SVCam Kit
Quick Guide

Version 2.5.4

SVS-Vistek GmbH
Muehlbachstr. 20
82229 Seefeld
Germany

SVCam Kit Installation Guide

Current SVCam Kit software version used in this document: 2.5.4 / 2020-09-02

Purpose of this document

This document explains how to install the SVCam-Kit driver and SVCapture. It also describes the SVCam Kit components and their use. In particular, you will find a short introduction to the usage of the GenICam tree and the assistants provided by SVCapture.

For development files, please read the *'Getting started with SDK'* guide, available in the SVS-Vistek download area.

Target audience

Users of industrial cameras from SVS-Vistek with basic knowledge about machine vision.

CONTENT

1 Introduction 4

1.1 Requirements 4

1.2 Where to get the software 4

2 Install SVCam Kit 5

2.1 Software installation 7

2.2 Driver installation 9

2.2.1 GigE Filter driver 9
2.2.2 GigE Jumbo Frames 10
2.2.3 USB3 driver 10
2.2.4 CoaXPress support 12

3 Operate SVCapture 14

3.1 Supported interfaces 14

3.2 Connect your camera 16

3.3 The GenICam tree 17

3.4 The SVS-Vistek GenICam tree 18

3.5 Assistants 24

3.5.1 LUT creation 24
3.5.2 I/O Config 25
3.5.3 Line status visualizer 26
3.5.4 Strobe controller 27
3.5.5 Sequencer 28
3.5.6 Event viewer 31
3.5.7 Pixel Histogram 31
3.5.8 Shading assistant 32
3.5.9 Defect Pixel assistant 34

3.6 Common procedures 37

3.6.1 Check strobe light 37
3.6.2 Setup single strobe light 38
3.6.3 Set input to physical trigger input 38
3.6.4 Debounce input trigger 39

4 Appendix 40

4.1 Connect strobe to camera 40

1 Introduction

SVS-Vistek produces high quality industrial cameras with various interfaces such as USB3, GigE Vision, 10GigE, Camera Link and CoaXPress. The SVCam Kit provides the drivers and development files to work with USB3, 10 GigE, GigE Vision and Camera link cameras.

CoaXpress support is currently in beta, but already very stable with some framegrabbers (currently mainly Euresys framegrabber). More framegrabbers will be supported in the future.

If you purchased a CoaXPress camera, in any case, refer as well to the manual of your CoaXPress grabber manufacturer.

1.1 Requirements

Windows 7 is the minimum Windows version required. You need Windows system administrator privileges on your computer to install the SVCam Kit.

1.2 Where to get the software

You can download the current SVCam Kit from the SVS-Vistek download center. Select your camera in the download center and download the matching SVCam Kit. Make sure the selected SVCam Kit is 32bit (x86) or 64bit, matching the type of your Windows operating system. The software comes compressed in a zip file.

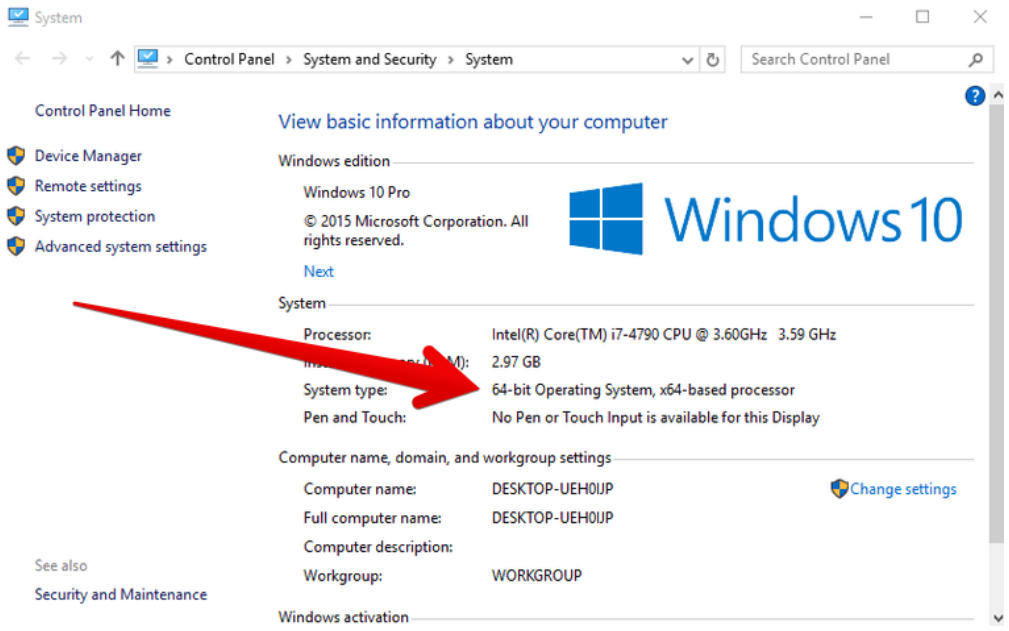
2 Install SVCam Kit

In the installation process, the device driver will be installed to your system. It is necessary for you to have admininstrative permissions.

You should run the 32-bit or 64-bit version of the setup routine (SVCam_Kit_v2.5.4._x64.exe or SVCam_Kit_v2.5.4._x86.exe) matching your windows version. This is how to determine your system type:

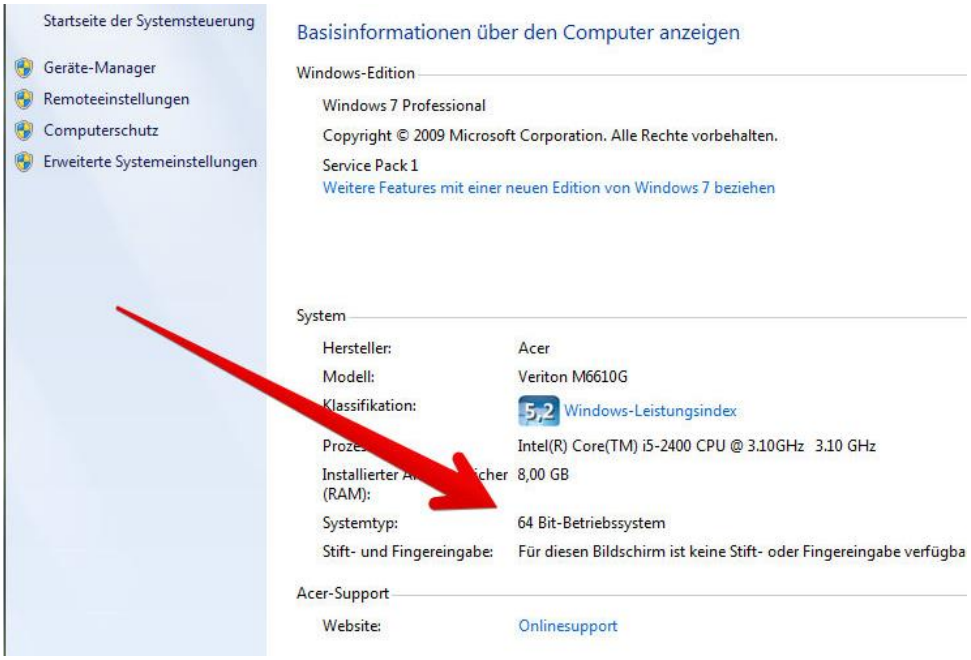
Windows 8 and Windows 10

Go to Start > Control Panel > System and Security



Windows 7

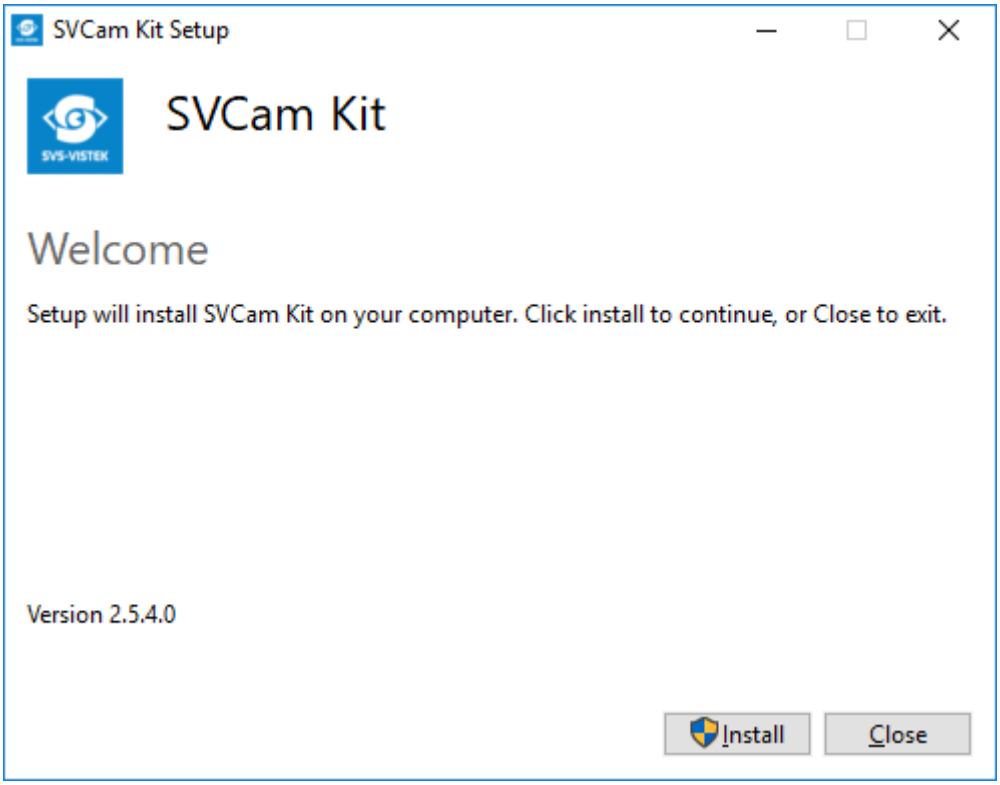
Go to Start > Control Panel > System



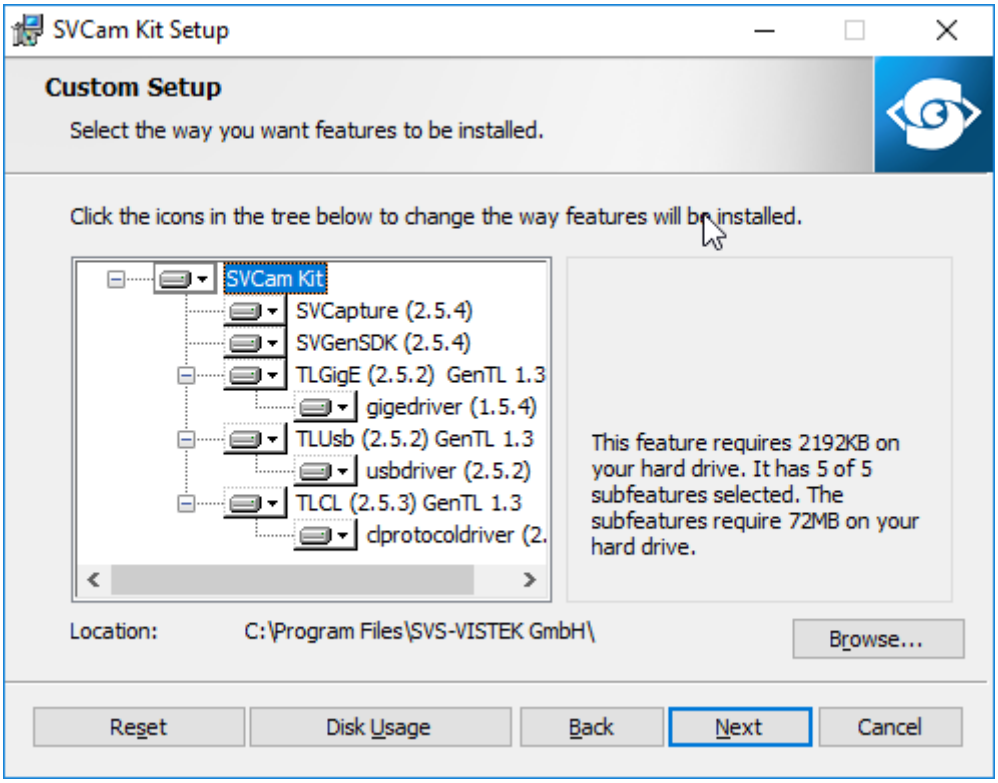
2.1 Software installation

It is strongly recommended to uninstall any existing version of SVCam Kit or SVCapture before installing the new version. While installing, please deactivate your firewall and antivirus programs.

Unzip (double-click) the zip file and run the installation procedure. Please note, the driver installation process will deactivate and re-activate all current network connections.



The installation process will then pop up with selectable options for installation.



Make sure your SVCam Kit installation includes all required components (this depends on your type of camera). The Transport Layer (TL) drivers are interfacing the hardware (GigE, Camera Link, USB3, CoaXPress) and are required depending on the type of your camera interface. You need a minimum of one TL driver. If you do not want to install the complete package, please keep in mind:

- Only the TL drivers you are not going to use can be omitted.
- If you want to actively develop with C# or C++, the SVGenSDK has to be installed as well.
- If you want to use the SVS-Vistek camera operated with external software (has to be able to work with transport layer drivers), you should install the required transport layer and drivers only.
- It is recommended to include SVCapture to be able to see / modify the GenICam camera tree.
- It is always the safest option to do a complete installation.

