# iOS LCOpenSDK开发指南

**修订历史（Revision history）**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| 编号 | 修订内容描述 | 修订人 | 修订日期 | 修订后版本号 | 批准人 |  |
| 1 | 初稿 | 39890 | 2019/08/01 | 3.X |  |  |
| 2 | 修订 | 119607 | 2019/12/25 | 3.X.1 |  |  |
| 3 | 修订 | 119607 | 2020/08/17 | 3.X.2 |  |  |
| 4 | 修订 | 119607 | 2020/10/20 | 3.10.1 |  |  |
| 5 | 修订 | 21589 | 2021/8/24 | 3.11.6 |  |  |
| 6 | 修订 | 311613 | 2021/10/28 | 3.11.8 |  |  |
| 7 | 修订 | 311613 | 2022/01/20 | 3.11.9 |  |  |
| 8 | 修订 | 357140 | 2022/11/11 | 3.12.0 |  |  |
| 9 | 修订 | 347885 | 2023/08/23 | 3.12.05 |  |  |
| 10 | 修订 | 317444 | 2023/11/29 | 3.12.06 |  |  |
| 10 | 接入S2V可视对讲设备  主要功能：  Demo：1.首页新增S2V模拟设备呼叫入口2.预览页新增S2V可视对讲入口3.新增S2V可视对讲页面：LCVisualTalkViewController  SDK：LCOpenSDK\_AudioTalk新增对讲方法：- -(NSInteger)playVisualTalk:(LCOpenSDK\_ParamTalk \*)paramTalk videoSampleCfg:(LCOpenSDK\_MediaVideoSampleConfigParam \*)videoSampleCfg;  enableTalkVideo()  enableTalkVideo(...)  StartSampleAudio()  StopSampleAudio()  PushMediaData(...) | 357089 | 2024/04/01 | 3.12.07 |  |  |

# 目录

[iOS LCOpenSDK开发指南 1](#_Toc830730911)

[目录 2](#_Toc1303232030)

[概述 9](#_Toc1235012457)

[1 SDK安装 10](#_Toc1424916544)

[1.1 CocoaPods安装 10](#_Toc1982207311)

[1.2 官网下载 10](#_Toc1044460066)

[1.3 权限配置 11](#_Toc708998684)

[2 主要流程 12](#_Toc1901608432)

[2.1 初始化 12](#_Toc1481281970)

[2.1.1 SDK初始化 12](#_Toc128150119)

[2.1.2 配置SDK日志级别 12](#_Toc2040435739)

[2.2 设备配网 13](#_Toc437106430)

[2.2.1 SmartConfig配网添加 13](#_Toc2053696270)

[2.2.2 软Ap配网 14](#_Toc2116035306)

[2.2.3 蓝牙配网 15](#_Toc1876193622)

[2.3 实时预览 16](#_Toc1683816053)

[2.4 录像回放 17](#_Toc356902605)

[2.5 语音对讲 21](#_Toc540256164)

[2.6 录像下载 22](#_Toc524488832)

[2.7 数据解密 25](#_Toc1810912136)

[3 SDK接口说明 26](#_Toc1862024468)

[3.1 概述 26](#_Toc1913529592)

[3.2 LCOpenSDK\_Api 27](#_Toc2124238919)

[3.2.1 shareMyInstance 27](#_Toc167880258)

[3.2.2 ­initOpenApi 27](#_Toc1917467695)

[3.2.3 uninitOpenApi 28](#_Toc1739942983)

[3.2.4 sdkVersion 28](#_Toc936894082)

[3.3 LCOpenSDK\_Bluetooth 28](#_Toc1028736370)

[3.3.1 startAsyncBLEConfig 28](#_Toc581328593)

[3.4 LCOpenSDK\_ConfigWifi 28](#_Toc1486552348)

[3.4.1 configWifiStart 28](#_Toc660563638)

[3.4.2 configWifiStop 29](#_Toc1750092523)

[3.5 LCOpenSDK\_IotApConfig 29](#_Toc1869004749)

[3.5.1 startAsyncIotApConfig 29](#_Toc1119511774)

[3.6 LCOpenSDK\_DeviceInit 29](#_Toc1530154251)

[3.6.1 ­initDev 29](#_Toc1185823732)

[3.6.2 checkPwdValidity 30](#_Toc1491219564)

[3.6.3 loginDeviceByIP 30](#_Toc1793051658)

[3.7 LCOpenSDK\_Download 30](#_Toc181197655)

[3.7.1 shareMyInstance 30](#_Toc257176139)

[3.7.2 setListener 31](#_Toc1622270409)

[3.7.3 getListener 31](#_Toc1046381751)

[3.7.4 startDownloadCloudRecord 31](#_Toc794503774)

[3.7.5 startDownloadDeviceRecordById 32](#_Toc171612572)

[3.7.6 startDownloadDeviceRecordByUtcTime 32](#_Toc221959683)

[3.7.7 stopDownload 33](#_Toc297297342)

[3.8 LCOpenSDK\_DownloadListener 33](#_Toc1629464072)

[3.8.1 onDownloadReceiveData 33](#_Toc1691191560)

[3.8.2 onDownloadState 34](#_Toc1910480875)

[3.9 LCOpenSDK\_EventListener 34](#_Toc276576181)

[3.9.1 onControlClick 34](#_Toc1261261959)

[3.9.2 onWindowDBClick 34](#_Toc218665376)

[3.9.3 onZoomBegin 35](#_Toc764454415)

[3.9.4 onZooming 35](#_Toc1938176551)

[3.9.5 onZoomEnd 35](#_Toc1901334961)

[3.9.6 onWindowLongPressBegin 35](#_Toc1180022167)

[3.9.7 onWindowLongPressEnd 35](#_Toc621080724)

[3.9.8 onSlipBegin 36](#_Toc1733203848)

[3.9.9 onSlipping 36](#_Toc1488885428)

[3.9.10 onSlipEnd 36](#_Toc1217933552)

[3.9.11 onPlayerResult 36](#_Toc2142568907)

[3.9.12 onResolutionChanged 37](#_Toc1149827053)

[3.9.13 onReceiveData 37](#_Toc2085424065)

[3.9.14 onStreamCallback 37](#_Toc641657768)

[3.9.15 onPlayBegan 38](#_Toc1826715189)

[3.9.16 onPlayFinished 38](#_Toc1175964011)

[3.9.17 onPlayerTime 38](#_Toc1135129536)

[3.9.18 onFileTime 38](#_Toc2024875251)

[3.9.19 onIVSInfo 38](#_Toc904989548)

[3.10 LCOpenSDK\_Log 39](#_Toc1680145182)

[3.10.1 shareInstance 39](#_Toc937599471)

[3.10.2 setLogInfo 39](#_Toc2146791058)

[3.10.3 closeLog 39](#_Toc1244558559)

[3.11 LCOpenSDK\_LoginManager 40](#_Toc804979333)

[3.11.1 shareInstance 40](#_Toc140673631)

[3.11.2 addDevices 40](#_Toc2069704517)

[3.11.3 getP2pPort 40](#_Toc583703113)

[3.12 LCOpenSDK\_PlayBackListener 41](#_Toc592920695)

[3.12.1 onPlayPause 41](#_Toc893998785)

[3.12.2 onPlayFinished 41](#_Toc1641985083)

[3.12.3 onRecordStop 41](#_Toc1678426031)

[3.12.4 onPlayerTime 41](#_Toc2108599672)

[3.13 LCOpenSDK\_PlayBackWindow 41](#_Toc1459544510)

[3.13.1 setPlayBackListener 41](#_Toc2006363536)

[3.13.2 playCloudRecord 42](#_Toc1163724358)

[3.13.3 playDeviceRecordByFileName 42](#_Toc1581711677)

[3.13.4 playDeviceRecordByUtcTime 43](#_Toc128089126)

[3.13.5 stopRecordStream 43](#_Toc1015326388)

[3.13.6 setPlaySpeed 44](#_Toc685544054)

[3.13.7 getPlaySpeed 44](#_Toc689149423)

[3.13.8 addToPlayGroup 44](#_Toc1155044090)

[3.13.9 hideVideoRender 44](#_Toc1721335397)

[3.14 LCOpenSDK\_PlayFileListener 44](#_Toc1731808642)

[3.14.1 onPlayPause 44](#_Toc1661978303)

[3.14.2 onPlayFinished 45](#_Toc549541992)

[3.14.3 onPlayerTime 45](#_Toc1972577444)

[3.14.4 onFileTime 45](#_Toc256558922)

[3.15 LCOpenSDK\_PlayFileWindow 45](#_Toc1986122525)

[3.15.1 setPlayFileListener 45](#_Toc275468707)

[3.15.2 setPlaySpeed 45](#_Toc1975299264)

[3.15.3 PlayFile 46](#_Toc905031075)

[3.15.4 stopFile 46](#_Toc230605824)

[3.16 LCOpenSDK\_AudioTalk 46](#_Toc1731584780)

[3.16.1 setListener 46](#_Toc47013316)

[3.16.2 getListener 46](#_Toc2026303563)

[3.16.3 playTalk 46](#_Toc1288309215)

[3.16.4 stopTalk 47](#_Toc1682847451)

[3.16.5 playSound 47](#_Toc1257477967)

[3.16.6 stopSound 47](#_Toc1045621242)

[3.17 LCOpenSDK\_PlayGroupManager 47](#_Toc897530893)

[3.17.1 shareInstance 47](#_Toc876582123)

[3.17.2 createPlayGroup 48](#_Toc977922841)

[3.18 LCOpenSDK\_PlayListenerProtocol 48](#_Toc1256838196)

[3.18.1 onPlayLoading 48](#_Toc1030408280)

[3.18.2 onPlayBegan 48](#_Toc763832552)

[3.18.3 onPlayStop 48](#_Toc76459698)

[3.18.4 onPlayFail 48](#_Toc862923380)

[3.18.5 onResolutionChanged 49](#_Toc1196179469)

[3.18.6 onReceiveData 49](#_Toc1593915916)

[3.18.7 onStreamCallback 49](#_Toc1233787534)

[3.19 LCOpenSDK\_PlayRealListener 49](#_Toc164988506)

[3.19.1 onPtzLimit 49](#_Toc560432065)

[3.19.2 onRecordStop 49](#_Toc318440713)

[3.20 LCOpenSDK\_PlayRealWindow 50](#_Toc503815067)

[3.20.1 setPlayRealListener 50](#_Toc91810948)

[3.20.2 playRtspReal 50](#_Toc1173344490)

[3.20.3 stopRtspReal 50](#_Toc58513029)

[3.20.4 controlMovePTZ 51](#_Toc2028451724)

[3.20.5 addToPlayGroup 51](#_Toc885229143)

[3.20.6 hideVideoRender 51](#_Toc279499985)

[3.21 LCOpenSDK\_ PlayWindowProtocol 51](#_Toc1009511906)

[3.21.1 initPlayWindow 51](#_Toc1745792842)

[3.21.2 index 52](#_Toc471226533)

[3.21.3 uninitPlayWindow 52](#_Toc2132133642)

[3.21.4 setWindowFrame 52](#_Toc1857987252)

[3.21.5 setSurfaceBGColor 52](#_Toc632033337)

[3.21.6 getWindowView 52](#_Toc1130176897)

[3.21.7 setTouchListener 52](#_Toc390250164)

[3.21.8 openTouchListener 53](#_Toc519448410)

[3.21.9 playAudio 53](#_Toc848401815)

[3.21.10 stopAudio 53](#_Toc1945372272)

[3.21.11 snapShot 53](#_Toc433249929)

[3.21.12 setStreamCallback 53](#_Toc1661993373)

[3.21.13 doScale 53](#_Toc802823482)

[3.21.14 getScale 54](#_Toc414507873)

[3.21.15 doTranslateX 54](#_Toc196870643)

[3.21.16 seek 54](#_Toc1680080521)

[3.21.17 pause 54](#_Toc1998325691)

[3.21.18 resume 54](#_Toc1363133204)

[3.21.19 startRecord 55](#_Toc824213432)

[3.21.20 stopRecord 55](#_Toc1285628474)

[3.21.21 setSEnhanceMode 55](#_Toc1724790051)

[3.22 LCOpenSDK\_SearchDevices 55](#_Toc1812119951)

[3.22.1 shareSearchDevices 55](#_Toc686934703)

[3.22.2 startSearchDevices 55](#_Toc439467049)

[3.22.3 stopSearchDevices 56](#_Toc926430510)

[3.22.4 getLastError 56](#_Toc1261140820)

[3.23 LCOpenSDK\_SoftApConfig 56](#_Toc330165850)

[3.23.1 startSoftAPConfig 56](#_Toc2147180749)

[3.23.2 getSoftApWifiList 57](#_Toc1351644255)

[3.23.3 queryWifiByLoginHandle 57](#_Toc1002975819)

[3.23.4 querySupportWlanConfigV3 57](#_Toc1415444630)

[3.24 LCOpenSDK\_TalkerListener 58](#_Toc1701538591)

[3.24.1 onTalkResult 58](#_Toc1866855485)

[3.24.2 onAudioRecord 58](#_Toc1504053725)

[3.24.3 onAudioReceive 58](#_Toc600947238)

[3.25 LCOpenSDK\_TouchListener 59](#_Toc504637225)

[3.25.1 onControlClick 59](#_Toc1024918572)

[3.25.2 onWindowDBClick 59](#_Toc840107017)

[3.25.3 onWindowLongPressBegin 59](#_Toc2121139341)

[3.25.4 onWindowLongPressEnd 59](#_Toc1760363987)

[3.26 LCOpenSDK\_Utils 60](#_Toc555324790)

[3.26.1 decryptPic 60](#_Toc379815668)

[3.26.2 decryptPic 60](#_Toc1240533192)

[3.26.3 decryptPic 61](#_Toc1870112868)

[3.26.4 mtsPreKeepalive 61](#_Toc416314984)

[3.27 LCOpenSDK\_Netsdk 61](#_Toc504214162)

[3.27.1 initSDKWithCallback 61](#_Toc356949672)

[3.27.2 logOpen 62](#_Toc1331311233)

[3.27.3 logClose 62](#_Toc715774938)

[3.27.4 initDevAccountWithDevInfo 62](#_Toc1973476119)

[3.27.5 logout 62](#_Toc328204118)

[3.27.6 resetPassword 62](#_Toc1388605730)

[3.27.7 queryPasswordResetType 63](#_Toc1591712161)

[3.27.8 queryProductDefinitionInfo 63](#_Toc702499248)

[3.27.9 queryDevicePtzAuth 63](#_Toc39769930)

[3.28 LCOpenSDK\_BuryPoint 63](#_Toc545799293)

[3.28.1 addBuryPointListener 63](#_Toc1346061114)

[3.28.2 removeBuryPointListener 64](#_Toc1656405500)

[3.29 LCOpenSDK\_BuryPointProtocol 64](#_Toc1376722439)

[3.29.1 buryPoint 64](#_Toc1585219495)

# 概述

乐橙是一家专注视频智能硬件、视频云和视频智能技术，为视频应用提供云视频能力的服务商。iOS LCOpenSDK作为一整套接入乐橙云的客户端接入框架，可以使开发者在自身iOS应用中快速集成乐橙云视频能力，本SDK只含真机调试的功能，不支持任何模拟器调试，支持Xcode 9.0以上。

本文档主要从以下几方面对如何集成 SDK进行说明：

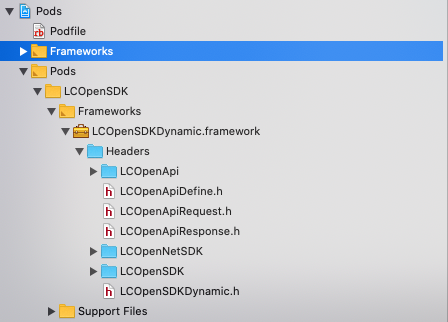
|  |  |
| --- | --- |
| SDK安装 | 介绍如何引入SDK，以及引入后的相关配置。 |
| 主要流程 | 介绍初始化SDK、直播、录像播放、对讲、录像下载、配网主要流程。 |
| [SDK接口说明](#_SDK接口说明) | 介绍LCOpenSDK各个接口的功能、参数、返回值以及注意事项。 |

# SDK安装

SDK的安装分为两种方式：cocoapods安装和官网下载(https://open.imou.com)。

## CocoaPods安装

通过CocoaPods引入比较简单，首先在Podfile文件中添加pod 'LCOpenSDK'，然后执行pod install，如果不能引入则可能是需要更新本地仓库，需执行pod repo update。安装成功后如下图：



其次，在项目中配置SDK路径：

1.Framework路径：Build Setting->Search Paths->Framework Search Paths

添加：

$(PROJECT\_DIR)/Pods/LCOpenSDK/Framework.

