

SDK 开发指南

武汉极动智能科技有限公司

二〇二三年七月

SDK 开发指南

编 制: 李慧珠

审 核: 曾 浩

标准化审查: 曾 浩

会 签: 曾 浩

批 准: 曾 浩

文档修订记录

文档版本号	修订说明	程序版本号	修订作者	修订日期
V1.0.1	适配新版本 SDK	V3.4.2.11	李慧珠	2023-06-01
V1.0.2	适配新版本 SDK	V3.4.2.13	李慧珠	2023-07-10

SDK 版本修订

SDK 版本号	修订说明	文档版本号	修订作者	修订日期
V3.4.2.11	更新版本	V1.0.1	baimax	2023-05-11
V3.4.2.13	新增算法功能	V1.0.2	baimax	2023-06-08

版权声明

版权所有武汉极动智能科技有限公司，保留所有权利。

保密声明

本文档（包括任何附件）包含的信息是保密信息。接收人了解其获得的本文档是保密的，除用于规定的目的外不得用于任何目的，也不得将本文档泄露给任何第三方。

目 录

1. 文档与 SDK 概述.....	1
1.1. 文档概述.....	1
1.2. SDK 概述.....	1
1.3. TOF 相机开图流程.....	2
2. 支持设备.....	2
3. XDYN_STREAMER_SDK.....	3
3.1. 基础参数类型.....	3
3.1.1. 枚举类型.....	3
3.1.2. 结构体类型.....	10
3.2. xdyn_memory.....	18
3.2.1. 枚举类型.....	18
3.2.2. 结构体类型.....	19
3.2.3. API.....	20
3.3. xdyn_streamer.....	22
3.3.1. XDYN_Streamer 成员解析.....	22
3.3.2. API.....	38
3.4. xdyn_utils.....	41
3.4.1. API 总体简要解析.....	41
3.4.2. API 详解.....	42
3.5. xdyn_rgbd.....	44
3.5.1. 数据类型.....	44
3.5.2. API.....	56
4. Demo 示例.....	63
4.1. 图像输出配置.....	63

4.2. 异常处理	65
4.3. 图像输出	65

SDK 开发指南

1. 文档与 SDK 概述

1.1. 文档概述

本文档供 Windows 系统下 C++-Debug/Release 开发使用, 支持 64 位/32 位 windows10 和 Windows11 系统, 适用于 X-D400、X-D500, X-D1000 设备。

表 1. 文档结构表

章节	标题	内容
1	概述	介绍 SDK 概况
2	支持设备	介绍支持产品
3	SDK 说明	介绍 SDK 接口及使用
4	SDK Demo 示例	部分 SDK Demo 展示

1.2. SDK 概述

SDK 是用于控制 TOF 相机的一个独立组件, 支持获取实时图像数据、配置参数、图像后处理等功能, 同时支持 USB 和 GigE 两种接口。目前支持 Windows、Linux 系统, 不同的操作系统下需要加载使用配套的 SDK 版本。SDK 对相机的控制上最基本的功能分为参数设置和图像获取两部分:

- 参数设置: 可以通过使用通用配置接口对相机积分时间、Binning 模式、Phase 模式、频率等参数进行设置, 并获取相机 ID、图像格式、图像大小、温度等参数。
- 图像获取: 当相机发送图像数据到 PC 时, 图像数据会存放在 SDK 内部, 本文档介绍 SDK 以回调的方式获取图像。使用不同的配置, 可获取原始数据、深度图、灰度图和点云图。

1.3. TOF 相机开图流程

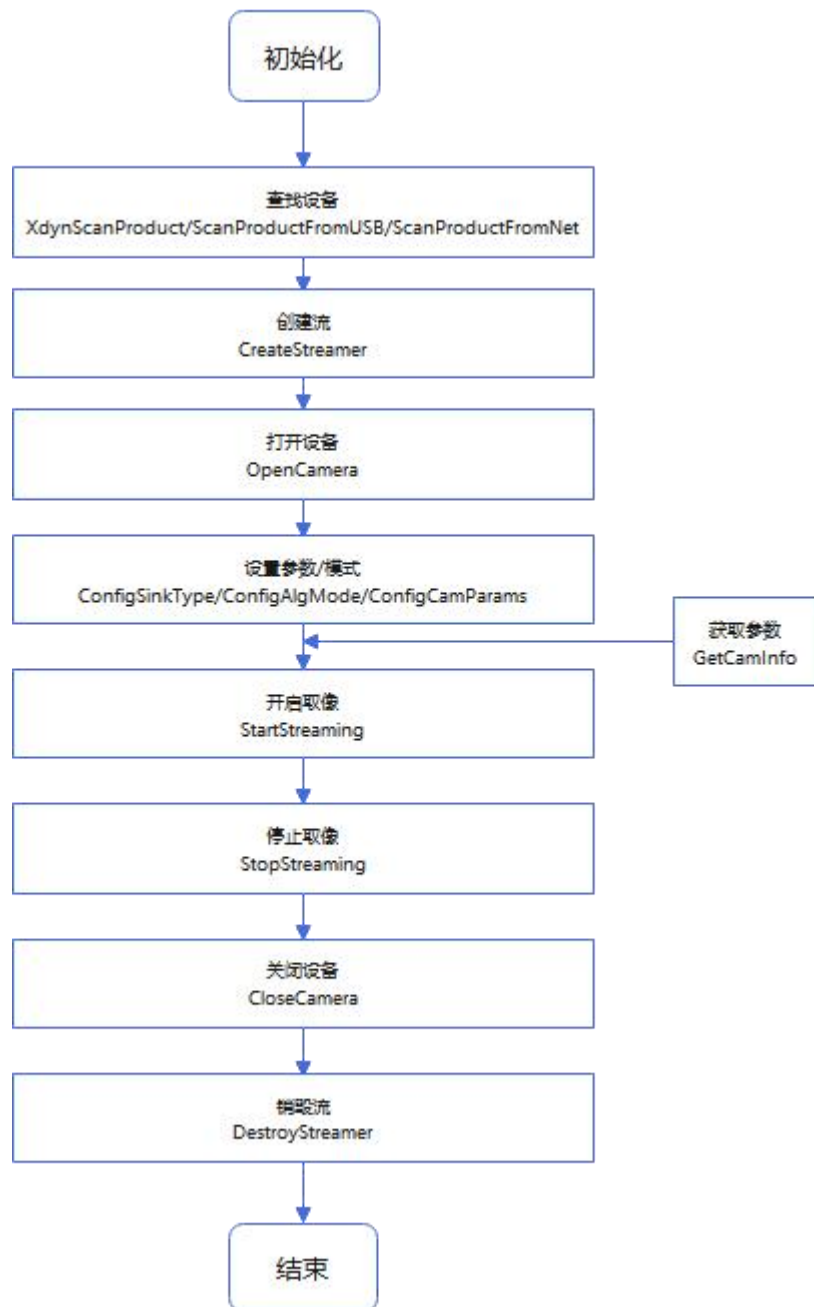


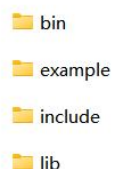
图 1. tof 相机开图流程图

2. 支持设备

当前 SDK 支持的设备包含 X-D400、X-D500、X-D1000。

3.XDYN_STREAMER_SDK

SDK 包含 bin, example, lib 等目录。目录结构如下图所示：



- bin 包含 SDK 二进制文件；
- example 包含已支持产品诸多例程，可进行深度图、灰度图、点云图等图像的数据输出；
- include 包含 SDK 通用头文件；
- lib 包含 SDK 导出的库文件；

3.1. 基础参数类型

3.1.1. 枚举类型

1) XDYN_PRODUCT_TYPE_e

【描述】

产品型号

【参数】

参数	值	描述
XDYN_PRODUCT_TYPE_XD_500	0u	X-D500
XDYN_PRODUCT_TYPE_XD_1000	1u	X-D1000
XDYN_PRODUCT_TYPE_XD_300	0x100u	X-D300
XDYN_PRODUCT_TYPE_XD_500_LH	0x201u	X-D500-LH
XDYN_PRODUCT_TYPE_XD_500_XB	0x202u	X-D500-XB

XDYN_PRODUCT_TYPE_XD_500_SJ	0x203u	X-D500-SJ
XDYN_PRODUCT_TYPE_XD_500_STD	0x204u	X-D500-STD
XDYN_PRODUCT_TYPE_XD_1001	0x401u	X-D1001
XDYN_PRODUCT_TYPE_XD_1002	0x402u	X-D1002
XDYN_PRODUCT_TYPE_XD_1003	0x403u	X-D1003
XDYN_PRODUCT_TYPE_XD_1100	0x464u	X-D1100
XDYN_PRODUCT_TYPE_XD_1101	0x465u	X-D1101
XDYN_PRODUCT_TYPE_XD_1102	0x466u	X-D1102
XDYN_PRODUCT_TYPE_XD_300_MASK	0x100u	X-D300 掩码
XDYN_PRODUCT_TYPE_XD_500_MASK	0x200u	X-D500 掩码
XDYN_PRODUCT_TYPE_XD_1000_MASK	0x400u	X-D1000 掩码
XDYN_PRODUCT_TYPE_INVALID	0xFFFFFFFFFu	--

