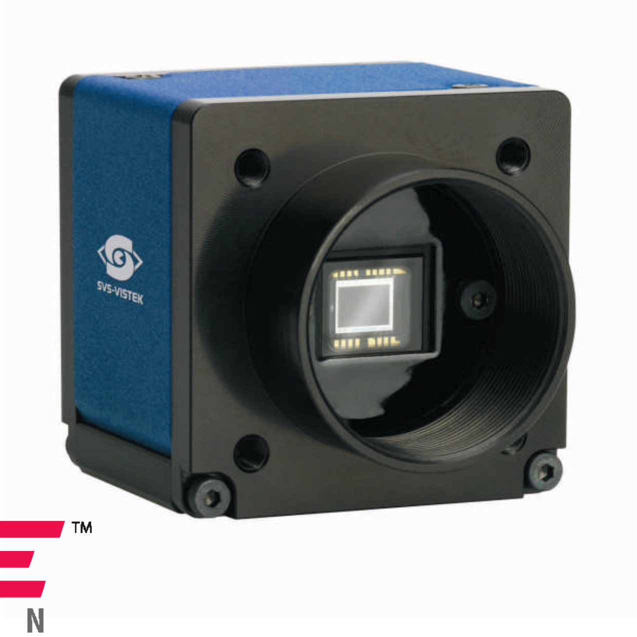
USERS MANUAL

SVCam-ECO Line, eco618, eco424, eco414, eco415, eco204, eco445, eco267, eco285, eco274 and eco655

Gigabit Ethernet Cameras



Digital Gigabit Ethernet Area-Scan Cameras

Version 2.0
Last Update 21.01.2011

Contact:

SVS-VISTEK GmbH

Mühlbachstraße 20
82229 Seefeld/Obb.
Germany

Tel. +49-(0)8152-99 85-0
www.svs-vistek.com

Service contact:
E-Mail: service@svs-vistek.com

Content

1 Introduction

Thank you for purchasing a SVS-VISTEK product. SVS-VISTEK stands for reliable competence and customer oriented solutions in the field of professional machine vision systems. We believe that we are more than just a trade partner of machine vision components and we believe in sharing our years of development expertise with our customers. We offer single source solutions and are a one-stop shopping center for all questions concerning professional machine vision. As a consequence, our insistence on the highest quality has resulted in many companies worldwide that use our products and services with great success and satisfaction. Customers at home and abroad trust and rely on our know-how and experience.
The following types are covered in this manual:

eco424 1/3" CCD, 640 x480 pixel, 12 Bit gray level resolution, ca. 124 fps; monochrome and color version

eco 414 1/2" CCD, 640 x 480 pixel, 12 Bit gray level resolution, ca. 125 fps; monochrome and color version

eco 415 1/2" CCD, 782 x 494 pixel, 12 Bit gray level resolution, ca. 86 fps; monochrome and color version

eco 618 1/4" CCD, 640 x 480 pixel, 12 Bit gray level resolution, ca. 160 fps; for IR applications

eco204 1/3"lCCD, 1024 x 768 pixel, 12 Bit gray level resolution, ca. 47 fps; monochrome and color version

eco445 1/3" CCD, 1280 x 960 pixel, 12 Bit gray level resolution, ca. 30fps; monochrome and color version

eco267 1/2" CCD, 1360 x 1024 pixel, 12 Bit gray level resolution, ca. 25 fps; monochrome and color version

svs285 2/3" CCD, 1360 x 1024 Pixel, 10/12 Bit gray level resolution, approx. 34 frames/sec, monochrome and color version

eco274 1/2" CCD, 1600 x 1200 pixel, 12 Bit gray level resolution, ca. 26 fps; monochrome and color version .

svs655 2/3" CCD, 2456 x 2048 Pixel, 12 Bit gray level resolution, approx. 10 frames/sec, monochrome and color version

For standard applications only 8 Bit are transmitted due to data load.
For more information on the color versions, please see chapter 12.6.

If you need modifications, we will be glad to offer you a custom camera suitable to your application.

2 Pinout and Installation

Warning:

⚠ The CCD camera is built with CMOS-LSI circuits. All internal electronics in the camera are sensitive to high voltage or electrostatic discharge. The camera can be destroyed if carelessly handled, so extreme care should be taken during set up and operation.

Do not expose the sensor to a direct laser beam as this could damage the sensor! **See Safety Instructions at Appendix H. Warranty will be void if not followed.**

2.1 Unpacking

- Camera
- Power supply (if ordered/option)
- Mating connector
- User Manual
- Disk with SDK including "Filter Driver", API and GUI "**SVCapture**" (for GigE camera) program. **Also a Firmware update tool.**
- XML File according to **GenIcam** standard released by AIA committee.

2.2 Power supply

Current consumption:
= 300 mA **(typical)**

Peak current on "Power on" up to 500 mA!
Current consumption increases rapidly when "partial scan" feature is used!

⚠

2.3 Connectors

2.3.1 One common signal connector for:

- **Trigger and TTL input to trigger camera**
- **General purpose I/Os**
- **UART communication**
- **Strobe Output (e.g. light)**
- **Power**

HR10A-10R-12PB (mating connector HR10A-10P-12S)

1 VIN- (GND)
2 VIN+ (10 to 25DC)
3 RXD data to camera (RS232 Level)
4 TXD data from camera (RS232 Level)
5 IN1 (0- 24 V) digital
6 IN2 (0- 24 V) digital
7 OUT1 (open drain max 24 V, 0.3 A) digital
8 OUT2 (open drain max 24 V, 0.3 A) digital
9 IN3+ (RS422 Level)
10 IN3- (RS422 Level)
11 OUT3 + (RS422 Level)
12 OUT3- (RS422 Level)

See outline in chapter 12.3

2.3.2 “Ethernet” Connector

RJ 45 “Western” Connector complies with Autosensing 100 T Ethernet and Gigabit specification. Features Auto MDIX.

1 LED for status indication

2.4 Installation/Getting started: Recommended PC

It is recommended to use a PC with a Pentium P4 processor at **2,4 GHz or higher**. The camera is working also on lower frequencies but it might not deliver the full frame rate in those cases.

If the camera is connected to the PC directly without using a network switch, **a fixed IP-Address and Subnet-Mask should be configured in the PC’s TCP/IP settings unless it is done by DHCP:**

See: Start -> Settings -> Network connections -> LAN-connection -> Properties -> TCP/IP

A Gigabit Ethernet network adapter is needed (100 MBit adapters would also work, but with reduced frame rates). If your PC does not have a Gigabit interface card purchase a card **using an original INTEL (TM) Chip set.**

⚠ **Currently PCI(e) bus cards with 82541 chip set version have been tested successfully. Currently we can recommend Intel pro 1000 PT,GT or MT. Recently problems with the "CT" version have been reported.**

DO NOT USE INTEL PRO 1000 CT !!

If you do not follow this, a significantly reduced data transmission rate is observed. Also the loss of data can appear.

2.5 Hardware

- Install camera in the desired location. You can use the ¼'' tripot adapter or use the 4 M3 holes in the adapter of the camera.
- Connect the power supply. If you have ordered a P/S connect it to the camera. If you use your own power supply (e.g. 10 -25 V DC) make sure you connect it to Pin1(GND) and Pin 2 (e.g. +12 -24 V).

If you – by mistake – supply line with GND pin the camera can be damaged! If power supply was ordered, do not modify it.

Connect an Ethernet cable to your PC or a network switch like you would build up a PC Network. Please note that cable length should not exceed 100 m for theses camera (Cat 5E version). In doubt consult your local distributor.

Wait 15 seconds until the green LED at the RJ45 connector of the Camera blinks sometimes.

Then start "**SVCapture.exe".**

2.6 Available operation modes

2.6.1 General

All modes are set by the interface which connects via **Ethernet standard cable**.

Minimum CAT 5E quality is required.

The default factory setting is **Free Running/ Fixed frequency** using the internal logic for exposure control. You can also trigger the camera by hardware and by PC (software trigger).

The color version will come on request with free software algorithm which will allow processing of the color image inside the PC or the camera. This must to be done in order to interpolate the colors for each pixel. Please note that no responsibility can be taken for the algorithm. It might be necessary to change the algorithm according to the application.

See chapter **12.6**

2.6.2 Operation Modes

- **Free Running**

- **Triggered, external exposure control**

If you want to trigger the camera and determine the exposure time by the pulse width of Exsync, choose this mode. Then apply a trigger signal at the appropriate pins e.g. on pins of Phoenix connector (see 2.3.1). With the starting edge of the pulse the camera will start exposure time. The exposure time ends with next upcoming edge of Exsync. Please check the timing diagram in the appendix of this manual.

- **External triggered, internal camera exposure control**

If you want to trigger the camera and use the convenient exposure time control of the micro controller then use this mode. You still can use the **SVCapture** GUI for setting exposure time (see "Exposure time"-field below). However you need to use the Exsync signal in order to trigger the camera (see "Trigger"-buttons below).

- **External triggered via Software trigger with internal exposure control**

If you want to trigger the camera with a software trigger and use the SVCapture exposure time control of camera logic then use this mode.

2.7 LED Signals SVCam-ECO

Camera status

Signals:

- No connection with network cable (yellow slow)
- Allotment of the network address (yellow quick)
- Network address allotted (yellow)
- Connected with application (green)
- Streaming channel available (green slow)
- Acquisition enabled (green quick)
- Problem with initialization (red slow)
- Overheating (red quick)

Optional instead of "acquisition enabled":

- Waiting for trigger (blue)
- Exposure active (cyan)
- Readout/FVAL (violet)

Blink codes

- On
- Slow (approx. 1 Hz)
- Quickl (approx. 8 Hz)

3 Software

3.1 SVCapture/GigE Vision Camera Viewer

Overview

You can use the SVCapture without the "Filter Driver" but if you want to lower the CPU load when grabbing images make sure it is installed.

⚠ **You will observe a loss of frames if you do not use the driver.**

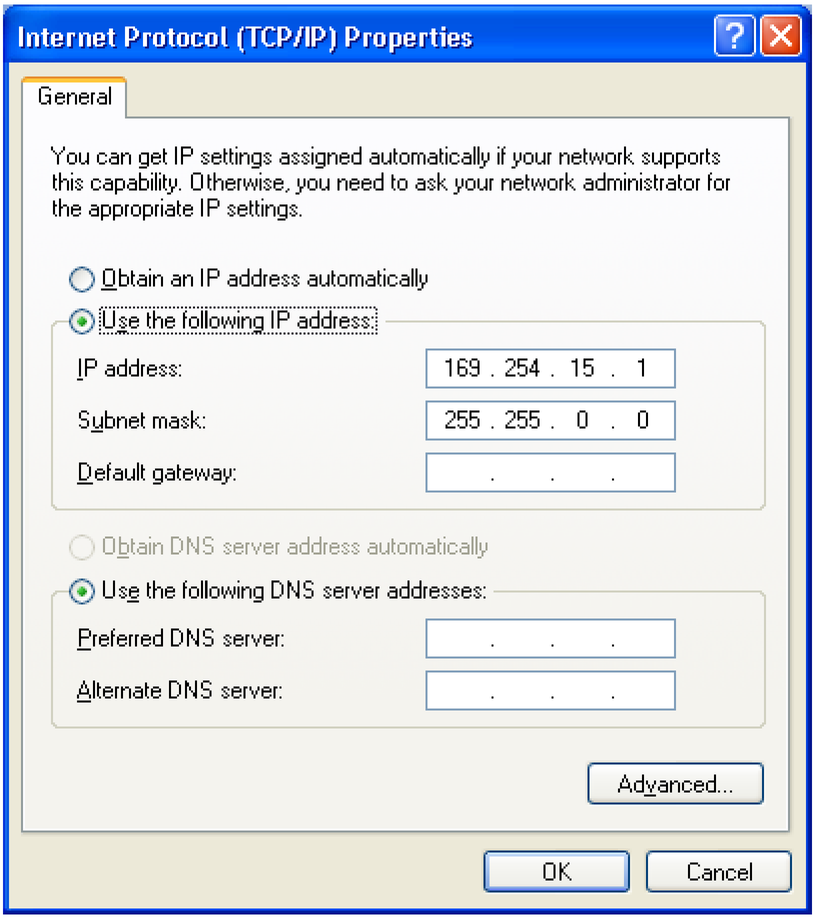
If you have installed the camera and connected the power, you can install the driver and the GUI to do the first test. Copy the "**SVCapture.exe**" to your disk drive, double click on the SVCapture icon and a window will open.

The software SVCapture displays images sent from SVS-GigE-Cameras via Gigabit Ethernet to your PC (with Windows XP) and allows adjusting basic camera settings.

3.2 Adjusting TCP/IP respective Network Settings

If the live image is not displayed, this is most often caused by inappropriate network settings. In this case the camera’s network settings have to be adjusted first before getting it into an operational mode.

Adjust settings accordingly
Settings -> Network connections -> LAN-connection -> Properties -> TCP/IP



You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.

○ Obtain an IP address automatically
● Use the following IP address:

IP address: 169 . 254 . 15 . 1
Subnet mask: 255 . 255 . 0 . 0
Default gateway: . . .

○ Obtain DNS server address automatically
● Use the following DNS server addresses:

Preferred DNS server: . . .
Alternate DNS server: . . .

In a network with a DHCP server the camera will obtain its IP address automatically and the settings shown in the screenshot above will be on “Obtain an IP address automatically.

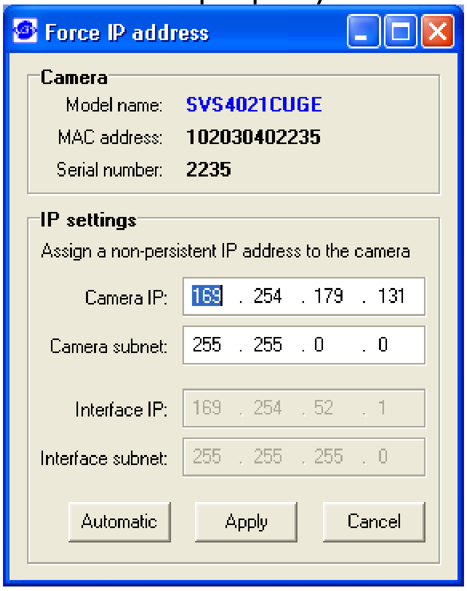
3.2.1 Network address assignment

The main components of a network address are the IP (Internet Protocol) address and the network mask. The usually applied network mask "255.255.255.0" for small networks up to 254 PCs represents a 32-bit long bit mask where the first 24 bits are set to 1 and the last 8 bits are set to 0. This network mask instructs the network hardware to let those devices exchange information with each other where the first 24 bits of the IP addresses match for all devices.

Thus the variable range of addresses is made from all possible combinations of the last 8 bits for which the network mask is set to 0. Exceptions are the first address 0 and the last address 255 which have special meanings for network management functions. All other 254 combinations are usually free for assigning them to network devices.

For a peer-to-peer connection of a GigE camera to a PC a network address assignment based on LLA (Local Link Address) is recommended. This involves a network mask "255.255.0.0" as well as a fixed first part "169.254.xxx.xxx" of the network address range. A GigE camera will fall back to LLA soon after recognizing that no DHCP server is available and that no fixed network address was assigned to the camera.

In case a camera can not be reached by the SVCapture application a dialog will appear that allows for adjusting a camera's network parameters. The simplest way is to click on "Automatic" which will provide to a search for a free IP address and to assigning it to the camera given that the interface IP could be determined properly.



Force IP address

Camera

Model name: SVS4021CUGE

MAC address: 102030402235

Serial number: 2235

IP settings

Assign a non-persistent IP address to the camera

Camera IP: 169 . 254 . 179 . 131

Camera subnet: 255 . 255 . 0 . 0

Interface IP: 169 . 254 . 52 . 1

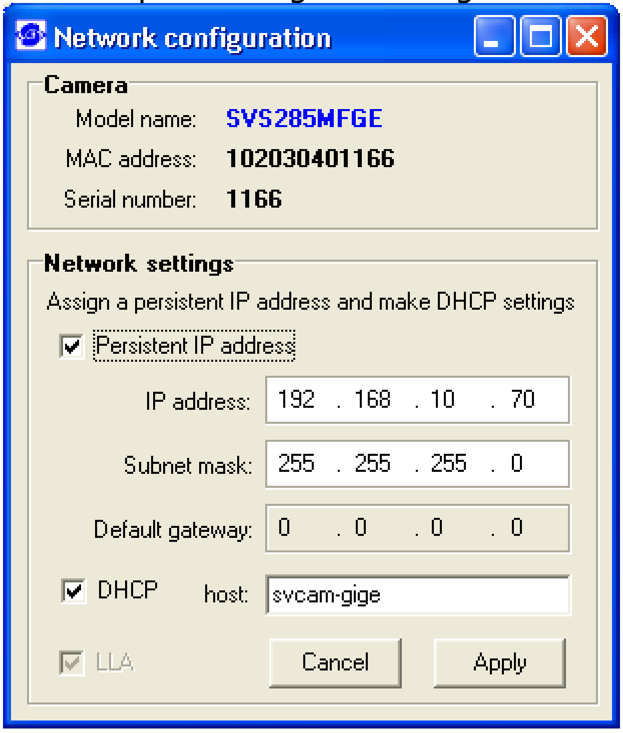
Interface subnet: 255 . 255 . 255 . 0

That dialog can also be brought up when right clicking on an entry in the discover dialog and subsequently selecting "Force IP address" in the context menu. However, when adjusting the PC network settings to LLA the camera must not get assigned a fixed network address

NOTE: The network settings performed in the "Force IP address" dialog are only valid until the next shutdown and restart of a camera. For permanently changing a camera's network settings the "Network settings" dialog can be used instead. That dialog can also be opened in the context menu which is displayed when right-clicking on an entry in the discover dialog.

3.2.2 Persistent Network address assignment

Open the dialog by selecting "Network settings" in the context menu that comes up when right-clicking on an entry in the discovery dialog:



Network configuration

Camera

Model name: SVS285MFGE

MAC address: 102030401166

Serial number: 1166

Network settings

Assign a persistent IP address and make DHCP settings

☑ Persistent IP address

IP address: 192 . 168 . 10 . 70

Subnet mask: 255 . 255 . 255 . 0

Default gateway: 0 . 0 . 0 . 0

☑ DHCP host: svcam-gige

☑ LLA

All settings in the "Network configuration" dialog will be transferred to the camera and they will be made persistent on the EEPROM inside the camera. Therefore this dialog allows for assigning a non-volatile network address to a camera which it will maintain also over shutdown and restart cycles.
Network settings performed in this dialog become valid but after a reboot of the camera.
In addition to a persistent network address it can be decided whether the camera will try to obtain a dynamic network address from a DHCP server on start-up. If this option is used the camera can no longer be identified uniquely by IP address. Instead, other items like MAC address, serial number or the user defined name can be used for this purpose. Usually the DHCP method is convenient for involving a camera in to an already existing network of computers which all obtain their IP addresses from a DHCP server. The camera will well behave in this environment when adjusting it also to the DHCP mode.

In some cases it may be that a camera will be connected for performance reasons directly to a dedicated network card. Given that this network card has not got assigned a valid IP address there is **a fall-back procedure** defined which takes automatically a network address from a special range that has been assigned for this purpose. This is the so called **"Link Local Address (LLA)"** behavior that constitutes the last item in a chain of fallback cases. It has been defined in the **GigE Vision standard document** that the LLA fall-back procedure can never be deactivated. Therefore the LLA checkbox is deactivated but always in a checked state.

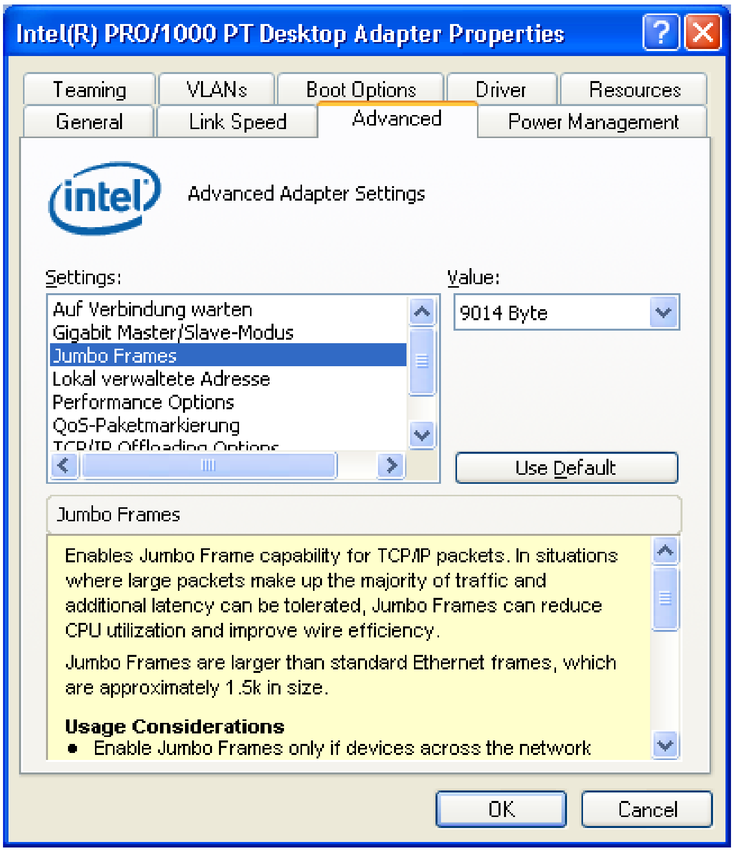
Currently the Default gateway setting is not used and therefore deactivated. This is related to performance considerations which recommend to operate

GigE cameras in the same network segment where the clients exist and not to route the data stream in to other networks.

3.2.3 Using jumbo frames

The transport efficiency in the streaming channel can be improved when using “jumbo frames” in network transport. This will reduce the overhead which is caused by maintaining header data with each data packet. A network packet has usually a size of about 1500 bytes which can be increased to e.g. 16112 by switching “jumbo frames” on. In addition to the network card “jumbo frames” have to be supported also by a switch that forwards the image data stream from a camera to the PC.

Adjusting higher packet sizes requires network cards that support jumbo packets, e.g. Intel PRO/1000 PT which offers a single network port or Intel PRO/1000 MT which offers two network ports. Other cards have to be checked whether they contain an adjustment which allows for switching “jumbo frames” on.



Settings:
Auf Verbindung warten
Gigabit Master/Slave-Modus
Jumbo Frames
Lokal verwaltete Adresse
Performance Options
QoS-Paketmarkierung
TCP/IP Offloading Options

Value: 9014 Byte

Jumbo Frames

Enables Jumbo Frame capability for TCP/IP packets. In situations where large packets make up the majority of traffic and additional latency can be tolerated, Jumbo Frames can reduce CPU utilization and improve wire efficiency.

Jumbo Frames are larger than standard Ethernet frames, which are approximately 1.5k in size.

Usage Considerations

- Enable Jumbo Frames only if devices across the network

NOTE: For Intel Pro/1000 cards the settings should be adjusted as follows:
Flow control: Generate
Interrupt throttling rate: Minimal
Jumbo frames: adjust to minimum 9014 byte
All SVCam-GigE cameras support flow control and jumbo frames upto 16112 bytes.

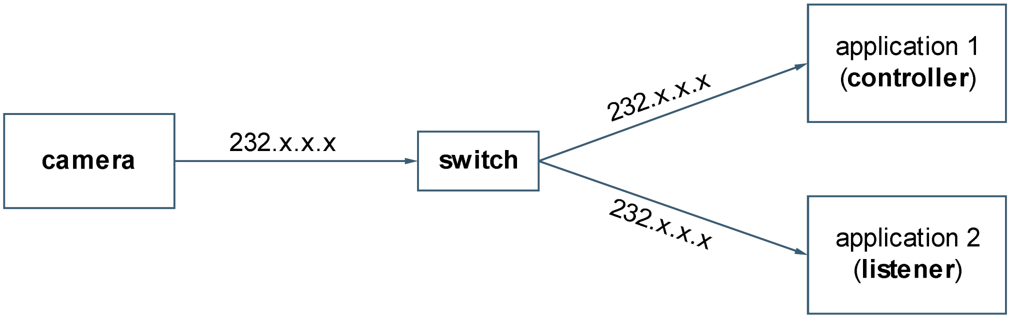
3.2.4 Performance considerations

Each GigE camera is a **high-performance device**. In order to use the full performance of a GigE camera a PC is required that meets at least the data transfer figures that the connected camera is capable to deliver to the application.

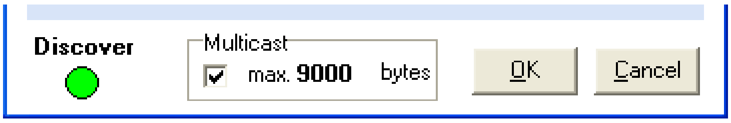
For example, a SVS625MFGE camera delivers a compound image data stream consisting of two times 50 Megabytes/sec. of payload data plus network overhead. Therefore the PC that operates that camera should be capable of transferring that amount of data as a net data stream over all its internal components like network card, PCI bus and others. A PC with a dual core Pentium D 3.0 GHz processor and an Intel PRO 1000 MT network card is an example of a well equipped system.
NOTE: All SVS GigE cameras can also be operated with computers of less performance than the camera delivers at highest frame rate. In this case the maximal available framerate will be below the camera's maximum. This is a valid operating mode, however one has to be aware of the limits.
HINT: **Notebooks are in most cases not capable to operate data streams at the maximum level that SVS GigE cameras deliver**. One has to be aware that connecting a SVS GigE camera to an average notebook will not allow for operating the camera at highest framerate but only on lower framerates.

3.2.5 Multicast

When images from a single camera are to be delivered to multiple PCs the usual way is to use multicast (RFC 2236). A switch receives an image data stream from a camera and distributes it to multiple destinations in this mode. Since a GigE camera always needs a single controlling application, there will be only one master application. That controlling master application has to open a camera in multicast mode (IP 232.x.x.x for local multicast groups) for allowing other applications to connect to the same image data stream. Other applications will become listeners to an existing image data stream. They will not get control access to the camera, however their eventual packet resend requests will be served in the same way as for the controlling application.



When using SVCapture as the controlling application, the "Multicast" checkbox has to be checked in the Device Discovery dialog before opening a camera.

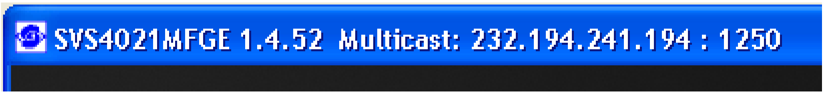

Discover ●

Multicast

☑ max. 9000 bytes

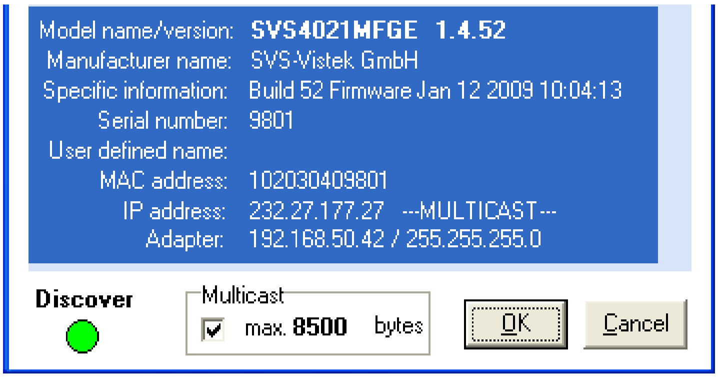
It is important to enter a suitable maximal packet size. This packet size is determined as the minimum packet size from all intended listeners. When for example all applications but one have jumbo frames adjusted to 16112 bytes and the one application has jumbo frames adjusted to 9000 bytes then the minimum (9000) has to be entered in the above shown dialog.

