

MV-EM E 系列

工业相机开发手册

开发手册 V2.2.1903

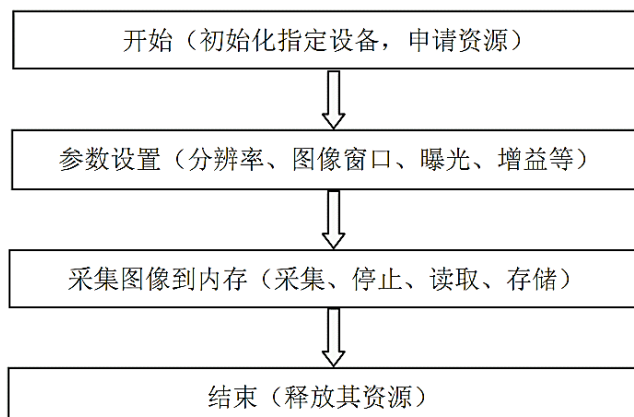
目录

1、概述.....	- 2 -
1.1、相机文件.....	- 2 -
1.2、开发工具.....	- 3 -
2、说明.....	- 3 -
2.1、定义	- 3 -
2.1.1、_IMAGE_INFO	- 3 -
2.1.2、MVCamInfo	- 4 -
2.1.3、MVStreamStatistic.....	- 5 -
2.1.4、MVAVI	- 5 -
2.1.5、MVImage	- 7 -
2.2、文件说明.....	- 10 -
2.2.1、MVAvi.h	- 15 -
2.2.2、MVCamProptySheet.h	- 15 -
2.2.3、MVImage.h.....	- 62 -
2.2.4、MVImageC.h.....	- 62 -
2.2.5、MVRecordDlg.h	- 68 -
2.2.6、MVSequenceDlg.h	- 70 -

1、概述

在 Microsoft 的 32 位\64 位 Windows 操作系统中，数字摄像机应用接口库直接操作数字摄像机，并提供给用户一个简单明确的应用接口。用户在编制自己的应用程序时，可以直接调用这些库函数来实现指定的功能。

数字摄像机工作流程如下：



一般情况下，摄像机的开始操作和初始化参数的设置，最好在用户应用程序的初始化中完成，摄像机的结束操作应在应用程序退出前执行。

1.1、相机文件

相机所需的文件在安装目录下，相机默认的安装路径为 C:\Program Files\Microvision\E_EM。

头文件:

GigECamera_Types.h

MVAvi.h

MVCamProptySheet.h

MVGigE.h

MVImage.h

MVImageC.h

MVRecordDlg.h

MVSequenceDlg.h

动态链接库：

MVGigE.dll

MVCamProptySheet.dll

MVTickDlg.dll

静态链接库：

MVGigE.lib

MVCamProptySheet.lib

MVTickDlg.lib

1.2、开发工具

应用接口库支持32 位\64 位编程开发工具 Microsoft Visual C\C++、Microsoft Visual Basic 。

使用 C/C++编程工具，用户应在程序中调用相关的包含文件（.h），并将静态链接库（.lib）

文件加入到工程文件中，供编译程序在链接（Link）时使用。

2、说明

数字摄像机应用接口库包括数字摄像机的控制、采集图像到内存、错误处理、其它功能等。

2.1、定义

文件中模块相关常量，数据结构，宏，函数的说明。

2.1.1、_IMAGE_INFO

```
#include<GigECamera_Types.h>
```

成员变量

- uint64_t nTimeStamp

时间戳，采集到图像的时刻，精度为0.01us

- USHORT nBlockId

帧号，从开始采集开始计数

- UCHAR * pImageBuffer

图像指针，即指向(0,0)像素所在内存位置的指针，通过该指针可以访问整个图像

- ULONG nImageSizeAcq

采集到的图像大小[字节数]

- UCHAR nMissingPackets

传输过程中丢掉的包数量

- uint64_t nPixelFormat

像素格式

- uint32_t nSizeX

图像宽度

- uint32_t nSizeY

图像高度

- uint32_t nOffsetX

图像水平偏移

- uint32_t nOffsetY

图像垂直偏移

2.1.2、MVCamInfo

```
#include <GigECamera_Types.h>
```

成员变量

- unsigned char mIpAddr [4]

相机 IP 地址

- unsigned char mEthernetAddr [6]

相机 MAC 地址

- char mMfgName [32]

相机厂商名称

- char mModelName [32]

相机型号

- char mSerialNumber [16]

相机序列号

- char mUserDefinedName [16]

用户自定义设置相机名称

- unsigned char m_IfIp [4]

相机连接计算机网卡 IP 地址

- unsigned char m_IfMAC [6]

相机连接到计算机网卡的 MAC 地址

2.1.3、MVStreamStatistic

```
#include <GigECamera_Types.h>
```

成员变量

- unsigned long m_nTotalBuf

从开始采集，总计成功收到的完整图像帧数

- unsigned long m_nFailedBuf

从开始采集，总计收到的不完整图像帧数

- unsigned long m_nTotalPacket

从开始采集，总计收到的图像数据包数

- unsigned long m_nFailedPacket

从开始采集，总计丢失的图像数据包数

- unsigned long m_nResendPacketReq

从开始采集，总计重发请求的图像数据包数

- unsigned long m_nResendPacket

从开始采集，总计重发成功的图像数据包数

2.1.4、MVAVI

```
#include <MVAvi.h>
```

成员函数

- MVAvi()

构造函数

- HRESULT Create(char* strFileName,double framesPersecondPlayback)

功能：创建 AVI 文件

参数说明：

strFileName 创建的 Avi 文件名及保存路径

framesPersecondPlayback 视频播放帧率 fps

返回值：成功返回0，否则返回非零值

- HRESULT SetVideoCompression(MVImage *pImage,AVICOMPRESSOPTIONS *opts,bool ShowDialog,HWND hparent)

功能：设置压缩参数

参数说明：

pImage 图像指针

opts 压缩参数，可以为 NULL

ShowDialog 是否显示压缩参数设置对话框

Hparent 如果显示压缩参数设置对话框，给出父窗口句柄。否则设为 NULL

返回值：成功返回0，否则返回非零值

- HRESULT AddFrame(MVImage *pImage)

功能：向 AVI 中添加新的图片

参数说明：

pImage 图像指针

返回值：成功返回0，否则返回非零值

- HRESULT Close()

功能：录像完成，关闭文件

参数说明：无

返回值：成功返回0，否则返回非零值

- unsigned int GetCountOfAddedImages()

功能：获取已经写入 AVI 文件的图像帧数

参数说明：无

返回值：成功返回0，否则返回非零值

- unsigned int GetImageDataBytesWritten()

功能：获取已经写入 AVI 文件的图像字节数

参数说明：无

返回值：成功返回0，否则返回非零值

2.1.5、MVImage

#include <MVImage.h>

成员函数：

- MVImage()

构造函数

- ~MVImage()

析构函数

- int Creat(int nWidth,int nHeight,int nBPP)

功能：分配 image 内存

参数说明：

nWidth 图像宽度

nHeight 图像高度

BPP 图像位深度

返回值：成功返回0，否则返回非零值

- Bool IsNull()

功能：判断是否获取到图像数据

参数说明：无

返回值：成功返回 true，否则返回 false

- `int GetWidth()`

功能：获取图像宽度

参数说明：无

返回值：图像宽度值

- `int GetHeight()`

功能：获取图像高度

参数说明：无

返回值：图像高度值

- `void * GetBits()`

功能：获取图像数据

参数说明：无

返回值：图像数据指针

- `int GetPitch()`

功能：获取图像跨度，即图像一行数据所占字节数

参数说明：无

返回值：字节数

- `int GetBPP()`

功能：获取图像位深度

参数说明：无

返回值：图像位深度值

- `HBITMAP GetHBitmap()`

功能：获取当前图像，存入 HBITMAP 结构

参数说明：无

返回值:HBITMAP 结构

- `BOOL Draw(HDC hDestDC,int xDest,int yDest,int nDestWidth,int nDestHeight,int xSrc,int ySrc,int nSrcWidth,int nSrcHeight)`

功能：在目标区域上显示当前图片

参数说明：

hDestDC 目标显示区域设备上下文

xDest 显示区左上角横轴坐标

yDest 显示区域左上角竖轴坐标

nDestWidth 显示区域宽度

nDestHeight 显示区域高度

xSrc 图像源横向偏移

ySrc 图像源纵向偏移

nSrcWidth 图像源宽度

nSrcHeight 图像源高度

返回值：成功返回 true,否则返回 false

- BOOL Draw(HDC hDestDC,int xDest,int yDest,int nDestWidth,int nDestHeight)

功能：在目标区域上显示当前图片

参数说明：

hDestDC 目标显示区域设备上下文

xDest 显示区左上角横轴坐标

yDest 显示区域左上角竖轴坐标

nDestWidth 显示区域宽度

nDestHeight 显示区域高度

返回值：成功返回 true,否则返回 false

- BOOL Draw(HDC hDestDC,int xDest,int yDest)

功能：在目标区域上显示当前图片

参数说明：

hDestDC 目标显示区域设备上下文

xDest 显示区左上角横轴坐标

yDest 显示区域左上角竖轴坐标

返回值：成功返回 true,否则返回 false

- int Save(LPCTSTR pszFileName)

功能：保存当前图像到文件

参数说明：PszFileName 文件名

返回值：成功返回0，否则返回非零值

- void *GetImageHandle()

功能：获取图像句柄

参数说明：无

返回值：成功图像句柄，失败返回 NULL

- void Destory()

功能：销毁创建的图像对象

参数说明：无

返回值：无

- HDC GetDC()

功能：获取显示设备描述

参数说明：无

返回值：无

- void ReleaseDC()

功能：释放申请的设备资源

参数说明：无

返回值：无

2.2、文件说明

GigECamera_Types.h

#include <stdint.h>

宏定义

- #define interface struct
- #define INT64_MAX 0x7fffffffffffffffLL /*maximum signed __int64 value */
- #define INT64_MIN 0x8000000000000000LL /*minimum signed __int64 value */
- #define UINT64_MAX 0xffffffffffffffffULL /*maximum unsigned __int64 value */
- #define INT32_MAX 0x000000007fffffffLL /*maximum signed __int32 value */
- #define INT32_MIN 0xffffffff80000000LL /*minimum signed __int32 value */
- #define UINT32_MAX 0x00000000ffffffffULL /*maximum unsigned __int32 value */
- #define INT8_MAX 0x0000000000000007fLL /*maximum signed __int8 value */
- #define INT8_MIN 0xffffffffffff80LL /*minimum signed __int8 value */
- #define UINT8_MAX 0x000000000000000ffULL /*maximum unsigned __int8 value */
- #define MVGIGE_API __declspec(dllimport)

类型定义

- typedef struct _IMAGE_INFO MV_IMAGE_INFO
- typedef struct _IMAGE_INFO * pMV_IMAGE_INFO
- typedef int(__stdcall*MVStreamCB)(MV_IMAGE_INFO*, long nUserVal);

