

Tina

Linux smartlink 使用说明文档 v1.5

文档履历

版本号	日期	制/修订人	制/修订记录
V1.0	2016/02/02		初始版本
V1.1	2016/02/14		修改 api 名称
V1.2	2016/03/30		增加 stop api, 使用注意
V1.3.1	2016/11/02		增加声波配网 api; 适配 R40 项目
V1.3.2	2016/11/14		对 V1.3.1 版本部分细节进行补充修改
V1.4	2017/05/12		适配 Tina2.0 平台
V1.5	2018/03/27		增加各协议单独收发 demo(包括手机端源码、测试 apk); 增加 xr819 airkiss、smartconfig 支持; 完善声波配网功能。



目 录

1. 概述.....	4
1.1. 编写目的.....	4
1.2. 适用范围.....	4
1.3. 相关人员.....	4
2. Smartlink 简介.....	5
2.1. Sdk 代码目录.....	5
2.2. 编译配置.....	6
2.3. APP 编写说明.....	6
APP demo 代码目录为:	7
2.4. 启动 smartlinkd server.....	7
3. API.....	7
3.1. 数据结构.....	7
3.2. 检测&准备环境.....	7
3.3. 初始化.....	7
3.4. 消息回调.....	8
3.5. 启动 airkiss.....	8
3.6. 停止 airkiss.....	8
3.7. 启动 cooee.....	8
3.8. 停止 cooee.....	8
3.9. 启动声波联网.....	9
3.10. 停止声波联网.....	9
3.11. 启动 smartconfig 联网.....	9
3.12. 停止 smartconfig 联网.....	9
3.13. 混合启动模式.....	9
3.14. 混合停止模式.....	9
4. 调试说明.....	11
4.1. 单独测试协议接收.....	11
4.1.1. Broadcom 模组.....	11
4.1.2. XR819 模组.....	11
4.1.3. 声波.....	12
4.2. 使用 smartlinkd_demo.....	12
5. Declaration.....	13

