

Camera SDK （C 或 C++）开发指南

Windows and Linux and Arm

浙江迈尔微视科技有限公司 版权所有

2024.4.7

目录

一、 概述.....	1
二、 环境搭建.....	2
2.1 主机推荐系统配置	2
2.2 环境搭建	2
2.2.1 硬件环境搭建	2
2.2.2 网络环境搭建	2
2.2.3 软件环境搭建	4
三、 Camera SDK 使用说明.....	6
3.1 Camera SDK 目录结构.....	6
3.2 开发流程	7
3.2.1 项目工程配置	7
3.2.2 同步方式接口调用流程	7
3.2.3 异步方式接口调用流程	8
四、 Camera SDK 调用注意事项.....	9
4.1 Camera SDK 接口调用总体原则.....	9
4.2 Camera SDK 接口部分调用前置条件.....	9
五、 数据类型说明.....	13
5.1 LX_STATE	13
5.2 LX_DATA_TYPE.....	14
5.3 LX_BINNING_MODE.....	14
5.4 LX_STRUCT_LIGHT_CODE_MODE	16
5.5 LX_ALGORITHM_MODE	16
5.6 LX_CAMERA_WORK_MODE.....	16
5.7 LX_TRIGGER_MODE	17
5.8 LX_CAMERA_FEATURE	17
5.8.1 int 型参数	17
5.8.2 float 型参数	21
5.8.3 bool 型参数	22

5.8.4 string 型参数	23
5.8.5 command 型参数	23
5.8.6 ptr 型参数	24
5.9 LxDeviceInfo	25
5.10 LxIntValueInfo	26
5.11 LxFloatValueInfo	26
5.12 LX_DEVICE_TYPE	27
5.13 LX_OPEN_MODE	27
5.14 FrameDataInfo	28
5.15 FrameInfo	28
六、 API 说明	29
6.1 查找、连接与关闭相机	29
6.1.1 DcGetDeviceList	29
6.1.2 DcOpenDevice	30
6.1.3 DcCloseDevice	30
6.2 开始取流、关闭取流	31
6.2.1 DcStartStream	31
6.2.2 DcStopStream	31
6.3 读取与设置相机参数	32
6.3.1 DcSetIntValue	32
6.3.2 DcGetIntValue	32
6.3.3 DcSetFloatValue	33
6.3.4 DcGetFloatValue	33
6.3.5 DcSetBoolValue	34
6.3.6 DcGetBoolValue	34
6.3.7 DcSetStringValue	35
6.3.8 DcGetStringValue	35
6.3.9 DcGetPtrValue	36
6.3.10 DcSetCmd	36

6.4 保存点云	37
6.4.1 DcSaveXYZ	37
6.5 特殊控制	38
6.5.1 DcSpecialControl	38
6.6 设置 ROI 与参数路径	39
6.6.1 DcSetRol	39
6.7 设置相机 IP	40
6.7.1 DcSetCameraIp	40
6.8 日志信息与版本信息	41
6.8.1 DcGetApiVersion	41
6.8.2 DcSetInfoOutput	41
6.8.3 DcLog	42
6.8.4 DcRegisterFrameCallback	42
6.8.5 DcUnregisterFrameCallback	43
6.9 设备状态回调	43
6.9.1 DcRegisterCameraStatusCallback	43
七、 版本与更新	45
7.1 最新版本号与更新内容	45

一、概述

本文档主要是面向相机设备接入相关的开发人员；目前支持 M4/M4Mega/M4Pro/V1Pro/S2/H3/S2Max 系列产品，本文档主要目的是对 SDK 提供的 API 的使用方法进行说明。Camera SDK 支持在 x86 架构的 windows 操作系统，以及 x86 和 arm 架构的 linux 操作系统上运行。

Camera SDK 提供统一的访问控制和参数配置接口,支持同步或者异步的方式获取设备数据(包括深度图、强度图、点云图、RGB 图)，部分相机型号支持避障、对接、定位等数据。

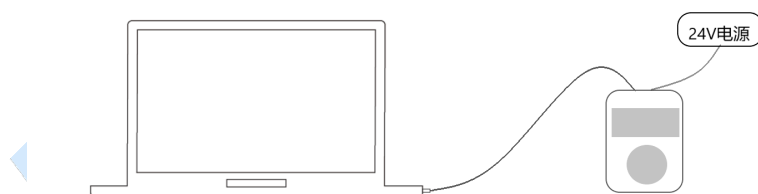
本文档在最大范围内提供开发帮助，提供的内容可能存在瑕疵或者不完整。

二、环境搭建 2.1 主机推荐系统配置

配置项	推荐配置
操作系统	Windows7 Windows10 Windows11 Ubuntu16.04 及以上
内存	4G 及以上
CPU	4 核 Cortex-A57 或同等配置及以上
网卡	千兆及以上

2.2 环境搭建

2.2.1 硬件环境搭建

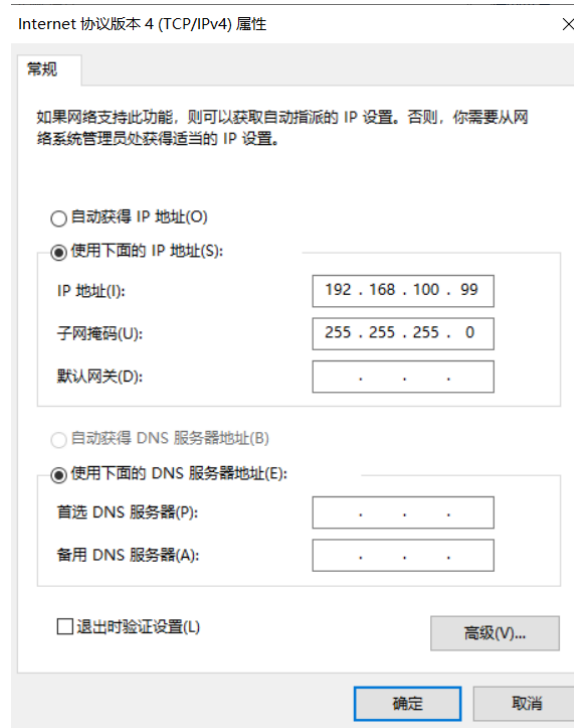


相机外接 24V 电压，并使用网线与上位机所在的主机连接，如上图所示。

2.2.2 网络环境搭建

相机默认 IP 网段为：192.168.100.*

在上位机将与相机相连网口的 IP 地址设置为 192.168.100.x 网段的某个地址并保证和设备不冲突，子网掩码为 255.255.255.0，如下图所示：



连接相机时建议关闭防火墙防止相机数据包被 SDK 所在主机拦截，如下图所示：



Linux 系统网络配置同 windows, 防火墙关闭的方法不同系统可能会有一定差异, 以 ubuntu 为例:

停止防火墙服务: `sudo systemctl stop ufw.service`

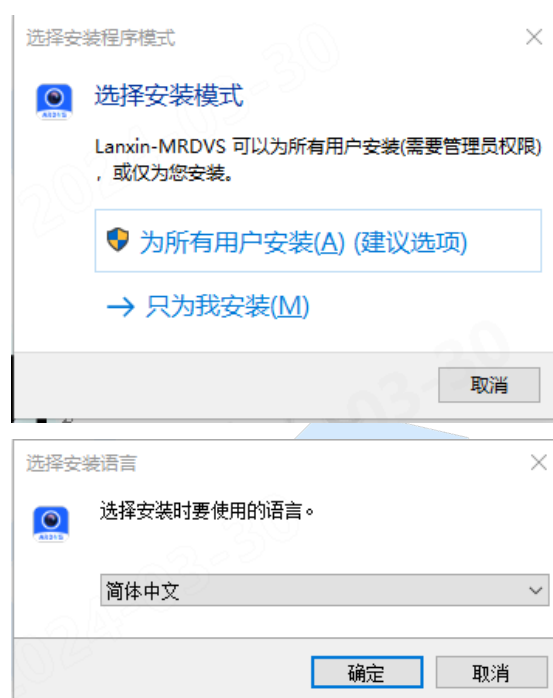
禁用防火墙服务: `sudo systemctl disable ufw.service`