2) XDYN_DEV_TYPE_e

【描述】

产品相机类型

【参数】

参数	值	描述
XDYN_DEV_TYPE_TOF	0	TOF 相机
XDYN_DEV_TYPE_RGB	1	RGB 相机
XDYN_DEV_TYPE_TOF_RGB	2	TOF + RGB 相机
XDYN_DEV_TYPE_VDRIVER	3	--
XDYN_DEV_TYPE_ISP1	4	--

XDYN_DEV_TYPE_ISP2	5	--
XDYN_DEV_TYPE_TOTAL_NUM	6	--

3) XDYN_DEV_TYPE_e

【描述】

输出数据流方式

【参数】

参数	值	描述
XDYN_SINK_TYPE_NONE	0x00000000	--
XDYN_SINK_TYPE_UVC	0x00000001	USB 连接
XDYN_SINK_TYPE_SHM	0x00000002	共享内存，用于设备使用时有效
XDYN_SINK_TYPE_NET	0x00000004	网口连接
XDYN_SINK_TYPE_CB	0x00000008	回调异步输出
XDYN_SINK_TYPE_CYPRESS	0x00000010	X-D300 的输出方式

4) XDYN_ALG_MODE_E

【描述】

输出算法模式

【参数】

参数	值	描述
XDYN_ALG_MODE_EMB_ALG_IPC_PASS	0	下位机运行算法
XDYN_ALG_MODE_EMB_PASS_IPC_ALG	1	上位机运行算法
XDYN_ALG_MODE_EMB_CORR_IPC_POST	2	--
XDYN_ALG_MODE_EMB_DEPTH_IPC_ALG	3	--

XDYN_ALG_MODE_EMB_DEBUG_IPC_ALG	4	--
XDYN_ALG_MODE_EMB_PASS_IPC_PASS	5	--
XDYN_ALG_MODE_INVALID	0xFFFFFFFF	--

5) XDYN_BINNING_MODE_e

【描述】

图像 binning 模式

【参数】

参数	值	描述
XDYN_BINNING_MODE_NONE	0	Bin11(640 * 480)
XDYN_BINNING_MODE_ANALOG	1	行 binning
XDYN_BINNING_MODE_DIGITAL	2	列 binning
XDYN_BINNING_MODE_2x2	3	Bin22(320 * 240)
XDYN_BINNING_MODE_4x4	4	Bin44(180 * 120)
XDYN_BINNING_MODE_TOTAL_NUM	5	--

6) XDYN_MIRROR_MODE_e

【描述】

图像镜像模式

【参数】

参数	值	描述
XDYN_MIRROR_MODE_NONE	0	--
XDYN_MIRROR_MODE_VERTICAL	1	垂直镜像
XDYN_MIRROR_MODE_HORIZONTAL	2	水平镜像

XDYN_MIRROR_MODE_BOTH	3	垂直+水平镜像
XDYN_MIRROR_MODE_TOTAL_NUM	4	--

7) XDYN_T_SENSOR_MODE_e

【描述】

Sensor 温度模式

【参数】

参数	值	描述
XDYN_TEMP_SENSOR_MODE_NONE	0	--
XDYN_TEMP_SENSOR_MODE_EVERYPHASE	1	每个 Phase 都附带温度信息
XDYN_TEMP_SENSOR_MODE_EVERYSUBFRAME	2	每个子帧都附带温度信息
XDYN_TEMP_SENSOR_MODE_TOTAL_NUM	3	--

8) XDYN_PHASE_MODE_e

【描述】

Phase 模式

【参数】

参数	值	描述
XDYN_PHASE_MODE_1	0	1Phase
XDYN_PHASE_MODE_2	1	2Phase
XDYN_PHASE_MODE_4	2	4Phase
XDYN_PHASE_MODE_8	3	8Phase
XDYN_PHASE_MODE_16	4	16Phase

XDYN_PHASE_MODE_TOTAL_NUM	5	--
---------------------------	---	----

9) XDYN_FSM_STATE_e

【描述】

相机状态信息

【参数】

参数	值	描述
XDYN_S_IDLE	0	Idle 状态
XDYN_S_CONNECTED	1	连接相机
XDYN_S_STREAMING	2	相机开流
XDYN_S_DISCONNECTED	3	断开连接
XDYN_S_TOTAL_NUM	4	--

10) XDYN_PP_DENOISE_LEVEL

【描述】

后处理算法去噪等级

【参数】

参数	值	描述
XDYN_PP_DENOISE_LEVEL_0	0	去噪等级 - 0 级/不去噪
XDYN_PP_DENOISE_LEVEL_1	1	去噪等级 - 1 级
XDYN_PP_DENOISE_LEVEL_2	2	去噪等级 - 2 级
XDYN_PP_DENOISE_LEVEL_3	3	去噪等级 - 3 级
XDYN_PP_DENOISE_LEVEL_4	4	去噪等级 - 4 级
XDYN_PP_DENOISE_LEVEL_5	5	去噪等级 - 5 级

11) XDYN_PP_TDENOISE_METHOD

【描述】

时域去噪方法

【参数】

参数	值	描述
XDYN_PP_TDENOISE_METHOD_MF	0	时域去噪方法 - MF
XDYN_PP_TDENOISE_METHOD_FIR	1	时域去噪方法 - FIR

12) XDYN_PP_SDENOISE_METHOD

【描述】

空域去噪方法

【参数】

参数	值	描述
XDYN_PP_SDENOISE_METHOD_MF	0	空域去噪方法 - MF
XDYN_PP_SDENOISE_METHOD_BF	1	空域去噪方法 - FIR
XDYN_PP_SDENOISE_METHOD_NLM	2	空域去噪方法 - NLM

13) XdRes

【描述】

通常错误反馈码，用于程序错误甄别。

【参数】

参数	值	描述
XD_SUCCESS	0	运行成功

XD_ERR	-1	运行失败
XD_INVALIDRES	-2	无效返回值
XD_TIMEOUT	-3	连接超时（无法收到设备相应信息，就会返回连接超时）
XD_UNINIT	-4	设备未初始化

15) XDYN_CB_EVENT_E

【描述】

事件信息反馈

【参数】

参数	值	描述
XDYN_CB_EVENT_INVALIDIB	0x00000000	无效事件
XDYN_CB_EVENT_CAM_INFO	0x00000001	cam 基础信息，在配置 camera 的时候返回
XDYN_CB_EVENT_STM_STATUS	0x00000002	设备当前状态信息
XDYN_CB_EVENT_ERROR	0x00010001	设备发生错误
XDYN_CB_EVENT_DRIVER_HOT	0x00010002	设备断开连接

3.1.2. 结构体类型

1) XdynVersion_t

【描述】

软件版本号

【参数】

参数	类型	描述
type	uint16_t	软件类型, 1-SDK 版本, 4-相机固件版本
major	uint16_t	主版本
minor	uint16_t	次版本
revision	uint16_t	修订版本
temp	uint16_t	临时版本

2) FrameResolution

【描述】

帧分辨率信息

【参数】

参数	类型	描述
width	uint16_t	图像长度
height	uint16_t	图像宽度
stride	uint16_t	跨距
fmt	uint32_t	图像格式

3) XdynFrameInfo_t

【描述】

单帧图像信息

【参数】

参数	类型	描述
width	uint16_t	图像长度
height	uint16_t	图像宽度

format	uint16_t	图像格式
index	uint16_t	预留
rx_ts	uint32_t	芯片开始获取图像数据时间戳
tx_us	uint32_t	芯片开始输出图像数据时间戳
temp0	uint16_t	芯片温度
temp1	uint16_t	预留
timestamp	uint64_t	图像输出时数据时间戳
reserved[18]	uint32_t	预留

4) XdynDepthFrameInfo_t

【描述】

深度图像信息

【参数】

参数	类型	描述
frameInfo	XdynFrameInfo_t	图像信息
eOutDepthValueType	uint32_t	深度值类型
fUnitOfDepth	float	整型深度图的量化精度
fModFreqMHZ[4]	float	深度图对应调制频率
fLensParas[5]	float	镜头参数

【说明】

当深度图的数据类型为 UINT16 时，每个像素值代表一个单位为 LSB 的整型深度值，整型深度值乘以量化精度 fUnitOfDepth 后，可将单位转化为 mm。当深度图的数据类型为 DEPTH16 时，每个像素值代表一个 DEPTH16 结构体数值，每个 LSB 对应