1. 概述

1.1. 编写目的

介绍 Tina smartlink 应用开发接口和 Demo 代码。

1.2. 适用范围

Allwinner 软件平台 Tina v2.0 以上
Allwinner 硬件平台 R6 R11 R16 R18 R30 R40

1.3. 相关人员

适用相关模块开发以及维护人员。

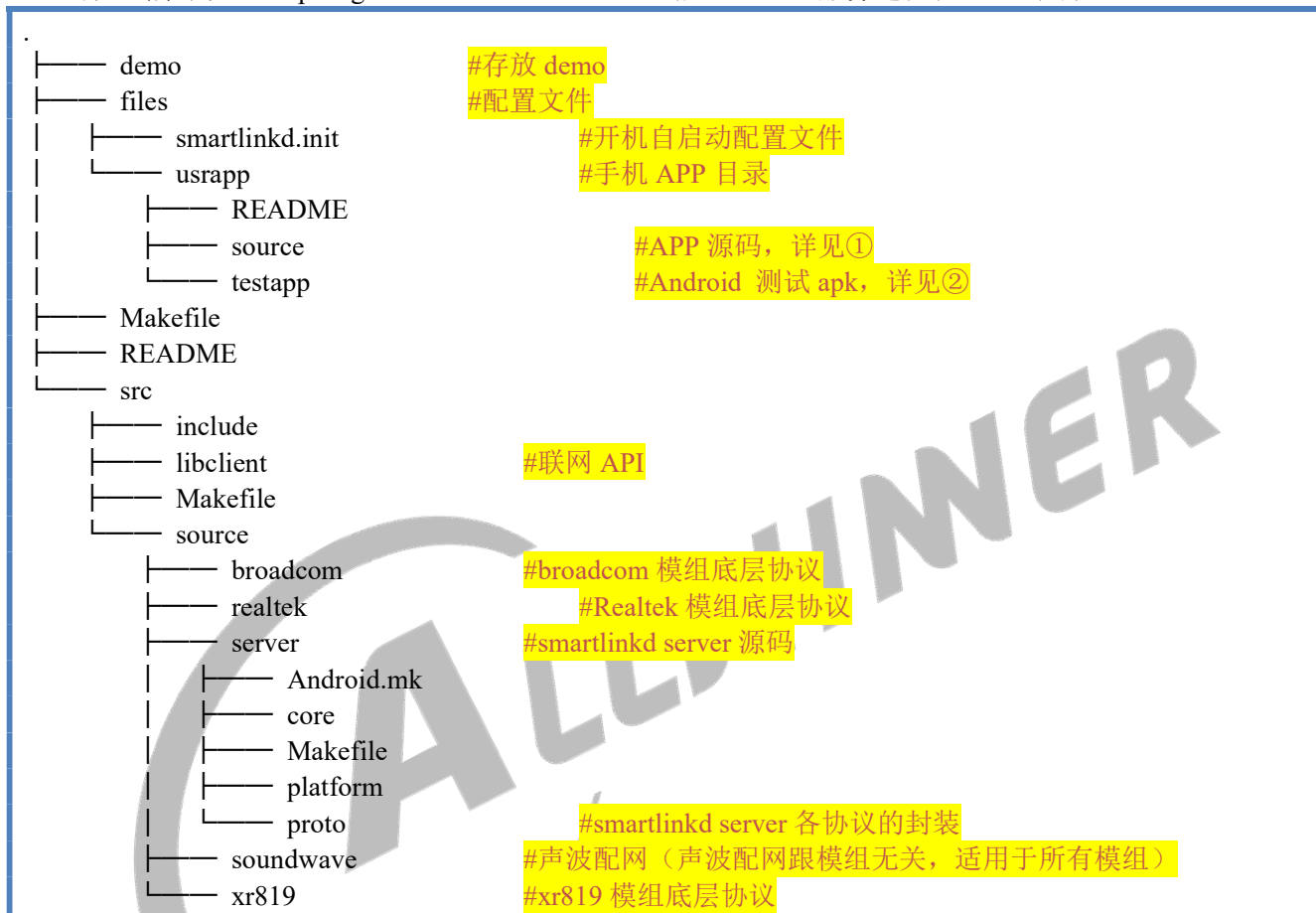


2. Smartlink 简介

Tina 中目前支持的 wifi 模组有 XR819, Broadcom AP 系列模组。支持的配网方式有 airkiss, soundwave (声波), 另外 XR819 还支持 smartconfig, Broadcom 还支持 cooee, softap (热点)。

2.1. Sdk 代码目录

源码路径为<tina>/packge/allwinner/smartlinkd。包括 smartlink 服务进程和 demo 程序。



①source 目录下



②test 目录下的 apk, 安装到 Android 手机中可直接使用, 方便测试



2.2. 编译配置

XR819 menuconfig 配置如下:

```

make menuconfig
└─> Allwinner
    └─> <*> smartlinkd..... smartlinkd for Tina --->
        └─><> smartlinkd-demo..... smartlinkd demo #详见①
        └─><> smartlinkd-soundwave..... smartlinkd_soundwave demo #详见②
        └─><> smartlinkd-xrairkiss..... xr819 airkiss demo #详见③
        └─><> smartlinkd-xrconfig..... xr819 smartconfig demo #详见④
    
```

- ①smartlinkd demo 程序，如果需要使用 demo 程序需要选上
 - ②demo 所使用的声波配网协议
 - ③demo 所使用的 airkiss 配网协议
 - ④demo 所使用的 smartconfig 配网协议
- PS: smartlinkd demo 依赖于配网协议，但是配网协议不依赖与 smartlinkd demo，配网协议 demo 可以单独使用（只用于接收数据，不进行联网）。

broadcom menuconfig 配置如下:

```

make menuconfig
└─> Allwinner
    └─> <*> smartlinkd..... smartlinkd for Tina --->
        └─><> smartlinkd-demo..... smartlinkd demo #详见①
        └─><> smartlinkd-dhd_priv..... dhd_priv #详见②
        └─><> smartlinkd-raw..... smartlink_raw #详见③
        └─><> smartlinkd-setup..... smartlinkd_setup #详见④
        └─><> smartlinkd-softAP..... smartlinkd_softAP demo #详见⑤
        └─><> smartlinkd-soundwave..... smartlinkd_soundwave demo #详见⑥
    
```

- ①smartlinkd demo 程序，如果需要使用 demo 程序需要选上
 - ②③为 broadcom 的调试工具
 - ④demo 所使用的配网协议包括 airkiss 和 cooece
 - ⑤demo 所使用的热点方式配网
 - ⑥demo 所使用的声波配网协议
- PS: smartlinkd demo 依赖于配网协议，但是配网协议不依赖与 smartlinkd demo，配网协议 demo 可以单独使用（只用于接收数据，不进行联网）。

2.3. APP 编写说明

(1) 导入接口文件

```
#include "aw_smartlinkd_connect.h"
```

(2) 链接动态库

```
libsmartlink_client.so
```

(3) 实例代码

APP demo 代码目录为:

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

2.4. 启动 smartlinkd server

再使用 app 配网之前, 需要启动 smartlinkd server, 很多平台默认是开机自启动的。可以使用 ps 命令, 查看是否有: smartlinkd 进程。如果没有, 执行如下命令启动:

```
/etc/init.d/smartlinkd start
```