枚举定义

- enum MV_BAYER_MODE{BayerRG,BayerBG,BayerGR,BayerGB,BayerInvalid}
- enum MV_PixelFormatEnums{

PixelFormat_Mono8 = 0x01080001, //!<8Bit 灰度

PixelFormat_BayerBG8=0x0108000B, //!<8Bit Bayer 图,颜色模式为 BGGR

PixelFormat_BayerRG8=0x01080009, //!<8Bit Bayer 图,颜色模式为 RGGB

PixelFormat_BayerGB8=0x0108000A, //!<8Bit Bayer 图,颜色模式为 GBRG

PixelFormat_BayerGR8=0x01080008, //!<8Bit Bayer 图,颜色模式为 GRBG

PixelFormat_BayerGRW8=0x0108000C, //!<8Bit Bayer 图,颜色模式为 GRW8

PixelFormat_Mono16=0x01100007, //!<16Bit 灰度

```
PixelFormat_BayerGR16=0x0110002E,    //!<16Bit Bayer 图,颜色模式为 GR
PixelFormat_BayerRG16=0x0110002F, //!<16Bit Bayer 图,颜色模式为 RG
PixelFormat_BayerGB16=0x01100030,    //!<16Bit Bayer 图,颜色模式为 GB
PixelFormat_BayerBG16=0x01100031 //!<16Bit Bayer 图,颜色模式为 BG
}
```

- enum MVSTATUS_CODES{

MVST_SUCCESS	= 0,	///< 没有错误
MVST_ERROR	= -1001,	///< 一般错误
MVST_ERR_NOT_INITIALIZED	= -1002,	///< 没有初始化
MVST_ERR_NOT_IMPLEMENTED	= -1003,	///< 没有实现
MVST_ERR_RESOURCE_IN_USE	= -1004,	///< 资源被占用
MVST_ACCESS_DENIED	= -1005,	///< 无法访问
MVST_INVALID_HANDLE	= -1006,	///< 错误句柄
MVST_INVALID_ID	= -1007,	///< 错误 ID
MVST_NO_DATA	= -1008,	///< 没有数据
MVST_INVALID_PARAMETER	= -1009,	///< 错误参数
MVST_FILE_IO	= -1010,	///< IO 错误
MVST_TIMEOUT	= -1011,	///< 超时
MVST_ERR_ABORT	= -1012,	///< 退出
MVST_INVALID_BUFFER_SIZE	= -1013,	///< 缓冲区尺寸错误
MVST_ERR_NOT_AVAILABLE	= -1014,	///< 无法访问
MVST_INVALID_ADDRESS	= -1015,	///< 地址错误

 }
- enum TriggerSourceEnums{

TriggerSource_Software=0,	///< 触发模式下, 由软触发(软件指令)来触发采集
TriggerSource_Line1=2	///< 触发模式下, 有外触发信号来触发采集

- ```
}

● enum TriggerModeEnums{
 TriggerMode_Off, //!<触发模式关，即 FreeRun 模式，相机连续采集
 TriggerMode_On //!<触发模式开，相机等待软触发或外触发信号再采集图像
}

● enum TriggerActivationEnums{
 TriggerActivation_RisingEdge, //!<上升沿触发
 TriggerActivation_FallingEdge //!<下降沿触发
}

● enum LineSourceEnums{
 LineSource_Off=0, //!<关闭
 LineSource_ExposureActive=5, //!<和曝光同时
 LineSource_Timer1Active=6, //!<由定时器控制
 LineSource_UserOutput0=12 //!<直接由软件控制
}

● enum UserSetSelectorEnums{
 UserSetSelector_Default, //!<出厂设置
 UserSetSelector_UserSet1, //!<用户设置1
 UserSetSelector_UserSet2 //!<用户设置2
}

● enum SensorTapsEnums{
 SensorTaps_One, //!<单通道
 SensorTaps_Two, //!<双通道
 SensorTaps_Three, //!<三通道
 SensorTaps_Four, //!<四通道
}
```

- enum AutoFunctionProfileEnums{
  - AutoFunctionProfile\_GainMinimum,     ///<保持增益为最小值
  - AutoFunctionProfile\_ExposureNinimum ///<保持曝光为最小值
 }
- enum GainAutoEnums{
  - GainAuto\_Off,             ///<自动增益关闭
  - GainAuto\_Once,          ///<一次自动增益
  - GainAuto\_Continuous    ///<连续自动增益
 }
- enum ExposureAutoEnums{
  - ExposureAuto\_Off,        ///<自动曝光关闭
  - ExposureAuto\_Once,       ///<一次自动曝光
  - ExposureAuto\_Continuous ///<连续自动曝光
 }
- enum BalanceWhiteAutoEnums{
  - BalanceWhiteAuto\_Off,     ///<自动白平衡关闭
  - BalanceWhiteAuto\_Once,    ///<一次自动白平衡
  - BalanceWhiteAuto\_Continuous ///<连续自动白平衡
 }
- enum ImageFlipType {
  - FlipHorizontal = 0,    ///< 左右翻转
  - FlipVertical = 1, ///< 上下翻转
  - FlipBoth = 2 ///< 旋转180度
 }
- enum ImageRotateType{
  - Rotate90DegCw=0,     ///<顺时针转90度

```
Rotate90DegCcw=1 //!<逆时针转90度
}
```

- enum EventIdEnums{  
EVID\_LOST = 0, //!< 事件 ID, 相机断开  
EVID\_RECONNECT = 1 //!< 事件 ID, 相机重新连上了

### 2.2.1、MVAvi.h

```
#include <mmreg.h>
#include <vfw.h>
#include "MVImage.h"
```

结构体

```
class MVAvi
```

宏定义

```
#define MVGIGE_API __declspec(dllimport)
```

### 2.2.2、MVCamProptySheet.h

```
#include "GigECamera_Types.h"
```

宏定义

```
#define MVPROPERTY_API __declspec(dllimport)
#define PAGE_NONE 0x0000
#define PAGE_ALL 0xffff
```



```
#define PAGE_ACQUISITION 0x0001

#define PAGE_WHITE_BALANCE 0x0002

#define PAGE_TRANS_LAYER 0x0004

#define PAGE_TRIGGER 0x0008

#define PAGE_CAMERA_INFO 0x0010

#define PAGE_IMAGE_FORMAT 0x0020

#define PAGE_AUTOGE_CONTROL 0x0040

#define PAGE_USERSET 0x0080
```

枚举

```
enum MVCameraRunEnums { MVCameraRun_ON, MVCameraRun_OFF}
```

函数

- MVPROPERTY\_API MVSTATUS\_CODES \_\_stdcall MVCamProptySheetInit( HANDLE\* phProptySheet, HANDLE hCam, CWnd\* pParentWnd = NULL, LPCTSTR lpszTitle = NULL, WORD nPageDisplay = PAGE\_ALL );

功能：创建相机属性页

参数说明：

phProptySheet 返回相机属性页句柄

hCam 相机句柄

pParentWnd 父窗口句柄,类型 CWnd\*

lpszTitle 相机属性页标题栏文字

nPageDisplay 相机属性页风格

返回值：成功返回 MVST\_SUCCESS,否则返回错误信息

- MVPROPERTY\_API MVSTATUS\_CODES \_\_stdcall MVCamProptySheetCreate( HANDLE\* phProptySheet, HANDLE hCam, CWnd\* pParentWnd = NULL, LPCTSTR lpszTitle = NULL, WORD nPageDisplay = PAGE\_ALL );

功能：创建相机属性页

参数说明：

phProptySheet 返回相机属性页句柄

hCam 相机句柄

pParentWnd 父窗口句柄,类型 CWnd\*

lpszTitle 相机属性页标题栏文字

nPageDisplay 相机属性页风格

返回值：成功返回 MVST\_SUCCESS,否则返回错误信息

- MVPROPERTY\_API MVSTATUS\_CODES \_\_stdcall  
MVCamProptySheetInitEx(HANDLE\* phProptySheet, HANDLE hCam, HWND  
pParentWnd = NULL, LPCTSTR lpszTitle = NULL, WORD nPageDisplay = PAGE\_ALL);

功能：创建相机属性页

参数说明：

phProptySheet 返回相机属性页句柄

hCam 相机句柄

pParentWnd 父窗口句柄,类型 HWND

lpszTitle 相机属性页标题栏文字

nPageDisplay 相机属性页风格

返回值：成功返回 MVST\_SUCCESS,否则返回错误信息

- MVPROPERTY\_API MVSTATUS\_CODES \_\_stdcall  
MVCamProptySheetCreateEx(HANDLE\* phProptySheet, HANDLE hCam, HWND  
pParentWnd = NULL, LPCTSTR lpszTitle = NULL, WORD nPageDisplay = PAGE\_ALL);

功能：创建相机属性页

参数说明：

phProptySheet 返回相机属性页句柄

hCam 相机句柄

pParentWnd 父窗口句柄,类型 HWND

lpszTitle 相机属性页标题栏文字

nPageDisplay 相机属性页风格

返回值：成功返回 MVST\_SUCCESS,否则返回错误信息

- MVPROPERTY\_API MVSTATUS\_CODES \_\_stdcall  
MVCamProptySheetDestroy(HANDLE hProptySheet);

功能：销毁相机属性页

参数说明：

hProptySheet 相机属性页句柄

返回值：成功返回 MVST\_SUCCESS,否则返回错误信息

- MVPROPERTY\_API MVSTATUS\_CODES \_\_stdcall  
MVCamProptySheetSetTitle(HANDLE hProptySheet, LPCTSTR lpszTitle = NULL);

功能：设置相机属性页标题栏文字

参数说明：

hProptySheet 相机属性页句柄

lpszTitle 相机属性页标题栏文字

返回值：成功返回 MVST\_SUCCESS,否则返回错误信息

- MVPROPERTY\_API MVSTATUS\_CODES \_\_stdcall  
MVCamProptySheetSetCamera(HANDLE hProptySheet, HANDLE hCam);

功能：设置相机属性页对应的相机

参数说明：

hProptySheet 相机属性页句柄

phCam 相机句柄

返回值：成功返回 MVST\_SUCCESS,否则返回错误信息

- MVPROPERTY\_API MVSTATUS\_CODES \_\_stdcall  
MVCamProptySheetGetCamera(HANDLE hProptySheet, HANDLE \*phCam);

功能：获取相机属性页当前对应的相机

参数说明：

hProptySheet 相机属性页句柄

phCam 相机句柄

返回值：成功返回 MVST\_SUCCESS,否则返回错误信息

- MVPROPERTY\_API MVSTATUS\_CODES \_\_stdcall  
MVCamProptySheetCameraRun( HANDLE hProptySheet, MVCameraRunEnums  
Run );