A multicast data stream can be identified in SVCapture by a change in a camera view's title. A multicast group (232.x.x.x) along with the used port will be shown instead of a camera's MAC that is usually displayed in that place.



SVS4021MFGE 1.4.52 Multicast: 232.194.241.194 : 1250

Further the camera is operated as usual in the controlling application without any changes to normal mode.
A listening application will see a hint "MULTICAST" in the "IP address" field of the discovery dialog for a camera that is streaming in multicast mode along with the IP address of the multicast group.



Model name/version: SVS4021MFGE 1.4.52
Manufacturer name: SVS-Vistek GmbH
Specific information: Build 52 Firmware Jan 12 2009 10:04:13
Serial number: 9801
User defined name:
MAC address: 102030409801
IP address: 232.27.177.27 ---MULTICAST---
Adapter: 192.168.50.42 / 255.255.255.0

Multicast
☑ max. 8500 bytes

The "Multicast" group box will display the packet size that applies to the running image data stream. **The listening application has to have at least that packet size available (jumbo frames) in order to properly connect to the image data stream**.
After selecting the multicast camera in the camera list of the discovery dialog the OK button can be clicked or the camera entry can be double clicked in order to connect to the already running image data stream.
Since the listening application does not have control access, all controls in the 'SVCam Settings' dialog will be disabled except the Close button.

4 Windows x64 support

The SVGigE SDK supports x64 natively by providing a SVGigE driver along with DLLs and a sample application which are compiled for x64 platforms.
Although the SVCapture.exe program is a Win32 application, it can be used with some limitations also on Windows x64 PCs using the win32 compatibility mode.
When running SVCapture on x64 platforms in the compatibility mode the SVGigE driver will not be available because a win32 application is not capable of handling the 64-bit image buffer addresses which are provided by the driver. Nevertheless SVCapture can use Winsock for both, communicating to a camera as well as streaming image data from the camera to the application.
NOTE: Though a Winsock connection to a camera can be established properly also on x64 platforms, there exist the same limitation for image streams regarding transport reliability also on that platform. Usually a certain amount

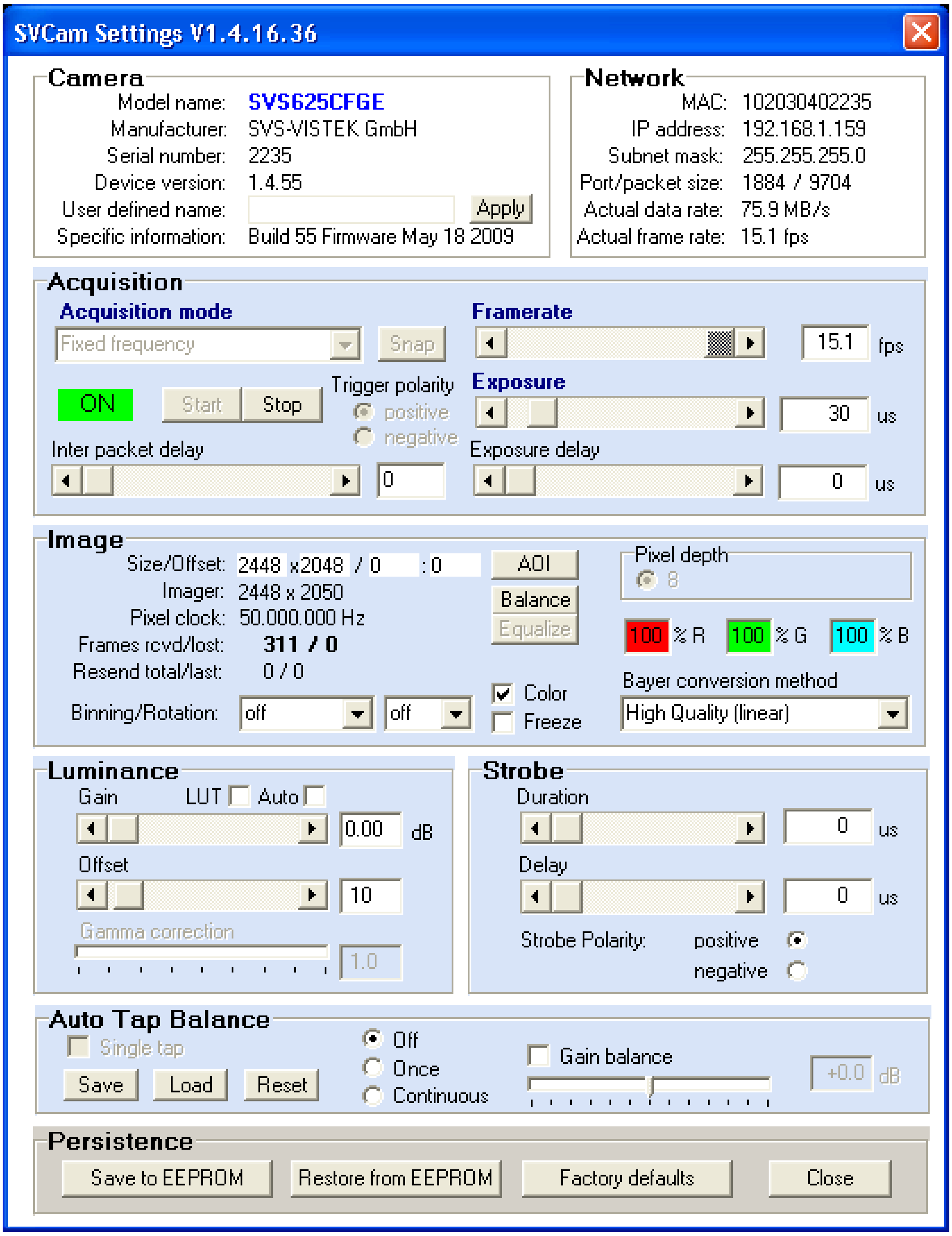
of network packets gets lost when working over Winsock. Though the GigE Vision standard allows for resending lost packets it might be that under certain circumstances visible failures in images can result from loosing network packets dependent on the system load. Therefore the recommended and most reliable way for streaming image data from a camera to a PC is by using the filter driver. The SVGigE_ExampleVC8.x64.exe allows for running a connection with both, filter driver as well as Winsock and for comparing those modes and their main characteristics against each other.

5 Camera settings dialog

Double clicking into the image area or selecting the "Camera settings" entry in the "Camera" menu brings up a settings dialog that allows for accessing the various camera settings:

Note that SVcapture is currently not working under Win XP 64 bit or Win 7 64 bit OS.

However, the sample program and driver work well . So it will not be a problem for a developement.



SVCam Settings V1.4.16.36

Camera

Model name: SVS625CFGE
Manufacturer: SVS-VISTEK GmbH
Serial number: 2235
Device version: 1.4.55
User defined name:
Specific information: Build 55 Firmware May 18 2009

Network

MAC: 102030402235
IP address: 192.168.1.159
Subnet mask: 255.255.255.0
Port/packet size: 1884 / 9704
Actual data rate: 75.9 MB/s
Actual frame rate: 15.1 fps

Acquisition

Acquisition mode: Fixed frequency
Trigger polarity: ● positive ○ negative
Inter packet delay: 0
Framerate: 15.1 fps
Exposure: 30 us
Exposure delay: 0 us

Image

Size/Offset: 2448 x2048 / 0 : 0
Imager: 2448 x 2050
Pixel clock: 50.000.000 Hz
Frames rcvd/lost: 311 / 0
Resend total/last: 0 / 0
Binning/Rotation: off off
Pixel depth: ● 8
100 % R 100 % G 100 % B
☑ Color
☐ Freeze
Bayer conversion method: High Quality (linear)

Luminance

Gain LUT ☐ Auto ☐ 0.00 dB
Offset 10
Gamma correction 1.0

Strobe

Duration 0 us
Delay 0 us
Strobe Polarity: positive ● negative ○

Auto Tap Balance

☐ Single tap
● Off ○ Once ○ Continuous
☐ Gain balance +0.0 dB

Persistence

While adjusting settings in the dialog the new values are transferred to the camera continuously and the live image will respond to all changes immediately.
In particular the following information and settings are available:

In particular the following information and settings are available:

5.1.1 Camera

In the camera panel the following information is queried from the camera and displayed:

- Model name: **ecoZZZXVGE**
- Manufacturer name: **SVS-VISTEK**
- Serial number
- Device version
- User defined name
- Manufacturer specific information

5.1.2 Network

The network panel shows the currently used network related settings and parameters. These are in particular:

- The camera's MAC address
- The camera's IP address
- The camera's subnet mask
- The camera's streaming port and streaming packet size
- The actual data rate
- The actual frame rate

5.1.3 Acquisition

The acquisition panel contains the settings that are related to image acquisition as follows:

- Acquisition mode: (free running, software trigger, external trigger with internal/external exposure)

  - ➢ **Free Running**: At the first installation it should be this mode. In this mode the camera creates all sync signals itself. There is no need to trigger the camera (by EXSYNC) in order to get data. **The Exposure time can be set by using the software Interface of the PC.** It is controlled by the internal FPGA**.** No further external signals. The enclosed program allows the user to set the values from e.g. about 1/100.000/sec to 1 second. Exposure time can be changed online during operation.

  - ➢ **External trigger and using Pulse width of Trigger (external exposure):** In this mode the camera is waiting for an external trigger which causes the integration and read out. The exposure time can be varied by the length of EXSYNC between the high going edge and the low going edge. The Time settings in the control menu are not activated. This mode is useful in applications where the light level of the scene changes during operation. A frame to frame variation is allowed. Trigger must be fed directly to the camera by into the Hirose connector. A TTL **(min. 5V)** signal is provided from encoder, flashlight or any other source. Details see Appendix G.

- **Software triggered and using PC:** The frame rate is determined by the number of Software TRIGGER pulses generated inside the PC per time. With each "Software pulse" the camera will readout a frame. The Exposure time is set as in free running mode. Exposure time can be changed online during operation.

- Frame rate: It allows to alter the frame rate between 1 and the specified maximum value.
- Exposure: The exposure time can be set in µsec. The minimal exposure time is about 50 µsec (depending on the camera type and speed). The longest is about several seconds (triggered modes). Due to the internal timing of the camera the program will adjust the values to the appropriate values.
- Acquisition control (Start/Stop)
- Single capture button (Snap)
- Exposure (enter values > 2 sec. [2..85 sec.] into the Edit filed)
- Exposure delay
- Trigger polarity (positive/negative)
- Single capture button: Takes a single shot of the actual scene
- Inter-packet delay

The inter-packet delay has impact on a camera's bandwidth usage. A setting of an inter-packet delay of zero will send all image packets as fast as they are available dependent on the camera's pixel clock, e.g. at 50 MHz. This is the preferred setting when operating a single camera on a network interface.
In case of multiple cameras or other devices working on the same physical network it might be desirable to send the packets of a camera's streaming channel with a certain inter-packet delay in order to allow multiple cameras or devices to share a given network bandwidth.

NOTE: The inter-packet delay should be below the value which would decrease the frame rate.

5.2 Image

The image panel displays information about the picture geometry, the pixel clock and it allows for the following settings:

- AOI (area of interest) which can be less or equal the imager size
- Binning mode (off, vertical, horizontal, 2x2)
  No Binning = full resolution: horizontal x 1, vertical x 1 (default setting)
  H2 x V2 x: vertical 2 x and horizontal 2 x at the same time; Resolution:

horizontal x ½, vertical x ½, Sensitivity is 4 x, pixel frequency is halved, max. frame rate is almost doubled. "No Binning" sets the camera to full resolution, H1 x 2 and 2 x 1 binning is also available

If you need other binning mode configurations consult factory or your nearest distributor.
Please note: **Using binning with a color version of the camera will cause incorrect colors and strange effects!** However, for fast focusing it might be useful.

- Pixel depth (8-bit, 12-bit, 16-bit, if supported by camera)
- Color (On/Off)
- Bayer method (Disabled, Nearest neighbor, Simple, Bilinear, HQ Linear, Edge Sense, Gray)
- Factors for white balance (Red, Green, Blue)
- "Balance" button for performing automatic white balance
- "Equalize" button for setting all colors to 100%
- Freeze (display switched off while acquisition continuous, for test purposes)

In addition the number of transferred frames is displayed as well as the number of eventually lost frames. A frame loss may happen for example in case of an insufficient network bandwidth or if the network connection gets interrupted for a short time or in case of other network failures.
Further the number of resent network packets is displayed. The second number is the last resent packet number and the first number indicates the total number of resent packets.

5.3 Gain

The gain panel allows for adjusting gain and autogain (planned) with the following controls:

The default gain setting is "0" dB. You may change the gain up to 6 dB (or higher) in steps of 1/10th. Note that the dark offset will increase and dynamic range will not be improved. Please note that noise also is amplified. For good image quality do not increase gain more than 6 dB; **higher Gain is possible but not specified!**

5.4 Offset

Dark level offset adjustment is possible. When 8 Bit are transmitted it can be changed from 0 to 255. Please note that factory adjustment is optimized for S/N ratio and sensitivity. You may lose dynamic range. Alter only if you operate at high temperatures like +40 °C. It can be altered for each channel separately (for 2 tap Kodak CCDs).

5.5 AutoGain/AutoExposure

When the "Auto" checkbox is activated for cameras that provide for the AutoGain/AutoExposure feature, the "Luminance" panel changes to show the settings that apply to this mode of operation. The manual settings for gain and offset will disappear since the luminance will be controlled automatically.
The automated luminance control algorithm takes advantage of both, exposure and gain settings. First the exposure will be tried to adjust it such that a set

brightness value will be met. Once the highest exposure value has been reached the algorithm will further increase luminance by increasing camera's gain. The upper and lower limits for both controls are available in the "Luminance" panel once the "Auto" checkbox is checked.
The actual exposure is shown on the "Acquisition" panel and the actual gain is displayed in the "Luminance" panel on the left to the "min" and "max" gain settings.

6 Assigning IO lines

If a camera supports IO settings (HR and eco family) then the input and output lines can be arbitrarily assigned to actual data lines.

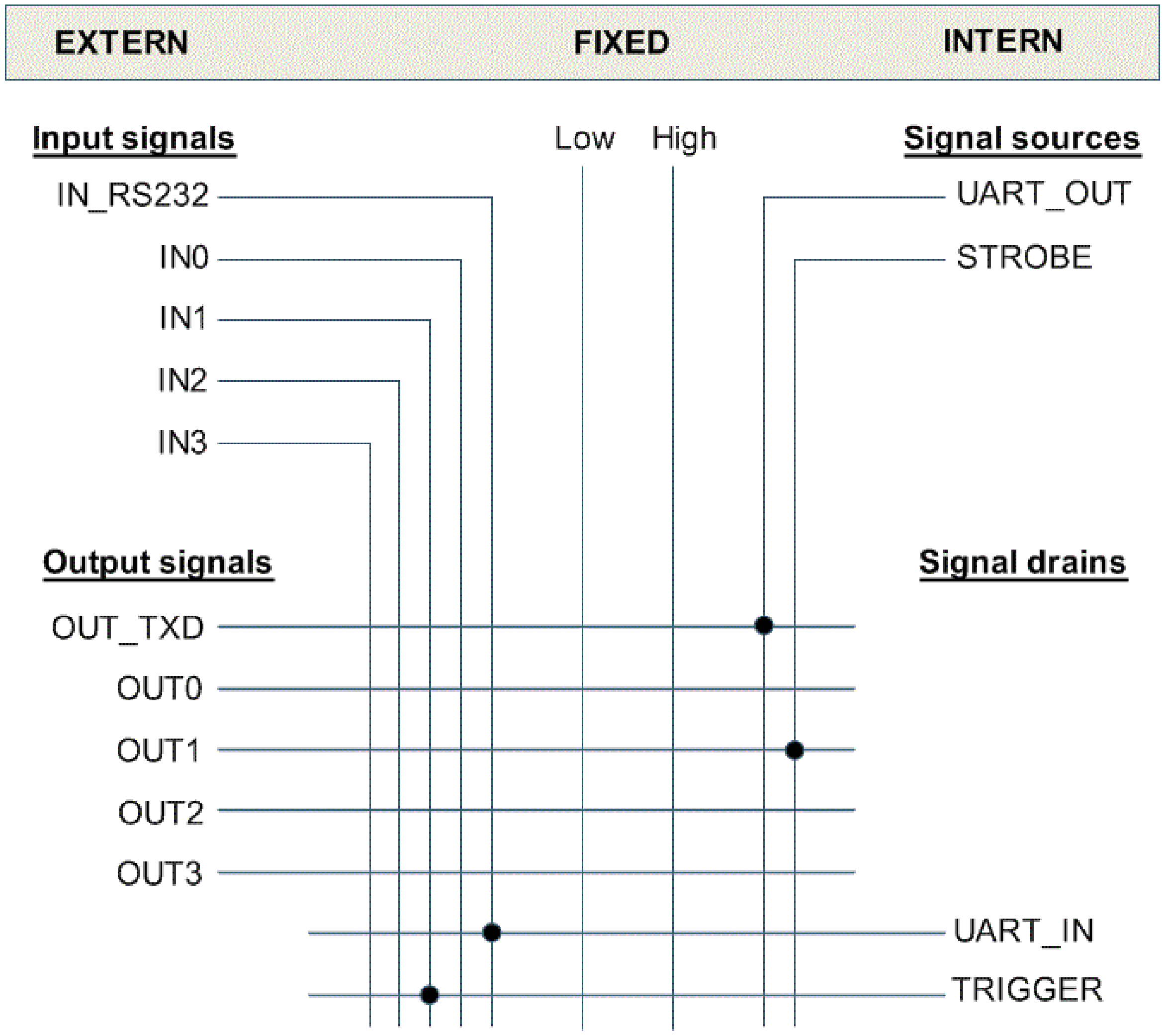


Chart: Flexible IO assignment matrix for SVS VISTEK cameras (with default settings)

The following outputs are available:

- OUT0 (only in HR family, not for eco family)
- OUT1
- OUT2
- OUT3
- OUT_TXD

Those output lines can be connected to the following signal sources:

- UART_OUT
- STROBE
- any of the input lines
- fixed signals Low or High

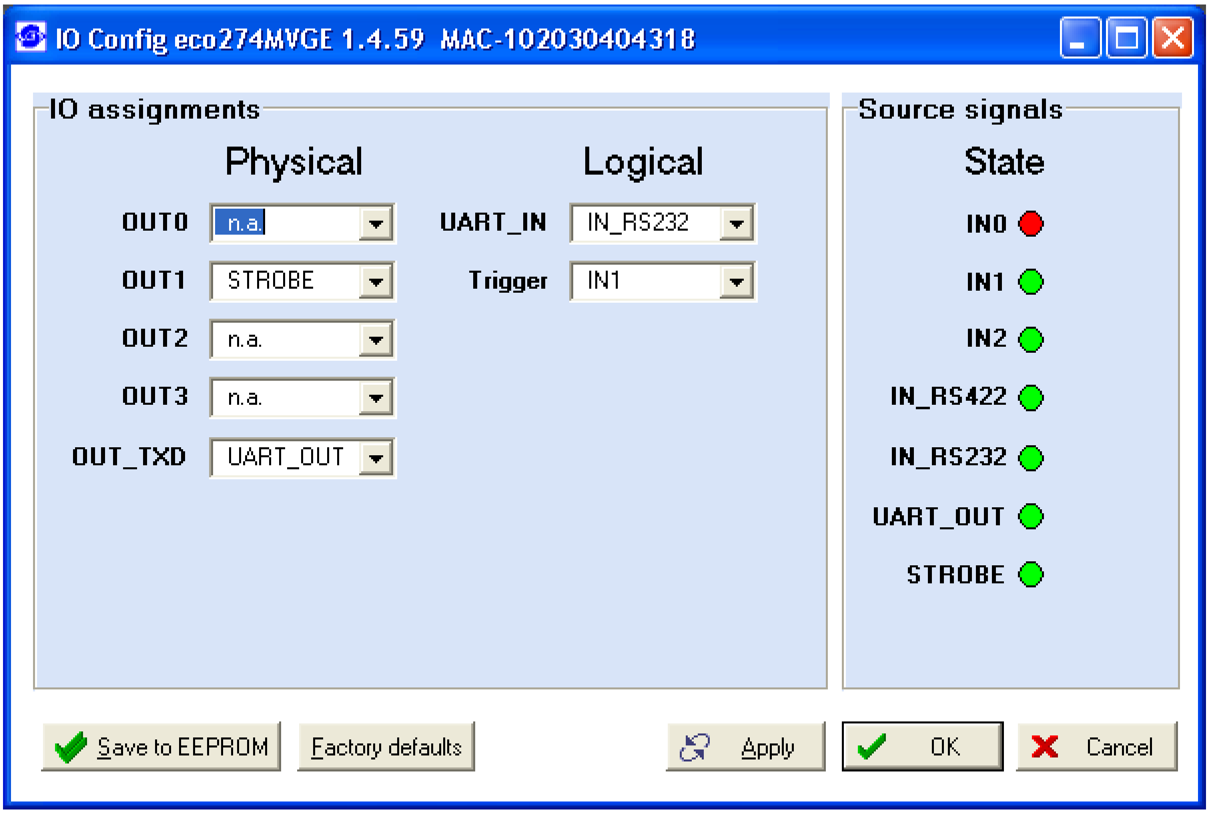
The following input lines are available:

- IN0 (only in HR family, not for eco family)
- IN1
- IN2
- IN3
- IN_RS232

The input lines can be connected to the following signal drains:

- UART_IN
- Trigger
- any of the output lines

The dialog allows for assigning signal sources to all physical output lines and for assigning physical inputs to logical input signals (signal drains).
When clicking on 'Apply' after making all adjustments then the new assignments take effect. One has to click on 'Save to EEPROM' in order to make them persistent. In case of any problems the 'Factory defaults' can be restored by clicking on that button. Source signals show current input.



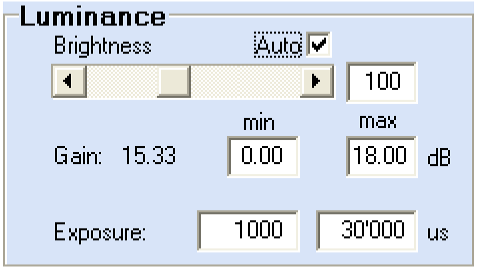
IO Config eco274MVGE 1.4.59 MAC-102030404318

IO assignments

Physical		Logical	
OUT0	n.a.	UART_IN	IN_RS232
OUT1	STROBE	Trigger	IN1
OUT2	n.a.		
OUT3	n.a.		
OUT_TXD	UART_OUT		

Source signals

State	
IN0	🔴
IN1	🟢
IN2	🟢
IN_RS422	🟢
IN_RS232	🟢
UART_OUT	🟢
STROBE	🟢



Luminance

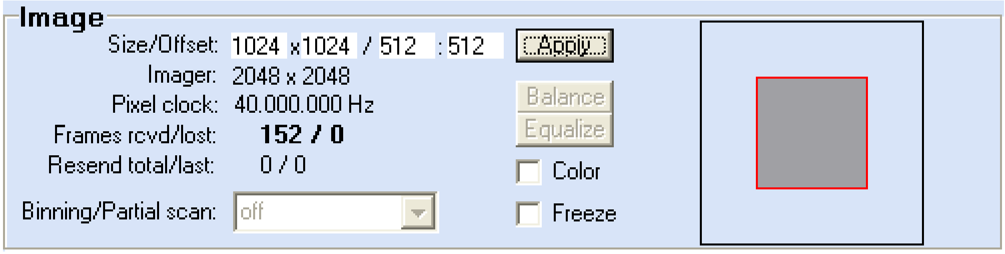
Brightness Auto ☑ 100

Gain: 15.33 min 0.00 max 18.00 dB

Exposure: 1000 30'000 us

6.1 Adjusting an AOI (area of interest)

When clicking on the "AOI" button in the "Image" panel a graph tablet becomes visible that allows for defining an AOI by dragging and resizing a gray target area with a red border inside the imager limits, represented by a rectangle with black borders.



Image

Size/Offset: 1024 x1024 / 512 : 512

Imager: 2048 x 2048

Pixel clock: 40.000.000 Hz

Frames rcvd/lost: **152 / 0**

Resend total/last: 0 / 0

Binning/Partial scan: off

Whenever the left mouse button is released after dragging/resizing the target area, the camera will be adjusted to the new settings. Alternatively the target area can be defined numerically by entering values into the edit fields for Size/Offset and subsequently clicking on "Apply". After clicking on "Apply" the graph tablet will disappear and the new settings are stored in the camera.

