# SDK对接引导v1.1.5

By TUTK FAE Team (SZ) 2020-1-1

\*本文档内容包含但不限于P2P SDK、KPNS推送服务，对文档的内容如有不理解之处，或者希望了解更多内容，欢迎随时咨询我们的技术人员，谢谢！

### 发布日志:

2021-11-09 V1.1.5:

1，新增UID自定义格式说明。

2021-07-13 V1.1.4:

1 , 新增Authkey&DTLS实作和兼容性设计

2021-04-19 V1.1.3：

1，新增kalay app扫码添加的流程

2021-04-19 V1.1.2：

1, 新增OTA功能的实作方法

1. 新增对讲主通道的实作方法
2. 新增基于AVAPIs的云存储实作方法

2020-12-17 V1.1.1：

1，新增日志相关描述。

2020-11-16 V1.1.0:

1，更新KPNS中文文档为KPNS\_20200728中文版。

2020-9-18 V1.0.9

1，增加对license key的说明。

2，增加对花屏原因说明。

2020-7-31 V1.0.8

1，修订部分错误。

2020-5-11 V1.0.7

1，调整部分KPNS相关内容，删除刷新设备列表时client的操作。

2，休眠与唤醒部分增加Hi3518EV300的说明。

3，更新KPNS推送说明文档。

2020-4-3 V1.0.6

1，新增-20033的报错说明。

2，新增公版APP的说明。

3，补充密码错误导致-20011的说明和例子讲解。

2020-1-15 V1.0.5:

1，新增配网方式说明。

2，新增event\_type=100的说明。

3，新增唤醒说明。

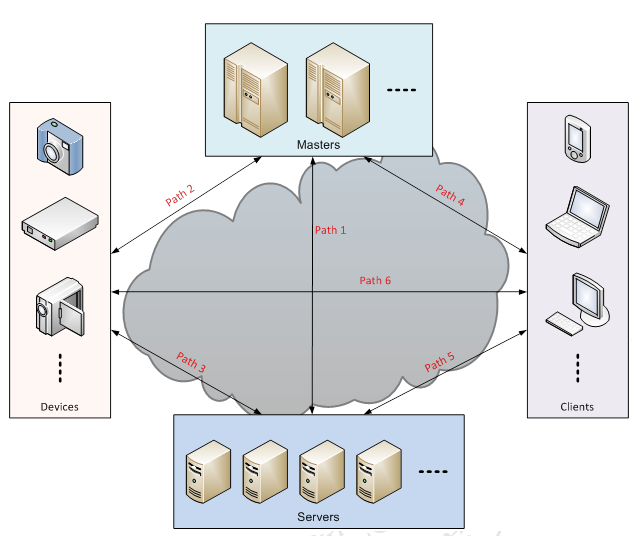
4，新增VPG/UID/SID说明。

2020-1-9 V1.0.4:

1. 新增KPNS相关内容。
2. 新增分区分流说明文档。
3. 增加-60的错误描述。

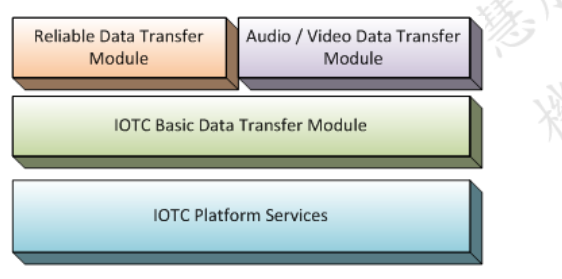
### 文档内容：

* [**选择使用哪个模块更好?**](#_选择使用哪个模块更好？)
* [**会话（Session）和通道（Channel）的说明**](#_会话（Session）和通道（Channel）的说明)
* [**该参考哪个sample code?**](#_三、该参考哪个sample_code？)
* [**公版APP怎么安装？**](#_四、公版APP怎么安装？)
* [**IOTCAPIs经常使用的APIs有哪些？**](#_四、IOTCAPIs经常使用的APIs有哪些？)
* [**AVAPIs经常使用的APIs有哪些？**](#_AVAPIs常使用的APIs有哪些？)
* [**RDTAPIs经常使用的APIs有哪些？**](#_六、RDTAPIs常使用的APIs有哪些？)
* [**TunnelAPIs经常使用的APIs有哪些？**](#_七、TunnelAPIs常使用的APIs有哪些？)
* [**TUTKGlobalAPIs常用的API有哪些？**](#_九、)
* [**如何自定义UID格式？**](#_十、如何自定义UID格式？)
* [**FAQ**](#_八、FAQS)



