SVCam Kit
Quick Guide

Version 2.5.2

SVS-Vistek GmbH
Muehlbachstr. 20
82229 Seefeld
Germany

SVCam Kit Installation Guide

Current SVCam Kit software version used in this document: 2.5.2 / 2018-09-20

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. 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.

CONTENT

1 Introduction

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

If you purchased a CoaXPress camera, refer to the manual of your CoaXPress grabber manufacturer.

1.1 Requirements

Windows 7 is minimal Windows version. 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.

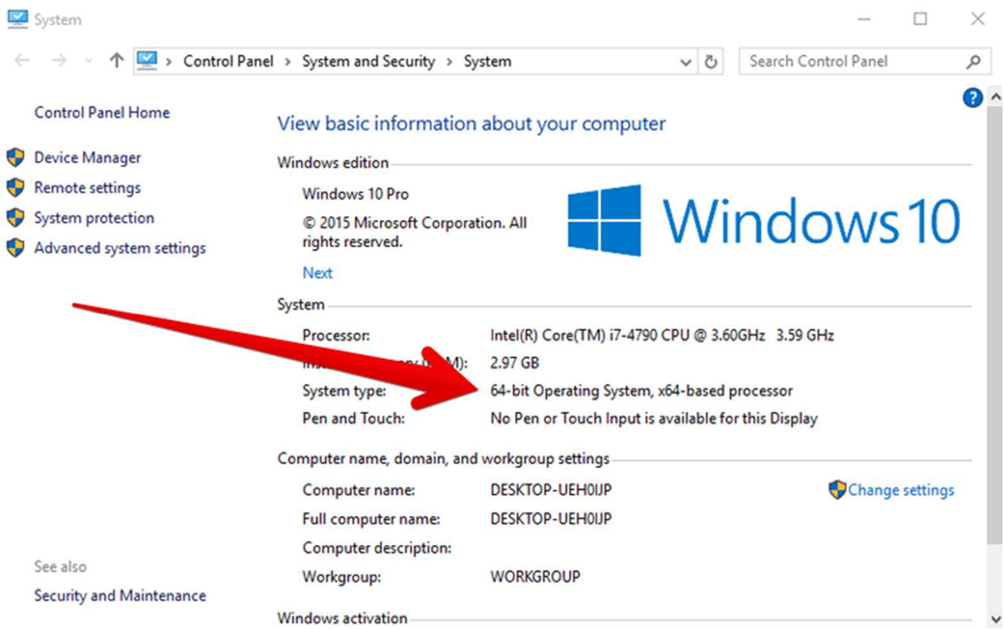
2 Install SVCam Kit

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_v2.5.2._x64.exe or SVCam_Kit_v2.5.2._x86.exe) matching your windows version. This is how to determine your system type:

