



Hi1131S V100

HisiConnect 应用指南

文档版本 **B02**

发布日期 **2017-04-18**

版权所有 © 深圳市海思半导体有限公司 2016。保留一切权利。

非经本公司书面许可，任何单位和个人不得擅自摘抄、复制本文档内容的部分或全部，并不得以任何形式传播。

商标声明



HISILICON、海思和其他海思商标均为深圳市海思半导体有限公司的商标。

本文档提及的其他所有商标或注册商标，由各自的所有人拥有。

注意

您购买的产品、服务或特性等应受海思公司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的购买或使用范围之内。除非合同另有约定，海思公司对本文档内容不做任何明示或默示的声明或保证。

由于产品版本升级或其他原因，本文档内容会不定期进行更新。除非另有约定，本文档仅作为使用指导，本文档中的所有陈述、信息和建议不构成任何明示或暗示的担保。

深圳市海思半导体有限公司

地址： 深圳市龙岗区坂田华为总部办公楼 邮编：518129
网址： <http://www.hisilicon.com/cn/>
客户服务电话： 4008302118
客户服务邮箱： support@hisilicon.com

前 言

概述

本文档主要介绍 Hi1131S Hisiconnect 的 Android 及 IOS APP 应用开发。

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

符号约定

在本文中可能出现下列标志，它们所代表的含义如下。

符号	说明
 危险	用于警示紧急的危险情形，若不可避免，将会导致人员死亡或严重的人身伤害。
 警告	用于警示潜在的危险情形，若不可避免，可能会导致人员死亡或严重的人身伤害。
 小心	用于警示潜在的危险情形，若不可避免，可能会导致中度或轻微的人身伤害。
 注意	用于传递设备或环境安全警示信息，若不可避免，可能会导致设备损坏、数据丢失、设备性能降低或其它不可预知的结果。 “注意”不涉及人身伤害。
 说明	用于突出重要/关键信息、最佳实践和小窍门等。 “说明”不是安全警示信息，不涉及人身、设备及环境伤害信息。

修改记录

修改记录累积了每次文档更新的说明。最新版本的文档包含以前所有文档版本的更新内容。

修改日期	版本	相关的 DTS 或 JIRA 单号	修改描述	修改作者
2017-4-18	B01	无	首版本发布	
2017-7-14	B02	无	添加 HisiConnect IOS 相关描述	

目 录

前 言.....	ii
1 HisiConnect Android 应用说明	7
1.1 HisiConnect 应用框架	7
1.1.1 整体调用关系	7
1.1.2 libHisiLink.so	8
1.1.3 hi1131s_hisilink.jar.....	8
1.1.3.2 ProcessAPLinkData.class	8
1.1.3.3 ProcessMultiCastLinkData.class	8
1.1.3.4 WbAes.class.....	9
1.1.3.5 WifiNetworkInfo.class.....	9
1.2 组播方案库使用方法	10
1.2.1 初始化通信信息	10
1.2.2 构建发送报文	10
1.2.3 启动上线信息接收线程	10
1.2.4 组播报文发送	11
1.3 传统 AP 方案库使用方法	12
1.3.1 初始化通信信息	12
1.3.2 构建发送报文	12
1.3.3 配置设备 AP 信息	12
1.3.4 断开当前 WiFi, 关联设备 WiFi	12
1.3.5 创建 TCP 连接发送报文	12
1.3.6 关联回家庭路由器	12
2 HisiConnect IOS 应用说明	13
2.1 HisiConnect 应用框架	13
2.1.1 整体调用关系	13
2.1.2 libhisiConnectLib.a.....	13
2.1.2.1 库支持的指令集	14
2.1.2.2 头文件说明	14
2.1.2.2.1 WifiNetworkInfo.h.....	14
2.1.2.2.2 ProcessMultiCastLinkData.h	14