2.2 Driver installation

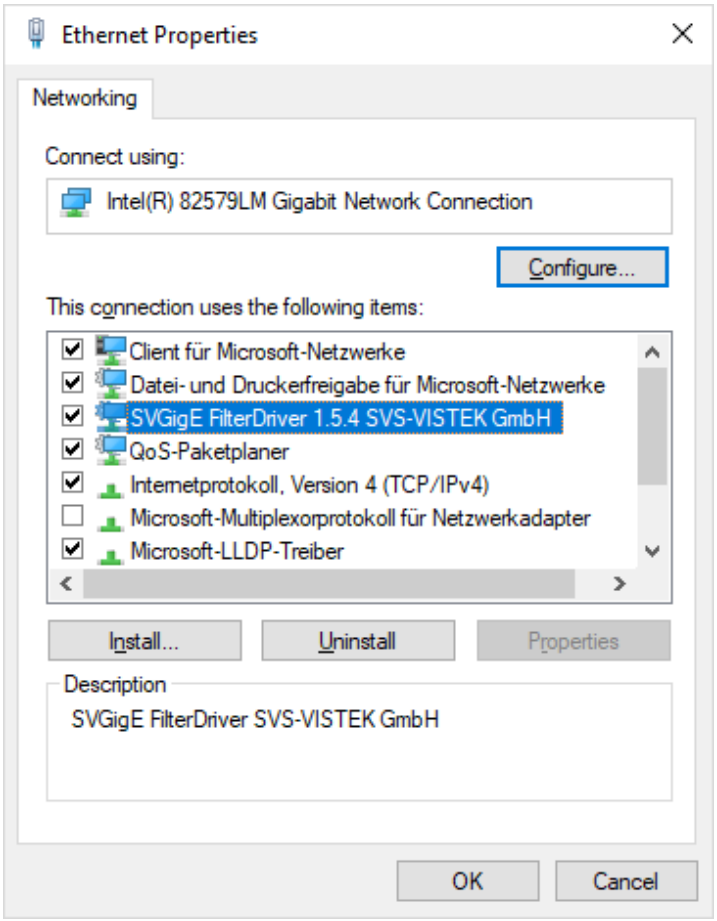
During the installation process, the required files will be written to your hard driveand the requested modifications will be automatically executed. Please verify as shown below.

2.2.1 GigE Filter driver

Once the TLGige installation has been completed, a packet filter driver will be installed as well. The filter driver will be attached to your networking device. This filter driver installation is necessary to drive the GigE interface. Without an activated filter driver, there will be no image acquisition from the camera.

Please check this in your windows system control after installation:

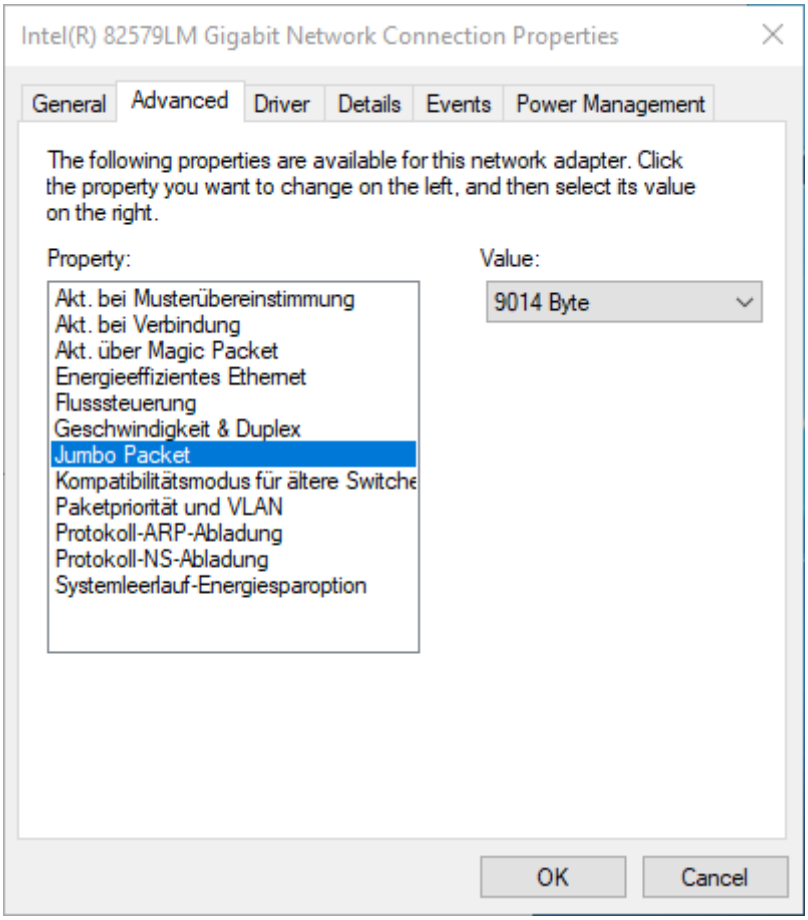
System control > network & internet > ethernet > change adapter options > Ethernet > Properties



2.2.2 GigE Jumbo Frames

Ethernet image data should be sent with activated jumbo frames to 9k (see properties of your ethernet card). This will reduce the amount of required ethernet packet header space, thus helping to make better use of ethernet bandwidth.

System control > network & internet > ethernet > change adapter options > Ethernet > Properties > Configure > Advanced > Jumbo Packet

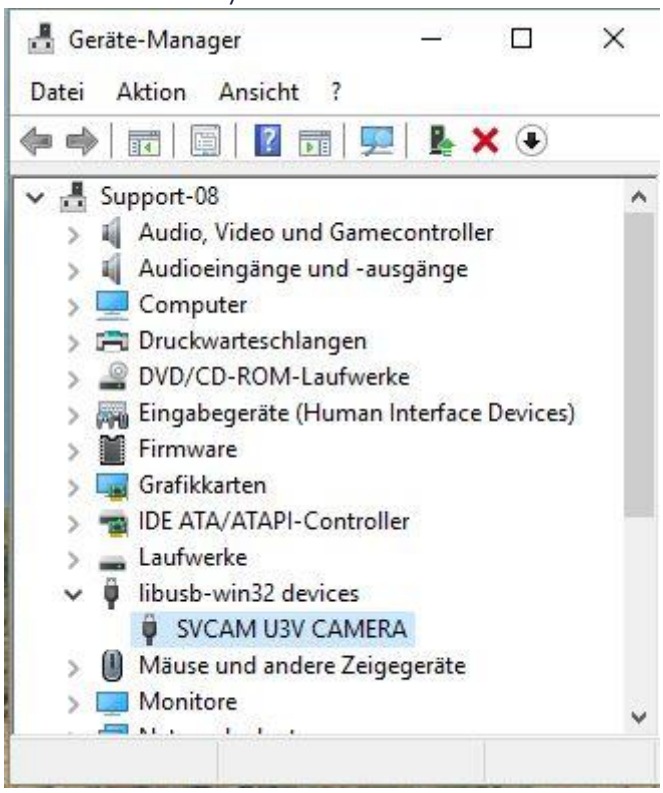


2.2.3 USB3 driver

With TLUsb and usbdriver installed you can operate USB3 cameras. The driver works according to the USB3Vision specification. This is how to verify whether the driver has been correctly installed and is connected properly to your camera:

1. Attach a SVS-Vistek USB3 camera (without I/O cables) to your computer. Wait until the power indicator is ON.

2. Go to windows device manager > libusb-win32 devices
   You should see your SVCam camera connected properly to the libusb driver

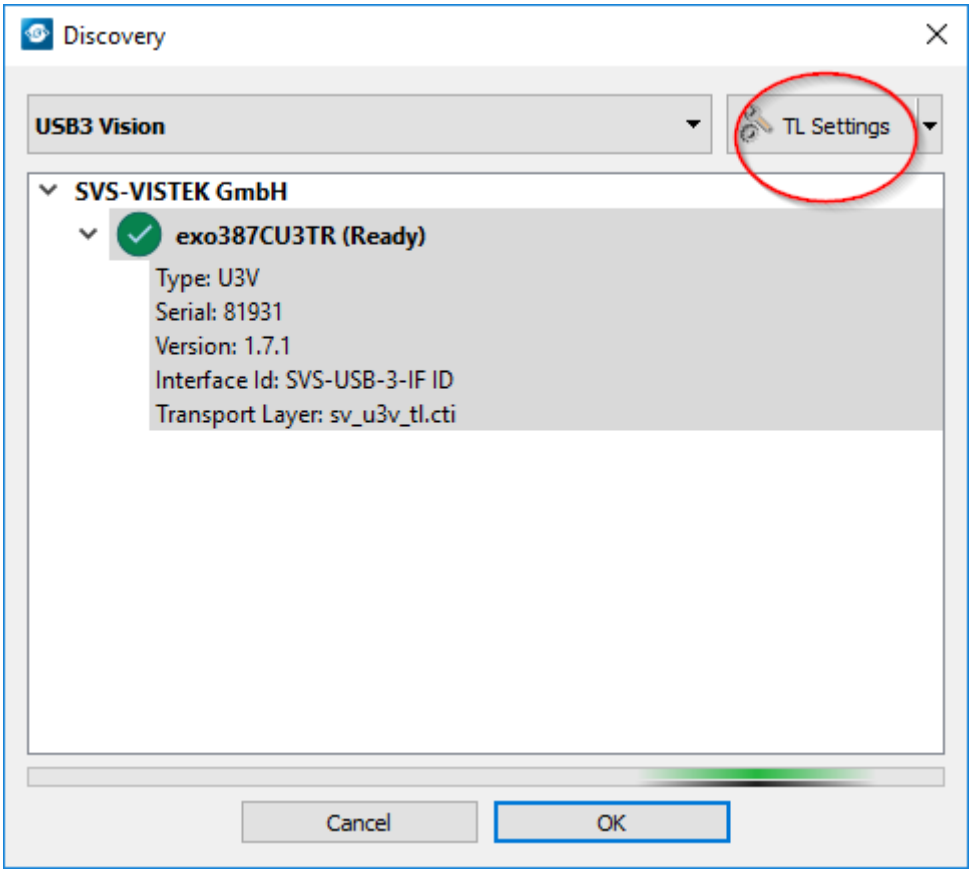


2.2.4 CoaXPress support

Currently, Euresys CoaXPress cards with latest driver should be supported. Install the Euresys software package first, then do the SVCam Kit install. In most environments SVCapture should find the Euresys driver out oft he box. In the camera discovery box you will find the CoaXPress interface as well.

Add GenTL driver manually

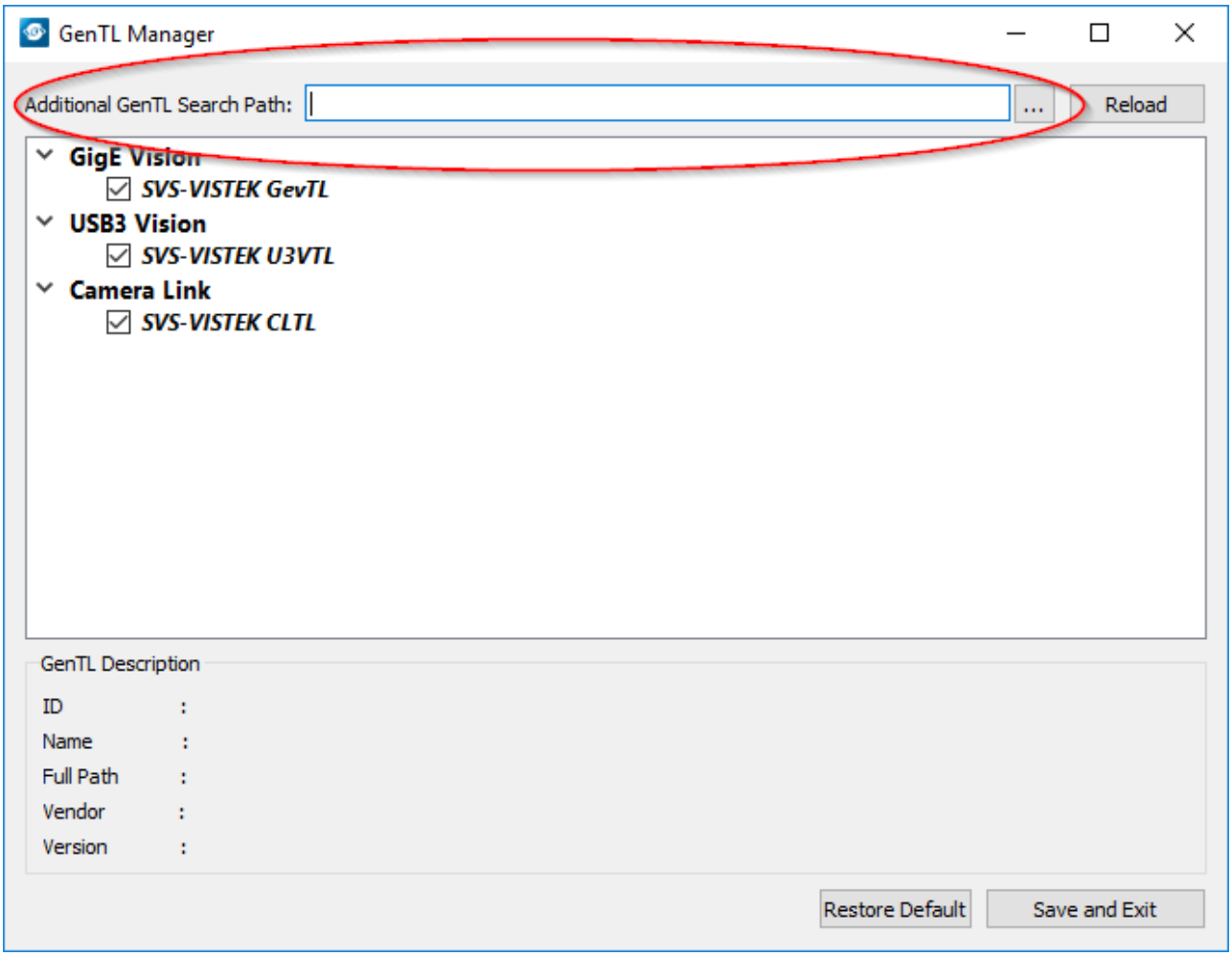
With current status, you might install the Euresys driver manually as well. In the camera discovery dialog, click on TL Settings



Using SVCapture with a grabber card (Camera Link or CoaXPress) requires support from the framegrabber software. Make sure, the software for your framegrabber is installed before installing SVCapture.

