



XR806 WLAN 配网应用 开发指南

版本号：1.0

发布时间：2020-11-12

版本历史

版本	日期	责任人	版本描述
1.0	2020-11-12	AWA 1029	创建文档。

目录

版本历史.....	i
目录.....	ii
图片目录.....	vi
表格目录.....	vii
1 前言.....	1
1.1 文档简介.....	1
1.2 目标读者.....	1
1.3 适用范围.....	1
1.4 文档约定.....	1
1.4.1 标志说明.....	1
1.4.2 地址与数据描述方法约定.....	1
1.4.3 数值单位约定.....	2
2 概述.....	3
2.1 背景说明.....	3
2.2 规格特性.....	3
2.3 文件位置.....	3
3 AirKiss 配网.....	5
3.1 技术说明.....	5
3.1.1 配网原理.....	5
3.1.1.1 物理层协议.....	5
3.1.1.2 链路层协议.....	5
3.2 应用说明.....	6
3.2.1 应用简述.....	6
3.2.2 配置说明.....	7
3.2.3 接口说明.....	7
3.2.3.1 wlan_airkiss_start.....	7
3.2.3.2 wlan_airkiss_wait.....	7
3.2.3.3 wlan_airkiss_get_status.....	8
3.2.3.4 wlan_airkiss_get_result.....	8
3.2.3.5 wlan_airkiss_connect_ack.....	8
3.2.3.6 wlan_airkiss_stop.....	9
3.3 示例说明.....	9

3.3.1 示例简介.....	9
3.3.1.1 获取方式.....	9
3.3.1.2 准备工作.....	9
3.3.1.3 操作步骤.....	9
3.3.2 关键实现.....	10
3.3.3 效果展示.....	12
4 SmartConfig 配网.....	14
4.1 技术说明.....	14
4.1.1 配网原理.....	14
4.2 应用说明.....	14
4.2.1 应用简述.....	14
4.2.2 配置说明.....	14
4.2.3 接口说明.....	14
4.2.3.1 wlan_smart_config_start.....	14
4.2.3.2 wlan_smart_config_wait.....	15
4.2.3.3 wlan_smart_config_get_status.....	15
4.2.3.4 smartconfig_get_result.....	15
4.2.3.5 wlan_smart_config_connect_ack.....	16
4.2.3.6 wlan_smart_config_stop.....	16
4.3 示例说明.....	16
4.3.1 示例简介.....	16
4.3.1.1 获取方式.....	16
4.3.1.2 准备工作.....	17
4.3.1.3 操作步骤.....	17
4.3.2 关键实现.....	17
4.3.3 效果展示.....	20
5 VoicePrint 配网.....	22
5.1 技术说明.....	22
5.1.1 配网原理.....	22
5.2 应用说明.....	22
5.2.1 应用简述.....	22
5.2.2 配置说明.....	22
5.2.3 接口说明.....	23
5.2.3.1 voiceprint_start.....	23
5.2.3.2 voiceprint_wait.....	23

5.2.3.3 voiceprint_get_status.....	24
5.2.3.4 wlan_voiceprint_get_raw_result.....	24
5.2.3.5 voiceprint_stop.....	24
5.3 示例说明.....	24
5.3.1 示例简介.....	24
5.3.1.1 获取方式.....	25
5.3.1.2 准备工作.....	25
5.3.1.3 操作步骤.....	25
5.3.2 关键实现.....	25
5.3.3 效果展示.....	28
6 SoftAP 配网.....	30
6.1 技术说明.....	30
6.1.1 配网原理.....	30
6.2 应用说明.....	30
6.2.1 应用简述.....	30
6.2.2 配置说明.....	30
6.2.3 接口说明.....	31
6.2.3.1 soft_ap_config_start.....	31
6.2.3.2 soft_ap_config_stop.....	31
6.2.3.3 soft_ap_config_get_state.....	32
6.2.3.4 soft_ap_config_get_result.....	32
6.2.3.5 soft_ap_config_set_cb.....	32
6.3 示例说明.....	32
6.3.1 示例简介.....	32
6.3.1.1 获取方式.....	33
6.3.1.2 准备工作.....	33
6.3.1.3 操作步骤.....	33
6.3.2 关键实现.....	33
6.3.3 效果展示.....	35
7 Sc_assistant 配网助手.....	38
7.1 技术说明.....	38
7.2 应用说明.....	38
7.2.1 应用简述.....	38
7.2.2 接口说明.....	38
7.2.2.1 sc_assistant_get_fun.....	38

7.2.2.2 sc_assistant_init.....	38
7.2.2.3 sc_assistant_deinit.....	39
7.2.2.4 sc_assistant_monitor_register_rx_cb.....	39
7.2.2.5 sc_assistant_monitor_unregister_rx_cb.....	39
7.2.2.6 sc_assistant_open_monitor.....	40
7.2.2.7 sc_assistant_open_sta.....	40
7.2.2.8 sc_assistant_get_status.....	40
7.2.2.9 sc_assistant_newstatus.....	41
7.3 示例说明.....	41

图片目录

图 3-1	802.2 SNAP 帧格式.....	5
图 3-2	Airkiss 通信协议格式.....	6
图 3-3	Prefix SSID/PSK code.....	6
图 3-4	Airkiss 配网流程.....	10
图 4-1	SmartConfig 配网流程.....	18
图 5-1	VoicePrint 配网架构图.....	22
图 5-2	VoicePrint 配网流程.....	26
图 6-1	SoftAP 配网流程.....	34

表格目录

表 2-1	XR806 SDK 支持的配网方式.....	3
表 2-2	WLAN config 的相关文件位置.....	3
表 3-1	Airkiss 模块配置列表.....	7
表 3-2	wlan_airkiss_start 接口函数说明.....	7
表 3-3	wlan_airkiss_wait 接口函数说明.....	7
表 3-4	wlan_airkiss_get_status 接口函数说明.....	8
表 3-5	wlan_airkiss_get_result 接口函数说明.....	8
表 3-6	wlan_airkiss_connect_ack 接口函数说明.....	8
表 3-7	wlan_airkiss_stop 接口函数说明.....	9
表 4-1	SmartConfig 模块配置列表.....	14
表 4-2	wlan_smart_config_start 接口函数说明.....	14
表 4-3	wlan_smart_config_wait 接口函数说明.....	15
表 4-4	wlan_smart_config_get_status 接口函数说明.....	15
表 4-5	smartconfig_get_result 接口函数说明.....	15
表 4-6	wlan_smart_config_connect_ack 接口函数说明.....	16
表 4-7	wlan_smart_config_stop 接口函数说明.....	16
表 5-1	VoicePrint 模块配置列表.....	22
表 5-2	voiceprint_start 接口函数说明.....	23
表 5-3	voiceprint_wait 接口函数说明.....	23
表 5-4	voiceprint_get_status 接口函数说明.....	24
表 5-5	wlan_voiceprint_get_raw_result 接口函数说明.....	24
表 5-6	voiceprint_stop 接口函数说明.....	24
表 6-1	SoftAP 模块配置列表.....	30
表 6-2	soft_ap_config_start 接口函数说明.....	31
表 6-3	soft_ap_config_stop 接口函数说明.....	31
表 6-4	soft_ap_config_get_state 接口函数说明.....	32
表 6-5	soft_ap_config_get_result 接口函数说明.....	32
表 6-6	soft_ap_config_set_cb 接口函数说明.....	32
表 7-1	sc_assistant_get_fun 接口函数说明.....	38
表 7-2	sc_assistant_init 接口函数说明.....	38
表 7-3	sc_assistant_deinit 接口函数说明.....	39

表 7-4	sc_assistant_monitor_register_rx_cb 接口函数说明.....	39
表 7-5	sc_assistant_monitor_unregister_rx_cb 接口函数说明.....	39
表 7-6	sc_assistant_open_monitor 接口函数说明.....	40
表 7-7	sc_assistant_open_sta 接口函数说明.....	40
表 7-8	sc_assistant_get_status 接口函数说明.....	40
表 7-9	sc_assistant_newstatus 接口函数说明.....	41

1 前言

1.1 文档简介

此文档用以解释 XR806 SDK 所使用的配网应用，帮助开发者了解如何使用配网接口完成配网的流程。

1.2 目标读者

XR806 SDK 用户。

1.3 适用范围

此文档适用于 XR806 SDK，支持 XR806 系列芯片产品。

1.4 文档约定

1.4.1 标志说明

本文档采用各种醒目的标志来表示在操作过程中应该特别注意的地方，这些标志的含义如下：

标识	说明
 警告	该标志后的说明应给予格外关注，如果不遵守，可能会导致人员受伤或死亡。
 注意	提醒操作中应注意的事项。不当的操作可能会损坏器件，影响可靠性、降低性能等。
 说明	为准确理解文中指令、正确实施操作而提供的补充或强调信息。
 窍门	一些容易忽视的小功能、技巧。了解这些功能或技巧能帮助解决特定问题或者节省操作时间。

1.4.2 地址与数据描述方法约定

本文档在描述地址、数据时遵循如下约定：

符号	例子	说明
0x	0x0200, 0x79	地址或数据以 16 进制表示。
0b	0b010, 0b00 000 111	数据采用二进制表示(寄存器描述除外)。
x	00X, XX1	数据描述中，x 代表 0 或 1。 例如，00X 代表 000 或 001；XX1 代表 001, 011, 101 或 111。