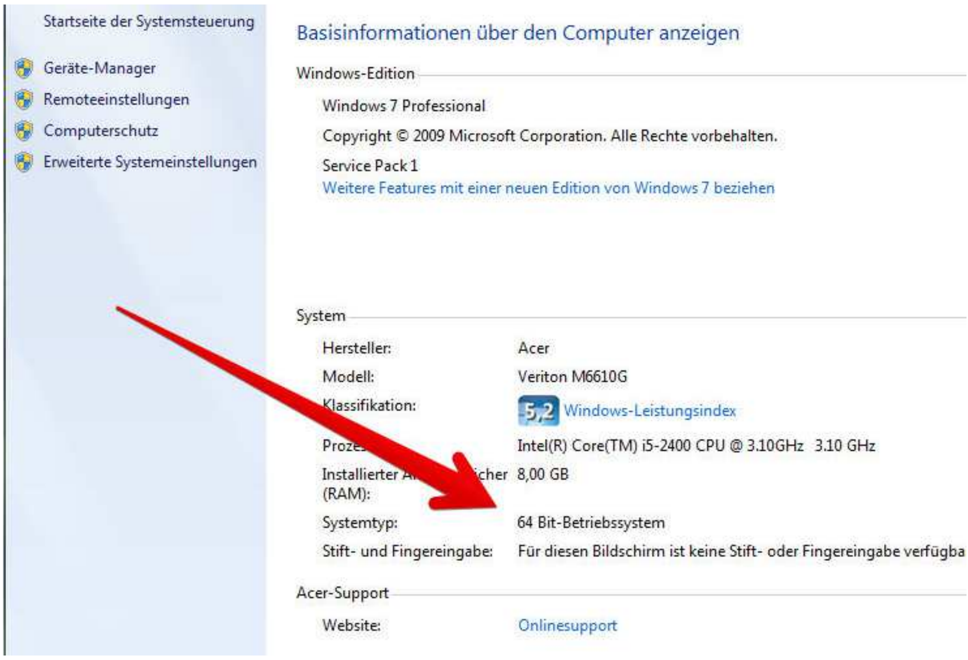
Windows 8 and Windows 10

Go to Start > Control Panel > System and Security



Windows 7

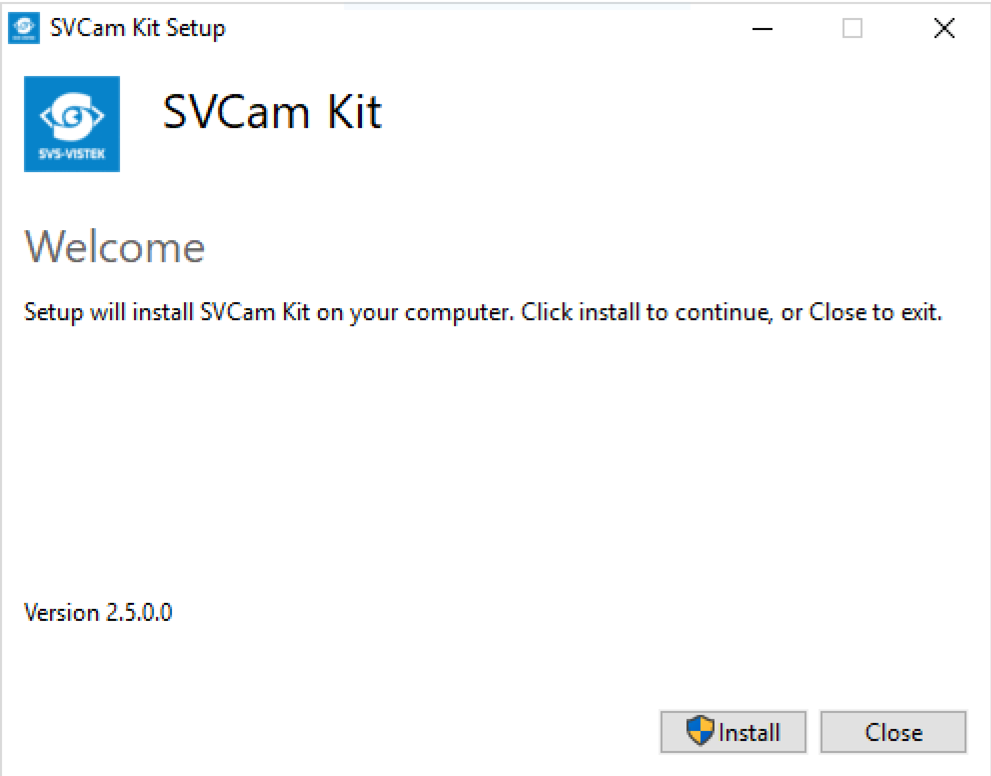
Go to Start > Control Panel > System



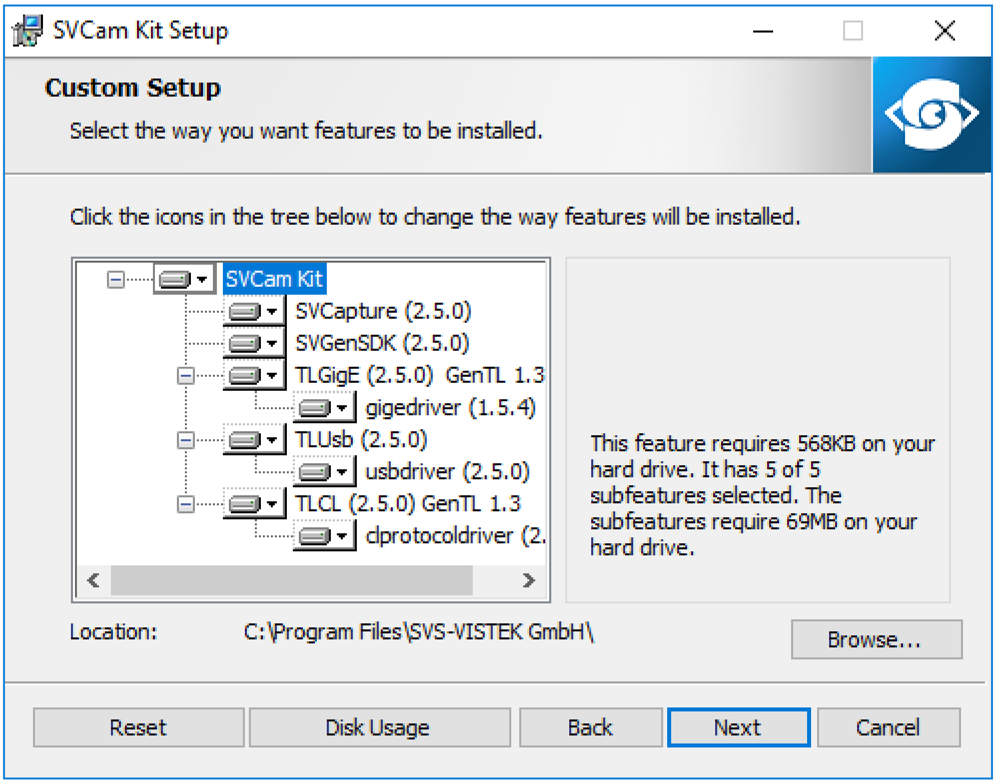
2.1 Software installation

It is strongly recommended to uninstall the 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) 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

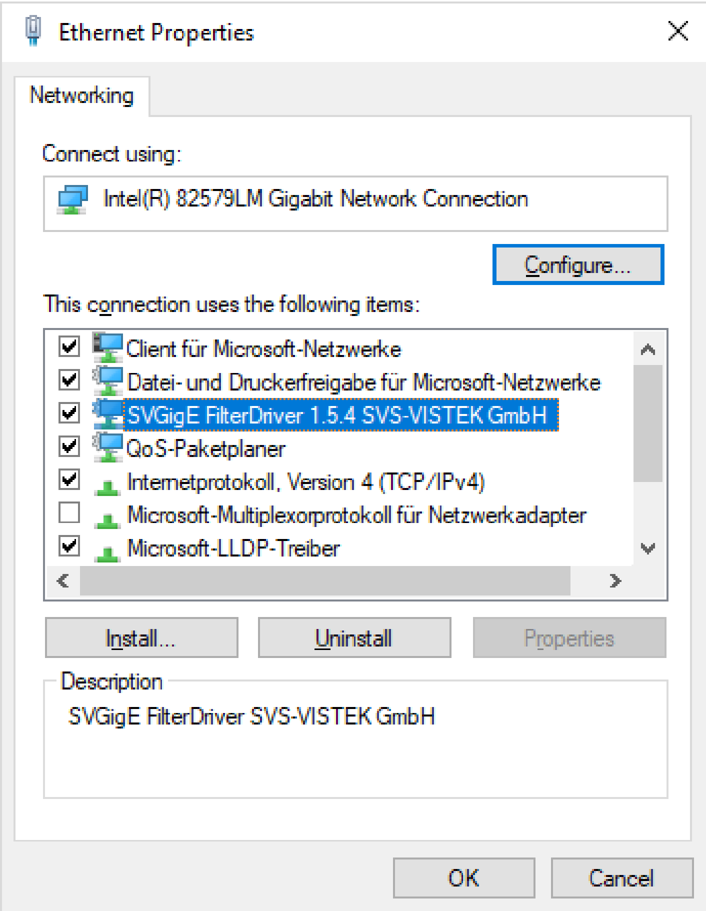
During the installation process, the required files will be written to your hard driveand the requested modifications will be automatically executed. Please verify as shown below.

