIFI_EVENT_T;
```

```
typedef struct
{
    unsigned char mac_addr[6];    //mac 地址
    unsigned char padding[2];    //边缘填充
    int rssi;    //接收信号强度指示
    unsigned int channel_number;    //频道数
} SC_WIFI_INFO_T;
```

### 3.2 SC\_WIFI\_SCANPARAM\_S

```
typedef struct SC_WIFI_SCANPARAM_S
{
    unsigned char round_number;    //循环扫描数，从 1 到 3，默认是 3。
    unsigned char max_bssid_number;    //最大扫描信息数，从 4 到 10，默认是 5。
    unsigned char timeout;    // 超时时间，单位秒，默认为 25。
    unsigned char priority;    //优先级 0 为 PS 优先, 1 为 wifi 优先，默认为 0。
} SC_WIFI_SCANPARAM_T;
```

## 4Example

### 4.1 sAPI\_WifiScanStart

#### Examples

```
#include "simcom_wifi.h"

sAPI_WifiScanStart();
```

### 4.2 sAPI\_WifiScanStop

#### Examples

```
#include "simcom_wifi.h"

sAPI_WifiScanStop();
```

### 4.3 sAPI\_WifiSetHandler

#### Examples

```
#include "simcom_wifi.h"

static void wifi_handle_event(const SC_WIFI_EVENT_T *param)
{
    char rspBuf[128];

    switch (param->type)
    {
        case SC_WIFI_EVENT_SCAN_INFO:
            snprintf(rspBuf, sizeof(rspBuf),
                "\r\nscan result: mac address(%02x:%02x:%02x:%02x:%02x:%02x), channel(%d),\n",
                rssi(%d).\r\n",
                param->info.mac_addr[5], param->info.mac_addr[4], param->info.mac_addr[3],
                param->info.mac_addr[2], param->info.mac_addr[1], param->info.mac_addr[0],
```

```
        param->info.channel_number, param->info.rssi);
    break;
case SC_WIFI_EVENT_SCAN_STOP:
    snprintf(rspBuf, sizeof(rspBuf), "\r\nscan stop.\r\n");
    break;
default:
    snprintf(rspBuf, sizeof(rspBuf), "\r\ninvalid wifi event.\r\n");
}

PrintfResp((INT8 *)rspBuf);
}

sAPI_WifiSetHandler(wifi_handle_event);
```

## 4.4 sAPI\_WifiSetScanParam

### Examples

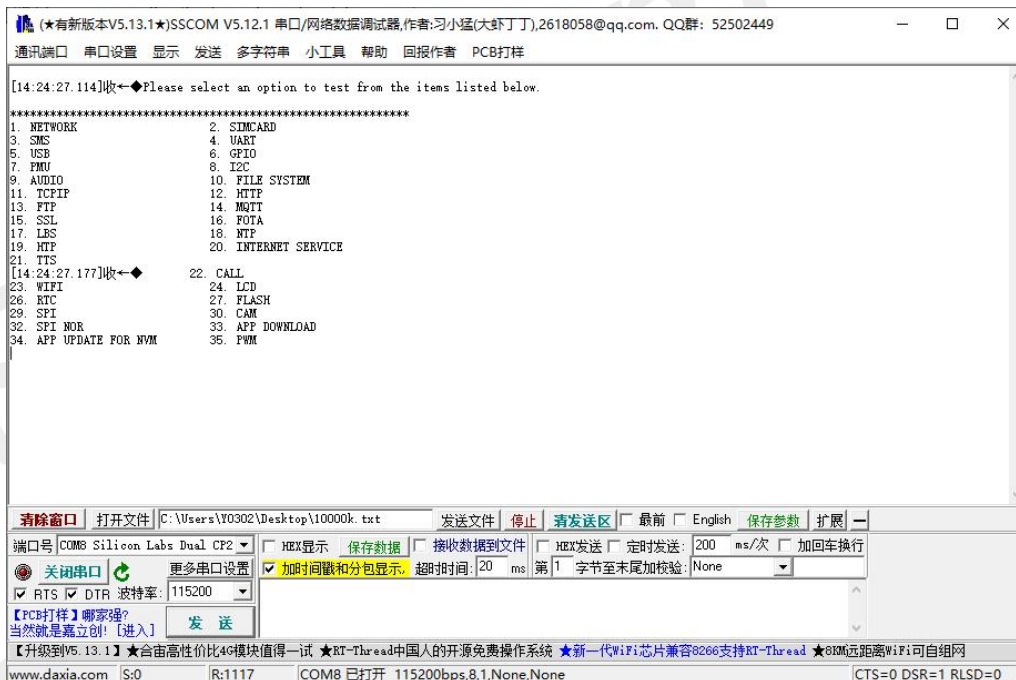
```
#include"simcom_wifi.h"

