



The LabVIEW Development Guide for Tucsen Cameras

Copyright (c) 2011-2025

Tucsen Photonics Co., Ltd.

Content

1. Read Before Use	1
2. Introduction	2
3. Overview	2
3.1 Layer Structure	2
3.2 Principle	3
3.3 Interface Types	3
3.4 Terminology	3
3.4.1 Capture Mode	3
3.4.2 Image Unit	4
3.4.3 Trigger Mode	4
3.4.4 Camera Status	5
4. Version Introduction	5
4.1. Development Environment	5
4.2. Performance Optimization	5
4.2.1 Interface quantity and call optimization	5
4.2.2 VI module and call optimization	5
4.3. Function addition	6
4.3.1 Camera-Related Information Display	6
4.3.2 External Trigger	6
4.3.3 Multi-Camera Operation	7
5. Reference	8
5.1 Types and Constants	8
5.1.1 TULVRET Error Codes: Used for TU_ErrorReport.vi	8

LabVIEW Development Guide

5.1.2 TULV_IDINFO Product Information ID: Used for TU_GetCameraInfo.vi.....	11
5.1.3. TULV_IDCAPA Performance ID: Used for TU_SetParameter.vi	11
5.1.4. TULV_IDPROP Attribute ID: Used for TU_SetParameter.vi	14
5.1.5.TULV_CAPTURE_MODES Capture Mode ID: Used in TU_SetTrigger.vi.....	15
5.1.6.TULVIMG_FORMATS Image Format ID: Used for saving images	15
5.1.7. TULV_TRIGGER_EXP Trigger Exposure Mode ID: Used in TU_SetTrigger.vi.....	16
5.1.8. TULV_TRIGGER_EDGE Trigger Edge Mode ID: Used in TU_SetTrigger.vi	16
6. VI Introduction and application	17
6.1 TU_InitCamera.vi.....	17
6.2 TU_UninitCamera.vi	17
6.3 TU_OpenCamera.vi	18
6.4 TU_SetParameter.vi.....	19
6.5 TU_GetParameter.vi	20
6.6 TU_GetParameterLimits.vi	21
6.7 TU_GetCameraInfo.vi	22
6.8 TU_SetTrigger.vi.....	23
6.9 TU_GetTrigger.vi	25
6.10 TU_SetROI.vi	26
6.11 TU_StartCapture.vi.....	27
6.12 TU_StopCapture.vi	28
6.13 TU_WaitForFrame.vi	28
6.14 TU_GetFrame.vi	29
6.15 TU_ErrorReport.vi	30
6.16 TU_OpenImageFile.vi	30
6.17 TU_StartRecorder.vi	31

LabVIEW Development Guide

6.18 TU_StopRecorder.vi	31
6.19 TU_SetOutPutTrigger.vi	32
6.20 TU_GetOutPutTrigger.vi	33
6.21 TU_StartMultiData.vi	34
6.22 TU_StopMultiData.vi	35
Appendix	36

1. Read Before Use

This document and sample code are internal files and public content of TUCSEN, provided to help users develop applications using TUCSEN digital cameras.

This document and sample code are disclosed only for the purpose mentioned above and do not constitute any license, transfer, or other rights from the owner.

All risks and consequences of using the software documentation remain the responsibility of the user.

This document may include technical errors or printing mistakes, and TUCSEN does not guarantee against damages arising from such errors or text.

TUCSEN does not commit to updating or maintaining the information contained in this document.

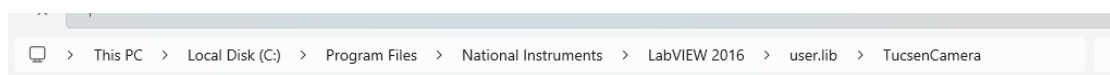
All brand and product names are trademarks or registered trademarks of their respective owners.

TUCSEN reserves all copyright rights to this document.

Without prior written permission from TUCSEN, no part of this document may be reproduced, transmitted, transcribed, stored in a retrieval system, or translated into any language or computer language in any form or by any means—whether electronic, mechanical, electromagnetic, optical, chemical, manual, or otherwise.

Usage Instructions for This Version

Due to limitations in the current version, to use it properly, you must copy the "TucsenCamera" folder from the compressed package to the "user.lib" folder within the LabVIEW installation path to access the corresponding VI modules. Future versions will package and install the relevant contents, omitting the need to copy folders manually.



The image below shows an example path for storing the corresponding folder

We apologize for any inconvenience this may cause and are working to make improvements as quickly as possible to better meet user needs.

2.Introduction

This manual provides a detailed description of operating TUCSEN digital cameras using the TULV_API specifications. The TULV_API Software Development Kit is referred to as the “SDK,” and the part of TULV_API that controls the digital camera is called the “module.”

The SDK includes VI modules and a sample application, demonstrating how to develop applications for the digital camera using VI modules. SDK users are free to use the software in any way they like, including partial modification of VI constructs or creating entirely independent projects.

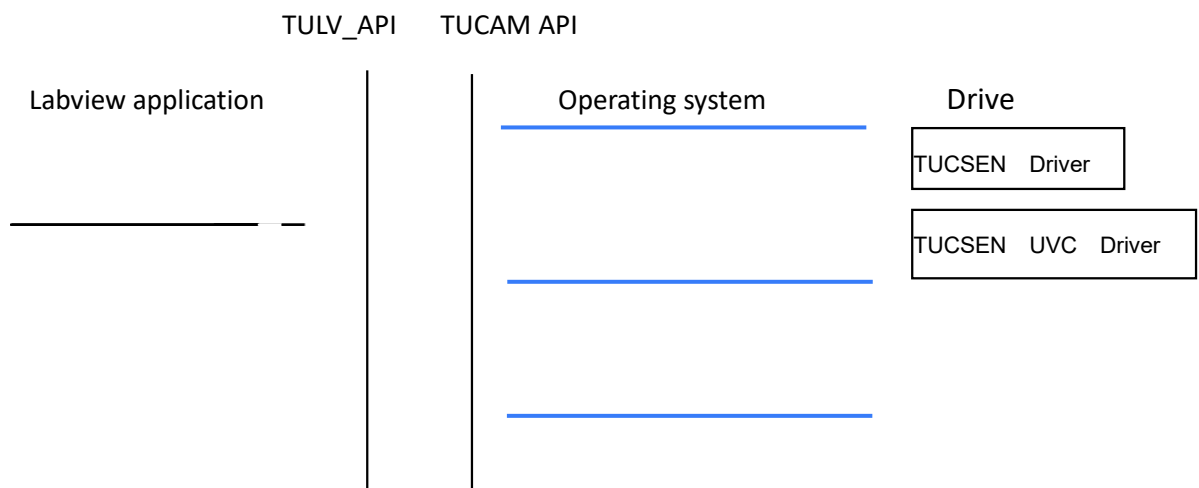
This SDK is designed for easy understanding. For this reason, the number of function interfaces is kept to a minimum, and the function call format is written in C language style.

