

Tucsen 相机 Labview 开发指南

鑫图光电有限公司

版权(c) 2011-2025 Xintu Photonics Co., Ltd.(TUCESN)

版权所有

目录

1. 使用前阅读.....	1
2. 简介.....	2
3. 概述.....	2
3.1 层结构.....	2
3.2 原理.....	3
3.3 接口类型.....	3
3.4 术语.....	3
3.4.1 捕获模式.....	3
3.4.2 图像单元.....	3
3.4.3 触发模式.....	3
3.4.4 相机状态.....	4
4. 版本介绍.....	4
4.1. 开发环境.....	4
4.2. 性能优化.....	5
4.2.1 接口数量及调用优化.....	5
4.2.2 VI 模块及调用优化.....	5
4.3. 功能添加.....	5
4.3.1 相机相关信息显示.....	5
4.3.2 外触发.....	6
4.3.3 多相机操作.....	6
5. 参考.....	7
5.1 类型和常量.....	7
5.1.1 TULVRET 错误代码：用于 TU_ErrorReport.vi.....	7
5.1.2 TULV_IDINFO 产品信息 ID：用于 TU_GetCameraInfo.vi.....	8
5.1.3. TULV_IDCAPA 性能 ID：用于 TU_SetParameter.vi.....	9
5.1.4. TULV_IDPROP 属性 ID：用于 TU_SetParameter.vi.....	10
5.1.5. TULV_CAPTURE_MODES 捕获模式 ID：用于 TU_SetTrigger.vi.....	11
5.1.6. TULVIMG_FORMATS 图像格式 ID：用于保存图像.....	11
5.1.7. TULV_TRIGGER_EXP 触发曝光模式 ID：用于 TU_SetTrigger.vi.....	11
5.1.8. TULV_TRIGGER_EDGE 触发激发边沿模式 ID：用于 TU_SetTrigger.vi.....	12
6. VI 介绍与运用.....	12
6.1 TU_InitCamera.vi.....	12
6.2 TU_UninitCamera.vi.....	12
6.3 TU_OpenCamera.vi.....	13
6.4 TU_SetParameter.vi.....	13
6.5 TU_GetParameter.vi.....	14
6.6 TU_GetParameterLimits.vi.....	15
6.7 TU_GetCameraInfo.vi.....	16
6.8 TU_SetTrigger.vi.....	17
6.9 TU_GetTrigger.vi.....	18
6.10 TU_SetROI.vi.....	18
6.11 TU_StartCapture.vi.....	19

Tucsen 相机 Labview 开发指南

6.12 TU_StopCapture.vi.....	20
6.13 TU_WaitForFrame.vi.....	20
6.14 TU_GetFrame.vi.....	21
6.15 TU_ErrorReport.vi.....	22
6.16 TU_OpenImageFile.vi.....	22
6.17 TU_StartRecorder.vi.....	22
6.18 TU_StopRecorder.vi.....	23
6.19 TU_SetOutPutTrigger.vi.....	23
6.20 TU_GetOutPutTrigger.vi.....	24
6.21 TU_StartMultiData.vi.....	24
6.22 TU_StopMultiData.vi.....	25
附录	26

1. 使用前阅读

这份文档和软件示例代码是 TUCSEN 的内部文件和公布内容，以用户能够创建使用 TUCSEN 数字相机中的应用。

本文档和软件示例代码只针对上述目的而公开的，并且不构成所有者的许可、转让或任何其他权力。

使用软件文档的所有风险和结果仍然取决于用户。

本文档可能包括技术错误或印刷错误。并且不能保证这样的错误或文本所产生的任何损害。

TUCSEN 不承诺更新或保持当前的这个文档中所包含的信息。

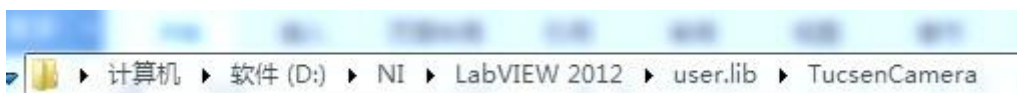
所有品牌和产品名称都是其各自所有者的商标或注册商标。

TUCSEN 对文档的版权保留所有权利。

在没有 TUCSEN 的事先书面许可下，文档的任何部分不得被复制、传递、转录，存储在检索系统或翻译成任何语言或计算机语言，以任何的形式，或以任何方式，任何的手段如：电子、机械、电磁、光学、化学手动或其他。

该版本的使用说明：

由于目前版本不够完善，所以要使用的话必须将压缩包中的 TucsenCamera 文件夹复制到 Labvie 安装路径下的 user.lib 文件夹下，才可以正常调用到相应 VI 模块。后续版本将以打包的形式对相应内容进行安装，省略复制文件夹这一步骤。



该图片为存放相应文件夹示例路径

以上对使用者造成的不便还望谅解，我们将会尽快做出修改修改，以满足用户需求。

2. 简介

本手册详细描述了 TULV_API 规范操作使用 TUCSEN 数字相机。TULV_API 软件开发工具包被称为“SDK”。TULV_API 控制数字相机的部分被称为“模块”。

SDK 包含 VI 模块和一个示例应用程序，展示了如何通过 VI 模块对数字相机进行二次开发。SDK 用户免费使用该软件在任何他们喜欢的方式，如部分修改 VI 构造或创建完全独立的项目。

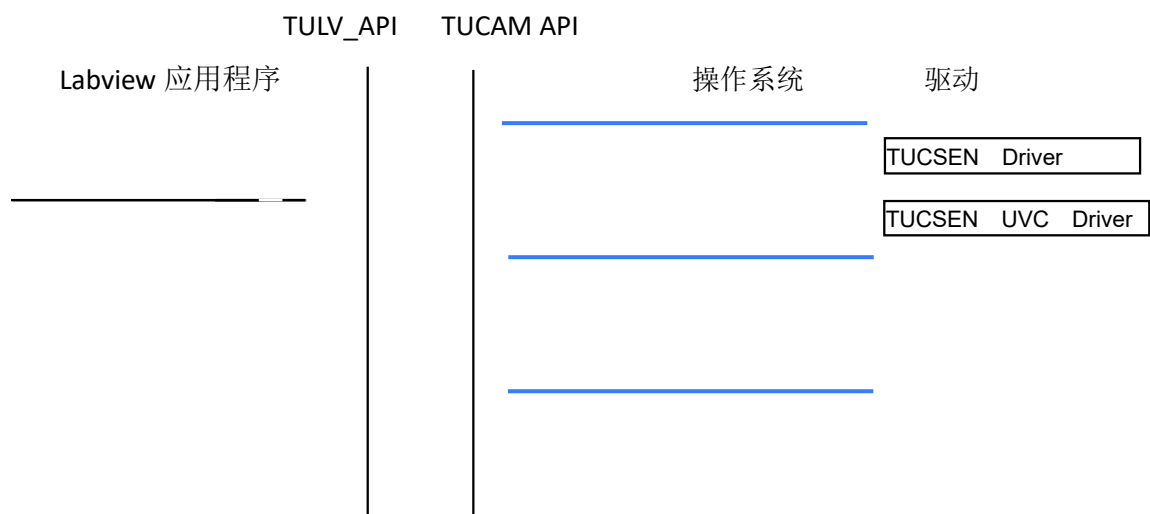
这个 SDK 设计特别容易理解。出于这个原因，函数接口的数量限制到最少，并且函数的调用格式采用 C 语言的写法。

