**先临三维消费扫描SDK用户手册**

**Version 1.5.0**

**先临三维科技股份有限公司**

|  |  |  |
| --- | --- | --- |
| 版本历史记录 | | |
| 版本号 | 时间 | 修改记录 |
| V1.0.0.0 | 20220824 | 新建 |
| V1.0.0.1 | 20230306 | 新增SDK接口 |
| V1.0.0.2 | 20230313 | 新增标定工具说明 |
| V1.0.0.3 | 20230316 | 新增网格数据接口 |
| V1.0.0.4 | 20230317 | 新增设备操作接口 |
| V1.0.0.5 | 20230330 | 新增获取当前亮度接口 新增设置工作距离范围接口 新增工作距离范围错误码 修改获取相机亮度范围bug 过滤longpress\_plus和longpress\_sub两个功能 |
| V1.0.0.6 | 20230713 | 删除标志点拼接模式  删除自动拼接模式  删除转台拼接模式  删除编码点拼接模式  新增平面检测接口  新增当前标志点接口  新增全量标志点接口 |

**目 录**

[1. SDK相关说明 5](#_Toc24065)

[1.1. 开发相关文件 5](#_Toc26674)

[SDK文件清单 5](#_Toc14821)

[使用注意事项 5](#_Toc13797)

[1.2. 主要软件执行组件 5](#_Toc26827)

[文件清单 5](#_Toc11480)

[使用注意事项 5](#_Toc5667)

[2. SDK头文件说明 6](#_Toc5852)

[3. SDK demo使用 6](#_Toc2070)

[4. 标定工具使用 8](#_Toc7846)

[5. SDK 已知问题 8](#_Toc3823)

[6. 数据结构与宏定义 9](#_Toc2583)

[数据结构 9](#_Toc15556)

[6.1.1. 图片数据 9](#_Toc24514)

[6.1.2. 点数据 10](#_Toc3942)

[6.1.3. 点云数据 11](#_Toc6548)

[增量点云数据 12](#_Toc18150)

[6.1.4. 相机视角 13](#_Toc21966)

[6.1.5. 面片与纹理索引 14](#_Toc13616)

[6.1.6. 纹理UV坐标 15](#_Toc16555)

[6.1.7. 纹理图 16](#_Toc30601)

[6.1.8. 网格数据 17](#_Toc1133)

[6.1.9. 标志点数据 17](#_Toc20727)

[返回值定义 18](#_Toc15017)

[拼接模式AlignType类型 19](#_Toc21714)

[设备操作DeviceEvent类型 19](#_Toc28931)

[6.2. 回调函数声明 20](#_Toc19229)

[6.2.1. 初始化回调 20](#_Toc9404)

[6.2.2. 点云回调 20](#_Toc17486)

[6.2.3. 当前帧回调 21](#_Toc29629)

[6.2.4. 增量点云回调 21](#_Toc16558)

[6.2.5. 相机视角回调 22](#_Toc30479)

[6.2.6. 跟踪丢失回调 22](#_Toc12163)

[6.2.7. 跟踪距离回调 23](#_Toc6610)

[6.2.8. 网格数据回调 23](#_Toc27795)

[6.2.9. 设备操作回调 24](#_Toc6453)

[6.2.10. 是否平面回调 24](#_Toc22219)

[6.2.11. 当前标志点回调 24](#_Toc32008)

[6.2.12. 全量标志点回调 25](#_Toc12553)

[7. 主要流程示意 26](#_Toc5579)

[8. 接口定义 27](#_Toc18018)

[8.1.1. 基本功能接口 27](#_Toc13433)

[8.1.2. 设备控制接口 29](#_Toc21842)

[8.1.3. 扫描功能接口 34](#_Toc12633)

[8.1.4. 封装功能接口 39](#_Toc8637)

# SDK相关说明

## **开发相关文件**

### **SDK文件清单**

|  |  |  |
| --- | --- | --- |
| 文件及文件夹 | 功能说明 | 备注 |
| 先临三维Einstar SDK用户手册.docx | 扫描软件接口说明文档 |  |
| Include | SDK头文件声明 |  |
| Lib | SDK相关 |  |
| Bin | 软件执行包 |  |
| ScannerDemo | SDK demo软件源码 |  |
| calibration\_tool | 标定工具 |  |

### **使用注意事项**

1. **编译环境vs2017+**Qt5.12.8
2. **编译后**ScannerDemo.exe输出在bin文件下(demo install会拷贝)
3. 拷贝zmq\_sdk\_config.ini到工作目录下修改isRemote=1(demo install会拷贝)
4. 拷贝sn3d\_smc.dll输出在bin文件下(demo install会拷贝)
5. 拷贝sn3dE10SDK.dll输出在bin文件下(demo install会拷贝)

## **主要软件执行组件**

### **文件清单**

|  |  |  |
| --- | --- | --- |
| 文件名 | 功能说明 | 配置参数 |
| Bin\zmq\_sdk\_config.ini | SDK配置文件 | isRemote：0 不支持SDK接入  1 支持SDK接入  Pub： public 消息通讯端口，  默认值11398  Rep： Request消息通讯端口  默认值11399 |
| Bin\scanservice.exe | 扫描服务 | 实现扫描功能和数据处理 |
| ScannerDemo.exe | 扫描客户端 demo | 远程调用扫描服务器实现扫描功能 |
| Bin\sn3dE10SDK.dll | SDK链接库 |  |
|  |  |  |