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:

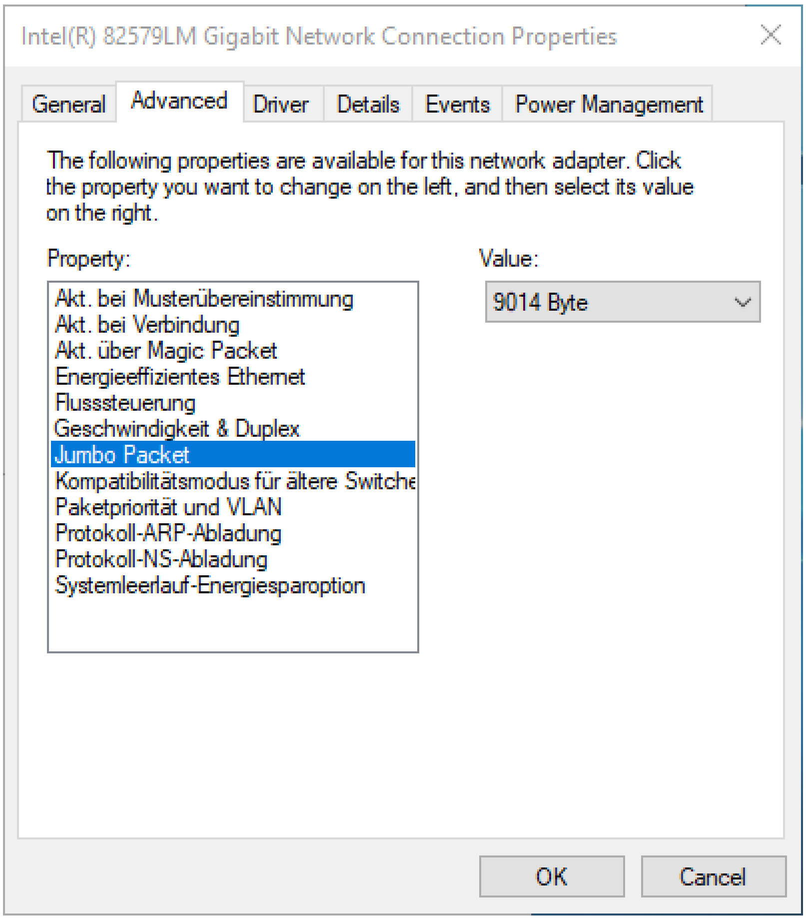
System control > network & internet > ethernet > change adapter options > Ethernet > Properties



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

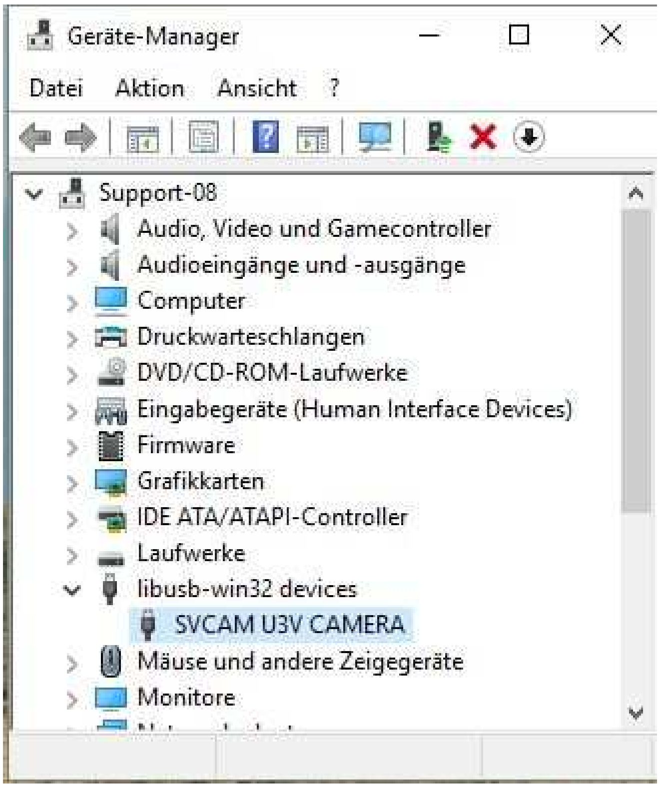


2.2.3 USB3 driver

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



3 Operate SVCapture

SVCapture is part of SVCam Kit and is recommended to be installed. It supports USB3, GigE Vision and Camera Link cameras. It is a tool with three main tasks:

- Display / edit the GenICam feature tree
  This permits you to see current camera settings, modify them and thereby to control the camera
- Display the current camera image
  Show the image which is acquired currently with your camera. Image display is supported for GigE and USB3 cameras only. If you want to view the image produced by a Camera Link device, please refer to the viewer software delivered with your Camera Link interface card.
- Provide help in adjusting the camera
  With SVCapture's all-new assistants, optimal parametrizing of the camera's imaging and I/O modules is easier than ever before

SVCapture will display the image being recorded by your camera – depending on the current parameters in its GenICam tree. As soon as you change GenICam parameters, you will see the resulting changes in the image. This makes SVCapture a nice tool to test and fine tune the GenICam parameters for your application.

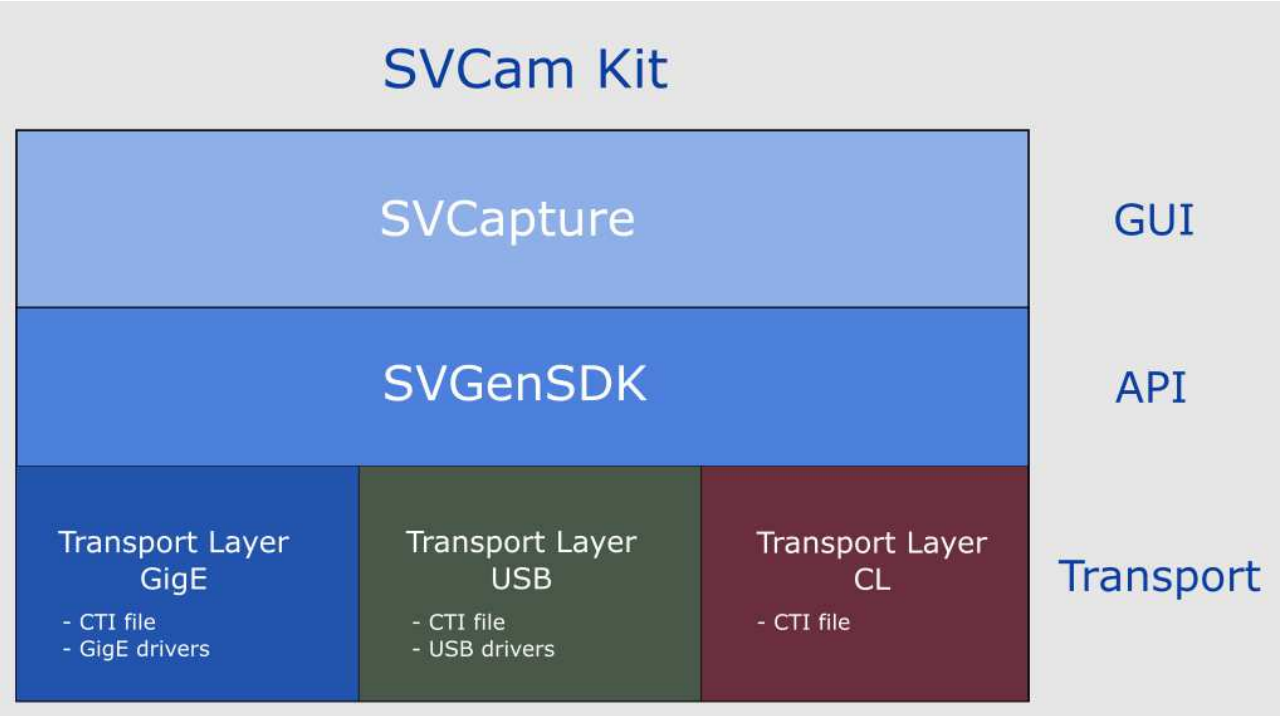
Note: SVCapture supports Camera Link cameras with a minimum firmware version as below