功能：设置相机现在是否正工作在采集模式

参数说明：

hProptySheet 相机属性页句柄

Run 如果相机正工作在采集模式下，设置为 TRUE,否则设置为 FALSE

返回值：成功返回 MVST\_SUCCESS,否则返回错误信息

- MVPROPERTY\_API MVSTATUS\_CODES \_\_stdcall MVCamProptySheetShow(HANDLE  
hProptySheet, int nCmdShow = SW\_SHOW);

功能：以非模式框方式显示或关闭相机属性页

参数说明：

hProptySheet 相机属性页句柄

nCmdShow SW\_SHOW:显示， SW\_HIDE:关闭

返回值：成功返回 MVST\_SUCCESS,否则返回错误信息

- MVPROPERTY\_API MVSTATUS\_CODES \_\_stdcall  
MVCamProptySheetDoModal(HANDLE hProptySheet);

功能：以模式框方式显示相机属性页

参数说明：

hProptySheet 相机属性页句柄

返回值：成功返回 MVST\_SUCCESS,否则返回错误信息

- MVPROPERTY\_API MVSTATUS\_CODES \_\_stdcall  
MVCamProptySheetSetParent(HANDLE hProptySheet, HANDLE hParent);

功能：设置属性页的父窗口

参数说明：

hProptySheet 相机属性页句柄

hParent 父窗口句柄,类型 HWND

返回值：成功返回 MVST\_SUCCESS,否则返回错误信息

- MVPROPERTY\_API MVSTATUS\_CODES \_\_stdcall  
MVCamProptySheetInsertPage(HANDLE hProptySheet, WORD nPageInsert);

功能：插入属性页

参数说明：

hProptySheet 相机属性页句柄

nPageInsert 属性页标识, 可以以或的形式插入多页

返回值：成功返回 MVST\_SUCCESS,否则返回错误信息

- MVPROPERTY\_API MVSTATUS\_CODES \_\_stdcall  
MVCamProptySheetDeletePage(HANDLE hProptySheet, WORD nPageDelete);

功能：删除属性页

参数说明：

hProptySheet 相机属性页句柄

nPageDelete 属性页标识, 可以以或的形式删除多页

返回值：成功返回 MVST\_SUCCESS,否则返回错误信息

MVGigE.h

```
#include "GigECamera_Types.h"
```

```
#include "MVImage.h"
```

宏定义

```
#define MSG_ID_LOST 0x1 //消息 ID , 相机断开
```

```
#define MSG_ID_RECONNECT 0x2 0x2 //消息 ID , 相机重新连上了
```

函数

- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVFilterAdaptiveMedian(HANDLE  
hCam,MVImage \*pImage,int th);

功能：自适应中值滤波，用于去除长时间曝光暗电流造成的图像中亮点，目前仅适用于8Bit 黑白图像

参数说明:

hCam 相机句柄

pImage 图像

th 阈值，和周围差异大于此阈值的被认为是噪点

返回值： 成功返回 MVST\_SUCCESS,否则返回错误信息

- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVOpenCamByIndexReadOnly(unsigned char idx,HANDLE \*hCam);

功能：以只读方式打开相机

参数说明:

idx idx 从0开始，按照相机的 IP 地址排序，地址小的排在前面。

hCam 如果成功,返回的相机句柄

返回值： 成功返回 MVST\_SUCCESS,否则返回错误信息

- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVSetUserDefinedName(HANDLE hCam, char \*pBuf,int szBuf );

功能：设置自定义名称

参数说明:

hCam 相机句柄

pBuf 相机名称缓冲区

szBuf 相机名称缓冲区长度指针

返回值： 成功返回 MVST\_SUCCESS,否则返回错误信息

- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVGetUserDefinedName(HANDLE hCam, char \*pBuf,int \*szBuf );

功能：获取自定义名称

参数说明:

hCam 相机句柄

pBuf 相机名称缓冲区

szBuf 相机名称缓冲区长度指针

返回值： 成功返回 MVST\_SUCCESS,否则返回错误信息

- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVEnableMessage(HANDLE hCam, int nMessageType, bool bEnable);

功能：是否允许发送某个消息

参数说明:

hCam 相机句柄

nMessageType 消息类型, MSG\_ID\_LOST,MSG\_ID\_RECONNECT

bEnable 如果为 TRUE,则发送该消息, 为 FALSE 则不发送该消息

返回值： 成功返回 MVST\_SUCCESS,否则返回错误信息

- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVRegisterMessage(HANDLE hCam, HWND hWnd, UINT nMsg);

功能：注册用于接收消息的窗口句柄和消息值。当相机断开或重新连上时会发送消息到该窗口。

参数说明:

hCam 相机句柄

hWnd 用于接收消息的窗口句柄

nMsg 消息值

返回值： 成功返回 MVST\_SUCCESS,否则返回错误信息

- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVGetColorCorrect(HANDLE hCam, int \*nColorCorrect );

功能：获取当前颜色校正模式

参数说明:

hCam 相机句柄

nColorCorrect 颜色校正模式指针

返回值： 成功返回 MVST\_SUCCESS,否则返回错误信息

- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVSetColorCorrect(HANDLE hCam, int

nColorCorrect );

功能：颜色校正

参数说明:

hCam 相机句柄

nColorCorrect, 颜色校正模式，目前仅支持0和1,0为不校正，1为校正

返回值： 成功返回 MVST\_SUCCESS,否则返回错误信息

- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVGetSaturation(HANDLE hCam, int \*nSaturation );

功能：获取当前饱和度

参数说明:

hCam 相机句柄

nSaturation 饱和度指针

返回值： 成功返回 MVST\_SUCCESS,否则返回错误信息

- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVSetSaturation(HANDLE hCam, int nSaturation );

功能：调节饱和度

参数说明:

hCam 相机句柄

nSaturation 饱和度,范围-100到100, -100为黑白，0为原图,100为最鲜艳

返回值： 成功返回 MVST\_SUCCESS,否则返回错误信息

- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVCopyImageInfoROI( HANDLE hCam, MV\_IMAGE\_INFO\* pInfoSrc, MV\_IMAGE\_INFO\* pInfoDst, RECT roi );

功能：直接从回调函数传回的图像信息中裁剪出图像的一部分，当相机不支持硬件 ROI 时，可以用此函数实现软件 ROI。

参数说明:

hCam 相机句柄

pInfoSrc 源图像指针，一般是回调函数传回的图像信息指针



pInfoDst 目标图像指针，nPixelFormat 要和源图像的相同，需要提前分配好内存。并且宽度和高度要和 roi 的宽高相同。

roi 感兴趣的区域,roi 的 left,right,top,bottom 均须是2的整倍数

返回值： 成功返回 MVST\_SUCCESS,否则返回错误信息

- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVConvertImage( HANDLE hCam, MImage\* pImageSrc,MImage\* pImageDst );

功能：图像格式转换

参数说明:

hCam 相机句柄

pImageSrc 源图像指针

pImageDst 目标图像指针

返回值： 成功返回 MVST\_SUCCESS,否则返回错误信息

- MVGIGE\_API bool \_\_stdcall MVIsRunning(HANDLE hCam);

功能：相机是否正在采集图像

参数说明:

hCam 相机句柄

返回值： 成功返回 MVST\_SUCCESS,否则返回错误信息

- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVGetDeviceDeviceID(HANDLE hCam,char \*pBuf,int \*szBuf);

功能：获取设备的 ID 号，即序列号

参数说明:

hCam 相机句柄

pBuf 用于保存序列号的缓冲区，大于等于16字节

szBuf 缓冲区大小

返回值： 成功返回 MVST\_SUCCESS,否则返回错误信息

- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVGetDeviceModelName(HANDLE hCam,char \*pBuf,int \*szBuf);

功能：获取设备的型号

参数说明:

hCam 相机句柄

pBuf 用于保存型号的缓冲区，大于等于32字节

szBuf 缓冲区大小

返回值： 成功返回 MVST\_SUCCESS,否则返回错误信息

- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVGetDeviceVendorName(HANDLE hCam,char \*pBuf,int \*szBuf);

功能：获取设备厂商名称

参数说明:

hCam 相机句柄

pBuf 用于保存名称的缓冲区，大于等于32字节

szBuf 缓冲区大小

返回值： 成功返回 MVST\_SUCCESS,否则返回错误信息

- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVGetDroppedFrame(HANDLE hCam,unsigned long \*pDroppedFrames);

功能：当计算机收到新的图像，而上一帧的 Callback 函数还没有执行完，SDK 中会扔掉新的一帧图像。此函数可以获取扔掉的帧数。

参数说明:

hCam 相机句柄

pDroppedFrames 扔掉的帧数

返回值： 成功返回 MVST\_SUCCESS,否则返回错误信息

- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVGetSampleGrab(HANDLE hCam, MVImage\* image, int\* nFrameID);

功能：当调用 MVFreezeGrabWindow(TRUE)后，调用此函数可以获取当前图像。

参数说明:

hCam 相机句柄

image 图像

nFrameID 图像的 ID 号

返回值： 成功返回 MVST\_SUCCESS,否则返回错误信息

- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVSetGrabWindow(HANDLE hCam, int xDest, int yDest, int wDest, int hDest, int xSrc, int ySrc, int wSrc, int hSrc);

功能：当采集到窗口时，设置图像显示的区域和比例。

参数说明:

hCam 相机句柄

xDest 指定显示窗口中目标矩形左上角的逻辑 X 坐标

yDest 指定显示窗口中目标矩形左上角的逻辑 Y 坐标

wDest 指定显示窗口中目标矩形的宽度

hDest 指定显示窗口中目标矩形的高度

xSrc 指定图像源位图左上角的逻辑 X 坐标

ySrc 指定图像源位图左上角的逻辑 Y 坐标

wSrc 指定图像源位图宽度

hSrc 指定图像源位图高度

返回值： 成功返回 MVST\_SUCCESS,否则返回错误信息

- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVFreezeGrabWindow(HANDLE hCam, bool bFreeze);

功能：当采集到窗口时，暂停或继续采集。

参数说明:

hCam 相机句柄

bFreeze

返回值： 成功返回 MVST\_SUCCESS,否则返回错误信息

- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVStopGrabWindow(HANDLE hCam);

功能：停止采集到窗口

参数说明:

hCam 相机句柄

返回值： 成功返回 MVST\_SUCCESS,否则返回错误信息

- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVStartGrabWindow(HANDLE hCam, HWND hWnd, HWND hWndMsg);

功能：开始采集，并将采集到的图像显示到指定窗口

参数说明:

hCam 相机句柄

hWnd 窗口句柄

hWndMsg 消息句柄 ,如果不为 NULL,当新的图像采集完毕,会发送消息(WM\_USER+0x0200)到此窗口

返回值： 成功返回 MVST\_SUCCESS,否则返回错误信息

- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVSinglGrab(HANDLE hCam, HANDLE hImage, unsigned long nWaitMs);

功能：采集一帧图像。

参数说明:

hCam 相机句柄

hImage 图像句柄。保存采集到的图像。

nWaitMs 等待多长时间，单位 ms

返回值： 成功返回 MVST\_SUCCESS,否则返回错误信息

- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVGetEnableLUT(HANDLE hCam, bool\* bEnable);

