Basler scout

USER'S MANUAL FOR GigE VISION CAMERAS

Document Number: AW000119
Version: 18 Language: 000 (English)
Release Date: 23 January 2015

BASLER
the power of sight

For customers in the USA

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 B 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 463 515
Fax +49 4102 463 599

support.europe@baslerweb.com

The Americas

Basler, Inc.
855 Springdale Drive, Suite 203
Exton, PA 19341
USA

Tel. +1 610 280 0171
Fax +1 610 280 7608

support.usa@baslerweb.com

Asia

Basler Asia Pte. Ltd.
35 Marsiling Industrial Road 3
05 - 06
Singapore 739257

Tel. +65 6367 1355
Fax +65 6367 1255

support.asia@baslerweb.com

www.baslerweb.com

Table of Contents

1   Specifications, Requirements, and Precautions

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

This chapter 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	scA640-70 gm/gc	scA640-74 gm/gc	scA640-120 gm/gc
Sensor Size (H x V pixels)	gm: 659 x 494 gc: 658 x 492	gm: 659 x 494 gc: 658 x 492	gm: 659 x 494 gc: 658 x 492
Sensor Type	Sony ICX424 AL/AQ	Sony ICX414 AL/AQ	Sony ICX618 ALA/AQA
	Progressive scan CCD		
Optical Size	1/3"	1/2"	1/4"
Pixel Size	7.4 µm x 7.4 µm	9.9 µm x 9.9 µm	5.6 µm x 5.6 µm
Max. Frame Rate (at full resolution)	70 fps	79 fps	122 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 RGB 8 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		
Camera Power Requirements	+12 VDC (-5 %) to +24 VDC (+5 %), < 1% ripple		
	3.0 W @ 12 V		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	scA640-70 gm/gc	scA640-74 gm/gc	scA640-120 gm/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 (*), RoHS, GenICam, GigE Vision, IP30, FCC * The CE Conformity Declaration is available on the Basler website: www.baslerweb.com		
Software	Basler pylon 4 Camera Software Suite (version 4.0 or higher) Available for Windows in 32 and 64 bit versions and Linux (x86 32 bit, x86 64 bit, ARM softfloat, ARM hardfloat).		

Table 1: General Specifications

Specification	scA750-60 gm/gc	scA780-54 gm/gc
Sensor Size (H x V pixels)	gm: 752 x 480 gc: 750 x 480	gm: 782 x 582 gc: 780 x 580
Sensor Type	Aptina MT9V022 (formerly known as the Micron MT9V022)	Sony ICX415 AL/AQ
	Progressive scan CMOS	Progressive scan CCD
Optical Size	1/3"	1/2"
Pixel Size	6.0 µm x 6.0 µm	8.3 µm x 8.3 µm
Max. Frame Rate (at full resolution)	64.9 fps	55 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) YUV 4:2:2 Packed (= DCAM YUV 4:2:2) YUV 4:2:2 (YUYV) Packed	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 RG 8 (= DCAM Raw 8) RGB 8 Packed YUV 4:2:2 Packed (= DCAM YUV 4:2:2) YUV 4:2:2 (YUYV) Packed	Mono 8 (= DCAM Mono 8) Bayer BG 8 (= DCAM Raw 8) Bayer BG 16 (= DCAM Raw 16) Bayer BG 12 Packed RGB 8 Packed YUV 4:2:2 Packed (= DCAM YUV 4:2:2) YUV 4:2:2 (YUYV) Packed
ADC Bit Depth	10 bits	12 bits
Synchronization	Via external trigger signal or via software	
Exposure Control	Programmable via the camera API	
Camera Power Requirements	+12 VDC (-5 %) to +24 VDC (+5 %), < 1% ripple	
	2.5 W @ 12 V	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	scA750-60 gm/gc	scA780-54 gm/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 front module and connectors) 97 mm x 44 mm x 41.8 mm (with front module and connectors)	
Weight (standard housing)	160 g (typical)	
(90° head housing)	180 g (typical)	
Conformity	CE (*), RoHS, GenICam, GigE Vision, IP30, FCC * The CE Conformity Declaration is available on the Basler website: www.baslerweb.com	
Software	Basler pylon 4 Camera Software Suite (version 4.0 or higher) Available for Windows in 32 and 64 bit versions and Linux (x86 32 bit, x86 64 bit, ARM softfloat, ARM hardfloat).	

Table 2: General Specifications

Specification	scA1000-30 gm/gc	scA1300-32 gm/gc	scA1390-17 gm/gc
Sensor Size (H x V pixels)	gm: 1034 x 779 gc: 1032 x 778	gm: 1296 x 966 gc: 1294 x 964	gm: 1392 x 1040 gc: 1390 x 1038
Sensor Type	Sony ICX204 AL/AK	Sony ICX445 ALA/AQA	Sony ICX267 AL/AK
	Progressive scan CCD		
Optical Size	1/3"	1/3"	1/2"
Pixel Size	4.65 µm x 4.65 µm	3.75 µm x 3.75 µm	4.65 µm x 4.65 µm
Max. Frame Rate (at full resolution)	31 fps	32 fps	17 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 RGB 8 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		
Camera Power Requirements	+12 VDC (-5 %) to +24 VDC (+5 %), < 1% ripple		
	3.0 W @ 12 V	3.5 W @ 12 V	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	scA1000-30 gm/gc	scA1300-32 gm/gc	scA1390-17 gm/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)	Not applicable	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)	160 g (typical)	160 g (typical)
(90° head housing)	180 g (typical)	Not applicable	180 g (typical)
Conformity	CE (*), RoHS, GenICam, GigE Vision, IP30, FCC * The CE Conformity Declaration is available on the Basler website: www.baslerweb.com		
Software	Basler pylon 4 Camera Software Suite (version 4.0 or higher) Available for Windows in 32 and 64 bit versions and Linux (x86 32 bit, x86 64 bit, ARM softfloat, ARM hardfloat).		

Table 3: General Specifications

Specification	scA1400-17 gm/gc	scA1400-30 gm/gc	scA1600-14 gm/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 ICX285 AL/AQ	Sony ICX285 AL/AQ	Sony ICX274 AL/AQ
	Progressive scan CCD		
Optical Size	2/3"	2/3"	1/1.8"
Pixel Size	6.45 µm x 6.45 µm	6.45 µm x 6.45 µm	4.4 µm x 4.4 µm
Max. Frame Rate (at full resolution)	17 fps	30 fps	14 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 RGB 8 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		
Camera Power Requirements	+12 VDC (-5 %) to +24 VDC (+5 %), < 1% ripple		
	3.5 W @ 12 V	4.2 W @ 12 V	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	scA1400-17gm/gc	scA1400-30gm/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)	170 g (typical)	170 g (typical)	160 g (typical)
(90° head housing)	190 g (typical)	190 g (typical)	180 g (typical)
Conformity	CE (*), RoHS, GenICam, GigE Vision, IP30, FCC * The CE Conformity Declaration is available on the Basler website: www.baslerweb.com		
Software	Basler pylon 4 Camera Software Suite (version 4.0 or higher) Available for Windows in 32 and 64 bit versions and Linux (x86 32 bit, x86 64 bit, ARM softfloat, ARM hardfloat).		

Table 4: General Specifications

Specification	scA1600-28 gm/gc
Sensor Size (H x V pixels)	gm: 1626 x 1236 gc: 1624 x 1234
Sensor Type	Sony ICX274 AL/AQ
	Progressive scan CCD
Optical Size	1/1.8"
Pixel Size	4.4 µm x 4.4 µm
Max. Frame Rate (at full resolution)	28 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 RGB 8 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
Camera Power Requirements	+12 VDC (-5 %) to +24 VDC (+5 %), < 1% ripple
	4.3 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	scA1600-28gm/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)	170 g (typical)
(90° head housing)	190 g (typical)
Conformity	CE (*), RoHS, GenICam, GigE Vision, IP30, FCC * The CE Conformity Declaration is available on the Basler website: www.baslerweb.com
Software	Basler pylon 4 Camera Software Suite (version 4.0 or higher) Available for Windows in 32 and 64 bit versions and Linux (x86 32 bit, x86 64 bit, ARM softfloat, ARM hardfloat).

Table 5: General Specifications

1.3 Spectral Response for Mono Cameras

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

ⓘ	The spectral response curves excludes lens characteristics and light source characteristics.
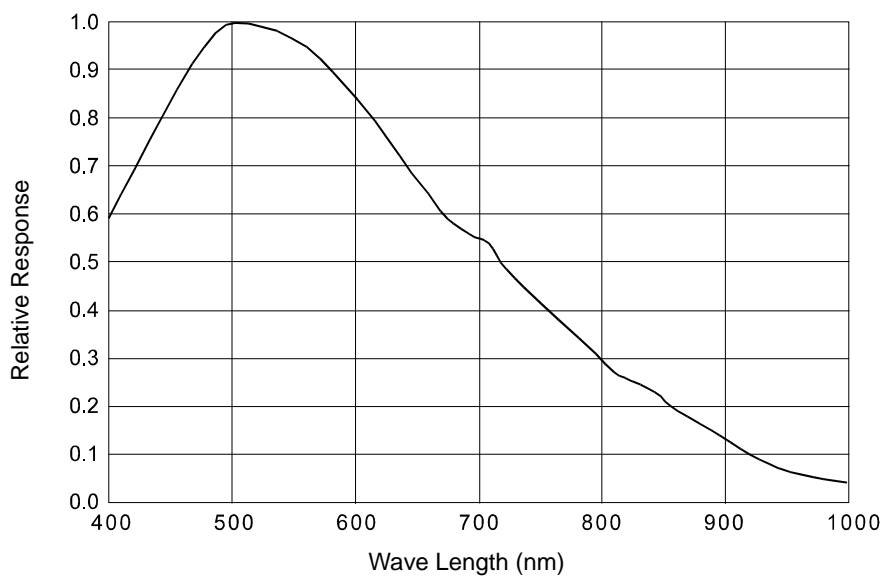
Fig. 1: scA640-70gm Spectral Response Spectral Response
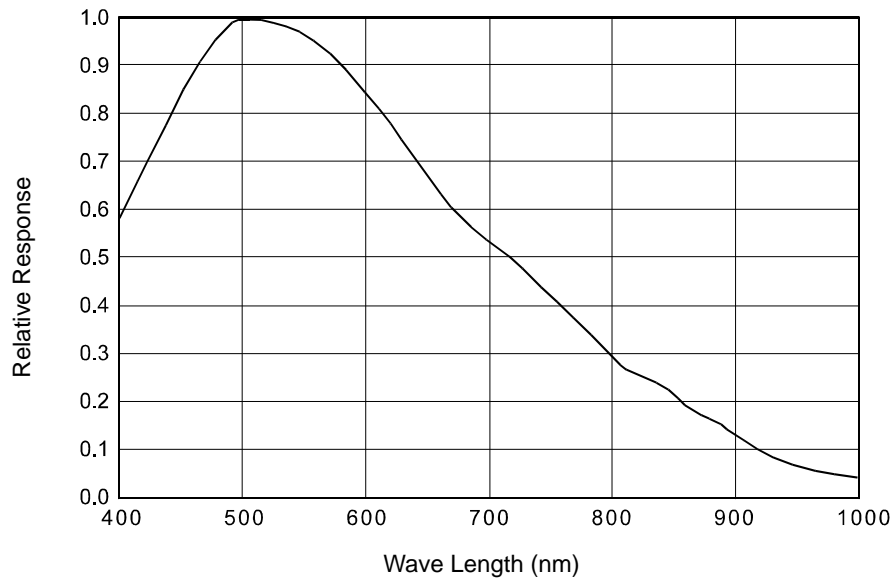
Fig. 2: scA640-74 gm Spectral Response
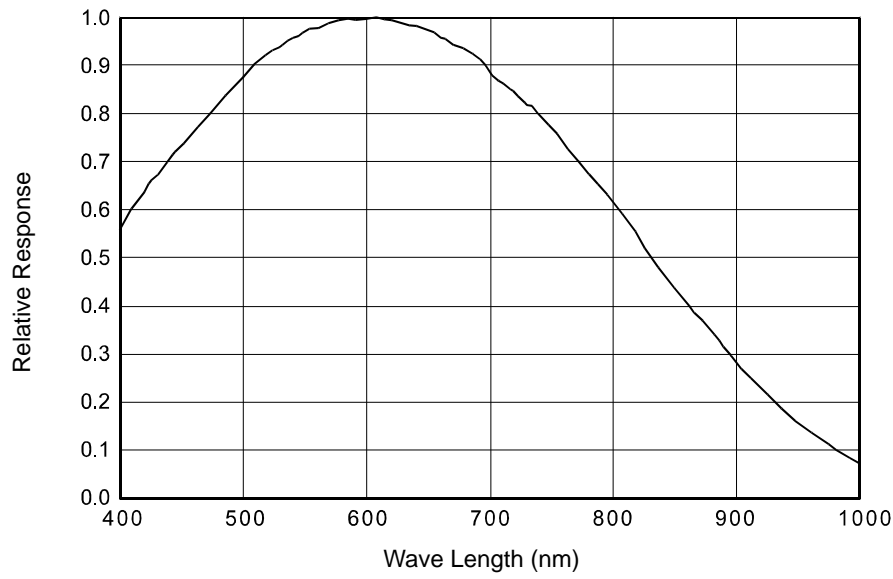
Fig. 3: scA640-120 gm Spectral Response
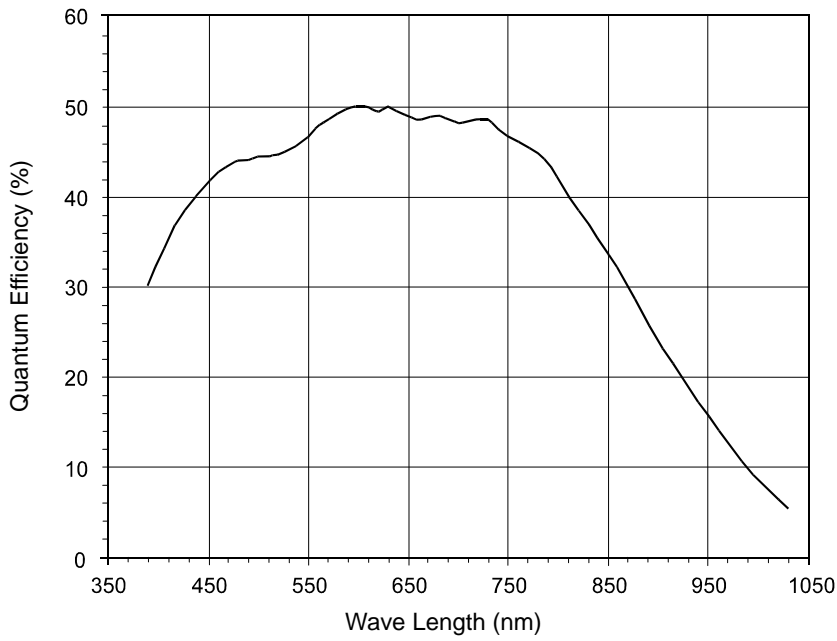
Fig. 4: scA750-60 gm Spectral Response
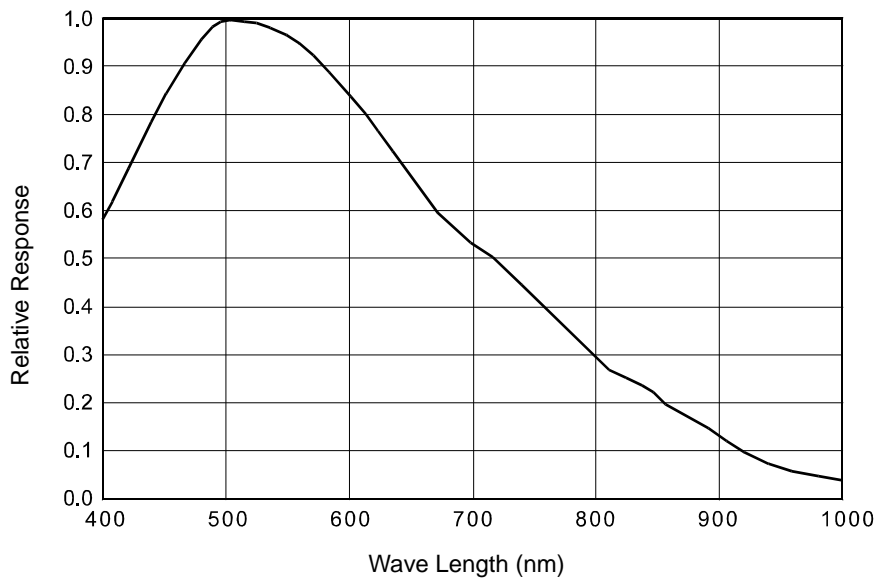
Fig. 5: scA780-54 gm Spectral Response
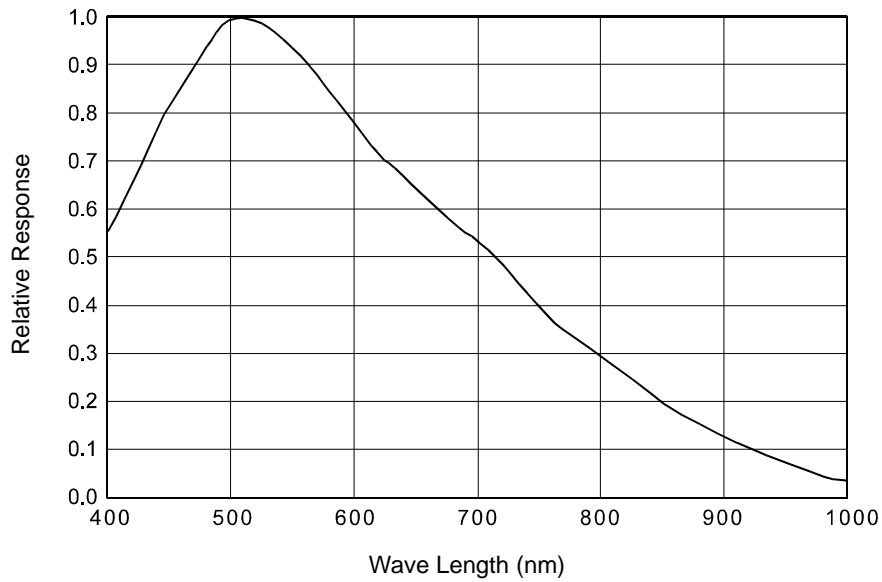
Fig. 6: scA1000-30gm Spectral Response
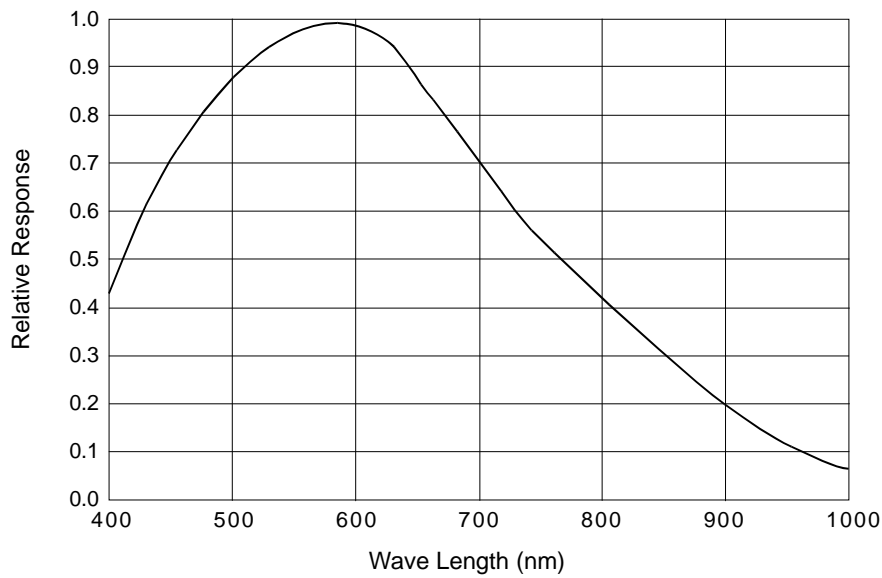
Fig. 7: scA1300-32 gm Spectral Response
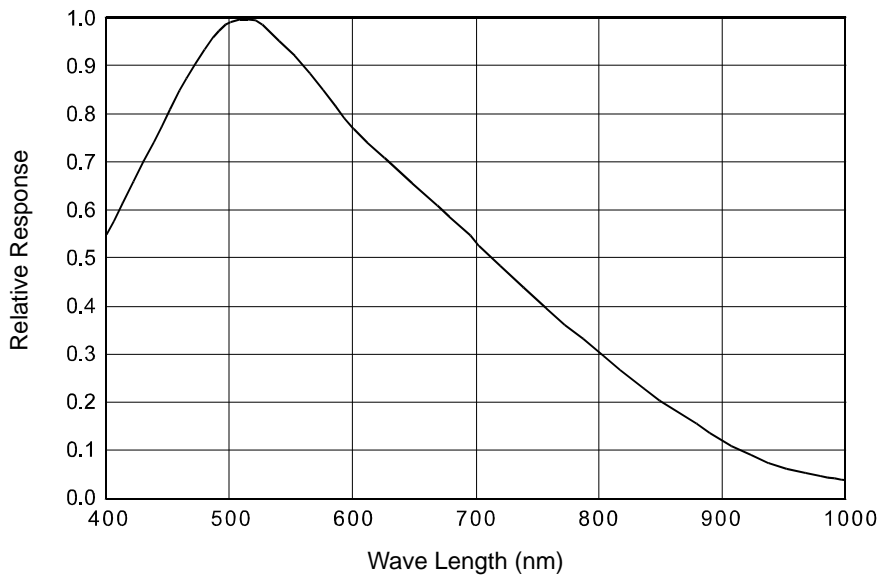
Fig. 8: scA1390-17gm Spectral Response
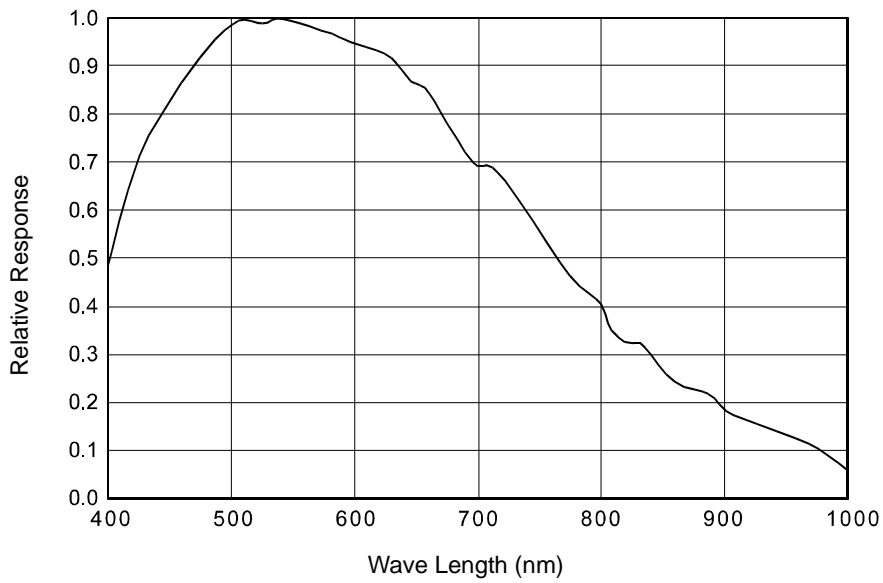
Fig. 9: scA1400-17gm and scA1400-30gm Spectral Response
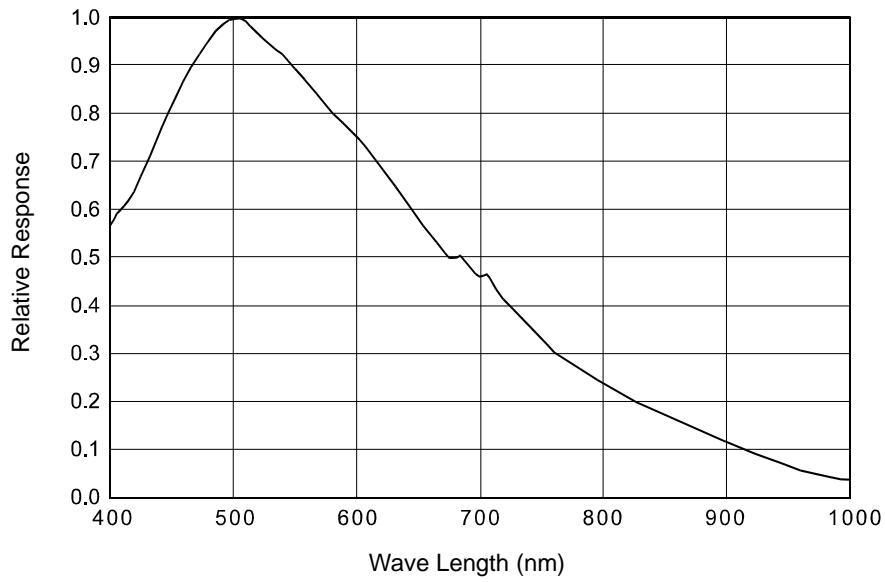
Fig. 10: scA1600-14 gm and scA1600-28 gm Spectral Response

1.4   Spectral Response for Color Cameras

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

---

ⓘ   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 640 ... 660 nm, and it should cut off from 640 ... 660 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).
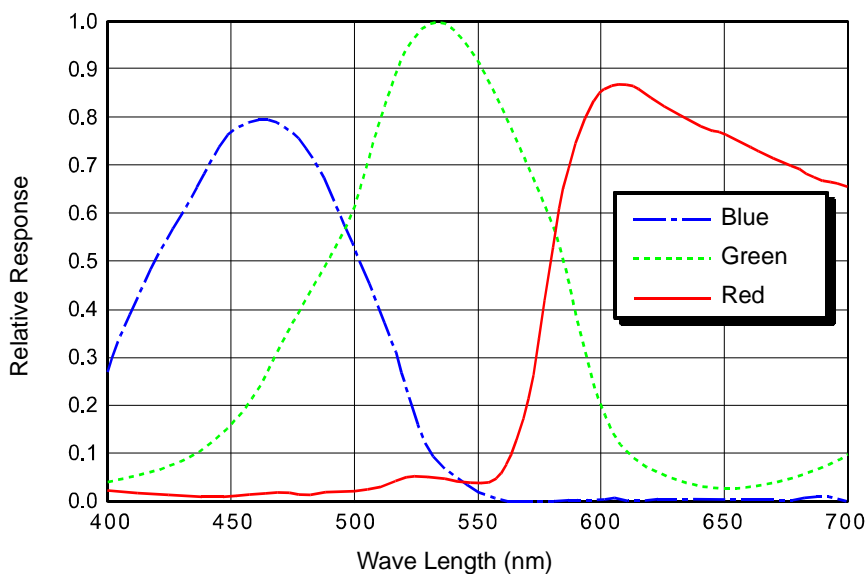
Fig. 11: scA640-70 gc Spectral Response
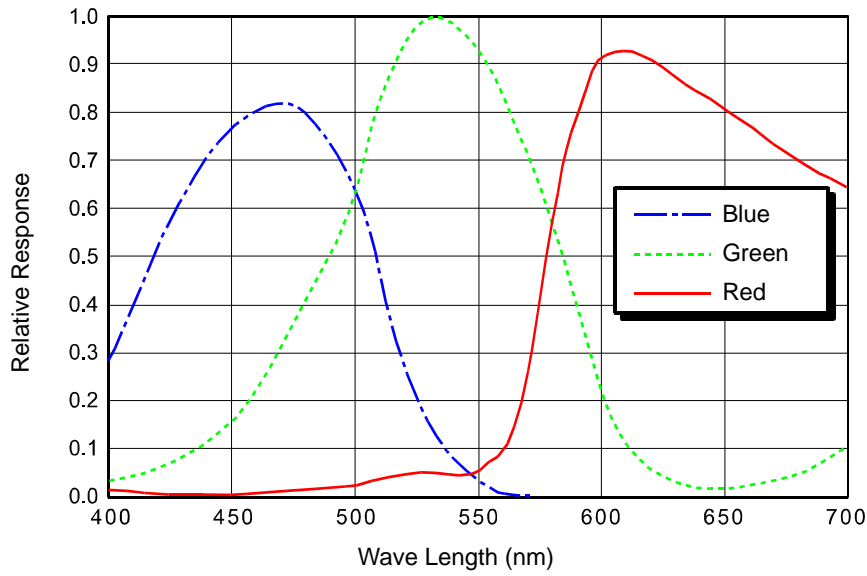
Fig. 12: scA640-74gc Spectral Response
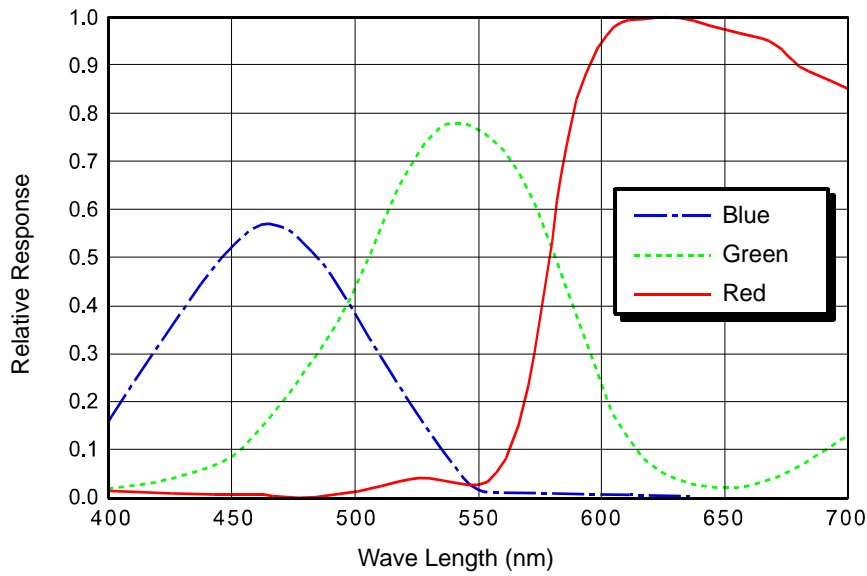
Fig. 13: scA640-120gc Spectral Response
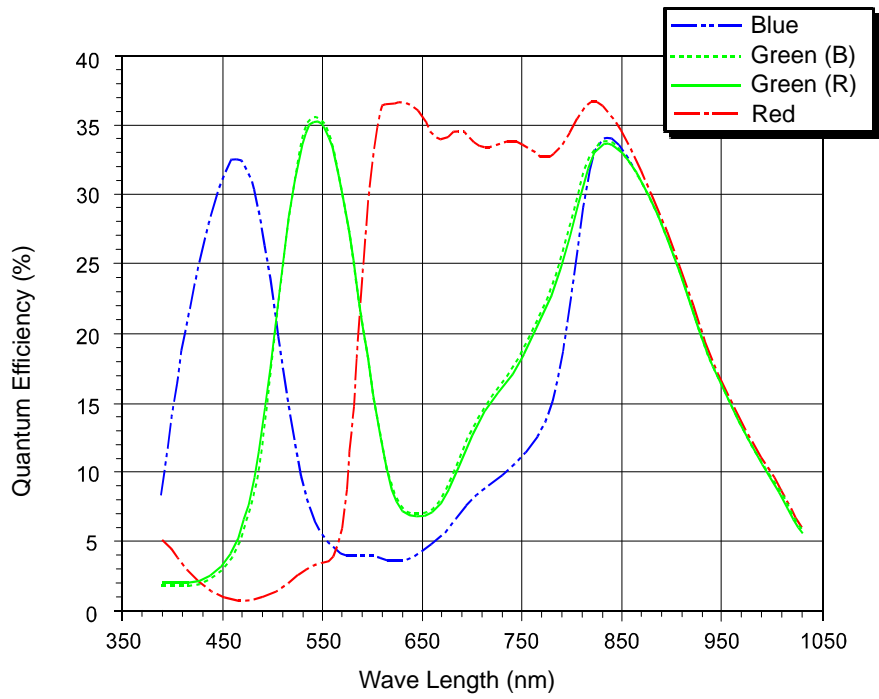
Fig. 14: scA750-60gc Spectral Response
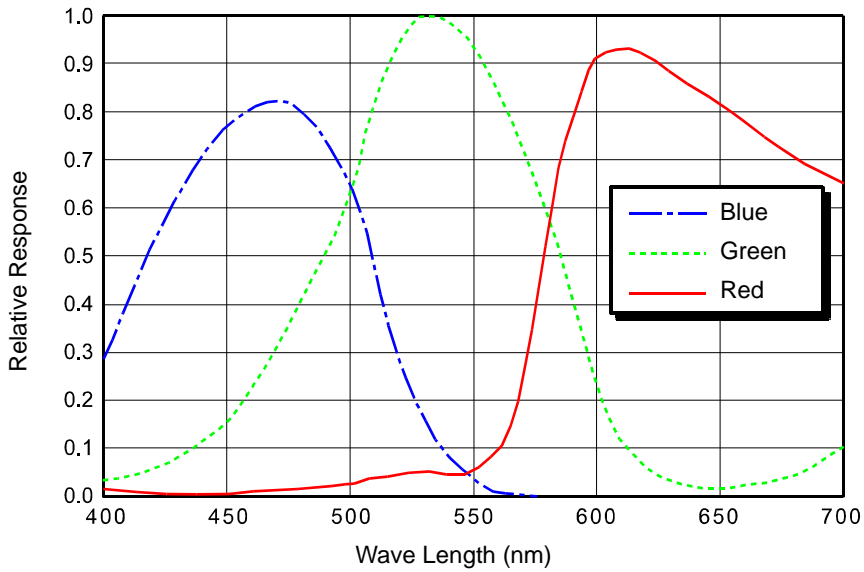
Fig. 15: scA780-54gc Spectral Response
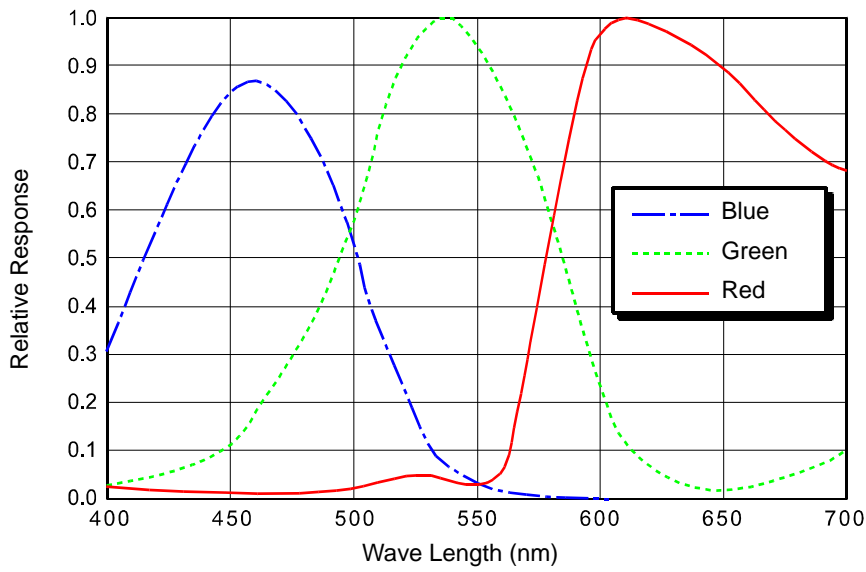
Fig. 16: scA1000-30gc Spectral Response
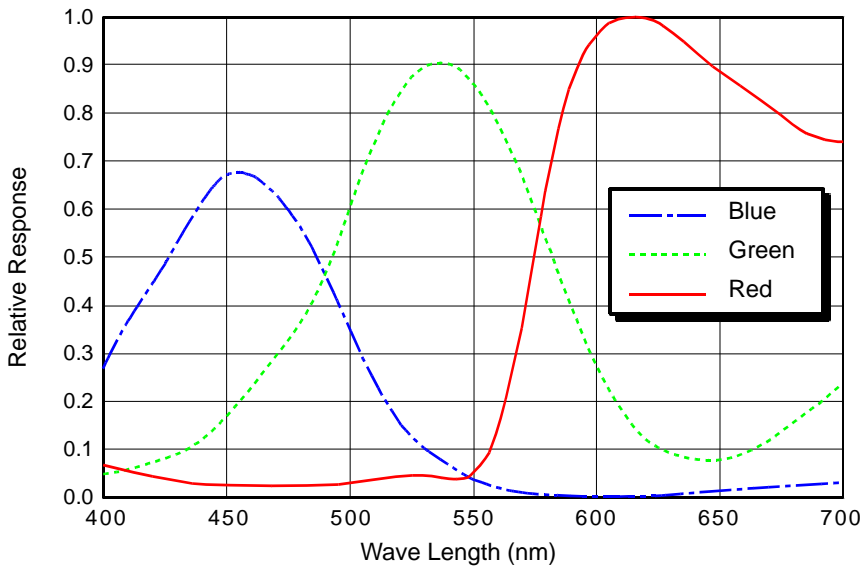
Fig. 17: scA1300-32gc Spectral Response
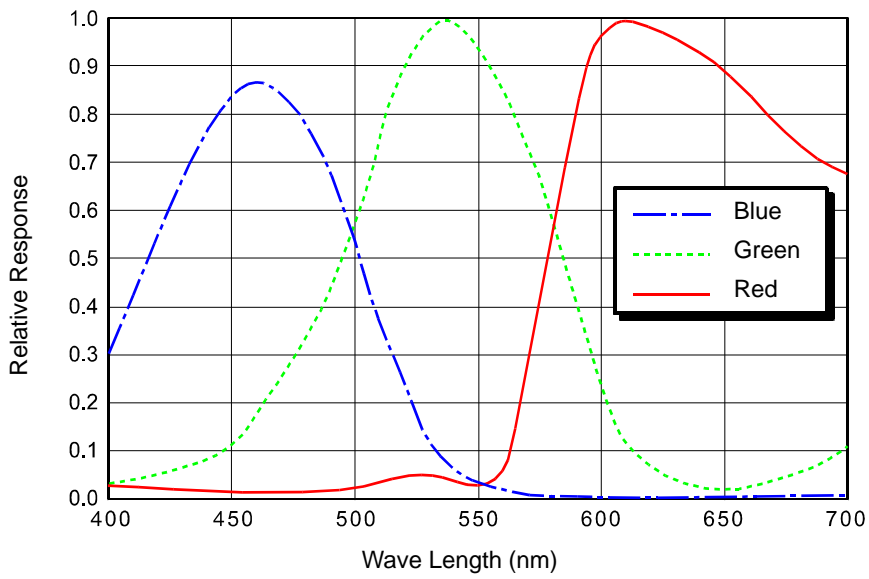
Fig. 18: scA1390-17gc Spectral Response
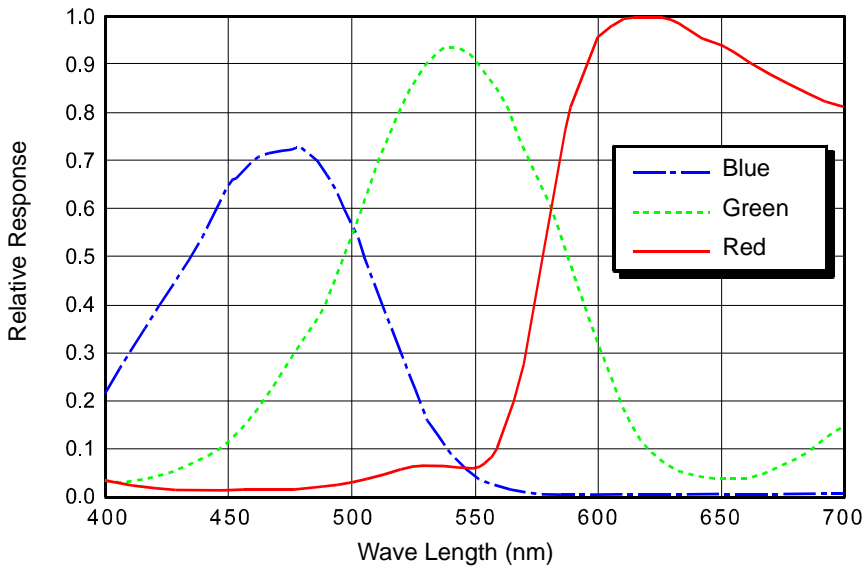
Fig. 19: scA1400-17gc and scA1400-30gc Spectral Response
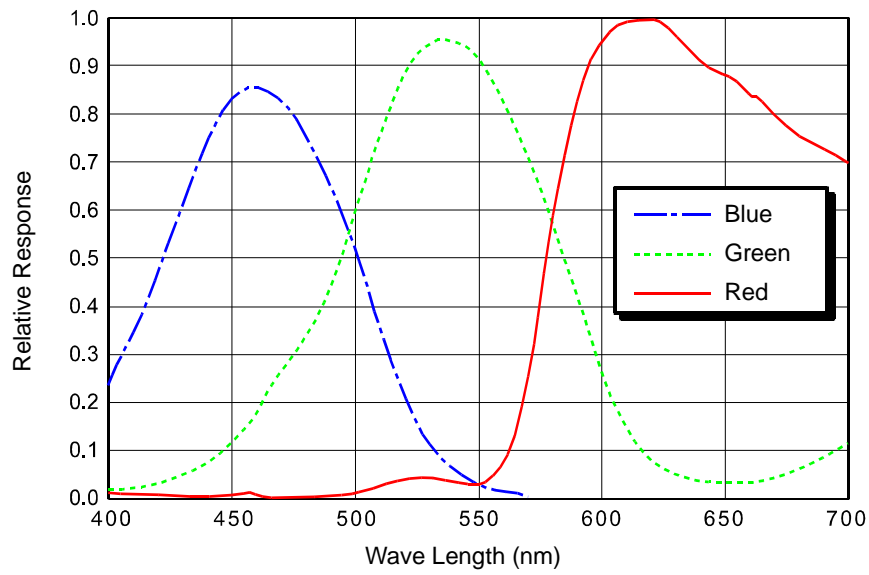
Fig. 20: scA1600-14gc and scA1600-28gc Spectral Response

1.5   Mechanical Specifications

1.5.1   Standard Housing

The camera housing conforms to protection class IP30 assuming that the lens mount is covered by a lens or by the cap that is shipped with the camera.

1.5.1.1   Camera Dimensions and Mounting Points

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

The dimensions in millimeters for cameras equipped with a standard C-mount lens adapter are as shown in Figure 21. The dimensions for cameras equipped with an optional CS-mount lens adapter are as shown in Figure 22.

Camera housings are equipped with four mounting holes on the top and four mounting holes on the bottom as shown in the drawings.
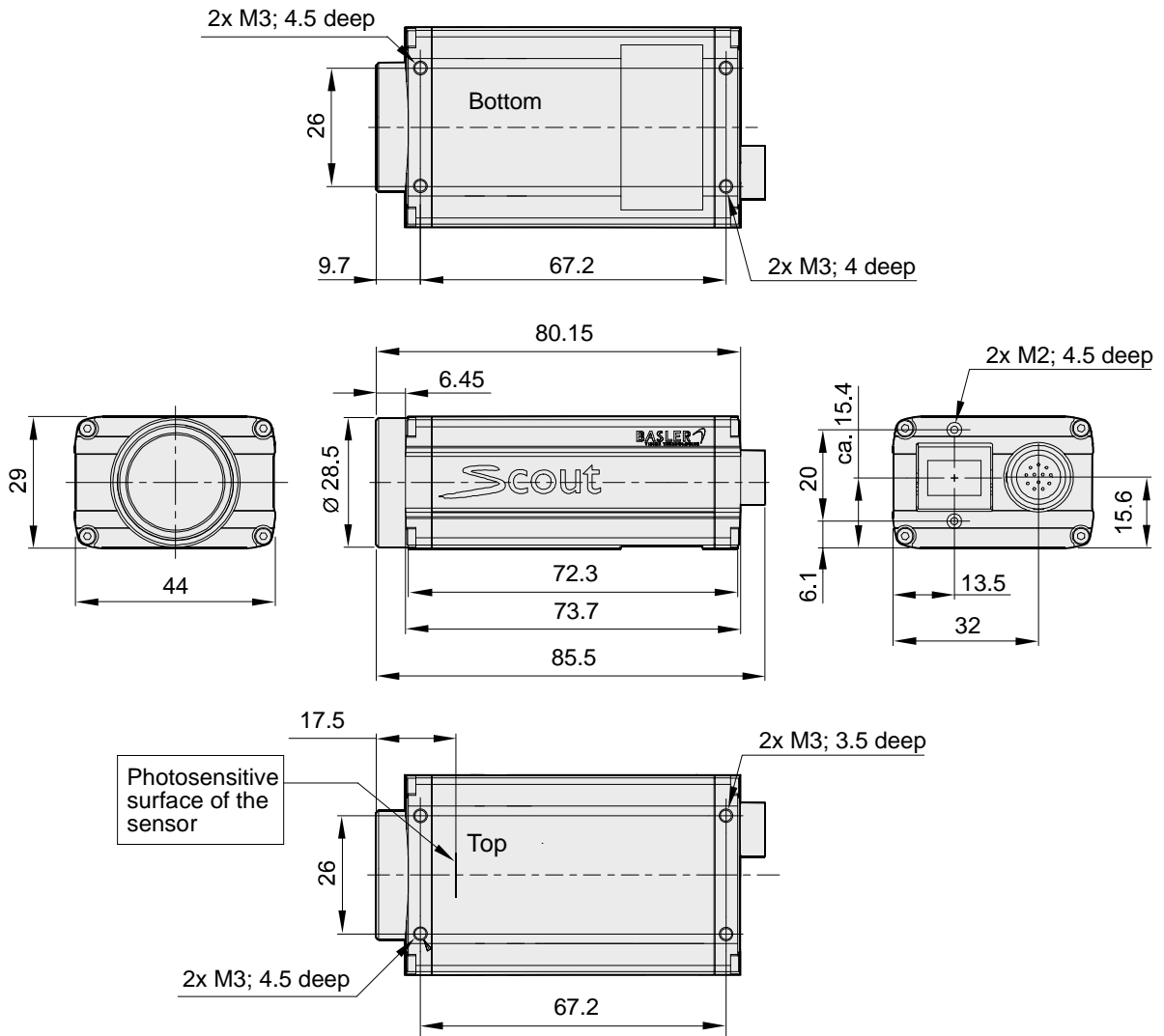
Fig. 21: Mechanical Dimensions (in mm) for Cameras with the Standard C-mount Lens Adapter
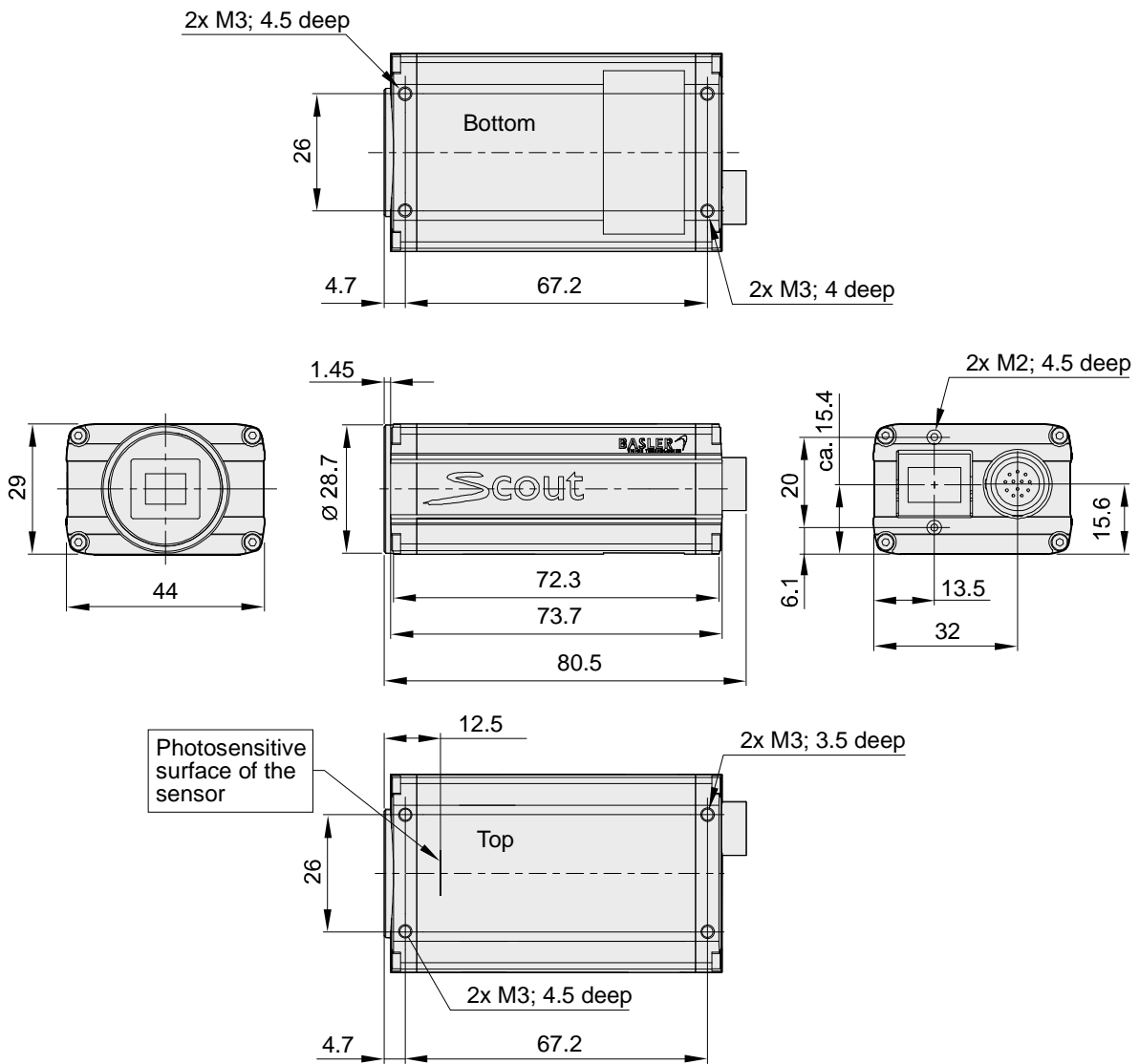
Fig. 22: Mechanical Dimensions (in mm) for Cameras with an Optional CS-mount Lens Adapter

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 23. The sensor positioning accuracy for cameras equipped with an optional CS-mount lens adapter is as shown in Figure 24 on page 28.

$\triangleright$ = reference plane

* = tolerance to the center of the lens mount (optical axis)

** = tolerance to the reference planes

± 0.02 (This is the sensor tilt tolerance. It applies to every point on the photosensitive surface and is relative to the center of the die)

Photosensitive surface of the sensor

17.5 $^{+0}_{-0.06}$ (This tolerance is for the distance between the front of the lens mount and the sensor's photosensitive surface. Note that this tolerance and the sensor tilt tolerance (see above) must be combined to obtain the total tolerance for every point on the photosensitive surface.)

Maximum Sensor Tilt Angle (Degrees)					
Camera Model	Tilt X	Tilt Y	Camera Model	Tilt X	Tilt Y
scA640-70gm/gc	0.47	0.63	scA1300-32gm/gc	0.47	0.63
scA640-74gm/gc	0.35	0.47	scA1390-17gm/gc	0.31	0.42
scA640-120gm/gc	0.62	0.83	scA1400-17gm/gc	0.25	0.34
scA750-60gm/gc	0.51	0.80	scA1400-30gm/gc	0.25	0.34
scA780-54gm/gc	0.35	0.47	scA1600-14gm/gc	0.34	0.52
scA1000-30gm/gc	0.46	0.63	scA1600-28gm/gc	0.34	0.52

Fig. 23: Sensor Positioning Accuracy for Cameras with the Standard C-mount Lens Adapter

(in mm unless otherwise noted)

$\pm 0.75°$

± 0.25 *

± 0.25 *

± 0.4 **

± 0.4 **

▷ = reference plane

* = tolerance to the center of the lens mount (optical axis)

** = tolerance to the reference planes

± 0.02 (This is the sensor tilt tolerance. It applies to every point on the photosensitive surface and is relative to the center of the die)

Photosensitive surface of the sensor

$12.5 \; {}^{+0}_{-0.06}$ (This tolerance is for the distance between the front of the lens mount and the sensor's photosensitive surface. Note that this tolerance and the sensor tilt tolerance (see above) must be combined to obtain the total tolerance for every point on the photosensitive surface.)

Maximum Sensor Tilt Angle (Degrees)					
Camera Model	Tilt X	Tilt Y	Camera Model	Tilt X	Tilt Y
scA640-70 gm/gc	0.47	0.63	scA1300-32 gm/gc	0.47	0.63
scA640-74 gm/gc	0.35	0.47	scA1390-17 gm/gc	0.31	0.42
scA640-120 gm/gc	0.62	0.83	scA1400-17 gm/gc	0.25	0.34
scA750-60 gm/gc	0.51	0.80	scA1400-30 gm/gc	0.25	0.34
scA780-54 gm/gc	0.35	0.47	scA1600-14 gm/gc	0.34	0.52
scA1000-30 gm/gc	0.46	0.63	scA1600-28 gm/gc	0.34	0.52

Fig. 24: Sensor Positioning Accuracy for Cameras with an Optional CS-mount Lens Adapter
(in mm unless otherwise noted)

1.5.2   90° Head Housing

The camera housing conforms to protection class IP30 assuming that the lens mount is covered by a lens or by the cap that is shipped with the camera.

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 ensure precise mounting with high repeatability.

The dimensions in millimeters for cameras equipped with a standard C-mount lens adapter are as shown in Figure 25.

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).

(i)	For optimum accuracy in the positioning of the camera's optical axis, we recommend using the front module reference plane (see the figure in the Sensor Positioning Accuracy section) as mounting surface.
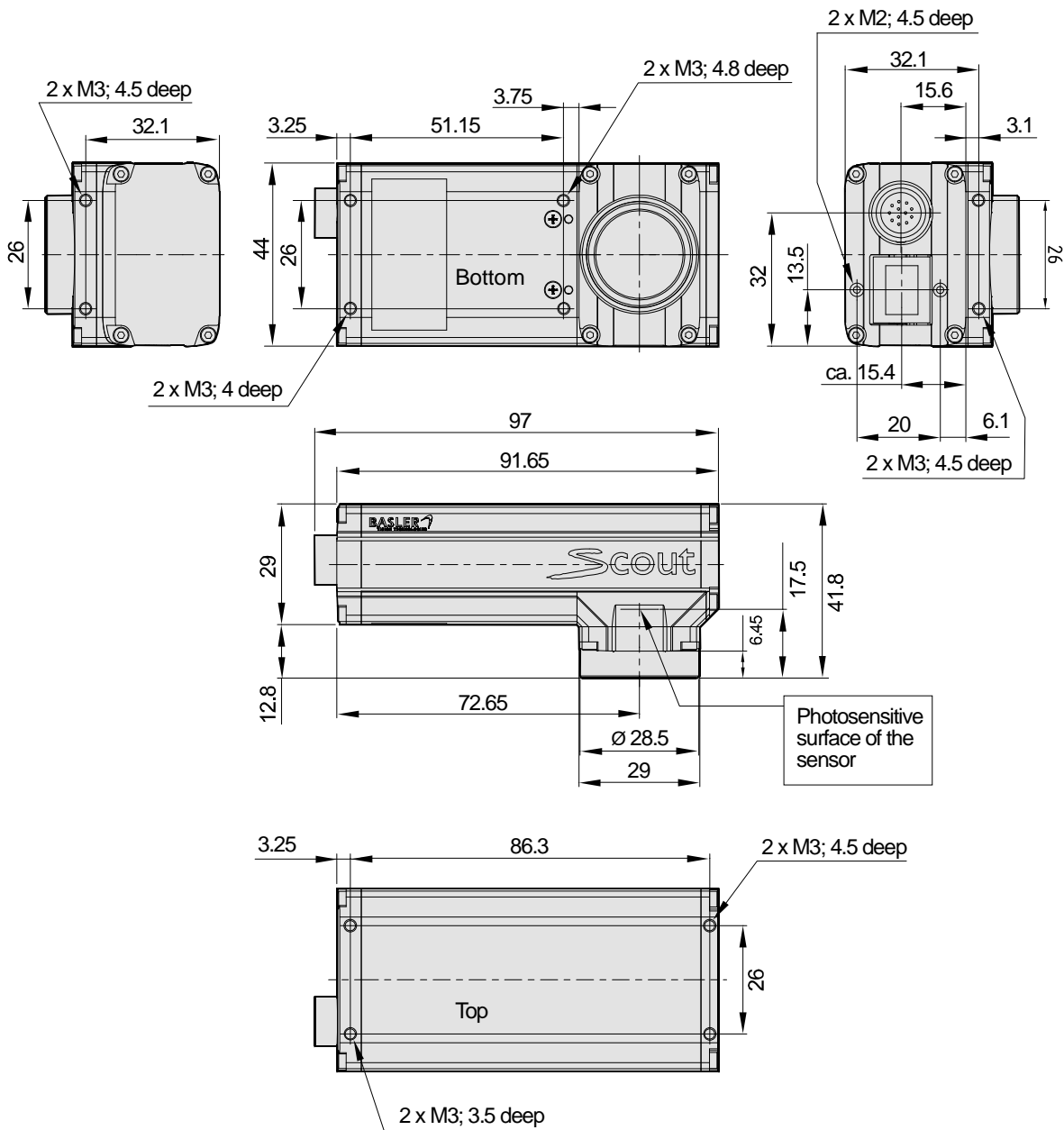
Fig. 25: Mechanical Dimensions (in mm) for Cameras (90° Head) with the Standard C-mount Lens Adapter

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 26.

