



# **D1-H Tina Linux 配网 开发指南**

**版本号: 1.0**  
**发布日期: 2021.04.06**

## 版本历史

版本号	日期	制/修订人	内容描述
1.0	2021.04.06	AWA1381	1. 建立初始版本。



# 目 录

<b>1 概述</b>	<b>1</b>
1.1 编写目的	1
1.2 适用范围	1
1.3 相关人员	1
<b>2 smartlink</b>	<b>2</b>
2.1 sdk 代码目录	2
2.2 编译配置	3
2.3 APP 编写说明	3
2.3.1 准备	3
2.3.2 API	3
2.4 demo 使用说明	5
<b>3 测试说明</b>	<b>7</b>
3.1 smartlink 联网测试	7
3.2 soudwave 配网测试	8
3.3 softap 配网	9

# 1 概述

---

## 1.1 编写目的

介绍 Allwinner 平台上 Wi-Fi 和 BT 的配网方式，包括 softap(Wi-Fi ap 模式热点配网),smartlink (透传配网) ,soundwave(声波配网),airkiss(微信配网),BLE(蓝牙低功耗配网)。

## 1.2 适用范围

Allwinner 软件平台 Tina linux。

Allwinner 硬件平台 D1-H。

## 1.3 相关人员

适用 Tina 平台的广大客户和对 Tina 网络感兴趣的开发者。

## 2 smartlink

D1-H 平台中目前支持的 Wi-Fi 模组只有全志 Xradio(xr829) 模组。支持的配网方式有 airkiss(微信)、soundwave (声波)、softap (热点)、smartconfig、BLE(蓝牙低功耗配网) 以及蓝牙 BLE 配网。

### 2.1 sdk 代码目录

源码路径为 tina/package/allwinner/smartlinkd。包括 smartlink 服务进程、配网协议和 demo 程序。

```

|— demo                #存放demo
|— files                #配置文件
|   |— smartlinkd.init  #开机自启动配置文件
|   |— usrapp           #手机APP目录
|       |— README
|       |— source        #APP源码，详见后说明
|       |— testapp       #Android 测试apk，详见后说明
|— Makefile
|— README
|— src
|— sm_link_manager.c
|— smg_log.c
|— smg_event.c
|— protocol
|   |— dependent         #模组相关协议
|       |— broadcom      #broadcom模组底层协议
|       |— realtek       #Realtek模组底层协议
|       |— xradio        #xr819/xr829模组底层协议
|   |— independent      #模组无关协议（适用于所有模组）
|       |— softAP        #热点配网协议
|       |— soundwave     #声波配网协议

```

source目录下

```

.
|— CooeeDemoAndroid-v1.8.0.zip  #broadcom cooee协议Android端源码
|— CooeeDemoIOS-v1.5.0.zip     #broadcom cooee 协议IOS端源码
|— SoundWaveAndroid.zip        #声波Android端源码
|— SoftApAndroid.zip           #Softap Android端源码
|— XConfig.zip                 #XR819 smartconfig协议Android 端源码

```

test目录下的apk，安装到Android手机中可直接使用，方便测试

```

.
|— AirKissDebugger.apk         #airkiss apk, xr819与broadcom通用
|— CooeeDemo.apk               #broadcom cooee Android端测试apk
|— smatconfig.apk              #XR819 smartconfig Android端测试apk
|— softAP.apk                  #softap Android端测试apk
|— soundwave.apk               #声波配网 Android端测试apk

```

## 2.2 编译配置

```
make menuconfig
  Allwinner --->
    smartlinkd --->
      <*> smartlinkd-demo..... smartlink demo
      -* smartlinkd-lib..... smartlinkd-lib
        [*] enable xradio airkiss support for lib
        [*] enable xradio smartconfig support for lib
        [*] enable soft ap support for lib
        [*] enable sound wave support for lib
```

- smartlinkd-demo: smartlinkd 实例程序。
- smartlinkd-lib: demo 所使用的声波配网协议库。

## 2.3 APP 编写说明

### 2.3.1 准备

(1) 导入接口头文件

```
#include "sm_link_manager.h"
```

(2) 链接动态库

```
libsm_mg.so
```

(3) 示例代码

```
<tina>/package/allwinner/smartlinkd/demo/main.c
```

### 2.3.2 API

- sm\_link\_init
- 【函数原型】：int sm\_link\_init(int protocol\_num);
- 【功能描述】：初始化 smartlink，注意是分配资源。
- 【参数说明】：
  1. protocol\_num: 需要启动的配网协议数量，支持同时启动多个协议，前提是模组能够支持多种模式共存。
- 【返回说明】：0: 初始化成功；非 0: 初始化失败。