2.头文件路径：Build Setting->Search Paths->Header Search Paths

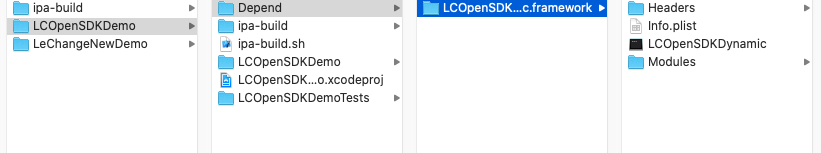
添加：

$(PROJECT\_DIR)/Pods/LCOpenSDK/Framework/LCOpenSDKDynamic.framework/Headers/LCOpenSDK $(PROJECT\_DIR)/Pods/LCOpenSDK/Framework/LCOpenSDKDynamic.framework/Headers/LCOpenNetSDK $(PROJECT\_DIR)/Pods/LCOpenSDK/Framework/LCOpenSDKDynamic.framework/Headers/LCOpenApi

3.Build Settings->Build Options ->Enable Bitcode 设置为NO

## 官网下载

官网下载是通过乐橙云开放平台官网，下载到本地进行手动引入，此时引入时要十分注意SDK的路径，以demo为例，SDK位于工程目录下的Depend文件夹下面。



SDK配置过程如下：

1.Framework路径：Build Setting->Search Paths->Framework Search Paths

添加：$(PROJECT\_DIR)/Depend

2.头文件路径：Build Setting->Search Paths->Framework Search Paths

添加：

$(PROJECT\_DIR)/Depend/LCOpenSDKDynamic.framework/Headers

$(PROJECT\_DIR)/Depend/LCOpenSDKDynamic.framework/Headers/LCOpenSDK

$(PROJECT\_DIR)/Depend/LCOpenSDKDynamic.framework/Headers/LCOpenNetSDK

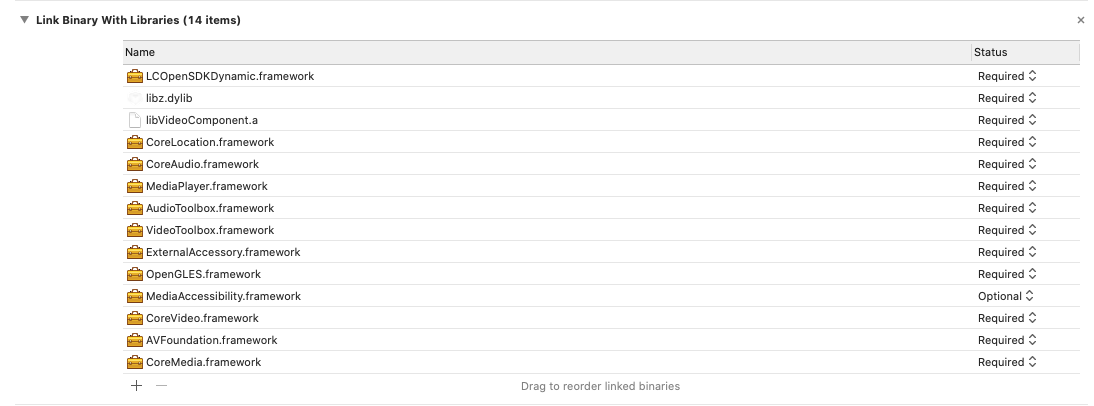
$(PROJECT\_DIR)/Depend/LCOpenSDKDynamic.framework/Headers/LCOpenApi

3.Build Phases ->Embed Frameworks

添加：LCOpenSDKDynamic.framework，Destinations选择Frameworks。

注：$(PROJECT\_DIR)为Demo工程所在目录，按要求添加LCOpenSDKDemo和Depend文件夹（对应iOS\_demo和iOS\_sdk文件夹）；Build Phases ->Embed Frameworks路径如果不存在选择左上角加号，然后选择NEW Copy Files Phase进行添加， 最后双击修改名称为Embed Frameworks。

4.配置依赖系统库：在工程TARGETS->Build Phases->Link Binary With Libraries下添加如下系统库：CoreAudio.framework、MediaPlayer.framework、AudioToolbox.framework、VideoToolbox.framework、OpenGLES.framework、MediaAccessibility.framework、CoreVideo.framework、AVFoundation.framework、 CoreMedia .framework、libz.tbd（libz.dylib），配置后如下图：



## 权限配置

SDK使用需配置相关系统权限，主要包含：

相册权限(NSPhotoLibraryUsageDescription)

照片存储权限(NSPhotoLibraryAddUsageDescription)

麦克风权限(NSMicrophoneUsageDescription)

相机权限(NSCameraUsageDescription)

本地网络权限(NSLocalNetworkUsageDescription)

地理位置权限(NSLocationUsageDescription)

蓝牙访问权限(NSBluetoothAlwaysUsageDescription)

蓝牙通信权限(NSBluetoothPeripheralUsageDescription)

注：配网另需组播权限，需从苹果开发者中心申请。

# 主要流程

SDK提供的主要功能包括：设备配网、实时预览、录像回放（包括设备录像回放和云录像回放）、语音对讲、录像下载（包括设备录像下载和云录像下载）。

## 初始化

### SDK初始化

在使用SDK相关功能前需要先进行初始化，初始化涉及开发者用户授权token和平台地址信息，初始化操作主要依赖于LCOpenSDK\_Api类，具体代码实现参考下面：

/\*\* 初始化平台信息 \*/

LCOpenSDK\_ApiParam \* apiParam = [[LCOpenSDK\_ApiParam alloc] init];

/\*\* 1. 设置协议 \*/

apiParam.procotol = PROCOTOL\_TYPE\_HTTPS;

/\*\* 2. 设置乐橙云平台服务地址 \*/

apiParam.addr = @"openapi.lechange.cn";

/\*\* 3. 设置服务端口 \*/

apiParam.port = 443;

/\*\* 4. 设置token，token根据AppId和AppSecret调用乐橙云开放平台获取 \*/

apiParam.token = @"\*\*\*\*\*\*\*\*";

/\*\* 5. 调用SDK接口进行初始化平台信息 \*/

LCOpenSDK\_Api\* m\_hc = [[LCOpenSDK\_Api shareMyInstance] initOpenApi:apiParam];

### 配置SDK日志级别

SDK为开发者提供LCOpenSDK\_Log日志类，在初始化SDK时默认日志等级为LogLevelTypeFatal，开发者可以根据需求设置日志等级，日志等级分为：LogLevelTypeFatal、LogLevelTypeError、LogLevelTypeWarning、LogLevelTypeInfo、LogLevelTypeDebug、logLevelTypeSecurityAll、LogLevelTypeALL。

// 初始化LCOpenSDK\_LogInfo实例对象

LCOpenSDK\_LogInfo \*info = [LCOpenSDK\_LogInfo new];

// 赋值日志等级字段

info.levelType = LogLevelTypeDebug;

// LCOpenSDK\_LogInfo实例配置日志等级

[[LCOpenSDK\_Log shareInstance] setLogInfo:info];

## 设备配网

设备配网是SDK基础功能之一，目前SDK提供设备配网方式分为：SmartConfig配网、软AP配网、蓝牙配网三种方式，具体采用哪种配网方式，需要根据设备型号进行确定。

### SmartConfig配网添加

1.获取设备信息(扫描标签或者手动输入)；

2.调用unBindDeviceInfo接口查询设备配网方式和绑定状态；

3.检测设备是否绑定；

4.判断是否支持smartConfig配网；

5.开始无线配网；

6.开始局域网内搜索设备初始化信息；

7.搜索到设备，停止搜索；

8.调用unBindDeviceInfo查询设备状态；

9.查询到设备在线，调用bindDevice接口绑定设备；

/\*\* 1.初始化LCOpenSDK\_ConfigWIfi对象 \*/

LCOpenSDK\_ConfigWIfi \*m\_configWifi = [[LCOpenSDK\_ConfigWIfi alloc] init];

/\*\* 2.开始配网

devId:设备序列号

ssid:wifi名称

pwd:wifi密码

security:安全码，SC码设备需要，非SC设备传@“”

voiceFreq:声波频率，需要根据设备类型区分。例如：普通11000，TP7C 17000

txMode:波形发送方式(0--新的fsk发送方式，1--老的fsk发送方式，2--新的和老的fsk波形发送方式)

\*/

NSString \* path = [m\_configWifi configWifiStart:@"\*\*\*\*\*\*\*\*"

ssid:@"\*\*\*\*\*\*\*\*"

password:@"\*\*\*\*\*\*\*\*"

secure:@"\*\*\*\*\*\*\*\*"

voiceFreq:11000

txMode:0];

if (path == nill || path.lenght == 0) {

NSLog(@"smartconfig failed\n");

}

### 软Ap配网

1.获取设备信息(扫描标签或者手动输入)；

2.调用unBindDeviceInfo接口查询设备信息；

3.检测设备是否绑定；

4.判断是否支持SoftAP配网；

5.连接设备热点；

6.开始搜索设备，如果搜索到了，直接绑定；

7.开始SoftAP配网；

8.配网成功，停止搜索；

9.调用unBindDeviceInfo查询设备状态；

10.查询到设备在线，调用bindDevice接口绑定设备；

注：SDK目前支持PaaS设备和IOT设备软Ap配网。

1. Paas设备软Ap配网

/\*\* 1. 初始化软Ap配网对象 \*/

LCOpenSDK\_SoftAPConfig \*m\_softAP = [[LCOpenSDK\_SoftAPConfig alloc] init];

/\*\* 2.开始配网

wifiName:wifi名字

wifiPwd:wifi密码

wifiEncry:wifi加密方式 0：off, 2：WEP64bit, 3：WEP128bit, 4:WPA-PSK-TKIP, 5: WPA-PSK-CCMP

netcardName:wifi网卡名字

deviceIp:设备IP

devicePwd:设备密码

isSC:设备是否使用安全码，设备已初始化时为false

handler:回调

timeout:超时时间

\*/

[m\_softAP startSoftAPConfig:@"\*\*\*\*\*" wifiPwd:@"\*\*\*\*\*" wifiEncry:0 netcardName:@"\*\*\*\*\*" deviceIp:@"\*\*\*\*\*" devicePwd:@"\*\*\*\*\*\*\*\*" isSC:True handler:^(NSInteger result) {

if (result < 0) {

NSLog(@"SoftAP failed\n");

} timeout:5000 \* 2];

1. IOT设备软AP配网

/\*\* 1. 初始化软Ap配网对象 \*/

LCOpenSDK\_IotApConfig \*m\_softAP = [[LCOpenSDK\_IotApConfig alloc] init];

/\*\* 2.开始配网

wifiName:wifi名字

wifiPwd:wifi密码

productId:产品ID

deviceId:设备序列号

finshed:回调

\*/

[m\_softAP startAsyncIotApConfig:@"\*\*\*\*\*" wifiPwd:@"\*\*\*\*\*" productId:@"\*\*\*\*\*" deviceId:@"\*\*\*\*\*" finshed:^(BOOL success, NSString \* \_Nullable errorMessage) {

if(success == false) {

NSLog(@"SoftAP failed\n 错误信息为:%@",errorMessage);

}

}];

### 蓝牙配网

1.获取设备信息(扫描标签或者手动输入)；

2.调用unBindDeviceInfo接口查询设备信息；

3.检测设备是否绑定；

4.判断是否支持蓝牙配网；

5.开始蓝牙配网；

6.配网成功；

7.调用unBindDeviceInfo查询设备状态；

8.查询到设备在线，调用bindDevice接口绑定设备；

/\*\* 1. 初始化蓝牙配网对象 \*/

LCOpenSDK\_Bluetooth \*m\_bluetooth = [[LCOpenSDK\_Bluetooth alloc] init];

/\*\* 2.开始配网

wifiName:wifi名字

wifiPwd:wifi密码

productId:产品ID

deviceId:设备序列号

finshed:回调

\*/

[m\_bluetooth startAsyncBLEConfig:@"\*\*\*\*\*" wifiPwd:@"\*\*\*\*\*" productId:@"\*\*\*\*\*" deviceId:@"\*\*\*\*\*" finshed:^(BOOL success, NSString \* \_Nullable errorMessage) {

if(success == false) {

NSLog(@"Bluetooth failed\n 错误信息为:%@",errorMessage);

}

}];

## 实时预览

实时预览是SDK的主要功能之一，开发者可以通过开放平台接口和SDK接口进行设备实时预览功能的开发，其主要步骤为：

第一步：通过乐橙云平台设备查询接口获取设备信息（https://open.imou.com/book/http/device/manage/query/listDeviceDetailsByIds.html）获取设备信息。

第二步：调用SDK接口对播放窗口初始化和监听，具体代码实现参考下面：

/\*\* 1. 初始化播放窗口信息，包括位置大小以及索引 \*/

LCOpenSDK\_PlayRealWindow \*m\_play = [[LCOpenSDK\_PlayRealWindow alloc] initPlayWindow:CGRectMake(0, 0, 400, 300) Index:0];

/\*\* 2. 设置播放窗口背景色 \*/

[m\_play setSurfaceBGColor:[UIColor blackColor]];

/\*\* 3. 添加播放窗口到父视图 \*/

[self.view addSubview:[m\_play getWindowView]];

/\*\* 4. 设置监听对象，可监听播放结果回调 \*/

[m\_play setPlayRealListener:(id<LCOpenSDK\_PlayRealListener>)listener];

第三步：调用SDK接口对播放参数设置，然后开始播放，具体代码实现参考下面：

/\*\* 1. 初始化实时预览播放参数对象 \*/

LCOpenSDK\_ParamReal \*paramReal = [[LCOpenSDK\_ParamReal alloc] init];

/\*\* 2. 设置token，token根据AppId和AppSecret调用乐橙云开放平台获取 \*/

paramReal.accessToken = @"\*\*\*\*\*\*\*\*";

/\*\* 3. 设置设备序列号 \*/

paramReal.deviceID = @"\*\*\*\*\*\*\*\*";

/\*\* 4. 设置产品ID ，非必传，只有IOT设备才有\*/

paramReal.productId = @"\*\*\*\*\*\*\*\*";

/\*\* 5. 设置设备通道，对于单通道设备为0， 对于NVR等多通道设备为具体通道号 \*/

paramReal.channel = 0;

/\*\* 6. 设置密钥，如果为用户自定义密钥为具体密钥， 非用户自定密钥可为设备序列号\*/

paramReal.psk = @"\*\*\*\*\*\*\*\*";

/\*\* 7. 设置播放密钥，非必传，传了会提升拉流速度 \*/

paramReal.playToken = @"\*\*\*\*\*\*\*\*";

/\*\* 8. 设置是否使用TLS加密 \*/

paramReal.useTLS = YES;

/\*\* 9. 设置是否展示播放window，默认展示 \*/

paramReal.showWindow = YES；

/\*\* 10. 设置码流模式，支持主码流和子码流 \*/

paramReal.defiMode = DEFINITION\_MODE\_HG;

/\*\* 11. 是否走拉流优化，YES会提升拉流速度，建议传YES即可 \*/

paramReal.isOpt = YES;

/\*\* 12. 设置码流分辨率，如设备支持720P、1080P、2K等分辨率 \*/

paramReal.imageSize = xxx;

/\*\* 13. 设置是否开启音频，默认开启 \*/

paramReal.isOpenAudio = YES;

/\*\* 14. 设置是否开启辅助帧，默认关闭 \*/

paramReal.isAssistFrame = No;

/\*\* 15. 开始播放 \*/

[m\_play playRtspReal:paramReal];

第四步：实现监听回调，具体代码实现参考下面：

/\*\* 1. 播放结果回调，可根据错误码进行不同处理 \*/

- (void)onPlayerResult:(NSString\*)code Type:(NSInteger)type Index:(NSInteger)index {

}

/\*\* 2. 播放开始回调 \*/

- (void)onPlayBegan:(NSInteger)index {

}

## 录像回放

录像回放是对设备存储在SD卡或硬盘上的录像和云存储上的录像进行播放的功能，针对设备录像和云存储录像，SDK提供了不同的接口，使用时要注意。

对于设备录像播放，SDK提供了两个接口，分别为按文件名播放和按时间播放，开发者可根据不同场景需要进行使用，推荐使用按文件名播放。

开发录像回放功能的主要步骤有：

第一步：通过乐橙云平台设备查询接口（https://open.imou.com/book/http/device/manage/query/listDeviceDetailsByIds.html）获取设备信息。

第二步：通过乐橙云平台云录像查询接口（https://open.imou.com/book/http/device/record/cloud/queryCloudRecordNum.html）获取云存储录像信息 ，通过本地录像查询接口（https://open.imou.com/book/http/device/record/local/summary.html）获取设备本地录像信息。

第三步：调用SDK接口对播放窗口初始化和监听，具体代码实现参考下面：

/\*\* 1. 初始化播放窗口信息，包括位置大小以及索引 \*/

LCOpenSDK\_PlayRealWindow \*m\_play = [[LCOpenSDK\_PlayRealWindow alloc] initPlayWindow:CGRectMake(0, 0, 400, 300) Index:0];

/\*\* 2. 设置播放窗口背景色 \*/

[m\_play setSurfaceBGColor:[UIColor blackColor]];

/\*\* 3. 添加播放窗口到父视图 \*/

[self.view addSubview:[m\_play getWindowView]];

/\*\* 4. 设置监听对象，可监听播放结果回调 \*/

[m\_play setPlayRealListener:(id<LCOpenSDK\_PlayRealListener>)listener];

第四步：调用SDK接口对播放参数设置，然后开始播放，具体代码实现参考下面：

1. 设备录像按文件名播放：

/\*\* 1. 初始化按文件名播放参数对象 \*/

LCOpenSDK\_ParamDeviceRecordFileName \* paramDeviceRecord = [[LCOpenSDK\_ParamDeviceRecordFileName alloc] init];