功能：获取当前是否使用查找表状态

参数说明:

hCam

bEnable 当前是否使用查找表状态

返回值： 成功返回 MVST\_SUCCESS,否则返回错误信息

- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVSetEnableLUT(HANDLE hCam, bool

bEnable);

功能：使用查找表

参数说明:

hCam 相机句柄

bEnable 是否允许

返回值： 成功返回 MVST\_SUCCESS,否则返回错误信息

- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVSetLUT(HANDLE hCam, unsigned long\* pLUT, int nCnt);

功能：设置查找表

参数说明:

hCam 相机句柄

pLUT 查找表数组, unsigned long pLUT[1024];

nCnt 查找表数组单元个数, 必须是1024

返回值： 成功返回 MVST\_SUCCESS,否则返回错误信息

- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVSetGamma(HANDLE hCam, double fGamma);

功能：设置伽马值

参数说明:

hCam 相机句柄

fGamma伽马值

返回值： 成功返回 MVST\_SUCCESS,否则返回错误信息

- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVGetGammaRange(HANDLE hCam, double\* pGammaMin, double\* pGammaMax);

功能：获取伽马值可设置的范围

参数说明:

hCam 相机句柄

pGammaMin 伽马最小值

pGammaMax 伽马最大值

返回值： 成功返回 MVST\_SUCCESS,否则返回错误信息

- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVGetGamma(HANDLE hCam, double\* pGamma);

功能：获取当前伽马值

参数说明:

hCam 相机句柄

pGamma 当前伽马值

返回值： 成功返回 MVST\_SUCCESS,否则返回错误信息

- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVSetAutoThreshold(HANDLE hCam, int nAutoThreshold);

功能：自动增益或自动曝光时，图像亮度与目标亮度差异的容差。

参数说明:

hCam 相机句柄

nAutoThreshold 图像亮度与目标亮度差异的容差

返回值： 成功返回 MVST\_SUCCESS,否则返回错误信息

- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVGetAutoThreshold(HANDLE hCam, int\* pAutoThreshold);

功能：自动增益或自动曝光时，图像亮度与目标亮度差异的容差。

参数说明:

hCam 相机句柄

pAutoThreshold 图像亮度与目标亮度差异的容差

返回值： 成功返回 MVST\_SUCCESS,否则返回错误信息

- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVSetAutoFunctionProfile(HANDLE hCam, AutoFunctionProfileEnums AutoFunctionProfile);

功能：当自动增益和自动曝光时间都打开时，设置哪一个值优先调整

参数说明:

hCam 相机句柄

AutoFunctionProfile 增益优先或曝光时间优先

返回值： 成功返回 MVST\_SUCCESS,否则返回错误信息

- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVGetAutoFunctionProfile(HANDLE hCam, AutoFunctionProfileEnums\* pAutoFunctionProfile);

功能：当自动增益和自动曝光时间都打开时，获取哪一个值优先调整

参数说明:

hCam 相机句柄

pAutoFunctionProfile 增益优先或曝光时间优先

返回值： 成功返回 MVST\_SUCCESS,否则返回错误信息

- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVSetAutoTargetValue(HANDLE hCam, int nAutoTargetValue);

功能：设置自动调整亮度(曝光、增益)时，期望调整到的图像亮度

参数说明:

hCam 相机句柄

nAutoTargetValue 期望调整到的图像亮度

返回值： 成功返回 MVST\_SUCCESS,否则返回错误信息

- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVGetAutoTargetValue(HANDLE hCam, int\* pAutoTargetValue);

功能：获取自动调整亮度(曝光、增益)时，期望调整到的图像亮度

参数说明:

hCam 相机句柄

pAutoTargetValue 期望调整到的图像亮度

返回值： 成功返回 MVST\_SUCCESS,否则返回错误信息

- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVSetAutoExposureTimeUpperLimit(HANDLE hCam, double fAutoExposureTimeUpperLimit);

功能：设置自动调整曝光时间时，曝光时间调整范围的最大值

参数说明:

hCam 相机句柄

fAutoExposureTimeUpperLimit

返回值：成功返回 MVST\_SUCCESS,否则返回错误信息

- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall  
MVGetAutoExposureTimeUpperLimit(HANDLE hCam, double\*  
pAutoExposureTimeUpperLimit);

功能：获取自动调整曝光时间时，曝光时间调整范围的最大值

参数说明:

hCam 相机句柄

pAutoExposureTimeUpperLimit 曝光时间调整范围的最大值

返回值：成功返回 MVST\_SUCCESS,否则返回错误信息

- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall  
MVSetAutoExposureTimeLowerLimit(HANDLE hCam, double  
fAutoExposureTimeLowerLimit);

功能：设置自动调整曝光时间时，曝光时间调整范围的最小值

参数说明:

hCam 相机句柄

fAutoExposureTimeLowerLimit 曝光时间调整范围的最小值

返回值：成功返回 MVST\_SUCCESS,否则返回错误信息

- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall  
MVGetAutoExposureTimeLowerLimit(HANDLE hCam, double\*  
pAutoExposureTimeLowerLimit);

功能：获取自动调整曝光时间时，曝光时间调整范围的最小值

参数说明:

hCam 相机句柄

pAutoExposureTimeLowerLimit 曝光时间调整范围的最小值



返回值： 成功返回 MVST\_SUCCESS,否则返回错误信息

- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVSetAutoGainUpperLimit(HANDLE hCam, double fAutoGainUpperLimit);

功能：设置自动调整增益时，增益调整范围的最大值

参数说明:

hCam 相机句柄

fAutoGainUpperLimit 曝光时间调整范围的最小值

返回值： 成功返回 MVST\_SUCCESS,否则返回错误信息

- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVGetAutoGainUpperLimit(HANDLE hCam, double\* pAutoGainUpperLimit);

功能：获取自动调整增益时，增益调整范围的最大值

参数说明:

hCam 相机句柄

pAutoGainUpperLimit 增益调整范围的最大值

返回值： 成功返回 MVST\_SUCCESS,否则返回错误信息

- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVSetAutoGainLowerLimit(HANDLE hCam, double fAutoGainLowerLimit);

功能：设置自动调整增益时，增益调整范围的最小值

参数说明:

hCam 相机句柄

fAutoGainLowerLimit增益调整范围的最小值

返回值： 成功返回 MVST\_SUCCESS,否则返回错误信息

- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVGetAutoGainLowerLimit(HANDLE hCam, double\* pAutoGainLowerLimit);

功能：获取自动调整增益时，增益调整范围的最小值

参数说明:

hCam 相机句柄

pAutoGainLowerLimit 增益调整范围的最小值

返回值： 成功返回 MVST\_SUCCESS,否则返回错误信息

- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVSetBalanceWhiteAuto(HANDLE hCam, BalanceWhiteAutoEnums BalanceWhiteAuto);

功能：设置自动白平衡模式

参数说明:

hCam 相机句柄

BalanceWhiteAuto 自动白平衡模式

返回值： 成功返回 MVST\_SUCCESS,否则返回错误信息

- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVGetBalanceWhiteAuto(HANDLE hCam, BalanceWhiteAutoEnums\* pBalanceWhiteAuto);

功能：获取当前自动白平衡模式

参数说明:

hCam 相机句柄

pBalanceWhiteAuto 当前自动白平衡模式

返回值： 成功返回 MVST\_SUCCESS,否则返回错误信息

- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVSetGainAuto(HANDLE hCam, GainAutoEnums GainAuto);

功能：设置当前自动增益模式

参数说明:

hCam 相机句柄

GainAuto 自动增益模式

返回值： 成功返回 MVST\_SUCCESS,否则返回错误信息

- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVGetGainAuto(HANDLE hCam, GainAutoEnums\* pGainAuto);

功能：获取当前自动增益模式

参数说明:

hCam 相机句柄

pGainAuto 当前自动增益模式的

返回值： 成功返回 MVST\_SUCCESS,否则返回错误信息

- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVSetExposureAuto(HANDLE hCam, ExposureAutoEnums ExposureAuto);

功能：设置自动曝光模式

参数说明:

hCam 相机句柄

ExposureAuto 自动曝光模式

返回值： 成功返回 MVST\_SUCCESS,否则返回错误信息

- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVGetExposureAuto(HANDLE hCam, ExposureAutoEnums\* pExposureAuto);

功能：获取当前自动曝光模式

参数说明:

hCam 相机句柄

pExposureAuto 当前自动曝光模式

返回值： 成功返回 MVST\_SUCCESS,否则返回错误信息

- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVGrayToBGR(HANDLE hCam, unsigned char\* pSrc, unsigned char\* pDst, int width, int height);

功能：将灰度单通道8bit 图像转换为彩色 BGR 三通道24bit 图像。转换后三个通道的值是相同的。

参数说明:

hCam 相机句柄

pSrc灰度单通道8bit 图像指针

pDst 彩色 BGR 三通道24bit 图像指针

width 图像宽度

height 图像高度

返回值： 成功返回 MVST\_SUCCESS,否则返回错误信息

- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVImageBGRToYUV(HANDLE hCam, MVImage\* pSrcImage, unsigned char\* pDst);

功能：将彩色 BGR 三通道24bit 图像转换为 YUV 图像

参数说明:

hCam 相机句柄

pSrcImage 彩色 BGR 三通道24bit 图像指针

pDst YUV 图像指针 (YUV422)

返回值： 成功返回 MVST\_SUCCESS,否则返回错误信息

- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVImageBGRToGray(HANDLE hCam, MVImage\* pSrcImage, MVImage\* pDstImage);

功能：将彩色 BGR 三通道24bit 图像转换为灰度单通道8bit 图像

参数说明:

hCam 相机句柄

pSrcImage 彩色 BGR 三通道24bit 图像指针

pDstImage 灰度单通道8bit 图像指针。宽度高度必须和源图相同

返回值： 成功返回 MVST\_SUCCESS,否则返回错误信息

- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVBGRToGray(HANDLE hCam, unsigned char\* psrc, unsigned char\* pdst, unsigned int width, unsigned int height);

功能：将彩色 BGR 三通道24bit 图像转换为灰度单通道8bit 图像

参数说明:

hCam 相机句柄

psrc彩色 BGR 三通道24bit 图像指针

pdst 灰度单通道8bit 图像指针

width 图像宽度

height 图像高度

返回值： 成功返回 MVST\_SUCCESS,否则返回错误信息

- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVImageRotate(HANDLE hCam, MVImage\* pSrcImage, MVImage\* pDstImage, ImageRotateType roateType);

功能：图像旋转

参数说明:

hCam 相机句柄

pSrcImage 源图像指针

pDstImage 结果图像指针,不能为 NULL。结果图像的宽度和高度应该和源图像的宽度和高度互换。

roateType 旋转类型：Rotate90DegCw, Rotate90DegCcw

返回值： 成功返回 MVST\_SUCCESS,否则返回错误信息

- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVImageFlip(HANDLE hCam, MVImage\* pSrcImage, MVImage\* pDstImage, ImageFlipType flipType);