SC_WIFI_SCANPARAM_T param = {3, 5, 25, 1};
sAPI_WifiSetScanParam (&param);
```

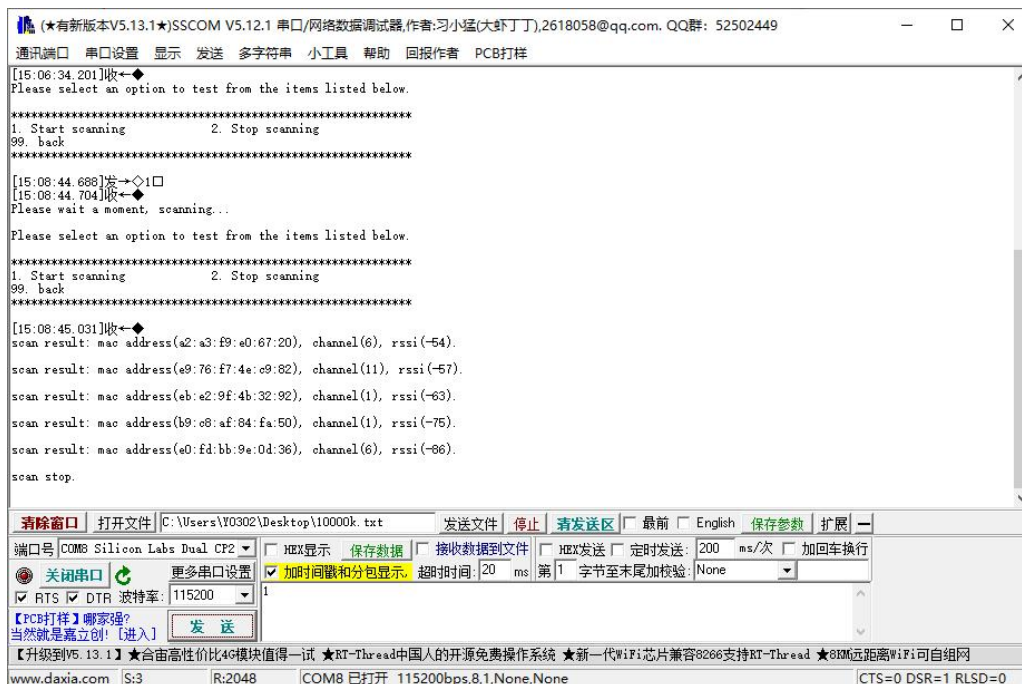
## 5Demo

### 5.1 参考 demo

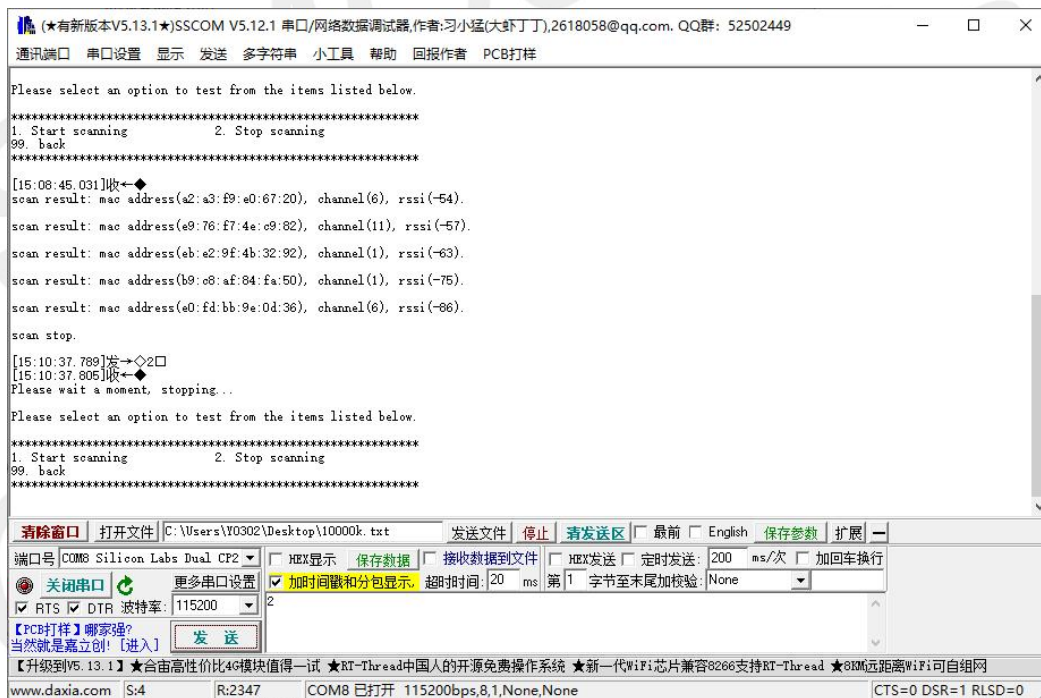
- 1) 将代码编译烧录至模块中，并重启模块
- 2) 打开串口/网络数据调试器，使用 Enhanced 口连接模块，串口助手会打印 demo UI



- 3) 进入 WIFI demo，然后选择 Start scanning 即可执行 WIFI SCAN 功能



## 4) 选择 Stop scanning 退出 WIFI SCAN 功能



## 5.2 Demo 源码