部分扩展的函数是某些特定数字相机可以使用的附加功能。

不同数字相机的数值可能不同，这取决于捕捉图像所使用的数字相机型号。数值应该简单地视为指南，而不是精确值。

3. 概述

3.1 层结构



TUCSEN 数字相机通过 SDK 连接不同的操作系统的数字相机驱动来达到控制数字相机和采集图像数据的作用。

目前的 SDK 只支持 Windows 系统。

3.2 原理

数字相机的具体总线接口和库通过 TULV_API 封装。各模块 VI 部分的接口调用，均来自 TULV_API 的接口，您可参考我们提供的 LabviewDemo 来了解接口的使用情况，如何使用可参考（6.VI 的介绍与运用）。

3.3 接口类型

TULV_API 功能可以分为很多类型

起始/终止处理

相机信息采集

性能/属性获取和设置

内存管理

捕获控制

文件控制

扩展控制

3.4 术语

3.4.1 捕获模式

相机的捕获模式分为以下 2 类：

序列模式（流模式）：用来捕获连续的图像数据。

触发模式：相机通过外部信号来捕获图像。我们称这个选项为“触发模式”，您可以调用 TU_SetTrigger() 来配置此选项。我们也把外部信号称为“外部触发”。

3.4.2 图像单元

通常情况下是二维的，具有垂直和水平方向。

帧：是一个用于图像数据的单位。对于一帧，一个像素的数据是从左到右和从上到下对齐的。这是一系列的图像数据单位。

3.4.3 触发模式

标准模式（Standard）：当相机接收到电平信号后（由激活边沿决定）开始进行一帧或多帧的图像捕获，捕获帧数由配置参数决定。

同步模式（Synchronization）：当相机接收到电平信号后（由激活边沿决定）开始进行曝

光，当收到相反的电平信号后，结束曝光、并且进行图像数据的捕获。即实现每一帧的曝光与读出，均与外触发信号完全同步。

全局模式 (Global)：在相机未触发前进行预触发，当相机接收到电平信号后（由激活边沿决定）或者为软件设定的曝光时间时，结束当前正在进行的重置操作，待曝光结束时捕获图像数据，并重新开始预触发。该方式用于控制卷帘曝光模式的相机实现全局曝光模式。

曝光模式：

曝光时间：接收到触发信号后，由 TULVIDP_EXPTM 设置的曝光时间决定

电平宽度：接收到触发信号后，曝光时间由电平的宽度所决定

注：标准模式 (Standard) 和全局模式 (Global) 可配置这两个选项。同步模式 (Synchronization) 只能是电平宽度。

激发电平类型：

上升电平 (Rising Edge)：接收到的触发电平处于上升沿时开始曝光

下降电平 (Falling Edge)：接收到的触发电平处于下降沿时开始曝光

3.4.4 相机状态

相机的状态决定了能调用那些函数。一些函数会改变相机的状态。下面描述了 4 种相机状态：

不稳定：参数设置和其他函数调用，但它们不在被设置的状态。

稳定的：参数和函数被设置，但是因为帧内存未被创建，捕获图像不能开始。

准备：帧内存已经被创建，图像捕获可以开始。

繁忙：图像捕获正在被执行。

4. 版本介绍

4.1. 开发环境

目前只支持 windows 系统下 Labview 开发。开发者可根据实际情况选择 X86 或 X64 安装包进行开发运用。

注：新版本 Labview 插件仅支持 X64 安装包。

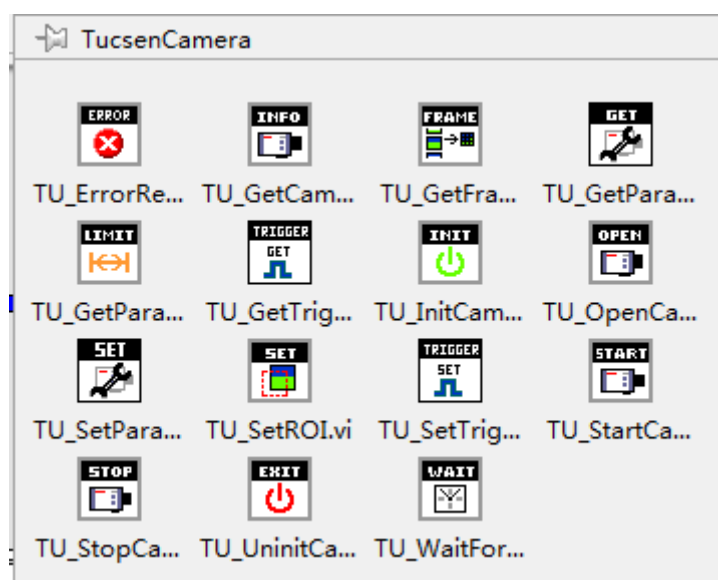
4. 2. 性能优化

4. 2. 1 接口数量及调用优化

版本 2.0 对接口函数进行了系统性的优化，在不减少 1.0 版本基本功能的情况下，减少了接口函数的数量，方便了用户的调用。

4. 2. 2 VI 模块及调用优化

2.0 版本将 1.0 版本中单一的 VI 模块拆分成多个子 VI，并且在 Labview 程序框图中可直接调用，方便了开发者对示例程序 LabviewSample 的拆分研究。VI 模块的介绍详见（6. VI 介绍与运用）。



(图例 4.2.1)

4. 3. 功能添加

4. 3. 1 相机相关信息显示

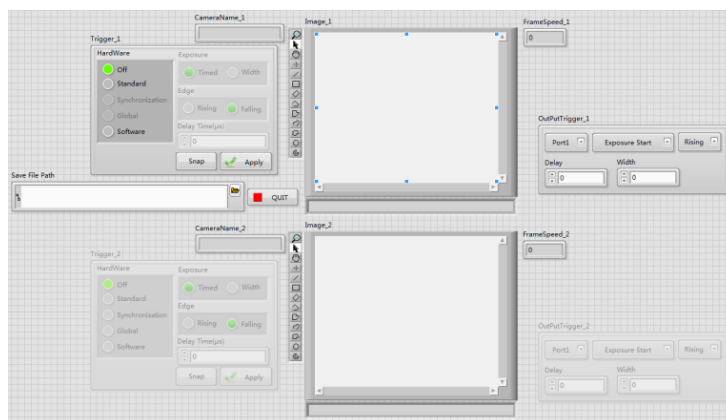
为了能准确识别相机，2.0 版本添加了包括相机名称、TYPE 类型、SN 码、SDK 版本信息、固件版本号的显示。(获取到 0x210 是 USB2.0, 0x03 是 FireBird, 0x04 是 Euresys, 其他是 USB3.0。Cameralink 不显示 SN 和固件号)



(图例 4.3.1)