1mm, 即 $fUnitOfDepth = 1\text{mm}/\text{LSB}$ 。

当点云图由外部软件自行通过深度图计算时使用镜头参数。若镜头标定参数加载失败, 则返回的镜头参数为全 0。其中, $fLensParas[0]$ -x 焦距, $[1]$ -焦距, $[2]$ -光心对应的图像 x 坐标, $[3]$ -y 光心对应的图像 y 坐标, $[4]$ -镜头校正的视场缩放系数: 小于 1-放大(裁黑边), 等于 1-不缩放, 大于 1-缩小。

5) XdynConfidenceInfo_t

【描述】

图像置信度信息。当置信度图的数据类型为 `UINT8` 时, 每个像素的置信度值为 0~255, 分别对应置信度为 $0.f$ (置信度低)~ $1.f$ (置信度高)。每个置信度值乘以 $fUnitOfConfidence$ 后, 可换算至 $0.f \sim 1.f$ 之间。

【参数】

参数	类型	描述
<code>frameInfo</code>	<code>XdynFrameInfo_t</code>	置信度图参数
<code>fConfidenceInt2Float</code>	<code>float</code>	置信度从整型向浮点型的转换单位

6) XdynFrame_t

【描述】

图像数据

【参数】

参数	类型	描述
<code>addr</code>	<code>void *</code>	图像数据指针
<code>size</code>	<code>uint32_t</code>	图像数据实际大小
<code>ex</code>	<code>void *</code>	图像的扩展参数指针

【说明】

对于不同的数据类型，数据格式说明如下：

数据	信息	格式	备注
Depth	格式	Uint16, 单位 LSB	实际的深度值=LSB× frameInfo->fUnitOfDepth
	大小	W * H * 2	--
	扩展参数	XdynDepthFrameInfo_t, 详见类型 说明	--
PointCloud	格式	[x y z confidence], 类型: float, 数据并排存放	--
	大小	W * H * 16	--
	扩展参数	NULL	--
Gray	格式	Uint16, 单位 LSB	实际灰度值=LSB
	大小	W * H * 2	--
	扩展参数	NULL	--
RGB	格式	NV12	--
	大小	W * H * 1.5	--
	扩展参数	NULL	--
Amp	格式	Uint16, 单位 LSB	--
	大小	W * H * 2	--
	扩展参数	NULL	--

Confidence	格式	Uint8	--
	大小	W * H	--
	扩展参数	XdynConfidenceInfo_t, 详见类型说明	--

7) XdynCamTInt_t

【描述】

相机积分时间

【参数】

参数	类型	描述
PHASE1_4TInt	uint32_t	相位 1-4 积分时间（单位：纳秒）
PHASE5_8TInt	uint32_t	相位 5-8 积分时间（单位：纳秒）
specialPhaseTInt	uint32_t	特殊相位积分时间（单位：纳秒）

8) XdynRes_t

【描述】

RGB 参数信息

【参数】

参数	类型	描述
width	uint16_t	图像宽度
height	uint16_t	图像盖度
stride	uint16_t	跨距
fmt	uint32_t	图像格式
fps	uint16_t	图像帧率

9) XdynCamInfo_t

【描述】

相机参数信息

【参数】

参数	类型	描述
info	MemSinkInfo	图像配置参数
tofChipID	uint32_t	相机 ID
tofUID[2]	uint32_t	[0] walfid [1] sensor ID
tSensor	int32_t	相机 Sensor 温度
tDriver	int32_t	相机 Driver 温度
tInt[4]	XdynCamTInt_t	相机积分时间

10) XdynStmStatus_t

【描述】

开流后相机状态

【参数】

参数	类型	描述
state	uint32_t	相机状态
tSesor	uint32_t	相机 Sensor 状态
tDriver	uint32_t	相机 Driver 状态
reserve[10]	uint32_t	预留

11) XdynReg_t

【描述】

寄存器地址和寄存器值

【参数】

参数	类型	描述
addr	uint32_t	寄存器地址
val	uint32_t	寄存器值

12) XdynLensParams_t

【描述】

相机镜头参数

【参数】

参数	类型	描述
fLenParas[9]	float	镜头校正参数[fx, fy, cx, cy, k1, k2, p1, p2, k3]
fZoomCoef	float	镜头校正后图像的缩放系数：<1-放大（裁黑边），1-不缩放，>1-缩小
fSkew	float	描述传感器轴间的倾斜，由传感器的安装没有与光轴垂直引起

13) XdynROIDef_t

【描述】

ROI 区域

【参数】

参数	类型	描述
xStart	uint16_t	RO 区域 x 起始位置（值从 0 开始）
xSizeOrg	uint16_t	x 原始长度

xStep	uint16_t	x 步长 (1-32)
yStart	uint16_t	起始位置 (值从 0 开始, 当使用模拟装箱时, yStart 必须是偶数)
ySizeOrg	uint16_t	y 原始长度
yStep	uint16_t	y 步长 (1-32, 当使用模拟装仓时, yStep 必须是偶数)

14) XdynRegParams_t

【描述】

RGBD 配准内外参矩阵

【参数】

参数	类型	描述
fTofIntrinsicMatrix[9]	float	TOF 相机的内参矩阵
fRgbIntrinsicMatrix[9]	float	RGB 相机的内参矩阵
fRotationMatrix[9]	float	右相机相对左相机的旋转矩阵
fTranslationMatrix[3]	float	右相机相对左相机的平移矩阵
bIsRgbCameraLeft	bool	RGB 是否在左侧

3.2. xdyn_memory

3.2.1. 枚举类型

1) MEM_SINK_DATA_TYPE

【描述】

图像数据类型存储

【参数】

参数	描述
MEM_AGENT_SINK_RAW	原始数据(UINT16)
MEM_AGENT_SINK_DEPTH	深度图数据(UINT16)
MEM_AGENT_SINK_POINT_CLOUD	点云图数据(fx (float) 、 fy (float) , fz(float)、 confidence(float))
MEM_AGENT_SINK_AMP	幅值图数据
MEM_AGENT_SINK_FLAG	标记图
MEM_AGENT_SINK_CONFID	置信度数据
MEM_AGENT_SINK_GRAY	灰度图数据(BYT8)
MEM_AGENT_SINK_BG	背景图数据
MEM_AGENT_SINK_RGB	RGB 数据(NV12)
MEM_AGENT_SINK_USR1	--
MEM_AGENT_SINK_USR2	--
MEM_AGENT_SINK_USR3	--
MEM_AGENT_SINK_USR4	--
MEM_AGENT_SINK_USR5	--
MEM_AGENT_SINK_MAX	--

3.2.2. 结构体类型

1) MemSinkInfo

【描述】

图像配置参数

【参数】

参数	类型	描述
width	unsigned int	图像宽度

height	unsigned int	图像高度
numPhases	unsigned int	图像 Phase 模式
addInfoLines	unsigned int	附加行
numInputFrames	unsigned int	输入帧数量
binningMode	unsigned int	图像 Binning 模式
rgbW	unsigned int	RGB 图像宽度
rgbH	unsigned int	RGB 图像高度
stride	unsigned int	跨距
fmt	unsigned int	图像格式

3.2.3. API

API 总体简要解析

功能	函数	类型
计算数据内存 size	CalcDataMemSize	unsigned int
计算图像 size	CalcImgSize	unsigned int
计算总内存	CalcTotalMemSize	unsigned int

API 详解如下：

1) CalcDataMemSize

函数原型

```
unsigned int CalcDataMemSize(MEM_SINK_DATA_TYPE type, MemSinkInfo info)
```

函数功能

计算对应数据类型需要配置内存大小

函数参数

参数	类型	功能
type	MEM_SINK_DATA_TYPE	图像数据类型
info	MemSinkInfo	图像配置参数 (Info, 输出信息, 包括图像长宽等)

返回值

bufLen: 返回计算后的数据长度

2) CalcImgSize

函数原型

```
unsigned int CalcImgSize(MEM_SINK_DATA_TYPE type, MemSinkInfo info)
```

函数功能

计算对应图像类型事件的图像尺寸大小

函数参数

参数	类型	功能
type	MEM_SINK_DATA_TYPE	图像数据类型
info	MemSinkInfo	图像配置参数 (Info, 输出信息, 包括图像长宽等)

返回值

bufLen: 返回计算后的数据长度

3) CalcTotalMemSize

函数原型

```
unsigned int CalcTotalMemSize(MemSinkCfg shmCfg, MemSinkInfo info)
```

函数功能

计算 sinkcfg 中所有 enable 类型的内存大小总和

函数参数

参数	类型	功能
shmCfg	MemSinkCfg	输出图像信息配置
info	MemSinkInfo	图像配置参数 (Info, 输出信息, 包括图像长宽等)

返回值

shmSize: 返回计算后的长度

3.3. xdyn_streamer

3.3.1. XDYN_Streamer 成员解析

3.3.1.1. 类总体简要解析

该类主要功能为对相机开机, 相机配置, 参数配置, 图像配置等进行操作, 其包含如下成员:

功能	函数	类型
获取版本	GetVersion	void
相机开关操作	OpenCamera	int
	CloseCamera	int
设置相机输出类型	ConfigSink	int
设置输出图像配置	ConfigSinkType	int
	ConfigSinkType	int
设置算法模式	ConfigAlgMode	int
设置 TOF 相机参数	SetWorkMode	void
	SetCamFreq	void
	SetCamInt	void
	SetFps	void
	SetCamBinning	void
	SetCamMirror	void
	SetPhaseMode	void

	SetROI	void
	ConfigCamParams	int
	GetCamInfo	XdynCamInfo_t
	GetCamInfo	void
获取图像输出配置	GetResolution	void
配置 RGB 参数	RgbSetRes	void
	CfgRgbParams	int
配置算法参数	ConfigCorrParam	void
	Corr_SetPreDist	void
	ConfigCorrParam	int
	PP_SetDepthDenoise	void
	PP_SetGrayDenoise	void
	PP_SetDeFlyPixel	void
	ConfigPPParam	int
	GetCaliRegParams	int
	GetRgbLensParams	int
流配置	StartStreaming	int
	StopStreaming	int
读写寄存器以及 DLL 配置	SetDLL	int
	WriteReg	int
	ReadReg	int

3.3.1.2. 类成员详解

1) GetVersion

函数原型

```
void GetVersion(uint16_t type, XdynVersion_t &version)
```

函数功能

获取相机版本号

函数参数

参数	类型	功能
type	uint16_t	版本对象类型，获取相机固件版本：type = 4, SDK

		版本号：type = 1
version	XdynVersion_t	版本号

返回值

NULL

2) OpenCamera

函数原型

```
int OpenCamera(XDYN_DEV_TYPE_e devType)
```

函数功能

开启相机

函数参数

参数	类型	功能
devType	XDYN_DEV_TYPE_e	相机类型，TOF 相机设置： devType = XDYN_DEV_TYPE_e::XDYN_DEV_TYPE_TOF

返回值

成功返回 XD_SUCCESS，失败返回对应错误码。

3) CloseCamera

函数原型

```
int CloseCamera()
```

函数功能

关闭相机

函数参数

NULL

返回值

成功返回 XD_SUCCESS，失败返回对应错误码。

4) ConfigSink

函数原型

```
int ConfigSink(XDYN_SINK_TYPE_E sinkType)
```

函数功能

配置相机数据输出类型

函数参数

参数	类型	功能
sinkType	XDYN_SINK_TYPE_E	输出数据流方式

返回值

成功返回 XD_SUCCESS，失败返回对应错误码。

5) ConfigSinkType

函数原型

```
int ConfigSinkType(uint32_t type, MemSinkCfg cfg)
```

```
int ConfigSinkType(uint32_t type, MemSinkCfg cfg, XdynStreamCB streamCB,  
void *stmHandle)
```

函数功能

设置输出图像配置

函数参数

参数	类型	功能
type	uint32_t	输出数据流方式
cfg	MemSinkCfg	设置需要获取的图像类型

streamCB	XdynStreamCB	回调函数
stmHandle	void *	数据流句柄

返回值

成功返回 XD_SUCCESS，失败返回对应错误码。

6) ConfigAlgMode

函数原型

```
int ConfigAlgMode(uint32_t Mode)
```

函数功能

设置算法模式

函数参数

参数	类型	功能
Mode	uint32_t	设置图像输出时的模式

返回值

成功返回 XD_SUCCESS，失败返回对应错误码。

7) SetWorkMode

函数原型

```
void SetWorkMode(uint32_t mode, uint32_t subMode)
```

函数功能

设置相机工作模式，其主要有单频和双频工作模式，其中单、双频中又分为是否带灰度和背景帧。

函数参数

参数	类型	功能
----	----	----

mode	uint32_t	设置相机的单频（SF）双频（DF）模式
subMode	uint32_t	设置单双频中是否带灰度帧或背景帧

返回值

NULL

8) SetCamFreq

函数原型

void SetCamFreq(uint32_t freq1, uint32_t freq2)

函数功能

设置相机频率，默认设置双频。

函数参数

参数	类型	功能
freq1	uint32_t	高频，TOF X-D500 默认值：99，TOF X-D1000 默认值：62
freq2	uint32_t	低频，TOF X-D500 默认值：18，TOF X-D1000 默认值：25

返回值

NULL

9) SetCamInt

函数原型

void SetCamInt(unsigned int *phaseInt, unsigned int *specialInt)

函数功能

设置相机积分时间，时间单位为 ns。

函数参数

参数	类型	功能
phaseInt	unsigned int *	N1、N2 积分时间，

		设置方式：unsigned int phaseInt[4] = {1000000, 1000000, 0, 0}
specialInt	unsigned int *	灰度、背景积分时间, 设置方式：unsigned int specialInt[4] = {1000000, 1000000, 0, 0}

返回值

NULL

10) SetFps

函数原型

void SetFps(unsigned int fps)

函数功能

设置相机帧率

函数参数

参数	类型	功能
fps	unsigned int	帧率

返回值

NULL

11) SetCamBinning

函数原型

void SetCamBinning(XDYN_BINNING_MODE_e mode)

函数功能

设置 binning 模式

函数参数

参数	类型	功能
mode	XDYN_BINNING_MODE_e	binning 模式：XDYN_BINNING_MODE_NONE 和

		XDYN_BINNING_MODE_2x2, XDYN_BINNING_MODE_NONE 分辨率 640 * 480; XDYN_BINNING_MODE_2x2 分辨率 320 * 240
--	--	--

返回值

NULL

12) SetCamMirror

函数原型

void SetCamMirror(XDYN_MIRROR_MODE_e mode)

函数功能

设置相机镜像模式

函数参数

参数	类型	功能
mode	XDYN_MIRROR_MODE_e	镜像模式，分为垂直和水平镜像

返回值

NULL

13) SetPhaseMode

函数原型

void SetPhaseMode(XDYN_PHASE_MODE_e mode)

函数功能

设置相机 Phase 模式

函数参数

参数	类型	功能
mode	XDYN_PHASE_MODE_e	Phase 模式，分为 1Phase,2Phase,4Phase,8Phase,16Phase

返回值

NULL

14) SetROI

函数原型

void SetROI(XdynROIDef_t roi)

函数功能

设置相机 ROI 区域

函数参数

参数	类型	功能
roi	XdynROIDef_t	绘制 ROI 区域的基本参数

返回值

NULL

15) ConfigCamParams

函数原型

int ConfigCamParams()

函数功能

配置相机参数，以上相机参数配置完成后，必须调用该函数才能生效。

函数参数

NULL

返回值

成功返回 XD_SUCCESS，失败返回对应错误码。

16) GetCamInfo

函数原型

XdynCamInfo_t GetCamInfo()

void GetResolution(MemSinkInfo &info)

函数功能

获取相机参数信息

函数参数

参数	类型	功能
info	MemSinkInfo	图像长宽，芯片 ID，温度等参数

返回值

NULL

17) GetResolution

函数原型

```
void GetResolution(MemSinkInfo &info)
```

函数功能

获取图像配置信息

函数参数

参数	类型	功能
info	MemSinkInfo	输出信息，输出图像长宽，binning 和 Phase 模式

返回值

NULL

18) RgbSetRes

函数原型

```
void RgbSetRes(XdynRes_t res)
```

函数功能

RGB 参数配置

函数参数

参数	类型	功能
res	XdynRes_t	RGB 图像长宽，帧率等信息

返回值

NULL

19) CfgRgbParams

函数原型

int CfgRgbParams()

函数功能

配置 RGB 参数

函数参数

NULL

返回值

成功返回 XD_SUCCESS，失败返回对应错误码。

20) Corr_SetAE

函数原型

void Corr_SetAE(uint8_t enable)

函数功能

设置相机自动曝光

函数参数

参数	类型	功能
enable	uint8_t	自动曝光使能设置

返回值

NULL

21) Corr_SetPreDist

函数原型

```
void Corr_SetPreDist(uint16_t dist)
```

函数功能

配置相机物体检测的预设距离

函数参数

参数	类型	功能
dist	uint16_t	相机预设距离设置

返回值

NULL

22) ConfigCorrParam

函数原型

```
int ConfigCorrParam()
```

函数功能

配置相机的校准算法参数

函数参数

NULL

返回值

成功返回 XD_SUCCESS，失败返回对应错误码。

23) PP_SetDepthDenoise

函数原型

```
Void PP_SetDepthDenoise(XDYN_PP_TDENOISE_METHOD tDenoiseMethod,
XDYN_PP_DENOISE_LEVEL tDenoiseLevel, XDYN_PP_SDENOISE_METHOD
sDenoiseMethod, XDYN_PP_DENOISE_LEVEL sDenoiseLevel)
```