```
/**
*****
* @file    demo_wifi.c
* @author  SIMCom OpenSDK Team
* @brief   Source file of wifi scan funtion,it is not full function wifi, just wifi RX for nearby hotspot scanning, mostly for wifi location.
*****
* @attention
*
* Copyright (c) 2022 SIMCom Wireless.
* All rights reserved.
*
*****
*/

/* Includes -----*/
#include "string.h"
#include "stdlib.h"
#include "stdio.h"
#include "simcom_os.h"
#include "simcom_wifi.h"
#include "simcom_common.h"
#include "simcom_debug.h"
#include "simcom_uart.h"

enum
{
    SC_WIFI_DEMO_START_SCANNING      = 1,
    SC_WIFI_DEMO_STOP_SCANNING       = 2,
    SC_WIFI_DEMO_MAX                  = 99
};

extern sMsgQRef simcomUI_msgq;
extern void PrintfOptionMenu(char* options_list[], int array_size);
extern void PrintfResp(char* format);

/**
* @brief   Event handler for wifi scan.
* @param   param
* @note    Will output scanned MAC\channel number\rssi.
* @retval  void
*/
static void wifi_handle_event(const SC_WIFI_EVENT_T *param)
{
    char rspBuf[128];
```

```
switch (param->type)
{
    case SC_WIFI_EVENT_SCAN_INFO:
        snprintf(rspBuf, sizeof(rspBuf),
            "\nscan result: mac address(%02x:%02x:%02x:%02x:%02x:%02x), channel(%d), rssi(%d).\n",
            param->info.mac_addr[5], param->info.mac_addr[4], param->info.mac_addr[3],
            param->info.mac_addr[2], param->info.mac_addr[1], param->info.mac_addr[0],
            param->info.channel_number, param->info.rssi);
        break;
    case SC_WIFI_EVENT_SCAN_STOP:
        snprintf(rspBuf, sizeof(rspBuf), "\nscan stop.\n");
        break;
    default:
        snprintf(rspBuf, sizeof(rspBuf), "\ninvalid wifi event.\n");
}

PrintfResp((char *)rspBuf);
}

/**
 * @brief  UI demo for WIFI scan.
 * @param  void
 * @note
 * @retval void
 */
void WIFIDemo(void)
{
    char flag = 1;
    char *note = "\nPlease select an option to test from the items listed below.\n";
    char *options_list[] = {
        "1. Start scanning",
        "2. Stop scanning",
        "99. back",
    };

    while (flag)
    {
        PrintfResp(note);
        PrintfOptionMenu(options_list, sizeof(options_list) / sizeof(options_list[0]));

        SIM_MSG_T optionMsg;
        sAPI_MsgQRecv(simcomUI_msgq, &optionMsg, SC_SUSPEND);
        if (SRV_UART != optionMsg.msg_id)
        {
            sAPI_Debug("%s,msg_id is error!!", __func__);
        }
    }
}
```



```
        break;
    }

    unsigned char opt = atoi(optionMsg.arg3);
    sAPI_Free(optionMsg.arg3);

    switch (opt)
    {
        case SC_WIFI_DEMO_START_SCANNING:
        {
            sAPI_WifiSetHandler(wifi_handle_event);
            sAPI_WifiScanStart();
            PrintfResp("\n\nPlease wait a moment, scanning...\n\n");
            break;
        }

        case SC_WIFI_DEMO_STOP_SCANNING:
        {
            sAPI_WifiScanStop();
            PrintfResp("\n\nPlease wait a moment, stopping...\n\n");
            break;
        }

        case SC_WIFI_DEMO_MAX:
        {
            flag = 0;
            PrintfResp("\n\nReturn to the previous menu!\n\n");
            break;
        }

        default:
            PrintfResp("\n\nPlease select again:\n\n");
    }
}
}
```