



NST Stereo Camera Driver Documentation

发布 1.0.0

NST

2021 年 08 月 16 日

目录

公司简介.....	4
一、 使用相机与 SDK.....	5
1.1 解压缩 SDK 包.....	5
1.2 SDK 目录介绍.....	5
1.3 编译示例代码.....	6
1.4 安装使用相机.....	7
二、 示例代码讲解.....	9
2.1 获取原始图像.....	9
2.2 获取校正图像.....	10
2.3 获取 RGBD 图像.....	11
2.4 设置相机参数.....	12
三、 SDK API 简述.....	13
3.1 功能简介.....	13
3.2 API 说明.....	13
3.2.1 moGetSdkVersion.....	13
3.2.2 moOpenUVCCameraByPath.....	13
3.2.3 moOpenUVCCameraByNumber.....	14
3.2.4 moCloseCamera.....	14
3.2.5 moSuspendCamera.....	14
3.2.6 moResumeCamera.....	15
3.2.7 moGetBxfAndBase.....	15
3.2.8 moGetCurrentFrame.....	16
3.2.9 moGetSafeCurrentFrame.....	16
3.2.10 moGetRGBDImage.....	16
3.2.11 moGetRGBDDisparityData.....	17
3.2.12 moGetRGBDYUVI420Image.....	17
3.2.13 moGetRawImage.....	18
3.2.14 moGetRawLeftBayerImage.....	18

3.2.15	moGetRawRightBayerImage.....	18
3.2.16	moGetRectifiedImage.....	19
3.2.17	moGetRectifiedLeftGrayImage.....	19
3.2.18	moGetRectifiedRightYUVI420Image.....	19
3.2.19	moSetVideoMode.....	20
3.2.20	moGetVideoMode.....	20
3.2.21	moGetVideoFrameParam.....	20
3.2.22	moGetRealTimeFPS.....	21
3.2.23	moSetFilllightType.....	21
3.2.24	moGetFilllightType.....	21
四、	常见问题答疑.....	23
4.1	程序是否可以在虚拟机运行.....	23
4.2	无法获得图像数据.....	23



公司简介

江苏钷芯集成电路技术股份有限公司是全球领先的双目立体视觉及芯片解决方案提供商，公司自成立以来，专注双目立体视觉领域，掌握自主知识产权，实现双目立体视觉领域 0-1 多项突破，公司在技术、产品、商业化落地等方面塑造唯一，力争第一。技术方面已成为全球顶尖的智能立体视觉理论与技术引领者；产品打造极致匠心，持续创新；商业模式上行业深度赋能，钉钉子式垂直深耕，再进行行业复制，业务呈爆发式增长。

国家高新技术企业，专注双目立体视觉领域，是软硬件一体的解决方案提供商。钷芯掌握自主知识产权，比肩世界最先进技术，持续行业赋能，从芯片、硬件、软件、平台、应用等多个层级赋能合作伙伴。钷芯将为机器赋予智慧的双眼，深度感知世界，与世界深度交互，创造更多可能，让机器更好的为人类服务。钷芯打破国际垄断，自主研发双目立体视觉芯片，并已量产，得到市场验证，为机器视觉添上一枚“中国芯”，以做世界的“中国双目”为己任，保持立体视觉技术持续领先，成为全球机器视觉发展的新高地。

双目立体视觉基于视差原理并利用成像设备从不同位置获取被测物体的图像，通过计算图像对应点见的位置偏差，来获取物体三维几何信息的方法。双目立体视觉将二维图像分析推广到了复杂的三维场景，并进行深度感知。双目立体视觉相对于其他立体深度感知技术，具有高效率、更高分辨率、更好适应室外环境等优点。

双目立体视觉领域，业界最关注的是双目原理从实验室到应用的转型，最大技术门槛在于可靠的高精度计算。钷芯硬核的技术实力，不仅在可靠性和精度方面达到了世界领先水平，并将算法硬件化，做成芯片，立体匹配更强，处理更高效，能耗却更低，用技术实力实现降本增效。