In the install process, your system will be searched for a *.cti file. If this transport layer file is found, it will be included automatically in SVCapture setup. In case it is not found, install the transport layer for your framegrabber maually.

The following dialog will enable you add the specific GenTL driver (most propable a *.cti file) location on your system to SVCapture. This dialog may also be used to activate/deactivate the interfaces available in the Discovery dialog.



3 Operate SVCapture

SVCapture is the viewer part of the SVCam Kit. It allows to manipulate the GenICam tree and displays the current image produced by the camera. If the GenICam settings are changed, the effect is immediately visible in the image. This makes SVCapture a great tool to adjust and test GenICam settings for your application. These GenICam settings can then be exported as a text file.

SVcapture has 2 main tasks:

> Display of the current image with the current GenICam tree as the basis
> Change the GenICam settings in the camera
> Simplify complex settings in the GenICam tree by using wizards

SVCapture is part of SVCam Kit and is recommended to be installed. SVCapture is the viewer software, permitting also the manipulation oft he camera GenICam tree. The image being displayed is

3.1 Supported interfaces

SVCam Kit is based on the GenTL protocol. The SVCam Kit GenICam API provides unified methods for camera access, regardless of the interface type. SVCam Kit includes GenTL producers for GigE Vision, USB3 and Camera Link. In addition, third-party GenTL producers can be integrated to support other interface types, such as CoaXPress.

The *.CTI files are the files providing the GenTL transport layer, necessary for other programs if they want to operate camera GenICam interface via GenTL.

This is the std location for CTI files:

C:\Program Files\SVS-VISTEK GmbH\SVCam Kit\TLGigE\bin\\sv_gev_tl_x64.cti

C:\Program Files\SVS-VISTEK GmbH\SVCam Kit\TLUsb\bin\\sv_u3v_tl_x64.cti

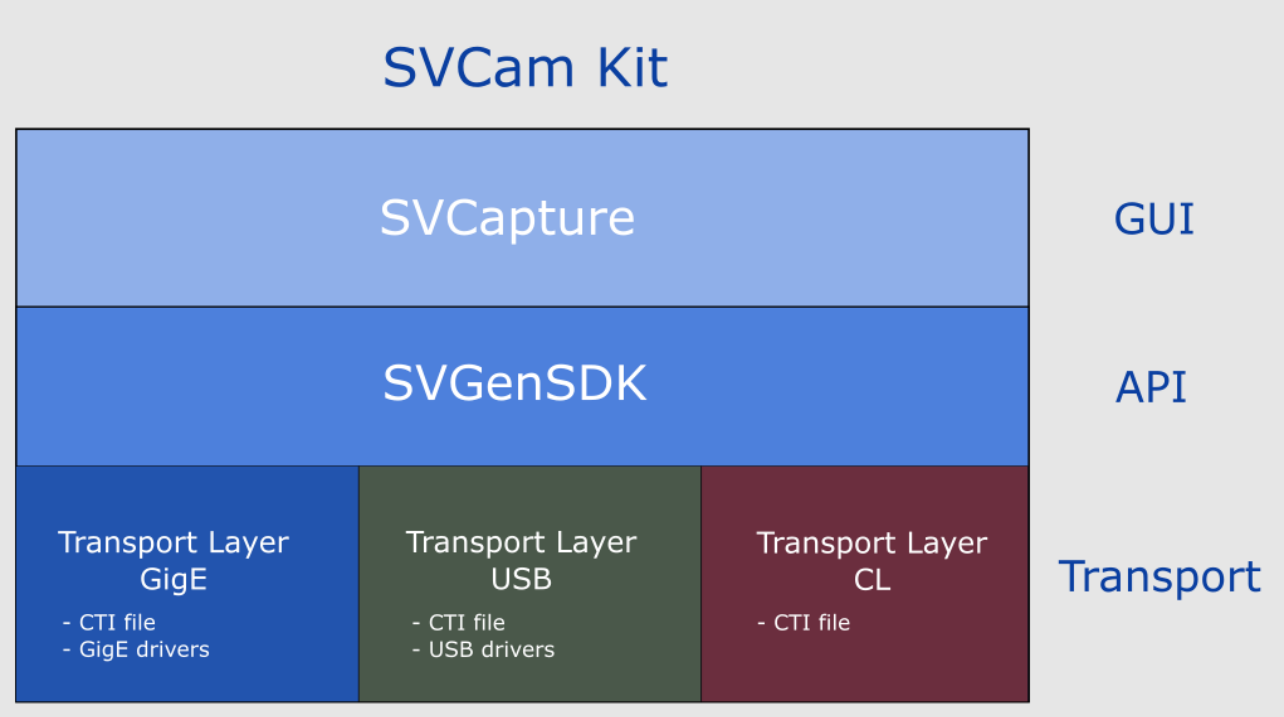
C:\Program Files\SVS-VISTEK GmbH\SVCam Kit\TLCL\bin\\sv_cl_tl_x64.cti

You might use these files in third-party software as well to have best driver support in your machine vision application.

NOTE
Following cameras require minimum firmware versions as below to run:

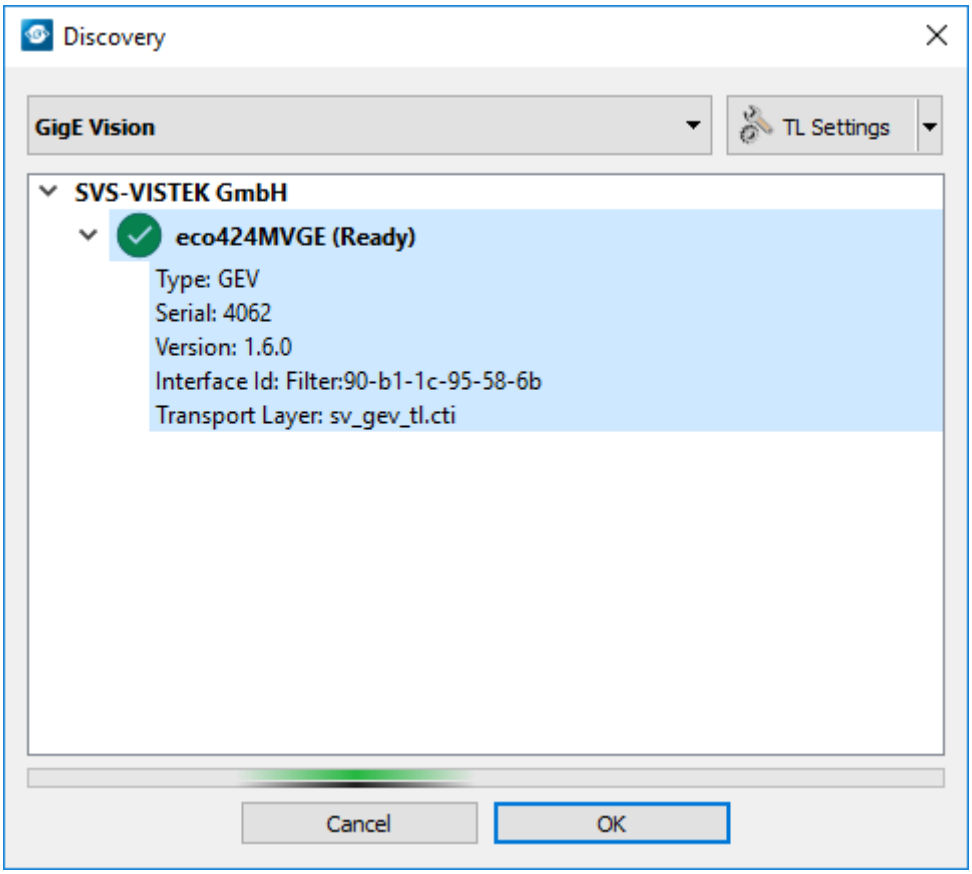
camera	firmware version
hr120xCL	b2927
EXO CL	b2945
shr47 CL	b2933
hr25000 CL, hr25 CL	b2931

The illustration below shows the modularized building structure of SVCapture. SVCapture is built on top the SVGen SDK. It is a GUI to operate the camera. The basis for interaction with the camera is the GenTL transport layer.

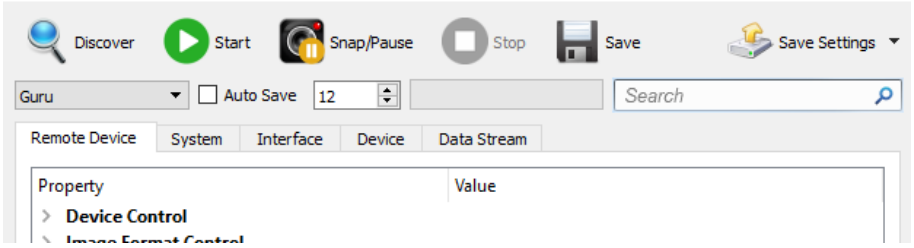


3.2 Connect your camera

To operate the camera in SVCapture, it is necessary to connect the camera to SVCapture. With SVCapture, select the interface type of your camera, make sure it is connected to your computer and powered on. You should see it like below. Connect with OK.



As soon the camera is connected successfully, the camera window is opening showing the GenICam tree of the connected camera. The Guru view will show all items, other views are hiding some less used items. Start will make the camera run, delivering images. You might save images in different formats as well as settings (use it for saving current configuration as a text file for your further use in programming). The search box is helping to find GenICam properties.



3.3 The GenICam tree

GenICam is a device operation interface. Item names can be set to values. The names and operations are standardized in the SNFC. In theory, any GenICam software would permit to operate all GenICam cameras independently of the brand. Most manufacturers do not have open interfaces in their own driver software, but big independant software packages like MIL, open eVision, HALCON, CVB, VisionPro or NI vision and many more are able to run GenICam devices.

GenICam consists of three modules:

- GenApi: By using an XML description file, access and control cameras
- SFNC: Recommended names and types for common features in cameras
- GenTL: Transport layer interface for enumerating cameras, grabbing images and data transfer

GenICam provides supports for basic functions:

- Configure the camera
- Grab images
- Transmit extra data
- Deliver events

3.4 The SVS-Vistek GenICam tree

Despite SFNC (Standard Feature Naming Convention), the GenICam tree still is specific to each manufacturer. This guide covers the SVS-Vistek GenICam tree, which is very close to official standards. Other manufacturers might implement names and features differently.

GenICam is organized in a tree structure. SVCapture displays the whole tree, grouped into functional blocks. Readonly items are displayed grey.

Device control

General information about the connected camera

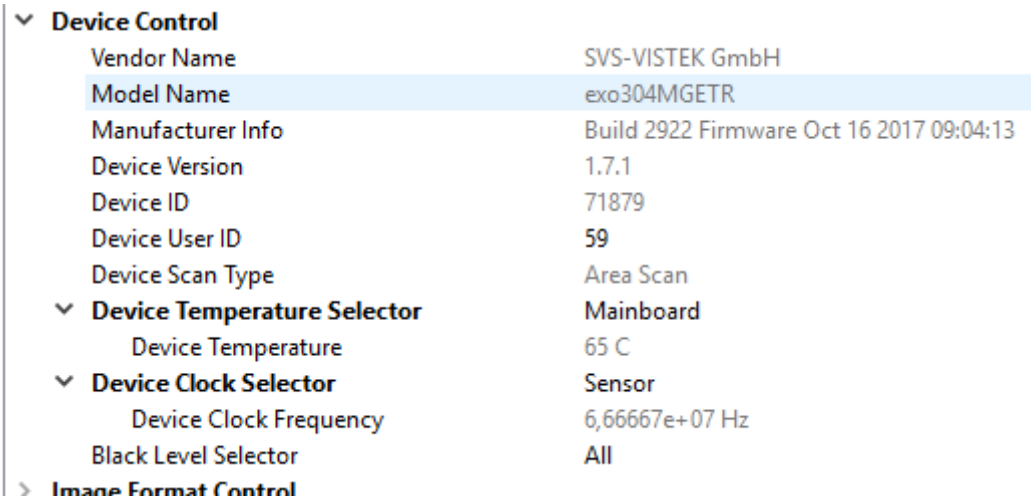
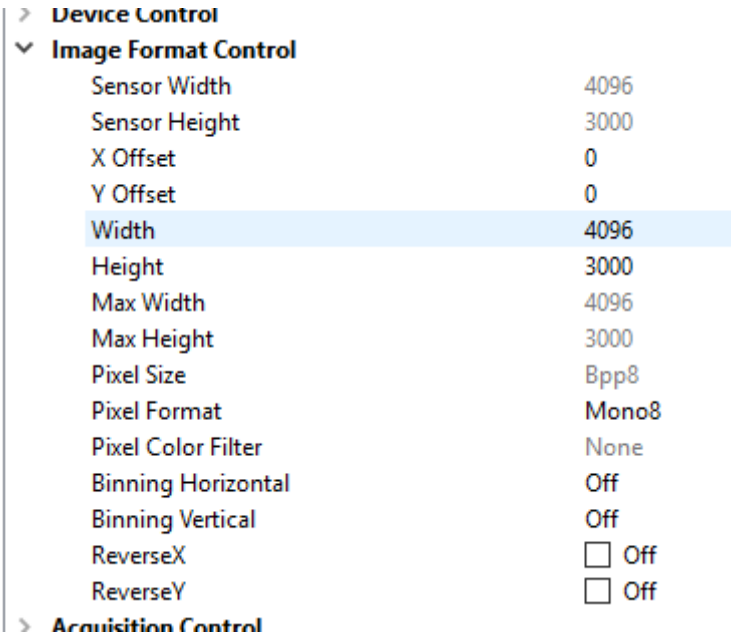