/\*\* 2. 设置token，token根据AppId和AppSecret调用乐橙云开放平台获取 \*/

paramDeviceRecord.accessToken = @"\*\*\*\*\*\*\*\*";

/\*\* 3. 设置设备序列号 \*/

paramDeviceRecord.deviceID = @"\*\*\*\*\*\*\*\*";

/\*\* 4. 设置产品ID ，非必传，只有IOT设备才有\*/

paramDeviceRecord.productId = @"\*\*\*\*\*\*\*\*";

/\*\* 5. 设置设备通道，对于单通道设备为0， 对于NVR等多通道设备为具体通道号 \*/

paramDeviceRecord.channel = 0;

/\*\* 6. 设置密钥，如果为用户自定义密钥为具体密钥， 非用户自定密钥可为设备序列号\*/

paramDeviceRecord.psk = @"\*\*\*\*\*\*\*\*";

/\*\* 7. 设置播放密钥，非必传，传了会提升拉流速度 \*/

paramDeviceRecord.playToken = @"\*\*\*\*\*\*\*\*";

/\*\* 8. 设置是否使用TLS加密 \*/

paramDeviceRecord.useTLS = YES;

/\*\* 9. 设置是否展示播放window，默认展示 \*/

paramDeviceRecord.showWindow = YES；

/\*\* 10. 设置是否开启音频，默认开启 \*/

paramDeviceRecord.isOpenAudio = YES;

/\*\* 11. 设置播放倍速，默认是1倍速 \*/

paramDeviceRecord.speed = 1;

/\*\* 12. 设置录像文件名即录像Id（recordId）\*/

paramDeviceRecord.fileName = @"\*\*\*\*\*\*\*\*";

/\*\* 13. 设置播放偏移时间，单位为秒，从文件开始播放设置为0即可\*/

paramDeviceRecord.offsetTime = 0 ;

/\*\* 14. 是否走拉流优化，YES会提升拉流速度，建议传YES即可\*/

paramDeviceRecord.isOpt = YES;

/\*\* 15. 开始播放 \*/

[m\_play playRecordStream:paramDeviceRecord];

1. 设备录像按时间播放

/\*\* 1. 初始化按时间播放参数对象\*/

LCOpenSDK\_ParamDeviceRecordUTCTime \*paramDeviceRecordUTCTime = [[LCOpenSDK\_ParamDeviceRecordUTCTime alloc] init];

/\*\* 2. 设置token，token根据AppId和AppSecret调用乐橙云开放平台获取 \*/

paramDeviceRecordUTCTime.accessToken = @"\*\*\*\*\*\*\*\*";

/\*\* 3. 设置设备序列号 \*/

paramDeviceRecordUTCTime.deviceID = @"\*\*\*\*\*\*\*\*";

/\*\* 4. 设置产品ID ，非必传，只有IOT设备才有\*/

paramDeviceRecordUTCTime.productId = @"\*\*\*\*\*\*\*\*";

/\*\* 5. 设置设备通道，对于单通道设备为0， 对于NVR等多通道设备为具体通道号 \*/

paramDeviceRecordUTCTime.channel = 0;

/\*\* 6. 设置密钥，如果为用户自定义密钥为具体密钥， 非用户自定密钥可为设备序列号\*/

paramDeviceRecordUTCTime.psk = @"\*\*\*\*\*\*\*\*";

/\*\* 7. 设置播放密钥，非必传，传了会提升拉流速度 \*/

paramDeviceRecordUTCTime.playToken = @"\*\*\*\*\*\*\*\*";

/\*\* 8. 设置是否使用TLS加密 \*/

paramDeviceRecordUTCTime.useTLS = YES;

/\*\* 9. 设置是否展示播放window，默认展示 \*/

paramDeviceRecordUTCTime.showWindow = YES；

/\*\* 10. 设置是否开启音频，默认开启 \*/

paramDeviceRecordUTCTime.isOpenAudio = YES;

/\*\* 11. 设置播放倍速，默认是1倍速 \*/

paramDeviceRecordUTCTime.speed = 1;

/\*\* 12. 录像的开始时间，不为0 \*/

paramDeviceRecordUTCTime.beginTime = 0;

/\*\* 13. 录像的结束时间，不为0 \*/

paramDeviceRecordUTCTime.endTime = 0;

/\*\* 14. 设置播放偏移时间，单位为秒，从文件开始播放设置为0即可\*/

paramDeviceRecordUTCTime.offsetTime = 0 ;

/\*\* 15. 是否走拉流优化，YES会提升拉流速度，建议传YES即可\*/

paramDeviceRecordUTCTime.isOpt = YES;

/\*\* 16. 设置码流模式，支持主码流和子码流 \*/

paramDeviceRecordUTCTime.defiMode = DEFINITION\_MODE\_HG;

/\*\* 17. 开始播放 \*/

[m\_play playRecordStream:paramDeviceRecordUTCTime];

1. 云存储录像播放

/\*\* 1. 初始化云存储录像播放参数对象 \*/

LCOpenSDK\_ParamCloudRecord \*paramCloudRecord = [[LCOpenSDK\_ParamCloudRecord alloc] init];

/\*\* 2. 设置token，token根据AppId和AppSecret调用乐橙云开放平台获取 \*/

paramCloudRecord.accessToken = @"\*\*\*\*\*\*\*\*";

/\*\* 3. 设置设备序列号 \*/

paramCloudRecord.deviceID = @"\*\*\*\*\*\*\*\*";

/\*\* 4. 设置产品ID ，非必传，只有IOT设备才有\*/

paramCloudRecord.productId = @"\*\*\*\*\*\*\*\*";

/\*\* 5. 设置设备通道，对于单通道设备为0， 对于NVR等多通道设备为具体通道号 \*/

paramCloudRecord.channel = 0;

/\*\* 6. 设置密钥，如果为用户自定义密钥为具体密钥， 非用户自定密钥可为设备序列号\*/

paramCloudRecord.psk = @"\*\*\*\*\*\*\*\*";

/\*\* 7. 设置播放密钥，非必传，传了会提升拉流速度 \*/

paramCloudRecord.playToken = @"\*\*\*\*\*\*\*\*";

/\*\* 8. 设置是否使用TLS加密 \*/

paramCloudRecord.useTLS = YES;

/\*\* 9. 设置是否展示播放window，默认展示 \*/

paramCloudRecord.showWindow = YES；

/\*\* 10. 设置是否开启音频，默认开启 \*/

paramCloudRecord.isOpenAudio = YES;

/\*\* 11. 设置播放倍速，默认是1倍速 \*/

paramCloudRecord.speed = 1;

/\*\* 12. 设置录像Id（recordRegionID）\*/

paramCloudRecord.recordRegionID = @"\*\*\*\*\*\*\*\*";

/\*\* 13. 设置播放偏移时间，单位为秒，从文件开始播放设置为0即可 \*/

paramCloudRecord.offsetTime = 0;

/\*\* 14. 设置录像类型，设置为RECORD\_TYPE\_ALL即可 \*/

paramCloudRecord.recordType = RECORD\_TYPE\_ALL;

/\*\* 15. 设置播放超时时间， 单位为秒\*/

paramCloudRecord.timeOut = 60;

/\*\* 16. 开始播放 \*/

[m\_play playRecordStream:paramCloudRecord]

第四步：实现监听回调，具体代码实现参考下面：

/\*\* 1. 播放结果回调，可根据错误码进行不同处理 \*/

- (void)onPlayerResult:(NSString\*)code Type:(NSInteger)type Index:(NSInteger)index {

}

/\*\* 2. 播放开始回调 \*/

- (void)onPlayBegan:(NSInteger)index {

}

/\*\* 3. 播放完成回调 \*/

- (void)onPlayFinished:(NSInteger)index {

}

/\*\* 4. 文件开始时间和结束时间回调 \*/

- (void)onFileTime:(long)beginTime EndTime:(long)endTime Index:(NSInteger)index {

}

/\*\* 5. 文件播放时间回调 \*/

- (void)onPlayerTime:(long)time Index:(NSInteger)index {

}

## 语音对讲

语音对讲是在设备进行实时预览时，设备和客户端进行对讲的功能。语音对讲功能开发主要步骤为：

第一步：进行实时预览功能开发。

第二步：调用SDK接口初始化语音对讲对象，初始化播放参数对象并设置参数开始对讲，具体代码实现参考下面：

/\*\* 1. 初始化语音对讲对象 \*/

LCOpenSDK\_AudioTalk \*m\_talker = [[LCOpenSDK\_AudioTalk alloc] init];

/\*\* 2. 设置监听对象 \*/

[m\_talker setListener:(id<LCOpenSDK\_TalkerListener>)self];

/\*\* 3. 初始化语音对讲参数对象 \*/

LCOpenSDK\_ParamTalk \*paramTalk = [[LCOpenSDK\_ParamTalk alloc] init];

/\*\* 4. 设置token，token根据AppId和AppSecret调用乐橙云开放平台获取 \*/

paramTalk.accessToken = @"\*\*\*\*\*\*\*\*";

/\*\* 5. 设置设备序列号 \*/

paramTalk.deviceID = @"\*\*\*\*\*\*\*\*";

/\*\* 6. 设置产品ID ，非必传，只有IOT设备才有\*/

paramTalk.productId = @"\*\*\*\*\*\*\*\*";

/\*\* 7. 设置设备通道，对于单通道设备为0， 对于NVR等多通道设备为具体通道号 \*/

paramTalk.channel = 0;

/\*\* 8. 设置密钥，如果为用户自定义密钥为具体密钥， 非用户自定密钥可为设备序列号\*/

paramTalk.psk = @"\*\*\*\*\*\*\*\*";

/\*\* 9. 设置播放密钥，非必传，传了会提升拉流速度 \*/

paramTalk.playToken = @"\*\*\*\*\*\*\*\*";

/\*\* 10. 设置是否使用TLS加密 \*/

paramTalk.useTLS = YES;

/\*\* 11. 开始对讲 \*/

[m\_talker playTalk:paramTalk];

第三步：实现对讲结果回调，具体代码实现参考下面：

/\*\* 对讲结果回调 \*/

-(void) onTalkResult:(NSString \*) error TYPE:(NSInteger) type {

}

## 可视对讲（仅针对可视对讲设备）

可视对讲是在设备进行实时预览时，设备和客户端进行可视对讲的功能。可视对讲功能开发主要步骤为：

第一步：调用SDK接口对播放窗口初始化和监听，具体代码实现参考下面：

/\*\* 1. 初始化播放窗口信息，包括位置大小以及索引 \*/

LCOpenSDK\_PlayRealWindow \*m\_play = [[LCOpenSDK\_PlayRealWindow alloc] initPlayWindow:CGRectMake(0, 0, 400, 300) Index:0];

/\*\* 2. 设置播放窗口背景色 \*/

[m\_play setSurfaceBGColor:[UIColor blackColor]];

/\*\* 3. 添加播放窗口到父视图 \*/

[self.view addSubview:[m\_play getWindowView]];

/\*\* 4. 设置监听对象，可监听播放结果回调 \*/

[m\_play setPlayRealListener:(id<LCOpenSDK\_PlayRealListener>)listener];

第三步：调用SDK接口对播放参数设置，然后开始播放，具体代码实现参考下面：

/\*\* 1. 初始化实时预览播放参数对象 \*/

LCOpenSDK\_ParamReal \*paramReal = [[LCOpenSDK\_ParamReal alloc] init];

/\*\* 2. 设置token，token根据AppId和AppSecret调用乐橙云开放平台获取 \*/

paramReal.accessToken = @"\*\*\*\*\*\*\*\*";

/\*\* 3. 设置设备序列号 \*/

paramReal.deviceID = @"\*\*\*\*\*\*\*\*";

/\*\* 4. 设置产品ID ，非必传，只有IOT设备才有\*/

paramReal.productId = @"\*\*\*\*\*\*\*\*";

/\*\* 5. 设置设备通道，对于单通道设备为0， 对于NVR等多通道设备为具体通道号 \*/

paramReal.channel = 0;

/\*\* 6. 设置密钥，如果为用户自定义密钥为具体密钥， 非用户自定密钥可为设备序列号\*/

paramReal.psk = @"\*\*\*\*\*\*\*\*";

/\*\* 7. 设置播放密钥，非必传，传了会提升拉流速度 \*/

paramReal.playToken = @"\*\*\*\*\*\*\*\*";

/\*\* 8. 设置是否使用TLS加密 \*/

paramReal.useTLS = YES;

/\*\* 9. 设置是否展示播放window，默认展示 \*/

paramReal.showWindow = YES；

/\*\* 10. 设置码流模式，支持主码流和子码流 \*/

paramReal.defiMode = DEFINITION\_MODE\_HG;

/\*\* 11. 是否走拉流优化，YES会提升拉流速度，建议传YES即可 \*/

paramReal.isOpt = YES;

/\*\* 12. 设置码流分辨率，如设备支持720P、1080P、2K等分辨率 \*/

paramReal.imageSize = xxx;

/\*\* 13. 设置是否开启音频，默认开启 \*/

paramReal.isOpenAudio = YES;

/\*\* 14. 设置是否开启辅助帧，默认关闭 \*/

paramReal.isAssistFrame = No;

/\*\* 14. 设置是否支持可视对讲，默认关闭 \*/

paramReal.isSupportVideoTalk= YES;

/\*\* 15. 开始播放 \*/

[m\_play playRtspReal:paramReal];

第四步：实现监听回调，具体代码实现参考下面：

/\*\* 1. 播放结果回调，可根据错误码进行不同处理 \*/

- (void)onPlayerResult:(NSString\*)code Type:(NSInteger)type Index:(NSInteger)index {

}

/\*\* 2. 播放开始回调 \*/

- (void)onPlayBegan:(NSInteger)index {

}

第二步：调用SDK接口初始化可视对讲对象，初始化播放参数对象并设置参数开始可视对讲，具体代码实现参考下面：

/\*\* 1. 初始化可视对讲对象 \*/

LCOpenSDK\_AudioTalk \*m\_talker = [[LCOpenSDK\_AudioTalk alloc] init];

/\*\* 2. 设置监听对象 \*/

[m\_talker setListener:(id<LCOpenSDK\_TalkerListener>)self];

/\*\* 3. 初始化语音对讲参数对象 \*/

LCOpenSDK\_ParamTalk \*paramTalk = [[LCOpenSDK\_ParamTalk alloc] init];

/\*\* 4. 设置token，token根据AppId和AppSecret调用乐橙云开放平台获取 \*/

paramTalk.accessToken = @"\*\*\*\*\*\*\*\*";

/\*\* 5. 设置设备序列号 \*/

paramTalk.deviceID = @"\*\*\*\*\*\*\*\*";

/\*\* 6. 设置产品ID ，非必传，只有IOT设备才有\*/

paramTalk.productId = @"\*\*\*\*\*\*\*\*";

/\*\* 7. 设置设备通道，对于单通道设备为0， 对于NVR等多通道设备为具体通道号 \*/

paramTalk.channel = 0;

/\*\* 8. 设置密钥，如果为用户自定义密钥为具体密钥， 非用户自定密钥可为设备序列号\*/

paramTalk.psk = @"\*\*\*\*\*\*\*\*";

/\*\* 9. 设置播放密钥，非必传，传了会提升拉流速度 \*/

paramTalk.playToken = @"\*\*\*\*\*\*\*\*";

/\*\* 10. 设置是否使用TLS加密 \*/

paramTalk.useTLS = YES;

/\*\* 11. 设置可视对讲类型，talk为客户端主动发起呼叫，call为设备端发起呼叫\*/

paramTalk.talkType = @"talk";

/\*\* 12. 创建可视对讲参数，设置采样宽度，高度，I帧间隔，编码类型，采样率，摄像头是否开启以及是否需要软编码 \*/

LCOpenSDK\_MediaVideoSampleConfigParam \*videoSampleCfg= [LCOpenSDK\_MediaVideoSampleConfigParam initWithParam:240 height:320 I\_frame\_interval:15 encodeType:2 frameRate:15 isCameraOpen:YES softEncodeMode:YES];

/\*\* 13. 开始对讲 \*/

[m\_talker playVisualTalk :paramTalk videoSampleCfg: videoSampleCfg];

第三步：开启手机摄像头（参考Demo中的摄像头管理类LCVisualTalkCaptureSession）：

lazy var captureSession: LCVisualTalkCaptureSession = {

let captureSession = LCVisualTalkCaptureSession(captureWith: .preset640x480)

captureSession.delegate = self

return captureSession

}()

captureSession.start()

第四步：实现摄像头结果回调，在回调中对采集到的摄像头数据进行软编码，传输具体代码实现参考下面：

/\*\* 摄像头采集结果回调 \*/

- (void)videoWithSampleBuffer:(CMSampleBufferRef)sampleBuffer;

第五步：音视频数据硬编码，（参考Demo中的视频编码管理类LCH264Encoder）具体代码实现参考下面：

lazy var h264Encoder: LCH264Encoder = {

let h264Encoder = LCH264Encoder()

h264Encoder.delegate = self

return h264Encoder

}()

h264Encoder.createEncodeSession(240, height: 320, fps: 15, bite: 240 \* 320 \* 12 \* 4)

/\*\* 视频编码结果回调 \*/

-(void)h264Encoder:(LCH264Encoder \*)encoder encodeData:(NSData \*)data;