6.2 Strobe

The strobe panel allows for accessing the following settings for controlling light sources:

- Strobe duration
- Strobe delay (Start of strobe related to a trigger pulse)
- Strobe polarity (positive/negative)

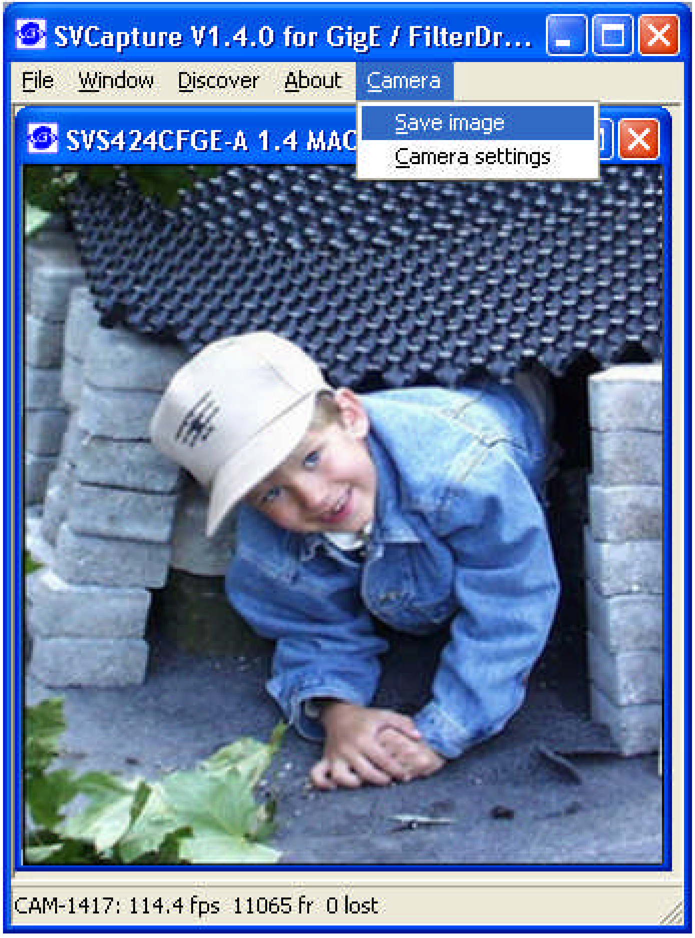
6.3 Persistence

The buttons in the persistence panel allow for saving and restoring all settings:

- Save to EEPROM
- Restore from EEPROM
- Factory defaults
- Close

7 Saving images to disk

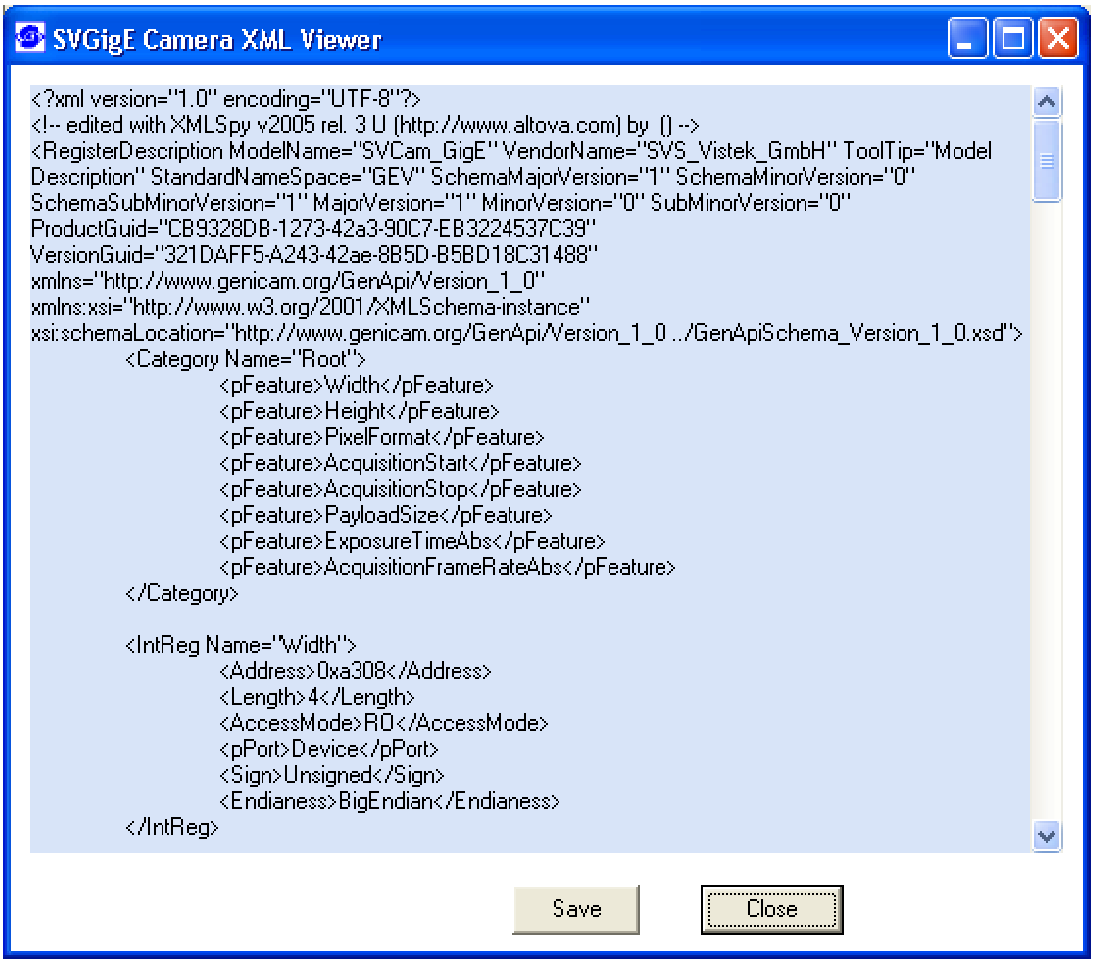
The live image can be saved to disk with the "Save as" item in the Image menu. After selecting a path and specifying a file name a picture in one of the formats JPEG, PNG (Portable Network Graphics), BMP or PIX (raw pixel data) will be saved to disk. Saving of 16-bit images is supported by the PNG and PIX (raw pixel data) formats.



CAM-1417: 114.4 fps 11065 fr 0 lost

8 Displaying a camera’s XML file

Accordingly to the GigE Vision standard a GigE camera provides for an XML file that defines the camera’s capabilities. Though the XML file will usually be processed by software, the SVCapture application allows for displaying a camera’s XML file on screen and saving it to disk. This functionality is available in the “Camera” menu when clicking on the “Show XML file” entry.

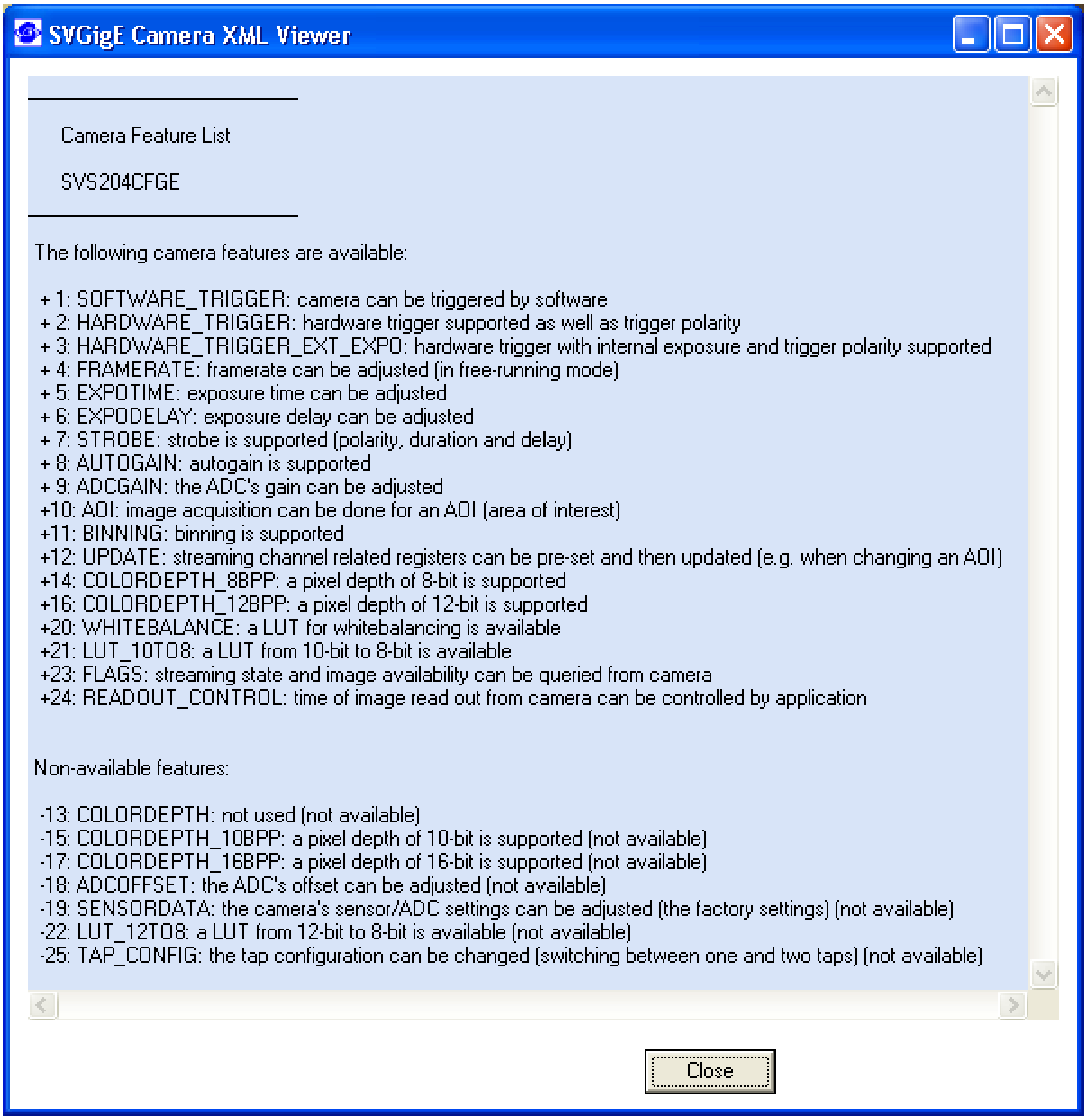


```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XMLSpy v2005 rel. 3 U (http://www.altova.com) by  () -->
<RegisterDescription ModelName="SVCam_GigE" VendorName="SVS_Vistek_GmbH" ToolTip="Model
Description" StandardNameSpace="GEV" SchemaMajorVersion="1" SchemaMinorVersion="0"
SchemaSubMinorVersion="1" MajorVersion="1" MinorVersion="0" SubMinorVersion="0"
ProductGuid="CB9328DB-1273-42a3-90C7-EB3224537C39"
VersionGuid="321DAFF5-A243-42ae-8B5D-B5BD18C31488"
xmlns="http://www.genicam.org/GenApi/Version_1_0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.genicam.org/GenApi/Version_1_0 ../GenApiSchema_Version_1_0.xsd">
        <Category Name="Root">
                <pFeature>Width</pFeature>
                <pFeature>Height</pFeature>
                <pFeature>PixelFormat</pFeature>
                <pFeature>AcquisitionStart</pFeature>
                <pFeature>AcquisitionStop</pFeature>
                <pFeature>PayloadSize</pFeature>
                <pFeature>ExposureTimeAbs</pFeature>
                <pFeature>AcquisitionFrameRateAbs</pFeature>
        </Category>

        <IntReg Name="Width">
                <Address>0xa308</Address>
                <Length>4</Length>
                <AccessMode>RO</AccessMode>
                <pPort>Device</pPort>
                <Sign>Unsigned</Sign>
                <Endianess>BigEndian</Endianess>
        </IntReg>
```

9 Displaying a camera's feature list

A SVGigE camera provides for a set of features out of a list of all defined features. The "Feature list" entry in the "Camera" menu allows for getting information which features a given camera provides for and which are not available.



Camera Feature List

SVS204CFGE

The following camera features are available:

+ 1: SOFTWARE_TRIGGER: camera can be triggered by software
+ 2: HARDWARE_TRIGGER: hardware trigger supported as well as trigger polarity
+ 3: HARDWARE_TRIGGER_EXT_EXPO: hardware trigger with internal exposure and trigger polarity supported
+ 4: FRAMERATE: framerate can be adjusted (in free-running mode)
+ 5: EXPOTIME: exposure time can be adjusted
+ 6: EXPODELAY: exposure delay can be adjusted
+ 7: STROBE: strobe is supported (polarity, duration and delay)
+ 8: AUTOGAIN: autogain is supported
+ 9: ADCGAIN: the ADC's gain can be adjusted
+10: AOI: image acquisition can be done for an AOI (area of interest)
+11: BINNING: binning is supported
+12: UPDATE: streaming channel related registers can be pre-set and then updated (e.g. when changing an AOI)
+14: COLORDEPTH_8BPP: a pixel depth of 8-bit is supported
+16: COLORDEPTH_12BPP: a pixel depth of 12-bit is supported
+20: WHITEBALANCE: a LUT for whitebalancing is available
+21: LUT_10TO8: a LUT from 10-bit to 8-bit is available
+23: FLAGS: streaming state and image availability can be queried from camera
+24: READOUT_CONTROL: time of image read out from camera can be controlled by application

Non-available features:

-13: COLORDEPTH: not used (not available)
-15: COLORDEPTH_10BPP: a pixel depth of 10-bit is supported (not available)
-17: COLORDEPTH_16BPP: a pixel depth of 16-bit is supported (not available)
-18: ADCOFFSET: the ADC's offset can be adjusted (not available)
-19: SENSORDATA: the camera's sensor/ADC settings can be adjusted (the factory settings) (not available)
-22: LUT_12TO8: a LUT from 12-bit to 8-bit is available (not available)
-25: TAP_CONFIG: the tap configuration can be changed (switching between one and two taps) (not available)

10 Other menu items

File | Enable filter driver - Enable or disable a filter driver

File | Exit - Leaving application

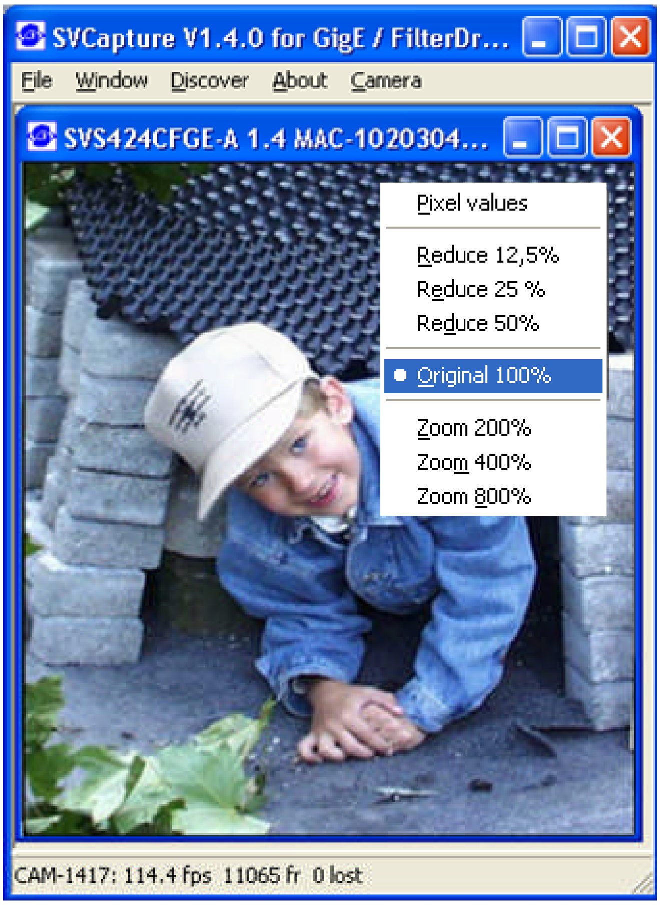
Window | Cascade - Cascade camera views in main window

Window | Tile - Assign each camera view a share of main window
Window | Delete - Close camera view that currently has the focus
About - Program and version information

11 Context menu

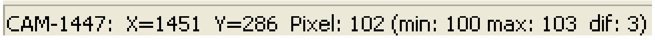
A context menu can be activated in the image window by right clicking on the mouse inside that window. The following functions can be controlled by this menu:

- Displaying pixel coordinates and values
- Reducing the size of an image to 12,5%
- Reducing the size of an image to 25%
- Reducing the size of an image to 50%
- Restoring the original 100% size of an image
- Magnifying resolution to 200% (image fragment scrolled by left mouse click)
- Magnifying resolution to 400% (image fragment scrolled by left mouse click)
- Magnifying resolution to 800% (image fragment scrolled by left mouse click)



CAM-1417: 114,4 fps 11065 fr 0 lost

If “Pixel values” is selected the display switches in the bottom left corner to displaying the min and max values at cursor position along with the difference max-min which represents the image’s noise amplitude at cursor position:



CAM-1447: X=1451 Y=286 Pixel: 102 (min: 100 max: 103 dif: 3)

11.1 Filter Driver

In order to lower CPU load on some PCs it is useful to save CPU power for other tasks. The driver delivered with the camera helps to do so. See **Fehler! Verweisquelle konnte nicht gefunden werden.** for installation.

It is recommended always to enable the filter driver, otherwise images can be lost!

⚠

11.2 Firmware update

A separate tool called **"Firmware Update Tool.exe"** is provided to execute a firmware update.

Depending on the **serial number a firmware update for the internal camera logic** might be useful or necessary. This can be essential in order to **use the current available SVCapture or SDK.**

⚠ **Using unmatched hardware and software (e.g. HW 1.2 with SW 1.4) will NOT work.** Numbering is consistent. So 1.4 SW requires 1.4 firmware!

● In doubt please contact your local distributor.

12 Technical Data

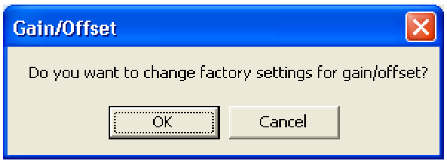
An easy logic allows the control of the camera by different signals to achieve optimum image quality.

12.1 CCD used/cosmetic issues:

- Sony: ICX 424, ICX618, ICX 414, ICX415, ICX204, ICX445, ICX267, ICX285, ICX274 +ICX655

For Details see Sony ICX datasheets

Please note: Factory settings for gain/offset will be irrevocably overwritten when confirming the final message box with OK.



Do you want to change factory settings for gain/offset?

12.2 Signal conditioning

The analog output of the sensor is conditioned by Correlated Double Sampling (CDS) for optimum S/N ratio. Dark level drift is compensated by an "auto zero" amplifier circuit and fed into a video ADC with 14 Bit Bit resolution (type dependant). The data are fed into an internal 64 MB Memory. The camera is controlled by an FPGA and the data packages are sent via an Ethernet controller to the PC.

12.3 Optical and mechanical issues

- Camera size: 38mm x 38mm x 33mm (without connectors and lens)
- Weight: Approx. 100 gram
- Front plate with CS-Mount adapter. 5 mm adapter for C-mount. Except eco285 (C-mount)

- **Distance from Chip surface to frontplate: 11.65mm when C-mount lens is used**

Views on Camera eco424, 415, eco204,eco445, eco267, eco274 svs655 front plate with **CS-Mount** adapter.

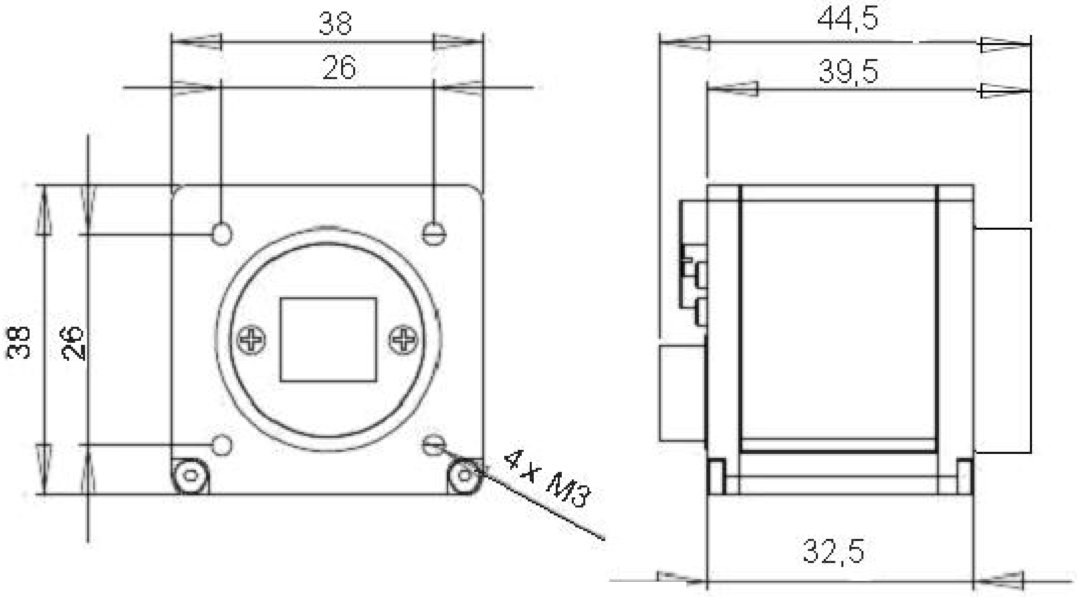
svs285 has front plate with **C-Mount** adapter.

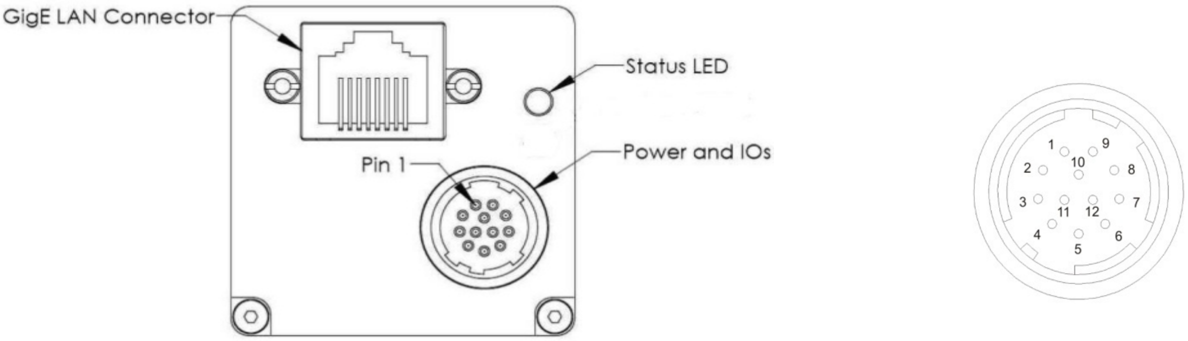
Actually showing C-mount

(Difference from CS to C-mount is + 5mm)

Use 4 x M3 holes in frontplate for mounting !!!

Proper heatsinking is required !!



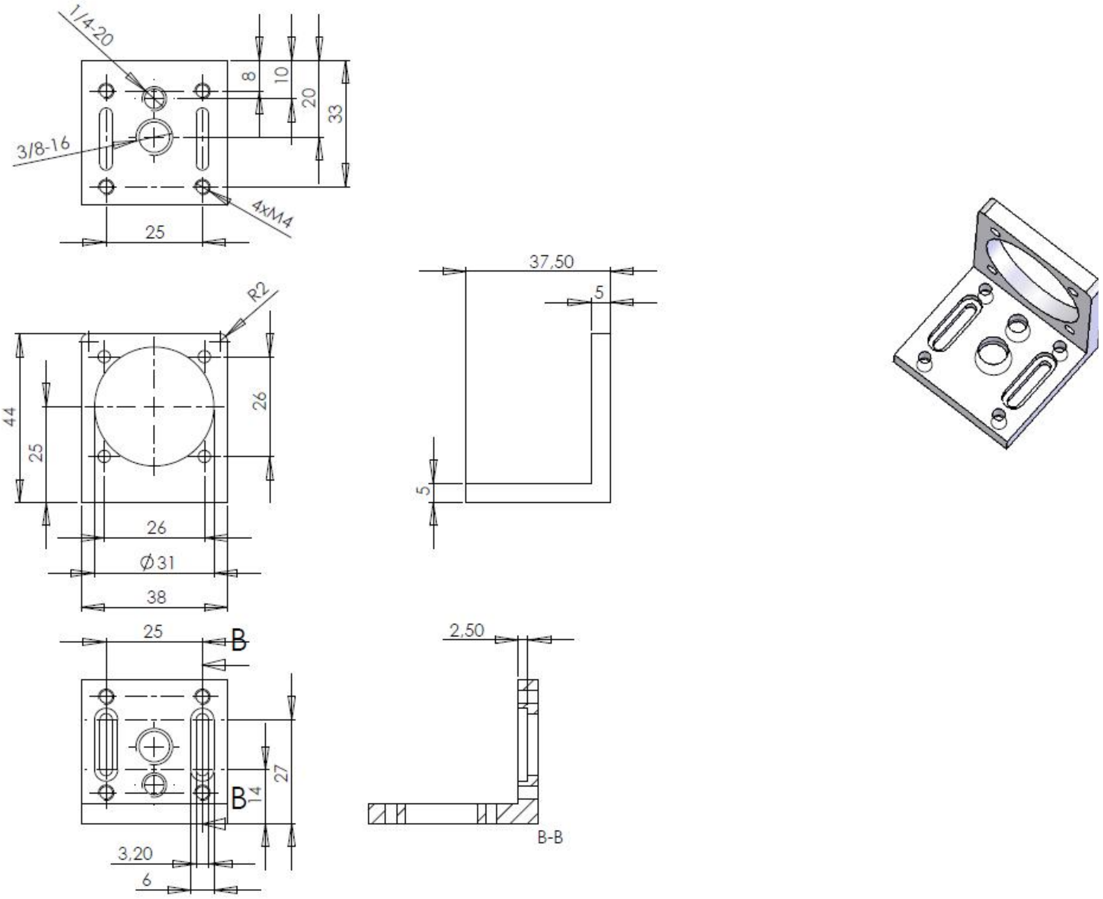


HR10A-10R-12PB (mating connector HR10A-10P-12S)

1 VIN- (GND)
2 VIN+ (10 to 25VDC)
3 RXD data to camera (RS232 Level)
4 TXD data from camera (RS232 Level)
5 IN1 (0- 24 V) digital
6 IN2 (0- 24 V) digital
7 OUT1 (open drain max 24 V, 0.3 A) digital
8 OUT2 (open drain max 24 V, 0.3 A) digital
9 IN3+ (RS422 Level)
10 IN3- (RS422 Level)
11 OUT3 + (RS422 Level)
12 OUT3- (RS422 Level)

Option

Mounting Bracket (do not scale)