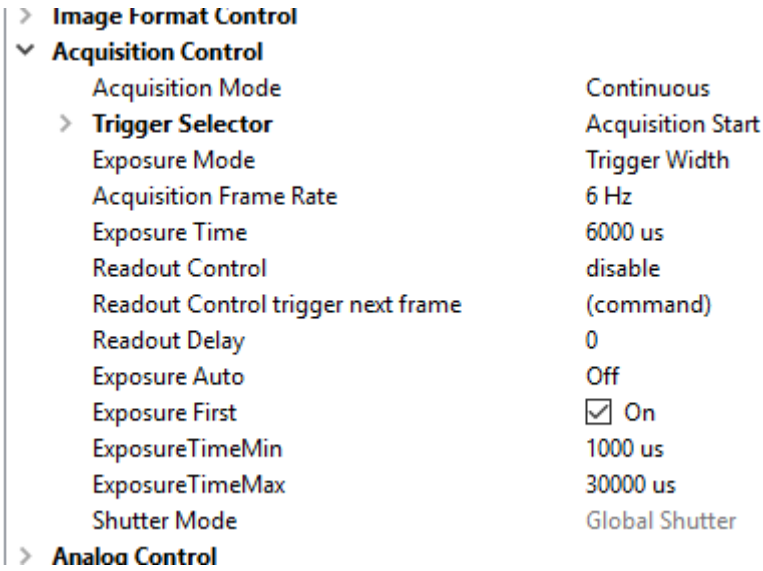
Image format control

Control bit-depth, size and position of the image



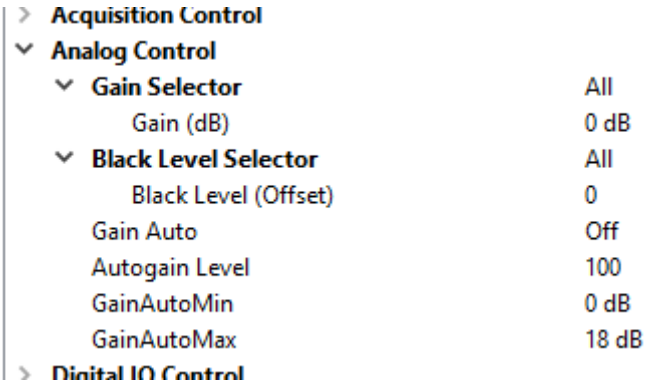
Acquisition control

Control trigger selection, exposure timing and –modes



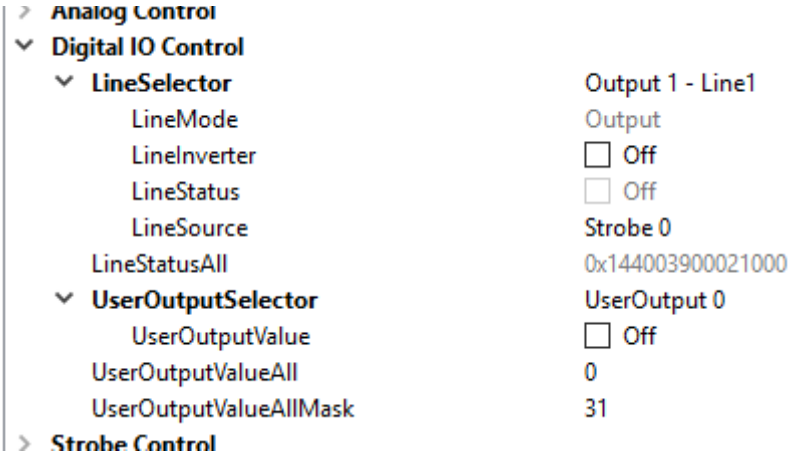
Analog control

Gain and blacklevel adjustments



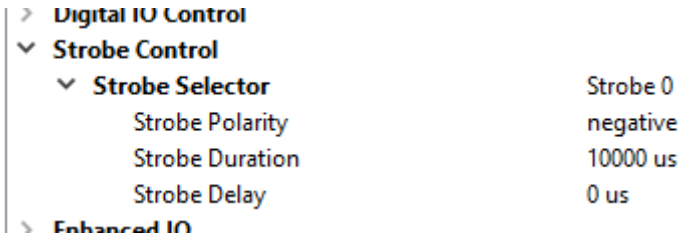
Digital IO control

Activate and connect inputs and outputs of I/O functionality modules. See IO assistant 3.5.2.



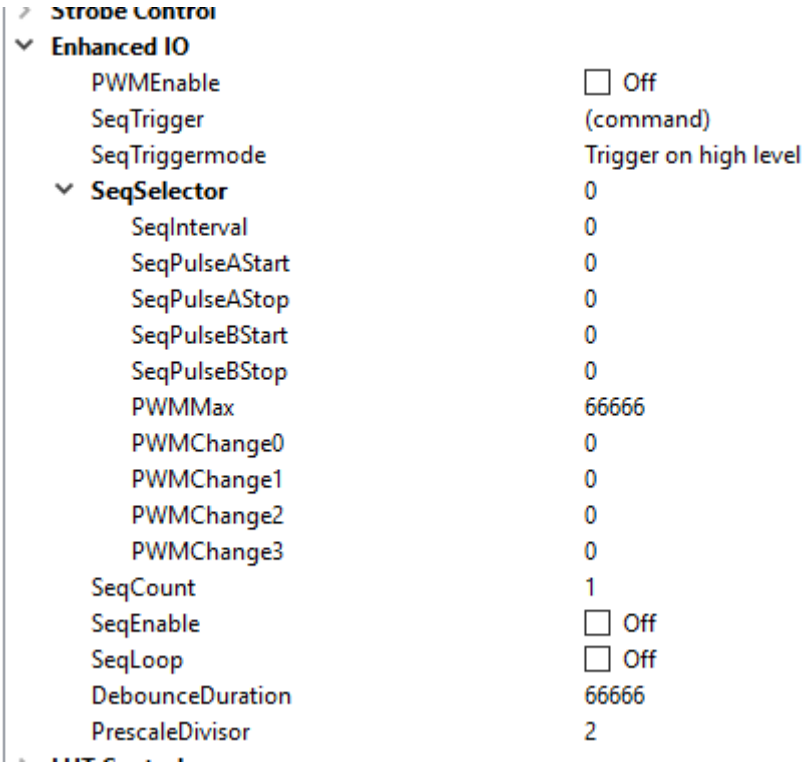
Strobe control

Adjust strobes for using PWM strobes without sequencer



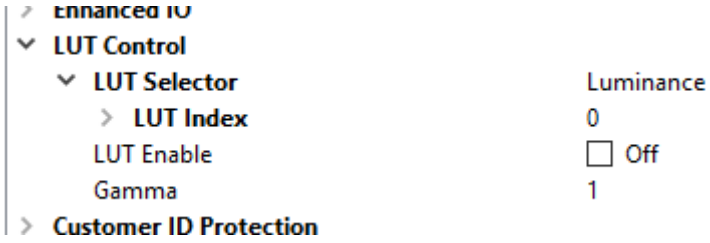
Enhanced IO

Adjust debouncer, sequencer, prescale divisor

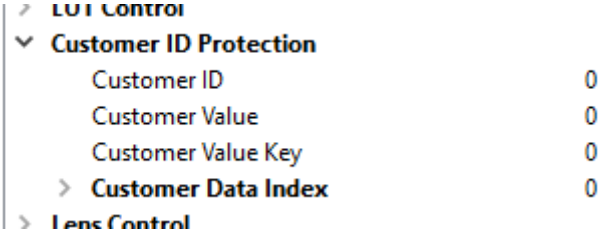


LUT control

Load and create various types of lookup tables

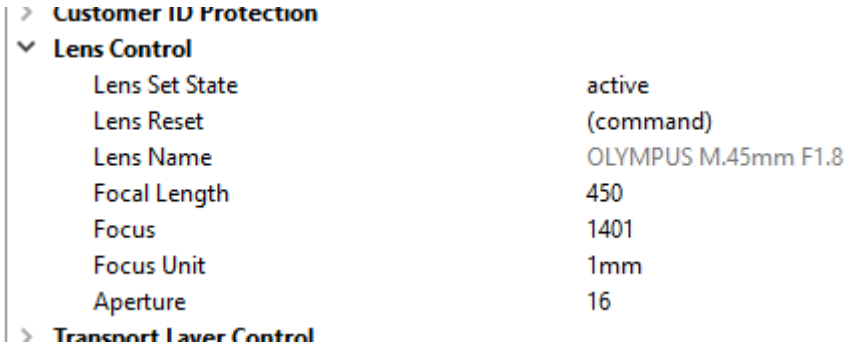


Customer ID protection



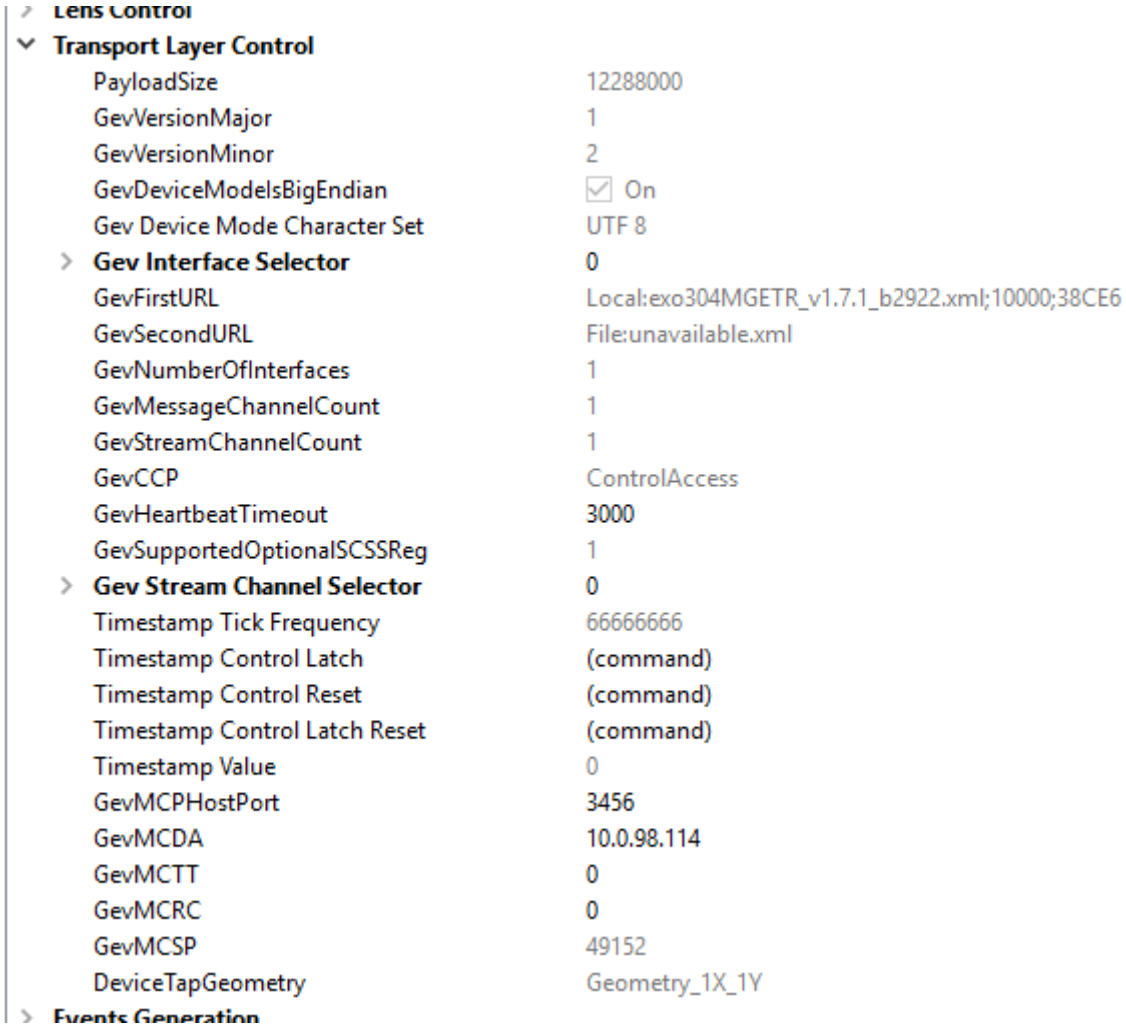
Lens control

Adjust attached lenses via RS232 interface like MFT, CANON EF, Varioptic



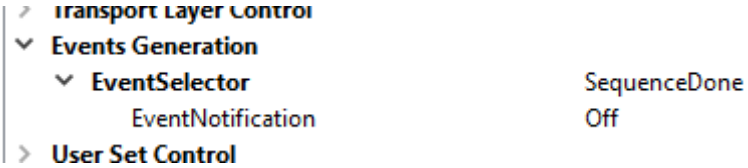
Transport layer control

Control the GenTL in the network



Events generation

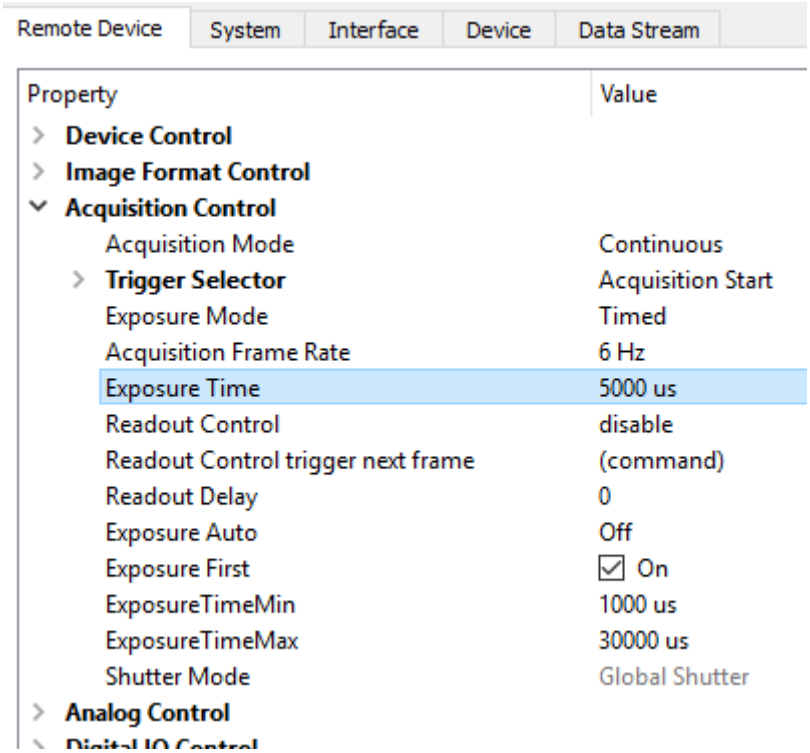
Determine which events should be reported



User set control

Load and save current camera configuration from/to user configuration memory

These function blocks contain device properties with values. By changing the property values the operation of the camera can be controlled.