camera	firmware version
hr120xCL	b2927
hr25000xCL, hr25xCL	b2931

SVCapture is created on top of the SVCam Kit SDK. The SVCam Kit GenICam API provides unified methods for camera access, regardless of the interface type. Thus SVCapture is working with all 3 supported interfaces: USB3, GigE, Camera Link. Drivers for Camera Link are supplied by the manufacturer of your camera interface card.

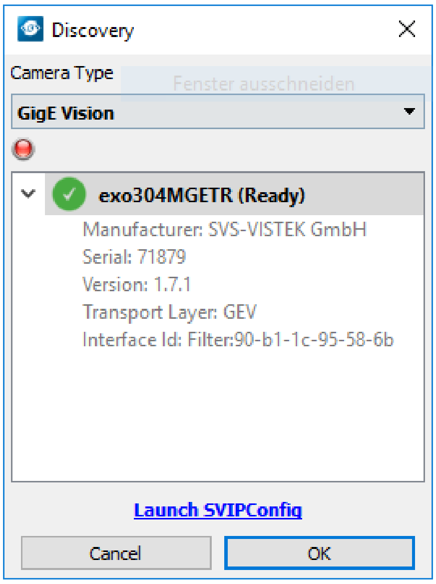
The CTI file is the file providing the GenTL transport layer, necessary for other programs if they want to operate with GenICam via GenTL.

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.

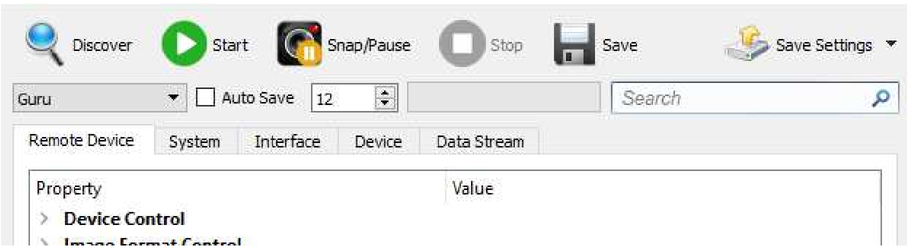


3.1 Connect your camera

To operate the camera in SVCapture, it is necessary to connect the camera to 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, 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 necessary items. Start will make the camera run, delivering images. You might save images 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.



3.2 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.

Despite SFNC (Standard Feature Naming Convention), the GenICam tree 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 differently.

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:

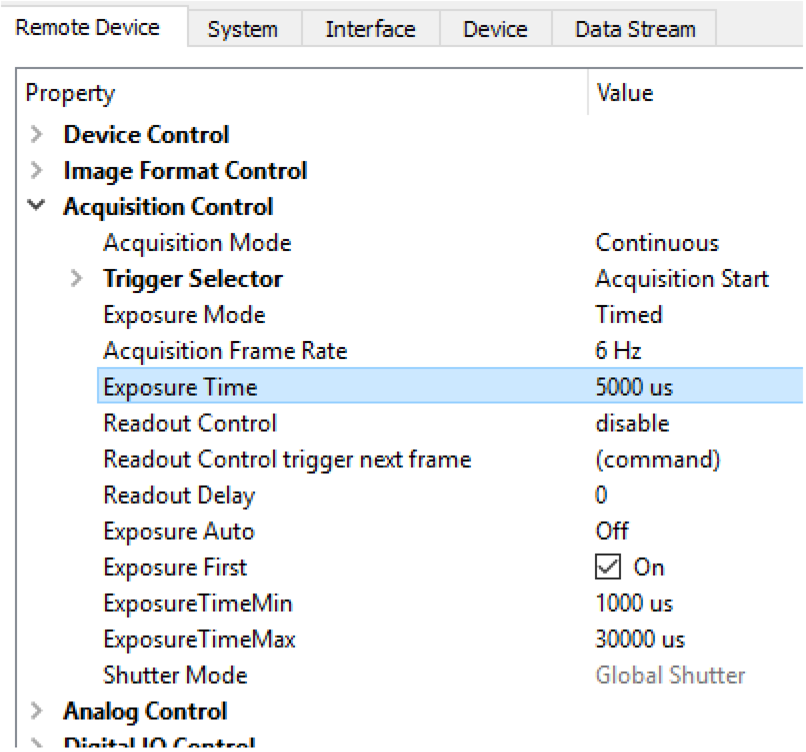
> Configure the camera
> Grab images
> Transmit extra data
> Deliver events

GenICam is organized in a tree structure. SVCapture displays the whole tree, grouped into functional blocks:

> 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
> 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.