Basic electro-optic Specifications of Cameras

eco424

CCD ICX 424 1/3" Interline	640 x 480 Pixel	Pixel Size 7.4 x 7.4 µm
Pixelclock	1 x 50 MHz	
Frame Rate (max.)	124 fps	
Frame Rate (2 x 2 Binning)		
Offset	ca. 40 counts in 12 Bit	
Gain	18 db	
S/N Ratio	> 56dB (9,0 Bit), (Saturation/Dark Noise (RMS))	
Fixed Pattern Noise	ca. 8 counts in 12 Bit	
Photo Response Non-uniformity (PRNU)	+/- 10%	
Spectral Response	380 – 950 nm	Monochrome
Exposure Time (internal)	3 µsec – 2 sec	
Power consumption	3.3W (0.27A @ 12V)	

eco414

CCD ICX 424 1/2" Interline	640 x 480 Pixel	Pixel Size 9.9 x 9.9 µm
Pixelclock	1 x 50 MHz	
Frame Rate (max.)	125 fps	
Frame Rate (2 x 2 Binning)		
Offset	ca. 40 counts in 12 Bit	
Gain	18 db	
S/N Ratio	> 58dB (9.5 Bit), (Saturation/Dark Noise (RMS))	
Fixed Pattern Noise	ca. 8 counts in 12 Bit	
Photo Response Non-uniformity (PRNU)	+/- 10%	
Spectral Response	380 – 950 nm	Monochrome
Exposure Time (internal)	21 µsec – 2 sec	
Power consumption	3.5W (0.29A @ 12V)	

eco618

CCD ICX 618 1/4" Interline	640 x 480 Pixel	Pixel Size 5,6 x 5.6 µm
Pixelclock	1 x 65 MHz	
Frame Rate (max.)	160 fps	
Frame Rate (2 x 2 Binning)		
Offset	ca. 40 counts in 12 Bit	
Gain	18 db	
S/N Ratio	> 58 dB (9,5 Bit), (Saturation/Dark Noise (RMS))	
Fixed Pattern Noise	ca. 8 counts in 12 Bit	
Photo Response Non-uniformity (PRNU)	+/- 10%	
Spectral Response	380 – 950 nm	Monochrome
Exposure Time internal)	65 µsec – 2 sec	
Power consumption	3.3W (0.27A @ 12V)	

eco415

CCD ICX 415 1/2" Interline	780 x 580 Pixel	Pixel Size 8.3 x 8,3 µm
Pixelclock	1 x 50 MHz	
Frame Rate (max.)	86 fps	
Frame Rate (2 x 2 Binning)		
Offset	ca. 40 counts in 12 Bit	
Gain	18 db	
S/N Ratio	> 58dB (9.5 Bit), (Saturation/Dark Noise (RMS))	
Fixed Pattern Noise	ca. 8 counts in 12 Bit	
Photo Response Non-uniformity (PRNU)	+/- 10%	
Spectral Response	380 – 950 nm	Monochrome
Exposure Time (internal)	21 µsec – 2 sec	
Power consumption	3.3W (0.29A @ 12V)	

eco204

CCD ICX 204 1/3" Interline	1024 x 768 Pixel	Pixel Size 4,65 x 4,65 µm
Pixelclock	1 x 50 MHz	
Frame Rate (max.)	47 fps	
Frame Rate (2 x 2 Binning)		
Offset	ca. 40 counts in 12 Bit	
Gain	18 db	
S/N Ratio	> 54dB (9 Bit), (Saturation/Dark Noise (RMS))	
Fixed Pattern Noise	ca. 40 counts in 12 Bit	
Photo Response Non-uniformity (PRNU)	+/- 10%	
Spectral Response	380 – 950 nm	Monochrome
Exposure Time (internal)	16 µsec – 2 sec	
Power consumption	3.0W (0.28A @ 12V)	

eco445

CCD ICX 204 1/3" Interline	1280 x 960 Pixel	Pixel Size 3,75 x 3,75 µm
Pixelclock	1 x 50 MHz	
Frame Rate (max.)	30 fps	
Frame Rate (2 x 2 Binning)		
Offset	ca. 40 counts in 12 Bit	
Gain	18 db	
S/N Ratio	> 56dB (9 Bit), (Saturation/Dark Noise (RMS))	
Fixed Pattern Noise	ca. 40 counts in 12 Bit	
Photo Response Non-uniformity (PRNU)	+/- 10%	
Spectral Response	380 – 950 nm	Monochrome
Exposure Time (internal)	12 µsec – 2 sec	
Power consumption	2.5W (0.28A @ 12V)	

eco267

CCD ICX 267 1/2" Interline	1360 x 1024 Pixel	Pixel Size 4,65 x 4,65 µm
Pixelclock	1 x 50 MHz	
Frame Rate (max.)	25 fps	
Frame Rate (2 x 2 Binning)		
Offset	ca. 40 counts in 12 Bit	
Gain	18 db	
S/N Ratio	> 56dB (9 Bit), (Saturation/Dark Noise (RMS))	
Fixed Pattern Noise	ca. 8 counts in 12 Bit	
Photo Response Non-uniformity (PRNU)	+/- 10%	
Spectral Response	380 - 950 nm	Monochrome
Exposure Time (internal)	39 µsec – 2 sec	
Power consumption	3.5W (0.29A @ 12V)	

eco285

CCD ICX 285 2/3" Interline	1360 x 1024 Pixel	Pixel Size 6,45 x 6,45 µm
Readout Frequency	1 x 40 MHz	
Frame Rate (max.)	34 fps after Jan 2011	
Frame Rate (2 x 2 Binning)	XX fps	
Offset	ca. 40 counts in 12 bit	
Gain	0 to 18 db	
S/N Ratio	> 54dB (9 Bit), (Saturation/Dark Noise (RMS))	
Fixed Pattern Noise	+/- 8 counts in 12 bit	
Photo Response Non-uniformity (PRNU)	+/- 10%	
Spectral Response	380 - 950 nm	Monochrome
Exposure Time (internal)	20 µsec – 2 sec	
Power consumption	3.6W (0.29A @ 12V)	

eco274

CCD ICX 274 1/2” Interline	1600 x 1200 Pixel	Pixel Size 4,4 x 4,4 µm
Pixelclock	1 x 50 MHz	
Frame Rate (max.)	26 fps after Jan 2011	
Frame Rate (2 x 2 Binning)		
Offset	ca. 40 counts in 12 Bit	
Gain	0 to 18 dB	
S/N Ratio	> 54dB (9 Bit), (Saturation/Dark Noise (RMS))	
Fixed Pattern Noise	ca. 8 counts in 12 Bit	
Photo Response Non-uniformity (PRNU)	+/- 10%	
Spectral Response	380 - 950 nm	Monochrome
Exposure Time (internal)	26 µsec – 2 sec	
Power consumption	3.7W (0.32A @ 12V)	

eco655

CCD ICX 655 2/3” Interline	2448 x 2050 Pixel	Pixel Size 3,45 x 3,45 µm
Readout Frequency	1x 50 MHz	
Frame Rate (max.)	10 fps	
Frame Rate (2 x 2 Binning)	18 fps	
Offset	ca. 40 counts in 12 Bit	
Gain	0 to 18 db	
S/N Ratio	> 58dB (9.5 Bit), (Saturation/Dark Noise (RMS))	
Fixed Pattern Noise	< 8 count in 12 Bit	
Photo Response Non-uniformity (PRNU)	+/- 10%	
Spectral Response	380 – 950 nm	Monochrome
Exposure Time (internal)	7 µsec – 1 sec	
Power consumption	3.1W (0.29A @ 12V)	

12.4 Environmental Issues

4.8.1 Europe

The camera is CE tested and the rules of EN 50022-2 apply.

4.8.2 USA and Canada

I. Labeling requirements:
This device complies with part 15 of the FCC Rules. Operation is subject to the following two conditions: (1) This device may not cause harmful interference, and (2) this device must accept any interference received, including interference that may cause undesired operation.

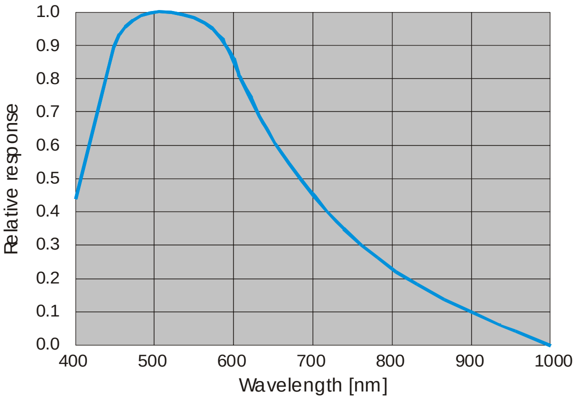
II. Information to the user:
Note: This equipment has been tested and found to comply with the limits for a Class A digital device, pursuant to part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference when the equipment is operated in a commercial environment. This equipment generates, uses, and can radiate radio frequency energy and, if not installed and used in accordance with the instruction manual, may cause harmful interference to radio communications. Operation of this equipment in a residential area is likely to cause harmful interference in which case the user will be required to correct the interference at his own expense.

It is necessary to use a shielded power supply cable. You can than use the "shield contact" on the connector which has GND contact to the camera housing. This is essential for any use. If not done and camera is destroyed due to Radio Magnetic Interference (RMI) **WARRANTY is void**!

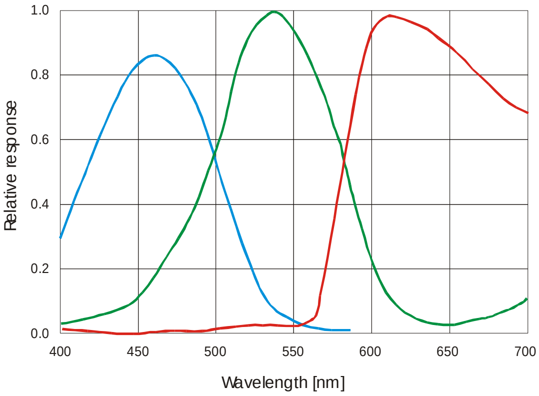
- Operating temperature Spec: -10 to +40 °C
  In order to keep dark current low. To achieve 9 optical Bit, operation at 25 °C max. is recommended.
- **Power:** US/UK and European line adapter can be delivered. Otherwise use +10 to +24V DC with filtered and stabilized power supply.
- **Shock is tested**: About 60 g in 6 ms. **Vibration test** is 10 g in 6 ms
- **RoHS:** All cameras comply with the recommendation of the European Union concerning RoHS Rules.

12.5 Spectral response curves

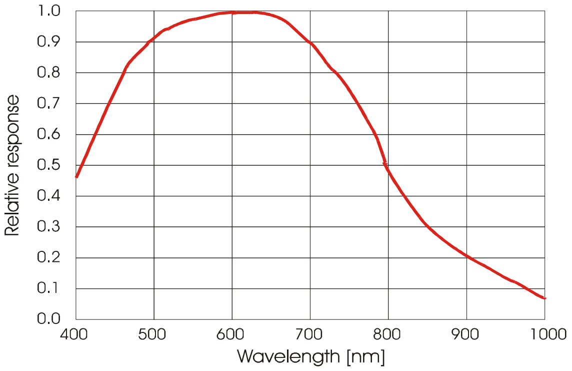
Monochrome Version
eco424, eco414, eco415,
eco204, eco267, eco247,eco655



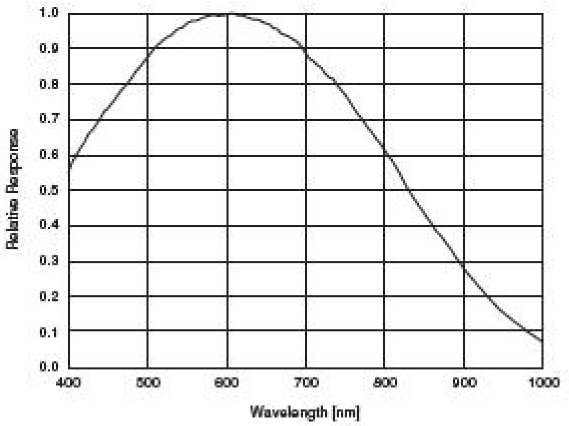
All Color Version



eco285



eco618



Comments on camera temperature

There is no need to worry because the camera has been tested at higher temperatures than specified (40 °C).

12.6 Comments on Color Version

If you have purchased a color version (e.g. **eco**ZZZ**XVGE**) of this camera, please note the following:

In all electrical terms the camera is identical to the black and white versions.

The camera uses a CCD which has a color mosaic filter. This filter is called "Bayer"-filter named after the person who invented it. It has a pattern on the lines which alternates as follows, e.g.:

First line:	RGRGRG.... and so on (R = Red, B = Blue, G = Green)
Second line:	GBGBGB.... and so on

Please note that about half of the pixel are green, a quarter red and a quarter blue. This is due to the maximum sensitivity of the human eye at about 550 nm (green).

Note that the green pixel in the “red” line has different sensitivity than the green pixel in the “blue” line.

Because this camera is a single chip camera it is necessary to use an algorithm which interpolates those colors which are "not known" by the specific pixel. E.g. the red pixel does not know its green and blue components.

This means that the performance of the image depends on the software used.

Please be aware that it is not possible to incorporate the algorithm into the camera so easily. Unlike NTSC/PAL cameras there is no hardware chip available which can do that for such large images. The user has the advantage to alter the colors depending on his needs. Thus the color image must be processed in the PC.

We offer several algorithms in the SVCapture program which influence display rate and image quality. However a color source code is available on request for those who want to write an application.

We offer a complete set for a system setup containing camera, cable, power supply, and software to help solving your applications.

Note that you can disable the color in the SVCapture program. Color processing is not done inside the camera but in the PC. So some CPU power is consumed.

If you have questions do not hesitate to contact us or your local dealer.

13 Warranty Terms

13.1 Standard products warranty and adjustment

Seller warrants that the article to be delivered under this order will be free from detects in material and workmanship under normal use and service for a period of **TWO** years from date of shipment. The liability of Seller under this warranty is limited solely to replacing or repairing or issuing credit (at the discretion for Seller) for such products that become defective during the warranty period.

In order to permit Seller to properly administer this warranty, Buyer shall notify Seller promptly in writing of any claims,; provide Seller with an opportunity to inspect and test the products claimed to be detective. Such inspection may be on customer's premises or Seller may request return of such products at customer's expense. Such expense will subsequently be reimbursed to customer if the product is found to be defective and Buyer shall not return any product without prior return authorization from Seller.

If a returned product is found to be out of warranty or found to be within the applicable specification, Buyer will have to pay an evaluation and handling charge, independent of possible repair and/or replacement costs. Seller will notify Buyer of the amount of said evaluation and handling charges at the time the return authorization is issued. The Seller will inform the Buyer of related repair and/or replacement costs and request authorization before incurring such costs. Buyer shall identify all returned material with Sellers invoice number, under which material has been received.

If more than one invoice applies, material has to be clearly segregated and identified by applicable invoice numbers. Adjustment is contingent upon Sellers examination of product, disclosing that apparent defects have not been caused by misuse, abuse, improper installation of application, repair, alteration, accident or negligence in use, storage, transportation or handling. In no event shall Seller be liable to Buyer for loss of profits, loss of use, or damages of any kind based upon a claim for breach of warranty.

13.2 Development Product Warranty

Developmental products of Seller are warranted to be free from defects in materials and workmanship and to meet the applicable preliminary specification only at the time of receipt by Buyer and for no longer period of time in all other respects the warranties made above apply to development products.

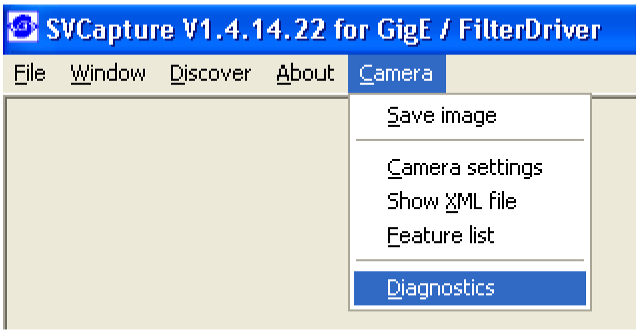
The aforementioned provisions do not extend the original warranty period of any article which has been repaired or replaced by Seller. If Warranty Label of camera is broken Warranty is void!

SELLER MAKES NO OTHER WARRANTIES; EXPRESS OR IMPLIED; AND SPECIFICALLY; SELLER MAKES NO WARRANTY OF MERCHANTABILITY OF FITNESS FOR PARTICULAR PURPOSE.

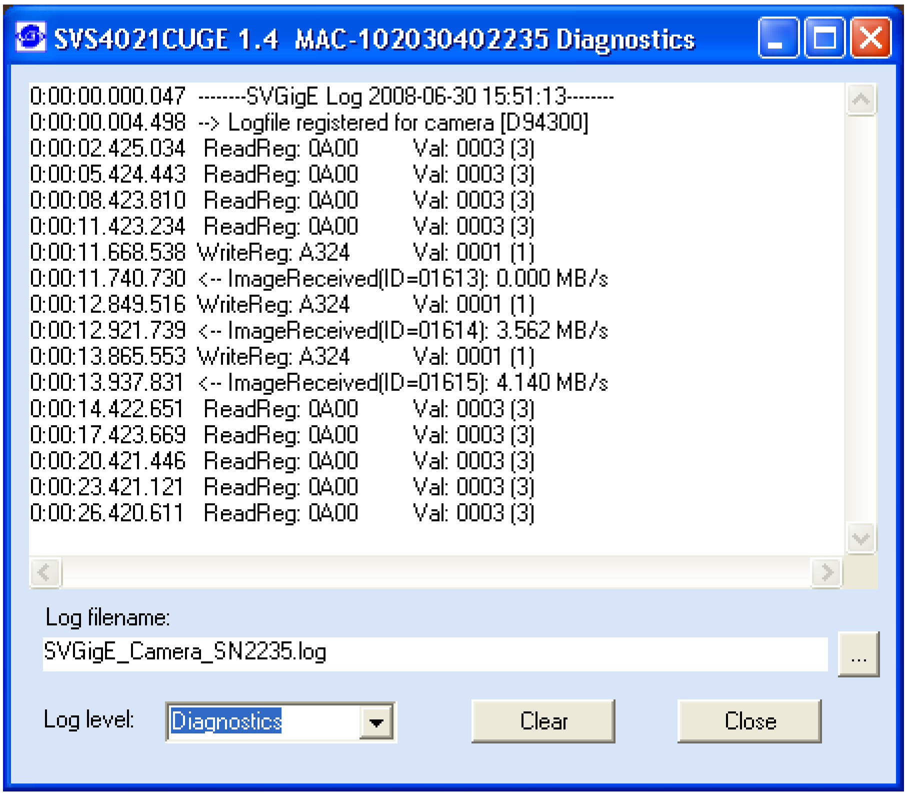
14 Appendix A – Troubleshooting

14.1 Get camera diagnostics

When clicking on the Camera | Diagnostics menu entry a dialog will open that allows for receiving camera diagnostics information.



The content of the Diagnostics window is low level register information. Usually it is needed only for remote analysis purposes of problem situations.



```
0:00:00.000.047  -------SVGigE Log 2008-06-30 15:51:13-------
0:00:00.004.498  --> Logfile registered for camera [D94300]
0:00:02.425.034  ReadReg: 0A00        Val: 0003 (3)
0:00:05.424.443  ReadReg: 0A00        Val: 0003 (3)
0:00:08.423.810  ReadReg: 0A00        Val: 0003 (3)
0:00:11.423.234  ReadReg: 0A00        Val: 0003 (3)
0:00:11.668.538  WriteReg: A324       Val: 0001 (1)
0:00:11.740.730  <-- ImageReceived(ID=01613): 0.000 MB/s
0:00:12.849.516  WriteReg: A324       Val: 0001 (1)
0:00:12.921.739  <-- ImageReceived(ID=01614): 3.562 MB/s
0:00:13.865.553  WriteReg: A324       Val: 0001 (1)
0:00:13.937.831  <-- ImageReceived(ID=01615): 4.140 MB/s
0:00:14.422.651  ReadReg: 0A00        Val: 0003 (3)
0:00:17.423.669  ReadReg: 0A00        Val: 0003 (3)
0:00:20.421.446  ReadReg: 0A00        Val: 0003 (3)
0:00:23.421.121  ReadReg: 0A00        Val: 0003 (3)
0:00:26.420.611  ReadReg: 0A00        Val: 0003 (3)
```

Log filename: SVGigE_Camera_SN2235.log

Log level: Diagnostics

The following items provide some hints how to solve problem situations where a connection between a PC and a camera can not be established successfully.

Problem: **A camera does not appear in the discovery dialog**

Solutions: Click on "Refresh" in order to repeat the Discovery action. Disconnect and reconnect the camera's power cable and repeat the Discovery action after a few seconds. Make sure the PC has got a valid IP address.Shutdown and restart the PC and repeat the Discovery action. Check firewall settings (See "Firewall considerations" below).

Problem: **A camera does not display a picture after it has been discovered and selected**

Solutions: Force a valid IP address in to the camera using the dialog that comes up when right- clicking on an entry in the discovery dialog. Close the camera view and select the camera again in the Discovery dialog. Shutdown and restart the PC and select the camera again after discovery. Check firewall settings (e.g. disable firewall temporarily and select the camera again after discovery). Check if the processor speed of the PC is sufficient for displaying images at the chosen frame rate. If available, connect the camera to a different network card.

Problem: **No communication is possible between the camera and the PC**

Solution: Check if a firewall is active. This can prevent to establish any communication because this is a network issue check firewall setting. Try to disable temporary the firewall and try again.

Problem: **The video stream stops when adjusting the camera settings**

Solution: This situation happens when the product between frame rate and exposure time becomes 1 second or greater. Any of the settings can be reduced in order to get the camera operational again. The exposure settings field will become red in order to signal that one of the frame rate or exposure setting has to be reduced.

Problem: **Camera does not respond to light**

Solution: Please execute following steps:
Check if camera is in a "Free Running"-mode. When done, check with "SVCapture" program if you can read back any data from the camera like: Type of CCD, S/N number exposure time settings and so on. If you trigger the camera by hardware: Check if the Exsync signal is present. The signal swing for must be minimum 5V (max. 24V) Source must provide 10 mA. Below such level the drivers in the camera will not work. Check also the quality and swing. If these signals are not there or do not have the right quality

(like spikes) the camera can not read out any frame or delivers distorted images.

Problem: **The image is present but distorted**

Solution: Try different operation mode. Like if triggered gives bad results try “Free Running” mode and reduce frame rate to minimum half possible one. Check if you are using original “INTEL”® chip set in your PC! If problem still exist call your local support.

Problem: **The image of a color version camera looks “ugly” or false colors appear.**

Solution: If the raw image looks ok than pixel need to be shifted by either one or one line. The image color depends on the algorithm used. If the algorithm is starting with the wrong pixel such effects appear.

Problem: **The colors of a color version are not perfect especially when using halogen light**

Solution: Halogen light contains strong portions of IR radiation. Use cut off filters at around 730 nm like Schott KG 3 to prevent IR radiation reaching the CCD.

Problem: **You use 64 bit OS system, but SVcapture does not work**

Solution: This is normal. Intead pls check the sample programs in SDK. They will wok without problem**.**

15 Troubleshooting Request List V1.3

Dear valued customer,

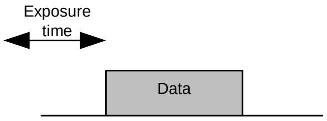
In order to help you with your camera and any interfacing problems we request that you fill in a description of your problems when you use the camera. Please fax this form to us:

1) Type of camera (e.g. eco247XVGE)
2) Serial Number
3) Accessories used and where purchased or self made
   a) Power supply
   b) Cable
   c) Lens type and focal length
4) Firmware version as well as operation mode, (send screenshot of **SVCapture** program)
5) Description of the phenomena, e.g.:
   a) missing lines or columns
   b) distorted or noisy image (if possible send jpg image)
   c) solarization effect
   d) missing Bit, contrast less image
6) PC used, Chip Set/Brand and type of Ethernet card
7) Operating system used, e.g. WIN/XP/7/Linux

16 Appendix B – Basic timing for different operation modes

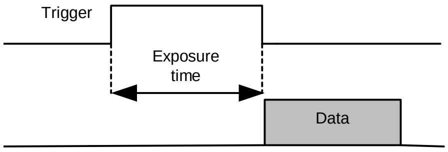
16.1 Free Running

A frame is readout automatically. There is no need to trigger the camera in order to get data. The enclosed software allows the user to set exposure time in µSec. The time set stays resident after power off if the configuration is saved to camera if stored before.



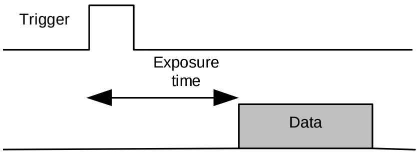
16.2 External Trigger and Pulsewidth of Trigger

In this mode the camera is waiting for an external trigger which starts integration and read out. Exposure time can be varied using the length of the Trigger pulse (I.E. between the high going edge and the low going edge). The time settings in the control software are not activated. This mode is useful in applications where the light level of the sceen changes during operation and the framegrabber can provide such a signal. Change of exposure time is possible from one frame to the next.



16.3 External and Software trigger and internal exposure time setting

The frame rate is determined by the number of Trigger pulses per time unit. With each positive transition (going high) the camera will readout a frame. Exposuretime is set in the same way as in the free running mode. Exposure time can be changed online during operation. The time set stays resident after power off, if the configuration is saved to the camera.



17 Appendix C: SDK Description for Camera Control

SVGigE API

GigE Vision Camera
Software Development Kit (SDK)

Version: 1.4.18.42
Date: 2009-09-24

17.1 Overview

The GigE Vision standard is an award-winning specification for connecting cameras to computers by a Gigabit Ethernet cable. That type of connection is well suited for transferring the huge data streams that are usually related to image capturing devices. The current SVS GigE API software development kit (SDK) provides for integrating GigE cameras into own applications.

17.2 Prerequisites

In order to give an example for data transfer requirements, a camera with a resolution of 1600 by 1200 pixels and which is running at 35 frames per second will generate a data stream of 1600 x 1200 x 35 = 67,2 Mbytes per second. The bandwidth of the network connection must be capable of transferring the image data stream from the camera to the client computer with the image viewing or processing application.

Before a GigE camera can be successfully used in a system, all network parameters have to be adjusted to appropriate settings. This will be done by the SVCapture program which usage is described in the "SVCapture user guide" document. A successful work with the SVS GigE API software development kit can be accomplished only when it has been proved that a camera can be operated over the network on full frame rate from the PC where the software development takes place.

17.3 SVGigE SDK components

The SVGigE SDK consists of the following components:

- SVGigE.h
- SVGigE.cpp
- SVGigE.dll
- SVGigE.sys
- SVGigETLFilter.dll
- SVGigETLWinsock.dll
- SVGigE SDK user guide.pdf (this document)

The **DLL components** have to be in the **same directory as the executable** or alternatively they can be put into the windows\system32 directory. The header file and the cpp file will usually be put into the development directories. They are needed for developing applications under various development environments.

Before running the SVGigEFilter.dll the SVGigE driver has to be installed accordingly to the instructions in the "SVCapture user guide" documentation.

17.4 Development environments

The SVGigE SDK has been designed as DLL components (one API and two transport layers: filter driver and winsock based) which export C-style functions to be used by applications. A cpp file is provided in the SDK which implements the more than 100 functions that constitute the SVGigE API. The cpp file forwards function calls from the software application to the corresponding functions that are contained in the API-DLL module. For this purpose the DLL modules will be loaded once when the first function call arrives in the API.
The chosen programming interface of the API-DLL module is the least common denominator which should ensure a broad applicability of the DLL module in most programming environments that can compile C code. Three example applications are currently provided that comprise the Borland C++ Builder 6 environment, the Microsoft VisualStudio 6 as well as the Microsoft Visual Studio 2005 environments.
The example applications show how to get a handle on a camera in the own network, how to adjust settings like frame rate and exposure time as well as how to receive the image data stream by a callback function in the application. For demonstration purposes the image data stream is being displayed on screen as a live image from the selected camera.

