Basler scout

USER'S MANUAL FOR GigE VISION CAMERAS

Document Number: AW000119
Version: 05   Language: 000 (English)
Release Date: 08 June 2007

BASLER VISION TECHNOLOGIES

For customers in the U.S.A.

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.

You are cautioned that any changes or modifications not expressly approved in this manual could void your authority to operate this equipment.

The shielded interface cable recommended in this manual must be used with this equipment in order to comply with the limits for a computing device pursuant to Subpart J of Part 15 of FCC Rules.

For customers in Canada

This apparatus complies with the Class A limits for radio noise emissions set out in Radio Interference Regulations.

Pour utilisateurs au Canada

Cet appareil est conforme aux normes Classe A pour bruits radioélectriques, spécifiées dans le Règlement sur le brouillage radioélectrique.

Life Support Applications

These products are not designed for use in life support appliances, devices, or systems where malfunction of these products can reasonably be expected to result in personal injury. Basler customers using or selling these products for use in such applications do so at their own risk and agree to fully indemnify Basler for any damages resulting from such improper use or sale.

Warranty Note

Do not open the housing of the camera. The warranty becomes void if the housing is opened.

Contacting Basler Support Worldwide

Europe:

Basler AG
An der Strusbek 60 - 62
22926 Ahrensburg
Germany

Tel.: +49-4102-463-500
Fax.: +49-4102-463-599

vc.support.europe@baslerweb.com

Americas:

Basler, Inc.
855 Springdale Drive, Suite 160
Exton, PA 19341
U.S.A.

Tel.: +1-877-934-8472
Fax.: +1-877-934-7608

vc.support.usa@baslerweb.com

Asia:

Basler Asia Pte. Ltd
8 Boon Lay Way
03 - 03 Tradehub 21
Singapore 609964

Tel.: +65-6425-0472
Fax.: +65-6425-0473

vc.support.asia@baslerweb.com

www.basler-vc.com

Table of Contents

1 Specifications, Requirements, and Precautions

This section lists the camera models covered by the manual. It provides the general specifications for those models and the basic requirements for using them.

This section also includes specific precautions that you should keep in mind when using the cameras. We strongly recommend that you read and follow the precautions.

1.1 Models

The current Basler scout GigE Vision camera models are listed in the top row of the specification tables on the next pages of this manual. The camera models are differentiated by their sensor size, their maximum frame rate at full resolution, and whether the camera's sensor is mono or color.

The scout GigE Vision camera models are available in the following housing variants:

- standard housing
- 90° head housing

The housing variants other than the standard housing are appended to the camera's name, e.g. scA640-70gm/gc 90° head.

Unless otherwise noted, the material in this manual applies to all of the camera models listed in the tables. Material that only applies to a particular camera model or to a subset of models, such as to color cameras or a specific housing variant only, will be so designated.

1.2 General Specifications

Specification	scA640-70gm/gc	scA640-74gm/gc	scA750-60gm/gc
Sensor Size (H x V pixels)	gm: 659 x 494 gc: 658 x 492	gm: 659 x 494 gc: 658 x 492	gm: 752 x 480 gc: 750 x 480
Sensor Type	Sony ICX424 AL/AQ	Sony ICX414 AL/AQ	Micron MT9V022
	Progressive scan CCD		Progressive scan CMOS
Optical Size	1/3"	1/2"	1/3"
Pixel Size	7.4 µm x 7.4 µm	9.9 µm x 9.9 µm	6.0 µm x 6.0 µm
Max. Frame Rate (at full resolution)	70 fps	79 fps	64.9 fps
Mono/Color	All models available in mono or color		
Data Output Type	Fast Ethernet (100 Mbit/s) or Gigabit Ethernet (1000 Mbit/s)		
Pixel Data Formats	Mono Models:		
	Mono 8 (= DCAM Mono 8) Mono 16 (= DCAM Mono 16) Mono 12 Packed YUV 4:2:2 Packed (= DCAM YUV 4:2:2) YUV 4:2:2 (YUYV) Packed		Mono 8 (= DCAM Mono 8) YUV 4:2:2 Packed (= DCAM YUV 4:2:2) YUV 4:2:2 (YUYV) Packed
	Color Models:		
	Mono 8 (= DCAM Mono 8) Bayer BG 8 (= DCAM Raw 8) Bayer BG 16 (= DCAM Raw 16) Bayer BG 12 Packed YUV 4:2:2 Packed (= DCAM YUV 4:2:2) YUV 4:2:2 (YUYV) Packed		Mono 8 (= DCAM Mono 8) Bayer RG 8 (= DCAM Raw 8) YUV 4:2:2 Packed (= DCAM YUV 4:2:2) YUV 4:2:2 (YUYV) Packed
ADC Bit Depth	12 bits		10 bits (8 bits effective)
Synchronization	Via external trigger signal or via software		
Exposure Control	Programmable via the camera API		
Power Requirements	+12 to +24 VDC, < 1% ripple		
	3.0 W @ 12 V		2.5 W @ 12 V
I/O Ports	2 opto-isolated input ports and 4 opto-isolated output ports		
Lens Adapter	C-mount (CS-mount optional)		

Specification	scA640-70gm/gc	scA640-74gm/gc	scA750-60gm/gc
Size (L x W x H) (standard housing)	73.7 mm x 44 mm x 29 mm (without lens adapter or connectors) 85.5 mm x 44 mm x 29 mm (with lens adapter and connectors)		
(90° head housing)	91.65 mm x 44 mm x 29 mm (without connectors and front module) 97 mm x 44 mm x 41.8 mm (with connectors and front module)		
Weight (standard housing)	160 g (typical)		
(90° head housing)	180 g (typical)		
Conformity	CE, FCC, GenICam, GigE Vision, IP30		

Table 1: General Specifications

Specification	scA780-54gm/gc	scA1000-20gm/gc	scA1000-30gm/gc
Sensor Size (H x V pixels)	gm: 782 x 582 gc: 780 x 580	gm: 1034 x 779 gc: 1032 x 778	gm: 1034 x 779 gc: 1032 x 778
Sensor Type	Sony ICX415 AL/AQ	Sony ICX204 AL/AK	Sony ICX204 AL/AK
	Progressive scan CCD		
Optical Size	1/2"	1/3"	1/3"
Pixel Size	8.3 µm x 8.3 µm	4.65 µm x 4.65 µm	4.65 µm x 4.65 µm
Max. Frame Rate (at full resolution)	55 fps	20 fps	31 fps
Mono/Color	All models available in mono or color		
Data Output Type	Fast Ethernet (100 Mbit/s) or Gigabit Ethernet (1000 Mbit/s)		
Pixel Data Formats	Mono Models:		
	Mono 8 (= DCAM Mono 8) Mono 16 (= DCAM Mono 16) Mono 12 Packed YUV 4:2:2 Packed (= DCAM YUV 4:2:2) YUV 4:2:2 (YUYV) Packed		
	Color Models:		
	Mono 8 (= DCAM Mono 8) Bayer BG 8 (= DCAM Raw 8) Bayer BG 16 (= DCAM Raw 16) Bayer BG 12 Packed YUV 4:2:2 Packed (= DCAM YUV 4:2:2) YUV 4:2:2 (YUYV) Packed		
ADC Bit Depth	12 bits		
Synchronization	Via external trigger signal or via software		
Exposure Control	Programmable via the camera API		
Power Requirements	+12 to +24 VDC, < 1% ripple		
	3.0 W @ 12 V		
I/O Ports	2 opto-isolated input ports and 4 opto-isolated output ports		
Lens Adapter	C-mount (CS-mount optional)		

Specification	scA780-54gm/gc	scA1000-20gm/gc	scA1000-30gm/gc
Size (L x W x H) (standard housing)	73.7 mm x 44 mm x 29 mm (without lens adapter or connectors) 85.5 mm x 44 mm x 29 mm (with lens adapter and connectors)		
(90° head housing)	91.65 mm x 44 mm x 29 mm (without connectors and front module) 97 mm x 44 mm x 41.8 mm (with connectors and front module)		
Weight (standard housing)	160 g (typical)		
(90° head housing)	180 g (typical)		
Conformity	CE, FCC, GenICam, GigE Vision, IP30		

Table 1: General Specifications

Specification	scA1390-17gm/gc	scA1400-17gm/gc	scA1600-14gm/gc
Sensor Size (H x V pixels)	gm: 1392 x 1040 gc: 1390 x 1038	gm: 1392 x 1040 gc: 1390 x 1038	gm: 1626 x 1236 gc: 1624 x 1234
Sensor Type	Sony ICX267 AL/AK	Sony ICX285 AL/AQ	Sony ICX274 AL/AQ
	Progressive scan CCD		
Optical Size	1/2"	2/3"	1/1.8"
Pixel Size	4.65 µm x 4.65 µm	6.45 µm x 6.45 µm	4.4 µm x 4.4 µm
Max. Frame Rate (at full resolution)	17 fps	17 fps	14 fps
Mono/Color	All models available in mono or color		
Data Output Type	Fast Ethernet (100 Mbits/s) or Gigabit Ethernet (1000 Mbits/s)		
Pixel Data Formats	Mono Models:		
	Mono 8 (= DCAM Mono 8) Mono 16 (= DCAM Mono 16) Mono 12 Packed YUV 4:2:2 Packed (= DCAM YUV 4:2:2) YUV 4:2:2 (YUYV) Packed		
	Color Models:		
	Mono 8 (= DCAM Mono 8) Bayer BG 8 (= DCAM Raw 8) Bayer BG 16 (= DCAM Raw 16) Bayer BG 12 Packed YUV 4:2:2 Packed (= DCAM YUV 4:2:2) YUV 4:2:2 (YUYV) Packed		
ADC Bit Depth	12 bits		
Synchronization	Via external trigger signal or via software		
Exposure Control	Programmable via the camera API		
Power Requirements	+12 to +24 VDC, < 1% ripple		
	3.5 W @ 12 V		
I/O Ports	2 opto-isolated input ports and 4 opto-isolated output ports		
Lens Adapter	C-mount (CS-mount optional)		

Specification	scA1390-17gm/gc	scA1400-17gm/gc	scA1600-14gm/gc
Size (L x W x H) (standard housing)	73.7 mm x 44 mm x 29 mm (without lens adapter or connectors) 85.5 mm x 44 mm x 29 mm (with lens adapter and connectors)		
(90° head housing)	91.65 mm x 44 mm x 29 mm (without connectors and front module) 97 mm x 44 mm x 41.8 mm (with connectors and front module)		
Weight (standard housing)	160 g (typical)	170 g (typical)	160 g (typical)
(90° head housing)	180 g (typical)	190 g (typical)	180 g (typical)
Conformity	CE, FCC, GenICam, GigE Vision, IP30		

1.3 Spectral Response for Mono Cameras

The following graphs show the spectral response for each available monochrome camera model.

Note

The spectral response curves excludes lens characteristics and light source characteristics.

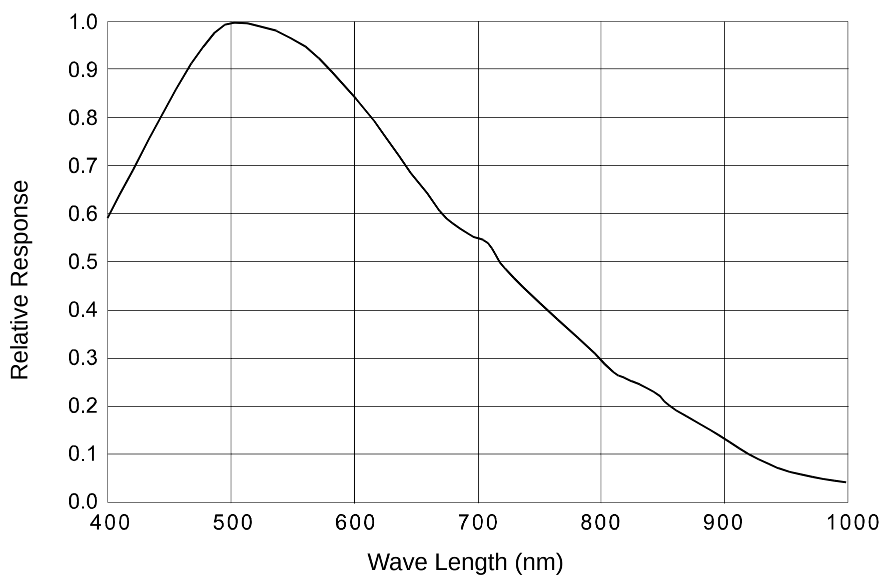


Fig. 1: scA640-70gm Spectral Response

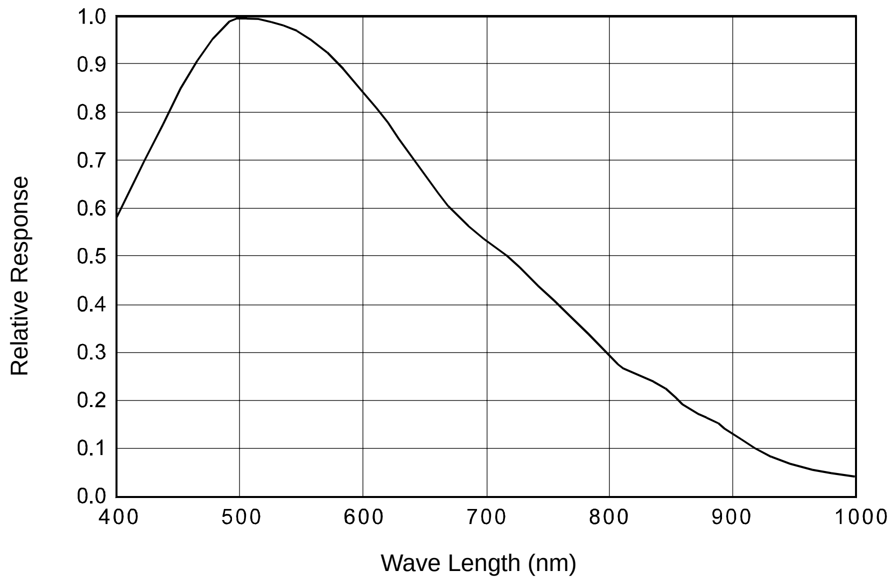


Fig. 2: scA640-74gm Spectral Response

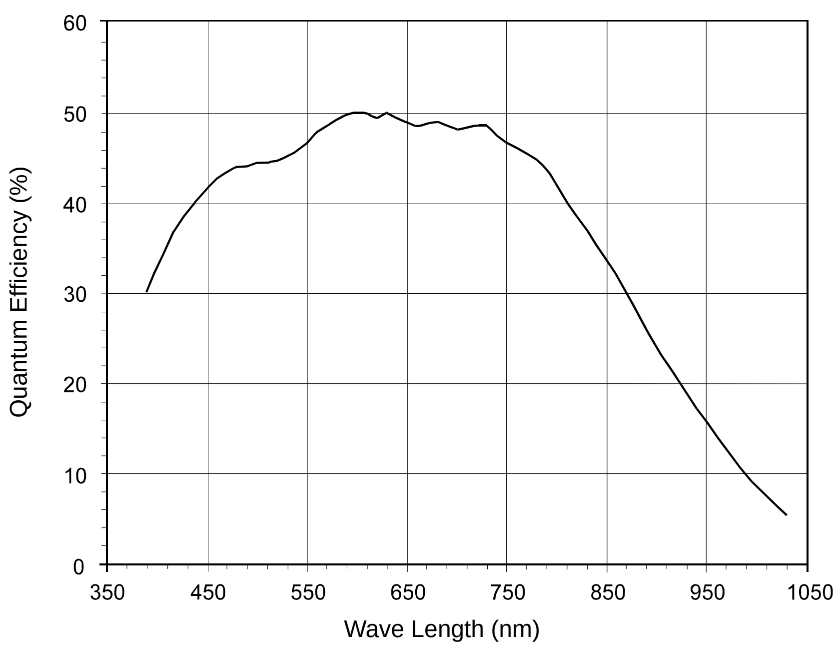


Fig. 3: scA750-60gm Spectral Response

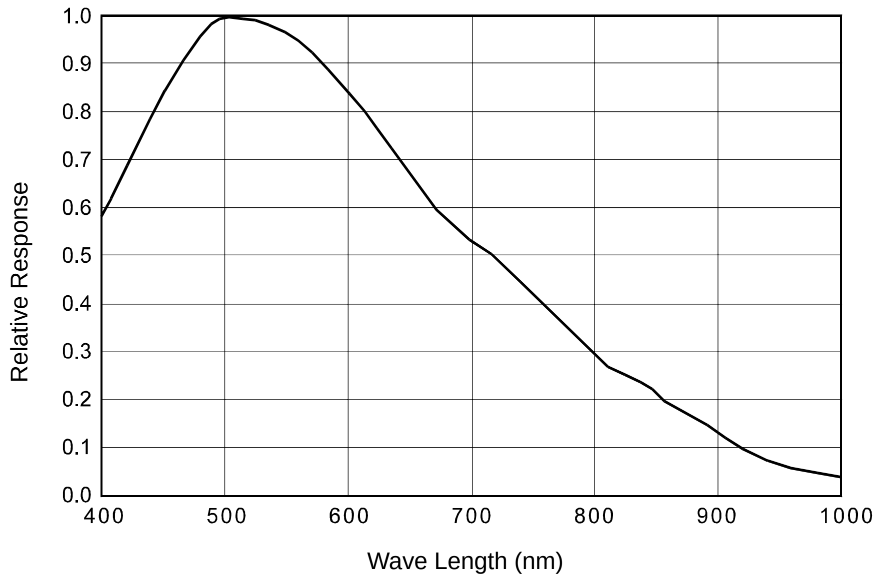


Fig. 4: scA780-54gm Spectral Response

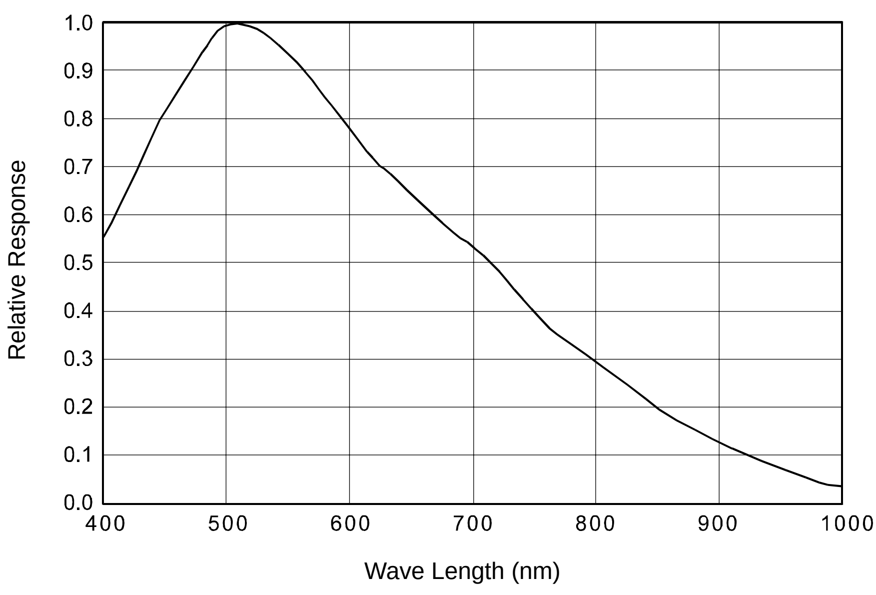


Fig. 5: scA1000-20gm and scA1000-30gm Spectral Response

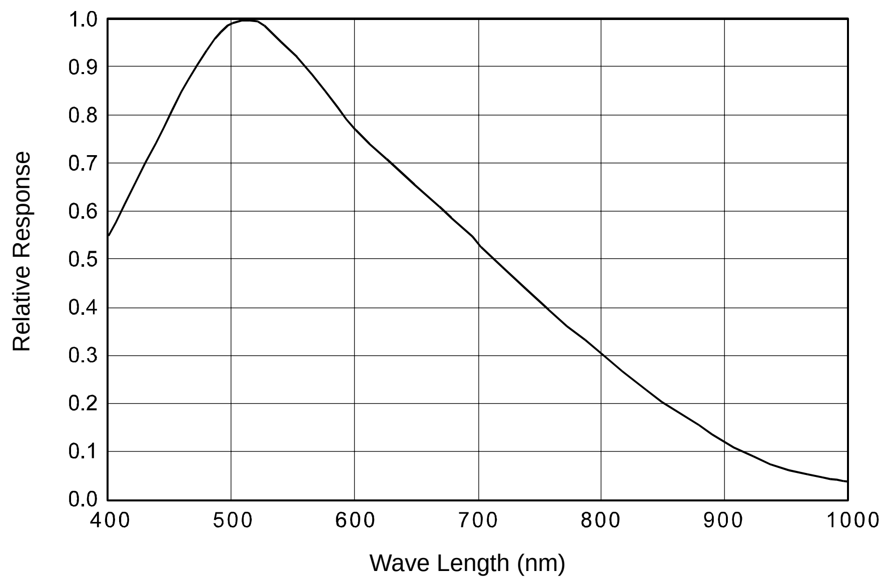


Fig. 6: scA1390-17gm Spectral Response

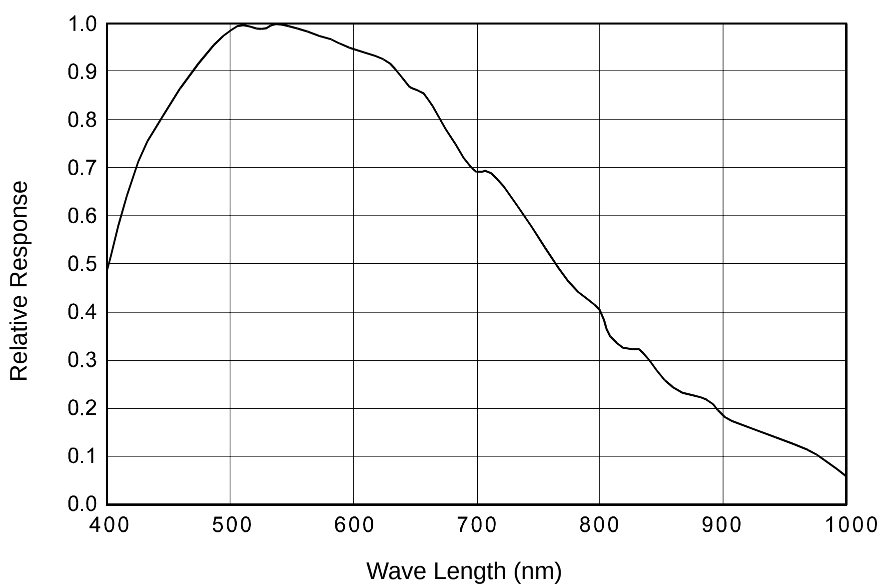


Fig. 7: scA1400-17gm Spectral Response

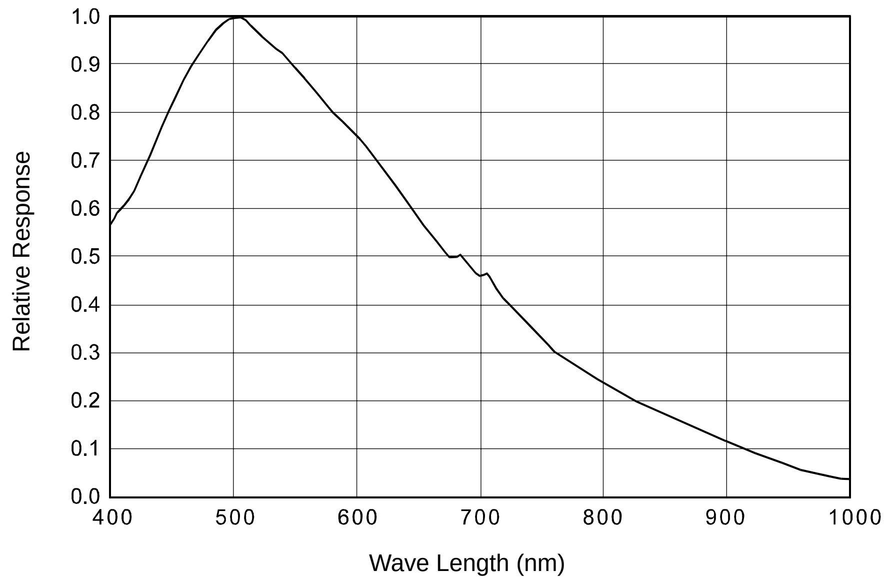


Fig. 8: scA1600-14gm

1.4 Spectral Response for Color Cameras

The following graphs show the spectral response for each available color camera model.

Note

The spectral response curves exclude lens characteristics, light source characteristics, and IR cut filter characteristics.

To obtain best performance from color models of the camera, use of a dielectric IR cut filter is recommended. The filter should transmit in a range from 400 nm to 700 ... 720 nm, and it should cut off from 700 ... 720 nm to 1100 nm.

A suitable IR cut filter is included in the standard C-mount lens adapter on color models of the camera. (An IR cut filter is not included in the optional CS-mount adapter.)

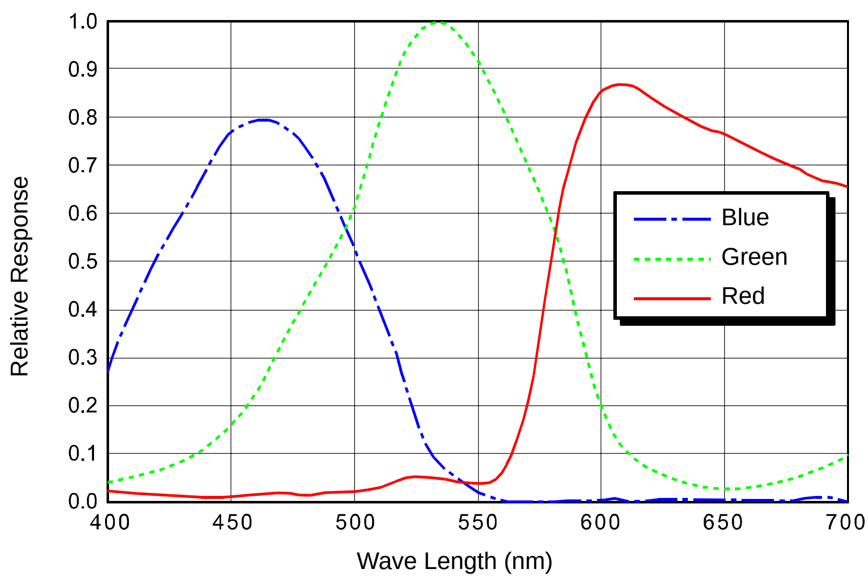


Fig. 9: scA640-70gc Spectral Response

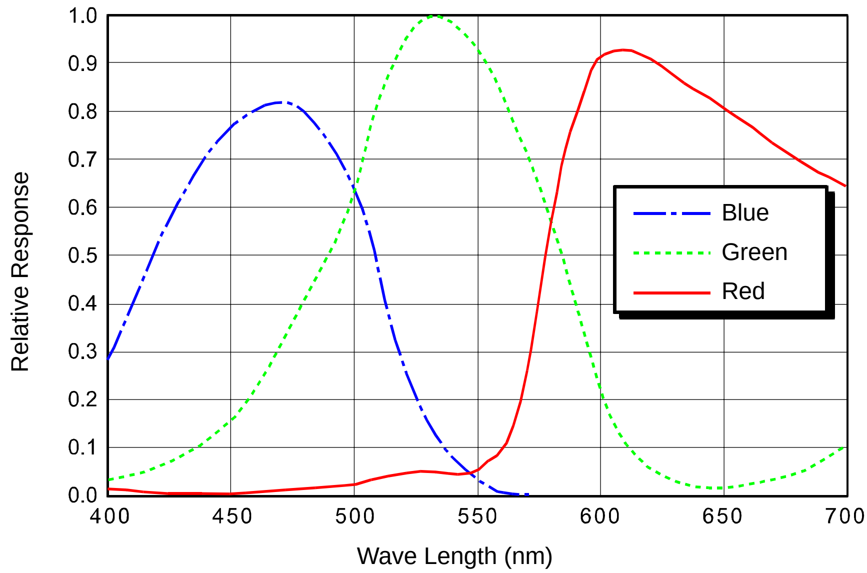


Fig. 10: scA640-74gc Spectral Response

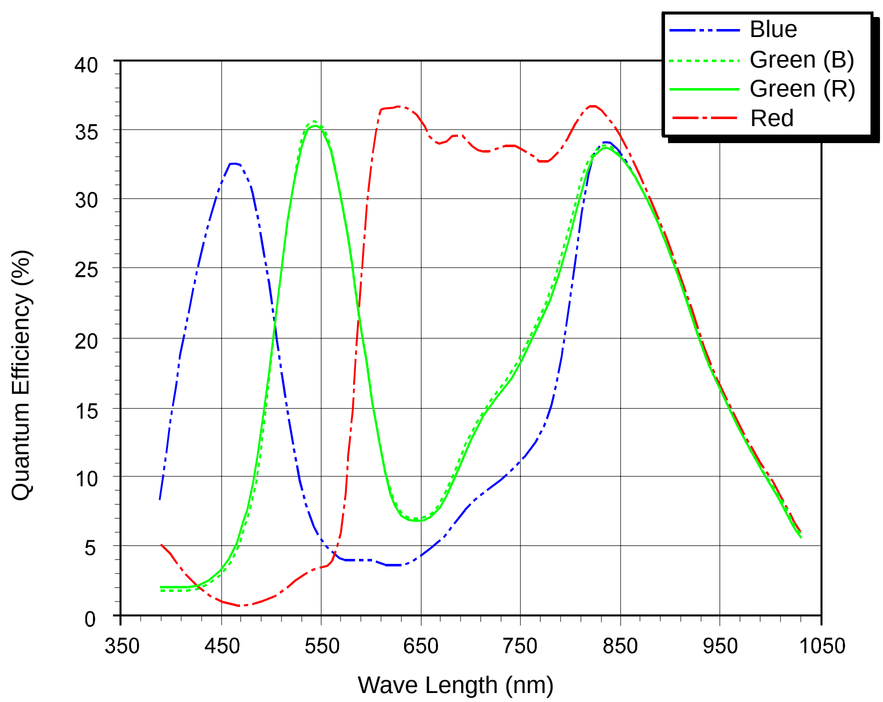


Fig. 11: scA750-60gc Spectral Response

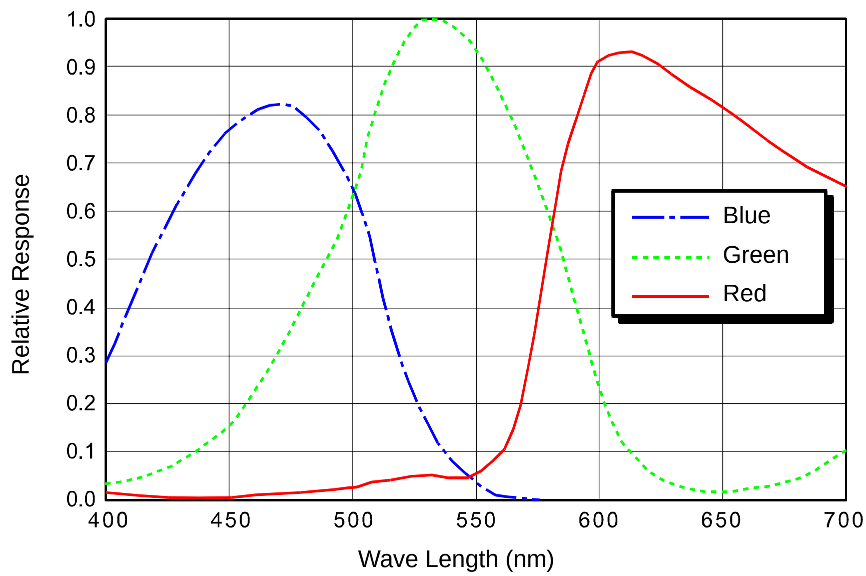


Fig. 12: scA780-54gc Spectral Response

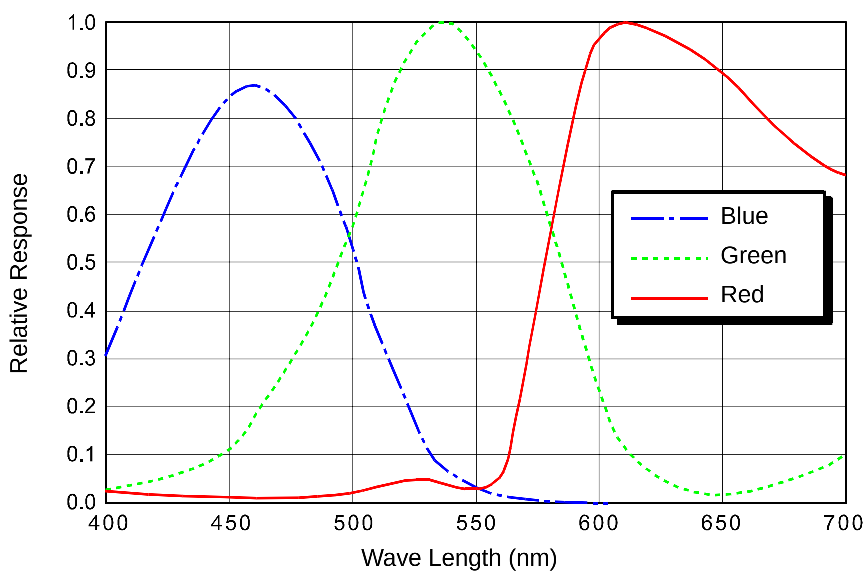


Fig. 13: scA1000-20gc and scA1000-30gc Spectral Response

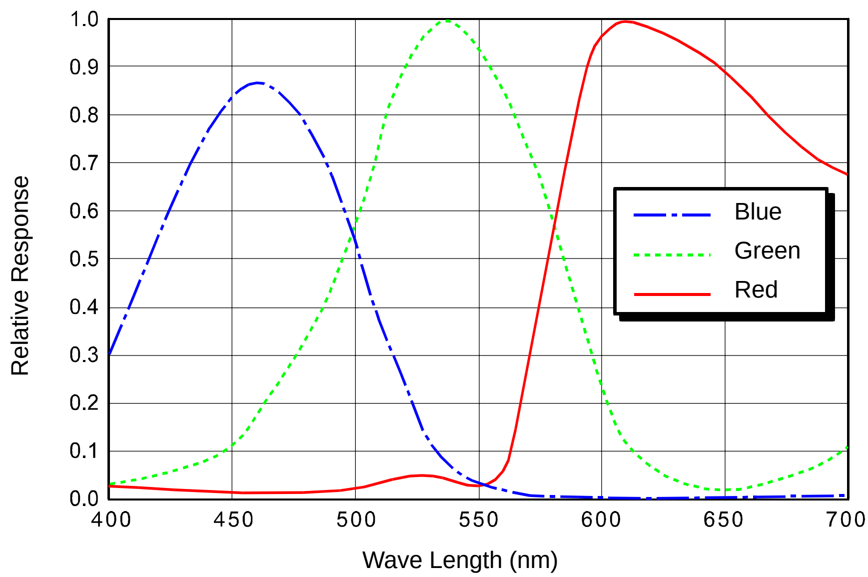


Fig. 14: scA1390-17gc Spectral Response

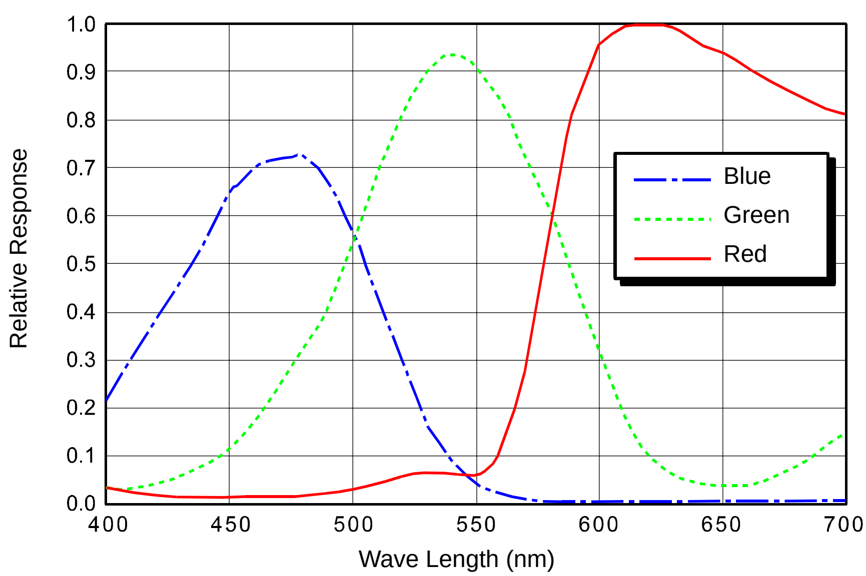


Fig. 15: scA1400-17gc Spectral Response

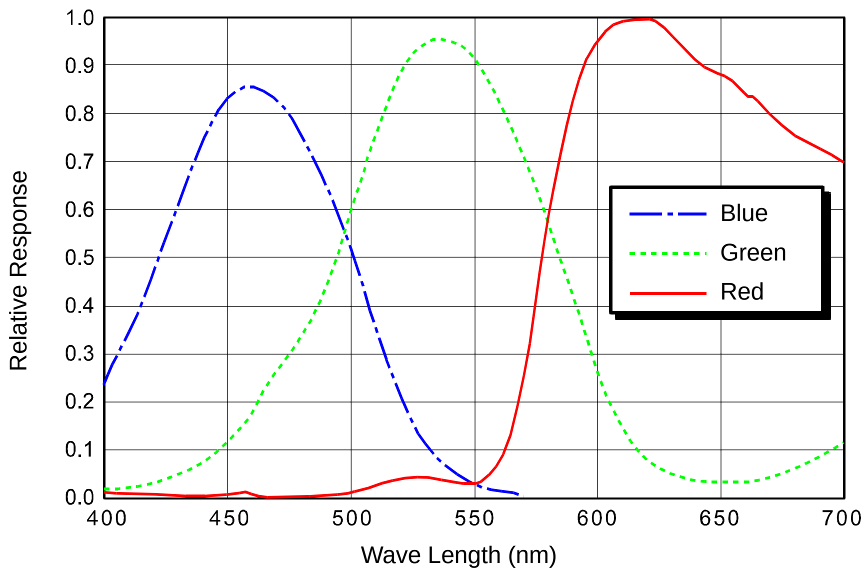


Fig. 16: scA1600-14gc

1.5 Mechanical Specifications

1.5.1 Standard Housing

1.5.1.1 Camera Dimensions and Mounting Points

The cameras are manufactured with high precision. Planar, parallel, and angular sides guarantee precise mounting with high repeatability.

The camera housings conform to protection class IP30 provided the lens mount is covered by a lens or by the cap that is shipped with the camera.

The dimensions in millimeters for cameras equipped with a standard C-mount lens adapter are as shown in Figure 17. The dimensions for cameras equipped with an optional CS-mount lens adapter are as shown in Figure 18 on page 20.

Camera housings are equipped with four mounting holes on the top and four mounting holes on the bottom as shown in the drawings.

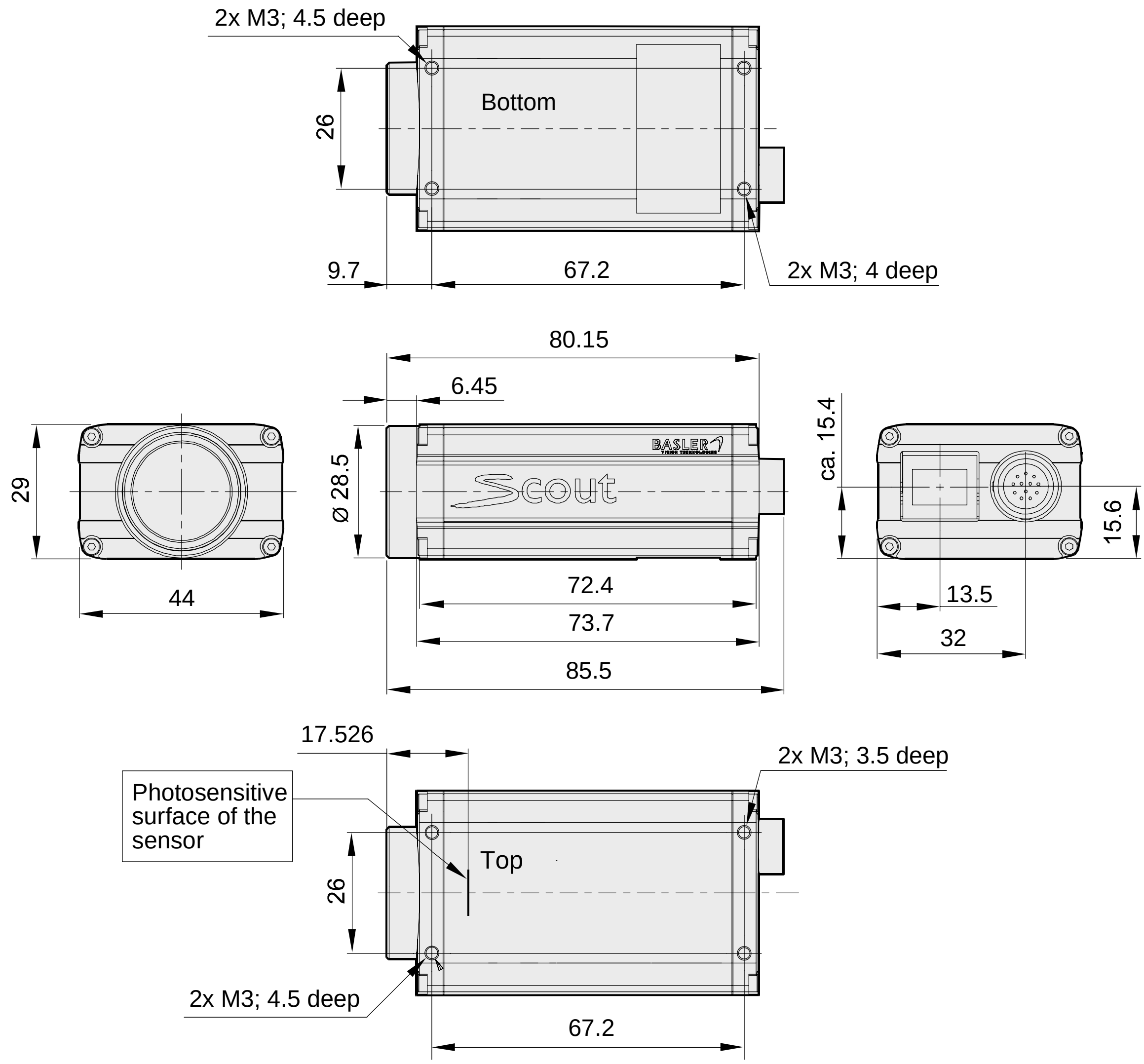


Fig. 17: Mechanical Dimensions (in mm) for Cameras with the Standard C-mount Lens Adapter

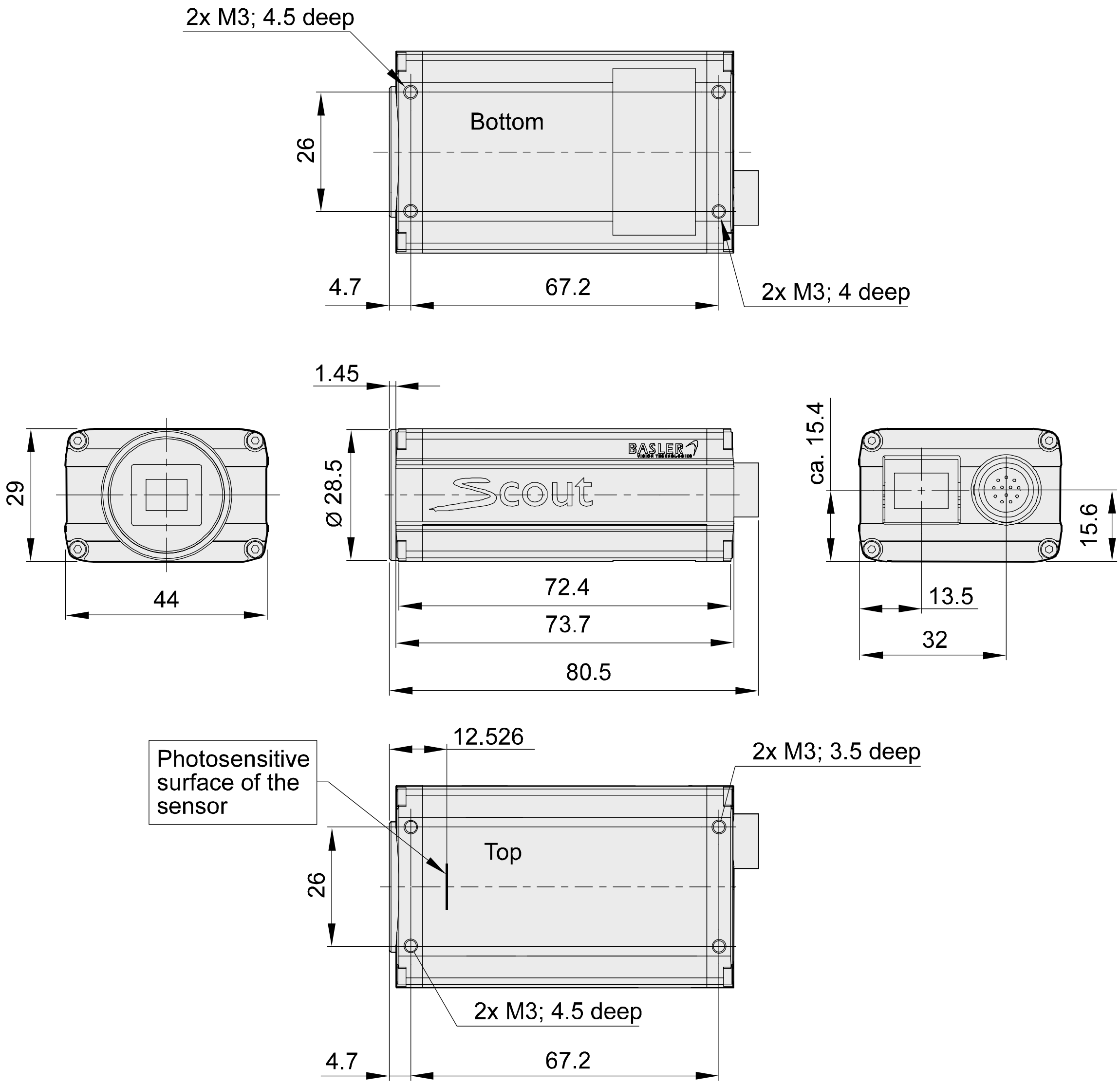
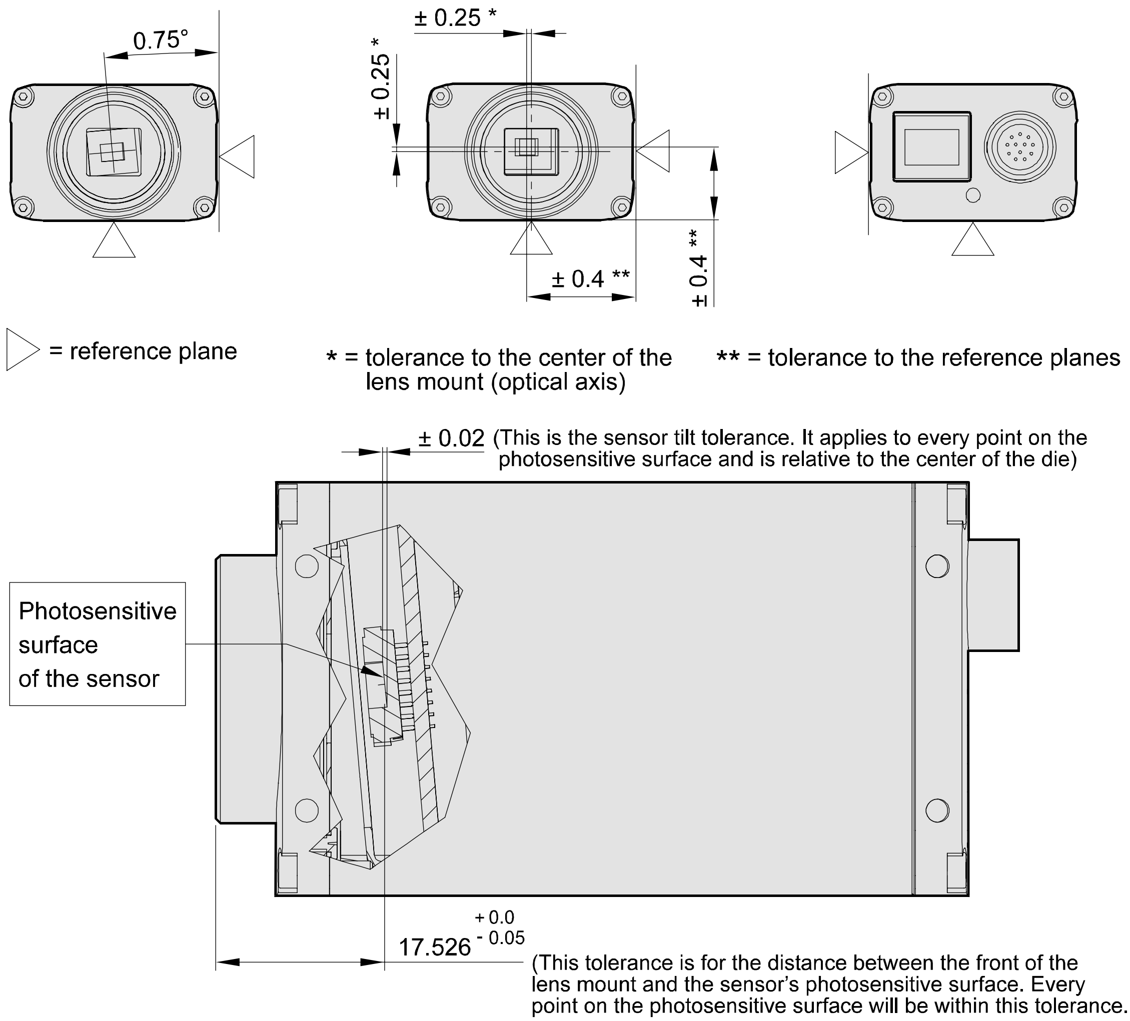


Fig. 18: Mechanical Dimensions (in mm) for Cameras with an Optional CS-mount Lens Adapter

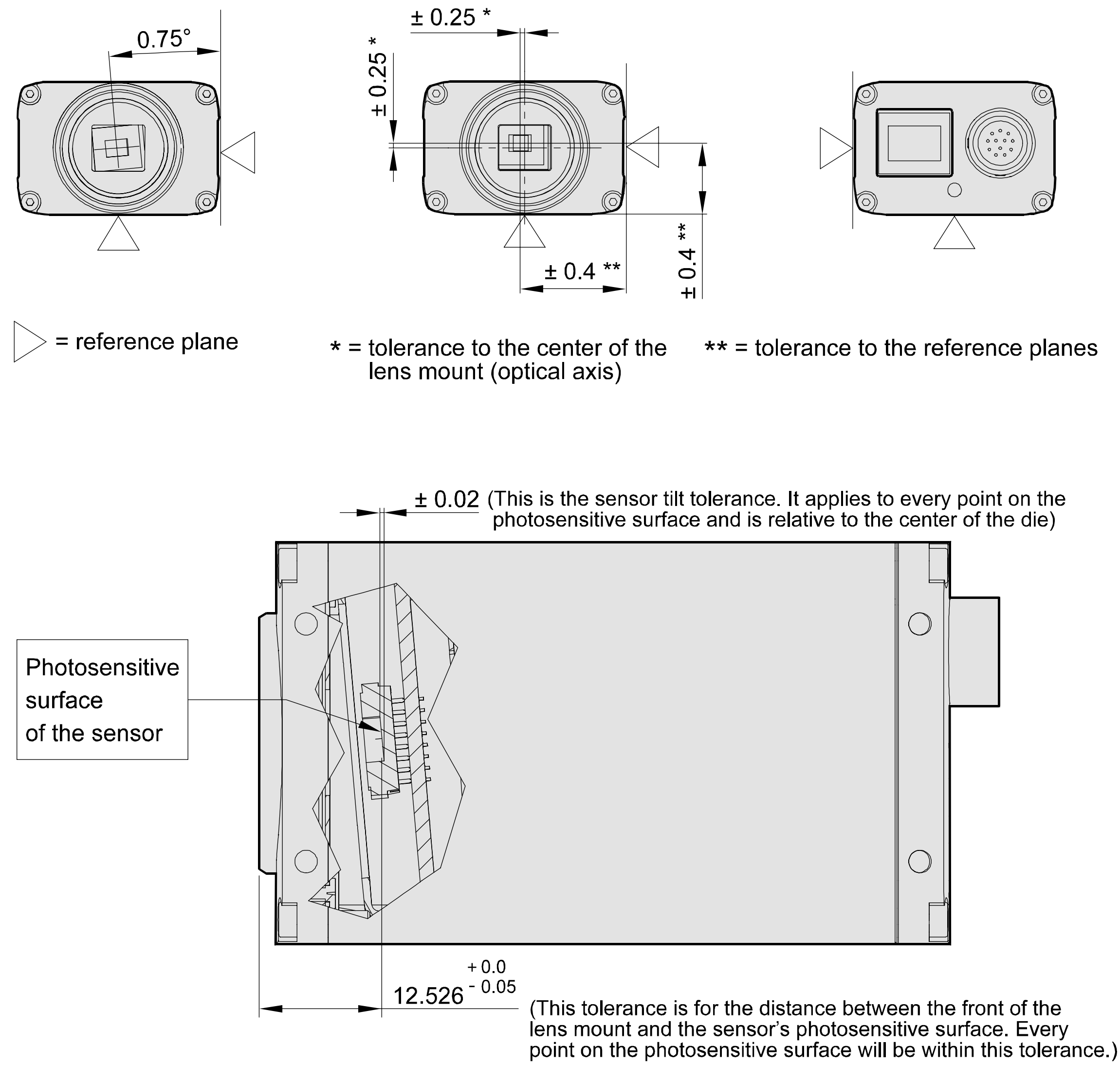
1.5.1.2 Sensor Positioning Accuracy

The sensor positioning accuracy for cameras equipped with a standard C-mount lens adapter is as shown in Figure 19. The sensor positioning accuracy for cameras equipped with an optional CS-mount lens adapter is as shown in Figure 20 on page 22.