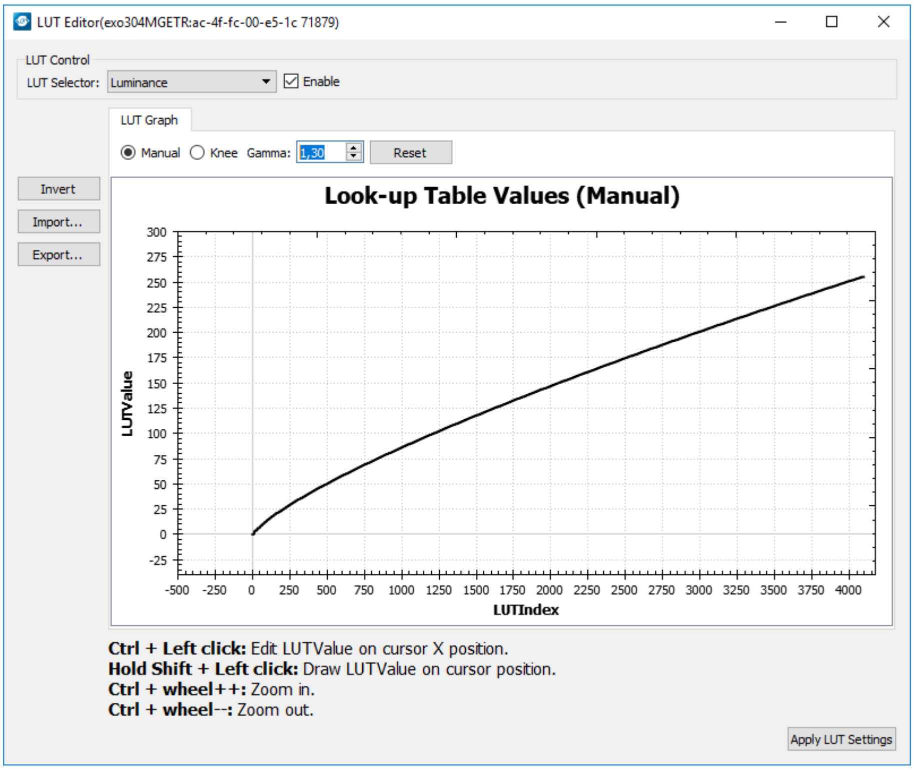
3.3 Assistants

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.

