楼宇对讲库接口说明-Android

**v0.2**

**创建日期：2016-01-15**

**最后修改：2016-01-19**

**青岛东软载波科技股份有限公司**

目录

[楼宇对讲库接口说明-Android 1](#_Toc440984386)

[1. 简介 4](#_Toc440984387)

[1.1. 楼宇对讲库 4](#_Toc440984388)

[1.2. Android 4](#_Toc440984389)

[2. 楼宇对讲库 4](#_Toc440984390)

[2.1. 楼宇对讲库初始化 4](#_Toc440984391)

[2.2. 销毁楼宇对讲库的服务 4](#_Toc440984392)

[2.3. 设备信息修改 4](#_Toc440984393)

[2.4. 地址信息修改 5](#_Toc440984394)

[3. 对讲通信 5](#_Toc440984395)

[3.1. 初始化对讲服务 5](#_Toc440984396)

[3.2. 取消函数注册 5](#_Toc440984397)

[3.3. 收到呼叫消息 5](#_Toc440984398)

[3.4. 通知ui视频播放地址 5](#_Toc440984399)

[3.5. 呼叫过程中对方挂断 6](#_Toc440984400)

[3.6. ui应答呼叫请求 6](#_Toc440984401)

[3.7. 对讲确认 6](#_Toc440984402)

[3.8. 对讲服务向ui推送音频数据 6](#_Toc440984403)

[3.9. ui发送音频数据 6](#_Toc440984404)

[3.10. ui挂断 7](#_Toc440984405)

[3.11. 对讲过程中对方挂断 7](#_Toc440984406)

[3.12. 主动呼叫用户 7](#_Toc440984407)

[3.13. 用户回复忙 7](#_Toc440984408)

[3.14. 呼叫中心管理机 7](#_Toc440984409)

[3.15. 开锁 7](#_Toc440984410)

[3.16. 开锁确认 8](#_Toc440984411)

[3.17. 监视门口机 8](#_Toc440984412)

[3.18. 门口机监视确认 8](#_Toc440984413)

[3.19. 主动挂断监视 8](#_Toc440984414)

[3.20. 被动挂断监视 8](#_Toc440984415)

[3.21. 监视回复忙 9](#_Toc440984416)

[4. IMP 9](#_Toc440984417)

[4.1. 初始化IMP任务 9](#_Toc440984418)

[4.2. 销毁IMP任务 9](#_Toc440984419)

[4.3. IMP文本推送 9](#_Toc440984420)

[4.4. IMP广告推送 9](#_Toc440984421)

[5. 安防报警 10](#_Toc440984422)

[5.1. 初始化报警任务 10](#_Toc440984423)

[5.2. 销毁报警任务 10](#_Toc440984424)

[5.3. 报警回调 10](#_Toc440984425)

[6. 结构体定义 10](#_Toc440984426)

[6.1. 设备类型 10](#_Toc440984427)

[6.2. 室内机信息 10](#_Toc440984428)

[6.3. 门口机信息 11](#_Toc440984429)

[6.4. 设备信息 11](#_Toc440984430)

[6.5. IMP广告推送 11](#_Toc440984431)

[6.6. 回调函数类型枚举 11](#_Toc440984432)

# 简介

此文档详细说明了楼宇对讲库对外接口及使用方法。

楼宇对讲库使用C语言编写，并编译成.a静态库。由于Android版程序是用java编写，且只能使用.so动态库，因此在楼宇对讲库的基础上封装一层JNI，并编译为.so动态库。

以下为楼宇对讲库实现的服务和ui程序应实现的接口功能

## 楼宇对讲库

* 对讲通信，音视频传输
* ~~历史记录（对讲记录、安防报警记录）~~
* 读取报警驱动，通知ui和imp报警消息
* 与IMP交互，将IMP的文本及广告推送通知ui

## Android

* 对讲通信，音视频播放、录制
* ~~读取对讲记录、安防报警记录~~
* 记录并读取对讲记录、安防报警记录
* 收到报警消息后声光报警
* 接收推送的文本消息和广告

# 楼宇对讲库

## 楼宇对讲库初始化

Android程序启动后，需要初始化楼宇对讲库

函数原型：int init\_intercom\_core(device\_info\_t \*device);

函数说明：参数device包含本设备的设备信息，详情参考“结构体定义”；返回值为0表示初始化成功，-1表示失败

## 销毁楼宇对讲库的服务

如果需要，ui可以销毁楼宇对讲库的服务，即各服务的线程和资源

函数原型：int destory\_intercom\_core(void);

函数说明：返回值为0表示初始化成功，-1表示失败

## 设备信息修改

函数原型：int modefy\_device\_info(device\_info\_t \*device);

函数说明：device\_info\_t设备信息结构体；返回值为0表示初始化成功，-1表示失败

## 地址信息修改

函数原型：int modefy\_add\_info(struct in\_addr \*imp\_addr, struct inaddr \*center\_addr);

函数说明：struct in\_addr：Linux IPv4地址结构体；返回值为0表示初始化成功，-1表示失败

# 对讲通信

## 初始化对讲服务

Android程序启动后，需要初始化对讲服务。

函数原型：int init\_talk\_task(void);