4.3.2 外触发

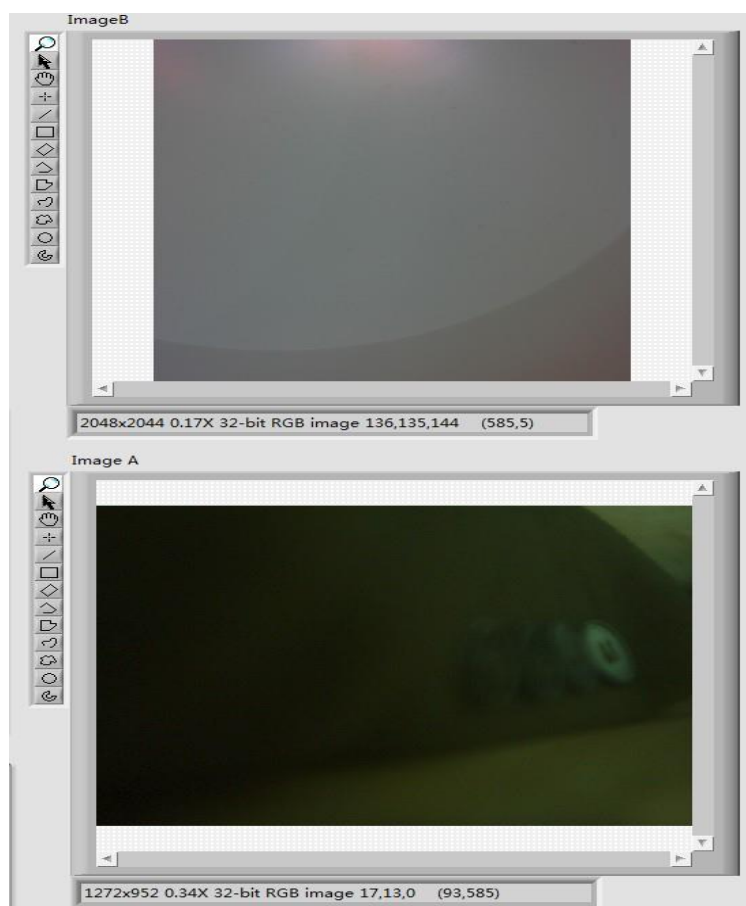
根据相机所支持的功能，2.0 版本在示例程序 SetTriggerDemo.VI 中，添加了外触发模块和触发输出模块，触发模式的介绍详见（3.4.3 触发模式），触发的使用详见 TU_SetTrigger.VI 介绍。



（图例 4.3.2）

4.3.3 多相机操作

2.0 版本 Labview 示例程序可支持多台相机同时预览，同时，可根据相机名称来对该相机进行参数设置，改变预览效果，方便用户做多台相机效果对比。



（图示 4.3.3）

5. 参考

5.1 类型和常量

5.1.1 TULVRET 错误代码：用于 TU_ErrorReport.vi

```

TULVRET_SUCCESS          = 0x00000001,          // 没有错误，一般成功的代码
TULVRET_FAILURE          = 0x80000000,          // 错误

// initialization error
TULVRET_NO_MEMORY        = 0x80000101,          // 没有足够的内存
TULVRET_NO_RESOURCE      = 0x80000102,          // 没有足够的资源（不包括内存）
TULVRET_NO_MODULE        = 0x80000103,          // 没有子模块
TULVRET_NO_DRIVER        = 0x80000104,          // 没有驱动
TULVRET_NO_CAMERA        = 0x80000105,          // 没有相机
TULVRET_NO_GRABBER       = 0x80000106,          // 没有取图
TULVRET_NO_PROPERTY      = 0x80000107,          // 没有代替的属性 ID

TULVRET_FAILOPEN_CAMERA  = 0x80000110,          // 打开相机失败
TULVRET_FAILOPEN_BULKIN  = 0x80000111,          // 打开批传输输入端点失败
TULVRET_FAILOPEN_BULKOUT = 0x80000112,          // 打开批传输输出端点失败
TULVRET_FAILOPEN_CONTROL = 0x80000113,          // 打开控制端点失败
TULVRET_FAILCLOSE_CAMERA = 0x80000114,          // 关闭相机失败
TULVRET_FAILOPEN_FILE    = 0x80000115,          // 打开文件失败

// status error
TULVRET_INIT              = 0x80000201,          // API 需要初始化状态.
TULVRET_BUSY              = 0x80000202,          // API 处于繁忙状态
TULVRET_NOT_INIT          = 0x80000203,          // API 未初始化
TULVRET_EXCLUDED          = 0x80000204,          // 一些资源被独占使用
TULVRET_NOT_BUSY          = 0x80000205,          // API 未处于繁忙状态
TULVRET_NOT_READY         = 0x80000206,          // API 未处于就绪状态

// wait error
TULVRET_ABORT             = 0x80000207,          // 终止处理
TULVRET_TIMEOUT           = 0x80000208,          // 超时
TULVRET_LOSTFRAME         = 0x80000209,          // 帧丢失
TULVRET_MISSFRAME        = 0x8000020A,          // 帧丢失但是是底层驱动问题

// calling error
TULVRET_INVALID_CAMERA    = 0x80000301,          // 无效相机

```

Tucsen 相机 Labview 开发指南

TULVRET_INVALID_HANDLE	= 0x80000302,	// 无效相机句柄
TULVRET_INVALID_OPTION	= 0x80000303,	// 无效配置的值
TULVRET_INVALID_IDPROP	= 0x80000304,	// 无效属性 ID
TULVRET_INVALID_IDCAPA	= 0x80000305,	// 无效性能 ID
TULVRET_INVALID_IDPARAM	= 0x80000306,	// 无效参数 ID
TULVRET_INVALID_PARAM	= 0x80000307,	// 无效参数
TULVRET_INVALID_FRAMEIDX	= 0x80000308,	// 无效帧序号
TULVRET_INVALID_VALUE	= 0x80000309,	// 无效值
TULVRET_INVALID_EQUAL	= 0x8000030A,	// 值相等, 参数无效
TULVRET_INVALID_CHANNEL	= 0x8000030B,	// 属性 ID 指定通道, 但是通道是无效的
TULVRET_INVALID_SUBARRAY	= 0x8000030C,	// 子数组的值是无效的
TULVRET_INVALID_VIEW	= 0x8000030D,	// 无效的显示窗口句柄
TULVRET_INVALID_PATH	= 0x8000030E,	// 无效的文件路径
TULVRET_NO_VALUETEXT	= 0x80000310,	// 属性没有值的文本
TULVRET_OUT_OF_RANGE	= 0x80000311,	// 值超出范围
TULVRET_NOT_SUPPORT	= 0x80000312,	// 不支持的功能或属性
TULVRET_NOT_WRITABLE	= 0x80000313,	// 属性不可写
TULVRET_NOT_READABLE	= 0x80000314,	// 属性不可读
TULVRET_WRONG_HANDSHAKE	= 0x80000410,	// 错误发生在获取错误代码时
TULVRET_NEWAPI_REQUIRED	= 0x80000411,	// 旧 API 不支持, 只有新的 API 支持
TULVRET_ACCESSDENY	= 0x80000412,	// 当相机处于某种状态属性无法访问
TULVRET_NO_CORRECTIONDATA	= 0x80000501,	// 没有校正数据.
// camera or bus trouble		
TULVRET_FAIL_READ_CAMERA	= 0x83001001,	// 从相机读取失败
TULVRET_FAIL_WRITE_CAMERA	= 0x83001002,	// 写入相机失败
TULVRET_OPTICS_UNPLUGGED	= 0x83001003,	// 未插入

