

专 业 3D 传 感 器 高 科 技 企 业



SSZN SDK使用说明









- 01) SDK简介
- 02 采图流程&参考范例
- 03 数据转换
- 04 异常分析



专业 3D 传感器高科技企业





SDK简介

Demo例程的主体由一个动态链接库(.dll),一个静态链接库(.lib)和一个头文件(.h)组成

SR7Link.dll	2021/1/28 15:53	应用程序扩展	231 KB
h SR7Link.h	2020/12/3 15:11	C++ Header file	23 KB
■ SR7Link.lib	2021/1/28 15:53	Object File Library	45 KB

C#使用的时候需要将dII放置到(三种任意一种即可,一般使用第一种即可)

- 1.软件运行目录(bin/x64(x86)/Debug(Release)目录下)
- 2.System32目录下
- 3.将路径添加到环境变量中

然后在.cs文件中添加using System. Runtime. InteropServices;从非托管代码中导出函数的必要调用信息,使用 D11Import函数将其导出使用,例如下图的函数SR7IF_EthernetOpen。

```
[D11Import("SR7Link.d11", CallingConvention = CallingConvention.Cdec1)]
1 个引用
internal static extern int SR7IF_EthernetOpen(int 1DeviceId, ref SR7IF_ETHERNET_CONFIG pEthernetConfig);
```

函数接口位于SR7LinkFunc.cs中,可直接将这个类添加到工程中使用,或参照其中的写法在自己的类中定义。

E Frogramics	בטבטן דוס ודידט	Visual C// Sourcil	LIND
	2023/8/3 18:54	Visual C# Sourc	57 KB

专业 3D 传感器高科技企业



PART 02

采图流程& 参考范例



⊙ 一次回调模式

● 无限循环模式



•一次回调模式,全称叫(一次性)回调模式

一次回调模式就是一次性把采集的数据全部返还。不用主动去获取数据,采集完成后自动触发回调函数,直接去回调函数中取 得数据即可。比如设定批处理长度是10000行,那么会在采集结束后一次性返还10000行数据,这个过程中数据是边采集边传输的 (数据暂存在SDK底层的缓存区),进入回调函数后是从SDK底层缓存区拷贝数据。

我们的文件大小基本可以看作是图像宽*图像高*字节数。这里假定为10000行,图像大小就是3200*10000*4个字节/1024 /1024 =122MB (开16bit压缩时,一个像素为2字节)。按传输速度40MB/s,也就是说整个传输过程需要3s,如果<mark>传输时间<采集</mark> 时间,那么相机运动完基本上就实时传输过去了,如果传输时间>采集时间,那么采集完后需要等待传输完成。



采图流程: 一次回调模式流程





采图流程: 一次回调模参考范例



```
SR7IF ETHERNET CONFIG a = new SR7IF ETHERNET CONFIG();//初始化网络结构体,控制器默认Ip是192.168.0.10
a. abyIpAddress = new Byte[4];
a. abyIpAddress[0] = Convert. ToByte(192);
a. abyIpAddress[1] = Convert. ToByte (168);
a. abyIpAddress[2] = Convert. ToByte(0);
a. abyIpAddress[3] = Convert. ToByte(10);
int DeveiceID =0;//控制器编号 0-63
int reT = SR7LinkFunc. SR7IF EthernetOpen(DeveiceID, ref a);
if(reT < 0)
  /*连接失败*/
int reT = SR7LinkFunc.SR7IF_SetBatchOneTimeDataHandler((int)DeveiceID, batchOneTimeCallBack);
//绑定回调函数(数据在回调函数中去获取)
if(reT < 0)
//回调函数设置失败
reT = SR7LinkFunc. SR7IF CommClose(DeveiceID); //断开连接
if(reT < 0)
/*关闭设备失败*/
else
// 此处设置为等待外部硬件触发批处理
reT = SR7LinkFunc. SR7IF StartMeasureWithCallback(DeveiceID, 1) //设置监听,有软触发或硬触发都会开始采集
if(reT < 0)
/*开始批处理失败*/
return -1:
return 0;
```