▷ = reference plane

▷ = front module reference plane

* = tolerance to the center of the lens mount (optical axis)

** = tolerance to the reference planes

± 0.02 (This is the sensor tilt tolerance. It applies to every point on the photosensitive surface and is relative to the center of the die)

Photosensitive surface of the sensor

(2 : 1)

$17.5 \, {}^{+0}_{-0.06}$

(This tolerance is for the distance between the front of the lens mount and the sensor's photosensitive surface. Note that this tolerance and the sensor tilt tolerance (see above) must be combined to obtain the total tolerance for every point on the photosensitive surface.)

Maximum Sensor Tilt Angle (Degrees)					
Camera Model	Tilt X	Tilt Y	Camera Model	Tilt X	Tilt Y
scA640-70 gm/gc	0.47	0.63	scA1390-17 gm/gc	0.31	0.42
scA640-74 gm/gc	0.35	0.47	scA1400-17 gm/gc	0.25	0.34
scA640-120 gm/gc	0.62	0.83	scA1400-30 gm/gc	0.25	0.34
scA750-60 gm/gc	0.51	0.80	scA1600-14 gm/gc	0.34	0.52
scA780-54 gm/gc	0.35	0.47	scA1600-28 gm/gc	0.34	0.52
scA1000-30 gm/gc	0.46	0.63			

Fig. 26: Sensor Positioning Accuracy for Cameras (90° Head) with the Standard C-mount Lens Adapter (in mm unless otherwise noted)

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.
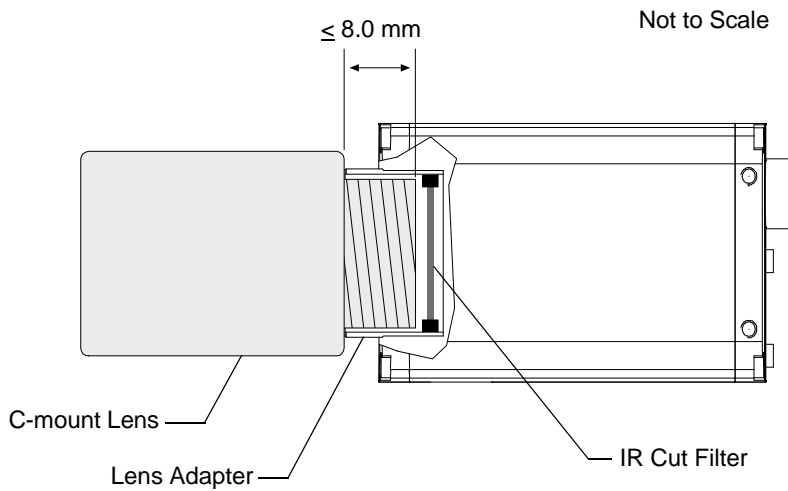
Fig. 27: Maximum Lens Thread Length on Color Cameras

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 stress tests were performed on selected camera models with standard housings. 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) / 00:30 h

Table 6: Mechanical Stress Tests

The mechanical stress tests were performed with a dummy lens connected to a C-mount. The dummy lens was 35 mm long and had a mass of 66 g. Using a heavier or longer lens requires an additional support for the lens.

1.6   Software Licensing Information

1.6.1   LWIP TCP/IP Licensing

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

Copyright (c) 2001, 2002 Swedish Institute of Computer Science. All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

3. The name of the author may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE AUTHOR "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED.

IN NO EVENT SHALL THE AUTHOR BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.6.2   LZ4 Licensing

The software in the camera includes the LZ4 implementation. The copyright information for this implementation is as follows:

LZ4 - Fast LZ compression algorithm

Copyright (C) 2011-2013, Yann Collet.

BSD 2-Clause License (http://www.opensource.org/licenses/bsd-license.php)

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

1.7   Avoiding EMI and ESD Problems

The cameras are frequently installed in industrial environments. These environments often include devices that generate electromagnetic interference (EMI) and they are prone to electrostatic discharge (ESD). Excessive EMI and ESD can cause problems with your camera such as false triggering or can cause the camera to suddenly stop capturing images. EMI and ESD can also have a negative impact on the quality of the image data transmitted by the camera.

To avoid problems with EMI and ESD, you should follow these general guidelines:

- Always use high quality shielded cables. The use of high quality cables is one of the best defenses against EMI and ESD.

- Try to use camera cables that are the correct length and try to run the camera cables and power cables parallel to each other. Avoid coiling camera cables. If the cables are too long, use a meandering path rather then coiling the cables.

- Avoid placing camera cables parallel to wires carrying high-current, switching voltages such as wires supplying stepper motors or electrical devices that employ switching technology. **Placing camera cables near to these types of devices may cause problems with the camera.**

- Attempt to connect all grounds to a single point, e.g., use a single power outlet for the entire system and connect all grounds to the single outlet. This will help to avoid large ground loops. (Large ground loops can be a primary cause of EMI problems.)

- Use a line filter on the main power supply.

- Install the camera and camera cables as far as possible from devices generating sparks. If necessary, use additional shielding.

- Decrease the risk of electrostatic discharge by taking the following measures:

    - Use conductive materials at the point of installation (e.g., floor, workplace).

    - Use suitable clothing (cotton) and shoes.

    - Control the humidity in your environment. Low humidity can cause ESD problems.

(i)	The Basler application note called *Avoiding EMI and ESD in Basler Camera Installations* provides much more detail about avoiding EMI and ESD. This application note can be obtained from the Downloads section of our website: www.baslerweb.com

1.8   Environmental Requirements

1.8.1   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

1.8.2   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.

1.9   Precautions

	Avoid Dust on the Sensor
⚠ **CAUTION**	■ The camera is shipped with a cap on the lens mount. To avoid collecting dust on the camera's IR cut filter (color cameras) or sensor (mono cameras), make sure that you always put the cap in place when there is no lens mounted on the camera. ■ Every time you remove or replace the plastic cap, a lens, or a lens adapter, make sure that the camera is pointing down. ■ Never apply compressed air to the camera. This can easily contaminate optical components, particularly the sensor. ■ To further enhance dust protection, the internal space in the camera that contains the imaging sensor is sealed off from the camera's other internal spaces.

|	**Lens Thread Length is Limited**
⚠ **CAUTION**	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 .

	Voltage Outside of Specified Range Can Cause Damage
⚠ **CAUTION**	The recommended operating voltage of the power to the camera is +12 VDC (-5 %) to +24 VDC (+5 %). If the voltage is less than +11.3 VDC, the camera can operate erratically.

	An Incorrect Plug Can Damage the 12-pin Connector
⚠ **CAUTION**	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.

<table>
<tr>
<td>

⚠️

CAUTION

</td>
<td>

Inappropriate Code Can Cause Unexpected Camera Behavior

The code snippets provided in this manual are included as sample code only. Inappropriate code can cause your camera to function differently than expected and can compromise your application. To ensure that the snippets will work properly in your application, you must adjust them to meet your specific needs and must test them thoroughly prior to use.

The code snippets in this manual are written in C++. Other programming languages can also be used to write code for use with Basler pylon. When writing code, you should use a programming language that is both compatible with pylon and appropriate for your application.

For more information about the programming languages that can be used with Basler pylon, see the documentation included with the pylon Camera Software Suite.

</td>
</tr>
</table>

Warranty Precautions

To ensure that your warranty remains in force:

Do not remove the camera's serial number label

If the label is removed and the serial number can't be read from the camera's registers, the warranty is void.

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.

Avoid Electromagnetic fields

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

Transport properly

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

Clean properly

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   Installation

The information you will need to install and operate the camera is included in the *Installation and Setup Guide for Cameras Used with pylon for Windows* (AW000611).

You can download the document from the Basler website: www.baslerweb.com

The guide includes the information you will need to install both hardware and software and to begin capturing images. It also describes the recommended network adapters, describes the recommended architecture for the network to which your camera is attached, and deals with the IP configuration of your camera and network adapter.

After completing your camera installation, refer to the "Basler Network Drivers and Parameters" and "Network Related Camera Parameters and Managing Bandwidth" sections of this camera User's Manual for information about improving your camera's performance in a network and about using multiple cameras.

ⓘ	After the camera is powered on, pylon software processes a camera description file included in the camera to make the camera features available for use.
	The following camera description files are available and are used alternatively:
	▪ The "Full" camera description file providing all features
	▪ The "Basic" camera description file providing most features.
	Processing the "Full" camera description file takes more time than processing the "Basic" camera description file. Accordingly, processing the "Basic" camera description file requires a shorter period to elapse until the camera features are available for use.
	In order to make the camera features sooner available after the initial start of the camera, the camera is configured by the factory to process the "Basic" camera description file. Therefore, some features will initially **not** be available.
	If you want to use any of the features that are initially not available you must activate the camera description file providing all features. If you will only use the smaller feature set, we recommend not to change the factory setting to make the camera features sooner available.
	For more information, particularly about activating a camera description file and about the features not provided by the smaller camera description file, see Section 12.19 on page 321.

3   Tools for Changing Camera Parameters

3.1   Basler pylon Camera Software Suite

The Basler pylon Camera Software Suite is designed for use with all Basler cameras with the following interface types: IEEE 1394a interface, IEEE 1394b, GigE, or USB 3.0. It can also be used with newer Camera Link cameras. The pylon Camera Software Suite offers reliable, real-time image data transport into the memory of your PC at a very low CPU load.

You can download the Basler Camera Software Suite from the Basler website: www.baslerweb.com

The pylon Camera Software Suite includes several tools that you can use to change the parameters on your camera, including the pylon Viewer and the pylon API for different programming languages. The remaining sections in this chapter provide an introduction to these tools.

For more information about installing pylon software, see the *Installation and Setup Guide for Cameras Used with pylon for Windows* (AW000611). You can download the guide from the Basler website: www.baslerweb.com

3.1.1   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.

For more information about using the viewer, see the *Installation and Setup Guide for Cameras Used with pylon for Windows* (AW000611). You can download the guide from the Basler website: www.baslerweb.com

3.1.2   pylon IP Configurator

The Basler pylon IP Configurator 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.

For more information about using the IP Configurator, see the *Installation and Setup Guide for Cameras Used with pylon for Windows* (AW000611). You can download the guide from the Basler website: www.baslerweb.com

3.1.3   pylon SDKs

Three pylon SDKs are part of the Basler pylon Camera Software Suite:

- pylon SDK for C++ (Windows and Linux)
- pylon SDK for C (Windows)
- pylon SDK for .NET / C# (Windows)

Each SDK includes an API, a set of sample programs, and documentation:

- You can access all of the camera's parameters and control the camera's full functionality from within your application software by using the matching pylon API (C++, C, or .NET).
- The sample programs illustrate how to use the pylon API to parameterize and operate the camera.
- For each environment (C++, C, and .NET), a *Programmer's Guide and Reference Documentation* is available. The documentation gives an introduction to the pylon API and provides information about all methods and objects of the API.

4 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.

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

For more information about compatible Intel chipsets and about installing the network drivers, see the *Installation and Setup Guide for Cameras Used with pylon for Windows* (AW000611).

4.1   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. Ensure the Packet Timeout parameter is set to a longer time interval than the time interval set for the inter-packet delay.

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 Basler pylon API. The following code snippet illustrates using the API to read and write the parameter values:

```
// Enable Resend
Camera_t::StreamGrabber_t StreamGrabber ( Camera.GetStreamGrabber(0) );
StreamGrabber.EnableResend.SetValue(false); // disable resends

// Packet Timeout/FrameRetention
Camera_t::StreamGrabber_t StreamGrabber ( Camera.GetStreamGrabber(0) );
StreamGrabber.PacketTimeout.SetValue( 40 );
StreamGrabber.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 the *Installation and Setup Guide for Cameras Used with pylon for Windows* (AW000611).

4.2   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 the *Installation and Setup Guide for Cameras Used with pylon for Windows* (AW000611).

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.

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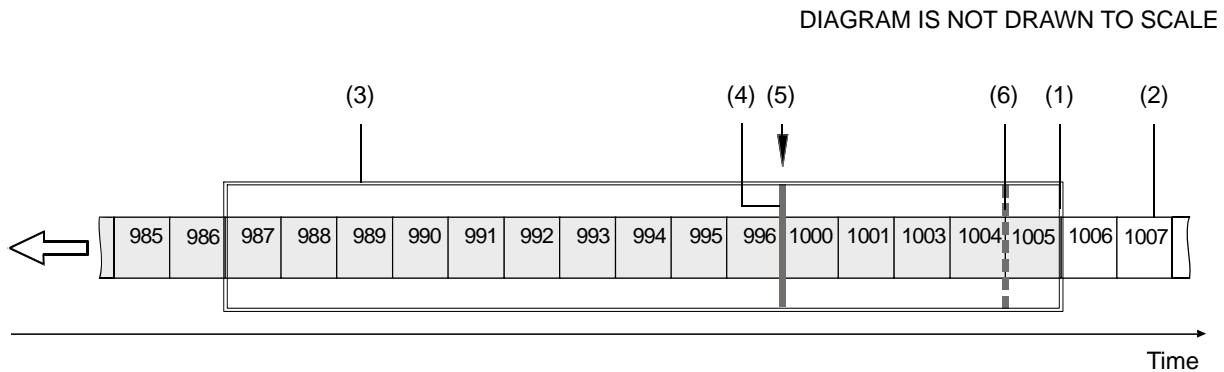
Fig. 28: Example of a Receive Window with Resend Request Threshold & Resend Request Batching Threshold

(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.

DIAGRAM IS NOT DRAWN TO SCALE
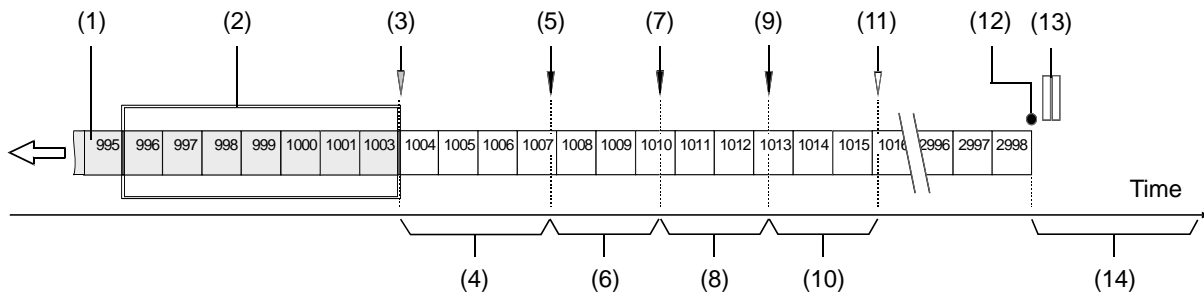
Fig. 29: Incomplete Stream of Packets and Part of the Resend Mechanism

(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. Make sure the Packet Timeout parameter is set to a longer time interval than the time interval set for the inter-packet delay.

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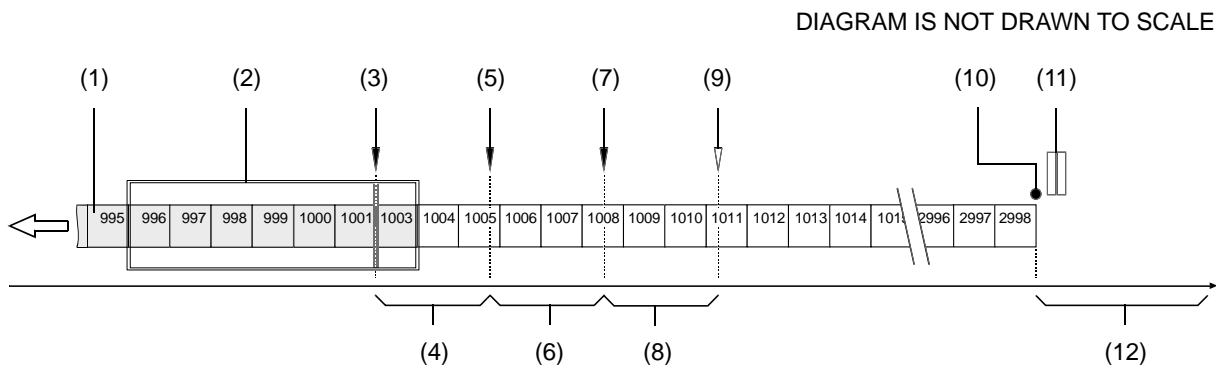
Fig. 30: Combination of Threshold Resend Mechanism and Timeout Resend Mechanism

(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 Basler pylon API. The following code snippet illustrates using the API to read and write the parameter values:

```
// Get the Stream Parameters object
Camera_t::StreamGrabber_t StreamGrabber( Camera.GetStreamGrabber(0) );

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

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

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

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

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

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

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

// Write the MaximumNumberResendRequests parameter
StreamGrabber.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 the *Installation and Setup Guide for Cameras Used with pylon for Windows* (AW000611).

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:

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

	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.

4.3   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.

> (i) 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 Basler 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( Camera.GetTLNodeMap() );
TlParams.ReadTimeout.SetValue(500);  // 500 milliseconds
TlParams.WriteTimeout.SetValue(500); // 500 milliseconds

// Heartbeat Timeout
Camera_t::TlParams_t TlParams( Camera.GetTLNodeMap() );
TlParams.HeartbeatTimeout.SetValue(5000);  // 5 seconds
```

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.

5   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.

5.1   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, which use a total of 36 bytes, and the last data packet may be a smaller size. The payload size will be packet size minus 36 bytes.

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.

When setting the time interval for the inter-packet delay, make sure that the time interval for the packet timeout is set to a higher value.

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.

Time

Time Period	1	2	3	4	5	6	7	8	9
	F A & T	F A & T	F A & T	F A & T	F A & T	F A & T		F A & T	F A & T
Resends available via the bandwidth reserve	5	5	5	5	5	5	5	5	5
Resends needed	0	7	4	10	20	1	0	0	1
Effect on the accumulator pool	0	-2	+1	-5	-9	+4	+5	+5	+1
Resends left in the accumulator pool after frame transmission	15	13	14	9	0	4	9	14	15

F A & T = Frame Acquired and Transmitted

Not enough resends available. Packet unavailable errors generated.

(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 retries 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 Basler 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.

5.2   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.
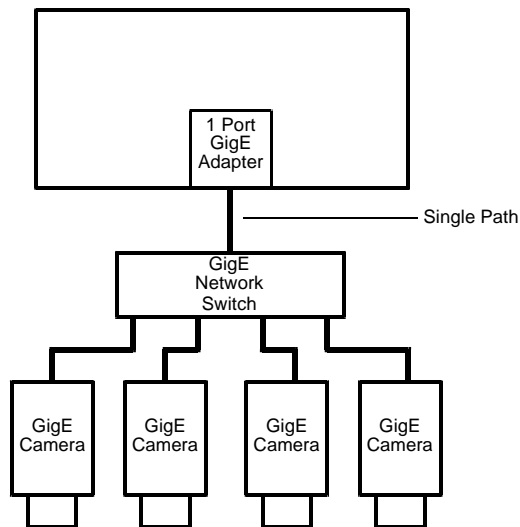
Fig. 31: Multiple Cameras on a Network

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.

5.2.1   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 - Improve the Network Performance.

If you use, as recommended, the Basler performance driver with an Intel PRO network adapter or a compatible network adapter, the network parameters for the network adapter are automatically optimized and need not be changed.

If you use the Basler filter driver and have already set network parameters for your network adapter during the installation of the Basler pylon software, continue with step two. Otherwise, open the **Network Connection Properties** window for your network adapter and check the following network parameters:

- If you use an Intel PRO network adapter: Make sure the **Receive Descriptors** parameter is set to its maximum value and the **Interrupt Moderation Rate** parameter is set to **Extreme**.

  Also make sure the **Speed and Duplex Mode** parameter is set to **Auto Detect**.

- If you use a different network adapter, see whether parameters are available that will allow setting the number of receive descriptors and the number of CPU interrupts. The related parameter names may differ from the ones used for the Intel PRO adapters. Also, the way of setting the parameters may be different. You may, e.g., have to use a parameter to set a low number for the interrupt moderation and then use a different parameter to enable the interrupt moderation.

  If possible, set the number of receive descriptors to a maximum value and set the number of

CPU interrupts to a low value.

If possible, also set the parameter for speed and duplex to auto.

Contact Basler technical support if you need further assistance.

Step 2 - 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.

Unless you have already set the packet size for your network adapter during the installation of the Basler pylon software, check 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 maximum size of 16 kB. 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 8 kB packets and the switch can handle 6 kB packets, then the maximum for the network is 6 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.

ⓘ	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 3 - 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 4 - 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 5 - 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 6 - 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.

	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 frame rate.

Step 7 - 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 GigE 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 GigE 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 9.13 on page 163.

For more information about the AOI, see Section 12.4 on page 234.

6   Camera Functional Description

This chapter 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 later chapters of the user's manual.

6.1   Overview (All Models Except scA750-60)

> (i)   The information in this section applies to all camera models except the scA750-60 gm/gc. For information about scA750-60 gm/gc cameras, see Section 6.2 on page 73.

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

Exposure start and exposure time 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 "frame start trigger" (ExFSTrig) signal applied to the camera's input line. The ExFSTrig signal facilitates periodic or non-periodic acquisition start. Modes are available that allow the length of exposure time to be directly controlled by the ExFSTrig 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 72). 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.

CCD Sensor

Fig. 32: CCD Sensor Architecture
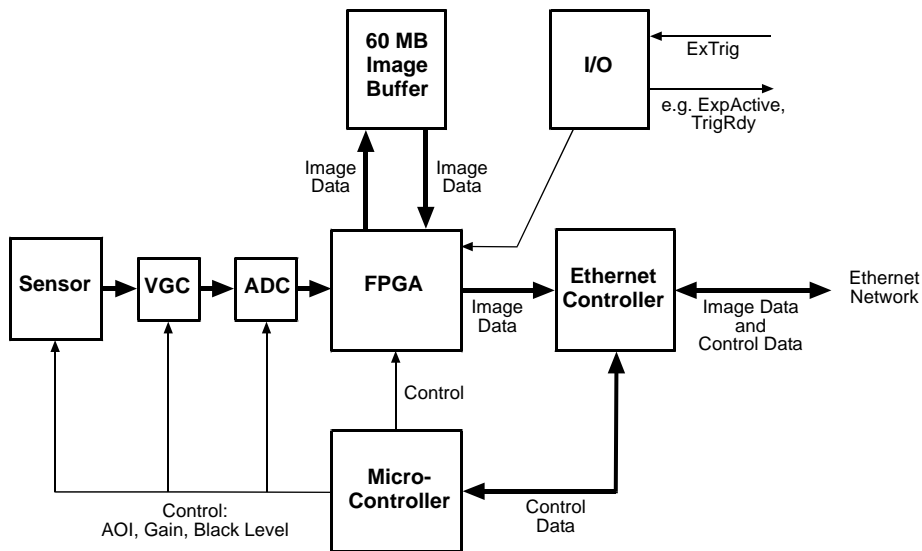
Fig. 33: Camera Block Diagram

6.2   Overview (scA750-60 Only)

(i)	The information in this section only applies to scA750-60 gm/gc cameras. For information about the other camera models, see Section 6.1 on page 71.

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 72). 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 likewise 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
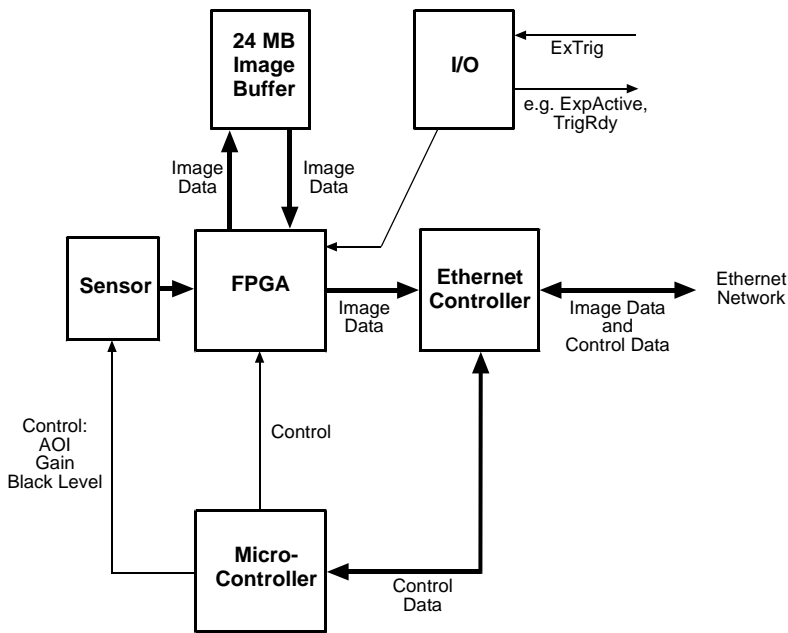
Fig. 35: Camera Block Diagram

7   Physical Interface

This chapter 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.

7.1   General Description of the Connectors

The camera is interfaced to external circuitry 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 power to the camera.

The drawing below shows the location of the two connectors and the LEDs.
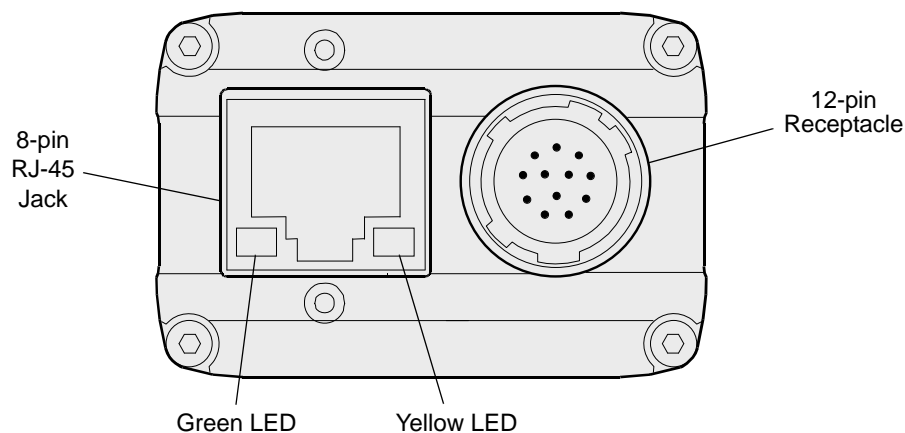
Fig. 36: Camera Connectors and LED

7.2   Camera Connector Types, Connection Numbering, and Assignments

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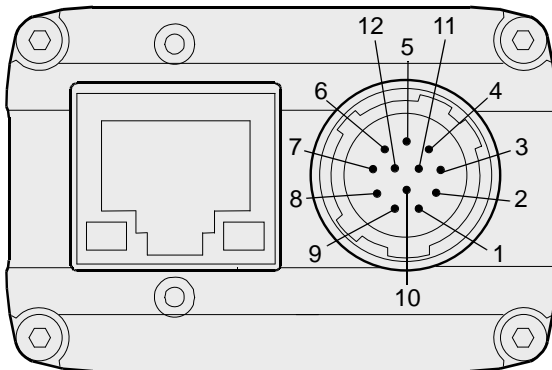
Fig. 37: Pin Numbering for the 12-pin Receptacle

7.2.2   8-pin RJ-45 Jack

The 8-pin RJ-45 jack provides Ethernet access to the camera. Pin assignments adhere to the Ethernet standard.

The recommended mating connector is any standard 8-pin RJ-45 plug. Cables terminated with screw-lock connectors are available from Basler. Contact your Basler sales representative to order cable assemblies.

To ensure that you order cables with the correct connectors, note the vertical orientation of the screws before ordering.

Green and Yellow LEDs

The 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.

7.2.3   12-pin Connector

> ⚠ **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.

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

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 7. Pin numbering is as shown in Section 7.2.1 on page 76.

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

Table 7: Pin Assignments for the 12-pin Receptacle

> ⓘ
>
> * 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 when there are long wires between your power suppy and the camera, we recommend that you provide camera power VCC through separate wires between your power supply and pins 8 and 9 on the camera. We also recommend that you provide camera power ground through separate wires between your power supply and pins 1 and 2 on the camera.

7.3   Cabling Requirements

7.3.1   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 strongly 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 a GigE network adapter in a PC or to a GigE network switch.

Close proximity to strong magnetic fields should be avoided.

7.3.2   Power and I/O Cable

ⓘ	You must use a power and I/O cable suitable for the camera and the connected I/O device. ▪ If the connected I/O device **can apply** signal voltages that meet the input voltage requirements of the camera, and if enhanced protection of the signal transmission against EMI or ESD is not required, use a **standard power and I/O cable**. ▪ If the connected I/O device **cannot apply** signal voltages that meet the input voltage requirements of the camera (e.g. if the I/O device's signal voltage for the "low" logic state is above 1.4 V), or if additional protection of the signal transmission against EMI or ESD is required, use a **PLC power and I/O cable**. For more information about input voltage requirements, see Section 7.6.1 on page 81. For more information about EMI and ESD, see Section 1.7 on page 36.

General Recommendations

The power and I/O cable that connects to the camera

▪ must be terminated with a suitable I/O connector. For more information, see Section 7.2.3 on page 77.

▪ 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.

▪ must have a cross section of at least 0,14 mm², approximately equal to an American wire gauge (AWG) rating of AWG 26.

The maximum recommended length of the power and I/O cable is 10 meters.

The required I/O connector is available from Basler. Basler also offers a cable assembly that is terminated with an I/O connector on one end and unterminated on the other. Contact your Basler sales representative to order connectors or cables.

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

PLC Power and I/O Cable

The PLC power and I/O cable adjusts the voltage levels of PLC devices to the voltage levels required by the camera, and it protects the camera against negative voltage and reverse polarity.

The cables also offer protection of the signal transmission against EMI and ESD.

Basler offers PLC power and I/O cables in various lengths. Contact your Basler sales representative to order the cables.

7.4   Camera Power

Camera power must be supplied to the camera's 12-pin I/O connector via a power and I/O cable. Power consumption is as shown in the specification tables in Section 1 of this manual.

.

⚠️ **CAUTION**	**Voltage Outside of Specified Range Can Cause Damage** The recommended operating voltage of the power to the camera is +12 VDC (-5 %) to +24 VDC (+5 %). If the voltage is less than +11.3 VDC, the camera may operate erratically.

The following voltage requirements apply to the camera power VCC (pins 8 and 9 of the 12-pin receptacle):

Voltage	Significance
< +11.3 VDC	The camera may operate erratically.
+12 to +24 VDC	Safe operating voltage; < 1 % ripple required. Make sure to use a power supply that supplies power in this voltage range.

Table 8: Voltage Requirements for the Camera Power VCC

For more information about the 12-pin connector and the power and I/O cables see Section 7.2 on page 76, Section 7.3 on page 78, and Section 7.3 on page 78.

7.5   Ethernet GigE Device Information

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

7.6   Input and Output Lines

7.6.1   Input Lines

7.6.1.1   Voltage Requirements

The following voltage requirements apply to the camera's I/O input lines (pins 3 and 4 of the 12-pin receptacle):

Voltage	Significance
+30.0 VDC	Absolute maximum. The absolute maximum must never be exceeded. Otherwise, the camera can be damaged and the warranty becomes void.
+0 to +24 VDC	Safe operating I/O input voltage range.
+0 to +1.4 VDC	The voltage indicates a logical 0.
> +1.4 to +3.2 VDC	Region where the transition threshold occurs; the logical state is not defined in this region.
> +3.2 VDC	The voltage indicates a logical 1.

Table 9: Voltage Requirements for the Camera's I/O Input Lines

(i)	If a PLC power and I/O cable is used, the cable adjusts the input voltage levels (up to 24 V) to the voltage levels required by the camera. For more information, see Section 7.3.2 on .

7.6.1.2   Line Schematic

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. See the previous section for input voltages and their significances. The current draw for each input line is between 5 and 15 mA.

Figure 38 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.
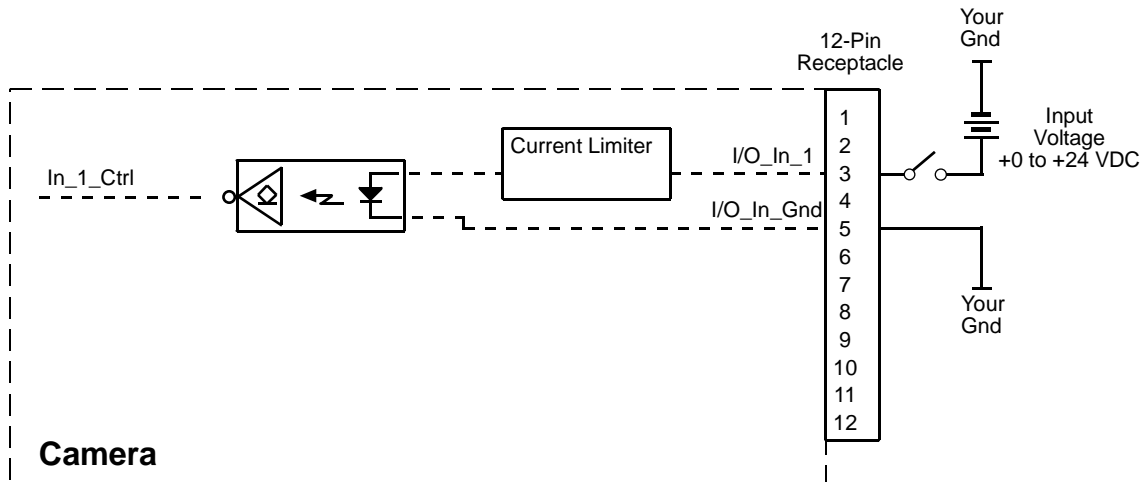
Fig. 38: Typical Voltage Input Circuit

For more information about input line pin assignments and pin numbering, see Section 7.2 on page 76.

For more information about how to use an ExTrig signal to control image acquisition, see Section 9.4.5 on page 112, Section 9.5.3 on page 120 and Section 9.6.3 on page 131.

For more information about configuring the input lines, see Section 8.1 on page 87.

7.6.2   Output Lines

7.6.2.1   Voltage Requirements

The following voltage requirements apply to the I/O output VCC (pin 10 of the 12-pin receptacle):

Voltage	Significance
+30.0 VDC	Absolute maximum.The absolute maximum must never be exceeded. Otherwise, the camera can be damaged and the warranty becomes void.
+3.3 to +24 VDC	Safe operating I/O output supply voltage range.
< +3.3 VDC	The I/O output may operate erratically.

Table 10: Voltage Requirements for the I/O Output VCC

7.6.2.2   Line Schematic

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. See the previous section for the recommended operating voltage. The maximum current allowed through an output circuit is 50 mA.

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

Figure 39 shows a schematic circuit you can use to monitor an output line with a voltage signal. The circuit in Figure 39 is monitoring output line 1.
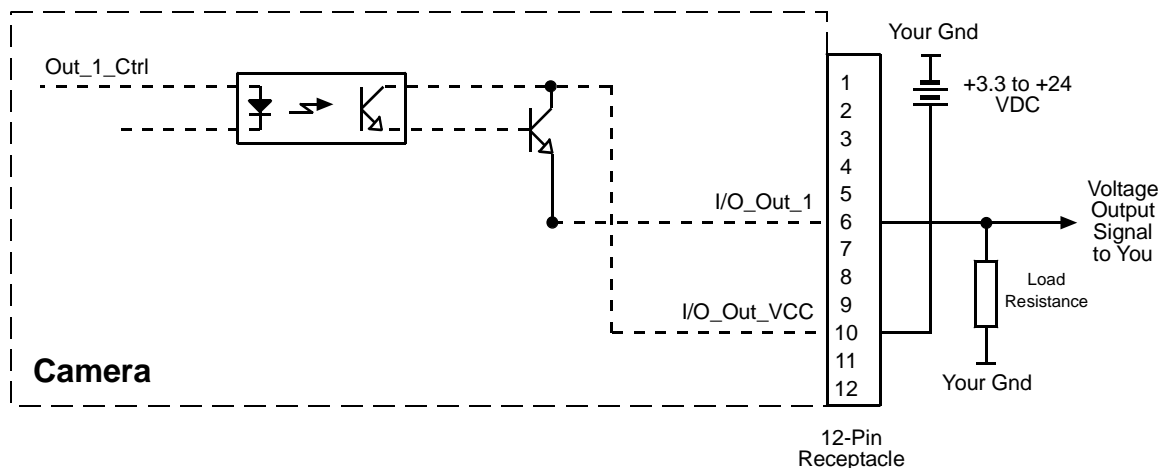
Fig. 39: Typical Voltage Output Circuit

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 7.2 on page 76.

For more information about the exposure active signal, see Section Section 9.11.1 on page 152.

For more information about the trigger ready signal, see Section Section 9.11.3 on page 153 and Section 9.11.4 on page 155.

For more information about assigning camera output signals to physical output lines, see Section 8.2.1 on page 89.

7.6.3   Output Line Response Time

Response times for the output lines on the camera are as defined below.
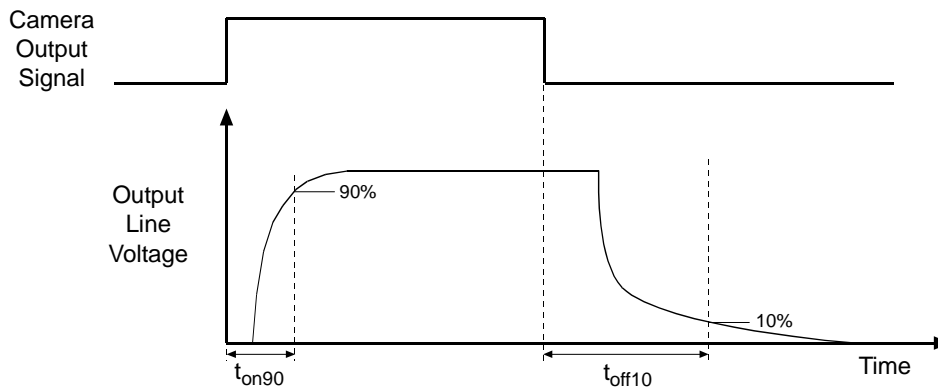
Fig. 40: Output Line Response Times

t_{on90}: Time (µs) from switching on the signal until the voltage has reached 90 % of its final level.

t_{off10}: Time (µs) from switching off the signal until the voltage has dropped to 10 % of its original level.

	The response time values for the output lines on your camera will depend on the load current and the applied voltage of your specific application. In addition, the exact values can vary between individual cameras.

As an example, t_{on90} and t_{off10} were measured for a specific camera, for load currents of 5 mA and 50 mA, and voltages between 3.3 V and 24 V. The results are shown on the figure below.
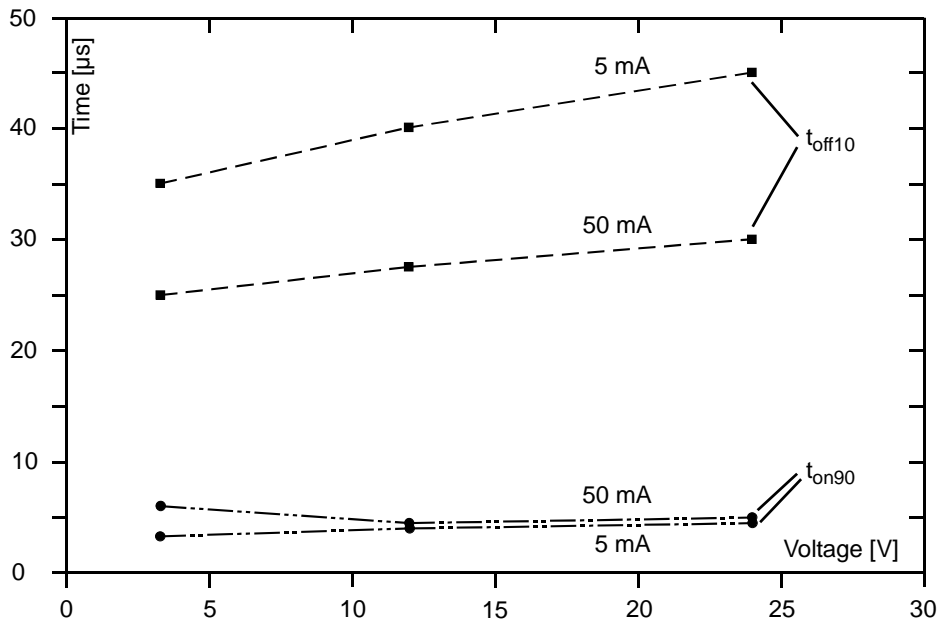
Fig. 41: Dependence of t_{on90} and t_{off10} on Voltage, for Different Load Currents; Measured for an Individual Camera as an Example

Some general tendencies can be seen from the figure:

- t_{off10} increases as the voltage increases and as the load current decreases
- t_{on90} increases as the voltage and the load current increase
- The effects due to different voltages and load currents are more pronounced on t_{off1} than on t_{on90}.

8   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 7.6 on page 81.

8.1   Configuring Input Lines

8.1.1   Assigning an Input Line to Receive a Hardware Trigger Signal

You can assign the camera's input lines to receive external hardware trigger (ExTrig) signals. The incoming ExTrig signals can then be used to control image acquisition.

Section 9.4.5.2 on page 113, Section 9.5.3.4 on page 123 and Section 9.6.3.4 on page 135 explain 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.

ⓘ	The default line assignments depend on the image acquisition control mode: ■ Standard mode: By default, physical input line 1 is assigned to receive an ExTrig signal to serve as the frame start trigger. By default, physical input line 2 is assigned to receive an ExTrig signal to serve as the acquisition start trigger. ■ Legacy mode: By default, physical input line 1 is assigned to receive the ExTrig signal to serve as the acquisition start trigger. You can assign only one line to receive the ExTrig input signal.

8.1.2   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.

(i)	A line assigned to receive an 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 8.3.1 on page 97 and Section 8.3.2 on page 97.

8.1.3   Setting an Input Line for Invert

You can set each individual input line to invert or not to invert the incoming electrical signal. To set the invert function on an input line:

■   Use the Line Selector to select an input line.

■   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:

```
// Enable the inverter on line 1
Camera.LineSelector.SetValue( LineSelector_Line1 );
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.

8.2   Configuring Output Lines

8.2.1   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 as the source signal for a physical output line. The camera has a variety of standard output signals available including:

- Acquisition Trigger Wait (only available when the image acquisition control is set to standard mode)
- Trigger Ready
- Exposure Active
- 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 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 source signal for the selected line

---

> ⓘ   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 3.1.1 on .

For more information about setting the state of user settable output signals, see Section 8.2.2 on .

8.2.2   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 42, 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.
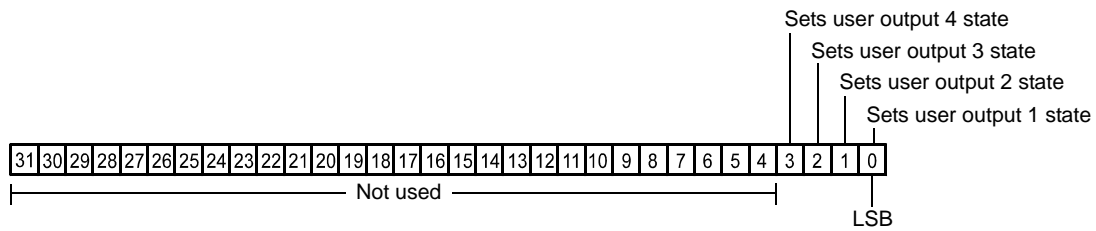
Fig. 42: User Output Value All Parameter Bits

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( );
```

> ⓘ   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.

8.2.3   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 an output line.
■   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:

```
// Enable the inverter on output line 1
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 3.1.1 on page 43.

8.2.4   Working with Timers

The camera has four timer output signals available: Timer 1, Timer 2, Timer 3, and Timer 4. As shown in Figure 43, 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.
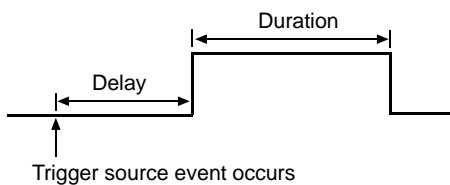
Fig. 43: Timer Signal

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.

Timer with Increased Signal Width

The explanations given above assume that the timer is not modified by the Minimum Output Pulse Width feature.

If the timer us used with the Minimum Output Pulse Width feature and if the set minimum output signal width is larger than the timer duration setting the Minimum Output Pulse Width feature will control the output signal width. Accordingly, the timer duration setting will be ignored.

If the set minimum output signal width is smaller than the timer duration setting the Minimum Output Pulse Width feature will have no effect.

For more information about the Minimum Output Pulse Width feature, see Section 12.12 on .

8.2.4.1   Setting the Trigger Source for a Timer

To set the trigger source for a timer:

■   Use the Timer Selector to select timer 1 or timer 2.

■   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 pylon Viewer, see Section 3.1.1 on .

8.2.4.2   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 0 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 a timer.

■   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 (minimum value), 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 a timer.
- 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.

8.2.4.3   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 a timer.

▪   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 a timer.

▪ 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.

8.3   Checking the State of the I/O Lines

8.3.1   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 an output line.

■   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:

```
// Select output line 2 and read the state
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 3.1.1 on .

8.3.2   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 44, 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 currently high.
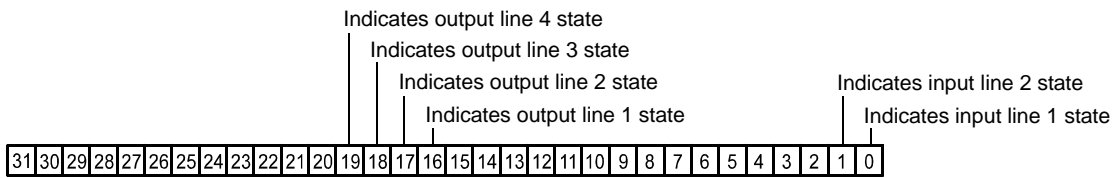
Fig. 44: Line Status All Parameter Bits

9   Image Acquisition Control

This section provides detailed information about controlling image acquisition. You will find details about choosing between image acquisition control modes, triggering image acquisition, 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.

9.1   Image Acquisition Control Modes: Legacy and Standard

Two different image acquisition control modes are available: the legacy mode and the standard mode. Previous Basler scout cameras with firmware version 3.2 and below only operate according to the legacy mode.

The legacy mode differs from the standard mode in only two respects:

- the acquisition start trigger of the standard mode is not available in the legacy mode.

- the frame start trigger of the standard mode is called "acquisition start trigger" in the legacy mode.

Recommendations for choosing the image acquisition control mode:

- If you want to operate the camera together with previous cameras we recommend choosing the legacy mode. In this mode, triggering the camera will be exactly as for previous cameras. Note also that you will not have to modify any code of your application.

- If you do not want to operate the camera together with previous cameras we most strongly recommend choosing the standard mode.

For more information about acquisition start trigger in the legacy mode, see Section 9.6 on page 125.

For more information about acquisition start trigger and frame start trigger in the standard mode, see Section 9.4 on page 107 and Section 9.5 on page 114, respectively.

For more information about determining the camera's firmware version, see Section 12.17 on page 313.

	When the camera is started for the first time after delivery from the factory the image acquisition control will be in legacy mode. If you want the camera to start in standard mode, set the camera to standard mode (see below), save the current parameter settings as a user set and designate this user set as the startup set.

For more information about saving parameter settings as a user set and about working with user sets, see Section 12.18 on .

Setting the Image Acquisition Control Mode

You can set the image acquisition control mode from within your application software by using the pylon API. The following code snippets illustrate using the API to set the image acquisition control mode to standard mode and to legacy mode, respectively:

```
Camera.TriggerControlImplementation =
TriggerControlImplementation_Standard;
```

```
Camera.TriggerControlImplementation =
TriggerControlImplementation_Legacy;
```

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 image acquisition control mode.

For more information about the pylon Viewer, see Section 3.1.1 on .

9.2 Means for Controlling Image Acquisition in Standard Mode

> (i) This section assumes that the standard mode is enabled for image acquisition control.
>
> In principle, this section also applies if the legacy mode is enabled instead.
> In this case, however, the following differences must be taken into account:
>
> - the acquisition start trigger of the standard mode is not available in the legacy mode.
> - the frame start trigger of the standard mode is called "acquisition start trigger" in the legacy mode.

> (i) When the camera is started for the first time after delivery from the factory the image acquisition control will not be in standard mode but in legacy mode.
>
> Use the legacy mode only if you want to operate the camera together with previous cameras not featuring the standard mode.

For more information about standard mode and legacy mode and how to set them, see Section 9.1 on .

This section presents an overview of the elements involved with controlling the acquisition of images. Reading this section will give you an idea about how these elements fit together and will make it easier to understand the detailed information in the sections that follow.

Four major elements are involved in controlling the acquisition of images:

- Acquisition start and acquisition stop commands and the acquisition mode parameter
- The acquisition start trigger
- The frame start trigger
- Exposure time control

When reading the explanations in the overview and in this entire chapter, keep in mind that the term "frame" is typically used to mean a single acquired image.

When reading the material in this section, it is helpful to refer to Figure 45 on and to the use case diagrams in Section 9.8 on . These diagrams present the material related to the acquisition start and stop commands, the acquisition mode, the acquisition start trigger, and the frame start trigger in a graphical format.

Acquisition Start and Stop Commands and the Acquisition Mode

The Acquisition Start command prepares the camera to acquire frames. The camera cannot acquire frames unless an Acquisition Start command has first been executed.

A parameter called the Acquisition Mode has a direct bearing on how the Acquisition Start command operates.

If the Acquisition Mode parameter is set to "single frame", you can only acquire one frame after executing an Acquisition Start command. When one frame has been acquired, the Acquisition Start command will expire. Before attempting to acquire another frame, you must execute a new Acquisition Start command.

If the Acquisition Mode parameter is set to "continuous frame", an Acquisition Start command does not expire after a single frame is captured. Once an Acquisition Start command has been executed, you can acquire as many frames as you like. The Acquisition Start command will remain in effect until you execute an Acquisition Stop command. Once an Acquisition Stop command has been executed, the camera will not be able to acquire frames until a new Acquisition Start command is executed.

Acquisition Start Trigger

The acquisition start trigger is essentially an enabler for the frame start trigger.

The acquisition start trigger has two modes of operation: off and on.

If the Trigger Mode parameter for the acquisition start trigger is set to off, the camera will generate all required acquisition start trigger signals internally, and you do not need to apply acquisition start trigger signals to the camera.

If the Trigger Mode parameter for the acquisition start trigger is set to on, the initial acquisition status of the camera will be "waiting for acquisition start trigger" (see Figure 45 on page 103). When the camera is in this acquisition status, it cannot react to frame start trigger signals. When an acquisition start trigger signal is applied to the camera, the camera will exit the "waiting for acquisition start trigger" acquisition status and enter a "waiting for frame start trigger" acquisition status. The camera can then react to frame start trigger signals. The camera will continue to react to frame start trigger signals until the number of frame start trigger signals it has received is equal to an integer parameter setting called the Acquisition Frame Count. At that point, the camera will return to the "waiting for acquisition start trigger" acquisition status and will remain in that status until a new acquisition start trigger signal is applied.

As an example, assume that the Trigger Mode parameter is set to on, the Acquisition Frame Count parameter is set to three, and the camera is in a "waiting for acquisition start trigger" acquisition status. When an acquisition start trigger signal is applied to the camera, it will exit the "waiting for acquisition start trigger" acquisition status and enter the "waiting for frame start trigger" acquisition status. Once the camera has received three frame start trigger signals, it will return to the "waiting for acquisition start trigger" acquisition status. At that point, you must apply a new acquisition start trigger signal to the camera to make it exit "waiting for acquisition start trigger".

Frame Start Trigger

Assuming that an acquisition start trigger signal has just been applied to the camera, the camera will exit from the "waiting for acquisition start trigger" acquisition status and enter a "waiting for frame start trigger" acquisition status. Applying a frame start trigger signal to the camera at this point will exit the camera from the "waiting for frame start trigger" acquisition status and will begin the

process of exposing and reading out a frame (see Figure 45 on page 103). As soon as the camera is ready to accept another frame start trigger signal, it will return to the "waiting for frame start trigger" acquisition status. A new frame start trigger signal can then be applied to the camera to begin another frame exposure.

The frame start trigger has two modes: off and on.

If the Trigger Mode parameter for the frame start trigger is set to off, the camera will generate all required frame start trigger signals internally, and you do not need to apply frame start trigger signals to the camera. The rate at which the camera will generate the signals and acquire frames will be determined by the way that you set several frame rate related parameters.

If the Trigger Mode parameter for the frame start trigger is set to on, you must trigger frame start by applying frame start trigger signals to the camera. Each time a trigger signal is applied, the camera will begin a frame exposure. When frame start is being triggered in this manner, it is important that you do not attempt to trigger frames at a rate that is greater than the maximum allowed. (There is a detailed explanation about the maximum allowed frame rate at the end of this chapter.) Frame start trigger signals applied to the camera when it is not in a "waiting for frame start trigger" acquisition status will be ignored.
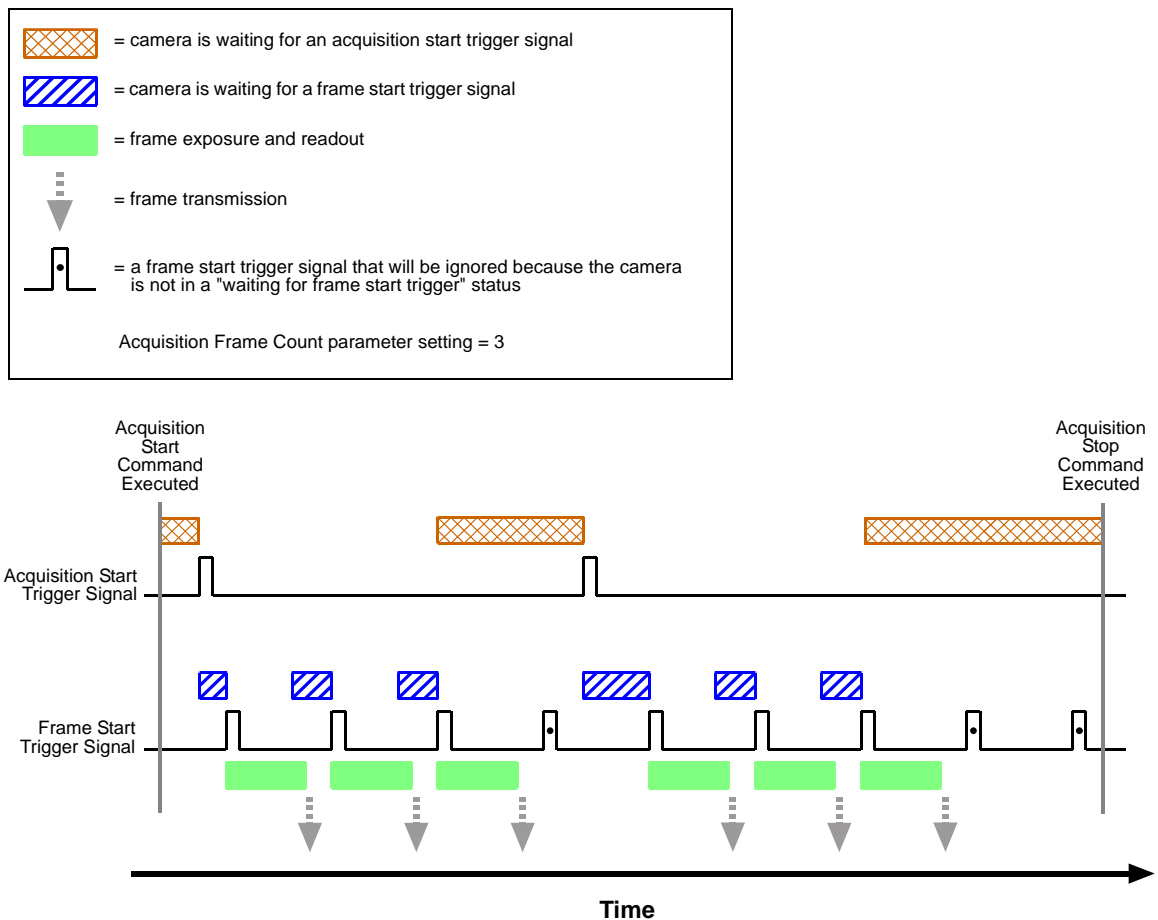
Fig. 45: Acquisition Start and Frame Start Triggering

Applying Trigger Signals

The paragraphs above mention "applying a trigger signal". There are two ways to apply an acquisition start or a frame start trigger signal to the camera: via software or via hardware.

To apply trigger signals via software, you must first select the acquisition start or the frame start trigger and then indicate that software will be used as the source for the selected trigger signal. At that point, each time a Trigger Software command is executed, the selected trigger signal will be applied to the camera.

To apply trigger signals via hardware, you must first select the acquisition start or the frame start trigger and indicate that input line 1 will be used as the source for the selected trigger signal. At that point, each time a proper electrical signal is applied to input line 1, an occurrence of the selected trigger signal will be recognized by the camera.

The Trigger Selector

The concept of the "trigger selector" is very important to understand when working with the acquisition start and frame start triggers. Many of the parameter settings and the commands that apply to the triggers have names that are not specific to a particular type of trigger, for example, the acquisition start trigger has a mode setting and the frame start trigger has a mode setting. But in Basler pylon there is a single parameter, the Trigger Mode parameter, that is used to set the mode for both of these triggers. Also, the Trigger Software command mentioned earlier can be executed for either the acquisition start trigger or the frame start trigger. So if you want to set the Trigger Mode or execute a Trigger Software command for the acquisition start trigger rather than the frame start trigger, how do you do it? The answer is, by using the Trigger Selector parameter. Whenever you want to work with a specific type of trigger, your first step is to set the Trigger Selector parameter to the trigger you want to work with (either the acquisition start trigger or the frame start trigger). At that point, the changes you make to the Trigger Mode, Trigger Source, etc., will be applied to the selected trigger only.

Exposure Time Control

As mentioned earlier, when a frame start trigger signal is applied to the camera, the camera will begin to acquire a frame. A critical aspect of frame acquisition is how long the pixels in the camera's sensor will be exposed to light during the frame acquisition.

If the camera is set for software frame start triggering, the exposure time parameters will determine the exposure time for each frame.

If the camera is set for hardware frame start triggering, there are two modes of operation: "timed" and "trigger width". With the "timed" mode, the exposure time parameters will determine the exposure time for each frame. With the "trigger width" mode, the way that you manipulate the rise and fall of the hardware signal will determine the exposure time. The "trigger width" mode is especially useful if you want to change the exposure time from frame to frame.

9.3   Acquisition Start and Stop Commands and the Acquisition Mode (Legacy and Standard Mode)

Executing an Acquisition Start command prepares the camera to acquire frames. You must execute an Acquisition Start command before you can begin acquiring frames.

Executing an Acquisition Stop command terminates the camera's ability to acquire frames. When the camera receives an Acquisition stop command:

■   If the camera is not in the process of acquiring a frame, its ability to acquire frames will be terminated immediately.

■   If the camera is in the process of acquiring a frame, the frame acquisition process will be allowed to finish and the camera's ability to acquire new frames will be terminated.

The camera's Acquisition Mode parameter has two settings: single frame and continuous. The use of Acquisition Start and Acquisition Stop commands and the camera's Acquisition Mode parameter setting are related.

If the camera's Acquisition Mode parameter is set for single frame, after an Acquisition Start command has been executed, a single frame can be acquired. When acquisition of one frame is complete, the camera will execute an Acquisition Stop command internally and will no longer be able to acquire frames. To acquire another frame, you must execute a new Acquisition Start command.

If the camera's Acquisition Mode parameter is set for continuous frame, after an Acquisition Start command has been executed, frame acquisition can be triggered as desired. Each time a frame trigger is applied while the camera is in a "waiting for frame trigger" acquisition status, the camera will acquire and transmit a frame. The camera will retain the ability to acquire frames until an Acquisition Stop command is executed. Once the Acquisition Stop command is received, the camera will no longer be able to acquire frames.

Setting the Acquisition Mode and Issuing Start/Stop Commands

You can set the Acquisition Mode parameter value and you can execute Acquisition Start or Acquisition Stop commands from within your application software by using the Basler pylon API. The code snippet below illustrates using the API to set the Acquisition Mode parameter value and to execute an Acquisition Start command. Note that the snippet also illustrates setting several parameters regarding frame triggering. These parameters are discussed later in this chapter.