一、使用相机与 SDK

1.1 解压缩 SDK 包

一台装有 Ubuntu16.04 64 位的 PC 上，使用命令：

```
$ tar -zxf mo_stereo_camera_driver_vx.x.x.tar.gz
```

解压缩该文件，可以得到一个 mo_stereo_camera_driver_vx.x.x 目录。

1.2 SDK 目录介绍

mo_stereo_camera_driver_vx.x.x 目录结构如下：

-- doc	# 本文档所在目录
-- NST_stereo_camera_driver_doc_zh_cn.pdf	# 本文档
-- include	# SDK 接口头文件
-- mo_stereo_camera_driver_c.h	# 基础功能接口头文件
-- mo_stereo_camera_driver_c_utilities.h	# 实用功能接口头文件
-- mo_stereo_camera_driver_macro_define.h	# 基础宏定义头文件
-- mo_stereo_camera_driver_type_define.h	# 基础类型定义头文件
-- lib	# SDK 库文件所在目录
-- libmoStereoCameraDriver.so	# 无版本号符号连接
-- libmoStereoCameraDriver.so.1.0	# 两位版本号符号连接
-- libmoStereoCameraDriver.so.1.0.0	# 三位版本号库文件
-- sample	# 示例代码所在目录
-- include	# 示例代码头文件目录
-- common_function.h	# 示例代码共用功能头文件
-- source	# 示例代码实现文件目录
-- common_function.cpp	# 示例代码共用功能实现文件
-- get_raw_sample.cpp	# 示例代码原始图像实现文件
-- get_rectify_sample.cpp	# 示例代码校正图像实现文件
-- get_rgb_sample.cpp	# 示例代码 RGB 图像实现文件
-- set_camera_sample.cpp	# 示例代码相机参数设置实现文件
-- Makefile	# 示例代码编译 Makefile 文件

1.3 编译示例代码

打开 `mo_stereo_camera_driver_vx.x.x` 目录下的 `Makefile` 文件:

```
$ cd mo_stereo_camera_driver_vx.x.x
```

```
$ gedit Makefile
```

(如下为 `Makefile` 内容片断)

开启示例代码使用 `OpenCV` 来展示图像, 就需要修改

```
WANNA_USE_OPENCV := NO
```

```
#WANNA_USE_OPENCV := YES
```

为

```
#WANNA_USE_OPENCV := NO
```

```
WANNA_USE_OPENCV := YES
```

```
ifeq (${WANNA_USE_OPENCV}, YES)
```

```
    EXTRA_CFLAGS := -DWANNA_USE_OPENCV
```

```
    LIBS += -lopencv_highgui -lopencv_core -lopencv_imgproc -lopencv_imgcodecs
```

```
    # 设置 OpenCV 头文件(如: /usr/local/include)与库文件(如: /usr/local/lib)的实际路径
```

```
    INCPATH      += -I/usr/local/include
```

```
    LIBPATH      += -L/usr/local/lib
```

```
endif
```

保存后, 执行:

```
$ make
```

示例程序将创建在 `bin` 目录下:

```
mo_stereo_camera_driver_vx.x.x
```

```
| -- bin
```

```
    | -- get_raw_sample
```

```
    | -- get_rectify_sample
```

```
    | -- get_rgbd_sample
```

```
    | -- set_camera_sample
```