1.4.3 数值单位约定

本文档在描述数据容量（如 NAND 容量）时，单位词头代表的是 1024 的倍数；描述频率、数据速率等时则代表的是 1000 的倍数。具体如下：

类型	符号	对应数值
数据容量（如 NAND 容量）	1 K	1024
	1 M	1 048 576
	1 G	1 073 741 824
频率，数据速率等	1 k	1000
	1 M	1 000 000
	1 G	1 000 000 000

2 概述

2.1 背景说明

配网（WLAN Config），即 WLAN 设备快速入网配置技术，主要用于使一些不具备输入能力的设备连接上互联网。一般会在以下场景中使用：

- 1.待接入互联网的设备不具备输入输出能力，如空调、空气净化器、烟雾报警器等。
- 2.用户不具备通过设备热点的方式进行配置的能力，如老人、家庭主妇等缺乏相关 IT 知识的用户人群。

2.2 规格特性

目前 XR806 SDK 支持 4 种配网方式，分别为 AirKiss 配网、SmartConfig 配网、VoicePrint 配网和 SoftAP 配网。为了兼容多种配网方式同时启动，SDK 提供了配网助手 sc_assistant，来协调各个配网模块的运行。本文主要介绍这 4 种配网方式和配网助手 sc_assistant。

表 2-1 XR806 SDK 支持的配网方式

规格类型	支持规格	规格描述	备注
配网方式	AirKiss 配网	使用 airkiss 软件通过无线进行配网	
	SmartConfig 配网	使用 SmartConfig 软件通过无线进行配网	
	VoicePrint 配网	使用 SoundConfig 软件通过声波进行配网	
	SoftAP 配网	使用 SoftAP 模式在手机端设置 SSID 和 PSK 进行配网	
配网助手	sc_assistant	协调各个配网模块的运行	

2.3 文件位置

以 SDK 包为根目录，本中间件涉及到的主要文件位置如下。

表 2-2 WLAN config 的相关文件位置

组件名	文件分类	文件位置
airkiss	源码文件	./src/smartlink/airkiss
	头文件	./include/smartlink/airkiss/wlan_airkiss.h
	示例工程	./project/example/airkiss/
smartconfig	源码文件	./src/smartlink/smartconfig
	头文件	./include/smartlink/smartconfig/wlan_smartconfig.h
	示例工程	./project/example/smartconfig/
Voice print	源码文件	./src/smartlink/voice_print
	头文件	./include/smartlink/voice_print/voice_print.h

组件名	文件分类	文件位置
	示例工程	./project/example/voice_print/
Sc_assistant	库文件	./lib/libsc_assistant.a
	头文件	./include/smartlink/sc_assistant.h



说明

XR806 SDK 可在以下 GitHub 仓库获取：https://github.com/XradioTech/xr806_sdk.git

3 AirKiss 配网

3.1 技术说明

3.1.1 配网原理

Airkiss 是微信硬件平台为 WLAN 设备提供的微信配网、局域网发现和局域网通讯的技术。XRADIO 为开发者提供了通过微信客户端对 WLAN 设备配网，以及通过微信客户端局域网发现 WLAN 设备的功能。其本质是通过 WLAN 收发包来达到传递信息的目的。

WLAN 设备有一种 monitor 模式，可以监听空中的所有 WLAN 数据包。WLAN 设备处于正常 station 模式下，设备是会过滤掉目标 mac 地址与自身 mac 地址不匹配的数据包，而在 monitor 模式下，WLAN 设备可以接收所有符合 802.11 协议的数据包。Airkiss 配网就是借用了 WLAN 设备的这种模式，来实现配网的目的。

当启动 airkiss 配网后，手机 app 端会在固定的信道发送特定的数据包。而 WLAN 设备会启动 monitor 模式，监听接收空中的所有 WLAN 数据包，且设备会在接收识别到 app 端发送的特定数据包前，不断切换 WLAN 信道，从而锁定 app 端信道，并在该信道进行后续的通信。

3.1.1.1 物理层协议

Airkiss 利用 802.2 SNAP（802.11 的物理层协议）的数据帧进行信息传递。其中该数据帧的格式如下：

图 3-1 802.2 SNAP 帧格式

DA	SA	Length	LLC	SNAP	DATA	FCS
6bytes	6bytes	2bytes	3bytes	5bytes	38-1492bytes	4bytes

其中 DA 表示目的 mac 地址，SA 表示源 mac 地址，Length 表示数据的长度，LLC 表示 LLC 头部，SNAP 包括了厂商代码和协议类型标识，DATA 为有效数据区，对于加密信道来说是密文，FCS 表示帧校验区域。

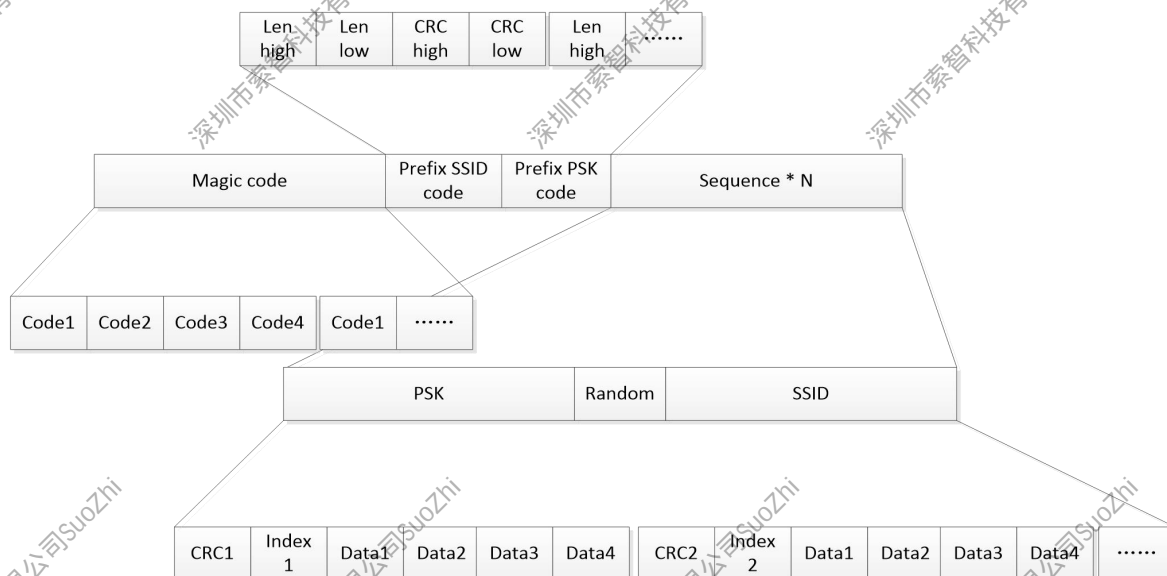
从无线信号监听方的角度来说，不管无线信道有没有加密，DA、SA、Length、LLC、SNAP、FCS 字段总是明文状态，因此监听方便可以从这 6 个字段获取到一定的信息。但从发送方的角度来说，由于操作系统的限制，DA、SA、LLC、SNAP、FCS 五个字段的控制需要很高的控制权限，发送方很难取得控制权进行修改。因此，只能利用 Length 字段进行信息传递。所以，只要制定出一套利用长度字段编码的通信协议，就可利用 802.2 SNAP 数据包中的 Length 字段进行信息传递。

在实际应用中，airkiss 采用 UDP 广播包作为信息的载体。信息发送方向空间中发送一系列的 UDP 广播包，其中每一包的 Length 字段都按照 airkiss 通信协议进行编码，信息接收方利用 monitor 模式监听接收空间中的无线信号，并从数据链路层获取 802.2 SNAP 格式数据包，便可得到已编码的 Length 字段，随后接收方便可根据 airkiss 通信协议解析出需要的信息。

3.1.1.2 链路层协议

Airkiss 通信协议利用 UDP 数据包的长度进行编码，即有效数据都存储在 UDP 包的 length 字段，每次最多可使用 9bit 的数据，因为每个数据包只能传输 1byte 数据。发送的数据可以分成 4 类：Magic code、Prefix SSID code、Prefix PSK code、Sequence * N，其组织结构如下图所示：

图 3-2 Airkiss 通信协议格式



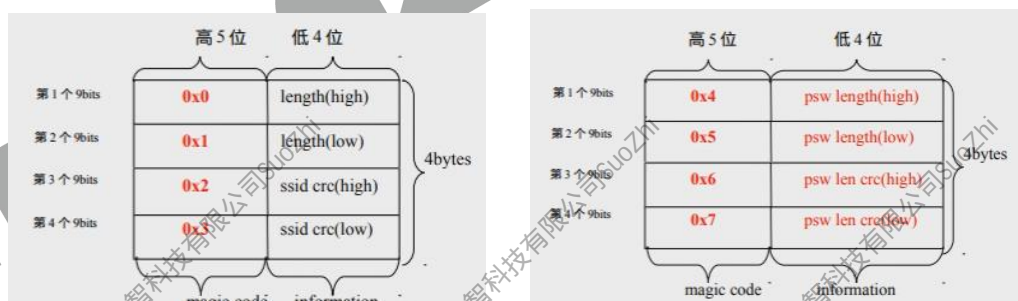
1、Magic code

Magic code 为 4 个连续的 9bit 数据，用于计算后续数据的基准偏移，例如 4 个值为 61,62,63,64，则偏移为第一个值减 1，即 60。

2、Prefix SSID/PSK code

Prefix SSID code 和 prefix PSK code 格式相同，都是以 4 个 9bit 数据为一组，每个数据以高 5bit 存储 index，前两个数据的低 4bit 为 length，后两个的数据的低 4bit 为 CRC。