（P2P架构图）

## 选择使用哪个模块更好？



（模块关系图）

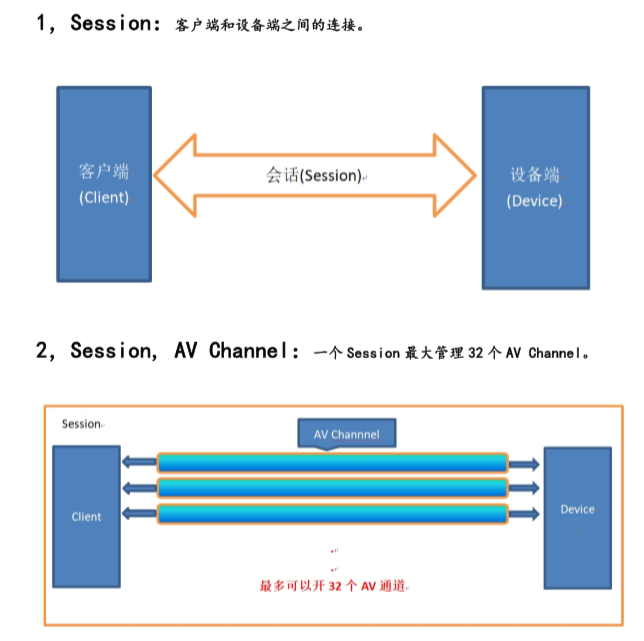
**IOTCAPIs:** 提供基础的连线和不可靠传输的API。设备端使用这个模块，可以登录服务器注册，以及监听客户端的连接。客户端使用这个模块，可以创建一条连接。一般仅用作P2P连线（和管理），不用以传输数据。**IOTCAPIs主要维护的是Session。**

**AVAPIs:** 提供音视频和控制指令传输的API。音视频的传输可以选择可靠和不可靠的方式传输，指令的传输是可靠的。通常用以传输音视频数据和控制指令，比如摄像头，门铃。**需要和IOTCAPIs搭配使用。AVAPIs主要维护的是Channel。**

**RDTAPIs:** 提供可靠的无协议的API，提供基本的Read和Write接口，和缓冲队列，但是因为传输过程可能会粘包，接收端需要自己去切包，比较适合文件下载这类应用。**需要和IOTCAPIs搭配使用。RDTAPIs主要维护的是Channel。**

**TunnelAPIs:** 提供代理的方式，接入RTSP、HTTP、TELNET等协议。底层使用的RDT通道。

## 会话（Session）和通道（Channel）的说明



（会话与AV通道）

1. 会话是客户端和设备端建立的一条连接，用一个会话ID标识（SID）。一个SID默认最多可以管理32个IOTC 通道。
2. 通道是传输的基本单位，AVAPIs和RDTAPIs都是直接使用通道进行数据传输。
3. 通常情况下，如果会话的状态发生变化，比如会话失效（包含掉线，或者主动关闭），应该停止数据的收发，并进行关闭通道，关闭会话的处理。

参考文件：[会话和通道的说明.pdf](file:///E:\整理的说明文件\SDK对接引导\SDK对接引导\详细项目\会话和通道的说明.pdf)

## 三、该参考哪个sample code？

1. AVAPI:

设备端(linux)：SDK\Sample\Linux\Sample\_AVAPIs\AVAPIs\_Server.c

设备端(android): SDK\Sample\Android\Sample\_AVAPIs\_Server

客户端：

Windows: SDK\Sample\Windows\Sample\_AVAPIs\_Client

Android: SDK\Sample\Android\Sample\_AVAPIs\_Client

IOS: SDK\Sample\iOS\Sample\_AVAPIs

(2) IOTC Wakeup:

设备端：SDK\Sample\Linux\Sample\_IOTCAPIs\IOTCAPIs\_WakeUp

客户端：较简单，只提供一个唤醒的API, 具体参考Readme.html

## 四、公版APP怎么安装？

安卓手机：应用宝、小米应用商店、谷歌市场，搜kalay。

苹果手机：app store，搜kalay S。

备注：

1，此APP仅支持使用AVAPI模块设备，不支持RDT, P2PTunnel协议的设备。

2，如果是要查看调试信息，可以在APP主页面-》点进某个设备-》设置-》勾选调试开关。勾选后，在实时观看的画面上方会出现一些图像相关的信息。

## 五、IOTCAPIs经常使用的APIs有哪些？

### 设备端：

（1）IOTC\_Initialize2 //必须用到

IOTC模块的初始化，参数填0，选择随机端口，程序启动时调用一次即可。

（2）IOTC\_DeInitialize //必须用到

IOTC模块的反初始化，程序退出时调用一次。

（3）IOTC\_Device\_LoginEx //必须用到

设备向服务器注册，早期版本使用IOTC\_Device\_Login的居多，建议使用IOTC\_Device\_LoginEx，因为此API提供更安全的authkey认证机制。