Maximum Sensor Tilt Angle (Degrees)					
Camera	**Tilt X**	**Tilt Y**	**Camera**	**Tilt X**	**Tilt Y**
scA640-70gm/gc	0.47	0.63	**scA1000-30gm/gc**	0.46	0.63
scA640-74gm/gc	0.35	0.47	**scA1390-17gm/gc**	0.31	0.42
scA750-60gm/gc	0.51	0.80	**scA1400-17gm/gc**	0.25	0.34
scA780-54gm/gc	0.35	0.47	**scA1600-14gm/gc**	0.34	0.52
scA1000-20gm/gc	0.46	0.63			

Fig. 19: Sensor Positioning Accuracy for Cameras with the Standard C-mount Lens Adapter (in mm unless otherwise noted)



Maximum Sensor Tilt Angle (Degrees)					
Camera	**Tilt X**	**Tilt Y**	**Camera**	**Tilt X**	**Tilt Y**
scA640-70gm/gc	0.47	0.63	**scA1000-30gm/gc**	0.46	0.63
scA640-74gm/gc	0.35	0.47	**scA1390-17gm/gc**	0.31	0.42
scA750-60gm/gc	0.51	0.80	**scA1400-17gm/gc**	0.25	0.34
scA780-54gm/gc	0.35	0.47	**scA1600-14gm/gc**	0.34	0.52
scA1000-20gm/gc	0.46	0.63			

Fig. 20: Sensor Positioning Accuracy for Cameras with an Optional CS-mount Lens Adapter (in mm unless otherwise noted)

1.5.2 90° Head Housing

1.5.2.1 Camera Dimensions and Mounting Points

In scout cameras with the 90° head housing the camera's direction of view is at right angle to the direction of view of standard scout cameras.

The cameras are manufactured with high precision. Planar, parallel, and angular sides guarantee precise mounting with high repeatability.

The camera housings conform to protection class IP30 provided the lens mount is covered by a lens or by the cap that is shipped with the camera.

The dimensions in millimeters for cameras equipped with a standard C-mount lens adapter are as shown in Figure 21.

Camera housings are equipped with four mounting holes on the top and four mounting holes on the bottom as shown in the drawings. In addition, there are four mounting holes in the front module (4x M3; 4.5 mm deep).

Note

We recommend using the reference planes of the front module for optimum accuracy in the positioning of the camera's optical axis (see Figure 21 on page 24).

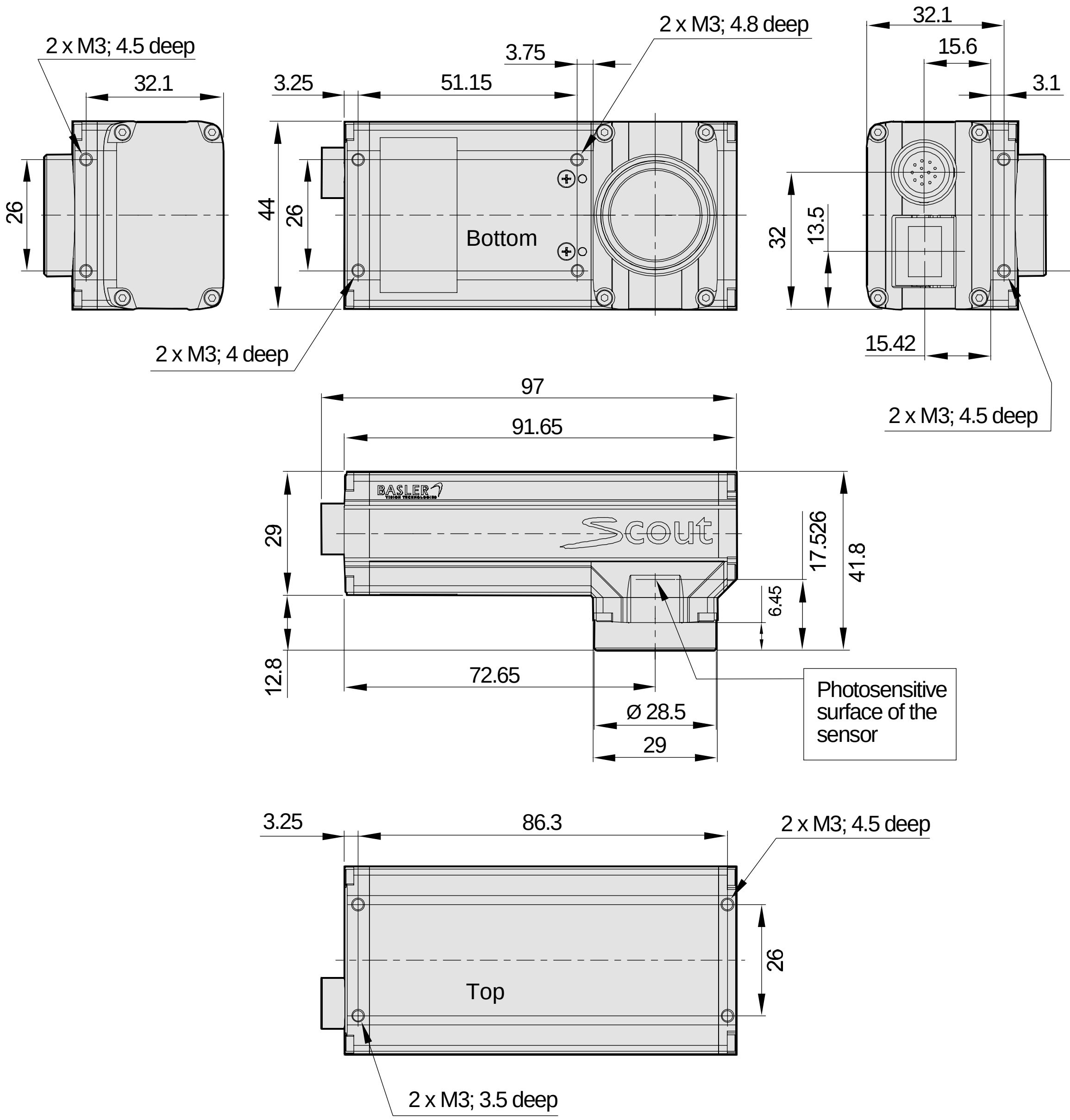


Fig. 21: Mechanical Dimensions (in mm) for Cameras (90° Head) with the Standard C-mount Lens Adapter

1.5.2.2 Sensor Positioning Accuracy

The sensor positioning accuracy for cameras equipped with a standard C-mount lens adapter is as shown in Figure 22 and Table 2.

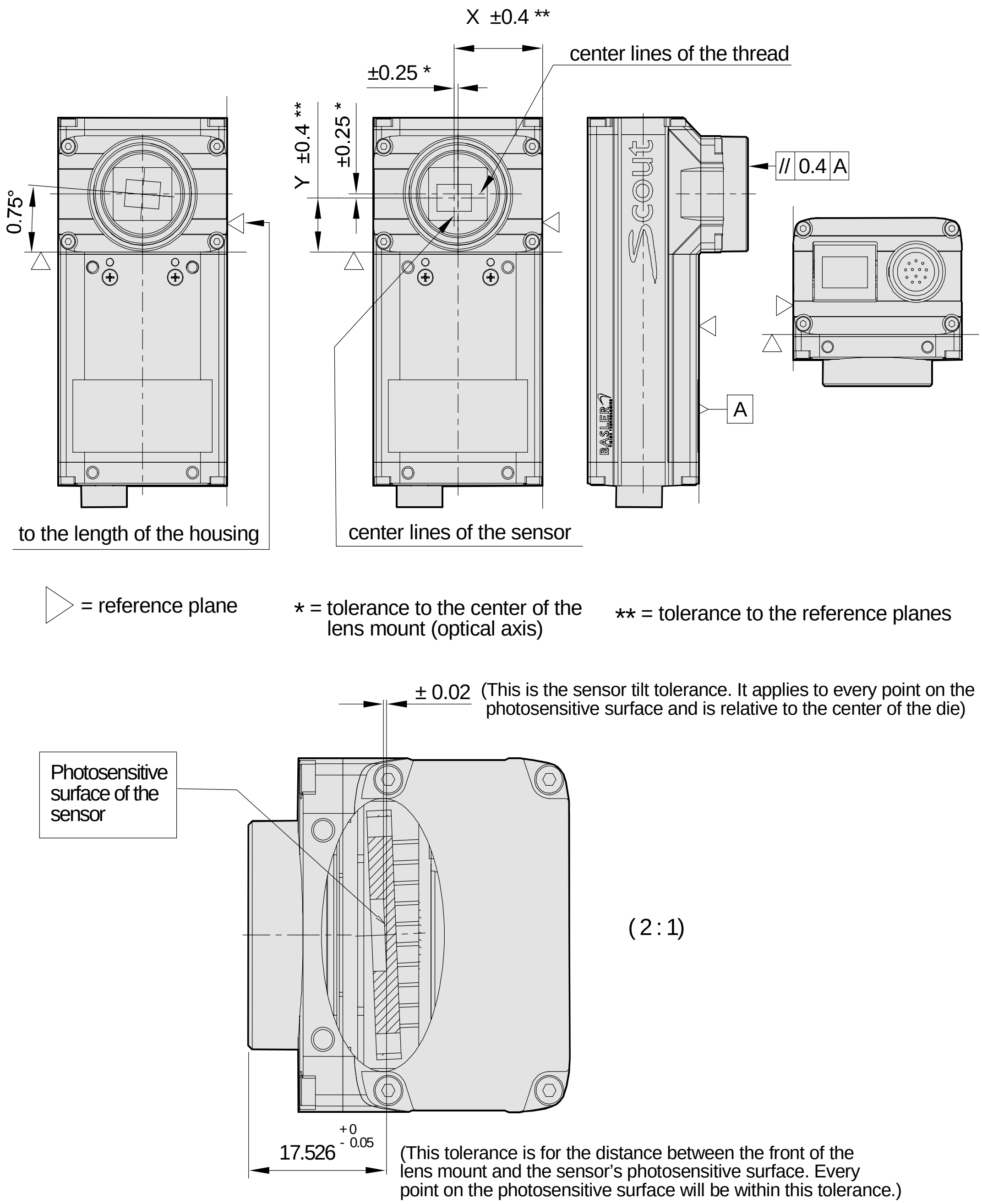


Fig. 22: Sensor Positioning Accuracy for Cameras (90° Head) with the Standard C-mount Lens Adapter (in mm unless otherwise noted)

Maximum Sensor Tilt Angle (Degrees)					
Camera	**Tilt X**	**Tilt Y**	**Camera**	**Tilt X**	**Tilt Y**
scA640-70gm/gc	0.47	0.63	**scA1000-30gm/gc**	0.46	0.63
scA640-74gm/gc	0.35	0.47	**scA1390-17gm/gc**	0.31	0.42
scA750-60gm/gc	0.51	0.80	**scA1400-17gm/gc**	0.25	0.34
scA780-54gm/gc	0.35	0.47	**scA1600-14gm/gc**	0.34	0.52
scA1000-20gm/gc	0.46	0.63			

Table 2: Sensor Positioning Accuracy for Cameras with 90° Head Housing

1.5.3 Maximum Thread Length on Color Cameras

The C-mount lens adapter on color models of the camera is normally equipped with an internal IR cut filter. As shown below, the length of the threads on any lens you use with a color camera must be less than 8.0 mm. If a lens with a longer thread length is used, the IR cut filter will be damaged or destroyed and the camera will no longer operate.

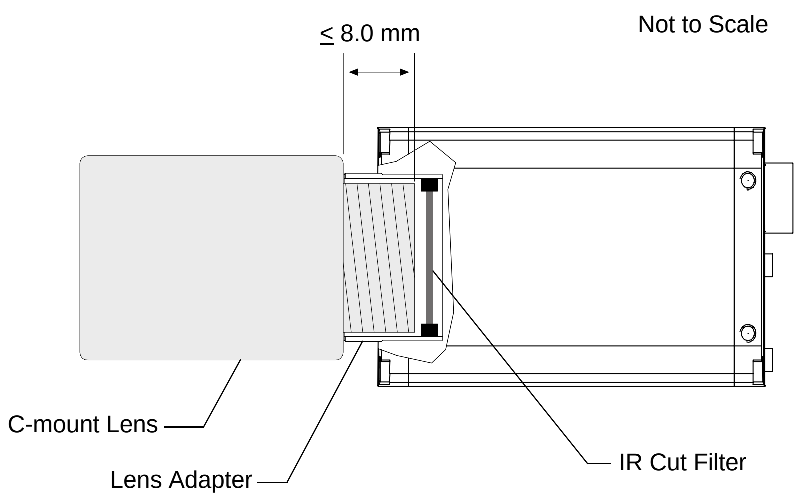


Fig. 23: Maximum Lens Thread Length on Color Cameras

Note

An internal IR cut filter is not included on color cameras equipped with the optional CS-mount adapter.

C-mount color cameras that do not include an internal IR cut filter are available on request.

Monochrome cameras are not normally equipped with an internal IR cut filter, however, they can be equipped with an internal filter on request.

1.5.4 Mechanical Stress Test Results

Scout cameras were submitted to an independent mechanical testing laboratory and subjected to the stress tests listed below. The mechanical shock tests were performed on selected camera models. After mechanical testing, the cameras exhibited no detectable physical damage and produced normal images during standard operational testing.

Test	Standard	Conditions
Vibration (sinusoidal, each axis)	DIN EN 60068-2-6	10-58 Hz / 1.5 mm_58-500 Hz / 20 g_1 Octave/Minute 10 repetitions
Shock (each axis)	DIN EN 60068-2-27	20 g / 11 ms / 10 shocks positive 20 g / 11 ms / 10 shocks negative
Bump (each axis)	DIN EN 60068-2-29	20 g / 11 ms / 100 shocks positive 20 g / 11 ms / 100 shocks negative
Vibration (broad-band random, digital control, each axis)	DIN EN 60068-2-64	15-500 Hz / 0.05 PSD (ESS standard profile) / 00:10 h

Table 3: Mechanical Tests

1.6 Environmental Requirements

1.6.1 Temperature and Humidity

Housing temperature during operation:	0 °C ... +50 °C (+32 °F ... +122 °F)
Humidity during operation:	20 % ... 80 %, relative, non-condensing
Storage temperature:	-20 °C ... +80 °C (-4 °F ... +176 °F)
Storage humidity:	20 % ... 80 %, relative, non-condensing

1.6.2 Ventilation

Allow sufficient air circulation around the camera to prevent internal heat build-up in your system and to keep the camera's housing temperature below 50 °C. Additional cooling devices such as fans or heat sinks are not normally required, but should be provided if necessary.

1.7 Precautions

CAUTION

Avoid Dust on the Sensor

The camera is shipped with a cap on the lens mount. To avoid collecting dust on the camera's sensor, make sure that you always put the cap in place when there is no lens mounted on the camera.

CAUTION

Lens Thread Length is Limited

Color models of the camera with a C-mount lens adapter are equipped with an IR cut filter mounted inside of the adapter. The location of this filter limits the length of the threads on any lens you use with the camera. If a lens with a very long thread length is used, the IR cut filter will be damaged or destroyed and the camera will no longer operate.

For more specific information about the lens thread length, see Section 1.5.3 on page 26.

CAUTION

Voltage Outside of Specified Range Can Damage the Camera

If the voltage of the input power to the camera is greater than +24 VDC damage to the camera can result. If the voltage is less than +12 VDC, the camera may operate erratically.

CAUTION

An Incorrect Plug Can Damage the 12-pin Connector

The plug on the cable that you attach to the camera's 12-pin connector must have 12 pins. Use of a smaller plug, such as one with 10 pins or 8 pins, can damage the pins in the camera's 12-pin connector.

Warranty Precautions

To ensure that your warranty remains in force:

Do not open the camera housing

Do not open the housing. Touching internal components may damage them.

Keep foreign matter outside of the camera

Be careful not to allow liquid, flammable, or metallic material inside of the camera housing. If operated with any foreign matter inside, the camera may fail or cause a fire.

Electromagnetic fields

Do not operate the camera in the vicinity of strong electromagnetic fields. Avoid electrostatic charging.

Transportation

Transport the camera in its original packaging only. Do not discard the packaging.

Cleaning

Avoid cleaning the surface of the camera's sensor if possible. If you must clean it, use a soft, lint free cloth dampened with a small quantity of high quality window cleaner. Because electrostatic discharge can damage the sensor, you must use a cloth that will not generate static during cleaning (cotton is a good choice).

To clean the surface of the camera housing, use a soft, dry cloth. To remove severe stains, use a soft cloth dampened with a small quantity of neutral detergent, then wipe dry.

Do not use solvents or thinners to clean the housing; they can damage the surface finish.

Read the manual

Read the manual carefully before using the camera!

2 Software and Hardware Installation

This section provides the information you will need to install and operate the camera. The installation procedure includes both software and hardware installation.

The installation procedure in this section assumes that you want to get your camera operational and begin capturing images as quickly and as simply as possible. Accordingly, the procedure describes a desktop installation for one camera and assumes that you will be using the "pylon Viewer" software from Basler to capture your first images.

2.1 General Preparations

The installation procedures assume that you will be making a peer-to-peer connection between your camera and a desktop computer. Make sure that the following items are available before starting the installation:

- A Basler GigE camera.
- A power supply for the camera. Make sure that the power supply meets all of the requirements listed in the Physical Interface section of this manual.
- A C-mount lens for the camera.

  If you already know what lens you will be using in your actual application, use this lens during the camera installation and setup. If not, we suggest that you use a zoom lens for your initial installation and setup. Contact Basler technical support if you need assistance in determining the best lens for your application. The support contact numbers appear in the title pages of this manual.
- A desktop computer with a GigE network adapter installed. We recommend that you use an Intel® PRO 1000 series adapter. These adapters have been tested with Basler cameras and work well. This installation procedure assumes that your computer is equipped with a PRO 1000 series adapter.

  The desktop computer must be equipped with a Windows 2000 (SP4) or a Windows XP (SP1 or SP2) operating system.
- A standard Ethernet patch cable. We recommend the use of a category 6 or category 7 cable that has S/STP shielding.
- The Basler pylon Software Development Kit (SDK) installation CD.

You should perform the software installation procedure first and the hardware installation procedure second.

2.2 Software Installation

2.2.1 What You Must Know Before You Begin

The software installation procedure assumes that you are installing version 1.0 of the pylon software. If you have an older version of the software, obtain version 1.0 before you begin the installation. If you have a newer version of the software, contact Basler technical support for assistance.

The software installation procedure assumes that there are no GigE cameras connected to the computer. If your GigE camera is connected, you should disconnect it now. You should connect your camera to the computer when you do the hardware installation procedure. (Connecting the camera during the hardware installation procedure simplifies the messages you will see.)

Two alternate software installation procedures are provided below:

- Installation of the Basler pylon Software Development Kit (SDK) allows you to install all components of the Basler pylon software.
- Installation of the Basler pylon Viewer package. If you do not intend developing software, you can install the Basler pylon Viewer package. The Basler pylon Viewer package includes all software components of the Basler pylon Software Development Kit (SDK) except the Basler pylon API and the sample programs.

The SDK includes these software components:

- The **Basler pylon API** - the API header files and libraries that you will need to create your own application programs.
- A programmer’s guide and API reference.
- A set of sample programs that illustrates how to use the camera API to parameterize the camera and acquire images.
- The **Basler pylon Viewer** - an application that can be used to easily parameterize the camera and acquire images.
- GigE vision network drivers:
  - The **Basler pylon filter driver** - a basic GigE Vision network driver that is compatible with all network adapters. The advantage of the filter driver is its extensive compatibility.
  - The **Basler pylon performance driver** - a hardware specific GigE Vision network driver. The performance driver is only compatible with network adapters that use specific Intel chipsets. The advantage of the performance driver is that it significantly lowers the CPU load needed to service the network traffic between the PC and the camera(s). It also has a

more robust packet resend mechanism.

- The **Basler pylon 1394 driver** - a camera driver for IEEE 1394 cameras.
- The **Basler pylon Direct Show driver** - a driver for use in combination with applications that display images using the Windows Direct Show module.
- The **IP Configuration Tool** - a tool for changing the IP configuration of the camera.
- The **Speed-O-Meter** - a tool for monitoring the camera's frame rate and its bandwidth usage.

For more information about compatible Intel chipsets, see Section 4.1 on page 65.

If you already have an older version of the pylon software on your computer, go through the software installation procedure in this manner:

- First, go to the "Removing Older Pylon Software" section and remove the older software if necessary.
- Next, read the "What Happens When Installing a Basler GigE Vision Network Driver" section.
- Next,
  - if you want to use the SDK, go to the "Installing the pylon SDK" section and install the SDK.
  - if you do not want to use the SDK, go to the "Installing the pylon Viewer" section and install the Basler pylon Viewer software.
  - if you only want to install an individual software component that is not included in your current Basler pylon installation, see the next step.
- Next, go to the "Adjusting the Installation" section to configure any Fast Ethernet or non-compatible network adapter used for the camera and to adjust the installation if necessary. The adjustments may, for example, involve reestablishing an original network driver-network adapter association, unbinding a Basler network driver from a network adapter, or installing an individual software component that is not included in your current Basler pylon installation.
- If you will be connecting cameras to more than one network adapter in a single PC, read the "Using Multiple Adapters in One PC" section.

If there is no pylon software on your computer, go through the software installation procedure in this manner:

- First, read the "What Happens When Installing a Basler GigE Vision Network Driver" section.
- Next,
  - if you want to use the SDK, go to the "Installing the pylon SDK" section and install the SDK.
  - if you do not want to use the SDK, go to the "Installing the pylon Viewer" section and install the Basler pylon Viewer software.
- Next, go to the "Adjusting the Installation" section to configure any Fast Ethernet or non-compatible network adapter used for the camera and to adjust the installation if necessary. The adjustments may, for example, involve reestablishing an original network driver-network adapter association, unbinding a Basler network driver from a network adapter, or installing an individual software component that is not included in your current Basler pylon installation.
- If you will be connecting cameras to more than one network adapter in a single PC, read the "Using Multiple Adapters in One PC" section.

2.2.1.1 What Happens When Installing a Basler GigE Vision Network Driver

During installation of the Basler pylon Software Development Kit or the Basler pylon Viewer, Basler network drivers are bound to all network adapters installed in your computer. This applies not only to all network adapters used to connect to cameras, but also to all other network adapters installed in your PC. Often, your PC will have two network adapters installed, with one used to connect to cameras and the other used to connect to a local area network.

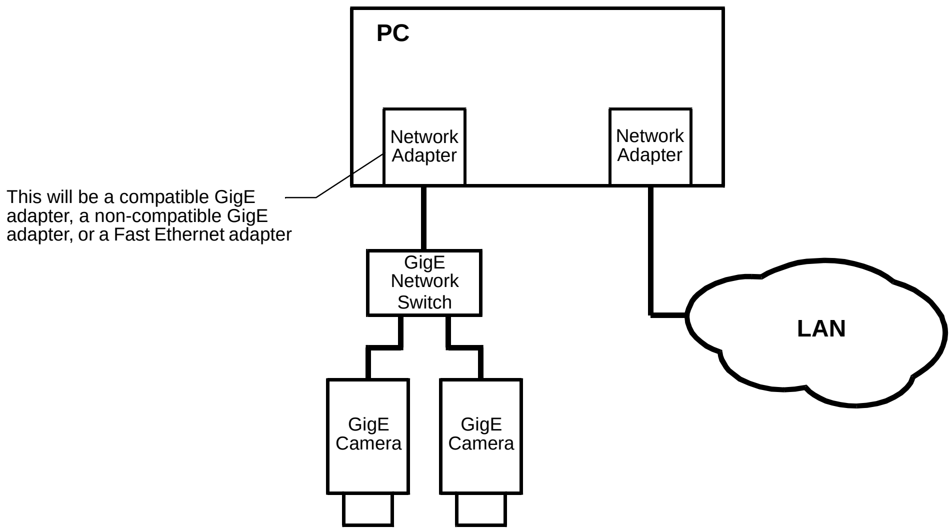


Fig. 24: Network Adapter for the Camera(s)

Two drivers are available for use with your GigE cameras:

- The **Basler filter driver** - is a basic GigE Vision network driver that is compatible with all network adapters. The advantage of the filter driver is its extensive compatibility.
- The **Basler performance driver** - is a hardware specific GigE Vision network driver. The performance driver is only compatible with network adapters that use specific Intel chipsets ("compatible chipsets"). The advantage of the performance driver is that it significantly lowers the CPU load needed to service the network traffic between the PC and the camera(s). It also has a more robust packet resend mechanism.

Note

If you are using a compatible GigE network adapter, the installation of the Basler pylon Software Development Kit or the Basler pylon Viewer will remove the driver that is currently installed for the adapter and will install the Basler performance driver.

If you are using a non-compatible GigE or a Fast Ethernet network adapter, the installation of the Basler pylon Software Development Kit or the Basler pylon Viewer will bind the Basler filter driver to the network adapter as a service and will enable the service.

For more information about compatible chipsets, see Section 4.1 on page 65.

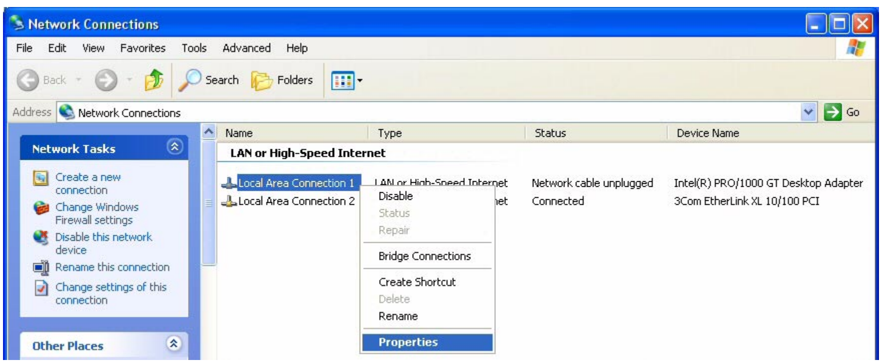
2.2.2 Removing Older Pylon Software

If you have version 0.9 or below of the Basler filter driver installed on your system, **it is critical that you remove the older version of the filter driver**. If you attempt to install a newer version of the filter driver with version 0.9 or below in place, the installation will fail without any type of service message. And if you install the performance driver on a system that has version 0.9 or below of the filter driver in place, the performance driver will not work correctly.

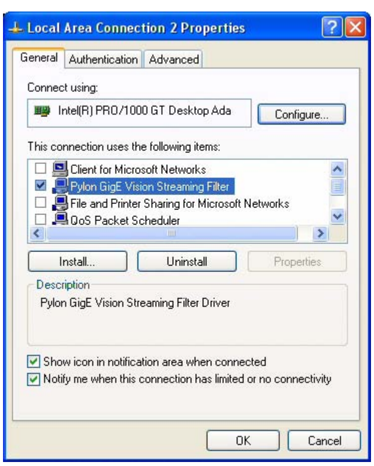
It is not necessary to remove older SDK installations. The SDK installer will automatically remove older versions of the SDK.

To remove version 0.9 or below of the filter driver:

1. Open a **Network Connections** window, and find the connection for a network adapter used with your cameras. Right click on the name of the connection and select **Properties** from the drop down menu as shown below.



2. A **LAN Connection Properties** window will open.
   a. Click on the **Pylon GigE Vision Streaming Filter** item so that it becomes highlighted as shown below.
   b. Click the **Uninstall** button.



3. When a window appears informing you that the component will be removed from all connections and asking you if you are sure, click the **Yes** button.

   The filter driver is removed from **all** network adapters, regardless of whether they are used with cameras or not.

4. When the removal process is complete, click the **Close** button on the **LAN Connection Properties** window.
5. Close the **Network Connections** window.

Filter driver removal is complete.

2.2.3 Installing the pylon SDK

> **Note**
>
> During installation of the Basler pylon SDK, the current network adapter-driver associations will be changed for **all** network drivers installed in your PC with Basler drivers replacing the current drivers.
>
> If you want to reestablish the original network adapter-driver association for a compatible GigE adapter, see the "Changing the Driver Association for a Compatible GigE Network Adapter" section.
>
> If you want to unbind a Basler network driver from a from a non-compatible GigE adapter or a Fast Ethernet adapter that is not used with a camera, see the "Unbinding the Basler Filter Driver from a Non-compatible or a Fast Ethernet Network Adapter" section.

1. Make sure your GigE camera is disconnected from your computer.
2. Insert the Basler pylon SDK Installation CD into your CD-ROM drive.
   a. Click **Start** and click **Run**.
   b. When the **Run** window opens, click the **Browse** button.
   c. Navigate to your CD-ROM drive and find the file called **Basler pylon SDK x.x.x.xxx.exe**.
   d. Click on the **Basler pylon SDK x.x.x.xxx.exe** file, click the **Open** button, and click the **OK** button.
3. The program will prepare to install and then a **Welcome** window will open. Click the **Next** button.
4. A **License Agreement** window will open. Accept the agreement and click the **Next** button.
5. A **Customer Information** window will open. Enter the appropriate information and click the **Next** button.
6. A **Setup Type** window will open. We recommend that you select **Complete** Installation and then click the **Next** button.

   (If you choose **Custom** Installation, the wizard will let you deselect the parts of the software that you do not want to install and will let you specify an alternate installation directory.)

7. A **Ready to Install the Program** window will open.

> **Note**
>
> If applicable, close all applications needing network connection before proceeding. Otherwise, the network connections for your applications will temporarily be lost.

   Click the **Install** button.

8. If a warning message appears indicating that Windows logo testing was not passed, click **Continue Anyway**.

   This action may be required several times.

9. When the installation process is complete, a **Completed** window will open. Click the **Finish** button.

10. An **Installer Information** window may open informing about the need to restart the computer.

    If you want to restart the computer now, click the **Yes** button.

    If you want to restart the computer later, click the **No** button.

    If the **Installer Information** window does not open, there is no need to restart the computer.

11. Note that the installation program has added shortcuts to the desktop for the **Pylon Viewer** and the **Pylon IP Configuration Tool**.

12. Click **Start**, click **All Programs**, click **Basler Vision Technologies**, and click **Pylon 1.0**. Note that shortcuts are available for the **Pylon IP Configuration Tool**, the **Pylon Viewer**, and the **Speed-O-Meter** tool. Also notice that there are shortcuts to the **Programmer's Guide and Reference Documentation** (i.e., the HTML version of the programmer's guide and API reference) and to the sample programs included with the SDK.

The SDK installation is complete.

> **Note**
>
> If you installed the Basler filter driver as part of the Basler pylon SDK installation (i.e. if you did not deselect the Basler filter driver) and if you are using a Fast Ethernet or non-compatible GigE network adapter:
>
> - Make sure to configure the Fast Ethernet or non-compatible GigE network adapter to be used with your cameras as described in the "Configuring a Non-compatible GigE or a Fast Ethernet Network Adapter Used with Your Cameras" section.
> - If you are using several adapters, you must configure each Fast Ethernet or non-compatible GigE network adapter to be used with your cameras individually.

2.2.4 Installing the pylon Viewer

Note

During installation of the Basler pylon Viewer, the current network adapter-driver associations will be changed for **all** network drivers installed in your PC with Basler drivers replacing the current drivers.

If you want to reestablish the original network adapter-driver association for a compatible GigE adapter, see the "Changing the Driver Association for a Compatible GigE Network Adapter" section.

If you want to unbind a Basler network driver from a from a non-compatible GigE adapter or a Fast Ethernet adapter that is not used with a camera, see the "Unbinding the Basler Filter Driver from a Non-compatible or a Fast Ethernet Network Adapter" section.

1. Make sure your GigE camera is disconnected from your computer.
2. Insert the Basler pylon SDK Installation CD into your CD-ROM drive.
   a. Click **Start** and click **Run**.
   b. When the **Run** window opens, click the **Browse** button.
   c. Navigate to your CD-ROM drive and find the file called **Basler pylon Viewer x.x.x.xxx.exe**.
   d. Click on the **Basler pylon Viewer x.x.x.xxx.exe** file, click the **Open** button, and click the **OK** button.
3. The program will prepare to install and then a **Welcome** window will open. Click the **Next** button.
4. A **License Agreement** window will open. Accept the agreement and click the **Next** button.
5. A **Customer Information** window will open. Enter the appropriate information and click the **Next** button.
6. A **Setup Type** window will open. We recommend that you select **Complete** Installation and then click the **Next** button.

   (If you choose **Custom** Installation, the wizard will let you deselect the parts of the software that you do not want to install and will let you specify an alternate installation directory.)
7. A **Ready to Install the Program** window will open.

   Note

   If applicable, close all applications needing network connection before proceeding. Otherwise, the network connections for your applications will temporarily be lost.

   Click the **Install** button.

8. If a warning message appears indicating that Windows logo testing was not passed, click **Continue Anyway**.

   This action may be required several times.
9. When the installation process is complete, a **Completed** window will open. Click the **Finish** button.
10. An **Installer Information** window may open informing about the need to restart the computer.

    If you want to restart the computer now, click the **Yes** button.

    If you want to restart the computer later, click the **No** button.

    If the **Installer Information** window does not open, there is no need to restart the computer.
11. Note that the installation program has added shortcuts to the desktop for the **Pylon Viewer** and the **Pylon IP Configuration Tool**.
12. Click **Start**, click **All Programs**, click **Basler Vision Technologies**, and click **Pylon 1.0**. Note that shortcuts are available for the **Pylon IP Configuration Tool**, the **Pylon Viewer**, and the **Speed-O-Meter** tool.

The pylon Viewer installation is complete.

Note

If you installed the Basler filter driver as part of the Basler pylon Viewer installation (i.e. if you did not deselect the Basler filter driver) and if you are using a Fast Ethernet or non-compatible GigE network adapter:

- Make sure to configure the Fast Ethernet or non-compatible GigE network adapter to be used with your cameras as described in the "Configuring a Non-compatible GigE or a Fast Ethernet Network Adapter Used with Your Cameras" section.
- If you are using several adapters, you must configure each Fast Ethernet or non-compatible GigE network adapter to be used with your cameras individually.

2.2.5 Adjusting the Installation

This section provides information on adjustments that must be made after the installation of the Basler pylon Software Development Kit or the Basler pylon Viewer and on additional adjustments that may be required.

Necessary adjustments:

- Configuring the Fast Ethernet or non-compatible GigE network adapters that you may use with your cameras

Possibly necessary additional adjustments:

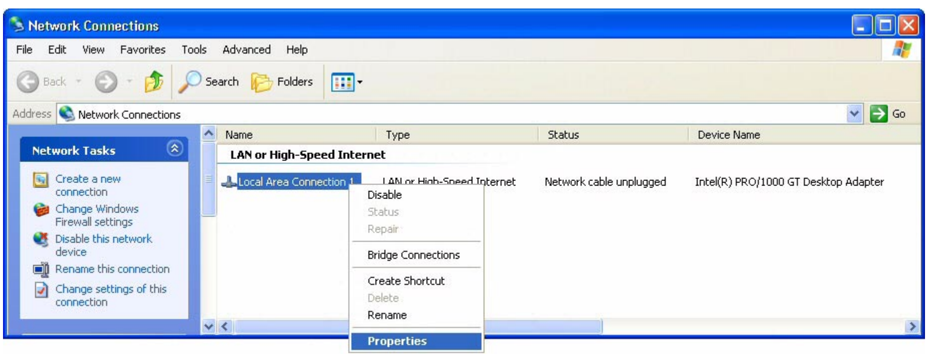
- Changing the driver association for a compatible GigE network adapter
- Unbinding the Basler filter driver from a non-compatible GigE or a Fast Ethernet network adapter
- Installing an individual software component that is not included in your current Basler pylon installation.

2.2.5.1 Configuring a Non-compatible GigE or a Fast Ethernet Network Adapter Used with your Cameras

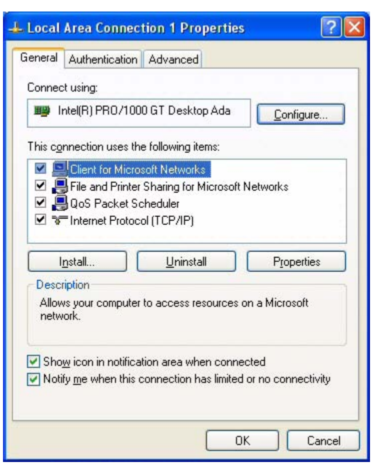
The following procedures assume that the Basler filter driver was installed on your PC during Installation of the Basler pylon SDK, during installation of the Basler pylon Viewer, or as an individual software component. After the installation, the settings of each Fast Ethernet or non-compatible network adapter used with your camera must be checked and adjusted, if necessary:

1. Open a **Network Connections** window and find the connection for your network adapter that is used with cameras. (Make sure that the window is set to display details (right click within the window, select **View**, and select **Details**).

   If you have multiple Fast Ethernet or non-compatible GigE adapters that are used with your cameras, select any one of them.
2. Right click on the name of the connection and select **Properties** from the drop down menu as shown below.

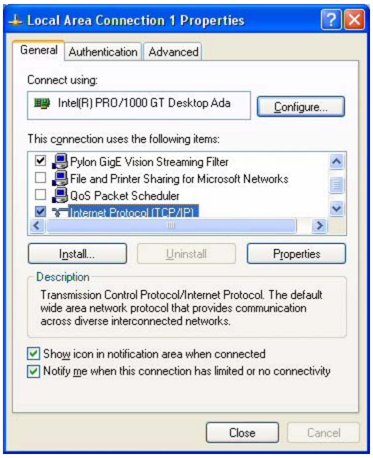


3. A **LAN Connection Properties** window will open as shown below.



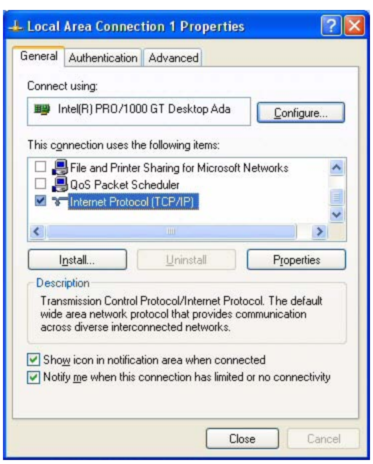
4. Look at the list of items in the center of the **Local Area Connection Properties** window.
   a. Make sure that the **Pylon GigE Vision Streaming Filter** and the **Internet Protocol (TCP/IP)** items are checked as shown below.
   b. Make sure that all of the other items in the list are unchecked. (Note that you may need to scroll the list up or down to see all of the items.)

c. Click the **Close** or the **OK** button on the **Local Area Connection Properties** window (either a **Close** or an **OK** button will be present depending on what changes you made).



5. In the **Network Connections** window, right click on the name of the connection for your network adapter, and select **Properties** from the drop down menu.
6. In the list box at the center of the **Local Area Connection Properties** window, click on **Internet Protocol (TCP/IP)** so that it becomes highlighted as shown below and then click the **Properties** button.

   (Note that you may need to scroll down through the items in the list box to find the **Internet Protocol (TCP/IP)** item.)



7. An **Internet Protocol (TCP/IP) Properties** window will open as shown below.

   a. Make sure that the **Obtain an IP address automatically** radio button is selected. If it is selected, go on to step 8.

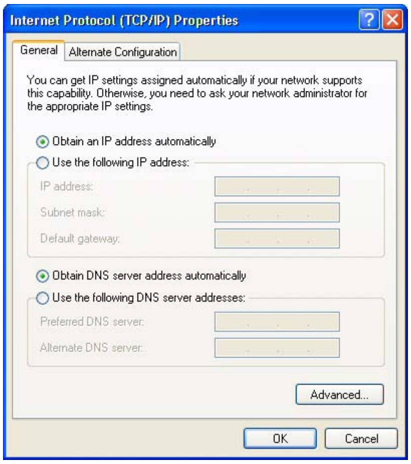
   b. If it is not selected:

   Select the **Obtain an IP address automatically** radio button.

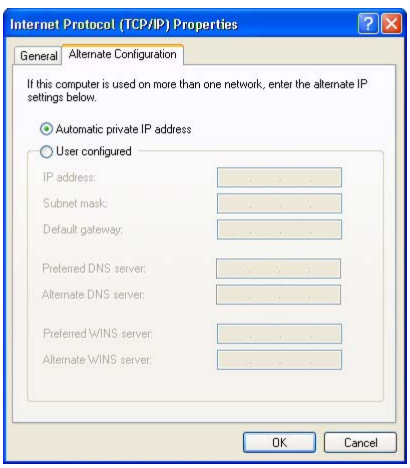
   Click the **OK** button. A **Local Area Connection Properties** window will open.

   On the **LAN Connection Properties** window, make sure that **Internet Protocol (TCP/IP)** item is highlighted and click the **Properties** button.

Go on to step 8.



8. Click on the **Alternate Configuration** tab.
   a. Make sure that the **Automatic private IP address** radio button is selected as shown below.
   b. Click the **OK** button on the **Internet Protocol Properties** window.
   c. Click the Close button on the **Local Area Connection Properties** window.



9. If you have only one Fast Ethernet or non-compatible network adapter in the PC, the configuration for the filter driver is complete and you can close the **Network Connections** window and exit this procedure.

   If you have more than one Fast Ethernet or non-compatible network adapter in the PC, return to step 1, select one of the other adapters, and perform steps 2 through 9 for each adapter.

2.2.5.2 Changing the Driver Association for a Compatible GigE Network Adapter

During installation of the Basler pylon SDK or the Basler pylon Viewer, the current network adapter-driver associations are changed for **all** network drivers installed in your PC with Basler drivers replacing the current drivers.

If you want to reestablish a network adapter-driver association with the original driver, follow the procedures below. As an example, the procedures describe how to change the driver on a GigE network adapter from the Performance Driver back to the original driver.

1. Open a **Network Connections** window. Make sure that the window is set to display details (right click within the window, select **View**, and select **Details**).
2. Find the entry for the network adapter whose association you want to change. When moving the pointer over the name of the network adapter, notice that the tooltip indicates **Basler GigE Vision Adapter**.
3. Right click on the name of the network adapter.
4. Select **Properties** from the drop down menu. A **LAN Connection Properties** window for the adapter will open.
5. Click the **Configure** button.
6. Select the **Driver** tab.
7. Click the **Update Driver** button.
8. When the hardware wizard opens, select **No, not this time** and click the **Next** button.
9. Select **Install from a list or specidfic location (Advanved)** and click the **Next** button.
10. Select **Don’t search, I will choose the driver to install** and click the **Next** button.
11. From the list that appears, select the original driver for the adapter (e.g., in a case where the Intel PRO 1000 GT Desktop Adapter is installed in the PC, select **Intel PRO 1000 Desktop Adapter** from the list) and click the **Next** button.
12. The system will update the driver. Click the **Finish** button.
13. Close the **Properties** window.
14. In the **Network Connections** window, notice that the Device Name for the adapter has changed (in the case of the Intel PRO 1000 GT, it will now be **Intel PRO/ 1000 GT Desktop Adapter**).

The network adapter-driver association is changed.

2.2.5.3 Unbinding the Basler Filter Driver from a Non-compatible GigE or a Fast Ethernet Network Adapter

During installation of the Basler pylon SDK or the Basler pylon Viewer, the current network adapter-driver associations will be changed for **all** network drivers installed in your PC with Basler drivers replacing the current drivers.

Your application may require that no Basler network driver is bound to an adapter that is not used for a camera.

As an example, the following procedures describe how to unbind the Basler filter driver from a Fast Ethernet (or non-compatible GigE) network adapter:

1. Open a **Network Connections** window. Make sure that the window is set to display details (right click within the window, select **View**, and select **Details**).
2. Find the entry for the network adapter from which you want to unbind the Basler network driver.
3. Right click on the name of the network adapter.
4. Select **Properties** from the drop down menu. A **LAN Connection Properties** window for the adapter will open.
5. Look for the list box in the middle of the window that is labelled **This connection uses the following items**.
6. In the list, find the entry for **Pylon GigE Vision Streaming Filter** and uncheck the entry.
7. Click the **OK** button.
8. The driver will be unbound and the **Properties** window will close.

The Basler network driver is unbound.

Note

You could completely uninstall the Basler network driver by clicking on its entry in the list and then clicking the **Uninstall** button. Be aware that doing so will remove the Basler network driver from all network adapters

2.2.5.4 Installing a Software Component

The procedure below describes how to install a Basler pylon software component that is not included in the current installation of Basler pylon software on your PC.

You will have to install a Basler pylon software component in, for example, one of these situations:

- You initially only installed the Basler pylon Viewer and now want to use the SDK.
- You initially installed the Filter Driver only because no compatible GigE network adapters were installed in your PC. You have now installed a compatible GigE network adapter in your PC and want to use the Performance Driver.

Note

The software component to be installed must have the same version number as the Basler pylon software currently installed on your PC.

If in doubt, obtain the latest version of Basler pylon software and install it. This will update the Basler pylon software currently installed on your PC and allows you to include the desired software component.

Note

During installation of a Basler pylon network driver, the current network adapter-driver associations will be changed for **all** network drivers installed in your PC with Basler drivers replacing the current drivers.

If you want to reestablish the original network adapter-driver association for a compatible GigE adapter, see the "Changing the Driver Association for a Compatible GigE Network Adapter" section.

If you want to unbind a Basler network driver from a from a non-compatible GigE adapter or a Fast Ethernet adapter that is not used with a camera, see the "Unbinding the Basler Filter Driver from a Non-compatible or a Fast Ethernet Network Adapter" section.

1. Make sure your GigE camera is disconnected from your computer.
2. Insert the Basler pylon SDK Installation CD into your CD-ROM drive.
    a. Click **Start** and click **Run**.
    b. When the **Run** window opens, click the **Browse** button.
    c. Navigate to your CD-ROM drive and find the file called **Basler pylon SDK x.x.x.xxx.exe**.
    d. Click on the **Basler pylon SDK x.x.x.xxx.exe** file, click the **Open** button, and click the **OK** button.
3. The program will prepare to install and then a **Welcome** window will open. Click the **Next** button.
4. A **License Agreement** window will open. Accept the agreement and click the **Next** button.

5. A **Customer Information** window will open. Enter the appropriate information and click the **Next** button.
6. A **Setup Type** window will open.
   a. Choose **Custom** Installation.
   b. Deselect all components of the software that you do not want to install.
   c. Click the **Next** button
7. A **Ready to Install the Program** window will open.

> **Note**
>
> If applicable, close all applications needing network connection before proceeding. Otherwise, the network connections for your applications will temporarily be lost.

   Click the **Install** button.
8. If a warning message appears indicating that Windows logo testing was not passed, click **Continue Anyway**.

   This action may be required several times.
9. When the installation process is complete, a **Completed** window will open. Click the **Finish** button.
10. An **Installer Information** window may open informing about the need to restart the computer.

    If you want to restart the computer now, click the **Yes** button.

    If you want to restart the computer later, click the **No** button.

    If the **Installer Information** window does not open, there is no need to restart the computer.

The installation of the software component is complete.

> **Note**
>
> If you installed the Basler filter driver and if you are using a Fast Ethernet or non-compliant GigE network adapter:
>
> - Make sure to configure the Fast Ethernet or non-compatible GigE network adapter to be used with your cameras as described in the "Configuring a Non-compatible GigE or a Fast Ethernet Network Adapter Used with Your Cameras" section.
> - If you are using several adapters, you must configure each Fast Ethernet or non-compatible GigE network adapter to be used with your cameras individually.

2.2.6 If You Are Using Multiple Network Adapters in a Single PC

With a typical installation of GigE cameras, the default network adapter settings will cause each adapter to use what is known as Automatic Private IP Addressing (APIPA) to assign itself an IP address. If you are using only one network adapter in you host PC, these adapter settings will work correctly with Basler GigE cameras that are set to default (e.g., new cameras coming out of the box).

However, if you have multiple network adapters in your PC and you will be connecting cameras to more than one adapter, this situation will cause a problem. The APIPA addressing scheme does not work correctly when it is used on more than one adapter.

The easiest way to avoid this problem when you are doing the initial camera installation and setup of your system is to change the adapter IP configuration settings and the camera IP configuration settings so that the cameras and the adapters will use fixed IP addressing. You should do this after you have completed the pylon software installation and you have followed the hardware installation procedure to connect the cameras to your adapters.

When you do the fixed IP addressing, keep in mind that each adapter in the PC must be assigned to a different subnet. Also keep in mind that all of the cameras attached to a single adapter must be on the same subnet as the adapter.

Be aware that if you are using a multiport network adapter in your PC, each port is considered to be a separate adapter.

For more information about changing the IP configuration of the cameras and the network adapters, see Section 5 on page 71.

2.2.7 Software Licensing Information

LWIP TCP/IP Implementation

The software in this camera includes the LWIP TCP/IP implementation. The copyright information for this implementation is as follows:

2.3 Hardware Installation

CAUTION	**Voltage Outside of Specified Range Can Damage the Camera** If the voltage of the input power to the camera is greater than +24 VDC damage to the camera can result. If the voltage is less than 12 VDC, the camera may operate erratically.

CAUTION	**An Incorrect Plug Can Damage the 12-pin Connector** The plug on the cable that you attach to the camera's 12-pin connector must have 12 pins. Use of a smaller plug, such as one with 10 pins or 8 pins, can damage the pins in the camera's 12-pin connector.

To install the camera hardware, follow these steps:

1. Mount a C-mount lens on the camera. Make sure that the lens is screwed onto the lens adapter as far as it will go.
2. Plug one end of an Ethernet cable into the network adapter in your PC and the other end of the cable into the GigE connector on the camera.Plug the 12-pin output connector from your power supply into the 12-pin connector on the camera.
3. Switch on the power supply.
4. The camera will start up and will go through an IP address assignment process. This takes about one minute.
5. Once the IP assignment process is finished, the camera will be ready for use.

   Note that if you have your system set so that a network connection icon appears in the system tray, you may see a yellow exclamation point on the icon. You may also see a message about limited connectivity as shown below. The message about limited connectivity is normal and this situation will have no effect on the camera. You can ignore this message.

2.4 Acquiring Your First Images

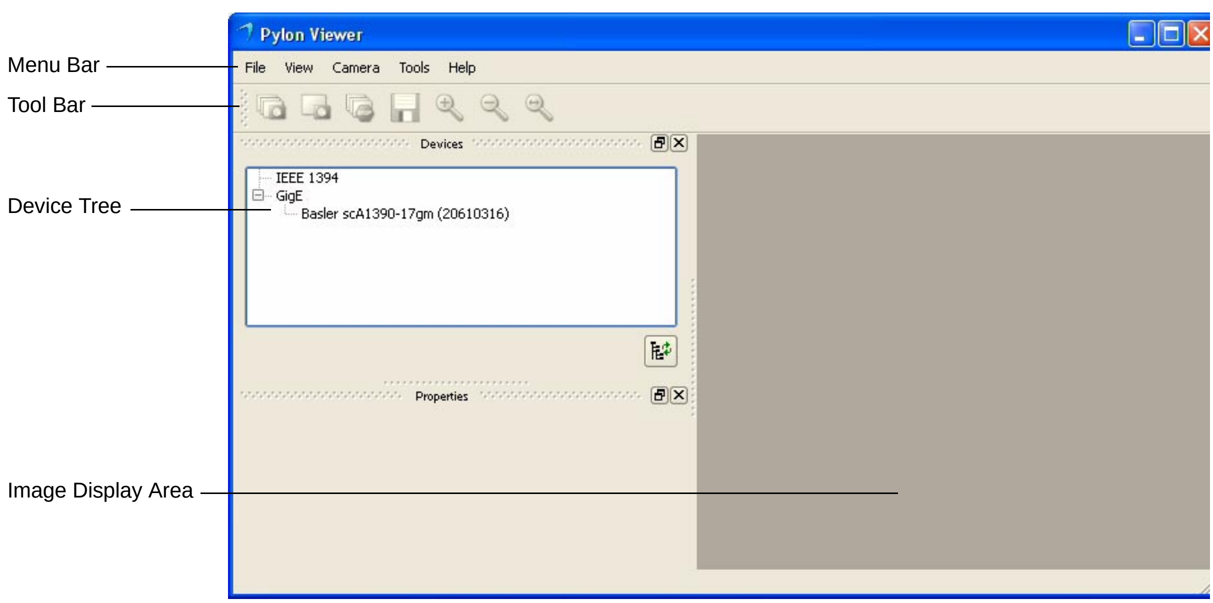
To adjust your camera's settings and to acquire and view your first images, we suggest that you use the pylon Viewer software. The following steps assume that you are using the pylon Viewer.

At this point, many of the camera's settings will be preliminary. Information about how to improve your image quality and to make camera settings more suitable for your specific application appear later in this section of this manual.

Before attempting to acquire images, make sure that the camera software and hardware have been installed as described in the beginning of the Installation section.

To change the camera's settings and acquire images:

1. Put an object within the camera's field of view.
2. Make sure that the object is properly illuminated. Use continuous illumination for this initial setup. If you want to use strobe light in your application, it is better to try this after the initial setup. If you need assistance in determining the optimum illumination for your application, contact Basler technical support using the numbers that appear in the title pages of this manual.
3. To start the pylon Viewer, click the  icon on your computer's desktop. The viewer window will open as shown below.