图 3-3 Prefix SSID/PSK code



3、Sequence

Sequence 包含了 PSK、random 值以及 SSID 的具体内容，airkiss 将其分成 4byte 一组，每一组为其加上 2byte 的 CRC 和 index，所以最后发出来的数据为 6byte 一组。需要注意的是，最后的数据如果没有 4byte 对齐，airkiss 不会自动补上数据，而是保持不对齐的格式进行解析。

3.2 应用说明

3.2.1 应用简述

Airkiss 配网方案已经在 XR806 SDK 中实现了，用户通过 airkiss 模块提供的接口即可开发出基于配网助手的 airkiss 配网功能。

3.2.2 配置说明

Airkiss 配网方式的启用需要使能 SDK 的 WLAN 功能，具体配置项如下表所示。

表 3-1 Airkiss 模块配置列表

配置项	配置说明
WLAN 功能使能	设置说明： 此项配置用于在 SDK 中启用 WLAN 功能。 设置方式： 1、在 prj_config.h 文件，添加或修改以下定义，其中 1 为使能 NET 功能，0 为不使能 NET 功能。 <pre>/* network and wlan enable/disable */ #define PRJCONF_NET_EN 1</pre> 2、在控制台执行命令“make menuconfig”，将“NET and WLAN”功能项打开。

3.2.3 接口说明

Airkiss 模块总共提供 6 套接口，主要功能点涵括了进入配网、等待获取配网信息、获取模块运行状态、连接网络、退出配网等过程。

3.2.3.1 wlan_airkiss_start

表 3-2 wlan_airkiss_start 接口函数说明

信息项	说明
原型	wlan_airkiss_status_t wlan_airkiss_start(struct netif *nif, char *key);
功能	启动 airkiss
参数	struct netif *nif 含义解释：网络接口 char *key 含义解释：airkiss 传输使用的密码
返回值	0：成功 -1：失败

3.2.3.2 wlan_airkiss_wait

表 3-3 wlan_airkiss_wait 接口函数说明

信息项	说明
原型	wlan_airkiss_status_t wlan_airkiss_wait(uint32_t timeout_ms);
功能	等待 airkiss 执行结束
参数	uint32_t timeout_ms

信息项	说明
	含义解释：等待超时时间
返回值	0：成功 -1：失败

3.2.3.3 wlan_airkiss_get_status

表 3-4 wlan_airkiss_get_status 接口函数说明

信息项	说明
原型	airkiss_status_t wlan_airkiss_get_status(void);
功能	获取配网的状态
参数	无
返回值	返回配网状态

3.2.3.4 wlan_airkiss_get_result

表 3-5 wlan_airkiss_get_result 接口函数说明

信息项	说明
原型	wlan_airkiss_status_t wlan_airkiss_get_result(wlan_airkiss_result_t *result);
功能	获取配网的结果
参数	wlan_airkiss_result_t *result 含义解释：配网结果结构体的指针
返回值	0：成功 -1：失败

3.2.3.5 wlan_airkiss_connect_ack

表 3-6 wlan_airkiss_connect_ack 接口函数说明

信息项	说明
原型	wlan_airkiss_status_t wlan_airkiss_connect_ack(struct netif *nif, uint32_t timeout_ms, wlan_airkiss_result_t *result);
功能	连接 ap，并返回 ack 给手机端 app
参数	struct netif *nif 含义解释：网络接口 uint32_t timeout_ms 含义解释：超时时间 wlan_airkiss_result_t *result 含义解释：获取到的配网结果

信息项	说明
返回值	0：成功 -1：失败

3.2.3.6 wlan_airkiss_stop

表 3-7 wlan_airkiss_stop 接口函数说明

信息项	说明
原型	int wlan_airkiss_stop(void);
功能	停止 airkiss 模块
参数	无
返回值	0：成功 -1：失败

3.3 示例说明

3.3.1 示例简介

Airkiss 示例工程展示了 XR806 SDK 中使用 airkiss 方式进行配网的代码实现方法。

3.3.1.1 获取方式

Airkiss 示例有示例工程代码，位于 XR806 SDK 的/project/example/airkiss 目录，以下此示例工程简称为 airkiss 示例工程。



说明

XR806 SDK 可在以下 GitHub 仓库获取：https://github.com/XradioTech/xr806_sdk.git

3.3.1.2 准备工作

Airkiss 示例工程的硬件准备有如下。

1. 评估板：运行示例工程代码。
2. 串口线：连接评估板的 UART0 插针，用于 console 控制台的输入输出。
3. PC 机：用于镜像烧录和 console 控制的输入输出。

Airkiss 示例工程的软件准备，包括烧写工具、代码编译和烧写操作，请参见《XR806_SDK_快速入门指南》。

3.3.1.3 操作步骤

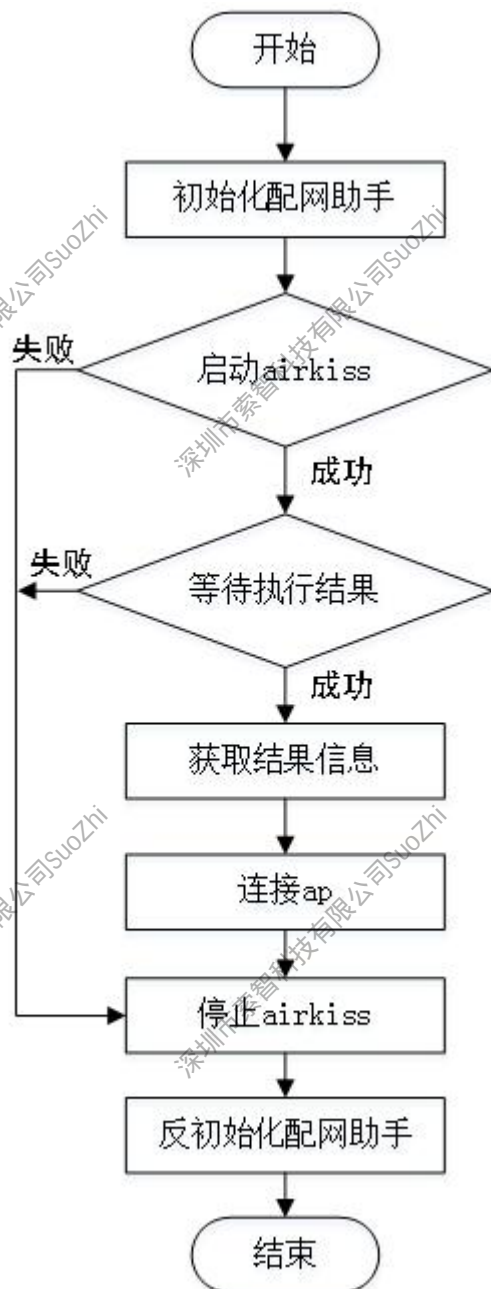
Airkiss 示例工程无需控制命令，完成烧写后，复位即可，示例代码自动运行。配合手机端的 airkiss app 操作即可完成配网。

3.3.2 关键实现

系统平台的初始化在 XR806 SDK 的 platform_init()函数中完成，具体请阅读此函数。本文重点描述 airkiss 模块启动、接收空中包并等待配网结果，最后完成联网的过程。

为了结合配网助手，xradio sdk 提供的配网流程和配网接口都使用配网助手进行了一定的封装。其中 airkiss 配网流程如下图：

图 3-4 Airkiss 配网流程



具体操作见下面的示例代码。

```
#define AK_TIME_OUT_MS 120000 //120s
```

```
static char airkiss_key[17] = "1234567812345678"; // airkiss aes key

static void AirkissDemo(void)
{
    wlan_airkiss_status_t ak_status;
    wlan_airkiss_result_t ak_result;
    sc_assistant_fun_t sca_fun;
    sc_assistant_time_config_t config;

    memset(&ak_result, 0, sizeof(wlan_airkiss_result_t));

    /*
     * Initialize the distribution network assistant,
     * which can improve the success rate of distribution network
     */
    sc_assistant_get_fun(&sca_fun);
    config.time_total = AK_TIME_OUT_MS;
    config.time_sw_ch_long = 400;
    config.time_sw_ch_short = 100;
    sc_assistant_init(g_wlan_netif, &sca_fun, &config);

    /* start airkiss */
    ak_status = wlan_airkiss_start(g_wlan_netif, airkiss_key);
    if (ak_status != WLAN_AIRKISS_SUCCESS) {
        printf("airkiss start failed!\n");
        goto out;
    }

    printf("%s getting ssid and psk...\n", __func__);

    /* Blocking waiting for distribution network to complete */
    if (wlan_airkiss_wait(AK_TIME_OUT_MS) == WLAN_AIRKISS_TIMEOUT) {
        printf("%s get ssid and psk timeout\n", __func__);
        goto out;
    }

    printf("%s get ssid and psk finished\n", __func__);

    /* if complete, then get the airkiss result */
    if (wlan_airkiss_get_status() == AIRKISS_STATUS_COMPLETE && \
        wlan_airkiss_get_result(&ak_result) == WLAN_AIRKISS_SUCCESS) {
        printf("ssid:%s psk:%s random:%d\n", (char *)ak_result.ssid,
            (char *)ak_result.passphrase, ak_result.random_num);

        /* if get result, then connect ap and send the airkiss connect ack */
    }
}
```

```

        if (!wlan_arkiss_connect_ack(g_wlan_netif, AK_TIME_OUT_MS, &ak_result)) {
            printf("connect and ack ok\n");
        }
    }

out:
    /* stop arkiss and deinit distribution network assistant */
    wlan_arkiss_stop();
    sc_assistant_deinit(g_wlan_netif);
}