Some extended functions are additional features available for certain specific digital cameras.

The values for different digital cameras may vary depending on the model used to capture images. These values should be considered as guidelines rather than exact figures.

3. Overview

3.1 Layer Structure



The TUCSEN digital camera connects to digital camera drivers on different operating systems via the SDK to control the camera and capture image data.

Currently, the SDK supports only Windows systems.

3.2 Principle

The specific bus interface and library of the digital camera are encapsulated by TULV_API. The interface calls in each module's VI section originate from TULV_API's interface. You can refer to the provided LabviewDemo to understand how to use these interfaces. For usage details, see section (6. Introduction and Application of VI).

3.3 Interface Types

TULV_API functions can be categorized into several types:

- Start/stop processing
- Camera Information collection
- Performance/Property setting and adjustment
- Memory management
- Capture control
- File Management
- Extended Control

3.4 Terminology

3.4.1 Capture Mode

The camera offers two types of capture modes:

Sequence Mode (Streaming Mode): Used to capture continuous image data.

Trigger Mode: The camera captures images based on an external signal. This mode is known as "Trigger Mode," which can be configured using `TU_SetTrigger()`. External signals used in this mode are also referred to as "External Triggers."

3.4.2 Image Unit

Generally, this is two-dimensional, with vertical and horizontal directions.

Frame: A unit of image data. For each frame, pixel data is aligned from left to right and top to bottom. This forms a series of image data units.

3.4.3 Trigger Mode

Standard Mode: When the camera receives a level signal (determined by the active edge), it begins capturing one or multiple frames, with the number of frames defined by the configuration parameters.

Synchronization Mode: When the camera receives a level signal (determined by the active edge), it starts exposure. When the opposite level signal is received, exposure ends, and image data is captured, achieving full synchronization of each frame's exposure and readout with the external trigger signal.

Global Mode: Pre-triggering begins before the camera is triggered. When the camera receives a level signal (determined by the active edge) or a software-set exposure time, it ends the current reset operation. When exposure ends, image data is captured, and pre-triggering restarts. This mode is used to implement global exposure mode for cameras with rolling shutter.

Exposure Mode

Exposure Time: The exposure time is determined by `TULVIDP_EXPTM` after receiving a trigger signal.

Level Width: The exposure time is determined by the width of the level signal after receiving a trigger signal.

Note: In Standard and Global modes, both options can be configured. In Synchronization mode, only level width can be set.

Trigger Signal Type

Rising Edge: Exposure starts when the trigger level is on the rising edge.

Falling Edge: Exposure starts when the trigger level is on the falling edge.

3.4.4 Camera Status

The camera's status determines which functions can be called. Certain functions may change the camera's status. Four camera statuses are described below:

Unstable: Parameters are set and other functions can be called, but they are not in a fixed state.

Stable: Parameters and functions are set, but image capture cannot start since frame memory has not been created.

Ready: Frame memory has been created, and image capture can begin.

Busy: Image capture is in progress.

4. Version Introduction

4.1. Development Environment

Currently, only LabVIEW development on Windows is supported. Developers can select either the x86 or x64 installation package based on their requirements.

Note: The new version of the Labview plug-in only supports the X64 package.

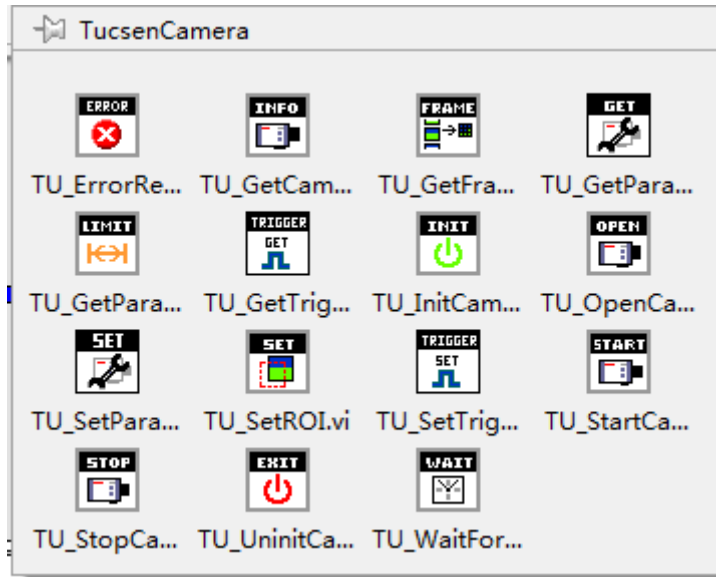
4.2. Performance Optimization

4.2.1 Interface quantity and call optimization

Version 2.0 includes systematic optimization of interface functions, reducing the number of interface functions without removing core functionality from version 1.0, simplifying user access.

4.2.2 VI module and call optimization

In version 2.0, the single VI module from version 1.0 has been divided into multiple sub-VIs. These can be directly called in the LabVIEW program block diagram, making it easier for developers to analyze the LabviewSample example program in parts. For details on VI modules, see section (6. VI Introduction and Application.)



(Figure 4.2.1)

4.3. Function addition

4.3.1 Camera-Related Information Display

To accurately identify the camera, version 2.0 includes displays for camera name, TYPE, SN code, SDK version, and firmware version. (A 0x210 code indicates USB 2.0; 0x03 indicates FireBird; 0x04 indicates Euresys, and others represent USB 3.0. Cameralink does not display SN and firmware version.)

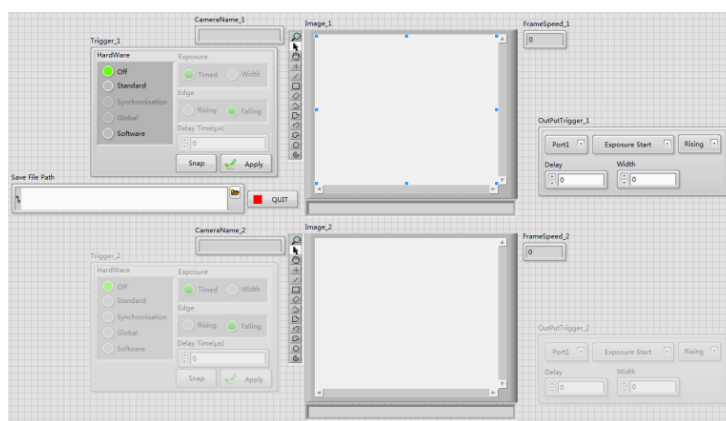


(Figure 4.3.1)