17.5 Windows x64 support

The SVGigE SDK supports 64-bit platforms. All functions in the SVGigE API apply to x64 platforms in the same way as they do for Win32 platforms. Although win32 applications like SVCapture.exe are also supported on x64 platforms in the compatibility mode, there exists a set of runtime components that run natively on x64 platforms:

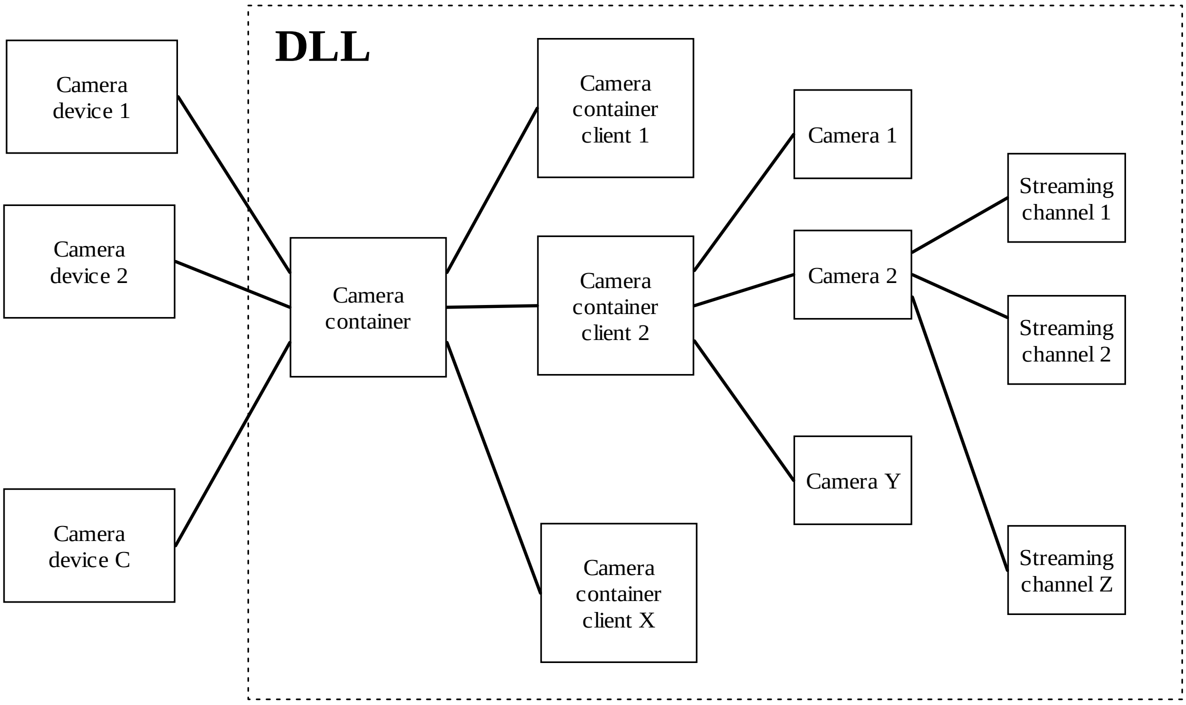
- **x64\SVGigE.sys** – a filter driver for high-performance image streaming
- **SVGigETLFilter.x64.dll** – a transport layer DLL for the SVGigE driver
- **SVGigETLWinsock.x64.dll** – a transport layer DLL based on Winsock
- **SVGigE.x64.dll** – the SVGigE API DLL used by camera applications
- **SVGigE_ExampleVC8.x64.exe** – a sample application using all listed components

17.6 Documentation

The SVGigE API documentation is provided in the following documents:

- SVGigE SDK user guide.pdf – an overview documentation (this document)
- SVGigE.h – a header file with inline documentation for each function
- Documentation\index.html – a html based documentation that has been generated from the inline information in the SVGigE.h document

17.7 System architecture



The system architecture has been chosen accordingly to the following objectives:

- Multiple cameras can be connected over the network to a single PC where a SVGigE.dll along with transport layer DLLs provide for a simultaneous access to all cameras
- Multiple client applications can use the same SVGigE.dll for connecting to cameras
- An application can use a subset or all of the cameras simultaneously
- Multiple streaming channels can be opened by an application, one to each of the cameras in order to perform different further image processing, for example sending all images to a viewer and running an image processing on them for each of the images

NOTE: Currently each camera should be controlled only by a single dedicated application although the system architecture would allow for connecting multiple applications to a single camera. The reason is that each camera maintains only a single control channel. Thus a second client application would

not get a handle on the camera as long as another client maintains a control channel to that camera.
Looking at the system architecture it should become obvious that the DLL needs a handle for performing operations on a camera container, on a camera or on a streaming channel respectively. The client application will obtain those handles when registering for a camera container client, a camera or a streaming channel respectively. Those handles have to be stored locally and be used in all subsequent function calls to the different areas in the DLL. All handles will be valid until closure activities will be performed for a streaming channel, a camera or a camera container client respectively.

17.8 Function groups

```
* Function categories:
 * ---------------------------------------------------------
 *    1 - Camera: Discovery and bookkeeping
 *    2 - Camera: Connection
 *    3 - Camera: Information
 *    4 - Stream: Channel control and statistics
 *    5 - Stream: Transfer parameters
 *    6 - Stream: Image access
 *    7 - Stream: Image conversion
 *    8 - Stream: Image characteristics
 *    9 - Stream: Image statistics
 *   10 - Stream: Messaging channel
 *   11 - Controlling camera: Frame rate
 *   12 - Controlling camera: Exposure
 *   13 - Controlling camera: Gain and offset
 *   14 - Controlling camera: Auto gain/exposure
 *   15 - Controlling camera: Acquisition trigger
 *   16 - Controlling camera: Strobe
 *   17 - Controlling camera: Tap balance
 *   18 - Controlling camera: Image parameter
 *   19 - Controlling camera: Image appearance
 *   20 - Special control: IOMux configuration
 *   21 - Special control: IO control
 *   22 - Special control: Serial communication
 *   23 - Special control: Direct register and memory access
 *   24 - Special control: Persistent settings and recovery
 *   25 - General functions
 *   26 - Diagnostics
 * ---------------------------------------------------------
 *   99 - Deprecated functions
```

17.9 Detailed function listing

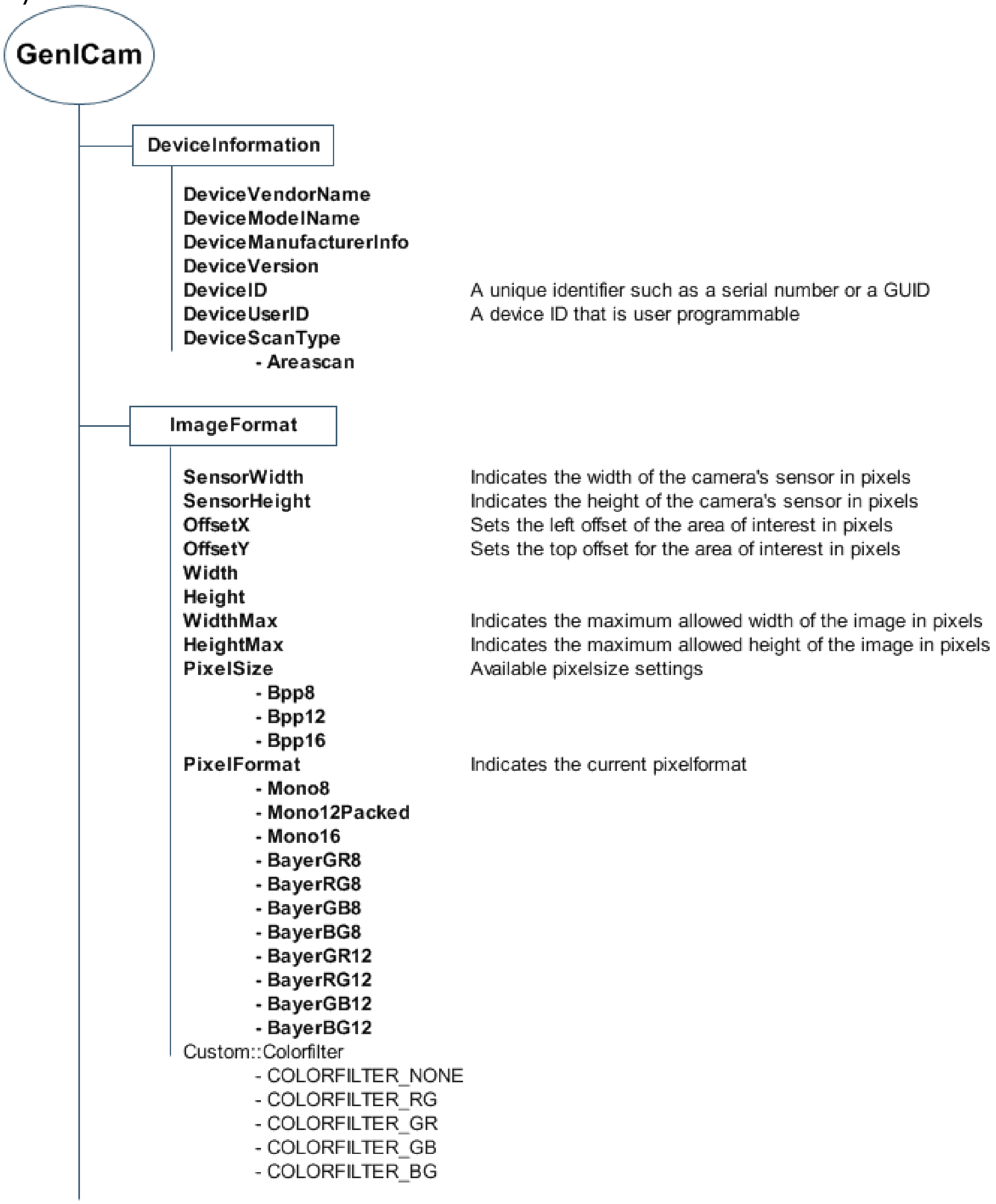
```
*
* 0 - GigE DLL (implicitly called)
 * ---------------------------------
 *  isVersionCompliantDLL()
 *  isLoadedGigEDLL()
 *  getCamera()
 *
 * 1 - Camera: Discovery and bookkeeping
 * -------------------------------------
 *  CameraContainer_create()
 *  CameraContainer_delete()
 *  CameraContainer_discovery()
 *  CameraContainer_getNumberOfCameras()
 *  CameraContainer_getCamera()
 *  CameraContainer_findCamera()
 *
 * 2 - Camera: Connection
 * ----------------------
 *  Camera_openConnection()
 *  Camera_closeConnection()
 *  Camera_setIPAddress()
 *  Camera_forceValidNetworkSettings()
 *
 * 3 - Camera: Information
 * ----------------------
 *  Camera_getManufacturerName()
 *  Camera_getModelName()
 *  Camera_getDeviceVersion()
 *  Camera_getManufacturerSpecificInformation()
 *  Camera_getSerialNumber()
 *  Camera_setUserDefinedName()
 *  Camera_getUserDefinedName()
 *  Camera_getMacAddress()
 *  Camera_getIPAddress()
 *  Camera_getSubnetMask()
 *  Camera_getPixelClock()
 *  Camera_isCameraFeature()
 *  Camera_readXMLFile()
 *
 * 4 - Stream: Channel control and statistics
 * ------------------------------------------
 *  StreamingChannel_create()
 *  StreamingChannel_delete()
 *  StreamingChannel_getBufferInfo()
 *  StreamingChannel_setReadoutTransfer()
 *  StreamingChannel_getReadoutTransfer()
 *  StreamingChannel_getFrameLoss()
 *  StreamingChannel_getActualFrameRate()
 *  StreamingChannel_getActualDataRate()
 *  StreamingChannel_getPeakDataRate()
 *
 * 5 - Stream: Transfer parameters
 * -------------------------------
 *  Camera_evaluateMaximalPacketSize()
 *  Camera_setStreamingPacketSize()
 *  Camera_setInterPacketDelay()
 *  Camera_getInterPacketDelay()
 *  Camera_setMulticastMode()
 *  Camera_getMulticastMode()
 *  Camera_getMulticastGroup()
 *
 * 6 - Stream: Image access
 * ------------------------
 *  Image_getDataPointer()
 *  Image_getBufferIndex()
 *  Image_getSignalType()
 *  Image_getCamera()
 *  Image_release()
 *
 * 7 - Stream: Image conversion
 * ----------------------------
 *  Image_getImageRGB()
 *  Image_getImageGray()
 *  Image_getImage12bitAs8bit()
 *  Image_getImage12bitAs16bit()
 *  Image_getImage16bitAs8bit()
 *
 * 8 - Stream: Image characteristics
 * ---------------------------------
 *  Image_getSizeX()
 *  Image_getSizeY()
 *  Image_getPitch()
 *  Image_getImageSize()
 *  Image_getPixelType()
 *
 * 9 - Stream: Image statistics
 * ----------------------------
 *  Image_getImageID()
 *  Image_getTimestamp()
 *  Image_getTransferTime()
 *  Image_getPacketCount()
 *  Image_getPacketResend()
 *
 * 10 - Stream: Messaging channel
 * ------------------------------
 *  Stream_createEvent()
 *  Stream_addMessageType()
 *  Stream_removeMessageType()
 *  Stream_isMessagePending()
 *  Stream_registerEventCallback()
 *  Stream_unregisterEventCallback()
 *  Stream_getMessage()
 *  Stream_getMessageData()
 *  Stream_getMessageTimestamp()
 *  Stream_releaseMessage()
 *  Stream_flushMessages()
 *  Stream_closeEvent()
 *
 * 11 - Controlling camera: Frame rate
 * --------------------------------------
 *  Camera_setFrameRate()
 *  Camera_getFrameRate()
 *  Camera_getFrameRateMin()
 *  Camera_getFrameRateMax()
 *  Camera_getFrameRateRange()
 *  Camera_getFrameRateIncrement()
 *
 * 12 - Controlling camera: Exposure
 * -----------------------------------
 *  Camera_setExposureTime()
 *  Camera_getExposureTime()
 *  Camera_getExposureTimeMin()
 *  Camera_getExposureTimeMax()
 *  Camera_getExposureTimeRange()
 *  Camera_getExposureTimeIncrement()
 *  Camera_setExposureDelay()
 *  Camera_getExposureDelay()
 *  Camera_getExposureDelayMax()
 *  Camera_getExposureDelayIncrement()
 *
 * 13 - Controlling camera: Gain and offset
 * -------------------------------------------
 *  Camera_setGain()
 *  Camera_getGain()
 *  Camera_getGainMax()
 *  Camera_getGainMaxExtended()
 *  Camera_getGainIncrement()
 *  Camera_setOffset()
 *  Camera_getOffset()
 *  Camera_getOffsetMax()
 *
 * 14 - Controlling camera: Auto gain/exposure
 * ----------------------------------------------
 *  Camera_setAutoGainEnabled()
 *  Camera_getAutoGainEnabled()
 *  Camera_setAutoGainBrightness()
 *  Camera_getAutoGainBrightness()
 *  Camera_setAutoGainDynamics()
 *  Camera_getAutoGainDynamics()
 *  Camera_setAutoGainLimits()
 *  Camera_getAutoGainLimits()
 *  Camera_setAutoExposureLimits()
 *  Camera_getAutoExposureLimits()
 *
 * 15 - Controlling camera: Acquisition trigger
 * -------------------------------------------
 *  Camera_setAcquisitionControl()
 *  Camera_getAcquisitionControl()
 *  Camera_setAcquisitionMode()
 *  Camera_setAcquisitionModeAndStart()
 *  Camera_getAcquisitionMode()
 *  Camera_softwareTrigger()
 *  Camera_softwareTriggerID()
 *  Camera_softwareTriggerIDEnable()
 *  Camera_setTriggerPolarity()
 *  Camera_getTriggerPolarity()
 *
 * 16 - Controlling camera: Strobe
 * -------------------------------
 *  Camera_setStrobePolarity()
 *  Camera_getStrobePolarity()
 *  Camera_setStrobePosition()
 *  Camera_getStrobePosition()
 *  Camera_getStrobePositionMax()
 *  Camera_getStrobePositionIncrement()
 *  Camera_setStrobeDuration()
 *  Camera_getStrobeDuration()
 *  Camera_getStrobeDurationMax()
 *  Camera_getStrobeDurationIncrement()
 *
 * 17 - Controlling camera: Tap balance
 * ------------------------------------
 *  Camera_saveTapBalanceSettings()
 *  Camera_loadTapBalanceSettings()
 *  Camera_setTapConfiguration()
 *  Camera_getTapConfiguration()
 *  Camera_setAutoTapBalanceMode()
 *  Camera_getAutoTapBalanceMode()
 *  Camera_setTapBalance()
 *  Camera_getTapBalance()
 *
 * 18 - Controlling camera: Image parameter
 * ---------------------------------------
 *  Camera_getImagerWidth()
 *  Camera_getImagerHeight()
 *  Camera_getSizeX()
 *  Camera_getSizeY()
 *  Camera_getPitch()
 *  Camera_getImageSize()
 *  Camera_setBinningMode()
 *  Camera_getBinningMode()
 *  Camera_setAreaOfInterest()
 *  Camera_getAreaOfInterest()
 *  Camera_getAreaOfInterestRange()
 *  Camera_getAreaOfInterestIncrement()
 *  Camera_resetTimestampCounter()
 *  Camera_getTimestampCounter()
 *  Camera_getTimestampTickFrequency()
 *
 * 19 - Controlling camera: Image appearance
 * ----------------------------------------------------
 *  Camera_getPixelType()
 *  Camera_setPixelDepth()
 *  Camera_getPixelDepth()
 *  Camera_setWhiteBalance()
 *  Camera_getWhiteBalance()
 *  Camera_getWhiteBalanceMax()
 *  Camera_setGammaCorrection()
 *  Camera_setLookupTableMode()
 *  Camera_getLookupTableMode()
 *  Camera_setLookupTable()
 *  Camera_getLookupTable()
 *
 * 20 - Special control: IOMux configuration
 * ----------------------------------------------------------------------
 *  Camera_getMaxIOMuxIn()
 *  Camera_getMaxIOMuxOut()
 *  Camera_setIOAssignment()
 *  Camera_getIOAssignment()
 *
 * 21 - Special control: IO control
 * ----------------------------------------------------------------------
 *  Camera_setIOMuxIn()
 *  Camera_getIOMuxIn()
 *  Camera_setIO()
 *  Camera_getIO()
 *  Camera_setAcqLEDOverride()
 *  Camera_getAcqLEDOverride()
 *  Camera_setLEDIntensity()
 *  Camera_getLEDIntensity()
 *
 * 22 - Special control: Serial communication
```

```
 * ----------------------------------------------------
 *  Camera_setUARTBuffer()
 *  Camera_getUARTBuffer()
 *  Camera_setUARTBaud()
 *  Camera_getUARTBaud()
 *
 * 23 - Special control: Direct register and memory access
 * ----------------------------------------------------
 *  Camera_setGigECameraRegister()
 *  Camera_getGigECameraRegister()
 *  Camera_writeGigECameraMemory()
 *  Camera_readGigECameraMemory()
 *
 * 24 - Special control: Persistent settings and recovery
 * ---------------------------------------------------
 *  Camera_writeEEPROM()
 *  Camera_readEEPROM()
 *  Camera_restoreFactoryDefaults()
 *
 * 25 - General functions
 * ---------------------
 *  SVGigE_estimateWhiteBalance()
 *  SVGigE_writeImageToBitmapFile()
 *
 * 26 - Diagnostics
 * ---------------
 *  getErrorMessage()
 *  Camera_registerForLogMessages()
 *
 * ------------------------------------------------------
 * 99 - Deprecated functions
 * ------------------------------------------------------
 *  Camera_startAcquisitionCycle()
 *  Camera_setTapCalibration()
 *  Camera_getTapCalibration()
 *  Camera_setLUTMode()
 *  Camera_getLUTMode()
 *  Camera_createLUTwhiteBalance()
 *  Camera_stampTimestamp()
 *  Camera_getTimestamp()
 *
```

17.10 GenICam Interface

Each GigE Vision compliant camera contains an XML file that describes available features as well as their mapping to registers. In the following, an overview shall be given about the features that are exposed from SVCam GigE cameras by XML file.

Currently, the features exposed by XML represent a subset of all features that the cameras have. This is a development snapshot of the XML file which is supposed to provide for access to all features in future, as they are available by C-functions.



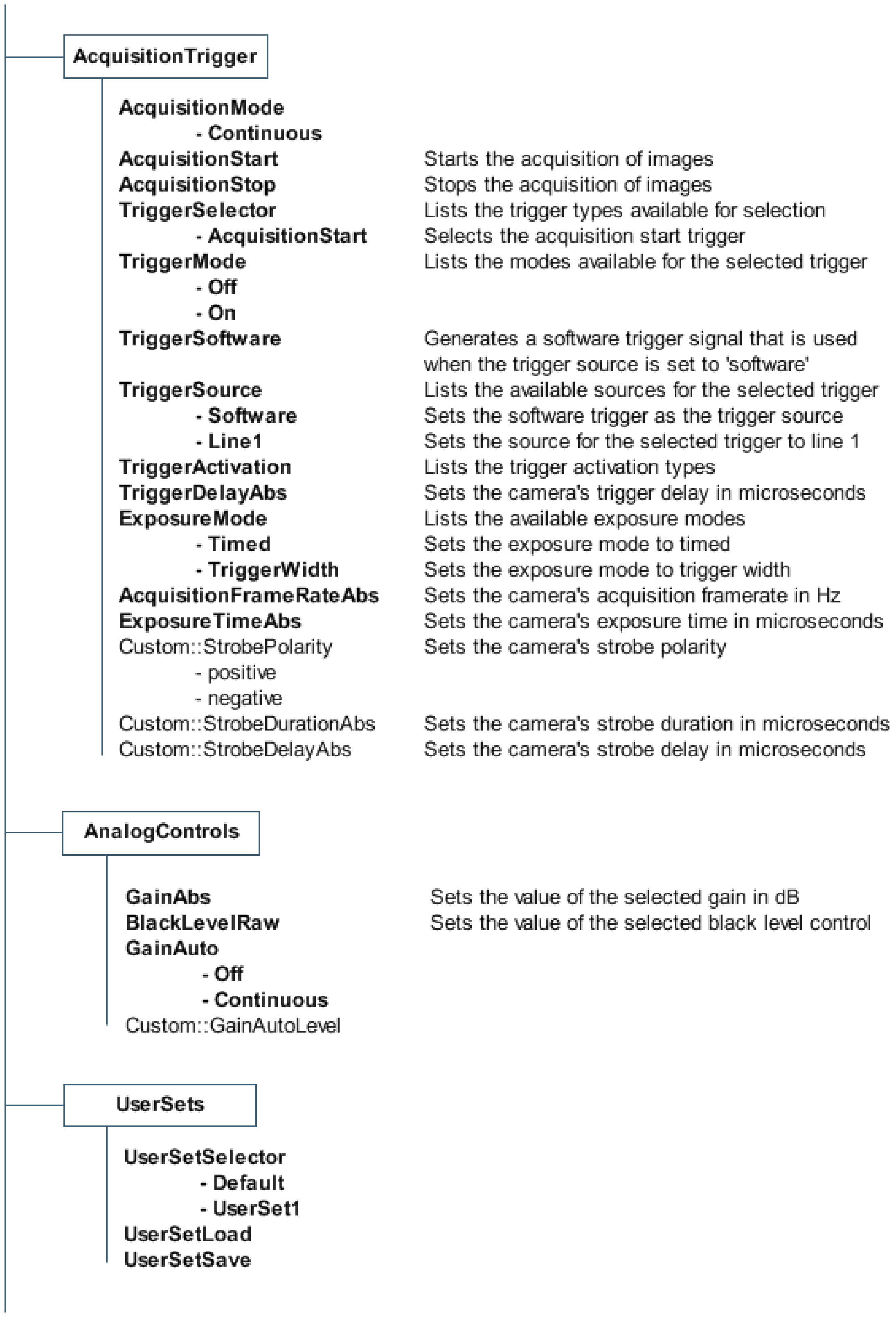
GenICam

DeviceInformation

- DeviceVendorName
- DeviceModelName
- DeviceManufacturerInfo
- DeviceVersion
- DeviceID — A unique identifier such as a serial number or a GUID
- DeviceUserID — A device ID that is user programmable
- DeviceScanType
  - Areascan

ImageFormat

- SensorWidth — Indicates the width of the camera's sensor in pixels
- SensorHeight — Indicates the height of the camera's sensor in pixels
- OffsetX — Sets the left offset of the area of interest in pixels
- OffsetY — Sets the top offset for the area of interest in pixels
- Width
- Height
- WidthMax — Indicates the maximum allowed width of the image in pixels
- HeightMax — Indicates the maximum allowed height of the image in pixels
- PixelSize — Available pixelsize settings
  - Bpp8
  - Bpp12
  - Bpp16
- PixelFormat — Indicates the current pixelformat
  - Mono8
  - Mono12Packed
  - Mono16
  - BayerGR8
  - BayerRG8
  - BayerGB8
  - BayerBG8
  - BayerGR12
  - BayerRG12
  - BayerGB12
  - BayerBG12
- Custom::Colorfilter
  - COLORFILTER_NONE
  - COLORFILTER_RG
  - COLORFILTER_GR
  - COLORFILTER_GB
  - COLORFILTER_BG



AcquisitionTrigger	
AcquisitionMode	
- Continuous	
AcquisitionStart	Starts the acquisition of images
AcquisitionStop	Stops the acquisition of images
TriggerSelector	Lists the trigger types available for selection
- AcquisitionStart	Selects the acquisition start trigger
TriggerMode	Lists the modes available for the selected trigger
- Off	
- On	
TriggerSoftware	Generates a software trigger signal that is used when the trigger source is set to 'software'
TriggerSource	Lists the available sources for the selected trigger
- Software	Sets the software trigger as the trigger source
- Line1	Sets the source for the selected trigger to line 1
TriggerActivation	Lists the trigger activation types
TriggerDelayAbs	Sets the camera's trigger delay in microseconds
ExposureMode	Lists the available exposure modes
- Timed	Sets the exposure mode to timed
- TriggerWidth	Sets the exposure mode to trigger width
AcquisitionFrameRateAbs	Sets the camera's acquisition framerate in Hz
ExposureTimeAbs	Sets the camera's exposure time in microseconds
Custom::StrobePolarity	Sets the camera's strobe polarity
- positive	
- negative	
Custom::StrobeDurationAbs	Sets the camera's strobe duration in microseconds
Custom::StrobeDelayAbs	Sets the camera's strobe delay in microseconds
AnalogControls	
GainAbs	Sets the value of the selected gain in dB
BlackLevelRaw	Sets the value of the selected black level control
GainAuto	
- Off	
- Continuous	
Custom::GainAutoLevel	
UserSets	
UserSetSelector	
- Default	
- UserSet1	
UserSetLoad	
UserSetSave	

TransportLayer

- PayloadSize
- GevVersionMajor
- GevVersionMinor
- GevDeviceModelsBigEndian
- GevDeviceModeCharacterSet
- GevInterfaceSelector
  - GevMACAddress
  - GevSupportedIPConfigurationLLA
  - GevSupportedIPConfigurationDHCP
  - GevSupportedIPConfigurationPersistentIP
  - GevCurrentIPConfiguration
  - GevCurrentIPConfigurationLLA
  - GevCurrentIPConfigurationDHCP
  - GevCurrentIPConfigurationPersistentIP
  - GevCurrentIPAddress
  - GevCurrentSubnetMask
  - GevCurrentDefaultGateway
- GevFirstURL
- GevSecondURL
- GevNumberOfInterfaces
- GevMessageChannelCount
- GevStreamChannelCount
- GevPersistentIPAddress
- GevPersistentSubnetMask
- GevPersistentDefaultGateway
- GevCCP
    - OpenAccess
    - ExclusiveAccess
    - ControlAccess
- GevHeartbeatTimeout
- GevStreamChannelSelector
  - GevSCPSPacketSize
  - GevSCPD

17.11 Best programming practices

In the following a step-by-step procedure will be given how to communicate to a SVCam GigE camera by a software application. A working application that implements all hints can be found in the sources of the SVGigE_Example.VC8 program which comes as a part of the SVGigE SDK.
The project file of the SVGige_Example.VC8 program was generated by the Visual Studio 2005 development environment which can be used to compile the project in order to get to a running application:
NOTE: An application which was compiled with Visual Studio 2005 needs a runtime environment in order to run on a WinXP computer. Dependent on the software components that have already been installed on the target computer it might be necessary to install the Microsoft VC runtime package (**vcredist_x86.exe**) in addition to the runtime DLLs that are shipped with Visual Studio 2005. A discussion of the involved problems can be found here: http://forums.microsoft.com/MSDN/ShowPost.aspx?PostID=164132&SiteId=1

17.11.1 Connecting to the camera container

Whenever an application is going to operate SVCam GigE cameras it has to become a client of the camera container first which is located inside the SVGigE.dll module.

```
CameraContainerClient_handle CameraClient;
CameraClient = CameraContainer_create(SVGigETL_TypeWinsock);
```

When establishing a connection to the camera container the application will demand for the type of transport layer to be used for the network connection to the camera, either a FilterDriver (recommended) or a Winsock (for first testing purposes) based transport layer. The FilterDriver based transport layer guaranties the highest possible performance
Once the application is done with cameras the connection to the container will be closed what will allow the SVGigE DLL to release all resources which were occupied for that client.

```
CameraContainer_delete(CameraClient);
```

Between becoming a camera client and shutting down the connection to the camera container an application will be allowed to operate all SVCam GigE cameras that are available in the attached network segment. The **CameraClient** handle is used for identifying the application against the camera container module inside the SVGigE.dll.