### **使用注意事项**

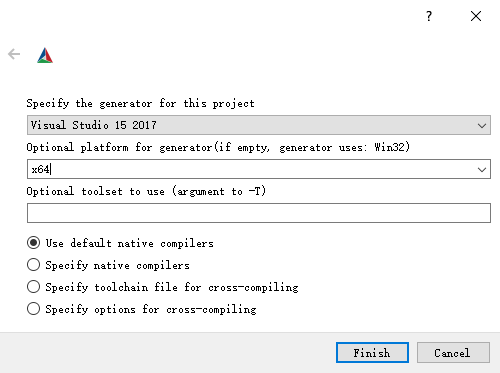
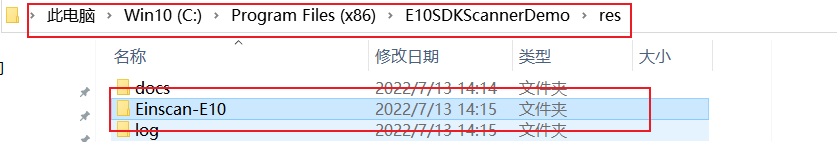
1 注意通讯端口11398、11399如果被机器上其它软件占用，需要修改zmq\_sdk\_config.ini

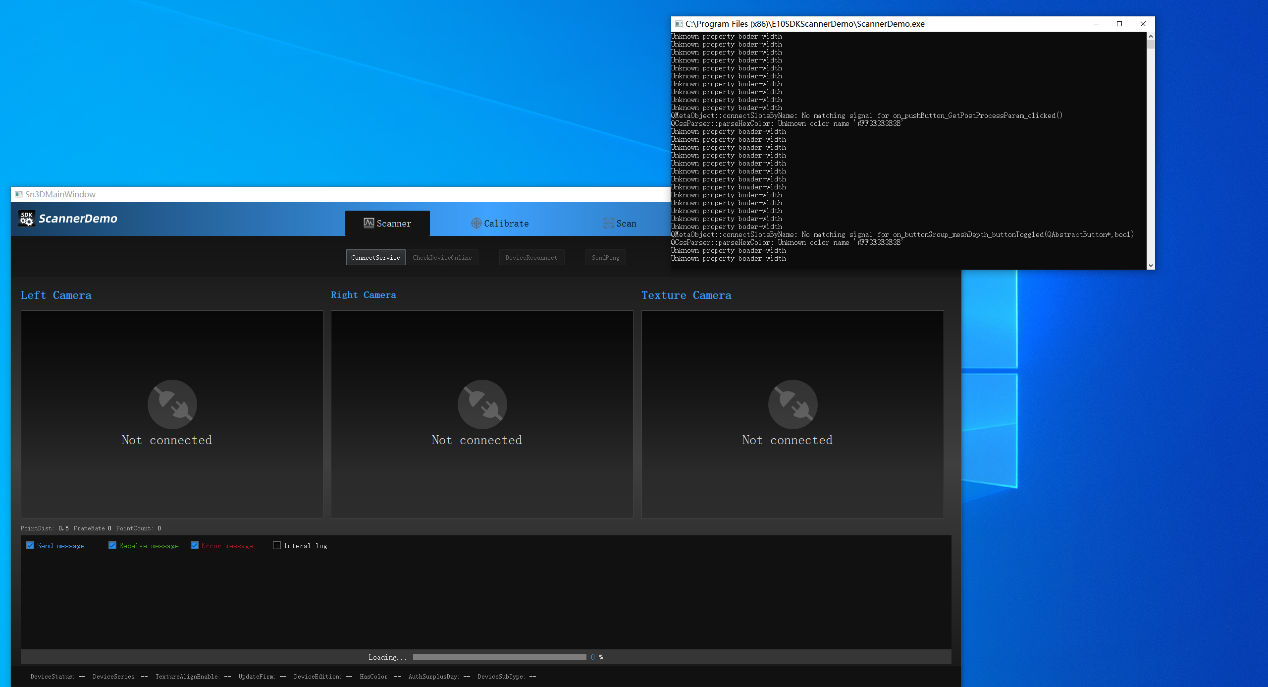
中的配置端口

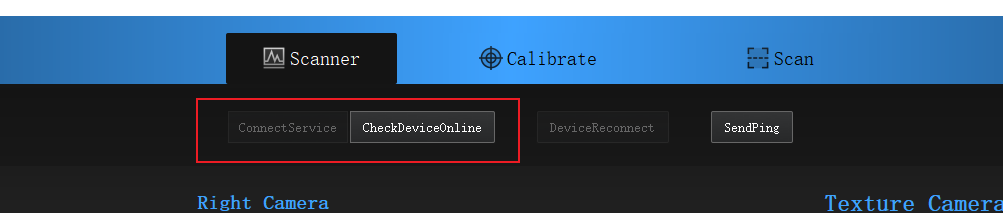
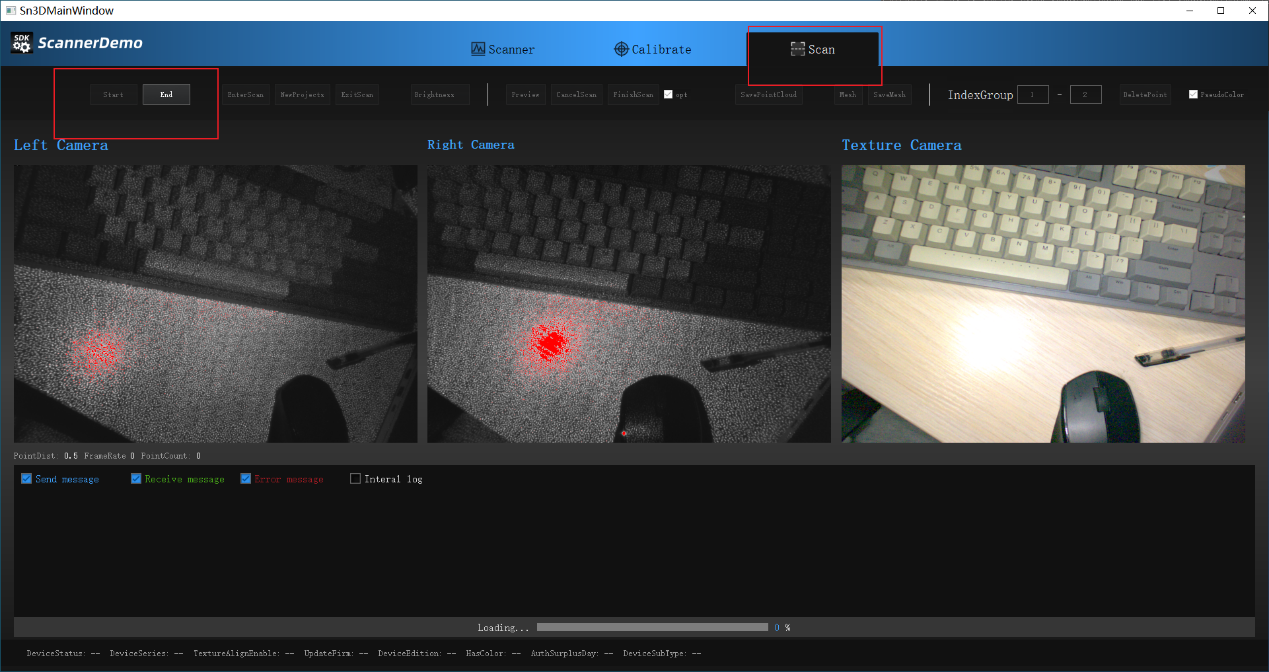
# SDK头文件说明