5.1.2 TULV_IDINFO 产品信息 ID: 用于 TU_GetCameraInfo.vi

TULVIDI_NAME	= 0x0001,	//相机名称
TULVIDI_SN	= 0x0002,	//SN 码
TULVIDI_FS	= 0x0003,	//固件版本号
TULVIDI_USB	= 0x0004,	//USB 接口类型 (USB2.0/USB3.0)
TULVIDI_SDKNO	= 0x0005,	//API 版本号
TULVIDI_RESOLUTION	= 0x0006,	//分辨率显示
TULVIDI_PGAHIGH	= 0x0007,	//pga high 显示
TULVIDI_PGALOW	= 0x0008,	//pga low 显示
TULVIDI_IMGMODE	= 0x0009,	//采集卡模式显示

Tucsen 相机 Labview 开发指南

TULVIDI_FANMODE	= 0x000A,	//风扇信息显示
TULVIDI_IMGTESTMODE	= 0x000B,	//测试图像显示
TULVIDI_GAINMODE	= 0x000C,	//增益模式显示
TULVIDI_RECORDCOMPRESS	= 0x000D,	//录像压缩格式
TULVIDI_SUMBINNING	= 0x000E,	//累加 bin 显示
TULVIDI_SHUTTER	= 0x000F,	//卷帘/全局显示
TULVIDI_DEPTH	= 0x0010,	//位深显示
TULVIDI_OUTPUTPORT	= 0x0011,	//触发输出端口显示
TULVIDI_OUTPUTKIND	= 0x0012,	//触发输出模式显示
TULVIDI_CAMMODE	= 0x0013,	//相机模式显示

5.1.3. TULV_IDCAPA 性能 ID: 用于 TU_SetParameter.vi

TULVIDC_AUTOWB	= 0x0200,	//自动白平衡 ID (DEC: 512)
TULCIDC_AUTOEXP	= 0x0205,	//自动曝光 ID (DEC: 517)
TULVIDC_DEPTH	= 0x020A,	//数据位宽 ID (DEC: 522)
TULVIDC_AUTOLEVEL	= 0x020F,	//自动色阶 ID (DEC: 527)
TULVIDC_RESOLUTION	= 0x0214,	//分辨率 ID (DEC: 532)
TULVIDC_NOISELEVEL	= 0x0219,	//降噪等级 ID (DEC: 537)
TULVIDC_IMGMODESELECT	= 0x021E,	//CMS(or 11bit) ID (DEC: 542)
TULVIDC_IMGSAVEPATH	= 0x0223,	//图片保存路径 ID (DEC: 547)
TULVIDC_IMGNAME	= 0x0224,	//图片名称 ID (DEC: 548)
TULVIDC_LEDENBALE	= 0x0228,	//LED 灯控制 ID (DEC: 552)
TULVIDC_PIENBALE	= 0x0229,	//PI 加热膜使能 ID (DEC: 553)
TULVIDC_BLACKBALANCE	= 0x022D,	//黑平衡 ID (DEC: 557)
TULVIDC_TESTIMGMODE	= 0x022E,	//测试图 ID (DEC: 558)
TULVIDC_SENSORRESET	= 0x022F,	//Sensor 复位 ID (DEC: 559)
TULVIDC_SUMBINNING	= 0x0230,	//BIN ID (DEC: 560)
TULVIDC_CAMMODE	= 0x0231,	//CameraMode ID (DEC: 561)
TULVIDC_SHUTTER	= 0x0235,	//Shutter ID (DEC: 565)
TULVIDC_SAVEFRAME	= 0x0300,	//保存图片状态 ID (DEC: 768)
TULVIDC_IMGFORMAT	= 0x0301,	//保存图片格式 ID (DEC: 769)
TULVIDC_IMGCOUNT	= 0x0302,	//保存图片张数 ID (DEC: 770)
TULVIDC_TRIGGERSAVE	= 0x0303,	//触发是否保存图片状态 ID(DEC: 771)
TULVIDC_TRIGGERBUFSAVE	= 0x0304,	//触发是否保存图片状态 ID(DEC: 772)
TULVIDC_CAMERASTATE	= 0x0305,	//相机状态 (0、1) ID (DEC: 773)
TULVIDC_FAN	= 0x0306,	//相机风扇档位 ID (DEC: 774)
TULVIDC_PGAAHIGH	= 0x030B,	//相机 PGAAHIGH ID (DEC: 779)
TULVIDC_PGALOW	= 0x0310,	//相机 PGALOW ID (DEC: 784)
TULVIDC_PIXCLK1_EN	= 0x0315,	//相机 PIXCLK1EN ID (DEC: 789)
TULVIDC_PIXCLK2_EN	= 0x031A,	//相机 PIXCLK2EN ID (DEC: 794)
TULVIDC_TRIOUT_EN	= 0x031B,	//相机触发输出使能开关 ID (DEC: 795)
TULVIDC_ROLLINGSCANMODE	= 0x031C,	//相机光片模式选择 ID (DEC: 796)