- `sm_link_protocol_enable`
- 【函数原型】： `int sm_link_protocol_enable(int type, struct proto_params *p, int protocol_num);`
- 【功能描述】： 协议使能。
- 【参数说明】：
  1. `type`: 需要启动的协议类型。
  2. `p`： 启动的协议需要传递的参数。
  3. `protocol_num`: 启动的协议数量。
- 【返回说明】： 0: 成功; 非 0: 初始化失败。
- `sm_link_wait_get_results`
- 【函数原型】： `int sm_link_wait_get_results(int type, struct net_info *info);`
- 【功能描述】： 获取协议使能结果。
- 【参数说明】：
  1. `type`: 需要启动的协议类型。
  2. `info`: 存放返回的网络信息。
- 【返回说明】： 0: 成功; 小于 0: 初始化失败。
- `sm_link_deinit`
- 【函数原型】： `int sm_link_deinit();`
- 【功能描述】： smartlink 资源销毁，配合 `sm_link_init` 使用。
- 【参数说明】： 无。
- 【返回说明】： 0: 成功; 小于 0: 初始化失败。

关于启动多个协议：目前 tina smartlink 同时支持启动多个 protocol，前提是 WiFi 模组能够支持多种模式共存。声波能配合其他协议同时启动。如同时启动 softap 和 soundwave，或者同时启动 airkiss 和 soundwave。另外，需要注意的时，在同时启动多个协议的时候，struct proto\_params 参数需要按照指定顺序赋值。以下是协议的顺序。

```
SM_LINK_AP_COOEE->
SM_LINK_AP_NEEZE->
SM_LINK_AP_AIRKISS->
SM_LINK_XR_AIRKISS->
SM_LINK_XR_SMARTCONFIG->
SM_LINK_SOUND_WAVE->
SM_LINK_SOFTAP->
.....
```

例子：同时启动 xradio\_airkiss 和 soundwave

```
void *xr_airkiss_argv[3] = {
    "60",          //等待超时时间
    "1234567890123456", //AES KEY
}
```

```

    "wlan0" //使用的网卡设备
};

void *sound_wave_argv[4] = {
    "default", //录音声卡设备
    "0",       //频率类型: 0-LOW, 1-MID , 2-HIGH
    "16000",   //录音采样率
    "60",      //等待超时时间
};

#define WORK_PROTOCOL_NUM 2
#define COM_PROTOCOL      (SM_LINK_XR_AIRKISS|SM_LINK_SOUND_WAVE)
struct proto_params params_com[WORK_PROTOCOL_NUM];
params_com[0].argv = xr_airkiss_argv; //第一个参数必须为xradio airkiss的参数
params_com[1].argv = sound_wave_argv;

```

例子：同时启动 xradio\_airkiss 和 softap

```

void *xr_airkiss_argv[3] = {
    "60", //等待超时时间
    "1234567890123456", //AES KEY
    "wlan0" //使用的网卡设备
};

#define WORK_PROTOCOL_NUM 2
#define COM_PROTOCOL      (SM_LINK_XR_AIRKISS|SM_LINK_SOFTAP)
struct proto_params params_com[WORK_PROTOCOL_NUM];
params_com[0].argv = xr_airkiss_argv; //第一个参数必须为xradio airkiss的参数
params_com[1].argv = NULL;           //softap不需要传参数

```

例子：同时启动 soundwave 和 softap

```

void *sound_wave_argv[4] = {
    "default", //录音声卡设备
    "0",       //频率类型: 0-LOW, 1-MID , 2-HIGH
    "16000",   //录音采样率
    "60",      //等待超时时间
};

#define WORK_PROTOCOL_NUM 2
#define COM_PROTOCOL      (SM_LINK_SOFTAP|SM_LINK_SOUND_WAVE)
struct proto_params params_com[WORK_PROTOCOL_NUM];
params_com[0].argv = sound_wave_argv; //第一个参数必须为声波的参数
params_com[1].argv = NULL;           //softap不需要传参数

```

## 2.4 demo 使用说明

```

example: smartlink_demo V

V:take the following value
    0-softap
    1-soundwave
    2-xradio smartconfig
    3-xradio airkiss

```



4-ampark (ap6212...) airkiss  
5-ampark (ap6212...) cooee  
6-composite (softap & soudwave)



## 3 测试说明

### 3.1 smartlink 联网测试

说明：只有 XRADIO 模组支持。

- 1.xr829 板子通过串口连接 PC 与开发板，系统起来，进入 Linux shell；
2. 执行：Smartlink\_demo 2
3. 手机提前连接上想要板子连接的 WiFi.
4. 手机打开 xconfig，发送 ssid 和 passwd 给板子.
5. 板子收到 ssid 和 passwd 后去联网.
6. 板子上执行 ping 操作测试.