```
Camera.AcquisitionMode.SetValue( AcquisitionMode_SingleFrame );
Camera.TriggerSelector.SetValue( TriggerSelector_FrameStart );
Camera.TriggerMode.SetValue( TriggerMode_On );
Camera.TriggerSource.SetValue ( TriggerSource_Line1 );
Camera.TriggerActivation.SetValue( TriggerActivation_RisingEdge );
Camera.ExposureMode.SetValue( ExposureMode_Timed );
Camera.ExposureTimeAbs.SetValue( 3000 );
```

```
Camera.AcquisitionStart.Execute( );
```

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

For more information about the pylon Viewer, see Section 3.1.1 on page 43.

(i)	When the camera's acquisition mode is set to single frame, the maximum possible acquisition frame rate for a given AOI cannot be achieved. This is true because the camera performs a complete internal setup cycle for each single frame and because it cannot be operated with "overlapped" exposure (the scA750-60 does not allow "overlapped" exposure on principle). To achieve the maximum possible acquisition frame rate set the camera for the continuous acquisition mode. And for all models except scA750-60 also use "overlapped" exposure. For more information about overlapped exposure, see Section 9.9 on page 149.

9.4   The Acquisition Start Trigger in Standard Mode

	This section only applies if the standard mode is enabled for image acquisition control.
	When the camera is started for the first time after delivery from the factory the image acquisition control will not be in standard mode but in legacy mode.
	Use the legacy mode only if you want to operate the camera together with previous cameras not featuring the standard mode.

For more information about standard mode and legacy mode and how to set them, see Section 9.1 on .

The acquisition start trigger is used in conjunction with the frame start trigger to control the acquisition of frames. In essence, the acquisition start trigger is used as an enabler for the frame start trigger. Acquisition start trigger signals can be generated within the camera or may be applied externally as software or hardware acquisition start trigger signals.

When the acquisition start trigger is enabled, the camera's initial acquisition status is "waiting for acquisition start trigger". When the camera is in this acquisition status, it will ignore any frame start trigger signals it receives. If an acquisition start trigger signal is applied to the camera, it will exit the "waiting for acquisition start trigger" acquisition status and enter the "waiting for frame start trigger" acquisition status. In this acquisition status, the camera can react to frame start trigger signals and will begin to expose a frame each time a proper frame start trigger signal is applied.

A primary feature of the acquisition start trigger is that after an acquisition start trigger signal has been applied to the camera and the camera has entered the "waiting for frame start trigger" acquisition status, the camera will return to the "waiting for acquisition start trigger" acquisition status once a specified number of frame start triggers has been received. Before more frames can be acquired, a new acquisition start trigger signal must be applied to the camera to exit it from "waiting for acquisition start trigger" status. Note that this feature only applies when the Trigger Mode parameter for the acquisition start trigger is set to on. This feature is explained in greater detail in the following sections.

9.4.1   Acquisition Start Trigger Mode (Standard Mode)

The main parameter associated with the acquisition start trigger is the Trigger Mode parameter. The Trigger Mode parameter for the acquisition start trigger has two available settings: off and on.

9.4.1.1   Acquisition Start Trigger Mode = Off

When the Trigger Mode parameter for the acquisition start trigger is set to off, the camera will generate all required acquisition start trigger signals internally, and you do not need to apply acquisition start trigger signals to the camera.

9.4.1.2   Acquisition Start Trigger Mode = On

When the Trigger Mode parameter for the acquisition start trigger is set to on, the camera will initially be in a "waiting for acquisition start trigger" acquisition status and cannot react to frame start trigger signals. You must apply an acquisition start trigger signal to the camera to exit the camera from the "waiting for acquisition start trigger" acquisition status and enter the "waiting for frame start trigger" acquisition status. The camera can then react to frame start trigger signals and will continue to do so until the number of frame start trigger signals it has received is equal to the current Acquisition Frame Count parameter setting. The camera will then return to the "waiting for acquisition start trigger" acquisition status. In order to acquire more frames, you must apply a new acquisition start trigger signal to the camera to exit it from the "waiting for acquisition start trigger" acquisition status.

When the Trigger Mode parameter for the acquisition start trigger is set to on, you must select a source signal to serve as the acquisition start trigger. The Trigger Source parameter specifies the source signal. The available selections for the Trigger Source parameter are:

- Software - When the source signal is set to software, you apply an acquisition start trigger signal to the camera by executing an Trigger Software command for the acquisition start trigger on the host PC.

- Line 1 - When the source signal is set to line 1, you apply an acquisition start trigger signal to the camera by injecting an externally generated electrical signal (commonly referred to as a hardware trigger signal) into physical input line 1 on the camera.

If the Trigger Source parameter for the acquisition start trigger is set to Line 1, you must also set the Trigger Activation parameter. The available settings for the Trigger Activation parameter are:

- Rising Edge - specifies that a rising edge of the electrical signal will act as the acquisition start trigger.

- Falling Edge - specifies that a falling edge of the electrical signal will act as the acquisition start trigger.

> **When the Trigger Mode parameter for the acquisition start trigger is set to on, the camera's Acquisition Mode parameter must be set to continuous.**

> By default, input line 2 is selected as the source signal for the acquisition start trigger.
>
> Keep in mind that the camera will only react to acquisition start triggers when it is in a "waiting for acquisition start trigger" acquisition status. For more information about the acquisition status, see Section 9.2 on and Section 9.4 on .

9.4.2 Acquisition Frame Count (Standard Mode)

When the Trigger Mode parameter for the acquisition start trigger is set to on, you must set the value of the camera's Acquisition Frame Count parameter. The value of the Acquisition Frame Count can range from 1 to 255.

With acquisition start triggering on, the camera will initially be in a "waiting for acquisition start trigger" acquisition status. When in this acquisition status, the camera cannot react to frame start trigger signals. If an acquisition start trigger signal is applied to the camera, the camera will exit the "waiting for acquisition start trigger" acquisition status and will enter the "waiting for frame start trigger" acquisition status. It can then react to frame start trigger signals. When the camera has received a number of frame start trigger signals equal to the current Acquisition Frame Count parameter setting, it will return to the "waiting for acquisition start trigger" acquisition status. At that point, you must apply a new acquisition start trigger signal to exit the camera from the "waiting for acquisition start trigger" acquisition status.

For more information about the pylon Viewer, see Section 3.1.1 on .

9.4.3   Setting the Acquisition Start Trigger Mode and Related Parameters (Standard Mode)

You can set the Trigger Mode and Trigger Source parameters for the acquisition start trigger and also set the Acquisition Frame Count parameter value from within your application software by using the Basler pylon API.

The following code snippet illustrates using the API to set the Trigger Mode to on, the Trigger Source to software, and the Acquisition Frame Count to 5:

```
// Set the acquisition mode to continuous (the acquisition mode must
// be set to continuous when acquisition start triggering is on)
Camera.AcquisitionMode.SetValue( AcquisitionMode_Continuous );

// Select the acquisition start trigger
Camera.TriggerSelector.SetValue( TriggerSelector_AcquisitionStart );
// Set the mode for the selected trigger
Camera.TriggerMode.SetValue( TriggerMode_On );
// Set the source for the selected trigger
Camera.TriggerSource.SetValue ( TriggerSource_Software );
// Set the acquisition frame count
Camera.AcquisitionFrameCount.SetValue( 5 );
```

The following code snippet illustrates using the API to set the Trigger Mode to on, the Trigger Source to line 1, the Trigger Activation to rising edge, and the Acquisition Frame Count to 5:

```
// Set the acquisition mode to continuous (the acquisition mode must
// be set to continuous when acquisition start triggering is on)
Camera.AcquisitionMode.SetValue( AcquisitionMode_Continuous );

// Select the acquisition start trigger
Camera.TriggerSelector.SetValue( TriggerSelector_AcquisitionStart );
// Set the mode for the selected trigger
Camera.TriggerMode.SetValue( TriggerMode_On );
// Set the source for the selected trigger
Camera.TriggerSource.SetValue ( TriggerSource_Line1 );
// Set the activation mode for the selected trigger to rising edge
Camera.TriggerActivation.SetValue( TriggerActivation_RisingEdge );
// Set the acquisition frame count
Camera.AcquisitionFrameCount.SetValue( 5 );
```

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

For more information about the pylon Viewer, see Section 3.1.1 on .

9.4.4   Using a Software Acquisition Start Trigger (Standard Mode)

9.4.4.1   Introduction

If the camera's Acquisition Start Trigger Mode parameter is set to on and the Acquisition Start Trigger Source parameter is set to software, you must apply a software acquisition start trigger signal to the camera before you can begin frame acquisition.

A software acquisition start trigger signal is applied by:

- Setting the Trigger Selector parameter to Acquisition Start.
- Executing a Trigger Software command.

The camera will initially be in a "waiting for acquisition start trigger" acquisition status. It cannot react to frame trigger signals when in this acquisition status. When a software acquisition start trigger signal is received by the camera, it will exit the "waiting for acquisition start trigger" acquisition status and will enter the "waiting for frame start trigger" acquisition status. It can then react to frame start trigger signals. When the number of frame start trigger signals received by the camera is equal to the current Acquisition Frame Count parameter setting, the camera will return to the "waiting for acquisition start trigger" acquisition status. When a new software acquisition start trigger signal is applied to the camera, it will again exit from the "waiting for acquisition start trigger" acquisition status and enter the "waiting for frame start trigger" acquisition status.

(Note that as long as the Trigger Selector parameter is set to Acquisition Start, a software acquisition start trigger will be applied to the camera each time a Trigger Software command is executed.)

9.4.4.2   Setting the Parameters Related to Software Acquisition Start Triggering and Applying a Software Trigger Signal

You can set all of the parameters needed to perform software acquisition start triggering from within your application software by using the Basler pylon API. The following code snippet illustrates using the API to set the parameter values and to execute the commands related to software acquisition start triggering with the camera set for continuous frame acquisition mode:

```
// Set the acquisition mode to continuous (the acquisition mode must
// be set to continuous when acquisition start triggering is on)
Camera.AcquisitionMode.SetValue( AcquisitionMode_Continuous );

// Select the acquisition start trigger
Camera.TriggerSelector.SetValue( TriggerSelector_AcquisitionStart );
// Set the mode for the selected trigger
Camera.TriggerMode.SetValue( TriggerMode_On );
// Set the source for the selected trigger
Camera.TriggerSource.SetValue ( TriggerSource_Software );
// Set the acquisition frame count
```

```
    Camera.AcquisitionFrameCount.SetValue( 5 );
// Execute an acquisition start command to prepare for frame acquisition
    Camera.AcquisitionStart.Execute( );
    while ( ! finished )
    {
        // Execute a trigger software command to apply a software acquisition
        // start trigger signal to the camera
        Camera.TriggerSoftware.Execute( );

        // Perform the required functions to parameterize the frame start
        // trigger, to trigger 5 frame starts, and to retrieve 5 frames here
    }
    Camera.AcquisitionStop.Execute( );

// Note: as long as the Trigger Selector is set to Acquisition Start, executing
// a Trigger Software command will apply a software acquisition start trigger
// signal to the camera
```

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

9.4.5   Using a Hardware Acquisition Start Trigger (Standard Mode)

9.4.5.1   Introduction

If the Trigger Mode parameter for the acquisition start trigger is set to on and the Trigger Source parameter is set to line 1, an externally generated electrical signal injected into physical input line 1 on the camera will act as the acquisition start trigger signal for the camera. This type of trigger signal is generally referred to as a hardware trigger signal or as an external acquisition start trigger signal (ExASTrig).

A rising edge or a falling edge of the ExASTrig signal can be used to trigger acquisition start. The Trigger Activation parameter is used to select rising edge or falling edge triggering.

When the Trigger Mode parameter is set to on, the camera will initially be in a "waiting for acquisition start trigger" acquisition status. It cannot react to frame start trigger signals when in this acquisition status. When the appropriate ExASTrig signal is applied to line 1 (e.g, a rising edge of the signal for rising edge triggering), the camera will exit the "waiting for acquisition start trigger" acquisition status and will enter the "waiting for frame start trigger" acquisition status. It can then react to frame start trigger signals. When the number of frame start trigger signals received by the camera is equal to the current Acquisition Frame Count parameter setting, the camera will return to the "waiting for acquisition start trigger" acquisition status. When a new ExASTrig signal is applied to line 1, the camera will again exit from the "waiting for acquisition start trigger" acquisition status and enter the "waiting for frame start trigger" acquisition status.

For more information about setting the camera for hardware acquisition start triggering and selecting the input line to receive the ExASTrig signal, see Section 9.4.5.2 on page 113.

For more information about the electrical requirements for input lines 1 and 2, see Section 7.6.1 on page 81.

9.4.5.2   Setting the Parameters Related to Hardware Acquisition Start Triggering and Applying a Hardware Trigger Signal

You can set all of the parameters needed to perform hardware acquisition start triggering from within your application by using the Basler pylon API. The following code snippet illustrates using the API to set the parameter values required to enable rising edge hardware acquisition start triggering with line 1 as the trigger source:

```
// Set the acquisition mode to continuous (the acquisition mode must
// be set to continuous when acquisition start triggering is on)
Camera.AcquisitionMode.SetValue( AcquisitionMode_Continuous );

// Select the acquisition start trigger
Camera.TriggerSelector.SetValue( TriggerSelector_AcquisitionStart );
// Set the mode for the selected trigger
Camera.TriggerMode.SetValue( TriggerMode_On );
// Set the source for the selected trigger
Camera.TriggerSource.SetValue ( TriggerSource_Line1 );
// Set the activation mode for the selected trigger to rising edge
Camera.TriggerActivation.SetValue( TriggerActivation_RisingEdge );
// Set the acquisition frame count
Camera.AcquisitionFrameCount.SetValue( 5 );
// Execute an acquisition start command to prepare for frame acquisition
Camera.AcquisitionStart.Execute( );
    while ( ! finished )
    {
        // Apply a rising edge of the externally generated electrical signal
        // (ExASTrig signal) to input line 1 on the camera

            // Perform the required functions to parameterize the frame start
            // trigger, to trigger 5 frame starts, and to retrieve 5 frames here
    }
    Camera.AcquisitionStop.Execute( );
```

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

For more information about the pylon Viewer, see Section 3.1.1 on page 43.

9.5   The Frame Start Trigger in Standard Mode

	This section only applies if the standard mode is enabled for image acquisition control.
(i)	When the camera is started for the first time after delivery from the factory the image acquisition control will not be in standard mode but in legacy mode.
	Use the legacy mode only if you want to operate the camera together with previous cameras not featuring the standard mode.

For more information about standard mode and legacy mode and how to set them, see Section 9.1 on page 99.

The frame start trigger is used to begin frame acquisition. Assuming that the camera is in a "waiting for frame start trigger" acquisition status, it will begin a frame acquisition each time it receives a frame start trigger signal.

Note that in order for the camera to be in a "waiting for frame start trigger" acquisition status:

■ The Acquisition Mode parameter must be set correctly.

■ A proper Acquisition Start command must be applied to the camera.

■ A proper acquisition start trigger signal must be applied to the camera (if the Trigger Mode parameter for the acquisition start trigger is set to on).

For more information about the Acquisition Mode parameter and about Acquisition Start and Acquisition Stop commands, see Section 9.2 on page 101 and Section 9.3 on page 105.

For more information about the acquisition start trigger, and about the acquisition status, see Section 9.2 on page 101 and Section 9.4 on page 107.

Referring to the use case diagrams that appear in Section 9.8 on page 140 can help you understand the explanations of the frame start trigger.

9.5.1   Frame Start Trigger Mode (Standard Mode)

The main parameter associated with the frame start trigger is the Trigger Mode parameter. The Trigger Mode parameter for the frame start trigger has two available settings: off and on.

9.5.1.1   Frame Start Trigger Mode = Off

When the Frame Start Trigger Mode parameter is set to off, the camera will generate all required frame start trigger signals internally, and you do not need to apply frame start trigger signals to the camera.

With the trigger mode set to off, the way that the camera will operate the frame start trigger depends on the setting of the camera's Acquisition Mode parameter:

- If the Acquisition Mode parameter is set to single frame, the camera will automatically generate a single frame start trigger signal whenever it receives an Acquisition Start command.

- If the Acquisition Mode parameter is set to continuous frame, the camera will automatically begin generating frame start trigger signals when it receives an Acquisition Start command. The camera will continue to generate frame start trigger signals until it receives an Acquisition Stop command.

  The rate at which the frame start trigger signals are generated may be determined by the camera's Acquisition Frame Rate Abs parameter:

  - If the parameter is not enabled, the camera will generate frame start trigger signals at the maximum rate allowed with the current camera settings.

  - If the parameter is enabled and is set to a value less than the maximum allowed frame rate with the current camera settings, the camera will generate frame start trigger signals at the rate specified by the parameter setting.

  - If the parameter is enabled and is set to a value greater than the maximum allowed frame rate with the current camera settings, the camera will generate frame start trigger signals at the maximum allowed frame rate.

For information about setting the Acquisition Frame Rate Abs parameter, see Section 9.5.1.3 on .

(i)	Keep in mind that the camera will only react to frame start triggers when it is in a "waiting for frame start trigger" acquisition status. For more information about the acquisition status, see Section 9.2 on page 101 and Section 9.4 on page 107.

Exposure Time Control with the Frame Start Trigger Off

When the Trigger Mode parameter for the frame start trigger is set to off, the exposure time for each frame acquisition is determined by the value of the camera's Exposure Time Abs parameter.

For more information about the camera's Exposure Time Abs parameter, see Section 9.7 on .

9.5.1.2   Frame Start Trigger Mode = On

When the Trigger Mode parameter for the frame start trigger is set to on, you must apply a frame start trigger signal to the camera each time you want to begin a frame acquisition.

	Do not trigger frame acquisition at a rate that
(i)	■ exceeds the maximum allowed for the current camera settings. If you apply frame start trigger signals to the camera when it is not ready to receive them, the signals will be ignored. For more information about determining the maximum allowed frame rate, see Section 9.13 on page 163 and Section 9.14 on page 167.
	■ exceeds the host computer's capacity limits for data transfer or storage or both. If you try to acquire more images than the host computer is able to process, frames may be dropped. For more information about bandwidth optimization, see the *Installation and Setup Guide for Cameras Used with Basler pylon for Windows* (AW000611).

The Trigger Source parameter specifies the source signal that will act as the frame start trigger signal. The available selections for the Trigger Source parameter are:

■ Software - When the source signal is set to software, you apply a frame start trigger signal to the camera by executing a Trigger Software command for the frame start trigger on the host PC.

■ Line 1 - When the source signal is set to line 1, you apply a frame start trigger signal to the camera by injecting an externally generated electrical signal (commonly referred to as a hardware trigger signal) into physical input line 1 on the camera.

If the Trigger Source parameter is set to Line 1, you must also set the Trigger Activation parameter. The available settings for the Trigger Activation parameter are:

■ Rising Edge - specifies that a rising edge of the electrical signal will act as the frame start trigger.

■ Falling Edge - specifies that a falling edge of the electrical signal will act as the frame start trigger.

For more information about using a software trigger to control frame acquisition start, see Section 9.4.4 on page 111.

For more information about using a hardware trigger to control frame acquisition start, see Section 9.4.5 on page 112.

(i)	By default, input line 1 is selected as the source signal for the frame start trigger.
	Keep in mind that the camera will only react to frame start triggers when it is in a "waiting for frame start trigger" acquisition status. For more information about the acquisition status, see Section 9.2 on page 101 and Section 9.4 on page 107.

Exposure Time Control with the Frame Start Trigger On

When the Trigger Mode parameter for the frame start trigger is set to on and the Trigger Source parameter is set to software, the exposure time for each frame acquisition is determined by the camera's exposure time parameters.

When the Trigger Mode parameter is set to on and the Trigger Source parameter is set to one of the input lines, the exposure time for each frame acquisition can be controlled with the exposure time parameters or it can be controlled by manipulating the hardware trigger signal.

For more information about controlling exposure time when using a software trigger, see Section 9.4.4 on .

For more information about controlling exposure time when using a hardware trigger, see Section 9.4.5 on .

For more information about exposure time parameters, see Section 9.7 on .

9.5.1.3   Setting the Frame Start Trigger Mode and Related Parameters

You can set the Trigger Mode and related parameter values for the frame start trigger from within your application software by using the Basler pylon API. If your settings make it necessary, you can also set the Trigger Source parameter.

The following code snippet illustrates using the API to set the Trigger Mode for the frame start trigger to on and the Trigger Source to input line 1:

```
// Select the frame start trigger
Camera.TriggerSelector.SetValue( TriggerSelector_FrameStart );
// Set the mode for the selected trigger
Camera.TriggerMode.SetValue( TriggerMode_On );
// Set the source for the selected trigger
Camera.TriggerSource.SetValue ( TriggerSource_Line1 );
```

The following code snippet illustrates using the API to set the Acquisition Mode to continuous, the Trigger Mode to off, and the Acquisition Frame Rate to 60:

```
// Set the acquisition mode to continuous frame
Camera.AcquisitionMode.SetValue( AcquisitionMode_Continuous );
// Select the frame start trigger
Camera.TriggerSelector.SetValue( TriggerSelector_FrameStart );
// Set the mode for the selected trigger
Camera.TriggerMode.SetValue( TriggerMode_Off );
// Set the exposure time
Camera.ExposureTimeAbs.SetValue( 3000 );
// Enable the acquisition frame rate parameter and set the frame rate. (Enabling
// the acquisition frame rate parameter allows the camera to control the frame
// rate internally.)
```

```
Camera.AcquisitionFrameRateEnable.SetValue( true );
Camera.AcquisitionFrameRateAbs.SetValue( 60.0 );
// Start frame capture
Camera.AcquisitionStart.Execute( );
```

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

For more information about the pylon Viewer, see Section 3.1.1 on .

9.5.2   Using a Software Frame Start Trigger (Standard Mode)

9.5.2.1   Introduction

If the Trigger Mode parameter for the frame start trigger is set to on and the Trigger Source parameter is set to software, you must apply a software frame start trigger signal to the camera to begin each frame acquisition. Assuming that the camera is in a "waiting for frame start trigger" acquisition status, frame exposure will start when the software frame start trigger signal is received by the camera. Figure 46 illustrates frame acquisition with a software frame start trigger signal.

When the camera receives a software trigger signal and begins exposure, it will exit the "waiting for frame start trigger" acquisition status because at that point, it cannot react to a new frame start trigger signal. As soon as the camera is capable of reacting to a new frame start trigger signal, it will automatically return to the "waiting for frame start trigger" acquisition status.

When you are using a software trigger signal to start each frame acquisition, the camera's Exposure Mode parameter must be set to timed. The exposure time for each acquired frame will be determined by the value of the camera's exposure time parameters.
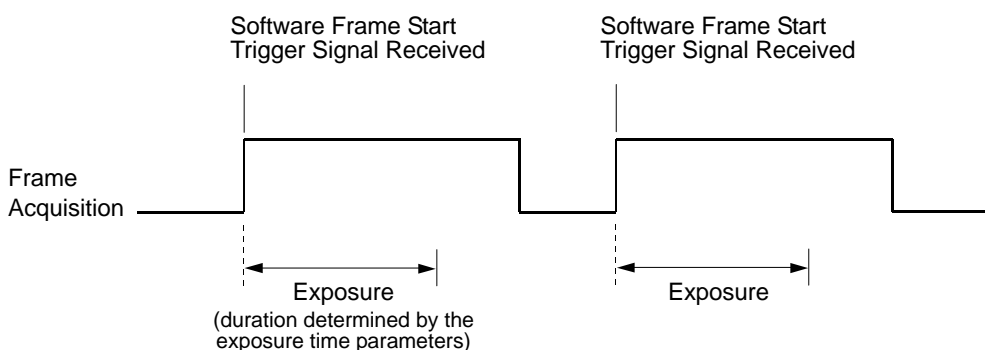
Fig. 46: Frame Acquisition with a Software Frame Start Trigger

9.5.2.2   Setting the Parameters Related to Software Frame Start Triggering and Applying a Software Trigger Signal

You can set all of the parameters needed to perform software frame start triggering from within your application software by using the Basler pylon API. The following code snippet illustrates using the API to set the parameter values and to execute the commands related to software frame start triggering with the camera set for continuous frame acquisition mode. In this example, the trigger mode for the acquisition start trigger will be set to off:

```
// Set the acquisition mode to continuous frame
Camera.AcquisitionMode.SetValue( AcquisitionMode_Continuous );
// Select the acquisition start trigger
Camera.TriggerSelector.SetValue( TriggerSelector_AcquisitionStart );
// Set the mode for the selected trigger
Camera.TriggerMode.SetValue( TriggerMode_Off );
// Disable the acquisition frame rate parameter (this will disable the camera's
// internal frame rate control and allow you to control the frame rate with
// software frame start trigger signals within the limits imposed by other
// parameter settings).
Camera.AcquisitionFrameRateEnable.SetValue( false );
// Select the frame start trigger
Camera.TriggerSelector.SetValue( TriggerSelector_FrameStart );
// Set the mode for the selected trigger
Camera.TriggerMode.SetValue( TriggerMode_On );
// Set the source for the selected trigger
Camera.TriggerSource.SetValue ( TriggerSource_Software );
// Set for the timed exposure mode
Camera.ExposureMode.SetValue( ExposureMode_Timed );
// Set the exposure time
Camera.ExposureTimeAbs.SetValue( 3000 );
// Execute an acquisition start command to prepare for frame acquisition
Camera.AcquisitionStart.Execute( );
while ( ! finished )
{
// Execute a Trigger Software command to apply a frame start
// trigger signal to the camera
Camera.TriggerSoftware.Execute( );
// Retrieve acquired frame here
}
    Camera.AcquisitionStop.Execute( );

// Note: as long as the Trigger Selector is set to FrameStart, executing
// a Trigger Software command will apply a software frame start trigger
// signal to the camera
```

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

For more information about the pylon Viewer, see Section 3.1.1 on .

9.5.3 Using a Hardware Frame Start Trigger (Standard Mode)

9.5.3.1 Introduction

If the Trigger Mode parameter for the frame start trigger is set to on and the Trigger Source parameter is set to e.g. input line 1, an externally generated electrical signal applied to input line 1 on the camera will act as the frame start trigger signal for the camera. This type of trigger signal is generally referred to as a hardware trigger signal or as an external frame start trigger signal (ExFSTrig).

A rising edge or a falling edge of the ExFSTrig signal can be used to trigger frame acquisition. The Trigger Activation parameter is used to select rising edge or falling edge triggering.

Assuming that the camera is in a "waiting for frame start trigger" acquisition status, frame acquisition will start whenever the appropriate edge transition is received by the camera.

When the camera receives a hardware trigger signal and begins exposure, it will exit the "waiting for frame start trigger" acquisition status because at that point, it cannot react to a new frame start trigger signal. As soon as the camera is capable of reacting to a new frame start trigger signal, it will automatically return to the "waiting for frame start trigger" acquisition status.

When the camera is operating under control of an ExFSTrig signal, the period of the ExFSTrig signal will determine the rate at which the camera is acquiring frames:

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

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

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

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

For more information about setting the camera for hardware acquisition start triggering and selecting the input line to receive the ExFSTrig signal, see Section 9.4.5.2 on page 113.

For more information about the electrical requirements for input lines 1 and 2, see Section 7.6.1 on page 81.

For more information about determining the maximum allowed frame rate, see Section 9.13 on page 163 and Section 9.14 on page 167.

9.5.3.2   Exposure Modes

If you are triggering the start of frame acquisition with an externally generated frame start trigger (ExFSTrig) signal, two exposure modes are available: timed and trigger width.

Timed Exposure Mode

When timed mode is selected, the exposure time for each frame acquisition is determined by the camera's exposure time parameters. If the camera is set for rising edge triggering, the exposure time starts when the ExFSTrig signal rises. If the camera is set for falling edge triggering, the exposure time starts when the ExFSTrig signal falls. The following figure illustrates timed exposure with the camera set for rising edge triggering.

ExFSTrig Signal Period

ExFSTrig Signal

Exposure
(duration determined by the
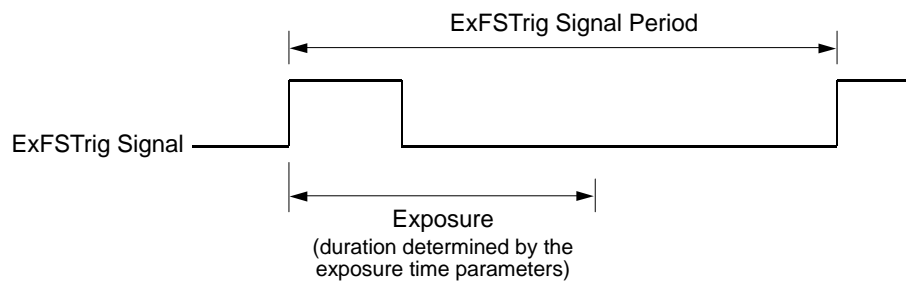exposure time parameters)

Fig. 47: Timed Exposure with Rising Edge Triggering

Note that if you attempt to trigger a new exposure start while the previous exposure is still in progress, the trigger signal will be ignored, and a Frame Start Overtrigger event will be generated.

This situation is illustrated below for rising edge triggering.

This rise in the trigger signal will be
ignored, and a Frame Start Overtrigger
event will be generated

ExFSTrig Signal

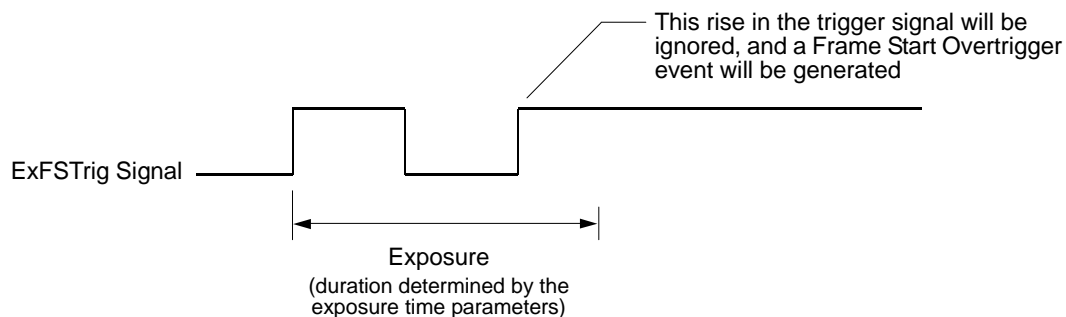Exposure
(duration determined by the
exposure time parameters)

Fig. 48: Overtriggering with Timed Exposure

For more information about the Frame Start Overtrigger event, see Section 12.15 on page 306.

For more information about the camera's exposure time parameters, see Section 9.7 on page 137.

Trigger Width Exposure Mode

When trigger width exposure mode is selected, the length of the exposure for each frame acquisition will be directly controlled by the ExFSTrig signal. If the camera is set for rising edge triggering, the exposure time begins when the ExFSTrig signal rises and continues until the ExFSTrig signal falls. If the camera is set for falling edge triggering, the exposure time begins when the ExFSTrig signal falls and continues until the ExFSTrig signal rises. Figure 49 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 frame.
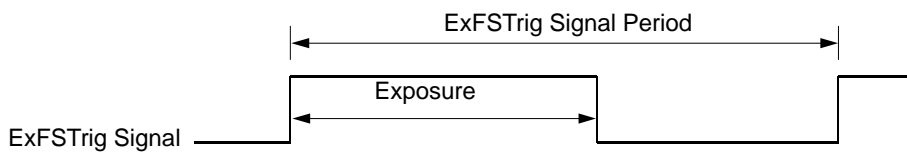
Fig. 49: Trigger Width Exposure with Rising Edge Triggering

When you operate the camera in trigger width exposure mode, you must also use the camera's exposure time prameters to set an exposure time. This parameter 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 mode and that you intend to use the ExFSTrig 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.

For more information about the Trigger Ready signal, see Section 9.11.3 on and Section 9.11.4 on .

For more information about the camera's exposure time parameters, see Section 9.7 on .

9.5.3.3   Frame Start Trigger Delay

The trigger delay feature lets you specify a delay (in microseconds) that will be applied between the receipt of a hardware frame start trigger and when the trigger will become effective.

> The trigger delay will not operate if the Trigger Mode parameter for frame start is set to off or if you are using a software frame start trigger.

For more information about the trigger delay feature and how to use it, see Section 12.13 on .

9.5.3.4   Setting the Parameters Related to Hardware Frame Start Triggering and Applying a Hardware Trigger Signal

You can set all of the parameters needed to perform hardware frame start triggering from within your application by using the Basler pylon API. The following code snippet illustrates using the API to set the camera for single frame acquisition mode with the trigger mode for the acquisition start trigger set to off. We will use the timed exposure mode with input line 1 as the trigger source and with rising edge triggering. In this example, we will use a trigger delay:

```
// Set the acquisition mode to single frame
Camera.AcquisitionMode.SetValue( AcquisitionMode_SingleFrame );
// Select the acquisition start trigger
Camera.TriggerSelector.SetValue( TriggerSelector_AcquisitionStart );
// Set the mode for the selected trigger
Camera.TriggerMode.SetValue( TriggerMode_Off );
// Select the frame start trigger
Camera.TriggerSelector.SetValue( TriggerSelector_FrameStart );
// Set the mode for the selected trigger
Camera.TriggerMode.SetValue( TriggerMode_On );
// Set the source for the selected trigger
Camera.TriggerSource.SetValue ( TriggerSource_Line1 );
// Set the trigger activation mode to rising edge
Camera.TriggerActivation.SetValue( TriggerActivation_RisingEdge );
// Set the trigger delay for one millisecond (1000us == 1ms == 0.001s)
double TriggerDelay_us = 1000.0;
Camera.TriggerDelayAbs.SetValue( TriggerDelay_us );
// Set for the timed exposure mode
Camera.ExposureMode.SetValue( ExposureMode_Timed );
// Set the exposure time
Camera.ExposureTimeAbs.SetValue( 3000 );
// Execute an acquisition start command to prepare for frame acquisition
Camera.AcquisitionStart.Execute( );

// Frame acquisition will start when the externally generated
// frame start trigger signal (ExFSTrig signal) goes high
```

The following code snippet illustrates using the API to set the parameter values and execute the commands related to hardware frame start triggering with the camera set for continuous frame acquisition mode and the trigger mode for the acquisition start trigger set to off. We will use the trigger width exposure mode with input line 1 as the trigger source and with rising edge triggering:

```
// Set the acquisition mode to continuous frame
Camera.AcquisitionMode.SetValue( AcquisitionMode_Continuous );
// Select the acquisition start trigger
Camera.TriggerSelector.SetValue( TriggerSelector_AcquisitionStart );
// Set the mode for the selected trigger
Camera.TriggerMode.SetValue( TriggerMode_Off );
// Disable the acquisition frame rate parameter (this will disable the camera's
```

```
// internal frame rate control and allow you to control the frame rate with
// external frame start trigger signals)
Camera.AcquisitionFrameRateEnable.SetValue( false );
// Select the frame start trigger
Camera.TriggerSelector.SetValue( TriggerSelector_FrameStart );
// Set the mode for the selected trigger
Camera.TriggerMode.SetValue( TriggerMode_On );
// Set the source for the selected trigger
Camera.TriggerSource.SetValue ( TriggerSource_Line1 );
// Set the trigger activation mode to rising edge
Camera.TriggerActivation.SetValue( TriggerActivation_RisingEdge );
// Set for the trigger width exposure mode
Camera.ExposureMode.SetValue( ExposureMode_TriggerWidth );
// Set the shortest exposure time -
// the shortest exposure time we plan to use is 1500 us
Camera.ExposureTimeAbs.SetValue( 1500 );
// Prepare for frame acquisition here
    Camera.AcquisitionStart.Execute( );
    while ( ! finished )
    {
        // Frame acquisition will start each time the externally generated
        // frame start trigger signal (ExFSTrig signal) goes high

        // Retrieve the captured frames
    }
    Camera.AcquisitionStop.Execute( );
```

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

For more information about the pylon Viewer, see Section 3.1.1 on page 43.

9.6　The Acquisition Start Trigger in Legacy Mode

> (i)　This section only applies if the legacy mode is enabled for image acquisition control.
>
> Use the legacy mode only if you want to operate the camera together with previous cameras not featuring the standard mode. Otherwise, we most strongly recommend enabling the standard mode.
>
> When the camera is started for the first time after delivery from the factory the image acquisition control will be in legacy mode.

> (i)　The acquisition start trigger of the legacy mode is called "frame start trigger" in the standard mode.
>
> The acquisition start trigger of the standard mode is not available in the legacy mode.

For more information about standard mode and legacy mode and how to set them, see Section 9.1 on page 99.

The acquisition start trigger is used to begin frame acquisition. Assuming that the camera is in a "waiting for acquisition start trigger" acquisition status, it will begin a frame acquisition each time it receives an acquisition start trigger signal.

Note that in order for the camera to be in a "waiting for acquisition start trigger" acquisition status:

■　The Acquisition Mode parameter must be set correctly.

■　A proper Acquisition Start command must be applied to the camera.

For more information about the Acquisition Mode parameter and about Acquisition Start and Acquisition Stop commands, see Section 9.2 on page 101 and Section 9.3 on page 105.

Referring to the use case diagrams that appear in Section 9.8 on page 140 can help you understand the explanations of the acquisition start trigger. Remember, however, that the diagrams apply to the standard mode. Accordingly, the acquisition start trigger shown in the diagrams is not available in legacy mode and the frame start trigger shown is equivalent to the acquisition start trigger in legacy mode.

9.6.1   Acquisition Start Trigger Mode (Legacy Mode)

The main parameter associated with the acquisition start trigger is the Trigger Mode parameter. The Trigger Mode parameter for the acquisition start trigger has two available settings: off and on.

9.6.1.1   Acquisition Start Trigger Mode = Off

When the Acquisition Start Trigger Mode parameter is set to off, the camera will generate all required acquisition start trigger signals internally, and you do not need to apply acquisition start trigger signals to the camera.

With the trigger mode set to off, the way that the camera will operate the acquisition start trigger depends on the setting of the camera's Acquisition Mode parameter:

■   If the Acquisition Mode parameter is set to single frame, the camera will automatically generate a single acquisition start trigger signal whenever it receives an Acquisition Start command.

■   If the Acquisition Mode parameter is set to continuous frame, the camera will automatically begin generating acquisition start trigger signals when it receives an Acquisition Start command. The camera will continue to generate acquisition start trigger signals until it receives an Acquisition Stop command.

The rate at which the acquisition start trigger signals are generated will be determined by the camera's Acquisition Frame Rate Abs parameter:

■   If the parameter is not enabled, the camera will generate acquisition start trigger signals at the maximum rate allowed with the current camera settings.

■   If the parameter is enabled and is set to a value less than the maximum allowed frame rate with the current camera settings, the camera will generate acquisition start trigger signals at the rate specified by the parameter setting.

■   If the parameter is enabled and is set to a value greater than the maximum allowed frame rate with the current camera settings, the camera will generate acquisition start trigger signals at the maximum allowed frame rate.

For information about setting the Acquisition Frame Rate Abs parameter, see Section 9.6.1.3 on .

	Keep in mind that the camera will only react to acquisition start triggers when it is in a "waiting for acquisition start trigger" acquisition status. For more information about the acquisition status, see Section 9.2 on page 101 and Section 9.4 on page 107.

Exposure Time Control with the Acquisition Start Trigger Mode Off

When the Trigger Mode parameter for the acquisition start trigger is set to off, the exposure time for each frame acquisition is determined by the camera's exposure time parameters.

For more information about the camera's exposure time parameters, see Section 9.7 on .

9.6.1.2   Acquisition Start Trigger Mode = On

When the Trigger Mode parameter for the acquisition start trigger is set to on, you must apply an acquisition start trigger signal to the camera each time you want to begin a frame acquisition. The Trigger Source parameter specifies the source signal that will act as the acquisition start trigger signal. The available selections for the Trigger Source parameter are:

■   Software - When the source signal is set to software, you apply an acquisition start trigger signal to the camera by executing a Trigger Software command for the acquisition start trigger on the host PC.

■   Line 1 or 2 - When the source signal is set e.g. to line 1, you apply an acquisition start trigger signal to the camera by injecting an externally generated electrical signal (commonly referred to as a hardware trigger signal) into physical input line 1 on the camera.

If the Trigger Source parameter is set to Line 1 or Line 2, you must also set the Trigger Activation parameter. The available settings for the Trigger Activation parameter are:

■   Rising Edge - specifies that a rising edge of the electrical signal will act as the acquisition start trigger.

■   Falling Edge - specifies that a falling edge of the electrical signal will act as the acquisition start trigger.

For more information about using a software trigger to control frame acquisition start, see Section 9.4.4 on page 111.

For more information about using a hardware trigger to control frame acquisition start, see Section 9.4.5 on page 112.

> ⓘ   By default, input line 1 is selected as the source signal for the acquisition start trigger.
>
> Keep in mind that the camera will only react to acquisition start triggers when it is in a "waiting for acquisition start trigger" acquisition status. For more information about the acquisition status, see Section 9.2 on page 101 and Section 9.4 on page 107.

Exposure Time Control with the Acquisition Start Trigger Mode On

When the Trigger Mode parameter for the acquisition start trigger is set to on and the Trigger Source parameter is set to software, the exposure time for each frame acquisition is determined by the camera's exposure time parameters.

When the Trigger Mode parameter is set to on and the Trigger Source parameter is set to e.g. input line 1, the exposure time for each frame acquisition can be controlled with the exposure time parameters or it can be controlled by manipulating the hardware trigger signal.

For more information about controlling exposure time when using a software trigger, see Section 9.4.4 on page 111.

For more information about controlling exposure time when using a hardware trigger, see Section 9.4.5 on page 112.

For more information about exposure time parameters, see Section 9.7 on page 137.

9.6.1.3   Setting the Acquisition Start Trigger Mode and Related Parameters

You can set the Trigger Mode and related parameter values for the frame start trigger from within your application software by using the Basler pylon API. If your settings make it necessary, you can also set the Trigger Source parameter.

The following code snippet illustrates using the API to set the Trigger Mode for the acquisition start trigger to on and the Trigger Source to input line 1:

```
// Select the acquisition start trigger
Camera.TriggerSelector.SetValue( TriggerSelector_AcquisitionStart );
// Set the mode for the selected trigger
Camera.TriggerMode.SetValue( TriggerMode_On );
// Set the source for the selected trigger
Camera.TriggerSource.SetValue ( TriggerSource_Line1 );
```

The following code snippet illustrates using the API to set the Acquisition Mode to continuous, the Trigger Mode to off, and the Acquisition Frame Rate to 60:

```
// Set the acquisition mode to continuous frame
Camera.AcquisitionMode.SetValue( AcquisitionMode_Continuous );
// Select the frame start trigger
Camera.TriggerSelector.SetValue( TriggerSelector_AcquisitionStart );
// Set the mode for the selected trigger
Camera.TriggerMode.SetValue( TriggerMode_Off );
// Set the exposure time
Camera.ExposureTimeAbs.SetValue( 3000 );
// Enable the acquisition frame rate parameter and set the frame rate. (Enabling
// the acquisition frame rate parameter allows the camera to control the frame
// rate internally.)
Camera.AcquisitionFrameRateEnable.SetValue( true );
Camera.AcquisitionFrameRateAbs.SetValue( 60.0 );
// Start frame capture
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 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 3.1.1 on page 43.

9.6.2 Using a Software Acquisition Start Trigger (Legacy Mode)

9.6.2.1 Introduction

If the Trigger Mode parameter for the acquisition start trigger is set to on and the Trigger Source parameter is set to software, you must apply a software acquisition start trigger signal to the camera to begin each frame acquisition. Assuming that the camera is in a "waiting for acquisition start trigger" acquisition status, frame exposure will start when the software acquisition start trigger signal is received by the camera. Figure 46 illustrates frame acquisition with a software acquisition start trigger signal.

When the camera receives a software trigger signal and begins exposure, it will exit the "waiting for acquisition start trigger" acquisition status because at that point, it cannot react to a new acquisition start trigger signal. As soon as the camera is capable of reacting to a new acquisition start trigger signal, it will automatically return to the "waiting for acquisition start trigger" acquisition status.

When you are using a software trigger signal to start each frame acquisition, the camera's Exposure Mode parameter must be set to timed. The exposure time for each acquired frame will be determined by the camera's exposure time parameters.
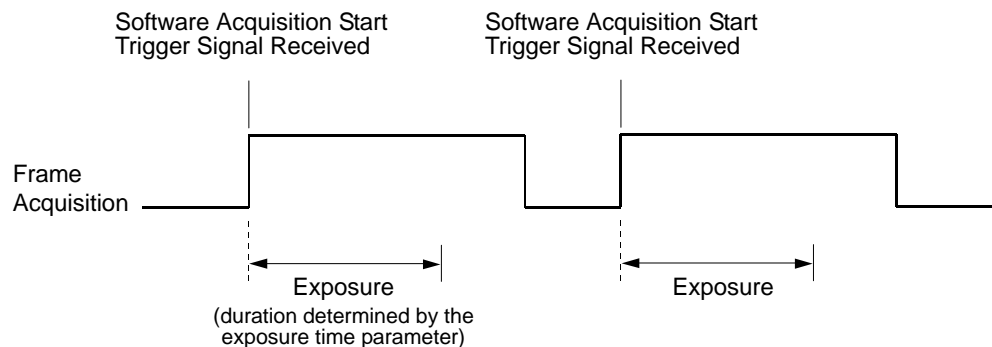

Fig. 50: Frame Acquisition with a Software Acquisition Start Trigger

When you are using a software trigger signal to start each frame acquisition, the frame rate will be determined by how often you apply a software trigger signal to the camera, and you should not attempt to trigger frame acquisition at a rate that exceeds the maximum allowed for the current camera settings. (There is a detailed explanation about the maximum allowed frame rate at the end of this chapter.) Software acquisition start trigger signals that are applied to the camera when it is not ready to receive them will be ignored.

Section 9.5.2.2 on page 119 includes more detailed information about applying a software acquisition start trigger to the camera using Basler pylon.

For more information about determining the maximum allowed frame rate, see Section 9.13 on page 163 and Section 9.14 on page 167.

For more information about exposure time parameters, see Section 9.7 on page 137.

9.6.2.2 Setting the Parameters Related to Software Acquisition Start Triggering and Applying a Software Trigger Signal

You can set all of the parameters needed to perform software acquisition start triggering from within your application software by using the Basler pylon API. The following code snippet illustrates using the API to set the parameter values and to execute the commands related to software acquisition start triggering with the camera set for continuous frame acquisition mode:

```
// Set the acquisition mode to continuous frame
Camera.AcquisitionMode.SetValue( AcquisitionMode_Continuous );
// Select the acquisition start trigger
// Disable the acquisition frame rate parameter (this will disable the camera's
// internal frame rate control and allow you to control the frame rate with
// software frame start trigger signals)
Camera.AcquisitionFrameRateEnable.SetValue( false );
// Select the frame start trigger
Camera.TriggerSelector.SetValue( TriggerSelector_AcquisitionStart );
// Set the mode for the selected trigger
Camera.TriggerMode.SetValue( TriggerMode_On );
// Set the source for the selected trigger
Camera.TriggerSource.SetValue ( TriggerSource_Software );
// Set for the timed exposure mode
Camera.ExposureMode.SetValue( ExposureMode_Timed );
// Set the exposure time
Camera.ExposureTimeAbs.SetValue( 3000 );
// Execute an acquisition start command to prepare for frame acquisition
    Camera.AcquisitionStart.Execute( );
    while ( ! finished )
    {
        // Execute a Trigger Software command to apply an acquisition start
        // trigger signal to the camera
        Camera.TriggerSoftware.Execute( );
        // Retrieve acquired frame here
    }
    Camera.AcquisitionStop.Execute( );

// Note: as long as the Trigger Selector is set to AcquisitionStart, executing
// a Trigger Software command will apply a software acquisition start trigger
// signal to the camera
```

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 3.1.1 on page 43.

9.6.3   Using a Hardware Acquisition Start Trigger (Legacy Mode)

9.6.3.1   Introduction

If the Trigger Mode parameter for the acquisition start trigger is set to on and the Trigger Source parameter is set to e.g. input line 1, an externally generated electrical signal applied to input line 1 on the camera will act as the acquisition start trigger signal for the camera. This type of trigger signal is generally referred to as a hardware trigger signal or as an external acquisition start trigger signal (ExASTrig).

A rising edge or a falling edge of the ExASTrig signal can be used to trigger frame acquisition. The Trigger Activation parameter is used to select rising edge or falling edge triggering.

Assuming that the camera is in a "waiting for acquisition start trigger" acquisition status, frame acquisition will start whenever the appropriate edge transition is received by the camera.

When the camera receives a hardware trigger signal and begins exposure, it will exit the "waiting for acquisition start trigger" acquisition status because at that point, it cannot react to a new acquisition start trigger signal. As soon as the camera is capable of reacting to a new acquisition start trigger signal, it will automatically return to the "waiting for acquisition start trigger" acquisition status.

When the camera is operating under control of an ExASTrig signal, the period of the ExASTrig signal will determine the rate at which the camera is acquiring frames:

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

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

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

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

(i)	If you are triggering frame acquisition with an ExASTrig signal and you attempt to acquire frames at too high a rate, some of the acquisition trigger signals that you apply will be received by the camera when it is not in a "waiting for acquisition start trigger" acquisition status. The camera will ignore any acquisition start trigger signals that it receives when it is not "waiting for acquisition start trigger". (This situation is commonly referred to as "over triggering" the camera. To avoid over triggering, you should not attempt to acquire frames at a rate that exceeds the maximum allowed with the current camera settings.

For more information about setting the camera for hardware acquisition start triggering and selecting the input line to receive the ExASTrig signal, see Section 9.6.3.4 on page 135.

For more information about the electrical requirements for input lines 1 and 2, see Section 7.6.1 on page 81.