（4）IOTC\_Get\_Login\_Info

检查设备端的login状况，一般情况下获取到值为7，并且返回值没有持续增加，表示设备login正常，能被外网用户连线，常用作掉线重连机制的检测。

（5）IOTC\_Listen //必须用到

监听用户的连线。如果有用户连线，会返回一个会话标识（SID），后续的通道创建，都会使用到这个SID。

（6）IOTC\_Session\_Check

IOTC会话检查，可以获取本次会话的信息，如果远程IP, NAT, Mode等等。

（7）IOTC\_Session\_Close //必须用到

关闭一个会话，在一个连接掉线或者无效的时候，需要调用这个API关闭会话，释放资源。

（8）IOTC\_Set\_LanSearchPort

设定局域网搜索的端口，默认端口是32761，如果不想被其他家的APP搜索到自己的UID，可以通过这个API设定其他端口。

（9）IOTC\_Set\_Max\_Session\_Number

设定本端同时支持的最大的会话数，默认是128。也即同时可以支持128个用户同时访问。

（10）IOTC\_Set\_Log\_Path

设定日志保存的路径，只对T库有效，路径中需带日志文件名。

（11）IOTC\_Get\_Version

获取IOTCAPIs版本号（SDK版本号）

（12）IOTC\_Setup\_Session\_Alive\_Timeout

设定会话心跳超时时间，默认是60s，建议不要低于15s。需要和客户端一样

（13）IOTC\_Set\_Device\_Name

设定设备名，该设备名会在客户端做局域网搜索的时候搜索到。常用以隐藏UID的场景，以设备名代替。

### 客户端：

1. IOTC\_Initialize2 //必须用到

IOTC模块初始化，参数填0，使用随机端口，一般在程序启动时调用。

1. IOTC\_DeInitialize //必须用到

IOTC模块的反初始化，一般在程序结束时调用

1. IOTC\_Search\_Device\_Start、IOTC\_Search\_Device\_Stop、IOTC\_Search\_Device\_Result

用于搜索局域网下的设备

1. IOTC\_Get\_SessionID //必须用到

预分配一个SID，用于IOTC\_Connect\_ByUIDEx，如果不调用IOTC\_Connect\_ByUIDEx，将会导致SID泄漏。

1. IOTC\_Connect\_ByUIDEx //必须用到

用以连接设备，创建一个会话。正常会返回一个非负的值，等于参数中传入的SID。

1. IOTC\_Session\_Close //必须用到

关闭一个会话，在一个连接掉线或者无效的时候，需要调用这个API关闭会话，释放资源。

1. IOTC\_Connect\_Stop\_BySID

中止连接，因为IOTC\_Connect\_ByUIDEx是个阻塞的接口，如果想提前中止，可以用此API。

1. IOTC\_Session\_Check\_Ex

IOTC会话检查，可以获取本次会话的信息，如果远程IP, NAT, Mode等等。

1. IOTC\_Set\_Log\_Path

设定日志保存路径，仅对T库有效，路径中需带日志文件名

1. IOTC\_Setup\_Session\_Alive\_Timeout

设定会话心跳超时时间，默认是60s，建议不要低于15s。需要和设备端设置一样。

## 六、AVAPIs常使用的APIs有哪些？

### 设备端

1. avInitialize //必需使用

AV模块的初始化

1. avDeInitialize //必需使用

AV模块的反初始化

1. avServStartEx //必需使用

创建一个AV通道，用以数据传输。使用此接口，会建立一个全双工的通道。

1. avServStop //必需使用

关闭一个AV通道

1. avServExit

中止avServStartEx，因为avServStartEx是一个阻塞的接口，提前中止，调用此API

1. avSendAudioData

发送音频

1. avSendFrameData

发送视频

1. avSendIOCtrl

发送指令

1. avRecvIOCtrl

接收指令

1. avSendIOCtrlExit

中止指令发送

1. avRecvAudioData

接收音频，用以对讲

1. avResendBufUsageRate

获取重传缓存区使用率

1. avServResetBuffer

清空重传缓存区

### 客户端

（1）avInitialize //必需使用

AV模块初始化

（2）avDeInitialize //必需使用

AV模块反初始化

（3）avClientStartEx //必需使用

创建一个AV通道，用以收发数据，使用此接口，会建立一个全双工通道。

（4）avClientStop //必需使用

关闭一个AV通道

（5）avClientExit

中止通道创建，因为avClientStartEx是一个阻塞的接口，如需要提前中止，调用此API

（6）avRecvAudioData

接收音频

（7）avRecvFrameData2

接收视频

（8）avSendIOCtrl

发送指令

（9）avRecvIOCtrl

接收指令

（10）avSendIOCtrlExit

中止指令发送

（11）avSendAudioData

发送音频，常用以对讲给设备端发送声音。

## 七、RDTAPIs常使用的APIs有哪些？

1. RDT\_Initialize //必须使用

RDT模块初始化

1. RDT\_DeInitialize //必须使用

RDT模块反初始化

1. RDT\_Create //必须使用

创建一条RDT通道

1. RDT\_Destroy //必须使用

销毁一条RDT通道

1. RDT\_Read //必须使用

读取数据

1. RDT\_Write //必须使用

写数据

1. RDT\_Status\_Check

检查通道的情况

1. RDT\_Flush

立刻读取或者发送数据

## 八、TunnelAPIs常使用的APIs有哪些？

### 客户端：

1. P2PTunnelAgentInitialize //必须使用

Agent的初始化

1. P2PTunnelAgentDeInitialize //必须使用

Agent的反初始化

1. P2PTunnelAgent\_GetFreeSession //必须使用

获取一个空闲的会话ID

1. P2PTunnelAgent\_Connect\_Parallel //必须使用

建立连线

1. P2PTunnelAgent\_Connect\_Stop\_Parallel

终止正在执行的连线

1. P2PTunnelAgent\_Disconnect //必须使用

关闭连线

1. P2PTunnelAgent\_Abort

立刻关闭连线，不等待设备端

1. P2PTunnelAgent\_PortMapping //必须使用

端口映射

1. P2PTunnelAgent\_StopPortMapping/P2PTunnelAgent\_StopPortMapping\_byIndexArray //必须使用

关闭端口映射

### 设备端：

1. P2PTunnelServerInitialize //必须使用

设备端初始化

1. P2PTunnelServerInitialize //必须使用

设备端反初始化

1. P2PTunnelServer\_Start //必须使用

启动服务

1. P2PTunnelServer\_Stop //必须使用

关闭服务

1. P2PTunnelServer\_Disconnect //必须使用

关闭连线

1. P2PTunnelServer\_GetSessionInfo

获取连线信息

## [TUTKGlobalAPIs常用的API有哪些？](#_九、)

（1）TUTK\_SDK\_Set\_License\_Key

设置license key, license key由TUTK提供，用以SDK的使用授权和定制化服务。3.3.x以上版本目前必须要搭配license key使用，否则SDK无法使用，在IOTC\_Initialize2前调用。