4.3.2 External Trigger

Based on the camera's capabilities, version 2.0 adds an external trigger module and a trigger output module to the SetTriggerDemo.VI example program. See section 3.4.3, "Trigger Mode," for an introduction to trigger modes, and refer to TU_SetTrigger.VI for

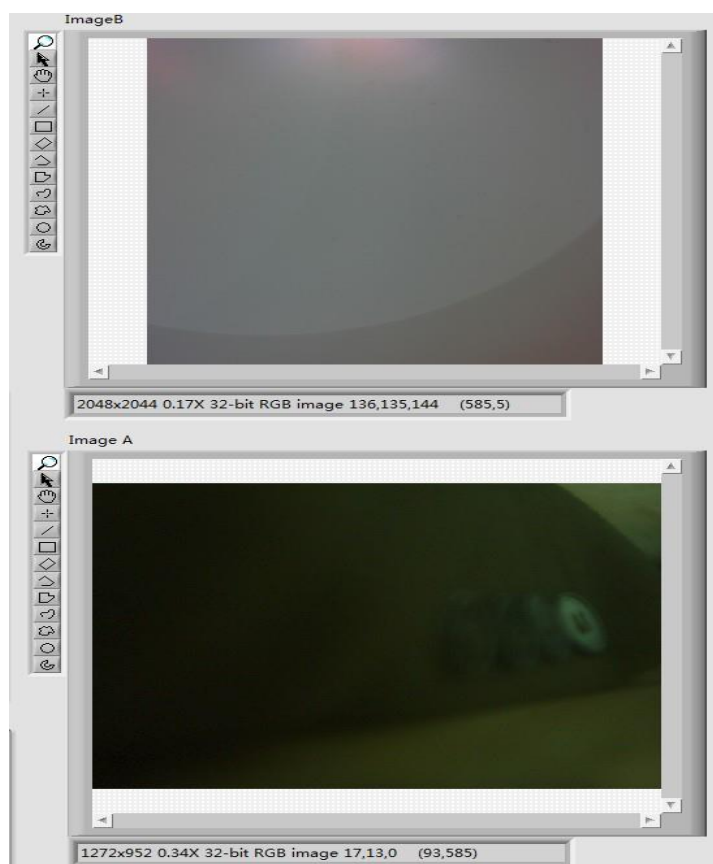
trigger usage.



(Figure 4.3.2)

4.3.3 Multi-Camera Operation

The version 2.0 LabVIEW example program supports simultaneous preview for multiple cameras. Additionally, camera parameters can be set by camera name, allowing users to adjust preview settings for comparing effects across multiple cameras.



(Figure 4.3.3)

5.Reference

5.1 Types and Constants

5.1.1 TULVRET Error Codes: Used for TU_ErrorReport.vi

TULVRET_SUCCESS	= 0x00000001,	// Error-free, generally successful code
TULVRET_FAILURE	= 0x80000000,	// Error
//initialization error		
TULVRET_NO_MEMORY	= 0x80000101,	// Not enough memory
TULVRET_NO_RESOURCE	= 0x80000102,	// Not enough resources (not including memory)
TULVRET_NO_MODULE	= 0x80000103,	// No submodules
TULVRET_NO_DRIVER	= 0x80000104,	// No drive
TULVRET_NO_CAMERA	= 0x80000105,	// No camera
TULVRET_NO_GRABBER	= 0x80000106,	// No picture taken
TULVRET_NO_PROPERTY	= 0x80000107,	// No substitute attribute ID
TULVRET_FAILOPEN_CAMERA	= 0x80000110,	// Failed to open camera
TULVRET_FAILOPEN_BULKIN	= 0x80000111,	// Failed to open the batch transport input endpoint
TULVRET_FAILOPEN_BULKOUT	= 0x80000112,	// Failed to open the batch transport output endpoint
TULVRET_FAILOPEN_CONTROL	= 0x80000113,	// Failed to open control endpoint
TULVRET_FAILCLOSE_CAMERA	= 0x80000114,	// Failed to close the camera
TULVRET_FAILOPEN_FILE	= 0x80000115,	// Failed to open the file

LabVIEW Development Guide

// status error

TULVRET_INIT	= 0x80000201,	// API needs initialization status.
TULVRET_BUSY	= 0x80000202,	// API is busy
TULVRET_NOT_INIT	= 0x80000203,	// API not initialized
TULVRET_EXCLUDED exclusively	= 0x80000204,	// Some resources are used
TULVRET_NOT_BUSY	= 0x80000205,	// API is not busy
TULVRET_NOT_READY	= 0x80000206,	// API is not ready

//wait error

TULVRET_ABORT	= 0x80000207,	// Termination processing
TULVRET_TIMEOUT	= 0x80000208,	// Time out
TULVRET_LOSTFRAME	= 0x80000209,	// Frame loss
TULVRET_MISSFRAME problem	= 0x8000020A,	// Frame loss but it is the underlying driver

// calling error

TULVRET_INVALID_CAMERA	= 0x80000301,	// Invalid camera
TULVRET_INVALID_HANDLE	= 0x80000302,	// Invalid camera handle
TULVRET_INVALID_OPTION	= 0x80000303,	// Invalid configuration values
TULVRET_INVALID_IDPROP	= 0x80000304,	// Invalid attribute ID
TULVRET_INVALID_IDCAPA	= 0x80000305,	// Invalid performance ID
TULVRET_INVALID_IDPARAM	= 0x80000306,	// Invalid parameter ID
TULVRET_INVALID_PARAM	= 0x80000307,	// Invalid parameter
TULVRET_INVALID_FRAMEIDX	= 0x80000308,	// Invalid frame number
TULVRET_INVALID_VALUE	= 0x80000309,	// Invalid values

LabVIEW Development Guide

TULVRET_INVALID_EQUAL = 0x8000030A, // The values are equal, but the parameters are invalid

TULVRET_INVALID_CHANNEL = 0x8000030B, // The attribute ID specifies the channel, but the channel is invalid

TULVRET_INVALID_SUBARRAY = 0x8000030C, // The value of the subarray is invalid

TULVRET_INVALID_VIEW = 0x8000030D, // Invalid display window handle

TULVRET_INVALID_PATH = 0x8000030E, // Invalid file path

TULVRET_NO_VALUETEXT = 0x80000310, // No text for the property value

TULVRET_OUT_OF_RANGE = 0x80000311, // Value out of range

TULVRET_NOT_SUPPORT = 0x80000312, // Unsupported feature or property

TULVRET_NOT_WRITABLE = 0x80000313, // Property is not writable

TULVRET_NOT_READABLE = 0x80000314, // Property is not readable

TULVRET_WRONG_HANDSHAKE = 0x80000410, // Error occurred during error code retrieval