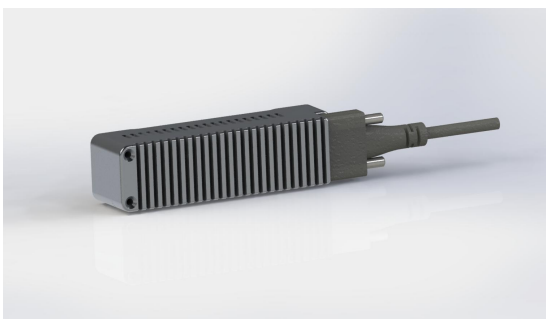
1.4 安装使用相机



图一 NST Depth Camera 5504k 相机正面



图二 NST Depth Camera 5504k 相机反面



图五 NST Depth Camera 5504k 相机自带数据线

请使用相机自带 USB3.0 数据线将相机与 PC 机连接, 在 Ubuntu16.04 64 位系统中打开终端程序并执行如下查询命令:

在相机连接电脑 PC 之前在终端运行如下命令:

```
$ ls /dev/video*
```

看一下 PC 上是否存在默认相机。

然后连接相机至 PC, 继续执行上述命令, 在结果中看到有新的相机设备节点增加, 该设备节点就是双目相机设备节点。eg: /dev/video0

运行示例程序测试相机功能：

```
mo_stereo_camera_driver_vx.x.x/bin$ ./get_raw_sample 相机序号 # 获取原始图像
mo_stereo_camera_driver_vx.x.x/bin$ ./get_rectify_sample 相机序号 # 获取校正图像
mo_stereo_camera_driver_vx.x.x/bin$ ./get_rgbd_sample 相机序号 # 获取 RGBD 图像
mo_stereo_camera_driver_vx.x.x/bin$ ./set_camera_sample 相机序号 # 设置相机参数
```


二、 示例代码讲解

2.1 获取原始图像

参考代码片段：

```
/**< 1. Open specific camera */
s32Result = moOpenUVCCameraByPath(caCameraPath, &hCameraHandle);

/**< 2. Get current video mode */
s32Result = moGetVideoMode(hCameraHandle, &eVideoMode);

/**< 3. Set RAW mode : frame = Bayer + Bayer */
if (MVM_RAW != eVideoMode) {
    s32Result = moSetVideoMode(hCameraHandle, MVM_RAW);
}

/**< 4. Get current video frame */
s32Result = moGetCurrentFrame(hCameraHandle, &u64ImageFrameNum, &pu8FrameBuffer);

/**< 5. Get left and right image */
s32Result = moGetRawImage(hCameraHandle, pu8FrameBuffer, &pu8LeftBayerImg, &pu8RightBayerImg);

/**< 6. Close specific camera */
moCloseCamera(&hCameraHandle);
```

```
#ifdef WANNA_USE_OPENCV
    waitKey(FETCH_AND_DISPLAY_TIME_LENGTH);
    Mat matLeftBGR;
    Mat matLeftBayer(video_frame_height, video_frame_width, CV_16UC1, pu8LeftBayerImg);
    cvtColor(matLeftBayer, matLeftBGR, COLOR_BayerGB2BGR_EA);
    imshow("LeftBayer", matLeftBGR);

    Mat matRightBGR;
    Mat matRightBayer(video_frame_height, video_frame_width, CV_16UC1, pu8RightBayerImg);
    cvtColor(matRightBayer, matRightBGR, COLOR_BayerGB2BGR_EA);
    imshow("RightBayer", matRightBGR);
#endif // WANNA_USE_OPENCV
```