For more information about determining the maximum allowed frame rate, see Section 9.13 on page 163 and Section 9.14 on page 167.

9.6.3.2 Exposure Modes

If you are triggering the start of frame acquisition with an externally generated acquisition start trigger (ExASTrig) signal, two exposure modes are available: timed and trigger width.

Timed Exposure Mode

When timed mode is selected, the exposure time for each frame acquisition is determined by the value of the camera's exposure time parameters. If the camera is set for rising edge triggering, the exposure time starts when the ExASTrig signal rises. If the camera is set for falling edge triggering, the exposure time starts when the ExASTrig signal falls. The following figure illustrates timed exposure with the camera set for rising edge triggering.

ExASTrig Signal Period

ExASTrig Signal

Exposure
(duration determined by the exposure time parameters)
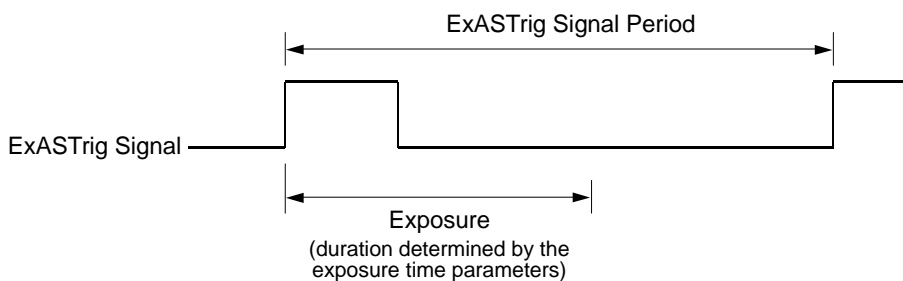
Fig. 51: Timed Exposure with Rising Edge Triggering

Note that if you attempt to trigger a new exposure start while the previous exposure is still in progress, the trigger signal will be ignored, and an Acquisition Start Overtrigger event will be generated. This situation is illustrated below for rising edge triggering.
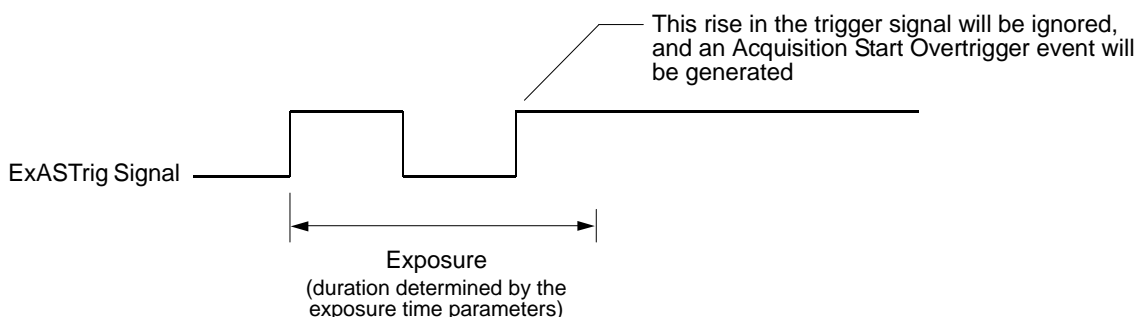
This rise in the trigger signal will be ignored, and an Acquisition Start Overtrigger event will be generated

ExASTrig Signal

Exposure
(duration determined by the exposure time parameters)

Fig. 52: Overtriggering with Timed Exposure

Trigger Width Exposure Mode

When trigger width exposure mode is selected, the length of the exposure for each frame acquisition will be directly controlled by the ExASTrig signal. If the camera is set for rising edge triggering, the exposure time begins when the ExASTrig signal rises and continues until the ExASTrig signal falls. If the camera is set for falling edge triggering, the exposure time begins when the ExASTrig signal falls and continues until the ExASTrig signal rises. The following figure 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 frame.
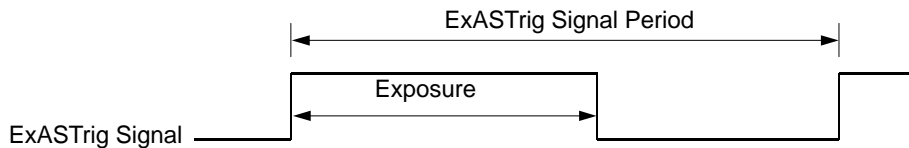
Fig. 53: Trigger Width Exposure with Rising Edge Triggering

When you operate the camera in trigger width exposure mode, you must use the camera's exposure time parameters to set an exposure time. This parameter 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 ExASTrig 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 ExASTrig 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 in Figure 54 for rising edge operation. On the first cycle of the ExASTrig 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 ExASTrig 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 ExASTrig 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 ExASTrig signal as you would expect.
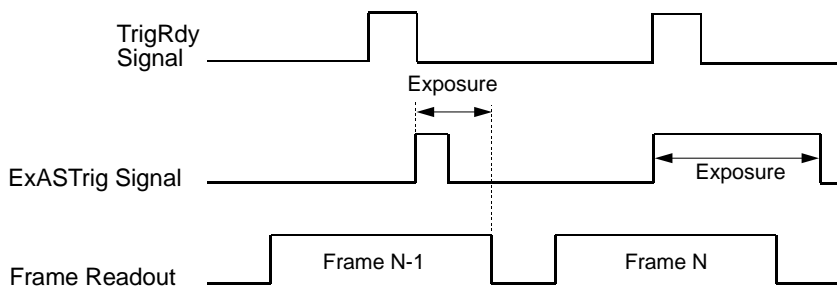
Fig. 54: Trigger Width Exposure Mode with Overlapped Exposure

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.

9.6.3.3   Acquisition Start Trigger Delay

The trigger delay feature lets you specify a delay (in microseconds) that will be applied between the receipt of a hardware acquisition start trigger and when the trigger will become effective.

The trigger delay will not operate if the Trigger Mode parameter for acquisition start is set to off or if you are using a software acquisition start trigger.

For more information about the trigger delay feature and how to set it, see Section 12.13 on .

9.6.3.4   Setting the Parameters Related to Hardware Acquisition Start Triggering and Applying a Hardware Trigger Signal

You can set all of the parameters needed to perform hardware acquisition start triggering from within your application by using the Basler pylon API. The following code snippet illustrates using the API to set the camera for single frame acquisition mode.

We will use the timed exposure mode with input line 1 as the trigger source and with rising edge triggering. In this example, we will use a trigger delay:

```
// Set the acquisition mode to single frame
Camera.AcquisitionMode.SetValue( AcquisitionMode_SingleFrame );
// Select the acquisition start trigger
Camera.TriggerSelector.SetValue( TriggerSelector_AcquisitionStart );
// Set the mode for the selected trigger
Camera.TriggerMode.SetValue( TriggerMode_On );
// Set the source for the selected trigger
Camera.TriggerSource.SetValue ( TriggerSource_Line1 );
// Set the trigger activation mode to rising edge
Camera.TriggerActivation.SetValue( TriggerActivation_RisingEdge );
// Set the trigger delay for one millisecond (1000us == 1ms == 0.001s)
double TriggerDelay_us = 1000.0;
Camera.TriggerDelayAbs.SetValue( TriggerDelay_us );
// Set for the timed exposure mode
Camera.ExposureMode.SetValue( ExposureMode_Timed );
// Set the exposure time
Camera.ExposureTimeAbs.SetValue( 3000 );
// Execute an acquisition start command to prepare for frame acquisition
Camera.AcquisitionStart.Execute( );

// Frame acquisition will start when the externally generated
// acquisition start trigger signal (ExASTrig signal) goes high
```

The following code snippet illustrates using the API to set the parameter values and execute the commands related to hardware acquisition start triggering with the camera set for continuous frame acquisition mode.

We will use the trigger width exposure mode with input line 1 as the trigger source and with rising edge triggering:

```
// Set the acquisition mode to continuous frame
Camera.AcquisitionMode.SetValue( AcquisitionMode_Continuous );
// Disable the acquisition frame rate parameter (this will disable the camera's
// internal frame rate control and allow you to control the frame rate with
// external acquisition start trigger signals)
Camera.AcquisitionFrameRateEnable.SetValue( false );
// Select the acquisition start trigger
Camera.TriggerSelector.SetValue( TriggerSelector_AcquisitionStart );
// Set the mode for the selected trigger
Camera.TriggerMode.SetValue( TriggerMode_On );
```

```
// Set the source for the selected trigger
Camera.TriggerSource.SetValue ( TriggerSource_Line1 );
// Set the trigger activation mode to rising edge
Camera.TriggerActivation.SetValue( TriggerActivation_RisingEdge );
// Set for the trigger width exposure mode
Camera.ExposureMode.SetValue( ExposureMode_TriggerWidth );
// Prepare for frame acquisition here
    Camera.AcquisitionStart.Execute( );
    while ( ! finished )
    {
        // Frame acquisition will start each time the externally generated
        // acquisition start trigger signal (ExASTrig signal) goes high

        // Retrieve the captured frames
    }
    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 3.1.1 on .

9.7   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 11.

The maximum exposure time that can be set also varies by camera model as shown in Table 11.

Camera Model	Minimum Allowed Exposure Time	Maximum Possible Exposure Time
scA640-70 gm/gc	24 µs	10000000 µs
scA640-74 gm/gc	24 µs	10000000 µs
scA640-120 gm/gc	16 µs	10000000 µs
scA750-60 gm/gc	124 µs	126976 µs
scA780-54 gm/gc	26 µs	10000000 µs
scA1000-30 gm/gc	32 µs	10000000 µs
scA1300-32 gm/gc	22 µs	10000000 µs
scA1390-17 gm/gc	34 µs	10000000 µs
scA1400-17 gm/gc	38 µs	10000000 µs
scA1400-30 gm/gc	28 µs	10000000 µs
scA1600-14 gm/gc	31 µs	10000000 µs
scA1600-28 gm/gc	31 µs	10000000 µs

Table 11: Minimum Allowed Exposure Time and Maximum Possible Exposure Time

(i)	Exposure time can not only be manually set (see below), but can also be automatically adjusted.
	Exposure Auto is an auto function and the "automatic" counterpart to manually setting an "absolute" exposure time. The exposure auto function automatically adjusts the Auto Exposure Time Abs parameter value.
	In contrast to the manually set "absolute" exposure time, the automatically adjusted "absolute" exposure time is not restricted to multiples of the current exposure time base. The automatic adjustment is not available when trigger width exposure mode is selected.
	For more information about auto functions, see Section 12.9.1 on page 283.
	For more information about the Exposure Auto function, see Section 12.9.3 on page 292.

For information about parameter settings for obtaining the maximum possible exposure time, see Section 9.7.1 on page 138.

9.7.1 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.

Settings for Obtaining the Maximum Possible Exposure Time

On all camera models except the scA750-60, you can obtain the maximum possible exposure time (10000000 µs) by setting the Exposure Time Raw parameter value to 1 and the Exposure Time Base Abs value to 10000000 µs.

On scA750-60 cameras, you can obtain the maximum possible exposure time (126976 µs) by e.g. setting the exposure time raw parameter value to 2048 and the Exposure Time Base Abs value to 62 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 3.1.1 on page 43.

9.7.2   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 parameters, the camera will automatically change the setting for the Exposure Time Abs parameter to the nearest achievable 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.

Setting the Absolute Exposure Time Parameter

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 3.1.1 on page 43.

9.8   Use Case Diagrams

> (i) This section assumes that the standard mode is enabled for image acquisition control.
>
> In principle, this section also applies if the legacy mode is enabled instead.
> In this case, however, the following differences must be taken into account:
>
> ■   the acquisition start trigger of the standard mode is not available in the legacy mode.
>
> ■   the frame start trigger of the standard mode is called "acquisition start trigger" in the legacy mode.

> (i) When the camera is started for the first time after delivery from the factory the image acquisition control will not be in standard mode but in legacy mode.
>
> Use the legacy mode only if you want to operate the camera together with previous cameras not featuring the standard mode.

For more information about standard mode and legacy mode and how to set them, see Section 9.1 on page 99.

The following pages contain a series of use case descriptions and diagrams. The descriptions and diagrams are designed to illustrate how acquisition start triggering and frame start triggering work in some common situations and with some common combinations of parameter settings.

These use cases do not represent every possible combination of the parameters associated with acquisition start and frame start triggering. They are simply intended to aid you in developing an initial understanding of how these two triggers interact.

In each use case diagram, the black box in the upper left corner indicates how the parameters are set.

> (i) The use case diagrams are representational. They are not drawn to scale and are not designed to accurately describe precise camera timings.

Use Case 1 - Acquisition and Frame Start Triggers Both Off (Free Run)

In this use case, the Acquisition Mode parameter is set to continuous. The Trigger Mode parameter for the acquisition start trigger and the Trigger Mode parameter for the frame start trigger are both set to off. The camera will generate all required acquisition start and frame start trigger signals internally. When the camera is set this way, it will constantly acquire images without any need for triggering by the user. This use case is commonly referred to as "free run".

The rate at which the camera will acquire images will be determined by the camera's Acquisition Frame Rate Abs parameter unless the current camera settings result in a lower frame rate. If the Acquisition Frame Rate Abs parameter is disabled, the camera will acquire frames at the maximum allowed frame rate.

Cameras are used in free run for many applications. One example is for aerial photography. A camera set for free run is used to capture a continuous series of images as an aircraft overflies an area. The images can then be used for a variety of purposes including vegetation coverage estimates, archaeological site identification, etc.

Use Case: "Free Run" (Acquisition Start Trigger Off and Frame Start Trigger Off)

The acquisition start trigger is off. The camera will generate acquisition start trigger signals internally with no action by the user.

The frame start trigger is off. The camera will generate frame start trigger signals internally with no action by the user.

Settings: Acquisition Mode = Continuous
Trigger Mode for the acquisition start trigger = Off
Trigger Mode for the frame start trigger = Off

- - - - = a trigger signal generated by the camera internally

= camera is waiting for an acquisition start trigger

= camera is waiting for a frame start trigger

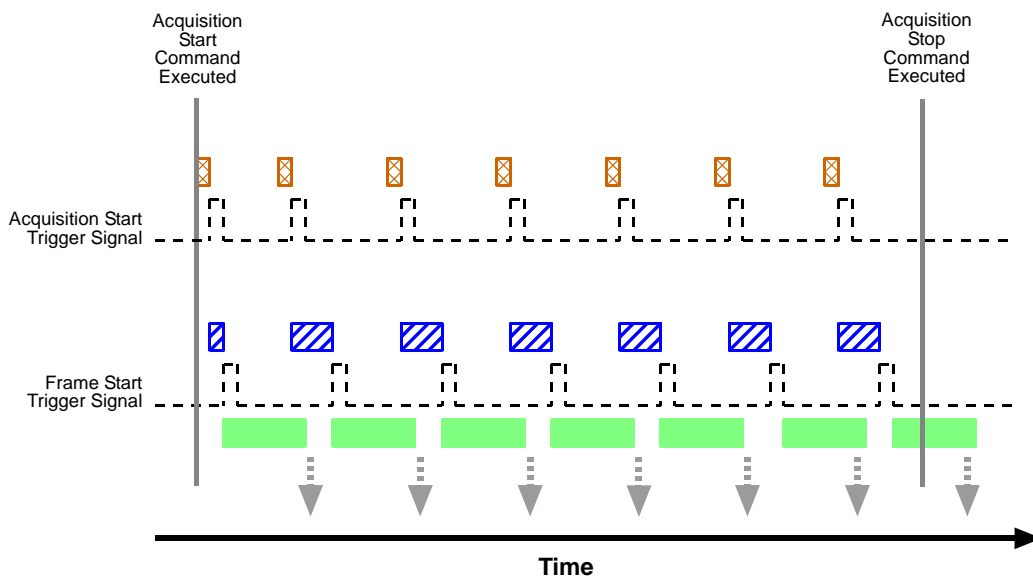= frame exposure and readout

= frame transmission

Fig. 55: Use Case 1 - Acquisition Start Trigger Off and Frame Start Trigger Off

Use Case 2 - Acquisition Start Trigger Off - Frame Start Trigger On

Use case two is illustrated on .

In this use case, the Acquisition Mode parameter is set to continuous. The Trigger Mode parameter for the acquisition start trigger is set to off and the Trigger Mode parameter for the frame start trigger is set to on.

Because the acquisition start trigger is set to off, the user does not need to apply acquisition start trigger signals to the camera. The camera will generate all required acquisition start trigger signals internally.

Because the frame start trigger is set to on, the user must apply a frame start trigger signal to the camera in order to begin each frame exposure. In this case, we have set the frame start trigger signal source to input line 1 and the activation to rising edge, so the rising edge of an externally generated electrical signal applied to line 1 will serve as the frame start trigger signal.

This type of camera setup is used frequently in industrial applications. One example might be a wood products inspection system used to inspect the surface of pieces of plywood on a conveyor belt as they pass by a camera. In this situation, a sensing device is usually used to determine when a piece of plywood on the conveyor is properly positioned in front of the camera. When the plywood is in the correct position, the sensing device transmits an electrical signal to input line 1 on the camera. When the electrical signal is received on line 1, it serves as a frame start trigger signal and initiates a frame acquisition. The frame acquired by the camera is forwarded to an image processing system, which will inspect the image and determine if there are any defects in the plywood's surface.

Use Case:	Acquisition Start Trigger Off and Frame Start Trigger On
	The acquisition start trigger is off. The camera will generate acquisition start trigger signals internally with no action by the user.
	The frame start trigger is on, and the frame start trigger source is set to input line 1. The user must apply a frame start trigger signal to input line 1 to start each frame exposure.
Settings:	Acquisition Mode = Continuous
	Trigger Mode for the acquisition start trigger = Off
	Trigger Mode for the frame start trigger = On
	Trigger Source for the frame start trigger = Line 1
	Trigger Activation for the frame start trigger = Rising Edge

- - - - = a trigger signal generated by the camera internally

———— = a trigger signal applied by the user

= camera is waiting for an acquisition start trigger signal

= camera is waiting for a frame start trigger signal

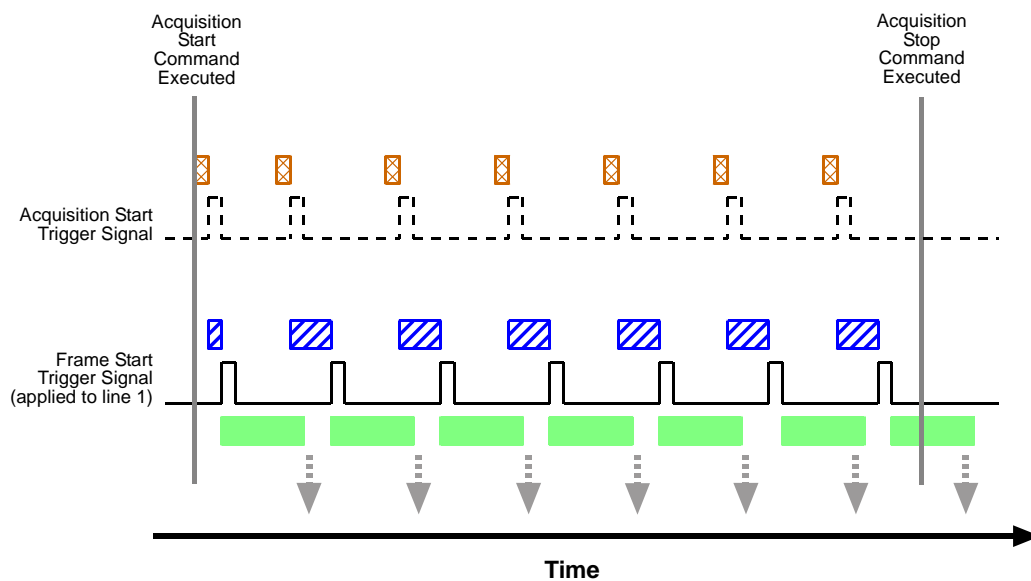= frame exposure and readout

= frame transmission

Fig. 56: Use Case 2 - Acquisition Start Trigger Off and Frame Start Trigger On

Use Case 3 - Acquisition Start Trigger On - Frame Start Trigger Off

Use case three is illustrated on .

In this use case, the Acquisition Mode parameter is set to continuous. The Trigger Mode parameter for the acquisition start trigger is set to on and the Trigger Mode parameter for the frame start trigger is set to off.

Because the acquisition start trigger mode is set to on, the user must apply an acquisition start trigger signal to the camera. In this case, we have set the acquisition start trigger signal source to input line 1 and the activation to rising edge, so an externally generated electrical signal applied to input line 1 will serve as the acquisition start trigger signal. The Acquisition Frame Count parameter has been set to 3.

When a rising edge of the electrical signal is applied to input line 1, the camera will exit the "waiting for acquisition start trigger" acquisition status and enter a "waiting for frame start trigger" acquisition status. Once the camera has acquired 3 frames, it will re-enter the "waiting for acquisition start trigger" acquisition status. Before any more frames can be acquired, a new rising edge must be applied to input line 1 to make the camera exit the "waiting for acquisition start trigger" acquisition status.

Because the frame start trigger is set to off, the user does not need to apply frame start trigger signals to the camera. The camera will generate all required frame start trigger signals internally. The rate at which the frame start trigger signals will be generated is normally determined by the camera's Acquisition Frame Rate Abs parameter. If the Acquisition Frame Rate Abs parameter is disabled, the camera will acquire frames at the maximum allowed frame rate.

This type of camera setup is used frequently in intelligent traffic systems. With these systems, a typical goal is to acquire several images of a car as it passes through a toll booth. A sensing device is usually placed at the start of the toll booth area. When a car enters the area, the sensing device applies an electrical signal to input line 1 on the camera. When the electrical signal is received on input line 1, it serves as an acquisition start trigger signal and the camera exits from the "waiting for acquisition start trigger" acquisition status and enters a "waiting for frame trigger" acquisition status. In our example, the next 3 frame start trigger signals internally generated by the camera would result in frame acquisitions. At that point, the number of frames acquired would be equal to the setting for the Acquisition Frame Count parameter. The camera would return to the "waiting for acquisition start trigger" acquisition status and would no longer react to frame start trigger signals. It would remain in this condition until the next car enters the booth area and activates the sensing device.

This sort of setup is very useful for traffic system applications because multiple frames can be acquired with only a single acquisition start trigger signal pulse and because frames will not be acquired when there are no cars passing through the booth (this avoids the need to store images of an empty toll booth area.)

For more information about the Acquisition Frame Rate Abs parameter, see Section 9.5.1.1 on and for information about setting the parameter, see Section 9.5.1.3 on .

Use Case: Acquisition Start Trigger On and Frame Start Trigger Off

The acquisition start trigger is on, and the acquisition start trigger source is set to input line 1. The user must apply an acquisition start trigger signal to input line 1 to make the camera exit the "waiting for acquisition start trigger" acquisition status. Because the acquisition frame count is set to 3, the camera will re-enter the "waiting for acquisition start trigger" acquisition status after 3 frames have been acquired.

The frame start trigger is off. The camera will generate frame start trigger signals internally with no action by the user.

Settings: Acquisition Mode = Continuous
Trigger Mode for the acquisition start trigger = On
Trigger Source for the acquisition start trigger = Line 1
Trigger Activation for the acquisition start trigger = Rising Edge
Acquisition Frame Count = 3
Trigger Mode for the frame start trigger = Off

- - - - = a trigger signal generated by the camera internally

——— = a trigger signal applied by the user

⬚⬚⬚ = camera is waiting for an acquisition start trigger signal

⬚⬚ = camera is waiting for a frame start trigger signal

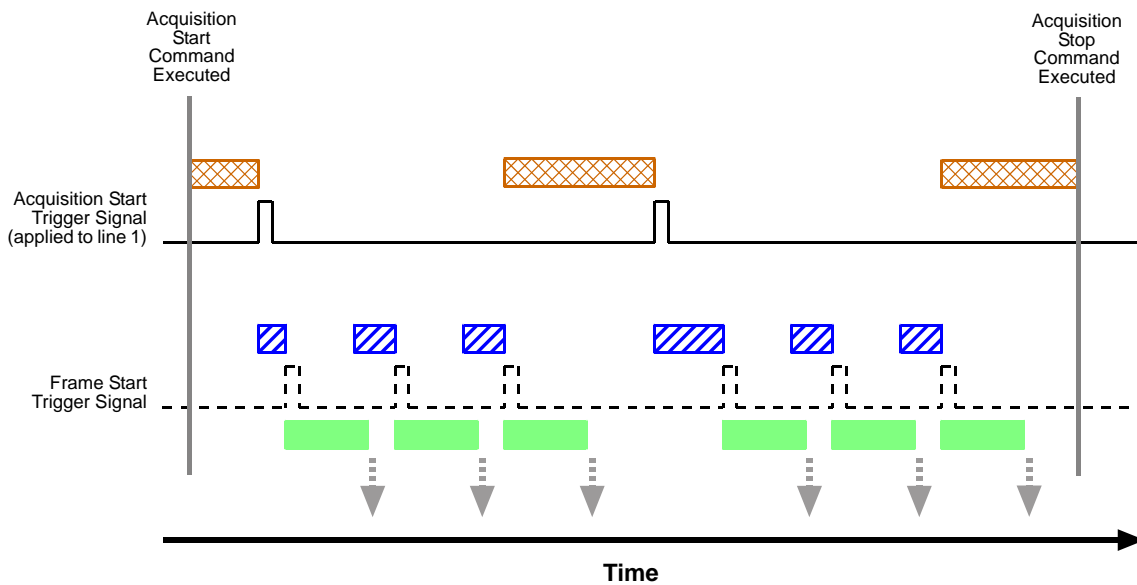▆ = frame exposure and readout

▼ = frame transmission

Fig. 57: Use Case 3 - Acquisition Start Trigger On and Frame Start Trigger Off

Use Case 4 - Acquisition Start and Frame Start Triggers Both On

In this use case, the Acquisition Mode parameter is set to continuous. The Trigger Mode parameter for the acquisition start trigger is set to on and the Trigger Mode parameter for the frame start trigger is set to on.

Because the acquisition start trigger mode is set to on, the user must apply an acquisition start trigger signal to the camera. In this case, we have set the acquisition start trigger signal source to software, so the execution of an acquisition trigger software command will serve as the acquisition start trigger signal. The Acquisition Frame Count parameter is set to 3.

When an acquisition trigger software command is executed, the camera will exit the "waiting for acquisition start trigger" acquisition status and enter a "waiting for frame start trigger" acquisition status. Once the camera has acquired 3 frames, it will re-enter the "waiting for acquisition start trigger" acquisition status. Before any more frames can be acquired, a new acquisition trigger software command must be executed to make the camera exit the "waiting for acquisition start trigger" acquisition status.

Because the frame start trigger is set to on, the user must apply a frame start trigger signal to the camera in order to begin each frame acquisition. In this case, we have set the frame start trigger signal source to input line 1 and the activation to rising edge, so the rising edge of an externally generated electrical signal applied to input line 1 will serve as the frame start trigger signal. Keep in mind that the camera will only react to a frame start trigger signal when it is in a "waiting for frame start trigger" acquisition status.

A possible use for this type of setup is a conveyor system that moves objects past an inspection camera. Assume that the system operators want to acquire images of 3 specific areas on each object, that the conveyor speed varies, and that they do not want to acquire images when there is no object in front of the camera. A sensing device on the conveyor could be used in conjunction with a PC to determine when an object is starting to pass the camera. When an object is starting to pass, the PC will execute an acquisition start trigger software command, causing the camera to exit the "waiting for acquisition start trigger" acquisition status and enter a "waiting for frame start trigger" acquisition status.

An electrical device attached to the conveyor could be used to generate frame start trigger signals and to apply them to input line 1 on the camera. Assuming that this electrical device was based on a position encoder, it could account for the speed changes in the conveyor and ensure that frame trigger signals are generated and applied when specific areas of the object are in front of the camera. Once 3 frame start trigger signals have been received by the camera, the number of frames acquired would be equal to the setting for the Acquisition Frame Count parameter, and the camera would return to the "waiting for acquisition start trigger" acquisition status. Any frame start trigger signals generated at that point would be ignored.

This sort of setup is useful because it will only acquire frames when there is an object in front of the camera and it will ensure that the desired areas on the object are imaged. (Transmitting images of the "space" between the objects would be a waste of bandwidth and processing them would be a waste of processor resources.)

Use Case:	Acquisition Start Trigger On and Frame Start Trigger On
	The acquisition start trigger is on, and the acquisition start trigger source is set to software. The user must execute an acquisition start trigger software command to make the camera exit the "waiting for acquisition start trigger" acquisition status. Because the acquisition frame count is set to 3, the camera will re-enter the "waiting for acquisition start trigger" acquisition status after 3 frame trigger signals have been applied.
	The frame start trigger is on, and the frame start trigger source is set to input line 1. The user must apply a frame start trigger signal to input line 1 to start each frame exposure.
Settings:	Acquisition Mode = Continuous Trigger Mode for the acquisition start trigger = On Trigger Source for the acquisition start trigger = Software Acquisition Frame Count = 3 Trigger Mode for the frame start trigger = On Trigger Source for the frame start trigger = Line 1 Trigger Activation for the frame start trigger = Rising Edge

──────── = a trigger signal applied by the user

▨▨▨▨ = camera is waiting for an acquisition start trigger signal

▧▧▧ = camera is waiting for a frame start trigger signal

▮ = frame exposure and readout

▼ = frame transmission

⊓ = a frame start trigger signal that will be ignored because the camera is not in a "waiting for frame start trigger" status
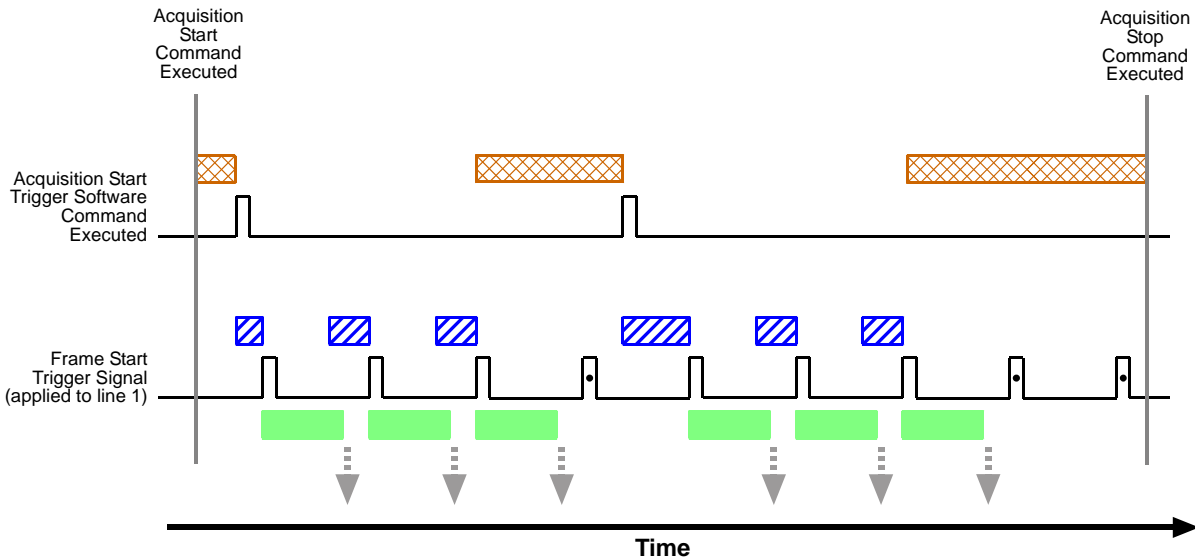

Fig. 58: Use Case 4 - Acquisition Start Trigger On and Frame Start Trigger On

9.9   Overlapping Exposure and Sensor Readout (All Models Except scA750-60)

> (i)   The information in this section applies to all camera models except the
>   scA750-60 gm/gc. For information about scA750-60 gm/g cameras, see
>   Section 9.10 on .

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 59.
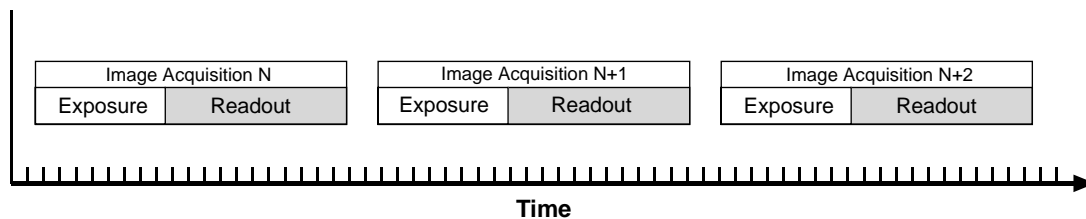
Fig. 59: Non-overlapped Exposure

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 60 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.
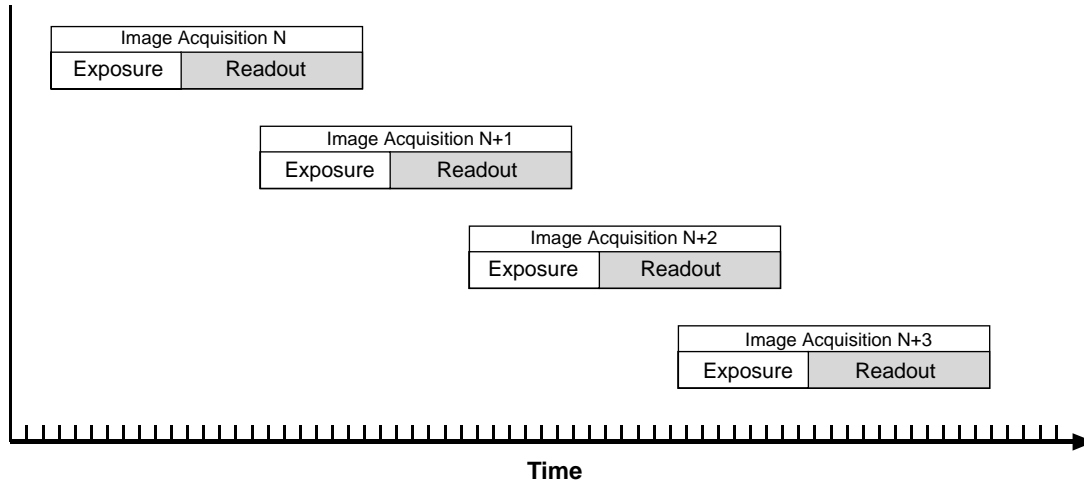
Fig. 60: Overlapped Exposure

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 .

9.9.1   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 9.11.3 on .

For more detailed guidelines about using an external trigger signal with the trigger width exposure mode and overlapped exposure, refer to the application notes called *Using a Specific External Trigger Signal with Overlapped Exposure* (AW000565). The application notes are available in the downloads section of the Basler website: www.baslerweb.com.

9.10 Exposure Must Not Overlap Sensor Readout (scA750-60 Only)

(i)	The information in this section only applies to scA750-60 gm/gc cameras. For information about the other camera models, see Section 9.9 on page 149.

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 59.
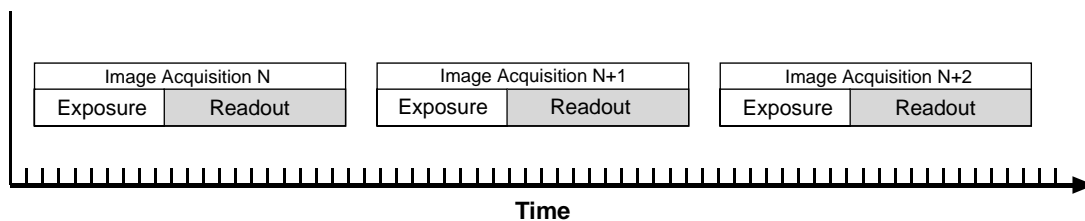
Fig. 61: Non-overlapped Readout and Exposure

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 9.11.4 on page 155

9.11   Acquisition Monitoring Tools

9.11.1 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 62. 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.
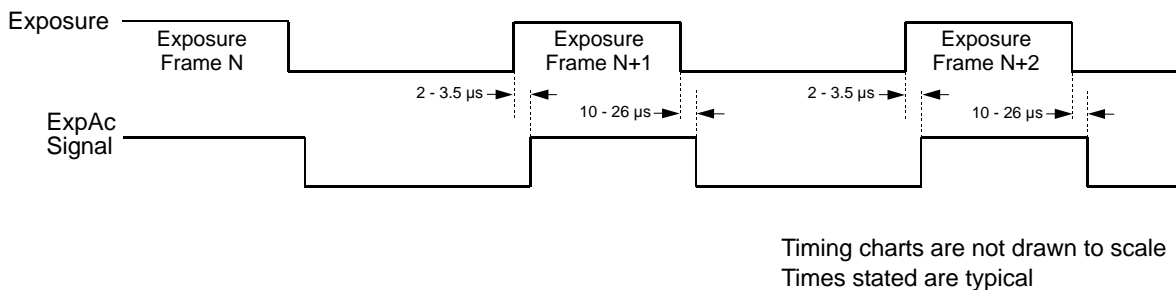
Fig. 62: Exposure Active Signal

> (i) 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 62 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.

Selecting the Exposure Active Signal as the Source Signal for an Output Line

The exposure active output signal can be selected to act as the source signal for e.g.output line 1. Selecting a source signal for the output line is a two step process:

- Use the Line Selector to select output line 1.
- Set the value of the Line Source Parameter to the exposure active output signal.

You can set the Line Selector and the Line Source parameter value from within your application software by using the Basler 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 );
```

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

For more information about changing the assignment of camera output signals to physical output lines, see Section 8.2.1 on .

For more information about the electrical characteristics of the camera's output lines, see Section 7.6.2 on .

9.11.2 Acquisition Status Indicator

When controlling image acquisition with a software trigger you can use the acquisition status feature to determine when the camera is ready to be triggered for an image acquisition.

For more information about the acquisition status feature, see Section 12.14 on .

9.11.3 Trigger Ready Signal (All Models Except scA750-60)

(i)	The information in this section applies to all camera models except the scA750-60 gm/gc. For information about scA750-60 gm/gc cameras, see Section 9.11.4 on page 155.

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. The trigger ready signal relates to different trigger signals, depending on the image acquisition control mode:

- In standard mode, the trigger ready signal relates to the frame start trigger signal
- In legacy mode, the trigger ready signal relates to the acquisition start trigger signal.

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 63). 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. 63: 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.

> (i) 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.

Selecting the Trigger Ready Signal as the Source Signal for an Output Line

The trigger ready signal can be selected to act as the source signal for e.g.output line 1. Selecting a source signal for the output line is a two step process:

▪   Use the Line Selector to select output line 1.

▪   Set the value of the Line Source Parameter to the trigger ready output signal.

You can set the Line Selector and the Line Source parameter value from within your application software by using the Basler 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_TriggerReady );
```

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

For more information about changing the assignment of camera output signals to physical output lines, see Section 8.2.1 on .

For more information about the electrical characteristics of the camera's output lines, see Section 7.6.2 on .

For more information about the standard and legacy image acquisition control modes, see Section 9.1 on .

9.11.4 Trigger Ready Signal (scA750-60 Only)

> ⓘ   The information in this section only applies to scA750-60 gm/gc cameras. For information about the other camera models, see Section 9.11.3 on .

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. The trigger ready signal relates to different trigger signals, depending on the image acquisition control mode:

▪   In standard mode, the trigger ready signal relates to the frame start tigger signal.

▪   In legacy mode, the trigger ready signal relates to the acquisition start tigger signal.

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 63). 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. 64: 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.

> 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 8.2.1 on .

For more information about the electrical characteristics of the camera's output lines, see Section 7.6.2 on .

Selecting the Trigger Ready Signal as the Source Signal for an Output Line

The trigger ready signal can be selected to act as the source signal for e.g. output line 1. Selecting a source signal for the output line is a two step process:

▪ Use the Line Selector to select output line 1.

▪ Set the value of the Line Source Parameter to the trigger ready output signal.

You can set the Line Selector and the Line Source parameter value from within your application software by using the Basler 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_TriggerReady );
```

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

For more information about changing the assignment of camera output signals to physical output lines, see Section 8.2.1 on page 89.

For more information about the electrical characteristics of the camera's output lines, see Section 7.6.2 on page 83.

For more information about the standard and legacy image acquisition control modes, see Section 9.1 on page 99.

9.11.5 Acquisition Trigger Wait Signal (Standard Mode Only)

> (i) The acquisition trigger wait signal is only available when the image acquisition control is set to standard mode.
>
> For more information about image acquisition control and setting the mode, see Section 9.1 on page 99.

As you are acquiring frames, the camera automatically monitors the acquisition start trigger status and supplies a signal that indicates the current status. The Acquisition Trigger Wait signal will go high whenever the camera enters a "waiting for acquisition start trigger" status. The signal will go low when an external acquisition start trigger (ExASTrig) signal is applied to the camera and the camera exits the "waiting for acquisition start trigger status". The signal will go high again when the camera again enters a "waiting for acquisition trigger" status and it is safe to apply the next acquisition start trigger signal.

If you base your use of the ExASTrig signal on the state of the acquisition trigger wait signal, you can avoid "acquisition start overtriggering", i.e., applying an acquisition start trigger signal to the camera when it is not in a "waiting for acquisition start trigger" acquisition status. If you do apply an acquisition start trigger signal to the camera when it is not ready to receive the signal, it will be ignored and an acquisition start overtrigger event will be reported.

Figure 65 illustrates the Acquisition Trigger Wait signal with the Acquisition Frame Count parameter set to 3 and with exposure and readout overlapped. The figure assumes that the trigger mode for the frame start trigger is set to off, so the camera is internally generating frame start trigger signals.
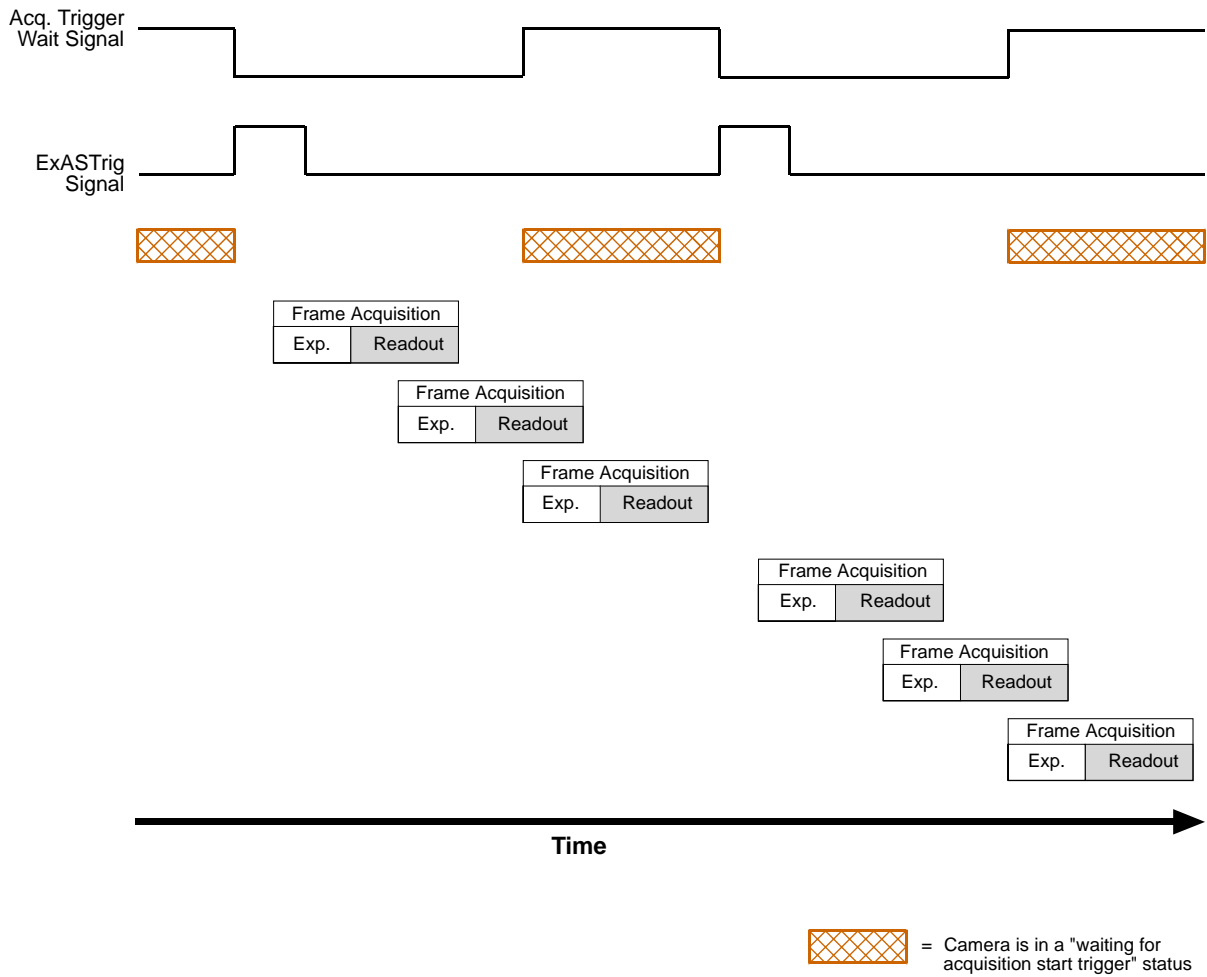
Fig. 65: Acquisition Trigger Wait Signal

> ⓘ The acquisition trigger wait signal will only be available when hardware
> acquisition start triggering is enabled.

Selecting the Acquisition Trigger Wait Signal as the Source Signal for an Output Line

The acquisition trigger wait signal can be selected to act as the source signal for camera output line 1, line 2, line 3, or line 4. Selecting a source signal for an output line is a two step process:

- Use the Line Selector to select output line 1, line 2, line 3, or line 4.
- Set the value of the Line Source Parameter to the acquisition trigger wait signal.

You can set the Line Selector and the Line Source parameter value from within your application software by using the Basler 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_AcquisitionTriggerWait );
```

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

9.11.6 Camera Events

Certain camera events allow you to get informed about the current camera acquisition status:

- AcquisitionStartEventData event: An acquisition start trigger has occurred.
- FrameStartEventData event: A frame start trigger has occurred.
- ExposureEndEventData event: The end of an exposure has occurred.

For more information about camera events and event reporting, see Section 12.15 on .

9.12   Acquisition Timing Chart

Figure 66 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.

The ExTrig signal will be an externally triggered frame start trigger signal when using image acquisition control in standard mode. The ExTrig signal will be an externally triggered acquisition start trigger signal when using image acquisition control in legacy mode.

As Figure 66 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	Camera Model	Exposure Start Delay
scA640-70 gm/gc	33.80 µs	scA1300-32 gm/gc	34.14 µs
scA640-74 gm/gc	30.05 µs	scA1390-17 gm/gc	65.31 µs
scA640-120 gm/gc	19.34 µs	scA1400-17 gm/gc	63.17 µs
scA750-60 gm/gc	180.0 µs	scA1400-30 gm/gc	36.94 µs
scA780-54 gm/gc	35.94 µs	scA1600-14 gm/gc	63.53 µs
scA1000-30 gm/gc	47.23 µs	scA1600-28 gm/gc	32 µs

Table 12: Exposure Start Delays

Note that, if the debouncer feature is used, the debouncer setting for the input line must be added to the exposure start delays shown in Table 12 to determine the total start delay. For example, assume that you are using an scA640-70 camera and that you have set the cameras for hardware triggering. Also assume that you have selected input line 1 to accept the hardware trigger signal and that you have set the Line Debouncer Time Abs parameter for input line 1 to 5 µs.

In this case:

Total Start Delay = Start Delay from Table 12 + Debouncer Setting

Total Start Delay = 33.80 µs+ 5 µs

Total Start Delay = 38.80 µs
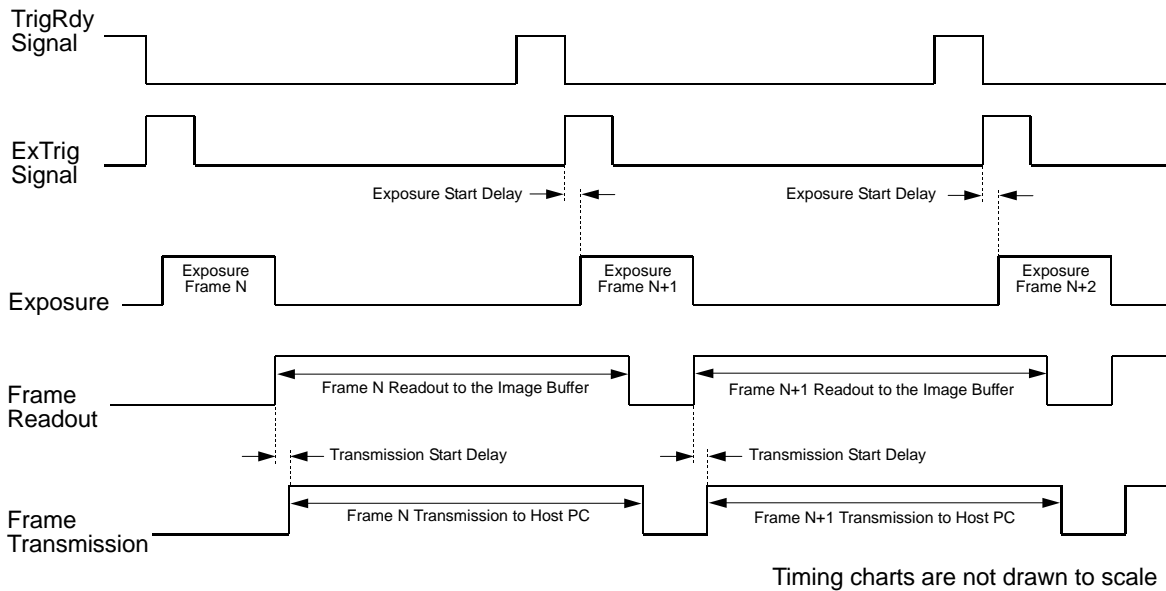
Fig. 66: Exposure Start Controlled with an ExTrig Signal

You can calculate the frame readout time by using this formula:

Frame Readout Time = ((AOI Height + 1) x C_1) + C_2

Where the values for the constants C_1 and C_2 are from the table in Section 9.13 on page 163 for all camera models except the scA750-60 or from the table in Section 9.14 on page 167 for scA750-60 cameras.

For more information about the AOI height, see Section 12.4 on page 234.

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 transmission 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 5.1 on page 57.

9.13   Maximum Allowed Acquisition Frame Rate (All Models Except scA750-60)

> (i) The information in this section applies to all camera models except the scA750-60 gm/gc. For information about scA750-60 gm/gc cameras, see Section 9.14 on page 167.

In general, the maximum allowed acquisition frame rate can be limited by three factors:

- The amount of time it takes to read an acquired frame out of the imaging sensor and into the camera's frame buffer. This time varies depending on the height of the frame. Frames with a smaller height 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.

> (i) When the camera's acquisition mode is set to single frame, the maximum possible acquisition frame rate for a given AOI cannot be achieved. To achieve the maximum possible acquisition frame rate, set the camera for the continuous acquisition mode.

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 the acquisition mode, see Section 9.2 on page 101 and Section 9.3 on page 105.

For more information about AOI Height settings, see Section 12.4 on .

For more information about the Resulting Frame Rate parameter, see Section 5.1 on .

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 which 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 12.4 on .

For more information on the settings that determine the bandwidth assigned to the camera, see Section 5.2 on .

Formula 1:

Calculates the maximum frame rate based on the sensor readout time:

$$\text{Max. Frames/s} = \frac{1}{[(\text{AOI Height} + 1) \times C_1] + C_2}$$

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:

Camera Model	C_1	C_2	Camera Model	C_1	C_2
scA640-70gm/gc	25.49 µs	1612.86 µs	scA1390-17gm/gc	46.53 µs	10244.54 µs
scA640-74gm/gc	22.66 µs	1433.66 µs	scA1400-17gm/gc	46.53 µs	9357.50 µs
scA640-120gm/gc	14.16 µs	1128.80 µs	scA1400-30gm/gc	25.80 µs	6418.04 µs
scA780-54gm/gc	26.94 µs	2308.54 µs	scA1600-14gm/gc	50.99 µs	6658.82 µs
scA1000-30gm/gc	35.39 µs	4507.20 µs	scA1600-28gm/gc	26.62 µs	2870.54 µs
scA1300-32gm/gc	28.65 µs	2806.10 µs			

Formula 2:

Calculates the maximum frame rate based on the exposure time for the acquired 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:

Camera Model	C_3	Camera Model	C_3
scA640-70gm/gc	95 µs	scA1390-17gm/gc	177 µs
scA640-74gm/gc	84 µs	scA1400-17gm/gc	175 µs
scA640-120gm/gc	24 µs	scA1400-30gm/gc	97 µs
scA780-54gm/gc	100 µs	scA1600-14gm/gc	176 µs
scA1000-30gm/gc	132 µs	scA1600-28gm/gc	95.23 µs
scA1300-32gm/gc	54 µs		

For more information about setting the exposure time, see Section 9.7 on .

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:

$$\text{Max Frames/s} = \frac{1}{[(400 + 1) \times 25.49] + 1612.86}$$

Max Frames/s = 84.4 frames/s

Formula 2:

$$\text{Max Frames/s} = \frac{1}{2000 \text{ µs} + 95 \text{ µs}}$$

Max Frames/s = 477.3 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 84.4 frames per second.

9.14   Maximum Allowed Acquisition Frame Rate (scA750-60 Only)

> (i)   The information in this section only applies to scA750-60 gm/gc cameras. For information about the other camera models, see Section 9.13 on page 163.

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.

> (i)   When the camera's acquisition mode is set to single frame, the maximum possible acquisition frame rate for a given AOI cannot be achieved. To achieve the maximum possible acquisition frame rate, set the camera for the continuous acquisition mode.

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 12.4 on page 234.

For more information about the Resulting Frame Rate parameter, see Section 5.1 on page 57.

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 12.4 on page 234.

For more information on the settings that determine the bandwidth assigned to the camera, see Section 5.2 on page 64.

Formula 1:

Calculates the maximum frame rate based on the sum of the exposure time plus the sensor readout time:

$$\text{Max. Frames/s} = \frac{1}{\text{Exposure Time in µs} + ((\text{AOI Height} + 1) \times C_1) + C_2}$$

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	30.99 µs
C_2	185.93 µs

For more information about setting the exposure time, see Section 9.7 on page 137.

Formula 2:

Calculates the maximum frame rate based on the frame transmission time:

$$\text{Max. Frames/s} = \frac{\text{Device Current Throughput Parameter Value}}{\text{Payload Size Parameter Value}}$$

Example

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:

$$\text{Max. Frames/s} = \frac{1}{2000\ \mu s + ((400 + 1) \times 30.99\ \mu s) + 185.93\ \mu s}$$

Max Frames/s = 68.4 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 68.4 frames per second.

10 Color Creation and Enhancement

This chapter provides information about how color images are created on color camera models and about the features available for adjusting the appearance of the colors.

10.1   Color Creation

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 67 (the alignment of the Bayer filter with respect to the sensor is shown as an example only; the figure shows the "BG" filter alignment). 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.)
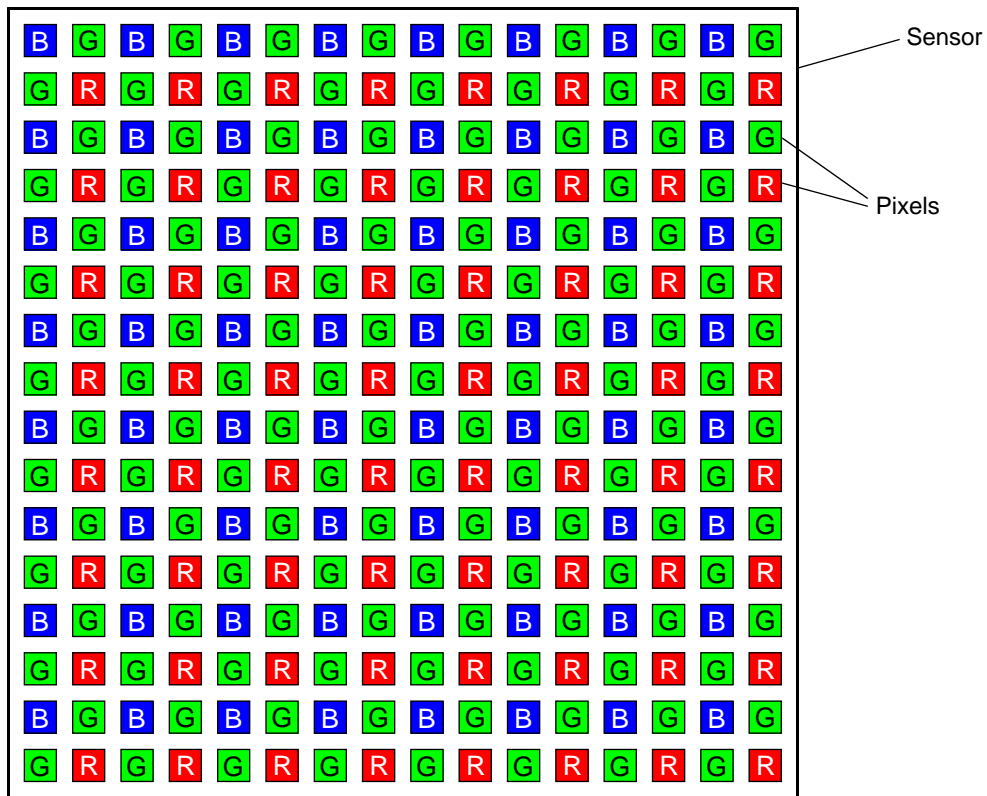
Fig. 67: Bayer Filter Pattern

10.1.1 Bayer 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 13 shows the filter alignment for each available camera model.

Color Camera Model	Filter Alignment
scA640-70	BG
scA640-74	BG
scA640-120	BG
scA750-60	RG
scA780-54	BG
scA1000-30	BG
scA1300-32	BG
scA1390-17	BG
scA1400-17	BG
scA1400-30	BG
scA1600-14	BG
scA1600-28	BG

Table 13: 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 12.4 on page 234.

10.1.2 Pixel Data Formats Available on Cameras with a Bayer Filter

Bayer Formats

Cameras equipped with a Bayer pattern color filter can output pixel data in the Bayer RG 8 or Bayer BG 8 pixel data output format, the Bayer BG 16 format, and the Bayer BG 12 Packed format. When a color camera is set for one of these pixel data output formats, the pixel data is not processed or interpolated in any way. For each pixel covered with a red portion of the filter, you get 8 or 12 bits of red data. For each pixel covered with a green portion of the filter, you get 8 or 12 bits of green data. And for each pixel covered with a blue portion of the filter, you get 8 or 12 bits of blue data. (This type of pixel data is sometimes referred to as "raw" output.)

For complete details of these pixel data output formats, see Section 11.1 on and Section 11.3 on .

RGB Format

All color cameras with a Bayer filter can output pixel data in RGB 8 Packed format.

When a color camera is set for this format, a demosaicing algorithm is performed to get RGB data for each pixel. This is required because color cameras with a Bayer filter on the sensor gather only one color of light for each individual pixel.

Once the demosaicing is complete, 8 bits of red, green, and blue data is transmitted for each pixel to the host PC.

For complete details of this pixel data output format, see Section 11.3.5 on .

YUV Formats

All color cameras with a Bayer filter can output pixel data in YUV 4:2:2 Packed format or in YUV 4:2:2 (YUYV) Packed format.

When a color camera is set for either of these formats, 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, a demosaicing algorithm is performed to get RGB data for each pixel. This is required because color cameras with a Bayer filter on the sensor gather only one color of light for each individual pixel.

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.

For complete details of the YUV data output formats, see Section 11.3 on .

Mono Format

Cameras equipped with a Bayer pattern color filter can output pixel data in the Mono 8 format.

When a color camera is set for Mono 8, the pixel values in each captured image are first demosaiced and converted to the YUV color model as described above. 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".)

For complete details of the Mono 8 format, see Section 11.3 on .

10.2   Integrated IR Cut Filter
(All 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.

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 32.

10.3   Color Enhancement Features

10.3.1 White Balance

The white balance feature lets you adjust the balance of red, green, and blue such that white objects in the camera's field of view appear white in the acquired images.

Setting the White Balance

(i)	This section (Section 10.3.1) describes how a color camera's white balance can be adjusted "manually", i.e., by setting the value of the Balance Ratio Abs parameters for red, green, and blue. The camera also has a White Balance Auto function that can automatically adjust the white balance. **Manual adjustment of the Balance Ratio Abs parameters for red, green, and blue will only work if the Balance White Auto function is disabled.** For more information about auto functions in general, see Section 12.9 on page 283. For more information about the Balance White Auto function, see Section 12.9.5 on page 295. When you are using matrix color transformation and you set the Light Source Selector parameter to match your light source characteristics, the camera will automatically make adjustments to the white balance settings so that they are best suited for the light source of the type you selected. For more information about matrix color transformation, see Section 10.3.3 on page 181.

With the white balancing scheme used on these cameras, the red intensity, green intensity, and blue intensity can be individually 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 be 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 0.00 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 the color can achieve. For this reason, we don't normally recommend setting a balance ratio less than 1 unless you want to correct for the strong predominance of one color.

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 Basler 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 API and the pylon Viewer, see Section 3 on .

White Balance Reset

The camera includes a White Balance Reset command that can be used to reset the white balance adjustments. This feature is especially useful if you have badly misadjusted the white balance and you want to quickly return to reasonable settings. When the reset command is used, it will return the camera to the settings defined by your current Light Source Selector parameter setting.

You can execute the White Balance Reset command from within your application software by using the pylon API. The following code snippet illustrates using the API to execute the command:

```
// Reset the white balance adjustments
Camera.BalanceWhiteReset.Execute( );
```

You can also use the Basler pylon Viewer application to easily execute the command.

For more information about the pylon API and the pylon Viewer, see Section 3 on .

For more information about the Light Source Selector parameter setting, see Section 10.3.3 on .

10.3.2 Gamma Correction

The gamma correction feature lets you modify the brightness of the pixel values output by the camera's sensor to account for a non-linearity in the human perception of brightness. There are two modes of gamma correction available on the camera: sRGB and User.

sRGB Gamma

When the camera is set for sRGB gamma correction, it automatically sets the gamma correction to adjust the pixel values so that they are suitable for display on an sRGB monitor. If you will be displaying the images on an sRGB monitor, using this type of gamma correction is appropriate.

User Gamma

With User type gamma correction, you can set the gamma correction value as desired.

To accomplish the correction, a gamma correction value (γ) is applied to the brightness value (Y) of each pixel according to the following formula:

$$Y_{corrected} = \left(\frac{Y_{uncorrected}}{Y_{max}} \right)^{\gamma} \times Y_{max}$$

The formula uses uncorrected and corrected pixel brightnesses that are normalized by the maximum pixel brightness. The maximum pixel brightness equals 255 for 8 bit output and 4095 for 12 bit output.

The gamma correction value can be set in a range from 0 to 3.99902.

When the gamma correction value is set to 1, the output pixel brightness will not be corrected.

A gamma correction value between 0 and 1 will result in increased overall brightness, and a gamma correction value greater than 1 will result in decreased overall brightness.

In all cases, black (output pixel brightness equals 0) and white (output pixel brightness equals 255 at 8 bit output and 4095 at 12 bit output) will not be corrected.

Enabling and Setting Gamma Correction

You can enable or disable the gamma correction feature by setting the value of the Gamma Enable parameter.

You can use the Gamma Selector to select either sRGB or user gamma correction.

If you select user gamma correction, you can use the Gamma parameter to set the gamma correction value.

You can set the Gamma Enable parameter, use the Gamma Selector, and set Gamma parameter values from within your application software by using the Basler pylon API. The following code snippet illustrates using the API to set the parameter values for sRGB type correction:

```
// Enable the Gamma feature
Camera.GammaEnable.SetValue( true );
// Set the gamma type to sRGB
Camera.GammaSelector.SetValue ( GammaSelector_sRGB );
```

The following code snippet illustrates using the API to set the parameter values for user type correction:

```
// Enable the Gamma feature
Camera.GammaEnable.SetValue( true );
// Set the gamma type to User
Camera.GammaSelector.SetValue ( GammaSelector_User );
// Set the Gamma value to 1.2
Camera.Gamma.SetValue( 1.2 );
```

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

For more information about the pylon API and the pylon Viewer, see Section 3 on .

10.3.3 Matrix Color Transformation on All Color Models

> (i) If matrix color transformation is not available, activate the "Full" camera description file.
>
> For more information, see Section 12.19 on page 321.

Introduction

The main objective of matrix color transformation is to make corrections to the color information that will account for the type of lighting used during image acquisition and to compensate for imperfections in the sensor's color generation process.

With the matrix color transformation, a first matrix transformation step ensures that the pixel values from the sensor are available in RGB color space, i.e. as R, G, or B component for each pixel. A second transformation step takes account of the specific pre-selected light source. The vector consisting of the R, G, or B component for each pixel in the image is multiplied by a matrix containing a set of correction values.

Matrix Color Transformation Parameters

The initial parameter that you must consider when working with the matrix color transformation feature is the Processed Raw Enable parameter. If the camera is set to output pixel data in one of the Bayer formats, then the Processed Raw Enable parameter must be set to "enabled" to allow color enhancements to be performed. Setting this parameter to enabled will allow the camera to perform color enhancements on the raw RGB data from the sensor and still be able to output the pixel data in one of the Bayer formats. If the camera is set for one of the Bayer pixel data output formats and the Processed Raw Enable parameter is not set to enabled, the matrix color transformation feature and the color adjustment feature will have no effect on camera operation.

The Processed Raw Enable parameter is not relevant if the camera is set to a color pixel data output format other than a Bayer format, e.g. to a YUV output format.

The first parameter associated with the matrix color transformation feature is the Color Transformation Selector parameter. This parameter is used to select the type of transformation that will be performed before color correction for a specific light source is performed (addressed by the second parameter). For cameras equipped with a Bayer pattern filter on the imaging sensor, RGB to RGB is the only setting available. This setting means that the matrix color transformation process will not transform the red, green, and blue pixel values from the sensor into a different color space.

The second parameter associated with matrix color transformation is the Light Source Selector parameter. The following settings are available for this parameter:

▪ Off - No alterations will be made to the pixel values.

▪ Tungsten - This setting will automatically populate the matrix with a pre-selected set of values that will make appropriate corrections for images captured with tungsten lighting that has a color temperature of about 2500K to 3000K. When you select this setting, the camera will also

adjust the white balance settings and the color adjustment settings so that they are appropriate for a tungsten light source.

- Daylight - This setting will automatically populate the matrix with a pre-selected set of values that will make appropriate corrections for images captured with daylight lighting that has a color temperature of about 5000K. When you select this setting, the camera will also adjust the white balance settings and the color adjustment settings so that they are appropriate for a daylight light source with a color temperature of about 5000K.

- Daylight 6500K - This setting will automatically populate the matrix with a pre-selected set of values that will make appropriate corrections for images captured with daylight lighting that has a color temperature of about 6500K. When you select this setting, the camera will also adjust the white balance settings and the color adjustment settings so that they are appropriate for a daylight light source with a color temperature of about 6500K.

- Custom - The user can set the values in the matrix as desired. When you select this setting, the camera will also adjust the white balance settings and the color adjustment settings so that they have neutral values that do not change the appearance of the colors.

In almost all cases, selecting one of the settings that populate the matrix with pre-selected values will give you excellent results with regard to correcting the colors for the light source you are using.

The custom setting should only be used by someone who is thoroughly familiar with matrix color transformations. Instructions for using the custom setting appear in the next section.

The third parameter associated with matrix color transformation is the Color Transformation Matrix Factor parameter. This parameter determines how strong an effect the matrix correction function will have on the colors output by the camera. The parameter setting is a floating point value that can range from 0 to 1. When the parameter value is set to 0, matrix correction will have no effect. When the value is set to 1, matrix correction will have its maximum effect.

As an alternative, the Color Transformation Matrix Factor parameter value can be entered as an integer value on a scale ranging from 0 to 65536. This integer range maps linearly to the floating point range with 0 being equivalent to 0 and 65536 being equivalent to 1. The integer values can be entered using the Color transformation Matrix Factor Raw parameter.

(i)	When the Light Source Selector parameter is set to off or custom, the Color Transformation Matrix Factor parameter will not be available.

Setting Matrix Color Transformation

You can set the Processed Raw Enable, Color Transformation Selector and Light Source Selector parameter values from within your application software by using the Basler pylon API. In this example, we assume that you want to set your camera for Bayer BG 8 output, and therefore you must set the Processed Raw Enable parameter value to enabled.

The following code snippet illustrates using the API to set the parameter values:

```
// Set the camera for Bayer BG8 pixel data output format
Camera.PixelFormat.SetValue( PixelFormat_BayerBG8 );
// Because the camera is set for a Bayer output format, the Processed Raw
// Enabled parameter must be set to enabled
```

```
Camera.ProcessedRawEnable.SetValue( true );


// Select the matrix color transformation type
Camera.ColorTransformationSelector.SetValue
     ( ColorTransformationSelector_RGBtoRGB );


// Set the light source selector so that no correction will be done
Camera.LightSourceSelector.SetValue
     ( LightSourceSelector_Off );
// Set the light source selector for tungsten lighting
Camera.LightSourceSelector.SetValue
     ( LightSourceSelector_Tungsten );
// Set the light source selector for daylight (at about 5000K)
Camera.LightSourceSelector.SetValue
     ( LightSourceSelector_Daylight );
// Set the light source selector for daylight (at about 6500K)
Camera.LightSourceSelector.SetValue
     ( LightSourceSelector_Daylight6500K );


// Set the matrix correction factor
Camera.ColorTransformationMatrixFactor.SetValue( 0.50 );
```

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

For more information about the pylon API and the pylon Viewer, see Section 3 on .

10.3.3.1 The Custom Light Source Setting

ⓘ	The "Custom" setting for the Light Source Selector parameter is intended for use by someone who is thoroughly familiar with matrix color transformations. **It is nearly impossible to enter correct values in the conversion matrix by trial and error.**

The RGB to RGB color matrix conversion for each pixel is performed by multiplying a 1 x 3 matrix containing R, G, and B color values with a 3 x 3 matrix containing correction values. Each column in the 3 x 3 matrix can be populated with values of your choice. In other words:

$$\begin{bmatrix} Gain00 & Gain01 & Gain02 \\ Gain10 & Gain11 & Gain12 \\ Gain20 & Gain21 & Gain22 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

Where Gain00, Gain01, etc. are settable values.

Each GainXY position can be populated with a floating point value ranging from -8.0 to +7.96875 by using the Color Transformation Value Selector to select one of the GainXY positions in the matrix and using the Color transformation Value parameter to enter a value for that position.

As an alternative the Gain XY values can each be entered as an integer value on a scale ranging from -256 to +255. This integer range maps linearly to the floating point range with -256 being equivalent to -8.0, 32 being equivalent to 1.0, and +255 being equivalent to +7.96875. The integer values can be entered using the Color transformation Value Raw parameter.

A reference article that explains the basics of color matrix transformation for video data can be found at:
http://www.its.bldrdoc.gov/publications/04-406.aspx

Setting Custom Matrix Values

You can set the Color Transformation Value Selector, Color Transformation Value, and Color Transformation Value Raw parameters from within your application software by using the Basler pylon API. The following code snippet illustrates using the API to set the values in the matrix. Note that the values in this example are just randomly selected numbers and do not represent values that you should actually use.

```
// Set the light source selector for custom
Camera.LightSourceSelector.SetValue ( LightSourceSelector_Custom );

// Select a position in the matrix
Camera.ColorTransformationValueSelector.SetValue
    ( ColorTransformationValueSelector_Gain01 );
// Set the value for the selected position as a floating point value
Camera.ColorTransformationValue.SetValue( 2.11 );

// Select a position in the matrix
Camera.ColorTransformationValueSelector.SetValue
    ( ColorTransformationValueSelector_Gain12 );
// Set the value for the selected position as an integer value
Camera.ColorTransformationValueRaw.SetValue( 135 );
```

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

10.3.4 Color Adjustment

	If color adjustment is not available, activate the "Full" camera description file. For more information, see Section 12.19 on .

The camera's color adjustment feature lets you adjust hue and saturation for the primary and secondary colors in the RGB color space. Each adjustment affects those colors in the image where the adjusted primary or secondary color predominates. For example, the adjustment of red affects the colors in the image with a predominant red component.

	For the color adjustments to work properly, the white balance must be correct. See Section 10.3.1 on for more information about the white balance and see Section 10.3.5 on for a overall procedure for setting the color enhancement features.

	Although color adjustment can be used without also using color matrix transformation, we nonetheless strongly recommend to also use color matrix transformation to make full use of the camera's color enhancement capabilities. See Section 10.3.3 on for more information about color matrix transformation.

The RGB Color Space

The RGB color space includes light with the primary colors red, green, and blue and all of their combinations. When red, green, and blue light are combined and when the intensities of R, G, and B are allowed to vary independently between 0% and 100%, all colors within the RGB color space can be formed. Combining colored light is referred to as additive mixing.

When two primary colors are mixed at equal intensities, the secondary colors will result. The mixing of red and green light produces yellow light (Y), the mixing of green and blue light produces cyan light (C), and the mixing of blue and red light produces magenta light (M).

When the three primary colors are mixed at maximum intensities, white will result. In the absence of light, black will result.

The color space can be represented as a color cube (see Figure 68 on) where the primary colors R, G, B, the secondary colors C, M, Y, and black and white define the corners. All shades of gray are represented by the line connecting the black and the white corner.

For ease of imagination, the color cube can be projected onto a plane (as shown in Figure 68) such that a color hexagon is formed. The primary and secondary colors define the corners of the color

hexagon in an alternating fashion. The edges of the color hexagon represent the colors resulting from mixing the primary and secondary colors. The center of the color hexagon represents all shades of gray including black and white.

The representation of any arbitrary color of the RGB color space will lie within the color hexagon. The color will be characterized by its hue and saturation:

- Hue specifies the kind of coloration, for example, whether the color is red, yellow, orange etc.
- Saturation expresses the colorfulness of a color. At maximum saturation, no shade of gray is present. At minimum saturation, no "color" but only some shade of gray (including black and white) is present.
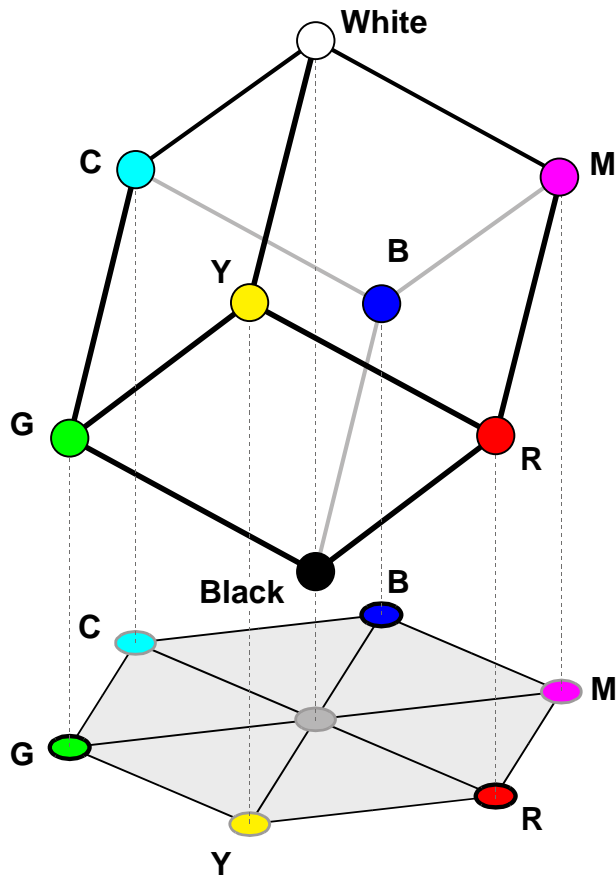
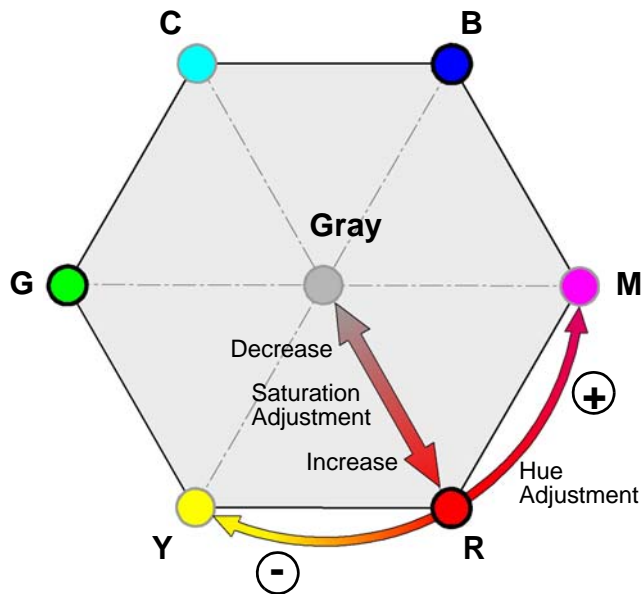Fig. 68: RGB Color Cube With YCM Secondary Colors, Black, and White, Projected On a Plane

Fig. 69: Hue and Saturation Adjustment In the Color Hexagon. Adjustments Are Indicated for Red as an Example

Hue and Saturation Adjustment

The color adjustment feature lets you adjust hue and saturation for the primary and the secondary colors. Each adjustment affects those areas in the image where the adjusted color predominates. For example, the adjustment of red affects the colors in the image with a predominantly red component.

Keep in mind that when you adjust a color, the colors on each side of it in the color hexagon will also be affected to some degree. For example, when you adjust red, yellow and magenta will also be affected.

- In the color hexagon, the adjustment of hue can be considered as a rotation between hues. Primary colors can be rotated towards, and as far as, their neighboring secondary colors. And secondary colors can be rotated towards, and as far as, their neighboring primary colors.

  For example, when red is rotated in negative direction towards yellow, then, for example, purple in the image can be changed to red and red in the image can be changed to orange.
  Red can be rotated as far as yellow, where red will be completely transformed into yellow.

  When red is rotated in a positive direction towards magenta, then, for example, orange in the image can be changed to red and red in the image can be changed to purple.
  Red can be rotated as far as magenta, where red will be completely transformed into magenta.

- Adjusting saturation changes the colorfulness (intensity) of a color. The color adjustment feature lets you adjust saturation for the primary and secondary colors.

  For example, if saturation for red is increased, the colorfulness of red colors in the image will increase. If red is set to minimum saturation, red will be replaced by gray for "red" colors in the image.

Color Adjustment Parameters

The initial parameter that you must consider when working with the color adjustment feature is the Processed Raw Enable parameter. If you are working with a camera that is set to output pixel data in one of the Bayer formats, then the Processed Raw Enabled parameter must be set to "enabled" if you want to use color enhancement. The camera will then be able to perform color enhancements on the raw RGB data from the sensor and still be able to output the pixel data in one of the Bayer formats. If the camera is set for one of the Bayer formats pixel data output formats and the Processed Raw Enable parameter is not set to enabled, the matrix color transformation feature and the color adjustment feature will have no effect on the camera operation.

The Processed Raw Enable parameter is not relevant if the camera is set to a color pixel data output format other than a Bayer format, e.g. to a YUV output format.

You can enable or disable the color adjustment feature by setting the value of the Color Adjustment Enable parameter to true or false.

You can use the Color Adjustment Selector parameter to select a color to adjust. The colors you can select are: red, yellow, green, cyan, blue, and magenta.

You can use the Color Adjustment Hue parameter to set the hue for the selected color as a floating point value in a range from -4.0 to +3.96875.

As an alternative, you can use the Color Adjustment Hue Raw parameter to set the hue as an integer value on a scale ranging from -128 to +127. This integer range maps linearly to the floating point range with -256 being equivalent to -4.0, +32 being equivalent to +1.0, and +255 being equivalent to +3.96875.

You can use the Color Adjustment Saturation parameter to set the saturation for the selected color as a floating point value in a range from 0.0 to +1.99219.

As an alternative, you can use the Color Adjustment Saturation Raw parameter to set the saturation as an integer value on a scale ranging from 0 to 255. This integer range maps linearly to the floating point range with 0 being equivalent to 0.0, +128 being equivalent to +1.0, and +255 being equivalent to +1.99219.

Enabling and Setting Color Adjustment

You can set the Processed Raw Enable, Color Adjustment Enable, Color Adjustment Selector, Color Adjustment Hue, Color Adjustment Hue Raw, Color Adjustment Saturation, and Color Adjustment Saturation Raw parameter values from within your application software by using the Basler pylon API. In this example, we assume that you want to set your camera for Bayer BG 8 output, and therefore you must set the Processed Raw Enable parameter value to enabled.

The following code snippet illustrates using the API to set the parameter values:

```
// Set the camera for Bayer BG8 pixel data output format
Camera.PixelFormat.SetValue( PixelFormat_BayerBG8 );
// Because the camera is set for a Bayer output format, the Processed Raw
// Enabled parameter must be set to enabled
Camera.ProcessedRawEnable.SetValue( true );
```

```
// Enable the Color Adjustment feature
Camera.ColorAdjustmentEnable.SetValue( true );

// Select red as the color to adjust
Camera.ColorAdjustmentSelector.SetValue( ColorAdjustmentSelector_Red );

// Set the red hue as a floating point value
Camera.ColorAdjustmentHue.SetValue( -1.125 );
// Set the red saturation as a floating point value
Camera.ColorAdjustmentSaturation.SetValue( 1.375 );

// Select cyan as the color to adjust
Camera.ColorAdjustmentSelector.SetValue( ColorAdjustmentSelector_Cyan );

// Set the cyan hue as an integer value
Camera.ColorAdjustmentHueRaw.SetValue( -36 );
// Set the cyan saturation as an integer value
Camera.ColorAdjustmentSaturationRaw.SetValue( 176 );
```

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

For more information about the pylon API and the pylon Viewer, see Section 3 on .

Color Adjustment Reset

The camera includes a Color Adjustment Reset command that can be used to reset the color adjustments. This feature is especially useful if you have badly misadjusted the colors and you want to quickly return to reasonable settings. When the reset command is used, it will return the camera to the settings defined by your current Light Source Selector parameter setting.

You can execute the Color Adjustment Reset command from within your application software by using the pylon API. The following code snippet illustrates using the API to execute the command:

```
// Reset the color adjustments
Camera.ColorAdjustmentReset.Execute( );
```

You can also use the Basler pylon Viewer application to easily execute the command.

For more information about the pylon API and the pylon Viewer, see Section 3 on .

10.3.5 A Procedure for Setting the Color Enhancements

When setting the color enhancements on the camera, we recommend using the procedure outlined below. Since it makes changing camera parameters quick and easy, we also recommend using the Basler pylon Viewer software when you are making adjustments.

1. Arrange your camera so that it is viewing a scene similar to what it will view during actual operation. Make sure that the lighting for the scene is as close as possible to the actual lighting you will be using during normal operation. (Using lighting that represents your normal operating conditions is extremely important.)

2. We recommend including a standard color chart within your camera's field of view when you are adjusting the color enhancements. This will make it much easier to know when the colors are properly adjusted. One widely used chart is the ColorChecker® chart (also known as the Macbeth chart).

3. To start, leave the Light Source Selector parameter at the default setting.

4. Begin capturing images and check the basic image appearance. Set the exposure time and gain so that you are acquiring good quality images. It is important to make sure that the images are not over exposed. Over exposure can have a significant negative effect on the fidelity of the color in the acquired images.

5. Adjust the white balance. An easy way to set the white balance is to use the "once" function on the camera's Balance White Auto feature.

6. Set the gamma value. You should set the value to match the gamma on the monitor you are using to view acquired images. When gamma is set correctly, there should be a smooth transition from the lightest to the darkest gray scale targets on your color chart.

   (The sRGB gamma preset will give you good results on most CRT or LCD monitors.)

7. Examine the colors and see if they are satisfactory at this point. If not, chose a different setting for the Light Source Selector parameter. Try each mode and determine which one gives you the best color results.

8. The color fidelity should now be quite good. If you want to make additional changes, adjust the hue and saturation by using the color adjustment feature. Keep in mind that when you adjust a color, the colors on each side of it in the color hexagon will also be affected to some degree. For example, when you adjust red, yellow and magenta will also be affected.

   When you are making hue and saturation adjustments, it is a good idea to start by concentrating on one line in the color chart. Once you have the colors in a line properly adjusted, you can move on to each of the other lines in turn.

When you first start working with the color enhancement tools, it is easy to badly misadjust the color adjustment settings and not be able to bring them back into proper adjustment. You can easily recover from this situation by using the camera's color adjustment reset command (see page 189).

Another way to recover is to make the cameras "color factory setup" the default configuration set and then to load the default configuration set into the camera's active set. See the next section for more information about the camera's color factory setup.

10.3.6 The "Color" Factory Setup

When a camera leaves the factory, it contains several "factory setups" stored in its permanent memory. A factory setup is simply a collection of settings for the parameters needed to operate the camera. Each one of the factory setups is optimized to make the camera perform well in a particular situation. One of the setups is known as the "color factory setup", and the parameter settings contained in the color factory setup are optimized to produce good color images under the most common lighting conditions.

To make the parameters contained in the color factory setup become the ones that are actively controlling camera operation, you must select the color factory setup as the default camera configuration set and then you must load the default configuration set into the camera's active configuration set. When you do this, it will:

■   Set the Gamma Selector parameter to sRGB

■   Set the Processed Raw Enable parameter to enabled.

■   Set the Light Source Selector parameter to Daylight 6500.

■   Sets the white balance parameters to values that are suitable for daylight lighting.

If you have badly misadjusted the settings for the color enhancement features on the camera, it may be difficult to bring the settings back into proper adjustment. Selecting the color factory setup as the default set and then loading the default set into the active set is a good way to recover from gross misadjustment of the color features.

For more information about the factory setups and about selecting and loading configuration sets, see Section 12.18 on page 315.

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 14 lists the pixel formats available on each monochrome camera model and Table 15 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	●	●	●	●	●
scA640-120	●	●	●	●	●
scA750-60	●			●	●
scA780-54	●	●	●	●	●
scA1000-30	●	●	●	●	●
scA1300-32	●	●	●	●	●
scA1390-17	●	●	●	●	●
scA1400-17	●	●	●	●	●
scA1400-30	●	●	●	●	●
scA1600-14	●	●	●	●	●
scA1600-28	●	●	●	●	●

Table 14: 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	RGB 8 Packed	YUV 4:2:2 Packed	YUV 4:2:2 (YUYV) Packed
scA640-70	●		●	●	●	●	●	●
scA640-74	●		●	●	●	●	●	●
scA640-120	●		●	●	●	●	●	●
scA750-60	●	●				●	●	●
scA780-54	●		●	●	●	●	●	●
scA1000-30	●		●	●	●	●	●	●
scA1300-32	●		●	●	●	●	●	●
scA1390-17	●		●	●	●	●	●	●
scA1400-17	●		●	●	●	●	●	●
scA1400-30	●		●	●	●	●	●	●
scA1600-14	●		●	●	●	●	●	●
scA1600-28	●		●	●	●	●	●	●

Table 15: Pixel Formats Available on Color Cameras (● = format available)

> ⓘ For optimum use of the available bandwidth, the scA1600-28gc cameras are by default set to the Bayer BG 8 pixel format and all other color cameras are by default set to the YUV 4:2:2 pixel format.

Details of the monochrome formats are described in Section 11.2 on page 195 and details of the color formats are described in Section 11.3 on page 201.

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 );
Camera.PixelFormat.SetValue( PixelFormat_RGB8Packed );
```

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 3.1.1 on page 43.

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.

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

> (i) 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.

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 .

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.

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 .

11.3   Pixel Data Output Formats for Color Cameras

11.3.1   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 rows 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 rows 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 .

The tables below describe how the data for the even rows and for the odd rows 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 row

P_n = the last pixel transmitted by the camera for a row

B_0 = the first byte of data for a row

B_m = the last byte of data for a row

Even Rows	
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}

Odd Rows	
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}	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

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

11.3.2 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.)

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 .

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	•
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

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}	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

11.3.3 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 .

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 for 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

> ⓘ 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 pixel format.

11.3.4 Bayer BG 12 Packed Format

When a color camera is set for the Bayer BG 12 Packed pixel data format, 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 .

The tables below describe how the data for the even rows and for the odd rows 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 row

P_n = the last pixel transmitted by the camera for a row

B_0 = the first byte of data for a row

B_m = the last byte of data for a row

Even Rows		
Byte	**Data**	
B_0	Blue value for P_0 bits 11 ... 4	
B_1	Green value for P_1 bits 3 ... 0	Blue value for P_0 bits 3 ... 0
B_2	Green value for P_1 bits 11 ... 4	
B_3	Blue value for P_2 bits 11 ... 4	
B_4	Green value for P_3 bits 3 ... 0	Blue value for P_2 bits 3 ... 0
B_5	Green value for P_3 bits 11 ... 4	
B_6	Blue value for P_4 bits 11 ... 4	
B_7	Green value for P_5 bits 3 ... 0	Blue value for P_4 bits 3 ... 0
B_8	Green value for P_5 bits 11 ... 4	
•	•	
•	•	•
•	•	
B_{m-5}	Blue value for P_{n-3} bits 11 ... 4	
B_{m-4}	Green value for P_{n-2} bits 3 ... 0	Blue value for P_{n-3} bits 3 ... 0
B_{m-3}	Green value for P_{n-2} bits 11 ... 4	
B_{m-2}	Blue value for P_{n-1} bits 11 ... 4	
B_{m-1}	Green value for P_n bits 3 ... 0	Blue value for P_{n-1} bits 3 ... 0
B_m	Green value for P_n bits 11 ... 4	

Odd Rows		
Byte	**Data**	
B_0	Green value for P_0 bits 11 ... 4	
B_1	Red value for P_1 bits 3 ... 0	Green value for P_0 bits 3 ... 0
B_2	Red value for P_1 bits 11 ... 4	
B_3	Green value for P_2 bits 11 ... 4	
B_4	Red value for P_3 bits 3 ... 0	Green value for P_2 bits 3 ... 0
B_5	Red value for P_3 bits 11 ... 4	
B_6	Green value for P_4 bits 11 ... 4	
B_7	Red value for P_5 bits 3 ... 0	Green value for P_4 bits 3 ... 0
B_8	Red value for P_5 bits 11 ... 4	
•	•	
•	•	•
•	•	
•	•	
•	•	•
•	•	
B_{m-5}	Green value for P_{n-3} bits 11 ... 4	
B_{m-4}	Red value for P_{n-2} bits 3 ... 0	Green value for P_{n-3} bits 3 ... 0
B_{m-3}	Red value for P_{n-2} bits 11 ... 4	
B_{m-2}	Green value for P_{n-1} bits 11 ... 4	
B_{m-1}	Red value for P_n bits 3 ... 0	Green value for P_{n-1} bits 3 ... 0
B_m	Red 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

11.3.5 RGB 8 Packed Format

When a color camera is set for the RGB 8 Packed pixel data format, it outputs 8 bits of red, 8 bits of green, and 8 bits of blue data per pixel. The pixel data result from demosaicing the raw data read out of the sensor. For each pixel, the raw data were either red or green or blue data due to the presence of a Bayer filter on the sensor.

For more information about the Bayer filter, see Section 11.3.1 on page 201.

When a color camera is set for the RGB 8 Packed pixel data format, it outputs 8 bits of red data, 8 bits of green data, and 8 bits of blue data for each pixel in the acquired frame.

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 RGB 8 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	Pixel - Data Bits
B_0	R P_0 7 ... 0
B_1	G P_0 7 ... 0
B_2	B P_0 7 ... 0
B_3	R P_1 7 ... 0
B_4	G P_1 7 ... 0
B_5	B P_1 7 ... 0
•	•
•	•

Byte	Pixel - Data Bits
•	•
•	•
B_{m-5}	R P_{n-1} 7 ... 0
B_{m-4}	G P_{n-1} 7 ... 0
B_{m-3}	B P_{n-1} 7 ... 0
B_{m-2}	R P_n 7 ... 0
B_{m-1}	G P_n 7 ... 0
B_m	B P_n 7 ... 0

With the camera set for RGB 8 Packed, 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

11.3.6 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.

For more information on the Bayer filter, see Section 11.3.1 on page 201.

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.

> 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.

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".)

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:

Row $_0$ Col $_0$,	Row $_0$ Col $_1$,	Row $_0$ Col $_2$	.. ..	Row $_0$ Col $_{m-2}$,	Row $_0$ Col $_{m-1}$,	Row $_0$ Col $_m$
Row $_1$ Col $_0$,	Row $_1$ Col $_1$,	Row $_1$ Col $_2$	.. ..	Row $_1$ Col $_{m-2}$,	Row $_1$ Col $_{m-1}$,	Row $_1$ Col $_m$
Row $_2$ Col $_0$,	Row $_2$ Col $_1$,	Row $_2$ Col $_2$	.. ..	Row $_2$ Col $_{m-2}$,	Row $_2$ Col $_{m-1}$,	Row $_2$ Col $_m$
:	:	:	:	:	:	:
:	:	:	:	:	:	:
Row $_{n-2}$ Col $_0$,	Row $_{n-2}$ Col $_1$,	Row $_{n-2}$ Col $_2$	.. ..	Row $_{n-2}$ Col $_{m-2}$,	Row $_{n-2}$ Col $_{m-1}$,	Row $_{n-2}$ Col $_m$
Row $_{n-1}$ Col $_0$,	Row $_{n-1}$ Col $_1$,	Row $_{n-1}$ Col $_2$	.. ..	Row $_{n-1}$ Col $_{m-2}$,	Row $_{n-1}$ Col $_{m-1}$,	Row $_{n-1}$ Col $_m$
Row $_n$ Col $_0$,	Row $_n$ Col $_1$,	Row $_n$ Col $_2$	.. ..	Row $_n$ Col $_{m-2}$,	Row $_n$ Col $_{m-1}$,	Row $_n$ Col $_m$

Where Row $_0$ 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 Standard Features

This chapter provides detailed information about the standard features available on each camera. It also includes an explanation of their operation and the parameters associated with each feature.

12.1   Gain

The camera's gain setting is adjustable. As shown in Figure 70, 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.

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.
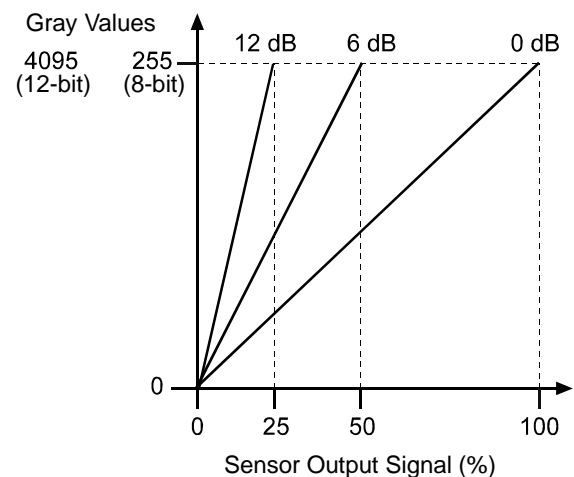
Fig. 70: Gain in dB

Setting the Gain (All Models Except scA750-60)

> (i) The information in this section applies to all camera models except the scA750-60 gm/gc. For information about scA750-60 gm/gc cameras, see the next section.

> (i) Gain can not only be manually set (see below), but can also be automatically adjusted. The Gain Auto function is the "automatic" counterpart of the gain feature and adjusts the Gain Raw parameter value automatically.
>
> For more information about auto functions, see Section 12.9.1 on .
>
> For more information about the Gain Auto function, see Section 12.9.2 on .

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 and on whether vertical binning is enabled (see Table 16). 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, RGB 8 Packed, 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	Min Setting with Vertical Binning	Max Setting (8 bit depth)	Max Setting (16 bit depth)
scA640-70	320	260	1023	511
scA640-74	280	200	1023	511
scA640-120	150	120	800	400
scA780-54	350	240	1023	511
scA1000-30	360	280	1023	511
scA1300-32	300	200	850	400
scA1390-17	360	280	1023	511
scA1400-17	192	110	1023	511
scA1400-30	150	0	850	400
scA1600-14	350	280	1023	511
scA1600-28	285	220	850	400

Table 16: 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 3.1.1 on .

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.

Formulas for All Models Listed on the Previous Page (except scA640-120, scA1300-32, scA1400-30, and scA1600-28):

For gain raw settings from 110 to 511:

$$\text{Gain}_{dB} = 20 \times \log_{10}\left(\frac{658 + \text{Gain Raw Setting}}{658 - \text{Gain Raw Setting}}\right) - G_c$$

For gain raw settings from 512 to 1023:

$$\text{Gain}_{dB} = (0.0354 \times \text{Gain Raw Setting}) - G_c$$

Where:

$$G_c = 20 \times \log_{10}\left(\frac{658 + \text{Min Gain Raw Setting}}{658 - \text{Min Gain Raw Setting}}\right)$$

Example:

Assume that you are working with a monochrome scA1400-17 camera that has a gain raw setting of 500. Calculating the gain is a two step process:

Step 1:

$$G_c = 20 \times \log_{10}\left(\frac{658 + 192}{658 - 192}\right)$$

$G_c = 5.22$ dB

Step 2:

$$\text{Gain}_{dB} = 20 \times \log_{10}\left(\frac{658 + 500}{658 - 500}\right) - 5.22 \text{ dB}$$

$$\text{Gain}_{dB} = 12.1 \text{ dB}$$

Calculation for the scA640-120, scA1300-32, scA1400-30, and scA1600-28:

For the entire range of raw settings:

$$\text{Gain}_{dB} = (0.0359 \times \text{Gain Raw Setting}) - G_c$$

Where:

$$G_c = (0.0359 \times \text{Min Gain Raw Setting})$$

Example:

Assume that you are working with a monochrome scA1400-30 camera that has a gain raw setting of 500. Calculating the gain is a two step process:

Step 1:

$$G_c = (0.0359 \times 150)$$

$$G_c = 5.385 \text{ dB}$$

Step 2:

$$\text{Gain}_{dB} = (0.0359 \times 500) - 5.385 \text{ dB}$$

$$\text{Gain}_{dB} = 12.57 \text{ dB}$$

Table 17 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
scA640-120	0	23.34	8.98
scA780-54	0	25.9	7.7
scA1000-30	0	25.5	7.3
scA1300-32	0	19.75	3.59
scA1390-17	0	25.5	7.3
scA1400-17	0	31.0	12.8
scA1400-30	0	25.13	9.0
scA1600-14	0	25.9	7.7
scA1600-28	0	20.29	4.13

Table 17: Minimum and Maximum dB of Gain

Setting the Gain (scA750-60 Only)

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

	Gain can not only be manually set (see below), but can also be automatically adjusted. The Gain Auto function is the "automatic" counterpart of the gain feature and adjusts the Gain Raw parameter value automatically.
	For more information about auto functions, see Section 12.9.1 on page 283.
	For more information about the Gain Auto function, see Section 12.9.2 on page 290.

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:

1. Set the Gain Selector to Gain All.
2. 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 3.1.1 on page 43.

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:

$$\text{Gain}_{dB} = 20 \times \log_{10}\left(1 + \frac{\text{Gain Raw Setting}}{6}\right)$$

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:

$$\text{Gain}_{dB} = 20 \times \log_{10}\left(1 + \frac{18}{6}\right)$$

$$\text{Gain}_{dB} = 12.0 \text{ dB}$$

Table 18 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 18: dB of Gain at Various Settings

12.2   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 scA640-120, scA750-60, scA1300-32, scA1400-30, and scA1600-28

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, RGB 8 Packed, 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.

Effect on scA640-120, scA1300-32, scA1400-30, and scA1600-28 Models

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, RGB 8 Packed, YUV 4:2:2 Packed, YUV 4:2:2 (YUYV) Packed), an increase of 64 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 64 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 4 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 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. On scA1300-32 gm/gc and scA1400-30 gm/gc cameras, the parameter value can range from 0 to 1023.

To set the Black Level Raw parameter value:

1. Set the Black Level Selector to Black Level All.
2. 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 3.1.1 on .

12.3 Digital Shift

> (i) The information in this section applies to all camera models except the scA750-60 gm/gc.

The digital shift feature lets you change the group of bits that is output from the ADC in the camera. Using the digital shift feature will effectively multiply the output of the camera by 2 times, 4 times, 8 times, or 16 times. The next two sections describe how the digital shift feature works when the camera is set for a 12 bit pixel format and when it is set for a 8 bit pixel format. There is also a section describing precautions that you must observe when using the digital shift feature and a section that describes enabling and setting the digital shift feature.

12.3.1 Digital Shift with 12 Bit Pixel Formats

No Shift

As mentioned in the Functional Description section of this manual, the camera uses a 12 bit ADC to digitize the output from the imaging sensor. When the camera is set for a pixel format that outputs pixel data at 12 bit effective depth, by default, the camera transmits the 12 bits that are output from the ADC.
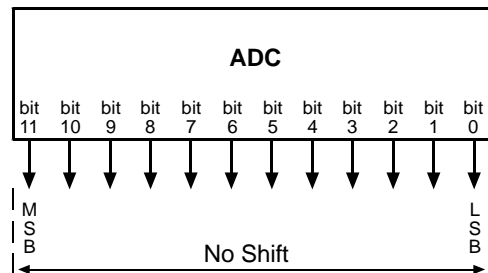
Shift by 1

When the camera is set to shift by 1, the output from the camera will include bit 10 through bit 0 from the ADC along with a zero as an LSB.

The result of shifting once is that the output of the camera is effectively multiplied by 2. For example, assume that the camera is set for no shift, that it is viewing a uniform white target, and that under these conditions the reading for the brightest pixel is 100. If you changed the digital shift setting to shift by 1, the reading would increase to 200.
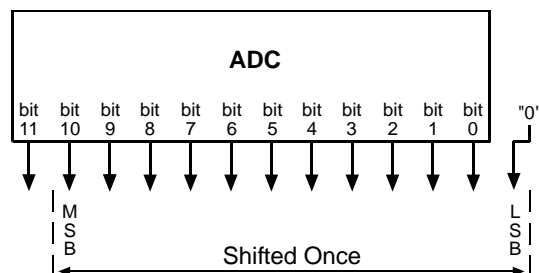
When the camera is set to shift by 1, the least significant bit output from the camera for each pixel value will be 0. This means that no odd gray values can be output and that the gray value scale will

only include values of 2, 4, 6, 8, 10, and so on. This absence of some gray values is commonly referred to as "missing codes".

If the pixel values being output by the camera's sensor are high enough to set bit 11 to 1, we recommend not using shift by 1. If you do nonetheless, all bits output from the camera will automatically be set to 1. Therefore, you should only use the shift by 1 setting when your pixel readings with a 12 bit pixel format selected and with digital shift disabled are all less than 2048.

Shift by 2

When the camera is set to shift by 2, the output from the camera will include bit 9 through bit 0 from the ADC along with 2 zeros as LSBs.

The result of shifting twice is that the output of the camera is effectively multiplied by 4.

When the camera is set to shift by 2, the 2 least significant bits output from the camera for each pixel value will be 0. This means that the gray value scale will only include every 4th value, for example, 4, 8, 16, 20, and so on.
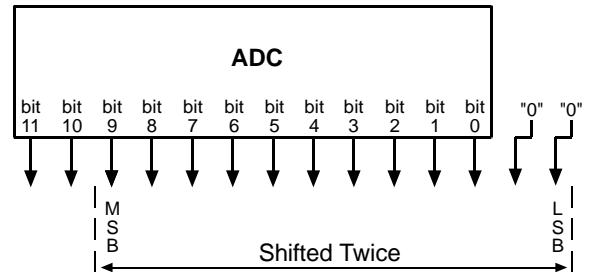
If the pixel values being output by the camera's sensor are high enough to set bit 10 or bit 11 to 1, we recommend not using shift by 2. If you do nonetheless, all bits output from the camera will automatically be set to 1. Therefore, you should only use the shift by 2 setting when your pixel readings with a 12 bit pixel format selected and with digital shift disabled are all less than 1024.

Shift By 3

When the camera is set to shift by 3, the output from the camera will include bit 8 through bit 0 from the ADC along with 3 zeros as LSBs.

The result of shifting 3 times is that the output of the camera is effectively multiplied by 8.

When the camera is set to shift by 3, the 3 least significant bits output from the camera for each pixel value will be 0. This means that the gray value scale will only include every 8th gray value, for example, 8, 16, 24, 32, and so on.
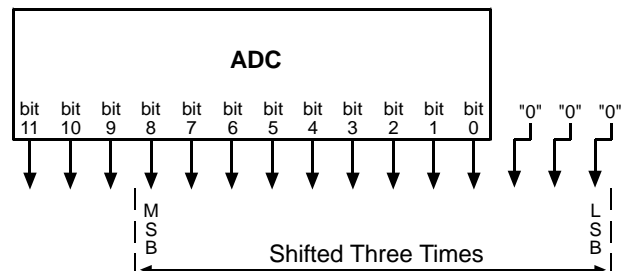
If the pixel values being output by the camera's sensor are high enough to set bit 9, bit 10, or bit 11 to 1, we recommend not using shift by 3. If you do nonetheless, all bits output from the camera will automatically be set to 1. Therefore, you should only use the shift by 3 setting when your pixel readings with a 12 bit pixel format selected and with digital shift disabled are all less than 512.

Shift By 4

When the camera is set to shift by 4, the output from the camera will include bit 7 through bit 0 from the ADC along with 4 zeros as LSBs.

The result of shifting 4 times is that the output of the camera is effectively multiplied by 16.

When the camera is set to shift by 4, the 4 least significant bits output from the camera for each pixel value will be 0. This means that the gray value scale will only include every 16th gray value, for example, 16, 32, 48, 64, and so on.

If the pixel values being output by the camera's sensor are high enough to set bit 8, bit 9, bit 10, or bit 11 to 1, we recommend not using shift by 4. If you do nonetheless, all bits output from the camera will automatically be set to 1. Therefore, you should only use the shift by 4 setting when your pixel readings with a 12 bit pixel format selected and with digital shift disabled are all less than 256.
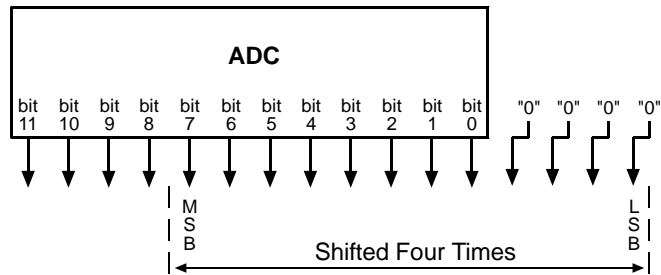
12.3.2 Digital Shift with 8 Bit Pixel Formats

No Shift

As mentioned in the Functional Description section of this manual, the camera uses a 12 bit ADC to digitize the output from the imaging sensor. When the camera is set for a pixel format that outputs pixel data at 8 bit effective depth, by default, the camera drops the 4 least significant bits from the ADC and transmits the 8 most significant bits (bit 11 through 4).
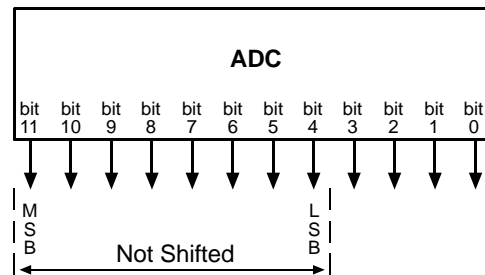
Shift by 1

When the camera is set to shift by 1, the output from the camera will include bit 10 through bit 3 from the ADC.

The result of shifting once is that the output of the camera is effectively multiplied by 2. For example, assume that the camera is set for no shift, that it is viewing a uniform white target, and that under these conditions the reading for the brightest pixel is 10. If you changed the digital shift setting to shift by 1, the reading would increase to 20.
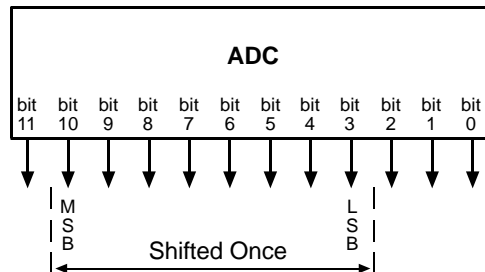
If the pixel values being output by the camera's sensor are high enough to set bit 11 to 1, we recommend not using shift by 1. If you do nonetheless, all bits output from the camera will automatically be set to 1. Therefore, you should only use the shift by 1 setting when your pixel readings with an 8 bit pixel format selected and with digital shift disabled are all less than 128.

Shift by 2

When the camera is set to shift by 2, the output from the camera will include bit 9 through bit 2 from the ADC.

The result of shifting twice is that the output of the camera is effectively multiplied by 4.

If the pixel values being output by the camera's sensor are high enough to set bit 10 or bit 11 to 1, we recommend not using shift by 2. If you do nonetheless, all bits output from the camera will automatically be set to 1. Therefore, you should only use the shift by 2 setting when your pixel readings with an 8 bit pixel format selected and with digital shift disabled are all less than 64.
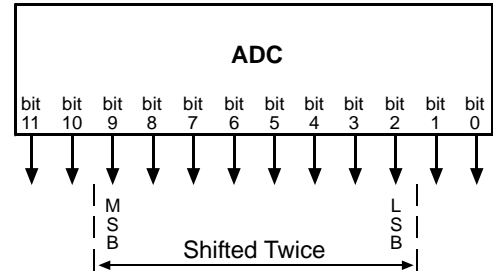
Shift by 3

When the camera is set to shift by 3, the output from the camera will include bit 8 through bit 1 from the ADC.

The result of shifting three times is that the output of the camera is effectively multiplied by 8.

If the pixel values being output by the camera's sensor are high enough to set bit 9, bit 10, or bit 11 to 1, we recommend not using shift by 3. If you do nonetheless, all bits output from the camera will automatically be set to 1. Therefore, that you should only use the shift by 3 setting when your pixel readings with an 8 bit pixel format selected and with digital shift disabled are all less than 32.
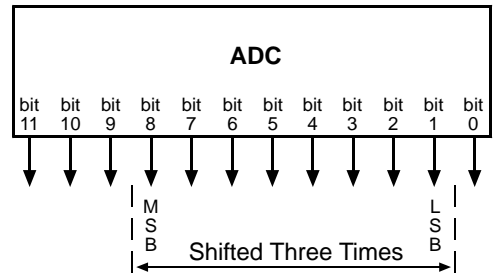
Shift by 4

When the camera is set to shift by 4, the output from the camera will include bit 7 through bit 0 from the ADC.

The result of shifting four times is that the output of the camera is effectively multiplied by 16.

If the pixel values being output by the camera's sensor are high enough to set bit 8, bit 9, bit 10, or bit 11 to 1, we recommend not using shift by 4. If you do nonetheless, all bits output from the camera will
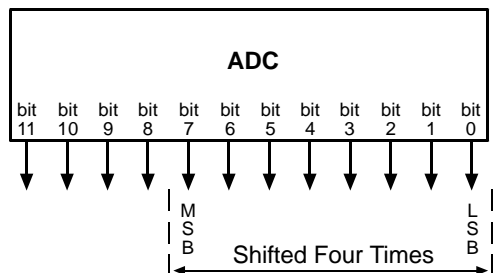
automatically be set to 1. Therefore, you should only use the multiply by 4 setting when your pixel readings with an 8 bit pixel format selected and with digital shift disabled are all less than 16.