函数功能

后处理算法中深度图去噪设置

函数参数

参数	类型	功能
tDenoiseMethod	XDYN_PP_TDENOISE_METHOD	深度图时域去噪方法
tDenoiseLevel	XDYN_PP_DENOISE_LEVEL	深度图时域去噪等级
sDenoiseMethod	XDYN_PP_SDENOISE_METHOD	深度图空域去噪方法
sDenoiseLevel	XDYN_PP_DENOISE_LEVEL	深度图空域去噪等级

返回值

NULL

24) PP_SetGrayDenoise

函数原型

```
void PP_SetGrayDenoise(XDYN_PP_TDENOISE_METHOD tDenoiseMethod,
XDYN_PP_DENOISE_LEVEL tDenoiseLevel, XDYN_PP_SDENOISE_METHOD
sDenoiseMethod, XDYN_PP_DENOISE_LEVEL sDenoiseLevel)
```

函数功能

后处理算法中灰度图去噪设置

函数参数

参数	类型	功能
tDenoiseMethod	XDYN_PP_TDENOISE_METHOD	灰度图时域去噪方法
tDenoiseLevel	XDYN_PP_DENOISE_LEVEL	灰度图时域去噪等级
sDenoiseMethod	XDYN_PP_SDENOISE_METHOD	灰度图空域去噪方法
sDenoiseLevel	XDYN_PP_DENOISE_LEVEL	灰度图空域去噪等级

返回值

成功返回 XD_SUCCESS，失败返回对应错误码。

25) PP_SetDeFlyPixel

函数原型

```
void PP_SetDeFlyPixel(uint8_t level)
```

函数功能

后处理中飞点修复设置

函数参数

参数	类型	功能
level	uint8_t	后处理设置中飞点修复等级

返回值

NULL

26) ConfigPPPParam

函数原型

```
int ConfigPPPParam()
```

函数功能

配置相机的后处理算法参数

函数参数

NULL

返回值

成功返回 XD_SUCCESS，失败返回对应错误码。

27) GetCaliRegParams

函数原型

```
int GetCaliRegParams(XdynRegParams_t &regParams)
```

函数功能

配置相机的校准算法参数

函数参数

参数	类型	功能
regParams	XdynRegParams_t	返回相机矩阵内参

返回值

成功返回 XD_SUCCESS，失败返回对应错误码。

28) GetRgbLensParams

函数原型

```
int GetRgbLensParams(XdynLensParams_t &rgbLensParams)
```

函数功能

获取 RGB 镜头校正参数，校正后图像的缩放系数等

函数参数

参数	类型	功能
rgbLensParams	XdynLensParams_t	返回 RGB 镜头校正参数

返回值

成功返回 XD_SUCCESS，失败返回对应错误码。

29) StartStreaming

函数原型

```
int StartStreaming()
```

函数功能

开流

函数参数

NULL

返回值

成功返回 XD_SUCCESS，失败返回对应错误码。

30) StopStreaming

函数原型

int StopStreaming()

函数功能

断流

函数参数

NULL

返回值

成功返回 XD_SUCCESS，失败返回对应错误码。

31) SetDLL

函数原型

int SetDLL(int step)

函数功能

配置相机的寄存器，主要用于标定中向移

函数参数

参数	类型	功能
step	int	步幅

返回值

成功返回 XD_SUCCESS，失败返回对应错误码。

32) WriteReg

函数原型

int WriteReg(XDYN_DEV_TYPE_e type, XdynReg_t *regLists, int size)

函数功能

向寄存器中对应的地址写值

函数参数

参数	类型	功能
type	XDYN_DEV_TYPE_e	产品类型
regLists	XdynReg_t *	寄存器列表，包含地址和值
size	int	大小

返回值

成功返回 XD_SUCCESS，失败返回对应错误码。

33) ReadReg

函数原型

```
int ReadReg(XDYN_DEV_TYPE_e type, XdynReg_t *regLists, int size)
```

函数功能

读取寄存器对应地址中的内容

函数参数

参数	类型	功能
type	XDYN_DEV_TYPE_e	产品类型
regLists	XdynReg_t *	寄存器列表，包含地址和值
size	int	大小

返回值

成功返回 XD_SUCCESS，失败返回对应错误码。

3.3.2. API

3.3.2.1. API 总体简要解析

功能	函数	类型
----	----	----

创建流指针	CreateStreamer	XDYN_Streamer *
创建流指针 (Net)	CreateStreamerNet	XDYN_Streamer *
销毁流	DestroyStreamer	void
销毁流	XdynContextInit	void
取消初始化	XdynContextUninit	void

3.3.2.2. API 详解

1) CreateStreamer

函数原型

XDYN_LIB_API XDYN_Streamer *CreateStreamer(XDYN_PRODUCT_TYPE_e ptType, XDYN_SINK_TYPE_E ptInter, XdynEventCB evCB, void *evHandle, std::string ip = "192.168.31.3")

函数功能

创建流指针

函数参数

参数	类型	功能
ptType	XDYN_PRODUCT_TYPE_e	相机型号, 即 prtList[index].type
ptInter	XDYN_SINK_TYPE_E	相机接口, 即 prtList[index].inter1 或者 prtList[index].Inter2
evCB	XdynEventCB	回调事件函数
evHandle	void *	事件句柄
ip	std::string	相机 IP 地址

返回值

流指针

2) CreateStreamerNet

函数原型

XDYN_LIB_API XDYN_Streamer

*CreateStreamerNet(XDYN_PRODUCT_TYPE_e ptType, XdynEventCB evCB, void *evHandle, std::string ip)

函数功能

创建流指针

函数参数

参数	类型	功能
ptType	XDYN_PRODUCT_TYPE_e	相机型号，即 prtList[index].type
evCB	XdynEventCB	回调事件函数
evHandle	void *	事件句柄
ip	std::string	相机 IP 地址

返回值

流指针

3) DestroyStreamer

函数原型

XDYN_LIB_API void DestroyStreamer(XDYN_Streamer *xdynStream)

函数功能

销毁流

函数参数

参数	类型	功能
xdynStream	XDYN_Streamer *	流指针

返回值

NULL

4) XdynContextInit

函数原型

XDYN_LIB_API void XdynContextInit()

函数功能

Context 初始化，包含日志、相机、算法默认初始化。

函数参数

NULL

返回值

NULL

5) XdynContextUninit

函数原型

XDYN_LIB_API void XdynContextUninit()

函数功能

取消初始化，包含清理 zlogAPI 申请的内存。

函数参数

NULL

返回值

NULL

3.4. xdyn_utils

3.4.1. API 总体简要解析

功能	函数	类型
枚举 USB 接口连接相机	ScanProductFromUSB	XDYN_LIB_API void
枚举网口连接相机	ScanProductFromNet	XDYN_LIB_API void

枚举所有连接相机	XdynScanProduct	XDYN_LIB_API void
获取相机参数	XdynGetProductStr	XDYN_LIB_API std::string

3.4.2. API 详解

1) ScanProductFromUSB

函数原型

XDYN_LIB_API void ScanProductFromUSB(std::vector<XdynProductDesc_t> &prtList)

函数功能

枚举 USB 接口连接的相机

函数参数

参数	类型	功能
prtList	std::vector<XdynProductDesc_t>	相机列表

返回值

NULL

2) ScanProductFromNet

函数原型

XDYN_LIB_API void ScanProductFromNet(std::vector<XdynProductDesc_t> &prtList, std::string ip)

函数功能

枚举网口连接的相机

函数参数

参数	类型	功能
prtList	std::vector<XdynProductDesc_t>	相机列表

ip	std::string	相机 IP
----	-------------	-------

返回值

NULL

3) XdynScanProduct

函数原型

XDYN_LIB_API void XdynScanProduct(std::vector<XdynProductDesc_t> &prtList, std::string ip = "192.168.31.3")

函数功能

枚举所有连接的相机，如果通过网口连接，则相机 IP 默认为 192.168.31.3。

函数参数

参数	类型	功能
prtList	std::vector<XdynProductDesc_t>	相机列表
ip	std::string	相机 IP: 192.168.31.3

返回值

NULL

4) XdynGetProductStr

函数原型

XDYN_LIB_API std::string XdynGetProductStr(uint32_t type)

函数功能

获取相机参数

函数参数

参数	类型	功能
type	uint32_t	相机类型

返回值

返回获取到的相机参数

3.5. xdyn_rgbd

3.5.1. 数据类型

3.5.1.1. 宏定义类型

参数	值	描述
TRUE	1	正确值
FALSE	0	错误值
RP_INTRINSIC_MATRIX_LEN	9	RP 模块内参矩阵长度
RP_ROTATION_MATRIX_LEN	9	RP 模块旋转矩阵长度
RP_TRANSLATION_MATRIX_LEN	9	RP 模块平移矩阵长度
RP_ARITH_VERSION_LEN_MAX	64	算法版本号字符数组最大长度