完整代码详见：sample/source/get_raw_sample.cpp

moGetRawImage API 同时获得左右原始图像。

以下两接口详见：include/mo_stereo_camera/mo_stereo_camera_driver_c.h

moGetRawLeftBayerImage API 只获得左原始图像。

moGetRawRightBayerImage API 只获得右原始图像。

2.2 获取校正图像

参考代码片段：

```
/**< 1. Open specific camera */
s32Result = moOpenUVCCameraByPath(caCameraPath, &hCameraHandle);

/**< 2. Get current video mode */
s32Result = moGetVideoMode(hCameraHandle, &eVideoMode);

/**< 3. Set RECTIFIED mode : frame = Gray(Only Y) + YUV_I420 */
if (MVM_RECTIFIED != eVideoMode) {
    s32Result = moSetVideoMode(hCameraHandle, MVM_RECTIFIED);
}

/**< 4. Get current video frame */
s32Result = moGetCurrentFrame(hCameraHandle, &u64ImageFrameNum, &pu8FrameBuffer);

/**< 5. Get left and right image */
s32Result = moGetRectifiedImage(hCameraHandle, pu8FrameBuffer,
                                &pu8OnlyY, &pu8RectifiedRightYUVI420Img);

/**< 6. Close specific camera */
moCloseCamera(&hCameraHandle);
```

```
#ifndef WANNA_USE_OPENCV
    waitKey(FETCH_AND_DISPLAY_TIME_LENGTH);

    Mat matLeftBGR;

    Mat matLeftRectify(video_frame_height, video_frame_width, CV_8UC1, pu8OnlyY);
    cvtColor(matLeftRectify, matLeftBGR, COLOR_GRAY2BGR);
    imshow("LeftRectify", matLeftBGR);

    Mat matRightBGR; // 1.5 = 12bits / 8bits
    Mat matRightRectify(video_frame_height * 1.5, video_frame_width, CV_8UC1,
                        pu8RectifiedRightYUVI420Img);
```

```
    cvtColor(matRightRectify, matRightBGR, COLOR_YUV2BGR_I420);  
    imshow("RightRectify", matRightBGR);  
#endif // WANNA_USE_OPENCV
```

完整代码详见：sample/source/get_rectify_sample.cpp

moGetRectifiedImage API 同时获得左右校正图像。

以下两接口详见：include/mo_stereo_camera/mo_stereo_camera_driver_c.h

moGetRectifiedLeftGrayImage API 只获得左校正图像。

moGetRectifiedRightYUVI420Image API 只获得右校正图像。

2.3 获取 RGBD 图像

参考代码片段：

```
/**< 1. Open specific camera */  
s32Result = moOpenUVCCameraByPath(caCameraPath, &hCameraHandle);  
  
/**< 2. Get current video mode */  
s32Result = moGetVideoMode(hCameraHandle, &eVideoMode);  
  
/**< 3. Set RAW mode : frame = Disparity + YUV_I420 */  
if (MVM_RGBD != eVideoMode) {  
    s32Result = moSetVideoMode(hCameraHandle, MVM_RGBD);  
}  
  
/**< 4. Get current video frame */  
s32Result = moGetCurrentFrame(hCameraHandle, &u64ImageFrameNum, &pu8FrameBuffer);  
  
/**< 5. Get left and right image */  
s32Result = moGetRGBDImage(hCameraHandle, pu8FrameBuffer,  
                            &pu16RGBDDisparityData, &pu8RGBDYUVI420Img);  
  
/**< 6. Close specific camera */  
moCloseCamera(&hCameraHandle);
```

```
#ifndef WANNA_USE_OPENCV  
    waitKey(FETCH_AND_DISPLAY_TIME_LENGTH);  
    Mat matYUV2BGR; // 12bits / 8bits  
    Mat matYUVI420(video_frame_height * 1.5, video_frame_width, CV_8UC1, pu8RGBDYUVI420Img);  
    cvtColor(matYUVI420, matYUV2BGR, COLOR_YUV2BGR_I420);
```

```
imshow("RGB", matYUV2BGR);

Mat matDisparity(video_frame_height, video_frame_width, CV_8UC3);

GetDisparityImage(matDisparity.rows, matDisparity.cols, matDisparity.step,
                  pul6RGBDDisparityData, matDisparity.data);

imshow("Disparity", matDisparity);
#endif // WANNA_USE_OPENCV
```