第六步：实现视频编码回调，具体代码实现参考下面：

/\*\* 对讲结果回调 \*/

-(void)h264Encoder:(LCH264Encoder \*)encoder encodeData:(NSData \*)data;

第七步：下发音视频数据，在对讲回调中，开始下发视频数据（需要先保证对讲连接成功，监听到对讲回调错误码为STATE\_RTSP\_PLAY\_READY时再开始下发视频数据），具体代码实现参考下面：

/\*\* 下发视频数据 \*/

-(BOOL)pushMediaData:(int)type mediaData:(NSData \*)data datalen:(int)datalen needSoftEncode:(BOOL)softEncode

第八步：停止视频采样，停止音频采样，具体代码实现参考下面：

/\*\* 停止视频采样 \*/

m\_talker. enableTalkVideo（false）

/\*\* 停止音频采样 \*/

m\_talker. stopSampleAudio（）

## 录像下载

录像下载是对设备存储在SD卡或硬盘上的录像和云存储上的录像进行下载，针对设备录像和云存储录像，SDK提供了不同的接口，使用时要注意。

对于设备录像下载，SDK提供了两个接口，分别为按文件名下载和按时间下载，开发者可根据不同场景需要进行使用。目前接口仅支持MP4格式的录像文件下载，暂不支持其他格式。录像下载功能开发主要步骤为：

第一步：通过乐橙云平台接口（https://open.imou.com/book/http/device/manage/query/listDeviceDetailsByIds.html）获取设备信息。

第二步：通过乐橙云平台云录像查询接口(https://open.imou.com/book/http/device/record/cloud/queryCloudRecordNum.html)获取云存储录像信息 、通过本地录像查询接口(https://open.imou.com/book/http/device/record/local/summary.html)获取设备本地录像信息。

第三步：调用SDK接口初始化下载对象，设置监听对象，具体代码实现参考下面：

/\*\* 1.路径名称 \*/

NSArray\* paths = NSSearchPathForDirectoriesInDomains(NSLibraryDirectory, NSUserDomainMask, YES);

NSString\* libraryDirectory = [paths objectAtIndex:0];

NSString\* myDirectory = [libraryDirectory stringByAppendingPathComponent:@"lechange"];

NSString\* downloadDirectory = [myDirectory stringByAppendingPathComponent:@"download"];

/\*\* 不要每个都一样，否则会覆盖 \*/

NSString\* infoPath = [downloadDirectory stringByAppendingPathComponent:[NSString stringWithFormat:@"record\_download"]];

NSString\* m\_downloadPath = [infoPath stringByAppendingString:@".mp4"];

/\*\* 2.根据路径名称创建路径 \*/

NSFileManager\* fileManage = [NSFileManager defaultManager];

NSError\* pErr;

BOOL isDir;

if (NO == [fileManage fileExistsAtPath:myDirectory isDirectory:&isDir]) {

[fileManage createDirectoryAtPath:myDirectory withIntermediateDirectories:YES attributes:nil error:&pErr];

}

if (NO == [fileManage fileExistsAtPath:downloadDirectory isDirectory:&isDir]) {

[fileManage createDirectoryAtPath:downloadDirectory withIntermediateDirectories:YES attributes:nil error:&pErr];

}

/\*\* 3.创建下载对象 \*/

LCOpenSDK\_Download\* m\_download = [LCOpenSDK\_Download shareMyInstance];

/\*\* 4.设置监听 \*/

[m\_download setListener:(id<LCOpenSDK\_DownloadListener>)self];

第四步：调用SDK接口设置录像下载参数，然后开始下载，具体代码实现参考下面：

1. 云录像下载：

/\*\* 1. 初始化云录像下载参数对象 \*/

LCOpenSDK\_DownloadByRecordIdParam \* paramRecord = [[LCOpenSDK\_DownloadByRecordIdParam alloc] init];

/\*\* 2. 设置下载序号 \*/

paramRecord.index = 0;

/\*\* 3. 设置token，token根据AppId和AppSecret调用乐橙云开放平台获取 \*/

paramRecord.accessToken = @"\*\*\*\*\*\*\*\*";

/\*\* 4. 设置设备序列号 \*/

paramRecord.deviceID = @"\*\*\*\*\*\*\*\*";

/\*\* 5. 设置密钥，如果为用户自定义密钥为具体密钥， 非用户自定密钥可为设备序列号\*/

paramRecord.psk = @"\*\*\*\*\*\*\*\*";

/\*\* 6. 设置产品ID ，非必传，只有IOT设备才有\*/

paramRecord.productId = @"\*\*\*\*\*\*\*\*";

/\*\* 7. 设置设备通道，对于单通道设备为0， 对于NVR等多通道设备为具体通道号 \*/

paramRecord.channeld = 0;

/\*\* 8. 设置播放密钥，非必传，传了会提升拉流速度 \*/

paramRecord.playToken = @"\*\*\*\*\*\*\*\*";

/\*\* 9. 设置是否使用TLS加密 \*/

paramRecord.useTLS = YES;

/\*\* 10. 设置下载速度，下载速度：1/2/4/8/16/32，1表示按正常视频播放速度下载 \*/

paramRecord.speed = 1;

/\*\* 11. 设置录像Id \*/

paramRecord.recordRegionId = @"\*\*\*\*\*\*\*\*";

/\*\* 12. 设置云录像类型 1000：报警，2000：定时\*/

paramRecord.cloudType = 1000;

/\*\* 13. 开始下载 \*/

[m\_download startDownloadCloudRecord:paramRecord];

1. 设备录像按文件名下载

/\*\* 1. 初始化按文件名下载参数对象\*/

LCOpenSDK\_DownloadByRecordIdParam \* paramRecord = [[LCOpenSDK\_DownloadByRecordIdParam alloc] init];

/\*\* 2. 设置下载序号 \*/

paramRecord.index = 0;

/\*\* 3. 设置token，token根据AppId和AppSecret调用乐橙云开放平台获取 \*/

paramRecord.accessToken = @"\*\*\*\*\*\*\*\*";

/\*\* 4. 设置设备序列号 \*/

paramRecord.deviceID = @"\*\*\*\*\*\*\*\*";

/\*\* 5. 设置密钥，如果为用户自定义密钥为具体密钥， 非用户自定密钥可为设备序列号\*/

paramRecord.psk = @"\*\*\*\*\*\*\*\*";

/\*\* 6. 设置产品ID ，非必传，只有IOT设备才有\*/

paramRecord.productId = @"\*\*\*\*\*\*\*\*";

/\*\* 7. 设置设备通道，对于单通道设备为0， 对于NVR等多通道设备为具体通道号 \*/

paramRecord.channeld = 0;

/\*\* 8. 设置播放密钥，非必传，传了会提升拉流速度 \*/

paramRecord.playToken = @"\*\*\*\*\*\*\*\*";

/\*\* 9. 设置是否使用TLS加密 \*/

paramRecord.useTLS = YES;

/\*\* 10. 设置下载速度，下载速度：1/2/4/8/16/32，1表示按正常视频播放速度下载 \*/

paramRecord.speed = 1;

/\*\* 11. 设置设备录像名\*/

paramRecord.fileId = @"\*\*\*\*\*\*\*\*";

/\*\* 12. 开始下载 \*/

[m\_download startDownloadDeviceRecordById:paramRecord];

1. 设备录像按时间下载

/\*\* 1. 初始化按时间时间下载参数对象 \*/

LCOpenSDK\_DownloadByRecordIdParam \* paramRecord = [[LCOpenSDK\_DownloadByUTCTimeParam alloc] init];

/\*\* 2. 设置下载序号 \*/

paramRecord.index = 0;

/\*\* 3. 设置token，token根据AppId和AppSecret调用乐橙云开放平台获取 \*/

paramRecord.accessToken = @"\*\*\*\*\*\*\*\*";

/\*\* 4. 设置设备序列号 \*/

paramRecord.deviceID = @"\*\*\*\*\*\*\*\*";

/\*\* 5. 设置密钥，如果为用户自定义密钥为具体密钥， 非用户自定密钥可为设备序列号\*/

paramRecord.psk = @"\*\*\*\*\*\*\*\*";

/\*\* 6. 设置产品ID ，非必传，只有IOT设备才有\*/

paramRecord.productId = @"\*\*\*\*\*\*\*\*";

/\*\* 7. 设置设备通道，对于单通道设备为0， 对于NVR等多通道设备为具体通道号 \*/

paramRecord.channeld = 0;

/\*\* 8. 设置播放密钥，非必传，传了会提升拉流速度 \*/

paramRecord.playToken = @"\*\*\*\*\*\*\*\*";

/\*\* 9. 设置是否使用TLS加密 \*/

paramRecord.useTLS = YES;

/\*\* 10. 设置下载速度，下载速度：1/2/4/8/16/32，1表示按正常视频播放速度下载 \*/

paramRecord.speed = 1;

/\*\* 11. 设置开始时间，不为0\*/

paramRecord.beginTime = 0;

/\*\* 12. 设置结束时间，不为0\*/

paramRecord.endTime = 0;

/\*\* 13. 开始下载 \*/

[m\_download startDownloadDeviceRecordByUtcTime:paramRecord];

第四步：实现监听回调，具体代码实现参考下面：

/\*\* 1.下载数据回调 \*/

- (void) onDownloadReceiveData:(NSInteger)index datalen:(NSInteger)datalen {

}

/\*\* 2.下载状态回调 \*/

- (void) onDownloadState:(NSInteger)index code:(NSString \*)code type:(NSInteger)type {

}

## 数据解密

目前设备告警缩略图和录像封面图默认是加密存储的，需要解密后才能正常打开展示。使用SDK的LCOpenSDK\_Utils类实现图片的解密，具体代码实现参考下面：

1. 非TCM设备解密：

NSData\* dataOut = [[NSData alloc] init];

NSInteger iret = [[LCOpenSDK\_Utils new] decryptPic:picData deviceID:@"\*\*\*\*\*\*\*\*" key:@"\*\*\*\*\*\*\*\*" bufOut:&dataOut];

if (0 == iret) {

UIImage\* img = [UIImage imageWithData:[NSData dataWithBytes:[dataOut bytes] length:[dataOut length]]];

[cache storeImage:img forKey:key\_temp toDisk:YES completion:^{

}];

[self setImage:img];

}

1. TCM设备解密：

NSData\* dataOut = [[NSData alloc] init];

NSInteger iret = [[LCOpenSDK\_Utils new] decryptPic:picData deviceID:@"\*\*\*\*\*\*\*\*" productId:@"\*\*\*\*\*\*\*\*" key:@"\*\*\*\*\*\*\*\*" token:@"\*\*\*\*\*\*\*\*" bufOut:&dataOut];

或

NSInteger iret = [[LCOpenSDK\_Utils new] decryptPic:picData deviceID:@"\*\*\*\*\*\*\*\*" productId:@"\*\*\*\*\*\*\*\*" key:@"\*\*\*\*\*\*\*\*" playtoken:@"\*\*\*\*\*\*\*\*" bufOut:&dataOut];

if (0 == iret) {

UIImage\* img = [UIImage imageWithData:[NSData dataWithBytes:[dataOut bytes] length:[dataOut length]]];

[cache storeImage:img forKey:key\_temp toDisk:YES completion:^{

}];

[self setImage:img];

}

# SDK接口说明

## 概述

开放平台SDK以动态Framework的形式发布，提供多个方法类：LCOpenSDK\_Api（平台接口类）、LCOpenSDK\_AudioTalk（语音对讲类）、LCOpenSDK\_Bluetooth（设备蓝牙配网类）、LCOpenSDK\_ConfigWifi（设备无线配网类）、LCOpenSDK\_DeviceInit （设备初始化类）、LCOpenSDK\_Download（录像下载类）、LCOpenSDK\_iotApConfig（IOT设备软AP配网类）、LCOpenSDK\_Log（日志接口类）、LCOpenSDK\_ LoginManager（P2P管理类）、LCOpenSDK\_PlayBackWindow（录像播放类）、LCOpenSDK\_PlayFileWindow（文件播放类）、LCOpenSDK\_PlayRealWindow（实时预览播放类）、LCOpenSDK\_SearchDevices（搜索设备类）、LCOpenSDK\_SoftApConfig（设备软AP配网类）、LCOpenSDK\_Utils（公用方法类），iOS应用只需集成上述方法类，即可快速实现乐橙开放平台所支持的绝大部分业务。

此外，提供LCOpenSDK\_ PlayRealListener、LCOpenSDK\_ PlayBackListener、LCOpenSDK\_DownloadListener、LCOpenSDK\_TalkerListener、LCOpenSDK\_ PlayFileListener等监听接口类，分别配合LCOpenSDK\_PlayRealWindow、LCOpenSDK\_PlayBackWindow、LCOpenSDK\_Download、LCOpenSDK\_AudioTalk、LCOpenSDK\_PlayFileWindow使用，iOS应用可继承并设置相应的监听接口类来接收播放、对讲和下载过程中的业务回调，业务回调包括但不限于：播放窗口事件、视频播放过程中的状态码、语音对讲过程中的状态码、云录像下载过程中的状态码、解码时间戳，回调接口请参考后续说明。

注：标注为选填的为非必填项，否则皆为必填。

## LCOpenSDK\_Api

### shareMyInstance

|  |  |
| --- | --- |
| 方法 | + (LCOpenSDK\_Api\*\_Nonnull) shareMyInstance; |
| 简介 | 初始化单例 |
| 参数 | 无 |
| 返回值 | 单例对象 |
| 备注 | 无 |

### ­initOpenApi

|  |  |
| --- | --- |
| 方法 | - (id \_Nonnull)initOpenApi:(LCOpenSDK\_ApiParam \*\_Nonnull)apiParam; |
| 简介 | 接口初始化 |
| 参数 | apiParam:LCOpenSDK\_ApiParam参数模型  其中，LCOpenSDK\_ApiParam属性：  procotol:协议类型（https）  addr:平台地址  port:端口（443）  token:授权token（需开发者从平台自行获取） |
| 返回值 | LCOpenSDK\_Api指针 |
| 备注 | 在实现其他功能类接口之前必须先进行初始化。  国内addr:openapi.lechange.cn,海外:addr:openapi.easy4ip.com, port:443 |

### uninitOpenApi

|  |  |
| --- | --- |
| 方法 | - (void)uninitOpenApi; |
| 简介 | 反初始化接口，释放内部资源 |
| 参数 | 无 |
| 返回值 | 无 |
| 备注 | 无 |

### sdkVersion

|  |  |
| --- | --- |
| 方法 | - ( NSString \* \_Nonnull)sdkVersion; |
| 简介 | 获取SDK版本号信息 |
| 参数 | 无 |
| 返回值 | SDK版本号 |
| 备注 | 无 |

## LCOpenSDK\_Bluetooth

### startAsyncBLEConfig

|  |  |
| --- | --- |
| 方法 | + (void)startAsyncBLEConfig:(NSString \*)wifiName wifiPwd:(NSString \*)wifiPwd productId:(NSString \*)productId deviceId:(NSString \*)deviceId finshed:(void(^\_Nullable)(BOOL success, NSString \*\_Nullable errorMessage))finshed; |
| 简介 | IOT设备异步蓝牙配网 |
| 参数 | wifiName:wifi名字  wifiPwd:wifi密码  productid:IOT设备产品ID  deviceId:设备序列号（S/N）  finshed:设备蓝牙配网完成后的回调，sussess是否成功，errorMessage失败错误提示 |
| 返回值 | 无 |
| 备注 | 无 |

## LCOpenSDK\_ConfigWifi

### configWifiStart

|  |  |
| --- | --- |
| 方法 | -(NSString \*)configWifiStart:(NSString\*)devId ssid:(NSString\*)ssid password:(NSString\*)pwd secure:(NSString\*)security voiceFreq:(NSInteger)voiceFreq txMode:(NSInteger)txMode; |
| 简介 | 开始smartConfig配网 |
| 参数 | devId:设备序列号（S/N）  ssid:wifi名称  pwd:wifi密码  security[非必传]:设备安全码，SC码设备需要，默认是空字符串  voiceFreq:声波频率  txMode:波形发送方式，0--新的fsk发送方式、1--老的fsk发送方式、2--新的和老的fsk波形发送方式 |
| 返回值 | 0:接口调用成功，-1：接口调用失败 |
| 备注 | 无 |

### configWifiStop

|  |  |
| --- | --- |
| 方法 | -(NSInteger)configWifiStop; |
| 简介 | 停止无线配网 |
| 参数 | 无 |
| 返回值 | 0:接口调用成功，-1：接口调用失败 |
| 备注 | 收到配网事件回调后，不论成功或失败，都需调用此接口停止无线配网 |

## LCOpenSDK\_IotApConfig

### startAsyncIotApConfig

|  |  |
| --- | --- |
| 方法 | + (void)startAsyncIotApConfig:(NSString \*)wifiName  wifiPwd:(NSString \*\_Nullable)wifiPwd  productId:(NSString \*)productId  deviceId:(NSString \*)deviceId  finshed:(void(^\_Nullable)(BOOL success,  NSString \*\_Nullable errorMessage))finshed; |
| 简介 | IOT设备异步软AP配网 |
| 参数 | wifiName:wifi名称  wifiPwd:wifi密码  productId:产品ID  deviceId:设备序列号（S/N）  finshed:配网完成后回调，success是否成功，errorMessage失败错误提示 |
| 返回值 | 无 |
| 备注 | 无 |