## 十、如何自定义UID格式？

因为目前的UID都是按照VID+PID+GID的规则来生成的，不能随意修改。如果想在UID上植入一些信息，可以在原有的UID前面加上前缀。

比如UID为：UUUUUUUUUUUUUUUUUUUU，需要增加子客户和对应的区域，那么可以这样做：子客户名简写-区域简写-UID。

举例：CUSTOMER-CN- UUUUUUUUUUUUUUUUUUUU。

应用层在拿到这样的之后，需要把三部分分别取出，得到客户名，区域，UID，然后再做不同的处理。

也可以把用户信息生成Json字符串，然后做base64编码，比如：

{

“uid”:”uuuuuuuuuuuuuuuuuuuu”,

“region”:”cn”,

“customer”:”customer\_name”

}

经过base64编码后生成ID:

ew0KCeKAnHVpZOKAnTrigJ11dXV1dXV1dXV1dXV1dXV1dXV1deKAnSwNCgnigJxyZWdpb27igJ064oCdY27igJ0sDQoJ4oCcY3VzdG9tZXLigJ064oCdY3VzdG9tZXJfbmFtZeKAnQ0KfQ==

应用层可以按照生成规则，进行逆向的解密，得到各种定制的信息。

总之，方式是可以多种多样的，可以尽情发挥想象力去做各种各样的定制。

## 十一、FAQS

### 功能对接

#### 对讲

因为3.1.10.x版本已经支持双向通道（avClientStartEx/avServStartEx），所以不需要另外建立新的通道，即可以进行对讲，只需要发送start speaker和stop speaker即可打开关闭对讲。

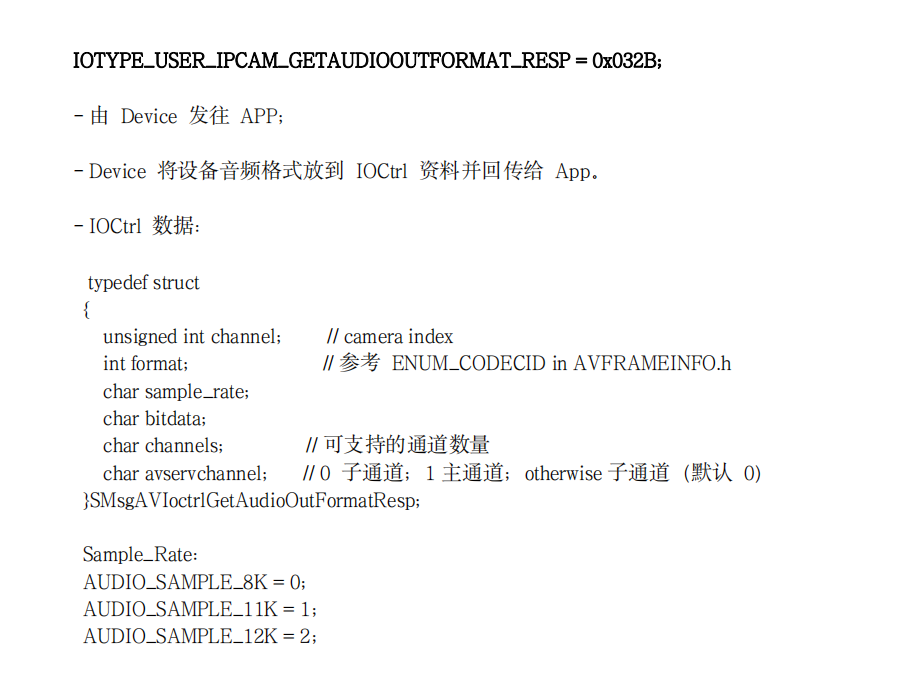
1. 如果是自行开发APP，可以使用这种方式，更简单。

(b) 如果是使用公版kalay app（V1.0）, 需使用旧的对讲方式，新建一条单向通道，以实现从APP往设备端传送音频，具体可以参考文件（SDK版本小于3.1.10.1）：[基于AVAPI的对讲流程.docx](file:///E:\整理的说明文件\SDK对接引导\SDK对接引导\详细项目\基于AVAPI的对讲流程.docx)

(c) 如果是使用公版kalay app（V2.0）, 则可以同时支持主通道对讲和子通道对讲，APP会通过以下命令来查询设备使用主通道还是子通道。

假如您手上的.h文件中的结构体与下面的不同，则可以手动修改成下面的格式。具体可以参考：[kalay app协议定义。](详细项目/公版APP协议_CN_20210409.pdf)





#### 事件回放

参考文件：[基于AVAPI的文件回放流程.docx](file:///E:\整理的说明文件\SDK对接引导\SDK对接引导\详细项目\基于AVAPI的文件回放流程.docx)