3.5.1.2. 枚举类型

1) DEPTH_VALUE_TYPE

【描述】

深度值类型

【参数】

参数	值	描述
DEPTH_VALUE_XYR	0	斜距
DEPTH_VALUE_XYZ	1	空间坐标 (X, Y, Z) 中的 Z 值

2) DEPTH_DATA_TYPE

【描述】

深度数据类型

【参数】

参数	值	描述
DEPTH_TYPE_UINT16	0	unsigned short (16bit 深度值)
DEPTH_TYPE_DEPTH16	1	DEPTH16 (详见 DEPTH16 结构体定义)

3) POINTCLOUD_DATA_TYPE

【描述】

点云数据类型

【参数】

参数	值	描述
PC_TYPE_PC32F	0	PC32F (详见 PC32F 结构体定义)
PC_TYPE_DEPTH_POINT_CLOUD	1	DEPTH_POINT_CLOUD (详见 DEPTH_POINT_CLOUD 结构体定义)
PC_TYPE_RGBD_POINT_CLOUD	2	RGBD_POINT_CLOUD (详见 RGBD_POINT_CLOUD 结构体定义)

4) OUT_IMAGE_NAME

【描述】

输出图像的名称

【参数】

参数	值	描述
OIMG_NAME_POINTCLOUD	0	点云图
OIMG_NAME_DEPTH	1	深度图
OIMG_NAME_AMP	2	幅度图

OIMG_NAME_GRAY	3	灰度图
OIMG_NAME_BG	4	背景图
OIMG_NAME_FLAG	5	标记图
OIMG_NAME_CONFIDENCE	6	置信度图
OIMG_NAME_REG_RGB	7	配准后 RGB 图
OIMG_NAME_RGB	8	原始的 RGB 图
OIMG_NAME_RP_RGB2DEPTH	9	可见光图像映射到深度图像
OIMG_NAME_RP_DEPTH2RGB	10	深度图像映射到可见光图像
OIMG_NAME_RP_RGBD_POINT_CLOUD	11	RGB 配准后的点云图

5) OUT_IMAGE_DATA_TYPE

【描述】

输出图像的数据类型

【参数】

参数	值	描述
OIMG_DTYPE_UINT8	0	unsigned char
OIMG_DTYPE_UINT16	1	unsigned short
OIMG_DTYPE_FLOAT32	2	float
OIMG_DTYPE_DEPTH16	3	DEPTH16
OIMG_DTYPE_PC32F	4	PC32F
OIMG_DTYPE_DEPTH_POINT_CLOUD	5	DEPTH_POINT_CLOUD
OIMG_DTYPE_FLAG	6	CORR_FLAG_PIXEL
OIMG_DTYPE_RGB	7	RGB

6) OUT_IMAGE_BUFFER_TYPE

【描述】

输出图像 buffer 分配方式

【参数】

参数	值	描述
OIMG_BUFFER_NULL	0	无，即算法不输出图像
OIMG_BUFFER_USER	1	由用户分配
OIMG_BUFFER_INNER	2	有算法内部分配

7) RP_ARITH_INIT_SUCCESS_FLAG

【描述】

定义算法初始化成功/失败标识，无符号 32 位（unsigned int），按 bit 位保存异常标识。

【参数】

参数	值	描述
RP_INIT_SUCCESS	0x00	初始化成功
RP_INIT_FAIL_GLB_BUFFER_SIZE_CALC_ERROR	0x01 << 0	buffer 空间大小计算异常
RP_INIT_FAIL_GLB_BUFFER_ALLOCATE	0x01 << 1	全局 buffer 空间分配失败
RP_INIT_FAIL_INVALID_FIXPARA_POINTER	0x01 << 2	无效的输入固定配置参数指针 (NULL == pstFixParas)
RP_INIT_FAIL_INVALID_PARAM_IN_DEPTH_WIDTH	0x01 << 5	无效的输入参数: usInDepthWidth
RP_INIT_FAIL_INVALID_PARAM_IN_DEPTH_HEIGHT	0x01 << 6	无效的输入参数: usInDepthHeight

RP_INIT_FAIL_INVALID_PARA_IN_RGB_WIDTH	0x01 << 7	无效的输入参数: usInRGBWidth
RP_INIT_FAIL_INVALID_PARA_IN_RGB_HEIGHT	0x01 << 8	无效的输入参数: usInRGBHeight
RP_INIT_FAIL_INVALID_PARA_OUT_IMG_WIDTH	0x01 << 19	无效的输入参数: usOutImgWidth
RP_INIT_FAIL_INVALID_PARA_OUT_IMG_HEIGHT	0x01 << 20	无效的输入参数: usOutImgHeight
RP_INIT_FAIL_MODULE_INIT_PRECALC	0x01 << 22	模块算法初始化和 提前计算失败

8) RP_ARITH_SUCCESS_FLAG

【描述】

定义算法执行成功/失败标识，无符号 32 位（unsigned int），按 bit 位保存异常标识。

【参数】

参数	值	描述
RP_ARITH_SUCCESS	0x00	算法执行成功
RP_ARITH_FAIL_INVALID_OUTER_BUFFER	0x01 << 0	无效的外部 buffer 指针
RP_ARITH_FAIL_GLB_BUFFER_ALLOCATE	0x01 << 1	全局 buffer 空间分配失败
RP_ARITH_FAIL_OUTER_BUFFER_CHANGED	0x01 << 2	外部 buffer 地址发生变化
RP_ARITH_FAIL_INVALID_	0x01 << 3	无效的 internal buffer 指针:

GLB_FIXVAR_BUFFER		GLB_FIXVAR
RP_ARITH_FAIL_NOT_INITIALIZED	0x01 << 4	未初始化
RP_ARITH_FAIL_INVALID_INOUTDATAS_POINTER	0x01 << 5	无效的算法输入数据结构体指针 (NULL == pstInOutDatas)
RP_ARITH_FAIL_CALC_POINTCLOUD	0x01 << 6	点云计算失败
RP_ARITH_FAIL_ALLOCATE_DYNAMIC_BUFFER	0x01 << 7	动态 buffer 空间分配失败

9) RP_SUCCESS_FLAG

【描述】

定义 RP 算法执行成功/失败标识，无符号 32 位 (unsigned int)，按 bit 位保存异常标识。

【参数】

参数	值	描述
RP_SUCCESS	0x00	算法执行成功
RP_FAIL_INVALID_IN_DEPTH_IMAGE	0x01 << 0	无效的输入深度图像指针
RP_FAIL_INVALID_IN_YUV_IMAGE	0x01 << 1	无效的输入 YUV 图像指针
RP_FAIL_INVALID_OUT_RGB_IMAGE	0x01 << 2	无效的算法输出 RGB 图指针
RP_FAIL_INVALID_INPARAM	0x01 << 3	无效的算法输入参数指针
RP_FAIL_INVALID_OUTPARAM	0x01 << 4	无效的算法输出参数指针
RP_FAIL_INVALID_GLOBAL_BUFFER	0x01 << 5	无效的模块全局 buffer

RP_FAIL_INVALID_GLB_VARS_POINTER	0x01 << 6	无效的全局变量结构体指针
RP_FAIL_INVALID_IN_RGB_IMG_POINTER	0x01 << 7	无效的输入 RGB 图像指针
RP_FAIL_INVALID_DEPTH_TO_RGB_MAP	0x01 << 8	无效的深度到 RGB 映射图
RP_FAIL_INVALID_INSIZE	0x01 << 9	无效的输入图像尺寸
RP_FAIL_INVALID_TOF_INTRINSIC_PARAS	0x01 << 10	无效的 Tof 相机内参矩阵
RP_FAIL_INVALID_CALIPARA_POINTER	0x01 << 11	无效的标定参数指针
RP_FAIL_INVALID_INCALIPARA	0x01 << 12	无效的输入标定参数指针
RP_FAIL_INVALID_CALI_FIRSTBYTEID	0x01 << 13	无效的 REG 模块起始字节数
RP_FAIL_INVALID_CALI_MODID	0x01 << 14	无效的标定文件模块 ID
RP_FAIL_INVALID_CALI_VALIDFLAG	0x01 << 15	无效的 REG 模块参数有效标识
RP_FAIL_DISABLE_REG_FLAG	0x01 << 16	配准功能未使能标识
RP_FAIL_CALIPARA_LOAD_FAIL	0x01 << 17	未成功加载标定参数
RP_FAIL_INVALID_CALI_CORR_PIXEL_MODE	0x01 << 18	无效的 REG 标定和校正像素匹配类型
RP_FAIL_INVALID_CORR_MAP_POINTER	0x01 << 19	无效的坐标映射表指针