功能：图像翻转

参数说明:

hCam 相机句柄

pSrcImage 源图像指针

pDstImage 结果图像指针。如果为 NULL，则翻转的结果还在源图像内。

flipType 翻转类型。FlipHorizontal:左右翻转,FlipVertical:上下翻转,FlipBoth:旋转180度

返回值： 成功返回 MVST\_SUCCESS,否则返回错误信息

- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVGetDefaultUserSet(HANDLE hCam, UserSetSelectorEnums\* pUserset);

功能：读取相机上电开机时默认读取并应用哪一组用户设置

参数说明:

hCam 相机句柄

pUserset 用户设置

返回值： 成功返回 MVST\_SUCCESS,否则返回错误信息

- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVSetDefaultUserSet(HANDLE hCam,

UserSetSelectorEnums userset);

功能：设置相机上电开机时默认读取并应用哪一组用户设置

参数说明:

hCam        相机句柄

userset

返回值： 成功返回 MVST\_SUCCESS,否则返回错误信息

- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVSaveUserSet(HANDLE hCam, UserSetSelectorEnums userset);

功能：将当前相机的参数保存到用户设置中。

参数说明:

hCam        相机句柄

userset

返回值： 成功返回 MVST\_SUCCESS,否则返回错误信息

- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVLoadUserSet(HANDLE hCam, UserSetSelectorEnums userset);

功能：读取并应用某组用户预设的参数

参数说明:

hCam        相机句柄

userset

返回值： 成功返回 MVST\_SUCCESS,否则返回错误信息

- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVGetStreamStatistic(HANDLE hCam, MVStreamStatistic\* pStatistic);

功能：获取数据传输的统计信息

参数说明:

hCam    相机句柄

pStatistic    统计信息

返回值： 成功返回 MVST\_SUCCESS,否则返回错误信息

- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVZoomImageBGR(HANDLE hCam, unsigned char\* pSrc, int srcWidth, int srcHeight, unsigned char\* pDst, double fFactorX, double fFactorY);

功能：BGR 格式三通道图像缩放

参数说明:

hCam 相机句柄

pSrc 源图像指针

srcWidth 源图像宽度

srcHeight 源图像高度

pDst 缩放后图像指针

fFactorX 缩放比例

fFactorY 缩放比例

返回值：成功返回 MVST\_SUCCESS,否则返回错误信息

- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVImageBayerToBGREx( HANDLE hCam,MV\_IMAGE\_INFO \*pInfo,MVImage \*pImage,double fGamma,bool bColorCorrect,int nContrast );

功能 将 Bayer 格式的8bit单通道图转换为 BGR 格式的8Bit 三通道图,同时调整图像的 GAMMA, 颜色和反差

参数说明:

hCam 相机句柄

pInfo 采集 Callback 函数中传来的图像信息指针

pImage 转换结果图像的指针

fGamma Gamma 校正值，1为不校正，<1时，将暗部提升。

bColorCorrect 是否进行颜色校正，进行颜色校正后，图像会变得更鲜艳。

nContrast 是否调整反差，范围为0 - 50，当该值大于0时，图像反差会更强。

返回值：成功返回 MVST\_SUCCESS,否则返回错误信息

- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVInfo2Image(HANDLE hCam, MV\_IMAGE\_INFO\* pInfo, MVImage\* pImage);

功能：将回调函数收到的图像信息转换为图像。

参数说明:

hCam 相机句柄

pInfo 采集 Callback 函数中传来的图像信息指针

pImage 转换结果图像的指针

返回值： 成功返回 MVST\_SUCCESS,否则返回错误信息

- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVImageBayerToBGR(HANDLE hCam, MV\_IMAGE\_INFO\* pInfo, MVImage\* pImage);

功能：将 Bayer 格式的8bit 单通道图转换为 BGR 格式的8Bit 三通道图

参数说明:

hCam 相机句柄

pInfo 采集 Callback 函数中传来的图像信息指针

pImage 转换结果图像的指针

返回值： 成功返回 MVST\_SUCCESS,否则返回错误信息

- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVBayerToRGB16(HANDLE hCam, void \*psrc,void \*pdst,unsigned int dststep,unsigned int width,unsigned int height,MV\_PixelFormatEnums pixelformat );

功能：将 Bayer 格式的16bit 单通道图转换为 RGB 格式的16Bit 三通道图

参数说明:

hCam 相机句柄

psrc 单通道图像的指针

pdst 三通道图像指针

dststep 三通道图像一行图像的字节数。

width 图像宽度

height 图像高度

pixelformat 像素格式，由 MVGetPixelFormat 取得

返回值： 成功返回 MVST\_SUCCESS,否则返回错误信息



- `MVGIGE_API MVSTATUS_CODES __stdcall MVBayerToRGB(HANDLE hCam, void *psrc,void *pdst,unsigned int dststep,unsigned int width,unsigned int height,MV_PixelFormatEnums pixelformat,bool bMultiCores=FALSE);`

功能：将 Bayer 格式的8bit 单通道图转换为 RGB 格式的8Bit 三通道图

参数说明:

hCam 相机句柄

psrc 单通道图像的指针

pdst 三通道图像指针

dststep 三通道图像一行图像的字节数。通常为图像宽度\*3，但是会为了4字节对齐会补几个字节。

width 图像宽度

height 图像高度

pixelformat 像素格式，由 MVGetPixelFormat 取得

bMultiCores 是否使用 CPU 多核计算

返回值：成功返回 MVST\_SUCCESS,否则返回错误信息

- `MVGIGE_API MVSTATUS_CODES __stdcall MVBayerToBGR16(HANDLE hCam, void *psrc,void *pdst,unsigned int dststep,unsigned int width,unsigned int height,MV_PixelFormatEnums pixelformat );`

功能：将 Bayer 格式的16bit 单通道图转换为 BGR 格式的16Bit 三通道图

参数说明:

hCam 相机句柄

psrc 单通道图像的指针

pdst 三通道图像指针

dststep 三通道图像一行图像的字节数。

width 图像宽度

height 图像高度

pixelformat 像素格式，由 MVGetPixelFormat 取得

返回值： 成功返回 MVST\_SUCCESS,否则返回错误信息

- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVBayerToBGR(HANDLE hCam, void \*psrc,void \*pdst,unsigned int dststep,unsigned int width,unsigned int height,MV\_PixelFormatEnums pixelformat,bool bMultiCores=FALSE);

功能：将 Bayer 格式的8bit 单通道图转换为 BGR 格式的8Bit 三通道图

参数说明:

hCam 相机句柄

psrc 单通道图像的指针

pdst 三通道图像指针

dststep 三通道图像一行图像的字节数。通常为图像宽度\*3，但是会为了4字节对齐会补几个字节。

width 图像宽度

height 图像高度

pixelformat 像素格式，由 MVGetPixelFormat 取得

bMultiCores 是否使用 CPU 多核计算

返回值： 成功返回 MVST\_SUCCESS,否则返回错误信息

- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVSetTimerDuration(HANDLE hCam, uint32\_t nDuration);

功能：当闪光同步源选为 Timer1时 MVSetStrobeSource(hCam,LineSource\_Timer1Active)

参数说明:

hCam

nDuration 设置 Timer1在开始计时后，计时多长时间(us)。即输出高/低电平的脉冲宽度。

返回值： 成功返回 MVST\_SUCCESS,否则返回错误信息

- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVGetTimerDurationRange(HANDLE hCam, uint32\_t\* pMin, uint32\_t\* pMax);

功能：读取定时器计时时长取值范围

参数说明:

hCam        相机句柄

pMin    定时器计时时长最小值

pMax    定时器计时时长最大值

返回值： 成功返回 MVST\_SUCCESS,否则返回错误信息

- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVGetTimerDuration(HANDLE hCam, uint32\_t\* pDuration);

功能：读取定时器计时时长

参数说明:

hCam        相机句柄

pDuration    定时器计时时长

返回值： 成功返回 MVST\_SUCCESS,否则返回错误信息

- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVSetTimerDelay(HANDLE hCam, uint32\_t nDelay);

功能：当闪光同步源选为 Timer1时 MVSetStrobeSource(hCam,LineSource\_Timer1Active)

参数说明:

hCam        相机句柄

nDelay 接到触发信号后延迟多少 us 开始计时

返回值： 成功返回 MVST\_SUCCESS,否则返回错误信息

- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVGetTimerDelayRange(HANDLE hCam, uint32\_t\* pMin, uint32\_t\* pMax);

功能：读取定时器延时的范围

参数说明:

hCam        相机句柄

pMin    定时器延时的最小值

pMax    定时器延时的最大值

返回值： 成功返回 MVST\_SUCCESS,否则返回错误信息

- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVGetTimerDelay(HANDLE hCam,

```
uint32_t* pDelay);
```

功能：读取定时器延时

参数说明:

hCam          相机句柄

pDelay 定时器延时

返回值： 成功返回 MVST\_SUCCESS,否则返回错误信息

- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVSetPacketDelay(HANDLE hCam, unsigned int nDelay\_us);

功能：设置网络数据包之间的时间间隔。如果网卡或电脑的性能欠佳，无法处理高速到达的数据包，会导致丢失数据包，

参数说明:

hCam

nDelay\_us 时间间隔(单位:微秒)

返回值： 成功返回 MVST\_SUCCESS,否则返回错误信息

- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVGetPacketDelayRange(HANDLE hCam, unsigned int\* pMin, unsigned int\* pMax);

功能：读取网络数据包间隔范围

参数说明:

hCam          相机句柄

pMin    数据包间隔时间最小值，单位 us

pMax 数据包间隔时间最大值，单位 us

返回值： 成功返回 MVST\_SUCCESS,否则返回错误信息

- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVGetPacketDelay(HANDLE hCam, unsigned int\* pDelay\_us);

功能：读取网络数据包间隔。

参数说明:

hCam          相机句柄

pDelay\_us 数据包间隔时间，单位 us

返回值： 成功返回 MVST\_SUCCESS,否则返回错误信息

- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVSetPacketSize(HANDLE hCam, unsigned int nPacketSize);

功能：设置网络数据包的大小。

参数说明:

hCam            相机句柄

nPacketSize 网络数据包大小(单位:字节)。该大小必须小于网卡能够支持的最大巨型帧(Jumbo Frame)。

返回值： 成功返回 MVST\_SUCCESS,否则返回错误信息

- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVGetPacketSizeRange(HANDLE hCam, unsigned int\* pMin, unsigned int\* pMax);

功能：读取网络数据包大小的范围。

参数说明:

hCam            相机句柄

pMin    网络数据包最小值

pMax 网络数据包最大值

返回值： 成功返回 MVST\_SUCCESS,否则返回错误信息

- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVGetPacketSize(HANDLE hCam, unsigned int\* pPacketSize);