|  |  |  |
| --- | --- | --- |
| 文件名 | 说明 | 备注 |
| Include\Sn3DE10Sdk.h | 包含sdk接口声明 |  |
| Include\Sn3DErrorCode.h | 包含错误码信息 |  |
| Include\Sn3DPublic.h | 包含自定义结构体 |  |
| Include\PublicClass.h | 一些公共定义 |  |

# SDK demo使用

1. 安装CMake3.17.2及以上版本, 安装QT 5.12.8版本;
2. 使用cmake构建解决方案, Visual Studio 15 2017 + QT 5.12.8;
3. vs 2017在release x64下生成解决方案
4. 将ScannerDemo.exe、zmq\_sdk\_config.ini及文件夹bin下的所有动态库拷贝到EinScan发行软件根目录下。
5. 发行软件目录下执行ScannerDemo.exe



1. 依次执行connectService(sdk初始化)和connectdevice(连接设备).
2. 跳转到tab页scanner, 点击start(启动扫描)和end(生成数据);

# 标定工具使用

1. 将calibration\_tool.exe拷贝到发行软件目录中；
2. 连接设备，直接启动即可，根据提示进行标定；
3. 标定计算完成后，在目录res/Einscan-E10/200x150会生成标定文件；

注：启动标定程序时，确保ple文件存在，设备正常接入，否则无法正常标定。

# SDK 已知问题

1. 中文路径下新建工程乱码

# 数据结构与宏定义

## **数据结构**

### **图片数据**

**SN3D\_IMAGE\_DATA**

|  |
| --- |
| typedef struct tag **SN3D\_**IMAGE\_DATA  {  int width;  int height ;  int channel ;  int length ;  unsigned char\* data;  } **SN3D\_IMAGE\_DATA**, **\*LPSN3D\_IMAGE\_DATA**; |

**Members**

width

图像宽度；

height

图像高度；

channel

图像通道数；

length

图像数据长度=width\*height；

data

图像数据。

R**emarks**

### **点数据**

**SN3D\_POINT\_DATA**

|  |
| --- |
| typedef struct  {  float x; //X coordinate value  float y; //Y coordinate value  float z; //Z coordinate value  }Sn3dPointData, \*LPSn3dPointData; |

**Members**

x

x坐标。

y

y坐标。

z

z坐标。

**Remarks**

### **点云数据**

**SN3D\_CLOUD\_POINT**

|  |
| --- |
| typedef struct  {  int vertex\_count; //The number of vertex .  LPSn3dPointData vertex\_data; //The data of vertex.  int norma\_count; //The number of vertex normal.  LPSn3dPointData norma\_data; //The data of vertex normal.  int vertex\_color\_count; //The number of vertex color.  LPSn3dPointData vertex\_color\_data; //The data of vertex color.  }Sn3DPointCloud,\*LPSn3dPointCloud; |

**Members**

vertex\_count

顶点数据个数。

vertex\_data

顶点数据。

norma\_count

顶点法向数据个数。

norma\_data

顶点法向数据。

vertex\_ color \_count

顶点颜色数据个数。

vertex\_color\_data

顶点颜色数据。

**Remarks**

增量点云数据

**SN3D\_INCREASE\_POINT\_CLOUD**

|  |
| --- |
| typedef struct  {  int vertex\_count; //The number of vertex .  LPSn3dPointData vertex\_data; //The data of vertex.  int norma\_count; //The number of vertex normal.  LPSn3dPointData norma\_data; //The data of vertex normal.  int vertex\_color\_count; //The number of vertex color.  LPSn3dPointData vertex\_color\_data; //The data of vertex color.  int index\_count; //The number of index  unsigned int\* index; //The data of index  }Sn3DIncreasePointCloud,\*LPSn3dIncreasePointCloud; |