采图流程: 一次回调模参考范例

回调函数定义:

```
private void BatchOneTimeCallBack(IntPtr info, IntPtr data)
          SR7IF STR CALLBACK INFO coninfo = new SR7IF STR CALLBACK INFO();//把Info转换成SR7IF STR CALLBACK INFO类型的结构体
          coninfo = (SR7IF STR CALLBACK INFO) Marshal. PtrToStructure(info, typeof(SR7IF STR CALLBACK INFO));
          OneCallProfile = coninfo. BatchPoints;
                                                                           public struct SR7IF STR CALLBACK INFO
          if (coninfo. BatchPoints != 0)
             CallBatchTimes++;
                                                                                                                        //x方向数据数量
                                                                                public int xPoints;
                                                                                                                        //批处理数量
                                                                                public int BatchPoints;
                                                                                                                        //批处理次数
                                                                                public int BatchTimes;
          if (coninfo. returnStatus != 0)
             return;//如果错误码不为0,判定为异常,直接返回
          int mBatchPoint = coninfo. BatchPoints;
                                                                                                                        //x方向点间距
                                                                                public double xPixth;
          int mBatchWidth = coninfo.xPoints;
          //申请内存
                                                                                public int startEncoder;
                                                                                                                        //批处理开始编码器值
          IntPtr[] mTmpData = new IntPtr[2];
                                                                                                                        //相机头数量
                                                                                public int HeadNumber;
          IntPtr[] mTmpGraydata = new IntPtr[2];
          IntPtr[] mTmpEncoderdata = new IntPtr[2];
                                                                                public int returnStatus;
                                                                                                                        //0:正常批处理
          for (int index = 0; index < coninfo.HeadNumber; index++)</pre>
             mTmpData[index] = SR7LinkFunc. SR7IF GetBatchProfilePoint(data, index)://获取高度数据
             mTmpGraydata[index] = SR7LinkFunc. SR7IF GetBatchIntensityPoint(data, index);//获取灰度数据
             mTmpEncoderdata[index] = SR7LinkFunc. SR7IF GetBatchEncoderPoint(data, index)://获取编码器数据
            //数据拷贝写在此处
          GC. Collect();
          bBatchFinish = 1;//批处理结束标志位,在显示线程中循环读这个标志位,批处理结束后就会实时显示图像
```



• 无限循环,全称是(无限制)循环取图模式

顾名思义,就是没有批处理行数上限的,严格来说只要内存够大,可以无限制的存储下去。

缺点:会由于网速原因和用户原因造成数据的溢出。

原因:无限循环的逻辑就是不断的去控制器取数据,如果这部分数据没有被及时取走,新的数据又生成了,那么这部分数据就有可能被覆盖。

网络原因就是短时间内的数据量过大,生成速度比传输速度快,造成数据溢出。 用户原因可能是程序运行时CPU占用率高,数据没及时取走,造成数据的溢出。

需要注意的是,要使用无限循环模式时,批处理模式必须处于循环模式中

有两个方式可以设置

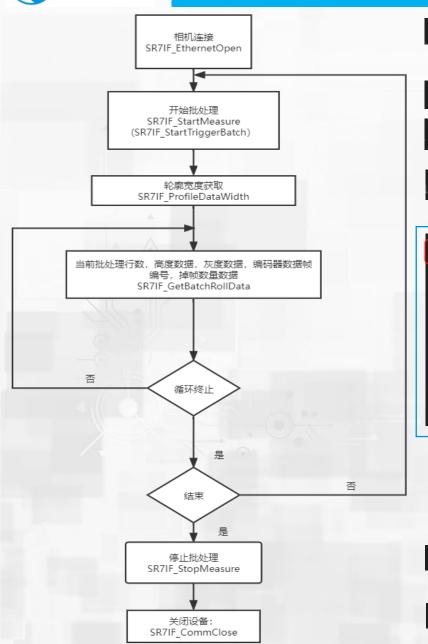
- 1.在EdgeImaging高级选项中将循环勾选。
- 2.用SetSetting函数将模式设置成循环。