2.2 组播方案库使用方法 15

2.2.1 初始化通信信息 15

2.2.2 构建发送报文 16

2.2.3 启动上线信息接收线程 16

2.2.4 组播报文发送 16

2.3 传统 AP 方案库使用说明 17

插图目录

图 1-1 HisiConnect 应用调用关系..... 7

图 1-2 libHisiLink.so 8

图 1-3 hi1131s_hisilink.jar 8

图 2-1 HisiConnect 应用调用关系..... 13

图 2-2 libhisiConnectLib.a 13

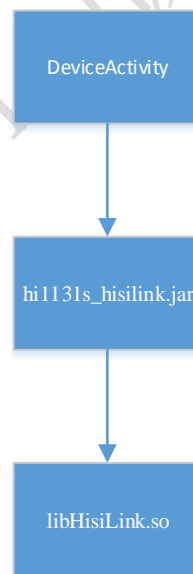
图 2-3 库支持的指令集..... 14

1 HisiConnect Android 应用说明

1.1 HisiConnect 应用框架

1.1.1 整体调用关系

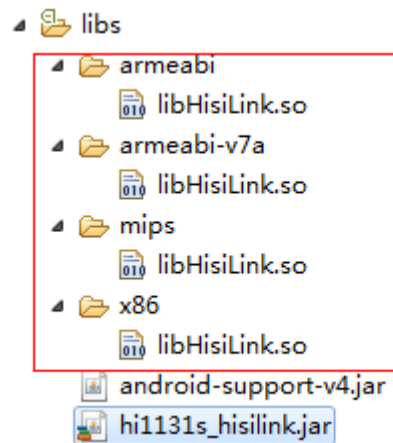
图1-1 HisiConnect 应用调用关系



如图 1-1 所示，应用的界面 Activity 直接调用 hi1131s_hisilink.jar 中的类创建实例，并调用该类中的方法实现报文构建及发送功能。该 jar 包中的类会自行调用 libHisiLink.so 中的加密方法进行加密。

1.1.2 libHisiLink.so

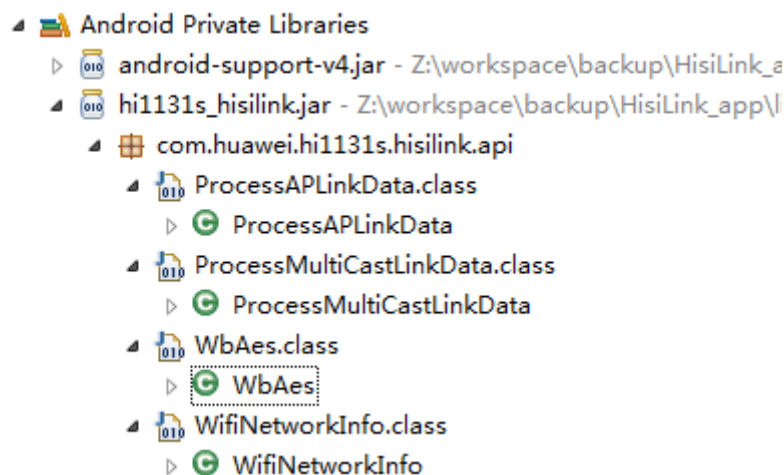
图1-2 libHisiLink.so



如上图所示，是各个版本的该库文件，该文件主要是数据进行加密操作，其加载和调用，均由 hi1131s_hisilink.jar 完成，用户只需将该文件添加到 libs 目录下即可。

1.1.3 hi1131s_hisilink.jar

图1-3 hi1131s_hisilink.jar



如图所示，该 jar 包共四个类，下面拆开分析：

1.1.3.2 ProcessAPLinkData.class

该类主要提供传统 AP 模式的获取设备 SSID 和密码的方法，及消息发送的方法。具体使用见《1.3 传统 AP 方案库使用方法》。

1.1.3.3 ProcessMultiCastLinkData.class

该类主要提供组播模式的报文构建，及发送的方法。具体使用见《1.2 组播方案库使用方法》。

1.1.3.4 WbAes.class

该类主要提供加密相关的算法，其使用经由 `ProcessAPLinkData.class` 和 `ProcessMultiCastLinkData.class` 调用，用户无需关心。