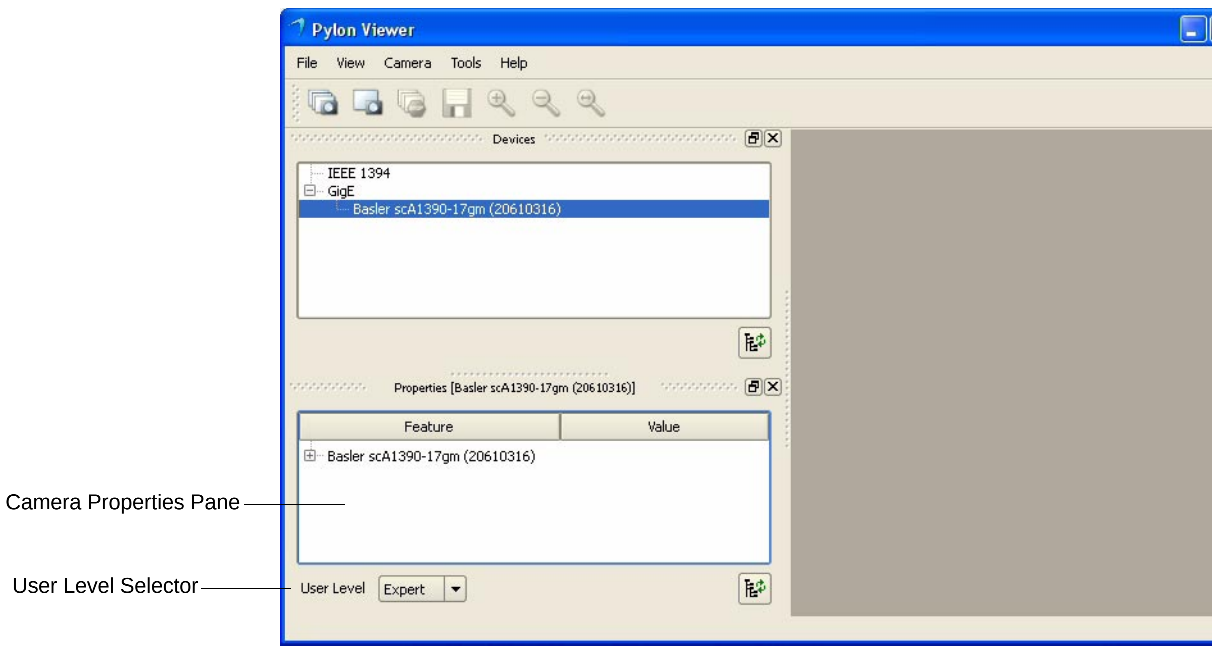


The device tree displays all available devices ordered by their type of interface (i.e., GigE, FireWire). All devices attached to a bus, including your camera, are also shown in the tree.

4. Click on your camera’s name in the device tree to select the camera. A camera properties pane will open in the viewer as shown below.

   You can use the selections in the camera properties pane to adjust all of the cameras settings such as gain, black level, and exposure.

   Notice the user level selector drop down box that now appears in the lower left corner of the properties pane. You can select the beginner, expert, or guru user level. For the beginner level, the properties pane will display only the most basic camera settings. For the expert level, the properties pane will display all of the most commonly used camera settings. And for the guru level, the properties pane will display all camera settings including the most advanced settings.



5. To get an initial feel for how to adjust your image quality, you will make adjustments to the camera lens and you will use the pylon Viewer to make adjustments to the camera’s black level, gain, and exposure time settings. You will also use the viewer to capture images and to see what they look like.
6. Open the lens aperture "halfway" by choosing an intermediate f-number.

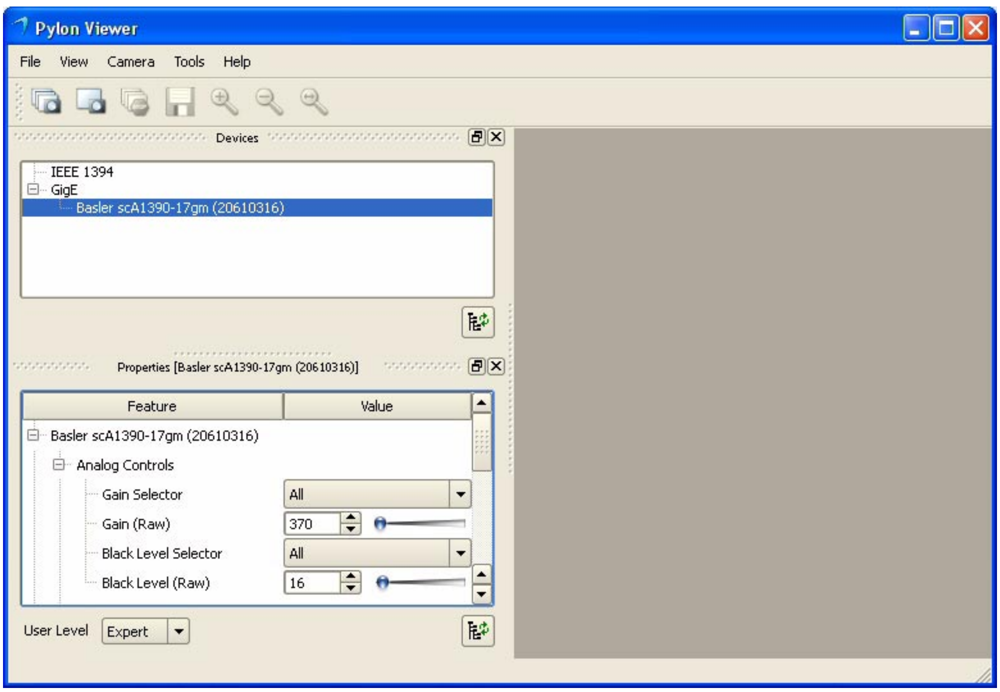
7. Change the **Black Level** setting to a value between 16 and 32 and change the **Gain** setting to its lowest allowed value:

   a. In the **Camera Properties Pane**, click the + **Sign** beside your camera's name. A list of setting categories will appear.

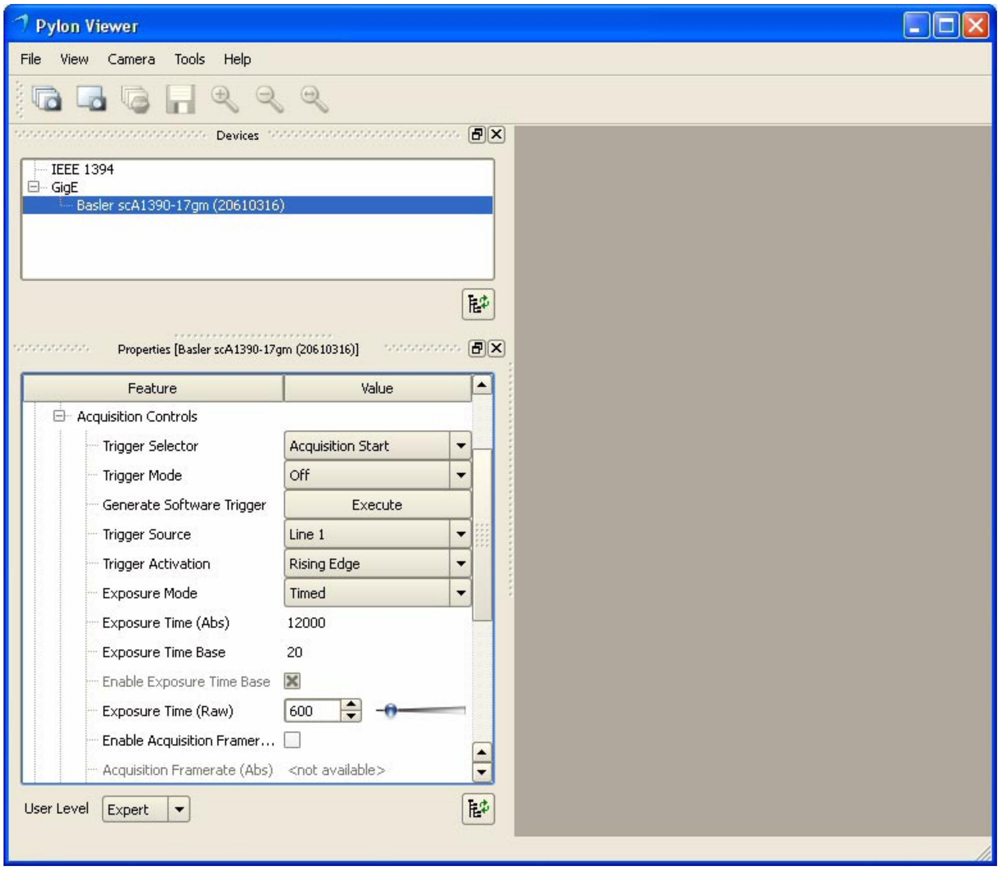
   b. Click the + **Sign** beside the **Analog Controls** category. The controls for setting **Gain** and **Black Level** will appear as shown below.

   You can set the **Black Level** by using the slider that appears next to the value box, by using the up / down arrows that appear next to the value box, or by typing a new value into the box.

   You can set the **Gain** by using the slider that appears next to the value box, by using the up / down arrows that appear next to the value box, or by typing a new value into the box.



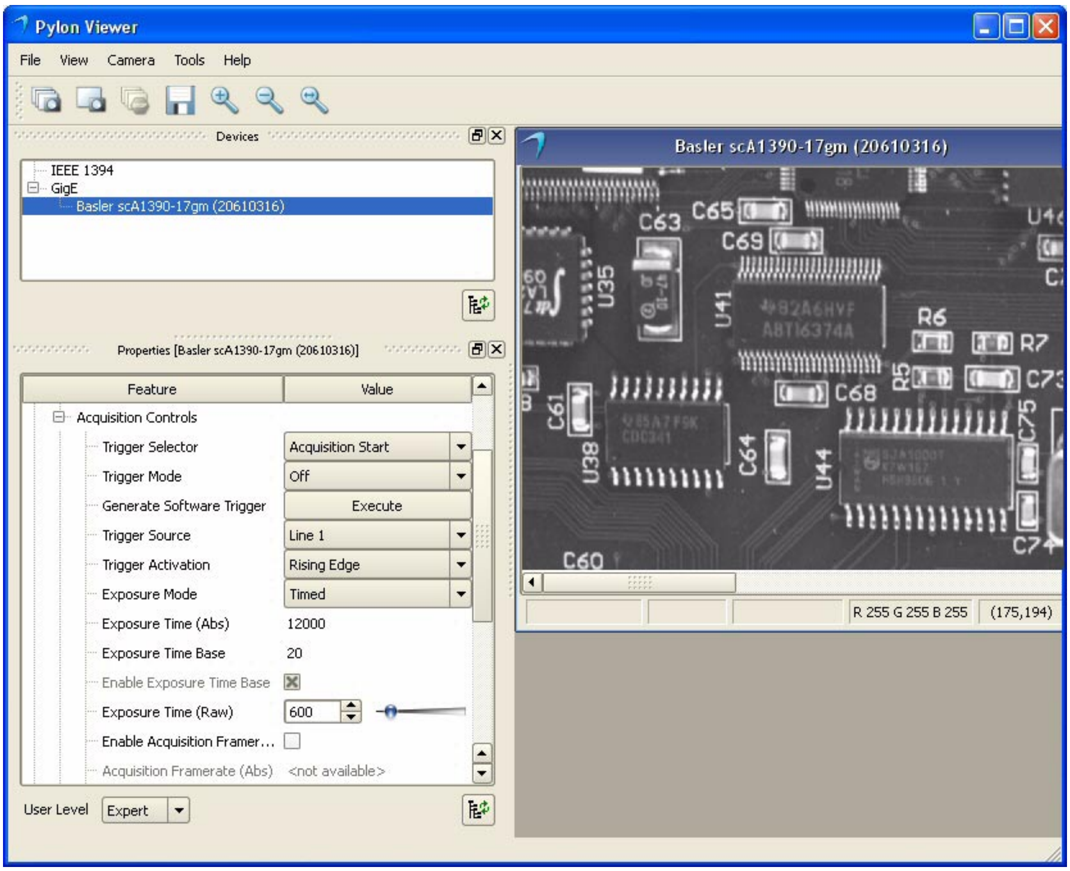
8. Change the **Shutter** setting to its lowest allowed value:
   a. Click the + **Sign** beside the **Acquisition Controls** category. The controls for setting the **Exposure Time** will appear as shown below.
   b. Use the slider to set the Exposure Time to its lowest allowed value.



9. Click the  icon in the tool bar or click **Camera** in the menu bar and select **Continuous Shot** from the drop down menu.

   The camera will begin to acquire images continuously. The acquired images are displayed in an image display window as shown below.

   With the current camera settings, the images you are seeing may be very dark or perhaps even black. The following steps assume that the current images are too dark.



10. Increase the **Exposure Time** setting enough so that your image brightness is almost at the desired level. (The viewer is capturing and displaying images continuously, so you should see the effect of any setting change almost immediately.)
11. If required, increase the **Gain** setting slightly to improve the contrast.

    Note: Make sure that detail is still visible in the brightest parts of the image. Increasing the Gain too much can cause the brightest parts of the image to lose detail. Increasing the Gain too much can also increase the noise in the acquired images.
12. Adjust the focus ring on the lens so that the image is properly focused.
13. Adjust the lens aperture to fine tune the overall image brightness. Also take note of the depth of focus in the acquired images. Adjusting the aperture affects the depth of focus. (The next section contains more information about depth of focus.)
14. Make fine adjustments to the **Black Level** to ensure that detail is still visible in the darkest parts of the acquired images. (The next section contains more information about brightness.)

15. Adjust **Exposure Time**, **Gain**, and **Black Level** (in that order) to further improve the quality of the acquired images.
16. Stop image acquisition by clicking the icon or by clicking **Camera** in the menu bar and selecting **Stop Grab** from the drop down menu.
17. Save the image in the display window by clicking the icon or by clicking **Camera** in the menu bar and selecting **Stop Grab** from the drop down menu

Now that you have made the basic adjustments to your image quality, go on to the next section of the Installation procedure. The next section provides more detailed information about making adjustments to achieve the best image quality.

2.5 Adjusting Image Quality

In the following descriptions, we will discuss image quality in terms of focus, depth of focus, brightness and contrast. You can adjust image quality with regard to these criteria by choosing appropriate settings. However, the "best" image quality will partly depend on the specific requirements of your application and therefore no generally applicable "best" setting can be recommended.

The adjustments will involve the following:

- adjusting the brightness of the illumination
- adjusting the focus
- setting the lens aperture
- setting the black level
- setting the shutter
- setting the gain
- adjusting white balance.

In addition, the quality of an image will be affected by other factors, e.g., by the choice of the lens.

We recommend carrying out all fine adjustments using the illumination that you want to use in your actual application.

Focus:

You will obtain a focused image only if the lens is screwed into the lens adapter of the camera as far as it will go and if the glass surfaces are clean. The object to be imaged must be within the range of focus of the lens.

- You can obtain a focused image by turning the focal ring on the lens.

Depth of Focus:

If the objects you want to image are located at different distances from the camera, you must consider depth of focus. The depth of focus must be sufficiently deep to allow all objects to appear focused in the image.

- You can change the depth of focus by turning the aperture ring on the lens. Closing the lens aperture (turning the aperture ring to higher f-numbers) increases the depth of focus and vice versa.

  Note that closing the aperture decreases the amount of light reaching the camera’s sensor and therefore results in a darker image.

Brightness:

Among the factors determining the brightness of an image are the intensity of the illumination, the setting of the lens aperture, and the settings for black level, exposure time, and gain.

- We recommend that you choose bright illumination if possible, but avoid excessive intensity. This will prevent you from needing to operate the camera using extreme camera settings. A bright but not excessively bright illumination is of central importance to achieving good image quality.
- In images acquired from CCD sensors, excessive brightness will cause artifacts such as smear (white stripes in the image) and blooming (local over-saturation that destroys contrast). You can decrease the proneness for smear and blooming by choosing a diffuse and less intense illumination.
- If illumination of sufficient brightness is not available, you can select a lens that is optimized for light utilization.
- Opening the lens aperture will allow more light to reach the camera’s sensor and will therefore increase the brightness of the image.

  Note that opening the lens aperture also increases the effects of optical aberrations. This causes image distortions and the intensity of light decreases towards the edges of the sensor (vignetting). In addition, the depth of focus decreases.
- You can change the brightness of the image by changing the camera’s black level setting. Normally, you should increase the black level setting only as far as is necessary to make detail visible in the darkest portions of an image. (This is equivalent to avoiding the clipping of the low gray values of noise.)

  Note that high brightness settings will prevent high contrast. We recommend not using brightness settings above 64 when the camera is set for any output format that is greater than 8 bits per pixel.
- You can increase the brightness of the image by increasing the camera’s exposure time setting. With this method, brightness is increased by increasing the amount of photons collected for pixel readout.

  Note that increasing the exposure time setting may decrease the acquisition frame rate. If you are acquiring images of moving objects, increasing the exposure time setting may increase motion blur.
- Increasing the gain will also increase image brightness.

  Note that unless your application requires extreme contrast, you should make sure that detail remains visible in the brightest portions of the image when increasing gain. Note also that noise is increased by increasing gain.

Exposure Time:

The exposure time setting determines the time interval during which the sensor is exposed to light.

- Choose an exposure time setting that takes account of whether you want to acquire images of still or moving objects:
  - If the object is not moving, you can choose a high exposure time setting (i.e., a long exposure interval).

    Note that high exposure time settings may reduce the camera's maximum allowed acquisition frame rate and may cause artifacts to appear in the image.
  - If the object is moving, choose a low exposure time setting to prevent motion blur. As a general rule, choose a short enough exposure time to make sure that the image of the object does not move by more than one pixel during exposure.

Gain:

Gain amplifies each pixel readout by a certain factor. Accordingly, signal and noise are both amplified.

Note that it is not possible to improve the signal-to-noise ratio by increasing gain.

- You can increase the contrast in the image by increasing the camera's gain setting.
- Increasing gain will increase the image brightness.

  Unless your application requires extreme contrast, make sure that detail remains visible in the brightest portions of the image when increasing gain. Note also that noise is increased by increasing gain.
- Set the gain only as high as is necessary.

Contrast:

Strong contrast in an image is obtained when objects of different brightnesses are represented by strongly different grey values. For most applications, optimum contrast is achieved when the image displays a wide range of gray values with fine detail remaining visible even in the darkest and brightest parts of an image. Some applications, however, may require extreme contrast.

- You can increase the contrast in the image by increasing the camera's gain setting. Gain amplifies the pixel readout.
- High black level settings will prevent high contrast. We recommend not using black level settings above 64 (at greater than 8 bit output).
- In images acquired from CCD sensors, contrast can be destroyed by local over-saturation (blooming) if the image brightness is too high. You can decrease the proneness for smear and blooming by choosing a diffuse and less intense illumination.
- Closing the lens aperture not only decreases image brightness but also increases contrast towards the edges of an image.
- If you must use insufficient illumination resulting in dark images, you may notice the blurring influence of noise. If you operate the camera near the high end of its specified temperature range, the effects may be particularly noticeable. You can increase contrast by lowering the operating temperature of the camera.

Note that it is not possible to improve the signal-to-noise ratio by increasing gain. Increasing the gain will increase both signal and noise in equal proportions.

White Balance:

- If you are using a color camera, the object that you use when adjusting the white balance should be a uniform gray and should fill the camera's entire field of view. Your white balance will be correct when images of this object show a uniform gray.
- Changing the black level setting will change the white balance. We therefore recommend that you check the white balance after setting the black level.
- Make sure the image is neither underexposed nor overexposed when checking the white balance.

2.6 Next Steps

We assume that you have succeeded in acquiring images and controlling the camera using the pylon Viewer and that you were able to optimize the image quality.

To meet the requirements of your application, you will likely need to make additional camera settings and to modify previous camera settings.

See the sections in this manual describing camera operation and features for details about additional camera settings. See the Basler pylon Programmer's Guide and API Reference for information about setting and controlling the camera via a GenICam based API. We recommend controlling the camera via the API when taking the next steps.

Contact Basler technical support if you need further assistance. The contact numbers appear on the title page of this manual.

If you have not already done so, implement the typical conditions of operation as required by your application before proceeding with the next steps. In particular, choose the lens and the illumination required by your application.

Before making the additional camera settings, you must know the requirements for your application regarding depth of focus, acquisition frame rate, size of the AOI, and contrast. And you must know what the priorities of the requirements are since some of the settings depend on each other or have opposite effects. For example, a desired high acquisition frame rate may not be reachable with the exposure time set to a high value or with the area of interest set to full resolution.

Your next steps will involve all or some of the following:

- Selecting the pixel format and frame rate.
- Defining an AOI.
- Controlling exposure by selecting a trigger scheme and by setting the exposure time. If you use an external device to supply the trigger, you must also connect the camera to the external device using the I/O cable.
- Controlling the frame rate by using the Acquisition Frame Rate Abs parameter or with trigger signals. If you are using more than one camera on the same network, choose the settings to make optimum use of the available network bandwidth.
- Enabling and parameterizing I/O signals.
- Enabling and parameterizing camera features.

3 Tools for Changing Camera Parameters

This section explains the options available for changing the camera's parameters. The available options let you change parameters either by using standalone tools that access the camera via a GUI or by accessing the camera from within your software application.

3.1 The pylon Viewer

The Basler pylon Viewer is a standalone application that lets you view and change most of the camera's parameter settings via a GUI based interface. The viewer also lets you acquire images, display them, and save them. Using The pylon Viewer software is a very convenient way to get your camera up and running quickly when you are doing your initial camera evaluation or doing a camera design-in for a new project.

The pylon Viewer is included in the pylon Software Development Kit and is also included in the freely available pylon run-time software package.

For more information about using the viewer, see Section 2.4 on page 52.

3.2 The IP Configuration Tool

The Basler IP Configuration Tool is a standalone application that lets you change the IP configuration of the camera via a GUI. The tool will detect all Basler GigE cameras attached to your network and let you make changes to a selected camera

The IP Configuration Tool is included in the pylon Software Development Kit and is also included in the freely available pylon run-time software package.

For more information about using the IP Configuration Tool, see Section 5.4 on page 81.

3.3 The pylon API

You can access all of the camera’s parameters and can control the camera’s full functionality from within your application software by using Basler’s pylon API. The Basler pylon Programmer’s Guide and API Reference contains an introduction to the API and includes information about all of the methods and objects included in the API.

The Basler pylon Software Development Kit (SDK) includes a set of sample programs that illustrate how to use the pylon API to parameterize and operate the camera. These samples include Microsoft® Visual Studio® solution and project files demonstrating how to set up the build environment to build applications based on the API.

4 Network Recommendations

This section describes the recommended adapters and architectures for the network to which your cameras are attached.

4.1 Recommended Network Adapters

The recommended network adapters for use with Basler GigE cameras are the Intel PRO 1000 series. The adapters in the PRO 1000 series have been tested with the cameras and compatibility with these adapters is assured. These adapters are attractively priced and readily available.

The currently available PRO 1000 adapters are listed in Table 4.

Adapter Name	PC Bus Type	Ports	Chipset
PRO/1000 PT Desktop	PCIe (x1)	1	Intel 82572EI
PRO/1000 PT Server	PCIe (x1)	1	Intel 82572EI
PRO/1000 PT Dual Port Server	PCIe (x4)	2	Intel 82571EB
PRO/1000 PT Quad Port Server	PCIe (x4)	4	Intel 82571EB
PRO/1000 GT Desktop	PCI	1	Intel 82541PI
PRO/1000 MT Server	PCI-X	1	Intel 82545GM

Table 4: Recommended Network Adapters

Network Adapters and the Basler Network Drivers

Although Basler recommends using PRO 1000 series adapters with our GigE cameras, the cameras will work with any Fast Ethernet (100 Mbit/s) or Gigabit Ethernet (1000 MBit/s) compatible network adapter card.

Two Basler network drivers are available, the Basler FIlter Driver and the Basler Performance Driver.

If the adapter you are using is not a PRO series adapter and does not have one of the Intel chipsets listed in Table 4, you must install the Basler Filter Driver. Your camera will use the filter driver to communicate via the adapter. The advantage of the filter driver is that it will work with any Fast Ethernet or Gigabit Ethernet compatible adapter. (If you will be using the filter driver with your adapter, there is no need to install the performance driver.)

If you are using a PRO series adapter or if your adapter has one of the Intel chipsets listed in Table 4, you can install the Basler Performance Driver. Your camera will use the performance driver to communicate via the network adapter. The advantage of the performance driver is that is requires significantly less CPU load to service the network communications between your camera and your PC. It also has a more robust packet resend mechanism.

For more information about installing the network drivers, see Section 2.2 on page 32.

4.2 Recommended Network Architectures

4.2.1 Peer-to-peer Network Architecture

A strongly recommended network architecture is direct peer-to-peer connection between your cameras and your host PC. As shown in Figure 25, the cameras can be connected to individual gigabit network adapters in the host PC or to a multiport adapter in the PC.

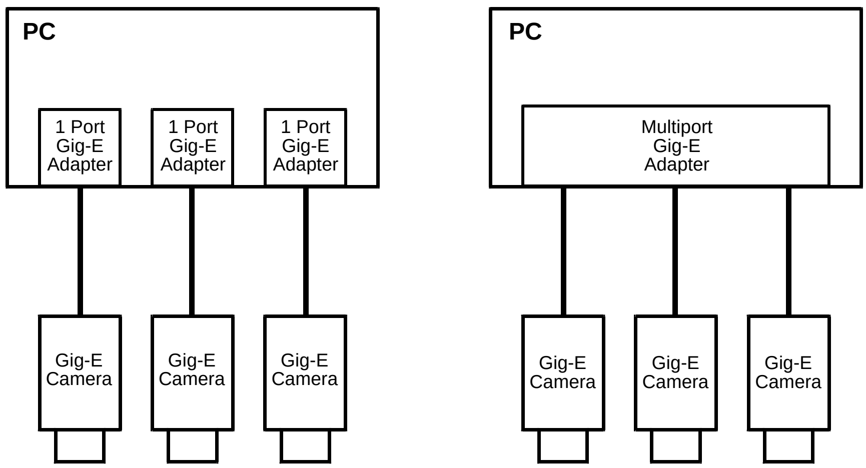


Fig. 25: Peer-to-peer Connections

The main advantage of the direct peer-to-peer architecture is that each camera has an individual connection to the host PC and thus each camera has the full connection bandwidth available for transmitting acquired images. This means that you can operate each camera at its full acquisition rate and not worry about sharing available network bandwidth between cameras.

The disadvantage of the peer-to-peer configuration is that it limits the number of cameras that can be connected to a single PC.

4.2.2 Connecting Via Network Switches

A second recommended network architecture involves connecting your cameras to the Gigabit Ethernet (GigE) adapter(s) in your host PC via network switches. As shown in Figure 26, the cameras can be connected to individual GigE network adapters in the host PC or to a multiport adapter in the PC. Note that the figure below only depicts simple schemes for connecting cameras to a PC via network switches.

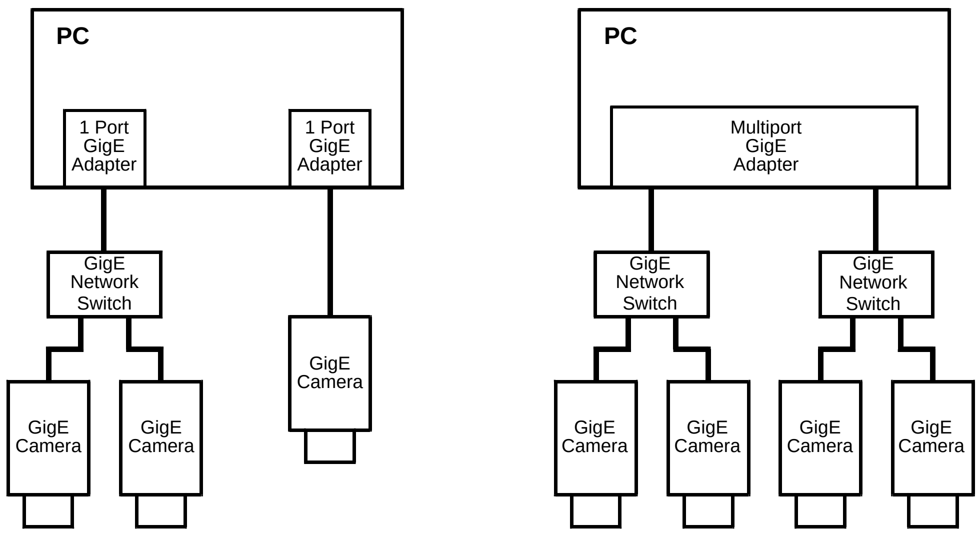


Fig. 26: Network Connections via Switches

One advantage of an architecture using switches is that it allows many cameras to connect to a single host PC. It also allows longer overall cable lengths because the cables between devices can each be up to 100 meters long.

The disadvantage of an architecture using switches is that the data from several cameras can end up passing through a single network connection and thus the cameras must share the bandwidth available on this single path. This situation is illustrated in Figure 27. The four cameras each have a connection to the network switch, but the switch only has a single connection to the PC. The four cameras must share the bandwidth available on the single path between the switch and the adapter. The bandwidth available on this single GigE path is about 125 MByte/s.

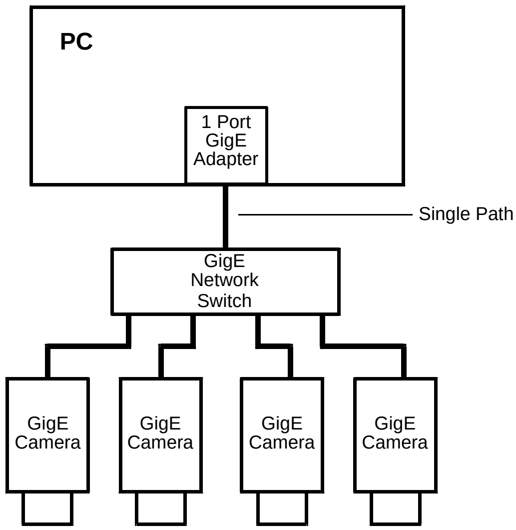


Fig. 27: Multiple Cameras Converging to a Single Path

For more information about managing network bandwidth when using multiple cameras on a single network path, see Section 6 on page 85.

Network Switch Issues

When selecting GigE network switches for use in the type of network described above, there are several issues you must keep in mind. First is that the switch must be able to handle large packets (also known as "jumbo packets" or "jumbo frames"). The typical maximum packet size on Ethernet devices used in the past was 1.5 kB. With newer "jumbo frame capable" devices, the maximum packet size can be up to 16 kB. Basler GigE cameras and the recommended Intel Pro 1000 adapters can both handle jumbo frames. For maximum network efficiency, your camera should be set to used the largest packet size that your network can handle. If you select a network switch that can only handle a small packet size, you will limit network efficiency.

A second issue involving the network switch is buffer capacity. In the situation where multiple cameras are attached to a switch, the switch must have enough buffer capacity to hold the incoming data from the cameras while it transmits the data out in an orderly fashion on the single outgoing line. In general, more buffer capacity is better.

4.3 PC Data Bus Issues

If you are connecting multiple cameras to your host PC, either through direct peer-to-peer connections or through network switches, you must be aware of the data bus type used in the PC. The PCI bus typical of older PCs is theoretically capable of handling 132 MByte/s of data. But in practice, the capacity of the PCI bus is lower and the bus bandwidth is shared by the network adapter and many other devices installed in the PC. To ensure adequate bandwidth on the PC's data bus, use of a PC with a PCI express data bus is recommended.

5 Camera and Network Adapter IP Configuration

This section describes the default IP configuration for your network adapter and camera. It also describes how to change the IP configuration on your camera and on your network adapter.

Note

This section of the user's manual assumes that you are familiar with basic Ethernet network concepts and with TCP/IP concepts such as IP addresses, subnet masks, and default gateways. If you are not, you should take some time to familiarize yourself with this basic information.

The following website provides comprehensive information about TCP/IP as it applies to Windows operating systems:

http://www.microsoft.com/technet/network/evaluate/technol/tcpipfund/tcpipfund.mspx

This section also assumes that you are familiar with using basic Windows operating system tools such as the Network Connections window to access your network adapter.

5.1 Network Adapter IP Behavior

The two most common ways to set the IP configuration on network adapters are:

- as "Fixed Address"
- as "DHCP / Alternate Configuration = APIPA (Automatic Private IP Addressing)"

When an adapter is set for Fixed Address, it will simply use a fixed address that has been assigned to the adapter by the user.

When an adapter is set for DHCP / Alternate Configuration = APIPA, it will do the following:

- It will first attempt to obtain an IP address from a Dynamic Host Configuration Protocol (DHCP) server. If a DHCP server is available, it will obtain an IP address from the server and use it.
- If no DHCP server is available, the adapter will use its alternate configuration. The alternate configuration will be for the adapter to use the Automatic Private IP Address routine to assign itself an IP address.

The default setting for most network adapters is DHCP / Alternate Configuration = APIPA. In most cases, the adapter used with your cameras will not have a DHCP server available. So with the default settings, the adapter will end up using automatic IP addressing to assign itself an IP address.

Note

There is another adapter IP configuration available called "DHCP / Alternate Configuration = Fixed Address", but this configuration is seldom used.

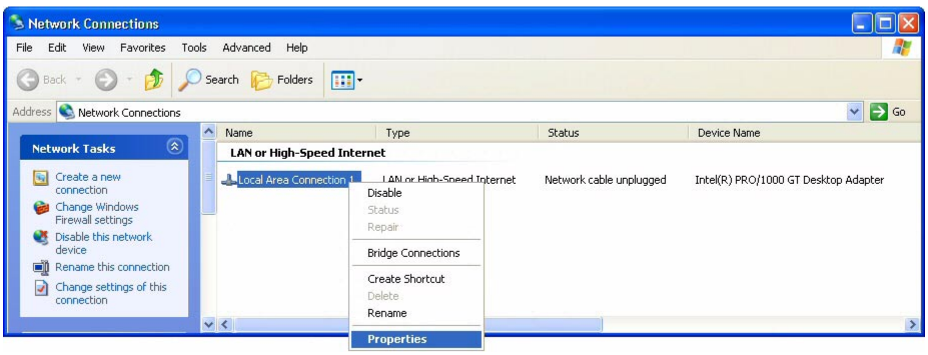
For more information about the Automatic Private IP Address routine, see Section 5.3 on page 79.

5.2 Changing a Network Adapter's IP Configuration

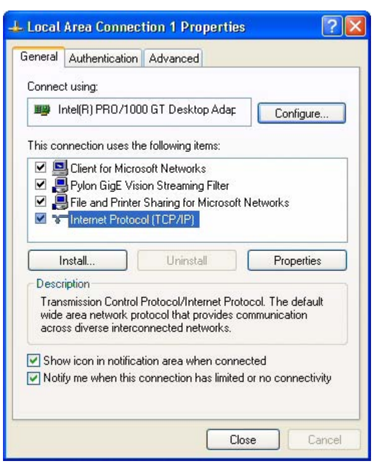
5.2.1 Setting an Adapter to Use a Fixed IP Address

You can configure a network adapter to use a fixed IP address by doing the following:

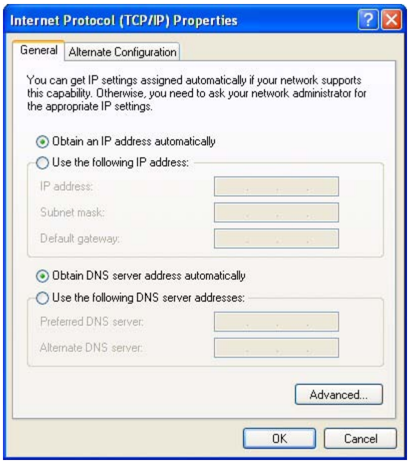
1. Open a **Network Connections** window.
   a. Find the connection for the adapter you want to configure. Right click on the name of the connection and select **Properties** from the drop down menu as shown below.



2. A **Local Area Connection Properties** window will open as shown below. Make sure that **Internet Protocol (TCP/IP)** is highlighted and click the **Properties** button.

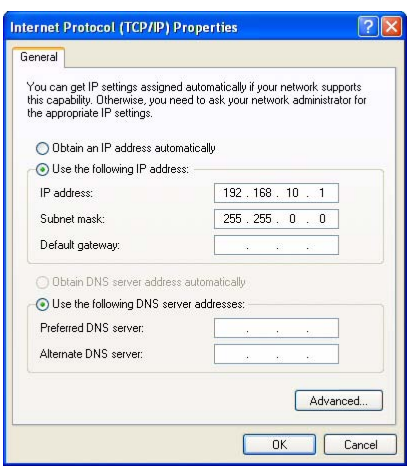


3. An **Internet Protocol (TCP/IP) Properties** window will open and the **General** tab will be selected as shown below.



4. Click the radio button next to **Use the following IP address**. The window will change and will now allow you to enter IP address information.
   a. Enter your desired IP address and subnet mask. The figure below shows the window with typical values entered.

      You can also enter a default gateway if desired, however, a default gateway is not normally needed.
   b. If you will be using a domain name server (DNS), enter the appropriate information. (A domain name server is not normally needed.)
   c. Click the **OK** button. The **Internet Protocol (TCP/IP) Properties** window will close.
   d. Click the **Close** button on the **Local Area Connection Properties** window.
   e. Your system will wait for several seconds while the new settings take effect. Once the new settings are in place, the **Local Area Connection Properties** window will close.



Note

When you configure an adapter to use a fixed address, there are some things that you must keep in mind:

- If your PC has multiple network adapters, each adapter must be in a different subnet.
- The recommended range for fixed IP addresses is from 172.16.0.1 to 172.32.255.254 and from 192.168.0.1 to 192.168.255.254. These address ranges have been reserved for private use according to IP standards.
- If you are assigning fixed IP addresses to you cameras, keep in mind that for a camera to communicate properly with a network adapter, it must be in the same subnet as the adapter to which it is attached.

Tip

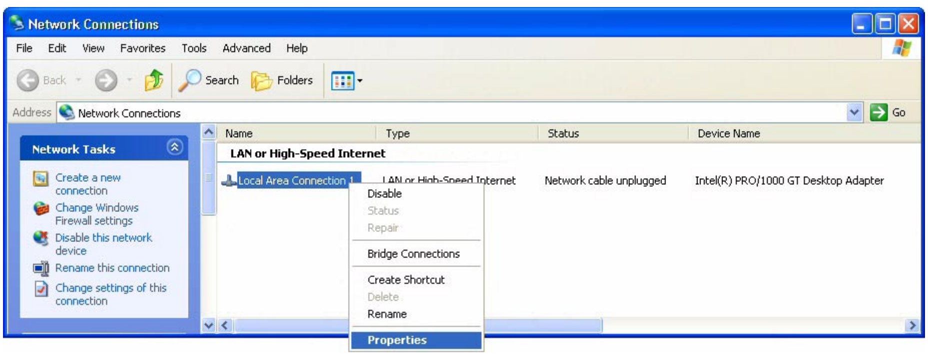
There is a convenient "trick" that is handy during your initial camera design-in process or when working with cameras in your lab. You can set your network adapter to a fixed address in the automatic IP address range (169.254.0.0 to 169.254.255.255) with a subnet mask of 255.255.0.0 and you can set your camera(s) for automatic IP address assignment. With these settings, a camera and an adapter can establish a network connection very quickly. This can save you some time if you are connecting and disconnecting cameras or switching the system on and off as you would during design-in.

5.2.2 Setting an Adapter to Use DHCP / Alternate Configuration = APIPA

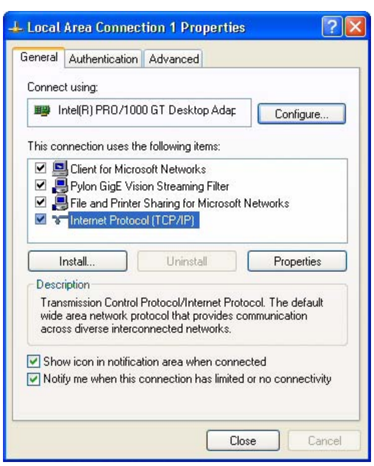
When a network adapter is set for DHCP / Alternate Configuration = APIPA, it will first try to find a DHCP server and to obtain an IP address from the server. If no DHCP server is available, the adapter will revert to the "alternate configuration." The alternate configuration will be for the adapter to use the Automatic Private IIP Address routine to assign itself an IP address.

You can configure a network adapter for DHCP / Alternate Configuration = APIPA by doing the following:

1. Open a **Network Connections** window.
   a. Find the connection for the network adapter you are using with the camera. Right click on the name of the connection and select **Properties** from the drop down menu as shown below.

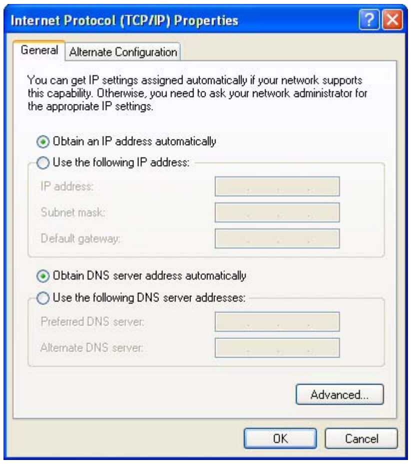


2. A **LAN Connection Properties** window will open as shown below. Make sure that Internet Protocol (TCP/IP) is highlighted and click the **Properties** button.



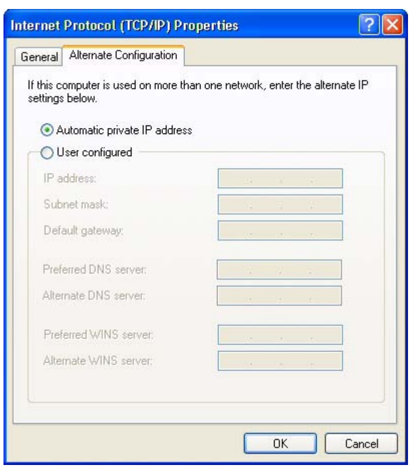
3. An **Internet Protocol Properties** window will open as shown below.

   a. On the **General** tab, make sure that the **Obtain an IP address automatically** radio button is selected. (This sets the adapter to check for a DHCP server as its first choice.)



4. Click on the **Alternate Configuration** tab. The settings on this tab are used to set the alternate configuration that the adapter will use if no DHCP server is found.

   a. Select the **Automatic private IP address** radio button as shown below.



5. Click the **OK** button on the **Internet Protocol Properties** window.
6. Click the **Close** button on the **LAN Connection Properties** window.

Note

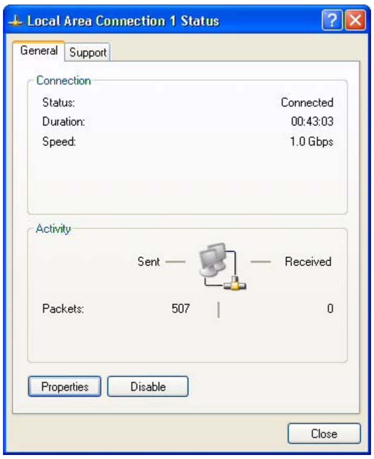
There is a limitation you must be aware of when set adapters to use Automatic Private IP Addressing (APIPA) as an alternate configuration. If a PC is equipped with multiple network adapters, APIPA can only be used on one of the adapters. If APIPA is enabled on more than one adapter, the network will not operate properly.

Note that if your PC is equipped with a multiport network adapter board, each port is considered to be the equivalent of a separate adapter. APIPA should be enabled on only one port.

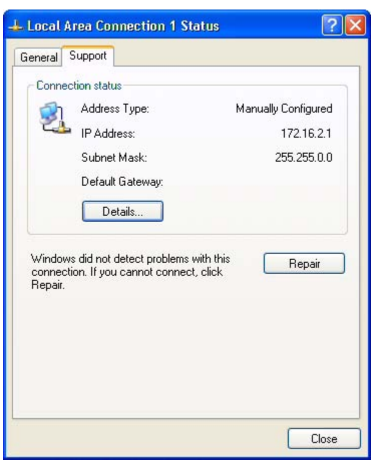
5.2.3 Checking a Network Adapter's IP Address

You can check the current IP address of a network adapter by doing the following.

1. Open a **Network Connections** window.
2. Find the connection for the adapter you want to check and make sure that the status of the connection is shown as "connected". (If the status is "disconnected," this procedure will not work. Double click on the name of the connection.
   A **LAN Connection Status** window will open as shown below.



3. Click on the Support Tab. The IP address information for the adapter will be displayed as shown below.



5.3 Camera IP Behavior

When a camera is powered on or reset, it exhibits the following behavior when it tries to connect to an Ethernet network:

- It checks to see if the camera has been configured to use a persistent (fixed) IP address. If the camera has a persistent IP address configured, it will use the persistent IP address.
- If no persistent IP address is configured, the camera will check to see if it is configured to use a Dynamic Host Configuration Protocol (DHCP) server to obtain an IP address and to see if a DHCP server is available. If the camera is configured to use a DHCP server and a DHCP server is available, the camera will obtain an IP address from the server.
- If no persistent IP address is configured and no address is obtained from a DHCP server, the camera will perform an Automatic Private IP Addressing (APIPA) routine and will assign itself an IP address.

Given this startup sequence, a new camera set to its default settings will do the following:

- The camera checks to see if it has been assigned a persistent (fixed) IP address. With the default settings, the camera is not assigned a persistent IP address. So no persistent address will be found.
- Because the camera does not have a persistent IP address assigned, it next checks to see if a DHCP server is available. Assuming that the camera is directly connected to a PC in peer-to-peer fashion, no DHCP server will be present. So an IP address will not be assigned to the camera by a DHCP server.
- Because the camera does not have a persistent IP address assigned and a DHCP server is not available, the camera will start an APIPA routine. When the routine is complete, the camera will be connected to the network and will have an auto-assigned IP address.

  Note that the search for a DHCP server and the performance of an APIPA routine can take up to one minute to complete.

Automatic Private IP Addressing (APIPA)

The Automatic Private IP Address routine is a network standard that dictates how an IP address will be assigned to a network adapter and to the devices connected to the adapter when no other means of address assignment is available. In essence, the adapter or the device will assign itself an IP address in the range of 169.254.0.0 to 169.254.255.255 with a subnet mask of 255.255.0.0. As part of the routine, the network adapter and the devices attached to the adapter negotiate to make sure that there are no duplicate address assignments and that the adapter and the devices are all on the same subnet.

For more detailed information about APIPA, you can visit:
http://www.duxcw.com/faq/network/autiop.htm

Note

For auto IP assignment to work correctly, the network adapter that the camera is plugged into must also be set for auto IP assignment or it must be set for a fixed address in the auto IP address range.

Note

If you have multiple network adapters in your PC, only one adapter can be set to use auto IP assignment. If more than one adapter is set to use auto assignment, auto assignment will not work correctly and the cameras **will not** be able to connect to the network.

In the case of multiple network adapters, it is best to assign fixed IP addresses to the adapters and to the cameras. You can also set the cameras and the adapters for DHCP addressing and install a DHCP server on your network.

The next sections in this manual describe how to change the IP configuration on the cameras and the network adapters.

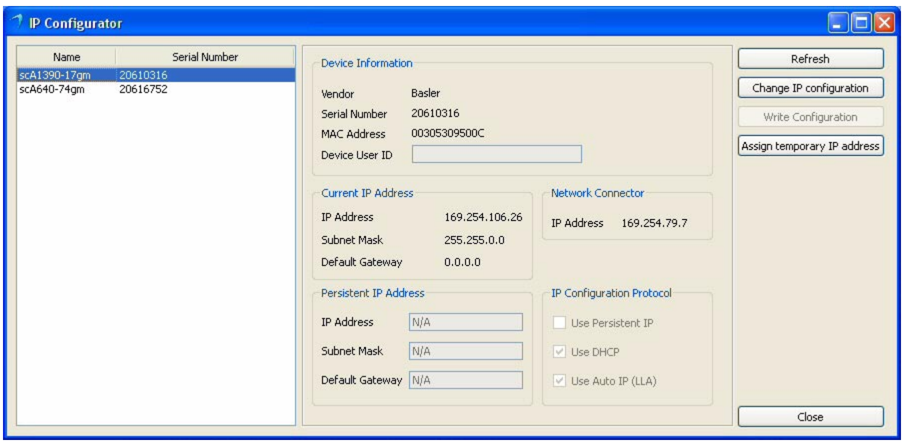
5.4 Changing a Camera's IP Configuration

An application called the IP Configuration Tool is included as part of the pylon driver installation package. The IP Configuration Tool lets you make changes to the IP configuration of your camera.

To start the IP Configuration Tool:

- Double click the **pylon IP Configuration Tool** icon on your desktop
- Or click **Start**, click **All Programs**, click **Basler Vision Technologies**, click **Pylon x.x**, click **Pylon IP Configuration Tool**.

The tool will start and an **IP Configurator** window will open as shown below.



When the tool starts, it scans for cameras attached to the PC's network adapters. All cameras detected will be listed by model name and serial number in a box on the left side of the **IP Configurator** window. To select a camera to work with, simple click on its name in the list.

When you select a camera, the tool will display IP configuration and other basic information for the camera in various boxes:

- The **Device Information** box will display the Vendor's name, the Serial Number, the MAC address, and the Device User ID (if one has been assigned) for the selected camera.
- When the tool is in edit mode, the **Current IP Address** box will display the current IP Address, Subnet Mask, and Default Gateway for the selected camera. (When the tool is in list mode, this information is not available and N/A is displayed.)
- If the selected camera has a persistent (fixed) IP Address assigned to it, the **Persistent IP Address** box will display the current persistent IP Address, Subnet Mask, and Default Gateway.
- The **Network Connector** box will display the IP Address for the network adapter to which the selected camera is attached.
- The **IP Configuration Protocol** box will display the current IP configuration of the selected camera.

If you would like to make sure that all of the displayed information is current, click the **Refresh** button. The tool will rescan the cameras and update the displayed information.

The IP Configuration Tool has two modes, List Mode and Edit mode:

- When the tool is first opened it is in **List** mode. In list mode, the tool can display a list of detected cameras, can display information about the camera selected in the list, and will let you assign a temporary IP address to the selected camera. When in list mode, the tool does not have a control channel open to any of the detected cameras. In list mode, the tool communicates with the cameras at a very low level.
- When the **Change IP Configuration** button is pressed, the tool will enter **Edit** mode. In edit mode, the tool will open a control channel to the selected camera. Once the control channel is open, the tool can make changes to things such as the selected camera's persistent IP address or Device User ID. Clicking the **Write Configuration** button takes the tool out of Edit mode and places it in List mode.

Assigning a Temporary IP Address to a Camera

You can use the IP Configuration Tool to assign a temporary IP address to a selected camera. Once a temporary IP address has been assigned to a camera, the camera will retain and use the temporary IP address until you do one of the following:

- Perform a camera reset or switch the camera off and back on.
- Assign a different temporary IP address to the camera.
- Use the tool to make permanent changes the camera's IP configuration.

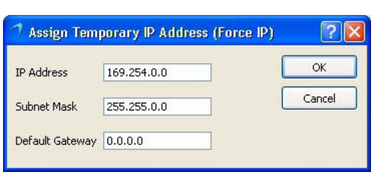
A situation where you might want to assign a temporary IP address to a camera would be, for example, if you temporarily moved the camera from the PC where it is normally connected to another PC that is in a different subnet.

To assign a temporary IP address to a camera:

1. Click on the camera's name in the list at the left side of the **IP Configurator** window.
2. Click the **Assign Temporary IP Address** button.
3. An **Assign Temporary IP Address (Force IP)** window will open as shown below.
   a. Enter your desired IP Address, Subnet mask, and Default Gateway.

      (Keep in mind that for the camera to communicate properly with the network adapter to which it is attached, the camera must be in the same subnet as the adapter and the camera must have a unique IP address.)

   b. Click the **OK** button.
   c. The selected camera's current IP address will immediately be changed and the IP Configurator window will be updated to reflect the changes.



Making Changes to the Camera's IP Configuration

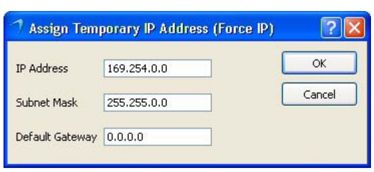
You can use the IP Configuration Tool to make permanent changes to the camera's IP configuration. (Permanent means that the changes will stay in place even when the camera is reset or switched off and back on.)

If you want to change the IP configuration of the camera:

1. Click on the **Change IP Configuration** button.
   a. In some cases (such as when the camera's current IP address is misconfigured), a **Force IP** window will appear as shown below. The message in the window will indicate that you must first assign a temporary IP address. If you see this message, go to step 2.
   b. If you do not see a message about assigning a temporary IP address, go to step 3.

2. Click the OK button in the **Force IP** window. An **Assign Temporary IP Address (Force IP)** window will open as shown below.
   a. Enter an IP Address, Subnet Mask, and Default Gateway. (If you are using the camera in a peer-to-peer network, you do not normally need to enter a default gateway.)

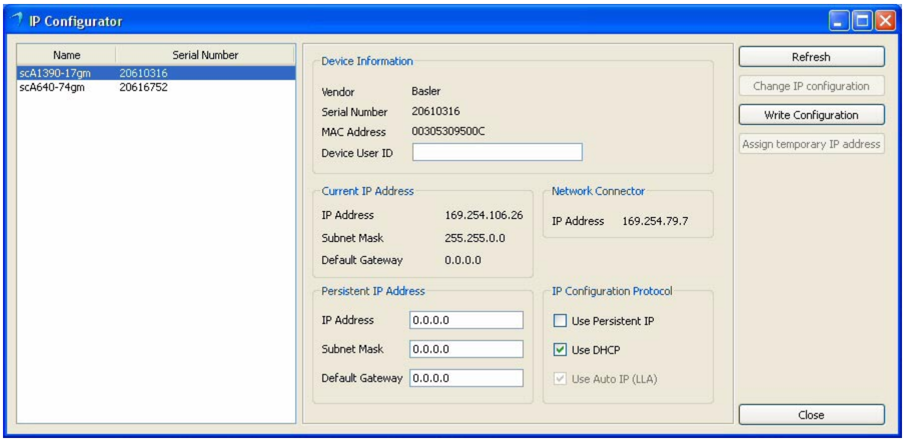
      (Keep in mind that for the camera to communicate properly with the network adapter to which it is attached, the camera must be in the same subnet as the adapter and the camera must have a unique IP address.)
   b. Click the **OK** button.
   c. The camera's IP address will be changed and the **Assign Temporary IP Address (Force IP)** window will close.
   d. Go on to step 3.