Tucsen 相机 Labview 开发指南

TULVIDC_ROLLINGSCANLTD	= 0x031D,	//相机光片模式行时间延时 ID (DEC: 797)
TULVIDC_ROLLINGSCANSPLIT	= 0x031E,	//相机光片模式狭缝高度 ID (DEC: 798)
TULVIDC_ROLLINGSCANDIR	= 0x031F,	//相机光片模式扫描方向 ID (DEC: 799)
TULVIDC_ROLLINGSCANRESET	= 0x0320,	//相机光片模式复位 ID (DEC: 800)
TULVIDC_ROLLINGSCANINT	= 0x0321,	//相机光片模式间隔时间 ID (DEC: 801)
TULVIDC_TECENABLE	= 0x0322,	//相机制冷开关 ID (DEC: 802)
TULVIDC_CAMSTART	= 0x0323,	//相机 Start 采图张数设置功能 ID (DEC: 803)
TULVIDC_CAMTRISTOP	= 0x0324,	//相机触发停止 (DEC: 804)
TULVIDC_CAMTRICNT	= 0x0325,	//相机触发数量 (DEC: 805)
TULVIDC_CAMTRIIDX	= 0x0326,	//相机触发数量 (DEC: 806)
TULVIDC_CAMACQFPS	= 0x0327,	//相机采集图像帧率 (DEC: 807)
TULVIDC_CAMSHOWFPS	= 0x0328,	//相机显示图像帧率 (DEC: 808)
TULVIDC_SIGNALFILTER	= 0x0329,	//相机触发滤波 (DEC: 809)
TULVIDC_RECORDCOMPRESS	= 0x0340,	//相机录像压缩功能 (DEC: 832)
TULVIDC_RECORDCONFIG	= 0x0341,	//相机录像配置功能 (DEC: 833)
TULVIDC_BUFNUMCONFIG	= 0x0342,	//上位机缓存张数功能 (DEC: 834)
TULVIDC_STACKNUMCONFIG	= 0x0343,	//上位机堆栈张数功能 (DEC: 835) TIF OR
RAW		
TULVIDC_STACKBLOCKCONFIG	= 0x0344,	//上位机堆栈块数功能 (DEC: 836)
TULVIDC_OUTPUTTRIEDGEEN	= 0x0350,	//触发输出边沿使能功能 (DEC: 848)
TULVIDC_OUTPUTTRIDELAYEN	= 0x0351,	//触发输出延迟使能功能 (DEC: 849)
TULVIDC_OUTPUTTRIWIDTHEN	= 0x0352,	//触发输出脉宽使能功能 (DEC: 850)
TULVIDC_OUTPUTTRIREADYEN	= 0x0353,	//触发 Ready 使能功能 (DEC: 851)
TULVIDC_OUTPUTTRIKINDEN	= 0x0354,	//触发模式使能功能 (DEC: 852)
TULVIDC_OUTPUTTRIPOINT	= 0x0358,	//触发输出端口范围功能 (DEC: 856)
TULVIDC_OUTPUTTRIKIND	= 0x0359,	//触发输出模式范围功能 (DEC: 857)
TULVIDC_TRIINEXPOSURE	= 0x0360,	//触发曝光使能功能 (DEC: 864)
TULVIDC_TRIINEDGE	= 0x0361,	//触发边沿使能功能 (DEC: 865)
TULVIDC_TRIINDELAY	= 0x0362,	//触延迟使能功能 (DEC: 866)
TULVIDC_TRIINSTART	= 0x0363,	//触发 Start 使能功能 (DEC: 867)
TULVIDC_TRIINFILTER	= 0x0364,	//触发 Filter 使能功能 (DEC: 868)

5.1.4. TULV_IDPROP 属性 ID: 用于 TU_SetParameter.vi

TULVIDP_EXPTM	= 0x0100,	//曝光时间 ID (DEC: 256)
TULVIDP_BRIGHTNESS	= 0x0101,	//亮度调节 ID (DEC: 257)
TULVIDP_PIXELRATIO	= 0x0102,	//像素比例调节 ID (DEC: 258)
TULVIDP_IMGMETADATA	= 0x0103,	//图像帧头数据开关 ID (DEC: 259)
TULVIDP_RGAIN	= 0x0105,	//Red 通道 (彩色相机) ID (DEC: 261)

Tucsen 相机 Labview 开发指南

TULVIDP_GGAIN	= 0x010A,	//Greed 通道 (彩色相机) ID(DEC: 266)
TULVIDP_BGAIN	= 0x010F,	//Blue 通道 (彩色相机) ID(DEC: 271)
TULVIDP_SATURATION	= 0x0114,	//饱和度 (彩色相机) ID(DEC: 276)
TULVIDP_GAMMA	= 0x0119,	//伽马值 ID (DEC: 281)
TULVIDP_CONTRAST	= 0x011E,	//对比度 ID (DEC: 286)
TULVIDP_LFTLEVEL	= 0x0123,	//左色阶 ID (DEC: 291)
TULVIDP_RGTLEVEL	= 0x0128,	//右色阶 ID (DEC: 296)
TULVIDP_GLOBGAIN	= 0x012D,	//全局增益 ID (DEC: 301)
TULVIDP_BLACKLEVEL	= 0x012E,	//相机偏置 ID (DEC: 302)
TULVIDP_GAINMODE	= 0x012F,	//相机增益模式值 ID(DEC: 303)
TULVIDP_SHARPNESS	= 0x0132,	//锐化 ID (DEC: 306)
TULVIDP_COLORTEMP	= 0x0137,	//色温 ID (DEC: 311)
TULVIDP_TRIKIND	= 0x013C,	//触发模式 (0:支持同步和全局, 1:不支持同步全局) ID (DEC: 316)
TULVIDP_FRAMERATE	= 0x0141,	// 帧率可调 ID (DEC: 321)
TULVIDP_TEMPERATURE	= 0x0146,	// 温度获取 ID (DEC: 326)
TULVIDP_TEMPERATURETARGET	= 0x0147,	//相机温度值 ID(DEC: 327)

5.1.5. TULV_CAPTURE_MODES 捕获模式 ID: 用于 TU_SetTrigger.vi

TULVCM_SEQUENCE	= 0x00,	//采用序列模式 (流模式)
TULVCM_TRIGGER_STANDARD	= 0x01,	//标准触发模式
TULVCM_TRIGGER_SYNCHRONOUS	= 0x02,	//同步触发模式
TULVCM_TRIGGER_GLOBAL	= 0x03,	//全局触发
TULVCM_TRIGGER_SOFTWARE	= 0x04,	//软件触发

5.1.6. TULVIMG_FORMATS 图像格式 ID: 用于保存图像

TULVFMT_RAW	= 0x01,	// RAW 格式
TULVFMT_TIF	= 0x02,	// TIFF 格式
TULVFMT_PNG	= 0x04,	// PNG 格式
TULVFMT_JPG	= 0x08,	// JPEG 格式
TULVFMT_BMP	= 0x10,	// BMP 格式

5.1.7. TULV_TRIGGER_EXP 触发曝光模式 ID: 用于 TU_SetTrigger.vi

TULVTE_EXPTM	= 0x00,	// 触发使用曝光时间模式
TULVTE_WIDTH	= 0x01,	// 触发使用电平宽度模式

5.1.8. TULV_TRIGGER_EDGE 触发激发边沿模式 ID: 用于 TU_SetTrigger.vi

```

TULVTD_RISING           = 0x01,           // 激发上升沿
TULVTD_FALLING          = 0x00,           // 激发下降沿

```

6. VI 介绍与运用

6.1 TU_InitCamera.vi



输入: error in

输出: error out、ReturnValue。

功能: 这是在使用所有 TUCSEN 控制相机之前使用的第一个 VI 模块, 它的作用是初始化所有已经连接上 USB 串口和电源的 TUCSEN 相机。在调用该 VI 后会获得一个输出值, 若 **ReturnValue** 值为 1, 说明相机初始化成功, 成功后即可调用其他 VI。若 **ReturnValue** 值不为 1, 则说明从初始化相机失败。如何使用该 VI 可参考提供的 Demo。

(注: 在一个程序中如果需要重复调用初始化模块, 在第二次调用之前, 必须要先调用 TU_UninitCamera.vi 才能成功调用 TU_InitCamera.vi)。

输出端说明

ReturnValue: 值为 1 说明调用成功, 不为 1 说明调用失败, 具体失败原因根据 TULVRET 错误代码查找。

6.2 TU_UninitCamera.vi



输入: error in、nCamCount。

输出: error out、ReturnValue。

功能: 在调用该 VI 后, 系统将释放所有相机资源, 如果相机仍处于预览状态, 相机将被强制关闭。相机资源被释放之后, 必须调用 TU_InitCamera.vi 才能再次初始化相机资源。释放资源是否成功, 可根据输出值 ReturnValue 来判断, 若 ReturnValue 值为 1, 说明释放相机资源成功, 若不为 1, 则释放相机资源失败。