功能：读取网络数据包大小

参数说明:

hCam            相机句柄

pPacketSize 数据包大小

返回值： 成功返回 MVST\_SUCCESS,否则返回错误信息

- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVSetHeartbeatTimeout(HANDLE hCam, unsigned long nTimeout);//unit ms

功能：设置心跳超时时间

参数说明:

hCam 相机句柄

nTimeOut 心跳超时时间 单位 ms

返回值： 成功返回 MVST\_SUCCESS,否则返回错误信息

- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVSetUserOutputValue0(HANDLE hCam, bool bSet);

功能：当闪光同步源选为 UserOutput 时

参数说明:

hCam 相机句柄

bSet 设置电平

返回值： 成功返回 MVST\_SUCCESS,否则返回错误信息

- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVGetUserOutputValue0(HANDLE hCam, bool\* pSet);

功能：读取用户设置的闪光同步

参数说明:

hCam 相机句柄

pSet

返回值： 成功返回 MVST\_SUCCESS,否则返回错误信息

- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVSetStrobeInvert(HANDLE hCam, bool bInvert);

功能：闪光同步是否反转，即闪光同步有效时输出高电平还是低电平。

参数说明:

hCam 相机句柄

bInvert

返回值： 成功返回 MVST\_SUCCESS,否则返回错误信息

- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVGetStrobeInvert(HANDLE hCam,

```
bool* pInvert);
```

功能：读取闪光同步是否反转

参数说明:

hCam          相机句柄

pInvert

返回值： 成功返回 MVST\_SUCCESS,否则返回错误信息

- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVSetStrobeSource(HANDLE hCam, LineSourceEnums source);

功能：闪光同步信号源

参数说明:

hCam

source

返回值： 成功返回 MVST\_SUCCESS,否则返回错误信息

- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVGetStrobeSource(HANDLE hCam, LineSourceEnums\* pSource);

功能：读取闪光同步信号源

参数说明:

hCam          相机句柄

pSource 闪光同步信号源

返回值： 成功返回 MVST\_SUCCESS,否则返回错误信息

- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVTriggerSoftware(HANDLE hCam);

功能：发出软件触发指令

参数说明:

hCam          相机句柄

返回值： 成功返回 MVST\_SUCCESS,否则返回错误信息

- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVSetTriggerDelay(HANDLE hCam, uint32\_t nDelay\_us);

功能：设置相机接到触发信号后延迟多少微秒后再开始曝光。

参数说明:

hCam        相机句柄

nDelay\_us

返回值： 成功返回 MVST\_SUCCESS,否则返回错误信息

- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVGetTriggerDelayRange(HANDLE hCam, uint32\_t\* pMin, uint32\_t\* pMax);

功能：读取触发延时范围

参数说明:

hCam        相机句柄

pMin        触发延时最小值,单位 us

pMax        触发延时最大值,单位 us

返回值： 成功返回 MVST\_SUCCESS,否则返回错误信息

- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVGetTriggerDelay(HANDLE hCam, uint32\_t\* pDelay\_us);

功能：读取触发延时

参数说明:

hCam        相机句柄

pDelay\_us    触发延时,单位 us

返回值： 成功返回 MVST\_SUCCESS,否则返回错误信息

- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVSetTriggerActivation(HANDLE hCam, TriggerActivationEnums act);

功能：当使用触发线触发时,设置是上升沿触发还是下降沿触发

参数说明:

hCam

act    上升沿或下降沿

返回值： 成功返回 MVST\_SUCCESS,否则返回错误信息



- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVGetTriggerActivation(HANDLE hCam, TriggerActivationEnums\* pAct);

功能：读取触发极性

参数说明:

hCam        相机句柄

pAct

返回值： 成功返回 MVST\_SUCCESS,否则返回错误信息

- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVSetTriggerSource(HANDLE hCam, TriggerSourceEnums source);

功能：设置触发源

参数说明:

hCam        相机句柄

source 触发源

返回值： 成功返回 MVST\_SUCCESS,否则返回错误信息

- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVGetTriggerSource(HANDLE hCam, TriggerSourceEnums\* pSource);

功能：读取触发源

参数说明:

hCam        相机句柄

pSource 触发源，软触发或外触发

返回值： 成功返回 MVST\_SUCCESS,否则返回错误信息

- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVSetTriggerMode(HANDLE hCam, TriggerModeEnums mode);

功能：设置触发模式

参数说明:

hCam        相机句柄

mode 触发模式

返回值： 成功返回 MVST\_SUCCESS,否则返回错误信息

- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVGetTriggerMode(HANDLE hCam, TriggerModeEnums\* pMode);

功能：读取触发模式

参数说明:

hCam 相机句柄

pMode 触发模式 TriggerMode\_Off,TriggerMode\_On

返回值： 成功返回 MVST\_SUCCESS,否则返回错误信息

- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVStopGrab(HANDLE hCam);

功能：停止采集图像

参数说明:

hCam 相机句柄

返回值： 成功返回 MVST\_SUCCESS,否则返回错误信息

- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVStartGrab(HANDLE hCam, MVStreamCB StreamCB, ULONG\_PTR nUserVal);

功能：开始采集图像

参数说明:

hCam 相机句柄

StreamCB 回调函数指针

nUserVal 用户数据，传递到回调函数的形参

返回值： 成功返回 MVST\_SUCCESS,否则返回错误信息

- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVSetFrameRate(HANDLE hCam, double fps);

功能：设置帧率

参数说明:

hCam 相机句柄

fps 帧率 帧/秒

返回值： 成功返回 MVST\_SUCCESS,否则返回错误信息

- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVGetFrameRate(HANDLE hCam, double\* fFPS);

功能：读取当前帧率

参数说明:

hCam 相机句柄

fFPS帧率 帧/秒

返回值： 成功返回 MVST\_SUCCESS,否则返回错误信息

- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVGetFrameRateRange(HANDLE hCam, double\* pFpsMin, double\* pFpsMax);

功能：读取帧率可设置的范围

参数说明:

hCam 相机句柄

pFpsMin 最低帧率

pFpsMax 最高帧率

返回值： 成功返回 MVST\_SUCCESS,否则返回错误信息

- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVSetExposureTime(HANDLE hCam,double nExp\_us);

功能：设置曝光时间

参数说明:

hCam 相机句柄

nExp\_us 曝光时间 单位为 us

返回值： 成功返回 MVST\_SUCCESS,否则返回错误信息

- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVGetExposureTimeRange(HANDLE hCam, double\* pExpMin, double\* pExpMax);

功能：读取曝光时间的设置范围

参数说明:

hCam        相机句柄

pExpMin    最短曝光时间 单位为 us

pExpMax    最长曝光时间 单位为 us

返回值： 成功返回 MVST\_SUCCESS,否则返回错误信息

- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVGetExposureTime(HANDLE hCam, double\* pExposuretime);

功能：读取当前曝光时间

参数说明:

hCam

pExposuretime 单位 us

返回值： 成功返回 MVST\_SUCCESS,否则返回错误信息

- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVSetGainBalance(HANDLE hCam, int nBalance);

功能：设置是否自动通道平衡

参数说明:

hCam        相机句柄

nBalance    是否自动通道平衡

返回值： 成功返回 MVST\_SUCCESS,否则返回错误信息

- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVGetGainBalance(HANDLE hCam, int\* pBalance);

功能：读取是否通道自动平衡

参数说明:

hCam        相机句柄

pBalance    是否自动平衡

返回值： 成功返回 MVST\_SUCCESS,否则返回错误信息

- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVSetWhiteBalance(HANDLE hCam, double fRed, double fGreen, double fBlue);

功能：设置白平衡系数

参数说明:

hCam        相机句柄

fRed        红色平衡系数

fGreen    绿色平衡系数

fBlue    蓝色平衡系数

返回值： 成功返回 MVST\_SUCCESS,否则返回错误信息

- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVGetWhiteBalanceRange(HANDLE hCam, double\* pMin, double\* pMax);

功能：读取白平衡设置的范围

参数说明:

hCam        相机句柄

pMin    系数最小值

pMax    系数最大值

返回值： 成功返回 MVST\_SUCCESS,否则返回错误信息

- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVGetWhiteBalance(HANDLE hCam, double\* pRed, double\* pGreen, double\* pBlue);

功能：读取当前白平衡系数

参数说明:

hCam        相机句柄

pRed    红色平衡系数

pGreen    绿色平衡系数

pBlue    蓝色平衡系数

返回值： 成功返回 MVST\_SUCCESS,否则返回错误信息

- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVGetGainRangeTaps(HANDLE hCam, double\* pGainMin, double\* pGainMax, int nTap);

功能：当相机传感器为多通道时，读取某个通道的增益可设置的范围

参数说明:

hCam        相机句柄

pGainMin   增益最小值

pGainMax   增益最大值

nTap        通道。双通道[0,1],四通道[0,1,2,3]

返回值：   成功返回 MVST\_SUCCESS,否则返回错误信息

- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVGetGainTaps(HANDLE hCam, double\* pGain, int nTap);

功能：当相机传感器为多通道时，读取某个通道的增益

参数说明:

hCam        相机句柄

pGain

nTap        通道。双通道[0,1],四通道[0,1,2,3]

返回值：   成功返回 MVST\_SUCCESS,否则返回错误信息

- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVSetGainTaps(HANDLE hCam, double fGain, int nTap);

功能：当相机传感器为多通道时，设置某个通道的增益

参数说明:

hCam        相机句柄

fGain    增益

nTap        通道。双通道[0,1],四通道[0,1,2,3]

返回值：   成功返回 MVST\_SUCCESS,否则返回错误信息

- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVSetGain(HANDLE hCam, double fGain);

功能：设置增益

参数说明:

hCam    相机句柄

fGain 增益

返回值： 成功返回 MVST\_SUCCESS,否则返回错误信息

- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVGetGainRange(HANDLE hCam, double\* pGainMin, double\* pGainMax);

功能：读取增益可以设置的范围

参数说明:

hCam 相机句柄

pGainMin 最小值

pGainMax 最大值

返回值： 成功返回 MVST\_SUCCESS,否则返回错误信息

- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVGetGain(HANDLE hCam, double\* pGain);

功能：读取当前增益值

参数说明:

hCam 相机句柄

pGain

返回值： 成功返回 MVST\_SUCCESS,否则返回错误信息

- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVGetSensorTaps(HANDLE hCam, SensorTapsEnums\* pSensorTaps);

功能：读取传感器的通道数

参数说明:

hCam 相机句柄

pSensorTaps

返回值： 成功返回 MVST\_SUCCESS,否则返回错误信息

- \*/ MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVSetPixelFormat( HANDLE hCam,MV\_PixelFormatEnums PixelFormat );

功能：设置图像的像素格式

参数说明:

hCam          相机句柄

PixelFormat

返回值： 成功返回 MVST\_SUCCESS,否则返回错误信息

- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVGetPixelFormat(HANDLE hCam, MV\_PixelFormatEnums\* pPixelFormat);