TULVRET_NEWAPI_REQUIRED = 0x80000411, // Only supported by the new API, old API not supported

TULVRET_ACCESSDENY = 0x80000412, // Property inaccessible in the current camera state

TULVRET_NO_CORRECTIONDATA = 0x80000501, // No correction data

// Camera or bus trouble

TULVRET_FAIL_READ_CAMERA = 0x83001001, // Failed to read from the camera

TULVRET_FAIL_WRITE_CAMERA = 0x83001002, // Failed to write to the camera

TULVRET_OPTICS_UNPLUGGED = 0x83001003, // Optics unplugged

5.1.2 TULV_IDINFO Product Information ID: Used for

TU_GetCameraInfo.vi

TULVIDI_NAME	= 0x0001,	// Camera name
TULVIDI_SN	= 0x0002,	// Serial number (SN)
TULVIDI_FS	= 0x0003,	// Firmware version
TULVIDI_USB	= 0x0004,	// USB interface type (USB2.0/USB3.0)
TULVIDI_SDKNO	= 0x0005,	// API version
TULVIDI_RESOLUTION	= 0x0006,	// Resolution display
TULVIDI_PGAAHIGH	= 0x0007,	// PGA high display
TULVIDI_PGALOW	= 0x0008,	// PGA low display
TULVIDI_IMGMODE	= 0x0009,	// Capture card mode display
TULVIDI_FANMODE	= 0x000A,	// Fan information display
TULVIDI_IMGTESTMODE	= 0x000B,	// Test image display
TULVIDI_GAINMODE	= 0x000C,	// Gain mode display
TULVIDI_RECORDCOMPRESS	= 0x000D,	// Video compression format
TULVIDI_SUMBINNING	= 0x000E,	// Cumulative binning display
TULVIDI_SHUTTER	= 0x000F,	// Rolling/global shutter display
TULVIDI_DEPTH	= 0x0010,	// Bit depth display
TULVIDI_OUTPUTPORT	= 0x0011,	// Trigger output port display
TULVIDI_OUTPUTKIND	= 0x0012,	// Trigger output mode display
TULVIDI_CAMMODE	= 0x0013,	// Camera mode display

5.1.3. TULV_IDCAPA Performance ID: Used for TU_SetParameter.vi

TULVIDC_AUTOWB = 0x0200, // Auto white balance ID (DEC: 512)

LabVIEW Development Guide

TULCIDC_AUTOEXP	= 0x0205,	// Auto exposure ID (DEC: 517)
TULVIDC_DEPTH	= 0x020A,	// Data bit width ID (DEC: 522)
TULVIDC_AUTOLEVEL	= 0x020F,	// Auto color level ID (DEC: 527)
TULVIDC_RESOLUTION	= 0x0214,	// Resolution ID (DEC: 532)
TULVIDC_NOISELEVEL	= 0x0219,	// Noise reduction level ID (DEC: 537)
TULVIDC_IMGMODESELECT	= 0x021E,	// CMS (or 11-bit) ID (DEC: 542)
TULVIDC_IMGSAVEPATH	= 0x0223,	// Image save path ID (DEC: 547)
TULVIDC_IMGNAME	= 0x0224,	// Image name ID (DEC: 548)
TULVIDC_LEDENBALE	= 0x0228,	// LED control ID (DEC: 552)
TULVIDC_PIENBALE	= 0x0229,	// PI heating film enable ID (DEC: 553)
TULVIDC_BLACKBALANCE	= 0x022D,	// Black balance ID (DEC: 557)
TULVIDC_TESTIMGMODE	= 0x022E,	// Test image ID (DEC: 558)
TULVIDC_SENSORRESET	= 0x022F,	// Sensor reset 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,	// Save image status ID (DEC: 768)
TULVIDC_IMGFORMAT	= 0x0301,	// Image save format ID (DEC: 769)
TULVIDC_IMGCOUNT	= 0x0302,	// Number of images saved ID (DEC: 770)
TULVIDC_TRIGGERSAVE	= 0x0303,	// Trigger save image status ID (DEC: 771)
TULVIDC_TRIGGERBUFSAVE	= 0x0304,	// Trigger buffer save status ID (DEC: 772)
TULVIDC_CAMERASTATE	= 0x0305,	// Camera state (0, 1) ID (DEC: 773)
TULVIDC_FAN	= 0x0306,	// Camera fan level ID (DEC: 774)
TULVIDC_PGAAHIGH	= 0x030B,	// Camera PGAAHIGH ID (DEC: 779)
TULVIDC_PGALOW	= 0x0310,	// Camera PGALOW ID (DEC: 784)

LabVIEW Development Guide

TULVIDC_PIXCLK1_EN	= 0x0315,	// Camera PIXCLK1EN ID (DEC: 789)
TULVIDC_PIXCLK2_EN	= 0x031A,	// Camera PIXCLK2EN ID (DEC: 794)
TULVIDC_TRIOUT_EN	= 0x031B,	// Camera trigger output enable switch ID (DEC: 795)
TULVIDC_ROLLINGSCANMODE	= 0x031C,	// Camera light-sheet mode selection ID (DEC: 796)
TULVIDC_ROLLINGSCANLTD	= 0x031D,	// Camera light-sheet mode line time delay ID (DEC: 797)
TULVIDC_ROLLINGSCANSLIT	= 0x031E,	// Camera light-sheet mode slit height ID (DEC: 798)
TULVIDC_ROLLINGSCANDIR	= 0x031F,	// Camera light-sheet mode scan direction ID (DEC: 799)
TULVIDC_ROLLINGSCANRESET	= 0x0320,	// Camera light-sheet mode reset ID (DEC: 800)
TULVIDC_ROLLINGSCANINT	= 0x0321,	// Camera light-sheet mode interval time ID (DEC: 801)
TULVIDC_TECENABLE	= 0x0322,	// Camera cooling switch ID (DEC: 802)
TULVIDC_CAMSTART	= 0x0323,	// Camera Start image capture count setting function ID (DEC: 803)
TULVIDC_CAMTRISTOP	= 0x0324,	// Camera trigger stop (DEC: 804)
TULVIDC_CAMTRICNT	= 0x0325,	// Camera trigger count (DEC: 805)
TULVIDC_CAMTRIIDX	= 0x0326,	// Camera trigger index (DEC: 806)
TULVIDC_CAMACQFPS	= 0x0327,	// Camera image acquisition frame rate (DEC: 807)
TULVIDC_CAMSHOWFPS	= 0x0328,	// Camera display image frame rate (DEC: 808)
TULVIDC_SIGNALFILTER	= 0x0329,	// Camera trigger filter (DEC: 809)
TULVIDC_RECORDCOMPRESS	= 0x0340,	// Camera video compression function (DEC: 832)
TULVIDC_RECORDCONFIG	= 0x0341,	// Camera video configuration function (DEC: 833)
TULVIDC_BUFNUMCONFIG	= 0x0342,	// Host machine buffer image count function (DEC: 834)