3. The **IP configurator** window will enter edit mode and will now allow you to enter new IP values as shown below.
   a. If desired, enter a Device User ID for the camera.
   b. If you want to assign the camera a persistent (fixed) IP address, enter an IP address, a subnet mask, and a default gateway.

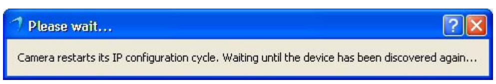
      Also make sure that the **Use Persistent IP** check box is checked.

      (Before you assign a persistent IP address, you should read the tips that appear in the note at the end of this procedure.)
   c. If you want the camera to use DHCP address assignment (i.e., to obtain an IP address from a DHCP server attached to the same network as the camera), make sure that the **Use DHCP** check box is checked and that the **Use Persistent IP** check box **is not** checked.



4. When you are finished making changes, click the Write Configuration button.

   A message will appear indicating that the camera is restarting its IP configuration cycle as shown below. When this message disappears the changes to the IP configuration will be in place and the **IP Configurator** window will now show the changed settings.



Note

When you configure a camera to use either a temporary or a fixed address, there are some things that you must keep in mind:

- For a camera to communicate properly, it must be in the same subnet as the adapter to which it is connected.
- The camera must have an IP address that is unique within the network.
- The recommended range for fixed IP addresses is from 172.16.0.1 to 172.32.255.254 and from 192.168.0.1 to 192.168.255.254. These address ranges have been reserved for private use according to IP standards.
- If your PC has multiple network adapters, each adapter must be in a different subnet.

6 Basler Network Drivers and Parameters

This section describes the Basler network drivers available for your camera and provides detailed information about the parameters associated with the drivers.

Two network drivers are available for the network adapter used with your GigE cameras:

- The **Basler filter driver** is a basic GigE Vision network driver that is compatible with all network adapters. The advantage of this driver is its extensive compatibility.
- The **Basler performance driver** is a hardware specific GigE Vision network driver. The driver is only compatible with network adapters that use specific Intel chipsets. The advantage of the performance driver is that it significantly lowers the CPU load needed to service the network traffic between the PC and the camera(s). It also has a more robust packet resend mechanism.

Note

During the installation process you should have installed either the filter driver or the performance driver.

For more information about compatible Intel chipsets, see Section 4.1 on page 65.

For more information about installing the network drivers, see Section 2.2 on page 32.

6.1 The Basler Filter Driver

The Basler filter driver is a basic driver GigE Vision network driver. It is designed to be compatible with most network adapter cards.

The functionality of the filter driver is relatively simple. For each frame, the driver checks the order of the incoming packets. If the driver detects that a packet or a group of packets is missing, it will wait for a specified period of time to see if the missing packet or group of packets arrives. If the packet or group does not arrive within the specified period, the driver will send a resend request for the missing packet or group of packets.

The parameters associated with the filter driver are described below.

Enable Resend - Enables or disables the packet resend mechanism.

If packet resend is disabled and the filter driver detects that a packet has been lost during transmission, the grab result for the returned buffer holding the image will indicate that the grab failed and the image will be incomplete.

If packet resend is enabled and the driver detects that a packet has been lost during transmission, the driver will send a resend request to the camera. If the camera still has the packet in its buffer, it will resend the packet. If there are several lost packets in a row, the resend requests will be combined.

Packet Timeout - The Packet Timeout parameter defines how long (in milliseconds) the filter driver will wait for the next expected packet before it initiates a resend request.

Frame Retention - The Frame Retention parameter sets the timeout (in milliseconds) for the frame retention timer. Whenever the filter driver detects the leader for a frame, the frame retention timer starts. The timer resets after each packet in the frame is received and will timeout after the last packet is received. If the timer times out at any time before the last packet is received, the buffer for the frame will be released and will be indicated as an unsuccessful grab.

You can set the filer driver parameter values from within your application software by using the pylon API. The following code snippet illustrates using the API to read and write the parameter values:

```
// Enable Resend
Camera_t::StreamGrabber_t &StreamGrabber =
    dynamic_cast<Camera_t::StreamGrabber_t&>( *Camera.GetStreamGrabber(0) );
Camera_t::StreamParams_t &StreamParams =
    dynamic_cast<Camera_t::StreamParams_t&>( *StreamGrabber.GetNodeMap() );
StreamParams.EnableResend.SetValue(false); // disable resends

// Packet Timeout/FrameRetention
Camera_t::StreamGrabber_t &StreamGrabber =
    dynamic_cast<Camera_t::StreamGrabber_t&>( *Camera.GetStreamGrabber(0) );
Camera_t::StreamParams_t &StreamParams =
    dynamic_cast<Camera_t::StreamParams_t&>( *StreamGrabber.GetNodeMap() );
StreamParams.PacketTimeout.SetValue( 40 );
StreamParams.FrameRetention.SetValue( 200 );
```

For detailed information about using the pylon API, refer to the Basler pylon Programmer's Guide and API Reference.

You can also use the Basler pylon Viewer application to easily set the parameters.

For more information about the pylon Viewer, see Section 2.2 on page 32and Section 2.4 on page 52.

6.2 The Basler Performance Driver

The Basler performance driver is a hardware specific GigE Vision network driver compatible with network adapters that use specific Intel chipsets. The main advantage of the performance driver is that it significantly lowers the CPU load needed to service the network traffic between the PC and the camera(s). It also has a more robust packet resend mechanism.

For more information about compatible Intel chipsets, see Section 4.1 on page 65.

The performance driver uses two distinct "resend mechanisms" to trigger resend requests for missing packets:

- The threshold resend mechanism
- The timeout resend mechanism

The mechanisms are independent from each other and can be used separately. However, for maximum efficiency and for ensuring that resend requests will be sent for all missing packets, we recommend using both resend mechanisms in a specific, optimized combination, as provided by the parameter default values.

The performance driver's parameter values determine how the resend mechanisms act and how they relate to each other. You can set the parameter values by using the pylon Viewer or from within your application software by using the pylon API.

Note

The parameter default values will provide for the following:

- The threshold resend mechanism precedes the timeout resend mechanism. This ensures that a resend request is sent for every missing packet, even at very high rates of arriving packets.
- The timeout resend mechanism will be effective for those missing packets that were not resent after the first resend request.

We strongly recommend using the default parameter settings. Only users with the necessary expertise should change the default parameter values.

The Basler performance driver uses a "receive window" to check the status of packets. The check for missing packets is made as packets enter the receive window. If a packet arrives from higher in the sequence of packets than expected, the preceding skipped packet or packets are detected as missing. For example, suppose packet (n-1) has entered the receive window and is immediately followed by packet (n+1). In this case, as soon as packet (n+1) enters the receive window, packet n will be detected as missing.

General Parameters

Enable Resend - Enables the packet resend mechanisms.

If the Enable Resend parameter is set to false, the resend mechanisms are disabled. The performance driver will not check for missing packets and will not send resend requests to the camera.

If the Enable Resend parameter is set to true, the resend mechanisms are enabled. The performance driver will check for missing packets. Depending on the parameter settings and the resend response, the driver will send one or several resend requests to the camera.

Receive Window Size - Sets the size of the receive window.

Threshold Resend Mechanism Parameters

The threshold resend request mechanism is illustrated in Figure 28 where the following assumptions are made:

- Packets 997, 998, and 999 are missing from the stream of packets.
- Packet 1002 is missing from the stream of packets.

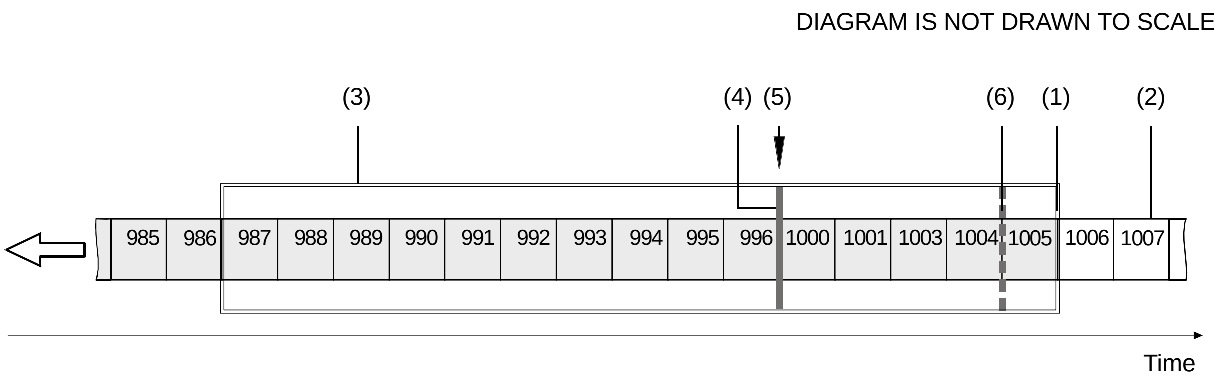


Fig. 28: Example of a Receive Window with Resend Request Threshold & Resend Request Batching Threshold

(1) Front end of the receive window. Missing packets are detected here.

(2) Stream of packets. Gray indicates that the status was checked as the packet entered the receive window. White indicates that the status has not yet been checked.

(3) Receive window of the performance driver.

(4) Threshold for sending resend requests (resend request threshold).

(5) A separate resend request is sent for each packets 997, 998, and 999.

(6) Threshold for batching resend requests for consecutive missing packets (resend request batching threshold). Only one resend request will be sent for the consecutive missing packets.

Resend Request Threshold - This parameter determines the location of the resend request threshold within the receive window as shown in Figure 28. The parameter value is in per cent of the width of the receive window. In Figure 28 the resend request threshold is set at 33.33% of the width of the receive window.

A stream of packets advances packet by packet beyond the resend request threshold (i.e. to the left of the resend request threshold in Figure 28). As soon as the position where a packet is missing advances beyond the resend request threshold, a resend request is sent for the missing packet.

In the example shown in Figure 28, packets 987 to 1005 are within the receive window and packets 997 to 999 and 1002 were detected as missing. In the situation shown, a resend request is sent to the camera for each of the missing consecutive packets 997 to 999. The resend requests are sent after packet 996 - the last packet of the intact sequence of packets - has advanced beyond the resend request threshold and before packet 1000 - the next packet in the stream of packets - can advance beyond the resend request threshold. Similarly, a resend request will be sent for missing packet 1002 after packet 1001 has advanced beyond the resend request threshold and before packet 1003 can advance beyond the resend request threshold.

Resend Request Batching - This parameter determines the location of the resend request batching threshold in the receive window (Figure 28). The parameter value is in per cent of a span that starts with the resend request threshold and ends with the front end of the receive window. The maximum allowed parameter value is 100. In Figure 28 the resend request batching threshold is set at 80% of the span.

The resend request batching threshold relates to consecutive missing packets, i.e., to a continuous sequence of missing packets. Resend request batching allows grouping of consecutive missing packets for a single resend request rather than sending a sequence of resend requests where each resend request relates to just one missing packet.

The location of the resend request batching threshold determines the maximum number of consecutive missing packets that can be grouped together for a single resend request. The maximum number corresponds to the number of packets that fit into the span between the resend request threshold and the resend request batching threshold plus one.

If the Resend Request Batching parameter is set to 0, no batching will occur and a resend request will be sent for each single missing packet. For other settings, consider an example: Suppose the Resend Request Batching parameter is set to 80 referring to a span between the resend request threshold and the front end of the receive window that can hold five packets (Figure 28). In this case 4 packets (5 x 80%) will fit into the span between the resend request threshold and the resend request batching threshold. Accordingly, the maximum number of consecutive missing packets that can be batched is 5 (4 + 1).

Timeout Resend Mechanism Parameters

The timeout resend mechanism is illustrated in Figure 29 where the following assumptions are made:

- The frame includes 3000 packets.
- Packet 1002 is missing within the stream of packets and has not been recovered.
- Packets 2999 and 3000 are missing at the end of the stream of packets (end of the frame).
- The Maximum Number Resend Requests parameter is set to 3.

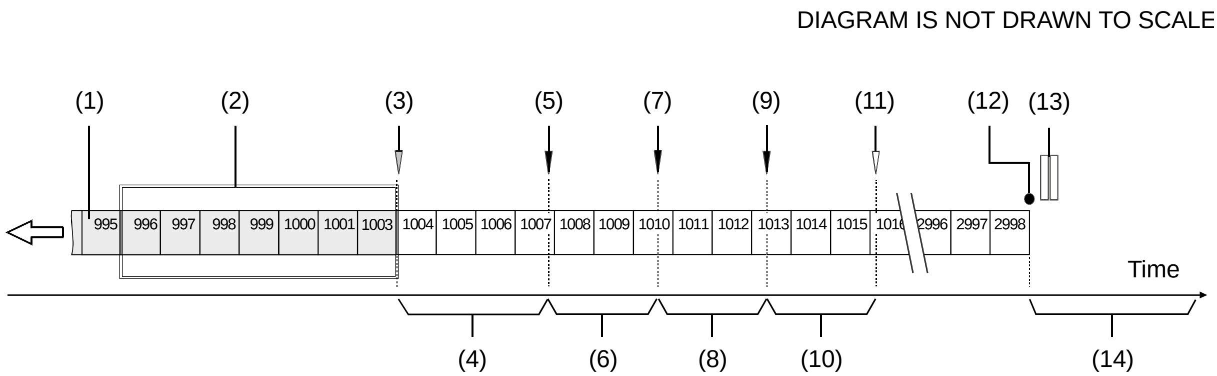


Fig. 29: Incomplete Stream of Packets and Part of the Resend Mechanism

(1) Stream of packets. Gray indicates that the status was checked as the packet entered the receive window. White indicates that the status has not yet been checked.

(2) Receive window of the performance driver.

(3) As packet 1003 enters the receive window, packet 1002 is detected as missing.

(4) Interval defined by the Resend Timeout parameter.

(5) The Resend Timeout interval expires and the first resend request for packet 1002 is sent to the camera. The camera does not respond with a resend.

(6) Interval defined by the Resend Response Timeout parameter.

(7) The Resend Response Timeout interval expires and a second resend request for packet 1002 is sent to the camera. The camera does not respond with a resend.

(8) Interval defined by the Resend Response Timeout parameter.

(9) The Resend Response Timeout interval expires and a third resend request for packet 1002 is sent to the camera. The camera still does not respond with a resend.

(10) Interval defined by the Resend Response Timeout parameter.

(11) Because the maximum number of resend requests has been sent and the last Resend Response Timeout interval has expired, packet 1002 is now considered as lost.

(12) End of the frame.

(13) Missing packets at the end of the frame (2999 and 3000).

(14) Interval defined by the Packet Timeout parameter.

Maximum Number Resend Requests - The Maximum Number Resend Requests parameter sets the maximum number of resend requests the performance driver will send to the camera for each missing packet.

Resend Timeout - The Resend Timeout parameter defines how long (in milliseconds) the performance driver will wait after detecting that a packet is missing before sending a resend request to the camera. The parameter applies only once to each missing packet after the packet was detected as missing.

Resend Request Response Timeout - The Resend Request Response Timeout parameter defines how long (in milliseconds) the performance driver will wait after sending a resend request to the camera before considering the resend request as lost.

If a resend request for a missing packet is considered lost and if the maximum number of resend requests as set by the Maximum Number Resend Requests parameter has not yet been reached, another resend request will be sent. In this case, the parameter defines the time separation between consecutive resend requests for a missing packet.

Packet Timeout - The Packet Timeout parameter defines how long (in milliseconds) the performance driver will wait for the next expected packet before it sends a resend request to the camera. This parameter ensures that resend requests are sent for missing packets near to the end of a frame. In the event of a major interruption in the stream of packets, the parameter will also ensure that resend requests are sent for missing packets that were detected to be missing immediately before the interruption.

Threshold and Timeout Resend Mechanisms Combined

Figure 30 illustrates the combined action of the threshold and the timeout resend mechanisms where the following assumptions are made:

- All parameters set to default.
- The frame includes 3000 packets.
- Packet 1002 is missing within the stream of packets and has not been recovered.
- Packets 2999 and 3000 are missing at the end of the stream of packets (end of the frame).

The default values for the performance driver parameters will cause the threshold resend mechanism to become operative before the timeout resend mechanism. This ensures maximum efficiency and that resend requests will be sent for all missing packets.

With the default parameter values, the resend request threshold is located very close to the front end of the receive window. Accordingly, there will be only a minimum delay between detecting a missing packet and sending a resend request for it. In this case, a delay according to the Resend Timeout parameter will not occur (see Figure 30). In addition, resend request batching will not occur.

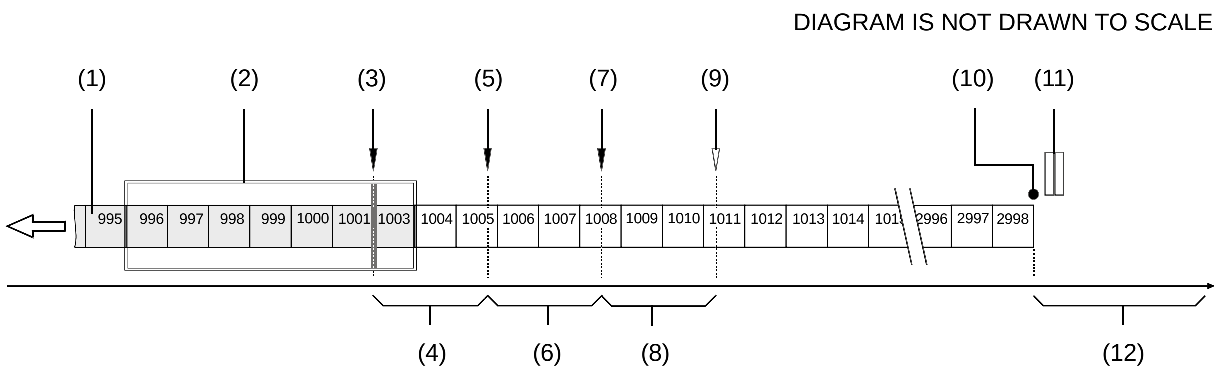


Fig. 30: Combination of Threshold Resend Mechanism and Timeout Resend Mechanism

(1) Stream of packets, Gray indicates that the status was checked as the packet entered the receive window. White indicates that the status has not yet been checked.

(2) Receive window of the performance driver.

(3) Threshold for sending resend requests (resend request threshold). The first resend request for packet 1002 is sent to the camera. The camera does not respond with a resend.

(4) Interval defined by the Resend Response Timeout parameter.

(5) The Resend Timeout interval expires and the second resend request for packet 1002 is sent to the camera. The camera does not respond with a resend.

(6) Interval defined by the Resend Response Timeout parameter

(7) The Resend Timeout interval expires and the third resend request for packet 1002 is sent to the camera. The camera does not respond with a resend.

(8) Interval defined by the Resend Response Timeout parameter

(9) Because the maximum number of resend requests has been sent and the last Resend Response Timeout interval has expired, packet 1002 is now considered as lost.

(10) End of the frame.

(11) Missing packets at the end of the frame (2999 and 3000).

(12) Interval defined by the Packet Timeout parameter.

You can set the performance driver parameter values from within your application software by using the pylon API. The following code snippet illustrates using the API to read and write the parameter values:

```
// Get the Stream Parameters object
Pylon::IStreamGrabber* pStreamGrabber = Camera.GetStreamGrabber(0);
Camera_t::StreamParams_t &StreamParams =
   dynamic_cast<Camera_t::StreamParams_t&>( *pStreamGrabber->GetNodeMap() );

// Write the ReceiveWindowSize parameter
StreamParams.ReceiveWindowSize.SetValue( 16 );

// Disable packet resends
StreamParams.EnableResend.SetValue( false );

// Write the PacketTimeout parameter
StreamParams.PacketTimeout.SetValue( 40 );

// Write the ResendRequestThreshold parameter
StreamParams.ResendRequestThreshold.SetValue( 5 );

// Write the ResendRequestBatching parameter
StreamParams.ResendRequestBatching.SetValue( 10 );

// Write the ResendTimeout parameter
StreamParams.ResendTimeout.SetValue( 2 );

// Write the ResendRequestResponseTimeout parameter
StreamParams.ResendRequestResponseTimeout.SetValue( 2 );

// Write the MaximumNumberResendRequests parameter
StreamParams.MaximumNumberResendRequests.SetValue( 25 );
```

For detailed information about using the pylon API, refer to the Basler pylon Programmer’s Guide and API Reference.

You can also use the Basler pylon Viewer application to easily set the parameters. (Note that the performance driver parameters will only appear in the viewer if the performance driver is installed on the adapter to which your camera is connected.)

For more information about the pylon Viewer, see Section 2.2 on page 32and Section 2.4 on page 52.

Adapter Properties

When the Basler Performance driver is installed, it adds a set of "advanced" properties to the network adapter. These properties include:

Max Packet Latency - A value in microseconds that defines how long the adapter will wait after it receives a packet before it generates a packet received interrupt.

Max Receive Inter-packet Delay - A value in microseconds that defines the maximum amount of time allowed between incoming packets.

Maximum Interrupts per Second - Sets the maximum number of interrupts per second that the adapter will generate.

Network Address - allows the user to specify a MAC address that will override the default address provided by the adapter.

Packet Buffer Size - Sets the size in bytes of the buffers used by the receive descriptors and the transmit descriptors.

Receive Descriptors - Sets the number of descriptors to use in the adapter's receiving ring.

Transmit Descriptors - Sets the number of descriptors to use in the adapter's transmit ring.

To access the advanced properties for an adapter:

5. Open a **Network Connections** window and find the connection for your network adapter.
6. Right click on the name of the connection and select **Properties** from the drop down menu.
7. A **LAN Connection Properties** window will open. Click the **Configure** button.
8. An **Adapter Properties** window will open. Click the **Advanced** tab.

Note

We strongly recommend using the default parameter settings. Changing the parameters can have a significant negative effect on the performance of the adapter and the driver.

6.3 Transport Layer Parameters

The transport layer parameters are part of the camera's basic GigE implementation. These parameters do not normally require adjustment.

Read Timeout - If a register read request is sent to the camera via the transport layer, this parameter designates the time out (in milliseconds) within which a response must be received.

Write Timeout - If a register write request is sent to the camera via the transport layer, this parameter designates the time out (in milliseconds) within which an acknowledge must be received.

Heartbeat Timeout - The GigE Vision standard requires implementation of a heartbeat routine to monitor the connection between the camera and the host PC. This parameter sets the heartbeat timeout (in milliseconds). If a timeout occurs, the camera releases the network connection and enters a state that allows reconnection.

Note

Management of the heartbeat time is normally handled by the Basler's basic GigE implementation and changing this parameter is not required for normal camera operation. However, if you are debugging an application and you stop at a break point, you will have a problem with the heartbeat timer. The timer will time out when you stop at a break point and the connection to the camera will be lost. When debugging, you should increase the heartbeat timeout to a high value to avoid heartbeat timeouts at break points. When debugging is complete, you should return the timeout to its normal setting.

You can set the driver related transport layer parameter values from within your application software by using the pylon API. The following code snippet illustrates using the API to read and write the parameter values:

```
// Read/Write Timeout
Camera_t::TlParams_t &TlParams =
    dynamic_cast<Camera_t::TlParams_t&>(*Camera.GetTLNodeMap());
TlParams.ReadTimeout.SetValue(500);  // 500 milliseconds
TlParams.WriteTimeout.SetValue(500); // 500 milliseconds

// Heartbeat Timeout
Camera_t::TlParams_t &TlParams =
    dynamic_cast<Camera_t::TlParams_t&>( *Camera.GetTLNodeMap() );
TlParams.HeartbeatTimeout.SetValue(5000);  // 5 seconds
```

You can also use the Basler pylon Viewer application to easily set the parameters.

7 Network Related Camera Parameters and Managing Bandwidth

This section describes the camera parameters that are related to the camera's performance on the network. It also describes how to use the parameters to manage the available network bandwidth when you are using multiple cameras.

7.1 Network Related Parameters in the Camera

The camera includes several parameters that determine how it will use its network connection to transmit data to the host PC. The list below describes each parameter and provides basic information about how the parameter is used. The following section describes how you can use the parameters to manage the bandwidth used by each camera on your network.

Payload Size (read only)

Indicates the total size in bytes of the image data plus any chunk data (if chunks are enabled) that the camera will transmit. Packet headers are not included.

Stream Channel Selector (read/write)

The GigE Vision standard specifies a mechanism for establishing several separate stream channels between the camera and the PC. This parameter selects the stream channel that will be affected when the other network related parameters are changed.

Currently, the cameras support only one stream channel, i.e., stream channel 0.

Packet Size (read/write)

As specified in the GigE Vision standard, each acquired image will be fit into a data block. The block contains three elements: a *data leader* consisting of one packet used to signal the beginning of a data block, the *data payload* consisting of one or more packets containing the actual data for the current block, and a *data trailer* consisting of one packet used to signal the end of the data block.

The packet size parameter sets the size of the packets that the camera will use when it sends the data payload via the selected stream channel. The value is in bytes. The value does not affect the leader and trailer size and the last data packet may be a smaller size.

The packet size parameter should always be set to the maximum size that your network adapter and network switches (if used) can handle.

Inter-packet Delay (read/write)

Sets the delay in ticks between the packets sent by the camera. Applies to the selected stream channel. Increasing the inter-packet delay will decrease the camera's effective data transmission rate and will thus decrease the network bandwidth used by the camera.

In the current camera implementation, one tick = 8 ns. To check the tick frequency, you can read the Gev Timestamp Tick Frequency parameter value. This value indicates the number of clock ticks per second.

Frame Transmission Delay (read/write)

Sets a delay in ticks (one tick = 8 ns) between when a camera would normally begin transmitting an acquired frame and when it actually begins transmission. This parameter should be set to zero in most normal situations.

If you have many cameras in your network and you will be simultaneously triggering image acquisition on all of them, you may find that your network switch or network adapter is overwhelmed if all of the cameras simultaneously begin to transmit image data at once. The frame transmission delay parameter can be used to stagger the start of image data transmission from each camera.

Bandwidth Assigned (read only)

Indicates the bandwidth in bytes per second that will be used by the camera to transmit image and chunk feature data and to handle resends and control data transmissions. The value of this parameter is a result of the packet size and the inter-packet delay parameter settings.

In essence, the bandwidth assigned is calculated this way:

$$\text{Bandwidth Assigned} = \frac{\dfrac{\text{X Packets}}{\text{Frame}} \times \dfrac{\text{Y Bytes}}{\text{Packet}}}{\left[\dfrac{\text{X Packets}}{\text{Frame}} \times \dfrac{\text{Y Bytes}}{\text{Packet}} \times \dfrac{8\text{ ns}}{\text{Byte}}\right] + \left[\left(\dfrac{\text{X Packets}}{\text{Frame}} - 1\right) \times (\text{IPD} \times 8\text{ ns})\right]}$$

Where: X = number of packets needed to transmit the frame

Y = number of bytes in each packet

IPD = Inter-packet Delay setting in ticks (with a tick set to the 8 ns standard)

When considering this formula, you should know that on a Gigabit network it takes one tick to transmit one byte. Also, be aware that the formula has been simplified for easier understanding.

Bandwidth Reserve (read/write)

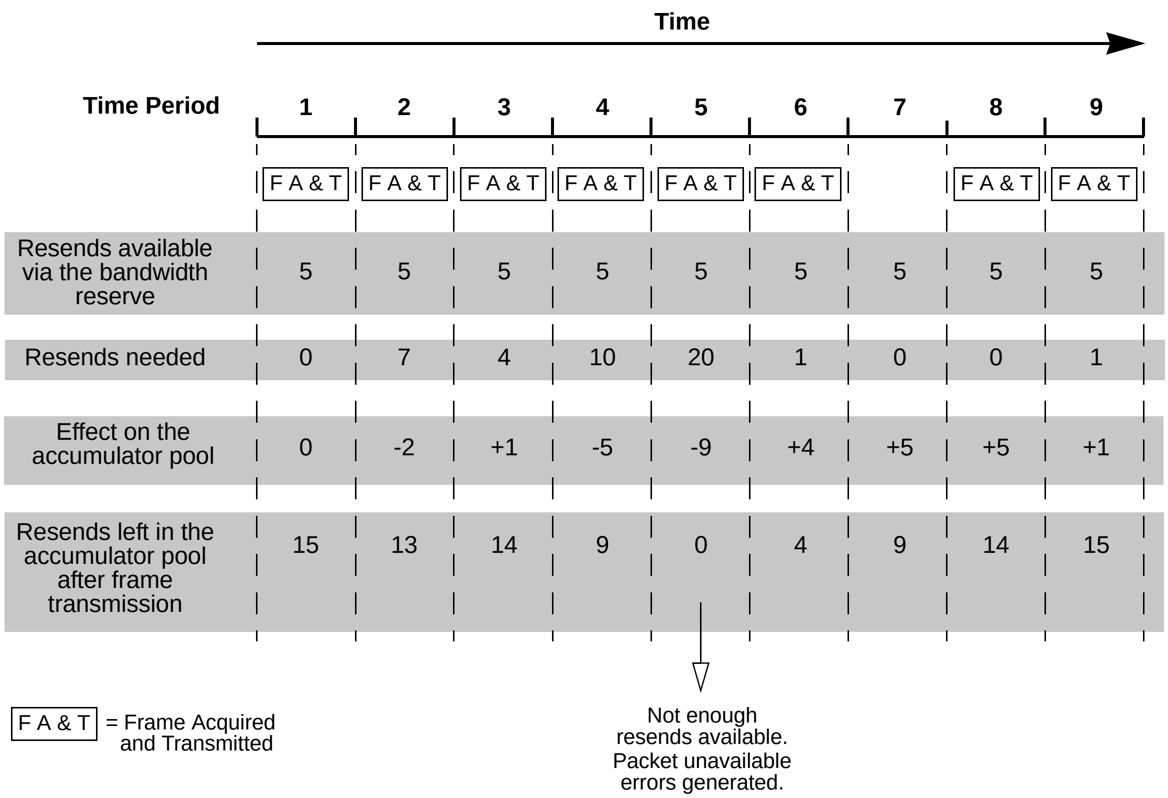
Used to reserve a portion of the assigned bandwidth for packet resends and for the transmission of control data between the camera and the host PC. The setting is expressed as a percentage of the Bandwidth Assigned parameter. For example, if the Bandwidth Assigned parameter indicates that 30 MByte/s have been assigned to the camera and the Bandwidth Reserve parameter is set to 5%, then the bandwidth reserve will be 1.5 MByte/s.

Bandwidth Reserve Accumulation (read/write)

A software device called the bandwidth reserve accumulator is designed to handle unusual situations such as a sudden EMI burst that interrupts an image transmission. If this happens, a larger than normal number of packet resends may be needed to properly transmit a complete image. The accumulator is basically an extra pool of resends that the camera can use in unusual situations.

The Bandwidth Reserve Accumulation parameter is a multiplier used to set the maximum number of resends that can be held in the "accumulator pool." For example, assume that the current bandwidth reserve setting for your camera is 5% and that this reserve is large enough to allow up to 5 packet resends during a frame period. Also assume that the Bandwidth Reserve Accumulation parameter is set to 3. With these settings, the accumulator pool can hold a maximum of 15 resends (i.e., the multiplier times the maximum number of resends that could be transmitted in a frame period). Note that with these settings, 15 will also be the starting number of resends within the accumulator pool.

The chart on the next page and the numbered text below it show an example of how the accumulator would work with these settings. The chart and the text assume that you are using an external trigger to trigger image acquisition. The example also assumes that the camera is operating in a poor environment, so many packets are lost and many resends are required. The numbered text is keyed to the time periods in the chart.



(1) You trigger image acquisition and during this time period, the camera acquires and transmits a frame. The bandwidth reserve setting would allow 5 resends during this time period, but no resends are needed. The accumulator pool started with 15 resends available and remains at 15.

(2) You trigger image acquisition and during this time period, the camera acquires and transmits a frame. The bandwidth reserve setting would allow 5 resends during this time period, but 7 resends are needed. The 5 resends available via the bandwidth reserve are used and 2 resends are used from the accumulator pool. The accumulator pool is drawn down to 13.

(3) You trigger image acquisition and during this time period, the camera acquires and transmits a frame. The bandwidth reserve setting would allow 5 resends during this time period and 4 resends are needed. The 4 resends needed are taken from the resends available via the bandwidth reserve. The fifth resend available via the bandwidth reserve is not needed, so it is added to the accumulator pool and brings the pool to 14.

(4) You trigger image acquisition and during this time period, the camera acquires and transmits a frame. The bandwidth reserve setting would allow 5 resends during this time period, but 10 resends are needed. The 5 resends available via the bandwidth reserve are used and 5 resends are used from the accumulator pool. The accumulator pool is drawn down to 9.

(5) You trigger image acquisition and during this time period, the camera acquires and transmits a frame. The bandwidth reserve setting would allow 5 resends during this time period, but 20 resends are needed. The 5 resends available via the bandwidth reserve are used. To complete all of the needed resends, 15 resends would be required from the accumulator pool, but the pool only has 9 resends. So the 9 resends in the pool are used and 6 resend requests are answered with a "packet unavailable" error code. The accumulator pool is reduced to 0.

(6) You trigger image acquisition and during this time period, the camera acquires and transmits a frame. The bandwidth reserve setting would allow 5 resends during this time period and 1 resend is needed. The 1 resend needed is taken from the resends available via the bandwidth reserve. The other 4 resends available via the bandwidth reserve are not needed, so they are added to the accumulator pool and they bring the pool up to 4.

(7) During this time period, you do not trigger image acquisition. You delay triggering acquisition for the period of time that would normally be needed to acquire and transmit a single image. The current camera settings would allow 5 resends to occur during this period of time. But since no data is transmitted, no resends are required. The 5 resends that could have occurred are added to the accumulator pool and they bring the pool up to 9.

(8) You trigger image acquisition and during this time period, the camera acquires and transmits a frame. The bandwidth reserve setting would allow 5 resends during this time period, but no resends are needed. The 5 resends available via the bandwidth reserve are not needed, so they are added to the accumulator pool and they bring the pool up to 14.

(9) You trigger image acquisition and during this time period, the camera acquires and transmits a frame. The bandwidth reserve setting would allow 5 resends during this time period and 1 resend is needed. The 1 resend needed is taken from the resends available via the bandwidth reserve. The other 4 resends available via the bandwidth reserve are not needed, so they are added to the accumulator pool. Note that with the current settings, the accumulator pool can only hold a maximum of 15 resends. So the pool is now 15.

Frame Max Jitter (read only)

If the Bandwidth Reserve Accumulation parameter is set to a high value, the camera can experience a large burst of data resends during transmission of a frame. This burst of resends will delay the start of transmission of the next acquired frame. The Frame Max Jitter parameter indicates the maximum time in ticks (one tick = 8 ns) that the next frame transmission could be delayed due to a burst of resends.

Device Max Throughput (read only)

Indicates the maximum amount of data (in bytes per second) that the camera could generate given its current settings and an ideal world. This parameter gives no regard to whether the GigE network has the capacity to carry all of the data and does not consider any bandwidth required for resends. In essence, this parameter indicates the maximum amount of data the camera could generate with no network restrictions.

If the Acquisition Frame Rate abs parameter has been used to set the camera's frame rate, the camera will use this frame rate setting to calculate the device max throughput. If software or hardware triggering is being used to control the camera's frame rate, the maximum frame rate allowed with the current camera settings will be used to calculate the device max throughput.

Device Current Throughput (read only)

Indicates the actual bandwidth (in bytes per second) that the camera will use to transmit image data and chunk data given the current area of interest settings, chunk feature settings, and the pixel format setting.

If the Acquisition Frame Rate abs parameter has been used to set the camera's frame rate, the camera will use this frame rate setting to calculate the device current throughput. If software or hardware triggering is being used to control the camera's frame rate, the maximum frame rate allowed with the current camera settings will be used to calculate the device current throughput.

Note that the Device Current Throughput parameter indicates the bandwidth needed to transmit the actual image data and chunk data. The Bandwidth Assigned parameter, on the other hand, indicates the bandwidth needed to transmit image data and chunk data plus the bandwidth reserved for retrys and the bandwidth needed for any overhead such as leaders and trailers.

Resulting Frame Rate (read only)

Indicates the maximum allowed frame acquisition rate (in frames per second) given the current camera settings. The parameter takes the current area of interest, exposure time, and bandwidth settings into account.

If the Acquisition Frame Rate abs parameter has been used to set the camera's frame rate, the Resulting Frame Rate parameter will show the Acquisition Frame Rate abs parameter setting. If software or hardware triggering is being used to control the camera's frame rate, the Resulting Frame Rate parameter will indicate the maximum frame rate allowed given the current camera settings.

You can read or set the camera's network related parameter values from within your application software by using the pylon API. The following code snippet illustrates using the API to set the selector and the parameter values:

```
// Payload Size
int64_t payloadSize = Camera.PayloadSize.GetValue();

// GevStreamChannelSelector
Camera.GevStreamChannelSelector.SetValue
( GevStreamChannelSelector_StreamChannel0 );

// PacketSize
Camera.GevSCPSPacketSize.SetValue( 1500 );

// Inter-packet Delay
Camera.GevSCPD.SetValue( 1000 );

// Frame-transmission Delay
Camera.GevSCFTD.SetValue( 1000 );

// Bandwidth Reserve
Camera.GevSCBWR.SetValue( 10 );
```

```
// Bandwidth Reserve Accumulation
Camera.GevSCBWRA.SetValue( 10 );


// Frame Jitter Max
int64_t jitterMax = Camera.GevSCFJM.GetValue();

// Device Max Throughput
int64_t maxThroughput = Camera.GevSCDMT.GetValue();

// Device Current Throughput
int64_t currentThroughput = Camera.GevSCDCT.GetValue();

// Resulting Framerate
double resultingFps = Camera.ResultingFrameRateAbs.GetValue();
```

For detailed information about using the pylon API, refer to the Basler pylon Programmer’s Guide and API Reference.

7.2 Managing Bandwidth When Multiple Cameras Share a Single Network Path

If you are using a single camera on a GigE network, the problem of managing bandwidth is simple. The network can easily handle the bandwidth needs of a single camera and no intervention is required. A more complicated situation arises if you have multiple cameras connected to a single network adapter as shown in Figure 31.

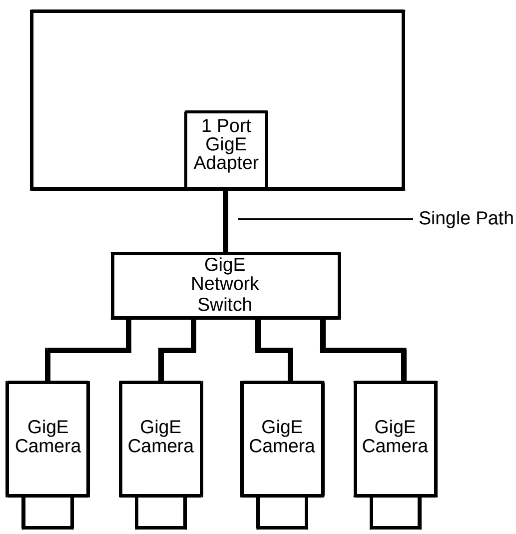


Fig. 31: Multiple Cameras on a Network

One way to manage the situation where multiple cameras are sharing a single network path is to make sure that only one of the cameras is acquiring and transmitting images at any given time. The data output from a single camera is well within the bandwidth capacity of the single path and you should have no problem with bandwidth in this case.

If you want to acquire and transmit images from several cameras simultaneously, however, you must determine the total data output rate for all the cameras that will be operating simultaneously and you must make sure that this total does not exceed the bandwidth of the single path (125 MByte/s).

An easy way to make a quick check of the total data output from the cameras that will operate simultaneously is to read the value of the Bandwidth Assigned parameter for each camera. This parameter indicates the camera’s gross data output rate in bytes per second with its current settings. If the sum of the bandwidth assigned values is less than 125 MByte/s, the cameras should be able to operate simultaneously without problems. If it is greater, you must lower the data output rate of one or more of the cameras.

You can lower the data output rate on a camera by using the Inter-packet Delay parameter. This parameter adds a delay between the transmission of each packet from the camera and thus slows the data transmission rate of the camera. The higher the inter-packet delay parameter is set, the greater the delay between the transmission of each packet will be and the lower the data transmission rate will be. After you have adjusted the Inter-packet Delay parameter on each camera, you can check the sum of the Bandwidth Assigned parameter values and see if the sum is now less than 125 MByte/s.

7.2.1 A Procedure for Managing Bandwidth

In theory, managing bandwidth sharing among several cameras is as easy as adjusting the inter-packet delay. In practice, it is a bit more complicated because you must consider several factors when managing bandwidth. The procedure below outlines a structured approach to managing bandwidth for several cameras.

The objectives of the procedure are:

- To optimize network performance.
- To determine the bandwidth needed by each camera for image data transmission.
- To determine the bandwidth actually assigned to each camera for image data transmission.
- For each camera, to make sure that the actual bandwidth assigned for image data transmission matches the bandwidth needed.
- To make sure that the total bandwidth assigned to all cameras does not exceed the network’s bandwidth capacity.
- To make adjustments if the bandwidth capacity is exceeded.

Step 1 - Set the Packet Size parameter on each camera as large as possible.

Using the largest possible packet size has two advantages, it increases the efficiency of network transmissions between the camera and the PC and it reduces the time required by the PC to process incoming packets. The largest packet size setting that you can use with your camera is determined by the largest packet size that can be handled by your network. The size of the packets that can be handled by the network depends on the capabilities and settings of the network adapter you are using and on capabilities of the network switch you are using.

Start by checking the documentation for your adapter to determine the maximum packet size (sometimes called “frame” size) that the adapter can handle. Many adapters can handle what is known as “jumbo packets” or "jumbo frames". These are packets with a 16 kB size. Once you have determined the maximum size packets the adapter can handle, make sure that the adapter is set to use the maximum packet size.

Next, check the documentation for your network switch and determine the maximum packet size that it can handle. If there are any settings available for the switch, make sure that the switch is set for the largest packet size possible.

Now that you have set the adapter and switch, you can determine the largest packet size the network can handle. The device with the smallest maximum packet size determines the maximum allowed packet size for the network. For example, if the adapter can handle 16 kB packets and the switch can handle 8 kB packets, then the maximum for the network is 8 kB packets.

Once you have determined the maximum packet size for your network, set the value of the Packet Size parameter on each camera to this value.

Tip

The manufacturer's documentation sometimes makes it difficult to determine the maximum packet size for a device, especially network switches. There is a "quick and dirty" way to check the maximum packet size for your network with its current configuration:

1. Open the pylon Viewer, select a camera, and set the Packet Size parameter to a low value (1 kB for example).
2. Use the Continuous Shot mode to capture several images.
3. Gradually increase the value of the Packet Size parameter and capture a few images after each size change.
4. When your Packet Size setting exceeds the packet size that the network can handle, the viewer will lose the ability to capture images. (When you use Continuous Shot, the viewer's status bar will indicate that it is acquiring images, but the image in the viewing area will appear to be frozen.)

Step 2 - Set the Bandwidth Reserve parameter for each camera.

The Bandwidth Reserve parameter setting for a camera determines how much of the bandwidth assigned to that camera will be reserved for lost packet resends and for asynchronous traffic such as commands sent to the camera. If you are operating the camera in a relatively EMI free environment, you may find that a bandwidth reserve of 2% or 3% is adequate. If you are operating in an extremely noisy environment, you may find that a reserve of 8% or 10% is more appropriate.

Step 3 - Calculate the "data bandwidth needed" by each camera.

The objective of this step is to determine how much bandwidth (in Byte/s) each camera needs to transmit the image data that it generates. The amount of data bandwidth a camera needs is the product of several factors: the amount of data included in each image, the amount of chunk data being added to each image, the "packet overhead" such as packet leaders and trailers, and the number of frames the camera is acquiring each second.

For each camera, you can use the two formulas below to calculate the data bandwidth needed. To use the formulas, you will need to know the current value of the Payload Size parameter and the Packet Size parameter for each camera. You will also need to know the frame rate (in frames/s) at which each camera will operate.

$$\text{Bytes/Frame} = \left[\left\lceil \frac{\text{Payload Size}}{\text{Packet Size}} \right\rceil^{1} \times \text{Packet Overhead} \right] + \lceil \text{Payload Size} \rceil^{4} + \text{Leader Size} + \text{Trailer Size}$$

$$\text{Data Bandwidth Needed} = \text{Bytes/Frame} \times \text{Frames/s}$$

Where:

Packet Overhead = 72 (for a GigE network)

78 (for a 100 MBit/s network)

Leader Size = Packet Overhead + 36 (if chunk mode is not active)

Packet Overhead + 12 (if chunk mode is active)

Trailer Size = Packet Overhead + 8

$\lceil x \rceil^1$ means round up x to the nearest integer

$\lceil x \rceil^4$ means round up x to the nearest multiple of 4

Step 4 - Calculate “data bandwidth assigned” to each camera.

For each camera, there is a parameter called Bandwidth Assigned. This read only parameter indicates the total bandwidth that has been assigned to the camera. The Bandwidth Assigned parameter includes both the bandwidth that can be used for image data transmission plus the bandwidth that is reserved for packet resents and camera control signals. To determine the “data bandwidth assigned,” you must subtract out the reserve.

You can use the formula below to determine the actual amount of assigned bandwidth that is available for data transmission. To use the formula, you will need to know the current value of the Bandwidth Assigned parameter and the Bandwidth reserve parameter for each camera.

$$\text{Data Bandwidth Assigned} = \text{Bandwidth Assigned} \times \frac{100 - \text{Bandwidth Reserved}}{100}$$

Step 5 - For each camera, compare the data bandwidth needed with the data bandwidth assigned.

For each camera, you should now compare the data bandwidth assigned to the camera (as determined in step 4) with the bandwidth needed by the camera (as determined in step 3).

For bandwidth to be used most efficiently, the data bandwidth assigned to a camera should be equal to or just slightly greater than the data bandwidth needed by the camera. If you find that this is the situation for all of the cameras on the network, you can go on to step 6 now. If you find a camera that has much more data bandwidth assigned than it needs, you should make an adjustment.

To lower the amount of data bandwidth assigned, you must adjust a parameter called the Inter-packet Delay. If you increase the Inter-packet Delay parameter value on a camera, the data bandwidth assigned to the camera will decrease. So for any camera where you find that the data bandwidth assigned is much greater then the data bandwidth needed, you should do this:

1. Raise the setting for the Inter-packet delay parameter for the camera.
2. Recalculate the data bandwidth assigned to the camera.
3. Compare the new data bandwidth assigned to the data bandwidth needed.
4. Repeat 1, 2, and 3 until the data bandwidth assigned is equal to or just greater than the data bandwidth needed.

Note

If you increase the inter-packet delay to lower a camera's data output rate there is something that you must keep in mind. When you lower the data output rate, you increase the amount of time that the camera needs to transmit an acquired frame (image). Increasing the frame transmission time can restrict the camera's maximum allowed acquisition frame rate.

Step 6 - Check that the total bandwidth assigned is less than the network capacity.

1. For each camera, determine the current value of the Bandwidth Assigned parameter. The value is in Byte/s. (Make sure that you determine the value of the Bandwidth Assigned parameter after you have made any adjustments described in the earlier steps.)
2. Find the sum of the current Bandwidth Assigned parameter values for all of the cameras.

If the sum of the Bandwidth Assigned values is less than 125 MByte/s for a Give network or 12.5 M/Byte/s for a 100 Bit/s network, the bandwidth management is OK.

If the sum of the Bandwidth Assigned values is greater than 125 MByte/s for a Give network or 12.5 M/Byte/s for a 100 Bit/s network, the cameras need more bandwidth than is available and you must make adjustments. In essence, you must lower the data bandwidth needed by one or more of the cameras and then adjust the data bandwidths assigned so that they reflect the lower bandwidth needs.

You can lower the data bandwidth needed by a camera either by lowering its frame rate or by decreasing the size of the area of interest (AOI). Once you have adjusted the frame rates and/or AOI settings on the cameras, you should repeat steps 2 through 6.

For more information about the camera's maximum allowed frame transmission rate, see Section 10.10 on page 152.

For more information about the AOI, see Section 13.5 on page 209.

8 Camera Functional Description

This section provides an overview of the camera's functionality from a system perspective. The overview will aid your understanding when you read the more detailed information included in the next sections of the user's manual.

8.1 Overview (all models except scA750-60)

Note

The information in this section applies to all camera models except the scA750-60 gm/gc. For information about scA750-60 cameras, see Section 8.2 on page 111.

Each camera provides features such as a full frame shutter and electronic exposure time control.

Exposure start, exposure time, and charge readout can be controlled by parameters transmitted to the camera via the Basler pylon API and the GigE interface. There are also parameters available to set the camera for single frame acquisition or continuous frame acquisition.

Exposure start can also be controlled via an externally generated hardware trigger (ExTrig) signal. The ExTrig signal facilitates periodic or non-periodic acquisition start. Modes are available that allow the length of exposure time to be directly controlled by the ExTrig signal or to be set for a pre-programmed period of time.

Accumulated charges are read out of the sensor when exposure ends. At readout, accumulated charges are transported from the sensor's light-sensitive elements (pixels) to the vertical shift registers (see Figure 32 on page 110). The charges from the bottom line of pixels in the array are then moved into a horizontal shift register. Next, the charges are shifted out of the horizontal register. As the charges move out of the horizontal shift register, they are converted to voltages proportional to the size of each charge. Each voltage is then amplified by a Variable Gain Control (VGC) and digitized by an Analog-to-Digital converter (ADC). After each voltage has been amplified and digitized, it passes through an FPGA and into an image buffer. All shifting is clocked according to the camera's internal data rate. Shifting continues in a linewise fashion until all image data has been read out of the sensor.

The pixel data leaves the image buffer and passes back through the FPGA to an Ethernet controller where it is assembled into data packets. The packets are then transmitted via an Ethernet network to a network adapter in the host PC. The Ethernet controller also handles transmission and receipt of control data such as changes to the camera's parameters.

The image buffer between the sensor and the Ethernet controller allows data to be read out of the sensor at a rate that is independent of the data transmission rate between the camera and the host computer. This ensures that the data transmission rate has no influence on image quality.

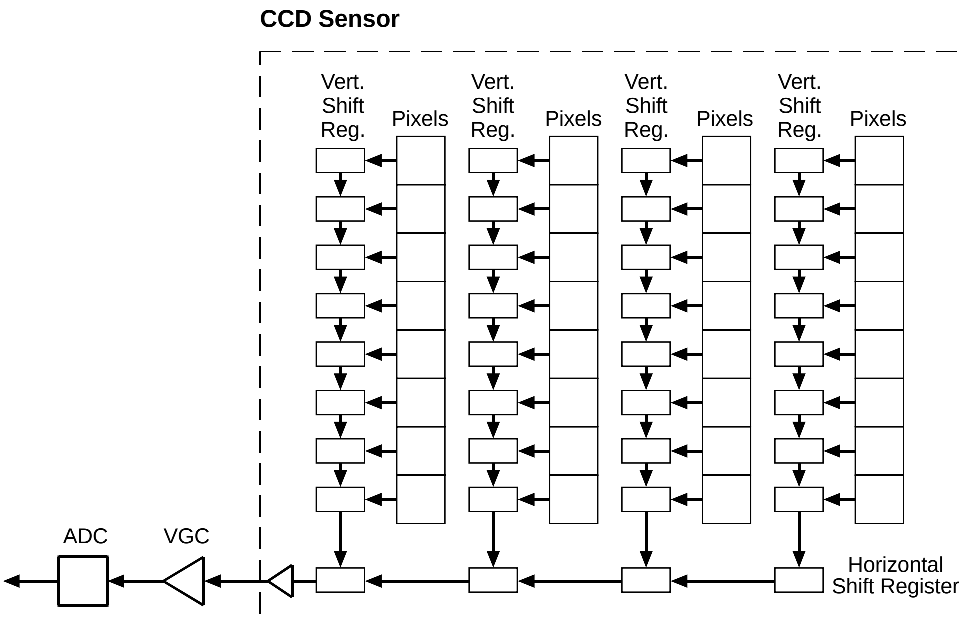


Fig. 32: CCD Sensor Architecture

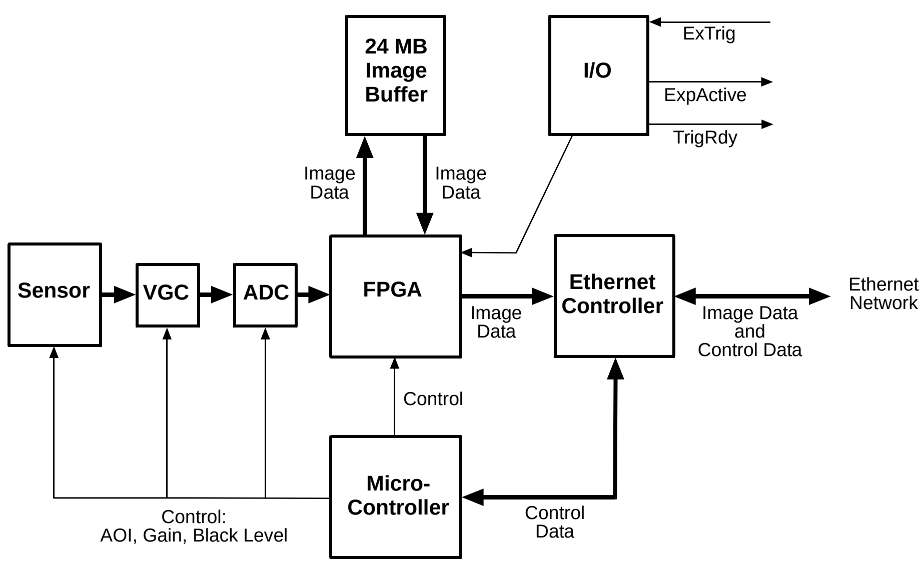


Fig. 33: Camera Block Diagram

8.2 Overview (scA750-60 only)

Note

The information in this section only applies to scA750-60 gm/gc cameras. For information about the other camera models, see Section 8.1 on page 109.