采图流程: 无限循环模式流程



```
int SR7IF_EthernetOpen(int lDeviceId, ref SR7IF_ETHERNET_CONFIG pEthernetConfig);
int SR7IF_StartIOTriggerMeasure(int 1DeviceId, int Timeout, int restart);
int SR7IF_StartMeasure(int lDeviceId, int Timeout);
                                                 软触发或者硬触发,选一种方式触发即可
int SR7IF_ProfileDataWidth(int lDeviceId, IntPtr DataObj);
int SR7IF_GetBatchRollData(int lDeviceId,
                         IntPtr DataObj,
                         IntPtr Profile,
                                         返回高度数据
                                                                   循环这个流
                         IntPtr Intensity,
                                         返回亮度 (灰度) 数据
                         IntPtr Encoder,
                                                                   程获取数据
                                         返回编码器数据
                         IntPtr FrameId,
                                         返回帧 (行数) 编号
```

IntPtr FrameLoss,

int GetCnt);

int SR7IF_StopMeasure(int 1DeviceId); 停止批处理后,下次运行从SR7IF_StartMeasureWithCallback开始

int SR7IF_CommClose(int lDeviceId);

关闭相机后,下次运行从SR7IF_EthernetOpen开始

批处理过快掉帧数量数据指针

单次获取数据最大长度(行数)

采图流程: 无限循环模参考范例



```
SR7IF ETHERNET CONFIG a = new SR7IF ETHERNET CONFIG();//初始化网络结构体,控制器默认Ip是192.168.0.10
a. abyIpAddress = new Byte[4];
a. abyIpAddress[0] = Convert. ToByte (192);
a. abyIpAddress[1] = Convert. ToByte(168);
a. abvIpAddress[2] = Convert. ToBvte(0):
a. abvIpAddress[3] = Convert. ToBvte(10):
int DEVICE ID = 0;
int reT = SR7LinkFunc.SR7IF EthernetOpen(DEVICE ID, ref a)://连接相机。
SR7LinkFunc. SR7IF StartMeasure (DEVICE ID, 20000); //开始批处理, IO触发时使用SR7IF StartIoTriggerMeasure();
int m DataWidth = SR7LinkFunc. SR7IF ProfileDataWidth(DEVICE ID, new IntPtr());//获取轮廓宽度
/*变量初始化*/
 //循环接收数据
int m curBatchPoint = SR7LinkFunc. SR7IF GetBatchRollData(DEVICE ID, DataObject, HeightData, GrayData, Encode
r, FrameId, FrameLoss, 500); //循环接收当前批次高度数据、灰度数据、编码器数据、帧编号、掉帧数量数据
if (m curBatchPoint == 0)
continue:
int TmpID = m curBatchPoint - 1;
OverFlowStartId = FrameId[TmpID];//帧编号
FrameLossID = FrameLoss[TmpID]; //丢帧数
 EncoderID = Encoder[TmpID]; //编码器
BatchPoint CurNo += m curBatchPoint;
Sleep(50); //此处可添加一个延时显示
 }while(b stop);
/****用户程序处理****/
/*******/
//内存释放
/*内存释放代码写在此处*/
SR7LinkFunc. SR7IF StopMeasure (DEVICE ID); //停止批处理
SR7LinkFunc. SR7IF CommClose(DEVICE ID); //关闭设备
return 0:
```



专业 3D 传感器高科技企业





数据转换

我们通过接口得到的高度数据是存在一维数组里的,要转换为三维的x,y,z可以参考如下代码

```
int width = 3200;//轮廓宽度
int height = 4000;//批处理点数
double xpitch = 0.03;//x间距
double ypitch = 0.03;//y间距

double x, y, z;
int[] heightdata = new[width * height];//申请空间
for(int i=0;i<width * height ; i++)//遍历数组转化
{
    x= i %width * xpitch;
    y= i/width * ypitch;
    z= (double)(heightdata[i] /100000);//除100000后单位是mm
}
```