#### 文件下载

参考文件：[基于AVAPIs的文件下载流程.docx](file:///E:\整理的说明文件\SDK对接引导\SDK对接引导\详细项目\基于AVAPIs的文件下载流程.docx)

#### 消息通知（KPNS）

##### APP端：

参考文件：[KPNS\_20200505中文版.pdf](详细项目/KPNS_20200505中文版.pdf)

时机问题：

1. 添加设备: client + mapping（每隔一段时间需要更新token，重新执行此步）
2. 删除设备(单个)：rm\_mapping

##### 设备端：

参考sample code: SDK\Sample\Linux\Sample\_AVAPIs\AVAPIs\_Server.c

相关接口：

* GetPushMessageString()
* SendPushMessage() //向手机推送消息，随用随调。
* GetRegMessageString()
* SendRegister() //新版本已经不需要调用这个接口了

在SendRegister里面，调用send之后，建议加上recv，接收服务器的回应。如果不加recv，立刻关闭socket fd，可能会导致本次推送失败，因为send的数据可能还在系统缓存区里面，还没发出去。加recv第一能获取到服务器回应，第二可以有一个延时时间。

附一：常见的推送消息如下

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| event\_type | Lang - notification string | | | | |
| zh\_CN | enUS | fr\_FR | ja\_JP | zh\_TW |
| 1或者4 | 移动侦测 | Motion Detection | Détection de mouvement | モーション検知 | 位移偵測 |
| 2000 | 用户呼叫 | User call |  | ユーザー呼び出し | 用戶呼叫 |
| 55 | 门磁报警 | door alarm |  |  |  |
| 37 | 您的门铃已经断线，请尽速确认! | Your doorphone is not connected with network, please confirm soon. |  |  | 您的門鈴已經斷線，請盡速確認! |
| 52 | 影像遗失 | video loss |  |  |  |
| 100 | 空白字串，配合&msg=自定义消息使用 | | | | |

附二：国内外推送服务以及对应的域名：

1. APNS： push.iotcplatform.com
2. FCM: push.iotcplatform.com
3. （国内）极光、信鸽、华为，小米、oppo：push.kalay.net.cn

端口：http是80/7379，https用7380

附三：各个推送服务需要提供的资料给TUTK：

1. APNS: bundle id、pem证书、pem类型（dev or dis）、pem证书密码（如果有密码的话，需要提供，没有则不需要，建议不设密码）
2. FCM: APPID、服务器密钥
3. 信鸽：APPID、APP ID、SECRET KEY

#### 动态码流

1. 动态码流：主要是让码流大小能适应不同网络状况，或者不同带宽大小，以便终端用户能较流畅平滑地观看图像。通常是在设备端进行处理。
2. 动态码流的依据：传输状况。
3. 相关API: avResendBufUsageRate()。检查重传缓存区使用率，越大，说明传输越不顺畅。
4. 动态码流要调整的参数：最终要调整的是bps，通常可以直接调整bps，也可以调整fps，或者两者同时调整。一般两者同时调整，既可以保持较好的流畅度，也可以保持较好的画质。通常要保证的优先级是：流畅度 > 画质
5. 清理缓存区：avServResetBuffer，建议在另外的线程进行，以免影响实时流的传输！
6. BPS/FPS说明：会跟连线模式相关。

RLY下建议：720P不超过1Mbps，480P不超过750kbps, 360P不超过500kbs。Fps在10~20fps之间，取中值15fps左右。

通常P2P/LAN不受服务器影响，所以网络会更稳定，bps和fps可以往上调，以获得更好的画质和流畅度体验。

1. 动态码流在调整的时候，不宜太频繁。并且上调的时候幅度要小，下调的时候幅度可以大一些。

#### Authkey和DTLS实作和兼容性设计。

可以参考文档：[文档](详细项目/Authkey&DTLS%20使用和兼容性设计.docx)。

#### 分区分流（含符合GDPR规范）

目的：符合GDPR规范

SDK版本要求：> 3.1.10.1

做法：

1. APP和设备端调用IOTC\_Set\_Master\_Region（）设定区域。
2. 服务器配置到对应区域的master。

注意: app，设备，P2P服务器必须配置一样的区域，否则可能连线失败！

参考文件：[分区分流说明文档.pdf](详细项目/分区分流说明文档.pdf)

#### 休眠与唤醒

具体代码可以参考：[sample code](#_三、该参考哪个sample code？)，说明：

##### 休眠

* 目前支持UDP和TCP的方式，心跳包大小40个字节，唤醒包大小48个字节。
* TCP保活，服务器端可用端口：80、8000、8080、443、21047，UDP用10001。
* 心跳包间隔：UDP的方式参考服务器下发的interval，TCP的方式可以比interval大。
* 保活的时候，建议最少同2台服务器进行保活，以增加唤醒的可靠性。

##### 唤醒

APP端直接调用对应的API唤醒就可以了。不用管UDP还是TCP。

备注：Hi3518EV300 liteOS的唤醒需要进行全字节比对，需要配合新版本的服务器程序和P2P SDK，具体版本可以咨询TUTK相关的技术人员！

#### 固件升级

##### 升级包上传

1. 先跟我们的业务申请OTA的后台账号，该账号后期用以管理固件的升级包版本发布。
2. 拿到账号后，按照[手册](详细项目/OTA服务器使用手册.pdf)把升级包上传到服务器后台。
3. 手动测试能否获取到下载链接。