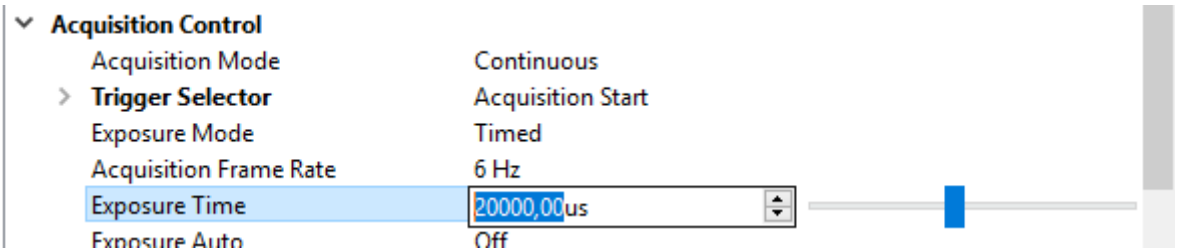


As soon as the camera is connected, the GenICam tree of the connected camera is displayed. In the example above, the camera is set to free run with 6fps and an exposure time of 5ms. If the start button has been pressed, the camera image will be displayed.

NOTE

Image content will be displayed in 8 bit only

Clicking the "**>**" of a block will show properties. Double click on a value will open the value editor. In some cases you'll find a slider at the right of the value for simplified input.



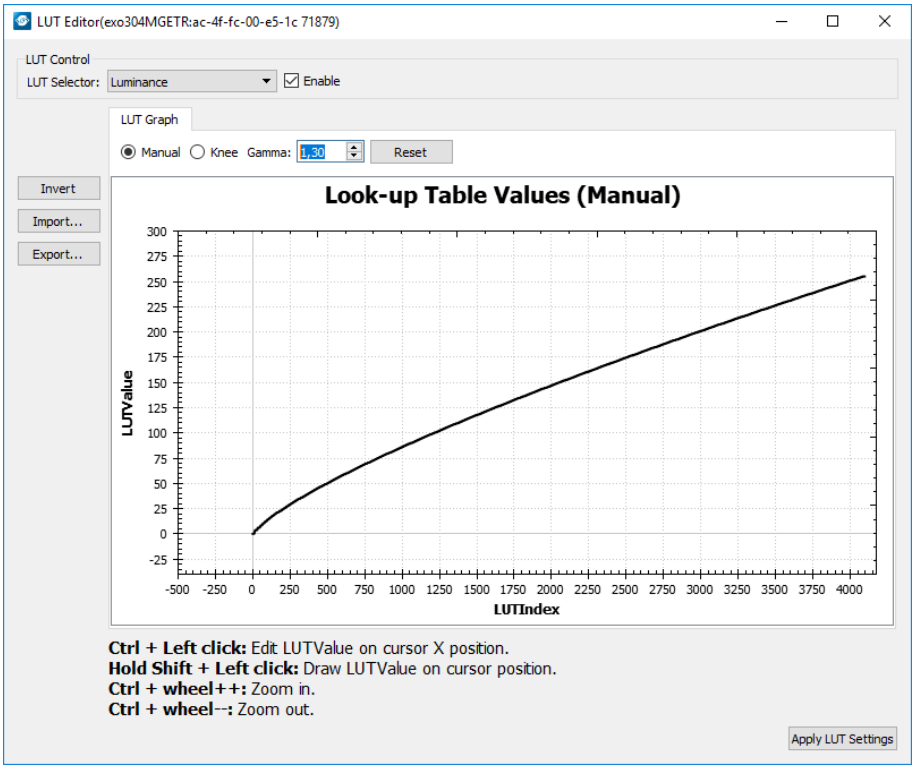
3.5 Assistants

The innovative SVS-Vistek 4I/O concept is unifying the operations of all input/output operations as well as signal processing features into the GenICam tree. Being a versatile system, SVCapture provides some assistants for simplified usage.

3.5.1 LUT creation

Some camera models provide the possibility to apply a custom lookup table to the image. Depending on the sensor type an 8-to-8, a 10-to-8 or a 12-to-8 bit lookup table can be applied. Basically, every luminance value will be replaced by a custom value. 12-to-8 LUT provide the advantage of fast 12-bit luminance processing while having a small, preprocessed 8-bit output for fast image transport to the host computer.

Create LUT with assistant



There are 3 possibilities to create LUTs with this tool:

- Gamma: Use a gamma value
- Manual: Edit freehand by drawing values
- Knee: Provide supporting points ('knee') which will be connected by straight lines

Enable LUT processing in the camera at the top line of the assistant, then download your custom LUT into the camera with the 'Apply' button

Read/write LUTs from file

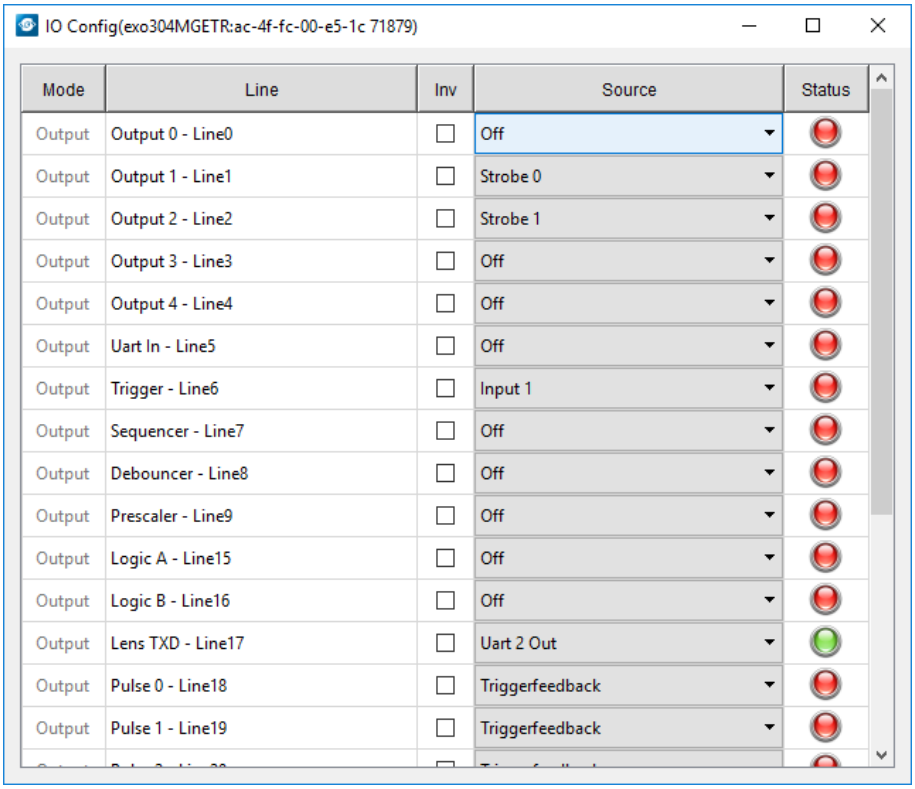
You can export and import your LUT as a file. It will be saved as XML file like this:

```
<?xml version="1.0" encoding="UTF-8"?>
<LUTSetting xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:xsd="http://www.w3.org/2001/XMLSchema">
    <LUTSelector enable="0" name="Luminance">
        <LUTData value="0" index="0"/>
        <LUTData value="0" index="1"/>
        <LUTData value="0" index="2"/>
…
```

The example file above is truncated. It is necessary to provide a target value for every index number (sensor value). This will result in 4096 LUTData lines for a 12 bit sensor readout.

3.5.2 I/O Config

The I/O Config assistant is the heart of your I/O and Strobe configuration. In one single window it provides you with the 4IO modules and their signal sources. The SVS-Vistek default defines Strobe0 connected to Output1.

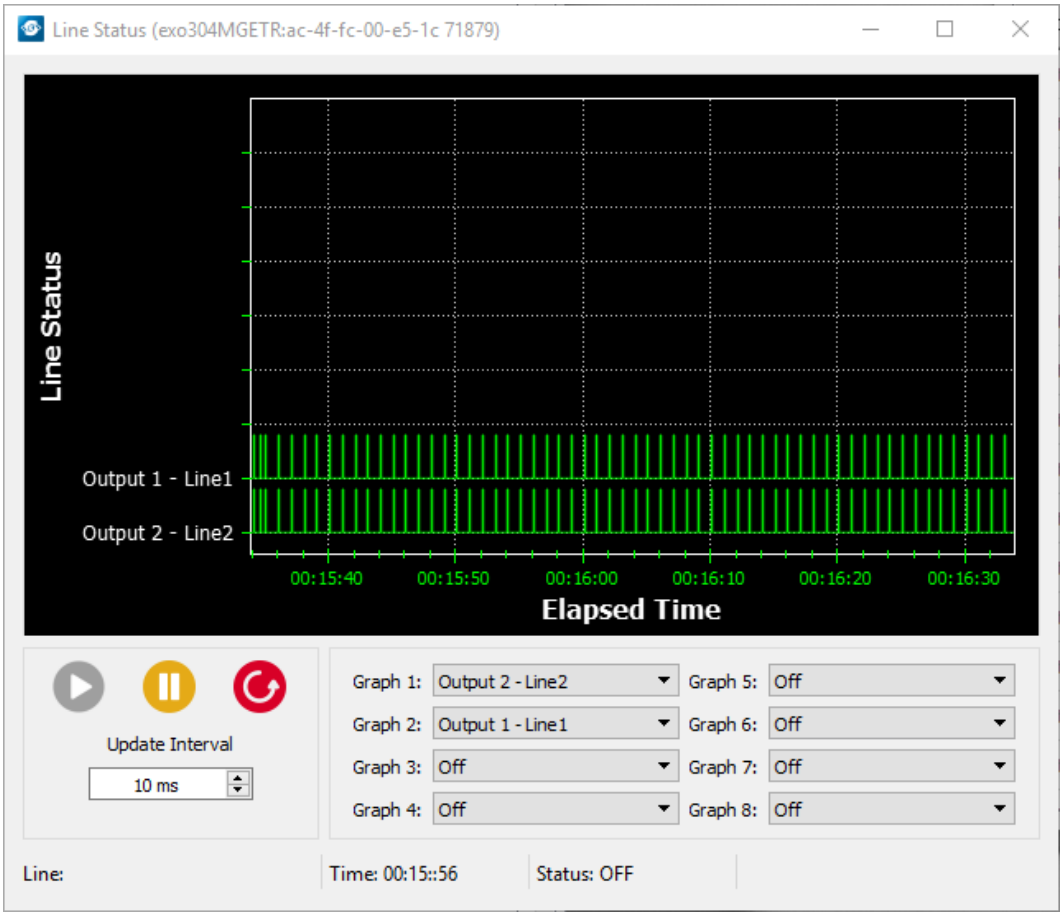


In the image above are – as an example – 2 strobe controllers connected to Output1 and Output2. On the left find you find the target module and you can select the input signal source on the right side for each module. "Inv" will will invert your signal (use this for testing).