17.11.2 Connecting to a camera

Based on a valid connection to the camera container an application can query for available cameras and can open connections to one or multiple cameras.

```
Camera_handle Camera;

if( SVGigE_SUCCESS != CameraContainer_discovery(CameraClient) )
  return;

int NumCameras = CameraContainer_getNumberOfCameras(CameraClient);

if( 0 == NumCameras )
  return;

Camera = CameraContainer_getCamera(CameraClient, Index);// Index < NumCameras
```

The Camera handle relates a specific camera to an application. The application will have to open a connection to the camera before being able to issue commands to the camera and to initiate an image stream.

```
Camera_openConnection(Camera);
```

Once an application is done with a camera it needs to close the connection and thus allowing the camera container to release all resources that have been kept occupied in meantime for the active application-camera relation.

```
Camera_closeConnection(Camera);
```

After a camera connection has been closed it can be re-opened again at any time as long as the program is a valid client to the camera container.

Opening a camera requires the existence of a valid IP address in the camera. The easiest way for making sure connectivity is to force valid network settings:

```
unsigned int IPAddress;
unsigned int SubnetMask;
SVGigERet = Camera_forceValidNetworkSettings(Camera, &IPAddress, &SubnetMask);
```

That function will do nothing in case the camera has already got a valid IP address. If this is not the case the function will first force a free IP address in current subnet into the camera. Subsequently that IP address may be assigned as a fixed IP address to the camera if this is wanted:

```
Camera_setIPAddress(hCamera,IPAddress,NetMask);
```

When the available cameras in the attached network segment have changed it will be necessary to run the discovery command again in order to synchronize the information in the camera container with the actually available set of cameras in the network. Afterwards one has to take care since a previous index of a specific camera might have changed in meantime.

Alternatively to selecting a camera by index a search string can be provided to a findCamera function:

```
Camera = CameraContainer_findCamera(CameraClient, CameraItem);
```

In this case a string CameraItem will be provide to a camera container function which tries to find a match between all discovered cameras where the given string is contained in at least one of the MAC address, IP address, serial number and user defined name items.

17.11.3 Identifying a camera

Every GigE camera maintains a set of information strings which allow for identifying a camera in the network.

```
CString CameraName    = Camera_getMacAddress(Camera);
CString CameraName    = Camera_getIPAddress(Camera);
CString ModelName     = Camera_getModelName(Camera);
CString Manufacturer  = Camera_getManufacturerName(Camera);
CString SerialNumber  = Camera_getSerialNumber(Camera);
CString DeviceVersion = Camera_getDeviceVersion(Camera);
CString CameraName    = Camera_getUserDefinedName(Camera);
```

An application could query those strings and could display one or a combination of those information strings to a user for allowing a selection of a specific camera. However, applications that know beforehand which camera to use for a given purpose can use the findCamera function that was described before.

17.11.4 Determining maximal packet size

From network history of last decades, a packet size upto 1500 bytes is quite common in most network infrastructures. This allows to transfer even huge files splitted into small packets in an efficient way. At the same time it becomes obvious that the overhead introduced by those relatively small packet sizes may limit the maximal performance.

A way out of the performance bottleneck is the usage of bigger packet sizes, in particular for fast streaming devices like digital cameras. Therefore current network cards allow for switching "Jumbo frames" on with upto 16 Kbytes packet size. The overall maximal usable network packet size, however, depends on all network system components and will be equal to the lowest value that is supported by any network card or switch on the network path.

The SDK allows for determining the maximal possible network packet size automatically based on "test frames" that are defined in the GigE Vision specification. Therefore the following function has to be called before opening a streaming channel on a given network topology:

```
SVGigE_RETURN
        Camera_evaluateMaximalPacketSize(hCamera,          // a valid camera handle
         &MaximalPacketSize);                       // the result will be returned to this variable
```

In all standalone (non-Multicast) applications the determined value should be used unchanged since it represents a value that allows for highest performance and reliability of the image data transfer under given circumstances.

Nevertheless, there might be situations like multicast scenarios, where the master of a camera needs to intentionally further limit the packet size in order to allow all clients to connect to an image stream with their specific maximal network packet size. In that case the master application can use the following function for limiting the network packet size to a value below the previously determined maximal packet size:

```
SVGigE_RETURN
                Camera_setStreamingPacketSize(hCamera,              // a valid camera handle
          StreamingPacketSize);                         // intended new streaming packet size
```

17.11.5 Opening a streaming channel

The main goal for operating cameras is to get an image stream back to the application. Once a connection to the desired camera has been opened a stream can be initiated and a callback function for receiving that stream can be registered.

```
if( SVGigE_SUCCESS !=
     StreamingChannel_create (
         &StreamingChannel,  // a streaming channel handle will be returned
         CameraClient,      // a valid camera container client handle
         Camera,           // a valid camera handle
         BufferCount,       // buffer count
         &ImageCallback,    // callback function pointer where images are delivered to
         this) )           // the this pointer serves as a context that will be returned
   return;                 // each time the callback function is called
```

The application will receive a handle for the streaming channel on success. That handle will be used later for shutting down the streaming channel once the application has finished working with the channel.

```
StreamingChannel_delete(StreamingChannel);
```

Between opening and closing the streaming channel an application specific callback function will be called for every image that the camera has captured. It is up to the application to process the image further in that callback function.

17.11.6 Receiving images in a callback function

It is important that the application defines the callback function as **__stdcall** in order to make sure that the arguments are passed through correctly.

```
SVGigE_RETURN __stdcall ImageCallback(Image_handle Image, void* Context);
```

The **Context** argument will have the same value as it was provided by the application when creating the streaming channel. It can be used for example for establishing a reference to the object that is expected to receive that image stream in one of its member functions.

In the **Image** parameter the application will find a handle to the just captured camera image. That handle will be used to query all needed information about the image from the SVGigE DLL. The image handle is valid only inside the current call to the callback function. Any use of an image handle at a time will always result in receiving an error code back from the function that was tried.

Thus the usual way for further image processing is to query all interesting information inside the callback function and to copy that information into a structure that remains accessible to the application after returning from the callback function.

```
IMAGE_INFO *ImageInfo = new IMAGE_INFO;
ImageInfo->ImageData  = Image_getDataPointer(Image);
ImageInfo->SizeX      = Image_getSizeX(Image);
ImageInfo->SizeY      = Image_getSizeY(Image);
ImageInfo->PixelType  = Image_getPixelType(Image);
ImageInfo->ImageID    = Image_getImageID(Image);
```

The newly allocated structure can be posted for example by a Windows message to a function that is responsible for displaying the image or for further image processing.

17.11.7 Processing image data

Once an image has been forwarded by a callback function to a specific processing function the image data will be worked on. Usually the camera has already started capturing a subsequent image at the time when a previous image is processed. This requires working with at least two image buffers in order to avoid overwriting data that is being processed by a subsequent image acquisition process.

The higher the delay between image acquisition and image processing becomes the more buffer might be needed for avoiding any conflicts. The number of needed buffers has to be estimated based on the specific runtime conditions. The resulting number will be passed to the GigE driver as an argument when creating a streaming channel. The driver will allocate that number of buffers where each buffer will have enough space for holding one complete image.

An image processing function will operate for example based on the following parameters:

```
ImageInfo->ImageData
ImageInfo->SizeX
ImageInfo->SizeY
```

The **ImageData** pointer will be a reference to a buffer that was allocated by the driver from persistent kernel memory and which address is mapped to user space. The application is responsible for not violating the boundary that is given by the product of the **SizeX** and **SizeY** parameters.

In case of color images that are coded as bayer pattern the SVGigE API offers a function for converting those images to a RGB format. The appropriate conversion needs to be done inside the callback function.

17.11.8 Decoding 12-bit images

When a camera is switched to 12-bit mode the received image usually needs decoding in order to further process that image e.g. for displaying purposes (8-

bit needed) respectively storage purposes (most often 16-bit needed). A conversion will be based on the layout of 12-bit data where 2 pixels are folded into 3 bytes in the following way:

<table>
<tr><td>pixel A
bit 11-4</td><td>pixel A
bit 3-0</td><td>pixel B
bit 3-0</td><td>pixel B
bit 11-4</td></tr>
<tr><td>byte 0</td><td colspan="2">byte 1</td><td>byte 2</td></tr>
</table>

The SDK provides for the following functions that perform conversion from 12-bit to 8-bit and 16-bit as well as from 16-bit to 8-bit:

```
Image_getImage12bitAs8bit(Camera, Buffer8bit, BufferLength);
Image_getImage12bitAs16bit(Camera, Buffer16bit, BufferLength);
Image_getImage16bitAs8bit(Camera, Buffer8bit, BufferLength);
```

17.11.9 **Adjusting tap balance automatically**

Whenever a dual-tap camera is opened by calling

```
Camera_openConnection(Camera);
```

a balance check between the right and the left tap will be performed on the first image that is regularly captured by the application. In case of a need to run a tap balance, new gain factors are determined for achieving an optimal tap balance. It may take multiple images until a final balance is reached. Once a balance is adjusted, the tap balance is finished and the gain factors are not changed anymore. Those gain factors will apply until the camera is closed.

An application can take more control over the auto tap balance process if this is needed. The following options are offered for "auto tap balance" mode:

- Off
- Once
- Continuous

An application can switch auto tap balance OFF before opening a streaming channel in order to prevent from running any adjustments. Further an application can switch auto tap balance to ONCE at any time in order to run a one-time adjustment of tap balance. Only in case of big changes of gain and exposure during a camera usage it might be recommended to switch auto tap balance to CONTINUOUS. In that case every captured image is checked for tap balance and individual gain factors for right and left balance are adjusted appropriately whenever this is needed. There is no visible runtime penalty when switching auto tap balance to CONTINUOUS since only a small stripe left and right the middle of the sensor are taken into account for evaluating a deviation from tap balance.

The following two functions allow for switching the auto tap balance mode to explained values and for retrieving current state:

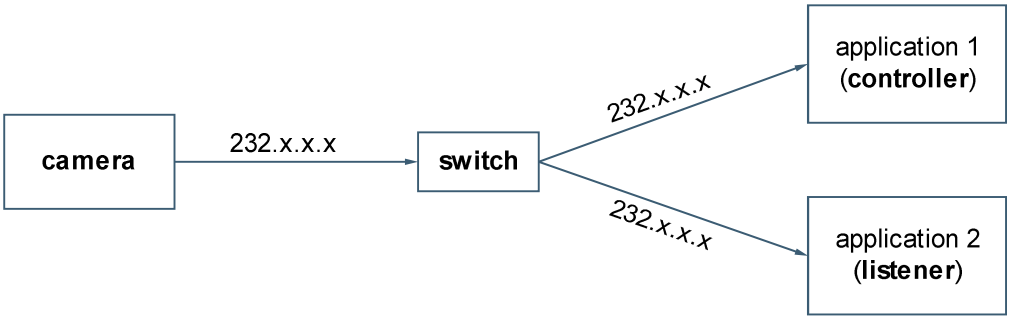
```
Camera_setAutoTapBalanceMode(Camera, AutoTapBalanceMode);
Camera_getAutoTapBalanceMode(Camera, *AutoTapBalanceMode);
```

A reset to factory defaults or to arbitrary values inside a range of +/- 1.5 dB can be performed by the following function. The second function retrieves current setting:

```
Camera_setTapBalance (Camera, TapBalance);
Camera_getTapBalance (Camera, *TapBalance);
```

17.11.10 Distributing image streams by multicast

The image stream of a GigE camera can be distributed to multiple PCs by using the multicast capabilities of current network hardware, in particular multicast-enabled switches. The following picture explains the basic principle:



A camera streams image data in a certain IP range (232.x.x.x) which has been determined for dynamic local host multicast allocation by the IANA organisation (see http://www.iana.org/assignments/multicast-addresses for more details).

A multicast-enabled switch is supposed to distribute the image stream to one, two or more applications. The advantage of having a switch distributing image streams is the optimal bandwidth usage between camera and switch as well as between switch and particular applications. There is no bandwidth wasted in the described scenario.

A multicast session consists of one controlling application and one or more listening applications. Only the controlling application has write-access to the camera. All other applications can only listen to the image stream that is initiated and controlled by the controlling application.

The connection between the controlling application and the camera is established as usual based on IP addresses. However, a switch has to be specifically instructed for forwarding an image stream also to one or more listening application. This is done by network packets which are sent out by applications regularly all 60 seconds and which detection in a switch is known as "**IGMP snooping**".

NOTE: A switch has to support "IGMP snooping" in order to establish a successful multicast session. Most often that feature is switched off in normal mode and has to be enabled by the web interface of a switch. Please refer to instructions of a switch that is supposed to work in multicast mode.

Programming interface

When opening a camera, an application is supposed to check whether the camera streams already in multicast mode using the function:

Camera_getMulticastMode()

The multicast mode will be one of the following items:

- MULTICAST_MODE_NONE
- MULTICAST_MODE_LISTENER
- MULTICAST_MODE_CONTROLLER

If the camera is already streaming in multicast mode then the application can connect to the camera but as listening application (MULTICAST_MODE_LISTENER).

Otherwise, if a camera is still idle (MULTICAST_MODE_NONE), an application can request for a multicast session by switching a camera's mode to "multicast" (MULTICAST_MODE_CONTROLLER) before opening the streaming channel. The appropriate function is:

Camera_setMulticastMode()

Afterwards the application might check the automatically generated IP for the multicast group in the "232.x.x.x" range using the following function:

Camera_getMulticastGroup()

Packet size consideration

A controlling application will usually open a streaming channel at the highest available packet size, e.g. 16112 bytes. If another application connects to that already running streaming channel then a packet size problem may arise if the listening application has not at least the packet size available that the image stream is currently working with.

Therefore a controlling application is supposed to limit the packet size of the image data stream to a value which allows all intended listening applications to connect to that image stream.

If a listening application encounters a packet size problem, it will usually receive the start and stop packages of each image since those are of small size. This causes a progress of image statistics like image counter, however, there will be no image data (image payload) available. The packet size has to be reduced in this case by the controlling application in order to allow all listening applications for receiving all image payload data properly.

An application might choose to display a message to the user in the above described case like "Image payload missing. Multicast packet size too big? " as it was done in the "SVGigE_Example.VC8" application in the SDK.

17.11.11 **Receiving messages asynchronously**

An application can receive messages like e.g. "start of transfer" asynchronously in three different ways:

1. by registering an event callback
2. by using a blocking call when asking for new messages
3. by polling the event FIFO at arbitrary time

Before receiving any messages by event, an application will have to create a new event and then register one or multiple message types against that event. Afterwards, one of the above methods can be used for receiving events.

17.12 Adjusting IO configuration

SVS cameras have been equipped with a flexible IO interface including standard signals like TRIGGER and STROBE as well as common IO and UART lines. The availability of particular signals is dependent on camera model. For the ECO-Line and HR-Line cameras the IO interface is available for free configuration.
The configuration of IO lines is based on a switching matrix as it is shown below.

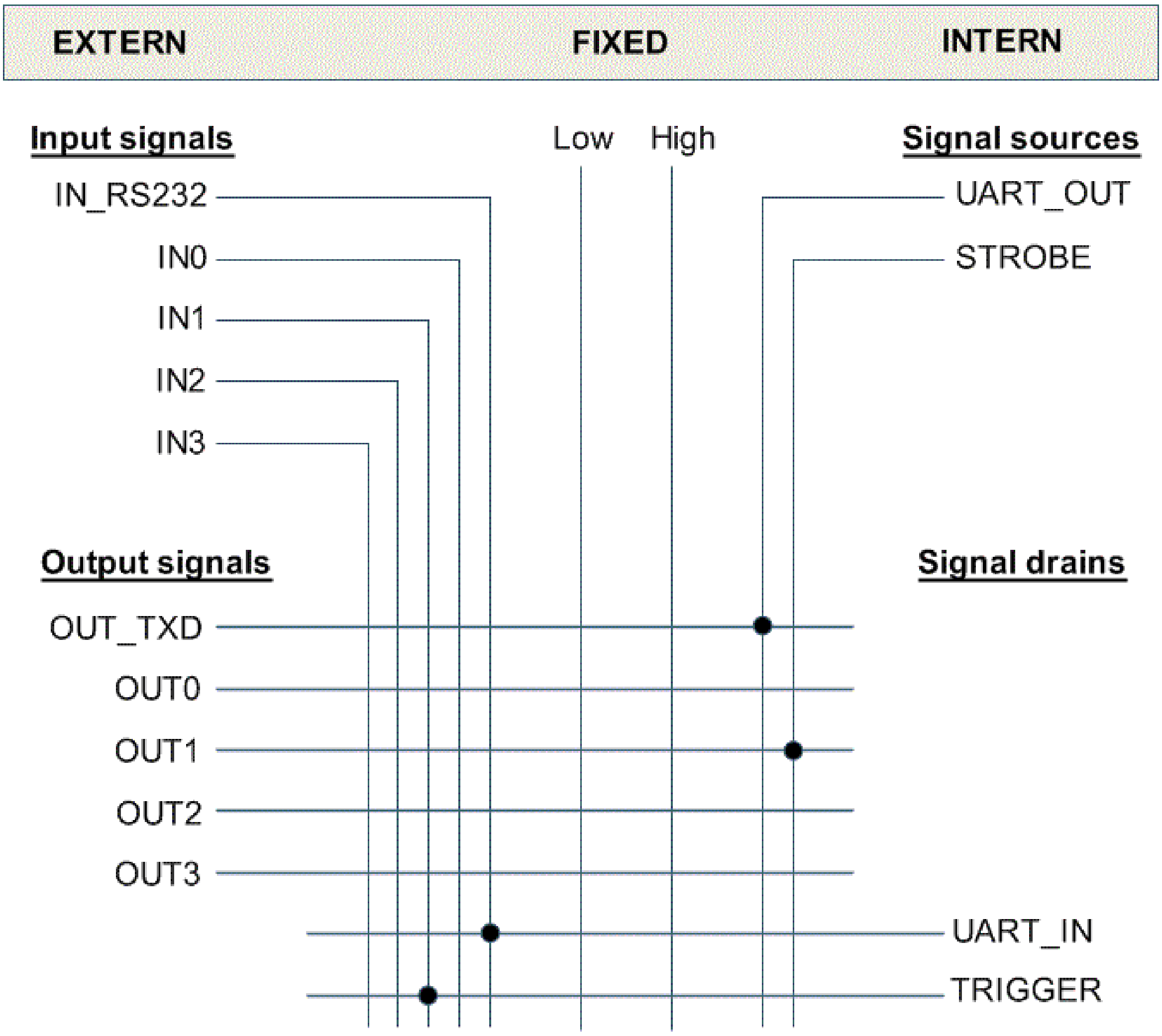


Chart: Flexible IO assignment matrix for SVS VISTEK cameras (with default settings)

The above shown default assignment can be set by program as follows:

```
        Camera_setIOAssignment(Camera,SVGigE_IOMux_OUT_IO_TXD,SVGigE_IOMux_IN_UART_OUT);
  Camera_setIOAssignment(Camera,SVGigE_IOMux_OUT_UART_IN,SVGigE_IOMux_IN_IO_RXD);
  Camera_setIOAssignment(Camera,SVGigE_IOMux_OUT_TRIGGER,SVGigE_IOMUX_IN1);
  Camera_setIOAssignment(Camera,SVGigE_IOMUX_OUT1,SVGigE_IOMux_IN_STROBE);
```

All not used outputs are usually not connected to any input. This is done as follows:

```
  Camera_setIOAssignment(Camera,SVGigE_IOMUX_OUT0,(SVGigE_IOMux_IN) 0);
  Camera_setIOAssignment(Camera,SVGigE_IOMUX_OUT2,(SVGigE_IOMux_IN) 0);
  Camera_setIOAssignment(Camera,SVGigE_IOMUX_OUT3,(SVGigE_IOMux_IN) 0);
```

Dependent on application requirements, above shown default configuration can be changed arbitrarily by assigning other than the default inputs to an output. This involves in particular, that external input lines can also be put through to external output lines, e.g. for signal conversion purposes.

Fixed output

An output line can also be driven to a determined value, Low or High.
e.g. put a Low signal to OUT2:

```
  Camera_setIOAssignment(Camera,SVGigE_IOMUX_OUT2, SVGigE_IOMux_IN_FIXED_LOW);
```

e.g. put a High signal to OUT2:

```
  Camera_setIOAssignment(Camera,SVGigE_IOMUX_OUT2, SVGigE_IOMux_IN_FIXED_HIGH);
```

Trigger simulation

The ability to drive an output line to a determined value can also be used for simulating a hardware trigger. From the camera operation point of view, this is almost identical to releasing a trigger by an external hardware trigger signal. The only difference is that the electrical signal which triggers an image acquisition will be generated inside the camera itself.
The described trigger simulation can be done as follows:

```
        Camera_setIOAssignment(Camera,SVGigE_IOMux_OUT_TRIGGER,
                                      SVGigE_IOMux_IN_FIXED_LOW);
  Camera_setIOAssignment(Camera,SVGigE_IOMux_OUT_TRIGGER,
                                SVGigE_IOMux_IN_FIXED_High);
```

The example applies to an active rising edge for releasing a trigger. Afterwards one should not forget to connect the trigger with an appropriate external trigger line since otherwise external trigger signals would not have any effect:

```
  Camera_setIOAssignment(Camera,SVGigE_IOMux_OUT_TRIGGER,SVGigE_IOMUX_IN1);
```

Input signals

In addition to putting input signals through, e.g. for TRIGGER, their current state can also be determined by program as follows:

```
IO_SIGNAL ProgrammedSignalValue;
Camera_getIO(Camera, SVGigE_IOMUX_IN2, &ProgrammedSignalValue;
```

The above function call determines the current state of the IN2 signal line. Although the state of input signals can also be set by program, this is not recommended. Instead, the fixed Low and High signals should be used in all cases where an output line has to be driven to a fixed value.

17.13 RS232 communication

Before starting RS232 communication, the UART’s parameters have to be adjusted appropriately. The following functions adjust and retrieve those settings:

```
Camera_setUARTBaud(Camera_handle hCamera,SVGigE_BaudRate BaudRate);
Camera_getUARTBaud(Camera_handle hCamera,SVGigE_BaudRate *ProgrammedBaudRate);
```

Subsequently, data can be sent and received over serial RS232 line using a block size of 512 bytes. If more than 512 bytes have to be sent or received, then those data will be transferred using that block size.

```
Camera_setUARTBuffer(Camera_handle hCamera,unsigned char *Data,int DataLength);
Camera_getUARTBuffer(Camera_handle hCamera,
                     unsigned char *Data,
                     int *DataLengthReceived,
                     int DataLengthMax,
                     float Timeout);
```

NOTE: If DataLengthMax is set to less than the serial buffer size (512 bytes) and if DataLengthMax is not sufficient for returning all buffered data then data loss will occur.

HINT: If more data will be transferred than the serial buffer size (512 bytes), then DataLengthMax has to match that serial buffer size. Only then a seamless transfer of data chunks can be performed, each chunk being the size of the serial buffer (512 bytes).

• Applying a LUT

SVS cameras may support a LUT, either 12-bit or 10-bit. They will be uploaded respectively downloaded to/from camera as an array of bytes using the following functions:

```
Camera_setLookupTable(Camera_handle hCamera,
                      unsigned char *LookupTable,
                      int LookupTableSize);
Camera_getLookupTable(Camera_handle hCamera,
                      unsigned char *ProgrammedLookupTable,
                      int LookupTableSize);
```

After uploading a LUT to the camera, the source pixel values from ADC will be translated to 8bit output values based on the LUT table as it is shown below.

12-bit input	8-bit output
4095	255
4094	
	1
18	
17	
16	
	0
2	
1	
0	

10-bit input	8-bit output
1023	255
1022	
1021	
1020	
7	1
6	
5	
4	
3	0
2	
1	
0	

Before uploading/downloading a LUT, a check needs to be performed what size of LUT is supported. The two values that need to be queried are:

- CAMERA_FEATURE_LUT_12TO8
- CAMERA_FEATURE_LUT_10TO8

Above values will be queried for by the following function:

```
Camera_isCameraFeature(Camera_handle hCamera, CAMERA_FEATURE Feature);
```

If the function returns ‘true’ then it is save to use appropriate LUT size. Otherwise a trial for upload/download will fail.

A LUT can be enabled/disabled and the current mode can be retrieved by the following functions:

```
Camera_setLookupTableMode(Camera_handle hCamera,
                          LUT_MODE LUTMode);
Camera_getLookupTableMode(Camera_handle hCamera,
                          LUT_MODE *ProgrammedLUTMode);
```

There are functions provided that are implemented based of a LUT:

- digital gain
- digital offset

- gamma

The following functions allow for generating a LUT in order to control the above features:

```
Camera_setGammaCorrection(Camera_handle Camera,
                      float GammaCorrection);
Camera_setGammaCorrectionExt(Camera_handle Camera,
                         float GammaCorrection,
                             float DigitalGain,
                             float DigitalOffset);
```

First function will apply a ‘gamma’ in the range from 0.4 to 2.5 to all images and second function will additionally implement a digital gain and offset based on LUT.