完整代码详见：sample/source/get_rgbd_sample.cpp

moGetRGBDImage API 同时获得视差数据和 RGB 图像。

以下两接口详见：include/mo_stereo_camera/mo_stereo_camera_driver_c.h

moGetRGBDDisparityData API 只获得视差数据。

moGetRGBDYUVI420Image API 只获得 RGB 图像。

2.4 设置相机参数

参考代码片段：

```
/**< 1. Open specific camera */
s32Result = moOpenUVCCameraByPath(caCameraPath, &hCameraHandle);

/**< 2. Set fill light type */
s32Result = moSetFillllightType(hCameraHandle, MFT_ON);

/**< 3. Get fill light type */
mo_fillllight_type eFillllightType = MFT_OFF;
s32Result = moGetFillllightType(hCameraHandle, &eFillllightType);

/**< 4. Close specific camera */
moCloseCamera(&hCameraHandle);
```

完整代码详见：sample/source/set_camera_sample.cpp

更多设置相机参数接口详见：include/mo_stereo_camera/mo_stereo_camera_driver_c.h

三、 SDK API 简述

3.1 功能简介

钜芯产品根据市场需求不断拓展，产品体系日益完善。钜芯双目立体视觉产品主要有双目视觉模块、通用性双目相机、车载 ADAS 视觉模块、双目相机标准款、其他领域视觉模块等。主要产品应用分为通用级、车规级和消费级。

钜芯双目产品通过构建三维图像精准识别、测量、检测等，进行全类型感知，无需建模，对运动物体、人、动物等的测量中，图像获取瞬间完成，深度成像分辨率达 1080p，纵向分辨率相当于超过 1000 线，作为软硬件一体的解决方案提供商，鲁棒性更强，帮助客户降本增效，互动体验更佳。

钜芯立体双目相机标准版（NST Depth Camera 5504k）搭载钜芯自主研发的 ASIC 第二代芯片，性能更卓越，已应用于无人机、工业设备、安防、智慧仓储等行业。标配的补光灯，非常完美地解决了无纹理物体的识别难题（如：室内白色的墙面）。此外，钜芯立体双目相机标准版还可获取 IMU 数据，为 VSLAM 框架应用提供精准的六轴坐标系数据。

钜芯立体双目相机标准版产品还提供丰富的 SDK 接口支持，帮助客户迅速进行二次开发，加速研发到产品实现的过程。

3.2 API 说明

3.2.1 moGetSdkVersion

得到当前 SDK 版本

参数

无

返回值

版本号字符串

3.2.2 moOpenUVCCameraByPath

通过设备路径打开 UVC 相机

参数

pcPath UVC 相机的设备路径，例如：/dev/video0
phCameraHandle 得到 UVC 相机的句柄

返回值

0 - 成功，负数 - 失败
 -1 - 无效参数
 -2 - 打开相机失败

3.2.3 moOpenUVCCameraByNumber

通过设备索引号打开 UVC 相机

参数

u8Number UVC 相机的设备索引号，取值范围：0 ~ 126
phCameraHandle 得到 UVC 相机的句柄

返回值

0 - 成功，负数 - 失败
 -1 - 无效参数
 -2 - 打开相机失败

3.2.4 moCloseCamera

关闭指定相机

参数

phCameraHandle 指定相机的句柄

返回值

0 - 成功，负数 - 失败
 -1 - 无效参数

3.2.5 moSuspendCamera

暂停指定相机的视频流

注意：

会影响图像获取, 调用 `moGetCurrentFrame/moGetSafeCurrentFrame` API 前, 先调用 `moResumeCamera` API。