All the configurations is visible in the GenICam tree. If you want to see what the assistant is doing, look into the GenICam tree. The IO Config does the module connection: Which module is getting its input from which input/module. The Modules themselves have to be activated and/or configured in the GenICam tree with their proper parameters (example: create here the connection Input – Debounce – Trigger. Configure debounce timing in the GenICam/Enhanced IO/Debounce duration)

3.5.3 Line status visualizer

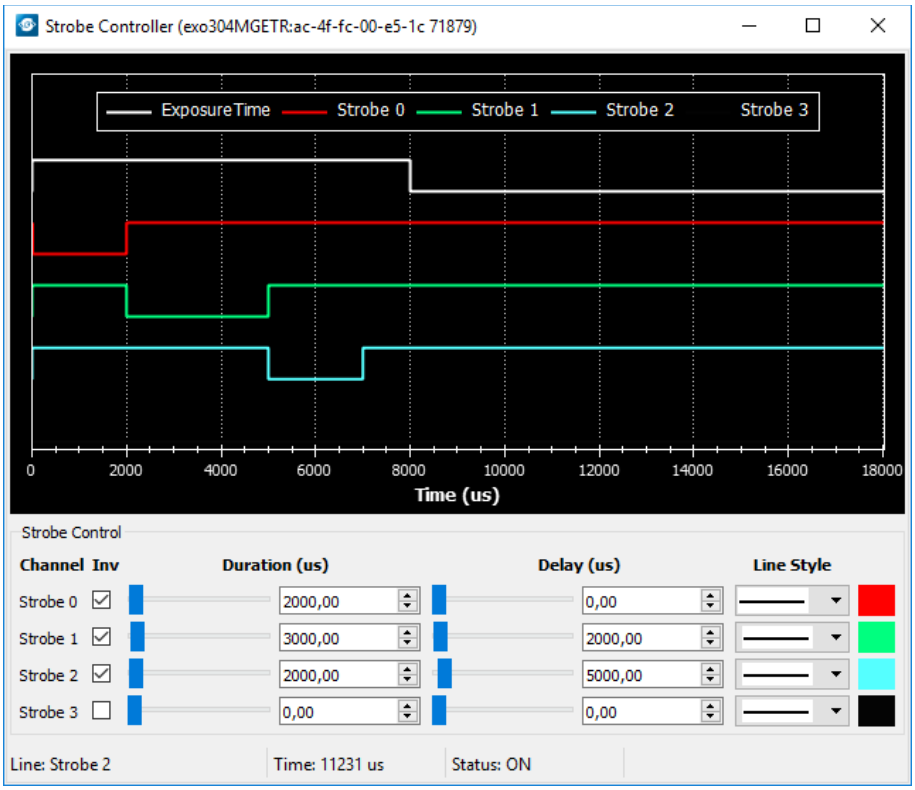
The line status visualizer shows the activity state of 4I/O modules versus time. Most common is the display for strobe activity as shown below (line 1 and line 2 status). It is possible to see different modules in a time correlation (e.g. trigger/debounce/exposure/flash). You might display any other signal being provided by the I/O modules like exposure or trigger signals. Select these in the „Graph“ drop-downs.



Make sure the update interval is short enough to cover your timings.

3.5.4 Strobe controller

Sometimes it does make sense to have a visual display of the active sensor exposure regarding the ON times of your strobes. The strobe controller assistant will do exactly this.



You can see what exposure time has been set in the GenICam tree. By modifying the values for duration and delay it is possible to adjust each of the strobe controllers. These adjustments will be visible in the GenICam tree immediately. Use your mouse wheel to zoom into time scale.

The image above shows 3 strobe controllers which are active in a sequence with different timings in one single exposure time

Please note, active exposure is HIGH, while active strobe controller display is LOW.

3.5.5 Sequencer

The sequencer is a core module of the 4I/O interface. You can program a sequence of images to be taken together with their specific I/O configuration, especially strobe light configuration. The sequencer is a possibility to save these items as a single sequence item:

- Exposure start
- Exposure stop
- Strobe start
- Strobe stop
- Strobe intensity

By programming these params into the sequencer, the camera will run through the whole sequence with a single trigger event (regardless whether software or hardware). Timings in the sequence assistants are done in µs [whereas in SDK programming the units are tics (=15ns)].

Example

For demonstration, imagine following task to be done:

Scenario

An object should be inspected with a monochrome camera. For accentuating different aspects of the image, 4 images should be taken in a row with 4 different colours of light: Red, Green, Blue, White. White light should be generated from the RGB lights being activated at the same time. Basis is a dark environment without other light sources.

Camera wiring

- 3 LED lights are physically connected to the camera on out 0-2 (red, green, blue)
- Out 3 is not used

I/O matrix

- 4 images to be taken (RGBW) result in 4 sequences
- RGB PWM change with different intensities (duty cycle) taking care for differences in spectral response of the camera sensor
- PWM change 0-2 is connected to out 0-2
- Seq pulse A is driving the exposure (trigger)
- Seq pulse B is driving the strobe
- Seq pulse B in WHITE sequence is reduced down to 33% as light intensities of 3 lights (RGB) will add up

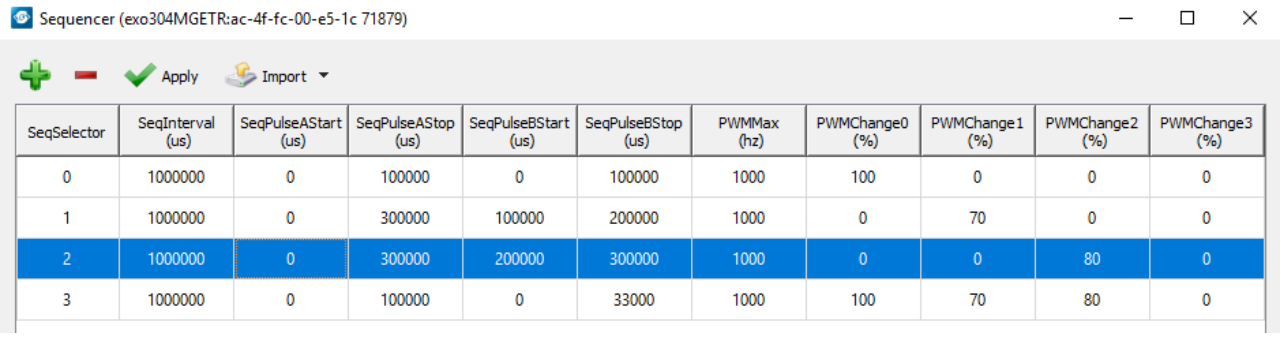
Notes

- Different exposure / strobe timings are used for illustration. In most cases they will show values same as exposure
- The resulting exposure time shows the period of sensor light exposure. (''masking'' of exposure time by creating strobe light impulses shorter than exposure time). This value is not adjustable at the camera

- PWM change is shown with reduced height for demonstrating reduced intensity. In reality though, PWM change will be full height (full voltage, shunt resistor might be necessary) with the adjusted duty cycle
- Use a PWM frequency high enough not to interfere with your timings (here: 1000 Hz)

Scenario values	Interval 0 (RED)	Interval 1 (GREEN)	Interval 2 (BLUE)	Interval 3 (WHITE)
Sequencer Interval	1000 ms	1000 ms	1000 ms	1000 ms
Seq pulse A start	0 ms	0 ms	100 ms	0 ms
Seq pulse A stop	100 ms	300 ms	300 ms	100 ms
Seq pulse B start	0 ms	100 ms	200 ms	0 ms
Seq pulse B stop	100 ms	200 ms	300 ms	33 ms
PWM Frequency f	1000 Hz	1000 Hz	1000 Hz	1000 Hz
PWM change 0 (RED)	100%	0%	0%	100%
PWM change 1 (GREEN)	0%	70%	0%	70%
PWM change 2 (BLUE)	0%	0%	80%	80%
PWM change 3	-	-	-	-

These values put into the sequencer would result in the sequencer assistant like this:

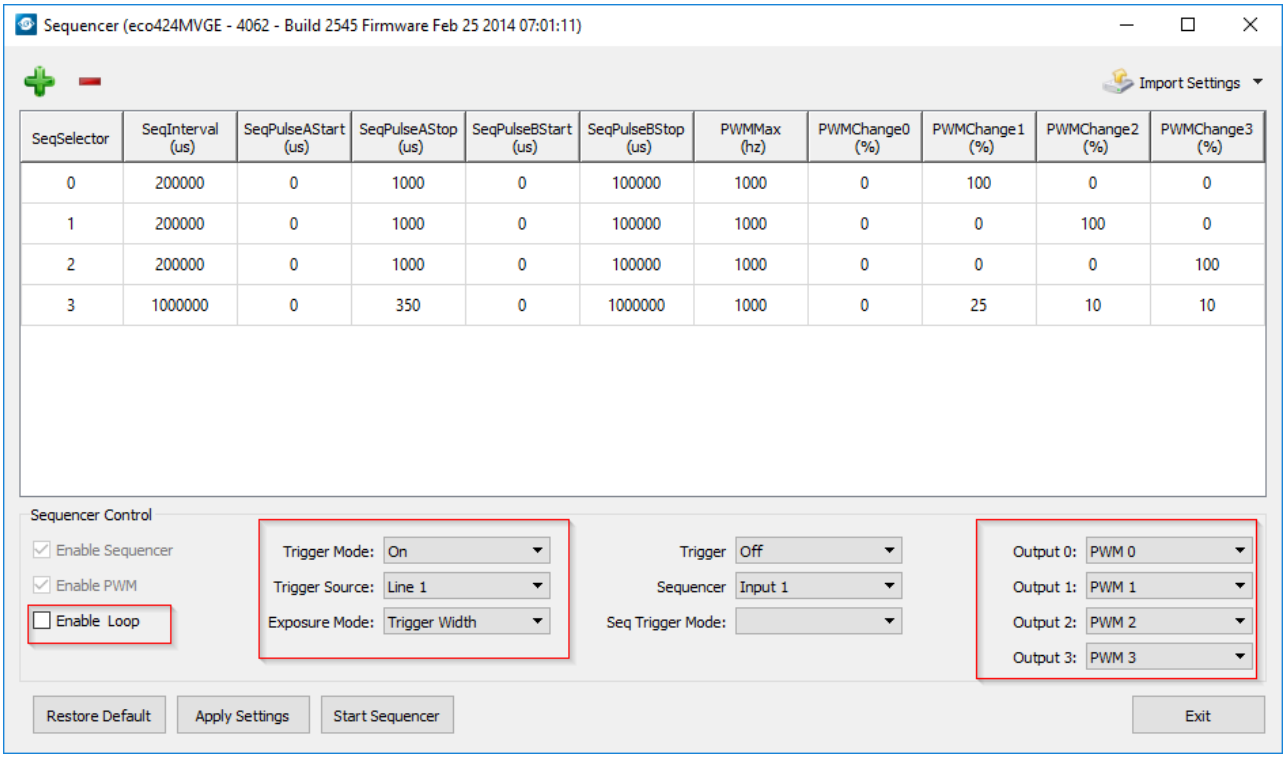


SeqSelector	SeqInterval (us)	SeqPulseAStart (us)	SeqPulseAStop (us)	SeqPulseBStart (us)	SeqPulseBStop (us)	PWMMax (hz)	PWMChange0 (%)	PWMChange1 (%)	PWMChange2 (%)	PWMChange3 (%)
0	1000000	0	100000	0	100000	1000	100	0	0	0
1	1000000	0	300000	100000	200000	1000	0	70	0	0
2	1000000	0	300000	200000	300000	1000	0	0	80	0
3	1000000	0	100000	0	33000	1000	100	70	80	0

SeqSelector = Sequence counter (max 32 depending on model)
seq pulse A start = Exposure time start
seq pulse A stop = Exposure time stop
seq pulse B start = Strobe time start
seq pulse B stop = Strobe time stop
Exposure time = (seq pulse A stop – seq pulse A start)
Strobe time = (seq pulse B stop – seq pulse B start)
PWMMax = PWM output frequency
PWMChange = Strobe intensity in %

Basically, every exposure starts at a certain point of time (“SeqPulseAStart“) and stops a certain amount of µs later (“SeqPulseAStop“). Same thing with light output (“SeqPulseBStart“) and (“SeqPulseBStop“).

The exposure trigger comes from SeqPulseA. The sequence trigger for starting the sequence (line 7) comes from Input 1. As soon as the sequencer assistant has applied the values, the values are visible in the GenICam tree.



Please note, you have to assign the physical outputs as well to their logical correspondants. The four hardware outputs 0-3 of the camera (where your lights are attached on) are corresponding to PWMChange0-3. The example above is running only 3 outputs (outputs 1,2,3), output 0 is not operated (no entry in PWMChange0).

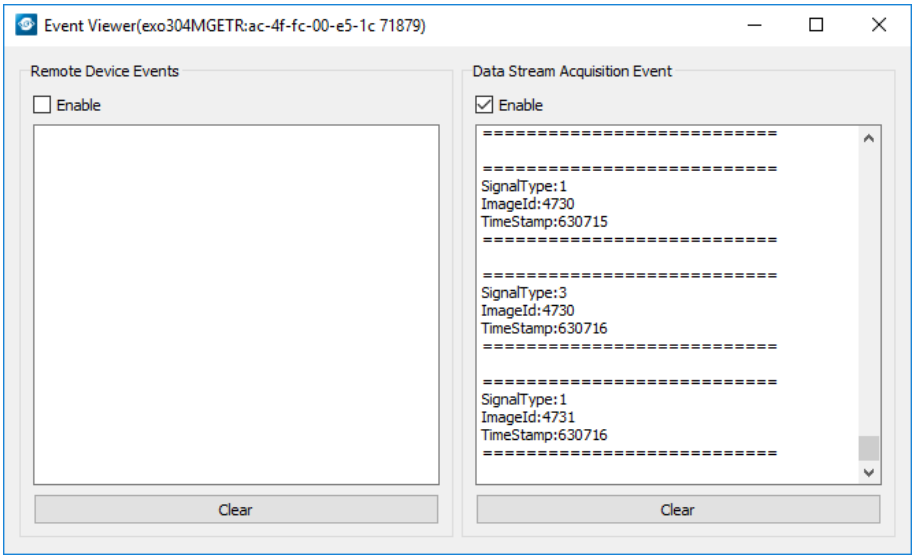
Up till now, the sequencer hasn't been activated, though. Activate the sequencer by selecting Trigger Mode, Trigger Source and Exposure mode as above. Apply Settings to download your sequencer setup into the camera.

The sequencer needs an input to start (line1). Clicking ‚Start Sequence' will initiate the sequence once also without hardware input signal. With loop enabled, it will run forever.