Each camera provides features such as a full frame shutter and electronic exposure time control. The sensor chip includes gain controls, ADCs, and other digital devices.

Exposure start, exposure time, and charge readout can be controlled by parameters transmitted to the camera via the Basler pylon API and the GigE interface. There are also parameters available to set the camera for single frame acquisition or continuous frame acquisition.

Exposure start can also be controlled via an externally generated hardware trigger (ExTrig) signal. The ExTrig signal facilitates periodic or non-periodic acquisition start. Exposure can be set for a preprogrammed period of time.

Accumulated charges are read out when the programmed exposure time ends. At readout, the accumulated charges are transported from the sensor's light-sensitive elements (pixels) to the sensor's column buses (see Figure 32 on page 110). The charges from the bottom line of pixels in the array are then moved into the analog processing section of the sensor. As the charges move from the pixels to the analog processing section, they are converted to voltages proportional to the size of each charge. The voltages from the analog processing section are next passed to a bank of Analog-to-Digital converters (ADCs).

Finally, the voltages pass through a section of the sensor where they receive additional digital processing and then they are moved out of the sensor. As each voltage leaves the sensor, it passes through an FPGA and into an image buffer. All shifting is clocked according to the camera's internal data rate. Shifting continues in a linewise fashion until all image data has been read out of the sensor.

The pixel data leaves the image buffer and passes back through the FPGA to an Ethernet controller where it is assembled into data packets. The packets are then transmitted via an Ethernet network to a network adapter in the host PC. The Ethernet controller also handles transmission and receipt of control data such as changes to the camera's parameters.

The image buffer between the sensor and the Ethernet controller allows data to be read out of the sensor at a rate that is independent of the data transmission rate between the camera and the host computer. This ensures that the data transmission rate has no influence on image quality.

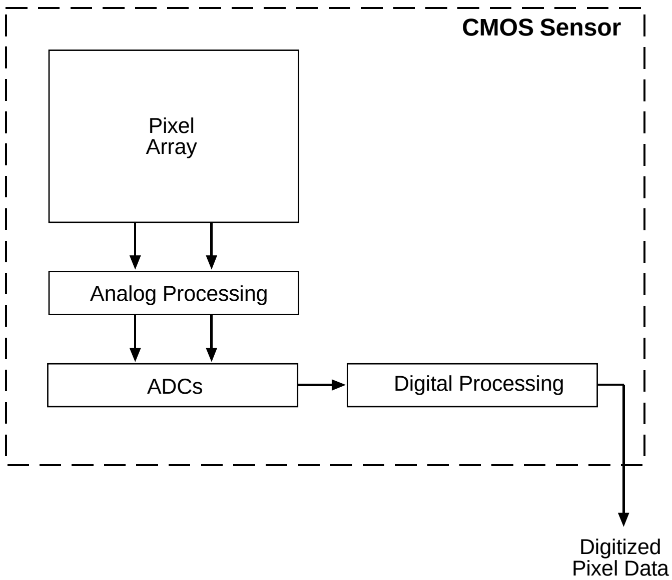


Fig. 34: CMOS Sensor Architecture

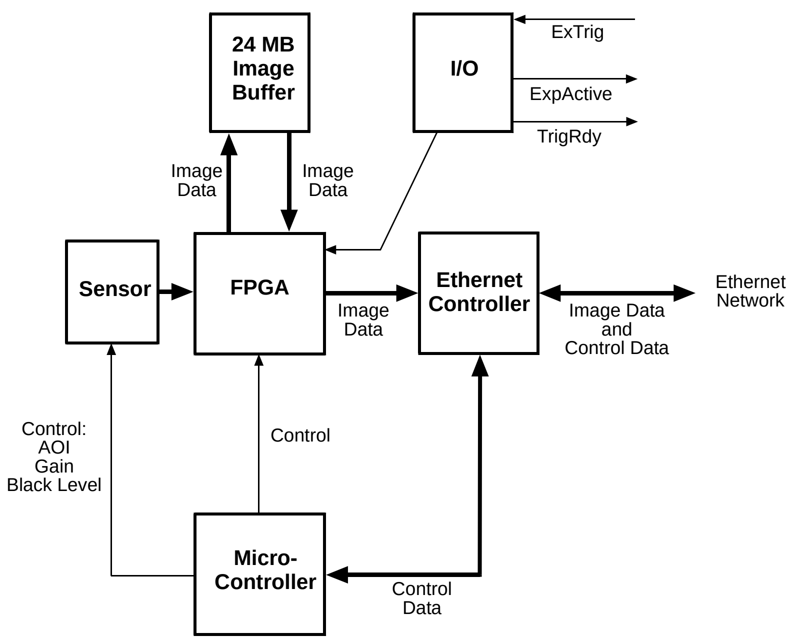


Fig. 35: Camera Block Diagram

9 Physical Interface

This section provides detailed information, such as pinouts and voltage requirements, for the physical interface on the camera. This information will be especially useful during your initial design-in process.

9.1 General Description of the Connections

The camera is interfaced to external circuity via connectors located on the back of the housing:

- An 8-pin, RJ-45 jack used to provide a 100/1000 Mbit/s Ethernet connection to the camera. This jack includes a green LED and a yellow LED that indicate the state of the network connection.
- A 12-pin receptacle used to provide access to the camera's I/O lines and to provide input power to the camera.

The drawing below shows the location of the two connectors and the LEDs.

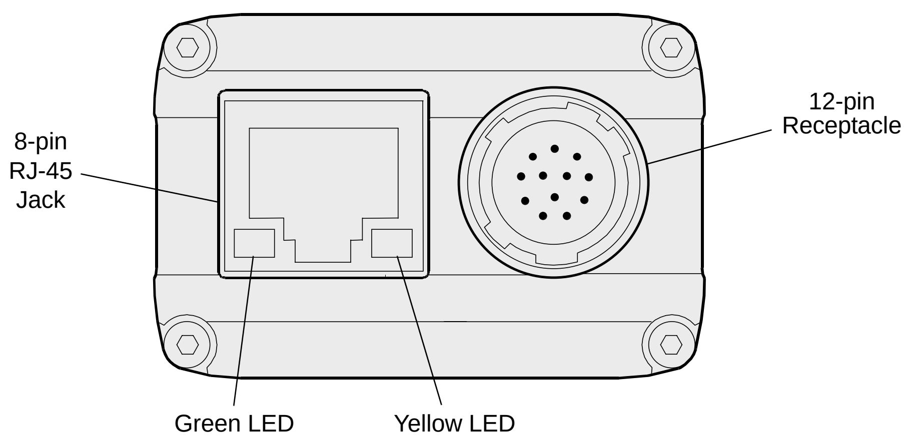


Fig. 36: Camera Connectors and LED

9.2 Connector Pin Assignments and Numbering

9.2.1 12-Pin Receptacle Pin Assignments

The 12 pin receptacle is used to access the two physical input lines and four physical output lines on the camera. It is also used to supply power to the camera. The pin assignments for the receptacle are shown in Table 5.

Pin	Designation
1	Camera Input Power Gnd *
2	Camera Input Power Gnd *
3	I/O Input 1 (+5 to +24 VDC)
4	I/O Input 2 (+5 to +24 VDC)
5	I/O Input Gnd
6	I/O Output 1
7	I/O Output 2
8	Camera Input Power VCC ** (+12 to +24 VDC)
9	Camera Input Power VCC ** (+12 to +24 VDC)
10	I/O Output VCC (+5 to +24 VDC)
11	I/O Output 3
12	I/O Output 4

Table 5: Pin Assignments for the 12-pin Receptacle

Note

* Pins 1 and 2 are tied together inside of the camera.

** Pins 8 and 9 are tied together inside of the camera.

To avoid a voltage drop with long input power wires, we recommend that you supply input power VCC through separate wires between your power supply and pins 8 and 9 on the camera. We also recommend that you supply input power ground through separate wires between your power supply and pins 1 and 2 on the camera.

9.2.2 RJ-45 Jack Pin Assignments

The 8-pin, RJ-45 jack provides Ethernet access to the camera. Pin numbering and assignment adhere to the Ethernet standard.

9.2.3 Pin Numbering

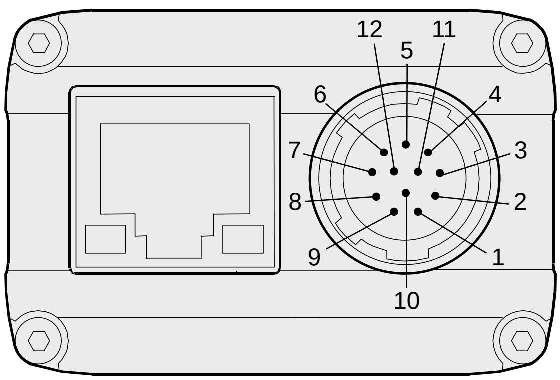


Fig. 37: Pin Numbering for the 12-pin Receptacle

9.3 Connector Types

9.3.1 8-pin RJ-45 Jack

The 8-pin jack for the camera’s Ethernet connection is a standard RJ-45 connector.

The recommended mating connector is any standard, 8-pin RJ-45 plug.

Green and Yellow LEDs

This RJ-45 jack on the camera includes a green LED and a yellow LED. When the green LED is lit, it indicates that an active network connection is available. When the yellow LED is lit, it indicates that data is being transmitted via the network connection.

9.3.2 12-Pin Connector

The 12-pin connector on the camera is a Hirose micro receptacle (part number HR10A-10R-12P) or the equivalent.

The recommended mating connector is the Hirose micro plug (part number HR10A-10P-12S) or the equivalent.

9.4 Cabling Requirements

9.4.1 Ethernet Cables

Use high-quality Ethernet cables. To avoid EMI, the cables must be shielded. Use of category 6 or category 7 cables with S/STP shielding is recommended. As a general rule, applications with longer cables or applications in harsh EMI conditions require higher category cables.

Either a straight-through (patch) or a cross-over Ethernet cable can be used to connect the camera directly to the PC or to a network switch.

9.4.2 Power and I/O Cable

A single cable is used to connect power to the camera and to connect to the camera's I/O lines as shown in Figure 38.

The end of the power and I/O cable that connects to the camera must be terminated with a Hirose micro plug (part number HR10A-10P-12S) or the equivalent. The cable must be wired to conform with the pin assignments shown in the pin assignment tables.

The maximum length of the power and I/O cable is at least 10 meters. The cable must be shielded and must be constructed with twisted pair wire. Use of twisted pair wire is essential to ensure that input signals are correctly received.

Close proximity to strong magnetic fields should be avoided.

The required 12-pin Hirose plug is available from Basler. Basler also offers a cable assembly that is terminated with a 12-pin Hirose plug on one end and unterminated on the other. Contact your Basler sales representative to order connectors or cables.

CAUTION

An Incorrect Plug Can Damage the 12-pin Connector

The plug on the cable that you attach to the camera's 12-pin connector must have 12 pins. Use of a smaller plug, such as one with 10 pins or 8 pins, can damage the pins in the camera's 12-pin connector.

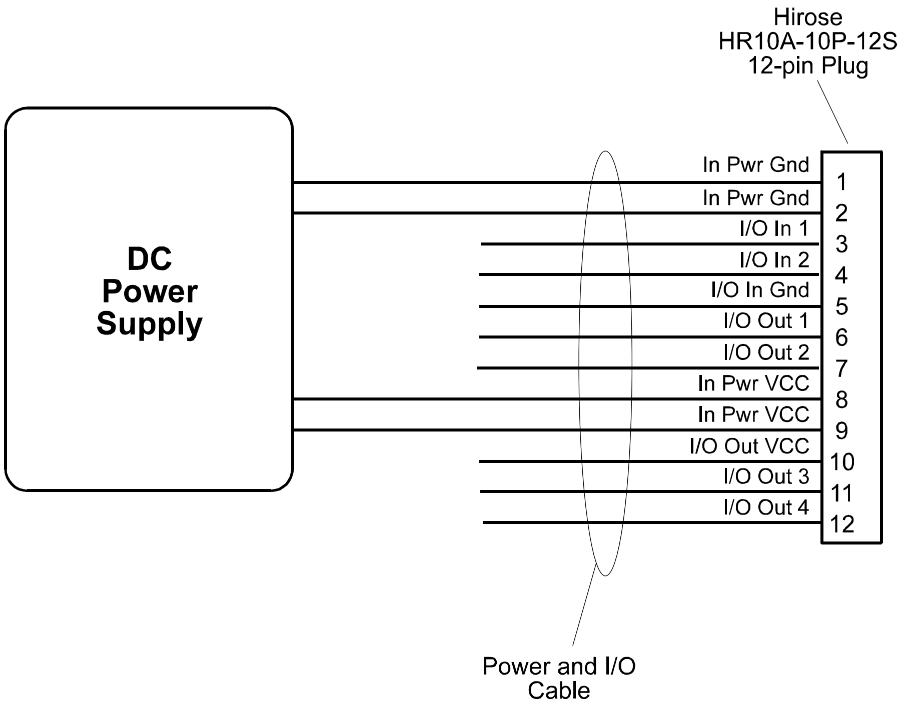


Fig. 38: Power and I/O Cable

Note

To avoid a voltage drop with long input power wires, we recommend that you supply input power VCC through two separate wires between the power supply and the camera as shown in the figure above.

We also recommend that you supply input power ground through two separate wires between the power supply and the camera as shown in the figure.

9.5 Input Power

Input power must be supplied to the camera's 12-pin connector via the power and I/O cable. Nominal input voltage is +12 VDC, however, the camera will operate properly on any input voltage between +12.0 VDC and +24.0 VDC with less than one percent ripple. Power consumption is as shown in the specification tables in Section 1 of this manual.

.

CAUTION

Voltage Outside of Specified Range Can Damage the Camera

If the voltage of the input power to the camera is greater than +24 VDC damage to the camera can result. If the voltage is less than +12 VDC, the camera may operate erratically.

CAUTION

An Incorrect Plug Can Damage the 12-pin Connector

The plug on the cable that you attach to the camera's 12-pin connector must have 12 pins. Use of a smaller plug, such as one with 10 pins or 8 pins, can damage the pins in the camera's 12-pin connector.

For more information about the 12-pin connector and the I/O cable see Section 9.2 on page 114 and Section 9.3 on page 116.

9.6 Ethernet GigE Device Information

The camera uses a standard Ethernet GigE transceiver. The transceiver is fully 100/1000 Base-T 802.3 compliant.

9.7 Input and Output Lines

9.7.1 Input Lines

The camera is equipped with two physical input lines designated as Input Line 1 and Input Line 2. The input lines are accessed via the 12-pin receptacle on the back of the camera.

As shown in the I/O line schematic, each input line is opto-isolated. For each line, the minimum input voltage to indicate a logical one is +5 VDC and the maximum is +24 VDC. An input voltage less than +5 VDC means a logical zero. The current draw for each input line is between 5 and 15 mA.

Figure 39 shows an example of a typical circuit you can use to input a signal into the camera.

By default, Input Line 1 is assigned to receive an external hardware trigger (ExTrig) signal that can be used to control the start of image acquisition.

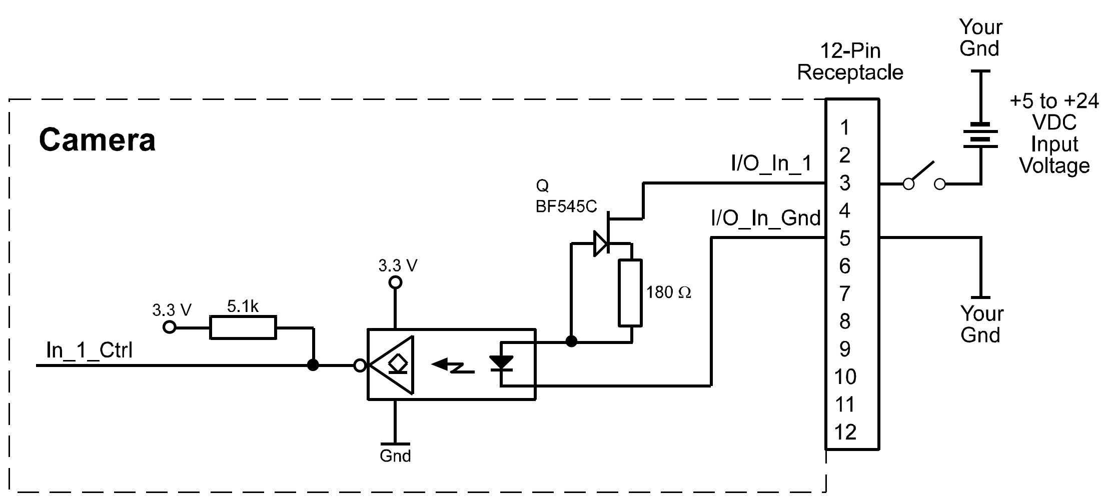


Fig. 39: Typical Input Circuit

For more information about input line pin assignments and pin numbering, see Section 9.2 on page 114.

For more information about how to use an ExTrig signal to control acquisition start, see Section 10.3 on page 132.

For more information about configuring the input lines, see Section 12.1 on page 189.

9.7.2 Output Lines

The camera is equipped with four physical output lines designated as Output Line 1, Output Line 2, Output Line 3, and Output Line 4. The output lines are accessed via the 12-pin receptacle on the back of the camera.

As shown in the I/O schematic, each output line is opto-isolated. The minimum VCC that must be applied is +5 VDC and the maximum is +24 VDC. The maximum current allowed through an output circuit is 100 mA.

A conducting transistor means a logical one and a non-conducting transistor means a logical zero.

Figure 40 shows a typical circuit you can use to monitor an output line with a voltage signal. The circuit in Figure 40 is monitoring output line 1.

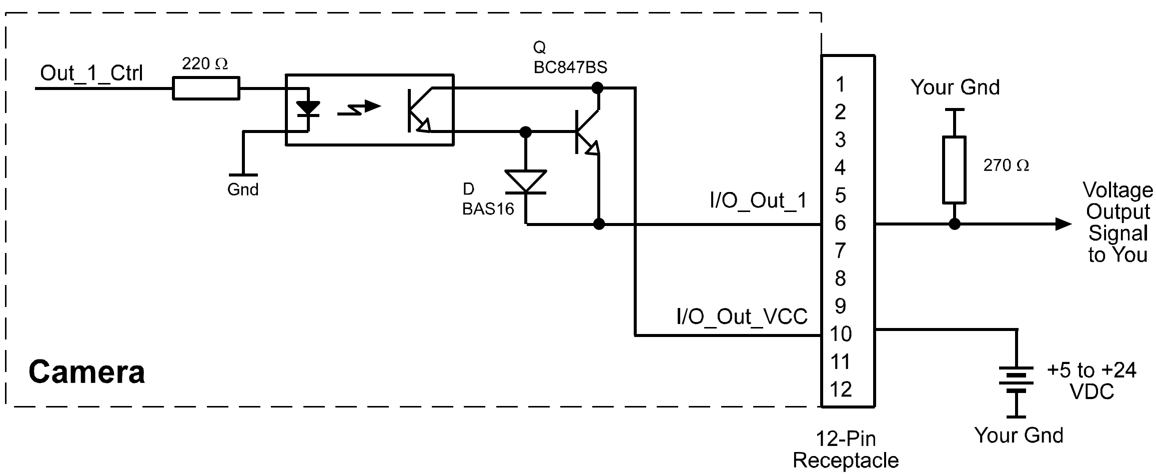


Fig. 40: Typical Voltage Output Circuit

Figure 41 shows a typical circuit you can use to monitor an output line with an LED or an opto-coupler. In this example, the voltage for the external circuit is +24 VDC. Current in the circuit is limited by an external resistor. The circuit in Figure 41 is monitoring output line 1.

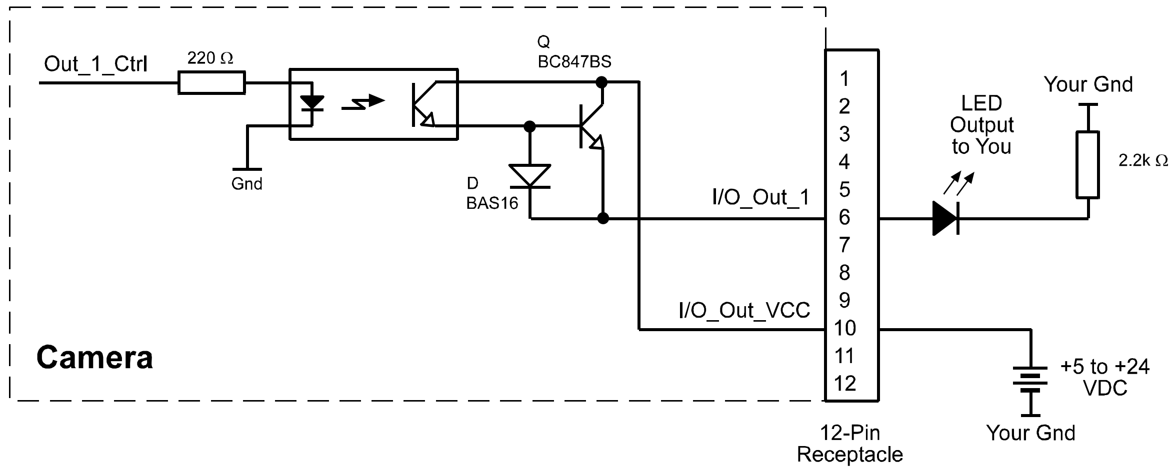


Fig. 41: Typical LED Output Signal

By default, the camera's exposure active (ExpAc) signal is assigned to Output Line 1. The exposure active signal indicates when exposure is taking place.

By default, the camera's trigger ready (TrigRdy) is assigned to Output Line 2. The trigger ready signal goes high to indicate the earliest point at which exposure start for the next frame can be triggered.

The assignment of camera output signals to physical output lines can be changed by the user.

For more information about output line pin assignments and pin numbering, see Section 9.2 on page 114.

For more information about the exposure active signal, see Section Section 10.8 on page 149.

For more information about the trigger ready signal, see Section Section 10.7 on page 145.

For more information about assigning camera output signals to physical output lines, see Section 12.2.1 on page 191.

9.7.3 Output Line Response Time

Response times for the output lines on the camera are as shown below.

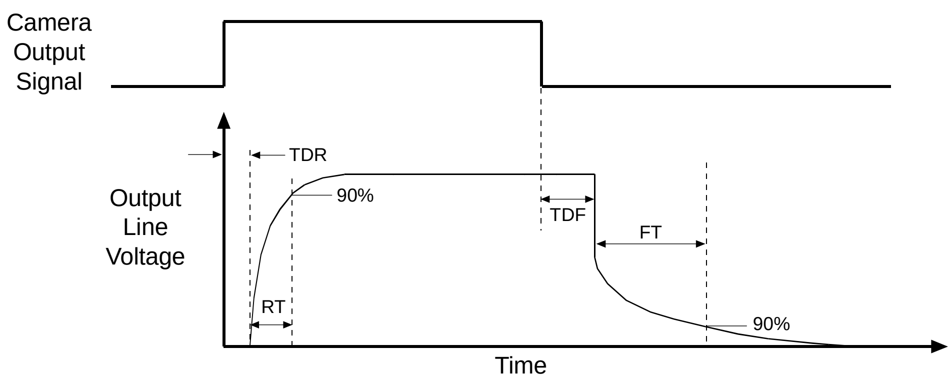


Fig. 42: Output Line Response Times

Time Delay Rise (TDR) = 0.7 µs

Rise Time (RT) = 1.3 - 3.0 µs

Time Delay Fall (TDF) = 1 - 12 µs

Fall Time (FT) = 10 - 14 µs

Note

The response times for the output lines on your camera will fall into the ranges specified above. The exact response time for your specific application will depend on the external resistor and the applied voltage you use.

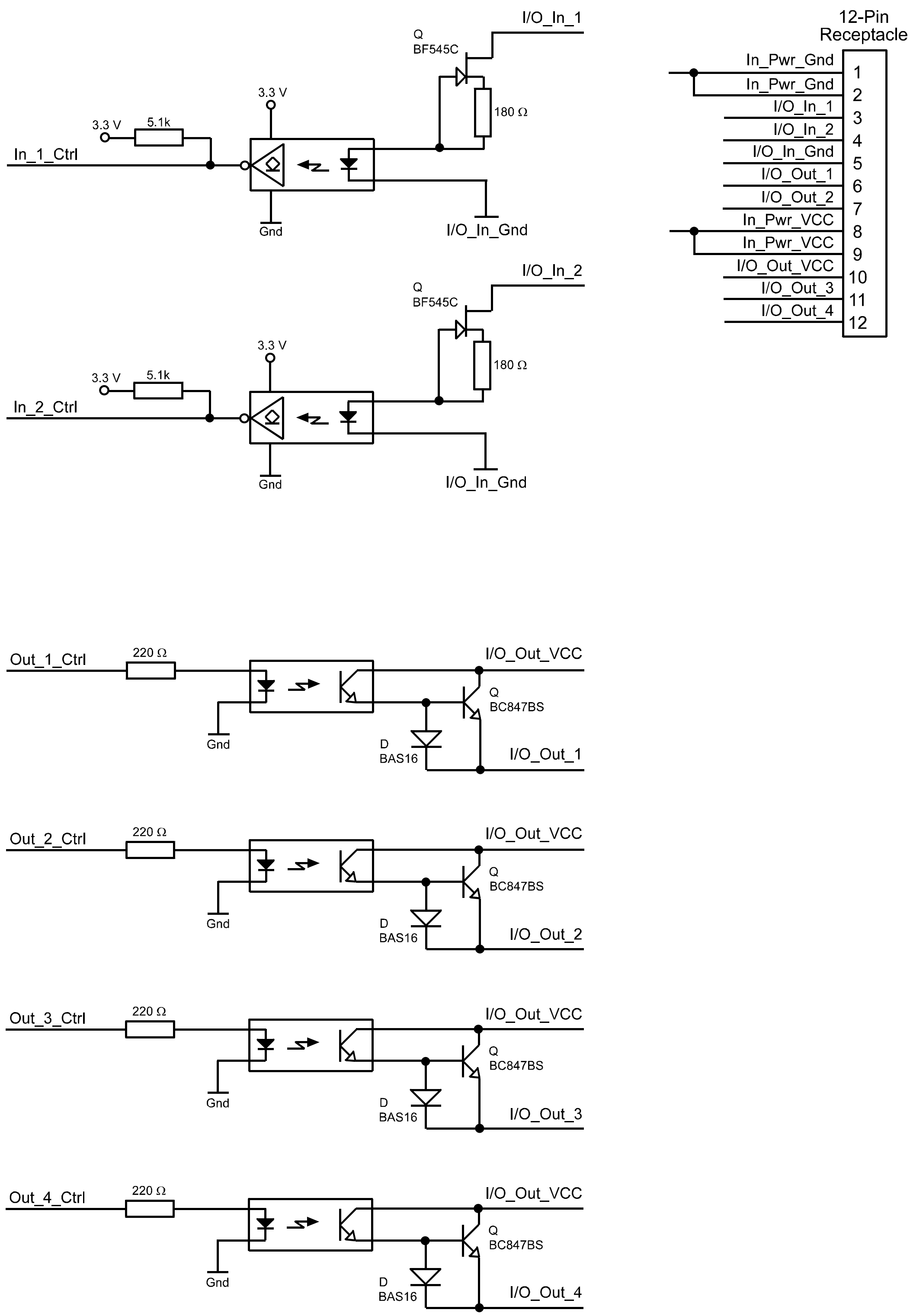


Fig. 43: I/O Line Schematic

10 Image Acquisition Control

This section provides detailed information about controlling image acquisition. You will find details about setting the exposure time for each acquired image and about how the camera's maximum allowed acquisition frame rate can vary depending on the current camera settings.

10.1 Controlling Image Acquisition with Parameters Only (No Triggering)

You can configure the camera so that image acquisition will be controlled by simply setting the value of several parameters via the camera's API. When the camera is configured to acquire images based on parameter values only, a software trigger or an external hardware trigger (ExTrig) signal is not required.

You can set the camera so that it will acquire images one at a time or so that it will acquire images continuously.

10.1.1 Switching Off Triggering

If you want to control image acquisition based on parameter settings alone, you must make sure that the camera's acquisition start trigger is set to off. Setting the acquisition start trigger is a two step process:

- First use the camera's Trigger Selector parameter to select the Acquisition Start trigger.
- Second use the camera's Trigger Mode parameter to set the selected trigger to Off.

You can set the Trigger Selector and the Trigger Mode parameter value from within your application software by using the pylon API. The following code snippet illustrates using the API to set the selector and the parameter value:

```
Camera.TriggerSelector.SetValue( TriggerSelector_AcquisitionStart );
Camera.TriggerMode.SetValue( TriggerMode_Off );
```

For detailed information about using the pylon API, refer to the Basler pylon Programmer's Guide and API Reference.

You can also use the Basler pylon Viewer application to easily set the parameters.

For more information about the pylon Viewer, see Section 2.2 on page 32 and Section 2.4 on page 52.

10.1.2 Acquiring One Image at a Time

In “single frame” operation, the camera acquires and transmits a single image. To select single frame operation, the camera’s Acquisition Mode parameter must be set to Single Frame.

To begin image acquisition, execute an Acquisition Start command. Exposure time is determined by the value of the camera’s exposure time parameter.

When using the single frame method to acquire images, you must not begin acquiring a new image until the previously captured image has been completely transmitted to the host PC.

You can set the Acquisition Mode parameter value from within your application software by using the pylon API. The following code snippet illustrates using the API to set the parameter value:

```
Camera.AcquisitionMode.SetValue( AcquisitionMode_SingleFrame );
```

You can also execute the Acquisition Start command by using the API.

For detailed information about using the pylon API, refer to the Basler pylon Programmer’s Guide and API Reference.

You can also use the Basler pylon Viewer application to easily set the parameters.

For more information about the pylon Viewer, see Section 2.2 on page 32 and Section 2.4 on page 52.

For more information about the camera’s exposure time parameter, see Section 10.4 on page 139.

10.1.3 Acquiring Images Continuously (Free-run)

In “continuous frame” operation, the camera continuously acquires and transmits images. To select continuous frame operation, the camera’s Acquisition Mode parameter must be set to Continuous. (Note that operating the camera in continuous frame mode without the use of a trigger is also commonly called "free run".)

To begin acquiring images, issue an Acquisition Start command. The exposure time for each image is determined by the value of the camera’s exposure time parameter. Acquisition start for the second and subsequent images is automatically controlled by the camera. Image acquisition and transmission will stop when you execute an Acquisition Stop command.

When the camera is operating in continuous frame mode without triggering, the acquisition frame rate is determined by the Acquisition Frame Rate Abs parameter:

- If the parameter is enabled and set to a value less than the maximum allowed acquisition frame rate, the camera will acquire images at rate specified by the parameter setting.
- If the parameter is disabled or is set to a value greater than the maximum allowed acquisition frame rate, the camera will acquire images at the maximum allowed.

Note that before you can use the Acquisition Frame Rate Abs parameter to control the frame rate, the parameter must be enabled.

You can set the Acquisition Mode parameter value and you can enable and set the Acquisition Frame Rate Abs parameter from within your application software by using the pylon API. The following code snippets illustrate using the API to set the parameter values:

```
// set camera in continous mode
Camera.AcquisitionMode.SetValue( AcquisitionMode_Continuous );
// set a frame rate and getting the resulting frame rate
Camera.AcquisitionFrameRateEnable.SetValue( true );
Camera.AcquisitionFrameRateAbs.SetValue( 20.5 );
double resultingFrameRate = Camera.ResultingFrameRateAbs.GetValue();
```

You can also execute the Acquisition Start and Stop commands by using the API.

For detailed information about using the pylon API, refer to the Basler pylon Programmer's Guide and API Reference.

You can also use the Basler pylon Viewer application to easily set the parameters.

For more information about the pylon Viewer, see Section 2.2 on page 32 and Section 2.4 on page 52.

For more information about the camera's exposure time parameter, see Section 10.4 on page 139.

For more information about determining the maximum allowed acquisition frame rate, see Section 10.10 on page 152.

Note

The explanations in Section 10.1.2 and Section 10.1.3 are intended to give you a basic idea of how parameters alone can be used to control image acquisition. For a more complete description, refer to the Basler pylon Programmer's Guide and to the sample programs included in the Basler pylon Software Development Kit (SDK).

10.2 Controlling Image Acquisition with a Software Trigger

You can configure the camera so that image acquisition will be controlled by issuing a software trigger. The software trigger is issued by executing a Trigger Software command.

Image acquisition starts when the Trigger Software command is executed. The exposure time for each image is determined by the value of the camera's exposure time parameter. Figure 44 illustrates image acquisition with a software trigger.

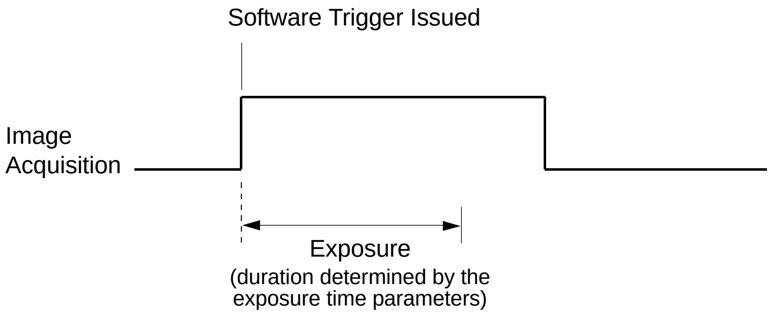


Fig. 44: Image Acquisition with a Software Trigger

When controlling image acquisition with a software trigger, you can set the camera so that it will react to a single software trigger or so that it will react to a continuous series of software triggers.

10.2.1 Enabling the Software Trigger Feature

To enable the software trigger feature:

- Use the camera's Trigger Selector parameter to select the Acquisition Start trigger.
- Use the camera's Trigger Mode parameter to set the mode to On.
- Use the camera's Trigger Source parameter to set the trigger source to Software.
- Use the Exposure Mode parameter to set the exposure mode to timed.

You can set these parameter values from within your application software by using the pylon API. The following code snippet illustrates using the API to set the parameter values:

```
Camera.TriggerSelector.SetValue(TriggerSelector_AcquisitionStart);
Camera.TriggerMode.SetValue( TriggerMode_On );
Camera.TriggerSource.SetValue( TriggerSource_Software );
Camera.ExposureMode.SetValue( ExposureMode_Timed );
```

For detailed information about using the pylon API, refer to the Basler pylon Programmer's Guide and API Reference.

You can also use the Basler pylon Viewer application to easily set the parameters.

For more information about the pylon Viewer, see Section 2.2 on page 32 and Section 2.4 on page 52.

10.2.2 Acquiring a Single Image by Applying One Software Trigger

You can set the camera to react to a single software trigger and then issue a software trigger to begin image acquisition. To do so, follow this sequence:

1. Access the camera's API and set the exposure time parameter for your desired exposure time.
2. Set the value of the camera's Acquisition Mode parameter to Single Frame.
3. Execute an Acquisition Start command.
   (In single frame mode, executing the start command prepares the camera to react to a single trigger.)
4. When you are ready to begin an image acquisition, execute a Trigger Software command.
5. Image acquisition will start and exposure will continue for the length of time you specified in step 1.
6. At the end of the specified exposure time, readout and transmission of the acquired image will take place.
7. At this point, the camera would ignore any additional software triggers. To acquire another image, you must:
   a. Repeat step 3 to prepare the camera to react to a software trigger.
   b. Repeat step 4 to issue a software trigger.

If you use the single image acquisition process repeatedly, you must not begin acquisition of a new image until transmission of the previously acquired image is complete.

You can set the exposure time and the Acquisition Mode parameter values from within your application software by using the pylon API. You can also execute the Acquisition Start and Trigger Software commands. The following code snippets illustrate using the API to set the parameter values and execute the commands:

```
Camera.ExposureTimeRaw.SetValue( 200 );
Camera.AcquisitionMode.SetValue( AcquisitionMode_SingleFrame );
// prepare for image capture
Camera.AcquisitionStart.Execute( );
Camera.TriggerSoftware.Execute( );
// retrieve the captured image
```

For detailed information about using the pylon API, refer to the Basler pylon Programmer's Guide and API Reference.

You can also use the Basler pylon Viewer application to easily set the parameters.

For more information about the pylon Viewer, see Section 2.2 on page 32 and Section 2.4 on page 52.

For more information about the camera's exposure time parameter, see Section 10.4 on page 139.

10.2.3 Acquiring Images by Applying a Series of Software Triggers

You can set the camera to react to multiple applications of the software trigger and then apply a series of software triggers to acquire images. To do so, follow this sequence:

1. Access the camera's API and set the exposure time parameter for your desired exposure time.
2. Set the value of the camera's Acquisition Mode parameter to Continuous.
3. Execute an Acquisition Start command, this prepares the camera to react to software triggers.
4. When you are ready to begin an image acquisition, execute a Trigger Software command.
5. Image acquisition will start and exposure will continue for the length of time you specified in step 1.
6. At the end of the specified exposure time, readout and transmission of the acquired image will take place.
7. To acquire another image, go to step 4.
8. Execute an Acquisition Stop command. The camera will no longer react to software triggers.

If you are acquiring images using a series of software triggers, you must avoid acquiring images at a rate that exceeds the maximum allowed with the current camera settings.

You should also be aware that if the Acquisition Frame Rate Abs parameter is enabled, it will influence the rate at which the Trigger Software command can be applied:

- If the Acquisition Frame Rate Abs parameter is set to a value less than the maximum allowed, you can trigger acquisition at any rate up to the set value.
- If the Acquisition Frame Rate Abs parameter is set to a value greater than the maximum allowed, you can trigger acquisition at any rate up to the maximum allowed image acquisition rate with the current camera settings.

You can set the exposure time and the Acquisition Mode parameter values from within your application software by using the pylon API. You can also execute the Acquisition Start and Trigger Software commands. The following code snippets illustrate using the API to set the parameter values and execute the commands:

```
// issuing software trigger commands
Camera.ExposureTimeRaw.SetValue( 200 );
Camera.AcquisitionMode.SetValue( AcquisitionMode_Continuous );
// prepare for image acquisition here
     Camera.AcquisitionStart.Execute( );
     while ( ! finished )
     {
         Camera.TriggerSoftware.Execute( );
         // retrieve acquired image here
     }
     Camera.AcquisitionStop.Execute( );

// how to set and test the Acquisition Frame Rate
Camera.AcquisitionFrameRateAbs.SetValue( 60.0 );
```

```
double resultingFrameRate = Camera.ResultingFrameRateAbs.GetValue( );

// how to disable the FrameRateAbs parameter
Camera.AcquisitionFrameRateEnable.SetValue( false );
```

For detailed information about using the pylon API, refer to the Basler pylon Programmer's Guide and API Reference.

You can also use the Basler pylon Viewer application to easily set the parameters.

For more information about the pylon Viewer, see Section 2.2 on page 32 and Section 2.4 on page 52.

For more information about the camera's exposure time parameter, see Section 10.4 on page 139.

For more information about determining the maximum allowed acquisition frame rate, see Section 10.10 on page 152.

Note

The explanations in Section 10.2.2 and Section 10.2.3 are intended to give you a basic idea of how the use of a software trigger works. For a more complete description, refer to the Basler pylon Programmer's Guide and to the sample programs included in the Basler pylon Software Development Kit (SDK).

10.3 Controlling Image Acquisition with a Hardware Trigger

You can configure the camera so that an external hardware trigger (ExTrig) signal applied to one of the input lines will control image acquisition. A rising edge or a falling edge of the ExTrig signal can be used to trigger image acquisition.

The ExTrig signal can be periodic or non-periodic. When the camera is operating under control of an ExTrig signal, the period of the ExTrig signal will determine the rate at which the camera is acquiring images:

$$\frac{1}{\text{ExTrig period in seconds}} = \text{Acquisition Frame Rate}$$

For example, if you are operating a camera with an ExTrig signal period of 20 ms (0.020 s):

$$\frac{1}{0.020} = 50 \text{ fps}$$

So in this case, the acquisition frame rate is 50 fps.

The minimum high time for a rising edge trigger (or low time for a falling edge trigger) is 100 nanoseconds.

By default, input line 1 is assigned to receive an ExTrig signal.

When you are triggering image acquisition with an ExTrig signal, you must not acquire images at a rate that exceeds the maximum allowed for the current camera settings.

For more information about setting the camera for hardware triggering and selecting the input line to receive the ExTrig signal, see Section 10.3.2 on page 135.

For more information about determining the maximum allowed acquisition frame rate, see Section 10.10 on page 152.

10.3.1 Exposure Modes

If you are triggering exposure start with an ExTrig signal, two exposure modes are available, "timed" and "trigger width."

Timed Exposure Mode

When timed mode is selected, the exposure time for each image is determined by the value of the camera's exposure time parameter. If the camera is set for rising edge triggering, the exposure time starts when the ExTrig signal rises. If the camera is set for falling edge triggering, the exposure time starts when the ExTrig signal falls. Figure 45 illustrates timed exposure with the camera set for rising edge triggering.

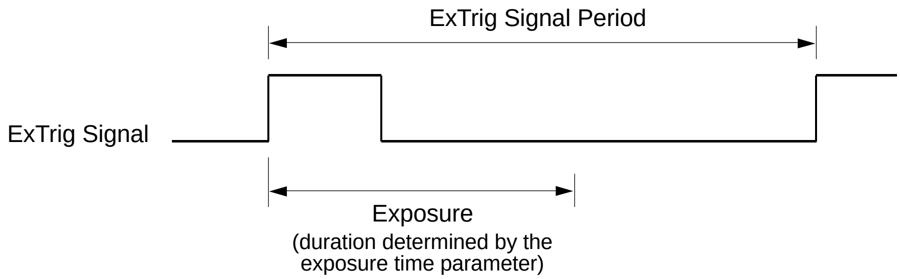


Fig. 45: Timed Exposure with Rising Edge Triggering

Trigger Width Exposure Mode

When trigger width exposure mode is selected, the length of the exposure will be directly controlled by the ExTrig signal. If the camera is set for rising edge triggering, the exposure time begins when the ExTrig signal rises and continues until the ExTrig signal falls. If the camera is set for falling edge triggering, the exposure time begins when the ExTrig signal falls and continues until the ExTrig signal rises. Figure 46 illustrates trigger width exposure with the camera set for rising edge triggering.

Trigger width exposure is especially useful if you intend to vary the length of the exposure time for each captured image.

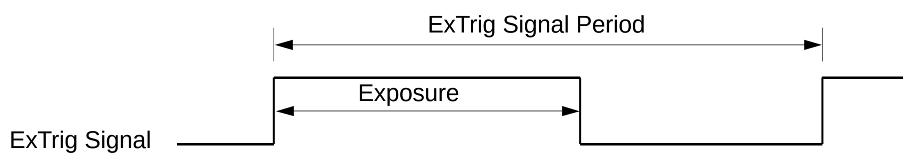


Fig. 46: Trigger Width Exposure with Rising Edge Triggering

> **Note**
>
> The trigger width exposure mode **is not** available on scA750-60 cameras. The trigger width exposure mode is available on all other camera models.

When you operate the camera in trigger width exposure mode, you must use the camera's exposure setting to set an exposure time. The exposure time setting will be used by the camera to operate the trigger ready signal.

You should adjust the exposure setting to represent the shortest exposure time you intend to use. For example, assume that you will be using trigger width exposure and that you intend to use the ExTrig signal to vary the exposure time in a range from 3000 µs to 5500 µs. In this case you would use the exposure setting to set the exposure time to 3000 µs.

If you are using the trigger width exposure mode and the camera is operating with overlapped exposures, there is something you must keep in mind. If the action of the ExTrig signal would end the current exposure while readout of the previously acquired image is still taking place, the camera will automatically continue the exposure until readout of the previous image is complete. This situation is illustrated Figure 45 for rising edge operation. On the first cycle of the ExTrig signal shown in the figure, the signal rises and falls while readout is taking place. Normally you would expect exposure to take place only when the ExTrig signal is high. But since the signal falls while the previous frame is still reading out, the camera automatically extends exposure until the readout is complete. On the second cycle of the ExTrig signal shown in the figure, the signal rises during previous frame readout, but falls after the readout is complete. This is a normal situation and exposure would be determined by the high time of the ExTrig signal as you would expect.

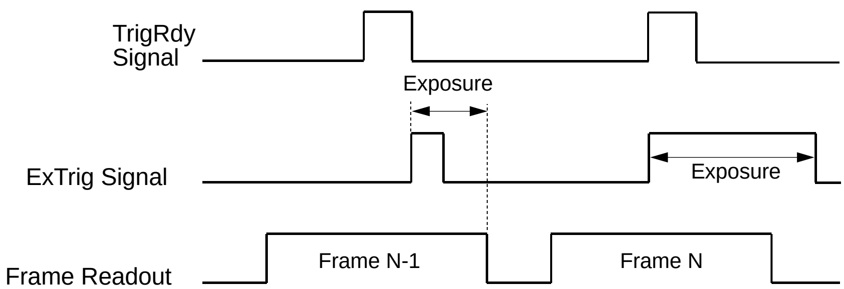


Fig. 47: Overlapped Level Control Exposure

You can set the exposure time parameter value and select an exposure mode from within your application software by using the pylon API. The following code snippets illustrate using the API to set the exposure time parameter and select the exposure mode:

```
// set for the timed exposure mode, set exposure time to 3000 µs
Camera.ExposureMode.SetValue( ExposureMode_Timed );

Camera.ExposureTimeAbs.SetValue( 3000 );
// set for the width exposure mode, set minimum exposure time to 3000 µs
Camera.ExposureMode.SetValue( ExposureMode_TriggerWidth );
Camera.ExposureTimeAbs.SetValue( 3000 );
```

For detailed information about using the pylon API, refer to the Basler pylon Programmer's Guide and API Reference.

You can also use the Basler pylon Viewer application to easily set the parameters.

For more information about the pylon viewer, see Section 2.2 on page 32 and Section 2.4 on page 52.

For more information about the camera's exposure time parameter, see Section 10.4 on page 139.

For more information about overlapped exposure, see Section 10.5 on page 142.

10.3.2 Setting the Camera for Hardware Triggering

To set the camera for hardware triggering:

- Use the Trigger Selector parameter to select the Acquisition Start trigger.
- Use the Trigger Mode parameter to set the trigger mode to On.
- Use the Trigger Source parameter to set the camera to accept the hardware trigger signal on input line 1 or on input line 2.
- Use the Trigger Activation parameter to set the camera for rising edge triggering or for falling edge triggering.

You can set these parameter values from within your application software by using the pylon API. The following code snippet illustrates using the API to set the parameter values:

```
Camera.TriggerSelector.SetValue( TriggerSelector_AcquisitionStart );
Camera.TriggerMode.SetValue( TriggerMode_On );
Camera.TriggerSource.SetValue ( TriggerSource_Line1 );
Camera.TriggerActivation.SetValue( TriggerActivation_RisingEdge );
```

For detailed information about using the pylon API, refer to the Basler pylon Programmer's Guide and API Reference.

You can also use the Basler pylon Viewer application to easily set the parameters.

For more information about the pylon Viewer, see Section 2.2 on page 32 and Section 2.4 on page 52.

10.3.3 Acquiring a Single Image by Applying One Hardware Trigger Transition

You can set the camera to react to a single transition of an external hardware trigger (ExTrig) signal and then you can transition the ExTrig signal to begin image acquisition. When you are using an ExTrig signal to start image acquisition, you should monitor the camera's trigger ready (TrigRdy) output signal and you should base the use of your ExTrig signal on the state of the trigger ready signal.

To set the camera to react to a single ExTrig signal transition, follow the sequence below. The sequence assumes that you have set the camera for rising edge triggering and for the timed exposure mode.

1. Access the camera's API and set the exposure time parameter for your desired exposure time.
2. Set the value of the camera's Acquisition Mode parameter to Single Frame.
3. Execute an Acquisition Start command.
   (In single frame mode, executing the start command prepares the camera to react to a single trigger.)
4. Check the state of the camera's Trigger Ready signal:
   a. If the TrigRdy signal is high, you can transition the ExTrig signal when desired.
   b. If the TrigRdy signal is low, wait until TrigRdy goes high and then transition the ExTrig signal when desired.
5. When the ExTrig signal transitions from low to high, image acquisition will start. Exposure will continue for the length of time you specified in step 1.
6. At the end of the specified exposure time, readout and transmission of the acquired image will take place.
7. At this point, the camera would ignore any additional ExTrig signal transitions. To acquire another image, you must:
   a. Repeat step 3 to prepare the camera to react to a hardware trigger transition.
   b. Repeat step 4 to check if the camera is ready to acquire an image.
   c. Repeat step 5 to begin image acquisition

You can set the exposure time and the Acquisition Mode parameter values from within your application software by using the pylon API. You can also execute the Acquisition Start command. The following code snippet illustrates using the API to set the parameter values and execute the command:

```
Camera.TriggerSelector.SetValue( TriggerSelector_AcquisitionStart );
Camera.ExposureMode.SetValue( ExposureMode_Timed );
Camera.ExposureTimeAbs.SetValue( 3000 );
Camera.TriggerActivation.SetValue( TriggerActivation_RisingEdge );
Camera.AcquisitionMode.SetValue( AcquisitionMode_SingleFrame );
Camera.AcquisitionStart.Execute( );
```

For detailed information about using the pylon API, refer to the Basler pylon Programmer's Guide and API Reference.

You can also use the Basler pylon Viewer application to easily set the parameters.

For more information about the pylon Viewer, see Section 2.2 on page 32 and Section 2.4 on page 52.

For more information about the Trigger Ready signal, see Section 10.7 on page 145.

For more information about the camera’s exposure time parameter, see Section 10.4 on page 139.

10.3.4 Acquiring Images by Applying a Series of Hardware Trigger Transitions

You can set the camera so that it will react to a continuous series of external hardware trigger (ExTrig) transitions and then you can cycle the ExTrig signal as desired to begin image acquisition. When you are using an ExTrig signal to start image acquisition, you should monitor the camera’s trigger ready (TrigRdy) output signal and you should base the use of your ExTrig signal on the state of the trigger ready signal.

To set the camera to react continuously to ExTrig signal transitions, follow the sequence below. The sequence assumes that you have set the camera for rising edge triggering and for the timed exposure mode.

1. Access the camera’s API and set the exposure time parameters for your desired exposure time.
2. Set the value of the camera’s Acquisition Mode parameter to Continuous.
3. Execute an Acquisition Start command. This prepares the camera to react to the trigger signals.
4. Check the state of the camera’s Trigger Ready signal:
   a. If the TrigRdy signal is high, you can transition the ExTrig signal when desired.
   b. If the TrigRdy signal is low, wait until TrigRdy goes high and then transition the ExTrig signal when desired.
5. When the ExTrig signal transitions from low to high, image acquisition will start. Exposure will continue for the length of time you specified in step 1.
6. At the end of the specified exposure time, readout and transmission of the acquired image will take place.
7. Repeat steps 4 and 5 each time you want to start another image acquisition.
8. Execute an Acquisition Stop command. The camera will no longer react to hardware triggers.

If you are acquiring images using a series of hardware trigger transitionss, you must avoid acquiring images at a rate that exceeds the maximum allowed with the current camera settings. You can avoid triggering image acquistion at too high a rate by using the trigger ready signal as described above.

You should also be aware that if the Acquisition Frame Rate Abs parameter is enabled, it will influence the rate at which images can be acquired:

- If the Acquisition Frame Rate Abs parameter is set to a value less than the maximum allowed, you can trigger acquisition at any rate up to the set value.
- If the Acquisition Frame Rate Abs parameter is set to a value greater than the maximum allowed, you can trigger acquisition at any rate up to the maximum allowed image acquisition rate with the current camera settings.

You can set the exposure time and the Acquisition Mode parameter values from within your application software by using the pylon API. You can also execute the Acquisition Start and Stop commands. The following code snippet illustrates using the API to set the parameter values and execute the commands:

```
Camera.TriggerSelector.SetValue( TriggerSelector_AcquisitionStart );
Camera.ExposureMode.SetValue( ExposureMode_Timed );
Camera.ExposureTimeAbs.SetValue( 3000 );
Camera.TriggerActivation.SetValue( TriggerActivation_RisingEdge );
Camera.AcquisitionMode.SetValue( AcquisitionMode_Continuous );
Camera.AcquisitionStart.Execute( );
Camera.AcquisitionStop.Execute( );
```

For detailed information about using the pylon API, refer to the Basler pylon Programmer's Guide and API Reference.

You can also use the Basler pylon Viewer application to easily set the parameters.

For more information about the pylon Viewer, see Section 2.2 on page 32 and Section 2.4 on page 52.

For more information about the Trigger Ready signal, see Section 10.7 on page 145.

For more information about the camera's exposure time parameter, see Section 10.4 on page 139.

Note

The explanations in Section 10.3.3 and Section 10.3.4 are intended to give you a basic idea of how the use of a software trigger works. For a more complete description, refer to the Basler pylon Programmer's Guide and to the sample programs included in the Basler pylon Software Development Kit (SDK).