查看防火墙服务是否关闭: `sudo ufw status`

2.2.3 软件环境搭建

2.2.3.1 Windows 系统安装:

打开安装包, 安装 Lanxin-MRDVS-xxx.exe. 如果需要为所有用户安装需要管理员权限运行安装包。





2.2.3.2 Linux 系统安装：

解压并执行安装脚本

```
tar xvf Lanxin-MRDVS-xxx.tar.gz
```

```
cd Lanxin-MRDVS
```

因为安装需要 root 权限，如果当前用户不是 root 的话，需要切换到 root 用户

```
sudo su root
```

然后再执行安装

```
chmod +x install.sh
```

```
./install.sh
```

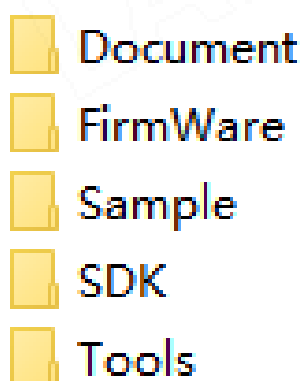
安装脚本会将 sdk 安装到/opt/Lanxin-MRDVS 目录并设置好环境变量。需要注意的是脚本中设置的环境变量是当前用户的环境变量，如果运行程序不是 root 用户，需要重新运行 install.sh 脚本设置环境变量或者手动在当前用户的 ~/.bashrc 末尾添加 `export LD_LIBRARY_PATH=/opt/Lanxin-MRDVS/lib/:$LD_LIBRARY_PATH`。

对于在安装前已经打开的终端会话需要在程序运行前再执行 `source ~/.bashrc` 更新 `LD_LIBRARY_PATH` 环境变量

三、Camera SDK 使用说明

3.1 Camera SDK 目录结构

Camera SDK 目录内容包含如下图：



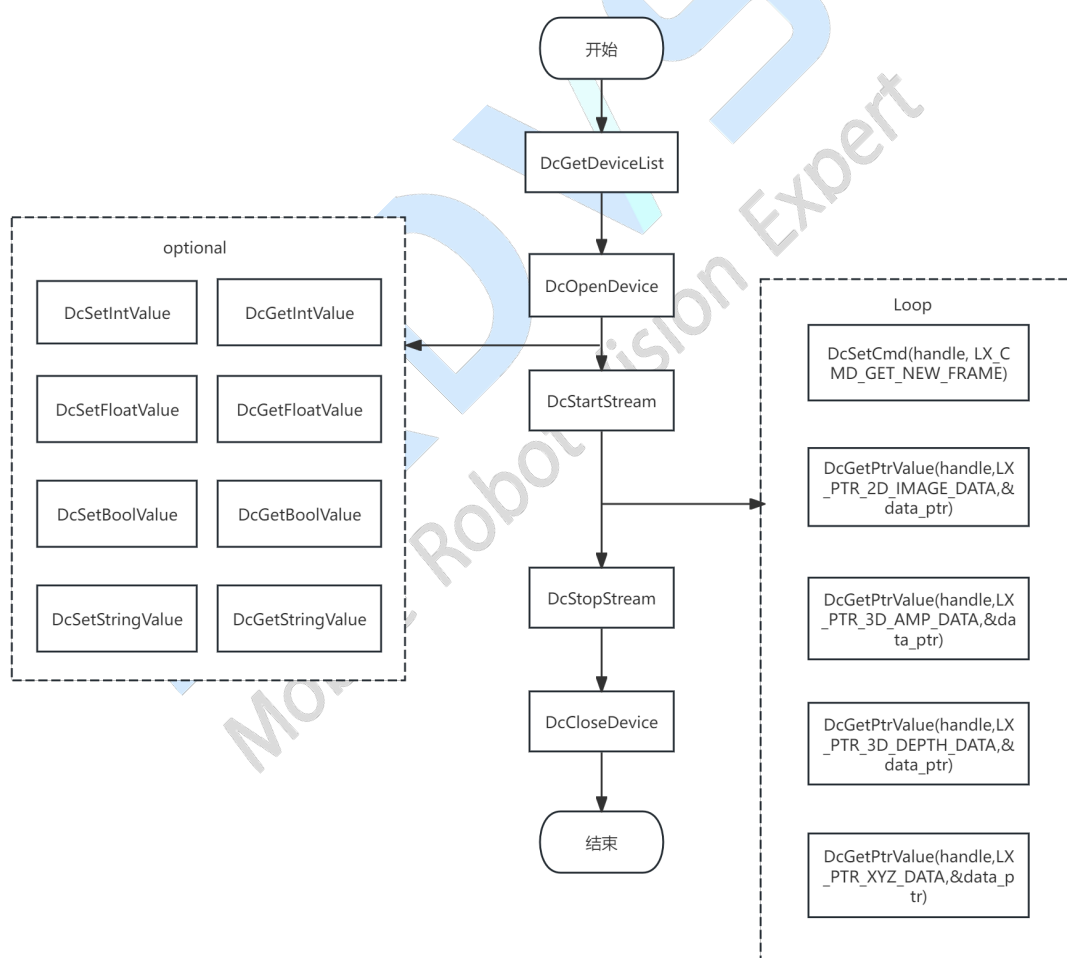
- Document 包含 SDK 的各种说明文档
- FirmWare 包含相机固件升级包
- Sample 包含各种编程语言使用 Camera SDK 的示例代码
- SDK 包含 Camera SDK 的头文件以及库文件