LabVIEW Development Guide

TULVIDC_STACKNUMCONFIG = 0x0343, // Host machine stack image count function (DEC: 835)
TIF OR RAW

TULVIDC_STACKBLOCKCONFIG = 0x0344, // Host machine stack block count function (DEC: 836)

TULVIDC_OUTPUTTRIEDGEEN = 0x0350, // Trigger output edge enable function (DEC: 848)

TULVIDC_OUTPUTTRIDELAYEN = 0x0351, // Trigger output delay enable function (DEC: 849)

TULVIDC_OUTPUTTRIWIDTHEN = 0x0352, // Trigger output pulse width enable function (DEC: 850)

TULVIDC_OUTPUTTRIREADYEN = 0x0353, // Trigger Ready enable function (DEC: 851)

TULVIDC_OUTPUTTRIKINDEN = 0x0354, // Trigger mode enable function (DEC: 852)

TULVIDC_OUTPUTTRIPOINT = 0x0358, // Trigger output port range function (DEC: 856)

TULVIDC_OUTPUTTRIKIND = 0x0359, // Trigger output mode range function (DEC: 857)

TULVIDC_TRIINEXPOSURE = 0x0360, // Trigger exposure enable function (DEC: 864)

TULVIDC_TRIINEDGE = 0x0361, // Trigger edge enable function (DEC: 865)

TULVIDC_TRIINDELAY = 0x0362, // Trigger delay enable function (DEC: 866)

TULVIDC_TRIINSTART = 0x0363, // Trigger Start enable function (DEC: 867)

TULVIDC_TRIINFILTER = 0x0364, // Trigger filter enable function (DEC: 868)

5.1.4. TULV_IDPROP Attribute ID: Used for TU_SetParameter.vi

TULVIDP_EXPTM = 0x0100, // Exposure Time ID (DEC: 256)

TULVIDP_BRIGHTNESS = 0x0101, // Brightness Adjustment ID (DEC: 257)

TULVIDP_PIXELRATIO = 0x0102, // Pixel Ratio Adjustment ID (DEC: 258)

TULVIDP_IMGMETADATA = 0x0103, // Image Frame Header Data Switch ID (DEC: 259)

TULVIDP_RGAIN = 0x0105, // Red Channel (Color Camera) ID (DEC: 261)

TULVIDP_GGAIN = 0x010A, // Green Channel (Color Camera) ID (DEC: 266)

TULVIDP_BGAIN = 0x010F, // Blue Channel (Color Camera) ID (DEC: 271)

LabVIEW Development Guide

TULVIDP_SATURATION = 0x0114,	// Saturation (Color Camera) ID (DEC: 276)
TULVIDP_GAMMA = 0x0119,	// Gamma Value ID (DEC: 281)
TULVIDP_CONTRAST = 0x011E,	// Contrast ID (DEC: 286)
TULVIDP_LFTLEVEL = 0x0123,	// Left Color Level ID (DEC: 291)
TULVIDP_RGTLEVEL = 0x0128,	// Right Color Level ID (DEC: 296)
TULVIDP_GLOBGAIN = 0x012D,	// Global Gain ID (DEC: 301)
TULVIDP_BLACKLEVEL = 0x012E,	// Camera Offset ID (DEC: 302)
TULVIDP_GAINMODE = 0x012F,	// Camera Gain Mode Value ID (DEC: 303)
TULVIDP_SHARPNESS = 0x0132,	// Sharpness ID (DEC: 306)
TULVIDP_COLORTEMP = 0x0137,	// Color Temperature ID (DEC: 311)
TULVIDP_TRIKIND = 0x013C,	// Trigger Mode (0: Supports Sync and Global, 1: Does Not Support Sync and Global) ID (DEC: 316)
TULVIDP_FRAMERATE = 0x0141,	// Frame Rate Adjustable ID (DEC: 321)
TULVIDP_TEMPERATURE = 0x0146,	// Temperature Retrieval ID (DEC: 326)
TULVIDP_TEMPERATURETARGET = 0x0147,	// Camera Temperature Value ID (DEC: 327)

5.1.5.TULV_CAPTURE_MODES Capture Mode ID: Used in

TU_SetTrigger.vi

TULVCM_SEQUENCE	= 0x00,	// Sequence mode (stream mode)
TULVCM_TRIGGER_STANDARD	= 0x01,	// Standard trigger mode
TULVCM_TRIGGER_SYNCHRONOUS	= 0x02,	// Synchronous trigger mode
TULVCM_TRIGGER_GLOBAL	= 0x03,	// Global trigger mode
TULVCM_TRIGGER_SOFTWARE	= 0x04,	// Software trigger

5.1.6.TULVIMG_FORMATS Image Format ID: Used for saving images

TULVfmt_RAW	= 0x01,	// RAW format
-------------	---------	---------------

TULVFMT_TIF	= 0x02,	// TIFF format
TULVFMT_PNG	= 0x04,	// PNG format
TULVFMT_JPG	= 0x08,	// JPEG format
TULVFMT_BMP	= 0x10,	// BMP format

5.1.7. TULV_TRIGGER_EXP Trigger Exposure Mode ID: Used in

TU_SetTrigger.vi

TULVTE_EXPTM	= 0x00,	// Trigger using exposure time mode
TULVTE_WIDTH	= 0x01,	// Trigger using pulse width mode

5.1.8. TULV_TRIGGER_EDGE Trigger Edge Mode ID: Used in

TU_SetTrigger.vi

TULVTD_RISING	= 0x01,	// Rising edge trigger
TULVTD_FALLING	= 0x00,	// Falling edge trigger

6.VI Introduction and application

6.1 TU_InitCamera.vi



Input: error in

Output: error out、ReturnValue

Function: This is the first VI module used before operating any TUCSEN-controlled camera. Its purpose is to initialize all TUCSEN cameras that are already connected via USB and powered on. After calling this VI, an output value is returned. If the **ReturnValue** is 1, the camera initialization is successful, and other VIs can then be called. If the **ReturnValue** is not 1, it indicates that the camera initialization has failed. For instructions on how to use this VI, refer to the provided demo.

Note: If the initialization module needs to be called multiple times in a program, you must call TU_UninitCamera.vi before the second call to TU_InitCamera.vi to ensure success.

Output Description

ReturnValue: A value of 1 indicates a successful call, while a value other than 1 indicates failure. The specific cause of failure can be determined by referencing the TULVRET error codes.