输入端说明

nCamCount: 需要退出的相机台数

输出端说明

ReturnValue: 值为 1 说明调用成功，不为 1 说明调用失败，具体失败原因根据 TULVRET 错误代码查找。

6.3 TU_OpenCamera.vi



输入: error in、nCamCount

输出: error out、CameraCount、ReturnValue。

功能: 开启相机操作。在初始化相机资源成功后，TU_OpenCamera.vi 将根据初始化操作过程中的资源分配情况，对多台相机进行开启操作。具体打开相机台数由输入端控制。该步骤的操作成功与否，可根据 ReturnValue 值来判断，值为 1，说明开启相机成功，不为 1 则开启失败。开启成功后，输出端 CameraCount 的值即为开启成功的相机总数值。只有在开启相机成功的情况下，才能对相机进行相应操作，如参数设置、参数范围获取等。如何使用该 VI 可参考示例。

输入端说明

nCamCount: 需要打开的相机台数。

输出端说明

nCamCount: 成功打开的相机台数。

nReturnValue: 值为 1 说明调用成功，不为 1 说明调用失败，具体失败原因根据 TULVRET 错误代码查找。

6.4 TU_SetParameter.vi



输入端: CameraIndex、Parameter、Value in、strInfo、error in。

输出端: ReturnValue、Error out。

功能: 设置相机参数。执行 TU_SetParameter.vi，可对相机进行一系列参数设置，其中包括曝光时间，Gamma 值，增益值，对比度等（具体可设置参数详见 TULV_IDCAPA 性能 ID 和 TULV_IDPROP 属性 ID）。参数是否设置成功，可根据输出端 ReturnValue 值来判断，ReturnValue 值为 1，代表参数设置成功，ReturnValue 值不为 1，代表设置参数失败，具体失败原因，可根据 ReturnValue 值对应 TULVRET 错误代码来查找。如何使用该 VI 可参考示例 SetParameterDemo.vi。

输入端说明

CameraIndex: 所需要操作的相机序列号。该序列号从 0 开始，0 代表第一台相机，1 代表第二台，以此类推。关于如何使相机序列号与相机名称一一对应，可参考示例 LabviewDemo。

Parameter: 需要设置的参数类型相对应的 ID。目前版本所支持的所有参数请参考 TULV_IDCAPA 和 TULV_IDPROP。根据需求后续逐步添加其他参数。

Value in: 需要设置的参数值。参数类型可设置的参数范围参考 TU_GetParameterLimits.vi。性能参数值的含义详见附录。

strInfo: 需要设置的字符串型参数，如图片存储地址等。

输出端说明

ReturnValue: 值为 1 说明调用成功，不为 1 说明调用失败，具体失败原因根据 TULVRET 错误代码查找。

6.5 TU_GetParameter.vi



输入端: CameraIndex、ParameterID、error in。

输出端: ParameterValue、ReturnValue、strInfo、error out。

功能: 获取参数值。执行 TU_GetParameter.vi，可对相机进行一系列参数当前值的值获取，其中包括曝光时间，Gamma 值，增益值，对比度等（具体可设置参数详见 TULV_IDCAPA 性能 ID 和 TULV_IDPROP 属性 ID）。参数是否获取成功，可根据输出端 ReturnValue 值来判断，ReturnValue 值为 1，代表参数获取成功，ReturnValue 值不为 1，代表获取参数失败，具体失败原因，可根据 ReturnValue 值对应 TULVRET 错误代码来查找。如何使用该 VI 可参考示例 Demo。

输入端说明

CameraIndex: 需要获取参数值的所对应的相机序列号。该序列号从 0 开始, 0 代表第一台相机, 1 代表第二台, 以此类推。关于如何使相机序列号与相机名称一一对应, 可参考示例 LabviewDemo。

ParameterID: 需要获取的参数类型 ID。

输出端说明

ParameterValue: 根据输入的参数 ID 获取上来的参数值(数值类型)。

ReturnValue: 值为 1 说明调用成功, 不为 1 说明调用失败, 具体失败原因根据 TULVRET 错误代码查找。

strInfo: 根据输入的参数 ID 获取上来的参数值(字符串类型)。

6.6 TU_GetParameterLimits.vi



输入端: CameraIndex、ParameterID、error in。

输出端: MaxValue、MinValue、StepValue、DefaultValue、ReturnValue、error out。

功能: 获取参数值范围。包括最大值、最小值、默认值、Step 值。参数范围是否获取成功, 可根据输出端 ReturnValue 值来判断, ReturnValue 值为 1, 代表参数范围获取成功, ReturnValue 值不为 1, 代表获取参数范围失败, 具体失败原因, 可根据 ReturnValue 值对应 TULVRET 错误代码来查找。如何使用该 VI 可参考示例 Demo。

输入端说明

CameraIndex: 需要获取的参数值范围所对应的相机序列号。该序列号从 0 开始, 0 代表第一台相机, 1 代表第二台, 以此类推。关于如何使相机序列号与相机名称一一对应, 可参考示例 LabviewDemo。

ParameterID: 需要获取的参数类型 ID。

输出端说明

MaxValue: 输入参数类型对应的最大值。

MinValue: 输入参数类型对应的最小值。

StepValue: 输入参数类型对应的 Step 值。

DefaultValue: 输入参数类型对应的默认值。

ReturnValue: 值为 1 说明调用成功，不为 1 说明调用失败，具体失败原因根据 TULVRET 错误代码查找。

6.7 TU_GetCameraInfo.vi



输入端: CameraIndex、TextIndex、TextID、error in。

输出端: MaxValue、MinValue、StepValue、DefaultValue、ReturnValue、error out。

功能: 获取相机相关信息。包括相机名称、SN 码（相机是否支持）、固件版本、SDK 版本、USB 接口类型（2.0 或 3.0）、分辨率名称。获取信息是否成功，可根据 ReturnValue 输出值来判断。如何使用该 VI 可参考示例 Demo。

输入端说明

CameraIndex: 需要获取参数信息所对应的相机序列号。该序列号从 0 开始，0 代表第一台相机，1 代表第二台，以此类推。关于如何使相机序列号与相机名称一一对应，可参考示例 Demo。

TextID: 需要获取的相机信息所对应的 ID（参考 TULV_IDINFO）。

TextIndex: 该输入端针对一个信息 ID 下包含多条相机信息，例如相机名称为 ISH130，该相机有两个分辨率，TextID 为 TULVIDI_RESOLUTION 时，要想显示所有分辨率，则需要分别设置 TextIndex 为 0，1。详情可参考示例 Demo。