```

3.3.3 效果展示

在评估板中运行 arkiss 示例程序后，使用手机 arkiss app 启动配网功能，在控制台中打印输出如下。

```

Airkiss demo started.

en1: CTRL-EVENT-TERMINATING
WAR join_status:0

wlan information =====
firmware:
    version : R0-XR_C07.08.52.65_02.05 Nov 17 2020 14:44:43-Y02.05
    buffer   : 8
driver:
    version : XR_V02.05
mac address:
    in use   : b4:cd:28:7f:a7:06
    in use   : b4:cd:28:7f:a7:07
=====

AirkissDemo getting ssid and psk...
AirkissDemo get ssid and psk finished
ssid:My_ssid psk:My_password random:212

wlan information =====
firmware:
    version : R0-XR_C07.08.52.65_02.05 Nov 17 2020 14:44:43-Y02.05
    buffer   : 8
driver:
    version : XR_V02.05
mac address:
    in use   : b4:cd:28:7f:a7:06
    in use   : b4:cd:28:7f:a7:07

```

```
=====
[net INF] no need to switch wlan mode 0
en1: Trying to associate with 80:37:73:f7:dc:f4 (SSID='AW-ANC-NETGEAR-JNR3300#24' freq=2412 MHz)
[net INF] msg <wlan scan success>
en1: Associated with 80:37:73:f7:dc:f4
en1: WPA: Key negotiation completed with 80:37:73:f7:dc:f4 [PTK=CCMP GTK=CCMP]
en1: CTRL-EVENT-CONNECTED - Connection to 80:37:73:f7:dc:f4 completed [id=0 id_str=]
[net INF] msg <wlan connected>
[net INF] netif is link up
[net INF] start DHCP...
[net INF] netif is up
[net INF] address: 192.168.47.32
[net INF] gateway: 192.168.47.1
[net INF] netmask: 255.255.255.0
[net INF] msg <network up>
connect and ack ok
[net INF] no need to switch wlan mode 0

Airkiss demo over
```

4 SmartConfig 配网

4.1 技术说明

4.1.1 配网原理

SmartConfig 配网是 xradio 基于自己的通信协议开发出来的配网技术，其本质是通过 WLAN 收发包来达到传递信息的目的。

4.2 应用说明

4.2.1 应用简述

SmartConfig 配网方案已经在 XR806 SDK 中实现了，用户通过 SmartConfig 模块提供的接口即可开发出基于配网助手的 SmartConfig 配网功能。

4.2.2 配置说明

SmartConfig 配网方式的启用需要使能 SDK 的 WLAN 功能，具体配置项如下表所示。

表 4-1 SmartConfig 模块配置列表

配置项	配置说明
WLAN 功能使能	设置说明： 此项配置用于在 SDK 中启用 WLAN 功能。 设置方式： 1、在 prj_config.h 文件，添加或修改以下定义，其中 1 为使能 NET 功能，0 为不使能 NET 功能。 <pre>/* network and wlan enable/disable */ #define PRJCONF_NET_EN 1</pre> 2、在控制台执行命令“make menuconfig”，将“NET and WLAN”功能项打开。

4.2.3 接口说明

SmartConfig 模块总共提供 6 套接口，主要功能点涵括了进入配网、等待获取配网信息、获取模块运行状态、连接网络、退出配网等过程。

4.2.3.1 wlan_smart_config_start

表 4-2 wlan_smart_config_start 接口函数说明

信息项	说明
原型	wlan_smart_config_status_t wlan_smart_config_start(struct netif *nif, char *key);

信息项	说明
功能	启动 smartconig
参数	struct netif *nif 含义解释：网络接口 char *key 含义解释：smartconig 传输使用的密码
返回值	0：成功 -1：失败

4.2.3.2 wlan_smart_config_wait

表 4-3 wlan_smart_config_wait 接口函数说明

信息项	说明
原型	wlan_smart_config_status_t wlan_smart_config_wait(uint32_t timeout_ms);
功能	等待 smartconfig 执行结束
参数	uint32_t timeout_ms 含义解释：等待超时时间
返回值	0：成功 -1：失败

4.2.3.3 wlan_smart_config_get_status

表 4-4 wlan_smart_config_get_status 接口函数说明

信息项	说明
原型	SMART_CONFIG_STATUS_T wlan_smart_config_get_status(void);
功能	获取配网的状态
参数	无
返回值	返回配网状态

4.2.3.4 smartconfig_get_result

表 4-5 smartconfig_get_result 接口函数说明

信息项	说明
原型	wlan_smart_config_status_t smartconfig_get_result(wlan_smart_config_result_t *result);
功能	获取配网的结果
参数	wlan_smart_config_result_t *result 含义解释：配网结果结构体的指针
返回值	0：成功

信息项	说明
	-1: 失败

4.2.3.5 wlan_smart_config_connect_ack

表 4-6 wlan_smart_config_connect_ack 接口函数说明

信息项	说明
原型	wlan_smart_config_status_t wlan_smart_config_connect_ack(struct netif *nif, uint32_t timeout_ms, wlan_smart_config_result_t *result);
功能	连接 ap，并返回 ack 给手机端 app
参数	struct netif *nif 含义解释：网络接口 uint32_t timeout_ms 含义解释：超时时间 wlan_smart_config_result_t *result 含义解释：获取到的配网结果
返回值	0: 成功 -1: 失败

4.2.3.6 wlan_smart_config_stop

表 4-7 wlan_smart_config_stop 接口函数说明

信息项	说明
原型	int wlan_smart_config_stop(void);
功能	停止 smartconfig 模块
参数	无
返回值	0: 成功 -1: 失败

4.3 示例说明

4.3.1 示例简介

SmartConfig 示例工程展示了 XR806 SDK 中使用 SmartConfig 方式进行配网的代码实现方法。

4.3.1.1 获取方式

SmartConfig 示例有示例工程代码，位于 XR806 SDK 的/project/example/SmartConfig 目录，以下此示例工程简称为 SmartConfig 示例工程。



说明

XR806 SDK 可在以下 GitHub 仓库获取：https://github.com/XradioTech/xr806_sdk.git

4.3.1.2 准备工作

SmartConfig 示例工程的硬件准备有如下。

1. 评估板：运行示例工程代码。
2. 串口线：连接评估板的 UART0 插针，用于 console 控制台的输入输出。
3. PC 机：用于镜像烧录和 console 控制的输入输出。

SmartConfig 示例工程的软件准备，包括烧写工具、代码编译和烧写操作，请参见《XR806_SDK_快速入门指南》。

4.3.1.3 操作步骤

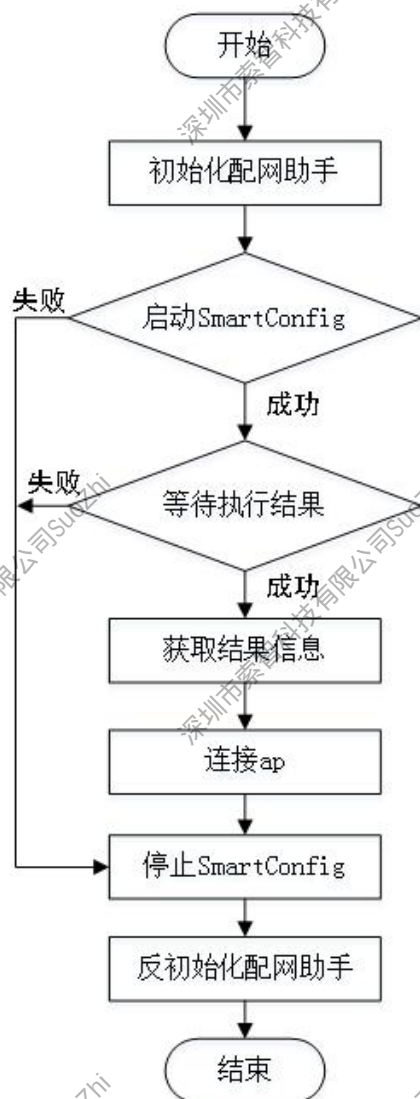
SmartConfig 示例工程无需控制命令，完成烧写后，复位即可，示例代码自动运行。配合手机端的 SmartConfig app 操作即可完成配网。