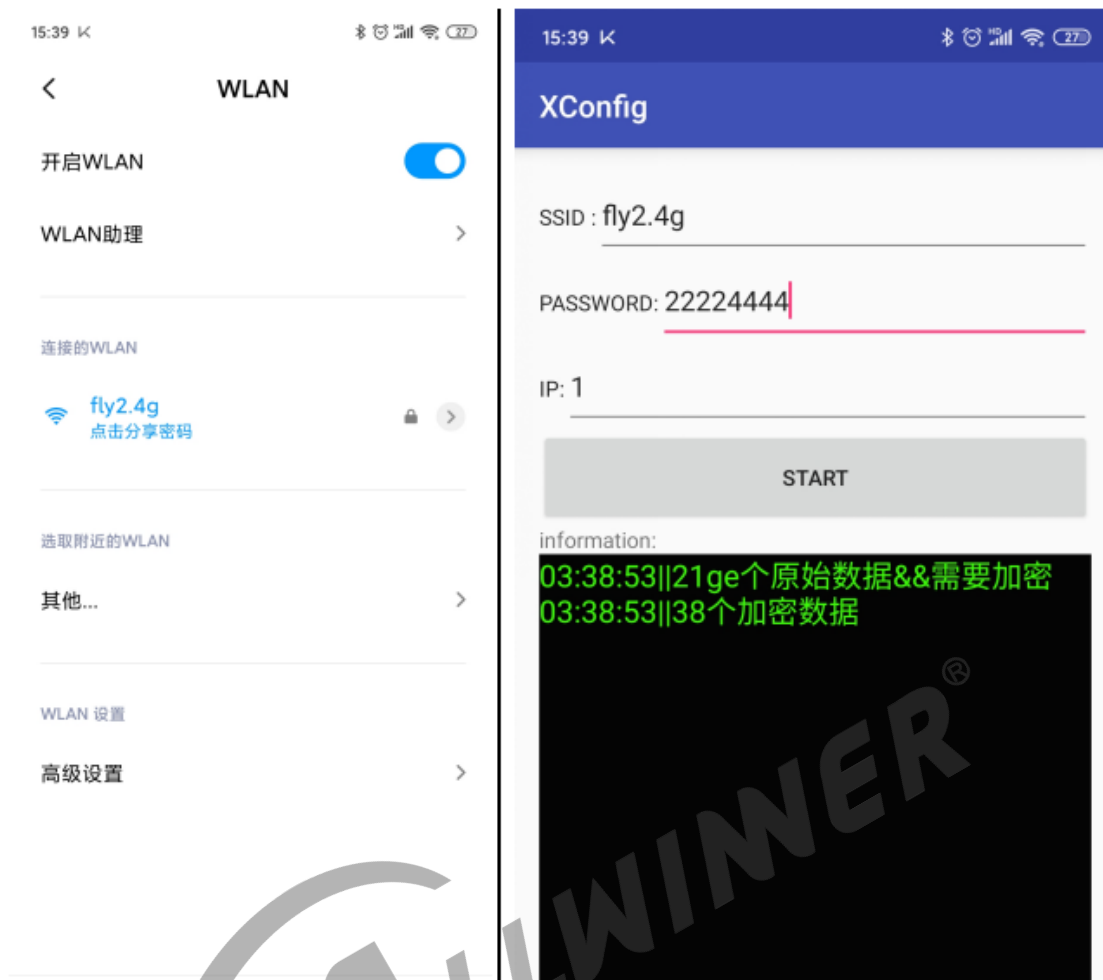


图 3-1: smartlink 配网

## 3.2 soudwave 配网测试

1. 板子通过串口连接 PC 与开发板，系统起来，进入 Linux shell；
2. 执行：smartlink\_demo 1。
3. 手机打开 SoundAuthenticationTest app 发送想要板子连接的 ssid 和 passwd。
4. 板子收到 ssid 和 passwd 后去联网。
5. 执行 ping 测试。