6.2 TU_UninitCamera.vi



Input: error in、nCamCount



Output: error out、ReturnValue

Function: After calling this VI, the system will release all camera resources. If the camera is still in preview mode, it will be forcibly closed. Once the camera resources are released, the TU_InitCamera.vi must be called again to reinitialize the camera resources. Whether the release of resources is successful can be determined by the output value ReturnValue. If ReturnValue is 1, it indicates the camera resources were successfully released; if not, the release failed.

Input Description

nCamCount: Number of cameras to exit.

Output Description

ReturnValue: A value of 1 indicates the function call was successful; any other value indicates failure. The specific failure reason can be found using the TULVRET error codes.

6.3 TU_OpenCamera.vi



Input: error in、nCamCount

Output: error out、CameraCount、ReturnValue

Function: Enable camera operation. After successfully initializing the camera resources, TU_OpenCamera.vi will enable multiple cameras based on the resource allocation during the initialization process. The number of cameras to open is controlled by the input. The success of this step can be determined by the ReturnValue. A value of 1 indicates the cameras were successfully opened, while any other value indicates failure.

After successfully opening the cameras, the output CameraCount will reflect the total number of cameras that were successfully opened. Only after successful camera opening can subsequent operations, such as parameter setting and parameter range retrieval, be performed. Refer to the example for details on how to use this VI.

Input Description

nCamCount: Number of cameras to open.

Output Description

nCamCount: Number of cameras successfully opened.

ReturnValue: A value of 1 indicates the function call was successful; any other value indicates failure. The specific failure reason can be found using the TULVRET error codes.

6.4 TU_SetParameter.vi



Input: CameraIndex、Parameter、Value in、strInfo、error in

Output: ReturnValue、Error out

Function: Set camera parameters. By executing TU_SetParameter.vi, a series of camera parameters can be set, including exposure time, gamma value, gain, contrast, etc. (refer to TULV_IDCAPA performance ID and TULV_IDPROP property ID for specific configurable parameters). The success of parameter settings can be determined by the ReturnValue. A ReturnValue of 1 indicates the parameters were successfully set, while any other value indicates failure. Specific failure reasons can be found using the ReturnValue corresponding TULVRET error codes. Refer to the example SetParameterDemo.vi for usage details.

Input Description

CameraIndex: The sequence number of the camera to operate. The sequence number starts from 0, where 0 represents the first camera, 1 represents the second, and so on. Refer to the LabviewDemo example for details on how to map camera sequence numbers to camera names.

Parameter: The ID corresponding to the type of parameter to be set. For all currently supported parameters, refer to TULV_IDCAPA and TULV_IDPROP. Additional parameters will be added as needed in future versions.

Value in: The value of the parameter to be set. Refer to TU_GetParameterLimits.vi for the parameter range of the configurable parameter types. Refer to the appendix for the meaning of performance parameter values.

strInfo: String-type parameter to be set, such as the image storage path.

Output Description

ReturnValue: A value of 1 indicates the function call was successful; any other value indicates failure. The specific failure reason can be found using the TULVRET error codes.

6.5 TU_GetParameter.vi



Input: CameraIndex、ParameterID、error in

Output: ParameterValue、ReturnValue、strInfo、error out

Function: Retrieve parameter values. By executing TU_GetParameter.vi, a series of current parameter values for the camera can be obtained, including exposure time, gamma value, gain value, contrast, etc. (For specific settable parameters, refer to

TULV_IDCAPA performance ID and TULV_IDPROP attribute ID). The success of parameter retrieval can be determined by the output ReturnValue. If the ReturnValue is 1, it indicates the parameter retrieval was successful; if the value is not 1, it indicates a failure. The specific reason for failure can be found by referencing the ReturnValue against the TULVRET error codes. Refer to the example Demo for usage instructions.

Input Description

CameraIndex: The serial number of the camera for which the parameter values are to be retrieved. The serial number starts at 0, where 0 represents the first camera, 1 represents the second camera, and so on. To map camera serial numbers to camera names, refer to the example LabviewDemo.

ParameterID: The ID of the parameter type to be retrieved.

Output Description

ParameterValue: The retrieved parameter value (numeric type) corresponding to the provided ParameterID.

ReturnValue: If the value is 1, the call was successful; if not, the call failed. Specific failure reasons can be found using the corresponding TULVRET error codes.

strInfo: The retrieved parameter value (string type) corresponding to the provided ParameterID.

6.6 TU_GetParameterLimits.vi



Input: CameraIndex、ParameterID、error in

Output: MaxValue、MinValue、StepValue、DefaultValue、ReturnValue、error out

Function: Retrieve parameter value ranges. This includes the maximum value, minimum value, default value, and step value. The success of retrieving the parameter range can be determined by the ReturnValue. If ReturnValue is 1, the parameter range retrieval was successful; if not, it indicates a failure. The specific reason for failure can be found by referencing ReturnValue against TULVRET error codes. Refer to the example Demo for usage instructions.

Input Description

CameraIndex: The serial number of the camera corresponding to the parameter value range to be retrieved. The serial number starts at 0, where 0 represents the first camera, 1 represents the second camera, and so on. To map camera serial numbers to camera names, refer to the example LabviewDemo.

ParameterID: The ID of the parameter type whose value range is to be retrieved.

Output Description

MaxValue: The maximum value corresponding to the input parameter type.

MinValue: The minimum value corresponding to the input parameter type.

StepValue: The step value corresponding to the input parameter type.

DefaultValue: The default value corresponding to the input parameter type.

ReturnValue: If the value is 1, the call was successful; if not, the call failed. Specific failure reasons can be found using the corresponding TULVRET error codes.

6.7 TU_GetCameraInfo.vi



Input: CameraIndex、TextIndex、TextID、error in

Output: MaxValue、MinValue、StepValue、DefaultValue、ReturnValue、error out

Function: Retrieve camera-related information. This includes camera name, serial number (if supported), firmware version, SDK version, USB interface type (2.0 or 3.0), and resolution name. The success of retrieving the information can be determined by the ReturnValue. Refer to the example Demo for usage instructions.

Input Description

CameraIndex: The serial number of the camera whose information is to be retrieved. The serial number starts at 0, where 0 represents the first camera, 1 represents the second camera, and so on. To map camera serial numbers to camera names, refer to the example Demo.

TextID: The ID corresponding to the camera information to be retrieved (refer to TULV_IDINFO).

TextIndex: This input is used when an information ID contains multiple camera-related details. For example, for the camera named ISH130, which has two resolutions, when TextID is TULVIDI_RESOLUTION, to display all resolutions, TextIndex should be set to 0 and 1 respectively. For more details, refer to the example Demo.

Output Description

CameraText: The camera-related information output based on the TextID and TextIndex.

ReturnValue: If the value is 1, the call was successful; if not, the call failed. Specific failure reasons can be found using the corresponding TULVRET error codes.