## LCOpenSDK\_DeviceInit

### ­initDev

|  |  |
| --- | --- |
| 方法 | +(BOOL)initDev:(NSString \*)mac byPwdResetWay:(NSInteger)byPwdResetWay initStatus:(NSInteger)initStatus passWord:(NSString \*)password ip:(NSString\*)ip; |
| 简介 | 初始化设备 |
| 参数 | mac:设备Mac地址  byPwdResetWay:支持密码重置方式。按位确定密码重置方式，只在设备有初始化账号时有意义，bit0--支持预置手机号 bit1--支持预置邮箱 bit2--支持文件导出 bit3--支持国内注册手机号  initStatus:设备初始化状态，按位确定初始化状态。  passWord:设备初始化密码  ip:设备IP |
| 返回值 | YES:表示初始化设备成功，NO：表示初始化设备失败 |
| 备注 | 该接口必须与设备在同一局域网内调用 |

### checkPwdValidity

|  |  |
| --- | --- |
| 方法 | - (int)checkPwdValidity:(NSString \*)deviceID ip:(NSString \*)ip port:(NSInteger)port password:(NSString \*)password |
| 简介 | 检查设备密码是否可用 |
| 参数 | deviceID:设备序列号（S/N）  ip:设备IP地址 port:端口号 password:密码 |
| 返回值 | 0:接口调用成功，<0:接口调用失败 |
| 备注 | 无 |

### loginDeviceByIP

|  |  |
| --- | --- |
| 方法 | + (long)loginDeviceByIP:(NSString \*)devIP port:(NSInteger)port username:(NSString \*)username password:(NSString \*)password highLevelSecurity:(BOOL)security errorCode:(unsigned int \*)errorCode; |
| 简介 | 登录设备 |
| 参数 | devIp:设备IP地址 port:端口号 username:用户名 password:密码 security:YES--安全模式登录，NO--兼容模式登录 errorCode:错误码 |
| 返回值 | 登录句柄 |
| 备注 | 无 |

## LCOpenSDK\_Download

### shareMyInstance

|  |  |
| --- | --- |
| 方法 | + (LCOpenSDK\_Download\*)shareMyInstance; |
| 简介 | 初始化单例 |
| 参数 | 无 |
| 返回值 | 单例对象 |
| 备注 | 无 |

### setListener

|  |  |
| --- | --- |
| 方法 | - (void)setListener:(id<LCOpenSDK\_DownloadListener>)listener; |
| 简介 | 设置监听对象 |
| 参数 | listener:监听对象指针 |
| 返回值 | 无 |
| 备注 | 监听者需继承LCOpensdk\_DownloadListener类，可接收云录像和设备录像下载的状态码回调 |

### getListener

|  |  |
| --- | --- |
| 方法 | - (id<LCOpenSDK\_DownloadListener>)getListener; |
| 简介 | 获取监听对象指针 |
| 参数 | 无 |
| 返回值 | 监听对象指针 |
| 备注 | 无 |

### startDownloadCloudRecord

|  |  |
| --- | --- |
| 方法 | - (NSInteger)startDownloadCloudRecord:  (LCOpenSDK\_DownloadByRecordIdParam \*)paramRecord; |
| 简介 | 云录像下载 |
| 参数 | paramRecord:LCOpenSDK\_DownloadByRecordIdParam录像下载参数模型  其中，LCOpenSDK\_DownloadByRecordIdParam属性：  index:下载序号  savePath:下载保存路径  accessToken:管理员token/用户token  deviceId:设备序列号（S/N）  psk:设备密钥  productId:IOT设备产品ID  channelId:设备通道号  playToken:播放token  playToken[非必传]:播放码  useTLS:是否使用TLS，YES-使用，NO-不使用  speed[非必传]:视频倍速播放下载，下载速度可设置：1/2/4/8/16/32  recordRegionId:录像Id  cloudType:云录像类型，1000--报警云录像，2000--定时云录像 |
| 返回值 | 0:接口调用成功，-1：接口调用失败 |
| 备注 | 1.异步接口，实际下载状态需根据监听接口onDownloadState回调状态码判断，可参考demo；  2.psk参数传空指针或空字符串时使用设备序列号作为解密秘钥，默认传入值为设备序列号，若用户自定义秘钥，该字段传入自定义秘钥；  3.报警云录像ID和定时云录像ID分别通过<https://open.imoulife.com/developDoc/8>中QueryCloudRecords和QueryCloudPlanRecords接口请求查询。  4.一个index对应一个下载任务，重复的index执行此接口，只有最先的那个index有效。只有 index下载任务已经完成，重复调用此index才会有效。 |

### startDownloadDeviceRecordById

|  |  |
| --- | --- |
| 方法 | - (NSInteger)startDownloadDeviceRecordById:  (LCOpenSDK\_DownloadByRecordIdParam \*)paramDevRecord; |
| 简介 | 按照录像文件名下载设备录像 |
| 参数 | paramDevRecord:LCOpenSDK\_DownloadByRecordIdParam录像下载参数模型  其中，LCOpenSDK\_DownloadByRecordIdParam属性：  index:下载序号  savePath:下载保存路径  accessToken:管理员token/用户token  deviceId:设备序列号（S/N）  psk:设备密钥  productId:IOT设备产品ID  channelId:设备通道号  playToken:播放token  playToken[非必传]:播放码  useTLS:是否使用TLS，YES-使用，NO-不使用  speed[非必传]:视频倍速播放下载，下载速度可设置：1/2/4/8/16/32  fileID:设备录像文件名，平台queryLocalRecords协议返回字段值 |
| 返回值 | 0:接口调用成功，-1：接口调用失败 |
| 备注 | 1.异步接口，实际下载状态需根据监听接口onDownloadState回调状态码判断，可参考demo；  2. decryptKey参数传空指针或空字符串时使用deviceID作为解密秘钥，默认传入值为设备序列号（S/N），若用户自定义秘钥，该字段必现传入自定义秘钥；  3.一个index对应一个下载任务，重复的index执行此接口，只有最先的那个index有效。只有 index下载任务已经完成，重复调用此index才会有效。  4.部分设备存在录像文件无法播放的问题，如果有该方面需求，可联系乐橙开放平台。 |

### startDownloadDeviceRecordByUtcTime

|  |  |
| --- | --- |
| 方法 | - (NSInteger)startDownloadDeviceRecordByUtcTime:  (LCOpenSDK\_DownloadByUTCTimeParam \*)paramDevRecord; |
| 简介 | 按照时间下载设备录像 |
| 参数 | paramDevRecord:LCOpenSDK\_DownloadByUTCTimeParam录像下载参数模型  其中，LCOpenSDK\_DownloadByUTCTimeParam 属性：  index:下载序号  savePath:下载保存路径  accessToken:管理员token/用户token  deviceId:设备序列号（S/N）  psk:设备密钥  productId:IOT设备产品ID  channelId:设备通道号  playToken:播放token  playToken[非必传]:播放码  useTLS:是否使用TLS，YES-使用，NO-不使用  speed[非必传]:视频倍速播放下载，下载速度可设置：1/2/4/8/16/32  beginTime:开始时间  endTime:结束时间 |
| 返回值 | 0:接口调用成功，-1：接口调用失败 |
| 备注 | 1.异步接口，实际下载状态需根据监听接口onDownloadState回调状态码判断，可参考demo；  2. decryptKey参数传空指针或空字符串时使用deviceID作为解密秘钥，默认传入值为设备序列号（S/N），若用户自定义秘钥，该字段必现传入自定义秘钥；  3.一个index对应一个下载任务，重复的index执行此接口，只有最先的那个index有效。只有 index下载任务已经完成，重复调用此index才会有效。  4.部分设备存在录像文件无法播放的问题，如果有该方面需求，可联系乐橙开放平台。 |

### stopDownload

|  |  |
| --- | --- |
| 方法 | - (BOOL)stopDownload:(NSInteger)index; |
| 简介 | 停止下载录像 |
| 参数 | index:下载索引值 |
| 返回值 | YES:表示初始化设备成功，NO：表示初始化设备失败 |
| 备注 | 下载完成或失败，sdk内部会自动调用该接口，只有在终止正在进行的下载任务时才需调用该接口。 |

## LCOpenSDK\_DownloadListener

### onDownloadReceiveData

|  |  |
| --- | --- |
| 方法 | -(void)onDownloadReceiveData:(NSInteger)index datalen:(NSInteger)datalen; |
| 简介 | 录像下载数据长度回调 |
| 参数 | index:下载索引值  datalen:下载的数据长度(增量，单位字节) |
| 返回值 | 无 |
| 备注 | 录像的文件长度可通过<https://open.imoulife.com/developDoc/9> 中[QueryCloudRecords](#_LCOpenApi.QueryCloudRecords_1)接口请求）接口查询，datalen的累加值可以反映下载进度。 |

### onDownloadState

|  |  |
| --- | --- |
| 方法 | -(void)onDownloadState:(NSInteger)index code:(NSString \*)code type:(NSInteger)type; |
| 简介 | 录像下载状态回调 |
| 参数 | index:下载索引值  code:状态码  type:业务类型 |
| 返回值 | 无 |
| 备注 | 业务类型type如下：  RESULT\_PROTO\_TYPE\_RTSP = 0 设备录像  RESULT\_PROTO\_TYPE\_HLS = 1 云录像  RESULT\_PROTO\_TYPE\_OPENAPI = 99 Rest请求  具体业务类型对应的状态码可在LCOpenSDK\_Define类中查看 |

## LCOpenSDK\_EventListener

### onControlClick

|  |  |
| --- | --- |
| 方法 | - (void) onControlClick:(CGFloat)dx dy:(CGFloat)dy Index:(NSInteger)index; |
| 简介 | 播放窗口单击事件 |
| 参数 | dx:窗口X坐标  dy:窗口Y坐标  Index:窗口索引值 |
| 返回值 | 无 |
| 备注 | 实现即可接收播放窗口点击事件。 |

### onWindowDBClick

|  |  |
| --- | --- |
| 方法 | - (void) onWindowDBClick:(CGFloat)dx dy:(CGFloat)dy  Index:(NSInteger)index; |
| 简介 | 播放窗口双击事件 |
| 参数 | dx:窗口X坐标  dy:窗口Y坐标  index:窗口索引值 |
| 返回值 | 无 |
| 备注 | 实现即可接收播放窗口双击事件。 |

### onZoomBegin

|  |  |
| --- | --- |
| 方法 | - (void)onZoomBegin:(NSInteger)index; |
| 简介 | 手势缩放事件开始的回调 |
| 参数 | Index:窗口索引值 |
| 返回值 | 无 |
| 备注 | 不关注此事件，可不实现。 |

### onZooming

|  |  |
| --- | --- |
| 方法 | - (void) onZooming:(CGFloat)scale Index:(NSInteger)index; |
| 简介 | 手势缩放事件中的回调 |
| 参数 | index:窗口索引值  scale:相对缩放比例 |
| 返回值 | 无 |
| 备注 | 不关注此事件，可不实现。 |

### onZoomEnd

|  |  |
| --- | --- |
| 方法 | - (void) onZoomEnd:(ZoomType) zoom Index:(NSInteger)index; |
| 简介 | 手势缩放事件结束的回调 |
| 参数 | Index:窗口索引值  zoom:放大or缩小 |
| 返回值 | 无 |
| 备注 | 不关注此事件，可不实现。 |

### onWindowLongPressBegin

|  |  |
| --- | --- |
| 方法 | - (void)onWindowLongPressBegin:(Direction)dir dx:(CGFloat)dx dy:(CGFloat)dy Index:(NSInteger)index; |
| 简介 | 播放窗口长按事件开始的回调 |
| 参数 | index:窗口索引值  dir:方向，参考Direction结构，相对于播放窗口的中心点  dx:所在位置相对屏幕左上角的绝对横坐标  dy:所在位置相对屏幕左上角的绝对纵坐标 |
| 返回值 | 无 |
| 备注 | 无 |

### onWindowLongPressEnd

|  |  |
| --- | --- |
| 方法 | - (void)onWindowLongPressEnd:(NSInteger)index; |
| 简介 | 播放窗口长按事件结束的回调 |
| 参数 | index:窗口索引值 |
| 返回值 | 无 |
| 备注 | 无 |

### onSlipBegin

|  |  |
| --- | --- |
| 方法 | - (void)onSlipBegin:(Direction)dir dx:(CGFloat)dx dy:(CGFloat)dy Index:(NSInteger)index; |
| 简介 | 滑动开始的回调 |
| 参数 | index:窗口索引值  dir:方向，参考Direction结构  dx:滑动开始点相对屏幕左上角绝对横坐标  dy:滑动开始点相对屏幕左上角绝对纵坐标 |
| 返回值 | 无 |
| 备注 | 可根据此接口实现云台控制功能或播放窗口放大后的拖动，参考demo。 |

### onSlipping

|  |  |
| --- | --- |
| 方法 | - (void) onSlipping:(Direction)dir preX:(CGFloat)preX preY:(CGFloat)preY dx:(CGFloat)dx dy:(CGFloat)dy Index:(NSInteger)index; |
| 简介 | 滑动中的回调 |
| 参数 | index:窗口索引值  dir:方向，参考Direction结构  preX:滑动开始点相对屏幕左上角绝对横坐标  preY:滑动开始点相对屏幕左上角绝对纵坐标  dx:滑动过程中位置点相对屏幕左上角绝对横坐标  dy:滑动过程中位置点相对屏幕左上角绝对纵坐标 |
| 返回值 | 无 |
| 备注 | 可根据此接口实现云台控制功能或播放窗口放大后的拖动，参考demo。 |

### onSlipEnd

|  |  |
| --- | --- |
| 方法 | - (void) onSlipEnd:(Direction)dir dx:(CGFloat)dx dy:(CGFloat)dy  Index:(NSInteger)index; |
| 简介 | 滑动结束的回调 |
| 参数 | index:窗口索引值  dir:方向，参考Direction结构  dx:滑动结束点相对屏幕左上角绝对横坐标  dy:滑动结束点相对屏幕左上角绝对纵坐标 |
| 返回值 | 无 |
| 备注 | 可根据此接口实现云台控制功能或播放窗口放大后的拖动，参考demo。 |

### onPlayerResult

|  |  |
| --- | --- |
| 方法 | - (void) onPlayerResult:(NSString \*)code Type:(NSInteger)type  Index:(NSInteger)index; |
| 简介 | 视频播放状态回调 |
| 参数 | index 窗口索引值  code 状态码  type 业务类型 |
| 返回值 | 无 |
| 备注 | 业务类型type如下：  RESULT\_PROTO\_TYPE\_RTSP = 0 RTSP业务  RESULT\_PROTO\_TYPE\_HLS = 1 HLS业务  RESULT\_PROTO\_TYPE\_FILE = 2 FILE业务  RESULT\_PROTO\_TYPE\_NETSDK = 3 NETSDK业务  RESULT\_PROTO\_TYPE\_LCHTTP = 5 HTTP优化拉流业务  RESULT\_PROTO\_TYPE\_OPENAPI = 99 平台业务  具体业务类型的状态码可在LCOpenSDK\_Define类中查看，对状态码具体判断和处理可参考Demo |

### onResolutionChanged

|  |  |
| --- | --- |
| 方法 | - (void) onResolutionChanged:(NSInteger)width  Height:(NSInteger)height Index:(NSInteger)index; |
| 简介 | 分辨率改变的回调 |
| 参数 | Index:窗口索引值  width:宽  height:高 |
| 返回值 | 无 |
| 备注 | 无 |

### onReceiveData

|  |  |
| --- | --- |
| 方法 | - (void) onReceiveData:(NSInteger)len Index:(NSInteger)index; |
| 简介 | 视频播放数据回调 |
| 参数 | index:窗口索引值  len:数据长度 |
| 返回值 | 无 |
| 备注 | 可根据len计算码流速率。 |

### onStreamCallback

|  |  |
| --- | --- |
| 方法 | - (void) onStreamCallback:(NSData\*)data Index:(NSInteger)index; |
| 简介 | TS/PS标准流导出数据回调 |
| 参数 | data:PS/TS标准流数据  index:窗口索引值 |
| 返回值 | 无 |
| 备注 | 无 |

### onPlayBegan

|  |  |
| --- | --- |
| 方法 | - (void)onPlayBegan:(NSInteger)index; |
| 简介 | 视频播放开始回调 |
| 参数 | index:窗口索引值 |
| 返回值 | 无 |
| 备注 | 无 |

### onPlayFinished

|  |  |
| --- | --- |
| 方法 | - (void)onPlayFinished:(NSInteger)index; |
| 简介 | 录像回放结束的回调 |
| 参数 | index:窗口索引值 |
| 返回值 | 无 |
| 备注 | 设备录像和云录像回放结束时都会触发此回调。 |

### onPlayerTime

|  |  |
| --- | --- |
| 方法 | - (void) onPlayerTime:(long)time Index:(NSInteger)index; |
| 简介 | (录像)视频播放时间回调 |
| 参数 | index:窗口索引值  time:当前录像时间 |
| 返回值 | 无 |
| 备注 | 无 |