4.3.2 关键实现

系统平台的初始化在 XR806 SDK 的 `platform_init()` 函数中完成，具体请阅读此函数。本文重点描述 SmartConfig 模块启动、接收空中包并等待配网结果，最后完成联网的过程。

为了结合配网助手，xradi sdk 提供的配网流程和配网接口都使用配网助手进行了一定的封装。其中 SmartConfig 配网流程如下图：

图 4-1 SmartConfig 配网流程



具体操作见下面的示例代码。

```

#define SC_TIME_OUT 120000 //120s

static char sc_key[17] = "1234567812345678"; // smart config aes key

static void SmartConfigDemo(void)
{
    wlan_smart_config_status_t sc_status;
    wlan_smart_config_result_t sc_result;
    sc_assistant_fun_t sca_fun;
    sc_assistant_time_config_t config;

    memset(&sc_result, 0, sizeof(wlan_smart_config_result_t));
    
```

```

/*
 * Initialize the distribution network assistant,
 * which can improve the success rate of distribution network
 */
sc_assistant_get_fun(&sca_fun);
config.time_total = SC_TIME_OUT;
config.time_sw_ch_long = 400;
config.time_sw_ch_short = 100;
sc_assistant_init(g_wlan_netif, &sca_fun, &config);

/* start smart config */
sc_status = wlan_smart_config_start(g_wlan_netif, sc_key);
if (sc_status != WLAN_SMART_CONFIG_SUCCESS) {
    printf("smartconfig start failed!\n");
    goto out;
}

printf("%s getting ssid and psk...\n", __func__);

/* Blocking waiting for distribution network to complete */
if (wlan_smart_config_wait(SC_TIME_OUT) == WLAN_SMART_CONFIG_TIMEOUT) {
    printf("%s get ssid and psk timeout\n", __func__);
    goto out;
}
printf("%s get ssid and psk finished\n", __func__);

/* if complete, then get the smart config result */
if (wlan_smart_config_get_status() == SC_STATUS_COMPLETE && \
    smartconfig_get_result(&sc_result) == WLAN_SMART_CONFIG_SUCCESS) {
    printf("ssid:%s psk:%s random:%d\n", (char *)sc_result.ssid,
        (char *)sc_result.passphrase, sc_result.random_num);

    /* if get result, then connect ap and send the smart config connect ack */
    if (!wlan_smart_config_connect_ack(g_wlan_netif, SC_TIME_OUT, &sc_result)) {
        printf("connect and ack ok\n");
    }
}

out:
/* stop airkiss and deinit distribution network assistant */
wlan_smart_config_stop();
sc_assistant_deinit(g_wlan_netif);
}

```

4.3.3 效果展示

在评估板中运行 SmartConfig 示例程序后，使用手机 SmartConfig app 启动配网功能，在控制台中打印输出如下。

```
SmartConfig demo started.

en1: CTRL-EVENT-TERMINATING
WAR join_status:0

wlan information =====
firmware:
    version : R0-XR_C07.08.52.65_02.05 Nov 23 2020 17:14:24-Y02.05
    buffer   : 8
driver:
    version : XR_V02.05
mac address:
    in use   : 24:f4:69:40:9d:7b
    in use   : 24:f4:69:40:9d:7c
=====

SmartConfigDemo getting ssid and psk...
SmartConfigDemo get ssid and psk finished
ssid:My_ssid psk:My_password random:104

wlan information =====
firmware:
    version : R0-XR_C07.08.52.65_02.05 Nov 23 2020 17:14:24-Y02.05
    buffer   : 8
driver:
    version : XR_V02.05
mac address:
    in use   : 24:f4:69:40:9d:7b
    in use   : 24:f4:69:40:9d:7c
=====

[net INF] no need to switch wlan mode 0
en1: Trying to associate with ec:26:ca:df:66:66 (SSID='AW-ANC-TPLINK_WR847N#82' freq=2412 MHz)
[net INF] msg <wlan scan success>
en1: Associated with ec:26:ca:df:66:66
en1: WPA: Key negotiation completed with ec:26:ca:df:66:66 [PTK=CCMP GTK=CCMP]
en1: CTRL-EVENT-CONNECTED - Connection to ec:26:ca:df:66:66 completed [id=0 id_str=]
[net INF] msg <wlan connected>
[net INF] netif is link up
[net INF] start DHCP...
```

```

WAR drop=1099, fctl=0x00d0
[net INF] netif (IPv4) is up
[net INF] address: 192.168.51.21
[net INF] gateway: 192.168.51.1
[net INF] netmask: 255.255.255.0
[net INF] msg <network up>
[net INF] IPv6 addr state change: 0x0 --> 0x1
[net INF] msg <network IPv6 state>
connect and ack ok
[net INF] no need to switch wlan mode 0
    
```

SmartConfig demo over

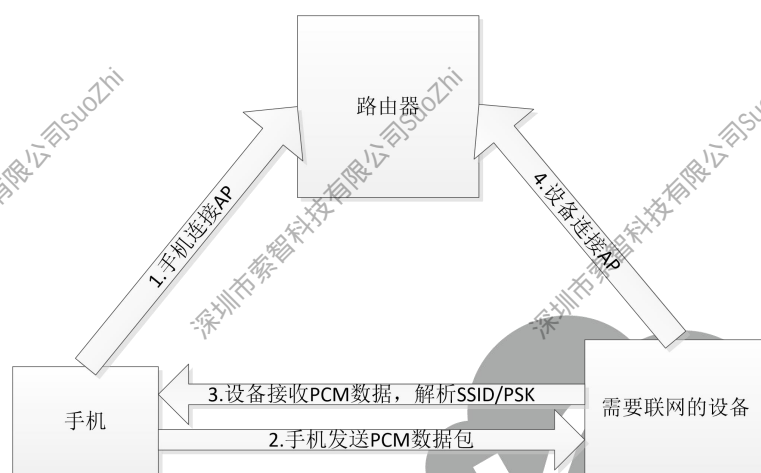
5 VoicePrint 配网

5.1 技术说明

5.1.1 配网原理

VoicePrint 是使用声波对 SSID 和 password 数据进行传输，WLAN 设备对收到的声波数据进行解码，解析出 SSID 和 password 数据，然后再去连接该 AP 即可。

图 5-1 VoicePrint 配网架构图



5.2 应用说明

5.2.1 应用简述

VoicePrint 配网方案已经在 XR806 SDK 中实现了，用户通过 VoicePrint 模块提供的接口即可开发出基于配网助手的 VoicePrint 配网功能。

5.2.2 配置说明

VoicePrint 配网方式的启用需要使能 SDK 的 WLAN 功能和声卡功能，具体配置项如下表所示。

表 5-1 VoicePrint 模块配置列表

配置项	配置说明
WLAN 功能使能	设置说明： 此项配置用于在 SDK 中启用 WLAN 功能。 设置方式： 1、在 prj_config.h 文件，添加或修改以下定义，其中 1 为使能 NET 功能，0 为不使能 NET 功能。 /* network and wlan enable/disable */ #define PRJCONF_NET_EN 1

配置项	配置说明
	2、在控制台执行命令“make menuconfig”，将“NET and WLAN”功能项打开。
声卡功能使能	设置说明： 此项配置用于在 SDK 中启用声卡功能。 设置方式： 1、在 prj_config.h 文件，添加或修改以下定义，其中 1 为使能声卡功能，0 为不使能声卡功能。 <pre>/* AC107 sound card enable/disable */ #define PRJCONF_AC107_SOUND_CARD_EN 1</pre>

5.2.3 接口说明

VoicePrint 模块总共提供 5 套接口，主要功能点涵括了进入配网、等待获取配网信息、获取模块运行状态、退出配网等过程。

5.2.3.1 voiceprint_start

表 5-2 voiceprint_start 接口函数说明

信息项	说明
原型	int voiceprint_start(voiceprint_param_t *param);
功能	启动声波配网
参数	voiceprint_param_t *param 含义解释：配网参数
返回值	0：成功 -1：失败

5.2.3.2 voiceprint_wait

表 5-3 voiceprint_wait 接口函数说明

信息项	说明
原型	voiceprint_ret_t voiceprint_wait(uint32_t timeout_ms);
功能	等待 VoicePrint 执行结束
参数	uint32_t timeout_ms 含义解释：等待超时时间
返回值	0：成功 -1：失败

5.2.3.3 voiceprint_get_status

表 5-4 voiceprint_get_status 接口函数说明

信息项	说明
原型	voiceprint_status_t voiceprint_get_status(void);
功能	获取配网的状态
参数	无
返回值	返回配网状态

5.2.3.4 wlan_voiceprint_get_raw_result

表 5-5 wlan_voiceprint_get_raw_result 接口函数说明

信息项	说明
原型	voiceprint_ret_t wlan_voiceprint_get_raw_result(char *result, int *len);
功能	获取配网结果的裸数据。该数据需要经过解析才能得到 ssid 和 password
参数	char *result 含义解释：裸数据的存放地址 int *len 含义解释：裸数据的长度
返回值	0：成功 -1：失败