In the GenICam tree, you can run the (already configured) sequencer with the SeqTrigger command like this (same thing as „Start sequencer" in the sequencer assistant):

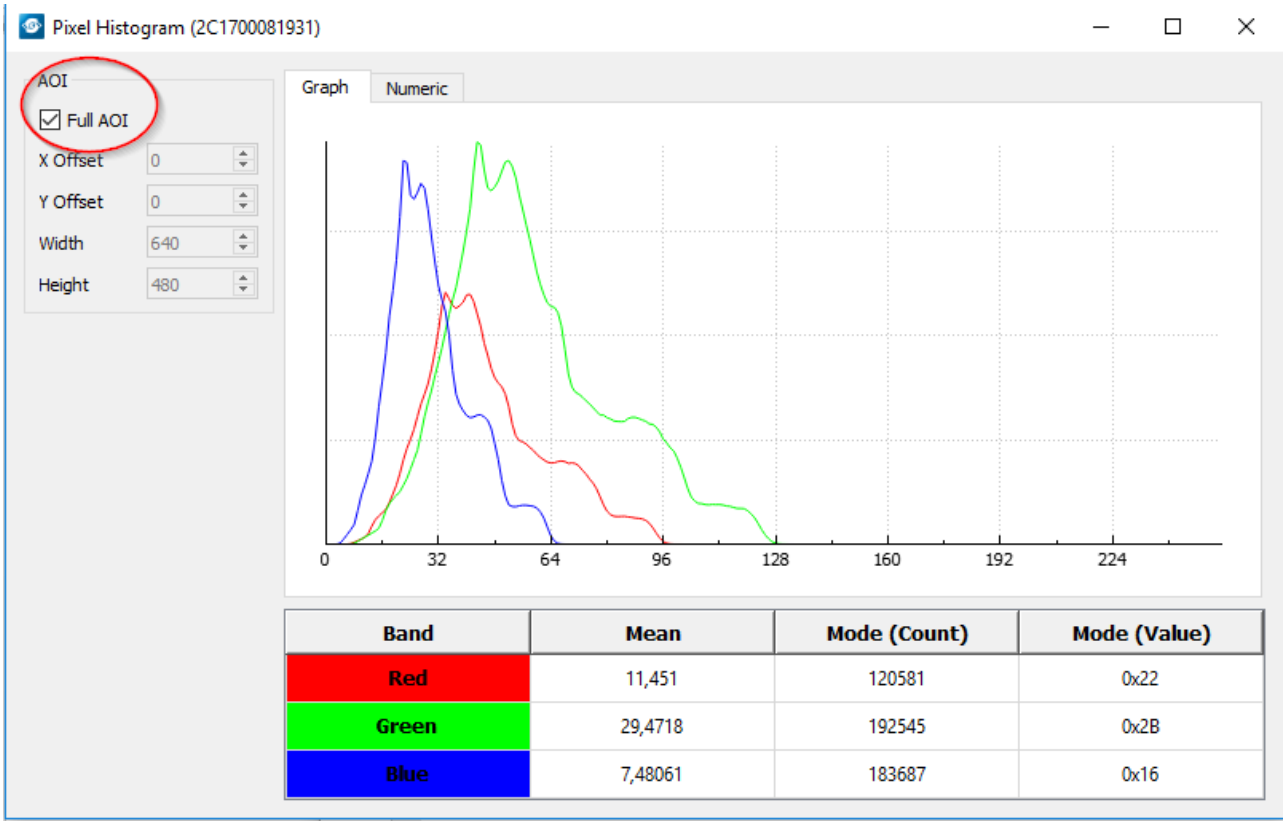
3.5.6 Event viewer

The event viewer might be interesting when developing software to see camera events.



3.5.7 Pixel Histogram

The pixel histogram shows the distribution of all your pixel intensities. Use this for getting an idea about whether your exposure is using the full sensor's optical range in intensity. (example below would suggest longer exposure time)



Reduce the measured area to your specific region of interest by adjusting the AOI.

3.5.8 Shading assistant

NOTE
Shading correction is a camera model-dependant feature. The assistant might be disabled when connected to a camera without this feature.

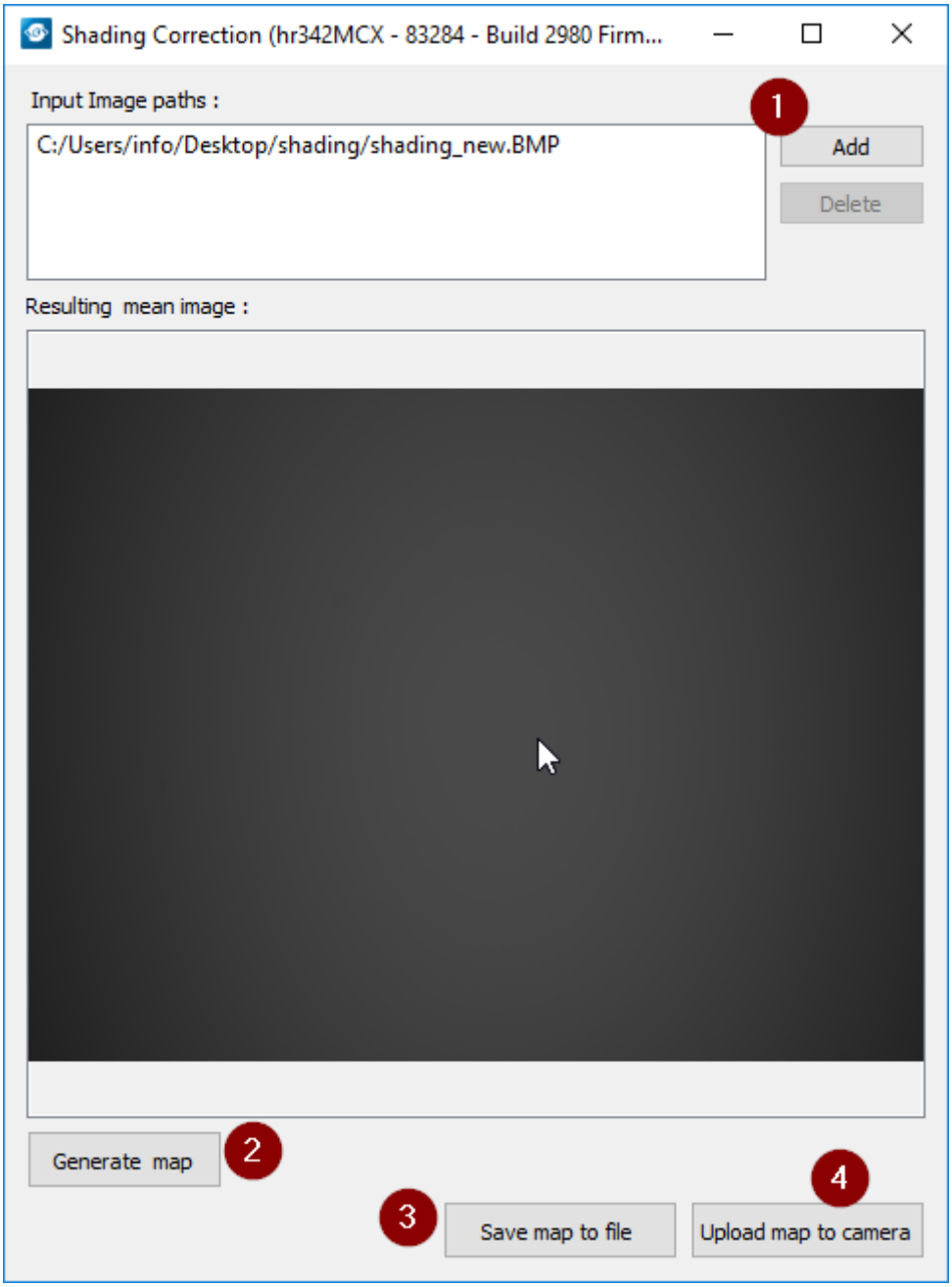
Shading correction is a process in which an uneven illumination of the image on the sensor is corrected. The uneven illumination might have 3 sources: An uneven sensor sensitivity over the surface, a lens shading or an uneven illumination. Modern sensors deliver a very even image, so that shading correction nowadays is almost exclusively used for correcting the lens shading or for correcting difficult lighting situations. Doing the shading in the camera is providing better dynamic range than doing it in the machine vision postprocessing.

SVCapture is providing an assistant for this, making the process fairly simple. You create a take an image (showing the shading problem), create a shading map information out of it, load it into the camera and activate it. These are the steps:

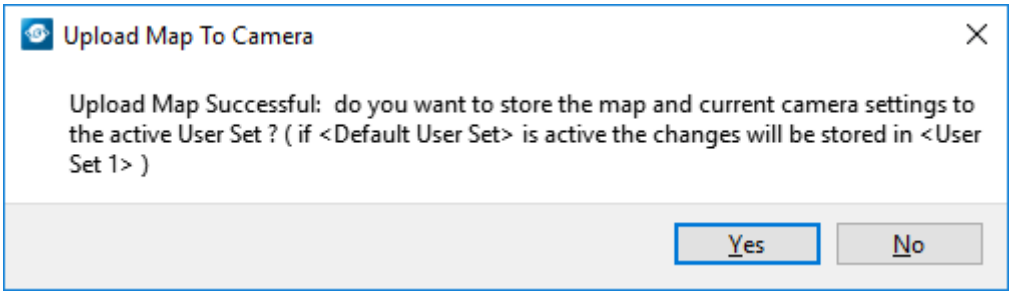
Create a shading map

Take a single image without the object to be shown later. The image should have as little structure as possible. Best would be just a flat grey surface, thus displaying only uneven illumination / lens. Take an image from this, save it as bitmap (BMP). All deviations from average brighness values will be corrected in future images.

1. Open the shading assistant and load the image just made into the assistant.
2. Generate the map
3. You might want to save the resulting file (optional)
4. Upload the map to the camera



This procedure will create the shading map into the camera and activate it. The shading map as below will be saved in User set 1.



3.5.9 Defect Pixel assistant

NOTE
Defect pixel correction is a camera model-dependant feature. The assistant and the feature might be unavailable when connected to a camera without this feature.

All CCD or CMOS sensors do have some defect pixels (dead or hot). The camera will correct these defect pixels.

Automatic defect pixel correction

There is some automatic correction of defect pixels, you might enable this in the GenICam tree. This works without user interaction, it has to be enabled.

Defect pixel correction with maps

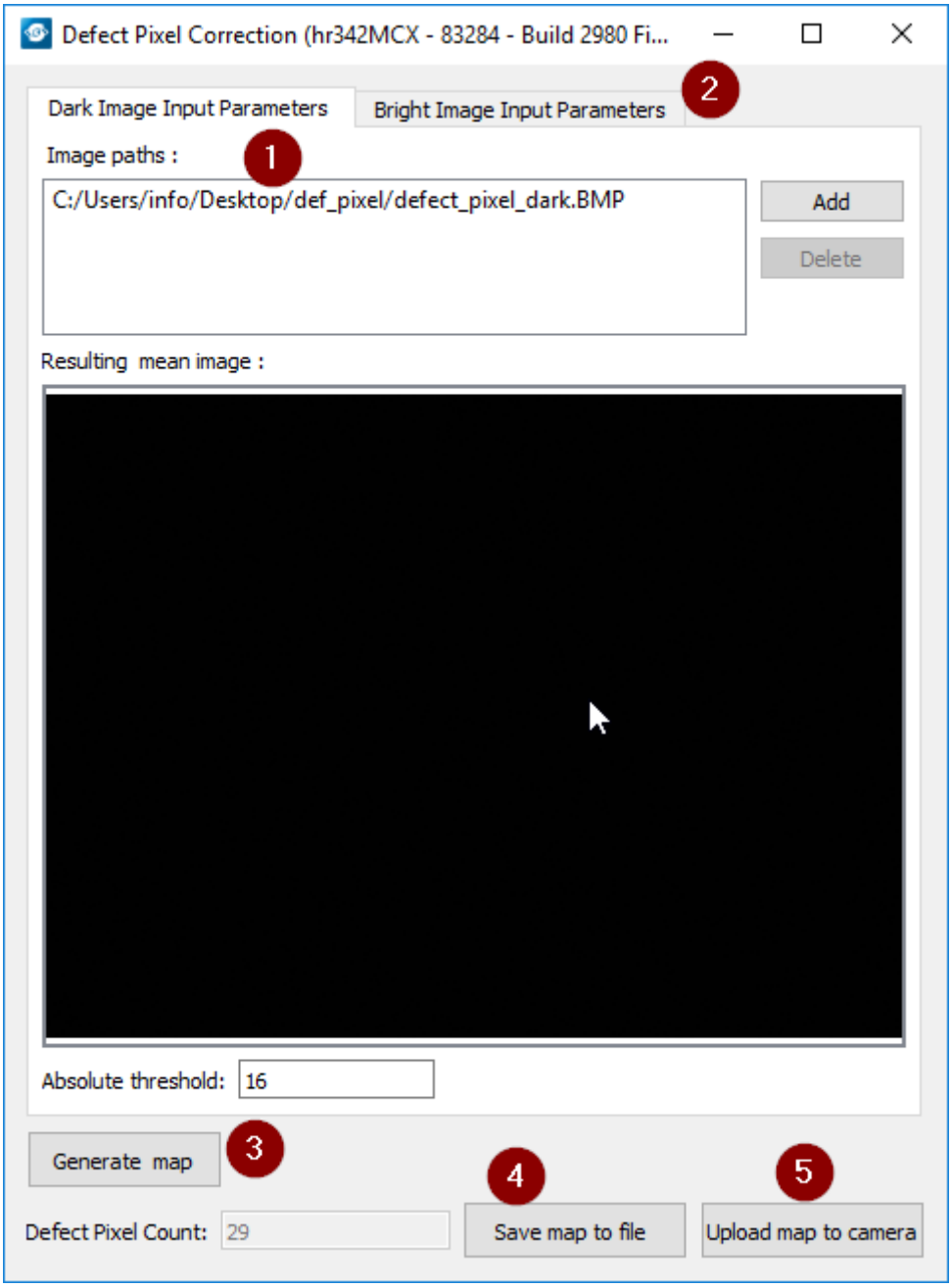
If you want to have more control regarding defect pixel correction, a defect pixel correction with defect pixel maps is the way to go. You need two images, a dark image and a bright image for detection of dead pixels and hot pixels.