**Members**

vertex\_count

顶点数据个数。

vertex\_data

顶点数据。

norma\_count

顶点法向数据个数。

norma\_data

顶点法向数据。

vertex\_ color \_count

顶点颜色数据个数。

vertex\_color\_data

顶点颜色数据。

index\_count

索引个数

index

索引

**Remarks**

### **相机视角**

**SN3D\_CAMERA\_POSITION**

|  |
| --- |
| typedef struct  {  QVector3D position;  QVector3D center;  QVector3D up;  }Sn3DCameraPosition,\*LPSn3dCameraPosition; |

**Members**

position

观察者位置（相机视角位置）

center

被观察者位置（视角中心）

up

视角顶部的向量

**Remarks**

### **面片与纹理索引**

**SN3D\_FACE\_ID**

|  |
| --- |
| typedef struct  {  int x;  int y;  int z;  }Sn3DFaceId, \*LPSn3dFaceId; |

**Members**

x, y, z: 面片索引

**Remarks**

### 纹理UV坐标

**SN3D\_VEC2F**

|  |
| --- |
| typedef struct  {  float x;  float y;  }Sn3DVec2F, \*LPSn3dVec2F; |

**Members**

x, y: 面片坐标值

**Remarks**

### 纹理图

**SN3D\_IMAGE**

|  |
| --- |
| typedef struct  {  int width;  int height;  int channel;  uchar\* data;  //格式默认为RGB  }Sn3DImage; |

**Members**

width: 纹理图宽

height: 纹理图高

channel: 纹理图通道数

data: 纹理图数据

**Remarks**

### 网格数据

SN3D\_MESHDATA

|  |
| --- |
| typedef struct  {  //点云数量和点云数据  int meshpoint\_count;  LPSn3dPointData meshpoint;  //法向数量和数据  int meshnormal\_count;  LPSn3dPointData meshnormal;  //点云索引  int meshtrifaceid\_count;  LPSn3dFaceId meshtrifaceid;  //纹理索引  int meshtextureid\_count;  LPSn3dId meshtextureid;  //纹理UV坐标  int textureUV\_count;  LPSn3dVec2F textureUV;  //纹理图  Sn3DImage image;  }Sn3DMeshData, \*LPSn3dMeshData |

**Members**

meshpoint\_coun: 点云数量

meshpoint: 点云数据

meshnormal\_count: 点云法向数量

meshnormal: 点云法向数据

meshtrifaceid\_count: 点云索引ID数量

meshtrifaceid: 点云索引ID数据

meshtextureid\_count: 纹理索引ID数量

meshtextureid: 纹理索引ID数据

textureUV\_count: 纹理图UV坐标数量

textureUV: 纹理图UV坐标

image: 纹理图

**Remarks**

### **标志点数据**

SN3D\_MARKERDATA

|  |
| --- |
| //单个标志点信息 single marker  struct MarkerInfo  {  Sn3dPointData coord; //坐标 data of marker  Sn3dPointData normal; //法向 data of noraml  double diameter; //直径 diameter  int codeId; //编码点ID(<=0即为普通的标志点) marker id (<=0 means common marker)  };  typedef struct {  int marker\_count; //marker count  MarkerInfo\* markerList; //marker list  }Sn3DMarkerData, \*LPSn3DMakerData; |

**Members**

单个标志点信息：

coord: 标志点坐标

normal: 标志点法向

diameter: 标志点直径

codeId: 标志点id，小于等于0代表此标志点是普通标志点

标志点集合信息:

marker\_count:标志点个数

markerList: 标志点集合

**Remarks**

### 返回值定义

|  |  |  |
| --- | --- | --- |
| **宏定义** | **宏定义值** | **含义** |
| **EC\_SUCCESS** | 0 | 没有错误 |
| **EC\_NOTINITIALIZED** | 1 | 没有初始化 |
| **EC\_INITIALIZEFIAILZED** | 2 | 初始化失败 |
| **EC\_AlREADYINITIALIZED** | 3 | 重复初始化 |
| **EC\_SAVEFAILED\_TYPEERROR** | 4 | 保存文件失败，格式错误 |
| **EC\_CHECKDEVICEFAILED\_PLENOTRIGHT** | 5 | 连接设备失败，ple错误 |
| **EC\_CHECKDEVICEFAILED\_NODEVICEFOUND** | 6 | 连接设备失败，未发现设备 |
| **EC\_OPENORCREATSLNFAILED** | 7 | 创建或打开工程失败 |
| **EC\_ENTERSCANFAILED** | 8 | 进入扫描失败 |
| **EC\_SCANFAILED** | 9 | 扫描中失败 |
| **EC\_ENDSCANFAILED** | 10 | 生成失败 |
| **EC\_MESHFAILED** | 11 | 网格化失败 |
| **EC\_EXITSCANFAILED** | 12 | 退出扫描失败 |
| **EC\_CANCELSCANFAILED** | 13 | 取消扫描失败 |
| **EC\_SAVEFAILED** | 14 | 保存失败 |
| **EC\_CREATNEWPROJECTFAILED** | 16 | 新建工程失败 |
| **EC\_WORKRANGE** | 17 | 工作距离范围不符合要求 |
| **EC\_PROCESSERROR** | 268435456 | 流程调用错误 |

### 拼接模式AlignType类型

|  |  |  |
| --- | --- | --- |
| **枚举** | **值** | **含义** |
| **AT\_FEATURES** | 0 | 特征拼接 |
| **AT\_HYBRID** | 2 | 混合拼接 |
| **AT\_GLOBLE\_POINT** | 7 | 框架点拼接 |
| **AT\_TEXTURE** | 8 | 纹理拼接 |

### 设备操作DeviceEvent类型

|  |  |  |
| --- | --- | --- |
| **枚举** | **值** | **含义** |
| **DE\_NULL** | -1 | 空 |
| **DE\_DOUBLECLICK** | 0 | 双击开始键 |
| **DE\_CLICK** | 1 | 单机开始键 |
| **DE\_PLUS** | 2 | 单机加键 |
| **DE\_SUB** | 3 | 单机减键 |
| **DE\_LONGPRESS** | 8 | 长按开始键 |