10) RP_ARITH_ABNORMAL_FLAG

【描述】

定义 RP 算法异常标识，无符号 32 位（unsigned int），按 bit 位保存异常标识。

【参数】

参数	值	描述
RP_ARITH_NORMAL	0x00	正常
RP_ARITH_ABNORMAL_INVALID_DYNPARA_POINTER	0x01 << 0	无效的输入动态配置参数 指针 (NULL == pstDynParas)
RP_ARITH_ABNORMAL_INVALID_DYNPARA_IN_TOF_WIDTH	0x01 << 1	无效的输入动态配置参数-inTOF Width
RP_ARITH_ABNORMAL_INVALID_DYNPARA_IN_TOF_HEIGHT	0x01 << 2	无效的输入动态配置参数-inTOF Height
RP_ARITH_ABNORMAL_INVALID_DYNPARA_IN_RGB_WIDTH	0x01 << 3	无效的输入动态配置参数-inRGB Width
RP_ARITH_ABNORMAL_INVALID_DYNPARA_IN_RGB_HEIGHT	0x01 << 4	无效的输入动态配置参数-inRGB Height
RP_ARITH_ABNORMAL_INVALID_DYNPARA_OUT_WIDTH	0x01 << 5	无效的输入动态配置参数-out Width
RP_ARITH_ABNORMAL_INVALID_DYNPARA_OUT_HEIGHT	0x01 << 6	无效的输入动态配置参数-out Height
RP_ARITH_ABNORMAL_INVALID_INOUTDATAS_DEPTH	0x01 << 7	无效的算法输入数据-pstDepth
RP_ARITH_ABNORMAL_INVALID_INOUTDATAS_RGB	0x01 << 8	无效的算法输入数据-pstRGB
RP_ARITH_ABNORMAL_FAIL_MALLOC	0x01 << 9	临时空间动态分配失败

3.5.1.3. 结构体类型

1) DEPTH16

【描述】

定义深度数据类型

【参数】

参数	类型	描述
usDepth	unsigned short	深度数据类型
usConfidence	unsigned short	置信度数据类型

2) UNION_DEPTH_DATA_TYPE

【描述】

定义深度图的像素类型

【参数】

参数	类型	描述
usDepth	unsigned short	深度图像素类型
stDepth	DEPTH16	深度数据

3) PC32F

【描述】

定义点云数据类型

【参数】

参数	类型	描述
fX	float	X 坐标
fY	float	Y 坐标

fZ	float	Z 坐标
----	-------	------

4) DEPTH_POINT_CLOUD

【描述】

定义深度-点云数据类型

【参数】

参数	类型	描述
fX	float	X 坐标
fY	float	Y 坐标
fZ	float	Z 坐标
fConfidence	float	置信度 Range: 0~1. (0-confidence 0%, 1-confidence 100%)

5) RGBD_POINT_CLOUD

【描述】

定义 RGB-点云数据类型

【参数】

参数	类型	描述
fX	float	X 坐标
fY	float	Y 坐标
fZ	float	Z 坐标
r	unsigned char	红色
g	unsigned char	绿色

b	unsigned char	蓝色
---	---------------	----

6) RP_INPARAS

【描述】

定义 RP 算法输入参数

【参数】

参数	类型	描述
pThisGlbBuffer	void*	外部分配好的当前模块的全局空间 buffer 首地址
pucYuvImg	unsigned char*	输入的 YUV 图像指针
usYuvWidth	unsigned short	输入图像宽度 (yuv 图)
usYuvHeight	unsigned short	输入图像高度 (yuv 图)
pusDepth	unsigned short*	输入的深度图指针
usDepthWidth	unsigned short	输入图像宽度 (深度图)
usDepthHeight	unsigned short	输入图像高度 (深度图)
pucConfidence	unsigned char*	输入的置信度图像
pucConfidence	unsigned short	输入图像宽度 (置信度图)
usConfHeight	unsigned short	输入图像高度 (置信度图)

7) RP_DYNPARA

【描述】

定义动态配置参数

【参数】

参数	类型	描述
----	----	----

usInDepthWidth	unsigned short	输入的 ToF 宽
usInDepthHeight	unsigned short	输入的 ToF 高
usInYuvWidth	unsigned short	输入的 RGB 宽
usInYuvHeight	unsigned short	输入的 RGB 高
usOutR2DWidth	unsigned short	输出的图像宽
usOutR2DHeight	unsigned short	输出的图像高
usOutD2RWidth	unsigned short	输出的图像宽
usOutD2RHeight	unsigned short	输出的图像高
uiOutRGBDLen	unsigned int	输出 RGB 图像数据长度
ucEnableOutR2D	unsigned char	是否输出 RGB2Depth 图像
ucEnableOutD2R	unsigned char	是否输出 Depth2RGB 图像
ucEnableRGBDPCL	unsigned char	是否输出 RGBD 点云图
ucThConfidence	unsigned char	置信度滤波的参数阈值

8) CALIPARAS_RP

【描述】

定义 RP 模块标定参数

【参数】

参数	类型	描述
fTofIntrinsicMatrix[RP_INTRINSIC_MATRIX_LEN]	float	Tof 相机的内参矩阵
fRgbIntrinsicMatrix[RP_INTRINSIC_MATRIX_LEN]	float	Rgb 相机的内参矩阵
fRotationMatrix[RP_ROTATION_MATRIX_LEN]	float	右相机相对左相机的旋转矩阵

fTranslationMatrix[RP_TRANSLATION_MATRIX_LEN]	float	右相机相对左相机的平移矩阵
bIsRgbCameraLeft	unsigned char	Rgb 相对 Tof 是否在左边, 以人正对传感器表面来定义左右

9) RP_OUTPARAS

【描述】

定义 RP 算法输出图像

【参数】

参数	类型	描述
pucRgb2DepthImg	unsigned char*	输出 RGB 图像映射到深度图像指针
usOutR2DWidth	unsigned short	输出 R2D 的图像宽
usOutR2DHeight	unsigned short	输出 R2D 的图像高
ucEnableOutR2D	unsigned char	是否输出 R2D 的图像
pusDepth2RgbImg	unsigned short*	输出输出深度图像映射到可见光图像指针
usOutD2RWidth	unsigned short	输出 D2R 的图像宽
usOutD2RHeight	unsigned short	输出 D2R 的图像高
ucEnableOutD2R	unsigned char	是否输出 D2R 的图像
pstrRGBD	RGBD_POINT_CLOUD*	输出 RGBD 的点云数图像
uiOutRGBDLen	unsigned int	输出 RGBD 点云的数量
ucEnableRGBDPCL	unsigned char	是否输出 RGBD 点云

3.5.2. API

3. 5. 2. 1. API 总体简要解析

功能	函数	类型
算法初始化	sitrpInit	void*
设置动态配置参数	sitrpSetDynamicConfig	void
执行 TOF 和 RGB 的 sensor 配准算法	sitrpRunRGBProcess	void
释放算法 buffer	sitrpRelease	void
获取算法版本号	sitrpGetVersion	void
获取 TOF 标定参数	sitrpSetTofIntrinsicMat	unsigned char
获取 RGB 标定参数	sitrpSetRgbIntrinsicMat	unsigned char
获取 TOF 和 RGB 的平移参数	sitrpSetTranslationMat	unsigned char
获取 TOF 和 RGB 的旋转参数	sitrpSetRotationMat	unsigned char
获取 TOF 和 RGB 的相对未知参数	sitrpSetRgbPos	unsigned char

3.5.2.2. API 详解

1) sitrpInit

函数原型

```
void* sitrpInit(unsigned int* puiInitSuccFlag, CALIPARAS_RP* pstCaliPara,
RP_DYNPARA* pstCaliPara, unsigned char ucEnableLogFile, unsigned char
ucEnableCostTimeLogFile)
```

函数功能

初始化算法

函数参数

参数	类型	描述
puiInitSuccFlag	unsigned int*	算法初始化成败标识
pstCaliPara	CALIPARAS_RP*	RP 模块标定参数

pstCaliPara	RP_DYNPARA*	RP 模块设置动态参数
ucEnableLogFile	unsigned char	是否生成初始化 log 文件开关
ucEnableCostTimeLogFile	unsigned char	是否生成耗时统计 log 文件开关

返回值

输出算法是否初始化成功

2) sitrpSetDynamicConfig

函数原型

```
void sitrpSetDynamicConfig(void* pGlbBuffer, RP_DYNPARA* pstDynParas,
unsigned int* puiSuccFlag, unsigned int* puiAbnormalFlag, unsigned char
ucEnableLogFile)
```

函数功能

设置动态配置参数

函数参数

参数	类型	描述
pGlbBuffer	void*	全局空间 buffer 指针
pstDynParas	RP_DYNPARA*	动态配置参数结构体指针
puiSuccFlag	unsigned int*	算法成败标识
puiAbnormalFlag	unsigned int*	算法执行异常标识
ucEnableLogFile	unsigned char	是否生成动态配置 log 文件开关

返回值

NULL

3) sitrpRunRGBProcess

函数原型

```
void sitrpRunRGBProcess(void* pGlbBuffer, RP_INPARAS* pstInDatas,
```

RP_OUTPARAS* pstOutDatas, unsigned int* puiSuccFlag, unsigned int* puiAbnormalFlag, unsigned char ucEnableLogFile)