12.3.3 Precautions When Using Digital Shift

There are several checks and precautions that you must follow before using the digital shift feature. The checks and precautions differ depending on whether the camera will be set for a 12 bit pixel format or for an 8 bit pixel format in your application.

If you will be using a 12 bit pixel format, make this check:

Use the pylon Viewer or the pylon API to set the camera for a 12 bit pixel format and **no digital shift**.

Check the output of the camera under your normal lighting conditions and note the readings for the brightest pixels.

- If any of the readings are above 2048, do not use digital shift.
- If all of the readings are below 2048, you can safely use the shift by 1 setting.
- If all of the readings are below 1024, you can safely use the shift by 1 or 2 settings.
- If all of the readings are below 512, you can safely use the shift by 1, 2, or 3 settings.
- If all of the readings are below 256, you can safely use the shift by 1, 2, 3, or 4 settings.

If you will be using an 8 bit format, make this check:

Use the pylon Viewer or the pylon API to set the camera for a 8 bit pixel format and **no digital shift**.

Check the output of the camera under your normal lighting conditions and note the readings for the brightest pixels.

- If any of the readings are above 128, do not use digital shift.
- If all of the readings are below 128, you can safely use the shift by 1 setting.
- If all of the readings are below 64, you can safely use the shift by 1 or 2 settings.
- If all of the readings are below 32, you can safely use the shift by 1, 2, or 3 settings.
- If all of the readings are below 16, you can safely use the shift by 1, 2, 3, or 4 settings.

12.3.4 Enabling and Setting Digital Shift

You can enable or disable the digital shift feature by setting the value of the Digital Shift parameter. When the parameter is set to zero, digital shift will be disabled. When the parameter is set to 1, 2, 3, or 4, digital shift will be set to shift by 1, shift by 2, shift by 3, or shift by 4 respectively.

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

```
// Disable digital shift
Camera.DigitalShift.SetValue( 0 );

// Enable digital shift by 2
Camera.DigitalShift.SetValue( 2 );
```

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 3.1.1 on .

12.4   Area of Interest (AOI)

The area of interest (AOI) feature lets you specify a portion of the 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 sensor array. The top left corner is designated as column 0 and row 0 as shown in Figure 71.

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 71.

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.
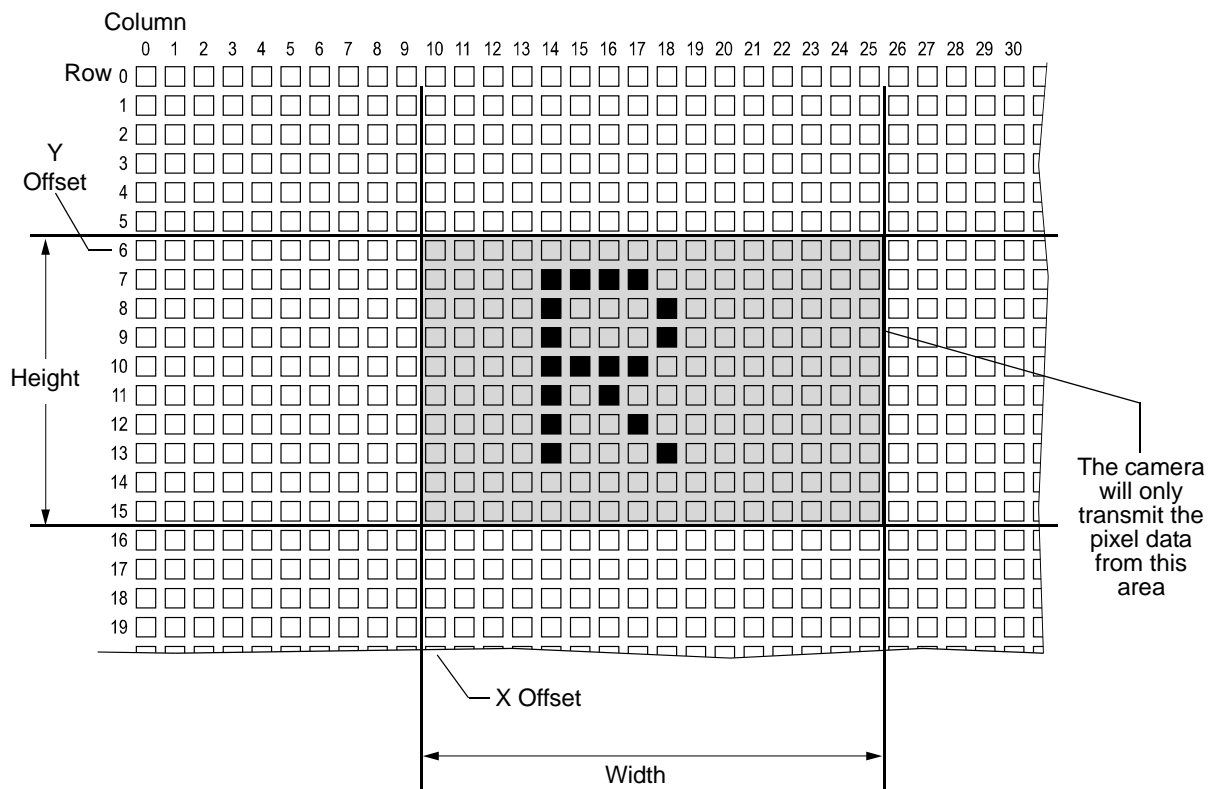
Fig. 71: Area of Interest

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 9.13 on .

Setting the AOI

Except for the scA1300-32 and scA1600-28 cameras, the AOI is set by default to use the full resolution of the camera's sensor. For the scA1300-32 cameras, the default resolution is set to 1280 x 960 pixels for mono models, and to 1278 x 958 pixels for color models. For the scA1600-28 cameras, the default resolution is set to 1626 x 1232 pixels for mono models, and to 1624 x 1230 pixels for color models.

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 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 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.

> Normally, the X Offset, Y Offset, Width, and Height parameter settings refer to the physical columns and rows in the sensor. But if binning is enabled, these parameters are set in terms of "virtual" columns and rows. For more information, see Section 12.6 on page 271.

You can set the X Offset, Y Offset, Width, and Height parameter values from within your application software by using the Basler pylon API. The following code snippets illustrate using the API to get the maximum allowed 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 3.1.1 on .

12.4.1 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 Offset X and Offset Y parameters. Changes to the AOI size are not allowed on-the-fly.

12.5   Sequencer

	The sequencer feature will not work if the auto functions feature is enabled.
(i)	For more information about the auto functions feature, see Section 12.9 on page 283.

	This section assumes that the standard mode is enabled for image acquisition control. We strongly recommend to use the standard mode when using the sequencer feature and avoid using the legacy mode.
(i)	For more information about the standard mode, see Section 8 on page 87.

	If the sequencer feature is not available, activate the "Full" camera description file.
(i)	For more information, see Section 12.19 on page 321.

The sequencer feature allows to apply specific sets of configuration parameter settings, called sequence sets, to a sequence of image acquisitions. As the images are acquired, one sequence set after the other is applied. This makes it possible to respond to different imaging requirements and conditions, that may, for example, result from changing illumination, while a sequence of images is acquired.

Three sequence advance modes provide different schemes for advancing from one sequence set to the next (see below for details).

The Sequencer and the Active Configuration Set

During operation, the camera is controlled by a set of configuration parameters that reside in the camera's volatile memory. This set of parameters is known as the active configuration set or "active set" for short. When you use the pylon API or the pylon Viewer to make a change to a camera parameter such as the Gain, you are making a change to the active set. And since the active set controls camera operation, you will see a change in camera operation when you change a parameter in the active set. For more information about the active set, see the "Configuration Sets" section.

The parameters in the active set can be divided into two types: "non-sequence" and "sequence" parameters (as shown in Figure 72 on page 238). The values of the non-sequence parameters cannot be changed using the sequencer feature. The values of the sequence parameters, however, can be set very quickly by using sequence sets: Because the sequence sets reside in the camera, you can replace the values in the active set with values from one of the sequence sets almost

instantaneously as images are acquired. Using the sequencer feature has no effect on the camera's frame rate.

The sequence set currently defining the parameter values of the active set is also called the "current set".
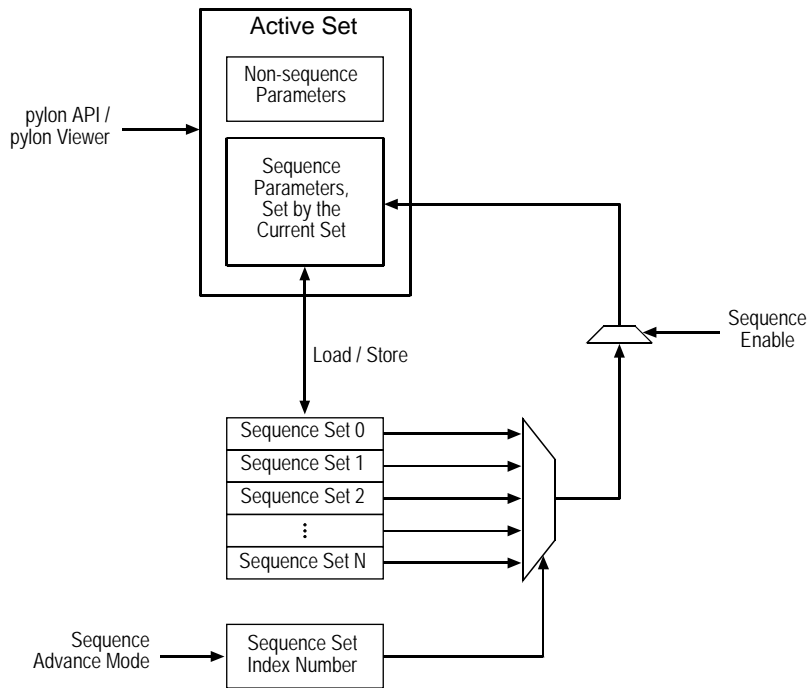
Fig. 72: Sequence Feature Block Diagram

The following parameters are included in each sequence set:

Exposure Time	Digital Shift
Enable Acquisition Frame Rate	LUT Enable
Acquisition Frame Rate	Color Transformation Value
Width	Color Transformation Matrix Factor
Height	Color Adjustment Enable
X Offset	Color Adjustment Hue
Y Offset	Color Adjustment Saturation
Center X	Chunk Mode Active
Center Y	Chunk Enable
Binning Horizontal	Timer Delay*
Binning Vertical	Timer Duration*
Pixel Format	Timer Delay Timebase*
Test Image	Timer Duration Timebase*
Gain	Sequence Set Executions**
Processed Raw Enable	
Black Level	

* This parameter is individually available for timer 1, timer 2, timer 3, and timer 4.
**This parameter is only available in auto sequence advance mode.

Sequence Set Configuration

Before the sequencer feature can be used you must populate the sequence sets with the parameter values of the sequence parameters and store the sequence sets in the camera's memory. Each sequence set is identified by a sequence set index number starting from zero. After storing, the sequence sets are available for use by the sequencer feature.

Some sequence advance modes require the storing of additional settings, for example, the total number of sequence sets you want to use, the number of consecutive uses of a sequence set or the source to control sequence set advance. For details about populating sequence sets and making related settings see the sections below explaining the sequence advance modes.

> (i) When the sequencer feature is enabled, the sequence parameter values of the current sequence set cannot be read or changed using the pylon API or the pylon Viewer. Only those sequence parameter values will be displayed that were active before the sequencer was enabled. You will not be able to "see" the parameter values set by the current set.
>
> Make sure the sequencer feature is disabled when configuring sequence sets.

<table>
<tr><td>ⓘ</td><td>Because the sequence sets only reside in volatile memory they are lost if the camera is reset or switched off. If you are using the sequencer feature, you must populate the sequence sets after each camera reset or startup.

Note also that sequence sets can not be saved in user sets.</td></tr>
</table>

Sequence Advance

As explained above, a sequence set can only control the operation of the camera after its parameter values were loaded into the active set. The loading into the active set and therefore the selection of a sequence set as the current set for a specific image acquisition are performed according to the selected sequence advance mode. The selection of a sequence set as the current set is always linked to the frame start trigger signals unless software commands are used (see below). Accordingly, a sequence advance mode provides a scheme for advancing from one sequence set to the next as frames are triggered.

The following sequence advance modes are available:

- Auto: Sequence set advance is automatically controlled by the camera. The camera will cycle through the available sequence sets in ascending sequence set index number as frames are triggered. Individual sequence sets can be used consecutively. After one sequence set cycle is complete another one will start automatically.

- Controlled: Sequence set advance is controlled by a source that can be selected. The available sources are automatic control by the camera (the "always active" setting), an input line or the "disabled" setting allowing sequence set advance only by software commands.The camera will cycle through the available sequence sets in ascending sequence set index number as frames are triggered. After one sequence set cycle is complete another one will start automatically.

- Free selection: Sequence set advance by selecting sequence sets at will from the available sequence sets. The selection is controlled by the states of the input lines.

The regular cycling through the sequence sets according to the Auto or Controlled advance modes can by modified at any time during the cycling:

- a restart starts a new sequence set cycle before the previous cycle is completed. The restart can be controlled by the states of an input line (controlled sequence advance only) or by a software command.

- a non-cyclical advance allows to skip a sequence set and will advance to the sequence set after the next. The non-cyclical advance can be controlled by a software command.

Advance or restart controlled by an input line are also called "synchronous advance" and "synchronous restart" because the checking of the states of an input line is always linked to a frame trigger signal.

Advance or restart controlled by a software command are also called "asynchronous advance" and "asynchronous restart" because they are not linked to a frame start trigger signal.

> Synchronous advance and restart are part of the standard operation of the sequencer feature and should generally be used. Asynchronous advance and restart are not suitable for standard operation because of the associated delays:
>
> The delay between sending a software command and it becoming effective will depend on the specific installation and the current load on the network. Accordingly, the number of image acquisitions that may occur between sending the software command and it becoming effective can not be predicted. Asynchronous advance and restart are therefore not suitable for real-time applications, they may, however, be useful for testing purposes.
>
> We strongly recommend to **only** use synchronous advance and synchronous restart for real-time applications.

> You can use the Sequence Set Index chunk feature to add a chunk to each acquired frame. The chunk contains the index number of the sequence set that was used for the frame acquisition. For more information about the Sequence set Index chunk, see the "Chunk Features" section.

Using the Load Command

There is also the Sequence Set Load command that may be useful when working with the sequence sets for testing purposes. If you use the Sequence Set Index parameter to select a sequence set and then you execute the Sequence Set Load command, the sequence parameter values in the active set will be replaced by the values stored in the selected sequence set.

This ability can be useful in two situations. First, if you simply want to see how the parameters currently stored in one of the sequence sets will affect camera operation, you can load the parameters from that sequence set into the active parameter set and see what happens. Second, if you want to prepare a new sequence set and you know that an existing set is already close to what you will need, you can load the existing sequence set into the active set, make some small changes to the active set, and then save the active set as a new sequence set.

Make sure the sequencer feature is disabled before issuing the Sequence Set Load command.

> Replacing the sequence parameter values in the active set via the Sequence Set Load command is associated with a delay between sending the software command and it becoming effective. The delay will depend on the specific installation and the current load on the network. Accordingly, the number of image acquisitions that may occur between sending the command and it becoming effective can not be predicted. The Sequence Set Load command is therefore not suitable for real-time applications, it may, however, be useful for testing purposes.

The following code snippet illustrates using the API to load the sequence parameter values from sequence set 0 into the active set:

```
// Select sequence set with index number 0
Camera.SequenceSetIndex.SetValue( 0 );
// Load the sequence parameter values from the sequence set into the active set
Camera.SequenceSetLoad.Execute( );
```

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

Use Case Diagrams Illustrating Sequencer Operation

The sections below explain the sequence advance modes in detail. Use case descriptions and diagrams are designed to illustrate how the sequence advance modes work in some common situations and with some common combinations of parameter settings.

These use cases do not represent every possible combination of the parameters associated with sequence advance mode operation. They are simply intended to aid you in developing an initial understanding of how the sequence advance modes work.

In each use case diagram, the black box in the upper left corner indicates how the parameters are set.

ⓘ	The use case diagrams are representational. They are not drawn to scale and are not designed to accurately describe precise camera timings.

12.5.1 Auto Sequence Advance Mode

When the auto sequence advance mode is selected the advance from one sequence set to the next occurs automatically as frame triggers are received. The advance proceeds in ascending sequence set index numbers and subject to the Sequence Set Executions parameter value. It specifies how many times each sequence set is consecutively used. After the sequence set with the highest index number was used as many times as specified by the Sequence Set Executions parameter value, the sequence set cycle starts again with sequence set 0.

The Sequence Set Total Number parameter specifies the total number of different sequence sets that are available and included within a sequence set cycle. The maximum number is 64.

Operating the Sequencer

The following use case (see also Figure 73) illustrates the operation of the sequencer in auto sequence advance mode. As images are captured continuously, the camera advances automatically with no action by the user from one sequence set to the next in ascending sequence set index numbers. The advance is also subject to the Sequence Set Executions parameter settings. After one sequence set cycle is complete, another one starts.

In this use case, the Sequence Set Total Number parameter was set to six. Accordingly, the available sequence set index numbers range from 0 through 5. The Sequence Set Executions parameter was set to 1 for sequence sets 0, 2, 3, and 4, to 2 for sequence set 5, and to 3 for sequence set 1. The frame start trigger is set for rising edge triggering.

Assuming that the camera is in the process of continuously capturing images, the sequencer feature operates as follows:

■ When the sequencer feature becomes enabled, the sequence set cycle starts: The parameter values of the sequence set with sequence set index number 0 are loaded into the active set modifying the active set.

   When a frame start trigger is received, sequence set 0 is used for the image acquisition.

■ When the next frame start trigger was received, the camera checks the current Sequence Set Executions parameter value. Because the Sequence Set Executions parameter was set to 1 for sequence set 0, this sequence set is only used once and therefore the camera advances to the next sequence set: The parameter values of sequence set 1 are loaded into the active set and are used for the image acquisition.

■ When the next frame start trigger was received, the camera checks the current Sequence Set Executions parameter value. Because the Sequence Set Executions parameter was set to 3 for sequence set 1, this sequence set is used a second time: The parameter values of sequence set 1 are used for the image acquisition.

■ When the next frame start trigger was received, the camera checks the current Sequence Set Executions parameter value. Because the Sequence Set Executions parameter was set to 3 for sequence set 1, this sequence set is used a third time: The parameter values of sequence set 1 are used for the image acquisition.

■ When the next frame start trigger was received, the camera checks the current Sequence Set Executions parameter value. Because the Sequence Set Executions parameter was set to 3 for sequence set 1, this sequence set can not, after three uses, be used again in the current sequence set cycle. Therefore, the camera advances to the next sequence set: The parameter values of sequence set 2 are used for the image acquisition.

■ When the next frame start trigger was received, the camera checks the current Sequence Set Executions parameter value. Because the Sequence Set Executions parameter was set to 1 for sequence set 2, this sequence set is only used once and therefore the camera advances to the next sequence set: The parameter values of sequence set 3 are used for the image acquisition.

■ When the next frame start trigger was received, the camera checks the current Sequence Set Executions parameter value. Because the Sequence Set Executions parameter was set to 1 for sequence set 3, this sequence set is only used once and therefore the camera advances to the next sequence set: The parameter values of sequence set 4 are used for the image acquisition.

■ When the next frame start trigger was received, the camera checks the current Sequence Set Executions parameter value. Because the Sequence Set Executions parameter was set to 1 for sequence set 4, this sequence set is only used once and therefore the camera advances to the next sequence set: The parameter values of sequence set 5 are used for the image acquisition.

■ When the next frame start trigger was received, the camera checks the current Sequence Set Executions parameter value. Because the Sequence Set Executions parameter was set to 2 for sequence set 5, this sequence set is used a second time: The parameter values of sequence set 5 are used for the image acquisition.

The camera has cycled once through the complete sequence set cycle.

▪ When the next frame start trigger was received, the camera checks the current Sequence Set Executions parameter value. Because the Sequence Set Executions parameter was set to 2 for sequence set 5, this sequence set can not, after two uses, be used again in the current sequence set cycle. Therefore the camera advances to the next sequence set: The parameter values of sequence set 0 are used for the image acquisition.

Another sequence set cycle has started.

▪ The sequencer feature is disabled while frame exposure and readout are in progress. The complete frame is transmitted and the cycling through sequence sets is terminated. The sequencer parameter values in the active set return to the values that existed before the sequencer feature was enabled.

Use Case:	Operation in auto sequence advance mode: Automatic cycling through the sequence set cycles with no action by the user. Enabling and disabling of the sequencer feature.
Settings:	Sequence Set Total Number = 6 Sequence Set Executions = 1 for sequence sets 0, 2, 3, and 4 Sequence Set Executions = 2 for sequence set 5 Sequence Set Executions = 3 for sequence set 1

▼   = camera selects a sequence set as the current sequence set

0 = current sequence set that is used for the image acquisition
(the sequence set index number is indicated)

▮   = frame exposure and readout

▼   = frame transmission
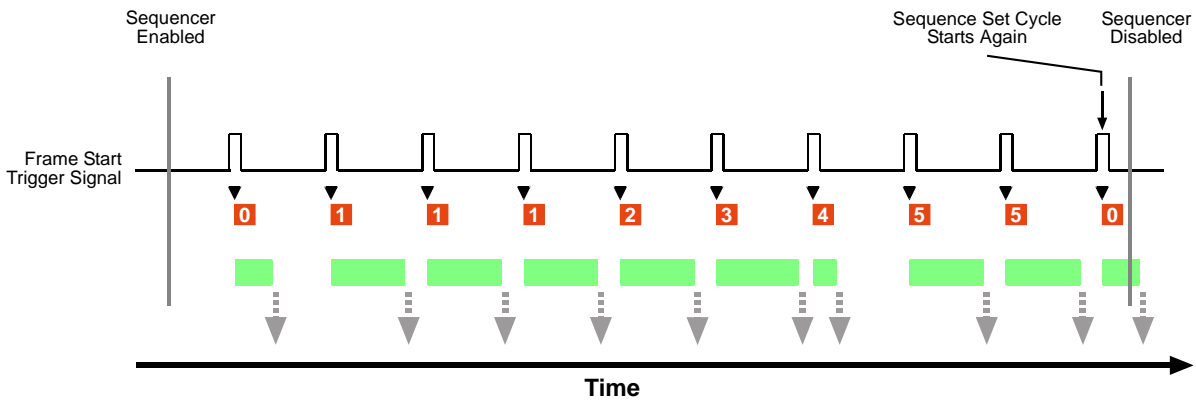
Fig. 73: Sequencer in Auto Sequence Advance Mode

Operating the Sequencer Using pylon

You can use the pylon API to set the parameters for operating the sequencer in Auto sequence advance mode from within your application software.

The following code snippet illustrates enabling the sequencer and disabling the sequencer. The example assumes that sequence sets were previously configured and are currently available in the camera's memory.

```
// Enable the sequencer feature
Camera.SequenceEnable.SetValue( true );


// Disable the sequencer feature
Camera.SequenceEnable.SetValue( false );
```

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

Configuring Sequence Sets

Use the following procedure for populating sequence sets and making the related settings:

1. Make sure that the sequencer feature is disabled.

2. Set the Sequence Advance Mode parameter to Auto.

3. Set the Sequence Set Total Number parameter. The maximum number is 64.

4. Select a sequence set index number by setting the Sequence Set Index parameter. The available numbers range from 0 to 63.

    When configuring sequence sets make sure to always use a continuous series of index numbers starting with index number 0 and ending with the Sequence Set Total Number parameter value minus one. For example, specifying a series of sequence sets only with index numbers 5, 6, and 8 is not allowed. If you did nonetheless, the not explicitly configured sequence sets would, within the scope of the sequence set total number, be populated by default parameter values.

5. Set up your first acquisition scenario (i.e., lighting, object positioning, etc.)

6. Adjust the camera parameters to get the best image quality with this scenario (you are adjusting all parameters in the active set).

7. Set the Sequence Set Executions parameter. The available numbers range from 1 to 256.

8. Execute the Sequence Set Store command to copy the sequence parameter values currently in the active set into the selected sequence set. Any already existing parameter values in the sequence set will be overwritten.

9. Repeat the above steps starting from step 4 for the other sequence sets.

Configuring Sequence Sets Using Basler pylon

You can use the pylon API to set the parameters for configuring sequence sets from within your application software.

The following code snippet gives example settings. It illustrates using the API to set the auto sequence advance mode, set the total number of sequence sets to 2, set the numbers of consecutive sequence set executions and populate sequence sets 0 and 1 by storing the sequence parameter values from the active set in the sequence sets:

```
// Disable the sequencer feature
Camera.SequenceEnable.SetValue( false );

// Set the Auto sequence advance mode
Camera.SequenceAdvanceMode.SetValue( SequenceAdvanceMode_Auto );

// Set the total number of sequence sets
Camera.SequenceSetTotalNumber.SetValue( 2 );

// Select sequence set with index number 0
Camera.SequenceSetIndex.SetValue( 0 );

// Set up the first acquisition scenario (lighting, object position, etc.) and
// adjust the camera parameters for the best image quality.

// Set the number of sequence set uses
Camera.SequenceSetExecutions.SetValue( 1 );

// Store the sequence parameter values from the active set in the selected sequence
// set
Camera.SequenceSetStore.Execute( );

// Select sequence set with index number 1
Camera.SequenceSetIndex.SetValue( 1 );

// Set up the second acquisition scenario (lighting, object position, etc.) and
// adjust the camera parameters for the best image quality.

// Set the number of sequence set uses
Camera.SequenceSetExecutions.SetValue( 4 );

// Store the sequence parameter values from the active set in the selected sequence
// set
Camera.SequenceSetStore.Execute( );
```

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

12.5.2 Controlled Sequence Advance Mode

When the controlled sequence advance mode is selected the advance from one sequence set to the next proceeds in ascending sequence set index numbers according to the selected sequence control source:

- Always Active: The advance from one sequence set to the next proceeds automatically as frame triggers are received.

- Input Line 1 or Input Line 2: The advance from one sequence set to the next proceeds according to the states of the selected input line.

- Disabled: The advance from one sequence set to the next is only controlled by AsyncAdvance software commands.

The Sequence Set Total Number parameter specifies the total number of different sequence sets that are available and included within a sequence set cycle. The maximum number is 64.

12.5.2.1 Operation with the "Always Active" Sequence Control Source

Operating the Sequencer

When the Always Active sequence control source is selected the advance from one sequence set to the next proceeds automatically in ascending sequence set index numbers as frame start triggers are received.

The following use case (see also Figure 74) illustrates the operation of the sequencer in controlled sequence advance mode with Always Active selected as the sequence control source. As images are captured continuously, the camera advances automatically with no action by the user from one sequence set to the next in ascending sequence set index numbers. After one sequence set cycle is complete, another one starts.

> ⓘ This way of operating the sequencer feature is similar to operating it in auto sequence advance mode when each sequence set is used only once per sequence set cycle.
>
> Here, however, the first sequence set used for image acquisition after the sequencer feature was enabled is sequence set 1 as opposed to sequence set 0 in auto sequence advance mode.

In this use case, the Sequence Set Total Number parameter was set to six. Accordingly, the available sequence set index numbers range from 0 through 5. The frame start trigger is set for rising edge triggering.

Assuming that the camera is in the process of continuously capturing images, the sequencer feature operates as follows:

- When the sequencer feature becomes enabled, the sequence set cycle starts: The parameter values of the sequence set with sequence set index number 0 are loaded into the active set modifying the active set.

  When a frame start trigger is received, the camera automatically advances to the next sequence set: The parameter values of sequence set 1 are used for the image acquisition.

- When the next frame start trigger was received, the camera advances to the next sequence set: The parameter values of sequence set 2 are used for the image acquisition.

- When the next frame start trigger was received, the camera advances to the next sequence set: The parameter values of sequence set 3 are used for the image acquisition.

- and so on. Note that the camera has cycled once through the complete sequence set cycle when sequence set 5 was used. With the next frame start trigger, a new sequence set cycle starts where sequence set 0 is used.

- After the sequencer feature is disabled, the cycling through sequence sets is terminated. The sequencer parameter values in the active set return to the values that existed before the sequencer feature was enabled.

Use Case: Operation in controlled sequence advance mode with Always Active as the
            sequence control source:

            Automatic cycling through the sequence set cycles with no action by the
            user. Enabling and disabling of the sequencer feature.

Setting: Sequence Set Total Number = 6

▼   = camera selects a sequence set as the current sequence set

0   = current sequence set that is used for the image acquisition
        (the sequence set index number is indicated)

        = frame exposure and readout

        = frame transmission
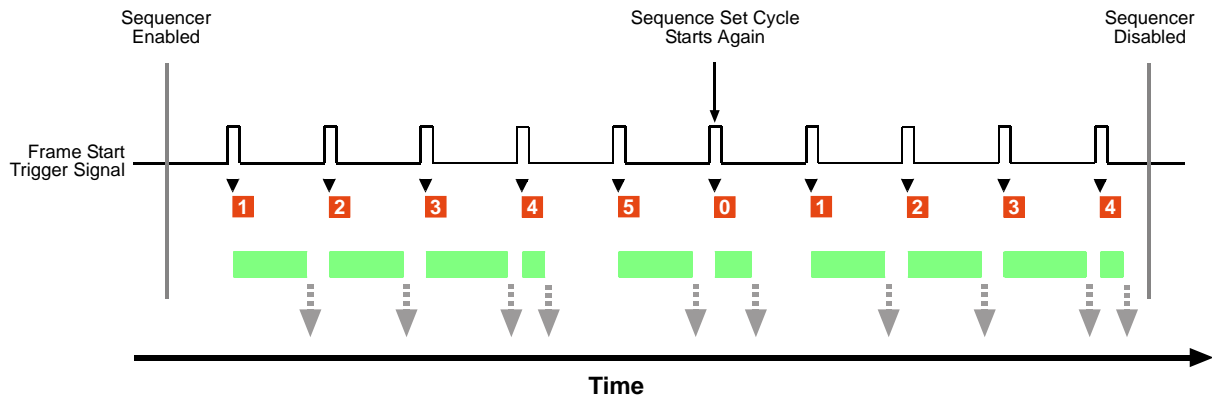
Fig. 74: Sequencer in Controlled Sequence Advance Mode with Always Active as the Sequence Control Source

Synchronous Restart

You can restart the sequence cycle with input line 1 or input line 2 as the source for controlling sequence cycle restart.

In the following use case (see also Figure 75), the same settings were made as in the previous use case: The Sequence Set Total Number parameter was set to six. Accordingly, the available sequence set index numbers range from 0 through 5. The frame start trigger is set for rising edge triggering. In addition, Line 1 was selected as the source for controlling restart. Line 1 is not set for invert.

Assuming that the camera is in the process of continuously capturing images, the sequencer feature operates as follows:

■   When the sequencer feature becomes enabled, the sequence set cycle starts: The parameter values of the sequence set with sequence set index number 0 are loaded into the active set modifying the active set.

    When a frame start trigger is received, the camera automatically advances to the next sequence set: The parameter values of sequence set 1 are loaded into the active set and are used for the

image acquisition.

- When the next frame start trigger was received, the camera advances to the next sequence set: The parameter values of sequence set 2 are used for the image acquisition.

- When the next frame start trigger was received, the camera advances to the next sequence set: The parameter values of sequence set 3 are used for the image acquisition.

- When the next frame start trigger was received, input line 1 is found to be high. Accordingly, another sequence set cycle is started and the parameter values of sequence set 0 are used for the image acquisition.

Note that the synchronous restart has priority here over the automatic sequence set advance that results from the Always Active sequence control source. Without the priority rule, sequence set 1 would be used.

Note that the state of input line 1 went high well ahead of the frame start trigger.

![info icon]	To ensure reliable synchronous sequence set restart, allow the elapse of at least one microsecond between setting the state of the input line and the rise of the frame start trigger signal.
	Also, maintain the state of the input line at least for one microsecond after the frame start trigger signal has risen.

Note also that the camera briefly exits the "waiting for frame start trigger" status while the input line changes its state. This happened when input line 1 changed its state before the fourth frame start trigger was received (see also Figure 75).

![info icon]	Make sure not to send a frame start trigger while the input line changes its state. During this period, the camera will not wait for a frame start trigger and any frame start trigger will be ignored.
	Make sure to only send a frame start trigger when the camera is in "waiting for frame start trigger" status.
	For information about possibilities of getting informed about the "waiting for frame start trigger" status, see the Acquisition Monitoring Tools section.

- When the next frame start trigger was received, the camera advances to the next sequence set: The parameter values of sequence set 1 are used for the image acquisition.

- When the next frame start trigger was received, input line 1 is found to be high. Accordingly, another sequence set cycle is started and the parameter values of sequence set 0 are used for the image acquisition. As explained above, synchronous restart has priority here over the automatic sequence set advance.

- When the next frame start triggers were received, the camera advances to the next sequence sets and uses them for image acquisition in accord with the Always Active sequence control source and as described in the previous use case.

<table>
<tr><td>Use Case:</td><td>Operation in controlled sequence advance mode with Always Active as the sequence control source:</td></tr>
<tr><td></td><td>Automatic cycling through the sequence set cycles with two synchronous restarts controlled by input line 1.</td></tr>
<tr><td>Setting:</td><td>Sequence Set Total Number = 6
Line 1 (not set for invert) is selected as the source for controlling restart</td></tr>
</table>

 = camera is waiting for a frame start trigger

▼ = camera selects a sequence set as the current sequence set

0 = current sequence set that is used for the image acquisition
(the sequence set index number is indicated)

 = frame exposure and readout

▼ = frame transmission
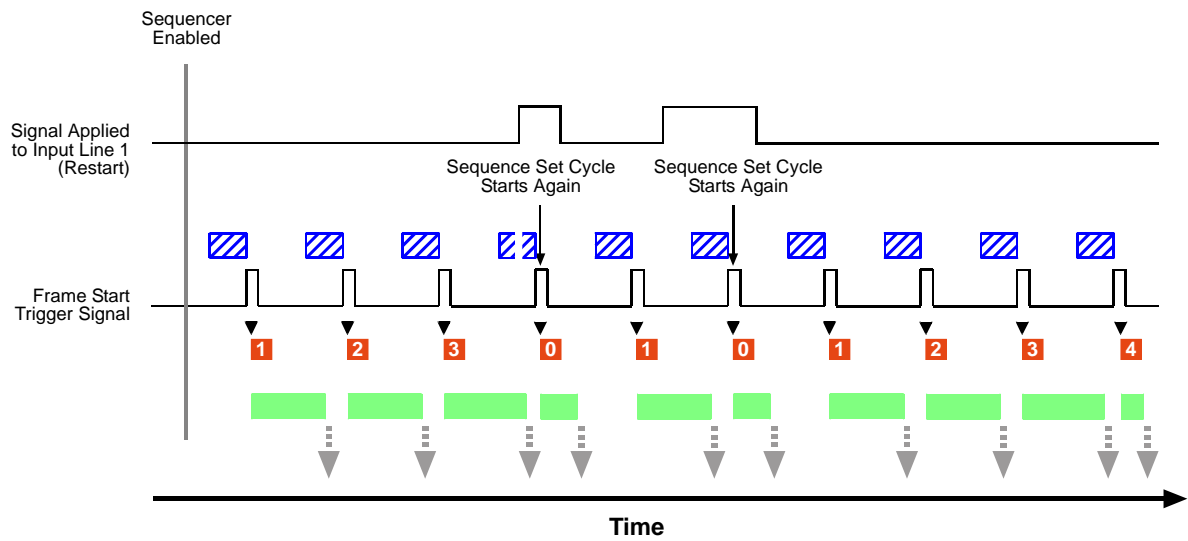
Fig. 75: Sequencer in Controlled Sequence Advance Mode with Always Active as the Sequence Control Source and Synchronous Restart Controlled by Line 1

12.5.2.2 Operation with an Input Line as Sequence Control Source

Operating the Sequencer

When the Line 1 or Line 2 sequence control source is selected the advance from one sequence set to the next is controlled according to the states of input line 1 or input line 2. The advance proceeds in ascending sequence set index numbers as frame start triggers are received.

> (i) This section assumes that Line 1 is selected as the sequence control source. All explanations, however, apply equally well to Line 2 as the sequence control source.

The following use case (see also Figure 76) illustrates the operation of the sequencer in controlled sequence advance mode with Line 1 selected as the sequence control source. The camera advances from one sequence set to the next in ascending sequence set index numbers. After one sequence set cycle is complete, another one starts. The sequence set advance is controlled by the states of Line 1. Line 1 is not set for invert.

In this use case, the Sequence Set Total Number parameter was set to six. Accordingly, the available sequence set index numbers range from 0 through 5. The frame start trigger is set for rising edge triggering.

Assuming that the camera is in the process of continuously capturing images, the sequencer feature operates as follows:

- When the sequencer feature becomes enabled, the sequence set cycle starts: The parameter values of the sequence set with sequence set index number 0 are loaded into the active set modifying the active set.

  When a frame start trigger is received, the camera checks the state of input line 1. Input line 1 is found to be low (the line status equals zero) and therefore no new sequence parameter values are loaded into the active set. The parameter values of sequence set 0 are used for the image acquisition.

- When the next frame start trigger was received, the camera checks the state of input line 1. Input line 1 is found to be high (the line status equals one) and therefore the parameter values of the next sequence set are loaded into the active set. The parameter values of sequence set 1 are used for the image acquisition.

  Note that the state of input line 1 went high well ahead of the frame start trigger.

> (i) To ensure reliable selection of a sequence set, allow the elapse of at least one microsecond between setting the states of the input lines and the rise of the frame start trigger signal.
>
> Also, maintain the states of the input lines at least for one microsecond after the frame start trigger signal has risen.

Note also that the camera briefly exits the "waiting for frame start trigger" status while an input line changes its state. This happened when input line 1 changed its state before the second frame start trigger was received (see also Figure 76).

> ⓘ Make sure not to send a frame start trigger while the input line changes its state. During this period, the camera will not wait for a frame start trigger and any frame start trigger will be ignored.
>
> Make sure to only send a frame start trigger when the camera is in "waiting for frame start trigger" status.
>
> For information about possibilities of getting informed about the "waiting for frame trigger" status, see the Acquisition Monitoring Tools section.

- When the next frame start trigger was received, the camera checks the state of input line 1. Input line 1 is found to be low and therefore no new sequence parameter values are loaded into the active set. The parameter values of sequence set 1 are used for the image acquisition.

- When the next frame start trigger was received, the camera checks the state of input line 1. Input line 1 is found to be low and therefore no new sequence parameter values are loaded into the active set. The parameter values of sequence set 1 are used for the image acquisition.

- When the next frame start trigger was received, the camera checks the state of input line 1. Input line 1 is found to be high and therefore the parameter values of the next sequence set are loaded into the active set. The parameter values of sequence set 2 are used for the image acquisition.

- When the next frame start trigger was received, the camera checks the state of input line 1. Input line 1 is found to be high and therefore the parameter values of the next sequence set are loaded into the active set. The parameter values of sequence set 3 are used for the image acquisition.

- When the next frame start trigger was received, the camera checks the state of input line 1. Input line 1 is found to be high and therefore the parameter values of the next sequence set are loaded into the active set. The parameter values of sequence set 4 are used for the image acquisition.

- When the next frame start trigger was received, the camera checks the state of input line 1. Input line 1 is found to be high and therefore the parameter values of the next sequence set are loaded into the active set. The parameter values of sequence set 5 are used for the image acquisition.

- When the next frame start trigger was received, the camera checks the state of input line 1. Input line 1 is found to be low and therefore no new sequence parameter values are loaded into the active set. The parameter values of sequence set 5 are used for the image acquisition.

  The camera has cycled once through the complete sequence set cycle.

- When the next frame start trigger was received, the camera checks the state of input line 1. Input line 1 is found to be high and therefore the parameter values of the next sequence set are loaded into the active set. The parameter values of sequence set 0 are used for the image acquisition.

Another sequence set cycle has started.

■ After frame exposure and readout are completed, the sequencer feature is disabled. The cycling through sequence sets is terminated. The sequencer parameter values in the active set return to the values that existed before the sequencer feature was enabled.

Use Case:	Operation in controlled sequence advance mode with Line 1 as the sequence control source: Cycling through the sequence set cycles according to the states of input line 1 (not set for invert). Enabling and disabling of the sequencer feature.
Setting:	Sequence Set Total Number = 6

▨ = camera is waiting for a frame start trigger

▼ = camera selects a sequence set as the current sequence set

0 = current sequence set that is used for the image acquisition
(the sequence set index number is indicated)

▣ = frame exposure and readout
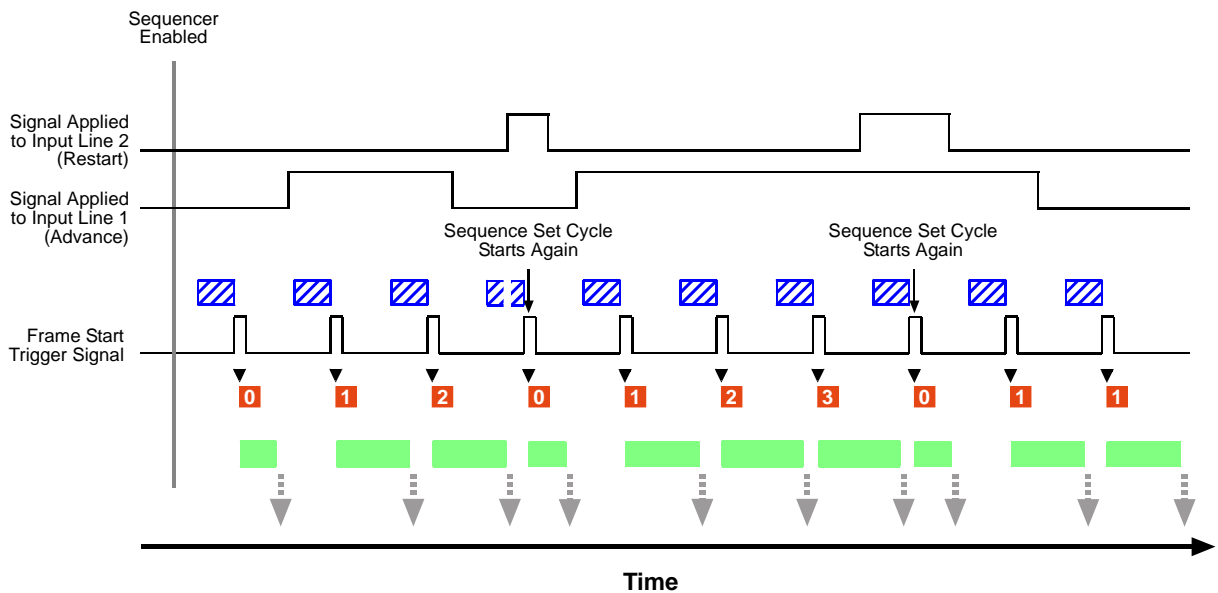
▼ = frame transmission
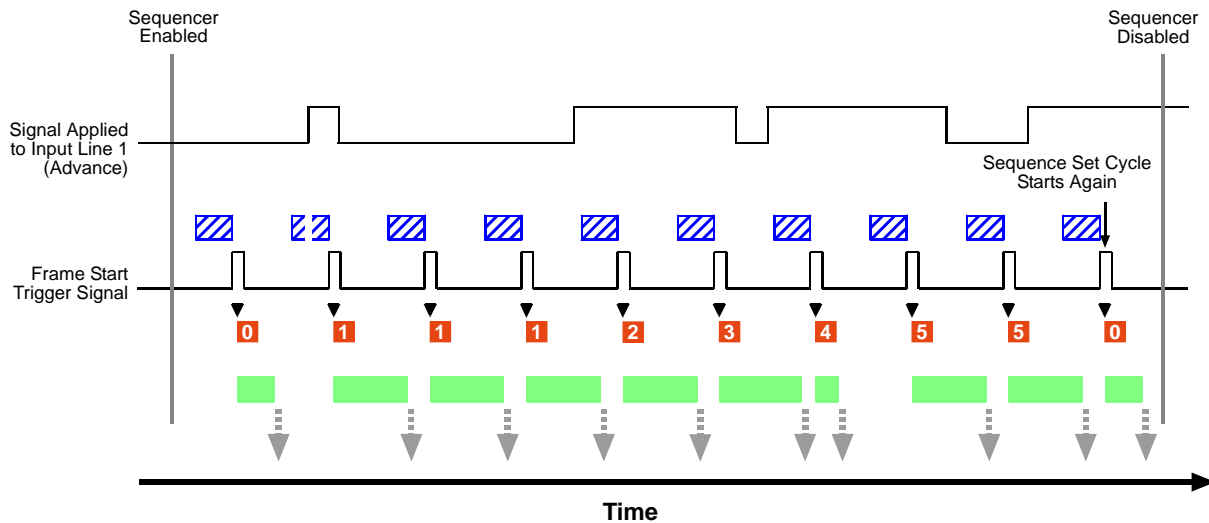
Fig. 76: Sequencer in Controlled Sequence Advance Mode with Line 1 as the Sequence Control Source

Synchronous Restart

You can restart the sequence cycle by selecting the input line that is not used for sequence advance control as the source for controlling sequence cycle restart.

In the following use case (see also Figure 77), the same settings were made as in the previous use case: The Sequence Set Total Number parameter was set to six. Accordingly, the sequence set index numbers range from 0 through 5. The frame start trigger is set for rising edge triggering.

Line 1 was selected as the sequence control source for controlling sequence set advance. In addition, Line 2 was selected as the source for controlling sequence cycle restart. Both input lines are not set for invert.

Assuming that the camera is in the process of continuously capturing images, the sequencer feature operates as follows:

■   When the sequencer feature becomes enabled, the sequence set cycle starts: The parameter values of the sequence set with sequence set index number 0 are loaded into the active set modifying the active set.

When a frame start trigger was received, the camera checks the states of input lines 2 and 1. Input line 2 is found to be low and therefore the sequence cycle is not restarted. Input line 1 is found to be low and therefore no new sequence parameter values are loaded into the active set. The parameter values of sequence set 0 are used for the image acquisition.

■   When the next frame start trigger was received, the camera checks the states of input lines 2 and 1. Input line 2 is found to be low and therefore the sequence cycle is not restarted. Input line 1 is found to be high and therefore the parameter values of the next sequence set are loaded into the active set. The parameter values of sequence set 1 are used for the image acquisition.

■   When the next frame start trigger was received, the camera checks the states of input lines 2 and 1. Input line 2 is found to be low and therefore the sequence cycle is not restarted. Input line 1 is found to be high and therefore the parameter values of the next sequence set are loaded into the active set. The parameter values of sequence set 2 are used for the image acquisition.

■   When the next frame start trigger was received, the camera checks the states of input lines 2 and 1. Input line 2 is found to be high and therefore the sequence cycle is restarted. Input line 1 is found to be low but this has no significance: Synchronous restart has priority over the sequence set advance control. The parameter values of sequence set 0 are used for the image acquisition.

Another sequence set cycle has started.

Note that the state of input line 2 went high well ahead of the frame start trigger.

	To ensure reliable selection of a sequence set, allow the elapse of at least one microsecond between setting the states of the input lines and the rise of the frame start trigger signal.
	Also, maintain the states of the input lines at least for one microsecond after the frame start trigger signal has risen.

Note also that the camera briefly exits the "waiting for frame start trigger" status while an input line changes its state. This happened, for example, when input line 2 changed its state before the fourth frame start trigger was received (see also Figure 77).

| | Make sure not to send a frame start trigger while an input line changes its state. During this period, the camera will not wait for a frame start trigger and any frame start trigger will be ignored.

Make sure to only send a frame start trigger when the camera is in "waiting for frame start trigger" status.

For information about possibilities of getting informed about the "waiting for frame trigger" status, see the Acquisition Monitoring Tools section. |
|---|---|

- When the next frame start trigger was received, the camera checks the states of input lines 2 and 1. Input line 2 is found to be low and therefore the sequence cycle is not restarted. Input line 1 is found to be high and therefore the parameter values of the next sequence set are loaded into the active set. The parameter values of sequence set 1 are used for the image acquisition.

- When the next frame start trigger was received, the camera checks the states of input lines 2 and 1. Input line 2 is found to be low and therefore the sequence cycle is not restarted. Input line 1 is found to be high and therefore the parameter values of the next sequence set are loaded into the active set. The parameter values of sequence set 2 are used for the image acquisition.

- When the next frame start trigger was received, the camera checks the states of input lines 2 and 1. Input line 2 is found to be low and therefore the sequence cycle is not restarted. Input line 1 is found to be high and therefore the parameter values of the next sequence set are loaded into the active set. The parameter values of sequence set 3 are used for the image acquisition.

- When the next frame start trigger was received, the camera checks the states of input lines 2 and 1. Input line 2 is found to be high and therefore the sequence cycle is restarted. Input line 1 is found to be high but this has no significance: Synchronous restart has priority over the sequence set advance control. The parameter values of sequence set 0 are used for the image acquisition.

  Another sequence set cycle has started.

- When the next frame start trigger was received, the camera checks the states of input lines 2 and 1. Input line 2 is found to be low and therefore the sequence cycle is not restarted. Input line 1 is found to be high and therefore the parameter values of the next sequence set are loaded into the active set. The parameter values of sequence set 1 are used for the image acquisition.

- When the next frame start trigger was received, the camera checks the states of input lines 2 and 1. Input line 2 is found to be low and therefore the sequence cycle is not restarted. Input line 1 is found to be low and therefore no new sequence parameter values are loaded into the active set. The parameter values of sequence set 1 are used for the image acquisition.

Use Case: Operation in controlled sequence advance mode with Line 1 as the sequence control source:

Cycling through the sequence set cycles according to the states of input line 1 (not set for invert) with two synchronous restarts controlled by input line 2.

Setting: Sequence Set Total Number = 6
Line 2 (not set for invert) is selected as the source for controlling restart

= camera is waiting for a frame start trigger

▼ = camera selects a sequence set as the current sequence set

0 = current sequence set that is used for the image acquisition (the sequence set index number is indicated)

= frame exposure and readout

= frame transmission

Fig. 77: Sequencer in Controlled Sequence Advance Mode with Line 1 as the Sequence Control Source and Synchronous Restart Controlled by Line 2

12.5.2.3 Operation with the "Disabled" Sequence Control Source

Operating the Sequencer

When the Disabled sequence control source is selected the advance from one sequence set to the next proceeds in ascending sequence set index numbers and is only possible by asynchronous advance.

Similarly, sequence set restart is only possible by asynchronous restart.

	The delay between sending an AsyncAdvance or an AsyncRestart software command and it becoming effective will depend on the specific installation and the current load on the network. Accordingly, the number of image acquisitions that may occur between sending the software command and it becoming effective can not be predicted. Using the sequencer feature with Disabled sequence control source is therefore not suitable for real-time applications, it may, however, be useful for testing purposes.
	We strongly recommend **not** to use the sequencer feature with Disabled sequence control source for real-time applications.

The following use case (see also Figure 78) illustrates the operation of the sequencer in controlled sequence advance mode with Disabled selected as the sequence control source. Sequence set advance proceeds in ascending sequence set index numbers subject to asynchronous advance commands. After one sequence set cycle is complete, another one starts. Sequence set cycle restarts are subject to asynchronous restart commands.

In this use case, the Sequence Set Total Number parameter was set to six. Accordingly, the available sequence set index numbers range from 0 through 5. The frame start trigger is set for rising edge triggering.

Assuming that the camera is in the process of continuously capturing images, the sequencer feature operates as follows:

■   When the sequencer feature becomes enabled, the sequence set cycle starts: The parameter values of the sequence set with sequence set index number 0 are loaded into the active set modifying the active set.

When a frame start trigger is received, the camera checks the active set and uses it for the image acquisition. The parameter values of sequence set 0 are used.

■   An AsyncAdvance command is sent. After some delay, the parameter values of the next sequence set will be loaded into the active set. It is assumed here that the delay between sending the AsyncRestart command and it becoming effective will allow the acquisition of two more images.

■   When the next frame start trigger is received, the camera checks the active set and uses it for the image acquisition. The parameter values of sequence set 0 are used.

The AsyncAdvance command has not yet become effective because of the assumed associated delay.

■   When the next frame start trigger is received, the camera checks the active set and uses it for the image acquisition. The parameter values of sequence set 0 are used.

The AsyncAdvance command has not yet become effective because of the assumed associated delay.

▪ When the AsyncAdvance command becomes effective, the camera happens to be in "waiting for frame start trigger" status. The parameter values of the next sequence set, i.e. of sequence set 1, are loaded into the active set. Note that the camera briefly exits the "waiting for frame start trigger" status while the parameter values of sequence set 1 are loaded into the active set (see also Figure 78).

ⓘ Make sure not to send a frame start trigger while the parameter values of a sequence set are loaded into the active set. During this period, the camera will not wait for a frame start trigger and any frame start trigger will be ignored.

Make sure to only send a frame start trigger when the camera is in "waiting for frame start trigger" status.

For information about possibilities of getting informed about the "waiting for frame start trigger" status, see the Acquisition Monitoring Tools section.

▪ When the next frame start trigger is received, the camera checks the active set and uses it for the image acquisition. The parameter values of sequence set 1 are used.

▪ When the next frame start trigger is received, the camera checks the active set and uses it for the image acquisition. The parameter values of sequence set 1 are used.

▪ When the next frame start trigger is received, the camera checks the active set and uses it for the image acquisition. The parameter values of sequence set 1 are used.

▪ An AsyncRestart command is sent. After some delay, the parameter values of sequence set 0 will be loaded into the active set. It is assumed here that the delay between sending the AsyncRestart command and it becoming effective will allow the acquisition of two more images.

▪ When the next frame start trigger is received, the camera checks the active set and uses it for the image acquisition. The parameter values of sequence set 1 are used.

The AsyncRestart command has not yet become effective because of the assumed associated delay.

▪ When the next frame start trigger is received, the camera checks the active set and uses it for the image acquisition. The parameter values of sequence set 1 are used.

The AsyncRestart command has not yet become effective because of the assumed associated delay.

▪ When the AsyncRestart command becomes effective, the camera happens to be in "waiting for frame start trigger" status. The parameter values of sequence set 0 are loaded into the active set. Note that the camera briefly exits the "waiting for frame start trigger" status while the parameter values of sequence set 1 are loaded into the active set (see also Figure 78).

> (i) Make sure not to send a frame start trigger while the parameter values of a
> sequence set are loaded into the active set. During this period, the camera will
> not wait for a frame start trigger and any frame start trigger will be ignored.
>
> Make sure to only send a frame start trigger when the camera is in "waiting for
> frame start trigger" status.
>
> For information about possibilities of getting informed about the "waiting for
> frame start trigger" status, see the Acquisition Monitoring Tools section.

- When the next frame start trigger is received, the camera checks the active set and uses it for
  the image acquisition. The parameter values of sequence set 0 are used.

  Another sequence set cycle has started

- When the next frame start trigger is received, the camera checks the active set and uses it for
  the image acquisition. The parameter values of sequence set 0 are used.

- While frame exposure and readout are in progress, the sequencer feature is disabled. The
  complete frame is transmitted and the cycling through sequence sets is terminated. The
  sequencer parameter values in the active set return to the values that existed before the
  sequencer feature was enabled.