功能：读取图像的像素格式

参数说明:

hCam          相机句柄

pPixelFormat

返回值： 成功返回 MVST\_SUCCESS,否则返回错误信息

- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVSetOffsetY(HANDLE hCam, int nOffsetY);

功能：设置垂直方向偏移量。图像宽度设置到小于最大宽度时，可以调整垂直偏移量，设置采集窗口的垂直起始位置。

参数说明:

hCam

nOffsetY 垂直偏移量。应该在垂直偏移量允许的范围之内。

返回值： 成功返回 MVST\_SUCCESS,否则返回错误信息

- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVGetOffsetYRange(HANDLE hCam, int\* pOffsetYMin, int\* pOffsetYMax);

功能：读取垂直方向偏移量取值范围。

参数说明:

hCam

pOffsetYMin 垂直偏移量最小值

pOffsetYMax      垂直偏移量最大值

返回值： 成功返回 MVST\_SUCCESS,否则返回错误信息



- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVGetOffsetY(HANDLE hCam, int\* pOffsetY);

功能：读取垂直方向偏移量。图像宽度设置到小于最大宽度时，可以调整垂直偏移量，设置采集窗口的垂直起始位置。

参数说明:

hCam

pOffsetY 垂直偏移量

返回值： 成功返回 MVST\_SUCCESS,否则返回错误信息

- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVSetOffsetX(HANDLE hCam, int nOffsetX);

功能：设置水平方向偏移量。图像宽度设置到小于最大宽度时，可以调整水平偏移量，设置采集窗口的水平起始位置。

参数说明:

hCam

nOffsetX 水平偏移量。应该在水平偏移量允许的范围之内。

返回值： 成功返回 MVST\_SUCCESS,否则返回错误信息

- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVGetOffsetXRange(HANDLE hCam, int\* pOffsetXMin, int\* pOffsetXMax);

功能：读取水平方向偏移量取值范围。

参数说明:

hCam

pOffsetXMin水平偏移量最小值

pOffsetXMax 水平偏移量最大值

返回值： 成功返回 MVST\_SUCCESS,否则返回错误信息

- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVGetOffsetX(HANDLE hCam, int\* pOffsetX);

功能：读取水平方向偏移量。图像宽度设置到小于最大宽度时，可以调整水平偏移量，设置采集窗口的水平起始位置。

参数说明:

hCam

pOffsetX 水平偏移量

返回值： 成功返回 MVST\_SUCCESS,否则返回错误信息

- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVSetHeight(HANDLE hCam, int nHeight);

功能：设置图像的高度

参数说明:

hCam 相机句柄

nHeight 图像高度，应该在高度可设置范围之内

返回值： 成功返回 MVST\_SUCCESS,否则返回错误信息

- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVGetHeightRange(HANDLE hCam, int\* pHeightMin, int\* pHeightMax);

功能：读取图像高度可设置的范围

参数说明:

hCam 相机句柄

pHeightMin 图像高度可设置的最小值

pHeightMax 图像高度可设置的最大值

返回值： 成功返回 MVST\_SUCCESS,否则返回错误信息

- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVGetHeight(HANDLE hCam, int\* pHeight);

功能：读取图像高度

参数说明:

hCam 相机句柄

pHeight 图像高度[像素]

返回值： 成功返回 MVST\_SUCCESS,否则返回错误信息

- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVSetWidth(HANDLE hCam, int nWidth);

功能：设置图像的宽度

参数说明:

hCam        相机句柄

nWidth 图像宽度，应该在宽度可设置范围之内，并且 = 最小宽度 + 步长 x 整数

返回值： 成功返回 MVST\_SUCCESS,否则返回错误信息

- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVGetWidthInc(HANDLE hCam, int\* pWidthInc);

功能：读取图像宽度调整的步长

参数说明:

hCam        相机句柄

pWidthInc 图像宽度的调整的步长，即图像的宽度 = 最小宽度 + 步长 x 整数

返回值： 成功返回 MVST\_SUCCESS,否则返回错误信息

- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVGetWidthRange(HANDLE hCam, int\* pWidthMin, int\* pWidthMax);

功能：读取图像宽度可设置的范围

参数说明:

hCam        相机句柄

pWidthMin 图像宽度可设置的最小值

pWidthMax 图像宽度可设置的最大值

返回值： 成功返回 MVST\_SUCCESS,否则返回错误信息

- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVGetWidth(HANDLE hCam, int\* pWidth);

功能：读取图像宽度

参数说明:

hCam        相机句柄

pWidth 图像宽度[像素]

返回值： 成功返回 MVST\_SUCCESS,否则返回错误信息

MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVCloseCam(HANDLE hCam);

功能：关闭相机。断开和相机的连接。

参数说明:

hCam 相机的句柄

返回值： 成功返回 MVST\_SUCCESS,否则返回错误信息

- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVOpenCamByIP( char \*ip,HANDLE \*hCam );

功能：打开指定 IP 的相机

参数说明:

ip 相机的 IP 地址。

hCam 如果成功,返回的相机句柄。如果失败，为 NULL。

返回值： 成功返回 MVST\_SUCCESS,否则返回错误信息

- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVOpenCamByUserDefinedName(char\* name, HANDLE\* hCam);

功能：打开指定 UserDefinedName 的相机

参数说明:

name UserDefinedName。

hCam 如果成功,返回的相机句柄。如果失败，为 NULL。

返回值： 成功返回 MVST\_SUCCESS,否则返回错误信息

- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVOpenCamByIndex(unsigned char idx, HANDLE\* hCam);

功能：打开第 idx 个相机

参数说明:

idx idx 从0开始，按照相机的 IP 地址排序，地址小的排在前面。

hCam 如果成功,返回的相机句柄

返回值： 成功返回 MVST\_SUCCESS,否则返回错误信息

- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVGetCameraInfo(unsigned char idx,

```
MVCamInfo* pCamInfo);
```

功能：得到第 idx 个相机的信息。

参数说明:

idx idx 从0开始，按照相机的 IP 地址排序，地址小的排在前面。

pCamInfo 相机的信息 (IP,MAC,SN,型号...)

返回值： 成功返回 MVST\_SUCCESS,否则返回错误信息

- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVGetNumOfCameras(int\* pNumCams);

功能：获取连接到计算机上的相机的数量

参数说明:

pNumCams 相机数量

返回值： 成功返回 MVST\_SUCCESS,否则返回错误信息

- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVUpdateCameraList();

功能：查找连接到计算机上的相机

参数说明:

返回值： 成功返回 MVST\_SUCCESS,否则返回错误信息

- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVTerminateLib();

功能：退出函数库。在程序退出前调用，以释放资源。

参数说明:

返回值： 成功返回 MVST\_SUCCESS,否则返回错误信息

- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVInitLib();

功能：初始化函数库。在调用函数所有函数之前调用。

参数说明:

返回值： 成功返回 MVST\_SUCCESS,否则返回错误信息

- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVGetDevInfo(unsigned char  
idx,MVCamInfo \*pCamInfo);

功能：获取 MVEnumerateAllDevices 搜索到的相机的信息

参数说明：

idx 相机序号; pCamInfo 相机信息

返回值：成功返回 MVST\_SUCCESS,否则返回错误信息

- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVSetPersistentIpAddress( HANDLE hCam, const char\* pIpAddress, const char\* pSubnetMask, const char\* pDefaultGateway);

功能：设置静态 IP 地址

参数说明：

pIpAddress IP 地址; pSubnetMask 子网掩码; pDefaultGateway 默认网关;

返回值：成功返回 MVST\_SUCCESS,否则返回错误信息

- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVForceIp( const char\* pMacAddress, const char\* pIpAddress, const char\* pSubnetMask, const char\* pDefaultGateway)

功能：为相机设置 IP 地址

参数说明：

pMacAddress 待设置 ip 相机的 MAC 地址; pIpAddress 设置给相机的 IP 地址

pSubnetMask 设置给相机的子网掩码; pDefaultGateway 设置给相机的默认网关

返回值：成功返回 MVST\_SUCCESS,否则返回错误信息

- MVGIGE\_API MVSTATUS\_CODES \_\_stdcall MVGetPersistentIpAddress( HANDLE hCam, char\* pIpAddress, size\_t\* pIpAddressLen, char\* pSubnetMask, size\_t\* pSubnetMaskLen, char\* pDefaultGateway, size\_t\* pDefaultGatewayLen);

功能：获取相机的静态 IP 设置

参数说明：

pIpAddressLen pIpAddress 缓冲区长度;

pSubnetMaskLen pSubnetMask 缓冲区长度;

pDefaultGatewayLen pDefaultGateway 缓冲区长度

返回值：成功返回 MVST\_SUCCESS,否则返回错误信息

### 2.2.3、MVImage.h

```
#include "GigECamera_Types.h"
```

结构体

```
class MVImage
```

宏定义

```
#define MVGIGE_API __declspec(dllimport)
```

### 2.2.4、MVImageC.h

宏定义

```
#define MVGIGE_API __declspec(dllimport)
```

函数

- MVGIGE\_API HANDLE \_\_stdcall MVImageCreate(int nWidth, int nHeight, int nBPP);

功能：创建图像

参数说明：

nWidth 图像宽度

nHeight 图像高度

nBPP 每像素 Bit 数。取值范围为8(8bit 灰度),16(16bit 灰度),24(8bit 彩色),48(16bit 彩色)

返回值：图像句柄

- MVGIGE\_API int \_\_stdcall MVImageCreateByPixelFormat( HANDLE hImage, int nWidth, int nHeight, MV\_PixelFormatEnums PixelFormat );

功能：根据宽度、高度、和像素格式创建图像

参数说明：

nWidth 宽度

nHeight 高度

PixelFormat 像素格式

注解：

\*

|   |                       |  |                 |  |
|---|-----------------------|--|-----------------|--|
| * | PixelFormat           |  | Image Format    |  |
| * |                       |  |                 |  |
| * | PixelFormat_Mono8     |  | 8 Bits 灰度图      |  |
| * |                       |  |                 |  |
| * | PixelFormat_Mono16    |  | 16 Bits 灰度图     |  |
| * |                       |  |                 |  |
| * | PixelFormat_BayerBG8  |  |                 |  |
| * | PixelFormat_BayerRG8  |  |                 |  |
| * | PixelFormat_BayerGB8  |  | 24 Bits BGR 彩色图 |  |
| * | PixelFormat_BayerGR8  |  |                 |  |
| * |                       |  |                 |  |
| * | PixelFormat_BayerBG16 |  |                 |  |
| * | PixelFormat_BayerRG16 |  |                 |  |
| * | PixelFormat_BayerGB16 |  | 48 Bits BGR 彩色图 |  |
| * | PixelFormat_BayerGR16 |  |                 |  |

返回值：图像句柄

- MVGIGE\_API BOOL \_\_stdcall MVImageIsNull(HANDLE hImage);