3.2 开发流程

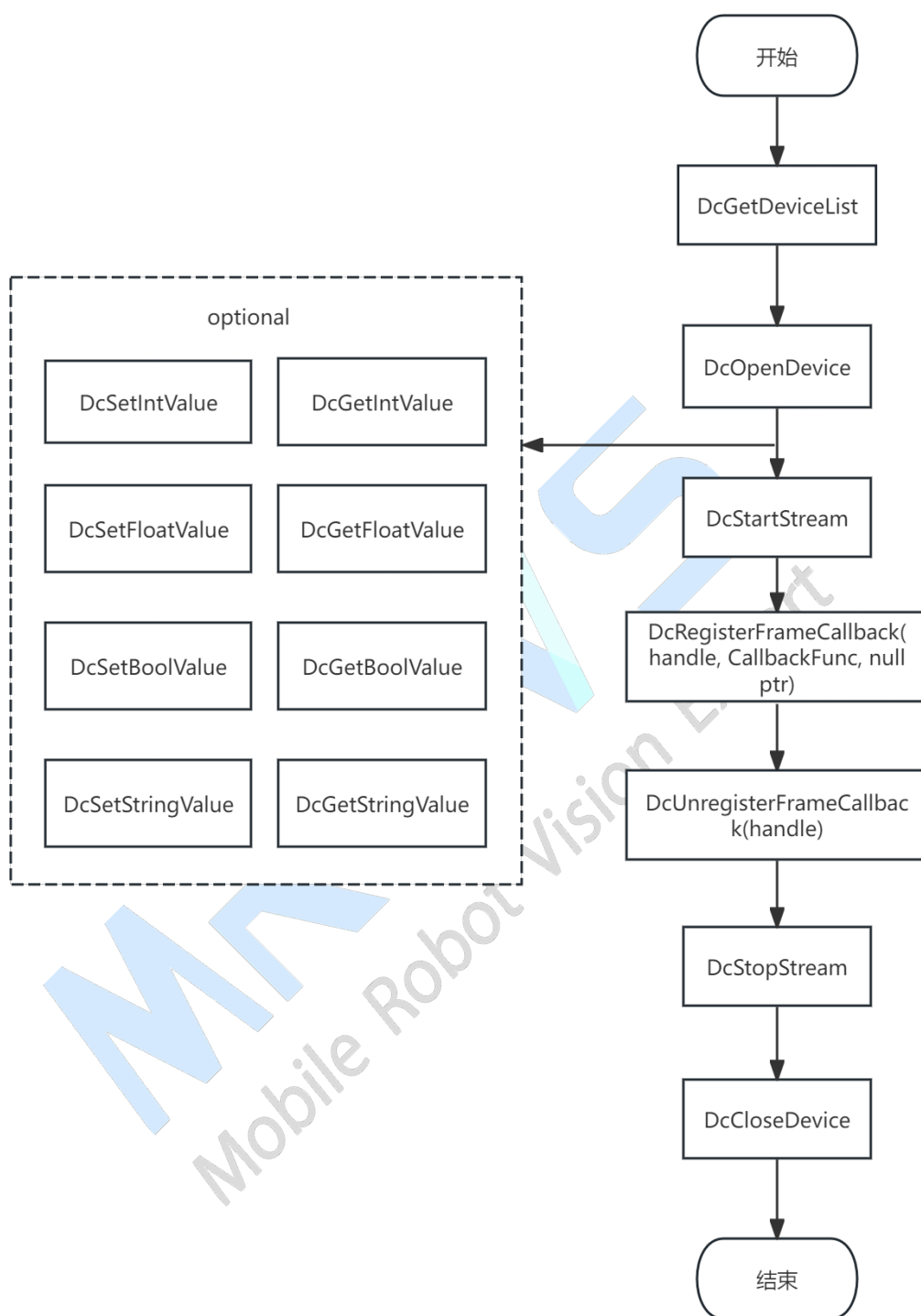
3.2.1 项目工程配置

推荐 c/c++ 作为主要的二次开发编程语言来集成 sdk，提供了通过 cmake 工具来构建工程的 CMakeLists.txt 脚本；同时 Sample 文件夹提供了不同语言开发方法示例。具体可参考 Sample 下的资料以及《linux 示例程序使用说明》。

3.2.2 同步方式接口调用流程



3.2.3 异步方式接口调用流程



四、Camera SDK 调用注意事项

4.1 Camera SDK 接口调用总体原则

枚举类型以 LX_INT 为前缀的使用 DcSetIntValue 或 DcGetIntValue 来设置或获取 INT 类型参数;

枚举类型以 LX_FLOAT 为前缀的使用 DcSetFloatValue 或 DcGetFloatValue 来设置或获取 FLOAT 类型参数;

枚举类型以 LX_BOOL 为前缀的使用 DcSetBoolValue 或 DcGetBoolValue 来设置或获取 BOOL 类型参数;

枚举类型以 LX_STRING 为前缀的使用 DcSetStringValue 或 DcGetStringValue 来设置或获取字符串类型参数;

枚举类型以 LX_CMD 为前缀的使用 DcSetCmd 来执行指令;

枚举类型以 LX_PTR 为前缀的使用 DcGetPtrValue 来获取指针类型结果;

4.2 Camera SDK 接口部分调用前置条件

在启流状态或工作模式 (LX_INT_WORK_MODE) 为常开模式下, 以下功能不允许设置:

DcSetRoi

LX_INT_3D_BINNING_MODE;

LX_INT_2D_BINNING_MODE;

LX_BOOL_ENABLE_2D_TO_DEPTH;

LX_BOOL_ENABLE_MULTI_EXPOSURE_HDR;

LX_BOOL_ENABLE_MULTI_MACHINE;

LX_INT_TRIGGER_MODE;

LX_INT_CALCULATE_UP;

LX_BOOL_ENABLE_HDR

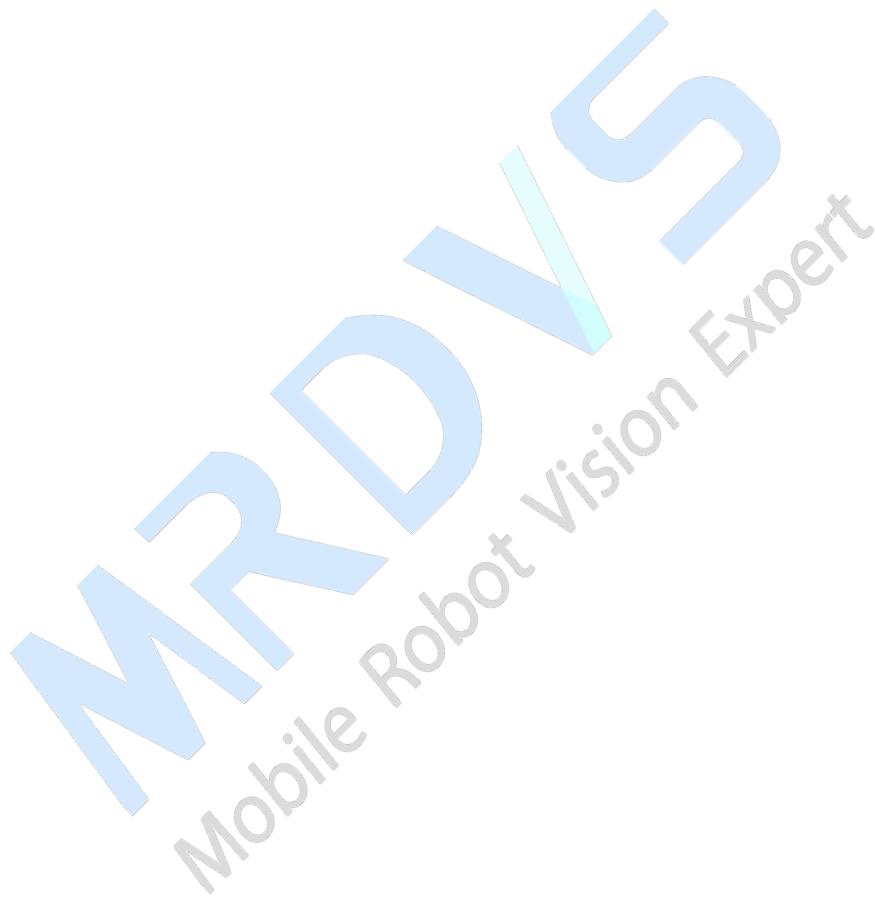
LX_BOOL_ENABLE_TOF_DEPTH_CALIBE

LX_BOOL_ENABLE_SYNC_FRAME

LX_INT_SAVE_PARAMS_GROUP

工作模式（LX_INT_WORK_MODE）为常开模式下也不允许设置

LX_INT_TOF_FREQMODE



3D 自动曝光(LX_BOOL_ENABLE_3D_AUTO_EXPOSURE)开启的情况下, 不允许操作以下功能枚举:

```
LX_BOOL_ENABLE_MULTI_EXPOSURE_HDR ;  
LX_INT_FIRST_EXPOSURE;  
LX_INT_SECOND_EXPOSURE;  
LX_INT_THIRD_EXPOSURE;  
LX_INT_GAIN;
```

3D 自动曝光(LX_BOOL_ENABLE_3D_AUTO_EXPOSURE)关闭的情况下, 不允许操作以下功能枚举:

```
LX_INT_3D_AUTO_EXPOSURE_LEVEL
```