10.4 Exposure Time Parameters

Many of the camera's image acquisition modes require you to specify an exposure time. There are two ways to set exposure time: by setting "raw" values or by setting an "absolute value". The two methods are described below. You can use whichever method you prefer to set the exposure time.

The exposure time must not be set below a minimum specified value. The minimum exposure time varies by camera model as shown in Table 6.

Camera Model	Minimum Allowed Exposure Time
scA640-70gm/gc	80 µs
scA640-74gm/gc	80 µs
scA750-60gm/gc	124 µs
scA780-54gm/gc	80 µs
scA1000-20gm/gc	120 µs
scA1000-30gm/gc	100 µs
scA1390-17gm/gc	100 µs
scA1400-17gm/gc	100 µs
scA1600-14gm/gc	100 µs

Table 6: Minimum Allowed Exposure Time

10.4.1 Setting the Exposure Time Using "Raw" Settings

When exposure time is set using "raw" values, the exposure time will be determined by a combination of two elements. The first element is the value of the Exposure Time Raw parameter, and the second element is the Exposure Time Base. The exposure time is determined by the product of these two elements:

Exposure Time = (Exposure Time Raw Parameter Value) x (Exposure Time Base)

By default, the Exposure Time Base is fixed at 20 µs on all camera models except the scA750-60. On scA750-60 cameras, the default Exposure Time Base is 31 µs.

Typically, the exposure time is adjusted by setting only the Exposure Time Raw parameter. The Exposure Time Raw parameter value can range from 1 to 4095. So if the parameter value was set to 100 on an scA640-70 camera, for example, the exposure time will be 100 x 20 µs or 2000 µs.

Changing the Exposure Time Base

Normally, the exposure time is adjusted by setting the value of the Exposure Time Raw parameter as explained above. However, if you require an exposure time that is longer than what you can achieve by changing the value of the Exposure Time Raw parameter alone, the Exposure Time Base Abs parameter can be used to change the exposure time base.

The Exposure Time Base Abs parameter value sets the exposure time base in µs and this parameter can be used to change the exposure time base.

On all camera models except the scA750-60, the default exposure time base is 20 µs and the time base can be changed in increments of 1 µs.

On scA750-60 cameras, the default exposure time base is 31 µs and the time base can be changed in increments of 31 µs.

You can set the Exposure Time Raw and Exposure Time Base Abs parameter values from within your application software by using the pylon API. The following code snippet illustrates using the API to set the parameter values:

```
Camera.ExposureMode.SetValue( ExposureMode_Timed );
Camera.ExposureTimeRaw.SetValue( 100 );

Camera.ExposureTimeBaseAbs.SetValue( 186 );
```

For detailed information about using the pylon API, refer to the Basler pylon Programmer's Guide and API Reference.

You can also use the Basler pylon Viewer application to easily set the parameters.

For more information about the pylon Viewer, see Section 2.2 on page 32 and Section 2.4 on page 52.

10.4.2 Setting the Exposure Time Using "Absolute" Settings

You can also set the exposure time by using an "absolute" value. This is accomplished by setting the Exposure Time Abs parameter. The units for setting this parameter are µs and the value can be set in increments of 1 µs.

When you use the Exposure Time Abs parameter to set the exposure time, the camera accomplishes the setting change by automatically changing the Exposure Time Raw parameter to achieve the value specified by your Exposure Time Abs setting. This leads to a limitation that you must keep in mind if you use Exposure Time Abs parameter to set the exposure time. That is, you must set the Exposure Time Abs parameter to a value that is equivalent to a setting you could achieve by using the Exposure Time Raw parameter with the current Exposure Time Base parameter. For example, if the time base was currently set to 62 µs, you could use the Exposure Time Base Abs parameter to set the exposure to 62 µs, 124 µs, 186 µs, etc.

Note that if you set the Exposure Time Abs parameter to a value that you could not achieve by using the Exposure Time Raw and Exposure Time Base parameter, the camera will automatically change the setting for the Exposure Time Abs parameter to the nearest achieveable value.

You should also be aware that if you change the exposure time using the raw settings, the Exposure Time Abs parameter will automatically be updated to reflect the new exposure time.

You can set the Exposure Time Abs parameter value from within your application software by using the pylon API. The following code snippet illustrates using the API to set the parameter value:

```
Camera.ExposureTimeAbs.SetValue( 124 );
double resultingExpTime = Camera.ExposureTimeAbs.GetValue( );
```

For detailed information about using the pylon API, refer to the Basler pylon Programmer's Guide and API Reference.

You can also use the Basler pylon Viewer application to easily set the parameters.

For more information about the pylon Viewer, see Section 2.2 on page 32 and Section 2.4 on page 52.

10.5 Overlapping Exposure and Sensor Readout (all models except scA750-60)

Note

The information in this section applies to all camera models except the scA750-60 gm/gc. For information about scA750-60 cameras, see Section 10.6 on page 144.

The image acquisition process on the camera includes two distinct parts. The first part is the exposure of the pixels in the imaging sensor. Once exposure is complete, the second part of the process – readout of the pixel values from the sensor – takes place.

In regard to this image acquisition process, there are two common ways for the camera to operate: with “non-overlapped” exposure and with “overlapped” exposure. In the non-overlapped mode of operation, each time an image is acquired, the camera completes the entire exposure/readout process before acquisition of the next image is started. This situation is illustrated in Figure 48.

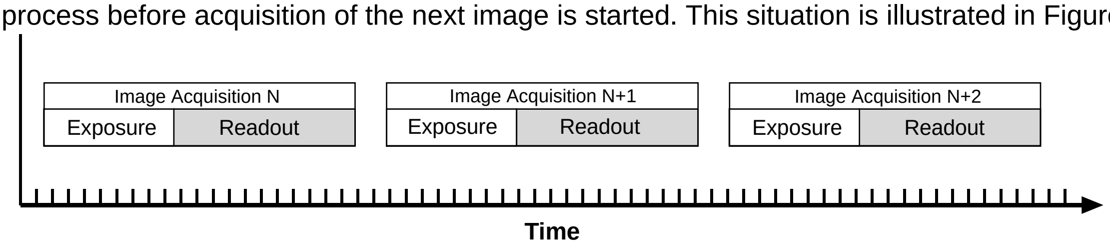


Fig. 48: Non-overlapped Exposure

While operating in a non-overlapped fashion is perfectly normal and is appropriate for many situations, it is not the most efficient way to operate the camera in terms of acquisition frame rate. On this camera, however, it is allowable to begin exposing a new image while a previously acquired image is being read out. This situation is illustrated in Figure 49 and is known as operating the camera with “overlapped” exposure.

As you can see, running the camera with readout and exposure overlapped can allow higher acquisition frame rates because the camera is performing two processes at once.

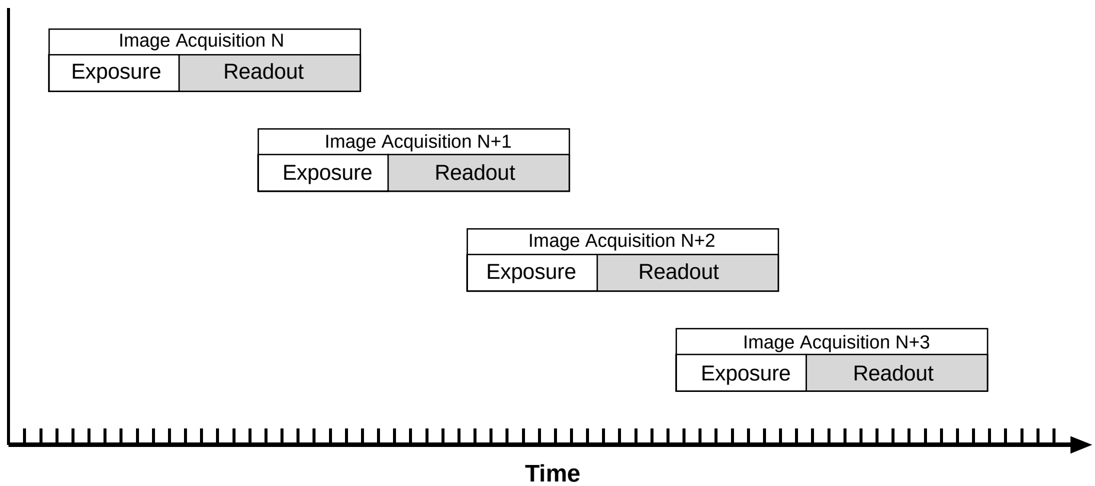


Fig. 49: Overlapped Exposure

Determining whether your camera is operating with overlapped or non-overlapped exposures is not a matter of issuing a command or switching a setting on or off. Rather the way that you operate the camera will determine whether the exposures are overlapped or not overlapped. If we define the “frame period” as the time from the start of exposure for one image acquisition to the start of exposure for the next image acquisition, then:

- Exposure will overlap when: Frame Period ≤ Exposure Time + Readout Time
- Exposure will not overlap when: Frame Period > Exposure Time + Readout Time

You can calculate the readout time for a captured image by using the formula on page 150.

10.5.1 Guidelines for Overlapped Operation

If you will be operating the camera with overlapped exposure, there are two important guidelines to keep in mind:

- You must not begin the exposure time for a new image acquisition while the exposure time of the previous acquisition is in progress.
- You must not end the exposure time of the current image acquisition until readout of the previously acquired image is complete.

The camera will ignore any trigger signals that violate these guidelines.

When you are operating a camera with overlapped exposure and using a hardware trigger signal to trigger image acquisition, you could use the camera’s exposure time parameter settings and timing formulas to calculate when it is safe to begin each new acquisition. However, there is a much more convenient way to know when it safe to begin each acquisition. The camera supplies a “trigger ready” signal that is specifically designed to let you trigger overlapped exposure safely and efficiently.

For more information about using the Trigger Ready signal with all camera models except the scA750-60 gm/gc, see Section 10.7.1 on page 145.

10.6 Exposure Must Not Overlap Sensor Readout (scA750-60 only)

Note

The information in this section only applies to scA750-60 gm/gc cameras. For information about the other camera models, see Section 10.5 on page 142.

The image acquisition process on the camera includes two distinct parts. The first part is the exposure of the pixels in the imaging sensor. Once exposure is complete, the second part of the process – readout of the pixel values from the sensor – takes place.

On these cameras, exposure for a new acquisition must not begin until readout of the previously acquired image is complete. This situation is illustrated in Figure 48.

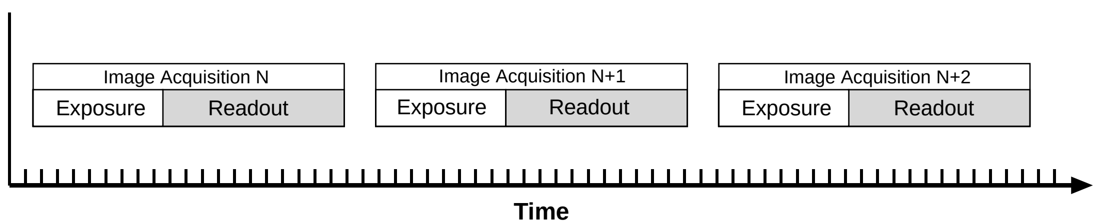


Fig. 50: Non-overlapped Readout and Exposure

A result of this characteristic is that the exposure time setting on the camera will have a direct effect on the camera's maximum allowed frame rate. At longer exposure times, the maximum allowed frame rate will be lower.

When you are operating a camera and using a hardware trigger to trigger image acquisition, you could use the camera's exposure time parameter settings and the timing formulas to calculate when it is safe to begin each new acquisition. However, there is a more convenient way to know when it safe to begin each acquisition. The camera supplies a "trigger ready" signal that is specifically designed to let you trigger acquisitions safely and efficiently.

For more information about using the Trigger Ready signal with scA750-60 gm/gc cameras, see Section 10.7.2 on page 147

10.7 Trigger Ready Signal

10.7.1 Trigger Ready Signal (all models except scA750-60)

Note

The information in this section applies to all camera models except the scA750-60 gm/gc. For information about scA750-60 cameras, see Section 10.7.2 on page 147.

As described in the previous section, the cameras can operate in an "overlapped" acquisition fashion. When the camera is operated in this manner, it is especially important that:

- the exposure time of a new image acquisition not start until exposure of the previously acquired image is complete, and
- the exposure time of a new image acquisition not end until readout of the previously acquired image is complete.

The camera supplies a "Trigger Ready" (TrigRdy) output signal you can use to ensure that these conditions are met when you are using a hardware trigger signal to trigger image acquisition. When you are acquiring images, the camera automatically calculates the earliest moment that it is safe to trigger each new acquisition. The trigger ready signal will go high when it is safe to trigger an acquisition, will go low when the acquisition has started, and will go high again when it is safe to trigger the next acquisition (see Figure 51). The camera calculates the rise of the trigger ready signal based on the current exposure time parameter setting, the current size of the area of interest, and the time it will take to readout the captured pixel values from the sensor.

The trigger ready signal is especially useful if you want to run the camera at the maximum acquisition frame capture rate for the current conditions. If you monitor the trigger ready signal and you trigger acquisition of each new image immediately after the signal goes high, you will be sure that the camera is operating at the maximum acquisition frame rate for the current conditions.

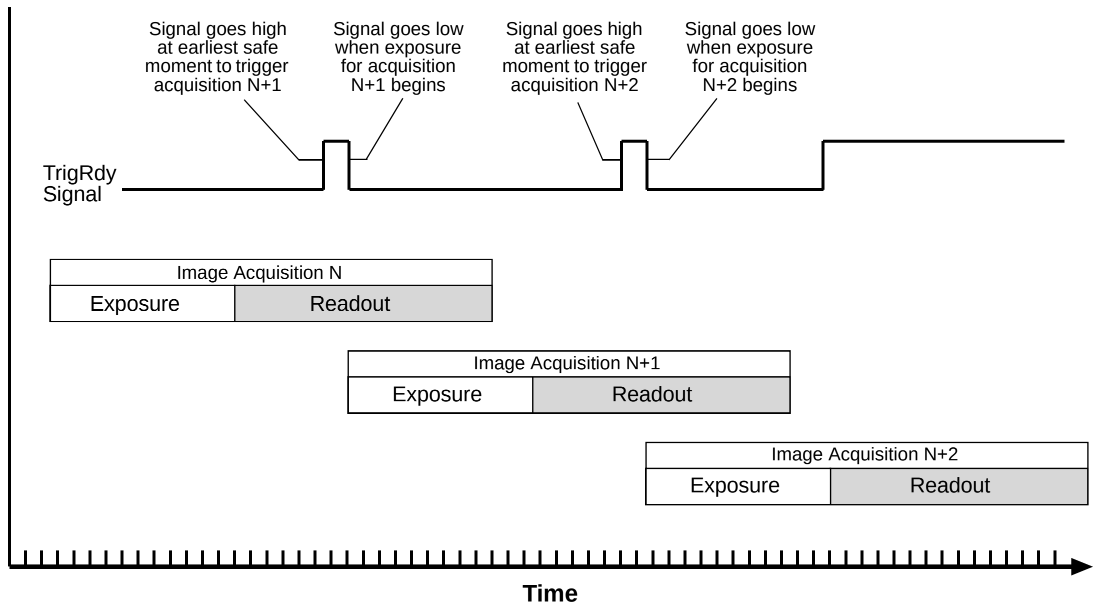


Fig. 51: Trigger Ready Signal

You should be aware that if the Acquisition Frame Rate Abs parameter is enabled, the operation of the trigger ready signal will be influenced by the value of the parameter:

- If the value of the parameter is greater than zero but less than the maximum allowed, the trigger ready will go high at the rate specified by the parameter value. For example, if the parameter is set to 10, the trigger ready signal will go high 10 times per second.
- If the value of the parameter is greater than the maximum allowed acquisition frame rate with the current camera settings, the trigger ready signal will work as described above and will go high at a point that represents the maximum acquisition frame rate allowed.

Note

If you attempt to start an image acquisition when the trigger ready signal is low, the camera will simply ignore the attempt.

The trigger ready signal will only be available when hardware triggering is enabled.

By default, the trigger ready signal is assigned to physical output line 2 on the camera. However, the assignment of the trigger signal to a physical output line can be changed.

For more information about changing the assignment of camera output signals to physical output lines, see Section 12.2.1 on page 191.

For more information about the electrical characteristics of the camera's output lines, see Section 9.7.2 on page 121.

10.7.2 Trigger Ready Signal (scA750-60 only)

Note

The information in this section only applies to scA750-60 gm/gc cameras. For information about the other camera models, see Section 10.7.1 on page 145.

As described in an earlier section, on these cameras the exposure for an image acquisition must not begin until readout of the previously acquired image has ended. The camera supplies a “Trigger Ready” (TrigRdy) output signal you can use to ensure that these conditions are met when you are using a hardware trigger signal to trigger image acquisition. When you are acquiring images, the camera automatically calculates the earliest moment that it is safe to trigger each new acquisition. The trigger ready signal will go high when it is safe to trigger an acquisition, will go low when the acquisition has started, and will go high again when it is safe to trigger the next acquisition (see Figure 51). The camera calculates the rise of the trigger ready signal based on the current exposure time parameter setting, the current size of the area of interest, and the time it will take to readout the captured pixel values from the sensor.

The trigger ready signal is especially useful if you want to run the camera at the maximum acquisition frame capture rate for the current conditions. If you monitor the trigger ready signal and you begin acquisition of each new image immediately after the signal goes high, you will be sure that the camera is operating at the maximum acquisition frame rate for the current conditions.

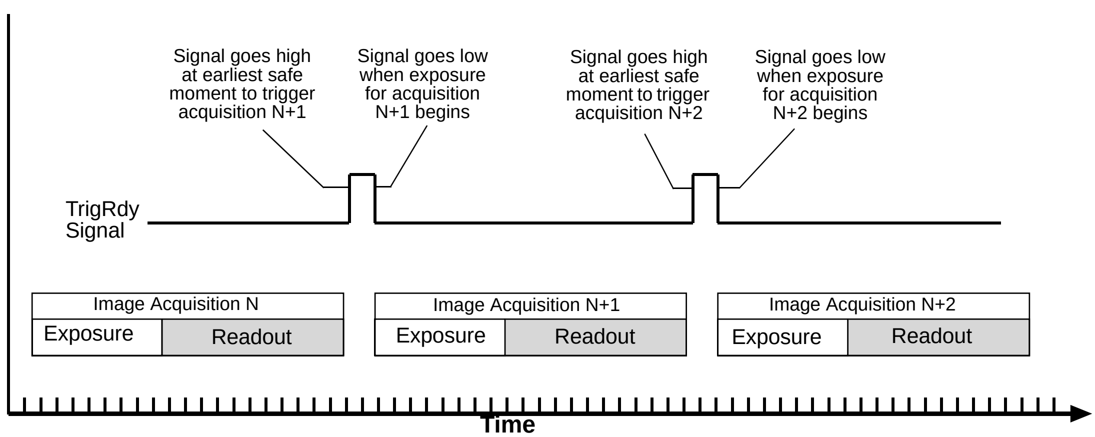


Fig. 52: Trigger Ready Signal

You should be aware that if the Acquisition Frame Rate Abs parameter is enabled, the operation of the trigger ready signal will be influenced by the value of the parameter:

- If the value of the parameter is greater than zero but less than the maximum allowed, the trigger ready will go high at the rate specified by the parameter value. For example, if the parameter is set to 10, the trigger ready signal will go high 10 times per second.
- If the value of the parameter is greater than the maximum allowed acquisition frame rate with the current camera settings, the trigger ready signal will work as described above and will go high at a point that represents the maximum acquisition frame rate allowed.

Note

If you attempt to start an image acquisition when the trigger ready signal is low, the camera will simply ignore the attempt.

The trigger ready signal will only be available when hardware triggering is enabled.

By default, the trigger ready signal is assigned to physical output line 2 on the camera. However, the assignment of the trigger signal to a physical output line can be changed.

For more information about changing the assignment of camera output signals to physical output lines, see Section 12.2.1 on page 191.

For more information about the electrical characteristics of the camera's output lines, see Section 9.7.2 on page 121.

10.8 Exposure Active Signal

The camera's "exposure active" (ExpAc) signal goes high when the exposure time for each image acquisition begins and goes low when the exposure time ends as shown in Figure 53. This signal can be used as a flash trigger and is also useful when you are operating a system where either the camera or the object being imaged is movable. For example, assume that the camera is mounted on an arm mechanism and that the mechanism can move the camera to view different portions of a product assembly. Typically, you do not want the camera to move during exposure. In this case, you can monitor the ExpAc signal to know when exposure is taking place and thus know when to avoid moving the camera.

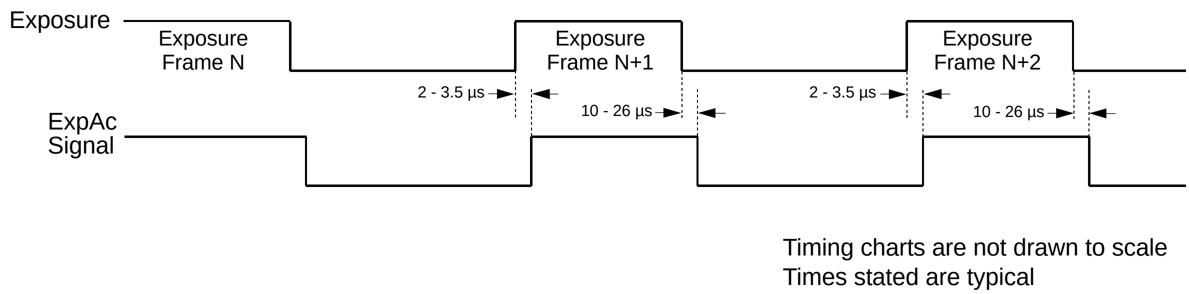


Fig. 53: Exposure Active Signal

Note

When you use the exposure active signal, be aware that there is a delay in the rise and the fall of the signal in relation to the start and the end of exposure. See Figure 53 for details.

By default, the ExpAc signal is assigned to physical output line 1 on the camera. However, the assignment of the ExpAc signal to a physical output line can be changed.

For more information about changing the assignment of camera output signals to physical output lines, see Section 12.2.1 on page 191.

For more information about the electrical characteristics of the camera's output lines, see Section 9.7.2 on page 121.

10.9 Acquisition Timing Chart

Figure 54 shows a timing chart for image acquisition and transmission. The chart assumes that exposure is triggered by an ExTrig signal with rising edge activation and that the camera is set for the timed exposure mode.

As Figure 54 shows, there is a slight delay between the rise of the ExTrig signal and the start of exposure. After the exposure time for an image acquisition is complete, the camera begins reading out the acquired image data from the CCD sensor into a buffer in the camera. When the camera has determined that a sufficient amount of image data has accumulated in the buffer, it will begin transmitting the data from the camera to the host PC.

This buffering technique avoids the need to exactly synchronize the clock used for sensor readout with the data transmission over your Ethernet network. The camera will begin transmitting data when it has determined that it can safely do so without over-running or under-running the buffer. This buffering technique is also an important element in achieving the highest possible frame rate with the best image quality.

The **exposure start delay** is the amount of time between the point where the trigger signal transitions and the point where exposure actually begins.

The **frame readout time** is the amount of time it takes to read out the data for an acquired image from the CCD sensor into the image buffer.

The **frame transmission time** is the amount of time it takes to transmit the acquired image from the buffer in the camera to the host PC via the network.

The **transmission start delay** is the amount of time between the point where the camera begins reading out the acquired image data from the sensor to the point where it begins transmitting the data for the acquired image from the buffer to the host PC.

The exposure start delay varies from camera model to camera model. The table below shows the exposure start delay for each camera model:

Camera Model	Exposure Start Delay
scA640-70gm/gc	31.72 µs
scA640-74gm/gc	28.19 µs
scA750-60 gm/gc	180.0 µs
scA780-54gm/gc	33.44 µs
scA1000-20gm/gc	68.06 µs
scA1000-30gm/gc	43.87 µs
scA1390-17gm/gc	59.04 µs
scA1400-17gm/gc	58.08 µs
scA1600-14gm/gc	58.92 µs

Table 7: Exposure Start Delays

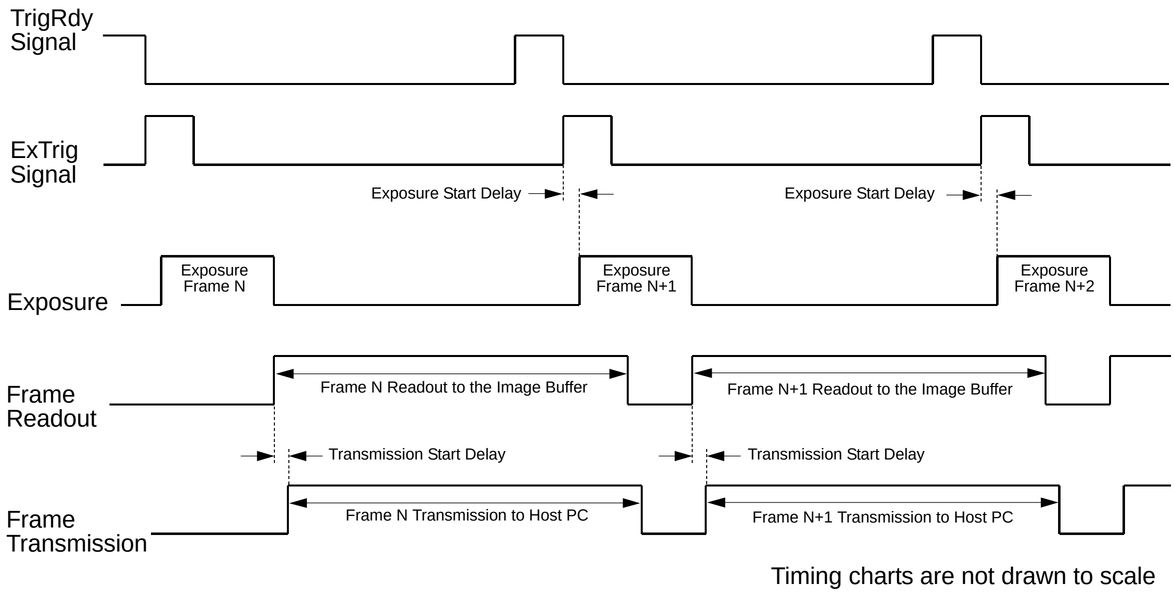


Fig. 54: Exposure Start Controlled with an ExTrig Signal

You can calculate the frame readout time by using this formula:

$$\text{Frame Readout Time} = (\text{AOI height} \times C_1) + C_2$$

Where the values for the constants C_1 and C_2 are from the table in Section 10.10 on page 152 for all camera models except the scA750-60 or from the table in Section 10.11 on page 155 for scA750-60 cameras.

For more information about the AOI height, see Section 13.5 on page 209.

You can calculate an approximate frame transmission time by using this formula:

$$\sim \text{Frame Transmission Time} = \frac{\text{Payload Size Parameter Value}}{\text{Device Current Throughput Parameter Value}}$$

Note that this is an approximate frame transmission time. Due to the nature of the Ethernet network, the transmission time could vary. Also note that the frame transmission cannot be less than the frame readout time. So if the frame transmission time formula returns a value that is less than the readout time, the approximate frame transmission time will be equal to the readout time.

Due to the nature of the Ethernet network, the transmission start delay can vary from frame to frame. The start delay, however, is of very low significance when compared to the transmission time.

For more information about the Payload Size and Device Current Throughput parameters, see Section 7.1 on page 97.

10.10 Maximum Allowed Acquisition Frame Rate (all models except scA750-60)

Note

The information in this section applies to all camera models except the scA750-60 gm/gc. For information about scA750-60 cameras, see Section 10.11 on page 155.

In general, the maximum allowed acquisition frame rate can be limited by three factors:

- The amount of time it takes to read an acquired image out of the imaging sensor and into the camera's frame buffer (an acquired image is also known as a frame). This time varies depending on the height of the frame. Shorter frames take less time to read out of the sensor. The frame height is determined by the camera's AOI Height settings.
- The exposure time for acquired frames. If you use very long exposure times, you can acquire fewer frames per second.
- The amount of time that it takes to transmit an acquired frame from the camera to your host PC. The amount of time needed to transmit a frame depends on the bandwidth assigned to the camera.

To determine the maximum allowed acquisition frame rate with your current camera settings, you can read the value of the camera's Resulting Frame Rate parameter. This parameter indicates the camera's current maximum allowed frame rate taking the AOI, exposure time, and bandwidth settings into account.

For more information about AOI Height settings, see Section 13.5 on page 209.

For more information about the Resulting Frame Rate parameter, see Section 7.1 on page 97.

Increasing the Maximum Allowed Frame Rate

You may find that you would like to acquire frames at a rate higher than the maximum allowed with the camera's current settings. In this case, you must first use the three formulas described below to determine what factor is restricting the maximum frame rate the most. Next, you must try to make that factor less restrictive:

- You will often find that the sensor readout time is most restrictive factor. Decreasing the AOI height for the acquired frames will decrease the sensor readout time and will make this factor less restrictive.
- If you are using normal exposure times and you are using the camera at it's maximum resolution, your exposure time will not normally be the most restrictive factor on the frame rate. However, if you are using long exposure times or small areas of interest, it is quite possible to find that your exposure time is the most restrictive factor on the frame rate. In this case, you

should lower your exposure time. (You may need to compensate for a lower exposure time by using a brighter light source or increasing the opening of your lens aperture.)

- The frame transmission time will not normally be a restricting factor. But if you are using multiple cameras and you have set a small packet size or a large inter-packet delay, you may find that the transmission time is restricting the maximum allowed rate. In this case, you could increase the packet size or decrease the inter-packet delay. If you are using several cameras connected to the host PC via a network switch, you could also use a multiport network adapter in the PC instead of a switch. This would allow you to increase the Ethernet bandwidth assigned to the camera and thus decrease the transmission time.

For more information about AOI settings, see Section 13.5 on page 209.

For more information on the settings that determine the bandwidth assigned to the camera, see Section 7.2 on page 104.

Formula 1 calculates the maximum frame rate based on the sensor readout time:

$$\text{Max. Frames/s} = \frac{1}{(\text{AOI Height} \times C_1) + C_2}$$

Where:

AOI Height = the height of the acquired frames as determined by the AOI Height settings.

The constants C_1 and C_2 depend on the camera model as shown in the table below:

	scA640-70 gm/gc	scA640-74 gm/gc	scA780-54 gm/gc	scA1000-20 gm/gc	scA1000-30 gm/gc	scA1390-17 gm/gc	scA1400-17 gm/gc	scA1600-14 gm/gc
C_1	25.49 µs	22.66 µs	26.94 µs	53.62 µs	35.39 µs	46.53 µs	46.53 µs	50.99 µs
C_2	1641 µs	1459 µs	2339 µs	8202 µs	4548 µs	10301 µs	9413 µs	6715 µs

Formula 2 calculates the maximum frame rate based on the exposure time for the captured frames:

$$\text{Max. Frames/s} = \frac{1}{\text{Exposure time in µs} + C_3}$$

Where the constant C_3 depends on the camera model as shown in the table below:

	scA640-70 gm/gc	scA640-74 gm/gc	scA780-54 gm/gc	scA1000-20 gm/gc	scA1000-30 gm/gc	scA1390-17 gm/gc	scA1400-17 gm/gc	scA1600-14 gm/gc
C_3	95.62 µs	84.99 µs	100.35 µs	204.29 µs	132.10 µs	177.15 µs	175.10 µs	176.86 µs

For more information about setting the exposure time, see Section 10.4 on page 139.

Formula 3 calculates the maximum frame rate based on the frame transmission time:

$$\text{Max. Frames/s} = \frac{\text{Device Current Throughput Parameter Value}}{\text{Payload Size Parameter Value}}$$

Example

Assume that you are using an scA640-70gm camera set for an exposure time of 2000 µs and for 600 x 400 resolution. Also assume that you have checked the value of the Device Current Throughput parameter and the Payload Size parameters and found them to be 110000000 and 240000 respectively.

Formula 1:

$$\text{Max Frames/s} = \frac{1}{(400 \times 25.49\ \mu s) + 1641\ \mu s}$$

Max Frames/s = 85.4 frames/s

Formula 2:

$$\text{Max Frames/s} = \frac{1}{2000\ \mu s + 95.62\ \mu s}$$

Max Frames/s = 477.2 frames/s

Formula 3:

$$\text{Max Frames/s} = \frac{110000000}{240000}$$

Max Frames/s = 458.3 frames/s

Formula one returns the lowest value. So in this case, the limiting factor is the sensor readout time, and the maximum allowed acquisition frame rate would be 85.4 frames per second.

10.11 Maximum Allowed Acquisition Frame Rate (scA750-60 only)

Note

The information in this section only applies to scA750-60 gm/gc cameras. For information about the other camera models, see Section 10.10 on page 152.

In general, the maximum allowed acquisition frame rate can be limited by two factors:

- The sum of the exposure time plus the amount of time it takes to read the acquired image out of the imaging sensor and into the camera's frame buffer. (An acquired image is also known as a frame.)

  The exposure time is set by the user. If you use very long exposure times, you can acquire fewer frames per second.

  The readout time varies depending on the height of the frame. Shorter frames take less time to read out of the sensor. The frame height is determined by the camera's AOI Height settings.
- The amount of time that it takes to transmit an acquired frame from the camera to your host PC. The amount of time needed to transmit a frame depends on the bandwidth assigned to the camera.

To determine the maximum allowed acquisition frame rate with your current camera settings, you can read the value of the camera's Resulting Frame Rate parameter. This parameter indicates the camera's current maximum allowed frame rate taking the AOI, exposure time, and bandwidth settings into account.

For more information about AOI settings, see Section 13.5 on page 209.

For more information about the Resulting Frame Rate parameter, see Section 7.1 on page 97.

Increasing the Maximum Allowed Frame Rate

You may find that you would like to acquire frames at a rate higher than the maximum allowed with the camera's current settings. In this case, you must first use the two formulas described below to determine what factor is restricting the maximum frame rate the most. Next, you must try to make that factor less restrictive:

- You will often find that the sum of the exposure time plus the sensor readout time is the most restrictive factor.

  Decreasing the AOI height for the acquired frames will decrease the sensor readout time and will make this factor less restrictive.

  If you are using long exposure times, it is quite possible to find that your exposure time is making this factor the most restrictive. In this case, you should lower your exposure time. (You may need to compensate for a lower exposure time by using a brighter light source or increasing the opening of your lens aperture.)

- The frame transmission time will not normally be a restricting factor. But if you are using multiple cameras and you have set a small packet size or a large inter-packet delay, you may find that the transmission time is restricting the maximum allowed rate. In this case, you could increase the packet size or decrease the inter-packet delay. If you are using several cameras connected to the host PC via a network switch, you could also use a multiport network adapter in the PC instead of a switch. This would allow you to increase the Ethernet bandwidth assigned to the camera and thus decrease the transmission time.

For more information about AOI settings, see Section 13.5 on page 209.

For more information on the settings that determine the bandwidth assigned to the camera, see Section 7.2 on page 104.

Formula 1 calculates the maximum frame rate based on the sum of the exposure time plus the sensor readout time:

$$\text{Max. Frames/s} = \frac{1}{\text{Exposure Time in µs} + (\text{AOI Height} \times C_1) + C_2}$$

Where:

AOI Height = the height of the acquired frames as determined by the AOI Height settings.

The constants C_1 and C_2 depend on the camera model as shown in the table below:

	scA750-60 gm/gc
C_1	31.0 µs
C_2	397.0 µs

For more information about setting the exposure time, see Section 10.4 on page 139.

Formula 2 calculates the maximum frame rate based on the frame transmission time:

$$\text{Max. Frames/s} = \frac{\text{Device Current Throughput Parameter Value}}{\text{Payload Size Parameter Value}}$$

Assume that you are using a monochrome scA750-60 camera set for an exposure time of 2000 µs and for 600 x 400 resolution. Also assume that you have checked the value of the Device Current Throughput parameter and the Payload Size parameters and found them to be 110000000 and 240000 respectively.

Formula 1:

$$\text{Max. Frames/s} = \frac{1}{2000\ \text{µs} + (400 \times 31.0\ \text{µs}) + 397.0\ \text{µs}}$$

Max Frames/s = 67.6 frames/s

Formula 2:

$$\text{Max Frames/s} = \frac{110000000}{240000}$$

Max Frames/s = 458.3 frames/s

Formula one returns the lowest value. So in this case, the limiting factor is the sum of the exposure time plus the sensor readout time and the maximum allowed acquisition frame rate would be 67.6 frames per second.

11 Pixel Data Formats

By selecting a pixel data format, you determine the format (layout) of the image data transmitted by the camera. This section provides detailed information about the available pixel data formats.

11.1 Setting the Pixel Data Format

The setting for the camera’s Pixel Format parameter determines the format of the pixel data that will be output from the camera. The available pixel formats depend on the camera model and whether the camera is monochrome or color. Table 8 lists the pixel formats available on each monochrome camera model and Table 9 lists the pixel formats available on each color camera model.

Mono Camera Model	**Mono 8**	**Mono 16**	**Mono 12 Packed**	**YUV 4:2:2 Packed**	**YUV 4:2:2 (YUYV) Packed**
scA640-70	●	●	●	●	●
scA640-74	●	●	●	●	●
scA750-60	●			●	●
scA780-54	●	●	●	●	●
scA1000-20	●	●	●	●	●
scA1000-30	●	●	●	●	●
scA1390-17	●	●	●	●	●
scA1400-17	●	●	●	●	●
scA1600-14	●	●	●	●	●

Table 8: Pixel Formats Available on Monochrome Cameras (• = format available)

Color Camera Model	Mono 8	Bayer RG 8	Bayer BG 8	Bayer BG 16	Bayer BG 12 Packed	YUV 4:2:2 Packed	YUV 4:2:2 (YUYV) Packed
scA640-70	•		•	•	•	•	•
scA640-74	•		•	•	•	•	•
scA750-60	•	•				•	•
scA780-54	•		•	•	•	•	•
scA1000-20	•		•	•	•	•	•
scA1000-30	•		•	•	•	•	•
scA1390-17	•		•	•	•	•	•
scA1400-17	•		•	•	•	•	•
scA1600-14	•		•	•	•	•	•

Table 9: Pixel Formats Available on Color Cameras (• = format available)

Details of the monochrome formats are described in Section 11.2 on page 161 and details of the color formats are described in Section 11.3 on page 169.

You can set the Pixel Format parameter value from within your application software by using the pylon API. The following code snippet illustrates using the API to set the parameter value:

```
Camera.PixelFormat.SetValue( PixelFormat_Mono8 );
Camera.PixelFormat.SetValue( PixelFormat_Mono12Packed );
Camera.PixelFormat.SetValue( PixelFormat_Mono16 );
Camera.PixelFormat.SetValue( PixelFormat_YUV422Packed );
Camera.PixelFormat.SetValue( PixelFormat_YUV422_YUYV_Packed );
Camera.PixelFormat.SetValue( PixelFormat_BayerBG8 );
Camera.PixelFormat.SetValue( PixelFormat_BayerBG16 );
```

For detailed information about using the pylon API, refer to the Basler pylon Programmer's Guide and API Reference.

You can also use the Basler pylon Viewer application to easily set the parameters.

For more information about the pylon Viewer, see Section 2.2 on page 32 and Section 2.4 on page 52.

11.2 Pixel Data Formats for Mono Cameras

11.2.1 Mono 8 Format (equivalent to DCAM Mono 8)

When a monochrome camera is set for the Mono 8 pixel data format, it outputs 8 bits of brightness data per pixel.

Note

The scA750-60 outputs 10 bits of brightness data per pixel with 8 bits effective. The 8 bits of effective pixel data fill from the least significant bit. The two unused most significant bits are filled with zero.

The table below describes how the pixel data for a received frame will be ordered in the image buffer in your PC when the camera is set for Mono8 output.

The following standards are used in the table:

P_0 = the first pixel transmitted by the camera

P_n = the last pixel transmitted by the camera

B_0 = the first byte in the buffer

B_m = the last byte in the buffer

Byte	Data
B_0	Brightness value for P_0
B_1	Brightness value for P_1
B_2	Brightness value for P_2
B_3	Brightness value for P_3
B_4	Brightness value for P_4
•	•
•	•

Byte	Data
•	•
•	•
B_{m-4}	Brightness value for P_{n-4}
B_{m-3}	Brightness value for P_{n-3}
B_{m-2}	Brightness value for P_{n-2}
B_{m-1}	Brightness value for P_{n-1}
B_m	Brightness value for P_n

With the camera set for Mono8, the pixel data output is 8 bit data of the “unsigned char” type. The available range of data values and the corresponding indicated signal levels are as shown in the table below.

This Data Value (Hexadecimal)	Indicates This Signal Level (Decimal)
0xFF	255
0xFE	254
•	•
•	•
•	•
0x01	1
0x00	0

11.2.2 Mono 16 Format (equivalent to DCAM Mono 16)

When a monochrome camera is set for the Mono16 pixel data format, it outputs 16 bits of brightness data per pixel with 12 bits effective. The 12 bits of effective pixel data fill from the least significant bit. The four unused most significant bits are filled with zeros.

The table below describes how the pixel data for a received frame will be ordered in the image buffer in your PC when the camera is set for Mono16 output. Note that the data is placed in the image buffer in **little endian format**.

The following standards are used in the table:

P_0 = the first pixel transmitted by the camera

P_n = the last pixel transmitted by the camera

B_0 = the first byte in the buffer

B_m = the last byte in the buffer

Byte	Data
B_0	Low byte of brightness value for P_0
B_1	High byte of brightness value for P_0
B_2	Low byte of brightness value for P_1
B_3	High byte of brightness value for P_1
B_4	Low byte of brightness value for P_2
B_5	High byte of brightness value for P_2
B_6	Low byte of brightness value for P_3
B_7	High byte of brightness value for P_3
B_8	Low byte of brightness value for P_4
B_9	High byte of brightness value for P_4
•	•
•	•
•	•
B_{m-7}	Low byte of brightness value for P_{n-3}
B_{m-6}	High byte of brightness value for P_{n-3}
B_{m-5}	Low byte of brightness value for P_{n-2}
B_{m-4}	High byte of brightness value for P_{n-2}
B_{m-3}	Low byte of brightness value for P_{n-1}
B_{m-2}	High byte of brightness value for P_{n-1}
B_{m-1}	Low byte of brightness value for P_n
B_m	High byte of brightness value for P_n

When the camera is set for Mono 16, the pixel data output is 16 bit data of the "unsigned short (little endian)" type. The available range of data values and the corresponding indicated signal levels are as shown in the table below. Note that for 16 bit data, you might expect a value range from 0x0000 to 0xFFFF. However, with the camera set for Mono16 only 12 bits of the 16 bits transmitted are effective. Therefore, the highest data value you will see is 0x0FFF indicating a signal level of 4095.

This Data Value (Hexadecimal)	Indicates This Signal Level (Decimal)
0x0FFF	4095
0x0FFE	4094
•	•
•	•
•	•
0x0001	1
0x0000	0

Note

When a camera that is set for Mono 16 has only 12 bits effective, the leader of transmitted frames will indicate Mono 12 as the pixel format.

11.2.3 Mono 12 Packed Format

When a monochrome camera is set for the Mono 12 Packed pixel data format, it outputs 12 bits of brightness data per pixel. Every three bytes transmitted by the camera contain data for two pixels.

The table below describes how the pixel data for a received frame will be ordered in the image buffer in your PC when the camera is set for Mono 12 Packed output.

The following standards are used in the table:

P_0 = the first pixel transmitted by the camera

P_n = the last pixel transmitted by the camera

B_0 = the first byte in the buffer

B_m = the last byte in the buffer

Byte	Data	
B_0	P_0 bits 11 ... 4	
B_1	P_1 bits 3 ... 0	P_0 bits 3 ... 0
B_2	P_1 bits 11 ... 4	
B_3	P_2 bits 11 ... 4	
B_4	P_3 bits 3 ... 0	P_2 bits 3 ... 0
B_5	P_3 bits 11 ... 4	
B_6	P_4 bits 11 ... 4	
B_7	P_5 bits 3 ... 0	P_4 bits 3 ... 0
B_8	P_5 bits 11 ... 4	
B_9	P_6 bits 11 ... 4	
B_{10}	P_7 bits 3 ... 0	P_6 bits 3 ... 0
B_{11}	P_7 bits 11 ... 4	
•	•	
•	•	•
•	•	
B_{m-5}	P_{n-3} bits 11 ... 4	
B_{m-4}	P_{n-2} bits 3 ... 0	P_{n-3} bits 3 ... 0
B_{m-3}	P_{n-2} bits 11 ... 4	
B_{m-2}	P_{n-1} bits 11 ... 4	
B_{m-1}	P_n bits 3 ... 0	P_{n-1} bits 3 ... 0
B_m	P_n bits 11 ... 4	

When a monochrome camera is set for Mono 12 Packed, the pixel data output is 12 bit data of the "unsigned" type. The available range of data values and the corresponding indicated signal levels are as shown in the table below.

This Data Value (Hexadecimal)	Indicates This Signal Level (Decimal)
0x0FFF	4095
0x0FFE	4094
•	•
•	•
•	•
0x0001	1
0x0000	0

11.2.4 YUV 4:2:2 Packed Format (equivalent to DCAM YUV 4:2:2)

When a monochrome camera is set for the YUV 4:2:2 Packed pixel data format, the camera transmits Y, U, and V values in a fashion that mimics the output from a color camera set for YUV 4:2:2 Packed.

The Y value transmitted for each pixel is an actual 8 bit brightness value similar to the pixel data transmitted when a monochrome camera is set for Mono 8. The U and V values transmitted will always be zero. With this color coding, a Y value is transmitted for each pixel, but the U and V values are only transmitted for every second pixel.

Note

The scA750-60 outputs 10 bits of brightness data per pixel with 8 bits effective. The 8 bits of effective pixel data fill from the least significant bit. The two unused most significant bits are filled with zero.

The order of the pixel data for a received frame in the image buffer in your PC is similar to the order of YUV 4:2:2 Packed output from a color camera.

For more information about the YUV 4:2:2 Packed format on color cameras, see Section 11.3.6 on page 179.

11.2.5 YUV 4:2:2 (YUYV) Packed Format

When a monochrome camera is set for the YUV 4:2:2 (YUYV) Packed pixel data format, the camera transmits Y, U, and V values in a fashion that mimics the output from a color camera set for YUV 4:2:2 (YUYV) Packed.

The Y value transmitted for each pixel is an actual 8 bit brightness value similar to the pixel data transmitted when a monochrome camera is set for Mono 8. The U and V values transmitted will always be zero. With this color coding, a Y value is transmitted for each pixel, but the U and V values are only transmitted for every second pixel.

Note

The scA750-60 outputs 10 bits of brightness data per pixel with 8 bits effective. The 8 bits of effective pixel data fill from the least significant bit. The two unused most significant bits are filled with zero.

The order of the pixel data for a received frame in the image buffer in your PC is similar to the order of YUV 4:2:2 (YUYV) Packed output from a color camera.

For more information about the YUV 4:2:2 (YUYV) Packed format on color cameras, see Section 11.3.7 on page 182.

11.3 Pixel Data Output Formats for Color Cameras

11.3.1 The Bayer Color Filter

The sensor used in color models of the camera is equipped with an additive color separation filter known as a Bayer filter. The pixel data output formats available on color cameras are related to the Bayer pattern, so you need a basic knowledge of the Bayer filter to understand the pixel formats. With the Bayer filter, each individual pixel is covered by a micro-lens that allows light of only one color to strike the pixel. The pattern of the Bayer filter used on the camera is as shown in Figure 55 (the alignment of the Bayer filter with repect to the sensor is shown as an example only). As the figure illustrates, within each square of four pixels, one pixel sees only red light, one sees only blue light, and two pixels see only green light. (This combination mimics the human eye's sensitivity to color.)

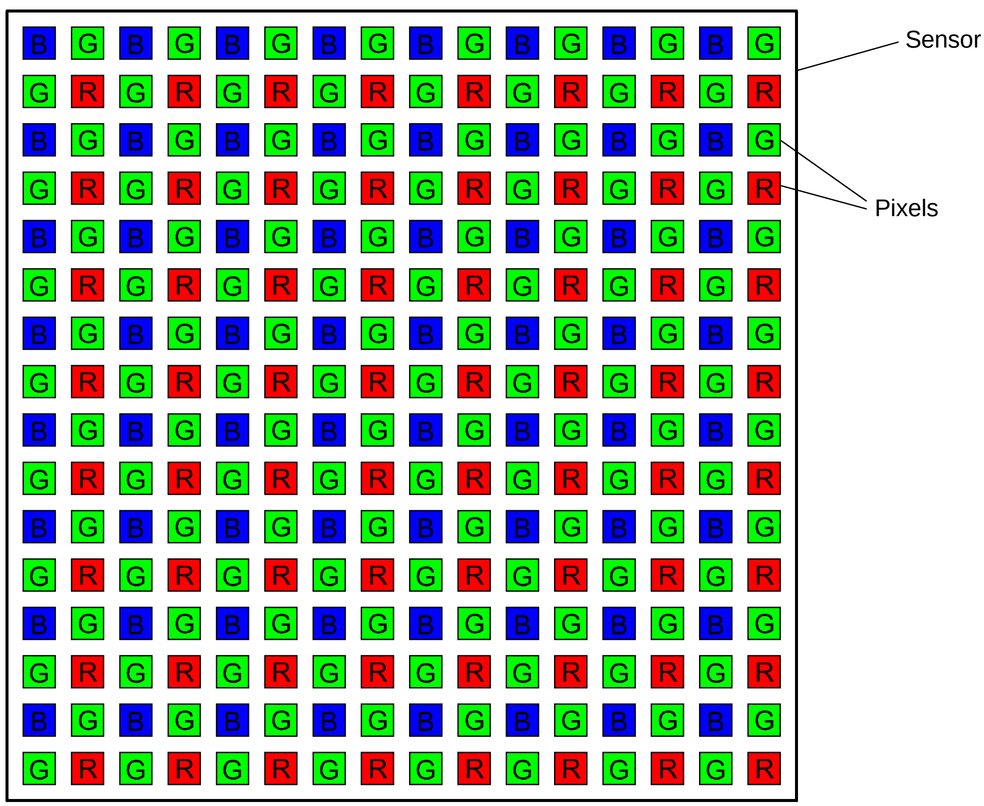


Fig. 55: Bayer Filter Pattern

11.3.1.1 Color Filter Alignment

The alignment of the Bayer filter to the pixels in the images acquired by color cameras is either Bayer BG or Bayer RG depending on the camera model. Table 10 shows the filter alignment for each available camera model.

Color Camera Model	Filter Alignment
scA640-70	BG
scA640-74	BG
scA750-60	RG
scA780-54	BG
scA1000-20	BG
scA1000-30	BG
scA1390-17	BG
scA1400-17	BG
scA1600-14	BG

Table 10: Bayer Filter to Sensor Alignment

Bayer BG alignment means that pixel one and pixel two of the first line in each image transmitted will be blue and green respectively. And for the second line transmitted, pixel one and pixel two will be green and red respectively. Since the pattern of the Bayer filter is fixed, you can use this information to determine the color of all of the other pixels in the image.

Bayer RG alignment means that pixel one and pixel two of the first line in each image transmitted will be red and green respectively. And for the second line transmitted, pixel one and pixel two will be green and blue respectively. Since the pattern of the Bayer filter is fixed, you can use this information to determine the color of all of the other pixels in the image.

Because the size and position of the area of interest on color cameras must be adjusted in increments of 2, the color filter alignment will remain the same regardless of the camera's area of interest (AOI) settings.

The Pixel Color Filter parameter indicates the current alignment of the camera's Bayer filter to the pixels in the images captured by a color camera. You can tell how the current AOI is aligned to the Bayer filter by reading the value of the Pixel Color Filter parameter.

For more information about the camera's AOI feature, see Section 13.5 on page 209.

11.3.2 Bayer BG 8 Format (equivalent to DCAM Raw 8)

When a color camera is set for the Bayer BG 8 pixel data format, it outputs 8 bits of data per pixel and the pixel data is not processed or interpolated in any way. So, for each pixel covered with a red lens, you get 8 bits of red data. For each pixel covered with a green lens, you get 8 bits of green data. And for each pixel covered with a blue lens, you get 8 bits of blue data. (This type of pixel data is sometimes referred to as "raw" output.)

The "BG" in the name Bayer BG 8 refers to the alignment of the colors in the Bayer filter to the pixels in the acquired images. For even lines in the images, pixel one will be blue, pixel two will be green, pixel three will be blue, pixel four will be green, etc. For odd lines in the images, pixel one will be green, pixel two will be red, pixel three will be green, pixel four will be red, etc.

For more information about the Bayer filter, see Section 11.3.1 on page 169.

The tables below describe how the data for the even lines and for the odd lines of a received frame will be ordered in the image buffer in your PC when the camera is set for Bayer BG 8 output.

The following standards are used in the tables:

P_0 = the first pixel transmitted by the camera for a line

P_n = the last pixel transmitted by the camera a line

B_0 = the first byte of data for a line

B_m = the last byte of data for a line

Even Lines	
Byte	**Data**
B_0	Blue value for P_0
B_1	Green value for P_1
B_2	Blue value for P_2
B_3	Green value for P_3
B_4	Blue value for P_4
B_5	Green value for P_5
2	•
2	•
2	•
B_{m-5}	Blue value for P_{n-5}
B_{m-4}	Green value for P_{n-4}
B_{m-3}	Blue value for P_{n-3}
B_{m-2}	Green value for P_{n-2}
B_{m-1}	Blue value for P_{n-1}
B_m	Green value for P_n

Odd Lines	
Byte	**Data**
B_0	Green value for P_0
B_1	Red value for P_1
B_2	Green value for P_2
B_3	Red value for P_3
B_4	Green value for P_4
B_5	Red value for P_5
2	•
2	•
2	•
B_{m-5}	Green value for P_{n-5}
B_{m-4}	Red value for P_{n-4}
B_{m-3}	Green value for P_{n-3}
B_{m-2}	Red value for P_{n-2}
B_{m-1}	Green value for P_{n-1}
B_m	Red value for P_n

With the camera set for Bayer BG 8, the pixel data output is 8 bit data of the “unsigned char” type. The available range of data values and the corresponding indicated signal levels are as shown in the table below.