3.3.1 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

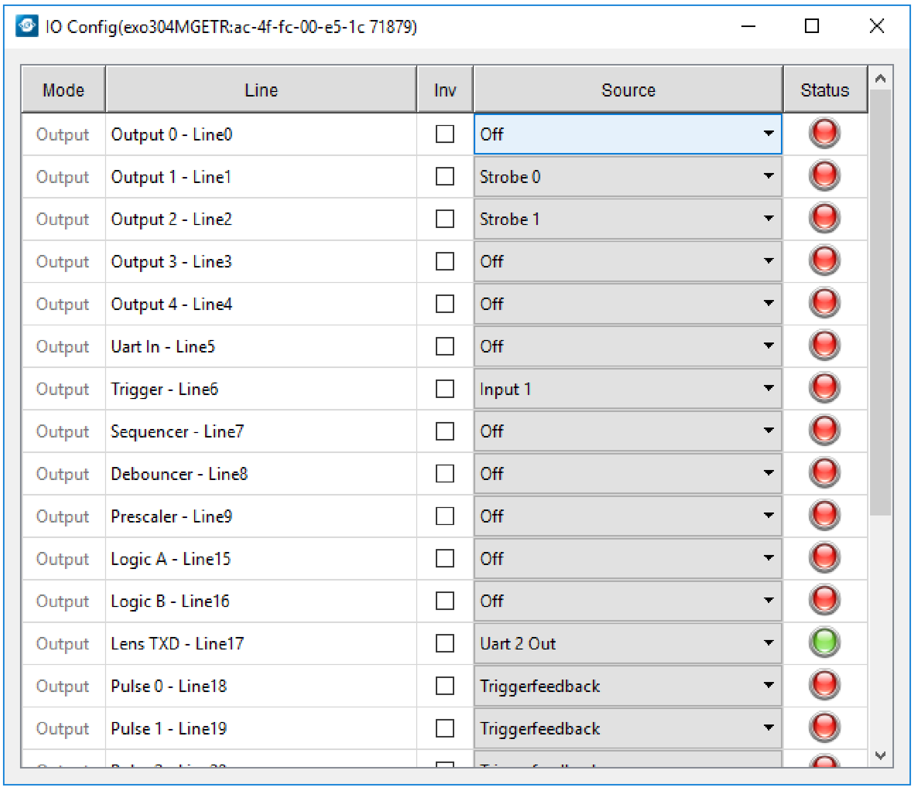
You can export and import your LUT as a file. It will be saved as XML file like this:

```xml
<?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.

3.3.2 IO Config

The IO Config assistant is the heart of your IO 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.

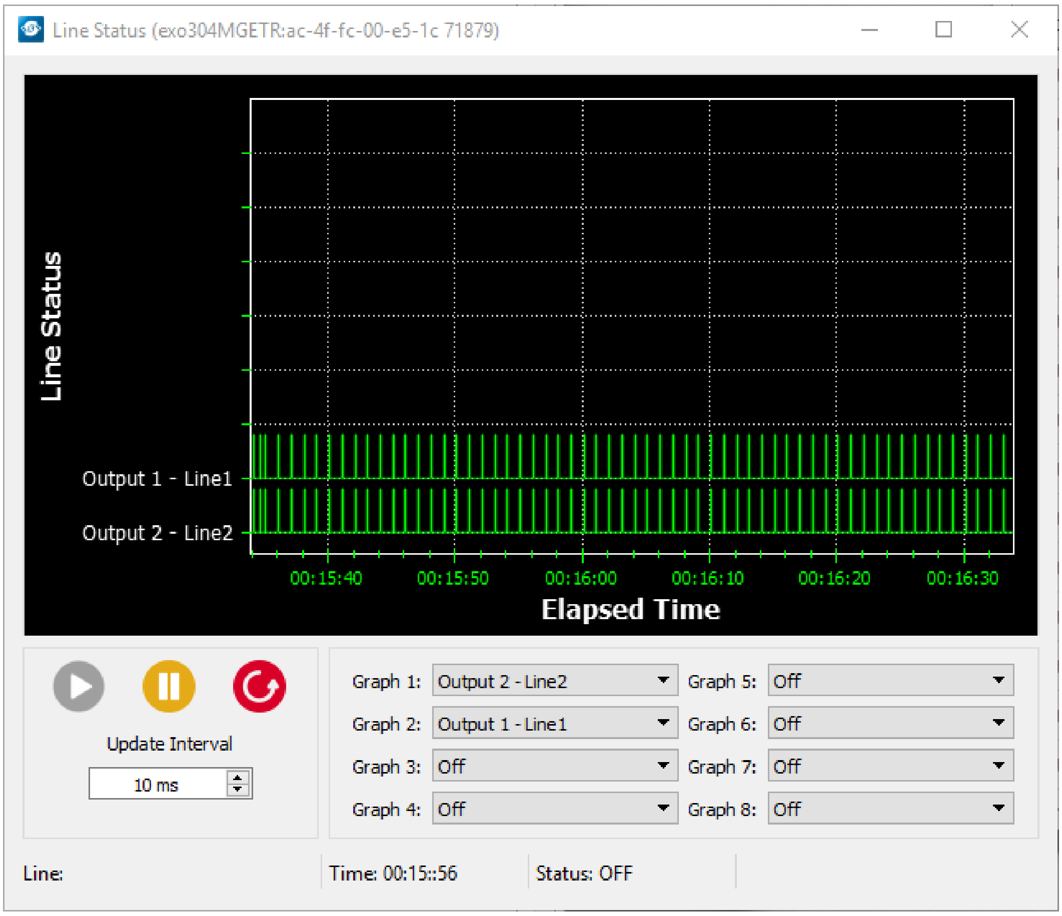


In the image above are – as an example – 2 strobe controllers connected to Output1 and Output2. On the left find you find the target module and you can select the input signal source on the right side for each module. “Inv” will will invert your signal (use this for testing).

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)

3.3.3 Line status visualizer

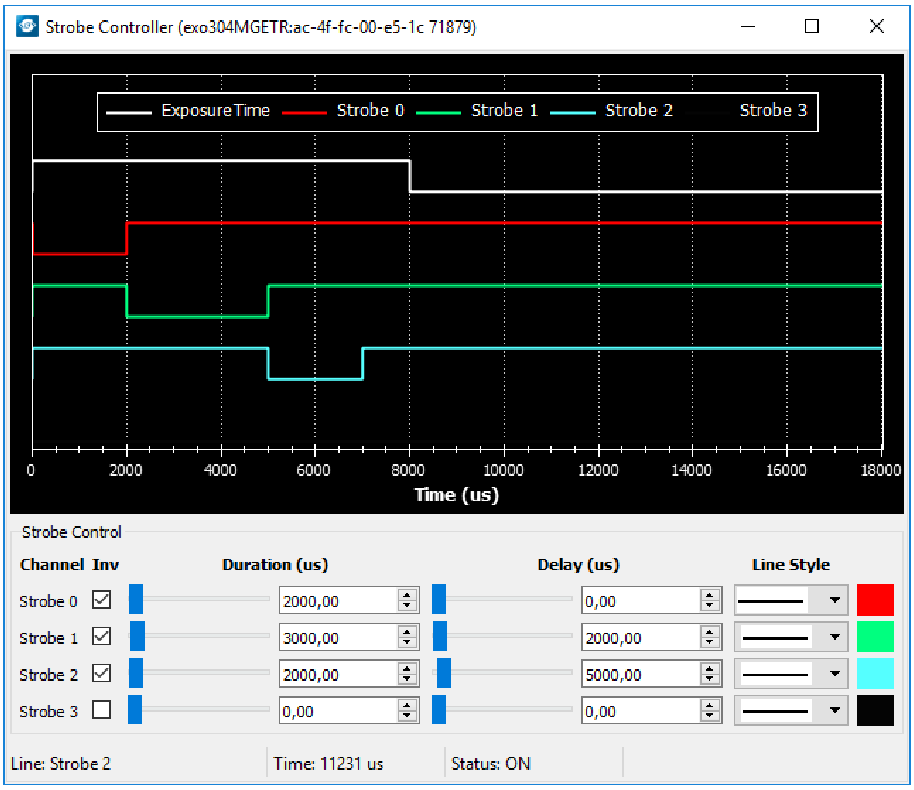
The line status visualizer shows you the activity state of 4I/O modules versus time. Most common is the display for strobe activity as shown below (line 1 and line 2 status). It is possible to see different modules in a time correlation (e.g. trigger/debounce/exposure/flash)



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

3.3.4 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.

3.3.5 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

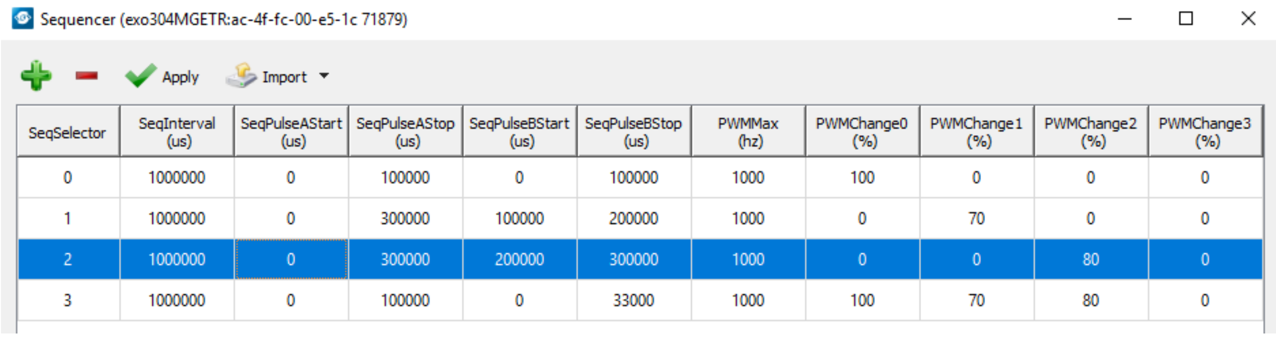
Notes

- 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	-	-	-	-