Dark image:
Take an image with lens cap closed, no light on the sensor. Save the image as BMP.

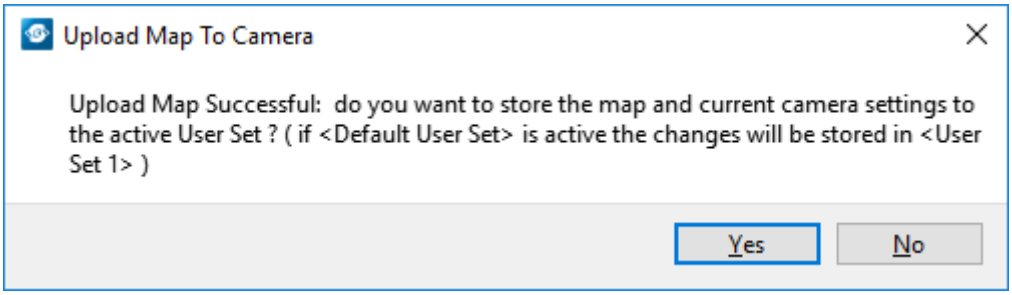
Bright image:
Take an image of an very homogenous flat area, best would be white surface with pixel intensity around 200. Save it as BMP.

Open the defect pixel generate map assistant

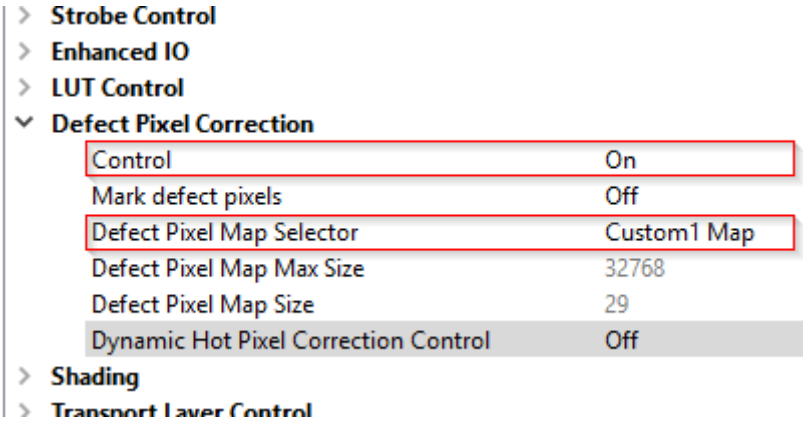
1. import the dark image
2. import the bright image
3. generate the map
4. save map to file (optional)
5. upload the map to the camera



The assistant will upload the map into the camera.



You can select / adjust defect pixel correction in the GenICam tree. Instead of your custom map, the factory setting can be selected.



3.6 Common procedures

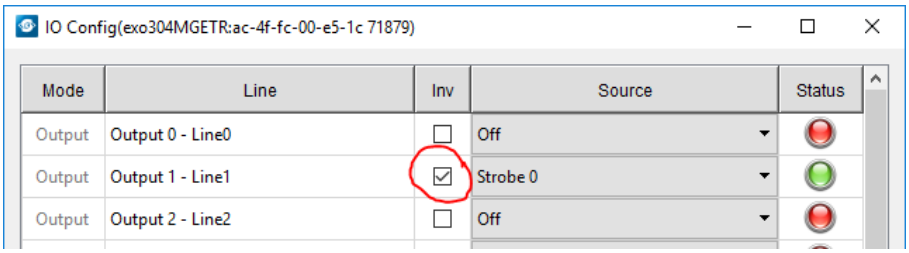
This chapter provides you common procedures for standard tasks, based on SVCapture 2.5.4 assistants. All these tasks can be done as well directly in the GenICam tree without assistants or programmatically with the SDK development files by modifying the GenICam tree.

3.6.1 Check strobe light

One of the basics of camera setup is to check whether the connected strobe light is properly connected. In this example, let's suppose the strobe light is connected to OUT1 on your camera's Hirose 12-pin connector. Easiest way to find out the correct electrical setup of strobe connection:

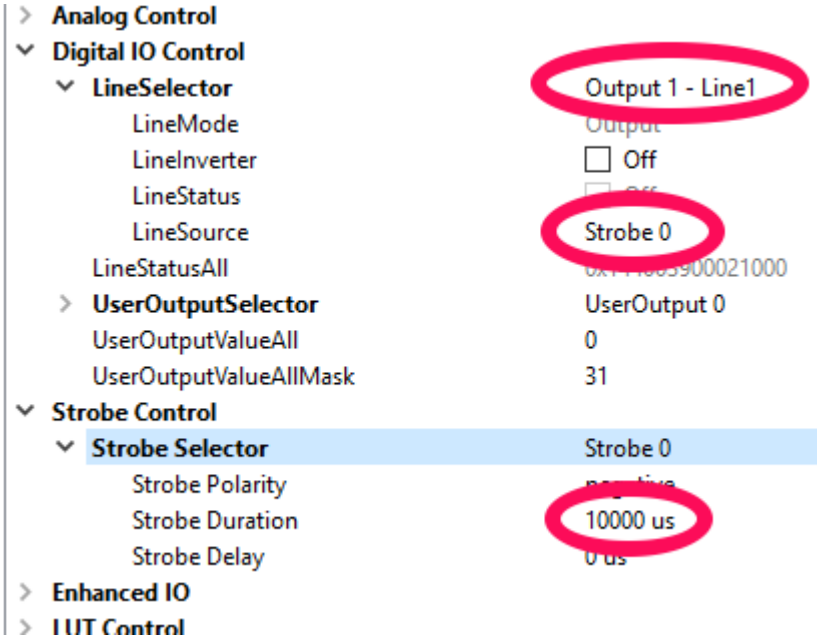
WARNING
The SVCam I/O module is supporting high currents and supply voltages up to 24V. Output will always run with current supply voltage. To avoid damage of your light equipment, make sure electrical setup is with correct shunt resistor. See appendix 4.1

For testing your correct physical connection of your lights being attached to the camera. Open IO Config assistant, click the "Inv" box of your Output line. Your strobe light should be permanently ON now, status will go green. Again, be careful with maximum current and voltage of your LED lights.



3.6.2 Setup single strobe light

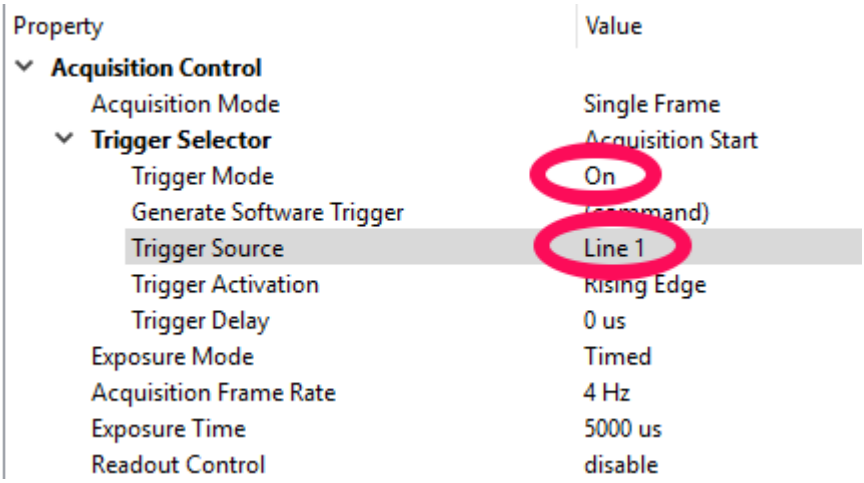
In this example, suppose your LED light is connected to OUT1 pin, please see chapter 4.1. In the GenICam tree, first set strobe0 as source for our OUT1 pin of the Hirose ("output 1") connector. Specify necessary strobe duration.



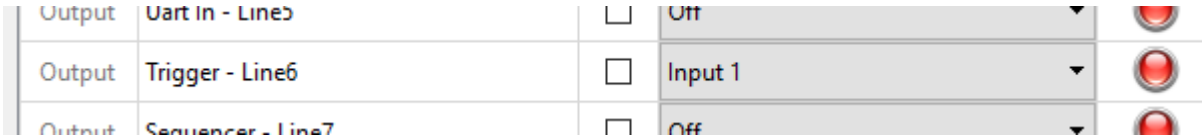
3.6.3 Set input to physical trigger input

The SVCam cameras have various electrical inputs, please refer to the Hirose connector pin layout of your specific camera model. For hardware trigger you have to connect your hardware input to the trigger event. The example below is supposing hardware input from INPUT1 (Line 1).

Enable hardware trigger in the GenICam tree (together with edge definition and delay in µs)



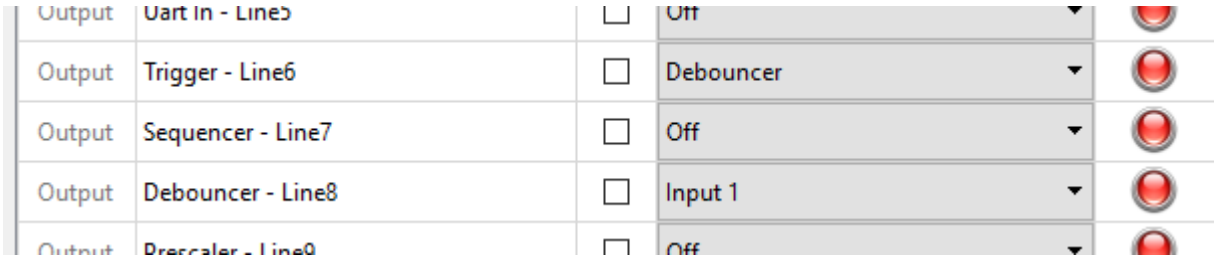
Connect the hardware INPUT1 line to trigger signal



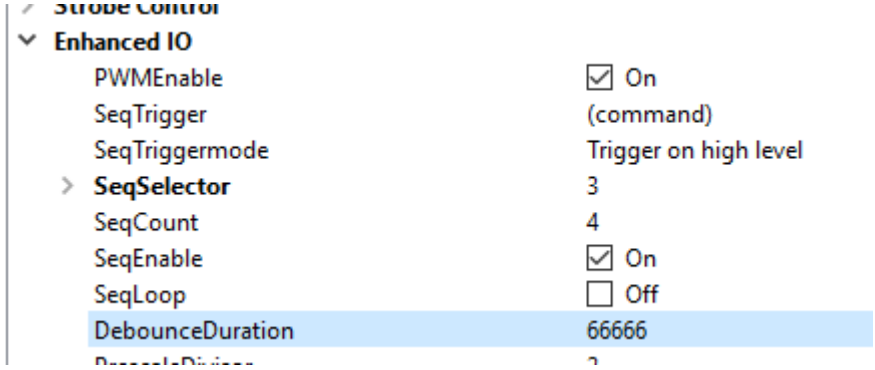
3.6.4 Debounce input trigger

The debouncer will make sure you have a clean trigger signal without spikes and wrong trigger events. In case of a trigger signal on your input, it will issue a valid trigger signal only if the input is still triggering after a (configurable) amount of time.

Enable debounce on INPUT1 in your I/O assistant



You need to configure debounce duration in your GenICam tree



NOTE

amount of time in this adjustment is in *tics* unit (15ns). In the example above the debounce duration would be

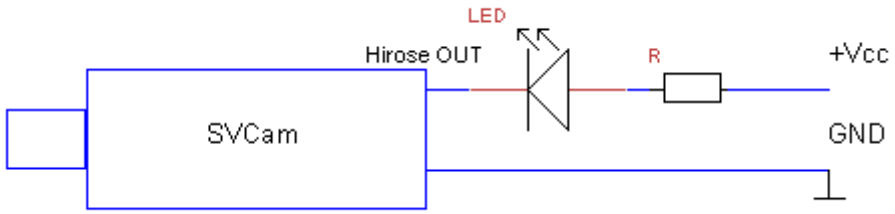
$$debounce\ duration = 15ns\ x\ 66666\ tics = 1ms$$

4 Appendix

4.1 Connect strobe to camera

LED strobe lights are available with different operating voltages from 2.1V to 24V. Max currents can vary widely. The integrated open drain output of the camera supports currents up to 1A max. The LED has to be ledged to GND, that is:

- Connect positive strobe power to shunt resistor
- Connect shunt resistor to anode lead of LED light
- Connect cathode lead of LED light to camera OUT
- Connect negative strobe power to camera GND (only necessary with separate power supplies for camera and LED)



Shunt resistor calculation

R	shunt resistor impedance
U_{VCC}	positive power supply
GND	Ground, negative power supply
I_{LED}	max current LED
U_{LED}	LED operating voltage
P_R	dissipation power shunt resistor

Formulae

$$R = \frac{U_{VCC} - U_{LED}}{I_{LED}}$$

$$P_R = I_{LED}(U_{VCC} - U_{LED})$$

Example

Let's find values for a (camera and LED) power supply of 24V, an LED light with 18V@100mA.

$$R = \frac{24V - 18V}{0.1A} = 60\Omega$$

$$P_R = 0.1A(24V - 18V) = 0.6W$$

You need a shunt resistor of 60 Ohms, capable of 0.6W power dissipation. You might consider to reduce the power supply to 18V (shunt resistor calculation shows 0 Ohm then, so no shunt necessary).