专业 3D 传感器高科技企业





#define: SR7IF_ERROR_NOT_FOUND:	(-999:))	// 功能(相机)不存在。	设备即号不在范围内,0-3
#define SR7IF_ERROR_COMMAND	(-998)	// 该命令不支持.	
define: SR7IF_ERROR_PARAMETER	(-997/)	// 参数错误.	传入的参数错误,如空指针、批处理行数不合理等
define SR7IF_ERROR_UNIMPLEMENTED	(-996))	/// 功能未实现.	
define SR7IF_ERROR_HANDLE	(-995)	// 句柄无效	
define SR7IF_ERROR_MEMORY	(-994)	// 内存(溢出/定义)错误.	内存申请失败,增加PC内存
#define SR7IF_ERROR_TIMEOUT		// 操作超时.	批处理超时,设置更大的等待时间,在函数SR7IF_StartMeasure/SR7IF_StartIOTriggerMeasure
	(-993)		使用千兆网卡
			外部10端口开始批处理,注意等待时间
define SR7IF_ERROR_DATABUFFER	(-992)	// 数据大缓冲区不足	
define SR7IF_ERROR_STREAM	(-991)	// 数据流错误.	
define SR7IF_ERROR_CLOSED	(-990))	// 接口关闭不可用。	相机未连接,先调用SR7IF_BthernetOpen
#define SK/IF_ERROR_CLOSED	(-880)		连接后断开-检查硬件接线等
define SR7IF_ERROR_VERSION	(-989)	// 当前版本无效.	控制器参数导入,版本不匹配
		// 操作被终止,如连接被关闭、连接中断	检查电脑IP和相机是否在一个网段
dofine SP7IE EPROR ABORT	(-988)		
#define SR7IF_ERROR_ABORT	(-000)		更换高质里的网线
			注意电脑网卡的稳定性,尽量使用PC自带网卡
		// 操作和现有的设置冲突.	未设置灰度数据输出下获取灰度数据;
define SR7IF_ERROR_ALREADY_EXISTS:	(-987)		网络已连接重新连接
			非批处理模式下获职批处理数据
define SR7IF_ERROR_FRAME_LOSS	(-986)	// 批处理帧丢失	批处理过快,一般在编码器触发模式下移动速度过快。设置建议:在理论范围内取80%
define SR7IF_ERROR_ROLL_DATA_OVERFL	_OW (-985)	// 无终止循环批处理出现置出	异常等.
define SR7IF_ERROR_ROLL_BUSY	(-984)	// 无終止循环批处理读数据忙。	
define SR7IF_ERROR_MODE_	(-983)	// 当前处理函数与设置的批处理模式有对	突打开批处理ON,或者关闭高级功能循环模式
			检查控制器和相机的线缆插针
define SR7IF_ERROR_CAMERA_NOT_ONLIN	NE. (-982)	// 相机(传感头)不在线。	检查控制器和相机的线缆是否正常;
			单相机使用必须连接到A口
			电脑和控制器间的网线连接检查
			更换高质量的网线
			注意电脑网卡的稳定性,
			尽量直连不要通过交换机,避免IP冲突
define SR7IF_NORMAL_STOP:	(-100)	// 正常停止,如外部10停止批处理操作等	10端口停止批处理
			命令停止批处理
#define SR7IF_OK: (0) // 正确操作.		正党派回	

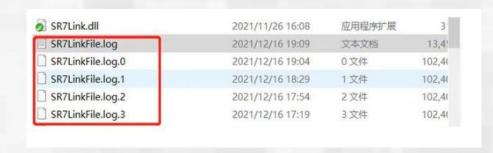


异常分析: 相机库LOG

1、在软件根目录下面 (SR7Link.dll所在位置) , 找到 "EnableLogout.ini" 配置文件, 用记事本打开



2、查看日志,用记事本打开日志,通过时间戳,找到有问题的日志,分析异常原因





深视智能 助力工业4.0 专业 3D 测量 专家



深圳市深视智能科技有限公司

 ${\tt SHENZHEN\,SINCEVISION\,TECHNOLOGY\,CO.,LTD}$

公司地址:深圳市南山区南山智园崇文园区2栋5层

华东办事处:江苏省昆山市创业路1588号象屿两岸贸易中心7号楼1304室 Http:www.cnsszn.com Mail:support@cnsszn.com Tel:0755-29655425

4009660626