参数

hCameraHandle 指定相机的句柄

返回值

- 0 - 成功, 负数 - 失败
- 1 - 无效参数
- 2 - 命令执行失败

3.2.6 moResumeCamera

恢复指定相机的视频流

参数

hCameraHandle 指定相机的句柄

返回值

- 0 - 成功, 负数 - 失败
- 1 - 无效参数
- 2 - 命令执行失败

3.2.7 moGetBxfAndBase

得到指定相机的 BxF (基线乘以焦距的积) 和基线值

参数

hCameraHandle 指定相机的句柄

pfBxf 基线乘以焦距的积

pfBase 基线值

返回值

- 0 - 成功, 负数 - 失败
- 1 - 无效参数
- 2 - 命令执行失败

3.2.8 moGetCurrentFrame

得到指定相机的指定视频模式的视频帧原始数据

注意：

数据内存由 SDK 管理，一定时间后数据将被刷新！

调用 moGetSafeCurrentFrame API 得到的视频帧原始数据则无此顾虑！

参数

<i>hCameraHandle</i>	指定相机的句柄
<i>pu64ImageFrameNum</i>	视频帧序号
<i>ppu8FrameBuffer</i>	视频帧原始数据

返回值

0 - 成功，负数 - 失败
-1 - 无效参数
-2 - 得到当前视频帧失败

3.2.9 moGetSafeCurrentFrame

得到指定相机的指定视频模式的视频帧原始数据

注意：

数据内存由 SDK 管理，一定时间后数据不会被刷新，直到此 API 再次被调用！

参数

<i>hCameraHandle</i>	指定相机的句柄
<i>pu64ImageFrameNum</i>	视频帧序号
<i>ppu8FrameBuffer</i>	视频帧原始数据

返回值

0 - 成功，负数 - 失败
-1 - 无效参数
-2 - 得到当前视频帧失败

3.2.10 moGetRGBDImage

得到指定相机的 RGBD 视频模式的视差数据和YUVI420图像数据

参数

<i>hCameraHandle</i>	指定相机的句柄
<i>pu8FrameBuffer</i>	视频帧原始数据
<i>ppu16DisparityData</i>	视差数据
<i>ppu8YUVI420Img</i>	YUVI420 图像数据

返回值

0 - 成功, 负数 - 失败
 -1 - 无效参数
 -2 - 命令执行失败

3.2.11 moGetRGBDDisparityData

只得到指定相机的 RGBD 视频模式的视差数据

参数

<i>hCameraHandle</i>	指定相机的句柄
<i>pu8FrameBuffer</i>	视频帧原始数据
<i>ppu16DisparityData</i>	视差数据

返回值

0 - 成功, 负数 - 失败
 -1 - 无效参数
 -2 - 命令执行失败

3.2.12 moGetRGBDYUVI420Image

只得到指定相机的 RGBD 视频模式的YUVI420图像数据

参数

<i>hCameraHandle</i>	指定相机的句柄
<i>pu8FrameBuffer</i>	视频帧原始数据
<i>ppu8YUVI420Img</i>	YUVI420 图像数据

返回值

0 - 成功, 负数 - 失败
 -1 - 无效参数
 -2 - 命令执行失败

3.2.13 moGetRawImage

得到指定相机的原始视频模式的左Bayer图像数据和右Bayer图像数据

参数

<i>hCameraHandle</i>	指定相机的句柄
<i>pu8FrameBuffer</i>	视频帧原始数据
<i>ppu8LeftBayerImg</i>	左 Bayer 图像数据
<i>ppu8RightBayerImg</i>	右 Bayer 图像数据

返回值

- 0 - 成功, 负数 - 失败
- 1 - 无效参数
- 2 - 命令执行失败

3.2.14 moGetRawLeftBayerImage

只得到指定相机的原始视频模式的左Bayer图像数据

参数

<i>hCameraHandle</i>	指定相机的句柄
<i>pu8FrameBuffer</i>	视频帧原始数据
<i>ppu8LeftBayerImg</i>	左 Bayer 图像数据

返回值

- 0 - 成功, 负数 - 失败
- 1 - 无效参数
- 2 - 命令执行失败

3.2.15 moGetRawRightBayerImage

只得到指定相机的原始视频模式的右Bayer图像数据

参数

<i>hCameraHandle</i>	指定相机的句柄
<i>pu8FrameBuffer</i>	视频帧原始数据
<i>ppu8RightBayerImg</i>	右 Bayer 图像数据