5.2.3.5 voiceprint_stop

表 5-6 voiceprint_stop 接口函数说明

信息项	说明
原型	int voiceprint_stop(uint32_t wait);
功能	停止 VoicePrint 模块
参数	uint32_t wait 含义解释：是否等待 VoicePrint 线程退出
返回值	0：成功 -1：失败

5.3 示例说明

5.3.1 示例简介

VoicePrint 示例工程展示了 XR806 SDK 中使用 VoicePrint 方式进行配网的代码实现方法。

5.3.1.1 获取方式

VoicePrint 示例有示例工程代码，位于 XR806 SDK 的 `/project/example/VoicePrint` 目录，以下此示例工程简称为 VoicePrint 示例工程。



说明

XR806 SDK 可在以下 GitHub 仓库获取：https://github.com/XradioTech/xr806_sdk.git

5.3.1.2 准备工作

VoicePrint 示例工程的硬件准备有如下。

1. 评估板：运行示例工程代码。
2. 串口线：连接评估板的 UART0 插针，用于 console 控制台的输入输出。
3. PC 机：用于镜像烧录和 console 控制的输入输出。

VoicePrint 示例工程的软件准备，包括烧写工具、代码编译和烧写操作，请参见《XR806_SDK_快速入门指南》。

5.3.1.3 操作步骤

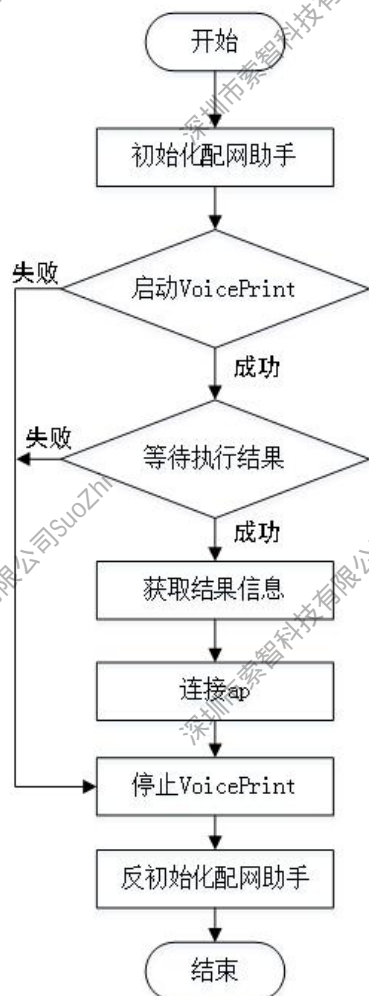
VoicePrint 示例工程无需控制命令，完成烧写后，复位即可，示例代码自动运行。配合手机端的 SoundConfig app 操作即可完成配网。

5.3.2 关键实现

系统平台的初始化在 XR806 SDK 的 `platform_init()` 函数中完成，具体请阅读此函数。本文重点描述 VoicePrint 模块启动、接收声波数据并解码获取配网结果，最后完成联网的过程。

为了结合配网助手，xradio sdk 提供的配网流程和配网接口都使用配网助手进行了一定的封装。其中 VoicePrint 配网流程如下图：

图 5-2 VoicePrint 配网流程



具体操作见下面的示例代码。

```

#define VP_TIME_OUT_MS 120000

int voice_print_example()
{
    int ret;
    int len;
    voiceprint_ret_t vp_ret;
    voiceprint_status_t status;
    wlan_voiceprint_result_t vp_result;
    voiceprint_param_t vp_param;
    sc_assistant_fun_t sca_fun;
    sc_assistant_time_config_t config;
    uint8_t *psk;
    char result[128];
}
    
```

```

printf("try to init sc_assistant.\n");
sc_assistant_get_fun(&sca_fun);
config.time_total = VP_TIME_OUT_MS;
config.time_sw_ch_long = 0;
config.time_sw_ch_short = 0;
ret = sc_assistant_init(g_wlan_netif, &sca_fun, &config);
if (ret) {
    printf("sc_assistant_init fail.\n");
    return -1;
}

printf("try to start voiceprint.\n");
vp_param.audio_card = AUDIO_SND_CARD_DEFAULT;
vp_param.nif = g_wlan_netif;
ret = voiceprint_start(&vp_param);
if (ret) {
    printf("voiceprint_start fail.\n");
    goto err0;
}

printf("waiting for result.\n");
vp_ret = voiceprint_wait(VP_TIME_OUT_MS);
if (vp_ret != WLAN_VOICEPRINT_SUCCESS) {
    printf("voiceprint_wait fail.\n");
    goto err1;
}

status = voiceprint_get_status();
if (status != VP_STATUS_COMPLETE) {
    printf("voiceprint status is not completed.\n");
    goto err1;
}

printf("try to get raw result.\n");
len = sizeof(result) - 1;
vp_ret = wlan_voiceprint_get_raw_result(result, &len);
if (vp_ret != WLAN_VOICEPRINT_SUCCESS) {
    printf("voiceprint get raw result fail.");
    goto err1;
}

printf("try to parser raw result.\n");
ret = vp_result_parser(result, len, &vp_result);
if (ret) {

```

```

        printf("parser raw result fail.\n");
        goto err1;
    }
    printf("result: [ssid:%s] [password:%s]\n", vp_result.ssid, vp_result.passphrase);

    g_wlan_netif = sc_assistant_open_sta();
    if (vp_result.passphrase[0] != '\0') {
        psk = vp_result.passphrase;
    } else {
        psk = NULL;
    }
    printf("try to connect ap.\n");
    ret = sc_assistant_connect_ap(vp_result.ssid, vp_result.ssid_len, psk, VP_TIME_OUT_MS);
    if (ret < 0) {
        printf("connect ap time out\n");
        goto err1;
    }

    return 0;

err1:
    voiceprint_stop(0);
err0:
    sc_assistant_deinit(g_wlan_netif);
    return -1;
}
    
```

5.3.3 效果展示

在评估板中运行 VoicePrint 示例程序后，使用手机 SoundConfig app 启动配网功能，在控制台中打印输出如下。

```

voice print start.
try to init sc_assistant.
en1: CTRL-EVENT-TERMINATING
WAR join_status:0

wlan information =====
firmware:
    version : R0-XR_C07.08.52.65_01.96 Oct 13 2020 11:45:57-Y01.96
    buffer   : 8
driver:
    version : XR_V02.05
mac address:
    
```

```

in use      : 84:92:34:40:93:3e
in use      : 84:92:34:40:93:3f
=====

try to start voiceprint.
waiting for result.
[AC107_CODEC] AMIC set volume Level-[1]
[AC107_CODEC] Route(cap): amic Enable
try to get raw result.
try to parser raw result.

wlan information =====
firmware:
  version : R0-XR_C07.08.52.65-01.96 Oct 13 2020 11:45:57-Y01.96
  buffer   : 8
driver:
  version : XR_V02.05
mac address:
  in use   : 84:92:34:40:93:3e
  in use   : 84:92:34:40:93:3f
=====

try to connect ap.
[net INF] no need to switch wlan mode 0
[net INF] msg <wlan scan success>
en1: Trying to associate with 6c:fa:89:65:fa:f2 (SSID='AWTest' freq=2437 MHz)
en1: Associated with 6c:fa:89:65:fa:f2
en1: WPA: Key negotiation completed with 6c:fa:89:65:fa:f2 [PTK=CCMP GTK=CCMP]
en1: CTRL-EVENT-CONNECTED - Connection to 6c:fa:89:65:fa:f2 completed [id=0 id_str=]
[net INF] msg <wlan connected>
[net INF] netif is link up
[net INF] start DHCP...
WAR drop=1099, fctl=0x00d0.
[net INF] netif is up
[net INF] address: 192.168.2.177
[net INF] gateway: 192.168.7.1
[net INF] netmask: 255.255.248.0
[net INF] msg <network up>
WAR drop=1099, fctl=0x00d0.