输出端说明

CameraText: 根据 TextID 和 TextIndex 而输出的相机相关信息。

ReturnValue: 值为 1 说明调用成功，不为 1 说明调用失败，具体失败原因根据 TULVRET 错误代码查找

6.8 TU_SetTrigger.vi



输入端：CameraIndex、CaptureMode、ExpourseMode、EdgeMode、DelayTime、Frame、nBufFrame、。

输出端：ReturnValue。

功能：相机外触发设置（该功能只针对支持外触发的相机进行开放，不支持的相机即使调用该模块也无效）。关于外触发的介绍详见文档（3.4.1 捕获模式）和（3.4.3 触发模式）。如何使用该 VI 可参考示例 TriggerDemo.vi。

输入端说明

CameraIndex：需要设置外触发所对应的相机序列号。该序列号从 0 开始，0 代表第一台相机，1 代表第二台，以此类推。关于如何使相机序列号与相机名称一一对应，可参考示例 TriggerDemo.vi。

CaptureMode：该输入端包含有 5 种模式，分别是 TULVCM_SEQUENCE、TULVCM_TRIGGER_STANDERD、TULVCM_TRIGGER_SYNCHRONOUS、TULVCM_TRIGGER_GLOBAL、TULVCM_TRIGGER_SOFTWARE，参数说明可参考 3.4.3 触发模式介绍。

ExpourseMode：该输入端包含两种模式分别为 TULVTE_EXPTM 和 TULVTE_WIDTH。参数说明可参考 3.4.3 触发模式介绍。

EdgeMode：该输入端包含两种模式分别为 TULVTD_RISING 和 TULVTD_FAILING。参数说明可参考 3.4.3 触发模式介绍。

DelayTime：当 ExpourseMode 选择 TULV_TEPTM 时，DelayTime 值决定在接收到触发信号后多久出图。

Frame：一次触发出图总张数（使用触发板进行触发时可不设置该参数）。

nBufFrame：SDK 的缓存图像张数。

输出端说明

ReturnValue：值为 1 说明调用成功，不为 1 说明调用失败，具体失败原因根据 TULVRET 错误代码查找。

6.9 TU_GetTrigger.vi



输入端：CameraIndex。

输出端：ReturnValue。

功能：获取外触发参数和判断是否支持外触发。该 VI 根据输入相机序列号来判断是否支持外触发模式。如何使用该 VI 可参考示例 LabviewDemo。

(注：TU_GetTrigger.vi 可根据客户需求自行更改该 VI，获取更多输出信息，包括当前出发模式、曝光模式、延时时长、电平触发模式)

输入端说明

CameraIndex：需要获取外触发值所对应的相机序列号。该序列号从 0 开始，0 代表第一台相机，1 代表第二台，以此类推。关于如何使相机序列号与相机名称一一对应，可参考示例 SetTriggerDemo.vi。

输出端说明

ReturnValue：值为 1 说明该相机支持外触发模式且调用成功，不为 1 说明调用失败和不支持外触发，具体失败原因根据 TULVRET 错误代码查找。

6.10 TU_SetROI.vi



输入端：CameraIndex、StartX、StartY、Width、Height、SetROI。

输出端：ReturnValue。

功能：设置 ROI 模式。该 VI 可以在允许范围内自行选定想要的区域进行数据采集。调用示例参考 ROI Demo.vi

输入端说明

CameraIndex：需要设置 ROI 模式所对应的相机序列号，该序列号从 0 开始，0 代表第一台相机，1 代表第二台，以此类推。关于如何使相机序列号与相机名称一一对应，可参考示例 ROI Demo.vi。

StartX：水平偏移量。

StartY: 垂直偏移量。

Width: 水平宽度。

Height: 垂直高度。

SetROI: 是否设置 ROI 模式。0、关闭， 1、开启。

输出端说明

ReturnValue: 值为 1 说明设置 ROI 模式成功，不为 1 说明设置 ROI 失败，具体失败原因可根据 TULVRET 错误代码查找。

6.11 TU_StartCapture.vi



输入端: CameraIndex、CaptureMode。

输出端: CameraIndex、ImageType、Width、Height、Channel、Depth。

功能: 调用 TU_InitCamera.vi 和 TU_OpenCamera.vi 成功之后，再调用 TU_StartCapture.vi 来对内存空间进行分配，分配成功后开始进行数据捕获。如何使用该 VI 可参考示例 Demo。

输入端说明

CameraIndex: 需要进行数据捕获的相机序列号。

CaptureMode: 数据捕获模式（流模式或触发模式）。输入类型详见 TULV_CAPTURE_MODE

输出端说明

CameraIndex: 数据捕获成功的相机序列号。

ImageType: 输出图像类型。用于 IMAQ Create 模块。用法详见示例 Demo。

Width: 输出图像宽度。

Height: 输出图像高度。

Channel: 输出图像通道数（彩色图像或黑白图像）。

Depth: 输出图像数据位宽（8 位或 16 位）。

6.12 TU_StopCapture.vi



输入端: CameraIndex。

输出端: ReturnValue。

功能: 停止数据捕获。根据输入的相机序列号来进行停止数据捕获的处理。 TU_UninitCam 的前提下, 可通过调用 TU_StartCapture.vi 实现数据的再次捕获。如何使用该 VI 可参考示例 Demo。

输入端说明

CameraIndex: 需要进行停止数据捕获的相机序列号。

输出端说明

ReturnValue: 值为 1 说明停止数据捕获成功, 值不为 1 则停止数据捕获失败, 具体失败原因可根据 TULVRET 错误代码查找。

6.13 TU_WaitForFrame.vi



输入端: CameraIndex。

输出端: nFrameIndex、dblTimeStamp、dblTimeLast、ReturnValue。

功能: 用于等待数据捕获的完成。具体使用方法可参考 Demo

输入端说明

CameraIndex: 需要进行处理的相机序列号。

输出端说明

ReturnValue: 值为 1 说明数据捕获完成, 值不为 1 说明数据捕获失败, 具体失败原因可根据 TULVRET 错误输出查找。

nFrameIndex: 时间戳 ID

dblTimeStamp:时间戳

dblTimeLast: 时间戳的持续时间

6.14 TU_GetFrame.vi



输入端: CameraIndex、ImageSRC_Average、ImageSRC_BKGD、ImageSRC_Normal、AverageCntIn、bSetBKGD、nFuncSelect。

输出端: calBKGD、calAverage、Image_BKGD、Image_Average、Image_Normal。

功能: 获取帧数据用预览画面的实时显示。 具体使用方法可参考示例 Demo。

输入端说明

CameraIndex: 需要进行获取帧数据的相机序列号。

ImageSRC_Average: 平均值输入源图像。

ImageSRC_BKGD: 减背景输入源图像。

ImageSRC_Normal: 正常输入源图像。

AverageCntIn: 求平均张数(0-99)。

BSetBKGD: 是否减背景(0:否 1:是)。

nFuncSelect: 功能选择。(0:正常输出 1: 减背景 2:求平均)

输出端说明

calBKGD: 是否计算完成减背景。

calAverage: 是否计算完成求平均。

Image_BKGD: 减背景输出。

Image_Average: 求平均输出。

Image_Normal: 正常输出。

(注：一个 GetFrame 中只能对应一种输出方式)