不允许设置 3D 自动曝光最大值(LX_INT_3D_AUTO_EXPOSURE_MAX)和最小值(LX_INT_3D_AUTO_EXPOSURE_MIN)

2D 自动曝光(LX_BOOL_ENABLE_2D_AUTO_EXPOSURE)开启的情况下, 不允许操作以下功能枚举:

```
LX_INT_2D_MANUAL_EXPOSURE
```

2D 自动曝光(LX_BOOL_ENABLE_2D_AUTO_EXPOSURE)关闭的情况下, 不允许操作以下功能枚举:

```
LX_INT_2D_AUTO_EXPOSURE_LEVEL;
```

滤波非常规模式下,不允许设置如下:

```
LX_INT_FILTER_SMOOTH_LEVEL  
LX_INT_FILTER_NOISE_LEVEL  
LX_INT_FILTER_TIME_LEVEL
```

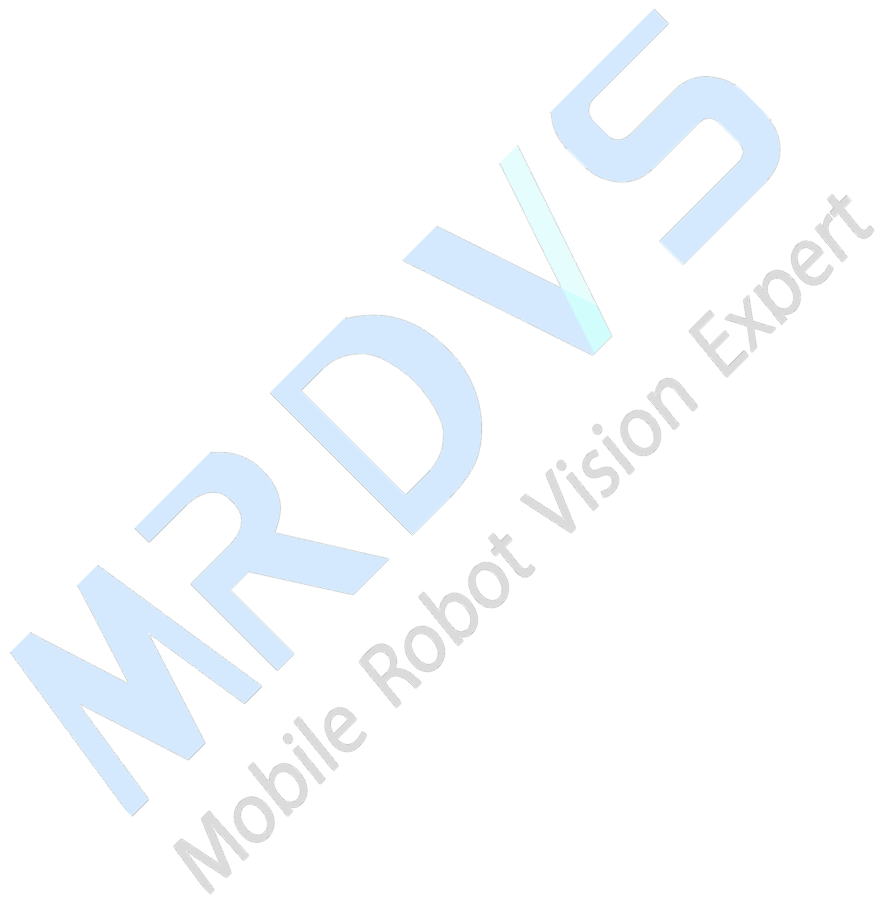
开启 2D 反畸变时, 必须先开启 RGBD 对齐

```
LX_BOOL_ENABLE_2D_TO_DEPTH
```

非 S2 类型的设备，开启内置算法时需先设置 tof 算法下移

开启了内置算法(LX_INT_ALGORITHM_MODE)情况下，才允许设置算法参数
(LX_STRING_ALGORITHM_PARAMS)

要获取点云数据 (LX_PTR_XYZ_DATA)，必须要先开启深度图
(LX_BOOL_ENABLE_3D_DEPTH_STREAM)



五、数据类型说明

5.1 LX_STATE

功能：SDK 接口函数返回值状态码

关键字	说明
LX_SUCCESS	函数执行成功
LX_ERROR	函数执行失败
LX_E_NOT_SUPPORT	所连接的相机不支持该功能
LX_E_NETWORK_ERROR	网络通讯错误
LX_E_INPUT_ILLEGAL	函数参数传入非法
LX_E_RECONNECTING	设备重连中
LX_E_DEVICE_ERROR	设备故障
LX_E_DEVICE_NEED_UPDATE	设备版本过低,需升级固件版本
LX_E_API_NEED_UPDATE	API 版本过低,需升级 SDK
LX_E_CTRL_PERMISS_ERROR	独占控制权限失败(设备同时只能一个 SDK 连接)
LX_E_GET_DEVICEINFO_ERROR	获取设备信息失败
LX_E_IMAGE_SIZE_ERROR	图像尺寸不匹配, 重新打开相机
LX_E_IMAGE_PARTITION_ERROR	图像解析失败
LX_E_DEVICE_NOT_CONNECTED	相机未连接
LX_E_DEVICE_INIT_FAILED	相机初始化失败
LX_E_DEVICE_NOT_FOUND	未找到匹配的相机
LX_E_FILE_INVALID	文件错误(文件名或类型或格式不正确)
LX_E_CRC_CHECK_FAILED	文件 crc 或 md5 校验失败
LX_E_TIME_OUT	超时
LX_E_FRAME_LOSS	漏帧
LX_E_ENABLE_ANYSTREAM_FAILED	开启任意流失败

LX_E_NOT_RECEIVE_STREAM	未收到流数据
LX_E_PARSE_STREAM_FAILED	启流成功但解析流数据失败
LX_E_PROCESS_IMAGE_FAILED	图像计算处理失败
LX_E_SETTING_NOT_ALLOWED	常开模式下不允许设置
LX_E_LOAD_DATAPROCESSLIB_ERROR	加载图像处理算法库错误
LX_E_FUNCTION_CALL_LOGIC_ERROR	函数调用逻辑错误
LX_E_IPAPPDR_UNREACHABLE_ERROR	IP 不可达或网络配置错误
LX_E_FRAME_ID_NOT_MATCH	超时范围内帧不同步错误
LX_E_FRAME_MULTI_MACHINE	帧中检测到多机干扰信号

5.2 LX_DATA_TYPE

功能：数据格式。3D 强度图和深度图有不同的数据格式，获取数据指针需要使用对应的 feature

关键字	说明
LX_DATA_UNSIGNED_CHAR	无符号字符型
LX_DATA_UNSIGNED_SHORT	无符号短整型
LX_DATA_SIGNED_SHORT	有符号短整型
LX_DATA_FLOAT	浮点型
LX_DATA_OBSTACLE	避障算法结构体类型
LX_DATA_PALLET	托盘算法结构体类型
LX_DATA_LOCATION	视觉定位算法结构体类型
LX_DATA_OBSTACLE2	避障算法 V2 结构体类型

5.3 LX_BINNING_MODE

功能：Bining 模式

关键字	说明
LX_BINNING_1X1	1x1Binning

LX_BINNING_2X2	2x2Binning
LX_BINNING_4X4	4x4Binning

MRDVS
Mobile Robot Vision Expert

5.4 LX_STRUCT_LIGHT_CODE_MODE

功能：

关键字	说明
LX_CODE_NORMAL	常规
LX_CODE_STATBLE	稳定
LX_CODE_ENHANCE	高精度加强

5.5 LX_ALGORITHM_MODE

功能：

关键字	说明
MODE_ALL_OFF	关闭相机内置算法
MODE_AVOID_OBSTACLE	内置避障算法
MODE_PALLET_LOCATE	内置托盘算法
MODE_VISION_LOCATION	内置视觉定位算法
MODE_AVOID_OBSTACLE2	内置避障算法 V2