```

6 SoftAP 配网

6.1 技术说明

6.1.1 配网原理

SoftAP 配网是将设备切换为 AP 模式，然后创建一个 webserver，手机通过浏览器输入 AP 的 ip 地址来访问 webserver，手机浏览器通过 post 的方式将填写好的 SSID 和 password 发送到 webserver，设备接收到 SSID 和 password 后，切换为 STA 模式去连接网络。SoftAP 配网实现 webserver 的方式有两种：

1. 通过 socket 方式创建 webserver，socket 接收到浏览器发送的 request 后，解析 request，然后构建 response（response 中包含需要发送的 html 网页内容），发送 response 到浏览器中。该方式实现起来比较繁琐、复杂。
2. 通过 shhttpd 模块来创建 webserver，所有的解析和构建操作都有 shhttpd 来完成。该方式实现起来比较简单。

本文采用方式 1 来构建 webserver。

6.2 应用说明

6.2.1 应用简述

SoftAP 配网方案已经在 XR806 SDK 中实现了，用户通过 SoftAP 模块提供的接口即可开发出基于配网助手 SoftAP 配网功能。

6.2.2 配置说明

SoftAP 配网方式的启用需要使能 SDK 的 WLAN 功能，具体配置项如下表所示。

表 6-1 SoftAP 模块配置列表

配置项	配置说明
WLAN 功能使能	设置说明： 此项配置用于在 SDK 中启用 WLAN 功能。 设置方式： 1、在 prj_config.h 文件，添加或修改以下定义，其中 1 为使能 NET 功能，0 为不使能 NET 功能。 <pre>/* network and wlan enable/disable */ #define PRJCONF_NET_EN 1</pre> 2、在控制台执行命令“make menuconfig”，将“NET and WLAN”功能项打开。

6.2.3 接口说明

SoftAP 模块总共提供 5 套接口，主要功能点涵括了进入配网、等待获取配网信息、获取模块运行状态、退出配网等过程。

SoftAP 配网模块使用到的结构体主要有以下两个，分别用来记录配网的结果信息和配网状态。

配网结果信息：

```
typedef struct {
    char ssid[32]; //获取到的 SSID
    char psk[32]; //获取到的 PSK
} soft_ap_config_result; //配网结果结构体
```

配网状态：

```
typedef enum {
    SOFT_AP_CONFIG_STOP, //配网停止
    SOFT_AP_CONFIG_START, //配网开始
    SOFT_AP_CONFIG_COMPLETE, //配网完成
} SOFT_AP_CONFIG_STA; //配网的状态
```

6.2.3.1 soft_ap_config_start

表 6-2 soft_ap_config_start 接口函数说明

信息项	说明
原型	int soft_ap_config_start(void);
功能	启动 SoftAP 配网
参数	无
返回值	0：成功 -1：失败

6.2.3.2 soft_ap_config_stop

表 6-3 soft_ap_config_stop 接口函数说明

信息项	说明
原型	int soft_ap_config_stop(void);
功能	停止 softap 配网
参数	无
返回值	0：成功 -1：失败

6.2.3.3 soft_ap_config_get_state

表 6-4 soft_ap_config_get_state 接口函数说明

信息项	说明
原型	int soft_ap_config_get_state(void);
功能	获取配网的状态
参数	无
返回值	返回配网状态

6.2.3.4 soft_ap_config_get_result

表 6-5 soft_ap_config_get_result 接口函数说明

信息项	说明
原型	int soft_ap_config_get_result(soft_ap_config_result *result);
功能	获取配网的结果，只有在配网完成后，才能获取，否则返回-1
参数	soft_ap_config_result *result 含义解释：配网结果结构体的指针
返回值	0：成功 -1：失败

6.2.3.5 soft_ap_config_set_cb

表 6-6 soft_ap_config_set_cb 接口函数说明

信息项	说明
原型	int soft_ap_config_set_cb(soft_ap_config_cb cb);
功能	设置配网的回调函数，配网完成后，会调用该回调函数
参数	soft_ap_config_cb cb 含义解释：回调函数指针
返回值	0：成功 -1：失败

6.3 示例说明

6.3.1 示例简介

Soft_ap_config 示例工程展示了 XR806 SDK 中使用 SoftAP 方式进行配网的代码实现方法。

6.3.1.1 获取方式

Soft_ap_config 示例有示例工程代码，位于 XR806 SDK 的 /project/example/soft_ap_config 目录，以下此示例工程简称为 soft_ap_config 示例工程。



说明

XR806 SDK 可在以下 GitHub 仓库获取：https://github.com/XradioTech/xr806_sdk.git

6.3.1.2 准备工作

Soft_ap_config 示例工程的硬件准备有如下。

1. 评估板：运行示例工程代码。
2. 串口线：连接评估板的 UART0 插针，用于 console 控制台的输入输出。
3. PC 机：用于镜像烧录和 console 控制的输入输出。

Soft_ap_config 示例工程的软件准备，包括烧写工具、代码编译和烧写操作，请参见《XR806_SDK_快速入门指南》。

6.3.1.3 操作步骤

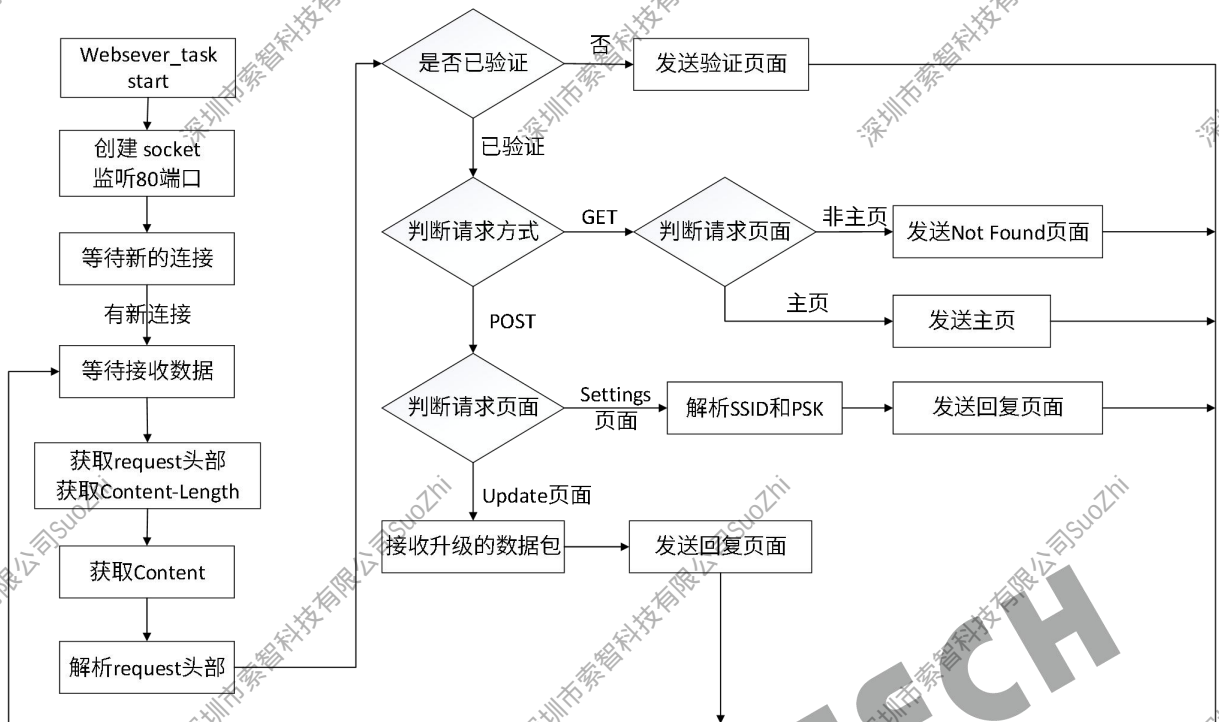
Soft_ap_config 示例工程无需控制命令，完成烧写后，复位即可，示例代码自动运行。配合手机端的网页操作即可完成配网。

6.3.2 关键实现

系统平台的初始化在 XR806 SDK 的 platform_init() 函数中完成，具体请阅读此函数。本文重点描述 softAP 配网模块启动、接收 AP 信息并获取配网结果的过程。

SoftAP 配网基本流程就是：创建 socket->监听端口->等待连接->获取客户端请求->解析请求->根据请求执行不同的操作并返回不同的页面。

图 6-1 SoftAP 配网流程



具体操作见下面的示例代码。

```

static char *softap_ssid = "XRADIO_SOFT_AP_CONFIG_TEST";
static int net_state = -1;
static soft_ap_config_result soft_ap_result;
static SOFT_AP_CONFIG_STA soft_ap_state;

static void net_cb(uint32_t event, uint32_t data, void *arg)
{
    uint16_t type = EVENT_SUBTYPE(event);
    net_state = type;
}

static void soft_ap_config_callback(soft_ap_config_result *result,
                                   SOFT_AP_CONFIG_STA state)
{
    /* copy the result and state */
    memcpy(&soft_ap_result, result, sizeof(soft_ap_result));
    soft_ap_state = state;

    printf("ssid:%s psk:%s state:%d\n", result->ssid, result->psk, state);
}

int main(void)

```

```

{
    int soft_ap_has_start = 0;

    platform_init();

    /* create an observer to monitor the net work state */
    static observer_base *net_ob;
    net_ob = sys_callback_observer_create(CTRL_MSG_TYPE_NETWORK,
                                          NET_CTRL_MSG_ALL,
                                          net_cb,
                                          NULL);

    if(net_ob == NULL)
        return -1;
    if(sys_ctrl_attach(net_ob) != 0)
        return -1;

    /* set to ap mode */
    net_switch_mode(WLAN_MODE_HOSTAP);
    wlan_ap_disable();
    wlan_ap_set((unsigned char*)softap_ssid, strlen(softap_ssid), NULL);
    wlan_ap_enable();

    /* set soft_ap_config callback */
    soft_ap_config_set_cb(soft_ap_config_callback);
    while (1) {
        if (net_state == NET_CTRL_MSG_NETWORK_UP && !soft_ap_has_start) {
            /* if the network is up, start the soft_ap_config */
            soft_ap_config_start();
            soft_ap_has_start = 1;
        }

        OS_MSleep(100);
    }

    return 0;
}