### onFileTime

|  |  |
| --- | --- |
| 方法 | - (void)onFileTime:(long)beginTime EndTime:(long)endTime Index:(NSInteger)index; |
| 简介 | 获取文件的起始结束时间 |
| 参数 | index:窗口索引值  time:UTC时间 |
| 返回值 | 无 |
| 备注 | 无 |

### onIVSInfo

|  |  |
| --- | --- |
| 方法 | - (void)onIVSInfo:(NSString\*)pIVSBuf nIVSType:(long)nIVSType nIVSBufLen:(long)nIVSBufLen nFrameSeq:(long)nFrameSeq; |
| 简介 | IVS解码回调函数 |
| 参数 | pIVSBuf:辅助帧数据  nIVSType:辅助帧数据类型  nIVSBufLen:辅助帧数据长度(json或智能帧结构体数据)  nFrameSeq:辅助帧id |
| 返回值 | 无 |
| 备注 | 可根据此接口实现判断云台操作是否到达边界，具体判断逻辑参考demo。 |

## LCOpenSDK\_Log

### shareInstance

|  |  |
| --- | --- |
| 方法 | + (instancetype)shareInstance; |
| 简介 | 初始化单例 |
| 参数 | 无 |
| 返回值 | 单例对象 |
| 备注 | 无 |

### setLogInfo

|  |  |
| --- | --- |
| 方法 | - (void)setLogInfo:(LCOpenSDK\_LogInfo \*)logInfo; |
| 简介 | 设置日志信息，设置后自动打开，打印设置等级以上的日志，包含设置等级，不设置默认LogLevelTypeFatal |
| 参数 | logInfo:LCOpenSDK\_LogInfo参数模型  其中，LCOpenSDK\_LogInfo 属性：  levelType:日志等级类型，分别为：  LogLevelTypeFatal 重大错误  LogLevelTypeError 错误信息  LogLevelTypeWarning 警告信息  LogLevelTypeInfo 主要用于生产环境中输出程序运行的一些重要信息  LogLevelTypeDebug 运行调试信息  logLevelTypeSecurityAll 安全打印所有日志信息，如密码，会以\*展示  LogLevelTypeALL 打印所有日志信息 |
| 返回值 | 无 |
| 备注 | 设置日志等级为Fatal不能关闭播放过程中和播放库调用系统硬件的相关debug日志打印。 |

### closeLog

|  |  |
| --- | --- |
| 方法 | - (void) closeLog; |
| 简介 | 关闭日志打印 |
| 参数 | 无 |
| 返回值 | 无 |
| 备注 | 此接口不能关闭播放过程中和播放库调用系统硬件的相关debug日志打印。 |

## LCOpenSDK\_LoginManager

### shareInstance

|  |  |
| --- | --- |
| 方法 | + (LCOpenSDK\_LoginManager \* \_Nonnull) shareMyInstance; |
| 简介 | 初始化单例 |
| 参数 | 无 |
| 返回值 | 单例对象 |
| 备注 | 无 |

### addDevices

|  |  |
| --- | --- |
| 方法 | - (NSInteger)addDevices:(NSString \* )token  devices:(NSArray <LCOpenSDK\_P2PDeviceInfo \*>\* )devices; |
| 简介 | P2P预打洞接口 |
| 参数 | token:账号token  devices:设备信息  其中，LCOpenSDK\_P2PDeviceInfo属性有：  did:设备序列号（S/N）  playToken:播放token |
| 返回值 | -2:初始化P2P服务失败  -1:devices数组为空或P2P预打洞失败  1:P2P预打洞成功 |
| 备注 | 无 |

### getP2pPort

|  |  |
| --- | --- |
| 方法 | - (NSUInteger)getP2pPort:(LCOpenSDK\_P2PPortInfo \* )p2pInfo  timeout:(NSInteger)time; |
| 简介 | 在P2P打洞成功后获取P2P端口 |
| 参数 | p2pInfo:LCOpenSDK\_P2PInfo模型  timeout：超时时间  其中，LCOpenSDK\_P2PInfo属性：  sn:设备序列号  productId:产品ID  port:端口号  user:用户名  pwd:密码 |
| 返回值 | 0:初始化P2P服务失败或P2P映射端口失败，>0:P2P映射端口 |
| 备注 | 无 |

## LCOpenSDK\_PlayBackListener

### onPlayPause

|  |  |
| --- | --- |
| 方法 | - (void)onPlayPause:(NSInteger)index; |
| 简介 | 视频播放暂停回调 |
| 参数 | index:播放窗口索引值 |
| 返回值 | 无 |
| 备注 | 无 |

### onPlayFinished

|  |  |
| --- | --- |
| 方法 | - (void)onPlayFinished:(NSInteger)index; |
| 简介 | 视频播放完成回调 |
| 参数 | index:播放窗口索引值 |
| 返回值 | 无 |
| 备注 | 无 |

### onRecordStop

|  |  |
| --- | --- |
| 方法 | - (void)onRecordStop:(NSInteger)error Index:(NSInteger)index; |
| 简介 | 停止录制 |
| 参数 | error:录制停止错误类型  index:播放窗口索引值 |
| 返回值 | 无 |
| 备注 | 无 |

### onPlayerTime

|  |  |
| --- | --- |
| 方法 | - (void)onPlayerTime:(long)time Index:(NSInteger)index; |
| 简介 | 视频播放帧时间回调 |
| 参数 | time:视频帧时间  index:播放窗口索引值 |
| 返回值 | 无 |
| 备注 | 无 |

## LCOpenSDK\_PlayBackWindow

### setPlayBackListener

|  |  |
| --- | --- |
| 方法 | -(void)setPlayBackListener:(id<LCOpenSDK\_PlayBackListener>)listener; |
| 简介 | 设置播放窗口监听对象 |
| 参数 | listener:监听对象指针 |
| 返回值 | 无 |
| 备注 | 无 |

### playCloudRecord

|  |  |
| --- | --- |
| 方法 | -(NSInteger)playCloudRecord:(LCOpenSDK\_ParamCloudRecord \*)paramCloud; |
| 简介 | 播放云录像 |
| 参数 | paramCloud:LCOpenSDK\_ParamCloudRecord云录像模型  其中，LCOpenSDK\_ParamCloudRecord属性：  accessToken:管理员token/用户token  deviceId:设备序列号  productId:产品ID  channel:通道  psk:设备密钥  playToken:播放token  useTLS:是否使用TLS  showWindow:是否展示播放窗口，默认展示  recordRegionID:录像ID  offsetTime:偏移时间  recordType:录像类型  timeOut:超时时间  isOpenAudio:是否开启音频，默认开启  speed:播放速度 |
| 返回值 | 0:接口调用成功，-1:接口调用失败 |
| 备注 | 无 |

### playDeviceRecordByFileName

|  |  |
| --- | --- |
| 方法 | -(NSInteger)playDeviceRecordByFileName:(LCOpenSDK\_ParamDeviceRecordFileName \*)paramDevRecord; |
| 简介 | 按文件名播放设备录像 |
| 参数 | paramDevRecord:LCOpenSDK\_ParamDeviceRecordFileName设备录像模型  其中，LCOpenSDK\_ParamDeviceRecordFileName属性：  accessToken:管理员token/用户token  deviceId:设备序列号  productId:产品ID  channel:通道  psk:设备密钥  playToken:播放token  useTLS:是否使用TLS  showWindow:是否展示播放窗口，默认展示  fileName:录像文件名  offsetTime:偏移时间  isOpt:是否使用长链接优化  defiMode:流媒体HD/SD模式  isOpenAudio:是否开启音频，默认开启  speed:播放速度 |
| 返回值 | 0:接口调用成功，-1:接口调用失败 |
| 备注 | 无 |

### playDeviceRecordByUtcTime

|  |  |
| --- | --- |
| 方法 | -(NSInteger)playDeviceRecordByUtcTime:(LCOpenSDK\_ParamDeviceRecordUTCTime \*)paramDevRecord; |
| 简介 | 按文件名播放设备录像 |
| 参数 | paramDevRecord:LCOpenSDK\_ParamDeviceRecordUTCTime设备录像模型  其中，LCOpenSDK\_ParamDeviceRecordUTCTime属性：  accessToken:管理员token/用户token  deviceId:设备序列号  productId:产品ID  channel:通道  psk:设备密钥  playToken:播放token  useTLS:是否使用TLS  showWindow:是否展示播放窗口，默认展示  beginTime:开始时间  endTime:结束时间  offsetTime:偏移时间  isOpt:是否使用长链接优化  defiMode:流媒体HD/SD模式  isOpenAudio:是否开启音频，默认开启  speed:播放速度 |
| 返回值 | 0:接口调用成功，-1:接口调用失败 |
| 备注 | 无 |

### stopRecordStream

|  |  |
| --- | --- |
| 方法 | - (NSInteger)stopRecordStream:(BOOL)isKeepLastFrame; |
| 简介 | 停止录像播放 |
| 参数 | isKeepLastFrame:是否保留最后一帧画面 |
| 返回值 | 0:接口调用成功，-1:接口调用失败 |
| 备注 | 无 |

### setPlaySpeed

|  |  |
| --- | --- |
| 方法 | - (NSInteger) setPlaySpeed:(float)speed; |
| 简介 | 设置倍速播放 |
| 参数 | speed:倍速速度 |
| 返回值 | 0:接口调用成功，-1:接口调用失败 |
| 备注 | 无 |

### getPlaySpeed

|  |  |
| --- | --- |
| 方法 | - (float)getPlaySpeed; |
| 简介 | 获取播放速度 |
| 参数 | 无 |
| 返回值 | 播放速度 |
| 备注 | 无 |

### addToPlayGroup

|  |  |
| --- | --- |
| 方法 | -(BOOL)addToPlayGroup:(long)groupId  isGroupBase:(BOOL)isGroupBase; |
| 简介 | 将播放窗口加入到分组中 |
| 参数 | groupId:视频播放组id  isGroupBase:分组基准 |
| 返回值 | YES:添加成功，NO:添加失败 |
| 备注 | 无 |

### hideVideoRender

|  |  |
| --- | --- |
| 方法 | - (void)hideVideoRender:(BOOL)hidden; |
| 简介 | 是否隐藏视频渲染 |
| 参数 | hidden:是否隐藏 |
| 返回值 | 无 |
| 备注 | 无 |

## LCOpenSDK\_PlayFileListener

### onPlayPause

|  |  |
| --- | --- |
| 方法 | - (void)onPlayPause:(NSInteger)index; |
| 简介 | 视频播放暂停回调 |
| 参数 | index:播放窗口索引值 |
| 返回值 | 无 |
| 备注 | 无 |

### onPlayFinished

|  |  |
| --- | --- |
| 方法 | - (void)onPlayFinished:(NSInteger)index; |
| 简介 | 视频播放完成回调 |
| 参数 | index:播放窗口索引值 |
| 返回值 | 无 |
| 备注 | 无 |

### onPlayerTime

|  |  |
| --- | --- |
| 方法 | - (void)onPlayerTime:(long)time Index:(NSInteger)index; |
| 简介 | 视频播放帧时间回调 |
| 参数 | time:视频帧时间  index:播放窗口索引值 |
| 返回值 | 无 |
| 备注 | 无 |

### onFileTime

|  |  |
| --- | --- |
| 方法 | - (void)onFileTime:(long)beginTime EndTime:(long)endTime Index:(NSInteger)index; |
| 简介 | 视频文件开始和结束时间回调 |
| 参数 | beginTime:开始时间  EndTime:结束时间  index:播放窗口索引值 |
| 返回值 | 无 |
| 备注 | 无 |

## LCOpenSDK\_PlayFileWindow

### setPlayFileListener

|  |  |
| --- | --- |
| 方法 | -(void)setPlayFileListener:(id<LCOpenSDK\_PlayFileListener>)listener; |
| 简介 | 设置播放窗口监听对象 |
| 参数 | listener:监听对象指针 |
| 返回值 | 无 |
| 备注 | 无 |

### setPlaySpeed

|  |  |
| --- | --- |
| 方法 | - (NSInteger)setPlaySpeed:(float)speed; |
| 简介 | 设置倍速播放 |
| 参数 | speed:倍速 |
| 返回值 | 0:接口调用成功，-1：接口调用失败 |
| 备注 | 无 |

### PlayFile

|  |  |
| --- | --- |
| 方法 | - (NSInteger)playFile:(NSString\*)fileName; |
| 简介 | 播放文件 |
| 参数 | fileName:文件名称 |
| 返回值 | 0:接口调用成功，-1：接口调用失败 |
| 备注 | 无 |

### stopFile

|  |  |
| --- | --- |
| 方法 | - (NSInteger)stopFile:(BOOL)isKeepLastFrame; |
| 简介 | 停止播放文件 |
| 参数 | isKeepLastFrame:是否保留最后一帧画面 |
| 返回值 | 0:接口调用成功，-1：接口调用失败 |
| 备注 | 无 |

## LCOpenSDK\_AudioTalk

### setListener

|  |  |
| --- | --- |
| 方法 | - (void) setListener:(id<LCOpenSDK\_TalkerListener>) lis; |
| 简介 | 设置监听对象 |
| 参数 | lis:监听指针 |
| 返回值 | 无 |
| 备注 | 监听者需继承LCOpenSDK\_TalkerListener类，可接收语音对讲的状态码回调 |

### getListener

|  |  |
| --- | --- |
| 方法 | - (id<LCOpenSDK\_TalkerListener>) getListener; |
| 简介 | 获取监听对象指针 |
| 参数 | 无 |
| 返回值 | 监听对象指针 |
| 备注 | 无 |

### playTalk

|  |  |
| --- | --- |
| 方法 | - (NSInteger) playTalk:(LCOpenSDK\_ParamTalk \*)paramTalk; |
| 简介 | 开始对讲 |
| 参数 | paramTalk:LCOpenSDK\_ParamTalk参数模型  其中，LCOpenSDK\_ParamTalk属性：  accessToken:管理员token  deviceID:设备序列号（S/N）  productId[非必传]:IOT设备产品号  channel:设备通道号  psk:解密密钥，未加密为设备序列号，加密为设置的密码  playToken[非必传]:播放码  useTLS:是否使用TLS，YES-使用，NO-不使用  isOpt:是否使用长连接优化拉流，YES-使用，NO-不使用  talkType:对讲类型，talk对讲，call呼叫，默认为talk |
| 返回值 | 0:接口调用成功，-1：接口调用失败 |
| 备注 | 1.四窗口以上同时播放isOpt建议选择false，四窗口以下isOpt建议选择true； |

### stopTalk

|  |  |
| --- | --- |
| 方法 | - (NSInteger) stopTalk; |
| 简介 | 停止对讲 |
| 参数 | 无 |
| 返回值 | 0:接口调用成功，-1：接口调用失败 |
| 备注 | 无 |

### playSound

|  |  |
| --- | --- |
| 方法 | - (NSInteger) playSound; |
| 简介 | 开启声音 |
| 参数 | 无 |
| 返回值 | 0:接口调用成功，-1：接口调用失败 |
| 备注 | 无 |

### stopSound

|  |  |
| --- | --- |
| 方法 | - (NSInteger) stopSound; |
| 简介 | 关闭声音 |
| 参数 | 无 |
| 返回值 | 0:接口调用成功，-1：接口调用失败 |
| 备注 | 无 |

### playVisualTalk（更新）

|  |  |
| --- | --- |
| 方法 | - (NSInteger)playVisualTalk:(LCOpenSDK\_ParamTalk \*)paramTalk videoSampleCfg:(LCOpenSDK\_MediaVideoSampleConfigParam \*)videoSampleCfg; |
| 简介 | 语音对讲开始 ,对讲前请设置监听 |
| 参数 | paramTalk:LCOpenSDK\_ParamTalk参数模型  其中，LCOpenSDK\_ParamTalk属性：  accessToken:管理员token  deviceID:设备序列号（S/N）  productId[非必传]:IOT设备产品号  channel:设备通道号  psk:解密密钥，未加密为设备序列号，加密为设置的密码  playToken[非必传]:播放码  useTLS:是否使用TLS，YES-使用，NO-不使用  isOpt:是否使用长连接优化拉流，YES-使用，NO-不使用  talkType:对讲类型，talk对讲，call呼叫，默认为talk  LCOpenSDK\_MediaVideoSampleConfigParam为可视对讲属性：  **可视对讲属性如下：**：  width **NSInteger [必传]** 设备摄像头分辨率宽度  heigh **NSInteger [必传]** 设备摄像头分辨率高度  I\_frame\_interval **NSInteger [必传]** 5s内的关键帧  encodeType **NSInteger [必传]** 解码类型  frameRate **NSInteger [必传]** 帧率  isCameraOpen **BOOL [必传]** 摄像头状态（true：开启，false：关闭）  softEncodeMode **BOOL [必传]** 是否软编码（true：是，false：否） |
| 返回值 | 0:接口调用成功，-1：接口调用失败 |
| 备注 | 无 |
|  |  |

### enableTalkVideo（更新）

|  |  |
| --- | --- |
| 方法 | - (BOOL)enableTalkVideo:(BOOL)isEnable |
| 简介 | 开启可视对讲音频 |
| 参数 | isEnable **BOOL [必传]** 是否开启 |
| 返回值 |  |
| 备注 | 用于可视对讲设备 |
|  |  |