## 回调函数声明

### 初始化回调

Sn3DScanServiceWatcherCallBack

初始化设备设置回调

|  |
| --- |
| **typedef void (CALLBACK \*Sn3DScanServiceWatcherCallBack)(int);** |

**Remarks**

注册服务端监控回调，服务端退出会收到通知

### 点云回调

**Sn3DWholePointCloudCallBack**

获取整体点云数据回调

|  |
| --- |
| **typedef void(CALLBACK \*Sn3DWholePointCloudCallBack)(LPSn3dPointCloud wholePointCloud);** |

**Parameter**

**wholePointCloud**

**[out] 点云数据**

**Remarks**

1. 在sn3dEndscan(结束扫描)和sn3dMesh(网格化)设置回调可以取到不同阶段的点云集合
2. 设置一次回调成功触发后需要重新设置回调.

### 当前帧回调

**Sn3DCurrentPointCloudCallBack**

获取当前帧点云数据回调

|  |
| --- |
| **typedef void(CALLBACK \*Sn3DCurrentPointCloudCallBack)(LPSn3dPointCloud currentPointCloud, void\* owner);** |

**Parameter**

**currentPointCloud**

**[out] 当前帧点云数据**

**owner**

**[out] 回调函数拥有者的指针**

**Remarks**

1.在Sn3DInitialize(初始化)后设置回调可以取到扫描全过程当前帧的点云集合。

### 增量点云回调

**Sn3DIncreasePointCloudCallBack**

获取增量点云数据回调

|  |
| --- |
| **typedef void(CALLBACK \*Sn3DIncreasePointCloudCallBack)(LPSn3dIncreasePointCloud increasePointCloud, void\* owner);** |

**Parameter**

**increasePointCloud**

**[out] 增量点云数据**

**owner**

**[out] 回调函数拥有者的指针**

**Remarks**

1.在Sn3DInitialize(初始化)后设置回调可以取到扫描全过程的增量点云集合。

### 相机视角回调

**Sn3DCameraPositionCallBack**

获取相机视角数据回调

|  |
| --- |
| **typedef void(CALLBACK \*Sn3DCameraPositionCallBack)(LPSn3dCameraPosition cameraPosition, void\* owner);** |

**Parameter**

**cameraPosition**

**[out] 相机视角数据**

**owner**

**[out] 回调函数拥有者的指针**

**Remarks**

1.在Sn3DInitialize(初始化)后设置回调可以取到扫描全过程的相机视角数据集合

### 跟踪丢失回调

**Sn3DTrackLostStatusCallBack**

获取跟踪丢失状态回调

|  |
| --- |
| **typedef void(CALLBACK \*Sn3DTrackLostStatusCallBack)(bool trackLostStatus, void\* owner);** |

**Parameter**

**trackLostStatus**

**[out] 跟踪丢失状态，true: 跟踪丢失；false: 未跟踪丢失**

**owner**

**[out] 回调函数拥有者的指针**

**Remarks**

1.在Sn3DInitialize(初始化)后设置回调可以取到扫描全过程的跟踪丢失的状态。

### 跟踪距离回调

**Sn3DScanDistCallBack**

获取跟踪距离挡位数据回调

|  |
| --- |
| **typedef void(CALLBACK \*Sn3DScanDistCallBack)(double scanDist, void\* owner);** |

**Parameter**

**scanDist**

**[out] 扫描距离挡位**

**owner**

**[out] 回调函数拥有者的指针**

**Remarks**

1.在Sn3DInitialize(初始化)后设置回调可以取到扫描全过程的扫描距离。

2.scanDist = -1时，距离太近；scanDist = 100时，距离太远；scanDist=-2时，距离无效（可能未重建点云）；其他为正常(1~12)。

### 网格数据回调

**Sn3DMeshDataCallBack**

获取网格数据回调函数

|  |
| --- |
| **typedef void(CALLBACK \*Sn3DMeshDataCallBack)(LPSn3dMeshData meshData, void\* owner);** |

**Parameter**

**meshData**

**[out] 网格数据**

**owner**

**[out] 回调函数拥有者的指针**

**Remarks**

需要在结束扫描Sn3DEndScan后调用

### 设备操作回调

**Sn3DDeviceEventCallBack**

获取网格数据回调函数

|  |
| --- |
| **typedef void(CALLBACK \*Sn3DDeviceEventCallBack)(DeviceEvent event, void\* owner);** |