```

6.3.3 效果展示

在评估板中运行 Soft_ap_config 示例程序后，使用手机连接 softAP 后，打开浏览器网页功能，输入需要的 SSID 和 password 后，在控制台中打印输出如下。

```
platform information =====
```

XR806 SDK v0.2.0 Nov 24 2020 19:06:05

heap space [0x2185e0, 0x24bc00), size 210464

cpu clock 160000000 Hz

HF clock 40000000 Hz

sdk option:

XIP : enable

INT LF OSC : enable

SIP flash : enable

mac address:

efuse : 00:00:00:00:00:00

in use : 70:32:16:fa:de:c2

en1: CTRL-EVENT-TERMINATING

WAR join_status:0

wlan information

firmware:

version : R0-XR_C07.08.52.65_02.05 Nov 17 2020 14:44:43-Y02.05

buffer : 8

driver:

version : XR_V02.05

mac address:

in use : 70:32:16:fa:de:c2

in use : 70:32:16:fa:de:c3

interface name: en1

Using interface en1 with hwaddr 70:32:16:fa:de:c2 and ssid "AP-XRADIO"

en1: interface state UNINITIALIZED->ENABLED

en1: AP-ENABLED

en1: AP-DISABLED

[net INF] msg <wlan connected>

[net INF] netif is link up

[net INF] bring up netif

[net INF] netif is up

[net INF] address: 192.168.51.1

[net INF] gateway: 192.168.51.1

[net INF] netmask: 255.255.255.0

vif0, AP/GO mode THROTTLE=38

```
[net INF] msg <network up>
en1: interface state ENABLED->DISABLED
[net INF] msg <wlan disconnected>
[net INF] netif is link down
Using interface en1 with hwaddr 70:32:16:fa:de:c2 and ssid "XRADIO_SOFT_AP_CONFIG_TEST"
en1: interface state DISABLED->ENABLED
en1: AP-ENABLED
[net INF] msg <wlan connected>
[net INF] netif is link up
[net INF] netif is already up
[net INF] netif is up
[net INF] address: 192.168.51.1
[net INF] gateway: 192.168.51.1
[net INF] netmask: 255.255.255.0
[net INF] msg <network up>
vif0, AP/GO mode THROTTLE=38

xr_newassoc(): link_id:0 mac:64:09:80:d3:4f:ba
en1: STA 64:09:80:d3:4f:ba IEEE 802.11: associated
en1: AP-STA-CONNECTED 64:09:80:d3:4f:ba
ssid:My_ssid psk:My_password state:2
```

7 Sc_assistant 配网助手

7.1 技术说明

由于多种配网方式有可能会被同时启动，为了兼容这种情况，需要一个额外的模块对各种配网方式进行管理，于是便引入了配网助手模块 `sc_assistant`。

配网助手 `sc_assistant` 并不是一种配网技术，而是用于协调各个配网方式的运行，对整个配网功能起到一种优化的作用。

使用 `sc_assistant` 模块，对比普通的配网流程有较大优势，如下列出：

1. 可同时使用多种配网方式，兼容各个配网流程；
2. 优化了切换信道的时机，增大配网成功的概率；

7.2 应用说明

7.2.1 应用简述

`assistant` 配网助手已经在 XR806 SDK 中实现了，用户通过 `assistant` 模块提供的接口即可开发出基于配网助手的配网功能。

7.2.2 接口说明

7.2.2.1 `sc_assistant_get_fun`

表 7-1 `sc_assistant_get_fun` 接口函数说明

信息项	说明
原型	<code>void sc_assistant_get_fun(sc_assistant_fun_t *fun);</code>
功能	获取配网助手操作函数
参数	<code>sc_assistant_fun_t *fun</code> 含义解释：获取到的操作接口
返回值	无

7.2.2.2 `sc_assistant_init`

表 7-2 `sc_assistant_init` 接口函数说明

信息项	说明
原型	<code>int sc_assistant_init(struct netif *nif, sc_assistant_fun_t *cb, sc_assistant_time_config_t *config);</code>
功能	初始化配网助手
参数	<code>struct netif *nif</code> 含义解释：网络接口 <code>sc_assistant_fun_t *cb</code>

信息项	说明
	含义解释：配网助手的操作函数 sc_assistant_time_config_t *config 含义解释：配置参数
返回值	0：成功 -1：失败

7.2.2.3 sc_assistant_deinit

表 7-3 sc_assistant_deinit 接口函数说明

信息项	说明
原型	int sc_assistant_deinit(struct netif *nif);
功能	反初始化配网助手
参数	struct netif *nif 含义解释：网络接口
返回值	0：成功 -1：失败

7.2.2.4 sc_assistant_monitor_register_rx_cb

表 7-4 sc_assistant_monitor_register_rx_cb 接口函数说明

信息项	说明
原型	int sc_assistant_monitor_register_rx_cb(struct netif *nif, wlan_monitor_rx_cb cb);
功能	注册一个 rx 回调函数，可以同时注册多个
参数	struct netif *nif 含义解释：网络接口 wlan_monitor_rx_cb cb 含义解释：rx 回调函数
返回值	0：成功 -1：失败

7.2.2.5 sc_assistant_monitor_unregister_rx_cb

表 7-5 sc_assistant_monitor_unregister_rx_cb 接口函数说明

信息项	说明
原型	int sc_assistant_monitor_unregister_rx_cb(struct netif *nif, wlan_monitor_rx_cb cb);
功能	释放一个 rx 回调函数，需要指明释放的函数名
参数	struct netif *nif 含义解释：网络接口 wlan_monitor_rx_cb cb

信息项	说明
	含义解释：需要被释放的 rx 回调函数
返回值	0：成功 -1：失败

7.2.2.6 sc_assistant_open_monitor

表 7-6 sc_assistant_open_monitor 接口函数说明

信息项	说明
原型	void sc_assistant_open_monitor(void);
功能	启动 monitor 模式
参数	无
返回值	无

7.2.2.7 sc_assistant_open_sta

表 7-7 sc_assistant_open_sta 接口函数说明

信息项	说明
原型	struct netif *sc_assistant_open_sta(void);
功能	启动 STA 模式
参数	无
返回值	指针网络接口

7.2.2.8 sc_assistant_get_status

表 7-8 sc_assistant_get_status 接口函数说明

信息项	说明
原型	sc_assistant_status sc_assistant_get_status(void);
功能	获取配网状态
参数	无
返回值	配网状态值，具体值列举如下： SCA_STATUS_END：配网初始和结束的状态 SCA_STATUS_CONTINUE：配网扫描信道状态 SCA_STATUS_CHANNEL_LOCKED：配网锁定信道的状态 SCA_STATUS_COMPLETE：配网获取到 SSID 和 PSK 的状态 SCA_STATUS_CRC_ERROR：配网获取的数据校验出错的状态

7.2.2.9 sc_assistant_newstatus

表 7-9 sc_assistant_newstatus 接口函数说明

信息项	说明
原型	extern void sc_assistant_newstatus(sc_assistant_status status, uint8_t *ap_mac, void *info);
功能	更新配网状态
参数	sc_assistant_status status 含义解释：更新后的配网状态 uint8_t *ap_mac 含义解释：触发锁定信道的 AP 的 MAC 地址 void *info 含义解释：触发锁定信道的帧信息
返回值	无

7.3 示例说明

请参考 3.3 章节 airkiss 示例工程的相关说明。

著作权声明

版权所有©2020 广州芯之联科技有限公司。保留一切权利。

本文档及内容受著作权法保护，其著作权由广州芯之联科技有限公司（“芯之联”）拥有并保留一切权利。

本文档是芯之联的原创作品和版权财产，未经芯之联书面许可，任何单位和个人不得擅自摘抄、复制、修改、发表或传播本文档内容的部分或全部，且不得以任何形式传播。

商标声明



（不完全列举）均为广州芯之联科技有限公司的商标或者注册商标。在本文档描述的产品中出现的其它商标，产品名称，和服务名称，均由其各自所有人拥有。

免责声明

您购买的产品、服务或特性应受您与广州芯之联科技有限公司（“芯之联”）之间签署的商业合同和条款的约束。本文档中描述的全部或部分产品、服务或特性可能不在您所购买或使用的范围内。使用前请认真阅读合同条款和相关说明，并严格遵循本文档的使用说明。您将自行承担任何不当使用行为（包括但不限于如超压，超频，超温使用）造成的不利后果，芯之联概不负责。

本文档作为使用指导仅供参考。由于产品版本升级或其他原因，本文档内容有可能修改，如有变更，恕不另行通知。芯之联尽全力在本文档中提供准确的信息，但并不确保内容完全没有错误，因使用本文档而发生损害（包括但不限于间接的、偶然的、特殊的损失）或发生侵犯第三方权利事件，芯之联概不负责。本文档中的所有陈述、信息和建议并不构成任何明示或暗示的保证或承诺。

本文档未以明示或暗示或其他方式授予芯之联的任何专利或知识产权。在您实施方案或使用产品的过程中，可能需要获得第三方的权利许可。请您自行向第三方权利人获取相关的许可。芯之联不承担也不代为支付任何关于获取第三方许可的许可费或版税（专利税）。芯之联不对您所使用的第三方许可技术做出任何保证、赔偿或承担其他义务。