6.15 TU_ErrorReport.vi



功能完善中。。。。。。。。。

6.16 TU_OpenImageFile.vi

输入端： NULL

输出端： bCanceled、 Image。

功能： 打开图片(用于减背景)。

输出端说明

Image： 打开的图片（可直接显示）。

bCanceled： 是否打开图片（用于减背景时的计算）（0:取消 1:打开图片）。

6.17 TU_StartRecorder.vi

输入端： nCamIndex

输出端： ReturnValue

功能说明： 开始录制视频文件

输入端说明

nCamIndex： 需要进行获取帧数据的相机序列号。

输出端说明

ReturnValue： 是否开始录制视频成功（0:失败 1:成功）。

6.18 TU_StopRecorder.vi

输入端: nCamIndex

输出端: ReturnValue

功能说明: 停止录制视频文件

输入端说明

nCamIndex: 需要进行获取帧数据的相机序列号。

输出端说明

ReturnValue: 是否停止录制视频成功 (0:失败 1:成功)。

6.19 TU_SetOutPutTrigger.vi



输入端: CameraIndex、OutPort、OutMode、EdgeMode、DelayTime、OutWidth。

输出端: ReturnValue。

功能: 相机外触发输出设置 (该功能只针对支持外触发输出的相机进行开放, 不支持的相机即使调用该模块也无效)。如何使用该 VI 可参考示例 TriggerDemo.vi。

输入端说明

CameraIndex: 需要设置外触发所对应的相机序列号。该序列号从 0 开始, 0 代表第一台相机, 1 代表第二台, 以此类推。关于如何使相机序列号与相机名称一一对应, 可参考示例 SetTriggerDemo.vi。

OutPort: 该输出端口模式, 分别是 Port1、Port2、Port3 端口可供选择参数。

OutMode: 该项是触发输出模式不同相机有不同的触发输出模式。具体可参考 TriggerDemo.vi 和 LabviewSample.vi 的模块例子。

EdgeMode: 该输入端包含两种模式分别为 TULVTD_RISING 和 TULVTD_FAILING。

DelayTime: 择 TULV_TEPTM 时, DelayTime 值决定在触发输出信号的延时。

OutWidth: 值决定了触发输出信号的脉宽宽度。

输出端说明

ReturnValue: 值为 1 说明调用成功，不为 1 说明调用失败，具体失败原因根据 TULVRET 错误代码查找。

6.20 TU_GetOutPutTrigger.vi



输入端: CameraIndex。

输出端: ReturnValue。

功能: 获取外触发输出参数和判断是否支持外触发输出。该 VI 根据输入相机序列号来判断是否支持外触发输出模式。如何使用该 VI 可参考示例 TriggerDemo.vi。

(注: TU_GetOutPutTrigger.vi 可根据客户需求自行更改该 VI，获取更多输出信息，包括当前出端口、输出模式、延时时长、电平触发模式、脉宽时长)

输入端说明

CameraIndex: 需要获取外触发值所对应的相机序列号。该序列号从 0 开始，0 代表第一台相机，1 代表第二台，以此类推。关于如何使相机序列号与相机名称一一对应，可参考示例 SetTriggerDemo.vi。

输出端说明

ReturnValue: 值为 1 说明该相机支持外触发输出模式且调用成功，不为 1 说明调用失败和不支持外触发输出功能，具体失败原因根据 TULVRET 错误代码查找。

6.21 TU_StartMultiData.vi



输入端: CameraIndex、StackNum、Format。

输出端: ReturnValue。

功能: 设置堆栈拍图格式 (TIFF、RAW)，堆栈数量，开启堆栈拍图。

输入端说明

CameraIndex: 需要获取外触发值所对应的相机序列号。该序列号从 0 开始，0 代表第一台

相机，1 代表第二台，以此类推。

StackNum: 表示设置堆栈的数量。

Foramat: 表示拍图格式（当前只支持 TIFF 和 RAW），设置 1 表示拍 RAW，设置 2 表示拍 TIF。

输出端说明

ReturnValue: 值为 1 说明该相机支持外触发输出模式且调用成功，不为 1 说明调用失败和不支持外触发输出功能，具体失败原因根据 TULVRET 错误代码查找。

6.22 TU_StopMultiData.vi



输入端: CameraIndex。

输出端: ReturnValue。

功能: 停止堆栈拍图。

输入端说明

CameraIndex: 需要获取外触发值所对应的相机序列号。该序列号从 0 开始，0 代表第一台相机，1 代表第二台，以此类推。

输出端说明

ReturnValue: 值为 1 说明该相机支持外触发输出模式且调用成功，不为 1 说明调用失败和不支持外触发输出功能，具体失败原因根据 TULVRET 错误代码查找。

附录

性能 ID	可设参数值	参数含义
TULVIDC_AUTOWB	0, 1, 2	0: 关闭自动白平衡 1: 一次白平衡 2: 自动白平衡
TULC IDC_AUTOEXP	0, 1, 2	0: 关闭自动曝光 1: 开启自动曝光 2: 一次曝光
TULVIDC_DEPTH	0, 1	0: 8 位数据宽度 1: 16 位数据宽度
TULVIDC_AUTOLEVEL	0, 1, 2, 3	0: 关闭自动色阶 1: 自动左色阶 2: 自动右色阶 3: 自动左右色阶
TULVIDC_RESOLUTION 率,	0-n	具体有几个分辨率依相机而定, 0 为第一个分辨率, 1 为第二个分辨率, 以此类推
TULVIDC_NOISELEVEL	0, 1, 2, 3	0: Off 1: Low 2: Medium 3: High
TULVIDC_IMGMODESELECT	0, 1,	0: 关闭 CMS 1: 开启 CMS
TULVIDC_SAVEFRAME	n	用于拍照时保存图片, 具体存几张依实际设置值而定
TULVIDC_IMGSAVEPATH	字符串	用于选择存储图片路径
TULVIDC_IMGFORMAT	1, 2, 4, 8, 16	1: RAW 2:TIF 4:PNG 8:JPG 16:BMP
TULVIDC_IMGCOUNT	0-n	存储图片张数
TULVIDC_CAMERASTATE		该性能 ID 不需要特定设置参数值, 只要调用了 TU_StartCapture.vi 或 TU_StopCapture.vi 后即可 通过调用 TU_GetParameter.vi 来获取当前相机状态 0: 预览状态 1: 挂起状态

属性 ID	可设参数值
TULVIDP_EXPTM	MinValue-MaxValue
TULVIDP_RGAIN	MinValue-MaxValue
TULVIDP_GGAIN	MinValue-MaxValue
TULVIDP_BGAIN	MinValue-MaxValue
TULVIDP_SATURATION	MinValue-MaxValue
TULVIDP_GAMMA	MinValue-MaxValue
TULVIDP_CONTRAST	MinValue-MaxValue
TULVIDP_LFTLEVEL	MinValue-MaxValue
TULVIDP_RGTLEVEL	MinValue-MaxValue
TULVIDP_GLOBGAIN	MinValue-MaxValue
TULVIDP_SHARPNESS	MinValue-MaxValue
TULVIDP_COLORTEMP	MinValue-MaxValue

(注: 以上参数可设的前提条件为相机支持以上功能, MinValue 与 MaxValue 值可调用 TU_GetParameterLimits.vi 来获取)