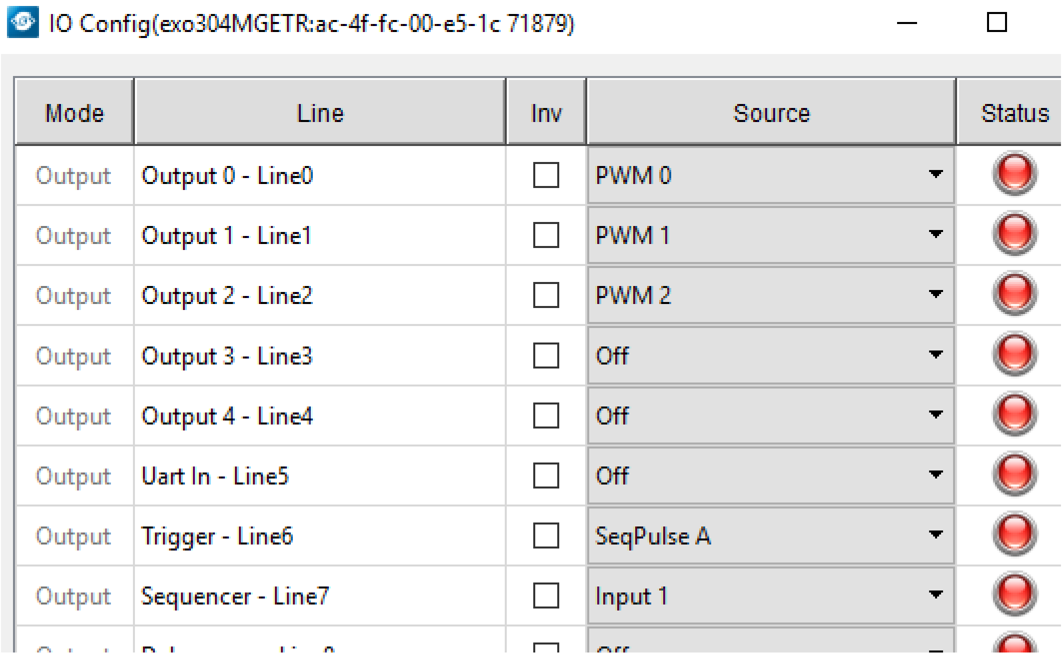
These values put into the sequencer would result in the sequencer assistant like this:



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

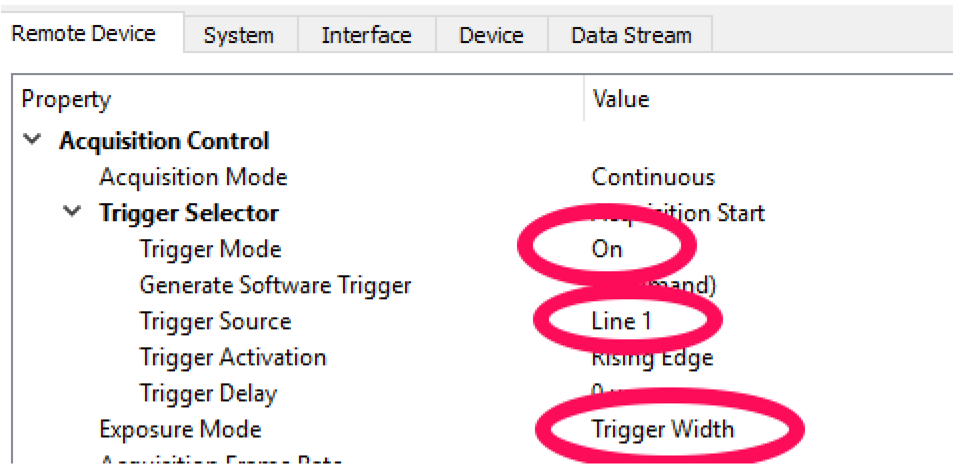
SeqSelector = Sequence counter (max 16)
Exposure time = (seq pulse A stop – seq pulse A start)
Strobe time = (seq pulse B stop – seq pulse B start)
PWMMax = PWM output frequency
PWMChange = Strobe intensity in %

Now, after applying this sequence configuration the configuration is downloaded to the camera, but it is still not active. Activate the sequencer like this:



If you want to use the strobe controller, verify your PWM outputs are connected to your outputs with your IO assistant. The connected outputs are sourced to their PWMs. 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. Up till now, the sequencer hasn't been activated, though.

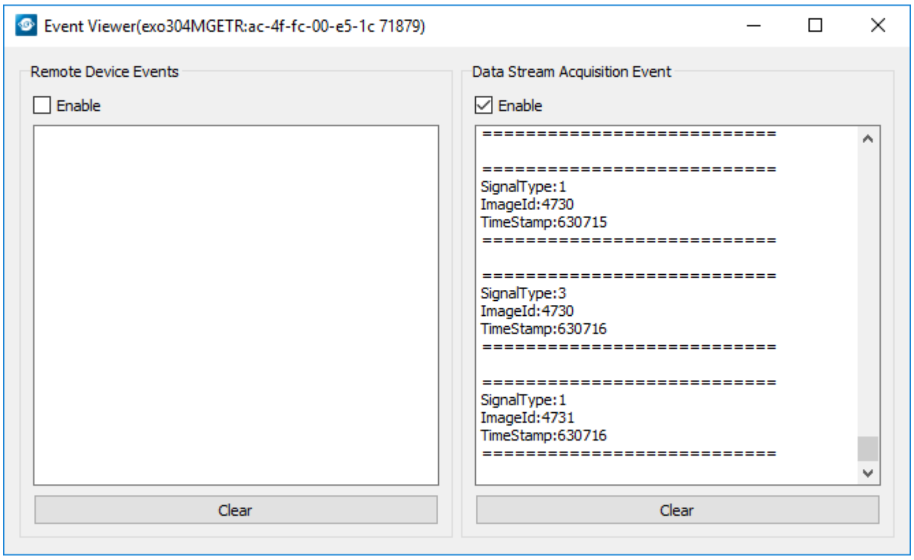
For sequencer activation, the trigger source has to be set to sequencer like this in the GenICam tree:



If you want to run the sequence, use the sequencer trigger command from the GenICam tree:

3.3.6 Event viewer

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



3.4 Common procedures

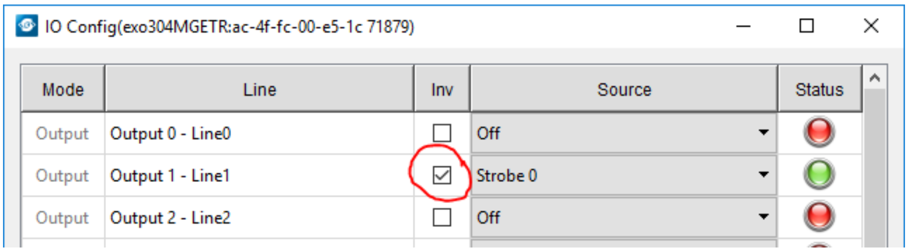