3. API

3.1. 数据结构

```
struct ap_info {
    char ssid[LEN];
    char password[LEN];
    int security;
};
struct sender_info {
    char ip[16]; /* ipv4 max length */
    int port;
};
struct _info {
    struct ap_info base_info; // ap 信息
    int protocol; // 协议类型
    int airkiss_random; // airkiss 协议 random 字段
    struct sender_info ip_info; // cooe 协议配置端信息
    char adt_str[ADT_STR]; // 声波联网信息
};
struct _cmd { // 传输数据包
    int cmd; // 数据包操作码
    char usekey;
    char AESKey[KEY_LEN];
    struct _info info; // 协议、ap 相关返回信息
};
```

3.2. 检测&准备环境

函数原型	int check_wlan_interface()
参数说明	无
返回说明	1: 成功; 0: 失败
功能描述	检测驱动模块是否加载

3.3. 初始化

函数原型	int aw_smartlinkd_init(int fd,int(f)(char*,int))
参数说明	fd: 当前版本未使用, 调用时设置为 0 即可 int(f)(char*,int): 事件回调接口
返回说明	-1: 初始化失败

	0: 成功
功能描述	初始化, 并注册消息状态回调函数。 当一次检测有结果返回后, 需要重新调用此接口再次建立与服务的连接。

3.4. 消息回调

函数原型	int(f)(char* buf ,int lenght)
参数说明	buf 为相关信息头指针。 lenght 为 buf 长度, 也复用作连接状态。 Lenght: 0 服务端主动断开 Lenght: -1 连接错误
返回说明	需要给回调的返回值: THREAD_EXIT 线程退出 THREAD_CONTINUE 线程保持
功能描述	事件回调函数

3.5. 启动 airkiss

函数原型	int aw_startairkiss()
参数说明	void
返回说明	int 0:失败; 非 0:成功。
功能描述	启动 airkiss。开始 airkiss 协议检测, 非阻塞调用, 检测维持 3 分钟, 超时后, 回复失败消息。

3.6. 停止 airkiss

函数原型	int aw_stopairkiss()
参数说明	void
返回说明	int 0:失败; 非 0:成功。
功能描述	停止 airkiss。airkiss start 检测维持 3 分钟中, 相关的回调还没有收到, 可以使用 stop 接口停止相关的检测。

3.7. 启动 cooee

函数原型	int aw_startcooee()
参数说明	void
返回说明	int 0:失败; 非 0:成功。
功能描述	启动 cooee。开始 cooee 协议检测, 非阻塞调用, 检测维持 3 分钟, 超时后, 回复失败消息。 仅仅适用于 broadcom 模组。

3.8. 停止 cooee

函数原型	int aw_stopairkiss()
参数说明	void
返回说明	int 0:失败; 非 0:成功。
功能描述	停止 cooee。cooee start 检测维持 3 分钟中, 相关的回调还没有收到, 可以使用 stop 接口停止相关的检测。 仅仅适用于 broadcom 模组。

3.9. 启动声波联网

函数原型	int aw_startadt()
参数说明	void
返回说明	int 0:失败; 非 0:成功。
功能描述	启动声波联网。开始声波联网检测, 非阻塞调用, 检测维持 3 分钟, 超时后, 回复失败消息。

3.10. 停止声波联网

函数原型	int aw_stoptadt()
参数说明	void
返回说明	int 0:失败; 非 0:成功。
功能描述	停止声波联网。声波联网 start 检测维持 3 分钟中, 相关的回调还没有收到, 可以使用 stop 接口停止相关的检测。

3.11. 启动 smartconfig 联网

函数原型	int aw_startxrsc()
参数说明	int
返回说明	int 0:失败; 非 0:成功。
功能描述	启动 smartconfig 联网。开始 smartconfig 联网检测, 非阻塞调用, 检测维持 3 分钟, 超时后, 回复失败消息。 仅仅适用于 XR819 模组。

3.12. 停止 smartconfig 联网

函数原型	int aw_stopxrsc()
参数说明	void
返回说明	int 0:失败; 非 0:成功。
功能描述	停止 smartconfig 联网。smartconfig 联网 start 检测维持 3 分钟中, 相关的回调还没有收到, 可以使用 stop 接口停止相关的检测。 仅仅适用于 XR819 模组。

3.13. 混合启动模式

函数原型	int aw_startcomposite(int proto_mask)
参数说明	proto_mask 启动的联网类型
返回说明	int 0:失败; 非 0:成功。
功能描述	随启动参数启动联网检测, 传入参数代表启动的联网类型, 其中: cooe 联网 AW_SMARTLINKD_PROTO_COOEE = 1 airkiss 联网 AW_SMARTLINKD_PROTO_AKISS = 1<<2 声波联网 AW_SMARTLINKD_PROTO_ADT = 1<<3