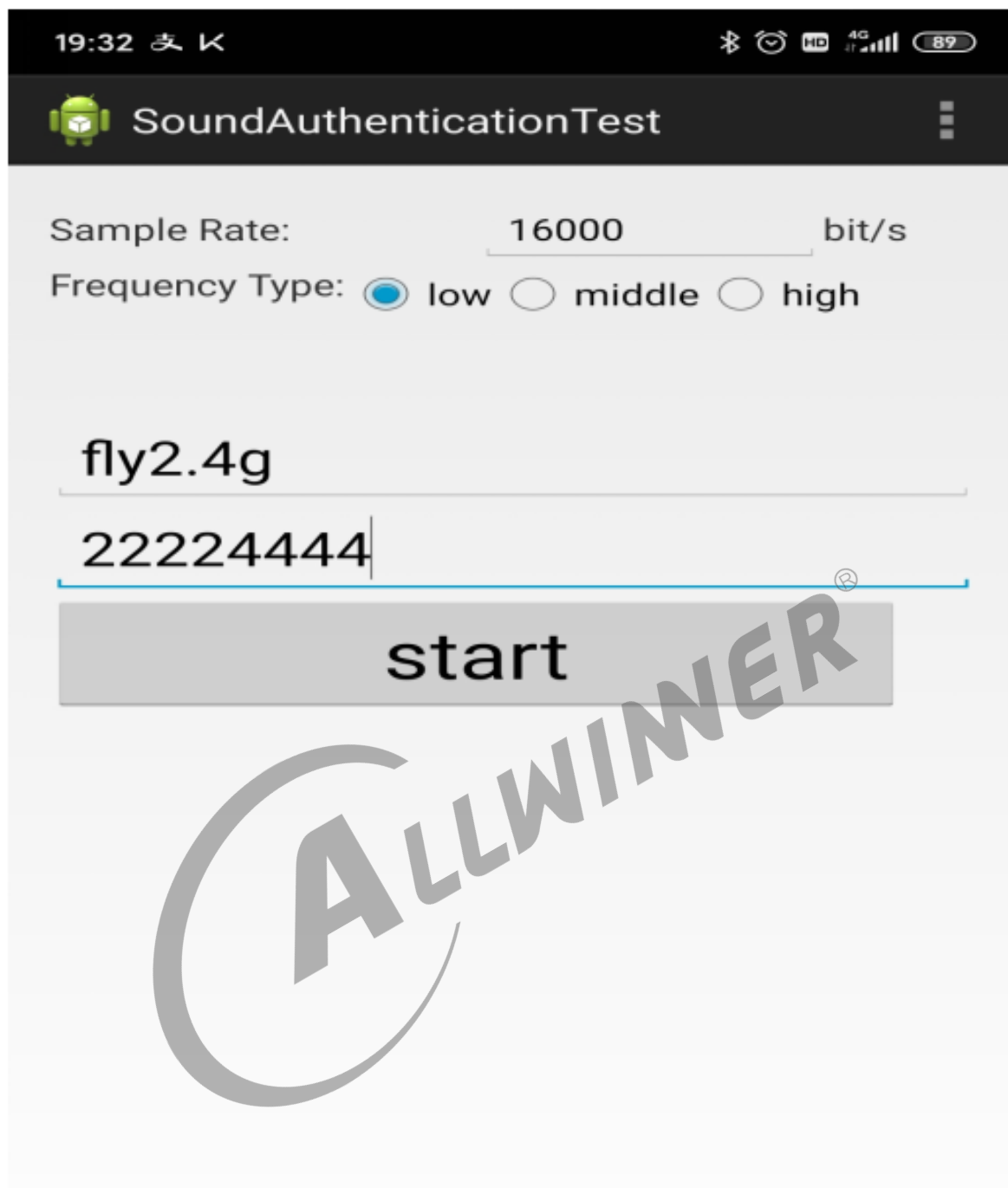


图 3-2: soundwave 配网

### 3.3 softap 配网

1. 板子通过串口连接 PC 与开发板，系统起来，进入 Linux shell;
2. 执行：smartlink\_demo 0
3. 此时手机可以扫描到 aw\_smartlink\_softap 热点，手机连接上。

4. 手机利用 ckysoftAPDemo 发送想要板子连接的 ssid 和 passwd。
5. 板子收到 ssid 和 passwd 后会退出当前 smartlink\_demo，然后就可以执行 ping 测试了。