Use Case:	Operation in controlled sequence advance mode with Disabled sequence control source:
	Cycling through the sequence set cycles only due to one asynchronous advance and one asynchronous restart. Enabling and disabling of the sequencer feature.
Setting:	Sequence Set Total Number = 6

= asynchronous advance (AsyncAdvance command)

...... = delay between sending the advance command and it becoming effective

= asynchronous restart (AsyncRestart command)

...... = delay between sending the restart command and it becoming effective

= camera is waiting for a frame start trigger

▼ = camera selects a sequence set as the current sequence set

0 = current sequence set that is used for the image acquisition
(the sequence set index number is indicated)

= frame exposure and readout

= frame transmission
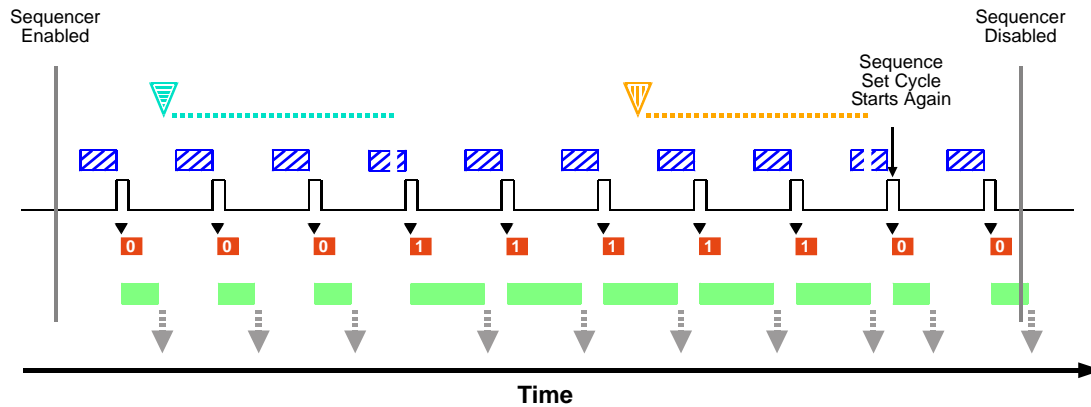
Fig. 78: Sequencer in Controlled Sequence Advance Mode with Disabled as the Sequence Control Source and Asynchronous Advance and Restart

Operating the Sequencer Using Basler pylon

You can use the pylon API to set the parameters for operating the sequencer in Controlled sequence advance mode from within your application software.

The following code snippet illustrates enabling and disabling the sequencer. The example assumes that sequence sets were previously configured and are currently available in the camera's memory.

```
// Enable the sequencer feature
Camera.SequenceEnable.SetValue( true );


// Disable the sequencer feature
Camera.SequenceEnable.SetValue( false );
```

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

12.5.2.4   Configuration

Configuring Sequence Sets and Advance Control

Use the following procedure for populating sequence sets and setting the sources for sequence set advance and sequence cycle restart:

◼ Make sure that the sequencer feature is disabled.

◼ Set the Sequence Advance mode to Controlled.

◼ Set the Sequence Set Total Number parameter. The maximum number is 64.

◼ Set the Sequence Control Selector parameter to Advance to configure sequence set advance.

◼ Set the Sequence Control Source parameter to specify the source that will control synchronous sequence set advance. Note that the same source will apply to all sequence sets available at the same time in the camera.

   The following sources are available:

   ◼ Always Active

   ◼ Line 1:

   ◼ Line 2:

   ◼ Disabled

◼ Set the Sequence Control Selector parameter to Restart to configure sequence set cycle restart.

◼ Set the Sequence Control Source parameter to specify the source for restart.

(i)	Never choose the same source for sequence set advance and sequence set cycle restart, with one exception:
	If you want to only use asynchronous advance and restart, choose Disabled as the source for advance and restart.

The following sources are available:

- Line 1:
- Line 2:
- Disabled

- Select a sequence set index number by setting the Sequence Set Index parameter. The available numbers range from 0 to 63.

  When selecting index numbers for configuring, make sure to always start a sequence with 0 and to only set a continuous series of index numbers. For example, specifying a sequence of sets only with index numbers 5, 6, and 8 is therefore not allowed. If you did nonetheless, the not explicitly configured sequence sets would - within the scope of the sequence set total number - be populated by default parameter values.

- Set up your first acquisition scenario (i.e., lighting, object positioning, etc.)

- Adjust the camera parameters to get the best image quality with this scenario (you are adjusting the parameters in the active set).

- Execute the Sequence Set Store command to copy the sequence parameter values currently in the active set into the selected sequence set. (Any existing parameter values in the sequence set will be overwritten.)

- Repeat the above steps for the other sequence sets.

For information about setting the input line for invert, see Section 8.1.3 on .

Configuring Sequence Sets and Advance Control Using Basler pylon

You can use the pylon API to set the parameters for configuring sequence sets from within your application software.

The following code snippet gives example settings. It illustrates using the API to set the controlled sequence advance mode. In the example, Line 1 is set as the sequence control source for synchronous sequence set advance, Disabled is set as the sequence control source to allow asynchronous sequence cycle reset, the total number of sequence sets is set to 2, sequence sets 0 and 1 are populated by storing the sequence parameter values from the active set in the sequence sets, and to enable the sequencer feature :

```
// Disable the sequencer feature
Camera.SequenceEnable.SetValue( false );


// Set the Controlled sequence advance mode and set line 1 as the sequence
// control source for synchronous sequence set advance
Camera.SequenceAdvanceMode.SetValue( SequenceAdvanceMode_Controlled );
Camera.SequenceControlSelector.SetValue( SequenceControlSelector_Advance );
Camera.SequenceControlSource.SetValue( SequenceControlSource_Line1 );


// Set Disabled as the source because synchronous sequence set cycle restart
// will not be used
Camera.SequenceControlSelector.SetValue( SequenceControlSelector_Restart );
Camera.SequenceControlSource.SetValue( SequenceControlSource_Disabled );
```

```
// Set the total number of sequence sets
Camera.SequenceSetTotalNumber.SetValue( 2 );


// Select sequence set with index number 0
Camera.SequenceSetIndex.SetValue( 0 );


// Set up the first acquisition scenario (lighting, object position, etc.) and
// adjust the camera parameters for the best image quality.


// Store the sequence parameter values from the active set in the selected
// sequence set
Camera.SequenceSetStore.Execute( );


// Select sequence set with index number 1
Camera.SequenceSetIndex.SetValue( 1 );


// Set up the second acquisition scenario (lighting, object position, etc.) and
// adjust the camera parameters for the best image quality.


// Store the sequence parameter values from the active set in the selected
// sequence set
Camera.SequenceSetStore.Execute( );


// Enable the sequencer feature
Camera.SequenceEnable.SetValue( true );
```

The following code snippet illustrates using the API to load the sequence parameter values from sequence set 0 into the active set:

```
// Select sequence set with index number 0
Camera.SequenceSetIndex.SetValue( 0 );
// Load the sequence parameter values from the sequence set into the active set
Camera.SequenceSetLoad.Execute( );
```

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

12.5.3 Free Selection Sequence Advance Mode

When the free selection sequence advance mode is selected the advance from one sequence set to the next as frame start triggers are received does not adhere to a specific preset sequence: The sequence sets can be selected at will using the states of input lines: The states of the input lines set the sequence set addresses. These correspond to the sequence set index numbers and accordingly, the related sequence set is selected. For details about selecting sequence sets via the sequence set address, see the "Selecting Sequence Sets" section.

The states of two input lines are checked if more than two sequence sets are available. The states of one input line is checked when only two sequence sets are available.

The Sequence Set Total Number parameter specifies the total number of different sequence sets that are available. The maximum number is 4.

12.5.3.1 Operation

Operating the Sequencer

The following use case (see also Figure 79) illustrates the operation of the sequencer in free selection sequence advance mode.

In this use case, the Sequence Set Total Number parameter was set to four. Accordingly, the sequence set index numbers range from 0 through 3. Input line 1 sets bit 0 of the sequence set address. Input line 2 sets bit 1 of the sequence set address. Both input lines are not set for invert. The frame start trigger is set for rising edge triggering.

Assuming that the camera is in the process of continuously capturing images, the sequencer feature operates as follows:

■   When the sequencer feature becomes enabled and a frame start trigger was received, the camera checks the states of input lines 1 and 2. Input line 1 is found to be high and input line 2 is found to be low. This corresponds to the address of sequence set 1. Accordingly, sequence set 1 is selected. Its parameter values are loaded into the active set and are used for the image acquisition.

Note that the state of input line 1 went high well ahead of the frame start trigger.

ⓘ	To ensure reliable selection of a sequence set, allow the elapse of at least one microsecond between setting the states of the input lines and the rise of the frame start trigger signal.
	Also, maintain the states of the input lines at least for one microsecond after the frame start trigger signal has risen.

Note also that the camera briefly exits the "waiting for frame start trigger" status while an input line changes its state. This happened, for example, when input line 1 changed its state before the first frame start trigger was received (see also Figure 79).

> ⓘ Make sure not to send a frame start trigger while an input line changes its
>   state. During this period, the camera will not wait for a frame start trigger and
>   any frame start trigger will be ignored.
>
>   Make sure to only send a frame start trigger when the camera is in "waiting for
>   frame start trigger" status.
>
>   For information about possibilities of getting informed about the "waiting for
>   frame trigger" status, see the Acquisition Monitoring Tools section.

- When the next frame start trigger was received, the camera checks the states of input lines 1
  and 2. Because the states have not changed the parameter values of sequence set 1 are used
  for the image acquisition.

- When the next frame start trigger was received, the camera checks the states of input lines 1
  and 2. The states of both input lines are found to be low. This corresponds to the address of
  sequence set 0. Accordingly, sequence set 0 is selected. The parameter values of sequence
  set 0 are used for the image acquisition.

- When the next frame start trigger was received, the camera checks the states of input lines 1
  and 2. Input line 1 is found to be low and input line 2 is found to be high. This corresponds to
  the address of sequence set 2. Accordingly, sequence set 2 is selected. The parameter values
  of sequence set 0 are used for the image acquisition.

- The sequence sets for the next five frame start triggers are selected and used according to the
  scheme that applied to the preceding frame start triggers.

- While frame exposure and readout for the fifth frame start trigger are in progress, the
  sequencer feature is disabled. The complete frame is transmitted. The sequencer parameter
  values in the active set return to the values that existed before the sequencer feature was
  enabled.

Use Case: Operation in free selection sequence advance mode.
           Sequence sets are selected at will. The selection is controlled by the states
           of the input lines.

Settings: Sequence Set Total Number = 4
           Input line 1 (not set for invert) sets bit 0 of the sequence set address.
           Input line 2 (not set for invert) sets bit 1of the sequence set address.

▨ = camera is waiting for a frame start trigger

▼ = camera selects a sequence set as the current sequence set

0 = current sequence set that is used for the image acquisition
     (the sequence set index number is indicated)

▭ = frame exposure and readout

▼ = frame transmission
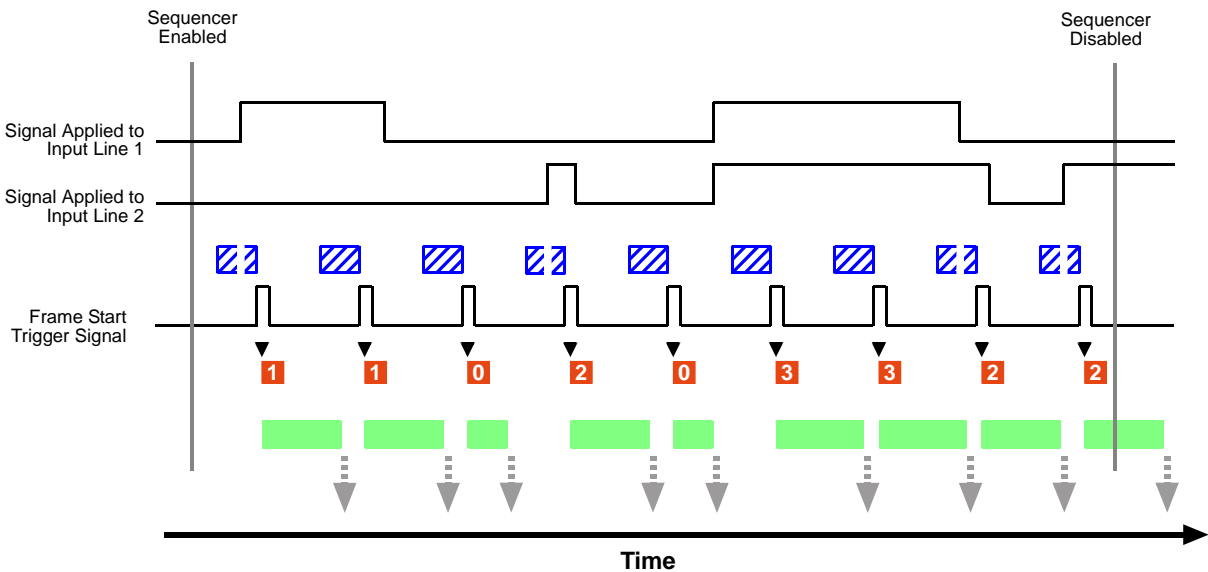
Fig. 79: Sequencer in Free Selection Mode

Operating the Sequencer Using Basler pylon

You can use the pylon API to set the parameters for operating the sequencer in Free Selection sequence advance mode from within your application software.

The following code snippet illustrates enabling and disabling the sequencer. The example assumes that sequence sets were previously configured and are currently available in the camera's memory.

```
// Enable the sequencer feature
Camera.SequenceEnable.SetValue( true );
```

```
// Disable the sequencer feature
Camera.SequenceEnable.SetValue( false );
```

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

Selecting Sequence Sets

Each sequence set is identified by a sequence set index number, starting from zero. The states of the input lines select between the sequence sets by setting the (big endian) sequence set addresses. The addresses are simply the binary expressions of the sequence set index numbers. A maximum of four sequence sets can be used:

Sequence Set Address		Related Sequence Set
Bit 1	**Bit 0**	
0	0	Sequence Set 0
0	1	Sequence Set 1
1	0	Sequence Set 2
1	1	Sequence Set 3

Table 19: Sequence Set Addresses and Related Sequence Sets

The Sequence Set Total Number parameter specifies the total number of sequence sets that will be available. The parameter also specifies the length of the settable sequence set address.

- If the Sequence Set Total Number parameter is set to two:
  Bit 0 of the binary sequence set index number (see Table 19) can be set. When the bit is set to 0, sequence set 0 will be selected and when the bit is set to 1, sequence set 1 will be selected (see Table 19).

  You can use the states of either input line 1 or input line 2 to set bit 0.

  - If the input line is not set for invert the high state of the input line will set bit 0 to 1 and the low state will set bit 0 to 0.

  - If the input line is set for invert the low state of the input line will set bit 0 to 1 and the high state will set bit 0 to 0.

- If the Sequence Set Total Number parameter is set to higher than two:
  Bits 1 and 0 of the binary sequence set index number (see Table 19) can be set. When e.g. bit 0 is set to 0 and bit 1 is set to 1, sequence set 2 will be selected (see Table 19)

  You can use the states of input line 1 to set bit 0 and the states of input line 2 to set bit 1 or vice versa.

  - If the input lines are not set for invert the high states of the input lines will set the bits to 1 and the low states will set the bits to 0.

  - If the input lines are set for invert the low states of the input lines will set the bits to 1 and the high states will set the bits to 0.

For Information about setting an input line for invert, see Section 8.1.3 on .

12.5.3.2 Configuration

Configuring Sequence Sets and Sequence Set Advance

Use the following procedure for populating sequence sets and setting the source for sequence set advance:

1. Make sure that the sequencer feature is disabled.
2. Set the Sequence Advance Mode parameter to Free Selection.
3. Set the Sequence Set Total Number parameter. The maximum number is 4.
4. Select the sequence set address bits and set the input lines that will act as the control sources:
   ▪ If the Set Total Number parameter was set to two:
      a. Bit 0 will be selected by default as the sequence set address bit. Set input line 1 or input line 2 as the control source for setting bit 0.
   ▪ If the Set Total Number parameter was set to higher than two:
      a. Select bit 0 of the sequence set address bit.
      b. Set input line 1 or input line 2 as the control source for setting bit 0.
      c. Select bit 1 of the sequence set address bit.
      d. Set the input line as the control source for setting bit 1: Chose the input line not used for setting bit 0.
5. Use the Sequence Set Index parameter to select a sequence set index number for the sequence set currently being populated. The available numbers are 0 through 3.
6. Set up your first acquisition scenario (i.e., lighting, object positioning, etc.)
7. Adjust the camera parameters to get the best image quality with this scenario (you are adjusting the parameters in the active set).
8. Execute the Sequence Set Store command to copy the sequence parameter values currently in the active set into the selected sequence set. (Any existing parameter values in the sequence set will be overwritten.)
9. Repeat the above steps for the other sequence sets, starting from step 5.

Configuring Sequence Sets and Sequence Set Advance Using Basler pylon

You can use the pylon API to set the parameters for configuring sequence sets from within your application software and make settings for their selection when images are acquired.

The following code snippet gives example settings. It illustrates using the API to set the free selection sequence advance mode with line 1 as the control source for bit 0 and line 2 as the control source for bit 1 of the sequence set address, set the total number of sequence sets to 3, and populate sequence sets 0 through 2 by storing the sequence parameter values from the active set in the sequence sets:

```
// Disable the sequencer feature
Camera.SequenceEnable.SetValue( false );
```

```
// Set the Free Selection sequence advance mode
Camera.SequenceAdvanceMode.SetValue( SequenceAdvanceMode_FreeSelection );


// Set the total number of sequence sets
Camera.SequenceSetTotalNumber.SetValue( 3 );


// Set line 1 as the control source for setting sequence set address bit 0
Camera.SequenceAddressBitSelector.SetValue( SequenceAddressBitSelector_Bit0 );
Camera.SequenceAddressBitSource.SetValue( SequenceAddressBitSource_Line1 );


// Set line 2 as the control source for setting sequence set address bit 1
Camera.SequenceAddressBitSelector.SetValue( SequenceAddressBitSelector_Bit1 );
Camera.SequenceAddressBitSource.SetValue( SequenceAddressBitSource_Line2 );


// Select sequence set with index number 0
Camera.SequenceSetIndex.SetValue( 0 );


// Set up the first acquisition scenario (lighting, object position, etc.) and
// adjust the camera parameters for the best image quality.


// Store the sequence parameter values from the active set in the selected
// sequence set
Camera.SequenceSetStore.Execute( );


// Select sequence set with index number 1
Camera.SequenceSetIndex.SetValue( 1 );


// Set up the second acquisition scenario (lighting, object position, etc.) and
// adjust the camera parameters for the best image quality.


// Store the sequence parameter values from the active set in the selected
// sequence set
Camera.SequenceSetStore.Execute( );


// Select sequence set with index number 2
Camera.SequenceSetIndex.SetValue( 2 );


// Set up the third acquisition scenario (lighting, object position, etc.) and
// adjust the camera parameters for the best image quality.


// Store the sequence parameter values from the active set in the selected
// sequence set
Camera.SequenceSetStore.Execute( );
```

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

12.6   Binning (All Models Except scA750-60)

	■ The binning feature is only available on the monochrome cameras.
(i)	■ The binning feature is not available on scA750-60 cameras.

Binning increases the camera's response to light by summing the charges from adjacent pixels into one pixel. Two types of binning are available: vertical binning and horizontal binning.

With vertical binning, adjacent pixels from 2 lines, 3 lines, or a maximum of 4 lines in the imaging sensor array are summed and are reported out of the camera as a single pixel. Figure 80 illustrates vertical binning.

Vertical Binning by 2   **Vertical Binning by 3**   **Vertical Binning by 4**
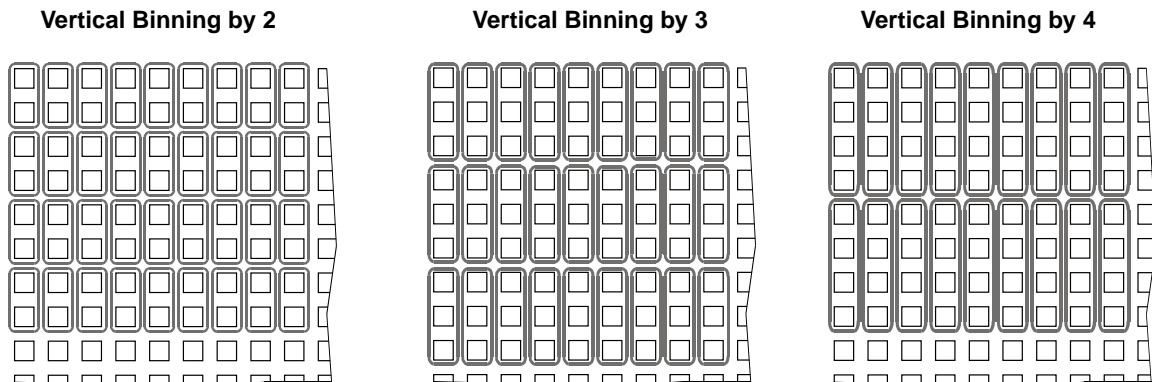
Fig. 80: Vertical Binning

With horizontal binning, adjacent pixels from 2 columns, 3 columns, or a maximum of 4 columns are summed and are reported out of the camera as a single pixel. Figure 81 illustrates horizontal binning.

Horizontal Binning by 2   **Horizontal Binning by 3**   **Horizontal Binning by 4**
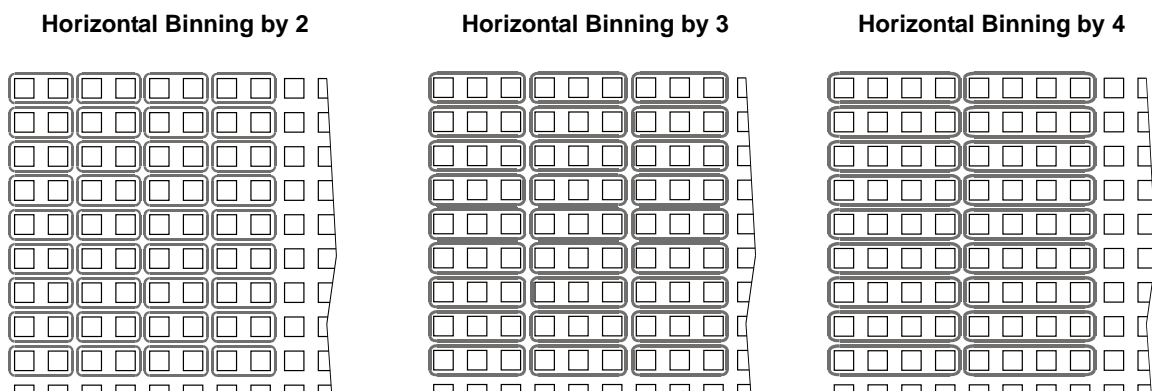
Fig. 81: Horizontal Binning

You can combine vertical and horizontal binning. This, however, may cause objects to appear distorted in the image. For more information on possible image distortion due to combined vertical and horizontal binning, see below.

Setting Binning

You can enable vertical binning by setting the Binning Vertical parameter. Setting the parameter's value to 2, 3, or 4 enables vertical binning by 2, vertical binning by 3, or vertical binning by 4 respectively. Setting the parameter's value to 1 disables vertical binning.

You can enable horizontal binning by setting the Binning Horizontal parameter. Setting the parameter's value to 2, 3, or 4 enables horizontal binning by 2, horizontal binning by 3, or horizontal binning by 4 respectively. Setting the parameter's value to 1 disables horizontal binning.

You can set the Binning Vertical or the Binning Horizontal parameter value from within your application software by using the Basler pylon API. The following code snippet illustrates using the API to set the parameter values:

```
// Enable vertical binning by 2
Camera.BinningVertical.SetValue( 2 );

// Enable horizontal binning by 4
Camera.BinningHorizontal.SetValue( 4 );

// Disable vertical and horizontal binning
Camera.BinningVertical.SetValue( 1 );
Camera.BinningHorizontal.SetValue( 1 );
```

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.

12.6.1 Considerations When Using Binning

Increased Response to Light

Using binning can greatly increase the camera's response to light. When binning is enabled, acquired images may look overexposed. If this is the case, you can reduce the lens aperture, reduce the intensity of your illumination, reduce the camera's exposure time setting, or reduce the camera's gain setting.

When using vertical binning, the limits for the minimum gain settings are automatically lowered. This allows you to use lower gain settings than would otherwise be available.

For the lowered limits for the minimum gain settings, see Section 12.1 on page 219.

Reduced Resolution

Using binning effectively reduces the resolution of the camera's imaging sensor. For example, the sensor in the scA780-54 gm camera normally has a resolution of 782 (H) x 582 (V). If you set this camera to use horizontal binning by 3 and vertical binning by 3, the effective resolution of the sensor is reduced to 260 (H) by 194 (V). (Note that the 782 pixel horizontal dimension of the sensor was not evenly divisible by 3, so we rounded down to the nearest whole number.)

Possible Image Distortion

Objects will only appear undistorted in the image if the numers of binned lines and columns are equal. With all other combinations, the imaged objects will appear distorted. If, for example, vertical binning by 2 is combined with horizontal binning by 4 the widths of the imaged objects will appear shrunk by a factor of 2 compared to the heights.

If you want to preserve the aspect ratios of imaged objects when using binning you must use vertical and horizontal binning where equal numbers of lines and columns are binned, e.g. vertical binning by 3 combined with horizontal binning by 3.

Binning's Effect on AOI Settings

When you have the camera set to use binning, keep in mind that the settings for your area of interest (AOI) will refer to the binned lines and columns in the sensor and not to the physical lines in the sensor as they normally would. Another way to think of this is by using the concept of a "virtual sensor." For example, assume that you are using a scA780-54 gm camera set for 3 by 3 binning as described above. In this case, you would act as if you were actually working with a 260 column by 194 line sensor when setting your AOI parameters. The maximum AOI width would be 260 and the maximum AOI height would be 194. When you set the X Offset and the Width for the AOI, you will be setting these values in terms of virtual sensor columns. And when you set the Y Offset and the Height for the AOI, you will be setting these values in terms of virtual sensor lines.

For more information about the area of interest (AOI) feature, see Section 12.4 on .

Binning's Effect on the Sensor Readout and Frame Rate Formulas

In several areas of the manual, formulas appear for sensor readout time and for calculating the maximum frame rate. In several of these formulas, you must enter the current height of the area of interest (AOI). If you are not using binning, you would enter the height of the AOI in physical sensor lines. If binning is enabled, however, you must use the concept of a "virtual" sensor as described above and the height of the AOI that you use in the formulas would be in terms of virtual sensor lines.

The affected formulas appear on , and.

12.7   Reverse X

The reverse X feature is a horizontal mirror image feature. When the reverse X feature is enabled, the pixel values for each line in a captured image will be swapped end-for-end about the line's center. This means that for each line, the value of the first pixel in the line will be swapped with the value of the last pixel, the value of the second pixel in the line will be swapped with the value of the next-to-last pixel, and so on.

Figure 82 shows a normal image on the left and an image captured with reverse X enabled on the right.
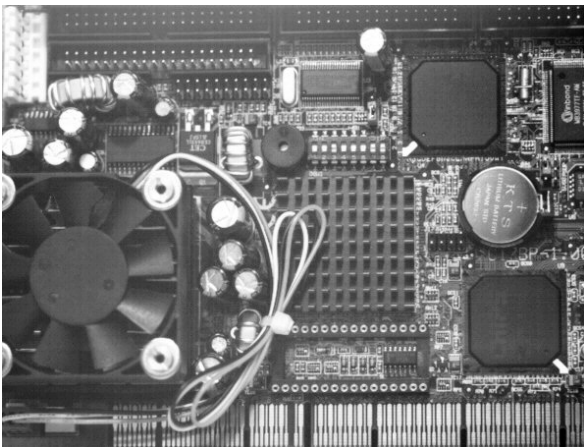
Normal Image   **Mirror Image**
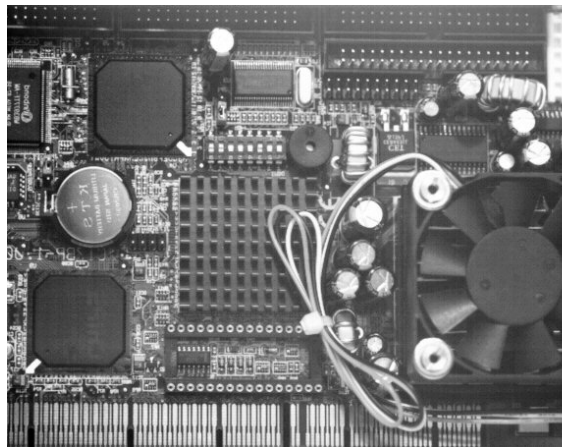
Fig. 82: Reverse X Mirror Imaging

Using AOIs with Reverse X

You can use the AOI feature when using the reverse X feature. Note, however, that the position of an AOI relative to the sensor remains the same regardless of whether or not the reverse X feature is enabled.

As a consequence, an AOI will display different images depending on whether or not the reverse X feature is enabled.

Normal Image   **Mirror Image**

AOI   AOI
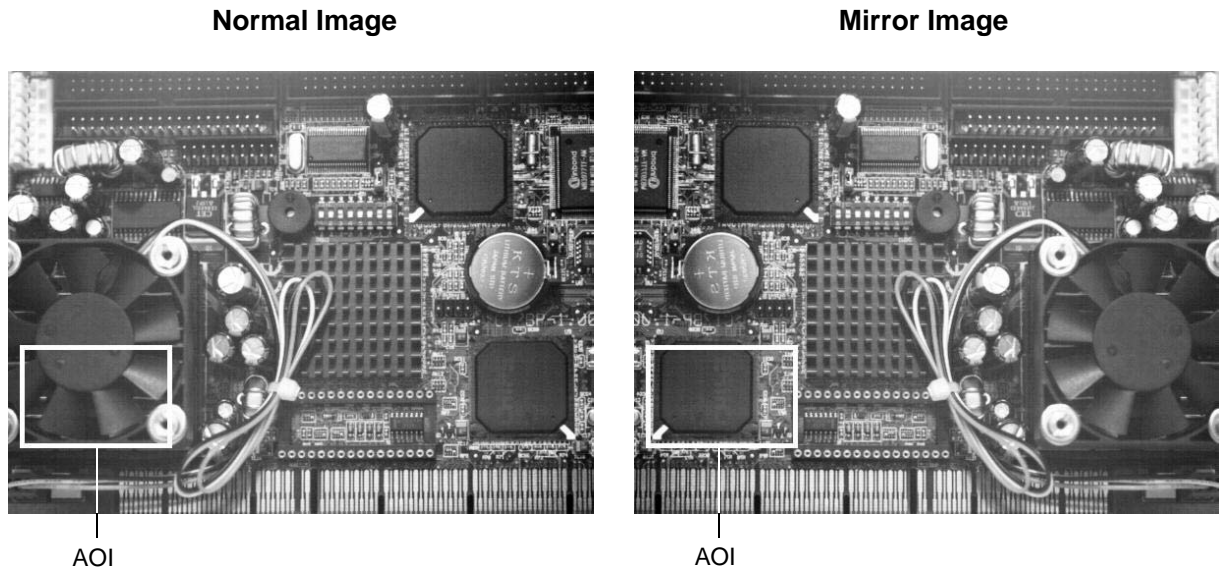
Fig. 83: Using an AOI with Reverse X Mirror Imaging

ⓘ	For color cameras, provisions are made ensuring that the effective color filter alignment will be constant for both, normal and mirror images.

ⓘ	AOIs used for the function feature will behave analogously to "standard" AOIs:
	Depending on whether or not the reverse X feature is enabled, an Image AOI will display different images and an Auto Function AOI will refer to different image contents.
	The positions of the AOIs relative to the sensor will not change.

For more information about auto functions, see Section 12.9 on .

Setting Reverse X

You can enable or disable the reverse X feature by setting the ReverseX parameter value. You can set the parameter value from within your application software by using the Basler pylon API. The following code snippet illustrates using the API to set the parameter value:

```
// Enable reverse X
Camera.ReverseX.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 parameter.

For more information about the pylon Viewer, see Section 3.1.1 on .

12.8   Luminance Lookup Table

12.8.1 Lookup Table (All Models Except scA750-60)

> (i) The information in this section applies to all camera models except the scA750-60 gm/gc. For information about scA750-60 gm/gc cameras, see Section 12.8.2 on page 280.

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.
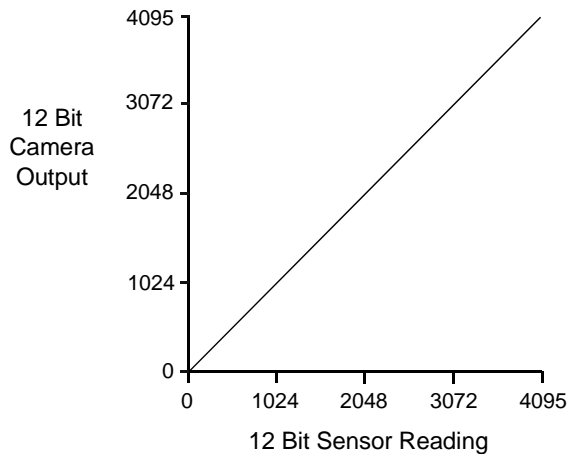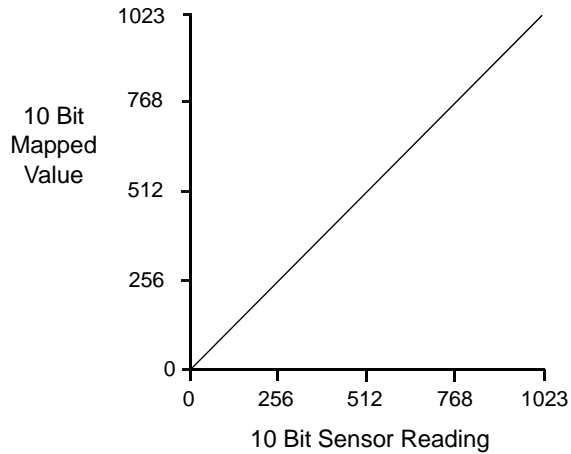

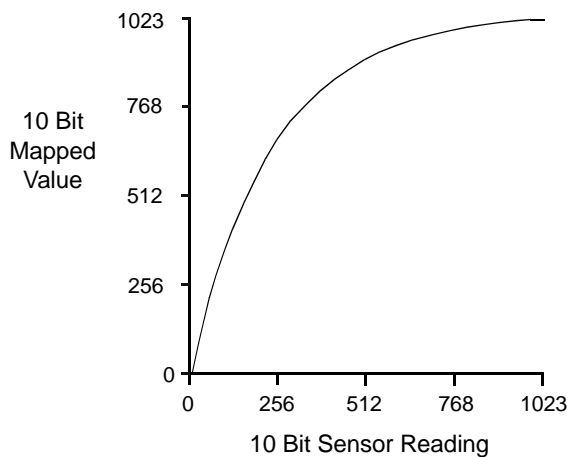Fig. 84: Lookup Table with Values Mapped in a Linear Fashion
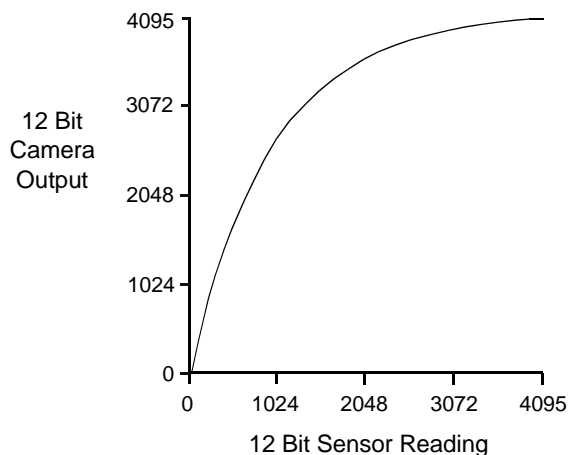
Fig. 85: Lookup Table with Values Mapped for Higher Camera Output at Low Sensor Readings

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 3.1.1 on .

12.8.2 Lookup Table (scA750-60 Only)

> ⓘ The information in this section only applies to scA750-60 gm/gc cameras. For information about the other camera models, see Section 12.8.1 on .

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. 86: Lookup Table with Values Mapped in a Linear Fashion

Fig. 87: 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 3.1.1 on .

12.9   Auto Functions

> ⓘ   The auto functions feature will not work if the sequencer feature is enabled. For
>   more information about the sequencer feature, see Section 12.5 on page 237.

12.9.1 Common Characteristics

Auto functions control image properties and are the "automatic" counterparts of certain features
such as the gain feature or the white balance feature, which normally require "manually" setting the
related parameter values. Auto functions are particularly useful when an image property must be
adjusted quickly to achieve a specific target value and when a specific target value must be kept
constant in a series of images.

An Auto Function Area of Interest (Auto Function AOI) lets you designate a specific part of the
image as the base for adjusting an image property. Each auto function uses the pixel data from an
Auto Function AOI for automatically adjusting a parameter value and, accordingly, for controlling
the related image property. Some auto functions use their own individual Auto Function AOI and
some auto functions share a single Auto Function AOI.

An auto function automatically adjusts a parameter value until the related image property reaches
a target value. Note that the manual setting of the parameter value is not preserved. For example,
when the Gain Auto function adjusts the gain parameter value, the manually set gain parameter
value is not preserved.

For some auto functions, the target value is fixed. For other auto functions, the target value can be
set, as can the limits between which the related parameter value will be automatically adjusted. For
example, the gain auto function lets you set an average gray value for the image as a target value
and also set a lower and an upper limit for the gain parameter value.

Generally, the different auto functions can operate at the same time. For more information, see the
following sections describing the individual auto functions.

> ⓘ   A target value for an image property can only be reached if it is in accord with all
>   pertinent camera settings and with the general circumstances used for capturing
>   images. Otherwise, the target value will only be approached.
>
>   For example, with a short exposure time, insufficient illumination, and a low setting
>   for the upper limit of the gain parameter value, the Gain Auto function may not be
>   able to achieve the current target average gray value setting for the image.

> You can use an auto function when binning is enabled (monochrome cameras only; not available on scA750-60 cameras). An auto function uses the binned pixel data and controls the image property of the binned image.

12.9.1.1 Modes of Operation

The following auto function modes of operation are available:

- All auto functions provide the "once" mode of operation. When the "once" mode of operation is selected, the parameter values are automatically adjusted until the related image property reaches the target value. After the automatic parameter value adjustment is complete, the auto function will automatically be set to "off" and the new parameter value will be applied to the following images.

    The parameter value can be changed by using the "once" mode of operation again, by using the "continuous" mode of operation, or by manual adjustment.

> If an auto function is set to the "once" operation mode and if the circumstances will not allow reaching a target value for an image property, the auto function will try to reach the target value for a maximum of 30 images and will then be set to "off".

- Some auto functions also provide a "continuous" mode of operation where the parameter value is adjusted repeatedly while images are acquired.

    Depending on the current frame rate, the automatic adjustments will usually be carried out for every or every other image.

    The repeated automatic adjustment will proceed until the "once" mode of operation is used or until the auto function is set to "off", in which case the parameter value resulting from the latest automatic adjustment will operate, unless the parameter is manually adjusted.

- When an auto function is set to "off", the parameter value resulting from the latest automatic adjustment will operate, unless the parameter is manually adjusted.

> You can enable auto functions and change their settings while the camera is capturing images ("on the fly").

	If you have set an auto function to "once" or "continuous" operation mode while the camera was continuously capturing images, the auto function will become effective with a short delay and the first few images may not be affected by the auto function.

12.9.1.2 Auto Function AOI

An Auto Function AOI must be set separately from the AOI used to define the size of captured images (the image AOI). You can specify a portion of the sensor array and only the pixel data from the specified portion will be used for auto function control.

An Auto Function AOI is referenced to the top left corner of the sensor array. The top left corner is designated as column 0 and row 0 as shown in Figure 88.

The location and size of an Auto Function AOI 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 14, the width as 5, the Y offset as 7, and the height as 6. The area of the array that is bounded by these settings is shown in Figure 88.

Only the pixel data from within the area defined by your settings will be used by the related auto function.
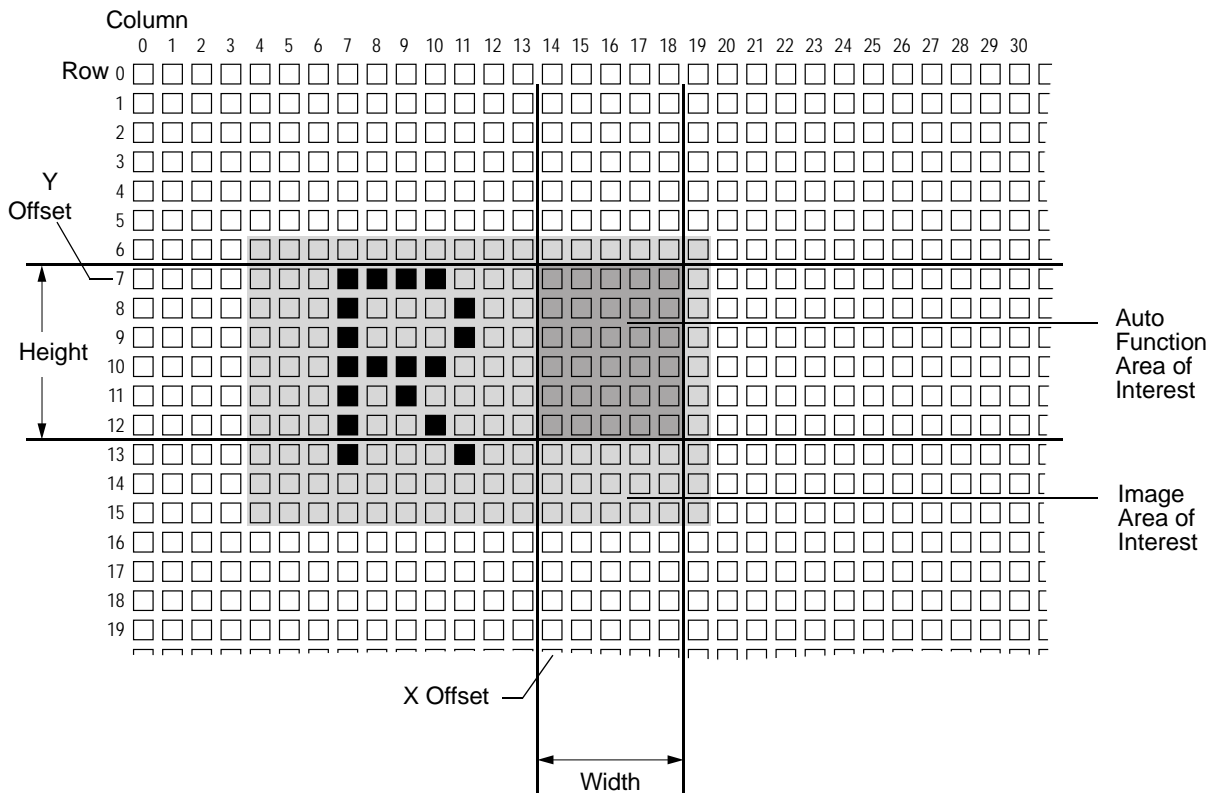
Fig. 88: Auto Function Area of Interest and Image Area of Interest

Relative Positioning of an Auto Function AOI

The size and position of an Auto Function AOI can be, but need not be, identical to the size and position of the Image AOI. Note that the overlap between Auto Function AOI and Image AOI determines whether and to what extent the auto function will control the related image property. Only the pixel data from the areas of overlap will be used by the auto function to control the image property of the entire image.

Different degrees of overlap are illustrated in Figure 89. The hatched areas in the figure indicate areas of overlap.

■ If the Auto Function AOI is completely included in the Image AOI (see (a) in Figure 89), the pixel data from the Auto Function AOI will be used to control the image property.

■ If the Image AOI is completely included in the Auto Function AOI (see (b) in Figure 89), only the pixel data from the Image AOI will be used to control the image property.

■ If the Image AOI only partially overlaps the Auto Function AOI (see (c) in Figure 89), only the pixel data from the area of partial overlap will be used to control the image property.

■ If the Auto Function AOI does not overlap the Image AOI (see (d) in Figure 89), the Auto Function will **not** or only to a **limited** degree control the image property. For details, see the sections below, describing the individual auto functions.

(i)	We strongly recommend completely including the Auto Function AOI within the Image AOI, or, depending on your needs, choosing identical positions and sizes for Auto Function AOI and Image AOI.

(i)	You can use auto functions when also using the reverse X feature. For information about the behavior and roles of Auto Function AOI and Image AOI when also using the reverse X feature, see the "Reverse X" section.
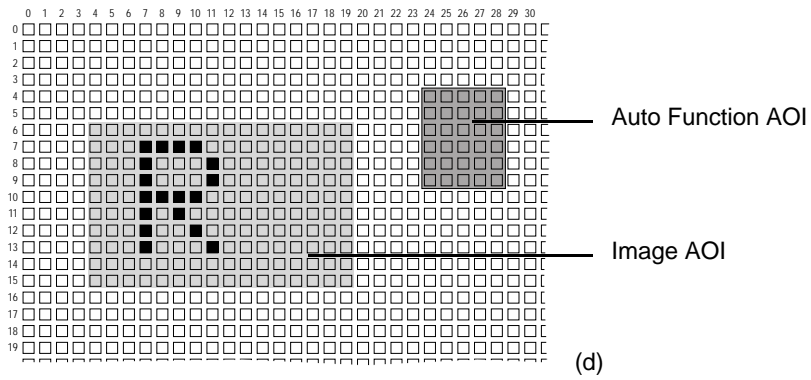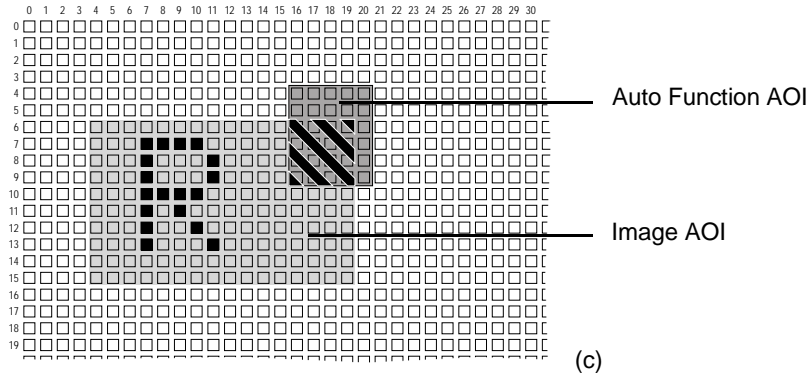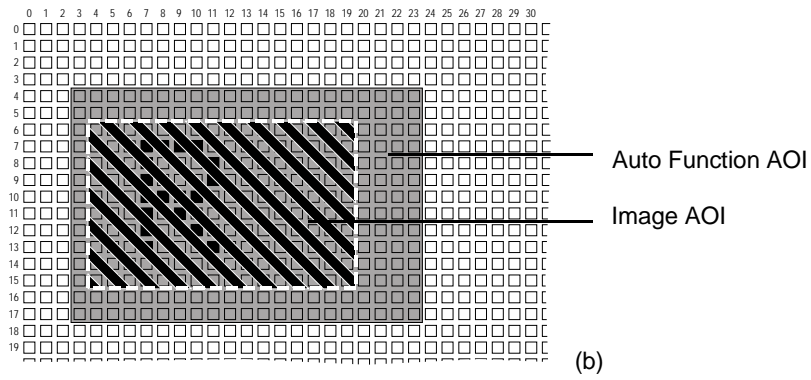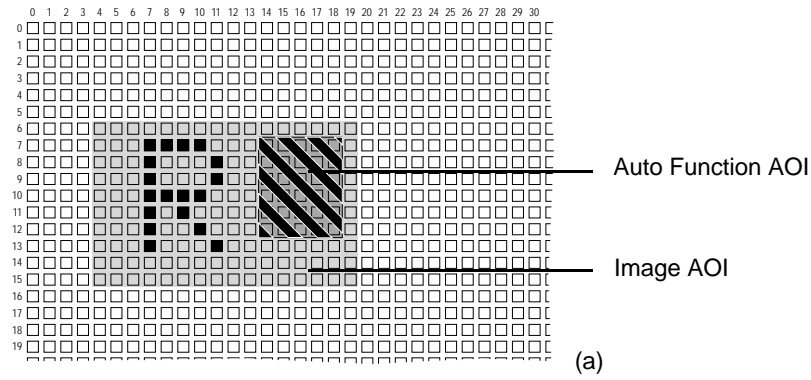
Fig. 89: Various Degrees of Overlap Between the Auto Function AOI and the Image AOI

Setting an Auto Function AOI

Setting an Auto Function AOI is a two-step process: You must first select the Auto Function AOI related to the auto function that you want to use and then set the size and the position of the Auto Function AOI.

By default, an Auto Function AOI is set to the full resolution of the camera's sensor. You can change the size and the position of an Auto Function AOI by changing the value of the Auto Function AOI's X Offset, Y Offset, Width, and Height parameters.

- The value of the X Offset parameter determines the starting column for the Auto Function AOI.
- The value of the Y Offset parameter determines the starting line for the Auto Function AOI.
- The value of the Width parameter determines the width of the Auto Function AOI.
- The value of the Height parameter determines the height of the Auto Function AOI.

When you are setting an Auto Function AOI, you must follow these guidelines:

- The sum of the X Offset setting plus the Width setting must not exceed the width of the camera's sensor. For example, on the scA640-70, the sum of the X Offset setting plus the Width setting must not exceed 659.
- The sum of the Y Offset setting plus the Height setting must not exceed the height of the camera's sensor. For example, on the scA640-70, the sum of the Y Offset setting plus the Height setting must not exceed 494.

The X Offset, Y Offset, Width, and Height parameters can be set in increments of 1.

	On color cameras, we strongly recommend setting the X Offset, Y Offset, Width, and Height parameters for an Auto Function AOI in increments of 2 to make the Auto Function AOI match the Bayer filter pattern of the sensor. For example, you should set the X Offset parameter to 0, 2, 4, 6, 8, etc.

	Normally, the X Offset, Y Offset, Width, and Height parameter settings for an Auto Function AOI refer to the physical columns and lines in the sensor. But if binning is enabled (monochrome cameras only; not available on scA750-60 cameras), these parameters are set in terms of "virtual" columns and lines, i.e. the settings for an Auto Function AOI will refer to the binned lines and columns in the sensor and not to the physical lines in the sensor as they normally would.

For more information about the concept of a "virtual sensor", see Section 12.6.1 on .

You can select an Auto Function AOI and set the X Offset, Y Offset, Width, and Height parameter values for the Auto Function AOI from within your application software by using the Basler pylon API. The following code snippets illustrate using the API to select an Auto Function AOI and to get the maximum allowed settings for the Width and Height parameters. The code snippets also illustrate setting the X Offset, Y Offset, Width, and Height parameter values. As an example, Auto Function AOI1 is selected:

```
// Select the appropriate auto function AOI for gain auto and exposure auto
// control. Currently auto function AOI 1 is predefined to gather the pixel
// data needed for gain auto and exposure auto control
// Set the position and size of the auto function AOI
Camera.AutoFunctionAOISelector.SetValue( AutoFunctionAOISelector_AOI1 );
Camera.AutoFunctionAOIOffsetX.SetValue( 0 );
Camera.AutoFunctionAOIOffsetY.SetValue( 0 );
Camera.AutoFunctionAOIWidth.SetValue( Camera.AutoFunctionAOIWidth.GetMax() );
Camera.AutoFunctionAOIHeight.SetValue( Camera.AutoFunctionAOIHeight.GetMax() );
```

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.

12.9.1.3 Using an Auto Function

To use an auto function, carry out the following steps:

1. Select an Auto Function AOI.
2. Assign the auto function you want to use to the selected Auto Function AOI.
3. Unassign the auto function you want to use from the other Auto Function AOI.
4. Set the position and size of the Auto Function AOI.
5. If necessary, set the lower and upper limits for the auto functions's parameter value.
6. If necessary, set the target value.
7. If necessary, set the auto function profile to define priorities between auto functions.
8. Enable the auto function by setting it to "once" or "continuous".

For more information about the individual settings, see the next sections that describe the individual auto functions.

12.9.2 Gain Auto

Gain Auto is the "automatic" counterpart of the manual gain feature. When the gain auto function is operational, the Gain Raw (All) parameter value is automatically adjusted within set limits, until a target average gray value for the pixel data from Auto Function AOI1 is reached.

The gain auto function uses Auto Function AOI1 and can be operated in the "once" and continuous" modes of operation.

If Auto Function AOI1 does not overlap the Image AOI (see the "Auto Function AOI" section) the pixel data from Auto Function AOI1 will not be used to control the image brightness. Instead, the current manual setting of the Gain Raw (All) parameter value will control the image brightness.

When the gain auto function is used, the exposure auto function can be used at the same time. In this case, however, you must also set the auto function profile feature.

For more information about gain, see Section 12.1 on page 219.

For more information about the auto function profile feature, see Section 12.9.4 on page 294.

To use the gain auto function, perform the following steps:

1. Select Auto Function AOI1.
2. Set the position and size of Auto Function AOI1.
3. Set the lower and upper limits for the Gain Raw (All) parameter value.
4. Set the target average gray value.
5. If necessary, set the auto function profile.
6. Enable the gain auto function by setting it to "once" or "continuous". You must choose the "continuous" setting when using the auto function profile.

The currently settable limits for the Auto Gain Raw parameter value depend on the current pixel data format, on the current settings for binning, and on whether or not the Gain Raw parameter limits for the manually set gain feature are disabled.

The target average gray value may range from 50 to 205. Note that this parameter value range applies to 8 bit **and** to 16 bit (12 bit effective) output modes. Note also that a given parameter value setting corresponds to an identical gray value regardless of the output mode.

You can carry out steps 1 to 6 from within your application software by using the pylon API. The following code snippets illustrate using the API to set the parameter values:

■ Selecting and setting Auto Function AOI1

■ Setting the limits for the Auto Gain Raw parameter value. The currently accessible minimum and maximum parameter values are chosen as examples

■ Setting the target average gray value. A medium gray value is chosen as an example

■ Enabling the gain auto function and selecting, for example, the "once" mode of operation

```
// Select the appropriate auto function AOI for luminance statistics
// Currently AutoFunctionAOISelector_AOI1 is predefined to gather
// luminance statistics
// Set position and size of the auto function AOI
Camera.AutoFunctionAOISelector.SetValue( AutoFunctionAOISelector_AOI1 );
Camera.AutoFunctionAOIOffsetX.SetValue( 0 );
Camera.AutoFunctionAOIOffsetY.SetValue( 0 );
Camera.AutoFunctionAOIWidth.SetValue( Camera.AutoFunctionAOIWidth.GetMax() );
Camera.AutoFunctionAOIHeight.SetValue( Camera.AutoFunctionAOIHeight.GetMax() );

// Select gain for automatic luminance control.
// Set gain limits for luminance control
Camera.GainSelector.SetValue( GainSelector_All );
Camera.AutoGainRawLowerLimit.SetValue( Camera.GainRaw.GetMin() );
Camera.AutoGainRawUpperLimit.SetValue( Camera.GainRaw.GetMax() );

// Set target value for luminance control. This is always expressed
// by an 8 bit value, regardless of the current pixel format
// Settable range: 50 to 205
Camera.AutoTargetValue.SetValue( 128 );

// Set mode of operation for gain auto function
Camera.GainAuto.SetValue( GainAuto_Once );
```

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 general information about auto functions, see Section 12.9 on .

For information about Auto Function AOIs and how to set them, see Section 12.9.1.2 on .

12.9.3 Exposure Auto

Exposure Auto is an auto function and the "automatic" counterpart to manually setting an "absolute" exposure time. The exposure auto function automatically adjusts the Exposure Time Abs parameter value within set limits, until a target average gray value for the pixel data from Auto Function AOI1 is reached.

In contrast to the manually set "absolute" exposure time, the automatically adjusted "absolute" exposure time and the settable limits for parameter value adjustment are not restricted to multiples of the current exposure time base.

The exposure auto function uses Auto Function AOI1 and can be operated in the "once" and continuous" modes of operation.

If Auto Function AOI1 does not overlap the Image AOI (see the "Auto Function AOI" section) the pixel data from Auto Function AOI1 will not be used to control the image brightness. Instead, the current manual setting of the Exposure Time Abs parameter value will control the image brightness.

The exposure auto function is not available, when trigger width exposure mode is selected.

When the exposure auto function is used, the gain auto function can be used at the same time. In this case, however, you must also set the auto function profile feature.

(i)	If the Auto Exposure Time Abs Upper Limit parameter is set to a sufficiently high value the camera's frame rate may be decreased.

For more information about "absolute" exposure time settings and related limitations, see Section 9.7.2 on page 139.

For more information about exposure modes and how to select them, see Section 9.5.2 on page 118, Section 9.5.3 on page 120, Section 9.6.2 on page 129, and Section 9.6.3 on page 131.

For more information about the auto function profile feature, see Section 12.9.4 on page 294.

To use the exposure auto function, carry out the following steps:

1. Make sure trigger width exposure mode is not selected.
2. Select Auto Function AOI1.
3. Set the position and size of Auto Function AOI1.
4. Set the lower and upper limits for the Exposure Time Abs parameter value.
5. Set the target average gray value.
6. If necessary, set the auto function profile.
7. Enable the exposure auto function by setting it to "once" or "continuous". You must choose the "continuous" setting when using the auto function profile.

The settable limits for the Exposure Time Abs parameter value are limited by the minimum allowed and maximum possible exposure time of the camera model.

The target average gray value may range from 50 to 205. Note that this parameter value range applies to 8 bit **and** to 16 bit (12 bit effective) output modes. Note also that a given parameter value setting corresponds to an identical gray value regardless of the output mode.

You can carry out steps 1 to 7 from within your application software by using the pylon API. The following code snippets illustrate using the API to set the parameter values:

- Selecting and setting Auto Function AOI1: See the "Auto Function AOI" section above.
- Setting the limits for the Exposure Time Abs parameter value (the set parameter values serve as examples):
- Setting the target average gray value. A medium gray value is selected as an example:
- Enabling the exposure auto function and selecting, for example, the "continuous" mode of operation:

```
// Select the appropriate auto function AOI for luminance statistics
// Currently AutoFunctionAOISelector_AOI1 is predefined to gather
// luminance statistics
// Set position and size of the auto function AOI
Camera.AutoFunctionAOISelector.SetValue( AutoFunctionAOISelector_AOI1 );
Camera.AutoFunctionAOIOffsetX.SetValue( 0 );
Camera.AutoFunctionAOIOffsetY.SetValue( 0 );
Camera.AutoFunctionAOIWidth.SetValue( Camera.AutoFunctionAOIWidth.GetMax() );
Camera.AutoFunctionAOIHeight.SetValue( Camera.AutoFunctionAOIHeight.GetMax() );

// Set exposure time limits for luminance control
Camera.AutoExposureTimeAbsLowerLimit.SetValue( 1000 );
Camera.AutoExposureTimeAbsUpperLimit.SetValue( 1.0E6 );

// Set target value for luminance control. This is always expressed
// by an 8 bit value, regardless of the current pixel format
// Settable range: 50 to 205
Camera.AutoTargetValue.SetValue( 128 );

// Set mode of operation for exposure auto function
Camera.ExposureAuto.SetValue( ExposureAuto_Continuous );
```

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 general information about auto functions, see Section 12.9 on .

For information about Auto Function AOIs and how to set them, see Section 12.9.1.2 on .

For information about minimum allowed and maximum possible exposure time, see Table 11 in Section 9.7 on .

12.9.4 Auto Function Profile

If you want to use the gain auto function and the exposure auto function at the same time, you must also set the auto function profile. The auto function profile specifies whether the gain or the exposure time will be kept as low as possible when the system is making automatic adjustments as it attempts to reach the target average gray value.

To use the gain auto function and the exposure auto function at the same time, carry out the following steps:

1. Set the auto function profile to specify whether gain or exposure time will be minimized during adjustments.
2. Set the gain auto function to the "continuous" mode of operation.
3. Set the exposure auto function to the "continuous" mode of operation.

You can set the auto function profile from within your application software by using the Basler pylon API. The following code snippets illustrate using the API to set the auto function profile. As an example, Gain Auto is set to be minimized during adjustments:

```
// Use the gain auto and exposure auto features simultaneously
Camera.AutoFunctionProfile.SetValue( AutoFunctionProfile_GainMinimum );
Camera.GainAuto.SetValue( GainAuto_Continuous );
Camera.ExposureAuto.SetValue( ExposureAuto_Continuous );
```

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.

12.9.5 Balance White Auto

Balance White Auto is the "automatic" counterpart to manually setting the white balance. The balance white auto function is only available on color models.

Automatic white balancing is a two-step process. First, the Balance Ratio Abs parameter values for red, green, and blue are each set to 1.5. Then, assuming a "gray world" model, the Balance Ratio Abs parameter values are automatically adjusted such that the average values for the "red" and "blue" pixels match the average value for the "green" pixels.

The balance white auto function uses Auto Function AOI 2 and can only be operated in the "once" mode of operation.

If Auto Function AOI 2 does not overlap the Image AOI (see the "Auto Function AOI" section) the pixel data from Auto Function AOI 2 will not be used to control the white balance of the image. However, as soon as the Balance White Auto function is set to "once" operation mode, the Balance Ratio Abs parameter values for red, green, and blue are each set to 1.5. These settings will control the white balance of the image.

For information on the white balance feature, see Section 10.3.1 on page 177.

To use the balance white auto function, carry out the following steps:

1. Select Auto Function AOI2.
2. Set the position and size of Auto Function AOI2.
3. Enable the balance white auto function by setting it to "once".

You can carry out steps 1 to 3 from within your application software by using the Basler pylon API. The following code snippet illustrates using the API to use the auto function:

- Selecting and setting Auto Function AOI2: See the "Auto Function AOI" section above.
- Enabling the balance white auto function and selecting the "once" mode of operation:

```
// Select the AOI to use for auto white balancing
// Currently AOI2 is predefined to gather the pixel data needed for
// automatic white balancing
// Set position and size of the auto function AOI
Camera.AutoFunctionAOISelector.SetValue( AutoFunctionAOISelector_AOI2 );
Camera.AutoFunctionAOIOffsetX.SetValue( 0 );
Camera.AutoFunctionAOIOffsetY.SetValue( 0 );
Camera.AutoFunctionAOIWidth.SetValue( Camera.AutoFunctionAOIWidth.GetMax() );
Camera.AutoFunctionAOIHeight.SetValue( Camera.AutoFunctionAOIHeight.GetMax() );

// Set the mode of operation for balance white auto function
Camera.BalanceWhiteAuto.SetValue( BalanceWhiteAuto_Once );
```

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 general information about auto functions, see Section 12.9 on page 283.

For information about Auto Function AOIs and how to set them, see Section 12.9.1.2 on page 285.

12.10 Disable Parameter Limits

For each camera parameter, the allowed range of parameter values normally is limited. The factory limits are designed to ensure optimum camera operation and, in particular, good image quality. For special camera uses, however, it may be helpful to set parameter values outside of the factory limits.

The disable parameter limits feature lets you disable the factory parameter limits for certain parameters. When the factory parameter limits are disabled, the parameter values can be set within extended limits. Typically, the range of the extended limits is dictated by the physical restrictions of the camera's electronic devices, such as the absolute limits of the camera's variable gain control.

The values for the extended limits can be seen using the Basler pylon Viewer or from within your application via the pylon API.

(i)	Currently, the parameter limits can only be disabled on the Gain feature.

Disabling Parameter Limits

To disable the limits for a parameter:

■   Use the Parameter Selector to select the parameter whose limits you wish to disable.

■   Set the value of the Remove Limits parameter.

You can set the Parameter Selector and the value of the Remove Limits 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 value:

```
// Select the feature whose factory limits will be disabled
Camera.ParameterSelector.SetValue( ParameterSelector_Gain );

// Disable the limits for the selected feature
Camera.RemoveLimits.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. Note that the disable parameter limits feature will only be available at the "guru" viewing level.