功能：判断图像是否为空。在调用 Create 或 CreateByPixelFormat 之前，图像为空

参数说明：

hImage 相机句柄

返回值：TRUE（图像还未创建）

- MVGIGE\_API int \_\_stdcall MVImageGetWidth(HANDLE hImage);

功能：获取图像宽度，单位为像素

参数说明：

hImage 相机句柄

返回值：图像宽度



- MVGIGE\_API int \_\_stdcall MVImageGetHeight(HANDLE hImage);

功能：获取图像高度，单位为像素

参数说明：

hImage 相机句柄

返回值：图像高度

- MVGIGE\_API void\* \_\_stdcall MVImageGetBits(HANDLE hImage);

功能：获取图像缓冲区的指针

参数说明：

hImage 相机句柄

返回值：指向图像缓冲区的指针

- MVGIGE\_API int MVImageGetPitch(HANDLE hImage);

功能：获取一行图像的字节数

参数说明：

hImage 相机句柄

返回值：一行图像的字节数

- MVGIGE\_API int MVImageGetBPP(HANDLE hImage);

功能：获取图像中每个像素所占位数

参数说明：

hImage 相机句柄

注解：8: 8Bit 灰度图，16: 16Bit 灰度图，24: 8Bit BGR 彩色图，48: 16Bit BGR 彩色图

返回值：每个像素所占位数

- MVGIGE\_API HBITMAP \_\_stdcall MVImageGetHBitmap(HANDLE hImage);

功能：获取 HBITMAP 类型的句柄。用于绘图或生成 AVI

参数说明：

hImage 相机句柄

返回值：HBITMAP 类型的句柄

- MVGIGE\_API BOOL \_\_stdcall MVImageDrawEx(HANDLE hImage, HDC hDestDC, int xDest, int yDest, int nDestWidth, int nDestHeight, int xSrc, int ySrc, int nSrcWidth, int nSrcHeight);

功能：在目标 DC 上的指定区域绘制本图像的全部或局部

参数说明：

hImage 相机句柄

hDestDC 目标 DC 的句柄

xDest x 坐标，目标矩形的左上角

yDest y 坐标，目标矩形的左上角

nDestWidth 宽度，目标矩形

nDestHeight 高度，目标矩形。

xSrc x 坐标，源矩形的左上角。

ySrc y 坐标，源矩形的左上角。

nSrcWidth 宽度，源矩形。

nSrcHeight 高度，源矩形。

返回值：

- MVGIGE\_API BOOL \_\_stdcall MVImageDraw(HANDLE hImage, HDC hDestDC, int xDest, int yDest);

功能：在目标 DC 上的指定位置1:1绘制本图像

参数说明：

hImage 相机句柄

hDestDC 目标 DC

xDest 图像绘制的起始水平坐标

yDest 图像绘制的起始垂直坐标

返回值：

- MVGIGE\_API BOOL \_\_stdcall MVImageDrawHwnd(HANDLE hImage, HWND hWnd, int xDest, int yDest);

功能：在目标窗体上的指定位置1:1绘制本图像

参数说明：

hImage 相机句柄

hWnd 目标窗体

xDest 图像绘制的起始水平坐标

yDest 图像绘制的起始垂直坐标

返回值：

- MVGIGE\_API BOOL \_\_stdcall MVImageDrawHwndEx( HANDLE hImage, HWND hWnd, int xDest, int yDest, int nDestWidth, int nDestHeight, int xSrc, int ySrc, int nSrcWidth, int nSrcHeight );

功能：在目标窗体上的指定区域绘制本图像的全部或局部

参数说明：

hImage 相机句柄

hWnd 目标窗体的句柄

xDest x 坐标，目标矩形的左上角。

yDest y 坐标，目标矩形的左上角。

nDestWidth 宽度，目标矩形。

nDestHeight 高度，目标矩形。

xSrc x 坐标，源矩形的左上角。

ySrc y 坐标，源矩形的左上角。

nSrcWidth 宽度，源矩形。

nSrcHeight 高度，源矩形。

返回值：

- MVGIGE\_API HDC \_\_stdcall MVImageGetDC(HANDLE hImage);

功能：获取图像的 DC，可用于在图像上绘制图形

参数说明：

hImage 相机句柄

注解：使用结束后应调用 ReleaseDC 进行释放

返回值：图像 DC

- MVGIGE\_API void \_\_stdcall MVImageReleaseDC(HANDLE hImage);

功能：释放使用 GetDC 获取的 DC

参数说明：

hImage 相机句柄

返回值：

- MVGIGE\_API int \_\_stdcall MVImageSave(HANDLE hImage, LPCSTR pszFileName);

功能：保存图片

参数说明：

hImage 相机句柄

pszFileName 文件名。文件类型可以为.bmp,.jpg,.tif,.raw(仅支持位深为16位的图像)

返回值：16位和48位图像仅在保存为.tif 格式时，保留16位或48位位深。保存为 bmp,jpg 均被转换为8位或24位图像保存。

- MVGIGE\_API int \_\_stdcall MVImageLoad( HANDLE hImage,LPCSTR pszFileName );

功能：从文件读取图像

参数说明：

hImage 相机句柄

pszFileName 文件名。支持的文件类型为.bmp,.jpg,.tif,.raw

返回值：

- MVGIGE\_API void \_\_stdcall MVImageDestroy(HANDLE hImage);

功能：释放图像资源

参数说明：

hImage 相机句柄

返回值：

## 2.2.5、MVRecordDlg.h

宏定义

```
#define MVRECORDDLG_API __declspec(dllimport)
```

函数

- MVRECORDDLG\_API BOOL \_\_stdcall MVRecordDlgCreate( HANDLE\* pRecDlg, HANDLE hCam, CWnd\* pParentWnd = NULL );

功能：创建录像计时器对话框

参数说明：

pRecDlg 返回录像计时器对话框句柄

hCam 相机句柄

pParentWnd 父窗口句柄

返回值：

- MVRECORDDLG\_API BOOL \_\_stdcall MVRecordDlgCreateEx( HANDLE\* pRecDlg, HANDLE hCam, HWND pParentWnd = NULL );

功能：创建录像计时器对话框

参数说明：

pRecDlg 返回录像计时器对话框句柄

hCam 相机句柄

pParentWnd 父窗口句柄

返回值：

- MVRECORDDLG\_API BOOL \_\_stdcall MVRecordDlgDestroy( HANDLE hRecDlg );

功能：销毁录像计时器对话框

参数说明：

hRecDlg 录像计时器对话框句柄

返回值：

- MVRECORDDLG\_API BOOL \_\_stdcall MVRecordDlgShow( HANDLE hRecDlg, int nCmdShow );

功能：以非模式框方式显示或关闭录像计时器对话框

参数说明：

hRecDlg 录像计时器对话框句柄

nCmdShow SW\_SHOW:显示， SW\_HIDE:关闭

返回值：

- MVRECORDDLG\_API void \_\_stdcall MVRecordDlgCamRun(HANDLE hRecDlg, BOOL bCamRun );

功能：设置相机现在是否正工作在采集模式

参数说明：

hRecDlg 录像计时器对话框句柄

bCamRun 如果相机正工作在采集模式下，设置为 TRUE,否则设置为 FALSE

注解：如果正工作在采集模式，属性页中将禁用一些采集状态下不允许改变的相机属性，如图像大小等。

返回值：

- MVRECORDDLG\_API int \_\_stdcall MVRecordDlgRecord( HANDLE hRecDlg, HANDLE hImage, unsigned short nBlockId );

功能：向录像中插入帧

参数说明：

hRecDlg 录像计时器对话框句柄

hImage 图像句柄

nBlockId 图像的帧号，从1开始，每采集一次帧号加1

返回值：

- MVRECORDDLG\_API int \_\_stdcall MVRecordDlgSetParent( HANDLE hRecDlg, HANDLE hParent );

功能：设置父窗口句柄

参数说明：

hRecDlg 序列帧计时器对话框句柄

hParent 父窗口句柄

返回值：

## 2.2.6、MVSequenceDlg.h

宏定义

MVSEQUENCEDLG\_API \_\_declspec(dllexport)

函数

- MVSEQUENCEDLG\_API BOOL \_\_stdcall MVSequenceDlgCreate( HANDLE\* pRecDlg, CWnd\* pParentWnd = NULL );

功能：创建序列帧计时器对话框

参数说明：

pRecDlg 返回序列帧计时器对话框句柄

pParentWnd 父窗口句柄

返回值：

- MVSEQUENCEDLG\_API BOOL \_\_stdcall MVSequenceDlgCreateEx( HANDLE\* pRecDlg, HWND pParentWnd = NULL );

功能：创建序列帧计时器对话框

参数说明：

pRecDlg 返回序列帧计时器对话框句柄

pParentWnd 父窗口句柄

返回值：

- MVSEQUENCEDLG\_API BOOL \_\_stdcall MVSequenceDlgDestroy( HANDLE hRecDlg );

功能：销毁序列帧计时器对话框

参数说明：

hRecDlg 序列帧计时器对话框句柄

返回值：

- MVSEQUENCEDLG\_API BOOL \_\_stdcall MVSequenceDlgShow( HANDLE hRecDlg, int nCmdShow );

功能：以非模式框方式显示或关闭序列帧计时器对话框

参数说明：

hRecDlg 序列帧计时器对话框句柄

nCmdShow SW\_SHOW:显示， SW\_HIDE:关闭

返回值：

- MVSEQUENCEDLG\_API void \_\_stdcall MVSequenceDlgCamRun( HANDLE hRecDlg, BOOL bCamRun );

功能：设置相机现在是否正工作在采集模式

参数说明：

hRecDlg 序列帧计时器对话框句柄

bCamRun 如果相机正工作在采集模式下，设置为 TRUE,否则设置为 FALSE

注解：如果正工作在采集模式，属性页中将禁用一些采集状态下不允许改变的相机属性，如图像大小等

返回值：

- MVSEQUENCEDLG\_API int \_\_stdcall MVSequenceDlgGetFileName( HANDLE hSeqDlg, char \*fname, int szBuf );

功能：获得即将进入队列的文件名，及当前图片是否需要保存

参数说明：

hSeqDlg 序列帧计时器对话框句柄

Fname 返回文件名

szBuf 文件名长度

返回值：

- MVSEQUENCEDLG\_API int \_\_stdcall MVSequenceDlgSetParent( HANDLE hSeqDlg,



```
HANDLE hParent);
```

功能：设置父窗口句柄

参数说明：

hSeqDlg 序列帧计时器对话框句柄

hParent 父窗口句柄

返回值：

**西安（总部）**

电话：4000-400-860

传真：4000-400-860 转 3

Email: sales@xamv.com

**深圳**

电话：0755-33034411

Email: sz.sale@xamv.com

**北京**

电话：010-58246500

Email: tuxiangmv@126.com

**上海**

电话：4000-400-860

Email: xamv10@126.com

©Microvision 2017

[www.microvision.com.cn](http://www.microvision.com.cn)



维视图像营销 QQ

4000400860

**Microvision**  
**维视图像**



维视图像微信

microvision400