**Parameter**

**event**

**[out] 设备操作的数据类型**

**owner**

**[out] 回调函数拥有者的指针**

**Remarks**

1. 需要在Sn3DInitialize(初始化)后设置回调

### 平面检测回调

**Sn3DTooFlatStatusCallBack**

获取当前扫描是否为平面，如果为平面则提示

**typedef void(CALLBACK \*Sn3DTooFlatStatusCallBack)(bool tooFlatStatus, void\* owner);**

**Parameter**

**tooFlatStatus**

**[out] 是否为平面,当是平面的时候输出true,否则输出false**

**owner**

**[out] 回调函数拥有者的指针**

### 当前标志点回调

**Sn3DCurrentMarkerCallBack**

获取当前标志点回调函数

**typedef void(CALLBACK \*Sn3DCurrentMarkerCallBack)(LPSn3DMakerData currentMarker, void\* owner);**

**Parameter**

**currentMarker**

**[out] 当前标志点数据**

**owner**

**[out] 回调函数拥有者的指针**

### 全量标志点回调

**Sn3DWholeMarkerCallBack**

获取当前标志点回调函数

**typedef void(CALLBACK \*Sn3DWholeMarkerCallBack)(LPSn3DMakerData wholeMarker, void\* owner);**

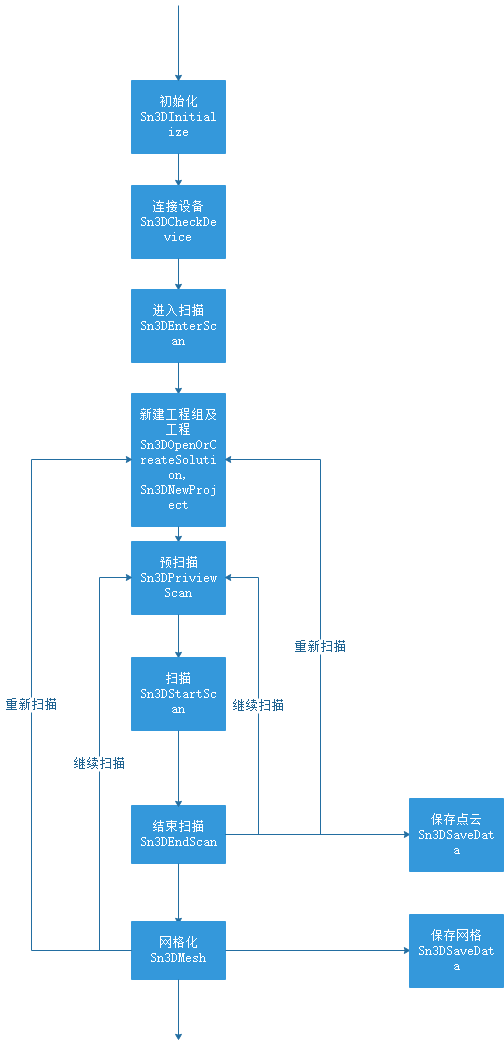
**Parameter**

**wholeMarker**

**[out] 当前标志点数据**

**owner**

# 主要流程示意



# 接口定义

### 基本功能接口

#### 初始化SDK

Sn3DInitialize

初始化SDK

|  |
| --- |
| **void\* Sn3DInitialize**(  Sn3DScanServiceWatctherCallBack watcher  ); |

**Parameters**

Sn3DScanServiceWatctherCallBack

[in] 注册服务端监控回调，服务端退出会收到通知

**Return Values**

参考返回值定义宏

**Remarks**

1 函数的功能为初始化SDK。

2 初始化一次就可以了

3 如果不在使用必须调用**Sn3DRelease**释放。

#### 释放SDK

**Sn3DRelease**

释放SDK

|  |
| --- |
| **int Sn3DRelease**(); |

**Return Values**

参考返回值定义宏

**Remarks**

1 函数的功能为释放SDK资源。

#### 设置SDK输出日志

**Sn3DInitialLog**

|  |
| --- |
| **int Sn3DInitialLog** (  char\* exeName,  char\* path  ); |

**Parameters**

exeName

[in] 调用sdk进程的名称

path

[in] 设置日志所在目录

**Return Values**

参考错误码定义

#### 保存数据

**Sn3DSaveData**

|  |
| --- |
| **int Sn3DSaveData** (  **char\* absolutePrjName,**  **char\*savePath,**  **char\*saveType**  ); |

**Parameters**

**absolutePrjName**

[in] 工程文件的完整路径

**savePath**

[in] 需要保存的文件的路径和名称（不带后缀）

**saveType**

[in] 需要保存的文件的格式(asc ply stl obj)

**Return Values**

参考错误码定义

### 设备控制接口

#### 连接设备

**Sn3DCheckDevice**

|  |
| --- |
| **int Sn3DConnectDevice** () |

**Return Values**

参考返回值定义宏

#### 获取相机亮度取值范围

**Sn3DGetBrightnessRange**

|  |
| --- |
| **SN3DSDKE10API int Sn3DGetBrightnessRange(int& min, int& max);** |

**Parameters**

min

[out] 相机亮度最小等级，默认为0保留参数

max

[out] 相机亮度最大等级

**Return Values**

参考错误码定义

**Remarks**

初始化后调用该函数

#### 设置相机亮度

**Sn3DSetBrightness**

|  |
| --- |
| **int Sn3DSetBrightness** (  int brightness  ) |

**Parameters**

brightness

[in] 相机亮度等级，取值范围由[S](#_获取相机亮度取值范围)**[n3DGetBrightnessRange](#_获取相机亮度取值范围)**返回的值决定，值越大越亮。

**Return Values**

参考错误码定义

**Remarks**

#### 获取视频流

**Sn3DIntallGetImagasCallBack**

|  |
| --- |
| **int Sn3DIntallGetImagasCallBack** (  **Sn3DGetImagesCallBack imageCallback,**  **void\* owner,**  ); |

**Parameters**

**imageCallback**

[in] 设置回调函数

**owner**

[in] 回调函数的拥有者的指针

#### 当前帧点云接口

**Sn3DSetCurrentPointCloudCallBack**

|  |
| --- |
| **int Sn3DSetCurrentPointCloudCallBack** (  **Sn3DCurrentPointCloudCallBack callBackFunc,**  **void\* owner,**  ); |

**Parameters**

**CallbackFunc**

[in] 设置回调函数

**owner**

[in] 回调函数的拥有者的指针

#### 设置增量点云数据回调接口

**Sn3DSetIncreasePointCloudCallBack**

|  |
| --- |
| **int Sn3DSetIncreasePointCloudCallBack**(  **Sn3DIncreasePointCloudCallBack callBackFunc,**  **void\* owner,**  ); |