3.14. 混合停止模式

函数原型	int aw_stoptcomposite()
参数说明	void

返回说明	int 0:失败; 非 0:成功。
功能描述	停止智能联网。start 检测维持 3 分钟中, 相关的回调还没有收到, 可以使用 stop 接口停止相关的检测。



4. 调试说明

4.1. 单独测试协议接收

各协议提供单独数据包收发 demo（不进行联网，只接收数据）

4.1.1. Broadcom 模组

Broadcom airkiss 和 cooe 测试可执行 smartlinkd_setup 程序，源码路径如下：

```
<tina>/package/allwinner/smartlinkd/src/source/broadcom/EasySetupTarget/jni
```

命令的含义可执行如下：

```
smartlinkd_setup -h
-h: show help message
-d: show debug message
-k <v>: set 16-char key for all protocols
-p <v>: bitmask of protocols to enable
-s,      Whether to connect smartlinkd server(yes or no),default yes
0x0001 - bcast
0x0002 - neeze
0x0004 - Air Kiss
```

(1) 测试 airkiss 功能

手机端连接路由器，打开 airkiss app，并发送数据包，默认 AESKey 为空，不必输入。

设备端：执行如下命令

```
smartlinkd_setup -p 4 -d -s no
```

(2) 测试 cooe 功能

手机端连接路由器，打开 BrcmCooe app，并发送数据包

设备端：执行如下命令

```
smartlinkd_setup -p 2 -d -s no
```

(3) 热点方式

热点方式的源码路径：

```
<tina>/package/allwinner/smartlinkd/src/source/softAP
```

测试之前需要配置 broadcom 模组支持 softap 模式，设置方式请参看《Tina softap API》文档。

配置好之后，手机端连接路由器，打开 ckySoftAPDemo，并发送数据包。

设备端：执行如下命令

```
smartlinkd_softap
```

4.1.2. XR819 模组

(1) 测试 airkiss 功能

源码路径

```
<tina>/package/allwinner/smartlinkd/src/source/xr819/airkiss
```

测试方法：

手机端连接路由器，打开 airkiss app，并发送数据包,注意:默认 AESKey=1234567890123456。

设备端，执行如下命令：

```
smartlinkd_xrairkiss -s no
```

(2) 测试 smartconfig 功能

源码路径

```
<tina>/package/allwinner/smartlinkd/src/source/xr819/smartconfig
```

测试方法:

手机端连接路由器, 打开 Xconfig app, 并发送数据包
设备端, 执行如下命令:

```
smartlinkd_xrsc -c no
```

4.1.3. 声波

声波配网方式不依赖与模组 (任何模组都可使用)。

源码路径

```
<tina>/package/allwinner/smartlinkd/src/source/soundwave
```

测试方法:

通过声波传输联网数据, 需要保证手机端与设备端的通信参数 (Sample Rate, Frequency Type) 匹配。
设备端默认的 Sample Rate: 16000, Frequency Type: LOW(0)。可以通过传入参数设置 Sample Rate 与 Frequency Type, 详情可执行命令: smartlinkd_soundwave -h。

手机端打开 SoundAuthenticationTest APP, 设置 Sample Rate: 16000 与 Frequency Type: low, 以及 SSID 和 PASSWORD。

设备端, 执行如下命令:

```
smartlinkd_soundwave -f 0 -r 16000 -s no
```

4.2. 使用 smartlinkd_demo

源码路径

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

smartlinkd_demo 可执行程序包含了数据收发到联网的整个过程, 同时整合了所有的联网协议。

测试方式:

(1) 确保 smartlinkd server 以及启动。

可使用 ps 命令查看: smartlinkd 进程。

如果没有启动, 执行: /etc/init.d/smartlinkd start

(2) 执行 smartlinkd_demo

example: smartlinkd_demo V

V:take the following value

```
0-airkiss
1-cooe ----- Only support Broadcom module
2-soundwave
3-softap ----- Only support Broadcom module
4-xrsc ----- Only support XR819 module
5-composite
```

5. Declaration

This document is the original work and copyrighted property of Allwinner Technology (“Allwinner”). Reproduction in whole or in part must obtain the written approval of Allwinner and give clear acknowledgment to the copyright owner.

The information furnished by Allwinner is believed to be accurate and reliable. Allwinner reserves the right to make changes in circuit design and/or specifications at any time without notice. Allwinner does not assume any responsibility and liability for its use. Nor for any infringements of patents or other rights of the third parties which may result from its use. No license is granted by implication or otherwise under any patent or patent rights of Allwinner. This datasheet neither states nor implies warranty of any kind, including fitness for any particular application.