This Data Value (Hexadecimal)	**Indicates This Signal Level (Decimal)**
0xFF	255
0xFE	254
•	•
•	•
•	•
0x01	1
0x00	0

11.3.3 Bayer RG 8 Format (equivalent to DCAM Raw 8)

When a color camera is set for the Bayer RG 8 pixel data format, it outputs 8 bits of data per pixel and the pixel data is not processed or interpolated in any way. So, for each pixel covered with a red lens, you get 8 bits of red data. For each pixel covered with a green lens, you get 8 bits of green data. And for each pixel covered with a blue lens, you get 8 bits of blue data. (This type of pixel data is sometimes referred to as "raw" output.)

Note

The scA750-60 outputs 10 bits of brightness data per pixel with 8 bits effective. The 8 bits of effective pixel data fill from the least significant bit. The two unused most significant bits are filled with zero.

The "RG" in the name Bayer RG 8 refers to the alignment of the colors in the Bayer filter to the pixels in the acquired images. For even lines in the images, pixel one will be red, pixel two will be green, pixel three will be red, pixel four will be green, etc. For odd lines in the images, pixel one will be green, pixel two will be blue, pixel three will be green, pixel four will be blue, etc.

For more information about the Bayer filter, see Section 11.3.1 on page 169.

The tables below describe how the data for the even lines and for the odd lines of a received frame will be ordered in the image buffer in your PC when the camera is set for Bayer RG 8 output.

The following standards are used in the tables:

P_0 = the first pixel transmitted by the camera for a line

P_n = the last pixel transmitted by the camera a line

B_0 = the first byte of data for a line

B_m = the last byte of data for a line

Even Lines	
Byte	**Data**
B_0	Red value for P_0
B_1	Green value for P_1
B_2	Red value for P_2
B_3	Green value for P_3
B_4	Red value for P_4
B_5	Green value for P_5
2	•
2	•
2	•

Odd Lines	
Byte	**Data**
B_0	Green value for P_0
B_1	Blue value for P_1
B_2	Green value for P_2
B_3	Blue value for P_3
B_4	Green value for P_4
B_5	Blue value for P_5
2	•
2	•
2	•

B_{m-5}	Red value for P_{n-5}
B_{m-4}	Green value for P_{n-4}
B_{m-3}	Red value for P_{n-3}
B_{m-2}	Green value for P_{n-2}
B_{m-1}	Red value for P_{n-1}
B_m	Green value for P_n

B_{m-5}	Green value for P_{n-5}
B_{m-4}	Blue value for P_{n-4}
B_{m-3}	Green value for P_{n-3}
B_{m-2}	Blue value for P_{n-2}
B_{m-1}	Green value for P_{n-1}
B_m	Blue value for P_n

With the camera set for Bayer RG8, the pixel data output is 8 bit data of the “unsigned char” type. The available range of data values and the corresponding indicated signal levels are as shown in the table below.

This Data Value (Hexadecimal)	**Indicates This Signal Level (Decimal)**
0xFF	255
0xFE	254
•	•
•	•
•	•
0x01	1
0x00	0

11.3.4 Bayer BG 16 Format (equivalent to DCAM Raw 16)

When a color camera is set for the Bayer BG 16 pixel data format, it outputs 16 bits of data per pixel with 12 bits effective. The 12 bits of effective pixel data fill from the least significant bit. The four unused most significant bits are filled with zeros.

With the Bayer BG 16 the pixel data is not processed or interpolated in any way. So, for each pixel covered with a red lens, you get 12 effective bits of red data. For each pixel covered with a green lens, you get 12 effective bits of green data. And for each pixel covered with a blue lens, you get 12 effective bits of blue data. (This type of pixel data is sometimes referred to as "raw" output.)

The "BG" in the name Bayer BG 16 refers to the alignment of the colors in the Bayer filter to the pixels in the acquired images. For even lines in the images, pixel one will be blue, pixel two will be green, pixel three will be blue, pixel four will be green, etc. For odd lines in the images, pixel one will be green, pixel two will be red, pixel three will be green, pixel four will be red, etc.

For more information about the Bayer filter, see Section 11.3.1 on page 169.

The tables below describe how the data for the even lines and for the odd lines of a received frame will be ordered in the image buffer in your PC when the camera is set for Bayer BG 16 output. Note that the data is placed in the image buffer in **little endian format**.

The following standards are used in the tables:

P_0 = the first pixel transmitted by the camera for a line

P_n = the last pixel transmitted by the camera a line

B_0 = the first byte of data for a line

B_m = the last byte of data for a line

Even Lines	
Byte	**Data**
B_0	Low byte of blue value for P_0
B_1	High byte of blue value for P_0
B_2	Low byte of green value for P_1
B_3	High byte of green value for P_1
B_4	Low byte of blue value for P_2
B_5	High byte of blue value for P_2
B_6	Low byte of green value for P_3
B_7	High byte of green value for P_3
•	•
•	•
•	•
B_{m-7}	Low byte of blue value for P_{n-3}
B_{m-6}	High byte of blue value for P_{n-3}

Odd Lines	
Byte	**Data**
B_0	Low byte of green value for P_0
B_1	High byte of green value for P_0
B_2	Low byte of red value for P_1
B_3	High byte of red value for P_1
B_4	Low byte of green value for P_2
B_5	High byte of green value for P_2
B_6	Low byte of red value for P_3
B_7	High byte of red value for P_3
•	•
•	•
•	•
B_{m-7}	Low byte of green value for P_{n-3}
B_{m-6}	High byte of green value for P_{n-3}

B_{m-5}	Low byte of green value for P_{n-2}
B_{m-4}	High byte of green value for P_{n-2}
B_{m-3}	Low byte of blue value for P_{n-1}
B_{m-2}	High byte of blue value for P_{n-1}
B_{m-1}	Low byte of green value for P_n
B_m	High byte of green value for P_n

B_{m-5}	Low byte of red value for P_{n-2}
B_{m-4}	High byte of red value for P_{n-2}
B_{m-3}	Low byte of green value for P_{n-1}
B_{m-2}	High byte of green value for P_{n-1}
B_{m-1}	Low byte of red value for P_n
B_m	High byte of red value for P_n

When the camera is set for Bayer BG 16, the pixel data output is 16 bit data of the “unsigned short (little endian)” type. The available range of data values and the corresponding indicated signal levels are as shown in the table below. Note that for 16 bit data, you might expect a value range from 0x0000 to 0xFFFF. However, with the camera set for Bayer BG 16 only 12 bits of the 16 bits transmitted are effective. Therefore, the highest data value you will see is 0x0FFF indicating a signal level of 4095.

This Data Value (Hexadecimal)	**Indicates This Signal Level (Decimal)**
0x0FFF	4095
0x0FFE	4094
•	•
•	•
•	•
0x0001	1
0x0000	0

Note

When a camera that is set for Bayer BG 16 has only 12 bits effective, the leader of transmitted frames will indicate Bayer BG 12 as the as the pixel format.

11.3.5 Bayer BG 12 Packed Format

When a color camera is set for the Bayer BG 12 Packed pixel dataformat, it outputs 12 bits of data per pixel. Every three bytes transmitted by the camera contain data for two pixels.

With the Bayer BG 12 Packed coding, the pixel data is not processed or interpolated in any way. So, for each pixel covered with a red lens in the sensor's Bayer filter, you get 12 bits of red data. For each pixel covered with a green lens in the filter, you get 12 bits of green data. And for each pixel covered with a blue lens in the filter, you get 12 bits of blue data. (This type of pixel data is sometimes referred to as "raw" output.)

For more information about the Bayer filter, see Section 11.3.1 on page 169.

The tables below describe how the data for the even lines and for the odd lines of a received frame will be ordered in the image buffer in your PC when the camera is set for Bayer BG12 Packed output.

The following standards are used in the tables:

P_0 = the first pixel transmitted by the camera for a line

P_n = the last pixel transmitted by the camera a line

B_0 = the first byte of data for a line

B_m = the last byte of data for a line

Even Lines		
Byte	**Data**	
B_0	Green value for P_0 bits 11 ... 4	
B_1	Blue value for P_1 bits 3 ... 0	Green value for P_0 bits 3 ... 0
B_2	Blue value for P_1 bits 11 ... 4	
B_3	Green value for P_2 bits 11 ... 4	
B_4	Blue value for P_3 bits 3 ... 0	Green value for P_2 bits 3 ... 0
B_5	Blue value for P_3 bits 11 ... 4	
B_6	Green value for P_4 bits 11 ... 4	
B_7	Blue value for P_5 bits 3 ... 0	Green value for P_4 bits 3 ... 0
B_8	Blue value for P_5 bits 11 ... 4	
•	•	
•	•	•
•	•	
B_{m-5}	Green value for P_{n-3} bits 11 ... 4	
B_{m-4}	Blue value for P_{n-2} bits 3 ... 0	Green value for P_{n-3} bits 3 ... 0
B_{m-3}	Blue value for P_{n-2} bits 11 ... 4	
B_{m-2}	Green value for P_{n-1} bits 11 ... 4	
B_{m-1}	Blue value for P_n bits 3 ... 0	Green value for P_{n-1} bits 3 ... 0
B_m	Blue value for P_n bits 11 ... 4	

Odd Lines		
Byte	**Data**	
B_0	Red value for P_0 bits 11 ... 4	
B_1	Green value for P_1 bits 3 ... 0	Red value for P_0 bits 3 ... 0
B_2	Green value for P_1 bits 11 ... 4	
B_3	Red value for P_2 bits 11 ... 4	
B_4	Green value for P_3 bits 3 ... 0	Red value for P_2 bits 3 ... 0
B_5	Green value for P_3 bits 11 ... 4	
B_6	Red value for P_4 bits 11 ... 4	
B_7	Green value for P_5 bits 3 ... 0	Red value for P_4 bits 3 ... 0
B_8	Green value for P_5 bits 11 ... 4	
•	•	
•	•	•
•	•	
•	•	
•	•	•
•	•	
B_{m-5}	Red value for P_{n-3} bits 11 ... 4	
B_{m-4}	Green value for P_{n-2} bits 3 ... 0	Red value for P_{n-3} bits 3 ... 0
B_{m-3}	Green value for P_{n-2} bits 11 ... 4	
B_{m-2}	Red value for P_{n-1} bits 11 ... 4	
B_{m-1}	Green value for P_n bits 3 ... 0	Red value for P_{n-1} bits 3 ... 0
B_m	Green value for P_n bits 11 ... 4	

When a color camera is set for Bayer BG 12 Packed, the pixel data output is 12 bit data of the “unsigned” type. The available range of data values and the corresponding indicated signal levels are as shown in the table below.

This Data Value (Hexadecimal)	**Indicates This Signal Level (Decimal)**
0x0FFF	4095
0x0FFE	4094
•	•
•	•
•	•
0x0001	1
0x0000	0

11.3.6 YUV 4:2:2 Packed Format (equivalent to DCAM YUV 4:2:2)

When a color camera is set for the YUV 422 Packed pixel data format, each pixel in the captured image goes through a two step conversion process as it exits the sensor and passes through the camera's electronics. This process yields Y, U, and V color information for each pixel.

In the first step of the process, an interpolation algorithm is performed to get full RGB data for each pixel. This is required because color cameras use a Bayer filter on the sensor and each individual pixel gathers information for only one color.

> **Note**
>
> The scA750-60 outputs 10 bits of brightness data per pixel with 8 bits effective. The 8 bits of effective pixel data fill from the least significant bit. The two unused most significant bits are filled with zero.

For more information on the Bayer filter, see Section 11.3.1 on page 169.

The second step of the process is to convert the RGB information to the YUV color model. The conversion algorithm uses the following formulas:

$$Y = 0.30\ R + 0.59\ G + 0.11\ B$$

$$U = -0.17\ R - 0.33\ G + 0.50\ B$$

$$V = 0.50\ R - 0.41\ G - 0.09\ B$$

Once the conversion to a YUV color model is complete, the pixel data is transmitted to the host PC.

> **Note**
>
> The values for U and for V normally range from -128 to +127. Because the camera transfers U values and V values with unsigned integers, 128 is added to each U value and to each V value before the values are transferred from the camera. This process allows the values to be transferred on a scale that ranges from 0 to 255.

The table below describes how the pixel data for a received frame will be ordered in the image buffer in your PC when the camera is set for YUV 4:2:2 Packed output.

The following standards are used in the table:

P_0 = the first pixel transmitted by the camera

P_n = the last pixel transmitted by the camera

B_0 = the first byte in the buffer

B_m = the last byte in the buffer

Byte	**Data**
B_0	U value for P_0
B_1	Y value for P_0
B_2	V Value for P_0
B_3	Y value for P_1
B_4	U value for P_2
B_5	Y value for P_2
B_6	V Value for P_2
B_7	Y value for P_3
B_8	U value for P_4
B_9	Y value for P_4
B_{10}	V Value for P_4
B_{11}	Y value for P_5
•	•
•	•
•	•
B_{m-7}	U value for P_{n-3}
B_{m-6}	Y value for P_{n-3}
B_{m-5}	V Value for P_{n-3}
B_{m-4}	Y value for P_{n-2}
B_{m-3}	U value for P_{n-1}
B_{m-2}	Y value for P_{n-1}
B_{m-1}	V Value for P_{n-1}
B_m	Y value for P_n

When the camera is set for YUV 4:2:2 Packed output, the pixel data output for the Y component is 8 bit data of the “unsigned char” type. The range of data values for the Y component and the corresponding indicated signal levels are shown below.

This Data Value (Hexadecimal)	**Indicates This Signal Level (Decimal)**
0xFF	255
0xFE	254
•	•
•	•
•	•
0x01	1
0x00	0

The pixel data output for the U component or the V component is 8 bit data of the “straight binary” type. The range of data values for a U or a V component and the corresponding indicated signal levels are shown below.

This Data Value (Hexadecimal)	**Indicates This Signal Level (Decimal)**
0xFF	127
0xFE	126
•	•
•	•
•	•
0x81	1
0x80	0
0x7F	-1
•	•
•	•
•	•
0x01	-127
0x00	-128

The signal level of a U component or a V component can range from -128 to +127 (decimal). Notice that the data values have been arranged to represent the full signal level range.

11.3.7 YUV 4:2:2 (YUYV) Packed Format

On color cameras, the YUV 4:2:2 (YUYV) packed pixel data format is similar to the YUV 4:2:2 pixel format described in the previous section. The only difference is the order of the bytes transmitted to the host PC. With the YUV 4:2:2 format, the bytes are ordered as specified in the DCAM standard issued by the 1394 Trade Association. With the YUV 4:2:2 (YUYV) format, the bytes are ordered to emulate the ordering normally associated with analog frame grabbers and Windows® frame buffers.

Note

The scA750-60 outputs 10 bits of brightness data per pixel with 8 bits effective. The 8 bits of effective pixel data fill from the least significant bit. The two unused most significant bits are filled with zero.

The table below describes how the pixel data for a received frame will be ordered in the image buffer in your PC when the camera is set for YUV 4:2:2 (YUYV) output.

With this format, the Y component is transmitted for each pixel, but the U and V components are only transmitted for every second pixel.

The following standards are used in the table:

P_0 = the first pixel transmitted by the camera

P_n = the last pixel transmitted by the camera

B_0 = the first byte in the buffer

B_m = the last byte in the buffer

Byte	Data
B_0	Y value for P_0
B_1	U value for P_0
B_2	Y value for P_1
B_3	V value for P_0
B_4	Y value for P_2
B_5	U value for P_2
B_6	Y value for P_3
B_7	V value for P_2
B_8	Y value for P_4
B_9	U value for P_4
B_{10}	Y value for P_5
B_{11}	V value for P_4
•	•
•	•
•	•

B_{m-7}	Y value for P_{n-3}
B_{m-6}	U value for P_{n-3}
B_{m-5}	Y value for P_{n-2}
B_{m-4}	V value for P_{n-3}
B_{m-3}	Y value for P_{n-1}
B_{m-2}	U value for P_{n-1}
B_{m-1}	Y value for P_n
B_m	V value for P_{n-1}

When a color camera is set for YUV 4:2:2 (YUYV) output, the pixel data output for the Y component is 8 bit data of the “unsigned char” type. The range of data values for the Y component and the corresponding indicated signal levels are shown below.

This Data Value (Hexadecimal)	**Indicates This Signal Level (Decimal)**
0xFF	255
0xFE	254
•	•
•	•
•	•
0x01	1
0x00	0

The pixel data output for the U component or the V component is 8 bit data of the “straight binary” type. The range of data values for a U or a V component and the corresponding indicated signal levels are shown below.

This Data Value (Hexadecimal)	**Indicates This Signal Level (Decimal)**
0xFF	127
0xFE	126
•	•
•	•
•	•
0x81	1
0x80	0
0x7F	-1
•	•
•	•
•	•
0x01	-127
0x00	-128

The signal level of a U component or a V component can range from -128 to +127 (decimal). Notice that the data values have been arranged to represent the full signal level range.

11.3.8 Mono 8 Format (equivalent to DCAM Mono 8)

When a color camera is set for the Mono 8 pixel data format, the pixel values in each captured image are first interpolated and converted to the YUV color model as described for the YUV 4:2:2 Packed format. The camera then transmits the 8 bit Y value for each pixel to the host PC. In the YUV color model, the Y component for each pixel represents a brightness value. This brightness value can be considered as equivalent to the value that would be sent from a pixel in a monochrome camera. So in essence, when a color camera is set for Mono 8, it outputs an 8 bit monochrome image. (This type of output is sometimes referred to as "Y Mono 8".)

Note

The scA750-60 outputs 10 bits of brightness data per pixel with 8 bits effective. The 8 bits of effective pixel data fill from the least significant bit. The two unused most significant bits are filled with zero.

The table below describes how the pixel data for a received frame will be ordered in the image buffer in your PC when a color camera is set for Mono 8 output.

The following standards are used in the table:

P_0 = the first pixel transmitted by the camera

P_n = the last pixel transmitted by the camera

B_0 = the first byte in the buffer

B_m = the last byte in the buffer

Byte	Data
B_0	Y value for P_0
B_1	Y value for P_1
B_2	Y value for P_2
B_3	Y value for P_3
B_4	Y value for P_4
B_5	Y value for P_5
B_6	Y value for P_6
B_7	Y value for P_7

•	•
•	•
•	•
B_{m-3}	Y value for P_{n-3}
B_{m-2}	Y value for P_{n-2}
B_{m-1}	Y value for P_{n-1}
B_m	Y value for P_n

With the camera set for Mono 8, the pixel data output is 8 bit data of the “unsigned char” type. The available range of data values and the corresponding indicated signal levels are as shown in the table below.

This Data Value (Hexadecimal)	**Indicates This Signal Level (Decimal)**
0xFF	255
0xFE	254
•	•
•	•
•	•
0x01	1
0x00	0

11.4 Pixel Transmission Sequence

For each captured image, pixel data is transmitted from the camera in the following sequence:

$\text{Row}_0\ \text{Col}_0$,	$\text{Row}_0\ \text{Col}_1$,	$\text{Row}_0\ \text{Col}_2$	.. ..	$\text{Row}_0\ \text{Col}_{m-2}$,	$\text{Row}_0\ \text{Col}_{m-1}$,	$\text{Row}_0\ \text{Col}_m$
$\text{Row}_1\ \text{Col}_0$,	$\text{Row}_1\ \text{Col}_1$,	$\text{Row}_1\ \text{Col}_2$	.. ..	$\text{Row}_1\ \text{Col}_{m-2}$,	$\text{Row}_1\ \text{Col}_{m-1}$,	$\text{Row}_1\ \text{Col}_m$
$\text{Row}_2\ \text{Col}_0$,	$\text{Row}_2\ \text{Col}_1$,	$\text{Row}_2\ \text{Col}_2$	.. ..	$\text{Row}_2\ \text{Col}_{m-2}$,	$\text{Row}_2\ \text{Col}_{m-1}$,	$\text{Row}_2\ \text{Col}_m$
:	:	:		:	:	:
:	:	:		:	:	:
$\text{Row}_{n-2}\ \text{Col}_0$,	$\text{Row}_{n-2}\ \text{Col}_1$,	$\text{Row}_{n-2}\ \text{Col}_2$	.. ..	$\text{Row}_{n-2}\ \text{Col}_{m-2}$,	$\text{Row}_{n-2}\ \text{Col}_{m-1}$,	$\text{Row}_{n-2}\ \text{Col}_m$
$\text{Row}_{n-1}\ \text{Col}_0$,	$\text{Row}_{n-1}\ \text{Col}_1$,	$\text{Row}_{n-1}\ \text{Col}_2$	.. ..	$\text{Row}_{n-1}\ \text{Col}_{m-2}$,	$\text{Row}_{n-1}\ \text{Col}_{m-1}$,	$\text{Row}_{n-1}\ \text{Col}_m$
$\text{Row}_n\ \text{Col}_0$,	$\text{Row}_n\ \text{Col}_1$,	$\text{Row}_n\ \text{Col}_2$	.. ..	$\text{Row}_n\ \text{Col}_{m-2}$,	$\text{Row}_n\ \text{Col}_{m-1}$,	$\text{Row}_n\ \text{Col}_m$

Where $\text{Row}_0\ \text{Col}_0$ is the upper left corner of the sensor

The columns are numbered 0 through m from the left side to the right side of the sensor

The rows are numbered 0 through n from the top to the bottom of the sensor

The sequence assumes that the camera is set for full resolution.

12 I/O Control

This section describes how to configure the camera's two physical input lines and four physical output lines. It also provides information about monitoring the state of the input and output lines.

For more detailed information about the physical and electrical characteristics of the input and output lines, see Section 9.7 on page 120.

12.1 Configuring Input Lines

12.1.1 Assigning an Input Line to Receive a Hardware Trigger Signal

You can assign one of the camera's input lines to receive a external hardware trigger (ExTrig) signal. The incoming ExTrig signal can then be used to control image acquisition.

Section 10.3.2 on page 135 explains how to configure the camera to react to a hardware trigger signal and how to assign an input line to receive the hardware trigger signal.

Note

By default, physical input line 1 is assigned to receive the ExTrig signal. You can assign only one line to receive the ExTrig input signal.

12.1.2 Using an Unassigned Input Line to Receive a User Input Signal

You can use an unassigned input line to receive your own, user-generated input signal. The electrical characteristics of your input signal must meet the requirements shown in the Physical Interface section of this manual.

You can use the Line Status or Line Status All parameters to monitor the state of the input line that is receiving the user-defined signal.

Note

The line assigned to receive the ExTrig input signal can’t be used to receive a user-designed input signal.

For more information about using the Line Status and Line Status All parameters, see Section 12.3.1 on page 199 and Section 12.3.2 on page 199.

12.2 Configuring Output Lines

12.2.1 Assigning a Camera Output Signal to a Physical Output Line

You can use the camera's output signal assignment capability to assign one of the camera's standard output signals to a physical output line. The camera has a variety of standard output signals available including:

- Exposure Active
- Trigger Ready
- Timer 1, Timer 2, Timer 3, Timer 4

You can also designate an output line as "user settable". If an output line is designated as a user settable, you can use the camera's API to set the state of the line as desired.

To assign an output signal to an output line or to designate the line as user settable:

- Use the Line Selector to select Output Line 1, Output Line 2, Output Line 3, or Output Line 4.
- Set the value of the Line Source Parameter to one of the available output signals or to user settable. This will set the signal for the selected line.

Note

By default, the Exposure Active signal is assigned to Output Line 1 and the Trigger Ready Signal is assigned to Output Line 2.

You can set the Line Selector and the Line Source parameter value from within your application software by using the pylon API. The following code snippet illustrates using the API to set the selector and the parameter value:

```
Camera.LineSelector.SetValue( LineSelector_Out1 );
Camera.LineSource.SetValue( LineSource_ExposureActive );
Camera.LineSelector.SetValue( LineSelector_Out2 );
Camera.LineSource.SetValue( LineSource_TriggerReady );
```

For detailed information about using the pylon API, refer to the Basler pylon Programmer's Guide and API Reference.

You can also use the Basler pylon Viewer application to easily set the parameters.

For more information about the pylon Viewer, see Section 2.2 on page 32 and Section 2.4 on page 52.

For more information about setting the state of user settable output signals, see Section 12.2.2 on page 192.

For more information about working with the timer output signals, see Section 12.2.4 on page 194

For more information about the exposure active signal, see Section 10.8 on page 149.

For more information about the trigger ready signal, see Section 10.7 on page 145.

12.2.2 Setting the State of User Settable Output Lines

As mentioned in the previous section, you can designate one or more of the user output lines as "user settable". Once you have designated an output line as user settable, you can use camera parameters to set the state of the line.

Setting the State of a Single User Settable Output Line

To set the state of a single user settable output line:

- Use the User Output Selector to select the output line you want to set. For example, if you have designated output line 3 as user settable, you would select user settable output 3.
- Set the value of the User Output Value parameter to true (high) or false (low). This will set the state of the selected line.

You can set the Output Selector and the User Output Value parameter from within your application software by using the pylon API. The following code snippet illustrates using the API to designate output line 3 as user settable and setting the state of the output line:

```
Camera.LineSelector.SetValue( LineSelector_Out3 );
Camera.LineSource.SetValue( LineSource_UserOutput );
Camera.UserOutputSelector.SetValue( UserOutputSelector_UserOutput3 );
Camera.UserOutputValue.SetValue( true );
bool currentUserOutput3State = Camera.UserOutputValue.GetValue( );
```

For detailed information about using the pylon API, refer to the Basler pylon Programmer's Guide and API Reference.

You can also use the Basler pylon Viewer application to easily set the parameters.

Setting the State of Multiple User Settable Output Lines

The User Output Value All parameter is a 32 bit value. As shown in Figure 56, the lowest four bits of the parameter value will set the state of the user settable outputs. If a bit is 0, it will set the state of the associated output to low. If a bit is high, it will set the state of the associated port to high.

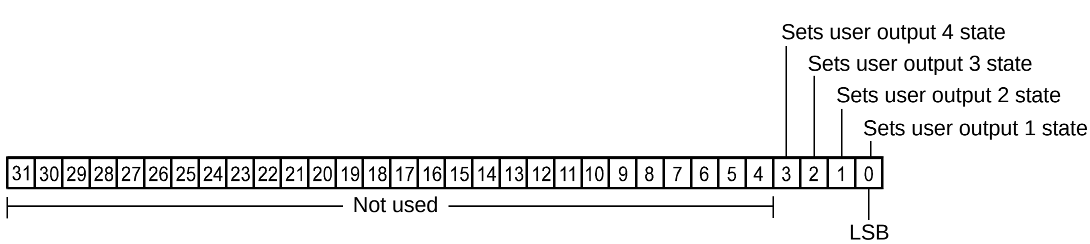


Fig. 56: User Output Value All Parameter Bits

To set the state of multiple user settable output lines:

- Use the User Output Value All parameter to set the state of multiple user settable outputs.

You can set the User Output Value All parameter from within your application software by using the pylon API. The following code snippet illustrates using the API to set the parameter:

```
Camera.UserOutputValueAll.SetValue( 0x3 );
int64_t currentOutputState = Camera.UserOutputValueAll.GetValue( );
```

Note

If you have the invert function enabled on an output line that is designated as user settable, the user setting sets the state of the line before the inverter.

12.2.3 Setting an Output Line for Invert

You can set each individual output line to invert or not to invert the outgoing signal. To set the invert function on an output line:

- Use the Line Selector to select Output Line 1, Output Line 2, Output Line 3, or Output Line 4.
- Set the value of the Line Inverter parameter to true to enable inversion on the selected line and to false to disable inversion.

You can set the Line Selector and the Line Inverter parameter value from within your application software by using the pylon API. The following code snippet illustrates using the API to set the selector and the parameter value:

```
Camera.LineSelector.SetValue( LineSelector_Out1 );
Camera.LineInverter.SetValue( true );
```

For detailed information about using the pylon API, refer to the Basler pylon Programmer's Guide and API Reference.

You can also use the Basler pylon Viewer application to easily set the parameters.

For more information about the pylon Viewer, see Section 2.2 on page 32 and Section 2.4 on page 52.

12.2.4 Working with Timers

The camera has four timer output signals available: Timer 1, Timer 2, Timer 3, and Timer 4. As shown in Figure 57, each timer works as follows:

- A trigger source event occurs that starts the timer.
- A delay period begins to expire.
- When the delay expires, the timer signal goes high and a duration period begins to expire.
- When the duration period expires, the timer signal goes low.

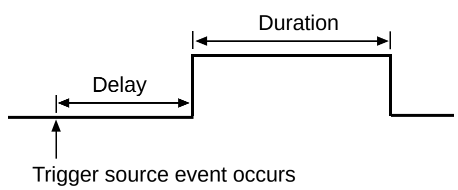


Fig. 57: Timer Signal

Currently, the only trigger source event available to start the timer is "exposure active". In other words, you can use exposure start to trigger the start of a timer.

Timer 1 can only be assigned to output line 1. Timer 2 can only be assigned to output line 2. Timer 3 can only be assigned to output line 3. Timer 4 can only be assigned to output line 4.

If you require the timer signal to be high when the timer is triggered and to go low when the delay expires, simply set the output line to invert.

12.2.4.1 Setting the Trigger Source for a Timer

To set the source event for a timer:

- Use the Timer Selector to select Timer 1, Timer 2, Timer 3, Or Timer 4.
- Set the value of the Timer Trigger Source parameter to exposure active. This will set the selected timer to use the start of exposure to begin the timer.

You can set the Trigger Selector and the Timer Trigger Source parameter value from within your application software by using the pylon API. The following code snippet illustrates using the API to set the selector and the parameter value:

```
Camera.TimerSelector.SetValue( TimerSelector_Timer1 );
Camera.TimerTriggerSource.SetValue( TimerTriggerSource_ExposureStart );
```

For detailed information about using the pylon API, refer to the Basler pylon Programmer's Guide and API Reference.

You can also use the Basler pylon Viewer application to easily set the parameters.

For more information about the pylonViewer, see Section 2.2 on page 32 and Section 2.4 on page 52.

12.2.4.2 Setting a Timer Delay Time

There are two ways to set the delay time for a timer: by setting "raw" values or by setting an "absolute value". You can use whichever method you prefer to set the delay time.

Setting the Delay with Raw Values

When the delay time for a timer is set using "raw" values, the delay time will be determined by a combination of two elements. The first element is the value of the Timer Delay Raw parameter, and the second element is the Timer Delay Time Base. The delay time is the product of these two elements:

Delay Time = (Timer Delay Raw Parameter Value) x (Timer Delay Time Base)

By default, the Timer Delay Time Base is fixed at 1 µs. Typically, the delay time is adjusted by setting the Timer Delay Raw parameter value.

The Timer Delay Raw parameter value can range from 1 to 4095. So if the value is set to 100, for example, the timer delay will be 100 x 1 µs or 100 µs.

To set the delay for a timer:

- Use the Timer Selector to select Timer 1, Timer 2, Timer 3, or Timer 4.
- Set the value of the Timer Delay Raw parameter.

You can set the Timer Selector and the Timer Delay Raw parameter value from within your application software by using the pylon API. The following code snippet illustrates using the API to set the selector and the parameter value:

```
Camera.TimerSelector.SetValue( TimerSelector_Timer1 );
Camera.TimerDelayRaw.SetValue( 100 );
```

For detailed information about using the pylon API, refer to the Basler pylon Programmer’s Guide and API Reference.

You can also use the Basler pylon Viewer application to easily set the parameters.

Changing the Delay Time Base

By default, the Timer Delay Time Base is fixed at 1 µs, and the timer delay is normally adjusted by setting the value of the Timer Delay Raw parameter. However, if you require a delay time that is longer than what you can achieve by changing the value of the Timer Delay Raw parameter alone, the Timer Delay Time Base Abs parameter can be used to change the delay time base.

The Timer Delay Time Base Abs parameter value sets the delay time base in µs. The default is 1 µs and it can be changed in 1 µs increments.

Note that there is only one timer delay time base and it is used by all four of the available timers.

You can set the Timer Delay Time Base Abs parameter value from within your application software by using the pylon API. The following code snippet illustrates using the API to set the parameter value:

```
Camera.TimerDelayTimebaseAbs.SetValue( 5 );
```

Setting the Delay with an Absolute Value

You can also set the Timer delay by using an "absolute" value. This is accomplished by setting the Timer Delay Abs parameter. The units for setting this parameter are µs and the value can be set in increments of 1 µs.

To set the delay for a timer using an absolute value:

- Use the Timer Selector to select Timer 1, Timer 2, Timer 3, or Timer 4.
- Set the value of the Timer Delay Abs parameter.

You can set the Timer Selector and the Timer Delay Abs parameter value from within your application software by using the pylon API. The following code snippet illustrates using the API to set the selector and the parameter value:

```
Camera.TimerSelector.SetValue( TimerSelector_Timer1 );
Camera.TimerDelayAbs.SetValue( 100 );
```

When you use the Timer Delay Abs parameter to set the delay time, the camera accomplishes the setting change by automatically changing the Timer Delay Raw parameter to achieve the value specified by the Timer Delay Abs setting. This leads to a limitation that you must keep in mind if you use Timer Delay Abs parameter to set the delay time. That is, you must set the Timer Delay Abs parameter to a value that is equivalent to a setting you could achieve by using the Timer Delay Raw and the current Timer Delay Base parameters. For example, if the time base was currently set to 50 µs, you could use the Timer Delay Abs parameter to set the delay to 50 µs, 100 µs, 150 µs, etc.

Note that if you set the Timer Delay Abs parameter to a value that you could not achieve by using the Timer Delay Raw and current Timer Delay Time Base parameters, the camera will automatically change the setting for the Timer Delay Abs parameter to the nearest achieveable value.

You should also be aware that if you change the delay time using the raw settings, the Timer Delay Abs parameter will automatically be updated to reflect the new delay time.

12.2.4.3 Setting a Timer Duration Time

There are two ways to set the duration time for a timer: by setting "raw" values or by setting an "absolute value". You can use whichever method you prefer to set the duration time.

Setting the Duration with Raw Values

When the duration time for a timer is set using "raw" values, the duration time will be determined by a combination of two elements. The first element is the value of the Timer Duration Raw parameter, and the second element is the Timer Duration Time Base. The duration time is the product of these two elements:

Duration Time = (Timer Duration Raw Parameter Value) x (Timer Duration Time Base)

By default, the Timer Duration Time Base is fixed at 1 µs. Typically, the duration time is adjusted by setting only the Timer Duration Raw parameter value.

The Timer Duration Raw parameter value can range from 1 to 4095. So if the value is set to 100, for example, the timer duration will be 100 x 1 µs or 100 µs.

To set the duration for a timer:

- Use the Timer Selector to select Timer 1, Timer 2, Timer 3, or Timer 4.
- Set the value of the Timer Duration Raw parameter.

You can set the Timer Selector and the Timer Duration Raw parameter value from within your application software by using the pylon API. The following code snippet illustrates using the API to set the selector and the parameter value:

```
Camera.TimerSelector.SetValue( TimerSelector_Timer1 );
Camera.TimerDurationRaw.SetValue( 100 );
```

For detailed information about using the pylon API, refer to the Basler pylon Programmer's Guide and API Reference.

You can also use the Basler pylon Viewer application to easily set the parameters.

Changing the Duration Time Base

By default, the Timer Duration Time Base is fixed at 1 µs, and the timer duration is normally adjusted by setting the value of the Timer Duration Raw parameter. However, if you require a duration time that is longer than what you can achieve by changing the value of the Timer Duration Raw parameter alone, the Timer Duration Time Base Abs parameter can be used to change the duration time base.

The Timer Duration Time Base Abs parameter value sets the duration time base in µs. The default is 1 µs and it can be changed in 1 µs increments.

Note that there is only one timer duration time base and it is used by all four of the available timers.

You can set the Timer Duration Time Base Abs parameter value from within your application software by using the pylon API. The following code snippet illustrates using the API to set the parameter value:

```
Camera.TimerDurationTimebaseAbs.SetValue( 5 );
```

Setting the Duration with an Absolute Value

You can also set the Timer duration by using an "absolute" value. This is accomplished by setting the Timer Duration Abs parameter. The units for setting this parameter are µs and the value can be set in increments of 1 µs.

To set the duration for a timer using an absolute value:

- Use the Timer Selector to select Timer 1, Timer 2, Timer 3, or Timer 4.
- Set the value of the Timer Duration Abs parameter.

You can set the Timer Selector and the Timer Duration Abs parameter value from within your application software by using the pylon API. The following code snippet illustrates using the API to set the selector and the parameter value:

```
Camera.TimerSelector.SetValue( TimerSelector_Timer1 );
Camera.TimerDurationAbs.SetValue( 100 );
```

When you use the Timer Duration Abs parameter to set the duration time, the camera accomplishes the setting change by automatically changing the Timer Duration Raw parameter to achieve the value specified by the Timer Duration Abs setting. This leads to a limitation that you must keep in mind if you use Timer Duration Abs parameter to set the duration time. That is, you must set the Timer Duration Abs parameter to a value that is equivalent to a setting you could achieve by using the Timer Duration Raw and the current Timer Duration Base parameters. For example, if the time base was currently set to 50 µs, you could use the Timer Duration Abs parameter to set the duration to 50 µs, 100 µs, 150 µs, etc.

If you read the current value of the Timer Duration Abs parameter, the value will indicate the product of the Timer Duration Raw parameter and the Timer Duration Time Base. In other words, the Timer Duration Abs parameter will indicate the current duration time setting.

You should also be aware that if you change the duration time using the raw settings, the Timer Duration Abs parameter will automatically be updated to reflect the new duration time.

12.3 Checking the State of the I/O Lines

12.3.1 Checking the State of a Single Output Line

You can determine the current state of an individual output line. To check the state of a line:

- Use the Line Selector parameter to select Output Line 1, Output Line 2, Output Line 3, or Output Line 4.
- Read the value of the Line Status parameter to determine the current state of the selected line. A value of true means the line's state is currently high and a value of false means the line's state is currently low.

You can set the Line Selector and read the Line Status parameter value from within your application software by using the pylon API. The following code snippet illustrates using the API to set the selector and read the parameter value:

```
Camera.LineSelector.SetValue( LineSelector_Out2 );
bool outputLine2State = Camera.LineStatus.GetValue( );
```

For detailed information about using the pylon API, refer to the Basler pylon Programmer's Guide and API Reference.

You can also use the Basler pylon Viewer application to easily set the parameters.

For more information about the pylon Viewer, see Section 2.2 on page 32 and Section 2.4 on page 52.

12.3.2 Checking the State of All Lines

You can determine the current state of all input and output lines with a single operation. To check the state of all lines:

- Read the value of the Line Status All parameter.

You can read the Line Status All parameter value from within your application software by using the pylon API. The following code snippet illustrates using the API to read the parameter value:

```
int64_t lineState = Camera.LineStatusAll.GetValue( );
```

The Line Status All parameter is a 32 bit value. As shown in Figure 58, certain bits in the value are associated with each line and the bits will indicate the state of the lines. If a bit is 0, it indicates that the state of the associated line is currently low. If a bit is 1, it indicates that the state of the associated line is current high.

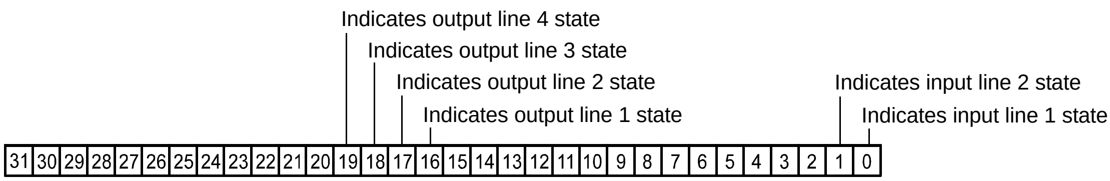


Fig. 58: Line Status All Parameter Bits

13 Standard Features

This section provides detailed information about the standard features available on each camera. It also includes an explanation of the operation and the parameters associated with each feature.

13.1 Gain

The camera's gain setting is adjustable. As shown in Figure 59, increasing the gain increases the slope of the response curve for the camera. This results in a higher gray value output from the camera for a given amount of output from the imaging sensor. Decreasing the gain decreases the slope of the response curve and results in a lower gray value for a given amount of sensor output.

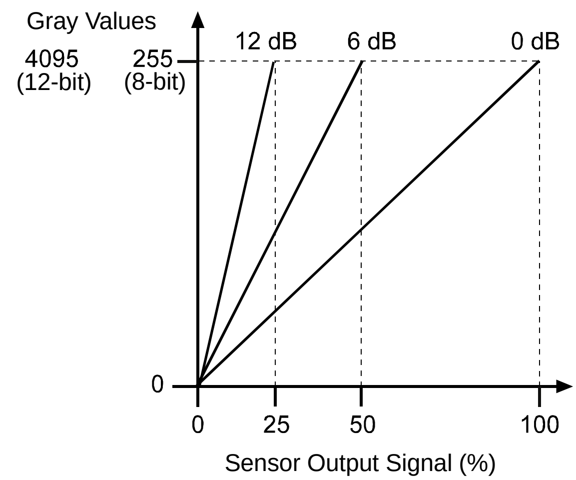


Fig. 59: Gain in dB

Increasing the gain is useful when at your brightest exposure, a gray value lower than 255 (in modes that output 8 bits per pixel) or 4095 (in modes that output 12 bits per pixels) is reached. For example, if you found that at your brightest exposure the gray values output by the camera were no higher than 127 (in an 8 bit mode), you could increase the gain to 6 dB (an amplification factor of 2) and thus reach gray values of 254.

Setting the Gain (all models except scA750-60)

Note

The information in this section applies to all camera models except the scA750-60 gm/gc. For information about scA750-60 cameras, see the next section.

The camera's gain is determined by the value of the Gain Raw parameter. Gain Raw is adjusted on a decimal scale. The minimum decimal setting varies depending on the camera model as shown in Table 11. The maximum setting depends on whether the camera is set for a pixel data format that yields 8 bit effective pixel depth (Mono 8, Bayer BG 8, YUV 4:2:2 Packed, YUV 4:2:2 (YUYV) Packed) or yields an effective pixel depth of 12 bits per pixel (Mono 16, Mono 12 Packed, Bayer BG 16, Bayer BG 12 Packed).

Camera Model	Min Setting	Max Setting (8 bit depth)	Max Setting (16 bit depth)
scA640-70	320	1023	511
scA640-74	280	1023	511
scA780-54	350	1023	511
scA1000-20	330	1023	511
scA1000-30	360	1023	511
scA1390-17	360	1023	511
scA1400-17	192	1023	511
scA1600-14	350	1023	511

Table 11: Minimum and Maximum Allowed Gain Raw Settings

To set the Gain Raw parameter value:

- Set the Gain Selector to Gain All.
- Set the Gain Raw parameter to your desired value.

You can set the Gain Selector and the Gain Raw parameter value from within your application software by using the pylon API. The following code snippet illustrates using the API to set the selector and the parameter value:

```
Camera.GainSelector.SetValue( GainSelector_All );
Camera.GainRaw.SetValue( 400 );
```

For detailed information about using the pylon API, refer to the Basler pylon Programmer's Guide and API Reference.

You can also use the Basler pylon Viewer application to easily set the parameters.

For more information about the pylon Viewer, see Section 2.2 on page 32and Section 2.4 on page 52.

If you know the current decimal setting for the gain raw, you can use the formulas below to calculate the dB of gain that will result from that setting.

For gain raw settings from 192 to 511:

$$\text{Gain}_{\text{dB}} = 20 \times \log_{10}\left(\frac{658 + \text{Gain Raw Setting}}{658 - \text{Gain Raw Setting}}\right) - G_c$$

For gain raw settings from 512 to 1023:

$$\text{Gain}_{\text{dB}} = (0.0354 \times \text{Gain Raw Setting}) - G_c$$

Where:

$$G_c = 20 \times \log_{10}\left(\frac{658 + \text{Min Gain Raw Setting}}{658 - \text{Min Gain RawSetting}}\right)$$

Example:

Assume that you are working with a monochrome scA1400-17 camera that is set for the Mono 8 color coding and has a gain raw setting of 500. Calculating the gain is a two step process:

Step 1:

$$G_c = 5.22\text{ dB}$$

Step 2:

$$\text{Gain}_{\text{dB}} = 12.1\text{ dB}$$

Table 12 shows the minimum and maximum gain in dB for each camera model.

Camera Model	dB Gain at Min Setting	dB Gain at Max Setting (8 bit depth)	dB Gain at Max Setting (16 bit depth)
scA640-70	0	27	8.8
scA640-74	0	28.3	10.1
scA780-54	0	25.9	7.7
scA1000-20	0	26.6	8.4
scA1000-30	0	25.5	7.3
scA1390-17	0	25.5	7.3
scA1400-17	0	31.0	12.8
scA1600-14	0	25.9	7.7

Table 12: Minimum and Maximum db of Gain

Setting the Gain (scA750-60 only)

Note

The information in this section only applies to scA750-60 gm/gc cameras. For information about the other camera models, see the previous section.

The camera's gain is determined by the value of the Gain Raw parameter. Gain Raw is adjusted on a decimal scale. The range for the Gain Raw parameter setting is from 0 to 22.

To set the Gain Raw parameter value:

- Set the Gain Selector to Gain All.
- Set the Gain Raw parameter to your desired value.

You can set the Gain Selector and the Gain Raw parameter value from within your application software by using the pylon API. The following code snippet illustrates using the API to set the selector and the parameter value:

```
Camera.GainSelector.SetValue( GainSelector_All );
Camera.GainRaw.SetValue( 20 );
```

For detailed information about using the pylon API, refer to the Basler pylon Programmer's Guide and API Reference.

You can also use the Basler pylon Viewer application to easily set the parameters.

For more information about the pylon Viewer, see Section 2.2 on page 32and Section 2.4 on page 52.

If you know the current decimal setting for the gain raw, you can use the following formula to calculate the dB of gain that will result from that setting:

$$\mathrm{Gain}_{dB} = 20 \times \log_{10}\left(1 + \frac{\text{Gain Raw Setting}}{6}\right)$$

Example:

Assume that you are working with an scA750-60 camera that has a gain raw setting of 18. The gain is calculated as follows:

$$\mathrm{Gain}_{dB} = 12.0\ \mathrm{dB}$$

Table 13 shows the dB of gain that will be achieved at various Gain Raw settings.

Gain Setting	dB Gain
0	0
5	5.3
10	8.5
15	10.9
20	12.7
22	13.4

Table 13: dB of Gain at Various Settings

13.2 Black Level

Adjusting the camera's black level will result in an offset to the pixel values output by the camera. Increasing the black level setting will result in a positive offset in the digital values output for the pixels. Decreasing the black level setting will result in a negative offset in the digital values output for the pixels.

Effect on All Camera Models Except the scA750-60

If the camera is set for a pixel data format that yields 8 bit effective pixel depth (Mono 8, Bayer BG 8, Bayer RG 8, YUV 4:2:2 Packed, YUV 4:2:2 (YUYV) Packed), an increase of 16 in the black level parameter setting will result in a positive offset of 1 in the digital values output for the pixels. And a decrease of 16 in the setting will result in a negative offset of 1 in the digital values output for the pixels.

If the camera is set for a pixel data format that yields an effective pixel depth of 12 bits per pixel (Mono 16, Mono 12 Packed, Bayer BG 16, Bayer RG 16, Bayer BG 12 Packed), an increase of 1 in the black level parameter setting will result in a positive offset of 1 in the digital values output for the pixels. A decrease of 1 in the setting will result in a negative offset of 1 in the digital values output for the pixels.

Effect on scA750-60 Models

An increase of 4 in the black level parameter setting will result in a positive offset of 1 in the digital values output for the pixels. And a decrease of 4 in the setting will result in a negative offset of 1 in the digital values output for the pixels.

Setting the Black Level

The black level can be adjusted by changing the value of the Black Level Raw parameter. The Black Level Raw parameter value can range from 0 to 255 on all camera models except the scA750-60 gm/gc. On scA750-60 gm/gc cameras, the parameter value can range from 0 to 64.

To set the Black Level Raw parameter value:

- Set the Black Level Selector to Black Level All.
- Set the Black Level Raw parameter to your desired value.

You can set the Black Level Selector and the Black Level Raw parameter value from within your application software by using the pylon API. The following code snippet illustrates using the API to set the selector and the parameter value:

```
Camera.BlackLevelSelector.SetValue ( BlackLevelSelector_All );
Camera.BlackLevelRaw.SetValue( 32 );
```

For detailed information about using the pylon API, refer to the Basler pylon Programmer's Guide and API Reference.

You can also use the Basler pylon Viewer application to easily set the parameters.

For more information about the pylon Viewer, see Section 2.2 on page 32and Section 2.4 on page 52.

13.3 White Balance (on color models)

White balance capability has been implemented on color models of the camera. White balancing can be used to adjust the color balance of the images transmitted from the camera.

Setting the White Balance

With the white balancing scheme used on these cameras, the red intensity, green intensity, and blue intensity can each be adjusted. For each color, a Balance Ratio parameter is used to set the intensity of the color. If the Balance Ratio parameter for a color is set to a value of 1, the intensity of the color will unaffected by the white balance mechanism. If the ratio is set to a value lower than 1, the intensity of the color will be reduced. If the ratio is set to a value greater than 1, the intensity of the color will be increased. The increase or decrease in intensity is proportional. For example, if the balance ratio for a color is set to 1.2, the intensity of that color will be increased by 20%.

The balance ratio value can range from 0.015 to 3.98. But you should be aware that if you set the balance ratio for a color to a value lower than 1, this will not only decrease the intensity of that color relative to the other two colors, but will also decrease the maximum intensity that color can achieve. For this reason, we don't normally recommend setting a balance ratio less than 1.

To set the Balance Ratio parameter for a color:

- Set the Balance Ratio Selector to red, green, or blue.
- Set the Balance Ratio Abs parameter to the desired value for the selected color.

You can set the Balance Ratio Selector and the Balance Ratio Abs parameter value from within your application software by using the pylon API. The following code snippet illustrates using the API to set the selector and the parameter value:

```
Camera.BalanceRatioSelector.SetValue( BalanceRatioSelector_Green );
Camera.BalanceRatioAbs.SetValue( 1.20 );
```

For detailed information about using the pylon API, refer to the Basler pylon Programmer's Guide and API Reference.

You can also use the Basler pylon Viewer application to easily set the parameters.

For more information about the pylon Viewer, see Section 2.2 on page 32and Section 2.4 on page 52.

13.4 Integrated IR Cut Filter (on color models)

Color models of the camera that have a C-mount lens adapter are equipped with an IR cut filter as standard equipment. The filter is mounted inside of the lens adapter. Cameras without an IR cut filter are available on request.

Color cameras that have the optional CS-mount lens adapter do not include an IR cut filter in the adapter.

Monochrome cameras do not include an IR cut filter in the lens adapter. Monochrome cameras with a C-mount lens adapter can be equipped with a filter on request.

CAUTION

Lens Thread Length is Limited

The location of the IR cut filter limits the length of the threads on any lens you use with the camera. If a lens with a very long thread length is used, the IR cut filter will be damaged or destroyed and the camera will no longer operate.

For more information about the location of the IR cut filter, see Section 1.5.3 on page 26.

13.5 Area of Interest (AOI)

The area of interest (AOI) feature lets you specify a portion of the CCD sensor array and after each image is acquired, only the pixel information from the specified portion of the array is transmitted to the host PC.

The area of interest is referenced to the top left corner of the CCD array. The top left corner is designated as column 0 and row 0 as shown in Figure 60.

The location and size of the area of interest is defined by declaring an X offset (coordinate), a width, a Y offset (coordinate), and a height. For example, suppose that you specify the x offset as 10, the width as 16, the y offset as 6, and the height as 10. The area of the array that is bounded by these settings is shown in Figure 60.

The camera will only transfer pixel data from within the area defined by your settings. Information from the pixels outside of the area of interest is discarded.

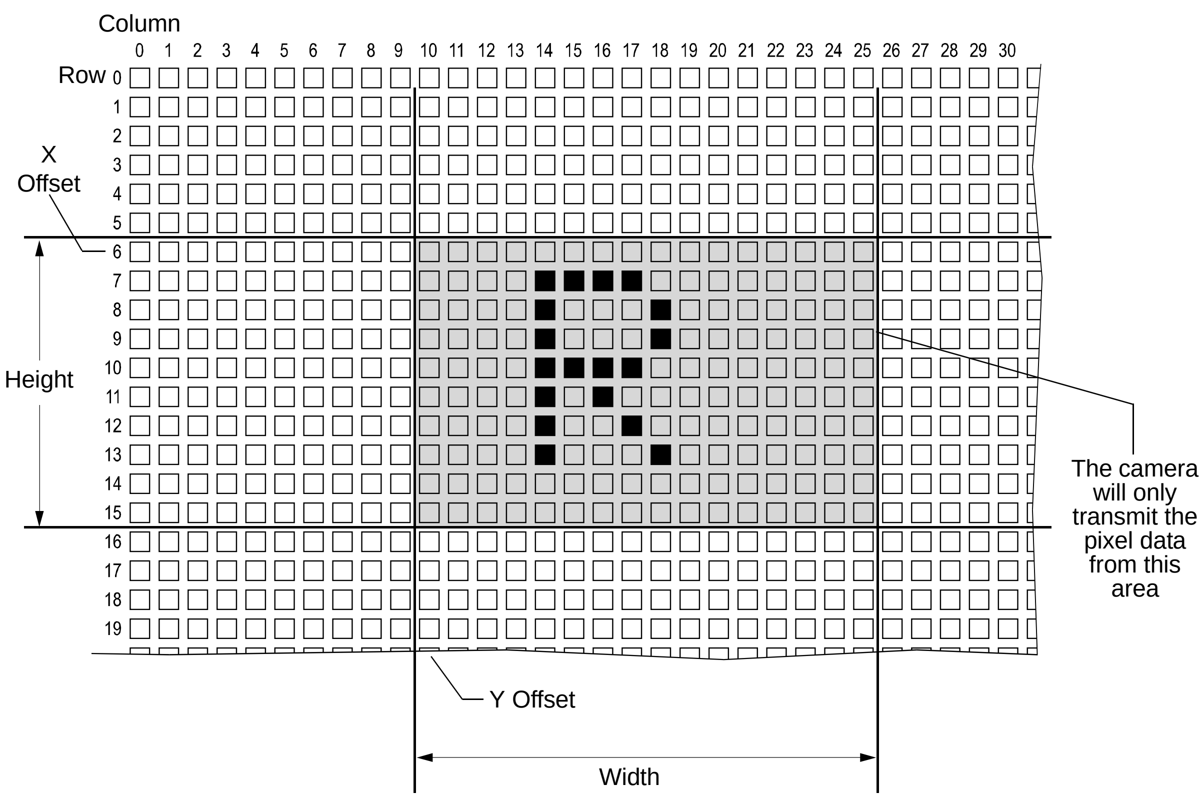


Fig. 60: Area of Interest

One of the main advantages of the AOI feature is that decreasing the height of the AOI can increase the camera's maximum allowed acquisition frame rate.