函数说明：初始化成功返回值为0，失败时返回值为-1

## 取消函数注册

函数原型：int unregister\_handler(hanler\_type\_t type)

函数说明：参数hanler\_type\_t为回调函数的类型枚举，参见“结构体定义”

## 收到呼叫消息

对讲服务收到呼叫消息后，通过回调通知ui：

ui需要事先通过下面的函数注册回调函数

注册函数：

void register\_inactive\_call\_handler(void (\*inactive\_call\_handler)(device\_info\_t \*));

回调函数原型：void inactive\_call\_handler(device\_info\_t \*device\_info)

回调函数说明：参数device \_info包含发起呼叫方的设备类型和对应的设备信息，详情参考“结构体定义”

## 收到设备找到消息

主动呼叫时需要先查找设备，ui显示“正在查找设备”，设备找到后改为“正在呼叫”。找到设备后对讲服务以回调方式通知ui。

注册函数：

void register\_call\_answer\_found\_handler(void (\*call\_answer\_found\_handler)(device\_info\_t \*));

回调函数原型：void call\_answer\_found\_handler(device\_info\_t \*device\_info)

回调函数说明：参数device \_info包含对方的设备类型和对应的设备信息，详情参考“结构体定义”

## 通知ui视频播放地址

对讲服务收到视频帧后，建立Rtsp服务，并通过回调函数通知ui视频播放地址

注册函数：int register\_video\_url\_handler(void (\*url\_handler)(char \*));

回调函数原型：void video\_url\_handler(char \*url)

回调函数说明：url为“\0”结尾的字符串

## 呼叫过程中对方挂断

呼叫过程中（即正在振铃，还未接听），如果对方主动挂断，则对讲服务通过回调通知ui此消息

注册函数：

void register\_inactive\_call\_hang\_up\_handler(void (\*inactive\_call\_hang\_up\_handler)(void));

回调函数原型：void inactive \_call\_hang\_up\_handler(void)

## ui应答呼叫请求

ui应答呼叫请求时需主动调用对讲服务的应答呼叫函数

函数原型：int ui\_talk\_answer(void);

函数说明：返回值为0表示应答成功，-1表示应答失败

## 收到对讲应答

主动呼叫用户时，对方应答呼叫后，对讲服务通过回调通知ui开始对讲

注册函数原型：

void register\_talk\_answer\_handler(void (\*talk\_answer\_handler)(void));

回调函数原型：void talk\_answer\_handler(void)

## 对讲确认

对讲服务收到对方确认对讲的报文后，通过回调通知ui开始对讲

注册函数：

void register\_talk\_confim\_handler(void (\*talk\_confim\_handler)(void));

回调函数原型：void talk\_confim\_handler(void)

## 对讲服务向ui推送音频数据

对讲服务收到音频数据后，通过回调函数通知ui处理音频数据

注册函数：

void register\_audio\_data\_handler(void (\*audio\_handler)(int ,char \*));

回调函数原型：audio\_data\_handler(int len, char \*data)

回调函数说明：len：音频数据长度，\*data：音频数据流

## ui发送音频数据

ui录制一段音频（和视频帧对应长度）后，调用对讲服务的函数发送数据

函数原型：void push\_audio\_data(int len, char \*data);

函数说明：len：发送音频数据的长度，\*data：音频数据流

## ui挂断

ui挂断对讲时，需要调用对讲服务的挂断函数

函数原型：void active\_hang\_up(void);

## 对讲过程中对方挂断

对讲过程中如果对方主动挂断，则对讲服务通过回调通知ui此消息

注册函数：

void register\_inactive\_talk\_hang\_up\_handler(void (\*inactive\_talk\_hang\_up\_handler)(void));

回调函数原型：void inactive\_talk\_hang\_up\_handler(void)

## 主动呼叫用户

函数原型：int active\_call\_user(room\_info\_t \*room);

函数说明：参数room包含被呼叫用户的室内机信息，详情参考“结构体定义”

## 用户回复忙

呼叫用户时，如果被叫设备回复忙，则对讲服务回调呼叫用户忙的函数。

注册函数：

void register\_call\_busy\_handler(void (\*call\_busy\_handler)(void));

函数原型：void call\_busy\_handler(void)

## 呼叫中心管理机

函数原型：int call\_center\_manager(struct in\_addr \*addr);

## 开锁

呼叫和对讲过程中室内机可以向门口机发送开锁命令

函数原型：int unlock\_door(void);

函数说明：返回值为0表示发送开锁报文成功，-1表示失败

## 开锁确认

发送开锁报文后，门口机开锁成功会回复确认报文，此时对讲库调用开锁确认的回调函数

注册函数：

void register\_unlock\_door\_confirm\_handler(void (\*unlock\_confirm\_hanler)(void));

回调函数原型：void unlock\_door\_confirm\_handler(void)

## 监视门口机

函数原型：int ui\_req\_monitor(device\_info\_t \*device);

函数说明：参数device\_info\_t为设备信息结构体，参加“结构体定义”