函数功能

执行 TOF 和 RGB 的 sensor 配准算法

函数参数

参数	类型	描述
pGlbBuffer	void*	全局空间的 buffer 指针
pstInDatas	RP_INPARAS*	算法输入数据结构体指针
pstOutDatas	RP_OUTPARAS*	算法输出数据结构体指针
puiSuccFlag	unsigned int*	算法成败标识
puiAbnormalFlag	unsigned int*	算法执行异常标识
ucEnableLogFile	unsigned char	是否生成动态配置 log 文件开关

返回值

NULL

4) sitrpRelease

函数原型

void sitrpRelease(void** pGlbBuffer, unsigned char ucEnableLogFile)

函数功能

释放算法 buffer

函数参数

参数	类型	描述
pGlbBuffer	void**	全局空间 buffer 指针
ucEnableLogFile	unsigned char	是否生成 log 文件开关

返回值

NULL

5) sitrpGetVersion

函数原型

void sitrpGetVersion(char* pcDllVersion)

函数功能

获取算法版本号

函数参数

参数	类型	描述
pcDllVersion	char*	算法版本号字符串指针

返回值

NULL

6) sitrpSetTofIntrinsicMat

函数原型

```
unsigned char sitrpSetTofIntrinsicMat(float* pfOutBuffer, float* pfInitri, int iInitriLen, unsigned int* puiSuccFlag, unsigned char ucEnableLogFile)
```

函数功能

获取 TOF 标定参数

函数参数

参数	类型	描述
pfOutBuffer	float*	输出内参指针
pfInitri	float*	TOF 内参指针
iInitriLen	int	TOF 内参数组长度
puiSuccFlag	unsigned int*	输出内参指针

ucEnableLogFile	unsigned char	是否生成 kog 文件开关
-----------------	---------------	---------------

返回值

返回函数是否成功标识

7) sitrpSetRgbIntrinsicMat

函数原型

```
unsigned char sitrpSetRgbIntrinsicMat(float* pfOutBuffer, float* pfInitri, int iInitriLen, unsigned int* puiSuccFlag, unsigned char ucEnableLogFile)
```

函数功能

获取 RGB 标定参数

函数参数

参数	类型	描述
pfOutBuffer	float*	输出内参指针
pfInitri	float*	RGB 的内参指针
iInitriLen	int	RGB 的内参数组长度
puiSuccFlag	unsigned int*	算法成败标识
ucEnableLogFile	unsigned char	是否生成 log 文件开关

返回值

返回函数成败标识

8) sitrpSetTranslationMat

函数原型

```
unsigned char sitrpSetTranslationMat(float* pfOutBuffer, float* pfInitri, int iInitriLen, unsigned int* puiSuccFlag, unsigned char ucEnableLogFile)
```

函数功能

获取 TOF 和 RGB 的平移参数

函数参数

参数	类型	描述
pfOutBuffer	float*	输出内参指针
pfInitri	float*	RGB 的内参指针
iInitriLen	int	RGB 的内参数组长度
puiSuccFlag	unsigned int*	算法成败标识
ucEnableLogFile	unsigned char	是否生成 log 文件开关

返回值

返回函数成败标识

9) sitrpSetRotationMat

函数原型

```
unsigned char sitrpSetTranslationMat(float* pfOutBuffer, float* pfInitri, int iInitriLen, unsigned int* puiSuccFlag, unsigned char ucEnableLogFile)
```

函数功能

获取 TOF 和 RGB 的旋转参数

函数参数

参数	类型	描述
pfOutBuffer	float*	输出内参指针
pfInitri	float*	RGB 内参指针
iInitriLen	int	RGB 内参数组长度
puiSuccFlag	unsigned int*	算法成败标识

ucEnableLogFile	unsigned char	是否生成 log 文件开关
-----------------	---------------	---------------

返回值

返回函数成败标识

10) sitrpSetRgbPos

函数原型

unsigned char sitrpSetRgbPos(unsigned char* pBOutBuffer, unsigned char bIsLeft, unsigned int* puiSuccFlag, unsigned char ucEnableLogFile)

函数功能

获取 TOF 和 RGB 的相对位置参数

函数参数

参数	类型	描述
pBOutBuffer	unsigned char*	输出内参指针
bIsLeft	unsigned char	位置参数
puiSuccFlag	unsigned int*	算法成败标识
ucEnableLogFile	unsigned char	是否生成 log 文件开关

返回值

返回函数成功标识

4. Demo 示例

4.1. 图像输出配置

SDK 中会有图像输出时的一些配置函数，例如：帧率配置、曝光时间配置、Binning 模式配置及 Phase 配置等，表格中仅介绍基本功能，详细配置请见示例。

API	功能
-----	----

OpenCamera	打开相机，在此函数中可以设置打开相机类型
SetFps	配置图像帧率
SetCamInt	配置图像曝光时间
SetCamBinning	配置图像 binning 模式，即 VGA/QVGA

示例

//仅截取关键代码, 详细代码请参阅 example 源码

```
MemSinkCfg memCfg;
XdynCamInfo_t camInfo;
unsigned int phaseInt[4] = {1000000, 1000000, 0, 0};
unsigned int specialInt[4] = {1000000, 1000000, 0, 0}; //设置曝光时间, 此处默认为 1000
XdynContextInit();
XDYN_Streamer *stream = CreateStreamerNet(XDYN_PRODUCT_TYPE_XD_500, EventCB, &userHdl,
"192.168.31.3");
if(stream == nullptr){
    printf("get streamer failed, return\n");
    return -1;
}

userHdl.stream = stream;

res = stream->OpenCamera(XDYN_DEV_TYPE_TOF); //设置打开相机类型, 此处默认为 TOF 相机, 具体相机
类型可参考 XDYN_DEV_TYPE_e 中的类型定义
if(res != XD_SUCCESS){
    printf("open camera failed, exitm [%d]\n", res);
    goto END;
}

memCfg.isUsed[MEM_AGENT_SINK_RAW] = true;
memCfg.isUsed[MEM_AGENT_SINK_DEPTH] = true;
memCfg.isUsed[MEM_AGENT_SINK_CONFID] = true;
res = stream->ConfigSinkType(XDYN_SINK_TYPE_CB, memCfg, StreamCB, &userHdl);
if(res != XD_SUCCESS){
    printf("config sink type failed, return [%d]\n", res);
    goto END;
}

res = stream->ConfigAlgMode(XDYN_ALG_MODE_EMB_PASS_IPC_ALG);
if(res != XD_SUCCESS){
    printf("config alg failed, return [%d]\n", res);
```

```

        goto END;
    }

    stream->SetFps(2);    //配置图像输出帧率，此处帧率默认为 2
    stream->SetCamInt(phaseInt, specialInt);    //配置图像曝光时间，此处默认为 phaseInt[4]、specialInt[4]
定义的 1000
    stream->SetCamBinning(XDYN_BINNING_MODE_2x2); // 使用 binning 2x2 的方法，分辨率为 320 * 240
    res = stream->ConfigCamParams();
    if(res != XD_SUCCESS) {
        printf("conifg cam params failed, [%d]\n", res);
        goto END;
    }

    res = stream->StartStreaming(); //上述配置成功后，相机开流，输出图像数据
    if(res != XD_SUCCESS) {
        printf("strart streaming failed, [%d]\n", res);
        goto END;
    }
}

```

4.2. 异常处理

当无法连接到相机时，会抛出异常处理，此时会退出程序。

示例

//仅截取关键代码，详细代码请参阅 example 源码

```

if(event == XDYN_CB_EVENT_DEVICE_DISCONNECT) {
    printf("device is disconnect, exit!!\n");
    doExit = true;
}

```

4.3. 图像输出

在连接好相机后，会获取相机配置，相机开流之后会回调此函数，输出图像数据流，同时也可以设置要保存的图像帧数将图像保存下来，示例中仅展示深度图像的输出及保存配置，其余请参考 SDK 源码。

示例

//仅截取关键代码，详细代码请参阅 example 源码

```

for(int i = 0; i < MEM_AGENT_SINK_MAX; i++) {
    if(cfg->isUsed[i] && data[i].addr) {
        if(i == MEM_AGENT_SINK_DEPTH) {
            SaveImageData_depth(data[i].addr, data[i].size, 10); //第四位参数可设置保存的图像帧数
        }
    }
}

```

```
if(data[i].ex) {  
    XdynDepthFrameInfo_t *depthInfo=(XdynDepthFrameInfo_t*)data[i].ex;  
    printf("get depth size [%d], fUnitOfDepth = %f, hz[%f, %f] timestamp %lu\n",  
        data[i].size, depthInfo->fUnitOfDepth, depthInfo->fModFreqMHZ[0],  
        depthInfo->fModFreqMHZ[1], depthInfo->frameInfo.timestamp);  
}
```