1.1.3.5 WifiNetworkInfo.class

该类主要存储当前网络信息，及通信端口信息等。为后续构建发送报文的基础，具体使用见 1.2、1.3 章。

1.2 组播方案库使用方法

1.2.1 初始化通信信息

```
WifiNetworkInfo wifiNetworkInfo = new WifiNetworkInfo();  
wifiNetworkInfo.setIp(mWiFiAdmin.getWifiIPAdress());  
wifiNetworkInfo.setSecurity(mWiFiAdmin.getSecurity());  
wifiNetworkInfo.setSsid(mWiFiAdmin.getWifiSSID());  
wifiNetworkInfo.setPassword(textPass.getText().toString());  
wifiNetworkInfo.setPort(1131);  
wifiNetworkInfo.setDeviceName(this.strName);  
wifiNetworkInfo.setOnlineProto(ONLINE_MSG_BY_UDP);
```

如上面代码所示，使用 **WifiNetworkInfo** 类建立变量，分别做以下信息的初始化：

setIp:当前手机的 IP 地址，用于后续设备发送上线包用。

setSecurity:家用路由器的加密方式。

setSsid:家用路由器的名称。

setPassword:家用路由器的密码。

setPort:上线包 server 端的端口，用户可自行设置可用端口。

setDeviceName:设备名称，注：通过扫描到的设备解析而出，在 **Mainactivity** 中给出，其最后两位为标示位，用于区分该手机是否是在配置该设备。

setOnlineProto:设置上线方式，目前组播方案，尽量使用 **TCP** 方式上线。

根据上述接口，用户可自行实现获取方式，代码中也有 **demo** 实现。

1.2.2 构建发送报文

```
ProcessMultiCastLinkData processLinkData = new ProcessMultiCastLinkData();  
processLinkData.constructMultiCastLinkMessageToSend(wifiNetworkInfo);
```

如上两行代码所示，用户使用 **ProcessMultiCastLinkData** 类定义实例，并调用 **constructMultiCastLinkMessageToSend** 方法，将 1.2.1 中生成的信息作为入参传递即可。

1.2.3 启动上线信息接收线程

该部分用户可自行实现，接收方式及端口一定要与 **WifiNetworkInfo** 的实例初始化一致。而上线报文内容为：**online**:+设备 6 位 **MAC** 地址，共 13 位。

1.2.4 组播报文发送

用户调用 1.2 实例的 `processLinkData.startSendMultiBroadcast(true)` 方法发送即可。同时可以调用 `stopSendMultiBroadcast(false)` 停止发送。

1.3 传统 AP 方案库使用方法

1.3.1 初始化通信信息

该部分与 1.2.1 基本相同，只是 setOnlineProto 上线包的接收方式，必须使用 UDP 方法。

1.3.2 构建发送报文

```
ProcessAPLinkData processAPLinkData = new ProcessAPLinkData();  
processAPLinkData.constructAPLinkMessageToSend(wifiNetworkInfo);
```

用法同 1.2.2 节。

1.3.3 配置设备 AP 信息

ssid: MainActivity 扫描中获取的。

密码: 将 ssid 作为入参, 传入 processAPLinkData.generateWifiPasswordBySSID 方法生成。

```
passwordString = processAPLinkData.generateWifiPasswordBySSID(ssid);
```

加密方式, 为固定的 wpa 加密方式。

1.3.4 断开当前 WiFi, 关联设备 WiFi

根据 1.3.3 生成的信息, 用户自行实现。

1.3.5 创建 TCP 连接发送报文

关联成功后, 建立 TCPclient 去连接设备, server 的 IP 地址为 192.168.43.1, 端口号为 5000。

然后将 outputStream 作为入参传入 processAPLinkData.startSendAPLinkData 方法。

```
processAPLinkData.startSendAPLinkData(outputStream);
```