• SVGigE FilterDriver installation/de-installation

The SDK provides for functionality to install and to de-install a SVGigE FilterDriver:

```
SVGigE_installFilterDriver(char *PathToDriverPackage,
                       char *FilenameINF,
                       char *FilenameINF_m);
```

The 3 arguments will usually get a NULL pointer assigned. This provides to taking a default path and default names of the INF files involved in driver installation.
For de-installation the following function will be called:

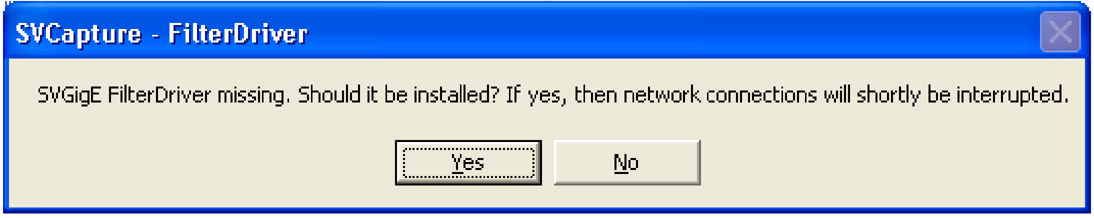
```
SVGigE_uninstallFilterDriver();
```

A SVGigE FilterDriver will be searched and subsequently de-installed from the system.

18 Appendix D

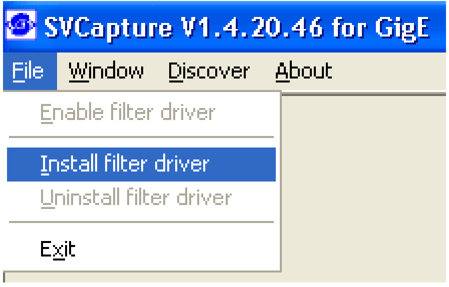
• Automated SVGigE Filter Driver installation

SVCapture checks on start-up, whether a SVGigE filter driver is installed. If this is not the case then the user will be asked whether a filter driver should be installed automatically. An information is given that network connections will shortly be interrupted:

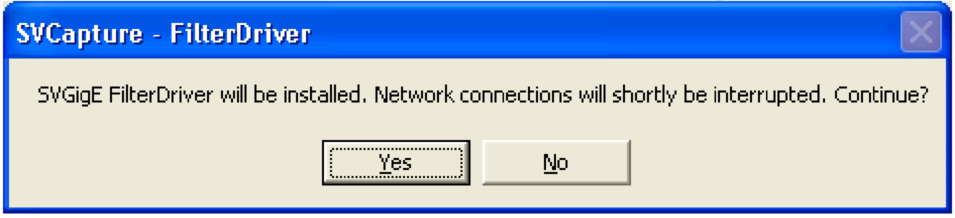


SVGigE FilterDriver missing. Should it be installed? If yes, then network connections will shortly be interrupted.

Alternatively, a FilterDriver installation can be started from menu:

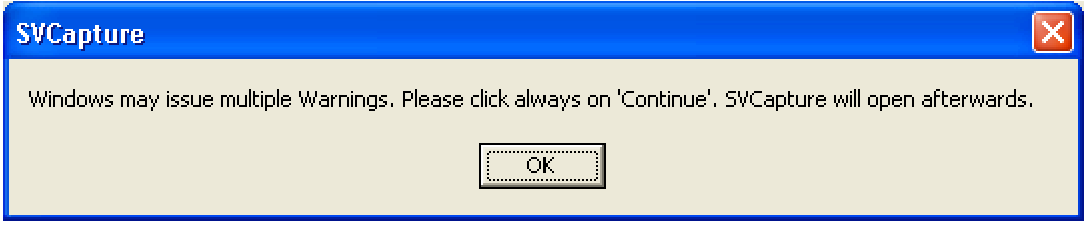


The user will be informed about a short network interruption:

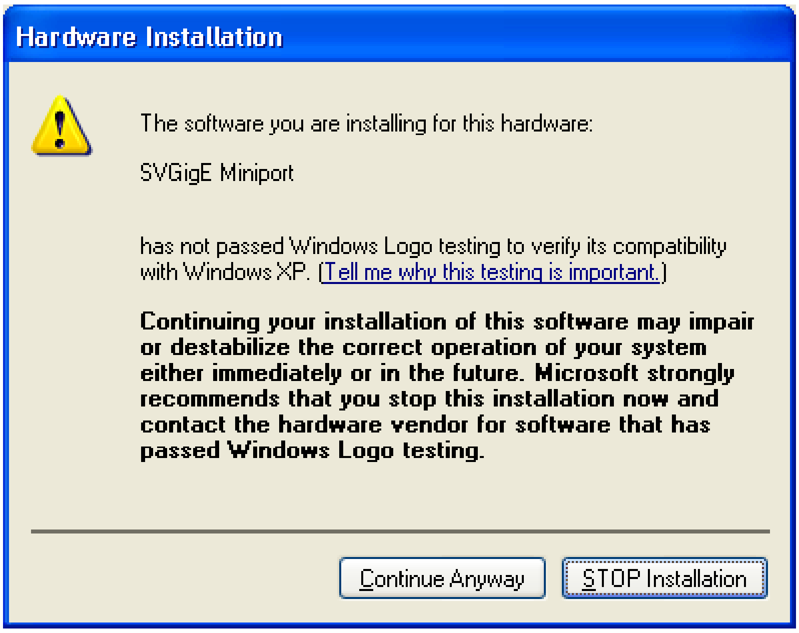


SVGigE FilterDriver will be installed. Network connections will shortly be interrupted. Continue?

For both variants, the user will be asked for confirming warning dialogs:



Windows may issue multiple Warnings. Please click always on 'Continue'. SVCapture will open afterwards.



Hardware Installation

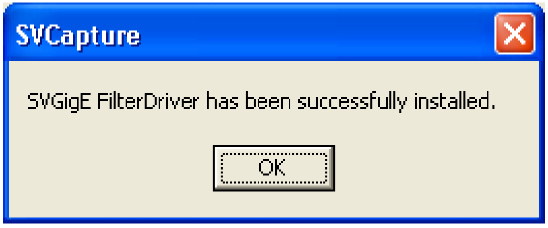
The software you are installing for this hardware:

SVGigE Miniport

has not passed Windows Logo testing to verify its compatibility with Windows XP. (Tell me why this testing is important.)

Continuing your installation of this software may impair or destabilize the correct operation of your system either immediately or in the future. Microsoft strongly recommends that you stop this installation now and contact the hardware vendor for software that has passed Windows Logo testing.

After some time, the driver will be installed and a message informs about success:

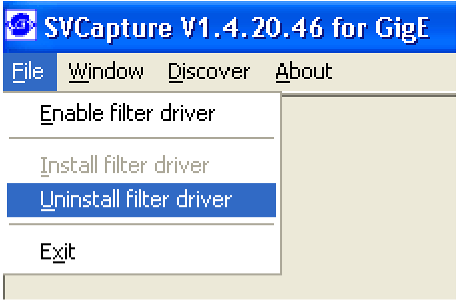


SVCapture

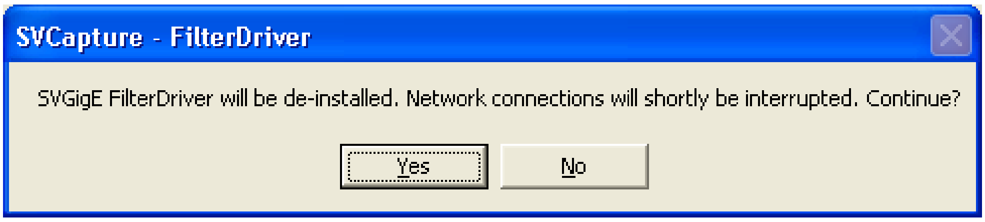
SVGigE FilterDriver has been successfully installed.

- **Automated SVGigE Filter**Driver **de**-installation

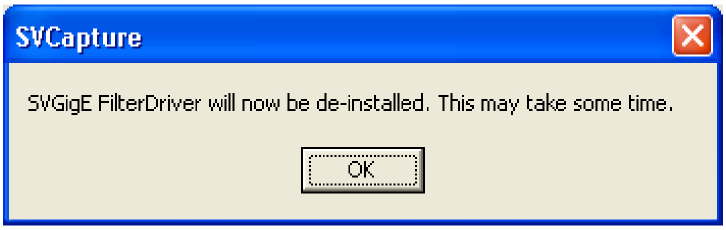
Please go to the File menu and select “Uninstall filter driver” for starting an automated SVGigE FilterDriver de-installation. Before, the filter driver has to be disabled with the “Disable filter driver” menu entry if needed.



The user will be informed about a short network interruption during de-installation of the filter driver:

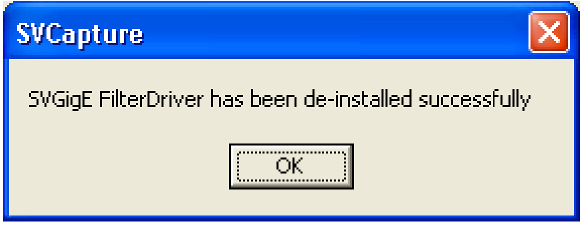


SVGigE FilterDriver will be de-installed. Network connections will shortly be interrupted. Continue?



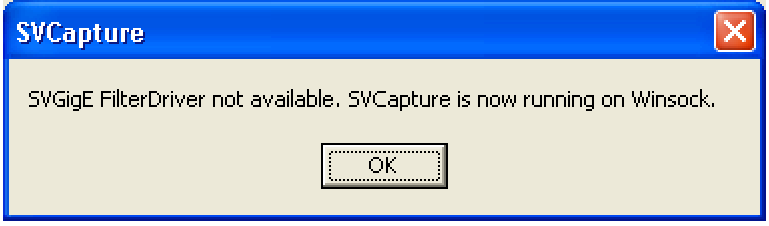
SVGigE FilterDriver will now be de-installed. This may take some time.

After some time, a message will inform about successful de-installation:



SVGigE FilterDriver has been de-installed successfully

Subsequently, a Winsock transport layer is loaded in order to operate cameras without a filter driver. This mode of operation is not recommended.



SVGigE FilterDriver not available. SVCapture is now running on Winsock.

NOTE: The SVGigE FilterDriver should always be installed. De-installation should only take place when the SVCam GigE SDK is intended to be de-installed on a PC.

Manual SVGigE Filter Driver installation

18.1 Installation Instruction for Filter Driver installation

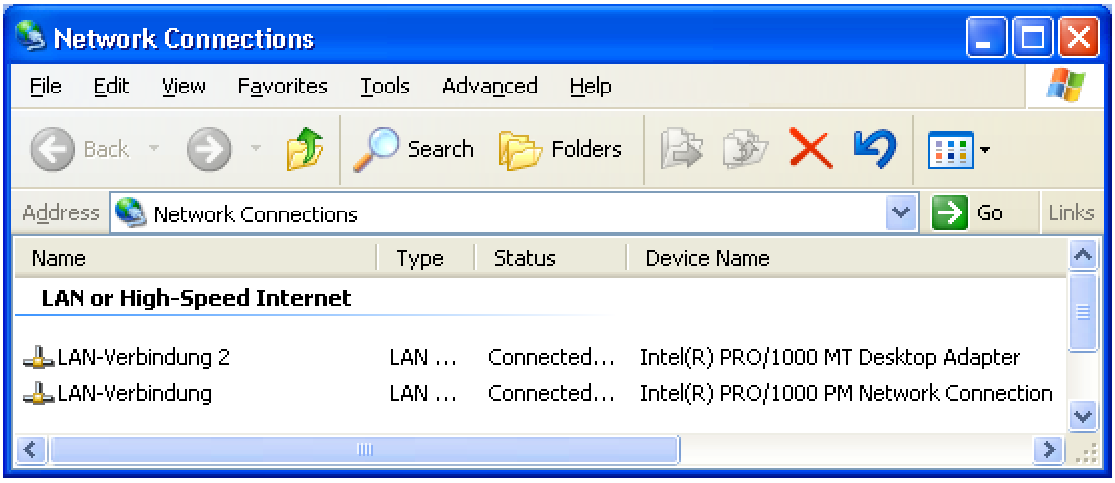
In order to use the SVS GigE driver it has to be installed first. The setup will copy the driver to disk but will not install it. On a standard Windows XP/2000 system this has to be done by the following steps:

IMPORTANT: Usually the driver will be copied by the setup procedure to the following location:

C:\Program Files\SVS-VISTEK GmbH\SVS GigE SDK 1.2.1 \SVS GigE FilterDriver

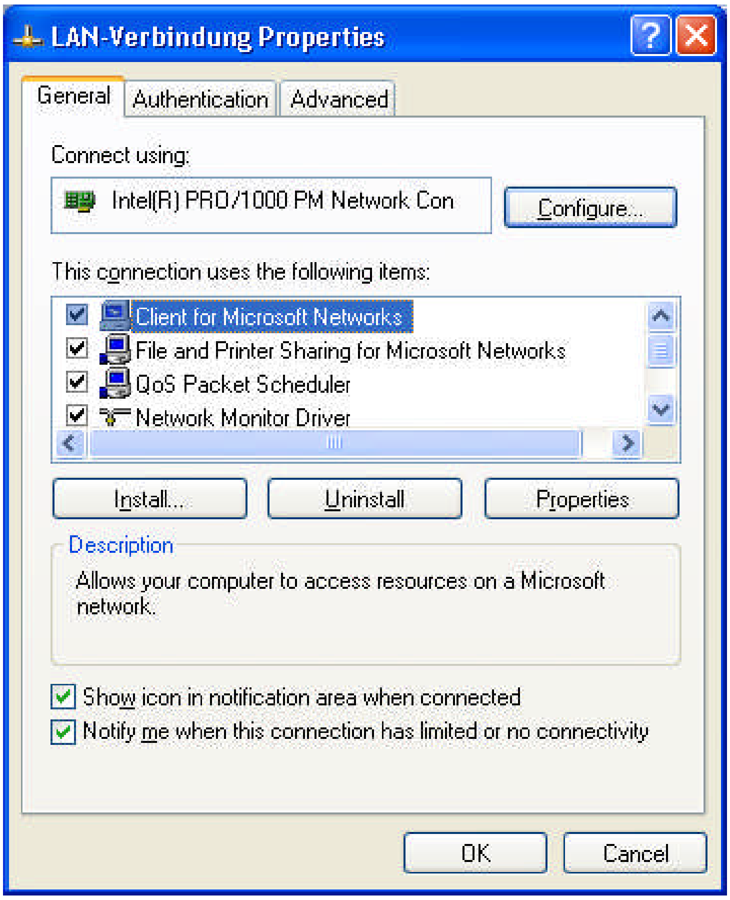
Step-by-step

Step 1: Open the Network Connections dialog



Name	Type	Status	Device Name
LAN or High-Speed Internet			
LAN-Verbindung 2	LAN ...	Connected...	Intel(R) PRO/1000 MT Desktop Adapter
LAN-Verbindung	LAN ...	Connected...	Intel(R) PRO/1000 PM Network Connection

Step 2: Open the Properties dialog for LAN connections and click on "Install…"



LAN-Verbindung Properties

General | Authentication | Advanced

Connect using:

Intel(R) PRO/1000 PM Network Con

Configure...

This connection uses the following items:

☑ Client for Microsoft Networks
☑ File and Printer Sharing for Microsoft Networks
☑ QoS Packet Scheduler
☑ Network Monitor Driver

Install... | Uninstall | Properties

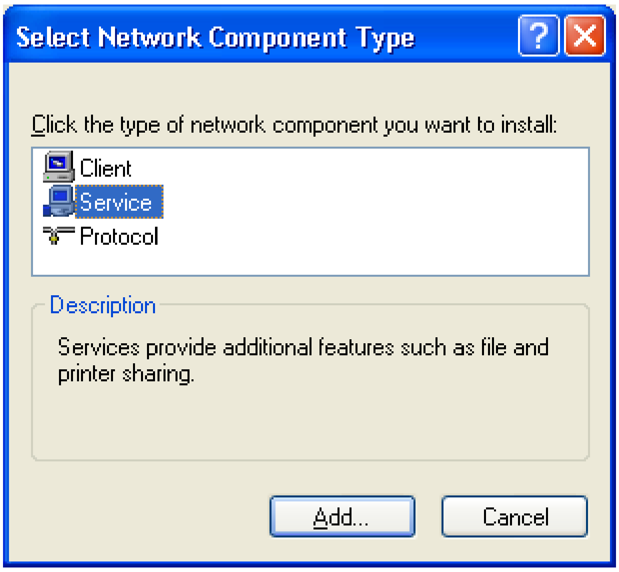
Description

Allows your computer to access resources on a Microsoft network.

☑ Show icon in notification area when connected
☑ Notify me when this connection has limited or no connectivity

OK | Cancel

Step 3: In the "Select Network Component Type" dialog select the Service option and click on "Add…"



Select Network Component Type

Click the type of network component you want to install:

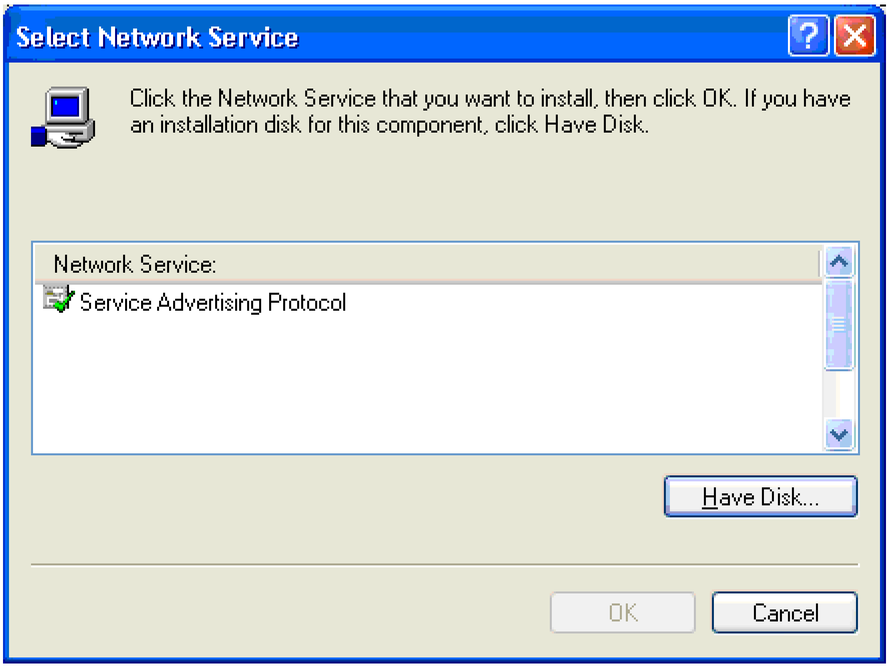
Client
Service
Protocol

Description

Services provide additional features such as file and printer sharing.

Add... | Cancel

Step 4: In the “Select Network Service” dialog click on “Have Disk…”



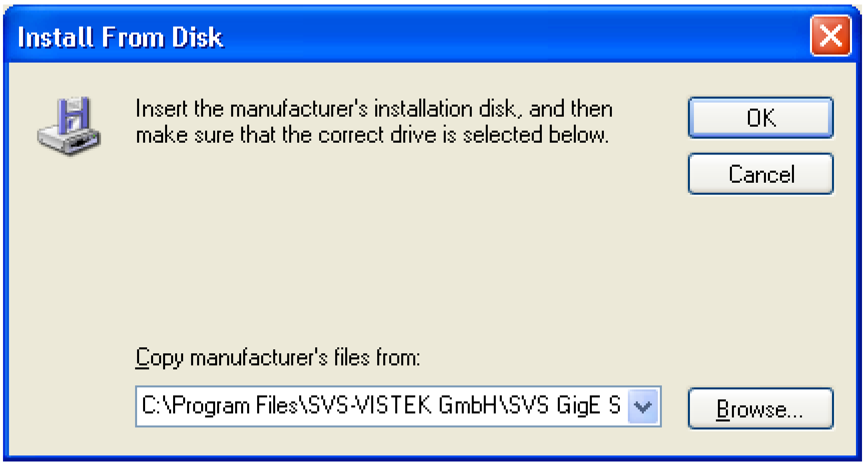
Select Network Service

Click the Network Service that you want to install, then click OK. If you have an installation disk for this component, click Have Disk.

Network Service:

Service Advertising Protocol

Step 5: *In the “Install From Disk” dialog the folder where the driver files are located needs to be navigated to or entered directly into the files field.*



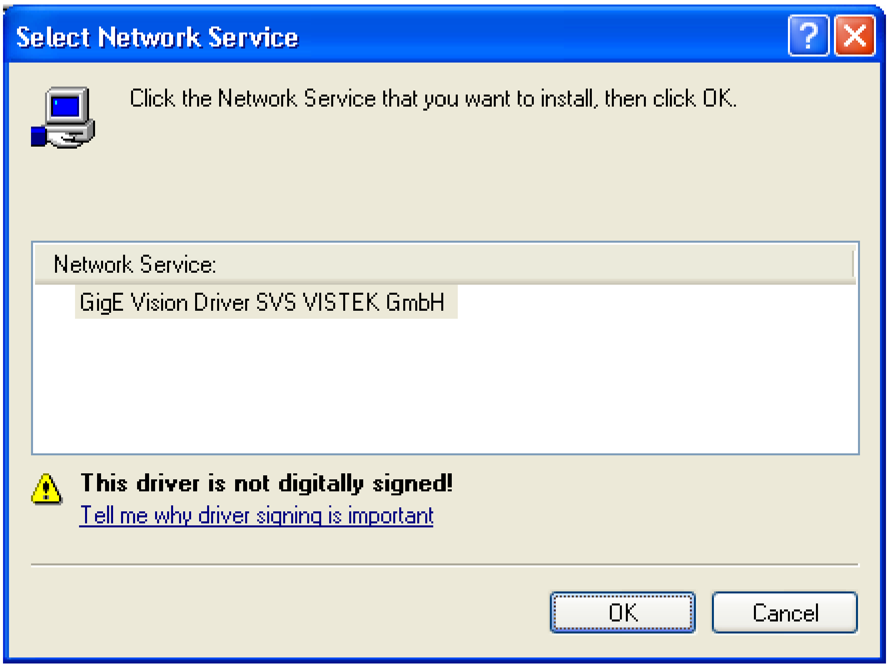
Install From Disk

Insert the manufacturer's installation disk, and then make sure that the correct drive is selected below.

Copy manufacturer's files from:

C:\Program Files\SVS-VISTEK GmbH\SVS GigE S

Step 6: After clicking on OK the "Select Network Service" dialog will display a message that the driver is not digitally signed. Click on OK.



Click the Network Service that you want to install, then click OK.

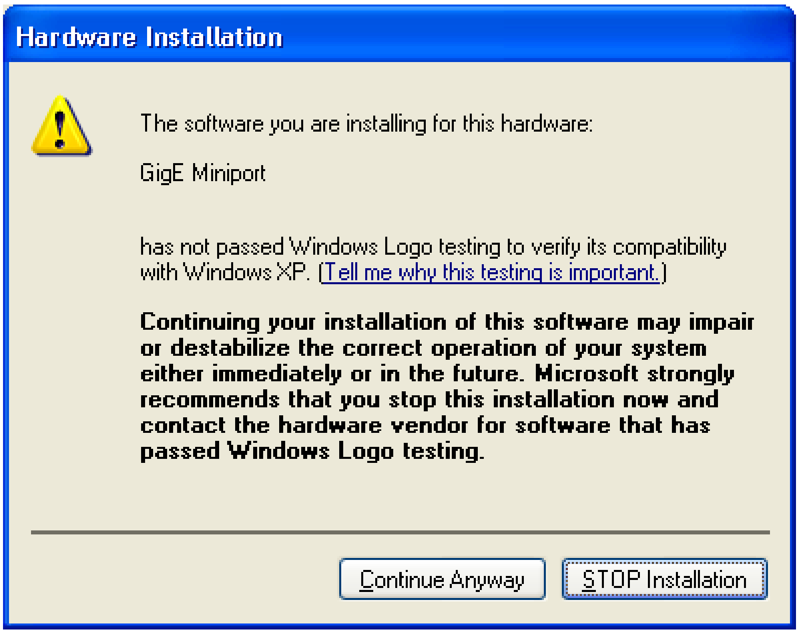
Network Service:

GigE Vision Driver SVS VISTEK GmbH

This driver is not digitally signed!

Tell me why driver signing is important

Step 7: Because the driver not being digitally signed, a dialog will appear with a warning. Answer with "Continue Anyway" as often as needed.



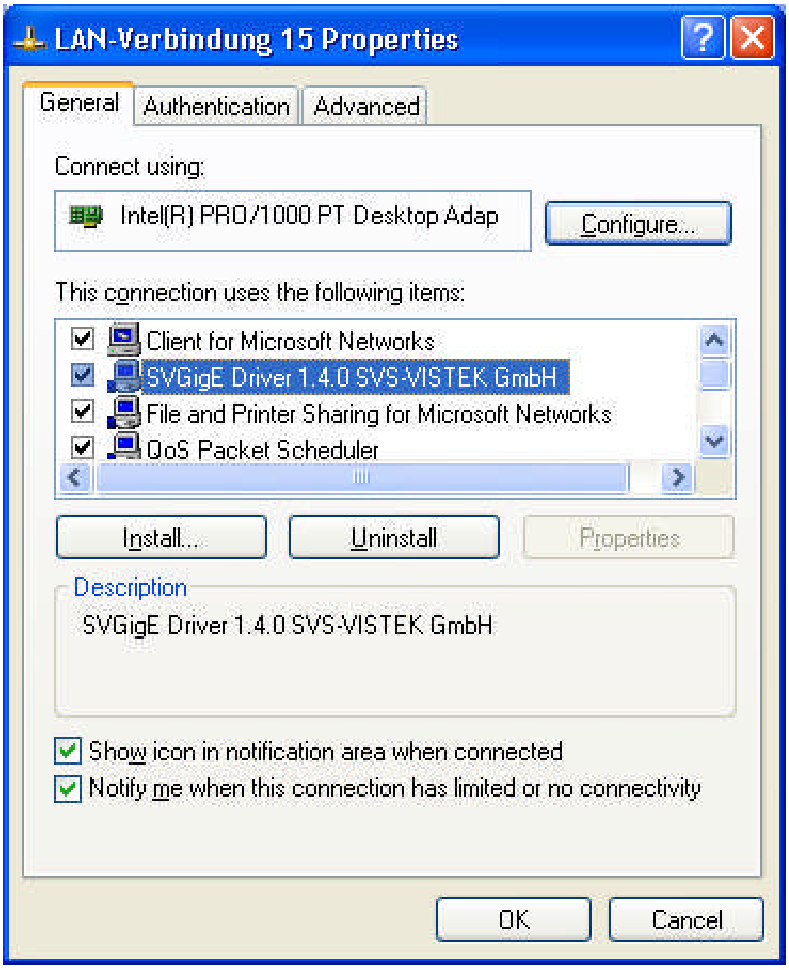
The software you are installing for this hardware:

GigE Miniport

has not passed Windows Logo testing to verify its compatibility with Windows XP. (Tell me why this testing is important.)