测试链接(格式)： https://{domain\_name}/ota/GET/i/{vender\_name}/{product\_name}/{model\_name}

-CN: https://kota.kalay.net.cn/ota/GET/i/{vender\_name}/{product\_name}/{model\_name}

-US: https://kota.kalayservice.com/ota/GET/i//{vender\_name}/{product\_name}/{model\_name}

例子：厂商tutksz ota相关信息如下：

厂商名：tutksz

产品名：TUTKCN

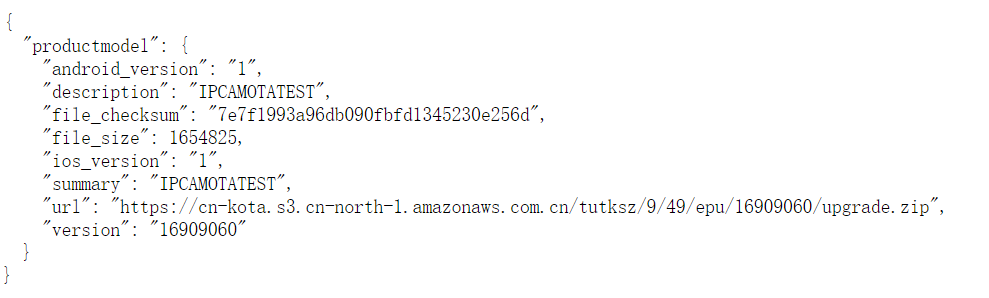
型号：CNModelA

区域：CN

## 测试上传的升级包最新版本信息方法如下：

在浏览器输入：<https://kota.kalay.net.cn/ota/GET/i/tutksz/TUTKCN/CNModelA>

服务器返回如下信息：



## 测试下载：

将<https://cn-kota.s3.cn-north-1.amazonaws.com.cn/tutksz/9/49/epu/16909060/upgrade.zip>复制到浏览器，看是否能下载成功。

##### 实作流程：

1. APP在相关页面会发送IOTYPE\_USER\_IPCAM\_DEVICE\_INFO\_REQ查询该设备的相关信息（厂商、产品名、型号、版本）
2. APP会向服务器查询相关的厂商+产品名+型号的固件最新版本信息。

（3）如果查询到的最新版本高于设备当前的版本，则通过IOTYPE\_USER\_IPCAM\_OTA\_REQ将升级包的链接发送给设备，设备进行远程升级。

#### 云存储

Kalay 云存储是基于AVAPIs+公版IOCtrl协议的来实作，所以设备端必须按照按照公版cmd，基于AVAPIs去实现云存储。

* SDK版本要求：3.3.0以上；如果是H265，则必须使用3.4.0以上SDK；
* 支持的视频格式：H264、H265；
* 音频格式：PCM、AAC、G711A/U
* 设备端demo路径: sdk\_path\Sample\Linux\Sample\_VSaaS\_AV

基于3.3.0.x实现H264编码格式云存储：

char starttime[MAX\_TIMESTAMP\_LENGTH + 1] = {0};

VSaaSPullStreamAttr attr;

attr.pullstream\_type = VSAAS\_EVENT\_GENERAL;

if (VSAAS\_EVENT\_TYPE) // 以回放的方式来传流，本地先录制到SD卡，然后服务器以回放的方式来拉流

{

sprintf(starttime, "%lu", timestamp\_sec); // timestamp\_sec为秒级时间戳

strcpy(attr.starttime, starttime);

}

else //以实时播放的方式来传流，这种方式会在云端存储的视频流会有延时。

{

strcpy(attr.starttime, "live"); //

}

ret = avServNotifyVSaasPullStream(&attr, 0);

基于3.4.0.x实现H264/H265编码格式云存储：

char att\_json\_str[2048] = {0};

unsigned long interval\_ms = timestamp\_sec - gLastVsaasEventTimestamp\_sec;

int ret = 0;

NebulaJsonObject \*attr\_obj = NULL;

if (VSAAS\_EVENT\_TYPE) {// meaning the video is playback

sprintf(att\_json\_str, "{\"starttime\":\"%lu\",\"protocol\":\"tutk\",\"event\_id\":\"%d\",\"event\_file\":\"%s\",\"media\_type\":\"0\"}", timestamp\_sec, VSAAS\_EVENT\_GENERAL, gRecordFileName);

}

else {// meaning the video is live stream.

sprintf(att\_json\_str,"{\"starttime\":\"live\",\"protocol\":\"tutk\",\"event\_id\":\"%d\",\"media\_type\":\"0\"}",VSAAS\_EVENT\_GENERAL);

}

Nebula\_Json\_Obj\_Create\_From\_String(att\_json\_str, &attr\_obj);

ret = avServNotifyCloudRecordStream(attr\_obj, 3000, NULL);

备注：

视频编码为H264格式，media\_type填0。

视频编码为H265格式，mdeia\_type填1。

### 错误码

#### IOTCAPIs

##### -19

解释： 此错误码表示，该UID所在的所有的P2P服务器中，只收到了部分的回应，所回应的服务器都是回复的设备不在线（-90）。

##### -42

解释：无法建立IOTC连线，包含LAN,P2P,RLY。可能的原因如下：

1. 网络不稳定，防火墙阻挡。
2. 设备可能已经掉线，这种情况下，过5~10分钟再连应该会有-90的报错。
3. 手机或者设备端网络可能会IP或者port会跳动。
4. 服务器的流量紧张。