## 门口机监视确认

门口机接收到监视请求后，回复监视确认。

注册函数：

void register\_monitor\_confirm\_handler(void (\*monitor\_confirm\_handler)(device\_info\_t \*));

函数原型：void monitor\_confirm\_handler(device\_info\_t \*device);

函数说明：参数device\_info\_t为设备信息结构体，参加“结构体定义”

## 主动挂断监视

监视过程中，用户主动挂断监视。

函数原型：int active\_hang\_up\_monitor(device\_info\_t \*device);

函数说明：参数device\_info\_t为设备信息结构体，参加“结构体定义”

## 被动挂断监视

如果在监视过程中，门口机或围墙机呼叫用户，则门口机或围墙机请求挂断监视。

注册函数：

void register\_inactive\_hang\_up\_monitor\_handler(void (\*hang\_up\_handler)(device\_info\_t \*));

函数原型：void inactive\_hang\_up\_monitor\_handler(device\_info\_t \*device);

函数说明：参数device\_info\_t为设备信息结构体，参加“结构体定义”

## 监视回复忙

用户监视门口机或围墙机，如果将被监视的设备正在通话中，则恢复忙。

注册函数：

void register\_monitor\_response\_busy\_handler(void (\*monitor\_busy\_handler)(device\_info\_t \*));

函数原型：void monitor\_response\_busy\_handler(device\_info\_t \*device);

函数说明：参数device\_info\_t为设备信息结构体，参加“结构体定义”

# IMP

## 初始化IMP任务

函数原型：int init\_imp\_task(struct in\_addr \*imp\_addr);

函数说明：参数imp\_addr为IMP地址；返回值为0表示初始化成功，-1表示初始化失败

## 销毁IMP任务

函数原型：int destory\_imp\_task(void);

函数说明：返回值为0表示销毁成功，-1表示销毁失败

## IMP文本推送

注册函数：

int register\_text\_push\_handler(void (\*text\_push\_handler)(char \*));

回调函数原型：void imp\_text\_push\_handler(char \*text)

回调函数说明：参数\*text为推送文明字符串，带“\0”结束符

## IMP广告推送

注册函数：

int register\_ad\_push\_handler(void (\*ad\_push\_handler)(imp\_ad\_push\_t \*));

回调函数原型：void imp\_ad\_push\_handler(imp\_ad\_push\_t \*ad)

回调函数说明：参数imp\_ad\_push\_t为广告推送结构体，相见“结构体定义”

# 安防报警

## 初始化报警任务

函数原型：int init\_alarm\_in(void);

## 销毁报警任务

函数原型：int destory\_alarm\_in(void);

## 报警回调

注册函数：

int register\_alarm\_handler(void (\*alarm\_handler)(int));

回调函数原型：void alarm\_in\_handler(int channel)

回调函数说明：参数channel为报警的通道

# 结构体定义

## 设备类型

**typedef** **enum** device\_type\_t

{

*DT\_Room\_Machine* = 0,

*DT\_Unit\_Door\_Machine*,

*DT\_Wall\_Machine*,

*DT\_Secondary\_Confirmation\_Machine*,

*DT\_Center\_Machine*,

}device\_type\_t;

## 室内机信息

**typedef** **struct** room\_info\_t

{

**int** building\_no;

**char** unit\_no;

**char** layer\_no;

**char** room\_no;

**unsigned** **int** dev\_no;

} room\_info\_t, \*p\_room\_info\_t;

## 门口机信息

**typedef** **struct** door\_info\_t

{

**int** building\_no;

**char** unit\_no;

**unsigned** **int** dev\_no;

}door\_info\_t, \*p\_door\_info\_t;

## 设备信息

**typedef** **struct** device\_info\_t

{

device\_type\_t device\_type;

**union**

{

room\_info\_t room;

door\_info\_t door;

};

}device\_info\_t, \*p\_device\_info\_t;、

## IMP广告推送

**typedef** **struct** imp\_ad\_push\_t

{

**int** enable;

**int** play\_time;

**char** url[IMP\_AD\_PUSH\_URL\_MAX\_LEN];

}imp\_ad\_push\_t, \*p\_imp\_ad\_push\_t;

## 回调函数类型枚举

**typedef** **enum** callback\_handler\_type\_t

{

*CALLBACK\_ALARM\_IN* = 100,

*CALLBACK\_IMP\_TEXT\_PUSH* = 200,

*CALLBACK\_IMP\_AD\_PUSH*,

*CALLBACK\_INACTIVE\_CALL* = 300,

*CALLBACK\_VIDEO\_URL*,

*CALLBACK\_AUDIO\_DATA*,

*CALLBACK\_INACTIVE\_CALL\_HANG\_UP*,

*CALLBACK\_INACTIVE\_TALK\_HANG\_UP*,

*CALLBACK\_TALK\_CONFIM*,

*CALLBACK\_CALL\_RESPONSE\_BUSY*,

*CALLBACK\_UNLOCK\_CONFIRM*,

*CALLBACK\_MONITOR\_CONFIRM*,

*CALLBACK\_INACTIVE\_HANG\_UP\_MONITOR*,

*CALLBACK\_MONITOR\_RESPONSE\_BUSY*,

}callback\_handler\_type\_t;