### pushMediaData（更新）

|  |  |
| --- | --- |
| 方法 | -(BOOL)pushMediaData:(int)type mediaData:(NSData \*)data datalen:(int)datalen needSoftEncode:(BOOL)softEncode |
| 简介 | 下发媒体数据 |
| 参数 | type **int[必传]** 数据加密类型（传1表示h264加密）  data **NSData [必传]** 编码后的媒体数据  datalen **int[必传]** 数据长度  softEncode BOOL **[必传]** 是否软解码 |
| 返回值 |  |
| 备注 | 用于可视对讲设备 |
|  |  |

### startSampleAudio（更新）

|  |  |
| --- | --- |
| 方法 | - (int)startSampleAudio; |
| 简介 | 关闭音频采集 |
| 参数 |  |
| 返回值 | 0:接口调用成功，-1：接口调用失败 |
| 备注 | 无 |
|  |  |

### stopSampleAudio（更新）

|  |  |
| --- | --- |
| 方法 | - (int)stopSampleAudio; |
| 简介 | 关闭音频采集 |
| 参数 |  |
| 返回值 | 0:接口调用成功，-1：接口调用失败 |
| 备注 | 无 |
|  |  |

## LCOpenSDK\_PlayGroupManager

### shareInstance

|  |  |
| --- | --- |
| 方法 | + (instancetype)shareInstance; |
| 简介 | 初始化单例 |
| 参数 | 无 |
| 返回值 | 单例对象 |
| 备注 | 无 |

### createPlayGroup

|  |  |
| --- | --- |
| 方法 | - (long)createPlayGroup; |
| 简介 | 创建一个播放组 |
| 参数 | 无 |
| 返回值 | 播放组ID |
| 备注 | 无 |

## LCOpenSDK\_PlayListenerProtocol

### onPlayLoading

|  |  |
| --- | --- |
| 方法 | - (void)onPlayLoading:(NSInteger)index; |
| 简介 | 视频开始加载回调 |
| 参数 | index:播放窗口索引值 |
| 返回值 | 无 |
| 备注 | 无 |

### onPlayBegan

|  |  |
| --- | --- |
| 方法 | - (void)onPlayBegan:(NSInteger)index; |
| 简介 | 视频播放开始回调 |
| 参数 | index:播放窗口索引值 |
| 返回值 | 无 |
| 备注 | 无 |

### onPlayStop

|  |  |
| --- | --- |
| 方法 | - (void)onPlayStop:(NSInteger)index; |
| 简介 | 视频播放暂停回调 |
| 参数 | index:播放窗口索引值 |
| 返回值 | 无 |
| 备注 | 无 |

### onPlayFail

|  |  |
| --- | --- |
| 方法 | - (void)onPlayFail:(NSString\*)code Type:(NSInteger)type  Index:(NSInteger)index; |
| 简介 | 视频播放状失败回调 |
| 参数 | code:错误码(根据type而定) type:视频类型（0:RTSP，1:HLS，99:OPENAPI）  index:播放窗口索引值 |
| 返回值 | 无 |
| 备注 | 无 |

### onResolutionChanged

|  |  |
| --- | --- |
| 方法 | -(void)onResolutionChanged:(NSInteger) widthHeight:(NSInteger)height  Index:(NSInteger)index; |
| 简介 | 分辨率改变时回调 |
| 参数 | width:新分辨率宽度  height:新分辨率高度  index:播放窗口索引值 |
| 返回值 | 无 |
| 备注 | 无 |

### onReceiveData

|  |  |
| --- | --- |
| 方法 | -(void)onReceiveData:(NSInteger)len Index:(NSInteger)index; |
| 简介 | 视频播放数据回调 |
| 参数 | len:数据长度  index:播放窗口索引值 |
| 返回值 | 无 |
| 备注 | 无 |

### onStreamCallback

|  |  |
| --- | --- |
| 方法 | - (void)onStreamCallback:(NSData\*)data Index:(NSInteger)index; |
| 简介 | TS/PS标准流导出数据回调 |
| 参数 | data:标准流导出数据  index:播放窗口索引值 |
| 返回值 | 无 |
| 备注 | 无 |

## LCOpenSDK\_PlayRealListener

### onPtzLimit

|  |  |
| --- | --- |
| 方法 | - (void)onPtzLimit:(LCOpenSDK\_Direction)direction; |
| 简介 | 云台到达限位点 |
| 参数 | direction:到达限位的方向 |
| 返回值 | 无 |
| 备注 | 无 |

### onRecordStop

|  |  |
| --- | --- |
| 方法 | - (void)onRecordStop:(NSInteger)error Index:(NSInteger)index; |
| 简介 | 停止录制 |
| 参数 | error:录制停止错误类型  index:播放窗口索引值 |
| 返回值 | 无 |
| 备注 | 无 |

## LCOpenSDK\_PlayRealWindow

### setPlayRealListener

|  |  |
| --- | --- |
| 方法 | -(void)setPlayRealListener:(id<LCOpenSDK\_PlayRealListener>)listener; |
| 简介 | 设置播放窗口监听对象 |
| 参数 | listener:监听对象指针 |
| 返回值 | 无 |
| 备注 | 无 |

### playRtspReal

|  |  |
| --- | --- |
| 方法 | -(NSInteger)playRtspReal:(LCOpenSDK\_ParamReal\*)paramReal; |
| 简介 | 播放实时视频 |
| 参数 | paramReal:实时视频参数模型LCOpenSDK\_ParamReal  其中，LCOpenSDK\_ParamReal属性：  accessToken:管理员token/用户token  deviceId:设备序列号  productId:产品ID  channel:通道  psk:设备密钥  playToken:播放token  useTLS:是否使用TLS  showWindow:是否展示播放窗口，默认展示  defiMode:流媒体HD/SD模式  isOpt:是否使用长链接优化  imageSize:码流分辨率  isOpenAudio:是否开启音频，默认开启  isAssistFrame:是否开启辅助帧，默认关闭 |
| 返回值 | 0:接口调用成功，-1：接口调用失败 |
| 备注 | 无 |

### playRtspReal（更新）

|  |  |
| --- | --- |
| 方法 | -(NSInteger)playRtspReal:(LCOpenSDK\_ParamReal\*)paramReal; |
| 简介 | 播放实时视频 |
| 参数 | paramReal:实时视频参数模型LCOpenSDK\_ParamReal  其中，LCOpenSDK\_ParamReal属性：  accessToken:管理员token/用户token  deviceId:设备序列号  productId:产品ID  channel:通道  psk:设备密钥  playToken:播放token  useTLS:是否使用TLS  showWindow:是否展示播放窗口，默认展示  defiMode:流媒体HD/SD模式  isOpt:是否使用长链接优化  imageSize:码流分辨率  isOpenAudio:是否开启音频，默认开启  isAssistFrame:是否开启辅助帧，默认关闭  isSupportVideoTalk:是否支持可视对讲，默认关闭  （可视对讲设备预览时，需要传true，其余设备传false） |
| 返回值 | 0:接口调用成功，-1：接口调用失败 |
| 备注 | 无 |

### stopRtspReal

|  |  |
| --- | --- |
| 方法 | - (NSInteger)stopRtspReal:(BOOL)isKeepLastFrame; |
| 简介 | 停止实时视频播放 |
| 参数 | isKeepLastFrame:是否保留最后一帧画面 |
| 返回值 | 0:接口调用成功，-1:接口调用失败 |
| 备注 | 无 |

### controlMovePTZ

|  |  |
| --- | --- |
| 方法 | - (void)controlMovePTZ:(LCOpenSDK\_PTZControllerInfo \*)PTZControllerInfo |
| 简介 | 云台控制 |
| 参数 | PTZControllerInfo:云台控制信息  其中，LCOpenSDK\_PTZControllerInfo属性：  operation:云台转动方向  duration:云台移动时长，时长必须大于0，设备会根据设置时长进行转动，时长过大会导致设备旋转一周到达边界 |
| 返回值 | 无 |
| 备注 | 无 |

### addToPlayGroup

|  |  |
| --- | --- |
| 方法 | -(BOOL)addToPlayGroup:(long)groupId isGroupBase:(BOOL)isGroupBase; |
| 简介 | 将播放窗口加入到分组中 |
| 参数 | groupId:视频播放组id  isGroupBase:分组基准 |
| 返回值 | YES:表示初始化设备成功，NO：表示初始化设备失败 |
| 备注 | 无 |

### hideVideoRender

|  |  |
| --- | --- |
| 方法 | - (void)hideVideoRender:(BOOL)hidden; |
| 简介 | 是否隐藏视频渲染 |
| 参数 | hidden:是否隐藏 |
| 返回值 | 无 |
| 备注 | 无 |

## LCOpenSDK\_ PlayWindowProtocol

### initPlayWindow

|  |  |
| --- | --- |
| 方法 | - (id)initPlayWindow:(CGRect)frame Index:(NSInteger)index; |
| 简介 | 初始化播放窗口 |
| 参数 | frame:窗口大小  index:播放窗口索引 |
| 返回值 | LCOpenSDK\_PlayWindow指针 |
| 备注 | 无 |

### index

|  |  |
| --- | --- |
| 方法 | - (NSInteger)index; |
| 简介 | 窗口索引 |
| 参数 | 无 |
| 返回值 | 窗口索引值 |
| 备注 | 无 |

### uninitPlayWindow

|  |  |
| --- | --- |
| 方法 | - (void)uninitPlayWindow; |
| 简介 | 反初始化播放窗口 |
| 参数 | 无 |
| 返回值 | 无 |
| 备注 | 无 |

### setWindowFrame

|  |  |
| --- | --- |
| 方法 | - (void)setWindowFrame:(CGRect)frame; |
| 简介 | 设置播放窗口大小 |
| 参数 | frame:窗口大小 |
| 返回值 | 无 |
| 备注 | 无 |

### setSurfaceBGColor

|  |  |
| --- | --- |
| 方法 | - (void)setSurfaceBGColor:(UIColor\*)normalColor; |
| 简介 | 设置播放窗口背景色 |
| 参数 | normalColor:背景色 |
| 返回值 | 无 |
| 备注 | 无 |

### getWindowView

|  |  |
| --- | --- |
| 方法 | - (UIView\*)getWindowView; |
| 简介 | 获取播放窗口 |
| 参数 | 无 |
| 返回值 | 播放窗口视图 |
| 备注 | 无 |

### setTouchListener

|  |  |
| --- | --- |
| 方法 | -(void)setTouchListener:(id<LCOpenSDK\_TouchListener>)listener; |
| 简介 | 设置播放窗口手势事件监听 |
| 参数 | listener:监听对象指针 |
| 返回值 | 无 |
| 备注 | 无 |

### openTouchListener

|  |  |
| --- | --- |
| 方法 | - (void)openTouchListener:(BOOL)flag; |
| 简介 | 设置播放窗口手势操作事件是否被上层窗口捕获 |
| 参数 | flag:布尔值 |
| 返回值 | 无 |
| 备注 | 无 |

### playAudio

|  |  |
| --- | --- |
| 方法 | - (NSInteger) playAudio; |
| 简介 | 播放音频 |
| 参数 | 无 |
| 返回值 | 0:接口调用成功，-1:接口调用失败 |
| 备注 | 无 |

### stopAudio

|  |  |
| --- | --- |
| 方法 | - (NSInteger)stopAudio; |
| 简介 | 停止音频 |
| 参数 | 无 |
| 返回值 | 0:接口调用成功，-1:接口调用失败 |
| 备注 | 无 |

### snapShot

|  |  |
| --- | --- |
| 方法 | - (NSInteger)snapShot:(NSString\*)filePath; |
| 简介 | 截图 |
| 参数 | filePath:图片保存路径 |
| 返回值 | 0:接口调用成功，-1:接口调用失败 |
| 备注 | 无 |

### setStreamCallback

|  |  |
| --- | --- |
| 方法 | -(void)setStreamCallback:(LC\_OUTPUT\_STREAM\_FORMAT)streamFormat; |
| 简介 | 设置PS/TS标准流导出 |
| 参数 | streamFormat:标准流类型 |
| 返回值 | 无 |
| 备注 | 无 |

### doScale

|  |  |
| --- | --- |
| 方法 | - (void)doScale:(CGFloat)scale; |
| 简介 | 缩放操作 |
| 参数 | scale:缩放比例 |
| 返回值 | 无 |
| 备注 | 无 |

### getScale

|  |  |
| --- | --- |
| 方法 | - (CGFloat)getScale; |
| 简介 | 获取缩放比例 |
| 参数 | 无 |
| 返回值 | -1:接口调用失败，其他:接口调用成功 |
| 备注 | 无 |

### doTranslateX

|  |  |
| --- | --- |
| 方法 | - (void)doTranslateX:(CGFloat)x Y:(CGFloat)y; |
| 简介 | 滑动操作 |
| 参数 | x:播放窗口X坐标值  y:播放窗口Y坐标值 |
| 返回值 | 无 |
| 备注 | 无 |

### seek

|  |  |
| --- | --- |
| 方法 | - (void)doTranslateX:(CGFloat)x Y:(CGFloat)y; |
| 简介 | 录像拖动 |
| 参数 | timeInfo:相对开始时间偏移的秒数 |
| 返回值 | 0:接口调用成功，-1:接口调用失败 |
| 备注 | 无 |

### pause

|  |  |
| --- | --- |
| 方法 | - (NSInteger)pause; |
| 简介 | 暂停播放 |
| 参数 | 无 |
| 返回值 | 0:接口调用成功，-1:接口调用失败 |
| 备注 | 无 |

### resume

|  |  |
| --- | --- |
| 方法 | - (NSInteger)resume; |
| 简介 | 恢复播放 |
| 参数 | 无 |
| 返回值 | 0:接口调用成功，-1:接口调用失败 |
| 备注 | 无 |

### startRecord

|  |  |
| --- | --- |
| 方法 | - (NSInteger)startRecord:(NSString\*)filePath recordType:(NSInteger)nRecordType; |
| 简介 | 开始录制视频 |
| 参数 | filePath:录制视频保存路径  nRecordType:录制视频格式（0:dav，1:mp4） |
| 返回值 | 0:接口调用成功，-1:接口调用失败 |
| 备注 | 无 |

### stopRecord

|  |  |
| --- | --- |
| 方法 | - (NSInteger)stopRecord; |
| 简介 | 停止录制视频 |
| 参数 | 无 |
| 返回值 | 0:接口调用成功，-1:接口调用失败 |
| 备注 | 无 |

### setSEnhanceMode

|  |  |
| --- | --- |
| 方法 | - (void)setSEnhanceMode:(LCOpenSDK\_EnhanceMode)mode; |
| 简介 | 设置去噪模式等级 |
| 参数 | mode:音频降噪水平 |
| 返回值 | 无 |
| 备注 | 扩展功能 |

## LCOpenSDK\_SearchDevices

### shareSearchDevices

|  |  |
| --- | --- |
| 方法 | +(instancetype)shareSearchDevices; |
| 简介 | 初始化单例 |
| 参数 | 无 |
| 返回值 | 单例对象 |
| 备注 | 无 |

### startSearchDevices

|  |  |
| --- | --- |
| 方法 | -(void)startSearchDevicesWithDeviceId:(NSString \*)deviceId timeOut:(int)timeOutcallback:(void(^)(LCOpenSDK\_SearchDeviceInfo \*deviceInfo))searchDeviceCallBack; |
| 简介 | 开始搜索设备 |
| 参数 | deviceId:设备序列号 timeOut:搜索设备超时时间 searchDeviceCallBack:搜索设备回调 |
| 返回值 | 无 |
| 备注 | 无 |

### stopSearchDevices

|  |  |
| --- | --- |
| 方法 | - (void)stopSearchDevices; |
| 简介 | 停止搜索设备 |
| 参数 | 无 |
| 返回值 | 无 |
| 备注 | 无 |

### getLastError

|  |  |
| --- | --- |
| 方法 | + (unsigned int)getLastError; |
| 简介 | 获取错误信息 |
| 参数 | 无 |
| 返回值 | 错误信息 |
| 备注 | 无 |

## LCOpenSDK\_SoftApConfig

### startSoftAPConfig

|  |  |
| --- | --- |
| 方法 | + (void)startSoftAPConfig:(NSString \*)wifiName                   wifiPwd:(NSString \*)wifiPwd                 wifiEncry:(int)wifiEncry               netcardName:(NSString \*)netcardName                  deviceIp:(NSString \*)deviceIp                 devicePwd:(NSString \*)devicePwd                      isSC:(BOOL)isSC                   handler:(void(^)(NSInteger result))handler                   timeout:(int)timeout; |
| 简介 | 软AP配网 |
| 参数 | wifiName:wifi名字 wifiPwd:wifi密码 wifiEncry:wifi加密方式（0:off，2:WEP64bit，3:WEP128bit，4:WPA-PSK-TKIP，5:WPA-PSK-CCMP） netcardName:wifi网卡名字 deviceIp:设备IP devicePwd:设备密码 isSC:设备是否使用安全码（设备已经初始化时，该值为false） handler:回调 timeout:超时时间 |
| 返回值 | 0:配网成功  -1:必传参数为空  -2:搜索设备失败  -3:设备初始化失败  -4:设备登录失败  -5:查询网卡失败  -6:查询wifi信息失败  -7:连接wifi失败 |
| 备注 | 1.是否为安全码设备，可通过设备背面信息中是否有8位安全码确认，有isSC为YES，否则为NO。  2.devicePwd SC码设备传设备SC码，没有sc码的为设备初始化的时候用户自定义的密码，K5锁需要传admin； |