返回值

- 0 - 成功, 负数 - 失败

- 1 - 无效参数
- 2 - 命令执行失败

3.2.16 moGetRectifiedImage

得到指定相机的校正视频模式的左灰度图像数据和右YUVI420图像数据

参数

<i>hCameraHandle</i>	指定相机的句柄
<i>pu8FrameBuffer</i>	视频帧原始数据
<i>ppu8LeftGrayImg</i>	左灰度图像数据
<i>ppu8RightYUVI420Img</i>	右 YUVI420 图像数据

返回值

- 0 - 成功, 负数 - 失败
- 1 - 无效参数
- 2 - 命令执行失败

3.2.17 moGetRectifiedLeftGrayImage

只得到指定相机的校正视频模式的左灰度图像数据

参数

<i>hCameraHandle</i>	指定相机的句柄
<i>pu8FrameBuffer</i>	视频帧原始数据
<i>ppu8LeftGrayImg</i>	左灰度图像数据

返回值

- 0 - 成功, 负数 - 失败
- 1 - 无效参数
- 2 - 命令执行失败

3.2.18 moGetRectifiedRightYUVI420Image

只得到指定相机的校正视频模式的右YUVI420图像数据

参数

<i>hCameraHandle</i>	指定相机的句柄
----------------------	---------

pu8FrameBuffer 视频帧原始数据
ppu8RightYUVI420Img 右 YUVI420 图像数据

返回值

0 - 成功, 负数 - 失败
-1 - 无效参数
-2 - 命令执行失败

3.2.19 moSetVideoMode

设置指定相机当前的视频模式

参数

hCameraHandle 指定相机的句柄
eVideoMode 要设置的视频模式

返回值

0 - 成功, 负数 - 失败
-1 - 无效参数
-2 - 命令执行失败

3.2.20 moGetVideoMode

得到指定相机当前的视频模式

参数

hCameraHandle 指定相机的句柄
peVideoMode 已设置的视频模式

返回值

0 - 成功, 负数 - 失败
-1 - 无效参数

3.2.21 moGetVideoFrameParam

获得指定相机当前的视频帧参数

参数

hCameraHandle 指定相机的句柄
pstVideoFrameParam 当前的视频帧参数

返回值

- 0 - 成功, 负数 - 失败
- 1 - 无效参数
- 2 - 命令执行失败

3.2.22 moGetRealTimeFPS

得到统计得出的每秒帧率

参数

hCameraHandle 指定相机的句柄
pd8FPS 当前的视频帧参数

返回值

- 0 - 成功, 负数 - 失败
- 1 - 无效参数
- 2 - 失败通常是由于开始时没有足够的时间来统计

3.2.23 moSetFilllightType

设置指定相机的补光灯类型

参数

hCameraHandle 指定相机的句柄
eFilllightType 想要设置的补光灯类型

返回值

- 0 - 成功, 负数 - 失败
- 1 - 无效参数
- 2 - 命令执行失败

3.2.24 moGetFilllightType

获得指定相机的当前补光灯类型

参数

hCameraHandle 指定相机的句柄
eFilllightType 当前 设置的补光灯类型

返回值

0 - 成功, 负数 - 失败
-1 - 无效参数
-2 - 命令执行失败

四、常见问题答疑

4.1 程序是否可以在虚拟机运行

确保 PC 机为真实物理机器（不支持虚拟机，如：Virtualbox、VMWare 等），且支持 USB3.0 (Super Speed USB) 标准接口, 安装有 Ubuntu16.04 64 位操作系统。

4.2 无法获得图像数据

确保 PC 机支持 USB3.0 (Super Speed USB) 标准接口, 且 USB3.0 数据线为相机自带或兼容 USB3.0 标准。



图三 蓝色 USB 接口为 USB3.0 接口



图四 附有 SS 标识的非蓝色 USB3.0 接口