##### -90

解释：设备从服务器上掉线。常见的因素是网络不稳定，某些SDK版本(3.1.5.38以及之前的版本)的bug也会导致。

##### -60

解释：Master server没有回应。可能的原因如下：

1. p2p server故障，含没架设、宕机等情况，这种情况下，通常会大面积出现无法使用。
2. 外网不通不稳定，在3.1.10.x的版本会有此报错，这个通常只是个别现象。

#### AVAPIs

##### -20000

参数SID/avIndex无效，通常的原因是本端关闭了相关的通道或者会话。

##### -20006

网络传输不顺畅，通常是带宽不足，网络不稳定，连线异常断开导致。

处理可以参考动态码流部分。

##### c)-20011

超时，主要出现在avClientStart和avServStart时，常见的原因如下：

1. 网络不顺畅
2. 没有在对方的timeout时间内启动start
3. 两端使用的IOTC Channel ID不同
4. avClientStart使用了错误的密码，avServStart会返回超时
5. 超时时间太短

**备注：**

1. **针对原因D所指密码不对的情况，有一种出现在avServStartEx里面。因为这个API里面比对密码的方式与以往不同，基于安全性原则，这个新的API验证密码的过程是由SDK底层完成（而不是SDK底层上抛给应用层比对），应用层在回调函数里面，需要把预设的密码，拷贝到底层的buffer。如果回调函数填成NULL，会导致-20011出现，对端出现-20009。**

Linux里面的例子avServStartEx所使用的回调函数如下：

*int ExPasswordAuthCallBackFn(const char \*account, char \*pwd, unsigned int pwd\_buf\_size)*

*{*

*//验证account是否正确，此步可做可不做，也可以让对方通过account携带额外的信息，来实现不同的account，做不同的权限管理。比如对account为guest和admin的用户，来做不同的密码比对和权限管理*

*if (strcmp(account, gAvAccount) != 0)*

*return -1;*

*//检查底层缓存区大小是否足够，足够的话，把密码拷贝到缓存区。*

*if (pwd\_buf\_size <= strlen(gAvPassword)) return -1;*

*strcpy(pwd, gAvPassword);*

*//正常返回*

*return 0;*

*}*

1. **对于-20011这个报错，通常需要跟对端联调才好排查。**

##### d) -20012

没有收到数据，主要出现在avRecvFrameData2和avRecvAudioData，通常是正常的，因为这两个API是检查本地的buffer是否有完整的帧数据，频率会远高于设备端的发送频率。通常的处理方法是sleep 5 ms。

如果发送方和接收方在不同通道收发，也会报这个错误。

##### -20015

远程关闭连接，是应用层的主动调用IOTC\_Session\_Close关闭连线的行为。通常的处理方法是关闭IOTC连线。

##### -20016

连接心跳超时，SDK默认是60秒没有收到对方心跳，会抛出此报错。通常可能的原因为：

--网络切换

--网络不稳

--对方程序异常退出

--宕机

通常的处理方法是关闭IOTC连线。

##### g) -20027

所使用的IOTC Channel ID已经被占用。

通常的处理方法是换一个IOTC Channel ID。

##### h) -20039

如果是AvClientStartEx / avServStartEx返回此报错，把参数结构体中的[security\_mode](F:/mysharefiles/TUTK_Kalay_Platform_SDK_3.1.10.5_nossl/UserManual/APIRefGuide/struct_a_v_serv_start_in_config.html#a0a65729e56bab4337a402f386f44e994)改为AV\_SECURITY\_SIMPLE即可，因为标准的SDK不支持DTLS加密。

#### RDTAPIs

##### -10007

#### TunnelAPIs

#### TUTKGlobalAPIs

##### -1005

未设置license key，在3.3.x版本出现，因为3.3.X版本SDK增加license key，用以SDK的授权和定制化服务。需在IOTC\_Initialize2前调用[TUTK\_SDK\_Set\_License\_Key](#_TUTKGlobalAPIs常用的API有哪些？)设置。

### 问题排查

#### 无法连线

无法连线，一般需要确认是无法建立IOTC连线，还是无法建立相关通道。具体需要参考API返回的错误码针对性排查。

#### 无法出图

无法出图，需要先确认连线是否正常，通道建立是否成功，指令交互是否正常。一般遇到这种情况，方便的话，可以切换其他网络测试下，或者重启下客户端。

如果切换网络仍然出现无法出图的情况，可以结合设备端进行同步的调试，确认传输是否有问题。

#### 花屏(灰屏)

花屏一般是因为有丢帧，通常可以从以下方面进行检查：

1. APP和设备端是否开启重传。
2. 设备端avSendFrameData是否有报错。
3. APP端或者设备端是否有调用API进行clean buffer。
4. 解码是否有是失败。

#### 搜不到局域网下设备

1. 某些版本的兼容性问题，如果两端都是3.1.5.x以及之后的版本，应该无此问题。
2. 某些版本的bug，比如设备端版本（3.1.10.1~3.1.10.6） 在AP mode下会有搜索不到UID的bug。
3. APP端或者设备端有修改过局域网搜索端口。