5.6 LX_CAMERA_WORK_MODE

功能：

关键字	说明
KEEP_HEARTBEAT	心跳模式，SDK 心跳中断后相机进入待机状态
WORK_FOREVER	常开模式，相机始终保持工作状态

5.7 LX_TRIGGER_MODE

功能：

关键字	说明
LX_TRIGGER_MODE_OFF	关闭触发模式, 流模式, 默认
LX_TRIGGER_SOFTWARE	软触发模式
LX_TRIGGER_HARDWARE	硬触发模式

5.8 LX_CAMERA_FEATURE

功能：各种可读、可写参数

5.8.1 int 型参数

5.8.1.1 通用参数

关键字	说明
LX_INT_FIRST_EXPOSURE	曝光值, 单位 us, 针对多积分情况为第一个积分的曝光时间
LX_INT_SECOND_EXPOSURE	针对多积分情况为第二个积分的曝光时间
LX_INT_THIRD_EXPOSURE	针对多积分情况为第三个积分的曝光时间, 部分型号支持
LX_INT_FOURTH_EXPOSURE	针对多积分情况为第四个积分的曝光时间, 部分型号支持
LX_INT_GAIN	增益, 与曝光效果等价。会引入噪声, 可适当调节增益防止曝光参数过大
LX_INT_MIN_DEPTH	最小深度值, 超出范围的值置为无效
LX_INT_MAX_DEPTH	最大深度值, 超出范围的值置为无效
LX_INT_MIN_AMPLITUDE	有效信号最小强度值, 超出范围深度值置

	为无效
LX_INT_MAX_AMPLITUDE	有效信号最大强度值，超出范围深度值置为无效
LX_INT_CODE_MODE	结构光相机编码模式，参考 CODE_MODE(部分型号支持)
LX_INT_WORK_MODE	工作模式，参考 LX_CAMERA_WORK_MODE
LX_INT_LINK_SPEED	协商的网卡网速 100-百兆,1000-千兆，只支持获取，不可设置
LX_INT_TOF_GLOBAL_OFFSET	TOF 深度数据偏移
LX_INT_ALGORITHM_MODE	内置算法开启模式，部分型号支持,对应的值参考 LX_ALGORITHM_MODE
LX_INT_TRIGGER_MODE	触发模式，对应的值参考 LX_TRIGGER_MODE
LX_INT_MODBUS_ADDR	modbus 地址，部分型号支持 MODBUS 协议通过串口输出
LX_INT_HEART_TIME	心跳时间，相机超时未收到心跳会自动待机
LX_INT_GVSP_PACKET_SIZE	GVSP 单包数据分包大小，单位字节
LX_INT_CALCULATE_UP	允许 tof 或 rgb 算法上下移，节省上位机或相机算力，可能影响帧率和延时
LX_INT_CAN_BAUD_RATE	can 的波特率值，单位 bps
LX_INT_SAVE_PARAMS_GROUP	将相机当前配置保存为指定的参数组
LX_INT_LOAD_PARAMS_GROUP	一键加载指定索引的参数组

5.8.1.2 3D 图像参数

关键字	说明
LX_INT_3D_IMAGE_WIDTH	3D 图像分辨率当前宽度
LX_INT_3D_IMAGE_HEIGHT	3D 图像分辨率当前高度

LX_INT_3D_IMAGE_OFFSET_X	ROI 水平偏移像素，设置参数请用 DcSetRol
LX_INT_3D_IMAGE_OFFSET_Y	ROI 垂直偏移像素，设置参数请用 DcSetRol
LX_INT_3D_BINNING_MODE	像素合并，参考 LX_BINNING_MODE
LX_INT_3D_DEPTH_DATA_TYPE	深度图像数据格式，只能获取，对应的值参考 LX_DATA_TYPE
LX_INT_3D_AMPLITUDE_CHANNEL	3D 强度图像通道数，与深度图通道共用
LX_INT_3D_AMPLITUDE_DATA_TYPE	强度图像数据格式，只能获取，对应的值参考 LX_DATA_TYPE
LX_INT_3D_AUTO_EXPOSURE_LEVEL	3D 自动曝光开启时的曝光等级,期间不允许设置曝光值与增益
LX_INT_3D_AUTO_EXPOSURE_MAX	3D 自动曝光上限值
LX_INT_3D_AUTO_EXPOSURE_MIN	3D 自动曝光下限值
LX_INT_3D_UNDISTORT_SCALE	3D 图像反畸变系数

5.8.1.4 2D 图像参数

关键字	说明
LX_INT_2D_IMAGE_CHANNEL	2D 图像通道数，单色为 1，彩色为 3
LX_INT_2D_IMAGE_WIDTH	2D 图像分辨率当前宽度
LX_INT_2D_IMAGE_HEIGHT	2D 图像分辨率当前高度
LX_INT_2D_IMAGE_OFFSET_X	2D 图像 ROI 水平偏移像素，只能获取，设置参数请用 DcSetRol
LX_INT_2D_IMAGE_OFFSET_Y	2D 图像 ROI 垂直偏移像素，只能获取，设置参数请用 DcSetRol
LX_INT_2D_BINNING_MODE	像素合并，参考 LX_BINNING_MODE
LX_INT_2D_MANUAL_EXPOSURE	2D 手动曝光时的曝光值

LX_INT_2D_MANUAL_GAIN	2D 手动曝光时的增益值
LX_INT_2D_AUTO_EXPOSURE_LEVEL	2D 图像自动曝光时曝光等级
LX_INT_2D_IMAGE_DATA_TYPE	2D 图像数据格式，只能获取,对应的值参考 LX_DATA_TYPE
LX_INT_2D_UNDISTORT_SCALE	2D 图像反畸变系数

5.8.1.5 触发参数

关键字	说明
LX_INT_TRIGGER_MODE	触发模式,对应的值参考 LX_TRIGGER_MODE
LX_INT_HARDWARE_TRIGGER_FILTER_TIME	硬触发滤波时间, 单位 us
LX_INT_TRIGGER_MIN_PERIOD_TIME	触发最小时间间隔, 单位 us
LX_INT_TRIGGER_DELAY_TIME	触发延迟时间,单位 us, 当值<=1000 时表示立刻生效(预留功能,若大于 1000 时表示延时生效)
LX_INT_TRIGGER_FRAME_COUNT	单次触发帧数
LX_INT_IO_WORK_MODE	GPIO 信号输出控制模式, 参考 LX_IO_WORK_MODE
LX_INT_IO_OUTPUT_STATE	GPIO 信号输出的用户控制模式, 参考 LX_IO_OUTPUT_STATE

5.8.1.6 滤波参数

关键字	说明
LX_INT_FILTER_MODE	滤波模式,参考 LX_FILTER_MODE
LX_INT_FILTER_SMOOTH_LEVEL	当 LX_INT_FILTER_MODE 为 FILTER_NORMAL 时,

	可设置滤波平滑等级, [0, 3], 值越大, 滤波越强
LX_INT_FILTER_NOISE_LEVEL	当 LX_INT_FILTER_MODE 为 FILTER_NORMAL 时, 可设置滤波噪声等级, [0, 3], 值越大, 滤波越强
LX_INT_FILTER_TIME_LEVEL	当 LX_INT_FILTER_MODE 为 FILTER_NORMAL 时, 可设置滤波时域等级, [0, 3], 值越大, 滤波越强