For more information about the pylon Viewer, see Section 3.1.1 on page 43.

12.11 Debouncer

The debouncer feature aids in discriminating between valid and invalid input signals and only lets valid signals pass to the camera. The debouncer value specifies the minimum time that an input signal must remain high or remain low in order to be considered a valid input signal.

	We recommend setting the debouncer value so that it is slightly greater than the longest expected duration of an invalid signal. Setting the debouncer to a value that is too short will result in accepting invalid signals. Setting the debouncer to a value that is too long will result in rejecting valid signals.

Note that the debouncer delays a valid signal between its arrival at the camera and its transfer. The duration of the delay will be determined by the debouncer value.

Figure 90 illustrates how the debouncer filters out invalid input signals, i.e. signals that are shorter than the debouncer value. The diagram also illustrates how the debouncer delays a valid signal.
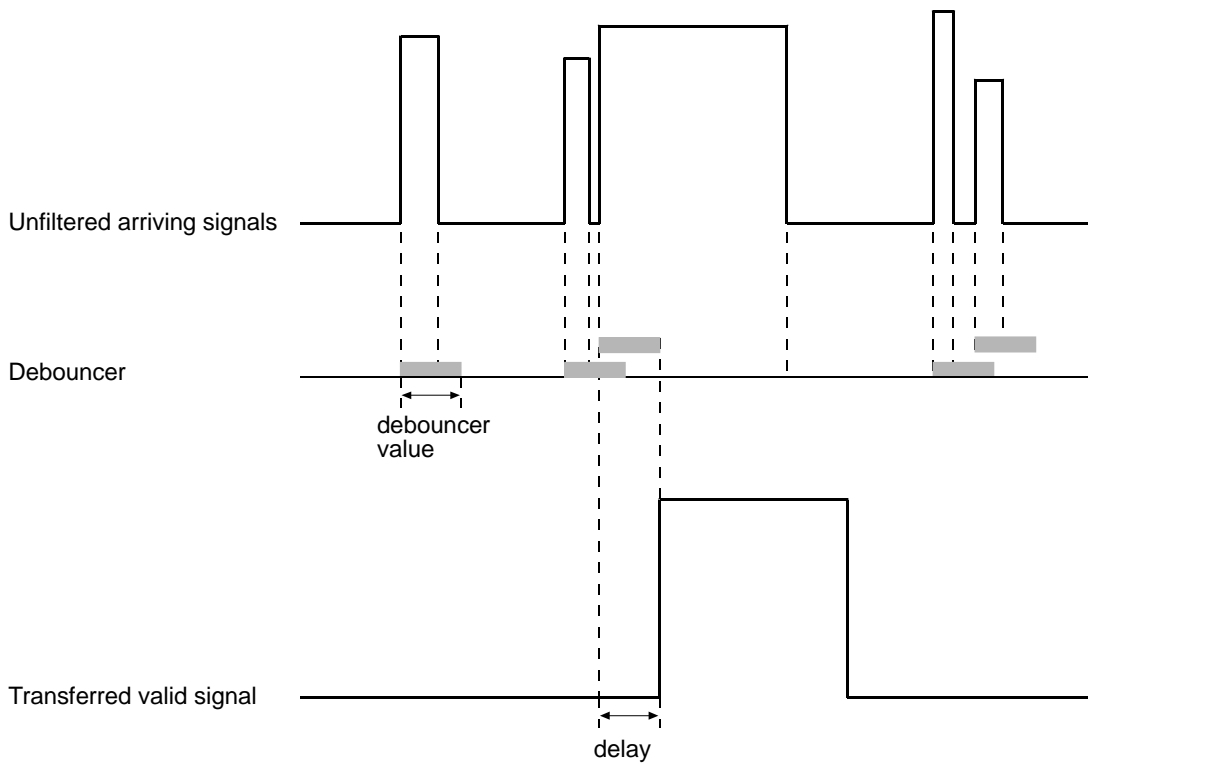
Fig. 90: Filtering of Input Signals by the Debouncer

Setting the Debouncer

The debouncer value is determined by the value of the Line Debouncer Time Abs parameter value. The parameter is set in microseconds and can be set in a range from 0 to approximately 1 s.

To set a debouncer:

■   Use the Line Selector to select the camera input line for which you want to set the debouncer (input line1 or 2).

■   Set the value of the Line Debouncer Time Abs parameter.

You can set the Line Selector and the value of the Line Debouncer Abs 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 value:

```
// Select the input line
Camera.LineSelector.SetValue( LineSelector_Line1 );


// Set the parameter value to 100 microseconds
Camera.LineDebouncerTimeAbs.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 3.1.1 on page 43.

12.12 Minimum Output Pulse Width

An output signal sent by the camera may be too narrow for some receivers to be detected. To ensure reliable detection, the Minimum Output Pulse Width feature allows you to increase the signal width to a set minimum width:

■ If the signal width of the original output signal is narrower than the set minimum the Minimum Output Pulse Width feature will increase the signal width to the set minimum before the signal is sent out of the camera (see the figure below).

■ If the signal width of the original output signal is equal to or wider than the set minimum the Minimum Output Pulse Width feature will have no effect. The signal will be sent out of the camera with unmodified signal width.
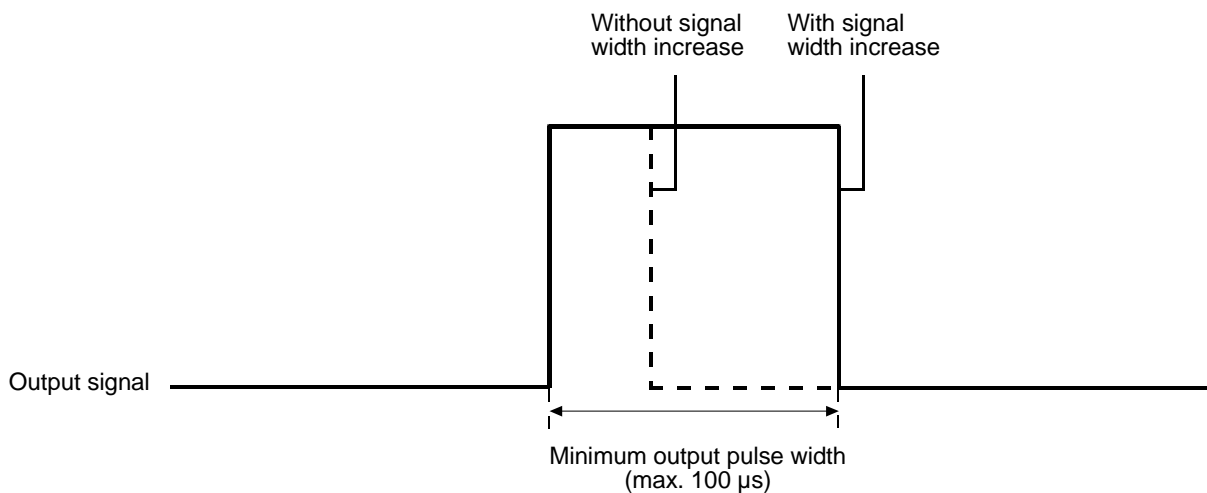
Fig. 91: Increasing the Signal Width of an Output Signal

Setting the Minimum Output Pulse Width

The minimum output pulse width is determined by the value of the MinOutPulseWidthAbs parameter. The parameter is set in microseconds and can be set in a range from 0 to 100 µs.

To set the minimum output pulse width parameter value:

■ Use the Line Selector to select the camera output line for which you want to set the minimum output pulse width parameter value (output line 1, 2, 3 or 4).

■ Set the value of the MinOutPulseWidthAbs parameter.

You can set the Line Selector and the value of the MinOutPulseWidthAbs 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 value:

```
// Select the input line
Camera.LineSelector.SetValue(LineSelector_Out1);


// Set the parameter value to 10.0 microseconds
Camera.MinOutPulseWidthAbs.SetValue(10.0);
```

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.

12.13 Trigger Delay

The trigger delay feature lets you specify a delay (in microseconds) that will be applied between the receipt of a hardware trigger and it becoming effective.

The trigger delay can be specified in the range from 0 to 10000000 μs (equivalent to 10 s). When the delay is set to 0 μs, no delay will be applied.

Note the different applicability of the trigger delay depending on the image acquisition control mode:

▪ Standard mode:

  ▪ The trigger delay can be applied to the acquisition start trigger.

> (i) The trigger delay will not operate if the Acquisition Start Trigger Mode parameter is set to off (the camera will generate all acquisition start trigger signals internally) or if you are using a software acquisition start trigger.

  ▪ The trigger delay can be applied to the frame start trigger.

> (i) The trigger delay will not operate if the Frame Start Trigger Mode parameter is set to off (the camera will generate all frame start trigger signals internally) or if you are using a software frame start trigger.

▪ Legacy mode:

  ▪ The trigger delay can be applied to the acquisition start trigger.

> (i) The trigger delay will not operate if the Acquisition Start Trigger Mode parameter is set to off (the camera will generate all acquisitiion start trigger signals internally) or if you are using a software acquisition start trigger.

Setting the Trigger Delay

You can set the Trigger Delay Abs parameter value from within your application software by using the pylon API. The following code snippets illustrate using the API to set the parameter values:

Standard mode:

```
// Select the acquisition start trigger
Camera.TriggerSelector.SetValue( TriggerSelector_Acquisition Start );
// Trigger delay
double TriggerDelay_us = 1000.0   // 1000us == 1ms == 0.001s;
Camera.TriggerDelayAbs.SetValue( TriggerDelay_us );

// Select the frame start trigger
```

```
Camera.TriggerSelector.SetValue( TriggerSelector_FrameStart );
// Trigger delay
double TriggerDelay_us = 1000.0   // 1000us == 1ms == 0.001s;
Camera.TriggerDelayAbs.SetValue( TriggerDelay_us );
```

- Legacy mode:

```
// Select the acquisition start trigger
Camera.TriggerSelector.SetValue( TriggerSelector_AcquisitionStart );
// Trigger delay
double TriggerDelay_us = 1000.0   // 1000us == 1ms == 0.001s;
Camera.TriggerDelayAbs.SetValue( TriggerDelay_us );
```

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 3.1.1 on .

For more information about the standard and legacy image acquisition control modes, see Section 9.1 on .

12.14 Acquisition Status

When controlling image acquisition with a software trigger you can use the acquisition status feature to determine when the camera is ready to be triggered for an image acquisition. Using this feature, you can avoid triggering the camera at a rate that exceeds the maximum allowed with the current camera settings.

For other means of checking the acquisition status, see also the "Acquisition Monitoring Signals" section.

(i)	It is not possible to monitor the status of the Acquisition Start command. Therefore, you cannot use the status of the Acquisition Start command to determine when the camera is ready to be triggered for an image acquisition.

Note the different applicability of the acquisition status feature depending on the image acquisition control mode:

- Standard mode: The acquisition status can be determined for the frame start trigger.
- Legacy mode: The acquisition status can be determined for the acquisition start trigger.

Determining the Acquisition Status

To determine the acquisition status of the camera:

- Use the Acquisition Status Selector to select the Frame Trigger Wait status. (The Frame Trigger Wait parameter also applies when operating the camera in legacy mode, where actually the status of the acquisition start trigger is determined).

- Read the value of the AcquisitionStatus parameter.

  Standard mode: If the value is set to "false", the camera is not ready to receive a frame start trigger, if the value is set to "true", the camera is ready to receive a frame start trigger.

  Legacy mode: If the value is set to "false", the camera is not ready to receive an acquisition start trigger, if the value is set to "true", the camera is ready to receive an acquisition start trigger.

You can set the Acquisition Status Selector and read the AcquisitionStatus parameter from within your application software by using the pylon API. The following code snippets illustrate using the API to set and read the parameter values:

```
// Set the Acquisition Status Selector
Camera.AcquisitionStatusSelector.SetValue( AcquisitionStatusSelector_FrameTrig-
gerWait );

// Read the acquisition status
bool IsWaitingForFrameTrigger = Camera.AcquisitionStatus.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 Acquisition Status Selector.

For more information about the pylon Viewer, see Section 3.1.1 on .

For more information about the standard and legacy image acquisition control modes, see Section 9.1 on .

12.15 Event Reporting

Event reporting is available on the camera. With event reporting, the camera can generate an "event" and after some intermediate steps transmit a related event message to the PC whenever a specific situation has occurred:

The camera can generate and transmit events for these types of situations:

- An acquisition start trigger has occurred (AcquisitionStartEventData)
- Overtriggering of the acquisition start trigger has occurred (AcquisitionStartOvertriggerEventData)
- A frame start trigger has occurred (FrameStartEventData)
- Overtriggering of the frame start trigger has occurred (FrameStartOvertriggerEventData)
- The end of an exposure has occurred (ExposureEndEventData)
- An event overrun has occurred (EventOverrunEventData); see below for more information.

An Example of Event Reporting

An example related to the Exposure End event illustrates how event reporting works. The example assumes that your system is set for event reporting (see below), and that an end of exposure has just occurred in the camera. In this case:

1. An Exposure End event is generated. The event contains the event in the strict sense and supplementary information:

   An *Event Type Identifier*. In this case, the identifier would show that an exposure end type event has occurred.

   A *Frame ID*. This number indicates the frame count at the time that the event occurred.

   A *Stream Channel Identifier*. Currently this identifier is always 0.

   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, an event message will be sent to the PC. 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 acknowledgment. If no acknowledgment is received within a specified timeout, the camera will resend the event message. If an acknowledgment is still not received, the timeout and resend mechanism will repeat until a specified maximum number of retries is reached. If the maximum number of retries is reached and no acknowledge has been received, the message will be dropped. During the time that the camera is waiting for an acknowledgment, no new event messages can be transmitted.

4. Event Reporting involves some further software-related steps and settings to be made. For more information, see the "Camera Events" code sample included with the pylon software development kit.

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 the 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

Event reporting must be enabled in the camera and some additional software-related settings must be made. This is described in the "Camera Events" code sample included with the pylon software development kit.

Event reporting must be specifically set up for each type of event using the parameter name of the event and of the supplementary information. The following table lists the relevant parameter names:

Event	Event Parameter Name	Supplementary Information Parameter Name
AcquisitionStart	AcquisitionStartEventData	AcquisitionStartEventStreamChannelIndex
		AcquisitionStartEventTimestamp
Acquisition Start Overtrigger	AcquisitionStartOvertriggerEventData	AcquisitionStartOvertriggerEventStreamChannelIndex
		AcquisitionStartOvertriggerEventTimestamp
FrameStart	FrameStartEventData	FrameStartEventStreamChannelIndex
		FrameStartEventTimestamp
Frame Start Overtrigger	FrameStartOvertriggerEventData	FrameStartOvertriggerEventStreamChannelIndex
		FrameStartOvertriggerEventTimestamp
Exposure End	ExposureEndEventData	ExposureEndEventFrameID
		ExposureEndEventStreamChannelIndex
		ExposureEndEventTimestamp
Event Overrun	EventOverrunEventData	EventOverrunEventStreamChannelIndex
		EventOverrunEventTimestamp

Table 20: Parameter Names of Events and Supplementary Information

You can enable event reporting and make the additional settings from within your application software by using the pylon API. For more information, see the "Camera Events" code sample included with the pylon software development kit.

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

12.16 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 a 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 Basler 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 3.1.1 on .

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 92.

The mathematical expression for this test image:

Gray Value = [column number + row number] MOD 256
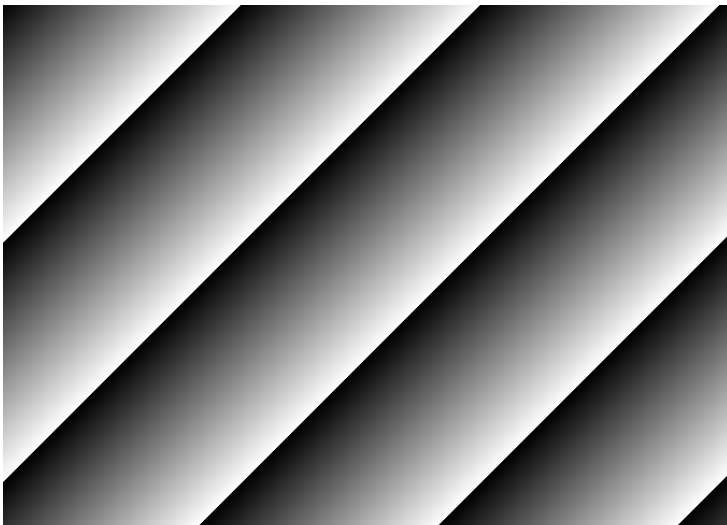
Fig. 92: Test Image One

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

> (i) 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.

> (i) 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 93, 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. 93: Test Image Six

12.17 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. This string usually indicates "none".
- 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 Configurator. 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.
- Sensor Board Temperature (read only) - Indicates the current temperature in degrees centigrade of the camera's sensor board.

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.DeviceID.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();

// Read the Temperature Abs parameter
camera.TemperatureSelector.SetValue(TemperatureSelector_Sensorboard);
double temperature = camera.TemperatureAbs.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 Configurator to read or write the Device User ID.

For more information about the pylon Viewer, see Section 3.1.1 on page 43.

For more information about the pylon IP Configurator, see Section 3.1.2 on page 44.

12.18 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 set, the default set, and user sets.

The Active Set

The active 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 direct register access, you are making changes to the active set. The active 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 Default Set

When a camera is manufactured, numerous tests are performed on the camera and four factory optimized setups are determined. The four factory optimized setups are:

Fig. 94: Configuration Sets

- The Standard Factory Setup - is optimized for average conditions and will provide good camera performance in many common applications. In the standard factory setup, the gain is set to a low value, and all auto functions are set to off.

- The High Gain Factory Setup - is similar to the standard factory setup, but the gain is set to + 6 dB.

- The Auto Functions Factory Setup - is similar to the standard factory setup, but the Gain Auto and the Exposure Auto auto functions are both enabled and are set to the continuous mode of operation. During automatic parameter adjustment, gain will be kept to a minimum.

- The Color Factory Setup - is optimized to yield the best color fidelity with daylight lighting.

The factory setups are saved in permanent files in the camera's non-volatile memory. They are not lost when the camera is reset or switched off and they cannot be changed.

You can select one of the four factory setups to be the camera's "default set". Instructions for selecting which factory setup will be used as the default set appear later in the Configuration Sets section. Note that your selection of which factory setup will serve as the default set will not be lost when the camera is reset or switched off.

When the camera is running, the default set can be loaded into the active set. The default set can also be designated as the "startup" set, i.e., the set that will be loaded into the active set whenever the camera is powered on or reset. Instructions for loading the default set into the active set and for designating which set will be the startup set appear later in the Configuration Sets section.
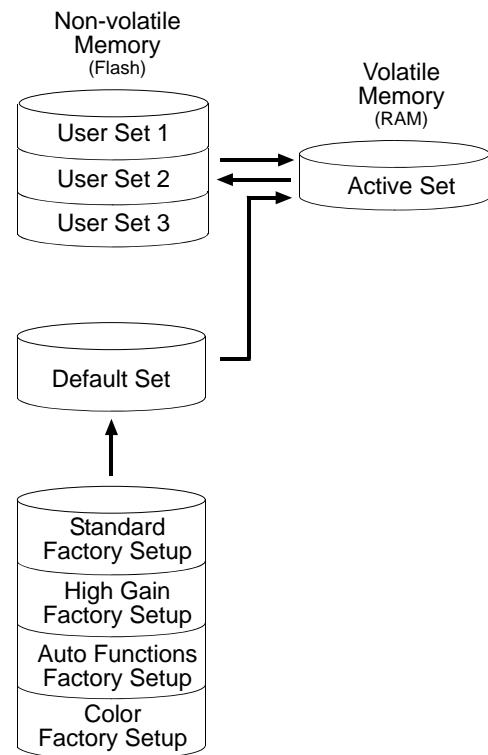
User Sets

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 that has been 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 set".

The three available user sets are called User Set 1, User Set 2, and User Set 3.

When the camera is running, a saved user set can be loaded into the active set. A saved user set can also be designated as the "startup" set, i.e., the set that will be loaded into the active set whenever the camera is powered on or reset. Instructions for loading a saved user set into the active set and for designating which set will be the startup set appear later in the Configuration Sets section.

> The values for the luminance lookup table are not saved in the user sets and are lost when the camera is reset or switched off. If you are using the lookup table feature, you must reenter the lookup table values after each camera startup or reset.

Designating a Startup Set

You can designate the default set or one of the user sets as the "startup" set. The designated startup set will automatically be loaded into the active set whenever the camera starts up at power on or after a reset. Instructions for designating the startup set appear below.

For more information about auto functions, see Section 12.9 on page 283.

12.18.1 Saving User Sets

Saving the current active set into a user set in 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 3.1.1 on .

12.18.2 Selecting a Factory Setup as the Default Set

When the camera is delivered, the Standard Factory Setup will be selected as the default set. You can, however, select any one of the four factory setups to serve as the default set.

To select which factory setup that will serve as the default set:

■   Set the Default Set Selector to the Standard Factory Setup, High Gain Factory Setup, Auto Functions Factory Setup or Color Factory Setup.

You can set the Default Set Selector from within your application software by using the Basler pylon API. The following code snippet illustrates using the API to set the selector:

If you want to select the Standard Factory Setup:

```
Camera.DefaultSetSelector.SetValue(DefaultSetSelector_Standard);
```

If you want to select the High Gain Factory Setup:

```
Camera.DefaultSetSelector.SetValue(DefaultSetSelector_HighGain);
```

If you want to select the Auto Functions Factory Setup:

```
Camera.DefaultSetSelector.SetValue(DefaultSetSelector_AutoFunctions);
```

If you want to select the Color Factory Setup:

```
Camera.DefaultSetSelector.SetValue(DefaultSetSelector_Color);
```

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

Selecting which factory setup will serve as the default 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.

Selecting the standard factory setup as the default set and then 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 standard factory setup is optimized for use in typical situations and will provide good camera performance in most cases.

12.18.3 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( );
```

> 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 with the Standard Factory Setup selected 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 standard factory setup is optimized for use in typical situations and will provide good camera performance in most cases.

12.18.4 Selecting the Startup Set

You can select the default configuration set (i.e., whichever was selected as the default configuration set, either the Standard Factory Setup, the High Gain Factory Setup, or the Auto Functions Factory Setup) or one of the user configuration sets stored in the camera's non-volatile memory to be the "startup set". The configuration set that you designate as the 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 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 );
```

12.19 Camera Feature Set

After the camera is powered on, pylon software processes a camera description file to make the camera features available for use. The camera description file is included in the camera and describes the camera features in accord with the GenICam specification.

scout GigE cameras include two camera description files that are used alternatively. The camera description files represent partially different combinations of features (feature sets).

To obtain the desired feature set for use you must select the related camera description file by setting the CameraFeatureSet parameter value and restarting the camera (see below).

> (i) The CameraFeatureSet parameter setting persists when the camera is switched off and on. Accordingly, the current camera description file will serve as the default camera description file whenever the camera is powered on, until the other camera description file is selected.

You can use either one of the following camera description files:

- The "Full" camera description file provides the complete feature set. All features will be available.
- The "Basic" camera description file provides all features **except** the following ones:
  - Matrix color correction (see Section 10.3.3 on page 181)
  - Color adjustment (see Section 10.3.4 on page 185)
  - Sequencer (see Section 12.5 on page 237)

  Accordingly, the features listed above will not be available.

Processing the selected camera description file takes some time. The time depends on the number and nature of the included features. Accordingly, the different camera description files require different periods to elapse until the camera features are available for use.

> (i) The camera features are sooner available when you use the Basic camera description file.

> (i) If you will only use the smaller feature set, we recommend to leave the CameraFeatureSet parameter value at the "Basic" factory setting. This will make the camera features sooner available.

To Activate a Camera Description File:

1. Select the camera description file by setting the CameraFeatureSet parameter value to either "Full" or "Basic", as desired.
2. Restart the camera by switching camera power off and on again.

    The feature set relating to the selected camera description file is available after some period.

Setting the Parameter Value

You can select a camera description file by setting the CameraFeatureSet 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 (in the example the camera description file providing the complete feature set is chosen):

```
// Select the camera description file providing all camera features
CEnumerationPtr ptrFeatureSet( Camera.GetNodeMap().GetNode("FeatureSet"));
ptrFeatureSet->FromString("Full");
```

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 parameter.

For more information about the pylon Viewer, see Section 3.1.1 on .

13 Chunk Features

This section provides detailed information about the chunk features available on each camera.

13.1   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.

13.2   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 Trigger Input Counter, the Time Stamp, the Line Status All, the CRC Checksum, and the Sequence Set Index 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 Basler 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 pylon Software Development Kit (SDK).

13.3   Frame Counter

The Frame Counter feature numbers frames sequentially as they are acquired. When the feature is enabled, a chunk is added to each frame containing the value of the counter.

The frame counter is a 32 bit value. The counter starts at 0 and increments by 1 for each acquired frame. The counter counts up to 4294967295 unless it is reset before (see below). After reaching the maximum value, the counter will reset to 0 and then continue counting.

Be aware that if the camera is acquiring frames 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.

> ⓘ 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 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 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 Basler 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 3.1.1 on .

Comparing Counter Chunk Data

When comparing trigger input counter data and frame counter data related to the same image, be aware that the trigger input counter initially starts at 1 whereas the frame counter starts at 0. Therefore, the trigger input count will always be ahead of the matching frame count by one if both counters were started at the same time and if an image was acquired for every trigger.

Whenever the counters restart after having reached 4294967295 they will both start another counting cycle at 0. Accordingly, the difference between matching counts will always be one, regardless of the number of counting cycles.

Note that if both counters were started at the same time and not reset since and if the trigger input counter is ahead of the matching frame counter by more than one, the camera was overtriggered and not all external triggers resulted in frame acquisitions.

Frame Counter Reset

Whenever the camera is powered off, the frame counter will reset to 0.

During operation, you can reset the frame counter via software or via I/O input line 1 or line 2. You can also disable the ability to perform a reset by setting the reset source to off. By default, the frame counter reset is disabled.

To use the frame counter reset feature:

- Configure the frame counter reset by setting the counter selector to Counter2 and setting the counter event source to FrameStart.
- Set the counter reset source to line 1, line 2, software, or off.
- Execute the command if using software as the counter reset source.

You can set the frame counter reset parameter values from within your application software by using the Basler pylon API. The following code snippets illustrate using the API to configure and set the frame counter reset and to execute a reset via software.

```
// configure reset of frame counter
Camera.CounterSelector.SetValue( CounterSelector_Counter2 );
Camera.CounterEventSource.SetValue( CounterEventSource_FrameStart );

// select reset by signal on input line 1
Camera.CounterResetSource.SetValue( CounterResetSource_Line1 );
```

```
// select reset by signal on input line 2
Camera.CounterResetSource.SetValue( CounterResetSource_Line2 );

// select reset by software
Camera.CounterResetSource.SetValue( CounterResetSource_Software );
// execute reset by software
Camera.CounterReset.Execute();

// disable reset
Camera.CounterResetSource.SetValue( CounterResetSource_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.

13.4   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.

> (i) 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 3.1.1 on .

13.5   Trigger Input Counter

The Trigger Input Counter feature numbers external image acquisition triggers sequentially as they are received. When the feature is enabled, a chunk is added to each image containing the related value of the trigger input counter.

The trigger input counter is a 32 bit value. On the first counting cycle, the counter starts at 1 and increments by 1 for each received trigger. The counter counts up to 4294967295 unless it is reset before (see below). After having reached the maximum value the counter will continue counting, starting at 0.

Be aware that if the camera is operating in continuous frame mode (free run) the trigger input counter will not be available.
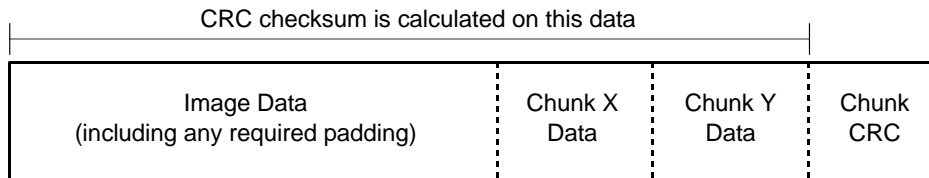
(i)	The chunk mode must be active before you can enable the trigger input counter feature or any of the other chunk feature. Making the chunk mode inactive disables all chunk features.

To enable the trigger input counter chunk:

▪ Use the Chunk Selector to select the Trigger Input Counter chunk.

▪ Use the Chunk Enable parameter to set the value of the chunk to true.

Once the trigger input counter chunk is enabled, the camera will add a trigger input counter 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 trigger input counter information by doing the following:

▪ Read the value of the Chunk Trigger Input 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 trigger input counter chunk, run the parser, and retrieve the trigger input counter chunk data:

```
// make chunk mode active and enable Trigger Input Counter chunk
Camera.ChunkModeActive.SetValue( true );
Camera.ChunkSelector.SetValue( ChunkSelector_Triggerinputcounter );
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 triggerinputCounter = Camera.ChunkTriggerinputcounter.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 3.1.1 on .

Comparing Counter Chunk Data

When comparing trigger input counter data and frame counter data related to the same image, be aware that the trigger input counter initially starts at 1 whereas the frame counter starts at 0. Therefore, the trigger input count will always be ahead of the matching frame count by one if both counters were started at the same time and if an image was acquired for every trigger.

Whenever the counters restart after having reached 4294967295 they will both start another counting cycle at 0. Accordingly, the difference between matching counts will always be one, regardless of the number of counting cycles.

Note that if both counters were started at the same time and not reset since and if the trigger input counter is ahead of the matching frame counter by more than one, the camera was overtriggered and not all external triggers resulted in frame acquisitions.

Trigger Input Counter Reset

Whenever the camera is powered off, the trigger input counter will reset to 0. During operation, you can reset the trigger input counter via I/O input 1, I/O input 2 or software, and you can disable the reset. By default, the trigger input counter reset is disabled.

To use the trigger input counter reset:

■ Configure the trigger input counter reset by setting the counter selector to Counter1 and setting the counter event source to FrameTrigger.

■ Set the counter reset source to Line1, Line2, Software or to Off.

■ Execute the command if using software as the counter reset source.

You can set the trigger input counter reset parameter values from within your application software by using the pylon API. The following code snippets illustrate using the API to configure and set the trigger input counter reset and to execute a reset via software.

```
// configure reset of trigger input counter
Camera.CounterSelector.SetValue( CounterSelector_Counter1 );
Camera.CounterEventSource.SetValue( CounterEventSource_FrameTrigger );

// select reset by signal on input line 1
Camera.CounterResetSource.SetValue( CounterResetSource_Line1 );
```

```
// select reset by signal on input line 2
Camera.CounterResetSource.SetValue( CounterResetSource_Line2 );


// select reset by software
Camera.CounterResetSource.SetValue( CounterResetSource_Software );
// execute reset by software
Camera.CounterReset.Execute();


// disable reset
Camera.CounterResetSource.SetValue( CounterResetSource_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.

13.6   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 95, 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.
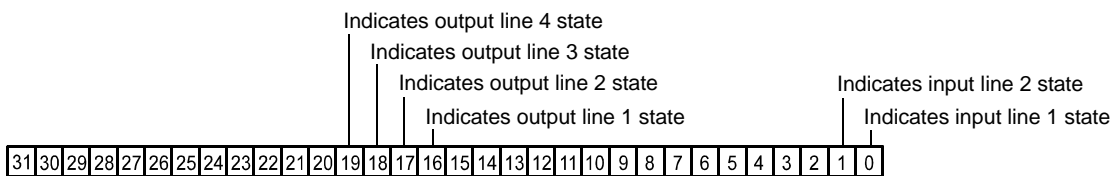
Fig. 95: Line Status All Parameter Bits

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 3.1.1 on page 43.

13.7   CRC Checksum

The CRC (Cyclic Redundancy Check) Checksum feature adds a chunk to each acquired image containing a CRC checksum calculated using the X-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. 96: CRC Checksum

> (i)   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 Basler 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 3.1.1 on .

13.8   Sequence Set Index

The Sequence Set Index chunk adds a chunk to each acquired frame containing the index number of the sequence set that was used for frame acquisition.

(i)	The sequencer feature must be enabled before you can enable the sequence set index feature. For more information about the sequencer feature, see the "Standard Features" section.

(i)	The chunk mode must be active before you can enable the sequence set index feature or any of the other chunk feature. Making the chunk mode inactive disables all chunk features.

To enable the sequence set index chunk:

■   Use the Chunk Selector to select the Sequence Set Index chunk.

■   Use the Chunk Enable parameter to set the value of the chunk to true.

Once the sequence set index chunk is enabled, the camera will add a sequence set index 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 sequence set index information by doing the following:

■   Read the value of the Chunk Sequence Set Index parameter.

You can set the Chunk Selector and Chunk Enable parameter value from within your application software by using the Basler 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 Sequence Set Index chunk
Camera.ChunkModeActive.SetValue( true );
Camera.ChunkSelector.SetValue( ChunkSelector_SequenceSetIndex );
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.ChunkSequenceSetIndex.GetValue();
```

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

14 Troubleshooting and Support

This chapter outlines the resources available to you if you need help working with your camera.

14.1 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. Basler 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 application notes in the Downloads and the Support sections of our website: www.baslerweb.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.

14.2 Obtaining an RMA Number

Whenever you want to return material to Basler, you must request a Return Material Authorization (RMA) number before sending it back. The RMA number **must** be stated in your delivery documents when you ship your material to us! Please be aware that if you return material without an RMA number, we reserve the right to reject the material.

You can find detailed information about how to obtain an RMA number in the Support section of our website: www.baslerweb.com

14.3 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 Basler 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. Basler 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   ☐ Once.   ☐ Every time.
    occur?
                                          ☐ 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   ☐ Yes   ☐ No
    without problems?

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 pylon 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	06 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 9.7 on page 137.
		Corrected the numbering in the drawing in Section 12.4 on page 234.
		Added information about the luminance lookup table feature in Section 12.6 on page 271.
AW00011903000	10 Jan 2007	Improved the content that describes network configuration and IP configuration.
		Updated the installation 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 24.
		Updated the description of the camera's network related parameters in Section 5.1 on page 57.
		Added Section 12.15 on page 306 to describe the new event reporting standard feature.
		Added Section 13.7 on page 336 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	08 Jun 2007	Added information on IP30, Section 1.5.1.1 on page 24 and Section 1.5.2.1 on page 29.
		Added information (drawings inclusive) on the 90° head housing variant: Section 1.5.1 on page 24.
		Added Section 1.5 on page 24 with information on mechanical stress test results.
		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 195, Section 11.2.4 on page 200, Section 11.2.5 on page 200 and Section 11.3.2 on page 203.
		Minor modifications throughout the manual: modified frame height to AOI height.

Doc. ID Number	Date	Changes
AW00011906000	12 Oct 2007	Modified mechanical drawings in Section 1.5.1 on page 24, Section 1.5.2 on page 29, Section 7.1 on page 75, Section 7.2.1 on page 76 (dimensions, holes for screw-lock connector), and Section 1.5.2.2 on page 31 (front module reference planes).

Added note on internal dust seal in Section 1.9 on page 38.

Added warning not to remove the serial number in Section Section 1.9 on page 38.

Updated times in Section Section 7.6.3 on page 85.

Corrected result of calculation example for formula 1 in Section 9.13.

Removed note on scA750-60 output in Sections 11.2.1, 11.2.4, 11.2.5, 11.3.2, 11.3.6, 11.3.8, and 11.3.7.

Corrected sequence of colors for Bayer BG 12 Packed format in Section 11.3.4 on page 207.

Minor corrections throughout the manual. |
| AW00011907000 | 21 Dec 2007 | Added guidelines for avoiding EMI and ESD problems in Section 2.3.1 on page 55.

Replaced recommendations of individual adapters by a reference to Basler technical support in Section 4.1 on page 67..

Removed web link for further information on APIPA in Section 5.3 on page 83.

Corrected the voltage ranges relating to logic 0 and logic 1 in Section 7.6.1 on page 81.

Added references to Application Notes AW000565 in Section 9.3.1 on page 108 and Section 9.9.1 on page 150.

Added minimum gain raw settings with vertical binning in Section 12.1 on page 219.

Added a reference to binning in Section 12.4 on page 234.

Added the Binning feature in Section 12.6 on page 271.

Added the Gamma feature in Section 12.11 on page 263.

Added the Disable Parameter Limits feature in Section 12.10 on page 297.

Added the Debouncer feature in Section 12.11 on page 298.

Minor corrections throughout the manual. |

Doc. ID Number	Date	Changes
AW00011908000	15 Feb 2008	Included the "Software Licensing Information" section on page 34. Moved the guidelines for avoiding EMI and ESD problems to Section 1.7 on page 36. Included the warning related to code snippets in Section 1.9 on page 38. Transferred the following sections to the "Installation and Setup Guide for Cameras Used with Basler's pylon API" (AW000611): "Software and Hardware Installation", "Network Recommendations", and "Camera and Network Adapter IP Configuration". Added the reference to the "Installation and Setup Guide for Cameras Used with Basler's pylon API" in Section 2 on page 41. Added the "Improve the Network Performance" step in Section 5.2.1 on page 63. Corrected the minimum value for the Timer Delay Raw parameter and indicated the minimum value for the Timer Delay Time Base Abs parameter in Section 8.2.4.2 on page 93. Minor modifications and corrections throughout the manual.
AW00011909000	05 Mar 2008	Integrated the scA1400-30gm/gc. Added information on the lens to which the mechanical shock tests apply in Section 1.5.4 on page 31. Added information on the input line transition threshold in Table 7 on page 77. Added the maximum exposure times and related settings in Section 9.7.1 on page 138. Minor modifications and corrections throughout the manual.

Doc. ID Number	Date	Changes
AW00011910000	18 Jul 2008	Integrated information for the scA1300-32 gm/gc.
		Expanded the voltage information in Section 1.2 on page 2.
		Updated the distances and related tolerances between the front of the lens mount and the sensor's photosensitive surface in Section 1.5 on page 24.
		Indicated the applicability of the mechanical stress test to cameras with standard housing and an extended duration of the broad-band vibration test in Section 1.5.4 on page 31.
		Modified the voltage information in Section 1.9 on page 38.
		Removed voltage information from Table 7 in Section 7.2.3 on page 77.
		Added Section 7.4.3 on page 79 and notes in Section 7.4.2 on page 77 introducing the PLC cable.
		Included detailed voltage information in Section 7.4 on page 80, Section 7.6.1.1 on page 81, and Section 7.6.2.1 on page 83.
		Modified the absolute maximum rating to +30.0 VDC in Section 7.6.1.2 on page 82 and Section 7.6.2.2 on page 83.
		Added a note relating to the debouncer in Section 9.12 on page 161.
		Updated the black level feature description and the Black Level Raw parameter range for the scA1400-30 in Section 12.2 on page 226.
		Corrected the minimum value for the white balance ratio in Section on page 227.
		Corrected the name of the Gamma parameter in Section 12.11 on page 263. (The Gamma parameter was incorrectly referred to as the Gamma Raw parameter.)
		Included the "Auto Functions" section on page 283 and added related information in other parts of the manual.
		Extended the description of the debouncer in Section 12.11 on page 298.
		Minor modifications and corrections throughout the manual.
AW00011911000	22 Aug 2008	Updated contact addresses and phone numbers.
		Official release of the auto functions.
AW00011912000	18 Feb 2009	Integrated information for the scA640-120 gm/gc.
		Added a note to Section 1.7 on page 36 indicating that an application note regarding EMI/ESD control is available.
		The text in Section 2 on page 41 and Section 3 on page 43 has been revised to reflect that the pylon driver package is now downloadable from the website.
		Added Section 13.2 on page 245 describing how to obtain an RMA number.

Doc. ID Number	Date	Changes
AW00011913000	16 June 2009	Updated sensor name "Micron MT9V022" to "Aptina MT9V022" in Section 1.2 on page 2.
		Removed the notes designating the information about the scA1300-32gm/gc cameras as preliminary in Section 1.2 on page 2.
		Updated minimum allowed exposure times in Section 9.7 on page 137.
		Updated the constants C_1, C_2, and C_3 for scA1300-32gm/gc cameras in Section 9.13 on page 163.
		Added the digital shift feature in Section 12.3 on page 228.
		Corrected the indications of x offset and y offset in Figure 71 in Section 12.4 on page 234 and in Figure 69 in Section 12.9.1.2 on page 285.
		Added the reverse X feature in Section 12.7 on page 274.
		Removed the statement that auto functions have no effect on frame rate in Section 12.9.1 on page 283 and added a note that frame rate may be affected if exposure auto is used in Section 12.9.3 on page 292.
		Added a reference to the reverse X feature in Section 12.9.1.2 on page 285.
		Replaced "Auto Gain Raw" by the correct "Gain Raw (All)" parameter name in Section 12.9.2 on page 290.
		Replaced "Auto Exposure Time Abs" by the correct "Exposure Time Abs" parameter name in Section 12.9.3 on page 292.
		Added the auto function profile feature in Section 12.9.4 on page 294 and adjusted Section 12.9.2 on page 290 and Section 12.9.3 on page 292 accordingly.
		Added the trigger delay feature in Section 12.13 on page 302.
		Added the acquisition status feature in Section 12.14 on page 304 and added a reference in Section 9.2.3 on page 106.
		Added the high gain and auto functions factory setups and the standard factory setup (formerly the "default set") in Section 12.18 on page 315.
		Removed the statement that settings for frame transmission delay and inter packet delay are not saved in the user sets in Section 12.18 on page 315.
		Added descriptions about resetting the frame counter and about relating frame and trigger input counter in Section 13.3 on page 326.
		Corrected the maximum value for the frame counter in Section 13.3 on page 326.
		Added the trigger input counter feature in Section 13.5 on page 331.

Doc. ID Number	Date	Changes
AW00011914000	14 Apr 2011	Integrated the scA1600-28gm/gc.
		Updated European and U.S. phone numbers and U.S. contact address.
		Indicated the availability of programming languages other than C++ for use with pylon in Section 1.9 on page 38.
		Added information about the relationship between the packet timeout and the inter-packet delay in Section 4.1 on page 46, Section 4.2 on page 47, and Section 5.1 on page 57.
		Added information about the payload and non-payload portion of the packet size in Section 5.1 on page 57.
		Changed the maximum allowed current for an output circuit to 50 mA in Section 7.6.2.2 on page 83.
		Modified input line assignment to take account of standard and legacy modes in image acquisition control in Section 8.1.1 on page 87.
		Added the Acquisition Trigger Wait signal to Section 8.2.1 on page 89.
		Re-wrote Section 9 on page 99 to describe acquisition start trigger and frame start trigger and to take account of standard and legacy modes in image acquisition control.
		Indicated that the maximum possible frame rate can not be achieved with the acquisition mode parameter set to single frame in Section 9.3 on page 105, Section 9.13 on page 163, and Section 9.14 on page 167.
		Updated the exposure start delay values and the frame readout time formula in Section 9.12 on page 161.
		Updated the C_2, and C_3 constants, formula 1, and the calculated example in Section 9.13 on page 163.
		Updated the C_1, C_2, and C_3 constants, formula 1, and the calculated example in Section 9.14 on page 167.
		Indicated the applicability of the gain calculation procedure for the scA640-120 and scA1300-32, corrected the dB gain values at max setting for the scA640-120 and scA1300-32, and included a calculated example in Section 12.1 on page 219.
		Indicated the effect of the black level parameter setting for the scA640-120 in Section 12.2 on page 226.
		Modified Section 12.13 on page 302 and Section 12.14 on page 304 to take account of standard and legacy mode in image acquisition control.
		Added events and parameter names in Section 12.15 on page 306, and removed some descriptions for transfer to the "Camera Events" code sample.
		Rearranged sequence of sections 8 through 11.
		Removed the feedback page.

Doc. ID Number	Date	Changes
AW00011915000	13 Jul 2011	Removed information related to the scA1000-20gm/gc cameras throughout the manual. Simplified Figure 38 and Figure 39 in Section 7.6 on page 81. Removed the "Typical LED Output Signal..." figure from Section 7.6.2.2 on page 83. Simplified the definition of response times, removed ranges of response time values, included a measured example, indicated general dependences of response times, and removed the "I/O Line Schematic" figure in Section 7.6.3 on page 85.

Doc. ID Number	Date	Changes
AW00011916000	14 Dec 2012	Indicated Basler AG as bearer of the copyright on the second page.
		Added information about the 90° head housing model for the scA1600-28gm/gc in Section 1.2 on page 2 and Section 1.5.2.2 on page 31.
		Removed abs. max. voltages from Section 1.2 on page 2, Section 1.9 on page 38, Section 7.4 on page 80, and Section 7.6 on page 81.
		Updated the upper limit for transmission in the IR cut filter recommendations in Section 1.4 on page 18.
		Added references to Section 12.19 on page 321 in sections 2,10.3.3, 10.3.4, and 12.5.
		Added AC In lines and Gnd to the power supply and cable shield in Fig 38 in Section 7.3.2 on page 78.
		Replaced max. input voltage by voltage range in Figure 38 in Section 7.6.1.2 on page 82.
		Added a current limiter to Figure 38 and a transistor to Figure 39 in Section 7.6 on page 81.
		Added inversion of the input lines in Section 8.1.3 on page 88.
		Removed all text referring to overlapped exposure from Section 9.11.4 on page 155.
		Added the "Color Creation and Enhancement" Section 10 on page 171.
		Transferred the following sections to Section 10 on page 171:
		◾ The Bayer Color Filter ◾ White Balance ◾ Integrated IR Cut Filter ◾ Gamma
		Included white balance reset in Section 10.3.1 on page 177.
		Included sRGB gamma correction in Section 10.3.2 on page 179.
		Included the RGB 8 Packed pixel data format with information wherever applicable in the manual, particularly in Section 11.1 on page 193 and Section 11.3.5 on page 209.
		Added the "Sequencer" Section 12.5 on page 237.
		Added a note that auto functions will not work when the sequencer is enabled in Section 12.9 on page 283.
		Added the " Minimum Output Pulse Width" Section 12.12 on page 300.
		Added the "Camera Events" Section 9.11.6 on page 160.
		Updated the exposure start delay value for the scA1600-28gm/gc in Section 9.12 on page 161.
		Added the AcquisitionStartEventData and FrameStartEventData events in Section 12.15 on page 306.
		Added the sensor board temperature device information parameter in Section 12.17 on page 313.
		Added the color factory setup in Section 12.18 on page 315.
		Added the "Camera Feature Set" section 12.19 on page 321.
		Corrected the method used for calculating the CRC checksum to "X-modem" in Section 13.7 on page 336.
		Added Section 13.8 on page 338 describing the sequence set index chunk and modified Section 13.2 on page 324 accordingly.

Doc. ID Number	Date	Changes
AW00011917000	28 May 2014	Updated the Basler contact details on the back of the front page.
		Added the "LZ4 Licensing" Section 1.6.2 on page 35.
		Added information to the "Avoid dust on the Sensor" notice in Section 1.9 on page 38.
		Updated the I/O voltage levels for the standard and PLC power and I/O cables in Section 7.6.1.1 on page 81.
		Updated the URL for the color matrix reference article in Section 10.3.3.1 on page 183.
		Corrected the average gray value range for the Auto Function AOI in Section 12.9.2 on page 290 and Section 12.9.3 on page 292.
AW00011918000	23 Jan 2015	Updated the Basler contact details on the back of the front page.
		Minor corrections in the FCC Rules section on the back of the front page.
		Added references to the CE Conformity Declaration in Section 1.2 on page 2.
		Added information about the available software to the camera specifications in Section 1.2 on page 2.
		Updated pylon-related details and language conventions:
		▪ Renamed "pylon driver package" to "pylon Camera Software Suite".
		▪ Renamed "IP Configuration Tool" to "pylon IP Configurator".
		▪ Updated pylon SDK content information in Section 3.1.3 on page 44.
		▪ Updated the title of the AW000611 document.
		Merged the "Connector Pin Assignments and Numbering" and "Connector Types" sections to form Section 7.2 on page 76 ("Camera Connector Types, Connection Numbering, and Assignments").
		Merged the "Standard Power and I/O Cable" and "PLC Power and I/O Cable" sections to form Section 7.3.2 on page 78 ("Power and I/O Cable").
		Added information about using a suitable power and I/O cable in Section 7.3.2 on page 78.
		Added information related to the maximum wire diameter in Section 7.3.2 on page 78.
		Updated information related to the maximum cable length in Section 7.3.2 on page 78.
		Removed information related to the PLC I/O cable voltage levels in Section 7.6.1 on page 81.

Index

A