**Parameters**

**CallbackFunc**

[in] 设置回调函数

**owner**

[in] 回调函数的拥有者的指针

#### 设置相机视角数据回调接口

**Sn3DSetCameraPositionCallBack**

|  |
| --- |
| **int Sn3DSetCameraPositionCallBack**(  **Sn3DCameraPositionCallBack callBackFunc,**  **void\* owner,**  ); |

**Parameters**

**CallbackFunc**

[in] 设置回调函数

**owner**

[in] 回调函数的拥有者的指针

#### 设置跟踪丢失状态数据回调接口

**Sn3DSetTrackLostStatusCallBack**

|  |
| --- |
| **int Sn3DSetTrackLostStatusCallBack**(  **Sn3DTrackLostStatusCallBack callBackFunc,**  **void\* owner,**  ); |

**Parameters**

**CallbackFunc**

[in] 设置回调函数

**owner**

[in] 回调函数的拥有者的指针

#### 设置扫描距离挡位回调接口

**Sn3DSetScanDistCallBack**

|  |
| --- |
| **int Sn3DSetScanDistCallBack**(  **Sn3DScanDistCallBack callBackFunc,**  **void\* owner,**  ); |

**Parameters**

**CallbackFunc**

[in] 设置回调函数

**owner**

[in] 回调函数的拥有者的指针

#### 设置设备操作回调接口

**Sn3DSetDeviceEventCallBack**

|  |
| --- |
| **int Sn3DSetDeviceEventCallBack**(  **Sn3DDeviceEventCallBack callBackFunc,**  **void\* owner,**  ); |

**Parameters**

**CallbackFunc**

[in] 设置回调函数

**owner**

[in] 回调函数的拥有者的指针

**Return Values**

参考错误码定义

**Remarks**

1. 已经过滤longpress\_plus和longpress\_sub两功能。
2. 双击OK键为切换模式，初始模式为相机视角模式，双击OK键后切换为其他模式(自定义)，再双击后切换会相机视角模式。

#### 设置工作距离范围接口

**Sn3DSetWorkRange**

|  |
| --- |
| **int Sn3DSetWorkRange(float min, float max);** |

**Parameters**

**min**

[in] 工作距离范围的最小值，默认160

**max**

[in] 工作距离范围的最大值，默认600

**Return Values**

参考错误码定义

**Remarks**

1. 设置的工作距离范围在区间[160, 600]内，并且最大值与最小值的差不得小于200即max - min >= 200
2. 必须在进入扫描后，取消扫描前调用

#### 获取当前亮度接口

**Sn3DGetCurrentBrightness**

|  |
| --- |
| **int Sn3DGetCurrentBrightness(int& currentBrightness)** |

**Parameters**

**currentBrightness**

[out] 当前亮度值

**Return Values**

参考错误码定义

**Remarks**

#### 平面检测接口

**Sn3DSetTooFlatStatusCallBack**

|  |
| --- |
| **int Sn3DSetTooFlatStatusCallBack(Sn3DTooFlatStatusCallBack callBackFunc, void\* owner)；** |

**Parameters**

**CallbackFunc**

[in] 设置回调函数

**owner**

[in] 回调函数的拥有者的指针

#### 当前标志点接口

**Sn3DSetCurrentMarkerCallBack**

|  |
| --- |
| **int Sn3DSetCurrentMarkerCallBack(Sn3DCurrentMarkerCallBack callBackFunc, void\* owner)；** |

**Parameters**

**CallbackFunc**

[in] 设置回调函数

**owner**

[in] 回调函数的拥有者的指针

#### 全量标志点接口

**Sn3DSetWholeMarkerCallBack**

|  |
| --- |
| **int Sn3DSetWholeMarkerCallBack(Sn3DWholeMarkerCallBack callBackFunc, void\* owner)；** |

**Parameters**

**CallbackFunc**

[in] 设置回调函数

**owner**

[in] 回调函数的拥有者

### 扫描功能接口

#### 进入扫描

Sn3DEnterScan

|  |
| --- |
| **SN3DSDKE10API int CALLMETHOD Sn3DEnterScan();** |

**Parameters**

**Return Values**

错误代码 参考返回值宏定义

**Remarks**

1 此函数只有在初始化检测设备在线后才能被调用

2 执行扫描初始化相关

#### 新建解决方案

Sn3DOpenOrCreateSolution

|  |
| --- |
| **SN3DSDKE10API int Sn3DOpenOrCreateSolution(**  **const char\* slnDirPath,**  **bool isCreate,**  **int scanMode,**  **bool hasTexture,**  **float pointDis);** |

**Parameters**

slnDirPath

[in] 解决方案路径

isCreate

[in] true新建, false打开

scanMode

[in] 扫描模式, 0 portrait, 1 object, 2 smallobject. Reference: ScanMode

hasTexture

[in] true纹理扫描, false不带纹理

pointDis

[in] 点距

**Return Values**

错误代码 参考返回值宏定义

**Remarks**

1 Sn3DEnterScan后调用

#### 新建工程

Sn3DNewProject

|  |
| --- |
| **SN3DSDKE10API int CALLMETHOD Sn3DNewProject(**  **const char\* slnDirPath,**  **int scanMode,**  **bool hasTexture,**  **float pointDis,**  **int alignType,**  **const char\* globalMarkerPath);** |

**Parameters**

slnDirPath

[in] 解决方案路径

scanMode

[in] 扫描模式, 0 portrait, 1 object, 2 smallobject. Reference: ScanMode

hasTexture

[in] true纹理扫描, false不带纹理

pointDis

[in] 点距

alignType

[in] 拼接模式. Reference: AlignType 见附录

globalMarkerPath

[in] 框架点路径, 非框架点扫描模式设置为空

**Return Values**

错误代码 参考返回值宏定义

**Remarks**

1 Sn3DOpenOrCreateSolution后调用

#### 打开工程

Sn3DOpenProject

|  |
| --- |
| **SN3DSDKE10API int CALLMETHOD Sn3DOpenProject(**  **const char\* projFileName);** |