For more information about how changing the AOI height effects the maximum allowed frame rate, see Section 10.10 on page 152.

Setting the AOI

By default, the AOI is set to use the full resolution of the camera's sensor. You can change the size and the position of the AOI by changing the value of the camera's X Offset, Y Offset, Width, and Height parameters.

- The value of the X Offset parameter determines the starting column for the area of interest.
- The value of the Y Offset parameter determines the starting row for the area of interest.
- The value of the Width parameter determines the width of the area of interest.
- The value of the Height parameter determines the height of the area of interest.

When you are setting the camera's area of interest, you must follow these guidelines:

- The sum of the current X Offset setting plus the current Width setting must not exceed the maximum width of the sensor in the camera model you are using. For example, on the monochrome version of the scA640-70, the sum of the current X Offset setting plus the current Width setting must not exceed 659.
- The sum of the current Y Offset setting plus the current Height setting must not exceed the maximum height of the sensor in the camera model you are using. For example, on the monochrome version of the scA640-70, the sum of the current Y Offset setting plus the current Height setting must not exceed 494.

On monochrome cameras:

- The X Offset, Y Offset, Width, and Height parameters can be set in increments of 1.

On color cameras:

- The X Offset, Y Offset, Width, and Height parameters can be set in increments of 2 and they must be set to an even number. For example, the X Offset parameter can be set to 0, 2, 4, 6, 8, etc.

You can set the X Offset, Y Offset, Width, and Height parameter values from within your application software by using the pylon API. The following code snippets illustrate using the API to get the maximum allows settings and the increments for the Width and Height parameters. They also illustrate setting the X Offset, Y Offset, Width, and Height parameter values

```
int64_t widthMax = Camera.Width.GetMax( );
int64_t widhInc = Camera.Width.GetInc();
Camera.Width.SetValue( 200 );
Camera.OffsetX.SetValue( 100 );

int64_t heightMax = Camera.Height.GetMax( );
int64_t heightInc = Camera.Height.GetInc();
Camera.Height.SetValue( 200 );
Camera.OffsetY.SetValue( 100 );
```

For detailed information about using the pylon API, refer to the Basler pylon Programmer's Guide and API Reference.

You can also use the Basler pylon Viewer application to easily set the parameters.

For more information about the pylon Viewer, see Section 2.2 on page 32and Section 2.4 on page 52.

13.5.1 Changing AOI Parameters "On-the-Fly"

Making AOI parameter changes “on-the-fly” means making the parameter changes while the camera is capturing images continuously. On-the-fly changes are only allowed for the parameters that determine the position of the AOI, i.e., the X Offset and Y Offset parameters. Changes to the AOI size are not allowed on-the-fly.

13.6 Luminance Lookup Table

13.6.1 Lookup Table (all models except scA750-60)

Note

The information in this section applies to all camera models except the scA750-60 gm/gc. For information about scA750-60 cameras, see Section 13.6.2 on page 215.

On these cameras, pixel data is acquired at 12 bit depth. When a monochrome camera is set for the Mono 16 or Mono 12 packed pixel format or a color camera is set for the Bayer BG 16 or the Bayer BG 12 packed pixel format, the camera outputs pixel data with 12 effective bits. Normally, the effective 12 bits transmitted out of the camera for each pixel directly represent the 12 bits reported by the camera's ADC. The luminance lookup table feature lets you use a custom 12 bit to12 bit lookup table to map the 12 bits reported out of the ADC to 12 bits that will be transmitted by the camera.

The lookup table is essentially just a list of 4096 values, however, not every value in the table is actually used. If we number the values in the table from 0 through 4095, the table works like this:

- The number at location 0 in the table represents the effective 12 bits that will be transmitted out of the camera when the sensor reports that a pixel has a value of 0.
- The numbers at locations 1 through 7 are not used.
- The number at location 8 in the table represents the effective 12 bits that will be transmitted out of the camera when the sensor reports that a pixel has a value of 8.
- The numbers at locations 9 through 15 are not used.
- The number at location 16 in the table represents the effective 12 bits that will be transmitted out of the camera when the sensor reports that a pixel has a value of 16.
- The numbers at locations 17 through 23 are not used.
- The number at location 24 in the table represents the effective 12 bits that will be transmitted out of the camera when the sensor reports that a pixel has a value of 24.
- And so on.

As you can see, the table does not include a user defined 12 bit value for every pixel value that the sensor can report. So what does the camera do when the sensor reports a pixel value that is between two values that have a defined 12 bit output? In this case, the camera performs a straight line interpolation to determine the value that it should transmit. For example, assume that the sensor reports a pixel value of 12. In this case, the camera would perform a straight line interpolation between the values at location 8 and location 16 in the table. The result of the interpolation would be reported out of the camera as the 12 bit output.

Another thing to keep in mind about the table is that location 4088 is the last location that will have a defined 12 bit value associated with it. (Locations 4089 through 4095 are not used.) If the sensor reports a value above 4088, the camera will not be able to perform an interpolation. In cases where the sensor reports a value above 4088, the camera simply transmits the 12 bit value from location 4088 in the table.

The advantage of the luminance lookup table feature is that it allows a user to customize the response curve of the camera. The graphs below show the effect of two typical lookup tables. The first graph is for a lookup table where the values are arranged so that the output of the camera increases linearly as the sensor output increases. The second graph is for a lookup table where the values are arranged so that the camera output increases quickly as the sensor output moves from 0 through 2048 and increases gradually as the sensor output moves from 2049 through 4096.

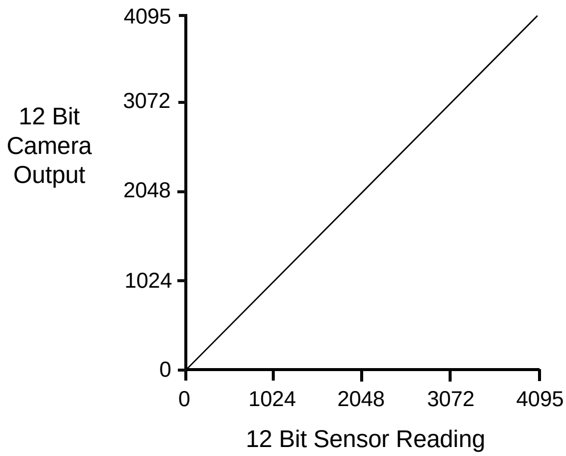


Fig. 61: Lookup Table with Values Mapped in a Linear Fashion

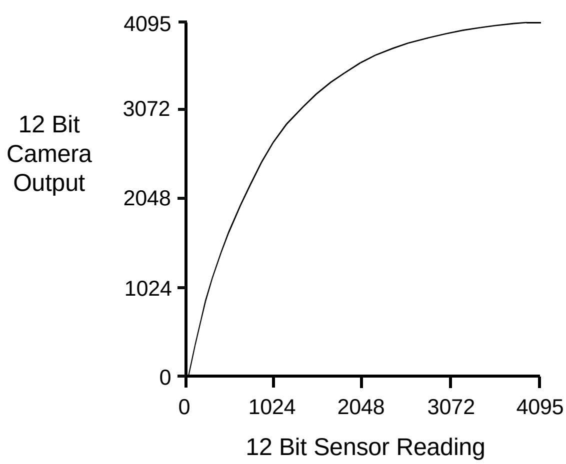


Fig. 62: Lookup Table with Values Mapped for Higher Camera Output at Low Sensor Readings

Using the Luminance Lookup Table to Get 8 Bit Output

As mentioned above, when the camera is set for a pixel format where it outputs 12 effective bits, the lookup table is used to perform a 12 bit to 12 bit conversion. But the lookup table can also be used in 12 bit to 8 bit fashion. To use the table in 12 bit to 8 bit fashion, you enter 12 bit values into the table and enable the table as you normally would. But instead of setting the camera for a pixel format that results in a camera output with 12 bits effective, you set the camera for a pixel format that results in 8 bit output (such as Mono 8, Bayer BG 8, or YUV 4:2:2 Packed). In this situation, the camera will first use the values in the table to do a 12 bit to 12 bit conversion. It will then drop the 4 least significant bits of the converted value and will transmit the 8 most significant bits.

Changing the Values in the Luminance Lookup Table and Enabling the Table

You can change the values in the luminance lookup table (LUT) and enable the use of the lookup table by doing the following:

- Use the LUT Selector to select a lookup table. (Currently there is only one lookup table available, i.e., the "luminance" lookup table described above.)
- Use the LUT Index parameter to select a value in the lookup table. The LUT Index parameter selects the value in the table to change. The index number for the first value in the table is 0, for the second value in the table is 1, for the third value in the table is 2, and so on.
- Use the LUT Value parameter to set the selected value in the lookup table.
- Use the LUT Index parameter and LUT value parameters to set other table values as desired.
- Use the LUT Enable parameter to enable the table.

You can set the LUT Selector, the LUT Index parameter and the LUT Value parameter from within your application software by using the pylon API. The following code snippet illustrates using the API to set the selector and the parameter values:

```
// Select the lookup table
Camera.LUTSelector.SetValue( LUTSelector_Luminance );

// Write a lookup table to the device.
// The following lookup table causes an inversion of the sensor values
// ( bright -> dark, dark -> bright )
for ( int i = 0; i < 4096; i += 8 )
{
   Camera.LUTIndex.SetValue( i );
   Camera.LUTValue.SetValue( 4095 - i );
}
// Enable the lookup table
Camera.LUTEnable.SetValue( true );
```

For detailed information about using the pylon API, refer to the Basler pylon Programmer's Guide and API Reference.

You can also use the Basler pylon Viewer application to easily set the parameters.

For more information about the pylon Viewer, see Section 2.2 on page 32 and Section 2.4 on page 52.

13.6.2 Lookup Table (scA750-60 only)

Note

The information in this section only applies to scA750-60 gm/gc cameras. For information about the other camera models, see Section 13.6.1 on page 212.

On these cameras, pixel data is acquired at 10 bit depth. Before the pixel values are transmitted out of the camera, the two least significant bits are dropped and the pixel data is transmitted at 8 bit depth.

Normally, the 10 bit pixel data reported by the sensor's ADCs is directly used to generate the 8 bit output transmitted by the camera. The luminance lookup table feature lets you use a custom 10 bit to 10 bit lookup table to map the 10 bit output reported by the ADCs to 10 bit values of your choice. The mapped 10 bit values will then be truncated and transmitted by the camera as 8 bit values.

The lookup table is essentially just a list of 1024 values, however, not every value in the table is actually used. If we number the values in the table from 0 through 1023, the table works like this:

- The number at location 0 in the table represents the mapped 10 bit value that will be used when the sensor reports that a pixel has a value of 0.
- The number at location 1 is not used.
- The number at location 2 in the table represents the mapped 10 bit value that will be used when the sensor reports that a pixel has a value of 2.
- The number at location 3 is not used.
- The number at location 4 in the table represents the mapped 10 bit value that will be used when the sensor reports that a pixel has a value of 4.
- The number at location 5 is not used.
- The number at location 6 in the table represents the mapped 10 bit value that will be used when the sensor reports that a pixel has a value of 6.
- And so on.

As you can see, the table does not include a mapped 10 bit output value for every pixel value that the sensor can report. So what does the camera do when the sensor reports a pixel value that is between two values that have a mapped 10 bit output? In this case, the camera performs a straight line interpolation between the two nearest neighbors to determine the value that it should use. For example, assume that the sensor reports a pixel value of 5. In this case, the camera would perform an interpolation between the values at location 4 and location 6 in the table. The result of the interpolation would be used as the mapped 10 bit value.

Another thing to keep in mind about the table is that location 1022 is the last location that will have a mapped 10 bit value associated with it. If the sensor reports a pixel value of 1023, the camera will not be able to perform an interpolation. In this case, the camera simply uses the mapped 10 bit value from location 1022 in the table.

The advantage of the luminance lookup table feature is that it allows a user to customize the response curve of the camera. The graphs on the next page show the effect of two typical lookup tables. The first graph is for a lookup table where the values are arranged so that the output of the camera increases linearly as the sensor output increases. The second graph is for a lookup table where the values are arranged so that the camera output increases quickly as the sensor output moves from 0 through 512 and increases gradually as the sensor output moves from 513 through 1023.

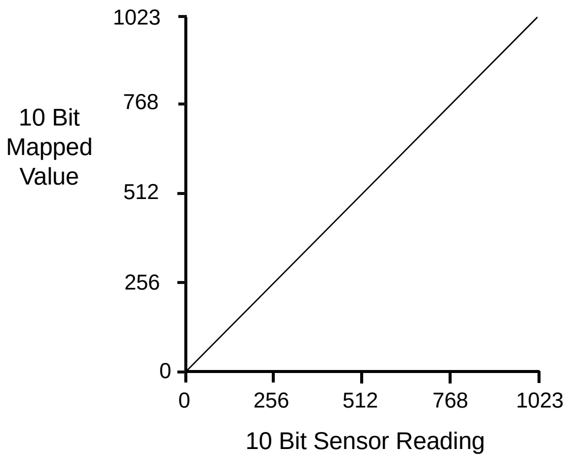


Fig. 63: Lookup Table with Values Mapped in a Linear Fashion

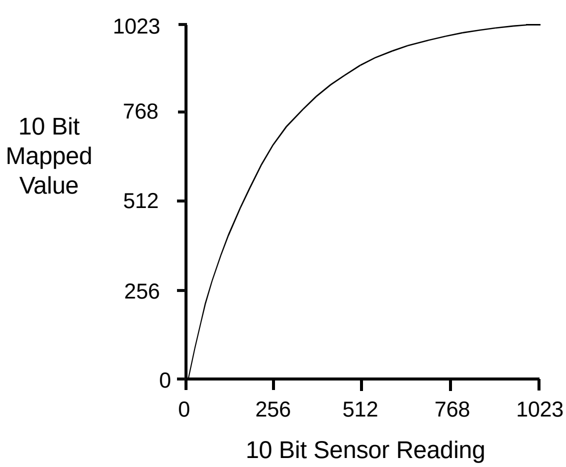


Fig. 64: Lookup Table with Values Mapped for Higher Camera Output at Low Sensor Readings

Changing the Values in the Luminance Lookup Table and Enabling the Table

You can change the values in the luminance lookup table (LUT) and enable the use of the lookup table by doing the following:

- Use the LUT Selector to select a lookup table. (Currently there is only one lookup table available, i.e., the "luminance" lookup table described above.)
- Use the LUT Index parameter to select a value in the lookup table. The LUT Index parameter selects the value in the table to change. The index number for the first value in the table is 0, for the second value in the table is 1, for the third value in the table is 2, and so on.
- Use the LUT Value parameter to set the selected value in the lookup table.
- Use the LUT Index parameter and LUT value parameters to set other table values as desired.
- Use the LUT Enable parameter to enable the table.

You can set the LUT Selector, the LUT Index parameter and the LUT Value parameter from within your application software by using the pylon API. The following code snippet illustrates using the API to set the selector and the parameter values:

```
// Select the lookup table
Camera.LUTSelector.SetValue( LUTSelector_Luminance );

// Write a lookup table to the device.
// The following lookup table causes an inversion of the sensor values
// ( bright -> dark, dark -> bright )
for ( int i = 0; i < 1024; i += 2 )
{
   Camera.LUTIndex.SetValue( i );
   Camera.LUTValue.SetValue( 1023 - i );
}
// Enable the lookup table
Camera.LUTEnable.SetValue( true );
```

For detailed information about using the pylon API, refer to the Basler pylon Programmer's Guide and API Reference.

You can also use the Basler pylon Viewer application to easily set the parameters.

For more information about the pylon Viewer, see Section 2.2 on page 32 and Section 2.4 on page 52.

13.7 Event Reporting

Event reporting is available on the camera. With event reporting, the camera can generate an "event" and transmit it to the PC whenever a specific situation has occurred.

Currently, the camera can generate and transmit an event for two types of situations:

- An end of an exposure has occurred
- An event overrun has occurred

An Example of Event Reporting

As an example of how event reporting works, assume that "end of exposure" event reporting has been enabled in the camera. Also assume that an end of exposure has just occurred in the camera. In this case:

1. An "end of exposure event" is created. The event contains:

   An *Event Type Identifier*. In this case, the identifier would show that an end of exposure type event has occurred.

   A *Stream Channel Identifier*. Currently this identifier is always 0.

   A *Frame ID*. This number indicates the frame count at the time that the event occurred.

   A *Timestamp*. This is a timestamp indicating when the event occurred. (The time stamp timer starts running at power off/on or at camera reset. The unit for the timer is "ticks" where one tick = 8 ns. The timestamp is a 64 bit value.)
2. The event is placed in an internal queue in the camera.
3. As soon as network transmission time is available, the camera will transmit an event message. If only one event is in the queue, the message will contain the single event. If more than one event is in the queue, the message will contain multiple events.
   a. After the camera sends an event message, it waits for an acknowledgement. If no acknowledgement is received within a specified timeout, the camera will resend the event message. If an acknowledgement is still not received, the timeout and resend mechanism will repeat until a specified maximum number of retrys is reached. If the maximum number of retrys is reached and no acknowledge has been received, the message will be dropped.

      During the time that the camera is waiting for an acknowledgement, no new event messages can be transmitted.

The Event Queue

As mentioned in the example above, the camera has an event queue. The intention of the queue is to handle short term delays in the camera’s ability to access the network and send event messages. When event reporting is working "smoothly", a single event will be placed in the queue and this event will be sent to the PC in an event message before the next event is placed in queue. If there is an occasional short term delay in event message transmission, the queue can buffer several events and can send them within a single event message as soon as transmission time is available.

However if you are operating the camera at high frame rates with a small AOI, the camera may be able to generate and queue events faster than they can be transmitted and acknowledged. In this case:

1. The queue will fill and events will be dropped.
2. An event overrun will occur.
3. Assuming that you have event overrun reporting enabled, the camera will generate an "event overrun event" and place it in the queue.
4. As soon as transmission time is available, an event message containing the event overrun event will be transmitted to the PC.

The event overrun event is simply a warning that events are being dropped. The notification contains no specific information about how many or which events have been dropped.

Setting Your System for Event Reporting

To use event reporting, two conditions must be met:

- Event reporting must be enabled in the camera
- A pylon "event grabber" must be created within your application (assuming that you are using the pylon API)

The main purpose of the pylon event grabber is to receive incoming event messages.

Another purpose of the pylon event grabber is to handle event message acknowledgement. The values for the event message timeout and the event message retry count are set via the event grabber.

An event adapter object of the event grabber can be used to parse the information contained within each event message.

You can enable event reporting, create a pylon event grabber, and use the event adapter object from within your application software by using the pylon API. The pylon software development kit includes a "Camera Events" code sample that illustrates the entire process.

For more detailed information about using the pylon API, refer to the Basler pylon Programmer’s Guide and API Reference.

13.8 Test Images

All cameras include the ability to generate test images. Test images are used to check the camera's basic functionality and its ability to transmit an image to the host PC. Test images can be used for service purposes and for failure diagnostics. For test images, the image is generated internally by the camera's logic and does not use the optics, the imaging sensor, or the ADC. Six test images are available.

The Effect of Camera Settings on Test Images

When any of the test image is active, the camera's analog features such as gain, black level, and exposure time have no effect on the images transmitted by the camera. For test images 1, 2, 3 and 6, the cameras digital features, such as the luminance lookup table, will also have no effect on the transmitted images. But for test images 4 and 5, the cameras digital features will affect the images transmitted by the camera. This makes test images 4 and 5 as good way to check the effect of using a digital feature such as the luminance lookup table.

Enabling a Test Image

The Test Image Selector is used to set the camera to output a test image. You can set the value of the Test Image Selector to one of the test images or to "test image off".

You can set the Test Image Selector from within your application software by using the pylon API. The following code snippets illustrate using the API to set the selector:

```
// set for no test image
Camera.TestImageSelector.SetValue( TestImageSelector_Off );

// set for the first test image
Camera.TestImageSelector.SetValue( TestImageSelector_Testimage1 );
```

For detailed information about using the pylon API, refer to the Basler pylon Programmer's Guide and API Reference.

You can also use the Basler pylon Viewer application to easily set the parameters.

For more information about the pylon Viewer, see Section 2.2 on page 32and Section 2.4 on page 52.

Test Image 1 - Fixed Diagonal Gray Gradient (8 bit)

The 8 bit fixed diagonal gray gradient test image is best suited for use when the camera is set for monochrome 8 bit output. The test image consists of fixed diagonal gray gradients ranging from 0 to 255.

If the camera is set for 8 bit output and is operating at full resolution, test image one will look similar to Figure 65.

The mathematical expression for this test image:

Gray Value = [column number + row number] MOD 256

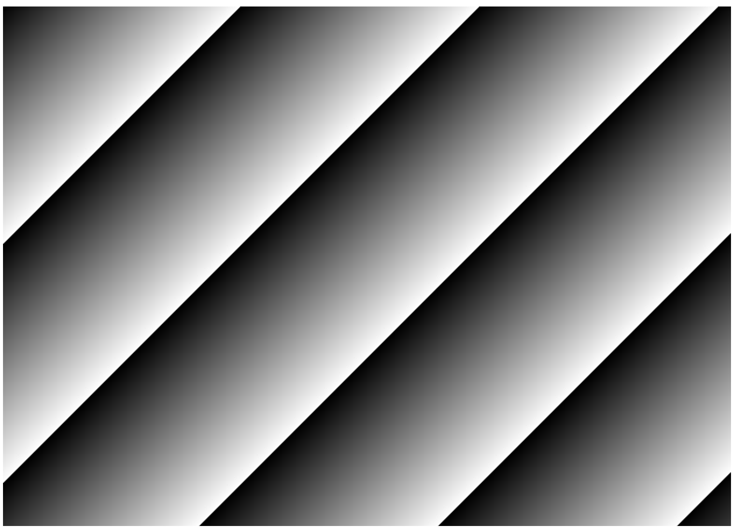


Fig. 65: Test Image One

Test Image 2 - Moving Diagonal Gray Gradient (8 bit)

The 8 bit moving diagonal gray gradient test image is similar to test image 1, but it is not stationary. The image moves by one pixel from right to left whenever a new image acquisition is initiated. The test pattern uses a counter that increments by one for each new image acquisition.

The mathematical expression for this test image is:

Gray Value = [column number + row number + counter] MOD 256

Test Image 3 - Moving Diagonal Gray Gradient (12 bit)

The 12 bit moving diagonal gray gradient test image is similar to test image 2, but it is a 12 bit pattern. The image moves by one pixel from right to left whenever a new image acquisition is initiated. The test pattern uses a counter that increments by one for each new image acquisition.

The mathematical expression for this test image is:

Gray Value = [column number + row number + counter] MOD 4096

> **Note**
>
> On scA750-60 cameras, test image 3 is a 10 bit pattern. Since these cameras do not have a 10 bit output mode available, use of test image 3 on scA750-60 cameras is not recommended.

Test Image 4 - Moving Diagonal Gray Gradient Feature Test (8 bit)

The basic appearance of test image 4 is similar to test image 2 (the 8 bit moving diagonal gray gradient image). The difference between test image 4 and test image 2 is this: if a camera feature that involves digital processing is enabled, test image 4 **will** show the effects of the feature while test image 2 **will not**. This makes test image 4 useful for checking the effects of digital features such as the luminance lookup table.

Test Image 5 - Moving Diagonal Gray Gradient Feature Test (12 bit)

The basic appearance of test image 5 is similar to test image 3 (the 12 bit moving diagonal gray gradient image). The difference between test image 5 and test image 3 is this: if a camera feature that involves digital processing is enabled, test image 5 **will** show the effects of the feature while test image 3 **will not**. This makes test image 5 useful for checking the effects of digital features such as the luminance lookup table.

> **Note**
>
> On scA750-60 cameras, test image 5 is a 10 bit pattern. Since these cameras do not have a 10 bit output mode available, use of test image 5 on scA750-60 cameras is not normally recommended. However, one situation where test image 5 is useful on scA750-60 cameras is to check the effect of the luminance lookup table.

Test Image 6 - Moving Diagonal Color Gradient

The moving diagonal color gradient test image is available on color cameras only and is designed for use when the camera is set for YUV output. As shown in Figure 66, test image six consists of diagonal color gradients. The image moves by one pixel from right to left whenever you signal the camera to capture a new image. To display this test pattern on a monitor, you must convert the YUV output from the camera to 8 bit RGB.

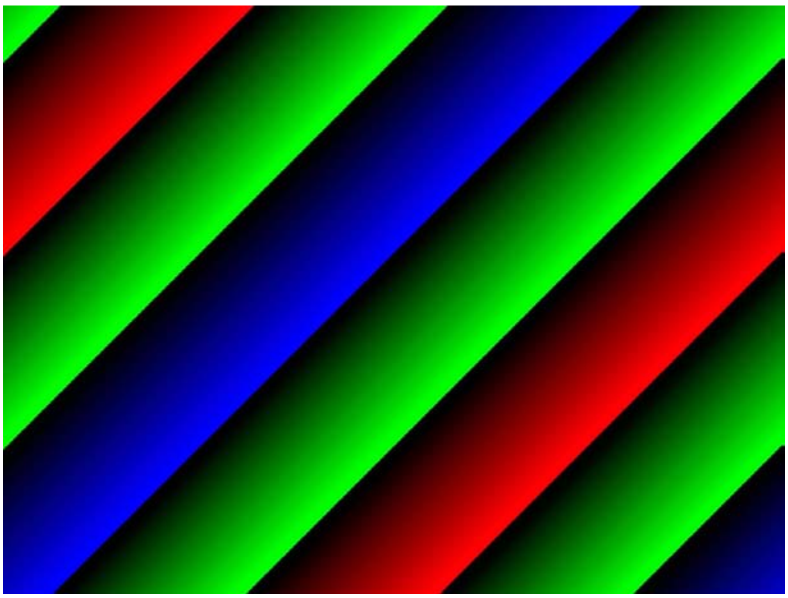


Fig. 66: Test Image Six

13.9 Device Information Parameters

Each camera includes a set of "device information" parameters. These parameters provide some basic information about the camera. The device information parameters include:

- Device Vendor Name (read only) - contains the name of the camera's vendor. For scout cameras, this string will always indicate Basler as the vendor.
- Device Model Name (read only) - contains the model name of the camera, for example, scA640-74gm.
- Device Manufacturer Info (read only) - can contain some information about the camera manufacturer. On scout cameras, this string is usually empty.
- Device Version (read only) - contains the device version number for the camera.
- Firmware Version (read only) - contains the version of the firmware in the camera.
- Device ID (read only) - contains the serial number of the camera.
- Device User ID (read / write) - is used to assign a user defined name to a device. This name will be displayed in the Basler pylon Viewer and the Basler pylon IP Configuration Tool. The name will also be visible in the "friendly name" field of the device information objects returned by pylon's device enumeration procedure.
- Device Scan Type (read only) - contains the scan type of the camera, for example, area scan.
- Sensor Width (read only) - contains the physical width of the sensor in pixels.
- Sensor Height (read only) - contains the physical height of the sensor.
- Max Width (read only) - Indicates the camera's maximum area of interest (AOI) width setting.
- Max Height (read only) - Indicates the camera's maximum area of interest (AOI) height setting.

You can read the values for all of the device information parameters or set the value of the Device User ID parameter from within your application software by using the pylon API. The following code snippets illustrate using the API to read the parameters or write the Device User ID:

```
// Read the Vendor Name parameter
Pylon::String_t vendorName = Camera.DeviceVendorName.GetValue();

// Read the Model Name parameter
Pylon::String_t modelName = Camera.DeviceModelName.GetValue();

// Read the Manufacturer Info parameter
Pylon::String_t manufacturerInfo = Camera.DeviceManufacturerInfo.GetValue();

// Read the Device Version parameter
Pylon::String_t deviceVersion = Camera.DeviceVersion.GetValue();

// Read the Firmware Version parameter
Pylon::String_t firmwareVersion = Camera.DeviceFirmwareVersion.GetValue();
```

```
// Read the Device ID parameter
Pylon::String_t deviceID = Camera.DeviceFirmwareVersion.GetValue();

// Write and read the Device User ID
Camera.DeviceUserID = "custom name";
Pylon::String_t deviceUserID = Camera.DeviceUserID.GetValue();

// Read the Sensor Width parameter
int64_t sensorWidth = Camera.SensorWidth.GetValue();

// Read the Sensor Height parameter
int64_t sensorHeight = Camera.SensorHeight.GetValue();

// Read the Max Width parameter
int64_t maxWidth = Camera.WidthMax.GetValue();

// Read the Max Height parameter
int64_t maxHeight = Camera.HeightMax.GetValue();
```

For detailed information about using the pylon API, refer to the Basler pylon Programmer's Guide and API Reference.

You can also use the Basler pylon Viewer application to easily read the parameters and to read or write the Device User ID.

You can use the Basler pylon IP Configuration tool to read or write the Device User ID.

For more information about the pylon Viewer, see Section 2.2 on page 32and Section 2.4 on page 52.

For more information about the pylon IP Configuration Tool, see Section 5.4 on page 81.

13.10 Configuration Sets

A configuration set is a group of values that contains all of the parameter settings needed to control the camera. There are three basic types of configuration sets: the active configuration set, the default configuration set, and user configurations sets.

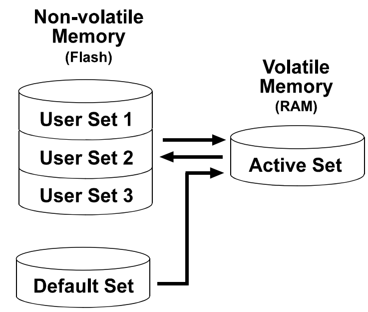


Fig. 67: Configuration Sets

Active Configuration Set

The active configuration set contains the camera's current parameter settings and thus determines the camera's performance, that is, what your image currently looks like. When you change parameter settings using the pylon API or the pylon Viewer, you are making changes to the active configuration set. The active configuration set is located in the camera's volatile memory and the settings are lost if the camera is reset or if power is switched off. The active configuration set is usually called the "active set" for short.

Default Configuration Set

When a camera is manufactured, a test setup is performed on the camera and an optimized configuration is determined. The default configuration set contains the camera's factory optimized configuration. The default configuration set is saved in a permanent file in the camera's non-volatile memory. It is not lost when the camera is reset or switched off and it cannot be changed. The default configuration set is usually just called the "default set" for short.

User Configuration Sets

As mentioned above, the active configuration set is stored in the camera's volatile memory and the settings are lost if the camera is reset or if power is switched off. The camera can save most of the settings from the current active set to a reserved area in the camera's non-volatile memory. A configuration set saved in the non-volatile memory is not lost when the camera is reset or switched off. There are three reserved areas in the camera's non-volatile memory available for saving configuration sets. A configuration set saved in a reserved area is commonly referred to as a "user configuration set" or "user set" for short.

The three available user sets are called User Set 1, User Set 2, and User Set 3.

Note

The settings for frame transmission delay, inter packet delay, and the luminance lookup table are not saved in the user sets and are lost when the camera is reset or switched off. If used, these settings must be set again after each camera reset or restart.

Default Startup Set

You can select the default configuration set or one of the user configuration sets stored in the camera's non-volatile memory to be the "default startup set." The configuration set that you designate as the default startup set will be loaded into the active set whenever the camera starts up at power on or after a reset. Instructions for selecting the default startup set appear on the next page.

13.10.1 Saving Configuration Sets

Saving the current active set into the camera's non-volatile memory is a three step process:

- Make changes to the camera's settings until the camera is operating in a manner that you would like to save.
- Set the User Set Selector to User Set 1, User Set 2, or User Set 3.
- Execute a User Set Save command to save the active set to the selected user set.

Saving an active set to a user set in the camera's non-volatile memory will overwrite any parameters that were previously saved in that user set.

You can set the User Set Selector and execute the User Set Save command from within your application software by using the pylon API. The following code snippet illustrates using the API to set the selector and execute the command:

```
Camera.UserSetSelector.SetValue( UserSetSelector_UserSet1 );
Camera.UserSetSave.Execute( );
```

For detailed information about using the pylon API, refer to the Basler pylon Programmer's Guide and API Reference.

You can also use the Basler pylon Viewer application to easily set the parameters.

For more information about the pylon Viewer, see Section 2.2 on page 32 and Section 2.4 on page 52.

13.10.2 Loading a Saved Set or the Default Set into the Active Set

If you have saved a configuration set into the camera's non-volatile memory, you can load the saved set from the camera's non-volatile memory into the camera's active set. When you do this, the loaded set overwrites the parameters in the active set. Since the settings in the active set control the current operation of the camera, the settings from the loaded set will now be controlling the camera.

You can also load the default set into the camera's active set.

To load a saved configuration set or the default set from the camera's non-volatile memory into the active set:

- Set the User Set Selector to User Set 1, User Set 2, User Set 3, or Default.
- Execute a User Set Load command to load the selected set into the active set.

You can set the User Set Selector and execute the User Set Load command from within your application software by using the pylon API. The following code snippet illustrates using the API to set the selector and execute the command:

```
Camera.UserSetSelector.SetValue( UserSetSelector_UserSet2 );
Camera.UserSetLoad.Execute( );
```

Note

Loading a user set or the default set into the active set is only allowed when the camera is idle, i.e. when it is not acquiring images continuously or does not have a single image acquisition pending.

Loading the default set into the active set is a good course of action if you have grossly misadjusted the settings in the camera and you are not sure how to recover. The default settings are optimized for use in typical situations and will provide good camera performance in most cases.

13.10.3 Selecting the Default Startup Set

You can select the default configuration set or one of the user configuration sets stored in the camera's non-volatile memory to be the "default startup set". The configuration set that you designate as the default startup set will be loaded into the active set whenever the camera starts up at power on or after a reset.

The User Set Default Selector is used to select the default startup set:

- Set the User Set Default Selector to User Set 1, User Set 2, User Set 3, or Default.

You can set the User Set Default Selector from within your application software by using the pylon API. The following code snippet illustrates using the API to set the selector:

```
Camera.UserSetDefaultSelector.SetValue( UserSetDefaultSelector_Default );
```

14 Chunk Features

This section provides detailed information about the chunk features available on each camera.

14.1 What are Chunk Features?

In most cases, enabling a camera feature will simply change the behavior of the camera. The Test Image feature is a good example of this type of camera feature. When the Test Image feature is enabled, the camera outputs a test image rather than a captured image. This type of feature is referred to as a "standard" feature.

When certain camera features are enabled, the camera actually develops some sort of information about each image that it acquires. In these cases, the information is added to each image as a trailing data "chunk" when the image is transferred to the host PC. Examples of this type of camera feature are the Frame Counter feature and the Time Stamp feature. When the Frame Counter feature is enabled, for example, after an image is captured, the camera checks a counter that tracks the number of images acquired and develops a frame counter stamp for the image. And if the Time Stamp feature is enabled, the camera creates a time stamp for the image. The frame counter stamp and the time stamp would be added as "chunks" of trailing data to each image as the image is transferred from the camera. The features that add chunks to the acquired images are referred to as “chunk” features.

Before you can use any of the features that add chunks to the image, you must make the chunk mode active. Making the chunk mode active is described in the next section.

14.2 Making the "Chunk Mode" Active and Enabling the Extended Data Stamp

Before you can use any of the camera’s "chunk" features, the "chunk mode" must be made active. Making the chunk mode active does two things:

- It makes the Frame Counter, the Time Stamp, and the Line Status All chunk features available to be enabled.
- It automatically enables the Extended Image Data chunk feature.

To make the chunk mode active:

- Set the Chunk Mode Active parameter to true.

You can set the Chunk Mode Active parameter value from within your application software by using the pylon API. The following code snippet illustrates using the API to set the parameter value:

```
Camera.ChunkModeActive.SetValue( true );
```

Note that making the chunk mode inactive switches all chunk features off.

Also note that when you enable ChunkModeActive, the PayloadType for the camera changes from "Pylon::PayloadType_Image" to "Pylon::PayloadType_ChunkData".

For detailed information about using the pylon API, refer to the Basler pylon Programmer’s Guide and API Reference.

You can also use the Basler pylon Viewer application to easily set the parameters.

Once the chunk mode is active and the Extended Image Data feature has been enabled, the camera will automatically add an "extended image data" chunk to each acquired image. The extended image data chunk appended to each acquired image contains some basic information about the image. The information contained in the chunk includes:

- The X Offset, Y Offset, Width, and Height for the AOI
- The Pixel Format of the image
- The Minimum Dynamic Range and the Maximum Dynamic Range

To retrieve data from the extended image data chunk appended to an image that has been received by your PC, you must first run the image and its appended chunks through the chunk parser included in the pylon API. Once the chunk parser has been used, you can retrieve the extended image data by doing the following:

- Read the value of the Chunk Offset X parameter.
- Read the value of the Chunk Offset Y parameter.
- Read the value of the Chunk Width parameter.
- Read the value of the Chunk Height parameter.
- Read the value of the Chunk Pixel Format parameter.
- Read the value of the Chunk Dynamic Range Min.
- Read the value of the Chunk Dynamic Range Max.

The following code snippet illustrates using the pylon API to run the parser and retrieve the extended image data:

```
// retrieve date from the extended image data chunk
IChunkParser &ChunkParser = *Camera.CreateChunkParser();
GrabResult Result;
StreamGrabber.RetrieveResult( Result );
ChunkParser.AttachBuffer( (unsigned char*) Result.Buffer(),
  Result.GetPayloadSize() );
int64_t offsetX = Camera.ChunkOffsetX.GetValue();
int64_t offsetY = Camera.ChunkOffsetY.GetValue();
int64_t width = Camera.ChunkWidth.GetValue();
int64_t height = Camera.ChunkHeight.GetValue();
int64_t dynamicRangeMin = Camera.ChunkDynamicRangeMin.GetValue();
int64_t dynamicRangeMax = Camera.ChunkDynamicRangeMax.GetValue();
ChunkPixelFormatEnums pixelFormat = Camera.ChunkPixelFormat.GetValue();
```

For more information about using the chunk parser, see the sample code that is included with the Basler scout Software Development Kit (SDK).

14.3 Frame Counter

The Frame Counter feature numbers images sequentially as they are acquired. When the feature is enabled, a chunk is added to each image containing the value of the counter.

The frame counter is a 32 bit value. The counter starts at 0 and wraps at 4294967296. The counter increments by 1 for each acquired image. Whenever the camera is powered off, the counter will reset to 0.

Be aware that if the camera is acquiring images continuously and continuous capture is stopped, several numbers in the counting sequence may be skipped. This happens due to the internal image buffering scheme used in the camera.

Note

The chunk mode must be active before you can enable the frame counter feature or any of the other chunk feature. Making the chunk mode inactive disables all chunk features.

To enable the frame counter chunk:

- Use the Chunk Selector to select the Frame Counter chunk.
- Use the Chunk Enable parameter to set the value of the chunk to true.

Once the frame counter chunk is enabled, the camera will add a frame counter chunk to each acquired image.

To retrieve data from a chunk appended an image that has been received by your PC, you must first run the image and its appended chunks through the chunk parser included in the pylon API. Once the chunk parser has been used, you can retrieve the frame counter information by doing the following:

- Read the value of the Chunk Frame Counter parameter.

You can set the Chunk Selector and Chunk Enable parameter value from within your application software by using the pylon API. You can also run the parser and retrieve the chunk data. The following code snippets illustrate using the API to activate the chunk mode, enable the frame counter chunk, run the parser, and retrieve the frame counter chunk data:

```
// make chunk mode active and enable Frame Counter chunk
Camera.ChunkModeActive.SetValue( true );
Camera.ChunkSelector.SetValue( ChunkSelector_Framecounter );
Camera.ChunkEnable.SetValue( true );

// retrieve date from the chunk
IChunkParser &ChunkParser = *Camera.CreateChunkParser();
GrabResult Result;
StreamGrabber.RetrieveResult( Result );
ChunkParser.AttachBuffer( (unsigned char*) Result.Buffer(),
```

```
    Result.GetPayloadSize() );
int64_t frameCounter = Camera.ChunkFramecounter.GetValue();
```

For detailed information about using the pylon API, refer to the Basler pylon Programmer's Guide and API Reference.

You can also use the Basler pylon Viewer application to easily set the parameters.

For more information about the pylon Viewer, see Section 2.2 on page 32 and Section 2.4 on page 52.

14.4 Time Stamp

The Time Stamp feature adds a chunk to each acquired image containing a time stamp that was generated when image acquisition was triggered.

The time stamp is a 64 bit value. The time stamp is based on a counter that counts the number of "time stamp clock ticks" generated by the camera. The unit for each tick is 8 ns (as specified by the Gev Timestamp Tick Frequency). The counter starts at camera reset or at power off/on.

Note

The chunk mode must be active before you can enable the time stamp feature or any of the other chunk feature. Making the chunk mode inactive disables all chunk features.

To enable the time stamp chunk:

- Use the Chunk Selector to select the Time Stamp chunk.
- Use the Chunk Enable parameter to set the value of the chunk to true.

Once the time stamp chunk is enabled, the camera will add a time stamp chunk to each acquired image.

To retrieve data from a chunk appended to an image that has been received by your PC, you must first run the image and its appended chunks through the chunk parser that is included in the pylon API. Once the chunk parser has been used, you can retrieve the time stamp information by doing the following:

- Read the value of the Chunk Time Stamp parameter.

You can set the Chunk Selector and Chunk Enable parameter value from within your application software by using the pylon API. You can also run the parser and retrieve the chunk data. The following code snippets illustrate using the API to activate the chunk mode, enable the time stamp chunk, run the parser, and retrieve the frame counter chunk data:

```
// make chunk mode active and enable Time Stamp chunk
Camera.ChunkModeActive.SetValue( true );
Camera.ChunkSelector.SetValue( ChunkSelector_Timestamp );
Camera.ChunkEnable.SetValue( true );

// retrieve data from the chunk
IChunkParser &ChunkParser = *Camera.CreateChunkParser();
GrabResult Result;
StreamGrabber.RetrieveResult( Result );
ChunkParser.AttachBuffer( (unsigned char*) Result.Buffer(),
   Result.GetPayloadSize() );
int64_t timeStamp = Camera.ChunkTimestamp.GetValue();
```

For detailed information about using the pylon API, refer to the Basler pylon Programmer's Guide and API Reference.

You can also use the Basler pylon Viewer application to easily set the parameters.

For more information about the pylon Viewer, see Section 2.2 on page 32 and Section 2.4 on page 52.

14.5 Line Status All

The Line Status All feature samples the status of all of the camera's input lines and output lines each time an image acquisition is triggered. It then adds a chunk to each acquired image containing the line status information.

The line status all information is a 32 bit value. As shown in Figure 68, certain bits in the value are associated with each line and the bits will indicate the state of the lines. If a bit is 0, it indicates that the state of the associated line was low at the time of triggering. If a bit is 1, it indicates that the state of the associated line is was high at the time of triggering.

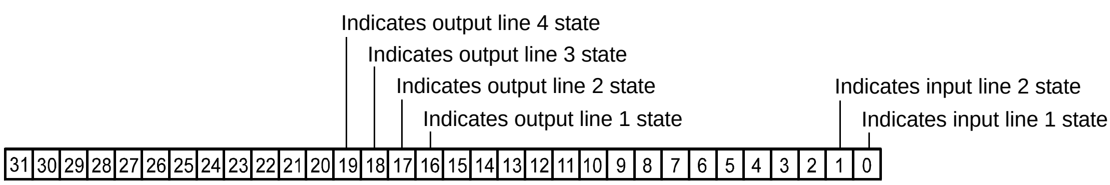


Fig. 68: Line Status All Parameter Bits

Note

The chunk mode must be active before you can enable the line status all feature or any of the other chunk feature. Making the chunk mode inactive disables all chunk features.

To enable the line status all chunk:

- Use the Chunk Selector to select the Line Status All chunk.
- Use the Chunk Enable parameter to set the value of the chunk to true.

Once the line status all chunk is enabled, the camera will add a line status all chunk to each acquired image.

To retrieve data from a chunk appended to an image that has been received by your PC, you must first run the image and its appended chunks through the chunk parser included in the pylon API. Once the chunk parser has been used, you can retrieve the line status all information by doing the following:

- Read the value of the Chunk Line Status All parameter.

You can set the Chunk Selector and Chunk Enable parameter value from within your application software by using the pylon API. You can also run the parser and retrieve the chunk data. The following code snippets illustrate using the API to activate the chunk mode, enable the line status all chunk, run the parser, and retrieve the line status all chunk data:

```
// make chunk mode active and enable Line Status All chunk
Camera.ChunkModeActive.SetValue( true );
Camera.ChunkSelector.SetValue( ChunkSelector_LineStatusAll );
```

```
Camera.ChunkEnable.SetValue( true );

// retrieve data from the chunk
IChunkParser &ChunkParser = *Camera.CreateChunkParser();
GrabResult Result;
StreamGrabber.RetrieveResult( Result );
ChunkParser.AttachBuffer( (unsigned char*) Result.Buffer(),
   Result.GetPayloadSize() );
int64_t lineStatusAll = Camera.ChunkLineStatusAll.GetValue();
```

For detailed information about using the pylon API, refer to the Basler pylon Programmer's Guide and API Reference.

You can also use the Basler pylon Viewer application to easily set the parameters.

For more information about the pylon Viewer, see Section 2.2 on page 32 and Section 2.4 on page 52.

14.6 CRC Checksum

The CRC (Cyclic Redundancy Check) Checksum feature adds a chunk to each acquired image containing a CRC checksum calculated using the Z-modem method. As shown in Figure 6-2, the checksum is calculated using all of the image data and all of the appended chunks except for the checksum itself. The CRC chunk is always the last chunk appended to the image data.

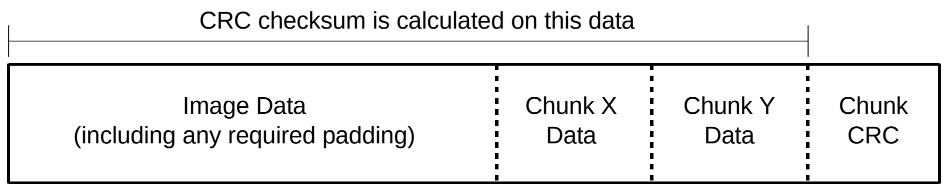


Fig. 69: CRC Checksum

Note

The chunk mode must be active before you can enable the CRC feature or any of the other chunk feature. Making the chunk mode inactive disables all chunk features.

To enable the CRC checksum chunk:

- Use the Chunk Selector to select the CRC chunk.
- Use the Chunk Enable parameter to set the value of the chunk to true.

Once the CRC chunk is enabled, the camera will add a CRC chunk to each acquired image.

To retrieve CRC information from a chunk appended to an image that has been received by your PC, you must first run the image and its appended chunks through the chunk parser included in the pylon API. Once the chunk parser has been used, you can retrieve the CRC information. Note that the CRC information provided by the chunk parser is not the CRC checksum itself. Rather it is a true/false result. When the image and appended chunks pass through the parser, the parser calculates a CRC checksum based on the received image and chunk information. It then compares the calculated CRC checksum with the CRC checksum contained in the CRC checksum chunk. If the two match, the result will indicate that the image data is OK. If the two do not match, the result will indicate that the image is corrupted.

You can set the Chunk Selector and Chunk Enable parameter value from within your application software by using the pylon API. You can also run the parser and retrieve the chunk data. The following code snippets illustrate using the API to activate the chunk mode, enable the time stamp chunk, run the parser, and retrieve the frame counter chunk data:

```
// Make chunk mode active and enable CRC chunk
Camera.ChunkModeActive.SetValue( true );
```

```
Camera.ChunkSelector.SetValue( ChunkSelector_PayloadCRC16 );
Camera.ChunkEnable.SetValue( true );

// Check the CRC checksum of an grabbed image
IChunkParser &ChunkParser =
  *Camera.CreateChunkParser();
GrabResult Result;
StreamGrabber.RetrieveResult( Result );
ChunkParser.AttachBuffer( (unsigned char*) Result.Buffer(),
  Result.GetPayloadSize() );
if ( ChunkParser.HasCRC() && ! ChunkParser.CheckCRC() )
  cerr << "Image corrupted!" << endl;
```

For detailed information about using the pylon API, refer to the Basler pylon Programmer's Guide and API Reference.

You can also use the Basler pylon Viewer application to easily set the parameters.

For more information about the pylon Viewer, see Section 2.2 on page 32 and Section 2.4 on page 52.

15 Troubleshooting and Support

This section outlines the resources available to you if you need help working with your camera.

15.1 Tech Support Resources

If you need advice about your camera or if you need assistance troubleshooting a problem with your camera, you can contact the Basler technical support team for your area. Technical support contact information is located in the front pages of this manual.

You will also find helpful information such as frequently asked questions, downloads, and technical notes at our website: www.basler-vc.com.

If you do decide to contact Basler technical support, please take a look at the form that appears on the last two pages of this section before you call. Filling out this form will help make sure that you have all of the information the Basler technical support team needs to help you with your problem.

15.2 Before Contacting Basler Technical Support

To help you as quickly and efficiently as possible when you have a problem with a Basler camera, it is important that you collect several pieces of information before you contact technical support.

Copy the form that appears on the next two pages, fill it out, and fax the pages to your local dealer or to your nearest Basler support center. Or, you can send an e-mail listing the requested pieces of information and with the requested files attached. Our technical support contact information is shown in the title section of this manual.

1 The camera's product ID: ______________________

2 The camera's serial number: ______________________

3 Network adapter that you use with the camera: ______________________

4 Describe the problem in as much detail as possible: ______________________
(If you need more space, use an extra sheet of paper.)

5 If known, what's the cause of the problem? ______________________

6 When did the problem occur?
- ☐ After start.
- ☐ While running.
- ☐ After a certain action (e.g., a change of parameters): ______________________
- ☐ ______________________

7 How often did/does the problem occur?
- ☐ Once.
- ☐ Every time.
- ☐ Regularly when: ______________________
- ☐ Occasionally when: ______________________

8 How severe is the problem?

- ☐ Camera can still be used.
- ☐ Camera can be used after I take this action:

- ☐ Camera can no longer be used.

9 Did your application ever run without problems?

- ☐ Yes
- ☐ No

10 Parameter set

It is very important for Basler Technical Support to get a copy of the exact camera parameters that you were using when the problem occurred.

To make note of the parameters, use Basler's "The Viewer" tool.

If you cannot access the camera, please try to state the following parameter settings:

- ☐ Image Size (AOI):
- ☐ Pixel Format:
- ☐ Packet Size:
- ☐ Exposure Time:
- ☐ Frame rate:

11 Live image/test image

If you are having an image problem, try to generate and save live images that show the problem. Also generate and save test images. Please save the images in BMP format, zip them, and send them to Basler technical support.

Revision History

Doc. ID Number	Date	Changes
AW00011901000	13 Sep 2006	Initial release. This release is a preliminary version of the document.
AW00011902000	6 Nov 2006	Added information for the newly released scA1000-20gm/gc camera. Updated all affected text and tables as appropriate. Added information about the minimum exposure times setting to Section 10.4 on page 139. Corrected the numbering in the drawing in Section 13.5 on page 209. Added information about the luminance lookup table feature in Section 13.6 on page 212.
AW00011903000	10 Jan 2007	Improved the content that describes network configuration and IP configuration. Updated the instalation procedures to reflect the pylon 0.9 release.
AW00011904000	20 Mar 2007	Updated the camera weights in the specification table in Section 1.2 on page 2. Added the dimensions for cameras equipped with CS-mount lens adapters to Section 1.5 on page 18. Updated the description of the camera's network related parameters in Section 7.1 on page 97. Added Section 13.7 on page 218 to describe the new event reporting standard feature. Added Section 14.6 on page 238 to describe the new CRC checksum chunk feature. Added information for the newly released scA750-60gm/gc camera. Updated all affected text and tables as appropriate.
AW00011905000	8 Jun 2007	Added information on IP30, Section 1.5.1.1 on page 18 and Section 1.5.2.1 on page 23. Added information (drawings inclusive) on the 90° head housing variant: Section 1.5.2 on page 23. Added information on mechanical stress test results: Section 1.5.4 on page 27. Modified Section 2 for the installation of the Basler pylon software, version 1.0. Added information on the 8 bits effective pixel data: Section 11.2.1 on page 161, Section 11.2.4 on page 167, Section 11.2.5 on page 167 and Section 11.3.3 on page 173. Minor modifications througout the manual: modified frame height to AOI height.

Feedback

Your feedback will help us improve our documentation. Please click the link below to access an online feedback form. Your input is greatly appreciated.

http://www.baslerweb.com/umfrage/survey.html

Index

A

B

C

D

M

N

O

P

R

S