### getSoftApWifiList

|  |  |
| --- | --- |
| 方法 | + (void)getSoftApWifiList:(NSString \* \_Nonnull)deviceIP  port:(NSInteger)port  devicePassword:(NSString \* \_Nonnull)password  isSC:(BOOL)isSC  success:(void(^ \_Nullable)(NSArray <LCOpenSDK\_WifiInfo \*>\* \_Nullable list))success  failure:(void(^ \_Nullable)(NSInteger code, NSString\* \_Nullable describe))failure; |
| 简介 | 软AP配网获取设备wifi列表 |
| 参数 | deviceIP:设备IP地址 port:端口号 password:设备密码 isSC:是否为安全码设备  success:成功回调  failure:失败回调 |
| 返回值 | 无 |
| 备注 | 无 |

### queryWifiByLoginHandle

|  |  |
| --- | --- |
| 方法 | +(LCOpenSDK\_WifiInfo\*)queryWifiByLoginHandle:(long)loginHandle mssId:(NSString \*)mssid errorCode:(unsigned int\*)errorCode; |
| 简介 | 用句柄获取wifi信息 |
| 参数 | loginHandle:登录句柄 mssid:网络ssid  errorCode:错误码 |
| 返回值 | wifi信息 |
| 备注 | 无 |

### querySupportWlanConfigV3

|  |  |
| --- | --- |
| 方法 | + (BOOL)querySupportWlanConfigV3:(long)loginHandle; |
| 简介 | 查询设备是否支持第三代扫描能力 |
| 参数 | loginHandle:登录句柄 |
| 返回值 | YES:支持，NO:不支持 |
| 备注 | 无 |

## LCOpenSDK\_TalkerListener

### onTalkResult

|  |  |
| --- | --- |
| 方法 | -(void) onTalkResult:(NSString \*) error TYPE:(NSInteger) type; |
| 简介 | 语音对讲状态回调 |
| 参数 | error:错误码  type:业务类型 |
| 返回值 | 无 |
| 备注 | 无 |

### onAudioRecord

|  |  |
| --- | --- |
| 方法 | -(void) onAudioRecord:(Byte\*) pData dataLen:(int) dataLen audioFormat:(int)audioFormat sampleRate(int)sampleRate samleDeph:(int)sampleDepth; |
| 简介 | 设备端收到客户端对讲声音回调 |
| 参数 | pData:对讲声音数据  dataLen:pData的长度  audioFormat:音频格式  sampleRate:采样率  sampleDepth:采样深度 |
| 返回值 | 无 |
| 备注 | 无 |

### onAudioReceive

|  |  |
| --- | --- |
| 方法 | -(void) onAudioReceive:(Byte\*) pData dataLen:(int) dataLen audioFormat:(int)audioFormat sampleRate(int)sampleRate samleDeph:(int)sampleDepth; |
| 简介 | 设备端收到客户端对讲声音回调 |
| 参数 | pData:对讲声音数据  dataLen:pData的长度  audioFormat:音频格式  sampleRate:采样率  sampleDepth:采样深度 |
| 返回值 | 无 |
| 备注 | 无 |

## LCOpenSDK\_TouchListener

### onControlClick

|  |  |
| --- | --- |
| 方法 | - (void)onControlClick:(CGFloat)dx dy:(CGFloat)dy Index:(NSInteger)index; |
| 简介 | 单击回调 |
| 参数 | dx:窗口X坐标 dy:窗口Y坐标  index:播放窗口索引值 |
| 返回值 | 无 |
| 备注 | 无 |

### onWindowDBClick

|  |  |
| --- | --- |
| 方法 | -(void)onWindowDBClick:(CGFloat)dx dy:(CGFloat)dy Index:(NSInteger)index; |
| 简介 | 双击回调 |
| 参数 | dx:窗口X坐标 dy:窗口Y坐标  index:播放窗口索引值 |
| 返回值 | 无 |
| 备注 | 无 |

### onWindowLongPressBegin

|  |  |
| --- | --- |
| 方法 | -(void)onWindowLongPressBegin:(Direction)dir dx:(CGFloat)dx dy:(CGFloat)dy Index:(NSInteger)index; |
| 简介 | 长按开始回调 |
| 参数 | dir:长按相对窗口中的方向  dx:窗口X坐标 dy:窗口Y坐标  index:播放窗口索引值 |
| 返回值 | 无 |
| 备注 | 无 |

### onWindowLongPressEnd

|  |  |
| --- | --- |
| 方法 | -(void)onWindowLongPressEnd:(NSInteger)index; |
| 简介 | 长按结束回调 |
| 参数 | index:播放窗口索引值 |
| 返回值 | 无 |
| 备注 | 无 |

## LCOpenSDK\_Utils

### decryptPic

|  |  |
| --- | --- |
| 方法 | -(NSInteger)decryptPic:(NSData \*)pSrcBufIn deviceID:(NSString\*)deviceID key:(NSString\*)key bufOut:(NSData\*\*)pDestBufOut; |
| 简介 | 图片解密（非TCM设备） |
| 参数 | pSrcBufIn:原始图片内容字节数组  deviceID:设备序列号（S/N）  key:解密密钥  pDestBufOut:解密后数据存放数组 |
| 返回值 | 0:表示解密成功  1:表示完整性校验失败  2:表示密钥错误  3:表示图片非加密  4:不支持的加密方式  5:缓冲区长度不够  99:内部错误 |
| 备注 | 1. key解密密钥默认使用设备序列号（S/N）；  2.报警图片、云录像缩略图均为加密图片，设备通道缩略图为非加密图片；  3.TCM设备，可从平台deviceOpenDetailList协议的ability字段中判断是否为TCM设备，TCM设备会有TCM能力；  4.具体使用方法可参见Demo。 |

### decryptPic

|  |  |
| --- | --- |
| 方法 | -(NSInteger)decryptPic:(NSData \*)pSrcBufIn deviceID:(NSString\*)deviceID key:(NSString\*)key token:(NSString\*)token bufOut:(NSData\*\*)pDestBufOut; |
| 简介 | 图片解密（TCM设备） |
| 参数 | pSrcBufIn:原始图片内容字节数组  deviceID:设备序列号（S/N）  key:解密密钥  pDestBufOut:解密后数据存放数组  token:授权token  pDestBufOut:解密后数据存放数组 |
| 返回值 | 0:表示解密成功  1:表示完整性校验失败  2:表示密钥错误  3:表示图片非加密  4:不支持的加密方式  5:缓冲区长度不够  99:内部错误 |
| 备注 | 1. key解密密钥默认使用设备序列号（S/N）；  2.报警图片、云录像缩略图均为加密图片，设备通道缩略图为非加密图片；  3.TCM设备，可从平台deviceOpenDetailList协议的ability字段中判断是否为TCM设备，TCM设备会有TCM能力；  4.具体使用方法可参见Demo。 |

### decryptPic

|  |  |
| --- | --- |
| 方法 | - (NSInteger)decryptPic:(NSData \*)pSrcBufIn deviceID:(NSString\*)deviceID productId:(NSString \*)productId key:(NSString\*)key playtoken:(NSString \*)playtoken bufOut:(NSData\*\*)pDestBufOut; |
| 简介 | 图片解密（TCM设备） |
| 参数 | pSrcBufIn：原始图片内容字节数组  deviceID ：设备序列号（S/N）  key：解密密钥  pDestBufOut：解密后数据存放数组  playtoken：授权token  pDestBufOut：解密后数据存放数组 |
| 返回值 | 0:表示解密成功  1:表示完整性校验失败  2:表示密钥错误  3:表示图片非加密  4:不支持的加密方式  5:缓冲区长度不够  99:内部错误 |
| 备注 | 无 |

### mtsPreKeepalive

|  |  |
| --- | --- |
| 方法 | + (void)mtsPreKeepalive:(NSArray \*)deviceDatas token:(NSString \*)token; |
| 简介 | mts拉流提前建立长连接， |
| 参数 | deviceDatas:需要建立长连接的设备数据  token:用户token |
| 返回值 | 无 |
| 备注 | 无 |

## LCOpenSDK\_Netsdk

### initSDKWithCallback

|  |  |
| --- | --- |
| 方法 | + (void)initSDKWithCallback:(LCNetSDKDisconnetCallback)callback; |
| 简介 | 初始化 netsdk |
| 参数 | callback:netsdk连接断开回调 |
| 返回值 | 无 |
| 备注 | 无 |

### logOpen

|  |  |
| --- | --- |
| 方法 | +(void)logOpen:(NSString\*)path callback:(LCNetSDKLogOpenCallback)logCallback; |
| 简介 | 开启日志 |
| 参数 | path:日志存放路径 path:log path  logCallback:日志回调 logCallback:log callback |
| 返回值 | 无 |
| 备注 | 无 |

### logClose

|  |  |
| --- | --- |
| 方法 | + (void)logClose; |
| 简介 | 关闭日志 |
| 参数 | 无 |
| 返回值 | 无 |
| 备注 | 无 |

### initDevAccountWithDevInfo

|  |  |
| --- | --- |
| 方法 | +(BOOL)initDevAccountWithDevInfo:(LCNetsdkDevInfo\*)  deviceNetInfo useIp:(BOOL)isUseIp; |
| 简介 | 初始化设备 |
| 参数 | isUseIp:是否使用ip的方式deviceNetInfo 设备信息 |
| 返回值 | 无 |
| 备注 | 无 |

### logout

|  |  |
| --- | --- |
| 方法 | + (void)logout:(long)loginHandle; |
| 简介 | 登出设备 |
| 参数 | loginHandle:句柄 |
| 返回值 | 无 |
| 备注 | 无 |

### resetPassword

|  |  |
| --- | --- |
| 方法 | +(LCDevicePWDResetInfo\*)resetPassword:(NSString\*)password device:(LCNetsdkDevInfo\*)device securityCode:(NSString\*)securityCode contact:(NSString\*)contact useAsPreset:(BOOL)useAsPreset byPhoneIp:(NSString \*)phoneIp; |
| 简介 | 重置设备密码 |
| 参数 | password:密码 device:设备信息 securityCode:安全码 contact:接收验证码的手机/邮箱 useAsPreset:是否设置为预留联系方式 phoneIp:手机ip |
| 返回值 | 重置后的密码信息 |
| 备注 | 重置设备密码 |

### queryPasswordResetType

|  |  |
| --- | --- |
| 方法 | +(LCDeviceResetPWDInfo\*)queryPasswordResetType:(LCNetsdkDevInfo\*)device byPhoneIp:(NSString \*)phoneIp;; |
| 简介 | 异步查询设备支持的密码重置方式 |
| 参数 | device:设备 phoneIp:手机ip,可以为空 |
| 返回值 | Result |
| 备注 | 无 |

### queryProductDefinitionInfo

|  |  |
| --- | --- |
| 方法 | +(LCNetsdkProductDefinition\*)queryProductDefinitionInfo:  (long)loginHandle; |
| 简介 | 获取设备信息 |
| 参数 | loginHandle:句柄 |
| 返回值 | Result |
| 备注 | 无 |

### queryDevicePtzAuth

|  |  |
| --- | --- |
| 方法 | +(BOOL)queryDevicePtzAuth:(long)loginHandle; |
| 简介 | 获取是否具有云台权限 |
| 参数 | loginHandle:句柄 |
| 返回值 | True或False |
| 备注 | 无 |

## LCOpenSDK\_BuryPoint

### addBuryPointListener

|  |  |
| --- | --- |
| 方法 | +(void)addBuryPointListener:(id<LCOpenSDK\_BuryPointProtocol>)listener |
| 简介 | 添加埋点监听 |
| 参数 | listener监听的对象 |
| 返回值 | 无 |
| 备注 | 无 |

### removeBuryPointListener

|  |  |
| --- | --- |
| 方法 | +(void)removeBuryPointListener:(id<LCOpenSDK\_BuryPointProtocol>)listener |
| 简介 | 移除埋点监听 |
| 参数 | listener监听的对象 |
| 返回值 | 无 |
| 备注 | 无 |

## LCOpenSDK\_BuryPointProtocol

### buryPoint

|  |  |
| --- | --- |
| 方法 | -(void)buryPoint:(NSString \*)jsonString |
| 简介 | 埋点回调协议 |
| 参数 | 无 |
| 返回值 | jsonString 埋点回调数据 |
| 备注 | 埋点回调参数解释：   |  |  |  |  | | --- | --- | --- | --- | | **参数** | **类型** | **Optional** | **解释** | | event\_id | string | 非空 | 埋点类型，详细见event\_id表，为空时不返回 | | step | string | 非空 | 执行步骤，详细见step表，为空时不返回 | | pid | string | 可空 | iot设备产品ID，为空时不返回 | | did | string | 可空 | 设备ID，为空时不返回 | | beginTime | long | > 0 | 埋点的调用时间 | | dealTime | long | >= 0 | 执行步骤的总耗时，为0时不返回 | | failDesc | string | 可空 | 失败描述（失败的方法或描述），为空时不返回 | | result | string | 可空 | 成功（success）或失败（fail） |  |  |  | | --- | --- | | **类型   event\_id** | **解释** | | BlueTooth | 蓝牙配网 | | SmartConfig | smartConfig配网 | | SoftAP | 软Ap配网 | | SoftAPWiFi | 软Ap获取设备wifi列表 | | IotSoftAp | IoT软Ap配网 | | DeviceInit | 设备初始化 | | SearchDevice | 设备搜索 |  |  |  | | --- | --- | | **步骤  step** | **解释** | | Start | 开始 | | Scan | 扫描 | | Connect | 连接 | | Config | 配网 | | End | 结束 | |

## LCOpenSDK\_BleLock

### startBleLockConnect

|  |  |
| --- | --- |
| 方法 | + (void) startBleLockConnect:(NSString \*)deviceId  productId:(NSString \* \_Nullable)productId  outTime:(NSTimeInterval)outTime  finished:(nullable void(^)(BOOL success, NSString \*errorMessage))finished; |
| 简介 | 连接蓝牙锁 |
| 参数 | deviceId NSString[必传] : 设备ID productId NSString[非必传] : IOT设备产品id timeout NSString[必传] : 超时时间 单位 s  finshed [必传] : 设备蓝牙连接完成后的回调，sussess是否成功，errorMessage失败错误提示 |
| 返回值 | 无 |
| 备注 | 无 |

### startBleLockBind

|  |  |
| --- | --- |
| 方法 | + (void) startBleLockBind:(NSString \*)deviceId  productId:(NSString \*\_Nullable)productId  finshed:(nullable void(^)(BOOL success, NSInteger errorCode, NSString \*\_Nullable bindToken))finshed; |
| 简介 | 绑定蓝牙锁 |
| 参数 | deviceId NSString[必传] : 设备ID productId NSString[非必传] : IOT设备产品id  finshed [必传] : 设备绑定完成后的回调，sussess是否成功，errorCode错误码，bindToken 绑定成功后，设备返回的绑定密钥 |
| 返回值 | 无 |
| 备注 | 回调： **bindToken** NSString 绑定成功后，设备返回的绑定密钥  错误码：  1001 蓝牙锁绑定成功  1002 蓝牙锁连接失败  1003 蓝牙锁绑定失败 |

### startBleLockLogin

|  |  |
| --- | --- |
| 方法 | + (void) startBleLockLogin:(NSString \*)deviceId  productId:(NSString \*\_Nullable)productId  bindToken:(NSString \*)bindToken  finshed:(nullable void(^)(BOOL success, NSInteger errorCode, NSString \*\_Nullable loginToken))finshed; |
| 简介 | 登录蓝牙锁 |
| 参数 | deviceId NSString[必传] : 设备ID productId NSString[非必传] : IOT设备产品id  bindToken String[必传] : 绑定接口返回的token  finshed:登录蓝牙锁完成后的回调，sussess是否成功，errorCode 错误码loginToken 绑定成功后设备返回绑定 |
| 返回值 | 无 |
| 备注 | **loginToken** String 登录成功后，设备返回的登录密钥  错误码：  1001 蓝牙锁登录成功  1002 蓝牙锁连接失败  1003 蓝牙锁登录失败 |

### startBleLockOpen

|  |  |
| --- | --- |
| 方法 | + (void) startBleLockOpen:(NSString \*)deviceId  productId:(NSString \*\_Nullable)productId  loginToken:(NSString \*)loginToken  bindToken:(NSString \*)bindToken  finshed:(nullable void(^)(BOOL success, NSInteger errorCode))finshed; |
| 简介 | 登录蓝牙锁 |
| 参数 | deviceId NSString[必传] : 设备ID productId NSString[非必传] : IOT设备产品id  loginToken String[必传] : 登录接口返回的token  bindToken String[必传] : 绑定接口返回的token  finshed:登录蓝牙锁完成后的回调，sussess是否成功，errorCode 错误码loginToken 绑定成功后设备返回绑定 |
| 返回值 | 无 |
| 备注 | 错误码：  3000 远程开锁成功  3001 远程开锁失败 强制锁定状态  3002 远程开锁失败 管理员界面  3003 远程开锁失败 禁用状态(仅允许管理员开门）  3004 远程开锁失败 随机数错误  3005 远程开锁失败 其他错误 |