Continuing your installation of this software may impair or destabilize the correct operation of your system either immediately or in the future. Microsoft strongly recommends that you stop this installation now and contact the hardware vendor for software that has passed Windows Logo testing.

Step 8: After confirming the prior dialogs the driver is installed and should show up as a new item in the "LAN Connections Properties" dialog.



Connect using:

Intel(R) PRO/1000 PT Desktop Adap

This connection uses the following items:

☑ Client for Microsoft Networks
☑ SVGigE Driver 1.4.0 SVS-VISTEK GmbH
☑ File and Printer Sharing for Microsoft Networks
☑ QoS Packet Scheduler

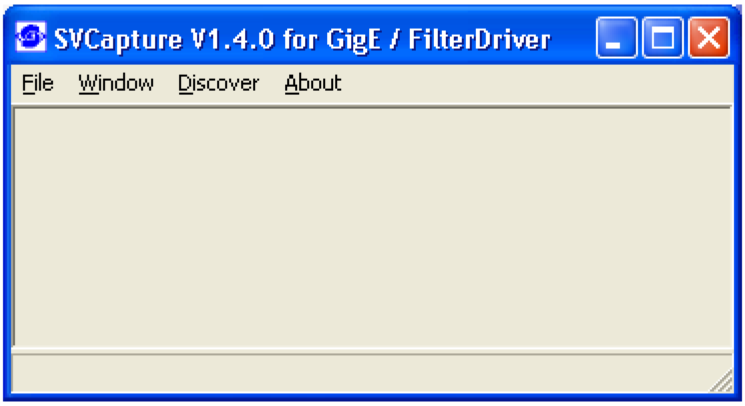
Description

SVGigE Driver 1.4.0 SVS-VISTEK GmbH

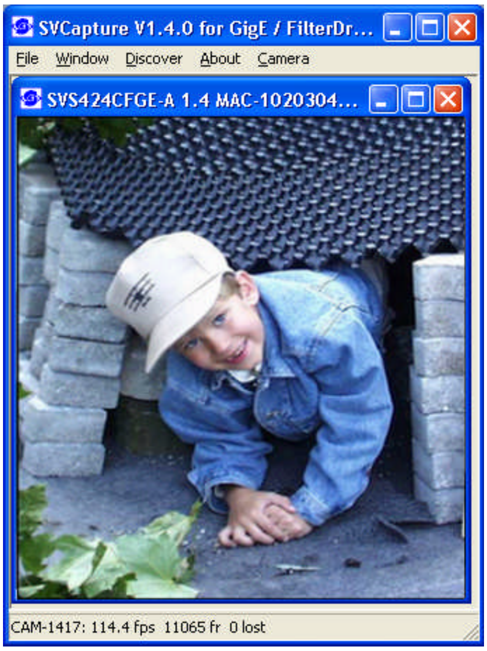
☑ Show icon in notification area when connected
☑ Notify me when this connection has limited or no connectivity

Step 9: The dialog can be closed and the "GigE Vision Driver SVS-VISTEK GmbH" is ready for use.

The main dialog in SVCapture will show an extension "FilterDriver" which indicates that the driver will be used next time when opening a camera. If this extension is not shown it can be activated in the "File | Enable filter driver" menu item.



Step 10: Follow point 5 "Displaying Live Images" in this documentation for starting an image view. If the driver was successfully installed and works properly the externsion "FilterDriver" will remain in the main dialog's caption. Otherwise it will disappear and the SVCapture program falls back to operating the GigE Vision camera over WinSock.


SVCapture V1.4.0 for GigE / FilterDr...
File
Window
Discover
About
Camera
SVS424CFGE-A 1.4 MAC-1020304...

CAM-1417: 114.4 fps 11065 fr 0 lost

19 Appendix E

19.1 PC and OS Requirements

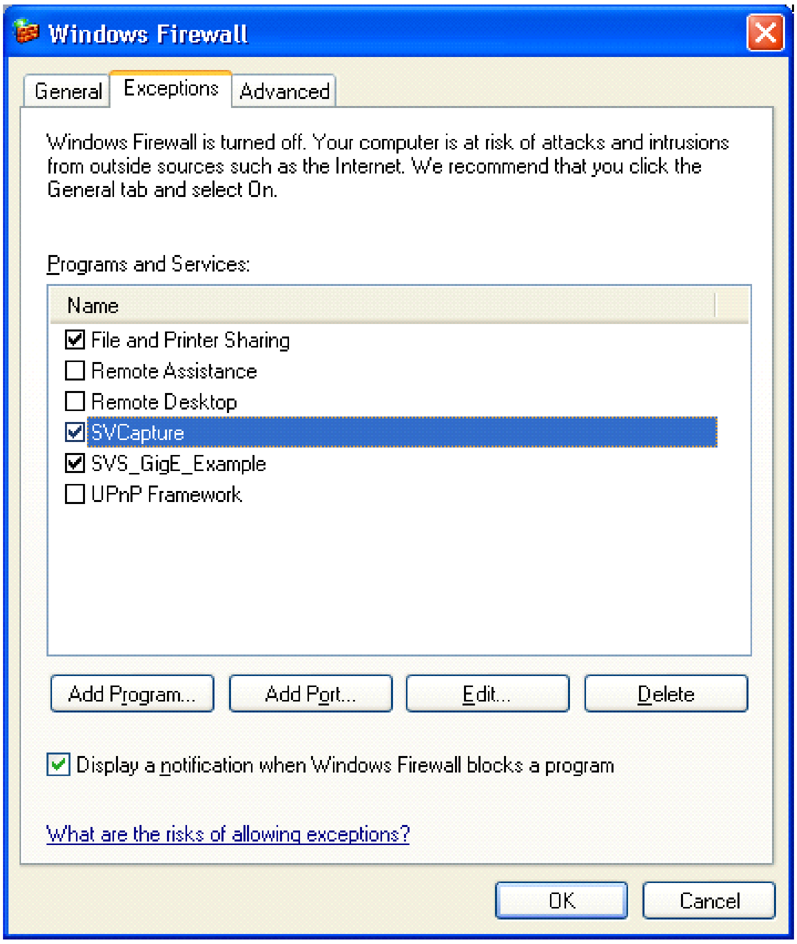
It is recommended to use a PC with a Pentium P4 processor at **2.4 GHz** or higher. The camera is working also on lower frequencies but it might not deliver the full frame rate in those cases. The operation system must be XP or Win 7. **We don’t** support WIN 98/ NT/2000 or older Operation systems. A Linux driver is available, too. For VXworks a source code for driver development can be offered on request. Interfacinging must be developed by the experienced user.

19.2 Firewall considerations

The GigE Vision standard defines a communication based on UDP packets between a GigE Vision compliant camera and a host PC. Often PC firewalls are adjusted to be restrictive and not to allow for sending UPD packets from the outside of the PC to a port that has been opened by an application

Sometimes the firewall settings have been adjusted during installation of the operating system or afterwards such that the user will be asked if the UDP data stream should be enabled once a camera starts sending UDP packets to the PC. When clicking OK in response to that question the application will be added to a list of exceptions. A firewall will let those applications communicate with network devices like a GigE Vision camera over the network. In case the Windows installation has been adjusted such that the user will not be asked when an outside device starts sending network packets to an application then that application has to be enabled manually in order to achieve the same as before.An appropriate dialog can be opened for example when right-clicking on the network icon in the tray and selecting “Change Windows.

After clicking on the “Add Program…” button the GigE Vision application can be searched for in a file explorer and it will be added to the list of Exceptions in the above dialog. Henceforth the communication to the GigE Vision compliant device will be enabled.



Windows Firewall

General | Exceptions | Advanced

Windows Firewall is turned off. Your computer is at risk of attacks and intrusions from outside sources such as the Internet. We recommend that you click the General tab and select On.

Programs and Services:

Name
☑ File and Printer Sharing
□ Remote Assistance
□ Remote Desktop
☑ SVCapture
☑ SVS_GigE_Example
□ UPnP Framework

☑ Display a notification when Windows Firewall blocks a program

What are the risks of allowing exceptions?

20 Appendix F

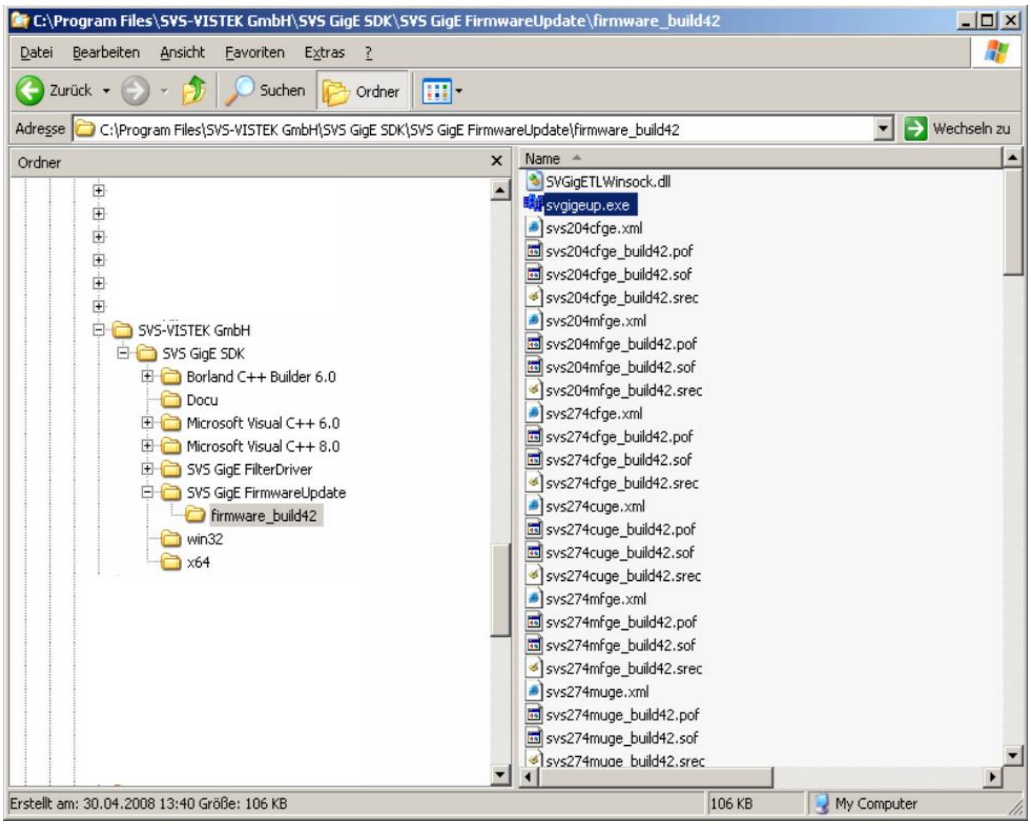
20.1 Firmware-Update with “GigE Update Tool.exe“ program

Note: In order to work with the latest driver and SVCapture it might be necessary to UPDATE the firmware inside the camera. This can be done in the field. It is usually not necessary to send the camera in.

⚠ **Warnings:**

- Camera must have power
- Camera must be access able from PC via network
- Camera SHOULD NOT be used during programming procedure
- Camera SHOULD NOT be disconnected during programming procedure or be disconnected from network
- Programming software SHOULD NOT be stopped during procedure.
- It might take up to some minutes until it is completed.

Step1: Locate „svgigeup.exe“ using Windows Explorer



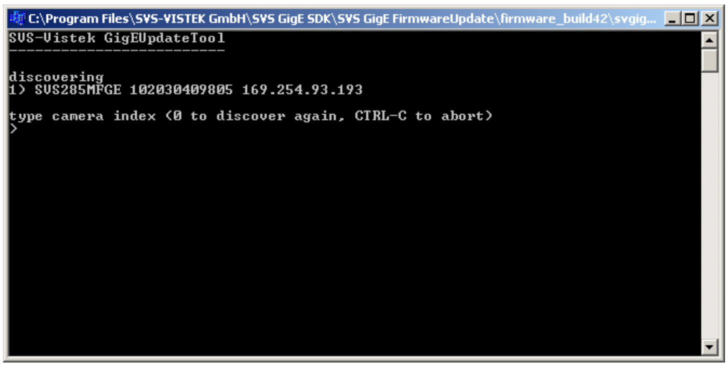
C:\Program Files\SVS-VISTEK GmbH\SVS GigE SDK\SVS GigE FirmwareUpdate\firmware_build42

- SVS-VISTEK GmbH
  - SVS GigE SDK
    - Borland C++ Builder 6.0
    - Docu
    - Microsoft Visual C++ 6.0
    - Microsoft Visual C++ 8.0
    - SVS GigE FilterDriver
    - SVS GigE FirmwareUpdate
      - firmware_build42
    - win32
    - x64

SVGigETLWinsock.dll
svgigeup.exe
svs204cfge.xml
svs204cfge_build42.pof
svs204cfge_build42.sof
svs204cfge_build42.srec
svs204mfge.xml
svs204mfge_build42.pof
svs204mfge_build42.sof
svs204mfge_build42.srec
svs274cfge.xml
svs274cfge_build42.pof
svs274cfge_build42.sof
svs274cfge_build42.srec
svs274cuge.xml
svs274cuge_build42.pof
svs274cuge_build42.sof
svs274cuge_build42.srec
svs274mfge.xml
svs274mfge_build42.pof
svs274mfge_build42.sof
svs274mfge_build42.srec
svs274muge.xml
svs274muge_build42.pof
svs274muge_build42.sof
svs274muge_build42.srec

Erstellt am: 30.04.2008 13:40 Größe: 106 KB | 106 KB | My Computer

Step 2 Execute svgigeup.exe

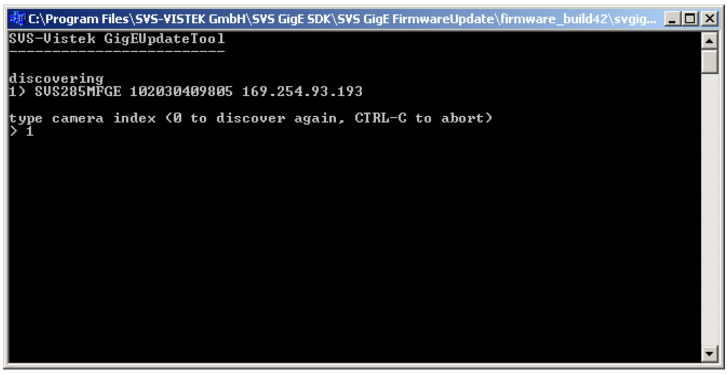

C:\Program Files\SVS-VISTEK GmbH\SVS GigE SDK\SVS GigE FirmwareUpdate\firmware_build42\svgig...

```
SVS-Vistek GigEUpdateTool
-------------------------

discovering
1) SVS285MFGE 102030409805 169.254.93.193

type camera index (0 to discover again, CTRL-C to abort)
>
```

Step 3 Select Camera (e.g. 1 and press Enter)

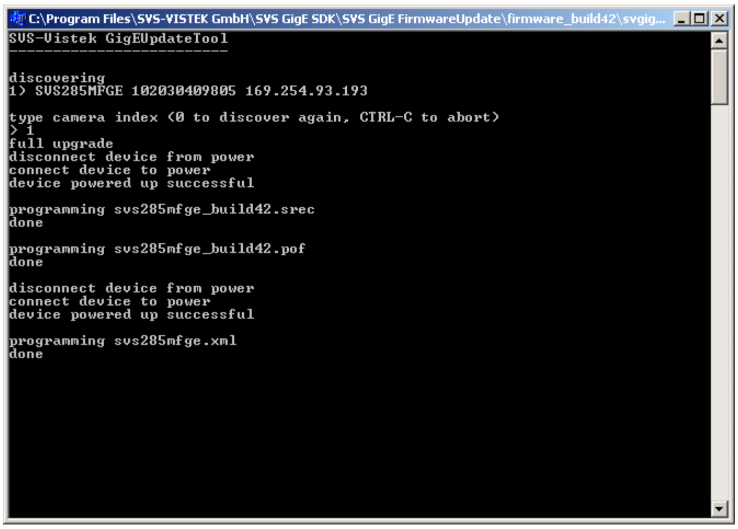

C:\Program Files\SVS-VISTEK GmbH\SVS GigE SDK\SVS GigE FirmwareUpdate\firmware_build42\svgig...

```
SVS-Vistek GigEUpdateTool
-------------------------

discovering
1) SVS285MFGE 102030409805 169.254.93.193

type camera index (0 to discover again, CTRL-C to abort)
> 1
```

Step 4: Wait and follow instructions


C:\Program Files\SVS-VISTEK GmbH\SVS GigE SDK\SVS GigE FirmwareUpdate\firmware_build42\svgig...

```
SVS-Vistek GigEUpdateTool
-------------------------

discovering
1) SVS285MFGE 102030409805 169.254.93.193

type camera index (0 to discover again, CTRL-C to abort)
> 1
full upgrade
disconnect device from power
connect device to power
device powered up successful

programming svs285mfge_build42.srec
done

programming svs285mfge_build42.pof
done

disconnect device from power
connect device to power
device powered up successful

programming svs285mfge.xml
done
```

YOU ARE DONE

After programming it is recommended to start the camera again. This means you have to disconnect power and – after some seconds – power up again.

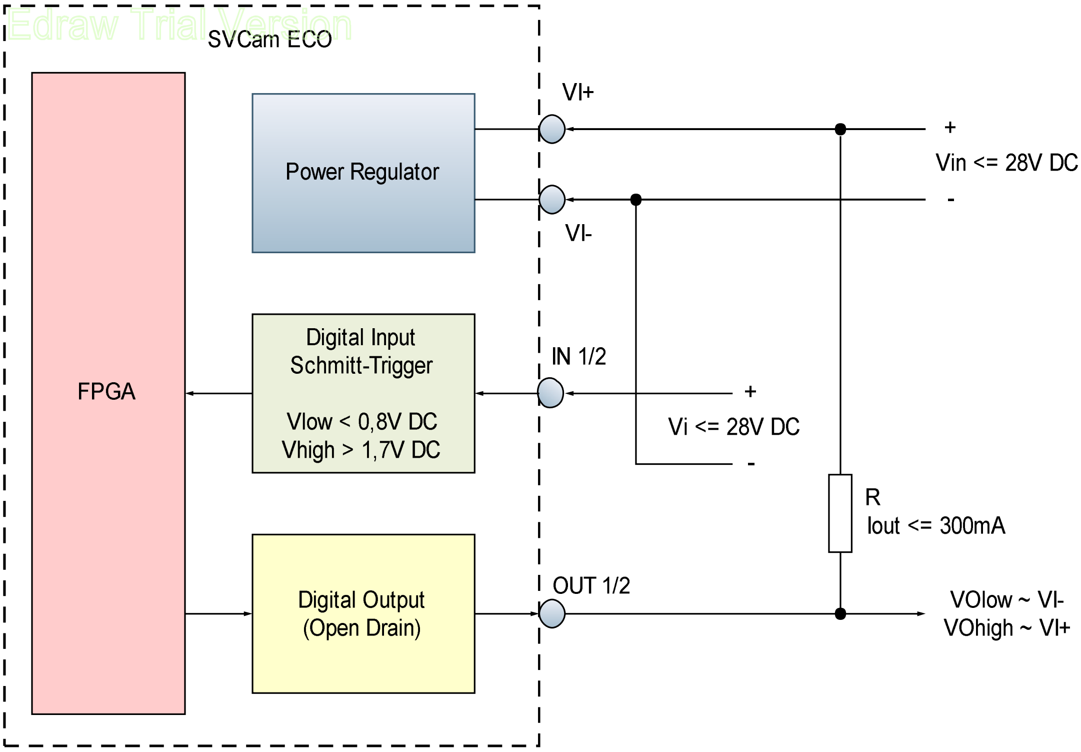
21 Appendix G
Basic circuits for Hardware Interfacing

As example: Use Hardwaretrigger connection on “IN1”

For Strobe use “OUT1”

Use GND to make connection complete

TTL input circuit/ output (i.e strobe)



22 Appendix H Safety Instructions

- In order to avoid personal injuries, or property damages against him/her or people around him/her by using this product correctly.
- Prior to use, read this section carefully to fully understand its instructions for correct use.

22.1 Definition of markings

The meaning of each mark used in this instruction manual is given below

DANGER	This mark warns the user that improper use, indicated with this mark, may cause death or severe personal injuries against the user or people around him/her.
CAUTION	This mark warns the user that improper use, indicated with this mark, may cause personal injuries (*1) or material damages (*2) against the user or people around him/her.

Notes
*1: Personal injuries mean wounds, burns, electric shocks, and others for which the person injured need neither to be hospitalized nor to be cared for the long term
*2: Material damages mean any direct or consequential damages related to property or material loss

🛇	This mark indicates what the user SHOULD NOT DO. The details of things, which the user should not do, are described next to this mark.
●	This mark indicates what the user MUST DO. The details of things, which the user must do, are described next to this mark.
◇	This mark indicates that the user must be alert against a possible DANGER. The details of the DANGER, which the user must be aware of are described next to this mark.
△	This mark indicates that the user is given a CAUTION against possible hazards. The details of the CAUTION, which the user must be aware of, are described next to this mark.

● Handling Precautions

DANGER	
●	If any overheating sign is observed, discontinue the use immediately. In the event that smoke, smell, or any other overheating sign is observed, turn the power switch of the system OFF immediately and remove the power cable(s) from the system connectors, like computer, camera, lightning. Do NOT try to continue to use the system. To do so in spite of clear signs of malfunction invites a fire, an electric shock hazard, or a serious damage to the system components. In such case, contact for repair service us or our dealer/distributor, from which you purchased the system.
●	If any malfunctioning sign is observed, discontinue the use immediately. Do NOT try to use the system when it is malfunctioning. (Ex. No images on the monitor) In the event of malfunction, turn the power switch of the system OFF immediately and remove the system power cables from the system components connectors. In such case, contact for repair service us or our dealer/distributor from which you purchased the system.
●	If any foreign object gets into the system components, discontinue the use immediately. In the event that liquids, small particles, or any other foreign objects get into the system components, do NOT try to continue to use the system. To do so invites a fire or an electric shock hazard. In that case, turn the power switch of the system components OFF immediately and remove the system power cables from the system components connectors. After that, contact us or our dealer/distributor from which you purchased the camera for repair service/technical advice.
●	Do NOT disassemble the system components. Do NOT attempt to pull apart; repair, or modify the system components on your own. To do so leads to a fire or an electric shock accident. Contact us or the dealer/distributor from which you purchased the camera for repair/modifacation.
●	Do NOT supply any power than specified. The system components are designed to work only under specified voltage. Do NOT attempt to drive the system components with the power other than specified. Operating the system components under power other than specified invites a fire or a electric shock hazard.
●	Do NOT use the system components in a high-humidity environment.

	Do NOT place the system components near a humidifier, or in other high-humidity environment. To do so may cause a fire or an electric shock accident.
⚠ **CAUTION**	
⚠	If the system components are operated in the electromagnetic field, there may be cases where noises (vertical, horizontal, or oblique stripes) may appear to a video output or causes other malfunctions. In that case, take preventive measures on the electromagnetic-wave generating source so that the system components do not receive the interference by the electromagnetic- wave. Take extra precautions against electromagnetic-wave-interference if the system components are used with a servomotor, inverter, or other electromagnetic-wave-generating equipment.
⚠	Avoid giving a strong shock against the system components. If your system components are used in the system where the connector are subjected to strong repetitive shocks, the connector are possible to break down. If you intend to use your system components in such a situation, make sure to use an optional-connector-fixing-hardware to connect the connector-plug to the system components body.
⚠	When the system components are not in use, put a lens or a lens-cap onto the cam head so that the image pickup plane of CCD is protected from dust, foreign object, or other flaw-causing object. If the glass plane (image pickup plane) gets dirty, clean it with a cotton swab. When it needs to be cleaned with a cleaner, be sure NOT to use any organic solvent other than ethyl alcohol. Do not clean the other system components. In such a case call us or the dealer/distributor from which you purchased the system components for cleaning or cleaning instructions. As a countermeasure against condensation, when the system components are moved from a warm place to a cold place, take appropriate precautions to prevent condensation from forming on the system components.
⚠	Do not pull strongly the any cable nor swing it. The stress from pulling or swinging may cause damage in the coating of the cable, or breaks in the inside wires.
⚠	Avoid short-circuiting its signal output. Otherwise, the system components might be damaged.
⚠	If too much amount of light, (= the incoming light amount of 100 times or greater in comparison with standard light) enters CCD image pickup plane, video output might not be obtained. In such a case, take measures to reduce the amount of incoming light.

⚠	Do NOT expose the camera to intensive light (sunlight, etc.) to prevent its inner CCD from getting damaged.
⚠	When mounting a lens, take extra caution so that the lens is not tilled, nor does flaw exist at the lens-mount-screw part. Also check to confirm that neither dirt nor other foreign object is put inside. Improper mounting might cause the parts to become locked.
◇ **DANGER**	
🛇	Do NOT use any optional unit other than manufacturer-supplied one. We disclaim any responsibility for damages or losses incurred by user due to the use of unauthorized / unofficial option units supplied by a third party

22.2 Restricted for Use

In case malfunction of this equipment (e.g. video output cut-off) can be expected to lead to significant accident, avoid using this equipment for such system integration use.

22.3 Cases for Indeminity (Limited Warranty)

We shall be exempted from taking responsibility and held harmless for damages or losses incurred by user in the following cases. In case damages or losses are caused by fire, earthquake, or other acts of Gods, the act by third party, misuse by the user deliberately or erroneously, use under extreme operating conditions.

- In case indirect, additional, consequential damages (loss of expected interest, suspension of business activities) are incurred as results of malfunction of non-function of the equipment, we shall be exempted from assuming responsibility for such damages.
- In case damages or losses are caused by incorrect use, which is not in line with the instructions in this instruction manual.
- In case damages or losses are caused by malfunction resulting from band connection with other equipment.
- In case damages or losses are caused by repair or modification done by the user.

IMPORTANT SAFETY INSTRUCTIONS

1. This device is designed and guaranteed to work under the temperature range of - - 10 through 45 degree C. Avoid using the equipment beyond that limits.
2. Do NOT expose the camera's image-pickup-plane to sunlight or other intense light directly. Its inner CCD (charge-coupled device) might be damaged.
3. Do NOT exposure all system components to sunlight or other intensive

light (UV, IR).

4. In the event that any abnormal condition is observed, turn the power switch OFF immediately. Do NOT try to continue to use the system components. To do so in spite of clear signs of malfunction invites a fire, an electric shock hazard, or any other serious damage to the system components. In such case, contact us or our dealer/distributor that you purchased the system components from for repair service.
5. To clean the body of this equipment, make sure to turn all power switches OFF first. To remove stubborn stains, use a soft cloth soaked in diluted acid free detergent. After that, clean with a dry cloth.
6. In case the image-pickup-plane should be settled with fine dust, dirt, or scratched, ask our distributor for technical advice