1.3.6 关联回家庭路由器

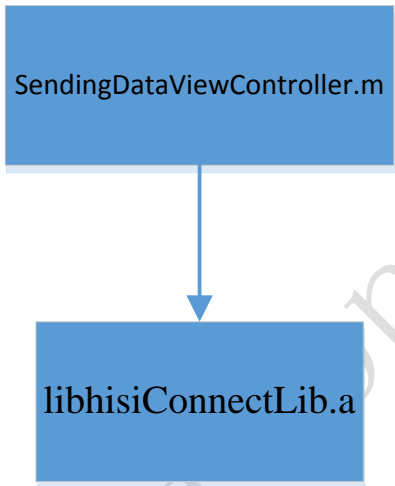
该部分由用户自行实现, 设备获取到足够的信息后, 会踢掉手机, 手机检测到去关联后, 关联家庭路由器即可。

2 HisiConnect IOS 应用说明

2.1 HisiConnect 应用框架

2.1.1 整体调用关系

图2-1 HisiConnect 应用调用关系



注意：当前 demo 是在 Xcode Version8.3.1(8E1000a)开发，已在真机上验证运行；由于 Demo View 布局存在不合适的地方，目前在虚拟上运行有问题，不影响库的使用，请客户将相关逻辑移植到自己的应用程序上运行，待后续完善。

2.1.2 libhisiConnectLib.a

图2-2 libhisiConnectLib.a

共享文件夹				
名称	^	修改日期	大小	种类
libhisiConnectLib.a		2017年6月24日 上午11:41	3 MB	文稿
ProcessMultiCastLinkData.h		2017年5月9日 下午4:40	660 字节	C Hea...Source
Release-iphoneos		2017年6月24日 上午11:33	--	文件夹
Release-iphonesimulator		2017年6月24日 上午11:38	--	文件夹
WifiNetworkInfo.h		2017年5月9日 上午9:21	674 字节	C Hea...Source

2.1.2.1 库支持的指令集

图2-3 库支持的指令集

```
hi1311deMac-mini:hisiconnect_lib hi1311$ ls
ProcessMultiCastLinkData.h  Release-iphonesimulator  libhisiConnectLib.a
Release-iphoneos             WifiNetworkInfo.h
hi1311deMac-mini:hisiconnect_lib hi1311$ lipo -info libhisiConnectLib.a
Architectures in the fat file: libhisiConnectLib.a are: i386 armv7 armv6 armv7s x86_64 arm64
```

目前 libhisiConnectLib.a 支持 armv7、armv6、armv7s、arm64、i386 以及 x86_64 指令集。单独 arm 指令集的库(Release-iphoneos \libhisiConnectLib.a)，Mac 处理器指令集库(Release-iphonesimulator\ libhisiConnectLib.a)。

2.1.2.2 头文件说明

2.1.2.2.1 WifiNetworkInfo.h

WifiNetworkInfo.h 主要用于存储当前网络信息，及通信端口信息等。为后续构建发送报文的基础，具体使用见 2.2 组播方案库使用方法。

```
@interface WifiNetworkInfo : NSObject
@property (nonatomic, assign) NSInteger port;
@property (nonatomic, assign) NSInteger security;
@property (nonatomic, strong) NSString *ssid;
@property (nonatomic, strong) NSString *password;
@property (nonatomic, strong) NSString *deviceName;
@property (nonatomic, strong) NSString *ip;
@property (nonatomic, assign) NSInteger onlineProto;
@property (nonatomic, strong) NSString *bssid;
@end
```

2.1.2.2.2 ProcessMultiCastLinkData.h

```
#import "WifiNetworkInfo.h"
#define GLocalProcessMultiCastLinkData [ProcessMultiCastLinkData sharedInstance]
@interface ProcessMultiCastLinkData : NSObject
+ (instancetype) sharedInstance;
-(Boolean)constructMultiCastLinkMessageToSend:(WifiNetworkInfo *)wifiNetworkInfo;
-(Boolean)startSendMultiBroadcast:(Boolean)enAllowSendData:(unsigned int)port;
-(void)stopSendMultiBroadcast;

@end
```