This chapter provides you common procedures for standard tasks, based on SVCapture 2.5.2 assistants. All these tasks can be done as well directly in the GenICam tree without assistants or programmatically with the SDK development files by modifying the GenICam tree.

3.4.1 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:

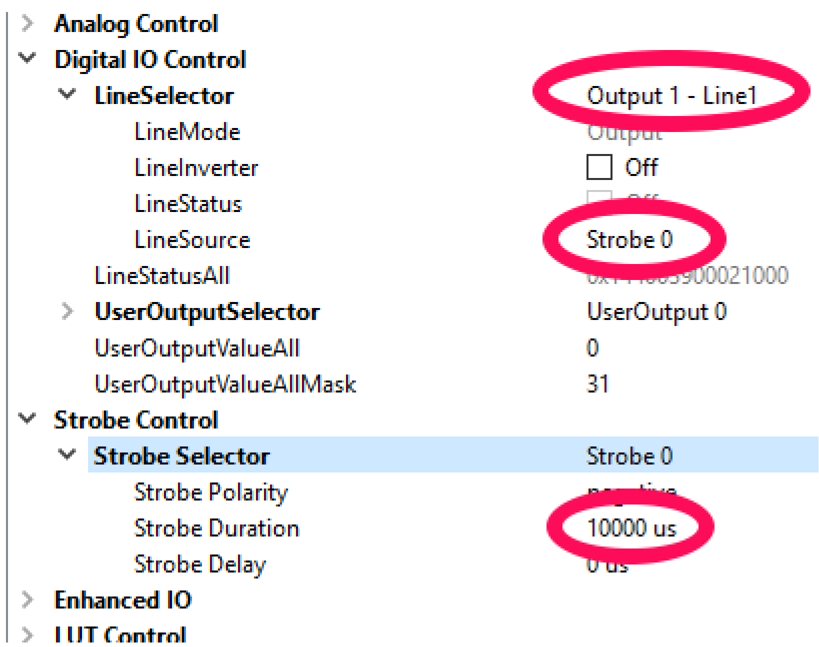
> **WARNING**
>
> For light protection, make sure electrical setup is with correct shunt resistor. See appendix 4.1

Connect your 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.



3.4.2 Setup single strobe light

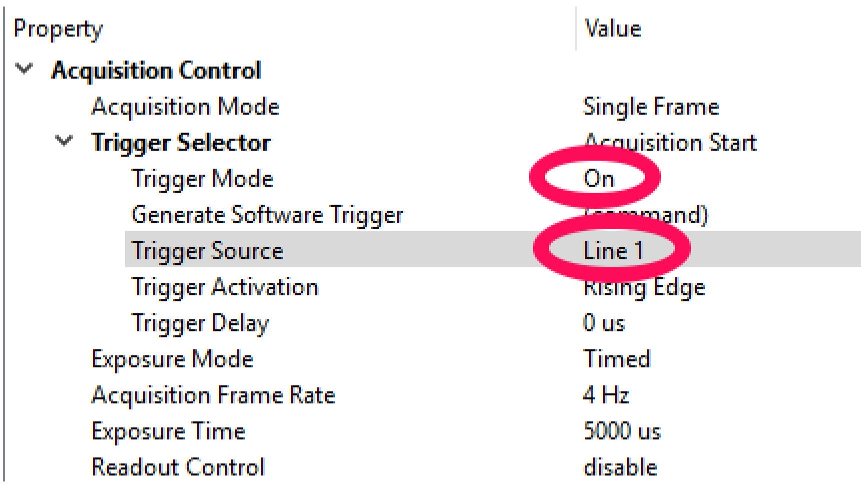
In this example, suppose your LED light is connected to OUT1 pin, please see chapter 4.1. In the GenICam tree, first set strobe0 as source for our OUT1 pin of the Hirose (“output 1”) connector. Specify necessary strobe duration.



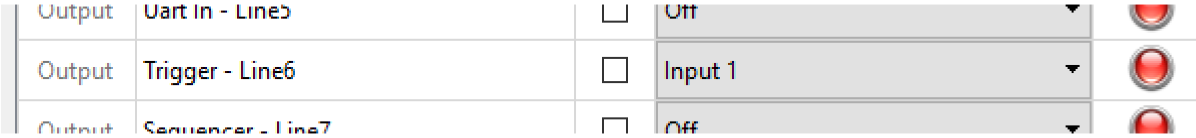
3.4.3 Set input to physical trigger input

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)



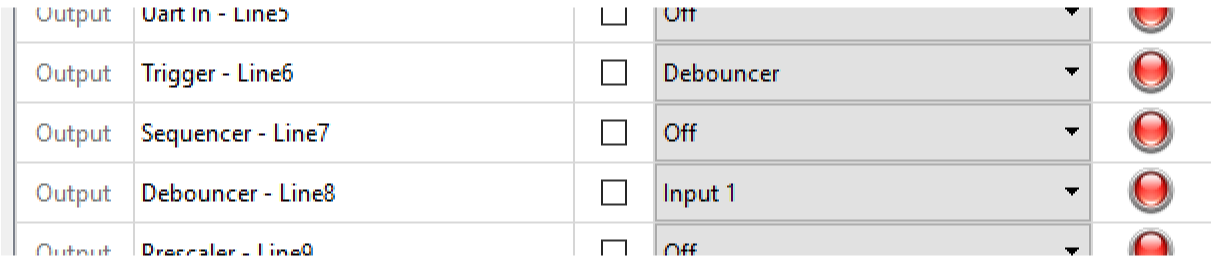
Connect the hardware INPUT1 line to trigger signal



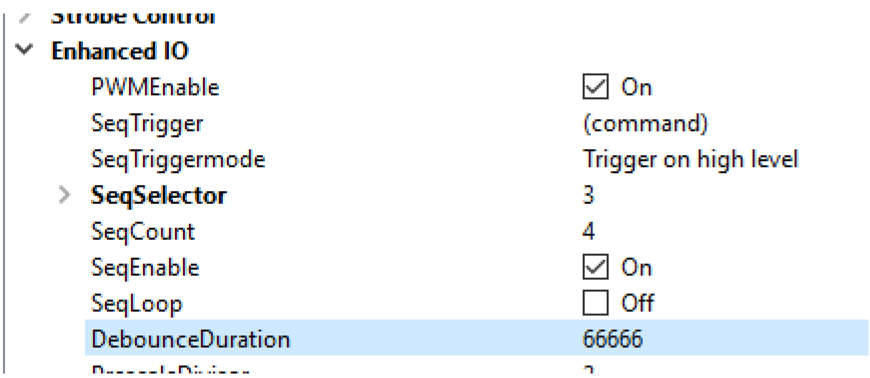
3.4.4 Debounce input trigger

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 IO 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

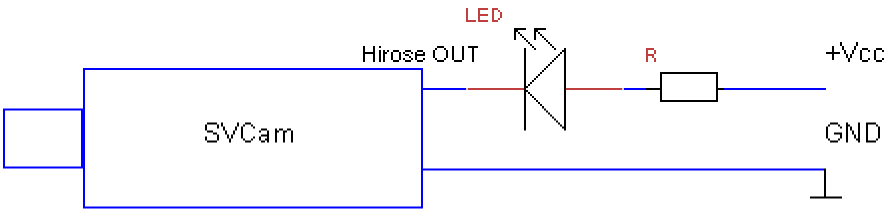
$$debounce\ duration = 15ns\ x\ 66666\ tics = 1ms$$

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).