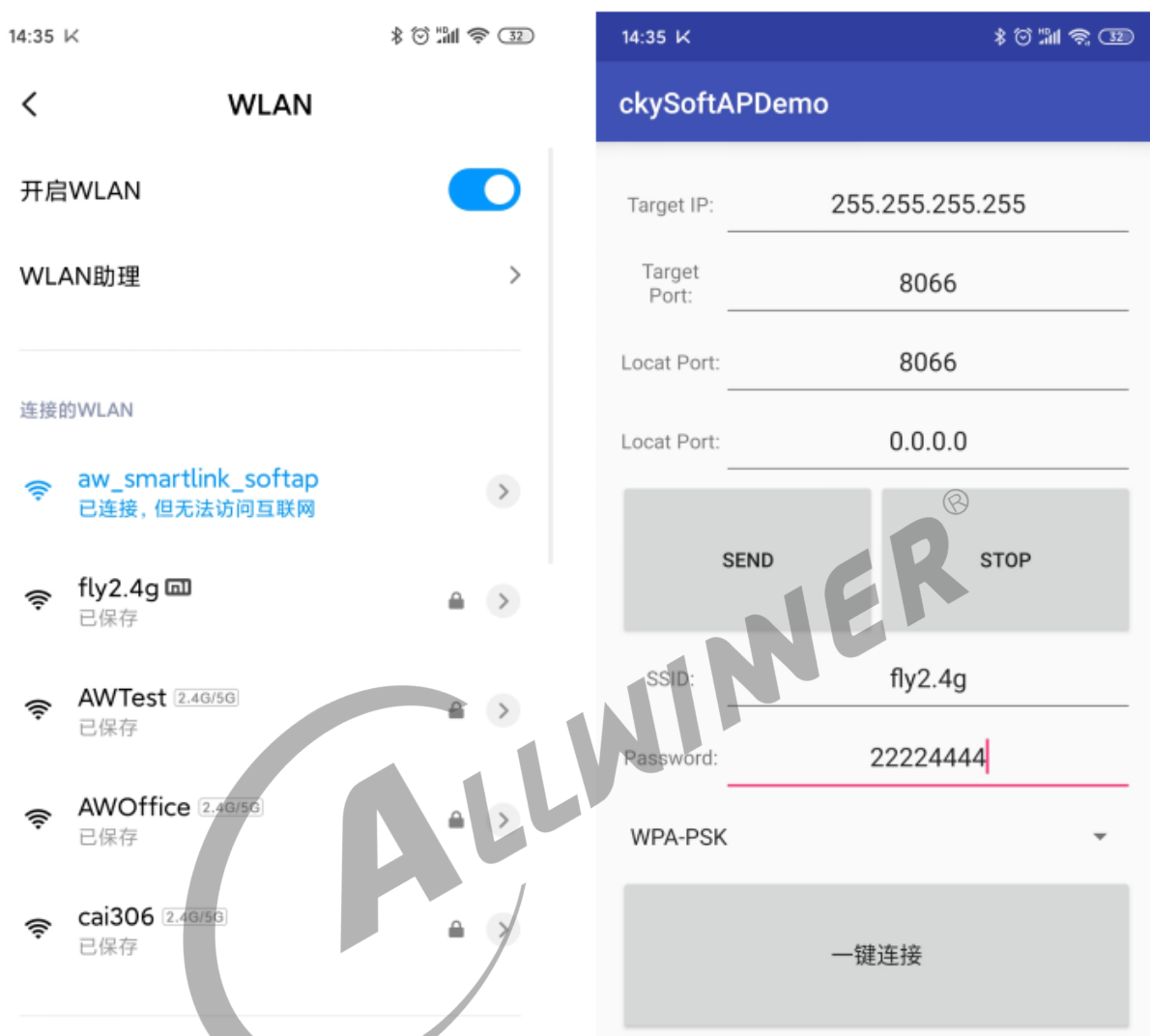


图 3-3: softap 配网




## 著作权声明

版权所有 © 2022 珠海全志科技股份有限公司。保留一切权利。

本文档及内容受著作权法保护，其著作权由珠海全志科技股份有限公司（“全志”）拥有并保留一切权利。

本文档是全志的原创作品和版权财产，未经全志书面许可，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

、 **全志科技** （不完全列举）均为珠海全志科技股份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珠海全志科技股份有限公司（“全志”）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，全志概不负责。

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。全志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，全志概不负责。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全志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。全志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）。全志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。