6.8 TU_SetTrigger.vi



Input: CameraIndex、CaptureMode、ExpourseMode、EdgeMode、DelayTime、Frame、nBufFrame

Output: ReturnValue

Function: Camera External Trigger Settings(This feature is only available for cameras that support external triggering. For cameras that do not support external triggering, invoking this module will have no effect.) For an introduction to external triggering, please refer to the documentation (3.4.1 Capture Mode) and (3.4.3 Trigger Mode). For how to use this VI, refer to the example TriggerDemo.vi.

Input Description

CameraIndex: The camera serial number corresponding to the external trigger. The sequence starts from 0, where 0 represents the first camera, 1 represents the second camera, and so on. For how to match the camera serial number to the camera name, refer to the example TriggerDemo.vi.

CaptureMode: This input includes five modes: TULVCM_SEQUENCE, TULVCM_TRIGGER_STANDERD, TULVCM_TRIGGER_SYNCHRONOUS, TULVCM_TRIGGER_GLOBAL, and TULVCM_TRIGGER_SOFTWARE. For parameter descriptions, refer to the 3.4.3 Trigger Mode section.

ExposureMode: This input includes two modes: TULVTE_EXPTM and TULVTE_WIDTH. For parameter descriptions, refer to the 3.4.3 Trigger Mode section.

EdgeMode: This input includes two modes: TULVTD_RISING and TULVTD_FAILING. For parameter descriptions, refer to the 3.4.3 Trigger Mode section.

DelayTime: When ExposureMode is set to TULVTE_EXPTM, the value of DelayTime determines how long it takes after receiving the trigger signal to output the image.

Frame: The total number of images output for one trigger (this parameter may not need to be set when using a trigger board).

nBufFrame: The number of buffered images in the SDK.

Output Description

ReturnValue: A value of 1 indicates a successful call, while any value other than 1

indicates a failure. Refer to the TULVRET error code for the specific failure reason.

6.9 TU_GetTrigger.vi



Input: CameraIndex

Output: ReturnValue

Function: Get external trigger parameters and check if external trigger is supported.

This VI checks if the external trigger mode is supported based on the input camera serial number. For how to use this VI, refer to the example LabviewDemo.

Note: TU_GetTrigger.vi can be customized according to customer needs to obtain more output information, including the current trigger mode, exposure mode, delay duration, and level trigger mode.

Input Description

CameraIndex: The camera serial number corresponding to the external trigger value. The sequence starts from 0, where 0 represents the first camera, 1 represents the second camera, and so on. For how to match the camera sequence number to the camera name, refer to the example SetTriggerDemo.vi.

Output Description

ReturnValue: A value of 1 indicates that the camera supports external triggering and the call was successful. Any value other than 1 indicates a failure or unsupported external trigger. Refer to the TULVRET error code for the specific failure reason.

6.10 TU_SetROI.vi



Input: CameraIndex、 StartX、 StartY、 Width、 Height、 SetROI

Output: ReturnValue

Function: Set ROI Mode. This VI allows you to select a region for data capture within the allowed range. Refer to the example ROI Demo.vi for how to use this VI.

Input Description

CameraIndex: The camera serial number corresponding to the ROI mode. The sequence starts from 0, where 0 represents the first camera, 1 represents the second camera, and so on. For how to match the camera sequence number to the camera name, refer to the example ROI Demo.vi.

StartX: Horizontal offset.

StartY: Vertical offset.

Width: Horizontal width.

Height: Vertical height.

SetROI: Whether to set ROI mode. 0 to disable, 1 to enable.

Output Description

ReturnValue: A value of 1 indicates the ROI mode was successfully set, while any value other than 1 indicates a failure. Refer to the TULVRET error code for the specific failure reason.

6.11 TU_StartCapture.vi



Input: CameraIndex、 CaptureMode

Output: CameraIndex、 ImageType、 Width、 Height、 Channel、 Depth

Function: After successfully calling TU_InitCamera.vi and TU_OpenCamera.vi, you can call TU_StartCapture.vi to allocate memory and start data capture. Refer to the example Demo for how to use this VI.

Input Description

CameraIndex: The camera serial number for data capture.

CaptureMode: Data capture mode (streaming mode or trigger mode). Refer to the TULV_CAPTURE_MODE input type for more details.

Output Description

CameraIndex: The serial number of the camera where data capture was successful.

ImageType: The type of output image. Used for the IMAQ Create module. Refer to the Demo example for usage.

Width: The width of the output image.

Height: The height of the output image.

Channel: The number of channels in the output image (color or grayscale).

Depth: The bit depth of the output image data (8-bit or 16-bit).

6.12 TU_StopCapture.vi



Input: CameraIndex

Output: Return Value

Function: Stops data capture. The processing of stopping data capture is based on the input camera serial number. Data capture can be resumed by calling TU_StartCapture.vi under the condition that TU_UninitCame has been executed. For how to use this VI, refer to the example demo.

Input Description

CameraIndex: The serial number of the camera for which data capture needs to be stopped.

Output Description

Return Value: A value of 1 indicates that data capture was successfully stopped, and a value other than 1 indicates failure to stop data capture. The specific reason for failure can be found by checking the TULVRET error code.

6.13 TU_WaitForFrame.vi



Input: CameraIndex

Output: nFrameIndex、 dbfTimeStamp、 dbfTimeLast、 Return Value

Function: Waits for data capture to complete. For detailed usage, refer to the demo.

Input Description

CameraIndex: The serial number of the camera that needs to be processed.

Output Description

ReturnValue: A value of 1 indicates that data capture is complete, while a value other than 1 indicates failure in data capture. The specific reason for failure can be found by checking the TULVRET error output.

nFrameIndex: Timestamp ID

dblTimeStamp: Timestamp

dblTimeLast: Duration of the timestamp

6.14 TU_GetFrame.vi



Input: CameraIndex、ImageSRC_Average、ImageSRC_BKGD、ImageSRC_Normal、AverageCntIn、bSetBKGD、nFuncSelect

Output: calBKGD、calAverage、Image_BKGD、Image_Average、Image_Normal

Function: Retrieves frame data for real-time display of the preview image. For detailed usage, refer to the example demo.

Input Description

CameraIndex: The serial number of the camera from which frame data is to be retrieved.

ImageSRC_Average: Input source image for averaging.

ImageSRC_BKGD: Input source image for background subtraction.

ImageSRC_Normal: Input source image for normal output.

AverageCntIn: Number of frames to average (0-99).

BSetBKGD: Whether to perform background subtraction (0: No, 1: Yes).

nFuncSelect: Function selection (0: Normal output, 1: Background subtraction, 2: Averaging).

Output Description

calBKGD: Whether background subtraction is completed.

calAverage: Whether averaging is completed.

Image_BKGD: Background subtraction output.