5.8.2 float 型参数

关键字	说明
LX_FLOAT_FILTER_LEVEL	滤波等级, 0-1, 值越大滤波强度越高
LX_FLOAT_EST_OUT_EXPOSURE	是否评估过曝数据, 0-1, 为 1 则过曝数据无效
LX_FLOAT_LIGHT_INTENSITY	光强度, 部分型号支持
LX_FLOAT_3D_DEPTH_FPS	深度图当前帧率
LX_FLOAT_3D_AMPLITUDE_FPS	强度图当前帧率
LX_FLOAT_2D_IAMGE_FPS	RGB 图当前帧率
LX_FLOAT_DEVICE_TEMPERATURE	获取当前相机温度

5.8.3 bool 型参数

关键字	说明
LX_BOOL_CONNECT_STATE	当前连接状态
LX_BOOL_ENABLE_3D_DEPTH_STREAM	开启/关闭深度图(部分相机支持)
LX_BOOL_ENABLE_3D_AMP_STREAM	开启/关闭强度图(部分相机支持)
LX_BOOL_ENABLE_3D_AUTO_EXPOSURE	3D 自动曝光使能
LX_BOOL_ENABLE_3D_UNDISTORT	3D 反畸变使能
LX_BOOL_ENABLE_BACKGROUND_AMP	强度背景光使能
LX_BOOL_ENABLE_ANTI_FLICKER	抗频闪使能，LED 环境照明可能导致数据存在明显波纹，部分型号支持
LX_BOOL_ENABLE_2D_STREAM	开启/关闭 RGB
LX_BOOL_ENABLE_2D_AUTO_EXPOSURE	2D 自动曝光使能
LX_BOOL_ENABLE_2D_UNDISTORT	2D 反畸变使能
LX_BOOL_ENABLE_2D_TO_DEPTH	RGBD 坐标对齐使能
LX_BOOL_ENABLE_MULTI_MACHINE	多机模式使能，部分型号支持
LX_BOOL_ENABLE_MULTI_EXPOSURE_HDR	HDR（多曝光高动态范围模式）使能
LX_BOOL_ENABLE_SYNC_FRAME	是否开启强制帧同步，默认数据实时性优先，若需要 RGBD 同步，需要开启该模式

5.8.4 string 型参数

关键字	说明
LX_STRING_DEVICE_VERSION	设备版本号
LX_STRING_FIRMWARE_NAME	固件文件名,用于升级设备版本,只能设置
LX_STRING_FILTER_PARAMS	滤波算法参数, json 格式的字符串
LX_STRING_ALGORITHM_PARAMS	内置算法参数, 不同的开启模式, 对应不同的 json 格式字符串,前提需要设置过 LX_ALGORITHM_MODE
LX_STRING_ALGORITHM_VERSION	内置算法版本号,不同的开启模式, 返回对应的版本号,只能获取,前提需要设置过 LX_ALGORITHM_MODE
LX_STRING_DEVICE_OS_VERSION	设备系统镜像版本号
LX_STRING_IMPORT_PARAMS_FROM_FILE	从本地文件加载参数到相机
LX_STRING_EXPORT_PARAMS_TO_FILE	将相机当前参数导出到本地文件

5.8.5 command 型参数

关键字	说明
LX_CMD_GET_NEW_FRAME	更新一次数据, 默认超时时间 1s
LX_CMD_RETURN_VERSION	回退上一版本
LX_CMD_RESTART_DEVICE	重启相机
LX_CMD_WHITE_BALANCE	自动白平衡
LX_CMD_RESET_PARAM	恢复默认参数

5.8.6 ptr 型参数

关键字	说明
LX_PTR_2D_IMAGE_DATA	获取 2D 图像数据指针，数据长度由 2D 图像尺寸、通道数和数据格式（LX_INT_2D_IMAGE_DATA_TYPE）确定
LX_PTR_3D_AMP_DATA	获取 3D 强度图数据指针，数据长度由 3D 图像尺寸、通道数和数据格式（LX_INT_3D_AMPLITUDE_DATA_TYPE）确定
LX_PTR_3D_DEPTH_DATA	获取 3D 深度图数据指针，数据长度由 3D 图像尺寸、通道数和数据格式（LX_INT_3D_DEPTH_DATA_TYPE）确定
LX_PTR_XYZ_DATA	获取点云数据指针，float*类型三通道(x, y, z 为一组数据，依次循环)，数据长度为 LX_INT_3D_IMAGE_WIDTH*LX_INT_3D_IMAGE_HEIGHT*sizeof(float)*3
LX_PTR_ALGORITHM_OUTPUT	获取内置算法输出，与算法模式有关。 MODE_AVOID_OBSTACLE 为 LxAvoidanceOutput 指针， MODE_PALLET_LOCATE 为 LxPalletPose 指针， MODE_VISION_LOCATION 为参考 LxLocation 指针
LX_PTR_2D_INTRIC_PARAM	获取 2D 图像内参，float*类型指针，长度固定为 9*sizeof(float)(fx,fy,cx,cy,k1,k2,p1,p2,k3)
LX_PTR_3D_INTRIC_PARAM	获取 3D 图像内参，float*类型指针，长度固定为 9*sizeof(float)(fx,fy,cx,cy,k1,k2,p1,p2,k3)
LX_PTR_3D_EXTRIC_PARAM	获取 3D 图像外参，float*类型指针，长度固定为 12*sizeof(float)(前 9 个表示旋转矩阵，后 3 个表示平移向量)
LX_PTR_FRAME_DATA	获取完整一帧数据，输出结果参考结构体 FrameInfo

5.9 LxDeviceInfo

功能：设备属性信息结构体

字段	类型	说明
handle	DCHandle	设备唯一标识
dev_type	LX_DEVICE_TYPE	设备类型
id	Char 数组	设备 id
ip	Char 数组	设备 ip:port
sn	Char 数组	设备序列号
mac	Char 数组	设备 mac 地址
firmware_ver	Char 数组	设备软件版本号
algor_ver	Char 数组	设备算法版本号
name	Char 数组	设备名称，如：camera_M3_192.168.11.13_9803
reserve	Char 数组	预留字段，子网掩码
reserve2	Char 数组	预留字段 2，网关
reserve3	Char 数组	预留字段 3
reserve4	Char 数组	预留字段 4

5.10 LxIntValueInfo

功能：设备属性信息结构体

字段	类型	说明
set_available	Bool	当前值是否可设置, true-可设置, false-不可设置
cur_value	Int	当前值
max_value	Int	最大值
min_value	Int	最小值
reserve	Int 数组	预留字段

5.11 LxFloatValueInfo

功能：设备属性信息结构体

字段	类型	说明
set_available	Bool	当前值是否可设置, true-可设置, false-不可设置
cur_value	float	当前值
max_value	float	最大值
min_value	float	最小值
reserve	float 数组	预留字段

5.12 LX_DEVICE_TYPE

功能：设备类型

关键字	说明
LX_DEVICE_M2	M2 相机
LX_DEVICE_M3	M3 相机
LX_DEVICE_M4Pro	M4Pro 相机
LX_DEVICE_M4_MEGA	M4MEGA 相机
LX_DEVICE_M4	M4 相机
LX_DEVICE_S1	S1 相机
LX_DEVICE_S2	S2 相机
LX_DEVICE_I1	I1 相机
LX_DEVICE_I2	I2 相机
LX_DEVICE_T1	T1 相机
LX_DEVICE_T2	T2 相机
LX_DEVICE_H3	H3 相机
LX_DEVICE_V1Pro	V1Pro 相机
LX_DEVICE_NULL	预留