### 其他

#### UID、VPG、SID是什么关系？

先有VPG，才会有UID和SID。VPG是一个群组的概念，相当于一个班级；UID是群组里面的一份子，相当于这个班级里面的一个学生。通常一个VPG里面会有很多UID（跟一个班级会有多个学生一样），这种设计是为了方便管理一批UID。

SID是由VPG衍生出来的，相当于VPG的别名，不过带了序号。SID是用来把一个VPG绑定一台或者多台P2P服务器。

#### IOS手机退到后台要进行怎样的处理？

由于IOS手机在app退到后台（大概3分钟左右）后，会回收app资源，导致app socket失效。可能会出现一些问题，比如说crash, p2p无法穿透。

**一般建议app在退到后台2分钟以上的时候，做如下处理：**

1. 停止AV以及IO数据的收发。
2. 关闭所有的AV通道
3. 关闭所有的IOTC连接
4. 反初始化。

**APP返回前台后，做如下处理：**

进行初始化，然后重连设备。

#### 手机切换网络要进行怎样的处理？

手机切换网络，通常会导致已经存在的连接掉线，但是由于在切换的时候，app无法及时通知到设备端，所以一般需要1分钟（超时时间可以设置）左右，两端才会返回对应的掉线错误：-20016。通常的处理是**重连**设备！

#### 设备端如何增加掉线重连的机制？

目前是设备端可以通过IOTC\_Get\_Login\_Info获取当前的登录状态。在掉线的情况下，IOTC\_Get\_Login\_Info的返回值会持续增大（代表login失败的次数）。如果一段时间内一直增大，那么很可能设备已经从服务器掉线。这个时候，可以考虑重启IOTC服务。

流程如下：

1. 判断当前是否有用户在访问设备，如果没有，进行下面操作
2. 判断当前网络是否畅通，通常可以通过ping判断，如果网络正常，进行下面操作
3. IOTC\_Listen\_Exit，先终止IOTC\_Listen
4. IOTC\_Listen会返回[IOTC\_ER\_EXIT\_LISTEN](file:///F:\mysharefiles\TUTK_Kalay_Platform_SDK_3.1.10.5_nossl\UserManual\APIRefGuide\_i_o_t_c_a_p_is_8h.html#a0b41b492208a847f6fc1c8f040416d15)，此时进行进行下一步操作
5. 反初始化
6. 初始化
7. 重新login和listen

#### avIndex怎么释放可能会更好？

因为avIndex是公共资源，使用完必须释放，目前释放的操作由应用层处理。正常可以在avRecvIOCtrl的线程处理，在检测到表明连线失效的情况下，先停止AV以及IO数据的收发，然后调用avStop关闭avIndex。

#### APP端解码要注意什么？

对于一般的H264和H265格式的视频帧，P帧依赖于前面的帧，所以第一帧会从收到的第一个I帧开始解码。中途如果avSendFrameData有报错，或者有做clean buffer，那么设备端需要从I帧开始发送，否则会出现花屏或者灰屏。如果没开启重传，检查到不完整帧或者丢帧，需要丢掉后面的P帧，从I帧开始解码。每个I帧前需要带SPS、PPS等编码参数。

注意：切换分辨率后，设备要从I帧开始送帧。

#### AV通道如何开启重传？

以下API可以设定Resend参数。

avClientStartEx

avServStartEx

如果使用的是早期的API：

重传开启的组合是avClientStart2 + avServStart3。

#### KPNS如何快速的绑定或者解绑多个设备？

Mapping和unmapping一次只能绑定或者解绑一个UID。而sync指令可以一次绑定或者解绑多个UID。

PS：目前尚无指令可以一次解绑某个UID下所有的设备。只能做好一次解绑掉某个设备绑定的所有的UID。

#### 常见的配网方式有哪些？

###### 简易WIFI设定

需要集成P2P SDK，设备先处于AP mode(热点模式)，手机连上设备AP，然后APP进行局域网搜索，搜索到设备的UID，然后建立IOTC连线，创建AV通道。最后发送WIFI信息给设备进行配网，设备切换到station mode（工作站模式）。单次操作成功率高。

###### 扫码配置

APP把需要配置的WIFI信息，生成二维码，然后由设备进行扫二维码配置。单次操作成功率高。

公版kalay app扫码流程可以参考：[流程](详细项目/kalay%20app扫码添加流程说明.doc)

###### 声波配置

需要集成声波配置的SDK。单次操作成功率较高。

###### Smartlink/一键WIFI配置

需要集成smartlink的SDK。单次操作成功率较低。

#### 如何记录日志？

##### SDK的日志分两部分，一是应用层的日志，一是SDK的日志。综合利用两种日志，对于问题的排查，会有很大的帮助。

1. 应用层的日志，即应用层自己产生的日志，可以记录于文件中或者自己的日志服务器，在出问题时对问题的初步排查非常重要。
2. SDK的日志，SDK的T库（库名后带T的版本），会产生对应的日志。目前IOTCAPI、AVAPI、RDTAPI、P2PTunnelAPI都有提供T库，以及日志设置的API，不同的API分别用以设置本模块的日志的参数。API如下：

IOTC\_Set\_Log\_Attr

AV\_Set\_Log\_Attr

RDT\_Set\_Log\_Attr

P2PTunnelAgent\_Set\_Log\_Path

需要注意的是，这些API只适用于T库，且path中需包含文件名，否则无法产生日志。如果path设置为NULL，那么将关闭日志。