Image_Average: Averaging output.

Image_Normal: Normal output.

Note: Only one output method can be selected per GetFrame call.

6.15 TU_ErrorReport.vi



Function: Under development...

6.16 TU_OpenImageFile.vi

Input: NULL

Output: bCanceled、Image

Function: Open Image (for background subtraction).

Output Description:

Image: The opened image (can be directly displayed).

bCanceled: Whether the image was opened (for background subtraction calculation)
(0: canceled, 1: image opened).

6.17 TU_StartRecorder.vi

Input: nCamIndex

Output: ReturnValue

Function Description: Start recording video file.

Input Description:

nCamIndex: The camera serial number from which frame data needs to be captured.

Output Description:

ReturnValue: Whether the video recording started successfully (0: failed, 1: successful).

6.18 TU_StopRecorder.vi

Input: nCamIndex

Output: ReturnValue

Function Description: Stop recording video file.

Input Description

nCamIndex: The camera serial number from which frame data needs to be captured.

Output Description

ReturnValue: Whether the video recording stopped successfully (0: failed, 1: successful).

6.19 TU_SetOutPutTrigger.vi



Input: CameraIndex、 OutPort、 OutMode、 EdgeMode、 DelayTime、 OutWidth

Output: ReturnValue

Function: Camera external trigger output setting (This function is only available for cameras that support external trigger output; for unsupported cameras, calling this module will have no effect). For usage, refer to the TriggerDemo.vi example.

Input Description

CameraIndex: The camera serial number corresponding to the external trigger setting. The serial number starts from 0 (0 represents the first camera, 1 represents the second, and so on). For how to match the camera serial number with the camera name, refer to the SetTriggerDemo.vi example.

OutPort: The output port mode, with Port1, Port2, Port3 available for selection.

OutMode: The trigger output mode, which varies depending on the camera. For details, refer to TriggerDemo.vi and LabviewSample.vi module examples.

EdgeMode: This input includes two modes: TULVTD_RISING and TULVTD_FAILING.

DelayTime: When TULV_TEPTM is chosen, the DelayTime value determines the delay before triggering the output signal.

OutWidth: This value determines the pulse width of the trigger output signal.

Output Description

ReturnValue: A value of 1 indicates success, and a value not equal to 1 indicates failure. For specific failure reasons, refer to the TULVRET error codes.

6.20 TU_GetOutPutTrigger.vi



Input: CameraIndex

Output: ReturnValue

Function: Get external trigger output parameters and check if external trigger output is supported. This VI checks if the external trigger output mode is supported based on the input camera serial number. For usage, refer to the TriggerDemo.vi example.

Note: TU_GetOutPutTrigger.vi can be modified as per customer requirements to get more output information, including current output port, output mode, delay time, level trigger mode, and pulse width duration.

Input Description:

CameraIndex: The camera serial number corresponding to the external trigger value. The serial number starts from 0 (0 represents the first camera, 1 represents the second, and so on). For how to match the camera serial number with the camera name, refer to the SetTriggerDemo.vi example.

Output Description:

ReturnValue: A value of 1 indicates that the camera supports external trigger output mode and the call was successful; otherwise, it indicates failure or that external trigger output is not supported. For specific failure reasons, refer to the TULVRET error codes.

6.21 TU_StartMultiData.vi



Input: CameraIndex、StackNum、Format

Output: ReturnValue

Function: Set stack capture format (TIFF, RAW), stack count, and enable stack capture.

Input Description:

CameraIndex: The camera serial number corresponding to the external trigger value. The serial number starts from 0 (0 represents the first camera, 1 represents the second, and so on).

StackNum: Represents the number of stacks to be set.

Format: Represents the capture format (currently only TIFF and RAW are supported). Setting 1 means RAW, setting 2 means TIFF.

Output Description

ReturnValue: A value of 1 indicates that the camera supports external trigger output mode and the call was successful; otherwise, it indicates failure or that external trigger output is not supported. For specific failure reasons, refer to the TULVRET error codes.

6.22 TU_StopMultiData.vi



Input: CameraIndex

Output: ReturnValue

Function: Stop stack capture.

Input Description

CameraIndex: The camera serial number corresponding to the external trigger value. The serial number starts from 0 (0 represents the first camera, 1 represents the second, and so on).

Output Description

ReturnValue: A value of 1 indicates that the camera supports external trigger output mode and the call was successful; otherwise, it indicates failure or that external trigger output is not supported. For specific failure reasons, refer to the TULVRET error codes.

Appendix

Performance ID	Setable Parameter Values	Parameter Description
TULVIDC_AUTOWB	0, 1, 2	0: Disable auto white balance 1: One-time white balance 2: Auto white balance
TULCIDC_AUTOEXP	0, 1, 2	0: Disable auto exposure 1: Enable auto exposure 2: One-time exposure
TULVIDC_DEPTH	0, 1	0: 8-bit data width 1: 16-bit data width
TULVIDC_AUTOLEVEL	0, 1, 2, 3	0: Disable auto color level 1: Auto left color level 2: Auto right color level 3: Auto left and right color levels
TULVIDC_RESOLUTION	0-n	The number of resolutions depends on the camera. 0 is the first resolution, 1 is the second resolution, and so on.
TULVIDC_NOISELEVEL	0, 1, 2, 3	0: Off 1: Low 2: Medium 3: High
TULVIDC_IMGMODESELECT	0, 1	0: Disable CMS 1: Enable CMS
TULVIDC_SAVEFRAME	n	Used to save images when capturing; the number of images saved depends on the actual setting value.
TULVIDC_IMGSAVEPATH	String	Used to select the storage path for images.
TULVIDC_IMGFORMAT	1, 2, 4, 8, 16	1: RAW 2: TIF 4: PNG 8: JPG 16: BMP
TULVIDC_IMGCOUNT	0-n	Number of images to store.
TULVIDC_CAMERASTATE		This performance ID does not require specific parameter values. After calling TU_StartCapture.vi or TU_StopCapture.vi, you can obtain the current camera status by calling TU_GetParameter.vi. 0: Preview state 1: Suspended state

Attribute ID	Setable Parameter Values
TULVIDP_EXPTM	MinValue-MaxValue
TULVIDP_RGAIN	
TULVIDP_GGAIN	
TULVIDP_BGAIN	
TULVIDP_SATURATION	
TULVIDP_GAMMA	
TULVIDP_CONTRAST	
TULVIDP_LFTLEVEL	
TULVIDP_RGTLEVEL	
TULVIDP_GLOBGAIN	
TULVIDP_SHARPNESS	
TULVIDP_COLORTEMP	

Note: The above parameters can only be set if the camera supports the respective functionality.

The MinValue and MaxValue values can be obtained by calling TU_GetParameterLimits.vi.