5.13 LX_OPEN_MODE

功能：设备类型

关键字	说明
OPEN_BY_INDEX	按搜索列表中索引下标方式打开，对应的参数为索引号，当搜索到的设备列表发生变化时，选择打开的设备也会不一样
OPEN_BY_IP	按搜索列表中对应 ip 方式打开，对应的参数为设备 ip 或 ip:port
OPEN_BY_SN	按搜索列表中对应 sn 方式打开，对应的参数为设备 sn
OPEN_BY_ID	按搜索列表中对应 id 方式打开，对应的参数为设备 id

5.14 FrameDataInfo

功能：图像显示信息结构体

字段	类型	说明
frame_data_type	LX_DATA_TYPE	帧数据类型
frame_width	Int	帧数据图像宽
frame_height	Int	帧数据图像高
frame_channel	Int	帧数据通道数
frame_data	Void *指针类型	帧数据内容
sensor_timestamp	Unsigned long long 类型	sensor 出图时间戳
recv_timestamp	Unsigned long long 类型	接收完帧数据时的时间戳

5.15 FrameInfo

功能：数据帧信息结构体

字段	类型	说明
frame_state	LX_STATE	帧数据状态
handle	DCHandle	设备唯一标识
depth_data	FrameDataInfo 类型	深度图帧数据结构体
amp_data	FrameDataInfo 类型	强度图帧数据结构体
rgb_data	FrameDataInfo 类型	Rgb 图帧数据结构体
app_data	FrameDataInfo 类型	算法输出结构体
reserve_data	Void *指针类型	扩展预留字段

六、API 说明

6.1 查找、连接与关闭相机

6.1.1 DcGetDeviceList

函数原型：

```
#ifndef __cplusplus
LX_API DcGetDeviceList(LxDeviceInfo** devlist, int* devnum, LX_DEVICE_SERIALS lx_serials = LX_SERIAL_ALL);
#else
// 纯C风格接口，原接口逐步弃用
LX_API DcGetDeviceList(LxDeviceInfo** devlist, int* devnum);
#endif
```

函数功能：

获取支持的相机列表。

函数参数：

- [out]devlist 查找到的相机列表
- [out]devnum 查找到的相机数量

返回值：

Int 数据，该值等于 LX_SUCCESS，说明接口调用成功。

备注：内部维护内存，外部无须分配内存，但因每次搜索前会清空原内容，故需外部保证线程安全

6.1.2 DcOpenDevice

函数原型:

```
LX_API DcOpenDevice(LX_OPEN_MODE open_mode, const char* param, DcHandle* handle, LxDeviceInfo* info);
```

函数功能:

连接设备。

函数参数:

- [in]open_mode 打开方式, 具体说明见 OpenMode
- [in]param 不同的打开方式, 填写不同的参数
- [out]handle 连接成功后返回的设备句柄,后续所有接口访问都依赖该 handle
- [out]info 连接成功后返回的相机详细信息

返回值:

Int 数据, 该值等于 LX_SUCCESS, 说明接口调用成功。

备注: 应确保传入参数有效

6.1.3 DcCloseDevice

函数原型:

```
LX_API DcCloseDevice(DcHandle handle);
```

函数功能:

关闭设备

函数参数:

- [in]handle 设备句柄

返回值:

Int 数据, 该值等于 LX_SUCCESS, 说明接口调用成功。

备注: 应确保传入参数有效

6.2 开始取流、关闭取流

6.2.1 DcStartStream

函数原型：

```
LX_API DcStartStream(DCHandle handle);
```

函数功能：

打开数据流

函数参数：

- [in]handle 设备句柄

返回值：

Int 数据，该值等于 LX_SUCCESS，说明接口调用成功。

备注：应确保传入参数有效

6.2.2 DcStopStream

函数原型：

```
LX_API DcStopStream(DCHandle handle);
```

函数功能：

关闭数据流

函数参数：

- [in]handle 设备句柄

返回值：

Int 数据，该值等于 LX_SUCCESS，说明接口调用成功。

备注：应确保传入参数有效

6.3 读取与设置相机参数

6.3.1 DcSetIntValue

函数原型:

```
LX_API DcSetIntValue(DCHandle handle, int cmd, int value);
```

函数功能:

设置 int 类型参数

函数参数:

- [in]handle 设备句柄
- [in]cmd 参考 LX_CAMERA_FEATURE
- [in]value 设置参数值

返回值:

Int 数据, 该值等于 LX_SUCCESS, 说明接口调用成功。

备注: 应确保传入参数有效

6.3.2 DcGetIntValue

函数原型:

```
LX_API DcGetIntValue(DCHandle handle, int cmd, LxIntValueInfo* value);
```

函数功能:

获取 int 类型参数

函数参数:

- [in]handle 设备句柄
- [in]cmd 参考 LX_CAMERA_FEATURE
- [out]value 返回参数结构体, 参数的最大最小值、当前值等

返回值:

Int 数据, 该值等于 LX_SUCCESS, 说明接口调用成功。

备注: 应确保传入参数有效

6.3.3 DcSetFloatValue

函数原型:

```
LX_API DcSetFloatValue(DCHandle handle, int cmd, float value);
```

函数功能:

设置 float 类型参数

函数参数:

- [in]handle 设备句柄
- [in]cmd 参考 LX_CAMERA_FEATURE
- [in]value 设置参数值

返回值:

Int 数据, 该值等于 LX_SUCCESS, 说明接口调用成功。

备注: 应确保传入参数有效

6.3.4 DcGetFloatValue

函数原型:

```
LX_API DcGetFloatValue(DCHandle handle, int cmd, LxFloatValueInfo* value);
```

函数功能:

获取 float 类型参数

函数参数:

- [in]handle 设备句柄
- [in]cmd 参考 LX_CAMERA_FEATURE
- [out]value 返回参数结构体, 参数的最大最小值、当前值等

返回值:

Int 数据, 该值等于 LX_SUCCESS, 说明接口调用成功。

备注: 应确保传入参数有效

6.3.5 DcSetBoolValue

函数原型:

```
LX_API DcSetBoolValue(DcHandle handle, int cmd, bool value);
```

函数功能:

设置 bool 类型参数

函数参数:

- [in]handle 设备句柄
- [in]cmd 参考 LX_CAMERA_FEATURE
- [in]value 设置参数值

返回值:

Int 数据, 该值等于 LX_SUCCESS, 说明接口调用成功。

备注: 应确保传入参数有效

6.3.6 DcGetBoolValue

函数原型:

```
LX_API DcGetBoolValue(DcHandle handle, int cmd, bool* value);
```

函数功能:

获取 bool 类型参数

函数参数:

- [in]handle 设备句柄
- [in]cmd 参考 LX_CAMERA_FEATURE
- [out]value 返回参数

返回值:

Int 数据, 该值等于 LX_SUCCESS, 说明接口调用成功。

备注: 应确保传入参数有效

6.3.7 DcSetStringValue

函数原型:

```
LX_API DcSetStringValue(DcHandle handle, int cmd, const char* value);
```

函数功能:

设置 string 类型参数

函数参数:

- [in]handle 设备句柄
- [in]cmd 参考 LX_CAMERA_FEATURE
- [in]value 设置参数值

返回值:

Int 数据, 该值等于 LX_SUCCESS, 说明接口调用成功。

备注: 应确保传入参数有效

6.3.8 DcGetStringValue

函数原型:

```
LX_API DcGetStringValue(DcHandle handle, int cmd, char** value);
```

函数功能:

获取 string 类型参数

函数参数:

- [in]handle 设备句柄
- [in]cmd 参考 LX_CAMERA_FEATURE
- [out]value 返回参数, 无需外部分配内存

返回值:

Int 数据, 该值等于 LX_SUCCESS, 说明接口调用成功。

备注: 应确保传入参数有效

6.3.9 DcGetPtrValue

函数原型：

```
LX_API DcGetPtrValue(DcHandle handle, int cmd, void** value);
```

函数功能：

获取指针类型参数

函数参数：

- [in]handle 设备句柄
- [in]cmd 参考 LX_CAMERA_FEATURE
- [out]value 返回参数，无需外部分配内存

返回值：

Int 数据，该值等于 LX_SUCCESS，说明接口调用成功。

备注：应确保传入参数有效

6.3.10 DcSetCmd

函数原型：

```
LX_API DcSetCmd(DcHandle handle, int cmd);
```

函数功能：

执行对应 CMD 类型指令操作

函数参数：

- [in]handle 设备句柄
- [in]cmd 参考 LX_CAMERA_FEATURE

返回值：

Int 数据，该值等于 LX_SUCCESS，说明接口调用成功。

备注：应确保传入参数有效

6.4 保存点云

6.4.1 DcSaveXYZ

函数原型：

```
LX_API DcSaveXYZ(DcHandle handle, const char* filename);
```

函数功能：

保存点云，可直接调用

函数参数：

- [in]handle 设备句柄
- [in]filename 文件名，支持 txt,ply 和 pcd 格式。txt 格式按图像顺序保存所有数据，ply 和 pcd 仅保存非零数据

返回值：

Int 数据，该值等于 LX_SUCCESS，说明接口调用成功。

备注：应确保传入参数有效

6.5 特殊控制

6.5.1 DcSpecialControl

函数原型：

```
LX_API DcSpecialControl DcHandle handle, const char* command, void* value);
```

函数功能：

LX_CAMERA_FEATURE 定义之外的特殊操作

函数参数：

- [in]handle 设备句柄
- [in]param 操作
- [inout]value 设置时为对应入参，获取时为对应出参，无需外
部分配内存

返回值：

Int 数据，该值等于 LX_SUCCESS，说明接口调用成功。

备注：应确保传入参数有效

6.6 设置 ROI 与参数路径

6.6.1 DcSetRol

函数原型:

```
LX_API DcSetROI DcHandle handle, int offsetx, int offsety, int width, int height, int img_type)
```

函数功能:

设置 ROI 区域, 输入数值若不是 8 的整数倍, 内部会自动处理为目标值最近的 8 的整倍数。设置后需要更新图像尺寸参数

函数参数:

- [in]handle 设备句柄
- [in]offsetx 起始点水平偏移像素
- [in]offsety 起始点垂直偏移像素
- [in]width roi 目标区域的宽
- [in]height roi 目标区域的高
- [in]type 0-3D 图像 1-2D 图像

返回值:

Int 数据, 该值等于 LX_SUCCESS, 说明接口调用成功。

备注: 应确保传入参数有效

6.7 设置相机 IP

6.7.1 DcSetCameraIp

函数原型:

```
#ifndef __cplusplus
LX_API DcSetCameraIp(DCHandle handle, const char* ip, const char* netmask = 0, const char* gateway = 0);
#else
//纯C风格接口, 原接口逐步弃用
LX_API DcSetCameraIp(DCHandle handle, const char* ip, const char* netmask, const char* gateway);
#endif
```

函数功能:

设置相机 IP 和子网掩码,网关

函数参数:

- [in]handle 设备句柄,
- [in]ip 设备 IP,
- [in]netmask 子网掩码(若传空则内部默认"255.255.0.0")
- [in]gateway 网关 ip(若传空则内部默认将 ip 最后网段置为"1"后作为网关)

返回值:

Int 数据, 该值等于 LX_SUCCESS, 说明接口调用成功。

备注: 当未连接上设备情况下, 需先通过搜索获取。若设备支持修改网关和子网掩码, 则同时将网关设置为同网段, 子网掩码设置为"255.255.0.0"。修改完之后设备列表会变化, 需重新调用 DcGetDeviceList 接口重新获取新的设备列表

6.8 日志信息与版本信息

6.8.1 DcGetApiVersion

函数原型:

```
LX_API_STR DcGetApiVersion();
```

函数功能:

获取 API 版本号

函数参数:

返回值:

char 指针数据。

备注:

6.8.2 DcSetInfoOutput

函数原型:

```
#ifndef __cplusplus
LX_API DcSetInfoOutput(int print_level = 2, bool enable_screen_print = false, const char* log_path = "", int language = 0);
#else
//纯C风格接口、原接口逐步弃用
LX_API DcSetLog(int print_level, bool enable_screen_print, const char* log_path);
#endif
```

函数功能:

设置打印信息等级

函数参数:

- [in]print_level 0: info 所有调试信息; 1: warn 重要及警告类调试信息; 2: error 仅输出错误信息
- [in]enable_screen_print 是否在窗口打印
- [in]log_path log 文件保存路径(不包含日志文件名), 可以为空。
Windows 默认在当前用户路径, linux 默认在/var/log 路径
- [in]language 语言(暂无用)

返回值:

Int 数据, 该值等于 LX_SUCCESS, 说明接口调用成功。

备注：应确保传入参数有效

6.8.3 DcLog

函数原型：

```
LX_API DcLog(const char* str);
```

函数功能：

允许用户输出调试信息到 log 文件

函数参数：

- [in]str 要输出的字符串, '\0'结尾

返回值：

Int 数据, 该值等于 LX_SUCCESS, 说明接口调用成功。

备注：应确保传入参数有效

6.8.4 DcRegisterFrameCallback

函数原型：

```
#ifdef __cplusplus
LX_API DcRegisterFrameCallback DcHandle handle, LX_FRAME_CALLBACK func, void* usr_data = 0);
#else
/*纯C风格接口, 原接口逐步弃用
LX_API DcRegisterFrameCallback DcHandle handle, LX_FRAME_CALLBACK func, void* usr_data);
#endif
```

函数功能：

注册数据帧回调函数,收到新的数据时自动调用

函数参数：

- [in]handle 设备句柄,
- [in]func 帧数据回调函数,
- [in]usr_data 用户自定义参数, 可传空

返回值：

Int 数据, 该值等于 LX_SUCCESS, 说明接口调用成功。

备注：应确保传入参数有效

6.8.5 DcUnregisterFrameCallback

函数原型:

```
LX_API DcUnregisterFrameCallback(DCHandle handle);
```

函数功能:

取消帧回调

函数参数:

- [in]handle 设备句柄,

返回值:

Int 数据, 该值等于 LX_SUCCESS, 说明接口调用成功。

备注: 应确保传入参数有效

6.9 设备状态回调

6.9.1 DcRegisterCameraStatusCallback

函数原型:

```
LX_API DcRegisterCameraStatusCallback(DCHandle handle, LX_CAMERA_STATUS_CALLBACK func, void* usr_data)
```

函数功能:

注册相机状态回调函数,相机状态变化时自动调用

函数参数:

- [in]handle 设备句柄,
- [in]func 帧数据回调函数,
- [in]usr_data 用户自定义参数, 可传空

备注: 应确保传入参数有效

6.9.2 DcUnregisterCameraStatusCallback

函数原型:

```
LX_API DcUnregisterCameraStatusCallback(DcHandle handle)
```

函数功能:

取消相机状态注册回调

函数参数:

- [in]handle 设备句柄,

返回值:

Int 数据, 该值等于 LX_SUCCESS, 说明接口调用成功。

备注: 应确保传入参数有效

MRDVS
Mobile Robot Vision Expert

七、版本与更新

7.1 最新版本号与更新内容

版本号	更新说明
2.4.16	1.支持 S2MAX 相机 2.新增参数导入导出和参数组切换 3.新增相关功能设置
2.4.9	1.优化并增加部分功能和配置参数 2.时间戳改为 us
2.4.2	支持 M4/M4Mega/M4Pro/V1Pro/S2/H3 相机
2.0.2	修复部分已知问题，调整部分接口与参数