2.2 组播方案库使用方法

2.2.1 初始化通信信息

```
#import "WifiNetworkInfo.h"

WifiNetworkInfo * myWifiNetworkInfo = [[WifiNetworkInfo alloc] init];

[myWifiNetworkInfo setSsid:[GLocalNetworkManagerByReachability getWifiSSID]];

if((myWifiNetworkInfo.security==SECURITY_WEP) ||
(myWifiNetworkInfo.security==SECURITY_PSK))
{
    /*get password*/
    NSString *password = passwordTextField.text;
    [myWifiNetworkInfo setPassword:password];
}
else
{
    [myWifiNetworkInfo setPassword:@""];
}

[myWifiNetworkInfo setSecurity:myWifiNetworkInfo.security];
[myWifiNetworkInfo setDeviceName:deviceIDTextField.text];
[myWifiNetworkInfo setIp:[GLocalNetworkManagerByReachability
getLocalIPAddress:@"en0"]];
[myWifiNetworkInfo setPort:ONLINE_PORT];
[myWifiNetworkInfo setOnlineProto:ONLINE_MSG_BY_TCP];
```

如上面代码所示，使用 WifiNetworkInfo 类建立变量，分别做以下信息的初始化：

setSsid:家用路由器的名称。

setPassword:家用路由器的密码。

setSecurity:家用路由器的加密方式。

setDeviceName:设备名称，用于区分设备（防止出现同一时刻多个设备进入 HisiConnect 模式），一般以 mac 地址的后四个字符（比如：28-6E-D4-8A-06-F8，传递 06F8）。该信息会通过 HisiConnect 发送给设备侧逻辑，当设备收到完整报文并解析成功时，会判断本设备的 mac 地址是否以设置的设备名称结尾（比如:06F8）？如果是，进行 Wi-Fi 关联过程，否则，终止 HisiConnect。

特别注意：IOS 应用侧逻辑和设备侧逻辑是成对出现，需要同时进行修改。

setIp:当前手机的 IP 地址，用于后续设备发送上线包用。

setPort:上线包 server 端的端口，用户可自行设置可用端口。

setOnlineProto:设置上线方式，目前组播方案，尽量使用 TCP 方式上线。

根据上述接口，用户可自行实现获取方式，代码中也有 demo 实现。

2.2.2 构建发送报文

```
#import "ProcessMultiCastLinkData.h"

/*构建数据包*/

[GlobalProcessMultiCastLinkData
constructMultiCastLinkMessageToSend:myWifiNetworkInfo];
```

如上述代码，用户使用 GlobalProcessMultiCastLinkData 定义实例，并调用 constructMultiCastLinkMessageToSend 方法，将《2.2.1 初始化通信信息》中生成的信息作为入参传递即可。

2.2.3 启动上线信息接收线程

该部分用户可自行实现，接收方式及端口一定要与 WifiNetworkInfo 的实例初始化一致。而上线报文内容可以按照 Android 实例：online:+设备 6 位 MAC 地址，共 13 位。目前 IOS 应用未对上线信息进行解析，客户根据自己的需求进行处理。

```
/*创建线程监听上线消息*/

[self recieveOnlineMessage];
```

2.2.4 组播报文发送

```
/*发送报文*/

dispatch_async(dispatch_get_global_queue(DISPATCH_QUEUE_PRIORITY_DEFAULT, 0),
^{
    [GlobalProcessMultiCastLinkData startSendMultiBroadcast:true:3516];
});
```

```
[GlobalProcessMultiCastLinkData stopSendMultiBroadcast];
```

用户调用 startSendMultiBroadcas 方法发送即可。同时可以调用 [GlobalProcessMultiCastLinkData stopSendMultiBroadcast];停止发送。

2.3 传统 AP 方案库使用说明

该方案较通用，用户一般都有该方案；如果没有，请用户自行实现。