**Parameters**

projFileName

[in] 工程路径

**Return Values**

错误代码 参考返回值宏定义

**Remarks**

1 Sn3DOpenOrCreateSolution后调用

#### 预扫描

Sn3DPreviewScan

|  |
| --- |
| **SN3DSDKE10API int CALLMETHOD Sn3DPreviewScan();** |

**Parameters**

**Return Values**

错误代码 参考返回值宏定义

**Remarks**

1. 预扫描只重建数据, 跟踪不融合到整体点云上.
2. 需要在新建或打开工程后调用

#### 扫描

Sn3DStartScan

|  |
| --- |
| **SN3DSDKE10API int CALLMETHOD Sn3DStartScan();** |

**Parameters**

**Return Values**

错误代码 参考返回值宏定义

**Remarks**

1. 需要预扫描Sn3DPreviewScan之后调用
2. 重建, 跟踪, 融合到整体点云

#### 暂停扫描

Sn3DPauseScan

|  |
| --- |
| **SN3DSDKE10API int CALLMETHOD Sn3DPauseScan ();** |

**Parameters**

**Return Values**

错误代码 参考返回值宏定义

**Remarks**

1. 必须在扫描Sn3DStartScan之后调用

#### 结束扫描

Sn3DEndScan

|  |
| --- |
| **SN3DSDKE10API int CALLMETHOD Sn3DEndScan(**  **bool globalOptimize,**  **double pointDist,**  **Sn3DWholePointCloudCallBack callback = nullptr);** |

**Parameters**

globalOptimize

[in] true 优化, false 不优化

pointDist

[in]点距设置

callback

[in]设置回调, 可以拿结束扫描后的点云数据

**Return Values**

错误代码 参考返回值宏定义

**Remarks**

1. 可以在Sn3DStartScan或者Sn3DPauseScan后调用;
2. 生成整体点云数据;
3. 如果设置回调可以拿到该阶段生成点云数据, demo中有示例;

#### 清空扫描数据

Sn3DCleanScan

|  |
| --- |
| **SN3DSDKE10API int Sn3DCancelScan(**  **bool isCancelCurrentProjectFramerMark);** |

**Parameters**

isCancelCurrentProjectFramerMark

[in] true 清空框架点信息

**Return Values**

错误代码 参考返回值宏定义

**Remarks**

1. 清空当前已扫描的点云数据

#### 退出扫描

Sn3DExitScan

|  |
| --- |
| **SN3DSDKE10API int CALLMETHOD Sn3DExitScan();** |

**Parameters**

**Return Values**

错误代码 参考返回值宏定义

**Remarks**

1. 退出扫描

#### 开启伪彩色

**Sn3DSetEnablePseudoColor**

|  |
| --- |
| **int Sn3DSetEnablePseudoColor**(  bool enable  ); |

**Parameters**

enable

[in] true开启，false关闭

**Return Values**

参考错误码定义

### 封装功能接口

#### 生成网格数据

Sn3DMesh

|  |
| --- |
| **SN3DSDKE10API int CALLMETHOD Sn3DMesh(**  **int meshType,**  **int filterLevel,**  **int SmoothLevel,**  **float PointDis,**  **bool fillSmallHole,**  **double smellHolePerimeter,**  **bool fillMarkerHole,**  **double spikeSensitivity,**  **int faceLimit,**  **Sn3DWholePointCloudCallBack callback = nullptr);** |

**Parameters**

meshType.

[in] Reference: MeshType

filterLevel.

[in] 0-3档, 不同的滤波力度

SmoothLevel.

[in] 1-3档, 在滤波基础上设置平滑力

PointDis.

[in]点距

fillSmallHole.

[in] true填充小洞

smellHolePerimeter.

[in]填充小洞周长

fillMarkerHole.

[in] true填充标志点

spikeSensitivity.

[in] 去除尖刺

faceLimit.

[in] 最大面片限制

callback.

[in] 设置回调, 取网格化数据后的点云数据

**Return Values**

参考错误码定义

**Remarks**

1. **需要在结束扫描Sn3DEndScan后调用**
2. **网格化数据**
3. **设置callback可以拿到网格化后的整体点云集合**

#### 生成网格数据

Sn3DMeshEx

|  |
| --- |
| **SN3DSDKE10API int CALLMETHOD Sn3DMesh(**  **int meshType,**  **int filterLevel,**  **int SmoothLevel,**  **float PointDis,**  **bool fillSmallHole,**  **double smellHolePerimeter,**  **bool fillMarkerHole,**  **double spikeSensitivity,**  **int faceLimit,**  **void\* owner**  **Sn3DMeshDataCallBack callback = nullptr);** |

**Parameters**

meshType.

[in] Reference: MeshType

filterLevel.

[in] 0-3档, 不同的滤波力度

SmoothLevel.

[in] 1-3档, 在滤波基础上设置平滑力

PointDis.

[in]点距

fillSmallHole.

[in] true填充小洞

smellHolePerimeter.

[in]填充小洞周长

fillMarkerHole.

[in] true填充标志点

spikeSensitivity.

[in] 去除尖刺

faceLimit.

[in] 最大面片限制

owner

[in] 回调函数拥有着的指针

callback.

[in] 设置回调, 取网格化数据后的点云数据

**Return Values**

参考错误码定义

**Remarks**

1.**需要在结束扫描Sn3DEndScan后调用**

2.**网格化数据**

3.**设置callback可以拿到网格化后的整体点云集合**